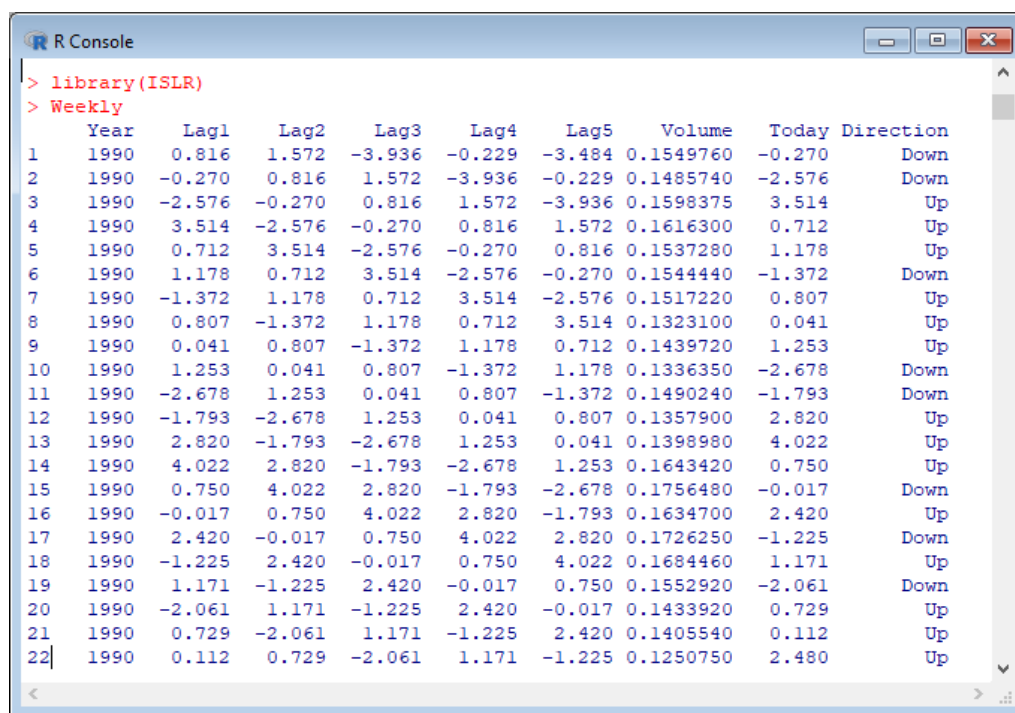


Звіт  
до індивідуального завдання №3  
з предмету Моделі статистичного навчання

Роботу виконала:  
**Мерцало Ірина Ігорівна,**  
студентка групи ПМІМ-11

**Завдання 1.** Розглянемо дані Weekly, які є частиною пакету ISLR.



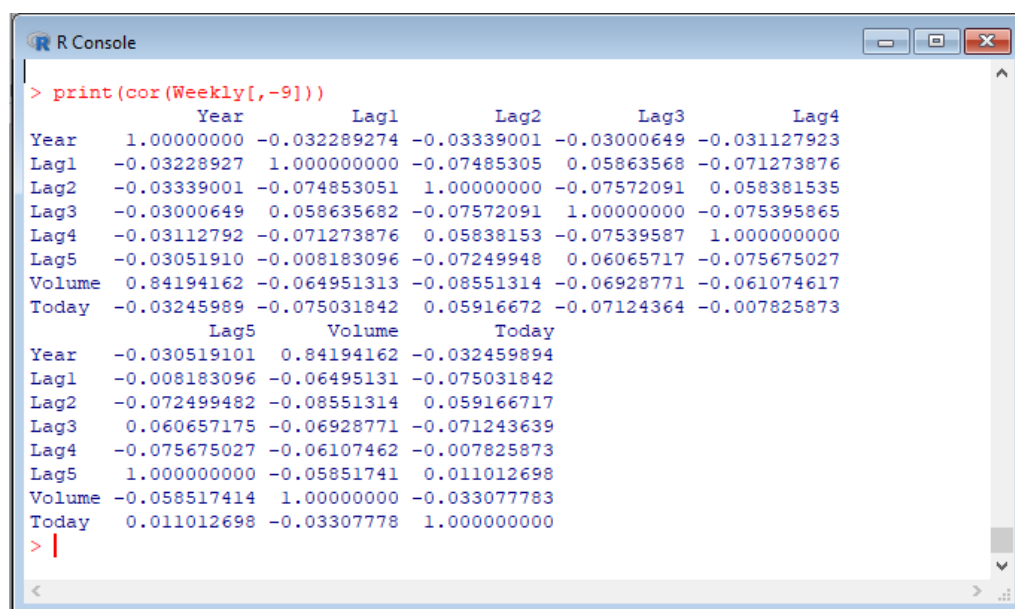
```

> library(ISLR)
> Weekly

```

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
1	1990	0.816	1.572	-3.936	-0.229	-3.484	0.1549760	-0.270	Down
2	1990	-0.270	0.816	1.572	-3.936	-0.229	0.1485740	-2.576	Down
3	1990	-2.576	-0.270	0.816	1.572	-3.936	0.1598375	3.514	Up
4	1990	3.514	-2.576	-0.270	0.816	1.572	0.1616300	0.712	Up
5	1990	0.712	3.514	-2.576	-0.270	0.816	0.1537280	1.178	Up
6	1990	1.178	0.712	3.514	-2.576	-0.270	0.1544440	-1.372	Down
7	1990	-1.372	1.178	0.712	3.514	-2.576	0.1517220	0.807	Up
8	1990	0.807	-1.372	1.178	0.712	3.514	0.1323100	0.041	Up
9	1990	0.041	0.807	-1.372	1.178	0.712	0.1439720	1.253	Up
10	1990	1.253	0.041	0.807	-1.372	1.178	0.1336350	-2.678	Down
11	1990	-2.678	1.253	0.041	0.807	-1.372	0.1490240	-1.793	Down
12	1990	-1.793	-2.678	1.253	0.041	0.807	0.1357900	2.820	Up
13	1990	2.820	-1.793	-2.678	1.253	0.041	0.1398980	4.022	Up
14	1990	4.022	2.820	-1.793	-2.678	1.253	0.1643420	0.750	Up
15	1990	0.750	4.022	2.820	-1.793	-2.678	0.1756480	-0.017	Down
16	1990	-0.017	0.750	4.022	2.820	-1.793	0.1634700	2.420	Up
17	1990	2.420	-0.017	0.750	4.022	2.820	0.1726250	-1.225	Down
18	1990	-1.225	2.420	-0.017	0.750	4.022	0.1684460	1.171	Up
19	1990	1.171	-1.225	2.420	-0.017	0.750	0.1552920	-2.061	Down
20	1990	-2.061	1.171	-1.225	2.420	-0.017	0.1433920	0.729	Up
21	1990	0.729	-2.061	1.171	-1.225	2.420	0.1405540	0.112	Up
22	1990	0.112	0.729	-2.061	1.171	-1.225	0.1250750	2.480	Up

1.1 На основі чисельних та графічних результатів щодо даних Weekly можна зробити висновок, що існують закономірності між Volume і Year. Їх побачила за допомогою функції `cor()`. Значення Year і Volume обидва близькі до 1.



```

> print(cor(Weekly[, -9]))

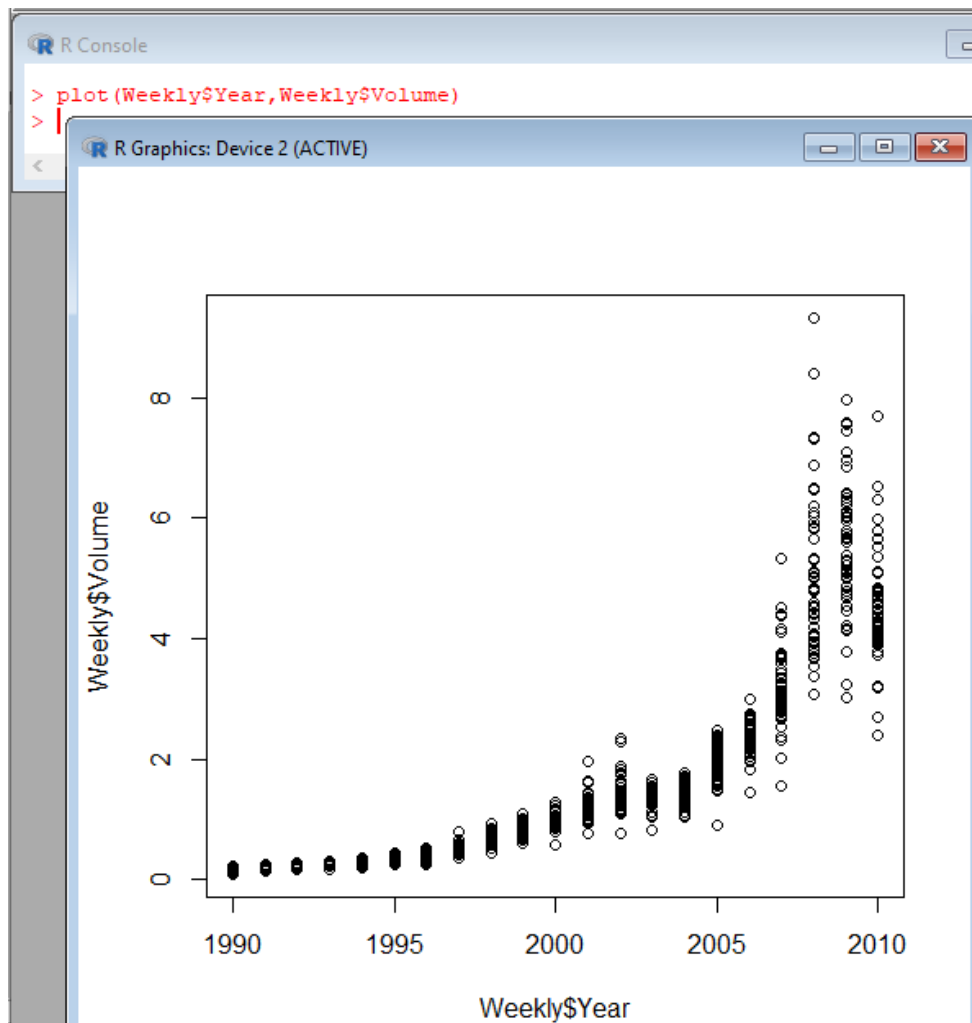
```

	Year	Lag1	Lag2	Lag3	Lag4
Year	1.00000000	-0.032289274	-0.03339001	-0.03000649	-0.031127923
Lag1	-0.03228927	1.00000000	-0.07485305	0.05863568	-0.071273876
Lag2	-0.03339001	-0.07485305	1.00000000	-0.07572091	0.058381535
Lag3	-0.03000649	0.05863568	-0.07572091	1.00000000	-0.075395865
Lag4	-0.03112792	-0.071273876	0.05838153	-0.07539587	1.00000000
Lag5	-0.03051910	-0.008183096	-0.07249948	0.06065717	-0.075675027
Volume	0.84194162	-0.064951313	-0.08551314	-0.06928771	-0.061074617
Today	-0.03245989	-0.075031842	0.05916672	-0.07124364	-0.007825873

	Lag5	Volume	Today
Year	-0.030519101	0.84194162	-0.032459894
Lag1	-0.008183096	-0.06495131	-0.075031842
Lag2	-0.072499482	-0.08551314	0.059166717
Lag3	0.060657175	-0.06928771	-0.071243639
Lag4	-0.075675027	-0.06107462	-0.007825873
Lag5	1.000000000	-0.05851741	0.011012698
Volume	-0.058517414	1.00000000	-0.033077783
Today	0.011012698	-0.03307778	1.000000000

Перевіряючи цю залежність графічно, можна побачити, що вона експонентна.



1.2 Використала повний набір даних, щоб побудувати логістичну регресію, де Direction – залежна змінна, а п'ять зміщених дохідностей та змінна Volume незалежні. Використала функцію `summary()` для виводу результатів.

```
R Console
> fit.glm=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Weekly, family=binomial)
> summary(fit.glm)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106  0.0019 **
Lag1        -0.04127    0.02641  -1.563  0.1181
Lag2         0.05844    0.02686   2.175  0.0296 *
Lag3        -0.01606    0.02666  -0.602  0.5469
Lag4        -0.02779    0.02646  -1.050  0.2937
Lag5        -0.01447    0.02638  -0.549  0.5833
Volume       -0.02274    0.03690  -0.616  0.5377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

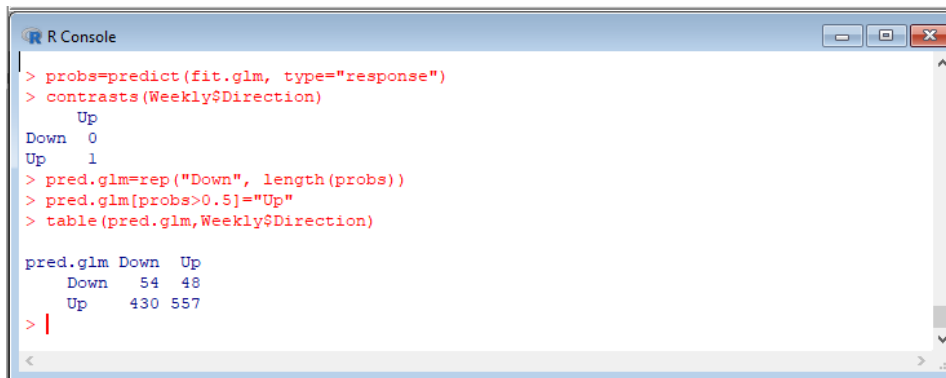
    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4

Number of Fisher Scoring iterations: 4

> |
```

Статистично значущим є Lag2, оскільки р дуже мале.

1.3 Визначивши які є прогнози, за допомогою функції `table()` побудувала матрицю помилок:

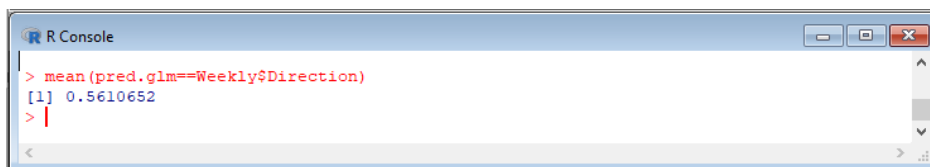


```
> probs=predict(fit.glm, type="response")
> contrasts(Weekly$Direction)
      Up
Down  0
Up    1
> pred.glm=rep("Down", length(probs))
> pred.glm[probs>0.5]="Up"
> table(pred.glm,Weekly$Direction)

pred.glm Down  Up
      Down   54  48
      Up    430 557
```

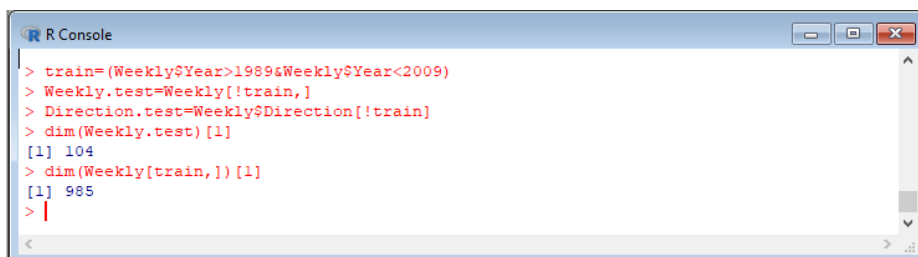
Матриця помилок показує, що 48 та 430 значень будуть мати одночасно і Up і Down, що є помилковим твердженням.

Обчислила загальну частку правильних прогнозів. Вона становить 56%.



```
> mean(pred.glm==Weekly$Direction)
[1] 0.5610652
```

1.4 Побудувала модель логістичної регресії з використанням навчальних даних з 1990 по 2008 рр., з єдиним предиктором Lag2:



```
> train=(Weekly$Year>1989&Weekly$Year<2009)
> Weekly.test=Weekly[!train,]
> Direction.test=Weekly$Direction[!train]
> dim(Weekly.test)[1]
[1] 104
> dim(Weekly[train,])[1]
[1] 985
```

```
R Console
> fit.glm2=glm(Direction~Lag2,data=Weekly,family=binomial,subset=train)
> summary(fit.glm2)

Call:
glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
     subset = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.536  -1.264   1.021   1.091   1.368

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.20326    0.06428   3.162  0.00157 **
Lag2         0.05810    0.02870   2.024  0.04298 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1350.5  on 983  degrees of freedom
AIC: 1354.5

Number of Fisher Scoring iterations: 4

> |
```

Побудувала матрицю помилок. Обчислила загальну частку правильних прогнозів на тестових даних (тобто даних за 2009 та 2010 роки), вона становить 62.5%, що є вищим порівняно з попереднім пунктом.

```
R Console
> probs2=predict(fit.glm2,Weekly.test, type="response")
> pred.glm2=rep("Down",length(probs2))
> pred.glm2[probs2>0.5]="Up"
> table(pred.glm2,Direction.test)
      Direction.test
pred.glm2 Down Up
Down      9  5
Up       34 56
> mean(pred.glm2==Direction.test)
[1] 0.625
> |
```

1.5 За допомогою функції `lda()`, повторила 1.4, використовуючи лінійний дискримінантний аналіз:

```
R Console
> fit.lda=lda(Direction~Lag2,data=Weekly,subset=train)
> fit.lda
Call:
lda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
      Down      Up 
0.4477157 0.5522843 

Group means:
      Lag2
Down -0.03568254
Up    0.26036581

Coefficients of linear discriminants:
      LD1
Lag2 0.4414162
> |
```

```
R Console
> pred.lda=predict(fit.lda,Weekly.test)
> table(pred.lda$class,Direction.test)
      Direction.test
      Down Up
Down    9  5
Up     34 56
> mean(pred.lda$class==Direction.test)
[1] 0.625
> |
```

Загальна частку правильних прогнозів на тестових даних співпадає з 1.4.

1.6 За допомогою функції `qda()`, повторила 1.4, використовуючи квадратичний дискримінантний аналіз:

```
R Console
> fit.qda=qda(Direction~Lag2,data=Weekly,subset=train)
> fit.qda
Call:
qda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
      Down      Up
0.4477157 0.5522843

Group means:
      Lag2
Down -0.03568254
Up    0.26036581
> pred.qda=predict(fit.qda,Weekly.test)
> table(pred.qda$class,Direction.test)
      Direction.test
      Down Up
Down    0  0
Up     43 61
> mean(pred.qda$class==Direction.test)
[1] 0.5865385
> |
```

Загальна частку правильних прогнозів на тестових даних менша ніж в попередніх пунктах і становить 58.7%.

1.7 За допомогою матриць `train.X` та `test.X` (до навчальної і тестової вибірки відповідно), а також функції `knn()`, повторила 1.4, використовуючи метод К-найближчих сусідів з  $K = 1$ .

```
R Console
> library(class)
> train.X=as.matrix(Weekly$Lag2[train])
> test.X=as.matrix(Weekly$Lag2[!train])
> Direction.test=Weekly$Direction[!train]
> set.seed(1)
> pred.knn=knn(train.X,test.X,Direction.train,k=1)
> table(pred.knn,Direction.test)
      Direction.test
pred.knn Down Up
Down    21 30
Up     22 31
> mean(pred.knn==Direction.test)
[1] 0.5
> |
```

Загальна частку правильних прогнозів на тестових даних ще менша ніж в попередніх пунктах і становить лише 50%.

1.8 Найкращі результати було отримано методом логістичної регресії та лінійного дискримінантного аналізу, адже там ми отримали загальну частку правильних прогнозів 62.5%.

1.9 Поекспериментувала з різними комбінаціями предикторів, використовуючи у тому числі можливі перетворення та взаємодії. Навела відомості про змінні, метод та пов'язану матрицю помилок, що забезпечує найкращі результати.

```
R Console
> fit.glm_19_1=glm(Direction~Lag1+Lag2,data=Weekly,family=binomial,subset=train)
> probs_19_1=predict(fit.glm_19_1,Weekly.test,type="response")
> pred.glm_19_1=rep("Down",length(probs2))
> pred.glm_19_1[probs_19_1>0.5]="Up"
> table(pred.glm_19_1,Direction.test)
      Direction.test
pred.glm_19_1 Down Up
      Down      7  8
      Up     36 53
> mean(pred.glm_19_1==Direction.test)
[1] 0.5769231
> |
```

```
R Console
> fit.lda_19_2=lda(Direction~Lag1-Lag3,data=Weekly,subset=train)
> pred.lda_19_2=predict(fit.lda_19_2,Weekly.test)
> table(pred.lda_19_2$class,Direction.test)
      Direction.test
      Down Up
Down      4  6
Up     39 55
> mean(pred.lda_19_2$class==Direction.test)
[1] 0.5673077
> |
```

```
R Console
> fit.qda_19_3=qda(Direction~Lag1*Lag4,data=Weekly,subset=train)
> pred.qda_19_3=predict(fit.qda_19_3,Weekly.test)
> table(pred.qda_19_3$class,Direction.test)
      Direction.test
      Down Up
Down      6 12
Up     37 49
> mean(pred.qda_19_3$class==Direction.test)
[1] 0.5288462
> |
```

Поекспериментувала зі значеннями для К у класифікаторі К-найближчих сусідів.

```
R Console
> pred.knn2=knn(train.X,test.X,Direction.train,k=2)
> table(pred.knn2, Direction.test)
      Direction.test
pred.knn2 Down Up
      Down  18 25
      Up   25 36
> mean(pred.knn2==Direction.test)
[1] 0.5192308
> |
```

```

R Console
> pred.knn4=knn(train.X,test.X,Direction.train,k=4)
> table(pred.knn4, Direction.test)
      Direction.test
pred.knn4 Down Up
      Down   20  21
      Up    23  40
> mean(pred.knn4==Direction.test)
[1] 0.5769231
> |

```

```

R Console
> pred.knn6=knn(train.X,test.X,Direction.train,k=6)
> table(pred.knn6, Direction.test)
      Direction.test
pred.knn6 Down Up
      Down   18  23
      Up    25  38
> mean(pred.knn6==Direction.test)
[1] 0.5384615
> |

```

**Завдання 2.** Розробити модель, щоб передбачити, чи вибране авто має велике або низьке споживання газу на базі даних Auto.

2.1 Створила двійкову змінну mpg01, яка містить 1, якщо mpg містить значення вище медіани, і 0, якщо mpg містить значення нижче його медіана. За допомогою функції data.frame () створила єдиний набір даних, що містить як mpg01, так і інші змінні з Auto.

```

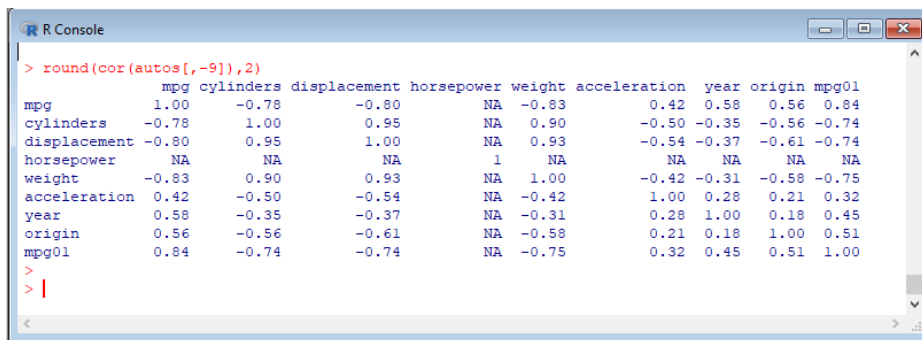
R Console
> mpg01=rep(0,length(autos$mpg))
> mpg01[autos$mpg>median(autos$mpg)]=1
> autos=data.frame(autos,mpg01)
> fix(autos)

```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	mpg01
1	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu	0
2	15	8	350	165	3693	11.5	70	1	buick skylark 320	0
3	18	8	318	150	3436	11	70	1	plymouth satellite	0
4	16	8	304	150	3433	12	70	1	amc rebel sst	0
5	17	8	302	140	3449	10.5	70	1	ford torino	0
6	15	8	429	198	4341	10	70	1	ford galaxie 500	0
7	14	8	454	220	4354	9	70	1	chevrolet impala	0
8	14	8	440	215	4312	8.5	70	1	plymouth fury iii	0
9	14	8	455	225	4425	10	70	1	pontiac catalina	0
10	15	8	390	190	3850	8.5	70	1	amc ambassador dpl	0
11	15	8	383	170	3563	10	70	1	dodge challenger se	0
12	14	8	340	160	3609	9	70	1	plymouth 'cuda 340	0
13	15	8	400	150	3761	9.5	70	1	chevrolet monte carlo	0
14	14	8	455	225	3086	10	70	1	buick estate wagon (sw)	0
15	24	4	113	95	2372	15	70	3	toyota corona mark ii	1
16	22	6	198	95	2833	15.5	70	1	plymouth duster	0
17	18	6	199	97	2774	15.5	70	1	amc hornet	0
18	21	6	200	85	2587	16	70	1	ford maverick	0
19	27	4	97	88	2130	14.5	70	3	datsun pl510	1

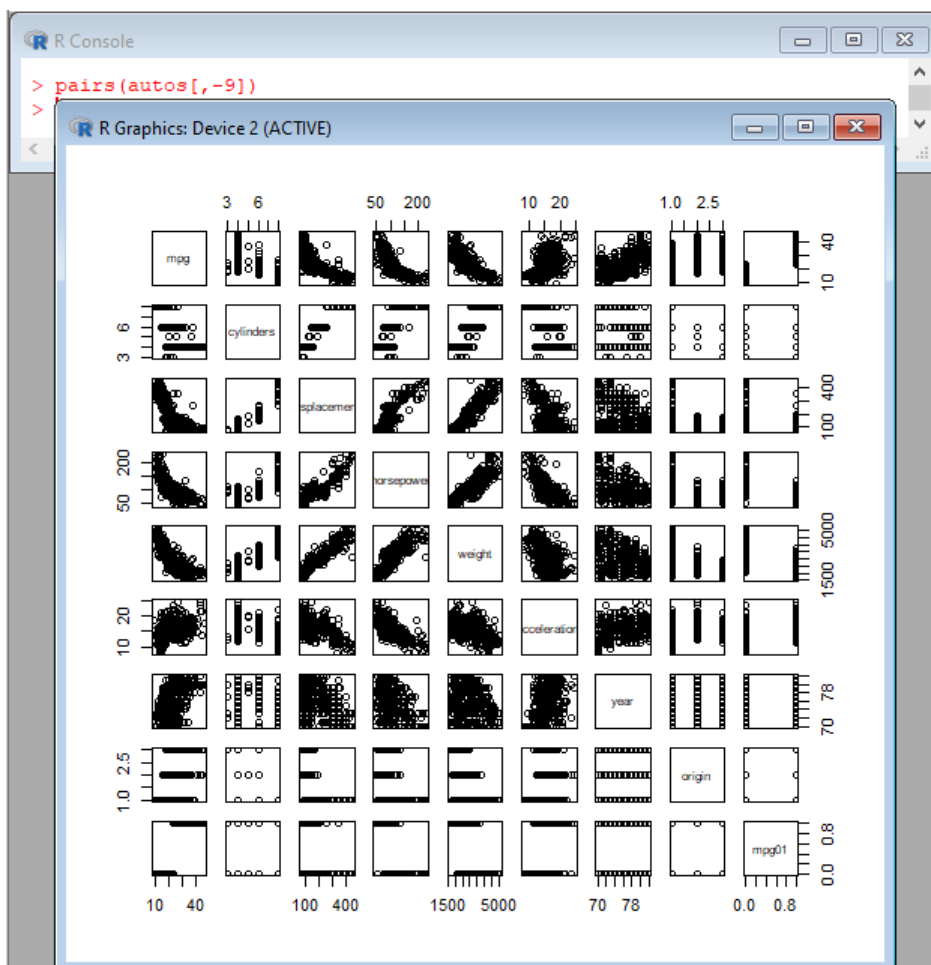
2.1 За допомогою функції cor() та pairs() визначила, які змінні будуть найбільш корисними для прогнозування mpg01:



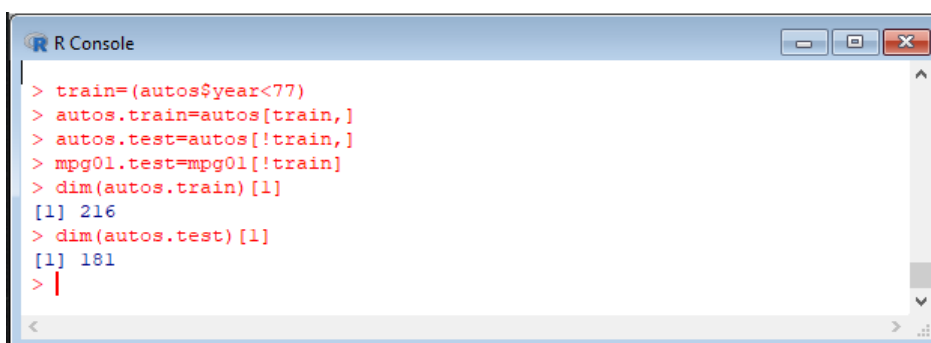


Між змінними cylinders, displacement, horsepower та weight можна побачити від'ємну кореляцію.

Дослідила дані графічно, щоб виявити залежність між mpg01 та цими характеристиками.



## 2.3 Розбила дані на навчальний та тестовий набори:



2.4 Застосувала лінійний дискримінантний аналіз на навчальних даних, щоб передбачити mpg01 з використанням змінних, які виявилися найбільше пов'язані з mpg01 у 2.2.

```
R Console
> fit.lda=lda(mpg01~cylinders+weight+displacement+horsepower,data=autos,subset=$
> fit.lda
Call:
lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = autos,
    subset = train)

Prior probabilities of groups:
      0      1 
0.7009346 0.2990654 

Group means:
  cylinders  weight displacement horsepower
0  6.693333 3614.700    274.0467   131.38667
1  4.031250 2182.172    102.2891    77.70312

Coefficients of linear discriminants:
              LD1
cylinders    -0.159332899
weight       -0.001154447
displacement -0.006423380
horsepower    0.015625322
> |
```

```
R Console
> pred.lda=predict(fit.lda,autos.test)
Warning messages:
1: In FUN(newX[, i], ...) : no non-missing arguments to min; returning Inf
2: In FUN(newX[, i], ...) : no non-missing arguments to min; returning Inf
3: In FUN(newX[, i], ...) : no non-missing arguments to min; returning Inf
> table(pred.lda$class,mpg01.test)
  mpg01.test
    0      1 
0  86    20 
1  12   124 
> mean(pred.lda$class!=mpg01.test)
[1] NA
> |
```

2.5. Застосувала логістичну регресію на навчальних даних, щоб передбачити mpg01 з використанням змінних, які виявилися найбільше пов'язані з mpg01 у 2.2.

```
R Console
> fit.glm=glm(mpg01~cylinders+weight+displacement+horsepower,data=autos,family=$
> fit.glm

Call: glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
  family = binomial, data = autos, subset = train)

Coefficients:
(Intercept)  cylinders      weight displacement  horsepower
 12.546165    0.574482   -0.004866   -0.034290    0.010504

Degrees of Freedom: 149 Total (i.e. Null); 145 Residual
(2 observations deleted due to missingness)
Null Deviance: 179.7
Residual Deviance: 57.91      AIC: 67.91
> |
```

```
R Console
> probs=predict(fit.glm,autos.test,type="response")
> pred.glm=rep(0,length(probs))
> pred.glm[probs>0.5]=1
> table(pred.glm,mpg01.test)
      mpg01.test
pred.glm  0  1
      0 98 62
      1  0 85
> mean(pred.glm!=mpg01.test)
[1] 0.2530612
> |
```

Загальна частка правильних прогнозів на тестових даних становить 25.3%.

### Завдання 3.

3.1 Написала функцію Power (), яка виведе результат піднесення 2 до 3-ї степені

```
R Console
> Power=function(){
+  2^3}
> Power()
[1] 8
> |
```

3.2. Написала нову функцію Power2 (), яка дозволяє передавати будь-які два числа, x і a, і виводить значення  $x^a$ :

```
R Console
> Power2=function(x,a){
+  x^a}
> |
```

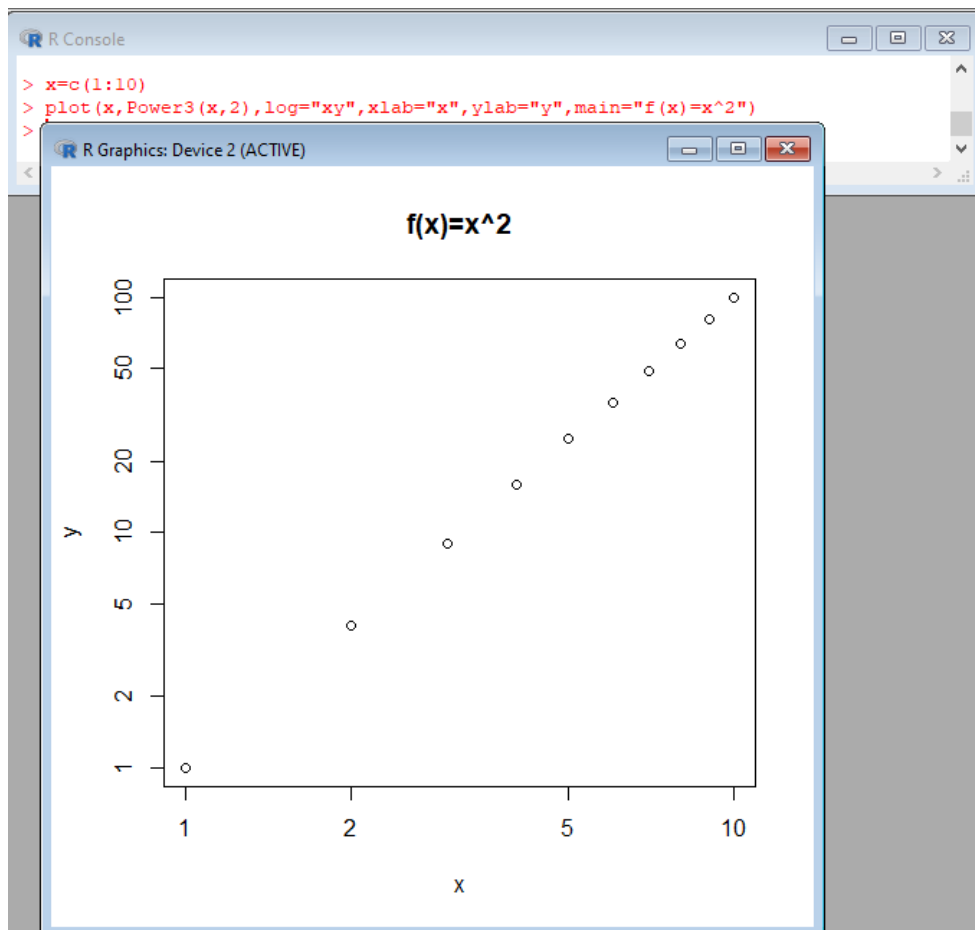
3.3. Продемонструвала роботу функції Power2():

```
R Console
> Power2(2,4)
[1] 16
> Power2(15,0)
[1] 1
> |
```

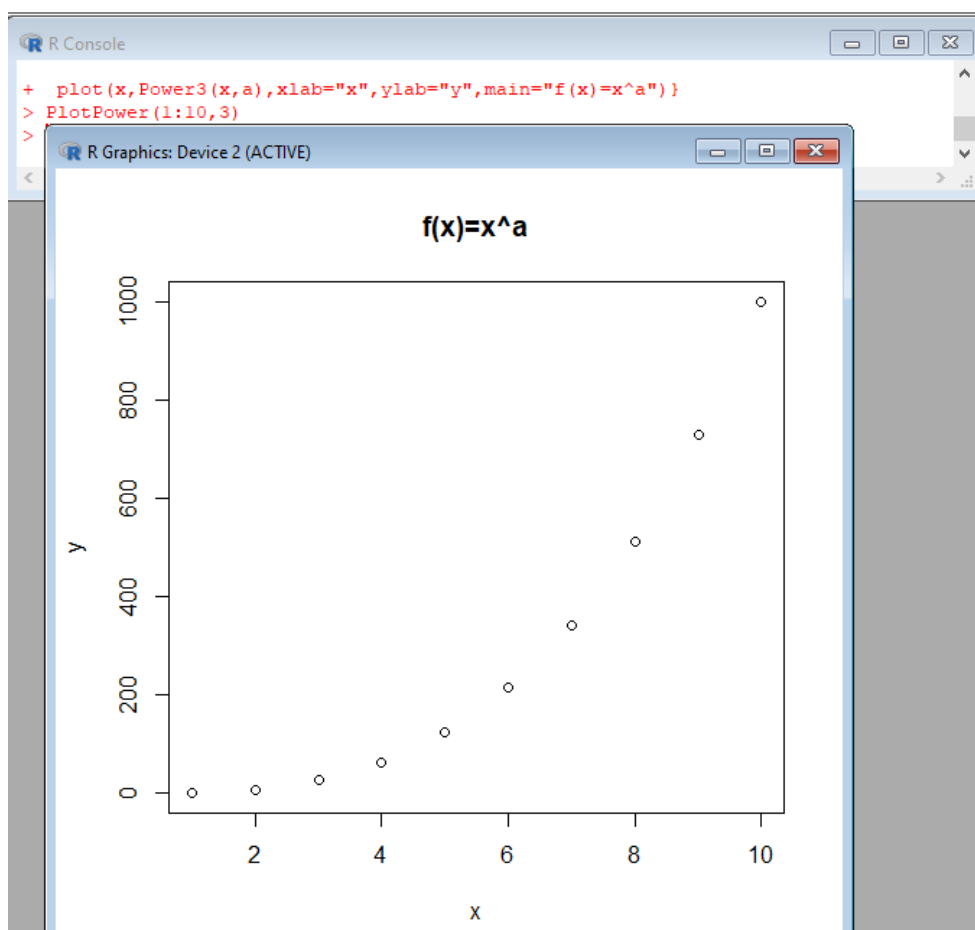
3.4. Написала нову функцію Power3 (), яка фактично повертає результат  $x^a$  як об'єкт R, а не просто друкує його на екран:

```
R Console
> Power3=function(x,a){
+  result=x^a
+  return(result)}
> result=Power3(2,3)
> result
[1] 8
> |
```

3.5. Використовуючи функцію Power3 (), побудувала графік  $f(x) = x^2$ :

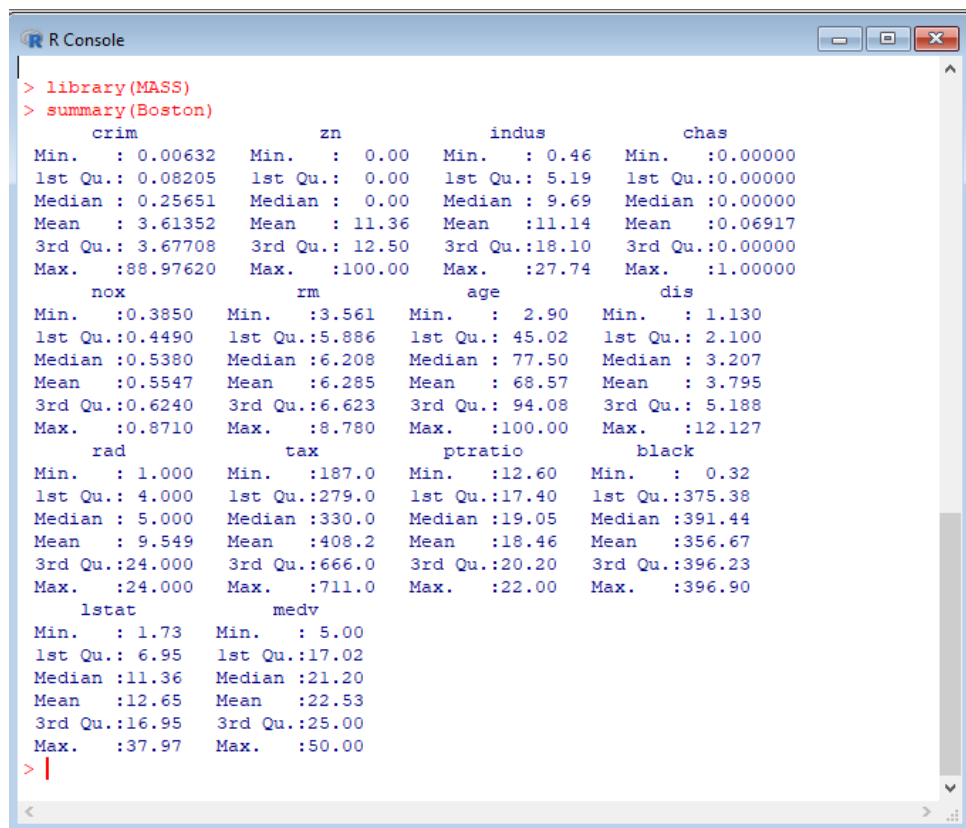


3.6. Написала функцію `PlotPower()`, яка дозволяє будувати графік функції  $x^a$  для фіксованого  $a$  та для діапазону значень  $x$ :



**Завдання 4.** Використовуючи дані Boston пристосувала модель класифікації для передбачення у вибраному районі рівня злочинності більшого чи меншого за медіану.

Отримала дані Boston:



```
R Console
> library(MASS)
> summary(Boston)

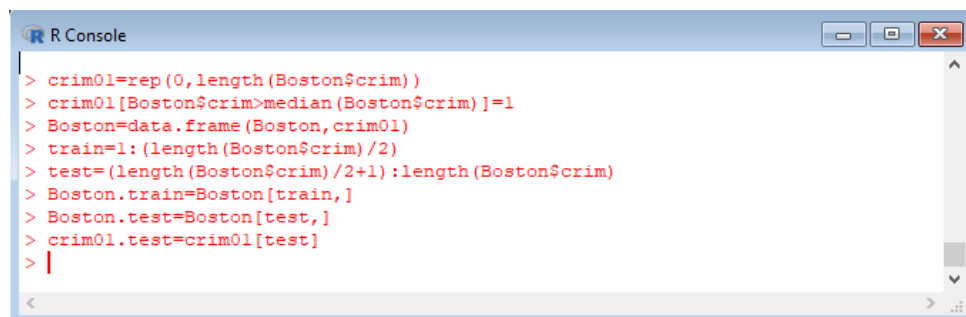
      crim      zn      indus      chas
Min.   :0.00632  Min.   : 0.00  Min.   : 0.46  Min.   :0.00000
1st Qu.: 0.08205  1st Qu.: 0.00  1st Qu.: 5.19  1st Qu.:0.00000
Median : 0.25651  Median : 0.00  Median : 9.69  Median :0.00000
Mean   : 3.61352  Mean   :11.36  Mean   :11.14  Mean   :0.06917
3rd Qu.: 3.67708  3rd Qu.:12.50  3rd Qu.:18.10  3rd Qu.:0.00000
Max.   :88.97620  Max.   :100.00  Max.   :27.74  Max.   :1.00000

      nox      rm      age      dis
Min.   :0.3850  Min.   :3.561  Min.   : 2.90  Min.   : 1.130
1st Qu.:0.4490  1st Qu.:5.886  1st Qu.:45.02  1st Qu.: 2.100
Median :0.5380  Median :6.208  Median :77.50  Median : 3.207
Mean   :0.5547  Mean   :6.285  Mean   :68.57  Mean   : 3.795
3rd Qu.:0.6240  3rd Qu.:6.623  3rd Qu.:94.08  3rd Qu.: 5.188
Max.   :0.8710  Max.   :8.780  Max.   :100.00  Max.   :12.127

      rad      tax      ptratio      black
Min.   : 1.000  Min.   :187.0  Min.   :12.60  Min.   : 0.32
1st Qu.: 4.000  1st Qu.:279.0  1st Qu.:17.40  1st Qu.:375.38
Median : 5.000  Median :330.0  Median :19.05  Median :391.44
Mean   : 9.549  Mean   :408.2  Mean   :18.46  Mean   :356.67
3rd Qu.:24.000  3rd Qu.:666.0  3rd Qu.:20.20  3rd Qu.:396.23
Max.   :24.000  Max.   :711.0  Max.   :22.00  Max.   :396.90

      lstat      medv
Min.   : 1.73  Min.   : 5.00
1st Qu.: 6.95  1st Qu.:17.02
Median :11.36  Median :21.20
Mean   :12.65  Mean   :22.53
3rd Qu.:16.95  3rd Qu.:25.00
Max.   :37.97  Max.   :50.00
> |
```

Створила двійкову змінну `crim01`, яка містить 1, якщо `crim` містить значення вище медіани, і 0, якщо `crim` містить значення нижче його медіана. За допомогою функції `data.frame()` створила єдиний набір даних, що містить як `crim01`, так і інші змінні з Boston.



```
R Console
> crim01=rep(0,length(Boston$crim))
> crim01[Boston$crim>median(Boston$crim)]=1
> Boston=data.frame(Boston,crim01)
> train=1:(length(Boston$crim)/2)
> test=(length(Boston$crim)/2+1):length(Boston$crim)
> Boston.train=Boston[train,]
> Boston.test=Boston[test,]
> crim01.test=crim01[test]
> |
```

Дослідила логістичну регресію:

```
R Console
> fit.glm=glm(crim~. -crim01-Boston$crim,data=Boston,family=binomial,subset=train)
Error in eval(family$initialize) : y values must be 0 <= y <= 1
> probs=predict(fit.glm,Boston.test,type="response")
> pred.glm[preds>0.5]=1
> table(pred.glm,crim01.test)
      crim01.test
pred.glm  0    1
      0  68  24
      1  22 139
> mean(pred.glm!=crim01.test)
[1] 0.1818182
> |
```

Побудувала матрицю помилок. Обчислила загальну частку правильних прогнозів на тестових даних, вона становить 18.2%, що є досить малим показником.

За допомогою функції `lda()`, повторила попередній пункт, використовуючи лінійний дискримінантний аналіз:

```
R Console
> fit.lda=lda(crim01 ~. -crim01-crim,data=Boston,subset=train)
> pred.lda=predict(fit.lda,Boston.test)
> table(pred.lda$class,crim01.test)
      crim01.test
      0    1
      0  80  24
      1  10 139
> mean(pred.lda$class!=crim01.test)
[1] 0.1343874
> |
```

Загальна частка правильних прогнозів на тестових даних становить 13.4%.

За допомогою функції `qda()`, повторила попередній пункт, використовуючи квадратичний дискримінантний аналіз:

```
R Console
> pred.qda=predict(fit.qda,Boston.test)
> table(pred.qda$class,crim01.test)
      crim01.test
      0    1
      0  84 159
      1   6   4
> mean(pred.qda$class!=crim01.test)
[1] 0.6521739
> |
```

Загальна частка правильних прогнозів на тестових даних становить 65.2%, що є досить хорошим показником.

За допомогою матриць `train.X` та `test.X` (до навчальної і тестової вибірки відповідно), а також функції `knn()`, повторила попередній пункт, використовуючи метод К-найближчих сусідів з  $K = 1, 2, 4, 6$ .

```
R Console
> train.X=cbind(Boston$indus,Boston$chas,Boston$nox,Boston$rm,Boston$age,
+ Boston$dis,Boston$tax,Boston$ptratio,Boston$black,Boston$lstat,Boston$medv
+ )[train,]
> test.X=cbind(Boston$indus,Boston$chas,Boston$nox,Boston$rm,Boston$age,
+ Boston$dis,Boston$tax,Boston$ptratio,Boston$black,Boston$lstat,Boston$medv
+ )[test,]
> crim01.train=crim01[train]
> set.seed(1)
> pred.knn=knn(train.X,test.X,crim01.train,k=1)
> table(pred.knn,crim01.test)
      crim01.test
pred.knn    0    1
      0  85 115
      1   5  48
> mean(pred.knn!=crim01.test)
[1] 0.4743083
> |
```

```
R Console
> pred.knn2=knn(train.X,test.X,crim01.train,k=2)
> table(pred.knn2,crim01.test)
      crim01.test
pred.knn2    0    1
      0  81  78
      1   9  85
> mean(pred.knn2!=crim01.test)
[1] 0.3438735
> |
```

```
R Console
> pred.knn4=knn(train.X,test.X,crim01.train,k=4)
> table(pred.knn4,crim01.test)
      crim01.test
pred.knn4    0    1
      0  83  53
      1   7 110
> mean(pred.knn4!=crim01.test)
[1] 0.2371542
> |
```

```
R Console
> pred.knn6=knn(train.X,test.X,crim01.train,k=6)
> table(pred.knn6,crim01.test)
      crim01.test
pred.knn6    0    1
      0  85  31
      1   5 132
> mean(pred.knn6!=crim01.test)
[1] 0.1422925
> |
```