НЕСПЛЯК Д. М.

ЗАСТОСУВАННЯ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ У ПСИХОЛОГІЇ

КОНСПЕКТ ЛЕКЦІЙ

Міністерство внутрішніх справ України
Львівський державний університету внутрішніх справ
кафедра інформатики

назва навчальної дисципліни застосування інноваційних технологій у психології
назва конспекту лекцій застосування інноваційних технологій у психології
шифр та назва галузі знань 05 «Соціальні та поведінкові науки»
шифр і назва напряму підготовки 053 «Психологія»
освітній ступінь бакалавр

Львів
2018

УДК 004.38
Ш 55

Рекомендовано до друку Методичною радою
Львівського державного університету внутрішніх справ
Протокол № "11" від 23 травня 2018 р.

Р е ц е н з е н т и:

П. Я Пукач – завідувач кафедри обчислювальної математики та програмування Інституту прикладної математики та фундаментальних наук Національного університету «Львівська політехніка», доктор технічних наук, професор

Т. В. Магеровська – доцент кафедри інформатики Львівського державного університету внутрішніх справ, кандидат фізико-математичних наук, доцент

Неспляк Д. М.

Ш 55 Застосування інноваційних технологій у психології: конспект лекцій / Д. М. Неспляк – Львів: Львівський державний університет внутрішніх справ, 2018. –149 с.

У даному конспекті лекцій висвітлено питання аналізу та візуалізації статистичних даних засобами середовища R. Детально розглянуті векторизовані обчислення в R. Конспект лекцій призначений для викладання навчальної дисципліни «Застосування інноваційних технологій у психології» для здобувачів вищої освіти галузі знань 05 «Соціальні та поведінкові науки» спеціальності 053 «Психологія».

УДК 004.38

ЗМІСТ

ВСТУП

Сьогодні R є безумовним лідером серед систем статистичного аналізу, які вільно розповсюджуються. Провідні університети світу, аналітики найбільших компаній і дослідницьких центрів регулярно використовують R при проведенні науково-технічних розрахунків і створення великих інформаційних проектів.Широке викладання статистики на базі цієї системи і всебічна підтримка науковим співтовариством зумовили те, що приведення скриптів коду на мові R поступово стає загальновизнаним стандартом як в журнальних публікаціях, так і при неформальному спілкуванні вчених усього світу.

Географія використання R дуже різноманітна. Важко знайти американськийабо західноєвропейський університет, де б не працювали з R. Дуже багато серйозних компаній (наприклад, Boeing) встановлюють R для роботи.

R – це мова, орієнтована на статистику. Її можна розглядати як конкурента для таких аналітичних систем, як SAS Analytics, не кажучи вже про такі більш прості пакети, як StatSoft STATISTICA або Minitab. Багато професійних статистиків вирішують свої завдання за допомогою таких продуктів, як IBM SPSS або SAS, без написання будь-якого коду на мові R. Таким чином, в значній мірі рішення про вивчення і використання R – це питання корпоративної культури і професійних переваг стосовно робочих інструментів. Переваги R над іншими інструментами можна пояснити наступним чином:

- R – це потужна скриптова мова. Для обробки невпорядкованих даних потрібні можливості мови програмування; продукти SAS і SPSS мають скриптові мови для задач, для вирішення яких недостатньо спадаючого меню, однак R був створений саме як мова програмування і тому є більш придатним інструментом для цієї мети.
- R – лідер напрямку. Багато нових розробок в статистиці спочатку з'являються як пакети для платформи R ("R-пакети") і тільки потім приходять на комерційні платформи такі як SPSS.

- Інтеграція із засобами публікації документів. R органічно інтегрується із системами публікації документів, що дозволяє вбудовувати статистичні результати і графіку з середовища R в документи публікаційної якості. Ця можливість не потрібна абсолютно всім. Однак у випадку якщо це потрібно, то найкоротший і елегантний шлях полягає у використанні R і LaTeX.

- Безкоштовність. R поширюється вільно. Навіть для більш великого підприємства досить непогано, коли в разі залучення потрібного фахівця на тимчасовій основі воно здатне негайно надати такому фахівцю робочу станцію з передовим аналітичним програмним забезпеченням. При цьому немає ніякої необхідності хвилюватися про бюджет.

Об'єктна орієнтація допомагає мові R залишатись актуальною, оскільки для аналізу нових джерел даних потрібні нові структури даних. Окрім цього R закріплює хороші звички і покращує аналіз.

Лекція 1.

ОСНОВИR.

План

1.1. Вступ

1.2. Статистика

1.3. Дані, рівні виміру, матриця даних, частотні таблиці, центральна тенденція

1.4. Візуальний аналіз. Типи діаграм.

1.5. Парадокс Сімпсона

1.1. Вступ

Є три основні компоненти, які потрібно знати, для того щоб аналізувати дані (рис. 1.1).

Рис. 1.1 Компоненти аналізу даних

В першу чергу це знання предметної області. Це дозволяє розуміти, які проблеми потребують першочергового вирішення. Друге – це знання математики та статистики. Вони дозволяють формалізувати рішення, перевести його в алгоритм та оцінити, яка ймовірність отримати результат. Оскільки

зараз є можливість застосовувати величезні обчислювальні потужності, тому вміння програмувати є важливим для побудови моделей.

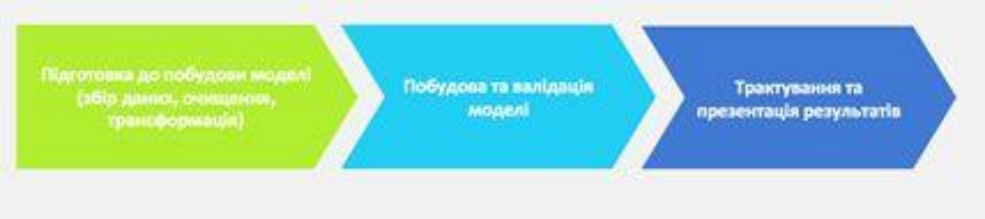


Рис. 1.2 Етапи аналізу даних

Процес аналізу даних складається з трьох етапів (рис. 1.2). Спочатку дані потрібно підготувати, тобто зібрати, очистити та відібрати ті, які потрібні для моделі. Далі ми будуємо модель та валідуємо її результати. Останній етап – це презентація результатів. Тут ми демонструємо на яке питання ми шукали відповідь, які дані використовували та що отримали в результаті.

1.2. Статистика

Статистика – це наука про навчання на основі даних, про вимірювання, контроль та тлумачення невизначеності; має важливе значення для управління ходом наукових і соціальних досягнень.

Статистика допомагає оцінити варіативність та зменшити невизначеність. Розрізняють описову та вивідну статистики.

- Описова – вивчає властивості спостережуваних даних.
- Вивідна статистика – виводимо припущення про властивості розподілу даних з яких походять спостережувані дані.

З допомогою статистики можна дати відповідь на наступні питання

- чи є залежність між кількістю злочинів та фазою Місяця?
- яка ймовірність викликати Uber в Києві?
- побудувати довірчий інтервал часу, за який ви потрапляєте на роботу.
- проводити опитування та трактувати їх результати.

1.3. Дані, рівні виміру, матриця даних, частотні таблиці, центральна тенденція

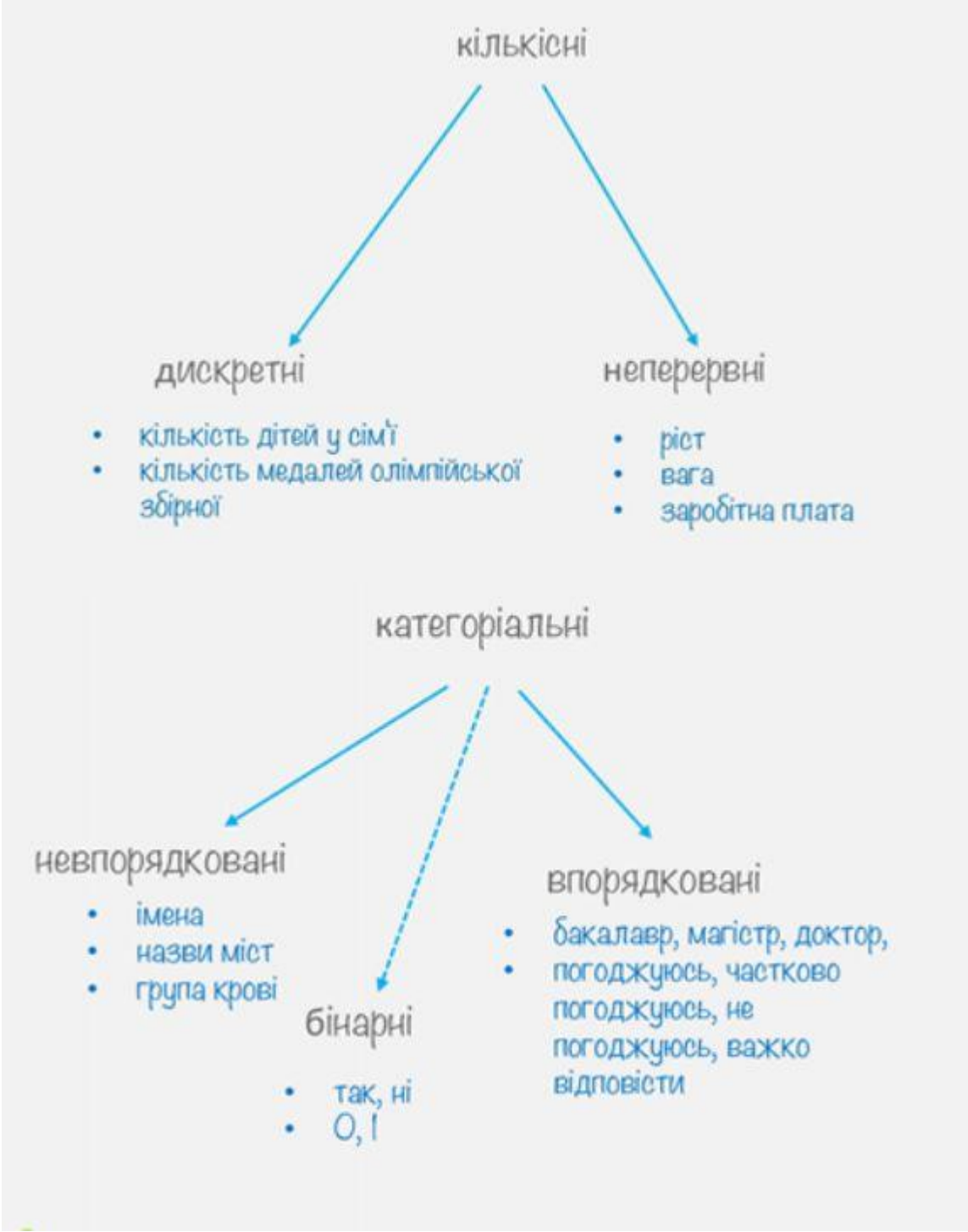


Рис. 1.3 Типи даних

Дані поділяються на кількісні та категоріальні. Кількісні, у свою чергу, поділяються на дискретні та неперервні, категоріальні – на невпорядковані, впорядковані та бінарні (рис. 1.3).

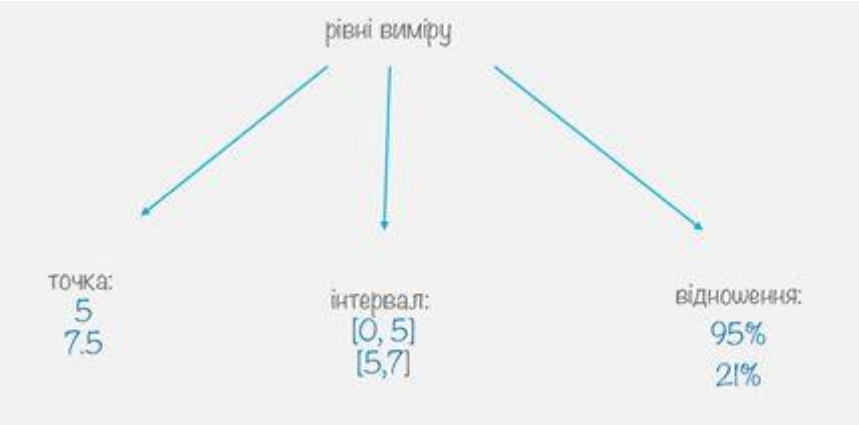


Рис. 1.4 Рівні виміру даних

Розрізняють три рівні виміру – це точка, інтервал та відношення (рис. 1.4).

Матриця даних – стартовий елемент для аналізу даних. Зазвичай йому передує етап збору, очищення та представлення у табличному вигляді. По рядках – респонденти, суб’єкти, учасники, спостереження По стовпцях – характеристики кожного запису(змінні). Також важливо звертати увагу на одиниці виміру, а також яким чином були зібрані ці дані.

Таблиця 1.1

Квартири, які продаються у Львові

Місто	Кількість кімнат	Загальна площа	Ціна
Львів	3	120	1875000
Львів	3	66	975000
Львів	2	66	1375000
Львів	2	44	637500
Львів	3	63	835000
Львів	1	31	562500

Табл. 1.1 включає в себе 6 рядків, однак зібрані нами дані мають майже 800 спостережень(спостереження зібрані з ресурсу і містять інформацію про квартири, які продаються). Для того, щоб описати вміст цієї таблиці в більш зрозумілій формі використовують узагальнення та опис типових чи середніх значень. Для цього важливо знати тип даних.

Таблиця 1.2

Квартири, які продаються

Місто	Кількість квартир
Вінниця	275
Дніпропетровськ	18
Запоріжжя	13
Івано-Франківськ	47
Києво-Святошинський район	19
Київ	186
Львів	16
Миколаїв	15
Одеса	43
Рівне	23
Тернопіль	93
Харків	14
Хмельницький	77

Для узагальнення категоріальних даних використовують частотні таблиці. Табл. 1.2 містить кількість квартир, що продаються у кожному місті.

Центральне або типове значення (центральна тенденція) дозволяє зрозуміти основну характеристику даних.

Середнє значення підходить для узагальнення кількісних даних(як дискретних, так і неперервних). Формула дляобчислення має наступний вигляд

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n},$$

де x_i –значення величини X, п – кількість пар значень, $i = \overline{1,n}$.

Тобто ми суму всіх чисел, ділимо на їх кількість. Наприклад, якщо в нас є група з 5 учнів, оцінки яких 12, 3, 5, 10, 5. Сума їх оцінок дорівнює 35, а середнє значення 7. Однак із використанням середнього значення в якості опису центральної тенденції в даниї є невелика проблема. Якщо є нетипово великі чи малі для даного набору значення – вони роблять великий внесок у

значення середнього. Наприклад, у нас є певне невелике підприємство, яке має 5 працівників. Заробітні плати працівників в гривнях: 5000, 7000, 2000, 4000, 50 000. Середнє значення заробітної плати 13600 грн. Однак, якщо ми відкинемо екстремальне значення 50 000, то отримаємо, що середнє значення зменшилося до 4500.

Медіана – це значення, яке ділить вибірку навпіл, тобто 50% є меншими за це значення, 50% більшими. Основна перевага використання медіани – менша чутливість до екстремальних значень. Для пошуку медіани, дані треба розташувати в зростаючому порядку та поділити на дві частини. Якщо в нас парна кількість спостережень то сусідні значення по краях сумуються та діляться на два. У випадку попереднього прикладу із заробіною платою: 2000, 4000, 5000, 7000, 50 000 Маємо, що посередині знаходиться значення 5000, то краще описує центральну тенденцію заробітної плати на підприємстві.

Що робити, якщо дані не є кількісними?

Мода використовується для визначення центральної тенденції категоріальних або кількісних дискретних даних. **Мода - це значення, яке найчастіше трапляється.** Наприклад, за інформацією міністерства юстиції України, минулого року хлопчиків найчастіше називали Дмитром, Артемом, Максимом та Іваном, дівчаток – Анею,Анастасією, Софією та Дар'єю. Ці імена є модою серед всіх імен.

1.4. Візуальний аналіз. Типи діаграм.

В 1848 році в лондонському районі Сохо було зафіксовано спалах холери, під час якого загинуло 616 жителів. Під час цього спалаху Лікар епідеміолог Джон Сноу на основі візуального аналізу даних зробив припущення, що джерелом зараження є вода.

Проблема вакцинації та відмови від вакцинації актуальна не лише для України. Після опублікованого у 1998 році в британському медичному журналі дослідження де щеплення були вказані як причина аутизму кількість батьків,

які відмовляються від щеплень збільшилась. Дослідження визнали помилковим, однак це не вплинуло на зростання кількості жителів США, які відмовлялися робити щепленння. Видання WSJ підготувало інтерактивні візуалізації, які відображають рівень захворюваності на кір, поліомієліт, кашлюк та інші хвороби до і після запровадження вакцини. У [1] дані показують рівень захворюваності у США протягом 80 років.

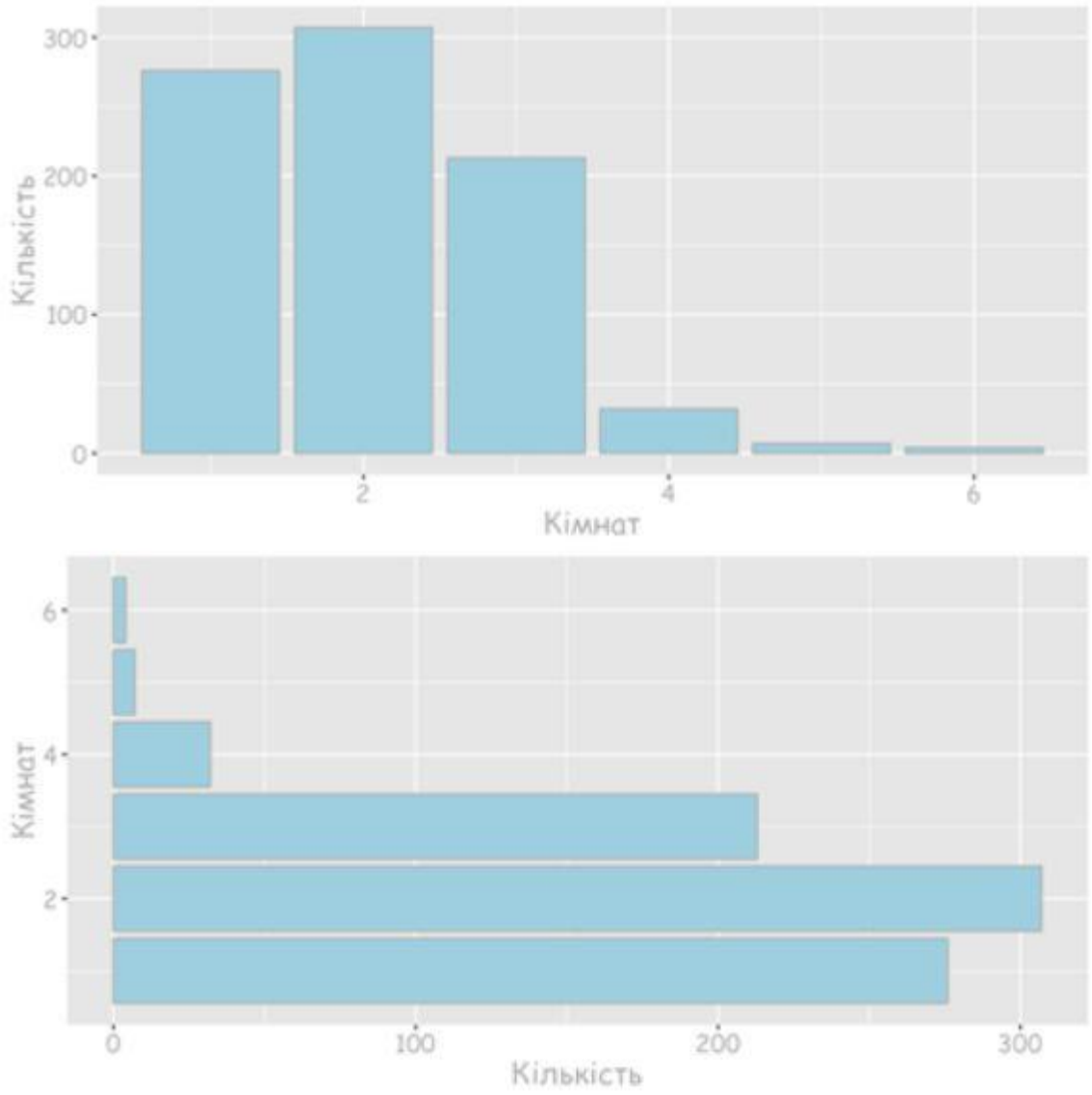


Рис. 1.5 Стовпчикова діаграма

Стовпчикова діаграмавикористовується для візуалізації категоріальних або кількісних дискретних даних (рис. 1.5).

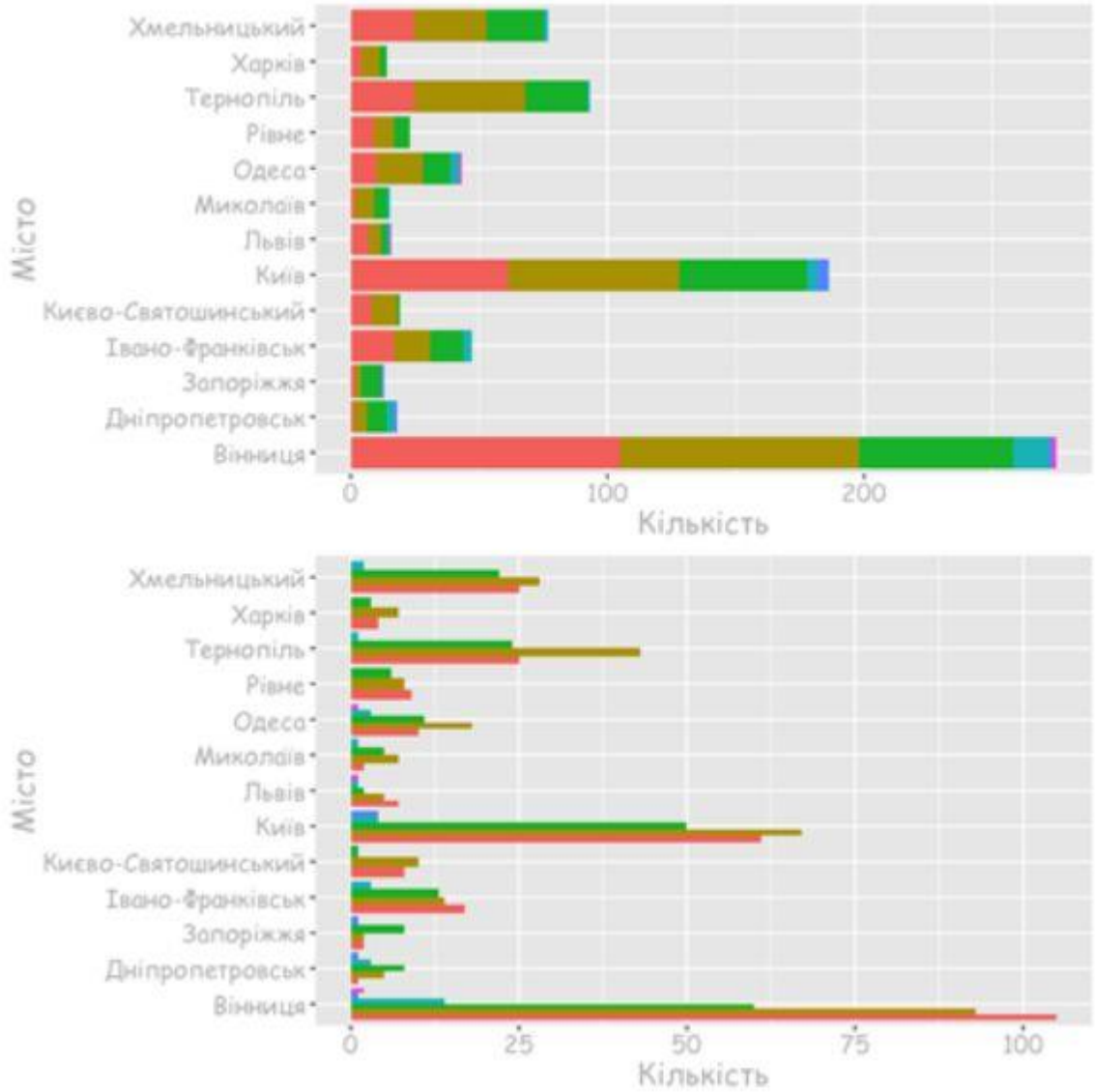


Рис. 1.6 Стовпчикова діаграма для двох змінних

Стовпчикова діаграма для двох змінних. Маємо категоріальну змінну місто та дискретну змінну кількість кімнат. Залежно від того, чи нас цікавить розуміння внеску кожної категорії чи порівння категорій між собою вибираємо тип візуалізації (рис. 1.6).

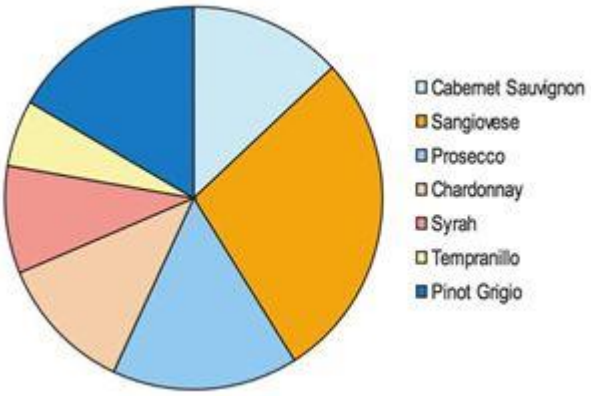


Рис. 1.7 Кругова діаграма

Кругова діаграма Використовується для візуалізації категоріальних або кількісних дискретних даних з метою зрозуміти відношення складових до загального значення (рис. 1.7). Наприклад, ми аналізуємо дохід від продажу вина і хочемо зрозуміти частку кожного сорту в загальному продажі.

Якщо ми захемо порівняти Caberne Sauvignon та Prosecco, то оцінити різницю між ними досить тяжко.

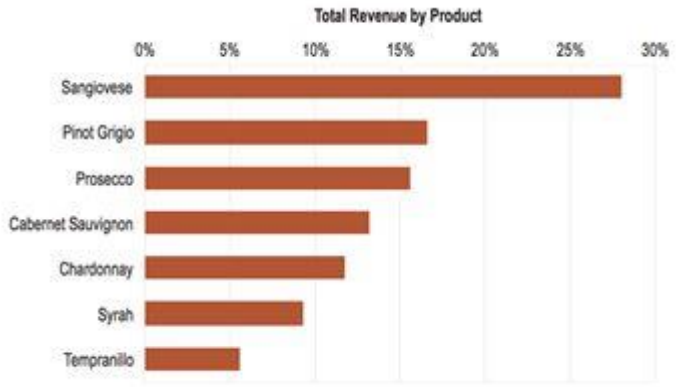


Рис. 1.8Стовпчикова діаграма у відсотках

Ці ж самі дані моджуть бути візуалізовані з допомогою стовпчикової діаграми (рис. 1.8).

Звідси зрозуміло, що різниця між доходом від продажу Caberne Sauvignon та Prosecco складає приблизно 3%. Власне, популярна у R бібліотека ggplot2 навіть не має фунціоналу для кругової діаграми.

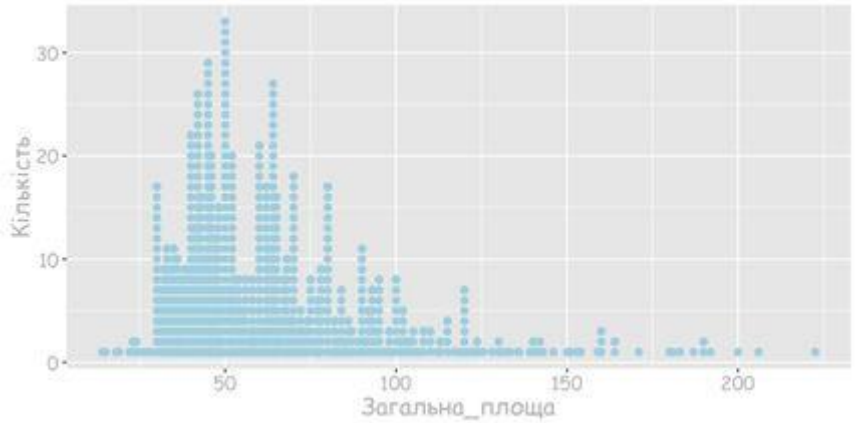


Рис. 1.9 Точковий графік

Точкові графіки.Кожна точка на графіку репрезентує одне спостереження (рис. 1.9). Це є приклад не зовсім вдало підібраного графіку, оскільки візуалізується загальна площа квартир, що продаються.

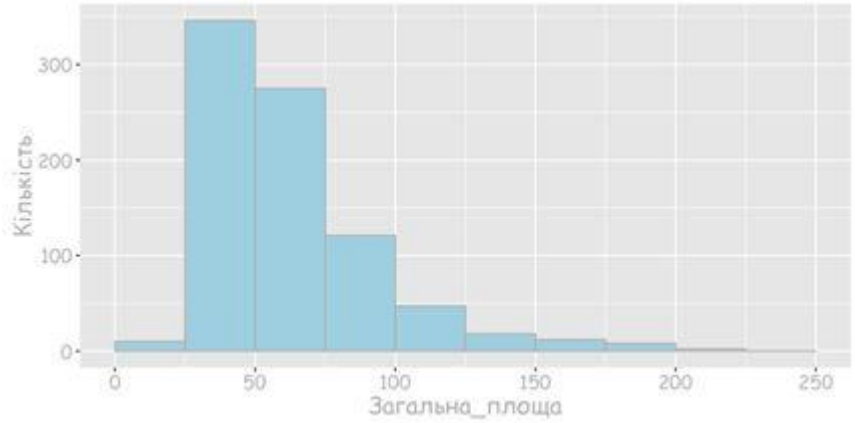


Рис. 10 Гістограма залежності кількості квартир від загальної площі

Загальна площа – неперервна кількісна змінна. Для її візуалізації краще використовувати гістограми (рис. 1.10). А точкові діаграми краще підходять для візуалізації дискретних даних.

Гістограмавикористовується для оцінки форми розподілу кількісної змінної. На графіку, що зображений на рис. 1.10 відображено розподіл квартир, які продаються за загальною площею з інтервалом 25.

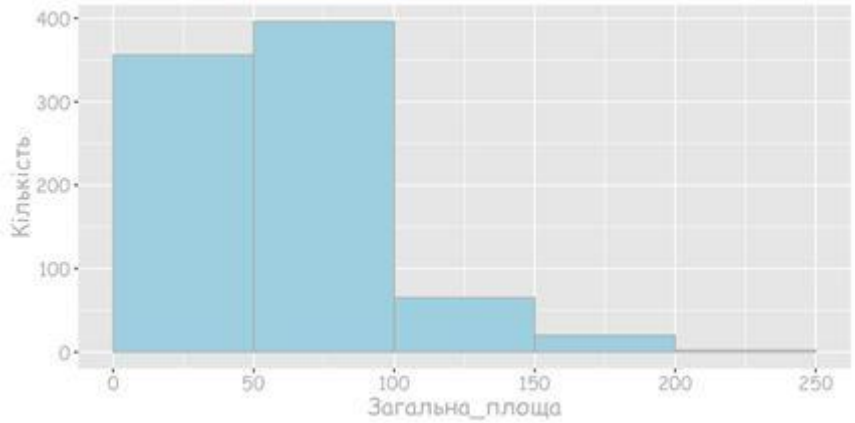


Рис. 1.11 Гістограма залежності кількості квартир від загальної площі з інтервалом 50 м.

Залежно від розміру інтервалу її форма може змінюватися. Наприклад змінимо інтервал з 25 метрів квадратних до 50 (рис 11).

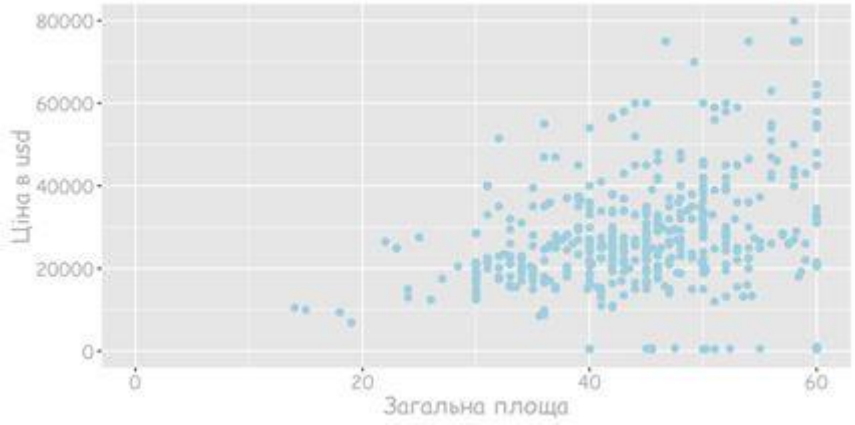


Рис. 1.12 Діаграма розсіювання двох кількісних змінних

Діаграма розсіюваннявикористовується для оцінки зв'язку двох кількісних змінних (рис. 1.12).

Рис. 1.13 Лінійний графік для оцінки однієї змінної за часом

Лінійний графікможе використовуватись для оцінки зміни однієї чи кількох змінних за часом(інформація Державної служби статистики [2], рис. 1.13).

При виборі типу графіка для візуалізації потрібно знати тип даних та те, що самепотрібно зрозуміти:

- **Порівнювати значення**: стовпчикова діаграма, лінійний графік, графік розсіювання.
- **Зрозуміти композицію(виділити складові)**: стовпчикова діаграма, кругова діаграма.
- **Оцінити розподіл даних**: лінійний графік, графік розсіювання, стовпчикова діаграма, гістограма.
- **Зрозуміти тренд**: лінійний графік, стовпчикова діаграма.
- **Зрозуміти відношення між даними**: лінійний графік, графік розсіювання.

Трактування результатів чи не найважливіша частина дослідження. Невірне трактування результатів дослідження дозволяє здійснювати маніпуляції. Детальніше про це можна прочитати у книзі Darell Huff "How to Lie with Statistics", яка була видана ще в 1954 році. Однак, іноді складається ситуація, що й правильне дослідженне може мати двояке трактування.

1.5. Парадокс Сімпсона

Парадокс Сімпсона названо на честь дослідника Едварда Сімпсона, який у 1951 описав цей феномен. Хорошою ілюстрацією буде ситуація, що склалася в університеті Берклі в 1973. Тоді університет звинуватили в гендерній нерівності.

Таблиця 1.3

Факультет А

Стать	Подано заяв	Прийнято заяв	Відсоток прийнятих
Чоловіки	900	450	50
Жінки	100	80	80

Таблиця 1.4

Факультет В

Стать	Подано заяв	Прийнято заяв	Відсоток прийнятих
Чоловіки	100	10	10
Жінки	900	180	20

Для ілюстрації ми дещо спростимо вихідні умови. Наприклад, в університеті є всього два факультети: А (табл. 1.3) та В (табл. 1.4).

Таблиця 1.5

Разом факультети А та В

Стать	Подано заяв	Прийнято заяв	Відсоток прийнятих
Чоловіки	1000	460	46
Жінки	1000	260	26

Якщо ми подивимось на відсоток прийнятих окремо по факультетах А та В то можемо зробити висновок, що дискримінують чоловіків. Однак, якщо об'єднати результати кількості прийнятих по факультетах, то ситуація виявиться зовсім іншою (табл. 1.5).

Лекція 2.

ОСНОВИ R. ТИПИ ДАНИХ. ВЕКТОРИ. МАТРИЦІ

План

2.1. Вступ

2.2. Типи даних

2.3. Вектори

2.4. Матриці

2.1. Вступ

Мова R належить до сімейства так званих високорівневих об'єктно-орієнтованих мов програмування. Для неспеціаліста суворе визначеня поняття «об'єкт» є досить абстрактним. Однак для простоти можна називати об'єктами все, що було створено в процесі роботи з R.

2.2. Типи даних

Виділяють два основних типи об'єктів:

1) об'єкти, призначені для зберігання даних («data objects») – це окремі змінні, вектори, матриці та масиви, списки, фактори, таблиці даних;

2) функції («function objects») – це іменовані програми, які призначенние для створення нових об'єктів або виконання певних дій над ними.

Об'єкти середовища R, призначені для колективного і вільного використання, часто комплектуються в пакети, що об'єднуються подібною тематикою або методами обробки даних. Є певна відмінність між термінами пакет («package») і бібліотека («library»). Термін «library» визначає директорію, яка може містити один або декілька пакетів. Термін «package» означає сукупність функцій, html-сторінок, посібників і наборів даних, призначених для тестування або навчання.

R працює з наступними елементарними типами даних (вони ж є класами відповідних змінних):

- **numeric** – змінні, що містять цілочисельні (integer) і дійсні числа (double);
- **logical** – змінні, що містять логічні значення: FALSE (скорочено F) і TRUE (T);
- **character** – символьні змінні (окремі значення таких змінних задаються в подвійних або одинарних лапках).

Імена різним об'єктам можна привсвоювати як на латиниці, так і на кирилиці, але слід врахувати, що а (кирилиця) і a (латиниця) – це два різних символи. Щоб уникнути плутанини і інших (пов'язаних з кодуванням) проблем рекомендується використовувати тільки латиницю. Крім того, середовище R чутливе до регістру, тобто маленькі і великі літери в ньому розрізняються. Ім'я змінної має починатися з літери (іноді дозволяєтьсякрапка), за якою слідує поєднання з букв, цифр, символів крапки і нижнього підкреслення. За допомогою команди? <Ім'я> можна перевірити, чи існує змінна або функція із зазначеним ім'ям.

Перевірка на приналежність змінної до певного класу здійснюється функціями is.numeric(<ім’я_об’єкту>), is.integer(<ім’я_об’єкту>), is.logical(<ім’я_об’єкту>), is.character(<ім’я_об’єкту>), а для перетворення об'єкта в інший тип використовуються функції as.numeric(<ім’я_об’єкту>), as.integer(<ім’я_об’єкту>), as.logical(<ім’я_об’єкту>), as.character(<ім’я_об’єкту>).

Змінним в R можуть присвоюватись наступні спеціальні значення:

- Inf – позитивна чи негативна нескінченність (зазвичай результат ділення дійсного числа на 0);
- NA – «відсутнє значення» («Not Available»);
- NaN – "не число» («Not a Number»).

Перевірити, чи відноситься те або інше значення конкретної змінної до до будь-якого з перелічених вище спеціальних типів, можна функціями

is.finite(<ім’я_об’єкту>), is.na(<ім’я_об’єкту>) і is.nan(<ім’я_об’єкту>) відповідно. Ці три функції повертають вектор з логічних значень TRUE або FALSE, який відображає результат перевірки.

Вираз («expression») мови R являє собою поєднання таких елементів, як оператор присвоєння, арифметичні або логічні оператори, імена об'єктів і імена функцій. Результат виконання виразу, як правило, відразу відображається в командному або графічному вікні. Однак при виконанні операції присвоювання результат зберігається у відповідному об'єкті і на екран не виводиться.

В якості оператора присвоєння в R можна використовувати символ = або пару символів <- (присвоєння певного значення об'єкту ліворуч) або -> (присвоєння значення об'єкту праворуч). Хорошим стилем програмування вважається використання <-.

Вирази мови R організовуються в скрипті за рядками. В одному рядку можна ввести декілька команд, розділяючи їх символом ;. Одну команду можна також розташувати на декількох рядках.

Об'єкти типу numeric можуть утворювати вираз з використанням традиційних арифметичних операцій + (додавання), - (віднімання), * (множення), / (ділення), ^ (піднесення до степеня),%/% (цілочисельне ділення), %% (залишок від ділення). Операції мають звичайний пріоритет, тобто спочатку виконується піднесення до степеня, потім множення або ділення, а потім вже додавання чи віднімання. У виразах можуть використовуватися круглі дужки, і операції, які містяться в них мають найвищий пріоритет.

Логічні вирази можна утворювати з використанням наступних логічних операторів:

- «дорівнює» ==;
- «Не дорівнює» !=;
- «Менше» <;
- «Більше»>;
- «Менше або дорівнює» <=;

- «Більше або дорівнює»>=;
- «Логічне І» &;
- «Логічне АБО» |;
- «Логічне НЕ»!.

2.3. Вектори

Вектор є іменним одновимірним об'єктом, що містить набір однотипних елементів (числові, логічні або текстові значення –їх поєднання не допускається). Для створення векторів невеликої довжини в R використовується функція конкатенації c() (від «concatenate» – об'єднувати, пов'язувати). В якості аргументів цієї функції через кому перелічують значення, які об'єднуються у вектор, наприклад:

```
my.vector <- c(1, 2, 3, 4, 5)
my.vector
[1] 1 2 3 4 5
```

Вектор можна створити також за допомогою функції scan(), яка послідовно «зчитує» значення, які вводяться з клавіатури:

```
X <- scan()
1: 2.9 # після кожного нового значення натиснути клавішу "Введення"
2: 3.1
3: 3.4
4: 3.4
5: 3.7
6: 3.7
7: 2.8
8: 2.5
9: # виконання команди scan завершують введенням порожнього рядка
Read 8 items # програма повідомляє про зчитування 8 значень
X
[1] 2.9 3.1 3.4 3.4 3.7 3.7 2.8 2.5
```

Один з недоліків створення векторів за допомогою функції scan() полягає в тому, що якщо при введенні значень з клавіатури допущена помилка, то доводиться або почати введення заново, або скористатися спеціальними інструментами коригування (наприклад, функцією fix(); ми не будемо розглядати ці способи). Для створення векторів, що містять послідовну сукупність чисел, зручна функція seq() (від «sequence» – послідовність). Так, вектор з ім'ям S, що містить сукупність цілих чисел від 1 до 7, можна створити наступним чином:

```
S <- seq(1,7)
S
[1] 1 2 3 4 5 6 7
```

Ідентичний результат можна отримати за допомогою команди

```
S <- 1:7
S
[1] 1 2 3 4 5 6 7
```

Як додатковий аргумент функції seq() можна вказати крок з яким будуть збільшуватись числа:

```
S <- seq(from = 1, to = 5, by = 0.5)
S
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Вектори, що містять однакові значення, можна створювати за допомогою функції rep() (від «repeat» – повторювати). Наприклад, для формування текстового вектора Text, що містить п'ять значень "test", потрібно виконати команду

```
Text <- rep("test", 5)
Text
[1] "test" "test" "test" "test" "test"
```

У системі R можна виконувати найрізноманітніші операції над векторами. Так, декілька векторів можна об'єднати в один, використовуючи вже розглянуту функцію конкатенації:

```
v1 <- c(1, 2, 3)
v2 <- c(4, 5, 6)
V <- c(v1, v2)
V
[1] 1 2 3 4 5 6
```

Якщо спробувати об'єднати, наприклад, текстовий вектор з числовим, з'явиться повідомлення про помилку – в результаті програма просто перетворює всі значення в текстові:

```
# створюємо текстовий вектор text.vector:
text.vect <- c("a", "b", "c")
# об'єднуємо числовий вектор v1 з текстовим вектором text.vector:
new.vect <- c(v1, text.vect)
# перегляд вмісту нового вектора new.vect:
new.vect
[1] "1" "2" "3" "a" "b" "c"
# Для підтвердження що всі елементи вектора текстового типу
# скористаємося командою class ():
class(new.vect)
# "character"
```

Для того, щоб отримати доступ до певного елемента вектора необхідно мати спосіб відрізняти його від інших елементів. Для цього при створенні вектора всім його компонентам автоматично присвоюються індексні номери, починаючи з 1. Щоб звернутися до конкретного елементу, необхідно вказати ім'я вектора і індекс цього елемента в квадратних дужках:

```
# створимо числовий вектор y, що містить 5 числових значень:
y <- c(5, 3, 2, 6, 1)
# перевіримо, чому дорівнює третій елемент вектора y:
y[3]
[1] 2
```

Використовуючи індексні номери, можна виконувати різні операції з обраними елементами різних векторів:

```
z <- c(0.5, 0.1, 0.6)
y[1]*z[3]
[1] 3
```

Індексування є потужним інструментом, що дозволяє створювати сукупності значень відповідно до визначених критеріїв. Наприклад, для виведення на екран 3-го, 4-го і 5-го значень вектора у необхідно виконати команду

```
y[3:5]
[1] 2 6 1
```

З цього ж вектора ми можемо вибрати, наприклад, тільки перше і четверте значення, використовуючи вже відому нам функцію конкатенації c():

```
y[c(1, 4)]
[1] 5 6
```

Схожим чином ми можемо видалити перше і четверте значення з вектора у, застосувавши знак «мінус» перед функцією конкатенації:

```
y[-c(1, 4)]
[1] 3 2 1
```

В якості критерію для вибору значень можнавмкористовувати логічний вираз. Для прикладу виберемо з вектора у усі значення> 2:

```
y[y > 2]
[1] 5 3 6
```

Індексування також є зручним інструментом для внесення змін до уже наявних векторів. Наприклад, так можна виправити друге значення створеного раніше вектора z з 0.1 на 0.3:

```
z[2] <- 0.3
z
[1] 0.5 0.3 0.6
```

Для упорядкування значень вектора за зростанням або спаданням використовують функцію sort() в поєднанні з аргументом decreasing = FALSE або decreasing = TRUE відповідно (за замовчуванням decreasing = FALSE):

```
sort(z)
[1] 0.3 0.5 0.6
sort(z, decreasing = TRUE)
[1] 0.6 0.5 0.3
```

2.4. Матриці

Матриця являє собою двовимірний вектор. В R для створення матриць використовують одноіменну функціюmatrix():

```
my.mat <- matrix(seq(1, 16), nrow = 4, ncol = 4)
my.mat
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

Варто звернути увагу на те, що за замовчуванням заповнення матриці відбувається за стовпцями, тобто перші чотири значення входять в перший стовпець, наступні чотири значення – у другий стовпець і т. д. Такий порядок заповнення можна змінити, надавши спеціальному аргументу byrow (від «by row »- за рядками) значення TRUE:

```
my.mat <- matrix(seq(1, 16), nrow = 4, ncol = 4, byrow = TRUE)
my.mat
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
```

Як заголовки рядків і стовпців створюваної матриці автоматично виводяться відповідні індексні номери (рядки: [1,], [2,] і т. д.; стовпці: [, 1], [2] і т. д.). Для присвоєння заголовкам рядків і стовпців матриць імен, які призначені для користувача, використовують функції rownames() і colnames() відповідно. Наприклад, для позначення рядків матриці my.mat буквами A, B, C і D необхідно виконати наступне:

```
rownames(my.mat) <- c("A", "B", "C", "D")
my.mat
  [,1] [,2] [,3] [,4]
A    1    2    3    4
B    5    6    7    8
C    9   10   11   12
D   13   14   15   16
```

У матриці my.mat є 16 значень, які поміщаються в наявні чотири рядки і чотири стовпці. Але що трапиться, якщо, наприклад, спробувати вмістити вектор з 12 чисел в матрицю такого ж розміру? У подібних випадках R заповнює відсутні значення за рахунок «зациклення» («recycling») короткого вектора. Покажемо як це виглядає на прикладі:

```
my.mat2 <- matrix(seq(1, 12), nrow = 4, ncol = 4, byrow = TRUE)
my.mat2
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]    1    2    3    4
```

Як бачимо, для заповнення останнього рядка матриці my.mat2 знову використовуються числа 1, 2, 3 і 4.

Альтернативний спосіб створення матриць полягає в застосуванні функції dim() (від «dimension» – вимірність). Так, матрицю my.mat можна сформувати з одновимірного вектора наступним чином:

```
dim(my.mat) <- c(4, 4)
my.mat
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

Функція dim() дозволяє перевірити вимірність уже наявної матриці:

```
dim(my.mat)
[1] 4 4
```

Матрицю можна зібрати також з декількох векторів, використовуючи функції cbind() (від «column» – стовпець і «bind» – пов'язувати) або rbind() (від «row» – рядок і «bind» – пов'язувати):

```
# Створимо чотири вектора однакової довжини:
a <- c(1, 2, 3, 4)
b <- c(5, 6, 7, 8)
d <- c(9, 10, 11, 12)
e <- c(13, 14, 15, 16)
# Об'єднаймо ці вектори за допомогою функції cbind():
cbind(a, b, d, e)
     a  b  d  e
[1,] 1  5  9  13
[2,] 2  6  10 14
[3,] 3  7  11 15
[4,] 4  8  12 16
#Об'єднаймо ці вектори за допомогою функції rbind():
  [,1] [,2] [,3] [,4]
A    1    2    3    4
B    5    6    7    8
D    9   10   11   12
```

E 13 14 15 16

Практично всі векторні операції однаково застосовуються щодо матриць і масивів. Так, за допомогою індексу ми можемо отримувати з матриць необхідні елементи і далі здійснювати над ними необхідні перетворення. Розглянемо декілька прикладів:

```
# Візьмемо елемент матриці my.mat, розташований на перетині
# 2-го рядка і 3-го стовпця:
my.mat[2, 3]
[1] 7
# Візьмемо з матриці всі елементи, що знаходяться в 4-му стовпці
my.mat[, 4]
[1] 4 8 12 16
# Візьмемо з матриці всі елементи, що знаходяться в 1-му рядку
my.mat[1, ]
[1] 1 2 3 4
# Перемножимо 1-й і 4-й стовпці матриці (поелементно):
my.mat[, 1]*my.mat[, 4]
[1] 4 40 108 208
```

При необхідності матрицю можна транспонувати (тобто поміняти місцями рядки і стовпці) за допомогою функції t() (від «transpose»):

```
t(my.mat)
     A  B  C  D
[1,] 1  5  9  13
[2,] 2  6  10 14
[3,] 3  7  11 15
[4,] 4  8  12 16
```

Лекція 3.

ОСНОВИR.ФАКТОРИ, СПИСКИ І ТАБЛИЦІ ДАНИХ

План

3.1. Вступ

3.2. Фактори

3.3. Списки і таблиці даних

3.1. Вступ

У статистиці дані дуже часто групують відповідно до тієї чи іншої ознаки, наприклад статі, соціального становища, стадії хвороби, місця відбору проб і т. д. В R існує спеціальний клас векторів – фактори («factors»), які призначені для зберігання міток відповідних рівнів номінальних змінних. Часто рівні факторів кодують у вигляді чисел. У таких випадках дуже важливо проінструктувати програму так, щоб вона правильно «розпізнавала» рівні номінальної змінної від чисел як таких.

3.2. Фактори

Припустимо, що в експерименті з випробування ефективності нового медичного препарату було задіяно 10 пацієнтів-добровольців, з яких шість пацієнтів приймали новий препарат, а четверо інших – плацебо (наприклад, таблетки активованого вугілля). Для позначення членів цих двох груп ми можемо використовувати мітки 1 (препарат) і 0 (плацебо). Відповідно, інформацію про всі десять учасників експерименту ми могли б зберегти за допомогою наступного вектора:

```
treatment <- c(1, 1, 1, 1, 1, 1, 0, 0, 0, 0)
treatment
[1] 1 1 1 1 1 1 0 0 0 0
```

Однак при такому підході, програма буде «розглядати» вектор treatment як числовий вектор (можна перевірити за допомогою оператора class(treatment)).

Проте це буде помилкою, оскільки нуль і одиниця позначають лише два рівня номінальної змінної. З таким же успіхом ми могли б використовувати, наприклад, 10 для позначення контрольної групи пацієнтів (тобто пацієнтів, які приймали плацебо) і 110 для позначення пацієнтів, що приймали тестований препарат. Для перетворення числового (або будь-якого іншого) вектора в фактор в R існує одноіменна функція factor():

```
treatment <- factor(treatment, levels = c(0, 1))
treatment
[1] 1 1 1 1 1 1 0 0 0 0
Levels: 0 1
```

Тепер при виведенні вмісту об'єкта treatment програма підказує нам, що цей об'єкт є фактором з двома рівнями (Levels: 0 1). Додатково переконатисьу цьому можна за допомогою команди class(treatment):

```
class(treatment))
[1] "factor"
```

Більш надійним підходом, що дозволяє не дезорієнтуватись при виконанні аналізу, є кодування рівнів факторів за допомогою текстових значень, а не чисел. Наприклад, в нашому прикладі можна присвоєти значення yes пацієнтам, які брали препарат, і значення no пацієнтам з контрольної групи. Ми можемо перекодувати рівні вже наявного фактора treatment за допомогою функції levels():

```
levels(treatment) <- c("no", "yes")
treatment
[1] yes yes yes yes yes yes no no no no
Levels: no yes
```

При виведенні вмісту вектора treatment коди пацієнтів не взяті в подвійні лапки, як у випадку з текстовими значеннями. Це є однією із зовнішніх ознак того, що ми маємо справу саме з фактором, а не з текстовим вектором, що містить шість значень "yes" і чотири значення "no". Фактор легко перетворити назад в числовий вектор, що складається з порядкових номерів рівнів фактора:

```
as.numeric(treatment)
[1] 2 2 2 2 2 2 1 1 1 1
```

Існує також спеціальна команда для створення факторів (від «generate levels» – згенерувати рівні):

```
gl(n, k, length = n*k, labels = 1:n),
```

де n – кількість рівнів фактора;

k –кількість повторів для кожного рівня;

length – розмір підсумкового об'єкта;

labels – необов'язковий аргумент, який можна використовувати для зазначення назв кожного рівня фактора.

Наприклад, виконання наступної команди призведе до створення вектора my.fac, що є фактором з двома рівнями – Control і Treatment, причому кожна з міток "Control" і "Treatment" буде повторена по 8 разів:

```
my.fac = gl(2, 8, labels = c("Control", "Treatment"))
my.fac
[1] Control Control Control Control Control Control Control
[8] Control Treatment Treatment Treatment Treatment Treatment Treatment
[15] Treatment Treatment
Levels: Control Treatment
```

Ще одна корисна команда створює фактори, розділяючи діапазон значень деякого числового вектора x на інтервали:

```
cut(x, breaks, labels),
```

де в якості аргументу breaks може виступати або необхідна кількість інтервалів, або вектор зі списком «точок розриву», а labels визначає назви рівнів:

```
x <- c(1, 2, 3, 4, 5, 2, 3, 4, 5, 6, 7)
cut(x, breaks = 3)
[1] (0.994,3] (0.994,3] (3,5] (3,5] (3,5] (0.994,3] (3,5]
[8] (3,5] (3,5] (5,7.01] (5,7.01]
Levels: (0.994,3] (3,5] (5,7.01]
```

```
cut(x, breaks = 3, labels = letters[1:3])
[1] a a b b b a b b b c c
Levels: a b c

cut(x, breaks = quantile(x, c(0, .25, .50, .75, 1)),
  labels = c("Q1", "Q2", "Q3", "Q4"), include.lowest = TRUE)
[1] Q1 Q1 Q2 Q2 Q3 Q1 Q2 Q2 Q3 Q4 Q4
Levels: Q1 Q2 Q3 Q4
```

Тут числовой вектор «розрізаний» по квартильним значенням, а параметр include.lowest вказано, щоб уникнути появи відсутніх значень (NA) для x = 1.

3.3. Списки і таблиці даних

На відміну від вектора або матриці, які можуть містити дані тільки одного типу, в список («list») або таблицю («data frame») можна включати поєднання будь-яких типів даних. Це дозволяє ефективно, тобто в одному об'єкті, зберігати різнотипну інформацію.

Кожен компонент списку може бути змінною, вектором, матрицею, фактором або іншим списком. Крім того, і самі ці елементи можуть належати до різних типів: числа, рядки символів, булеві змінні. Списки служать найбільш загальним способом зберігання внутрішньосистемної інформації.Варто зазначити, що результати більшості статистичних аналізів в R зберігаються в об'єктах-списках. Для створення списків використовують однойменну функцію list():

```
# Спочатку створимо три різнотипних вектора – з текстовими,
# числовими і логічними значеннями:
vector1 <- c("A", "B", "C")
vector2 <- seq(1, 3, 0.5)
vector3 <- c(FALSE, TRUE)
```

```
# Тепер об'єднаємо ці три вектори в один об'єкт-список,
# компоненти якого назвемо Text, Number і Logic:
my.list <- list(Text=vector1, Number=vector2, Logic=vector3)
my.list
$Text
[1] "A" "B" "C"
$Number
[1] 1.0 1.5 2.0 2.5 3.0
$Logic
[1] FALSE TRUE
```

До елементів списку можна отримати доступ за допомогою трьох різних операцій індексації. Для доступу до іменованих компонентів застосовують знак $. Так, для отримання компонентів Text, Number і Logic зі створеного нами списку my.list необхідно послідовно ввести наступні команди:

```
my.list$Text
[1] "A" "B" "C"

my.list$Number
[1] 1.0 1.5 2.0 2.5 3.0

my.list$Logic
[1] FALSE TRUE
```

Є можливість отримувати зі списку не тільки його іменовані компоненти-вектори, а й окремі елементи, що входять до цих векторів. Для цього необхідно скористатися вже розглянутим раніше способом – індексацією за допомогою квадратних дужок. Єдина особливість роботи зі списками тут полягає в тому, що спочатку необхідно вказати ім'я компонента списку, використовуючи знак $, а вже потім номер(и) окремих елементів цього компонента:

```
my.list$Text[2]
[1] "B"
```

```
my.list$Number[3:5]
[1] 2.0 2.5 3.0

my.list$Logic[1]
[1] FALSE
```

Отримання компонентів списку можна здійснювати також з використанням подвійних квадратних дужок, в яких вказується номер компонента списку:

```
my.list[[1]]
[1] "A" "B" "C"

my.list[[2]]
[1] 1.0 1.5 2.0 2.5 3.0

my.list[[3]]
[1] FALSE TRUE
```

Після подвійних квадратних дужок з індексним номером компонента списку можна також вказати номер(и) окремих елементів цього компонента:

```
my.list[[1]][2]
[1] "B"

my.list[[2]][3:5]
[1] 2.0 2.5 3.0

my.list [[3]][1]
[1] FALSE
```

Створений список my.list містив тільки три невеликих вектори, і відомо, які це вектори і на якому місці в списку вони стоять. Однак на практиці можна зіткнутися з набагато більш складно організованими списками, індексування

яких ускладнюється через відсутність уявлень про їх структуру. Для з'ясування структури об'єктів в мові R є спеціальна і дуже корисна функція str() (від «structure»):

```
str(my.list)
List of 3
$ Text : chr [1:3] "A" "B" "C"
$ Number: num [1:5] 1 1.5 2 2.5 3
$ Logic : logi [1:2] FALSE TRUE
```

З наведеного прикладу випливає, що список my.list включає 3 компоненти (List of 3) з іменами Text, Number і Logic (перераховані в окремих рядках після знака $). Ці компоненти відносяться до символьного (chr), числового (num) і логічного (logic) типів даних відповідно. Окрім цього, команда str() виводить на екран перші декілька елементів кожного вектора.

Таблиця даних є об'єктом R, що за структурою нагадує лист електронної таблиці Microsoft Excel. Кожен стовпець таблиці є вектором, що містить дані певного типу. При цьому діє правило, згідно з яким всі стовпці повинні мати однакову довжину (власне, з «точки зору» R таблиця даних є окремим випадком списку, в якому всі компоненти-вектори мають однаковий розмір).

Таблиці даних – це основний клас об'єктів R, які використовуються для зберігання даних. Зазвичай такі таблиці готуються за допомогою зовнішніх програм (особливо популярна і зручна програма Microsoft Excel) та потім завантажуються в середовище R. Проте невелику таблицю можна зібрати з декількох векторів засобами самої системи R. Для цього використовують функцію data.frame().

Припустимо що у нас є спостереження за загальною чисельністю чоловічого (Male) та жіночого (Female) населення в трьох містах City1, City2 і City3. Представимо ці дані у вигляді однієї таблиці з ім'ям CITY. Для початку створимо текстові вектори з назвами міст (City) і статі (sex), а також вектор зі значеннями чисельності представників кожної статі (number):

```
city <- c("City1", "City1", "City2", "City2", "City3", "City3")
```

```
sex <- c("Male", "Female", "Male", "Female", "Male", "Female")
number <- c(12450, 10345, 5670, 5800, 25129, 26000)
```

Тепер об'єднаємо ці три вектори в одну таблицю даних:

```
CITY <- data.frame(City = city, Sex = sex, Number = number)
CITY
```

	City	Sex	Number
1.	City1	Male	12450
2.	City1	Female	10345
3.	City2	Male	5670
4.	City2	Female	5800
5.	City3	Male	25129
6.	City3	Female	26000

Звернемо увагу на те, щоаргументи функції data.frame()задаютьсяу форматі «заголовок стовпця = вектор, що додається». В якості заголовків стовпців можуть використовуватись будь-які призначені для користувача імена, що відповідають вимогам R.

Як і у випадку зі списками, отримати окремі компоненти таблиць можна з використанням знака $, квадратних дужок із зазначенням двох індексів [<номер_рядка>, <номер_стовпчика>], подвійних квадратних дужок [[]] або безпосередньо імені стовпця:

```
CITY$Sex
[1] Male Female Male Female Male Female
Levels: Female Male

CITY$Number
[1] 12450 10345 5670 5800 25129 26000

CITY[, 2]
[1] Male Female Male Female Male Female
Levels: Female Male
```

```
CITY[[3]]
[1] 12450 10345 5670 5800 25129 26000

CITY["Sex"]
[1] Male Female Male Female Male Female
Levels: Female Male

CITY["Number"]
[1] 12450 10345 5670 5800 25129 26000
```

Після імені або індексного номера стовпця можна вказувати індексні номери окремих комірок таблиці, що дозволяє отримувати їх вмістиме:

```
# Отримуємо 4-й елемент із стовпця Number:
CITY$Number[4]
[1] 5800

# Отримуємо елементи 1 – 3 із стовпця Number:
CITY$Number[1:3]
[1] 12450 10345 5670

# Отримуємо всі значення Number, що перевищують 10000
CITY$Number[CITY$Number > 10000]
[1] 12450 10345 25129 26000

# Отримуємо всі значення чисельності чоловічого населення:
CITY$Number[CITY$Sex == "Male"]
[1] 12450 5670 25129

# Повторюємо ті ж команди, але з використанням []:
CITY[4, 3]
```

```
[1] 5800

CITY[1:3, 3]
[1] 12450 10345 5670

CITY[CITY$Number > 10000, 3]
[1] 12450 10345 25129 26000

CITY[CITY$Sex == "Male", 3]
[1] 12450 5670 25129
```

При роботі з великими таблицями даних буває складно візуально досліджувати все їхнє вмістиме перед початком аналізу. Однак візуального перегляду вмістимого таблиць і не потрібно – повну зведену інформацію про них (так само як і про інші об'єкти R) можна легко отримати за допомогою функції str():

```
str(CITY)
'data.frame':   6 obs. of 3 variables:
$ City : Factor w/ 3 levels "City1","City2",..: 1 1 2 2 3 3
$ Sex   : Factor w/ 2 levels "Female","Male": 2 1 2 1 2 1
$ Number: num 12450 10345 5670 5800 25129 ...
```

Як випливає з представленого звіту, об'єкт CITY є таблицею даних, до складу якої входять три змінні з шістьма спостереженнями кожна. Дві з цих змінних – City і Sex – програма автоматично розпізнала як фактори з трьома і двома рівнями відповідно. Змінна Number є кількісною. Для зручності виводяться також кілька перших значень кожної змінної.

Часто виникає необхідність з'ясувати тільки імена змінних, що входять в таблицю даних. Це можна зробити за допомогою команди names():

```
names(CITY)
[1] "City"      "Sex" "Number"
```

Є також можливість швидко переглянути декілька перших або декілька останніх значень кожної змінної, що входить до складу таблиці даних. Для цього використовуються функції head() і tail() відповідно:

```
head(CITY, n = 3)
  City   Sex    Number
1City1 Male  12450
2City1 Female        10345
3City2 Male  5670


tail(CITY, n = 2)
  City   Sex    Number
5City3 Male  25129
6City3 Female        26000
```

За необхідності внесення виправлень в таблицю можна скористатись вбудованим в R редактором даних. Зовні цей редактор нагадує звичайний лист Excel, однак має дуже обмежені функціональні можливості. Все, що він дозволяє робити, – це додавати нові або виправляти вже введені значення змінних, змінювати заголовки стовпців, а також додавати нові рядки і стовпці. Працюючи в стандартній версії R, редактор даних можна запустити з меню Редагувати> Редактор даних або виконавши команду fix() (від «fix» – виправляти, лагодити) з командного рядка (наприклад, fix(CITY)). Після внесення виправлень редактор просто закривають – всі зміни будуть збережені автоматично.

Лекція 4.

ОСНОВИ R. ФУНКЦІЇ, РОЗГАЛУЖЕННЯ ТА ЦИКЛИ. ВЕКТОРИЗОВАНІ ОБЧИСЛЕННЯ В R

План

4.1. Вступ

4.2. Написання власних функцій

4.3. Умови та цикли

4.4. Векторизовані обчислення в R

4.1. Вступ

Абсолютна більшість процедур обробки даних в R реалізується за допомогою функцій. Функції можна представити собі як програмний код, що має свою назву та складається з деякого набору змінних, констант, операторів та інших функцій і призначений для виконання конкретних операцій і завдань. Як правило (але не завжди), функції повертають результат свого виконання у вигляді об'єкта мови R – змінної певного класу: вектора, списку, таблиці і т. д.

За своїм призначенням функції можна розділити на характерні групи: арифметичні, символьні, статистичні та інші. Функції можуть бути вбудованими (тобто представленими в базових або підвантажуваних пакетах) і власними (тобто написаними безпосередньо самими користувачами).

4.2. Написання власних функцій

Трьома характерними рисами мови R як мови високого рівня є модульність побудови, орієнтація на роботу з об'єктами і векторизація обчислень. Під модульностью розуміється широке використання груп виразів і функцій. Вирази expr, що складаються з об'єктів даних, викликів функцій і операторів мови, можуть групуватися в фігурних дужках: {expr_1; ...; expr_m}. Значення, яке повертає ця група, являє собою результат виконання останнього виразу. Оскільки така група також є виразом, то вона може бути, наприклад,

включена в круглі дужки і використовуватися як частина ще більш загального виразу.

Представлена нижче група команд виконує розрахунок середнього і стандартного відхилення ряду чисел від 1 до 10 і повертає вектор з цих значень:

{aver <- mean(1:10); stdev <- sd(1:10); c(MEAN=aver, SD=stdev)}

MEANSD

5.50000 3.02765

Однак якщо цеобчисленнянеобхідно виконати неодноразово для різних наборів вихідних даних, то його варто оформити у вигляді функції. Загальний синтаксис оформлення будь-якої функції має наступний вигляд:

ім’я_функції <- function(argl, arg2, ...) {

група_операторів

return(object) },

де ім’я_функції – ім'я створюваної функції; argl, arg2, ... – аргументи функції, а return() повертає результати обчислень, збережених в об'єкті object. Залежно від ходу виконуваних обчислень, використовувати return() необов'язково.

Перед своїм першим виконанням функція повинна бути визначена в поточному скрипті або завантажена за допомогою команди source() з зовнішнього скриптового файлу.

Для представленого вище прикладу оформимо функцію (потрібно звернути увагу, що return() тут не використовується – результатом обчислень автоматично стане вектор з MEAN і SD, оскільки він створюється останнім)

stat_param <- function(x){

aver <- mean(x); stdev <- sd(x); c(MEAN = aver, SD = stdev)}

і включити її в колекцію власних функцій, що зберігаються, наприклад, у файлі my_func.R. Тоді необхідний результат, наведений вище, можна отримати, виконавши

source("my_func.R")

```
stat_param(1:10)
```

Аргументи функцій можуть бути обов'язковими і необов'язковими. Наприклад, наступна функція підносить числовий об'єкт x до степеня n, проте якщо степінь не вказана, то автоматично відбувається піднесення до кубу:

```
power <- function(x, n = 3){ x^n }
```

Аргументами функцій можуть бути об'єкти різного типу, наприклад назви інших функцій. Так, наступна функція виконує довільні перетворення випадкових рівномірно розподілених величин:

```
my_exampl <- function(n, func_trans)
{ x <- runif(n); abs(func_trans(x)) }
```

Тоді згенерувати 5 прологарифмованих значень можна, виконавши

```
my_exampl(5, log)
[1] 0.6345459 0.6522557 2.4180118 0.1007311 1.6983938
```

4.3. Умови та цикли

Як і в будь-якій мові програмування, в R широко використовуються розгалуження і цикли обчислювального процесу. Оператор умовного виконання команд if має наступну структуру:

```
if(логічний_вираз)
{група_виразів_1, якщологічний_вираздорівнює TRUE}
else {група_виразів_2 у протилежному випадку}
```

Наприклад, наступна функція порівнює розміри двох векторів:

```
compare <- function(x, y){ nl <- length(x); n2 <- length(y)
if(nl!= n2){
if(nl > n2){ z = (nl - n2)
cat("Перший вектор має на ", z," елементів більше \ n") }
else{ z = (n2 - nl)
cat("Другий вектор має на ", z," елементів більше \ n")}}
else{cat("Кількість елементів одинакова", nl, "\n") }
}
```

```
x <- c(1:4)
y <- c(1:9)

compare(x, y)
Перший вектор має на 5 елементів більше
```

Є також скорочена форма реалізації розгалужень:

```
ifelse(логічний_вираз,
група_операторів_1, група_операторів_2)
```

Повторення в циклі одних і тих же обчислювальних операцій здійснюється з використанням операторів for(), while() або repeat(), які мають наступний синтаксис:

```
for (index in for_object) {група_операторів}
while(логічний_вираз) { група_операторів}
repeat {група_операторів; break}
```

Тут об'єкт for_object може бути вектором, масивом, таблицею або списком, а група_операторів виконується кожен раз для кожного елемента index цього об'єкєкта.

4.4. Векторизовані обчислення в R

Використання оператора циклів for(), що характерно для мов низького рівня типу Basic, не вважається хорошим стилем програмування на мові R, яка орієнтована на векторизацію обчислень. Замість того щоб виконувати послідовно скалярні операції над кожним з елементів масиву, набагато ефективніше виконувати паралельні обчислення, при яких програма обробляє одночасно весь масив (вектор) цілком або по декілька елементів вектора в кожний момент часу. Очевидно, що такий підхід потенційно може привести до значного прискорення однотипних обчислень над великими масивами даних.

Розглянемо найпростіший приклад векторизованих обчислень в R. Припустимо, що у нас є вектор з 10 додатних чисел, і потрібно взяти

квадратний корінь з кожного із них. Замість написання циклу для почергового виконання цієї операції над кожним елементом достатньо просто подати весь цей вектор на вхід функції sqrt(), яка повертає вектор з результатами обчислень:

```
x <- 1:10
sqrt(x)
[1] 1.000 1.414 1.732 2.000 2.236 2.449 2.645 2.828 3.000 3.162
```

Принцип векторизованих обчислень можна застосувати не тільки до векторів, а й до більш складних об'єктів R – матриць, списків і таблиць даних. У базовій комплектації R є багато функцій, призначених для організації векторизованих обчислень над такими об'єктами. У назві всіх цих функцій є слово apply(«застосувати»), якому передує літера, яка вказує на принцип роботи тієї чи іншої функції. При цьому:

- apply(), на відміну від for(), можна легко распаралелити, задіявши всі наявні процесори (завдяки використанню «паралельної» версії цієї функції з пакету snow);
- алгоритми, записані без циклів, легше модифікуються, містять менше помилок і легше набираються в командному рядку;
- результат роботи apply() може бути аргументом функції, не кажучи вже про те, що будь-яка функція може бути використана в якості аргументу в apply().

Функція apply() використовується у випадках, коли необхідно застосувати будь-яку функцію до всіх рядків або стовпців матриці (або масивів більшої розмірності):

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

де x - це об'єкт над яким необхідно здійснити певні перетворення, MARGIN – індекс, що позначає напрямок процесу обчислень (по стовпцях або рядках), FUN – вживана для обчислень функція, а ... – це будь-які параметри, що керують поведінкою функції FUN. Для матриці або таблиці даних MARGIN = 1 позначає рядки, а MARGIN = 2 – стовпці. Оскільки FUN означає будь-яку

функцію R, в тому числі і самостійно написану, то функція apply()– це потужний засіб модульної обробки даних.

```
# Створимо звичайну матрицю:
M <- matrix(seq(1, 16), 4, 4)

# Знайдемо мінімальні значення в кожному рядку матриці
apply(M, 1, min)
[1] 1 2 3 4

# Знайдемо мінімальні значення в кожному стовпці матриці
apply(M, 2, max)
[1] 4 8 12 16

# Приклад з тривимірним масивом:
M <- array(seq(32), dim = c(4,4,2))

# Застосуємо функцію sum() до кожного елементу M[*,,],
# Тобто виконаємо додаванняза вимірюваннями 2 і 3:
apply(M, 1, sum)
# Результат – одновимірний вектор:
[1] 120 128 136144

# Застосуємо функцію sum() до кожного елементу M [*, *,],
# тобто виконаємо підсумовування по третій вимірності:
apply(M, c(1,2), sum)
# Результат – матриця:
[,1]    [,2]    [,3]    [,4]
[1,]    18      26      34      42
[2,]    20      28      36      44
[3,]    22      30      38      46
```

```
[4,]    24    32    40    48
```

При необхідності обчислення сум і середніх значень за рядками або стовпцями матриць рекомендується також використовувати спеціально оптимізовані для цього функції colSums(), rowSums(), colMeans() і rowMeans().

Функція lapply() використовується у випадках, коли необхідно застосуватидеяку функцію до кожного компоненту списку і отримати результат також у вигляді списку (буква «l» в назві lapply() означає «list» – список).

```
# Створимо список з трьома компонентами-векторами:
x <- list(a = 1, b = 1:3, c = 10:100)

# З'ясуємо розмір кожного компонента списку x (функція length()):
lapply(x, FUN = length)
$a
[1] 1
$b
[1] 3
$c
[1] 91

# Виконаємо підсумовування елементів в кожному компоненті списку x:
lapply(x, FUN = sum)
$a
[1] 1
$b
[1] 6
$c
[1] 5005
```

Функція sapply() використовується у випадках, коли необхідно застосувати певну функцію до кожного компонента списку, але результат

вивести у вигляді вектора (буква «s» в назві sapply() означає «simplify» – спрощувати).

```
# Список з трьох компонентів:
x <- list(a = 1, b = 1:3, c = 10:100)

# З'ясуємо розмір кожного компонента списку x:
sapply(x, FUN = length)
a b c
1 3 91 # результат у вигляді вектора

# Підсумовування всіх елементів в кожному компоненті списку x:
sapply(x, FUN = sum)
a    b  c
1    6  5005 # Результат повернений у вигляді вектора
```

Функція replicate() є узагальненням функції sapply() і дозволяє здійснити серію обчислень з метою генерації набору чисел за заданим алгоритмом. Синтаксис функції має вигляд:

```
replicate(n, expr, simplify = TRUE),
```

де n – число повторів, expr – функція або група виразів, які треба повторити n разів, simplify = TRUE – необов'язковий параметр, використання якого дозволяє (по можливості) спростити результат і представити його у вигляді вектора або матриці значень.

Функція vapply() подібна до sapply(), однак працює трохи швидше за рахунок того, що користувач однозначно вказує тип значень, що повертаються (буква «v» у назві vapply() означає «velocity» – швидкість). Такий підхід дозволяє також уникати повідомлень про помилки (і переривання обчислень), що виникають при роботі з sapply() в деяких ситуаціях. Під час виклику vapply() користувач повинен навести приклад очікуваного типу значень, що повертаються. Для цього служить аргумент FUN.VALUE:

```
# Аргументу FUN.VALUE присвоєно логічне значення FALSE.
```

```
# Цим задається тип значень, які повертаються функцією
a <- vapply(NULL, is.factor, FUN.VALUE = FALSE)
# Функція sapply() застосована до того ж об'єкту NULL:
b <- sapply(NULL, is.factor)

# Перевірка типу змінних:
is.logical(a)
[1] TRUE
is.logical(b)
[1] FALSE
```

Функція mapply() використовується у випадках, коли необхідно поелементно застосувати будь-яку функцію одночасно до декількох об'єктів (наприклад, отримати суму перших елементів векторів, потім суму других елементів векторів і т. д.). Результат повертається у вигляді вектора або масиву іншої розмірності. Буква «m» в назві mapply() означає «multivariate» – багатовимірний (мається на увазі одночасне виконання обчислень над елементами декількох об'єктів).

```
mapply(sum, 1:5, 1:5, 1:5)
[1] 3 6 9 12 15
```

Лекція 5. **ОСНОВИR. БАЗОВІ ГРАФІЧНІ МОЖЛИВОСТІ R**

План

5.1. Вступ

5.2. Функція plot() і її параметри

5.3. Загальні аргументи графічних функцій

5.4. Ширина і тип лінії

5.1. Вступ

Графічне представлення даних відіграє дуже важливу роль в статистиці та анілізі даних. Так, графіки є невід'ємною частиною розвідувального аналізу даних, дозволяють виявляти закономірності і тренди в складних наборах даних, а також безпосередньо можуть бути результатом статистичного аналізу.

Як правило, створення графіка починається з деякої функції високого рівня, яка визначає його загальну структуру: розмірність (1D, 2D, 3D), масштаби осей, назви та ін. Графічними функціями високого рівня, які найчастіше використовуються є plot(), hist(), boxplot(), scatterplot() (з пакета lattice) і pairs(). За допомогою багатого набору функцій низького рівня до збудованого графіком можуть бути додані додаткові елементи: текст, лінії, легенда та інше (прикладами таких функцій є lines(), points(), text(), axis() та ін.).

Налаштування окремих деталей діаграм виконується за допомогою параметрів, які, як правило, однакові для більшості високорівневих і низькорівневих функцій. Такі параметри, зокрема, визначають колір ліній і символів, тип і розмір символів, товщину і характер ліній, штрихування, рамку графіка та інше.

5.2. Функція plot() і її параметри

Функція plot() – головна функція, яка використовується для побудови графіків в R. Поведінка цієї базової функції високого рівня визначається

класом об'єктів, які до неї подаються.Відповідно, за допомогою plot() можна створити дуже великий набір різнотипних графіків[3].

Як приклад використаємо дані Kwan et al. (1976, http://bit.ly/1E7gqwl) по швидкості виведення з організму людини індометацина – одного з найбільш активних протизапальних препаратів. В експерименті взяли участь шість піддослідних. Результати цього дослідження входять в базовий набір даних R і доступні за командою

data(Indometh)

Застосувавши команду

names(Indometh)

[1] "Subject" "time" "conc"

ми побачимо, що до складу таблиці Indometh входять змінні Subject (випробуваний), time(час з моменту введення препарату) і conc(концентрація препарату в крові). Для полегшення подальшої роботи закріпимо таблицю Indometh до пошукового шляху R:

attach(Indometh)

Завдяки цій команді тепер ми можемо безпосередньо звертатись до змінних таблиці Indometh (тобто використовувати її ім’я безпосередньо – наприклад, time замість Indometh$time).

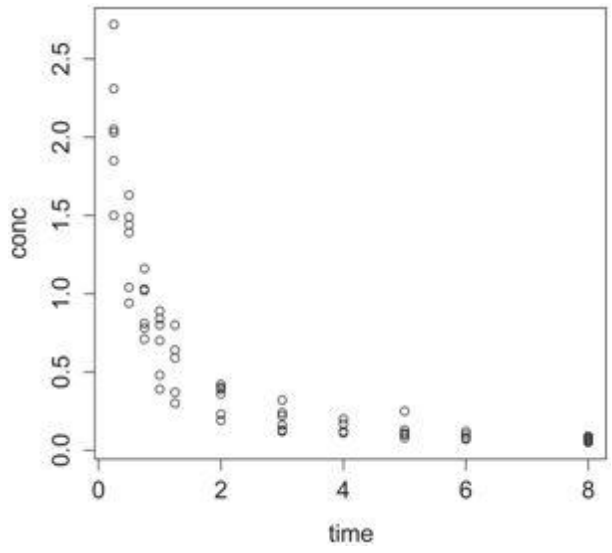


Рис. 5.1 Залежність концентрації індометацину в крові за часом

Залежність концентрації індометацину в крові від часу можна легко відобразити за допомогою простої команди plot(time, conc) (рис. 5.1)

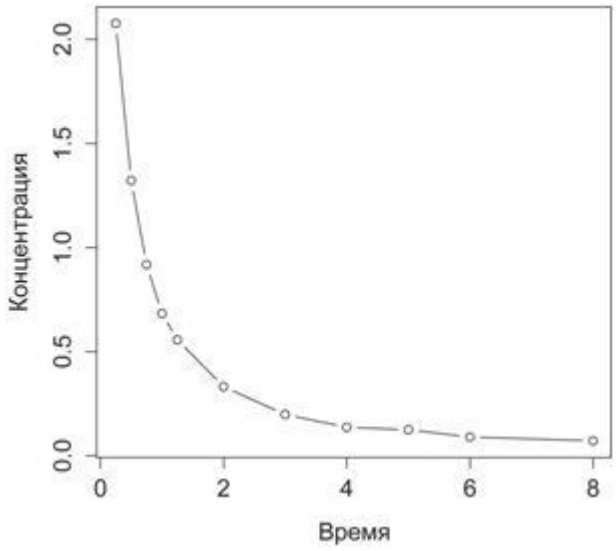


Рис. 5.2 Середні значення концентрації індометацину в крові за часом

Припустимо, що перед нами стоїть завдання відобразити на графіку не всі вихідні дані, а тільки середні значення концентрації індометацину для кожної тимчасової точки (рис. 5.2). Обчислити середні значення (або будь-які інші кількісні величини) для окремих груп даних дозволяє функція tapply():

```
(means <- tapply(conc, time, mean))
```

0.25	0.5	0.75	1	1.25
2.0766	1.3216	0.9183	0.6833	0.5566
6667	6667	3333	3333	6667
3	4	5	6	8
0.1983	0.1366	0.1250	0.0900	0.0716
3333	6667	0000	0000	6667

Зверніть увагу на те, що при створенні вектора means функція tapply() автоматично присвоєла кожному з середніх значень ім'я, відповідне часу обліку концентрації індометацину. Це легко перевірити:

```
names(means)
```

```
[1] "0.25" "0.5" "0.75" "1" "1.25" "2" "3" "4" "5" "6" "8"
```

Ми можемо скористатися цією обставиною при побудові графіка і легко створити числовий вектор зі значеннями часу обліку концентрації препарату:

```
indo.times <- as.numeric(names(means))
# Будуємо графік типу "точки з лініями":
plot(indo.times, means, type = "b", xlab = "Час", ylab = "Концентрація")
```

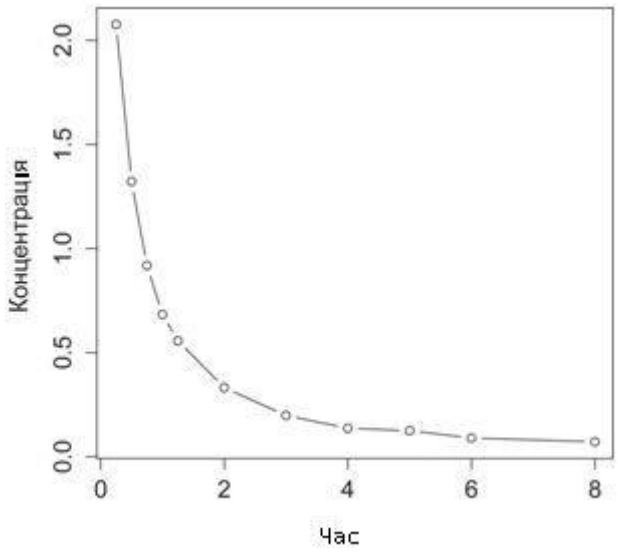


Рис. 5.3 Значення концентрації препарату за часом

Функція plot() має велику кількість параметрів, які керують її поведінкою.Ці параметри дозволяють здійснити налаштування зовнішнього вигляду графіка. Нижче розглянутотільки деякі з них.

Параметри xlab і ylab служать для позначення назв осей X та Y відповідно.

Параметр type дозволяє змінювати зовнішній вигляд точок на графіку. Він приймає одне з таких значень:

- "p" – точки («points»; використовується за умовчанням);
- "l" – лінії («lines»);
- "b" – зображуються і точки, і лінії («both points and lines»);
- "o" – точки зображуються поверх ліній («points over lines»);
- "h" – гістограма («histogram»);
- "s" – ступінчаста крива («steps»);
- "n" – дані не відображаються («no points»).

Параметри xlim і ylimконтролюють діапазон значень на кожній з осей графіка. За замовчуваннямобидва вони приймають значення NULL – в цьому випадку розмах вибирається програмою автоматично. Для скасування

автоматичних налаштувань відповідного параметру необхідно присвоїти значення у вигляді числового вектора, що містить мінімальне і максимальне значення, які повинні відображатися на осі. наприклад:

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", xlim = c(0, 15))

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", ylim = c(0, 5))

Параметри axes і ann контролюють відображення осей і їх назв відповідно. Кожен з них може приймати одне з двох можливих значень – TRUE або FALSE:

plot(indo.times, means, xlab = "Час", ylab = "Концентрація", axes = TRUE, ann = TRUE)

plot(indo.times, means, xlab = "Час", ylab = "Концентрація", axes = FALSE, ann = TRUE)

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", axes = TRUE, ann = FALSE)

За допомогою аргументу log можна перевести одну або обидві осі графіка на логарифмічну шкалу, наприклад:

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", log = "x")

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", log = "y")

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", log = "xy")

Аргумент main використовують для створення заголовка графіка. За замовчуванням назва розміщується у верхній частині малюнка:

plot(indo.times, means, xlab = "Час", ylab = "Концентрація",main = "Швидкість виведення індометацину", type = "o")

5.3. Загальні аргументи графічних функцій

Як видно з рис. 5.3, окремі вимірювання за замовчуванням зображуються у вигляді кругів. Інші символи можна задати за допомогою аргументу pch (від «plotting character» – символ зображення). У стандартних випадках цей аргумент приймає чисельні значення від 1 до 25. Наприклад, при pch = 2 символи перетворяться з кругів в незафарбовані трикутники:

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", main = "Швидкість виведення індометацину", type = "o", pch = 2)

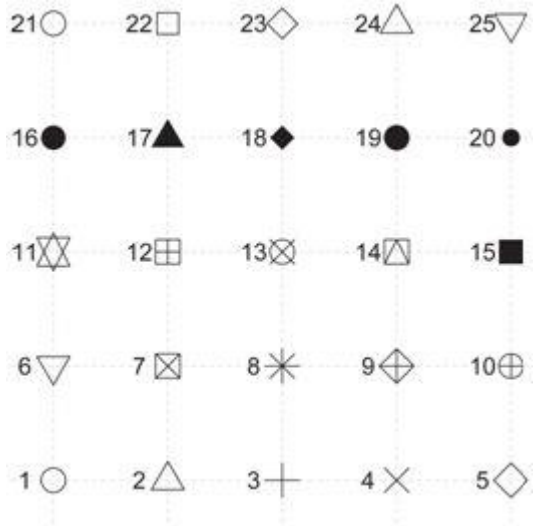


Рис. 5.4 Стандартні символи зображення і відповідні їм чисельні коди

Стандартні символи і відповідні їм чисельні коди представлені на рис. 5.4.

Набір символів, які використовуються для відображення даних, може бути значно розширений у випадку, коли аргумент pch використовується в комбінації з аргументом font, що задає шрифт. Параметру pch може при цьому присвоюватись будь-яке ціле число від 1 до 128 і від 160 до 254. Наприклад, при font = 5 маркеру у вигляді «сердечка» відповідає код 169:

plot(indo.times, means, xlab = "Час", ylab = "Концентрація", main = "Швидкість виведення індометацину", type = "o", pch = 169, font = 5)

В якості маркерів можна також використовувати звичайні друковані символи, наприклад літери:

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", main = "Швидкість виведення індометацину", type = "o", pch = "A")

Розмір символів можна задати за допомогою аргументу cex (від «character extension» – розмір символу), який за замовчуванням дорівнює 1. Зменшення або збільшення цього параметра призводить до відповідних пропорційних змін.

За необхідністю ми можемо також змінити ширину лінії символу. Для цього потрібно використати параметр lwd (від «line width» – ширина лінії).

Колір будь-якого графічного об'єкта може бути заданий декількома способами:

- за назвою кольору: наприклад, col = "red" (червоний), col = "green" (зелений) або col = "black" (чорний). Всього в R є 675 стандартних кольорів. Їх назви доступні за командою colors();
- шляхом безпосередньої вказівки червоного, зеленого і синього компонентів RGB-спектра з використанням шістнадцяткової системи численняу форматі "#RRGGBB";
- за чисельним кодом, наприклад: col = 2 (червоний), col = 3 (зелений) або col = 1 (чорний).

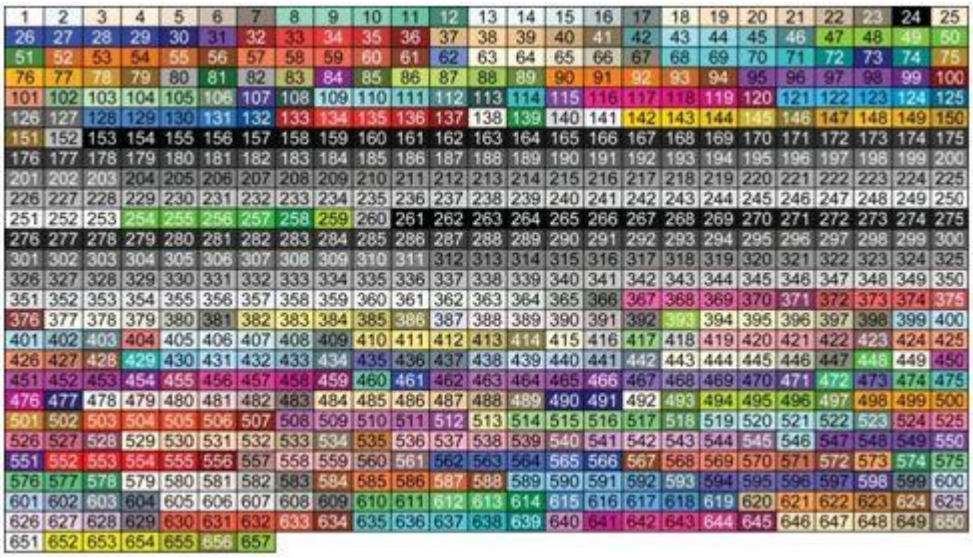


Рис. 5.5 Діаграма кольорів

Колір маркерів задається аргументом col (від «color» – колір). Номер необхідного кольору можна підібрати за допомогою діаграми, яка представлена на рис. 5.5.

Є також окремі аргументи для налаштування кольору інших елементів графіка, таких як заголовок (col.main), назви осей (col.lab), мітки осей (col.axes) та ін.

5.4. Ширина і тип лінії

Ширина лінії задається за допомогою аргументу lwd функції plot(). Цей аргумент приймає додатні числові значення, що показують, у скільки разів

ширина лінії повинна бути більшоюпо відношенню до ширини, заданої за замовчуванням. Ширина лінії (за замовчуванням дорівнює 1) є безрозмірною величиною, оскільки на різних графічних пристроях лінії з однаковими параметрами можуть виглядати по-різному. Нижче приведені приклади трьох графіків з різними значеннями параметра lwd:

plot (indo.times, means, xlab = "Час", ylab = "Концентрація",main = "lwd = 2", type = "l", lwd = 2)

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", main = "lwd = 5", type = "l", lwd = 5)

plot (indo.times, means, xlab = "Час", ylab = "Концентрація", main = "lwd = 10", type = "l", lwd = 10)

Аргумент lend (від «line end») функції plot() дозволяє налаштувати зовнішній вигляд кінців лінії. Цей аргумент приймає значення 0 (за замовчуванням), 1 або 2, що відповідає округленим, усіченим квадратним і квадратним кінцям відповідно. Місця з'єднання ліній також можуть виглядати по-різному, що визначається аргументом ljoin (від «line» – лінія і «join» – місце з'єднання). Аргумент ljoin приймає значення 0 (за замовчуванням), 1 або 2, що відповідає округленому, гострокутному і відсіченому з'єднанням відповідно.

Тип лінії налаштовується за допомогою аргументу lty (від «line» – лінія і «type» – тип) функції plot(). Існує шість попередньо встановлених типів ліній, які задаються числами від 1 до 6 відповідно.

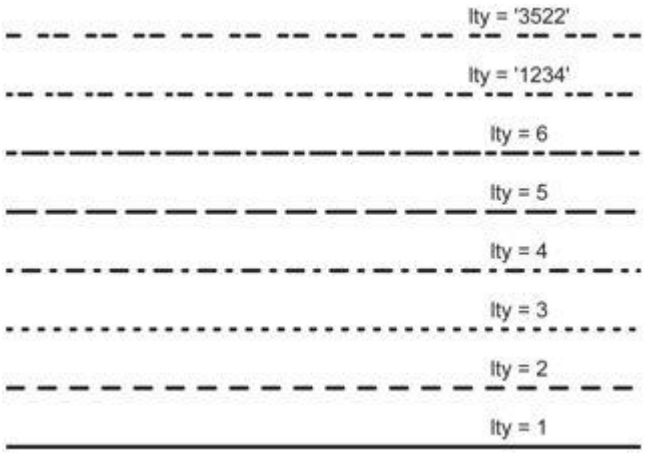


Рис. 5.6 Типи ліній

За необхідністю можна також створити призначені для користувача типи ліній. У таких випадках в якості значення аргументу lty виступає текстова послідовність з чотирьох цифр. Ці числа (від 1 до 9) визначають розмір чотирьох елементів, якіутворюють паттерн, що повторюється «штрих – пробіл – штрих – пробіл». Наприклад, при lty = "4241" лінія буде складатися з повторюваного паттерна, в якому є штрих довжиною 4 одиниці, пробіл довжиною 2 одиниці, знову штрих довжиною 4 одиниці і пробіл в 1 одиницю. Приклади стандартних і користувацьких типів ліній наведені на рис. 5.6.

Колір ліній задається за допомогою аргумента col. Використання аргумента col щодо ліній нічим не відрізняється від його використання по відношенню до графічних символів.

Для налаштування зовнішнього вигляду рамки графіка використовують аргумент bty (від «box» – коробка і «type» – тип) функції plot(). Цей аргумент приймає одне з таких шести текстових значень: "O", "L", "7", "C", "U", "[". Рамка прийматиме вигляд відповідно до форми зазначеного символу (допускається використання також малих літер o, l, c, u).

Лекція 6.

ОСНОВИ R. ГІСТОГРАМИ

План

6.1. Вступ

6.2. Приклади використання гістограм

6.3. Діаграми розмахів

6.4. Кругові і стовпчикові діаграми

6.5. Діаграми Клівленда

6.1. Вступ

Гістограма є важливим інструментом статистики, що дозволяє наочно уявити емпіричний розподіл значень аналізованої змінної. В системі R для побудови гістограм використовується функція hist(). Її основним аргументом є ім'я аналізованої змінної.

6.2. Приклади використання гістограм

Як приклад створимо нормально розподілену сукупність X з 100 спостережень із середнім значенням 15 і стандартним відхиленням 5:

```
X <- rnorm(n = 100, mean = 15, sd = 5)
```

Для створення змінної X використана функція rnorm() (від «random» – випадковий і «norm» – нормальний). Використовуючи генератор (псевдо)випадкових чисел, ця функція формує нормально розподілені сукупності з заданим розміром n, середнім значенням mean і стандартним відхиленням sd. Зобразити значення змінної X у вигляді гістограми можна наступним чином (рис. 6.1)

```
hist(X)
```

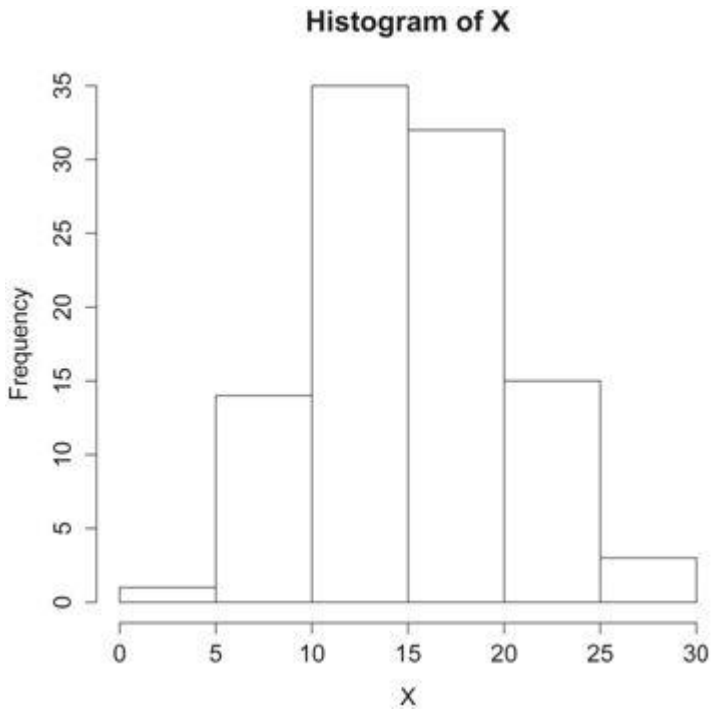


Рис. 6.1 Гістограма змінної X

Як видно з рис. 6.1, функція hist() автоматично підбирає кількість стовпців для відображення на графіку, а також створює назви осей і заголовок графіка. Такого малюнка, одержуваного з використанням автоматичного налаштування, може виявитися цілком достатньо (наприклад, при проведенні швидкого розвідувального аналізу даних). Однак часто потрібно його додаткове доопрацювання.

Перш за все важливо звернути увагу на розмір кроку, який використовується для розбиття даних на класи при побудові гістограми. У наведеному вище прикладі (рис. 6.1) програма розбила значення змінної X на 6 класів. Однак таке грубе розбиття може недостатньо точно відображати властивості аналізованої сукупності. Для більш детального вивчення цих властивостей можна збільшити величину ділення даних на класи (тобто використовувати менший класовий проміжок). Зробити це дозволяє аргумент breaks (розломи) функції hist(). При необхідності стовпці гістограми можна залити бажаним кольором. Для цього потрібно скористатися аргументом col – це аналогічний аргумент, який використовується при налаштуванні функції plot(). У наведеному нижче прикладі вибрано світло-блакитний колір стовпців (lightblue):

hist(X, breaks = 20, col = "lightblue")

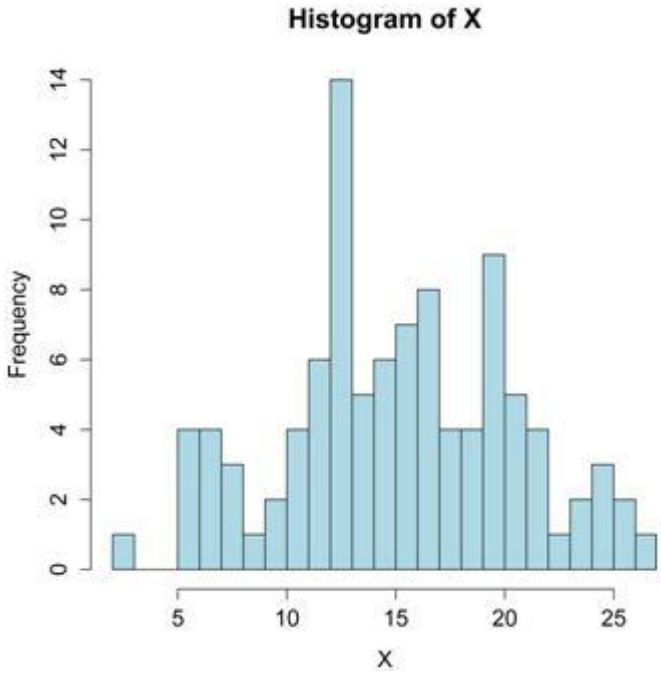


Рис. 6.2 Гістограма змінної X з додатковими налаштуваннями

Як бачимо, результатом виконання попередньої команди стала гістограма з двадцятьма стовпцями (рис. 6.2), що дозволяє більш детально проаналізувати розподіл значень змінної X.

За замовчуванням функція hist() відображає по осі ординат частоти появи кожного класу значень X. Таку поведінку функції можна змінити, надавши аргументу freq (від «frequency» – частота) значення FALSE. У цьому випадку вісь ординат буде відображати щільність ймовірності кожного класу так, що сумарна площа під гістограмою складе 1:

hist(X, breaks = 20, freq = FALSE)

У деяких випадках, зокрема при невеликій кількості спостережень, гістограми можуть давати неправильне уявлення про властивості сукупності, наприклад через невеликукількість рідко розташованих стовпців:

X <- rnorm(n = 50, mean = 15, sd = 5)

hist(X, breaks = 20, freq = FALSE, col = "lightblue")

Замість гістограми (або паралельно з нею) в таких випадках рекомендується скористатись кривою щільності ймовірності. Оцінка щільності ймовірності виконується за допомогою функції density(), яку можна

застосувати в якості аргументу функції plot() для графічного зображення результату:

plot(density(X))

Гладкість отриманої кривої регулюється за допомогою аргументу bw (від «Bandwidth» – ширина вікна), наприклад:

plot(density(X, bw = 0.8))

6.3. Діаграми розмахів

Діаграми розмахів, або «ящики з вусами» (англ. «box plots», «box-whisker plots»), отримали свою назву за характерний вид: точку або лінію, яка відповідає деякій мірі центральної тенденції в даних, оточує прямокутник («ящик»), довжина якого відповідає певному показнику розкиду. Додатково від цього прямокутника відходять «вуса», які також відображають розкид в даних або, в деяких випадках, точність оцінки величини центральної тенденції. Графіки цього типу дуже популярні, оскільки дозволяють дати максимально повну статистичну характеристику аналізованої сукупності. Крім того, діаграми розмахів можна використовувати для візуальної експрес-оцінки різниці між двома і більше групами (наприклад, між датами відбору проб, експериментальними групами, ділянками простору і т. д.).

В R для побудови діаграм розмахіввикористовують функцію boxplot(). На відміну від багатьох інших статистичних програм, в R при побудові діаграм розмахів використовуються стійкі (робастні) оцінки центральної тенденції (медіани) і розкиду (інтерквартільниого розмаху–ІКР).

Особливості використання функції boxplot() розглянемо на прикладі даних InsectSprays, що описують експеримент з вивчення ефективності шести видів інсектицидних засобів. Для побудови діаграми размахів за цими даними досить виконати команду

boxplot(count ~ spray, data = InsectSprays)

Цікаво, що побудувати діаграми розмаху можна не тільки за допомогою спеціалізованої функції boxplot(), але також і функції plot().Наприклад, по

команді plot (count ~ spray, data = InsectSprays). Справа в тому, що функція plot()автоматично розпізнає, що змінна count є кількісною, а spray – номінальною, і тому починає поводитися як boxplot().

6.4. Кругові і стовпчикові діаграми

Кругові діаграми (англ. «Pie charts»), м'яко кажучи, не в пошані у професійних статистів. Інформація, яка надається за допомогою кругової діаграми, погано сприймається візуально, і практично завжди найкращою альтернативою цьому способу візуалізації даних буде розглянута нижче точкова діаграма Клівленда або, в крайньому випадку, стовпчикова діаграма. Тому не дивно, що в перших версіях R навіть не було окремої функції для побудови кругових діаграм. Пізніше така функція з'явилася, оскільки в деяких випадках цей вид діаграм все ж може виявитися корисним. Відповідна функція називається pie().

Функція pie() має декілька аргументів (докладніше див. ?Pie). Основними з них є наступні:

- x - вектор додатних чисел, на основі яких будується діаграма;
- labels – текстовий вектор, що містить підписи секторів діаграми; якщо значення x вже мають атрибут names (імена), то аргумент labels вказувати не обов'язково;
- radius – змінює розмір квадрата, всередині якого будується діаграма; у випадках, коли підписи секторів діаграми занадто довгі, розмір цього квадрата можна зменшити (можливі варіанти: від -1 до 1);
- init.angle – кут повороту діаграми;
- col – числовий або текстовий вектор, що містить коди кольорів для залвки секторів діаграми;
- main – текстовий вектор, що містить заголовок діаграми;
- ... – інші графічні параметри (наприклад, параметри, що визначають розмір підписів секторів діаграми, колір ліній і т. д.).

Для створення стовпчикових діаграм (рідше «лінійчатих»; англ. «Bar plots» або «bar charts») в системі R служить функція barplot(). У цій функції є велика кількість аргументів (докладніше див. ?barplot), з яких до основних відносяться:

- hight (висота) – числовий вектор або матриця зі значеннями, які використовуються для побудови діаграми. Якщо аргумент hight вказано у вигляді вектора, то будується графік з послідовно розташованих стовпців, висоти яких відповідають значенням з цього вектора. Якщо hight вказано у вигляді матриці і аргумент beside = FALSE, то буде побудована стовпчикова діаграма з накопиченням. Якщо ж hight вказано у вигляді матриці і аргумент beside = TRUE, то стовпці діаграми будуть згруповані відповідно до стовпців матриці;
- width (ширина) – необов'язковий параметр, що дозволяє регулювати ширину стовпців на діаграмі. Вказується у вигляді числового вектора, значення якого відповідають ширині стовпців;
- space (простір) – величина зазору між стовпцями (пропорціно їх середній ширині). Може бути вказана або у вигляді одного числа, або у вигляді вектора з чисел, які відповідають кожному стовпцю діаграми;
- names.arg – текстовий вектор, що містить підписи (уздовж осі X) для кожного стовпця або групи стовпців. Якщо цей аргумент не вказано, то в якості підписів автоматично будуть використані імена елементів вектора height (якщо такі є) або заголовки стовпців, якщо height представляет собою матрицю;
- legend.text – вектор, що містить текстові елементи легенди графіка. Цей аргумент корисний, тільки якщо height є матрицею. В цьому випадку мітки легенди повинні відповідати рядкам матриці. Аргументу legend.text можна також привласнити значення TRUE, і тоді імена рядків матриці (якщо такі є) будуть використані в якості міток легенди автоматично;
- horiz – приймає логічне значення: TRUE для горизонтального розташування стовпців і FALSE - для вертикального;

•density – числовий вектор, що задає щільність штрихування стовпців;

•angle – кут нахилу штрихів (в градусах);

•col – вектор колірних кодів для стовпців або їх елементів. За замовчуванням стовпці зафарбовуються сірим кольором, якщо height – вектор, і різними градаціями сірого, якщо height – матриця;

•border – код кольору для обведення стовпців. Якщо межу стовпців обводити не потрібно, то можна вказати border = NA;

•... – інші графічні параметри (наприклад ?Plot і ?Par).

6.5. Діаграми Клівленда

Точковими діаграмами Клівленда є графіки, на яких точки використовуються для відображення значень деякої кількісної змінної (або змінних), розбитої(их) на групи відповідно до рівнів деякою номінальною змінною (або змінних). Цей інструмент графічного аналізу даних отримав свою назву на честь Вільяма Клівленда, який запропонував його (William Cleveland). В роботі Cleveland & McGill (1984, http://bit.ly/1Kt7oh2) було показано, що стовпчикові діаграми, які використовуються для зображення згрупованих значень кількісних змінних, візуально погано сприймаються людьми. Як альтернатива були запропоновані точкові діаграми.

Для пояснення цієї ідеї використаємо стовпчикову діаграму (рис. 6.3), на якій зображено розподіл 32 моделей автомобілів 1973-1974 років випуску за економічностю двигуна (вимірюється в кількостях миль, які автомобіль проїзджає на одному галоні палива). Дані, використані для побудови діаграми, були опубліковані в американском Журналі Motor Trend в 1974 р і входять в стандартний набір даних R та доступні за допомогою команди data(mtcars).

Завантажимо відповідну таблицю з даними mtcars в середовище R і дослідимо її вміст:

```
data(mtcars)
mtcars
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2

У таблиці є дані за 11 параметрами, які характеризують кожну модель. Детальніше про те, які параметри були враховані, можна ознайомитися у файлі допомоги, доступному за командою ?Mtcars.

Побудуємо стовпчикову діаграму так, щоб стовпчики на ній були залиті різними кольорами відповідно до кількості циліндрів cyl в двигунах автомобілів (рис. 6.3):

```
par(mar = c(6, 4, 1, 1))
x <- mtcars[order(mtcars$mpg),
x$color[x$cyl==4] <- 1
x$color[x$cyl==6] <- 2
x$color[x$cyl==8] <- 3
with(x, barplot(mpg, names.arg = rownames(x), las = 2, cex.axis = 0.6, cex.lab = 0.8, cex = 0.6, ylab = "Миль/галлон", col = color))
legend(1, 30, legend = c(8, 6, 4), fill = c(3, 2, 1), cex = 0.6)
```

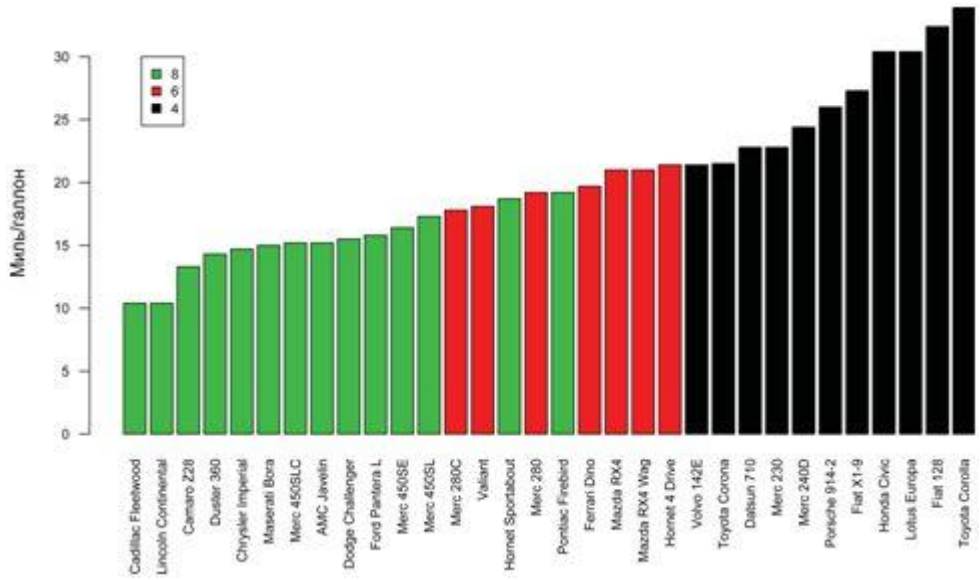


Рис. 6.3 Розподіл 32 моделей автомобілів за економічностю двигуна

На графіку добре видно, що автомобілі з меншою кількістю циліндрів здатні проїхати більшу відстань на одному галоні палива. Крім того, можна простежити розподіл моделей по економічності в межах кожної з груп, виділених за кількістю циліндрів.

Проблема зі стовпчиковими діаграми полягає в тому, що вони значною мірою надлишкові, оскільки площі, обмежені стовпчиками, не несуть ніякого змістовного навантаження (тобто, вибравши більш вузькі або ширші стовпчики, ми б все одно прийшли до тих же висновків про характер розподілу автомобілів). Тому більш відповідним типом статистичних графіків для такої ситуації була б точкова діаграма. В системі R побудова точкових діаграм здійснюється за допомогою функції dotchart(). Розглянемо її використання на тому ж прикладі з автомобілями.

Для нас зараз цікавий стовпець mgp (від «miles per gallon» – миль на галон), в якому містяться дані по пробігу кожної моделі в розрахунку на галон палива. Точкову діаграму за цими даними можна швидко побудувати наступним чином:

dotchart(mtcars$mpg, labels = row.names(mtcars), main = "Економіяпалива у 32 моделей автомобілів", xlab = "Миль/галлон", cex = 0.8)

У приведеному коді вказані такі параметри функції dotchart(): а) змінна, для якої будується графік (mtcars$mpg); б) текстовий вектор, що містить назви

моделей автомобілів (в даному випадку вони є назвами рядків таблиці – row.names(mtcars)); в) заголовок графіка (аргумент main) і назва осі X (аргумент xlab), і г) розмір точок на графіку і одночасно розмір шрифту для назв моделей (cex = 0.8).

Такий графік поки ще досить сирий, хоча і дозволяє досліджувати розкид наявних значень mpg у різних моделях. Картина стане набагато зрозумілішою, якщо ми відсортуємо дані за зростанням пробігу, згрупуємо дані відносно кількості циліндрів в двигуні і розфарбуємо відповідні групи різними кольорами.

Спочатку відсортуємо вихідну таблицю по зростанню mpg з використанням функції order() і збережемо результат у вигляді нової таблиці даних з ім'ям x:

```
x <- mtcars[order(mtcars$mpg), ]
```

Перетворимо кількісну змінну cyl (від «cylinder» – цілінр) в новій таблиці x в фактор – це потрібно зробити, оскільки ми збираємося згрупувати значення mpg саме за кількістю циліндрів:

```
x$cyl <- factor(x$cyl)
```

Створимо новий стовпець color в таблиці x, який буде містити числові коди кольорів для кожної з трьох груп автомобілів:

```
x$color[x$cyl==4] <- 1
x$color[x$cyl==6] <- 2
x$color[x$cyl==8] <- 3
```

Тепер у нас є все необхідне для побудови бажаного графіка (рис. 6.4):

```
dotchart(x$mpg, labels = row.names(x),
  groups = x$cyl, gcolor = "blue", pch = 16,
  main = "Економність двигуна у 32 моделей автомобілів",
  xlab = "Миль/галлон", cex = 0.8, color = x$color)
```

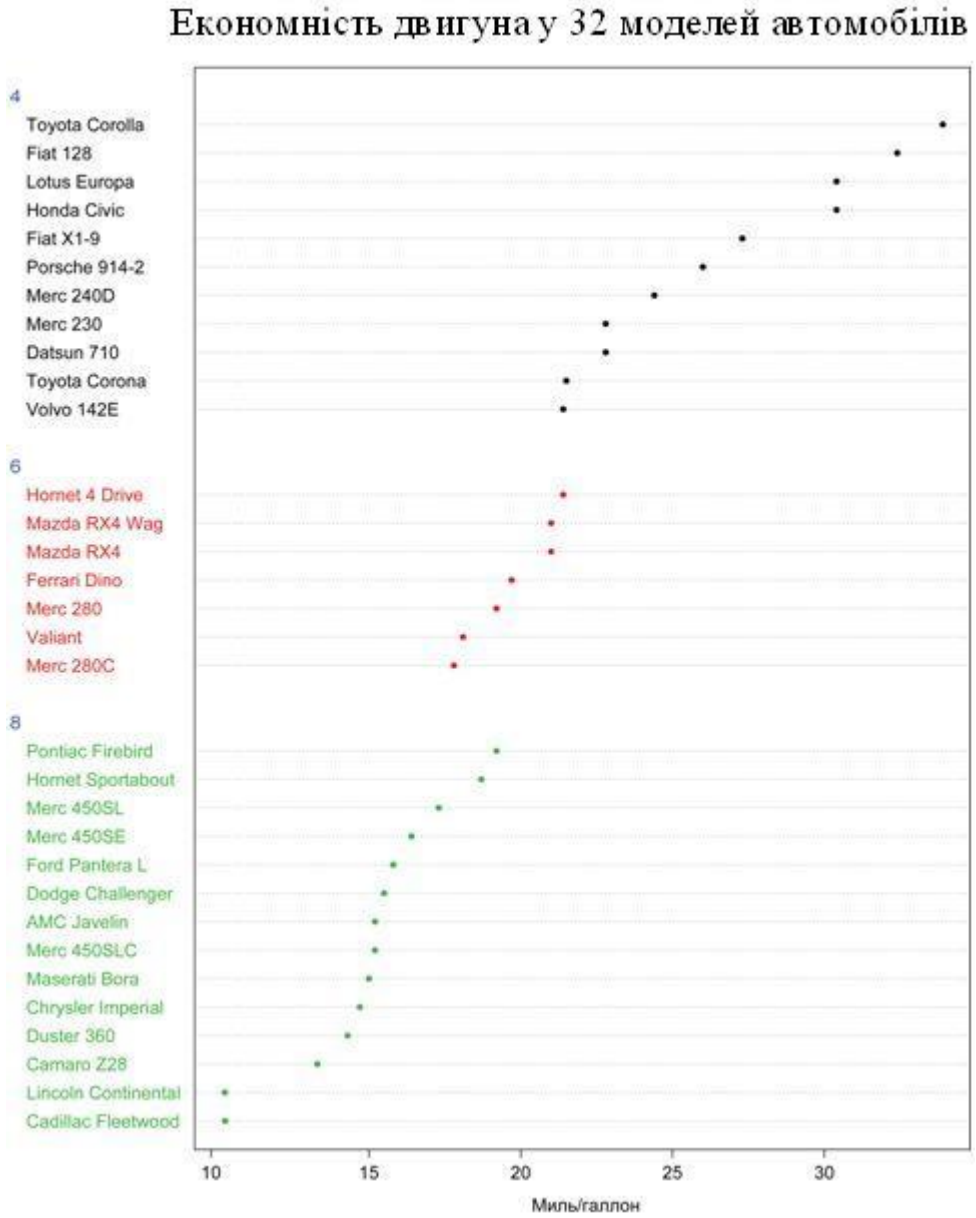


Рис. 6.4 Економність двигуна у 32 моделей автомобілів

У наведеному коді аргумент groups використовується для вказівки групуючій змінній (в нашому випадку – перетворена в фактор змінна cyl). За допомогою аргументу gcolor задається колір назв груп (тут - блакитний). Аргумент color служить для задавання кольорів, специфічних для кожної групи (в нашому прикладі – 1 (чорний) для машин з чотирма циліндрами, 2 (червоний) для машин з шістьма циліндрами і 3 (зелений) для машин з вісьмома двигунами. І всі ці числові коди кольорів зберігаються в стовпці color таблиці x).

Завдяки сортуванню і угрупованню даних, а також використання точок різного кольору замість стовпчиків отримана точкова діаграма сприймається

набагато легше, ніж наведена вище «важка» стовпчикова діаграма.

Лекція 7.

ОПИСОВА СТАТИСТИКА. ОЧИЩЕННЯ ДАНИХ З ДОПОМОГОЮ R

План

7.1.Вступ

7.2. Квартилі та інтерквартильний розмах

7.3. Коробчата діаграма

7.4. Дисперсія та середньоквадратичне відхилення

7.1. Вступ

Описова статистика або дескриптивна статистика (англ. descriptive statistics) – розділ статистики, який займається обробкою емпіричних даних, їх систематизацією, наочним представленням у вигляді графіків та таблиць, а також їх кількісним описом через основні статистичні показники.

Основні статистичні показники, які використовуються для опису набору дани – це міри центральної тенденції та міри мінливості. До мір центральної тенденції включають середнє значення, медіану, моду, а до мір мінливості – стандартне відхилення (чи дисперсію), мінімальне та максимальне значення змінної, розмах, ексцес та коефіцієнт асиметрії.

7.2. Квартилі та інтерквартильний розмах

Якщо медіана ділить дані порівну, то квартилі ділять їх на чотири частини. Вони позначаються Q1, Q2, Q3, Q4:

- Q1 – 25%;
- Q2 – 50% (співпадає з медіаною);
- Q3 – 75%;
- Q4 – 100%.

Інтерквартильний розмах(IQR) = Q3 - Q1

Наприклад, маємо ряд 62, 81, 63, 77, 64, 81, 64, 70, 72, 76.

Спочатку відсортуємо дані в зростаючому порядку: 62, 63, 64, 64, 70, 72,

76, 77, 81, 81.

Медіана(та Q2)): $\frac{70+71}{2}=71.$

Для обрахунку Q1 та Q3 значення медіани включаються до інтервалу.

Q1: $\frac{64+64}{2}=64$

Q3: $\frac{76+77}{2}=76,5$

Інтерквартильний розмах(ІКР): Q3 - Q1 = 76,5 - 64 = 12,5.

Середнє, мода та медіана є описовими статистиками. Також виділяють **узагальнення п'яти чисел (five numbers summary)**. Воно включає в себе:

- Найменше значення;
- Перший квартиль;
- Медіана (другий квартиль);
- Третій квартиль;
- Найбільше значення.

Узагальнення п'яти чисел для ряду можна отримати використовуючи R функцію summmary:

```
summary(c(62, 81, 63, 77, 64, 81, 64, 70, 72, 76))
##    Min. 1st Qu. Median Mean 3rd Qu.     Max.
## 62.00 64.00 71.0071.00  76.75    81.00
```

Наприклад, застосуємо цю статистику для оцінки довжини анаконд (завантажити набір даних можна за посиланням [4]):

```
anaconda <- read.table("anaconda.dat")
summary(anaconda$V1)
##     Min.   1st Qu. Median Mean 3rd Qu.     Max.
## 172.0229.4 258.8288.5333.0477.0
quantile(anaconda$V1)
##       0%      25%      50%     75%     100%
## 172.000 229.350  258.750 332.975 477.000
```

7.3. Коробчата діаграма

Коробчата діаграма дозволяє візуалізувати узагальнення п'яти чисел та знайти нетипові дані(так звані "викиди").

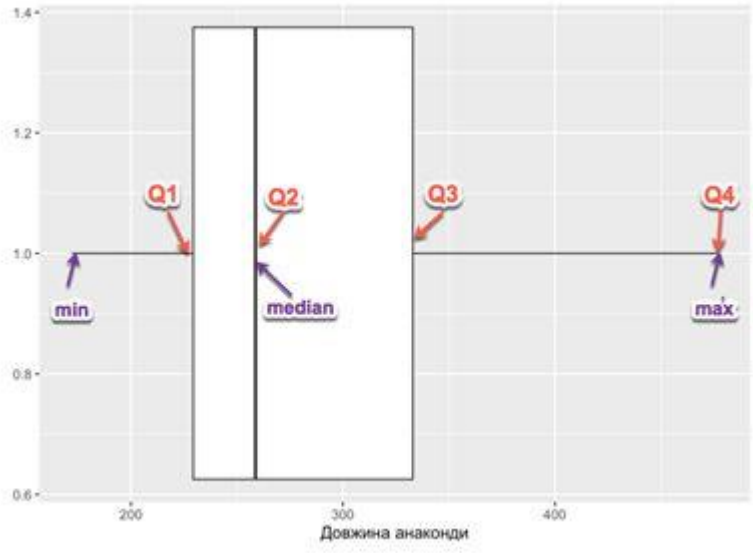
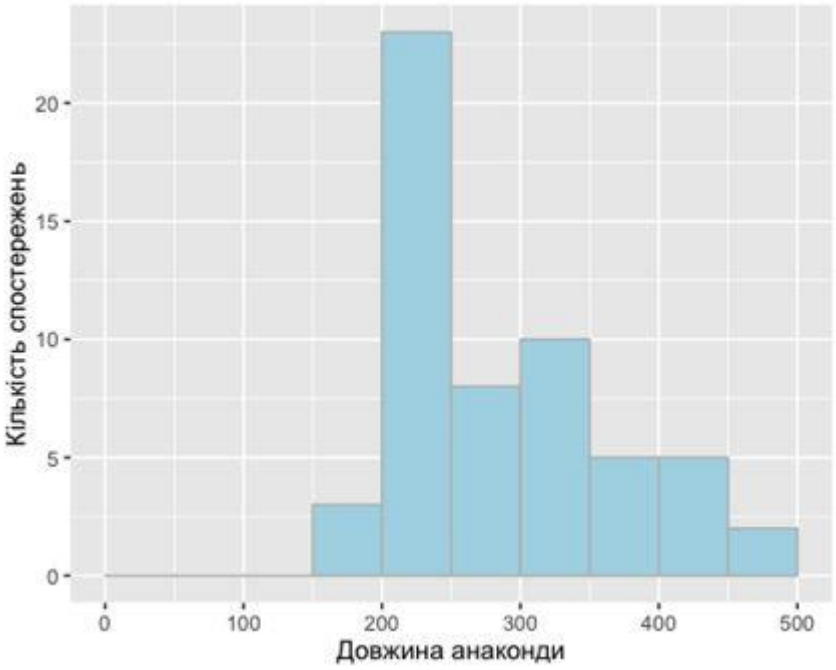


Рис. 7.1 Коробчата діаграма

"Викид" (outlier) – це значення, яке знаходиться на відстані, меньшій ніж 1.5 ІКР від Q1 або більшій від Q3 (рис. 7.1).

Якщо ми хочемо детальніше оцінити розподіл довжини анаконд, можемо використати гістограму:

```
ggplot(anaconda, aes(x=V1)) +
geom_histogram(breaks=seq(0, 500, by = 50), fill="lightblue", col="grey") +
xlab("Довжина анаконди") + ylab("Кількість спостережень")
```



7.4. Дисперсія та середньоквадратичне відхилення

Середньоквадратичне відхилення(standard deviation) дає розуміння, наскільки далеко знаходиться типове спостереження від середнього значення.

Дисперсія(variance)– обчислюється як середнє значення відстаней від всіх спострежень до середнього значення у квадраті.

$$\text{var} = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}$$

Середньоквадратичне відхилення(standard deviation) у – обчислюється як корінь квадратний з дисперсії.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}}$$

Давайте обчислимо дисперсію та середньоквадратичне відхилення на прикладі.

Наприклад, у нас є ряд: 5, 3, 2, 8, 2.

Середнє значення μ = 4.

Обчислимо дисперсію:

$$\text{var} = \frac{(5-4)^2 + (3-4)^2 + (2-4)^2 + (8-4)^2 + (2-4)^2}{5} = 5,2$$

Середньоквадратичне відхилення:

$$\sigma = \sqrt{\text{var}} = 2,28$$

Дисперсія для ряду 3, 5, 5, 3, 4, середнє значення якого теж 4:

$$\text{var} = \frac{(3-4)^2 + (5-4)^2 + (5-4)^2 + (3-4)^2 + (4-4)^2}{5} = 0,8$$

Середньоквадратичне відхилення:

$$\sigma = \sqrt{\text{var}} = 0,89$$

Якщо поглянути на обидва ряди, бачимо що значення другого більш тісно розташовані навколо середнього. Значення дисперсії та середньоквадратичного дозволяють це виразити чисельно.

Приклад 7.1. Степан має 800 друзів у Facebook, Аня 1000 послідовників в Instagram. Хто більш популярний?

Для відповіді на це питання нам потрібні будуть дані про середнє значення та середньоквадратичне відхилення для кількості друзів у Facebook та послідовників у Instagram.

Facebook: середнє значення μ = 649, середньоквадратичне відхилення у = 50.

Instagram: середнє значення μ = 843, середньоквадратичне відхилення у = 60.

Обчислимо відстань до середнього значення в середньоквадратичних відхиленнях:

Степан: $\frac{x-\mu}{\sigma} = \frac{800-649}{50} = 3,02.$

Аня: $\frac{x-\mu}{\sigma} = \frac{1000-843}{60} = 2,61$

Можемо вважати, що Степан більш популярний, оскільки значення кількості його друзів знаходиться далі від середнього значення.

Процес перетворення даних з допомогою формули $\frac{x-\mu}{\sigma}$ має назву **z-стандартизація**, а отримані значення – **z-значення**.

Лекція 8.

ОСНОВИ ТЕОРІЇ ЙМОВІРНОСТЕЙ

План

8.1. Вступ

8.2. Ймовірність однієї події

8.3. Ймовірність кількох подій

8.4. Теорема Байеса

8.1. Вступ

Теорія імовірності – розділ математики, що вивчає закономірності випадкових явищ: випадкові події, випадкові величини, їхні функції, властивості й операції над ними.

Математичні моделі в теорії ймовірності описують з деяким ступенем точності випробування (експерименти, спостереження, вимірювання), результати яких неоднозначно визначаються умовами випробування.

Під випробуванням мається на увазі здійснення запланованих дій і отримання результату за виконання певного комплексу умов. При цьому припускається, що ці умови є фіксованими; вони або об'єктивно існують, або створюються штучно й можуть бути відтворені необмежену кількість разів.

8.2. Ймовірність однієї події

Розглянемо класичний приклад із підкиданням монетки. Коли ми підкидаємо монетку, ми не можемо сказати "орлом" чи "решкою" вона впаде. Підкидання монетки – випробування.

Випробування (експеримент) – це сукупність умов, за яких спостерігається певне явище чи результат. Результатом у прикладі з монеткою є факт, що монетка впала "орлом" або "решкою". Випадання "орла" чи "решки" – подія.

Подія – це факт, який в результаті експерименту може відбутись чи не

відбутись. Якщо ми будемо підкидати цю монетку велику кількість раз(наприклад тисячу) і щоразу записувати результат, то зможемо оцінити ймовірність настання кожної події.

Ймовірність – чисельна міра впевненості в появі даної події внаслідок нового випробування. Тобто, якщо нас цікавить, яка ймовірність випадання "решки" для даної монетки, то ми підкидаємо монетку n разів та обчислюємо ймовірність p(A) за формулою

$$P(A) = \frac{m}{n},$$

де A - подія "монета впала решкою";

p(A) - ймовірність цієї події;

m - кількість разів, коли настала подія A;

n - кількість випробувань.

Оскільки m та n - цілі числа і $0 \le m \le n$, то $0 \le P(A) \le 1$.

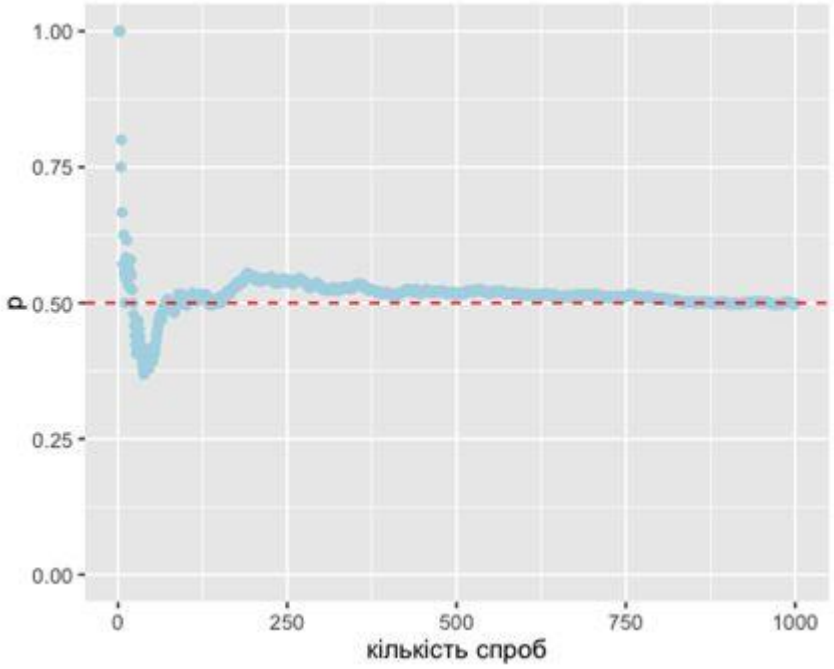


Рис. 8.1 Кількість спроб підкидання монети

Закон великих чисел стверджує, що якщо ми будемо повторювати експеримент нескінченну кількість разів(на практиці просто достатньо багато), то частка настання нашої події до кількості випадків буденаближатися до реальної ймовірності настання цієї події (рис. 8.1). На графіку ви бачите, як

змінюється ймовірність випадання орла p в залежності від кількості спроб.

Таке трактування ймовірності має назву "**частотна інтерпретація**" і передбачає, що випробування можна здійснити достатньо велику кількість разів. Існує інший підхід до визначення ймовірності, так звана "**байесівська інтерпретація**". Вона передбачає, що в нас є якась початкова ймовірність, наприклад корумпованості конкретного чиновника, і ми коригуємо цю ймовірність в залежності від фактів (для чиновника це можуть бути подані декларації, нерухоме майно).

Сума ймовірностей всіх можливих подій, які можуть настати в результаті випробування(для монети це випадання орлом або решкою) дорівнює одиниці.

8.3. Ймовірність кількох подій

Щоб оцінити ймовірність настання кількох подій, потрібно зрозуміти, як вони співвідносяться між собою.

Події можуть бути Несумісними (disjoint) або Сумісними (joint).

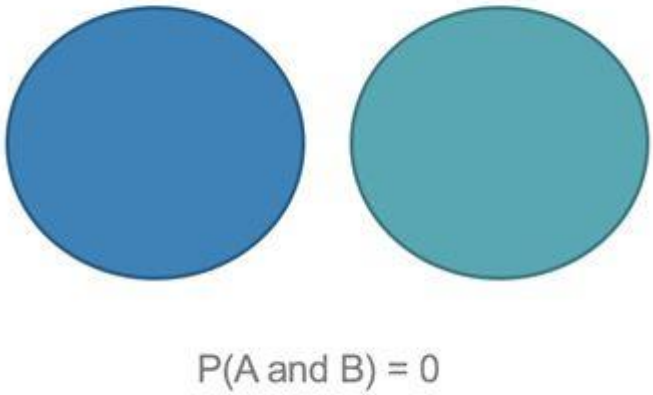


Рис. 8.2 Несумісні події

Несумісні(disjoint) – це події, які не можуть відбуватись одночасно (рис. 8.2).

Приклад несумісних подій:

- монета не може впасти і орлом і решкою;
- не можна здати і провалити іспит одночасно.

Для обчислення ймовірністі настання події A або події B, які є несумісними, можна використати наступну формулу:

P(A or B) = P(A) + P(B)

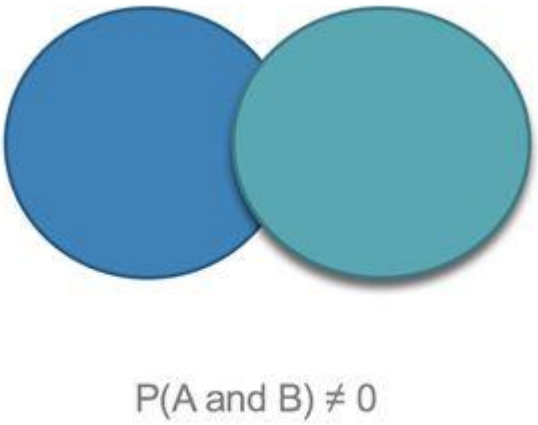


Рис. 8.3 Сумісні події

Сумісні(joint) – це події, які можуть відбуватись одночасно (рис. 8.3).

Студент може одночасно проходити курс статистики та англійської мови. Це приклад сумісних подій, де подія A – проходить курс статистики, подія B – проходить курс англійської мови.

Формула для визначення ймовірності настання події A або події B, якщо події є сумісними:

$$P(A \text{ or } B) = P(A) + P(B) – P(A \text{ and } B) \quad (8.1)$$

Приклад 8.1. Було опитано 77882 людини з 57 країн світу. 36.2% погоджуються з твердженням “Чоловіки повинні мати більше прав ніж жінки”. 13.8% мають університетську освіту. 3.6% належать до обох категорій. Яка ймовірність, що випадковим чином обрана людина має вищу освіту або погоджується з твердженням “Чоловіки повинні мати більше прав ніж жінки”?

Події "випадковим чином обрана людина має університетську освіту" та "випадковим чином обрана людина погоджується з твердженням" є сумісними. Тому для обчислення ймовірністі, що випадковим чином обрана людина з вищою освітою або погоджується з твердженням скористаємось формулою (8.1). Тобто P(A or B) = 36.2% + 13.8% - 3.6% = 46.4%.

Дві події є **незалежними(independent)**, якщо знання про настання однієї з

них не дає можливості оцінити ймовірність настання іншої. Для прикладу, знання того, що надворі йде дощ не дає нам додаткової інформації для оцінки ймовірності виграти в лотереї.

Для обрахування ймовірності настання одночасно **незалежних** подій A та B використовуємо формулу

$$P(A \text{ and } B) = P(A) \times P(B)$$

Буває ситуація, коли події залежать одна від одної. Наприклад, у дощовий день ймовірність викликати таксі зменшується. Такі ймовірності називають **умовними (conditional)**. Можемо записати це так:

P(викликали таксі | дощ) = 0,4;

P(викликали таксі |¬дощ) = 0,8.

Тут знак ¬ означає заперечення. Крім того, також є ймовірність дощу в конкретному місті. Така ймовірність називається **апріорною**. Наприклад, ймовірність дощу в цьому місті дорівнює 0,7. Тоді, відповідно, ¬ дощ = 0,3.

Для умовних ймовірностей ймовірність одночаного настання подій A та B обчислюється за формулою

$$P(A \text{ and } B) = P(A|B) \times P(B) \qquad (8.2)$$

Якщо ми хочемо обчислити ймовірність викликати таксі, то спочатку оцінимо ймовірність викликати таксі, коли йде дощ і ймовірність цієї події, коли дощу нема. Позначимо:

A - викликали таксі;

B – дощ.

Використавши формулу (8.2) отримаємо:

P(викликали таксі) =

P(викликали таксі і дощ) + P(викликали таксі і ¬дощ) = P(викликали таксі | дощ) x P(дощ) +

P(викликали таксі | ¬дощ) x P(¬дощ) =

0.4x0.7 + 0.8x0.3 = 0.28 + 0.24 = 0.52.

8.4. Теорема Байєса

Теорема Байєса названа на честь проповідника вісімнадцятого сторіччя Томаса Байєса. Теорема має багато застосувань і вважається головноютеоремою статистики.

Теорема Байєса(її ще називають правилом Байєса) записується наступним чином

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)},$$

де

A та B – події;

P(B) > 0;

P(A | B) ймовірність настання події A, якщо відбулася подія B;

P(B | A) ймовірність настання події B, якщо відбулася подія A.

Приклад 8.2.В 2009 році в найвищий відсоток захворюваності на ВІЛ/СНІД було зафіксовано в Свазіленді і становить 25.9%. Тест ELISA один з найкращих та найточніших тестів. Для тих, хто хворий на СНІД тест має точність 99.7%, для тих хто не хворий 92.6%. Якщо за результатами тесту людина ВІЛ інфікована, яка ймовірність що вона дійсно хвора? Тоді:

- P(хвора) = 0.259 - пріорна можливість захворіти;
- P(тест +| хвора) = 0.997 - ймовірність. що тест покаже позитивний результат, якщо людина хвора;
- P(тест -|не хвора) = 0.926 ймовірність. що тест покаже негативний результат, якщо людина здорова

Потрібно оцінити ймовірність що вона дійсно хвора, якщо тест показав позитивний результат, тобто P(хвора | тест +).

За теоремою Байєса

$$P(\text{õâîðà} \mid \text{òåñò} +) = \frac{P(\text{òåñò} + \mid \text{õâîðà}) \times P(\text{õâîðà})}{P(\text{òåñò} +)}.$$

Для обчислення нам не вистачає лише інфомації яка ймовірність того, що

тест дасть позитивний результат для будь-якого жителя (чи жительки) Свазіленда. Обчислимо цю ймовірність

P(тест +) = P(тест+ і хвора) + P(тест+ і ¬хвора) =

P(тест+ | хвора)P(хвора) + P(тест+ | ¬хвора)P(¬хвора);

P(¬хвора) = 1 - P(хвора) = 0.741;

P(тест + | ¬хвора) = 1 - P(тест- | ¬не хвора) = 1 - 0.926 = 0.074;

P(тест+) = 0.997x0.259 + 0.074x0.741 = 0.2582 + 0.0548 = 0.313;

$$P\left(\text{õâî ðà} \mid \text{ò åñò } +\right) = \frac{P\left(\text{ò åñò } + \mid \text{õâî ðà}\right) \times P\left(\text{õâî ðà}\right)}{P\left(\text{ò åñò } +\right)} =$$

$$= \frac{0,997 \times 0,259}{0,313} = 0,825.$$

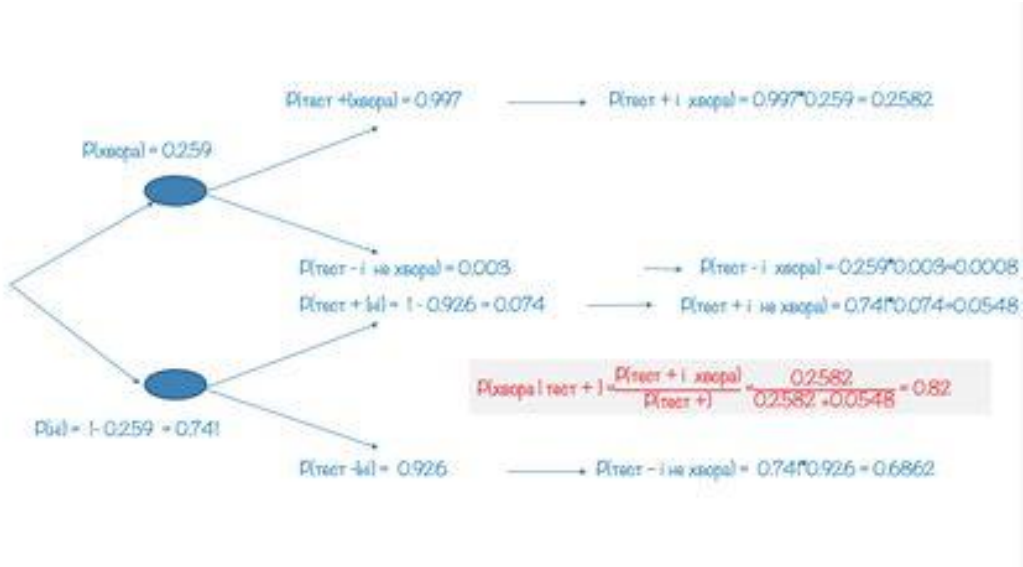


Рис. 8.4 Візуалізація ймовірностей

Для знаходження ймовірностей також зручно скористатись візуалізацією (рис. 8.4).

Лекція 9.

КЛАСИЧНІ РОЗПОДІЛИ

Вступ

9.1. Вступ

Якщо ми знаємо, що наші дані належать до якогось з класичних розподілів, то можемо використати вже вивчені властивості цих розподілів. Є дискретні та неперервні класичні розподіли. Як приклад дискретного розглянемо біноміальний, а як приклад неперервного – нормальний розподіл.

9.2. Біноміальний розподіл

Біноміальний розподіл– це дискретний розподіл, тобто розподіл величини, яка може набирати фіксованих значень. Уявіть собі підкидання монетки 5 разів. Монета може випасти орлом 0, 1, 2, 3, 4 або 5 разів, але не 0.67 чи 3.57. Відповідно змінна, яка описує кількість разів, які монета впала орлом є дискретною змінною.

Біноміальний розподіл підходить для опису розподілу даних, де результати, можуть набирати лише двох можливих значень (від виходу з ладу деталей машин до студентів, які здають іспит).

Події в біноміальному розподілі генеруються внаслідок процесу Бернуллі. Одне випробування в процесі Бернуллі має назву випробування Бернуллі. Коли кожне випробування має лише два можливих наслідки. Ці наслідки класифікуються як "успіх" чи "невдача". "Успіх" – не обов'язково має позитивний контекст. Наприклад, "успіхом" може бути наслідок "вихід з ладу важливої деталі".

Розглянемо біноміальний розподіл на прикладі експеримента Мілґрема

[5].

Експеримент почався в липні 1961, через три місяці після того, як почався процес над нацистським військовим злочинцем Адольфом Ейхманом в Єрусалимі. Мілґрем задумав експеримент, щоб дати відповідь на питання: "Чи міг Ейхман і мільйони його спільників по Голокосту просто виконувати накази? Чи можемо ми їх всіх називати спільниками?" Експериментатор (E) вимагав від "вчителя" (T) давати "учневі" (L) прості завдання на запам'ятовування і при кожній помилці "учня" натискати на кнопку, нібито карає його ударом струму (насправді актор, що грав "учня", тільки вдавав, що отримує удари). Почавши з 15 вольт, "вчитель" з кожною новою помилкою повинен був збільшувати напругу на 15 вольт (верхня допустима межа в експерименті 450 вольт). В одній серії дослідів основного варіанту експерименту 26 досліджуваних з 40, замість того щоб змилосердитися над жертвою, продовжували збільшувати напругу (до 450 В) до тих пір, поки дослідник не віддавав розпорядження закінчити експеримент.

Будемо розглядати кожну особу в експерименті Мілгрема як випробування. **Успіхом** будемо вважати подію, коли особа відмовилась продовжувати експеримент, **невдачею** – якщо погодилась. Оскільки 35%(14 із 40) відмовляється то ймовірність успіху в одній спробі 35%.

Якщо ми виберемо для експерименту трьох випадкових людей, яка ймовірність що один з них відмовиться? Назвемо цих людей Антон, Богдан та Вікторія.

Якщо відмовиться Антон, то цей варіант опишемо як Варіант 1: (Успіх Невдача Невдача), ймовірність незалежних подій дорівнює добутку ймовірностей, тобто ймовірність того, що відмовиться саме Антон (це означає, що Богдан та Вікторія не відмовляться) дорівнює 0.35 * 0.65 * 0.65 = 0.149.

Якщо відмовиться Богдан, варіант описується як Варіант 2: (Невдача Успіх Невдача), ймовірність 0.65 * 0.35 * 0.65 = 0.149.

Якщо ж відмовиться Вікторія, то маємо Варіант 3: (Невдача Невдача Успіх), ймовірність якого 0.35 * 0.35 * 0.65 = 0.149.

Для оцінки ймовірності, що відмовиться продовжувати одна людина, нам не важливо знати, хто саме це буде. Тобто нам треба знайти ймовірність настання варінту 1, 2 або 3, що дорівнює сумі ймовірностей цих варіантів і дорівнює 0.44.

Як бачимо, для обчислення ймовірності мати k успіхів в n незалежних випробуваннях Бернуллі з ймовірністю успіху p в кожному випробуванні використовується два компоненти:

- кількість можливих сценаріїв. Обчислюється за формулою

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

- ймовірність одного сценарія Обчислюється за формулою

$$p^k (1-p)^{(n-k)}$$

В загальному формула ймовірності P мати k успіхів в n незалежних випробуваннях Бернуллі з ймовірністю успіху p в кожному випробуванні може бути обчислена наступним чином

$$P = \frac{n!}{k!(n-k)!} p^k (1-p)^{(n-k)}.$$

При цьому повинні виконуватись наступні умови

- Випробування незалежні;
- Кількість випробувань n фіксована;
- Кожен результат класифікується як успіх або невдача;
- Ймовірність успіху p однакова для кожного випробування.

Приклад 9.1.Згідно опитування Gallup poll 2012 26.2% жителів США мають надмірну вагу. Яка ймовірність серед 20 випадковим чином обраних жителів отримати 5 з надлишковою вагою?

Отже, маємо n = 20, k = 5, p = 0.262.

Кількість варіантів $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.

n! (n факторіал) – добуток всіх чисел від 1 до n (тобто $1 \times 2 \times 3 \times ... \times n$).

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{20!}{5!(20-5)!} = 15504.$$

Для обчислення $\binom{n}{k}$ в R можемо використати функцію choose

```
choose(n=20, k=5)
## [1] 15504
```

Ймовірність одного варіанту

$$p^k(1-p)^{(n-k)} = 0,262^5 \times 0,738^{15} = 0,00001295$$

Ймовірність обрати серед 20 жителів 5 з надлишковою вагою дорівнює добутку 15504 і 0.00001295 і дорівнює 0.2. Також для знаходження цього значення в R можемо скористатись функцією dbinom

```
dbinom(x=5, size=20, prob=0.262)
## [1] 0.2008148
```

9.3. Нормальний розподіл

Нормальний розподіл є класичним неперервним розподілом. Він описує розподіл багатьох неперервних величин від зросту людини до результатів виборів. Ще має назву "розподіл Гауса" (на честь Карла Фрідріха Гауса, який використовував цей розподіл для аналізу даних в астрономії).

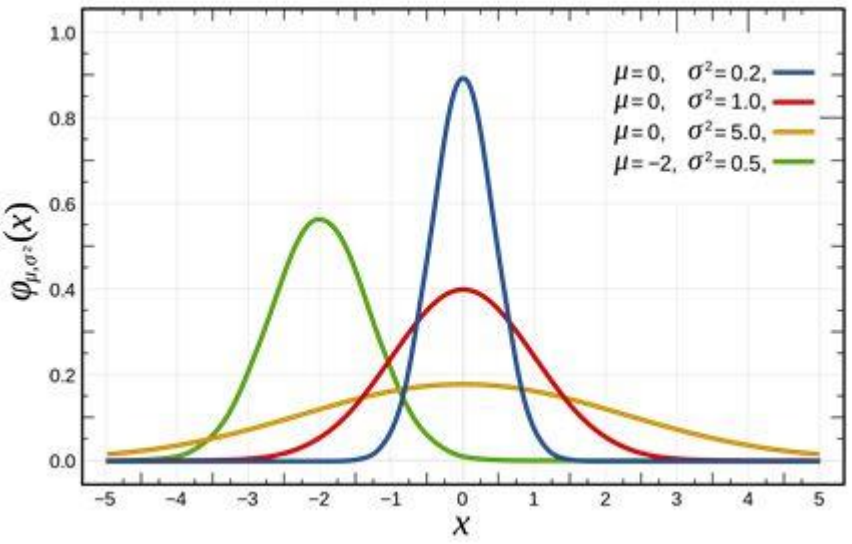


Рис. 9.1 Нормальні розподіли

Існує нескінченна кількість нормальних розподілів в залежності від

їхнього середнього значення μ та середньоквадратичного відхилення σ (рис. 9.1).

Нормальний розподіл з середнім значенням μ = 0 та σ = 1 має назву **стандартний нормальний розподіл** або **Z-розподіл**. Будь-який нормальний розподіл може бути зведений до стандартного нормального розподілу шляхом Z-стандартизації. Формула для обчислення z-значень має наступний вигляд

$$\frac{x-\mu}{\sigma}.$$

Відзначимо, що в процесі z-стандартазиції (нормалізації):

- Форма розподілу не змінюється;
- Середнє значення стає нулем;
- Середньоквадратичне відхилення стає одиницею.

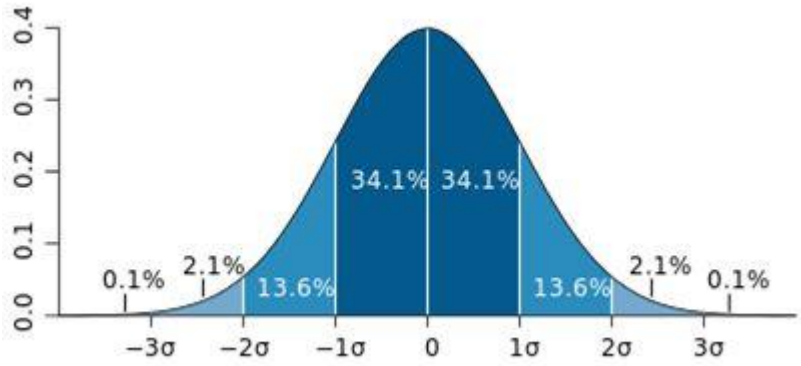


Рис. 4.2 Властивості нормального розподілу

Властивості нормального розподілу наведено у [6] (рис. 4.2):

- Симетричність;
- Юнімодальність(лише одна мода);
- Набирає значень від -∞ до +∞;
- Загальна площа під кривою дорівнює 1;
- Однакове значення медіани, моди та середнього значення.

Також для нормального розподілу відомо, що

- близько 68% значень знаходяться в межах одного середньоквадратичиного відхилення від середнього значення;
- близько 95% значень знаходяться в межах двох

середньоквадратичиних відхилень від середнього значення;

- близько 99% значень знаходяться в межах трьох середньоквадратичиних відхилень від середнього значення.

Тобто, знаючи що розподіл є нормальним ми можемо визначити, наскільки типовим чи екстремальним є конкретне значення. Ми можемо також оцінити, якою є **ймовірність отримати конкретне z-значення**.

Для того щоб оцінити ймовірність отримати конкретне z-значення потрібно:

- Зробити z- стандартизацію;
- Зобразити розподіл;
- Визначаємо, який саме відрізок площі під кривою нас цікавить;
- Знайти значення в z-таблицях чи з допомогою функції pnormв R.

Лекція 10.

ОСНОВИ КОРЕЛЯЦІЙНОГО АНАЛІЗУ

План

10.1. Вступ

До цього часу ми працювали над аналізом однієї змінної. Тепер ми перейдемо до аналізу взаємозв’язків між двома змінними. В багатьох прикладних задачах необхідно виявити залежність між двома властивостями (ознаками) X і Y одного і того ж об’єкта або між певними ознаками різних об’єктів. Якщо вказані ознаки допускають кількісне вимірювання, і ознака Y залежить від ознаки X,тоді X можна назвати незалежною змінною або **факторною ознакою**, а Y – залежною змінною або **результативною ознакою**.

10.2. Поняття кореляційного зв’язку між досліджуваними величинами

Якщо кожному значенню факторної ознаки X відповідає одне і тільки одне значення результативної ознаки Y, то говорять, що між цими ознаками існує **функціональний зв’язок**: $Y= f(X)$.

Якщо кожному значенню факторної ознаки X відповідає безліч значень результативної ознаки Y, то говорять, що між цими ознаками існує **статистичний зв’язок.**

Наприклад, якщо X приймає l значень $X = \{x_1, x_2, ... x_l\}$ і кожному її значенню x_i відповідає множина значень Y, тобто

Значенню x_1 відповідає множина $\{y_{11}, y_{12}, ... y_{1m_1}\}$;

Значенню x_2 відповідає множина $\{y_{21}, y_{22}, \dots y_{2m_2}\}$;

…

Значенню x_l відповідає множина $\{y_{l1}, y_{l2}, \dots y_{lm_l}\}$,

то між *X* та *Y* існує статистичний зв’язок.

Вивчення статистичного зв'язку вважається дуже складним і трудомістким процесом, у якому потрібно аналізувати багатовимірні таблиці даних. Тому, зазвичай, вивчається не статистичний, а кореляційний зв'язок між X та Y.

Якщо кожному значенню факторної ознаки X відповідає певне середнє значення результативної ознаки Y, то говорять, що між цими ознаками існує кореляційний зв’язок. Тобто кореляційною є функціональна залежність між значеннями X і середніми значеннями Y: $\overline{Y} = f(X)$.

Наприклад, якщо X приймає l значень $X = \{x_1, x_2, \dots x_l\}$ і кожному її значенню x_i відповідає середнє множини значень *Y*, тобто

Значенню x_1 відповідає множина $\overline{y}_{x_1} = \dfrac{y_{11} + y_{12} + \dots + y_{1m_1}}{m_1}$;

Значенню x_2 відповідає множина $\overline{y}_{x_2} = \dfrac{y_{21} + y_{22} + \dots + y_{2m_2}}{m_2}$;

…

Значенню x_l відповідає множина $\overline{y}_{x_l} = \dfrac{y_{l1} + y_{l2} + \dots + y_{lm_l}}{m_l}$,

то між *X* та *Y* існує кореляційний зв’язок.

Основними задачами кореляційного аналізу є:

− вивчення сили зв’язку між двома і більше ознаками досліджуваного об’єкта;

− встановлення факторів, що найбільш суттєвовпливаютьна результативну ознаку;

− виявлення невідомих причинно-наслідкових зв’язків між ознаками об’єкта.

10.3. Групування даних для кореляційного аналізу

Вибіркові дані для вивчення кореляційного зв'язку між ознаками X та Y мають вигляд пар їх значень: $(x_1; y_1)$, $(x_2; y_2)$, …, $(x_n; y_n)$, x_i –значення величини X, y_i –значення величини Y, n – кількість пар значень, $i = \overline{1,n}$.

Якщо кількість пар значень достатньо велика (принаймні $n > 20$), то для зручності розрахунків дані групуються.

Для групування даних необхідно:

1) Розбити множини значень X та Y на інтервали, використовуючи формулу Стерджеса, тобто $k = 1 + 1,4 \ln n$. Кількість інтервалів для X та Y у загальному випадку може бути різною (позначення: k – кількість інтервалів для X; m – кількість інтервалів для Y).

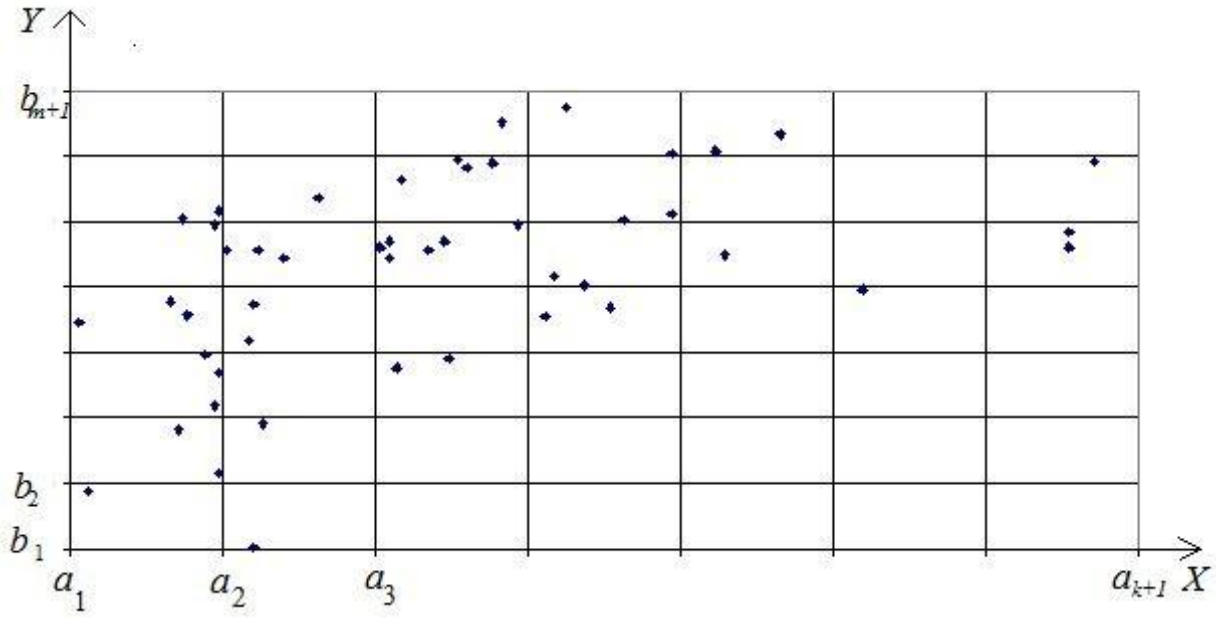


Рис. 10.1Поле кореляції

2) Зобразити дані графічно: побудувати на площині точки з координатами $(x_i; y_j)$. В результаті отримується площина, розбита на прямокутники, в кожному з яких може бути множина точок(рис. 10.1).Вказане графічне зображення вибіркових даних називається **полем кореляції**.

Таблиця 10.1

Кореляційна таблиця

X (інтервали і їх середини)		$[a_1;a_2)$	$[a_2;a_3)$	…	$[a_k;a_{k+1})$	$n_j=\sum_{i=1}^{k} n_{ij}$
Y(інтервали і їх середини)		x_1	x_2	…	x_k	
$[b_1;b_2)$	y_1	n_{11}	n_{21}	…	n_{k1}	n_1
$[b_2;b_3)$	y_2	n_{12}	n_{22}	…	n_{k2}	n_2
…	…	…	…	…	…	…
$[b_m;b_{m+1})$	y_m	n_{1m}	n_{2m}	…	n_{km}	n_m
$n_i=\sum_{j=1}^{m} n_{ij}$		n_1	n_2	…	n_k	

3) Побудувати кореляційну таблицю(табл. 10.1). В першому рядку, розбитому на дві частини, записуються інтервали $[a_i;a_{i+1})$ для X та їх середини x_i. У першому стовпці, розбитому на дві частини, записуються інтервали $[b_j;b_{j+1})$для Y та їх середини y_j.В центральній частині таблиці записуються частоти n_{ij} – кількість точок, що потратили в прямокутник, обмежений по X інтервалом $[a_i;a_{i+1})$ і по Y інтервалом $[b_j;b_{j+1})$. В останньому рядку таблиці записуються частоти n_i для X – кількості точок, що потратили в прямокутники, які відповідають інтервалу $[a_i;a_{i+1})$, тобто $n_i=\sum_{j=1}^{m} n_{ij}$ – сума частот n_{ij} в стовпці з номером i. В останньому стовпці таблиці записуються частоти n_j для Y – кількості точок, що потрапили в прямокутники, які відповідають інтервалу $[b_j;b_{j+1})$, тобто $n_j=\sum_{i=1}^{k} n_{ij}$ – сума частот n_{ij} в рядку з номером j.Кореляційну таблицю можна розглядати як своєрідний подвійний статистичний ряд.

Таблиця 10.2

Залежність середнього значення Y від X

x_i	x_1	x_2	…	x_k
$\overline{y}_{x_i}$	$\overline{y}_{x_1}$	$\overline{y}_{x_2}$	…	$\overline{y}_{x_k}$
n_i	n_1	n_2	…	n_k

4) За даними кореляційної таблиці будується ряд, що відображає залежність середнього значення Y від X (табл. 10.2).В першому рядку таблиці записуються середини інтервалів x_i. В другому – відповідні середні значення $\overline{y}_{x_i}$, що знаходяться за формулами

$$\overline{y}_{x_1} = \frac{y_1 n_{11} + y_2 n_{12} + \ldots + y_m n_{1m}}{n_1};$$

$$\overline{y}_{x_2} = \frac{y_1 n_{21} + y_2 n_{22} + \ldots + y_m n_{2m}}{n_2};$$

…

$$\overline{y}_{x_k} = \frac{y_1 n_{k1} + y_2 n_{k2} + \ldots + y_m n_{km}}{n_k}.$$

В результаті отримується статистичний ряд, що містить значення *X*, відповідні середні значення *Y* та частоти. За даними такого ряду проводиться кореляційний аналіз.

10.4. Коефіцієнт кореляції Пірсона

До цього часу ми працювали над аналізом однієї змінної. Тепер ми перейдемо до аналізу взаємозв’язків між двома змінними.

Коваріація – міра лінійної залежності двох випадкових величин одна від одної, яка визначається формулою

$$\operatorname{cov}(X,Y) = \sum_{i=1}^{k}\sum_{j=1}^{m}\left(x_i - \overline{x}\right)\left(y_j - \overline{y}\right).$$

Для оцінки тісноти (або сили) зв’язку між X та Y існуєзважена версія коваріації, яка називається**кореляцією**. У випадку, коли між *X* та *Y* існує

лінійний зв’язок та вибіркові дані розподілені за нормальним законом, використовується **коефіцієнт кореляції Пірсона**, який ще називається параметричним коефіцієнтом кореляції. Коефіцієнт кореляції Пірсона розраховується за формулою

$$r = \frac{\overline{xy} - \bar{x} \times \bar{y}}{S_X \times S_Y},$$

де

$\bar{x}$ – вибіркове середнє величини *X*;

$\bar{y}$ – вибіркове середнє величини *Y*;

$\overline{xy}$ – вибіркове середнє величини *XY*;

S_X – вибіркове середнє квадратичне відхилення величини *X*;

S_Y – вибіркове середнє квадратичне відхилення величини *Y*.

Враховуючи формули для знаходження вибіркових середніх і середніх квадратичних відхилень, а саме

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{k} x_i n_i ;$$

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{m} y_j n_j$$

$$\overline{xy} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{m} x_i y_j n_{ij} ;$$

$$S_X = \sqrt{\frac{1}{n}\sum_{i=1}^{k} x_i^2 n_i - \left(\frac{1}{n}\sum_{i=1}^{k} x_i n_i\right)^2} ;$$

$$S_Y = \sqrt{\frac{1}{n}\sum_{j=1}^{m} y_j^2 n_j - \left(\frac{1}{n}\sum_{j=1}^{m} y_j n_j\right)^2}$$

можна отримати більш зручну для розрахунків формулу

$$r = \frac{\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{m} x_i y_j n_{ij} - \left(\frac{1}{n}\sum_{i=1}^{k} x_i n_i\right) \times \left(\frac{1}{n}\sum_{j=1}^{m} y_j n_j\right)}{\sqrt{\frac{1}{n}\sum_{i=1}^{k} x_i^2 n_i - \left(\frac{1}{n}\sum_{i=1}^{k} x_i n_i\right)^2} \times \sqrt{\frac{1}{n}\sum_{j=1}^{m} y_j^2 n_j - \left(\frac{1}{n}\sum_{j=1}^{m} y_j n_j\right)^2}}.$$

У випадку незгрупованих даних розрахункова формула суттєво спрощується

$$r = \frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var}(X)}\sqrt{\operatorname{var}(Y)}} = \frac{\sum_{i=1}^{k}\left(x_i - \bar{x}\right) \times \sum_{j=1}^{m}\left(y_m - \bar{y}\right)}{\sqrt{\sum_{i=1}^{k}\left(x_i - \bar{x}\right)^2}\sqrt{\sum_{j=1}^{m}\left(y_j - \bar{y}\right)^2}},$$

де

$\operatorname{var}(X)$ – дисперсія величини X;

$\operatorname{var}(Y)$ – дисперсія величини Y.

В R коефіцієнт для обчислення коефіцієнта кореляції використовується функція cor, яка по замовчуванню обчислює коефіцієнт кореляції Пірсона.

Абсолютне значення коефіцієнта кореляції дає уявлення про силу лінійного зв'язку між двома змінними. Знак коефіцієнта вказує напрямок зв'язку. Коефіцієнт кореляції набуває значень [– 1,1]. Якщо коефіцієнт близький до 1 – говорять про сильну позитивну кореляцію, до -1 про сильну негативну. Значення коефіцієнта близькі до 0 вказують на відсутність лінійної кореляції.

Властивості коефіцієнта кореляції Пірсона:

- коефіцієнт кореляції не змінюється при зміні одиниць виміру;
- коефіцієнт кореляції є симетричним r(x, y) = r(y, x);
- коефіцієнт кореляції є чутливий до викидів.

Слід пам’ятати, що коефіцієнт кореляції Пірсона показує силу лінійного зв’язку. Якщо між X та Y існує сильний нелінійний зв’язок, коефіцієнт кореляції Пірсона може дорівнювати нулю.

Оскільки сила зв'язку між *X* та *Y* оцінюється за вибірковими даними, то необхідна перевірка її **статистичної значущості**, тобто оцінка можливості розповсюдити отримані результати на всю генеральну сукупність.

Перевірка статистичної значущості коефіцієнта кореляції Пірсона здійснюється за допомогою так званої *t*-статистики, яка розраховується за формулою

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}.$$

Розраховане значення *t*-статистики порівнюється з критичним значенням $t_{êðèò}$. $t_{êðèò}$ – табличне значення розподілу Стьюдента.Якщо розраховане значення *t*-статистики більше критичного $|t| > t_{êðèò}$, то коефіцієнт кореляції вважається значущим.

10.5. Коефіцієнт кореляції Спірмена

Для оцінки сили зв'язку між *X* та *Y* у випадку, коли між *X* та *Y* існує нелінійний зв'язок або вибіркові дані не розподілені за нормальним законом, варто використовувати коефіцієнт кореляції Спірмена.

Коефіцієнт кореляції Спірмена розраховується за формулою

$$r_S(X,Y) = 1 - \frac{6\sum_{i=1}^{n} d_i^2 + T_X + T_Y}{n\left(n^2-1\right)},$$

де

n –кількість пар вибіркових даних;

d_i – різниці між рангами *i*-го значення *X* та відповідного значення *Y*;

T_X, T_Y – поправки, пов'язані з однаковими рангами; розраховуються за формулами

$$T_X = \frac{\sum_{i=1}^{L_X}\left(T_{X_i}^3 - T_{X_i}\right)}{12};\ T_Y = \frac{\sum_{i=1}^{L_Y}\left(T_{Y_i}^3 - T_{Y_i}\right)}{12},$$

де

L_X, L_Y – кількість зв’язок (груп однакових рангів);

T_{X_i}, T_{Y_i} – розміри *i*-тих зв’язок (кількість елементів в них).

Даним вибірки присвоюються ранги (порядкові номери) за правилом: найменшому значенню присвоюється ранг 1, наступному найменшому – ранг 2 і т. д. При цьому, якщо деякі елементи вибірки співпадають, то їм присвоюються середні ранги. Для цього додаються ранги, які мали б ці елементи, якщо були б різні; розраховується їх середнє арифметичне; кожному із однакових елементів присвоюється ранг, що дорівнює розрахованому середньому арифметичному.

Статистична значущість коефіцієнта кореляції Спірмена перевіряється так, як і коефіцієнта кореляції Пірсона.

Лекція 11.

ЛІНІЙНА РЕГРЕСІЯ

План

11.1. Вступ

11.2. Встановлення виду кореляційної залежності

11.3. Лінійна регресія

11.1. Вступ

При вивченні тісноти зв'язку між різними ознаками об'єкта головною задачею є встановлення виду кореляційної залежності результативної ознаки (*Y*) від факторної (*X*), тобто видуфункціональної залежності $\overline{Y} = f\left(X\right)$.В першу чергу це пов'язано з необхідністю прогнозування досліджуваних процесів. Математико-статистичний апарат, що дозволяє встановити вид кореляційної залежності називається **регресійним аналізом**, а функція, яка описує цю залежність, називається **рівнянням регресії**.

11.2. Встановлення виду кореляційної залежності

Регресійний аналіз проводиться за такими етапами:

1) Встановлення виду кореляційної залежності результативноїознаки *Y* від факторної ознаки *X*;
2) Побудова регресійної моделі;
3) Перевірка статистичної значущості побудованої моделі.

Перший етап регресійного аналізу є найважливішим, оскільки помилки у виборі виду залежності призводять до побудови регресійної моделі, що не відповідає емпіричним даним і не може використовуватися для прогнозування.

Таблиця 11.1

Статистичний ряд

x_i	x_1	x_2	…	x_k
$\overline{y}_{x_i}$	$\overline{y}_{x_1}$	$\overline{y}_{x_2}$	…	$\overline{y}_{x_k}$
n_i	n_1	n_2	…	n_k

Вибіркові дані для вивчення кореляційного зв'язку між ознаками *X* та *Y*, зазвичай, мають вигляд пар їх значень: $(x_1; y_1)$, $(x_2; y_2)$, …, $(x_n; y_n)$, x_i – значення величини *X*, y_i – значення величини *Y*, *n* – кількість пар значень, $i = \overline{1,n}$.Якщо їх кількість достатньо велика, то для зручності розрахунків дані групуються і будується статистичний ряд, що містить значення *X*, відповідні середні значення *Y* та частоти (табл. 11.1).

Згруповані дані (табл. 11.1) зображуються графічно, що часто дозволяє визначити вид залежності *Y* від *X*.

Ламаналінія, щоз'єднуєточкизкоординатами $(x_i; y_i)$ називається**емпіричноюлінієюрегресії**.

Рис. 11.1 Гіпотетична лінійна залежність

Якщо емпірична лінія регресії наближається до прямої лінії, то висувається гіпотеза про наявність лінійного зв'язку між досліджуваними ознаками (рис. 11.1).

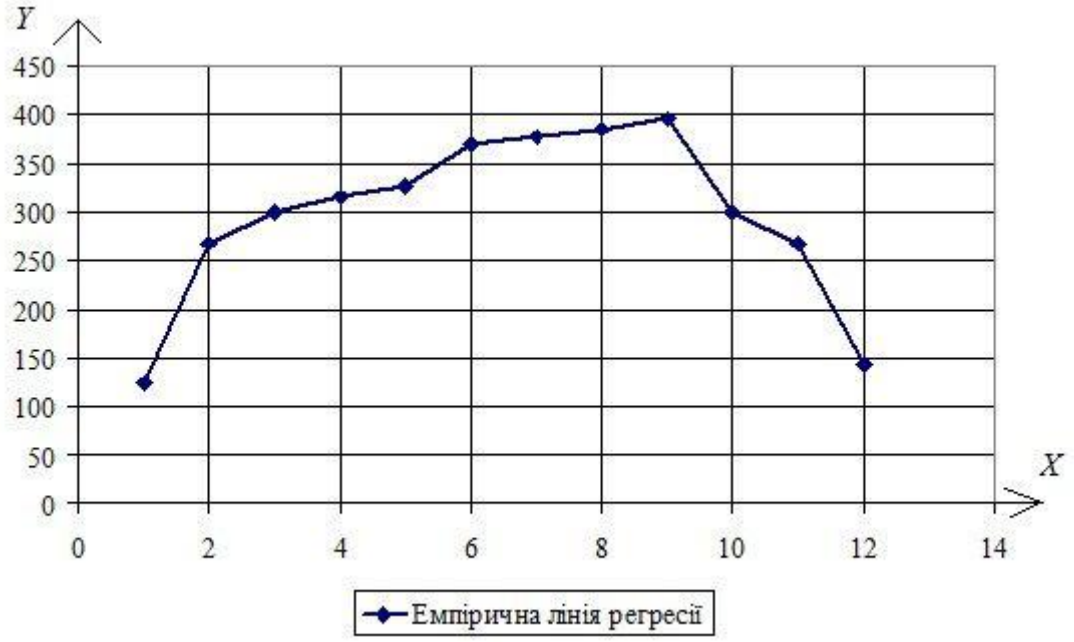


Рис. 11.2 Гіпотетична нелінійна залежність

В іншому випадку висувається гіпотеза про наявність нелінійного зв'язку (рис. 11.2).

11.3. Лінійна регресія

Якщо висунуто гіпотезу про наявність лінійної залежності результативної ознаки (Y) від факторної (X), то рівняння регресії має вид

$$\bar{y} = ax + b, \qquad (11.1)$$

де a, b − параметри моделі.

Побудова лінійної регресійної моделі – це знаходження параметрів рівняння (11.1). Параметри рівняння регресії можна знайти за **методом найменших квадратів**.

Наприклад, при вивчення залежності Y від X було отримано наступні вибіркові дані: $x_1, x_2, \ldots, x_n$ – значення величини X, $y_1, y_2, \ldots, y_n$ – відповідні значення Y. За вибірковими даними було побудовано рівняння регресії

$y = ax + b$. Якщо в рівняння підставити замість x значення $x_1, x_2, \dots, x_n$, то будуть отримані теоретичні значення Y: $y_{1,\text{òåîð}}, y_{2,\text{òåîð}}, \dots, y_{n,\text{òåîð}}$, які відрізняються від $y_1, y_2, \dots, y_n$. Різниця значень $y_{i,\text{òåîð}} - y_i$ називається помилкою регресійної моделі і позначається e_i. Якщо параметри рівняння підбираються так, щоб сума квадратів помилок була мінімальною, то говорять, що вони отримані за методом найменших квадратів.Тобто у випадку лінійної регресії потрібно знайти такі коефіцієнти a та b, при якихфункція

$$L(a,b) = \sum_{i=0}^{n} \left(y_i - ax_i - b\right)^2$$

буде приймати мінімальне значення.

Існування вказаного мінімуму не викликає сумнівів, оскільки функція $L(a,b)$, як функція від a та b є многочленом другої степені, а також $L(a,b) \geq 0$.

Умовою екстремуму функції $L(a,b)$ слід вважати рівність нулю часткових похідних, узятих за параметрами a та b

$$\begin{cases} \dfrac{\partial}{\partial a} L(a,b) = 0 \\ \dfrac{\partial}{\partial b} L(a,b) = 0 \end{cases}.$$

Звідси

$$\frac{dL}{da} = -2\sum_{i=1}^{n} \left[y_i - (ax_i + b)\right] x_i = 0$$

$$\frac{dL}{da} = -2\sum_{i=1}^{n} \left[y_i - (ax_i + b)\right] = 0.$$

Скоротивши на -2 і розкривши квадратні дужки, параметри рівняння регресії за методом найменших квадратів знаходяться з системи лінійних алгебраїчних рівнянь

$$\begin{cases} a\sum_{i=1}^{k} x_i^2 n_i + b\sum_{i=1}^{k} x_i n_i = \sum_{i=1}^{k} x_i n_i \overline{y}_{x_i} \\ a\sum_{i=1}^{k} x_i n_i + b\sum_{i=1}^{k} n_i = \sum_{i=1}^{k} n_i \overline{y}_{x_i} \end{cases}. \qquad (11.2)$$

Якщо вибіркові дані не згруповані, то система (11.2) значно спрощується

$$\begin{cases} a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \\ a\sum_{i=1}^{n} x_i + bn = \sum_{i=1}^{n} y_i \end{cases}$$

Перевірка правильності побудови рівняння регресії здійснюється за основним варіаційним рівнянням

$$Q = Q_p + Q_o$$

де

$Q = \sum_{i=1}^{k} \left(\overline{y}_{x_i} - \overline{y} \right)^2 n_i$ – загальна варіація, тобто сума квадратів відхиленьемпіричних значень *Y* від середнього $\overline{y} = \frac{\sum_{i=1}^{k} \overline{y}_{x_i} n_i}{n}$;

$Q_p = \sum_{i=1}^{k} \left(y_{i,\grave{o}\hat{a}\hat{i}\eth} - \overline{y} \right)^2 n_i$ –варіація регресії, тобто сума квадратів відхиленьтеоретичних значень *Y* від середнього, що обумовленарегресією;

$Q_o = \sum_{i=1}^{k} \left(y_{i,\grave{o}\hat{a}\hat{i}\eth} - \overline{y}_{x_i} \right)^2 n_i$ –варіація залишків, тобто сума квадратів відхиленьтеоретичних значень *Y* від емпіричних.

У випадку незгрупованих даних загальна варіація, варіації регресії ізалишківзнаходятьсязаформулами

$$Q = \sum_{i=1}^{n} \left(\overline{y}_{x_i} - \overline{y} \right)^2;\ Q_p = \sum_{i=1}^{n} \left(y_{i,\grave{o}\hat{a}\hat{i}\eth} - \overline{y} \right)^2;\ Q_o = \sum_{i=1}^{k} \left(y_{i,\grave{o}\hat{a}\hat{i}\eth} - y_i \right)^2 n_i,$$

де $\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}$.

Для перевірки статистичної значущості рівняння регресії розраховуєтьсяF-статистика за формулою

$$F = \frac{Q_p\left(n-l\right)}{Q_o\left(l-1\right)},$$

де n – кількість спостережень, l – кількість груп у кореляційній таблиці або кількість параметрів моделі у випадку незгрупованих даних.Розраховане значення F-статистики порівнюється з критичним значенням $F_{кр}$ розподілу Фішера, яке можна знайти за статистичними таблицями.

Адекватність моделі вибірковим даним можна оцінити за коефіцієнтом детермінації R^2, що показує частину варіації значень результативної ознаки Y, що пояснюється рівнянням регресії. Коефіцієнт детермінації розраховується за формулою

$$R^2 = 1 - \frac{Q_o}{Q} = \frac{Q_p}{Q}.$$

Значення коефіцієнта детермінації знаходяться в інтервалі $\left[0;1\right]$, тобто $0 \le R^2 \le 1$. Чим ближче R^2 до 1, тим краще отримане рівняння регресії пояснюєповедінку результативної ознаки.Наприклад, якщо $R^2 = 0{,}98$, то 98% варіаціїрезультативної ознаки Y пояснюється рівнянням регресії.

У R для знаходження найкращої лінії, яка й буде нашою моделлю, можна використовувати функцію lm.

Для оцінки результатів лінійної моделі використовується функція summary.

Отже, можемо визначити наступні умови для побудови лінійної регресії:

- Лінійність (тобто наявність лінійної залежності між незалежною та залежною змінною);
- Нормальний розподіл залишків;
- Гомоскедастичність (стала варіативність залишків).

За посиланням https://gallery.shinyapps.io/slr_diag/ ми можемо змоделювати дані з різними типами залежності та дослідити, як при цьому будуть виглядати лінія регресії, коефіцієнт кореляції, R^2, та як виглядає розподіл залишків.

Екстраполяція – застосування моделі (у данному випадку лінійної регресії), до діапазону даних, дляякого моделювання не проводилося. Сам підхід гарно ілюструє XKCD комікс http://xkcd.com/605/. Якщо ви сьогодні вийшли заміж, то вчора у вас було 0 чоловіків, сьогодні 1, через місяць 30, а через рік 365. Важливо уникати екстраполяції, оскільки ми не знаємо, як зміниться тренд для даних, яких ми ще не бачили.

Лекція 12.

ВИВІДНА СТАТИСТИКА

План

12.1. Вступ

12.2. Оцінки параметрів вибірки

12.3. Центральна гранична теорема

12.4. Довірчий інтервал

12.5. Довірчий інтервал для середнього значення

12.1. Вступ

Описова статистика – вивчає властивості спостережуваних даних. **Вивідна статистика** – виводимо припущення про властивості розподілу даних з яких походять спостережувані дані.

Описова статистика цікавиться виключно властивостями спостережуваних даних, і не припускає, що ці дані можуть походити з більшої сукупності. Вивідна статистика дозволяє робити висновки про генеральну сукупність на основі вибірки. Впевненість у цих висновках можна представити чисельно.

12.2. Оцінки параметрів вибірки

Розуміння термінів "**генеральна сукупність**" та "**вибірка**" є надзвичайно важливим для розуміння вивідної статистики.

"Генеральна сукупність" складається з усіх об’єктів (наприклад людей, лососів в Атлантичному океані, деталей літака) предмету, які хотів би вивчити дослідник, якби в нього була необмежена кількість ресурсів. Визначення генеральної сукупності, яка досліджується, є першим кроком у вивідній статистиці. Прикладом генеральної сукупності може бути населення США у 2016 році або кількість канадських чоловіків віком 65-70 років з діагнозом серцева недостатність.

Таблиця 12.1

Статистичний процес для вибірки та генеральної сукупності

	Вибірка	Генеральна сукупність
Розмір	n	N
Середнє значення	$\bar{x} = \frac{\sum x}{n}$	$\mu = \frac{\sum x}{n}$
Дисперсія	$S^2 = \frac{\sum \left(x - \bar{x}\right)^2}{n-1}$	$\sigma^2 = \frac{\sum \left(x - \mu\right)^2}{n}$
Середньоквадратичне відхилення	$S = \sqrt{S^2}$	$\sigma = \sqrt{\sigma^2}$
пропорція	$\bar{p} = \frac{n\ óñïì³õ³â}{n\ âèïðîáóâàíü}$	$p = \frac{N\ óñïì³õ³â}{N\ âèïðîáóâàíü}$

Статистичний процес для вивідної та описової статистик однаковий, але трактування різне (табл. 12.1). Наприклад, середнє значення обчислюється однаково як для популяції так і для вибірки. Для їх розрізнення використовується інша нотація.

Потрібно зауважити, що формули для обчислення дисперсії (а отже, і середньоквадратичного відхилення) для генеральної сукупності та вибірки відрізняються. Така корекція називається **поправкою Бесселя**[7].

Практично всі дослідження базуються на даних, отриманих звибірок, а не на даних генеральної сукупності з практичних міркувань (обмеження часу та ресурсів). Винятками є дослідження на кшталт U.S.Census [8], головна мета яких охопити всіх жителів США у визначеному році.

Репрезентативнавибірка – це така вибірка, що представляє генеральну сукупність. ЇЇ можна використовувати для вивідної статистики.

Нерепрезентативна вибірка – це коли вибіркатагенеральна сукупність мають різні характеристики. Використання цієї вибірки призведе до

неправильних результатів аналізу.

Розрізняють ймовірнісні вибірки та вибірки сформовані згідно певних правил. При використанні останніх є високий ризик отримати не репрезентативну вибірку. Прикладом є "волонтерська вибірка" – коли опитують лише бажаючих взяти участь в дослідженні чи "зручна вибірка" (convinience sampling), яка формується з доступних для дослідження об'єктів. Опитування студентів університету про їх політичні уподобання (якщо в якості генеральної сукупності виступає все доросле населення України) буде водночас і волонтерською, і зручною вибіркою.

Якщо потрібно сформувати репрезентативну вибірку – то **ймовірнісні вибірки** є найкращим вибором.

Найпоширенішим видом ймовірнісної вибірки є **простий випадковий вибір** (simple random sampling). Всі об'єкти генеральної сукупності мають однакову можливість бути вибраними. Також виділяють:

- стратометричний вибір – сукупність ділиться на страти (наприклад, населення за рівнем освіти чи віковою групою);
- кластерний вибір – сукупність ділиться на кластери, які однакові за своєю структурою, потім випадковим чином обираються кластери, а тоді елементи цих кластерів;
- систематичний вибір – елементи сукупності впорядковуються і вибирається кожен k-ий елемент (елементи на конвейері з метою виявлення дефектів).

12.3. Центральна гранична теорема

Наприклад, ми досліджуємо зріст дорослих чоловіків, які проживають на території Європи. В нашому дослідженні вони представляютьгенеральну сукупність з параметрами μ та σ. Ми можемо сформувати вибірки для оцінки середнього значення зросту дорослого чоловіка. Наприклад, визначити середнє значення зросту дорослого чоловіка та середньоквадратичне відхилення для кожної країни Європи. Чи сформувати 1000 вибірок, випадковим чином

обравши 40 чоловіків для кожної. Середнє значення кожної вибірки буде **точковою оцінкою** для середнього значення генеральної сукупності.

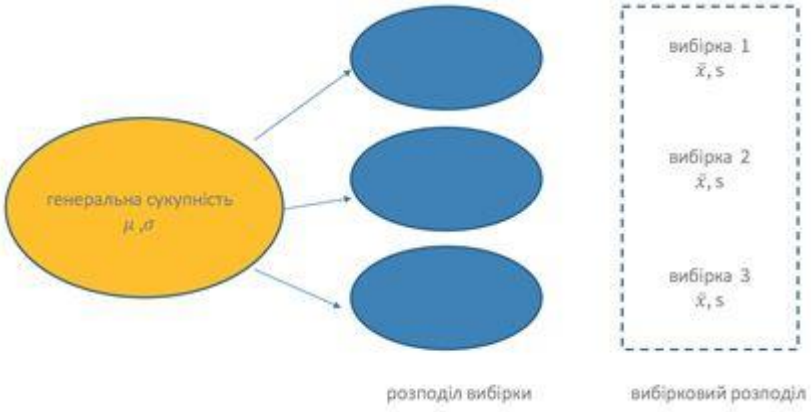


Рис. 12.1 Вибірковий розподіл

Отже, маємо три розподіли:

- **розподіл генеральної сукупності** (середнє значення μ, середньоквадратичне відхилення σ, переважно невідомі);
- **розподіл вибірки** (середнє значення x, середньоквадратчине відхилення s, використовуються для оцінки параметрів генеральної сукупності);
- **вибірковий розподіл**– розподіл середніх значень вибірок (рис. 12.1).

Центральна гранична теорема. Незалежно від того, якийрозподіл має змінна у популяції (генеральній сукупності),вибірковий розподіл середніх значень вибірок має приблизно нормальний розподіл, якщо розмір вибірки n принаймі 30. Середнє значення вибіркового розподілу μ_x дорівнює середньому значенню генеральної сукупності μ, а середньоквадратичне відхилення буде обчислюватись за формулою $S = \frac{\sigma}{\sqrt{n}}$, де σ – середньоквадратичне відхилення генеральної сукупності, а n – розмір вибірки.

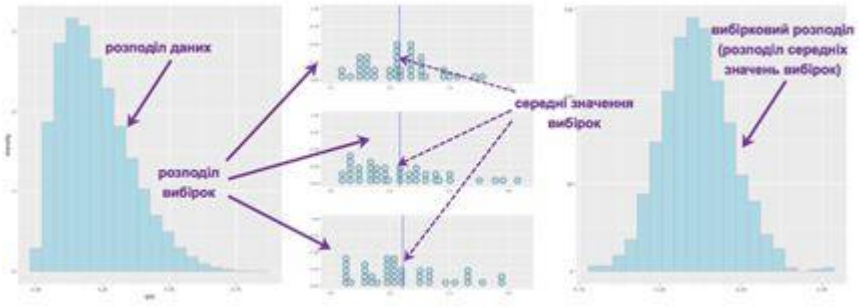


Рис. 12.2 Вибірковий розподіл

Вибірковий розподіл пропорцій вибірок (рис. 12.2) має приблизно нормальний розподіл за умови наявності принаймі 15 успіхів та 15 невдач, тобто np ≥ 15 та n(1-p) ≥ 15. При цьому середнє значення пропорцій вибірок μ_p дорівнює значенню пропорції генеральної сукупності p, а середньоквадратичне відхилення $S\left(\overline{p}\right)=\sqrt{\frac{p(1-p)}{n}}$.

Знання, що розподіл є нормальним, дозволяє оцінити ймовірність на основі Z-значень. Тобто, використовуючи центральну граничну теорему ми можемо оцінити ймовірність отримати вибірку з певним середнім значенням чи значенням пропорції.

12.4. Довірчий інтервал

Однією з основних задач математичної статистики є оцінка числових характеристик (параметрів) генеральної сукупності за вибірковими даними.

Для вибірки можна обчислити такі числові характеристики, як: вибіркове середнє, мода, медіана, вибіркова дисперсія та вибіркове середнє квадратичне відхилення. Для генеральної сукупності часто визначаються не самі ці параметри, а довірчі інтервали.

Точкова оцінка– це одне число, яке є нашою найкращою здогадкою про значення параметрів популяції. Однак, одна точкова оцінка не говорить, наскільки близькою ця оцінка є до справжньої пропорції генеральної сукупності. Тобто наступним кроком має бути оцінка точності цієї точкової оцінки.

Довірчим інтервалом(Інтервальна оцінка, confidenceinterval, CI**)** для певного параметру генеральної сукупності називається такий числовий інтервал, в межах якого знаходиться цей параметр. Ймовірність, з якою довірчий інтервал покриє істинне значення параметра, називається **довірчою ймовірністю** або **рівнем надійності** (рівнем довіри, confidencelevel)і позначається α. Тобто довірчий інтервал – це інтервал значень, який з високою ймовірністю містить справжні параметри генеральної сукупності. Ймовірність, що цей інтервал містить середнє значення чи значення пропорції генеральної сукупності визначається рівнем довіри. Рівень довіри має значення близьке до одиниці (найбільш поширені значення 90%, 95%, 99%).Для довірчого інтервалу можемо записати ніступні вирази

Довірчий інтервал = точковаоцінка(estimate)±межапохибки(marginoferror)

або

Довірчий інтервал = точковаоцінка(estimate)±критичнезначення×середньоквадратичневідхиленняточковоїоцінки

Точкова оцінка – значення середнього чи пропорції для вибірки.

Критичне значення – це значення для $\frac{1-\alpha}{2}$, де α – рівень довіри.Наприклад, для рівня довіри α = 95 % критичне значення z розподілу становить 1.96(тобто 95% лежать в межах ±1.96) середньоквадратчиних відхилень.

Довірчі інтервали розраховуються з урахуванням певних вимог догенеральної сукупності. Зазвичай це вимога нормального розподілу її даних.

Використовувати властивості нормального розподілу для побудови довірчого інтервалу нам дозволяє центральна гранична теорема, яка говорить, що неважливо який розподіл даних у генеральній сукупності, вибірковий розподіл (тобто розподіл середніх значень вибірок) буде мати нормальний розподіл.

Давайте поглянемо на формули для визначення середньоквадратичного відхилення вибіркового розподілу $S = \frac{\sigma}{\sqrt{n}}$ для середнього значення та $S\left(\overline{p}\right) = \sqrt{\frac{p(1-p)}{n}}$ для пропорції.В обох формулах у знаменнику маємо n – розмір вибірки. Відповідно, при збільшенні n, значення середньоквадратичного відхилення (а отже і межі похибки) будуть зменшуватись. Тобто, ми можемо визначити таке n при якому межа похибки набуває визначеного значення.

12.5. Довірчий інтервал для середнього значення

Давайте розглянемо процес побудови довірчого інтервалу для оцінки середнього значення генеральної сукупності на основі даних вибірки

Таблиця 12.2

Значення $z_{\frac{1-\alpha}{2}}$

α	0,4	0,25	0,2	0,15	0,1	0,05	0,025	0,01	0,005	0,001
$z_{\frac{1-\alpha}{2}}$	0,253	0,675	0,842	1,036	1,282	1,645	1,960	2,326	2,576	3,090

Наприклад,X – генеральна сукупність, що підкоряється нормальному законурозподілу; σ^2 –відома генеральна дисперсія. $\{x_1, x_2, \ldots, x_n\}$ –вибірка згенеральної сукупності об’ємом n; $\overline{x}$ –вибіркове середнє.Тоді довірчий інтервал для генерального середнього μ із заданим рівнем надійності α знаходиться за формулою

$$\overline{x} - z_{\frac{1-\alpha}{2}} \frac{\sigma}{\sqrt{n}} \le \mu \le \overline{x} + z_{\frac{1-\alpha}{2}} \frac{\sigma}{\sqrt{n}},$$

де $z_{\frac{1-\alpha}{2}}$ знаходиться з табл. 12.2.

Однак, в реальному житті (як і в прикладі оцінки радуісу Марса), нам зазвичай **невідоме значення** σ.У цьому випадку для оцінки

середньоквадратичного відхилення вибіркового розподілу використовують **середньквадратичне значення вибірки s**, однак в якості теоретичного розподілу середніх значень використовують **t-розподіл** (розподіл Стьюдента)

$$\bar{x} - t_{\frac{1-\alpha}{2},n-1}\frac{s}{\sqrt{n-1}} \le \mu \le \bar{x} + t_{\frac{1-\alpha}{2},n-1}\frac{s}{\sqrt{n-1}},$$

де $t_{\frac{1-\alpha}{2},n-1}$ знаходиться з таблиці розподілу Стьюдента, яка є устатистичних довідниках, або за допомогоюR функціїqts параметрами $\frac{1-\alpha}{2}$ та кількістю ступенів вільності df.

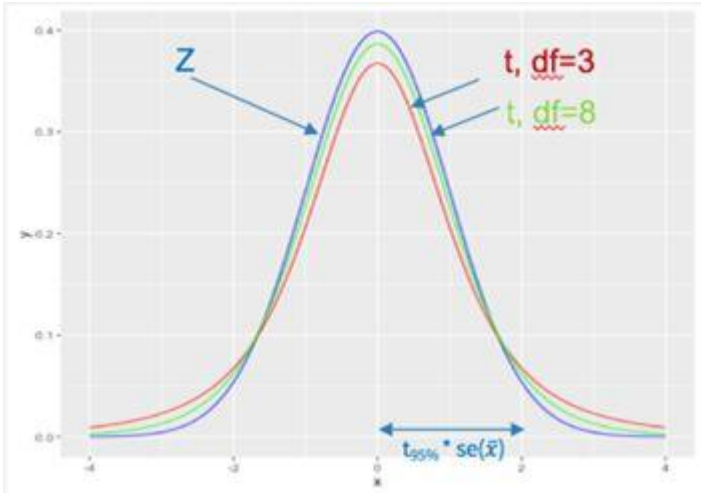


Рис. 12.3 t-розподілСтюдента

Євеликакількість t-розподілів (рис. 12.3), взалежностівідкількостіступеніввільності (degrees of freedom), формацихрозподіліввідрізняється (аотжеіплощапідкривою, іймовірностіматизначеннябільшечименшезаданого).

Що таке ступені вільності? Розглянемо це на прикладі однієї змінної. Наприклад, у нас є 9 клітинок та 9 чисел. Якщо для перших 8 є можливість вибору, в яку клітинку їх розмістити, то останнє число буде в останній клітинці, яка залишиться (без можливості вибору). Тобто тут кількість ступенів вільності становить **n – 1**.

Отже, Покроковий план побудови довірчого інтервалу може мати наступний вигляд

- Визначити рівень довіри (поширені значення 90%, 95%, 99%);

- Визначити середнє значення чи пропорція.
  - — Для пропорції:
    - ✓ знаходимо критичне значення z-розподілу для заданого рівня довіри;
    - ✓ знаходимо межу похибки(margin of error) $z_\alpha \times \sqrt{\frac{p(1-p)}{n}}$.
  - — Для середнього значення:
    - ✓ Якщо відоме значення середньоквадратичного відхилення генеральної сукупності:
      - ➢ знаходимо критичне значення z-розподілу для заданого рівня довіри;
      - ➢ знаходимо межу похибки(margin of error) $z_\alpha \times \frac{\sigma}{\sqrt{n}}$.
    - ✓ Якщозначеннясередньоквадратичного відхилення генеральної сукупності невідоме:
      - ➢ знаходимо кількість ступенів вільності за формулою df=n-1, де n - розмір вибірки;
      - ➢ знаходимо критичне значення t-розподілу відповідно до кількості ступенів вільності;
      - ➢ знаходимо межу похибки(margin of error) $z_\alpha \times \frac{s}{\sqrt{n}}$.
- Визначаємо межі інтервалу за формулою точковаоцінка(estimate) ± межапохибки(margin of error).

Лекція 13.

ПЕРЕВІРКА СТАТИСТИЧНИХ ГІПОТЕЗ

План

13.1. Вступ

13.2. Поняття про статистичні гіпотези

13.3. Перевірка гіпотези про вид закону розподілу досліджуваної величини

13.4. Перевірка гіпотези про рівність генеральних дисперсій. F-критерій (Фішера)

13.5. Перевірка гіпотези про рівність генеральних дисперсій. Критерій Зігеля-Тьюкі

13.6. Перевірка гіпотези про рівність генеральних середніх. Критерій Стьюдента

13.1. Вступ

Коли дослідники мають очікування щодо параметрів генеральної сукупності – говорять про статистичну гіпотезу. Зазвичай, гіпотеза формулюється як твердження що параметр генеральної сукупності має певне значення або знаходиться в певному інтервалі. Це твердження базується на попередніх дослідженнях та теорії. На основі інформації, отриманої з вибірки оцінюють чи має сенс (справедливе) це твердження чи ні. Це те, що ми називаємо **тест на значущість**. Тест на значущість, як і побудова довірчих інтервалів, є методом вивідної статистики. Ми пробуємо оцінити параметри генеральної сукупності на основі вибірки.

13.2. Поняття про статистичні гіпотези

Статистичні гіпотези позначаються латинськими буквами H_0, H_1, і т. д. Гіпотеза H_0 формулюється як основна в тому розумінні, що при перевірці бажано було б встановити її справедливість. Основній гіпотезіпротиставляються інші гіпотези H_1, H_2, …, які називаються

альтернативними.

Прийняття основної або однієї з альтернативних гіпотез здійснюється на основі дослідження статистичних даних. Дослідження проводиться за певним **критерієм**, який обирається відповідно до змісту гіпотези і виду наявних статистичних даних.

Якщо сформульовані гіпотези H_0 –основна та H_1 –альтернативна (конкуруюча) і обраний критерій перевірки справедливості основної гіпотези, то прийняття H_0 означає відкидання H_1, а відкидання H_0 означає справедливість H_1.Нульова гіпотеза стверджує, що параметр генеральної сукупності набирає конкретного значення. Ця гіпотеза може бути відхилена, якщо дані вибірки кажуть що це дуже нетипові очікування. Альтернативна гіпотеза стверджує, що параметр, який досліджується має альтернативне значення чи набір значень. Нульова та альтернативна гіпотези завжди взаємовиключні (mutually exclusive). Коли ви робите тест на значимість, то вважаєте, що нульова гіпотеза правдива поки дані вибірки не дадуть достатньо сильні аргументи, що це не так.

Оскільки прийняття гіпотези здійснюється на основі статистичних даних, то завжди існує ймовірність помилки.

Ймовірність відкидання гіпотези H_0, якщо вона справедлива, називається ймовірністю помилки першого роду або **рівнем значущості**і позначається α .Величина $1-\alpha$ є ймовірністю прийняття справедливої гіпотези і називається **рівнем довіри**. Ймовірність прийняття гіпотези H_0, якщо вона не вірна, називається ймовірністю помилки другого роду і позначається β. Величина $1-\beta$ є ймовірністю відкидання невірної гіпотези і називається **потужністю критерію**.

Чим менше значення рівня значущості, тим менша ймовірність відкинути вірну гіпотезу. Зазвичай рівень значущості обирається дослідником рівним 0,1; 0,05; 0,01 або 0,001. Якщо, наприклад, обраний рівень значущості $\alpha = 0,01$, то ризик відкинути вірну гіпотезу виникає в одному випадку із ста.

Перевірка статистичної гіпотези не надає точного висновку щодо її

вірності або невірності. Прийняття гіпотези означає, що на прийнятому рівні значущості вона не суперечить статистичним даним.

13.3. Перевірка гіпотези про вид закону розподілу досліджуваної величини

Перевірка гіпотези про вид закону розподілу досліджуваної величини маєвелике значення для прикладних досліджень. Необхідність такої перевіркивиникає при виборі критерію, оскільки для багатьох з них висувається вимоганормального розподілу статистичних даних. Означені гіпотези перевіряютьсяпри проектуванні систем масового обслуговування, перевірки якості продукціїабо праці і т. ін.

Таблиця 13.1

Статистичний ряд

x_i	x_1	x_2	…	x_k
n_i	n_1	n_2	…	n_k

Припустимо, що з деякої генеральної сукупності X, яка розглядається яквипадкова величина, обрана вибірка $\{x_1, x_2, ..., x_n\}$. За даними вибірки побудовано статистичний ряд (табл. 13.1), що містить варіанти x_i та відповіднічастоти n_i, $i = \overline{1,k}$, k – кількість варіант у випадку дискретного ряду. У випадку інтервального ряду x_i – середини інтервалів, k – кількість інтервалів.

Отриманий на основі вибіркових даних статистичний ряд називається**емпіричним законом розподілу** величини X.

За даними статистичного ряду можна знайти числові характеристики, які є вибірковими параметрами закону розподілу X. Вид закону розподілу визначається відповідно до умов формування вибірки або залежно від виду графіка емпіричної щільності розподілу (гістограми) у випадку неперервної випадкової величини X і полігону частот, якщо величина X дискретна. Параметри обраного закону розподілу змінюються відповідними вибірковими параметрами.

Закон розподілу випадкової величини X, параметрами якого є відповідні

вибіркові числові характеристики, називається **теоретичним законом розподілу**.

При здійсненні такої заміни немає впевненості, що закон розподілу обраний правильно. Тому розроблено процедуру, яка дозволяє оцінити степінь відповідності обраного закону даним вибірки. Критерії здійснення такої перевірки називаються **критеріями згоди**, найбільш відомим з яких є **критерій Пірсона** χ^2 (хі-квадрат).

Критерій Пірсона χ^2 обчислюється за формулою

$$\chi^2 = \sum_{i=1}^{k} \frac{\left(n_i - n_i^{'}\right)^2}{n_i^{'}}, \qquad (13.1)$$

де $n_i^{'}$ – частоти, отримані за теоретичним законом розподілу (теоретичні).

З формули (13.1) видно, що у випадку, коли відповідні теоретичні та емпіричні частоти співпадають, $\chi^2 = 0$. Тобто, чим ближче χ^2 до нуля, тим краще узгоджуються вибіркові дані та обраний теоретичний закон розподілу.

Розраховане значення критерія χ^2 порівнюється з його критичним значенням $\chi^2_{\alpha,l}$, яке знаходиться за статистичними таблицями, або за допомогою стандартної R-функції qchisq. Параметрами функції qchisq є: α – рівень значущості; l – степінь свободи, $l = k - r - 1$, де k – кількість груп емпіричного розподілу, r – кількість параметрів теоретичного розподілу (наприклад, для нормального розподілу r= 2, оскільки параметрів два– $\bar{x}$ і σ).Якщо $\chi^2 < \chi^2_{\alpha,l}$, то гіпотеза про закон розподілу приймається. У противному випадку гіпотеза відкидається.

Зазначимо, щонемаєнеобхідностівиконуватиобчисленнявідповіднодоформули (13.1) вручну, оскількивRдляцьогоєстандартнафункціяchisq.test(). При роботі із цією функцією дані оформляються у виді матриці.

13.4. Перевірка гіпотези про рівність генеральних дисперсій. F-критерій (Фішера)

В прикладних задачах часто виникає необхідність перевірки рівності середніх значень та дисперсій за даними двох або більше вибірок. Наприклад, коли визначається перевага однієї з технологій виготовлення певної продукції, або наявність підвищення продуктивності праці після внесення змін в процес виробництва, або при перевірці якості продукції. Здійснення означеної перевірки виконується за критеріями, що обираються залежно від виду розподілу вибіркових даних і мети дослідження. Для деяких критеріїв перевірки рівності середніх значень висувається додаткова вимога − про рівність генеральних дисперсій.

Перевіркагіпотезипрорівністьгенеральнихдисперсійздійснюєтьсяза*F*-критерієм (Фішера) тількитоді, колистатистичніданінезалежніірозподіленізанормальнимзаконом.

Формулюютьсягіпотези:

H_0–дисперсії двох нормально розподілених генеральних сукупностей рівні, тобто $\sigma_1^2 = \sigma_2^2$;

H_1–дисперсії двох нормально розподілених генеральних сукупностей не рівні, тобто $\sigma_1^2 \neq \sigma_2^2$.

Генеральні дисперсії оцінюються на основі вибірок, і сам*F*-критерій (Фішера) обчислюється як відношення більшої вибіркової дисперсії до меншої за формулою

$$F = \frac{s_1^2}{s_2^2},\ s_1^2 > s_2^2.$$

Гіпотеза H_0 приймається, якщо розраховане значення *F* менше критичного значення розподілу Фішера $F_{êðèò}$, взятого із рівнем значущості α і степенями свободи l_1 та l_2 для чисельника і знаменника відповідно для чисельника і знаменника відповідно: $l_1 = n_1 - 1,\ l_2 = n_2 - 1,$ де $n_1,\ n_2$ – об'єми вибірок. $F_{êðèò}$.

Для обчислення теста Фішера у R використовується функція var.test() (від variance – дисперсия, и test – тест)[9].

Очевидно, що чим більше обчислене значення F наближається до 1, тим більше у нас підстав прийняти гіпотезу H_0. І навпаки – чим більше значення F, тим більше підстав відкинути гіпотезу H_0 про рівність дисперсій.

13.5. Перевірка гіпотези про рівність генеральних дисперсій. Критерій Зігеля-Тьюкі

Якщо статистичні дані не розподілені за нормальним законом або вимірюються з використанням порядкової шкали, то перевірка гіпотези про рівність генеральних дисперсій здійснюється за критерієм Зігеля-Тьюкі. Формулюються гіпотези:

H_0 –дисперсії двох генеральних сукупностей рівні, тобто $\sigma_1^2 = \sigma_2^2$;

H_1 –дисперсії двох генеральних сукупностей не рівні, тобто $\sigma_1^2 \neq \sigma_2^2$.

Перевірка виконується за даними двох вибірок у такій послідовності:

1) Формується об’єднана вибірка;

2) Даним об’єднаної вибірки присвоюються ранги (порядкові номери) за правилом: найменшому значенню присвоюється ранг 1, двом найбільшим – ранги 2 і 3; наступним двом найменшим – ранги 4 і 5; наступним найбільшим – ранги 6 і 7 і т. д. При цьому, якщо кількість елементів вибірки непарна, то її центральний елемент (тобто медіана) не отримує ніякого рангу;

3) Обчислюєтьсясуми рангів елементів вихідних вибірок R_1 і R_2;

4) Обчислюється нормальна випадкова величина Z за формулою

$$Z = \frac{2R_1 - n_1(n_1 + n_2 + 1) + 1}{\frac{n_2}{3}\sqrt{n_1(n_1 + n_2 + 1)}},$$

де n_1, n_2 – об’єми вибірок. При цьому R_1 сума рангів меншої за об’ємом

вибірки. Якщо $2R_1 > n_1(n_1 + n_2 + 1) + 1$, то Zобчислюється за формулою

$$Z = \frac{2R_1 - n_1(n_1 + n_2 + 1) - 1}{\frac{n_2}{3}\sqrt{n_1(n_1 + n_2 + 1)}}$$

5) У випадку, коли перевіряються вибірки різних об'ємів, обчислюється скоректована нормальна випадкова величина Z' за формулою

$$Z' = Z\left(\frac{1}{10n_1} - \frac{1}{10n_2}\right)\left(Z^3 - 3Z\right);$$

6) Обирається рівень значущості α;

7) За допомогою таблиці значень функції нормального розподілу або R-функції dnorm знаходиться ймовірність $P(Z)$ або $P\left(Z'\right)$;

8) Порівнюються рівень значущості α і величина$P(Z)$ або $P\left(Z'\right)$.

Якщо $2P\left(Z\right) > \alpha$ (або $2P\left(Z'\right) > \alpha$), то гіпотеза H_0 про рівність генеральних дисперсій приймається.

Для перевірки правильності присвоєння рангів можнаскористатися формулою $R_1 + R_2 = \frac{(n_1 + n_1)(n_1 + n_1 + 1)}{2}$ у випадку парноїкількості елементів об'єднаної вибіркиабо $R_1 + R_2 = (n_1 + n_1)\left(\frac{n_1 + n_1 + 1}{2} - 1\right)$ у випадку непарної кількості цих елементів.

13.6. Перевірка гіпотези про рівність генеральних середніх. Критерій Стьюдента

Критерій Стьюдента використовується для перевірки гіпотез про рівність генеральних середніх, якщо статистичні дані розподілені за нормальним законом. Формулюються гіпотези:

H_0 –середні двох генеральних сукупностей рівні, тобто $\mu_1 = \mu_2$;

H_1 –середні двох генеральних сукупностей не рівні, тобто $\mu_1 \neq \mu_2$.

Перевірка виконується за даними двох вибірок об'ємом n_1 та n_2. При цьому можливі такі випадки.

Випадок 1. Генеральні дисперсії рівні ($\sigma_1^2 = \sigma_2^2$).Тоді *t*-критерій Стьюдента обчислюється за формулою

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{n_1 S_1^2 + n_2 S_2^2}{n_1 + n_2 - 2}\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}.$$

Розраховане значення *t*-критерія порівнюється з критичним значенням $t_{êðèò}$, де $t_{êðèò}$ – критичне значення розподілу Стьюдента з параметрами $\frac{\alpha}{2}$ і степенем свободи $l = n_1 + n_2 - 2$, яке надається в статистичних таблицях.УR тест Стьюдента виконується за допомогою функції t.test().

Випадок 2. Генеральні дисперсії не рівні($\sigma_1^2 \neq \sigma_2^2$).Тоді *t*-критерій Стьюдента обчислюється за формулою

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)}}.$$

Розраховане значення *t*-критерію також порівнюється з критичним значенням $t_{êðèò}$, але степінь свободи розраховується за формулою

$$l = \frac{\left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)^2}{\frac{\left(\frac{S_1^2}{n_1}\right)^2}{n_1 + 1} + \frac{\left(\frac{S_2^2}{n_2}\right)^2}{n_2 + 1}}.$$

Випадок 3. Вибірки не є незалежними, оскільки на них впливає певний фактор і його вплив невідомий, або вибірки є даними, отриманими до і після проведення певного експерименту. Тоді формується парна вибірка і для кожної пари елементів знаходиться *d* – різниця їх значень. Подальша перевірка здійснюється над вибіркою різниць. *t*-критерій Стьюдента обчислюється за

формулою

$$t = \frac{\overline{x_d}}{S_d / \sqrt{n-1}},$$

де $\overline{x_d}$ –вибіркове середнє для вибірки різниць, S_d –вибіркове середнє квадратичне відхилення для вибірки різниць, n – об'єм вибірки різниць.

Розраховане значення t-критерію також порівнюється з критичнимзначенням розподілу Стьюдента з параметрами $\frac{\alpha}{2}$ і степенем свободи $l = n - 1$.

У всіх випадках гіпотеза H_0 приймається, якщо розраховане значенняt-критерія менше критичного значення $t_{êðèò}$ за абсолютною величиною, тобто

$$|t| < t_{êðèò}.$$

Лекція 14.

ОСНОВИ ВІЗУАЛІЗАЦІЇ ДАНИХ

План

14.1. Вступ

14.2. Мова візуалізації

14.3. Табличні дані і графіки

14.1. Вступ

Візуалізація – це представлення інформації, даних, фактів у візуальній формі. Водночас, візуалізація є мовою, в якій використовуються геометричні об'єкти – точка, лінія, частина поверхні, а також візуальні канали – колір, довжина, орієнтація, розмір. Фактично, мова візуалізації – це продовження звичайної мови, тому що тексти – її частина [10].

Одночасно, як і будь яка мова, її базові елементи можна комбінувати багатьма способами. Проте, не всі комбінації мають сенс. До того ж, різні типи даних вимагають різних способів їх представлення мовою візуалізації –для них потрібно використовувати різні способи візуального кодування.

З одного боку, може здатися, що це ускладнює задачу інформаційного дизайнера. Насправді, якщо знати мову візуалізації та правила, у який спосіб краще представляти ті чи інші дані, це сильно полегшує роботу – тому що обмежує кількість можливих варіантів.

Отже, саме цим ми зараз і займемося: подивимося, які типи даних існують, і як їх кодувати за домомогою цієї мови у найбільш ефективний спосіб.

14.2. Мова візуалізації

Ми використаємо найпростішу схему класифікації. За нею, дані поділяються на три типи:

- кількісні (quantitative) – все, що можна порахувати та записати у числовій формі;

•впорядковані (ordered) – якісні дані, те, що можна розташувати у якомусь порядку – дні тижня, градації шкали оцінювання (наприклад, від "дуже погано" до "дуже добре");

•категорійні (categorical) – невпорядковані якісні дані. Практично все, що не відноситься до перших двох типів – назви країн, назви з будь яких наборів, різноманітні типи, тощо.

Елементами мови візуалізації є мітки та візуальні канали.

Мітки– це базові графічні елементи (найпростіші геометричні об'єкти):

•точка;

•лінія;

•площина (на 2D поверхні);

•об'ємне тіло (в 3D).

Канали– це спосіб, у який ми можемо показати наші позначки. Тобто, ми можемо контролювати як буде виглядати позначка, за допомогою таких візуальних каналів, як:

•позиція;

•розмір;

•форма;

•орієнтація;

•відтінок, насиченість, яскравість (кольору).

Отже, для візуалізації ваших даних, по-перше, необхідно

✓ порахувати кількість змінних (наприклад, скільки колонок є у вашій таблиці з даними);

✓ визначити для кожної із цих змінних, до якого типу даних вона відноситься: до кількісних, впорядкованих чи категорійних.

Після цього, для кожної змінної ми можемо вибрати мітку та візуальний канал, який найкраще для неї підійде.

Приклад 14.1.Вважаємо, що задано

Країна	Показник

Парагвай	3
Зімбабве	5
Мексика	6

Змінна	Тип даних	Візуальні канали
Країна	категорійні	Орієнтація, Колір, Форма, Текстура, X (або Y)
Показник	кількісні	Орієнтація, Розмір, Значення, X (або Y)

Можливі комбінації:

1) Форма (різні категорії) + Значення (намалюємо число). Картинка, що вийшла виглядає як головоломка, а не як гарна інфографіка.

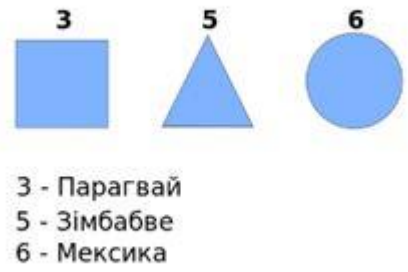


Рис. 14.1Форма (різні категорії) + Значення (число)

2) Колір (різні країни) + Розмір кола (показники). Виглядає трохи краще, але потрібно вказувати легенду – до якого кольору відноситься яка країна.

Рис. 14.2Колір (різні країни) + Розмір кола (показники)

3) Значення по Y (показник) + по X (країна). Майже ідеально, бо немає легенди. Підписи на графіку значно краще легенди.

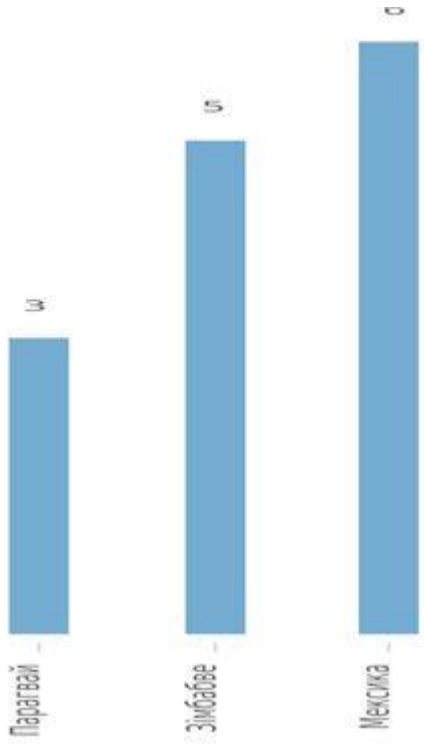


Рис. 14.3Значення по Y (показник) + по X (країна)

Приклад 14.2.Вважаємо, що задано

Тип задачі	Приоритет	Кількість зусиль, бали	Час, дні
Характеристика	Обов'язково	30	40
Характеристика	Добре мати	20	40

Характеристика	Непогано мати	15	20
Помилка	Виправити негайно	2	2
Помилка	Виправити колись	2	8
Помилка	Виправити вчасно	5	12
Побажання користувача	Обов'язково	8	10
Побажання користувача	Непогано мати	5	7
Побажання користувача	Добре мати	8	7

Візуально ці дані можна зобразити наступним чином (рис. 14.4)

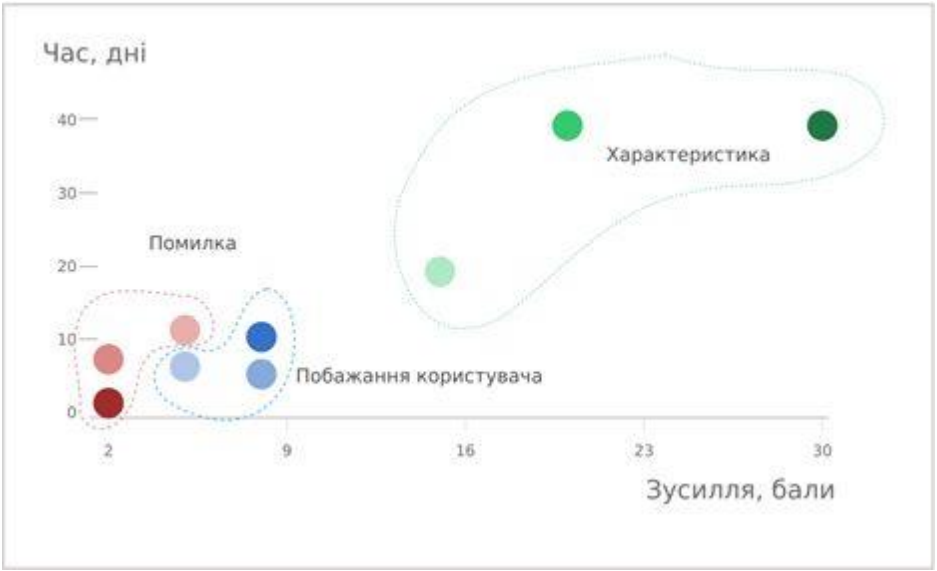


Рис. 14.4 Візуалізовані дані

Якщо використовувати візуальні канали, які згадувались, то виходить більше 500 варіантів комбінацій. Виберемо найбільш прийнятні у цьому випадку

Змінна	Тип даних	Візуальний канал
Тип задачі	Категорійні	відтінок кольору
Приоритет	Впорядковані	насиченість кольору
Кількість зусиль	Кількісні	X
Час	Кількісні	Y

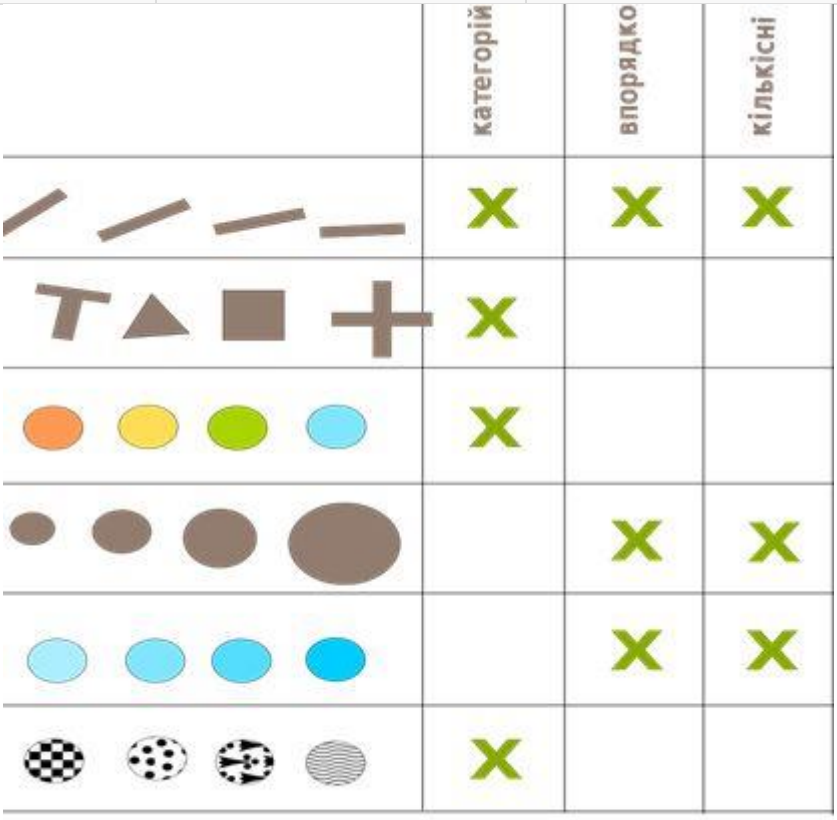


Рис. 14.5найбільш прийнятнівізуальні канали

Очевидно, що не всі візуальні канали можна використовувати для всіх типів даних (рис. 14.5). Для початку, давайте розіб'єм всі візуальні канали на дві частини

- ті які можна використовувати для кодування категорійних (у першому випадку);
- ті які можна використовувати для кодування кількісних (та впорядкованих) даних (у другому випадку);

Канали для відповіді на питання "Що?" (категорійні дані)

- форма;
- місце розташування;
- колір (відтінок).

Канали для відповіді "Наскільки багато?" (кількісні), або наскільки сильно? (впорядковані)

- довжина (1D);
- площа (2D);
- об'єм (3D);
- нахил;
- позиція;
- колір (яскравість).

Отже, рекомендації щодо перегляду візуалізації або інфографіки є наступними

- ✓ розглядайте їх з двох боків – дивіться, які дані використані, до якого типу даних відноситься кожна з змінних;
- ✓ порівняйте які канали візуалізації були задіяні для того, щоб візуально представити цей конкретний тип даних.

14.3. Табличні дані і графіки

Графіки є одним із головних компонент візуалізації.Завжди потрібно зберігати дані у файлі, для того, щоб у майбутньому можна було до них повернутися, доповнити або виправити, і швидко перебудувати графік. Якщо втрачаються початкові дані, то потрібно кожного разу будувати графік з нуля.

Отже, ми знайшли дані, записали їх у електронну таблицю. Записувати дані потрібно наступним чином – по горизонталі, в рядках записувати повний набір значень для всіх змінних, що є в наших даних (наприклад, назва, кількість населення, ВВП на душу, інфляція за останній рік, тощо для якоїсь конкретної країни). По вертикалі в колонках, йдуть значення кожної конкретної змінної із наших даних, наприклад інфляція для всіх країн.

Однак перед тим, як безпосередньо будувати графік, потрібно впевнитися, що дані мають коректну форму. Необхідно провести просту перевірку:

- кожна колонка повинна містити значення лише одної змінної з ваших даних;
- кількість колонок повинна бути фіксована і однакова для всього файлу (колонки/змінні не з'являються і не зникають, комірки не можна роздвоювати);
- в кожній колонці тип даних має бути однаковим (якщо числа - то всі числа, якщо текст - то весь час текст);
- формат для чисел повинен підходити під інструмент для побудови графіків - наприклад, для ChartBuilder потрібно використовувати точку у якості роздільного знака між цілою та дробною частиною, а не кому.

Наведемо декілька прикладів візуалізацій даних.

Приклад 14.3. Усейн Болт (рис. 14.6). Змінні: **роки** (Y), в які були встановлені світові рекорди на дистанцію 100 м., **колір** – тип медалей, **відстань** попередніх спортсменів від Усейн Болта (вісь X).

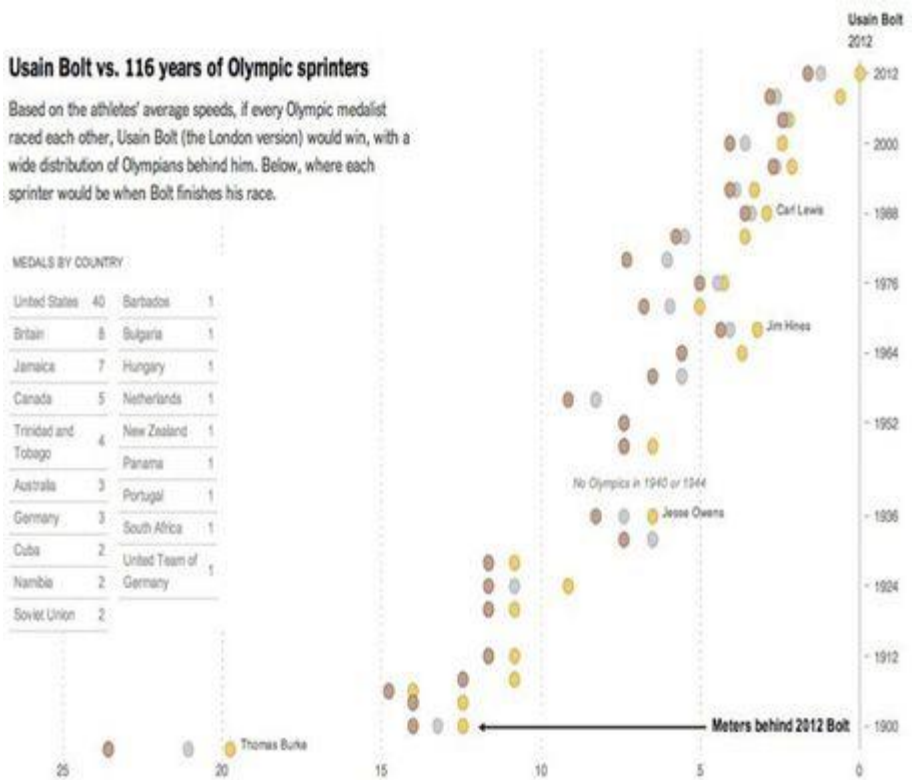


Рис. 14.6 Світові рекорди на дистанцію 100 м.

Приклад 14.4. Графік типу "чупа-чупс" (рис. 14.7). Змінні: Y – тип файлу (категорійні дані), X – кількість цих файлів (кількісні дані)

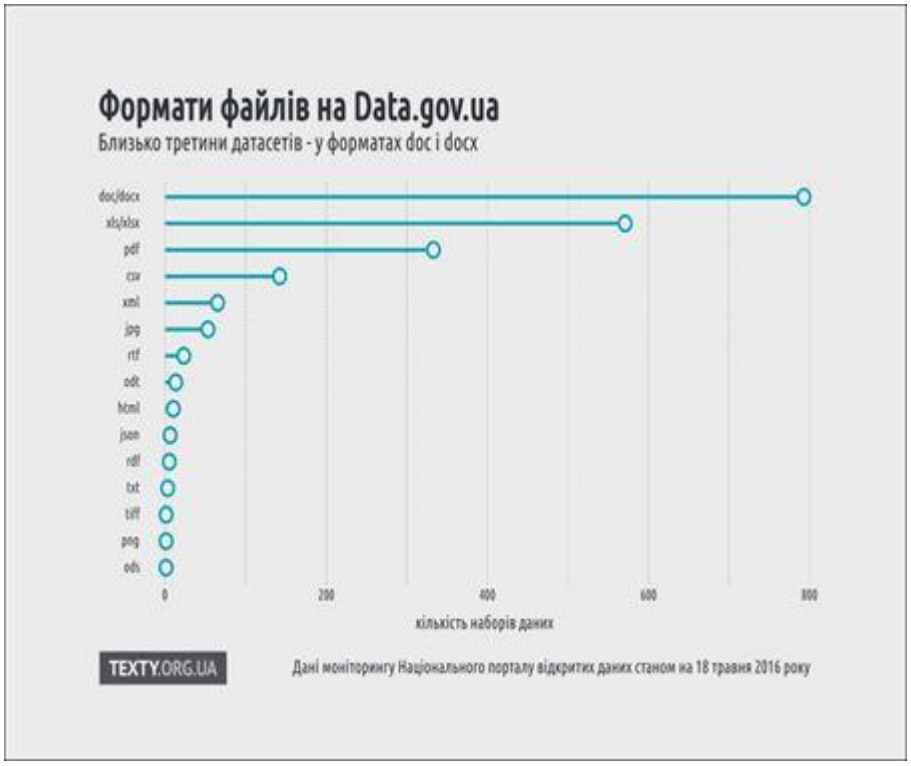


Рис. 14.7Графік типу "чупа-чупс"

Позиція є найбільш ефективним способом візуального кодування. Тому графік «Результати олімпіади» добре показує категорійні дані і кількісні показники, що з ним пов’язані (рис. 14.8).

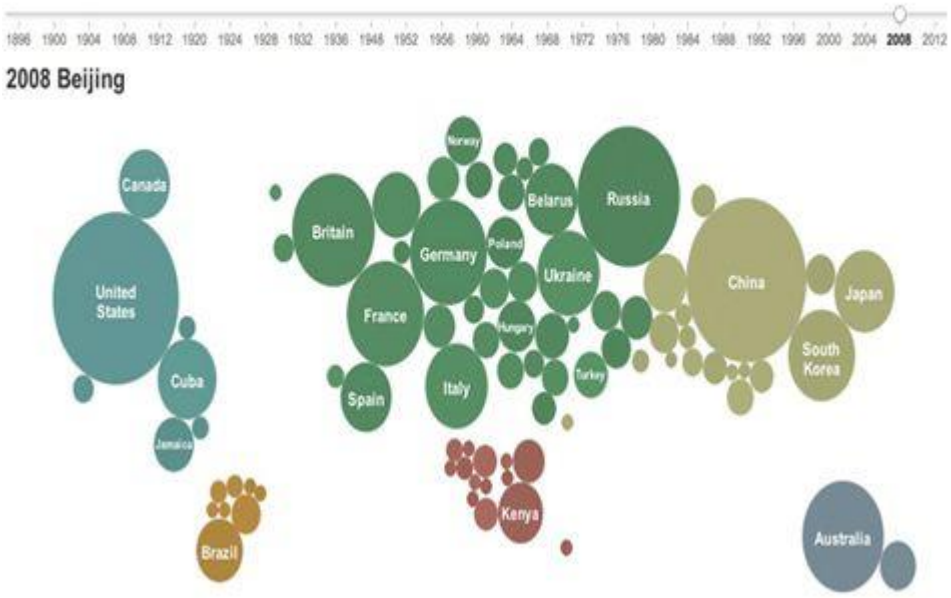


Рис. 14.8Результати олімпіади

Приклад 14.5.точковакартаNewYork, «stopandfrisk» (рис. 14.9).

Змінні: 1 та 2 - позиціїXтаY: зайнятігеографічнимикоординатами (кількіснідані), 3) кількістьобшуківтаоглядів, яківідбувалисьутомучиіншомурайоні – кількістьпоказаназадопомогоюрозмірукола, та 4 змінна – цеінтенсивністьобшуку (впорядкованідані) – показанаувидікольору.

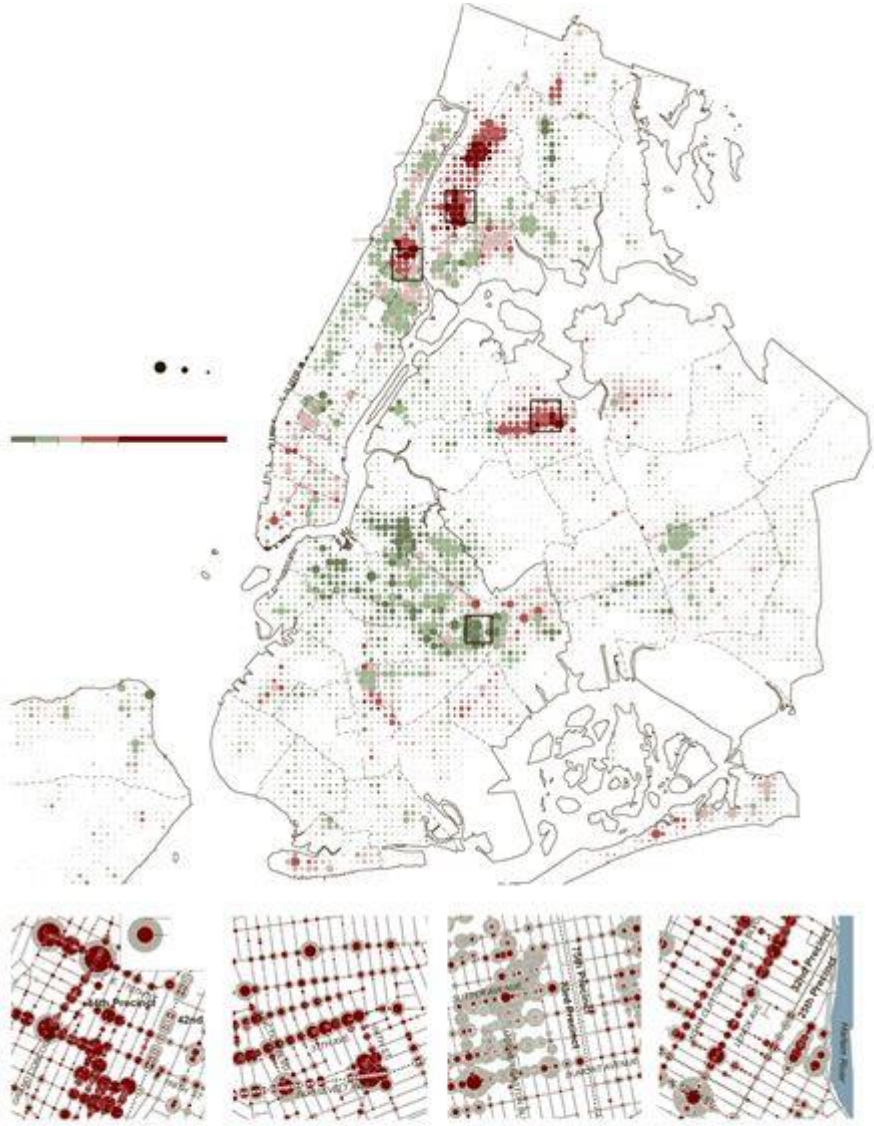


Рис. 14.9 New York, «stop and frisk»

Висновки. На реальних прикладах ми познайомилися з елементами мови візуалізації, з мітками (позначками), та візуальними каналами. Також ми дізнались про те, які типи даних існують, і які візуальні канали більше підходять для різних типів даних.

Лекція 15.

ІНФОДИЗАЙН: ГРАФІКИ ТА ЕФЕКТИВНІСТЬ ВІЗУАЛЬНОГО КОДУВАННЯ

План

15.1. Вступ

15.2. Типи графіків. Як вибрати вірний графік для різних задач

15.3. Ефективність візуального кодування

15.1. Вступ

Графіки дозволяють ефективно показувати різноманітні зв’язки, відношення між різними атрибутами (змінними) у наших даних. Вони надають характерну візуальну форму для кожного типу зв’язку. Корисно розуміти, які типи графіків можуть бути застосовані для різних типів зв’язків.

15.2. Типи графіків. Як вибрати вірний графік для різних задач

Є декілька таких графіків:

- Еволюціявчасі;
- Ранжування;
- Співвідношеннячасткиіцілого;
- Відхилення;
- Розподіл;
- Кореляція;
- Географічнідані;
- Номінальнепорівняння.

Для визначення, який саме тип нам потрібно показати (тобто який графік вибрати), потрібно пошукати в описі задачі задані ключові слова, за якими можна визначити тип зв’язку:

1. **Номінальне порівняння** – серія невпорядкованих дискретних кількісних значень. Це є найпростіший тип зв’язку.

Потрібно показати серію дискретних кількісних значень – кожна з яких відноситься до своєї категорії, щоб порівняти їх відносний розмір. Наші змінні – категорійна і кількісна, кодуємо їх як позицію.

Хорошими варіантами мають бути **стовпчикові графіки** (вертикальні або горизонтальні) або **точкові графіки**. Особливість – точкові графіки можуть починатися не від нульового значення, а стовпчикові – лище з нуля.

Рис 15.1 Точковий графік

2. Еволюція в часі

Ключові слова: тренд, зміна, зростання (падіння), збільшення (зменшення), підвищення (пониження), коливання (флуктуація).

Змінні – впорядкована (час) та кількісна. Кодуємо позицією. **Лінійний графік**– перший вибір. Також, **вертикальні стовпчики**– не горизонтальні, в яких час йде по вертикалі. Чому? Тому що показуємо значення впродовж (довжина, не висота!) якогось часу – сильна культурна традиція зліва направо. Такі графіки не показуємо по вертикалі. Точковий графік погано піходить, тому що точки гірше показують зв’язок між сусідніми часовими інтервалами.

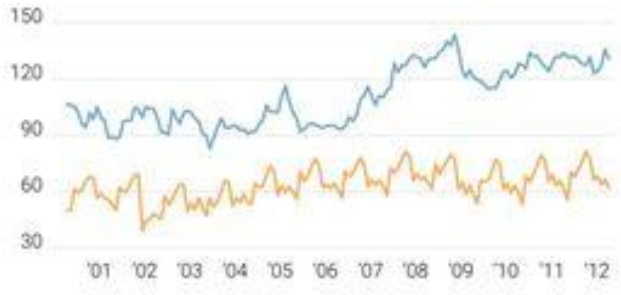


Рис 15.2 Еволюція в часі

3. Ранжування

Ключові слова: більше (менше) ніж, дорівнює.

Те ж саме, що номінальне порівняння, однак обов'язково використовуємо сортування! У порядку зменщення або навпаки – в залежності, що саме потрібно показати.

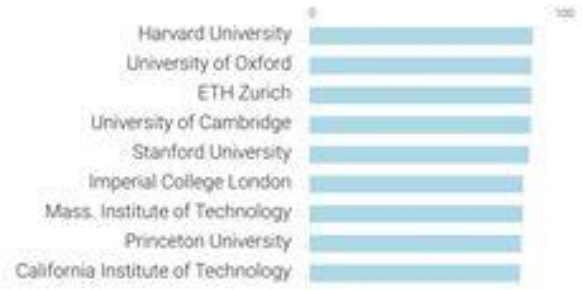


Рис 15.3 Ранжування

4. Співвідношення частки і цілого

Ключові слова: відношення, відсоток, частка.

Ми натреновані розуміти частку як відсоток. Наші змінні – це категорії (частки цілого) та їх внесок у ціле. Кодування може бути – колір для категорій та довжина для значень часток, виходить складена стовпчикова діаграма.

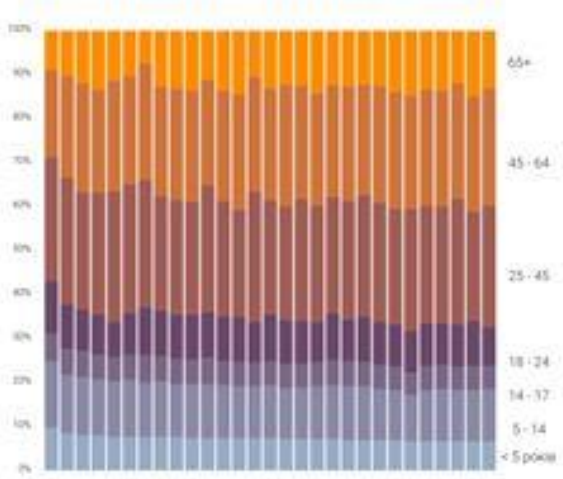


Рис 15.4 Співвідношення частки і цілого

Після того, як складені фрагменти будуть відсортовані, то кращим варіантом буде позиція-довжина. Зверніть увагу, що стовпчики поєднані і в заголовку точно вказано що це частки цілого.

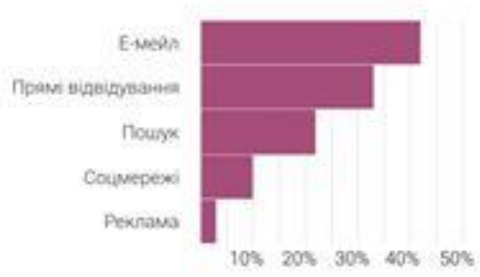


Рис 15.5 Співвідношення позиції і довжини

5. Відхилення

Ключові слова: плюс або мінус, варіація (відхилення), різниця, порівнюючи з.

Знову маємо кількість і категорію в даних та використовуємо позицію для кодування. Перший варіант – парні стовпчики, однак якщо цікаватільки

різниця, то потрібно її показувати і підкреслити кольором.

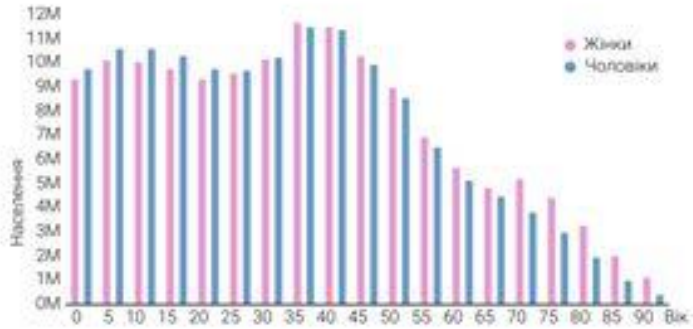


Рис 15.6 Різниця відхилення

Інший варіант – використати різницю у відсотках, для того щоб об’єктивно оцінити відхидення для різних категорій. Для відхилення у часі використовують лінійний графік з різницею між показниками.

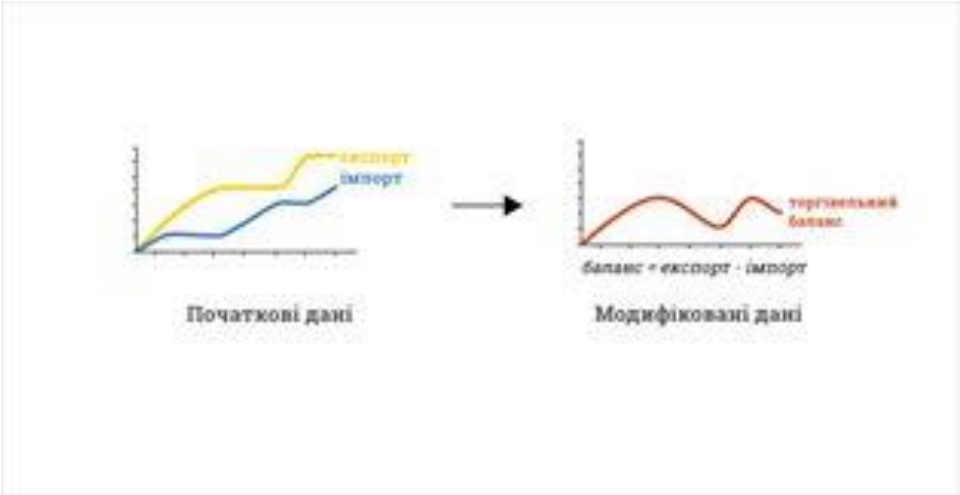


Рис 15.7 Відхилення за часом

6. Розподіл

Ключові слова: частота, розподіл, концентрація, нормальний розподіл (крива Гауса, крива Белла).

Графік розподілу показує, наскільки часто значення кількісної змінної зустрічаються вздовж всього діапазону своїх значень, від найменшого до набільшого. Зазвичай, весь цей діапазон розбивається на рівні інтервали (номер такого інтервалу– це змінна впорядкованого типу даних), і для кожного інтервалу визначпється скільки разів кількісна змінна потрапила в цей

інтервал.Для такої задачі найчастіше використовується –**гістограма**.

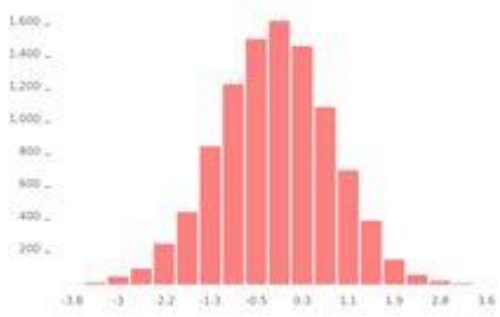


Рис 15.8 Гістограма розподілу

Кумулятивна частота – теж корисний графік, для того, щоб швидко рахувати внесок декількох діапазонів.

Ефективним способом показати всі значення є «точкова гістограма», в якій кожна точка є одним значенням змінної, що потрапила у відповідний інтервал.

Рис. 15.9 Точкова гістограма

Смужкова діаграма використовує прозорість, щоб показати скупчення в тому чи іншому інтервалі. Цей трюк з прозорістю часто використовують на точкових ‘x y’ - графіках та на картах.

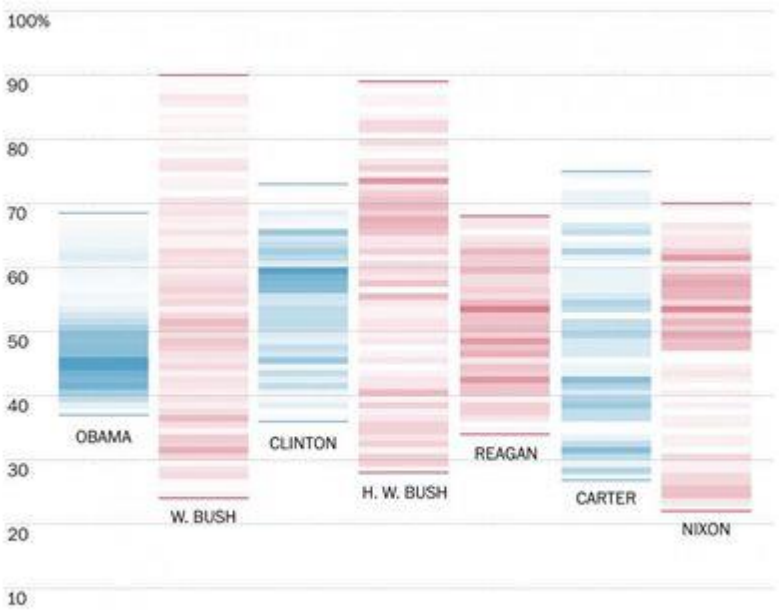


Рис. 15.10 Смужкова діаграма

Багато розподілів показують за допомогою найбільш популярного способу – **лінійного частотного полігона** або **бокс**-графіка.

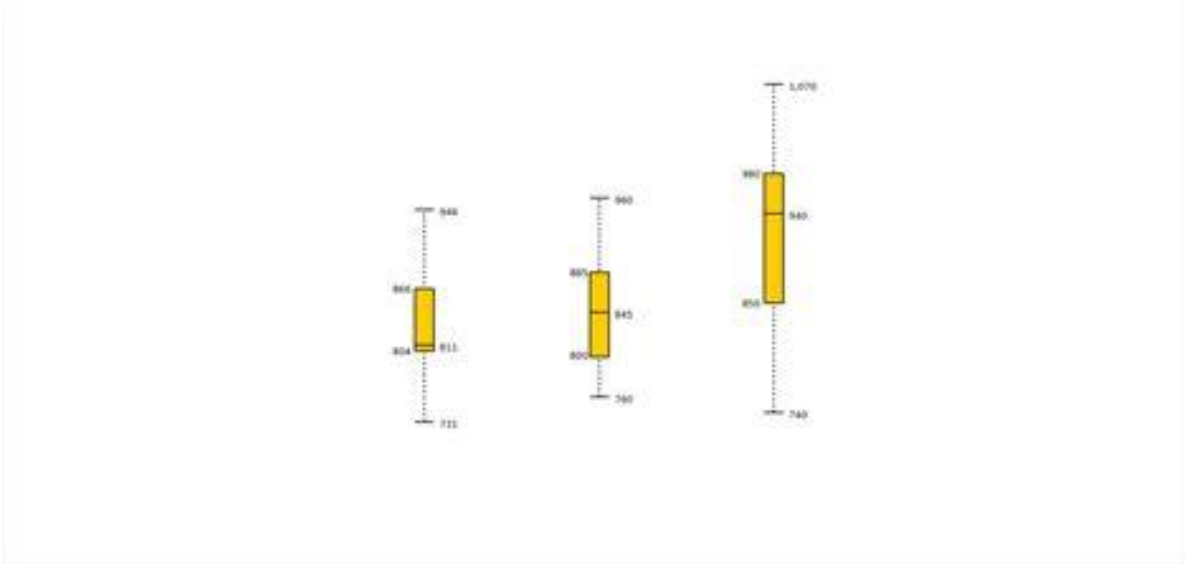


Рис. 15.11 Лінійний частотний полігон

Бокс дозволяє показати одразу розмах значень, найбільше і найменше значення конкретного розподілу, медіану та діапазон куди потрапляє 50 відсотків усіх значень (і відповідно 25 та 75 відсотків).

Бокс-графік – один із найбільш ефективних способів показати одразу багато розподілів різних величин на одному графіку.

7. Кореляція

Ключові слова: зростає разом з, падає разом з, змінюється разом з, викликане, причина якого, слідує за.

Щоб показати зв’язок між парами кількісних змиихвикористовуємо точковий графік.

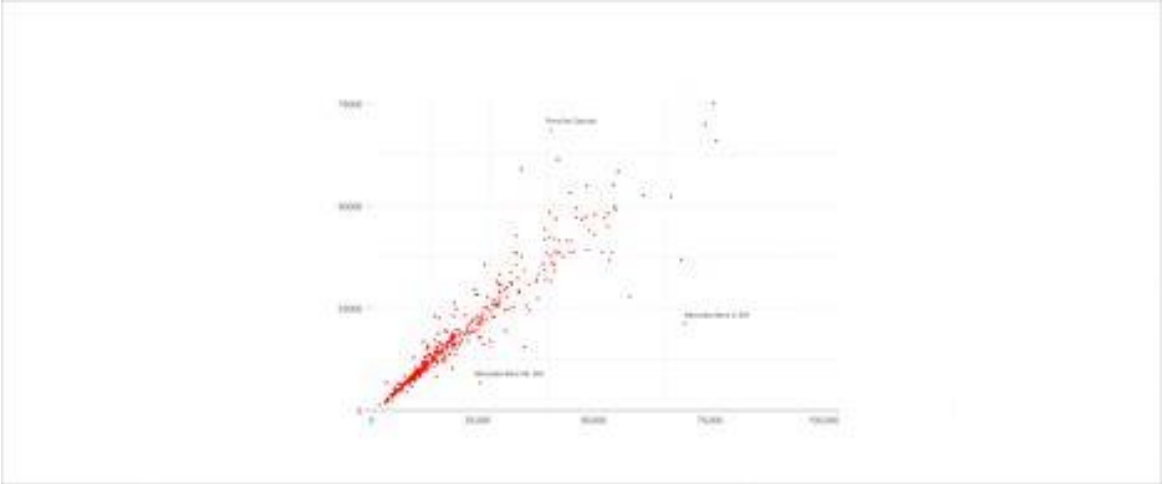


Рис. 15.12 Точковий графік

8. Географічні дані

Ключові слова: географія, локація (позиція), де розташоване, регіон, територія, країна, місто, область тощо.

Два найкращих способи кодування, які найзручніше використовути зайняті для карти (х у та довжина із спільною базою). Залишається розмір, інтенсивність кольору та ширина для кількісних даних, хоча вони не такі точні:

- точки різного розміру;
- точки або інші форми з різною інтенсивністю кольору;

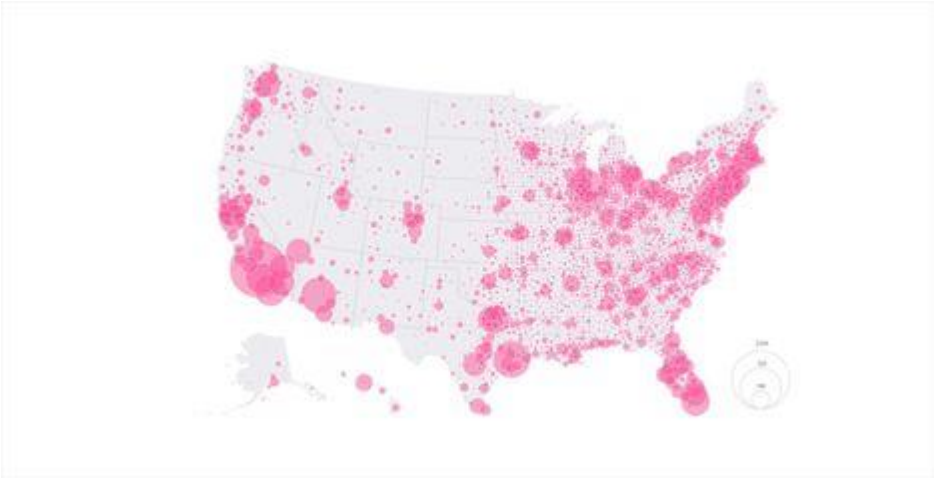


Рис. 15.13 Точки або інші форми з різною інтенсивністю кольору

•інтенсивніть кольору для різних гео-регіонів;

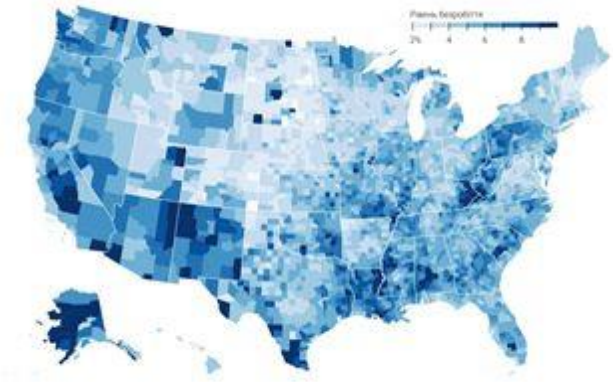


Рис. 15.14 Інтенсивністю кольору для різних гео-регіонів

•лінії різної ширини або інтенсивність кольору

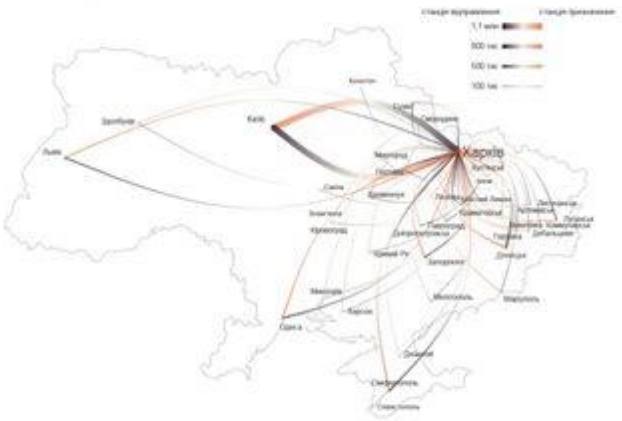


Рис. 15.15 Лінії різної ширини або інтенсивність кольору

15.3. Ефективність візуального кодування

Дані першого типу (категорійні, невпорядковані) потрібно показувати за допомогою такого візуального кодування, що зберігає відмінність та ідентичність – наприклад, різні кольори або різна геометрична форма. Дані другого типу(впорядковані та кількісні) потрібно показувати так, щоб наша візуальна система сприймала порядок.Тобто, якщо у якості каналу задіяний колір, тоді це має бути або перехід від кольору до сірого (десатурація), або однаковий колір з різними інтенсивностями, а не різні кольори.

З іншого боку, різні канали навіть якщо придатні - не однаково ефективні. Тому необхідно використовувати принцип - "за допомогою найбільш видимого (сильного) каналу потрібно кодувати найголовнішу інформацію" (атрибут в даних). Іншими словами - більш важливі змінні (атрибути) у даних повинні кодуватися більш ефективними, найбільш помітними візуальними каналами. Менш важливі - менш ефективними. Це є найбільш важлививй принцип інформаційного дизайну.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Battling Infectious Diseases in the 20th Century: The Impact of Vaccines [Електронний ресурс]. – Режим доступу: http://graphics.wsj.com/infectious-diseases-and-vaccines/.

2. Вищі навчальні заклади [Електронний ресурс]. – Режим доступу: https://ukrstat.org/uk/operativ/operativ2005/osv_rik/osv_u/vuz_u.html.

3. Базовые графические возможности R: функция plot() [Електронний ресурс]. – Режим доступу: http://r-analytics.blogspot.com/2011/10/r-plot.html#.Wumz9rj2M2A.

4. Набір даних [Електронний ресурс]. – Режим доступу: http://www.public.iastate.edu/~maitra/stat501/datasets/anaconda.dat.

5. Експеримент Мілґрема [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D0%B8%D0%BC%D0%B5%D0%BD%D1%82_%D0%9C%D1%96%D0%BB%D2%91%D1%80%D0%B5%D0%BC%D0%B0.

6. Explained: Sigma [Електронний ресурс]. – Режим доступу: http://news.mit.edu/2012/explained-sigma-0209.

7. Bessel's correction [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Bessel%27s_correction#.

8. United States Census Bureau [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/United_States_Census_Bureau.

9. Классические методы статистики: F-критерий Фишера [Електронний ресурс]. – Режим доступу: http://r-analytics.blogspot.com/2012/03/f.html#.WuMC9bj2M2A.

10. Школа інфографіки [Електронний ресурс]. – Режим доступу: http://texty.org.ua/pg/article/devrand/read/40074/Shkola_infografiky_Mova_vizualnyh_symvoliv_Konspekt_lekciji.

НАВЧАЛЬНЕ ВИДАННЯ

Неспляк Дмитро Михайлович
кандидат фізико-математичних наук
доцент кафедри інформатики ЛЬВДУВС

ЗАСТОСУВАННЯ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ У ПСИХОЛОГІЇ

Конспект лекцій

Комп’ютерний набір і верстка
Неспляк Д. М.

Електронний ресурс

Львівський державний університет внутрішніх справ
79007, м. Львів, вул. Городоцька, 26

Неспляк Д. М.

Ш 55 Застосування інноваційних технологій у психології: конспект лекцій / Д. М. Неспляк – Львів: Львівський державний університет внутрішніх справ, 2018. – 149 с.