

Министерство образования и науки РФ
Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 5
«Регрессия»
по дисциплине «Машинное обучение»

Выполнила:

студентка гр. 3540201/20301

_____ Климова О. А.

подпись, дата

Проверил:

д.т.н., проф.

_____ Уткин Л. В.

подпись, дата

Санкт-Петербург

2022

Содержание

Постановка задачи.....	3
1 Задание 1.....	5
2 Задание 2.....	6
3 Задание 3.....	7
4 Задание 4.....	8
5 Задание 5.....	9
6 Задание 6.....	10
7 Задание 7.....	11
8 Задание 8.....	12
9 Задание 9.....	13
Приложение 1. Код для задания 1	14
Приложение 2. Код для задания 2	15
Приложение 3. Код для задания 3	16
Приложение 4. Код для задания 4	17
Приложение 5. Код для задания 5	19
Приложение 6. Код для задания 6	20
Приложение 7. Код для задания 7	22
Приложение 8. Код для задания 8	23
Приложение 9. Код для задания 9	24

Постановка задачи

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 выбрать подмножество признаков мощности, 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 , 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.

1 Задание 1

В данном задании были загружены данные из файла reglab1.txt. Используя функцию `lm`, была построена регрессия с использованием трех моделей:

$$z = b_0 + b_1x + b_2y = 0.0175 + 4.0506x + 4.95y$$

$$x = b_0 + b_1z + b_2y = 0.03608 + 0.22636z - 1.11505y$$

$$y = b_0 + b_1z + b_2x = 0.01816 + 0.19251z - 0.77603x$$

Среднеквадратичная ошибка для данного обучающего множества при 80% данных для обучения: ≈ 0.36 ; ≈ 1.61 ; ≈ 1.56 . Таким образом, более точной моделью является первая, которая была построена на зависимости параметра z от x и y .

Код представлен в Приложении 1.

2 Задание 2

Был реализован алгоритм для уменьшения количества признаков, используемых для построения регрессии. Для данных из файла reglab2.txt было получено:

Формула	Максимальная остаточная сумма квадратов
$y \sim 3.622$ (пустое под-во пр-в)	10.361126840
$y \sim x_1$	2.638007093
$y \sim x_2$	4.052606748
$y \sim x_3$	10.487300603
$y \sim x_4$	10.164198855
$y \sim x_1 + x_2$	0.017042363
$y \sim x_1 + x_3$	2.701446563
$y \sim x_1 + x_4$	2.662181527
$y \sim x_2 + x_3$	4.053856866
$y \sim x_2 + x_4$	4.486667219
$y \sim x_3 + x_4$	10.329880691
$y \sim x_1 + x_2 + x_3$	0.011771231
$y \sim x_1 + x_2 + x_4$	0.009143424
$y \sim x_1 + x_3 + x_4$	2.748087659
$y \sim x_2 + x_3 + x_4$	4.473704655
$y \sim x_1 + x_2 + x_3 + x_4$	0.006607641

При пустом подмножестве параметров было использовано среднее значение y . Для генерации всех возможных сочетаний по m элементов из некоторого множества x была использована функция `combn(x, m, ...)`.

По полученным данным можно сделать вывод о том, что оптимальным подмножеством параметров является: $\{x_1; x_2; x_3; x_4\}$.

Код представлен в Приложении 2.

3 Задание 3

Были загружены данные из файла cugage.txt и построена регрессия, выражающая зависимость возраста исследуемых отложений от глубины залегания, используя веса наблюдений:

$$\text{calAge} = 334.5 + 22.34 * \text{Depth}$$

Оценим качество построенной модели с помощью вычисления среднеквадратичного отклонения:

$$\delta = \frac{\sum (x_i - \bar{x})^2}{n}$$

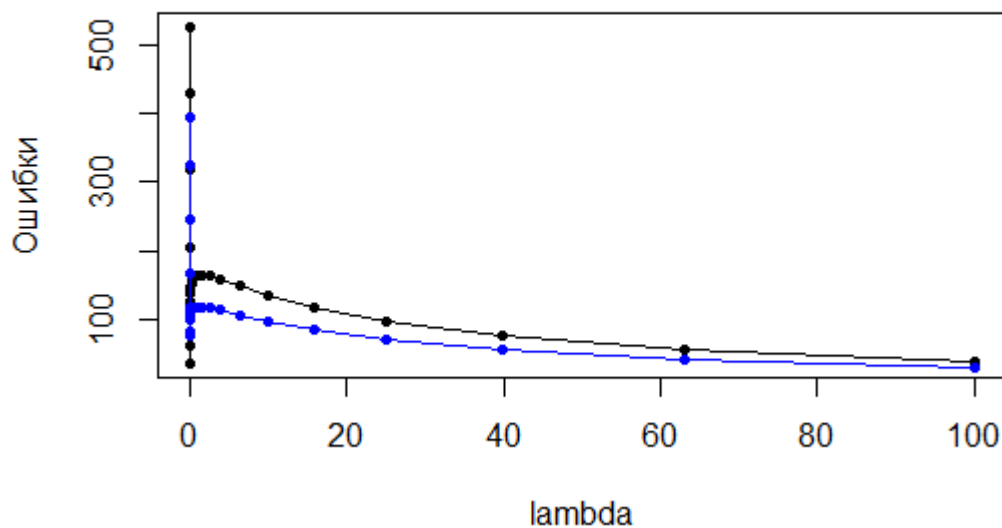
$$\text{MSE} = 124641.7$$

Код представлен в Приложении 3.

4 Задание 4

Были загружены данные Longley (макроэкономические данные), которые состоят из 7 экономических переменных, наблюдаемых с 1947 по 1962 годы. Из этих данных была исключена переменная "Population". Данные были разделены на тестовую и обучающую выборки равных размеров случайным образом. Была построена гребневая регрессия и зависимость ошибок (среднеквадратичных) на тесте (синий график) и на обучающей выборке (черный график) от лямбды:

Зависимость ошибок от lambda



Можно видеть, что ошибка на обучающей выборке больше, чем на тестовой и что обе ошибки убывают с увеличением лямбда.

Код представлен в Приложении 4.

5 Задание 5

Были загружены данные EuStockMarkets из пакета «datasets». Данные содержат ежедневные котировки на момент закрытия фондовых бирж: Germany DAX (Ibis) (красный), Switzerland SMI (синий), France CAC (черный), и UK FTSE (коричневый).

Были построены на одном графике все кривые изменения котировок во времени:



Была построена линейная регрессия для каждой модели в отдельности и для всех моделей вместе:

$$\text{DAX} = -894557.9 + 449.7 * \text{Time}$$

$$\text{SMI} = -1428160.2 + 717.5 * \text{Time}$$

$$\text{CAC} = -405915.3 + 204.6 * \text{Time}$$

$$\text{FTSE} = -865200.4 + 435.5 * \text{Time}$$

$$\text{DAX} + \text{SMI} + \text{CAC} + \text{FTSE} = -3593834 + 1807 * \text{Time}$$

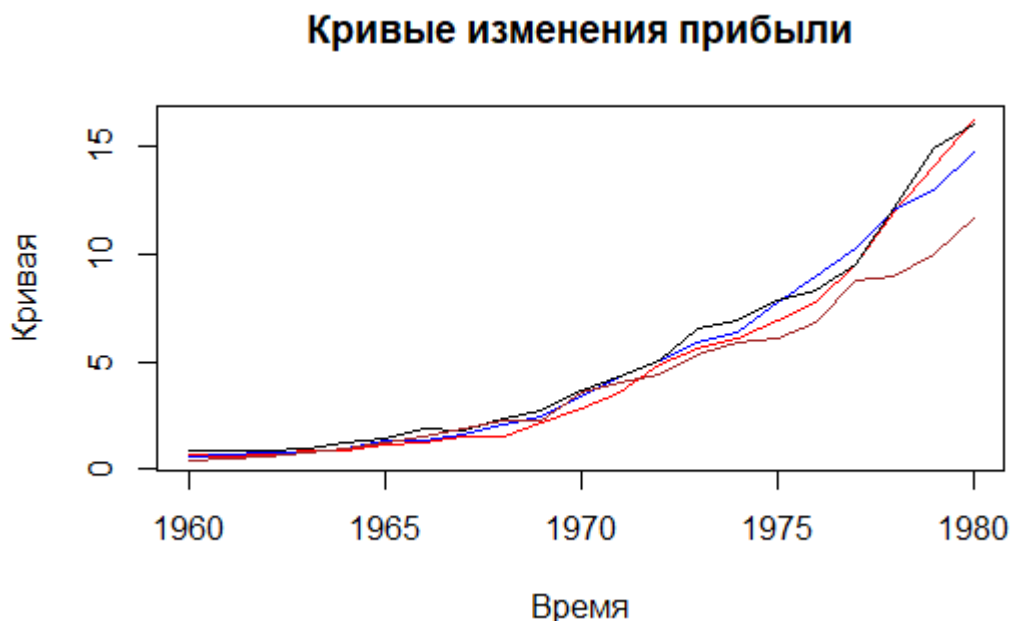
Наибольшую динамику имеет биржа Switzerland SMI, так как синий график стремительнее всех поднимается вверх, что обусловлено самым высоким коэффициентом при $\text{Time} = 717.5$.

Код представлен в Приложении 5.

6 Задание 6

Были загружены данные JohnsonJohnson из пакета «datasets», содержащие поквартальную прибыль компании Johnson & Johnson с 1960 по 1980 гг.

Были построены все кривые изменения прибыли во времени на одном графике (красный – первый квартал; синий – второй; черный – третий; коричневый – четвертый):



Была построена линейная регрессия для каждого квартала в отдельности и для всех кварталов вместе:

$$\text{Qtr1} = -1364.282 + 0.695 * \text{Time}$$

$$\text{Qtr2} = -1345.0727 + 0.6853 * \text{Time}$$

$$\text{Qtr3} = -1382.3170 + 0.7044 * \text{Time}$$

$$\text{Qtr4} = -1049.5828 + 0.5349 * \text{Time}$$

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

По полученным моделям можно сделать вывод, что в 3-м квартале самая большая динамика прибыли, а в четвертом самая маленькая.

Прогноз по прибыли в 2016 году в каждом из кварталов:

$$\text{Qtr1} = 36.75964; \text{Qtr2} = 36.48945; \text{Qtr3} = 37.65394; \text{Qtr4} = 28.79391$$

В среднем по году: 34.92424

Код представлен в Приложении 6.

7 Задание 7

Были загружены данные sunspot.year из пакета «datasets». Данные содержат количество солнечных пятен с 1700 по 1988 гг.

Был построен график изменения числа солнечных пятен во времени:



Была построена линейная регрессия:

$$\text{Spots} = -130.42119 + 0.09709 * \text{Year}$$

Код представлен в приложении 7.

8 Задание 8

Были загружены данные из файла пакета «UKgas.scv». Данные содержат объемы ежеквартально потребляемого газа в Великобритании с 1960 по 1986 гг.

Была построена линейная регрессия для каждого квартала в отдельности и для всех кварталов вместе:

$$\text{Qtr1} = -78854.23 + 40.22 * \text{Time}$$

$$\text{Qtr2} = -35297.23 + 18.04 * \text{Time}$$

$$\text{Qtr3} = -15403.73 + 7.89 * \text{Time}$$

$$\text{Qtr4} = -59112.72 + 30.14 * \text{Time}$$

$$\text{Qtr1} + \text{Qtr2} + \text{Qtr3} + \text{Qtr4} = -188636.85 + 96.29 * \text{Time}$$

За счет коэффициента перед Time можно видеть, что наибольшая динамика доходности в первом квартале (40.22), а наименьшая в третьем (7.89).

Прогноз по потреблению газа в 2016 году во всех кварталах:

$$\text{Qtr1} = 2230.936; \text{Qtr2} = 1072.375; \text{Qtr3} = 501.9919; \text{Qtr4} = 1654.785$$

В среднем по году: 1372.787

Код представлен в Приложении 8.

9 Задание 9

Были загружены данные cars из пакета «datasets». Данные содержат информацию о различных видах автомобилей:

	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10

Была построена регрессионная модель:

$$\text{Dist} = -17.579 + 3.932 * \text{Speed}$$

С использованием полученной модели был определен тормозной путь при скорости 40 миль в час: 139.7173

Код представлен в Приложении 9.

Приложение 1. Код для задания 1

```
#ЗАДАНИЕ 1
library(datasets)
reglab1 <- read.table("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/reglab1.txt", sep
= "\t", header = TRUE)
n <- dim(reglab1)[1]
reglab1_rand <- reglab1[ order(runif(n)),]
nt <- as.integer(n*0.8)
reglab1_train <- reglab1_rand[1:nt, ]
reglab1_test <- reglab1_rand[(nt+1):n, ]
#регрессия с использованием 1-й модели
f1 = lm(z ~ ., reglab1_train)
pred1 <- predict(f1, reglab1_test)
#ошибка 1
mist1 = sd(reglab1_test$z-pred1)
f1
mist1
#регрессия с использованием 2-й модели
f2 = lm(x ~ ., reglab1_train)
pred2 <- predict(f2, reglab1_test)
#ошибка 2
mist2 = sd(reglab1_test$z-pred2)
f2
mist2
#регрессия с использованием 3-й модели
f3 = lm(y ~ ., reglab1_train)
pred3 <- predict(f3, reglab1_test)
#ошибка
mist3 = sd(reglab1_test$z-pred3)
f3
mist3
```

Приложение 2. Код для задания 2

#ЗАДАНИЕ 2

```
library(datasets)
```

```
reglab2 <- read.table("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/reglab2.txt", sep  
= "\t", header = TRUE)
```

```
n <- dim(reglab2)[1]
```

```
reglab2_rand <- reglab2[ order(runif(n)),]
```

```
#80% для обучения
```

```
nt <- as.integer(n*0.8)
```

```
reglab2_train <- reglab2_rand[1:nt, ]
```

```
reglab2_test <- reglab2_rand[(nt+1):n, ]
```

```
x <- c("x1", "x2", "x3", "x4")
```

```
res <- list(x = NULL, rss = NULL)
```

```
#среднее значение y по обучающей выборке записываем в вектор предсказаний
```

```
pred <- rep(summary(reglab2_train$y)[4],n-nt)
```

```
#значение максимальной остаточной суммы квадратов при пустом подмножестве
```

```
rss = summary((reglab2_test$y-pred)^2)[6]
```

```
res$x <- append(res$x,summary(reglab2_train$y)[4])
```

```
res$rss <- append(res$rss, rss)
```

```
for (k in 1:4) {
```

```
  combs <- combn(x, k)
```

```
  combs
```

```
  for (i in 1:dim(combs)[2]) {
```

```
    f <- lm(as.formula(paste("y", paste(combs[,i], collapse = " + "),  
                                sep = " ~ ")),reglab2_train)
```

```
    pred <- predict(f, reglab2_test)
```

```
    #остаточная сумма квадратов
```

```
    rss = summary((reglab2_test$y-pred)^2)[6]
```

```
    res$x <- append(res$x, paste(combs[,i], collapse = " "))
```

```
    res$rss <- append(res$rss, rss)
```

```
  }
```

```
}
```

```
res
```

Приложение 3. Код для задания 3

```
#ЗАДАНИЕ 3
library(datasets)
cygage <- read.table("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/cygage.txt", sep =
"\t", header = TRUE)
n <- dim(cygage)[1]
cygage_rand <- cygage[ order(runif(n)),]
#разделим множество на обучающее и тестовое (80% для обучения)
nt <- as.integer(n*0.8)
cygage_train <- cygage_rand[1:nt, ]
cygage_test <- cygage_rand[(nt+1):n, ]
f = lm(calAge ~ Depth, cygage_train, weights=cygage_train$Weight)
f
pred <- predict(f, cygage_test)
f_summary <- summary(f)
MSE = mean(f_summary$residuals^2)
MSE
```


Приложение 4. Код для задания 4

```
#ЗАДАНИЕ 4
library(datasets)
library(dplyr)
library(MASS)
library(glmnet)
#загрузка данных
data(longley)
#исключение переменной "population"
longley <- longley %>% select(-Population)
n <- dim(longley)[1]
longley_rand <- longley[ order(runif(n)),]
#разделяем датасет на обучающую и тестовую выборки по 50%
nt <- as.integer(n*0.5)
longley_train <- longley_rand[1:nt, ]
longley_test <- longley_rand[(nt+1):n, ]
#записываем значения лямбд
lambdas = c()
for(i in 0:25){
  lambdas <- append(lambdas, 10^(-3+0.2*i))
}

#вектора ошибок
train_mistake = c()
test_mistake = c()

#предсказание
pred <- function(f, longley) {
  result <- NULL
  for (i in 1:dim(longley)[1]) {
    sum <- 0
    for (k in names(f))
      if (k == "(Intercept)" | k == "") sum <- sum + as.numeric(f[k])
      else sum <- sum + longley[i, k] * as.numeric(f[k])
    result <- c(result, sum)
  }
  return(result)
}

#строим гребневую регрессию
f <- lm.ridge(Employed ~ ., longley_train, lambda = lambdas)
for(i in 1:length(lambdas)){
  #ошибки на обучающей выборке и тесте
  train_mistake <- append(train_mistake, sd(longley_train$Employed -
pred(f$coef[i],longley_train)))
}
```

```

test_mistake <- append(test_mistake, sd(longley_test$Employed -
pred(f$coef[,i],longley_test)))
}
#строим графики
plot(lambdas, train_mistake, pch=20, type = "o", col="black",
      main = "Зависимость ошибок от lambda",
      xlab="lambda", ylab="Ошибки")
lines(lambdas, test_mistake, pch=20, type = "o", col="blue")

```

Приложение 5. Код для задания 5

#ЗАДАНИЕ 5

```
library(datasets)
```

```
data(EuStockMarkets)
```

```
#построение котировок на одном графике
```

```
plot(EuStockMarkets[,1], type = "l",
```

```
      main = "Кривые изменения котировок во времени",
```

```
      xlab="Время", ylab="Котировка", col="red")
```

```
lines(EuStockMarkets[,2], type = "l", col="blue")
```

```
lines(EuStockMarkets[,3], type = "l", col="black")
```

```
lines(EuStockMarkets[,4], type = "l", col="brown")
```

```
#построение регрессий по-отдельности
```

```
DAX = lm(EuStockMarkets[,1] ~ time(EuStockMarkets), EuStockMarkets)
```

```
SMI = lm(EuStockMarkets[,2] ~ time(EuStockMarkets), EuStockMarkets)
```

```
CAC = lm(EuStockMarkets[,3] ~ time(EuStockMarkets), EuStockMarkets)
```

```
FTSE = lm(EuStockMarkets[,4] ~ time(EuStockMarkets), EuStockMarkets)
```

```
#построение регрессии для всех котировок
```

```
all =
```

```
lm(EuStockMarkets[,1]+EuStockMarkets[,2]+EuStockMarkets[,3]+EuStockMarkets[,4]~time(EuStockMarkets),
```

```
    EuStockMarkets)
```

```
#вывод коэффициентов
```

```
DAX
```

```
SMI
```

```
CAC
```

```
FTSE
```

```
all
```

Приложение 6. Код для задания 6

#ЗАДАНИЕ 6

```
library(datasets)
```

```
data(JohnsonJohnson)
```

```
length(JohnsonJohnson)
```

```
time(JohnsonJohnson)
```

```
#построение графиков
```

```
plot(time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
```

```
      JohnsonJohnson[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
```

```
      type = "l", main = "Кривые изменения прибыли", xlab="Время",
```

```
      ylab="Кривая", col="red")
```

```
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="black")
```

```
lines(time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson), by = 4)],
```

```
      JohnsonJohnson[seq(from = 4, to = length(JohnsonJohnson), by = 4)],
```

```
      type = "l", col="brown")
```

```
#построение моделей
```

```
qtr1 = lm(JohnsonJohnson[seq(from = 1, to = length(JohnsonJohnson), by = 4)] ~  
          time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson),  
                                     by = 4)], JohnsonJohnson)
```

```
qtr2 = lm(JohnsonJohnson[seq(from = 2, to = length(JohnsonJohnson), by = 4)] ~  
          time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson),  
                                     by = 4)], JohnsonJohnson)
```

```
qtr3 = lm(JohnsonJohnson[seq(from = 3, to = length(JohnsonJohnson), by = 4)] ~  
          time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson),  
                                     by = 4)], JohnsonJohnson)
```

```
qtr4 = lm(JohnsonJohnson[seq(from = 4, to = length(JohnsonJohnson), by = 4)] ~  
          time(JohnsonJohnson)[seq(from = 1, to = length(JohnsonJohnson),  
                                     by = 4)], JohnsonJohnson)
```

```
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)
```

```
#вывод параметров моделей:
```

```
qtr1
```

```
qtr2
```

```
qtr3
```

```
qtr4
```

```
all
```

```
#предсказание на 2016 год
year=2016
pred1 = coef(qtr1)[1]+coef(qtr1)[2]*year
pred2 = coef(qtr2)[1]+coef(qtr2)[2]*year
pred3 = coef(qtr3)[1]+coef(qtr3)[2]*year
pred4 = coef(qtr4)[1]+coef(qtr4)[2]*year
#прогноз на год в среднем
pred_all = (coef(all)[1]+coef(all)[2]*year)/4
pred1
pred2
pred3
pred4
pred_all
```

Приложение 7. Код для задания 7

```
#ЗАДАНИЕ 7
library(datasets)
data(sunspot.year)
sunspot.year
#построим таблицу на полученных данных (где будут года и пятна)
sunspot = data.frame(year = seq(1700,1988,1),
                      spots=sunspot.year[seq(1,length(sunspot.year),1)])
plot(sunspot$year, sunspot$spots, type = "l", col="blue",
     main = "Изменение числа солнечных пятен", xlab="Время",
     ylab="Число пятен")
#линейная регрессия
f = lm(spots ~ year, sunspot)
f
```

Приложение 8. Код для задания 8

#ЗАДАНИЕ 8

```
library(datasets)
```

```
gas <- read.csv("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/UKgas.csv", header =  
TRUE, sep = ",", dec = ".",  
stringsAsFactors = FALSE)
```

#построение регрессии

```
qtr1 = lm(UKgas[seq(from = 1, to = dim(gas)[1], by = 4)] ~  
time[seq(from = 1, to = dim(gas)[1], by = 4)], gas)
```

```
qtr2 = lm(UKgas[seq(from = 2, to = dim(gas)[1], by = 4)] ~  
time[seq(from = 2, to = dim(gas)[1], by = 4)], gas)
```

```
qtr3 = lm(UKgas[seq(from = 3, to = dim(gas)[1], by = 4)] ~  
time[seq(from = 3, to = dim(gas)[1], by = 4)], gas)
```

```
qtr4 = lm(UKgas[seq(from = 4, to = dim(gas)[1], by = 4)] ~  
time[seq(from = 4, to = dim(gas)[1], by = 4)], gas)
```

#по всем

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

#вывод параметров моделей

```
qtr1
```

```
qtr2
```

```
qtr3
```

```
qtr4
```

```
all
```

#предсказание по кварталам

```
coef(qtr1)[1]+coef(qtr1)[2]*year
```

```
coef(qtr2)[1]+coef(qtr2)[2]*year
```

```
coef(qtr3)[1]+coef(qtr3)[2]*year
```

```
coef(qtr4)[1]+coef(qtr4)[2]*year
```

#в среднем по году

```
(coef(all)[1]+coef(all)[2]*year)/4
```

Приложение 9. Код для задания 9

```
#ЗАДАНИЕ 9
library(datasets)
data(cars)
cars
#регрессионная модель
f = lm(dist ~ speed, cars)
f
#предсказание
predict(f, data.frame(speed=40,dist=0))
```