

Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до самостійних робіт з дисципліни
**«ЕМПІРИЧНІ МЕТОДИ В ПРОЕКТУВАННІ
ЕЛЕКТРОННИХ ЗАСОБІВ»**
для студентів напряму підготовки
122 «Комп'ютерні науки та інформаційні технології»
(всіх форм навчання)

2017

Методичні вказівки до самостійних робіт з дисципліни «Емпіричні методи в проектуванні електронних засобів» для студентів напряму підготовки 122 «Комп'ютерні науки та інформаційні технології» (всіх форм навчання) / А. В. Притула, Н.О. Миронова. – Запоріжжя: ЗНТУ, 2017. – 36 с.

Автори: А. В. Притула, к.т.н., професор
Н. О. Миронова, к.т.н., доцент

Рецензент: Т.В. Федорончак, к.т.н., доцент

Відповідальний
за випуск: А.В. Притула, к.т.н., професор

Затверджено
на засіданні кафедри
програмних засобів

Протокол №11
від 06 червня 2017

ЗМІСТ

Вступ	4
Самостійна робота №1 Знайомство з середовищем розробки R-Studio	5
Самостійна робота №2 Статистичний аналіз і первинна обробка даних з використанням мови R	10
Самостійна робота №3 Статистична перевірка гіпотез засобами мови R	19
Самостійна робота №4 Кореляційний та регресійний аналіз з використанням мови R	24
Самостійна робота №5 Повний факторний експеримент	30
Самостійна робота №6 Ортогональне центральне композиційне планування	33
Література	36

ВСТУП

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання основ емпіричних методів програмної інженерії.

Відповідно до графіка студенти перед виконанням самостійної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою. Звичайно, в дані методичні вказівки неможливо було внести весь матеріал, необхідний для виконання самостійних робіт. Тому тут містяться основні, базові теоретичні відомості, необхідні для виконання самостійних робіт. Таким чином для виконання самостійної роботи необхідно ознайомитись з конспектом лекцій та проробити весь матеріал, наведений в переліку рекомендованої літературі. При цьому не варто обмежуватись лише наведеним списком.

Для одержання заліку з кожної роботи студент здає викладачу цілком оформлений звіт, а також демонструє на екрані комп'ютера результати виконання самостійної роботи.

Звіт має містити:

- титульний аркуш;
- тему та мету роботи;
- завдання до роботи;
- лаконічний опис теоретичних відомостей;
- результати виконання самостійної роботи;
- змістовний аналіз отриманих результатів та висновки. Звіт виконують на білому папері формату А4 (210 × 297 мм).

Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок або вміщують у канцелярський файл.

САМОСТІЙНА РОБОТА №1

ЗНАЙОМСТВО З СЕРЕДОВИЩЕМ РОЗРОБКИ R-STUDIO

Мета роботи

Ознайомитися з можливостями середовища розробки R-Studio та мови програмування R.

Короткі теоретичні відомості

RStudio є безкоштовним інтегрованим середовищем розробки (IDE) для R (рис.1.1). Цей програмний продукт з відкритим вихідним кодом для мови програмування R, яка призначена для статистичної обробки даних і роботи з графікою, робить роботу з R дуже зручною.

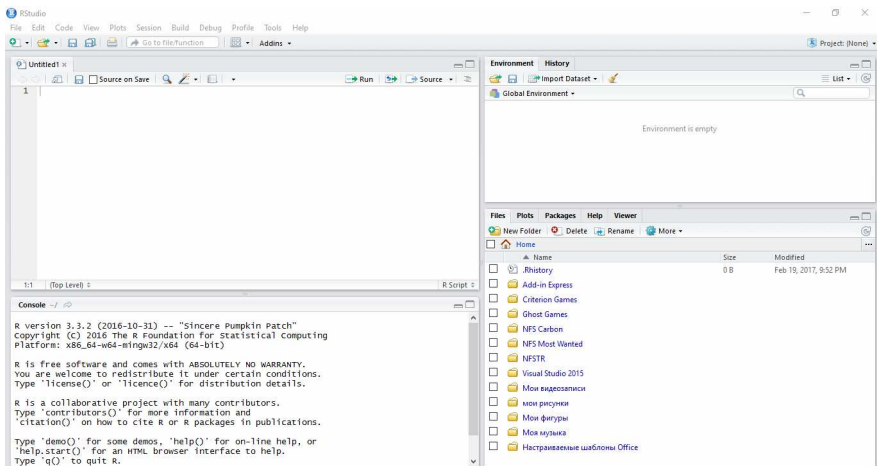


Рисунок 1.1 – Середовище розробки R-Studio

Середовище розробки R-Studio складається з наступних компонентів:

- панель меню;
- панель інструментів;
- консоль;
- запрошення (командний рядок);

- панель, що містить історію і робочий простір;
- панель з графіками.

Консоль RStudio надає цілий ряд опцій, що полегшують роботу з мовою R. Пропонуємо ознайомитися з ними нижче. Наприклад, автоматичне завершення коду: набираючи початок команди середовище пропонує користувачу продовження (рис.1.2).

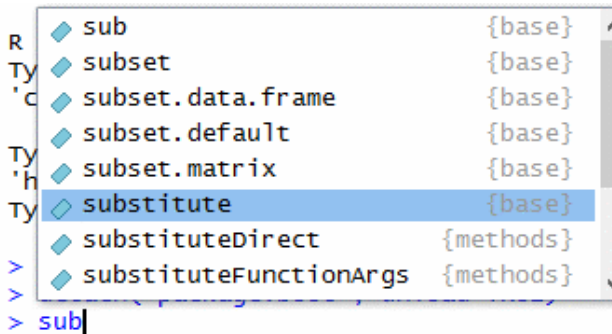


Рисунок 1.2 – Демонстрація автоматичного завершення коду в консолі

Повернення до попередніх команд. За допомогою сполучень клавіш Ctrl + ↑ та Ctrl + ↓ можна переходити до раніше викликаних команд (рис.1.3).

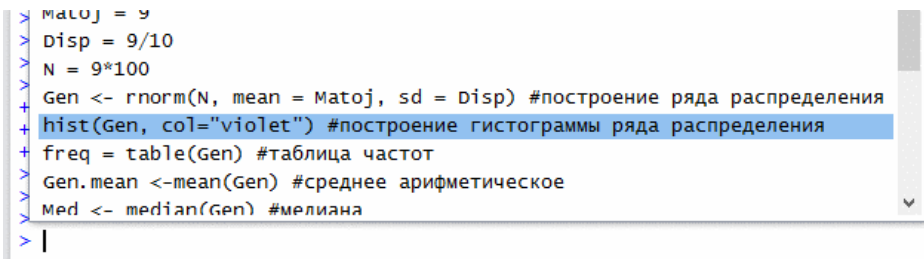


Рисунок 1.3 – Перехід до раніше викликаних команд в консолі

«Гарячі» клавіші. Переглянути гарячі клавіші можливо викликавши HELP – R HELP, або обрати потрібний пункт меню Code, View або інше (рис.1.4).

Move Focus to Source	Ctrl+1
Move Focus to Console	Ctrl+2
Move Focus to Help	Ctrl+3
Show History	Ctrl+4
Show Files	Ctrl+5
Show Plots	Ctrl+6
Show Packages	Ctrl+7
Show Environment	Ctrl+8
Show Viewer	Ctrl+9
Show Connections	Ctrl+F5

Рисунок 1.4 – Список «гарячих» клавіш пункту меню View

Створення нових скриптів. Для створення нових скриптів необхідно обрати File – New File → R Script (рис. 1.5). Якщо проект буде мати декілька R файлів, то спочатку New Project, а потім New File.

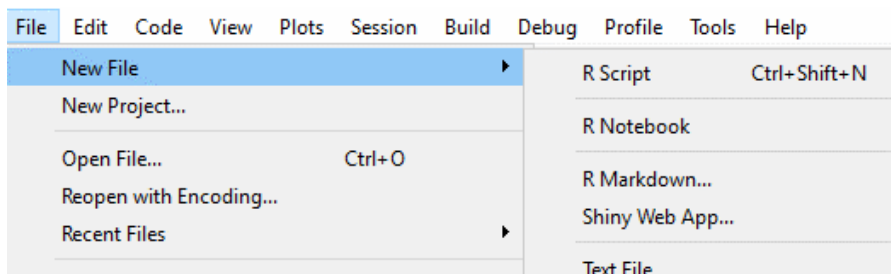


Рисунок 1.5 – Створення нового файлу

Створення функцій в коді. Щоб створити функцію треба виділити частину коду та обрати пункт Extract Function (рис.1.6).

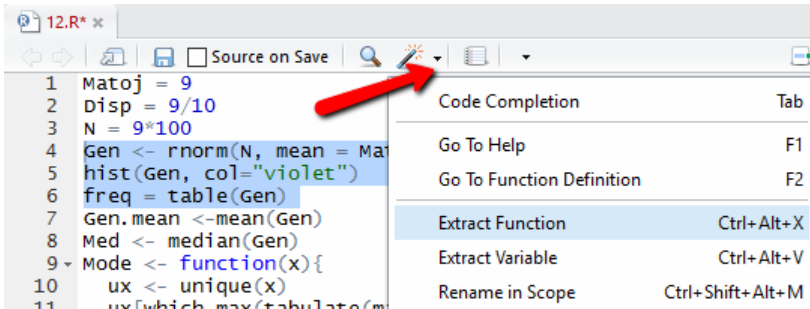


Рисунок 1.6 – Створення функції з виділеного коду

Пошук та заміна частин коду. Щоб знайти та/або замінити частину тексту треба викликати вікно пошуку за допомогою меню Edit -> Find and Replace, або Ctrl + F (рис. 1.7).

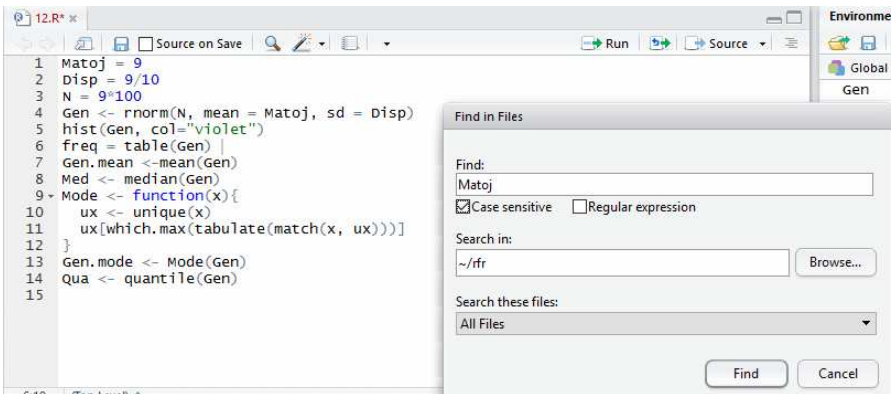


Рисунок 1.7 – Пошук частини коду в тексті

Коментування коду. Для коментування/зняття коментування з фрагменту коду – виділити його на натиснути пункт Comment/Uncomment Lines (рис. 1.8).

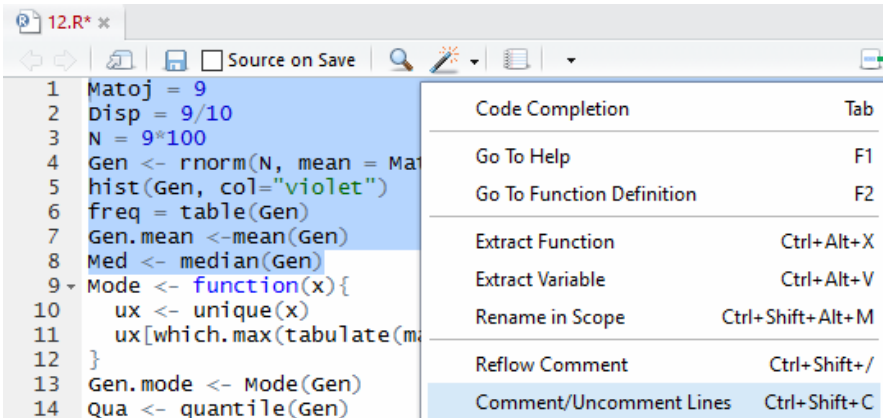


Рисунок 1.8 – Коментування коду

Виконання коду. Щоб виконати код – виділіть потрібну частину та натисніть Run. Для виконання поточного рядку – натисніть Ctrl + Enter. Після цього редактор автоматично перейде на наступний рядок. Для виконання усіх рядків – натисніть Ctrl + Shift + Enter.

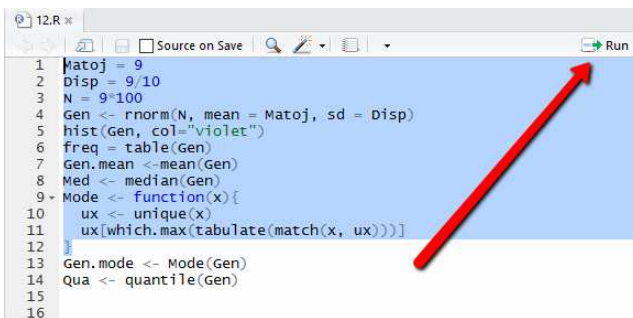


Рисунок 1.9 – Виконання коду

Завдання на самостійну роботу

Скориставшись рекомендованою літературою вивчити можливості середовища розробки R-Studio та мови програмування R.

САМОСТІЙНА РОБОТА №2

СТАТИСТИЧНИЙ АНАЛІЗ І ПЕРВИННА ОБРОБКА ДАНИХ З ВИКОРИСТАННЯМ МОВИ R

Мета роботи

Навчитися використовувати середовище розробки R-Studio та мову програмування R для первинного аналізу даних. Засвоїти основи мови програмування R для статистичних обчислень та аналізу та даних.

Основні функції та команди, необхідні для виконання роботи

В роботі необхідно використати стандартні функції та команди R, які наведені у таблиці 2.1.

Таблиця 2.1 – Перелік стандартних функцій та команд R

Назва функції	Опис
<i>mean()</i>	Функція для знаходження середнього арифметичного, вона приймає параметр ряду розподілу.
<i>sd()</i>	Функція для пошуку стандартного відхилення, вона приймає параметр ряду розподілу.
<i>rnorm()</i>	Функція слугує для випадкової генерації нормально розподілених чисел. Вона має параметри: <i>N</i> – заданий розмір, <i>mean</i> – середнє значення, <i>sd</i> – стандартне відхилення.
<i>hist()</i>	Функція для створення гістограми. Вона може приймати ті ж параметри, що і у функції для створення графіків <i>plot()</i> . Одна з них: <i>col</i> – слугує для задання кольору стовпчиків або інших елементів графіків. Слугує оцінкою щільності відповідного розподілу.
<i>table()</i>	Функція визначає таблицю частот відповідних рівнів.
<i>median()</i>	Функція що повертає медіану, та приймає ті ж параметри, що й <i>mean()</i> .
<i>which.max()</i>	Функція знаходить порядкові номери елементів з максимальним значенням, а якщо елементів декілька, то

	буде повернуто номер першого такого елемента.
<i>unique()</i>	Функція повертає вектор, таблицю, або масив, але з однаковими елементами.
<i>tabulate()</i>	Функція бере ціле значення з вектору та підраховує кількість разів яке кожне ціле знаходиться в ньому.
<i>match()</i>	Функція повертає вектор тієї довжини, що і вектор з елементами у місці пошуку.
<i>quantile()</i>	Функція розраховує квантілі, приймає параметр ряду розподілу.
<i>print()</i>	Функція виводить на екран об'єкт, це функція загального призначення – конкретний результат її роботи буде залежати від класу об'єкта.
<i>cat()</i>	Функція більш розширена, ніж print і теж служить для виведення інформації на консоль.

Перевірка умов в мові R виконується наступним чином:

```
if (умова){
    виконується, якщо умова вірна
} else {
    виконується, якщо умова не вірна
}
```

Циклічні вирази із заданою великою кількістю ітерацій виконуються за допомогою конструкції:

```
for (<змінна> in <вираз-1>)
    <вираз-2>
```

Тут результатом <вираз-1> повинен бути вектор, а <змінна> на кожній ітерації циклу приймає значення чергового елемента цього вектора. кількість ітерацій дорівнює кількості елементів у векторі.

Відповідно, в <вираз-2> може використовуватися <змінна>, яка буде змінною (лічильником) циклу:

```
sum = 0
for (i in 1:20)
    sum = sum + i;
sum
```

[1] 210

Аналогічно працює цикл `while`.

`while (<умова>) <вираз>`

Переривання циклу здійснюється командою `break`. Для переривання поточної ітерації і переходу до наступної служить команда `next`.

Приклад реалізації у R-Studio

Згенерувати стовпець даних на основі наступної інформації $N = 900$, $\mu = 9$, $\sigma^2 = 0.9$, де N – кількість дослідів, μ – математичне очікування, σ^2 – дисперсія. Створити нормальний ряд розподілу на основі цих даних. Побудувати гістограму, таблицю частот, отримати описові статистики ряду розподілу.

```
Matobj = 9
Disp = 9/10
N = 9*100
Gen <- rnorm(N, mean = Matobj, sd = Disp) #побудова ряду
розподілу
hist(Gen, col="violet") #побудова гістограми ряду розподілу
freq = table(Gen) #таблиця частот
Gen.mean <- mean(Gen) #середнє арифметичне
Med <- median(Gen) #медіана
Mode <- function(x){
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
Gen.mode <- Mode(Gen) #мода
Qua <- quantile(Gen) #розмах (в процентах)
```

Результатом буде побудована гістограма (рис. 2.1) та дані (рис. 2.2). Таблиця частот, середнє арифметичне та інші дані можна переглянути у Global Environmental та вивести на консоль раніше розглянутими у пункті 2.1 функціями.

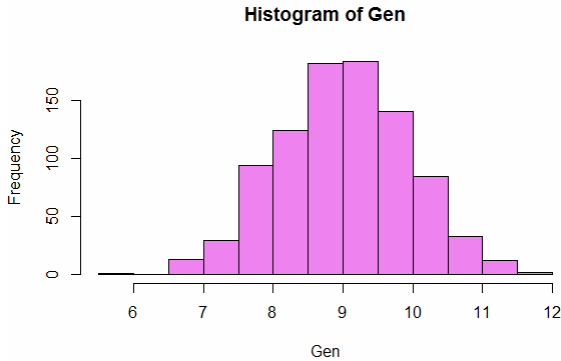


Рисунок 2.1 – Гістограма нормального розподілу

Global Environment	
values	
Disp	0.9
freq	'table' int [1:900(1d)] 1 1 1 1 1 1 1 1 1 ...
Gen	num [1:900] 9.77 7.17 9.68 8.99 10.22 ...
Gen.mean	9.00345852740242
Gen.mode	9.77118723912186
k	900L
Matoj	9
Med	9.02447287134164
Med2	9.06123718789205
N	900
Nya	12.3157844979733
Qua	Named num [1:5] 5.85 8.37 9.02 9.65 11.7
sum	8142.73930732055

Рисунок 2.2 – Результати аналізу

Приклад реалізації у R-Studio з використанням функцій, написаних самостійно

Приведемо приклад коду з використанням функцій, написаних самостійно:

```
Matoj = 9
Disp = 9/10
N = 9*100
```

```

cat("Мат. очікування = ")
cat( Matoj, "\n")
cat("Дисперсія = ")
cat(Disp, "\n")
cat("N = ")
cat(N, "\n")
Gen <- rnorm(N, mean = Matoj, sd = Disp) #побудова рядка
розподілу
hist(Gen, col="violet") #побудова гістограми рядка розподілу
freq = table(Gen) #таблиця частот
Gen <- sort(Gen,decreasing = FALSE)#сортування за спаданням
cat("Вибірка:", "\n")
print(Gen)
cat("таблиця частот:", "\n")
print(freq)

#розрахунок середнього арифметичного
sum = 0
for(k in 1:N){
  sum <-sum+Gen[k]
}
Gen.mean <- sum/N
cat("середнє арифметичне:", "\n")
print(Gen.mean)

#медіана
# N – парне, отже
Med2 <- (Gen[N/2] + Gen[N/2-1])/2
cat("медіана:", "\n")
print(Med2)
Mode <- function(x){
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
Gen.mode <- Mode(Gen) #мода
cat("мода:", "\n")
print(Gen.mode)

```

#квантиль

```

inqua <- function(L, a){
  if (a == 1){
    K <- (a * N)
    if ((K+1)<(N*a)) {
      m <- L[K+1]
      print(m)
    }
    else if((K+1)==(N*a))
    {
      m <- (L[K] + L[K+1])/2
      print(m)
    }
    else if((K+1)>(N*a))
    {
      m <- L[K]
      print(m)
    }
  }
  else{
    K <- (a * N) + 1
    if ((K+1)<(N*a)) {
      m <- L[K+1]
      print(m)
    }
    else if((K+1)==(N*a))
    {
      m <- (L[K] + L[K+1])/2
      print(m)
    }
    else if((K+1)>(N*a))
    {
      m <- L[K]
      print(m)
    }
  }
}
cat("квантиль: ", "\n")

```

```

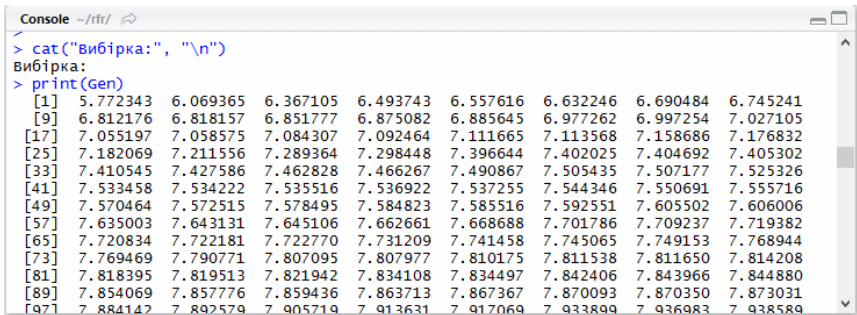
Quak <- function(L){
  cat("0%", "\n")
  R <- inqua(L,0)
  cat("25%", "\n")
  R <- inqua(L,0.25)
  cat("50%", "\n")
  R <- inqua(L,0.5)
  cat("75%", "\n")
  R <- inqua(L, 0.75)
  cat("100%", "\n")
  R <- inqua(L,1)
}
Nya <- Quak(Gen)

```

У даному коді функції середнього арифметичного, медіани та іншого написані вручну за допомогою базових команд мови R.

Функція *Quak(L)* використовується для виведення квантиля. У середині функції для зручності розрахунків була використана функція *inqua (L, a)*, яка власне і розраховує квантиль залежно від параметрів, що вводяться.

В результаті інформація про вибірку та результати будуть знаходитись на консолі (рис. 2.3 – 2.4).



```

Console ~/itr/
> cat("Вибірка:", "\n")
Вибірка:
> print(Gen)
[1] 5.772343 6.069365 6.367105 6.493743 6.557616 6.632246 6.690484 6.745241
[9] 6.812176 6.818157 6.851777 6.875082 6.885645 6.977262 6.997254 7.027105
[17] 7.055197 7.058575 7.084307 7.092464 7.111665 7.113568 7.158686 7.176832
[25] 7.182069 7.211556 7.289364 7.298448 7.396644 7.402025 7.404692 7.405302
[33] 7.410545 7.427586 7.462828 7.466267 7.490867 7.505435 7.507177 7.525326
[41] 7.533458 7.534222 7.535516 7.536922 7.537255 7.544346 7.550691 7.555716
[49] 7.570464 7.572515 7.578495 7.584823 7.585516 7.592551 7.605502 7.606006
[57] 7.635003 7.643131 7.645106 7.662661 7.668688 7.701786 7.709237 7.719382
[65] 7.720834 7.722181 7.722770 7.731209 7.741458 7.745065 7.749153 7.768944
[73] 7.769469 7.790771 7.807095 7.807977 7.810175 7.811538 7.811650 7.814208
[81] 7.818395 7.819513 7.821942 7.834108 7.834497 7.842406 7.843966 7.844880
[89] 7.854069 7.857776 7.859436 7.863713 7.867367 7.870093 7.870350 7.873031
[97] 7.884142 7.892579 7.905719 7.913631 7.917069 7.933899 7.936983 7.938589

```

Рисунок 2.3 – Дані про вибірку на консолі

```

Console ~/rtr/
+ k <- inqua(L, 0.75)
+ cat("100%", "\n")
+ R <- inqua(L,1)
+ }
> Nya <- Quak(Gen)
0%
[1] 5.772343
25%
[1] 8.415897
50%
[1] 9.020881
75%
[1] 9.632315
100%
[1] 12.13423
> |

```

Рисунок 2.4 – Дані про вибірку на консолі (продовження)

Завдання на самостійну роботу

1. Скориставшись рекомендованою літературою вивчити можливості мови програмування R.
2. Вивчити загальні положення теорії ймовірностей.
3. Згенерувати стовпець на основі наступної інформації:
 $N = Var * 10$, $\mu = Var$, $\sigma^2 = Var / 10$, де Var – номер варіанта, N – кількість дослідів, μ – математичне сподівання, σ^2 – дисперсія.
4. Побудувати гістограму, таблицю частот, отримати описові статистики ряду розподілу з використанням внутрішніх функцій мови R або реалізувати функції на мові R самостійно.
5. Зробити висновок.
6. Оформити звіт.
7. Відповісти на контрольні питання.

Зміст звіту

- 1 Назва і мета роботи.
2. Постановка завдання.
3. Вихідні дані.
4. Результати аналізу, отримані з використання мови програмування R, зокрема таблиця частот, гістограми і описові статистики.
5. Висновки по результатам аналізу.

Контрольні питання

1. Основні характеристики розподілу ймовірностей. Записати аналітичні вирази.
2. Що таке квантіль, мода, медіана?
3. Як визначити знак коефіцієнта асиметрії з вигляду графіка щільності ймовірності?
4. Як визначити знак коефіцієнта ексцесу з вигляду графіка щільності ймовірності?
5. Що таке ряди розподілу?
6. Які характеристики розподілу ви знаєте?
7. Як будується гістограма?
8. Що таке таблиця частот?
9. На що впливає ширина інтервалу?

САМОСТІЙНА РОБОТА №3 СТАТИСТИЧНА ПЕРЕВІРКА ГІПОТЕЗ ЗАСОБАМИ МОВИ R

Мета роботи

Вивчити методику статистичної перевірки гіпотез. Отримати основні характеристики розподілу ймовірностей випадкової величини та перевірити гіпотезу про закон розподілу вибірки з використанням мови програмування R.

Основні функції та команди, необхідні для виконання роботи

В роботі необхідно використати наступні стандартні функції та команди R. Функція *array()* – призначена для створення багатовимірних масивів.

Приклад:

`aux=array(rep(0,60),dim=c(3,4,5))` # створюємо набір з 5 матриць, що складаються з 3 рядків, 4 стовпців, заповнених нулями

Функція *list()* – призначена для створення списків. В список можна включати поєднання будь-яких типів даних.

Приклад створення та заповнення списку:

`list1 <- c("A", "B", "C")`

`list2 <- c(FALSE, TRUE)`

`my.list <- list(Text= list1, Logic= list2)`

Звертання до елементів списку:

`> my.list$Text[2]`

`[1] "B"`

Функція *density()* – оцінює щільність ймовірності.

Приклад використання функції:

`plot(density(my.array),main="Density estimate of data")`

Функція *shapiro.test()* – призначена для виконання тесту Шапіро-Уїлка. Функції передається вибірка у вигляді масива.

Функція *chisq.test()* – призначена для перевірки критерію узгодженості Пірсона. Функції передається вибірка у вигляді масива.

Функція *t.test()* – призначена для виконання тесту Ст'юденту.

Функції передається дві вибірки у вигляді масива.

Функція *install.packages()* – завантажує пакет зі стороннього джерела.

Функція *data()* – завантажує таблицю з даними.

Функція *attach()* – додає таблицю з даними до списку наявних змінних.

Функція *tapply()* – застосовує функцію до кожної сукупності значень, створеної відповідно до рівнів певного фактора.

Перевірка гіпотез у R-Studio для однієї вибірки

Виконаємо перевірку гіпотез у R-Studio для однієї вибірки. Для цього заповнюємо одновимірний масив 25 випадковими числами:

```
my.array <- array( rnorm(25, mean = 25, sd = 1),dim = c(25,1),
dimnames = list(c(1:25) , c("Col1") ))
# виводимо вміст масиву
my.array
```

```
hist(my.array,main="Histogram of observed data")
# будуємо гістограму за щільністю
plot(density(my.array),main="Density estimate of data")
```

```
#для виконання тесту Шапіро-Уїлка виконаємо функцію
shapiro.test(), де як аргумент передаємо масив
shapiro.test(my.array)
# у масив my1 заносимо ймовірності розподілення масиву my
my1.array<- rnorm(my.array, mean = 25, sd = 1)
# для пошуку критерію узгодженості Пірсона використовуємо
функцію chisq.test
chisq.test(my1.array)
# для виконання тесту Колмогорова-Смирнова використовуємо
функцію ks.test
ks.test(my.array, my1.array)
```

Результатом буде побудована гістограма (рис. 3.1). Результати тесту Колмогорова – Смирнова та тесту Шапіро-Уїлка будуть виведені на консоль (рис. 3.2).

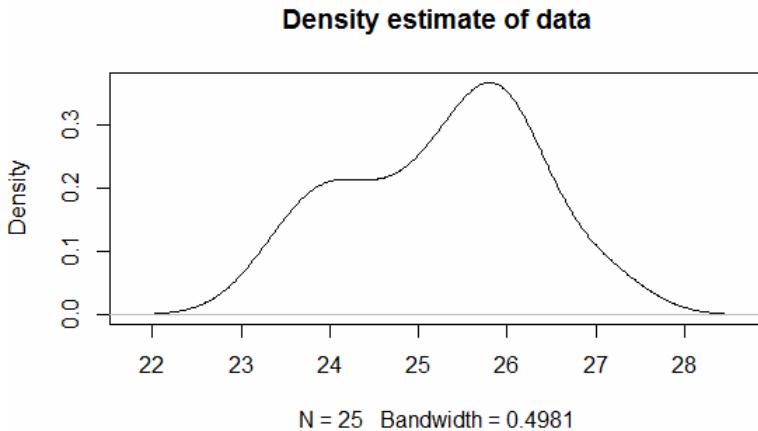


Рисунок 3.1 – Гістограма за щільністю

```
shapiro-wilk normality test
data:  my.array
W = 0.95496, p-value = 0.3232

chi-squared test for given probabilities
data:  my1.array
X-squared = 4.164, df = 24, p-value = 1
```

Рисунок 3.2 – Результати перевірки гіпотез

Перевірка гіпотез у R-Studio для двох вибірок

Виконаємо перевірку гіпотез у R-Studio двох вибірок. Результати наведено на рисунку 3.3.

```
# Набрати в консолі для роботи прикладу
# > install.packages("ISwR")
```

```
library(ISwR)    #Завантаження ISwR
data(energy)     #Завантажує таблицю energy
```

```
attach(energy) #Додає таблицю energy
energy         #Виводить energy
tapply(expend, stature, mean) # застосовує функцію mean до
expend, за допомогою у stature
```

```
# побудова діаграми розмаху
boxplot(expend ~ stature,
        xlab = "Тип будови тіла",
        ylab = "Витрати енергії",
        main = "T-test",
        col = "coral", data = energy)
#t-тест Стьюдента
t.test(expend ~ stature)
```

```
> boxplot(expend ~ stature,
+         xlab = "тип будови тіла",
+         ylab = "Витрати енергії",
+         main = "T-test",
+         col = "coral", data = energy)
> t.test(expend ~ stature)

welch Two Sample t-test

data: expend by stature
t = -3.8555, df = 15.919, p-value = 0.001411
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.459167 -1.004081
sample estimates:
mean in group lean mean in group obese
      8.066154      10.297778
```

Рисунок 3.3 – Результати t-Testy

Завдання на самостійну роботу

1. Вивчити загальні положення теорії статистичної перевірки гіпотез.
2. Згенерувати стовпець на основі наступної інформації:
 $N = Var * 10$, $\mu = Var$, $\sigma^2 = Var/10$, де Var – номер варіанта, N – кількість дослідів, μ – математичне сподівання, σ^2 – дисперсія.
3. Перевірити гіпотезу про нормальний розподіл отриманої

вибірки на основі критерію Пірсона χ^2 та критерію Колмогорова з використанням внутрішніх функцій мови R.

4. Зробити висновок.
5. Оформити звіт.
6. Відповісти на контрольні питання.

Зміст звіту

1. Назва і мета роботи.
2. Теоретичний аналіз щодо критеріїв перевірки статистичних гіпотез.
3. Індивідуальне завдання.
4. Отримані результати обчислень та графіки.
5. Аналіз отриманих результатів та висновки.

Контрольні питання

1. Які ви знаєте параметри нормального закону розподілу випадкових величин.
2. Наведіть аналітичне вираження щільності ймовірності для нормального закону розподілу.
3. Накресліть функцію щільності ймовірності для нормального закону.
4. Чому дорівнюють коефіцієнт асиметрії та коефіцієнт ексцесу для нормального закону?
5. Яким критерієм визначається закон розподілу випадкової величини? Записати аналітичний вираз.
6. Як змінюється вигляд гістограми при зміні величини інтервалу?
7. Як використовується правило трьох сігм для визначення закону розподілу випадкової величини?
8. Необхідність застосування статистичних методів обробки результатів спостережень.
9. У чому полягає основний принцип сатистичних гіпотез?
10. Які гіпотези можна перевірити за допомогою критерія Пірсона, Колмогорова, Фішера, Стьюдента, Кохрена?
11. Як визначити довірчі інтервали для математичного сподівання?

САМОСТІЙНА РОБОТА №4 КОРЕЛЯЦІЙНИЙ ТА РЕГРЕСІЙНИЙ АНАЛІЗ З ВИКОРИСТАННЯМ МОВИ R

Мета роботи

Ознайомитися з можливостями мови програмування R для вирішення задач кореляційного та регресійного аналізу.

Основні функції та команди, необхідні для виконання роботи

Функція $cor(x)$ – застосовується для обчислення кореляційної матриці.

Функція $cor(x, y)$ – застосовується для обчислення коефіцієнта кореляції.

Приклад:

```
cor(x, y = NULL, use = "everything",
method = c("pearson", "kendall", "spearman"))
```

де x – матриця чи таблиця;

y – необов'язковий аргумент, матриця чи таблиця;

use – необов'язковий параметр, що надає можливість обчислення кореляції за відсутності даних (може приймати такі значення: "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs");

$method$ – визначає тип кореляції. Може приймати такі значення: pearson, spearman(для непараметричних даних) або kendall.

Функція $str()$ – для з'ясування структури об'єкту.

Функція $summary()$ – загального призначення. Швидкий розрахунок основних параметрів описової статистики. Таких як мінімальне, максимальне значення, розрахунок дисперсії, середнього значення та квантилів.

Функція $with()$ – слугує показником на вибірку, з якою треба працювати іншій функції.

Функція $cor.test()$ – схожа з $cor()$, проте, на відміну від $cor()$, виконує ще й оцінку статистичної значимості коефіцієнтів регресії, перевіряючи нульову гіпотезу про рівність їх нулю.

Функція $lm(formula = y \sim x)$ призначена для задання формули

простої лінійної регресії $y=a+bx$.

Функція *summary(lm(y~x))* призначена для отримання таблиці значимості коефіцієнтів рівняння лінійної регресії $y=a+bx$.

Функція *abline(lm(y~x))* призначена для побудови графіку простої лінійної регресії $y=a+bx$ методом найменших квадратів.

Функція *anova(lm(y~x))* призначена для побудови таблиці дисперсійного аналізу.

Функція *read.csv()* – призначена для читання значень, розділених комами з файлів формату ('.csv') .

Приклад:

```
MyData <- read.csv(file="TheDataIWantToReadIn.csv",
header=TRUE, sep=",", fill=TRUE)
```

Аргумент *file* – місце розташування файлу.

Аргумент *header = TRUE* означає, що в файлі присутній заголовок.

Аргумент *sep* – символ роздільник. Значення в кожному рядку файлу розділені цим символом. За замовченням *sep = ""*.

Аргумент *fill = TRUE* означає, що в файлі відключені коментарі.

Функція *view()* – відображає таблицю, зчитану з файлу формату .csv.

Розрахунок коефіцієнту кореляції у R-Studio

Розглянемо приклад розрахунку коефіцієнту кореляції засобами мови R.

```
X.vector <- 5:15 # вибірка X
```

```
Y.vector <- 7:17 # вибірка Y
```

```
cor(X.vector,Y.vector) # коефіцієнт кореляції
```

```
# результат виконання:
```

```
#[1] 1
```

```
z.vector <- -3*X.vector + 8
```

```
cor(X.vector,z.vector)
```

```
# результат виконання:
```

```
#[1] -1
```

```
# Створимо дві вибірки: роки та відсотки
year <- c(2010 , 2011 , 2012 , 2013 , 2014)
rate <- c(9.34 , 8.50 , 7.62 , 6.93 , 6.60)
```

```
# будуємо графік залежності проценту за кредитом від року
```

```
plot(year,rate, main="Процентна ставка на 4 роки для кредиту на авто")
```

```
# розраховуємо коефіцієнт кореляції
cor(year,rate)
# [1] -0.9880813
```

Процентна ставка на 4 роки для кредиту на авто

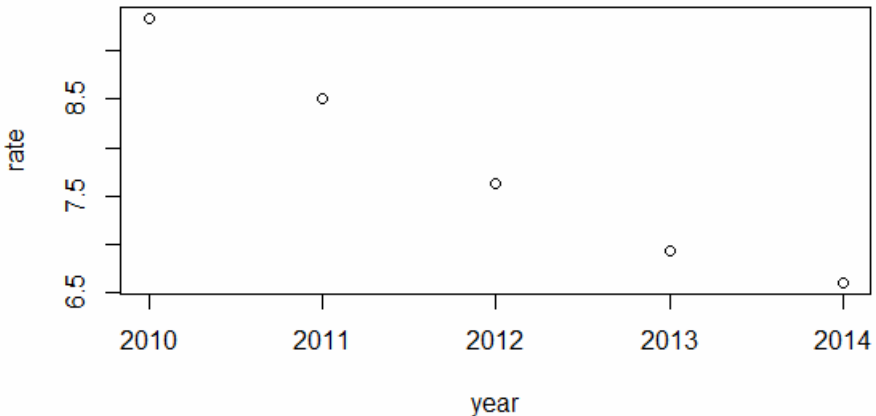


Рисунок 4.1 – Графік залежності проценту за кредитом від року

Побудова лінійної регресії засобами мови R

Розглянемо приклад побудови лінійної регресії засобами мови

R.

```
x = scan('D:/x.txt')
y = scan('D:/y.txt')
print(cor.test(x, y, use="complete.obs"))
p.lm<-lm(formula = y~x)
```

```
print(summary(p.lm))
plot(x,y)
abline(lm(y~x))
```

Результати виконання програми:

Pearson's product-moment correlation

```
data: x and y
t = -1.0939, df = 10, p-value = 0.2996
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7585132  0.3040154
sample estimates:
      cor
-0.3269267
```

Call:

```
lm(formula = y ~ x)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-5.1580	-1.7529	0.4109	1.7730	4.6695

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.57184	1.72650	0.910	0.384
x	-0.04138	0.03783	-1.094	0.300

Residual standard error: 2.881 on 10 degrees of freedom

Multiple R-squared: 0.1069, Adjusted R-squared: 0.01757

F-statistic: 1.197 on 1 and 10 DF, p-value: 0.2996

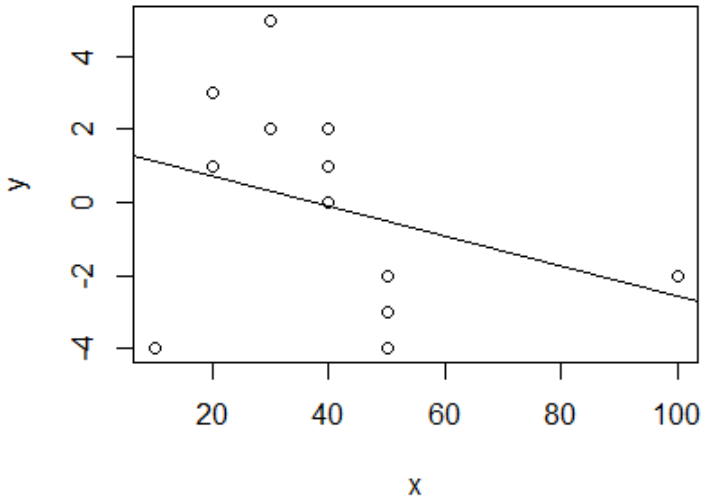


Рисунок 4.2 – Графік лінійної регресії

Завдання на самостійну роботу

1. Вивчити теоретичний матеріал по кореляційному та регресійному аналізу.
2. Отримати індивідуальне завдання.
3. Використовуючи дані згідно індивідуального завдання, виконати кореляційний та регресійний аналіз з використанням мови R.
4. Отримати таблицю кореляційного аналізу, перевірити коефіцієнти рівняння на значимість, отримати таблицю дисперсійного аналізу, перевірити адекватність моделі, побудувати рівняння регресії та графік, отримати множинний коефіцієнт регресії, зробити висновки за результатами аналізу.
5. Оформити звіт.

Зміст звіту

1. Назву і мету роботи.
2. Постановку задачі.
3. Таблицю вихідних даних.
4. Основні формули розрахунку параметрів та критеріїв.
5. Таблиці кореляційного аналізу і графіки з коментаріями.
6. Висновки по результатам аналізу.

7. Вихідні дані.
8. Коефіцієнти рівнянь, таблиця дисперсійного аналізу, рівняння регресії
9. Графічна інтерпретація задачі регресійного аналізу.
10. Прогнози та таблиця R^2 .
11. Висновки.

Контрольні питання

1. Що значать поняття статистична залежність, функціональна залежність, кореляційна залежність?
2. Коли використовується кореляційний аналіз?
3. Як визначається індекс кореляції і що він показує?
4. Як визначається коваріація і коефіцієнт кореляції? Як визначити вибіркове значення коефіцієнта кореляції?
5. Як визначається кореляційне відношення? Який його фізичний зміст?
6. Яким чином перевіряється гіпотеза про відсутність кореляційного зв'язку?
7. Для чого призначений узагальнений засіб обчислення парних кореляційних характеристик?
8. Як розраховується узагальнений коефіцієнт кореляції?
9. В чому полягає мета регресійного аналізу?
10. Для чого використовується метод найменших квадратів?
11. Наведіть передумови регресійного аналізу.
12. Як виконується перевірка значущості оцінок коефіцієнтів рівняння регресії?
13. За допомогою якого критерію виконується перевірка адекватності рівняння регресії?
14. Як виконується вибір найкращої моделі рівняння регресії?

САМОСТІЙНА РОБОТА №5 ПОВНИЙ ФАКТОРНИЙ ЕКСПЕРИМЕНТ

Мета роботи

Вивчити апарат математичного моделювання методом планування експерименту. Побудувати математичну модель за експериментальними даними, отриманими при проведенні повного факторного експерименту типу 2^k з використанням мови R.

Програмна реалізація повного факторного експерименту засобами мови R

Наведемо приклад програмної реалізації побудови математичної моделі за експериментальними даними для кількості факторів $N = 6$, де $x_1, x_2, \dots, x_6$ – відповідні фактори, y – відгук.

```
x1 <- c(-1,... 1)
```

```
x2 <- c(-1,... 1)
```

```
x3 <- c(1,... 1)
```

```
x4 <- c(-1,... -1)
```

```
x5 <- c(-1,... -1)
```

```
x6 <- c(1,... 1)
```

```
y<- c(19.0832376437 ,... 17.4004932366)
```

$x_{12} <- x_1 * x_2$ # в залежності від кількості x скласти всі можливі комбінації

```
x13<-x1*x3
```

```
x14<-x1*x4
```

```
x15<-x1*x5
```

```
x16<-x1*x6
```

```
x23<-x2*x3
```

```
x24<-x2*x4
```

```
x25<-x2*x5
```

```
x26<-x2*x6
```

```
x34<-x3*x4
```

```
x35<-x3*x5
```

```
x36<-x3*x6
```

```
x45<-x4*x5
```

```
x46<-x4*x6
```

```
x56<-x5*x6
```

```

x<-rbind(x1, x2, x3, x4, x5, x6, x12,x13,x14,x15,
         x16,      x23,      x24,      x25,
         x26,      x34,      x35,
         x36,      x45,      x46,
         x56
        )
N=length(y)
a0<-sum(y)/N
a<-c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) #
кількість нулів відповідає кількості комбінацій з x
M=length(a)
for(j in 1:N)
  for(i in 1:M)
    a[i]<-a[i]+(y[j]*x[i, j])
for(i in 1:M)
  a[i]<-a[i]/N
s<-c("x1", "x2", "x3", "x4", "x5", "x6", "x1*x2", "x1*x3",
"x1*x4", "x1*x5", "x1*x6", "x2*x3", "x2*x4", "x2*x5",
"x2*x6", "x3*x4", "x3*x5", "x3*x6", "x4*x5", "x4*x6",
"x5*x6")
s1=""
for(i in 1:M)
  s1<-paste0(s1, sprintf("% + f%s", a[i], s[i]))
s2<-paste0("y = ", a0, s1)
s2

```

Результати роботи RStudio

```

[1] "y = 15.0781017293312-0.233105*x1-0.308488*x2+
0.261353*x3-0.187244*x4-0.084091*x5+0.048129*x6-
0.123289*x1*x2+0.156115*x1*x3-0.080164*x1*x4-
0.281490*x1*x5+0.033936*x1*x6-0.276591*x2*x3-0.067796*x2*x4-
0.196198*x2*x5+0.059099*x2*x6-0.235224*x3*x4-
0.160939*x3*x5+0.213224*x3*x6-0.125783*x4*x5+0.551356*x4*x6-
0.262511*x5*x6"

```

Завдання на самостійну роботу

1. Ознайомитись із методикою ПФЕ.
2. Отримати індивідуальне завдання у викладача.

3. Згенерувати план експерименту 2^k , де

$$k = \begin{cases} Var, & Var \leq 5 \\ Round(Var * 2 / 5), & 5 < Var \leq 15 \\ Round(Var / 10), & Var > 15 \end{cases}$$

4. Згенерувати випадкову вибірку для відгуку.

5. Побудувати матрицю планування, отримати оцінки коефіцієнтів регресії та рівняння регресії з використанням мови R.

6. Зробити висновки.

7. Відповісти на контрольні питання.

8. Оформити звіт.

Зміст звіту

1. Мета та назва роботи.
2. Початкові дані.
3. Матриця планування.
4. Оцінки коефіцієнтів регресії та таблиця перевірки адекватності рівняння регресії.
5. Висновки.

Контрольні питання

1. Дайте визначення теорії планування експерименту.
2. Дайте визначення активного та пасивного експерименту.
3. Вимоги для вибору відгуків, факторів, координат базової точки та ступенів варіювання.
4. Призначення ПФЕ та етапи його проведення.
5. Що таке матриця планування. Як визначається кількість можливих комбінацій рівнів варіювання.
6. Властивості МП. Наведіть приклади кодування факторів.
7. З якою метою необхідна рандомізація дослідів.
8. Як впливає величина ступенів варіювання на значущість коефіцієнтів і на адекватність рівняння регресії.

САМОСТІЙНА РОБОТА №6

ОРТОГОНАЛЬНЕ ЦЕНТРАЛЬНЕ КОМПОЗИЦІЙНЕ ПЛАНУВАННЯ

Мета роботи

Вивчити методику ортогонального центрального композиційного планування (ОЦКП) при обробці результатів активного експерименту та побудуванні нелінійних математичних моделей; побудувати математичні моделі другого порядку, оцінити їх адекватність, використовуючи мову програмування R.

Програмна реалізація ортогонального центрального композиційного планування засобами мови R

Наведемо приклад програмної реалізації побудови математичної моделі за експериментальними даними для кількості факторів $N = 3$, де x_1, x_2, x_3 – відповідні фактори, y – відгук.

```
x1 <- c(-1,... 1)
x2 <- c(-1,... 1)
x3 <- c(1,... 1)
y<- c(19.0832376437 ,... 17.4004932366)
x11<-x1*x1 # в залежності від кількості x скласти всі можливі
комбінації
x12<-x1*x2
x13<-x1*x3
x22<-x2*x2
x23<-x2*x3
x33<-x3*x3
x<-rbind(x1, x2, x3, x11, x12, x13, x22, x23, x33)
N=length(y)
a0<-sum(y)/N
a<-c(0, 0, 0, 0, 0, 0, 0, 0, 0) # кількість нулів відповідає
кількості комбінацій з x
M=length(a)
for(j in 1:N)
  for(i in 1:M)
    a[i]<-a[i]+(y[j]*x[i, j])
```

```

for(i in 1:M)
  a[i]<-a[i]/N
  s<-c("x1", "x2", "x3", "x1*x1", "x1*x2", "x1*x3", "x2*x2",
"x2*x3", "x3*x3")
  s1=""
  for(i in 1:M)
    s1<-paste0(s1, sprintf("% + f%s", a[i], s[i]))
  s2<-paste0("y = ", a0, s1)
  s2

```

Результати роботи RStudio

```

[1] "y" = 23.3531106225444-3.057971*x1+2.859360*x2+
10.335217*x3+14.433793*x1*x1+0.794178*x1*x2+0.088605*x1*x3+14.
321433*x2*x2-0.199925*x2*x3+19.589340*x3*x3"

```

Завдання на самостійну роботу

1. Ознайомитись із методикою ОЦКП.
2. Отримати індивідуальне завдання у викладача.
3. Згенерувати план експерименту 2^k , де

$$k = \begin{cases} Var, & Var \leq 5 \\ Round(Var * 2 / 5), & 5 < Var \leq 15 \\ Round(Var / 10), & Var > 15 \end{cases}$$

4. Згенерувати випадкову вибірку для відгуку.
5. Розрахувати кількість зоряних точок, кількість центральних точок, розмір плеча. Отримати матрицю планування, оцінки коефіцієнтів регресії та рівняння регресії з використанням мови R.
6. Зробити висновки.
7. Відповісти на контрольні питання.
8. Оформити звіт.

Зміст звіту

1. Назва і мета роботи.
2. Постановка задачі.
3. Таблиця вихідних даних.
4. Основні формули розрахунку параметрів.
5. Результати виконання ОЦКП з коментарями.

6. Висновки по результатам.

Контрольні питання

1. Коли використовується ОЦКП?
2. З яких частин складається ЦКП?
3. Назвіть види ЦКП. Чим відрізняються види ЦКП?
4. Розкажіть о зоряних точках при ЦКП.
5. Що є критерієм оптимальності для ОЦКП?
6. Що є критерієм оптимальності для РЦКП?
7. Яким чином в ОЦКП забезпечується ортогональність векторів-стовпців, включаючи квадратичні вектори-стовпці?
8. Яким чином, знаючи величину зоряного плеча в нормованому масштабі, можна перейти до звичайного масштабу?
9. Як будується МП при ОЦКП?
10. Методика обчислювання формул ОЦКП.
11. Як виконується аналіз моделей другого порядку? Поняття стаціонарної точки.

ЛІТЕРАТУРА

1. Алексахина С.В. Прикладной статистический анализ данных. Теория. Компьютерная обработка. Области применения / С.В. Алексахина, А.В. Балдина, А.Б. Николаева, В.Ю. Строганова. – М.: ПРИОР, 2002. – 688 с.
2. Тюрин Ю. Н. Анализ данных на компьютере / Ю. Н. Тюрин, А. А. Макаров; под ред. В. И. Фигурнова. – М. : ИНФА-М., 2003. – 544 с.
3. Томашевський О. В. Комп'ютерні технології статистичної обробки даних: навч. посібник для студ. вищ. навч. закл. / О. В. Томашевський, В. П. Рисіков. – Запоріжжя: ЗНТУ, 2006. – 174 с.
4. Мاستицкий С.Э. Статистический анализ и визуализация данных с помощью R / Мастицкий С.Э., Шитиков В.К. – М.: ДМК Пресс., 2014. – 401 с.
5. Gareth James An Introduction to Statistical Learning with Applications in R / Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. – Springer, New York, 2013. – 426 p.
6. Trevor Hastie The Elements of Statistical Learning Data Mining, Inference, and Prediction /Trevor Hastie Robert Tibshirani Jerome Friedman. – Springer, New York, 2013. – 745 p.
7. Вуколов Э.А. Основы статистического анализа. Практикум по статистическим методам и исследованию операций с использованием пакетов STATISTICA и EXCEL/ Э.А. Вуколов. – М.: ФОРУМ, 2008. – 464 с.
8. Кобзарь А.И. Прикладная математическая статистика. Для инженеров и научных работников. / А.И. Кобзарь. – М.: Физматлит, 2006. – 816 с.
9. Берк К. Анализ данных с помощью Microsoft Excel. / К.Берк, П. Кэйри. – М.: Вильямс, 2005. – 560 с.
10. Шипунов А.Б. Наглядная статистика. Используем R! / А.Б. Шипунов, Е.М. Балдин, П.А. Волкова, А.И. Коробейников, С.А.Назарова, С.В. Петров, В.Г. Суфиянов. – М.: ДМК Пресс, 2012. – 298 с.