МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

ЗВІТ

до індивідуального завдання №3

з дисципліни «Моделі статистичного навчання»

Виконав
студент групи ПМіМ-12:
Зелінський Олександр

Перевірив:
Проф. Заболоцький Т. М.

Львів – 2021

Хід виконання

1.Аналіз даних Weekly

Розглянемо дані Weekly.

Data Editor

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
1	1990	0.816	1.572	-3.936	-0.229	-3.484	0.154976	-0.27	Down
2	1990	-0.27	0.816	1.572	-3.936	-0.229	0.148574	-2.576	Down
3	1990	-2.576	-0.27	0.816	1.572	-3.936	0.1598375	3.514	Up
4	1990	3.514	-2.576	-0.27	0.816	1.572	0.16163	0.712	Up
5	1990	0.712	3.514	-2.576	-0.27	0.816	0.153728	1.178	Up
6	1990	1.178	0.712	3.514	-2.576	-0.27	0.154444	-1.372	Down
7	1990	-1.372	1.178	0.712	3.514	-2.576	0.151722	0.807	Up
8	1990	0.807	-1.372	1.178	0.712	3.514	0.13231	0.041	Up
9	1990	0.041	0.807	-1.372	1.178	0.712	0.143972	1.253	Up
10	1990	1.253	0.041	0.807	-1.372	1.178	0.133635	-2.678	Down
11	1990	-2.678	1.253	0.041	0.807	-1.372	0.149024	-1.793	Down
12	1990	-1.793	-2.678	1.253	0.041	0.807	0.13579	2.82	Up
13	1990	2.82	-1.793	-2.678	1.253	0.041	0.139898	4.022	Up
14	1990	4.022	2.82	-1.793	-2.678	1.253	0.164342	0.75	Up
15	1990	0.75	4.022	2.82	-1.793	-2.678	0.175648	-0.017	Down
16	1990	-0.017	0.75	4.022	2.82	-1.793	0.16347	2.42	Up
17	1990	2.42	-0.017	0.75	4.022	2.82	0.172625	-1.225	Down
18	1990	-1.225	2.42	-0.017	0.75	4.022	0.168446	1.171	Up
19	1990	1.171	-1.225	2.42	-0.017	0.75	0.155292	-2.061	Down

A data frame with 1089 observations on the following 9 variables.

`Year`

The year that the observation was recorded

`Lag1`

Percentage return for previous week

`Lag2`

Percentage return for 2 weeks previous

`Lag3`

Percentage return for 3 weeks previous

`Lag4`

Percentage return for 4 weeks previous

`Lag5`

Percentage return for 5 weeks previous

`Volume`

Volume of shares traded (average number of daily shares traded in billions)

`Today`

Percentage return for this week

`Direction`

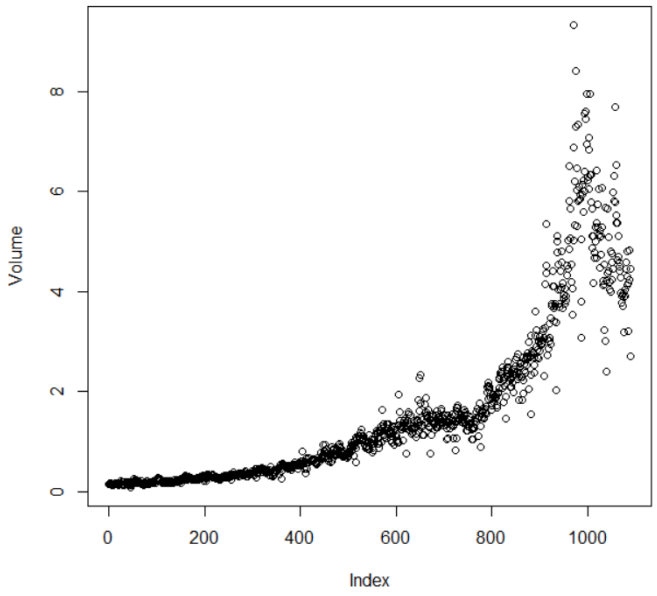
A factor with levels `Down` and `Up` indicating whether the market had a positive or negative return on a given week

```
> fix(Weekly)
> ?Weekly
starting httpd help server ... done
> summary(Weekly)
      Year           Lag1               Lag2               Lag3         
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
      Lag4               Lag5              Volume           Today         
 Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950  
 1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540  
 Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410  
 Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499  
 3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050  
 Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260  
 Direction 
 Down:484  
 Up  :605  
```

1.1 Для того, щоб оцінити закономірності розглянемо кореляції між змінними.

```
> print(cor(Weekly[, -9]))
              Year         Lag1        Lag2        Lag3         Lag4
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
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
```

Чітку кореляцію можна побачити лише між змінними Year та Volume, бо їх значення близькі до 1. З графіка чітко видно не лінійну, а скоріше квадратичну або експонентну залежність.



1.2

```
> 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
```

Чітко видно, що для заданої моделі найменше p відповідає змінній Lag2, тобто предиктор статистично значущий.

1.3

```
> probs = predict(fit.glm, type = "response")
> contrasts(Direction)
     Up
Down  0
Up    1
>
> pred.glm = rep("Down", length(probs))
> pred.glm[probs > 0.5] = "Up"
> table(pred.glm, Direction)
        Direction
pred.glm Down  Up
    Down   54  48
    Up    430 557
>
> paste("Частка правильних прогнозів: ", mean(pred.glm == Direction))
[1] "Частка правильних прогнозів:  0.561065197428834"
>
> paste("Частка правильних прогнозів коли ринок росте: ",
+   sum(pred.glm == Direction & Direction == "Up") / sum(Direction == "Up"))
[1] "Частка правильних прогнозів коли ринок росте:  0.920661157024793"
>
> paste("Частка правильних прогнозів коли ринок падає: ",
+   sum(pred.glm == Direction & Direction == "Down") / sum(Direction == "Down"))
[1] "Частка правильних прогнозів коли ринок падає:  0.111570247933884"
```

Після побудови матриці прогнозів та обчислень видно, що загальна частка правильних прогнозів становить 56%, що є не дуже хорошим результатом прогнозування. У ті тижні коли ринок йде вгору, модель правильно прогнозує у 92% випадків, проте у ті тижні коли ринок іде вниз, модель правильно прогнозує у 11.2% випадків.

1.4 Логістична регресія

Зважаючи на те що дані подано з 1990 року обмеження можна зробити лише на верхню дату тобто 2009 рік.

```
> train = (Year < 2009)
> Weekly.test = Weekly[!train, ]
> Direction.test = Direction[!train]
>
> cat("\n")

> paste("Кількість рядків в тестовій вибірці: ", dim(Weekly.test)[1])
[1] "Кількість рядків в тестовій вибірці:  104"
> paste("Кількість рядків в тренувальній вибірці: ", dim(Weekly[train, ])[1])
[1] "Кількість рядків в тренувальній вибірці:  985"
>
```

Можемо побачити що тепер тренувальна вибірка складається з 985 записів, а тестова зі 104.

```
> 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
```

```
> 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
>
> paste("Частка правильних прогнозів: ", mean(pred.glm2 == Direction.test))
[1] "Частка правильних прогнозів:  0.625"
>
> paste("Частка правильних прогнозів коли ринок росте: ",
+   sum(pred.glm2 == Direction.test & Direction.test == "Up") / sum(Direction.test == "Up"))
[1] "Частка правильних прогнозів коли ринок росте:  0.918032786885246"
>
> paste("Частка правильних прогнозів коли ринок падає: ",
+   sum(pred.glm2 == Direction.test & Direction.test == "Down") / sum(Direction.test == "Down"))
[1] "Частка правильних прогнозів коли ринок падає:  0.209302325581395"
>
```

Зважаючи на дані з матриці помилок, можна помітити, що модель позначає меншу частину тестових даних як спуск ринку, а більшу як підйом ринку. Загалом частка правильних прогнозів становить 62.5%. У ті тижні коли ринок йде вгору, модель правильно прогнозує у 91.8% випадків, проте у ті тижні коли ринок іде вниз модель правильно прогнозує у 20.9% випадків.

1.5 Лінійний дискримінантний аналіз

```
> library(MASS)
> fit.lda = lda(Direction ~ Lag2, data = Weekly, subset = train)
> print(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
```

```
> pred.lda = predict(fit.lda, Weekly.test)
> table(pred.lda$class, Direction.test)
      Direction.test
       Down Up
  Down    9  5
  Up     34 56
>
> paste("Частка правильних прогнозів: ", mean(pred.lda$class == Direction.test))
[1] "Частка правильних прогнозів:  0.625"
>
> paste("Частка правильних прогнозів коли ринок росте: ",
+   sum(pred.lda$class == Direction.test & Direction.test == "Up") / sum(Direction.test == "Up"))
[1] "Частка правильних прогнозів коли ринок росте:  0.918032786885246"
>
> paste("Частка правильних прогнозів коли ринок падає: ",
+   sum(pred.lda$class == Direction.test & Direction.test == "Down") / sum(Direction.test == "Down"))
[1] "Частка правильних прогнозів коли ринок падає:  0.209302325581395"
>
```

У цьому випадку видно , що дані є добре розподілені, тобто 55% вибірки позначено як зростання, а 45% як спадання ринку.

Зважаючи на дані з матриці помилок, можна помітити, що модель позначає більшу частину тестових даних як підйом ринку. Загалом частка правильних прогнозів становить 62.5%. У ті тижні коли ринок йде вгору, модель правильно прогнозує у 91.8% випадків, проте у ті тижні коли ринок іде вниз модель правильно прогнозує у 20.9% випадків.

1.6 Квадратичний дискримінантний аналіз

```
> fit.qda = qda(Direction ~ Lag2, data = Weekly, subset = train)
> print(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$class, Direction.test)
      Direction.test
       Down Up
  Down    0  0
  Up     43 61
>
> paste("Частка правильних прогнозів: ", mean(pred.qda$class == Direction.test))
[1] "Частка правильних прогнозів:  0.586538461538462"
>
> paste("Частка правильних прогнозів коли ринок росте: ",
+   sum(pred.qda$class == Direction.test & Direction.test == "Up") / sum(Direction.test == "Up"))
[1] "Частка правильних прогнозів коли ринок росте:  1"
>
> paste("Частка правильних прогнозів коли ринок падає: ",
+   sum(pred.qda$class == Direction.test & Direction.test == "Down") / sum(Direction.test == "Down"))
[1] "Частка правильних прогнозів коли ринок падає:  0"
```

Зважаючи на дані з матриці помилок видно, що модель позначає всі тестові дані як зростання ринку. Загалом частка правильних прогнозів становить 58.7%. У ті тижні коли ринок йде вгору, модель правильно прогнозує у 100% випадків, проте у ті тижні коли ринок іде вниз модель не прогнозує правильно взагалі.

1.7 Класифікатор K-найближчих сусідів з K=1

Для функції прогнозування формуються дві матриці з предиктора Lag2, які пов’язані окремо з навчальними та окремо тестовими даними.

```
> library(class)
> train.X = as.matrix(Lag2[train])
> test.X = as.matrix(Lag2[!train])
>
> Direction.train = 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
>
> paste("Частка правильних прогнозів: ", mean(pred.knn == Direction.test))
[1] "Частка правильних прогнозів:  0.5"
>
> paste("Частка правильних прогнозів коли ринок росте: ",
+   sum(pred.knn == Direction.test & Direction.test == "Up") / sum(Direction.test == "Up"))
[1] "Частка правильних прогнозів коли ринок росте:  0.508196721311475"
>
> paste("Частка правильних прогнозів коли ринок падає: ",
+   sum(pred.knn == Direction.test & Direction.test == "Down") / sum(Direction.test == "Down"))
[1] "Частка правильних прогнозів коли ринок падає:  0.488372093023256"
```

Зважаючи на дані з матриці помилок, можна помітити, що частка правильних загалом частка правильних прогнозів становить 50%. У ті тижні коли ринок йде вгору, модель правильно прогнозує у 50.8% випадків, проте у ті тижні коли ринок іде вниз модель правильно прогнозує у 48.8% випадків.

1.8

```
> error_rate.knn = 1 - mean(pred.knn == Direction.test)
> error_rate.qda = 1 - mean(pred.qda$class == Direction.test)
> error_rate.lda = 1 - mean(pred.lda$class == Direction.test)
> error_rate.glm2 = 1 - mean(pred.glm2 == Direction.test)
>
> paste("Коефіцієнт помилок для класифікатора K-найближчих сусідів з K=1", ": ", error_rate.knn)
[1] "Коефіцієнт помилок для класифікатора K-найближчих сусідів з K=1 :  0.5"
> paste("Коефіцієнт помилок для квадратичного дискримінантного аналізу", ": ", error_rate.qda)
[1] "Коефіцієнт помилок для квадратичного дискримінантного аналізу :  0.413461538461538"
> paste("Коефіцієнт помилок для лінійного дискримінантного аналізу", ": ", error_rate.lda)
[1] "Коефіцієнт помилок для лінійного дискримінантного аналізу :  0.375"
> paste( "Коефіцієнт помилок для логістичної регресії", ": ", error_rate.glm2)
[1] "Коефіцієнт помилок для логістичної регресії :  0.375"
```

З наведених вище коефіцієнтів тестових помилок можна зробити висновок, що найбільш відповідними для нашої вибірки виявились моделі лінійного дискримінантного аналізу та логістичної регресії.

1.9

```
> fit.glm3 =  glm(Direction ~ Lag1:Lag2, data = Weekly, family = binomial, subset = train)
> probs3 = predict(fit.glm3, Weekly.test, type = "response")
> pred.glm3 =  rep("Down", length(probs2))
> pred.glm3[probs3 > 0.5] = "Up"
> table(pred.glm3, Direction.test)
         Direction.test
pred.glm3 Down Up
     Down    1  1
     Up     42 60
> paste("Частка правильних прогнозів: ", mean(pred.glm3 == Direction.test))
[1] "Частка правильних прогнозів:  0.586538461538462"
```

```
> fit.lda2 = lda(Direction ~ Lag1*Lag2, data = Weekly, subset = train)
> pred.lda2 = predict(fit.lda2, Weekly.test)
> table(pred.lda2$class, Direction.test)
      Direction.test
       Down Up
  Down    7  8
  Up     36 53
> paste("Частка правильних прогнозів: ", mean(pred.lda2$class == Direction.test))
[1] "Частка правильних прогнозів:  0.576923076923077"
```

```
> fit.qda2 = qda(Direction ~ Lag1 + Lag2 + sqrt(abs(Lag1)), data = Weekly, subset = train)
> pred.qda2 = predict(fit.qda2, Weekly.test)
> table(pred.qda2$class, Direction.test)
      Direction.test
       Down Up
  Down   13 13
  Up     30 48
> paste("Частка правильних прогнозів: ", mean(pred.qda2$class == Direction.test))
[1] "Частка правильних прогнозів:  0.586538461538462"
```

Зважаючи на частку правильних прогнозів можна легко помітити, що ці моделі мають гіршу точність ніж початкова модель лінійного дискримінантного аналізу.

```
> pred.knn2 = knn(train.X, test.X, Direction.train, k = 1)
> table(pred.knn2, Direction.test)
         Direction.test
pred.knn2 Down Up
     Down   21 29
     Up     22 32
> paste("Частка правильних прогнозів: ", mean(pred.knn2 == Direction.test))
[1] "Частка правильних прогнозів:  0.509615384615385"
>
> pred.knn3 = knn(train.X, test.X, Direction.train, k = 2)
> table(pred.knn3, Direction.test)
         Direction.test
pred.knn3 Down Up
     Down   19 30
     Up     24 31
> paste("Частка правильних прогнозів: ", mean(pred.knn3 == Direction.test))
[1] "Частка правильних прогнозів:  0.480769230769231"
>
> pred.knn4 = knn(train.X, test.X, Direction.train, k = 4)
> table(pred.knn4, Direction.test)
         Direction.test
pred.knn4 Down Up
     Down   19 21
     Up     24 40
> paste("Частка правильних прогнозів: ", mean(pred.knn4 == Direction.test))
[1] "Частка правильних прогнозів:  0.567307692307692"
>
> pred.knn5 = knn(train.X, test.X, Direction.train, k = 8)
> table(pred.knn5, Direction.test)
         Direction.test
pred.knn5 Down Up
     Down   17 20
     Up     26 41
> paste("Частка правильних прогнозів: ", mean(pred.knn5 == Direction.test))
[1] "Частка правильних прогнозів:  0.557692307692308"
```

В результаті виведено матриці помилок при значеннях K= 1, 2, 4, 8. З цих значень добре видно, що для значення K=4 значення частки правильних прогнозів є найкращим, а саме 56.7%.

2. Модель для передбачення, чи вибране авто має велике або низьке споживання газу на базі даних Auto.

2.1 Створено змінну mpg01 та додано її до набору даних Autos.

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

Data Editor

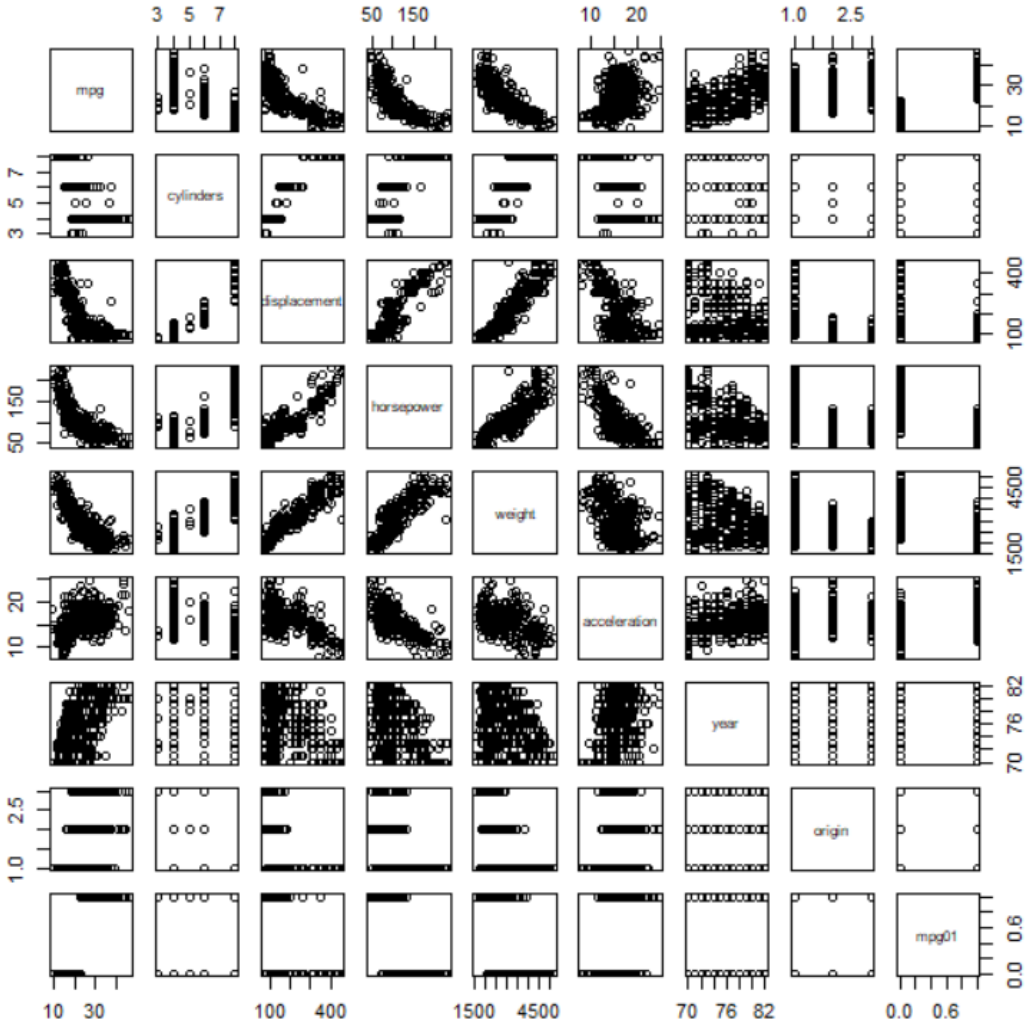
	row.names	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	mpg01
1	1	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu	0
2	2	15	8	350	165	3693	11.5	70	1	buick skylark 320	0
3	3	18	8	318	150	3436	11	70	1	plymouth satellite	0
4	4	16	8	304	150	3433	12	70	1	amc rebel sst	0
5	5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

2.2

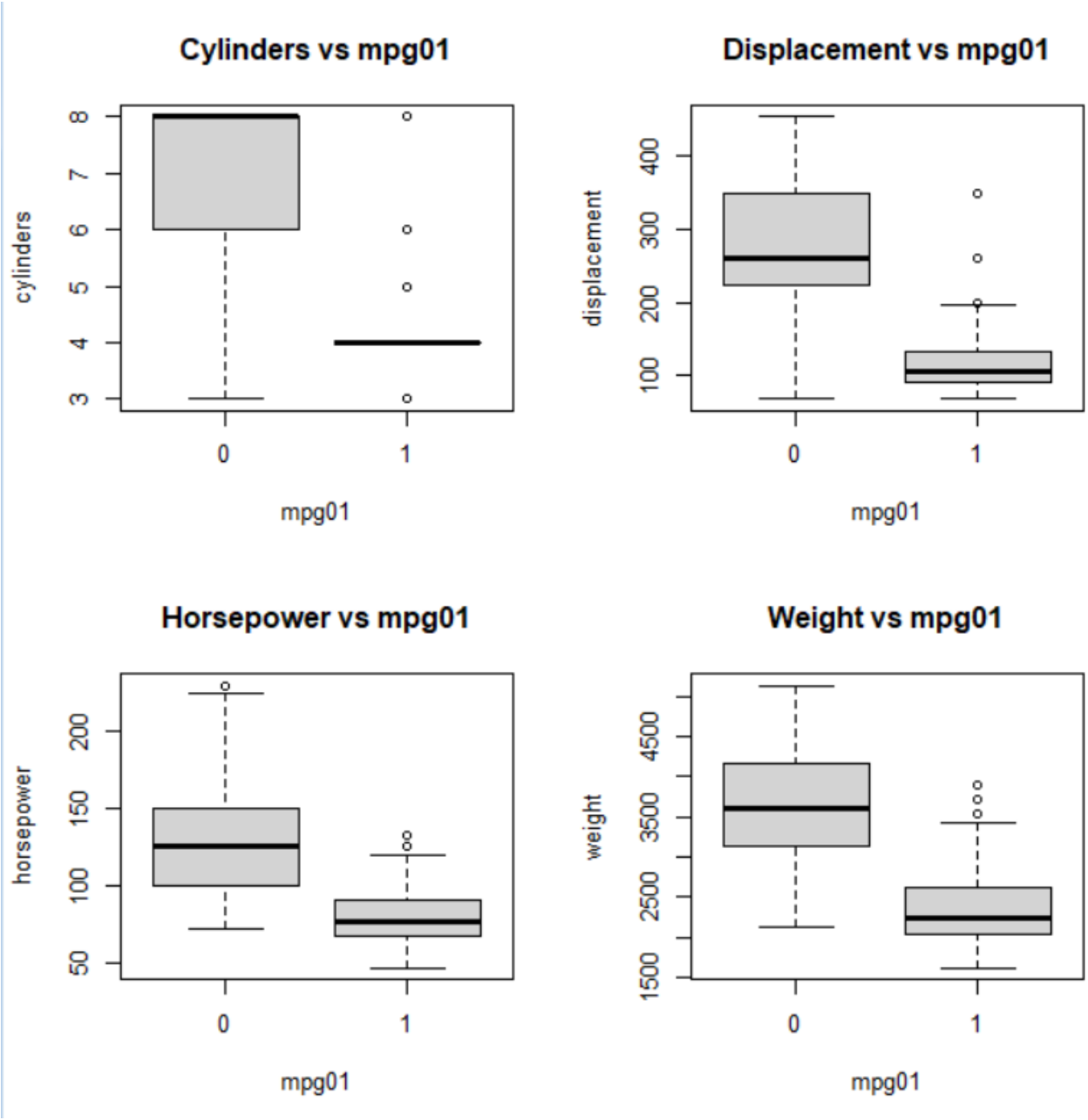
Використано функцію cor(), щоб побачити настільки сильна кореляція між змінними.

```
> round(cor(autos[, -9]), 2)
              mpg cylinders displacement horsepower weight acceleration  year
mpg           1.00     -0.78        -0.81      -0.78  -0.83         0.42  0.58
cylinders    -0.78      1.00         0.95       0.84   0.90        -0.50 -0.35
displacement -0.81      0.95         1.00       0.90   0.93        -0.54 -0.37
horsepower   -0.78      0.84         0.90       1.00   0.86        -0.69 -0.42
weight       -0.83      0.90         0.93       0.86   1.00        -0.42 -0.31
acceleration  0.42     -0.50        -0.54      -0.69  -0.42         1.00  0.29
year          0.58     -0.35        -0.37      -0.42  -0.31         0.29  1.00
origin        0.57     -0.57        -0.61      -0.46  -0.59         0.21  0.18
mpg01         0.84     -0.76        -0.75      -0.67  -0.76         0.35  0.43
              origin mpg01
mpg             0.57  0.84
cylinders      -0.57 -0.76
displacement   -0.61 -0.75
horsepower     -0.46 -0.67
weight         -0.59 -0.76
acceleration    0.21  0.35
year            0.18  0.43
origin          1.00  0.51
mpg01           0.51  1.00
```

На додачу за допомогою pairs() виведено графічну залежність між всіма змінними.



Отже, з наведених вище даних видно, що існує залежність між mpg01 та cylinders, displacement, horsepower та weight, проте кореляція між ними є від’ємною. Саме тому за допомогою функції boxplot() продовжимо дослідження вищезгаданих змінних.



Всі наведені вище boxplot-и вказують, що змінні cylinders, displacement, horsepower, weight набувають більших значень при значенні змінної mpg01 = 0 ніж при значенні 1. Проте варто зауважити, що для всіх кожної змінної з цього списку існують значення при mpg01 = 1, які є більші за середнє при mpg01 = 0.

2.3 Вибірку розбито на тестову і навчальну по критерію парний рік чи ні

```
> train <- (year %% 2 == 0)
> autos.train = autos[train,
> autos.test = autos[!train,
> mpg01.test = mpg01[!train]
>
> print(dim(autos.train)[1])
[1] 210
> print(dim(autos.test)[1])
[1] 182
> fix(autos)
```

2.4 Лінійний дискримінантний аналіз

В ролі предикторів було взято змінні cylinders, displacement, horsepower та weight, які є найбільш залежними від mpg01.

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

Prior probabilities of groups:
        0         1
0.4571429 0.5428571

Group means:
  cylinders   weight displacement horsepower
0  6.812500 3604.823     271.7396  133.14583
1  4.070175 2314.763     111.6623   77.92105

Coefficients of linear discriminants:
                     LD1
cylinders    -0.6741402638
weight       -0.0011465750
displacement  0.0004481325
horsepower    0.0059035377
```

```
> pred.lda = predict(fit.lda, autos.test)
> table(pred.lda$class, mpg01.test)
   mpg01.test
     0  1
  0 86  9
  1 14 73
>
> paste("Коефіцієнт помилок: ", mean(pred.lda$class != mpg01.test))
[1] "Коефіцієнт помилок:  0.126373626373626"
```

Отже в результаті бачимо, що тестова помилка отриманої моделі є 12.6%

2.5 Квадратичний дискримінантний аналіз

В ролі предикторів було взято змінні cylinders, displacement, horsepower та weight, які є найбільш залежними від mpg01.

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

Prior probabilities of groups:
        0         1
0.4571429 0.5428571

Group means:
  cylinders   weight displacement horsepower
0  6.812500 3604.823     271.7396  133.14583
1  4.070175 2314.763     111.6623   77.92105
```

```
> pred.qda = predict(fit.qda, autos.test)
> table(pred.qda$class, mpg01.test)
   mpg01.test
     0  1
  0 89 13
  1 11 69
> paste("Коефіцієнт помилок: ", mean(pred.qda$class != mpg01.test))
[1] "Коефіцієнт помилок:  0.131868131868132"
```

Отже в результаті бачимо, що тестова помилка отриманої моделі є 13.2%

2.6 Логістична регресія.

```
> fit.glm = glm(mpg01 ~ cylinders + weight + displacement + horsepower,
+   data = autos, family = binomial, subset = train)
> summary(fit.glm)

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

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.48027  -0.03413   0.10583   0.29634   2.57584

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  17.658730   3.409012   5.180 2.22e-07 ***
cylinders    -1.028032   0.653607  -1.573   0.1158
weight       -0.002922   0.001137  -2.569   0.0102 *
displacement  0.002462   0.015030   0.164   0.8699
horsepower   -0.050611   0.025209  -2.008   0.0447 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 289.58  on 209  degrees of freedom
Residual deviance:  83.24  on 205  degrees of freedom
AIC: 93.24

Number of Fisher Scoring iterations: 7
```

```
> 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 89 11
       1 11 71
> paste("Коефіцієнт помилок: ", mean(pred.glm != mpg01.test))
[1] "Коефіцієнт помилок:  0.120879120879121"
```

Отже в результаті бачимо, що тестова помилка отриманої моделі є 12.1%

2.7 Метод К-найближчих сусідів з різними значеннями К

```
> train.X = cbind(cylinders, weight, displacement, horsepower)[train, ]
> test.X = cbind(cylinders, weight, displacement, horsepower)[!train, ]
> mpg01.train = mpg01[train]
> set.seed(1)
>
> pred.knn = knn(train.X, test.X, mpg01.train, k = 1)
> table(pred.knn, mpg01.test)
        mpg01.test
pred.knn  0  1
       0 83 11
       1 17 71
> paste("Коефіцієнт помилок: ", mean(pred.knn != mpg01.test))
[1] "Коефіцієнт помилок:  0.153846153846154"
>
> pred.knn2 = knn(train.X, test.X, mpg01.train, k = 2)
> table(pred.knn2, mpg01.test)
         mpg01.test
pred.knn2  0  1
        0 81  9
        1 19 73
> paste("Коефіцієнт помилок: ", mean(pred.knn2 != mpg01.test))
[1] "Коефіцієнт помилок:  0.153846153846154"
>
> pred.knn3 = knn(train.X, test.X, mpg01.train, k = 4)
> table(pred.knn3, mpg01.test)
         mpg01.test
pred.knn3  0  1
        0 84  8
        1 16 74
> paste("Коефіцієнт помилок: ", mean(pred.knn3 != mpg01.test))
[1] "Коефіцієнт помилок:  0.131868131868132"
> pred.knn4 = knn(train.X, test.X, mpg01.train, k = 8)
> table(pred.knn4, mpg01.test)
         mpg01.test
pred.knn4  0  1
        0 78  7
        1 22 75
> paste("Коефіцієнт помилок: ", mean(pred.knn4 != mpg01.test))
[1] "Коефіцієнт помилок:  0.159340659340659"
```

В результаті виведено матриці помилок при значеннях K= 1, 2, 4, 8. З цих значень добре видно, що для значення K=4 значення тестової помилки є найкращим, а саме 13.2%.

3. Написання функцій

3.1

```
> Power = function() {2^3}
> paste("2 ^ 3 =", Power())
[1] "2 ^ 3 = 8"
```

3.2

```
> Power2 = function(x, a) {x^a}
```

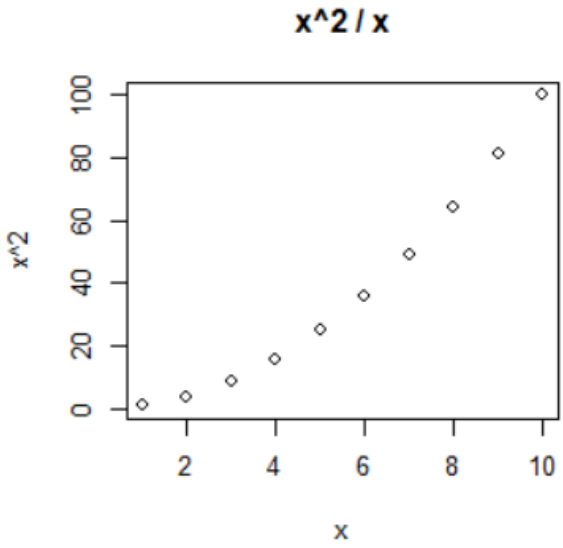
3.3

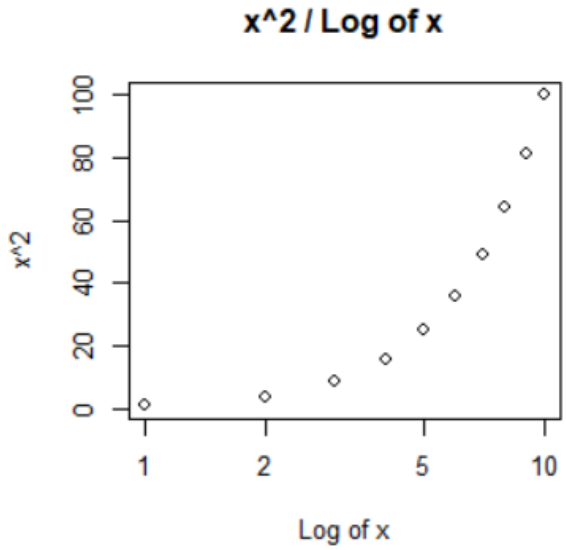
```
> paste("2 ^ 2 =", Power2(2, 2))
[1] "2 ^ 2 = 4"
> paste("7 ^ 3 =", Power2(7, 3))
[1] "7 ^ 3 = 343"
> paste("10 ^ 5 =", Power2(10, 5))
[1] "10 ^ 5 = 1e+05"
```

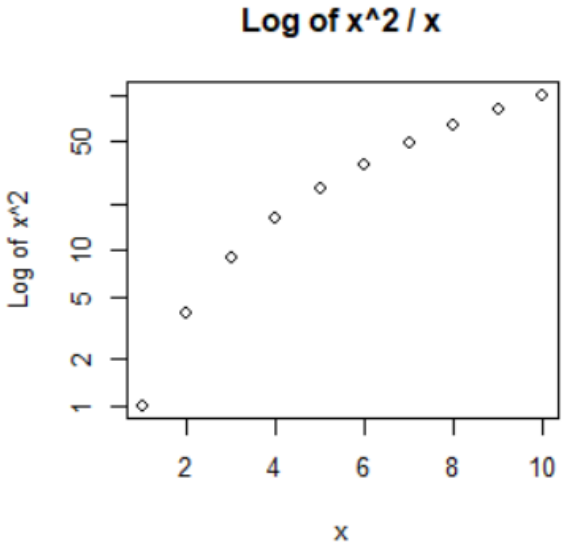
3.4

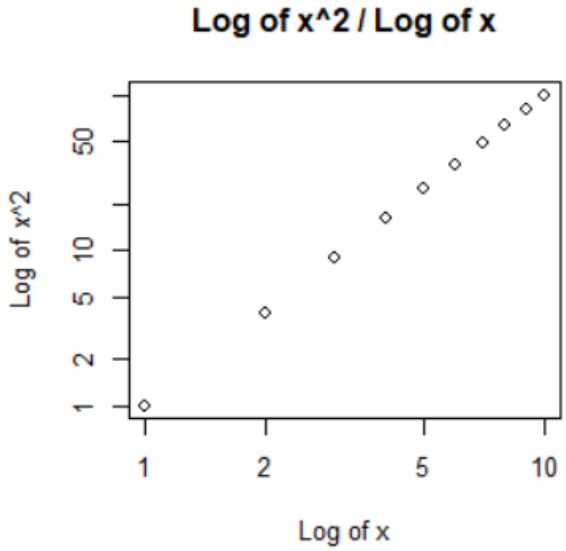
```
> Power3 = function(x, a) {
+     result = x^a
+     return(result)
+ }
>
> power_2_3 = Power3(2, 3)
> paste("2 ^ 3 =", power_2_3)
[1] "2 ^ 3 = 8"
```

3.5

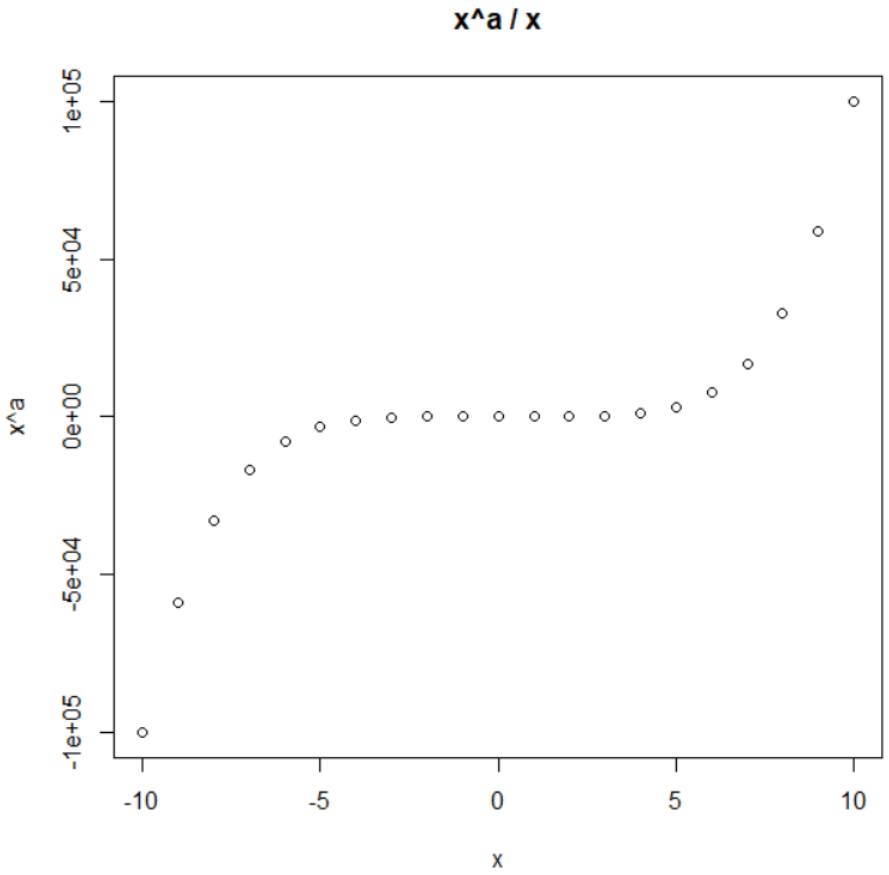

x^2 / x
x^2
0
20
40
60
80
100
2
4
6
8
10
x


x^2 / Log of x
x^2
0
20
40
60
80
100
1
2
5
10
Log of x


Log of x^2 / x
Log of x^2
1
2
5
10
50
2
4
6
8
10
x


Log of x^2 / Log of x
Log of x^2
1
2
5
10
50
1
2
5
10
Log of x

3.6 Для прикладу, використано графік де x від -10 до 10 в 5 степені.



4. Boston

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

З початку було додано змінну crim01, де її значення 1 якщо crim більше медіани, та 0 якщо менше. Змінну crim01 було додано до Boston. Після цього дані поділені на тренувальну та тестову вибірку.

Для побудови логістичної регресії в ролі предикторів було взято всі змінні, окрім crim01 та crim,

```
> fit.glm = glm(crim01 ~. - crim01 - crim, data = Boston, family = binomial, subset = train)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> probs = predict(fit.glm, Boston.test, type = "response")
> pred.glm = rep(0, length(probs))
> pred.glm[probs > 0.5] = 1
> table(pred.glm, crim01.test)
        crim01.test
pred.glm   0   1
       0  68  24
       1  22 139
> paste("Коефіцієнт помилок: ", mean(pred.glm != crim01.test))
[1] "Коефіцієнт помилок:  0.181818181818182"
```

З отриманих даних видно, що тестова помилка отриманої моделі є 18.2%. Наступною розглянемо модель лінійного дискримінантного аналізу.

```
> fit.lda = lda(crim01 ~. - crim01 - crim, data = Boston, subset = train)
> pred.lda = predict(fit.lda, Boston.test)
> table(pred.lda$class, crim01.test)
   crim01.test
      0   1
  0  80  24
  1  10 139
> paste("Коефіцієнт помилок: ", mean(pred.lda$class != crim01.test))
[1] "Коефіцієнт помилок:  0.134387351778656"
```

Тут тестова помилка буде 13.4%.

```
> fit.qda = qda(crim01 ~. - crim01 - crim, data = Boston, subset = train)
> pred.qda = predict(fit.qda, Boston.test)
> table(pred.qda$class, crim01.test)
   crim01.test
      0   1
  0  84 159
  1   6   4
> paste("Коефіцієнт помилок: ", mean(pred.qda$class != crim01.test))
[1] "Коефіцієнт помилок:  0.652173913043478"
```

В цьому випадку тестова помилка 65%. Що означає що дана модель не підходить для даної задачі

Після цього, розглянемо метод К-найближчих сусідів з різними значеннями К

```
> library(class)
> train.X = cbind(indus, chas, nox, rm, age, dis,
+  tax, ptratio, black, lstat, medv)[train, ]
> test.X = cbind(indus, chas, nox, rm, age, dis,
+  tax, ptratio, black, lstat, 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
> paste("Коефіцієнт помилок: ", mean(pred.knn != crim01.test))
[1] "Коефіцієнт помилок:  0.474308300395257"
>
> 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
> paste("Коефіцієнт помилок: ", mean(pred.knn2 != crim01.test))
[1] "Коефіцієнт помилок:  0.343873517786561"
>
> pred.knn3 = knn(train.X, test.X, crim01.train, k = 4)
> table(pred.knn3, crim01.test)
         crim01.test
pred.knn3   0   1
        0  83  53
        1   7 110
> paste("Коефіцієнт помилок: ", mean(pred.knn3 != crim01.test))
[1] "Коефіцієнт помилок:  0.237154150197628"
>
> pred.knn4 = knn(train.X, test.X, crim01.train, k = 8)
> table(pred.knn4, crim01.test)
         crim01.test
pred.knn4   0   1
        0  84  26
        1   6 137
> paste("Коефіцієнт помилок: ", mean(pred.knn4 != crim01.test))
[1] "Коефіцієнт помилок:  0.126482213438735"
```

В результаті виведено матриці помилок при значеннях K= 1, 2, 4, 8. З цих значень добре видно, що для значення K=8 значення тестової помилки є найкращим, а саме 12.6%.