

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 IІІ курсу каф. ПЗКС ФПМ

групи КП-82

Мельничук Олексій Геннадійович

Перевірила

доц. каф. ПЗКС ФПМ

Олещенко Л.М.

Київ 2021

1. **Індивідуальне завдання**

**Мета**: ознайомитись з можливостями мови програмування R для аналізу та

візуалізації даних.

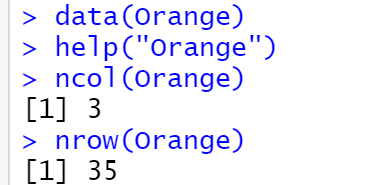
1. **Хід роботи**

**Практичне завдання**

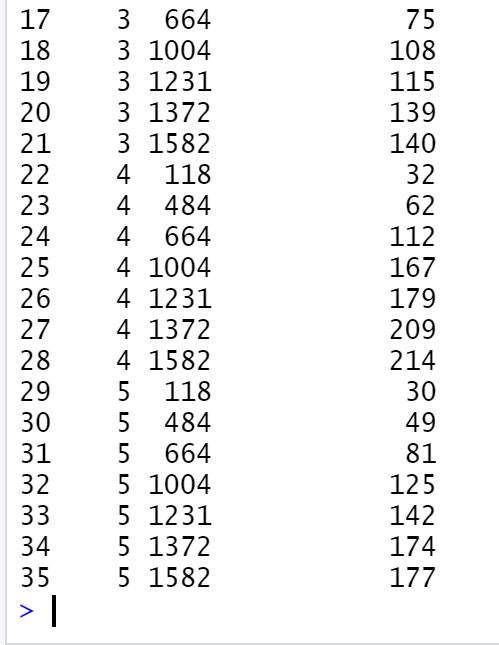
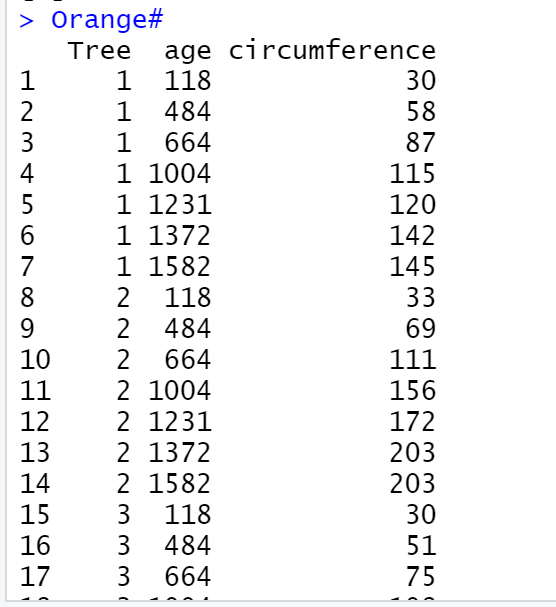
**Частина 1.**

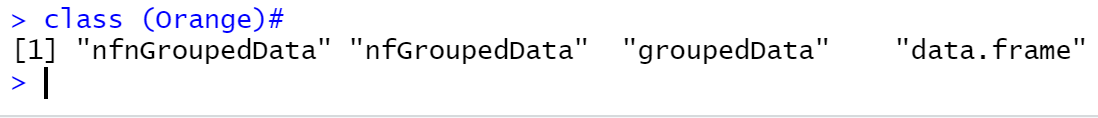
**Завдання 1.**

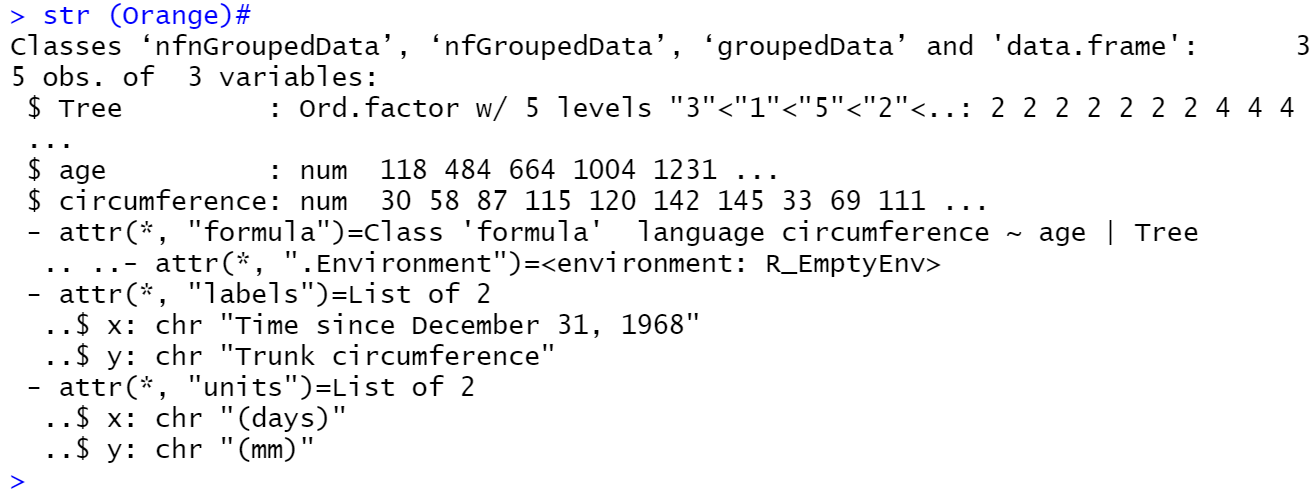
Скільки рядків та стовпців містить цей датафрейм?

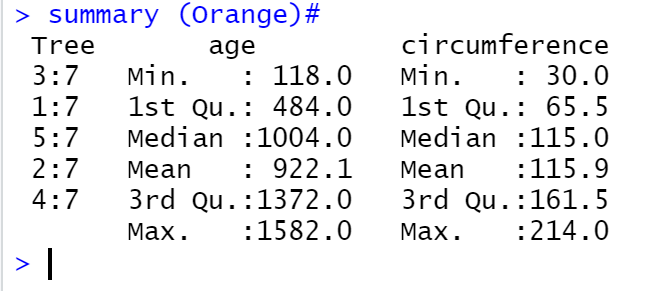


Orange # (навести скріншот)

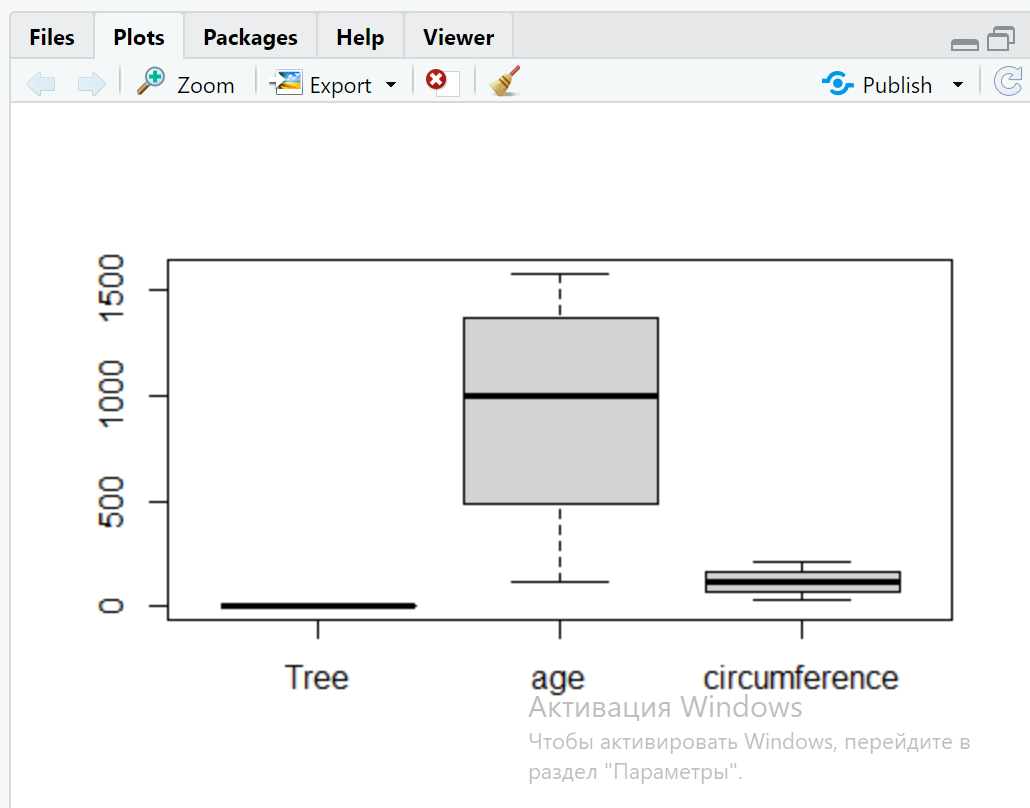


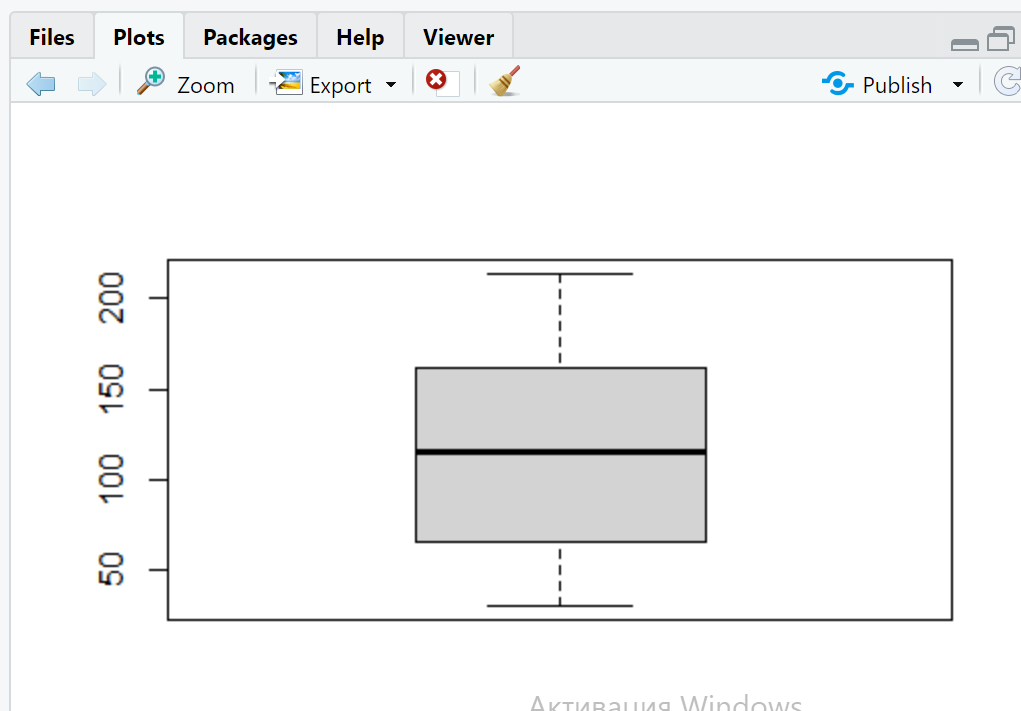
Class (Orange) # (навести скріншот)

str (Orange) # (навести скріншот)

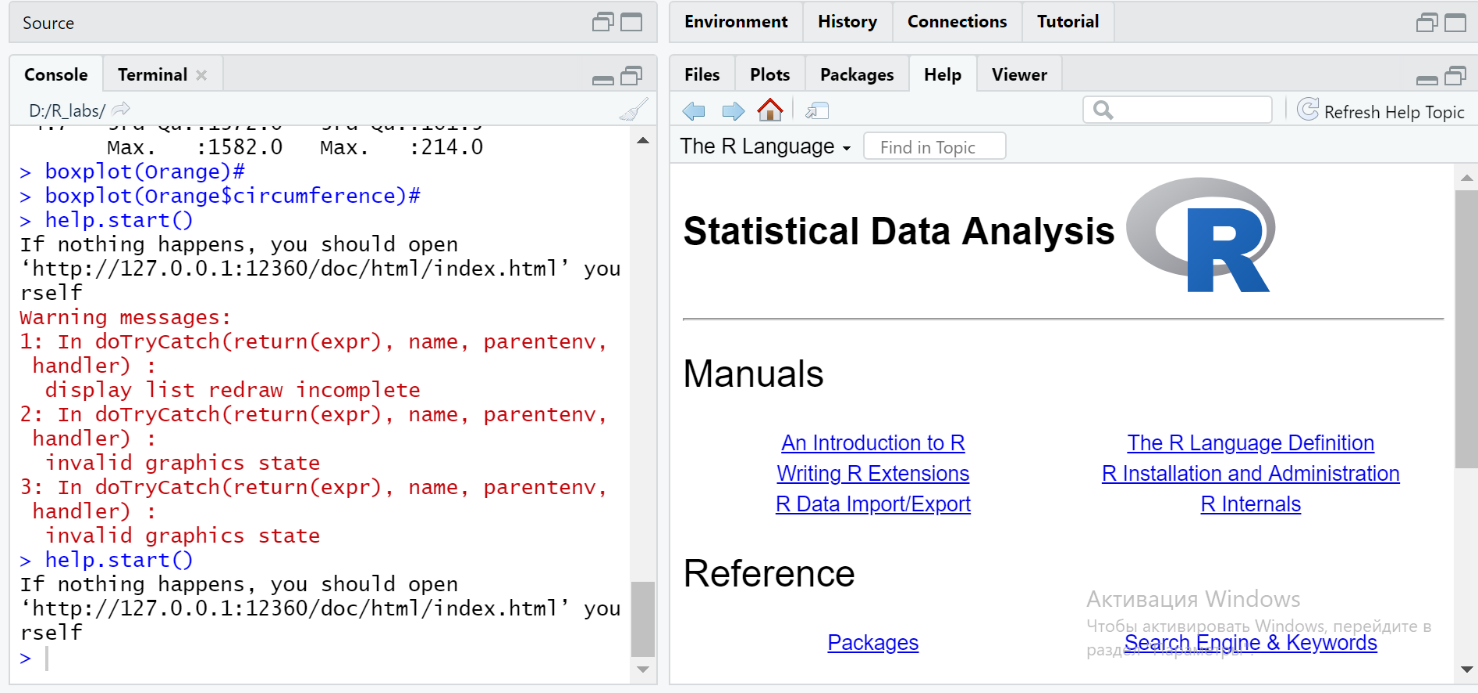
summary (Orange) # (навести скріншот)

boxplot (Orange) # (навести скріншот)

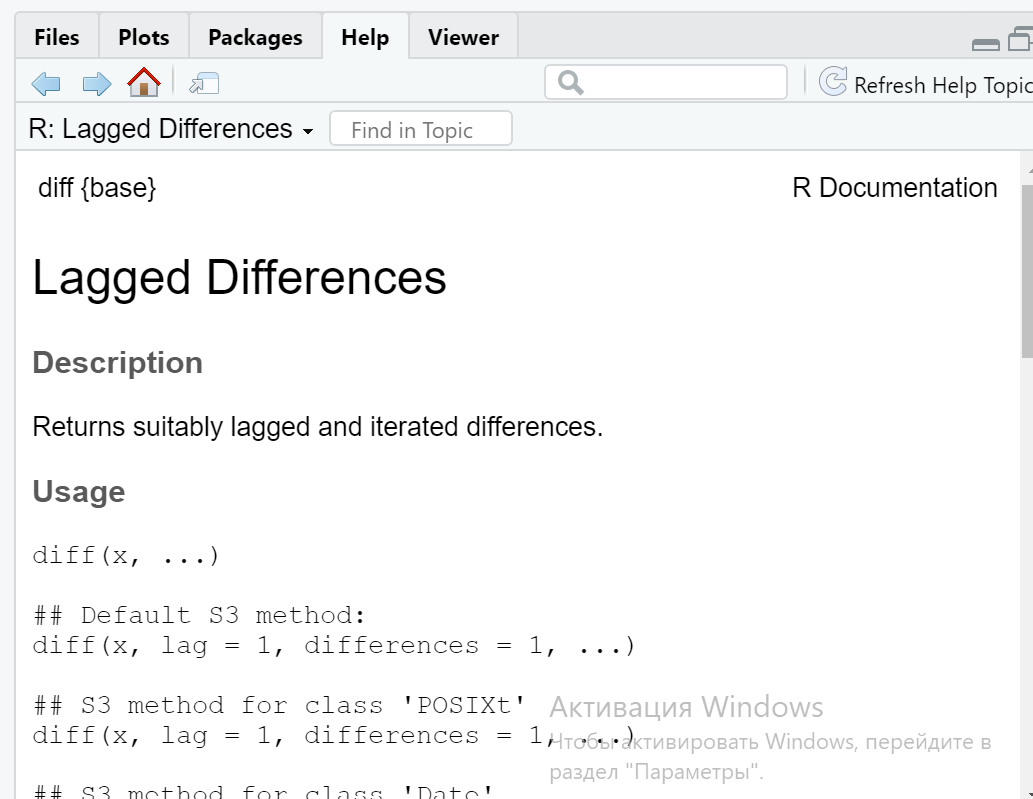


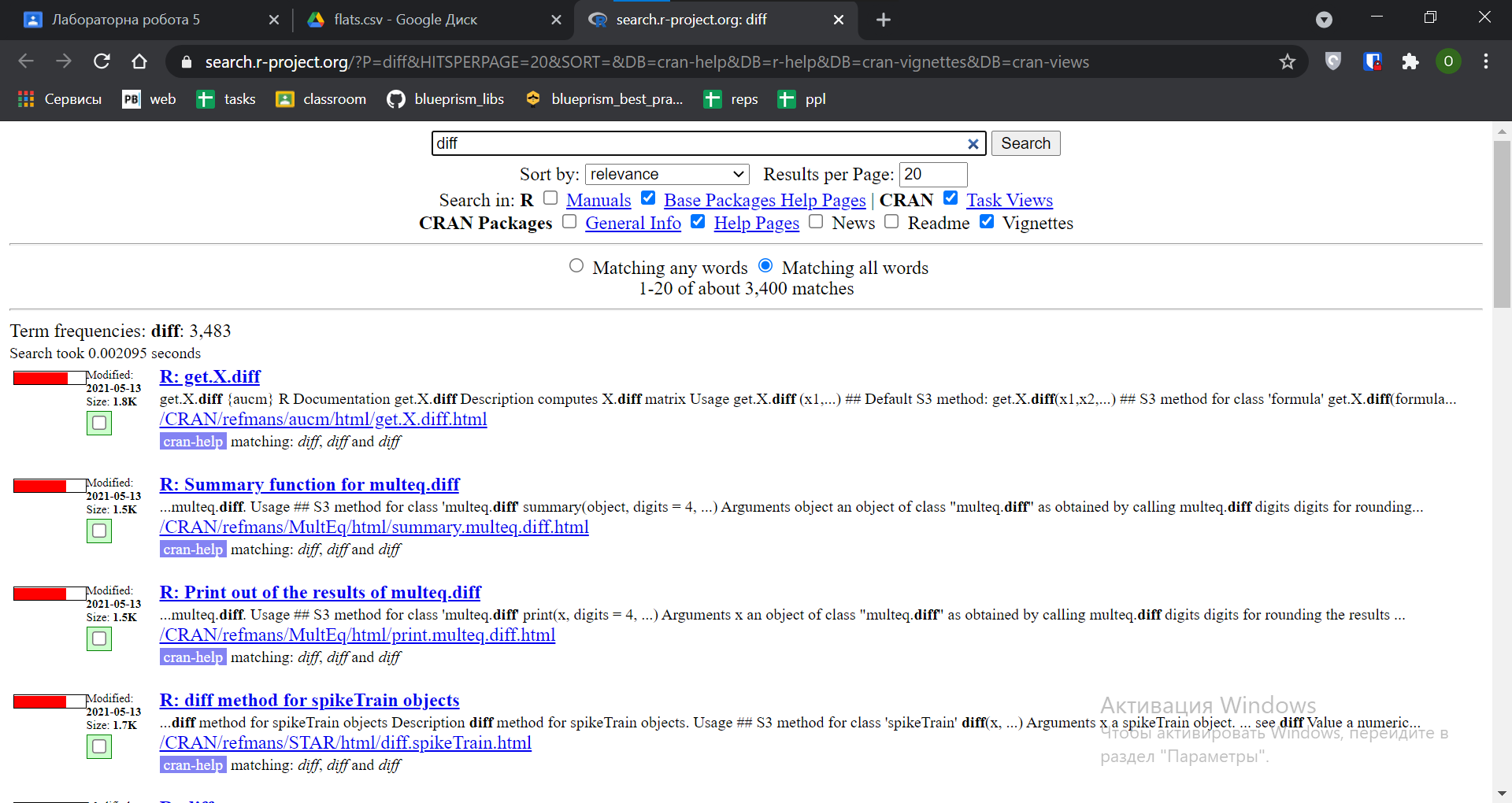
boxplot (Orange$circumference) # (навести скріншот)

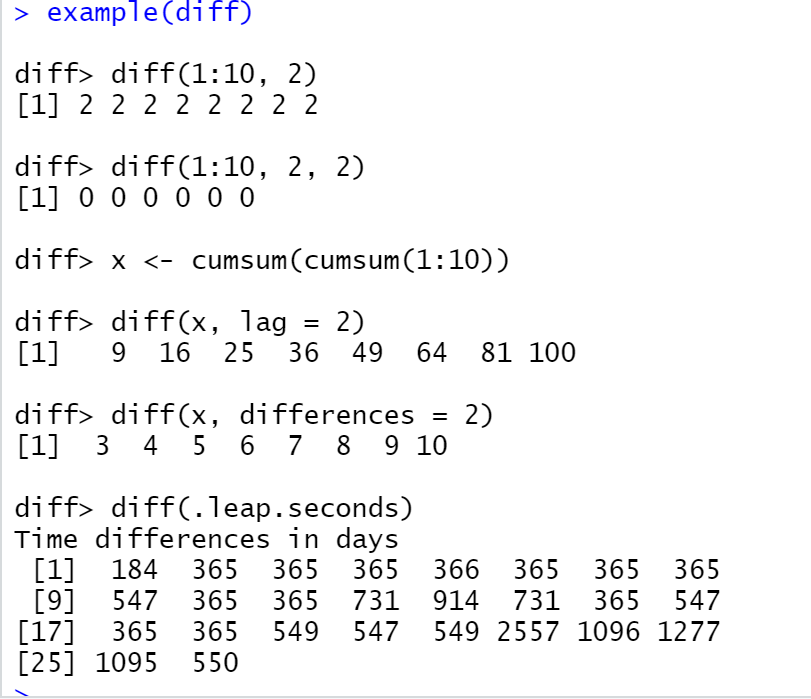
**Завдання 2.**

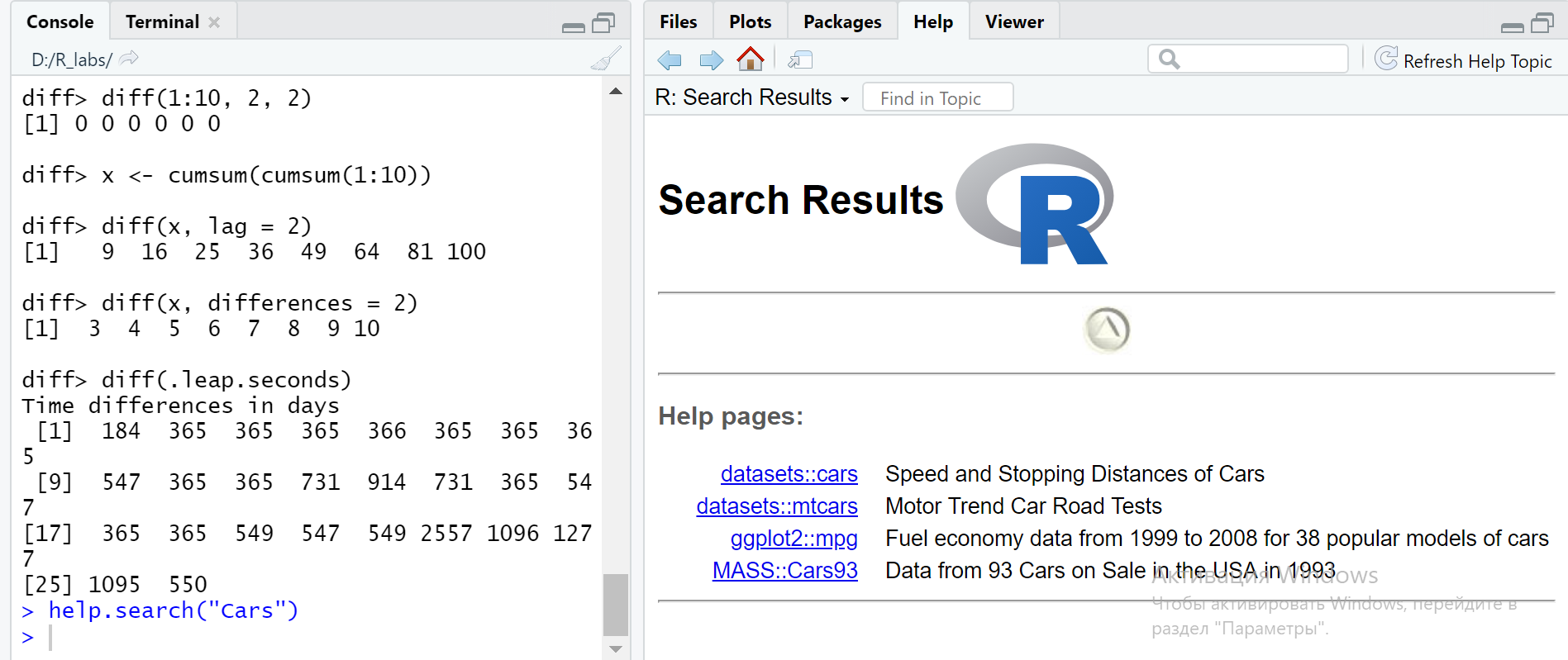
Help.start() (навести скріншот)

Help(diff)

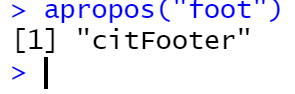


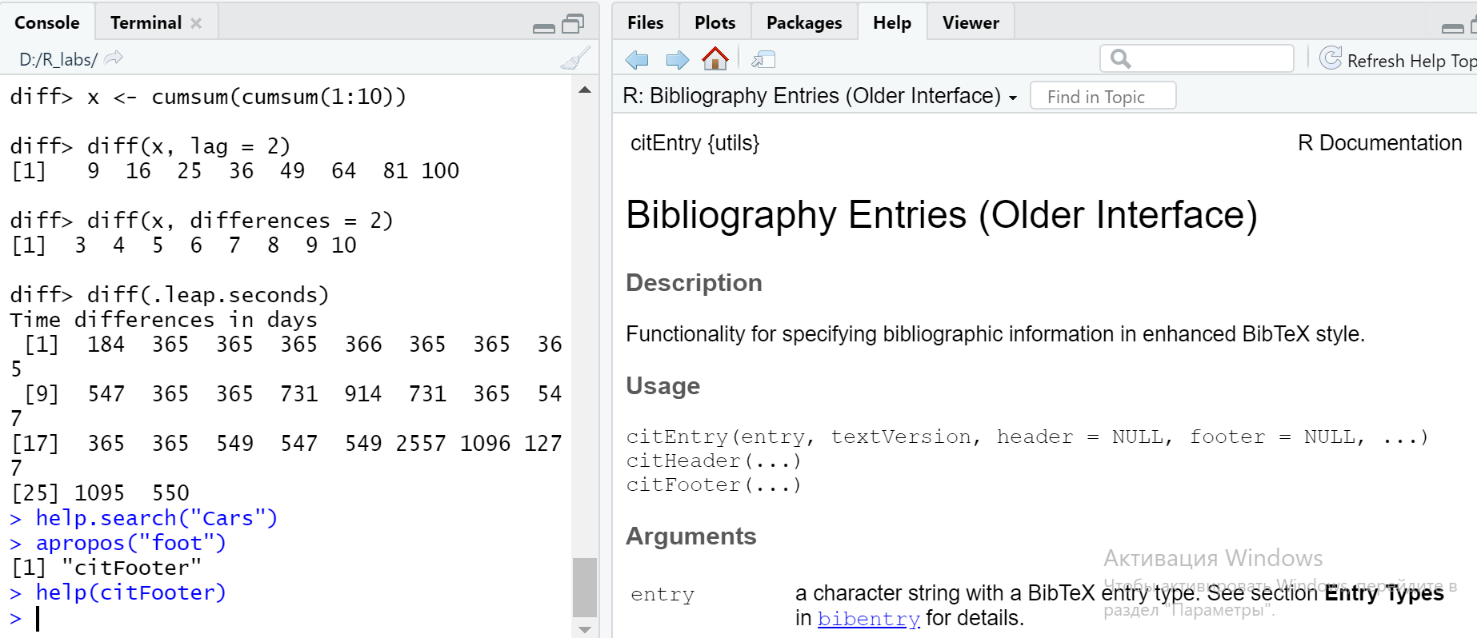
RSiteSearch("diff")

example(diff)

Help.search(“Cars”)

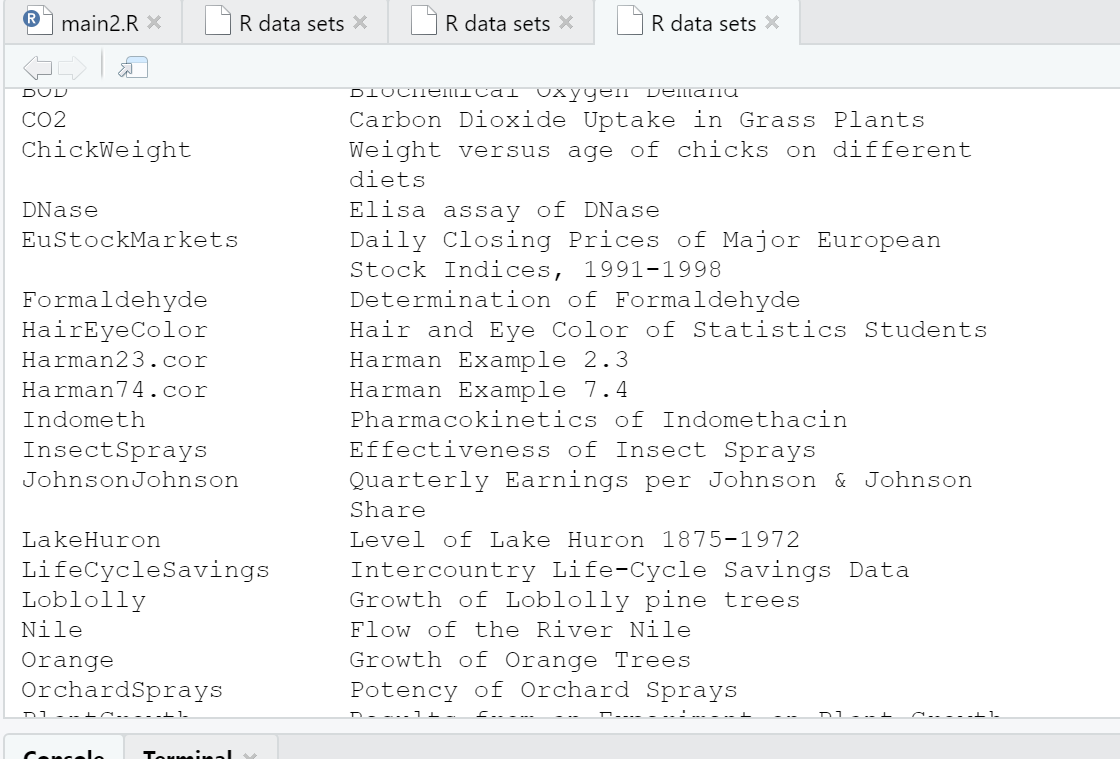
Apropos(“foot”)



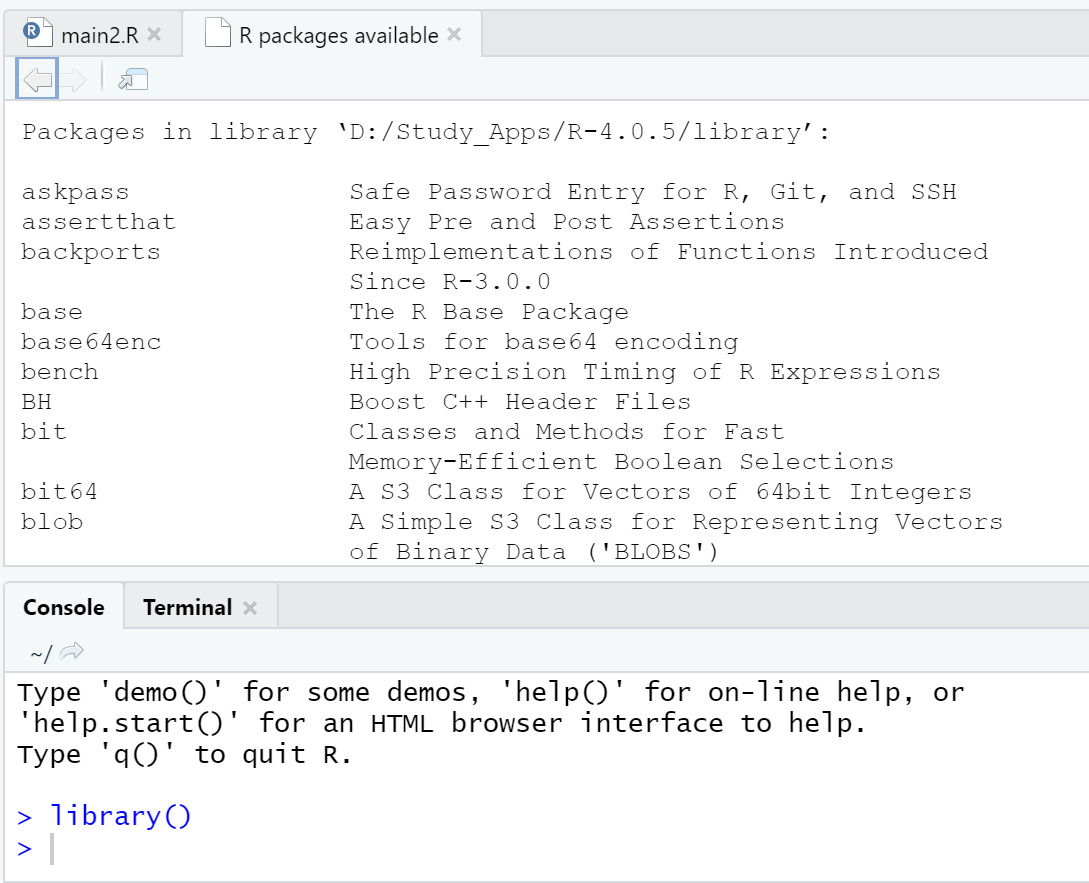
Help(citFooter)

**Завдання 3.**

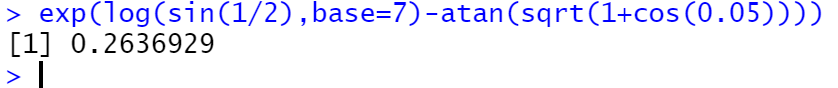
Data() – видає список усіх доступних датасетів



Library()



Обчислити значення виразу:

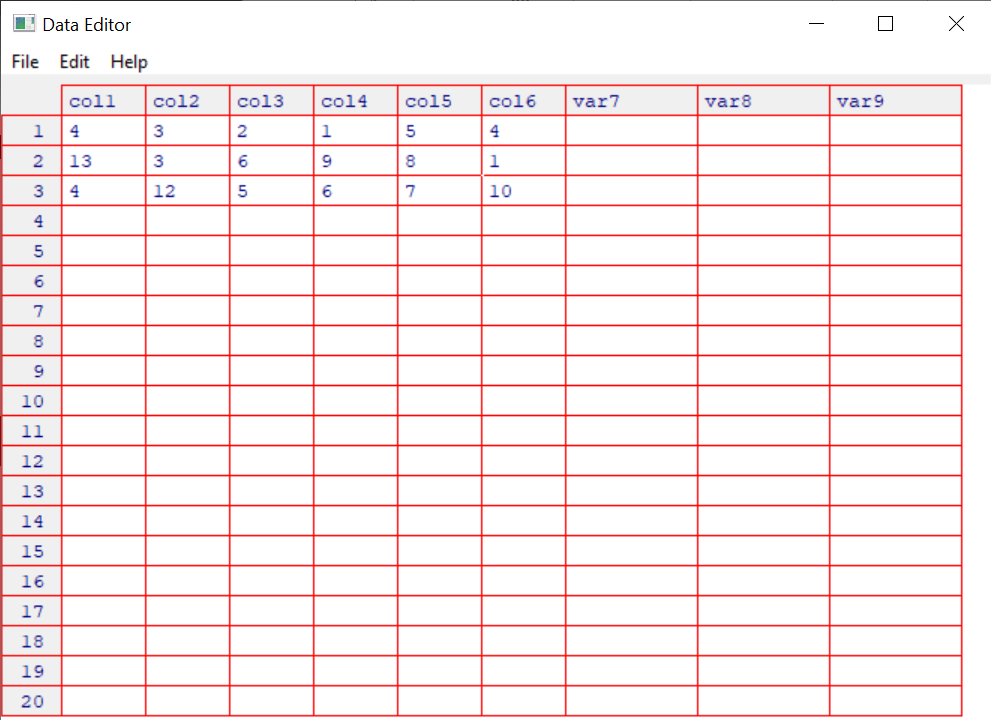


Завантижити дані у змінну вектор.

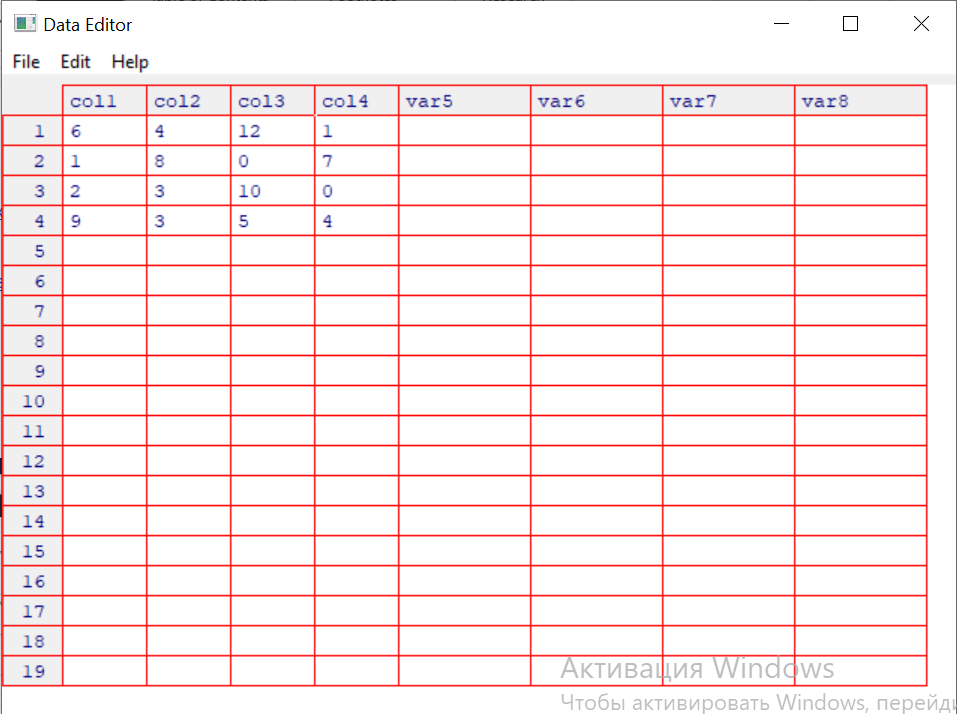
Виконати наступні завдання та скопіювати скрипти та результати їх виконання.

|  |
| --- |
| **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 В і позначте її через С. 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 квартілі. Розбийте компонентивектора на 4 інтервали (межі – квартілі). 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") |

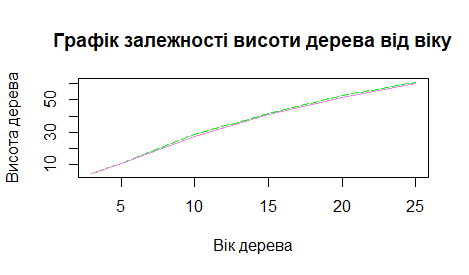
|  |
| --- |
| **Консоль** |
| > #завантаження даних  There were 20 warnings (use warnings() to see them)  > x = c(1, 8, 2, 6, 3, 8, 5, 5, 5, 5)  > x  [1] 1 8 2 6 3 8 5 5 5 5  > #створення вектора з інтервалу [5; 6]  > y<-x[x>=5 & x<=6]  > y  [1] 6 5 5 5 5  > #сума, добуток і різниця компонент  > sum(y)  [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) |



|  |
| --- |
| **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 В і позначте її через С. 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) |

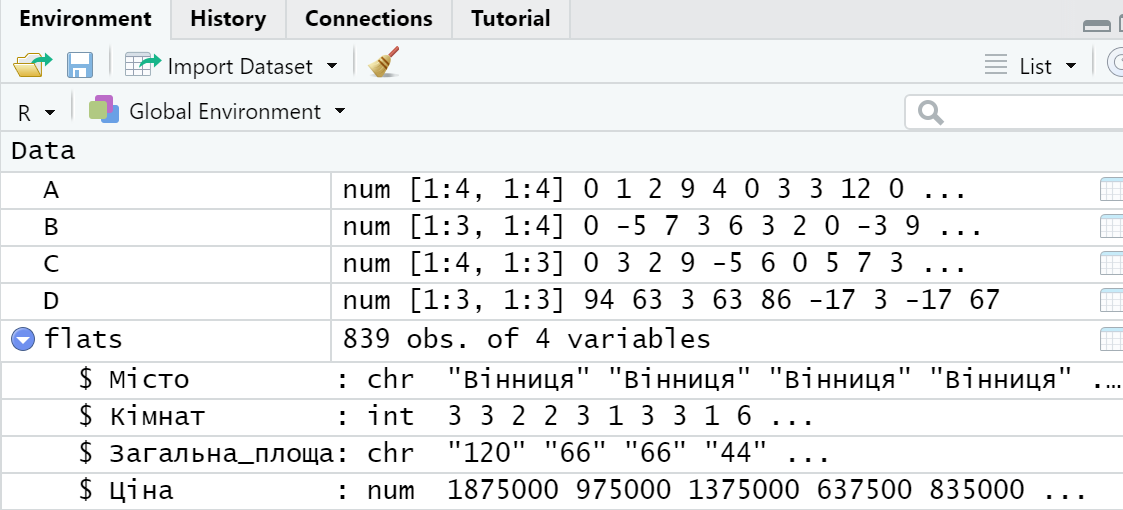


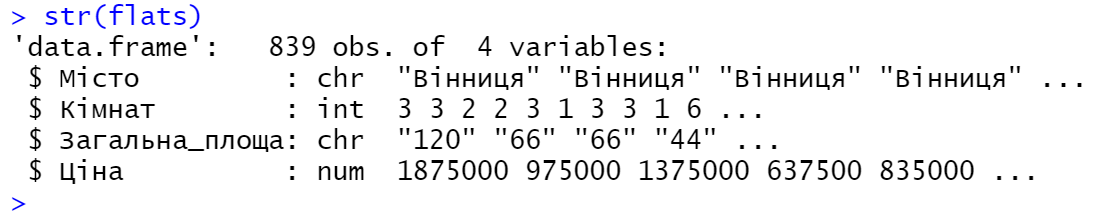
|  |
| --- |
| **Консоль (продовження після вводу другої матриці)** |
| > A <- edit(vec)  > A  col1 col2 col3 col4  [1,] 6 4 12 1  [2,] 1 8 0 7  [3,] 2 3 10 0  [4,] 9 3 5 4  > diag(A)<-replace(diag(A), diag(A)> mean(A), c(0))  > A  col1 col2 col3 col4  [1,] 0 4 12 1  [2,] 1 0 0 7  [3,] 2 3 0 0  [4,] 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 квартілі. Розбийте компонентивектора на 4 інтервали (межі – квартілі). 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  [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  > #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  [1] 7.221429 40.543571 51.468571 60.289286  > #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")  > |

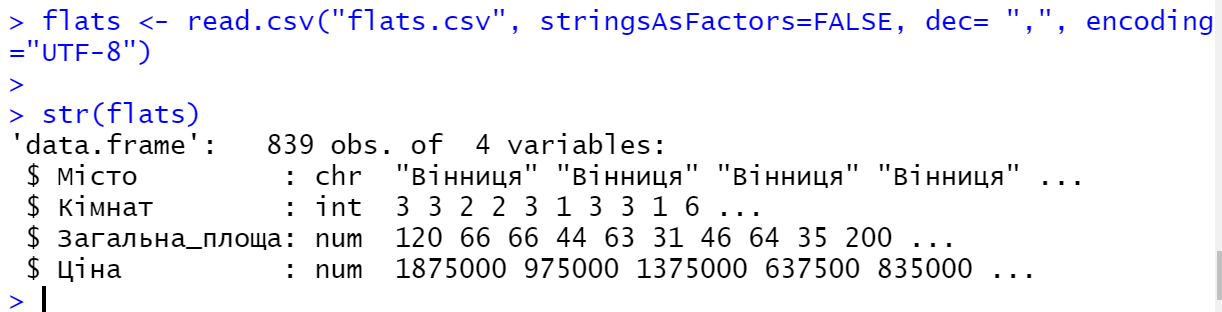
****Результат: побудований графік

**Частина 2. Імпорт, аналіз та візуалізація даних в R**

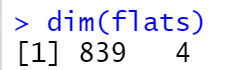
**Імпорт даних**

****

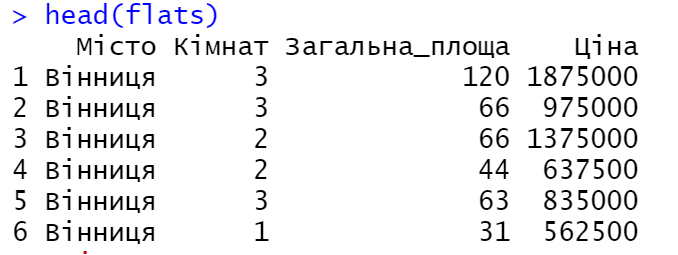
839 спостережень, стовпці «Місто», «Кімнат», «Загальна\_площа», «Ціна»

****

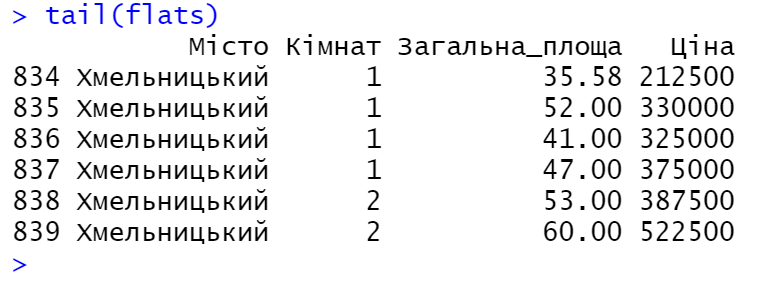
**Дослідження даних**

****

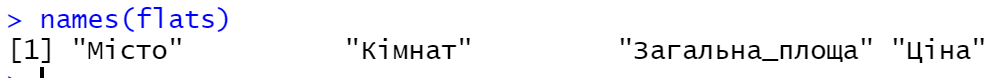
Вимірність датасету



Перша частина об’єкта

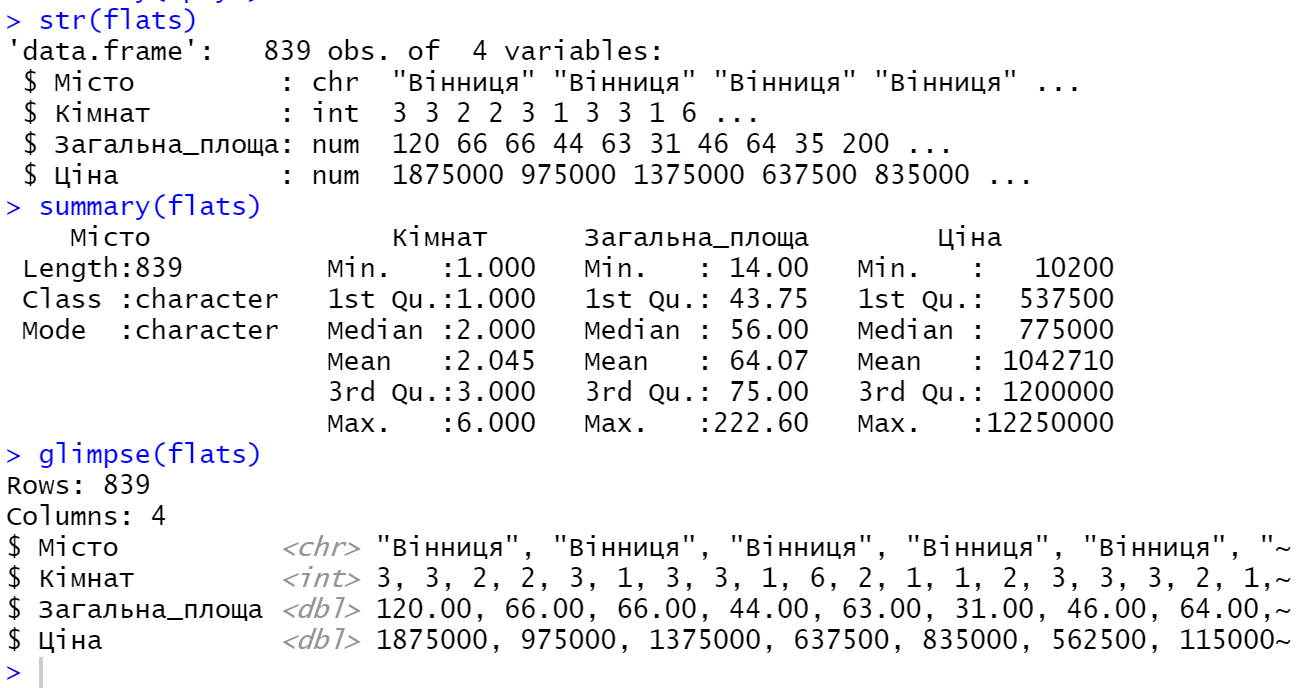


Остання частина об’єкта



Імена пов’язані з об’єктом

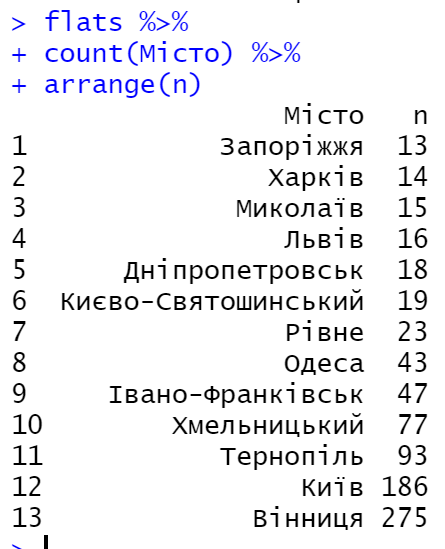
**Трансформація даних**



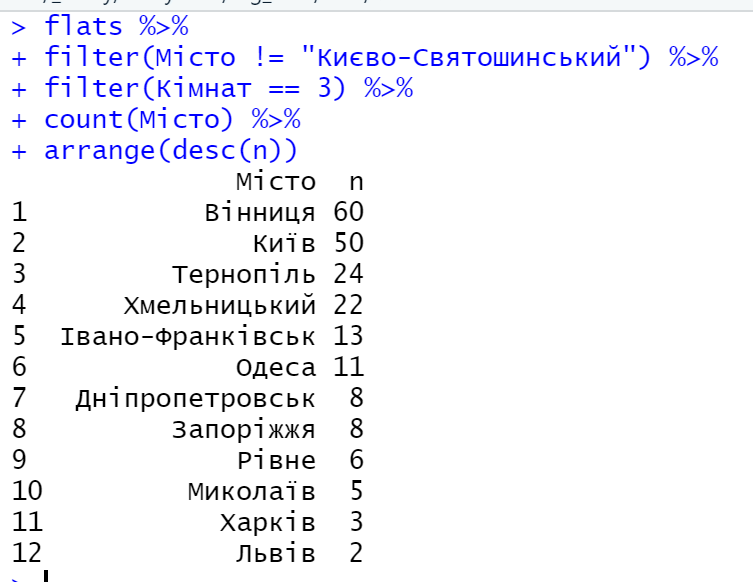
Кількість квартир:

****

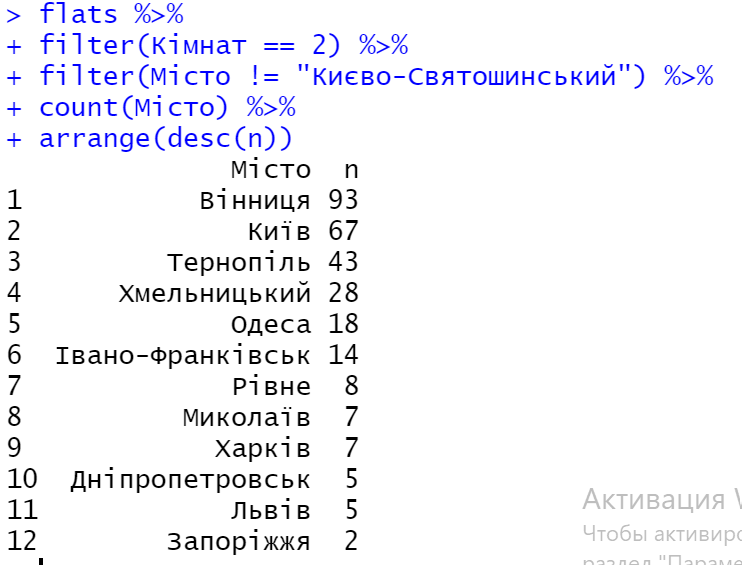
Сортування даних по кількості квартир в місті в зростаючому порядку

****

Вилучення даних по Києво-Святошинському району з відображення

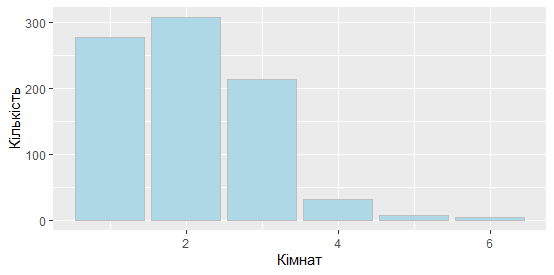
****

Квартири з кількістю кімнат 2:

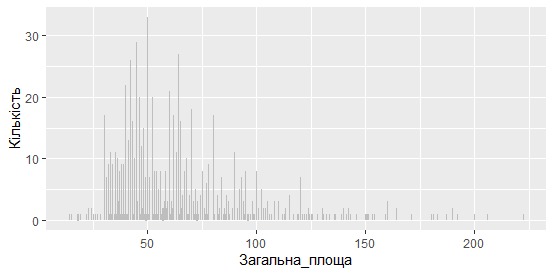


**Візуалізація даних**

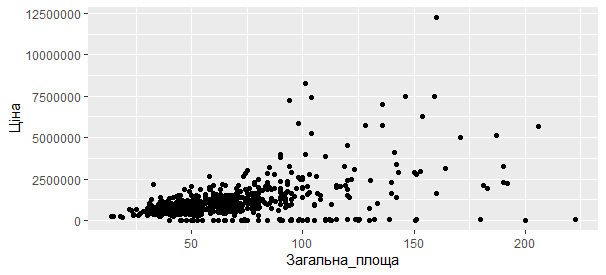
Стовипчикова діаграма для кількості кімнат:



Стовпчикова діаграма для змінної Загальна\_площа:



**Графік розсіювання**

Графік залежності ціни від загальної площі

**Висновки**

В ході виконанні 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.