**Практичне заняття 2**

**Робота з файлом .csv**

R це мова програмування, яка широко використовується для аналізу даних. Будемо використовувати бібліотеки

• dplyr: для очищення та трансформації даних

• ggplot2: для візуалізації даних

Ці бібліотеки завантажуються з допомогою команди install.packages. Створіть новий файл (File -> New File -> RScript) та скопіюйте наступні рядки:

**install.packages**("dplyr")

**install.packages**("ggplot2")

Щоб виконати код, виділіть рядки та натисніть піктограму Run з зеленою стрілкою або комбінацію клавіш CTRL + ENTER або COMMAND + ENTER. Далі завантажте ці бібліотеки до вашого робочого середовища. Це можна зробити з допомогою функції library.

**Встановлюємо** бібліотеку один раз, але завантажувати її потрібно щоразу, як ви перезапускаєте RStudio. Тобто при наступному запуску RStudio команди інсталяції не будуть потрібні і їх можна буде закоментувати використовуючи символ #:

*#install.packages("dplyr")*

*#install.packages("ggplot2")*

Додайте рядки:

**library**(dplyr)

**library**(ggplot2)

Завантажте файл "flats.csv" та помістіть у ту ж папку, де знаходиться створений вами RScript. Завантажте дані з файла "flats.csv" у змінну flats використовуючи функцію read.csv.

flats <- **read.csv**("flats.csv", stringsAsFactors=FALSE, encoding="UTF-8")

Параметр encoding="UTF-8" використовується для коректного відображення кирилиці у OS Windows.

Параметр stringsAsFactors=FALSE вказує, що змінні, які мають тип character не будуть перетворюватись у тип даних factor. Цей тип використовується для роботи з категоріальними змінними, однак в межах цієї лабораторної ми не будемо його використовувати. Якщо отримали помилку

Error in file(file, "rt") : cannot open the connection In addition: Warning message: In file(file, "rt") : cannot open file 'flats.csv': No such file or directory

вкажіть шлях до цієї директорії використовуючи командуsetwd (скорочення від set working directory). Виконання цієї команди дозволяє не вказувати повний шлях до цієї директорії.

**setwd**("шлях до файла")

*# приклад:*

*#setwd("~/work/stats\_course/materials/week2")*

Визначимо клас обє'кта flats з допомогою команди class()

**class**(flats)

## [1] "data.frame"

Клас об'єкта flats data.frame або ж таблиця даних. Кожен рядок цієї таблиці репрезентує спостереження, а кожна колонка відображає змінну, тобто частину інформації про це спостереження.

В R можна використовувати функцію str(скорочення від structure) щоб швидко оцінити, чи правильно зчиталися дані.

**str**(flats)

## 'data.frame': 839 obs. of 4 variables:

## $ Місто : chr "Вінниця" "Вінниця" "Вінниця" "Вінниця" ...

## $ Кімнат : int 3 3 2 2 3 1 3 3 1 6 ...

## $ Загальна\_площа: chr "120" "66" "66" "44" ...

## $ Ціна : num 1875000 975000 1375000 637500 835000 ...

Бачимо, що змінна Загальна\_площа має тип "character", тобто розпізналася як текстова змінна. Переглянемо документацію по функції read.csv використовуючи функцію ?

?read.csv

Бачимо, що в якості десяткового розділювача по замовчуванню використовується крапка dec ='.'. А в наших даних десятковим розділювачем є кома.





Заново зчитаємо дані, вказавши параметр десяткового розділювача:

flats <- **read.csv**("flats.csv", stringsAsFactors=FALSE, dec= ",")

Перевіримо їх структуру:

**str**(flats)

## 'data.frame': 839 obs. of 4 variables:

## $ Місто : chr "Вінниця" "Вінниця" "Вінниця" "Вінниця" ...

## $ Кімнат : int 3 3 2 2 3 1 3 3 1 6 ...

## $ Загальна\_площа: num 120 66 66 44 63 31 46 64 35 200 ...

## $ Ціна : num 1875000 975000 1375000 637500 835000 ...

**Дослідження даних**

• Для того, шоб знайти кількість вимірів, використовується функція dim()

• head() відображає першу частину об'єкта, першим параметром є об'єкт(тут

таблиця даних flats, другим параметром можна вказати кількість рядків)

• tail() відображає останню частину об'єкта, теж можна вказати кількість рядків

• names() імена, пов'язані з об'єктом

**Вправи**

• Знайдіть кількість вимірів датафрейму flats.

• Відобразіть перші шість рядків, перші п'ятнадцять рядків, останні шість рядків.

• Відобразіть імена датафрейму.

**Трансформація даних**

В R можна використовувати функцію str() та summary() щоб отримати перші знання про таблицб. Бібліотека dplyr має функцію glimpse() для швидкого узагальнення таблиці.

*# Look at structure of flats*

**str**(flats)

## 'data.frame': 839 obs. of 4 variables:

## $ Місто : chr "Вінниця" "Вінниця" "Вінниця" "Вінниця" ...

## $ Кімнат : int 3 3 2 2 3 1 3 3 1 6 ...

## $ Загальна\_площа: num 120 66 66 44 63 31 46 64 35 200 ...

## $ Ціна : num 1875000 975000 1375000 637500 835000 ...

*# View a summary of flats*

**summary**(flats)

## Місто Кімнат Загальна\_площа Ціна

## Length:839 Min. :1.000 Min. : 14.00 Min. : 10200

## Class :character 1st Qu.:1.000 1st Qu.: 43.75 1st Qu.: 537500

## Mode :character Median :2.000 Median : 56.00 Median : 775000

## Mean :2.045 Mean : 64.07 Mean : 1042710

## 3rd Qu.:3.000 3rd Qu.: 75.00 3rd Qu.: 1200000

## Max. :6.000 Max. :222.60 Max. :12250000

*# Get a glimpse of flats*

**glimpse**(flats)

## Observations: 839

## Variables: 4

## $ Місто <chr> "Вінниця", "Вінниця", "Вінниця", "Вінниця", "Ві...

## $ Кімнат <int> 3, 3, 2, 2, 3, 1, 3, 3, 1, 6, 2, 1, 1, 2, 3, 3,...

## $ Загальна\_площа <dbl> 120.00, 66.00, 66.00, 44.00, 63.00, 31.00, 46.0...

## $ Ціна <dbl> 1875000, 975000, 1375000, 637500, 835000, 56250...

Дізнаємося, яка кількість квартир продається у кожному місті (згідно цього набору даних):

В бібліотеці dplyr для цього є функція count:

**count**(flats, Місто)

## # A tibble: 13 × 2

## Місто n

## <chr> <int>

## 1 Вінниця 275

## 2 Дніпропетровськ 18

## 3 Запоріжжя 13

## 4 Івано-Франківськ 47

## 5 Києво-Святошинський 19

## 6 Київ 186

## 7 Львів 16

## 8 Миколаїв 15

## 9 Одеса 43

## 10 Рівне 23

## 11 Тернопіль 93

## 12 Харків 14

## 13 Хмельницький 77

Якщо ми хочемо виконати послідовно кілька операцій в dplyr можна використати оператор %>%, який дозволяє застосувати наступну команду до результатів виконання поточної. Наприклад, посортуємо дані по кількості квартир у кожному місті у зростаючому порядку:

flats %>%

**count**(Місто) %>%

**arrange**(n)

## # A tibble: 13 × 2

## Місто n

## <chr> <int>

## 1 Запоріжжя 13

## 2 Харків 14

## 3 Миколаїв 15

## 4 Львів 16

## 5 Дніпропетровськ 18

## 6 Києво-Святошинський 19

## 7 Рівне 23

## 8 Одеса 43

## 9 Івано-Франківськ 47

## 10 Хмельницький 77

## 11 Тернопіль 93

## 12 Київ 186

## 13 Вінниця 275

Як бачимо, Києво-Святошинський район виділений в окреме місто. Можливо тому, що його адміністративним центром є місто Київ. Вилучимо ці дані з відображення використовуючи команду filter.

Нагадаю, що

умова дорівнює позначається як ==, а не дорівнює як !=. Також посортуємо

результати в спадаючому порядку для цього вкажемо arrange(desc(n)).

flats %>%

**filter**(Місто != "Києво-Святошинський") %>%

**filter**(Кімнат == 3) %>%

**count**(Місто) %>%

**arrange**(**desc**(n)) *# arrange - сортування, desc - спадаючий порядок*

## # A tibble: 12 × 2

## Місто n

## <chr> <int>

## 1 Вінниця 60

## 2 Київ 50

## 3 Тернопіль 24

## 4 Хмельницький 22

## 5 Івано-Франківськ 13

## 6 Одеса 11

## 7 Дніпропетровськ 8

## 8 Запоріжжя 8

## 9 Рівне 6

## 10 Миколаїв 5

## 11 Харків 3

## 12 Львів 2

Якщо нас цікавлять кількість двокімнатних квартир в кожному місті, то виберемо лише квартири з кількістю кімнат 2:

flats %>%

**filter**(Кімнат == 2) %>%

**filter**(Місто != "Києво-Святошинський") %>%

**count**(Місто) %>%

**arrange**(**desc**(n))

## # A tibble: 12 × 2

## Місто n

## <chr> <int>

## 1 Вінниця 93

## 2 Київ 67

## 3 Тернопіль 43

## 4 Хмельницький 28

## 5 Одеса 18

## 6 Івано-Франківськ 14

## 7 Рівне 8

## 8 Миколаїв 7

## 9 Харків 7

## 10 Дніпропетровськ 5

## 11 Львів 5

## 12 Запоріжжя 2

Функція summarise дозволяє узагальнити дані. Наприклад, знайти середнє значення площі квартир в кожному регіоні. Для обрахунку середнього значення використаємо функцію mean.

flats %>%

**filter**(Кімнат == 2) %>%

**filter**(Місто != "Києво-Святошинський") %>%

**summarise**(**mean**(Загальна\_площа))

## mean(Загальна\_площа)

## 1 60.81832

Можна обчислити не лише площу, але й середньоквадратичне відхилення з допомогою функції sd:

flats %>%

**filter**(Кімнат == 2) %>%

**filter**(Місто != "Києво-Святошинський") %>%

**summarise**(**mean**(Загальна\_площа), **sd**(Загальна\_площа))

## mean(Загальна\_площа) sd(Загальна\_площа)

## 1 60.81832 16.61458

Можна задати назви стовпців, наприклад mean=mean(Загальна\_площа):

flats %>%

**filter**(Кімнат == 1) %>%

**filter**(Місто != "Києво-Святошинський") %>%

**group\_by**(Місто) %>%

**summarise**(mean=**median**(Загальна\_площа), sd=**sd**(Загальна\_площа))

## # A tibble: 12 × 3

## Місто mean sd

## <chr> <dbl> <dbl>

## 1 Вінниця 40.0 7.665871

## 2 Дніпропетровськ 32.0 NA

## 3 Запоріжжя 36.4 9.050967

## 4 Івано-Франківськ 40.7 4.989404

## 5 Київ 39.0 8.015938

## 6 Львів 43.0 6.269465

## 7 Миколаїв 37.5 6.363961

## 8 Одеса 39.0 5.015531

## 9 Рівне 35.0 13.086362

## 10 Тернопіль 43.0 8.079379

## 11 Харків 18.5 10.472185

## 12 Хмельницький 42.0 6.669957

**Вправи**

• Скільки змінних у наборі даних flats?

• Яка кількість міст у наборі даних flats?

• Чи всі з них дійсно є містами?

• Яка кількість трикімнатних квартир продається в місті Одеса?

• Яка медіана площі однокімнатної квартири в місті Львів?

**Візуалізація даних:**

Для візуалізації даних будемо використовувати бібліотеку ggplot2. В процесі розвідувального аналізу даних (Exploratory Data Analysis) процеси очищення та візуалізації даних є циклічними (як ви вже бачили у відео лекції). Для побудови графіків використовується функція ggplot() Після виконання коду можна побачити графік у вкладці *Plots* у нижній правій панелі в RStudio.

• Першим аргументом цієї фyнкції є набір даних (dataset)

• Далі вказуємо змінні з набору даних як параметр aesthetic, які будуть відображатись, наприклад, по осях x та y

• Наступним кроком ми додаємо ще один рівень (об'єднавши їх знаком + ) щоб задати geometric об'єкт. Наприклад, для графіка розсіювання це geom\_point, для лінійного графіка geom\_line, для стовпчикової діаграми geom\_bar

Побудуємо стовпчикову діаграму для кількості кімнат:

**ggplot**(flats, **aes**(x=Кімнат)) +

**geom\_bar**(fill="lightblue",

col="grey") +

**ylab**('Кількість')

Побудуємо стовпчикову діаграму для змінної загальна площа:

****

**Стовпчикова діаграма**

p <- **ggplot**(flats, **aes**(x=Загальна\_площа)) +

**geom\_bar**(fill="lightblue",

col="grey") +

**ylab**('Кількість')

p



**Гістограма**

Використовується для оцінки форми розподілу кількісної змінної. На цьому графіку розподіл квартир, які продаються за загальною площею.



Залежно від розміру інтервалу її форма може змінюватися. Наприклад змінимо інтервал з 25 метрів квадратних до 50:

**ggplot**(flats, **aes**(x=Загальна\_площа)) +

**geom\_histogram**(breaks=**seq**(0, 250, by = 50),

fill="lightblue",

col="grey") +

**ylab**('Кількість')



Графік розсіювання. Побудуємо графік залежності ціни від загальної площі.

**library**(ggplot2)

**ggplot**(flats, **aes**(x=Загальна\_площа, y=Ціна)) +

**geom\_point**()



**Коробчата діаграма**

Порівняємо розподіл цін по містах та використаємо параметр coord\_flip() щоб

розмістити коробчаті діаграми горизонтально:

****

**Вправи**

• Побудуйте коробчату діаграму для візулізації розподілу цін в залежності від кількості кімнат

• Побудуйте графік розсіювання, який відображатиме залежність ціни від

загальної площі

• Побудуйте гістограму для оцінки розподілу ціни квартир\_