

10. Лабораторна робота. Візуалізація даних мовою R

Мета: засвоїти можливості роботи мови R до застосування розв'язування задач візуалізації даних.

Теоретичні відомості та методичні рекомендації

R є однією з найкращих мов програмування для візуалізації даних завдяки вбудованим графічним можливостям та потужним бібліотекам. Візуалізація у R дозволяє аналізувати великі обсяги даних, знаходити закономірності та ефективно презентувати результати.

1. Базова графіка в R. Мова R має вбудовані засоби для побудови простих графіків без встановлення додаткових пакетів.

Наприклад, лінія по точках (див. рис. 10.1).

```
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)
```

```
plot(x, y, type="o", col="blue", main="Простий графік",
      xlab="X-вісь", ylab="Y-вісь")
```

Простий графік

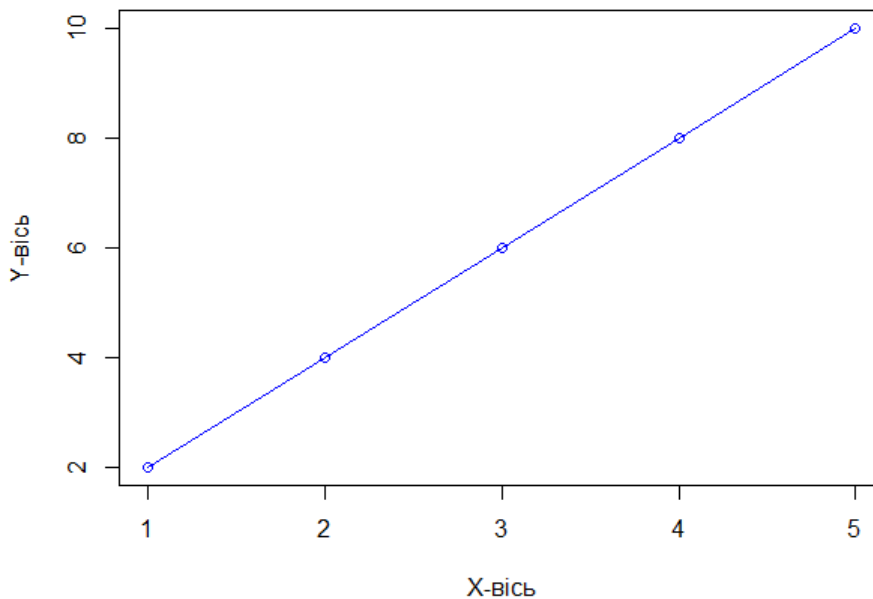


Рисунок 10.1 – Лінія по точках

Тут

- `type="o"` – лінія з точками.
- `col="blue"` – колір лінії.
- `main` – заголовок.
- `xlab, ylab` – підписи осей.

Наприклад, побудуємо графік функції $y = \sin x$ (див. рис. 10.2).

```
# Генеруємо значення x від -π до π
x <- seq(-pi, pi, length.out = 100)
```

```
# Обчислюємо значення  $y = \sin(x)$ 
y <- sin(x)

# Будуємо графік
plot(x, y, type="l", col="blue", lwd=2,
      main="Графік функції  $y = \sin(x)$ ",
      xlab="x", ylab="sin(x)")

# Додаємо сітку
grid()
```

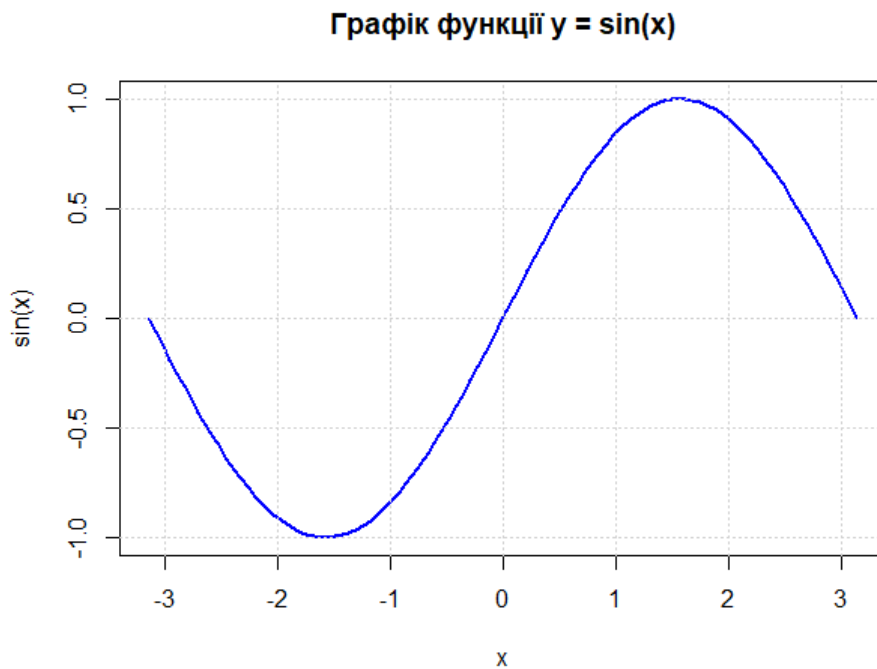


Рисунок 10.2 – Графік функції $y = \sin x$

Тут

- `seq(-pi, pi, length.out = 100)` – створює 100 точок від $-\pi$ до π .
- `sin(x)` – обчислює значення функції.
- `plot(..., type="l")` – будує лінійний графік.
- `col="blue"` – задає колір синусоїди.
- `lwd=2` – збільшує товщину лінії.
- `grid()` – додає сітку.

Мова R має вбудовані засоби для побудови стовпчикових гістограм за допомогою функції `hist()` (див. рис. 10.3).

```
data <- rnorm(1000) # Генеруємо 1000 випадкових значень
hist(data, col="skyblue", main="Гістограма",
      xlab="Значення", breaks=30) # breaks=30 – кількість
інтервалів
```

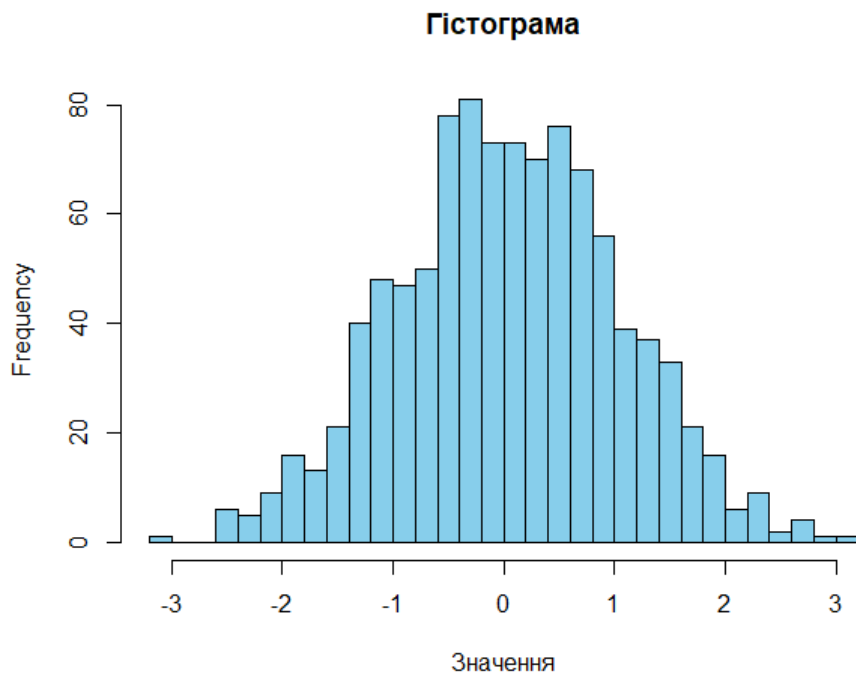


Рисунок 10.3 – Стовпчикова гістограма `hist()`

Коробковий графік (boxplot) показує розподіл даних, медіану та кватилі (див. рис. 10.4).

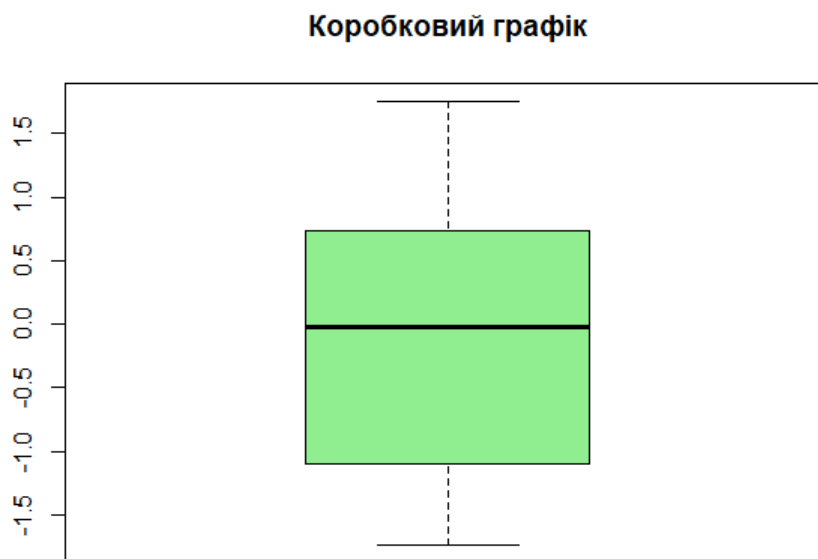


Рисунок 10.4 – Коробковий графік

```
values <- rnorm(50)
boxplot(values, main="Коробковий графік", col="lightgreen")
```

Кругова діаграма використовується для візуалізації часток категорій у загальному обсязі даних. У R можна створити кругову діаграму використовуючи функцію `pie()` (Base R Graphics);

Наприклад, (див. рис. 10.5).

```
# Дані
categories <- c("A", "B", "C", "D")
values <- c(25, 30, 20, 25) # Частки у відсотках
```

```
# Побудова кругової діаграми
pie(values, labels = categories, col =
rainbow(length(values)), main = "Кругова діаграма")
Кругова діаграма
```

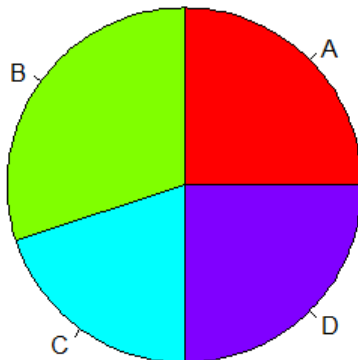


Рисунок 10.5 – Кругова діаграма

Тут

- values – вектор часток;
- labels = categories – підписи для секторів;
- col = rainbow(length(values)) – кольори секторів;
- main = "Кругова діаграма" – заголовок.

2. Графіка в R за допомогою бібліотеки ggplot2. Бібліотека ggplot2 – це одна з найпопулярніших бібліотек для візуалізації даних у R. Вона базується на концепції Grammar of Graphics (граматика графіки), що дозволяє легко створювати складні графіки шляхом додавання різних шарів.

Основні переваги ggplot2:

- гнучкість та налаштовуваність;
- професійний вигляд графіків;
- можливість поєднувати різні типи візуалізацій;
- можливість додавання інших графічних елементів;
- інтуїтивний синтаксис.

Якщо бібліотека ще не встановлена, то потрібно виконати команду у консолі:

```
install.packages("ggplot2")
```

Після встановлення пакета ви можете завантажити його за допомогою функції `R library("ggplot2")`. Наприклад, (див. рис. 10.6).

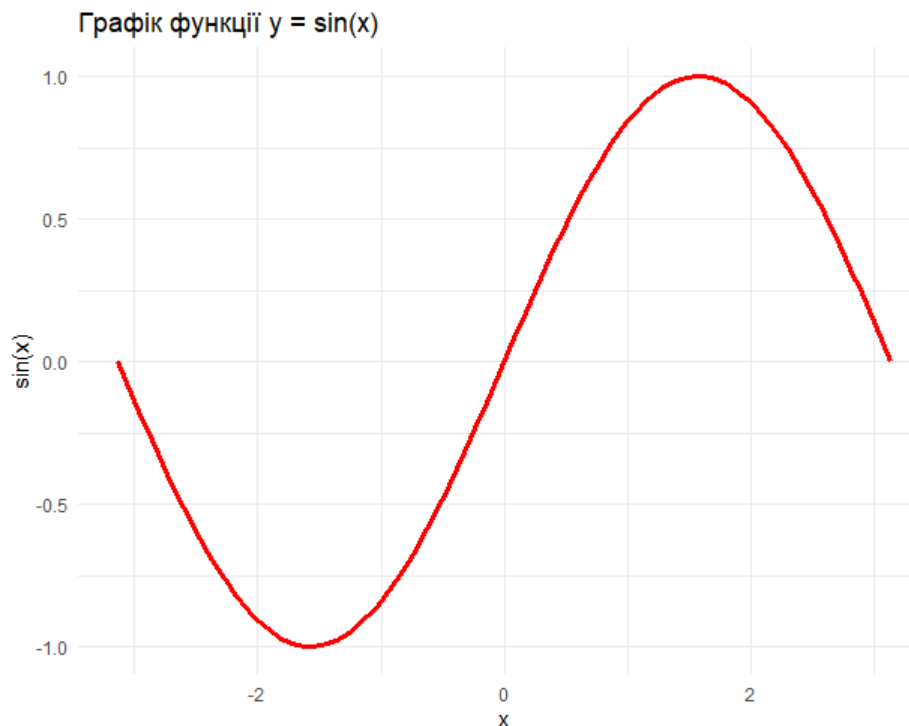


Рисунок 10.6 – Графік функції $y = \sin x$ за допомогою бібліотеки ggplot2

```
# Завантажуємо бібліотеку ggplot2
library(ggplot2)

# Створюємо датафрейм для ggplot2
df <- data.frame(x = seq(-pi, pi, length.out = 100))
df$y <- sin(df$x)

# Будуємо графік за допомогою ggplot2
ggplot(df, aes(x=x, y=y)) +
  geom_line(color="red", size=1.2) +
  ggtitle("Графік функції  $y = \sin(x)$ ") +
  xlab("x") + ylab("sin(x)") +
  theme_minimal()
```

Можна вивести на екран дві функції. Наприклад, додамо функцію $y = \cos x$, отримаємо (див. рис. 10.7).

```
df$cos_y <- cos(df$x)

ggplot(df, aes(x=x)) +
  geom_line(aes(y=y, color="sin(x)"), size=1.2) +
  geom_line(aes(y=cos_y, color="cos(x)"), size=1.2) +
  ggtitle("Графіки  $y = \sin(x)$  та  $y = \cos(x)$ ") +
  xlab("x") + ylab("Значення") +
  scale_color_manual(name="Функції",
values=c("sin(x)"="red", "cos(x)"="blue")) +
  theme_minimal()
```

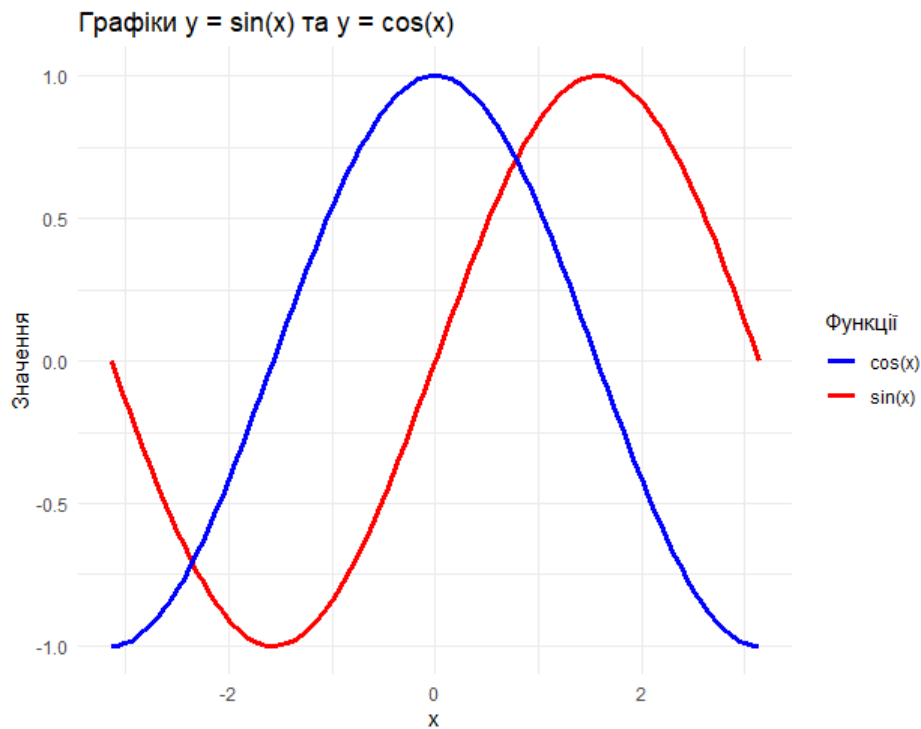


Рисунок 10.7 – Графіки функцій $y = \sin x$ і $y = \cos x$

Приклад створення гистограми за допомогою функції

`geom_histogram()` з бібліотеки `ggplot2` (див. рис. 10.8).

```
df <- data.frame(values = rnorm(1000))
```

```
ggplot(df, aes(x = values)) +
```

```
  geom_histogram(binwidth = 0.5, fill="blue",
```

```
  color="black") +
```

```
  ggtitle("Гистограма") +
```

```
  theme_classic()
```

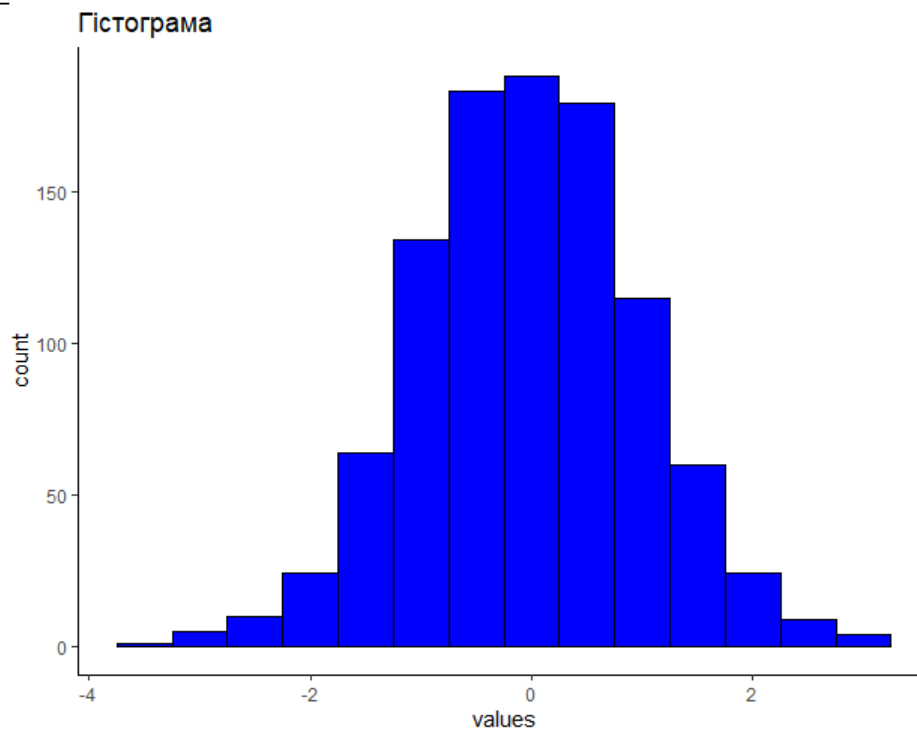


Рисунок 10.8 – Гистограма у `ggplot2`

Приклад створення кругової діаграми за допомогою функції `geom_bar()` з бібліотеки `ggplot2` (див. рис. 10.9).
Кругова діаграма

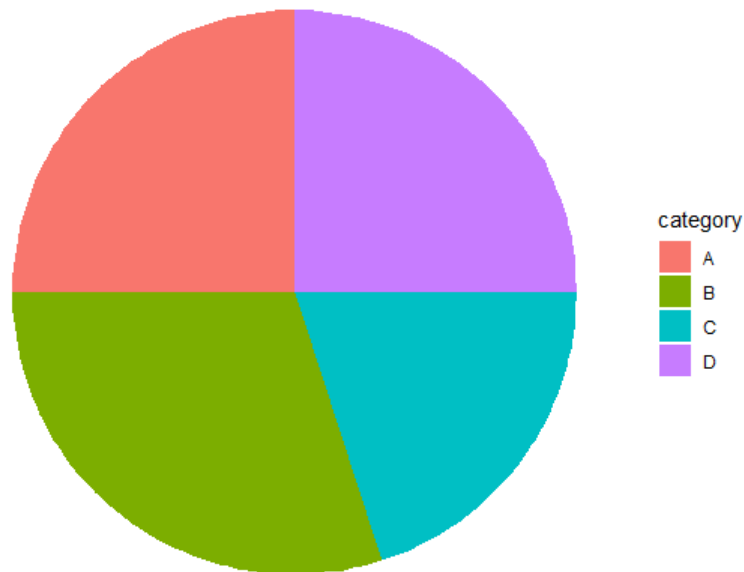


Рисунок 10.9 – Кругова діаграма у `ggplot2`

```
library(ggplot2)

# Дані
df <- data.frame(category = c("A", "B", "C", "D"),
                 values = c(25, 30, 20, 25))

# Побудова кругової діаграми
ggplot(df, aes(x = "", y = values, fill = category)) +
  geom_bar(stat = "identity", width = 1) + # Створюємо
стовпчики
  coord_polar("y", start = 0) + # Полярна координатна
система
  ggtitle("Кругова діаграма") +
  theme_void() # Видаляємо осі та сітку
Тут
- geom_bar(stat="identity", width=1) – створює стовпчики,
які потім перетворюються в кругові сектори;
- coord_polar("y", start=0) – змінює систему координат на
полярну, утворюючи кругову діаграму;
- theme_void() – прибирає сітку та осі;
- fill = category – задає кольори для секторів.
```

Отже, R має широкий вибір інструментів для візуалізації даних, що робить його ідеальним для аналізу та презентації інформації. Якщо потрібно швидко створити кругову діаграму – використовуйте функцію `pie()`, а для професійної візуалізації – пакет `ggplot2`.

Завдання до лабораторної роботи

1. Побудова базових графіків у R.

- 1) Створити випадковий набір даних з 50 чисел за допомогою функції `rnorm()`.
- 2) Побудувати графік розсіювання через функцію `plot()`.
- 3) Побудувати гістограму через функцію `hist()`.
- 4) Побудувати коробковий графік через функцію `boxplot()`.
- 5) Побудувати стовпчикову діаграму через функцію `barplot()`.

2. Побудувати графіки функцій. Використати `ggplot2` для побудови графіків.

1) $y = \sin(x) + \frac{1}{2}\sin(2x) + \frac{1}{3}\sin(3x)$, $y = \sqrt[7]{x^3 - 3x^2 + 5}$.

2) $y = 8x^2(x^2 - 1)^3$, $y = \operatorname{tg}\left(9x - \frac{\pi}{2}\right)$.

3) $y = \frac{x}{\ln x}$, $y = x^5 + 6x^4 - \frac{1}{x}$.

4) $y = \frac{x^3 - 2x^2 - x + 2}{x}$, $y = \operatorname{ctg}\left(3x + \frac{\pi}{2}\right)$.

5) $y = (x^2 - 2)e^{-2x}$, $y = \frac{x^5 - 8x^4 + 3x - 5}{\sqrt{x}}$.

6) $y = \frac{x^3}{x^2 - 1}$, $y = \sin\left(5x - \frac{\pi}{2}\right)$.

7) $y = \sin x - \ln(\sin x)$, $y = e^{2x+3}$.

8) $y = \cos x + \frac{1}{2}\cos(2x) + \frac{1}{3}\cos(3x)$, $y = \sqrt[3]{7x^6 - 5}$.

9) $y = \sqrt[3]{x(7-x)^2} - \frac{x}{2}$, $y = \cos\left(2x + \frac{\pi}{2}\right)$.

10) $y = \left(\frac{x+2}{x-2}\right)^2$.

3. Побудувати гістограму та зробити оцінки розподілу даних. Побудувати гістограму з кривою густини, що показує нормальний розподіл.

- 1) Згенерувати 1000 випадкових чисел за нормальним розподілом, використовуючи функцію `rnorm()`.
- 2) Побудувати гістограму `hist()` та `ggplot2::geom_histogram()`.
- 3) Додати криву густини, використовуючи функцію `density()`.

4. Ознайомитися з побудовою кругових діаграм. Побудувати кругові діаграми.

- 1) Створити набір даних із категоріями ("A", "B", "C", "D") та значеннями.
 - 2) Побудувати кругову діаграму за допомогою функції `pie()`.
 - 3) Побудувати аналогічну кругову діаграму з використанням `ggplot2::coord_polar()`.
 - 4) Додати відсоткові підписи.
- ### 5. Візуалізація часових рядів.

- 1) Згенерувати набір даних для часових рядів, використовуючи функцію `ts()`.

- 2) Побудувати лінійний графік зміни значень у часі за допомогою `plot()`.
- 3) Використати `ggplot2::geom_line()` для візуалізації тих самих даних.

Питання для самоконтролю

1. Охарактеризуйте можливості для візуалізації у мові R.
2. Охарактеризуйте можливості бібліотеки `ggplot2`.
3. Розкажіть про способи побудови стовпчикових гістограм.
4. Розкажіть про способи побудови кругових діаграм.
5. Охарактеризуйте особливості візуалізації часових рядів.