

Лекція 2. Візуалізація та розвідковий аналіз даних в R

Данило Тавров

15.02.2023

Ідея сьогоднішньої лекції

- Ми розглянемо деякі принципи EDA та його основні складові
- Корисними матеріалами є:
 - Книжка *Exploratory Data Analysis with R* (PDF версія доступна на диску в каталогі з лекцією), особливо розділи 1, 4, 5, 6, 16
 - Відповідний [розділ](#) у книжці *R for Data Science*
 - Книжка *Exploratory Data Analysis Using R* (PDF версія доступна на диску в каталогі з лекцією), особливо розділи 3 і 9.6–9.7
 - [Курс з EDA](#) авторства Джима Алберта (Jim Albert)
 - Цей курс базується на класичній¹ книжці Джона Т'юкі (John Tukey) *Exploratory Data Analysis*, який є основоположником багатьох принципів EDA
- Проведенню EDA буде присвячено Лабораторну роботу №1
- Очікується, що студенти проведуть додатковий аналіз джерел та застосують методи за межами лекційного матеріалу

¹Хоча й доволі застарілій!

План лекції

- 1 Загальні міркування про EDA
- 2 Дескриптивний аналіз даних
- 3 Основи використання ggplot2
- 4 Візуалізація даних у рамках EDA

Що таке EDA (1)

- Як такого визначення EDA не існує
- Можна провести корисну паралель зі зйомками кіно
 - Як правило, знімаючи кіно, знімають багато матеріалу, і не весь буде використано в остаточному продакшенні
 - Понад те, сцени знімають не в тому порядку, у якому вони з'являться в остаточній версії
 - У монтажній кімнаті режисер та монтажер можуть експериментувати з різними варіантами сцен та відбирати найліпші версії
 - Слабкі сцени можуть викинути, а цікаві — розширити та взагалі перезняти
 - Такий монтаж проводять нашвидкуруч, щоб відразу зрозуміти, як рухатися далі
 - Тонкощі на кшталт світла чи графіки на цьому етапі не розглядають
- У цьому сенсі EDA подібний до роботи в монтажній кімнаті
- Він передує повноцінному аналізу з метою:
 - Перевірки наявних даних на помилки чи інші проблеми, які варто усунути (пропущені дані тощо)
 - Виявлення додаткових потреб у даних, які потрібно зібрати
 - Виявлення залежностей між змінними, які становлять цікавість або є неочікуваними
 - Пошуку свідчень на підтримку деяких гіпотез
- Отже EDA дає зрозуміти досліднику, на що можна звернути особливу увагу, а які ідеї не варто розвивати

Що таке EDA (2)

- EDA можна також порівняти з роботою детектива
 - Детектив шукає злочинця на підставі наявних доказів
 - Виконуючи EDA, ми намагаємося з'ясувати якісь закономірності на підставі наявних даних
 - Ми прагнемо підсумувати інформацію про дані та описати у відносно простий спосіб, про що вони нам кажуть
- EDA варто відрізняти від аналізу даних, спрямованого на підтвердження (чи спростування) гіпотез (confirmatory data analysis)
 - Ми цим будемо займатися майже весь інший час нашого курсу
- Аналіз гіпотез потребує застосування відповідних статистичних методів
- Його можна порівняти з судовим процесом
- Але якщо детектив не збере свідчень і не сформулює обвинувачення, суду взагалі не буде з чим працювати

Що таке EDA (3)

- EDA — це ітеративний процес, що передбачає:
 - Формулювання питання
 - Відповідь на нього за допомогою візуалізації, перетворення чи моделювання даних
 - Уточнення питання на основі здобутої інформації
- Це доволі творчий процес без чітко встановлених правил
- На початкових етапах потрібно розглянути будь-які ідеї, що спадають на думку
- Найліпші ідеї можуть бути предметом дальших досліджень у рамках інференційного, прогнозного або причиново-наслідкового аналізу даних

Що таке EDA (4)

- Набори даних, із якими доводиться працювати, дуже часто мають великий обсяг
- Їх можуть збирати люди чи організації, до яких у нас немає доступу і діяльність яких ми не можемо контролювати
- Навіть якщо дані збирає сам дослідник, він часто використовує спеціальні засоби на кшталт веб-скрейперів (web scrapers)
- Тому першим кроком в аналізі даних є ознайомлення з їхньою структурою, наявністю помилок, пропущених значень тощо
- Після цього в рамках EDA нас цікавлять відповіді на два широкі класи питань:
 - Якого роду варіація має місце в наших змінних? (Який розподіл?)
 - Якого роду **коваріація** має місце **між** нашими змінними?

Чотири принципи EDA

- Розгляньмо чотири принципи EDA², також відомі як *четири R*
- **Revelation** (відкриття): мається на увазі візуалізація даних як ключова складова EDA
- **Residuals** (залишки): мається на увазі важливість аналізу різниць між наявними даними та результатами застосування формальної «моделі»
 - У контексті EDA моделі дуже простенькі (наприклад, накладена зверху щільність нормального розподілу), їх не варто плутати з тими моделями, які ми вивчатимемо далі в нашому курсі
- **Reexpression** (трансформація): часто для того, щоб побачити щось корисне в даних, їх спочатку потрібно трансформувати (взяти логарифм, піднести до ступеня тощо)
- **Resistance** (стійкість): аналіз даних не повинен залежати від наявності викидів (дуже великих чи дуже малих значень, які вибиваються з загального розподілу)

²Velleman P. F., Hoaglin D. C. Data analysis. In: Hoaglin D. C., Moore D. S. (eds.), Perspectives on Contemporary Statistics, Mathematical Association of America (1991)

Основні кроки в рамках EDA

- Спочатку потрібно розібратися з основними характеристиками набору даних:
 - Скільки в ньому спостережень і змінних?
 - Які назви мають змінні, наскільки вони адекватні, чи варто їх перейменовувати?
 - Якого типу змінні, наскільки вони відповідають суті, чи варто їх перекодувати?
 - Скільки унікальних значень має кожна змінна? Які значення повторюються найчастіше?
 - Чи є пропущені дані? Наскільки багато? Чому?
- Дляожної змінної, яка становить інтерес, доцільно провести дескриптивний аналіз
- Для важливих змінних потрібно виконати візуалізацію як особливостей розподілуожної змінної окремо, так і взаємозв'язку між різними змінними
- Якщо в наборі даних наявні викиди чи інші аномалії, із ними потрібно розібратися

План лекції

- 1 Загальні міркування про EDA
- 2 **Дескриптивний аналіз даних**
- 3 Основи використання `ggplot2`
- 4 Візуалізація даних у рамках EDA

Огляд структури набору даних (1)

- Для початку роботи з набором даних доцільно ознайомитися з його структурою
- Використаймо знову [дані про пасажирів «Титаніку»](#)
- Після скачування відповідного файлу формату CSV та розміщення у відповідному каталогі, ми його зчитуємо та проглядаємо структуру

```
passengers <- read_csv("data/titanic.csv")
str(passengers, give.attr = FALSE)

## # spc_tbl_ [891 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## # $ PassengerId: num [1:891] 1 2 3 4 5 6 7 8 9 10 ...
## # $ Survived : num [1:891] 0 1 1 1 0 0 0 1 1 ...
## # $ Pclass : num [1:891] 3 1 3 1 3 3 1 3 3 2 ...
## # $ Name : chr [1:891] "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" ...
## # $ Sex : chr [1:891] "male" "female" "female" "female" ...
## # $ Age : num [1:891] 22 38 26 35 35 NA 54 2 27 14 ...
## # $ SibSp : num [1:891] 1 1 0 1 0 0 0 3 0 1 ...
## # $ Parch : num [1:891] 0 0 0 0 0 0 1 2 0 ...
## # $ Ticket : chr [1:891] "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## # $ Fare : num [1:891] 7.25 71.28 7.92 53.1 8.05 ...
## # $ Cabin : chr [1:891] NA "C85" NA "C123" ...
## # $ Embarked : chr [1:891] "S" "C" "S" "S" ...
```

Огляд структури набору даних (2)

- Можемо бачити, що в ньому 891 спостереження і 12 змінних
- Також бачимо типи всіх змінних
 - Змінні Sex та Embarked мають символічний формат, хоча перелік їхніх значень свідчить, що це по суті категорійні змінні

```
table(passengers$Sex)

##
##   female     male
##      314      577

table(passengers$Embarked)

##
##   C     Q     S
##  168    77  644
```

- Те саме можна сказати про «числові» змінні Survived і Pclass

```
table(passengers$Survived)

##
##   0     1
##  549  342

table(passengers$Pclass)

##
##   1     2     3
##  216  184  491
```

- Перетворімо ці змінні у фактори (а Pclass додатково — у впорядкований фактор)

```
passengers <- passengers %>% mutate(Sex = as.factor(Sex), Embarked = as.factor(Embarked),
                                         Survived = as.factor(Survived), Pclass = as.ordered(Pclass))
```

- Усі назви приблизно мають сенс (і принаймні не містять недозволених символів)
 - За потреби можна перейменовувати деякі назви типу SibSp чи Parch

Огляд структури набору даних (3)

- Також у наборі даних корисно подивитися, скільки для кожної змінної є пропущених значень

```
passengers %>% summarise(across(everything(), ~ sum(is.na(.)))) %>%
  select(where(~ all(.) > 0))

## # A tibble: 1 x 3
##       Age Cabin Embarked
##   <int> <int>     <int>
## 1    177     687        2
```

- Також може бути корисно проглянути частку пропущених значень

```
passengers %>% summarise(across(everything(), ~ mean(is.na(.)))) %>%
  select(where(~ all(.) > 0))

## # A tibble: 1 x 3
##       Age Cabin Embarked
##   <dbl> <dbl>     <dbl>
## 1 0.199 0.771    0.00224
```

- Як можна бачити, тільки 2 значення пропущено для змінної Embarked, але понад 77% даних пропущено в стовпчику Cabin
- Також майже 20% пропущено значень віку пасажирів (Age)
- Тому якщо рядки з пропущеними значеннями Embarked можна просто не враховувати (їх дуже мало і вони ні на що не впливають), то змінну Cabin, певно, навряд чи можна з користю застосувати
- Що стосується змінної Age, то потрібний детальніший аналіз

Огляд структури набору даних (4)

- Варто пам'ятати, що пропущені дані можуть кодувати у специфічний спосіб
- Наприклад, змінна може бути цілочисельна і не мати пропусків
- Але вона може мати значення -1 , 999 тощо, які варто сконвертувати в пропущені
- Перевірмо це, наприклад, для наших даних

```
table(passengers$Age)
```

```
## #  
## 0.42 0.67 0.75 0.83 0.92 1 2 3 4 5 6 7 8 9 10 11  
## 1 1 2 2 1 7 10 6 10 4 3 3 4 8 2 4  
## 12 13 14 14.5 15 16 17 18 19 20 20.5 21 22 23 23.5 24  
## 1 2 6 1 5 17 13 26 25 15 1 24 27 15 1 30  
## 24.5 25 26 27 28 28.5 29 30 30.5 31 32 32.5 33 34 34.5 35  
## 1 23 18 18 25 2 20 25 2 17 18 2 15 15 1 18  
## 36 36.5 37 38 39 40 40.5 41 42 43 44 45 45.5 46 47 48  
## 22 1 6 11 14 13 2 6 13 5 9 12 2 3 9 9  
## 49 50 51 52 53 54 55 55.5 56 57 58 59 60 61 62 63  
## 6 10 7 6 1 8 2 1 4 2 5 2 4 3 4 2  
## 64 65 66 70 70.5 71 74 80  
## 2 3 1 2 1 2 1 1
```

- Ми бачимо, що вік усюди лежить в адекватних межах, але вік деяких пасажирів указано з точністю до місяця
- Округлювати їх не обов'язково, але варто врахувати, що не всі є цілочисельними
- Пропущені дані можуть також ховатися у даних символового типу
 - Часто порожні рядки або рядки з суцільних пробілів насправді позначають пропущені дані
- Корисний приклад проведення аналізу структури даних можна почитати у відповідному розділі книжки *Exploratory Data Analysis*

Підсумкові характеристики чисової змінної (1)

- Найпростіше, що можна дізнатися про розподіл (чисової) змінної — це її підсумкові характеристики (summary)
- Це дасть змогу, у першу чергу, зрозуміти **типові** значення та загальні риси розподілу
- Ключовими характеристиками чисової змінної є її «середнє» значення так «розкид» значень навколо «середнього»
- Із теорії ймовірностей ми знаємо, що відповідні характеристики випадкової величини X мають назву **сподівання** (expectation) $\mathbb{E}[X]$ та **дисперсії** (variance) $\text{Var}(X)$
 - На практиці замість дисперсії ліпше використовувати **середньоквадратичне відхилення** (standard deviation) $\sigma_X = \sqrt{\text{Var}(X)}$
 - Воно має ті ж одиниці виміру, що й сама величина

Підсумкові характеристики числовової змінної (2)

- Нехай маємо вибірку $X_1, \dots, X_n, X_i \sim \mathbb{P}_X$, де \mathbb{P}_X — розподіл X
- Тоді оцінити сподівання й дисперсію можна за допомогою **середнього вибіркового** (sample mean)

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

та **вибіркової дисперсії** (sample variance)

$$S_X = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

- Проте ці вибіркові статистики є чутливі до викидів
- Тому часто основними підсумковими значеннями про величину вважають такі п'ять:

 - Мінімум
 - Перший квартиль
 - Медіана
 - Третій квартиль
 - Максимум

Визначення 2.1

Число M називають **медіаною** (median) випадкової величини X із розподілом \mathbb{P}_X , якщо

$$\mathbb{P}_X(X \leq M) \geq \frac{1}{2}, \quad \mathbb{P}_X(X \geq M) \geq \frac{1}{2}$$

□

- Визначення 2.1 медіані в теорії ймовірностей трішки складніше, ніж потрібно на практиці
- На практиці медіану вибірки розміром n визначають так:
- Якщо n парне, M дорівнює середньому елементу у відсортованій вибірці
- Якщо n непарне, M дорівнює середньоаритметичному двох середніх елементів

Визначення 2.2

- Число x_p називають **p -им квантилем** (p th quantile) випадкової величини X із розподілом \mathbb{P}_X , якщо

$$\mathbb{P}_X(X \leq x_p) \geq p, \quad \mathbb{P}_X(X \geq x_p) \geq 1 - p$$

- $x_{0.25}$ називають **першим квартилем** (first quartile)
- $x_{0.5}$ — це і є медіана
- $x_{0.75}$ називають **третім квартилем** (third quartile)



- На практиці існують різні підходи до підрахунку квартилів:
 - Функція `summary` рахує їх згідно з вищеведеним визначенням
 - Функція `stats::fivenum` рахує їх як медіани відповідних інтервалів
- Відстань між першим і третім квартилями називають **міжквартильним розмахом** (interquartile range, IQR)
 - На відміну від середньоквадратичного відхилення (standard deviation), IQR стійкий до викидів
 - Інтерпретація проста: в IQR потрапляють (приблизно) середніх 50% даних

Підсумкові характеристики числової змінної (5)

- Наприклад, можемо обчислити всі середні та середньоквадратичні відхилення для всіх (числових) змінних

```
bind_rows(  
  mean = passengers %>% summarize(across(where(is.numeric), mean, na.rm = TRUE)),  
  sd = passengers %>% summarize(across(where(is.numeric), sd, na.rm = TRUE)),  
  .id = "statistic"  
)  
  
## # A tibble: 2 x 6  
##   statistic PassengerId  Age SibSp Parch  Fare  
##   <chr>       <dbl> <dbl> <dbl> <dbl>  
## 1 mean        446    29.7  0.523  0.382  32.2  
## 2 sd          257.   14.5  1.10   0.806  49.7
```

- Ми вказали na.rm = TRUE, інакше в стовпчику Age були б NA замість результату
- Аналогічно можна порахувати будь-які інші статистики

Підсумкові характеристики числової змінної (6)

- Також можна використати функцію `summary`

```
summary(passengers %>% select(where(is.numeric)))
```

	PassengerId	Age	SibSp	Parch
## Min.	1.0	Min. : 0.42	Min. :0.000	Min. :0.0000
## 1st Qu.	223.5	1st Qu.:20.12	1st Qu.:0.000	1st Qu.:0.0000
## Median	446.0	Median :28.00	Median :0.000	Median :0.0000
## Mean	446.0	Mean :29.70	Mean :0.523	Mean :0.3816
## 3rd Qu.	668.5	3rd Qu.:38.00	3rd Qu.:1.000	3rd Qu.:0.0000
## Max.	891.0	Max. :80.00	Max. :8.000	Max. :6.0000
## NA's		NA's :177		
##				
## Fare				
## Min.	0.00			
## 1st Qu.	7.91			
## Median	14.45			
## Mean	32.20			
## 3rd Qu.	31.00			
## Max.	512.33			
##				

- На перший погляд нічого особливого у вічі не кидається
- Такого роду інформацію корисно візуалізувати, до чого ми повернемося пізніше

Підсумкові характеристики числової змінної (7)

- Класичним прикладом³, що ілюструє обмеженість тільки дескриптивних статистик, є датафрейм anscombe із пакета datasets

```
bind_rows(  
  mean = anscombe %>% summarize(across(everything(), mean, na.rm = TRUE)),  
  sd = anscombe %>% summarize(across(everything(), sd, na.rm = TRUE)),  
  median = anscombe %>% summarize(across(everything(), median, na.rm = TRUE)),  
  .id = "statistic"  
)  
  
##   statistic      x1      x2      x3      x4      y1      y2      y3  
## 1      mean 9.000000 9.000000 9.000000 9.000000 7.500909 7.500909 7.500000  
## 2      sd 3.316625 3.316625 3.316625 3.316625 2.031568 2.031657 2.030424  
## 3      median 9.000000 9.000000 9.000000 8.000000 7.580000 8.140000 7.110000  
##      y4  
## 1 7.500909  
## 2 2.030579  
## 3 7.040000
```

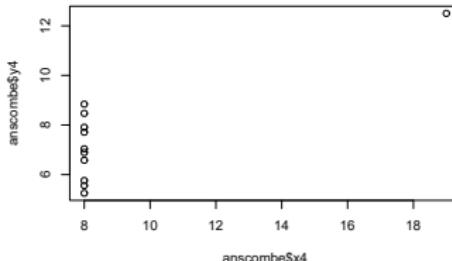
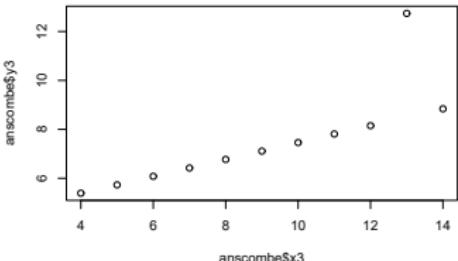
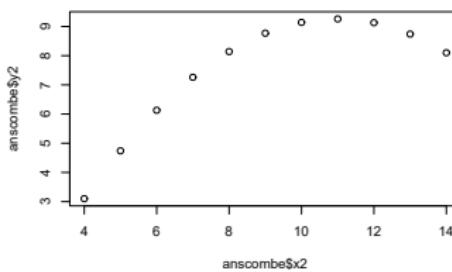
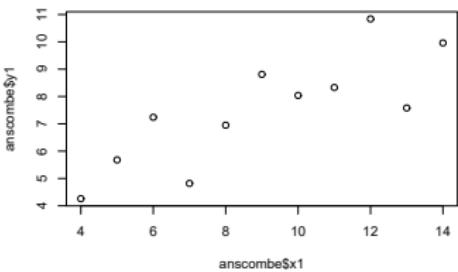
- Як можна помітити, середні та середньоквадратичні відхилення (але не медіани) дуже подібні як для змінних x, так і для змінних y

³Anscombe F. J. Graphs in statistical analysis, The American Statistician 27, 17–21 (1973)

Підсумкові характеристики чисової змінної (8)

- Проте графічні зображення свідчать про посутні відмінності між цими наборами даних

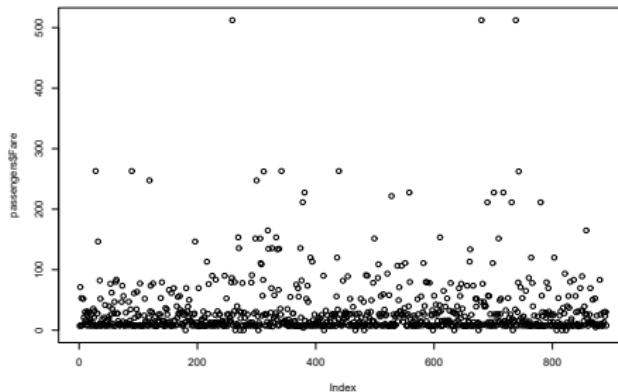
```
par(mfrow = c(2, 2))
plot(anscombe$x1, anscombe$y1)
plot(anscombe$x2, anscombe$y2)
plot(anscombe$x3, anscombe$y3)
plot(anscombe$x4, anscombe$y4)
```



Викиди в числових змінних (1)

- Кажучи неформально, **викид** (outlier) — це таке значення змінної, яке несумісне з розподілом інших
- Викиди важко визначити формально, але дуже добре видно на графіках

```
plot(passengers$Fare)
```



- У вічі кидаються три викиди, пов'язані з дуже високими цінами на квитки
- Наявність таких викидів спотворює статистичні характеристики вибірки
- Раніше ми порахували, що середнє змінної Age дорівнює 29.70, а медіана — 28
- У той же час середнє змінної Fare дорівнює 32.20, а медіана — 14.45
- Тобто наявність великих викидів сильно завищила середнє значення
- Використання даних у статистичних моделях вестиме до хибних висновків

Викиди в числових змінних (2)

- Існують різні методи, як можна визначити викиди в деякій вибірці $\mathbf{x} = (x_1, \dots, x_n)$, проте жоден не є ідеальним
- Якщо вибірка походить із нормального розподілу (тобто вибірка є реалізацією $\mathbf{X} = (X_1, \dots, X_n)$, $X_i \stackrel{\text{i.i.d.}}{\sim} N(\mu, \sigma^2)$), то можна застосувати **правило трьох сигм** (the three sigma rule)
 - Тобто вважати викидом будь-яке значення $x \notin [\bar{x} - 3\sqrt{S_x}; \bar{x} + 3\sqrt{S_x}]$
 - Звісно, такий підхід погано працюватиме, якщо дані містять викиди, адже сподівання буде спотворене!
- Стійкіший метод полягає в тому, щоб вважати за викид будь-яке спостереження $x \notin [M - 1.5 \cdot \text{IQR}; M + 1.5 \cdot \text{IQR}]$
- Іще ліпший підхід полягає в застосуванні так званого **фільтру Гампеля** (Hampel filter)⁴
 - Викидом є будь-яке спостереження $x \notin [M - 3 \cdot \text{MAD}; M + 3 \cdot \text{MAD}]$
 - Тут MAD — **медіанне абсолютне відхилення** (median absolute deviation):

$$\text{MAD} = \frac{1}{\Phi^{-1}(0.75)} \cdot \text{median}(\mathbf{x} - M) \approx 1.4826 \cdot \text{median}(\mathbf{x} - M)$$

- (Відповідний коефіцієнт потрібний для того, щоб $\mathbb{E}(\text{MAD}) = \sigma$ для нормальної вибірки)

⁴Швейцарський математик Франк Гампель (Frank Hampel, 1941–2018)

Викиди в числових змінних (3)

- Застосування до наших даних дає

```
passengers %>% filter(Fare < median(Fare) - 3*mad(Fare) | Fare > median(Fare) + 3*mad(Fare)) %>%  
  arrange(desc(Fare))  
  
## # A tibble: 171 x 12  
##   PassengerId Survived Pclass Name     Sex   Age SibSp Parch Ticket   Fare Cabin  
##   <dbl> <fct>   <ord>  <chr>   <fct> <dbl> <dbl> <dbl> <chr>   <dbl> <chr>  
## 1      259 1       1      "Ward~ fema~ 35     0     0  PC 17~ 512. <NA>  
## 2      680 1       1      "Card~ male  36     0     1  PC 17~ 512. B51 ~  
## 3      738 1       1      "Lesu~ male  35     0     0  PC 17~ 512. B101  
## 4      28  0       1      "Fort~ male  19     3     2  19950 263  C23 ~  
## 5      89  1       1      "Fort~ fema~ 23     3     2  19950 263  C23 ~  
## 6      342 1       1      "Fort~ fema~ 24     3     2  19950 263  C23 ~  
## 7      439 0       1      "Fort~ male  64     1     4  19950 263  C23 ~  
## 8      312 1       1      "Ryer~ fema~ 18     2     2  PC 17~ 262. B57 ~  
## 9      743 1       1      "Ryer~ fema~ 21     2     2  PC 17~ 262. B57 ~  
## 10     119 0       1      "Baxt~ male  24     0     1  PC 17~ 248. B58 ~  
## # ... with 161 more rows, and 1 more variable: Embarked <fct>
```

- Як можна бачити, ця процедура доволі жорстка, адже викидів виявлено доволі багато

Викиди в числових змінних (4)

- Можна сформулювати такі загальні рекомендації
- Застосувати різні методи виявлення викидів і порівняти відповідні результати
- Застосувати логічне мислення та знання предметної області, щоб визначити, чи є формально виявлені викиди такими насправді
- Проаналізувати відповідні графіки на предмет виявлення справді великих викидів, на які потрібно звернути увагу

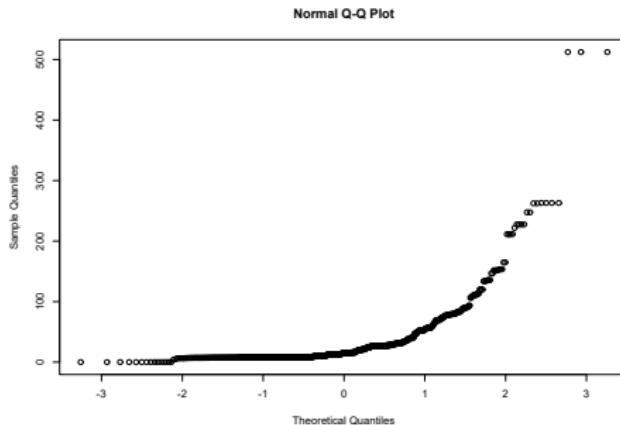
Викиди в числових змінних (5)

- Корисний (візуальний) інструментарій для виявлення викидів полягає в побудові так званого **Q-Q-графіка** (Q-Q plot, quantile-quantile plot)
- Якщо просто, ідея полягає в тому, щоб на одній осі розмістити квантилі одного розподілу, а на іншій — другого
- Дуже часто на осі абсцис розміщують теоретичні квантилі (стандартного) нормальногорозподілу
- Тоді якщо на графіку ми бачимо (приблизно) пряму лінію, ми робимо висновок, що дані мають «приблизно нормальній» розподіл
- Також ці графіки дають змогу побачити викиди

Викиди в числових змінних (6)

- Побудуймо відповідний графік для змінної `Fare`

```
qqnorm(passengers$Fare)
```



- По-перше, ми бачимо, що розподіл не є нормальний (про це — далі)
- Але також ми чітко бачимо три великих викиди і декілька дивних значень, близьких до нуля

Викиди в числових змінних (6)

- Викиди в загальному випадку можуть бути як результатом помилок під час уведення даних, так і особливостями самої вибірки
- Подивімось, наприклад, на пасажирів із нульовою ціною квитка

```
passengers %>% filter(Fare == 0)

## # A tibble: 15 x 12
##   PassengerId Survived Pclass Name   Sex   Age SibSp Parch Ticket  Fare Cabin
##   <dbl> <fct>   <ord>  <chr> <fct> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
## 1     180 0      3      "Leon~ male   36    0     0  LINE   0 <NA>
## 2     264 0      1      "Harr~ male   40    0     0  112059 0 B94
## 3     272 1      3      "Torn~ male   25    0     0  LINE   0 <NA>
## 4     278 0      2      "Park~ male   NA    0     0  239853 0 <NA>
## 5     303 0      3      "John~ male   19    0     0  LINE   0 <NA>
## 6     414 0      2      "Cunn~ male   NA    0     0  239853 0 <NA>
## 7     467 0      2      "Camp~ male   NA    0     0  239853 0 <NA>
## 8     482 0      2      "Fros~ male   NA    0     0  239854 0 <NA>
## 9     598 0      3      "John~ male   49    0     0  LINE   0 <NA>
## 10    634 0      1      "Parr~ male   NA    0     0  112052 0 <NA>
## 11    675 0      2      "Wats~ male   NA    0     0  239856 0 <NA>
## 12    733 0      2      "Knig~ male   NA    0     0  239855 0 <NA>
## 13    807 0      1      "Andr~ male   39    0     0  112050 0 A36
## 14    816 0      1      "Fry, ~ male   NA    0     0  112058 0 B102
## 15    823 0      1      "Reuc~ male   38    0     0  19972 0 <NA>
## # ... with 1 more variable: Embarked <fct>
```

- За цими даними неможливо чітко стверджувати, що нульова ціна є помилковою, адже інші значення цілком адекватні
- Відповідне питання потребує детального аналізу, у тому числі зовнішніх джерел
 - Можливо, справді були пасажири, які пройшли безкоштовно?
- В інших випадках може бути очевидно, що дані явно неправильні
 - Не там стоїть кома, зайвий нуль і тому подібне

Інші проблеми з даними

- Поширеною є ситуація, коли змінна є нібіто числовою і дійсною, але окремі значення зустрічаються **надто** часто
 - Це може свідчити про використання якихось кодів на позначення пропущених даних
 - Також це може бути пов'язано з особливостями збору даних, коли замість невідомого значення вказують якесь значення за замовчуванням
- Наприклад, розгляньмо датафрейм Boston із пакета MASS (інформація про деякі будинки в передмістях Бостона⁵)

```
table(MASS::Boston$tax)
```

```
## # 187 188 193 198 216 222 223 224 226 233 241 242 243 244 245 247 252 254 255 256
## # 1 7 8 1 5 7 5 10 1 9 1 2 4 1 3 4 2 5 1 1
## # 264 265 270 273 276 277 279 280 281 284 285 287 289 293 296 300 304 305 307 311
## # 12 2 7 5 9 11 4 1 4 7 1 8 5 3 8 7 14 4 40 7
## # 313 315 329 330 334 335 337 345 348 351 352 358 370 384 391 398 402 403 411 422
## # 1 2 6 10 2 2 2 3 2 1 2 3 2 11 8 12 2 30 2 1
## # 430 432 437 469 666 711
## # 3 9 15 1 132 5
```

- Як можна бачити, змінна `tax` (податок, у доларах, на кожні 10 000 долларів повної вартості будинку) набуває різних значень
- Але значення 666 вона набуває особливо часто
 - Це може бути просто збіг обставин
 - Або це може свідчити про особливості кодування даних тощо

⁵Harrison D., Rubinfeld D. L. Hedonic prices and the demand for clean air. Journal of Environmental Economics and Management (5), 81–102 (1978)

Таблиці спряженості (1)

- Якщо маємо справу з категорійними змінними, то корисним способом їх відображення є **таблиця спряженості** (contingency table)
- У цій таблиці рядки та стовпці відповідають категоріям двох змінних
- В R можна підрахувати таблицю спряженості за допомогою функції `xtab`

```
cont_tab <- xtabs(~ Survived + Sex, data = passengers)
cont_tab
```

```
##           Sex
## Survived female male
##          0     81  468
##          1    233  109
```

- Першим аргументом функції є так звана **формула**
 - Ми їх часто зустрічатимемо в нашому курсі в різних контекстах
- Тут справа від `~` стоять змінні, які потрібно зобразити в таблиці
- Зліва може стояти змінна, у якій містяться значення, які потрібно сумувати за категоріями
 - За замовчуванням рахують просто загальне число спостережень

Таблиці спряженості (2)

- Дуже корисним є додавати в таблицю спряженості **маржинальні розподіли** (marginal distributions)

```
addmargins(cont_tab)

##          Sex
## Survived female male Sum
##      0      81  468 549
##      1     233  109 342
##     Sum     314  577 891
```

- Можна побудувати маржинальні розподіли окремо за рядками (margin = 1) та стовпцями (margin = 2)

```
margin.table(cont_tab, margin = 1)

## Survived
## 0 1
## 549 342

margin.table(cont_tab, margin = 2)

## Sex
## female male
## 314 577
```

Таблиці спряженості (3)

- Можна перетворити таблицю спряженості в таблицю емпіричного спільного розподілу:

```
joint_dist <- prop.table(cont_tab)
joint_dist

##           Sex
## Survived   female     male
##        0 0.09090909 0.52525253
##        1 0.26150393 0.12233446

sum(joint_dist)

## [1] 1
```

- А вказавши аргумент `margin`, можна збудувати ще й умовні розподіли

```
cond_dist_given_row <- prop.table(cont_tab, margin = 1)
cond_dist_given_row

##           Sex
## Survived   female     male
##        0 0.1475410 0.8524590
##        1 0.6812865 0.3187135
```

- Наприклад, 68.13% жінок вижило **серед усіх тих, хто вижив**

Таблиці спряженості (4)

- Також можна будувати таблиці спряженості з декількома змінними

```
ftable(xtabs(~ Survived + Sex + Pclass, data = passengers))
```

		Pclass	1	2	3	
#	# Survived	Sex	female	3	6	72
			male	77	91	300
#	1	Sex	female	91	70	72
			male	45	17	47

- Функція `ftable` сплющає (flattens) багатовимірну таблицю у приємний формат

План лекції

- 1 Загальні міркування про EDA
- 2 Дескриптивний аналіз даних
- 3 Основи використання `ggplot2`
- 4 Візуалізація даних у рамках EDA

- Як і в попередній лекції, метою не є вичерпний огляд можливостей пакету `ggplot2` для візуалізації даних
- Ми розглянемо найважливіші види графіків та засоби їх побудови
- Корисну інформацію можна дістати з книжки *ggplot2: Elegant Graphics for Data Analysis*
 - PDF версія доступна на диску (у каталозі з літературою), але видання різні, тому й структура різна
 - Особливу увагу варто звернути на такі розділи: 1, 2, 3–4 (по діагоналі), 8, 11, 15, 18
- **Прекрасним джерелом** є [галерея графіків](#) із відповідними поясненнями та кодом, як їх будувати
- Додаткову інформацію про можливості `ggplot2` можна дізнатися [на офіційному сайті](#)
- Інші корисні джерела:
 - [R Graphics Cookbook](#): набір конкретних прикладів
 - Відповідний розділ із книжки [Modern Dive](#)
 - Українською можна подивитися відео від [І. Мірошниченка](#)
 - Корисною є шпаргалка (cheat sheet) за [посиланням](#), її також викладено на диск
- Також завжди можна звернутися до сайтів [Stack Overflow](#) та [Stack Exchange](#)

Що таке ggplot2

- Пакет `ggplot2` є одним із базових пакетів `tidyverse`
- Літери `gg` розшифровуються як *grammar of graphics*⁶
- Побудова графіка здійснюється пошарово:
 - Спочатку вказуються дані в охайному форматі
 - Потім додаються параметри «естетики» (aesthetics) графіка (що стоїть на осіх координат, колір, розмір, форма тощо)
 - Потім зазначається «геометрія» графіка (geometry), тобто якими геометричними фігурами потрібно відобразити дані (точки, лінії, стовпчики тощо)
 - Поверх цього уточнюється координатна система, підписи осей, легенда тощо

⁶Wilkinson L. The Grammar of Graphics, Springer-Verlag, New York (2005)

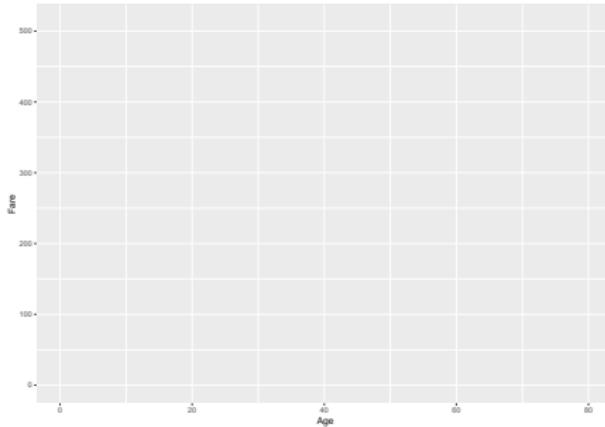
Дуже коротко про граматику графіків

- Розглянемо базові поняття, які будемо використовувати
- Графіки будуються на основі **даних** та **відображення** (mapping), яке показує, як змінні з датафрейму відображаються на атрибути естетики
- Існує 5 основних компонентів відображення:
 - **layer**: набір геометричних елементів та статистичних перетворень
 - **geom**: точки, лінії, багатокутники тощо
 - **stat**: побудова гістограми, оцінювання параметрів прямої за методом найменших квадратів тощо
 - **scale**: ставить у відповідність дані до їхнього візуального вигляду (колір, форма точки, товщина лінії тощо), також відповідає за легенду та осі
 - **coord**: описує, як дані розташовані на графіку (включає в себе осі та координатні сітки). Як правило, ми використовуємо прямокутову систему за замовчуванням
 - **theme**: управляє розміром шрифту, фоновим кольором тощо
 - **facet**: уточнює, як на одному графіку зобразити декілька панелей

Перший графік

- Використовуючи датафрейм про пасажирів «Титаніку», можемо побудувати простий графік:

```
ggplot(passengers, aes(x = Age, y = Fare))
```



- Використано такі аргументи функції `ggplot`:
 - Перший аргумент — завжди датафрейм
 - Другий аргумент — параметри естетики (у нашому випадку: яка змінна стоїть на якій координатній осі)

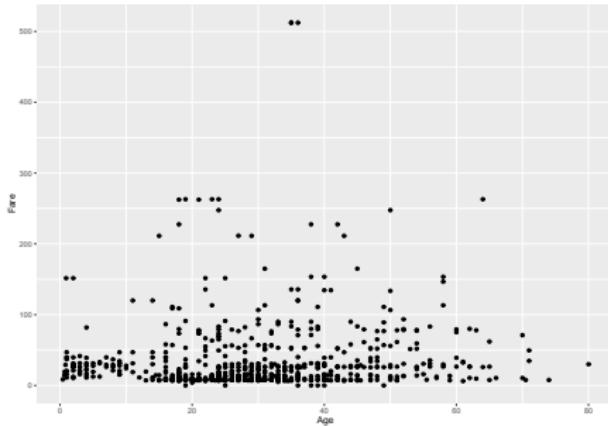
Додавання геометрії в графік

- Але побудований графік нічого не показує!
- Це тому, що ми тільки вказали естетичні параметри графіка
- Додатково потрібно задати геометрію графіка: усі відповідні функції починаються з `geom`
 - `geom_point` використовують для побудови **діаграм розсіювання** (scatter plots)
 - `geom_line` використовують для побудови кривих (дуже корисно для зображення часових рядів)
 - `geom_bar` використовують для побудови **стовпчастих діаграм** (bar plots)
 - `geom_boxplot` використовують для побудови **коробкових графіків** (box plots)
 - `geom_histogram` використовують для побудови гістограм
 - ... і багато інших

Scatter plots

- Збудуємо перший scatter plot

```
ggplot(passengers, aes(x = Age, y = Fare)) +  
  geom_point()  
  
## Warning: Removed 177 rows containing missing values (`geom_point()`).
```

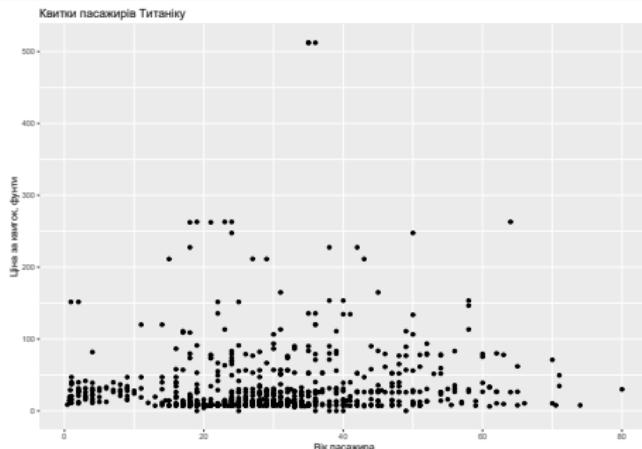


- Кожного пасажира зображенено окремою точкою
- Функція `geom_point` у даному випадку не має власних аргументів
- Це тому, що вона **наслідує** як датафрейм, так і атрибути естетики з виклику функції `ggplot`
- Що цікавого видно на графіку?

Заголовок графіка та підписи осей

- Для того, щоб графік мав привабливіший вигляд, можна уточнити параметри його відображення
- Для початку можна додати заголовок графіка та підписати осі координат

```
ggplot(passengers, aes(x = Age, y = Fare)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти",  
       title = "Квитки пасажирів Титаніку")
```

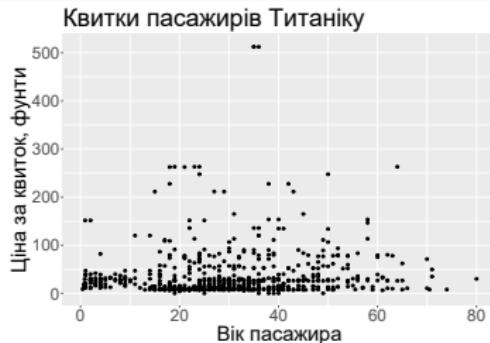


- Очевидно, розмір шрифту для осей недостатній

Збільшення розміру шрифту

- Для збільшення розміру шрифту потрібно використати функцію `theme`

```
ggplot(passengers, aes(x = Age, y = Fare)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти",  
       title = "Квитки пасажирів Титаніку") +  
  theme(text = element_text(size = 30))
```

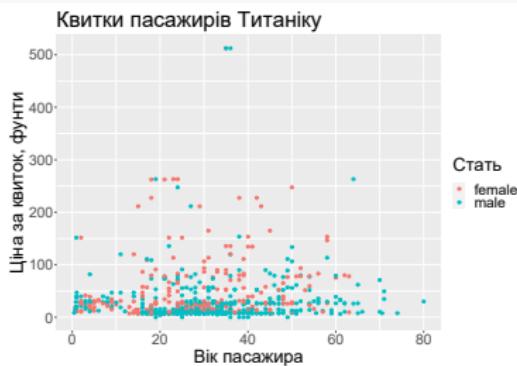


- Можна змінювати шрифти:
 - Окремо для міток на осіх (замість `text` указати `axis.text`)
 - Окремо для назв осей (замість `text` указати `axis.title`)
 - Окремо для заголовку графіка (замість `text` указати `plot.title`)
 - Окремо для елементів легенди (замість `text` указати `legend.text`)
 - Окремо для заголовку легенди (замість `text` указати `legend.title`)
- Можна змінювати не тільки розмір (`size`), а й колір (`color`), сам шрифт (`family`), задавати жирність або курсив (`face`) тощо

Додавання кольору (1)

- Часто корисним є зображення спостережень із різних класів різними кольорами
- У нашому випадку доцільним є використання різних кольорів для статей
- Для колоризації потрібно вказати додатковий параметр естетики

```
ggplot(passengers, aes(x = Age, y = Fare, color = Sex)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти",  
       title = "Квитки пасажирів Титаніку", color = "Стать") +  
  theme(plot.title = element_text(size = 30),  
        axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

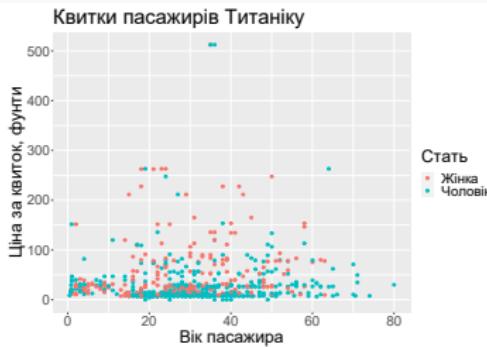


- Можна помітити, що серед власників найдешевших квитків чоловіків більше
- Зверніть увагу, що ми вказали назву «Стать» для легенди

Додавання кольору (2)

- Назви ключів легенди нас не задовольняють: потрібні україномовні відповідники

```
ggplot(passengers, aes(x = Age, y = Fare, color = Sex)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти",  
       title = "Квитки пасажирів Титаніку", color = "Стать") +  
  scale_color_discrete(labels = c("Жінка", "Чоловік")) +  
  theme(plot.title = element_text(size = 30),  
        axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

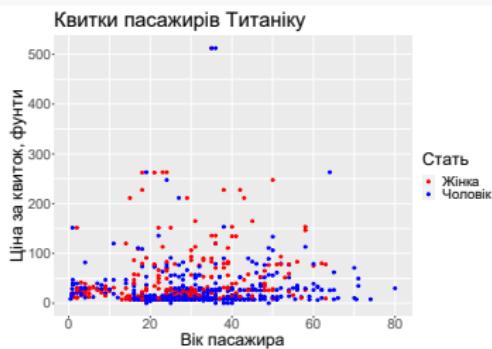


- Ми використали функцію, яка починається з префікса `scale_`
- Як і інші аналогічні, вона відображає дані в простір естетики (тут, колір)
- `scale_color_discrete` є за замовчуванням для категорійних змінних
- Ми тільки уточнили перелік назв ключів за допомогою аргументу `labels`

Додавання кольору (3)

- Якщо ми хочемо власноруч задати кольори, це можна зробити так:

```
ggplot(passengers, aes(x = Age, y = Fare, color = Sex)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти",  
       title = "Квитки пасажирів Титаніку", color = "Стать") +  
  scale_color_manual(values = c("red", "blue"),  
                     labels = c("Жінка", "Чоловік")) +  
  theme(plot.title = element_text(size = 30),  
        axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



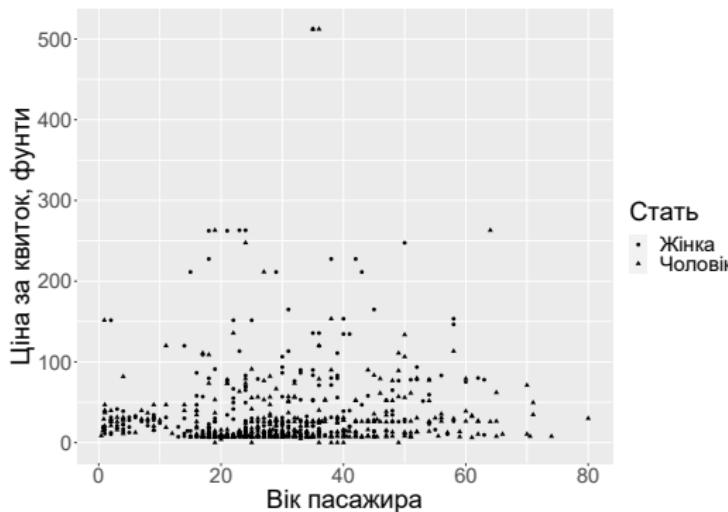
- Вектор значень кольорів впорядковано за значеннями факторної змінної

```
levels(passengers$Sex)  
## [1] "female" "male"
```

Модифікація інших атрибутів естетики (1)

- Можна уточнити форму зображення точок

```
ggplot(passengers, aes(x = Age, y = Fare, shape = Sex)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти", shape = "Стать") +  
  scale_shape_discrete(labels = c("Жінка", "Чоловік")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

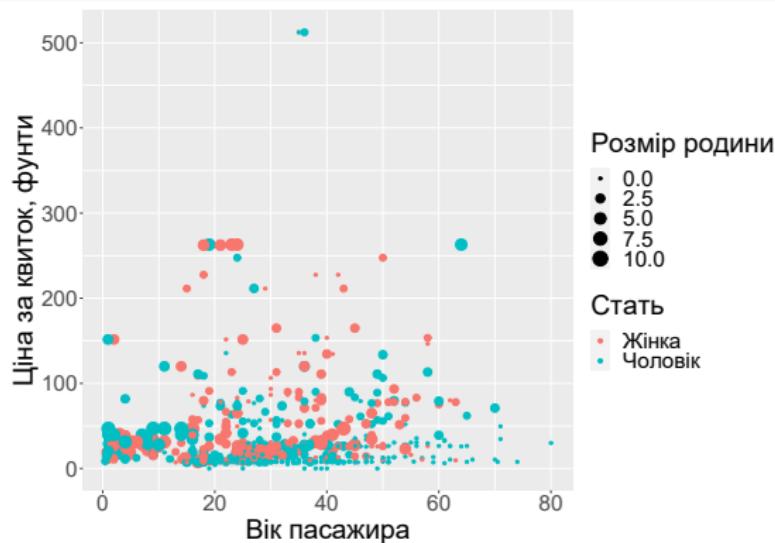


- Ми використали `scale_shape_discrete`, який іде за замовчуванням, але уточнили ключі легенди

Модифікація інших атрибутів естетики (2)

- Нарешті, можна керувати розміром точок
- Наприклад, поставити у відповідність розмір точки до числа членів родини

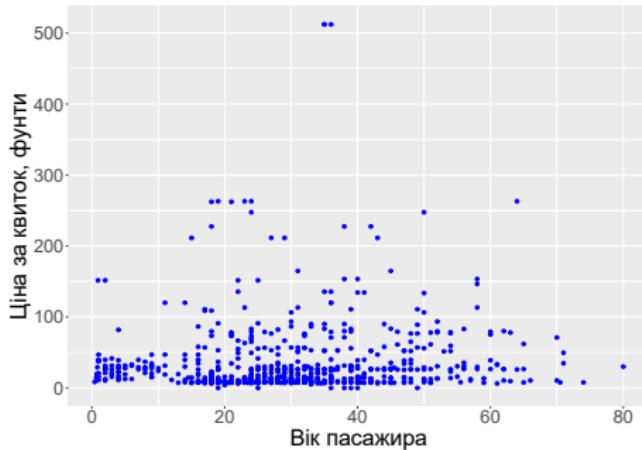
```
ggplot(passengers, aes(x = Age, y = Fare, color = Sex, size = Parch + SibSp)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти",  
       color = "Стать", size = "Розмір родини") +  
  scale_color_discrete(labels = c("Жінка", "Чоловік")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



Задавання атрибутів естетики для всього графіка

- Інколи є потреба задати колір, розмір тощо єдиний для всього графіка
- Тоді відповідні параметри можна **винести за межі** відповідної естетики, в окремий шар

```
ggplot(passengers, aes(x = Age, y = Fare)) +  
  geom_point(color = "blue") +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



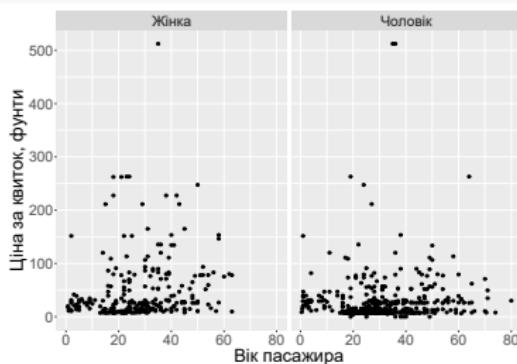
Особливості використання атрибутів естетики

- Різні атрибути естетики варто використовувати по-різному, залежно від типу зображуваних змінних
- Використання різних кольорів чи форм доцільно для категорійних змінних, де перелік можливих варіантів обмежений
- Для неперервних змінних кориснішим є використання розміру, оскільки це також неперервна величина
- Потрібно зважати на загальну інтерпретовність графіка
- Якщо графік стає перевантаженим, ліпше розбити його на окремі панелі

Фацетовані графіки (1)

- Наприклад, можна створити два окремі графіки — для чоловіків і для жінок
- Точніше, це будуть **панелі** одного графіка, причому координатні осі будуть однакові
- Такі графіки називають **фацетованими** (faceted)

```
ggplot(passengers, aes(x = Age, y = Fare)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        strip.text = element_text(size = 20)) +  
  facet_wrap(~Sex, labeller = as_labeller(c("female" = "Жінка", "male" = "Чоловік")))
```

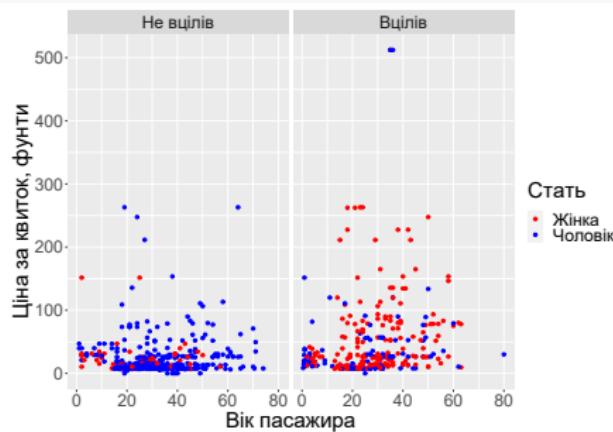


- Аргументом `facet_wrap` є формула
- Також ми додали додатковий аргумент для україномовних заголовків

Фацетовані графіки (2)

- Також можна створити фацетований графік, на якому різні панелі відповідатимуть статусу вцілості пасажирів

```
ggplot(passengers, aes(x = Age, y = Fare, color = Sex)) +  
  geom_point() +  
  labs(x = "Вік пасажира", y = "Ціна за квиток, фунти", color = "Стать") +  
  scale_color_manual(values = c("red", "blue"),  
                     labels = c("Жінка", "Чоловік")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        strip.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20)) +  
  facet_wrap(~Survived, labeller = as_labeller(c("0" = "Не вцілів", "1" = "Вцілів")))
```

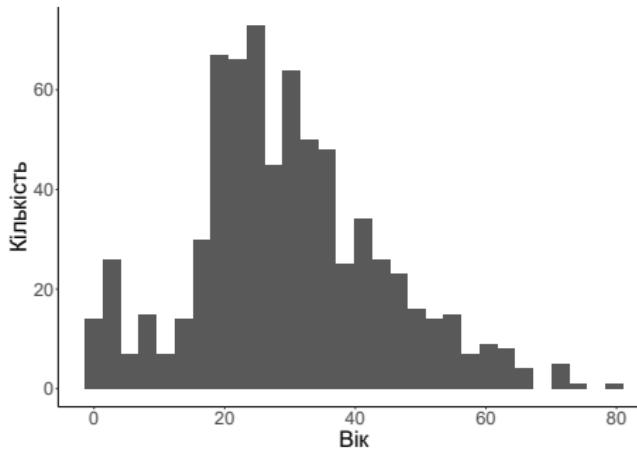


- Стає очевидним, що серед уцілілих пасажирів переважають жінки

Модифікація загального вигляду графіка (1)

- Ми використовували стандартну theme із `ggplot2` (`theme_grey`)
- Єдине, що ми дозволяли собі міняти — це розміри шрифтів
- Окрім шрифтів, міняти можна колір фону, параметри сітки і багато іншого
- Щоб не міняти кожний елемент окремо, існує низка готових themes
- Наприклад, класична тема

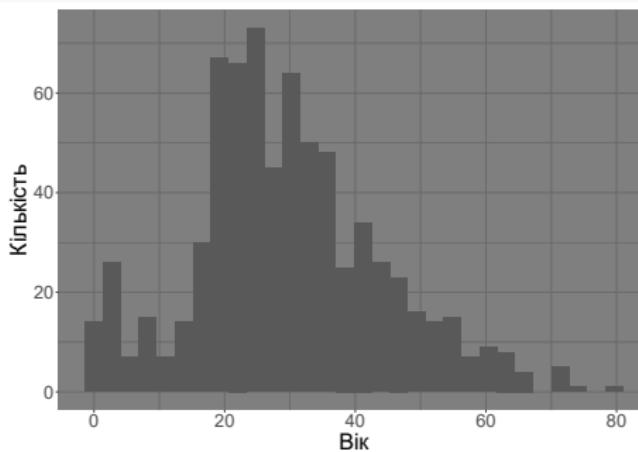
```
ggplot(passengers, aes(x = Age)) +  
  geom_histogram() +  
  labs(x = "Вік", y = "Кількість") +  
  theme_classic() +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



Модифікація загального вигляду графіка (2)

- Або темна

```
ggplot(passengers, aes(x = Age)) +  
  geom_histogram() +  
  labs(x = "Вік", y = "Кількість") +  
  theme_dark() +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



- Перелік інших цікавих тем (у т.ч. з інших пакетів) можна знайти [тут](#)

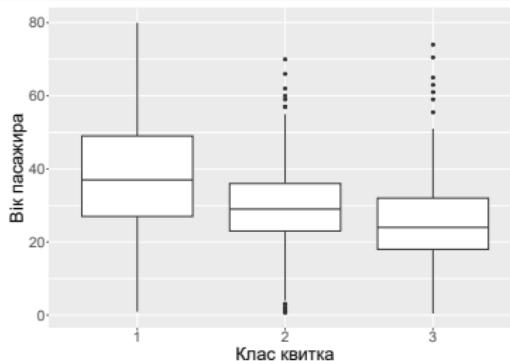
План лекції

- 1 Загальні міркування про EDA
- 2 Дескриптивний аналіз даних
- 3 Основи використання ggplot2
- 4 Візуалізація даних у рамках EDA

Box plots (1)

- Візуально п'ять основних підсумкових статистик про вибірку можна зобразити за допомогою box plots
- Можемо зобразити вікові розподіли пасажирів «Титаніка» за класом квитка

```
ggplot(passengers, aes(x = Pclass, y = Age)) +  
  geom_boxplot() +  
  labs(x = "Клас квитка", y = "Вік пасажира") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



- Box plot показує п'ять основних дескриптивних статистик:
 - Мінімум — нижня точка нижнього вуса
 - Перший квартиль — нижня границя скриньки
 - Медіана — товста лінія всередині скриньки
 - Третій квартиль — верхня границя скриньки
 - Максимум — верхня точка верхнього вуса
- Висота скриньки дорівнює IRQ

Box plots (2)

- Розгляньмо дані про учасниць бостонського марафону 2001 р.⁷
 - Подивімось на перші декілька рядків та структуру

⁷ Дані з курсу Джима Алберта

Box plots (3)

- Можна помітити, що перші рядки присвячено жінкам віком 20 р.
- Розгляньмо інші значення

```
table(boston_marathon$age)
```

```
##  
## 20 30 40 50 60  
## 22 25 25 25 11
```

- Тепер можемо бачити, що в датафреймі є жінки тільки 5 вікових категорій

Box plots (4)

- Можемо обчислити п'ять підсумкових статистик для часу для 20-річних учасниць

```
summary(boston_marathon %>% filter(age == 20) %>% .$time)
```

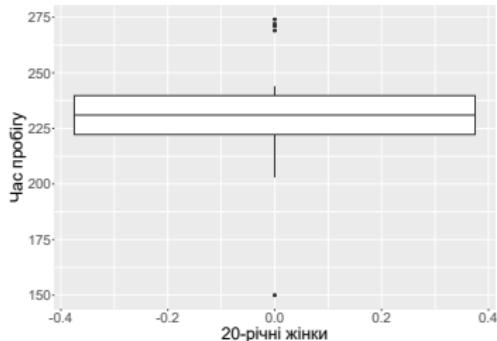
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
# #	150.0	222.2	231.0	231.5	239.8	274.0

- (Функція `summary` для вектора також додатково рахує середнє аритметичне)
- Як можна бачити, «типовим» значенням є час у 231 хв.
 - (Приблизно) половина жінок пробігла швидше, і (приблизно) половина — повільніше
- Також робимо висновок, що
 - (Приблизно) чверть жінок пробігла швидше від 222.2 хв.
 - (Приблизно) чверть жінок пробігла від 222.2 до 231 хв. тощо
- Медіана не сильно відрізняється від середнього вибіркового, тому розподіл може бути більш-менш симетричний

Box plots (5)

- Візуально це можна зобразити за допомогою box plot

```
ggplot(boston_marathon %>% filter(age == 20), aes(y = time)) +  
  geom_boxplot() +  
  labs(x = "20-річні жінки", y = "Час пробігу") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```

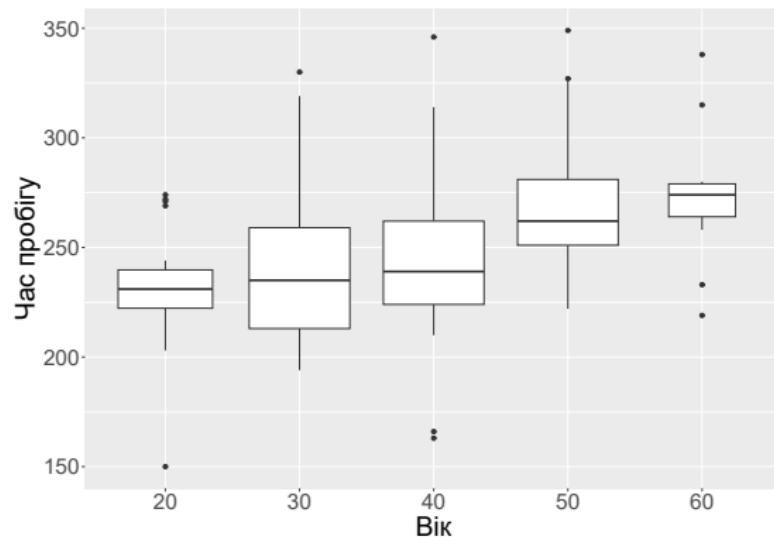


- Як можна бачити, розподіл все ж таки не є дуже симетричним, адже вуса мають різну довжину
- Також наявні викиди: одна жінка бігла дуже швидко, декілька — дуже повільно
- Загалом, можна сформулювати такі принципи:
 - Для симетричного розподілу медіана розташована посередині коробки
 - Вуса мають приблизно однакову довжину
 - Якщо розподіл скошено вправо, то медіана розташована нижче, а верхній вус довший
 - Якщо розподіл скошено вліво, то медіана розташована вище, а нижній вус довший

Box plots (6)

- Особливо корисним є використання box plots для швидкого порівняння розподілів різних категорій спостережень
- Побудуймо box plots для жінок різних вікових категорій

```
ggplot(boston_marathon, aes(x = factor(age), y = time)) +  
  geom_boxplot(varwidth = TRUE) +  
  labs(x = "Вік", y = "Час пробігу") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



- Параметр `varwidth = TRUE` робить різні скриньки різної ширини, залежно від кількості спостережень у відповідній групі

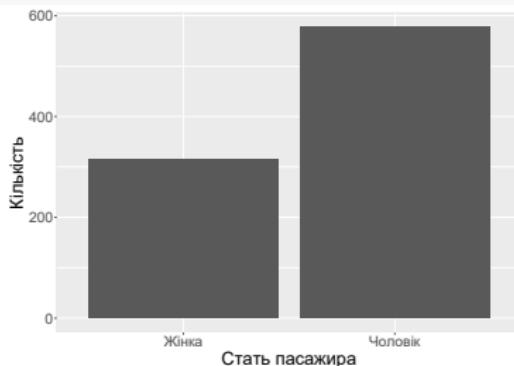
Box plots (7)

- Можна зробити такі висновки:
 - Зі збільшенням віку збільшується медіана часу пробігу
 - Впадають у вічі викиди в кожній віковій категорії
 - Три середні вікові категорії мають приблизно одинаковий IRQ
 - 20- і 60-річні жінки мають приблизно одинаковий IRQ
 - 20-, 30- і 40-річні жінки мають приблизно одну медіану, тоді як 50- і 60-річні біжать (у середньому) повільніше
- Щоправда, на основі таких візуальних спостережень не варто робити серйозних висновків
- Різниця між віковими категоріями може виявитися статистично незначущою (на це впливає розмір вибірки, форма розподілу тощо)
- Зв'язок між різними змінними не варто сприймати як причиново-наслідковий без додаткового аналізу

Bar plots (1)

- Якщо ми хочемо візуально оцінити весь розподіл, ми можемо:
 - Для категорійних змінних — збудувати bar plot
 - Для неперервних змінних — збудувати гістограму
- Для того, щоб збудувати bar plot, достатньо вказати тільки атрибут естетики x

```
ggplot(passengers, aes(x = Sex)) +  
  geom_bar() +  
  labs(x = "Стать пасажира", y = "Кількість") +  
  scale_x_discrete(labels = c("Жінка", "Чоловік")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



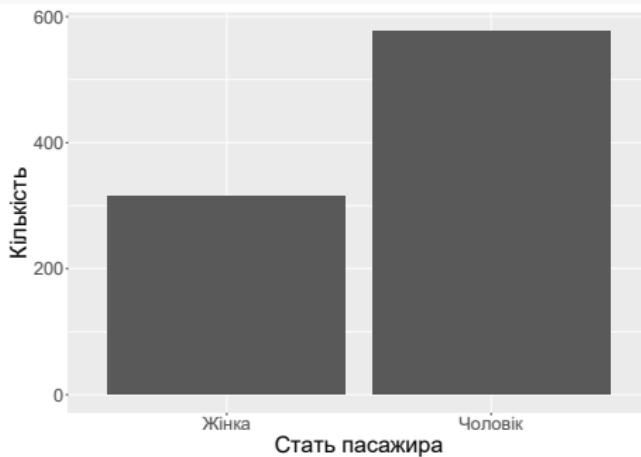
- Для зміни підписів на графіку ми використали `scale_x_discrete`, яка керує віссю абсцис
- Якщо в `aes` натомість указати `y = Sex`, то стовпчики будуть горизонтальні

Bar plots (2)

- За замовчуванням `geom_bar` рахує кількість спостережень у кожній категорії
- Якщо ж потрібно просто вивести на графік деякі числа у стовпчастому форматі, то потрібно вказати у та додатковий аргумент `stat = "identity"`

```
passengers_grouped <- passengers %>% group_by(Sex) %>% summarize(Total = n())

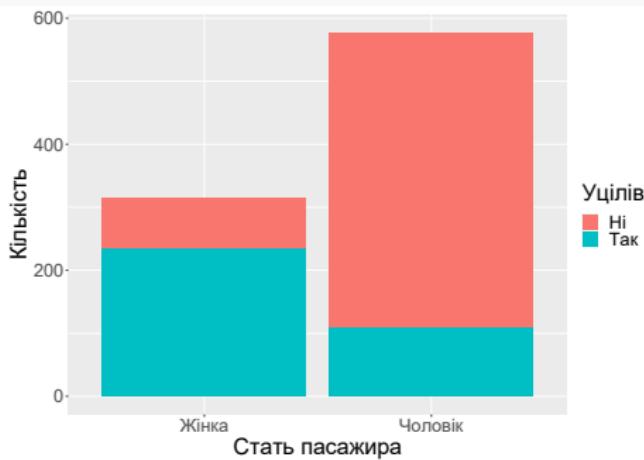
ggplot(passengers_grouped, aes(x = Sex, y = Total)) +
  geom_bar(stat = "identity") +
  labs(x = "Стать пасажира", y = "Кількість") +
  scale_x_discrete(labels = c("Жінка", "Чоловік")) +
  theme(axis.title = element_text(size = 25),
        axis.text = element_text(size = 20))
```



Bar plots (3)

- За допомогою bar plots можна відображати спільний розподіл двох категорійних змінних (факторів)
- Для цього потрібно вказати атрибут естетики fill

```
ggplot(passengers, aes(x = Sex, fill = Survived)) +  
  geom_bar() +  
  labs(x = "Стать пасажира", y = "Кількість", fill = "Уцілів") +  
  scale_x_discrete(labels = c("Жінка", "Чоловік")) +  
  scale_fill_discrete(labels = c("Hi", "Tak")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

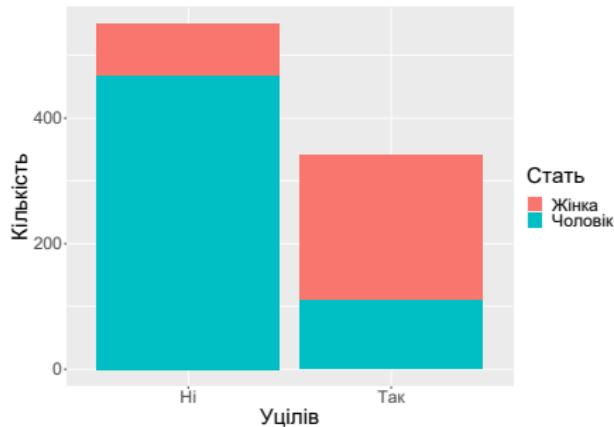


- Такі bar plots називають складеними (stacked)
- Як можна бачити, серед жінок вижила значно більша частка

Bar plots (4)

- Це, звісно, можна показати і в іншому порядку

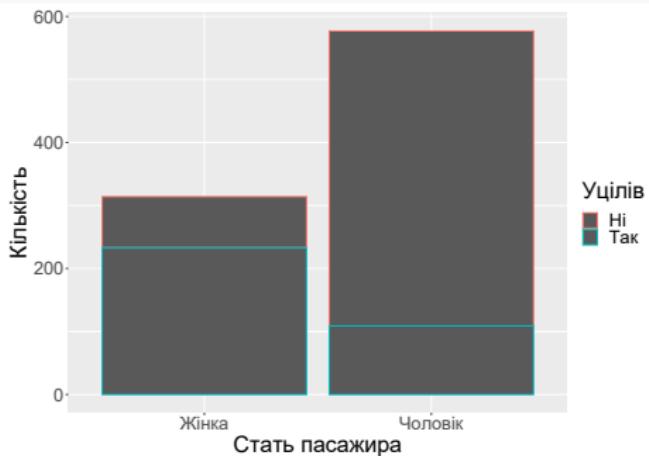
```
ggplot(passengers, aes(x = Survived, fill = Sex)) +  
  geom_bar() +  
  labs(x = "Уцілів", y = "Кількість", fill = "Стать") +  
  scale_x_discrete(labels = c("Ні", "Так")) +  
  scale_fill_discrete(labels = c("Жінка", "Чоловік")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



Bar plots (5)

- Варто звернути увагу, що нас цікавить саме атрибут `fill`, а не `color`
- `color` відповідає за колір сторін прямокутиків, і тому є зовсім нецікавим

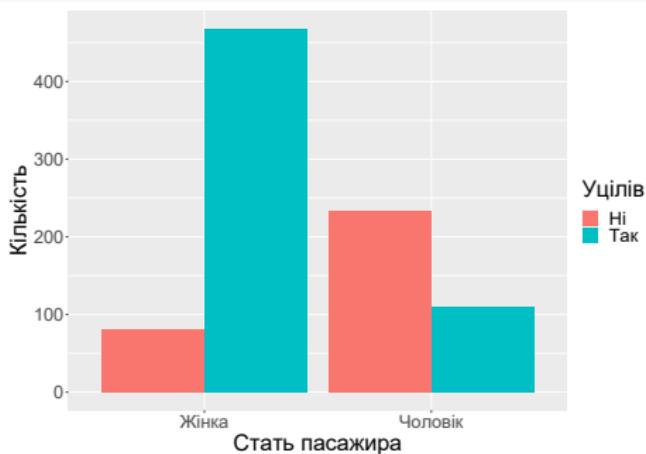
```
ggplot(passengers, aes(x = Sex, color = Survived)) +  
  geom_bar() +  
  labs(x = "Стать пасажира", y = "Кількість", color = "Уцілів") +  
  scale_x_discrete(labels = c("Жінка", "Чоловік")) +  
  scale_color_discrete(labels = c("Hi", "Tak")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



Bar plots (6)

- Альтернативою складеним є так звані згруповани (dodged) діаграми
- Щоб змінити тип діаграми, потрібно уточнити аргумент `position`
- За замовчуванням він дорівнює `stack`

```
ggplot(passengers, aes(x = Survived, fill = Sex)) +  
  geom_bar(position = "dodge") +  
  labs(x = "Стать пасажира", y = "Кількість", fill = "Уцілів") +  
  scale_x_discrete(labels = c("Жінка", "Чоловік")) +  
  scale_fill_discrete(labels = c("Ні", "Так")) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



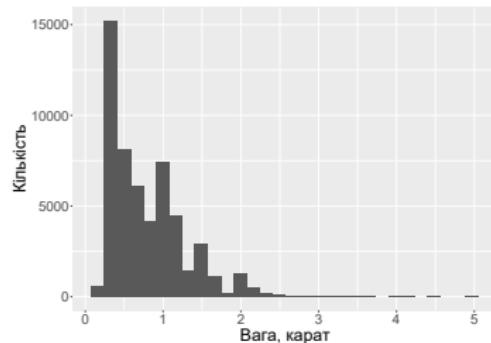
Гістограми (1)

- Класичним способом зображення розподілу *неперервних* даних є гістограма
 - Розгляньмо відомий [набір даних про діаманти](#)
 - Він містить інформацію про ціни та інші характеристики майже 54 000 діамантів

Гістограми (2)

- Збудуємо гістограму для ваги діамантів

```
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram() +  
  labs(x = "Вага, карат", y = "Кількість") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))  
  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- За замовчуванням використовується 30 інтервалів (**bins**, дослівно «сміттєвий бак»), тобто `geom_histogram(bins = 30)`
- Як правило, потрібно власноруч підбирати або число інтервалів (`bins =`), або їхню ширину (`binwidth =`)

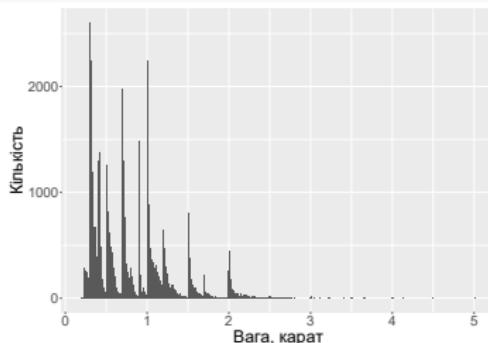
Гістограми (3)

- Гістограма дає змогу швидко зрозуміти, наскільки симетричним або скощеним є розподіл у вибірці
 - Як можна бачити з гістограми, розподіл скошений управо
- Інші особливості, на які може звернути увагу гістограма:
 - Викиди (дуже малі або великі значення)
 - Пропуски в даних
 - Наявність декількох кластерів (багатомодальні розподіли)
 - Інші особливості на кшталт нагромадження (bunching)

Гістограми (4)

- Оскільки даних доволі багато, 30 інтервалів явно недостатньо

```
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram(bins = 500) +  
  labs(x = "Вага, карат", y = "Кількість") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



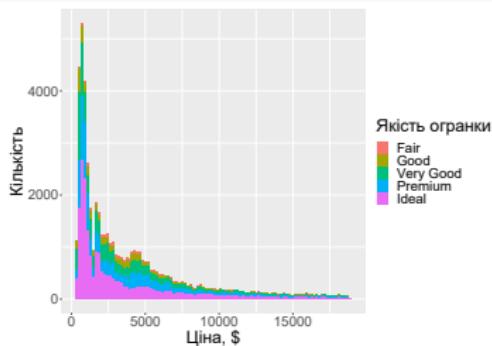
- Відразу виникають додаткові питання:
 - Чому вага зосереджена навколо цілих значень каратів та деяких простих дробів?
 - Чому справа від кожного локального максимуму діамантів більше, ніж зліва?
 - Чому так мало діамантів мають вагу більше 3 каратів?
- Маємо справу з декількома *підгрупами* (кластерами) спостережень
- Можна поставити додаткові питання:
 - Чи схожі інші параметри діамантів у рамках одного кластера?
 - Чи відмінні інші параметри діамантів **між** різними кластерами?
 - Чим пояснюється така відмінність?

Гістограми (5)

- На одному графіку можна зображати декілька гістограм для порівняння

```
diamonds <- diamonds %>%
  mutate(cut = factor(cut, levels = c("Fair", "Good", "Very Good", "Premium", "Ideal")))

ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram(bins = 100) +
  labs(x = "Ціна, $", y = "Кількість", fill = "Якість огранки") +
  theme(axis.title = element_text(size = 25),
        axis.text = element_text(size = 20),
        legend.title = element_text(size = 25),
        legend.text = element_text(size = 20))
```

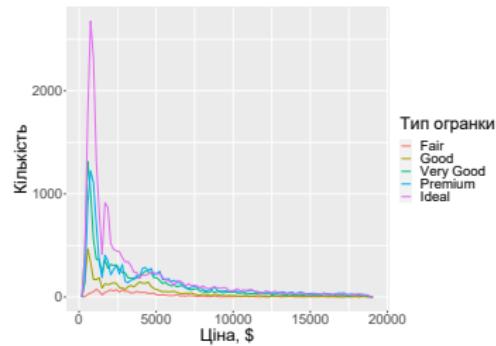


- Ми перетворили змінну `cut` у факторну і вказали правильний порядок категорій (а не за алфавітом)

Гістограми (6)

- Альтернативно, для спрощення зображення, можна зображати гістограми лінійними графіками

```
ggplot(diamonds, aes(x = price, color = cut)) +  
  geom_freqpoly(bins = 100) +  
  labs(x = "Ціна, $", y = "Кількість", color = "Тип огранки") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



Гістограми (7)

- Проте на такому графіку мало що можна побачити, адже в різних категоріях різна кількість спостережень

```
diamonds %>% group_by(cut) %>% summarize(total = n())
```

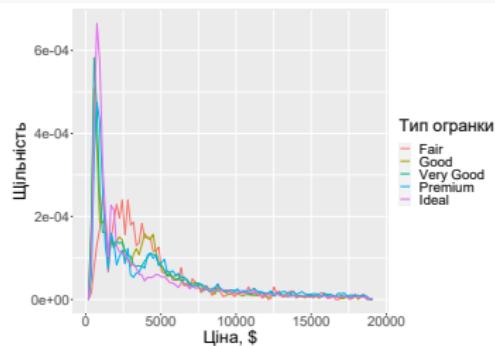
```
## # A tibble: 5 x 2
##   cut     total
##   <fct>   <int>
## 1 Fair     1610
## 2 Good     4906
## 3 Very Good 12082
## 4 Premium   13791
## 5 Ideal     21551
```

- Тоді доречно не просто рахувати кількість спостережень в інтервалах, а будувати апроксимацію **щільності** (density) розподілу
- Тобто нормалізувати висоту стовпчиків так, щоб сума площ дорівнювала 1

Гістограми (8)

- У нашому контексті це можна зробити за допомогою параметра естетики `y = after_stat(density)`

```
ggplot(diamonds, aes(x = price, y = after_stat(density), color = cut)) +  
  geom_freqpoly(bins = 100) +  
  labs(x = "Ціна, $", y = "Щільність", color = "Тип огранки") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

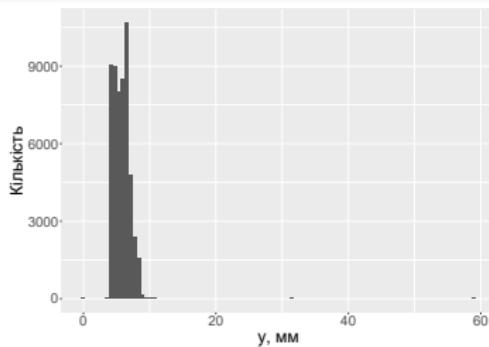


- Тепер порівняння можна здійснювати безпосередньо
- Зокрема, можна зробити висновок, що в середньому діаманти з гіршою огранкою мають вищу ціну!
 - Вочевидь, є інші фактори, які впливають на ціну (наприклад, розмір каменя, його вага)
- Взагалі всі розподіли доволі скошені

Гістограми і викиди (1)

- Розгляньмо гістограму параметра y («ширина») діаманта

```
ggplot(diamonds, aes(x = y)) +  
  geom_histogram(bins = 100) +  
  labs(x = "y, мм", y = "Кількість") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```

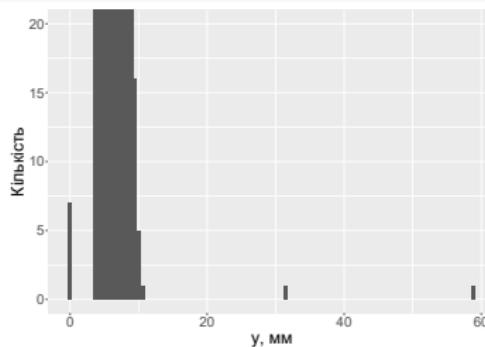


- На такій гістограмі важко побачити викиди, хоча вони є (інакше б стільки місця не пустувало)

Гістограми і викиди (2)

- Можна звузити вікно перегляду за віссю ординат:

```
ggplot(diamonds, aes(x = y)) +  
  geom_histogram(bins = 100) +  
  labs(x = "y, мм", y = "Кількість") +  
  coord_cartesian(ylim = c(0, 20)) +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



- Тепер чітко видно, що є три специфічні значення, які можна вважати викидами
 - Діаманти зі значенням $y = 0$ не можуть існувати в принципі, і тут очевидно має місце помилка в самих даних
 - Також сумнівними є діаманти з показником y , що перевищує 20 мм

Гістограми і викиди (3)

• Подивімось, що це за діаманти

```
diamonds %>%
  filter(y < 3 | y > 20) %>%
  arrange(y)

## # A tibble: 9 x 11
##   carat cut     color clarity depth table price     x     y     z
##   <dbl> <dbl> <fct>   <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 11.964 1 Very Good H     VS2    63.3   53  5139   0     0     0
## 2 15.952 1.14 Fair     G     VS1    57.5   67  6381   0     0     0
## 3 24.521 1.56 Ideal    G     VS2    62.2   54 12800   0     0     0
## 4 26.244 1.2 Premium  D     VVS1   62.1   59 15686   0     0     0
## 5 27.430 2.25 Premium H     SI2    62.8   59 18034   0     0     0
## 6 49.557 0.71 Good    F     SI2    64.1   60 2130    0     0     0
## 7 49.558 0.71 Good    F     SI2    64.1   60 2130    0     0     0
## 8 49.190 0.51 Ideal    E     VS1    61.8   55 2075    5.15  31.8  5.12
## 9 24.068 2 Premium  H     SI2    58.9   57 12210   8.09  58.9  8.06
```

- Як можна бачити, перші 7 рядків відповідають діамантам, розмір яких у всіх трьох напрямках нульовий ($x = y = z = 0$)
- Що стосується двох останніх, то з високою ймовірністю можна припустити, що замість 31.8 і 58.9 повинно стояти 3.18 та 5.89
- У таких випадках, як правило, простіше всього **викинути** ці спостереження, бо їх усього 9 на майже 54 000
 - Або ж принаймні замінити дивні значення на NA, щоб не викидати весь рядок
- В інших ситуаціях можна провести дальший аналіз з та **без** викидів, щоб подивитися, який вони мають вплив

Трансформація шкали вимірювання (1)

- Якщо дані додатні і сильно скошені, то перед їх візуалізацією корисно перейти до логаритмічної шкали (logarithmic scale)
- Розгляньмо дані про щільність населення в кожному штаті США в таких роках: 1960, 1970, 1980, 1990, 2000, 2008⁸

```
pop_densities <- read_csv("data/pop_densities.csv")

## Rows: 51 Columns: 8
## -- Column specification -----
## Delimiter: ","
## chr (2): State, Abbrev
## dbl (6): y1960, y1970, y1980, y1990, y2000, y2008
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

str(pop_densities, give.attr = FALSE)

## #> #> spc_tbl_ [51 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## #> #> $ State : chr [1:51] "Alabama" "Alaska" "Arizona" "Arkansas" ...
## #> #> $ Abbrev: chr [1:51] "AL" "AK" "AZ" "AR" ...
## #> #> $ y1960 : num [1:51] 64.4 0.4 11.5 34.3 100.8 ...
## #> #> $ y1970 : num [1:51] 67.88 0.53 15.62 36.94 128.05 ...
## #> #> $ y1980 : num [1:51] 76.7 0.7 23.9 43.9 151.8 ...
## #> #> $ y1990 : num [1:51] 79.62 0.96 32.26 45.15 191.15 ...
## #> #> $ y2000 : num [1:51] 87.6 1.1 45.2 51.3 217.2 ...
## #> #> $ y2008 : num [1:51] 91.9 1.2 57.2 54.8 235.7 ...
```

- Нас цікавить аналіз швидкості зміни щільності за відповідний період

⁸Дані з курсу Джима Алберта

Трансформація шкали вимірювання (2)

- Збудуймо відповідні box plots
 - Для цього нам потрібно перетворити дані в охайній формат
 - Наші дані перебувають у так званому **широкому** (wide) форматі (кожний рік має окремий стовпчик)
 - Нам потрібний **довгий** (long) формат, коли під рік відведено окрему змінну, яка може набувати різних значень
 - Для перетворення даних в охайній (довгий) формат існує функція `pivot longer`

```

pop_densities_long <- pop_densities %>%
  pivot_longer(cols = starts_with("y"),
               names_to = "Year", names_prefix = "y",
               values_to = "Density")
str(pop_densities_long, give.attr = FALSE)

## # tibble [306 x 4] (S3:tbl_df/tbl/data.frame)
## # $ State : chr [1:306] "Alabama" "Alabama" "Alabama" "Alabama" ...
## # $ Abbrev : chr [1:306] "AL" "AL" "AL" "AL" ...
## # $ Year : chr [1:306] "1960" "1970" "1980" "1990" ...
## # $ Density: num [1:306] 64.4 67.9 76.7 79.6 87.6 ...

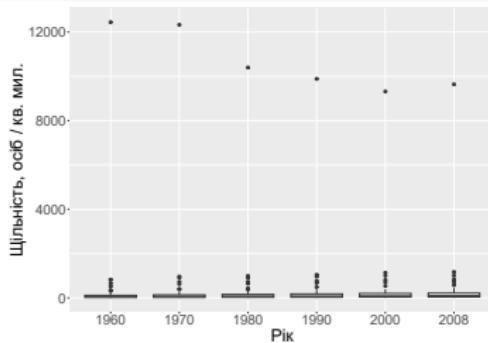
```

- Ми вказали:
 - Які стовпці потрібно перетворити (усі, що починаються на `y`)
 - Як назвати новий стовпець із роками (`names_to = "Year"`)
 - Щоб у цьому стовпці фігурували не назви початкових стовпців типу `y1960`, а числа 1960, ми вказали `names_prefix = "y"`
 - Як назвати новий стовпець зі значеннями для кожного року (`values_to = "Density"`)

Трансформація шкали вимірювання (3)

- Власне, самі графіки

```
ggplot(pop_densities_long, aes(x = factor(Year), y = Density)) +  
  geom_boxplot() +  
  labs(x = "Рік", y = "Щільність, осіб / кв. мил.") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



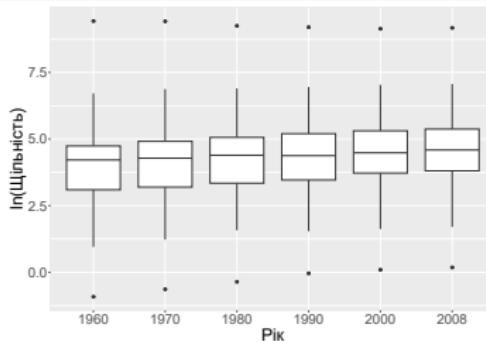
- Як можна бачити, ці графіки складно аналізувати:

- Усі розподіли сильно скошено вправо, і складно розрізнати більшу частину даних, скучених зліва
- Наявність дуже великих викидів спотворює всю картину

Трансформація шкали вимірювання (4)

- Розгляньмо тепер графіки на логаритмічних шкалах

```
ggplot(pop_densities_long, aes(x = factor(Year), y = log(Density))) +  
  geom_boxplot() +  
  labs(x = "Рік", y = "ln(Щільність)") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



- Як можна бачити, ситуація суттєво поліпшилася
 - Усі розподіли стали фактично симетричними
 - Оскільки логаритм від дуже великого числа є дуже малим, наявність викидів не настільки принципова
 - IRQ у всіх роках стали дуже близькі
- Зокрема, можна порівняти медіани між собою

Трансформація шкали вимірювання (5)

- Проте потрібно пам'ятати, що **різниця** логаритмів відповідає **відношенню** початкових даних:

$$e^{\ln b - \ln a} = \frac{b}{a}$$

- Тому нас цікавлять такі перетворення:

```
pop_densities_summary <- pop_densities_long %>%
  group_by(Year) %>%
  summarise(med = median(log(Density)), iqr = IQR(log(Density))) %>%
  mutate(med_rat = c(1, exp(diff(med))), iqr_rat = c(1, exp(diff(iqr))))
pop_densities_summary

## # A tibble: 6 x 5
##   Year     med    iqr med_rat iqr_rat
##   <chr>   <dbl>  <dbl>    <dbl>    <dbl>
## 1 1960     4.21   1.64     1       1
## 2 1970     4.28   1.72     1.07    1.08
## 3 1980     4.39   1.72     1.12    1.00
## 4 1990     4.38   1.74     0.983   1.02
## 5 2000     4.48   1.59     1.11    0.862
## 6 2008     4.59   1.57     1.11    0.979

prod(pop_densities_summary$med_rat)
## [1] 1.454411

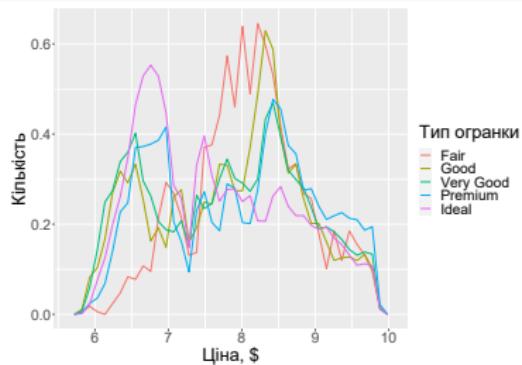
prod(pop_densities_summary$iqr_rat)
## [1] 0.9264059
```

- Як можемо бачити, **мультиплікативно** медіани та IQR не сильно змінилися

Трансформація шкали вимірювання (6)

- Усі міркування стосуються також і гістограм
- Наприклад, можемо прологаритмувати ціну діамантів

```
ggplot(diamonds, aes(x = log(price), y = after_stat(density), color = cut)) +  
  geom_freqpoly(bins = 40) +  
  labs(x = "Ціна, $", y = "Кількість", color = "Тип огранки") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

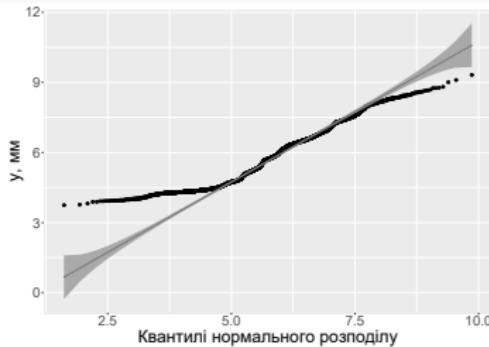


- Тепер розподіли стали симетричніші
- Це дає змогу помітити, що для всіх категорій (окрім, хіба що, Fair), насправді мають місце два кластери каменів (розподіли бімодальні)

Особливості застосування QQ-графіків (1)

- Як ми вже знаємо, QQ-графіки можна застосовувати для пошуку викидів

```
library(ggplot2)
diamonds <- slice_sample(diamonds, n = 5000)
ggplot(diamonds, aes(sample = y)) +
  stat_qq_point() + stat_qq_line() + stat_qq_band() +
  labs(x = "Квантилі нормального розподілу", y = "y, мм") +
  theme(axis.title = element_text(size = 25),
        axis.text = element_text(size = 20))
```

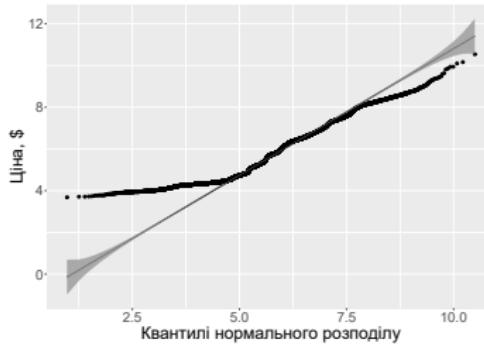


- Бачимо ті самі викиди для змінної y , що й раніше
- Також ми додали довірчі інтервали (сірі смуги)
 - Як можна бачити, емпіричні квантилі вибиваються за межі цих інтервалів
 - Це здійснює свідчення того, що розподіл не є нормальним

Особливості застосування QQ-графіків (2)

- Якщо прибрати ці викиди, бачимо таку картину

```
ggplot(diamonds %>% filter(y > 0 & y < 20), aes(sample = y)) +  
  stat_qq_point() + stat_qq_line() + stat_qq_band() +  
  labs(x = "Квантилі нормального розподілу", y = "Ціна, $") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```

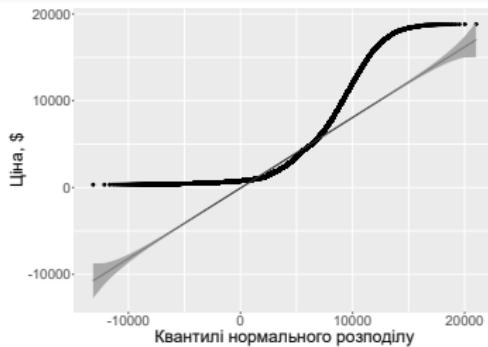


- Як можна бачити, змінна у має нормальній розподіл тільки посередині
- На хвостах маємо суттєві відхилення:
 - Малі квантилі у більші від теоретичних
 - Великі квантилі у менші від теоретичних
 - Тобто розподіл у **вужчий** від стандартного нормального розподілу

Особливості застосування QQ-графіків (3)

- Розгляньмо QQ-графік для ціни

```
ggplot(diamonds, aes(sample = price)) +  
  stat_qq_point() + stat_qq_line() + stat_qq_band() +  
  labs(x = "Квантилі нормального розподілу", y = "Ціна, $") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```

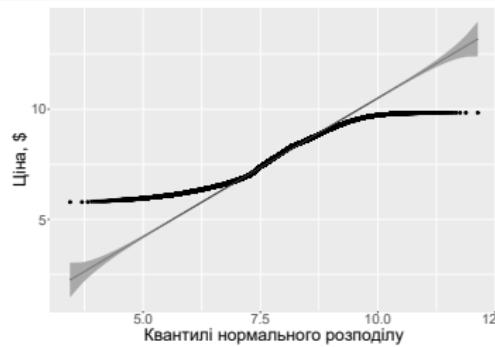


- Як можна бачити, розподіл не є нормальним

Особливості застосування QQ-графіків (4)

- Але навіть логаритмування слабко допомагає

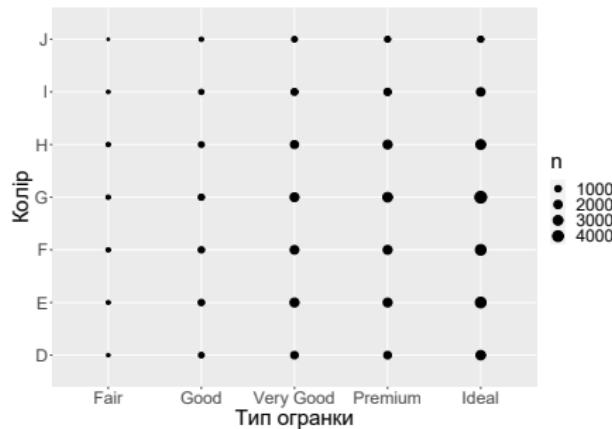
```
ggplot(diamonds, aes(sample = log(price))) +  
  stat_qq_point() + stat_qq_line() + stat_qq_band() +  
  labs(x = "Квантилі нормального розподілу", y = "Ціна, $") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```



Візуалізація двох категорійних змінних (1)

- Якщо обидві змінні категорійні, то можливий варіант зобразити їх такий

```
ggplot(diamonds, aes(x = cut, y = color)) +  
  geom_count() +  
  labs(x = "Тип огранки", y = "Колір") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```

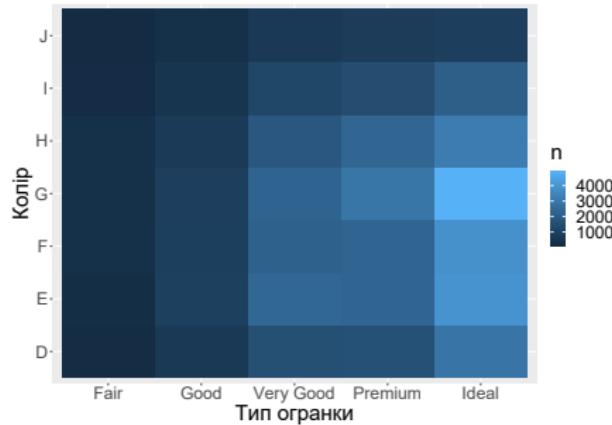


- Можемо бачити, що більше всього каменів мають колір G (ні дуже поганий, ні дуже добрий) в усіх категоріях огранки

Візуалізація двох категорійних змінних (2)

- Схожу інформацію можна зобразити за допомогою так званої **теплової карти** (heatmap)

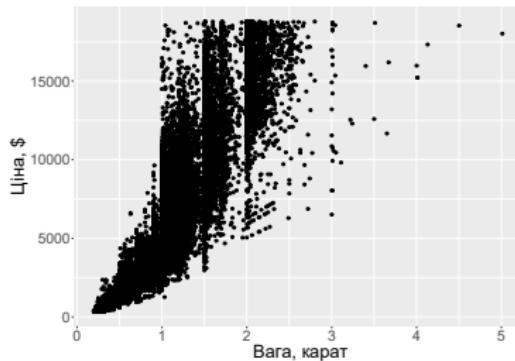
```
ggplot(diamonds %>% count(color, cut), aes(x = cut, y = color, fill = n)) +  
  geom_tile() +  
  labs(x = "Тип огранки", y = "Колір") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20),  
        legend.title = element_text(size = 25),  
        legend.text = element_text(size = 20))
```



Візуалізація двох неперервних змінних (1)

- Чи не найліпший спосіб візуалізації двох неперервних змінних — scatter plots
- Проте в окремих випадках це може бути не ідеальним рішенням

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point() +  
  labs(x = "Вага, карат", y = "Ціна, $") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```

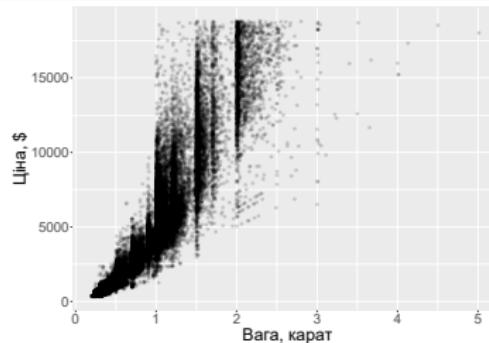


- Ми бачимо, що існує експоненційна залежність, але оскільки точок дуже багато, вони налізають одна на одну

Візуалізація двох неперервних змінних (2)

- Можливим поліпшенням є встановлення параметру прозорості для точок — `alpha`, яке лежить від 0 (повністю прозорі) до 1 (повністю непрозорі)

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point(alpha = 0.1) +  
  labs(x = "Бара, карат", y = "Ціна, $") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20))
```

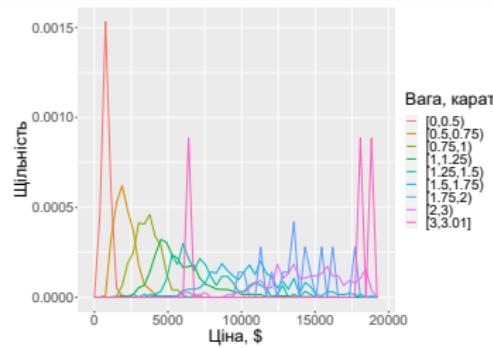


- Тепер видно, що ціни зростають не зовсім неперервно, а певними стрибками
- Альтернативно можна використати функцію `geom_bin2d`, яка будує «двовимірну гістограму», де кожній прямокутній комірці на площині відповідає кількість спостережень у ній
- Також можна згрупувати значення неперервної змінної в окремі інтервали, фактично перетворивши її в категорійну

Візуалізація двох неперервних змінних (3)

- Якщо розділити вагу діамантів на інтервали по 0.5 карата, то розподіли ціни будуть такі

```
ggplot(diamonds,
       aes(x = price, y = after_stat(density),
           color = cut(carat, breaks = c(0, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 3, max(carat)),
                   include.lowest = TRUE, right = FALSE)) +
       geom_freqpoly(bins = 50) +
       labs(x = "Ціна, $", y = "Щільність", color = "Вага, карат") +
       theme(axis.title = element_text(size = 25),
             axis.text = element_text(size = 20),
             legend.title = element_text(size = 25),
             legend.text = element_text(size = 20))
```

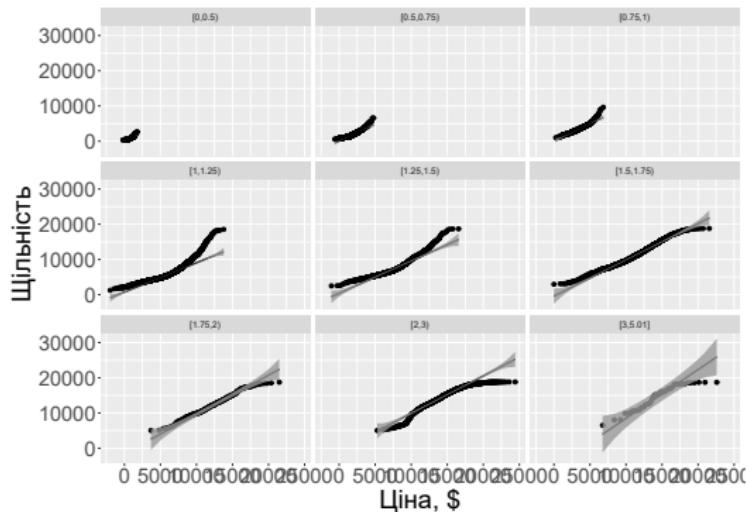


- Розподіли ціни для каменів малої ваги мають приблизно логнормальний розподіл
- Камені середньої ваги мають розподіл, схожий на нормальній
- Камені більше 2 карат вагою мають специфічний розподіл, оскільки їх відносно небагато (1889)

Візуалізація двох неперервних змінних (4)

- Це можна побачити також на QQ-графіках

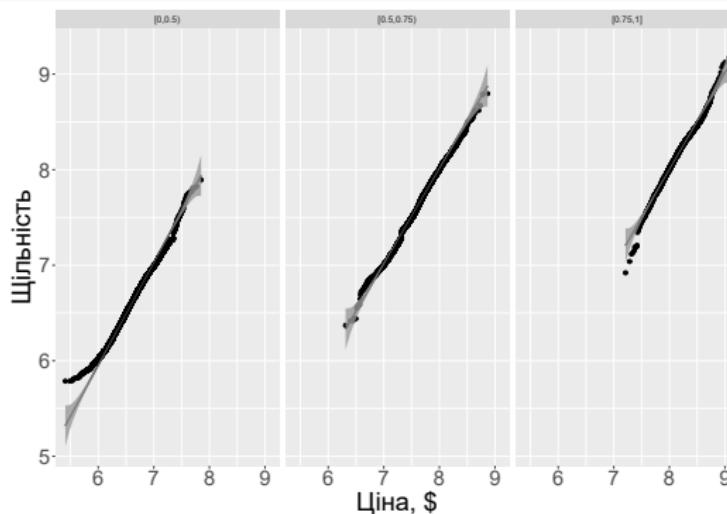
```
ggplot(diamonds, aes(sample = price)) +  
  stat_qq_point() + stat_qq_line() + stat_qq_band() +  
  labs(x = "Ціна, $", y = "Щільність") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20)) +  
  facet_wrap(~ cut(carat,  
                  breaks = c(0, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 3, max(carat)),  
                  include.lowest = TRUE, right = FALSE))
```



Візуалізація двох неперервних змінних (5)

- Можемо застосувати логаритм до перших трьох вагових категорій

```
ggplot(diamonds %>% filter(carat < 1), aes(sample = log(price))) +  
  stat_qq_point() + stat_qq_line() + stat_qq_band() +  
  labs(x = "Ціна, $", y = "Щільність") +  
  theme(axis.title = element_text(size = 25),  
        axis.text = element_text(size = 20)) +  
  facet_wrap(~ cut(carat,  
                    breaks = c(0, 0.5, 0.75, 1),  
                    include.lowest = TRUE, right = FALSE))
```



- Як бачимо, застосування логаритма зробило розподіл більш нормальним

Зображення кореляцій між різними змінними (1)

- Як відомо з теорії ймовірностей, корисною властивістю спільного розподілу двох випадкових величин X та Y є їх **коваріація** (covariance):
$$\text{Cov}(X, Y) \equiv \sigma_{XY} = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$
- Варто згадати, що замість **дисперсії** (variance) $\text{Var}(X)$ ми використовуємо середньоквадратичне відхилення $\sigma_X = \sqrt{\text{Var}(X)}$
 - Воно має ті ж одиниці виміру, що й X
- Аналогічно, $\text{Cov}(X, Y)$ має інші одиниці виміру, аніж X чи Y , тому доцільно перейти до **коефіцієнта кореляції** (correlation):

$$\text{Corr}(X, Y) \equiv \rho_{XY} = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

- Його також називають **коефіцієнтом кореляції Пірсона**⁹
- Коефіцієнт кореляції є безрозмірною величиною і завжди $\rho_{XY} \in [-1; 1]$

⁹Карл Пірсон — англійський математик (Karl Pearson, 1857–1936)

Зображення кореляцій між різними змінними (2)

- Коваріація двох випадкових величин $\text{Cov}(X, Y)$ показує ступінь лінійного зв'язку між ними
- Якщо $\text{Cov}(X, Y) > 0$, то між величинами існує **додатний** лінійний зв'язок, тобто більші значення однієї величини свідчать про більші значення іншої
- Якщо $\text{Cov}(X, Y) < 0$, то між величинами існує **від'ємний** лінійний зв'язок, тобто більші значення однієї величини свідчать про менші значення іншої
- Якщо $\text{Cov}(X, Y) = 0$, то лінійного зв'язку між величинами немає
- Те саме можна сказати про $\text{Corr}(X, Y)$
- Понад те, що більші за модулем до 1 значення $\text{Corr}(X, Y)$, то сильніша лінійна залежність

Зображення кореляцій між різними змінними (3)

- Очевидною оцінкою коефіцієнта кореляції є статистика, де замість коваріації та дисперсії використано їхні вибіркові аналоги
- Нехай маємо вибірку $((X_1, Y_1)^\top, \dots, (X_n, Y_n)^\top)$ таку, що $(X_i, Y_i)^\top$ незалежні і мають одинаковий розподіл
- Тоді

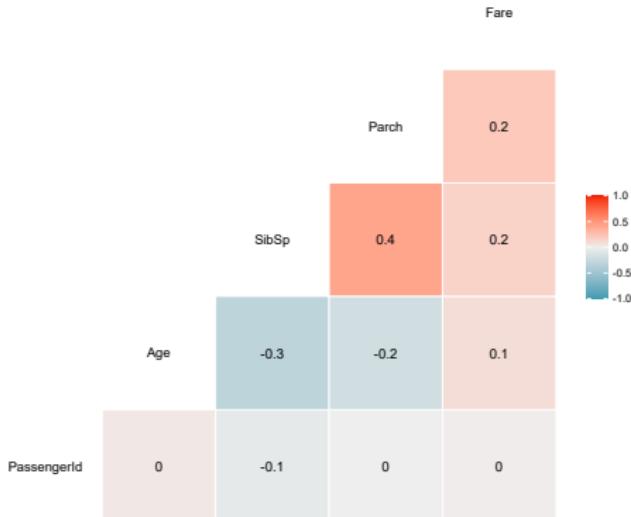
$$\hat{\rho}_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

- В R коефіцієнт кореляції можна обчислити за допомогою функції `cor`

Зображення кореляцій між різними змінними (4)

- А візуалізувати як **корелограму** (correlogram) можна за допомогою спеціальної функції з пакета GGally

```
ggcorr(passengers %>% select(where(is.numeric)), label = TRUE)
```

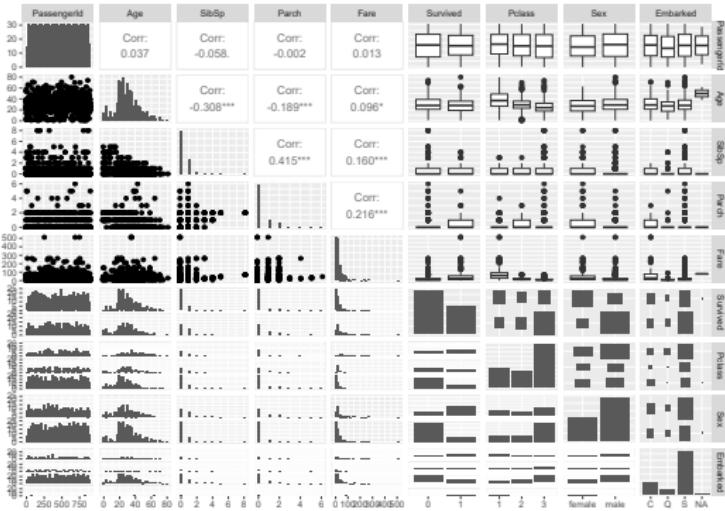


- Як можна бачити, особливо сильної кореляції між змінними немає

Зображення кореляцій між різними змінними (5)

- Інша корисна функція з цього пакета показує не тільки кореляції, а й попарні scatter plots, box plots, гістограми тощо

```
ggpairs(passengers %>% select(where(is.numeric) | where(is.factor)),  
       diag = list(continuous = "barDiag", discrete = "barDiag", na = "naDiag"))
```



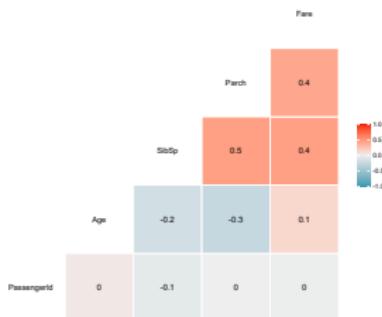
- Використовуючи аргументи `upper`, `lower` та `diag`, можна вказувати, які саме візуалізації використовувати для неперервних та категорійних змінних

Зображення кореляцій між різними змінними (6)

- Коефіцієнт кореляції Пірсона свідчить тільки про наявність лінійного зв'язку між випадковими величинами
 - Як існує не лінійний, але монотонний зв'язок, то можна натомість використати **коефіцієнт кореляції Спірмана (Spearman correlation)**¹⁰

- Для цього треба у функції `cor` вказати `method = "spearman"`
 - ... а у функції `qqcorr` — `method = c("pairwise", "spearman")`

```
ggcorr(passengers %>% select(where(is.numeric)), label = TRUE,  
       method = c("pairwise", "spearman"))
```



- Коефіцієнт кореляції Спірмана — це просто коефіцієнт кореляції Пірсона, але між *порядковими номерами відсортованих* X_i та Y_i
 - Він стійкіший до наявності викидів

¹⁰Чарльз Спірман — британський психолог (Charles Edward Spearman, 1863–1945)

Зображення кореляцій між різними змінними (7)

- Також коефіцієнт кореляції Спірмана можна застосовувати до категорійних змінних, які є **впорядкованими** (ordered)

```
ggcorr(passengers %>% select(where(is.numeric) | where(is.ordered)) %>%  
  mutate(Pclass = as.numeric(Pclass)), label = TRUE, method = c("pairwise", "spearman"))
```



- Як можна бачити, Fare і Pclass мають посутню від'єму кореляцію, що цілком очікувано
- Також слабенька від'ємна кореляція існує між Pclass та Age

Принципи добрих аналітичних графіків

- Розгляньмо 6 принципів побудови інформативних та корисних графіків, описаних Едвардом Тафті (Edward Tufte) у книжці *Beautiful Evidence*
- ① Демонструвати на графіку порівняння, наприклад, box plots для різних категорій чи фацетовані гістограми тощо
- ② Зображені на графіку механізмів, які пояснюють, чому ми спостерігаємо ту чи ту картину
- ③ Показувати на графіку декілька змінних одночасно (використовуючи кольори, розмір і форму точок тощо)
- ④ Використовувати на графіку, за потреби, текстові мітки, числа тощо
- ⑤ Вказувати підписи осей, шкали вимірювань, промовисту назву, авторів чи використаних джерел тощо
- ⑥ Фокусуватися на контенті графіка, а не на дизайнерських рішеннях
- Приклади неякісних графіків та типових помилок можна проглянути [тут](#) і [тут](#)