Лабораторная работа №6 «Регрессия»

Работу выполнила студентка группы 5140201/30301 Фазылова Алика

Задание 1.

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

Решение

Построим три регрессионных модели в зависимости от разных параметров z(x, y), x(y, z) и y(x, z) (рисунки 1-3).

```
Call:
lm(formula = z ~ ., data = data1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.97246 -0.16759  0.01308  0.20537  0.81127 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.02163    0.06384  -0.339    0.735    
x            4.10248    0.08698  47.168   <2e-16 ***
y            4.94308    0.08035  61.517   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3376 on 197 degrees of freedom
Multiple R-squared:  0.9686,	Adjusted R-squared:  0.9683 
F-statistic:  3041 on 2 and 197 DF,  p-value: < 2.2e-16
```

Рисунок 1 – Регрессионная модель 1 z(x, y)

```
Call:
lm(formula = x ~ ., data = data1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.21585 -0.04566 -0.00422  0.04407  0.25627 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.045460   0.014564   3.121  0.00207 ** 
z            0.223927   0.004747  47.168  < 2e-16 ***
y           -1.105949   0.030227 -36.588  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.07888 on 197 degrees of freedom
Multiple R-squared:  0.9187,	Adjusted R-squared:  0.9178 
F-statistic:  1113 on 2 and 197 DF,  p-value: < 2.2e-16
```

Рисунок 2 – Регрессионная модель 2 x(y, z)

```
Call:
lm(formula = y ~ ., data = data1)

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 3 – Регрессионная модель 3 y(x, z).

Оценим модели по метрикам RSE и R-squared.

Residual Standard Error (RSE) - представляет собой меру разброса остатков в регрессионной модели. Остатки — это разница между фактическими значениями зависимой переменной и значениями, предсказанными моделью. Более низкое значение RSE указывает на то, что остатки имеют меньший разброс относительно предсказанных значений, что в свою очередь говорит о лучшей адаптации модели к данным.

Самым низким RSE обладает модель 3 y(x,z) 0.06659, далее идет модель 2 x(y,z) 0.0788, и затем модель 1 z(x,y) 0.3376.

R-квадрат (коэффициент детерминации) показывает, насколько условная дисперсия модели отличается от дисперсии реальных значений. Если этот коэффициент близок к 1, то условная дисперсия модели достаточно мала и вероятно, что модель хорошо описывает данные.

Самым большим R-squared обладает модель 1 z(x,y) 0.9686, далее с незначительным отставанием идет модель 3 y(x,z) 0.9505 , и затем модель 2 x(y,z) 0.9187.

Таким образом модель 3 y(x,z) является наиболее оптимальной, так как она имеет хорошие показатели по анализируемым метрикам.

Данная модель имеет вид: $Y=0.02861+0.12229 Z-0.78821 X$

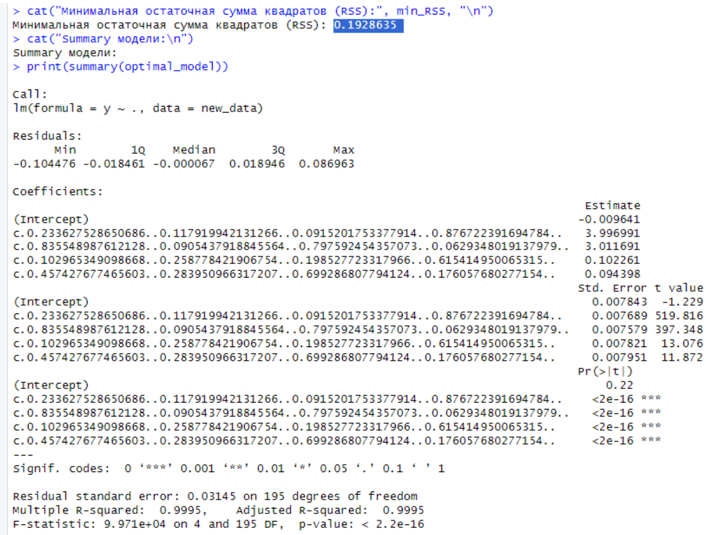
Листинг кода 1 задачи:

```
data1 <- read.table( "reglab1.txt", sep = "\t", header = TRUE)

model1 <- lm(z ~ ., data = data1)
model2 <- lm(x ~ ., data = data1)
model3 <- lm(y ~ ., data = data1)

summary(model1)
summary(model2)
summary(model3)
```

Задание 2.

Реализуйте следующий алгоритм для уменьшения количества признаков, используемых для построения регрессии: для каждого $k \in \{0,1,\ldots,d\}$ выбрать подмножество признаков мощности k^1, минимизирующее остаточную сумму квадратов *RSS*. Используя полученный алгоритм, выберите оптимальное подможество 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, ...).

Выполним программу получив следующие результаты (рисунок)



```
> cat("Минимальная остаточная сумма квадратов (RSS):", min_RSS, "\n")
Минимальная остаточная сумма квадратов (RSS): 0.1928635
> cat("Summary модели:\n")
Summary модели:
> print(summary(optimal_model))

Call:
lm(formula = y ~ ., data = new_data)

Residuals:
      Min        1Q    Median        3Q       Max
-0.104476 -0.018461 -0.000067  0.018946  0.086963

Coefficients:
                                                                              Estimate
(Intercept)                                                                  -0.009641
c.0.233627528650686..0.117919942131266..0.0915201753377914..0.876722391694784..   3.996991
c.0.835548987612128..0.0905437918845564..0.797592454357073..0.0629348019137979..  3.011691
c.0.102965349098668..0.258778421906754..0.198527723317966..0.615414950065315..    0.102261
c.0.457427677465603..0.283950966317207..0.699286807794124..0.176057680277154..    0.094398
                                                                              Std. Error t value
(Intercept)                                                                     0.007843  -1.229
c.0.233627528650686..0.117919942131266..0.0915201753377914..0.876722391694784..   0.007689 519.816
c.0.835548987612128..0.0905437918845564..0.797592454357073..0.0629348019137979..  0.007579 397.348
c.0.102965349098668..0.258778421906754..0.198527723317966..0.615414950065315..    0.007821  13.076
c.0.457427677465603..0.283950966317207..0.699286807794124..0.176057680277154..    0.007951  11.872
                                                                              Pr(>|t|)
(Intercept)                                                                       0.22
c.0.233627528650686..0.117919942131266..0.0915201753377914..0.876722391694784..   <2e-16 ***
c.0.835548987612128..0.0905437918845564..0.797592454357073..0.0629348019137979..  <2e-16 ***
c.0.102965349098668..0.258778421906754..0.198527723317966..0.615414950065315..    <2e-16 ***
c.0.457427677465603..0.283950966317207..0.699286807794124..0.176057680277154..    <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03145 on 195 degrees of freedom
Multiple R-squared:  0.9995,	Adjusted R-squared:  0.9995
F-statistic: 9.971e+04 on 4 and 195 DF,  p-value: < 2.2e-16
```

Рисунок 4 – Результат работы программы

В результате работы программы минимальной остаточной суммой квадратов RSS равной 0.1928635 обладает регрессионная модель со всем множеством параметром, имеющая следующий вид:

$$Y = -0.00964 + 3.99699X1 + 3.011691X2 + 0.10226X3 + 0.094398X4.$$

Листинг кода 2 задачи:

```
data2 <- read.table("reglab2.txt", header = TRUE)
min_RSS <- Inf
d <- ncol(data2) - 1

for (k in 1:d) {
  combination <- combn(data2[, -1], k)
  for (i in 1:nrow(combination)) {
    new_data <- data.frame(y = data2[, 1], combination[i, ])
    f <- lm(y ~ ., new_data)
    RSS <- sum((residuals(f))^2)
    if (RSS < min_RSS) {
      min_RSS <- RSS
      optimal_model <- f
    }
  }
}
cat("Минимальная остаточная сумма квадратов (RSS):", min_RSS, "\n")
cat("Summary модели:\n")
print(summary(optimal_model))
cat("Коэффициенты модели:\n")
print(optimal_model$coefficients)
plot(optimal_model)
```

Задание 3.

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

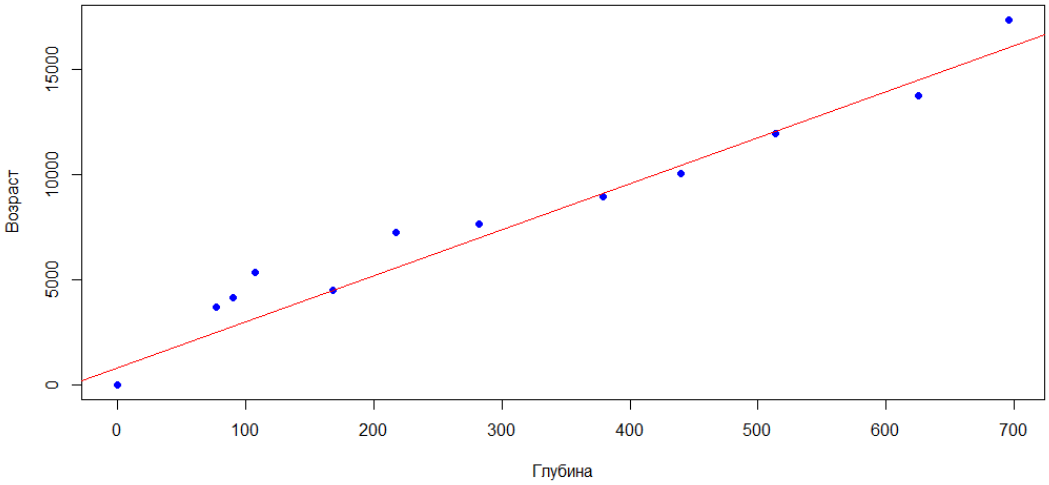


Рисунок 5 – Зависимость возраста исследуемых отложений от глубины залегания

Регрессионная модель имеет вид:

$$calAge=784.570+21.909\times Depth$$

Полученные результаты (рисунок 6)

```
> summary(model)

Call:
lm(formula = calAge ~ Depth, data = data3, weights = Weight)

Weighted Residuals:
   Min     1Q Median     3Q    Max 
-784.6 -137.7  200.8  466.6  702.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  784.570    422.882   1.855   0.0932 .  
Depth         21.909      1.139  19.235 3.14e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 522.1 on 10 degrees of freedom
Multiple R-squared:  0.9737,    Adjusted R-squared:  0.9711 
F-statistic:   370 on 1 and 10 DF,  p-value: 3.141e-09

> summary<-summary(model)
> 
> # Вычисление MSE
> mse <- mean(summary$residuals^2)
> 
> cat("Mean Squared Error (MSE):", mse, "\n")
Mean Squared Error (MSE): 227161.2
> 
```

Рисунок 6 – Результаты анализа

Среднее квадратов разности между фактическими значениями и предсказанными значениями MSE равно 227161.2, что приемлимо для заданных данных. R-squared близок к единице (0.9737), что означает что модель хорошо соответствует данным. Также F-статистика и маленькое p-value 3.141e-09 говорят о статистической значимости модели.

Листинг кода 3 задачи:

```
data3 <- read.table("cygage.txt", stringsAsFactors = TRUE, header = TRUE)
model <- lm(calAge ~ Depth, data = data3, weights = Weight)

summary(model)
summary<-summary(model)

# Вычисление MSE
mse <- mean(summary$residuals^2)

cat("Mean Squared Error (MSE):", mse, "\n")

plot(data3$Depth, data3$calAge, pch = 16, col = "blue", xlab = "Глубина", ylab = "Возраст")
abline(model, col = "red")
```

Задание 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 $\lambda = 10^{-3+0.2\cdot i}, i = 0,\ldots,25$, подсчитайте ошибку на тестовой и обучающей выборке для данных значений λ, постройте графики. Объясните полученные результаты.

В результате работы были сформированы следующие графики зависимости среднеквадратичной ошибкой (Mean Squared Error, MSE) от лямбды для тестовой и тренировочной выборки. (рисунок 7).

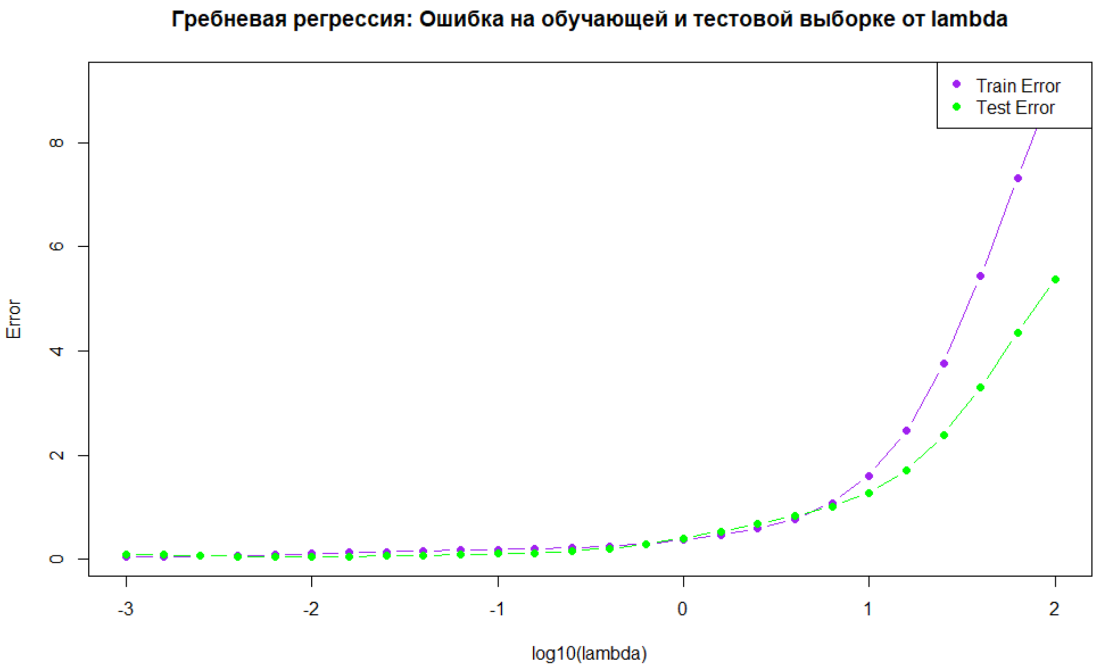


Рисунок 7 – График MSE от значений λ

Проанализировав график, можно сказать, что с увеличением значений λ начиная от нуля ошибка начинает увеличиваться, при этом ошибка на тестовой выборке меньше, чем на тренировочной.

<u>Листинг кода 4 задачи:</u>

```
data(longley)
```

```
model_base <- lm(Employed ~ ., longley)
longley <- subset(longley, select = -c(Population))

set.seed(123)
indices <- sample(1:nrow(longley), nrow(longley) / 2)
train_data <- longley[indices, ]
test_data <- longley[-indices, ]

# Построение гребневой регрессии
lambda_values <- 10^(-3 + 0.2 * (0:25))
ridge_errors_train <- numeric(length(lambda_values))
ridge_errors_test <- numeric(length(lambda_values))

for (i in seq_along(lambda_values)) {
  lambda <- lambda_values[i]
  ridge_model <- lm.ridge(Employed ~ ., train_data, lambda = lambda)
  ridge_coef <- coef(ridge_model)

  # Предсказание на обучающей выборке
  train_pred <- cbind(1, as.matrix(train_data[, -ncol(train_data)])) %*% ridge_coef
  ridge_errors_train[i] <- mean((train_pred - train_data$Employed)^2)

  # Предсказание на тестовой выборке
  test_pred <- cbind(1, as.matrix(test_data[, -ncol(test_data)])) %*% ridge_coef
  ridge_errors_test[i] <- mean((test_pred - test_data$Employed)^2)
}

plot(log10(lambda_values), ridge_errors_train, type = "b",pch = 16, col = "purple", xlab = "log10(lambda)",
   ylab = "Error", main = "Гребневая регрессия: Ошибка на обучающей и тестовой выборке от lambda")
lines(log10(lambda_values), ridge_errors_test, type = "b", pch = 16, col = "green")
legend("topright", legend = c("Train Error", "Test Error"), col = c("purple", "green"), pch = 16)
```

Задание 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.

Построим на одном графике все кривые изменения котировок во времени (рисунок 8)

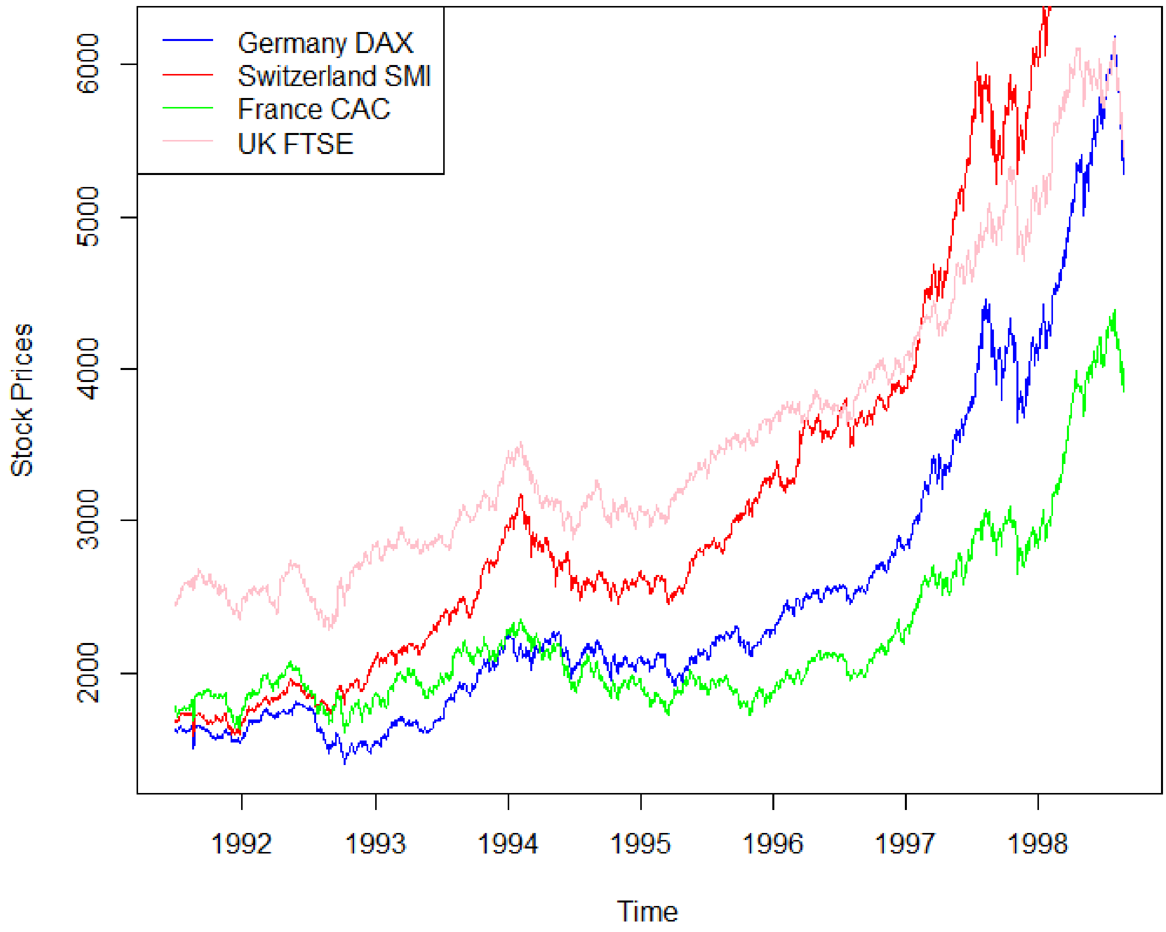


Рисунок 8 – Кривые изменения котировок во времени

Построим линейную регрессию для каждой модели в отдельности (рисунок 9) и для всех вместе (рисунок 10).

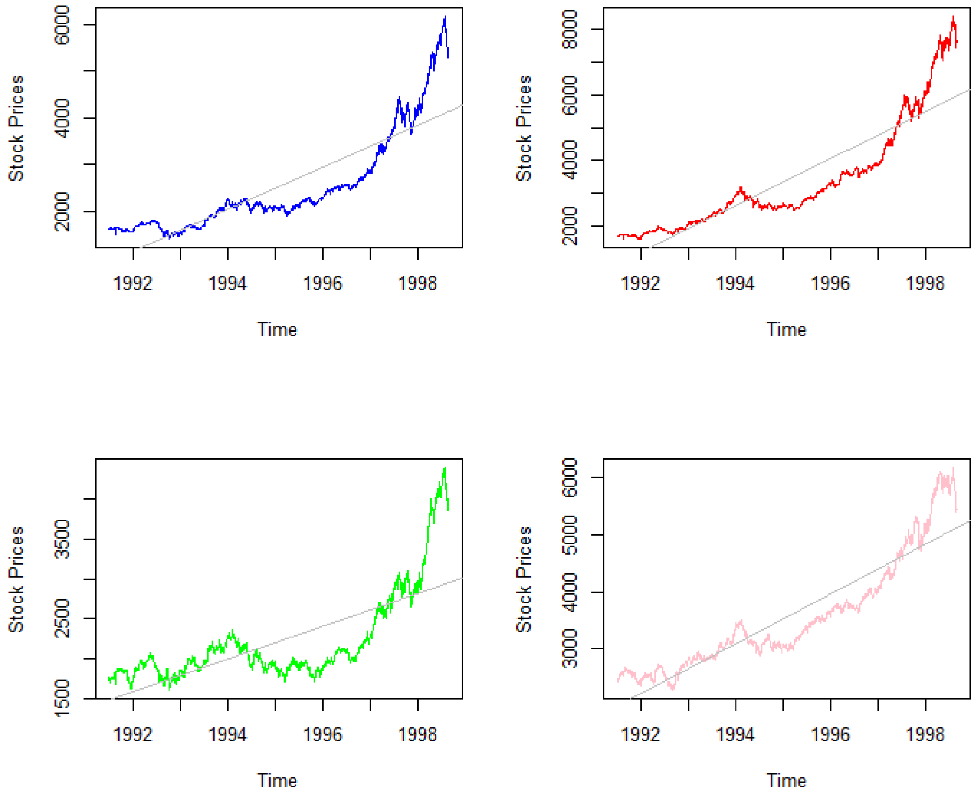


Рисунок 9 – Линейная регрессия для каждой модели

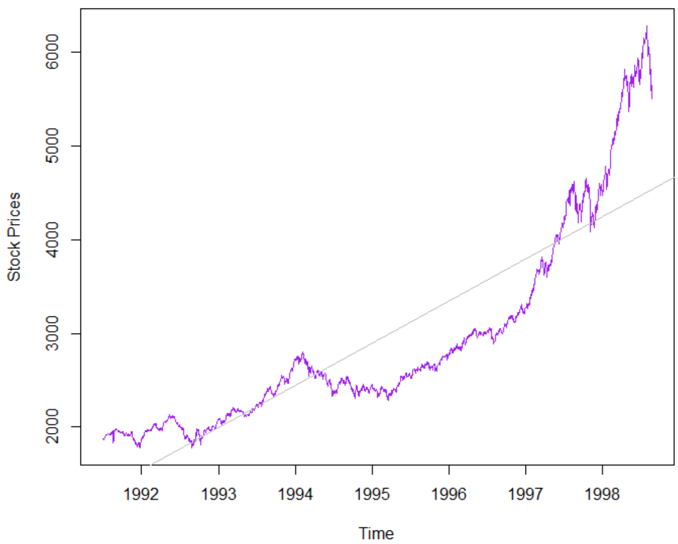


Рисунок 10 – Линейная регрессия для всех моделей

Оценим, какая из бирж имеет наибольшую динамику.

Модель Germany DAX: -8.946e+05 + 449.7 Time

Модель Switzerland SMI: -1.428e+06 + 717.5 Time

Модель France CAC: -4.059e+05 + 204.6 Time

Модель UK FTSE: -8.652e+05 + 435.5 Time

В модели с биржей Switzerland SMI коэффициент у параметра времени (Time) наибольший, значит эта биржа обладает наибольшей динамикой.

<u>Листинг кода 5 задачи:</u>

```
data(EuStockMarkets)

plot(EuStockMarkets[,1], type = "l", col="blue",main = names(EuStockMarkets)[i], xlab = "Time", ylab = "Stock Prices")
lines(EuStockMarkets[,2], type = "l", col="red")
lines(EuStockMarkets[,3], type = "l", col="green")
lines(EuStockMarkets[,4], type = "l", col="pink")
legend("topleft", legend = c("Germany DAX", "Switzerland SMI", "France CAC", "UK FTSE"), col = c("blue", "red", "green", "pink"), lty = 1)

par(mfrow = c(2, 2))
models <- list()

for (i in 1:4) {
  model <- lm(EuStockMarkets[, i] ~ time(EuStockMarkets))
  models[[i]] <- model
  summary(model)

  plot(time(EuStockMarkets), EuStockMarkets[, i], type = "l", col = c("blue", "red", "green", "pink")[i], main = names(EuStockMarkets)[i], xlab = "Time", ylab = "Stock Prices")
  abline(model, col = "gray")
}

par(mfrow = c(1, 1))
```

```
# Линейная регрессии для всех моделей
overall_model <- lm(rowMeans(EuStockMarkets) ~ time(EuStockMarkets))
plot(time(EuStockMarkets), rowMeans(EuStockMarkets), type = "l", col = "purple", xlab = "Time", ylab = "Stock Prices")
abline(overall_model, col = "gray")
```

Задание 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.

Построим на одном графике все кривые изменения прибыли во времени (рисунок 11)

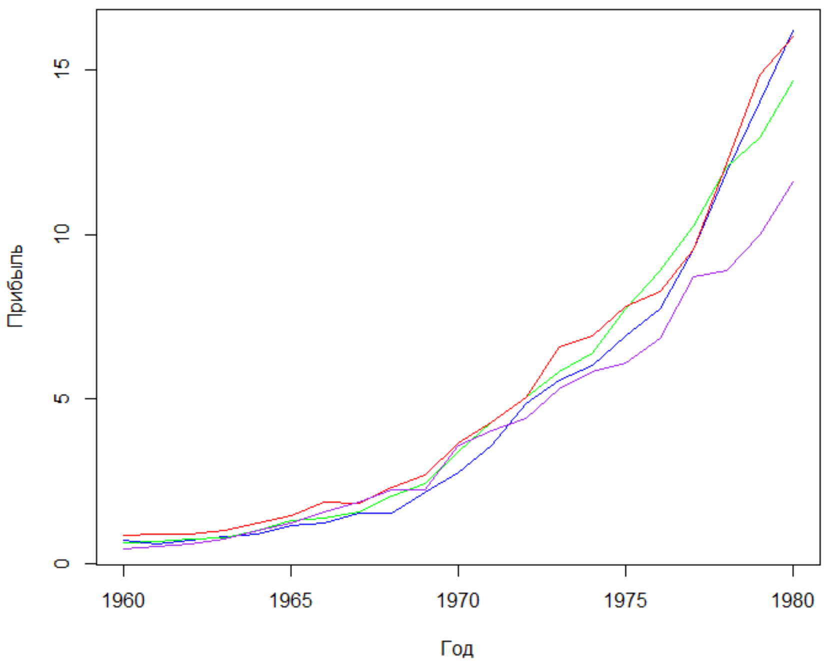


Рисунок 11 – Кривые изменения прибыли во времени

Построим линейную регрессию для каждого квартала в отдельности (рисунок 12)

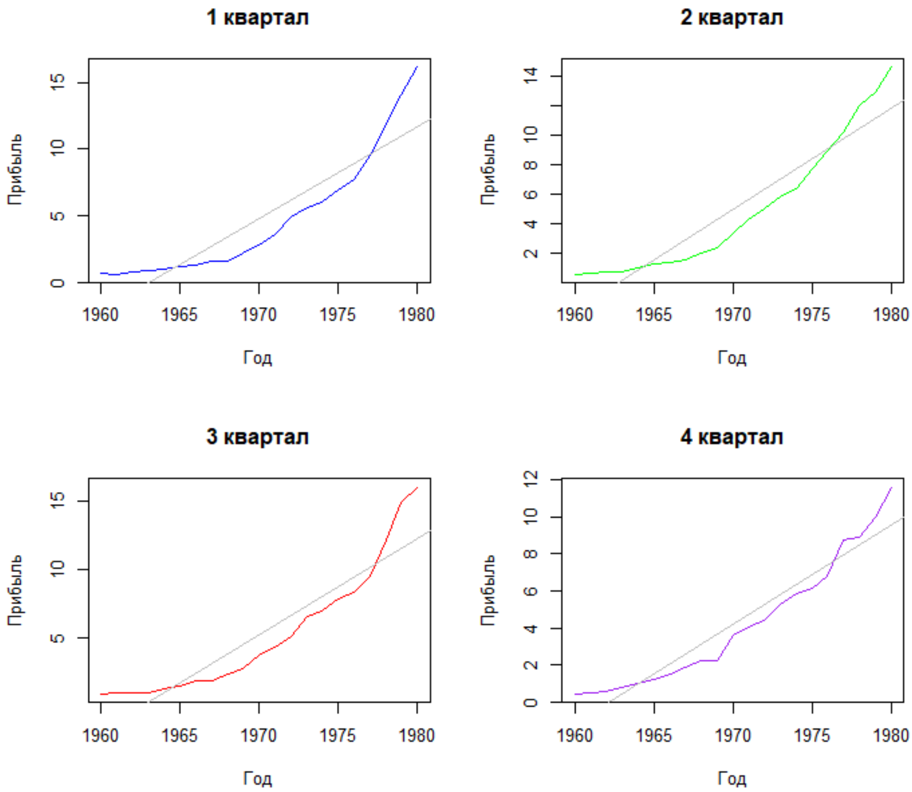


Рисунок 12 – Кривые изменения прибыли во времени

Оценим, какая из моделей имеет наибольшую, а какая наименьшую динамику.

Модель 1-го квартала: -1364.2818+0.695 Time

Модель 2-го квартала: -1345+0.6853 Time

Модель 3-го квартала: -1382+0.7044 Time

Модель 4-го квартала: -1050+0.5349 Time

Модель для всех: -5141.2543+2.6195 Time

В модели квартала 3 коэффициент у параметра времени (Time) наибольший, значит эта модель обладает наибольшей динамикой. Наименьшая динамика наблюдается в 4-ом квартале.

Прогноз по прибыли на 2016 год.

1 квартал: 36.75964

2 квартал: 36.48945

3 квартал: 37.65394

4 квартал: 28.79391

В среднем по году: 34.92424

<u>Листинг кода 6 задачи:</u>

```
library(datasets)
data(JohnsonJohnson)

qtr1 <- JohnsonJohnson[seq(from = 1, to = length(JohnsonJohnson), by = 4)]
qtr2 <- JohnsonJohnson[seq(from = 2, to = length(JohnsonJohnson), by = 4)]
qtr3 <- JohnsonJohnson[seq(from = 3, to = length(JohnsonJohnson), by = 4)]
qtr4 <- JohnsonJohnson[seq(from = 4, to = length(JohnsonJohnson), by = 4)]

JohnsonJohnson.qtr1 <- data.frame(Year = year, Qtr = qtr1)
JohnsonJohnson.qtr2 <- data.frame(Year = year, Qtr = qtr2)
JohnsonJohnson.qtr3 <- data.frame(Year = year, Qtr = qtr3)
JohnsonJohnson.qtr4 <- data.frame(Year = year, Qtr = qtr4)
JohnsonJohnson.year <- data.frame(Year = year, Qtr1 = qtr1, Qtr2 = qtr2, Qtr3 = qtr3, Qtr4 = qtr4)

plot(JohnsonJohnson.qtr1, type = "l", col="blue", xlab="Год", ylab="Прибыль")
lines(JohnsonJohnson.qtr2, col="green")
lines(JohnsonJohnson.qtr3, col="red")
lines(JohnsonJohnson.qtr4, col="purple")

# Функция для вывода графика с регрессией
plot_with_regression <- function(data, col, qtr_name) {
  model <- lm(Qtr ~ Year, data)
  plot(data$Year, data$Qtr, type = "l", col = col, xlab = "Год", ylab = "Прибыль", main = qtr_name)
  abline(model, col = "gray")
  print(summary(model))

  # Предсказание для 2016 года
  prediction <- predict(model, data.frame(Year = 2016))
  cat(qtr_name, "прогноз на 2016 год:", prediction, "\n\n")
}
```

```
par(mfrow = c(2, 2))

plot_with_regression(JohnsonJohnson.qtr1, "blue", "1 квартал")
plot_with_regression(JohnsonJohnson.qtr2, "green", "2 квартал")
plot_with_regression(JohnsonJohnson.qtr3, "red", "3 квартал")
plot_with_regression(JohnsonJohnson.qtr4, "purple", "4 квартал")

par(mfrow = c(1, 1))

model <- lm(Qtr1 + Qtr2 + Qtr3 + Qtr4  ~ Year, JohnsonJohnson.year )
summary(model)
prediction <- predict(model ,data.frame(Year = 2016))/4
prediction
```

Задание 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.

Построим график модели (рисунок 13).

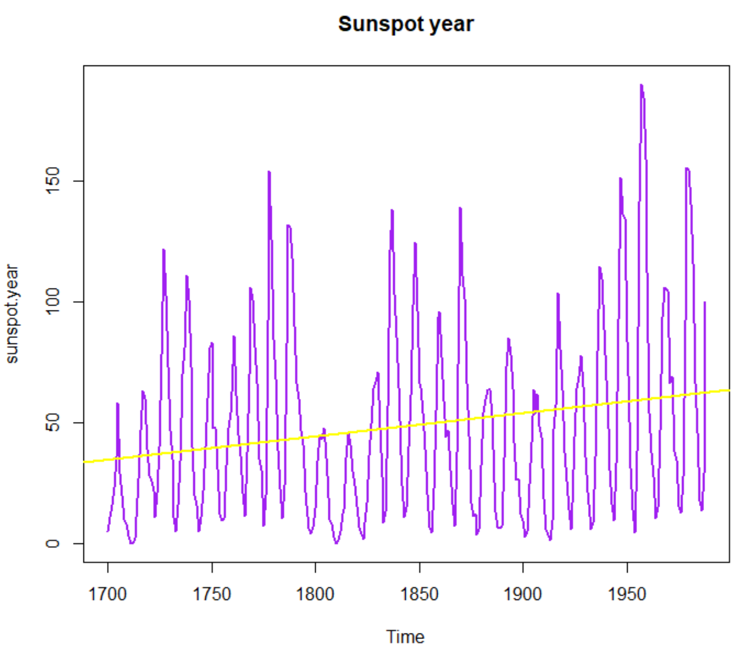


Рисунок 13 – Линейная регрессия и кривая изменения для sunspot.year

Выведем информацию о модели (рисунок 14).

```
Call:
lm(formula = sun ~ ., data = data.frame(time = time(sunspot.year),
    sun = sunspot.year))

Residuals:
    Min      1Q  Median      3Q     Max 
-54.893 -29.989  -7.767  22.882 130.615 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -130.42119   50.36684  -2.589 0.010104 *  
time           0.09709    0.02729   3.558 0.000437 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 38.7 on 287 degrees of freedom
Multiple R-squared:  0.04225,	Adjusted R-squared:  0.03891 
F-statistic: 12.66 on 1 and 287 DF,  p-value: 0.0004367
```

Рисунок 14 – Информация о модели

Листинг кода 7 задачи:

```
data(sunspot.year)
model <- lm(sun ~ ., data.frame(time = time(sunspot.year), sun = sunspot.year))
summary(model)
plot(sunspot.year, main = "Sunspot year", col = "purple",lwd = 2)
abline(model, col = "yellow", lwd = 2)
```

Задание 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.

Построим линейную регрессию для каждого квартала в отдельности и для всех кварталов вместе (рисунок 15)

```
Call:
lm(formula = qtr1 ~ data$time[seq(from = 1, to = nrow(data),
    by = 4)], data = data)

Coefficients:
                                       (Intercept)  data$time[seq(from = 1, to = nrow(data), by = 4)]
                                         -78854.23                                              40.22

> model2 <- lm(qtr2 ~ data$time[seq(from = 2, to = nrow(data), by = 4)], data)
> model2

Call:
lm(formula = qtr2 ~ data$time[seq(from = 2, to = nrow(data),
    by = 4)], data = data)

Coefficients:
                                       (Intercept)  data$time[seq(from = 2, to = nrow(data), by = 4)]
                                         -35297.23                                              18.04

> model3 <- lm(qtr3 ~ data$time[seq(from = 3, to = nrow(data), by = 4)], data)
> model3

Call:
lm(formula = qtr3 ~ data$time[seq(from = 3, to = nrow(data),
    by = 4)], data = data)

Coefficients:
                                       (Intercept)  data$time[seq(from = 3, to = nrow(data), by = 4)]
                                         -15403.73                                               7.89

> model4 <- lm(qtr4 ~ data$time[seq(from = 4, to = nrow(data), by = 4)], data)
> model4

Call:
lm(formula = qtr4 ~ data$time[seq(from = 4, to = nrow(data),
    by = 4)], data = data)

Coefficients:
                                       (Intercept)  data$time[seq(from = 4, to = nrow(data), by = 4)]
                                         -59112.72                                              30.14

> model5 <- lm((qtr1 + qtr2 + qtr3 + qtr4) ~ data$time[seq(from = 1, to = nrow(data), by = 4)], data)
> model5

Call:
lm(formula = (qtr1 + qtr2 + qtr3 + qtr4) ~ data$time[seq(from = 1,
    to = nrow(data), by = 4)], data = data)

Coefficients:
                                       (Intercept)  data$time[seq(from = 1, to = nrow(data), by = 4)]
                                        -188636.85                                              96.29

>
```

Рисунок 15 – Модели регрессии

Модель 1-го квартала: -78854.23+40.22 Time

Модель 2-го квартала: -35297.23+18.04 Time

Модель 3-го квартала: -15403.73+7.89 Time

Модель 4-го квартала: -59112.72+30.14 Time

Модель для всех: -188636.85+96.29 Time

Оценив полученные результаты, исходя из коэффициента при параметре время, можно сказать, что наибольшая динамика доходности потребления газа у 1-го квартала, а наименьшая у 3-го.

Прогноз для 2016 года для 1 квартала: 2230.936

Прогноз для 2016 года для 2 квартала: 1072.375

Прогноз для 2016 года для 3 квартала: 501.9919

Прогноз для 2016 года для 4 квартала: 1654.785

Прогноз для 2016 года в среднем по году: 1372.787

Листинг кода 8 задачи:

```
data <- read.csv("UKgas.csv", stringsAsFactors = TRUE)
head(data)

qtr1 <- data$UKgas[seq(from = 1, to = nrow(data), by = 4)]
qtr2 <- data$UKgas[seq(from = 2, to = nrow(data), by = 4)]
qtr3 <- data$UKgas[seq(from = 3, to = nrow(data), by = 4)]
qtr4 <- data$UKgas[seq(from = 4, to = nrow(data), by = 4)]

model1 <- lm(qtr1 ~ data$time[seq(from = 1, to = nrow(data), by = 4)], data)
model1
model2 <- lm(qtr2 ~ data$time[seq(from = 2, to = nrow(data), by = 4)], data)
model2
model3 <- lm(qtr3 ~ data$time[seq(from = 3, to = nrow(data), by = 4)], data)
model3
model4 <- lm(qtr4 ~ data$time[seq(from = 4, to = nrow(data), by = 4)], data)
model4
model5 <- lm((qtr1 + qtr2 + qtr3 + qtr4) ~ data$time[seq(from = 1, to = nrow(data), by = 4)], data)
model5

predict1 = coef(model1)[1]+coef(model1)[2]*2016
predict2 = coef(model2)[1]+coef(model2)[2]*2016
predict3 = coef(model3)[1]+coef(model3)[2]*2016
predict4 = coef(model4)[1]+coef(model4)[2]*2016
predict5 = (coef(model5)[1]+coef(model5)[2]*2016)/4

cat("Прогноз для 2016 года для 1 квартала: ", predict1, "\n")
cat("Прогноз для 2016 года для 2 квартала: ", predict2, "\n")
cat("Прогноз для 2016 года для 3 квартала: ", predict3, "\n")
cat("Прогноз для 2016 года для 4 квартала: ", predict4, "\n")
cat("Прогноз для 2016 года в среднем по году: ", predict5, "\n")
```

Задание 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.

Построим регрессионную модель по данным Cars (рисунок 16)

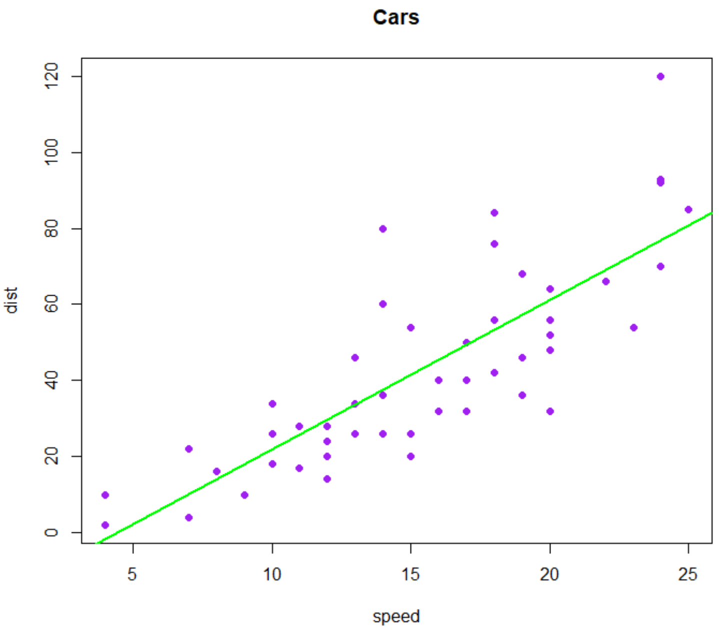


Рисунок 16 – Регрессионная модель

Длина тормозного пути при скорости 40 миль в час равна 139.7173.

<u>Листинг кода 9 задачи:</u>

```
data(cars)
model <- lm(dist ~ ., data = cars)
plot(cars, main = "Cars", pch = 16, col = "purple", lwd = 2)
abline(model, col = "green", lwd = 2)
predict.lm(model, data.frame(speed = 40))
```