

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский Политехнический Университет Петра Великого

—
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта

ЛАБОРАТОРНАЯ РАБОТА №6

«Линейная регрессия»

по дисциплине «Машинное обучение, часть1»

Выполнил: студент группы
3540201/20302

С.А. Ляхова

<подпись>

Проверил:
д.т.н., профессор

Л.В. Уткин

<подпись>

Санкт-Петербург
2022

Содержание

1. Цель работы	3
2. Формулировка задания	3
3. Ход работы	5
4. Вывод	21
Приложение 1	22
Приложение 2	23
Приложение 3	24
Приложение 4	25
Приложение 5	27
Приложение 6	29
Приложение 7	31
Приложение 8	32
Приложение 9	33

1. Цель работы

Исследовать метод `lm` языка R, восстанавливающий регрессию с помощью QR-разложения матрицы X, выполнив поставленные задачи и проанализировав результаты.

2. Формулировка задания

1. Загрузите данные из файла `reglab1.txt`. Используя функцию `lm`, постройте регрессию (используйте разные модели). Выберите наиболее подходящую модель, объясните свой выбор.

2. Реализуйте следующий алгоритм для уменьшения количества признаков, используемых для построения регрессии: для каждого $k \in \{0, 1, \dots, d\}$ выбрать подмножество признаков мощности k^1 , минимизирующее остаточную сумму квадратов RSS. Используя полученный алгоритм, выберите оптимальное подмножество признаков для данных из файла `reglab2.txt`. Объясните свой выбор. Для генерации всех возможных сочетаний по m элементов из некоторого множества x можно использовать функцию `combn(x, m, ...)`.

3. Загрузите данные из файла `cugage.txt`. Постройте регрессию, выражающую зависимость возраста исследуемых отложений от глубины залегания, используя веса наблюдений. Оцените качество построенной модели.

4. Загрузите данные Longley (макроэкономические данные). Данные состоят из 7 экономических переменных, наблюдаемых с 1947 по 1962 годы ($n=16$):

GNP.deflator - дефлятор цен,

GNP - валовой национальный продукт,

Unemployed – число безработных

Armed.Forces – число людей в армии

Population – население, возраст которого старше 14 лет

Year - год

Employed – количество занятых

Построить регрессию `lm(Employed ~ .)`.

Исключите из набора данных longley переменную "Population". Разделите данные на тестовую и обучающую выборки равных размеров случайным образом. Постройте гребневую регрессию для значений $\lambda = 10^{-3+0.2 \cdot i}, i = 0, \dots, 25$, подсчитайте ошибку на тестовой и обучающей

выборке для данных значений λ , постройте графики. Объясните полученные результаты.

5. Загрузите данные EuStockMarkets из пакета «datasets». Данные содержат ежедневные котировки на момент закрытия фондовых бирж: Germany DAX (Ibis), Switzerland SMI, France CAC, и UK FTSE. Постройте на одном графике все кривые изменения котировок во времени. Постройте линейную регрессию для каждой модели в отдельности и для всех моделей вместе. Оцените, какая из бирж имеет наибольшую динамику.

6. Загрузите данные JohnsonJohnson из пакета «datasets». Данные содержат поквартальную прибыль компании Johnson & Johnson с 1960 по 1980 гг. Постройте на одном графике все кривые изменения прибыли во времени. Постройте линейную регрессию для каждого квартала в отдельности и для всех кварталов вместе. Оцените, в каком квартале компания имеет наибольшую и наименьшую динамику доходности. Сделайте прогноз по прибыли в 2016 году во всех кварталах и в среднем по году.

7. Загрузите данные sunspot.year из пакета «datasets». Данные содержат количество солнечных пятен с 1700 по 1988 гг. Постройте на графике кривую изменения числа солнечных пятен во времени. Постройте линейную регрессию для данных.

8. Загрузите данные из файла пакета «UKgas.scv». Данные содержат объемы ежеквартально потребляемого газа в Великобритании с 1960 по 1986 гг. Постройте линейную регрессию для каждого квартала в отдельности и для всех кварталов вместе. Оцените, в каком квартале потребление газа имеет наибольшую и наименьшую динамику доходности. Сделайте прогноз по потреблению газа в 2016 году во всех кварталах и в среднем по году.

9. Загрузите данные cars из пакета «datasets». Данные содержат зависимости тормозного пути автомобиля (футы) от его скорости (мили в час). Данные получены в 1920 г. Постройте регрессионную модель и оцените длину тормозного пути при скорости 40 миль в час.

3. Ход работы

Задание №1

Данные reglab.txt

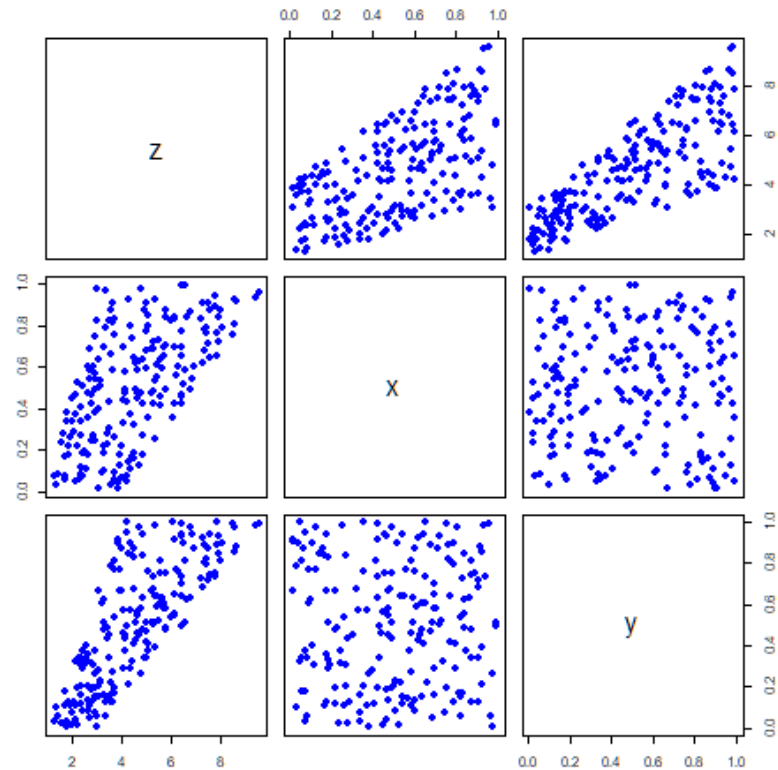


Рисунок 1. Исходные данные датасета reglab1

Регрессия построена для трех моделей: $z(x, y)$, $x(y, z)$ и $y(x, z)$.

```
> f
Call:
lm(formula = y ~ ., data = reglab, model = TRUE)

Coefficients:
(Intercept)          z          x
  0.02861      0.19229     -0.78821

> summary(f)

Call:
lm(formula = y ~ ., data = reglab, model = TRUE)

Residuals:
    Min       1Q   Median       3Q      Max
-0.178865 -0.040686 -0.001283  0.030132  0.190856

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.028606   0.012430   2.301   0.0224 *
z            0.192293   0.003126  61.517 <2e-16 ***
x           -0.788209   0.021543 -36.588 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06659 on 197 degrees of freedom
Multiple R-squared:  0.9505,    Adjusted R-squared:  0.95
F-statistic: 1893 on 2 and 197 DF,  p-value: < 2.2e-16
```

Рисунок 2 Регрессионная модель y от x, z

Наименьшие значения стандартных ошибок: 0.012430 для b_0 , 0.003126 для b_1 , 0.021543 для b_2 у модели $y(x,z)$, признанная наиболее подходящей.

Была выбрана регрессионная модель

$$Y = 0.02861 + 0.12229 Z - 0.78821 X$$

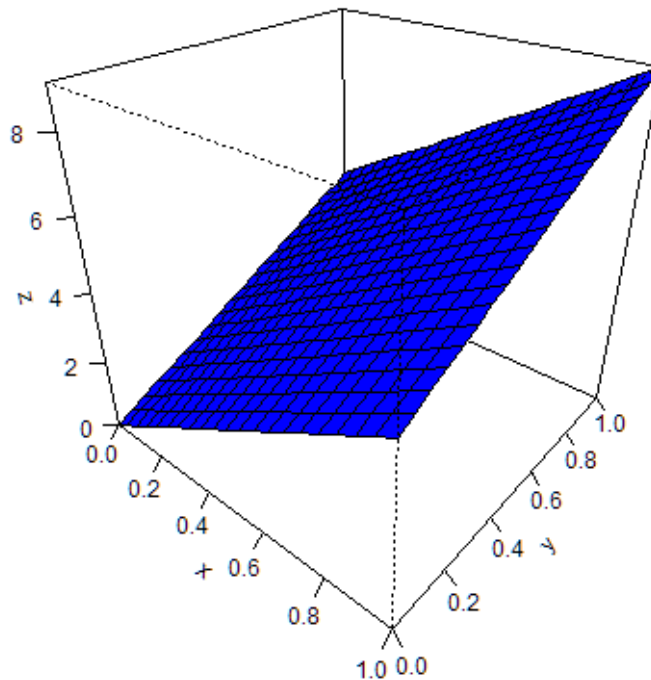


Рисунок 3. Полученная регрессия

Задание №2

Данные reglab2.txt

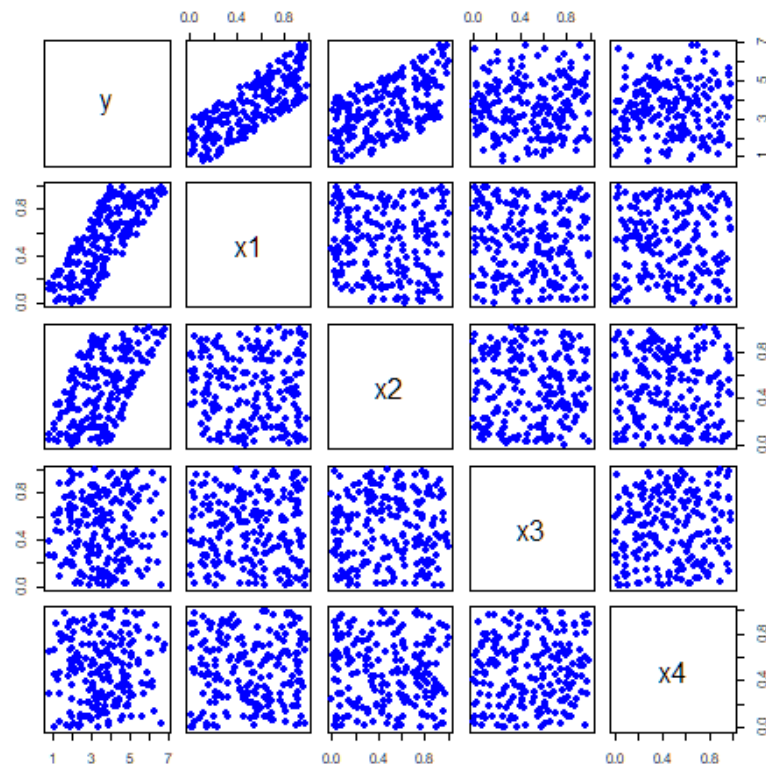


Рисунок 4. Исходные данные датасета reglab2

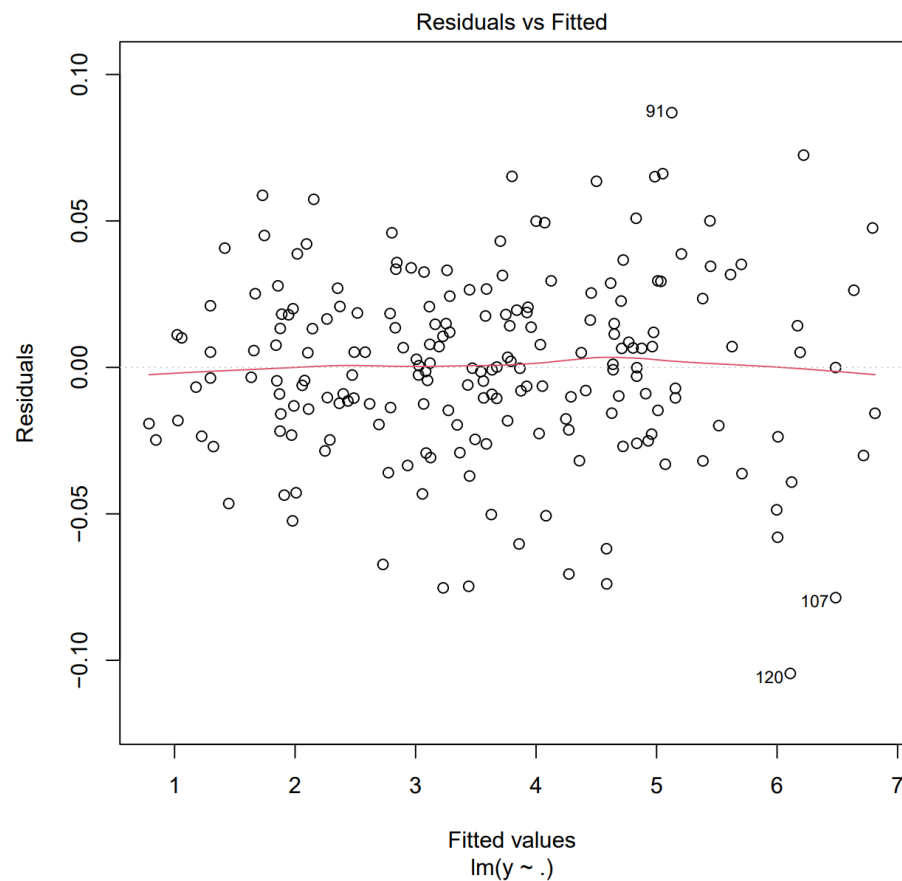


Рисунок 5. График соотношения residuals vs fitted

В результате работы программы была построена регрессионная модель $y = -0.00964 + 3.99699x_1 + 3.011691x_2 + 0.10226x_3 + 0.094398x_4$.

Из этого следует, что оптимальным подмножеством параметров является все множество параметров.

Данная модель обладает минимальным из возможных в рамках предложенного алгоритма стандартным отклонением 0.1928635.

Задание №3

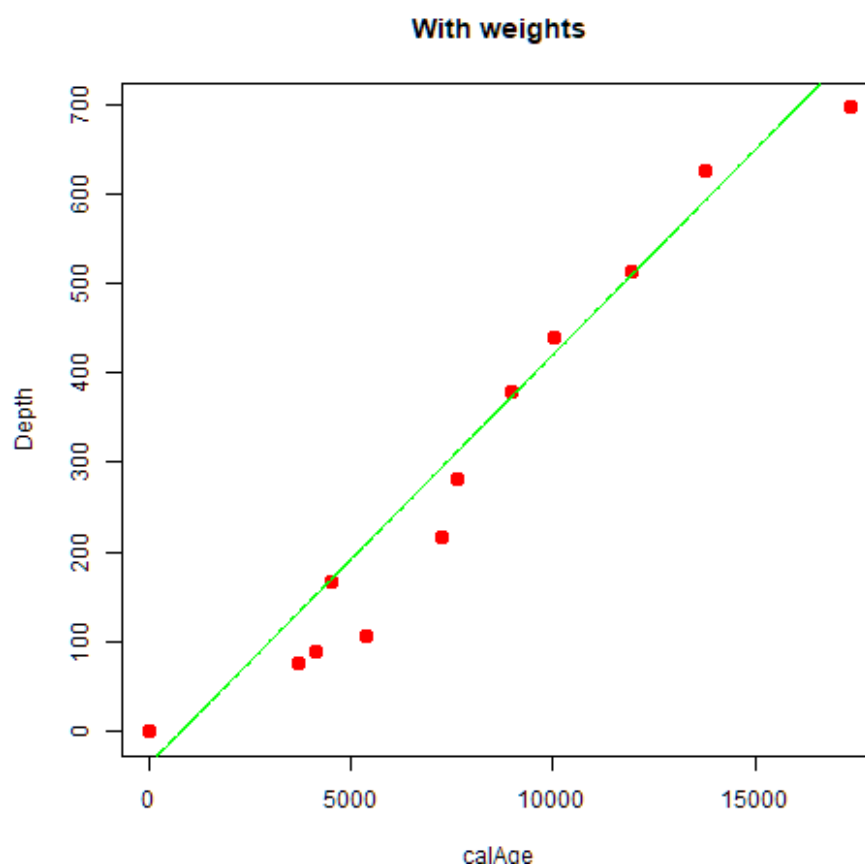


Рисунок 6. Исходные данные с регрессией, построенной с использованием весов

$$calAge = 784.56992 + 21.90919Depth$$

$$RSS = 2725934$$

Регрессия, построенная с использованием весов, оказывается точнее, чем регрессия, построенная без их учета.

Задание №4

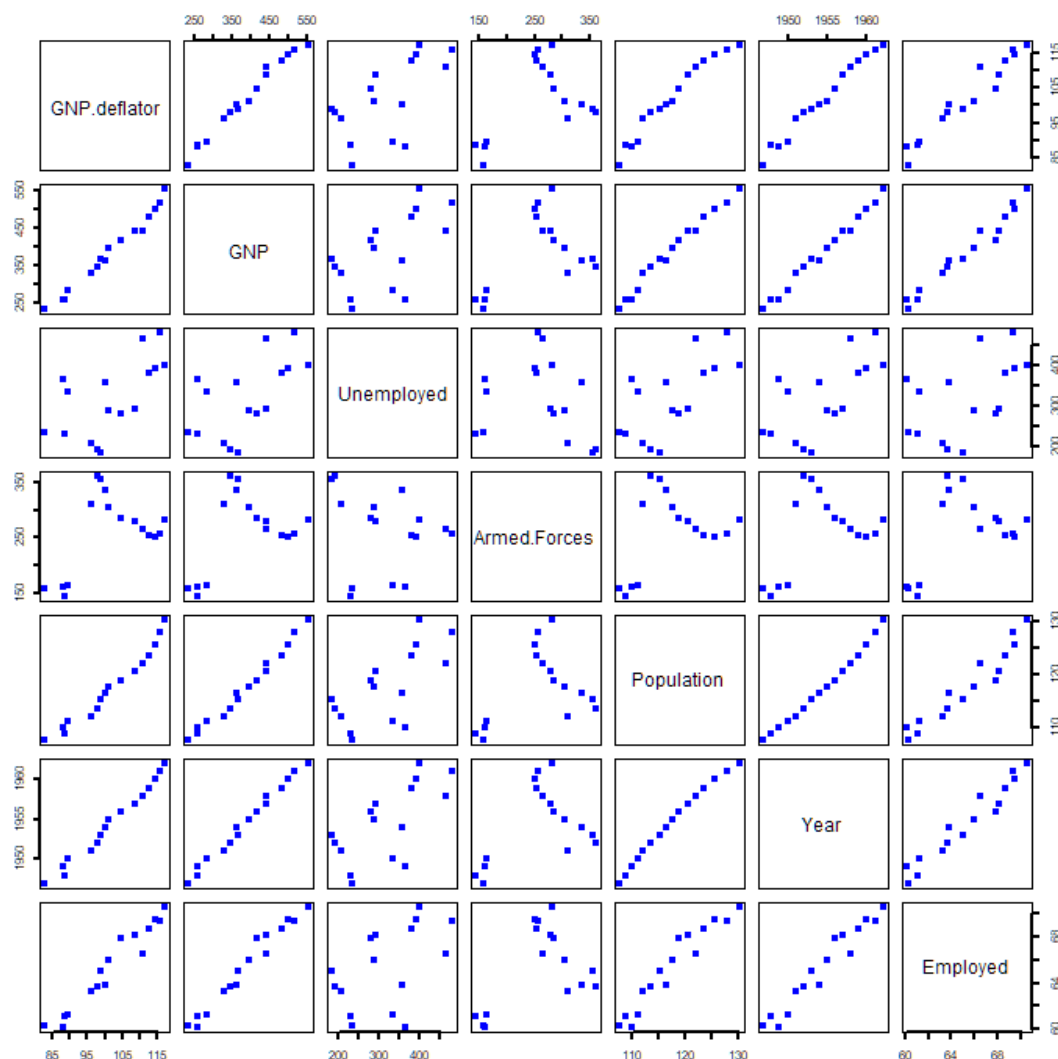


Рисунок 7. Исходные данные longley

Построена регрессионная модель зависимости количества занятых от всех остальных категориальных переменных.

В результате эксперимента была получена следующая линейная регрессия:

$$\begin{aligned} \text{Employed} = & -3482 + 0.01506 \text{ GNP.deflator} - 0.03582 \text{ GNP} - 0.0202 \text{ Unemployed} \\ & - 0.01033 \text{ Armed.Forces} - 0.0511 \text{ Population} + 1.829 \text{ Year} \end{aligned}$$

Далее из набора данных longley была исключена переменная "Population". Данные были разделены на тестовую и обучающую выборки. На основе обучающей выборки была построена гребневая регрессия для значений $\lambda = 10^{-3+0.2 \cdot i}, i = 0, \dots, 25$.

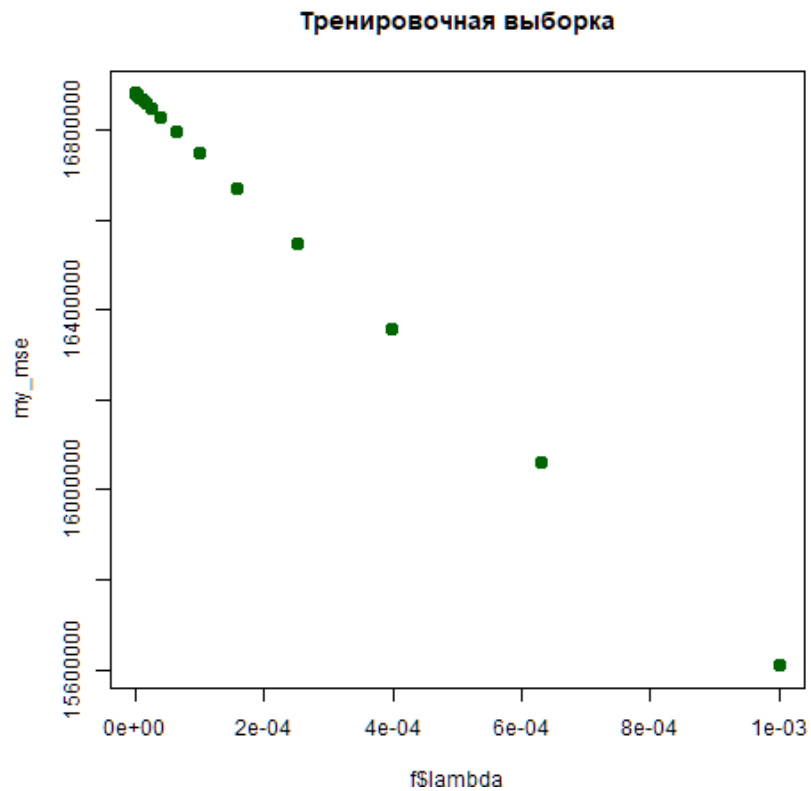


Рисунок 8. Зависимость MSE от λ для тренировочной выборки при гребневой регрессии

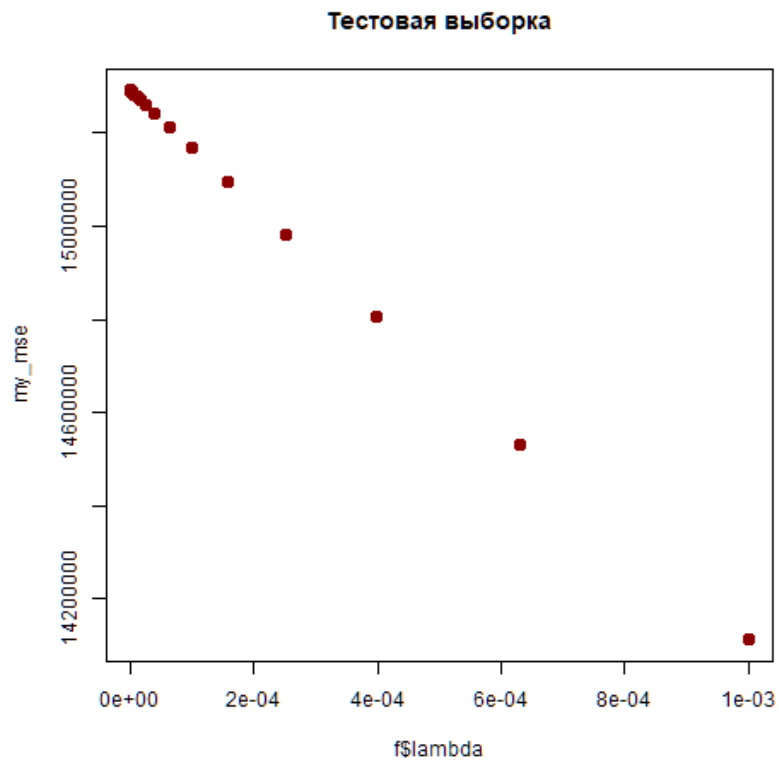


Рисунок 9. Зависимость MSE от λ для тестовой выборки при гребневой регрессии

Из графиков можно сделать выводы:

- среднеквадратичная ошибка на тестовой выборке меньше, чем на обучающей (модель не переобучена);
- среднеквадратичная ошибка убывает с увеличением значения параметра λ .

Задание №5

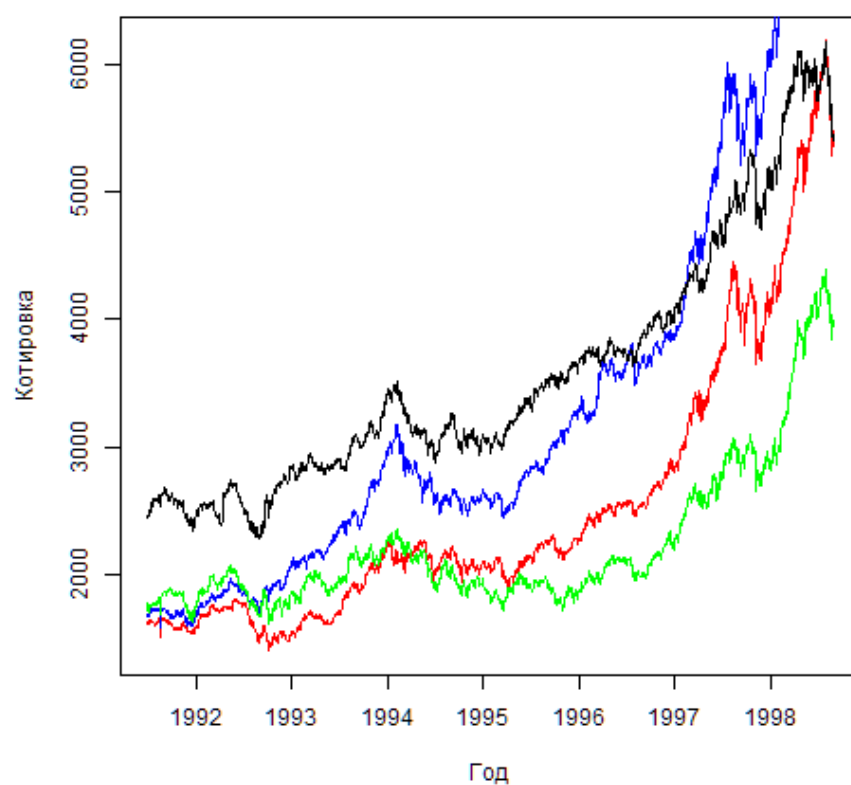


Рисунок 10. Исходные данные фондовых бирж на одном графике
Deutscher Aktienindex (DAX)

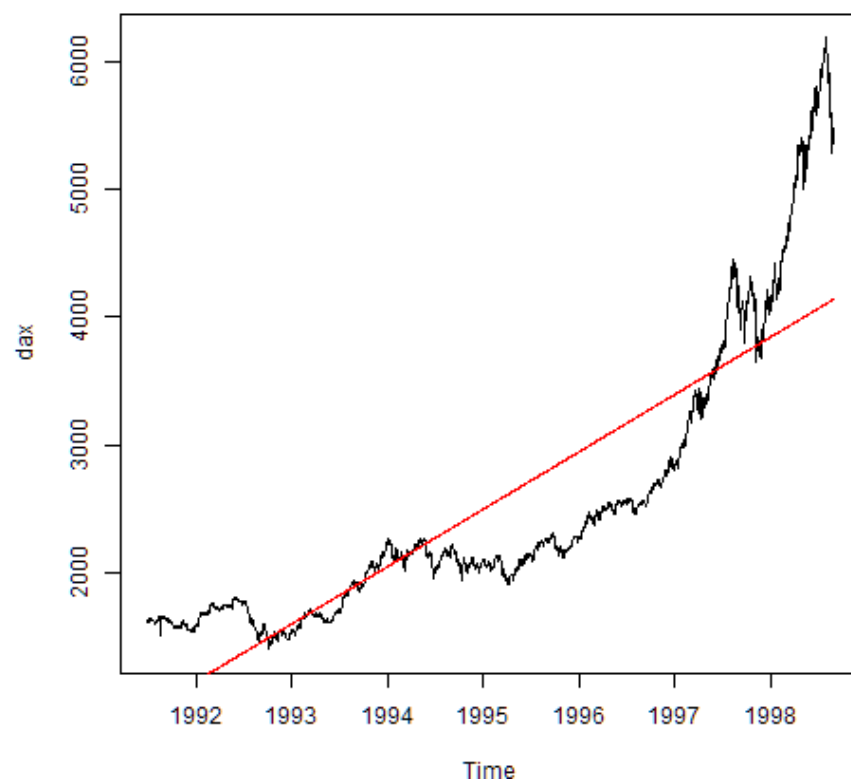


Рисунок 11. Регрессия и данные биржи DAX
 $DAX = -894557.9 + 449.7 \text{ Time}; RSS = 583772212$

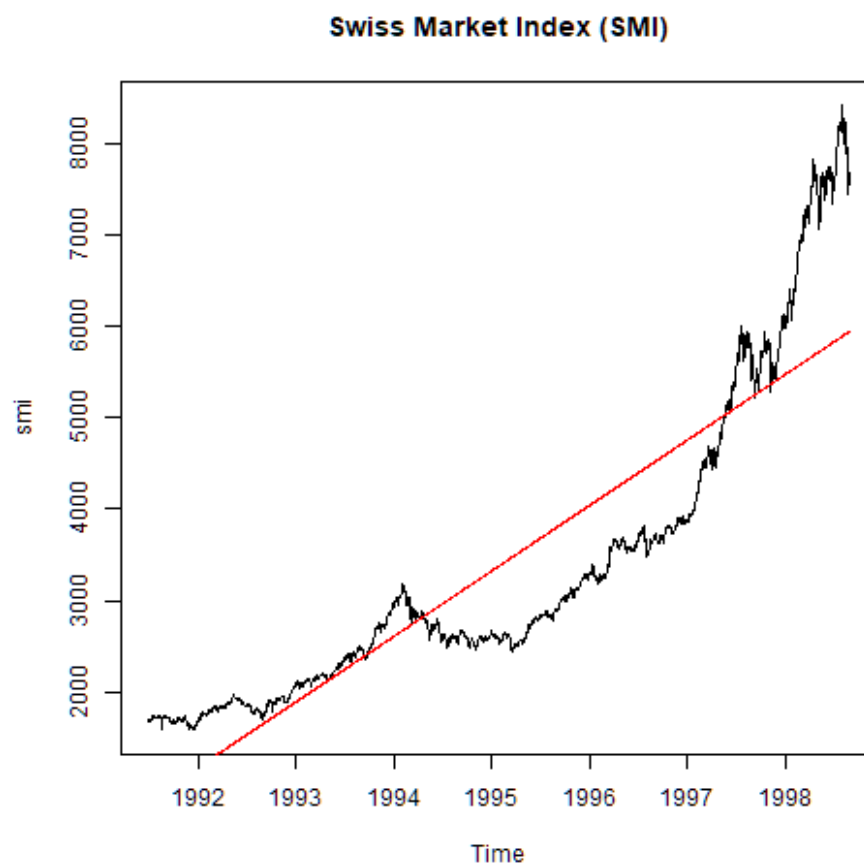


Рисунок 12. Регрессия и данные биржи SMI
 $SMI = -1428160.2 + 717.5 \text{ Time}; RSS = 1057233822$
Cotation Assistée en Continu (CAC)

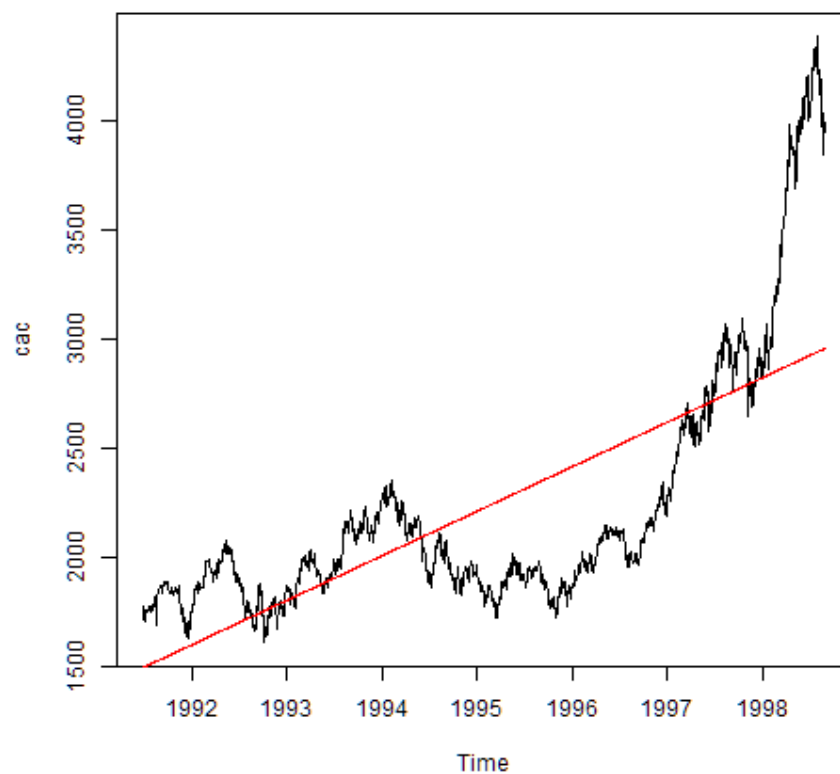


Рисунок 13. Регрессия и данные биржи CAC
 $CAC = -405915.3 + 204.6 \text{ Time}; RSS = 294059996$

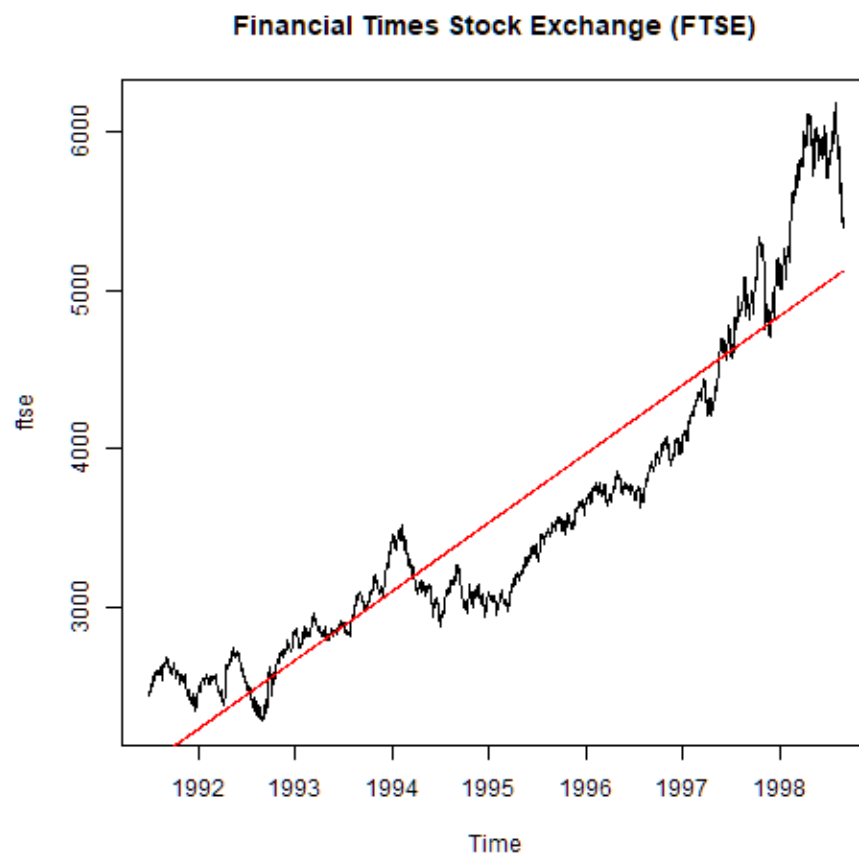


Рисунок 14. Регрессия и данные биржи FTSE
 $FTSE = -865200.4 + 435.5 \text{ Time}; RSS = 269256724$
Overall

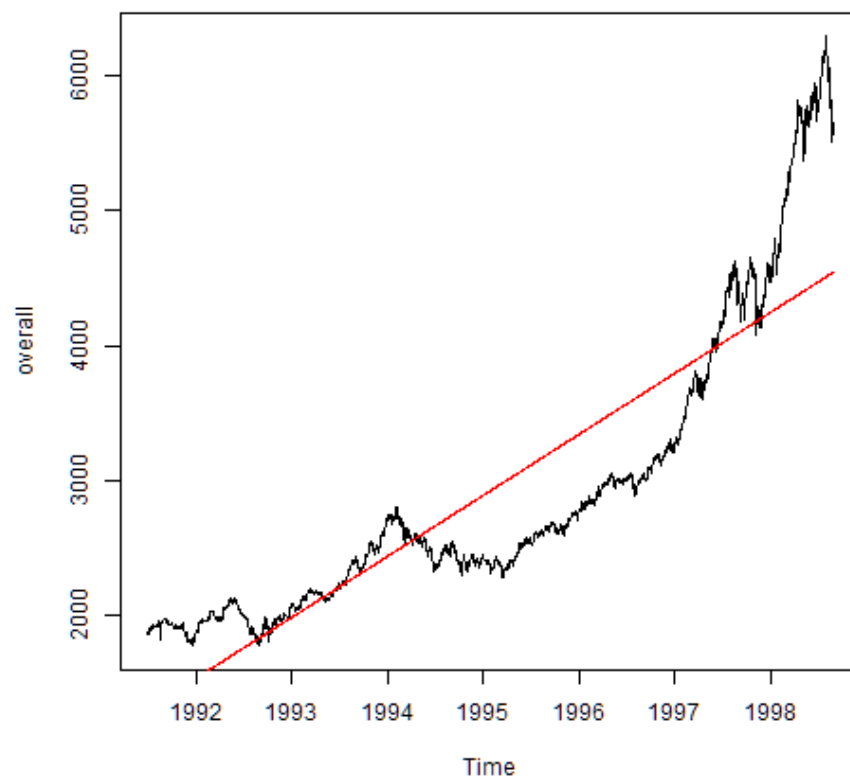


Рисунок 15. Регрессия и данные всех бирж
 $DAX + SMI + CAC + FTSE = -3593834 + 1807 \text{ Time};$
 $RSS = 492333681$

Коэффициент при параметре *Time* больше всего у Switzerland SMI, а значит именно эта биржа имеет наибольшую динамику.

Также можно заметить, что наибольшую RSS имеет биржа SMI, а наименьшую – FTSE.

Задание №6



Рисунок 16. Изменение прибыли во времени

Красным цветом представлено изменение прибыли во времени с 1960 по 1980 гг. в первый квартал, синим - во второй, зеленым - в третий и черным - в четвертый.

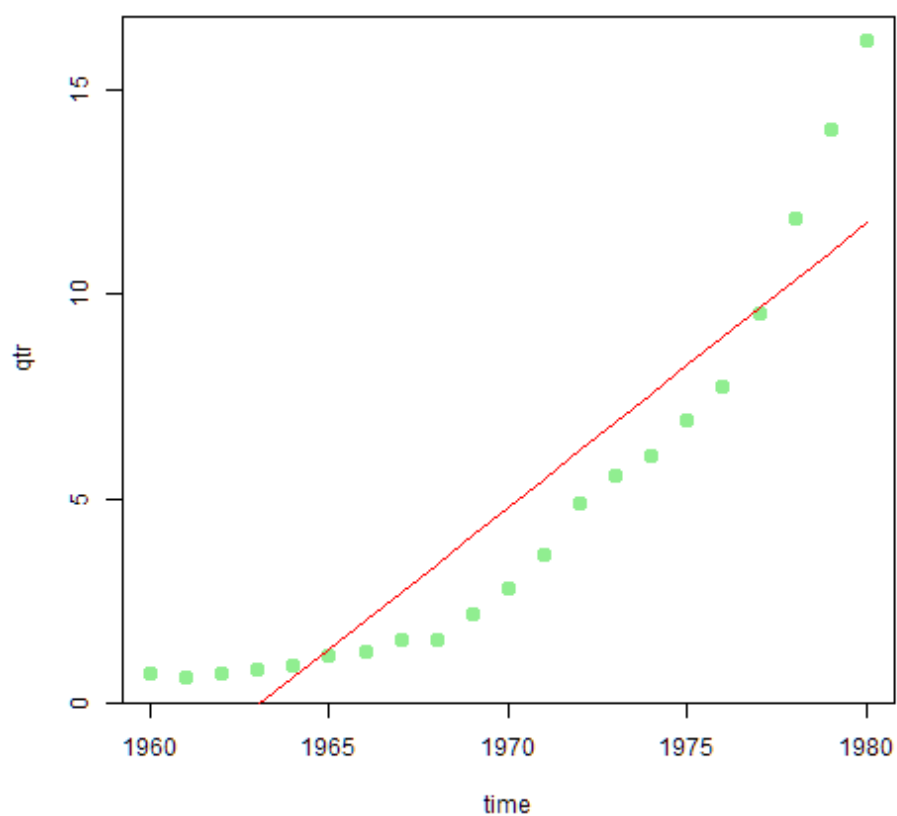


Рисунок 17. Исходные данные JohnsonJohnson за первый квартал
 $Qtr1 = -1364.282 + 0.695 \text{ Time}$

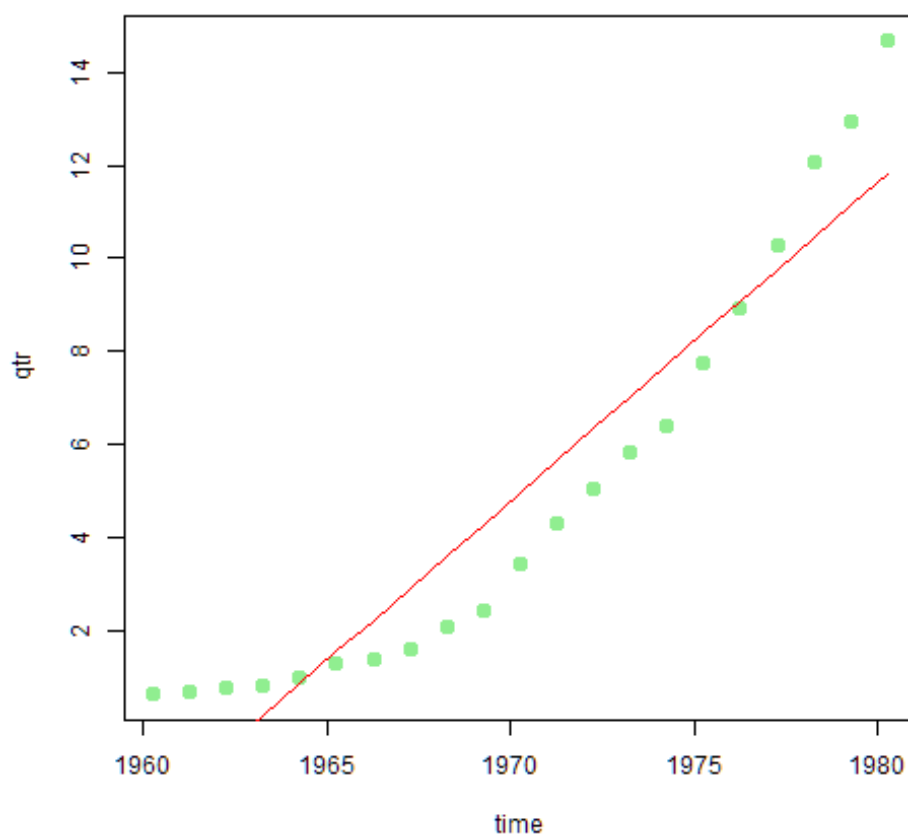


Рисунок 18. Исходные данные JohnsonJohnson за второй квартал
 $Qtr2 = -1345.0727 + 0.6853 \text{ Time}$

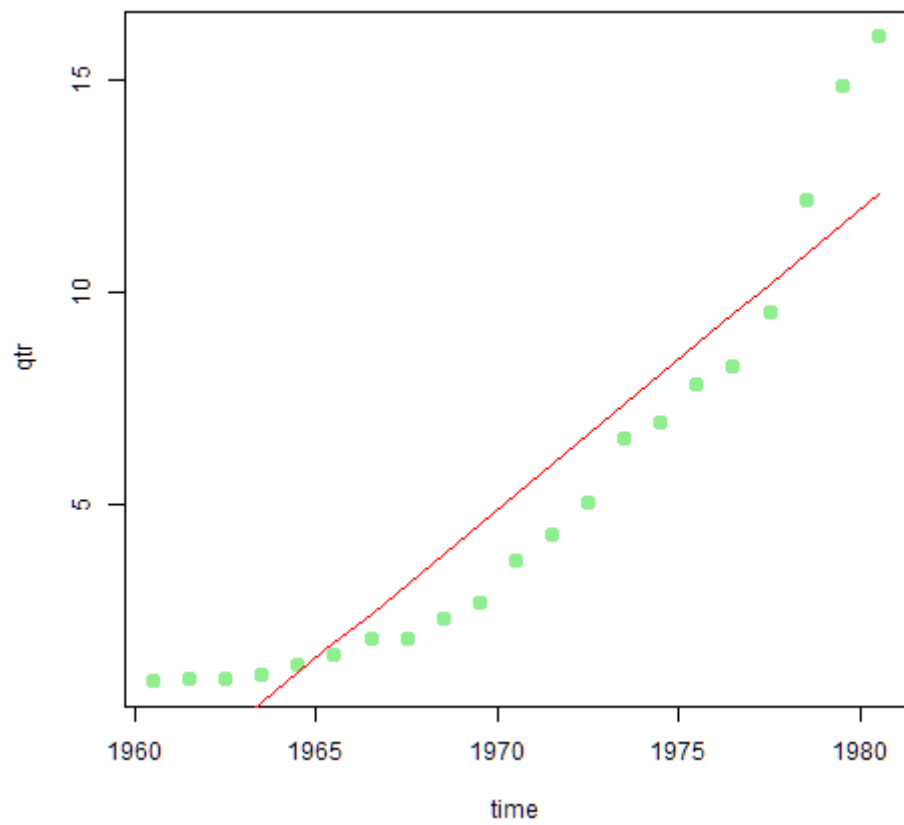


Рисунок 19. Исходные данные JohnsonJohnson за третий квартал
 $Qtr3 = -1382.3170 + 0.7044 \text{ Time}$

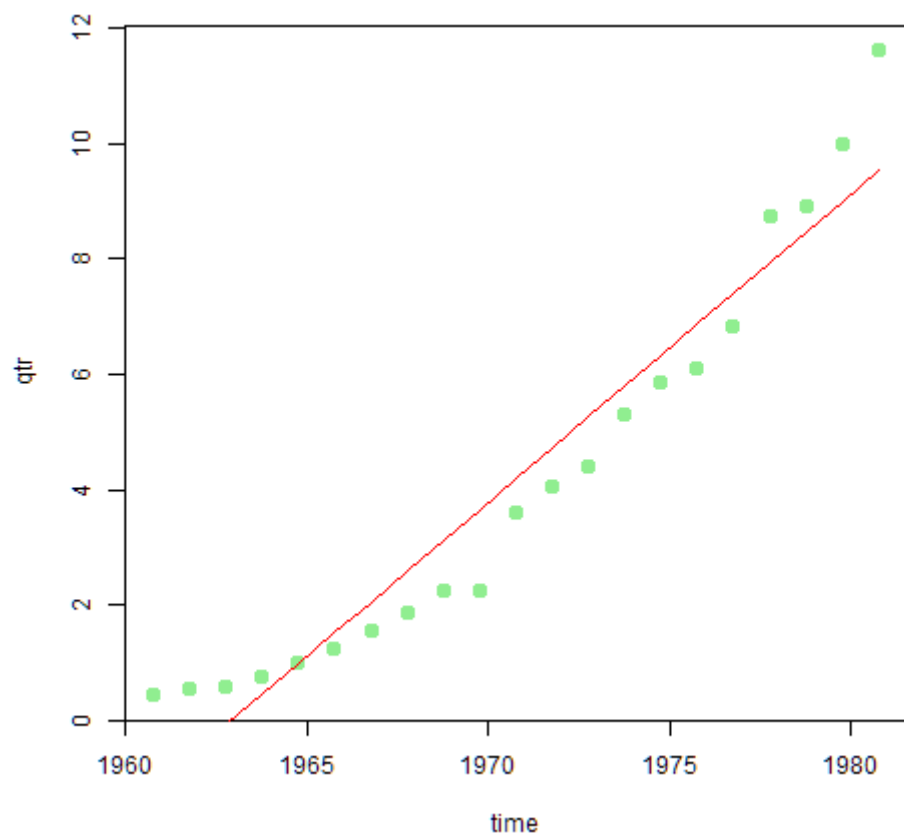


Рисунок 20. Исходные данные JohnsonJohnson за четвертый квартал
 $Qtr4 = -1049.5828 + 0.5349 \text{ Time}$

$$Qtr1 + Qtr2 + Qtr3 + Qtr4 = -5141.25 + 2.62 \text{ Time}$$

Исходя из полученных моделей, можно сказать, что наибольшую динамику доходности компания имеет в третьем квартале (коэффициент при параметре Time максимальный), а наименьшую - в четвертом (коэффициент при параметре Time минимальный).

Прогноз по прибыли на 2016 год.

- В первый квартал: 36.75964
- Во второй квартал: 36.48945
- В третий квартал: 37.65394
- В четвертый квартал: 28.79391
- В среднем по году: 34.88217

Задание №7

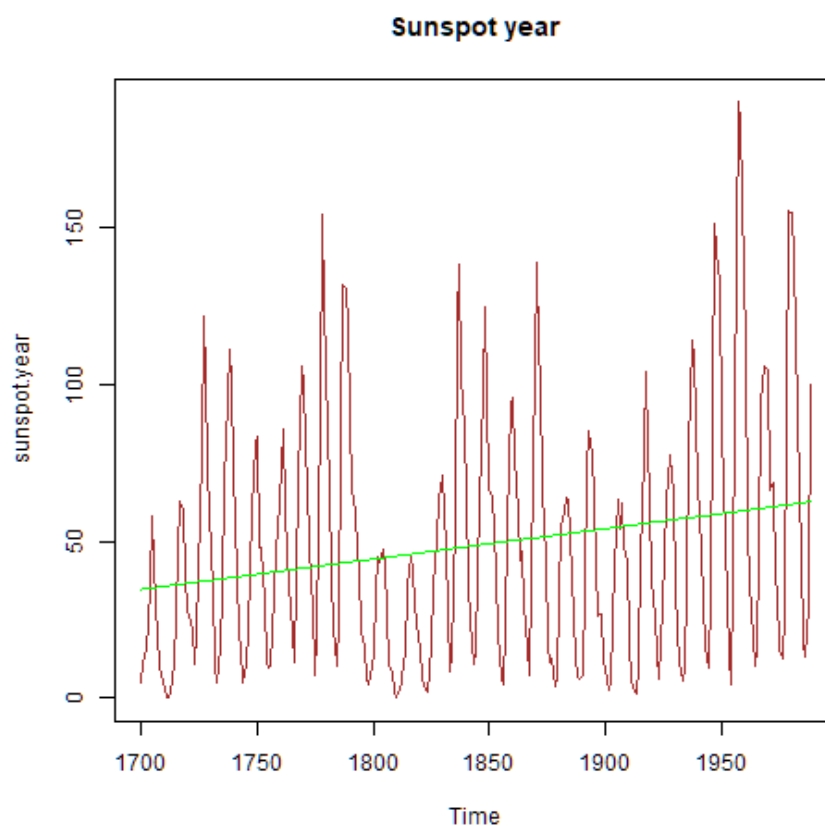


Рисунок 21. Исходные данные sunspot.year с построенной регрессией

Для данных sunspot.year построена линейная регрессия:

$$Spots = -130.42118979 + 0.09709039 \text{ Year}$$

Для построенной регрессии $RSS = 429802$

Задание №8

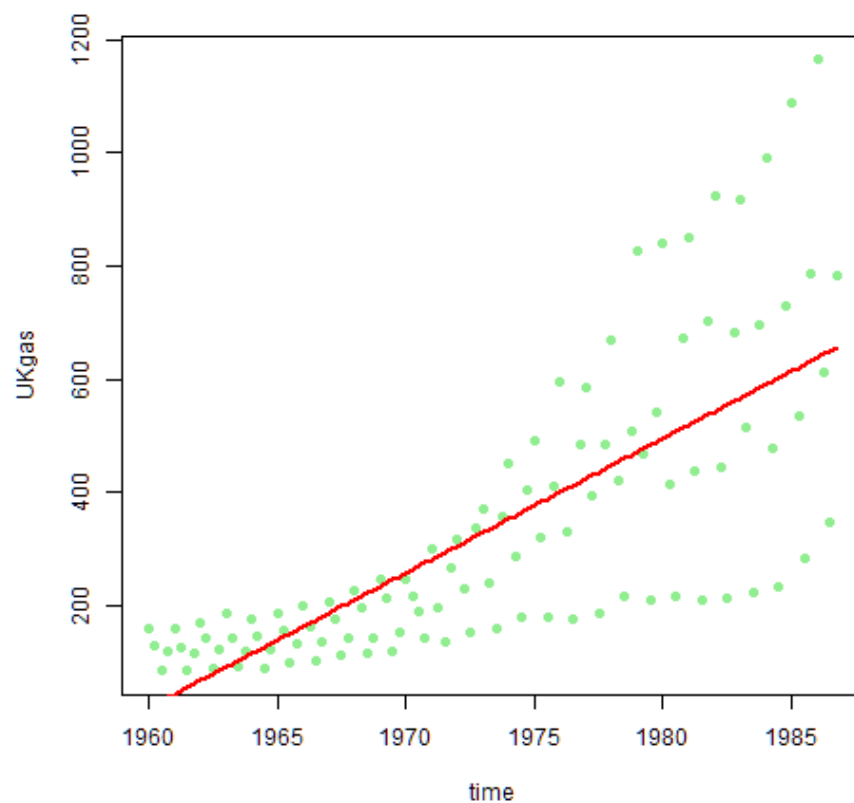


Рисунок 22. Построенная регрессия и исходные данные UKgas
 $Qtr1 + Qtr2 + Qtr3 + Qtr4 = -46604.60794 + 23.78779 \text{ Time}$

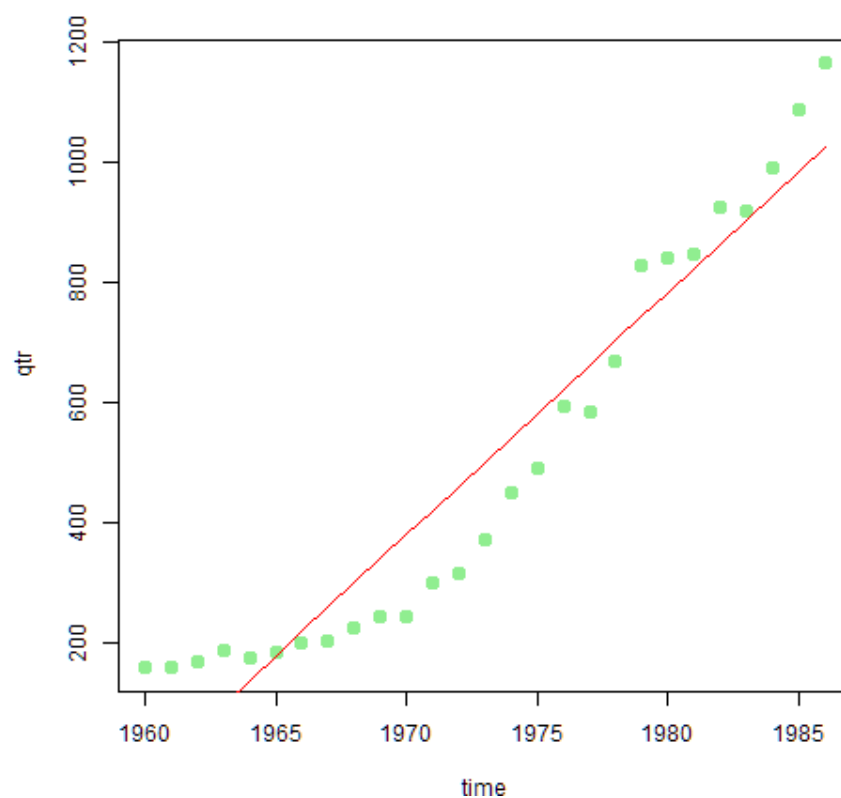


Рисунок 23. Построенная регрессия и данные UKgas за первый период
 $Qtr1 = -78854.23331 + 40.22082 \text{ Time}$

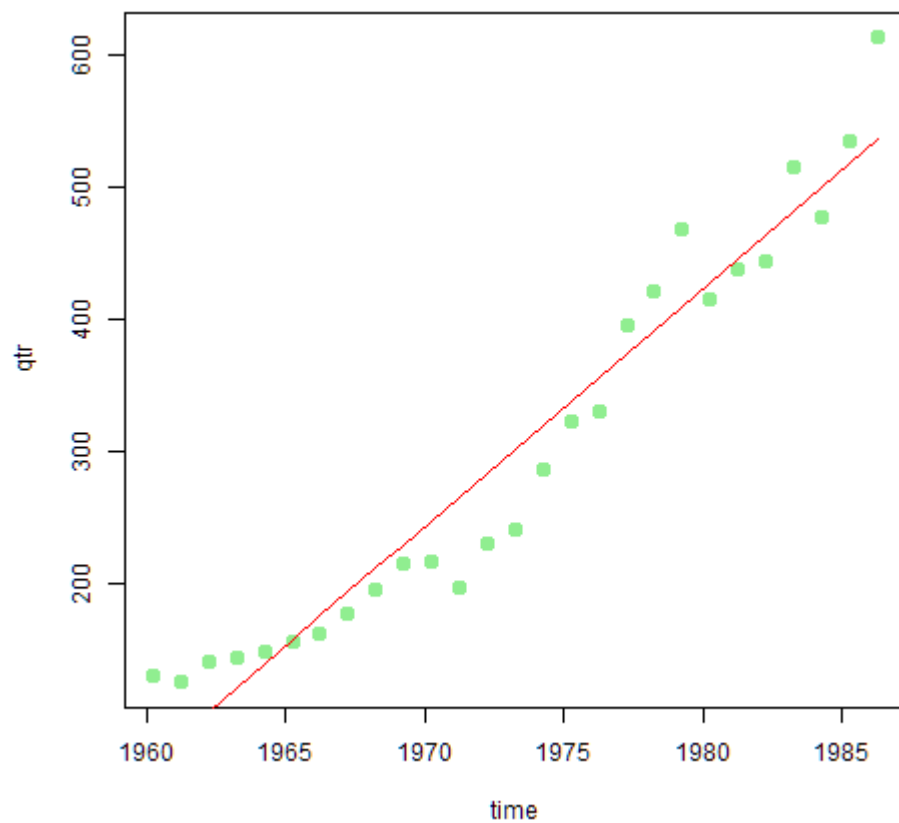


Рисунок 24. Построенная регрессия и данные UKgas за второй период
 $Qtr2 = -35297.22520 + 18.04048 Time$

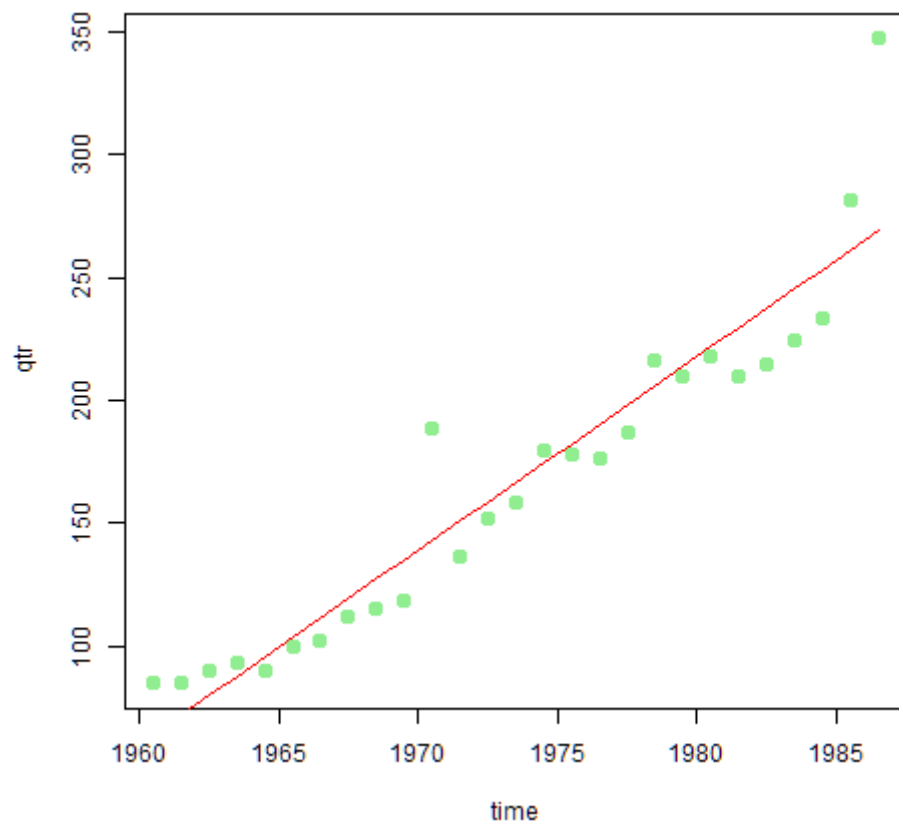


Рисунок 25. Построенная регрессия и данные UKgas за третий период
 $Qtr3 = -15403.731197 + 7.889744 Time$

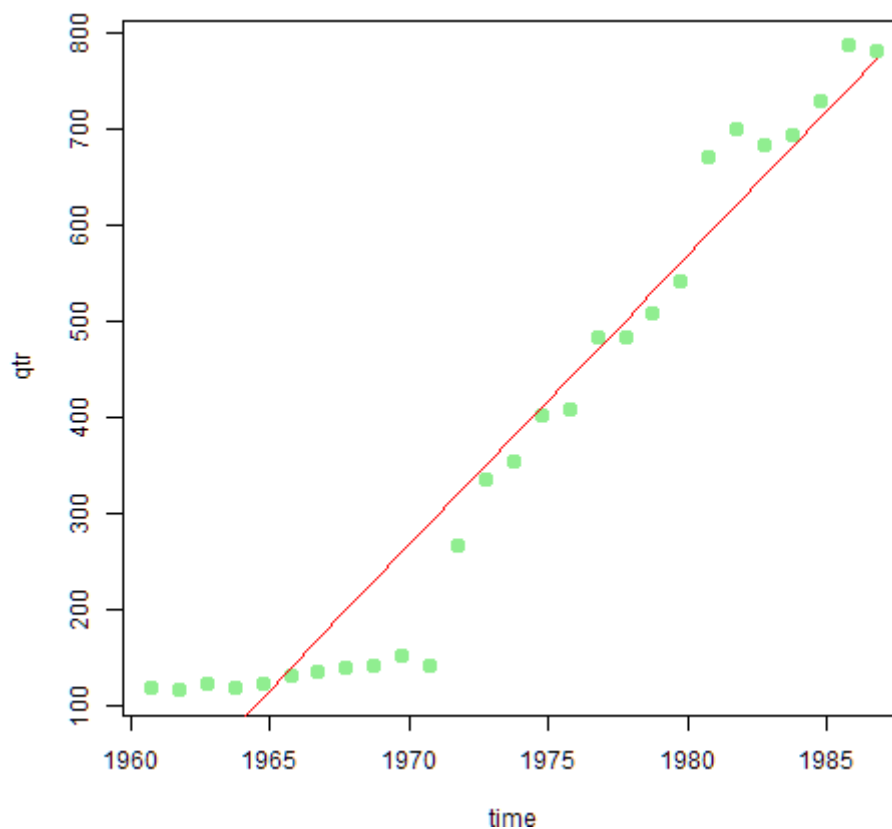


Рисунок 26. Построенная регрессия и данные UKgas за четвертый период

$$Qtr4 = -59112.7230430.14261 \text{ Time}$$

Исходя из полученных моделей, можно заключить, что наибольшую динамику доходности потребление газа имеет в первом квартале (коэффициент при параметре *Time* максимальный), а наименьшую - в третьем (коэффициент при параметре *Time* минимальный).

Прогноз по потреблению газа на 2016 год.

- В первый квартал: 2230.936
- Во второй квартал: 1072.375
- В третий квартал: 501.9919
- В четвертый квартал: 1654.785
- В среднем по году: 1372.787

Задание №9

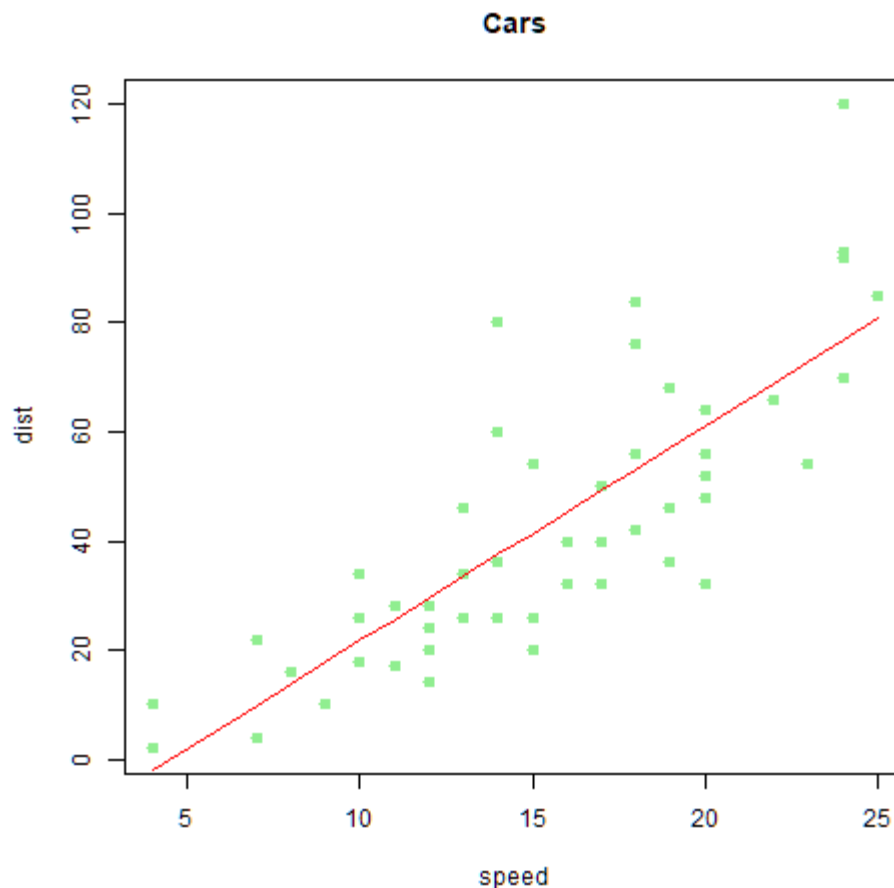


Рисунок 27. Исходные данные cars и построенная регрессия

Была построена следующая регрессионная модель:

$$dist = -17.579095 + 3.932409 \text{ speed}$$

Оценка длины тормозного пути при скорости 40 миль в час - 139.7173 футов.

4. Вывод

В ходе выполнения лабораторной работы были изучены методы линейной и гребневой регрессии. Регрессия позволяет оценить динамику изменения величины, а также предугадать ее значение.

Выбор модели оказывает решающую роль на метрики точности и ошибок. Потребность в отсечении признаков возникает в результате недостатка общности линии регрессии относительно тестовых данных.

Статистические критерии позволяют однозначно оценить значимость построенной модели путём принятия/отвержения гипотез. Однако, без учёта уровня значимости критерии не способны дать осмысленного понимания о значимости коэффициентов линии регрессии, что делает его выбор основополагающим при регрессионном анализе.

```
# Задание 1-----
library(plot3D)
library(pracma)

path <- "C:\\Users\\Софья\\Desktop\\1_курс_1_семестр\\Машинное обучение\\-
Лабораторная 6. Regression\\"

reglab <- read.table(paste(path, "reglab1.txt", sep = ""), sep = "\t", header = TRUE)
xy <- reglab[-1]
z <- reglab$z

png(paste(path, "reglab1.png"))
plot(reglab, pch = 19, col = "blue")
dev.off()

f <- lm(z ~ ., reglab)
f
summary(f)

f = lm(x ~ ., data = reglab, model = TRUE)
f
summary(f)
plot(f)

f = lm(y ~ ., data = reglab, model = TRUE)
f
summary(f)
plot(f)

x <- y <- seq(0, 1, 0.05)
r <- meshgrid(x)
z <- f$coefficients[1] + f$coefficients[2] * r$X +
  f$coefficients[3] * r$Y

png(paste(path, "surface.png"))
persp(x,y,z, theta=40, phi=30, r=2, ticktype="detailed", col="blue")
dev.off()
```

```
# Задание 2-----
library(qpcR)
reglab2 <- read.table(paste(path, 'reglab2.txt', sep = ""), sep = '\t', header = TRUE)

png(paste(path, "reglab2.png"))
plot(reglab2, pch = 19, col = "blue")
dev.off()

min_res = Inf

d = ncol(reglab2)-1

for(k in 1:d)
{
  comb <- combn(reglab2[, -1], k)
  for (s in 1:nrow(comb))
  {
    new_data <- data.frame(y = reglab2[, 1], comb[s, ])
    f <- lm(y ~ ., new_data)
    res <- RSS(f)
    if (res < min_res) {
      min_res <- res
      ans <- f
    }
  }
}

min_res
summary(ans)
ans$coefficients

pdf(paste(path, "reglab2_result.pdf"))
plot(ans)
dev.off()
```

```
# Задание 3-----
cygage <- read.table(paste(path, 'cygage.txt', sep = ""), sep = '\t', header = TRUE)

xy <- cygage[,-ncol(cygage)]
z <- cygage$Weight
y <- seq(-50, max(cygage[2]) + 50, 1)

# with weights
f <- lm(calAge ~ ., xy, weights = z)
summary(f)
png(paste(path, "With weights.png"))
plot(xy, pch = 19, col = "red", lwd = 4, main = "With weights")
lines(x = y*f$coefficients[2] + f$coefficients[1], y = y, col = "green")
dev.off()
RSS(f)
```



```
# Задание 4-----
library(datasets)
library(MASS)
library(Metrics)
data(longley)

png(paste(path, "longley.png"), width = 720, height = 720)
plot(longley, lwd = 2, pch = 19, col = "blue")
dev.off()

f <- lm(Employed ~ ., longley)
summary(f)

longl <- longley[-3]

n <- nrow(longl)
longley_rand <- longl[order(runif(n)), ]
longley_train <- longley_rand[1:as.integer(n*0.8), ]
longley_test <- longley_rand[as.integer(n*0.8):n, ]

f <- lm.ridge(
  Employed ~ .,
  longley_train,
  lambda = 10^(-3-0.2*seq(0,25, by=1)),
  model = TRUE,
  x = TRUE,
  y = TRUE
)

x <- longley_train[-6]
y <- longley_train$Employed

my_mse <- vector()

for (c in 1:ncol(f$coef))
{
  coefs <- f$coef[, c]
  tmp <- vector()
  for (s in 1:ncol(x)) tmp <- append(tmp, mse(y[s], sum(coefs*x[s, ])))
  tmp <- mean(tmp)
  my_mse <- append(my_mse, tmp)
}

png(paste(path, "train.png"))
plot(x = f$lambda, y = my_mse, pch = 19, col = "darkgreen", lwd = 4, main = "Тренировочная
выборка")
dev.off()
```

```

x <- longley_test[-6]
y <- longley_test$Employed

my_mse <- vector()

for (c in 1:ncol(f$coef)) {
  coefs <- f$coef[, c]
  tmp <- vector()
  for (s in 1:ncol(x)) tmp <- append(tmp, mse(y[s], sum(coefs*x[s, ])))
  tmp <- mean(tmp)
  my_mse <- append(my_mse, tmp)
}

png(paste(path, "test.png"))
plot(x = f$lambda, y = my_mse, pch = 19, col = "darkred", lwd = 4, main = "Тестовая выборка")
dev.off()

```

```
# Задание 5-----
data("EuStockMarkets")
library(mltools)

dax <- EuStockMarkets[, 1]
smi <- EuStockMarkets[, 2]
cac <- EuStockMarkets[, 3]
ftse <- EuStockMarkets[, 4]

png(paste(path, "EuStockMarket.png"))
#plot(EuStockMarkets)
plot(EuStockMarkets[,1], type = "l", col="red",
      xlab="Год", ylab="Котировка")
lines(EuStockMarkets[,2], type = "l", col="blue")
lines(EuStockMarkets[,3], type = "l", col="green")
lines(EuStockMarkets[,4], type = "l", col="black")
dev.off()

name <- time(EuStockMarkets) # названия строк

all_mse <- vector()
rss <- vector()

# -----

f <- lm(DAX ~ ., data.frame(time = name, DAX = dax))
png(paste(path, "DAX.png"))
plot(dax, main = "Deutscher Aktienindex (DAX)")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
                f$coefficients[1]), col = "red")
dev.off()

dax_mse <- mse(dax, name*f$coefficients[2] + f$coefficients[1])
all_mse <- append(all_mse, dax_mse)
rss <- append(rss, RSS(f))

# -----

f <- lm(SMI ~ ., data = data.frame(time = name, SMI = smi))
png(paste(path, "SMI.png"))
plot(smi, main = "Swiss Market Index (SMI)")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
                f$coefficients[1]), col = "red")
dev.off()

smi_mse <- mse(smi, name*f$coefficients[2] + f$coefficients[1])
all_mse <- append(all_mse, smi_mse)
rss <- append(rss, RSS(f))
```

```

# -----

f <- lm(CAC ~ ., data = data.frame(time = name, CAC = cac))
png(paste(path, "CAC.png"))
plot(cac, main = "Cotation Assistée en Continu (CAC)")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
                f$coefficients[1]), col = "red")
dev.off()
cac_mse <- mse(cac, name*f$coefficients[2] + f$coefficients[1])
all_mse <- append(all_mse, cac_mse)
rss <- append(rss, RSS(f))

#-----

f <- lm(FTSE ~ ., data = data.frame(time = name, FTSE = ftse))
png(paste(path, "FTSE.png"))
plot(ftse, main = "Financial Times Stock Exchange (FTSE)")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
                f$coefficients[1]), col = "red")
dev.off()
ftse_mse <- mse(ftse, name*f$coefficients[2] + f$coefficients[1])
all_mse <- append(all_mse, ftse_mse)
rss <- append(rss, RSS(f))

#-----

overall = (dax + smi + cac + ftse) / 4
f <- lm(overall ~ ., data = data.frame(time = name, overall = overall))
png(paste(path, "Overall.png"))
plot(overall, main = "Overall")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
                f$coefficients[1]), col = "red")
dev.off()
overall_mse <- mse(overall, name*f$coefficients[2] + f$coefficients[1])
all_mse <- append(all_mse, overall_mse)
rss <- append(rss, RSS(f))

```

```
# Задание 6-----
data(JohnsonJohnson)

png(paste(path, "JohnsonJohnson.png"))
plot(time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
     JohnsonJohnson[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
     type = "l", col="red", main = "Изменение прибыли во времени", xlab="Год",
     ylab="Прибыль")
lines(time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
     JohnsonJohnson[seq(from = 2, to = length(JohnsonJohnson), by = 4)],
     type = "l", col="blue")
lines(time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
     JohnsonJohnson[seq(from = 3, to = length(JohnsonJohnson), by = 4)],
     type = "l", col="green")
lines(time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
     JohnsonJohnson[seq(from = 4, to = length(JohnsonJohnson), by = 4)],
     type = "l", col="black")
dev.off()

name <- time(JohnsonJohnson)
f <- lm(JohnsonJohnson ~ .,
      data = data.frame(time = name,
                        JohnsonJohnson = JohnsonJohnson))

png(paste(path, "JohnsonJohnson.png"))
plot(JohnsonJohnson, main = "JohnsonJohnson")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
                f$coefficients[1]), col = "red")
dev.off()

mean(predict.lm(f, data.frame(time = time(ts(start = 2016, end = 2017, frequency = 365))))))

qtr_mean <- vector()

for (i in 1:4)
{
  name <- time(JohnsonJohnson)[seq(i,length(JohnsonJohnson), 4)]
  qtr <- JohnsonJohnson[seq(i, length(JohnsonJohnson), 4)]
  quater <- data.frame(time = name, qtr = qtr)
  f <- lm(qtr ~ ., data = quater)
  png(paste(path, "Quater №", i, ".png", sep = ""))
  plot(quater, main = paste("Quater №", i, sep = ""), pch = 19, lwd = 4, col = "lightgreen")
  lines(xy.coords(x = name, y = name*f$coefficients[2] + f$coefficients[1]), col = "red")
  dev.off()
  qtr_mean <- append(qtr_mean, mean(predict.lm(f, data.frame(time = time(ts(start = c(2016, i),
end = c(2016, i), frequency = 4))))))
}
qtr_mean
```

```
all = lm(JohnsonJohnson[seq(from = 1, to = length(JohnsonJohnson), by = 4)]+
        JohnsonJohnson[seq(from = 2, to = length(JohnsonJohnson), by = 4)]+
        JohnsonJohnson[seq(from = 3, to = length(JohnsonJohnson), by = 4)]+
        JohnsonJohnson[seq(from = 4, to = length(JohnsonJohnson), by = 4)]~
        time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
        JohnsonJohnson)
all
```

```
# Задание 7 -----  
library(qpcR)  
data(sunspot.year)  
  
name <- time(sunspot.year)  
f <- lm(sun ~ ., data.frame(time = name, sun = sunspot.year))  
png(paste(path, "Sunspot.png"))  
plot(sunspot.year, main = "Sunspot year", col = "brown")  
lines(xy.coords(x = name, y = name*f$coefficients[2] +  
                f$coefficients[1]), col = "green")  
dev.off()  
RSS(f)
```

```
# Задание 8 -----
gas <- read.csv(paste(path, "UKgas.csv", sep = ""), sep = ',', row.names = 1)

name = gas$time
f <- lm(UKgas ~ ., data = gas)
png(paste(path, "UKgas.png"))
plot(gas, main = "Ukgas", pch = 19, col = "lightgreen")
lines(xy.coords(x = name, y = name*f$coefficients[2] +
               f$coefficients[1]), col = "red", lwd = 2)
dev.off()
summary(f)

mean(predict.lm(f, data.frame(time = time(ts(start = 2016, end = 2017, frequency = 365))))))

qtr_mean <- vector()

for (i in 1:4)
{
  name <- gas$time[seq(i, length(gas$time), 4)]
  qtr <- gas$UKgas[seq(i, length(gas$UKgas), 4)]
  quater <- data.frame(time = name, qtr = qtr)
  f <- lm(qtr ~ ., data = quater)
  png(paste(path, "Quater №", i, "(Ukgas).png", sep = ""))
  plot(quater, main = paste("Quater №", i, "(Ukgas)", sep = ""), pch = 19, lwd = 4, col =
"lightgreen")
  lines(xy.coords(x = name, y = name*f$coefficients[2] + f$coefficients[1]), col = "red")
  dev.off()
  qtr_mean <- append(qtr_mean, mean(predict.lm(f, data.frame(time = time(ts(start = c(2016, i),
end = c(2016, i), frequency = 4))))))
}
qtr_mean
```



```
# Задание 9 -----  
data(cars)  
f <- lm(dist ~ ., data = cars)  
png(paste(path, "cars.png"))  
plot(cars, main = "Cars", pch = 19, lwd = 2, col = "lightgreen")  
lines(x = cars$speed,  
      y = cars$speed*f$coefficients[2] +  
        f$coefficients[1], col = "red")  
dev.off()  
predict.lm(f, data.frame(speed = 40))
```