

Лабораторна робота №4
Євчика Олексія, Інформатика, Магістр

Тема: Стовпчикові діаграми на основі аналізу декількох параметрів. Кругові діаграми в середовищі R

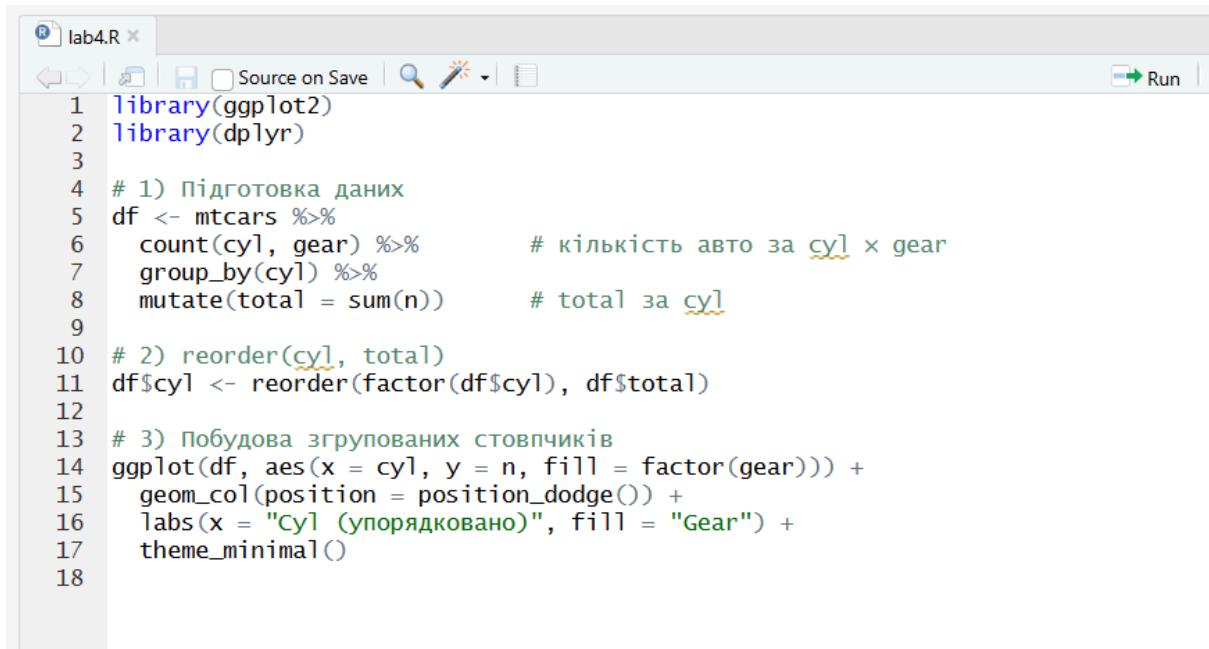
Варіант 6

Упорядкування категорій

Мета: Відсортувати категорії за підсумком.

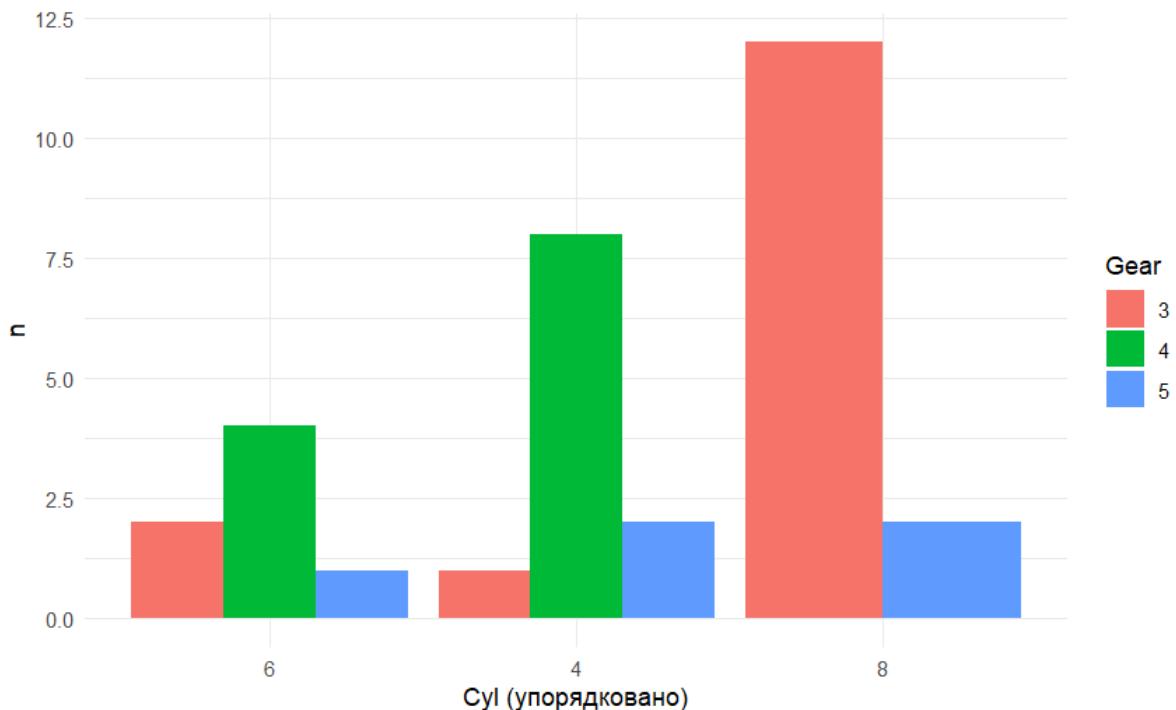
Кроки: 1) Додайте total = sum(n) за cyl. 2) reorder(cyl, total) у естетиці x. 3) Побудуйте згруповані стовпчики.

Очікуваний результат: Категорії впорядковані від меншого до більшого (або навпаки).



The screenshot shows the RStudio interface with a script file named "lab4.R". The code in the script is as follows:

```
1 library(ggplot2)
2 library(dplyr)
3
4 # 1) Підготовка даних
5 df <- mtcars %>%
6   count(cyl, gear) %>%
7   group_by(cyl) %>%
8   mutate(total = sum(n)) # total за cyl
9
10 # 2) reorder(cyl, total)
11 df$cyl <- reorder(factor(df$cyl), df$total)
12
13 # 3) Побудова згрупованих стовпчиків
14 ggplot(df, aes(x = cyl, y = n, fill = factor(gear))) +
15   geom_col(position = position_dodge())
16   labs(x = "Cyl (упорядковано)", fill = "Gear") +
17   theme_minimal()
```



Отриманий результат:

- категорії cyl відсортовані за total (від меншого до більшого або навпаки, залежно від reorder).
- побудовано згрупований стовпчикову діаграму, де порядок категорій відповідає сумарним значенням.

2. Практична частина

1. Підготуйте середовище (ggplot2, dplyr, tidyverse, scales).

```

library(ggplot2)
library(dplyr)
library(tidyverse)
library(scales)

```

так як в мене не було бібліотеки tidyverse, встановлю її командою

```

2 library(dplyr)
3 install.packages("tidyverse")
4 library(tidyverse)

```

2. Оберіть або згенеруйте набір даних із ≥ 2 категоріальними ознаками та однією кількісною (або рахуйте частоти).

Створюю дані:

Group – категорія

Category – друга категорія

Value – кількісна метрика

```
5
6   set.seed(123)
7
8   data <- tibble(
9     Group = rep(c("A", "B", "C"), each = 6),
10    Category = rep(c("X", "Y"), times = 9),
11    Value = round(runif(18, 10, 50), 1)
12  )
13
```

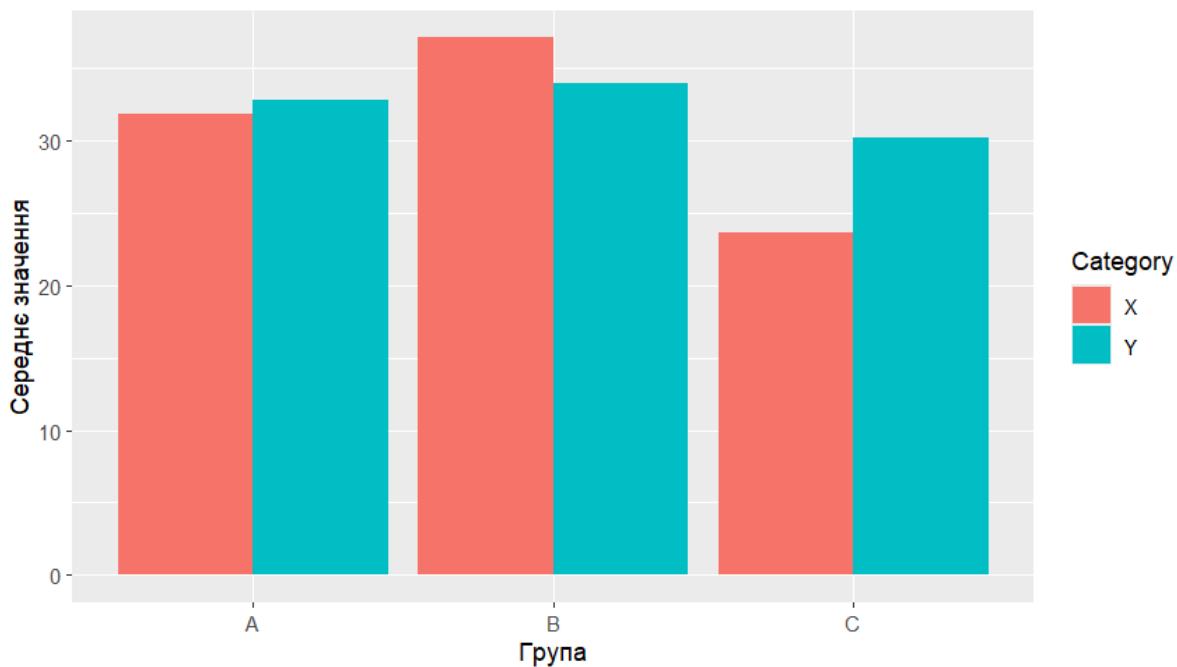
3. Виконайте агрегацію (наприклад, count, summarise) та/або перетворення до «довгого» формату (pivot_longer).

```
16
17   agg <- data %>%
18     group_by(Group, Category) %>%
19     summarise(
20       mean_value = mean(Value),
21       se = sd(Value) / sqrt(n()),
22       .groups = "drop"
23     )
24
25   agg
```

4. Побудуйте просту стовпчикову діаграму; перевірте підписи осей та легенд.

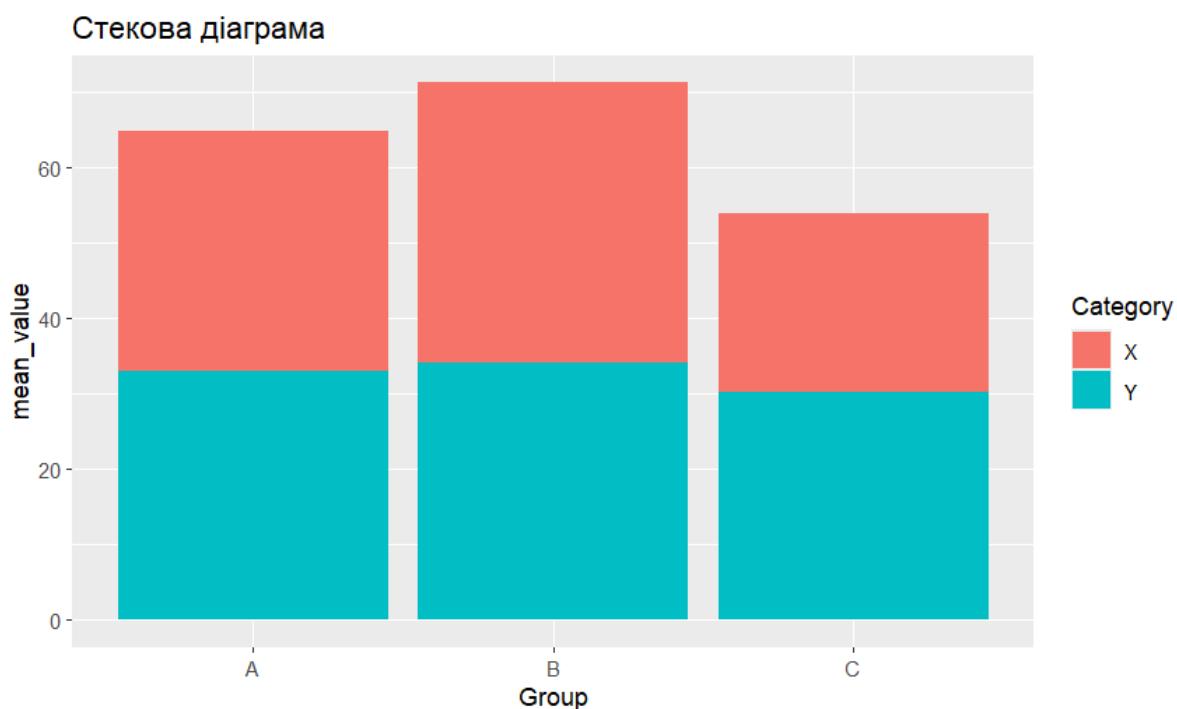
```
26
27   ggplot(agg, aes(x = Group, y = mean_value, fill = Category)) +
28     geom_col(position = "dodge") +
29     labs(
30       title = "Проста стовпчикова діаграма",
31       x = "Група",
32       y = "Середнє значення"
33     )
```

Проста стовпчикова діаграма



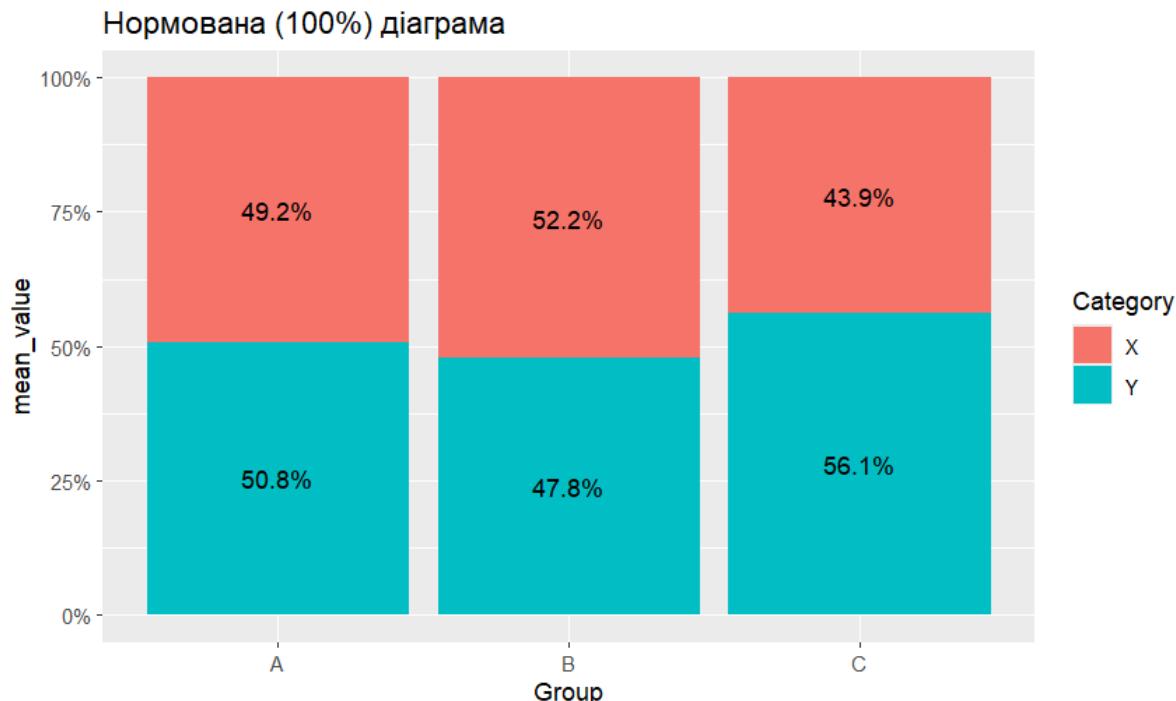
5. Побудуйте згруповани та стекову діаграми для двох категорій.

```
33
34 ggplot(agg, aes(x = Group, y = mean_value, fill = Category)) +
35   geom_col(position = position_dodge()) +
36   labs(title = "Згрупована діаграма")
37
38 ggplot(agg, aes(x = Group, y = mean_value, fill = Category)) +
39   geom_col(position = "stack") +
40   labs(title = "Стекова діаграма")
41 |
```



6. Побудуйте нормовану (100%) діаграму з відсотковою шкалою та підписами всередині.

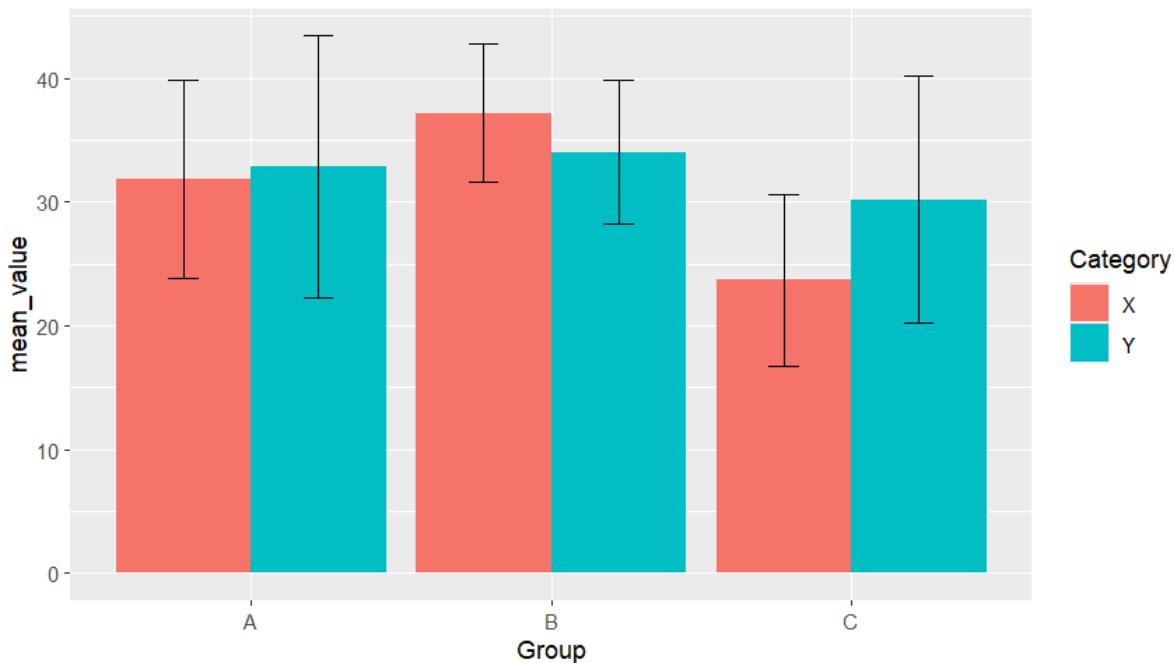
```
42 ggplot(agg, aes(x = Group, y = mean_value, fill = Category)) +  
43   geom_col(position = "fill") +  
44   geom_text(  
45     aes(label = percent(mean_value / tapply(mean_value, Group, sum)[Group])),  
46     position = position_fill(vjust = 0.5)  
47   ) +  
48   scale_y_continuous(labels = percent_format()) +  
49   labs(title = "Нормована (100%) діаграма")
```



7. Додайте похибки (SE/CI) для агрегованої метрики.

```
50  
51 ggplot(agg, aes(x = Group, y = mean_value, fill = Category)) +  
52   geom_col(position = position_dodge(width = 0.9)) +  
53   geom_errorbar(  
54     aes(ymin = mean_value - se, ymax = mean_value + se),  
55     position = position_dodge(width = 0.9),  
56     width = 0.2  
57   ) +  
58   labs(title = "Столбчики з похибками (SE)")
```

Столбчики з похибками (SE)



8. Застосуйте фасетування за третім параметром.

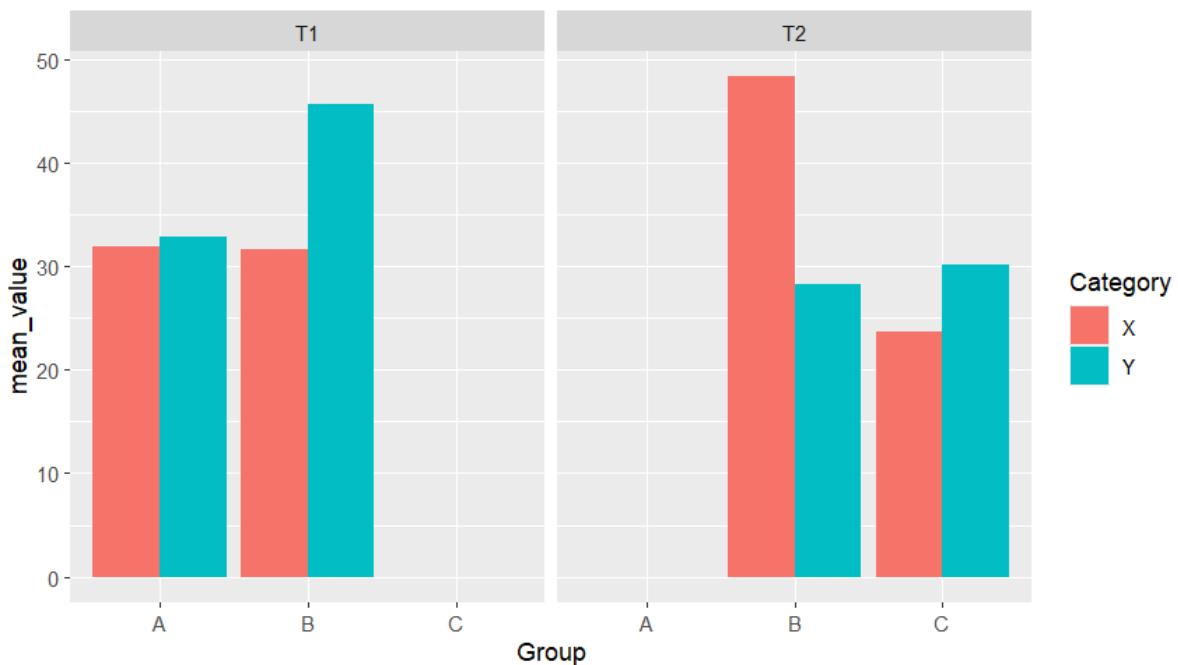
Введемо третю категорію:

```
59
60 data2 <- data %>%
61   mutate(Type = rep(c("T1", "T2"), each = 9))
62
63 agg2 <- data2 %>%
64   group_by(Type, Group, Category) %>%
65   summarise(mean_value = mean(value), .groups = "drop")
66
```

Фасети:

```
6/
68 ggplot(agg2, aes(x = Group, y = mean_value, fill = Category)) +
69   geom_col(position = "dodge") +
70   facet_wrap(~Type) +
71   labs(title = "фасетування за Type")
```

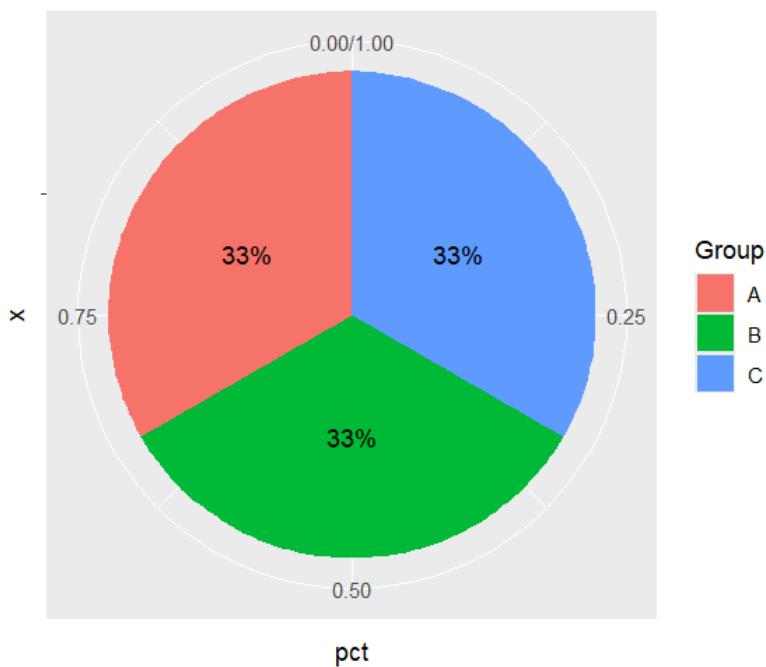
Фасетування за Type



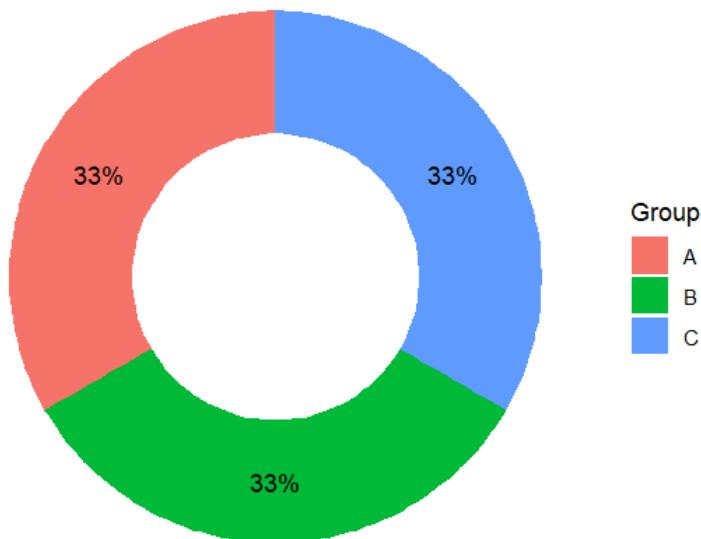
9. Побудуйте кругову та donut-діаграми для одного виміру з відсотками у підписах.

```
73 pie_data <- data %>%
74   count(Group) %>%
75   mutate(pct = n / sum(n))
76
77 ggplot(pie_data, aes(x = "", y = pct, fill = Group)) +
78   geom_col() +
79   coord_polar(theta = "y") +
80   geom_text(aes(label = percent(pct)), position = position_stack(vjust = 0.5)) +
81   labs(title = "Кругова діаграма")
82
83 ggplot(pie_data, aes(x = 2, y = pct, fill = Group)) +
84   geom_col() +
85   coord_polar(theta = "y") +
86   geom_text(aes(label = percent(pct)), position = position_stack(vjust = 0.5)) +
87   xlim(0.5, 2.5) +
88   labs(title = "Donut-діаграма") +
89   theme_void()
```

Кругова діаграма



Donut-діаграма



10. Порівняйте повідомлюваність та читабельність барів проти кругових.

Висновки:

- Стовпчикові діаграми краще показують порівняння величин, порядок та різницю між групами.
- Кругові діаграми важко читаються, особливо коли багато категорій.
- Donut-графіки — естетичні, але менш інформативні.

- Для точних порівнянь краще використовувати bar charts, а не pie charts.