Київський національний університет імені Тараса Шевченка
Кафедра теорії ймовірностей, статистики та актуарної математики

Р. Майборода

КОМП'ЮТЕРНА СТАТИСТИКА

(з використанням R)

(Версія від 15.02.2016)

Київ — 2016

Передмова

Це — робоча версія підручника, призначеного для студентів спеціальностей статистика, актуарна та фінансова математика, математика. Не всі його розділи повністю закінчені, а деякі — іще не початі. Тому прохання, не розміщувати цю версію у відкритому доступі, а всіх охочих отримати підручник направляти до автора за останньою версією.

Я буду вдячний за всі зауваження, доповнення, виправлення.

Зміст

Розділ 1

Початок роботи з системою R

1.1 Що таке R і де його роздають

R це середовище програмування для статистичного аналізу даних. Це середовище складається з базової програми R, що працює як інтерпретатор мови статистичного програмування S та окремих пакетів, які реалізують спеціальні методи та технології статистичної обробки даних. Базова програма створена у рамках проекта у рамках проекту GNU, як альтернативна програмна реалізація мови S (ця мова та комерційний пакет S+ для її реалізації були розроблені у Bell Laboratories під керівництвом Дж. Чемберса). На відміну від S+, програма R є некомерційною і вільно розповсюджується за умови дотримання вимог GNU General Public License. Комерційний проект S+ нині практично не активний, остання версія програми випущена у 2007р. Подальший розвиток ідей, закладених у мові S та їх реалізація продовжується в рамках системи R . Тому сучасна версія мови також має назву R . Але ряд книжок, написаних з орієнтацією на S та S+ зберігає свою актуальність, оскільки у них питання прикладного застосування часто пояснюються детальніше і зрозуміліше ніж у документації до R , розробленій ентузіастами.

Офіційна сторінка проекту R http://www.r-project.org/ . Отримати останню версію інсталятора базової програми R для операційної системи Windows можна за адресою: http://cran.r-project.org/bin/windows/base/ . (На 20 січня 2016 це була версія R-3.2.3). Інсталятор завантажується у вигляді exe-файлу. Для інсталяції програми досить запустити цей файл і відповідати на його запитання. При першій спробі роботи з R рекомен-

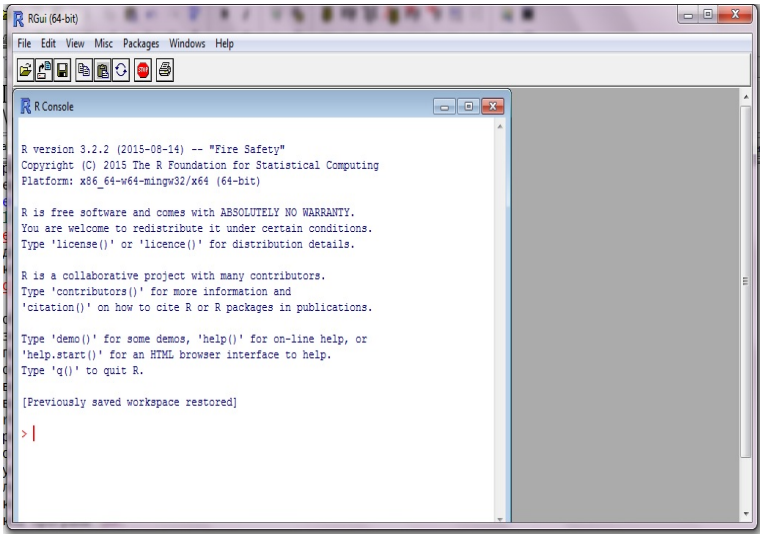

Рис. 1.1: Початок роботи з R

довано погоджуватись з усіма пропозиціями, які робить інсталятор.

Проблеми можуть виникнути, якщо на вашому комп'ютері встановлені різні права доступу для різних користувачів. Справа в тому, що R наприкінці кожної сесії роботи зберігає на диску "робочий простір" (workspace) - сукупність даних та програм, які були завантажені під час сесії. На початку наступної сесії workspace завантажується з диску. Якщо під час інсталяції для зберігання workspace буде обрано директорію, недоступну певному користувачеві, то при роботі з R можуть виникати повідомлення про неможливість завантаження або зберігання workspace. Для усунення таких повідомлень потрібно або вибрати директорію вільного доступу при інсталяції, або змінити директорію, використовуючи пункт File->Change Dir... у головному меню головного вікна програми R .

Після інсталяції R його можна запустити і отримати приблизно таке вікно, як зображено на рис. 1.1. Тут вгорі знаходиться головне меню, а нижче відкрито вікно "консолі R "у якій можна давати команди програмі та отримувати її відповіді. Синім кольором у цьому вікні виведено початкову інформацію про вашу версію базової програми R . Далі червоним кольором можуть бути вказані команди, які R виконав автоматично при завантаженні. Нарешті, червоний символ > є запрошенням користувачу вводити власні команди. Для перевірки роботи системи можна після >

ввести 2+2 і натиснути Enter. Результат буде виведено на консоль:

```
[1] 4
```

R виводить результати виконання безпосередньо після команди синім кольором, після чого переходить у режим очікування наступної команди, про що повідомляє червоним знаком >. При роботі з R можна виконувати одразу багато команд, що записані у окремому файлі. Найпростіший спосіб зробити це - завантажити такий файл в якому-небудь текстовому редакторі, зробити там сору, а потім - paste на консолі. При цьому, якщо команди у файлі розміщені у окремих рядочках, розділових знаків між ними не потрібно. Команди, вміщені в одному рядочку, розділяють символом ;.

Якщо довга команда не вміщується у одному рядочку, її можна розбити на декілька рядочків, причому, при переході до наступного рядочку R автоматично виводить символ продовження +. R сам здогадується, що команда не закінчена за її синтаксисом. Тому деякі синтаксичні помилки (як от - забуті дужки) можуть сприйматись як незакінчені команди. У цьому випадку R виставить + на початку наступного рядочка і перейде у режим очікування. Натисніть esc щоб перейти у режим введення нової команди без продовження аналізу попередньої.

Програми, що складаються з команд R називають скриптами (script). Вони мають стандартне розширення .r. У базовій програмі є можливість відкрити вікно редактора для створення нового скрипту, або завантажити файл зі скриптом, використовуючи пункти головного меню File->New script або File->Open script. Виконати завантажений у вікні редактора скрипт повністю можна, використовуючи Edit->Run all. Можна також виконати виділену частину скрипту використовуючи кнопку “Run line or selection”. Закінчивши роботу зі скриптом, його можна зберегти, використовуючи File->Save.

1.2 Система R-Studio

За потреби, всі технології статистичної обробки можна реалізовувати використовуючи лише базовий пакет R . Але він спеціально розроблений так, щоб забезпечувати лише мінімально необхідні засоби реалізації. Для більш зручного користування R -технологіями можна використовувати спеціальні надбудови-оболонки над R , які дають більше можливостей для програмування, проглядання використаних змінних, користування

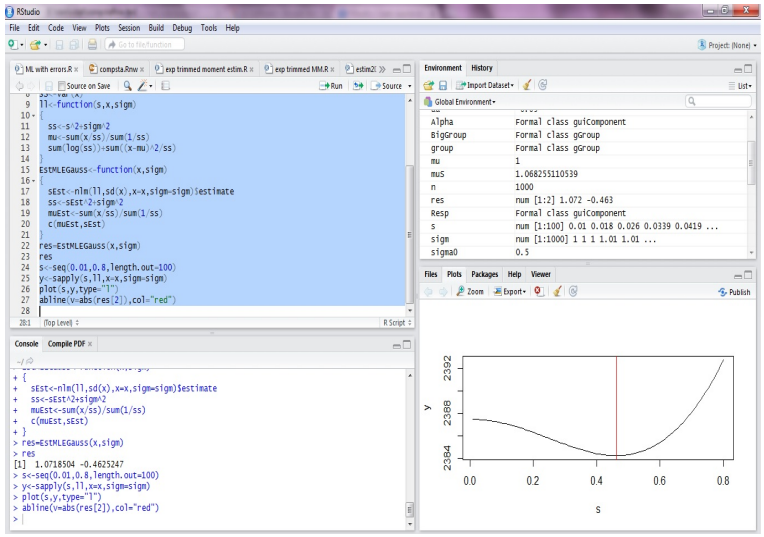

Рис. 1.2: Початок роботи з `R`

графікою та роботи з Help-системою.

Такою оболонкою-інтегратором є система `R-Studio`. Вона також розповсюджується безкоштовно. Інсталятор `R-Studio` можна отримати на офіційному сайті `www.rstudio.com`. Перш, ніж інсталювати цю програму, треба встановити на комп'ютері базовий `R` . Після цього можна запустити інсталятор `R-Studio` і погоджуватись з усіма його запитаннями.

При роботі вікно `R-Studio` може мати приблизно такий вигляд, як зображено ра рис. 1.2. Основне вікно розділене на чотири дочірних вікна. У лівому верхньому вікні виведено script — програму, яка редагується. (У цього вікна багато закладинок, які дозволяють працювати з кількома файлами-скриптами одразу). У лівому нижньому вікні — консоль, у якій виконуються команди. Тут же можна запускати скрипти або їх частини. У правому верхньому вікні можна переглядати активні змінні, з якими працює програма. Тут також можна побачити історію роботи.

Найбільш навантажене вікно внизу праворуч. Сюди виводять рисунки, які робить програма, тут можна проглянути Help, подивитись, які додаткові пакети завантажені, а також працювати з різними файлами з вашого комп'ютера.

Звичайно, користувач може міняти ці вікна місцями, змінювати їх розміри та користуватись іншими можливостями системи.

Зокрема, користуючись головним меню, можна перезавантажувати `R`

, вибирати новий робочий каталог (тобто каталог, з якого `R` завантажує файли за умовчанням), зберігати та завантажувати у пам'ять workspace, отримувати з інтернету нові бібліотеки програм/даних (packages). В принципі, все це можна робити і безпосередньо з `R` , але в R-Studio такі речі організовані зручніше.

R-Studio корисний також підказками, які він робить під час набору команд на консолі та у вікні скриптів.

Іще одна додаткова зручність R-Studio — можливість генерації текстових звітів, які виконуються з поєднанням системи програмування `R` та системи форматування текстів LaTeX. При цьому R-Studio спирається на `R` пакет `knitr`. За допомогою цієї технології підготовлена дана книга. Нажаль, для повного опису `knitr` потрібно пояснювати не тільки роботу `R` , а і принципи організації LaTeX, що виходить за рамки цієї книги.

1.3 Завантаження пакетів, робота з Help та інші організаційні питання

Базовий `R` має великий набір функцій для реалізації математичних та статистичних алгоритмів. Але користувачі весь час розробляють свої власні функції, що доповнюють базові. Коли деякий набір набір функцій, що реалізують певну технологію статистичної обробки даних буде відпрацьований настільки, що у розробника виникає бажання поділитись ним із іншими можливими користувачами, він оформлює такий набір у вигляді пакету (package). Пакет повинен мати також help-документацію, яка дозволить можливим користувачам зрозуміти його призначення. До пакету часто включають і набори даних, на яких можна перевірити роботу його функцій. Бувають пакети, складені лише з даних — це просто колекції цікавих або популярних прикладів, які хтось підібрав для власних потреб.

Правильно оформлені пакети розробники відсилають до депозитаріїв, звідки їх можна переписати на свій комп'ютер у каталог, доступний для `R` (інсталювати). Оскільки пакети створюються різними розробниками за власною ініціативою, незалежно один від одного, між ними можуть існувати неузгодженості. Наприклад, функції з різних пакетів можуть мати однакові імена та типи параметрів, тоді при завантаженні у пам'ять комп'ютера обох пакетів користувач не зможе правильно їх ви-

користовувати[1]. Тому рекомендується завантажувати не всі інстальовані на комп'ютері пакети, а лише ті, які дійсно потрібні для роботи під час даної сесії.

Просунуті користувачі `R` розрізняють поняття *пакет* (package) і *бібліотека* (library). Пакетом називають файл, або набір файлів з скриптами та їх описом, а бібліотекою - місце, тобто каталог у файловій системі, де лежить пакет. З точки зору користувача-початківця ця відмінність несуттєва. Старожили пам'ятають, що в мові S library означало приблизно те ж, що у `R` зветься package. У цій книжці ми теж не будемо надавати ваги цій відмінності.

Для того, щоб інсталювати пакет на комп'ютері, тобто отримати його з інтернету у вигляді zip-архіву, розархівувати і покласти у зручне для `R` місце, можна:

1. Під час сесії роботи з `R` з консолі, викликавши функцію `install.packages`. Наприклад, команда `install.packages('raster')` викличе звертання комп'ютера до стандартного депозитарію (як правило, це `cran.us.r-project.org`), отримання від нього пакету і розміщення його у відповідному каталозі на комп'ютері. Звичайно, якщо комп'ютер не має виходу в інтернет, або депозитарій недоступний, в результаті виконання функції виникне помилка.

2. При роботі безпосередньо з базовим `R` інсталяцію можна робити використовуючи пункти головного меню Packages->Install package(s). Спочатку програма пропонує вибрати інтернет-архів, з якого робиться інсталяція. Варіант 0-cloud, що пропонується за умовчанням, як правило, працює цілком задовільно. Після цього треба у списку вибрати потрібний для вас пакет. Якщо цей пакет використовує які-небудь інші, котрих немає на вашому комп'ютері, вони будуть інстальовані автоматично.

3. При роботі з R-Studio, можна скористатись пунктами головного меню Tools->Install packages... При цьому відкривається діалогове вікно, де ви можете вказати, звідки проводиться інсталяція (з інтернет-депозитарію, чи з zip-архіву на вашому комп'ютері) який пакет ви хочете інсталювати, місце, де буде розміщений пакет і чи треба інсталювати інші пакети, які ним використовуються.

Видалити непотрібний пакет з каталогу можна використовуючи функ-

[1]Функції, що мають однакові імена але працюють з параметрами різних типів — це нормальне явище для об'єктно-орієнтованих мов. Комп'ютер при виклику обирає правильну функцію виходячи з специфікації її параметрів.

цію `remove.packages()`.

Для завантаження (підключення) пакету у пам'ять під час сесії, використовують функцію `library()`. Наприклад, `library(raster)` підключає пакет +raster+ і дає змогу використовувати всі його функції у подальшій сесії.

Відключити пакет можна використовуючи функцію `detach()`. Так `detach("package:raster")` зробить пакет `raster` неактивним — його функції перестануть бути доступними у подальшій сесії.

Для того, щоб отримати довідку по яким-небудь можливостям `R` можна скористатись help-системою. Для цього призначена функція `help()`, або скорочено `?`. Набравши на консолі `R`

`?sin` ви отримаєте довідку про тригонометричні функції в `R` , зокрема — про і про функцію $\sin(x)$. Довідка, як правило, починається з інформації про те, у якому пакеті знаходиться функція (дані, об'єкти, тощо) У випадку `?sin` це виглядає як

`Trig {base}` що вказує на набір тригонометричних функцій з базового `R` .

Інколи назву функції (тему help) після знаку запитання потрібно задавати у лапках. Так, при спробі викликати help запитом `?+` (або `?for`) ви у відповідь отримаєте запрошення продовжувати введення команди (+). Якщо набрати `?"+"` (відповідно — `?"for"`) можна отримати довідку про реалізацію арифметичних операцій (або про цикл `for`) у `R` .

Базовий `R` для перегляду help-документації може запускати інтернет-браузер, але це не означає, що документація шукається у інтернеті. Все, що видається за командою `?` або `??` знаходиться на вашому комп'ютері і не потребує доступу до інтернету.

Команда `?` виводить основний файл, пов'язаний з темою запитання. Якщо ви хочете проглянути всі файли, де згадується дана тема, можна скористатись функцією `help.search()`, скорочено — `??`. Наприклад, за запитом

`??"linear models"` у браузері буде виведена сторінка з переліком усіх документів help-у де згадуються лінійні моделі з коротким описом їх змісту. Більшість з цих сторінок буде стосуватись лінійних регресійних моделей, або узагальнених лінійних моделей. Але можливі і посилання на лінійні моделі чогось зовсім іншого. Переходячи за гіперпосиланнями можна продивлятись ці документи.

Задаючи спеціальні параметри функцій `help()` або `help.search()` можна отримувати довідки по окремих пакетах, або тільки за ключовими

словами, або тільки по документах з певного каталогу і т.д.

Документацію для help автори пакетів розробляють самі і поставляється вона разом з пакетами. Тому за запитом `?` ви отримуєте інформацію лише про ті функції, які знаходяться у пакетах, доступних під час сесії (підключених при запуску `R` або додатково командою `library()`). Якщо, скажімо, на початку сесії я запитаю

`?ginv` відповідь буде

```
No documentation for ‘ginv’ in specified packages and libraries:
you could try ‘??ginv’
```

При запиті `??ginv` виводиться вся інформація про `ginv`, що є на комп'ютері (як підключена до сесії, так і не підключена). Зокрема, на моєму комп'ютері на сторінці довідки з'являється гіперпосилання

`MASS::ginv Generalized Inverse of a Matrix` що вказує на наявність функції `ginv` у пакеті `MASS` і коротко описує її призначення — знаходження узагальнених обернених матриць. А після підключення пакету `MASS` довідка про цю функцію стане доступною за запитом `?ginv`. При роботі в R-Studio комп'ютер стане підказувати вам параметри цієї функції при наборі і т.д.

У R-Studio у вікні help (праворуч знизу на екрані у стандартній конфігурації) є поле для пошуку (Search) яке діє аналогічно запиту `?` але при цьому дає додаткову підказку при наборі.

Якщо ви хочете отримувати інформацію про можливості всіх функцій з усіх пакетів, що лежать у всіх доступних депозитаріях, то ви можете інсталювати на своєму комп'ютері пакет `sos`. Після підключення його до сесії (`library(sos)`) стане доступним запит у формі `???`<тема> за яким буде видаватись результат пошуку заданої теми по всіх `R` -депозитаріях світу. Розібратись у таких багатосторінкових переліках буває не просто, але інколи вони дають несподівані і дуже корисні результати.

Оскільки документацію до пакетів розробляють їх автори, то вона часто буває переобтяженою технічними подробицями не дуже зрозумілими початківцю. Логіка застосування програми (очевидна авторам) при цьому втрачається. Тому дуже корисним буває ознайомлення з думками користувачів. Найпростіше знайти такі думки скориставшись якою-небудь пошуковою інтернет-машиною (я віддаю перевагу Google). Набравши, скажімо, запит

`"inverse matrix in r"`

ви отримаєте посилання на багато різних рекомендацій по знаходженню обернених матриць за допомогою R . Не всі вони будуть адекватними! Я рекомендую звертати увагу на рекомендації сайтів:

`stackoverflow.com`

— це сайт програмістів та математиків, тут можна знайти поради спеціалістів та обговорення проблем на серйозному рівні.

`www.statmethods.net`

— тут можна шукати швидкі і прості поради у стилі Quick-R.

`cran.r-project.org/doc/FAQ/`

— це офіційний сайт R , місце де зібрані відповіді на запитання, що виникають особливо часто.

На ютубі можна також побачити лекції з багатьох окремих питань використання R у стилі "зрозуміло навіть немовлятам". Вони можуть бути корисними на перших етапах вивчення R , щоб не почувати себе зовсім безпорадним. Потім їх зрозумілість починає дратувати. Але коли ви набуваєте певного досвіду у роботі з R і виникає потреба поділитись ним з іншими, перегляд таких лекцій знову може стати у пригоді.

Розділ 2

Мова статистичного програмування R

Мова R складається з *команд*. Кожна команда може виконуватись окремо, або у складі програми. Програми у R звуться *скриптами* (script). Окрема команда записується у командному рядочку системи R після *запрошення* "`>`" і запускається на виконання клавішею Enter:

```
> 1+1

[1] 2
```

(Запрошення комп'ютер видає автоматично).

Команда може бути *виразом* (тоді результат її виконання просто виводиться на екран, як у попередньому прикладі) або привласненням:

```
> x<-1+1
> x

[1] 2
```

Тут `<-` це символ привласнення, праворуч від нього іде вираз, значення якого обчислюється, а ліворуч — ім'я змінної, якій привласнено обчислене значення. Саме значення не виводится на екран, щоб побачити, чому тепер дорівнює змінна `x`, ми ввели її назву у наступному рядочку після запрошення.

Команди виконуються після натискання на клавішу Enter. Коли при цьому комп'ютер за синтаксисом помічає, що команда не закінчена, він

у наступному рядочку замість запрошення “>” виводить символ продовження вводу “+” і ви можете закінчити введення команди:

```
> x<-2*
+ 3
> x

[1] 6
```

2.1 Типи даних та елементарні функції

2.1.1 Вектори. Арифметичні та логічні операції.

Найпростішою структурою у мові R є вектор (скаляри як окремі структури не існують, а трактуються як вектори одиничної довжини).

R використовує п’ять простих векторних типів об’єктів:

- `logical`: логічний — вектор складений з елементів що приймають значення істинно (`TRUE` або `T`) та хибно (`FALSE` або `F`);
- `numeric`: числовий — вектор, складений з дійсних чисел;
- `integer`: цілий — вектор, складений з цілих чисел;
- `complex`: вектор, складений з комплексних чисел;
- `character`: символьний — вектор, елементами якого є символьні рядочки.

Якщо в одному наборі даних потрібно об’єднати елементи різної природи, використовують об’єкт типу `list` — список.

Створити будь-який вектор (список) можна використовуючи функцію `c()`, яка об’єднує різні списки в один об’єкт (конкатенація):

```
>  c(1,5,-3,4)

[1]  1  5 -3  4
```

(з чотирьох одноелементних векторів створений числовий вектор що складається з чотирьох елементів (1,5.-3,4) і результат роботи виведено на екран).

З числовими векторами можна виконувати звичайні дії додавання, множення і т., д. З логічними — операції & (логічне і), | (логічне або), ! (заперечення) та ін. Операції порівняння (<, >, <=,=>,==,!=) застосовуються до числових даних і дають логічний результат.

Такі операції застосовуються до векторів поелементно:

```
>   x<-c(1,5,-3,4)
>   y<-c(3,-1,2,1)
>   x+y

[1]  4  4 -1  5
```

Також поелементно застосовуються до векторів елементарні функції `sin`, `log` і т.д.

Для цілочисельного ділення використовується операція `%/%` для знаходження залишку від ділення — `%%`. Якщо у бінарній операції вектори-аргументи мають різну довжину, то коротший аргумент повторюється циклічно при виконанні операції:

```
>   x<-c(1,2)
>   y<-c(3,3,3,3,3)
>   x*y

Warning in x * y: длина большего объекта не является произведением длины меньшего

[1] 3 6 3 6 3
```

(При цьому, якщо довжина довшого вектора не кратна довжині коротшого, комп'ютер видає попередження (warning) про це на зразок наведеного у прикладі.

При виконанні арифметичних дій можуть виникати значення Inf (нескінченність) та NaN (невизначено). З ними можна виконувати різні дії, які дають осмислений результат:

```
>   x<-1/0
>   x
```

```
[1] Inf

>   3-x

[1] -Inf

>   x>3

[1] TRUE

>   x/x

[1] NaN
```

Крім значення NaN, яке відповідає невизначеності, пов'язаній з арифметичними операціями, в R використовується також значення NA, що позначає *пропущені значення*, тобто значення, які є невідомими статистику. Особливості обробки NA-значень обговорюються далі. Крім того, можливе іще значення NULL, яке позначає пустий список.

Вектори можуть бути іменованими (named), у такому випадку, кожен елемент вектора має ім'я. Щоб зробити вектор іменованим, потрібно задати для нього *атрибут* names:

```
>  x<-c(5,4,3,2,1)
>  names(x)<-c("відмінно","добре","задовільно","незадовільно", "погано")
>  x

    відмінно        добре   задовільно незадовільно       погано
           5            4            3            2            1
```

(Тут виклик функції стоїть ліворуч від знаку привласнення. У R такий синтаксис дозволений лише у невеликій кількості випадків. Як правило, вживання виклику довільної функції ліворуч від <- трактується як помилка). Використання іменованих векторів часто буває зручним саме у статистичних застосуваннях, зокрема, при звертанні до того чи іншого елемента вектора або масиву складнішої структури.

Відмітимо дві зручні функції для створення векторів. Якщо потрібен вектор, елементи якого утворюють арифметичну прогресію, можна скористатись функцією seq():

```
> seq(2.5,6,0.5)

[1] 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
```

Виклик `seq` можливий у різних форматах (це характерна особливість не лише `seq` а всіх функцій мови `R`) Формальна специфікація цієї функції така:

```
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)),
   length.out = NULL, along.with = NULL, ...)
```

У цьому записі `seq` — ім'я функції, `from` (перший елемент), `to` (останній), `by` (крок), `length.out` (кількість елементів) і `along.with` — імена формальних параметрів. Після знаку рівності вказані значення, яких ці параметри набувають за умовчанням, якщо вони не вказані у виклику функції. Наприклад, можливий виклик:

```
> seq(2,10,length.out=6)

[1]  2.0  3.6  5.2  6.8  8.4 10.0
```

Тут крок прогресії не заданий явно, він обирається комп'ютером так, щоб кількість елементів дорівнювала заданому `length.out`.

(`...` позначає, що у функції можуть бути і інші параметри).

Як працюватиме ця функція при виклику з іншими наборами параметрів можна подивитись у help, задавши команду `?seq`.

Для випадку, коли крок послідовності дорівнює ± 1, можна використовувати скорочений запис `seq` у вигляді `from:to`, наприклад:

```
> 5:10

[1]  5  6  7  8  9 10

> -5:10

 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5  6  7  8  9 10

> -(5:10)

[1]  -5  -6  -7  -8  -9 -10
```

```
> -10:-5

[1] -10  -9  -8  -7  -6  -5
```

Функція `rep()` розмножує свій перший параметр задану кількість разів:

```
> x<-1:4
> rep(x,3)

 [1] 1 2 3 4 1 2 3 4 1 2 3 4

> rep(x,each=3)

 [1] 1 1 1 2 2 2 3 3 3 4 4 4
```

Числовий вектор, що складається з нулів можна створити також функцією `numeric(n)`, де `n` — кількість елементів вектора:

```
> x<-numeric(5)
> x

[1] 0 0 0 0 0
```

2.1.2 Індексація векторів.

Для того, щоб при обробці мати можливість використати певну частину вектора (матриці, багатовимірного масиву), у `R` застосовується дуже гнучка система індексації. Зараз ми обмежимось прикладами її застосування для векторів, матриці розглянемо далі. Як звичайно, i-тий елемент вектора можна виділити використовуючи прямі дужки:

```
> x<-5:1
> names(x)<-c("відмінно","добре","задовільно","незадовільно", "погано")
> x[2]

добре
    4
```

(Нумерація елементів векторів завжди починається з 1).

Можна звернутись до елемента за ім'ям, якщо воно є:

```
> x["задовільно"]

задовільно
         3
```

Якщо у прямих дужках вказати вектор індексів, то виділиться підвектор відповідних елементів:

```
> x[c(3,1,5)]

задовільно   відмінно     погано
         3          5          1
```

(елементи переставлені в тому порядку, в якому йдуть індекси). І нарешті:

```
> x[-c(3,1,5)]

       добре незадовільно
          4            2
```

— якщо задати від'ємні значення індексів, то відповідні елементи будуть вилучені з підвектора.

Це ще не все. Можна для індексації використовувати логічні вектори, тоді включатись у підвектор будуть тільки елементи, яким відповідає значення TRUE:

```
> x[c(T,T,F,F,T)]

відмінно    добре   погано
       5        4        1
```

У прямих дужках можна записувати будь-який вираз, значення якого будуть використані для індексації:

```
> x[x%%2==0]

       добре незадовільно
          4            2
```

Роботу цієї команди можна описати так: спочатку створюється логічний вектор `x%%2==0` в якому TRUE відповідає тим елементам, які є парними числами, а потім за цим логічним вектором робиться відбір відповідних елементів у підвектор.

Вираз вигляду `x[...]` може стояти і у лівій частині команди привласнення, наприклад:

```
> alp<-c('a','b','c','d','e','f')
> alp[2]<-'bbb'
> alp

[1] "a"   "bbb" "c"   "d"   "e"   "f"

> alp[c(1,3)]<-c('u','v')
> alp

[1] "u"   "bbb" "v"   "d"   "e"   "f"
```

2.1.3 Фактори.

Іще один векторний тип даних — *фактори* (`factors`) заслуговує спеціального розгляду. Елементи вектора факторів можуть приймати значення лише з фіксованого набору значень. Дані такого типу часто виникають у статистичних дослідженнях, коли досліджувані об'єкти розбиваються на кілька груп (категорій) за деякою ознакою, наприклад — люди за національністю, статтю відношенням до військової служби, юридичні особи — за формою власності, слова — за частинами мови (іменник, прикметник, дієслово...) тощо. Різні значення, які може приймати фактор, прийнято називати рівнями (levels).

Різні рівні зручно позначати їх назвами, наприклад, тип валюти — USD, EUR, UAH, RUR. Скажімо, набір даних про тип валют, якими було зроблено платежі протягом дня може мати вигляд:

('USD','EUR','EUR','UAH','EUR','USD','UAH','RUR').

Якщо задати такий вектор конкатенацією

```
> z<-c('USD','EUR','EUR','UAH','EUR','USD','UAH','RUR')
> z

[1] "USD" "EUR" "EUR" "UAH" "EUR" "USD" "UAH" "RUR"
```

то z буде мати тип character (символьні рядочки). Щоб пояснити комп'ютеру, що йдеться про рівні деякого фактора, потрібно зробити перетворення типу:

```
> zf<-factor(z)
> zf

[1] USD EUR EUR UAH EUR USD UAH RUR
Levels: EUR RUR UAH USD
```

Тепер, хоча на екрані рівні фактора відображаються їх назвами, у внутрішньому представленні комп'ютера вони кодуються натуральними числами. Перелік різних рівнів виведено у рядочку Levels в порядку зростання кодів. Якщо вам потрібен тільки цей перелік у вигляді символьного рядочка, можна скористатись функцією levels()

```
> zl<-levels(zf)
> zl

[1] "EUR" "RUR" "UAH" "USD"
```

Відповідні коди можна побачити, використовуючи функцію unclass:

```
> unclass(zf)

[1] 4 1 1 3 1 4 3 2
attr(,"levels")
[1] "EUR" "RUR" "UAH" "USD"
```

Зрозуміло, що використання векторів з факторів замість символьних рядочків дозволяє економити місце у пам'яті комп'ютера, якщо довжина вектора велика, а кількість рівнів — помірна. Крім того, задання переліку рівнів дозволяє перевірити наявність зайвих назв, що могли б утворитись внаслідок якихось помилок. У статистиці є багато алгоритмів обробки даних, що працюють саме з категорійними даними (наприклад, у дисперсійному аналізі та у аналізі таблиць спряженості). З цим пов'язано виділення факторів у окремий тип.

Відмітимо, що у векторі факторів можуть зустрічатись не всі допустимі рівні, але вони будуть інформація про можливість їх появи зберігається у *атрибуті* levels:

```
> z2<-zf[c(1,2)]
> z2
```

```
[1] USD EUR
Levels: EUR RUR UAH USD
```

Якщо при виділенні підмножини вектора факторів потрібно вилучити рівні, що не зустрічаються у підмножині, це можна зробити, задавши *опцію* `drop`:

```
> z2d<-zf[c(1,2),drop=T]
> z2d
```

```
[1] USD EUR
Levels: EUR USD
```

У статистичних дослідженнях часто розбиття досліджуваних об'єктів на категорії проводится в залежності від того, у який діапазон потрапляє певна числова характеристика цих об'єктів. Наприклад, домогосподарства можна розділити на категорії з високим (high), середнім (mid) та низьким рівнем прибутку (low) в залежності від числового розміру їх прибутків. Для того, щоб робити це автоматично, застосовується функція `cut`:

```
> u<-c(6,5,4,3,2,1)
> ul<-cut(u,breaks=c(-Inf,2.5,3.5,Inf),labels=(c('low','mid','high')))
> ul
```

```
[1] high high high mid  low  low
Levels: low mid high
```

Тут ми створили вектор зі значеннями числової характеристики u і розбили досліджувані об'єкти на три категорії в залежності від значень u. Опція `breaks` вказує межі інтервалів, що визначають ці категорії: до першої потрапляють об'єкти, для яких $u \in (-\infty, 2.5]$, до другої — з $u \in (2.5, 3.5]$, до третьої — $u \in (3.5, \infty]$. Назви цих категорій (рівнів факторів) задані у опції `labels`.

У цьому випадку (а також у багатьох інших) для рівнів фактора можна вказати природний порядок: low<mid<high. Для деяких інших факторів (як от — для національності) такого порядку не існує. Щоб вказати комп'ютеру на наявність порядку рівнів, вводиться тип `ordered` (впорядкований фактор).

```
> ulo<-ordered(ul)
> ulo

[1] high high high mid  low  low
Levels: low < mid < high
```

Деякі функції R аналізують впорядковані фактори спеціальним чином, не так, як невпорядковані.

2.1.4 Матриці, масиви та фрейми даних.

Матриці в R обов'язково складаються з елементів одного типу (наприклад, тільки з чисел, або тільки з логічних значень). Є багато різних способів створити матрицю, наприклад, її можна скласти з окремих векторів-стовпчиків функцією `rbind` або з векторів-рядочків функцією `cbind`

```
> x1<-1:3
> x2<-5:7
> u<-rbind(x1,x2)
> u

   [,1] [,2] [,3]
x1    1    2    3
x2    5    6    7

> v<-cbind(x1,x2)
> v

     x1 x2
[1,]  1  5
[2,]  2  6
[3,]  3  7
```

(Зверніть увагу, що імена векторів перетворились на імена відповідних стовпчиків або рядочків).

Правила індексації зрозумілі з цього прикладу — перший індекс позначає рядочок, другий — стовпчик, тобто `u[2,3]` це елемент на перетині другого рядочка і третього стовпчика матриці `u`. Використання індексів та імен дуже гнучке, як показують наступні приклади:

```
> u[,1]

x1 x2
 1  5

> u[2,]

[1] 5 6 7

> v[,"x2"]

[1] 5 6 7

> v[1:2,"x2"]

[1] 5 6
```

"Вийнятий" з матриці стовпчик перетворюється на вектор-рядочок. Якщо ви хочете отримати як результат матрицю, що складається з одного стовпчика, скористайтесь опцією `drop=F`

```
> v[,"x2",drop=F]

     x2
[1,]  5
[2,]  6
[3,]  7
```

Інший спосіб створення матриці — функція `matrix`, яка перетворює вектор у матрицю. Першим параметром функції є вектор, який використовується для заповнення матриці, параметри `ncol` і `nrow` задають кількість стовпчиків і рядочків утвореної матриці.

Логіка роботи функції зрозуміла з наступних прикладів:

```
> x<-1:10
> matrix(x,nrow=2)

     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

```
> matrix(x,ncol=2)

     [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10

> matrix(x,ncol=2,nrow=2)

     [,1] [,2]
[1,]    1    3
[2,]    2    4

> matrix(x,ncol=3)

Warning in matrix(x, ncol = 3): длина данных [10] не является множителем количеств

     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7    1
[4,]    4    8    2
```

У останньому прикладі для того, щоб заповнити матрицю прийшлось циклічно повторити вектор `x`.

Імена рядочків та стовпчиків матриці можна задавати, використовуючи функцію `dimnames`, як показано у наступному прикладі:

```
> x<-1:10
> X<-matrix(x,nrow=2)
> dimnames(X)<-list(c('first','second'),letters[1:5])
> X

       a b c d  e
first  1 3 5 7  9
second 2 4 6 8 10
```

(Тут функція `list` створює список що складається з двох елементів, кожний з яких є вектором. `letters` у `R` це вектор, складений з латинських літер у алфавітному порядку.)

Для задання назв тільки рядочків (стовпчиків) можна використовувати функції `rownames` (`colnames`). Ті ж функції використовуються, якщо потрібно дізнатись імена для існуючої матриці:

```
> rownames(X)

[1] "first"  "second"
```

Часто буває корисною функція `diag`, яку можна застосовувати різними способами. Якщо її параметром є вектор, вона породжує діагональну матрицю:

```
> x<-1:3
> diag(x)

     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
```

Якщо параметр — матриця, `diag` виділяє її головну діагональ у вигляді вектора:

```
> X<-matrix(1:9,ncol=3)
> diag(X)

[1] 1 5 9
```

Нарешті, якщо `diag` зустрічається ліворуч від символа привласнення, вона замінює діагональ свого матричного параметра:

```
> diag(X)<-rep(0,3)
> X

     [,1] [,2] [,3]
[1,]    0    4    7
[2,]    2    0    8
[3,]    3    6    0
```

Арифметичні та логічні дії виконуються з матрицями поелементно. Для того, щоб виконати матричне множення, потрібно застосувати операцію `%*%`. Функція `t()` транспонує матрицю.

Функція `solve(A,b)` розв'язує рівняння $Ax = b$. Якщо викликати її без другого параметра, вона підраховує обернену матрицю: значенням `solve(A)` буде A^{-1}.

Обернену матрицю можна підрахувати, використовуючи функцію `ginv()`, яка не входить у ядро `R` , а міститься у пакеті (бібліотеці) `MASS`. Якщо ця бібліотека не була підключена раніше, її потрібно підключити перед використанням `ginv()`. (Точніше, `ginv()` обчислює псевдообернену матрицю Мура-Пенроуза, яка для невироджених матриць дорівнює звичайній оберненій).

```
> X<-matrix(1:6,ncol=2)
> X

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

> Y<-t(X)
> Z<-Y%*%X
> Z

     [,1] [,2]
[1,]   14   32
[2,]   32   77

> library(MASS)
> iZ<-ginv(Z) # iZ матриця обернена до Z
> Z%*%iZ      # матричне множення дає одиничну матрицю:

     [,1]         [,2]
[1,]    1 1.776357e-15
[2,]    0 1.000000e+00

> Z*Z         # тут множення поелементне:
```

```
     [,1] [,2]
[1,]  196 1024
[2,] 1024 5929
```

При підрахунках оберненої матриці `ginv()` буде більш стабільною ніж `solve()` — вона дає точніші результати коли визначник матриці близький до 0. Це добре, якщо ви використовуєте функції правильно. Але якщо ви помилитеся і параметр функції буде виродженою або не квадратною матрицею, то `solve()` повідомить вас про помилку, а `ginv()` — ні, тому що узагальнена обернена визначена і для таких матриць.

Фрейми даних відрізняються від матриць у першу чергу тим, що в них стовпчики можуть мати різні типи. Такий формат особливо зручний для запису типових статистичнх даних у вигляді таблиці, в якій кожному спостережуваному об'єкту відповідає один рядочок, а змінні, що характеризують об'єкти, записуються у відповідні стовпчики. При цьому кожна змінна може бути свого типу — числового, логічного, символьного чи факторного.

Наприклад, у наборі `iris` міститься набір даних про квіти півники (іриси). Кожен рядочок цих даних відповідає одній квітці. Для кожної дослідженої квітки у відповідному стовпчику записано характеристики `Sepal.Length`, `Sepal.Width` (довжина та ширина чашолистків), `Petal.Length`, `Petal.Width` (довжина та ширина пелюсток) а також `Species` — вид роду Iris, до якого належить дана квітка.

Наступний приклад показує, як можна вивести на екран значення, що містяться у 45–55 рядочках цього набору даних. (Набір `iris` входить у колекцію даних `Datasets`, що оформлена як один з пакетів для `R` . Як правило, цей пакет завантажується системою автоматично, якщо це не так, його можна завантажити командою library(Datasets)).

```
> print(iris[45:55,])
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
45          5.1         3.8          1.9         0.4     setosa
46          4.8         3.0          1.4         0.3     setosa
47          5.1         3.8          1.6         0.2     setosa
48          4.6         3.2          1.4         0.2     setosa
49          5.3         3.7          1.5         0.2     setosa
50          5.0         3.3          1.4         0.2     setosa
51          7.0         3.2          4.7         1.4 versicolor
```

```
52          6.4         3.2          4.5         1.5 versicolor
53          6.9         3.1          4.9         1.5 versicolor
54          5.5         2.3          4.0         1.3 versicolor
55          6.5         2.8          4.6         1.5 versicolor
```

Створити фрейм даних з окремих векторів-стовпчиків змінних можна використовуючи функцію `data.frame()`:

```
> numb<-1:5
> let<-letters[numb]
> Name<-c('Alfa','Bravo','Charlie','Delta','Echo')
> type<-factor(c('vowel','consonant','consonant','consonant','vowel'))
> L<-data.frame(numb,let,type,row.names=Name, stringsAsFactors = FALSE)
> print(L)

        numb let      type
Alfa       1   a     vowel
Bravo      2   b consonant
Charlie    3   c consonant
Delta      4   d consonant
Echo       5   e     vowel
```

Параметр `row.names` вказує імена об'єктів (рядочків таблиці). Іменами змінних стають імена векторів стовпчиків, з яких склали фрейм. При необхідності імена рядочків і стовпчиків можна продивитись і змінити функціями `row.names()` і `names()`.

Параметр `stringsAsFactors` показує, чи слід при створенні фрейму перетворювати вектори символьних рядочків у змінні типу фактор. За умовчанням таке перетворення виконується, тому, якщо вам потрібні саме символьні змінні, вказуйте `stringsAsFactors = FALSE`.

Для того, щоб продивлятись, а при необхідності — і виправляти великі фрейми даних, можна використовувати вбудований редактор `R` , який викликається функцією `edit()`.

З даними, що складають фрейм, можна працювати як з елементами матриці, наприклад:

```
> L[2,]

      numb let      type
Bravo    2   b consonant
```

```
> L[,2]

[1] "a" "b" "c" "d" "e"

> L[,'let']

[1] "a" "b" "c" "d" "e"
```

Крім того, змінні є атрибутами фрейму, тому до них можна звертатись, використовуючи формат

ім'я об'єкта$*ім'я атрибуту*

наприклад, задавши команду

```
> L$let

[1] "a" "b" "c" "d" "e"
```

отримуємо вектор значень змінної `let` для об'єкта (фрейма) `L`

При роботі з фреймами інколи виникає потреба перевірити, який тип у тієї чи іншої змінної. Це можна зробити, використовуючи функції перевірки типів `is.numeric`, `is.logical`, `is.integer` та подібні їм. Ці функції видають логічне значення T якщо їх параметр має відповідний тип і F — якщо тип не той. Наприклад:

```
> is.numeric(L$num)

[1] TRUE

> is.factor(L['let'])

[1] FALSE

> is.character(L$let)

[1] TRUE
```

Зміну типу можна робити, використовуючи відповідно функції `as.numeric`, `as.character` ті ін. Наприклад:

```
> x<-c('12','3')
> x[1]+x[2]
```

```
Error in x[1] + x[2]: нечисловой аргумент для бинарного оператора
```

```
> y<-as.numeric(x)
> y[1]+y[2]
```

```
[1] 15
```

Варто мати на увазі, що R виконує автоматичне перетворення типів у елементарних операціях, наприклад,

```
> 1+TRUE
```

```
[1] 2
```

```
> TRUE&(-0.5)
```

```
[1] TRUE
```

(TRUE трактується як 1, FALSE як 0 у арифметичних операціях. Ненульові числа трактуються як TRUE, а 0 — як FALSE у логічних операціях).

2.1.5 Векторні і матричні функції. Функція apply. Пропущені значення.

Як вже відмічалось, елементарні операції та функції виконуються над масивами поелементно. Цю властивість мають не всі функції. Якщо потрібно явно вказати, що деяка функція повинна застосовуватись до кожного елемента вектора, застосовують функцію `sapply` або `lapply()`. Першим параметром `sapply` має бути вектор `x`, до якого застосовується функція, другим — функція `FUN`, яка застосовується до кожного елемента `x`. Точніше, елементи `x` підставляються у `FUN` замість першого її параметра. Якщо у `FUN` є іще параметри, їх можна вказати як додаткові параметри-опції `sapply`, і вони будуть передані у `FUN` за їх назвами. Наприклад, тут ми застосовуємо двійковий логарифм до вектора ступенів двійки:

```
> sapply(c(1,2,4,8),log,base=2)
```

```
[1] 0 1 2 3
```

(У цьому прикладі застосування `sapply()` не обов'язкове, такий самий ефект буде при виклику `log(c(1,2,4,8),base=2)`).

Часто зустрічаються функції, які працюють з вектором "в цілому". Наприклад,

`length()` — функція, що повертає кількість елементів масиву;

`sum()`, `prod()` — функції, що підраховують суму або добуток всіх елементів масиву;

`sort` — функція, що відсортовує елементи масиву у порядку зростання (або спадання, якщо вказана опція `decreasing=T`).

та інші. При застосуванні такої функції до матриці, вони трактують її як "довгий" вектор, складений з усіх елементів цієї матриці.

```
> u1<-c(3,1,2)
> u2<-c(0,-1,-2)
> z<-cbind(u1,u2)
> z

     u1 u2
[1,]  3  0
[2,]  1 -1
[3,]  2 -2

> sum(z)

[1] 3

> sort(z)

[1] -2 -1  0  1  2  3

> sort(z,decreasing=T)

[1]  3  2  1  0 -1 -2
```

Часто буває потрібно застосувати функцію від векторного елемента до кожного рядочка, або до кожного стовпчика матриці окремо. У такому випадку використовується функція `apply()`, що має специфікацію

```
apply(X, MARGIN, FUN, ...)
```

де `X` — масив, до якого буде застосовуватись функція; `MARGIN=1` якщо функція застосовується до рядочків і `MARGIN=2` — якщо до стовпчиків; `FUN` — ім'я функції, яку потрібно застосувати.

`...` позначає, що у виклику функції `apply()` можна також задавати будь-які інші опції. Ці опції `apply()` передасть у функцію `FUN` без змін.

Першим параметром функції `FUN` повинен бути вектор, замість цього вектора `apply()` підставляє послідовно рядочки (або стовпчики) матриці `X` і результат також записує у список результатів. Наприклад, використовуючи матрицю `z` з попереднього прикладу, отримуємо

```
> apply(z,1,sum)

[1] 3 0 0

> apply(z,2,sum)

u1 u2
 6 -3

> apply(z,2,sort)

     u1 u2
[1,]  1 -2
[2,]  2 -1
[3,]  3  0
```

У третьому прикладі елементи матриці `z` відсортувались окремо всередині кожного стовпчика. А от для того, щоб отримати матрицю з елементами, відсортованими всередині кожного рядочка, результат роботи `apply` потрібно транспонувати:

```
> apply(z,1,sort)

   [,1] [,2] [,3]
u2    0   -1   -2
u1    3    1    2

> t(apply(z,1,sort))
```

```
     u2 u1
[1,]  0  3
[2,] -1  1
[3,] -2  2
```

Приклад передачі опції при виклику `sort()` через `apply()`:

```
> apply(z,2,sort,decreasing=T)
```

```
     u1 u2
[1,]  3  0
[2,]  2 -1
[3,]  1 -2
```

Іще один варіант поелементної обробки векторів реалізує функція `outer()`. Перші два її параметри `x, y` є векторами, третій `FUN` — функцією, у якої не менше двох параметрів. Замість цих параметрів `outer()` підставляє послідовно всі можливі пари елементів `x` і `y` (x замість першого параметра, y — замість другого). Отримані значення утворюють матрицю. Наприклад,

```
> x<-1:4
> y<-5:7
> f<-function(x,y){x^2+y^2}
> z<-outer(x,y,f)
> z
```

```
     [,1] [,2] [,3]
[1,]   26   37   50
[2,]   29   40   53
[3,]   34   45   58
[4,]   41   52   65
```

Тут ми створили нову функцію `f`, яка обчислює суму квадратів двох своїх аргументів[1] і застосували її до всіх пар елементів векторів `x` і `y`. Як і функція `apply()`, `outer()` вміє передавати додаткові параметри всередину функції `FUN`.

[1]Про створення власних функцій див. далі, у п. 2.3.1.

Такі функції як `sum()` та `prod()` можуть обробляти пропущені значення (NA) по різному. Можна вважати, що якщо якесь значення у векторі невідоме, то і сума невідома. А можна вилучити всі пропущені значення і підрахувати суму не пропущених. Вибір реалізується за допомогою опції `na.rm` ("NA remove" — видалення пропущених):

```
> x<-c(2,NA,1,4,3)
> sum(x)

[1] NA

> sum(x,na.rm=T)

[1] 10
```

При застосуванні функції `sort()` значення NA видаляються за умовчанням. Можна скористатись опцією `na.last=T` щоб при сортуванні значення NA потрапляли у кінець вектора. При цьому значення NA та NaN обробляються однаково:

```
> x<-c(2,NA,1,NaN,4,NA,3,NaN)
> sort(x,na.last=T)

[1]   1   2   3   4  NA NaN  NA NaN
```

2.2 Експорт та імпорт даних у R

2.2.1 Експорт та імпорт даних у внутрішньому форматі

Інколи буває потрібно зберігти деякі результати роботи програми у форматі R , наприклад, для використання їх іншим користувачем R у своїй програмі. Для цього можна скористатись функцією `save()`:

```
> a<-1:10
> save(a,file="c:/rem/term/example.Rdata")
```

У цьому прикладі ми створили вектор `a`, а потім записали його у файлі `example.Rdata` у каталозі `term` на диску `c`. (Зверніть увагу, що при записі шляху до файлу використовується символ `/` прийнятий в Unix, а не \, як це прийнято у Windows).

`save()` зберігає об'єкти у внутрішньому кодуванні системи `R` . Прочитати записаний файл можна лише у `R` . Якщо продивлятись його у якому-небудь текстовому редакторі, будуть відображатись лише незрозумілі символи. Для читання можна використати функцію `load()`.

```
> a<-0
> a

[1] 0

> load(file="c:/rem/term/example.Rdata")
> a

 [1]  1  2  3  4  5  6  7  8  9 10
```

(Ми спочатку надали нове значення `a`, а потім відновили старе, прочитавши його з файлу). Об'єкти записуються разом із своїми іменами, тому `load` розуміє без додаткових пояснень, що саме потрібно змінити.

Якщо в одному файлі потрібно зберігти багато об'єктів, їх перелічують через кому у списку параметрів `save()`.

Якщо файл для запису або читання потрібно вибрати під час роботи програми інтерактивно, використовують функцією `file.choose()`, яка відкриває стандартне вікно вибору файлу. Скажімо, для завантаження з файлу, який ви хочете обрати вручну, можна написати:

`load(file=file.choose())`.

2.2.2 Експорт та імпорт текстових таблиць з даними.

Практично кожна статистична програма загального призначення має можливості створення та читання файлів даних у вигляді текстових таблиць. Зміст таких файлів легко зрозуміти, проглядаючи їх у звичайних текстових редакторах. Тому природно використовувати такі файли для обміну статистичними даними між програмами.

Нехай таблиця записана у текстовому файлі у зручному для людського сприйняття вигляді:

```
  Name       Weight Married
Ahmad        70     T
John         82     F
Victoria     60     T
Olga         54     F
```

Тут у першому рядочку записані назви змінних, а у кожному наступному рядочку — значення цих змінних для певної людини. Для читання таких таблиць використовують функцію `read.table()`. Якщо таблиця записана у файлі `c:/rem/term/table.txt`, то прочитати її можна так:

```
> tbl<-read.table(file="c:/rem/term/table.txt",header=T)
> tbl

      Name Weight Married
1    Ahmad     70    TRUE
2     John     82   FALSE
3 Victoria     60    TRUE
4     Olga     54   FALSE
```

(Результат читання записано у фрейм `tbl`. Опція `header=T` вказує на те, що у першому рядочку містяться назви змінних).

Якщо один з стовпчиків таблиці треба прочитати як імена об'єктів-рядочків, це можна зробити задавши опцію `row.names`. У ній можна вказати або номер стовпчика імен, або його назву, наприклад:

```
>  tbl<-read.table(file="c:/rem/term/table.txt",header=T,row.names="Name")
>  tbl

         Weight Married
Ahmad        70    TRUE
John         82   FALSE
Victoria     60    TRUE
Olga         54   FALSE
```

Записати таку таблицю можна, використовуючи функцію `write.table()`:
`write.table(tbl,file="c:/rem/term/table.txt")`

За умовчанням, у першому стовпчику таблиці будуть записані імена об'єктів-рядочків фрейму, а у першому рядочку — назви змінних-стовпчиків фрейму. Якщо це непотрібно, слід вказати опції `row.names=F`, `col.names=F`.

При читанні з файлу `read.table` визначає кількість змінних (стовпчиків) у таблиці за кількістю назв у першому рядочку. Тип змінної визначається за форматом запису елементів у відповідному стовпчику. Скажімо, якщо всі елементи стовпчика мають вигляд `TRUE`, `FALSE` або `NA`, то відповідна змінна отримає у прочитаному фреймі даних тип `logical`. Якщо хоча б один елемент не можна трактувати як логічний — тип буде `character` навіть, якщо всі інші елементи виглядають як логічні.

Різна кількість елементів, розділених пробілами у різних рядочках таблиці приводить до помилки читання. Якщо у файлі зустрічаються символьні рядочки з пробілами всередині, ці рядочки треба вміщувати у лапки, як у наступному прикладі:

```
Name        Weight Married
Ahmad         70     T
"John R.C."   82     F
Victoria      60     T
"Olga V."     54     F
```

Інший текстовий формат — csv (comma separated values) в якому окремі значення змінних розділяються комами. Цей формат менш зручний для людського сприйняття, ніж табличний, але він дає більше можливостей для передачі даних різних типів.

Щоб записати (або прочитати) файл у форматі csv можна використовувати функції `write.csv` (`read.csv`). В основному, вони влаштовані аналогічно `write.table` та `read.table`. (По суті, відмінність між функціями, що обробляють table та csv полягає лише в іншому виборі значень за умовчанням тих опцій, які регулюють вибір символів, що розділяють значення. Вибирати ці опції (вони описані у help) можна самому, якщо потрібно створити або прочитати файл з нестандартного формату.

2.3 Програмування у R

2.3.1 Створення власних функцій

SecOunFun

Функції у R є об'єктами, тому для того, щоб ввести нову функцію, треба створити об'єкт типу `function` і привласнити його значення деякій змінній. Наприклад, тут ми у першому рядочку створюємо функцію `t.sum()`, а у наступному — викликаємо її з параметрами `x=1:10` та `t=8`:

```
> t.sum<-function(x,t){sum(x[x>t])}
> t.sum(1:10,8)

[1] 19
```

Призначення цієї функції зрозуміле — вона підраховує суму тих елементів вектора-параметра `x`, які перевищують поріг заданий параметром `t`. В загальному вигляді команда створення нової функції (специфікація) має формат

`function(`*список формальних параметрів* `){`*тіло функції*`}`

Тіло функції — це послідовність команд, які будуть виконані при виклику функції. Результат останньої виконаної у тілі функції команди є *значенням* функції. Це значення і буде результатом виразу виклику функції.

При виклику функції фактичні значення параметрів, задані у дужках після імені функції, підставляються замість формальних параметрів, вказаних у специфікації функції. Можна використовувати неіменований спосіб підстановки, коли формальні параметри заміняються фактичними в порядку їх переліку у специфікації. Так зроблено у попередньому прикладі. Можна застосувати іменовану підстановку, вказуючи ім'я формального параметра, який потрібно замінити при виклику:

```
> t.sum(t=9,x=1:10)

[1] 10
```

Виконання такого виклику нічим не відрізняється від попереднього. Можна комбінувати ці два способи:

```
> t.sum(1:10,t=9)
```

```
[1] 10
```

R дозволяє задавати при виклику функції менше параметрів, ніж вказано у специфікації. При цьому функція мусить знати, яких значень ці параметри набувають за умовчанням (коли вони не вказані). У специфікації такі значення за умовчанням можна вказати використовуючи знак = після імені формального параметра:

```
> t.sum<-function(x,t=0){sum(x[x>t])}
> t.sum(-5:5)

[1] 15

> t.sum(1:10,t=9)

[1] 10
```

Тут за умовчанням функція `t.sum` підраховує суму додатних елементів вектора, але поріг можна змінити, задавши значення параметра `t` явно. Параметри, що, як правило, використовуються за умовчанням, прийнято називати опціями функції. З точки зору комп'ютера опції нічим не відрізняються від інших параметрів.

При заданні значень за умовчанням можна використовувати вирази, в які входять інші формальні параметри функції:

```
> t.sum<-function(x,t=sum(x)/length(x)){sum(x[x>t])}
> t.sum(1:10)

[1] 40
```

— за умовчанням, функція підраховує суму всіх елементів `x`, які перевищують середнє `x`.

У тілі функції може бути багато команд (їх можна записувати у окремих рядочках програми або розділяти крапкою з комою). Можна також використовувати змінні, що не входять до списку формальних параметрів. Так функція `t.sum()` з попередніх прикладів може бути реалізована наступним чином:

```
> t.sum<-function(x,t=0)
+        {
+        z<-x>t
+        sum(z)
+        }
```

Тут `z` — допоміжна змінна, що використовується у функції. За правилами `R` всі такі змінні є *локальними* тобто вони існують лише всередині функції і знищуються при завершенні виконання її виклику. Якщо поза тілом функції було введено змінну з тим самим іменем (*глобальну змінну*) — її значення не зміниться після виклику функції:

```
> z<-0
> t.sum(1:10,t=8)

[1] 2

> z

[1] 0
```

(Значення глобального `z` залишилось 0 не зважаючи на виклик функції, в якій локальній змінній `z` було зроблене привласнення).

Такий підхід дозволяє усунути можливість небажаних побічних ефектів (side effect) коли функція змінює значення змінних, що не мають відношення до її виклику. Інколи буває потрібно, щоб функція мала побічний ефект, впливаючи на певну глобальну змінну. Для цього можна використати *глобальне привласнення* `<<-`. Нехай, наприклад, потрібно підрахувати, скільки разів відбувався виклик функції `t.sum()` у програмі. Для цього можна завести глобальну змінну `n` і модифікувати функцію так:

```
> n<-0
> t.sum<-function(x,t=0){n<<-n+1;sum(x>t)}
> t.sum(1:10)

[1] 10

> n
```

```
[1] 1

> t.sum(1:10)

[1] 10

> n

[1] 2
```

Глобальні привласнення рекомендується використовувати дуже обережно, оскільки вони можуть зробити логіку виконання функції незрозумілою.

Насправді, при виклику функції, з усіх фактичних параметрів робляться копії і саме ці копії підставляються у функцію замість формальних параметрів. Тому навіть якщо в тілі функції деякому формальному параметру привласнюється нове значення, це не вплине на відповідний фактичний параметр у зовнішній програмі:

```
> my.sort<-function(x){x<-sort(x)}
> z<-c(3,5,1)
> y<-my.sort(z)
> y

[1] 1 3 5

> z

[1] 3 5 1
```

(Змінна `z` не відсортувалась, хоча вона використана у виклику функції, яка сортує свій формальний параметр у тілі). Цей спосіб передачі параметрів *за значенням* а не *за назвою* також запобігає небажаним побічним ефектам. Всю інформацію, яку потрібно передати про роботу функції слід вкладати у її значення.

У списку формальних параметрів функції можна використовувати символ `...` — трикрапка. Він позначає, що функцію можна викликати з довільною кількістю параметрів. Параметри, що стоять на місці `...` можна використовувати у тілі функції так само, як інші, єдина їх відмінність полягає в тому, що вони не мають індивідуальних імен. У

наступному прикладі створюється функція, що має один іменований параметр `x` та може викликатись із довільною кількістю інших параметрів. Параметр `x` не використовується, а всі параметри, що підставляються у виклиці на місці `...` збираються в один список, який і є результатом виконання функції.

```
> f<-function(x,...){list(...)}
> z<-f(2,1:5,"aaa",T)
> z

[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "aaa"

[[3]]
[1] TRUE

> z[1]

[[1]]
[1] 1 2 3 4 5
```

2.3.2 Структури управління виконанням програм у мові R

У R порівняно небагато мовних структур, які забезпечують зміну порядку виконання команд у програмі. Мова розроблена так, щоб мінімізувати потребу їх використання. Наприклад, там, де у інших мовах програміст змушений використовувати цикл `for`, у R часто можна скористатись векторними виразами. Можливість використання логічних масивів при індексації та параметрів-опцій зі значеннями по умовчанню помітно звужує область застосування структур умовних переходів типу `if...else`. Хороший стиль програмування у R полягає в тому, щоб не використовувати подібні структури там, де можна обійтись іншими.

Тим не менше, у деяких випадках саме використання цих структур робить програму ефективною, а код — зрозумілим. Опишемо ці структури послідовно.

Умовне виконання: if

У R є три варіанти структур, що реалізують класичний умовний перехід:

— `if(`*умова*`)`*команда* — якщо *умова* істина, *команда* виконується, інакше — не виконується (тут і далі *команда* може бути складною, тобто складатись із послідовності команд, об'єднаних фігурними дужками)

— `if(`*умова*`)` *команда1* `else` *команда2* — тут *команда1* виконується, якщо *умова* істина, *команда2* виконується, якщо *умова* — хибна.

— `ifelse(` *умова*,*команда1*,*команда2*`)` — логіка виконання така ж, як і у попередньому варіанті.

Наприклад:

```
> x<-1
> y<-2
> if(x<y) x else y

[1] 1
```

(результатом виконання `if` тут буде менше з чисел x та y).

У третьому варіанті `ifelse()` працює як функція, зокрема, при застосуванні до векторів, вона дає векторні значення:

```
> x<-c(-4,4)
> sqrt(x)

Warning in sqrt(x): созданы NaN

[1] NaN   2

> sqrt(ifelse(x>0,x,NA))

[1] NA  2
```

В умовах `if` та інших структур управління можна використовувати логічні операції `&` (логічне і) та `|` (логічне або). При визначенні результату цих операцій спочатку обчислюється значення виразів праворуч та ліворуч від знаку операції а потім виконується сама операція. Інколи результат операції можна визначити лише за значенням, що стоїть ліворуч, наприклад значення `T|x` завжди `T`, яким би ні був `x`. Якщо у таких ситуаціях вам не потрібно обчислювати вираз праворуч від знаку операції, можна скористатись операціями `&&` та `||`:

```
> T|(sqrt(-5)>0)

Warning in sqrt(-5): созданы NaN

[1] TRUE

> T||(sqrt(-5)>0)

[1] TRUE
```

(у другому варіанті не було спроби обчислити `sqrt(-5)`).

2.3.3 Вибір з кількох умов: switch

Функція `switch()` дозволяє обирати один з багатьох варіантів виконання програми в залежності від значення деякого виразу. Її формат

`switch(`*вираз-умова*`,`*список варіантів*`)`

Як приклад, розглянемо застосування `switch()` у функції `f()`, що обчислює сумму або добуток елементів вектора в залежності від значення параметра `type`:

```
> f <- function(x, type)
+ {
+ switch(type, add = sum(x), multiply = prod(x), NA)
+ }
> f(1:4,type="add")

[1] 10

> f(1:4,type="multiply")

[1] 24

> f(1:4,type="x")

[1] NA
```

У цьому прикладі `switch()` обчислює значення виразу `type`, знаходить далі у списку параметрів такий параметр, назва якого відповідає `type` і обчислює вираз, що стоїть після знаку `=` для цього параметру (тобто значення цього параметру по умовчанню). Результат обчислення є значенням, яке дає `switch()`.

Останній елемент у списку параметрів `switch` у цьому прикладі (`NA`), записаний без знаку =, задає дії, котрі будуть виконані, якщо значення виразу-умови не дорівнює ні одному з попередніх варіантів. Якщо такого останнього елемента немає, жодні дії в цьому випадку не виконуються а значення `switch` дорівнює `NULL`.

2.3.4 Цикли while та repeat

`R` , як і інші мови програмування, використовує цикли для організації серій повторних обчислень. Загальний формат циклу `while`:

`while(`*умова*`)`*команда*

Спочатку перевіряється *умова*, і якщо вона дає результат `TRUE`, виконується *команда*. Цей процес повторюється циклічно і зупиняється як тільки *умова* прийме значення `FALSE`

У наступному прикладі цей цикл використано для наближеного обчислення кореня рівняння $x = \cos(x)$ (`eps` — точність обчислень):

```
> x<-1
> eps<-0.0000001
> while(abs(x-cos(x))>eps)x<-cos(x)
> x

[1] 0.7390851

> cos(x)

[1] 0.7390852
```

Інколи буває зручно розмістити перевірку умови не на початку циклу, а в кінці, або навіть посередині. Для організації таких циклів зручно використовувати структуру `repeat` з командами `break` та `next`.

Формат команди

`repeat` *команда*

де *команда* — це "тіло циклу", тобто послідовність команд, які повинні виконуватись циклічно. Щоб комп'тер міг зупинитись, всередині тіла циклу повинна бути команда `break`, яка перериває виконання циклу і передає управління на команду, що йде одразу після тіла циклу.

```
> x<-NULL
> t<-100
> i<-0
> repeat
+ {
+ i<-i+1
+ if(i^2>t) break
+ x<-c(x,i^2)
+ }
> x

 [1]   1   4   9  16  25  36  49  64  81 100
```

(квадрати натуральних чисел додаються до списку доти, доки вони не перевищують поріг `t`).

Команда `next` всередині тіла циклу перериває виконання даного циклу і передає управління на першу команду тіла.

2.3.5 Цикл for

Цикл for використовується тоді, коли одну і ту ж дію потрібно виконати для певної послідовності (вектора) індексів. Формат відповідної структури

`for(`*індекс* in *послідовність*`)` *команда*

тут *індекс* — назва змінної, що використовується для індексації у тілі циклу; *послідовність* — вектор значень, що будуть підставлені замість індексу при виконанні циклу; *команда* — тіло циклу, тобто команда, або набір команд, вміщених у фігурні дужки, який буде виконано для всіх значень індексу.

Наприклад, якщо для вектора $x = (x_1, \ldots, x_n)$ потрібно підрахувати вектор різниць $y = (x_2 - x_1, \ldots, x_n - x_{n-1})$, це можна зробити, використовуючи цикл:

```
> x<-(1:10)^2
> y<-rep(NA,length(x)-1)
> for(i in 1:(length(x)-1))y[i]<-x[i+1]-x[i]
> x

 [1]   1   4   9  16  25  36  49  64  81 100

> y

[1]  3  5  7  9 11 13 15 17 19
```

Потреба використання циклів for у R значно менша, ніж у більшості класичних мов програмування, завдяки можливостям застосування векторних функцій та гнучкій індексації масивів. Так, у попередньому прикладі, вектор різниць можна підрахувати як

```
> x[-1]-x[-length(x)]

[1]  3  5  7  9 11 13 15 17 19
```

(Нагадаємо, що від'ємний індекс наказує вилучити відповідний елемент з вектора: `x[-1]` — вектор з усіх елементів `x` крім першого).

Хороший стиль програмування у R вимагає не використовувати цикли for якщо без них можна обійтись.

Розділ 3

Базова графіка в R

3.1 Стовпцеві та кругові діаграми

Один з найбільш популярних способів відображення не дуже великих наборів чисел — діаграми, на яких кожному числу відповідає один стовпчик. Англійською мовою такі рисунки звуть barplot або barchart. Для їх відображення можна використовувати функцію `barplot()`. Наприклад, у наборі даних `ldeaths` вміщені помісячні дані про кількості смертей у Великій Британії від бронхітів, астми та емфіземи легенів. Перші 12 елементів набору відносяться до 1974 року. Зобразимо їх на стовпцевій діаграмі:

```
> barplot(ldeaths[1:12],names.arg =month.abb,main="(a)")
> barplot(ldeaths[1:12],names.arg =1:12,horiz=T,col=2:4,main="(b)")
```

Результати виконання відображені на рис. 3.1 (a) — перший виклик `barplot()`, (b) — другий.

Перший параметр функції `barplot()`, `height` задає висоти стовпчиків, якщо стовпчики вертикальні, як на рисунку (a) або довжини — коли стовпчики горизонтальні, як на (b). Вибір орієнтації стовпчиків задає параметр `horiz` (T — горизонтальні, F — вертикальні). Параметр `col` задає колір стовпчика, `names.arg` — назви, які будуть підписані під стовпчиками. (У першому прикладі ці назви взяті з масиву `month.abb`, який містить скорочені імена місяців).

Параметр `main` задає заголовок, що виводиться над рисунком. Можна також задати текст підпису під рисунком — параметром `sub`.

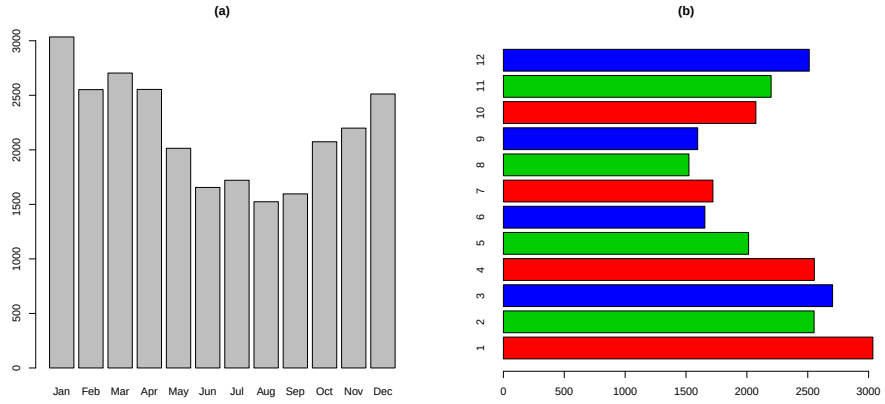

Рис. 3.1: Стовпцева діаграма загальної смертності

Параметр `height` можна задати як матрицю. Це дає можливість порівнювати різні набори даних на одній діаграмі. Наприклад, у наборах `mdeaths` і `fdeaths` знаходяться дані про смертність окремо чоловіків та жінок. Зберемо ці дані в одну матрицю і виведемо:

```
> h<-rbind(fdeaths[1:12],mdeaths[1:12])
> rownames(h)<-c("female","male")
> colnames(h)<-month.abb
> barplot(h,main="(a)",
+         legend.text=T,args.legend = list(x = "top"),col=c(2,3))
> barplot(h,main="(b)",legend.text=T,
+       args.legend = list(x = "topright",inset=0.2),col=c(2,3),beside=T)
```

Результати виконання відображені на рис. 3.2 (a) — перший виклик `barplot()`, (b) — другий.

Тут перший рядочок матриці відповідає жіночій смертності по місяцях, другий - чоловічій. Ми надали цим рядочкам імена `female` і `male`, а стовпчики матриці даних назвали скороченими іменами місяців. У варіанті (a) стовпчики на діаграмі, що відповідають чоловікам виведені як продовження стовпчиків для жінок. Це зручно тим, що одразу можна порівнювати сумарні смертності чоловіків та жінок протягом різних місяців. У варіанті (b) стовпчики чоловіків та жінок виводяться поруч, так зручніше порівнювати їх між собою. Вибір з цих варіантів відображення робить параметр `beside`.

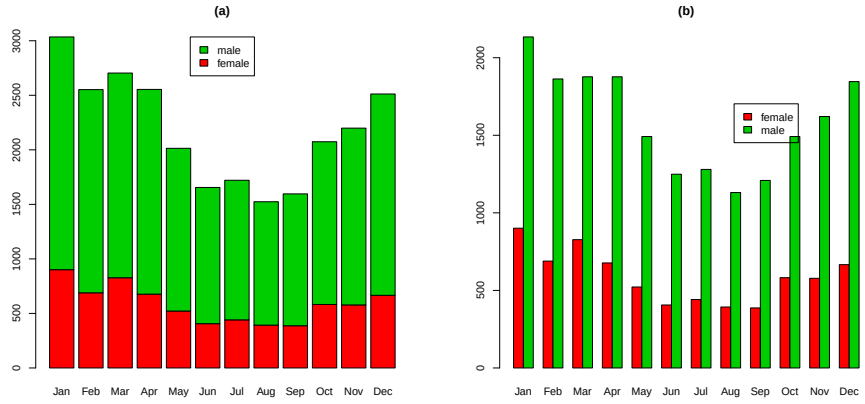

Рис. 3.2: Стовпцева діаграма смертності чоловіків та жінок

Для того, щоб читач легше міг зрозуміти, де на діаграмі виводиться який рядок матриці, можна відобразити пояснення (легенду). Текст пояснення вказується у параметрі `legend.text`. Якщо просто задати `legend.text=T`, то для пояснення будуть використані назви рядочків матриці `height`, як це зроблено у нашому випадку. Для рисунку (a) ми задали положення легенди на рисунку задавши конструкцію з опцій `args.legend = list(x = "top")` яка вказує, що легенда має бути вгорі. Можливі варіанти `"bottomright"`, `"bottom"`, "topleft `"center"` та аналогічні. Для рисунку (b) задано також відступ від правого поля рисунку середини легенди опцією `inset`.

У функції `barplot()` є багато інших опцій, зокрема параметри `density`, `angle`, `border` регулюють штриховку стовпчиків та рисування контуру аналогічно тому, як це робиться у функції рисування прямокутників `rect()`. Опції, що керують рисуванням осей координат: `axes`, `xlab`, `ylab` та розміром рисунку `xlim`, `ylim` аналогічні таким опціям функції `plot()` (див. підрозділ 3.2).

Логічна опція `add` вказує, чи треба відкривати нове вікно для рисування діаграми (`add=F`) чи діаграма відображається у вже відкритому вікні доповнюючи існуючий рисунок.

Інколи для унаочнення даних використовують не стовпчикові, а кругові діаграми. Ідея полягає в тому, щоб зобразити “частки спільного пирога”, які дістались різним їдокам. Відповідно англійська назва таких

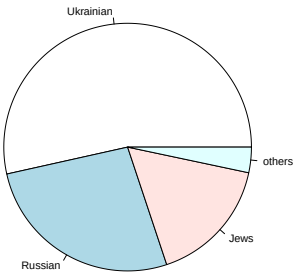

Рис. 3.3: Національний склад населення Києва у 1939р.

діаграм — pie charts. Наприклад, за даними перепису 1939 року у Києві проживало

українців — 450 556,
євреїв — 224 236,
росіян — 139 495,
людей інших національностей — 27991.

Кругову діаграму для цих даних можна зобразити так:

```
> population<-c(450556,224236,139495,27991)
> names(population)<-c("Ukrainian", "Russian","Jews","others")
> pie(population)
```

Результат — на рис. 3.3.

Кругові діаграми вважаються менш візуально сприйнятними ніж стовпчикові, тому у серйозних дослідженнях як окремий засіб графічного відображення застосовуються не часто. Як правило, стовпчикові діаграми використовувати доцільніше. Але кругові діаграми завдяки своїй компактності можуть бути зручними для порівняння великої кількості наборів даних, наприклад, при відображенні на географічній карті складу населення різних міст країни, тощо.

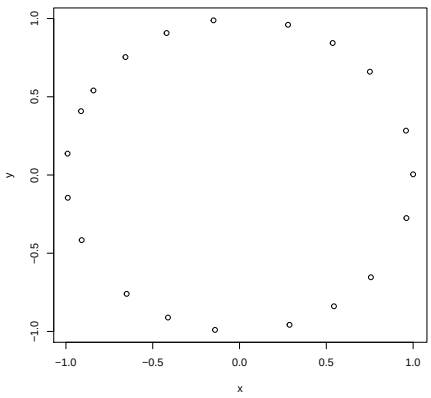

Рис. 3.4: Набір точок виведений функцією plot()

3.2 Точки та лінії на площині

У R для графічного відображення об'єктів часто використовується функція `plot()`. Ця функція є родовою, тобто вона працює по різному для різних об'єктів. Зараз ми розглянемо простіше застосування `plot()` у випадку, коли потрібно відобразити які-небудь точки або лінії на площині. У такому випадку координати точок можна задати у вигляді векторних параметрів функції:

```
> x<--sin(1:20)
> y<-cos(1:20)
> plot(x,y)
```

Результат виконання цієї функції — зображення на координатній площині набору точок, у яких горизонтальні координати взяті з вектора `x`, а вертикальні — з `y` — див. рис. 3.4. Зрозуміло, що для нормальної роботи функції ці вектори повинні мати однакову довжину.

Функція `plot()` спочатку створює нове вікно виводу, а потім виводить у нього об'єкти (у нашому випадку — точки). Якщо потрібно додати нові об'єкти на старому рисунку, замість `plot()` слід використати іншу функцію, наприклад — `points(x,y,...)` або `lines(x,y,...)`.

```
>  x<--4:4
>  y<-x^2
```

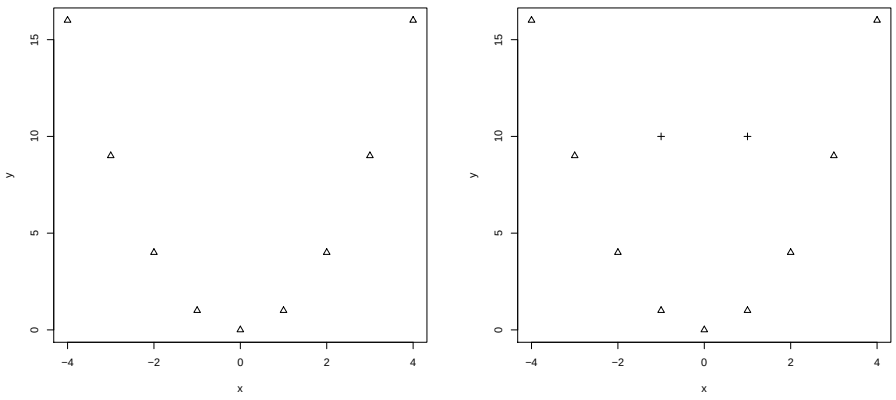

Рис. 3.5: Додали точки, використовуючи points()

```
> plot(x,y,pch=2)
> x1<-c(-1,1)
> y1<-c(10,10)
> points(x1,y1,pch=3)
```

На рис. 3.5 ліворуч — результат, що буде відображений після виконання `plot(x,y,pch=2)`, праворуч — як зміниться рисунок після додавання точок `points(x1,y1,pch=3)`.

Опція `pch` задає символ, яким будуть відображатись точки на рисунку.

Якщо до рисунку потрібно додати лінії, можна скористатись функцією `lines(x,y,...)`.

Функція `plot()` має багато опцій, які дозволяють обирати формат відображення рисунку. Перелічимо найбільш вживані.

`axes` — логічна, якщо T — вісі координат відображаються, якщо F — ні.

`xlab`, `ylab` — символьні, задають текст, яким будуть підписані координатні вісі.

`sub` — задає підпис знизу під рисунком;

`main` — задає заголовок над рисунком.

`xlim`, `ylim` — задають межі для значень по горизонтальній та вертикальній осях.

`asp` — "aspect ratio" — співвідношення масштабних одиниць по вертикалі та горизонталі. Якщо не задавати цю опцію, то масштаб по вертикалі та горизонталі буде задаватись незалежно, виходячи із зручності розміщення рисунку. Якщо задати `asp=1` масштабні одиниці по обох осях матимуть однакову довжину.

Наступні опції можна використовувати як у `plot()`, так і у `points()`, `lines()` та ряді інших графічних функцій.

`type` — тип точок/ліній. Ця опція може приймати значення:

"p— відображати лише точки;

"l— відображати лише прямі лінії, що з'єднують задані точки;

"b— відображати і точки і лінії, причому лінії торкаються точок;

"o— відображати лінії, що перекривають точки;

"s "S— з'єднати точки ступінчастими лініями, ("s— стрибок ліворуч, "S— праворуч);

"h— відображаються вертикальні відрізки, що з'єднують задані точки з віссю абсцисс.

"n— на рисунку буде відображена координатна площина, у якій вміщуються всі задані точки, але ні самі точки, ні лінії, що їх з'єднують, не відображаються. (Ця опція використовується для того, щоб підготувати місце для рисування іншими функціями)

`col` — колір або кольори, яким будуть відображатись об'єкти (це може бути числовий вектор, або кольори можна задавати їх англійськими назвами "red "blue тощо).

`pch` — символ, яким відображаються точки.

`cex` — контролює розмір символів.

`lwd` — контролює ширину ліній.

Наприклад, результат виконання наступних команд відображено на рис 3.6:

```
> x<-1:10
> plot(c(0,10),c(-0.2,1.2),type="n")
> points(x,rep(1,10),pch=x,cex=2-0.1*x,col=x)
> points(x,rep(0,10),pch=10+x,col=10+x)
```

Для того, щоб відображати на рисунках написи у заданих точках використовується функція `text(x,y,labels,...)`. Тут вектори `x` та `y` задають координати точок де розміщується текст, `labels` — вектор символьних рядків, які будуть виводитись у заданих точках.

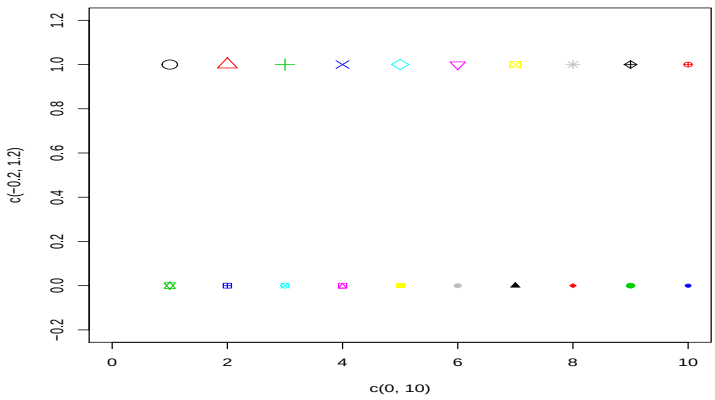

Рис. 3.6: Символи та кольори

Для цієї функції опція `pos` вказує як розміщується текст по відношенню до заданої точки (1 — під точкою, 2 — ліворуч, 3 — над, 4 — праворуч). `ofset` задає зміщення тексту по відношенню до точки.

Опції `cex` та `col` задають розмір символів та їх колір у `text()` так само, як у функції `plot()`.

Функція `segments(x0,y0,x1,y1)` рисує набір відрізків. У векторах `x0, y0` знаходяться x і y координати точок-початків відрізків, у `x1, y1` — точок-кінців.

Аналогічно, функція `arrows(x0,y0,x1,y1,length,angle)` рисує стрілки, `lenght` задає довжину "наконечника" стріли, `angle` — гостроту кута наконечника.

Функція `rect(xleft, ybottom, xright, ytop, density = NULL, angle = 45,col = NA,` рисує прямокутники, координати лівих нижніх кутів беруться з `xleft, ybottom`, правих верхніх — з `xright, ytop`. Прямокутник може заповнюватись штриховкою, щільність цієї штриховки задається `density`, кут нахилу — `angle`, колір — `col`. Параметр `border` визначає колір контуру прямокутника.

Наприклад, результат виконання наступних команд зображено на рис. 3.7

```
> plot(c(0,10),c(0,10),type="n")
> rect(1:9,rep(1,9),2:10,2:10,density=5:13,col=1:9,border=10:19)
> arrows(rep(2,4),4:8,4:8,5:9,angle=20,length=0.1)
```

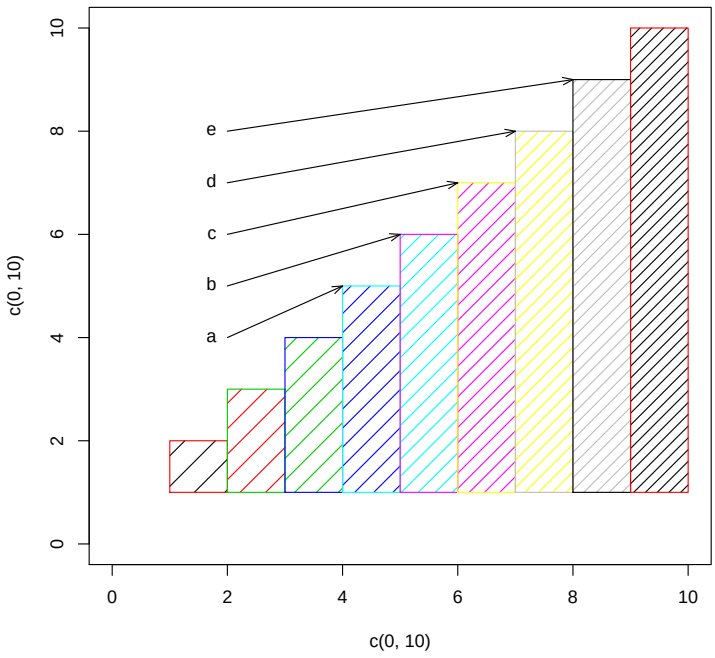

Рис. 3.7: Рисування прямокутників, стрілок та написів

```
> text(rep(2,4),4:8,labels=c("a","b",'c',"d","e"),pos=2)
```

Функція `abline(a,b)` рисує пряму лінію, що описується рівнянням $y = a + bx$. Якщо треба провести вертикальну лінію з горизонтальною координатою x, це можна зробити функцією `abline(h=x)`.

Для відображення графіків нелінійних функцій можна використовувати функцію

```
curve(expr,from=NULL,to=NULL,n=101,add=FALSE,
      type="l",xname="x",xlab=xname,ylab = NULL)
```

Параметри цієї функції:

`expr` — ім'я функції, що залежить від параметра `x`, або вираз, що залежить від змінної `x`;

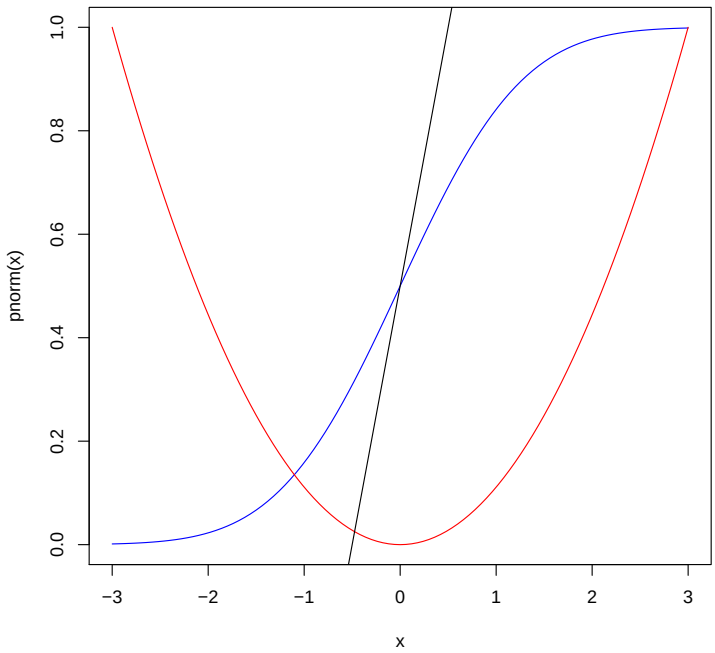

Рис. 3.8: Рисування кривих та прямих

`from`, `to` — лівий та правий кінці інтервалу зміни `x`, на якому будується графік;

`n` — кількість точок для малювання графіку;

`add` — логічний параметр, якщо він TRUE, графік будується на старому рисунку, якщо FALSE — для рисунку відкривається нове вікно;

`type, xlab, ylab` — такі ж, як у функції `plot`;

`xname` — ім'я що використовується для осі x.

Наприклад (див. рис. 3.8)

```
> curve(pnorm,-3,3,add=FALSE,col="blue")
> curve((x/3)^2,col="red",add=TRUE)
> abline(0.5,1)
```

Тут `pnorm` — функція розподілу для стандартного нормального розподілу (див. табл. 5.1).

У цьому прикладі другий виклик функції рисування кривих — `curve((x/3)^2,col="red",add=TRUE)` має опцію `add=TRUE`, тобто новий рисунок не створюється, крива відображається на старому. При цьому не задані параметри, що вказують діапазон по горизонталі (такі як `from`, `to`). У такій ситуації криву рисують через весь старий рисунок — від його лівого до правого поля.

3.3 Елементи тривимірної графіки

У статистиці досить часто спостережувані дані відображають у вигляді точок у просторі. Якщо кожній точці відповідає одне спостереження, а змінним — координати цієї точки, то такий рисунок називають діаграмою розсіювання. Якщо для відображення використовують лише дві змінні, утворюється двовоімірна діаграма розсіювання, яку можна вивести на екран функцією verb+plot()+ як описано вище.Для відображення трьох змінних одразу використовують тривимірні діаграми розсіювання. У `R` їх можна виводити багатьма різними способами, один з найпростіших — використання функції `scatterplot3D()` з пакету `scatterplot3`

Приклад виклику цієї функції:

```
> library(scatterplot3d)
> z <- seq(-20, 20, 0.15)
> x <- z*cos(z)
> y <- z*sin(z)
> scatterplot3d(x, y, z, highlight.3d=TRUE, col.axis="blue",
+              col.grid="lightblue", main="Spiral", pch=1,angle=30)
```

У виклику цієї функції перші три параметри `x, y, z` задають положення точок у тривимірному просторі. Параметр `highlight.3d` визначає, чи потрібно розфарбовувати точки в залежності від того, як вони розташовані по осі `x`. Два наступних параметри визначають колір осей координат та координатної сітки зображеної у площині x-y. Нарешті, параметр `angle` визначає кут, який будуть утворювати на двовимірній проекції вісі Ox та Oy. (Проекція завжди будується так, щоб вісь Ox на ній розташовувалась горизонтально, вісь Oz — вертикально, а от напрямок Oy на рисунку визначається напрямком проектування. Міняючи `angle` можна розглядати дані “з різних точок зору”.

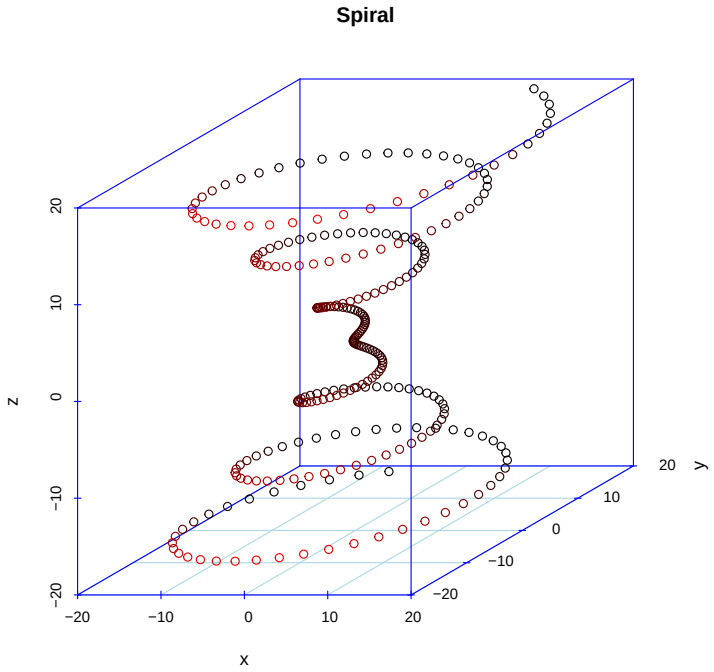

Рис. 3.9: Тривимірна діаграма розсіювання

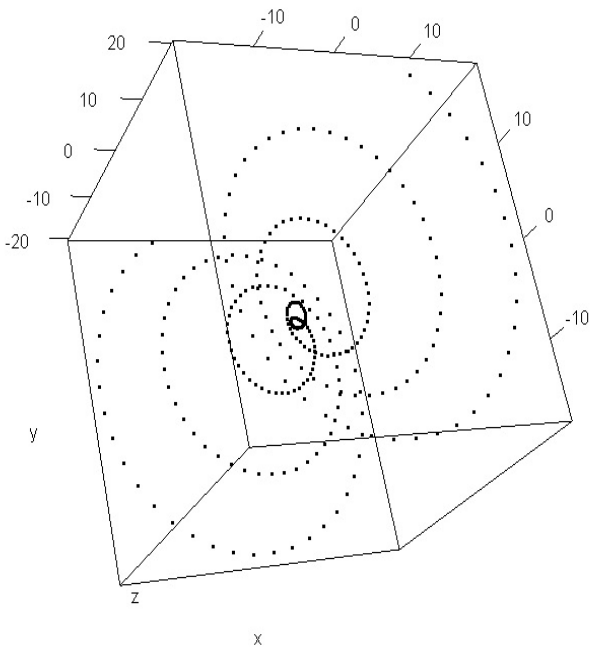

Рис. 3.10: Тривимірна діаграма розсіювання за допомогою plot3d

Як бачимо, можливості `scatterplot3D()` при виборі напрямку проекції досить обмежені. Для того, щоб мати змогу повертати зображення даних інтерактивно, можна скористатись функцією `plot3d()` з пакету `rgl`. Наприклад, викликавши цю функцію на даних попереднього прикладу у такий спосіб

```
library(rgl)
plot3d(x,y,z)
```

і покрутивши отриманий рисунок мишею, можна отримати зображення з рис. 3.10.

Менш часто ніж діаграми розсіювання, але інколи також буває потрібно відображати на рисунках поведінку числових функцій двовимірного аргументу. Досить популярним засобом такого відображення є контурні графіки (contour plot), на яких зображають "лінії рівня" функції.

Лінія рівня функції $f(x, y)$, що відповідає рівню c, це множина всіх точок на площині з такими координатами (x, y), що $f(x, y) = c$. На кон-

турних графіках відображають лінії рівня для різних рівнів, підбираючи їх так, щоб можна було побачити горби та западини функції подібно до того, як це роблять на географічних картах. У R контурні графіки зображають, використовуючи функцію `contour()`.

Інший варіант тривимірного графіка — зображення його проекції на площину аналогічно тому, як це було зроблено вище для діаграм розсіювання. Таке відображення тривимірних графіків у перспективі забезпечує функція `persp()`.

І `contour()` і `persp()` працюють не безпосередньо з функцією f яку потрібно відобразити, а із матрицею `z` значень цієї функції у вузлах прямокутної сітки, визначеної векторами координат `x`, `y`, тобто `z[i,k]=f(x[i],y[k])`, `i=1,...,length(x)`, `k=1,...,length(y)`. Для того, щоб підраховувати значення `z` зручно використовувати функцію `outer`/. Приклад застосування `contour()` і `persp()` для відображення функції $f(x, y) = \sin^2 x + \cos^2 x$ коли $x \in (0, 5)$, $y \in (2, 7)$:

```
> x<-(1:50)/10
> y<-(20:70)/10
> f<-function(x,y){sin(x)^2+cos(y)^2}
> z<-outer(x,y,f)
> contour(x,y,z)
> persp(x, y, z, theta = 30, phi = 30, expand = 0.75,
+ col = "lightblue")
```

Результат виконання цього скрипту на рис. 3.11. Функція `persp()` має більше можливостей вибору напрямку проекції, ніж `scatterplot3D()`. В ній цей напрямок задається двома кутами `theta` (азимут) та `phi` (90^o – широта). Крім того, параметр `expand` (як правило, його обирають від 0 до 1) можна використовувати для стиснення графіка по осі `z`.

3.4 Географічні карти

Статистична інформація часто має географічну прив'язку, тому для її відображення природно використовувати географічні карти. У R передбачений великий вибір можливостей такого відображення. У цьому підрозділі розглядаються лише два найпростіші приклади: відображення інформації фарбуванням різних областей різними кольорами та відображення кругових діаграм на географічних картах.

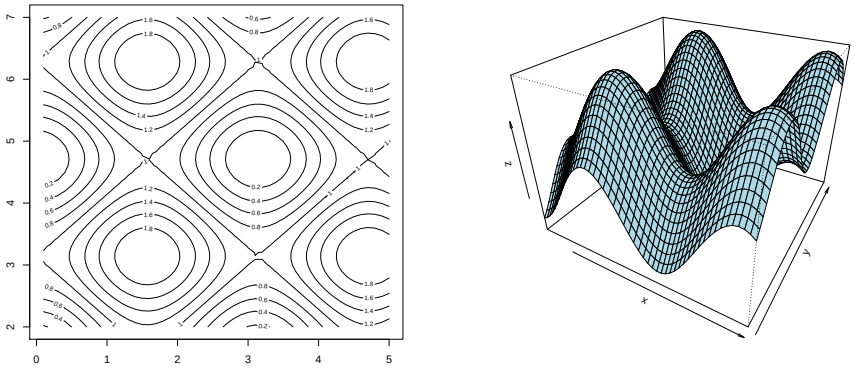

Рис. 3.11: Контурний та тривимірний графіки

Спочатку розберемось, як у R малюються географічні карти. Це можна робити багатьма різними способами. Для нашої мети одним з найпростіших є використання пакетів (бібліотек) `sp`, `maptools` та `raster`. Вони не входять у стандартну поставку R , тому їх потрібно інсталювати на комп'ютері звичайним способом та завантажувати перед використанням у робочу область використовуючи функцію `library()`.

У пакеті `maptools` міститься політична карта земної кулі під назвою `wrld_simpl`. Для того, щоб вивести її у вигляді рисунку можна скористатись звичайною функцією `plot()`, наприклад:

```
> library('sp')
> library('maptools')

Checking rgeos availability: FALSE
         Note: when rgeos is not available, polygon geometry         computations
         which has a restricted licence. It is disabled by default;
         to enable gpclib, type gpclibPermit()

> data(wrld_simpl)
> par(mai=c(0,0,0,0))
> par(mar=c(0,0,0,0))
> plot(wrld_simpl, xlim=c(-10,50),
+    ylim=c(-40,35), bg='azure2', col='khaki',
+    border='black')
```

Рис. 3.12: Карта Африки

Тут ми спочатку функцією `data()` завантажили потрібну змінну у пам'ять. Потім задали командою `par` нульовий розмір полів рисунку.

І, нарешті надрукували карту функцією `plot()`, використовуючи параметри:

`col` — колір, яким фарбується основна частина (суходіл),

`bg` — колір заднього плану (background) — море,

`border` — колір для границь позначених на карті країн,

`xlim`, `ylim` — межі регіону, який потрібно відобразити на карті по горизонталі та вертикалі.

На картах використовується географічна шкала координат, горизонтальна вісь відповідає довготі (longitude), вертикальна — широті (latitude). Як відомо, це кутові міри, вони визначаються у градусах та хвилинах. Один градус складає 60 хвилин. У `R` використовується звичайне десяткове позначення для цих координат, тобто, скажімо, latitude -10.5 це 10 градусів 30 хвилин південної широти.

Результат зображено на рис. 3.12. (Карта вже дещо застаріла, на ній не відмічено, наприклад, таку країну, як Південний Судан).

Карти кордонів окремих країн також містяться у об'єкті `wrld_simpl`. Їх можна використовувати, звертаючись по номерах країн у списку, розта-

шованому приблизно за алфавітним порядком. Цей список міститься у атрибуті `word_simp$NAME`. Вивівши цей список у `R` , можна побачити, що Демократична республіка Конго має номер 28, Нігерія — 153, Мадагаскар — 108. Нехай ми хочемо на вже існуючій карті Африки відмітити ці країни різними кольорами: Нігерію — зеленим, Мадагаскар — червоним, Конго — білим. Це можна зробити викликваши знову функцію `plot()` з параметром `add=T`, що означає — дорисувати новий рисунок поверх попереднього:

`plot(wrld_simpl[c(28,153,108),],col=c('white','green','blue'),add=T).`

Написи на картах можна наносити використовуючи функцію `text` так, як це було описано у п. 3.2.

Покажемо, як відображати на картах кругові діаграми, подібні до тих, що описані у п. 3.1. Для цього ми використаємо функцію `floating.pie()` з пакету `plotrix` (його треба інсталювати на комп'ютері та завантажити).

Цю функцію можна викликати так:

`floating.pie(xpos,ypos,x,col,radius)`

де

`xpos`, `ypos` — координати центру кругової діаграми на рисунку (на карті),

`x` — вектор, координати якого відповідають розмірам секторів на круговій діаграмі.

`col` — кольори секторів,

`radius` — радіус діаграми.

Нехай, наприклад, ми хочемо для вибраних нами країн відобразити круговими діаграмами розподіл населення за релігійною ознакою. Для Нігерії, скажімо, це виглядає так: 58% християн, 41% — прибічники ісламу, 1% — інші релігії. Програма може мати такий вигляд:

```
>  par(mai=c(0,0,0,0))
>  par(mar=c(0,0,0,0))
>  plot(wrld_simpl, xlim=c(-10,50),
+     ylim=c(-40,35), bg='azure2', col='khaki',
+     border='black')
> # Brushing of Congo, Nigeria, Madagaskar
> plot(wrld_simpl[c(28,153,108),],  col=c('white','green','blue'),
+      add=T)
> library(plotrix)
```

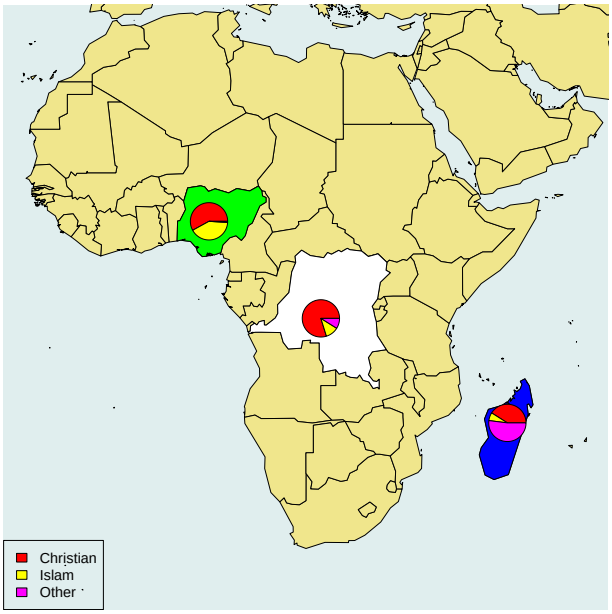

Рис. 3.13: Карта Африки з розподілом релігій

```
> # Nigeria
> floating.pie(7,9,c(58,41,1),col=c("red","yellow","magenta"),radius=2.5)
> # Congo
> floating.pie(22,-4,c(79.8,11.3,8.9),col=c("red","yellow","magenta"),radius=2.5)
> # Madagaskar
> floating.pie(47,-18,c(41,7,52),col=c("red","yellow","magenta"),radius=2.5)
> legend("bottomleft",
+       legend=c("Christian","Islam","Other"),fill=c("red","yellow","magenta"))
```

Спочатку рисуємо карту Африки як у попередньому прикладі, потім розфарбоувуємо три країни і рисуємо кругові діаграми. Остання виконана функція — `legend()` створює легенду (пояснення) до карти. Перший параметр `"bottomleft"` визначає положення легенди у лівому нижньому кутку карти. Параметр `legend` це вектор символьних рядочків, кожен з яких відповідає одному рядочку легенди. Параметр `fill` задає кольори, які будуть пояснюватись легендою.

Карти країн, що містяться у наборі `wrld_simpl` є досить грубими, вони не містять адміністративного поділу. Тому для того, щоб відображати статистику по регіонах якої-небудь країни, потрібні більш детальні

карти. Користування такими картами надає пакет `raster`. (Не забудьте його завантажити). У цьому пакеті є функція `getData()`, яка завантажує з інтернету карти різних країн за їх кодами `ISO`. Отримати перелік всіх країн з їх кодами можна, викликавши

```
getData('ISO3')
```

Наприклад, код України — `UKR`. Для завантаження карти викликаємо `getData()`:

```
library(raster)
ukraine <- getData('GADM', country='UKR', level=1)
```

Параметр `'GADM'` показує, що карта буде завантажена з бази даних про адміністративні кордони (є іще бази кліматичних та топографічних карт). Параметр `country='UKR'` вказує, що буде завантажуватись карта України. Параметр `level` визначає деталізацію карти. Значенню 0 відповідає карта з лише державними кордонами, 1 — регіональні кордони (для України — областні, для США — штатів, для Польщі — воєводств), 2 відповідає районам для України, графствам для США, повітам для Польщі. Можливий також 3й рівень для іще дрібніших одиниць (гміни у Польщі).

Таким чином, ми завантажили карту України з границями областей і зберегли її у вигляді змінної `ukraine`. Ця змінна знаходиться у робочій області `R` . Якщо робочу область не зберігати наприкінці сеансу, карта загубиться. Доцільно зберегти її окремо у файлі для подальшого користування. це можна зробити, використовуючи функцію `save()`:

```
save(ukraine,file="c:/rem/term/ukrmap.Rdata")
```

— зберігає карту у вигляді об'єкту `R` на диску `c` у каталозі `term` під назвою `ukrmap.Rdata`. Назва і каталог можуть бути довільними, розширення `Rdata` стандартне для `R` , втім, при бажанні можна використовувати і інші розширення, але формат файлу при цьому не зміниться (тобто, якщо вказати `ukrmap.pdf`, `R` збереже карту у такому файлі, але не у форматі pdf, а у своєму внутрішньому форматі).

Для того, щоб завантажити збережену карту під час нової сесії роботи з `R` тепер досить набрати

```
load(file="c:/rem/term/ukrmap.Rdata")
```

— після цього карта стане доступною у вигляді об'єкту з назвою `ukraine`. Її можна надрукувати, використовуючи `plot()`.

Відображати карти окремих регіонів тепер можна, викликаючи цю функцію, наприклад, так: `plot(ukraine[list_reg],col=list_col)`, де

`list_reg` — список номерів регіонів, які треба відобразити, `list_col` — список кольорів, якими ці регіони будуть зафарбовані. Регіони у змінній `ukraine` розташовані у алфавітному порядку їх англійських назв. Щоб побачити ці назви і їх порядок можна вивести атрибут `NAME_1` змінної `ukraine`, тобто `ukraine$NAME_1`.

Як приклад, розглянемо відображення густоти населення України у різних областях (див. рис. 3.14).

```
> library(raster)


Attaching package: 'raster'

Следующие объекты скрыты от 'package:MASS':

    area, select

> load(file="c:/rem/term/ukrmap.Rdata")
> dens<-read.csv(file="c:/rem/term/gustotan.csv")
> brk<-seq(30,170,20)
> int<-8-findInterval(dens[,2],brk)
> palette(gray(0:7/7))
> par(mai=c(0,0,0,0))
> par(mar=c(0,0,0,0))
> plot(ukraine,col=int)
> plot(ukraine[c(11,20),],col="red",add=T)
> legend("bottomleft",title="Population Density",
+      legend=c("150-170","130-150","110-130","90-110","70-90","50-70","30-50"),
+      fill=gray(0:7/7))
```

Дані по густоті населення знаходяться у файлі `gustotan.csv`. Перший стовпчик таблиці даних зветься `region` і містить англійські назви областей (регіонів України) в алфавітному порядку. Другий — густоту населення (кількість чоловік на кв.км) у даному регіоні. Ці дані коливаються у діапазоні від 32.9 у Чернігівській області до 3442.6 у місті Києві. Зрозуміло, що міста Київ та Севастополь у цьому наборі даних різко виділяються (є викидами) тому при побудові шкали густот їх краще не враховувати. Серед областей найбільшу густоту населення має Донецька —

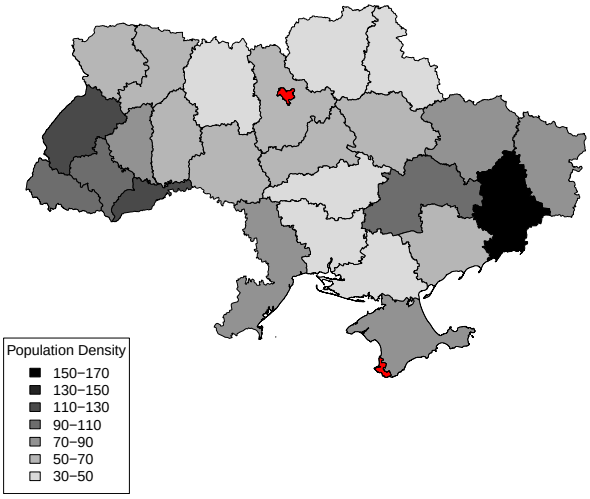

Рис. 3.14: Густота населення України

161.3. Тому ми вибрали інтервал від 30 до 170 і розбили його на підінтервали ширини 20. Кожному підінтервалу відповідає певна насиченість білого/сірого/чорного кольору, яким зафарбовується область. Така палітра кольорів створюється функцією `grey(0:7/7)` (0 відповідає чорний, 1 — білий колір). Функція `palette(grey(0:7/7))` встановлює такий набір кольорів як палітру, що використовується іншими функціями, подібними до `plot()`. Щоб визначити, який номер кольору відповідає густоті населення кожної області використовується функція `findInterval`, котра визначає, якому з інтервалів, заданих набором точок `brk` належать густоти різних областей (ці густоти знаходяться у другому стовпчику фрейму `dens`).

Далі ми розфарбовуємо всі регіони функцією `plot()`. Щоб виділити Київ та Севастополь (11-й і 20-й регіони), зафарбовуємо їх червоним кольором, використовуючи `plot()` з параметром `add=T` — тобто зверху попереднього рисунку. Нарешті `legend()` додає пояснення кольорової шкали у лівому нижньому кутку карти.

Розділ 4

Описова статистика

Статистик, як правило, має справу з великими обсягами даних. Тому часто виникає потреба описати основні особливості таких даних одною або кількома числовими характеристиками. Техніка такого опису зветься описовою (дескриптивною) статистикою, а самі числові характеристики даних — (дескриптивними) статистиками. При використанні та аналізі таких статистик дослідник намагається вивчати дані не на основі якоїсь наперед заданої теоретичної моделі, а виходячи з структури самих даних. У цьому розділі ми розглядаємо техніку дескриптивної статистики саме з такої точки зору. Значна кількість дескриптивних статистик може використовуватись також у рамках певних теоретичних моделей, скажімо, як оцінки параметрів моделі, статистики тестів для перевірки гіпотез, прогнози для очікуваних спостережень. Такі застосування розглядаються у наступних розділах, але інколи, для пояснення переваг тої чи іншої статистики ми будемо згадувати трактовку даних як "кратної вибірки" — набору незалежних, однаково розподілених випадкових величин. Читачі, яким така трактовка не зовсім зрозуміла, або здається недоречною при застосуванні до їх даних, можуть просто пропускати такі пояснення.

4.1 Описова статистика одновимірних числових даних

У цьому підрозділі ми обговоримо основні способи опису наборів числових статистичних даних, у яких для кожного спостереження вимірюється одна числова характеристика (змінна). Наприклад, спостережувани-

ми об'єктами можуть бути призовники до армії, а змінною, що досліджується — їх зріст. (Властивості цієї характеристики важливі для тих, хто займається забезпеченням одягом). Інший приклад — вимірювання температури повітря на вулиці, які проводяться протягом року щодня о певній годині. Тут кожне спостереження відповідає дню вимірювання, а змінною, що досліджується є температура.

У випадку зросту призовників порядок, в якому розташовані об'єкти у наборі несуттєвий, він склався випадково і не пов'язаний з досліджуваним явищем. Перетасувавши виміряні значення зросту у довільному порядку ми не втрачаємо корисної інформації. Такі набори даних прийнято називати вибірками.

Для вимірювань температури порядок суттєвий: температура на вулиці залежить від пори року, сьогоднішня температура залежить від вчорашньої і т.д. Дані, для яких важливими є такі ефекти, називають часовими рядами. Зрозуміло, що перетасувавши елементи часового ряду ми втратимо інформацію про ці залежності. Але інформація про деякі важливі особливості досліджуваної температури збережеться: якщо, наприклад, нас цікавить найбільша температура протягом року, на порядок вимірювань можна не звертати уваги. При дослідженні таких особливостей часові ряди (з певними застереженнями) можна розглядати як вибірки.

У цьому підрозділі ми зосередимось на аналізі вибірок, тобто таких наборів даних, для яких порядок спостережень несуттєвий.

Надалі ми будемо позначати X_j — значення досліджуваної змінної у j-тому спостереженні, n — кількість елементів у вибірці, $\mathbf{X} = (X_1, \ldots, X_n)$ — вибірка.

4.1.1 Статистики середнього положення

SecCentralPos

Найпростіший спосіб охарактеризувати вибірку в цілому одним числом полягає в тому, щоб вказати "середнє положення", "центр вибірки" навколо якого коливаються вибіркові значення. Існує багато способів визначення такого числа і, відповідно, різні статистики середнього положення. Далі ми розглянемо найбільш поширені з них та обговоримо їх властивості.

Вибіркове середнє — статистика, що першою спадає на думку, коли потрібно визначити центр вибірки. Для вибірки $\mathbf{X}$ воно визначається за

формулою

$$\bar{X} = \frac{1}{n}\sum_{j=1}^{n} X_j.$$

(У статистиці вибіркове середнє стандартно позначається тією ж літерою, що і усереднювана змінна з рискою над нею.)

Поширеність цієї статистики пов'язана із загальновживаним способом міркування на зразок: "середня врожайність цього сорту картоплі у наших умовах складає 20 тон з гектару, отже з наших трьох гектарів можна сподіватись приблизно 60 т. врожаю". В основі такого прогнозування лежить уявлення про стабільність середніх по великих обсягах даних: врожай з одного куща картоплі міняється під впливом багатьох причин, але при усередненні по гектару індивідуальні особливості, що змінюють врожай різних кущів у різних напрямках, взаємно врівноважуються і отримується результат, котрий має бути приблизно однаковим для всіх трьох гектарів нашого поля. У теорії ймовірностей цей ефект називають законом великих чисел і він є властивістю вибіркових середніх для даних, що описуються певними теоретичними моделями.

Крім закону великих чисел у нашому міркуванні була використана іще "властивість адитивності" врожаю: повний врожай з усіх ділянок дорівнює сумі врожаїв кожної окремої ділянки.

Інше просте пояснення цієї характеристики — комуністичне: "якщо у всіх відібрати і розділити порівну, то кожен отримає вибіркове середнє". Це пояснення дозволяє зрозуміти, чому у деяких випадках вибіркове середнє заміняють іншими характеристиками.

Середнє геометричне визначається для вибірок $\mathbf{X} = (X_1, \ldots, X_n)$, у яких значення змінної X_j можуть бути лише додатними. Воно дорівнює

$$\mathrm{GM}(\mathbf{X}) = \left(\prod_{j=1}^{n} X_j\right)^{1/n}. \tag{4.1}$$

Прикклад 1. (Застосування геометричного середнього у фінансовій математиці). Нехай укладено кредитну умову на n років, у якій боржник на початку терміну умови отримує суму S. За перший рік кредитування нараховується відсоток p_1, за другий — p_2 і т.д. Нарахування відбувається за схемою складних відсотків. Сплата боргу з усіма відсотками передбачається наприкінці терміну дії угоди. Як визначити середній відсоток по кредитуванню за цією угодою?

Що таке середній відсоток? Це таке $\hat{p}$, що, якби ми уклали угоду на n з фіксованим відсотком $\hat{p}$, то виплата при поверненні боргу були б такі самі, як і в розглянутій угоді зі змінними відсотками.

Виплата у схемі змінних відсотків дорівнює

$$S_n = S\prod_{j=1}^{n}(1 + p_j/100),$$

а виплата з фіксованим відсотком $\hat{p}$ —

$$S'_n = S(1 + \hat{p}/100)^n.$$

Прирівнюючи S_n і S'_n, отримуємо

$$\hat{p} = 100\left(\prod_{j=1}^{n}(1 + p_j/100)\right)^{1/n} - 100.$$

У цьому виразі легко побачити геометричне середнє величин $X_j = (1 + p_j/100)$ — приростів боргу протягом j-того року дії угоди.$\diamond$

Отже, геометричне середнє природно застосовувати там, де загальний ефект виражається не як сума, а як добуток ефектів окремих спостережень.

Відмітимо також, що логарифм геометричного середнього є вибірковим середнім логарифмів спостережень:

$$\log(\mathrm{GM}(X)) = \overline{\log(X)}.$$

Можна сказати, що логарифмічне перетворення даних переводить геометричні середні у вибіркові.

Середнє гармонійне це величина, обернена до вибіркового середнього обернених величин спостережень:

$$\mathrm{HM(X)} = \frac{n}{\sum_{j=1}^{n}\frac{1}{X_j}} = \frac{1}{\overline{1/X}}. \tag{4.2}$$

Гармонійні середні природно застосовувати для характеризації середніх положень змінних, які самі можна визначити, як відношення двох характеристик одного об'єкта, якщо чисельник є менш мінливим ніж знаменник.

Прикклад 2. (Середнє гармонійне у підрахунку mpg) Важливою характеристикою економічності автомобіля є шлях, який він проходить, витративши одиницю об'єму пального. У країнах з британською системою мір ця величина визначається у мілях шляху на галон пального і позначається mpg.

Для визначення mpg даного автомобіля використовуються тестові поїздки по заданому маршруту. Якщо довжина маршруту у мілях S, а об'єм витраченого пального у галонах — V, то $mpg = S/V$. Для надійності тестові поїздки повторюють декілька разів по одному маршруту, отримуючи різні об'єми витраченого пального V_1, V_2,...,V_n. Відповідно, для кожного тесту можна визначити своє значення $mpg_j = S/V_j$. Середнє значення mpg за всіма тестами природно визначити як відношення загальної довжини пройденого в усіх тестах шляху до об'єму всього витраченого пального:

$$\widehat{mpg} = \frac{\sum_{j=1}^{n} S}{\sum_{j=1}^{n} V_j} = \frac{n}{\sum_{j=1}^{n} 1/mpg_j} = \text{НМ(mpg)}.$$

Таким чином, для усереднення mpg отриманих у серії тестових поїздок слід використовувати середнє гармонійне.

На основі схожих міркувань рекомендується застосовувати середнє гармонійне для визначення середнього значення коефіцієнту ціна/прибуток (P/E, earnings multiple) при порівнянні інвестиційної привабливості акціонерних компаній[3].

Забруднення і робастність. При виборі статистики для характеризації середнього положення вибірки доцільно враховувати можливість забруднень. Забрудненою зветься вибірка, у якій присутні значення, що не пов'язані з досліджуваним явищем, а потрапили до неї внаслідок помилки. Якщо таке неадекватне значення можна розпізнати і вилучити з вибірки, його називають "грубою помилкою" (наприклад, якщо вибірка складається з зростів людей, всі від'ємні значення у ній будуть грубими помилками).

Але бувають забруднення, які не можна однозначно розпізнати, тому вони впливають на значення сумарних статистик, що обчислюються за вибіркою. Якщо дослідник не може з теоретичних міркувань виключити таку можливість, то для загальної характеризації вибірки бажано використовувати статистики, які не дуже сильно змінюються при наявності невеликої кількості забруднень. Такі статистики називають робастними (стійкими по відношенню до забруднень).

Наприклад, вибіркове середнє $\bar{X}$ не є робастним: забруднення, при якому змінюється одне єдине значення у вибірці X може змінити $\bar{X}$ як завгодно сильно, якщо змінене значення обрати достатньо великим.

Те ж можна сказати і про середнє геометричне: збільшуючи лише один множник у добутку (4.1) можна зробити весь добуток, а отже і середнє як завгодно великим. А от для середнього гармонійного це невірно. Дійсно, якщо у (4.2) одне спостереження, наприклад, X_n спрямувати до нескінченності, то середнє гармонійне прямуватиме до

$$\frac{n}{\sum_{j=1}^{n-1} 1/X_j},$$

тобто до величини, яка при великих n, приблизно дорівнює гармонійному середньому, обчисленому за даними $X_1, \ldots, X_{n-1}$. Тобто наявність одного великого викиду змінює гармонійне середнє не дуже сильно. Але якщо спрямувати X_n до 0, то гармонійне середнє вибірки прямуватиме до 0, тобто до величини, шо може як завгодно сильно відрізнятись від гармонійного середнього початкової вибірки. Отже, для гармонійного середнього небезпечними є не великі, а малі (близькі до 0) викиди.

Зрізані середні. Розглянуті середні характеристики можна зробити стійкими до невеликої кількості забруднень, якщо застосувати техніку зрізання (truncation, trimming).

Переставимо елементи нашої вибірки у порядку зростання:

$$X_{[1]} \le X_{[2]} \le \cdots \le X_{[n-1]} \le X_{[n]}.$$

Тут $X_{[1]}$ — найменше значення у вибірці, $X_{[2]}$ — наступне за величиною, і т.д. аж до $X_{[n]}$ — найбільшого значення. $X_{[j]}$ називають j-тою **порядковою статистикою** вибірки $\mathbf{X}$, а послідовність порядкових статистик — **варіаційним рядом**.

Для того, щоб знайти зрізане середнє вибірки з рівнем зрізання α, потрібно відкинути $\lceil n\alpha/2 \rceil$ найбільших та $\lceil n\alpha/2 \rceil$ найменших порядкових статистик і усереднити те, що залишилось:

$$\mathrm{TM}_\alpha(X) = \frac{1}{n - 2\lceil n\alpha/2 \rceil} \sum_{j=\lceil n\alpha/2 \rceil + 1}^{n - \lceil n\alpha/2 \rceil} X_{[j]}. \tag{4.3}$$

Аналогічно можна використовувати зрізане геометричне або гармонійне середнє.

Чим більшою вибрати частку відкинутих порядкових статистик, тим більш стійким до забруднення буде зрізане середнє. Граничний випадок досягається, коли відкидають всі спосереження крім того одного або двох, що знаходяться посередині варіаційного ряду. В результаті отримуємо характеристику середнього положення, яка зветься вибірковою медіаною.

Вибіркова медіана це статистика, що обчислюється за формулою

$$\mathrm{med}(X) = \begin{cases} X_{[(n+1)/2]}, & \text{якщо } n \text{ непарне}, \\ \frac{1}{2}(X_{[n/2]} + X_{[n/2+1]}), & \text{якщо } n \text{ непарне}. \end{cases} \tag{4.4}$$

Коротко можна сказати, що медіана — це середина варіаційного ряду: ліворуч від медіани знаходиться стільки ж значень, скільки і праворуч.

Медіана — найбільш робастна характеристика середнього положення у вибірці. Цим, значною мірою, пояснюється її популярність у багатьох застосуваннях.

Можна сказати, що порядкові статистики, які розташовані поблизу середини варіаційного ряду є найбільш робастними. І навпаки — найбільш чутливими до забруднень є "екстремальні" порядкові статистики $X_{[1]} = \min(X_1, \ldots, X_n)$ та $X_{[n]} = \max(X_1, \ldots, X_n)$. Інтервал $[X_{[1]}, X_{[n]}]$ називають діапазоном вибірки, а величину

$$\mathrm{MR}(X) = \frac{X_{[1]} + X_{[n]}}{2}$$

— **серединою діапазону** (midrange). $\mathrm{MR}(X)$ також є характеристикою середнього положення у вибірці, хоча і зовсім не робастною. Скажімо, якщо у вибірці є одне забруднення, воно може помітно змінити вибіркове середнє. Але при зростанні обсягу вибірки вплив цього забруднення буде зменшуватись. Для середини діапазону це не так: одне значення забруднення, яке є більшим ніж всі спостережувані значення досліджуваної змінної, залишиться $X_{[n]}$, скільки б нових спостережень ми ні зробили. Отже, використовувати середину діапазону для характеризації середнього положення слід дуже обережно.

Однак, у тих випадках, коли забруднень немає, середина діапазону може виявитись значно більш точною оцінкою теоретичного середнього положення (скажімо, для математичного сподівання рівномірного розподілу за кратною вибіркою) ніж вибіркове середнє.

4.1.2 Статистики розкиду

Щоб одним числом показати, як далеко вибіркові значення можуть відхилятись від середнього положення, використовують статистики розкиду.

Найбільш популярною такою статистикою є **вибіркова дисперсія** (sample variance). Вона визначається як середнє квадратів відхилень спостережень від вибіркового середнього:

$$S^2(X) = \frac{1}{n}\sum_{j=1}^{n}(X_j - \bar{X})^2. \tag{4.5}$$

Часто використовується **виправлена вибіркова дисперсія**, яка відрізняється від звичайної лише нормуючим множником[1] $(n-1)/n$:

$$S_0^2(X) = \frac{1}{n-1}\sum_{j=1}^{n}(X_j - \bar{X})^2.$$

Використання виправленої вибіркової дисперсії пов'язане з тим, що вона є незміщеною оцінкою для теоретичної дисперсії по кратній вибірці. У багатьох підручниках та комп'ютерних програмах $S_0^2(X)$ називають просто вибірковою дисперсією, а $S^2(X)$ — популяційною дисперсією, або дисперсією генеральної сукупності. Вибіркову дисперсію інколи позначають σ^2.

Корінь квадратний з вибіркової дисперсії називають (вибірковим) **середньоквадратичним відхиленням** (або стандартним відхиленням):

$$S(X) = \sqrt{S^2(X)} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(X_j - \bar{X})^2}$$

і, аналогічно,

$$S_0(X) = \sqrt{S_0^2(X)} = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(X_j - \bar{X})^2}.$$

[1] Цей множник називають поправкою Бесселя.

Вибірковим середнім абсолютним відхиленням (mean absolute deviation або просто mean deviation) називають середнє абсолютних відхилень вибіркових значень від вибіркового середнього:

$$\mathrm{MAD}(X) = \frac{1}{n}\sum_{j=1}^{n} |X_j - \bar{X}|.$$

Інколи використовують також середнє абсолютне відхилення від медіани:

$$\mathrm{MAD}_\mu(X) = \frac{1}{n}\sum_{j=1}^{n} |X_j - \mathrm{med}(X)|.$$

Зрозуміло, що вибіркова дисперсія та середнє абсолютне відхилення не є робастними — одне забруднення, що лежить далеко від інших спостережень може змінити ці характеристики як завгодно сильно. Тому для забруднених вибірок розроблені спеціальні робастні характеристики розкиду, серед яких найбільш поширений інтерквартильний розмах.

Квартилі та інтерквартильний розмах (quartiles and interquartile range). Як ми знаємо, медіана розбиває варіаційний ряд на дві частини однакового розміру. Медіани кожної з цих двох частин називають квартилями. За тою частиною, де значення менше медіани всієї вибірки визначають лівий (перший) квартиль $Q_1(X)$, а за тією, у якій значення більші медіани — правий (третій) квартиль $Q_3(X)$. Медіану інколи звуть другим квартилем $\mathrm{med}(X) = Q_2(X)$. Таким чином, квартилі розбивають вибірку на чотири частини приблизно однакового розміру.

Інтерквартильний розмах це відстань від лівого до правого квартиля:

$$\mathrm{IQ}(X) = Q_3(X) - Q_1(X).$$

Шириною інтервалу або розмахом вибірки (range) називають

$$\mathrm{Range}(X) = X_{[n]} - X_{[1]},$$

тобто відстань від найменшого до найбільшого значення у вибірці. Зрозуміло, що це найменш стійка до забруднень характеристика розкиду вибірки.

4.1.3 Групування та навантаження

Групування. Серед даних у вибірці можуть зустрічатись однакові значення. Якщо різних значень, які набуває змінна порівняно небагато і

більшість з них зустрічається у вибірці кілька разів, то зручно не виписувати всю вибірку, а перелічити ці різні значення і вказати їх частоти (кількість повторень). Запис вибірки у такому вигляді називають групуванням, а саму вибірку — групованою (grouped).

Нехай $x_1 < \cdots < x_K$ — всі різні значення, яких може набувати досліджувана змінна (варіанти). Абсолютна частота n_i варіанти x_i у вибірці $X = (X_1, \ldots, X_n)$ це кількість номерів[2] $j = 1, \ldots, n$, для яких $X_j = x_i$.

Груповані дані часто записують у вигляді таблиці

Варіанти	x_1	x_2	$\ldots$	x_K
Частоти	n_1	n_2	$\ldots$	n_K

яку називають рядом розподілу вибірки.

Ситуація "групованої вибірки" природно виникає, наприклад, тоді, коли досліджувана величина є цілочисловою по своїй суті. Скажімо, це може бути кількість бракованих виробів виявлених на контролі протягом одного для виробництва. У інших випадках групування виникає внаслідок обмеженої точності вимірювання досліджуваних величин: якщо довжину комах вимірювати лінійкою, на якій є лише міліметрові поділки, то результат вимірювання у міліметрах буде цілим числом, хоча справжні довжини можуть приймати в принципі, будь-які додатні значення.

Нарешті, інколи виникає потреба[3] провести "примусове групування" (grouping або binning[4]), коли дані спеціально огрублюють. У цьому випадку весь інтервал $[a, b]$ можливих значень змінної розбивають на підінтервали $A_k = [t_{k-1}, t_k)$, де $a = t_0 < t_1 < \cdots < t_K = b$ — деякі точки. (Наприклад, при рівномірному розбитті беруть $t_k = a + kh$, де $h = (b-a)/K$ — ширина інтервалу розбиття). Якщо спостережуване значення X_j потрапляє у інтервал A_k, його заміняють на значення середини цього інтервалу $x_k = (t_k + t_{k-1})/2$. Утворену огрублену вибірку групують. Зрозуміло, що в такій вибірці n_k це кількість тих спостережень, які потрапили у інтервал A_k.

Статистики групованих вибірок.

Легко бачити, що вибіркове середнє по груповані вибірці можна під-

[2]Кількість об'єктів у вибірці

[3]Наприклад, при побудові гістограм, або при застосуванні тестів типу χ^2.

[4]інтервали розбиття називають bins — кошики, відповідно, примусово груповану вибірку — binned sample

рахувати так:

$$\bar{X}_w = \frac{1}{n}\sum_{j=1}^{n} X_j = \frac{1}{n}\sum_{k=1}^{K} n_k x_k = \sum_{k=1}^{K} w_k x_k,$$

де $w_k = n_k/n$ — вагові коефіцієнти (навантаження, ваги, weights).

Аналогічно, середнє геометричне рахується за формулою

$$\mathrm{GM}_w(X) = \left(\prod_{k=1}^{K} (x_k)^{n_k}\right)^{1/n} = \prod_{k=1}^{K} (x_k)^{w_k}.$$

Дещо складніше визначити медіану групованої вибірки. Для цього потрібно знайти таке значення k, для якого $\sum_{i:x_i<x_k} w_i < 1/2$ і, в той же час, $\sum_{i:x_i>x_k} w_i \le 1/2$. Тоді x_k буде вибірковою медіаною групованої вибірки. (При такому означенні вибіркова медіана, підрахована по групованій вибірці, може трохи відрізнятись від медіани, підрахованої без групування).

Груповане середнє абсолютне відхилення можна рахувати як

$$\mathrm{MAD}_w(X) = \sum_{k=1}^{K} w_k |x_k - \bar{X}_w|.$$

Групована вибіркова дисперсія має вигляд

$$S_w^2(X) = \sum_{k=1}^{K} w_k (x_k - \bar{X}_w)^2.$$

Якщо вибірка є природно групованою (скажімо, змінна приймає лише цілочислові значення) то $S_w^2(X)$ це в точності теж саме, що звичайна вибіркова дисперсія. Але, якщо дані були огрублені примусовим групуванням, то групована дисперсія є огрубленням справжньої вибіркової. Якщо розбиття при групуванні було рівномірним, можна ввести спеціальну поправку, яка дозволяє більш точно наблизити справжню дисперсію:

$$S_{corr}^2(X) = S_w^2(X) - \frac{h^2}{12},$$

де h — ширина інтервалу розбиття. Величина $h^2/12$ зветься поправкою Шеппарда (Sheppard's correction).

Навантажені статистики. Навантажені середні вигляду $\bar{X}_w = \sum_k w_k X_k$ природно використовувати не тільки для групованих даних. Такі суми часто виникають і у аналізі інших статистичних даних.

Приклад 1. Нормою прибутку підприємства називають прибуток, отриманий ним протягом року, ділений на обсяг капіталу, інвестованого у це підприємство. Нехай результатом спостережень n підприємств є розмір прибутку p_j та розмір інвестованого капіталу c_j для j-того обстеженого підприємства ($j = 1, \ldots, n$). Як визначити середню норму прибутку цих підприємств?

Можливі два варіанти. По-перше, можна підрахувати норми прибутку по кожному підприємству окремо:

$$r_j = \frac{p_j}{c_j}$$

і усереднити їх, отримавши

$$\bar{r} = \frac{1}{n} \sum_{j=1}^{n} r_j.$$

По-друге, можна знайти сумарний прибуток всіх обстежених підприємств одразу і розділити його на сумарний капітал цих підприємств:

$$\bar{r}_w = \frac{\sum_{j=1}^{n} p_j}{\sum_{j=1}^{n} c_j} = \sum_{j=1}^{n} w_j r_j,$$

де

$$w_j = \frac{c_j}{\sum_{i=1}^{n} c_i}.$$

Тобто у цьому випадку ми отримали навантажене середнє з ваговими коефіцієнтами, пропорційними капіталам підприємств. Це і зрозуміло — можна сподіватись, що чим більший капітал підприємства, тим більшим повинен бути його внесок у економіку, отже при підсумовуванні його варто враховувати з більшою вагою.

Який варіант середнього є більш "правильним" для цих даних? Відповідь залежить від задачі, яка стоїть перед дослідником. Якщо дослідження проводиться, наприклад, для міністерства фінансів, яке хоче оцінити можливий майбутній прибуток підприємств країни в залежності від вкладених інвестицій, то скоріше слід орієнтуватись на навантажене

середнє. Якщо дослідження виконується для фіскальної служби, яка має на меті виявити підприємства з аномальними значеннями норми прибутку, то, можливо, більш правильним орієнтиром нормального підприємства буде просте вибіркове середнє. А можливо, для визначення середнього положення норми прибутку у цьому випадку краще скористатись вибірковою медіаною.

Крім цієї та аналогічних ситуацій визначення середнього відношення двох спостережуваних величин, можливо багато інших задач, у яких природним буде застосування навантажених середніх. Для повного розуміння того, чому у цих задачах навантаження набуває певної форми, потрібно описати відповідні дані певними ймовірнісними моделями, які обговорюються пізніше. Тому тут ми лише побіжно згадаємо найбільш поширені варіанти навантажень.

Приклад 2. Нехай проводяться вимірювання однієї і тієї ж фізичної величини різними приладами. X_j — результат вимірювання j-тим приладом, $j = 1, \ldots, n$. Точність вимірювань різна у різних приладів. Дисперсія похибки[5] j-того приладу дорівнює σ_j^2. Тоді найбільш точною оцінкою справжнього значення вимірюваної величини дорівнює

$$\bar{X}_\sigma = \frac{1}{\sum_{j=1}^n 1/\sigma_j^2} \sum_{j=1}^n \frac{X_j}{\sigma_j^2} = \sum_{j=1}^n w_j X_j,$$

де

$$w_j = \frac{1/\sigma_j^2}{\sum_{i=1}^n 1/\sigma_i^2}.$$

Інтуїтивний зміст цієї формули зрозумілий: чим більша дисперсія похибки, тим менша точність відповідного вимірювання, тому спостереження, що мають більші дисперсії, включаються у сумарну оцінку з меншими коефіцієнтами.

Приклад 3. Нехай досліджувані об'єкти мають різні шанси потрапити до вибірки, причому ці шанси пов'язані з характеристикою, що досліджується. Такі вибірки звуться зміщеними.

Наприклад, об'єктами можуть бути риби, виловлені у ставку, а характеристикою — довжина рибини. Чим більшою є рибина, тим більше у неї шансів потрапити до рибальської сітки. Якщо метою дослідження

[5]Маємо на увазі дисперсію, вказану у паспорті приладу, яка характеризує точність вимірювань цим приладом, визначену при його сертифікації.

є оцінювання середньої довжини риб у ставку, то середнє довжин виловлених риб буде завищеною оцінкою цієї характеристики. Тобто оцінка по зміщеній вибірці є зміщеною.

Для виправлення цього зміщення використовують навантажені середні з ваговими коефіцієнтами, обернено пропорційними ймовірності того, що дане спостереження потрапить до вибірки. Такі вагові коефіцієнти називають коефіцієнтами Горвіца-Томпсона.

Крім навантажених середніх можуть використовуватись також інші навантажені статистики, такі як навантажена медіана або навантажена дисперсія. Формули для цих статистик використовуються такі ж, як наведено вище для групованих даних, але вагові коефіцієнти мають інший зміст.

Інколи змістовні вагові коефіцієнти у формулах для навантажених статистик не є нормованими, тобто їх сума не дорівнює 1. У такому випадку нормують саму статистику. Наприклад, якщо $\sum_{j=1}^{n} w_j \neq 1$, то навантажене вибіркове середнє слід рахувати за формулою

$$\bar{X} = \frac{1}{\sum_{j=1}^{n} X_j} \sum_{j=1}^{n} w_j X_j,$$

а навантажене геометричне середнє — за формулою

$$\mathrm{GM}(X) = \left(\prod_{j=1}^{n} (X_j)^{w_j} \right)^{1/\sum_{j=1}^{n} w_j}.$$

4.1.4 Обчислення описових статистик у R

Підрахунок більшості основних описових статистик у R реалізовано у вигляді функцій однотипної структури. Для менш поширених статистик часто можна написати простий вираз котрий їх обчислює. Зведення по цих функціях дано у таблиці 4.1.

У всіх цих функцій першим елементом `x` є вибірка, за якою рахується відповідна статистика. Цей параметр може бути числовим вектором або матрицею. В обох випадках результатом виконання функції одне число - значення відповідної статистики підраховане за всіма елементами `x`. Вийнятком з цього правила є функція `var`. Якщо її аргументом `x` є матриця, вона підраховує матрицю коваріацій для стовпчиків `x`.

Наприклад:

Статистика	Позначення	Функція
Вибіркове середнє	$\bar{X}$	`mean(x)`
Геометричне середнє	$\mathrm{GM}(X)$	`prod(x)^{(1/length(x))}`
Гармонійне середнє	$\mathrm{HM}(X)$	`1/mean(1/x)`
Зрізане середнє	$\mathrm{TM}_{2a}(X)$	`mean(x,trim=a)`
Медіана	$\mathrm{med}(X)$	`median(x)`
Виправлена вибіркова дисперсія	$S_0^2(X)$	`var(x)`
Середньоквадратичне відхилення	$S_0(X)$	`sd(x)`
Середнє абсолютне відхилення	$\mathrm{MAD}(X)$	`mad(x,mean(x),1)`
Середнє абсолютне відхилення	$\mathrm{MAD}_\mu(X)$	`mad(x,1)`
Інтерквартильний розмах	$\mathrm{IQ}(X)$	`quantile(x,0.75)-quantile(x,0.25)`

Таблиця 4.1: Функції для підрахунку описових статистик

```
> x=cbind(1:3,4:6)
> x

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

> mean(x)

[1] 3.5

> sd(x)

[1] 1.870829

> var(x)

     [,1] [,2]
[1,]    1    1
[2,]    1    1
```

У всіх розглянутих функцій є також логічний параметр-опція `na.rm`. Якщо вказати `na.rm=T`, то перед підрахунком статистики з вибірки будуть вилучатись всі пропущені значення (статистика рахується тільки за не

пропущеними). За умовчанням `na.rm=F`, у цьому випадку, за наявності пропущених значень у вибірці значенням функції теж буде `NA`.

Функція `mad()` має додатковий параметр `center`, що вказує функцію для підрахунку статистики середнього положення від якої відраховується середнє абсолютне відхилення. За умовчанням це медіана, але задавши, наприклад, `center=mean(x)` отримуємо середнє абсолютне відхилення від медіани. Крім того, у цій функції є параметр `constant` — константа, на яку домножається підраховане середнє абсолютне відхилення. За умовчанням, `constant=1.4826`. При використанні такого множника `mad(x)` буде консистентною оцінкою для середньоквадратичного відхилення вибірки з нормальним розподілом. Якщо потрібне справжнє значення $\mathrm{MAD}(X)$, слід задати `constant=1`.

У стандартній поставці `R` немає окремих функцій для обчислення навантажених статистик. Але більшість з них неважко записати, безпосередньо використовуючи формули для їх обчислення:

```
> x<-1:5 # вибірка
> w<-c(2,2,2,1,1) # вагові коефіцієнти
> #
> sum(x*w)/sum(w) # навантажене вибіркове середнє

[1] 2.625

> (prod(x^w))^(1/sum(w)) # навантажене гармонійне середнє

[1] 2.27597
```

Складніше запрограмувати навантажену медіану. Якщо вагові коефіцієнти приймають лише цілі значення, це можна зробити так:

```
> x<-1:5 # вибірка
> w<-c(2,2,2,1,1) # вагові коефіцієнти
> #
> median(rep(x,w)) # навантажена медіана

[1] 2.5
```

Тут функція `rep(x,w)` розмножила кожен елемент x_j вибірки `x` w_j разів. Після цього `median()` підрахувала медіану цієї розмноженої вибірки.

Можна сказати, що ми трактували нашу вибірку як груповану і по цій груповаій вибірці відновили початкову (із повторами).

Зрозуміло, що такий спосіб підрахунку навантаженої медіани є дуже не ефективним, особливо, коли вагові коефіцієнти великі. У пакеті `laeken` є функція `weightedMedian()` котра рахує навантажену медіану при довільних вагових коефіцієнтах.

Найбільш часто описові статистики використовуються коли потрібно порівняти багато вибірок однотипних даних. Якщо ці вибірки зібрані у матрицю, то виникає потреба підраховувати статистики окремо для кожного стовпчика (або рядочка) матриці. Це можна зробити, використовуючи функцію `apply()` так, як описано у підрозділі 2.1.5.

Наприклад, змінна `fmg` містить значення концентрацій формальдегіду (мг на м^3) у атмосферному повітрі виміряні на Бесарабській площі міста Києва у різні години доби (о першій, сьомій, тринадцятій і дев'ятнадцятій годинах) за період з 15 по 21 жовтня 2015 року (дані з сайту ЦГО України `http://www.cgo.kiev.ua/`). Рядок матриці відповідає одній добі спостережень, стовпчик — певній годині доби. Нас може цікавити наскільки міняються концентрації протягом доби і наскільки вони міняються при вимірюванні у певний час у різні дні спостережень. Вибравши на роль характеристики розкиду середньоквадратичне відхилення, підрахуємо його по кожному рядку і кожному стовпчику:

```
> # Концентрації формальдегіду по днях
> d15=c(0.005,0.008,0.010,0.005)
> d16=c(0.004,0.005,0.015,0.008)
> d17=c(0.004,0.010,0.012,0.009)
> d18=c(NA,NA,NA,NA)
> d19=c(0.008,0.011,0.014,0.015)
> d20=c(0.009,0.011,0.014,0.007)
> d21=c(0.007,0.009,NA,NA)
> # Створюємо матрицю концентрацій:
> fm=rbind(d15,d16,d17,d18,d19,d20,d21)
> colnames(fm)<-(c("t01","t07","t13","t19"))
> apply(fm,1,sd,na.rm=T)
       d15         d16         d17         d18         d19         d20
0.002449490 0.004966555 0.003403430          NA 0.003162278 0.002986079
       d21
0.001414214
```

```
> apply(fm,2,sd,na.rm=T)

        t01         t07         t13         t19
0.002136976 0.002280351 0.002000000 0.003768289
```

Бачимо, що за порядком величини розкиди однакові.

Відмітимо, що у функції `apply()` як параметр `fun` (тобто функція, яка буде застосована до рядків або стовпчиків матриці) можна використовувати не тільки ім'я функції, а і опис. Наприклад, якщо за даними `fm` потрібно підрахувати гармонійні середні по кожній добі спостережень, це можна зробити так:

```
apply(fm,1,function(x)(prod(x)^(1/length(x))))
```

Часто буває також, що всі дані для аналізу зібрані у одному фреймі, причому досліджувана характеристика є одною із змінних цього фрейму. Розбиття на окремі підвибірки потрібно зробити за іншими змінними-факторами, що характеризують приналежність досліджуваних об'єктів до різних груп. У таких ситуаціях зручно використовувати функцію `tapply()`. Вона призначена для застосування певної функції-статистики окремо до кожної підвибірки, заданої комбінацією певних факторів. Значенням функції є таблиця значень статистики для всіх можливих комбінацій факторів.

Приклад. У фреймі даних `ToothGrowth` містяться дані про експеримент по впливу різних дієт на швидкість росту зубів у свиней. Всього у фреймі 60 спостережень, кожне відповідає одній свині. Змінна `len` вказує довжину зубів, `sup` — харчову добавку, яку використовували для внесення у раціон свині вітаміну C (VC — хімічна аскорбінова кислота, OJ — помаранчовий сік), `dose` — щоденна доза вітаміну, яку отримувала свиня із цією добавкою (лише три варіанти доз: 0.5, 1 або 2 міліграми). Нас цікавить, як відрізняються середні значення та середньоквадратичні відхилення `len` при різних комбінаціях факторів `sup` і `dose`.

```
> # Таблиця вибіркових середніх:
> tapply(ToothGrowth$len,list(ToothGrowth$supp,ToothGrowth$dose),mean)

     0.5     1     2
OJ 13.23 22.70 26.06
VC  7.98 16.77 26.14

> # Таблиця середньоквадратичних відхилень:
> tapply(ToothGrowth$len,list(ToothGrowth$supp,ToothGrowth$dose),sd)
```

```
        0.5        1        2
OJ 4.459709 3.910953 2.655058
VC 2.746634 2.515309 4.797731
```

Розділ 5

Основні ймовірнісні розподіли

5.1 Загальні поняття та схема використання основних розподілів в R

У математичній статистиці дані прийнято розглядати як випадкові об'єкти — випадкові величини або вектори, процеси, поля, множини... Статистичні характеристики даних природно описувати у термінах ймовірнісних розподілів цих об'єктів.

Розподіл будь-якої випадкової величини ξ можна задати вказуючи функцію розподілу, тобто $F_\xi(x) = \mathsf{P}\{\xi \le x\}$. Якщо величина ξ є абсолютно неперервною, тобто існує така функція $f_\xi(x)$, що $F_\xi(x) = \int_{-\infty}^{x} f_\xi(t)dt$ при всіх $x \in \mathbb{R}$, то f_ξ, яка зветься щільністю розподілу, також однозначно задає розподіл.

Якщо розподіл є дискретним, тобто існує зліченний набір $T = \{t_1, t_2, \dots, \} \in \mathbb{R}$, такий, що $\mathsf{P}\{\xi \in T\} = 1$, то функцію $f_\xi(x) = \mathsf{P}\{\xi = x\}$ можна трактувати як щільність розподілу ξ відносно рахуючої міри. Цю функцію інколи також називають розподілом дискретної випадкової величини.

Квантилем $Q^\xi(\alpha)$ розподілу випадкової величини ξ рівня α називають найменше таке число x, для якого $F_\xi(x) \ge \alpha$. Якщо існує функція $F_\xi^{-1}(x)$ обернена до функції розподілу, то $Q^\xi(\alpha) = F_\xi^{-1}(\alpha)$.

Для опису розподілу даних та функцій від них (статистик) часто використовуються параметричні моделі, у яких функція розподілу вважається відомою з точністю до деяких параметрів. У R для ряду найбільш поширених параметричних моделей реалізовані функції, що обчислюють функцію розподілу, щільність, квантилі для заданого розподі-

Розподіл	Ім'я	Параметри
бета	beta	shape1, shape2
біноміальний	binom	size, prob
гамма	gamma	shape, rate
геометричний	geom	prob
гіпергеометричний	hyper	m, n, k
експоненційний	exp	rate
Коші	cauchy	location, scale
логістичний	logis	location scale
логнормальний	lnorm	meanlog, sdlog
негативний біноміальний	nbinom	size, prob
нормальний	norm	mean, sd
Пуассона	pois	lambda
рівномірний	unif	min, max
Вейбула	weibull	shape, scale
Вілкоксона	wilcox	m,n
χ^2	chisq	df
F-Фішера	f	df1, df2
T-Стьюдента	t	df

Таблица 5.1: Імена функцій для основних ймовірнісних розподілів

лу та генерують псевдовипадкову величину із заданим розподілом. Ці функції організовані за єдиною схемою. Ім'я функції утворюється з імені розподілу (див. табл. 5.1) та префіксу, який вказує, що обчислює дана функція. Префікси можуть бути:

p — обчислення функції розподілу (probability). Наприклад, `pnorm(1.96)` — функція стандартного нормального розподілу у точці 1.96;

d — обчислення щільності (density) розподілу (для абсолютно неперервних випадкових величин) або ймовірності попадання у точку (для дискретних): `dbinom(1,size=1,prob=0.5)` — ймовірність того, що випадкова величина з біноміальним розподілом дорівнює 1, якщо ймовірність успіху 0.5, а кількість випробувань — 1.

q — обчислення квантиля (quantile) заданого рівня: значенням функції `qnorm(c(0.025,0.975))` буде вектор квантилів рівня 0.025 і 0.975, тобто (-1.959964, 1.959964).

r — генерація псевдовипадкових чисел із заданим розподілом: `rnorm(100)`

генерує 100 псевдовипадкових значень, що моделюють вибірку з незалежних стандартних нормальних випадкових величин.

У функцій з префіксами p, d і q першим параметром є вектор значень аргументів, для яких треба обчислити відповідну функцію (ф.р., щільність, квантиль). У функцій з префіксом r (псевдовипадкових генераторів) перший аргумент — розмір вибірки, тобто кількість генерованих величин.

Наступні параметри є параметрами розподілу. Вони різні для різних розподілів (див. третій стовпчик таблиці 5.1) але однакові для всіх функцій, пов'язаних з даним розподілом. Наприклад, для нормального розподілу, параметри `mean` та `sd` вказують математичне сподівання та стандартне відхилення (корінь квадратний з дисперсії).

У всіх функцій з префіксом p і q є логічний параметр-опція `lower.tail`. Його значення за умовчанням — FALSE. Якщо задати `lower.tail=T` то p-функція буде замість функції розподілу обчислювати функцію виживання $\mathsf{P}\{\xi > x\} = 1 - F_\xi(x)$, а q-функція — верхній квантиль, тобто $Q^\xi(1-\alpha)$.

Якщо у функції p задати опцію `log.p=T`, то вони будуть обчислювати логарифм ф.р.:

```
> pnorm(-1.96,log.p=T)

[1] -3.688964

> log(pnorm(-1.96))

[1] -3.688964
```

(Насправді `pnorm(-1.96,log.p=T)` не обчислює спочатку ф.р., а потім її логарифм, а одразу шукає цей логарифм за спеціальним алгоритмом наближеного обчислення. Тому цей варіант працює швидше і дає точніший результат ніж `log(pnorm(-1.96))`, хоча для більшості застосувань різниця практично непомітна).

5.2 Генерація псевдовипадкових послідовностей

SecGenRand

У статистиці часто виникає потреба отримати послідовність чисел, які описуються певною ймовірнісною моделлю. У простішому випадку це

може бути послідовність незалежних, однаково розподілених випадкових величин. З точки зору класичної теорії ймовірностей, випадковість є властивістю не конкретної числової послідовності, а способу, у який ця послідовність була отримана. Наприклад, у сучасній фізиці вважається, що спонтанний розпад атомних ядер відбувається випадково, незалежно у різних ядрах зі сталою ймовірністю. Тому, якщо у ході вимірювань кількості розпадів у певному зразку речовини протягом 1 хвилини було зафіксовано послідовність спостережень 3, 1, 4, 1, 5, 9, 2, 6 — ця послідовність є випадковою. Ця сама послідовність, отримана комп'ютерною програмою при обчисленні знаків числа π — випадковою не є.

З цієї точки зору, всі послідовності чисел, які можна згенерувати на звичайному комп'ютері без використання яких-небудь зовнішніх джерел випадковості, не є випадковими. Але можна розглядати алгоритми, що генерують послідовності, які імітують випадковість, тобто мають основні властивості, притаманні послідовностям випадкових величин. Такі алгоритми і програми що їх реалізують називають генераторами (датчиками) псевдовипадкових чисел (pseudorandom numbers generators). Префікс псевдо- часто пропускають і кажуть про генерацію випадкових чисел. Це не є помилкою, якщо пам'ятати про імітаційний характер такої випадковості.

Останнім часом набула розвитку техніка генерування квазівипадкових чисел (quasirandom numbers) — послідовностей, що поєднують деякі риси випадкових з такими особливостями, яких справжні випадкові послідовності мати не можуть в принципі. Зокрема, такі числа використовуються при наближеному інтегруванні багатовимірних функцій за методом Монте-Карло (див. розділ 7.7 у книжці [5]). У даній книжці ця тематика не розглядається.

Як правило, генерація псевдовипадкових чисел починається із створення рівномірних чисел, тобто послідовності, яка імітує поведінку послідовності незалежних, однаково розподілених випадкових величин з рівномірним розподілом на $[0, 1]$. Потім, використовуючи ті чи інші перетворення цієї послідовності, отримують псевдовипадкові послідовності із заданим розподілом, наприклад, нормальні або такі, що утворюють ланцюг Маркова із заданими ймовірностями переходу.

Генерація рівномірних псевдовипадкових послідовностей має вже більш ніж 70-літню історію, тут відібрані найкращі генератори, які і реалізовані у базовому `R` . Намагатись самостійно покращити їх без глибокого знання відповідної теорії та власного досвіду у цій області не варто.

Але я включив у цю книжку елементарні відомості про таку генерацію, щоб читач мав змогу, по-перше, зрозуміти, як відбувається генерація у стандартних програмах, а по-друге, при бажанні створити свій власний генератор, якщо раптом виникне недовіра до стандартного. Ці відомості вміщені у п.5.2.1.

У R реалізовані також функції, що дозволяють отримати послідовності які імітують кратні вибірки з основними ймовірнісними розподілами, такими як нормальний, експоненційний, пуассонів, тощо. Але цих функцій може бути недостатньо, якщо вам потрібно згенерувати псевдовипадкову послідовність з яким-небудь менш поширеним розподілом, наприклад, з розподілом Парето. Тому розуміння загальних підходів до такої генерації є важливим елементом роботи статистика. З елементарними відомостями про це можна ознайомитись у п. 5.2.2.

Коротко про те, як ці техніки генерації реалізовані у R можна прочитати у п.5.2.3.

5.2.1 Генератори рівномірних псевдовипадкових чисел

SecGenUnif

Отже, рівномірні псевдовипадкові числа це числові послідовності, які відтворюють основні властивості послідовностей незалежних однаково розподілених випадкових величин з рівномірним розподілом на $[0, 1]$. Для створення таких послідовностей, як правило, використовують рекурсивну техніку. При цьому задаються деякі значення початкових елементів послідовності x_1, x_2,... x_k і функція $f(t_1, \ldots, t_k)$, що породжує наступний елемент послідовності. Після цього послідовність визначається як

$$x_{k+1} = f(x_1, \ldots, x_k),$$

$$x_{k+2} = f(x_2, \ldots, x_{k+1}),$$

...

$$x_n = f(x_{n-k}, \ldots, x_{n-1}),$$

...

У найпростішому випадку $k = 1$, тобто кожен наступний елемент послідовності визначається за попереднім:

$$x_n = f(x_{n-1}).$$

При виборі функції f у першу чергу керуються міркуваннями простоти ралізації та швидкості виконання. Найбільш поширена сім'я генераторів — лінійні конгруентні генератори у яких функція f будується з використанням лінійної залежності зі сталими коефіцієнтами. Розрізняють два типи генераторів: з цілочисловою та дійснозначною арифметикою.

У генераторі з дійснозначною арифметикою x_1 вибирають з інтервалу $(0,1)$ і послідовність породжується за правилом

$$x_n = \{ax_{n-1} + c\},\ n = 2, 3, \ldots,$$

де $\{x\}$ — дробова частина числа x. Тут a і c — фіксовані дійсні числа.

У генераторі з цілочисловою арифметикою спочатку будується допоміжна послідовність натуральних чисел I_1, I_2,... I_n,... Початкове число I_1 вибирають з інтервалу 1,...,$m-1$, послідовність формується за правилом

$$I_n = aI_{n-1} + c \mod m,\ n = 2, 3, \ldots$$

де a, c та m — натуральні числа. Якщо $c = 0$ генератор називають мультиплікативним. Послідовність дійсних чисел з інтервалу $[0,1]$ отримують з I_n діленням на m:

$$x_n = \frac{I_n}{m}.$$

Число a називають множником, c — приростом а m — модулем генератора.

У сучасних генераторах, як правило, використовують цілочислові схеми, оскільки правила округлення у дійснозначній арифметиці відрізняються на різних комп'ютерах. Тому один і той же дійснозначний генератор може на одному комп'ютері давати хорошу послідовність, а на іншому — погану. Цілочислова арифметика на всіх комп'ютерах реалізована однаково (якщо організувати обчислення без переповнень). З цієї точки зору цілочислові генератори є більш надійними.

Нехай I_j, $j = 1, 2, \ldots$ — послідовнscnm, згенерованf лінійним конгруентним генератором з цілочисловою арифметикою. Зрозуміло, що якщо, при деяких n і k, $I_n = I_{n-k}$, то для всіх $i = 1, 2, \ldots$ буде виконано $I_{n+i} = I_{n-k+i}$, тобто послідовність буде циклічно повторюватись. Найменше k при якому це буде виконуватись, називають періодом (або довжиною періоду) генератора. Очевидно, що циклічно повторювана послідовність не може вважатись випадковою, тому генератори не доцільно

використовувати для породження послідовностей з довжиною, більшою ніж період. Отже, хороший генератор мусить мати великий період.

Оскільки між 0 і $m-1$ є рівно m чисел, період цілочислового лінійного конгруентного генератора не може бути більшим ніж m. Відомі умови на параметри генератора, при яких він має найбільший період (тобто m):

Теорема 5.2.1 ***(Халла-Добелла)*** *Для того, щоб цілочисловий лінійний конгруентний генератор мав період m необхідно і достатньо, щоб виконувались умови:*

1. c і m взаємно прості.

2. Всі прості дільники m є дільниками $a-1$.

3. Якщо m кратне 4, то $a-1$ теж кратне 4.

Але вимога максимального періоду не єдина, що визначає псевдовипадкову послідовність як хорошу. Дійсно, послідовність 1, 2,... m має період m, але на випадкову вона не схожа. Тому для оцінки якості генератора потрібно проводити спеціальні "тести на випадковість". Такі тести, як правило, будують за звичайною схемою статистичних тестів для перевірки того, що певна послідовність даних відповідає обраній імовірнісній моделі. Ми зупинимось зараз лише на двох елементарних графічних способах перевірки якості генератора псевдовипадкових чисел.

Нехай $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ — послідовність незалежних, рівномірно на $[0,1]$ розподілених псевдовипадкових чисел. Емпіричною функцією розподілу даних $\mathbf{X}$ називають

$$\hat{F}_n(x) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{\xi_j < x\}.$$

Зрозуміло, що $\hat{F}_n(x)$ — це відносна частота інтервалу $(-\infty, x)$ у вибірці. За законом великих чисел, при великих n, $\hat{F}_n(x) \approx F(x)$, де $F(x)$ — функція розподілу для рівномірного розподілу на $[0,1]$, тобто

$$F^{U[0.1]}(x) = \mathsf{P}\{\xi_1 < x\} = \begin{cases} 0 & \text{при } x < 0 \\ x & \text{при } 0 \le x \le 1 \\ 1 & \text{при } x > 1 \end{cases}.$$

Для графічної перевірки якості генератора можна відобразити на одному графіку емпіричну функцію розподілу згенерованої псевдовипадкової

послідовності та $F^{U[0.1]}(x)$. Якщо вони будуть близькими одна до одної — генератор пройшов це випробування. Якщо помітно систематичне відхилення емпіричної функції від теоретичної — генератор не адекватний.

Наступний приклад демонструє, як працює лінійний конгруентний генератор з цілочисловою арифметикою і параметрами $a = 65539$, $c = 0$, $m = 2^{31}$ з початковим значенням $I_1 = 2^{15} + 2$. Кількість спостережень $n = 200$.

Цей генератор був досить популярним у 60-70-ті роки XX ст. під назвою RANDU, зокрема, використовувався як стандарт на комп'ютерах фірми IBM.

```
>   n<-200   # кількість чисел
>   a<-65539 # RANDU параметри
>   c0<-0     #
>   m<-2^31  #
>   I<-numeric(n) # цілочислова послідовність
>   I[1]<-2^15+2
>   for(i in 2:n){
+   I[i]<-(a*I[i-1]+c0)%% m
+   }
>   x<-I/m # псевдовипадкові числа
>   plot(1:n,x,cex=0.3) # рисуємо діаграму чисел
>   sx<-sort(x)
>   # рисуємо емпіричну функцію розподілу:
>   plot(sx,(1:n)/n,type="s",xlim=c(0,1),ylim=c(0,1))
>   # графік теоретичної функції розподілу:
>   abline(a=0,b=1,col="red")
```

Результати роботи відображений на рис. 5.1. Ліворуч — діаграма, у якій координати точок по горизонталі відповідають номеру псевдовипадкового числа, а по вертикалі — його значенню. Праворуч — емпірична функція розподілу, побудована за псевдовипадковою послідовністю. Рисунок ліворуч демонструє "випадкову" поведінку послідовності: не помітно яких-небудь закономірностей, що свідчили б про не випадковість. Рисунок ліворуч показує рівномірність розподілу — емпірична функція розподілу коливається навколо теоретичної. При збільшенні довжини послідовності відхилення емпіричної функції від теоретичної стають все менш помітними.

Можна вважати, що цей тест генератор RANDU пройшов.

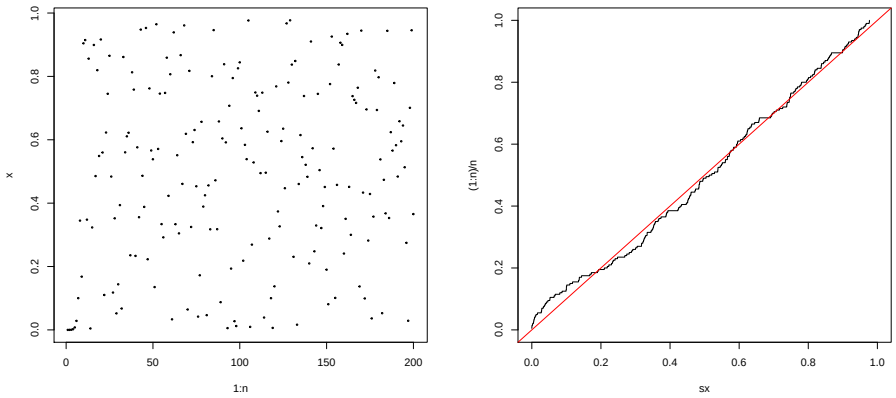

Рис. 5.1: Генератор RANDU: розкид та емпірична функція розподілу

Іще один важливий вид тестів — графічна перевірка залежності двох або трьох сусідніх елементів послідовності на графіку пар/трійок. Для того, щоб побачити залежності, будують точки на площині з координатами (x_j, x_{j+1}), $j = 1, \ldots, n-1$ або у тривимірному просторі — з координатами (x_j, x_{j+1}, x_{j+2}), $j = 1, n-2$. На відповідних діаграмах намагаються знайти закономірності, що відрізняють поведінку послідовонсті від справжньої випадкової. Для лінійних конгруентних генераторів такою закономірністю часто є розташування точок вздовж невеликої кількості прямих ліній на площині або площин — у тривимірному просторі. Зрозуміло, що така особливість генератора свідчить про не випадковість.

Продовжуючи попередній приклад, ці тести можна реалізувати так:

```
x1<-x[1:(n-2)]
x2<-x[2:(n-1)]
x3<-x[3:n]
library(rgl)
plot3d(x1,x2,x3)    # 3D-графіка
plot(x1,x3,cex=0.3) # точки на площині
```

Результати тестів — на рис. 5.2. На двовимірній діаграмі розсіювання не видно закономірностей, що характеризували б послідовність як не випадкову: точки розкидані хаотично і заповнюють квадрат з приблизно однаковою щільністю. Отже, цей тест пройдений.

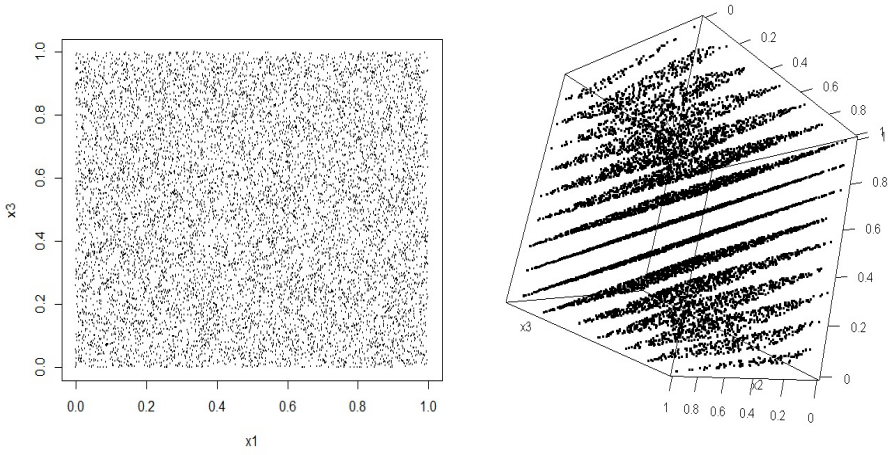

Рис. 5.2: Діаграми розсіювання пар та трійок для RANDU

На тривимірній картинці теж спочатку закономірності не були помітні, але після повороту вдалось отримати те, що зображено на рис. 5.2 праворуч: точки розташовані на кількох (приблизно 15) площинах всередині кубу. Зрозуміло, що така поведінка не відповідає уявленням про незалежні випадкові величини з рівномірним розподілом, отже цей тест генератор RANDU не пройшов. Саме тому його зараз не використовують для генерації псевдовипадкових чисел.

Насправді всі лінійні конгруентні генератори дають послідовності, що породжують тривимірні структури, подібні до виявлених нами у генератора RANDU. Але у хороших генераторів кількість площин, на яких розташовуються точки — велика і ці площини знаходяться поруч одна від одної, тому такі генератори проходять цей тест.

У книжці [5] як "мінімальний стандарт" рекомендовано використовувати генератор Парка та Мілера з $a = 7^5$, $c = 0$, $m = 2^{31} - 1$. Цей генератор проходить описані нами тести а також більшість тестів, які прийнято застосовувати до таких генераторів. Його період $2^{31} - 2 \approx 2.1 \times 10^9$. Це велике число, але для деяких застосувань воно може бути недостатнім.

Існують більш складні техніки генерації псевдовипадкових послідовностей, що мають значно більші періоди. Наприклад, у п.7.1 книги [5] розглядається техніка комбінування двох лінійних конгруентних генераторів з різними періодами, яка дозволяє отримати послідовність з періодом, не меншим ніж найменше спільне кратне комбінованих генераторів.

Іще один спосіб генерації псевдовипадкових чисел, що набув популярності останнім часом — генератори Фібоначчі із запізненням (lagged Fibonacci generator), у яких для породження чергового елемента послідовності використовується не один попередній елемент, а два, взяті з фіксованим запізненням. Наприклад, адитивний генератор Фібоначчі має вигляд

$$I_n = I_{n-k} + I_{n-l} \mod m,$$

де $k < l$ фіксовані числа (лаги). Для створення послідовності цим генератором потрібно задати не один, а l початкових елементів, після чого можна використовувати генеруючу формулу. Модуль m, як правило, вибирають ступенем двійки: $m = 2^b$. При правильному виборі лагів, цей генератор дозволяє отримати період $2^{b-1}(2^l - 1)$. Прикладами "хороших" лагів є $k = 7$, $l = 10$ або $k = 5$, $l = 17$.

Подальші відомості про генератори рівномірних послідовностей можна знайти у книзі Д. Кнута [2].

5.2.2 Генерація псевдовипадкових чисел із заданим розподілом

SecGenOther

Якщо деяким генератором створена псевдовипадкова послідовність з рівномірним розподілом, то отримати з неї послідовність, що імітує незалежні випадкові величини з іншим розподілом можна використовуючи різні перетворення. При цьому, як правило, те, що початкова послідовність лише імітує випадковість — ігнорується. Тобто у цьому підрозділі ми будемо трактувати початкову послідовність $\eta_1, \ldots \eta_n, \ldots$ як послідовність незалежних однаково розподілених випадкових величин з певним розподілом G. Цей розподіл назвемо початковим. (Поки що ми вміємо генерувати лише послідовності з рівномірним розподілом, але далі нам інколи буде зручно використовувати як початковий який-небудь інший розподіл).

Завдання полягає в тому, щоб побудувати послідовність $\eta_1, \ldots, \eta_n, \ldots$ незалежних випадкових величин із заданим розподілом F. Цей розподіл називають цільовим. Методи генерації таких послідовностей розрізняються в залежності від того, в якій формі заданий цільовий розподіл.

Квантильне перетворення.

Нехай задана функція розподілу для цільового розподілу $F(x) = \mathsf{P}\{\eta_1 < x\}$, причому $F(x)$ є неперервною і строго зростаючою там, де вона не дорівнює 0 або 1. Розглянемо випадкову величину $\eta = F^{-1}(\xi)$, де випадкова величина ξ рівномірно розподілена на [0,1], F^{-1} — функція, обернена до F.

Легко бачити, що функція розподілу η

$$F_\eta(x) = \mathsf{P}\{\eta < x\} = \mathsf{P}\{F^{-1}(\xi) < x\} = \mathsf{P}\{\xi < F(x)\} = F(x),$$

тобто η якраз і має цільовий розподіл.

Отже, отримати випадкову послідовність з ф.р. F можна, застосувавши перетворення $x \to F^{-1}(x)$ до кожного елемента рівномірної початкової послідовності ξ_j окремо: $\eta_j = F^{-1}(\xi_j)$. Оскільки випадкові величини початкової послідовності були незалежними між собою, незалежними будуть і отримані ξ_j.

Це перетворення називають квантильним, тому що $F^{-1}(\alpha) = Q^F(\alpha)$ — квантиль рівня α для розподілу F.

Приклад 1. Нехай потрібно згенерувати послідовність незалежних, однаково розподілених випадкових величин з експоненційним розподілом. Функція розподілу — $F_\lambda(x) = 1 - e^{-\lambda x}$. Маємо $F^{-1}(y) = -\log(1-y)/\lambda$. Якщо η — рівномірно розподілена на $[0,1]$, то і $1-\eta$ теж. Тому з рівномірної початкової послідовності $\eta_1,\ldots,\eta_n$ цільову експоненційну послідовність можна отримати перетворенням

$$\xi_j = -\frac{\log \eta_j}{\lambda}.$$

У R генерація n експоненційних псевдовипадкових величин може виглядати так:

```
> n<-100    # кількість спостережень
> lambda=0.5 # інтенсивність exp розподілу
> a<-7^5
> c0<-0
> m<-2^31-1
> y<-numeric(n)
> y[1]<-1000
> for(i in 2:n){
```

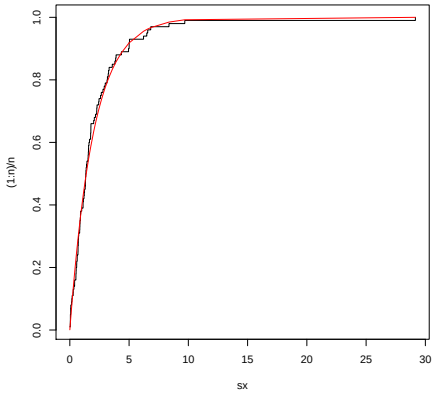

Рис. 5.3: Емпірична функція розподілу для експоненційного генератора випадкових чисел

```
+  y[i]<-(a*y[i-1]+c0)%% m
+  }
>  y<-y/m             # рівномірна послідовність
>  x<--log(y)/lambda # квантильне перетворення
>  #
>  # рисуємо емпіричну функцію розподілу:
>  sx<-sort(x)
>  plot(sx,(1:n)/n,type="s")
>  # графік теоретичної функції розподілу:
>  lines(sx,pexp(sx,rate=lambda),col="red")
```

Тут ми скористались генератором Парка і Міллера для отримання рівномірної послідовності `y` а потім застосували квантильне перетворення, щоб отримати цільову послідовність `x`. Графік її емпіричної функції розподілу у порівнянні з відповідною теоретичною функцією — на рис. 5.3.

Метод проріджування.

Квантильне перетворення дозволяє отримати незалежні випадкові величини з будь-яким розподілом. Але для цього потрібна функція, що знаходить квантилі цільового розподілу. Часто такі функції важко записати у явному вигляді а чисельний підрахунок квантилі становить самостійну задачу.

Метод проріджування дозволяє генерувати послідовності із заданим розподілом використовуючи для цього не квантилі, а щільності розподілу. Пояснимо ідею цього методу.

Нехай випадкова величина η має щільність розподілу g, а нам потрібна випадкова величина з щільністю f. Припустимо, що для всіх x $f(x) \le Cg(x)$ для деякого фіксованого числа $0 < C < \infty$. Введемо іще одну випадкову величину u, що має рівномірний розподіл на $[0,1]$ і є незалежною від η.

Підрахуємо умовну ймовірність

$$\mathsf{P}\left\{\eta < x \mid u < \frac{f(\eta)}{Cg(\eta)}\right\} = \frac{\mathsf{P}\{\eta < x,\ u < \frac{f(\eta)}{Cg(\eta)}\}}{\mathsf{P}\{u < \frac{f(\eta)}{Cg(\eta)}\}}.$$

Для чисельника маємо

$$\mathsf{P}\{\eta < x,\ u < \frac{f(\eta)}{Cg(\eta)}\} = \int_{-\infty}^{x} \int_{0}^{f(y)/(Cg(y))} dtg(y)dy = \frac{1}{C}\int_{-\infty}^{x} f(y)dy.$$

Аналогічно для знаменника

$$\mathsf{P}\{u < \frac{f(\eta)}{Cg(\eta)}\} = \frac{1}{C}.$$

Отже функція розподілу для розподілу η при умові $u < \frac{f(\eta)}{Cg(\eta)}$, дорівнює

$$\mathsf{P}\left\{\eta < x \mid u < \frac{f(\eta)}{Cg(\eta)}\right\} = \int_{-\infty}^{x} f(y)dy,$$

тобто це як раз ф.р. цільового розподліу зі щільністю f.

Ідея методу проріджування полягає в тому, щоб згенерувати послідовність пар $(\eta_1, u_1), (\eta_2, u_2), \ldots$, де η_j мають щільність g, u_j — рівномірні на [0,1] і всі в.в. незалежні в сукупності, а потім відібрати з елементів цієї послідовності ті, які задовольняють умову $u_j < f(\eta_j)/(Cg(\eta_j))$. Послідовність, створена відібраними η_j буде мати цільовий розподіл.

Приклад 2. Розглянемо задачу генерації послідовності з півнормальним розподілом з параметром $\sigma = 1$. Нагадаємо, що це розподіл випадкової величини $|\zeta|$, де ζ — стандартна гауссова випадкова величина. Його функція розподілу $F(x) = \mathsf{P}\{|\zeta| < x\} = 2\Phi(x) - 1$ при $x > 0$ і 0 при $x \le 0$.

Щільність розподілу —

$$f(x) = \begin{cases} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) & \text{при } x \geq 0, \\ 0 & \text{при } x < 0. \end{cases}$$

Щільність цільового розподілу записується у явному вигляді, а квантилі — ні. Тому природно скористатись для генерації методом проріджування. Оскільки $f(x) > 0$ для всіх додатних x, рівномірний розподіл не підходить як початковий. Але можна взяти як початкові експоненційно розподілені випадкові величини з інтенсивністю $\lambda = 1$. Щільність цього розподілу на додатній півосі — $g(x) = \exp(-x)$.

Легко бачити, що $f(x) \leq Cg(x)$ для $C = \sqrt{2e/\pi}$ і

$$\frac{f(x)}{Cg(x)} = \exp\left(-\frac{(x-1)^2}{2}\right).$$

Для генерації експоненційно розподіленої послідовності використаємо квантильне перетворення, як у прикладі 1. Оформимо знаходження чергового елемента псевдовипадкової послідовності у вигляді окремої функції. У скрипті, що наведений нижче, `rand()` — функція, яка генерує одне чергове рівномірне [0,1] число. (При цьому відповідне значення цілочислової послідовності `I` записується у глобальну змінну за допомогою глобального привласнення `I<<-` всередині тіла функції (див. п. 2.3.1). Функція, що генерує півнормальне число зветься `rhnorm`.

```
> n<-1000    # кількість спостережень
> a<-7^5     #   параметри генератора
> m<-2^31-1  #   Парка і Мілера
> I<-500     # початкове значення для генератора
> #
> # генератор рівномірної послідовності:
> rand<-function(){I<<-(a*I)%%m; I/m}
> #
> # генератор півнормальної послідовності:
> rhnorm<-function()
+ {
+ repeat{
+ u<-rand()
+ x<--log(rand()) # x - експоненційне
```

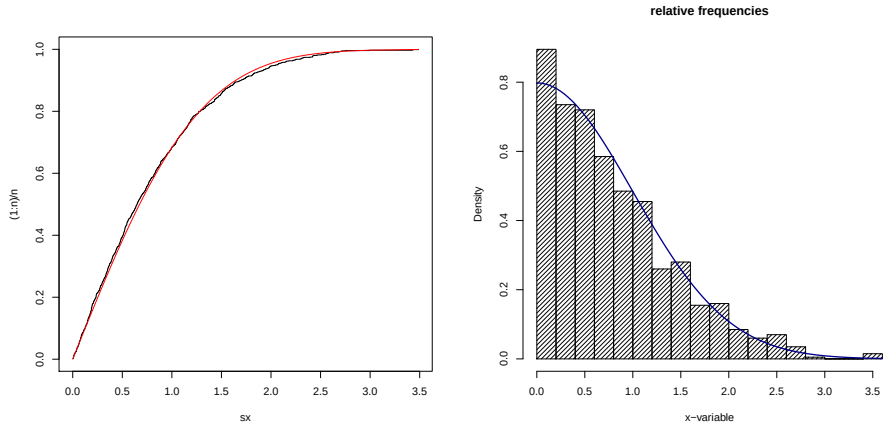

Рис. 5.4: Емпірична функція розподілу для експоненційного генератора випадкових чисел

```
+ if(u<exp(-0.5*(x-1)^2))return(x)
+ }
+ }
> # генеруємо півнормальну послідовність:
> x<-numeric(n)
> for(i in 1:n){x[i]=rhnorm()}
> # рисуємо графік емпіричної функції розподілу
> sx<-sort(x)
> plot(sx,(1:n)/n,type="s")
> # графік теоретичної функції розподілу:
> lines(sx,2*pnorm(sx)-1,col="red")
> # рисуємо гістограму відносних частот:
> hist(x, density=20, breaks=20, prob=TRUE,
+ xlab="x-variable",
+ main="relative frequencies")
> # рисуємо графік півнормальної щільності:
> curve(2*dnorm(x),
+ col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

Результати графічної перевірки якості генерації зображені ра рисунку 5.4. Тут ліворуч емпірична функція розподілу порівнюється з теоретич-

ною, а праворуч — гістограма відносних частот[1] згенерованої послідовності з щільністю півнормального розподілу (синя крива).

Як бачимо, щільність та функція розподілу півнормального розподілу добре відтворюються нашим генератором.

5.2.3 Випадкові числа в R

SecGenR

У базовому R реалізовані генератори псевдовипадкових послідовностей з основними ймовірнісними розподілами, вказаними у таблиці 5.1. Назви всіх цих функцій починаються з літери `r`, після чого йде назва розподілу: `rnorm()` генерує нормальні послідовності, `rexp()` — експоненційні і т.п.

Першим параметром всіх цих функцій є кількість елементів послідовності. Після цього параметра можна вказувати параметри розподілу. Наприклад,

`rnorm(10)` — генерує вектор з 10 псевдовипадкових стандартних нормальних чисел;

`rnorm(5, mean=1,sd=0.5)` — вектор з 5 нормальних чисел з математичним сподіванням 1 та дисперсією 0.25;

`rexp(1,rate=0.5)` одне число з експоненційним розподілом з інтенсивністю $\lambda = 0.5$.

Генерація псевдовипадкових чисел у стандартних функціях базового R організована за схемою подібною до прикладу 2 з п. 5.2.2. Використовується одна цілочислова послідовність, на основі якої будуються значення всіх псевдовипадкових чисел, які генеруються під час сеансу роботи з R . Чергове значення цілочислової послідовності зберігається у глобальній зімінній і змінюється при виконанні кожної функції-генератора.

Початкове значення цілочислової послідовності зветься seed — зернина. Ця зернина за умовчанням вибирається на початку сеансу роботи з R за останніми цифрами часу, який на цей момент показує годинник комп'ютера. Таким чином, кожного разу, коли ви запускаєте R , генерується нова послідовність псевдовипадкових чисел.

Це зручно, якщо ви перевіряєте статистичні особливості ваших алгоритмів: кожна нова перевірка відбувається на нових даних. Але на етапі відлагоджування програми, коли вам треба пересвідчитись, що її робота відповідає теоретичному алгоритму і виловити невідповідності, така генерація створює незручності. Помилка програми, яка була помітною на

[1] Про гістограму як оцінку щільності див. п.6.1.

одній послідовності, може загубитись при повторному тестуванні. Щоб усунути цей ефект бажано при відладці кожного разу запускати програму на одній і тій же псевдовипадковій послідовності. Це можна зробити, зафіксувавши зернину.

Вибір зернини робить функція `set.seed()`. Як параметр цієї функції можна вказати будь-яке ціле додатне число. За цим числом буде обрана зернина. Далі у цій книжці при використанні генераторів псевдовипадкових чисел зернина, як правило, фіксується. Це зроблено для того, щоб опис результатів у книжці відповідав тому, що видає скрипт. При самостійній роботі з скриптами з цієї книжки фіксувати зернину не потрібно, якщо ви хочете подивитись на випадковий розкид результатів.

Розділ 6

Методи графічного аналізу одновимірних даних

6.1 Гістограми

sect:Histogram

Гістограма є найбільш популярним способом графічного відображення розподілу числових даних. Розрізняють гістограми абсолютних та відносних частот.

Нехай спостерігаються значення змінної X у n об'єктів, рівні $(X_1, \ldots, X_n)$. Задамо деякий інтервал $[a, b]$, на якому розміщені всі спостережувані значення. Розіб'ємо цей інтервал на K підінтервалів $A_1, \ldots A_K$ однакової ширини $h = (b-a)/K$. Інтервали A_i, $i = 2, \ldots, K$ визначаються як $A_i = (t_{i-1}, t_i]$, де $t_i = a + ih$, $A_1 = [t_1, t_2]$.

Позначимо $n_i = \sum_{j=1}^{n} \mathbb{1}\{X_j \in A_i\}$ — кількість спостережуваних значень, що потрапили на інтервал A_i. Величину n_i звуть абсолютною частотою (absolute frequency, count) інтервалу A_i у вибірці X. Величину $\nu_i = n_i/n$ звуть відносною частотою (relative frequency). [1]

Гістограма абсолютних частот будується так. На горизонтальній осі відкладаються інтервали A_i і над кожним інтервалом будується стовпчик

[1] Зауважимо, що при нашому виборі відкритих зліва інтервалів A_i, спостереження, яке опинилось на межі двох інтервалів, потрапляє до інтервалу, що лежить ліворуч. (Так реалізований підрахунок частот для гістограм в `R`). Інколи навпаки, задають інтервали розбиття, відкриті зправа. Іще один можливий варіант, коли спостереження, що лежить на межі двох інтервалів враховується у частотах обох, але з вагою $1/2$. При великій кількості спостережень без повторень ці відмінності не грають ролі, але у деяких випадках можуть бути важливими для розуміння поведінки гістограми.

висоти n_i (див. ліву частину рис. 6.1).

У гістограмі відносних частот висота стовпчика визначається як $f_i = \nu_i/h = n_i/(nh)$. Таким чином, на рисунку гістограма відносних частот відрізняється від гістограми абсолютних лише масштабом по вертикалі (див. рис. 6.1). Нормуючий множник $1/(nh)$ для гістограми відносних частот обраний так, щоб її можна було використовувати як оцінку для щільності розподілу вибірки.

Дійсно, нехай $X = (X_1, \ldots, X_n)$ — вибірка з незалежних однаково розподілених випадкових величин, що мають щільність розподілу f. За законом великих чисел, при великому обсязі вибірки n,

$$\nu_i = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{X_j \in [t_{i-1}, t_i)\} \approx \mathsf{P}\{X_1 \in [t_{i-1}, t_i)\} = \int_{t_{i-1}}^{t_i} f(t)dt.$$

Якщо $x \in [t_{i-1}, t_i)$, f — гладенька функція і h маленьке, то $\int_{t_{i-1}}^{t_i} f(t)dt \approx f(x)h$. Отже $f_i \approx f(x)$, тобто функція

$$\hat{f}(x) = \begin{cases} f_1 & \text{якщо } x \in A_1 \\ f_2 & \text{якщо } x \in A_2 \\ \ldots & \\ f_K & \text{якщо } x \in A_K \\ 0 & \text{якщо } x \in [a, b] \end{cases}$$

є хорошим наближенням для $f(x)$. Гістограму відносних частот можна розглядати як графік цієї функції, а саму $\hat{f}(x)$ називають гістограмною оцінкою щільності розподілу.

Таким чином, якщо гістограму рисують щоб побачити щільність розподілу даних, доцільно використовувати саме гістограму відносних частот. В той же час, певні переваги має гістограма абсолютних частот: по висоті її стовпчиків одразу можна побачити скільки спостережень потрапило на той чи інший інтервал розбиття.

У `R` для рисування гістограм використовується стандартна функція `hist(x, ...)`. Перелічимо деякі параметри/опції цієї функції:

`x` — набір даних (вибірка) за яким будується гістограма.

`breaks` — параметр, що контролює вибір точок розбиття. Якщо він не заданий, то за умовчанням кількість точок розбитття обирається за

формулою Стургеса: $K = \lfloor \log_2 n + 1 \rfloor$, де n — кількість елементів `x`. Якщо `breaks` це одне число, його використовують як кількість інтервалів розбиття. При цьому як кінцеві точки всього інтервалу, на якому будується гістограма, беруть `min(x)`, `max(x)`. Якщо `breaks` — числовий вектор, його розглядають як набір точок розбиття $t_0 < t_1 < \cdots < t_K$.

`probability` — логічна опція, за умовчанням — `FALSE`. Якщо вона дорівнює `TRUE`, будується гістограма відносних частот, інакше — абсолютних.

`right` — логічна, якщо вона TRUE, то інтервали розбиття вважаються замкненими з права, відкритими зліва.

`density, angle, col, border` — параметри, що контролюють штриховку та колір прямокутників гістограми так само, як у функції `rect()`.

`main, xlab, ylab` — параметри, що задають основну назву та назви осей гістограми.

`plot` — якщо цей параметр зробити `FALSE`, гістограма відображатись не буде. При цьому параметри гістограми (інтервали розбиття та висоти стовпчиків) розраховуються і значенням функції `hist` є об'єкт, що містить ці параметри. Його можна зберегти для подальшого використання. (Скажімо, для відображення пізніше на іншому рисунку).

Приклад. У файлі `tips.csv` знаходяться дані про чайові, які отримував один офіціант ресторану у США протягом двох з половиною місяців роботи у 1990р. Розмір чайових отриманий за кожне обслуговування записаний у змінній `tip`, змінна `sex` вказує стать особи, що оплачувала рахунок ("F— жінка, "M— чоловік). Щоб отримати гістограми розміру чайових, виконаємо наступні команди:

```
> z<-read.csv("c:/rem/rstat/data/tips.csv")
> hist(z$tip,main="Absolute frequencies",xlab="tip")
> hist(z$tip,probability=T,main="Relative frequencies",xlab="tip")
```

Спочатку ми прочитали дані за допомогою функції `read.csv` (тут `c:/rem/rstat/data/` — каталог де знаходиться файл `tips.csv` на моєму комп'ютері). Потім ми вивели гістограму абсолютних частот і гістограму відносних частот.

Результат виконання зображений на рис. 6.1. З цього рисунку можна зробити висновок, що щільність розподілу розміру чайових є монотонно спадною. Зсунемо початкову точку гістограми [2] на 1/2 (рис. 6.2 ліворуч).

[2] origin, тобто лівий кінець інтервалу, на якому побудована гістограма

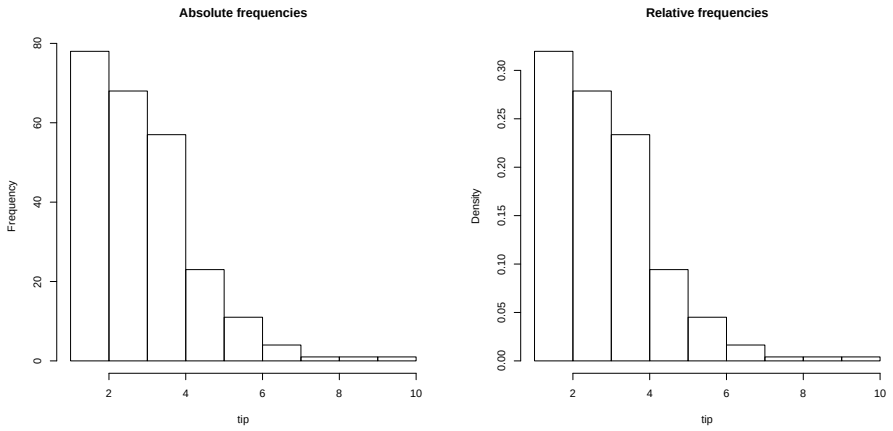

Рис. 6.1: Гістограми абсолютних та відносних частот

Тепер рисунок виглядає так, наче щільність спочатку зростає, а потім починає спадати.

Зменшимо ширину інтервалу розбиття — покладемо $h = 0.125$ — отримуємо картинку на рис. 6.2 праворуч. Команди, якими це було зроблено мають наступний вигляд:

```
> z<-read.csv("c:/rem/rstat/data/tips.csv")
> hist(z$tip,main="Origin shift",xlab="tip",breaks=(1:11)-0.5)
> hist(z$tip,main="Bin length changed",xlab="tip",breaks=(1:88)*0.125)
```

Якщо уважно придивитись до останнього рисунку, то можна побачити, що піки на гістограмі відповідають цілим розмірам чайових (2, 3, 4, 5 доларів) а також цілим значенням плюс півдолара. Крім того, праворуч від основної маси спостережень розташовані окремі невисокі стовпчики, що відповідають аномально великим чайовим. Ці спостереження легко пояснити з соціально-психологічних міркувань: людина може залишити "на чай" дрібні монети здачі, або дати гроші з свого гаманця. У другому випадку, як правило, залишають круглу суму. Більшість людей дотримуються загальноприйнятого розміру чайових, але дехто часом виявляє аномальну щедрість. Таким чином, у даному випадку не можна казати про якусь спільну щільність розподілу даних, що описує всі спостереження. Тим не менше, гістограма абсолютних частот дає можливість візуально проаналізувати такі дані і зробити певні висновки про їх розподіл.

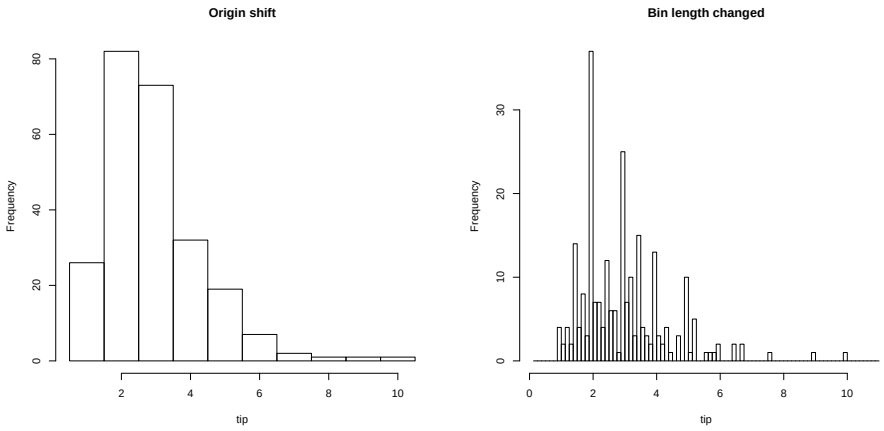

Рис. 6.2: Гістограми для tip: початкова точка та ширина інтервалу

Слід відмітити, що при зменшенні ширини інтервалу розбиття h, розкид стовпчиків гістограми зростає і тоді, коли дані являють собою кратну вибірку з розподілу, що має гладеньку щільність. Це легко зрозуміти: відносна частота інтервалу у вибірці наближається до ймовірності попадання у цей інтервал лише при великій кількості спостережень. Але, якщо інтервал малий, то мала і ймовірність потрапити на нього, отже на нього попаде мало спостережень і його частота буде помітно коливатись навколо ймовірності. Як це виглядає видно у наступному прикладі (рис. 6.3):

```
> z<-rnorm(200)
> hist(z,10)
> hist(z,50)
```

Ліворуч гістограма побудована з 10-ма широкими інтервалами, праворуч — з 50-ма вузенькими. Як і у попередньому прикладі, звуження інтервалів привело до появи піків та стовпчиків, розміщених окремо від основної маси спостережень. Але у розміщенні піків не помітно якої-небудь закономірності, а окремі стовпчики знаходяться досить близько від інших. Висоти всіх стовпчиків невеликі, тобто спостережень недостатньо для надійної оцінки щільності на кожному інтервалі. Тому ці ефекти природно трактувати, як випадкові. У даному прикладі ми знаємо, що вони дійсно є випадковими, оскільки спостереження `z` були створені генератором псевдовипадкових чисел зі стандартним нормальним розподілом.

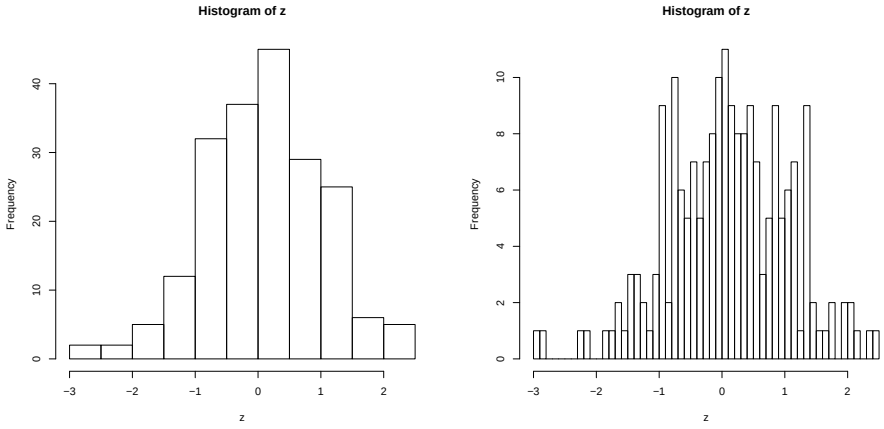

Рис. 6.3: Гістограми нормального розподілу

Але у загальному випадку відрізнити випадкові ефекти від значущих особливостей на гістограмі може бути непросто.

6.2 Графічна перевірка узгодженості розподілу. P-P та Q-Q діаграми

Одне з найбільш поширених застосувань гістограми — візуальне визначення типу розподілу та перевірка узгодженості даних з цим розподілом. Як ми з'ясували у попередньому підрозділі, гістограма відносних частот є оцінкою щільності розподілу. Зобразивши таку гістограму разом з теоретичною щільністю на одному рисунку, можна побачити, наскільки теоретична модель відповідає реальним даним.

Наприклад, у наборі даних `airquality` містяться дані щоденних вимірювань метеорологічної станції у Нью-Йорку з травня по вересень 1973р. Зокрема, змінна `airquality$Wind` вказує силу вітру у відповідний день. Ми хочемо перевірити, чи є розподіл цієї характеристики нормальним. Наведемо два варіанти програми відображення відповідної гістограми та щільності розподілу:

```
> # 1. гістограма відносних частот.
> #
> g = airquality$Wind
```

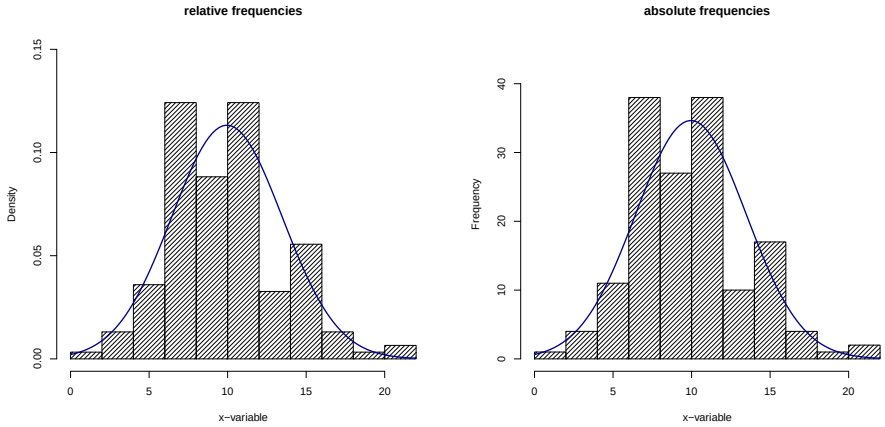

Рис. 6.4: Гістограми з графіком щільності

```
> m<-mean(g)
> std<-sqrt(var(g))
> hist(g, density=20, breaks=10, prob=TRUE,
+     xlab="x-variable", ylim=c(0, 0.15),
+     main="relative frequencies")
> curve(dnorm(x, mean=m, sd=std),
+      col="darkblue", lwd=2, add=TRUE, yaxt="n")
> #
> # 2. гістограма абсолютних частот
> #
> hi<-hist(g, density=20, breaks=10,
+     xlab="x-variable", ylim=c(0, 45),
+     main="absolute frequencies")
> curve(dnorm(x, mean=m, sd=std)*length(g)*(hi$breaks[2]-hi$breaks[1]),
+      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

У першому варіанті будується гістограма відносних частот (параметр `prob=TRUE`) і нормальна щільність, параметри якої оцінюються відповідно середнім та коренем з вибіркової дисперсії змінної `g`. Результат зображено на рис. 6.4 ліворуч. Як бачимо, посередині гістограми є провал там, де мав бути пік щільності. Чи можна вважати його випадковим, чи це дійсно відхилення від нормальності розподілу даних сили вітру?

За гістограмою відносних частот вирішити це неможливо. На гісто-

грамі абсолютних частот можна побачити, скільки спостережень припало на цей провал, але масштаб цієї гістограми не відповідає масштабу графіку щільності. Тому у другому варіанті (праворуч на рис. 6.4) виводиться графік щільності, помноженої на нормуючий множник nh, де n — обсяг вибірки, h — ширина підінтервалу розбиття. Щоб правильно визначити цей інтервал, ми зберігли значення результату функції `hist()` у змінній `hi`. Цей результат є об'єктом класу `histogram` і має атрибут `hi$breaks`, у якому містяться значення точок розбиття для побудованої гістограми. Різниця між сусідніми точками якраз і дорівнює h.

З гістограми абсолютних частот на рис. 6.4) видно, що кількість спостережень, які припадають на інтервал між двома піками становить близько 25, а кожному піку відповідає близько 40 спостережень. Це великі кількості даних і помітна відмінність між піками та провалом. Навряд чи вона викликана випадковим відхиленням. Скоріше, така гістограма свідчить про те, що розподіл даних не є нормальним.

Перевірка розподілу даних на основі гістограм зручна тим, що за формою гістограми часто можна вгадати розподіл: гістограму, що відповідає нормальному розподілу не зплутаєш із гісторгамою експоненційно розподілених даних. Але у гістограм є і незручності: невірно обравши ширину інтервалів розбиття або початок діапазону гістограми, можна отримати невдалий результат.

Тому поруч з гістограмами використовуються інші техніки графічної перевірки того, наскільки розподіл даних узгоджується з певною теоретичною моделлю: P-P (ймовірність проти ймовірності) та Q-Q (квантиль проти квантиля) діаграми. Ці діаграми побудовані на порівнянні емпіричної функції розподілу або емпіричних квантилів з відповідними характеристиками теоретичної моделі. Вони не потребують задання додаткових параметрів налаштування, подібних до ширини інтервалу розбиття для гістограми. Але їх недоліком є те, що теоретичний розподіл потрібно визначити наперед: за формою діаграми його вгадувати не можна.

Почнемо з розгляду P-P діаграм.

Нехай $X = (X_1, \ldots, X_n)$ — набір даних. Дослідник трактує X як кратну вибірку і хоче перевірити гіпотезу H_0 про те, що X_j мають функцію розподілу F. Якщо ця гіпотеза є вірною, то для будь-якого $x \in \mathbb{R}$, ем-

пірична функція розподілу вибірки $\hat{F}_n(x)$ є близькою до F:

$$\hat{F}_n(x) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{X_j \le x\} \approx F(x)$$

при великих обсягах вибірки.

Підставимо у $\hat{F}_n(x)$ та $F(x)$ вибіркові значення X_j, $j = 1, \ldots, n$ і зобразимо на рисунку точки з координатами $(F(X_j), \hat{F}_n(X_j),)$. Це і є P-P діаграма. Якщо гіпотеза H_0 є вірною. ордината та абсциса кожної точки повинні бути близькми одна до одної, отже точки мають вишикуватись поблизу від бісектриси першого координатного кута, як це зображено на рис. 6.5 ліворуч. Якщо це не так, гіпотезу H_0 слід відхилити. Рисунок 6.5 праворуч ілюструє ситуацію, коли для підгонки розподілу даних була обрана функція розподілу з невірною (завищеною) дисперсією.

Припустимо, що всі значення X_j у вибірці є різними і впорядкуємо їх у порядку зростання, отримавши варіаційний ряд: $X_{[1]} < X_{[2]} < \cdots < X_{[n]}$. Тоді $\hat{F}_n(X_{[j]}) = j/n$, отже P-P діаграма складається з точок $(F(X_{[j]}), j/n)$, $j = 1, \ldots, n$.

У `R` P-P діаграму, наприклад, для стандартного нормального розподілу, можна зобразити наступним чином:

```
> # Генеруємо дані для прикладу
> n<-100
> x<-rnorm(n)
> y<-rnorm(n,sd=3)
> # Рисуємо P-P для x з стандартним нормальним розподілом
> plot(pnorm(sort(x)),(1:length(x))/length(x),asp=1,
+     ylab="Empirical P",
+     xlab="Theoretical P")
> # Виводимо бісектрису координатного кута
> abline(0,1,col=2)
> # P-P для y з стандартним нормальним розподілом
> plot(pnorm(sort(y)),(1:length(y))/length(y),asp=1,
+     ylab="Empirical P",
+     xlab="Theoretical P")
> abline(0,1,col=2)
```

(Тут у `plot()` опції `xlab`, `ylab` вказують написи при осях координат, опція `asp=1` забезпечує однаковий масштаб по вертикалі та горизонталі).

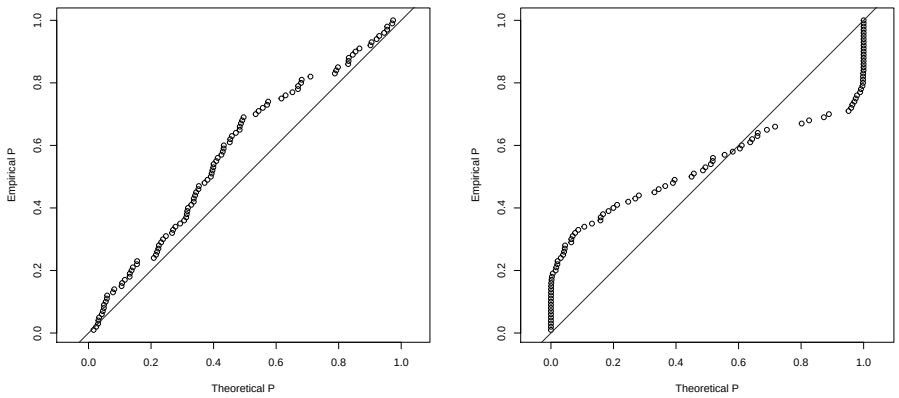

Рис. 6.5: P-P діаграми

Побудова Q-Q діаграми аналогічна, але по горизонталі та вертикалі відкладаються відповідно теоретичні та емпіричні квантилі. Точніше, роль емпіричних квантилів відіграють порядкові статистики $X_{[j]}$, яким відповідають теоретичні квантилі $Q^F(p_j)$, де $p_j = j/n - 1/(2n)$. (Значення p_j відповідає середині стрибка емпіричної функції розподілу $\hat{F}_n(x)$ у точці $x = X_{[j]}$). Таким чином, на Q-Q діаграмі відображаються точки з координатами $(Q^F(p_j), X_{[j]})$, $j = 1, \ldots, n$. Якщо розподіл даних описується ф.р. F, ці точки повинні знаходитись поблизу від бісектриси першого координатного кута.

Q-Q діаграма має важливу перевагу над P-P діаграмою. Її зручно використовувати, коли теоретична функція розподілу відома з точністю до невідомих параметрів зсуву та масштабу. Тобто відомо, що $F(x) = F_0((x-a)/s)$, де a (зсув) і s (масштаб) — невідомі параметри. (Наприклад, для нормального розподілу F_0 може бути ф.р. стандартного нормального розподілу, a — математичним сподіванням, s — середньоквадратичним відхиленням). У цьому випадку $Q^F(\alpha) = sQ^{F_0}(\alpha) + a$, отже, якщо на Q-Q діаграмі відобразити точки з координатами $(Q^{F_0}(p_j), X_{[j]})$, вони розташуються поблизу від прямої з рівнянням $y = sx + a$. Це дозволяє перевіряти гіпотезу про розподіл даних не оцінюючи параметри зсуву та масщтабу. Більше того, ці параметри можна оцінити візуально за Q-Q діаграмою.

Для нормального розподілу Q-Q діаграму у `R` можна побудувати, використовуючи функції `qqnorm()` та `qqline()`:

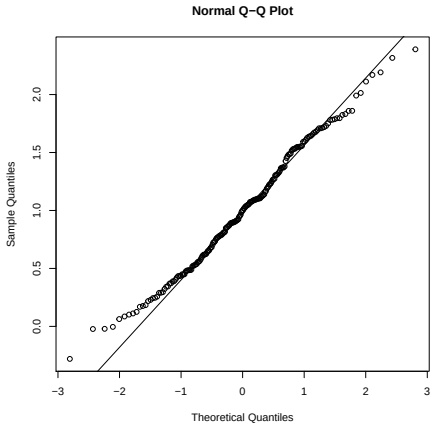

Рис. 6.6: Q-Q діаграма

```
> x<-rnorm(200,mean=1,sd=0.5)
> qqnorm(x)
> qqline(x)
```

(У x створена вибірка з нормального розподілу з середнім 1 та дисперсією 0.25, потім функція `qqnorm()` будує Q-Q діаграму у якій по осі абсцис відкладені квантилі стандартного нормального розподілу, функція `qqline()` оцінює математичне сподівання a та стандартне відхилення s за даними і проводить на діаграмі пряму $y = sx + a$.

Результат виконання цих команд зображено на рис. 6.6 Зверніть увагу, що побудована пряма не є бісектрисою першого координатного кута, але точки розташовані біля неї. Так і повинно бути, оскільки розподіл даних є нормальним, але не стандартним нормальним.

Якщо теоретичний розподіл не є нормальним, значення квантилів потрібно підраховувати, використовуючи відповідну функцію для даного розподілу. Наприклад, перевірка того, що розподіл даних є логістичним може виглядати так:

```
> x<-rnorm(200,mean=1,sd=0.5)
> plot(qlogis(ppoints(x)),sort(x))
> abline(lm(sort(x)~qlogis(ppoints(x)))$coefficients)
```

У цьому прикладі дані генеруються з нормальним розподілом, а перевірка проводится для теоретичного логістичного розподілу. Функція

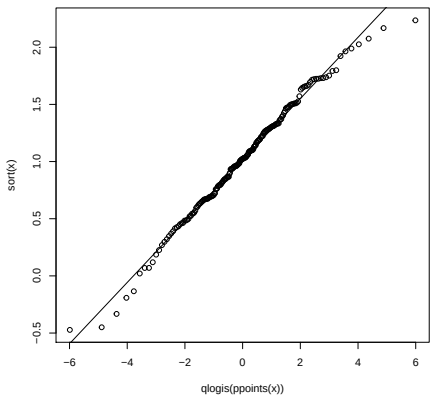

Рис. 6.7: Q-Q діаграма

`ppoints(x)` обчислює значення рівнів квантилів p_j, отже `qlogis(ppoints(x))` одразу видає вектор теоретичних квантилів, що відкладаються по горизонталі. Функція `abline()` рисує пряму лінію, коефіцієнти якої отримуються підгонкою за методом найменших квадратів (функція `lm`).

Відмітимо, що за цією Q-Q діаграмою помітити відмінність розподілу даних (нормального) від логістичного практично неможливо.

6.3 Q-Q діаграма з прогнозними інтервалами

Розглядаючи Q-Q діаграми, можна побачити, що навіть коли розподіл даних відповідає теоретичному, точки на діаграмі відхиляються від бісектриси першого координатного кута, хоча і не дуже сильно. Причому у різних частинах діаграми такі випадкові відхилення можуть бути різними. Як правило, відхилення крайніх точок більш помітні ніж точок всередині діаграми. Тому бажано крім бісектриси зобразити іще інтервали, у які з великою ймовірністю можуть потрапляти точки на діаграмі, якщо теоретичний розподіл правильно описує дані.

Стандартні функції `R` не надають такої можливості. Розглянемо спосіб побудови таких прогнозних інтервалів за допомогою імітаційного моделювання.

Нехай нам потрібно побудувати інтервал у який потраплятиме точка, що відповідає j-тій порядковій статистиці із заданою ймовірністю

$1-\alpha$. Ідея полягає в тому, щоб згенерувати багато (K) вибірок з розподілом, який відповідає теоретичному. Всі згенеровані вибірки повинні мати один і той же обсяг n, який дорівнює обсягу тієї реальної вибірки, що досліджується. По кожній такій вибірці візьмемо j-ту порядкову статистику. Отримаємо K значень $X^{(k)} = (X^k_{[j]}, k = 1, \ldots, K)$ де $X^k_{[j]}$ — j-та статистика для k-тої вибірки. За цими значеннями знайдемо емпіричні квантилі $X^-_j = Q^{X^{(k)}}(\alpha/2)$, $X^+_j = Q^{X^{(k)}}(1-\alpha/2)$. В інтервалі (X^-_j, X^+_j) буде знаходитись приблизно $(1-\alpha)K$ елементів $X^{(k)}$. За законом великих чисел, при великих K, ймовірність для j-тої порядкової статистики потрапити у цей інтервал приблизно дорівнює $1-\alpha$.

Зрозуміло, що для побудови діаграми разом з інтервалами такі підрахунки потрібно повторити для всіх $j = 1, \ldots, n$. Модельовані вибірки можуть бути одні і ті ж для різних j.

Приклад реалізації цієї ідеї у вигляді функції `QQplot`, яка перевіряє відповідність до стандартного нормального розподілу:

```
> QQplot<-function(x,K=1000,alpha=0.05){
+ n<-length(x)
+ normQ<-qnorm((1:n-0.5)/n)
+ sx<-sort(x)
+ W<-matrix(rnorm(K*n),nrow=n,ncol=K)
+ W<-apply(W,2,sort)
+ tops<-apply(W,1,quantile,probs=1-alpha/2)
+ bots<-apply(W,1,quantile,probs=alpha/2)
+ plot(c(normQ,normQ,normQ),c(tops,bots,sx),type="n",
+      xlab="theoretical quantiles",ylab="empirical quantiles")
+ points(normQ,sx,col=2)
+ segments(normQ,bots,normQ,tops,col=4)
+ abline(0,1,col=1)
+ }
> x<-rnorm(100)
> QQplot(x)
```

Результат роботи програми див. на рис. 6.8.

Розберемо роботу функції. Її параметри

`x` — вибірка, для якої будується Q-Q діаграма;

`K` — кількість псевдовипадкових вибірок, що будуть згенеровані для отримання прогнозних інтервалів (`K=1000` за умовчанням);

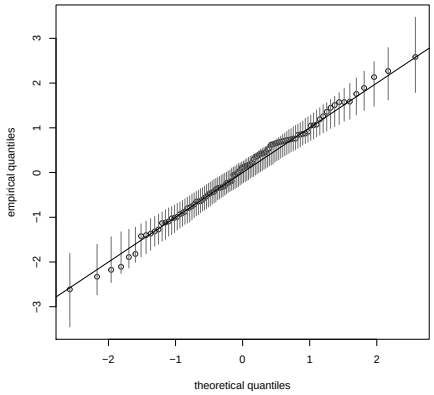

Рис. 6.8: Q-Q діаграма з прогнозними інтервалами

`alpha` — ймовірність, з якою точка повинна потрапляти до інтервалу (`alpha=0.05` за умовчанням).

У тілі функції спочатку підраховуються абсциси точок на діаграмі — у векторі `normQ`. Створюється варіаційний ряд даних — `sx`. Потім генерується матриця `W`, стовпчиками якої є K псевдовипадкових вибірок з стандартного нормального розподілу. Команда `W<-apply(W,2,sort)` впорядковує стовпчики `W` у порядку зростання. Тепер вони містять варіаційні ряди модельованих вибірок. Кожен (j-тий) рядочок матриці `W` складається тепер з порядкових статистик модельованих вибірок з індексом j. Ми шукаємо X_j^- і X_j^+ як відповідні квантилі для j-того рядочка та вміщуємо їх у вектори `bots` і `tops` для всіх $j = 1, \ldots, n$. Далі йде виведення рисунку. Спочатку виводиться тільки рамка з підписами, підігнана так, щоб у ній розмістились всі елементи рисунку. Після цього `points()` виводить точки діаграми, `segments()` — інтервали, `abline` — бісектрису координатного кута.

6.4 Порівняння розподілів кількох наборів даних.

У статистиці часто виникає задача порівняння розподілів різних наборів однотипних даних. Скажімо, за даними податкової інспекції можна по-

ставити питання: чи відрізняється розподіл доходів населення у минулому та у позаминулому році? Для порівняння розподілів двох наборів даних можна використовувати рисунки, на яких зображено дві гістограми одразу, або Q-Q діаграми, де по горизонталі відкладено квантилі одного набору, а по вертикалі - іншого.

Наприклад, розглянемо дані про чайові з набору `tips.csv`, який ми вже використовували у підрозділі 6.1. Ми хочемо перевірити, чи відрізняються розподіли чайових в залежності від того, хто їх сплачує — чоловік, чи жінка? Гістограми та Q-Q діаграми для такої перевірки можна вивести наступним чином:

```
> # читаємо дані з файлу:
> z<-read.csv("c:/rem/rstat/data/tips.csv")
> #
> # Будуємо дві гістограми на одному рисунку
> #
> hist(z$tip[z$sex=="M"],breaks=10,probability=T,
+     angle=0,density=12,xlim=c(0,10),ylim=c(0,0.45))
> hist(z$tip[z$sex=="F"],probability=T,
+     breaks=10,angle=90,density=12,  xlim=c(0,10),add=T)
> #
> # Q-Q діаграма
> #
>  qqplot(z$tip[z$sex=="F"],z$tip[z$sex=="M"],
+       xlab="females",ylab="males")
>  abline(0,1)
```

У цій програмі гістограма розподілу чайових для клієнтів-чоловіків (`z$sex=="M"`) виводиться першою. Її стовпчики заштриховані вертикально. Потім на тому ж рисунку виводиться гістограма для жінок з горизонтальною штриховкою. Ми обрали для порівняння гістограми відносних частот, тому, що вибірки мають помітно різний обсяг (чоловіки розплачувались частіше ніж жінки). Якби порівнювались абсолютні частоти, "жіноча" гістограма була б майже непомітна на фоні "чоловічої" (перевірте).

І гістограма, і Q-Q діаграма свідчать, що принципової різниці у розподілі чайових не помітно для основної маси спостережень. Але для чоловіків помітно кілька випадків з аномально великими чайовими, для жінок таких випадків немає.

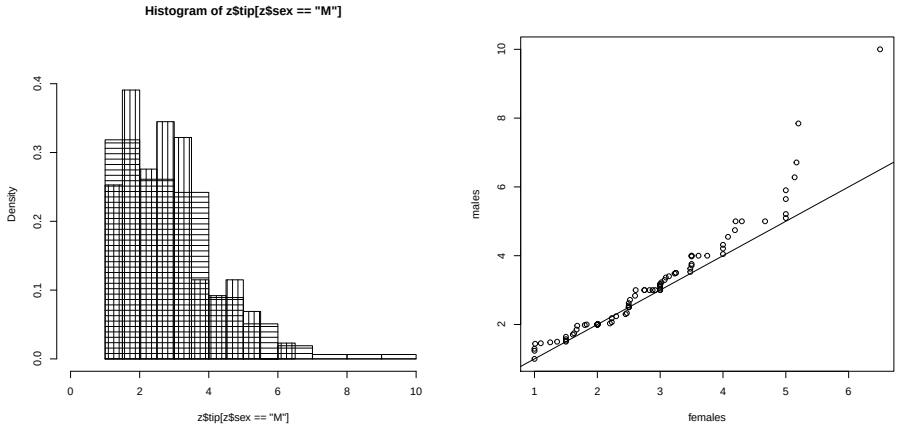

Рис. 6.9: Порівняння двох розподілів

Коли стовпчики кількох гістограм перекриваються, це створює незручність для їх візуального аналізу. Більш зручним може бути застосування діаграм, на яких стовпчики розташовані поруч (рис. 6.10). Як ми бачили у п. 3.1, такі діаграми можна рисувати, використовуючи функцію `barplot`:

```
>   z<-read.csv("c:/rem/rstat/data/tips.csv")
>   ctip<-cut(z$tip,breaks=1:10,labels=(1:9)+0.5)
>   counts<-table(z$sex,ctip)
>   counts["F",]=counts["F",]/sum(counts["F",])
>   counts["M",]=counts["M",]/sum(counts["M",])
>   barplot(counts,beside=T,col=c("darkblue","red"))
>   legend(x=16,y=0.31,c("Female","Male"),fill=c("darkblue","red"))
>   #
```

Тут функція `cut()` використана для групування даних: отримуючи на вході числовий вектор `z$tip`, вона видає вектор, елементами якого є фактори, що показують, на який інтервал розбиття потрапило відповідне значення `z$tip`. Функція `table(z$sex,ctip)` складає таблицю (матрицю) частот поява пар значень факторів `(z$sex,ctip)`:

```
> table(z$sex,ctip)

   ctip
    1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
```

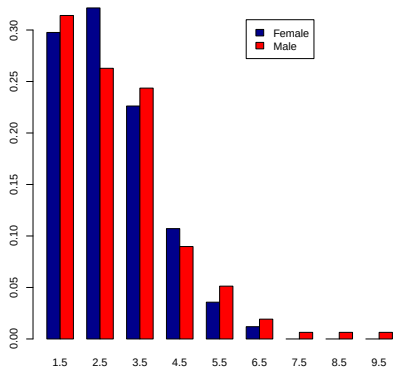

Рис. 6.10: Гістограма через barplot

```
F  25  27  19   9   3   1   0   0   0
M  49  41  38  14   8   3   1   1   1
```

— жінок (F), що дали чайові в інтервалі від 2 до 3 (позначений 2.5) було 25 і т.д.

Далі функція `barplot()` рисує стовпчикову діаграму як описано у п.3.1, а функція `legend()` виводить пояснення-легенду.

6.5 Скриньки з вусами

Гістограми дають, взагалі кажучи, найбільш повне уявлення про розподіл одновимірних даних. Однак, коли потрібно порівняти розподіли багатьох (більше трьох) наборів даних, зображення гістограм усіх цих наборів на одному рисунку стає занадто складним для візуального сприйняття. Тому для забезпечення можливості графічного аналізу даних потрібно пожертвувати частиною інформації, відображаючи для кожного набору не гістограму а лише найбільш характерні риси розподілу.

Цей підхід приводить до діаграми, яка англійською мовою зветься box-whisker plot, або просто boxplot. Українською це можна перекласти як “скринька з вусами”.

Для одного набору одновимірних даних скриньки з вусами будуються за схемою, зображеною на рис.6.11. На цьому рисунку значення даних

відображаються по вертикальній осі. Прямокутник (скриньку) рисують від нижнього квартиля Q_1 (тобто квантиля рівня 1/4) до верхнього квартиля Q_3 (квантиля рівня 3/4) порахованих за даними. Лінія, що розрізає прямокутник відповідає медіані med. Вусики, що стирчать зі скриньки відмічають діапазон розташування даних, які не є викидами. Тобто верхній вусик відповідає найбільшому не викиду, нижній - найменшому. Кожен кружечок поза діапазоном відповідає одному індивідуальному значенню-викиду.

Для визначення того, які спостереження слід віднести до викидів є різні підходи, що мають евристичний характер. При найбільш поширеному, викидами вважають ті значення, що перевищують $Q_3 + 1.5\text{IQ}$ або є меншими ніж $Q_1 - 1.5\text{IQ}$, де $\text{IQ}{=}Q_3 - Q_1$ — інтерквартильний розмах. Іноді виділяють іще "далекі" викиди, або екстремальні значення, тобто ті значення даних, які виходять за межі інтервалу $[Q_1 - 3\text{IQ}, Q_3 + 3\text{IQ}]$. Якщо цей підхід використовується, то екстремальні значення позначають на діаграмі хрестиками, а помірні викиди (тобто такі, які не є екстремальнми) — кружечками.

Множники 1.5 та 3 у цих формулах не мають якогсь науково-математичного або потаємно-містичного змісту, а використовуються лише за домовленістю.

Інколи у стінках скриньки роблять трикутні зарубки (notches), зовнішні краї яких відповідають довірчому інтервалу для медіани розподілу даних з рівнем значущості 0.95. (На рисунку такий довірчий інтервал позначений стрілками).

Як правило, ширина прямокутника-скриньки та вусиків обирається так, щоб рисунок було зручно сприймати на око, інформації про дані вона не несе. Але інколи ширину скриньки вибирають пропорційно кореню квадратному з кількості елементів у наборі даних, за яким вона побудована — чим ширша скринька, тим більше у ній даних.

Можливе також горизонтальне розташування скриньки з вусами. Одна скринька для єдиного набору даних несе небагато інформації. Але розмістивши декілька таких скриньок паралельно для різних наборів, можна одразу помітити характерні відмінності розподілів даних у різних наборах.

Для рисування кількох скриньок з вусами у `R` можна використовувати функцію `boxplot`. Першим (основним) параметром цієї функції є список наборів (векторів) даних, для яких будуються скриньки з вусами. Наприклад:

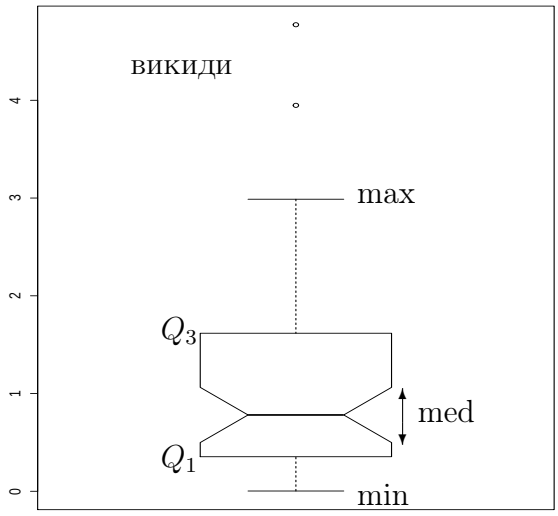

Рис. 6.11: Скринька з вусами

```
> set.seed(20)
> a<-rexp(200)
> b<-rnorm(100,2,1)
> c<-rchisq(40,5)
> x<-list(a,b,c)
> boxplot(x,notch=T,varwidth=T,names=c("exp","norm","chisq"))
```

Тут ми згенерували[3] три вибірки з різними розподілами: експоненційним, нормальним та хі-квадрат, склали їх в один список і відобразили за допомогою `boxplot`.

На рисунку 6.12 можна помітити симетрію нормальної вибірки, асиметрію експоненційної. Хі-квадрат розподіл є асиметричним, але на рисунку ця асиметрія виражена не сильно. Викиди не відмічені у нормальній вибірці, два викиди — у хі-квадрат. Сім "викидів" зафіксовано у експоненційній вибірці, але за їх розташуванням можна скоріше твердити, що більшість з них не далеко відійшли від основної маси спостережень,

[3]функція `set.seed()` встановлює стартове значення (зернину) для генерації псевдовипадкових послідовностей. Таким чином, при кожному запуску цієї програми вона буде генерувати одні і ті ж числа та відповідні рисунки.

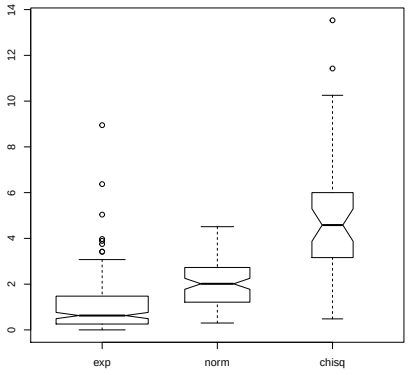

Рис. 6.12: Порівняння трьох розподілів

тобто трактування їх як викидів є питанням смаку.

Ми скористались опцією `notch=T` для того, щоб відобразити довірчі інтервали для медіан у вигляді зарубок на скриньках. За цими інтервалами можна зробити попередній висновок, що медіани теоретичних розподілів вибірок не слід вважати однаковими.

Опція `varwidth=T` вказує, що ширину скриньок слід обирати пропорційно до кореня з обсягу вибірки — тому скринька для `exp` вийшла помітно ширшою ніж інші.

Опція `names` задає імена, що будуть підписані під скриньками. Аналогічно можна використовувати опцію `col` щоб задавати кольори скриньок.

У комп'ютерній статистиці часто виникають задачі аналізу даних, що записані у єдиному фреймі, причому одна змінна містить певну числову характеристику (відгук) об'єктів, що досліджуються, а інша (фактор) — клас, до якого належить даний об'єкт. При цьому питання полягає в тому, щоб проаналізувати залежність розподілу відгука від фактора. У таких випадках для опису задачі у `boxplot()` перший параметр можна задати формулою вигляду

відгук $\sim$ фактор[4]..

Наприклад, у фреймі даних `InsectSprays` містяться дані про випробування якості різних видів інсектицидів. Один рядочок даних відповідає одному випробуванню. У кожному випробуванні обчислювалась

[4]Можна вказати декілька факторів, наприклад: відгук $\sim$ фактор1+фактор2.

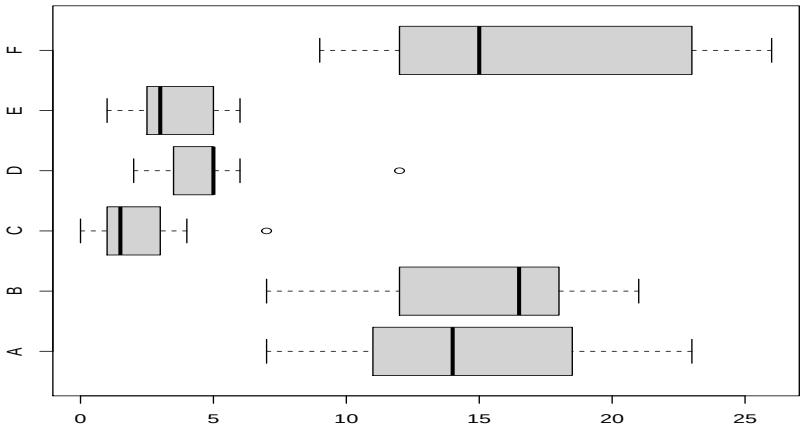

Рис. 6.13: Порівняння ефективності інсектицидів

кількість комах, що загинули під дією інсектициду — змінна `count`. У змінній `spray` вказується тип інсектициду (літера A-F). Нас цікавить, як розподіл `count` пов'язаний з `spray`. Відповідні скриньки задає програма

```
>  boxplot(count ~ spray, data = InsectSprays,
+          col = "lightgray",horizontal=T)
```

Тут `data` задає фрейм даних, з якого вибирають змінні, опція `horizontal=T` показує, що скриньки розміщуються горизонтально.

На рис. 6.13 бачимо, що інсектициди C, D, E виявились значно менш ефективними ніж інші, інскетицид F у деяких експериментах виявив себе найкращим, але найкраща медіана — у B і т.д.

Відмітимо, що аналогічну діаграму можна отримати, якщо записати `plot(count~spray,data=InsectSprays)`.

Розділ 7

Оцінювання невідомих параметрів розподілу

7.1 Оцінки узагальненого методу моментів

SecMomEst

Нехай спостережувані змінні являють собою кратну вибірку $\mathbf{X} = (\xi_i, \ldots, \xi_n)$, де $\xi_j \in \mathbb{R}^d$ — незалежні випадкові вектори з розподілом

$$\mathbf{P}_\vartheta(A) = \mathbf{P}^\xi_\vartheta(A) = \mathsf{P}\{\xi_j \in A\},$$

де $\vartheta \in \Theta \in \mathbb{R}^p$ — p-вимірний невідомий параметр, Θ — множина можливих значень невідомого параметра. (ϑ можна трактувати, як набір p числових невідомих параметрів).

Для того, щоб оцінити ϑ, задамо деяку вимірну функцію $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^p$, так, щоб для всіх $\mathbf{t} \in \Theta$ було скінченним математичне сподівання

$$\mathbf{H}(\mathbf{t}) = \mathsf{E}_\mathbf{t}\, \mathbf{h}(\xi_1) = \int_{\mathbb{R}^d} \mathbf{h}(\mathbf{x}) \mathbf{P}_\mathbf{t}(d\mathbf{x}).$$

Внаслідок закону великих чисел, при великих обсягах вибірки n

$$\hat{\mathbf{h}}_n = \frac{1}{n} \sum_{j=1}^{n} \mathbf{h}(\xi_j) \approx \mathbf{H}(\vartheta).$$

Прирівняємо

EqMM1

$$\hat{\mathbf{h}}_n = \mathbf{H}(\mathbf{t}) \tag{7.1}$$

і виберемо на роль оцінки ϑ таку статистику[1] $\hat{\vartheta} = \hat{\vartheta}(\mathbf{X})$, при підстановці якої замість $\mathbf{t}$ рівність у (7.1) виконується майже напевно. $\hat{\vartheta}_n$ називають оцінкою методу моментів (моментною оцінкою) для ϑ, з моментною функцією $\mathbf{h}$. Функцію $\mathbf{H}(\vartheta)$ називають (узагальненим) теоретичним моментом (або вектором моментів) розподілу $\mathbf{P}_\vartheta$, а $\hat{\mathbf{h}}_n$ — емпіричним моментом вибірки $\mathbf{X}$. У випадку одновимірних спостережень ($d = 1$), при $h(x) = x^k$, $H(\vartheta) = \mathsf{E}_\vartheta\, \xi^k$ називають k-тим теоретичним моментом, а $\hat{h}_n = \frac{1}{n}\sum_{j=1}^n \xi_j^k$ — k-тим емпіричним моментом.

Якщо рівняння (відносно $\mathbf{t}$)

$$\boxed{\text{EqMM2}}\quad \mathbf{H}(\mathbf{t}) = \mathbf{x} \tag{7.2}$$

має єдиний корінь для всіх x що належать множині можливих значень функції $\mathbf{h}$, то $\hat{\vartheta} = \mathbf{H}^{-1}(\hat{\mathbf{h}})$, де $\mathbf{H}^{-1}$ — функція, обернена до функції $\mathbf{H}$. (При цьому потрібно, щоб $\mathbf{H}^{-1}$ була вимірною функцією).

Якщо рівняння (7.1) має декілька коренів, то оцінка методу моментів визначена неоднозначно: будь-який з коренів можна використовувати як оцінку.

Приклад 1. Нехай $\mathbf{X} = (\xi_1, \dots, \xi_n)$ кратна вибірка з експоненційного розподілу з невідомою інтенсивністю λ, тобто щільність розподілу ξ

$$f_\lambda(x) = \lambda e^{-\lambda x} \mathbb{1}\{x > 0\}.$$

Задача полягає в оцінці $\lambda \in (0, \infty)$. Виберемо на роль моментної функцію $h^{(1)}(x) = x$. Тоді

$$H(\lambda) = \mathsf{E}_\lambda\, h^{(1)}(\xi_1) = \int_0^\infty x f_\lambda(x) dx = \frac{1}{\lambda}.$$

Отже оцінка методу моментів з цією моментною функцією має вигляд

$$\hat{\lambda}_n^{(1)} = \frac{1}{\hat{h}_n^{(1)}} = \frac{1}{\bar{\xi}} = \frac{n}{\sum_{j=1}^n \xi_j}.$$

Якщо обрати моментну функцію $\hat{h}^{(2)}(x) = x^2$, отримуємо іншу оцінку:

$$\mathsf{E}_\lambda (\xi_1)^2 = \frac{2}{\lambda^2},$$

[1] тобто вимірну функцію від даних $\mathbf{X}$.

тому оцінка методу моментів має вигляд

$$\hat{\lambda}_n^{(2)} = \sqrt{\frac{2}{\hat{h}_n^{(2)}}}.$$

Приклад 2. Нехай $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ — кратна вибірка з нормального розподілу з невідомим математичним сподіванням μ та невідомою дисперсією σ^2. Позначимо невідомий векторний параметр $\vartheta = (\mu, \sigma^2)^T \in \Theta = \mathbb{R} \times (0, +\infty)$. Позначимо також $\mathbf{t} = (m, s^2)^T$. Виберемо на роль моментної функції $\mathbf{h}(x) = (x, x^2)^T$. Тоді $\mathbf{H}(\mathbf{t}) = (m, s^2 + m^2)^T$. Отже, оцінка методу моментів знаходиться як розв'язок системи рівнянь

$$\begin{cases} \bar{\xi} = & m, \\ \overline{\xi^2} = & s^2 + m^2, \end{cases}$$

де $\bar{\xi} = \frac{1}{n}\sum_{j=1}^n \xi_j$, $\overline{\xi^2} = \frac{1}{n}\sum_{j=1}^n (\xi_j)^2$ — перший і другий вибіркові моменти.

Отже $\hat{\vartheta}_n = (\bar{\xi}, \overline{\xi^2} - (\bar{\xi})^2)^T$, тобто оцінками для μ та σ^2 є вибіркове середнє та (не виправлена) вибіркова дисперсія.

Легко бачити, що всі оцінки у прикладах 1-2 є сильно констстентними. Наступна теорема дає достатні умови консистентності моментних оцінок.

Теорема 7.1.1 *Нехай* $\mathbf{X}$ *— кратна вибірка,* $\mathbf{H}(\mathbf{t})$ *існує для всіх* $\mathbf{t} \in \Theta$*,* $\mathbf{H}^{-1}$ *існує і є неперервною на множині всіх можливих значень моментної функції. Тоді*

$$\hat{\vartheta}_n = \mathbf{H}^{-1}(\hat{\mathbf{h}}_n) \to \vartheta \text{ м.н. при } n \to \infty.$$

Доведення безпосередньо випливає з підсиленого закону великих чисел.

Метод моментів інколи можна узагальнити на випадок неоднаково розподілених спостережень.

Приклад 3. Нехай випадкові величини, що самі мають нормальний розподіл, вимірюються різними приладами, які мають певні похибки вимірювання. Таким чином, результати вимірювання $\xi_j = \eta_j + \varepsilon_j$, де η_j — справжнє значення величини, виміряної у j-тому досліді, ε_j — похибка вимірювання. η_j, ε_j $j = 1, \ldots, n$ вважаються незалежними в сукупності, $\eta_j \sim N(\mu, \sigma^2)$, $\varepsilon_j \sim N(0, s_j^2)$, де s_j^2 — відома дисперсія похибки при j-тому вимірюванні, μ і σ^2 — невідомі параметри, які треба оцінити за даними $\mathbf{X} = (\xi_1, \ldots, \xi_n)$.

Будемо вважати, що дисперсії похибок обмежені зверху: $\sigma_j^2 < S < \infty$.

Легко бачити, що, хоча ξ_j не є однаково розподіленими випадковими величинами, але $\bar{\xi} = \bar{\eta} + \bar{\varepsilon} \to \mu$ при $n \to \infty$ м.н., оскільки $\bar{\eta} \to \mu$ за підсиленим законом великих чисел, а $\bar{\varepsilon} \sim N(0, \sum_{j=1}^n \sigma_j^2/n)$ збігається до 0 м.н. (Це легко довести, використовуючи лему Бореля-Кантеллі).

Отже $\bar{\xi}$ є незміщеною та консистентною оцінкою μ.

Задача. Побудуйте консистентну оцінку σ^2 у цьому прикладі.

Розглянемо тепер приклад застосування `R` для обчислення оцінок методу моментів у випадку, коли розв'язати моментне рівняння (7.1) аналітично не вдається.

Приклад 4. Нехай дані $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ являють собою кратну вибірку зі зрізаного експоненційного розподілу з функцією розподілу

$$\boxed{\text{EqMomEstF}} \quad F_\xi(x) = F(x; \lambda, C) = \begin{cases} 0 & \text{при } x < 0, \\ \frac{1-\exp(-\lambda x)}{1-\exp(-\lambda C)} & \text{при } 0 \le x < C, \\ 1 & \text{при } x \ge C. \end{cases} \tag{7.3}$$

Вважаємо поріг зрізання C — відомим, а $\lambda > 0$ — невідомим параметром розподілу, який потрібно оцінити.

На роль моментної функції виберемо $h(x) = x$. Легко бачити, що

$$H(\lambda) = \mathsf{E}_\lambda \xi_1 = \frac{C}{1 - \exp(C\lambda)} + \frac{1}{\lambda}.$$

Позначимо розв'язок рівняння $\bar{\xi} = H(l)$ (відносно l) через $\hat{\lambda}_n^{MM}$ — це і буде оцінка методу моментів для λ. Оскільки розв'язати моментне рівняння аналітично не можна, для знаходження оцінки застосуємо техніку наближеного обчислення кореня цього рівняння.

Наприклад, для цього можна використати функцію `nleqslv` з бібліотеки `nleqslv`. Найпростіший варіант виклику цієї функції - `nleqslv(x, fn)`, де `x` — початкове наближене значення для кореня, `fn` — функція, корінь якої потрібно знайти. (Тобто ми шукаємо розв'язок рівняння `fn(x)=0`). Значенням функції `nleqslv` є об'єкт, що має багато атрибутів, зокрема у атрибуті `$x` знаходиться отримане наближене значення кореня, у атрибуті `$fvec` — значення функції у точці `x` (якщо корінь знайдено вірно, це значення має бути практично 0).

Оформимо обчислення оцінки за даними `X` у вигляді функції:

```
> library(nleqslv)
> # функція eqv задає рівняння H(l)=Mx
> # trun - поріг зрізання експоненційного розподілу
> eqv<-function(l,Mx,trun){
+  trun/(1-exp(trun*l))+1/l-Mx
+ }
> # функція EstMM рахує оцінку lambda за даними X
> # методом моментів
> EstMM<-function(x,trun){
+  Mx<-mean(x)
+  nleqslv(1/Mx,eqv,Mx=Mx,trun=trun)$x
+ }
```

Тут ми спочатку створили функцію `eqv`, коренем якої буде наша оцінка, а потім — функцію `EstMM`, яка рахує оцінку. Аргументами цієї функції є `x` — вибірка, по якій будується оцінка і `trun` — параметр зрізання (відомий).

Функція `EstMM` спочатку знаходить вибіркове середнє і записує його як змінну `Mx`, а потім викликає функцію `nleqslv` для розв'язування моментного рівняння. При цьому як початкове наближення для кореня рівняння вибрано `1/Mx`, тобто моментну оцінку для інтенсивності не зрізаного експоненційного розподілу.

Перевіримо, чи правильно працює наша функція на модельованих данних, які мають зрізаний експоненційний розподіл. Для цього потрібно спочатку згенерувати дані з таким розподілом, а потім викликати функцію `EstMM`:

```
>  set.seed(2)
> # Генерація псевдовипадкових даних
> U<-2     # поріг зрізання
> l<-0.5   #  інтенсивність
> n<-10000 # обсяг вибірки
> # функція rexptr генерує одне псевдовипадкове
> # число зі зрізаним експоненційним розподілом
> # з інтенсивністю lambda та порогом зрізання trun
> rexptr<-function(lambda=1,trun=1){
+  repeat{
+    x<-rexp(1,lambda)
+    if(x<trun) break
+  }
```

```
+  x
+ }
> # Генеруємо вектор зрізаних експоненційних в.в.
> X<-rep(1,n)
> X<-sapply(X,rexptr,trun=U)
> # Рахуємо оцінку
> EstMM(X,U)

[1] 0.5033029
```

Спочатку ми створили функцію `rexptr`, яка генерує одне псевдовипадкове число, використовуючи генератор експоненційного розподілу `rexp()` і зрізаючи його результат доти, доки він не стане меншим ніж поріг зрізання. Потім генеруємо вибірку використовуючи `sapply()` і підраховуємо оцінку для інтенсивності за цією вибіркою.

Справжня інтненсивність `l=0.5`, оцінка — 0.5033029.

Результат виглядає задовільним. Але, звичайно, якість алгоритму оцінювання не можна визначити лише за оцінкою по лише одній вибірці.

Відмітимо, що розв'язок моментного рівняння може бути від'ємним для деяких вибірок. Оскільки за змістом λ додатне, немає рації використовувати від'ємне значення як його оцінку. У такому випадку можна лише стверджувати, що λ настільки мале, що його неможливо оцінити точно. В принципі, при $\lambda \to 0$ функція розподілу зрізаного нормального розподілу перетворюється у рівномірну на інтервалі $[0, C]$. Якщо для реальних даних які розглядаються модель рівномірного розподілу допустима, по можна вибрати як оцінку λ величину $\hat{\lambda}_n^{MMtr} = \max(\hat{\lambda}_n^{MM}, 0)$, вважаючи, що нульовому значенню оцінки відповідає рівномірний розподіл.

7.2 Оцінки методу квантилів

SecQuantEst

Як ми бачили у розділі 4.1.1, такі вибіркові моменти як вибіркове середнє і дисперсія є не робстними характеристиками вибірки. Тому коли припускається, що дані можуть бути забруднені викидами, доцільно замість моментів використовувати для оцінювання більш робастні статистики. Такими статистиками є вибіркові квантилі, якщо їх рівні не є близькими до 0 або 1. Найбільш робастною статистикою є вибіркова медіана, тобто квантиль рівня $1/2$.

Нехай $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ — кратна вибірка з неперервногю розподілу F_ϑ спостереження ξ_j, $\vartheta \in \Theta \subseteq \mathbb{R}$ — невідомий параметр. Позначимо $Q^{\mathbf{X}}(\alpha)$ — вибіркову квантиль рівня α, $Q^{F_\vartheta}(\alpha)$ — теоретичну квантиль розподілу F_ϑ. Тоді для всіх α таких, що $F_\vartheta(\cdot)$ є строго зростаючою у деякому околі $Q^{F_\vartheta}(\alpha)$, має місце збіжність

$$Q^{\mathbf{X}}(\alpha) \to Q^{F_\vartheta}(\alpha), \text{ м.н. при } n \to \infty.$$

Нехай при деякому α функція $q(t) = Q^{F_t}(\alpha)$ має неперервну обернену $q^{-1}(u)$ на множині можливих значень $Q^{\mathbf{X}}(\alpha)$ (для всіх можливих значень $\mathbf{X}$). Покладемо $\hat{\vartheta}^Q = q^{-1}(Q^{\mathbf{X}}(\alpha))$. Тоді, якщо при справжньому значенні невідомого параметра ϑ $F_\vartheta(\cdot)$ є строго зростаючою в околі $Q^{F_\vartheta}(\alpha)$, то $\hat{\vartheta}^Q$ — строго консистентна оцінка ϑ.

Приклад 1. Нехай $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ має експоненційний розподіл з невідомою інтенсивністю λ. Тоді $F_\lambda(x) = (1 - \exp(-\lambda x))\mathbb{1}\{x > 0\}$, отже $Q^{F_\lambda}(1/2) = \log 2/\lambda$. На роль оцінки для λ можна обрати

$$\hat{\lambda}^{med} = \frac{\log 2}{\text{med}(\mathbf{X})}$$

Ця оцінка є сильно консистентною і робастною. Її звуть медіанною оцінкою інтенсивності експоненційного розподілу.

Приклад 2. Нехай $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ — кратна вибірка з розподілу $F \sim N(\mu, \sigma^2)$, параметри μ та σ^2 — невідомі, їх потрібно оцінити. Оскільки щільність нормального розподілу симетрична навколо μ, то μ є медіаною цього розподілу, отже як оцінку для нього можна взяти вибіркову медіану $\hat{\mu}_n^{med} = \text{med}(\mathbf{X})$.

Для оцінки σ скористаємось тим, що

$$Q^{N(\mu,\sigma^2)}(\alpha) = \mu + \sigma Q^{N(0,1)}(\alpha).$$

Тому, для будь-якого α,

$$\sigma = \frac{Q^F(1-\alpha) - Q^F(\alpha)}{2\lambda_\alpha},$$

де $\lambda_\alpha = Q^{N(0,1)}(1-\alpha)$. Традиційно для побудови оцінки вибирають $\alpha = 1/2$ і отримують

$$\hat{\sigma}_n^{IQ} = \frac{Q^{\mathbf{X}}(3/4) - Q^{\mathbf{X}}(1/4)}{2\lambda_{1/4}} \approx \frac{\text{IQ}(\mathbf{X})}{1.34898}$$

де $\mathrm{IQ}(\mathbf{X})$ — інтерквартильний розмах вибірки $\mathbf{X}$. Ця оцінка зветься інтерквартильною оцінкою середньоквадратичного відхилення.

Оцінки $\hat{\mu}_n^{med}$ та $\hat{\sigma}_n^{IQ}$ є сильно консистентними.

Приклад 3. Розглянемо дані спостережень нормальних випадкових величин з нормальною похибкою, описані у прикладі 3 розділу 7.1: $\mathbf{X} = (\xi_1, \ldots, \xi_n)$, $\xi_j \sim N(\mu, \sigma^2 + \sigma_j^2)$, спостереження незалежні.

Хоча спостереження не є однаково розподіленими, але медіани всіх ξ_j — однакові і дорівнюють μ. Використовуючи це, при додатковій умові $\sigma_j^2 < S < \infty$ можна показати, що $\mathrm{med}(\mathbf{X})$ буде консистентною оцінкою μ.

Ми, фактично, визначили квантильну оцінку як розв'язок рівняння

$$\boxed{\text{EqQuantEst3}} \quad Q^{F_t}(\alpha) = Q^{\mathbf{X}}(\alpha) \tag{7.4}$$

відносно t. Часто функцію $Q^{F_t}(\alpha)$ буває неможливо записати у явному вигляді і розв'язування цього рівняння становить самостійну проблему.

У таких випадках можна переписати (7.4) у еквівалентному вигляді

$$\boxed{\text{EqQuantEst4}} \quad F_t(Q^{\mathbf{X}}(\alpha)) = \alpha, \tag{7.5}$$

і шукати оцінку як розв'язок цього рівняння відносно t.

Приклад 4. Розглянемо знову кратну вибірку зі зрізаного експоненційного розподліу $\mathbf{X} = (\xi_1, \ldots, \xi_n)$, описану у прикладі 4 з п. 7.1. Для медіани рівняння (7.5) перетворюється на $F(\mathrm{med}(\mathbf{X}), \lambda, C) = 1/2$, де $F(x, \lambda, C)$ задано (7.3). Отже медіанна оцінка для λ є коренем рівняння (відносно l):

$$\frac{1 - \exp(-l\,\mathrm{med}(\mathbf{X}))}{1 - \exp(-lC)} = 1/2.$$

У R оформити підрахунок таких оцінок можна так само, як це було зроблено для моментних оцінок:

```
> # функція eqvmed медіанне рівняння F(medi,l)=1/2
> # medi - медіана вибірки, l - оцінка інтенсивності
> eqvmed<-function(l,medi,trun){
+     (1-exp(-l*medi))/(1-exp(-l*trun))-1/2
+  }
>  # функція EstMmed рахує оцінку lambda за даними X
>  # методом медіан
>  EstMed<-function(x,trun){
+     Mx<-median(x)
```

```
+     nleqslv(log(2)/Mx,eqvmed,medi=Mx,trun=trun)$x
+  }
```

На даних, згенерованих у п. 7.1, функція `EstMed()` дає значення оцінки 0.5057097 (при справжньому $\lambda = 1/2$. Це трохи менш точно, ніж результат моментного оцінювання, але теж досить добре.

Ця оцінка теж може приймати від'ємні значення, як і оцінка методу моментів у цій задачі, розглянута у п. 7.1 які також можна замінити 0.

Відмітимо, що у цьому прикладі забруднення даних дуже великими викидами неможливе в принципі: спостереження, що знаходяться за межами інтервалу $[0, C]$ не можуть належати зрізаному експоненційному розподілу. Такі спостереження, якщо вони потраплять до вибірки, слід трактувати не як забруднення, а як грубі помилки — і вилучати з розгляду. (Або відмовитись від моделі зрізаного експоненційного розподілу для таких даних). Тому застосування медіанної оцінки у цій задачі навряд чи можна обгрунтувати посилаючись на вимогу робастності.

7.3 Оцінки методу найбільшої вірогідності

SecMLEst

На відміну від методів моментів і квантилів, метод найбільшої вірогідності у загальному випадку не потребує однорідних незалежних спостережень. Але при використанні цього методу потрібно, щоб розподіл даних описувався оцінюваними параметрами однозначно. Отже, нехай дані $\mathbf{X}$ розглядаються як випадковий елемент деякого простору можливих значень даних $\mathcal{X}$, що має розподіл $P_\vartheta^{\mathbf{X}}(A) = \mathsf{P}\{\mathbf{X} \in A\}$, $\vartheta \in \Theta \subseteq \mathbb{R}^d$ — невідомий параметр цього розподілу.

Припустимо, що існує міра μ на просторі $\mathcal{X}$ і сім'я функцій $f_\vartheta^{\mathbf{X}}(\mathbf{x})$, $f_\vartheta : \mathcal{X} \to \mathtt{R}$, $\vartheta \in \Theta$, така, що

$$P_\vartheta^{\mathbf{X}}(A) = \int_A f_\vartheta^{\mathbf{X}}(\mathbf{x})\mu(d\mathbf{x})$$

для всіх вимірних підмножин $A \in \mathcal{X}$ та всіх $\vartheta \in \Theta$.

Функція $f_\vartheta^{\mathbf{X}}$ зветься щільністю розподілу $\mathbf{X}$ відносно міри μ. Якщо $\mathcal{X} \subseteq \mathbb{R}^n$, а міра μ є мірою Лебега, функцію $f_\vartheta^{\mathbf{X}}$ називають сумісною щільністю елементів вектора $\mathbf{X}$ (спостережень).

Функцією вірогідності називають випадкову функцію від невідомого параметра, яка отримується при підстановці даних замість аргумента у

щільність розподілу:

$$L(\mathbf{t}) = f_{\mathbf{t}}(\mathbf{X}),\ \mathbf{t} \in \Theta.$$

Логарифмічна функція вірогідності це логарифм $L(\mathbf{t})$, тобто $l(\mathbf{t}) = \log L(\mathbf{t})$.

Оцінкою методу найбільшої вірогідності для параметра ϑ називають таку статистику $\hat{\vartheta}_n^{ML}$, на якій досягається максимум функції вірогідності:

$$L(\hat{\vartheta}_n^{ML}) = \max_{\mathbf{t}\in\Theta} L(\mathbf{t}).$$

Зрозуміло, що оцінка найбільшої вірогідності є також точкою максимуму логарифмічної функції вірогідності.

У випадку, коли дані $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ являють собою набір незалежних спостережень ξ_j, функція вірогідності є добутком щільностей окремих спостережень:

$$L(\mathbf{t}) = \prod_{j=1}^{n} f_{\mathbf{t}}^j(\xi_j),$$

де $f_\vartheta^j(\mathbf{x})$ — щільність розподілу спостереження ξ_j в припущенні, що справжнє значення невідомого параметра дорівнює ϑ.

Для кратної вибірки $f_\vartheta^j(\mathbf{x}) = f_\vartheta(\mathbf{x})$ не залежить від j.

Приклад 1. Знову розглянемо кратну вибірку $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ з експоненційним розподілом. Щілність розподілу $f_\lambda(x) = \lambda e^{-\lambda x}\mathbb{1}\{x > 0\}$, невідомий параметр λ потрібно оцінити. Запишемо логарифмічну функцію вірогідності:

$$l(\lambda) = \log\left(\prod_{j=1}^{n} f_\lambda(\xi_j)\right) = n\log(\lambda) - \lambda\sum_{j=1}^{n}\xi_j.$$

Легко бачити, що максимум цієї функції по λ досягається при

$$\hat{\lambda}_n^{MLE} = \frac{1}{\bar{\xi}}.$$

Таким чином, у цій задачі оцінка методу найбільшої вірогідності дорівнює моментній оцінці з моментною функцією $h(x) = x$.

Приклад 2. Розглянемо кратну вибірку $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ з нормального розподілу з невідомими математичним сподіванням μ та дисперсією σ^2. Щільність одного спостереження має вигляд

$$f_{\mu,\sigma^2} = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Логарифмічна функція вірогідності має вигляд

$$l(\mu,\sigma^2) = -n(\log(2\pi)/2 + \log\sigma) - \frac{\sum_{j=1}^n (\xi_j-\mu)^2}{2\sigma^2}.$$

Взявши похідні від цієї функції по обох аргументах і прирівнявши їх до 0, знаходимо точку максимуму, яка і буде оцінкою методу найбільшої вірогідності:

$$\hat\mu_n^{MLE} = \bar\xi,\ \hat\sigma_n^{2\ MLE} = S^2(\mathbf{X}).$$

Отже, і у цьому випадку оцінки методу найбільшої вірогідності дорівнюють отриманим у п. 7.1 оцінкам методу моментів.

Приклад 3. Як виглядатиме оцінка методу найбільшої вірогідності у задачі оцінювання математичного сподівання та дисперсії гауссового розподілу за спостереженнями з неоднорідними похибками з прикладу 3 п. 7.1? У цьому випадку щільність розподілу одного спостереження ξ_j

$$f^j_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi(\sigma^2+\sigma_j^2)}}\exp\left(-\frac{(x-\mu)^2}{2(\sigma^2+\sigma_j^2)^2}\right).$$

Логарифмічна функція вірогідності має вигляд $l(\mu,\sigma) = \prod_{j=1}^n \log f^j_{\mu,\sigma}(\xi_j)$. Перетворюючи цей вираз отримуємо, що точки максимуму функції $l(\mu,\sigma)$ співпадають з точками мінімуму функції

$$r(\mu,\sigma) = \sum_{j=1}^n \log(\sigma^2+\sigma_j^2) + \sum_{j=1}^n \left(\frac{(\xi_j-\mu)^2}{(\sigma^2+\sigma_j^2)}.\right)$$

При фіксованому σ мінімум цієї функції по μ досягається при

$$\mu = \mu(\sigma) = \frac{\sum_{j=1}^n \frac{\xi_j}{\sigma^2+\sigma_j^2}}{\sum_{j=1}^n \frac{1}{\sigma^2+\sigma_j^2}}.$$

Таким чином, для знаходження оцінки найбільшої вірогідності параметрів μ та σ, можна спочатку знайти оцінку $\hat\sigma_n^{MLE}$ як точку мінімуму функції $r(\mu(s),s)$ по s, а потім отримати оцінку для μ як $\hat\mu_n^{MLE} = \mu(\hat\sigma_n^{MLE})$.

Реалізуємо цю ідею в `R` . Мінімізувати функцію $r(\mu(s),s)$ аналітично не можна, тому будемо робити це наближеним методом Ньютона, використовуючи функцію `nlm()`. Виклик цієї функції: `nlm(f,p,...)`, де `f` — числова функція векторного аргументу, яку потрібно мінімізувати, `p` —

вектор початкових значень для точки мінімуму. Функція `f` повинна мати першим параметром вектор, по якому іде мінімізація, він повинен бути тієї ж довжини, що і `p`. Замість `...` у виклику `nlm()` можна вказати значення інших параметрів функції `f` якщо вони потрібні. Значення точки мінімуму функція `nlm()` повертає у атрибуті `$estimate`.

Оцінку (μ, σ) можна організувати так:

```
> ll<-function(s,x,sigm)
+ {
+  ss<-s^2+sigm^2
+  mu<-sum(x/ss)/sum(1/ss)
+  sum(log(ss))+sum((x-mu)^2/ss)
+ }
> EstMLEGauss<-function(x,sigm)
+ {
+  sEst<-nlm(ll,sd(x),x=x,sigm=sigm)$estimate
+  ss<-sEst^2+sigm^2
+  muEst<-sum(x/ss)/sum(1/ss)
+  c(muEst,sEst)
+ }
```

Тут функція `ll(s,x,sigm)` забезпечує обчислення $r(\mu(s), s)$. Параметр `x` це вибірка, за якою проводиться оцінювання, `sigm` — вектор значень стандартних відхилень помилок $(\sigma_1, \ldots, \sigma_n)$ (він повинен мати таку ж довжину, як і `x`.

Функція `EstMLEGauss` знаходить точку мінімуму функції `ll()`, використовуючи як початкове наближення стандартне відхилення вибірки (це, вочевидь, завищена оцінка, оскільки у неї входять дисперсії похибок).

Перевіримо роботу цієї оцінки на модельованих даних. Стандартні відхилення похибок σ_j для моделювання виберемо так, щоб вони рівномірно збільшувались від 1 на початку до 3 наприкінці спостережень. Оцінюване стандартне відхилення виберемо рівним $\sigma = 0.5$, математичне сподівання $\mu = 1$.

```
> set.seed(2)
> n<-1000     # обсяг вибірки
> mu<-1       # математичне сподівання
> sigma0<-0.5 # стандартне відхилення
```

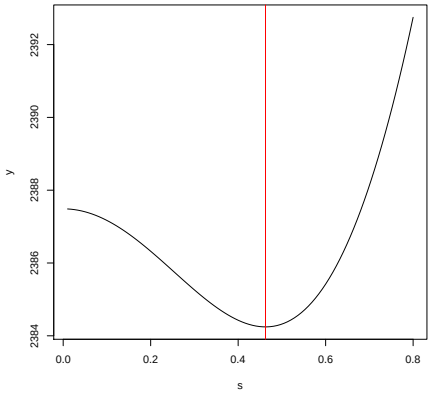

Рис. 7.1: Перетворена функція вірогідності для даних з похибками

```
> # стандартні відхилення похибок:
> sigm<-seq(1,3,length.out = n)
> # генерація даних:
> x<-rnorm(n,mu,sigma0)+sigm*rnorm(n)
> res=EstMLEGauss(x,sigm) # підрахунок оцінки
> res # значення оцінок для мат.спод. та ст. відх.:

[1]  1.0718504 -0.4625247

> # графік функції r(s):
> s<-seq(0.01,0.8,length.out=100)
> y<-sapply(s,ll,x=x,sigm=sigm)
> plot(s,y,type="l")
> abline(v=abs(res[2]),col="red")
```

Графік функції $r(s)$ для цього прикладу зображений на рис. 7.1. На ньому червоною лінією відмічено знайдене нами положення точки мінімуму — оцінки $\hat{\sigma}_n^{MLE} = -0.4625247$.

Але ж вона від'ємна? Так, насправді ми всюди при оцінці використовували не s, а s^2, тому, якщо s, точка мінімуму $r(s)$, то і $-s$ — так само. Тому алгоритм наближеного пошуку може знайти або додатну, або від'ємну точку. Якщо потрібне саме додатне значення, не забудьте взяти модуль від оцінки.

Оцінка μ за методом найбільшої вірогідності у цьому прикладі дорівнює $\hat{\mu}_n^{MLE} = 1.0718504$.

Приклад 4. Тепер розглянемо оцінку методу найбільшої вірогідності для інтенсивності λ зрізаного експоненційного розподілу з прикладу 4 розділу 7.1.

Логарифм функції вірогідності у цій задачі має вигляд

$$l(\lambda) = n(\log(\lambda) - \log(1 - e^{-C\lambda}) - \lambda\bar{\xi}).$$

Підрахунок оцінки можна організувати аналогічно тому, як це зроблено у прикладі 3:

```
> # функція ll рахує - log(вірогідність) з точністю
> # до константи. Mx - вибіркове середнє,
> # trun - поріг зрізання експоненційного розподілу
> ll<-function(l,Mx,trun){
+  -log(l/(1-exp(-l*trun)))+l*Mx
+ }
> # функція EstMLtr рахує оцінку lambda за даними x
> # методом найбільшої вірогідності
> EstMLEtr<-function(x,trun) {
+  Mx<-mean(x)
+  nlm(ll,1/Mx,Mx=Mx,trun=trun)$estimate
+ }
```

Підрахувавши цю оцінку на тих же даних, на яких перевірялась робота моментних оцінок $\hat{\lambda}_n^{MM}$, можна пересвідчитись, що значення оцінок співпадають з точністю до округлення. І дійсно, продиференціювавши функцію вірогідності та прирівнявши її до 0 для знаходження екстремуму, отримуємо в точності моментне рівняння для $\hat{\lambda}_n^{MM}$. Таким чином, ми фактично отримали дві алгоритмічні реалізації однієї і тієї ж оцінки: в першому випадку за допомогою чисельного розв'язування нелінійного рівняння, у другому — з використанням чисельної нелінійної оптимізації. Яка з цих реалізацій виявиться кращою (більш швидкодійною, більш стабільною, більш точною) залежить від того, як запрограмовані відповідні методи розв'язку рівнянь та мінімізації.

7.4 Асимптотична нормальність і матриця розсіювання оцінок

SecAsNorm

У попередніх підрозділах описано три способи побудови оцінок невідомих параметрів. Їх застосування, як ми бачили, приводить до різних, взагалі кажучи, оцінок. Наприклад, для оцінювання інтенсивності λ експоненційного розподілу ми отримали три різних оцінки: оцінку на основі першого моменту $\hat{\lambda}_n^{(1)}$ (вона також є оцінкою найбільшої вірогідності), оцінку на основі другого моменту $\hat{\lambda}_n^{(2)}$ та медіанну оцінку $\hat{\lambda}_n^{med}$.

Яка з цих оцінок краща? Поки що, ми можемо стверджувати лише, що медіанна оцінка є робастною, а моментні — ні. Інакше кажучи, якщо дані забруднені спостереженнями, що мають не такий розподіл, як основна маса, на моментні оцінки покладатись не варто, а медіанна може давати більш відповідний результат.

А яка з цих оцінок точніша, якщо наша модель повністю відповідає даним? Для того, щоб відповісти на це запитання можна провести комп'ютерний експеримент: згенерувати дані із заданим розподілом, підрахувати різні оцінки і порівняти їх із справжнім значенням параметра. Зрозуміло, що за одним набором випадкових даних результат буде один, за іншим — інший. Тому в експерименті потрібно згенерувати багато різних наборів даних з одним і тим же розподілом, по кожному набору підрахувати всі оцінки, які порівнюються. Після цього можна порівнювати розподіли отриманих оцінок: які з них мають більший розкид навколо середнього, і наскільки середнє оцінок відхиляється від оцінюваного параметра.

Проведення таких експериментів є нині практично обов'язковим елементом розробки нових алгоритмів статистичного оцінювання. Але, звичайно, у такий спосіб неможливо перевірити роботу оцінок для всіх можливих значень оцінюваних параметрів.

Виявляється, що задача теоретичного порівняння оцінок часто значно спрощується, якщо розглядати їх поведінку при нескінченному зростанні обсягу даних. Часто при цьому оцінки виявляються асимптотично нормальними, тобто їх розподіл стає близьким до нормального розподілу з нульовим середнім. Оскільки такий розподіл в одновимірному випадку характеризується одним числом — дисперсією, то і порівнювати різні асимптотично нормальні оцінки можна лише за цією дисперсією — коефіцієнтом розсіювання.

Опишемо цей підхід більш детально, розглядаючи одразу випадок d-вимірного невідомого параметра $\vartheta = (\vartheta_1, \dots, \vartheta_d)^T \in \Theta \subseteq \mathbb{R}^d$ та відповідної консистентної оцінки $\hat{\vartheta}_n = (\vartheta_{1n}, \dots, \vartheta_{dn})^T$. З консистентності оцінки випливає збіжність $\hat{\vartheta}_n - \vartheta \to 0$ (за ймовірністю) коли $n \to \infty$. Для характеризації точності оцінки важливо знати, як швидко ця різниця прямує до 0. Швидкість збіжності досліджують домножаючи $\hat{\vartheta}_n - \vartheta$ на нормуючу послідовність a_n, що прямує до нескінченності. Цю послідовність підбирають так, щоб $a_n(\hat{\vartheta}_n - \vartheta)$ прямувало і не до 0, і не до нескінченності, а до деякого проміжного значення.

Виявляється, що, за досить широких умов, при правильному виборі нормування, розподіл такої нормованої різниці прямує до нормального з нульовим математичним сподіванням — $N(0, \mathbf{V}_{\hat{\vartheta}}(\vartheta))$. Тут $\mathbf{V}_{\hat{\vartheta}}(\vartheta)$ — коваріаційна матриця граничного нормального розподілу, що залежить від справжнього значення невідомого параметра ϑ. Цю матрицю називають **матрицею розсіювання** оцінки $\hat{\vartheta}_n$.

У одновимірному випадку $d = 1$, коли оцінюваний параметр це одне число, матриця розсіювання теж складається з одного елемента — дисперсії $v_{\hat{\vartheta}}(\vartheta)$ граничного нормального розподілу нормованої оцінки. Це число звуть **коефіцієнтом розсіювання**.

Отже в одновимірному випадку, зі збіжності $\sqrt{n}(\vartheta_n - \vartheta)$ до $N(0, v_{\hat{\vartheta}}(\vartheta))$ випливає, що для будь-якого $\lambda > 0$,

$$\boxed{\text{EqConvNorm}} \quad \mathsf{P}\left\{ \frac{|\sqrt{n}(\hat{\vartheta}_n - \vartheta)}{\sqrt{v_{\hat{\vartheta}}(\vartheta)}} \le \lambda \right\} \to \mathsf{P}\{|\zeta| \le \lambda\} = 1 - 2\Phi(-\lambda), \tag{7.6}$$

де $\zeta \sim N(0,1)$, Φ — функція розподілу $N(0,1)$. Поклавши $\lambda_\alpha = Q^\Phi(1-\alpha)$, отримуємо

$$\boxed{\text{EqProbabConc}} \quad \mathsf{P}\left\{ |\hat{\vartheta}_n - \vartheta| \le \frac{\sqrt{v_{\hat{\vartheta}}(\vartheta)}\lambda_{\alpha/2}}{\sqrt{n}} \right\} = 1 - \alpha. \tag{7.7}$$

Таким чином, при великих обсягах вибірки ширина інтервалу, у який відхилення оцінки від оцінюваного значення попадає із заданою ймовірністю $1 - \alpha$, прямо пропорційна $\sqrt{v_{\hat{\vartheta}}(\vartheta)}$ (для всіх $\alpha > 0$). Тому точність асимпотично нормальних оцінок прийнято характеризувати за допомогою коефіцієнта розсіювання: чим він менший, тим оцінка точніша.

У багатовимірному випадку також, чим "менша" матриця $\mathbf{V}_{\hat{\vartheta}}(\vartheta)$, тим оцінка $\hat{\vartheta}_n$ — точніша. Порівняння матриць тут робиться у розумінні

Льовнера: $\mathbf{A} < \mathbf{B}$ рівносильно тому, що $\mathbf{B} - \mathbf{A}$ є невід'ємно визначеною матрицею.

З'ясуємо тепер, як обчислювати матриці розсіювання. Розглянемо випадок, коли дані являють собою кратну вибірку $\mathbf{X} = (\xi_1, \dots, \xi_n)$, невідомий параметр $\vartheta = (\vartheta_1, \dots, \vartheta_d)$ є d-вимірним і його оцінка $\hat{\vartheta}_n = (\vartheta_{1n}, \dots, \vartheta_{dn})$ — також.

Матриця розсіювання моментної оцінки. Нехай моментна функція має вигляд $\mathbf{h}(\xi) = (h_1(\xi), \dots, h_d(\xi))^T$, і вектор теоретичних моментів —

$$\mathbf{H}(\mathbf{t}) = (H_1(\mathbf{t}), \dots, H_d(\mathbf{t}))^T = \mathsf{E}_{\mathbf{t}}\, \mathbf{h}(\xi_1), \mathbf{t} = (t_1, \dots, t_d)^T \in \Theta.$$

Позначимо $\mathbf{H}(\mathbf{t})$ матрицю перших похідних від $\mathbf{H}(\mathbf{t})$:

$$\mathbf{H}'(\mathbf{t}) = \frac{\partial}{\partial \mathbf{t}^T}\mathbf{H}(\mathbf{t}) = \begin{pmatrix} \frac{\partial H_1(\mathbf{t})}{\partial t_1} & \dots & \frac{\partial H_1(\mathbf{t})}{\partial t_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial H_d(\mathbf{t})}{\partial t_1} & \dots & \frac{\partial H_d(\mathbf{t})}{\partial t_d} \end{pmatrix}$$

Якщо елементи коваріаційної матриці $\mathbf{D}_\vartheta = \operatorname{cov}(\mathbf{h}(\xi_1))$ є скінченними, існує обернена функція $\mathbf{H}^{-1}$ і функція $\mathbf{H}'(\mathbf{t})$ є неперервною по $\mathbf{t}$ у деякому околі ϑ, то моментна оцінка $\hat{\vartheta}_n$, яка задовольняє рівняння $\mathbf{H}(\hat{\vartheta}_n) = \hat{\mathbf{h}}_n$ є асимптотично нормальною з матрицею розсіювання

EqDispMatrMM
$$\mathbf{V}_{\hat{\vartheta}}(\vartheta) = (\mathbf{H}'(\vartheta))^{-T}\mathbf{D}_\vartheta(\mathbf{H}'(\vartheta))^{-1}. \tag{7.8}$$

У одновимірному випадку ця формула перетворюється на

EqDisp1MM
$$v_{\hat{\vartheta}} = \frac{\mathsf{D}_\vartheta\, h(\xi_1)}{(H'(\vartheta))^2}. \tag{7.9}$$

З'ясуємо, звідки взялась ця формула. Замінимо моментне рівняння його наближенням, використовуючи розклад $\mathbf{H}$ за формулою Тейлора в околі точки ϑ:

$$\mathbf{H}(\vartheta) + \mathbf{H}'(\tau)(\hat{\vartheta}_n - \vartheta) = \hat{\mathbf{h}}_n,$$

де τ — проміжна точка між ϑ і $\hat{\vartheta}_n$. Враховуючи, що $\mathbf{H}(\vartheta) = \mathsf{E}\,\hat{\mathbf{h}}_n$, отримуємо

EqAsNorm1
$$\sqrt{n}(\hat{\vartheta}_n - \vartheta) = (\mathbf{H}(\tau))^{-1}\sqrt{n}(\hat{h}_n - \mathsf{E}\,\hat{h}_n). \tag{7.10}$$

За центральною граничною теоремою, розподіл $\sqrt{n}(\hat{h}_n - \mathsf{E}\,\hat{h}_n)$ збігається до розподілу випадкового вектора $\zeta \sim N(0, \mathbf{D}_\vartheta)$. Враховуючи неперервність $\mathbf{H}'(\mathbf{t})$, отримуємо звідси формулу (7.8).

Матриця розсіювання оцінки найбільшої вірогідності. Нехай розподіл спостережень має щільність $f_\vartheta(\mathbf{x})$ відносно деякої міри μ. Позначимо

$$\mathbf{I}(\vartheta) = \mathsf{E}_\vartheta \frac{\partial}{\partial\vartheta} \ln f_\vartheta(\xi_1) \left(\frac{\partial}{\partial\vartheta} \ln f_\vartheta(\xi_1)\right)^T$$

$$\left(\int \frac{\frac{\partial}{\partial\vartheta_i} f_\vartheta(\mathbf{x}) \frac{\partial}{\partial\vartheta_k} f_\vartheta(\mathbf{x})}{f_\vartheta(x)} \mu(dx)\right)_{i,k=1}^{d}$$

— інформаційна матриця Фішера для параметра ϑ за одним спостереженням ξ_1.

Міркування, подібні розглянутим для моментних оцінок, приводять до наступної формули для матриці розсіювання оцінок методу найбільшої вірогідності $\hat\vartheta_n$:

$$\mathbf{V}_{\hat\vartheta}(\vartheta) = (\mathbf{I}(\vartheta))^{-1}$$

— матриця розсіювання є матрицею, оберненою до інформаційної.

У одновиміронму випадку для коефіцієнта розсіювання отримуємо:

$$v_{\hat\vartheta}(\vartheta) = \frac{1}{I(\vartheta)},$$

де

$$I(\vartheta) = \int \frac{\left(\frac{\partial}{\partial\vartheta} f_\vartheta(x)\right)^2}{f_\vartheta(x)} \mu(dx).$$

Матриця розсіювання для квантильних оцінок. Нехай знову, дані являють собою кратну вибірку $\mathbf{X}$ випадкових величин ξ_j з функцією розподілу F_ϑ та щільністю $f_\vartheta(x)$, $\vartheta \in \Theta \in \mathbb{R}^d$. Зафіксуємо набір рівнів $\alpha = (\alpha_1, \ldots, \alpha_d)$, $0 < \alpha_i < 1$. Позначимо $\mathbf{q}^\alpha(\vartheta) = (Q^{F_\vartheta}(\alpha_1), \ldots, Q^{F_\vartheta}(\alpha_1))$ — вектор теоретичних квантилей, $\hat{\mathbf{q}}_n^\alpha = (Q^{\mathbf{X}}(\alpha_1), \ldots, Q^{\mathbf{X}}(\alpha_1))$ — набір емпіричних квантилей. Нехай для всіх α_i, $f_\vartheta(Q^{F_\vartheta}(\alpha_i) > 0$. Тоді з наслідку 1 п. 7 гл. 1 [1] випливає, що $\sqrt{n}(\hat{\mathbf{q}}_n^\alpha - \mathbf{q}^\alpha(\vartheta))$ збігається за розподілом до $N(0, \mathbf{C})$, де $\mathbf{C} = (c_{i,k})_{i,k=1}^d$,

$$\boxed{\text{EqCorQuant}} \quad c_{ik} = \frac{\min(\alpha_i, \alpha_k) - \alpha_i\alpha_k}{f_\vartheta(Q^{F_\vartheta}(\alpha_i)) f_\vartheta(Q^{F_\vartheta}(\alpha_k))}. \tag{7.11}$$

Нехай квантильна оцінка ϑ_n^α для ϑ є розв'язком рівняння

$$\mathbf{q}^\alpha(\mathbf{t}) = \hat{\mathbf{q}}_n^\alpha$$

відносно $\mathbf{t}$.

Тоді міркування, аналогічні до тих, які ми використали для моментних оцінок, приводять до наступного виразу для матриці розсіювання оцінки $\hat{\vartheta}_n^{\alpha}$:

$$\boxed{\text{EqDispQuant}} \quad \mathbf{V}_{\hat{\vartheta}^{\alpha}}(\vartheta) = \mathbf{Q}^{-T}\mathbf{C}\mathbf{Q}^{-1}, \tag{7.12}$$

де $\mathbf{Q} = \frac{\partial}{\partial \vartheta^T}\mathbf{q}^{\alpha}(\vartheta)$.

Зокрема, для медіанної оцінки ϑ_n^{med}, що є розв'язком рівняння

$$q^{1/2}(t) = \operatorname{med}(X),$$

коефіцієнт розсіювання дорівнює

$$\boxed{\text{EqDispMed}} \quad v_{\vartheta^{med}} = \frac{1}{4(f_{\vartheta}(\operatorname{med}(\xi_1))(q^{1/2}(\vartheta))')^2}. \tag{7.13}$$

Подивимось також, як записати коефіцієнт розсіювання квантильної оцінки, якщо вона визначається як розв'язок рівняння (7.5). Формулу, яку ми отримаємо, можна вивести з (7.12), але ми зробимо це безпосередньо.

Отже, нехай для оцінки $\hat{\vartheta}_n$ виконується рівняння

$$F_{\hat{\vartheta}_n}(\hat{q}_n) = \alpha,$$

де $\hat{q}_n = Q^{\mathbf{X}}(\alpha)$. В умовах, що вказані вище, $\sqrt{n}(\hat{q}_n - q_{\alpha})$ збігається за розподілом до $N(0, c)$, де $q_{\alpha} = Q^{F_{\vartheta}}(\alpha)$, $c = \alpha(1-\alpha)/f_{\vartheta}(q_{\alpha})$.

Розкладаючи ліву частину цієї рівності в околі точки (ϑ, q_{α}), отримуємо

$$F_{\vartheta}(q_{\alpha}) + \frac{\partial}{\partial t}F_t(q)(\hat{\vartheta}_n - \vartheta) + \frac{\partial}{\partial q}F_t(q)(\hat{q} - q_{\alpha}) = \alpha,$$

де t — проміжна точка між $\hat{\vartheta}_n$ і ϑ, q — проміжна точка між $\hat{q}_n$ і q_{α}. Звідси отримуємо

$$\sqrt{n}(\hat{\vartheta}_n - \vartheta) \sim \frac{\frac{\partial}{\partial q}F_{\vartheta}(q_{\alpha})}{\frac{\partial}{\partial \vartheta}F_{\vartheta}(q_{\alpha})}\sqrt{n}(\hat{q}_n - q_{\alpha}).$$

Використовуючи асимптотичну нормальність $\hat{q}_n$, отримуємо коефіцієнт розсіювання $\hat{\vartheta}$:

$$\boxed{\text{EqQuantVi}} \quad v_{\hat{\vartheta}_n} = \left(\frac{\frac{\partial}{\partial q}F_{\vartheta}(q_{\alpha})}{\frac{\partial}{\partial \vartheta}F_{\vartheta}(q_{\alpha})}\right)^2 \frac{\alpha(1-\alpha)}{(f_{\vartheta}(q_{\alpha}))^2} \tag{7.14}$$

Приклад 1. Повернемося до розгляду задачі оцінки інтенсивності λ експоненційного розподілу за кратною вибіркою $\mathbf{X} = (\xi_1, \ldots, \xi_n)$. У попередніх розділах були введені три оцінки:

$$\hat{\lambda}_n^{(1)} = 1/\bar{\xi},\ \hat{\lambda}_n^{(2)} = \sqrt{\frac{2n}{\sum_{j=1}^n \xi_j^2}},\ \hat{\lambda}_n^{\mathrm{med}} = \frac{\log 2}{\mathrm{med}(X)}.$$

Перші дві оцінки отримані методом моментів з моментними функціями $h_1(x) = x$ та $h_2(x) = x^2$. Враховуючи, що

$$\mathsf{D}_\lambda\, h_1(\xi_1) = \frac{1}{\lambda^2},\ \mathsf{D}_\lambda\, h_2(\xi_1) = \frac{23}{\lambda^4},$$

за (7.9) отримуємо коефіцієнти розсіювання цих оцінок:

$$v_{\hat{\lambda}^{(1)}} = \lambda^2, v_{\hat{\lambda}^{(2)}} = \frac{23}{16}\lambda^2.$$

Третя оцінка — медіанна. Теоретична медіана експоненційного розподілу $\mathrm{med}(\xi_1) = \log 2/\lambda$, а щільність розподілу у медіані — $f_\lambda(\mathrm{med}(\xi_1)) = \lambda/2$. Тому коефіцієнт розсіювання цієї оцінки

$$v_{\hat{\lambda}^{med}} = \frac{\lambda^2}{(\log 2)^2}.$$

Оскільки $23/16 \approx 1.4375 < 2.08137 \approx 1/(\log 2)^2$, ці результати показують, що найбільш точною при великих обсягах вибірок є оцінка $\hat{\lambda}_n^{(1)}$, наступною — $\hat{\lambda}_n^{(2)}$, а найменш точною з трьох розглянутих є медіанна оцінка.

Відношення коефіцієнтів варіації двох різних оцінок одного параметра називають їх **відносною асиптотичною ефективністю** (asymptotic relative efficiency, ARE). наприклад, $v_{\hat{\lambda}^{med}}/v_{\hat{\lambda}^{(1)}} = 1/(\log 2)^2 \approx 2.08137$ — ARE оцінки найбільшої вірогідності порівняно з медіанною оцінкою. ARE має простий статистичний зміст, який легко зрозуміти враховуючи (7.7). Якщо ми раніше користувались оцінкою $\hat{\lambda}_n^{(1)}$, а тепер замість неї хочемо використати $\hat{\lambda}_n^{(med)}$, то для забезпечення такої ж точності як і раніше, нам прийдеться збільшити обсяг вибірки у два (точніше у 2.08137) рази. Це варто робити, якщо вигоди від робастності медіанної оцінки перевищують додаткові витрати на збільшення обсягу спостережень. Інакше слід використовувати оцінку найбільшої вірогідності.

Те, що найкращою виявиться $\hat{\lambda}_n^{(1)}$ можна було сказати вже тоді, коли виявилось, що це оцінка найбільшої вірогідності. Справа в тому, що при виконанні досить широких умов[2] ОНВ є асимптотично нормальними

[2]умов регулярності

оцінками з коефіцієнтом розсіювання найменшим серед всіх "правильних" (т. зв. регулярних) оцінок.

Приклад 2. Нехай тепер оцінюються математичне сподівання μ і дисперсія σ^2 за кратною вибіркою гауссових спостережень $\mathbf{X}$. Ми отримали по дві оцінки для кожного параметра: метод моментів дав той же результат, що і метод найбільшої вірогідності —

$$\hat{\mu}^{MLE} = \bar{\xi},\ \hat{\sigma}_n^{2\ MLE} = S^2(\mathbf{X}),$$

а метод квантилів —

$$\hat{\mu}_n^{med} = \operatorname{med}\mathbf{X},\ \hat{\sigma}^{2\ IQ} = \left(\frac{Q^{\mathbf{X}}(3/4) - Q^{\mathbf{X}}(1/4)}{2\lambda_{\alpha/4}}\right)^2.$$

Для підрахунку матриці розсіювання оцінок найбільшої вірогідності знайдемо інформаційну матрицю для $\vartheta = (\mu, \sigma^2)^T$. Легко бачити, що[3]

$$\frac{\partial}{\partial\mu} f_\vartheta(\xi_1) = \frac{\xi_1 - \mu}{\sigma^2}\ \frac{\partial}{\partial\sigma^2} f_\vartheta(\xi_1) = -\frac{1}{2\sigma^2} - \frac{(\xi_1-\mu)^2}{2\sigma^4}.$$

Отже інформаційна матриця для одного спостереження має вигляд

$$\mathbf{I}(\vartheta) = \mathsf{E}\begin{pmatrix} \frac{(\xi_1-\mu)^2}{\sigma^4} & \frac{\xi_1-\mu}{2\sigma^4} + \frac{(\xi_1-\mu)^3}{2\sigma^6} \\ \frac{\xi_1-\mu}{2\sigma^4} + \frac{(\xi_1-\mu)^3}{2\sigma^6} & \frac{((\xi_1-\mu)^2-\sigma^2)^2}{4\sigma^8} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{2\sigma^4} \end{pmatrix}.$$

Таким чином, матриця розсіювання оцінок найбільшої вірогідності

$$\mathbf{V}_{\hat{\vartheta}^{MLE}}(\vartheta) = \mathbf{I}^{-1}(\vartheta) = \begin{pmatrix} \sigma^2 & 0 \\ 0 & 2\sigma^4 \end{pmatrix}.$$

Ми отримали, що коефіцієнт розсіювання $\hat{\mu}_n^{MLE}$ дорівнює σ^2, а коефіцієнт розсіювання $\hat{\sigma}_n^{2\ MLE}$ — $2\sigma^4$. Ці оцінки є асимптотично некорельованими.

Підрахуємо коефіцієнти розсіювання квантильних оцінок. Для $\hat{\mu}_n^{med}$ це можна зробити безпосередньо за формулою (7.13):

$$v_{\hat{\mu}^{med}} = \frac{1}{4(f_\vartheta(\mu))^2} = \frac{\pi\sigma^2}{2}.$$

[3]Тут $f_\vartheta(x)$ — щільність нормального розподілу з параметрами μ, σ^2, причому диференціюючи по σ^2 слід розуміти це як єдиний символ, а не як квадрат σ.

Для

$$\hat{\sigma}_n^{2\ IQ}$$

підрахунок дещо складніший. Почнемо з визначення граничної коваріаційної матриці для вектора $\mathbf{z}_n = (z_n^1, z_n^2)^T = \sqrt{n}(\hat{\mathbf{q}}_n - \mathbf{q})$, де $\hat{\mathbf{q}}_n = (Q^{\mathbf{X}}(1/4), Q^{\mathbf{X}}(3/4))^T$, $\mathbf{q} = (Q^{N(\mu,\sigma^2)}(1/4), Q^{N(\mu,\sigma^2)}(3/4))^T$. За (7.11) коваріаційна матриця розподілу двовимірного нормального вектора $\mathbf{z}$ до якого збігається розподіл $\mathbf{z}_n$ дорівнює

$$\mathbf{C} = (c_{ik})_{i,k=1}^2 \frac{1}{(f_\vartheta(\mu + \sigma\lambda_{1/4}))^2} \begin{pmatrix} \frac{3}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{3}{16} \end{pmatrix}.$$

Звідси отримуємо, nщо послідовність $\tilde{z}_n = (z_n^1 - z_n^2)/(2\lambda_{1/4})$ також є асимптотично нормальною з асимптотичною дисперсією

$$\tilde{c} = \frac{1}{(2\lambda_{1/4})^2}(c_{11} - 2c_{12} + c_{22}) = \frac{\pi e^{-\lambda_{1/4}^2}\sigma^2}{8\lambda_{1/4}^2}.$$

Оскільки при великих n

$$\sqrt{n}(\hat{\sigma}_n^{2\ IQ} - \sigma^2) \sim 2\sigma\tilde{z},$$

то

$$v_{\hat{\sigma}^2\ IQ} = 4\sigma^2\tilde{c} = \frac{\pi e^{-\lambda_{1/4}^2}\sigma^4}{2\lambda_{1/4}^2} \approx 5.44184\sigma^4.$$

Таким чином, відносна асимптотична ефективність оцінки найбільшої вірогідності для μ порівняно з медіанною

$$v_{\hat{\mu}^{med}}/v_{\hat{\mu}^{MLE}} = \pi/2 \approx 1.5708.$$

Для ОНВ оцінки дисперсії порівняно з квартильною відносна асимптотична ефективність

$$v_{\hat{\sigma}^2\ IQ}/v_{\hat{\sigma}^2\ MLE} = \frac{\pi e^{-\lambda_{1/4}^2}}{4\lambda_{1/4}^2} \approx 2.72092.$$

Тобто при використанні квантильної оцінки потрібно у 2.72 рази більше спостережень ніж при використанні звичайної вибірокової дисперсії для досягнення однакової точності оцінювання.

Зрозуміло, що вся ця асимптотична теорія працює лише при достатньо великих обсягах вибірки. Наскільки великих? Якою буде ситуація для невеликих обсягів? Щоб відповісти на такі запитання, проводять спеціальні імітаційні експерименти (simulation study). Подивимось, як це може виглядати у нашому прикладі.

Ми згенеруємо `B=1000` різних вибірок з одним і тим же нормальним розподілом з параметрами `mu=1` (математичне сподівання) і `sigma=1` (стандартне відхилення). По кожній вибірці будуть підраховані чотири оцінки, які вміщуються у масиви оцінок —

$\hat{\mu}_n^{MLE}$ у `EstMuMom`, $\hat{\mu}_n^{med}$ у `EstMuMed`, $\hat{\sigma}_n^{2\ MLE}$ у `EstSMom`, $\hat{\sigma}_n^{2\ IQ}$ у `EstSMed`.

По кожному з цих масивів ми рахуємо вибіркове середнє, що має бути наближенням для математичного сподівання відповідної оцінки і віднімаємо від нього справжнє значення оцінюваного параметру. Отримуємо приблизне значення зміщення оцінки. Це значення домножається на $\sqrt{n}$. Асимптотичний розподіл нормованої оцінки має нульове математичне сподівання, тому можна сподіватись, що при достатньо великих n таке нормоване зміщення буде близьким до 0.

Далі ми підраховуємо вибіркові дисперсії по масивах оцінок і домножаємо на n. Ця величина має приблизно дорівнювати коефіцієнту розсіювання оцінки. Якщо це не так, можна запідозрити, що наші теоретичні розрахунки не адекватні, або що обсяг вибірки недостатньо великий для застосування асимптотичної теорії.

Наведемо скрипт, що реалізує цю ідею для обсягу вибірки `n=200`.

```
> set.seed(3)
> B<-1000  # number of samples
> mu<-1    # mean
> n<-200   # sample size
> sigma<-1 # standard deviation
> EstMuMom<-numeric(B)
> EstMuMed<-numeric(B)
> EstSMom<-numeric(B)
> EstSMed<-numeric(B)
> d<-c(1,-1)
> alpha<-c(0.75,0.25)
> for(i in 1:B)
+ {
+ x<-rnorm(n,mu,sigma)
```

```
+ EstMuMom[i]<-mean(x)
+ EstMuMed[i]<-median(x)
+ EstSMom[i]<-var(x)
+ EstSMed[i]<-(sum(quantile(x,alpha)*d)/1.34898)^2
+ }
>  # Moment estimate for mean
>  (mean(EstMuMom)-mu)*sqrt(n) # bias

[1] -0.001903953

>  n*var(EstMuMom)             # dispersion

[1] 0.9389821

>  sigma^2                  # theoretical dispersion

[1] 1

>  # Median estimate for mean
>  (mean(EstMuMed)-mu)*sqrt(n) # bias

[1] -0.005031941

>  n*var(EstMuMed)             # dispersion

[1] 1.459559

>  3.1415*sigma^2/2     # theoretical dispersion

[1] 1.57075

>  # Moment estimate for variance
>  (mean(EstSMom)-sigma^2)*sqrt(n) # bias

[1] 0.05725523

>  n*var(EstSMom)                  # diaspersion

[1] 2.010686

>  2*sigma^4           # theoretical dispersion
```

```
[1] 2

> # Median estimate for variance
> (mean(EstSMed)-sigma^2)*sqrt(n) # bias

[1] 0.01135744

> n*var(EstSMed)                  # dispersion

[1] 5.561874

> 5.44184*sigma^4     # theoretical dispersion

[1] 5.44184
```

Як ми бачимо, при $n = 200$ результати імітаційного моделювання непогано (хоча і не ідеально) узгоджуються з асимптотичними формулами. Наприклад, теоретичний коефіцієнт розсіювання медіанної оцінки 1.57075, а його аналог отриманий моделюванням — 1.459559. Нормоване зміщення дорівнює -0.005031941. Це дуже мало, порівняно з дисперсією, тому зміщенням як джерелом похибки можна знехтувати і характеризувати цю оцінку лише дисперсією. (Насправді легко бачити, що медіанна оцінка у даному випадку незміщена, тобто відхилення зміщення від 0 це результат неточності нашого імітаційного експерименту).

Варто відмітити, що зі збіжності розподілів, взагалі кажучи, не випливає збіжність моментів. Тому навіть у асимптотично нормальних оцінок дисперсія нормованої оцінки при зростанні обсягу вибірки не обов'язково прямує до коефіцієнта розсіювання. Зокрема, так може бути, коли у маленькому відсотку випадків оцінка дозволяє грубі відхилення від справжнього значення параметра. Такі відхилення можуть зіграти роль викидів, що спотворюють дисперсію оцінки при скінченному обсязі вибірки.

У такому випадку доречно використати для наближення коефіцієнта розсіювання яку-небудь робастну оцінку дисперсії. Такою оцінкою може бути квантильна оцінка, якою ми тільки що скористались у нашому прикладі. Крім того, доцільно перевірити, чи дійсно розподіл вибірки з оцінок добре узгоджується з нормальним. От як може виглядати скрипт, що реалізує цю ідею:

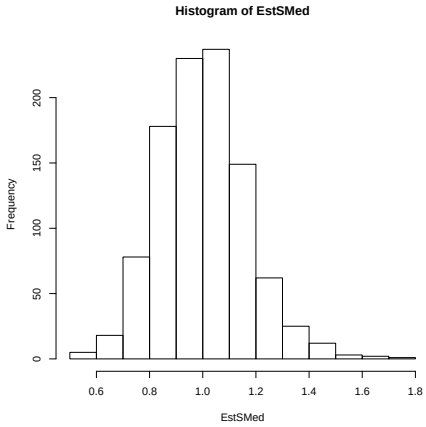

Рис. 7.2: Гістограма вибірки з оцінок

```
> # Quantile estimate for dispersion:
> (sum(quantile(EstSMed,alpha)*d)/1.34898)^2*n

[1] 5.04423

> # histogram
> hist(EstSMed)
```

Як бачимо, значення цього робастного наближення — 5.04423 помітно відрізняється як від значення нормованої вибіркової дисперсії оцінок — 5.561874, так і від теоретичного коефіцієнта розсіювання — 5.44184 (вони, доречі, досить добре узгоджуються). Це може пояснюватись тим, що, внаслідок порівняно малого обсягу вибірки n, розподіл оцінок недостатньо добре наближається нормальним. І дійсно, гістограма на рис. 7.2 вказує на помітне відхилення від нормальності, зокрема, на асиметрію розподілу. З цього можна зробити висновок, що при таких n цілком покладатись на асимптотичні формули не варто.

Приклад 3. Перейдемо до розгляду задачі оцінки параметрів нормального розподілу за спостереженнями з похибкою з прикладу з прикладу 3 п. 7.1. У підрозділі 7.3 ми розібрались, як підраховувати оцінки найбільшої вірогідності для середнього та дисперсії у цій моделі. Чи можна скористатись нашою асимптотичною теорією щоб охарактеризувати точність таких оцінок. Наприклад, нас може цікавити наскільки вона погір-

шилась, порівняно з випадком прикладу 2, коли дані спостерігались без похибок.

Безпосередньо формули для матриці розсіювання застосувати не можна, оскільки у цьому прикладі спостереження не є однаково розподіленими. Однак відомо, що асимптотична нормальність виконана і для ОНВ у багатьох моделях з не однаковим розподілом спостережень. Зокрема, за умови обмеженості дисперсій похибок, вона буде виконуватись у нашій моделі. В таких випадках матрицю розсіювання можна обчислювати використовуючи "середню інформацію на одне спостереження", тобто

$$\bar{\mathbf{I}}_n(\vartheta) = \frac{1}{n}\sum_{j=1}^{n}\mathbf{I}_j(\vartheta),$$

де $\mathbf{I}_j(\vartheta)$ — інформаційна матриця для j-того спостереження. Матриця розсіювання ОНВ дорівнює

$$\mathbf{V}_{\hat{\vartheta}^{MLE}}(\vartheta) = \lim_{n\to\infty}(\bar{\mathbf{I}}_n(\vartheta))^{-1}.$$

Для практичних наближень замість границі при $n \to \infty$ беруть значення $(\bar{\mathbf{I}}_n(\vartheta))^{-1}$ при тому обсязі вибірки n, для якого робляться розрахунки. (Зрозуміло, що n має бути достатньо великим, інакше наша асимптотична теорія працювати не буде).

Аналогічно тому, як це було зроблено у прикладі 2, отримуємо, що інформаційна матриця для j-того спостереження (воно у нашій моделі має розподіл $N(\mu, \sigma^2 + \sigma_j^2)$), має вигляд

$$\mathbf{I}_j(\vartheta) = \begin{pmatrix} \frac{1}{\sigma^2+\sigma_j^2} & 0 \\ 0 & \frac{1}{2(\sigma^2+\sigma_j^2)^2} \end{pmatrix}$$

(Тут, як і раніше, $\vartheta = (\mu, \sigma^2)$). Отже коефіцієнти розсіювання оцінок дорівнюють

$$v_{\hat{\mu}^{MLE}}(\sigma^2) = \frac{n}{\sum_{j=1}^{n}(\sigma^2+\sigma_j^2)^{-1}},$$

$$v_{\hat{\sigma}^2\ MLE}(\sigma^2) = \frac{2n}{\sum_{j=1}^{n}(\sigma^2+\sigma_j^2)^{-2}},$$

Приклад 4. Проведемо порівняння ефективності оцінок інтенсивності λ зрізаного експоненційного розподілу, отриманих у прикладі 4 попереннх

підрозділів. У п. 7.1 була побудована оцінка методу моментів (позначимо її $\hat{\lambda}^{MM}$), яка виявилась також оцінкою найбільшої вірогідності у п. 7.3. У п. .7.2 введена медіанна оцінка (позначимо її $\hat{\lambda}^{med}$. Обидві оцінки знаходяться як розв'язки відповідних рівнянь, у явному вигляді їх виразити не можна. Тим більш цікаво, що їх коефіцієнти розсіювання цілком можна знайти аналітично.

Почнемо з $\hat{\lambda}^{MM}$. Оскільки це оцінка найбільшої вірогідності, її коефіцієнт розсіювання можна знайти як $1/I(\lambda)$, де

$$I(\lambda) = \int_0^c \frac{(f'_\lambda(x))^2}{f_\lambda(x)} dx$$

— інформація за Фішером на одне спостереження,

$$f_\lambda(x) = \frac{\lambda e^{-\lambda x}}{1 - e^{-\lambda c}}$$

— щільність одного спостереження при $x \in [0, c]$,

$$f'_\lambda(x) = \frac{\partial}{\partial \lambda} f_\lambda(x) = \frac{e^{\lambda(c-x)}(\lambda(x-c) + e^{\lambda c}(\lambda x - 1) - 1)}{(e^{\lambda c} - 1)^2}.$$

Інформацію можна підрахувати звичайним інтегруванням, вона дорівнює

$$I(\lambda) = \frac{1}{\lambda^2} + \frac{c^2}{2 - (e^{\lambda c} + e^{-\lambda c})}.$$

Про всяк випадок перевіримо правильність цієї формули а заодно покажемо, як можна наближено обчислювати інтеграли в `R` у тих випадках, коли для них немає явних виразів.

У `R` для наближеного обчислення інтегралів використовується функція `integrate(f,lower,upper)`, де

`f` дійсна функція дійсного аргументу, від якої підраховується інтеграл,

`lower`, `upper` — нижня та верхня межі інтегрування.

От як можна використати її для перевірки результату інтегрування у нашому випадку:

```
>  l<-0.5 # інтенсивність
> U<-1   # поріг зрізання
> # щільність розподілу
```

```
> f<-function(x,l,U){l*exp(-l*x)/(1-exp(-l*U))}
> # похідна розподілу за l
> fp<-function(x,l,U){exp(l*(U-x))*(l*(x-U)
+    +exp(l*U)*(-l*x+1)-1)/(exp(l*U)-1)^2}
> # підінтегральна функція для інформації
> g<-function(x){(fp(x,l,U))^2/f(x,l,U)}
> # аналітичний вираз для інформації
> inf<-function(l,U){1/l^2 +U^2/(2-(exp(U*l)+exp(-U*l)))}
> #
> inf(l,U) # інформація за формулою

[1] 0.08230191

> integrate(g,0,U) # наближений інтеграл для інформації

0.08230191 with absolute error < 9.1e-16
```

Як бачимо, значення інформації за нашою формулою та значення, отримане наближеним інтегруванням однакові. Отже при $\lambda = 0.5$, $c = 1$, коефіцієнт розсіювання моментної оцінки $v_{\hat{\lambda}^{MM}}(\lambda) = 1/I(\lambda) = 1/0.08230191 = 12.15039$.

Тепер підрахуємо коефіцієнт розсіювання медіанної оцінки. Для цього скористаємось формулою (7.14). Помітимо, що у нашому випадку $\alpha = 1/2$ медіана

$$q_\alpha = -\frac{1}{l}(\log(1 + e^{-cl}) - \ln 2),$$

$$\frac{\frac{\partial}{\partial x}F_\lambda(x)}{\frac{\partial}{\partial \lambda}F_\lambda(x)} = \frac{\lambda(e^{\lambda c} - 1)}{c(1 - e^{\lambda x}) + x(e^{\lambda c} - 1).}$$

Підставляючи це у формулу (7.14), отримуємо коефіцієнт розсіювання $\hat{\lambda}^{med}$:

$$v_{\hat{\lambda}^{med}}(\lambda) = \frac{l^2(e^{\lambda c} - 1)^2}{(\lambda c + (e^{\lambda c} + 1)\log((e^{-\lambda c} + 1)/2))^2}.$$

Підставляючи, як і у попередньому прикладі, значення $\lambda = 0.5$, $c = 1$, отримуємо $v_{\hat{\lambda}^{med}}(\lambda) = 16.3344$

Отже, при цих значеннях параметрів відносна асимптотична ефективність оцінки найбільшої вірогідності по відношенню до медіанної складає $16.3344/12.15039 = 1.34435$. При використанні медіанних оцінок потрібно використовувати вибірки на 34% більші порівняно з вибірками

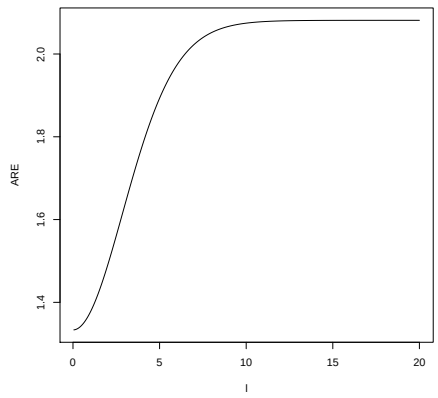

Рис. 7.3: Гістограма вибірки з оцінок

для оцінок найбільшої вірогідності, якщо ми хочемо забезпечити однакову точність оцінювання.

На відміну від прикладів 1 і 2 відносна асимптотична ефективність (ARE) медіанної оцінки та ОНВ залежить тепер від невідомого параметра λ. Ми можемо подивитись на графіку, як вона змінюється при різних λ (рис.7.3):

```
> vmm<-function(l){1/inf(l,U)}
> # коефіцієнт розсіювання для медіанної оцінки
> vmed<-function(l){
+ (l*(exp(l*U)-1)/
+    (l*U+(1+exp(l*U))*log((1+exp(-l*U))/2)))^2
+ }
> l<-(1:400)/20
> ARE<-sapply(l,vmed)/sapply(l,vmm)
> plot(l,ARE,type="l")


> vmed(0.01)/vmm(0.01)

[1] 1.333337

> vmed(20)/vmm(20)

[1] 2.081368
```

Як бачимо, ARE зростає із зростанням λ від 1.333 до 2.081368. Тобто у найгіршому випадку (при дуже великих λ медіанна оцінка вимагає вдвічі більше спостережень ніж ОНВ для забезпечення еквівалентної точності оцінювання. Цікаво, що такий самий ефект ми отримали і для експоненційного розподілу без зрізання, тільки там така ARE була для всіх можливих значень λ.

7.5 Довірчі інтервали

Зрозуміло, що оцінка $\hat{\vartheta}_n$, побудована за випадковими даними $\mathbf{X}$, як правило, не дорівнює справжньому значенню невідомого параметра ϑ. Як далеко може знаходитись ϑ від його оцінки? Щоб охарактеризувати область можливих значень одновимірного параметра використовують техніку довірчих інтервалів. А саме, замість однієї оцінки $\hat{\vartheta}_n$ використовують пару статистик[4] ϑ_n^-, ϑ_n^+, таких, що $\vartheta_n^- < \vartheta_n^+$ і

$$\boxed{\text{EqConfInt1}} \quad \mathsf{P}\{\vartheta \in [\vartheta_n^-, \vartheta_n^+]\} = 1 - \alpha, \tag{7.15}$$

де α — задане статистиком мале число, яке звуть рівнем значущості.

Таким чином, довірчий інтервал це інтервал, побудований за спостережуваними даними, який покриває невідомий параметр із заданою ймовірністю $1 - \alpha$.

Якщо (7.15) виконується точно для заданого обсягу даних n і всіх $\vartheta \in \Theta \subseteq \mathbb{R}$, то $[\vartheta_n^-, \vartheta_n^+]$ називають точним (або строгим) довірчим інтервалом. Якщо рівність у (7.15) досягається лише асимптотично, тобто

$$\boxed{\text{EqConfInt1a}} \quad \lim_{n\to\infty} \mathsf{P}\{\vartheta \in [\vartheta_n^-, \vartheta_n^+]\} = 1 - \alpha, \tag{7.16}$$

то довірчий інтервал називають асимптотичним. Нарешті, якщо виконується нерівність

$$\mathsf{P}\{\vartheta \in [\vartheta_n^-, \vartheta_n^+]\} \geq 1 - \alpha,$$

довірчий інтервал називають нестрогим.

Теорія асимптотичної нормальності дозволяє будувати асимптотичні довірчі інтервали з використанням коефіцієнтів розсіювання оцінок. Дійсно, нехай для невідомого параметра ϑ існує асимптотично нормальна оцінка $\hat{\vartheta}_n$ з коефіцієнтом розсіювання $v(\vartheta) = v_{\hat{\vartheta}_n}(\vartheta)$. Припустимо, що

[4] Тобто вимірних функцій від даних $\mathbf{X}$.

$v(\vartheta)$ є неперервною функцією $\vartheta \in \Theta$. Тоді $v(\vartheta)/v(\hat{\vartheta}_n) \to 1$ при $n \to \infty$ (за ймовірністю) і з (7.7) випливає, що[5]

$$\boxed{\text{EqConfInt2}} \quad \lim_{n\to\infty} \mathsf{P}\left\{ \frac{\sqrt{n}|\hat{\vartheta}_n - \vartheta|}{\sqrt{v(\hat{\vartheta}_n)}} \le \lambda_{\alpha/2} \right\} = 1 - \alpha. \tag{7.17}$$

Покладемо

$$\boxed{\text{EqConfIntAss}} \quad \vartheta_n^{\pm} = \hat{\vartheta}_n \pm \lambda_{\alpha/2} \sqrt{\frac{v(\hat{\vartheta}_n)}{n}}. \tag{7.18}$$

Оскільки (7.17) еквівалентно $\lim_{n\to\infty}\{\vartheta \in [\vartheta_n^-, \vartheta_n^+]\} = 1 - \alpha$, то $[\vartheta_n^-, \vartheta_n^+]$ є асимптотичним довірчим інтервалом з рівнем значущості $1 - \alpha$.

Відмітимо, що ця техніка дозволяє застосування і у тому випадку, коли крім параметра ϑ у розподілу даних є і інші невідомі параметри. Для побудови довірчого інтервалу потрібна лише асимптотично нормальна оцінка ϑ та консистентна оцінка його коефіцієнта розсіювання. Якщо вони визначені, то довірчий інтервал можна будувати за формулою (7.18).

Зрозуміло, що чим менше коефіцієнт розсіювання оцінки, тим вужчим буде довірчий інтервал, побудований за цією технікою. Тому при побудові довірчих інтервалів природно обирати оцінки з найменшим коефіцієнтом розсіювання, якщо немає інших важливих вимог (як от, робастність).

Приклад 1. Знову розглянемо задачу оцінки інтенсивності експоненційного розподілу λ за кратною вибіркою $\mathbf{X}$. Як ми бачили у прикладі 1 з п. 7.4, оцінкою з найменшим коефіцієнтом розсіювання для λ є $1/\bar{\xi}$, причому її коефіцієнт розсіювання $v(\lambda) = \lambda^2$.

Нехай потрібно побудувати довірчий інтервал для λ з рівнем значущості $\alpha = 0.05$. Помітимо, що[6] $\lambda_{\alpha/2} = Q^{N(0,1)}(0.975) \approx 1.96$. Отже, за

[5] Тут, як і раніше, $\lambda_\alpha = Q^{N(0.1)}(1-\alpha)$.

[6] Є певна незручність в тому, що літера λ позначає і невідомий параметр і квантиль нормального розподілу. Якщо записати формули для довірчого інтервалу у цих позначеннях, вони виглядатимуть трохи дивно. Можна було б спеціально для цього прикладу ввести якесь особливе позначення для інтенсивності або для квантилі. Але обидва позначення є стандартними і відступ від них теж заплутав би справу. Я вийшов з положення зафіксувавши $\alpha = 0.95$. Це дуже популярний рівень значущості і відповідне йому $\lambda_{\alpha/2} = 1.96$ більшість статистиків знає напам'ять. У багатьох прикладних книжках число 1.96 з'являється без пояснень, як магічна константа. Його зв'язок з рівнем значущості залишається для користувачів таємницею. Довірчий інтервал який ми отримаємо може бути прикладом таких формул.

(7.18) отримуємо межі довірчого інтервалу:

$$\lambda^- = \frac{1}{\bar{\xi}} - \frac{1.96}{\bar{\xi}\sqrt{n}},\ \lambda^+ = \frac{1}{\bar{\xi}} + \frac{1.96}{\bar{\xi}\sqrt{n}}.$$

Рівень значущості довірчого інтервалу 0.05 означає, що в середньому, на 100 задач оцінювання, у 5 випадках такий довірчий інтервал не покриє справжнє значення інтенсивності. Оскільки довірчий інтервал асимптотичний, це має бути "при достатньо великих обсягах вибірки". Спробуємо у імітаційному експерименті подивитись, наскільки точним виявиться це передбачення для вибірок помірного обсягу.

```
>  set.seed(2)
> l<-0.5      # інтенсивність експ. розподілу
> B<-10000    # кількість вибірок
> n<-100      # обсяг вибірки
> lambda<-qnorm(0.975)
> res<-numeric(B)  # масив результатів випробувань
> estm<-numeric(B) # масив оцінок
> dif<-numeric(B)  # півширини довірчих інтервалів
> for(i in 1:B)
+ {
+   x<-rexp(n,l)
+   estl<-1/mean(x)
+   res[i]<-ifelse(abs(l-estl)<lambda*estl/sqrt(n),1,0)
+ }
> err=1-mean(res) # частота помилок
> err

[1] 0.0474
```

Тут вибрано $\lambda = 0.5$, моделюється $B = 10000$ вибірок і по кожній вибірці перевіряється, чи попаде 0.5 у довірчий інтервал (точніше, чи є різниця $\hat{\lambda} - \lambda$ меншою ніж половина ширини інтервалу). Якщо для i-тої модельованої вибірки ця умова виконана, на i-тому місці у масиві результатів `res` записується 1, інакше - 0. Потім частота попадань визначається як середнє `res`.

Як бачимо, результат експерименту показує помірне узгодження з теорією — частота 0.0474 при теоретичній ймовірності 0.05. При обсязі вибірки $n = 1000$ цей же скрипт дасть 0.0505 — чудова узгодженість.

Розділ 8

Перевірка статистичних гіпотез

ChTest

8.1 Загальні відомості

SecGenTest

Поруч із задачами оцінювання параметрів у статистиці велику роль грають задачі перевірки гіпотез. Тут ми не намагаємось оцінити невідомий параметр якомога точніше. Задача полягає в перевірці того, наскільки наші дані підтверджують або суперечать певним припущенням про досліджуване явище.

Статистичними гіпотезами називають припущення про розподіл спостережуваних статистичних даних. На практиці вони пов'язані з деякими змістовними гіпотезами про природу досліджуваного явища, об'єкта, процессу. Правильний підхід до аналізу даних полягає в тому, щоб почати з висунення змістовної гіпотези. Далі потрібно сформулювати ймовірнісну модель та статистичну гіпотезу, яка відповідає змістовній. Після цього — вибрати правильний алгоритм перевірки обраної статистичної гіпотези (статистичний тест), застосувати його та інтерпретувати отримані результати[1].

У цьому підрозділі ми розглянемо формальну постановку задачі перевірки статистичних гіпотез у загальному вигляді. Як ці гіпотези можуть бути пов'язані із змістовними гіпотезами у різних прикладних об-

[1]Досить розповсюджена протилежна практика: до статистичних даних намагаються застосувати різноманітні тести, сподіваючись, що їх результати наштовхнуть дослідника на змістовні гіпотези. На мою думку, такий підхід є невдалим. Якщо у статистика немає розумної ймовірнісної моделі даних, доцільно скористатись дескриптивними методами і спробувати відшукати її. Тільки після цього є сенс висувати статистичні гіпотези та застосовувати тести для їх перевірки.

ластях, буде видно з прикладів, що розглядаються далі.

Нехай весь набір статистичних даних $\mathbf{X}$ є випадковим елементом простору даних $\mathcal{X}$. Розподіл даних $\mathbf{P}_\vartheta^{\mathbf{X}}(A) = \mathsf{P}\{\mathbf{X} \in A\}$ відомий з точністю до невідомого параметра $\vartheta \in \Theta$, де Θ — деяка множина (простір) можливих значень параметра. Наприклад, $\mathbf{X}$ може бути кратною вибіркою обсягу n, а ϑ — числовим параметром. Тоді $\mathcal{X} = \mathbb{R}^n$, $\Theta \subseteq \Theta$. Ми будемо вважати, що ϑ однозначно задає розподіл даних $\mathbf{X}$. Тоді гіпотези про цей розподіл є гіпотезами про можливі значення ϑ. Ми обмежимось розглядом двоальтернативних гіпотез. Будемо вважати, що простір параметрів Θ розбитий на дві множини, що не перетинаються: $\Theta = \Theta_0 \cup \Theta_1$, $\Theta_0 \cap \Theta_1 = \emptyset$. Гіпотезою H_i назвемо припущення про те, що $\vartheta \in \Theta_i$, $i = 0, 1$.

При класичному підході до задачі перевірки гіпотез H_0 і H_1 є не рівноправними.

Гіпотеза H_0 вважається основною, тобто її відхиляють лише тоді, коли дані переконливо показують, що вона є хибною. Гіпотеза H_1 вважається альтернативною, її приймають лише тоді, коли дані переконливо свідчать на її користь.

Для перевірки статистичних гіпотез за даними використовують алгоритми, які звуть статистичними тестами[2] З формальної точки зору тест можна розглядати як функцію, що будь-якому можливому значенню з простору даних співставляє номер гіпотези, яку тест приймає при цьому значенні $\mathbf{X}$. Таким чином, тест це вимірна функція $\pi : \mathcal{X} \to \{0, 1\}$. Якщо $\pi(\mathbf{X}) = 0$, тест приймає основну гіпотезу, якщо $\pi(\mathbf{X}) = 1$ — альтернативу.

На практиці тести часто мають вигляд $\pi(\mathbf{X}) = \mathbb{1}\{S(X) > C\}$, або $\pi(\mathbf{X}) = \mathbb{1}\{S(X) < C\}$, де $S(\mathbf{X})$ — статистика, тобто вимірна функція від даних, а C — деяке фіксоване число. У такому випадку $S(X)$ називають статистикою тесту, а C — порогом.

Характеризація якості тестів. Якість тесту характеризується ймовірністю того, що тест прийме невірну гіпотезу (помилиться). При використанні тесту можливі помилки двох родів.

1. **Помилка першого роду:** вірна основна гіпотеза, а тест її відхиляє, тобто $\vartheta \in \Theta_0$, але $\pi(\mathbf{X}) = 1$. Ймовірність такої помилки

$$\alpha_\pi(\vartheta) = \alpha(\vartheta) = \mathsf{E}_\vartheta\, \pi(\mathbf{X}),\ \vartheta \in \Theta_0.$$

[2]У англомовній літературі прийнята назва *test*, у російськомовній — *критерий*. В українській літературі вживають як назву тест, так і назву критерій. Інколи "критерієм" називають те, що ми далі називаємо статистикою тесту.

2. **Помилка другого роду:** основна гіпотеза хибна, а тест її приймає, тобто $\vartheta \in \Theta_1$, але $\pi(\mathbf{X}) = 0$. Ймовірність такої помилки

$$\beta_\pi(\vartheta) = \beta(\vartheta) = 1 - \mathsf{E}_\vartheta\, \pi(\mathbf{X}),\ \vartheta \in \Theta_1.$$

Ймовірність того, що тест правильно прийме альтернативу, коли вона є вірною, називають **потужністю** (power) тесту і позначають

$$\varphi_\pi(\vartheta) = 1 - \beta_\pi(\vartheta) = \mathsf{E}_\vartheta\, \pi(\mathbf{X}),\ \vartheta \in \Theta_1.$$

Найбільше можливе значення ймовірності помилки першого роду для тесту π називають **рівнем значущості тесту** (test signficance level):

$$\alpha_\pi = \sup_{\vartheta \in \Theta_0} \alpha_\pi(\vartheta).$$

Серед усіх можливих тестів бажано вибрати такий, який матиме найменшу ймовірність помилки. Але, як правило, при зменшенні α_π збільшується β_π і навпаки. Тому у статистиці прийнятий наступний підхід до вибору тестів.

Фіксують деяке мале додатне число $\alpha = \alpha_0$, яке звуть **стандартним рівнем значущості** і розглядають лише тести π, для яких

$$\boxed{\text{EqSigDef}}\ \ \alpha_\pi \le \alpha_0 \tag{8.1}$$

(тобто ймовірність помилково відхилити основну гіпотезу не перевищує стандартного рівня значущості). Серед таких тестів вибирають тест з найбільшою потужністю $\varphi_\pi(\vartheta)$.

Якщо вдається знайти тест π^*, такий, що $\alpha_{\pi^*} \le \alpha_0$ і для всіх інших тестів π, які задовольняють умову (8.1) виконано

$$\varphi_{\pi^*}(\vartheta) \ge \varphi_\pi(\vartheta) \text{ для всіх } \vartheta \in \Theta_1,$$

то тест π^* називають **рівномірно найбільш потужним тестом** (р.н.п.) рівня α для перевірки гіпотези H_0 проти альтернативи H_1.

Р.н.п. тести існують не для всіх гіпотез. Часто їх шукають не в класі всіх можливих тестів, а лише в якомусь класі "правильних" тестів. Якщо р.н.п. тест у даній задачі перевірки гіпотез знайти не вдається, використовують тести, потужність яких є достатньою для практичних потреб.

Вибір основної гіпотези та рівня значущості. Оскільки H_0 і H_1 не є рівноправними, важливо правильно визначитись, яке з двох альтернативних припущень вважати основним при перевірці гіпотез. Нехай,

наприклад, дослідник проводить експеримент з метою виявити певний ефект (скажімо, ефектом може бути вплив ліків на протікання хвороби). В результаті експерименту отримані дані $\mathbf{X}$. Якщо вірним є припущення про відсутність ефекту, $\mathbf{X}$ має розподіл R, якщо ефект є — розподіл S. На роль основної можна взяти гіпотезу $F^{\mathbf{X}} = R$ або гіпотезу[3] $F^{\mathbf{X}} = S$. Який вибір кращий?

Якщо дослідник хоче своїми даними переконати колег в тому, що ефект виявлений, він повинен як основну взяти гіпотезу про те, що ефекту немає — $H_0 : F^{\mathbf{X}} = R$. Тоді, якщо відповідний статистичний тест прийме $H_1 : F^{\mathbf{X}} = S$, це буде підтвердженням наявності ефекту на основі даних експерименту, а не внаслідок того, що гіпотеза $F^{\mathbf{X}} = S$ апріорі була вибрана основною.

І навпаки, якщо досліди проводять, наприклад, для перевірки відсутності нехороших побічних ефектів ліків, основною повинна бути гіпотеза про те, що такі ефекти є.

Наступним важливим кроком в організації перевірки гіпотез є вибір стандартного рівня значущості α. Цей рівень також визначається не з математичних, а з прикладних міркувань. Наприклад, нехай у ситуації аналізу ефекту ліків основна гіпотеза — відсутність ефекту. Тоді, обравши $\alpha = 0.05$, ми, в середньому один раз на двадцять випадків, коли ліки не дають ефекту, будемо хибно його виявляти. І, відповідно, рекомендувати ці ліки для подальшого використання. Якщо така ситуація нас не влаштовує, можна зменшити α, встановивши його, наприклад, 0.01. Але тоді ми ризикуємо частіше не помічати тих ефектів, які ліки справді дають.

З цих міркувань у різних предметних областях встановлюються різні стандартні рівні значущості. Найменший з рекомендованих рівнів 0.05 прийнятий у соціології, економіці, медицині. Більш строгі рівні 0.01 або 0.001 прийняті у експериментальній фізиці та інженерних науках.

Досягнутий рівень значущості тесту. (attained significance, p-value) Оскільки вибір рівня значущості може змінюватись в залежності від обставин, статистичні тести прийнято одразу розробляти так, щоб вони могли працювати з будь-яким обраними стандартним рівнем значущості. При цьому і результат роботи тесту на конкретних даних часто зручно подавати так, щоб по ньому можна було одразу сказати, при яко-

[3] У цьому випадку невідомим параметром можна вважати сам розподіл, який приймає значення з двоелементної множини $\Theta = \{R, S\}$.

му рівні значущості приймається основна гіпотеза, а при якому — альтернатива. Для такого запису використовується спеціальна статистика $p(\mathbf{X})$.

Справа в тому, що один і той же тест, як правило, можна записати у багатьох еквівалентних формах. Нехай, наприклад, тест має вигляд

$$\boxed{\text{EqTestSt}}\quad \pi(\mathbf{X}) = \mathbb{1}\{S(\mathbf{X}) > C_\alpha\}, \tag{8.2}$$

де поріг C_α і статистика тесту $S(\mathbf{X})$ підібрані так, що рівень значущості тесту $\alpha_\pi = \alpha$. Візьмемо довільну строго зростаючу функцію h. Тоді (8.2) еквівалентно

$$\pi(\mathbf{X}) = \mathbb{1}\{h(S(\mathbf{X})) > h(C_\alpha)\}.$$

Таким чином, пара (статистика,поріг) $(h(S(\mathbf{X})), h(C_\alpha))$ є еквівалентною парі $(S(\mathbf{X}), C_\alpha)$ — вони породжують один і той же тест.

Серед всіх еквівалентних статистик тесту є одна — $p(\mathbf{X})$, при використанні якої тест набуває вигляду

$$\pi(\mathbf{X}) = \mathbb{1}\{p(\mathbf{X}) < C_\alpha\}.$$

Ця статистика $p(\mathbf{X})$ і зветься досягнутим рівнем значущості тесту.

Таким чином, якщо на конкретних даних тест дає значення досягнутого рівня значущості p, то основна гіпотеза приймається при $p \geq \alpha$ і відхиляється при $p < \alpha$.

Розглянемо важливий частковий випадок, коли статистика $S(\mathbf{X})$ тесту $\pi(\mathbf{X})$, заданого (8.2), підібрана так, щоб її розподіл був тим самим при всіх значеннях параметра, що відповідають основній гіпотезі[4]. Позначимо функцію розподілу для цього розподілу $G(s)$:

$$G(x) = \mathsf{P}_\vartheta\{S(\mathbf{X}) < s\} \text{для всіх } \vartheta \in \Theta_0.$$

Обмежимось випадком, коли G — неперервна строго зростаюча функція. Тоді тільки поріг $C_\alpha = Q^G(1-\alpha)$ забезпечує рівень значущості α для тесту π.

Таким чином, тест з рівнем значущості α має вигляд

$$\pi(\mathbf{X}) = \mathbb{1}\{S(\mathbf{X}) > G^{-1}(1-\alpha)\} = \mathbb{1}\{1 - G(S(\mathbf{X})) < \alpha\}.$$

Отже досягнутий рівень значущості цього тесту можна підрахувати за формулою $p(\mathbf{X}) = 1 - G(S(\mathbf{X}))$.

[4]Такі тести називають **незалежними від розподілу**. Зрозуміло, що на альтернативі розподіл статистики має відрізнятись від розподілу на основній гіпотезі, інакше від такого тесту користі не буде.

8.2 Тест відношення вірогідності для перевірки простих гіпотез

SecLRSimp

Статистична гіпотеза зветься простою, якщо вона однозначно визначає розподіл даних. Розглянемо випадок, коли і основна гіпотеза і альтернатива є простими. У цьому випадку параметричну модель розподілу даних можна не описувати, а задавати гіпотези безпосередньо вказуючи відповідний розподіл.

Як і раніше, позначимо набір статистичних даних через $\mathbf{X}$. Нехай гіпотезі H_i $(i = 0, 1)$ відповідає розподіл даних $F_i^{\mathbf{X}}$. Припустимо, що для деякої міри μ на просторі даних $\mathcal{X}$ існують щільності розподілів $F_i^{\mathbf{X}}$ відносно μ, які ми позначимо $f_i^{\mathbf{X}}$:

$$F_i^{\mathbf{X}}(A) = \int_A f_i^{\mathbf{X}}(x)\mu(dx).$$

Відношенням вірогідності (likelihood ratio) для перевірки простої гіпотези H_0 проти простої альтернативи H_1 називають статистику

$$\mathrm{LR}(\mathbf{X}) = \frac{f_1^{\mathbf{X}}(\mathbf{X})}{f_0^{\mathbf{X}}(\mathbf{X})}.$$

Тестом відношення вірогідності з порогом C називають тест

$$\pi(\mathbf{X}) = \pi_C(\mathbf{X}) = \begin{cases} 1 & \text{якщо } \mathrm{LR}(\mathbf{X}) > C, \\ 0 & \text{якщо } \mathrm{LR}(\mathbf{X}) \le C. \end{cases}$$

У випадку простих основної та альтернативної гіпотез, ймовірності помилок першого і другого роду для будь-якого тесту π виражаються кожна одним числом (а не функцією від невідомого параметру ϑ, як у загальному випадку) і позначаються відповідно α_π та β_π.

Теорема 8.2.1 *Якщо при виконанні H_0 випадкова величина $LR(\mathbf{X})$ має неперервну функцію розподілу G, то тест відношення вірогідності $\pi_\alpha^*(\mathbf{X}) = \pi_C(\mathbf{X})$ з $C = C_\alpha = Q^G(1-\alpha)$ буде найбільш потужним[5] тестом для перевірки H_0 проти H_1 з рівнем значущості α.*

Досягнутий рівень значущості цього тесту

$$p(\mathbf{X}) = 1 - G(LR(\mathbf{X})).$$

[5]У термінології попереднього підрозділу — рівномірно найбільш потужним, але зараз потужність це не функція, а одне число, тому казати "рівномірно" немає сенсу.

Зрозуміло, що такий найбільш потужний тест можна будувати, використовуючи не безпосередньо статистику відношення вірогідності $\mathrm{LR}(\mathbf{X})$, а будь-яку монотонну функцію $h(\mathrm{LR}(\mathbf{X}))$ від неї. Для визначення порогу c_α у такому тесті $\pi(h(\mathrm{LR}(\mathbf{X})))$ можна скористатись умовою

$$\alpha = \alpha_\pi = \mathsf{E}_0\, \pi(h(\mathrm{LR}(\mathbf{X}))).$$

(математичне сподівання береться в припущенні, що вірною є H_0).

У випадку, коли $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ є кратною вибіркою, щільність всього набору даних $f_i^{\mathbf{X}}$ записується як добуток щільностей окремих спостережень ξ_i. Тому

$$\mathrm{LR}(\mathbf{X}) = \prod_{j=1}^{n} \frac{f_1(\xi_j)}{f_0(\xi_j)},$$

де f_i — щільність розподілу ξ_j при виконанні гіпотези H_i.

Часто на практиці працювати з сумами буває зручніше ніж з добутками, тому для кратних вибірок використовують також логарифмічне відношення вірогідності

$$\mathrm{lr}(\mathbf{X}) = \log \mathrm{LR}(\mathbf{X}) = \sum_{j=1}^{n} \log\left(\frac{f_1(\xi_j)}{f_0(\xi_j)}\right).$$

У багатьох ситуаціях практичного застосування статистичних тестів описати аналітично розподіл відношення вірогідності даних при виконанні основної гіпотези (або альтернативи) не вдається. Але підібрати правильний поріг тесту та визначити ймовірність помилки другого роду можна використовуючи імітаційне моделювання.

Для цього згенеруємо спочатку велику кількість B незалежних між собою вибірок обсягу n з розподілом, що відповідає основній гіпотезі. Позначимо ці вибірки $\mathbf{X}^{(1;0)}, \ldots, \mathbf{X}^{(B;0)}$. Підрахуємо значення статистики lr на кожній з цих вибірок: $\mathrm{lr}_j^0 = \mathrm{lr}(\mathbf{X}^{(j;0)})$. Набір $\mathbf{L}^{(0)} = (\mathrm{lr}_1^0, \ldots, \mathrm{lr}_B^0)$ є кратною вибіркою з розподілом, який відповідає розподілу G статистики $\mathrm{lr}(\mathbf{X})$ при виконанні основної гіпотези. Тому вибірковий квантиль $\hat{c}_\alpha = Q^{\mathbf{L}^0}(1-\alpha)$ є хорошою оцінкою для порогу тесту $c_\alpha = Q^G(1-\alpha)$, що відповідає рівню значущості α. Поріг $\hat{c}_\alpha$ можна використовувати при реалізації тесту для перевірки гіпотез на конкретних даних $\mathbf{X}$. При цьому тест π приймає гіпотезу H_0, якщо $\mathrm{lr}(\mathbf{X}) \le \hat{c}_\alpha$ і відхиляє, якщо $\mathrm{lr}(\mathbf{X}) > \hat{c}_\alpha$.

Досягнутий рівень значущості тесту можна оцінити як

$$\hat{p}(\mathbf{X}) = 1 - \hat{F}_B^{\mathbf{L}^0}(\mathrm{lr}(\mathbf{X})) = \frac{1}{B}\sum_{j=1}^{B} \mathbb{1}\{\mathrm{lr}_j^0 > \mathrm{lr}(\mathbf{X})\}.$$

Для того, щоб оцінити ймовірність помилки другого роду цього тесту, потрібно згенерувати вибірки з розподілом, що відповідає альтернативі: $\mathbf{X}^{(1;1)},\ldots,\mathbf{X}^{(B;1)}$ і підрахувати статистику lr на них: $\mathrm{lr}_j^1 = \mathrm{lr}(\mathbf{X}^{(j;1)})$, $\mathbf{L}^{(1)} = (\mathrm{lr}_1^1,\ldots,\mathrm{lr}_B^1)$.

Оцінкою для β_π буде частота помилок тесту π на вибірках $\mathbf{X}^{(j;1)}$:

$$\hat{\beta}_\pi = \frac{1}{B}\sum_{j=1}^{B} \mathbb{1}\{\mathrm{lr}_j^1 < \hat{c}_\alpha\}.$$

Приклад 1. Розглянемо наступну умовну задачу. Нехай деяким підприємством була закуплена партія n електричних лампочок. Всі лампочки були використані і відмічений час роботи кожної лампочки до перегоряння ξ_j. Відомо, що фірмові лампочки мають експоненційний розподіл часу роботи до перегоряння з інтенсивністю λ_0, а час роботи дешевого аналогу фірмових виробів — також експоненційний з інтенсивністю $\lambda_1 > \lambda_0$. Лампочки були закуплені як фірмові, але за спостереженнями виникла підозра, що це дешевий аналог. Потрібно вирішити, чи слід виставляти рекламацію поставнику товару.

Гіпотеза — A: "лампочки є фірмовими" не вимагає додаткових дій. Гіпотеза — B: "лампочки є дешевим аналогом" приводить до необхідності подавати рекламацію. Отже для прийняття гіпотези B потрібне обґрунтування на основі спостережуваних даних. Тому як основну слід обрати гіпотезу A, і дотримуватись її доти, доки дані не змусять нас прийняти B.

Таким чином, $\mathbf{X} = (\xi_1,\ldots,\xi_n)$ — кратна вибірка з експоненційного розподілу з інтенсивністю λ. Потрібно за цією вибіркою перевірити гіпотезу H_0: $\lambda = \lambda_0$ проти альтернативи H_1: $\lambda = \lambda_1$, причому $\lambda_0 < \lambda_1$.

Легко бачити, що логарифмічне відношення вірогідності для цих гіпотез дорівнює

$$\mathrm{lr}(\mathbf{X}) = \sum_{j=1}^{n} \log\left(\frac{\lambda_1 e^{-\lambda_1\xi_j}}{\lambda_0 e^{-\lambda_0\xi_j}}\right) = n(\log\lambda_1 - \log\lambda_0) + (\lambda_0 - \lambda_1)\sum_{j=1}^{n}\xi_j.$$

Оскільки $\sum_{j=1}^n \xi_j$ є монотонно спадною функцією від $\mathrm{lr}(\mathbf{X})$, тест відношення вірогідності можна записати у еквівалентній формі $\pi_c(\mathbf{X}) = \mathbb{1}\{\sum_{j=1}^n < c\}$. Для заданого рівня значущості α відповідний поріг $c = c_\alpha$ можна вибрати з умови

$$\alpha = \mathsf{E}_0\, \pi_c(\mathbf{X}) = \mathsf{P}_{\lambda_0}\{\sum_{j=1}^n \xi_j < c\}.$$

Якщо ξ_j — експоненційно розподілені з інтенсивністю λ незалежні випадкові величини, то $\sum_{j=1}^n \xi_j$ має Γ розподіл з інтенсивністю λ і параметром форми n, тобто $\Gamma(n, \lambda)$. Отже $c_\alpha = Q^{\Gamma(n,\lambda_0)}(\alpha)$.

Ймовірність помилки другого роду для цього тесту

$$\beta_\pi = 1 - \mathsf{E}_{\lambda_1}\, \pi(\mathbf{X}) = \mathsf{P}\{\sum_{j=1}^n \xi_j > c_\alpha\} = F^{\Gamma(n,\lambda_1)}(Q^{\Gamma(n,\lambda_0)}(\alpha)).$$

Приклад 2. Цей приклад також є умовним. Нехай досліджується генотип деякої рослини. Дослідника цікавить певний ген, який впливає на конкретну ознаку рослини ξ (наприклад, співвідношення довжини і ширини листка). Відомо, що є два алеля (варіанти) цього гена — домінантний A та рецесивний — a. Дослідника цікавить конкретна рослина Z, генотип якої може бути Aa або aa. Для з'ясування того, яким насправді є генотип Z, цю рослину схрещують з рослиною, яка точно має генотип aa. Якщо генотип Z є aa, то всі нащадки теж будуть мати генотип aa. Якщо генотип Z — Aa, то кожен з нащадків з ймовірністю $p = 1/2$ отримає генотип Aa, і з ймовірністю $1 - p$ — генотип aa.

Відомо, що ознака ξ у рослин з генотипом aa має щільність розподілу f_{aa}, а у рослин з генотипом Aa — f_{Aa}. Отримано n нащадків, значення ξ у j-того нащадка — ξ_j. Потрібно за $\mathbf{X} = (\xi_1, \ldots, \xi_n)$ визначити, яким був генотип рослини Z. Основною є гіпотеза про генотип aa.

Зрозуміло, що щільність, яка відповідає основній гіпотезі — це $f_0(x) = f_{aa}(x)$. Щільність, що відповідає альтернативі є сумішшю двох щільностей з ймовірностями p та $1-p$: $f_1(x) = pf_{Aa}(x) + (1-p)f_{aa}(x)$. Логарифмічне відношення вірогідності має вигляд

$$\mathrm{lr}(\mathbf{X}) = \sum_{j=1}^n \log\left(\frac{pf_{Aa}(\xi_j) + (1-p)f_{aa}(\xi_j)}{f_{aa}(\xi_j)}\right).$$

Аналітично розподіл цієї величини виразити не можна. Тому скористаємось імітаційним моделюванням, як описано вище. У прикладі розглядається випадок, коли f_{aa} та f_{Aa} — гауссові щільності з математичним сподіванням і середньоквадратичним відхиленням `m0=2`, `s0=0.4` для f_{aa} і `m1=3`, `s1=0.75` для f_{Aa}. Кількість спостережень (кількість нащадків досліджуваної рослини) `n=12`.

```
> set.seed(5)
> m0<-2    # середнє для aa
> s0<-0.4 # сер. кв. відх. для aa
> m1<-3    # середнє для Aa
> s1<-0.75 # сер. кв. відх. для Aa
> p<-0.5   # ймовірність aa при альтернативі
> n<-12    # обсяг вибірки
> # щільність, що відповідає основній гіпотезі
> f0<-function(x){log(dnorm(x,m0,s0))}
> # щільність, що відповідає альтернативі
> f1<-function(x){log(p*dnorm(x,m0,s0)+(1-p)*dnorm(x,m1,s1))}
> # відношення вірогідності (x - вибірка)
> lr<-function(x)sum(sapply(x,f1)-sapply(x,f0))
> # генератор однієї вибірки при H0
> gen0<-function(n)rnorm(n,m0,s0)
> # генератор однієї вибірки при H1
> gen1<-function(n){
+ z<-rbinom(n,size=1,prob=p)
+ rnorm(n,z*m0+(1-z)*m1,z*s0+(1-z)*s1)
+ }
> alpha<-0.05 # стандартний рівень значущості
> B<-10000    # кількість модельованих вибірок
> lr0<-numeric(B) # масив значень lr при H0
> for(i in 1:B){
+ lr0[i]<-lr(gen0(n))
+ }
> # поріг тесту, що відповідає рівню alpha
> Ca<-quantile(lr0,1-alpha)
> Ca
      95%
-1.326246
```

```
> lr1<-numeric(B) # масив значень lr при H1
> for(i in 1:B){
+ lr1[i]<-lr(gen1(n))
+ }
> # оцінка ймовірності помилок першого роду
> mean(lr1<Ca)

[1] 0.0264

> # відображення гістограм для lr
> mi<-min(c(lr0,lr1))
> mx<- 60 #max(c(lr0,lr1))
> hist(lr0,breaks=15,probability=T,
+ angle=0,density=12,xlim=c(mi,mx),ylim=c(0,0.33),
+ col="red",xlab="lr",main="Histogram of lr")
> hist(lr1,probability=T,
+ breaks=15,angle=90,density=12, xlim=c(mi,mx),
+ col="blue",add=T)
> abline(v=Ca)
```

Як бачимо, для рівня значущості $\alpha = 0.05$ ми отримали поріг тесту $c_\alpha = -1.326246$. При цьому ймовірність помилки другого роду оцінюється як $\beta_\pi = 0.0264$. Ця оцінка вийшла навіть меншою ніж встановлена нами ймовірність помилки першого роду.

На рис. 8.1 зображені гістограми значень (логарифмічного) відношення вірогідності для основної гіпотези (червоним кольорм, горизонтальна штриховка) та альтернативи (синім кольором, вертикальна штриховка). Вертикальна лінія відмічає положення порогу c_α. Площа тієї частини гістограми для H_0, що лежить праворуч від c_α відповідає ймовірності помилки першого роду $\alpha_\pi = 0.05$. Площа частини гістограми для H_1, яка лежить ліворуч від c_α відповідає ймовірності помилки другого роду β_π. Зміщуючи положення порогу праворуч можна зменшити α_π, але β_π при цьому збільшиться. І навпаки, зменшуючи поріг, ми збільшуємо ймовірність помилки першого роду та збільшуємо — другого. Таким чином, отримані гістограми дозволяють побачити, наскільки хорошим може бути тест для розпізнавання цих двох гіпотез. Чим ближче ці гістограми одна до одної, тим менше шансів побудувати хороший тест.

Нехай в ході експерименту було отримано наступні значення ξ:

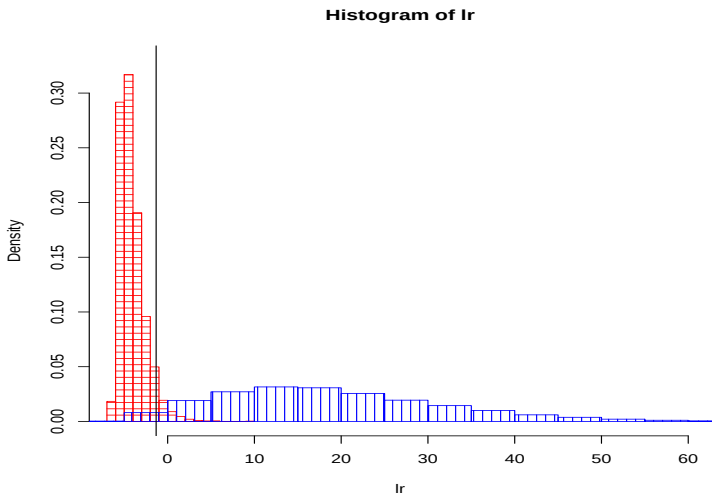

Рис. 8.1: Гістограми для відношень вірогідності з прикладу 2. Червоним - при основній гіпотезі, синім - при альтернативі. Вертикальна лінія відповідає порогу тесту

2.6	1.7	1.9	3.1	3.9	1.7
1.6	1.9	1.6	2.9	2.6	1.4

Перевіримо, на користь якої гіпотези свідчать ці дані:

```
> # Вибірка для перевірки гіпотези:
> x<-c(2.6,1.7,1.9,3.1,3.9,1.7,
+ 1.6,1.9,1.6,2.9,2.6,1.4)
> # статистика відношення вірогідності:
> lr(x)

[1] 9.685149

> # досягнутий рівень значущості:
> mean(lr0>lr(x))

[1] 0
```

Тут ми спочатку ввели вибірку і позначили її `x`. Потім підрахували логарифмічне відношення вірогідності — воно виявилось рівним 9.685149.

Це більше, ніж обчислене нами c_α, отже, при рівні значущості 0.05 слід прийняти альтернативу: генотип досліджуваної рослини Aa. Далі ми підрахували досягнутий рівень значущості, він вийшов рівним 0. (Насправді, звичайно, 0 це лише наша оцінка, справжнє $p(\mathbf{X})$ додатне, але настільки мале, що наша техніка оцінювання не дозволяє помітити його відмінність від 0). Таким чином, при всіх розумних рівнях значущості ці дані свідчать на користь альтернативи.

Література

[1] Боровков А.А. Математическая статистика. - Наука, Москва, 1984. - 472 с.

[2] Кнут Д. Э., Искусство программирования. Том 2. Получисленные методы — Вильямс. 2001.

[3] Liu, J., D. Nissim, and J. Thomas (2002). Equity valuation using multiples. Journal of Accounting Research, 40(1), 135-172.

[4] Makino J. Lagged-Fibonacci random number generator on parallel computers.- Parallel Computing, 20: 1357-1367, 1994.

[5] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P.Flannery Numerical Recipes in C: The Art of Scientific Computing (1992) Cambridge University Press New York, NY, USA