SECTION 3. COMPUTER SCIENCE

DOI: 10.46299/ISG.2023.MONO.TECH.2.3.1

3.1 Обробка емпіричних даних у середовищі RStudio з використанням мови програмування R

Емпіричні дослідження – це спостереження і дослідження конкретних явищ, а також узагальнення, класифікація та опис результатів дослідження і експерименту, впровадження їх у практичну діяльність людей.

Емпіричні дослідження використовуються для відповіді на емпіричні питання, які повинні бути визначені з даними.

Дослідник має певні теорії на тему, з якої ведуться дослідження. На основі цієї теорії пропонуються певні припущення або гіпотези.

З цих гіпотез роблять прогнозування конкретних подій чи визначення значень величини. Ці прогнозування можуть бути перевірені конкретними експериментами.

Залежно від результатів експерименту теорії, на яких гіпотези або прогнози були засновані, будуть підтверджуватися або спростовуватися. Точний аналіз емпіричних (дослідних даних) з використання стандартних статистичних методів у дослідженнях має вирішальне значення для визначення обґрунтованості емпіричних досліджень.

Статистичні формули такі, як регресія, коефіцієнт невизначеності, Т – критерій, Хі – квадрат критерій і різні види дисперсійного аналізу мають основне значення для формування висновків. Якщо емпіричні дані досягають значущості у відповідних статистичних формулах, гіпотеза підтверджується. Якщо ні, то підтверджується нульова гіпотеза.

Емпіричний цикл.

1. Спостереження (Observation) – збір та групування емпіричних даних, формування гіпотези.

2. Індукція (Induction) – розробка гіпотез.

3. Дедукція (Deduction) – виведення послідовності гіпотез.

4. Перевірка (Testing) – перевірка гіпотези з нового емпіричного матеріалу.

5. Оцінка (Evaluation) – оцінка результатів перевірки.

Основні задачі емпіричних методів пов’язані з наближеним визначенням розподілів ймовірностей на множині значень досліджуваних випадкових величин та їх параметрів на основі статистичних даних, одержаних в результаті проведення *n* незалежних випробувань, пов’язаних з певним стохастичним експериментом.

Використання мови програмування R та середовища RStudio для емпіричних досліджень.

R – це мова програмування та програмне середовище для аналізу емпіричних даних, графічного представлення та звітності. R знаходиться у вільному доступі під General Public License і існують її версії для різних операційних систем, таких як Linux, Windows та Mac. Ця мова програмування була названа R відповідно до першої літери імені двох авторів R (Роберт Джентльмен і Росс Іхака), як мова вона продовжує розроблятися та вдосконалюється.

Основою R є інтерпретована комп'ютерна мова, яка дозволяє виконувати автоматизацію обчислювальних процесів з різними типами алгоритмів, а також модульне програмування з використанням функцій. R забезпечує ефективність інтеграції з процедурами, написаними мовами C, C++, .Net, Python.

R була написана Россом Іхакой та Робертом Джентльменом на факультеті статистики Оклендського університету в Окленді, Нова Зеландія та зробила свою першу появу в 1993 році. Велика група людей зробила свій внесок у R, надіславши код та звіти про помилки. З середини 1997 року існує основна група (R Core Team), яка може змінювати архів вихідного коду R.

При програмуванні мовою R досягаються важливі особливості:

R – це добре розроблена, проста і ефективна мова програмування, яка включає умовні вирази, цикли, користувальницькі рекурсивні функції та засоби

введення та виведення.

R має ефективні засоби обробки та зберігання даних,

R надає набір операторів для обчислень над масивами, списками, векторами та матрицями.

R надає великий, узгоджений та інтегрований набір інструментів для аналізу даних.

R надає графічні засоби для аналізу даних та відображення безпосередньо на комп'ютері, або для друку на папері.

Напевно потрібно вказати, що R – найширше використовувана у світі мова програмування для статистики. Це вибір дослідників даних, який підтримується енергійною та талановитою спільнотою учасників. R викладається в університетах і використовується в критично важливих бізнес-додатках.

Працювати мовою програмування R значно швидше і простіше у середовищі RStudio, так як RStudio дозволяє писати множинні рядки коду, підключати та підтримувати бібліотеки і взагалі більш продуктивно облаштовувати своє робоче середовище.

R застосовується скрізь, де потрібна робота з даними, не лише статистика у вузькому значенні слова, а й «первинний» аналіз (графіки, таблиці спряженості), і просунуте математичне моделювання. R без особливих проблем може використовуватися і там, де прийнято використовувати комерційні програми аналізу рівня MatLab. З іншого боку, цілком природно, що основна обчислювальна міць R найкраще проявляється при емпіричному аналізі: від обчислення середніх величин до вейвлет-перетворень часових рядів. Географія використання R дуже різноманітна.

RStudio - це нове інтегроване середовище розробки для R, яке поєднує в собі інтуїтивно зрозумілий інтерфейс користувача з потужними інструментами кодування, що дозволяє максимально використовувати можливості R і швидше виконувати поставлені завдання. Як і R, RStudio доступний під ліцензією з відкритим вихідним кодом, що гарантує свободу обміну та зміни програмного забезпечення.

У цьому розділі розглядається виключно RStudio і R як стандартний блок для досліджень та аналізу завдань, де використовуються емпіричні дані та обчислення з ними.

Основні принципи та прийоми роботи в RStudio.

Розглянемо основні принципи та прийоми роботи в RStudio.

Консольний режим.

Після запуску RStudio користувач потрапляє в консольний режим роботи. Будь-яка команда, написана користувачем, буде відразу виконана R після натискання Enter.

Отримати допомогу по функції можна командою help (function name) або ?function name. У правому нижньому кутку на вкладці Help з'явиться довідка.

Написання скриптів.

В R зручніше писати не по одній команді, а відразу цілий набір команд і потім запускати їх все на виконання. Для цього потрібні скрипти. Щоб створити скрипт, слід вибрати File -> New File -> R Script. Відкриється нова область, в якій можна писати команди. Коментарі, які не виконуватиме R, пишуться зі знака #. При натисканні Enter команди у скріпті виконуватися не будуть, а буде лише здійснений перехід на новий рядок. Щоб виконати команду в режимі скрипта, слід поставити курсор на потрібний рядок і натиснути Ctrl + Enter. Якщо команда займає більше одного рядка, то необхідно ставити знак + в кінці кожного рядка. Команда виконується по рядках, у кожному рядку потрібно натискати Ctrl + Enter, або виділити всю команду і натиснути Ctrl + Enter один раз.

R може підказувати, які команди доступні, якщо почати вводити перші символи і натиснути або Tab, або Ctrl + Enter.

В основному в R працюють з наборами даних. Така структура має вRназвydata.frame та являє собою таблицю, де кожний стовпчик – це деяка змінна а кожний рядок – це деяке спостереження.

Створимо у режимі скрипта data.frame. Нехай є спостереження за зростом і вагою деяких людей. Задамо два вектора:

>rost <- c (160, 175, 155, 190, NA)

>ves <- c (NA, 70, 48, 85, 60)

І об'єднаємо їх в набір даних, який помістимо в пере- менную df, а потім виведемо на екран:

> df<- data.frame (rost,ves)

Звертатися до конкретних спостереженнями df можна, використовуючи квадратні дужки:

> df [217,215]

[215] 155

Звертатися до змінних можна, використовуючи знак $ або вказуючи на стовпець з пропуском номера рядка:

> df$rost

[215] 160 175 155 190 NA

або

> df [,215]

[215] 160 175 155 190 NA

Основні описові статистики (середнє, стандартневідхилення і медіану) можна отримати за допомогою функцій mean, sd і median:

mean (df$rost, na.rm = T)

[215] 170

sd (df$rost, na.rm = T)

[215] 15.81139

median (df $ rost, na.rm = T)

[215] 167.5

Можна зберегти скрипт, натиснувши File -> Save. При першому зберіганні R запропонує вибрати кодування. Рекомендується вказати UTF-8, щоб букви (наприклад, у коментарях) відображалися коректно. Потім необхідно вибрати директорію і задати ім'я файлу, який буде збережений з розширенням * .R.

Установка пакетів.

В R існує базовий набір пакетів (бібліотек), який містить самі необхідні функції. Для реалізації завдань економетрики потрібні додаткові пакети. Для роботи з зовнішніми пакетами необхідно виконати дві дії - встановлення та підключення потрібного пакету. Установку неободно виконати один раз, а підключати у кожній робочій сесії. Для установки пакетів існує функція install.packages. Також автоматично будуть доустановлені пов'язані пакети. Для підключення встановленого пакета слід скористатися функцією library. Наприклад, встановимо такі пакети:

psych - містить функції для розрахунку описових статистик.

dplyr - містить функції для роботи з data.frame;

ggplot2 - найпотужніший пакет для побудови красивих графіків, діаграм, карт і т. д.

install.packages (c ("psych", "dplyr", "ggplot2"))

Прямим повідомленням про помилку установки є тільки слово Error, що з'являється в консолі. Всі інші повідомлення Warning є просто попередженнями про що-небудь.

Для того щоб визначити, які пакети є необхідними для роботи, можна скористатися пошуком в мережі Інтернет, задавши питання на англійській мові. Наприклад, щоб знайти, в якому пакеті знаходиться алгоритм Левенберга-Марквардта для розрахунку нелінійного МНК можна набрати у пошуку «levenberg-marquuardt algorithm in r» та першим посиланням буде пакет minpack.lm в R.

```
library ( "psych") # описові статистики
library ( "lmtest") # тестування гіпотез в лінійних моделях
library ( "ggplot2") # графіки
library ( "dplyr") # маніпуляції з даними
library ( "MASS") # підгонка розподілів
```

Отримаємо, наприклад, опис набору даних по автомобілям cars командою:

help (cars)

Результат виконання команди (в правому нижньому кутку на вкладці Help) показаний на рисунку 1. У цьому наборі даних 50 спостережень і дві змінні (швидкість, миль / годину і довжина гальмівного шляху в футах).

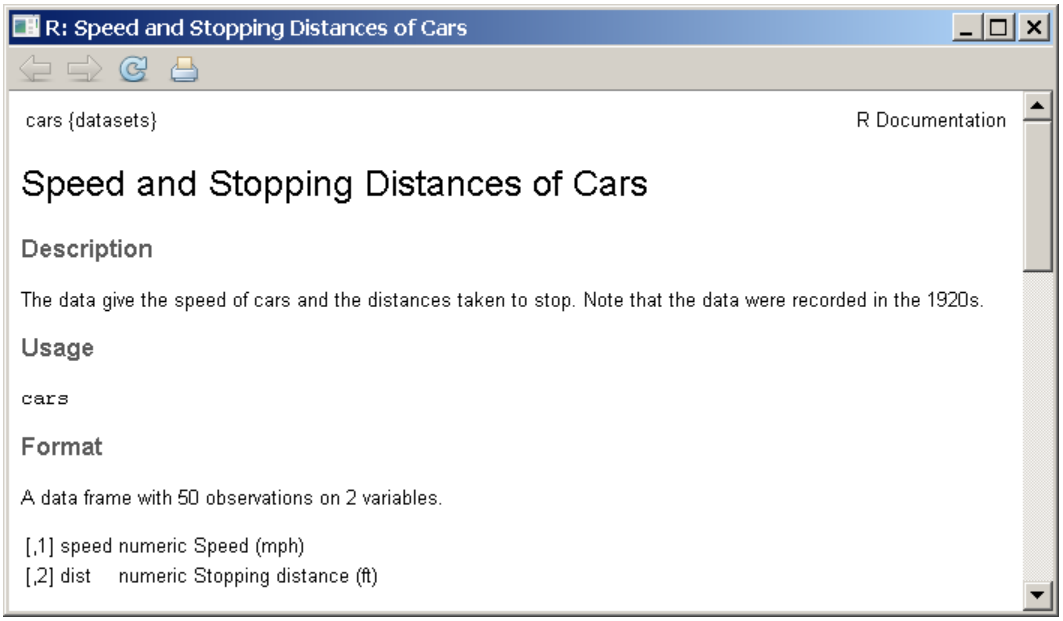

Рис. 1. Довідка з набору даних cars.

Помістимо у змінну d вбудований в R набір даних по автомобілям:

d <- cars # цей набір даних знаходиться в базовому пакеті datasets

Тепер d має тип data.frame (набір даних), у чьому можна впевнитися, подивившись у правому верхньому куті вікна таблиці середовища Environment (рис. 2). Для цього повинен бути обраний режим Grid. За допомогою кнопки можна переглянути вміст набору даних у вигляді таблиці.

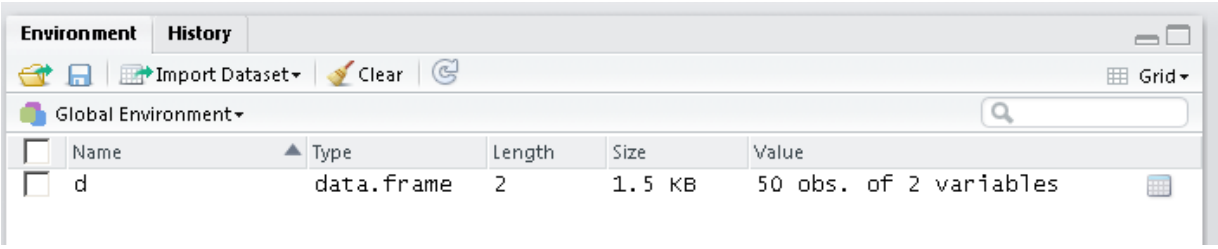

Рис. 2. Опис набору даних d в таблиці середовища Environment.

Наступною командою можна подивитися на цей набір даних, в результаті чого будуть перераховані всі змінні і типи даних:

> glimpse (d) # функція з пакету dplyr

Результат виконання команди з'явиться в консолі:

> d <- cars

> glimpse(d)

Observations: 50

Variables: 2

$ speed (dbl) 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 13, …

$ dist (dbl) 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28, 14, 20, 24, 28, ...

Змінні speed і dist мають тип даних dbl (double) та містить по 50 спостережень. Для інших типів даних використовуються такі скорочення:chr (character / string), int (integer), fctr (factor), tims (time), lgl (logical).

Подивимося на перші шість спостережень набору даних d:

> head (d) # функція з базового пакету utils

speed dist

і останні шість спостережень:

> tail(d) #функція з базового пакету utils

speed dist

Отримаємо таблицю з описовими статистиками: середнє, мода, медіана, стандартне відхилення, мінімум , максимум, асіметрія, ексцес тощо:

> describe(d) # функция из пакета psych

Vars	n	mean	sd	median	trim	med	mad	min	max	range
speed	1	50	15.40	5.29	15	15.47	5.93	4	25	21
dist	2	50	42.98	25.77	36	40.88	23.72	2	120	118

Побудуємо гістограму абсолютних частот для змінної dist (довжини гальмівного шляху). Скористаємося функцією qplot, задавши джерело даних d (аргумента data), змінну для побудови графіка (dist), підпишемо осі (параметри функцій xlab, ylab) та назву графіка (параметр main).

функція з пакету ggplot2

> qplot(data=d,dist,xlab=”Довжина гальмівного шляху”, ylab="Число …

автомобілів”, main=”Дані по автомобілях 1920-х років")

Можна побудувати також діаграму щільності розподілу (рис. 3):

функція з базового пакету graphics

>hist(d$dist,probability=TRUE, col="grey")

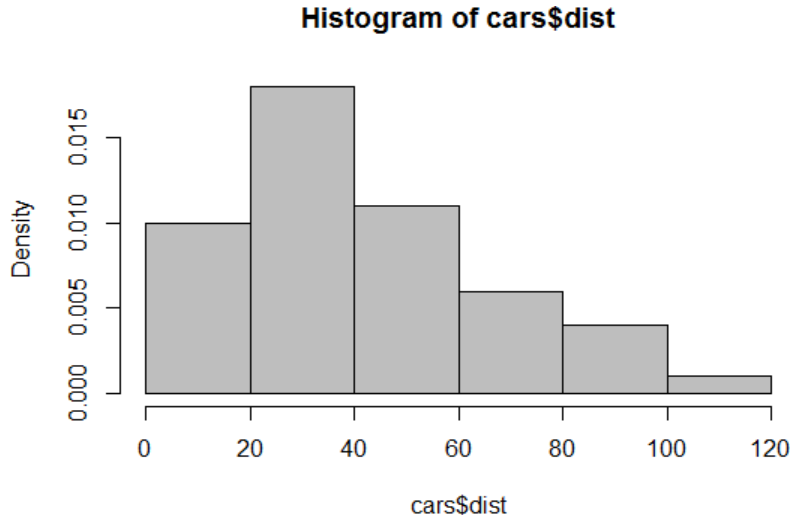

Рис. 3. Гістограма щільності розподілу для змінної dist.

Підгонимо щільність розподілу Вейбула, розташувавши результат (оцінки параметрів розподілу) у змінній fit:

fit<-fitdistr(d$dist,"weibull") # функція з пакету MASS

Змінна fit тепер буде представлена у вигляді списку (List) із п'яти елементів, це відображається у Environment.

Доступ до елементів списку можна отримати за допомогою символу $.

Оцінки двох параметрів розподілу Вейбула були розраховані методом максимальної правдоподібності. Переглянемо їх, звернувшись до елементу списку fit:

> fit $ estimate

shape scale

1.72371 48.15234

Покажемо на тому ж графіку теоретичну щільність розподілу Вейбула (рис. 4).

>xvals<-seq(0,120,.20) # значення по осі абсцис від 0 до 120 з кроком 0,2

>lines(xvals,dweibull(xvals,shape=fit$estimate[215], scale=fit$estimate[216]))

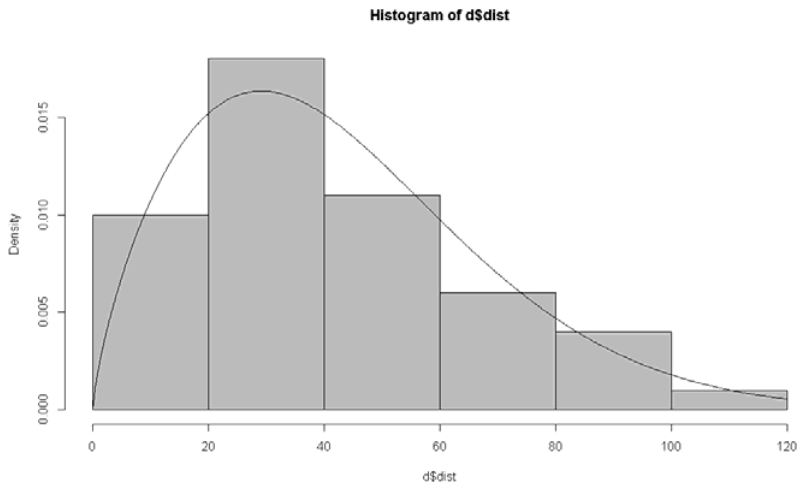

Рис. 4. Гістограма розподілу змінної dist.

Перший аргумент функції **lines**– це значення по осі абсцис, на основі яких буде побудовано графік. Далі вказується функція щільності **dweibull**, для неї потрібно визначити значення аргументу для розрахунку та значення двох параметрів розподілу: коефіцієнт форми **(shape)** та масштабу **(scale)**.