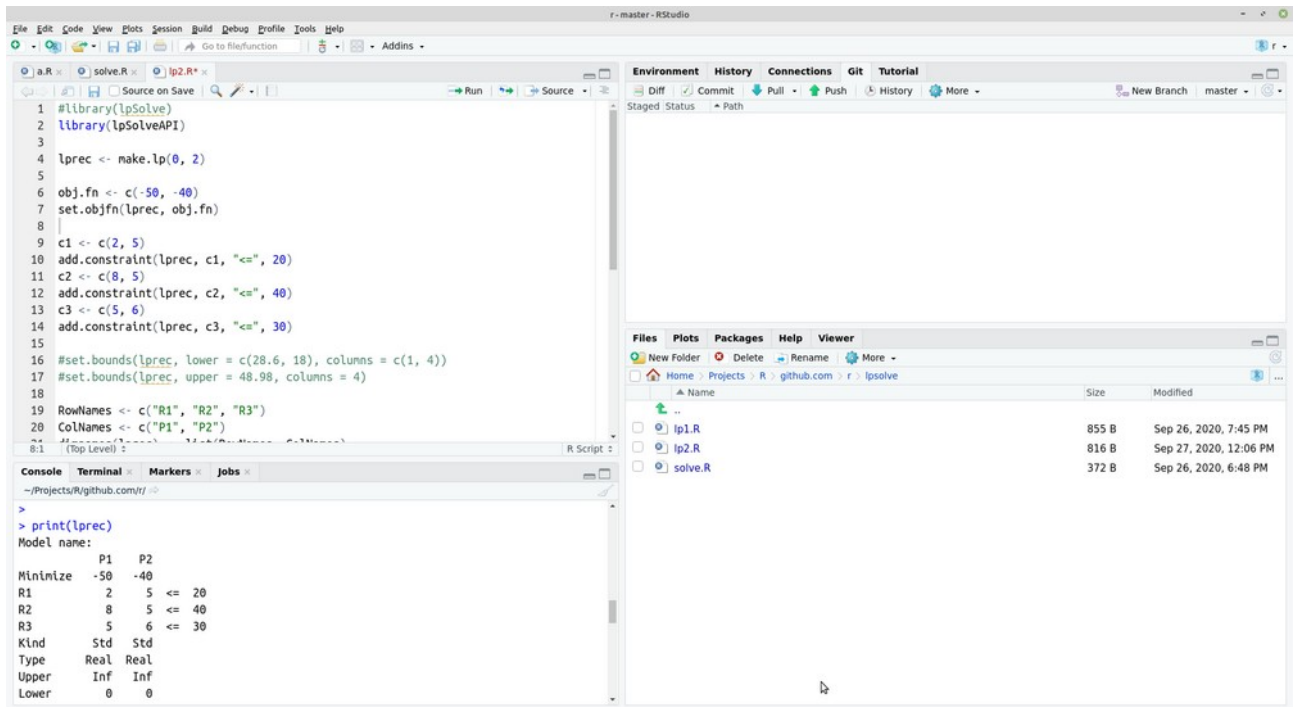


# Лабораторна робота 1

Тема: Встановлення R. Ознайомлення з основними компонентами R.

## Встановити RStudio. Вивчити інтерфейс.

RStudio вже був

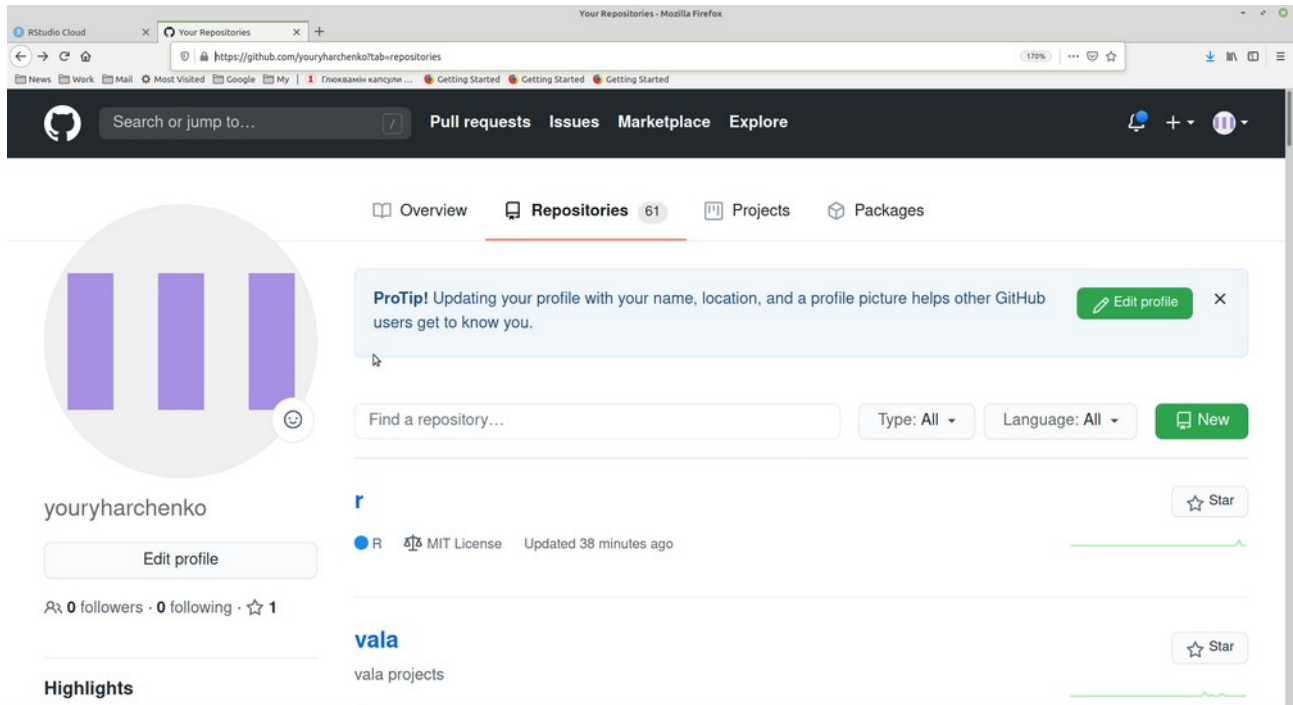


Встановив пакети скриптом

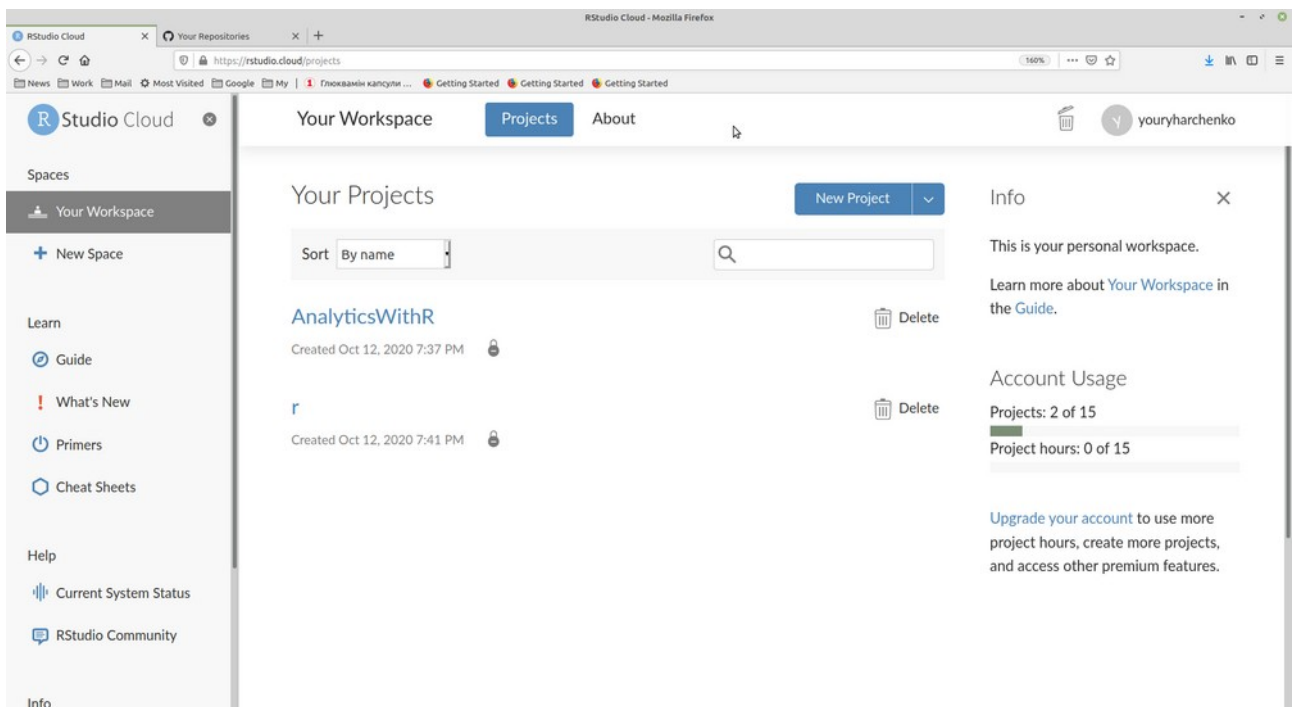
[https://raw.githubusercontent.com/bdemeshev/coursera\\_metrics/master/install\\_all.R](https://raw.githubusercontent.com/bdemeshev/coursera_metrics/master/install_all.R)

# Створити профілі GitHub.com та cloud.rstudio.com. Вивчити інтерфейс, наповнення на функцію відповідних ресурсів.

Мій профіль на GitHub



Створив профіль на rstudio.cloud



# Вивчити матеріал за темами Intro to basic, Vectors, Matrices

Вивчав самостійно раніше.

## Провести елементарні математичні операції, використовуючи код1, код2

### Код1

Протокол:

```
> #Step 2: Variable assiment
> my_var1 <- 42    # my_var1 тепер дорівнює 42

> my_var2 <- 35.25 # my_var2 тепер дорівнює 35.25

> my_var1 + 100          # надрукує 142
[1] 142

> my_var1 + my_var2 - 12    # надрукує 65.25
[1] 65.25

> my_var3 <- my_var1^2 + my_var2^2 # my_var3 тепер дорівнює 3006.562

> #Step 3: Logical opperartions
> my_var3 > 200             # надрукує TRUE
[1] TRUE

> my_var3 > 3009            # надрукує FALSE
[1] FALSE

> my_var1 == my_var2        # надрукує FALSE
[1] FALSE

> my_var1 != my_var2        # надрукує TRUE
[1] TRUE

> my_var3 >= 200            # надрукує TRUE
[1] TRUE

> my_var3 <= 200            # надрукує FALSE
[1] FALSE

> my_new_var <- my_var1 == my_var2 # my_new_var3 тепер дорівнює FALSE

> #Step 6, 7, 10, 11: Vectors
> 1 : 67                    # надрукує вектор з 67-и компонент від 1-го до 67-и
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[27] 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
[53] 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67
```



```
[66] TRUE TRUE
```

```
> x <- 23      # x тепер дорівнює 23
```

```
> my_vector1 > 23 # надрукує вектор, логічних значень, де TRUE, якщо компонента більша за 23, інакше FALSE
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
[14] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE
```

```
[27] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
[40] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
[53] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
[66] TRUE TRUE
```

```
> x == 23      # надрукує TRUE
```

```
[1] TRUE
```

```
> my_vector2 > 0 # надрукує вектор, логічних значень, де TRUE, якщо компонента більша за 0, інакше FALSE
```

```
[1] FALSE TRUE TRUE TRUE TRUE FALSE FALSE
```

```
> my_vector2[my_vector2 > 0] # надрукує вектор з компонент вектора my_vector2, які більші нуля
```

```
[1] 45.00 67.00 12.78 129.00
```

```
> my_vector2[my_vector2 < 0] # надрукує вектор з компонент вектора my_vector2, які менші нуля
```

```
[1] -32 -65
```

```
> my_vector2[my_vector2 == 0] # надрукує вектор з компонент вектора my_vector2, які дорівнюють нулю
```

```
[1] 0
```

```
> my_vector1[my_vector1 > 20 & my_vector1 < 30] # надрукує вектор з компонент вектора my_vector1, які більші 20 та менші 30
```

```
[1] 21 22 23 24 25 26 27 28 29
```

```
> my_numbers <- my_vector1[my_vector1 > 20 & my_vector1 < 30] # my_numbers тепер вектор з компонент вектора my_vector1, які більші 20 та менші 30
```

```
> positive_numbers <- my_vector2[my_vector2 > 0] # positive_numbers тепер вектор з компонент вектора my_vector2, які більші нуля
```

```
> v1 <- c(165, 178, 180, 181, 167, 178, 187, 167, 187) # v1 - вектор з вказаних компонент
```

```
> mean_v1 <- mean(v1) # mean_v1 - середнє значення компонент вектора v1
```

```
> v1[v1 > mean_v1] # надрукує вектор компонент з v1, які більше середнього
```

```
[1] 178 180 181 178 187 187
```

```
> greater_than_mean <- v1[v1 > mean_v1] # запишемо те саме в змінну greater_than_mean
```

```
> #Step 13: Lists and dataframes
```

```
> age <- c(16, 18, 22, 27) # список значень віку
```

```
> is_married <- c(F, F, T, T) # список сімейного стану
```

```
> name <- c("Olga", "Maria", "Nastya", "Polina") # список імен
```

```
> my_data <- data.frame(Name = name, Age = age, Status = is_married) # таблиця, де стовпчики  
вказані списки значень
```

```
>
```

## Код2

Протокол:

```
> ### vectors, data, matrices, subsetting
```

```
> #Vectors
```

```
> x = c(2,7,5)
```

```
> x
```

```
[1] 2 7 5
```

```
> y=seq(from=4,length=3,by=3) # генерує послідовність з 3-х чисел починаючи з 4-х з кроком  
3
```

```
> ?seq # help по функції seq
```

```
> y
```

```
[1] 4 7 10
```

```
> #Operations on vectors
```

```
> x+y # по-компонентне додавання
```

```
[1] 6 14 15
```

```
> x/y # по-компонентне ділення
```

```
[1] 0.5 1.0 0.5
```

```
> x^y # по-компонентна степінь
```

```
[1] 16 823543 9765625
```

```
> #Subsetting
```

```
> x[2] # друга
```

```
[1] 7
```

```
> x[2:3] # з другої по третю
```

```
[1] 7 5
```

```
> x[-2] # без другої
```

```
[1] 2 5
```

```
> x[-c(1,2)] # без першої та другої
[1] 5
```

```
> #Matrices
```

```
> z=matrix(seq(1,12),4,3) # створюємо матрицю - 4 рядка, 3 стовпчика, заповнюємо по
стовпцях числами від 1 до 12
```

```
> z
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

```
> #Subsetting
```

```
> z[3:4,2:3] # вибираємо елементи на перетині 3:4 рядки та 2:3 стовпчики
     [,1] [,2]
[1,]    7   11
[2,]    8   12
```

```
> z[,2:3] # вибираємо елементи на перетині всі рядки та 2:3 стовпчики
     [,1] [,2]
[1,]    5    9
[2,]    6   10
[3,]    7   11
[4,]    8   12
```

```
> z[,1] # вибираємо перший стовпчик, як вектор
[1] 1 2 3 4
```

```
> z[,1,drop=FALSE] # вибираємо перший стовпчик, як вектор-стовпчик
     [,1]
[1,]    1
[2,]    2
[3,]    3
[4,]    4
```

```
> #Dimension of matrix
```

```
> dim(z) # розмірності матриці
[1] 4 3
```

```
> #Variables in environment
```

```
> ls() # список всіх змінних
```

```
[1] "a"          "age"         "args"        "Auto"
[5] "c1"         "c2"          "c3"          "Col2"
[9] "ColNames"   "constr"      "constr.dir"  "cylinders"
[13] "greater_than_mean" "i"          "is_married"  "lprec"
[17] "mean_v1"    "my_data"     "my_new_var"  "my_numbers"
[21] "my_var1"    "my_var2"     "my_var3"     "my_vector1"
[25] "my_vector2" "n"           "name"        "obj.fn"
[29] "obj.fun"    "positive_numbers" "prod.sol"    "res.constr"
```

```
[33] "res.obj"      "res.s"      "res.vars"   "rhs"
[37] "RowNames"    "v1"        "x"         "y"
[41] "z"
```

```
> #Remove variable from environment
```

```
> rm(y) # видалити змінну y
```

```
> ls()
```

```
[1] "a"          "age"        "args"       "Auto"
[5] "c1"         "c2"        "c3"        "Col2"
[9] "ColNames"   "constr"     "constr.dir" "cylinders"
[13] "greater_than_mean" "i"         "is_married" "lprec"
[17] "mean_v1"     "my_data"    "my_new_var"  "my_numbers"
[21] "my_var1"     "my_var2"    "my_var3"     "my_vector1"
[25] "my_vector2"  "n"         "name"        "obj.fn"
[29] "obj.fun"     "positive_numbers" "prod.sol"    "res.constr"
[33] "res.obj"     "res.s"      "res.vars"   "rhs"
[37] "RowNames"    "v1"        "x"         "z"
```

```
> ### Generating random data, graphics
```

```
>
```

```
> #Generating uniformly distributed variable
```

```
> x=runif(50) # 50 випадкових рівномірно розподілених чисел (0, ... [TRUNCATED])
```

```
> #Generating normaly distributed variable
```

```
> y=rnorm(50) # 50 випадкових нормальнорозподілених чисел
```

```
> #Plot variables
```

```
> plot(x,y) # графік для x та y
```

```
> plot(x,y,xlab="Random Uniform",ylab="Random Normal",pch="*",col="blue") # графік для x та y з підписами синім кольором
```

```
> points(x[1:2], y[1:2], col = "red") # дві перші точки в червоний колір
```

```
> #Pair-plots
```

```
> par(mfrow=c(2,1)) # встановлюємо розміщення графіків (2 по вертикалі, 2 по горизонталі)
```

```
> plot(x,y) # виводимо графік для x та y
```

```
> hist(y) # виводимо гістограму для y
```

```
> par(mfrow=c(1,1)) # встановлюємо розміщення графіків (1 по вертикалі, 1 по горизонталі)
```

```
> ### Reading in data
```

```
> Auto <- read.csv("Auto.csv") # завантажує csv-файл в data.frame Auto
```

```
> #Names of the object
```

```
> names(Auto) # друкує список імен стовпчиків
```

```
[1] "mpg"      "cylinders" "displacement" "horsepower" "weight"
[6] "acceleration" "year"      "origin"      "name"
```



```

> #Dimension of data.frame
> dim(Auto) # друкує розмірність data.frame Auto
[1] 397  9

> #Class of the data.frame
> class(Auto) # друкує class data.frame Auto
[1] "data.frame"

> #Summary of object
> summary(Auto) # друкує статистики для числових стовпчиків data.frame Auto
   mpg     cylinders  displacement  horsepower      weight
Min.   : 9.00  Min.   :3.000  Min.   : 68.0  Length:397    Min.   :1613
1st Qu.:17.50  1st Qu.:4.000  1st Qu.:104.0  Class :character 1st Qu.:2223
Median :23.00  Median :4.000  Median :146.0  Mode  :character Median :2800
Mean   :23.52  Mean   :5.458  Mean   :193.5      Mean   :2970
3rd Qu.:29.00  3rd Qu.:8.000  3rd Qu.:262.0      3rd Qu.:3609
Max.   :46.60  Max.   :8.000  Max.   :455.0      Max.   :5140
 acceleration  year      origin      name
Min.   : 8.00  Min.   :70.00  Min.   :1.000  Length:397
1st Qu.:13.80  1st Qu.:73.00  1st Qu.:1.000  Class :character
Median :15.50  Median :76.00  Median :1.000  Mode  :character
Mean   :15.56  Mean   :75.99  Mean   :1.574
3rd Qu.:17.10  3rd Qu.:79.00  3rd Qu.:2.000
Max.   :24.80  Max.   :82.00  Max.   :3.000

> #Plot relation
> plot(Auto$cylinders,Auto$mpg) # виводить графік, де x - стовпчик cylinders, а y - стовпчик mpg

> attach(Auto) # приєднує data.frame Auto до списку пошуку імен змін
The following objects are masked _by_ .GlobalEnv:

   cylinders, name

The following objects are masked from Auto (pos = 3):

   acceleration, cylinders, displacement, horsepower, mpg, name, origin,
   weight, year

> #Active library
> search() # друкує список пошуку імен
[1] ".GlobalEnv"      "Auto"
[5] "tools:rstudio"   "package:stats"   "package:graphics" "package:grDevices"
[9] "package:utils"   "package:datasets" "package:methods"  "Autoloads"
[13] "package:base"

> #Plot bar plot
> plot(cylinders,mpg) # імена стовпчиків можна використовувати без кваліфікації об'єктом

> cylinders=as.factor(cylinders) # перетворює вектор cylinders на фактор

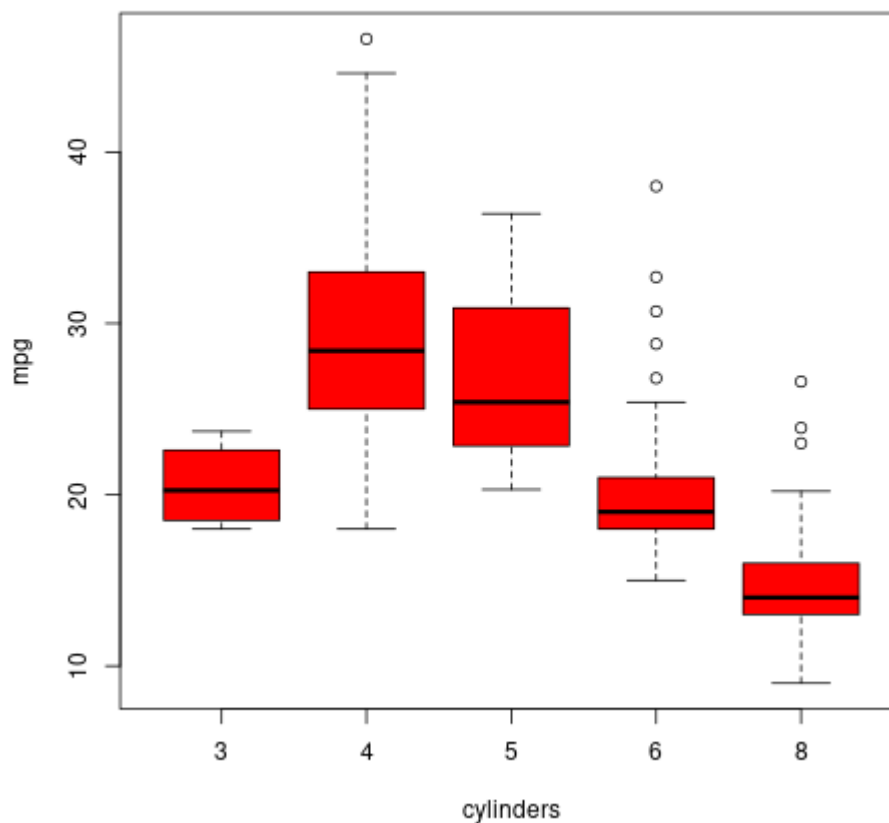
```

```

> plot(cylinders,mpg,xlab="cylinders",ylab="mpg",col="red") # виводить графік з cylinders як фактором
> #Plot to .png file
> png(file="mpg.png") # відкриває файл для графіку в форматі png
> plot(cylinders,mpg,xlab="cylinders",ylab="mpg",col="red") # виводить графік
> dev.off() # записує та закриває файл

```

RStudioGD  
2



```

> #Pair plot
> # pairs(Auto,col="brown") видає помилку Error in pairs.default(Auto, col = "brown") : non-numeric argument to 'pairs'
> pairs(mpg~cylinders+acceleration+weight,Auto) # виводить попарні матриці розсіювання
>

```

