



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”
Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем



Лабораторна робота №5

з дисципліни: «Технології оброблення великих даних»

на тему: «Аналіз та візуалізація даних в R»

Виконав

студент III курсу каф.
ПЗКС ФПМ

групи КП-82

Мельничук Олексій
Геннадійович

Перевірила

доц. каф. ПЗКС ФПМ

Олещенко Л.М.

1. Індивідуальне завдання

Мета: ознайомитись з можливостями мови програмування R для аналізу та візуалізації даних.

2. Хід роботи

Практичне завдання

Частина 1.

Завдання 1.

Скільки рядків та стовпців містить цей датафрейм?

```
> data(Orange)
> help("Orange")
> ncol(Orange)
[1] 3
> nrow(Orange)
[1] 35
```

Orange # (навести скріншот)

```
> Orange#
```

	Tree	age	circumference
1	1	118	30
2	1	484	58
3	1	664	87
4	1	1004	115
5	1	1231	120
6	1	1372	142
7	1	1582	145
8	2	118	33
9	2	484	69
10	2	664	111
11	2	1004	156
12	2	1231	172
13	2	1372	203
14	2	1582	203
15	3	118	30
16	3	484	51
17	3	664	75
18	3	1004	108
19	3	1231	115
20	3	1372	139
21	3	1582	140
22	4	118	32
23	4	484	62
24	4	664	112
25	4	1004	167
26	4	1231	179
27	4	1372	209
28	4	1582	214
29	5	118	30
30	5	484	49
31	5	664	81
32	5	1004	125
33	5	1231	142
34	5	1372	174
35	5	1582	177

```
> |
```

Class (Orange) # (навести скріншот)

```
> class (Orange)#  
[1] "nfnGroupedData" "nfGroupedData" "groupedData" "data.frame"  
> |
```

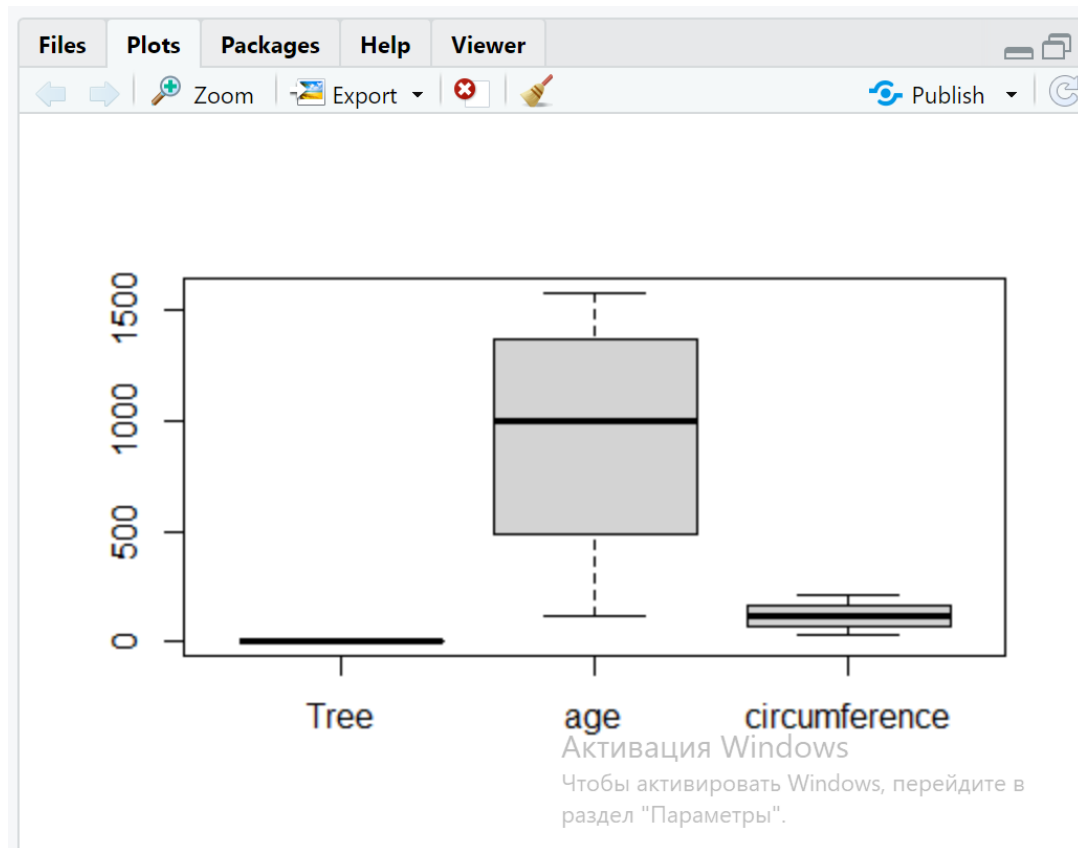
str (Orange) # (навести скріншот)

```
> str (Orange)#  
Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':      3  
5 obs. of  3 variables:  
 $ Tree      : Ord.factor w/ 5 levels "3"<"1"<"5"<"2"<...: 2 2 2 2 2 2 2 4 4 4  
 ...  
 $ age       : num  118 484 664 1004 1231 ...  
 $ circumference: num  30 58 87 115 120 142 145 33 69 111 ...  
 - attr(*, "formula")=Class 'formula' language circumference ~ age | Tree  
 .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>  
 - attr(*, "labels")=List of 2  
 ..$ x: chr "Time since December 31, 1968"  
 ..$ y: chr "Trunk circumference"  
 - attr(*, "units")=List of 2  
 ..$ x: chr "(days)"  
 ..$ y: chr "(mm)"  
>
```

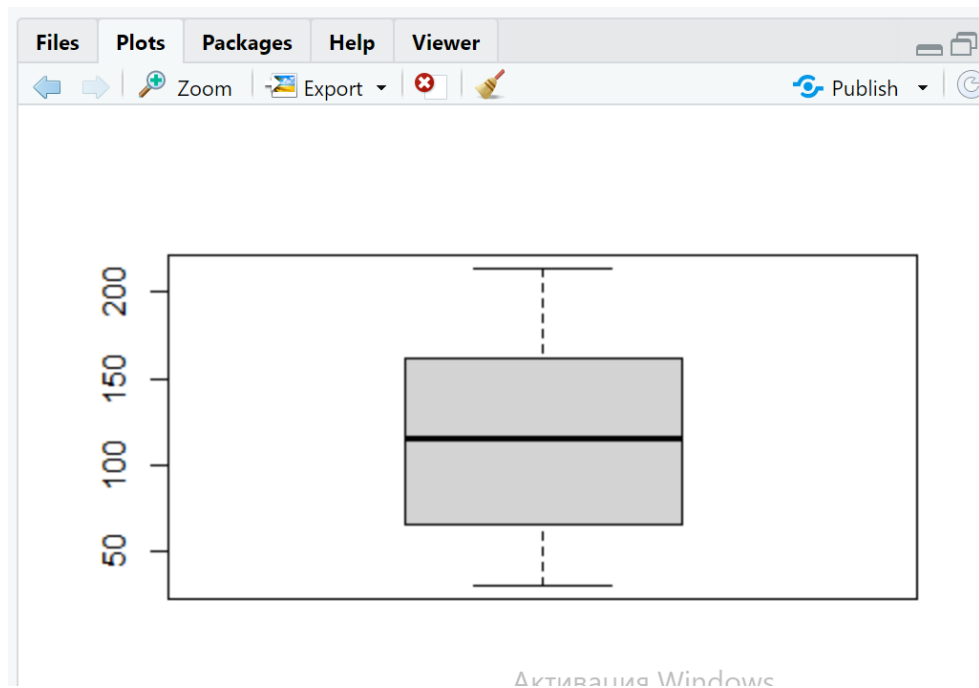
summary (Orange) # (навести скріншот)

```
> summary (Orange)#  
Tree      age      circumference  
3:7   Min.   : 118.0   Min.     : 30.0  
1:7   1st Qu.: 484.0   1st Qu.  : 65.5  
5:7   Median :1004.0   Median   :115.0  
2:7   Mean    : 922.1   Mean     :115.9  
4:7   3rd Qu.:1372.0   3rd Qu.  :161.5  
      Max.    :1582.0   Max.     :214.0  
> |
```

boxplot (Orange) # (навести скріншот)

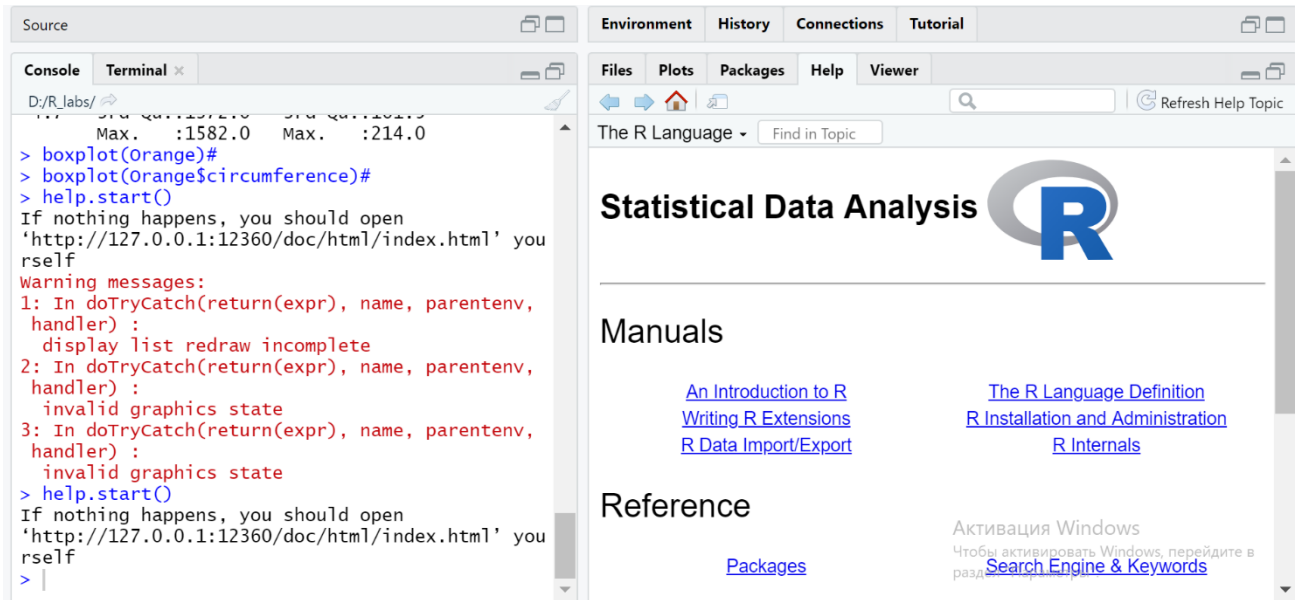


boxplot (Orange\$circumference) # (навести скріншот)

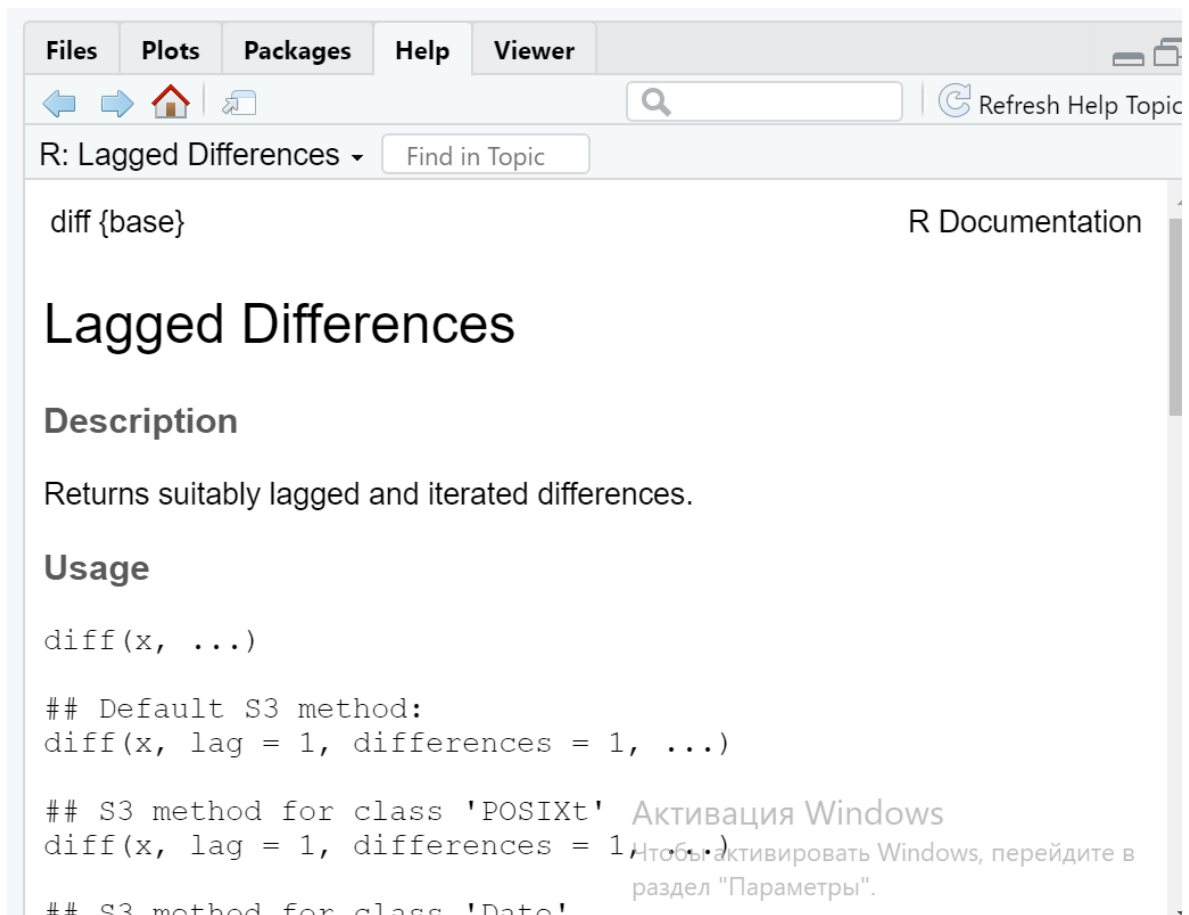


Завдання 2.

Help.start() (навести скріншот)



Help(diff)



RSiteSearch("diff")

The screenshot shows a web browser window with the URL `search.r-project.org/?P=diff&HITSPPERPAGE=20&SORT=&DB=cran-help&DB=r-help&DB=cran-vignettes&DB=cran-views`. The search bar contains the text "diff". Below the search bar, there are options for "Search in:" with checkboxes for "R", "Manuals", "Base Packages", "Help Pages", "CRAN", and "Task Views". There are also checkboxes for "CRAN Packages", "General Info", "Help Pages", "News", "Readme", and "Vignettes". The search results are sorted by "relevance" and show 20 results per page. The first four results are listed below:

- R: get.X.diff**
get.X.diff {aucm} R Documentation get.X.diff Description computes X.diff matrix Usage get.X.diff(x1,...) ## Default S3 method: get.X.diff(x1,x2,...) ## S3 method for class 'formula' get.X.diff(formula...
[/CRAN/refmans/aucm/html/get.X.diff.html](#)
cran-help matching: diff, diff and diff
- R: Summary function for multeq.diff**
...multeq.diff. Usage ## S3 method for class 'multeq.diff' summary(object, digits = 4, ...) Arguments object an object of class "multeq.diff" as obtained by calling multeq.diff digits digits for rounding...
[/CRAN/refmans/MultEq/html/summary.multeq.diff.html](#)
cran-help matching: diff, diff and diff
- R: Print out of the results of multeq.diff**
...multeq.diff. Usage ## S3 method for class 'multeq.diff' print(x, digits = 4, ...) Arguments x an object of class "multeq.diff" as obtained by calling multeq.diff digits digits for rounding the results ...
[/CRAN/refmans/MultEq/html/print.multeq.diff.html](#)
cran-help matching: diff, diff and diff
- R: diff method for spikeTrain objects**
...diff method for spikeTrain objects Description diff method for spikeTrain objects. Usage ## S3 method for class 'spikeTrain' diff(x, ...) Arguments x a spikeTrain object ... see diff Value a numeric...
[/CRAN/refmans/STAR/html/diff.spikeTrain.html](#)
cran-help matching: diff, diff and diff

example(diff)

```
> example(diff)

diff> diff(1:10, 2)
[1] 2 2 2 2 2 2 2 2

diff> diff(1:10, 2, 2)
[1] 0 0 0 0 0 0 0

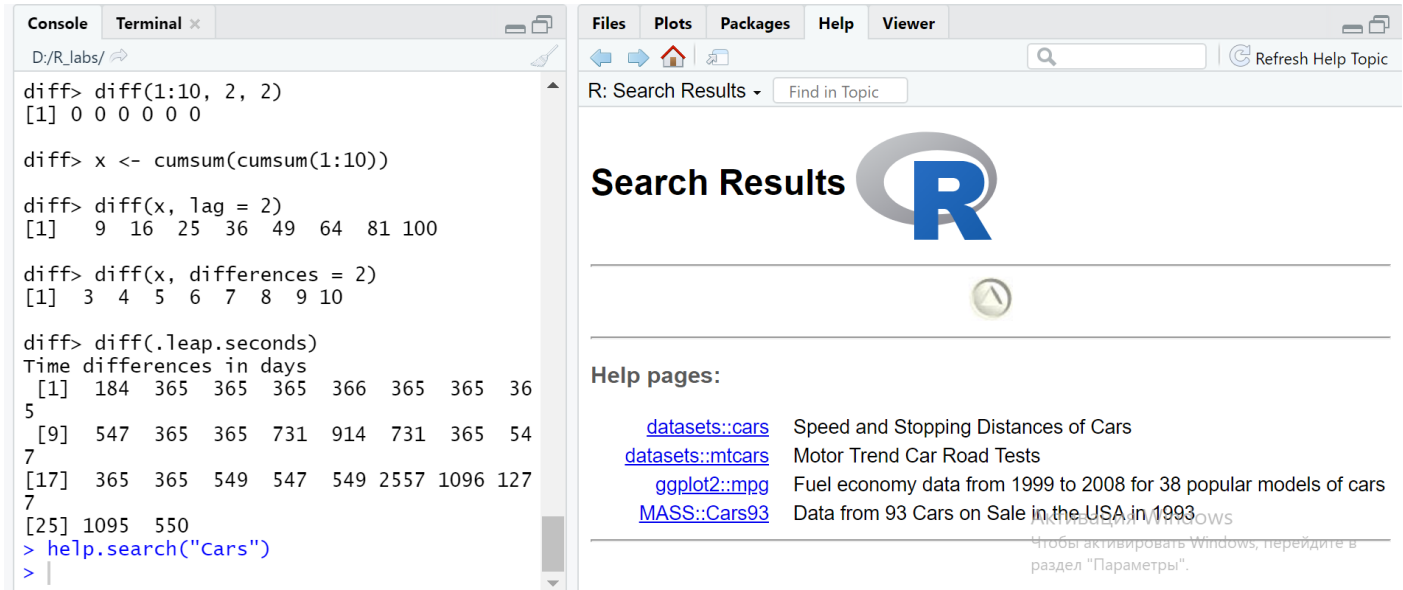
diff> x <- cumsum(cumsum(1:10))

diff> diff(x, lag = 2)
[1] 9 16 25 36 49 64 81 100

diff> diff(x, differences = 2)
[1] 3 4 5 6 7 8 9 10

diff> diff(.leap.seconds)
Time differences in days
[1] 184 365 365 365 366 365 365 365
[9] 547 365 365 731 914 731 365 547
[17] 365 365 549 547 549 2557 1096 1277
[25] 1095 550
```

Help.search("Cars")



The screenshot shows the R Studio interface. The console on the left contains the following R code and output:

```
diff> diff(1:10, 2, 2)
[1] 0 0 0 0 0 0

diff> x <- cumsum(cumsum(1:10))

diff> diff(x, lag = 2)
[1] 9 16 25 36 49 64 81 100

diff> diff(x, differences = 2)
[1] 3 4 5 6 7 8 9 10

diff> diff(.leap.seconds)
Time differences in days
[1] 184 365 365 365 366 365 365 36
5
[9] 547 365 365 731 914 731 365 54
7
[17] 365 365 549 547 549 2557 1096 127
7
[25] 1095 550
> help.search("Cars")
> |
```

The help window on the right displays the search results for "Cars". It features the R logo and a list of help pages:

Search Results

Help pages:

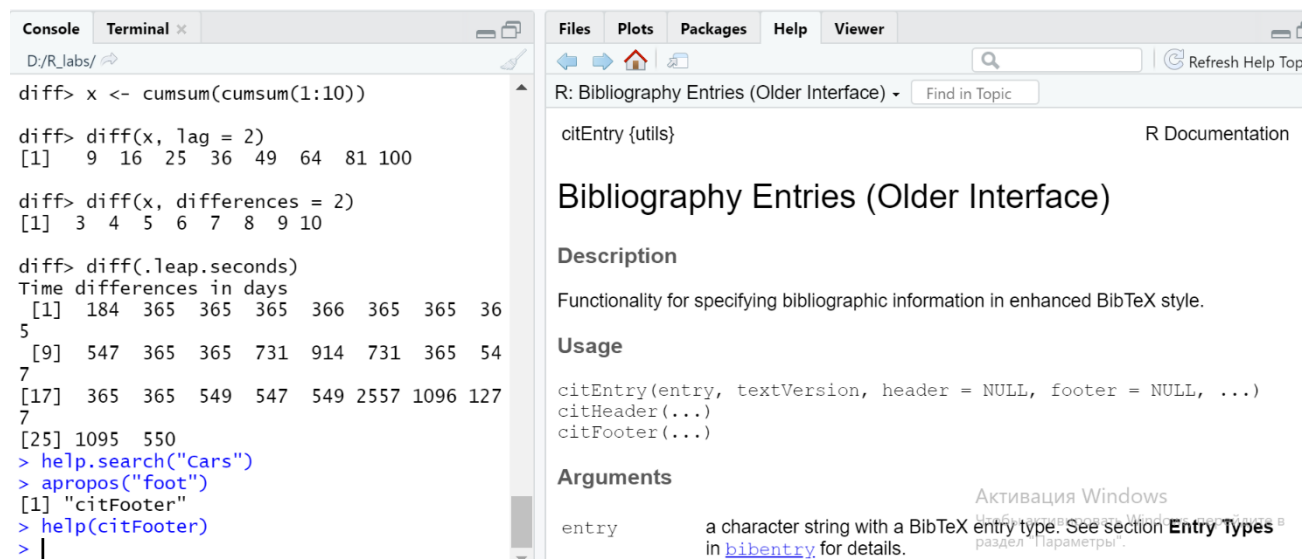
- [datasets::cars](#) Speed and Stopping Distances of Cars
- [datasets::mtcars](#) Motor Trend Car Road Tests
- [ggplot2::mpg](#) Fuel economy data from 1999 to 2008 for 38 popular models of cars
- [MASS::Cars93](#) Data from 93 Cars on Sale in the USA in 1993

At the bottom, there is a message: "Активация Windows. Чтобы активировать Windows, перейдите в раздел 'Параметры'."

Apropos("foot")

```
> apropos("foot")
[1] "citFooter"
> |
```

Help(citFooter)



The screenshot shows the R Studio interface. The console on the left contains the following R code and output:

```
diff> x <- cumsum(cumsum(1:10))

diff> diff(x, lag = 2)
[1] 9 16 25 36 49 64 81 100

diff> diff(x, differences = 2)
[1] 3 4 5 6 7 8 9 10

diff> diff(.leap.seconds)
Time differences in days
[1] 184 365 365 365 366 365 365 36
5
[9] 547 365 365 731 914 731 365 54
7
[17] 365 365 549 547 549 2557 1096 127
7
[25] 1095 550
> help.search("Cars")
> apropos("foot")
[1] "citFooter"
> help(citFooter)
> |
```

The help window on the right displays the documentation for "Bibliography Entries (Older Interface)". It includes the following sections:

R: Bibliography Entries (Older Interface)

citEntry {utils} R Documentation

Description

Functionality for specifying bibliographic information in enhanced BibTeX style.

Usage

```
citEntry(entry, textVersion, header = NULL, footer = NULL, ...)
citHeader(...)
citFooter(...)
```

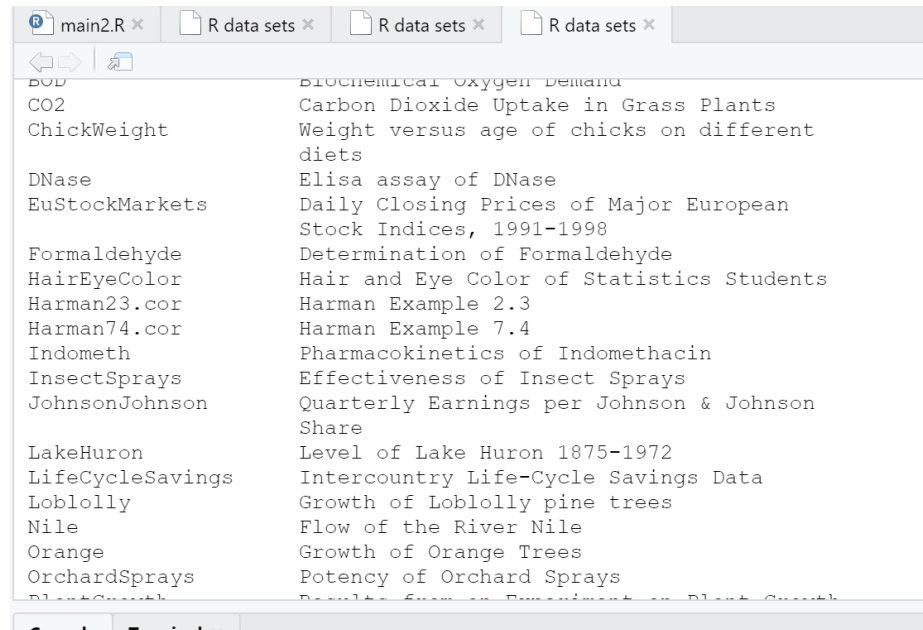
Arguments

entry a character string with a BibTeX entry type. See section **Entry Types** in [bibentry](#) for details.

At the bottom, there is a message: "Активация Windows. Чтобы активировать Windows, перейдите в раздел 'Параметры'."

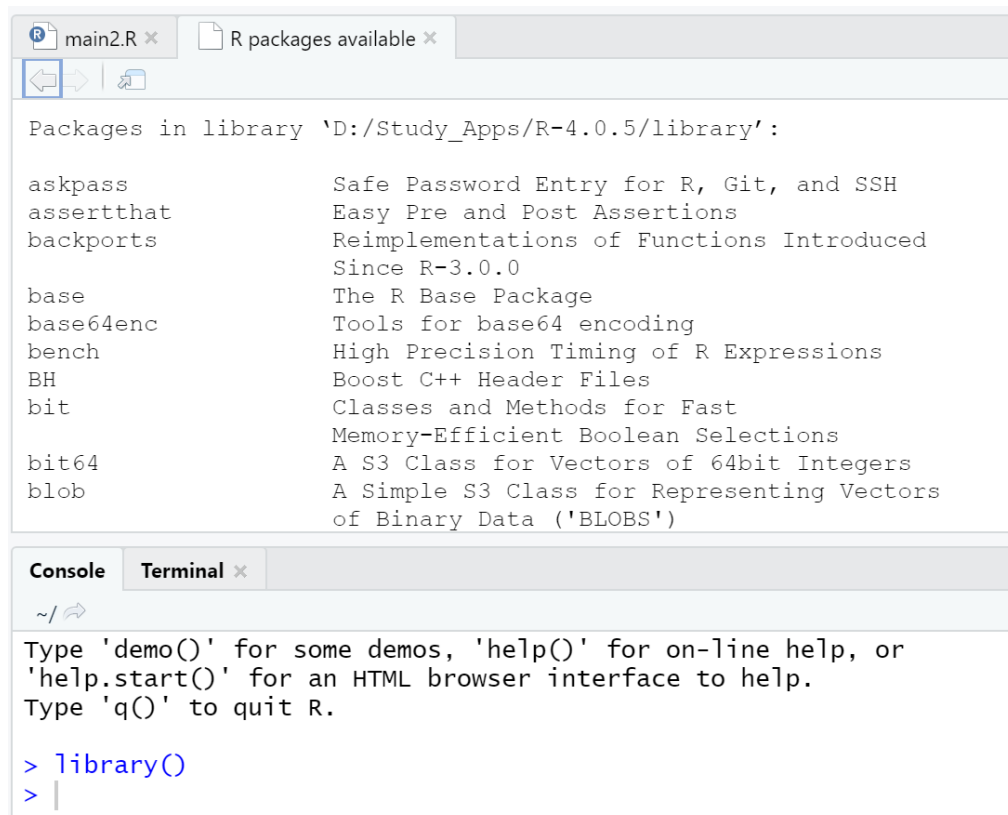
Завдання 3.

Data() – видає список усіх доступних датасетів



BOD	Biochemical Oxygen Demand
CO2	Carbon Dioxide Uptake in Grass Plants
ChickWeight	Weight versus age of chicks on different diets
DNase	Elisa assay of DNase
EuStockMarkets	Daily Closing Prices of Major European Stock Indices, 1991-1998
Formaldehyde	Determination of Formaldehyde
HairEyeColor	Hair and Eye Color of Statistics Students
Harman23.cor	Harman Example 2.3
Harman74.cor	Harman Example 7.4
Indometh	Pharmacokinetics of Indomethacin
InsectSprays	Effectiveness of Insect Sprays
JohnsonJohnson	Quarterly Earnings per Johnson & Johnson Share
LakeHuron	Level of Lake Huron 1875-1972
LifeCycleSavings	Intercountry Life-Cycle Savings Data
Loblolly	Growth of Loblolly pine trees
Nile	Flow of the River Nile
Orange	Growth of Orange Trees
OrchardSprays	Potency of Orchard Sprays
PlantGrowth	Results from an Experiment on Plant Growth

Library()



Packages in library 'D:/Study_Apps/R-4.0.5/library':	
askpass	Safe Password Entry for R, Git, and SSH
assertthat	Easy Pre and Post Assertions
backports	Reimplementations of Functions Introduced Since R-3.0.0
base	The R Base Package
base64enc	Tools for base64 encoding
bench	High Precision Timing of R Expressions
BH	Boost C++ Header Files
bit	Classes and Methods for Fast Memory-Efficient Boolean Selections
bit64	A S3 Class for Vectors of 64bit Integers
blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')

~/

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R.

```
> library()
> |
```


Обчислити значення виразу:

```
> exp(log(sin(1/2),base=7)-atan(sqrt(1+cos(0.05))))  
[1] 0.2636929  
> |
```

Завантажити дані у змінну вектор.

Виконати наступні завдання та скопіювати скрипти та результати їх виконання.

Script.R

```
#завантаження даних  
x = c(1, 8, 2, 6, 3, 8, 5, 5, 5, 5)  
x  
#створення вектора з інтервалу [5; 6]  
y<-x[x>=5 & x<=6]  
y  
#сума, добуток і різниця компонент  
sum(y)  
prod(y)  
diff(y)  
#вектор чисел від -15 до 100 що кратне 5  
z<-seq(-15,100,by=5)  
z  
#заміна кожного числа що кратне 10 на 0  
x<-replace(z,z%%10==0,c(0))  
x  
#виділення компонентів що більші за середнє значення  
y<-x[x>mean(x)]  
y  
#сума компонент вектора  
sum(y)  
#вектор з парною кількістю елементів  
vect<-c(-3,5,6,7,8,9,4,-4,-6,7,8,12,80,-5,0,6,21,-3,7,10)  
#матриця з двома рядками  
M1<-matrix(vect,nrow=2)  
M1  
#матриця з двома стовпцями  
M2<-matrix(vect,ncol=2)  
M2  
#дані для A  
vec <- numeric(18)  
dim(vec)<-c(3,6)  
A <- edit(vec)  
A  
#множення  
A*A  
#Введіть дані для вектора довжиною 12. Побудуйте з нього матрицю з трьох рядків і чотирьох  
стовпців . Транспонуйте матрицю B і позначте її через C. Позначте через D = BC .  
vect1<-c(0,3,2,9,-5,6,0,5,7,3,-3,0)  
B<-matrix(vect1,nrow = 3, ncol = 4, byrow = TRUE)  
B
```

```

C<-t(B)
C
D<-B%*%C
D
A<-C%*%B
A
#Знайдіть матрицю, обернену до D, за допомогою функції
solve(D)
#Введіть квадратну матрицю порядку 4. Елементи головної діагоналі замініть на 0, якщо вони
більші за середнє значення всіх елементів матриці.
vec <- numeric(16)
dim(vec)<-c(4,4)
A <- edit(vec)
A
diag(A)<-replace(diag(A), diag(A)> mean(A), c(0))
A
a<-c(0,3,-2)
y<-solve(A,a) #розв'язання лінійного рівняння.
y
#Запишіть вектор з 20 значень. Знайдіть його квартилі. Розбийте компонентивектора на 4
інтервали (межі - квартилі). Перетворіть вектор на фактор, де рівнями будуть інтервали, в які
попадають значення вектора.
vect<-c(-3,5,6,7,8,9,4,-4,-6,7,8,12,80,-5,0,6,21,-3,7,10) #створення вектора
vect
g<-quantile(vect,c(0,0.25,0.5,0.75,1))
g
u<-cut(vect,breaks=g,labels=c('Перший','Другий','Третій','Четвертий'),include.lowest=TRUE)
u
#Введіть два вектора однакової довжини (наприклад, 15).
#Створити фактор, використавши ці два вектора.
vik<-c(21,18,76,35,72,29,45,67,43,23,25,87,45,24,25)
x<-sample(c(0,1),15,replace=TRUE)
x
mean(vik)
#З набору даних Loblolly вибрати стовпчики, які відповідають висоті і віку сосен.
#Утворити числовий вектор, який складається з середньої висоти дерева для таких
# категорій віку сосни: до 10 років, 10-15, 16-20, більше 20.
x<-
c(mean(Loblolly$height[Loblolly$age<10]),mean(Loblolly$height[Loblolly$age>10&Loblolly$age<16]),
,mean(Loblolly$height[Loblolly$age>15&Loblolly$age<21]),mean(Loblolly$height[Loblolly$age>20]))
x
#З набору даних Loblolly вибрати тільки ті дані, які відповідають насінню типу
#301 і 315. Зеленою пунктирною лінією зобразити графік залежності висоти дерева
#для насіння 301 від віку сосни, а неперервною фіолетовою лінією - для насіння 315.
#Зобразити в одному вікні. Підписати рисунок.
h301<-Loblolly$height[Loblolly$Seed==301]
a301<-Loblolly$age[Loblolly$Seed==301]
a315<-Loblolly$age[Loblolly$Seed==315]
h315<-Loblolly$height[Loblolly$Seed==315]
plot(a301,h301,type='l',lty=5,col='green',ylab='Висота дерева', xlab="Вік дерева",main="Графік
залежності висоти дерева від віку")
lines(a315,h315, add=T,col = "violet")

```

Консоль

```
> #завантаження даних
There were 20 warnings (use warnings() to see them)
> x = c(1, 8, 2, 6, 3, 8, 5, 5, 5, 5)
> x
[1] 1 8 2 6 3 8 5 5 5 5
> #створення вектора з інтервалу [5; 6]
> y<-x[x>=5 & x<=6]
> y
[1] 6 5 5 5 5
> #сума, добуток і різниця компонент
> sum(y)
[1] 26
> prod(y)
[1] 3750
> diff(y)
[1] -1 0 0 0
> #вектор чисел від -15 до 100 що кратне 5
> z<-seq(-15,100,by=5)
> z
[1] -15 -10 -5 0 5 10 15 20 25 30 35 40 45
[14] 50 55 60 65 70 75 80 85 90 95 100
> #заміна кожного числа що кратне 10 на 0
> x<-replace(z,z%%10==0,c(0))
> x
[1] -15 0 -5 0 5 0 15 0 25 0 35 0 45
[14] 0 55 0 65 0 75 0 85 0 95 0
> #виділення компонентів що більші за середнє значення
> y<-x[x>mean(x)]
> y
[1] 25 35 45 55 65 75 85 95
> #сума компонент вектора
> sum(y)
[1] 480
> #вектор з парною кількістю елементів
> vect<-c(-3,5,6,7,8,9,4,-4,-6,7,8,12,80,-5,0,6,21,-3,7,10)
> #матриця з двома рядками
> M1<-matrix(vect,nrow=2)
> M1
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   -3    6    8    4   -6    8   80    0   21    7
[2,]    5    7    9   -4    7   12   -5    6   -3   10
> #матриця з двома стовпцями
> M2<-matrix(vect,ncol=2)
> M2
      [,1] [,2]
[1,]   -3    8
[2,]    5   12
[3,]    6   80
[4,]    7   -5
[5,]    8    0
[6,]    9    6
[7,]    4   21
[8,]   -4   -3
[9,]   -6    7
[10,]    7   10
> #дані для A
> vec <- numeric(18)
```

```
> dim(vec)<-c(3,6)
> A <- edit(vec)
```

Data Editor									
File Edit Help									
	col1	col2	col3	col4	col5	col6	var7	var8	var9
1	4	3	2	1	5	4			
2	13	3	6	9	8	1			
3	4	12	5	6	7	10			
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									

Консоль (продовження після вводу матриці)

```
> A <- edit(vec)
> A
      col1 col2 col3 col4 col5 col6
[1,]    4    3    2    1    5    4
[2,]   13    3    6    9    8    1
[3,]    4   12    5    6    7   10
> #множення
> A*A
      col1 col2 col3 col4 col5 col6
[1,]   16    9    4    1   25   16
[2,]  169    9   36   81   64    1
[3,]   16  144   25   36   49  100
> #Введіть дані для вектора довжиною 12. Побудуйте з нього матрицю з трьох рядків і чотирьох
стовпців . Транспонуйте матрицю B і позначте її через C. Позначте через D = BC .
> vect1<-c(0,3,2,9,-5,6,0,5,7,3,-3,0)
> B<-matrix(vect1,nrow = 3, ncol = 4, byrow = TRUE)
> B
```

```

      [,1] [,2] [,3] [,4]
[1,]    0    3    2    9
[2,]   -5    6    0    5
[3,]    7    3   -3    0
> C<-t(B)
> C
      [,1] [,2] [,3]
[1,]    0   -5    7
[2,]    3    6    3
[3,]    2    0   -3
[4,]    9    5    0
> D<-B%*%C
> D
      [,1] [,2] [,3]
[1,]   94   63    3
[2,]   63   86  -17
[3,]    3  -17   67
> A<-C%*%B
> A
      [,1] [,2] [,3] [,4]
[1,]   74   -9  -21  -25
[2,]   -9   54   -3   57
[3,]  -21   -3   13   18
[4,]  -25   57   18  106
> #Знайдіть матрицю, обернену до D, за допомогою функції
> solve(D)
      [,1]      [,2]      [,3]
[1,]  0.022677644 -0.017701242 -0.005506777
[2,] -0.017701242  0.026058780  0.007404522
[3,] -0.005506777  0.007404522  0.017050705
> #Введіть квадратну матрицю порядку 4. Елементи головної діагоналі замініть на 0, якщо вони
більші за середнє значення всіх елементів матриці.
> vec <- numeric(16)
> dim(vec)<-c(4,4)
> A <- edit(vec)

```

Data Editor								
File Edit Help								
	col1	col2	col3	col4	var5	var6	var7	var8
1	6	4	12	1				
2	1	8	0	7				
3	2	3	10	0				
4	9	3	5	4				
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

Активация Windows

Чтобы активировать Windows, перейдите на сайт Microsoft.com

Консоль (продовження після вводу другої матриці)

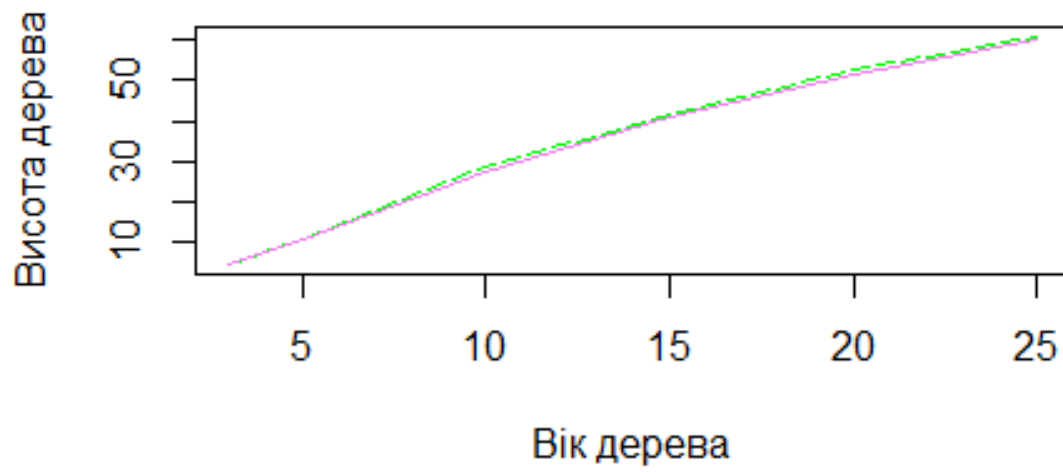
```
> A <- edit(vec)
> A
      col1 col2 col3 col4
[1,]     6     4    12     1
[2,]     1     8     0     7
[3,]     2     3    10     0
[4,]     9     3     5     4
> diag(A) <- replace(diag(A), diag(A) > mean(A), c(0))
> A
      col1 col2 col3 col4
[1,]     0     4    12     1
[2,]     1     0     0     7
[3,]     2     3     0     0
[4,]     9     3     5     4
> a <- c(0, 3, -2)
> y <- solve(A, a) #розв'язання лінійного рівняння.
> y
[1] 25 35 45 55 65 75 85 95
> #Запишіть вектор з 20 значень. Знайдіть його квартилі. Розбийте компонентивектора на 4
інтервали (межі - квартилі). Перетворіть вектор на фактор, де рівнями будуть інтервали, в які
попадають значення вектора.
> vect <- c(-3, 5, 6, 7, 8, 9, 4, -4, -6, 7, 8, 12, 80, -5, 0, 6, 21, -3, 7, 10) #створення вектора
> vect
[1] -3  5  6  7  8  9  4 -4 -6  7  8 12 80 -5  0  6 21 -3  7 10
[18] -3  7 10
```

```

> g<-quantile(vect,c(0,0.25,0.5,0.75,1))
> g
  0%   25%   50%   75%  100%
-6.00 -0.75  6.50  8.25 80.00
> u<-cut(vect,breaks=g,labels=c('Перший','Другий','Третій','Четвертий'),include.lowest=TRUE)
> u
 [1] Перший    Другий    Другий    Третій    Третій
 [6] Четвертий Другий    Перший    Перший    Третій
[11] Третій     Четвертий Четвертий Перший    Другий
[16] Другий     Четвертий Перший    Третій    Четвертий
Levels: Перший Другий Третій Четвертий
> #Введіть два вектора однакової довжини (наприклад, 15).
> #Створити фактор, використавши ці два вектора.
> vik<-c(21,18,76,35,72,29,45,67,43,23,25,87,45,24,25)
> x<-sample(c(0,1),15,replace=TRUE)
> x
 [1] 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1
> mean(vik)
[1] 42.33333
> #3 набору даних Loblolly вибрати стовпчики, які відповідають висоті і віку сосен.
> #Утворити числовий вектор, який складається з середньої висоти дерева для таких
> # категорій віку сосни: до 10 років, 10-15, 16-20, більше 20.
> x<-
c(mean(Loblolly$height[Loblolly$age<10]),mean(Loblolly$height[Loblolly$age>10&Loblolly$age<16]),
,mean(Loblolly$height[Loblolly$age>15&Loblolly$age<21]),mean(Loblolly$height[Loblolly$age>20]))
> x
 [1] 7.221429 40.543571 51.468571 60.289286
> #3 набору даних Loblolly вибрати тільки ті дані, які відповідають насінню типу
> #301 і 315. Зеленою пунктирною лінією зобразити графік залежності висоти дерева
> #для насіння 301 від віку сосни, а неперервною фіолетовою лінією - для насіння 315.
> #Зобразити в одному вікні. Підписати рисунок.
> h301<-Loblolly$height[Loblolly$Seed==301]
> a301<-Loblolly$age[Loblolly$Seed==301]
> a315<-Loblolly$age[Loblolly$Seed==315]
> h315<-Loblolly$height[Loblolly$Seed==315]
> plot(a301,h301,type='l',lty=5,col='green',ylab='Висота дерева', xlab="Вік
дерева",main="Графік залежності висоти дерева від віку")
> lines(a315,h315, add=T,col = "violet")
>

```

Графік залежності висоти дерева від віку



Результат: побудований графік

Частина 2. Імпорт, аналіз та візуалізація даних в R

Імпорт даних

Environment	History	Connections	Tutorial
Import Dataset			
R Global Environment			
Data			
A	num [1:4, 1:4]	0 1 2 9 4 0 3 3 12 0 ...	
B	num [1:3, 1:4]	0 -5 7 3 6 3 2 0 -3 9 ...	
C	num [1:4, 1:3]	0 3 2 9 -5 6 0 5 7 3 ...	
D	num [1:3, 1:3]	94 63 3 63 86 -17 3 -17 67	
flats	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 ...	

839 спостережень, стовпці «Місто», «Кімнат», «Загальна_площа», «Ціна»

```
> 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 ...
>
```

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

```
> dim(flats)
[1] 839 4
```

Вимірність датасету

```
> head(flats)
      Місто Кімнат Загальна_площа  Ціна
1 Вінниця      3      120 1875000
2 Вінниця      3       66  975000
3 Вінниця      2       66 1375000
4 Вінниця      2       44  637500
5 Вінниця      3       63  835000
6 Вінниця      1       31  562500
```

Перша частина об'єкта

```
> tail(flats)
      Місто Кімнат Загальна_площа  Ціна
834 Хмельницький  1      35.58 212500
835 Хмельницький  1      52.00 330000
836 Хмельницький  1      41.00 325000
837 Хмельницький  1      47.00 375000
838 Хмельницький  2      53.00 387500
839 Хмельницький  2      60.00 522500
>
```

Остання частина об'єкта

```
> names(flats)
[1] "Місто"      "Кімнат"     "Загальна_площа" "Ціна"
```

Імена пов'язані з об'єктом

Трансформація даних

```
> 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 ...

> 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

> glimpse(flats)
Rows: 839
Columns: 4
$ Місто      <chr> "Вінниця", "Вінниця", "Вінниця", "Вінниця", "Вінниця", "~
$ Кімнат     <int> 3, 3, 2, 2, 3, 1, 3, 3, 1, 6, 2, 1, 1, 2, 3, 3, 3, 2, 1,~
$ Загальна_площа <dbl> 120.00, 66.00, 66.00, 44.00, 63.00, 31.00, 46.00, 64.00,~
$ Ціна       <dbl> 1875000, 975000, 1375000, 637500, 835000, 562500, 115000~
> |
```

Кількість квартир:

```
> count(flats, Місто)
      Місто      n
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
```

Сортування даних по кількості квартир в місті в зростаючому порядку

```
> flats %>%  
+ count(Місто) %>%  
+ arrange(n)
```

	Місто	n
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

Вилучення даних по Києво-Святошинському району з відображення

```
> flats %>%  
+ filter(Місто != "Києво-Святошинський") %>%  
+ filter(Кімнат == 3) %>%  
+ count(Місто) %>%  
+ arrange(desc(n))
```

	Місто	n
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

Квартири з кількістю кімнат 2:

```
> flats %>%  
+ filter(Кімнат == 2) %>%  
+ filter(Місто != "Києво-Святошинський") %>%  
+ count(Місто) %>%  
+ arrange(desc(n))
```

	Місто	n
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

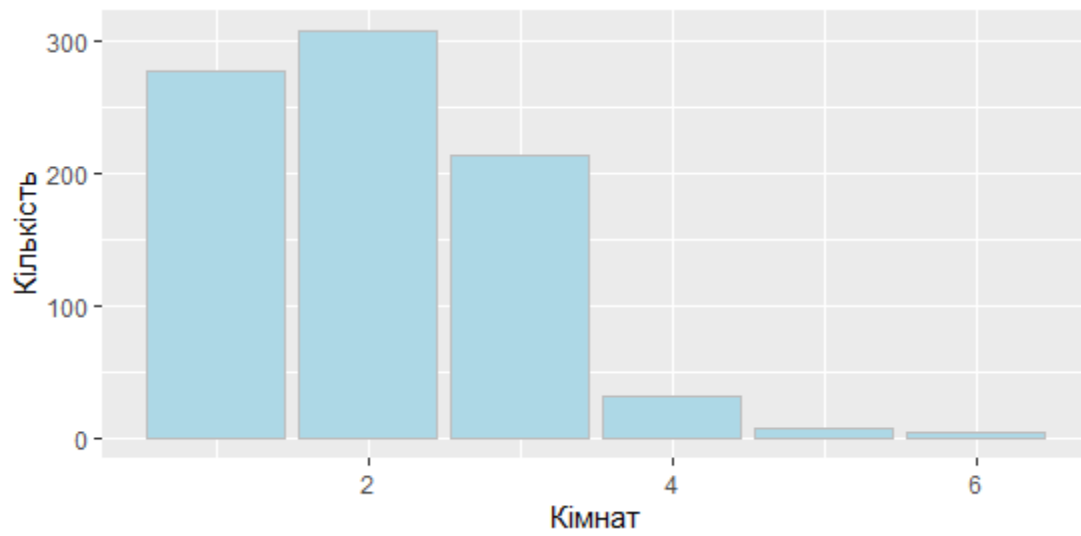
Активация \

Чтобы активирс

раздел "Параме

Візуалізація даних

Стовпчикова діаграма для кількості кімнат:

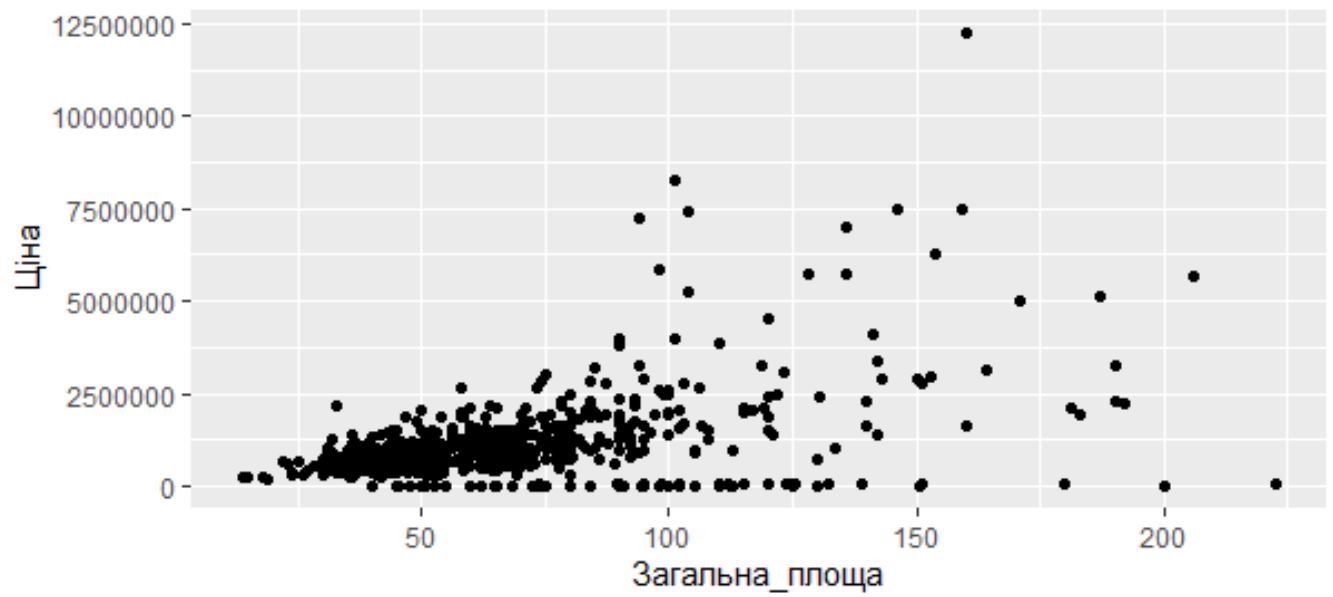


Стовпчикова діаграма для змінної Загальна_площа:



Графік розсіювання

Графік залежності ціни від загальної площі



Висновки

В ході виконання лабораторної роботи була освоєна мова програмування R для аналізу та візуалізації даних та використана для маніпуляції з даними з датасету та для графічного їх представлення.