

Кнопова В.П.

*Вступ до
багатовимірної
статистики*

Київ

2024

Вступ до багатовимірної статистики

Курс лекцій

В. Кнопова

Київ

2024

Зміст

1 Лінійна алгебра. Нормальний розподіл.	
Відстань Махalanобіса	6
1.1 Відомості з лінійної алгебри	6
1.2 Багатовимірний нормальний розподіл	8
1.3 Приклади	10
2 Тести для перевірки нормальності розподілу	14
2.1 Тест Колмогорова-Сміrnova (Kolmogorov-Smirnoff test)	14
2.2 Тест Шапіро-Уілка (Shapiro-Wilk test)	15
2.3 Деякі інші тести на багатовимірну нормальність	17
3 Оцінки максимальної вірогідності (ОМВ) Тестування середнього	19
3.1 Розподіли \bar{y} та S	21
4 Перевірка гіпотези про середнє	23
4.1 Гіпотеза про середнє. Відома коваріаційна матриця	24
4.2 Гіпотеза про середнє. Невідома коваріаційна матриця	25
4.3 Гіпотеза про рівність середніх двох виборок	27
4.4 Приклади	28
5 Перевірка гіпотези про коваріаційну матрицю	34
5.1 Статистика відношня вірогідностей. Тест про сферичність	34
5.2 Тест на рівність коваріаційних матриць.	36
5.3 Тест про незалежність двох субвекторів	38
5.4 Приклади	40
6 ANOVA: одновимірний дисперсійний аналіз (однофакторна модель)	45
6.1 ANOVA модель	46
6.2 Приклади	49
7 MANOVA: багатовимірний дисперсійний аналіз (однофакторна модель)	51
7.1 MANOVA модель	51
7.2 Приклади	54
8 Множинна лінійна регресія та багатовимірна лінійна регресія	58
8.1 Проста лінійна множинна регресія	59
8.2 Багатовимірна регресія	63
8.3 Приклади	65
9 Лінійний дискримінантний аналіз	70
9.1 Побудова дискримінантної функції	70
9.2 Приклади	74
10 Метод головних компонент	79
10.1 Геометричний підхід	80
10.2 Алгебраїчний підхід	81
10.3 Приклади	83

11 Задачі класифікації	90
11.1 Метод Фішера (лінійний класифікатор), 2 групи	90
11.2 Байесів класифікатор	90
11.3 Випадок декількох груп, спільна коваріаційна матриця	91
11.4 Похибка класифікації	93
11.5 Метод найближчих сусідів	94
11.6 Приклади	94
12 Кластерний аналіз	101
12.1 Типи кластерізації	101
12.2 Приклади	110
13 Факторний аналіз	118
13.1 Метод головних компонент у факторному аналізі	122
13.2 Метод головних факторів	123
13.3 Вибір кількості факторів та інтерпретація	124
13.4 Приклади	124
14 Додаток	135

Вступ

Всі моделі хибні, але деякі з них є корисними

Дж. Е. П. Бокс

Даний курс лекцій базується на матеріалах занять, проведених автором на механіко-математичному факультеті КНУТШ та в КАУ. В першій частині ми будемо вивчати багатовимірні моделі лінійної регресії та пов'язані з ними задачі оцінювання параметрів та перевірки гіпотез про ці параметри. На відміну від одновимірного випадку, параметри в багатовимірній лінійній регресії – це вектори або матриці, що робить процедуру оцінювання складнішою за одновимірний випадок.

Другу частину курсу присвячено задачам класифікації та проблемі вибору оптимальної кількості важливих факторів, які із достатньою точністю описують нашу модель. А саме: ми розберемо основні принципи дискримінантного аналізу, методу головних компонент, кластерного та факторного аналізу.

В нашому курсі ми будемо використовувати пакет R з тих міркувань, що мова R є простою для розуміння і для ілюстрації математичних моделей.

Курс лекцій базується на підручниках Алвіна Ренчера [R02] та Алвіна Ренчера в співавторстві з Брюсом Шаальє [RS08].

Практичні заняття на Python або на R можна знайти за наступними посиланнями:

Практичні заняття на Python:

<https://github.com/VicKnopova/MultidimStat---Python.git>

Практичні заняття на R:

<https://github.com/VicKnopova/MultidimStat---R.git>

Там же можна знайти посилання на використані web-ресурси; деякі посилання на онлайн ресурси також надані прямо в тексті лекцій. Пакети, процитовані в курсі, постійно оновлюються, тому до прикладів кодів треба ставитися творчо, і в разі необхідності дивитися документацію та оновлення відповідних пакетів.

Автор дякує співробітникам Чернівецького університету Ігорю Малику та Тарасу Лукашеву за гостинність під час перебування в Чернівцях в березні – квітні 2022, коли і було написано основну частину цього курсу.

Окрема подяка студентам Київського академічного університету Софії Гуракі, Данилу Петраківському, Михайлу Поліщуку, Миколі Корабльову, Федіру Прокопенко та Володимиру Бараннику за запитання та зауваження, які допомогли зробити ці лекції кращими.

Автор щиро дякує своєму чоловіку Бориславу Строньку за підтримку, а також своїм котам Францу-Йозефу, Ніндзі та Аньоті за активну участь в написанні цих лекцій.

Автор буде вдячна за рекомендації та зауваження, які можна надсилати за адресу vickknopova@gmail.com

Позначення

- Великі літери A, B, \dots означають матриці.
- Малі літери x, y, \dots означають елементи \mathbb{R} .

- Малі літери жирним шрифтом \mathbf{x} , \mathbf{y} , ... означають вектори \mathbb{R}^d , $d \geq 1$.
- A' , \mathbf{x}' - транспоновані матриця A та вектор \mathbf{x} .
- В деяких випадках великі літери можуть позначати статистики (наприклад, T^2 , Z^2).

В наведених кодах ми не будемо розрізняти позначення для вектору та для числового елементу, тобто x може позначати вектор або число, в залежності від контексту.

1 Лінійна алгебра. Нормальний розподіл. Відстань Махalanобіса

Література: [R02, Гл. 2–4]. Пакети MASS [MASS], ellipse [ellipse], mvtnorm [mvtnorm].

1.1 Відомості з лінійної алгебри

Пригадаємо деякі відомості з лінійної алгебри.

Нехай A - квадратна матриця розмірності $n \times n$. Обернена матриця A^{-1} існує, якщо $\text{rank } A = n$; це означає, що A не сингулярна. Розглянемо деякі властивості.

- Якщо матриці A та B не сингулярні та однієї розмірності, то

$$(AB)^{-1} = B^{-1}A^{-1}.$$

- Якщо матриця B не сингулярна, то

$$AB = CB \implies A = C.$$

- Нехай A' - матриця транспонована до A , тоді

$$(A')^{-1} = (A^{-1})'.$$

- Матриця A називається *додатно визначеною* (*відповідно, напіввизначеною*), якщо

$$\forall \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{x} \neq 0 : \quad \mathbf{x}' A \mathbf{x} > 0 \quad (\mathbf{x}' A \mathbf{x} \geq 0).$$

Твердження 1.1. Якщо B є матрицею розмірності $n \times p$, $p \leq n$, то матриця $A = B'B$ є додатно визначеною.

Твердження 1.2. Якщо матриця A є додатно визначеною, то існує єдина сингулярна низькорангова матриця B така, що $A = B'B$.

Останнє представлення має назву *розвклад Холецького*, причому матриця B будується явно.

- Позначимо через $|A|$ визначник матриці A . Тоді:

- визначник добутку двох матриці дорівнює добутку визначників відповідних матриць:

$$|AB| = |A||B|;$$

- визначники матриці та відповідної її транспонованої матриці рівні між собою:

$$|A| = |A'|;$$

- якщо існує обернена матриця A^{-1} до матриці A , то

$$|A^{-1}| = \frac{1}{|A|}.$$

6. Матриця $A^{\frac{1}{2}}$ визначається із співвідношення $A^{\frac{1}{2}}A^{\frac{1}{2}} = A$ (тобто квадрат цієї матриці має дорівнювати A). Тоді визначник $A^{\frac{1}{2}}$ означенням дорівнює

$$|A^{\frac{1}{2}}| = |A|^{\frac{1}{2}}.$$

В більш загальному випадку, для матриці A^k , $k \geq 1$, маємо

$$|A^k| = |A|^k$$

7. Власними числами матриці A називаються корені характеристичного рівняння

$$|A - \lambda I| = 0.$$

Власними векторами матриці A називаються такі вектори A' , $\mathbf{v} \neq 0$, що

$$A\mathbf{v} = \lambda\mathbf{v}.$$

Твердження 1.3. Якщо матриця A розмірності $n \times n$ є додатно визначеною, то $\lambda_i > 0$, $i = 1, \dots, n$. Якщо матриця A є додатно напіввизначеною, то $\lambda_i \geq 0$, та кількість додатних λ_i дорівнює рангу A , тобто $\#\{\lambda_i : \lambda_i > 0\} = \text{rank } A$.

Наступне твердження є дуже важливим для побудови "власного базису".

Твердження 1.4. Власні вектори \mathbf{v}_i додатно визначеної симетричної матриці A є ортогональними.

Тобто, з векторів додатно визначеної матриці можна утворити базис, причому розмірність натягнутого на ці вектори простору співпадає із рангом матриці A . Нехай $\text{rank } A = n$. Нормуємо власні вектори \mathbf{v}_i , $i = 1, \dots, n$ та утворимо з них матрицю C :

$$C = (\mathbf{v}_1, \dots, \mathbf{v}_n) = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nn} \end{pmatrix}, \quad \text{де} \quad \mathbf{v}_i = \begin{pmatrix} v_{1i} \\ v_{2i} \\ \vdots \\ v_{ni} \end{pmatrix}.$$

Оскільки вектори нормовані, то $CC' = I$. Тоді

$$A = ACC' = (A\mathbf{v}_1, \dots, A\mathbf{v}_n) C' = (\lambda_1\mathbf{v}_1, \dots, \lambda_n\mathbf{v}_n) C' = CDC', \quad (1.1)$$

де D - діагональна матриця вигляду

$$\begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}.$$

Отже, для додатно визначеної симетричної матриці має місце представлення $A = CDC'$, яке називається спектральним розкладом матриці A .

9. Спектральний розклад дозволяє легко рахувати степені матриці. Наприклад, для матриці $A^{\frac{1}{2}}$ маємо

$$A^{\frac{1}{2}} = CD^{\frac{1}{2}}C',$$

де

$$D^{\frac{1}{2}} = \begin{pmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_n} \end{pmatrix}.$$

1.2 Багатовимірний нормальній розподіл

Пригадаємо означення та властивості багатовимірного нормального розподілу.

Розглянемо вектор $\mathbf{x} = (x_1, \dots, x_p)$ розмірності p , утворений з незалежних у сукупності стандартних нормальніх величин. Тоді ми будемо говорити, що \mathbf{x} має стандартний нормальній розподіл в \mathbb{R}^p та позначати це $\mathbf{x} \sim N_p(0, I_p)$.

Нехай тепер $\boldsymbol{\mu} \in \mathbb{R}^p$ та A є невиродженою матрицею в \mathbb{R}^p . Розглянемо перетворення $\mathbf{y} = \boldsymbol{\mu} + A\mathbf{x}$. Лінійне перетворення зберігає нормальність, тому \mathbf{y} теж має нормальній розподіл, але тепер координати цього нормального вектора вже (в загальному випадку) залежні.

Вправа 1.1. Яким має бути лінійний оператор A , щоб координати перетвореного нормального вектора $A\mathbf{x}$ були незалежними?

Наведемо тепер формальне означення.

Означення 1.1. Якщо вектор \mathbf{y} можна зобразити у вигляді

$$\mathbf{y} = \boldsymbol{\mu} + A\mathbf{x}, \quad (1.2)$$

де $\boldsymbol{\mu} \in \mathbb{R}^p$ та A є невиродженою матрицею в \mathbb{R}^p , то такий вектор будемо називати нормальним вектором в \mathbb{R}^p .

Твердження 1.5. Нехай \mathbf{y} має вигляд (1.2). Тоді щільність \mathbf{y} має вигляд

$$g(\mathbf{v}) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{v}-\boldsymbol{\mu})' \Sigma^{-1} (\mathbf{v}-\boldsymbol{\mu})}, \quad \mathbf{v} \in \mathbb{R}^p, \quad (1.3)$$

де $\Sigma = AA'$ є матрицею коваріації вектору \mathbf{y} . Ми будемо позначати нормальній вектор з середнім $\boldsymbol{\mu}$ та коваріаційною матрицею Σ наступним чином: $\mathbf{y} \sim N_p(\boldsymbol{\mu}, \Sigma)$.

Навпаки, маючи зображення $\mathbf{x} = \boldsymbol{\mu} + B\mathbf{z}$, можна знайти \mathbf{z} :

$$\mathbf{x} = \boldsymbol{\mu} + B\mathbf{z} \Rightarrow \mathbf{x} - \boldsymbol{\mu} = B\mathbf{z} \Rightarrow \mathbf{z} = B^{-1}(\mathbf{x} - \boldsymbol{\mu}).$$

Зауважимо, що $B = \Sigma^{\frac{1}{2}}$.

Означення 1.2. Величина $\sqrt{(\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})} =: \Delta$ називається відстанню Махаланобіса між векторами \mathbf{x} та $\boldsymbol{\mu}$ в \mathbb{R}^p .

В одновимірному випадку, тобто при $p = 1$, щільність розподілу має вигляд

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{|x-\mu|^2}{2\sigma^2}},$$

а квадрат відстані Махаланобіса дорівнює $\Delta^2 = \frac{|x-\mu|^2}{\sigma^2}$. Нехай $\Delta^2 = R^2$. Тоді

$$|x - \mu|^2 = \sigma^2 R^2,$$

що геометрично означає в одновимірному випадку відрізок з центром в т. μ та довжиною $2\sigma R$. Таким чином, якщо зафіксувати Δ^2 , то можна оцінити ймовірність потрапляння елементів вибірки (ξ_1, \dots, ξ_n) в цей відрізок, тобто

$$\mathbb{P}(|\bar{\mu} - \mu| \leq \sigma R) = 1 - \alpha, \quad \text{де} \quad \bar{\mu} = \frac{1}{n} \sum_{k=1}^n \xi_k.$$

Аналогічно до того, як в одновимірному випадку дисперсія σ^2 вказує на розкид значень навколо середнього μ , визначник коваріаційної матриці $|\Sigma|$ вказує на розкид векторів \mathbf{x} навколо $\boldsymbol{\mu}$ в p -вимірному просторі. Так, при малих значеннях $|\Sigma|$ вектори \mathbf{x} сконцентровані більшіше до $\boldsymbol{\mu}$, і навпаки.

Зазначимо, що матриця Σ є симетричною (оскільки вона є матрицею коваріацій) та додатно визначеною, тому можемо записати її спектральний розклад

$$\Sigma = CDC',$$

де C – матриця з власних векторів Σ , а D – діагональна матриця з власних чисел Σ . Обчислимо визнаник матриці Σ .

$$|\Sigma| = |CDC'| = |C||D||C'| = |CC'||D| = |D| = \prod_{k=1}^p \lambda_k.$$

Оскільки Σ додатно визначена, то $\lambda_k > 0$, $k = \overline{1, p}$. Помітимо, що в такому разі малим значенням $|\Sigma|$ відповідають малі значення власних чисел матриці Σ .

Розглянемо властивості багатовимірного нормальногоподілу.

Лема 1.1. 1. Нехай $\mathbf{a} = (a_1, \dots, a_p)$, тоді $\mathbf{a}'\mathbf{x} \sim N_1(\mathbf{a}'\boldsymbol{\mu}, \mathbf{a}'\Sigma\mathbf{a})$.

2. Якщо A – матриця розмірності $q \times p$, $q \leq p$, то $A\mathbf{x} \sim N_q(A\boldsymbol{\mu}, A\Sigma A')$.

Вправа 1.2. a) Знайти характеристичну функцію $\mathbf{y} \sim N_p(\boldsymbol{\mu}, \Sigma)$.

b) Довести Лему 1.1.

Сформулюємо інші властивості.

3. Нехай $\mathbf{z} \sim N_p(0, I_p)$, тоді

$$\mathbf{z}'\mathbf{z} = (z_1, \dots, z_p) \begin{pmatrix} z_1 \\ \vdots \\ z_p \end{pmatrix} = \sum_{i=1}^p z_i^2 \sim \chi_p^2.$$

Вище було отримано, що $\mathbf{z} = B^{-1}(\mathbf{x} - \boldsymbol{\mu}) = \Sigma^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu})$, тому

$$\begin{aligned} \mathbf{z}'\mathbf{z} &= \left(\Sigma^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu})\right)' \Sigma^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu}) = (\mathbf{x} - \boldsymbol{\mu})' \left(\Sigma^{-\frac{1}{2}}\right)' \Sigma^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu}) \\ &= (\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) = \Delta^2 \sim \chi_p^2. \end{aligned}$$

Якщо \mathbf{x} – випадковий вектор розмірності p , то квадрат відстані Махalanобіса від \mathbf{x} до $\boldsymbol{\mu}$ не перевищує $x_{p,\alpha}$ з ймовірністю $1 - \alpha$:

$$\mathbb{P}\left((\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) \leq x_{p,\alpha}\right) = 1 - \alpha,$$

де $x_{p,\alpha}$ – квантиль рівня $1 - \alpha$ розподілу χ_p^2 .

4. Розглянемо вектори \mathbf{x} в \mathbb{R}^p та \mathbf{y} в \mathbb{R}^q такі, що

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N_{p+q} \left(\begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \right).$$

У випадку, коли $p = q = 1$, та $x \sim N(\mu_x, \sigma_x^2)$, $y \sim N(\mu_y, \sigma_y^2)$, маємо

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim N_2 \left(\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix} \right),$$

де $\rho = \frac{\text{cov}(x,y)}{\sqrt{\sigma_x^2\sigma_y^2}}$.

5. В одновимірному випадку, при $p = q = 1$, регресією y на x є умовне математичне сподівання y , якщо відоме x :

$$\mathbb{E}(y|x) = \min_g \mathbb{E}(y - g(x))^2,$$

де функція $g = g(x)$ – борелева.

Це умовне математичне сподівання можна знайти явно:

$$\mathbb{E}(y|x) = \mu_y + \rho \frac{\sigma_y}{\sigma_x} (x - \mu_x).$$

Функція $m(y) := \mathbb{E}(y|x)$ називається функцією регресії. Аналогічно, в \mathbb{R}^{p+q} регресією \mathbf{y} на \mathbf{x} є

$$\mathbb{E}(\mathbf{y}|\mathbf{x}) = \boldsymbol{\mu}_y + \Sigma_{xy}\Sigma_{xx}^{-1}(\mathbf{x} - \boldsymbol{\mu}_x),$$

та регресією \mathbf{x} на \mathbf{y} є

$$\mathbb{E}(\mathbf{x}|\mathbf{y}) = \boldsymbol{\mu}_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \boldsymbol{\mu}_y).$$

1.3 Приклади

Пригадаємо спочатку деякі базові операції в R .

Задомо вектор \mathbf{x} :

```
x<-c(1,2,3,4,5,6,7,8,9)    # вектор x
matrix(x,nrow=3)            # 3 рядки, заповнення по стовбчикам
```

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

Якщо ми хочимо заповнювати матрицю по рядкам, то це можна зробити наступним чином:

```
matrix(x,nrow=3,byrow= TRUE)
```

Позначимо тепер цю матрицю через A та порахуємо її детермінант:

```
A<-matrix(x, nrow=3) # визначаємо матрицю A
det(A) # порахуємо детермінант
```

На жаль, детермінант цієї матриці дорівнює 0.

Пригадаємо деякі операції з матрицями. Задамо також матрицю, детермінант якої не дорівнює 0.

```
x<-c(1,2,3,4,5,6,7,8,8) # змінимо матрицю
B<-matrix(x, nrow=3)
det(B) # тепер вже не 0
t(B) # транспонована матриця
TrB<-sum(diag(B)) # слід матриці
A%*%B # добуток матриць
solve(B) # обернена до B
```

Також, пригадаємо, як в *R* можна згенерувати вибірку з певним (заданим) розподілом, знайти значення щільності в точці та знайти квантілі розподілу.

Якщо ви щось забули, можна викликати справку за допомогою знаку питання ?, тобто наступний виклик дає посилання на відповідну сторінку документації:

```
?dnorm # задати запитання, що таке dnorm
```

```
dnorm(z, mean = 0, sd = 1, log = FALSE) # щільність в точці z нормального
розділу з середнім 0 і стандартним відхиленням 1;

pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE) # значення
в точці X функції розподілу нормального розподілу з середнім 0 і стандартним
відхиленням 1;

qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE) # квантіль
рівня p нормального розподілу з середнім 0 і стандартним відхиленням 1.
```

Якщо обрати lower.tail = FALSE, то ми отримаємо квантіль рівня $1 - p$.

Для того, щоб вивести на екран перші 6 значень згенерованої вибірки, використаємо функцію head:

```
N<-rnorm(1000, 0, 1)
head(N)
```

```
-0.1535418 0.9394166 1.7203574 -1.6493087 0.3559709 0.1189281
```

Тобто, ми згенерували 1000 стандартних нормальні розподілених випадкових величин, але вивели на екран 6.

Ще одна базова функція, яку ми будемо часто використовувати, це hist, яка дозволяє побудувати гістограму розподілу. В залежності від потреби, можна побудувати гістограму частот, або гістограму щільності. Відповідно, у першому випадку по осі *OY* будуть відкладені відносні частоти, а у другому – абсолютні.

```

hist(N, breaks=150, xlim=c(0, 20), freq=FALSE) # графік щільності
hist(N, breaks=150, xlim=c(0, 20), freq=TRUE) # графік частот

```

Для того, щоб вивчати багатовимірний нормальній розподіл, нам знадобляться наступні пакети і бібліотеки MASS, mvtnorm:

```

install.packages("mvtnorm") # Завантажуємо пакети
install.packages("MASS")
library(mvtnorm)           # відкриваєте бібліотеки
library(MASS)

```

Задамо тепер вектор середніх значень, матрицю кореляцій і побудуємо проекцію двовимірного розподілу на площину XOY .

```

mean<-c(0,0)                                # вектор середніх
sig<-matrix(c(1, .5, .5, 1), nrow =2, byrow= FALSE) # коваріація
mv<-rmvnorm(1000, mean, sig)                 # генеруємо нормальній розподіл з
# цими середнім і коваріацією
mv.kde <- kde2d(mv[,1], mv[,2], n = 50)    # генеруємо щільність. kde = kernel
# density estimation
image(mv.kde)                               # малюємо картинку в 2d
contour(mv.kde, add = TRUE)                  # малює "бокс" навколо малюнку
box()                                         # малює "бокс" навколо малюнку
title(main = "Проекція на XY", font.main = 4)

```

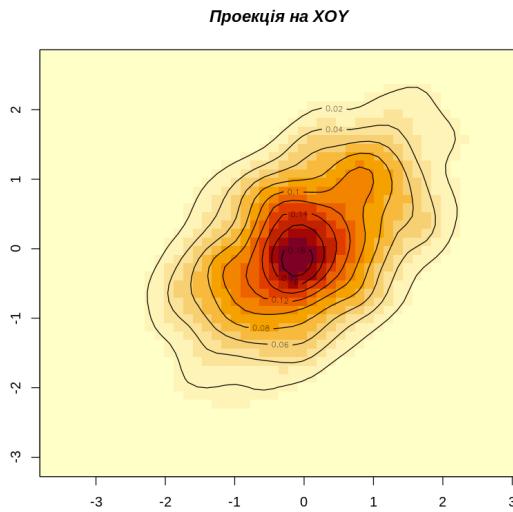


Рис. 1: Проекція двовимірної щільності розподілу на площину XOY

На Рисунку 1 наведені контури, які відповідають значенням щільності нормального розподілу, спроектовані на XOY .

Розглянемо тепер двовимірні довірчі інтервали. На площині двовимірним довірчим інтервалом буде еліпс. Для графічного зображення завантажимо пакет ellipse.

```

install.packages('ellipse')
library(ellipse)
rho <- cor(mv)
y_on_x <- lm(mv[,2] ~ mv[,1]) # регресія Y на X
x_on_y <- lm(mv[,1] ~ mv[,2]) # регресія X на Y
lines(ellipse(rho), level = .95), col="red")
lines(ellipse(rho, level = .99), col="green")
lines(ellipse(rho, level = .90), col="blue")
abline(y_on_x, col="magenta") # малюємо лінію регресії у на x
abline(x_on_y, col="blue") # малюємо лінію регресії x на у

```

На Рисунку 2 зеленим кольором позначено еліпс, який відповідає 99%-му двовимірному довірчому інтервалу (тобто точка має попасти в середину еліпса з ймовірністю 0.99). Відповідно, червоним та синім кольорами позначено 95%-й та 90%-й довірчі еліпси. Рожевим кольором зображена лінія регресії y на x , та синім – x на y . Зауважимо, що лінії регресії не співпадають з головними осями еліпсу!

Якщо задана коваріаційна матриця, а не кореляційна, то треба перешкалювати дані:

```

ellipse(x, scale = c(1, 1), centre = c(0, 0), level = 0.95,
t = sqrt(qchisq(level, 2)), which = c(1, 2), npoints = 100,
center = centre,
...)

```

Двовимірні довірчі інтервали

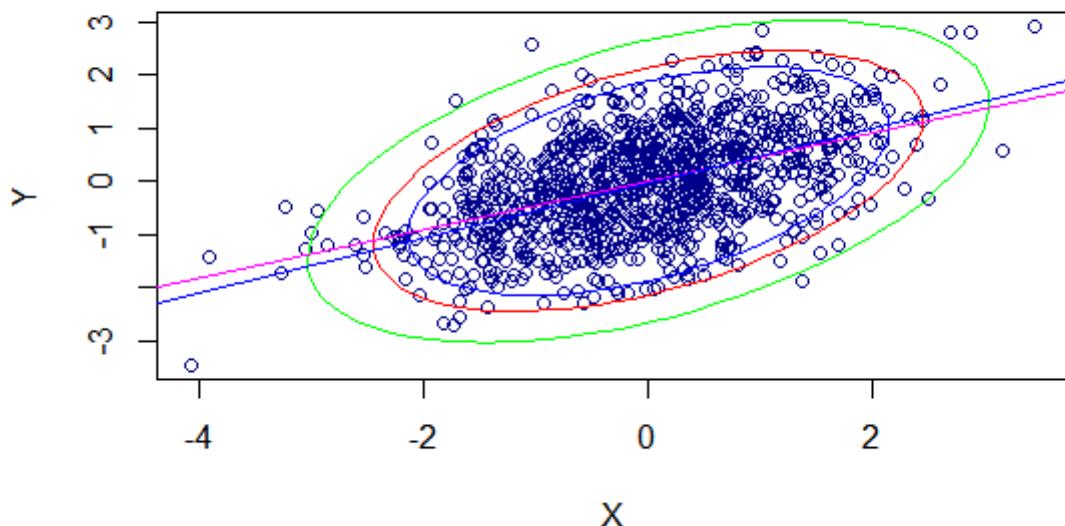


Рис. 2: Еліпсоїди розсіювання та лінії регресії

Вправа 1.3. Розглянемо нормальний розподiл в \mathbb{R}^2 з параметрами $(1, 1, 0.6, 2, 2)$. Пo-

будувати довірчий еліпсоїд рівня 0.95 (тобто в еліпсоїд попаде не менше 95% значень вибірки), а саме:

- 1) знайти рівняння головних осей та записати в цих нових координатах рівняння елісу.
- 2) Побудувати лінії регресії однієї координати на іншій.
- 3) Пояснити, як буде змінюватися форма елісу при зміні параметрів.

2 Тести для перевірки нормальності розподілу

В наступних 2-х розділах ми будемо перевіряти гіпотези про вектори середніх значень та коваріаційні матриці. Але перш за все треба переконатися, що наша вибірка дійсно нормальнa.

Найпростіше, що можна зробити – це візуально перевірити, чи може вибірка мати нормальний розподiл. Якщо ми маємо справу з одновимірними спостереженнями, то можна побудувати Q-Q plot, або зобразити гістограму. Але таке графічне зображення не є дуже надiйним, хоча дозволяє відкинути припущення нормальностi, якщо навiть вiзуально ясно, що воно не виконано. Для того, щоб переконатися, що вибірка має нормальний розподiл, застосовують статистичнi тести.

Розглянемо деякi тести на розподiл.

2.1 Тест Колмогорова-Смірнова (Kolmogorov-Smirnov test)

Література: [K07].

Тест Колмогорова-Смірнова дозволяє перевiрити, чи збiгається функцiя розподiлу окремих спостережень вибiрки $\mathbf{x} = (x_1, \dots, x_n)$ з заданою (неперервною) функцiєю розподiлу $F(x)$ на \mathbb{R} .

$$\hat{F}_n(x) = \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{x_k < x}.$$

Тестова статистика має вигляд

$$\hat{\kappa}_n(X) = \sqrt{n} \sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)|.$$

Перевiряємо нульову гiпотезу

$$H_0 : \quad \mathbb{P}(x_1 < x) = F(x) \quad \forall x \in \mathbb{R},$$

проти альтернативної

$$H_1 : \quad \mathbb{P}(x_1 < x) = G(x) \quad \forall x \in \mathbb{R}, \quad G(x) \neq F(x), \quad G \in C(\mathbb{R}).$$

За умови нульової гiпотези статистика Колмогорова $\hat{\kappa}_n$ слабко збiгається до випадкової величини κ , яка має розподiл Колмогорова, тобто

$$\mathbb{P}(\kappa < x) = K(x) = \sum_{k=-\infty}^{\infty} (-1)^k \exp(-2k^2 x^2).$$

Зауважимо, що це одновимірний тест, хоча існують узагальнення на багатовимірний випадок. Причина полягає у складності визначення у багатовимірному випадку емпіричної функції розподілу \hat{F}_n .

В R є вбудована функція, яка перевіряє гіпотезу H_0 за допомоги критерію Колмогорова-Смірнова:

```
ks.test(x, y, alternative = c("two.sided", "less", "greater"))
```

Тут x, y – вектори даних, причому y генерується за допомогою функції розподілу F :

```
y <- rnorm(50, -1, 1)
```

Однобічний тест означає, що ми перевіряємо, де саме лежить вибіркова функція розподілу (CDF). Якщо `alternative = "less"`, то за умови виконання альтернативи CDF знаходить знизу по відношенню до графіку справжньої функції розподілу (або CDF y): $F \leq G$. Якщо `alternative = "greater"` – то зверху: $F \geq G$.

Можна задати розподіл, наприклад, $\Gamma(3, 2)$:

```
x <- rnorm(50)
ks.test(x, "pgamma", 3, 2)
```

Тест Колмогорова-Смірнова може частіше відхиляти нульову гіпотезу на малих виборках, та є більш чутливим до поведінку функції розподілу в середині інтервалу, ніж на кінцях.

Якщо ми перевіряємо гіпотезу про нормальній розподіл, але не знаємо параметри, можна застосувати поправку Ліллієфорса. В цьому тесті справжні значення параметрів замінюють на вибіркові, тому сама статистика менша за статистику Колмогорова. Розподіл цієї статистики відомий лише чисельно.

Зауважимо, що в такому варіанті, як описано вище, тест Колмогорова-Смірнова застосовується лише для неперервних розподілів. Існують версії цього тесту для дискретних розподілів, але то окрема тема.

2.2 Тест Шапіро-Уілка (Shapiro-Wilk test)

Література: [SW65] ($p = 1$), [VAG09] ($p \geq 1$), [AD54].

Тест Шапіро-Вілка є тестом одновимірної вибірки $p = 1$. Розглянемо впорядковану виборку $\mathbf{x} = (x_1, \dots, x_n)$. За умови нормальності маємо $\mathbf{x} \sim N_n(\mathbf{m}, V)$ для певних \mathbf{m} і V :

$$\mathbf{m} = \mathbb{E}\mathbf{x} = (m_1, \dots, m_n), \quad V = (\text{cov}(x_i, x_j))_{i,j=1}^n.$$

Нехай тепер $\mathbf{y} = (y_1, \dots, y_n)$ – інша впорядкована виборка. За умови $y_i \sim N(\mu, \sigma^2)$, координати можна зобразити наступним чином:

$$y_i = \mu + \sigma x_i, \quad i = 1, \dots, n,$$

для деяких μ та σ^2 . Оцінимо μ та σ^2 методом найменших квадратів, тобто мінімізуємо

$$(\mathbf{y} - \mu\mathbf{1} - \sigma\mathbf{m})'V(\mathbf{y} - \mu\mathbf{1} - \sigma\mathbf{m}) \mapsto \min,$$

де $1 = (1, \dots, 1)'$. Отримаємо наступні оцінки:

$$\hat{\mu} = \frac{\mathbf{m}'V^{-1}(\mathbf{m}1' - 1'\mathbf{m})V^{-1}}{1'V^{-1}1\mathbf{m}'V^{-1}\mathbf{m} - (1'V^{-1}\mathbf{m})^2}, \quad \hat{\sigma}^2 = \frac{1'V^{-1}(1\mathbf{m}' - \mathbf{m}1')V^{-1}\mathbf{y}}{1'V^{-1}1\mathbf{m}'V^{-1}\mathbf{m} - (1'V^{-1}\mathbf{m})^2}$$

За умови симетричності, тобто $1'V^{-1}m = 0$, ці оцінки можна спростити та отримати

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n y_i, \quad \hat{\sigma}^2 = \frac{\mathbf{m}'V^{-1}\mathbf{y}}{\mathbf{m}'V^{-1}\mathbf{m}}.$$

Розглянемо W -статистику Шапіро-Уілка:

$$W = \frac{(\sum_{i=1}^n y_i a_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

де $\mathbf{a} = (a_1, \dots, a_n) = \frac{\mathbf{m}'V^{-1}}{(\mathbf{m}'V^{-1}V^{-1}\mathbf{m})^{1/2}}$. За умови нормальності вибірки \mathbf{x} , розподіл цієї статистики залежить лише від n і є табулюваним (хоча точне аналітичне зображення невідоме). Тому цю статистику можна використовувати для перевірки гіпотези про нормальність. В залежності від того, скільки точок попадає в "допустимий інтервал", таке буде значення статистики W . Критичними значеннями є такі значення емпіричної W , яка менша за табличне значення. У цьому випадку нульова гіпотеза відхиляється. Іншими словами, H_0 відхиляємо, якщо $W < c_{\alpha,n}$, де

$$\alpha = \mathbb{P}(W < c_{\alpha,n} | H_0).$$

Максимальне значення статистики Шапіро-Уілка – це 1, і в разі прийняття нульової гіпотези значення W близьке до 1 (див. [SW65]).

В R тест Шапіро-Вілка реалізується за допомогою будованої функції

```
shapiro.test(x)
```

В якості x можна взяти, наприклад, $\mathbf{x} = rnorm(100, 0, 1)$.

Існує багатовимірний аналог тесту Шапіро-Уілка. Нехай $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ вибірка випадкових векторів з \mathbb{R}^p .

Розглядається нульова гіпотеза $H_0: \mathbf{x}_i \sim N_p(\boldsymbol{\mu}, \Sigma)$, $i = 1, \dots, n$, $\boldsymbol{\mu}$, Σ – невідомі.

Нехай

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \quad S = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'). \quad (2.1)$$

Розглянемо нормовані вектори

$$\mathbf{z}_i^* = S^{-1/2}(\mathbf{x}_i - \bar{\mathbf{x}}), \quad z_i^* = (z_{1i}, \dots, z_{pi}).$$

В якості узагальненої статистика Шапіро-Уілка беруть

$$W^* = \frac{1}{p} \sum_{i=1}^p W_{\mathbf{z}_i},$$

де $W_{\mathbf{z}_i}$ – статистика Шапіро-Уілка для (одновимірних!) спостережень (z_{i1}, \dots, z_{in}) , $i = 1, \dots, p$. Як і в одновимірному випадку, H_0 відхиляємо, якщо $W^* < c_{\alpha,p,n}$, де

$$\alpha = \mathbb{P}(W^* < c_{\alpha,p,n} | H_0).$$

Як і в одновимірному випадку, хороші значення статистики W^* – це значення, близькі до 1. Таблицю кватнілей можна подивитися в [VAG09].

Для перевірки нормальності випадкових векторів застосовується, наприклад, пакет **mvnormtest**, який потрібно встановити та завантажити відповідну бібліотеку. Для перевірки нормальності застосовується функція (тут *DataX* у форматі матриці.)

```
mshapiro.test(t(DataX))
```

Тест Шапіро-Уілка не рекомендують використовувати для великих (більше 50) виборок. Це пов'язано з тим, що саму статистику важко порахувати чисельно (див. [Ro92], [Ro95]). Ройстон [Ro95] запропонував модифікацію тесту Шапіро-Уілка, яку може бути використано для вибірки розміру від 3 до 5000. Зауважимо також, що для симетричних розподілів тест Шапіро-Уілка працює найкраще, що пояснюється тим, що оцінки мають більш простий вигляд, що зменшує похибки.

2.3 Деякі інші тести на багатовимірну нормальність

Література: [M70]–[M80], [EH21], [HZ90], [AD54], [ZS14], [RW11], [YA07]. Пакети: **MVN**, **mvnTest**.

Існує багато різних тестів розподілу вибірки, наприклад тест Андерсона-Дарлінга ([AD54]; тест можна застосовувати також для інших розподілів), тест Мардіа [M70] – [M80], Генце-Цирклера [HZ90]).

Тест Мардіа полягає в тому, що будуть статистики для вимірювання параметрів асиметрії (skewness) та куртозису (kurtosis). В одновимірному випадку асиметрія і куртозис задаються наступним чином:

$$Skew = \frac{m_3}{m_2^{3/2}} \quad (\text{асиметрія}),$$

$$Kurt = \frac{m_4}{m_2^2} \quad (\text{куртозис}),$$

і вимірюють "скошеність" (skewness) та висоту і форму піка ймовірнісної щільності (kurtosis).

А саме, виходячи з векторів \mathbf{x}_i , $i = 1, \dots, n$, будуть наступні статистики:

$$b_{1,p} := \frac{1}{n^2} \sum_{i,j=1}^n ((\mathbf{x}_i - \boldsymbol{\mu})' S^{-1} (\mathbf{x}_i - \boldsymbol{\mu}))^3, \quad (2.2)$$

$$b_{2,p} := \frac{1}{n} \sum_{i,j=1}^n ((\mathbf{x}_i - \boldsymbol{\mu})' S^{-1} (\mathbf{x}_i - \boldsymbol{\mu}))^4. \quad (2.3)$$

За умовою нормальності розподілу, маємо асимптотичні співвідношення:

$$\frac{n}{6} b_{1,p} \sim \chi^2_{\frac{p(p+1)(p+2)}{6}}, \quad b_{2,p} \sim N \left(p(p+2), \frac{8p(p+2)}{n} \right). \quad (2.4)$$

Критичними значеннями статистик будуть відповідно великі значення.

Проблеми з тестом Мардіа виникають тоді, коли вибірка є "еліптично симетричною" (тобто контур рівня імовірнісної щільності має форму еліпса), що приводить до того, що параметри асиметрії нульові.

Наступні два тести схожі в тому, що обидва дозволяють вимірювати відстань між розподілами в інтегральній метриці. А саме, тест Андерсона-Дарлінга ([AD52], [AD54]), вимірює відстань між емпіричною \hat{F}_n і теоретичною F функціями розподілу в $L_2(\psi(F)dF)$:

$$W_n := n \int_{\mathbb{R}} |\hat{F}_n(x) - F(x)|^2 \psi(F(x)) dF(x),$$

де ψ – деяка невід’ємна вагова функція, яку ми обираємо так, щоб підкреслити ту область в \mathbb{R} , яка де різниця $\hat{F}_n(x) - F(x)$ максимально велика. Якщо обрати $\psi(x) = 1$, то ми отримаємо $n\omega^2$, де ω^2 – критерій фон Мізеса. З одного боку, тест Андерсона-Дарлінга є більш чутливим – для конкретної функції розподілу можна підібрати адекватну вагу і таким чином оптимізувати процес прийняття – відхилення гіпотези про розподіл. З іншого боку, критичні значення доводиться рахувати для кожного конкретного розподілу, що є в деякій мірі недоліком тесту.

Найбільш універсальним тестом нормальності великих виборок є тест Хенце-Циклера. Тест Хенце-Ціклера теж вимірює відстань між двома мірами: імовірнісною мірою даного розподілу та нормальним розподілом Q з діагональною коваріаційною матрицею. При цьому, відстань вимірюється в сенсі $L_1(\phi(\mathbf{x})d\mathbf{x})$, де ϕ – щільністьального розподілу Q . Обчислюється статистика $T_{n,\beta}$, яка за умови виконання нульової гіпотези про нормальній розподіл, має логнормальний розподіл. У випадку, коли $Q = I$, цей тест має вигляд (див. [HZ90]):

$$n \int_{\mathbb{R}^d} |\Phi_n(t) - \Psi(t)|^2 w_\beta(t) dt,$$

де

$$\Phi_n(t) = \frac{1}{n} \sum_{j=1}^n e^{it' \mathbf{Y}_{jn}}, \quad \Psi(t) = e^{-\|t\|^2/2}, \quad (2\pi\beta^2)^{-d/2} e^{-\frac{\|t\|^2}{2\beta^2}},$$

де $\Phi_n(t)$, $\Psi(t)$, $w_\beta(t)$ є, відповідно, емпіричною функцією розподілу $N_p(0, I_p)$, характеристичною функцією $N_p(0, I_p)$, та ваговою функцією з параметром $\beta > 0$.

На практиці, і тест Андерсона-Дарлінга, і тест Хенце-Циклера використовують не в інтегральній формі, а у вигляді суми, яку отримують після інтегрування.

Явний вигляд цих тестів, а також порівняння тестів, можна знайти в [SW03], [EH21], а також в [RW11], [YA07].

Застосування тестів Мардія, Хенце-Циклера та Андерсона-Дарлінга за допомогою пакету **MVN** (детальні пояснення можна подивитися за посиланням):

```
mvn(data,
subset = NULL,
mvnTest = "hz",
covariance = TRUE,
tol = 1e-25,
alpha = 0.5,
scale = FALSE,
desc = TRUE,
transform = "none",
R = 1000,
univariateTest = "AD",
univariatePlot = "none",
multivariatePlot = "none",
```

```

multivariateOutlierMethod = "none",
bc = FALSE,
bcType = "rounded",
showOutliers = FALSE,
showNewData = FALSE
)

```

За замовченнем використовується тест Хенце-Циклера, для тесту Мардія треба обрати `mvnTest="mardia"`. Для одновимірних тестів можна обирати `univariateTest = "SW"` для тесту Шапіро-Уілка, за замовченнем стоять тест Андерсона-Дарлігна ("AD").

У пакеті `mvnTest` є тести Хенце-Циклера, багатовимірний варіант тесту Андерсона-Дарлігна, та багато іншого:

```

AD.test(data, qqplot = FALSE) # Anderson-Darling test
HZ.test(data, qqplot = FALSE) # Henze-Zikler test

```

3 Оцінки максимальної вірогідності (OMB) Тестування середнього

Література: [PP12, (108)], [R02], [RS08].

Нехай $\mathbf{x} = (\xi_1, \dots, \xi_n)$ – нормальню розподілена вибірка розміру n з середнім μ та дисперсією σ^2 . Оцінки для параметрів μ та σ можна отримати за допомогою методу максимальної вірогідності. Так, ОМВ для μ є вибікове середнє

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^n \xi_i.$$

OMB для σ^2 є зміщеною оцінкою

$$\bar{s}^2 = \frac{1}{n} \sum_{i=1}^n (\xi_i - \bar{\mu})^2. \quad (3.1)$$

Позначимо s^2 через також незміщену оцінку

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (\xi_i - \bar{\mu})^2,$$

а також через \tilde{s}^2 оцінку дисперсії при відомому μ :

$$\tilde{s}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \mu)^2 \quad (3.2)$$

В багатовимірному випадку, тобто коли $\mathbf{x}_i \sim N_p(\boldsymbol{\mu}, \Sigma)$, $i = 1, \dots, n$, також можна знайти ОМВ для невідомого параметру $\theta = (\boldsymbol{\mu}, \Sigma)$. Надалі ми будемо позначати через $X \sim N_p(\boldsymbol{\mu}, \Sigma)$ той факт, що $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ є вибіркою розміру n випадкових векторів $\mathbf{x}_i \sim N_p(\boldsymbol{\mu}, \Sigma)$, $i = 1, \dots, n$.

Нагадаємо, що $\bar{\mathbf{y}}$ позначає вектор середніх значень: $\bar{\mathbf{y}} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i$. Запишемо функцію вірогідності:

$$\begin{aligned} L(\mathbf{y}, \theta) &= \prod_{i=1}^n f(\mathbf{y}_i, \theta) \\ &= \prod_{i=1}^n \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{y}_i - \boldsymbol{\mu})} \\ &= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{1}{2} \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu} \pm \bar{\mathbf{y}})' \Sigma^{-1} (\mathbf{y}_i - \boldsymbol{\mu} \pm \bar{\mathbf{y}})} \\ &= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{1}{2} [\sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \Sigma^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}) + n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \Sigma^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})]}. \end{aligned}$$

Оскільки матриця Σ є додатно визначену, то Σ^{-1} також додатно визначена, тому

$$(\bar{\mathbf{y}} - \boldsymbol{\mu})' \Sigma^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu}) > 0.$$

Тоді

$$\begin{aligned} L(\mathbf{y}, \theta) &= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{1}{2} [\sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \Sigma^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}) + n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \Sigma^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})]} \\ &\leq \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{1}{2} \sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \Sigma^{-1} (\mathbf{y}_i - \bar{\mathbf{y}})}. \end{aligned}$$

Отже, максимум функції $L(\mathbf{y}, \theta)$ досягається при $\boldsymbol{\mu} = \bar{\mathbf{y}}$.

Для того, щоб знайти ОМВ для матриці коваріацій, використаємо наступні результати з лінійної алгебри [PP12].

Теорема 3.1. *Існує $\mathbf{x} \in \mathbb{R}^p$, A - матриця розмірності $p \times p$. Тоді*

$$\mathbf{x}' A \mathbf{x} = \text{trace} [\mathbf{x}' A \mathbf{x}] = \text{trace} [A \mathbf{x} \mathbf{x}'].$$

Теорема 3.2. *Похідна сліду добутку двох матриць дорівнює*

$$\frac{\partial}{\partial A} \text{trace} [AB] = B'.$$

Дійсно,

$$\frac{\partial}{\partial a_{ij}} \text{trace} [AB] = \frac{\partial}{\partial a_{ij}} \sum_k \sum_l a_{kl} b_{lk} = b_{ji}.$$

Теорема 3.3. *Похідна визначника матриці по цій матриці дорівнює*

$$\frac{\partial}{\partial A} |A| = |A| (A^{-1})'.$$

Застосовуючи останній результат, можемо обчислити похідну від логарифму:

$$\frac{\partial}{\partial A} \ln |A| = \frac{1}{|A|} \frac{\partial}{\partial A} |A| = \frac{1}{|A|} |A| (A^{-1})' = (A^{-1})' = (A')^{-1}.$$

Тепер ми можемо отримати ОМВ для коваріаційної матриці.

Використовуючи твердження Теореми 3.1 та підставляючи ОМВ для середнього $\boldsymbol{\mu} = \bar{\mathbf{y}}$, маємо

$$\begin{aligned} L(\mathbf{y}, \theta) \Big|_{\mu=\bar{\mathbf{y}}} &= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{1}{2} [\sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \Sigma^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}) + n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \Sigma^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})]} \Big|_{\mu=\bar{\mathbf{y}}} \\ &= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{n}{2} \sum_{i=1}^n \text{trace} [\Sigma^{-1} \frac{1}{n} (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}})']} \\ &= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{n}{2} \text{trace} [\Sigma^{-1} \bar{\Sigma}]}, \end{aligned}$$

де $\bar{\Sigma}$ є багатовимірним аналогом \bar{s}^2 :

$$\bar{\Sigma} = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}})'.$$

Знайдемо максимум функції вірогідності $L(\mathbf{y}, \theta)$, продиференціювавши її логарифм по Σ^{-1} . За теоремами 3.2 та 3.3 отримаємо наступне:

$$\begin{aligned} \frac{\partial}{\partial \Sigma^{-1}} \ln L(\mathbf{y}, \theta) &= \frac{\partial}{\partial \Sigma^{-1}} \ln \left(\frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{n}{2} \text{trace} [\Sigma^{-1} \bar{\Sigma}]} \right) \\ &= \frac{n}{2} \frac{\partial}{\partial \Sigma^{-1}} \ln |\Sigma^{-1}| - \frac{n}{2} \frac{\partial}{\partial \Sigma^{-1}} \text{trace} [\Sigma^{-1} \bar{\Sigma}] \\ &= \frac{n}{2} \left[\left((\Sigma^{-1})^{-1} \right)' - \bar{\Sigma}' \right] = \frac{n}{2} \left[\Sigma - \bar{\Sigma}' \right] = 0. \end{aligned} \quad (3.3)$$

Оскільки матриця $\bar{\Sigma}$ є симетричною, то $\bar{\Sigma} = \bar{\Sigma}'$, тоді з останнього співвідношення максимум функції вірогідності досягається при $\Sigma = \bar{\Sigma}$. Таким чином, отримали наступні ОМВ для $\theta = (\boldsymbol{\mu}, \Sigma)$:

$$\bar{\mathbf{y}} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i, \quad \bar{\Sigma} = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}}'). \quad (3.4)$$

3.1 Розподіли $\bar{\mathbf{y}}$ та S

В одновимірному випадку $\bar{y} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$.

$$\mathbb{E}\bar{y} = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n y_i \right) = \mu, \quad \mathbb{D}\bar{y} = \mathbb{D} \left(\frac{1}{n} \sum_{i=1}^n y_i \right) = \frac{1}{n^2} \sum_{i=1}^n \mathbb{D}y_i = \frac{\sigma^2}{n}.$$

Аналогічно, для довільного $p > 1$

$$\bar{\mathbf{y}} \sim N_p \left(\boldsymbol{\mu}, \frac{1}{n} \Sigma \right).$$

Для оцінки дисперсії при відомому середньому (3.2) в одновимірному випадку маємо:

$$\frac{n\tilde{s}^2}{\sigma^2} := \sum_{i=1}^n \left(\frac{y_i - \mu}{\sigma} \right)^2 \sim \chi_n^2. \quad (3.5)$$

Якщо μ – невідоме, то при $p = 1$ для незміщеної оцінки дисперсії s^2 при невідомому середньому маємо

$$\frac{(n-1)s^2}{\sigma^2} = \sum_{i=1}^n \left(\frac{y_i - \bar{\mu}}{\sigma} \right)^2 \sim \chi_{n-1}^2. \quad (3.6)$$

Аналог розподілу хі-квадрат в багатовимірному випадку є розподіл Уішарта. Щільність ймовірності розподілу Уішарта відома, див. [PP12], причому аргументом цього розподілу є матриця!

Наведемо без доведення аналоги (3.5) та (3.6):

$$n\tilde{\Sigma}_n := \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu}) (\mathbf{y}_i - \boldsymbol{\mu})' \sim W_p(n, \Sigma),$$

$$(n-1)S := \sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}}) (\mathbf{y}_i - \bar{\mathbf{y}})' \sim W_p(n-1, \Sigma),$$

де $W_p(n, \Sigma)$ – розподіл Уішарта в \mathbb{R}^p з n ступенями свободи та відомою матрицею коваріації.

При оцінювання параметрів нормального розподілу та тестуванні гіпотез ключову роль відіграє лема Фішера. Зокрема, за цією лемою, вибіркове середнє та вибіркова дисперсія – незалежні випадкові величини, частка яких має певний (відомий) розподіл.

Пригадаємо лему Фішера в одновимірному випадку.

Лема 3.1. (Фішер) *Нехай $y \sim N(\mu, \sigma^2)$, тоді*

1) вибіркове середнє \bar{y} та вибіркова дисперсія s^2 – незалежні;

2) нормоване відношення вибіркового середнього \bar{y} та вибіркової дисперсії s^2 має розподіл Стьюдента з $n-1$ ступенями свободи:

$$\frac{\sqrt{n}(\bar{y} - \mu)}{\sqrt{s^2}} \sim t_{n-1}.$$

У багатовимірному випадку маємо схоже твердження. Розглянемо квадрат статистики t_{n-1} :

$$t_{n-1}^2 = n(\bar{y} - \mu)(s^2)^{-1}(\bar{y} - \mu).$$

Праву частину можна легко узагальнити на багатовимірний випадок.

Лема 3.2. *Нехай $Y \sim N_p(\boldsymbol{\mu}, \Sigma)$. Тоді*

1) вибіркове середнє $\bar{\mathbf{y}}$ та вибіркова матриця коваріації S – незалежні,

$$(n-1)S \sim W_p(n-1, \Sigma), \quad \bar{\mathbf{y}} \sim N_p\left(\boldsymbol{\mu}, \frac{1}{n}\Sigma\right);$$

2) Статистика T^2 має розподіл Хотеллінга $T_{p,n-1}^2$ з p та $n-1$ ступенями свободи:

$$T^2 := n(\bar{\mathbf{y}} - \boldsymbol{\mu})' S^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu}) \sim T_{p,n-1}^2. \quad (3.7)$$

Нагадаємо, що вибіркова коваріаційна матриця завжди додатно напіввизначена, а отже, статистика T^2 невід’ємна.

Розподіл Хотеллінга ще називають багатовимірним розподілом Стьюдента, або багатовимірним t -розподілом. Вигляд щільності можна теж знайти в [PP12].

Зауважимо також, що $t_n^2 \sim F_{1,n}$, де $F_{1,n}$ є розподілом Фішера з 1 та n ступенями свободи. В багатовимірному випадку нормований розподіл Хотеллінга має розподіл Фішера [Но31]:

$$\frac{n-p+1}{np} T_{p,n}^2 \sim F_{p,n-p+1}. \quad (3.8)$$

Тоді нормована статистика Хотеллінга має наступний розподіл:

$$F := \frac{n-p}{(n-1)p} T^2 \sim F_{p,n-p}. \quad (3.9)$$

Таке співвідношення важливо, оскільки дозволяє не шукати квантилі розподілу Хотеллінга, що набагато складніше, а використовувати простійші в практичному застосуванні розподіл Фішера.

4 Перевірка гіпотези про середнє

Література: [R02], [RS08]. Пакети **Hotelling**, **ICSNP**.

У цьому розділі ми будемо перевіряти гіпотезу про середнє значення вектору нормального розподілу у випадку, коли коваріаційна матриця а) відома б) невідома. Також, ми перевіримо гіпотезу про те, що середні двох виборок співпадають.

Така потреба в перевірці гіпотези про середнє виникає, наприклад, коли треба перевірити, що середнє значення спостережень дорівнює певному фіксованому μ_0 . Необхідність перевірки гіпотези про рівність середніх в двох вибірках виникає, наприклад, в клінічних дослідженнях, коли потрібно перевірити значення показників до терапії та після.

Перед тим, як перейти безпосередньо до тестування, подивимось, як відрізняється перевірка гіпотези про середнє по кожному параметру та для всього векторного параметру.

Зрозуміло, що при збільшенні розмірності простору кількість невідомих параметрів зростає. Припустимо, що ми проводимо $p = 10$ одновимірних тестів перевірки гіпотези H_0 , в яких похибка першого роду не перевищує $\alpha = 0.05$. Тоді

$$\begin{aligned} \mathbb{P}(\text{при найменні 1 відхилення}) &= 1 - \mathbb{P}(\text{при найменні 1 відхилення } H_0) \\ &= 1 - (1 - \alpha)^p = 1 - (0.95)^{10} = 0.40. \end{aligned}$$

Тобто ми маємо відхиляти основну гіпотезу в 40 випадках із 100. З іншого боку, якщо ми проводимо багатовимірний тест, то ми маємо рівень відхилення $\alpha = 0.05$, виходячи з багатовимірного розподілу тестової статистики, що значно краще- тепер ми відхиляємо основну гіпотезу в 5 випадках зі 100.

Зауважимо також, що одновимірні тести повністю ігнорують кореляцію між змінними, в той час як багатовимірні тести її враховують.

Багатовимірні тести мають більшу потужність ніж одновимірні. Нагадаємо, що потужність тесту – це ймовірність відхилити H_0 тоді, коли вона хибна (або: це $1 - \beta$, де $\beta = \mathbb{P}(H_1|H_0)$ – це похибка 2го роду). Ми кажемо, що тест є значущим (significant), якщо в цьому тесті $p-value < 0.05$. Може статися так, що покоординатний тест не буде значущим, але багатовимірний тест буде, завдяки тому, що малі ефекти по кожній змінній акумулюють. Іншими словами: координати, які при окремих тестах не є значущими (тобто на яких ми прийняли нульову гіпотезу), в сукупності є значущими (тобто по їх сукупності ми відхиляємо нульову гіпотезу). Але є обмеження на кількість змінних та розмірність, див. (4.6) (див. також [R02, §5.3]).

В нашому курсі ми будемо часто використовувати вбудовані функції для того, щоб перевірити ту чи іншу гіпотезу. Як результат, ми отримуємо $p-value$, виходячи з якого ми маємо прийняти чи відхилити основну гіпотезу. Тому пригадаємо, що таке $p-value$ на прикладі, коли тестова статистика T має, наприклад, розподіл Стьюдента. Припустимо, що альтернативна гіпотеза двостороння. Нехай t_α – квантиль рівня $1 - \alpha/2$. Ми приймаємо основну гіпотезу, якщо

$$|T| < t_\alpha. \quad (4.1)$$

За означенням, у цьому випадку двосторонньої альтернативи,

$$p-value = 2\mathbb{P}(|t| > |T|).$$

Але $2\mathbb{P}(|t| > t_\alpha) = 0.05$, отже,

$$2\mathbb{P}(|t| > |T|) > 0.05 = 2\mathbb{P}(|t| > t_\alpha) \iff |T| < t_\alpha. \quad (4.2)$$

Іншими словами, $p-value$ – це ймовірність попасти в той інтервал, куди за умови основної гіпотези випадкова величина мала б попасти з імовірністю $1 - \alpha$. Як правило, обирають $\alpha = 0.05$, отже така імовірність становить 0.95. Або, іншими словами, в 19 з 20 випадків ми мали б попасти в ”хороший” інтервал. Якщо ми туди все ж таки не попали, то мабуть основна гіпотеза хибна і ми її відхиляємо.

4.1 Гіпотеза про середнє. Відома коваріаційна матриця

Нехай є нормальна вибірка $Y \sim N_p(\boldsymbol{\mu}, \Sigma)$, де відома коваріаційна матриця. Перевіримо гіпотезу

$$H_0: \boldsymbol{\mu} = \boldsymbol{\mu}_0 \text{ проти альтернативи } H_1: \boldsymbol{\mu} \neq \boldsymbol{\mu}_0.$$

Зauważимо, що тут $\boldsymbol{\mu}_0, \boldsymbol{\mu}$ є векторами, тобто H_0 виконано, коли всі координати вектора $\boldsymbol{\mu}$ співпадають з координатами вектора $\boldsymbol{\mu}_0$, а H_1 – принаймні 2 координати відрізняються.

Використаємо наступну статистику:

$$Z^2 = n(\bar{\mathbf{y}} - \boldsymbol{\mu}_0)' \Sigma^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu}_0) = n\Delta^2 \quad (4.3)$$

(нагадаємо, що Δ^2 є квадратом відстані Махalanobіса).

За умови виконання H_0 ,

$$Z^2 \sim \chi_p^2. \quad (4.4)$$

Якщо $\boldsymbol{\mu} \neq \boldsymbol{\mu}_0$, то значення Δ^2 велике, оскільки середнє $\bar{\mathbf{y}}$ буде відрізнятися від $\boldsymbol{\mu}_0$, а отже, Z^2 велике. Іншими словами, критичними значеннями є великі значення Z^2 . Отже, алгоритм перевірки гіпотези H_0 є наступним.

- Обчислюємо Z^2 ;
- Обчислюємо квантиль χ_p^2 розподілу рівня $1 - \alpha$, тобто $\chi_{\alpha,p}^2$, де α задане, наприклад, $\alpha = 0.05$.
- Якщо $Z^2 > \chi_{\alpha,p}^2$, то відхиляємо H_0 .

Нехай $p = 2$. Розглянемо, як будуть відрізнятися області прийняття нульової гіпотези при використанні статистики Z^2 та при використанні одновимірних тестів. Графічно, ми приймаємо нульову гіпотезу у випадку, коли тестове середнє значення попадає в середину еліпса з центром в т. μ_0 , орієнтованим вздовж власних векторів матриці Σ та пів-осями довжини σ_1 та σ_2 , де σ_1^2 та σ_2^2 діагональні елементи (власні числа) матриці Σ :

$$(\bar{\mathbf{y}} - \mu_0)' \Sigma^{-1} (\bar{\mathbf{y}} - \mu_0) = \frac{\chi_{\alpha,p}^2}{n}$$

(тобто відстань Махalanобіса та менша за $\sqrt{\chi_{\alpha,p}^2/n}$). Якщо ми проводимо одновимірні тести, до допустимі області для вибіркових середніх \bar{y}_i мають вигляд

$$\mu_{0,i} - z_{\alpha/2} \frac{\sigma_1}{\sqrt{n}} \leq \bar{y}_i \leq \mu_{0,i} + z_{\alpha/2} \frac{\sigma_1}{\sqrt{n}}, \quad i = 1, 2,$$

де $z_{\alpha/2}$ – квантиль рівня $1 - \alpha/2$ стандартного нормального розподілу $N(0, 1)$, тобто це розв'язок рівняння $P(N(0, 1) > z_{\alpha/2}) = \alpha/2$. В цьому випадку допустима область – це прямокутник

$$[\mu_{0,1} - z_{\alpha/2}, \mu_{0,1} + z_{\alpha/2}] \times [\mu_{0,2} - z_{\alpha/2}, \mu_{0,2} + z_{\alpha/2}].$$

Зобразимо ці області графічно, див. Рисунок 3. Як ми бачимо, ці області хоча і перетинаються, але не співпадають. Тобто може бути так, що ми приймаємо нульову гіпотезу, якщо керуємося одновимірними тестами, але відхиляємо, якщо використовуємо багатовимірний, і навпаки. Такий ефект має назву *парадокс Rao*.

Взагалі, ймовірність потрапляння в область $\{x : |x - \mu_0| \leq \sigma\}$ катастрофічно змінюється зі збільшенням розмірності. При $p = 1, 2, 3, 4$ ймовірності потрапляння в еліпсоїд $\{\mathbf{x} : \mathbf{x}' \Sigma^{-1} \mathbf{x} = 1\}$ та потрапляння в p -вимірний куб $\{\mathbf{x} : |x_i| \leq \sigma_i\}$, де σ_i – діагональні елементи матриці Σ , змінюються наступним чином:

$$p = 1 : 0.68 = 0.68$$

$$p = 2 : 0.39 < (0.68)^2 = 0.4624$$

$$p = 3 : 0.20 < (0.68)^3 = 0.314432$$

$$p = 4 : 0.09 < (0.68)^4 = 0.2138138$$

З цього випливає, що перевірка нульової гіпотези про середнє значення покоординатно буде значно частіше приводити до прийняття нульової гіпотези, в той час як перевірка за допомогою статистики Z^2 (див. (4.4)) буде частіше приводити до її відхилення.

4.2 Гіпотеза про середнє. Невідома коваріаційна матриця

Нехай тепер матриця Σ невідома. У цьому випадку використаємо для перевірки гіпотези H_0 статистику Хотеллінга T^2 (див. (3.7)). За умови виконання H_0 ,

$$T^2 \sim T^2_{p,n-1} \tag{4.5}$$

тобто T^2 має розподіл Хотеллінга з p та $n - 1$ ступенями свободи. Так само, критичними значеннями статистики є велиці значення. Зауважимо, що має виконуватися нерівність

$$n - 1 > p, \tag{4.6}$$

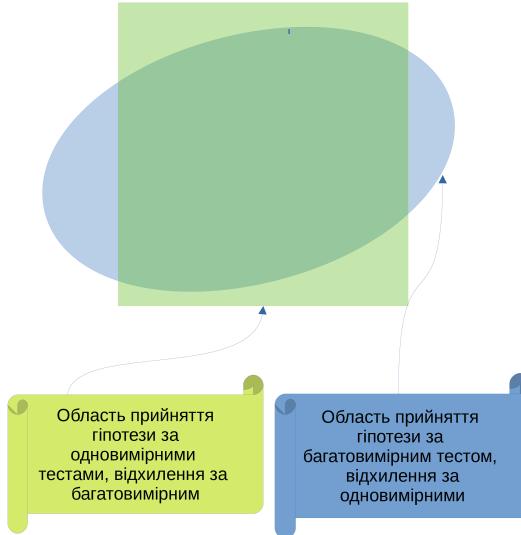


Рис. 3: Парадокс Рао

інакше матриця S , що входить до статистики T^2 , є сингулярною, а отже, ми не можемо обчислити статистику.

Алгоритм перевірки H_0 є наступним.

- Обчислюємо T^2 .
- Обчислюємо квантіль $T_{\alpha,p,n-1}^2$ рівня $1 - \alpha$ розподілу $T_{p,n-1}^2$.
- Якщо $T^2 > T_{\alpha,p,n-1}^2$, то відхиляємо H_0 .

Перевірку гіпотези можна також виконати за допомогою розподілу Фішера. Для цього використаємо (3.9), тобто за умови виконання H_0 випадкова величина F має розподіл Фішера з відхиляємо H_0 , якщо $F > F_{\alpha,p,n-p}$.

Розглянемо ще один варіант гіпотези H_0 , а саме,

$$H_0: \mu_1 = \mu_2 = \dots = \mu_p, \text{ проти альтернативи } H_1: \exists i, j: \mu_i \neq \mu_j.$$

У векторному випадку гіпотезу H_0 можна сформулювати наступним чином:

$$H_0 : \begin{pmatrix} \mu_1 - \mu_2 \\ \mu_2 - \mu_3 \\ \vdots \\ \mu_{p-1} - \mu_p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Альтернативно, гіпотезу H_0 можна задати так:

$$H_0 : \begin{pmatrix} \mu_1 - \mu_2 \\ \mu_1 - \mu_3 \\ \dots \\ \mu_1 - \mu_p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

Взагалі, гіпотезу H_0 можна сформулювати наступним чином:

$$H_0: C\boldsymbol{\mu} = 0, \text{ проти альтернативи } H_1: C\boldsymbol{\mu} \neq 0,$$

де C – така матриця, сума в кожному рядку якої дорівнює 0 та $\text{rank } C = p - 1$. Така матриця називається *матрицею контрастів*. Перетворення можна також використати при перевірці гіпотези про коваріаційну матрицю.

4.3 Гіпотеза про рівність середніх двох виборок

Для простоти розглянемо спочатку одновимірний випадок.

Нехай є 2 виборки $Y_1 = (y_{11}, y_{12}, \dots, y_{1n_1})$ та $Y_2 = (y_{21}, y_{22}, \dots, y_{2n_2})$, $y_{1i} \sim N(\mu_1, \sigma_1^2)$, $1 \leq i \leq n_1$, $y_{2j} \sim N(\mu_2, \sigma_2^2)$, $1 \leq j \leq n_2$. Припустимо, що виборки незалежні та $\sigma_1^2 = \sigma_2^2 = \sigma^2$ невідоме.

Оскільки виборки можуть бути різного розміру, розглянемо *зважену дисперсію* (Eng.: pooled variance):

$$s_{pl}^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

$$s_i^2 := \sum_{k=1}^{n_i} \frac{(y_{ik} - \bar{y}_i)^2}{n_i - 1}, \quad \bar{y}_i := \frac{1}{n_i} \sum_{k=1}^{n_i} y_{ik}, \quad i = 1, 2.$$

Тоді s_{pl}^2 – незміщена оцінка дисперсії σ^2 : $\mathbb{E}s_{pl}^2 = \sigma^2$.

Перевіримо основну гіпотезу

$$H_0: \mu_1 = \mu_2 \text{ проти альтернативи } H_1: \mu_1 \neq \mu_2.$$

Розглянемо

$$t := \frac{\bar{y}_1 - \bar{y}_2}{s_{pl} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}.$$

За умови виконання H_0 , ця статистика має розподіл Стьютента $t_{n_1+n_2-2}$ з $n_1 + n_2 - 2$ ступенями свободи. Тому для перевірки гіпотези H_0 можна використати тестову статистику $|t|$. Якщо $|t| > t_{\alpha/2, n_1+n_2-2}$, де $t_{\alpha/2, n_1+n_2-2}$ квантіль рівня $1 - \alpha/2$ розподілу Стьюдента з $n_1 + n_2 - 2$ ступенем свободи, то H_0 відхиляємо.

Альтернативно, використовуючи наступне співвідношення між розподілом Стьюдента та розподілом Фішера $t_{n_1+n_2-2}^2 = F_{1, n_1+n_2-2}$, можна перевіряти гіпотезу H_0 , обчисливши відповідну статистику та знайшовши квантіль рівня розподілу Фішера.

Розглянемо багатовимірний випадок. Тепер елементи \mathbf{y}_{ik} виборок Y_1, Y_2 – це вектори, які мають нормальній розподіл $N_p(\boldsymbol{\mu}_i, \Sigma_i)$, $i = 1, 2$, відповідно. Припустимо, що коваріаційні матриці співпадають: $\Sigma_1 = \Sigma_2$ (трохи згодом ми розглянемо, як перевіряти гіпотезу про рівність коваріаційних матриць).

Ми перевіримо основну гіпотезу $H_0: \mu_1 = \mu_2$ проти альтернативи $H_1: \mu_1 \neq \mu_2$. Позначимо через

$$S_i := \frac{1}{n_i - 1} \sum_{k=1}^{n_i} (\mathbf{y}_{ik} - \bar{\mathbf{y}}_i)(\mathbf{y}_{ik} - \bar{\mathbf{y}}_i)', \quad i = 1, 2,$$

вибіркові коваріаційні матриці у першій та другій вибірці, відповідно, та через

$$S_{pl} := \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 2}$$

зважену коваріаційну матрицю (Eng.: pooled covariance). Тестовою статистикою є статистика Хотеллінга (тут розміри виборок можуть бути різними!)

$$T^2 = \frac{n_1 n_2}{n_1 + n_2} (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)' S_{pl}^{-1} (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2). \quad (4.7)$$

Якщо справедлива гіпотеза H_0 , то статистика T^2 має розподіл Хотеллінга $T_{p, n_1+n_2-2}^2$ з n_1 та n_2 ступенями свободи. Ми відхиляємо гіпотезу H_0 , якщо $T^2 < T_{\alpha, p, n_1+n_2-2}^2$, де $T_{\alpha, p, n_1+n_2-2}^2$ є квантілем розподілу $T_{p, n_1+n_2-2}^2$ рівня $1 - \alpha$.

Як і в одновимірному випадку, можна перевірити гіпотезу H_0 , використовуючи наступне співвідношення між статистикою Хотеллінга та розподілом Фішера (див. [R02, (5.7)]):

$$F := \frac{n_1 + n_2 - p - 1}{(n_1 + n_2 - 2)p} T_{p, n_1+n_2-2}^2 \sim F_{p, n_1+n_2-p-1}. \quad (4.8)$$

Якщо гіпотеза H_0 є хибою для векторного випадку, тоді має сенс перевірити одновимірний варіант цієї гіпотези покоординатно.

4.4 Приклади

1. Розглянемо приклад тесту про середнє при відомій дисперсії (див. [R02, Ex.5.2.2]). В Таблиці 3.1 наведено зріст та вагу (в дюймах та фунтах) учнів коледжу.¹ Переміримо гіпотезу про те, що середній зріст учнів 70 дюймів, а середня вага – 170 фунтів (в європейській системі це, відповідно, 177,8 см. та 77,1 кг). Зчитуємо Таблицю 3.1 (див. [R02, Ch.3, p.45]) та виведемо перші 6 рядочків. Якщо Ви використовуєте colab, треба спочатку, наприклад, приєднати google disc за допомогою функції

```
from google.colab import drive
drive.mount('/content/drive')
```

в середовищі python, тобто переключити середовище на python, а потім знову на R. Після цього треба вказати шлях до того файлу таблиці, який вже знаходиться на google disc. Шлях вказується той, по якому ми знаходимо саме в colab. Якщо ж використовується Jupyter, то досить покласти файл в ту саму папку, що і робочий файл.

```
Tab_3_1<-read.table("T3_1_HEIGHTWT.DAT")
head(Tab_3_1)
```

¹ 1 дюйм (1in)=2,54cm, та 1 фунт (1lb)= 0,453кг.

V1	V2	V3
1	69	153
2	74	175
3	68	155
4	70	135
5	72	172
6	67	150

Припустимо, що ми знаємо коваріаційну матрицю E та припускаємо, що вектор середніх – це $\mu_0 = (70, 170)$. Задамо ці величини:

```
E<-matrix(c(20,100,100,1000), nrow = 2, byrow = FALSE)
mu0<-c(70,170)
```

Зчитаємо вектори $V2$ та $V3$ та знайдемо вектор середніх значень:

```
X1<-Tab_3_1$V2
X2<-Tab_3_1$V3
xbar<-c(mean(X1), mean(X2))
```

71.45 164.70

Позначимо через m та p кількість людей та розмірність простору, відповідно:

```
Tab_dim = dim(Tab_3_1)
m = Tab_dim[1]
p = Tab_dim[2] - 1
m
p
```

20
2

Обчислюємо Z^2 , див. (4.3).

```
Z2<-m*(t(xbar-mu0) * solve(E) * (xbar-mu0))
```

8.4026

А можна рахувати будованою функцією. Використаємо відстань Махalanобіса між вектором середніх $xbar$ та $mu0$:

```
m*mahalanobis(xbar, mu0, E)
```

8.4026

Рахуємо квантиль рівня 0.95 хі-квадрат родподілу з 2-ма ступенями свободи:

```
qchisq(0.05, df = p, lower.tail = FALSE, log.p = FALSE)
```

5.991465

Перевіримо основну гіпотезу $H_0: \mu = \mu_0$. Оскільки $Z2 > qchisq(...)$, ми відхиляємо гіпотезу H_0 ,

Перевіримо тепер гіпотезу про середнє за допомоги вбудованої функції, а саме, тесту Хотеллінга. Для цього потрібно встановити пакет **Hotelling**.

```
install.packages("Hotelling")
library(Hotelling)
```

Виконаємо тест (при цьому, ми вважаємо, що коваріаційна матриця невідома).

```
xbar=as.matrix(xbar)
mu0=as.matrix(mu0)
P=hotelling.test(xbar, mu0, shrinkage = FALSE, var.equal = TRUE)
P
```

```
Test stat:      0.00079284
Numerator df:   1
Denominator df: 2
P-value:        0.9801
```

Параметр `shrinkage` в пакеті використовується у випадках, коли розмір вибірки малий порівняно з розмірністю простору. У цьому випадку матриця S , яку ми використовували для оцінювання коваріаційної матриці, не є гарним наближенням коваріаційної матриці. Тому використовується інший метод обчислення коваріації; більш детально про цей метод можна почитати в документації к пакету **Hotelling**. У нашому випадку ми використовуємо матрицю S , тому обираємо `shrinkage = FALSE`. Оскільки ми маємо справу з однією вибіркою, то ми обираємо `var.equal = TRUE`.

2. Розглянемо тепер тест про середнє при невідомій дисперсії. В Таблиці 3.3 наведені вимірювання концентрації кальцію в землі та в листі ріпи в 10 різних локаціях. Зчитуємо Таблицю 3.3 (див. Rencher, Chapter 3, p.56).

```
Tab_3_3<-read.table("T3_3_CALCIMUM.DAT")
```

	V1	V2	V3	V4
1	35	3.5	2.80	
2	35	4.9	2.70	
3	40	30.0	4.38	
4	10	2.8	3.21	
5	6	2.7	2.73	
6	20	2.8	2.81	
7	35	4.6	2.88	

```

8   35  10.9  2.90
9   35  8.0   3.28
10  30   1.6   3.20

```

Вводимо вектори V_2, V_3, V_4 та утворимо з них матрицю M (в наступному прикладі ми розглянемо більш простий спосіб утворити матрицю з векторів таблиці та порахувати їх середні значення):

```

Y1 <- Tab_3_2$V2
Y2 <- Tab_3_2$V3
Y3 <- Tab_3_2$V4
M <- cbind(Y2, Y3, Y4)

```

Знаходимо коваріаційну матрицю, що пов'язана з M та вектор середніх значень:

```

cov(M)
ybar <- c(mean(Y1), mean(Y2), mean(Y3))

```

```
28.100 7.180 3.089
```

Задамо вектор середніх μ_0 , який будемо використовувати для перевірки H_0 :

```

mu0 <- c(15.0, 6.0, 2.85)
n <- c(dim(M))[1]
p <- c(dim(M))[2]

```

Обчислюємо статистику Хотеллінга:

```

T2 <- n*t(ybar-mu0) %*% solve(cov(M)) %*% (ybar-mu0)
T2

```

```
24.55891
```

А можна рахувати вбудованою функцією. Для цього використаємо відстань Махalanобіса між вектором середніх і μ_0 :

```
n*mahalanobis(ybar, mu0, cov(M))
```

```
24.55891
```

А тепер обчислимо статистику F , наведену в (3.9):

```

F <- T2 * (n-p) / ((n-1)*p)
F

```

```
6.367124
```

F має розподіл Фішера $F_{p,n-p}$. Обчислюємо квантіль рівня 0.05 розподілу $F_{3,10-3} = F_{3,7}$.

```
qf(0.95, p, n-p, lower.tail = TRUE, log.p = FALSE)
```

4.066181

Відхиляємо H_0 , оскільки F більша за $qf(\dots)$

Можна також застосувати функцію HotellingsT2 з пакету **ICSNP**.

```
HotellingsT2(M, Y = NULL, mu = mu0, test = "chi", na.action = na.fail)
```

Hotelling's one sample T2-test

```
data: M
T.2 = 24.559, df = 3, p-value = 1.909e-05
alternative hypothesis: true location is not equal to c(15,6,2.85)
```

Параметр `chi` означає, що ми використовуємо хі-квадрат розподіл (тобто обчислюємо T^2). А тепер обчислимо HotellingsT2, але використовуючи розподіл Фішера:

```
HotellingsT2(M, Y = NULL, mu = mu0, test = "f", na.action = na.fail)
```

Hotelling's one sample T2-test

```
data: M
T.2 = 6.3671, df1 = 3, df2 = 7, p-value = 0.02068
alternative hypothesis: true location is not equal to c(15,6,2.85)
```

3. Розглянемо тепер перевірку гіпотези про рівність середніх у 2х вибірках.

Розглянемо таблицю 5.1 (див. Rencher, Chapter 5, p.125). В таблиці наведено результати психологічних досліджень в групах з 32 чоловіків та 32 жінок. Зчитаємо таблицю та запишемо дані в змінні *Male* та *Female*.

```
Tab_5_1<-read.table("T5_1_PSYCH.DAT")
head(Tab_5_1)
```

```
V1 V2 V3 V4 V5
1 15 17 24 14
1 17 15 32 26
1 15 14 29 23
1 13 12 10 16
1 20 17 26 28
1 15 21 26 21
```

Надалі нам потрібно працювати з матрицями, отже, ми перетворимо наші дані у матриці.

```
Male <- as.matrix(Tab_5_1[1:32, 2:5])
Female <- as.matrix(Tab_5_1[33:64, 2:5])
```

Будуємо вектори середніх значень та коваріації.

```
Fmean = colMeans(Female)
Mmean = colMeans(Male)
S1 <- cov(Male)
S2 <- cov(Female)
```

Знайдмо кількість елементів у вибірках *Female* та *Male*.

```
n1 <- dim(Male)[1]
n2 <- dim(Female)[1]
p <- dim(Female)[2]
```

Знайдемо pooled covariance.

```
Spl <- ((n1 - 1) * S1 + (n2 - 1) * S2) / (n1 + n2 - 2)
```

Знайдемо тепер *T2* статистику:

```
T2 <- ((n1 * n2) / (n1 + n2)) * mahalanobis(Mmean, Fmean, Spl)
T2
```

97.6015

Рахуємо *F* статистику (4.8):

```
F <- T2 * (n1 + n2 - p - 1) / ((n1 + n2 - 2) * p)
F
```

23.21971

Обчислимо квантіль рівня 0.95 розподілу F_{p,n_1+n_2-p-1} :

```
qf(0.95, p, n1 + n2 - p - 1, lower.tail = TRUE, log.p = FALSE)
```

2.527907

Відхиляємо H_0 , оскільки *F* більша за $qf(\dots)$.

Можна також зробити тест Хотеллінга для двох виборок, використовуючи пакет **ICSNP**.

```
HotellingsT2(Male, Female, test = "chi", na.action = na.fail)
```

```
Hotelling's two sample T2-test
```

```
data: Male and Female  
T.2 = 97.601, df = 4, p-value < 2.2e-16  
alternative hypothesis: true location difference is not equal to c(0,0,0,0)
```

```
HotellingsT2(XT5, YT5, test = "f", na.action = na.fail)
```

```
Hotelling's two sample T2-test
```

```
data: Male and Female  
T.2 = 23.22, df1 = 4, df2 = 59, p-value = 1.464e-11  
alternative hypothesis: true location difference is not equal to c(0,0,0,0)
```

У першому випадку ми використовували тест χ^2 , у другому – тест Фішера. Насправді, треба ще перевірити, що коваріаційні матриці однакові, оскільки це умова використання тесту!

5 Перевірка гіпотези про коваріаційну матрицю

Літкратура: [R02, Гл.7], [Ma40], [Ba51], [Le60], [BF74]. Пакети **stats**, **biotools**.

В цьому розділі ми розглянемо тестування, яке спрямоване на пояснення структури коваріаційної матриці. В свою чергу, структура коваріаційної матриці важлива для інших тестів, які ми будемо використовувати. Можна виділити три основних типи гіпотез: 1) коваріаційна матриця має певну структуру; 2) дві або більше коваріаційних матриць рівні; 3) деякі елементи коваріаційної матриці нульові, що означає некорельованість (а у випадку багатовимірного нормального розподілу і незалежність) даних.

В цьому розділі ми будемо розглядати лише багатовимірний нормальній розподіл, а для побудови тестової статистики будемо використовувати відношення вірогідностей. Саме структура ймовірності багатовимірного нормального розподілу дозволяє побудувати тестову статистику та знайти її розподіл за умови виконання нульової гіпотези. Як правило, мова іде про (нормоване) відношення детермінантів коваріаційних матриць.

5.1 Статистика відношення вірогідностей. Тест про сферичність

Ми скористаємося відношенням вірогідностей $L(\mathbf{y}, \theta)$, $\theta = (\boldsymbol{\mu}, \Sigma)$, причому ми одразу підставимо $\bar{\boldsymbol{\mu}}$ замість $\boldsymbol{\mu}$:

$$L(\mathbf{y}, \bar{\mathbf{y}}, \Sigma) = (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp\left(-\frac{n}{2} \text{trace}(\Sigma^{-1}\bar{\Sigma})\right). \quad (5.1)$$

Підставивши $\bar{\Sigma}$ замість Σ в (5.1), отримаємо

$$L(\mathbf{y}, \bar{\mathbf{y}}, \bar{\Sigma}) = (2\pi)^{-np/2} |\bar{\Sigma}|^{-n/2} \exp\left(-\frac{np}{2}\right).$$

З іншого боку, підставимо в (5.1) $\Sigma = \Sigma_0$ і розглянемо відношення вірогідності (правдоподібності, Likelihood Relation)

$$LR := \frac{L(\mathbf{y}, \bar{\boldsymbol{\mu}}, \bar{\Sigma})}{L(\mathbf{y}, \bar{\boldsymbol{\mu}}, \Sigma_0)} = \left(\frac{|\Sigma_0|}{|\bar{\Sigma}|} \exp(\text{trace}(\Sigma_0^{-1}\bar{\Sigma}) - p) \right)^{\frac{n}{2}}. \quad (5.2)$$

Розглянемо модифіковані статистики відношення вірогідностей (а також замінimo $\bar{\Sigma}$ на незміщену оцінку S):

$$U := \nu [\ln |\Sigma_0| - \ln |S| + \text{trace}(\Sigma_0^{-1}S) - p],$$

$$\tilde{U} := \left[1 - \frac{1}{6\nu - 1} \left(2p + 1 - \frac{2}{p+1} \right) \right] U,$$

де ν – кількість ступенів свободи в S (наприклад, у випадку однієї вибірки $\nu = n - 1$). Сформулюємо без доведення наступні твердження.

Твердження 5.1. *Припустимо, що гіпотеза $H_0: \Sigma = \Sigma_0$ має місце.*

- Для великих n статистика U має наближено $\chi_{\frac{p(p+1)}{2}}^2$ розподiл.
- Для вибірки середнього розміру n статистика \tilde{U} має наближено $\chi_{\frac{p(p+1)}{2}}^2$ розподiл.

Розглянемо частковий випадок H_0 , а саме, гіпотезу про сферичність:

$$H_0: \Sigma = \sigma^2 I_p \text{ проти альтернативи } H_1: \Sigma \neq \sigma^2 I_p.$$

Тут σ^2 є невідомою, тобто H_0 є гіпотезою про форму матриці Σ . При цьому, якщо

$$(\mathbf{y} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu}) = c^2,$$

де $\mathbf{y} = (y_1, \dots, y_p)$, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)$, є рівнянням еліпсу, то за гіпотези сферичності маємо

$$(\mathbf{y} - \boldsymbol{\mu})' (\mathbf{y} - \boldsymbol{\mu}) = \sum_{i=1}^d (y_i - \mu_i)^2 = c^2 \sigma^2,$$

що є рівнянням сфери.

Знайдемо за умови H_0 відношення вірогідностей. Замінимо $\bar{\Sigma}$ на S . Якщо $\Sigma_0 = \sigma^2 I_p$, то $\text{trace}(\Sigma_0^{-1}S) = \sigma^{-2} \text{trace} S$, а $|\Sigma_0| = (\sigma^2)^p = (\text{trace} \Sigma / p)^p$. За гіпотези H_0 , $\text{trace}(S) \approx p\sigma^2$, тому

$$\exp(\text{trace}(\Sigma_0^{-1}S) - p) = \exp(\sigma^{-2} \text{trace}(S) - p) \approx 1.$$

Тому розглядаємо відношення вірогідностей у вигляді

$$LR \approx \left(\frac{|\Sigma_0|}{|S|} \right)^{\frac{n}{2}} = \left(\frac{(\text{trace}(S)/p)^p}{|S|} \right)^{\frac{n}{2}}.$$

Визначимо тестову статистику u наступним чином:

$$-2 \ln LR = n \ln \left(\frac{(\text{trace}(S)/p)^p}{|S|} \right) = -n \ln u,$$

або

$$u = \frac{p^p |S|}{(\text{trace}(S))^p} \quad (5.3)$$

Також розглянемо модифікацію цієї статистики:

$$u' = - \left(n - 1 - \frac{2p^2 + p + 2}{6p} \right) \ln u \quad (5.4)$$

Твердження 5.2. *Припустимо, що гіпотеза H_0 має місце.*

- Для великих n статистика $-n \ln u$ має наближено $\chi_{\frac{p(p+1)}{2}-1}^2$ розподіл.
- Для вибірку середнього розміру n статистика u' має наближено $\chi_{\frac{p(p+1)}{2}-1}^2$ розподіл.

Статистику u' було вперше отримано в роботі [Ma40], тому відповідний статистичний тест носить назив Маучлі (Mauchly test).

Нехай R – кореляційна матриця, та $S = R$. Тоді статистика u має вигляд $u = |S| = |R|$. В роботі Бартлетта ([Ba51], Bartlett test) було розглянуто наступний тест сферичності (по суті, ця статистика схожа на u'):

$$b = -(n - 1 - \frac{2p + 5}{6}) \ln |R|.$$

Твердження 5.3. *Припустимо, що гіпотеза H_0 має місце. Тоді при великих n статистика наближено має розподіл $\chi_{\frac{p(p-1)}{2}}^2$:*

$$b \sim \chi_{\frac{p(p-1)}{2}}^2.$$

Можна також перевіряти гіпотезу про коваріаційну матрицю не для самої вибірки X , а для її перетворення. Розглянемо наступне перетворення X : $\mathbf{z} = C\mathbf{y}$, де C є ортонормованою матрицею констрастів. Тоді Z має розмірність $p - 1$, $\mathbf{z} = (z_1, \dots, z_p) \sim N_{p-1}(0, C\Sigma C')$, $\bar{\mathbf{z}} = C\bar{\mathbf{y}}$, а матриця $S_{\mathbf{z}} = CSC'$ має розмірність $(p - 1) \times (p - 1)$. Маємо:

$$\bar{\mathbf{z}} \sim N_{p-1} \left(0, \frac{C\Sigma C'}{n} \right), \quad T^2 := n\bar{\mathbf{z}}' S_{\mathbf{z}}^{-1} \bar{\mathbf{z}} \sim T_{p-1, n-1}^2.$$

Тоді можна перевіряти гіпотезу

$$H_0: C\Sigma C' = \sigma^2 I_{p-1}, \text{ проти альтернативи } H_1: C\Sigma C' \neq \sigma^2 I_{p-1}.$$

Тестовою статистикою для цього є вищепередана статистика T^2 .

5.2 Тест на рівність коваріаційних матриць.

Розглянемо спочатку одновимірний випадок. В одновимірному випадку, коли ϵ 2 вибірки ($k = 2$) розміру n_1 та n_2 відповідно, розглянемо основну гіпотезу

$$H_0: \sigma_1 = \sigma_2 \text{ проти альтернативи } H_1: \sigma_1 \neq \sigma_2.$$

Тестовою статистикою у цьому випадку є (див. [K07])

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

де s_1^2 та s_2^2 є незміщеними оцінкам дисперсій. Нагадаємо, що s_1 та s_2 є незалежними випадковими величинами (оскільки серії спостережень є незалежними). За умови H_0 , статистика F має розподіл Фішера F_{n_1-1, n_2-1} з $n_1 - 1$ та $n_2 - 1$ ступенями свободи.

У випадку, коли є k серій незалежних спостережень, для перевірки гіпотези

$$H_0 = \sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2,$$

проти альтернативи $\exists i, j : \sigma_i^2 \neq \sigma_j^2$, використовується тест однорідності Бартлетта. Для перевірки гіпотези H_0 використовується статистика

$$\frac{m}{c} \approx \chi_{k-1}^2,$$

де

$$m = \ln \left[\frac{(s^2)^{\sum_{i=1}^k \nu_i}}{(s_1^2)^{\nu_1} (s_2^2)^{\nu_2} \dots (s_k^2)^{\nu_k}} \right], \quad c = 1 + \frac{1}{3(k-1)} \left[\sum_{i=1}^k \frac{1}{\nu_i} - \frac{1}{\sum_{i=1}^k \nu_i} \right], \quad (5.5)$$

$$s^2 = \frac{\sum_{i=1}^k \nu_i s_i^2}{\sum_{i=1}^k \nu_i}, \quad \nu_i = n_i - 1, \quad i = 1, \dots, k. \quad (5.6)$$

Якщо H_0 справедлива, то дріб під знаком логарифму у зображені m близький до 1, тому критичними значеннями статистики $\frac{m}{c}$ є великі значення. Ми відхиляємо гіпотезу H_0 , якщо $\frac{m}{c} > \chi_{\alpha, k-1}^2$, де $\chi_{\alpha, k-1}^2$ квантіль рівня $1 - \alpha$ розподілу χ_{k-1}^2 .

Справедливість гіпотези H_0 можна також перевірити за допомогою розподілу Фішера. Для цього статистику m треба перетворити.

Нехай

$$a_1 = k - 1, \quad a_2 = \frac{k+1}{(c-1)^2}, \quad b = \frac{a_2}{2 - c + 2/a_2}.$$

Тоді статистика наближено має розподіл Фішера F_{a_1, a_2} з a_1 та a_2 ступенями свободи:

$$F := \frac{a_2 m}{a_1(b-m)} \approx F_{a_1, a_2}.$$

Критичними значеннями статистики F є великі значення, тому ми відхиляємо H_0 , якщо $F > F_{\alpha, a_1, a_2}$, де F_{α, a_1, a_2} є квантілем рівня $1 - \alpha$ розподілу F_{a_1, a_2} .

Існують і інші тести на рівність дисперсій. Наприклад, тест Левене (Levene test [Le60]) є більш стійким до відхилення від нормального розподілу (це фактично дисперсійний аналіз ANOVA, див. Розділ 1). Якщо взяти перетворену вибірку, а саме, $z_{ij} = |y_{ij} - \bar{y}_{..}|$, і записати статистику Фішера (8.14) для цієї перетвореної вибірки, отримаємо тест Левене. Якщо брати не середні значення в z_{ij} , а медіани (що має сенс, наприклад, якщо ми досліджуємо, наприклад, не нормальний розподіл, а скончений хі-квадріт розподіл, тобто беремо справжній "центр" розподілу), то отримаємо тест Брауна-Форсайта, [BF74]. Якщо вибірки дійсно мають однакову дисперсію, то така статистика Фішера F має розподіл Фішера $F_{k-1, N-k}$, $N = \sum_{i=1}^k \nu_i = \sum_{i=1}^k (n_i - 1)$. Відповідно, якщо значення F велике, це означає, що якась дисперсія суттєва відрізняється від інших, а отже, нульова гіпотеза про однаковість дисперсій – хибна.

В багатовимірному випадку ситуація схожа, але для перевірки основної гіпотези H_0 ми порівнюємо детермінанти матриць вибіркових коваріацій S_i , $i = 1, \dots, k$.

Нехай є k виборок нормального розподілу, причому i -та вибірка $N_p(\mu_i, \Sigma_i)$ має розмір n_i . Розглянемо основну гіпотезу

$$H_0: \Sigma_1 = \Sigma_2 = \dots = \Sigma_k \text{ проти альтернативи } H_1: \exists i, j: \Sigma_i \neq \Sigma_j.$$

Розглянемо наступну статистику:

$$M := \frac{|S_1|^{\nu_1/2} |S_2|^{\nu_2/2} \dots |S_k|^{\nu_k/2}}{|S_{pl}|^{\sum_{i=1}^k \nu_i/2}}, \quad (5.7)$$

де $\nu_i = n_i - 1$ та $\nu_E = \sum_{i=1}^k \nu_i = \sum_{i=1}^k n_i - k$.

Теорема 5.1. [Test Boxa (Box M-method test)] Нехай

$$c_1 := \left[\sum_{i=1}^k \frac{1}{\nu_i} - \frac{1}{\sum_{i=1}^k \nu_i} \right] \left(\frac{2p^2 + 3p - 1}{6(p+1)(k-1)} \right).$$

Тоді за умови виконання H_0

$$u := -2(1 - c_1) \ln M \sim \chi_{\frac{1}{2}(k-1)p(p+1)}^2. \quad (5.8)$$

Якщо випадку, коли $\nu_1 = \nu_2 = \dots = \nu_k = \nu$,

$$c_1 = \frac{(k+1)(2p^2 + 3p - 1)}{6k\nu(p+1)}. \quad (5.9)$$

5.3 Тест про незалежність двох субвекторів

Нехай вектор спостережень рокладається на 2 групи \mathbf{y} та \mathbf{x} розмірністю $p \times 1$ та $q \times 1$, відповідно. При цьому коваріаційну, вибіркову коваріаційну та кореляційну матриці Σ , S та R можна теж розкласти наступним чином:

$$\Sigma = \begin{pmatrix} \Sigma_{\mathbf{yy}} & \Sigma_{\mathbf{yx}} \\ \Sigma_{\mathbf{xy}} & \Sigma_{\mathbf{xx}} \end{pmatrix}, \quad S = \begin{pmatrix} S_{\mathbf{yy}} & S_{\mathbf{yx}} \\ S_{\mathbf{xy}} & S_{\mathbf{xx}} \end{pmatrix}, \quad R = \begin{pmatrix} R_{\mathbf{yy}} & R_{\mathbf{yx}} \\ R_{\mathbf{xy}} & R_{\mathbf{xx}} \end{pmatrix}. \quad (5.10)$$

Перевіримо нульову гіпотезу H_0 :

$$H_0: \Sigma = \Sigma_0 = \begin{pmatrix} \Sigma_{\mathbf{yy}} & 0 \\ 0 & \Sigma_{\mathbf{xx}} \end{pmatrix} \quad (5.11)$$

проти альтернативи, що коваріаційна матриця не має блокової структури (5.11). Основна гіпотеза означає, що вектори X і Y є некорельзованими, а отже, за припущенням нормальності, незалежними.

Розглянемо статистику (5.2) з Σ_0 ,

$$S_0 = \begin{pmatrix} S_{yy} & 0 \\ 0 & S_{xx} \end{pmatrix} \quad (5.12)$$

За нульової гіпотези експонента в статистиці (5.2) має порядок $1 + o(1)$ при великому n , тому

$$LR \approx \left(\frac{|S_{xx}| |S_{yy}|}{|S|} (1 + o(1)) \right)^{\frac{n}{2}}.$$

Виходячи з цього, розглянемо статистику Уілкса (Λ Уілкса):

$$\Lambda := \frac{|S|}{|S_{xx}| |S_{yy}|} = \frac{|R|}{|R_{xx}| |R_{yy}|}. \quad (5.13)$$

За умови існування оберненої матриці S_{xx}^{-1} , детермінант матриці можна перетворити наступним чином (див. Задачу 5.2):

$$|S| = |S_{xx}| |S_{yy} - S_{yx} S_{xx}^{-1} S_{xy}|, \quad (5.14)$$

а отже, статистику можна переписати у вигляді

$$\Lambda = \frac{|E|}{|E + H|}, \quad (5.15)$$

де

$$E := S_{yy} - S_{yx} S_{xx}^{-1} S_{xy}, \quad H := S_{yx} S_{xx}^{-1} S_{xy},$$

називаються матрицями похибок (Error matrix) та гіпотези (Hypothesis matrix). Якщо матриця H є близькою до нульової матриці, то $\Lambda \approx 1$. Отже, "хорошими значеннями" Λ є значення, близькі до 1.

За умови H_0 статистика Λ має розподіл Уілкса $\Lambda(p, m, n)$, який визначається як частка детермінантів двох незалежних розподілів Уішарта: нехай є два незалежних розподіли Уішарта $A \sim W_p(m, \Sigma)$ та $B \sim W_p(n, \Sigma)$, та $m \geq p$. Тоді розподіл випадкової величини

$$\lambda = \frac{|A|}{|A + B|} = \frac{1}{|I_p + A^{-1}B|} \sim \Lambda(p, m, n).$$

називається розподілом Уілкса з параметрами p , m та n . Можна порівнювати значення статистики Уілкса зі значеннями квантілю рівня $1 - \alpha$ розподілу Уілкса, але на практиці це не так просто зробити, якщо p , m та n великі.

У нашому випадку

$$\Lambda \sim \Lambda_{p, \nu_H, \nu_E},$$

де ν_H та ν_E є кількостями ступеней свободи в матрицях H та E відповідно. У нашому випадку $\nu_H = q$, $\nu_E = n - 1 - \nu_H$.

За умови H_0 , розподіл статистики Λ можна апроксимувати розподілом Фішера $F_{a,b}$ з певними параметрами a і b , а саме [R02, (6.15)]:

$$F = \frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \frac{df_2}{df_1} \sim F_{df_1, df_2}, \quad (5.16)$$

де p – кількість змінних, ν_H – кількість ступенів свободи матриці H , ν_E – кількість ступенів свободи матриці E ,

$$df_1 = p\nu_H, \quad df_2 = wt - \frac{1}{2}(p\nu_H - 2), \quad (5.17)$$

$$w = \nu_E + \nu_H - \frac{1}{2}(p + \nu_H + 1), \quad t = \sqrt{\frac{p^2\nu_H^2 - 4}{p^2 + \nu_H^2 - 5}}.$$

Насправді, іноді задачі тестуванні гіпотез про коваріаційну матрицю та про середнє взаємно доповнюють одна одну. Ми розглянемо такий ефект в Прикладі 1 в наступному підрозділі.

Вправа 5.1. *A. Довести (5.14).*

Підказка: використати "доповнення Шура"². Див. також [Ta].

B. Довести $\frac{|S|}{|S_{xx}||S_{yy}|} = \frac{|R|}{|R_{xx}||R_{yy}|}$ (див. (5.13)). Для цього

- 1) *знати співвідношення між матрицями Σ та R ,*
- 2) *використати властивості детермінанту.*

Вправа 5.2. *Довести (5.14).*

5.4 Приклади

1. Наступний приклад взято з [R02, Ex.7.2.2]. Психолог проводить дослідження щодо того, наскільки впливає положення слова в реченні на здатність людини його запам'ятати. Для цього взяті так звані “пробні слова” (“probe words”). Взято 5 позицій пробних слів та замірюється кількість часу, яка знадобилась учасникам експерименту для того, щоб пригадати слово, яке було пов'язано з пробним. Дослідимо, чи залежить кількість часу, яка потрібна для виконання завдання, від того, на якій позиції знаходилося слово. Іншими словами, у нас є нормальний вектор розмірності 5 (див. таблицю нижче), і нам треба з'ясувати, чи є матриця коваріації діагональною.

Завантажимо бібліотеку **stats** та файл з даними.

```
library("stats")
probe<-read.table('T3_5_PROBE.DAT')
head(probe)
```

	V1	V2	V3	V4	V5	V6
1	51	36	50	35	42	
2	27	20	26	17	27	
3	37	22	41	37	30	
4	42	36	32	34	27	
5	27	18	33	14	29	
6	43	32	43	35	40	

Можна одразу зчитати дані у вигляді матриці. Зауважимо, що нам потрібні лише стовбчики $V2-V6$.

²<https://www.statlect.com/matrix-algebra/Schur-complement>

```
probe <- as.matrix(read.table('T3_5_PROBE.DAT'))[, 2:6]
```

Перевіримо гіпотезу

$$H_0 : \Sigma = \sigma^2 I_p.$$

Для цього порахуємо статистику (5.3). Обчислимо $p, n, S = cov(probe)$, $\det(S)$ та $trace(S)$.

```
n <- dim(probe)[1]
p <- dim(probe)[2]
S <- cov(probe)
U <- round(((p^p)*det(S))/(sum(diag(S)))^p, digits = 3)
U
```

0.039

Для вибірки середнього розміру n статистика u' (див. (5.4)) має наближено $\chi^2_{\frac{p(p+1)}{2}-1}$ розподіл.

```
u1 <- - (n-1 - (2*p^2 + p + 2)/(6*p)) * log(u)
u1
```

26.278

```
round(qchisq(0.05, p*(p+1)/2-1, lower.tail=FALSE), digits = 3)
```

23.685

Ми округлили до третього знаку за допомогою функції `round(x, digits = 3)`. Отже, відхиляємо H_0 .

Також можна перевірити гіпотезу H_0 за допомогою вбудованої функції `mauchly.test`. Для цього 1) утворюємо лінійну модель, 2) знаходимо матрицю дисперсій SSD.³

```
mlmfit <- lm(S ~ 1)
SSD(mlmfit)
mauchly.test(SSD(mlmfit))
```

`$SSD`

	V2	V3	V4	V5	V6
V2	650.9091	336.4545	475.9091	367.7273	254.2727
V3	336.4545	460.7273	289.4545	403.3636	283.6364
V4	475.9091	289.4545	606.9091	373.7273	411.2727
V5	367.7273	403.3636	373.7273	628.1818	316.8182
V6	254.2727	283.6364	411.2727	316.8182	582.1818

³<https://rdrr.io/r/stats/SSD.html>

```

$call
lm(formula = S ~ 1)

$df
[1] 10

attr(,"class")
[1] "SSD"

Mauchly's test of sphericity

data: SSD matrix from lm(formula = S ~ 1)
W = 0.039489, p-value = 0.02986

```

В цій моделі ми використовуємо залежність

$$\text{response variable} \sim \text{explanatory variable(s)}.$$

В нашому випадку ~ 1 означає, що у нас є лише відгук, константа в правій частині і шум.

Ми отримали те саме значення u , що і вище. Оскільки $p-value$ менше 0.05, відхиляємо нульову гіпотезу.

Насправді, дослідження психологів показують навіть більше – ”в середньому” краще за все запам'ятовується те, що в кінці речення.

Трансформуємо тепер цю задачу. Ми розглянемо ортонормовану матрицю контрастів:

$$C = \begin{pmatrix} \frac{4}{\sqrt{20}} & -\frac{1}{\sqrt{20}} & -\frac{1}{\sqrt{20}} & -\frac{1}{\sqrt{20}} & -\frac{1}{\sqrt{20}} \\ 0 & \frac{3}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} \\ 0 & 0 & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad (5.18)$$

Рядки цієї матриці задовольняють умові $\sum_{j=1}^5 c_{ij} = 0$, та ортонормовані: $\sum_{j=1}^5 c_{ij}^2 = 1$. Ми задаємо останній рядок $(0, 0, 0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$, а далі – добавляємо координати так, щоб зберігались ці умови.

Зауважимо, що

$$C\mu = 0 \quad (5.19)$$

означає рівність середнії в моделі.

Перевіримо основну гіпотезу

$$H_0 : C\Sigma C' = \sigma^2 I_{p-1}.$$

Тоді, використовуючи CSC' замість S в статистиці u , ми отримаємо

$$u = \frac{(p-1)^{p-1} |CSC'|}{|\text{trace}(CSC')|^{p-1}} = 0.480, \quad u' = 6.170.$$

У нас тепер $\frac{(p-1)p}{2} - 1 = 9$ ступені свободи. Обчислюємо $\chi^2_{0.5,9} = 16.92$. Отже, тепер ми не відхиляємо нульову гіпотезу, та маємо

$$Cy \sim N_{p-1}(C\mu, \sigma^2 I_{p-1}).$$

Тепер ми або перевіряємо гіпотезу про те, що $C\mu = 0$, або застосовуємо ANOVA аналіз (див. Розділ 1) для перевірки гіпотези про те, що в групах однакові середні. Попередній аналіз показує, всі спостереження незалежні та мають однакові дисперсії, а отже, ми вкладаємося в модель ANOVA.

2. Box M test, рівність коваріаційних матриць.

Розглянемо Таблицю 5.1, результатами якої є дані психологічних тестів для чоловіків і жінок.

Встановимо пакет **biotools** для того, щоб застосувати вбудований тест Бокса.

```
install.packages("biotools")
library(biotools)
```

Завантажимо дані:

```
Psych<-read.table('T5_1_PSYCH.DAT')
head(Psych)
```

	V1	V2	V3	V4	V5
1	1	15	17	24	14
2	1	17	15	32	26
3	1	15	14	29	23
4	1	13	12	10	16
5	1	20	17	26	28
6	1	15	21	26	21

Застосуємо вбудований тест Бокса. Для його застосування досить використовувати дані у вигляді таблиці.

```
boxM(Psych[, 2:5], Psych[, 1])
```

Box's M-test for Homogeneity of Covariance Matrices

```
data: Psych[, 2:5]
Chi-Sq (approx.) = 13.551, df = 10, p-value = 0.1945
```

Отже, ми приймаємо нульову гіпотезу про те, що в обох групах коваріаційні матриці однакові.

Покажемо, як зробити ці обчислення вручну. $V1$ – це колонка, яка позначає фактори: 1, якщо учасник тесту чоловік, і 2 – якщо жінка. Зробимо групування по факторам за допомоги функції `split`.

```
PsychM <- split(Psych, f = Psych$V1)$'1'
PsychF <- split(Psych, f = Psych$V1)$'2'
```

Обчислимо кількості елементів у вибірках M та F , та коваріаційні матриці.

```

n1 <- dim(PsychM)[1]
n2 <- dim(PsychF)[1]
p <- dim(PsychM)[2]
k <- 2
PsychF1 <- as.matrix(PsychF[,2:5])
PsychM1 <- as.matrix(PsychM[,2:5])
S1 <- cov(PsychF1)
S2 <- cov(PsychM1)

```

Порахуємо зважену коваріаційну матрицю:

```

Spl<-((n1-1)*S1+ (n2-1)*S2)/(n1+n2-2)
Spl

```

	V2	V3	V4	V5
V2	7.164315	6.047379	5.693044	4.700605
V3	6.047379	15.894153	8.492440	5.855847
V4	5.693044	8.492440	29.356351	13.980847
V5	4.700605	5.855847	13.980847	22.320565

Обчислимо тепер статистику Бокса (5.7):

```

M<-((det(S1))^(n1/2))*((det(S2))^(n2/2))/(det(Spl)^(n1/2+n2/2))
round(M, digits = 5)

```

0.00054

Порахуємо сталу (5.9) та статистику (5.8):

```

n <- n1
C1 <- (k+1)*(2*p^2+3*p-1)/(6*k*n*(p+1))
u <- -round(2*(1-C1)*log(M), digits = 3)
u

```

13.778

Отже, результат співпадає з отриманим раніше.

```
round(qchisq(0.05, 0.5*(k-1)*p*(p+1), lower.tail = FALSE), digits = 3)
```

24.996

Отже, приймаємо H_0 .

3. Тест про незалежність субвекторів.

Перевіримо, що результати психологічних тестів для чоловіків і для жінок, які ми розглядали в попередньому пункті, є незалежними. Для цього утворимо наступні коваріаційні матриці:

```
S_FM <- cov(PsychF1, PsychM1)
S_FF <- cov(PsychF1)
S_MM <- cov(PsychM1)
```

Утворимо спільну матрицю коваріацій:

```
A <- cbind(S_FF, S_FM)
B <- cbind(S_FM, S_MM)
S <- rbind(A, B)
```

Обчислимо статистику Уілкса:

```
Lambda <- det(S)/(det(S_FF)*det(S_MM))
round(Lambda, 3)
```

0.792

Тепер обчислимо статистику F (див. (5.16)):

```
n <- dim(PsychM)[1]
p <- dim(PsychM)[2] - 1
k <- 2
nu_H = dim(PsychM)[2] - 1
nu_E = n - 1 - nu_H
w = nu_E + nu_H - 0.5 * (p + nu_H + 1)
t = sqrt(((p * nu_H)^2 - 4) / (p^2 + nu_H^2 - 5))
df1 = p * nu_H
df2 = w * t - 0.5 * (df1 - 2)
F = ((1 - Lambda^(1/t)) / (Lambda^(1/t))) * df2 / df1
round(F, 3)
```

1.782

Знайдемо квантіль рівня 0.95 розподілу $F_{df1, df2}$.

```
round(qf(1 - 0.05, df1, df2, lower.tail = TRUE, log.p = FALSE), 3)
```

1.782

Отже, ми приймаємо гіпотезу про незалежність результатів спостережень для чоловіків і для жінок.

6 ANOVA: одновимірний дисперсійний аналіз (однофакторна модель)

⁴

Література: [R02, Гл.6], [Ma07, §2.5], [Be]. Пакет **car**.

⁴Eng.: Univariate One-Way Analysis of Variance

6.1 ANOVA модель

Нехай є k груп незалежних спостережень нормально розподілених випадкових величин, причому в кожній групі однакова кількість спостережень n . Дисперсії в кожній групі вважаємо одинаковими і рівними σ^2 (яке є невідомим), середні значення також є невідомими. Задамо ці спостереження в Таблиці 1.

Ми застосуємо одновимірний дисперсійний аналіз (ANOVA) для того, щоб перевірити гіпотезу

$$H_0: \mu_1 = \mu_2 = \dots = \mu_k \text{ проти альтернативи } H_1: \mu_i \neq \mu_j \text{ для деяких } i, j.$$

Якщо має місце H_0 , то всі дані y_{ij} належать одній популяції. Для оцінювання σ^2 можна застосувати 2 типи оцінок: перший базується на обчисленні вибіркових дисперсій $s_1^2, s_2^2, \dots, s_k^2$ та є їх середнім

$$s_e^2 = \frac{1}{k} \sum_{i=1}^k s_i^2 = \sum_{i=1}^k \sum_{j=1}^n \frac{(y_{ij} - \bar{y}_{i\cdot})^2}{k(n-1)}, \quad (6.1)$$

де

$$\bar{y}_{i\cdot} = \sum_{j=1}^n y_{ij}/n,$$

Табл. 1: ANOVA

Вибірки				
$N(\mu_1, \sigma^2)$	$N(\mu_2, \sigma^2)$	\dots	$N(\mu_k, \sigma^2)$	
y_{11}	y_{21}	\dots	y_{k1}	
y_{12}	y_{22}	\dots	y_{k2}	
\dots	\dots	\dots	\dots	
y_{1n}	y_{2n}	\dots	y_{kn}	
Сума	$y_{1\cdot}$	$y_{2\cdot}$	\dots	$y_{k\cdot}$
Середнє	$\bar{y}_{1\cdot}$	$\bar{y}_{2\cdot}$	\dots	$\bar{y}_{k\cdot}$
Дисперсія	s_1^2	s_2^2	\dots	s_k^2

а інший – на обчисленні вибіркової дисперсії

$$s_{\bar{y}}^2 := \sum_{i=1}^k \frac{(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2}{k-1}, \quad (6.2)$$

де

$$\bar{y}_{\cdot\cdot} = \sum_{i=1}^k \frac{\bar{y}_{i\cdot}}{k}. \quad (6.3)$$

Припустимо, що наші дані можна зобразити у вигляді

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij} = \mu_i + \varepsilon_{ij}, \quad i = 1, \dots, k, \quad j = 1, \dots, n, \quad (6.4)$$

де $\varepsilon_{ij} \sim N(0, \sigma^2)$ є незалежними, а α_i , $1 \leq i \leq k$, є константами. Тобто, на y_{ij} впливає один фактор, який відображається у значенні α_i . Тому така модель називається **однофакторною**. Модель (6.4) можна переписати у наступному вигляді:

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (6.5)$$

або

$$\begin{pmatrix} y_{11} \\ y_{12} \\ \dots \\ y_{21} \\ y_{22} \\ \dots \\ y_{k1} \\ y_{kn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \dots & \dots & \dots \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ \dots & \dots & \dots \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_{k-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \dots \\ \dots \\ \dots \\ \varepsilon_{kn} \end{pmatrix}, \quad (6.6)$$

де $\beta_i = \mu + \alpha_{i+1}$, $i = 0, \dots, k-1$. В матриці X кожна 1 повторюється n разів. Ця модель є частковим випадком моделі лінійної регресії, яку ми розглянемо в Розділі 8.

Оцінку $s_{\bar{y}}^2$ можна переписати у вигляді

$$ns_{\bar{y}}^2 = \frac{n}{k-1} \sum_{i=1}^k (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2 = \frac{1}{k-1} \left(\sum_{i=1}^k \frac{y_{i\cdot}^2}{n} - \frac{y_{\cdot\cdot}^2}{kn} \right) =: \frac{SSH}{k-1}, \quad (6.7)$$

де $y_{\cdot\cdot}^2 = \sum_{i=1}^k \sum_{j=1}^n y_{ij}$, а

$$SSH := n \sum_{i=1}^k (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2. \quad (6.8)$$

Цю статистику легко модифікувати на випадок різної кількості елементів n_i , $i = 1, \dots, k$, в групі:

$$SSH := \sum_{i=1}^k n_i (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2. \quad (6.9)$$

Позначимо також середнє значення по всій вибірці через $\bar{y}_{\cdot\cdot}^2 = y_{\cdot\cdot}^2/k$. За умови виконання H_0 , $ns_{\bar{y}}^2$ є оцінкою σ^2 : оскільки $\bar{y}_{\cdot\cdot}$ є оцінкою $\bar{y}_{i\cdot} =: \bar{y}$ (тобто $\bar{y}_{i\cdot}$ однакові за умови H_0), то

$$\mathbb{E}s_{\bar{y}}^2 = \mathbb{D}\bar{y} = \frac{\sigma^2}{n}.$$

Зауважимо, що у будь-якому випадку

$$\mathbb{E}s_e^2 = \frac{1}{k} \sum_{i=1}^k \mathbb{E}s_i^2 = \sigma^2.$$

З іншого боку, s_e^2 можна переписати наступним чином:

$$s_e^2 = \frac{\sum_{i=1}^k \sum_{j=1}^n y_{ij}^2 - \sum_{i=1}^k y_{i\cdot}^2/n}{k(n-1)} =: \frac{SSE}{k(n-1)}, \quad (6.10)$$

де

$$SSE := \sum_{i=1}^k \sum_{j=1}^n (y_{ij} - \bar{y}_{i\cdot})^2. \quad (6.11)$$

Значення SSH (англ.: Sum of Squares Hypothesis, "between", тому що просумували по всім значенням) є сумаю квадратів між вибірками, в той час як SSE є сумаю квадратів у вибірці (англ.: Sum of Squares Error, "within", тому що просумували всередині кожної вибірки).⁵

Якщо гіпотеза H_0 не виконується, то (**довести!**)

$$\mathbb{E}[ns_{\bar{y}}^2] = \sigma^2 + \frac{n}{k-1} \sum_{i=1}^k \alpha_i^2, \quad (6.12)$$

де $\alpha_i \neq 0$ для деяких i (інакше має місце H_0). Отже, дисперсія в цьому випадку більша за σ^2 . За умови H_0 , статистики SSE та SSH є незалежними, та їх відношення має розподіл Фішера з $k-1$ та $k(n-1)$ ступенями свободи.⁶

Запишемо тестову статистику (тут ми допускаємо різну кількість елементів n_i в групі i , $i = 1, \dots, k$):

$$F = \frac{ns_{\bar{y}}^2}{s_e^2} = \frac{SSH/(k-1)}{SSE/(k(n-1))} = \frac{N-k}{k-1} \cdot \frac{\sum_{i=1}^k n_i (\bar{y}_{i\cdot} - \bar{y}_{..})^2}{\sum_{i=1}^k \sum_{j=1}^n (y_{ij} - \bar{y}_{i\cdot})^2}, \quad (6.13)$$

де $N = \sum_{i=1}^k n_i$. У випадку, якщо кількість елементів в усіх групах однакова, маємо $N = nk$. За умови H_0 маємо $F \sim F_{k-1, k(n-1)}$. Якщо $F > F_{\alpha, k-1, k(n-1)}$, де $F_{\alpha, k-1, k(n-1)}$ – квантиль рівня $1 - \alpha$ розподілу Фішера $F_{k-1, k(n-1)}$, відхиляємо H_0 .

Хорошою характеристикою адекватності моделі є коефіцієнт детермінації

$$R^2 = 1 - \frac{SSE}{SST} = \frac{SSH}{SST}, \quad (6.14)$$

де

$$SST = SSH + SSE = \sum_{i=1}^k \sum_{j=1}^n (y_{ij} - \bar{y}_{..})^2. \quad (6.15)$$

де SST ("total") є повною сумаю квадратів. Якщо гіпотеза H_0 справедлива, то значення SSH мале, а отже, $R^2 \approx 0$. Іншими словами: якщо всі групи належать одній популяції, то статистично середнє значення в окремій групі і середнє по об'єднанню груп майже не відрізняються. Тому величина SSH має бути малою. В рамках лінійної регресії – коефіцієнти $\beta_1 = \dots = \beta_{k-1}$ значно менші, ніж β_0 . Тому те, що $R \approx 0$, означає, що модель лінійної регресії в цьому випадку працює погано, наша модель фактично не залежить від змінних, а залежить від одного параметру β_0 .

⁵В літературі значення ще позначають як SSB ("between") або SSR при $k = 1$ ("residual", у моделі лінійної регресії, див. (8)).

⁶Це аналог леми Фішера. Ми доведемо це твердження в Розділі 8, де будемо розглядати моделі лінійної регресії.

6.2 Приклади

У наведених нижче прикладах і надалі нам знадобиться пакет **car**.

Розглянемо наступний приклад⁷. Порівнюють вагу рослини в залежності від групи, до якої ця рослина належить.

```
data("PlantGrowth")      # Дані з пакету R
PlantGrowth
```

	weight	group
1	4.17	ctrl
2	5.58	ctrl
3	5.18	ctrl
4	6.11	ctrl
5	4.50	ctrl
6	4.61	ctrl
7	5.17	ctrl
8	4.53	ctrl
9	5.33	ctrl
10	5.14	ctrl
11	4.81	trt1
12	4.17	trt1
13	4.41	trt1
14	3.59	trt1
15	5.87	trt1
16	3.83	trt1
17	6.03	trt1
18	4.89	trt1
19	4.32	trt1
20	4.69	trt1
21	6.31	trt2
22	5.12	trt2
23	5.54	trt2
24	5.50	trt2
25	5.37	trt2
26	5.29	trt2
27	4.92	trt2
28	6.15	trt2
29	5.80	trt2
30	5.26	trt2

Як ми бачимо, є три групи рослин та 30 спостережень (по 10 в кожній групі), тобто в описаній вище моделі $k = 3$ та $n = 10$.

Ми можемо застосувати "лінійну модель" для того, щоб перевірити гіпотезу

$$H_0: \alpha_1 = \alpha_2 = \alpha_3.$$

⁷<https://rpubs.com/aaronsc32/anova-compare-more-than-two-groups>

(див. (6.6)). Для цього ми спочатку будуємо лінійну модель за допомоги функції lm.

```
lm(weight ~ group, data = PlantGrowth)
```

Тут group -це незалежна змінна, а weight- залежна. В результаті отримаємо оцінки на коефіцієнти регресії ($\beta_0, \beta_1, \beta_2$) за формулою (8.7), див. Розділ (8.1).

Call:

```
lm(formula = weight ~ group, data = PlantGrowth)
```

Coefficients:

(Intercept)	grouptrt1	grouptrt2
5.032	-0.371	0.494

По суті, ми задаємо тим самим модель лінійної регресії

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2,$$

але зараз нас цікавить не оцінка коефіцієнтів регресії x_1 та x_2 (що буде зроблено пізніше в Розділі 8), а аналіз сум квадратів та перевірка гіпотези про рівність середніх значень. Якщо коефіцієнти при x_1 та x_2 нульові, це означає, що ми не можемо розрізнати групи, тобто всі середні значення μ_i однакові.

```
plant.aov <- aova(lm(weight ~ group, data = PlantGrowth))
plant.aov
```

Analysis of Variance Table

Response: weight	
	Df Sum Sq Mean Sq F value Pr(>F)
group	2 3.7663 1.8832 4.8461 0.01591 *
Residuals	27 10.4921 0.3886

Signif. codes:	0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Для зручності, пояснемо дані за допомогою наступної таблиці.

Табл. 2: Пояснення до ANOVA таблиці

	Df	Sum Sq	Mean Sq	F value	Pr($F >$)
Group	$k - 1$	SSH	$SSH/(k - 1)$	$\frac{SSH/(k-1)}{SSE/(k(n-1))}$	p-value
Residuals	$k(n - 1)$	SSE	$SSE/(k(n - 1))$		

У нас $k-1 = 3-1 = 2$, $k(n-1) = 3*9 = 27$, а суми SSH та SSE обчислюються, відповідно, за (6.9) та (8.10). Оскільки $p\text{-value} < 0.05$, то відхиляємо гіпотезу про рівність середніх.

Альтернативно, можна зробити аналіз дисперсій за допомогою функції `aov`.

```
aov(lm(weight ~ group, data = PlantGrowth))
```

Call:

```
aov(formula = lm(weight ~ group, data = PlantGrowth))
```

Terms:

```
group Residuals
```

```
Sum of Squares 3.76634 10.49209
```

```
Deg. of Freedom 2 27
```

Residual standard error: 0.6233746

Estimated effects may be unbalanced

Residual standard error - це $\sqrt{Mean Sq}$.

Попередження "Estimated effects may be unbalanced" означає, що дані можуть містити різну кількість спостережень (модель використовується, якщо кількість спостережень однаакова). У нашому випадку в кожній групі по 10 спостережень, отже, модель є сбалансованою.

7 МАНОВА: багатовимірний дисперсійний аналіз (однофакторна модель)

Література: [Ma07, §2.6], [R02, Гл.6]. Пакет `cars`.

7.1 МАНОВА модель

Розглянемо однофакторну модель багатовимірного дисперсійного аналізу (Multivariate One-Way Analysis of Variance, або MANOVA). За винятком рядочку "дисперсія", спостереження можна записати у вигляді Таблиці 1, тільки тепер всі значення цієї таблиці - це вектори розмірності $p \times 1$, які мають багатовимірний нормальний розподіл $N_p(\boldsymbol{\mu}_i, \Sigma)$, $1 \leq i \leq k$ (відповідно до номеру групи).

Перевіримо нульову гіпотезу

$$H_0: \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \cdots = \boldsymbol{\mu}_k \text{ проти альтернативи } \boldsymbol{\mu}_i \neq \boldsymbol{\mu}_j \text{ для деяких } i, j.$$

Оскільки ми розглядаємо векторний випадок, нульова гіпотеза означає, що

$$\begin{pmatrix} \mu_{11} \\ \mu_{12} \\ \vdots \\ \mu_{1p} \end{pmatrix} = \begin{pmatrix} \mu_{21} \\ \mu_{22} \\ \vdots \\ \mu_{2p} \end{pmatrix} = \cdots = \begin{pmatrix} \mu_{k1} \\ \mu_{k2} \\ \vdots \\ \mu_{kp} \end{pmatrix}.$$

У випадку, коли вибірки різного розміру (тобто модель не є сбалансованою), тобто вибірка i має розмір n_i ,

$$\mathbf{y}_i = \boldsymbol{\mu} + \boldsymbol{\alpha}_i + \varepsilon_{ij}, \quad i = 1, \dots, k, \quad j = 1, \dots, n_i.$$

Відповідно,

$$\bar{\mathbf{y}}_{i\cdot} = \sum_{j=1}^{n_i} \mathbf{y}_{ij} / n_i, \quad \mathbf{y}_{\cdot\cdot} = \sum_{i=1}^k \sum_{j=1}^{n_i} \mathbf{y}_{ij}, \quad \bar{\mathbf{y}}_{\cdot\cdot} = \mathbf{y}_{\cdot\cdot} / N, \quad \text{де} \quad N = \sum_{i=1}^k n_i.$$

Тоді (порівняйте з (6.8) та (6.11)):

$$H := \sum_{i=1}^k n_i (\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})(\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})' \quad (7.1)$$

та

$$E := \sum_{i=1}^k \sum_{j=1}^{n_i} (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})'. \quad (7.2)$$

Ця модель є природнім узагальненням одновимірного випадку (ANOVA). Відповідно, замість скалярних добутків ми будемо мати квадрати $(\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})$ та $(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})$, відповідно.

У випадку, коли всі вибірки мають одинаковий розмір n , маємо (порівняйте з статистиками SSH та SSE в одновимірному випадку!)

$$H := n \sum_{i=1}^k (\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})(\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})' = \sum_{i=1}^k \frac{1}{n} \mathbf{y}_{i\cdot} \mathbf{y}'_{i\cdot} - \frac{1}{kn} \mathbf{y}_{\cdot\cdot} \mathbf{y}'_{\cdot\cdot} \quad (7.3)$$

та

$$E := \sum_{i=1}^k \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})' = \sum_{ij} \mathbf{y}_{ij} \mathbf{y}'_{ij} - \sum_{i=1}^k \frac{1}{n} \mathbf{y}_{i\cdot} \mathbf{y}'_{i\cdot}. \quad (7.4)$$

Ці (випадкові!) матриці є незалежними та мають розподілі Уішарта $W_p(\nu_H, \Sigma)$ та $W_p(\nu_E, \Sigma)$, див. [R98, с.124, Th.4.1B]. Отже, $\frac{|E|}{|E+H|}$ має розподіл Уілкса Λ_{p,ν_H,ν_E} . Тому замість статистики F (з (8.14)) будемо використовувати статистику Уілкса Λ , при чому "матриця гіпотези" E і "матриця похибок" H мають ступені свободи, відповідно,

$$\nu_H = k - 1, \quad \nu_E = k(n - 1). \quad (7.5)$$

(у випадку, коли всі вибірки мають розмір n), та

$$\nu_H = k - 1, \quad \nu_E = N - k = \sum_{i=1}^k n_i - k, \quad (7.6)$$

у випадку, коли кількість різна. Порівняйте з однофакторною моделлю у попередній главі!

Зважена коваріаційна матриця у цьому випадку має вигляд

$$S_{pl} = \frac{E}{\nu_E}. \quad (7.7)$$

Статистика Λ така сама, як і в одновимірному випадку, та

$$\text{rank } H = \min(p, \nu_H) =: s, \quad \text{rank } E = \min(p, \nu_E). \quad (7.8)$$

Для перевірки гіпотези розглянемо теж статистику $\Lambda = \frac{|E|}{|E+H|}$, див. (5.15). Як і раніше (див. Розділ 5.3), ми відхиляємо нульову гіпотезу, якщо відповідна статистика F (cf. (5.16)) перевищує F_{α, df_1, df_2} , див. визначення df_1 та df_2 в (5.17).

Замість статистики Уілкса застосовують також інші тести.

Статистика Роя (Roy's statistics)

Розглянемо наступне перетворення вектору \mathbf{y}_{ij} : $z_{ij} = \mathbf{a}' \mathbf{y}_{ij}$, де $\mathbf{a} \in \mathbb{R}^p$, $1 \leq j \leq n$; тобто, ми проектуємо \mathbf{y}_{ij} на пряму. Розглянемо тепер статистику F з (8.14), але застосовану до z_{ij} :

$$\begin{aligned} F &= \frac{n \sum_{i=1}^k (\bar{z}_{i\cdot} - \bar{z}_{\cdot\cdot})^2 / (k-1)}{\sum_{i=1}^k \sum_{j=1}^n (z_{ij} - \bar{z}_{i\cdot})^2 / (k(n-1))} = \frac{SSH_z / (k-1)}{SSE_z / (k(n-1))} \\ &= \frac{\mathbf{a}' H \mathbf{a} / (k-1)}{\mathbf{a}' E \mathbf{a} / (k(n-1))}. \end{aligned} \quad (7.9)$$

(тут індекс z означає, що величини SSE та SSH записані для значень z , а не для значень \mathbf{y} , для яких вони не мають сенсу). Оберемо тепер в якості \mathbf{a} власний вектор матриці $E^{-1}H$, який відповідає найбільшому власному числу λ_1 . Тоді на цьому значенні \mathbf{a}_1 досягається $\max F$, оскільки λ_1 -це максимальний власний вектор:

$$\max_{\mathbf{a} \in \mathbb{R}^p} F = \frac{\mathbf{a}'_1 E E^{-1} H \mathbf{a}_1 / (k-1)}{\mathbf{a}'_1 E \mathbf{a}_1 / (k(n-1))} = \frac{\mathbf{a}'_1 E \lambda_1 \mathbf{a}_1 / (k-1)}{\mathbf{a}'_1 E \mathbf{a}_1 / (k(n-1))} = \frac{\lambda_1 k(n-1)}{k-1}.$$

В розділі 9 Лінійний дискримінантний аналіз ми покажемо, що максимум дійсно досягається на власних векторах матриці $E^{-1}H$. Зauważимо, що на відміну від F величина $\max_{\mathbf{a} \in \mathbb{R}^p} F$ не є розподіленою за Фішером. Для перевірки гіпотези H_0 застосуємо тест Роя (англ.: Roy's largest number test):

$$\theta = \frac{\lambda_1}{1 + \lambda_1}. \quad (7.10)$$

В загальному випадку точний розподіл θ невідомий, але розглядають “верхню межу”

$$F := \frac{(\nu_E - d - 1)\lambda_1}{d} \sim F_{d, \nu_E - d - 1}, \quad d = \max(p, \nu_H).$$

Термін “верхня межа” означає, що реальне значення F більше за $F_{d, \nu_E - d - 1}$. Тому ми відхиляємо H_0 якщо $F < F_{d, \nu_E - d - 1}$.

Розглянемо ще дві статистики.

Статистика Пілая (Pillai statistics):

$$V^{(s)} = \text{trace}[(E + H)^{-1} H] = \sum_{i=1}^s \frac{\lambda_i}{1 + \lambda_i}. \quad (7.11)$$

За умови H_0 ,

$$F := \frac{(2w + s + 1)V^{(s)}}{(2m + s + 1)(s - V^{(s)})} \sim F_{s(2m+s+1), s(2w+s+1)}, \quad (7.12)$$

де $m := (|p - \nu_H| - 1)/2$, $w := (\nu_E - p - 1)/2$.

Статистика Лаули-Хотеллінга (Lawley-Hotelling statistics):

$$U^{(s)} = \text{trace}[E^{-1}H] = \sum_{i=1}^s \lambda_i. \quad (7.13)$$

Цю статистику також можна трансформувати в величину, яка за виконання нульової гіпотези "наблизено" має розподіл Фішера з певними ступенями свободи, див. [R02, (6.29)–(6.31)].

Всі ці тести є еквівалентними при $\nu_H = 1$. В іншому випадку результати тестів можуть розрізнятися завдяки різному впливу факторів на статистики. Тому для більш повного аналізу краще перевіряти гіпотезу декількома тестами.

7.2 Приклади

У наведених нижче прикладах і надалі нам також знадобиться пакет car.

Розглянемо Приклад 6.1.7 з [R02]. Наведений нижче код запропоновано на сторінці [Sh], зокрема, <https://rpubs.com/aaronsc32/manova>. В таблиці нижче наведені результати вимірювань параметрів прищеп для 6 груп прищеп до яблуневих дерев. Ми перевіримо гіпотезу про те, що насправді параметри однакові для всіх 6-ти груп. Зчитаємо Таблицю 6.2 з [R02]:

```
root <- read.table('T6_2_ROOT.DAT',
                    col.names = c('V1', 'V2', 'V3', 'V4', 'V5'))
```

Ми позначили через колонки наступні величини (для зручності, наведемо оригінальні назви, використані в [R02]):

V1: Номер дерева ('Tree.Number');

V2: Обхват дерева через 4 роки ('Trunk.Girth.4.Years')

V3: Висота через 4 роки ('Ext.Growth.4.Years')

V4: Обхват дерева через 15 років ('Trunk.Girth.15.Years')

V5 : Вага над ґрунтом через 15 років. ('Weight.Above.Ground.15.Years')

Наведемо перші 6 значень цієї таблиці (в першій колонці у нас поки лише результати для дерев з першої групи, усього в таблиці 6 груп дерев, по 8 у кожній групі, тобто $k = 6$, $n = 8$; також, у нас 4 змінні, а отже, $p = 4$):

	V1	V2	V3	V4	V5
1	1	1.11	2.569	3.58	0.760
2	1	1.19	2.928	3.75	0.821
3	1	1.09	2.865	3.93	0.928
4	1	1.25	3.844	3.94	1.009
5	1	1.11	3.027	3.60	0.766
6	1	1.08	2.336	3.51	0.726

Далі ми поділимо дані на групи відповідно до номеру дерева та позначимо залежні змінні. Після цього ми можемо застосувати функцію aov.

```
root$V1 <- as.factor(root$V1)
dependent.vars <- as.matrix(read.table('T6_2_ROOT.DAT'))[, 2:5]
aov(dependent.vars ~ root$V1)
```

```

Call:
aov(formula = dependent.vars ~ root$V1)

Terms:
root$V1      Residuals
resp 1        0.073560   0.319987
resp 2        4.199662   12.142790
resp 3        6.113935   4.290813
resp 4        2.493091   1.722525
Deg. of Freedom      5           42

Residual standard errors: 0.08728545 0.5376933 0.3196282 0.2025154
Estimated effects may be unbalanced

```

Такий самий результат дає застосування функції manova:

```
manova(dependent.vars ~ root$V1)
```

Або, у розгорнотому вигляді,

```
summary(aov(dependent.vars ~ root$V1))
```

```

Response 1 :
            Df    Sum Sq    Mean Sq     F value    Pr(>F)
root$V1      5    0.07356   0.0147121   1.931      0.1094
Residuals    42   0.31999   0.0076187
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Response 2 :
            Df    Sum Sq    Mean Sq     F value    Pr(>F)
root$V1      5    4.1997   0.83993    2.9052     0.0243 *
Residuals    42   12.1428   0.28911
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Response 3 :
            Df    Sum Sq    Mean Sq     F value    Pr(>F)
root$V1      5    6.1139   1.22279    11.969     3.112e-07 ***
Residuals    42   4.2908   0.10216
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Response 4 :
            Df    Sum Sq    Mean Sq     F value    Pr(>F)
root$V1      5    2.4931   0.49862    12.158     2.587e-07 ***
Residuals    42   1.7225   0.04101
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Аналіз проводиться для кожної змінної окремо (Response 1–Response 4).

Розберемо, що знаходиться в цій таблиці. Перша колонка- це кількість ступенів свободи: $\nu_H = k - 1 = 5$ та $\nu_E = k(n - 1) = 6 * 7 = 42$ (див. (7.5)). Друга колонка- це діагональні елементи матриць H та E , відповідно. Ці матриці можна отримани з MANOVA наступним чином:

```
root.manova <- summary(manova(dependent.vars ~ root$V1))
H <- root.manova$SS[1]
E <- root.manova$SS[2]
```

Як результат, отримаємо матриці H

```
$`root$V1`
      [,1]      [,2]      [,3]      [,4]
[1,] 0.07356042 0.5373852 0.3322646 0.208470
[2,] 0.53738521 4.1996619 2.3553885 1.637108
[3,] 0.33226458 2.3553885 6.1139354 3.781044
[4,] 0.20847000 1.6371084 3.7810437 2.493091
```

та E :

```
$Residuals
      [,1]      [,2]      [,3]      [,4]
[1,] 0.3199875 1.696564 0.5540875 0.217140
[2,] 1.6965637 12.142790 4.3636125 2.110214
[3,] 0.5540875 4.363612 4.2908125 2.481656
[4,] 0.2171400 2.110214 2.4816562 1.722525
```

Ми наведемо також код для обчислення цих матриць, див. [Sh].

```
root.group <- split(root[,2:5], root$V1)

root.means <- sapply(root.group, function(x) {apply(x, 2, mean)} ,
  simplify = 'data.frame')

root.means
n <- dim(root)[1] / length(unique(root$V1))

total.means <- colMeans(root[,2:5])
total.means

H = matrix(data = 0, nrow = 4, ncol = 4) # Обчислення H
for (i in 1:dim(H)[1]) {
  for (j in 1:i) {
    H[i,j] <- n * sum((root.means[i,] - total.means[i]) *
      (root.means[j,] - total.means[j]))}
```

```

        H[j, i] <- n * sum((root.means[j,] - total.means[j])
                            * (root.means[i,] - total.means[i]))
    }
}

E = matrix(data = 0, nrow = 4, ncol = 4)      # Обчислення E
for (i in 1:dim(E)[1]) {
    for (j in 1:i) {
        b <- c()
        for (k in root.group) {
            a <- sum((k[,i] - mean(k[,i])) * (k[,j] - mean(k[,j])))
            b <- append(b, a)
        }
        E[i,j] <- sum(b)
        E[j,i] <- sum(b)
    }
}

```

Інший спосіб підрахування матриць E і H наведено в відповідній лабораторній роботі на Python, та в Розділі 9. Зауважмо, що на відміну від R, відповідна функція manova на Python не містить атрибутів, що відповідають E і H . Більш того, будована функція manova в Python взагалі використовує підхід лінійної регресії змістъ обчислення лямбди Уілкса! Ці матриці E і H можна отримати як атрибути з функції, що використовується в лінійному дискримінантному аналізі. Ми повернемося до цієї теми в Розділі 9.

Повернемося до таблиці, яку ми отримали після застосування функції manova. Числа в третьій колонці – це числа у другій колонці, поділені на кількість ступенів свободи, тобто на відповідні числа у першій колонці. Числа у четвертій колонці – це відношення зваженої суми квадратів, що відповідає змінній, і суми квадратів, що відповідає залишкам, тобто, наприклад, для першої змінної $0.0147121/0.0076187 = 1.931051$. Як і в попередньому розділі, за умови гіпотези H_0 – це відношення має розподіл Фішера. Остання колонка – це p -value.

З іншого боку, можна застосувати summary(manova) для того, щоб перевірити гіпотезу H_0 про те, всі групи мають однакові середні. Для цього треба викликати

```
root.manova
```

	Df	Pillai	approx F	num Df	den Df	Pr(>F)
root\$V1	5	1.3055	4.0697	20	168	1.983e-07 ***
Residuals	42					
Signif. codes:	0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

Тобто, використання root.manova дає нам значення статистики Пілайя (7.11). Маємо: $\nu_H = k - 1 = 5$, $\nu_E = k(n - 1) = 42$, $m = 0$, $2w = 37$. Випадкова величина F з (7.12) має за виконання гіпотези розподіл $F_{20,168}$.

Для того, щоб обчислити це значення вручну, застосуємо матриці E та H . Вище ми зчитали матриці E та H як таблиці. Перетворимо їх у матриці та обчислимо статистику Пілайя:

```

E1 <- as.matrix(E$Residuals)
H1 <- as.matrix(H$'root$V1')
Vs <- sum(diag(solve(E1 + H1) %*% H1))

```

Отримаємо $V^{(s)} = 1.305472$. Пряме обчислення (див. (7.11)) приводить до значення $F = 4.069846$, що лежить в критичній області (ми бачимо, що $p-value$ дуже мале).

Можна також вказати інші тести:

```

summary(manova(dependent.vars ~ root$V1), test="Wilks")
summary(manova(dependent.vars ~ root$V1), test="Roy")
summary(manova(dependent.vars ~ root$TV1),
        test="Hotelling-Lawley")

```

Так само, треба обчислити відповідну статистику, і використати те, що після певного перетворення статистики θ , $U^{(s)}$ $V^{(s)}$ мають наближено розподіл Фішера.

Є метод, як візуалізувати дані на площині, будуючи "елліпсоїди довіри" по (нормованим або ні) матрицям H та E . Розбиваючи дані на пари та будуючи такі $H - E$ еліпсоїди, ми можемо по їх розміщенню (орієнтації, розміру) зробити висновки, які саме дані можуть впливати на відхилення гіпотези H_0 . Про це можна почитати в [FFM09], [Fr] та пакет **heplots**.

8 Множинна лінійна регресія та багатовимірна лінійна регресія

Література: [R02, Гл.10]. Пакети **car**, **dplyr**.

Розглянемо наступні моделі лінійної регресії.

1. Проста лінійна регресія: маємо одну змінну регресії x і один відгук y . Наприклад, наша задача спрогнозувати середній бал студента вузу, маючи середній бал цього студента під час навчання в школі.
2. Множинна лінійна регресія: маємо один відгук y і декілька змінних x . Така задача виникає, наприклад, коли треба спрогнозувати середній бал у вузі, виходячи з оцінок з певних предметів у школі.
3. Багатовимірна ланайна регресія: маємо декілька відгуків y та декілька змінних x регресії. Продовжуючи попередні приклади, така задача виникає, коли ми пригнозуємо бали по певним предметам у вузі, маючи бали по певним предметам у школі.

Незалежні змінні можуть бути фіксованими або випадковими. У попередніх прикладах всі були випадковими, оскільки ми випадковим чином обираємо студента. Якщо, наприклад, ми дослуждуємо вплив певних ліків на рівень холестерину в крові, можна зафіксувати кількість медикаментів та спостерігати зміни в рівні холестерину.

Ми будемо розглядати модель множинної регресії, в яких змінні регресії x є фіксованими.

8.1 Проста лінійна множинна регресія

Розглянемо наступну (одновимірну за відгуком) модель:

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_{11} + \cdots + \beta_q x_{1q} + \varepsilon_1 \\ y_2 &= \beta_0 + \beta_1 x_{21} + \cdots + \beta_q x_{2q} + \varepsilon_2 \\ &\dots \\ y_n &= \beta_0 + \beta_1 x_{n1} + \cdots + \beta_q x_{nq} + \varepsilon_n. \end{aligned} \tag{8.1}$$

В цій моделі у нас \mathbf{x} є вектором розмірності q (тобто є q змінних). Величини $\beta_i, i = 0, \dots, q$, називаються коефіцієнтами регресії; вільний доданок ще називають інтерсепт (intercept). Похибки ε_i є незалежними нормальними однаково розподіленими випадковими величинами. Припустимо, що виконані наступні умови:

1. $\mathbb{E}\varepsilon_i = 0, i = 1, \dots, n$ (тобто похибки центровані);
2. $\mathbb{E}\varepsilon_i^2 = \sigma^2, i = 1, \dots, n$ (похибки мають однакову дисперсію);
3. $\text{cov}(\varepsilon_i, \varepsilon_j) = 0, i \neq j$ (похибки некорельовані).

Якщо x_i є фіксованими, то

$$\mathbb{E}y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_q x_{iq}.$$

При цьому, з пункту 2 випливає, що $\mathbb{D}y_i = \sigma^2$, та з 3) отримаємо $\text{cov}(y_i, y_j) = 0$.

У матричному вигляді ця модель має наступний вигляд:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1q} \\ 1 & x_{21} & x_{22} & \dots & x_{2q} \\ \dots & \dots & \dots & & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nq} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_q \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}, \tag{8.2}$$

або

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{8.3}$$

де $\mathbf{y} = (y_1, y_2, \dots, y_n)$ є вектором $n \times 1$,

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1q} \\ 1 & x_{21} & x_{22} & \dots & x_{2q} \\ \dots & \dots & \dots & & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nq} \end{pmatrix} \tag{8.4}$$

є матрицею $n \times (1+q)$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)$, $\mathbb{E}\boldsymbol{\varepsilon} = 0$, $\mathbb{E}\mathbf{y} = X\boldsymbol{\beta}$, $\text{cov } \boldsymbol{\varepsilon} = \sigma^2 I_n$, $\text{cov } \mathbf{y} = \sigma^2 I_n$. Надалі ми будемо припускати, що $n > q + 1$, тобто спостережень досить багато; тоді матриця $X'X$ не є сингулярною. Зауважимо також, що випадок, коли всі y_i однаково розподілені, можливий лише тоді, коли $\beta_i = 0, 1 \leq i \leq q$.

Наша задача – побудувати оцінки коефіцієнтів регресії β_i , $i = 0, \dots, q$,

$$\hat{y}_i = \mathbb{E}y_i = \hat{\beta}_0 + \hat{\beta}_i x_{i1} + \dots + \hat{\beta}_q x_{iq}, \quad (8.5)$$

та перевірити гіпотезу

$$H_0: \beta_1 = \beta_2 = \dots = \beta_q = 0$$

Зауважимо, що \hat{y}_i є оцінкою $\mathbb{E}y_i$, а не y_i . Побудуємо оцінку методом найменших квадратів (least squares estimate, LSE) векторного параметру β :

$$\hat{\beta} = \arg \min_{\beta} |\mathbf{y} - \hat{\mathbf{y}}|^2 = \arg \min_{\beta} \sum_{i=1}^n |y_i - \hat{y}_i|^2 = \arg \min_{\beta} SSE_{\beta},$$

де SSE є сумаю квадратів залишків регресії SSE (Sum of Squares Error) (див. також Розділ 6.1). Іншими словами,

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \hat{\varepsilon}_i^2. \quad (8.6)$$

Теорема 8.1. *Припустимо, що $n > q+1$ та $(x_j)_{j=1}^q$ є лінійно незалежними. Тоді оцінкою β методом найменших квадратів є*

$$\hat{\beta} = (X'X)^{-1}X'\mathbf{y}. \quad (8.7)$$

Доведення. Розглянемо $SSE_{\beta} = (\mathbf{y} - X\beta)'(\mathbf{y} - X\beta)$. Добавимо та відніммо $X\hat{\beta}$, де $\hat{\beta}$ визначено в (8.7). Маємо:

$$\begin{aligned} SSE_{\beta} &= (\mathbf{y} - X\hat{\beta} + X(\hat{\beta} - \beta))'(\mathbf{y} - X\hat{\beta} + X(\hat{\beta} - \beta)) \\ &= (\mathbf{y} - X\hat{\beta})'(\mathbf{y} - X\hat{\beta}) + (X(\hat{\beta} - \beta))'(X(\hat{\beta} - \beta)) \\ &\quad + (\mathbf{y} - X\hat{\beta})'(X(\hat{\beta} - \beta)) + (X(\hat{\beta} - \beta))'(\mathbf{y} - X\hat{\beta}) \\ &= I_1 + I_2 + I_3 + I_4. \end{aligned}$$

Підставивши $\hat{\beta} = (X'X)^{-1}X'\mathbf{y}$, отримаємо $I_3 = I_4 = 0$. Наприклад,

$$I_3 = (\mathbf{y} - X\hat{\beta})'(X(\hat{\beta} - \beta)) = (\mathbf{y}'X - \mathbf{y}'X(X'X)^{-1}(X'X))(\hat{\beta} - \beta) = 0.$$

Оскільки $I_2 \geq 0$, SSE_{β} набуває мінімуму при $\beta = \hat{\beta}$, тобто коли $I_2 = 0$. \square

Надалі ми будемо позначати $SSE \equiv SSE_{\hat{\beta}}$. Зауважимо, що (див. Лема 1.1)

$$\mathbb{E}\hat{\beta} = (X'X)^{-1}X'\mathbb{E}\mathbf{y} = (X'X)^{-1}X'X\beta = \beta, \quad (8.8)$$

$$\text{Var}(\hat{\beta}) = (X'X)^{-1}X' \text{Var} \mathbf{y} X (X'X)^{-1} = \sigma^2 (X'X)^{-1}, \quad (8.9)$$

де ми використали $\text{Var} \mathbf{y} = \sigma^2 I_n$. З теореми 8.1 випливає, що

$$\begin{aligned} SSE &= \mathbf{y}'\mathbf{y} - \mathbf{y}'X(X'X)^{-1}X'\mathbf{y} - \mathbf{y}'X(X'X)^{-1}X'\mathbf{y} \\ &\quad + \mathbf{y}'X(X'X)^{-1}(X'X)(X'X)^{-1}X'\mathbf{y} \\ &= \mathbf{y}'\mathbf{y} - \mathbf{y}'X(X'X)^{-1}X'\mathbf{y} \\ &= \mathbf{y}'\mathbf{y} - \hat{\beta}'X'\mathbf{y}. \end{aligned} \quad (8.10)$$

Підставимо оцінку $\hat{\beta}$ та позначимо $\Pi = X(X'X)^{-1}X'$. Зауважимо, що ця матриця має розмірність $(q+1) \times n$,

$$\text{trace } \Pi = \text{trace}(X'X)^{-1}(X'X) = I_{q+1}.$$

та $\Pi = \Pi' = \Pi'\Pi = \Pi'\Pi$. Маємо:

$$SSE = \mathbf{y}'\mathbf{y} - \mathbf{y}'X(X'X)^{-1}X'\mathbf{y} = (\mathbf{y} - \Pi\mathbf{y})'(\mathbf{y} - \Pi\mathbf{y}). \quad (8.11)$$

Використовуючи це обчислення, знайдемо математичне сподівання SSE . Маємо:

$$\Pi\mathbf{y} = X(X'X)^{-1}X'(X\beta + \varepsilon) = X\beta + X(X'X)^{-1}X'\varepsilon = X\beta + \Pi\varepsilon.$$

Підставивши в SSE , отримаємо:

$$\begin{aligned} SSE &= (\mathbf{y} - X\beta - \Pi\varepsilon)'(\mathbf{y} - X\beta - \Pi\varepsilon) = (\varepsilon - \Pi\varepsilon)'(\varepsilon - \Pi\varepsilon) \\ &= \varepsilon'\varepsilon - \varepsilon'\Pi\varepsilon \\ &= \varepsilon'\varepsilon - \text{trace } \Pi\varepsilon'\varepsilon. \end{aligned}$$

Тепер, якщо взяти математичне сподівання, отримаємо

$$\mathbb{E}SSE = \mathbb{E}\varepsilon'\varepsilon - \text{trace } \Pi\mathbb{E}\varepsilon'\varepsilon = n\sigma^2 - (q+1)\sigma^2.$$

Отже, $SSE/(n-1-q)$ є незміщеною оцінкою дисперсії σ^2 :

$$\frac{\mathbb{E}SSE}{n-1-q} = \sigma^2.$$

Зауважимо також, що поки ми ніде не використали припущення H_0 !

Розглянемо тепер загальну суму квадратів (Sum of Squares Total):

$$SST := \sum_{i=1}^n (y_i - \bar{y})^2 = \mathbf{y}'\mathbf{y} - n\bar{y}^2.$$

Зауважимо, що за умови

$$H_0 : \beta_1 = \beta_2 = \dots = \beta_q = 0 \quad (8.12)$$

маємо $y_i \sim N(\beta_0, \sigma^2)$, а отже, $\frac{SST}{n-1}$ є незміщеною оцінкою σ^2 :

$$\mathbb{E}SST = \sigma^2(n-1),$$

та

$$\frac{SST}{(n-1)\sigma^2} \sim \chi_{n-1}^2.$$

Розкладемо SST в суму квадратів залишків регресії SSE та залишкової суми квадратів SSR (Sum of Squares Regression), що пояснюється регресією:

$$SST = (\mathbf{y}'\mathbf{y} - \hat{\beta}'X'\mathbf{y}) + (\hat{\beta}'X'\mathbf{y} - n\bar{y}^2) = SSE + SSR.$$

Покажемо, що за умови виконання H_0 статистики $\frac{SSE}{(n-q-1)\sigma^2}$ та $\frac{SSR}{q\sigma^2}$ незалежні та мають χ^2 -розділ з $n-q-1$ та q ступенями свободи, відповідно.

Запишемо SSE та SSR у вигляді норм $\|\cdot\|_n$ в \mathbb{R}^n :

$$SSE = \|(I - \Pi)\mathbf{y}\|_n^2,$$

$$SSR = \mathbf{y}'\Pi\mathbf{y} - (B_n\mathbf{y})'(B_n\mathbf{y}) = \|(\Pi - B_n)\mathbf{y}\|_n^2.$$

де B_n – це матриця, всі елементами якої є $1/n$ (зауважимо, що $B'_n B_n = B_n = B'_n$).

Зауважимо, що матриці $I - \Pi$ і $\Pi - B_n$ ортогональні:

$$(I - \Pi)(\Pi - B_n) = 0.$$

Оскільки за H_0 виконано $\mathbf{y} \sim N_n(0, \sigma^2 I_n)$ та $\text{cov}((I - \Pi)\mathbf{y}, (\Pi - B_n)\mathbf{y}) = 0$, це означає, що $(I - \Pi)\mathbf{y}$ та $(\Pi - B_n)\mathbf{y}$ некорельовані (перевірити!), а отже, незалежні. Більш того лінійні перетворення нормальних випадкових величин $(I - \Pi)\mathbf{y}$ та $(\Pi - B_n)\mathbf{y}$ також нормальню розподілені. А отже, після нормування, SSE та SSR мають χ^2 -квадрат розподіл як квадрат нормально розподілених випадкових величин:

$$\frac{SSE}{(n - q - 1)\sigma^2} \sim \chi_{n-1-q}^2, \quad \frac{SSR}{q\sigma^2} \sim \chi_q^2. \quad (8.13)$$

Перевіримо нульову гіпотезу (8.12) (зауважимо, що ми не робимо припущення відносно β_0 !). З (8.13) випливає, що

$$F := \frac{SSR/q}{SSE/(n - q - 1)} \sim F_{q, n-q-1}. \quad (8.14)$$

Подивимось, які значення є критичними для статистики F . За умови виконання H_0 ,

$$\begin{aligned} \hat{\mathbf{y}}'\mathbf{y} &= \hat{\beta}'X'\mathbf{y} = \begin{pmatrix} \hat{\beta}_0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_{11} & x_{21} & \dots & x_{1n} \\ & & \ddots & \\ x_{1q} & x_{2q} & \dots & x_{nq} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \\ &= \begin{pmatrix} \hat{\beta}_0 & \hat{\beta}_0 & \dots & \hat{\beta}_0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \\ &= \hat{\beta}_0 \sum_{i=1}^n y_i = \hat{\beta}_0 n \bar{y}. \end{aligned} \quad (8.15)$$

З іншого боку, за умови H_0 маємо $\hat{\beta}_0 = \bar{y}$. Отже, якщо виконується H_0 , то $SSR = 0$, а отже, $SST = SSE = \sum_{i=1}^n y_i^2 - n(\bar{y})^2$. Це означає, що критичними значеннями для F є великі значення. Тобто, фіксуючи рівень надійності α , ми відхиляємо H_0 , якщо $F > F_{\alpha, q, n-q-1}$.

По аналогії з (6.14), відношення SSR до SST є частиною дисперсії, яка пояснюється регресією, називається коефіцієнтом детермінації (R squared) і позначається R^2 :

$$R^2 = 1 - \frac{SSE}{SST} = \frac{SSR}{SST} \in (0, 1). \quad (8.16)$$

Якщо справедлива H_0 , то коефіцієнт детермінації має бути близьким до 0, тобто \mathbf{y} не залежить від координат x_1, \dots, x_q . Іншими словами, лінійна регресія погано пояснює значення \mathbf{y} . В багатовимірному випадку має сенс врахувати кількість векторів та кількість спостережень у вибірці, тому використовують скоригований коефіцієнт детермінації (adjusted R squared):

$$R_{adj}^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1} \in (0, 1). \quad (8.17)$$

Статистику F також можна зобразити за допомогою R^2 :

$$F = \frac{n - q - 1}{q} \frac{R^2}{1 - R^2}.$$

8.2 Багатовимірна регресія

У багатовимірному випадку маємо спостереження

$$Y = \begin{pmatrix} y_{11} & y_{12} & \dots & y_{1p} \\ y_{21} & y_{22} & \dots & y_{2p} \\ \dots \\ y_{n1} & y_{n2} & \dots & y_{np} \end{pmatrix} = \begin{pmatrix} \mathbf{y}'_1 \\ \mathbf{y}'_2 \\ \dots \\ \mathbf{y}'_n \end{pmatrix}.$$

Позначимо

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1q} \\ 1 & x_{21} & x_{22} & \dots & x_{2q} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nq} \end{pmatrix}, \quad B = \begin{pmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0p} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1p} \\ \dots & \dots & \dots & \dots \\ \beta_{q1} & \beta_{q2} & \dots & \beta_{qp} \end{pmatrix}, \quad \mathcal{E} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1p} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2p} \\ \dots & \dots & \dots & \dots \\ \varepsilon_{n1} & \varepsilon_{n2} & \dots & \varepsilon_{np} \end{pmatrix}.$$

Тоді модель багатовимірної лінійної регресії можна записати у наступному вигляді:

$$Y = XB + \mathcal{E}. \quad (8.18)$$

Аналогічно одновимірному випадку, $\mathcal{E}_i \sim N_p(0, \Sigma)$, $\mathcal{E}_i = (\varepsilon_{1i}, \dots, \varepsilon_{ni})'$,

1. $\mathbb{E}Y = XB$,
2. $\text{cov}(\mathbf{y}_i) = \Sigma$, $i = 1, \dots, n$.
3. $\text{cov}(\mathbf{y}_i, \mathbf{y}_j) = 0$ при $i \neq j$.

Наступна теорема є узагальненням Теореми 8.1.

Теорема 8.2. *Оцінкою методу найменших квадратів матриці B є*

$$\hat{B} = (X'X)^{-1}X'Y, \quad (8.19)$$

тобто

$$\hat{B} = \arg \min E = \arg \min \mathcal{E}'\mathcal{E} = (Y - X\hat{B})'(Y - X\hat{B}) =: E.$$

Ця оцінка має наступні властивості:

1. $\mathbb{E}\hat{B} = B$ (оцінка є незміщеною);
2. дисперсія $\mathbb{D}\hat{\beta}_{ij}$ є мінімальною у класі всіх незміщених оцінок (теорема Гауса-Маркова);
3. величини $\hat{\beta}_{ij}$ є корельованими.

Оскільки стовбчики в B є корельованими, ми не можемо використовувати F -тести як в простій регресії, тому потрібен інший механізм тестування.

Аналогом матриці SSE є

$$E = (Y - X\hat{B})'(Y - X\hat{B}). \quad (8.20)$$

Позначимо

$$S_e := \frac{E}{n - q - 1}. \quad (8.21)$$

Ця оцінка є незміщеною, $\mathbb{E}S_e = \Sigma$. Зобразимо матрицю B у вигляді

$$B = \begin{pmatrix} \beta'_0 \\ B_1 \end{pmatrix} = \begin{pmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0p} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1p} \\ \dots & \dots & & \\ \beta_{q1} & \beta_{q2} & \dots & \beta_{qp} \end{pmatrix},$$

та перевіримо гіпотезу

$$H_0: B_1 = 0, \text{ проти альтернативи } H_1: B_1 \neq 0. \quad (8.22)$$

Розкладемо сумарну похибку у суму матриць E та H :

$$Y'Y - n\bar{Y}\bar{Y}' = (Y'Y - \hat{B}'X'Y) + (\hat{B}'X'Y - n\bar{Y}\bar{Y}') = E + H.$$

Використовуючи матриці E та H , обчислимо статистику Уілкса $\Lambda = \frac{|E|}{|E+H|}$. За умови виконання гіпотези H_0 , Λ близьке до 1, а отже, критичні значення Λ – це малі значення. Як і в Розділі 5, можна використати статистику (5.16) для перевірки гіпотези H_0 .

Приклад 8.1. Розглянемо приклад з $p = 2, q = 3$. Якщо $B_1 = 0$, то $b_{ij} = 0, i \neq 0$, а тоді

$$\hat{B} = \begin{pmatrix} \hat{\beta}_{01} & \hat{\beta}_{02} \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \end{pmatrix}.$$

Далі,

$$(\hat{X}\hat{B})'Y = \begin{pmatrix} \hat{\beta}_{01} & \hat{\beta}_{01} & \dots & \hat{\beta}_{01} \\ \hat{\beta}_{02} & \hat{\beta}_{02} & \dots & \hat{\beta}_{02} \end{pmatrix} Y = \begin{pmatrix} \hat{\beta}_{01} \sum_{i=1}^n y_{i1} & \hat{\beta}_{01} \sum_{i=1}^n y_{i2} \\ \hat{\beta}_{02} \sum_{i=1}^n y_{i1} & \hat{\beta}_{02} \sum_{i=1}^n y_{i2} \end{pmatrix}.$$

З іншого боку,

$$\begin{aligned}\bar{Y}\bar{Y}' &= \begin{pmatrix} \frac{1}{n} \sum_{k=1}^n y_{k1} \\ \frac{1}{n} \sum_{k=1}^n y_{k2} \end{pmatrix} \begin{pmatrix} \frac{1}{n} \sum_{k=1}^n y_{k1} & \frac{1}{n} \sum_{k=1}^n y_{k2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{n^2} (\sum_{k=1}^n y_{k1})^2 & \frac{1}{n^2} (\sum_{k=1}^n y_{k1}) (\sum_{k=1}^n y_{k2}) \\ \frac{1}{n^2} (\sum_{k=1}^n y_{k1}) (\sum_{k=1}^n y_{k2}) & \frac{1}{n^2} (\sum_{k=1}^n y_{k2})^2 \end{pmatrix}\end{aligned}$$

За умови виконання гіпотези H_0 ,

$$\begin{pmatrix} y_{1i} \\ y_{2i} \\ \dots \\ \dots \\ y_{ni} \end{pmatrix} \approx \begin{pmatrix} \beta_{0i} \\ \beta_{0i} \\ \dots \\ \dots \\ \beta_{0i} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1i} \\ \varepsilon_{2i} \\ \dots \\ \dots \\ \varepsilon_{ni} \end{pmatrix}, \quad i = 1, 2,$$

де вектор-стовбчик $(\beta_{0i}, \dots, \beta_{0i})'$ має розмірність $(q+1) \times 1$. Отже, $\hat{\beta}_{0i} = \frac{1}{n} \sum_{k=1}^n y_{ki}$, $i = 1, 2$. Тоді $n\bar{Y}\bar{Y}' \approx (\hat{X}\hat{B})'Y$, звідки $H \approx 0$. Отже, ми приймаємо H_0 , якщо $\Lambda \approx 1$.

8.3 Приклади

Розглянемо наступний приклад, в якому дані взято з так званого "квартету Анскомбе"⁸ (Anscombe's Quartet). Квартет Анскомбе складається з чотирьох послідовностей, які задовільняють моделі лінійної регресії, але їхні графіки істотно відрізняються. **Зауважимо, що наведені нижче обчислення залежать від згенерованих нормальними розподіленими випадкових величин, тому мають суттєво ілюстративну мету.**

```
y<-c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68)
x1<-c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)

x2<-sqrt(y)+rnorm(length(y))

model=lm(y~x1+x2)      # задаємо модель лінійної регресії
model                      # отримаємо коротку інформацію про модель
```

Call:

```
lm(formula = y ~ x1 + x2)
```

Coefficients:

(Intercept)	x1	x2
0.9563	0.4800	0.8345

⁸<http://www.learnbymarketing.com/tutorials/explaining-the-lm-summary-in-r/>

Тобто, ми отримали оцінки на вільний доданок (Intercept), та на коефіцієнти при x_1 та x_2 (нижній рядок звіту). Переконаємося, що це дійсно оцінки, які задаються формулою (8.7):

```
X<-cbind(1,x1,x2)
est_b<-solve(t(X) %*% X) %*% (t(X) %*% y)
est_b
```

```
[,1]
0.9563469
x1 0.4800062
x2 0.8344520
```

Викликати інформацію про кожну колонку можна за допомогою наступних функцій:

```
coef(summary(model))[, "Std. Error"]
coef(summary(model))[, "t value"]
coef(summary(model))[, "Pr(>|t|)"]
```

Для того, щоб отримати більш детальну інформацію одночасно, використаємо

```
summary(model)
```

В результаті отримаємо

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q      Median      3Q      Max 
-1.3780 -0.6855 -0.0448  0.5484  1.5407 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.9563    1.5197   0.629   0.54670    
x1          0.4800    0.1062   4.521   0.00195 **  
x2          0.8345    0.4647   1.796   0.11026    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.107 on 8 degrees of freedom
Multiple R-squared:  0.7623, Adjusted R-squared:  0.7029 
F-statistic: 12.83 on 2 and 8 DF,  p-value: 0.00319
```

В розділі Residuals ми отримали мінімальне значення (-1.3780), перший квартіль, медіану та третій квартіль (-0.6855, -0.0448, 0.5484), та максимальне значення (1.5407) залишків.

Друга колонка – це оцінки на коефіцієнти, які ми отримали в est_b . Третя колонка – стандартне відхилення від теоретичного значення, тобто корені з діагональних елементів матриці

$$\frac{SSE}{n-k}(X'X)^{-1},$$

де $k = q + 1$ (ми підставили замість σ^2 її незміщену оцінку $\frac{SSE}{n-k}$, див. (8.9)).

Отримати третю колонку "вручну" можна наступним чином:

```
k=length(model$coefficients)      # кількість змінних +1, k=3, q=k-1
n=length(model$residuals)         # n=11
s2<-sum(model$residuals**2)/(n-k) # s2 = SSE/(n-k) n=11,
Sigma<- s2*solve(t(X)%*%X)       # матриця коваріацій оцінок коефіцієнтів
sqrt(diag(Sigma))                # стандартні відхилення похибок
```

За нульової гіпотези $H_0: b_i = 0, i = 0, 1, 2$, випадкові величини $t_i = \hat{b}_i/\sigma_i$ мають розподіл Стьюдента. Тому при перевірці гіпотези H_0 для кожного коефіцієнта ми обчислюємо значення t_i та відповідні $p-value$. Наприклад, перший елемент четвертого стовбчика – це

```
t_0<- est_b[1] / sqrt(diag(Sigma))[1]
t_0
```

0.6292939

Відповідно,

$$p-value = 2\mathbb{P}(|t| \geq t_1 | H_0).$$

де випадкова величина t має розподіл Стьюдента з $n - k$ ступенями свободи.

```
2*(1-pt(t1, n-k))
```

Іншими словами, якщо виконана H_0 , а t_α – це квантель рівня $1 - \alpha/2$, то має бути $|t_1| < t_\alpha$. Якщо ця нерівність не виконується, ми відхиляємо нульову гіпотезу.

0.5467026

Залишкова стандартна похибка (Residual Standard Error):

```
R_Err<- sqrt(SSE/(n-k))
R_Err
```

1.10729

Обчислимо коефіцієнт детермінації (Multiple R Squared):

```
SST<-sum((y-mean(y))**2)
R2<- (SST-SSE)/SST
R2
```

0.7623434

Обчислимо також скоригований коефіцієнт детермінації (Adjusted R Squared):

```
Adj_R2 <- 1 - (SSE / SST) * (n - 1) / (n - k)
Adj_R2
```

0.7029292

Нарешті, обчислимо F -статистику:

```
F1 <- ((SST - SSE) / (k - 1)) / (SSE / (n - k))
F1
```

12.83101

та порівняємо її із теоретичним квантілем:

```
pf(0.05, k - 1, n - k)
```

0.04847572

Отже, ми відхиляємо нульову гіпотезу про те, що коефіцієнти дорівнюють 0, оскільки отримане значення статистики значно більше за теоретичне.

Або, обчислимо $p - value$:

```
1 - pf(F1, k - 1, n - k)
```

0.003190067

(що співпадає з $p - value$, яке обчислено у вбудованій функції). Тут ми використали те, що розподіл Фішера – додатний, а отже, $p - value = \mathbb{P}(F > F_1 | H_0)$.

Нарешті, "Signif. codes:" визначає, наскільки впливає коефіцієнт на залежну змінну. Наприклад, "***" означає, що $p - value$ знаходиться в межах $[0, 0.001]$, "**" – в межах $(0.001, 0.01]$, "*" – в межах $(0.01, 0.05]$, "." – в межах $(0.05, 0.1]$, та порожнє значення означає, що $p - value$ знаходиться в межах $(0.1, 1.0]$. Образно кажучи, чим більше зірочок, тим більше залежність моделі від змінної. В останніх двох випадках ми приймаємо гіпотезу про те, що коефіцієнт дорівнює нулю, або що відповідна змінна не є значущою.

Розглянемо приклад багатовимірної лінійної регресії. Див. Rencher, Table 10.1, Examples 10.4.2, 10.5.1. В цій таблиці наведено дані результату хімічного експерименту, в якому X_1 , X_2 , X_3 – це дані, а Y_1 , Y_2 , Y_3 – відгук.

```
chem <- read.table('T10_1_CHEM.DAT',
                     col.names = c('V1', 'Y1', 'Y2', 'Y3', 'X1', 'X2', 'X3'))
head(chem)
```

	V1	Y1	Y2	Y3	X1	X2	X3
1	1	41.5	45.9	11.2	162	23	3
2	2	33.8	53.3	11.2	162	23	8
3	3	27.7	57.5	12.7	162	30	5
4	4	21.7	58.8	16.0	162	30	8
5	5	19.9	60.6	16.2	172	25	5
6	6	15.0	58.0	22.6	172	25	8

Згрупуємо дані по X та Y та застосуємо функцію `lm()` для того, щоб побудувати модель лінійної регресії. Можна використовувати функцію `%>%`, яка прописана в пакеті `dplyr`. Фактично, функція `%>%` підставляє вираз в лівій частині в функцію, яка написана в правій (наприклад, в одновимірному випадку $x \%>% f$ означає $f(x)$).

```
library(dplyr)
chem_x <- chem %>% select(starts_with("X"))
x <- as.matrix(chem_x)
chem_y <- chem %>% select(starts_with("y"))
y <- as.matrix(chem_y)

chem.lm <- lm(y ~ x)
chem.lm
```

Call:

`lm(formula = y ~ x)`

Coefficients:

	[,1]	[,2]	[,3]
(Intercept)	332.1110	-26.0353	-164.0789
x1	-1.5460	0.4046	0.9139
x2	-1.4246	0.2930	0.8995
x3	-2.2374	1.0338	1.1535

Ми отримали оцінку матриці B (див. (8.19)):

$$\hat{B} = \begin{pmatrix} -1.5460 & 0.4046 & 0.9139 \\ -1.4246 & 0.2930 & 0.8995 \\ -2.2374 & 1.0338 & 1.1535 \end{pmatrix} \quad (8.23)$$

Для перевірки гіпотези H_0 використаємо MANOVA.

```
chem.manova <- summary(manova(y ~ x), test="Wilks")
chem.manova
```

	Df	Wilks	approx F	num Df	den Df	Pr(>F)	
x	3	0.033158	10.787	9	31.789	1.884e-07 ***	
Residuals	15						
Signif. codes:	0	'***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Розберемо, що в цій таблиці:

$$nu_H = \text{degrees of freedom in the model} = q = 3$$

$$nu_E = \text{degrees of freedom in the residuals} = n - q - 1 = 19 - 3 - 1 = 15$$

$$numDf = df10 = 9, denDf = df2 = 31.789,$$

approx F- це статистика (5.16).

Ми відхиляємо H_0 , оскільки $p-value$ (тобто $\Pr(>F)$) < 0.05 .

9 Лінійний дискримінантний аналіз

Література: [R02, Гл.8]. Пакети MASS, car, dplyr.

9.1 Побудова дискримінантної функції

Розглянемо наступну задачу. Потрібно знайти лінійну функцію, за допомогою якої можна було б розбити вибірку на 2 або більше груп. Дискримінантна функція – це лінійна комбінація змінних, яка найкращим чином розділяє вибірку на групи.

Припустимо, що ми маємо 2 нормально розподілені популяції $\mathbf{y}_{11}, \mathbf{y}_{12}, \dots, \mathbf{y}_{1n_1}$ та $\mathbf{y}_{21}, \mathbf{y}_{22}, \dots, \mathbf{y}_{2n_2}$ в \mathbb{R}^p , з однаковою коваріаційною матрицею Σ , але різними середніми $\boldsymbol{\mu}_1$ та $\boldsymbol{\mu}_2$.

Дискримінантна функція є лінійною комбінацією координат, яка максимізує відстань між двома (перетвореними) групами векторів. Позначимо $z = \mathbf{a}'\mathbf{y}$:

$$\begin{aligned} z_{1i} &= \mathbf{a}'\mathbf{y}_{1i} = a_1y_{1i1} + a_2y_{1i2} + \dots + a_py_{1ip}, & i &= 1, \dots, n_1, \\ z_{2i} &= \mathbf{a}'\mathbf{y}_{2i} = a_1y_{2i1} + a_2y_{2i2} + \dots + a_py_{2ip}, & i &= 1, \dots, n_2, \end{aligned} \tag{9.1}$$

або

$$\begin{pmatrix} \mathbf{y}_{11} \\ \mathbf{y}_{12} \\ \vdots \\ \mathbf{y}_{1n_1} \end{pmatrix} \rightsquigarrow \begin{pmatrix} z_{11} \\ z_{12} \\ \vdots \\ z_{1n_1} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{y}_{21} \\ \mathbf{y}_{22} \\ \vdots \\ \mathbf{y}_{2n_1} \end{pmatrix} \rightsquigarrow \begin{pmatrix} z_{21} \\ z_{22} \\ \vdots \\ z_{2n_1} \end{pmatrix},$$

Тут $\mathbf{y}_{1i}, i = 1, \dots, n_1$ та $\mathbf{y}_{2i}, i = 1, \dots, n_2$, є векторами, а $z_{1i}, i = 1, \dots, n_1$ та $z_{2i}, i = 1, \dots, n_2$, склярами.

Позначимо

$$\bar{z}_{1\cdot} = \sum_{i=1}^{n_1} \frac{z_{1i}}{n_1} = \mathbf{a}'\bar{\mathbf{y}}_{1\cdot}, \quad \bar{z}_{2\cdot} = \sum_{i=1}^{n_2} \frac{z_{2i}}{n_2} = \mathbf{a}'\bar{\mathbf{y}}_{2\cdot},$$

де

$$\bar{\mathbf{y}}_{1\cdot} = \sum_{i=1}^{n_1} \frac{\mathbf{y}_{1i}}{n_1}, \quad \bar{\mathbf{y}}_{2\cdot} = \sum_{i=1}^{n_2} \frac{\mathbf{y}_{2i}}{n_2}.$$

Значення $\bar{z}_{1\cdot}$ і $\bar{z}_{2\cdot}$ є "централами" груп 1 та 2. Наша задача – обрати вектор \mathbf{a} так, щоб максимізувати відстань між цими двома побудованими централами. Оберемо вектор \mathbf{a} наступним чином:

$$\mathbf{a} := \arg \max_{\mathbf{a}} \left(\frac{\bar{z}_{1\cdot} - \bar{z}_{2\cdot}}{s_z} \right)^2 = \arg \max_{\mathbf{a}} \left(\frac{[\mathbf{a}'(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})]^2}{\mathbf{a}'S_{pl}\mathbf{a}} \right),$$

де $s_z^2 = \mathbf{a}' S_{pl} \mathbf{a}$. А саме, ми будемо максимізувати нормовану відстань $\bar{z}_{1\cdot} - \bar{z}_{2\cdot}$. Зауважимо, що для того, щоб існувала S_{pl}^{-1} , необхідно виконання нерівності $n_1 + n_2 - 2 > p$. Позначимо $\mathbf{v} := S_{pl}^{1/2} \mathbf{a}$. Тоді

$$\begin{aligned} \frac{[\mathbf{a}'(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})]^2}{\mathbf{a}' S_{pl} \mathbf{a}} &= \frac{|v|^2 |S_{pl}^{-1/2}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})|^2 |\cos(\mathbf{v}, S_{pl}^{-1/2}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot}))|}{|v|^2} \\ &= |S_{pl}^{-1/2}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})|^2 |\cos(\mathbf{v}, S_{pl}^{-1/2}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot}))|. \end{aligned}$$

Цей вираз буде максимальний, якщо $\cos(\mathbf{v}, S_{pl}^{-1/2}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})) = \pm 1$, тобто

$$\mathbf{v} \parallel S_{pl}^{-1/2}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot}),$$

або, що те саме,

$$S_{pl} \mathbf{a} = S_{pl}^{1/2} \mathbf{v} = \lambda(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot}),$$

звідки

$$\mathbf{a} = \lambda S_{pl}^{-1}(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot}).$$

Без обмеження загальності, покладемо $\lambda = 1$ (воно все одно скоротиться завдяки нормуванню за допомогою знаменника). При такому виборі вектору \mathbf{a} отримаємо

$$\max \left(\frac{\bar{z}_{1\cdot} - \bar{z}_{2\cdot}}{s_z} \right)^2 = (\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})' S_{pl}^{-1} (\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot}).$$

Отже, з точністю до множника, максимум є статистикою Хотеллінга T^2 . А отже, можна перевірити гіпотезу про рівність середніх: $H_0: \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$.

Припустимо, що нам треба класифікувати дані на $k \geq 2$ групи. Тоді алгоритм дії наступний.

1. Якщо груп більше, ніж дві, то потрібно більше ніж дві дискримінантні функції, які б описували розбиття на групи. Якщо точки p -вимірного простору спроектувати на площину, задану першими двома дискримінантними функціями, то ми отримаємо найкраще можливе наближення розбиття на групи.
2. Знайдемо множину вихідних змінних, використання яких дозволяє розбити на групи максимально якісно.
3. Впорядкуємо змінні по мірі ваги внеску до процедури розбиття.
4. Інтерпретуємо нові змінні, які задаються за допомогою дискримінантних функцій.
5. Виконуємо аналіз MANOVA.

Як у випадку $k \geq 2$ обирати вектор(и) \mathbf{a} ? Потрібно знайти аналог виразу $\frac{\bar{z}_{1\cdot} - \bar{z}_{2\cdot}}{s_z}$ у багатовимірному випадку.

Пригадаємо, як виглядає матриця H у випадку двох груп (див. (7.1)):

$$H = \sum_{i=1}^2 n_i (\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})(\bar{\mathbf{y}}_{i\cdot} - \bar{\mathbf{y}}_{\cdot\cdot})' = \frac{n_1 n_2}{n_1 + n_2} (\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})(\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})'. \quad (9.2)$$

Обчислимо $\mathbf{a}'H\mathbf{a}$:

$$\mathbf{a}'H\mathbf{a} = \frac{n_1 n_2}{n_1 + n_2} (\bar{z}_{1\cdot} - \bar{z}_{2\cdot})^2.$$

(оскільки $(\bar{z}_{1\cdot} - \bar{z}_{2\cdot})' = (\bar{\mathbf{y}}_{1\cdot} - \bar{\mathbf{y}}_{2\cdot})'\mathbf{a} \in \mathbb{R}$, тобто є числом!). Отже, $\mathbf{a}'H\mathbf{a}$ – це з точністю до множника відстань в квадраті між проекціями вибірки на вектор \mathbf{a} .

Аналогічно, при $k = 2$ можна трасформувати E (див. (7.2)) наступним чином:

$$E = \sum_{i=1}^2 \sum_{j=1}^{n_i} (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})' = (n_1 + n_2 - 2)S_{pl}. \quad (9.3)$$

Домножаючи на вектори \mathbf{a}' і \mathbf{a} відповідно, отримаємо

$$\mathbf{a}'E\mathbf{a} = (n_1 + n_2 - 2)\mathbf{a}'S_{pl}\mathbf{a}.$$

Іншими словами, $\mathbf{a}'E\mathbf{a}$ – це з точністю до множника дисперсія проекції. Отже,

$$\frac{(\bar{z}_{1\cdot} - \bar{z}_{2\cdot})^2}{\mathbf{a}'S_{pl}\mathbf{a}} = \frac{(n_1 + n_2)(n_1 + n_2 - 2)}{n_1 n_2} \cdot \frac{\mathbf{a}'H\mathbf{a}}{\mathbf{a}'E\mathbf{a}}.$$

З іншого боку, $\mathbf{a}'H\mathbf{a}$ та $\mathbf{a}'E\mathbf{a}$ можна записати через $SSH(z)$ та $SSE(z)$, які застосовані до змінних z , а не до Y :

$$\begin{aligned} \mathbf{a}'H\mathbf{a} &= \sum_{i=1}^2 (\bar{z}_{i\cdot} - \bar{z}_{..})^2 = SSH(z), \\ \mathbf{a}'E\mathbf{a} &= \sum_{i=1}^2 \sum_{j=1}^{n_i} (z_{ij} - \bar{z}_{i\cdot})^2 = SSE(z). \end{aligned}$$

Ці формули мають узагальнення на випадок k груп. Отже, у випадку $k > 2$ можна максимізувати не $\left(\frac{\bar{z}_{1\cdot} - \bar{z}_{2\cdot}}{s_z}\right)^2$, а співвідношення

$$\lambda = \frac{SSH(z)}{SSE(z)} = \frac{\mathbf{a}'H\mathbf{a}}{\mathbf{a}'E\mathbf{a}}. \quad (9.4)$$

Знайдемо a , для яких рівняння (9.4) має розв'язок. Запишемо рівняння $\mathbf{a}'H\mathbf{a} = \lambda\mathbf{a}'E\mathbf{a}$ у вигляді

$$\mathbf{a}'(H\mathbf{a} - \lambda E\mathbf{a}) = 0. \quad (9.5)$$

Розв'язок $\mathbf{a} = 0$ не підходить, оскільки ми маємо тоді $\lambda = 0/0$. Іншими розв'язками є числа $\lambda_1, \dots, \lambda_s$, де $s = \text{rank}(E^{-1}H)$. Впорядкуємо ці λ_i . Тоді

$$\lambda_1 = \max_a \frac{\mathbf{a}'H\mathbf{a}}{\mathbf{a}'E\mathbf{a}}.$$

Нехай $\mathbf{a}_1, \dots, \mathbf{a}_s$ – власні вектори, що відповідають власним числам $\lambda_1, \dots, \lambda_s$. Дискримінантними функціями тоді є $z_1 = \mathbf{a}'_1 \mathbf{y}$, $z_2 = \mathbf{a}'_2 \mathbf{y}, \dots, z_s = \mathbf{a}'_s \mathbf{y}$. Функції z_i , $i = 1, \dots, s$, описують різницю між середніми $\bar{\mathbf{y}}_1, \dots, \bar{\mathbf{y}}_k$. Власні вектори \mathbf{a}_i є некорельзованими, але не ортогональними, оскільки матриця $E^{-1}H$ не є симетричною. Зауважимо також, що побудоване розбиття відповідає власним числам, а не початковому розбиттю на k груп. Функція, яка максимальне розбиває на групи – це дискримінантна функція, побудована

за першим власним вектором $z_1 := \mathbf{a}'_1 \mathbf{y}$. "Відносну важливість" дискримінантних функцій можна порівняти, порівнюючи внесок λ_i , а саме, відношення

$$\frac{\lambda_i}{\sum_{i=1}^s \lambda_i}, \quad i = 1, \dots, s.$$

Як правило, в тих задачах, що ми будемо розглядати, двох-трьох дискримінантних функцій досить для того, щоб розділити виборку на класи.

Розглянемо ще одну цікаву властивість функції \mathbf{a} , у випадку, коли ми розглядаємо класифікацію вибірки на дві групи. Виявляється, що вектор \mathbf{a} є колінеарним вектору оцінці параметрів у певній моделі лінійної регресії (див. [R02, Ch.5.6])!

Введемо фіктивні групові змінні ("dummy" group variable) $w_i = \frac{n_2}{n_1+n_2}$ для значень $\mathbf{y}_{11}, \mathbf{y}_{12}, \dots, \mathbf{y}_{1n_1}$ вибірки 1, та $w_i = -\frac{n_1}{n_1+n_2}$ для значень $\mathbf{y}_{11}, \mathbf{y}_{12}, \dots, \mathbf{y}_{1n_1}$ вибірки 2, $\mathbf{y}_{ij} \in \mathbb{R}^p$. Тоді, якщо просумувати всі n_1+n_2 змінних, отримаємо $\bar{w} = 0$. Розглянемо модель лінійної регресії: $\mathbf{b} \in \mathbb{R}^p$, $\varepsilon_i \in N(0, 1)$:

$$\mathbf{w} = (\mathbf{y} - \bar{\mathbf{y}}_{..})' \mathbf{b} + \varepsilon,$$

а саме,

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n_1+n_2} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_{11} - \bar{\mathbf{y}}_{..} \\ \mathbf{y}_{12} - \bar{\mathbf{y}}_{..} \\ \vdots \\ \mathbf{y}_{2n_2} - \bar{\mathbf{y}}_{..} \end{pmatrix}' \mathbf{b} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_{n_2} \end{pmatrix}. \quad (9.6)$$

В наших попередніх позначеннях: $\mathbf{w} = X\mathbf{b} + \varepsilon$. Зауважимо, що

$$X'X = T = E + H,$$

тобто насправді ми рахуємо (точністю до множника) вибіркову коваріацію між двома групами \mathbf{y}_{1j} , $1 \leq j \leq n_1$, та \mathbf{y}_{2j} , $1 \leq j \leq n_2$. Припустимо, що T не є сингулярною. З попереднього,

$$\hat{\mathbf{b}} = (X'X)^{-1} X' \mathbf{w}.$$

Покажемо, що вектор регресії $\hat{\mathbf{b}}$ є колінеарним \mathbf{a} та $\bar{\mathbf{y}}_{1.} - \bar{\mathbf{y}}_{2.}$:

$$\hat{\mathbf{b}} \parallel \mathbf{a}, \quad \hat{\mathbf{b}} \parallel \bar{\mathbf{y}}_{1.} - \bar{\mathbf{y}}_{2.}. \quad (9.7)$$

Маємо:

$$\begin{aligned} X' \mathbf{w} &= \frac{n_2}{n_1+n_2} \sum_{j=1}^{n_1} (\mathbf{y}_{1j} - \bar{\mathbf{y}}_{..}) - \frac{n_1}{n_1+n_2} \sum_{j=1}^{n_2} (\mathbf{y}_{2j} - \bar{\mathbf{y}}_{..}) \\ &= \frac{n_1 n_2}{n_1+n_2} (\bar{\mathbf{y}}_{1.} - \bar{\mathbf{y}}_{2.}). \end{aligned}$$

Зауважимо, що власні вектори матриць T , T^{-1} , $E^{-1}H$, HE^{-1} , E , E^{-1} - колінеарні (див. Вправу 9.1)! З іншого боку, у випадку 2-х груп з (9.5) випливає, що (з точністю до множника) вектор $\mathbf{a} = S_{pl}^{-1}(\bar{\mathbf{y}}_{1.} - \bar{\mathbf{y}}_{2.})$ співпадає з власним вектором $E^{-1}H$ (оскільки обидва цих вектори є розв'язками задачі максимізації). Оскільки $S_{pl} = \frac{E}{n_1+n_2-2}$ (див. (9.3)), то застосувавши E до $\mathbf{a} = S_{pl}^{-1}(\bar{\mathbf{y}}_{1.} - \bar{\mathbf{y}}_{2.})$ отримаємо

$$(n_1 + n_2 - 2)(\bar{\mathbf{y}}_{1.} - \bar{\mathbf{y}}_{2.}) = E\mathbf{a} = \lambda_E \mathbf{a},$$

де λ_E є власним числом матриці E . Отже, $\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 \parallel \mathbf{a}$. Тоді

$$\begin{aligned} T^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) &= \frac{\lambda_E T^{-1} \mathbf{a}}{(n_1 + n_2 - 2)} = \frac{\lambda_E (I + E^{-1}H)^{-1} E^{-1} \mathbf{a}}{(n_1 + n_2 - 2)} \\ &= \frac{(I + E^{-1}H)^{-1} \mathbf{a}}{(n_1 + n_2 - 2)} = \frac{\mathbf{a}}{(1 + \lambda)(n_1 + n_2 - 2)} \end{aligned}$$

де λ є максимальним власним числом матриці $E^{-1}H$. Отже,

$$\hat{\mathbf{b}} = \frac{n_1 n_2}{(n_1 + n_2)(n_2 + n_2 - 2)(1 + \lambda)} \mathbf{a}.$$

Але у випадку двох груп $\Lambda = \frac{1}{1+\lambda}$. Отже,

$$\hat{\mathbf{b}} = \frac{n_1 n_2}{(n_1 + n_2)(n_2 + n_2 - 2)} \Lambda \mathbf{a}.$$

Зauważення 9.1. Лінійний дискримінантний аналіз відноситься до "навчання з учителем" (*supervised learning*): ми максимізуємо відстань між класами даних. Зауважимо, що ми припускали, що дані є нормально розподіленими.

Задача 9.1. Довести, що власні вектори матриць T , T^{-1} , $E^{-1}H$, HE^{-1} , $E E^{-1}$ - колінеарні.

9.2 Приклади

1. Розглянемо Приклад 8.2 з [R02]⁹. Зчитаємо наступні дані, див. [R02, Таб.8.1]. В цій таблиці наведені максимальна міцність y_1 і межа текучості y_1 для сталі, прокатаної при двох температурах T_1 та T_2 . Наша задача – вдало розділити дані на 2 групи.

```
Temp<-read.table("T8_1_STEEL.DAT")
Temp
```

	V1	V2	V3
[1,]	1	33	60
[2,]	1	36	61
[3,]	1	35	64
[4,]	1	38	63
[5,]	1	40	65
[6,]	2	35	57
[7,]	2	36	59
[8,]	2	38	59
[9,]	2	39	61
[10,]	2	41	63
[11,]	2	43	65
[12,]	2	41	59

⁹<https://rpubs.com/aaronsc32/classification-linear-discriminant-analysis>

```

Temp<-as.matrix(Temp)           # зчитаємо у вигляді матриці
Temp1<-Temp[1:5,2:3]            # Температури 1
Temp2<-Temp[6:12,2:3]           # Температури 2

plot(Temp2, col = "red", xlim=c(32,48), ylim=c(55,68)) # точки Temp2
points(Temp1, col= "blue")      # точки Temp1

```

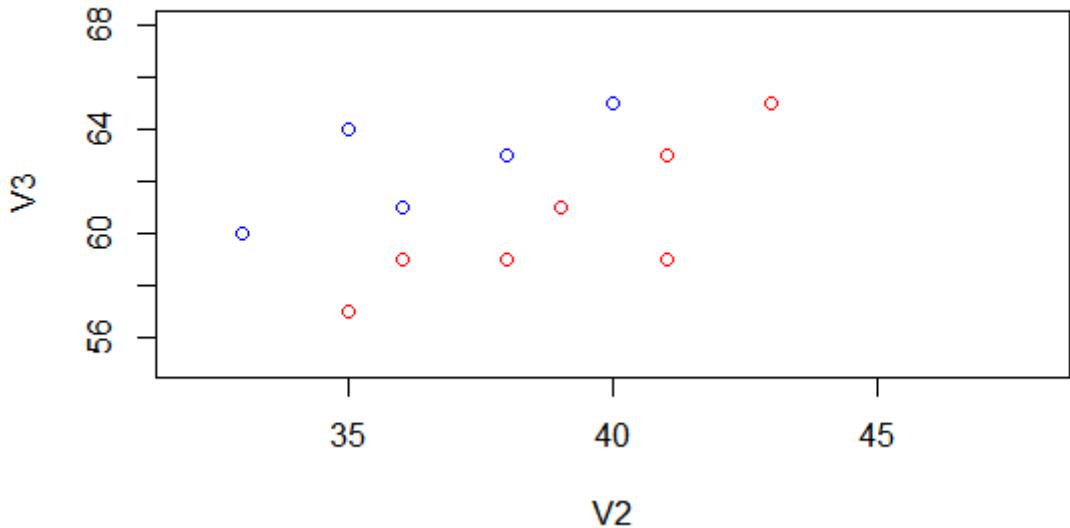


Рис. 4: Зображення в старих координатах

Схоже на те, що вибірку можна розділити на 2 групи. Для цього побудуємо дискримінантну функцію.

```

y1<-c(mean(Temp1[,1]),mean(Temp1[,2])) # вектор середніх Temp1
y2<-c(mean(Temp2[,1]),mean(Temp2[,2])) # вектор середніх Temp2

# обчислимо зважену коваріаційну матрицю

S1<-cov(Temp1)
S2<-cov(Temp2)
d1<-length(Temp1[,1])   # кількість елементів в Temp1
d2<-length(Temp2[,1])   # кількість елементів в Temp2

Sp1<-((d1-1)*S1+ (d2-1)*S2)/(d1+d2-2) # зважена коваріаційна матриця

```

	V2	V3
V2	7.92	5.680000
V3	5.68	6.291429

```
a<-solve(Spl) %*% (y1-y2) # дискримінантна функція
z1<- Temp1 %*% a
z2<- Temp2 %*% a
```

Отримаємо:

```
> a
 [,1]
V2 -1.633377
V3 1.819779

> z1
 [,1]
[1,] 55.28530
[2,] 52.20494
[3,] 59.29766
[4,] 52.57775
[5,] 52.95055

> z2
 [,1]
[1,] 46.55921
[2,] 48.56539
[3,] 45.29863
[4,] 47.30481
[5,] 47.67762
[6,] 48.05042
[7,] 40.39850
```

Розділимо на 2 групи виходячи з того, де знаходяться елементи по відношенню до середнього значення

```
zmean<-0.5*(mean(z1)+mean(z2))
t<- as.matrix(Temp[,2:3])
z<-t%*%a
z
```

```
 [,1]
[1,] 55.28530
[2,] 52.20494
[3,] 59.29766
[4,] 52.57775
[5,] 52.95055
[6,] 46.55921
[7,] 48.56539
[8,] 45.29863
[9,] 47.30481
```

```
[10,] 47.67762
[11,] 48.05042
[12,] 40.39850
```

Розділимо тепер елементи t на групи 1 та 2:

```
group <- ifelse(z[,1] > zmean, 1, 2)
group
```

Отже, ми можемо віднести перші 5 елементів до першої групи, а останні 7 – до другої.

```
1 1 1 1 1 2 2 2 2 2 2
```

2. Розглянемо ще один приклад, а саме, Приклад 8.4.1 з [R02]. Для цього завантажимо наступні бібліотеки: **car MASS**, **dplyr**, та завантажимо Таблицю 8.3 (див. Глава 8). В таблиці знаходяться вимірювання параметрів шоломів в залежності від того, до якої групи відноситься людина (гравець шкільної команди, гравець команди коледжа, або людина взагалі не грає в футбол). Для простоти, ми перейменуємо колонки цієї таблиці.

```
Foot<-read.table('T8_3_FOOTBALL.DAT',
  col.names = c('Group', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7'))
head(Foot)
```

Визначимо залежні змінні (у вигляді матриці) та визначимо змінну, за якою робимо групування:

```
dependent.vars3<-cbind(Foot$V2,Foot$V3,Foot$V4,Foot$V5,Foot$V6,Foot$V7)

DepVar<-as.matrix(dependent.vars3)

Foot$Group <- as.factor(Foot$Group)
```

Застосуємо функцію `lda`:

```
lda1<-lda(Foot$Group ~ ., data=Foot)
```

```
Call:
lda(Foot$Group ~ ., data = Foot)
```

```
Prior probabilities of groups:
1          2          3
0.3333333 0.3333333 0.3333333
```

```
Group means:
      V2        V3        V4        V5        V6        V7
1 15.20 58.93700 20.10833 13.08333 14.73333 12.26667
2 15.42 57.37967 19.80333 10.08000 13.45333 11.94333
3 15.58 57.77000 19.81000 10.94667 13.69667 11.80333
```

Coefficients of linear discriminants:

```
LD1           LD2
V2  0.948423100  1.4067750094
V3 -0.003639865 -0.0005126312
V4 -0.006439599 -0.0286176430
V5 -0.647483088  0.5402700415
V6 -0.504360916 -0.3839132257
V7 -0.828535064 -1.5288556226
```

Proportion of trace:

```
LD1   LD2
0.943 0.057
```

Вектори $LD1$ та $LD2$ – це перші 2 власні вектори матриці. Відповідно. 0.943 та 0.057 – це відношення $\frac{\lambda_1}{\sum_{i=1}^5 \lambda_i}$ та $\frac{\lambda_2}{\sum_{i=1}^5 \lambda_i}$.

Цей самий аналіз можна зробити вручну наступним чином. Застосуємо функцію manova для того, щоб знайти матриці E та H , та знайти власні числа та власні вектори матриці $E^{-1}H$.

```
Foot.manova<-summary(manova(dependent.vars3~Foot$Group), test="Wilks")
H<-Foot.manova$SS[1]
E<-Foot.manova$SS[2]
```

В практичних заняттях, що знаходяться за посиланням на GitHub, можна знайти ще один алгоритм отримання H та E вручну.

Запишемо E та H у вигляді матриць та обчислимо власні вектори та власні числа матриці $E^{-1}H$ (зверніть увагу на синтаксис при визначенні матриці $H1$):

```
H1<-as.matrix(H$'Foot$Group')
E1<-as.matrix(E$Residuals)
eigen(solve(E1) %*% H1)$values          # власні числа
lam1<-eigen(solve(E1) %*% H1)$values[1]    # перше власне число
lam2<-eigen(solve(E1) %*% H1)$values[2]    # друге власне число
```

Перші 2 власних числа відносно велики: 1.9177 та 0.1159, інші – порівняно малі. Крім того, нам потрібно розбити дані на 3 групи, тому 2-х власних векторів досить.

```
eigen(solve(E1) %*% H1)$vectors          # нормовані власні вектори
a1<-eigen(solve(E1) %*% H1)$vectors[,1]    # перший власний вектор
a2<-eigen(solve(E1) %*% H1)$vectors[,2]    # другий власний вектор
```

Зауважимо, що ці власні вектори є нормованими, в той час як функція lda знаходить ненормовані власні вектори.

Подивитися атрибути lda1 можна наступним чином:

```
attributes(lda1)
```

```
$names
```

```
'prior' 'counts' 'means' 'scaling' 'lev' 'svd' 'N' 'call' 'terms' 'xlevels',
$class
'lda'
```

Наприклад:

```
lda1$scaling                                # власні вектори
lda1$scaling[,1]                             # перший власний вектор
b1 <- as.vector(lda1$scaling[,1])
b2 <- as.vector(lda1$scaling[,2])
b1 / (sqrt(sum(b1^2)))                      # це як раз a1
```

Побудуємо дискриміантні функції і побачимо, як виглядає картинка після розділення на групи за допомогою цих функцій:

```
z11 <- DepVar[1:30,] %*% b1
z12 <- DepVar[1:30,] %*% b2
z21 <- DepVar[31:60,] %*% b1
z22 <- DepVar[31:60,] %*% b2
z22 <- DepVar[31:60,] %*% b2
z31 <- DepVar[61:90,] %*% b1
z32 <- DepVar[61:90,] %*% b2

plot(z11,z12, col = "red", xlim= c(-15,-5), ylim = c(0,8))
points(z21,z22, col= "blue")
points(z31,z32, col="green")
```

Отже, червону групу можна відокремити від синьої та зеленої досить просто, в той час як синя та зелена досить сильно перемішані.

10 Метод головних компонент

Література: [R02, Гл.12]. Пакети **stats**, **ggfortify**, **jpeg**.

При великій розмірності матриця дисперсій стає дуже великою і, що ще гірше, вона може бути виродженою. Отже, потрібно зменшити кількість змінних та знайти змінні, які є базисними. Метод головних компонент може бути застосований до будь-якого розподілу, не обов'язково нормального. Зауважимо, що тести в MANOVA використовують $E^{-1}H$, а отже, ця матриця не має бути сингулярною. Тому перед тим, як робити аналіз MANOVA, потрібно зменшити розмірність методом головних компонент.

Метод головних компонент – це приклад ”навчання без учителя” (тобто unsupervised learning). Ми фактично проектуємо всі дані на інший простір і при цьому знаходимо осі, де дані мають найбільше розсіювання. Тоді саме ці данні будуть мати найбільший внесок в загальну дисперсію.

На відміну від лінійного дискриміантного аналізу, метод головних компонент працює не тільки для нормально розподілених спостережень. Тому в цьому розділі ми не робимо припущення щодо нормальності.

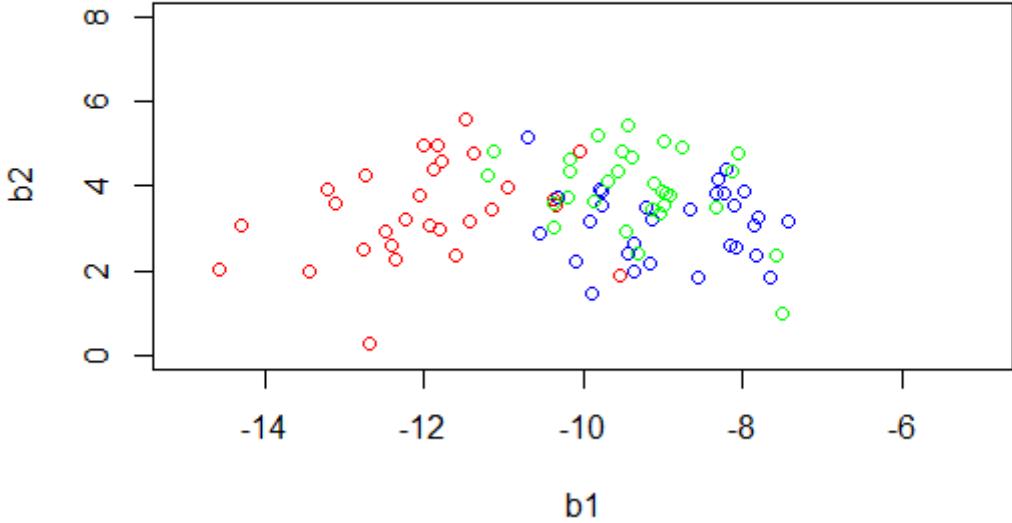


Рис. 5: Зображення в нових координатах

10.1 Геометричний підхід

Якщо змінні є корельованими, то еліпсоїд розсіювання не є орієнтованим вздовж жодної з осей. Розглянемо матрицю повороту A , яка повертає еліпс так, щоб в нових координатах (головних компонентах, principal components) змінні були некорельованими.

Припустимо, що \mathbf{y}_i є центрованими, інакше розглянемо $\mathbf{y}_i - \bar{\mathbf{y}}$. Нехай A є ортогональною матрицею, та розглянемо

$$\mathbf{z}_i = A\mathbf{y}_i, \quad i = 1, \dots, n. \quad (10.1)$$

Оскільки $AA' = I$, то

$$\mathbf{z}_i' \mathbf{z}_i = \mathbf{y}_i' A' A \mathbf{y}_i = \mathbf{y}_i' \mathbf{y}_i.$$

Таким чином, ортогональне перетворення A зберігає відстані від початку координат.

Нехай \mathbf{y} – вибірка спостережень $\mathbf{y}_i, i = 1, \dots, n$; утворимо з цієї вибірки \mathbf{z} за допомогою перетворення (10.1).

Знаходження головних осей еліпса еквівалентно знаходженню такої ортогональної матриці A , яка повертає осі таким чином, щоб нові змінні \mathbf{z}_i були некорельованими. Тоді матриця вибіркових дисперсій після перетворення має вигляд

$$S_{\mathbf{z}} = ASA' = \begin{pmatrix} s_{z_1}^2 & 0 & \cdot & 0 \\ 0 & s_{z_2}^2 & \cdot & 0 \\ 0 & \cdot & \ddots & s_{z_p}^2 \end{pmatrix}.$$

Тут S – вибіркова коваріаційна матриця $\mathbf{y}_1, \dots, \mathbf{y}_n$. Нехай C – матриця, утворена з нормованих (тобто $\mathbf{a}' \mathbf{a} = 1$) власних векторів. Використовуючи (1.1), маємо

$$C' S C = D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p).$$

$$A = C' = \begin{pmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_2 \\ \vdots \\ \mathbf{a}'_p \end{pmatrix}$$

Головними компонентами є змінні $z_1 = \mathbf{a}'_1 \mathbf{y}$, $z_2 = \mathbf{a}'_2 \mathbf{y}$, ..., $z_p = \mathbf{a}'_p \mathbf{y}$. Наприклад, $z_1 = a_{11}y_1 + a_{12}y_2 + \dots + a_{1p}y_p$. При цьому вибіркові дисперсії дорівнюють власним числам: $s_{z_i}^2 = \lambda_i$. Тому "частка поясненої дисперсії" – це і є відношення перших k власних чисел до всіх власних чисел:

$$\frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_p} = \frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{\text{trace}(S)}.$$

Таким чином, ми записуємо наші дані в новій системі координат $(\mathbf{a}_1, \dots, \mathbf{a}_k)$, які дають максимальний внесок в дисперсію, замість $(\mathbf{e}_1, \dots, \mathbf{e}_p)$. Якщо кілька змінних сильно корелювані, фактична розмірність значно менше, ніж p .

10.2 Алгебраїчний підхід

Нехай є спостереження \mathbf{y} з коваріаційною матрицею S . Ми шукаємо лінійну комбінацію векторів, яка максимізує дисперсію, тобто діагональні елементи S . Вибіркова дисперсія вектору $\mathbf{z} = \mathbf{a}' \mathbf{y}$ дорівнує $\mathbf{a}' S \mathbf{a}$. Оскільки $\mathbf{a}' S \mathbf{a}$ не досягає максимуму якщо норма $\|\mathbf{a}\|$ не обмежена, тому ми будемо шукати максимум

$$\lambda = \frac{\mathbf{a}' S \mathbf{a}}{\mathbf{a}' \mathbf{a}}.$$

Задача зводиться до задачі Лагранжа

$$L(\mathbf{a}) := \mathbf{a}' S \mathbf{a} - \lambda(\mathbf{a}' \mathbf{a} - 1) \mapsto \max.$$

Для того, щоб максимізувати цей вираз, використаємо наступне твердження (див. [PP12, (108)]):

$$\frac{\partial}{\partial X} X' A X = \frac{\partial}{\partial X} \text{trace}(X' A X) = A X + A' X. \quad (10.2)$$

Використовуючи (10.2) та враховуючи симетрію S , отримаємо

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{a}} &= 2S\mathbf{a} - 2\lambda\mathbf{a} = 0, \\ \frac{\partial L}{\partial \lambda} &= \mathbf{a}' \mathbf{a} - 1 = 0. \end{aligned}$$

Отже, \mathbf{a} є розв'язком системи рівнянь

$$S\mathbf{a} = \lambda\mathbf{a}, \quad \mathbf{a}' \mathbf{a} = 1,$$

що співпадає з результатом, отриманим у попередньому підрозділі.

Нехай \mathbf{a}_1 – власний вектор, який відповідає власному числу λ_1 . Тоді дисперсія дисперсія випадкової величини $z_1 = \mathbf{a}'_1 \mathbf{y}$ є максимальною.

На відміну від дискриміантного аналізу, ми не обчислюємо тут обернену матрицю, а отже, S може бути сингулярною, а деякі власні числа (а отже, і власні вектори) можуть бути нульовими. Наприклад, матриця сингулярна, якщо $n < p$, тобто розмір вибірки менше, ніж розмірність простору. Далі так само, як і в геометричному підході, ми відбираємо власні вектори з найвищим рівнем впливу на величину дисперсії.

Є кілька підходів до того, яку кількість головних компонент досить обрати.

- Обрати ті компоненти, відповідні власні числа яких становлять мінімум 80% від усіх власних чисел, тобто ті вектори, для яких

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i} > 0.8.$$

Цей метод має той недолік, що ми не враховуємо якісну складову відбору компонент – може статися, що ми не врахуємо компоненту, яка не має кількісно великого внеску, але якісно є важливою.

- Обрати ті компоненти, які більше ніж середнє значення:

$$\lambda_i > \frac{1}{p} \sum_{i=1}^p \lambda_i, \quad i = 1, \dots, p.$$

Але цей метод має такий самий недолік, як і попередній.

- Графічний аналіз: аналіз scree plot. Зобразимо точки (i, λ_i) , $i = 1, \dots, p$, та оберемо ті λ_i , де ламана має злам, після чого ланки майже паралельні. Перевага цього способу в його наглядності, але по суті ми самі обираємо точку “зламу”.
- Протестувати важливість кожної компоненти. Розглянемо більш детально цей метод.

Спочатку ми перевіряємо гіпотезу

$$H_0 : \quad \Sigma = \text{diag}(\sigma_{11}^2, \sigma_{22}^2, \dots, \sigma_{pp}^2).$$

Якщо змінні незалежні, то виділяти далі головні компоненти не має сенсу.

Нехай γ_k є власними числами матриці коваріації Σ . Перевіримо гіпотезу

$$H_{0k} : \quad \gamma_{p-k+1} = \gamma_{p-k+2} = \dots = \gamma_p.$$

Тоді перші k компонент – це фактично наші головні компоненти, інші – це “шум”. Якщо справедлива гіпотеза H_{0k} , то перші власні вектори – це і є наші головні компоненти. Статистикою для перевірки цієї гіпотези є (див. [R02, (12.15)])

$$u = \left(n - \frac{2p+4}{6} \right) \left(k \ln \bar{\lambda} - \sum_{i=p-k+1}^p \ln \lambda_i \right),$$

де

$$\bar{\lambda} = \sum_{i=p-k+1}^p \frac{\lambda_i}{p}.$$

Якщо справедлива H_{0k} , то $u \sim \chi^2_\nu$, $\nu = \frac{(k-1)(k+2)}{2}$. Отже, ми відхиляємо гіпотезу H_{0k} , якщо $u > \chi^2_{\alpha, \nu}$.

Компоненти, що відповідають порівняно малим власним числам, можна інтерпретувати наступним чином. Ці власні числа – це дисперсії спостережень в нових координатах. Якщо ці дисперсії малі, це означає, що змінні наближено дорівнюють сталим, а отже, їх можна інтерпретувати як лінійно залежні.

Зауважимо, що головні компоненти, отримані за допомогою кореляційної матриці, відрізняються від головних компонент, які отримані за допомогою коваріаційної матриці. Якщо дисперсії і коваріації сильно відрізняються, то доцільно використовувати кореляційну матрицю, а не коваріаційну.

10.3 Приклади

1. Розглянемо Приклад 12.2.1 з [R02]. В наступній таблиці наведені вимірювання параметрів голови (довжина та ширина щелепа) для першого (колонки $V1$ та $V2$) та другого (відповідно, колонки $V3$ та $V4$) синів в родині.

```
sons <- read.table('T3_7_SONS.DAT')
sons
```

	V1	V2	V3	V4
1	191	155	179	145
2	195	149	201	152
3	181	148	185	149
4	183	153	188	149
5	176	144	171	142
6	208	157	192	152

Щоб проілюструвати, що знаходження головних компонент – це фактично ротація осей координат, наведемо дані для першого сина в координатах та в нових координатах, які являють собою власні вектори коваріаційної матриці.

```
son1<-matrix(cbind(sons$V1,sons$V2), nrow = 25, byrow = FALSE)
m1<-mean(sons$V1)
m2<-mean(sons$V2)
ybar<-c(m1,m2)
Sson1<-cov(son1) # коваріація
eigen(Sson1)$values # власні числа
eigen(Sson1)$vectors # власні вектори
a1<- -eigen(Sson1)$vectors[,1] # перша дискриміантна функція
a2<- -eigen(Sson1)$vectors[,2] # друга дискриміантна функція
```

Центруємо всю вибірку і побудуємо графік в старих (зелений) і нових (червоний) координатах (для цього ми завантажили пакет `grid` для того, щоб обрати розмір картинки): ***

```
son1_cntr<-matrix(c(son1[,1]-m1,son1[,2]-m2), nrow = 25, byrow = FALSE)
z1<- son1_cntr%*% a1
z2<- son1_cntr%*% a2
```

```

options(repr.plot.width=5, repr.plot.height=4)
plot(sons$V1-m1, sons$V2-m2, col="green")
points(z1, z2, col="red")

```

Можна помітити, що в нових осіх координат наші дані розташовані вздовж осі OX .

2. В наступній таблиці наведено результати 6 тестів 20 інженерів-студентів та 20 пілотів¹⁰. Ми наведемо перші 6 значень таблиці; тут 1 означає, що мова йде про студента, а 2-про пілота. Тому ми зробимо групування даних за принципом, чи є людина, що бере участь в експерименті, студентом чи пілотом.

```

pilots <- read.table('T5_6_PILOT.DAT',
col.names = c('Group', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6'))
head(pilots)
pilots$Group<- ifelse(pilots$Group == 1, 'Apprentice', 'Pilot')

```

	Group	V1	V2	V3	V4	V5	V6
1	1	121	22	74	223	54	254
2	1	108	30	80	175	40	300
3	1	122	49	87	266	41	223
4	1	77	37	66	178	80	209
5	1	140	35	71	175	38	261
6	1	108	37	57	241	59	245

За допомогою функції `eigen` отримаємо інформацію про власті числа та власні вектори матриці коваріацій:

```

S <- cov(pilots[,2:7])
sum(diag(S)) # сумарна дисперсія, тобто сума власних значень

s.eigen <- eigen(S)
s.eigen

```

```

eigen() decomposition
$values
[1] 1722.0424 878.3578 401.4386 261.0769 128.9051 50.3785

$vectors
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] -0.21165160 -0.38949336 0.88819049 0.03082062 -0.04760343 0.10677164
[2,]  0.03883125 -0.06379320 0.09571590 -0.19128493 -0.14793191 -0.96269790
[3,] -0.08012946  0.06602004 0.08145863 -0.12854488  0.97505667 -0.12379748
[4,] -0.77552673  0.60795970 0.08071120 0.08125631 -0.10891968 -0.06295166
[5,]  0.09593926 -0.01046493 0.01494473 0.96813856 0.10919120 -0.20309559
[6,] -0.58019734 -0.68566916 -0.43426141 0.04518327 0.03644629 -0.03572141

```

¹⁰див. також <https://rpubs.com/aaronsc32/principal-component-analysis>

Нагадаємо, що для того, щоб отримати власні вектори або власні числа, можна застосувати функції, відповідно, `s.eigen$vectors` та `s.eigen$values`.

Обчислимо пропорцію власних чисел до сумарної дисперсії $\lambda_i / (\sum \lambda_i)$:

```
x <- c()
for (s in s.eigen$values) {
    x <- c(x, s / sum(s.eigen$values))
}
print(x)
```

```
0.50027387 0.25517343 0.11662269 0.07584597 0.03744848 0.01463556
```

```
plot(s.eigen$values, xlab = 'Номер власного числа', ylab = 'Власні числа')
lines(s.eigen$values)
```

Намалюємо scree plot (див. Рис. 7):

```
plot(s.eigen$values, xlab = 'eigenvalue number', ylab = 'eigenvalues')
lines(s.eigen$values)
```

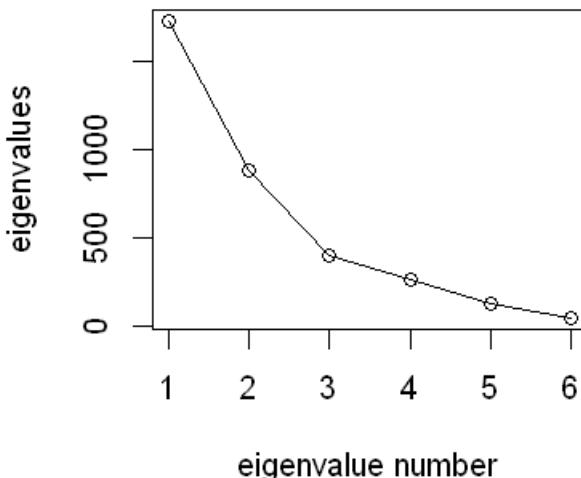


Рис. 6: Scree plot

З іншого боку, щоб отримати стандартні відхилення (тобто корені з власних чисел), можна застосувати функції `princomp` та `prcomp` (пакет `stats`) до `pilots[,2:7]`. Отримаємо:

```
princomp(pilots[, 2:7])
```

```
princomp(x = pilots[, 2:7])
```

Standard deviations:

Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6
40.975497	29.264294	19.783897	15.954623	11.210822	7.008498

6 variables and 40 observations.

Функція prcomp() дає більш повну інформацію, а саме:

```
prcomp(pilots[,2:7])
```

Standard deviations (1, ..., p=6):

[1]	41.497499	29.637102	20.035932	16.157875	11.353640	7.097781
-----	-----------	-----------	-----------	-----------	-----------	----------

Rotation (n x k) = (6 x 6):

	PC1	PC2	PC3	PC4	PC5	PC6
V1	0.21165160	-0.38949336	0.88819049	-0.03082062	-0.04760343	-0.10677164
V2	-0.03883125	-0.06379320	0.09571590	0.19128493	-0.14793191	0.96269790
V3	0.08012946	0.06602004	0.08145863	0.12854488	0.97505667	0.12379748
V4	0.77552673	0.60795970	0.08071120	-0.08125631	-0.10891968	0.06295166
V5	-0.09593926	-0.01046493	0.01494473	-0.96813856	0.10919120	0.20309559
V6	0.58019734	-0.68566916	-0.43426141	-0.04518327	0.03644629	0.03572141

Зауважимо, що наведені РС відрізняються від власних векторів знаком! Те, що ще і стандартні відхилення різні, пов'язано з тим, що нормування матриці коваріації різне в обох методах: $\frac{1}{N}$ в prcomp $\frac{1}{N-1}$ в princomp¹¹.

Для того, щоб отримати повну інформацію і про стандартні відхилення і про стандартні відхилення, можна використати функцію summary():

```
pilots.pca <- prcomp(pilots[,2:7])
summary(pilots.pca)
```

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	41.4975	29.6371	20.0359	16.15788	11.35364	7.09778
Proportion of Variance	0.5003	0.2552	0.1166	0.07585	0.03745	0.01464
Cumulative Proportion	0.5003	0.7554	0.8721	0.94792	0.98536	1.00000

Запишемо тепер дані в координатах, коли осі – це перешкальовані власні вектори PC1 та PC2. В якості шкалюючого множника використаємо \sqrt{n} (корінь квадратний з кількості випробування).

Знайдемо $pc1 = (X - barX)'PC1/\sqrt{\lambda_1 n}$ та $pc2 = (X - barX)'PC2/\sqrt{\lambda_2 n}$:

```
scaling <- pilots.pca$sdev[1:2] * sqrt(nrow(pilots))
pc1 <- rowSums(t(t(sweep(pilots[,2:7], 2, colMeans(pilots[,2:7]))))
                * s.eigen$vectors[,1] * -1) / scaling[1])
pc2 <- rowSums(t(t(sweep(pilots[,2:7], 2, colMeans(pilots[,2:7]))))
                * s.eigen$vectors[,2]) / scaling[2])
```

¹¹<https://stats.stackexchange.com/questions/9500/why-do-the-r-functions-princomp-and-prcomp-give-different-results>

Тут використовується функція `sweep` для того, щоб центрувати компоненти `pilots[2:7]` (індекс 2 тут використовується для того, щоб зазначити, що ми проводимо цю операцію зі стовбчиками; якби було 1, це означало б, що операція проводиться із рядками).

Тепер ми задамо дані в цих нових координатах, і зобразимо графічно за допомогою `ggplot` (або більш новий `ggplot2`):

```
df <- data.frame(pc1, pc2, c(rep('Студент', 20), rep('Пілот', 20)))
colnames(df) <- c('PC1', 'PC2', 'Група')

ggplot(df, aes(x=PC1, y=PC2, color=Group)) +
  geom_point()
```

Результат зображенено на Рис. 7. Таке розділення на групи можна також зробити вбудованою функцією `autoplot` (при цьому нам знадобиться бібліотека `ggfortify`)

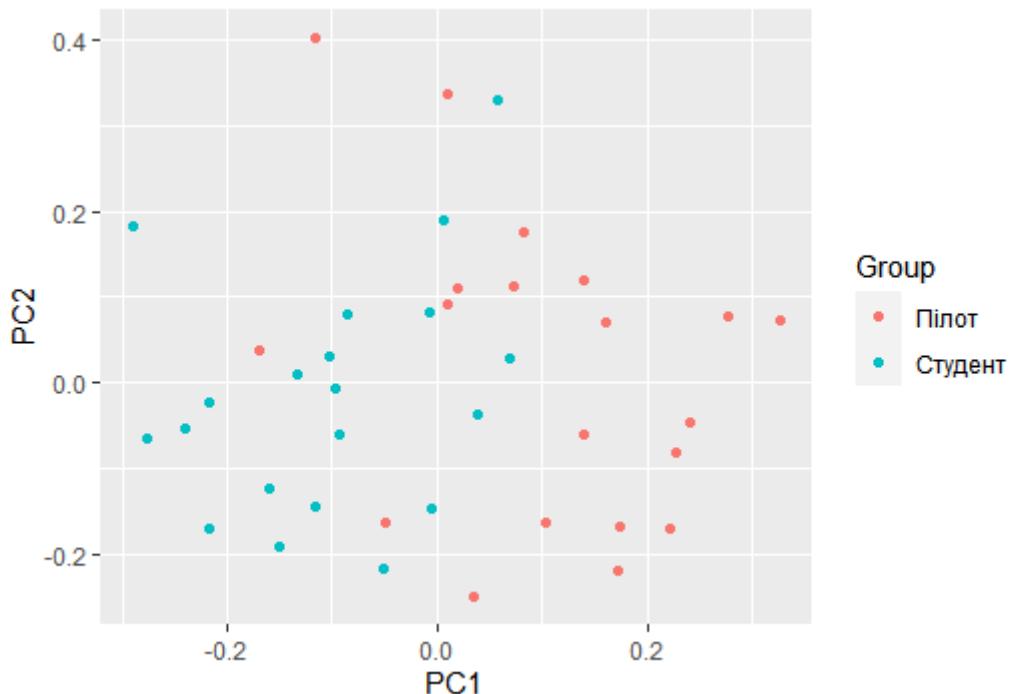


Рис. 7: Розділення на групи методом головних компонент

довоаною функцією `autoplot` (при цьому нам знадобиться бібліотека `ggfortify`)

```
library(ggfortify)
pca.plot <- autoplot(pilots.pca, data = pilots, colour = 'Group')
pca.plot
```

Альтернативно, такий самий аналіз можна зробити, використовуючи не коваріаційну матрицю, а кореляційну.

Розглянемо ще один приклад застосування методу головних компонент, а саме, стискання фотографії. Розглянемо наступний приклад¹².

Ми застосуємо метод головних компонент для того, щоб стиснути фотографію котика:

¹²<https://rpubs.com/aaronsc32/image-compression-principal-component-analysis>

Для цього завантажимо пакет jpeg та бібліотеку jpeg. Функція readJPEG перетворює

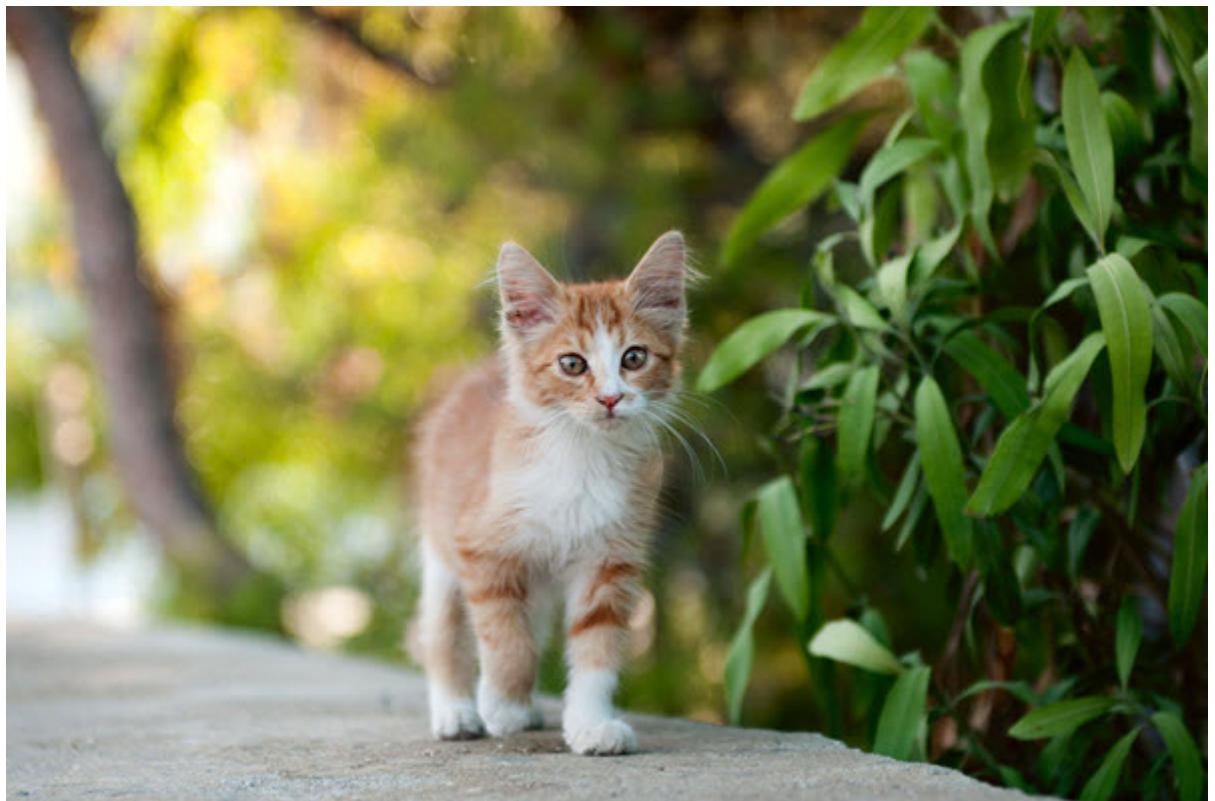


Рис. 8: Кошеня Патрик

фотографію в матрицю. ***

```
cat <- readJPEG('cat.jpg')
dim(cat)
ncol(cat)
nrow(cat)
```

Щоб вивести картинку на екран в jupyter, треба завантажити бібліотеку та застосуємо функцію (для цього треба покласти файл в ту ж папку, що і ірунб файл):

```
library(IRdisplay)
display_jpeg(file = 'cat.jpg')
```

Запустивши ці функції ми побачимо, що масив має розмірність $398 \times 600 \times 3$, тобто що фотографію можна зобразити за допомогою трьох матриць розміру (в пікселях) 398×600 , кожна матриця є представленням кольорів в схемі RGB. Наша задача – зменшити розмірності, вибравши ”ключові кольори”. А саме, ми у нас 398 спостережень, кожне розмірності 600. За допомогою методу головних компонент ми будемо зменшувати розмірність.

Спочатку виділимо кольори в окремі матриці, і виконаємо аналіз методом головних компонент для кожної з матриць.

```

r <- cat[, , 1]
g <- cat[, , 2]
b <- cat[, , 3]
cat.r.pca <- prcomp(r, center = FALSE)
cat.g.pca <- prcomp(g, center = FALSE)
cat.b.pca <- prcomp(b, center = FALSE)
rgb.pca <- list(cat.r.pca, cat.g.pca, cat.b.pca)

```

Ми обрали `center = FALSE` для того, щоб зберегти зображення кольорів, сентрування змістило б кольоровий спектр. Далі ми зібрали головні компоненти в один список.

Тепер ми будемо обирати, відповідно, 3, 10, 20, 50, 100, 300 головних компонент, та перетворимо відповідні матриці на фотографії (функція `writeJPEG`).

```

for (i in c(3,10,20,50,100,300)) {
  pca.img <- sapply(rgb.pca, function(j) {
    compressed.img <- j$x[,1:i] %*% t(j$rotation[,1:i])
  }, simplify = 'array')
  writeJPEG(pca.img, paste('c:/your_path/cat_compressed_',
  round(i,0), '_components.jpg', sep = ''))
}

```

Як ми бачимо, зі збільшенням кількості головних компонент ми отримаємо більш якісну картинку, див. Рисунок 9.

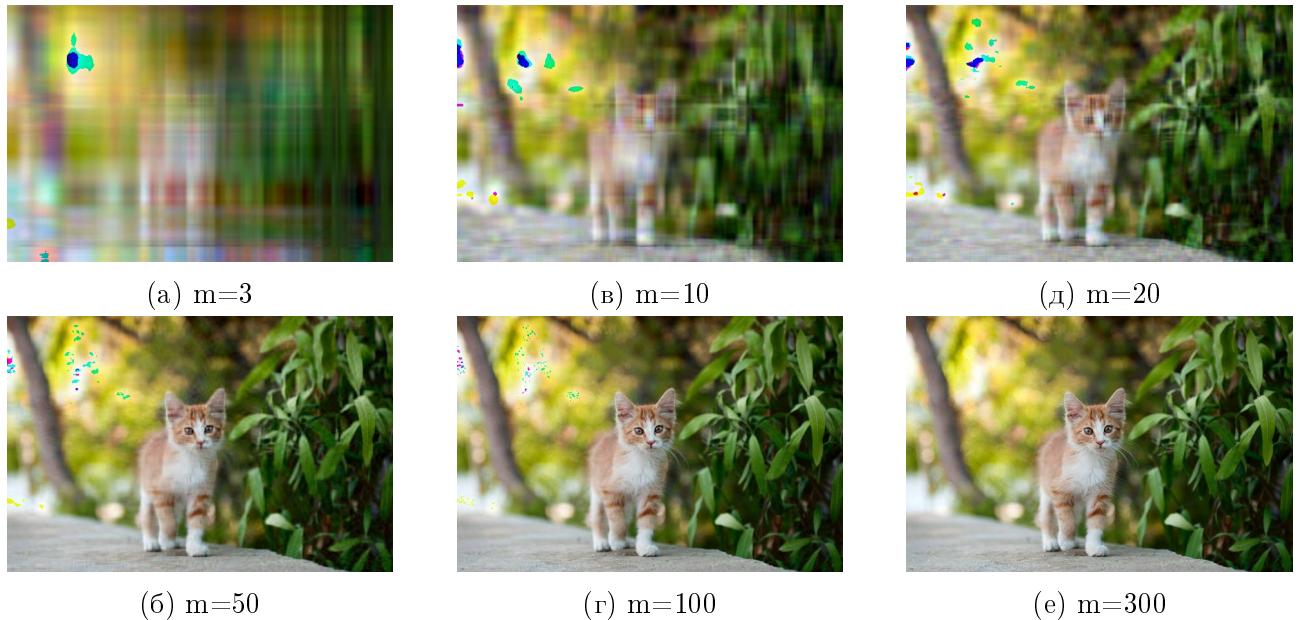


Рис. 9: Фото Кошеня Патріка після обробки

Заваження 10.1. Зменшити розмірність можна також за допомогою *кластерізації*, див. Розділ 12.1. А саме, можна виділити основні кольори і розфарбувати картину по принципу "точки одного кластеру мають спільний колір".

11 Задачі класифікації

Література: [R02, Гл.9]. Пакети **e1071**, **caret**, **caTools**.

Задача класифікації може виникнути, наприклад, коли треба класифікувати пацієнтів, чи мають вони певну хворобу чи ні, на основі тестів.

Нехай є вибірка спостережень, про яку попередньо відомо те, що ця вибірка складається з k груп. Наша задача – класифікувати кожне спостереження, тобто віднести його до однієї з груп.

Один із методів – це зробити класифікацію, беручи до уваги вже відому інформацію про групи. А саме, на основі результатів попередньої класифікації на тренувальній групі. Наприклад, можна порівнювати спостереження з векторами середніх, отриманих на попередньому кроці, та обрати групу відповідно тому, до якого середнього близче наше спостереження.

Розглянемо деякі методи класифікації.

11.1 Метод Фішера (лінійний класифікатор), 2 групи

Нехай є дві популяції, які мають однакові невироджені коваріаційні матриці $\Sigma_1 = \Sigma_2$ (тут можна не припускати, що популяції нормально розподілені). Побудуємо дискримінантну функцію

$$z = \mathbf{a}'\mathbf{y} = (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'S_{pl}^{-1}\mathbf{y}.$$

Ми будемо говорити, що спостереження \mathbf{y} належить до G_1 , якщо z знаходиться близче до $\bar{z}_1 = \mathbf{a}'\bar{\mathbf{y}}_1$ ніж до $\bar{z}_2 = \mathbf{a}'\bar{\mathbf{y}}_2$. В іншому випадку відносимо z до G_2 .

Нехай $\bar{z}_1 > \bar{z}_2$, та

$$z > \frac{\bar{z}_1 + \bar{z}_2}{2}. \quad (11.1)$$

З того, як обране \mathbf{a} , маємо

$$\bar{z}_1 - \bar{z}_2 = \mathbf{a}'(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) = (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'S_{pl}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) > 0.$$

Оскільки $\frac{\bar{z}_1 + \bar{z}_2}{2}$ є середньою точкою, то якщо виконано (11.1), то z знаходиться близче до $\bar{z}_1 = \mathbf{a}'\bar{\mathbf{y}}_1$ ніж до $\bar{z}_2 = \mathbf{a}'\bar{\mathbf{y}}_2$. Отже, ми класифікуємо z до групи G_1 .

Це правило класифікації, яке має назву метод Фішера, є лінійним. Більш того, якщо розподіл виборок є нормальним, то таке розбиття є асимптотично оптимальним, тобто похибка невірної класифікації буде мінімальною (див. [R02, с.302]).

11.2 Байесів класифікатор

Розглянемо спочатку випадок двох груп. Нехай

$$p_i = \mathbb{P}(\text{спостереження належить } G_i), \quad i = 1, 2, \quad p_1 + p_2 = 1,$$

є априорними ймовірностями того, що дане спостереження належить до групи G_i . Припустимо також, що відомі щільності розподілів $f(\cdot|G_i)$, $i = 1, 2$, за умови того, що спостереження знаходиться в групі G_i . Тоді відносимо спостереження \mathbf{y} до G_1 , якщо

$$p_1 f(\mathbf{y}|G_1) > p_2 f(\mathbf{y}|G_2), \quad (11.2)$$

інакше відносимо \mathbf{y} до G_2 .

При $p_1 = p_2$, то $y \in G_1$, якщо $\frac{f(\mathbf{y}|G_1)}{f(\mathbf{y}|G_2)} > 1$, тобто відношення вірогідностей більше за 1.

Запишемо (11.2), $i = 1, 2$, $p_1 + p_2 = 1$, у випадку, коли у нас є нормально розподілена вибірка.

Приклад 11.1. Нехай $f(\cdot|G_i)$ є щільностями нормальногорозподілу $N_p(\boldsymbol{\mu}_i, \Sigma)$, $i = 1, 2$. Тоді (11.2) має місце тоді і тільки тоді, коли

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1} \mathbf{y} > \frac{1}{2} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) + \ln \frac{p_2}{p_1}. \quad (11.3)$$

Дійсно, використовуючи симетричність матриці Σ , маємо $\mathbf{a}' \Sigma^{-1} \mathbf{b} = \mathbf{b}' \Sigma^{-1} \mathbf{a}$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, а отже,

$$\frac{p_2}{p_1} < \frac{f(\mathbf{y}|G_1)}{f(\mathbf{y}|G_2)} = \exp \left\{ (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1} \mathbf{y} - \frac{1}{2} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \right\}.$$

Взявши логарифм і перегрупувавши, і отримаємо (11.3).

На практиці, в нерівності (11.3) ми не знаємо ані середні $\boldsymbol{\mu}_i$, $i = 1, 2$, ані коваріаційну матрицю Σ . Тому для класифікації нам треба спочатку оцінити ці середні і коваріаційну матрицю через $\bar{\mathbf{y}}_1$, $\bar{\mathbf{y}}_2$ і S_{pl} , відповідно. Отже, нам потрібно замінити нерівність (11.3) на

$$(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)' S_{pl}^{-1} \mathbf{y} > \frac{1}{2} (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)' S_{pl}^{-1} (\bar{\mathbf{y}}_1 + \bar{\mathbf{y}}_2) + \ln \frac{p_2}{p_1}. \quad (11.4)$$

Таким чином, якщо для спостереження виконується (11.4), то відносимо спостереження \mathbf{y} до групи G_1 , якщо не виконується – то до групи G_2 .

За теоремою Байєса ми можемо переоцінити априорні ймовірності p_i та визначити

$$\begin{aligned} \tilde{p}_i := \mathbb{P}(G_i|\mathbf{y}) &= \frac{\mathbb{P}(G_i)f(\mathbf{y}|G_i)}{\mathbb{P}(G_1)f(\mathbf{y}|G_1) + \mathbb{P}(G_2)f(\mathbf{y}|G_2)} \\ &= \frac{p_i f(\mathbf{y}|G_2)}{p_1 f(\mathbf{y}|G_1) + p_2 f(\mathbf{y}|G_2)}, \quad i = 1, 2. \end{aligned} \quad (11.5)$$

У випадку двох груп така переоцінка нічого не змінює: $\tilde{p}_i > \tilde{p}_2$ тоді і тільки тоді, коли має місце (11.2).

11.3 Випадок декількох груп, спільна коваріаційна матриця

Нехай у нас є k груп нормально розподілених спостережень та відомі априорні ймовірності

$$p_i = \mathbb{P}(\text{спостереження належить } G_i), \quad i = 1, \dots, k.$$

Ми будемо відносити спостереження \mathbf{y} до групи G_i , якщо

$$p_i f(\mathbf{y}|G_i) > p_j f(\mathbf{y}|G_j), \quad j = 1, \dots, k, \quad j \neq i.$$

Розглянемо

$$\begin{aligned} \ln(p_i f(\mathbf{y}|G_i)) &= \ln p_i - \ln ((2\pi)^{p/2} |\Sigma|^{1/2}) - \frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}_i)' \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu}_i) \\ &= \ln p_i - \ln ((2\pi)^{p/2} |\Sigma|^{1/2}) + \boldsymbol{\mu}_i' \Sigma^{-1} \mathbf{y} - \frac{1}{2} \boldsymbol{\mu}_i' \Sigma^{-1} \boldsymbol{\mu}_i - \frac{1}{2} \mathbf{y}' \Sigma^{-1} \mathbf{y}. \end{aligned} \quad (11.6)$$

Зауважимо, що $\mathbf{y}'\Sigma^{-1}\mathbf{y}$ та $|\Sigma|$ не залежать від номеру групи.

Розглянемо функцію (замінимо теоретичні середні $\boldsymbol{\mu}_i$ на вибіркові $\bar{\mathbf{y}}_i$)

$$L_i(\mathbf{y}) := \ln p_i + \bar{\mathbf{y}}_i' S_{pl}^{-1} \mathbf{y} - \frac{1}{2} \bar{\mathbf{y}}_i' S_{pl}^{-1} \bar{\mathbf{y}}_i.$$

Тут $L_i(\mathbf{y})$ є лінійною функцією від \mathbf{y} ; нагадаємо, що у випадку k груп різного обсягу n_i зважена коваріація визначається наступним чином:

$$S_{pl} = \frac{1}{N-k} \sum_{i=1}^k (n_i - 1) S_i = \frac{E}{N-k}, \quad N = \sum_{i=1}^k n_i.$$

Можна зробити квадратичну класифікацію. Для цього розглянемо перший рядок (11.6), в якому теоретичні середні і коваріація замінені на $\bar{\mathbf{y}}_i$, $i = 1, 2$, та S_{pl} . Але вираз

$$\ln p_i - \ln ((2\pi)^{p/2} |\Sigma|^{1/2}) - \frac{1}{2} (\mathbf{y} - \bar{\mathbf{y}}_i)' S_{pl}^{-1} (\mathbf{y} - \bar{\mathbf{y}}_i)$$

досягає максимума тоді і тільки тоді, коли $L_i(\mathbf{y})$, оскільки частина $\frac{1}{2} \mathbf{y}' \Sigma^{-1} \mathbf{y}$ є спільною для всіх k груп спостережень.

Розглянемо квадрат відстані Махalanобіса \mathbf{y} від $\bar{\mathbf{y}}_i$:

$$D_i^2(\mathbf{y}) = (\mathbf{y} - \bar{\mathbf{y}}_i)' S_{pl}^{-1} (\mathbf{y} - \bar{\mathbf{y}}_i).$$

Тоді

$$-\ln p_i + D_i^2(\mathbf{y}) \quad \text{мінімальне} \iff L_i(\mathbf{y}) \quad \text{максимальне.}$$

Отже, ми відносимо спостереження \mathbf{y} до групи G_i , якщо функція $L_i(\mathbf{y})$ є максимальною для i серед всіх k функцій.

Розглянемо апостеріорні ймовірності

$$\tilde{p}_i = \mathbb{P}(G_i | \mathbf{y}) = \frac{p_i f(\mathbf{y} | G_i)}{\sum_{j=1}^k p_j f(\mathbf{y} | G_j)}.$$

Ми відносимо \mathbf{y} до групи G_i , якщо ймовірність \tilde{p}_i максимальна; зауважимо, що це відбувається тоді і тільки тоді, коли максимальною є ймовірність $p_i f(\mathbf{y} | G_i)$.

Приклад 11.2. Нехай $f(\mathbf{y} | G_i)$ є щільностями нормального розподілу $N_p(\boldsymbol{\mu}_i, \Sigma)$, $i = 1, 2$, але вектор середніх $\boldsymbol{\mu}_i$ і коваріація Σ – невідомі. Тоді розглянемо апостеріорні ймовірності, в яких теоретичні значення замінені оцінками:

$$\tilde{\mathbb{P}}(G_i | \mathbf{y}) := \frac{p_i e^{-D_i^2/2}}{\sum_{j=1}^k p_j e^{-D_j^2/2}}, \quad i = 1, \dots, k.$$

Відповідно, робимо класифікацію на основі ймовірностей $\tilde{\mathbb{P}}(G_i | \mathbf{y})$, $i = 1, \dots, k$.

11.4 Похибка класифікації

Розглянемо похибку класифікації у випадку двох груп (див. [R98, с.240]):

$$\begin{aligned}\mathcal{E} &= \mathbb{P}\{\text{випробування } \mathbf{y} \text{ класифіковано до хибної групи }\} \\ &= p_1 \mathbb{P}(\mathbf{y} \text{ класифіковано до } G_2 | G_1) + p_2 \mathbb{P}(\mathbf{y} \text{ класифіковано до } G_1 | G_2),\end{aligned}$$

де p_1, p_2 – априорні ймовірності. Фактично, наведена вище формула – це формула повної ймовірності. Знайдемо $\mathbb{P}(\text{класифіковано до } G_i | G_j)$, $i, j = 1, 2$, $i \neq j$. Якщо ми класифікували \mathbf{y} до G_1 , то виконується (11.3). Позначимо

$$\boldsymbol{\alpha} := \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \quad \Delta^2 := (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2).$$

Тоді

$$\begin{aligned}\mathbb{P}(\text{класифіковано до } G_1 | G_2) &= \mathbb{P}_{G_2} \left(\boldsymbol{\alpha}' \mathbf{y} > \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) + \ln \frac{p_2}{p_1} \right) \\ &= \mathbb{P}_{G_2} \left(\frac{\boldsymbol{\alpha}' \mathbf{y} - \boldsymbol{\alpha}' \boldsymbol{\mu}_2}{\Delta} > \frac{\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) + \ln \frac{p_2}{p_1} - \boldsymbol{\alpha}' \boldsymbol{\mu}_2}{\Delta} \right) \\ &= \mathbb{P}_{G_2} \left(\frac{\boldsymbol{\alpha}' (\mathbf{y} - \boldsymbol{\mu}_2)}{\Delta} > \frac{\frac{1}{2} \Delta^2 + \ln \frac{p_2}{p_1}}{\Delta} \right).\end{aligned}$$

Покажемо, що $\frac{\boldsymbol{\alpha}' (\mathbf{y} - \boldsymbol{\mu}_2)}{\Delta} \sim N_1(0, 1)$. Оскільки $\mathbf{y} \sim N(\boldsymbol{\mu}_2, \Sigma)$, то для $\mathbf{b} = \Sigma^{-1/2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ отримаємо $\|\mathbf{b}\|^2 = \Delta^2$ та, відповідно,

$$\frac{\boldsymbol{\alpha}' (\mathbf{y} - \boldsymbol{\mu}_2)}{\Delta} = \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu}_2)}{\Delta} = \frac{\mathbf{b}' \Sigma^{-1/2} (\mathbf{y} - \boldsymbol{\mu}_2)}{\|\mathbf{b}\|}.$$

Використаємо тепер те, що якщо $\mathbf{z} \sim N_p(0, B)$, то $A\mathbf{z} \sim N_p(0, ABA')$. Тоді

$$\frac{\mathbf{b}' \Sigma^{-1/2} (\mathbf{y} - \boldsymbol{\mu}_2)}{\|\mathbf{b}\|} \sim N \left(0, \frac{\mathbf{b}' \Sigma^{-1/2} \Sigma \Sigma^{-1/2} \mathbf{b}}{\|\mathbf{b}\|^2} \right) = N(0, 1).$$

Отже,

$$\begin{aligned}\mathbb{P}(\text{класифіковано до } G_1 | G_2) &= \mathbb{P} \left(w > \frac{\frac{1}{2} \Delta^2 + \ln \frac{p_2}{p_1}}{\Delta} \right) \\ &= \mathbb{P} \left(w < -\frac{\frac{1}{2} \Delta^2 + \ln \frac{p_2}{p_1}}{\Delta} \right) \\ &= \Phi \left(-\frac{\frac{1}{2} \Delta^2 + \ln \frac{p_2}{p_1}}{\Delta} \right),\end{aligned}$$

де Φ – функція розподілу $w \sim N(0, 1)$. Аналогічно,

$$\mathbb{P}(\text{класифіковано до } G_2 | G_1) = \Phi \left(-\frac{\frac{1}{2} \Delta^2 + \ln \frac{p_1}{p_2}}{\Delta} \right).$$

Отже,

$$\mathcal{E} = p_1 \Phi \left(-\frac{\frac{1}{2}\Delta^2 + \ln \frac{p_2}{p_1}}{\Delta} \right) + p_2 \Phi \left(-\frac{\frac{1}{2}\Delta^2 + \ln \frac{p_1}{p_2}}{\Delta} \right).$$

Підставивши замість Δ^2 оцінку $D^2 = (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'S_{pl}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)$, отримаємо оцінку похибки класифікації

$$\hat{\mathcal{E}} = p_1 \Phi \left(-\frac{\frac{1}{2}D^2 + \ln \frac{p_1}{p_2}}{D} \right) + p_2 \Phi \left(-\frac{\frac{1}{2}D^2 + \ln \frac{p_2}{p_1}}{D} \right).$$

Якщо спостереження не є нормальними, можна застосувати ядерні оцінки щільності для оцінювання $f(\mathbf{y}|G)$, див. Додаток 1.

11.5 Метод найближчих сусідів

Одним з простих найбільш поширених методів є метод k найближчих сусідів (KNN). Розглянемо квадрат відстані між y_i та y_j :

$$d^2(\mathbf{y}_i, \mathbf{y}_j) = (\mathbf{y}_i - \mathbf{y}_j)'S_{pl}^{-1}(\mathbf{y}_i - \mathbf{y}_j).$$

Для того, щоб віднести спостереження до тієї чи іншої групи, дивимося на k найближчих сусідів цього спостереження. Якщо більшість з них належить до G_1 , то відносимо \mathbf{y} до G_1 , і навпаки. Якщо вибірки різного розміру, то відносимо \mathbf{y}_i до G_1 , якщо $\frac{k_1}{n_1} > \frac{k_2}{n_2}$, де k_1 та k_2 – кількості сусідів в групах з n_1 та n_2 елементами, відповідно. Якщо відомі апріорні ймовірності p_1, p_2 , то ми відносимо \mathbf{y}_i до G_1 , якщо

$$p_1 \frac{k_1}{n_1} > p_2 \frac{k_2}{n_2}.$$

Як обирати оптимальні k_i ? Пропонується обирати (див. [R02, §9.7.3]) $k_i \sim \sqrt{n_i}$, але на практиці краще спробувати кілька значень і обрати те, яке дає кращу похибку класифікації.

11.6 Приклади

Розглянемо спочатку класифікацію за допомоги дискримінантної функції¹³. Нам потрібні наступні пакети: **MASS**, **e1071**, **caret**, **caTools**.

1. Завантажимо таблицю, в якій наведені вимірювання 2-х типів жуків (для простоти позначимо вимірювання через колонки V3–V6).

```
beetles <- read.table('T5_5_FBEETLES.DAT',
  col.names = c('Measurement.Number', 'Species', 'V3', 'V4', 'V5', 'V6'))
head(beetles)
```

	Measurement.Number	Species	V3	V4	V5	V6	
1			1	189	245	137	163
2			1	192	260	132	217
3			1	217	276	141	192

¹³<https://rpubs.com/aaronsc32/classification-linear-discriminant-analysis>

```

4           1   221 299 142 213
5           1   171 239 128 158
6           1   192 262 147 173

```

Розділимо цю таблиці на 2 за принципом: якщо Species = 1, то це перша група, якщо Species = 2 – то друга.

```

beetle1 <- beetles[beetles$Species == 1,] [,3:6]
beetle2 <- beetles[beetles$Species == 2,] [,3:6]

```

Знайдемо середні значення та зважену коваріацію (нагадаємо, що 2 в функції apply означає, що ми сумуємо по стовбчикам).

```

n1 <- nrow(beetle1)
n2 <- nrow(beetle2)

beetle1.means <- apply(beetle1, 2, mean)
beetle2.means <- apply(beetle2, 2, mean)

w1 <- (n1 - 1) * var(beetle1)
w2 <- (n2 - 1) * var(beetle2)

sp1 <- (w1 + w2)/(n1 + n2 - 2)

```

Позначимо через cutoff середнє значення, за допомогою якого ми будемо визначати, до якої групи належить випробування (див. також розділ 9).

```

cutoff <- .5 * (beetle1.means - beetle2.means) %*% solve(sp1)
            %*% (beetle1.means + beetle2.means)

```

У нашому випадку $a'\mu_1 > a'\mu_2$. Отже, $z > \text{cutoff} = -15.81$, то ми класифікуємо спостереження до першої групи, якщо ні – то до другої.

```

species.prediction <- apply(beetles[,3:6], 1, function(y) {
  z <- (beetle1.means - beetle2.means) %*% solve(sp1) %*% y
  ifelse(z > cutoff, 1, 2)
})

```

Розглянемо матрицю змішування (confusion matrix), яка показує, наскільки якісно ми віднесли елементи до тієї чи іншої групи.

```

table(beetles$Species, species.prediction,
      dnn = c('Actual Group', 'Predicted Group'))

```

		Predicted Group	
Actual Group		1	2
1	1	19	0
	2	1	19

Ми класифікували вірно всі спостереження, які відносяться до першої групи, але помилково класифікували одне спостереження з другої групи. Похибка класифікації – це кількість хибних класифікацій поділити на кількість елементів у вибірці. Кількість спостережень у вибірці:

```
n <- dim(beetles)[1]
```

Тобто $1/n = 0.02564103$. Як ми бачимо, навіть при малому об'ємі виборки похибка є досить малою.

Для того, щоб зробити класифікацію за допомогою вбудованої функції, використаємо функцію lda() з пакету MASS.

```
beetle.lda <- lda(Species ~ .-Measurement.Number, data = beetles)
lda.pred <- predict(beetle.lda)$class
```

Матриця змішання демонструє співвідношення результату класифікації та фактичної належності до тієї чи іншої групи. Виведемо результати в таблицю:

```
table(beetles$Species, lda.pred,
      dnn = c('Actual Group', 'Predicted Group'))
```

		Predicted Group	
		1	2
Actual Group	1	19	0
	2	1	19

2. У наступному прикладі ми розглянемо байесівську класифікацію (корисний приклад, як розбивати вибірку на тестову та тренувальну вибірки можна подивитися тут). Для цього ми розглянемо дані по квіткам ірис та спробуємо побудувати класифікатор.

База даних квітів ірис містить три типи квіток: setosa, virginica, versicolor, та їх характеристики- довжина та ширина стебла, та довжина та ширина пелюстоків.

```
data(iris)
head(iris)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

Розіб'ємо дані на дві групи у співвідношенні 0.7:0.3, тобто 70% даних – це навчальний набір, та 30% – це тестувальний набір:

```

iris$spl=sample.split(iris,SplitRatio=0.7)
train=subset(iris, iris$spl==TRUE)
test=subset(iris, iris$spl==FALSE)
head(iris)

```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	spl
1	5.1	3.5	1.4	0.2	setosa	TRUE
2	4.9	3.0	1.4	0.2	setosa	FALSE
3	4.7	3.2	1.3	0.2	setosa	TRUE
4	4.6	3.1	1.5	0.2	setosa	TRUE
5	5.0	3.6	1.4	0.2	setosa	FALSE
6	5.4	3.9	1.7	0.4	setosa	TRUE

Альтернативно, це розбиття можна зробити наступним чином:

```

split <- sample.split(iris, SplitRatio = 0.7)
train_cl <- subset(iris, split == "TRUE")
test_cl <- subset(iris, split == "FALSE")

```

Застосуємо тепер байесівську класифікацію до навчального набору:

```

set.seed(120)
classifier_cl <- naiveBayes(Species ~ ., data = train_cl)
classifier_cl

```

Naive Bayes Classifier for Discrete Predictors

Call:

```
naiveBayes.default(x = X, y = Y, laplace = laplace)
```

A-priori probabilities:

Y

	setosa	versicolor	virginica
Y	0.33	0.34	0.33

Conditional probabilities:

Sepal.Length

Y	[,1]	[,2]
setosa	5.003030	0.3901000
versicolor	5.897059	0.5578559
virginica	6.660606	0.6123415

Sepal.Width

Y	[,1]	[,2]
setosa	3.418182	0.4216445
versicolor	2.705882	0.3365972

```

virginica 2.948485 0.3308334

Petal.Length
Y [,1] [,2]
setosa 1.487879 0.1745666
versicolor 4.226471 0.4937849
virginica 5.584848 0.5489315

Petal.Width
Y [,1] [,2]
setosa 0.2454545 0.09711755
versicolor 1.3205882 0.19814199
virginica 1.9939394 0.26450354

spl
Y FALSE TRUE
setosa 0.3636364 0.6363636
versicolor 0.4117647 0.5882353
virginica 0.4242424 0.5757576

```

Зауважимо, що елементами вищеприведеної таблиці є не умовні ймовірності, а середні значення (перша колонка) та стандартні відхилення (друга колонка).

При цьому ми припускали, що дані є нормальню розподіленими випадковими величинами.

Обчислимо матрицю невідповідностей:

```

y_pred1 <- predict(classifier_cl, newdata = test_cl)
cm1 <- table(test_cl$Species, y_pred1)
cm1

```

	setosa	versicolor	virginica
setosa	17	0	0
versicolor	0	14	2
virginica	0	2	15

Отже, майже всі спостереження класифіковані правильно.

Якщо розподіл не є наперед відомим, потрібно використовувати ядерні оцінки щільності ядра. Застосуємо функцію класифікації, використовуючи, наприклад, трикутне ядро.

```

classifier2_cl <- naiveBayes(Species ~ ., data = train_cl, usekernel=TRUE,
kernel="triangular" )
classifier2_cl

```

Naive Bayes Classifier for Discrete Predictors

```

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace, usekernel = TRUE,
kernel = "triangular")

A-priori probabilities:
Y
setosa versicolor virginica
0.33      0.34      0.33

Conditional probabilities:
Sepal.Length
Y      [,1]      [,2]
setosa 5.003030 0.3901000
versicolor 5.897059 0.5578559
virginica 6.660606 0.6123415

Sepal.Width
Y      [,1]      [,2]
setosa 3.418182 0.4216445
versicolor 2.705882 0.3365972
virginica 2.948485 0.3308334

Petal.Length
Y      [,1]      [,2]
setosa 1.487879 0.1745666
versicolor 4.226471 0.4937849
virginica 5.584848 0.5489315

Petal.Width
Y      [,1]      [,2]
setosa 0.2454545 0.09711755
versicolor 1.3205882 0.19814199
virginica 1.9939394 0.26450354

spl
Y      FALSE      TRUE
setosa 0.3636364 0.6363636
versicolor 0.4117647 0.5882353
virginica 0.4242424 0.5757576

```

Можна вивести інформацію, яка випробування відноситься до якого класу:

```

y_pred2 <- predict(classifier_cl, newdata = test_cl)
y_pred2

```

```

[1] setosa    setosa    setosa    setosa    setosa    setosa    setosa
[8] setosa    setosa    setosa    setosa    setosa    setosa    setosa

```

```
[15] setosa      setosa      setosa      virginica versicolor versicolor versicolor
[22] versicolor versicolor virginica versicolor versicolor versicolor versicolor
[29] versicolor versicolor versicolor versicolor virginica virginica virginica
[36] versicolor virginica virginica virginica virginica virginica virginica
[43] virginica  virginica  versicolor virginica virginica virginica virginica
[50] virginica

Levels: setosa versicolor virginica
```

Або можна вивести матрицю змішування:

```
cm2 <- table(test_cl$Species, y_pred2)
cm2
```

	setosa	versicolor	virginica
setosa	17	0	0
versicolor	0	14	2
virginica	0	2	15

```
confusionMatrix(cm)
```

Confusion Matrix and Statistics

	setosa	versicolor	virginica
setosa	17	0	0
versicolor	0	14	2
virginica	0	2	15

Overall Statistics

```
Accuracy : 0.92
95% CI : (0.8077, 0.9778)
No Information Rate : 0.34
P-Value [Acc > NIR] : < 2.2e-16
```

Kappa : 0.88

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.00	0.8750	0.8824
Specificity	1.00	0.9412	0.9394

Pos Pred Value	1.00	0.8750	0.8824
Neg Pred Value	1.00	0.9412	0.9394
Prevalence	0.34	0.3200	0.3400
Detection Rate	0.34	0.2800	0.3000
Detection Prevalence	0.34	0.3200	0.3400
Balanced Accuracy	1.00	0.9081	0.9109

12 Кластерний аналіз

Література: [R02, Гл.14], [Ev93], [ELLS11], [Br]. Пакети **dplyr**, **mvtnorm**, **mclust**, та **ggplot2** для графіки.

12.1 Типи кластерізації

В кластерному аналізі ми групуємо спостереження в групи таким чином, щоб групи розрізнялися між собою, але при цьому в середині групи об'єкти мали схожі властивості. Наша мета – знайти таке оптимальне групування.

Кластерний аналіз принципово відрізняється від класифікації даних, яку було розглянуто в главі 11. Задача класифікації – розподілити об'єкти по попередньо визначенім групам. В кластерному аналізі ані самі групи ані число груп не визначено напаред. Для того, щоб віднести об'єкт до певного кластеру, знаходять спільні риси об'єктів (наприклад, вимірюють "відстань" між ними). Інший підхід полягає в тому, щоб обрати центр кластеру та порівнювати відстані об'єктів до центру або до іншого кластеру. Можна також відносити об'єкти до того чи іншого кластеру, порівнюючи кореляції.

Розглянемо наступні типи кластерізації. В *ієархічній кластерізації* починаємо з n об'єктів і поступово об'єднуємо їх в кластери; на виході ми отримаємо один кластер. Можна також робити кластерізацію у зворотньому боці – почати з одного кластеру і розбивати на частини.

В *кластерізації розбиттям* ми просто розбиваємо вибірку на t кластерів. Це можна зробити обравши центри і вимірюючи відстані до центрів. Інші (статистичні) методи використовують матриці H і E з MANOVA.

При вимірюванні відстані до центру використовуються наступні методи.

- **Метод найближчого сусіда (nearest neighbour або single linkage method).** Відстань між кластерами задається наступним чином:

$$D(A, B) = \min\{d(\mathbf{y}_i, \mathbf{y}_j) : \mathbf{y}_i \in A, \mathbf{y}_j \in B\},$$

де $d(\cdot, \cdot)$ є евклідовою (або іншою відстанню) між точками. Два кластери, відстань між якими мінімальна, об'єднують в один. В результаті отримуємо "дерево", або *дендрограму*.

- **Метод найшвидшого сусіда (fastest neighbour або complete linkage method).**

Нехай

$$D(A, B) = \max\{d(\mathbf{y}_i, \mathbf{y}_j) : \mathbf{y}_i \in A, \mathbf{y}_j \in B\},$$

Два кластери, у яких $D(A, B)$ мінімальне, об'єднують в один.

- **Метод середньої відстані (average linkage method).**

Середня відстань між n_A точками A та n_B точками B рахується наступним чином:

$$D(A, B) = \frac{1}{n_A n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} d(\mathbf{y}_i, \mathbf{y}_j).$$

Так само: два кластери, у яких відстань $D(A, B)$ мінімальна, об'єднують в один.

- **Метод центроїда (centroid method).**

Відстань між двома кластерами визначається через евклідову відстань між середніми цих кластерів:

$$D(A, B) = d(\bar{\mathbf{y}}_A, \bar{\mathbf{y}}_B), \quad \bar{\mathbf{y}}_A := \frac{1}{n_A} \sum_{i=1}^{n_A} \mathbf{y}_{iA}, \quad \bar{\mathbf{y}}_B := \frac{1}{n_B} \sum_{i=1}^{n_B} \mathbf{y}_{iB}.$$

Об'єднуємо в один два кластери, у яких відстань $D(A, B)$ мінімальна, та обираємо новий центр за формулою

$$\bar{\mathbf{y}}_{AB} = \frac{n_A \bar{\mathbf{y}}_A + n_B \bar{\mathbf{y}}_B}{n_A + n_B}.$$

- **Метод медіан (median method).**

Якщо $n_A \gg n_B$, то новий центроїд $\bar{\mathbf{y}}_{AB}$ значно більший за величину ніж $\bar{\mathbf{y}}_A$ але менший за величину ніж $\bar{\mathbf{y}}_B$. Для того, щоб не нормувати середні відповідно до ваги, можна використати медіану

$$m_{AB} = \frac{1}{2}(\bar{\mathbf{y}}_A + \bar{\mathbf{y}}_B)$$

для обчислення відстанні між кластерами.

Два кластери з найменшою відстанню об'єднуємо в один. Зауважимо, що це не "звичайна" медіана в статистичному сенсі, це медіана в геометричному сенсі.

- **Метод Варда (Ward's method).**

Метод Варда (або the incremental sum of squares method), використовує квадрати відстаней всередині кластерів та квадрати відстаней між кластерами. Якщо АВ є кластером, який отримано шляхом комбінування кластерів А і В, то суми всередині кластерів дорівнюють

$$\begin{aligned} SSE_A &:= \sum_{i=1}^{n_A} (\mathbf{y}_i - \bar{\mathbf{y}}_A)' (\mathbf{y}_i - \bar{\mathbf{y}}_A), \\ SSE_B &:= \sum_{i=1}^{n_B} (\mathbf{y}_i - \bar{\mathbf{y}}_B)' (\mathbf{y}_i - \bar{\mathbf{y}}_B), \\ SSE_{AB} &:= \sum_{i=1}^{n_{AB}} (\mathbf{y}_i - \bar{\mathbf{y}}_{AB})' (\mathbf{y}_i - \bar{\mathbf{y}}_{AB}). \end{aligned}$$

Метод Варда полягає в тому, щоб з'єднувати два кластери, у яких відстань

$$I_{AB} = SSE_{AB} - (SSE_A + SSE_B) = \frac{n_A n_B}{n_A + n_B} (\bar{\mathbf{y}}_A - \bar{\mathbf{y}}_B)' (\bar{\mathbf{y}}_A - \bar{\mathbf{y}}_B).$$

є мінімальною.

Порівняємо метод Варда і метод центроїда. Якщо піднести відстань у методі центроїда до квадрату, то побачимо, що єдина відмінність – це коефіцієнт $\frac{n_A n_B}{n_A + n_B}$. Отже, розмір кластера має вплив на метод Варда, але не на метод центроїда. Оскільки

$$\frac{n_A n_B}{n_A + n_B} = \frac{1}{1/n_A + 1/n_B},$$

то $\frac{n_A n_B}{n_A + n_B}$ зростає при зростанні n_A та n_B . З іншого боку,

$$\frac{n_A n_B}{n_A + n_B} = \frac{n_A}{1 + n_A/n_B},$$

а отже, якщо n_B зростає при фіксованому n_A , то $\frac{n_A n_B}{n_A + n_B}$ зростає. Отже, порівняно з методом центроїда, метод Варда скоріше з'єднує малі кластери ніж велики.

Розглянемо тепер неієархічні методи: метод розбиття, методи з використанням MANOVA, та методи, що базуються на оцінюванні щільності розподілу.

У кластерізації методом розбиття спостереження розбивають на t груп без ієархічного групування. Ідеально було б розглянути всі можливі розбиття, але така процедура дуже громіздка, тому розглядають інші підходи.

• Метод k середніх (k-means method)

Цей метод дозволяє пересувати елементи з одного кластеру в інший, що не є можливим в ієархічних методах кластерізації. Спочатку ми обираємо t елементів центроїдами кластерів. Ці t елементів можна обрати випадково, наприклад, обрати перші елементів у вибірці, або точки, які мають максимальні відстані, або точки, в яких щільність (розподілу) є максимальна і т.д. При цьому число кластерів t обирається наперед. Після цього інші точки відносять до кластерів за принципом мінімальної відстані. Як тільки кластер має більш ніж один елемент, центр кластеру обчилюється за методом центроїда. Як тільки всі елементи розподілені між кластерами, кожен елемент кластеру повторно розглядається, чи знаходиться він близче до іншого центроїда чи ні. Якщо так, то цей елемент відноситься до іншого кластеру. Цю процедуру повторюють до тих пір, поки покращення вже неможливі. Також, метод k середніх можна комбінувати з ієархічними методами.

• MANOVA методи

Розглянемо методи, засновані на аналізі матриць H та E для однофакторної моделі MANOVA. Якщо кластери "добре визначені", то матриця похибок має бути "малою", а матриця гіпотези – "великою" (тобто середні суттєво відрізняються).

Це можна зробити наступними методами.

1. Мінімізувати

$$\text{trace}(E) = \text{trace} \sum_{i=1}^m \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})';$$

2. Мінімізувати $|E|$, що є еквівалентом мінімізації Л Уілкса;

3. Максимізувати $\text{trace}(E - H)$, або максимізувати

$$\text{trace } E^{-1} H = \sum_{i=1}^s \lambda_i.$$

Подивимось, як працює перший метод. Після алгебраїчних перетворень отримаємо:

$$\begin{aligned}\text{trace } E &= \text{trace} \left[\sum_{i=1}^m \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})' \right] \\ &= \sum_{i=1}^m \text{trace} \left[\sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})' \right] \\ &=: \sum_{i=1}^m \text{trace } E_i.\end{aligned}$$

Зауважимо, що (довести!)

$$\text{trace } E_i = \sum_{j=1}^n \text{trace}(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})' = (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot})'(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\cdot}).$$

Отже, мінімізація $\text{trace } E$ – це фактично мінімізація відстаней від середніх значень всередині групи. Відповідно, ми розглядаємо можливі комбінації \mathbf{y} по групам, та обираємо таке розбиття, на якому $\text{trace } E$ – мінімальне. При цьому ми фактично намагаємося будувати “сферичні кластери”, оскільки мінімізуючи слід, ми мінімізуємо діагональні елементи, тобто дисперсії.

Другий метод, а саме, мінімізація E – це те саме, що мінімізація $\Lambda = \frac{|E|}{|E+H|}$. При цій мінімізації ми враховуємо діагональні елементи (тобто кореляцію), і наші ”оптимальні” кластери будуть мати приблизно однакову ”еліптичну форму” (тому що в обчисленнях присутня зважена коваріаційна матриця ($S_{pl} = E/\nu_E$)).

Третій підхід базується на максимізації $\text{trace } E^{-1}H$. При цьому найбільше власне значення λ_1 (яке відповідає дискримінантній функції $z_1 = \mathbf{a}'_1 \mathbf{y}$, де \mathbf{a}_1 є відповідним власним вектором) має найбільший вплив на $\text{trace } E^{-1}H$. У цьому випадку отримані кластери також будуть мати еліптичну форму та будуть витягнутими вздовж вектору \mathbf{a}_1 , якщо перше власне число суттєво більше за інші. Але така кластеризація, якщо використовується ітеративно, може не давати адекватного результату, так як може змінюватися орієнтація найбільшого власного вектору.

• Метод сумішей

У цьому методі ми припускаємо існування t (багатовимірних нормальних) розподілів та маємо на меті розподілити кожне із спостережень по цим t класам (див. також розділ 11.2). Визначимо щільність суміші наступним чином:

$$h(\mathbf{y}, \Theta) = \sum_{i=1}^m \alpha_i f(\mathbf{y}, \boldsymbol{\mu}_i, \Sigma_i),$$

де $0 \leq \alpha_i \leq 1$, $\sum_{i=1}^m \alpha_i = 1$, а $f(\mathbf{y}, \boldsymbol{\mu}_i, \Sigma_i)$ є щільністю нормальногорозподілу $N_p(\boldsymbol{\mu}_i, \Sigma_i)$. Параметр $\Theta = (\alpha_i, \boldsymbol{\mu}_i, \Sigma_i)_{i=1}^m$ можна отримати з попереднього аналізу (наприклад, застосувавши спочатку метод k середніх).

Далі формуємо кластери наступним чином. Спостереження \mathbf{y} відносимо до кластеру G_i , якщо апостеріорна вірогідність

$$\frac{\hat{\alpha}_j f(\mathbf{y}, \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}{h(\mathbf{y})},$$

найбільша для $j = i$. Тут $\hat{\alpha}_j$, $\hat{\boldsymbol{\mu}}_j$, $\hat{\Sigma}_j$ є оцінками максимальної вірогідності. Як знайти ці оцінки? Запишемо задачу максимізації:

$$L(Y, \Theta) = \prod_{j=1}^n h(\mathbf{y}_j, \Theta) \implies \max,$$

$$\sum_{i=1}^m \alpha_i = 1,$$

де $Y = (\mathbf{y}_1, \dots, \mathbf{y}_n)$. Розглянемо відповідно задачу Лагранжа:

$$\ell(Y, \Theta) := \ln L(Y, \Theta) - \lambda \left(\sum_{i=1}^m \alpha_i - 1 \right) \implies \max. \quad (12.1)$$

За формулою Байєса можна знайти апостеріорні ймовірності

$$\mathbb{P}(G_i | \mathbf{y}_j) := \frac{\alpha_i f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i)}{h(\mathbf{y}_j, \Theta)}. \quad (12.2)$$

Ми запишемо рівняння, яким мають задовольняти α_i , $\boldsymbol{\mu}_i$ та Σ_i , $1 \leq i \leq m$, щоб бути розв'язками системи рівнянь

$$\partial_\Theta \ell(Y, \Theta) = 0,$$

через $\mathbb{P}(G_i | \mathbf{y}_j)$. Потім треба визначити $\mathbb{P}(G_i | \mathbf{y}_j)$ ітеративно. Наприклад, на першій ітерації покласти $\mathbb{P}(G_i | \mathbf{y}_j) = \mathbb{P}(G_i) = \frac{1}{m}$, обчислити α_i , $\boldsymbol{\mu}_i$ та Σ_i , $1 \leq i \leq m$, потім перепозначити

$$\hat{\mathbb{P}}(G_i | \mathbf{y}_j) := \frac{\hat{\alpha}_i f(\mathbf{y}_j, \hat{\boldsymbol{\mu}}_i, \hat{\Sigma}_i)}{h(\mathbf{y}_j)},$$

і т.д.

Отже, шукаємо критичні точки $\ell(Y, \Theta)$. Подиференціювавши (12.1) по α_i та прирівнявши отриманий вираз до 0, отримаємо

$$\frac{f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i)}{h(\mathbf{y}_j, \Theta)} - \lambda = 0, \quad i = 1, \dots, m. \quad (12.3)$$

Домноживши на α_i і просумувавши, ми отримаємо $\lambda = n$:

$$\lambda = \lambda \sum_{i=1}^m \alpha_i = \sum_{i=1}^n \frac{\sum_{i=1}^m \alpha_i f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i)}{h(\mathbf{y}_j, \Theta)} = \sum_{j=1}^n 1 = n.$$

Тоді, підставивши в (12.3) та використавши означення $\mathbb{P}(G_i | \mathbf{y}_j)$, отримаємо

$$\alpha_i = \frac{1}{n} \sum_{j=1}^n \mathbb{P}(G_i | \mathbf{y}_j).$$

Продиференцюємо тепер $\ell(Y, \Theta)$ по $\boldsymbol{\mu}_i$. Отримаємо

$$\sum_{j=1}^n \alpha_i (\mathbf{y}_j - \boldsymbol{\mu}_i) \frac{f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i)}{h(\mathbf{y}_j, \Theta)} = 0.$$

Тому

$$\boldsymbol{\mu}_i \sum_{j=1}^n \mathbb{P}(G_i | \mathbf{y}_j) = \boldsymbol{\mu}_i \alpha_i n = \sum_{j=1}^n \mathbf{y}_j \mathbb{P}(G_i | \mathbf{y}_j),$$

звідки

$$\boldsymbol{\mu}_i = \frac{1}{n \alpha_i} \sum_{j=1}^n \mathbf{y}_j \mathbb{P}(G_i | \mathbf{y}_j).$$

Тепер продиференцюємо $\ell(Y, \Theta)$ по Σ_j^{-1} по Σ_i^{-1} . Ми використаємо обчислення, схоже до того, що було зроблено в (3.3):

$$\begin{aligned} \partial_{\Sigma_i^{-1}} f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i) &= f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i) \partial_{\Sigma_i^{-1}} \ln f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i) \\ &= f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i) \frac{1}{2} (\Sigma_i - (\mathbf{y}_j - \boldsymbol{\mu}_i)(\mathbf{y}_j - \boldsymbol{\mu}_i)'). \end{aligned}$$

Тому, диференцюючи $\ell(Y, \Theta)$ по Σ_j^{-1} та прирівнюючи результат до 0, отримаємо

$$\begin{aligned} \partial_{\Sigma_i^{-1}} \ell(Y, \Theta) &= \sum_{j=1}^n \frac{\alpha_i}{h(\mathbf{y}_j, \Theta)} \partial_{\Sigma_i^{-1}} f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i) \\ &= \frac{1}{2} \sum_{j=1}^n \frac{\alpha_i f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i)}{h(\mathbf{y}_j, \Theta)} (\Sigma_i - (\mathbf{y}_j - \boldsymbol{\mu}_i)(\mathbf{y}_j - \boldsymbol{\mu}_i)') \\ &= 0, \end{aligned}$$

звідки

$$\Sigma_i n \alpha_i = \Sigma_i \sum_{j=1}^n \mathbb{P}(G_i | \mathbf{y}_j) = \Sigma_i \sum_{j=1}^n \frac{\alpha_i f(\mathbf{y}_j, \boldsymbol{\mu}_i, \Sigma_i)}{h(\mathbf{y}_j, \Theta)} = \sum_{j=1}^n (\mathbf{y}_j - \boldsymbol{\mu}_i)(\mathbf{y}_j - \boldsymbol{\mu}_i)' \mathbb{P}(G_i | \mathbf{y}_j),$$

отже,

$$\Sigma_i = \frac{1}{\alpha_i n} \sum_{j=1}^n (\mathbf{y}_j - \boldsymbol{\mu}_i)(\mathbf{y}_j - \boldsymbol{\mu}_i)' \mathbb{P}(G_i | \mathbf{y}_j).$$

Замінюючи ймовірності $\mathbb{P}(G_i | \mathbf{y}_j)$ на оцінки $\hat{\mathbb{P}}(G_i | \mathbf{y}_j)$, отримуємо систему рівнянь, яким мають задовольняти оцінки $\hat{\alpha}_j$, $\hat{\boldsymbol{\mu}}_j$, $\hat{\Sigma}_j$:

$$\begin{aligned} \hat{\alpha}_i &= \frac{1}{n} \sum_{j=1}^n \hat{\mathbb{P}}(G_i | \mathbf{y}_j), \quad i = 1, \dots, m-1, \\ \hat{\boldsymbol{\mu}}_i &= \frac{1}{n \hat{\alpha}_i} \sum_{j=1}^n \mathbf{y}_j \hat{\mathbb{P}}(G_i | \mathbf{y}_j), \quad i = 1, \dots, m, \\ \hat{\Sigma}_i &= \frac{1}{n \hat{\alpha}_i} \sum_{j=1}^n (\mathbf{y}_j - \hat{\boldsymbol{\mu}}_i)(\mathbf{y}_j - \hat{\boldsymbol{\mu}}_i)' \hat{\mathbb{P}}(G_i | \mathbf{y}_j), \quad i = 1, \dots, m, \end{aligned}$$

Для того, щоб розв'язати цю систему (нелінійних) рівнянь, треба застосовувати чисельні методи, наприклад, метод Ньютона-Рафсона. Посилання на роботи в цьому напрямку можна знайти в [Ev93], [ELLS11].

Отже, ми оцінили параметри і знайшли (рекурсивно) апостеріорні ймовірності $\hat{P}(G_i|\mathbf{y})$. Тоді можна віднести \mathbf{y} до кластеру G_i з найбільшим значенням $\hat{P}(G_i|\mathbf{y})$. Цю процедуру можна повторювати. Якщо m невідомо, можна почати з $m = 1$, а потім спробувати $m = 2$, $m = 3$ і т.д. до тих пір, поки результат не буде задовільний.

Розглянемо кілька способів вибору кількості кластерів. Наприклад, можна проаналізувати, наскільки змінюються відстані між елементами при переході від k кластерів до $k+1$. Наприклад, можна обчислити середній квадрат відстані (наприклад, в евклідовій метриці) елементів від центрів кластерів. Цей параметр називається деформацією (Distortion) Distortion. З іншого боку, можна обчислити суми квадратів відстаней від елементів до найближчого кластеру. Цей параметр називається інерцією (Inertia). Тепер можна проаналізувати залежність цих параметрів від кількості кластерів, намалювавши scree plot. За оптимальну кількість кластерів можна вибрати ту, де ламана має найбільший злам (тобто при збільшенні кількості кластерів на 1 відстіні різко зменшуються).

Розглянемо ще один метод. Обираємо кількість груп n_0 , для якої вперше виконується нерівність

$$\alpha_i > \bar{\alpha} + ks_\alpha,$$

де $\alpha_1, \alpha_2, \dots, \alpha_n$ є відстянями між кластерами на етапі з $n, n-1, \dots, 1$ кластерами, $\bar{\alpha}$ та s_α є відповідно вибірковим середнім та середньоквадратичним відхиленням α_i , k – деяка константі (підбирається “вручну”, див. посилання в [R02, §14.5]).

Посилання на інші методи (та тестову статистику для перевірки гіпотези про кількість кластерів) можна подивитися в [R02, §14.5].

Такі методи, як k -means, метод медоїда (PAM-clustering: partition around medoid) добре підходять для кластерізації об'єктів, які мають сферичну або опуклу форму. Іншими словами, утворені кластери можна відносно просто розділити. Більш того, ці методи є чуттєвими до шуму та викидів. В реальній ситуації кластери можуть мати довільну форму, вихідні дані можуть бути зашумленими та мати викиди. Наприклад, якщо ми розфарбовуємо трафаретну картинку в декілька кольорів (і тим самим побудувати кластири), форма розфарбованих фігурок може бути якою завгодно.

Розглянемо метод кластерізації DBSCAN (Density-Based Spatial Clustering and Application with Noise), який було запропоновано в [EKSX96] саме для ідентифікації таких кластерів. Нижче ми наведемо результати, отримані в [EKSX96]¹⁴.

DBSCAN має наступні переваги:

- a) На відміну від k -means, DBSCAN не вимагає наперед визначити кількість кластерів.
- b) Кількість кластерів визначається самим методом.
- c) Кластери можуть мати довільну форму.
- d) DBSCAN може визначати викиди.

DBSCAN використовує підхід, близький до того, як групи об'єктів візуально сприймає людське око. Наприклад, подивимось на Рис. 10. На картинці зображено 5 кластерів, але метод k -means бачить ці кластери інакше, ніж людське око, див. Рис. 11

На картинці є області, де точки розподілені щільно, а є області, де точки з'являються як “шум”. Іншими словами, щільність точок в кластері значно вища, ніж поза ним (в так

¹⁴http://www.sthda.com/english/wiki/wiki.php?id_contents=7940

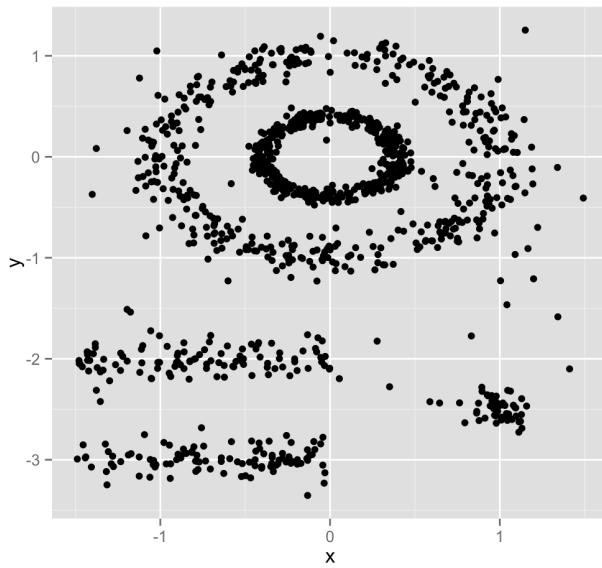


Рис. 10: Несферичні кластери

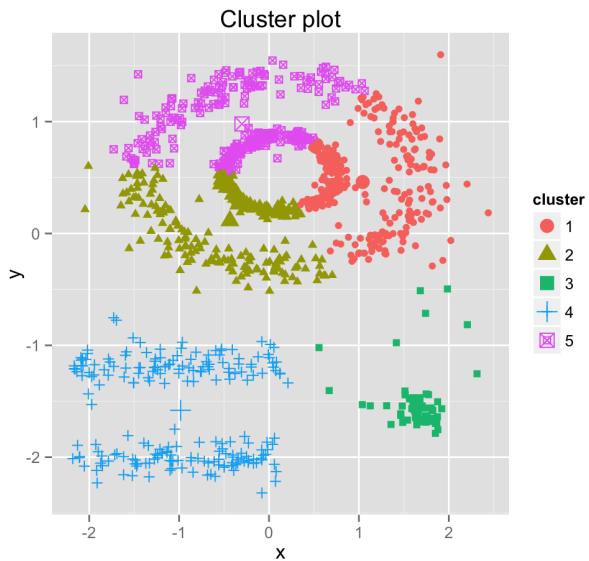


Рис. 11: Несферичні кластери

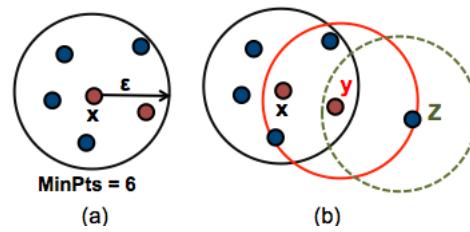


Рис. 12: Основна точка, гранична точка, шум

званих “areas of noise”). Основна ідея побудови кластерів полягає в тому, що окіл заданого радіусу кожної точки кластеру має містити принаймі мінімальну (задану) кількість точок. Тобто, в кластери об’єднують ті точки, які з людської точки зору потрібно було б згрупувати в одну множину.

Для того, щоб задати алгоритм побудови кластеру, потрібно задати два параметри – радіус ε -окола точки та мінімальну кількість точок MinPt в цьому ε -околі.

Точка x датасету, яка має принаймні MinPts сусідів в своєму околу, називається ядровою точкою (core point). Точка x називається граничною точкою (border point), якщо вона має менше ніж MinPt сусідів в своєму ε -околі, але належить ε -околу деякої основної точки. Якщо точка не є основною або граничною, ми будемо її називати викидом або шумом. На Рис. 12 зображені різні типи точок з MinPts = 6. Тут x – основна точка, y – гранична, а z – викид.

Ми будемо говорити, що точка x прямо щільно досяжна (directly density reachable) з точки y , якщо

i) x належить ε -околу точки y

ii) y – основна точка.

Точка x називається щільно досяжною (density reachable) з точки y , якщо існує множина точок, які ведуть з x в y . Точка x називається щільно зв’язною (density connected) з точки y , якщо існує множина точок, які ведуть з x в y .

Точки x та y називаються щільно зв’язними, якщо існує основна точка z , така що x та y є щільно досяжними з z . Щільно оснований кластер (density-based, надалі: DB) визначається як група щільно зв’язаних точок.

Алгоритм DBSCAN працює наступним чином:

- Дляожної точки x_i обчислюється відстань між x_i та іншими точками;
- Знаходимо всі сусідні точки в ε -околі x_i ;
- Кожна точка, кількість сусідів якої в ε -околі більше або дорівнює MinPts, позначається як основні;
- Дляожної такої основної точки, якщо вона ще не приписана до якогось кластеру, утворюємо новий кластер;
- Знаходимо рекурсивно всі її щільно зв’язні точки та припишемо їх тому ж кластеру, що і основна точка;
- Ітеруємо цей процес для всіх тих точок, які ще не відвідали;
- Ті точки, які не попали в жоден кластер, назовемо викидами, або шумом.

Ми розглянули лише декілька методів кластерізації. Більше можна прочитати в [ELLS11]. Детальний огляд сучасних методів кластерізації можна почитати в [ИА] (мова Python, але аналіз містить загальні пояснення, в якій ситуації який метод краще працює).

12.2 Приклади

Завантажимо пакети **mclust**, **mvtnorm**, **factoextra** та відповідні бібліотеки. Завантажимо також **ggplot2**, **dplyr** та **stats**. Завантажимо наступні бібліотеки:

1. Розглянемо спочатку ієархічну кластерізацію. Завантажимо таблицю, в якій вказані дані по злочинності у містах (див. Таблиця 15.13 [R02]).

```
city <- read.table("T14_1_CITYCRIME.dat")
city
```

V1	V2	V3	V4	V5	V6	V7	V8	
1	Atlanta	16.5	24.8	106	147	1112	905	494
2	Boston	4.2	13.3	122	90	982	669	954
3	Chicago	11.6	24.7	340	242	808	609	645
4	Dallas	18.1	34.2	184	293	1668	901	602
5	Denver	6.9	41.5	173	191	1534	1368	780
6	Detroit	13.0	35.7	477	220	1566	1183	788
7	Hartford	2.5	8.8	68	103	1017	724	468
8	Honolulu	3.6	12.7	42	28	1457	1102	637
9	Houston	16.8	26.6	289	186	1509	787	697
10	KC	10.8	43.2	255	226	1494	955	765
11	LA	9.7	51.8	286	355	1902	1386	862
12	NO	10.3	39.7	266	283	1056	1036	776
13	NY	9.4	19.4	522	267	1674	1392	848
14	Portland	5.0	23.0	157	144	1530	1281	488
15	Tucson	5.1	22.9	85	148	1206	756	482
16	Washington	12.5	27.6	524	217	1496	1003	739

Для виконання ієархічної кластерізації використаємо функцію **hclust**; за замовченням, використовується евклідова відстань.

```
hc <- hclust(dist(city[, 2:8]))
hc
```

Call:

```
hclust(d = dist(city[, 2:8]))
```

```
Cluster method : complete
Distance       : euclidean
Number of objects: 16
```

Якщо потрібно змініти відстань, треба задати інший параметр в **method**, наприклад, **method="manhattan"**,

$$d(x, y) := \sum_{j=1}^n |x_i - y_j|.$$

Зауважимо, що ми спочатку застосовуємо функцію до матриці **city[,2:8]**, тому відстані треба змінити всередині функції **dist**. Щоб вибрати інший тип кластерізації, треба змінити параметр **method** в функції **hclust**.

```
hc1 <- hclust(dist(city[,2:8], method="manhattan"), method = "single")
hc1
```

Call:
hclust(d = dist(city[, 2:8], method = "manhattan"), method = "single")

Cluster method : single
Distance : manhattan
Number of objects: 16

Або використовуючи максимум:

$$d(x, y) := \max_{1 \leq j \leq n} |x_i - y_j|.$$

```
hc2 <- hclust(dist(city[,2:8], method="maximum"), method = "single")
hc2
```

Call:
hclust(d = dist(city[, 2:8], method = "maximum"), method = "single")

Cluster method : single
Distance : maximum
Number of objects: 16

У даному випадку всі ці методи дають однакові результати. Намалюємо дендрограму:

```
hcd1 <- as.dendrogram(hc1)
plot(hcd1, main = "Manhattan distance")
```

2. Застосуємо до попереднього прикладу метод k середніх.

```
kmeans.city <- kmeans(city[,2:8], 4, iter.max = 10, nstart = 1)
kmeans.city
```

K-means clustering with 2 clusters of sizes 6, 10

Cluster means:
V2 V3 V4 V5 V6 V7 V8
1 8.366667 22.36667 164.5 168.8333 1030.167 783.1667 636.5
2 10.580000 31.57000 290.9 212.7000 1583.000 1135.8000 720.6

Clustering vector:
[1] 1 1 1 2 2 2 1 2 2 2 1 2 2 1 2

Within cluster sum of squares by cluster:

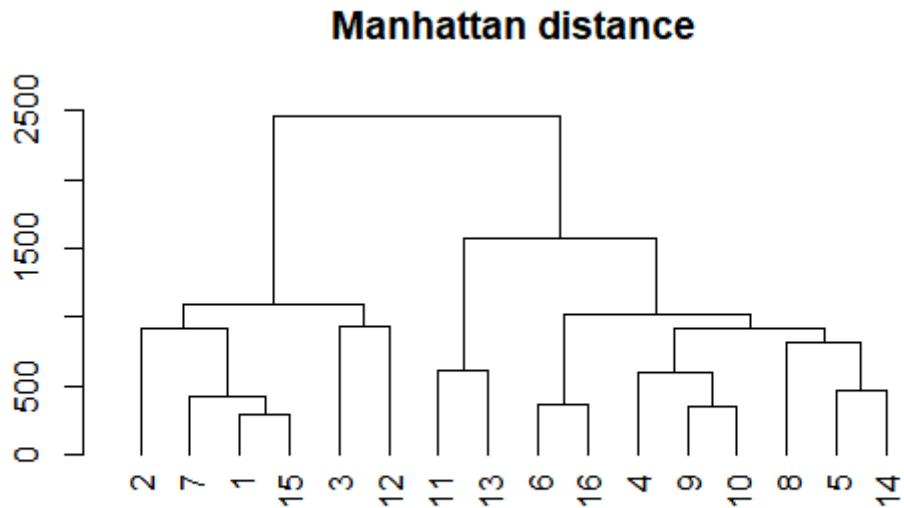


Рис. 13: Кластерізація за допомоги відстані manhattan, method = complete

```
[1] 501946.6 1035952.6
(between_SS / total_SS = 52.6 %)
```

Available components:

```
[1] "cluster"      "centers"       "totss"         "withinss"      "tot.withinss"
[6] "betweenss"    "size"          "iter"          "ifault"
```

Вибір параметру *nstarts* більшим за 1 означає, що буде взято декілька конфігурацій, і в результаті повертається та, де сума квадратів відстаней всередині кластерів найменша.

Виклик kmeans.city\$cluster дає розподіл по кластерам:

```
1 1 1 3 2 4 1 2 3 3 4 1 4 2 1 3
```

3. Розглянемо метод сумішей. Ми зробимо кластерізацію використовуючи пакет **mclust**. Ми згенеруємо гаусівську суміш і розіб'ємо вибірку по 3-м кластерам.

```
m_1 <- c(1, 1)
m_2 <- c(-3, 4)
m_3 <- c(5, 5)
sig_1<- matrix(c(1, -.5, -.5, 1), nrow =2, byrow= FALSE)
sig_2<- matrix(c(2, -.5, -.5, 2), nrow =2, byrow= FALSE)
sig_3<- matrix(c(1, 0, 0, 1), nrow =2, byrow= FALSE)
mv_1<- rmvnorm(30, m_1, sig_1)
mv_2<- rmvnorm(20, m_2, sig_2)
mv_3<- rmvnorm(50, m_3, sig_3)
```

```

plot(mv_1, col = "red", xlim=c(-5,10), ylim=c(-2,10))
points(mv_2, col= "blue")
points(mv_3, col= "green")

```

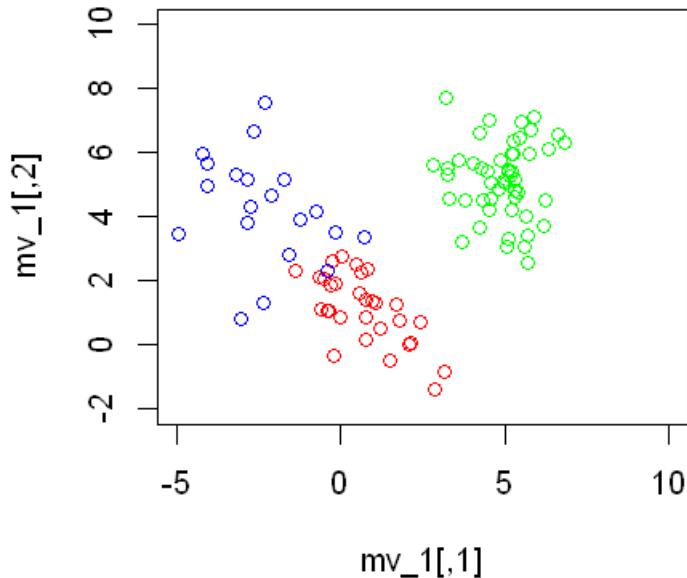


Рис. 14: 3 класи нормальню розподілених векторів

Як видно на картинці, ми маємо 3 яскраво виражені групи. Зробимо тепер кластеризацію методом сумішей. Для цього використаємо функцію Mclust (параметр 3: кількість кластерів).

```

X <- rbind(mv_1,mv_2,mv_3)
data_1 = Mclust(X, G = 3)
summary(data_1, parameters = TRUE)

-----
Gaussian finite mixture model fitted by EM algorithm
-----

Mclust EII (spherical, equal volume) model with 3 components:

log-likelihood    n      df          BIC          ICL
-406.8201        100     9       -855.0868     -857.784

Clustering table:
1  2  3

```

36 14 50

Mixing probabilities:

1	2	3
0.3626648	0.1376143	0.4997208

Means:

	[,1]	[,2]	[,3]
[1,]	0.3799153	-2.897224	4.937656
[2,]	1.3840730	5.031433	5.182865

Variances:

	[,,1]		[,1]	[,2]
[1,]	1.314383	0.000000		
[2,]	0.000000	1.314383		

	[,,2]		[,1]	[,2]
[1,]	1.314383	0.000000		
[2,]	0.000000	1.314383		

	[,,3]		[,1]	[,2]
[1,]	1.314383	0.000000		
[2,]	0.000000	1.314383		

Про метод ЕМ (Expectation-Maximization) можна подивитись, наприклад, в он-лайн матеріалі [Br].

Тут

$$BIC = -2 \ln L(X, \theta) + k \ln n,$$

де $L(X, \theta)$ – функція вірогідностей, n – кількість спостережень, k – кількість незалежних параметрів. У нас кількість незалежних параметрів 9, $\ln L(X, \theta) = -381.5052$. Відповідно, $2 * 381.5052 + 9 * \ln(100) = 804.457 = -BIC$.

Можна застосовувати інший критерій для максимізації, який теж враховує кількість класів

$$AIC = 2(k - \ln L(X, \theta))$$

де k – кількість невідомих параметрів (див. Akaike information criterion (AIC), [Ak74]).

Зобразимо результати графічно (див. Рисунок 15).

```
plot(data_1, what = "classification")
```

Параметр `what = "classification"` означає, що ми обираємо графічне зображення, яке відображає класифікацію. Якщо обрати параметр `what = "uncertainty"` – отримаємо кар-

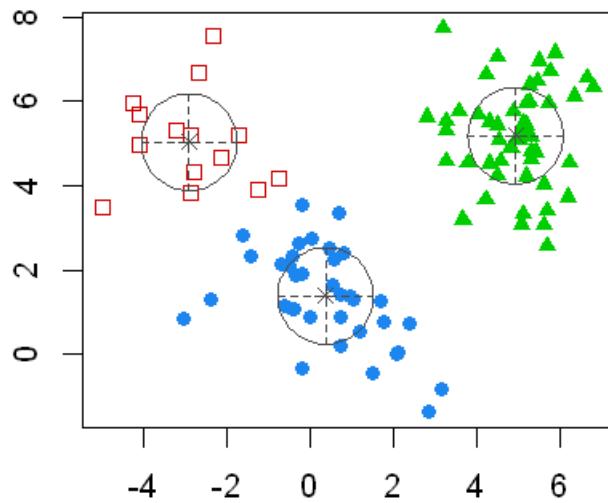


Рис. 15: Результат кластерізації

тинку, на якій будуть чітко відображені "проблемні" елементи, тобто ті, які невідомо, до якого кластеру приєднати.

Як подивитися атрибути:

```
attributes(data_1)
```

```
$names
'call' 'data' 'modelName' 'n' 'd' 'G' 'BIC' 'loglik' 'df' 'bic' 'icl' 'hypvol'
'parameters' 'z' 'classification' 'uncertainty'
```

```
$class
'Mclust'
```

Подивимось, наскільки адекватно зроблена кластерізація. Визначимо параметр, який визначає, до якого кластеру належить спостереження.

```
cl = data_1$classification
```

Створимо вектор класів по вихідній вибірці та переформатуємо наші дані в наступну таблицю:

```
cl = data_1$classification
num = c(rep(1,30), rep(2,20), rep(3,50))
Dat_old = data.frame(cbind(num,X))
```

Задамо дані, отримані в результаті кластерізації.

```
Dat_new = data.frame(cbind(cl, X))
```

Нарешті, побудуємо матрицю невідповідностей:

```
cm2 <- table(Dat_old$num, Dat_new$cl)
cm2
```

```
1   2   3
1 30  0  0
2  6  14  0
3  0  0  50
```

Отже, ми майже все класифікували правильно!

Перейдмо тепер до методу DBSCAN. Ми будемо використовувати пакети **fpc**, **dbSCAN** (безпосередньо для самого методу) та **factoextra** (для візуалізації даних). Функція `dbSCAN()` присутня в обох пакетах. Тому, для того, щоб визначити, який пакет ми використовуємо, ми будемо писати, `fpc::dbSCAN()` або `dbSCAN::dbSCAN()`, відповідно.

```
dbSCAN(data, eps, MinPts = 5, scale = FALSE,
       method = c("hybrid", "raw", "dist"))
```

Тут `data` – це матриця або `data frame` або так звана матриця відстаней (dissimilarity matrix), тобто дані формату `*.dist`. В останньому випадку треба обирати `method = "dist"`, оскільки ми будемо працювати з матрицею відстаней; інакше буде використана евклідова відстань. Метод `raw` означає, що ми не оброблюємо наші дані, а в гібридному методі ми працюємо з необробленими даними, але обчислюємо матрицю відстаней.

Завантажимо дані з пакету **factoextra** та зробимо аналіз (див. Рис. 16).

```
data("multishapes", package = "factoextra")
df <- multishapes[, 1:2]
set.seed(123)
res.fpc <- fpc::dbSCAN(df, eps = 0.15, MinPts = 5)
plot(res.fpc, df, main = "DBSCAN", frame = FALSE)
```

Функція `plot.dbSCAN()` використовує різні типи точок (трикутники та кружочки) для основних точок та для граничних. Чорний колір відповідає викидам.

Можна вивести інформацію про те, як кластерізовані точки за допомогою `fpc::dbSCAN()`:

```
print(res.fpc)
```

```
dbSCAN Pts=1100 MinPts=5 eps=0.15
       0   1   2   3   4   5
border  31  24   1   5   7   1
seed      0 386 404  99  92  50
total    31 410 405 104  99  51
```

В цій таблиці назви колонок – це номер кластеру. Кластер з номерам відповідає викидам (тобто чорним точкам на графіку).

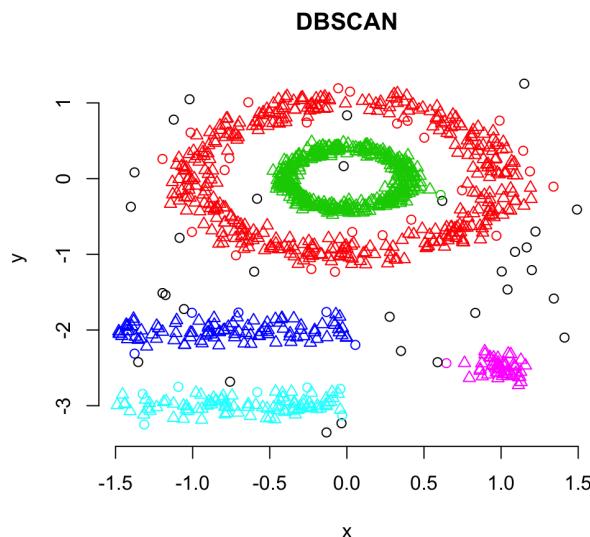


Рис. 16: DBSCAN кластерізація

Можна перевірити, що DBSCAN краще класетрізує такі дані ніж, наприклад, k -means. Ще одна функція, яка виконує кластерізацію, це fviz_cluster() з пакету **factoextra**:

```
fviz_cluster(res.fpc, df, stand = FALSE, frame = FALSE, geom = "point")
```

Можна зробити аналіз обома методами, та перевірити, що результати співпадають:

```
res.db <- dbSCAN::dbSCAN(df, 0.15, 5)
print(res.db)
```

```
DBSCAN clustering for 1100 objects.
Parameters: eps = 0.15, minPts = 5
Using euclidean distances and borderpoints = TRUE
The clustering contains 5 cluster(s) and 31 noise points.
```

```
0   1   2   3   4   5
31 410 405 104  99  51
```

Available fields: cluster, eps, minPts, dist, borderPoints

Як ми бачимо, результати співпадають.

В методі DBSCAN потрібно визначити оптимальне ε та MinPts. У прикладі, що наведено вище, ми взяли $\varepsilon = 0.15$ та MinPts = 5. Одним з обмежень DBSCAN є те, що він є чутливим до вибору ε , зокрема, якщо кластери мають різну щільність точок. Якщо ε мале, розріжені кластери будуть сприйматися як шум. Якщо ε велике, то більш щільні кластери будуть склеюватись. Можливо, для кластерізації даних з різним рівнем щільності, треба використовувати різні ε . Тому постає питання – а як же обрати ε ?

Можна застосувати метод k найближчих сусідів, а саме: обчислити середню відстань будь-якої точки до її k найближчих сусідів. Значення k визначається самим користувачем. Далі, будується графік цих відстаней у зростаючому напрямку, див. Рис. 17. Як і в методі головних компонент, ми визначаємо те значення ε , на якому графік має “злам”. Цей злам означає той момент, де відбувається різка зміна в відстанях до k найближчих сусідів (k -distance), при яких відбувається кластерізація.

Використаємо функції `kNNdistplot()` (в пакеті `dbSCAN`) для побудови такого графіку залежності k -distance від ε :

```
dbSCAN::kNNdistplot(df, k = 5)
abline(h = 0.15, lty = 2)
```

Тут при значенні $\varepsilon = 0.15$ відбувається злам, тому будемо вважати, що оптимальне значення ε дорівнює 0.15.

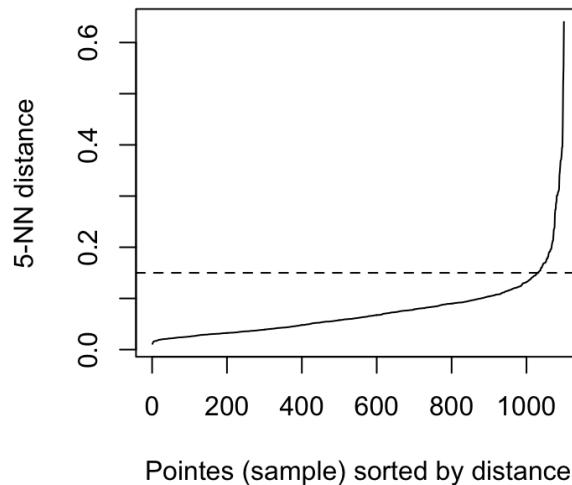


Рис. 17: Вибір оптимального ε

Зробимо тепер прогноз кластерізації перших 10 елементів, маючи результати кластерізації 11-111 елементу.

```
df1 <- multishapes[11:111, 1:2]
df2 <- multishapes[1:10, 1:2]
res2.fpc <- fpc::dbSCAN(df1, eps = 0.15, MinPts = 5)
predict.dbSCAN(res2.fpc, df1, df2)
```

0 0 0 0 0 0 3 1 0 5

13 Факторний аналіз

Література: [R02, Гл.13]. Пакети `car`, `ggplot2`.

Нехай є змінні y_1, y_2, \dots, y_p . Якщо ці змінні залежні, то фактична розмірність системи менша за p . Задача факторного аналізу полягає в тому, щоб зменшити розмірність шляхом "зменшення повторень" та виділення головних факторів, які впливають на систему.

Наприклад, розглянемо кореляційну матрицю

$$\begin{pmatrix} 1 & 0.9 & 0.05 & 0.05 & 0.05 \\ 0.9 & 1 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 1 & 0.9 & 0.9 \\ 0.05 & 0.05 & 0.9 & 1 & 0.9 \\ 0.05 & 0.05 & 0.9 & 0.9 & 1 \end{pmatrix}.$$

Виходячи зі значень кореляцій, доцільно розділити систему на два фактори:

фактор 1: y_1, y_2

фактор 2: y_3, y_4, y_5 .

Розглянемо ще один приклад, який виникає в хімічних експериментах (див. [Ma02, §1.4]).

Маємо 5 різних сумішей, через які пропускають ультрафіолетові промені певної частоти (6 випадків). Запишемо результати вимірювань (коєфіцієнти поглинання) в матрицю A , див. Таблицю 3.

Табл. 3: Таблиця поглинання, 5 сумішей

Номер випробування	Довжина хвилі	Суміш 1	Суміш 2	Суміш 3	Суміш 4	Суміш 5
1	278 nm	0.005	0.031	0.063	0.091	0.046
2	274 nm	0.040	0.172	0.356	0.444	0.218
3	270 nm	0.103	0.283	0.484	0.471	0.208
4	266 nm	0.116	0.323	0.562	0.548	0.241
5	262 nm	0.125	0.318	0.516	0.450	0.185
6	258 nm	0.104	0.267	0.430	0.376	0.154

Припустимо, що на коєфіцієнт поглинання впливають n факторів. Тоді елемент a_{ij} матриці A (що справа в таблиці) можна зобразити наступним чином:

$$a_{ij} = \sum_{j=1}^n w_{ij} m_{jk},$$

де w_{ij} та m_{jk} – це коєфіцієнти поглинання довжини хвилі під номером i у суміші k , або в матричному вигляді:

$$A = W_{abstr} M_{abstr},$$

де $W_{abstr} = (w_{ij})$, $M = (m_{jk})$. Наша задача – зменшити кількість факторів, які впливають на результат (не всі компоненти розчину впливають однаково, бажано їх "згрупувати").

Іншими словами, треба знайти такі матриці W_{real} та M_{real} , що

$$A = W_{real}M_{real}.$$

Такий запис дозволить ефективно утворювати розчин з "вірними" пропорціями компонентів.

Порівняємо методи факторного аналізу з методом головних компонент.

Метод головних компонент	Факторний аналіз
Головні компоненти є комбінаціями вихідних змінних	Вихідні компоненти є комбінаціями факторів
Пояснююємо найбільшу частку дисперсії	Обираємо фактори згідно з тим, як кореллюють змінні.

Розглянемо наступну модель. Зобразимо y_i як лінійні комбінації факторів f_1, f_2, \dots, f_m , та похибки ε_k , $1 \leq k \leq m$. А саме, наша модель допускає зображення:

$$\begin{aligned} y_1 - \mu_1 &= \lambda_{11}f_1 + \lambda_{12}f_2 + \dots + \lambda_{1m}f_m + \varepsilon_1 \\ y_2 - \mu_2 &= \lambda_{21}f_1 + \lambda_{22}f_2 + \dots + \lambda_{2m}f_m + \varepsilon_2 \\ &\dots \\ y_p - \mu_p &= \lambda_{p1}f_1 + \lambda_{p2}f_2 + \dots + \lambda_{pm}f_m + \varepsilon_p. \end{aligned} \tag{13.1}$$

В ідеалі m має бути значно меншим за p , інакше описання \mathbf{y} за допомоги факторів \mathbf{f} не є якісним. Тут \mathbf{f} – випадкові змінні, які описують \mathbf{y} . Коєфіцієнти λ_{ij} називаються навантаженнями (loadings), наприклад, вплив фактору j на змінну y_i . Ще одна принципова відмінність моделі факторного аналізу від моделі лінійної регресії полягає в тому, що 1) насправді ми не спостерігаємо фактори, а їх треба знайти; 2) у нас може бути не n спостережень, а одне, тобто у нас одна вибірка з популяції, з середнім значенням $\boldsymbol{\mu}$ та коваріаційною матрицею Σ .

Нам потрібно оцінити навантаження λ_{ij} та адекватно підібрати фактори.

Зробимо наступні припущення.

1. $\mathbb{E}f_i = 0$, $\text{Var } f_i = 1$, $i = 1, \dots, m$;
2. $\mathbb{E}\varepsilon_i = 0$, $\text{Var } \varepsilon_i = \psi_i$, $\text{cov}(\varepsilon_i, \varepsilon_j) = 0$, $i, j = 1, \dots, p$;
3. $\text{cov}(f_i, f_j) = 0$, $i \neq j$, $i, j = 1, \dots, m$;
4. $\text{cov}(\varepsilon_i, f_j) = 0$, $i = 1, \dots, p$, $j = 1, \dots, m$.

Тоді дисперсія i -го спостереження дорівнює

$$\text{Var } y_i = \lambda_{i1}^2 + \lambda_{i2}^2 + \dots + \lambda_{im}^2 + \psi_i. \tag{13.2}$$

У векторному вигляді цю модель можна записати наступним чином:

$$\mathbf{y} - \boldsymbol{\mu} = \Lambda \mathbf{f} + \boldsymbol{\varepsilon}, \tag{13.3}$$

де $\mathbf{y} = (y_1, y_2, \dots, y_p)', \mathbf{f} = (f_1, f_2, \dots, f_m)', \boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p)'$,

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1m} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2m} \\ \dots & & & \\ \lambda_{p1} & \lambda_{p2} & \dots & \lambda_{pm} \end{pmatrix}, \quad \text{cov}(\boldsymbol{\varepsilon}) = \Psi = \begin{pmatrix} \Psi_1 & 0 & \dots & 0 \\ 0 & \Psi_2 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & \Psi_p \end{pmatrix}, \quad \text{cov}(\mathbf{f}, \boldsymbol{\varepsilon}) = 0.$$

Нехай $(\sigma_{ij}) = \text{cov}(\mathbf{y}) = \text{cov}(\Lambda \mathbf{f} + \boldsymbol{\varepsilon})$. Оскільки $\text{cov}(\mathbf{f}, \boldsymbol{\varepsilon}) = 0$,

$$\text{cov}(\mathbf{y}) = \text{cov}(\Lambda \mathbf{f}) = \Lambda \text{cov}(\mathbf{f}) \Lambda' + \Psi = \Lambda I \Lambda' + \Psi.$$

Розглянемо випадок $m = 2$. Тоді $\text{cov}(y_1, y_2) = \lambda_{11}\lambda_{21} + \lambda_{12}\lambda_{22}$. Якщо y_1 та y_2 мають "багато спільного", то вони мають схожі навантаження на f_1 та f_2 . У цьому випадку або $\lambda_{11}\lambda_{21}$ або $\lambda_{12}\lambda_{22}$ великі. Якщо y_1 та y_2 мають мало спільного, то навантаження λ_{11} та λ_{21} на f_1 та навантаження λ_{12} , λ_{22} на f_2 різні. У цьому випадку $\lambda_{11}\lambda_{21}$ та $\lambda_{12}\lambda_{22}$ малі.

Розглянемо матрицю $\Lambda = \text{cov}(\mathbf{y}, \mathbf{f})$. Наприклад, елемент $\text{cov}(y_1, f_2)$ щієї матриці дорівнює

$$\begin{aligned} \text{cov}(y_1, f_2) &= \mathbb{E}(y_1 - \mu_1)f_2 \\ &= \mathbb{E}((\lambda_{11}f_1 + \dots + \lambda_{1m}f_m)f_2) - \mathbb{E}(\lambda_{11}f_1 + \dots + \lambda_{1m}f_m)\mathbb{E}f_2 \\ &= \lambda_{11} \text{cov}(f_1, f_2) + \lambda_{12} \text{cov}(f_2, f_2) + \dots + \lambda_{1m} \text{cov}(f_m, f_2) \\ &= \lambda_{12}. \end{aligned}$$

Отже,

$$\text{cov}(y_i, f_j) = \lambda_{ij}, \quad i = 1, \dots, p, \quad j = 1, \dots, m. \quad (13.4)$$

Також,

$$\sigma_{ii} = \text{Var}(y_i) = \lambda_{i1}^2 + \lambda_{i2}^2 + \dots + \lambda_{im}^2 + \psi_i = h_i^2 + \psi_i. \quad (13.5)$$

Величина h_i^2 називається спільною дисперсією (communality), а величина ψ_i – залишковою дисперсією (specific variance, residue variance). Спільна дисперсія – це частка дисперсії, яка пояснюється факторами, а ψ_i – це частина дисперсії, яка властива лише y_i (див. також [Ма02, §3.5]). Величина ψ_i складається з систематичної та випадкової похибки, які притаманні лише i -й змінній. Можна влаштувати модель таким чином, щоб для інформація, що стосується i -ї змінної, містилась в ψ_i , а h_i^2 містила інформацію, яка відображається кореляцією y_i та іншими змінними. Саме тому у методі головних факторів (див. нижче) обирають множинний коефіцієнт кореляції в якості оцінки h_i^2 .

Навантаження на фактори обираються неоднозначно. Нехай T ортогональна матриця, тобто $TT' = I$. Тоді

$$\mathbf{y} - \boldsymbol{\mu} = \underbrace{\Lambda T}_{\Lambda^*} \underbrace{T' \mathbf{f}}_{\mathbf{f}^*} + \boldsymbol{\varepsilon} = \Lambda^* \mathbf{f}^*,$$

При цьому $\Lambda^* \Lambda^* = \Lambda T (\Lambda T)' = \Lambda \Lambda'$. Фактори \mathbf{f}^* задовольняють ті самі умови, що і \mathbf{f} : $\mathbb{E} \mathbf{f}^* = 0$, $\text{cov}(\mathbf{f}^*) = I$, $\text{cov}(\mathbf{f}^*, \boldsymbol{\varepsilon}) = 0$. Величини h_i^2 теж не змінилися. Дійсно, запишемо h_i^2 як $h_i^2 = \boldsymbol{\lambda}'_i \boldsymbol{\lambda}$. Зауважимо, що i -й рядок матриці Λ^* – це $(\boldsymbol{\lambda}_i^*)' = \boldsymbol{\lambda}'_i T$. Тоді

$$(h_i^2)^2 = (\boldsymbol{\lambda}_i^*)' \boldsymbol{\lambda}_i^* = \boldsymbol{\lambda}'_i T T' \boldsymbol{\lambda}_i = \boldsymbol{\lambda}'_i \boldsymbol{\lambda}_i = h_i^2.$$

У наступних двох підрозділах ми розглянемо методи оцінки навантажень та спільної дисперсії.

13.1 Метод головних компонент у факторному аналізі

Позначимо через S вибіркову коваріаційну матрицю. Наша задача – знайти оцінку $\hat{\Lambda}$ матриці навантажень Λ . При цьому має виконуватись співвідношення

$$S = \hat{\Lambda}\hat{\Lambda}' + \Psi. \quad (13.6)$$

Запишемо спектральний розклад $S = CDC'$, де C – ортогональна матриця, побудована з власних векторів S , $D = \text{diag}(\theta_1, \theta_2, \dots, \theta_p)$, де $\theta_i > 0$, $1 \leq i \leq p$, ϵ власними числами S .

Тоді

$$S = CD^{1/2}D^{1/2}C' = CD^{1/2}(CD^{1/2})' =: \Lambda^*(\Lambda^*)'.$$

В якості оцінки $\hat{\Lambda}$ матриці Λ можна було б обрати $CD^{1/2}$, але ця матриця має розмірність $p \times p$, а нам хотілося б зменшити кількість факторів, тобто було б добре задати як матрицю розмірністю $m < p$.

Нехай $D_1 = \text{diag}(\theta_1, \theta_2, \dots, \theta_m)$, $C_1 = (c_1, c_2, \dots, c_m)$. Покладемо $\hat{\Lambda} = C_1 D_1^{1/2}$. Наприклад, при $p = 3$, $m = 2$

$$\begin{pmatrix} \hat{\lambda}_{11} & \hat{\lambda}_{12} \\ \hat{\lambda}_{21} & \hat{\lambda}_{22} \\ \hat{\lambda}_{31} & \hat{\lambda}_{32} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix} \begin{pmatrix} \sqrt{\theta_1} & 0 \\ 0 & \sqrt{\theta_2} \end{pmatrix} = \begin{pmatrix} \sqrt{\theta_1}c_{11} & \sqrt{\theta_2}c_{12} \\ \sqrt{\theta_1}c_{21} & \sqrt{\theta_2}c_{22} \\ \sqrt{\theta_1}c_{31} & \sqrt{\theta_2}c_{32} \end{pmatrix} \quad (13.7)$$

Наведена вище формула пояснює називу ”метод головних компонент”. Стовбчики в (13.7) є пропорційними власним векторам S , а отже навантаження на j -й фактор є пропорційним до j -ї головній компоненті. Фактори пов’язані з m першими головними компонентами. Але після повороту навантажень інтерпретація як правило змінюється, тому не зовсім ясно, як інтерпретувати ці нові фактори.

З попереднього, i -й діагональний елемент $\hat{\Lambda}$ дорівнює $\sum_{j=1}^m \hat{\lambda}_{ij}^2$. Отже, для того, щоб апроксимувати (13.6), покладемо

$$\hat{\psi}_i = s_{ii} - \sum_{j=1}^m \hat{\lambda}_{ij}^2.$$

Тоді

$$S \approx \hat{\Lambda}\hat{\Lambda}' + \hat{\Psi}, \quad (13.8)$$

де $\Psi = \text{diag}(\hat{\psi}_1, \hat{\psi}_2, \dots, \hat{\psi}_p)$. Дисперсії в (13.8) задаються точно, в той час як ми лише апроксимуємо недіагональні елементи матриці S .

Сума квадратів елементів $\hat{\Lambda}$ по колонкам і по рядкам дорівнює спільній дисперсії \hat{h}_i^2 та власному числу θ_j , відповідно:

$$\hat{h}_i^2 = \sum_{j=1}^m \hat{\lambda}_{ij}^2,$$

$$\sum_{i=1}^p \hat{\lambda}_{ij}^2 = \sum_{i=1}^p (\sqrt{\theta_j}c_{ij})^2 = \theta_j \sum_{i=1}^p c_{ij}^2 = \theta_j,$$

де ми використали те, що власні вектори є нормованими. Отже, дисперсія змінної розкладається в частину, яка пояснюється факторами і частину, яка є притаманною змінній:

$$s_{ii} = \hat{h}_i^2 + \hat{\psi}_i.$$

Таким чином, j -й фактор має внесок $\hat{\lambda}_{ij}^2$ до дисперсії s_{ii} , а внесок в сумарну дисперсію $\text{trace}(S) = s_{11} + s_{22} + \dots + s_{pp}$ дорівнює, відповідно, $\sum_{i=1}^p \hat{\lambda}_{ij}^2$. Отже, пропорція j -го фактору дорівнює

$$\frac{\sum_{i=1}^p \hat{\lambda}_{ij}^2}{\text{trace}(S)} = \frac{\theta_j}{\text{trace}(S)}. \quad (13.9)$$

Якщо ми виходили з матриці кореляцій R , а не з матриці коваріацій S , то має вигляд

$$\frac{\sum_{i=1}^p \hat{\lambda}_{ij}^2}{\text{trace}(R)} = \frac{\theta_j}{p}. \quad (13.10)$$

Наскільки хороши є підгонка можна побачити, розглянувши матрицю похибок

$$E = S - \hat{\Lambda}\hat{\Lambda}' - \hat{\Psi}.$$

Діагональні елементи цієї матриці дорівнюють 0. Якщо підгонка є хорошию, то і інші елементи близькі до 0.

13.2 Метод головних факторів

Розглянемо тепер метод головних факторів (principal factor method, principal axis method).

В методі головних компонент у факторному аналізі ми працювали з матрицею S , але не з Ψ (матрицю Ψ ми добудували потім). У методі головних факторів ми спочатку будуємо оцінку $\hat{\Psi}$ та задаємо $\hat{\Lambda}$ за допомоги співвідношення (13.8) (або $R - \hat{\Psi} \approx \hat{\Lambda}\hat{\Lambda}'$, якщо виходити з матриці кореляцій).

Діагональним елементом $S - \hat{\Psi}$ є $\hat{h}_i^2 = \hat{s}_{ii}^2 - \hat{\psi}_i$ (або $\hat{h}_i^2 = 1 - \hat{\psi}_i$, якщо виходити з матриці R). Оскільки спільна дисперсія \hat{h}_i^2 спільна для всіх факторів, то природно оцінити її через коефіцієнт кореляції змінної i та $p - 1$ інших змінних. Тому в якості оцінки \hat{h}_i^2 оберемо множинний коефіцієнт кореляції R_i^2 між y_i та іншими $p - 1$ змінними y_j . Нагадаємо, що

$$R_i^2 = \rho_{y_i X}^2,$$

де $\mathbf{x} = (y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_p)$, а множинний коефіцієнт кореляції між $y \in \mathbb{R}$ та $\mathbf{x} \in \mathbb{R}^{p-1}$ визначається наступним чином:

$$\rho_{y\mathbf{x}} = \sqrt{\Sigma'_{y\mathbf{x}} \Sigma_{\mathbf{x}}^{-1} \Sigma_{y\mathbf{x}}} / \sigma_y.$$

Тут $\Sigma'_{y\mathbf{x}}$ вектор, що складається з коваріацій y та x_i , та $\Sigma_{\mathbf{x}}$ – коваріаційна матриця \mathbf{x} . Тоді оцінімо \hat{h}_i^2 наступним чином (див. [R02, (13.42)]):

$$\hat{h}_i^2 = s_{ii} R_i^2 = s_{ii} - \frac{1}{s_{ii}}, \quad (13.11)$$

де s_{ii} є i -м елементом S на діагоналі, та where s^{ii} є i -м елементом S^{-1} на діагоналі.

Для того, щоб використати оцінку (13.11), матриця S не має бути сингулярною.

Після того, як ми отримали оцінки на середні дисперсії \hat{h}_i^2 , обчислимо власні значення і власні числа матриць $S - \hat{\Psi}$, візьмемо цю матрицю у якості матриці $\hat{\Lambda}\hat{\Lambda}'$, та використаємо розклад $\hat{\Lambda} = C_1 D_1^{1/2}$ для оцінювання навантажень на фактори. Тоді колонки і рядки $\hat{\Lambda}$ можна використати для отримання нових власних значень та середніх дисперсій. Як і в методі головних компонент, сума квадратів j -го стовбчика $\hat{\Lambda}$ є j -м власним значенням $S - \hat{\Psi}$, та сума квадратів j -го рядка є середньою дисперсією y_i , а також має місце (13.9).

Матриця $S - \hat{\Psi}$ (відповідно, $R - \hat{\Psi}$) не обов'язково додатно визначена, а отже, може мати від'ємні власні числа. У цьому випадку відносна частка поясненої дисперсії може бути більше 1, а потім спадає до 1 по мірі додавання власних чисел.

13.3 Вибір кількості факторів та інтерпретація

Існує декілька критеріїв, щоб обрати кількість факторів m .

1. Обрати кількість факторів m , якої достатньо, щоб пояснити 80% дисперсії.
2. Обрати кількість факторів m , яка дорівнює кількості власних чисел, які більше середнього з них. Для матриці S це число дорівнює $\sum_{j=1}^p \theta_j/p$. Якщо використовувати матрицю R , то це середнє число дорівнює 1.
3. Використати графічне зображення для того, щоб оцінити, скільки власних значень S (або R) досить. Якщо графік ламаної має крутий злам, то для того, то доцільно обрати кількість до цього крутого зламу.
4. Перевірити гіпотезу про те, що вірне значення розмірності матриці дорівнює m , тобто $H_0: \Sigma = \Lambda\Lambda' + \Psi$, де Λ має розмірність $p \times m$.

В останньому випадку використовується статистика (див. [R02, (13.47)])

$$\left(n - \frac{2p + 4m + 11}{11} \right) \ln \left(\frac{|\hat{\Lambda}\hat{\Lambda}' + \hat{\Psi}|}{|S|} \right),$$

яка за виконання гіпотези H_0 має розподіл χ^2_ν , де $\nu = \frac{1}{2}[(p - m)^2 - m - p]$, а $\hat{\Lambda}$ та $\hat{\Psi}$ є оцінками максимальної вірогідності. Якщо гіпотезу H_0 відкидають, то це означає, що m факторів не достатньо для опису моделей, а саме, ми маємо включити більше факторів.

В залежності від потреби можна проводити факторний аналіз, використовуючи коваріаційну або кореляційну матриці. Якщо дисперсії дуже розрізняються, має сенс розглядати матрицю кореляцій R замість S . Те, що дисперсія однієї з компонент значно більша за інші означає, що ця компонента домінує. З іншого боку, якщо фактори некорельовані, їх можна обрати у якості головних компонент.

Головні компоненти отримані за допомоги повороту осей. Якщо після повороту результат не має прозорої інтерпретації, можна здійснити ще один поворот, шукаючи підпростір певної розмірності, щоб інтерпретувати фактори. Тобто, при проектуванні на цей підпростір більшість коефіцієнтів перед факторами має дорівнювати 0.

13.4 Приклади

За допомогою факторного аналізу проаналізуємо дані з Прикладу 6.1.7 з [R02] (див. Розділ 7, та не забуваємо завантажити бібліотеки **car**, **ggplot2** та **psych**. Див. також <https://rpubs.com/aaronsc32/factor-analysis-introduction>). Цей приклад у нас вже зустрічався в розділі 7; мова йде про вимірювання параметрів прищеп дерев. Обчислимо коваріаційну матрицю та її власні числа.

```
S <- cov(root[, 2:5])
S.eigen <- eigen(S)
S.eigen
```

```
eigen() decomposition
$values
```

```
[1] 0.495986813 0.162680761 0.006924035 0.001565068
```

```
$vectors
[,1]      [,2]      [,3]      [,4]
[1,] -0.1011191  0.09661363 -0.21551730  0.9664332
[2,] -0.7516463  0.64386366  0.06099466 -0.1294103
[3,] -0.5600279 -0.62651631 -0.52992316 -0.1141384
[4,] -0.3334239 -0.42846553  0.81793239  0.1903481
```

Останні 2 власних числа близькі до 0, тому досить обрати кількість факторів $m = 2$. Можна також зобразити власні числа графічно:

```
plot(S.eigen$values, xlab = 'Eigenvalue Number', ylab =
      'Eigenvalue Size', main = 'Eigenvalues Size',
      type = 'b', xaxt = 'n')
axis(1, at = seq(1, 4, by = 1))
```

Тут `type = b` означає, що ми малюємо і лінії і відповідні точки; `xaxt = n` означає, що ми самі задаємо, що саме буде відображатися по осі OX, а саме, ми обираємо `axis(1, at = seq(1, 4, by = 1))`, тобто кількість власних чисел.

Запишімо матрицю коваріації S у вигляді $S = CD^{1/2}D^{1/2}C'$, та знайдемо C та D . Для того, щоб зобразити матрицю C , візьмемо перші 2 власних вектори. Для того, щоб отримати матрицю D , ми спочатку створюємо порожню матрицю (але відповідного розміру!), а потім додаємо в неї власні числа.

```
C <- as.matrix(S.eigen$vectors[, 1:2])
D <- matrix(0, dim(C)[2], dim(C)[2])
diag(D) <- S.eigen$values[1:2]
```

Запишемо тепер $\hat{\Lambda} = CD^{1/2}$:

```
S.loadings <- C %*% sqrt(D)
S.loadings
```

```
[,1]      [,2]
[1,] -0.07121445  0.03896785
[2,] -0.52935694  0.25969406
[3,] -0.39440707 -0.25269723
[4,] -0.23481824 -0.17281602
```

Зауважимо, що отримані власні вектори – ненормовані.

Все, що ми проробили вище, можна зробити за допомоги вбудованої функції `prcomp`:

```
root.pca <- prcomp(root[, 2:5])$rotation[, 1:2]
root.pca
```

```
PC1      PC2
V2 -0.1011191 0.09661363
```

```
V3 -0.7516463 0.64386366
V4 -0.5600279 -0.62651631
V5 -0.3334239 -0.42846553
```

Отримані власні вектори – нормовані.

Операція `$rotation` повертає значення власних векторів матриці S (порівняйте з застосуванням `S.eigen$vectors[,1:2]`). Операція `$sdev` повертає стандартне відхилення компонент:

```
root.pca2 <- prcomp(root[,2:5])$sdev
round(root.pca2, 3)
```

```
0.704 0.403 0.083 0.04
```

Обчислимо спільні дисперсії $\hat{h}_i^2 = \sum_{j=1}^m \hat{\lambda}_{ij}^2$:

```
S.h2 <- rowSums(S.loadings^2)
round(S.h2, 3)
```

```
0.007 0.348 0.219 0.085
```

Тепер знайдемо залишкові дисперсії $\psi_i = s_{ii} - \hat{h}_i^2$.

```
S.u2 <- diag(S) - S.h2
S.u2, 3
```

V1	V2	V3	V4
1.783368e-03	5.197004e-05	1.964786e-03	4.688978e-03

Знайдемо пропорції власних чисел:

```
prop.loadings <- colSums(S.loadings^2)
prop.var <- cbind(prop.loadings[1] / sum(S.eigen$values),
                    prop.loadings[2] / sum(S.eigen$values))
round(prop.var, 3)
```

```
0.743 0.244
```

Тобто, перше власне значення має внесок 0.743 в загальну суму власних значень, а друге 0.244 (з точністю до округлення). Зауважимо, що сума цих двох власних чисел (`sum(prop.var)`) менша за 1 (`sum(prop.var)`).

Якщо виключити інші власні числа, то отримаємо наступні внески:

```
prop.exp <- cbind(prop.loadings[1] / sum(prop.loadings),
                    prop.loadings[2] / sum(prop.loadings))
round(prop.exp, 3)
```

```
[,1]      [,2]
[1,] 0.753    0.247
```

З іншого боку, можна застосувати вбудовану функцію fa() з пакету **psych** (але є проблема в jupyter: пакет psych на момент написання цього курсу не встановлюється напряму, тому треба використовувати RStudio або Colab). Для цього треба встановити пакет **psych** та завантажити бібліотеку.

Функція principal() виконує факторний аналіз за допомоги методу головних компонент. Поки що ми не використовуємо ротацію факторів, тому поставили rotate = 'none'. Зауважимо, що для аналізу використовується матриці кореляцій, а не матриця коваріацій (див. також [Cl]).

```
root.fa.covar <- principal(root[,2:5], nfactors = 2, rotate = 'none',
                                covar = TRUE)
root.fa.covar
```

```
Principal Components Analysis
Call: principal(r = root[, 2:5], nfactors = 2, rotate = "none",
covar = TRUE)
Standardized loadings (pattern matrix) based upon correlation matrix
          PC1    PC2    h2   u2   com
V1     0.79  0.57  0.95  0.051  1.8
V2     0.85  0.47  0.94  0.061  1.6
V3     0.87 -0.45  0.97  0.027  1.5
V4     0.82 -0.55  0.98  0.022  1.7

          PC1    PC2
SS loadings       2.78 1.05
Proportion Var    0.70 0.26
Cumulative Var   0.70 0.96
Proportion Explained 0.73 0.27
Cumulative Proportion 0.73 1.00

Mean item complexity =  1.7
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is  0.03
with the empirical chi square  0.39 with prob < NA

Fit based upon off diagonal values = 1
```

Перевіримо, що саме ці результати ми отримаємо, якщо використаємо кореляційну матрицю у попередніх обчисленнях. Зауважимо, що вектори, отримані за допомогою fa() не є нормованими!

```
R<- cor(root[,2:5])
R.eigen <- eigen(R)
```

```
R.eigen
plot(R.eigen$values, xlab = 'Eigenvalue Number', ylab =
      'Eigenvalue Size', main = 'Eigenvalues Size',
      type = 'b', xaxt = 'n') axis(1, at = seq(1, 4, by = 1))
```

```
eigen() decomposition
$values
[1] 2.78462702 1.05412174 0.11733950 0.04391174
```

```
$vectors
[,1]      [,2]      [,3]      [,4]
[1,] -0.4713465  0.5600120  0.6431731  0.2248274
[2,] -0.5089667  0.4544775 -0.7142114 -0.1559013
[3,] -0.5243109 -0.4431448  0.2413716 -0.6859012
[4,] -0.4938456 -0.5324091 -0.1340527  0.6743048
```

Можна перевірити, що отримані власні вектори – нормовані. Значення R.eigen\$values[1:2] дорівнюють, відповідно, 2.784627 та 1.054122 (порівняйте з рядком SS loadings вище).

Зобразимо R у вигляді $R = CR(DR)^{1/2}(DR)^{1/2}CR'$ та знайдемо CR та DR .

```
CR <- as.matrix(R.eigen$vectors[,1:2])
DR <- matrix(0, dim(CR)[2], dim(CR)[2])
diag(DR) <- R.eigen$values[1:2]
R.loadings <- CR %*% sqrt(DR)
R.loadings
```

```
[,1]      [,2]
[1,] 0.7865453  0.5749668
[2,] 0.8493229  0.4666140
[3,] 0.8749282 -0.4549787
[4,] 0.8240901 -0.5466267
```

Тобто, R.loadings – це як раз вектори $PC1$, $PC2$, які знайдено функцією principal().

Обчислимо тепер $\hat{h}_i^2 = \sum_{j=1}^m \hat{\lambda}_{ij}^2$ та $\psi_i = r_{ii} - \hat{h}_i^2$

```
R.h2 <- rowSums(R.loadings^2)
R.h2
```

```
0.95 0.94 0.97 0.98
```

(порівняйте з колонкою h2 вище).

Далі, залишкова дисперсія:

```
R.u2 <- diag(R) - R.h2
round(R.u2, 3)
```

V1	V2	V3	V4
0.051	0.061	0.027	0.022

(порівняйте з колонкою u2). Складність фактору, або індекс складності Хоффмана (Hoffman's index of complexity, колонка com) можна обчислити наступним чином:

$$com = \frac{(\sum_{j=1}^n \lambda_{ij}^2)^2}{\sum_{j=1}^n \lambda_{ij}^4}.$$

```
comR <- rowSums(R.loadings^2)^2 / rowSums(R.loadings^4)
round(comR, 1)
```

1.8 1.6 1.5 1.7

Чим ближчі до 1 значення com, тим краще поясненою є наша модель, тобто квадрат суми квадратів не суттєво відрізняється від суми навантажень в четвертому степені (див. посилання в [R02] на обґрунтування). Знайдемо пропорцію власних чисел:

```
prop.loadingsR <- colSums(R.loadings^2)
prop.varR <- cbind(prop.loadingsR[1] / sum(R.eigen$values),
prop.loadingsR[2] / sum(R.eigen$values))
prop.varR
```

Отримаємо, відповідно, рядок Proportion Var. Розглянемо лише ці два перші власні числа:

```
prop.expr <- cbind(prop.loadingsR[1] / sum(prop.loadingsR),
prop.loadingsR[2] / sum(prop.loadingsR))
prop.expr
```

Отримаємо, відповідно, рядок Proportion explained.

Перейдемо тепер до вбудованих функцій. Функція varimax() дозволяє знайти найкращу комбінацію факторів (тобто здійснити "поворот" осей) для того, щоб знайти найкращі навантаження. Найкращим розв'язком був би такий, в якому складність була б близькою до 1, що в свою чергу означає, що одна змінна найкраще пояснюється одним фактором.

А саме (метод Кайзера), шукаємо таку ортогональну матрицю T , яка б максимізувала таку різницю моментів:

$$\frac{1}{p} \sum_{j=1}^k \sum_{i=1}^p (\Lambda T)_{ij}^4 - \sum_{j=1}^k \left(\frac{1}{p} \sum_{i=1}^p (\Lambda T)_{ij}^2 \right)^2 \implies \max.$$

Фактично, ми при цьому змінюємо базис та знаходимо нову матрицю навантажень Λ . Елементи цієї матриці називаються loadings.

В результаті ми отримали б $\Lambda^* = \Lambda T$, де матриця T є ортогональною та такою, що максимізує дисперсії навантажень в кожному стовбчику матриці Λ^* . Зауважимо, що при цьому середні дисперсії не змінюються: ортогональне перетворення не змінює власні числа.

```

factors<-R.loadings
varimax(factors)
factors.v <- varimax(factors)$loadings
round(factors.v, 2)

```

```

Loadings:
 [,1] [,2]
[1,] 0.16 0.96
[2,] 0.28 0.93
[3,] 0.94 0.29
[4,] 0.97 0.19

 [,1] [,2]
SS loadings 1.928 1.907
Proportion Var 0.482 0.477
Cumulative Var 0.482 0.959

```

```

h2.v <- rowSums(factors.v^2)
h2.v

```

0.9492403 0.9390781 0.9725050 0.9779253

(що те саме, що R.h2) Перевіримо, що середні дисперсії не змінилися:

```

u2.v <- 1 - h2.v
u2.v

```

0.05075965 0.06092192 0.02749496 0.02207470

(що те саме, що R.u2)

Порахуємо складність:

```

com.v <- rowSums(factors.v^2)^2 / rowSums(factors.v^4)
com.v

```

1.054355 1.179631 1.185165 1.074226

Весь цей аналіз можна проробити за допомоги наступної вбудованої функції:

```

root.fa2 <- principal(root[,2:5], nfactors = 2, rotate = 'varimax')
root.fa2

```

```

Principal Components Analysis
Call: principal(r = root[, 2:5], nfactors = 2, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
  RC1   RC2   h2    u2   com
V1  0.16  0.96  0.95  0.051  1.1
V2  0.28  0.93  0.94  0.061  1.2
V3  0.94  0.29  0.97  0.027  1.2
V4  0.97  0.19  0.98  0.022  1.1

  RC1   RC2
SS loadings      1.94  1.90
Proportion Var   0.48  0.48
Cumulative Var   0.48  0.96
Proportion Explained  0.50  0.50
Cumulative Proportion 0.50  1.00

```

Mean item complexity = 1.1
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is 0.03
with the empirical chi square 0.39 with prob < NA

Fit based upon off diagonal values = 1

Як ми бачимо, при цьому складність (mean item complexity) вже дорівнює 1.1, тобто цей метод працює краще, ніж попередній (без повороту осей).

Розглянемо аналіз методом головних факторів.

Замінимо діагональні елементи на оцінки (13.11). Зауважимо, що у випадку, коли ми розглядаємо матрицю кореляцій, а не матрицю коваріації, $s_{ii} = 1$ в (13.11), тобто

$$\hat{h}_i^2 = 1 - \frac{1}{r_{ii}}.$$

Потім ми замінимо діагональні елементи R на R.smc (R.smc - це squared multiple correlation):

```

R.smc <- (1 - 1 / diag(solve(R)))
print(round(R.smc, 2))
diag(R) <- R.smc
round(R, 2)

```

V2	V3	V4	V5
0.80	0.81	0.91	0.91

V2	V3	V4	V5	
V2	0.80	0.88	0.44	0.33
V3	0.88	0.81	0.52	0.45

```
V4 0.44 0.52 0.91 0.95
V5 0.33 0.45 0.95 0.91
```

Тепер знайдемо власні числа та власні вектори матриці $R - \hat{\Psi}$.

```
r.eigen <- eigen(R)
print(round(r.eigen$values, 2))
```

```
[1] 2.65 0.91 -0.03 -0.09
```

Матриця R вже не є додатно визначеною, оскільки ми вже замінили елементи на діагоналі їх оцінками. Отже, декілька власних чисел можуть бути від'ємними. Оскільки від'ємні власні числа не використовуються для оцінки λ , оберемо $m = 2$.

```
r.lambda <- as.matrix(r.eigen$vectors[, 1:2]) %*%
diag(sqrt(r.eigen$values[1:2]))
r.lambda
```

Обчислимо середні дисперсії (communalities, specific variances) та складність навантажень:

```
r.h2 <- rowSums(r.lambda^2)
r.u2 <- 1 - r.h2
com <- rowSums(r.lambda^2)^2 / rowSums(r.lambda^4)
```

Зберемо результати в наступну data.frame.

```
cor.pa <- data.frame(cbind(round(r.lambda, 2),
round(r.h2, 2), round(r.u2, 3), round(com, 1)))
colnames(cor.pa) <- c('PA1', 'PA2', 'h2', 'u2', 'com')
cor.pa
```

	PA1	PA2	h2	u2	com
1	-0.74	0.54	0.84	0.158	1.8
2	-0.80	0.45	0.85	0.150	1.6
3	-0.88	-0.41	0.93	0.067	1.4
4	-0.83	-0.50	0.93	0.072	1.6

Застосуємо тепер вбудовану функцію fa(). Функція fa() може застосовувати ітеративно метод головних факторів до тих пір, поки ми не отримаємо задовільний результат. Параметри fm = 'pa' означає, що ми використовуємо метод головних факторів. SMC = TRUE означає, що ми використаємо квадратичну множинну кореляцію як першу апроксимацію h^2 (інакше $h^2 = 1$). Насправді, цей параметр використовується за замовчуванням, тобто його можна опустити; max.iter = 1 означає, що ми робимо лише одну ітерацію.

```

root.cor.fa2 <- fa(root[,2:5], nfactors = 2, rotate = 'none',
                      fm = 'pa', SMC= TRUE, max.iter = 1)
root.cor.fa2

```

Factor Analysis using method = pa
 Call:
 fa(r = root[, 2:5], nfactors = 2, rotate = "none", max.iter = 1, fm = "pa")
 Standardized loadings (pattern matrix) based upon correlation matrix

	PA1	PA2	h2	u2	com
V1	0.74	0.54	0.84	0.158	1.8
V2	0.80	0.45	0.85	0.150	1.6
V3	0.88	-0.41	0.93	0.067	1.4
V4	0.83	-0.50	0.93	0.072	1.6

PA1 PA2

	SS loadings	Proportion Var	Cumulative Var	Proportion Explained	Cumulative Proportion
	2.65	0.66	0.66	0.74	0.74
	0.91	0.23	0.89	0.26	1.00

Mean item complexity = 1.6
 Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 6 and the objective function was 4.19 with Chi Square of 187.92
 The degrees of freedom for the model are -1 and the objective function was 0.17

The root mean square of the residuals (RMSR) is 0.02
 The df corrected root mean square of the residuals is NA

The harmonic number of observations is 48
 with the empirical chi square 0.24 with prob < NA
 The total number of observations was 48
 with Likelihood Chi Square = 7.26 with prob < NA

Tucker Lewis Index of factoring reliability = 1.281
 Fit based upon off diagonal values = 1
 Measures of factor score adequacy

	PA1	PA2
Correlation of (regression) scores with factors	0.98	0.93
Multiple R square of scores with factors	0.95	0.86
Minimum correlation of possible factor scores	0.90	0.72

Порівняємо результати, якщо встановити поворот осей (тобто параметр rotate = 'varimax')

```
root.cor.fa <- fa(root[,2:5], nfactors = 2, rotate = 'varimax',
    fm = 'pa', max.iter = 1)
root.cor.fa
```

Factor Analysis using method = pa
Call:
fa(r = root[, 2:5], nfactors = 2, rotate = "varimax", max.iter = 1, fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix

	PA1	PA2	h2	u2	com
V1	0.19	0.90	0.84	0.158	1.1
V2	0.30	0.87	0.85	0.150	1.2
V3	0.92	0.29	0.93	0.067	1.2
V4	0.95	0.18	0.93	0.072	1.1

SS loadings PA1 PA2
Proportion Var 1.87 1.68
Cumulative Var 0.47 0.42
Proportion Explained 0.47 0.89
Cumulative Proportion 0.53 1.00

Mean item complexity = 1.1
Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 6 and the objective function was 4.19 with Chi Square of 187.92
The degrees of freedom for the model are -1 and the objective function was 0.17

The root mean square of the residuals (RMSR) is 0.02
The df corrected root mean square of the residuals is NA

The harmonic number of observations is 48 with the empirical chi square 0.24 with prob < NA
The total number of observations was 48 with Likelihood Chi Square = 7.26 with prob < NA

Tucker Lewis Index of factoring reliability = 1.281
Fit based upon off diagonal values = 1
Measures of factor score adequacy

	PA1	PA2
Correlation of (regression) scores with factors	0.97	0.94
Multiple R square of scores with factors	0.94	0.87
Minimum correlation of possible factor scores	0.88	0.75

З наведеного вище можна зробити висновок, що модель можна прийняти.

14 Додаток

Ядерні оцінки щільності

Література: [ELLS11].

Нехай ξ має щільність розподілу $f(y)$, та нехай $N(y_0)/n$ – кількість спостережень, що попали в інтервал $(y_0 - h, y_0 + h)$. Оскільки

$$\int_{y_0-h}^{y_0+h} f(z)dz = \mathbb{P}(y_0 - h < \xi \leq y_0 + h) \asymp \frac{N(y_0)}{n},$$

то

$$\hat{f}(y_0) \asymp \frac{N(y_0)}{2hn}.$$

Наша задача – оцінити $N(y)$ за допомогою ядра $K(y)$, тому такі оцінки щільності називаються *ядерними*.

Нехай

$$K(u) = \begin{cases} \frac{1}{2}, & |u| \leq 1; \\ 0, & |u| > 1. \end{cases}$$

Тоді

$$N(y_0) = 2 \sum_{i=1}^n K\left(\frac{y_0 - y_i}{h}\right)$$

підраховує кількість точок, які попали в інтервал $y_0 - h \leq y_i \leq y_0 + h$ (якщо $|y_i - y_0| \leq h$, то це як раз і означає, що точка y_i попала в інтервал $y_0 - h \leq y_i \leq y_0 + h$). Тоді

$$\hat{f}(y_0) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{y_0 - y_i}{h}\right).$$

Можна обирати і інші ядра. Наведемо кілька прикладів.

- Ядро Єпанечнікова:

$$K(u) = \frac{3}{4}(1 - x^2)\mathbf{1}_{|x| \leq 1};$$

- Трикутне ядро:

$$K(u) = (1 - |u|)\mathbf{1}_{|u| \leq 1}.$$

- Гаусівське ядро:

$$K(u) = \frac{e^{-u^2/2}}{\sqrt{2\pi}}.$$

- І ще одне ядро:

$$K(u) = \frac{1}{\pi} \frac{\sin^2 u}{u^2}.$$

У багатовимірному випадку оцінка щільності виглядає наступним чином:

$$\hat{f}(y_0) = \frac{1}{nh_1 h_2 \dots h_p} \sum_{i=1}^n K\left(\frac{Y_{01} - Y_{i1}}{h_1}, \frac{Y_{02} - Y_{i2}}{h_1}, \dots, \frac{Y_{0p} - Y_{ip}}{h_1}\right).$$

Розглянемо кілька прикладів наближення щільності. Розглянемо дані, які вже використовувались у розділах 11, 12. Нам знадобиться пакет **ggplot2**.

```
Foot<-read.table('T8_3_FOOTBALL.DAT',
                  col.names = c('Group', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7'))
```

Ми використаємо функцію density. Параметром може бути одна з координат вектору $kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine", "optcosine")$. Наприклад,

```
den1<-density(x, bw = "nrd0", adjust = 1, kernel = "rectangular",
                weights = NULL)
plot(den1)
```

```
den2<-density(x, bw = "nrd0", adjust = 1, kernel ="rectangular",
                weights = NULL)
plot(den2)
```

Література

- [Ak74] Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19 (6)** (1974), 716–723.
- [AD52] Anderson, T.W., Darling, D.A., Asymptotic theory of certain 'goodness of fit' criteria based on stochastic processes. *Annals of Mathematical Statistics*, **23** (1952), 193–212.
- [AD54] Anderson T.W., Darling D.A. A test of goodness of fit. *J. Amer. Statist. Assoc.*, **29** (1954), 765–769.
- [Ba51] Bartlett, M. S. The effect of standardization on a Chi-square approximation in factor analysis. *Biometrika*, **38** (1951), 337–344.
- [BF74] Levene, H. Robust tests for equality of variances. In Ingram Olkin; Harold Hotelling; et al. (eds.). *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press (1960), 278–292.
- [EH21] Ebner, B., Henze, N. Test for multivariate normality – a critical review with emphasis on weighted L_2 -statistics. ArXiv <https://arxiv.org/pdf/2004.07332.pdf>
- [EKSX96] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996. http://www.sthda.com/english/wiki/wiki.php?id_contents=7940
- [Ev93] Everitt, B.S. *Cluster Analysis*, 3rd ed., London: Edward Arnold, 1993.
- [ELLS11] Everitt, B.S., Landau, S., Leese, M., Stahl, D. *Cluster analysis*. Wiley, 2011.
- [FFM09] Fox, J., Friendly, M., Monette, G. Visualizing hypothesis tests in multivariate linear models: the heplots package for R. *Comput. Stat.*, **24** (2009), 233–246.
- [HZ90] Henze, N., Zirkler, B. A class of invariant consistent tests for multivariate normality. *Comm. in Statistics: Theory and Methods*, **19(10)** (1990), 3595–3617.
- [Ho31] Hotelling, H. (1931). The generalization of Student's ratio. *Annals of Mathematical Statistics* **2 (3)** (1931), 360–378.
- [K07] Карташов, М.В. *Ймовірність, процеси, статистика*. Видавничо-поліграфічний центр "Київський університет". Київ, 2007.
- [Le60] Levene, H. In: Ingram Olkin; Harold Hotelling; et al. (eds.) *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. I. Olkin et al. eds., Stanford University Press (1960) 278–292.
- [Li72] Ling, R.F. On the theory and construction of k -clusters. *The Computer Journal*, **15 (4)** (1972), 326–332.
- [M70] Mardia, K.V. Measures of Multivariate Skewness and Kurtosis with Applications. *Biometrika*, **57** (1970), 519–530.

- [M74] Mardia, K.V. Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies. *Sankhya, Ser. B.* **36**(2) (1974), 115–128.
- [M80] Mardia, K.V. Tests of univariate and multivariate normality. In: P.R. Krishnaiah (ed.), *Handbook of Statistics*, 1, North Holland, 1980, 279–320.
- [Ma40] Mauchly, J.W. Significance Test for Sphericity of a Normal n -Variate Distribution. *The Annals of Math. Stat.* **11** (2) (1940), 204–209.
- [Ma07] Майборода, Р.Є. Регресія: лінійні моделі. ВПЦ Київський університет. 2007.
- [Ma02] Malinovski, E. Factor Analysis. 3-rd ed. Wiley, New York 2002.
- [PP12] Petersen, K.B., Pedersen, M.S. The Matrix Cookbook, 2012. <http://matrixcookbook.com>
- [RW11] Razali, N.M., Wah, Y.B. Power comparisons of Shapiro-Silk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics*, **2** (1) (2011), 21–33.
- [R98] Rencher, A.C. *Multivariate statistical inference and applications*. Wiley, NY, 1998.
- [R02] Rencher, A.C. *Methods of Multivariate Analysis*. Second ed., Wiley, NY, 2002.
- [RS08] Rencher, A.C., Schaalje, G.B. *Linear Models in Statistics*. Wiley, NY, 2002.
- [Ro92] Royston, P. Approximating the Shapiro-Wilk W-Test for Non-Normality. *Statistics and Computing*, **2**(3) (1992), 117–19.
- [Ro95] Royston, P. Remark AS R94: A Remark on Algorithm AS 181: The W-Test for Normality. *Journal of the Royal Stat. Soc., Series C (Applied Statistics)*, **44** (4) (1995), 547–51.
- [SW65] Shapiro, S.S., Wilk, M.B. An Analysis of Variance Test for Normality (Complete Samples) *Biometrika*, **52** (3/4) (1965), 591–611.
- [SW03] Svantesson, Th., Wallace, J.W. Tests for assessing multivariate normality and the covariance structure of MIMO data. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), Hong Kong, Apr. 6-10, 2003.
- [VAG09] Villasenor Alva, J.A. Estrada, E.G. A Generalization of Shapiro–Wilk’s Test for Multivariate Normality. *Comm. in Stat. - Theory and Methods*. **38**(11) (2009), 1870–1883.
- [YA07] Yazici, B., Asma, S. A comparison of various tests of normality *Journal of Statistical Computation and Simulation*, **77**(2) (2007), 175–183.
- [ZS14] Zhoua, M., Shaob, Y. A Powerful Test for Multivariate Normality. *J Appl Stat.* **41**(2) (2014), 351–363.
- [Ar] Arquez, M. On-line ресурс. Посилання на сторінку в RPubs: <https://rpubs.com/arquez9512/>
- [Be] Bevans, R. ANOVA in R. A Complete Step-by-Step Guide with Examples. *Scribbr*. Retrieved June 13 (2023), from <https://www.scribbr.com/statistics/anova-in-r/>

- [Br] Brown, N.B. Expectation-maximization. On-line Lecture notes. https://rstudio-pubs-static.s3.amazonaws.com/154174_78c021bc71ab42f8add0b2966938a3b8.html
- [Cl] Clark, M. Factor Analysis with the psych package. Online pecypc. <https://m-clark.github.io/posts/2020-04-10-psych-explained/>
- [Fr] Friendly, M. Penguins: MANOVA and HE plots. <https://rpubs.com/friendly/penguin-manova>
- [IHA] McInnes, L., Healy, J., Astels, S. Comparing Python Clustering Algorithms. https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html#hdbscan
- [Mu] Mulla, R. A Gentle Introduction to Pandas 2022 - Python Tutorial on Kaggle. https://www.youtube.com/watch?v=_Eb0utIRdkw
- [Ri] Rickert, J. Generating and Visualizing Multivariate Data with R. https://blog.revolutionanalytics.com/2016/02/multivariate_data_with_r.html
- [Sh] Shegel, A. Online pecypc. Посилання на сторінку в RPubs: <https://rpubs.com/aaronsc32>
- [Ta] Taboga, M. Schur complement. Lectures on matrix algebra. <https://www.statlect.com/matrix-algebra/>
- [biotools] da Silva, A. R. Tools for Biometry and Applied Statistics in Agricultural Science. Package 'biotools', Version 4.2. <https://cran.r-project.org/web/packages/biotools/biotools.pdf>
- [car] Fox, J., et.al. Companion to Applied Regression. Package 'car', Version 3.1-2. <https://cran.r-project.org/web/packages/car/car.pdf>
- [caret] Kuhn, M. Classification and Regression Training. Package 'caret', Version 6.0-94. <https://cran.r-project.org/web/packages/caret/caret.pdf>
- [caTools] Tuszynski, J. Tools: Moving Window Statistics, GIF, Base64, ROC AUC, etc. Package 'caTools', Version 1.18.2. <https://cran.r-project.org/web/packages/caTools/caTools.pdf>
- [dbSCAN] Hahsler, M., Piekenbrock, M., Arya, S., Mount, D. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Related Algorithms. Package 'dbSCAN', Version 1.1-12. <https://cran.r-project.org/web/packages/dbSCAN/dbSCAN.pdf>
- [dplyr] A Grammar of Data Manipulation. Version 1.1.2. <https://cran.r-project.org/web/packages/dplyr/dplyr.pdf>
- [e1071] Meyer, D., et. al. Misc Functions of the Department of Statistics, Probability Theory Group. Package 'e1071', Version 1.7-13. <https://cran.r-project.org/web/packages/e1071/e1071.pdf>

- [ellipse] Murdoch, D., Chow, E.D. Functions for Drawing Ellipses and Ellipse-Like Confidence Regions. Package ‘ellipse’, Version 0.4.5. <https://cran.r-project.org/web/packages/ellipse/ellipse.pdf>
- [factoextra] Kassambara, A., Mundt, F. Extract and Visualize the Results of Multivariate Data Analyses. Package ‘factoextra’, Version 1.0.7. <https://cran.r-project.org/web/packages/factoextra/factoextra.pdf>
- [fpc] Hennig, Ch. Flexible Procedures for Clustering. Package ‘fpc’, Version 2.2-12. <https://cran.r-project.org/web/packages/fpc/fpc.pdf>
- [heplots] Friendly, M. Visualizing Hypothesis Tests in Multivariate Linear Models. Package ‘Heplots’, Version 1.4-2. <https://cloud.r-project.org/web/packages/heplots/heplots.pdf>
- [Hotelling] Curran, J., Hersh, T. Hotelling’s T^2 Test and Variants. Package ‘Hotelling’, Version 1.0-8. <https://cran.r-project.org/web/packages/Hotelling/Hotelling.pdf>
- [ICSNP] Nordhausen, K., Sirkia, S., Oja, H., Tyler, D.E. Tools for Multivariate Nonparametrics. Package ‘ICSNP’, Version 1.1-1. <https://cran.r-project.org/web/packages/ICSNP/ICSNP.pdf>
- [jpeg] Urbanek, S. Read and write JPEG images. Package ‘jpeg’, Version 0.1-10. jpeg
- [ggplot2] Wickham H., et. al. Create Elegant Data Visualisations Using the Grammar of Graphics. Package ‘ggplot2’, Version 3.4.2. <https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>
- [ggfortify] Horikoshi, M., et.al. Data Visualization Tools for Statistical Analysis Results. Package ‘ggfortify’, Version 0.4.16. <https://cran.r-project.org/web/packages/ggfortify/ggfortify.pdf>
- [MASS] Ripley, B., Venables, B., Bates, D.M., Hornik, K., Gebhardt, A., Firth, D. Support Functions and Datasets for Venables and Ripley’s MASS. Package ‘MASS’. Version 7.3-60. <https://cran.r-project.org/web/packages/MASS/MASS.pdf>
- [mclust] Fraley, Ch., et al. Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation. Package ‘mclust’, Version 6.0.0. <https://cran.r-project.org/web/packages/mclust/mclust.pdf>
- [MVN] Korkmaz, S., Goksuluk, D., Zararsiz, G. Multivariate Normality Tests. Version 5.9. <https://cran.r-project.org/web/packages/MVN/MVN.pdf>
- [mvnTest] Pya, N., Voinov, V., Makarov, R., Voinov, Y. Goodness of Fit Tests for Multivariate Normality. Package ‘mvnTest’. Version 1.1-0. <https://cran.r-project.org/web/packages/mvnTest/mvnTest.pdf>
- [mvtnorm] Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., Bornkamp, B., Maechler, M., Hothorn, T. Multivariate Normal and t Distributions. Package ‘mvtnorm’. Version 1.2-2. <https://cran.r-project.org/web/packages/mvtnorm/mvtnorm.pdf>

- [psych] Revelle, W. Procedures for Psychological, Psychometric, and Personality Research. Packet 'psych', Version 2.3.3. <https://cran.r-project.org/web/packages/psych/psych.pdf>
- [repr] Angerer, Ph. Serializable Representations. Package 'repr', Version 1.1.7. <https://cran.r-project.org/web/packages/repr/repr.pdf>
- [stats] R Core Team and contributors worldwide. The R Stats Package. Package 'stats', Version: 4.4.0. <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html>