**Лабораторная работа №5**

**“Линейная регрессия”**

по дисциплине “Машинное обучение”

|  |  |
| --- | --- |
| Выполнил  студент гр. 33504/2 | Лелюхин Д.О. |
| Руководитель | Селин И.А. |

**Оглавление**

[**Первое задание:** 3](#_Toc512329333)

[**Второе задание:** 3](#_Toc512329334)

[**Третье задание:** 4](#_Toc512329335)

[**Четвертое задание:** 4](#_Toc512329336)

[**Пятое задание:** 6](#_Toc512329337)

[**Шестое задание:** 8](#_Toc512329338)

[**Седьмое задание:** 9](#_Toc512329339)

[**Восьмое задание:** 11](#_Toc512329340)

[**Девятое задание:** 12](#_Toc512329341)

# **Первое задание:**

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

**Код программы:**

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

data <- lm(z ~ ., data = A)

summary(data)

**Результаты:**

Для модели x:

Multiple R-squared: 0.9187, Adjusted R-squared: 0.9178

F-statistic: 1113 on 2 and 197 DF, p-value: < 2.2e-16

Для модели y:

Multiple R-squared: 0.9505, Adjusted R-squared: 0.95

F-statistic: 1893 on 2 and 197 DF, p-value: < 2.2e-16

Для модели z:

Multiple R-squared: 0.9686, Adjusted R-squared: 0.9683

F-statistic: 3041 on 2 and 197 DF, p-value: < 2.2e-16

Выбираем модель z, т.к. коэффициенты детерминации у данной модели наиболее близки к 1 чем у других моделей.

# **Второе задание:**

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, ...).

**Код программы:**

A <- read.table("reglab2.txt", header = TRUE, sep = "\t")

combination <- combn(A[, 2:5], 2)

for(i in 1:length(combination[1,]))

{

tmp <- data.frame(A$y, combination[1, i], combination[2, i])

res <- lm(tmp$A.y ~., data = tmp)

print(sum((tmp$A.y - res$fitted.values)^2))

}

**Результаты:**

[1] 0.5379617

[1] 156.3541

[1] 157.2193

[1] 267.7955

[1] 267.8061

[1] 393.4587

По результатам видно, что минимальная RSS у первого столбца.

# **Третье задание:**

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

**Код программы:**

A <- read.table("cygage.txt", header = TRUE, sep = "\t")

f <- lm(calAge~Depth, data=A, weights = A$Weight)

summary(f)

**Результаты:**

Multiple R-squared: 0.9737, Adjusted R-squared: 0.9711

F-statistic: 370 on 1 and 10 DF, p-value: 3.141e-09

Коэффициенты детерминации близки к единице.

# **Четвертое задание:**

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 λ = , *i* = 0,…,25, подсчитайте ошибку на тестовой и обучающей выборке для данных значений λ, постройте графики. Объясните полученные результаты.

**Код программы:**

library(MASS)

library(datasets)

A<-longley

res <- lm(Employed ~., data = A)

summary(res)

A.1 <- A[,-5]

set.seed(12345)

n <- dim(A.1)[1]

A\_rand <- A.1[ order(runif(n)), ]

A\_train <- A\_rand[1:(n/2),]

A\_test <- A\_rand[((n/2)+1):n,]

ml = c()

for (i in 1:25){

ml <- c(ml, 10^(-3 + 0.2 \* i))

}

res <- lm.ridge(Employed ~., data = A\_train, lambda = ml)

plot(x=res$lambda, y=res$GCV, type="o")

lambda <- res$GCV[which.min(res$GCV)]

res <- lm.ridge(Employed ~., data = A\_train, lambda = lambda)

print(res)

for(i in 1:25){

res.test <- lm(Employed ~., data = A\_train)

res.error <- (res.test$fitted.values - A\_train$Employed) ^ 2

res <- lm.ridge(Employed ~., data = A\_train, lambda = ml[i])

res.error <- res.error + ml[i] \* sum(abs(res$coef))

print(res.error)

}

**Результаты:**

Построил регрессию lm(Employed~.)

Multiple R-squared: 0.9955, Adjusted R-squared: 0.9925

F-statistic: 330.3 on 6 and 9 DF, p-value: 4.984e-10

Построил гребневую регрессию для значений λ = , *i* = 0,…,25.

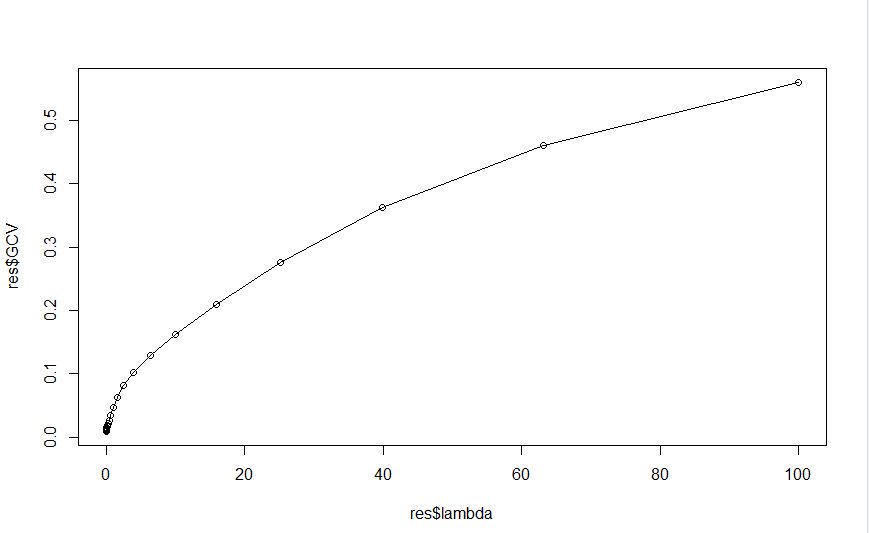


График зависимости критериев GCV от значений заданных лямбд. Оптимальное значение лямбды равно 0.00158.

# **Пятое задание:**

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.

**Код программы:**

#Unit 5

library(datasets)

data("EuStockMarkets")

plot(EuStockMarkets)

for(i in 1:4){

res <- lm(EuStockMarkets[,i]~., data = EuStockMarkets[,-i])

print(res)

print(summary(res))

png(file = paste(toString(i),'lm.png'))

plot(res)

dev.off()

}

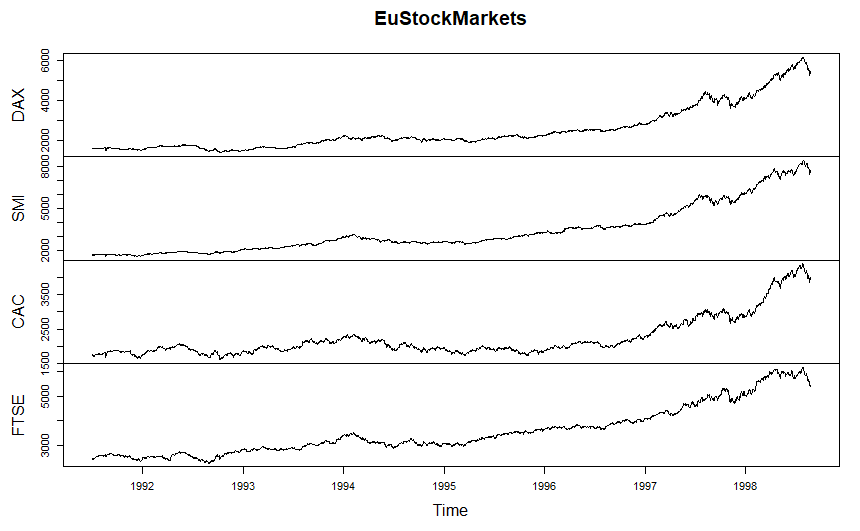
plot(EuStockMarkets[, 1], col = "red", ylab = "DAX SMI CAC FTSE")

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

lines(EuStockMarkets[, 3], col = "green")

lines(EuStockMarkets[, 4], col = "yellow")

**Результаты:**



Линейные регрессии:

Multiple R-squared: 0.9898, Adjusted R-squared: 0.9898

F-statistic: 6.032e+04 on 3 and 1856 DF, p-value: < 2.2e-16

Multiple R-squared: 0.9936, Adjusted R-squared: 0.9936

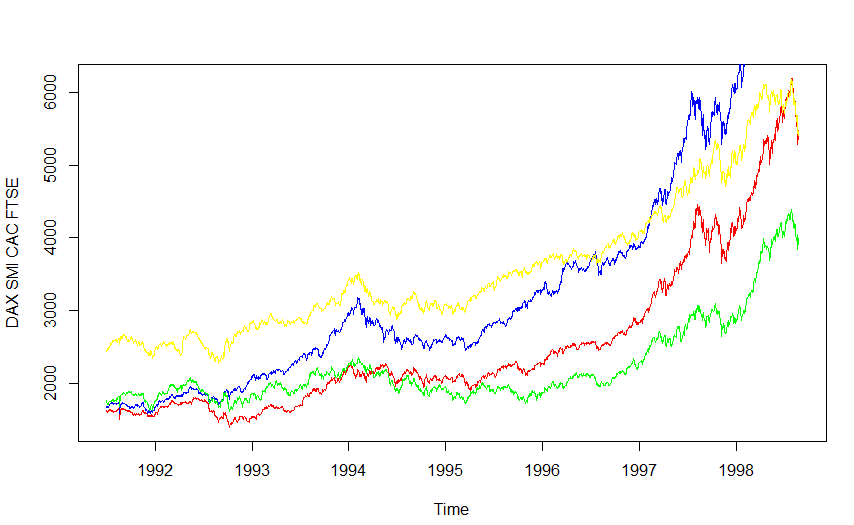
F-statistic: 9.656e+04 on 3 and 1856 DF, p-value: < 2.2e-16

Multiple R-squared: 0.9484, Adjusted R-squared: 0.9484

F-statistic: 1.138e+04 on 3 and 1856 DF, p-value: < 2.2e-16

Multiple R-squared: 0.9845, Adjusted R-squared: 0.9845

F-statistic: 3.941e+04 on 3 and 1856 DF, p-value: < 2.2e-16



На графике видно, что SMI выглядит лучше, чем DAX, CAC, FTSE.

# **Шестое задание:**

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.

**Код программы:**

library(datasets)

A <- JohnsonJohnson

plot(A)

i = 1

A.1 <- cbind(A[i], A[i+1], A[i+2], A[i+3])

i = 5

while(i < 84){

myTmp <- cbind(A[i], A[i+1], A[i+2], A[i+3])

A.1 <- rbind(A.1, myTmp)

i <- i + 4

}

A.1 <- data.frame(A.1, 1960:1980)

res<-lm(X1+X2+X3+X4~X1960.1980, data = A.1)

res1 <- lm(X1~X1960.1980, data = A.1)

res2<-lm(X2~X1960.1980, data = A.1)

res3<-lm(X3~X1960.1980, data = A.1)

res4<-lm(X4~X1960.1980, data = A.1)

plot(res$fitted.values, x = (1960:1980))

lines(res1$fitted.values, col = "black", x = (1960:1980))

lines(res2$fitted.values, col = "blue", x = (1960:1980))

lines(res3$fitted.values, col = "green", x = (1960:1980))

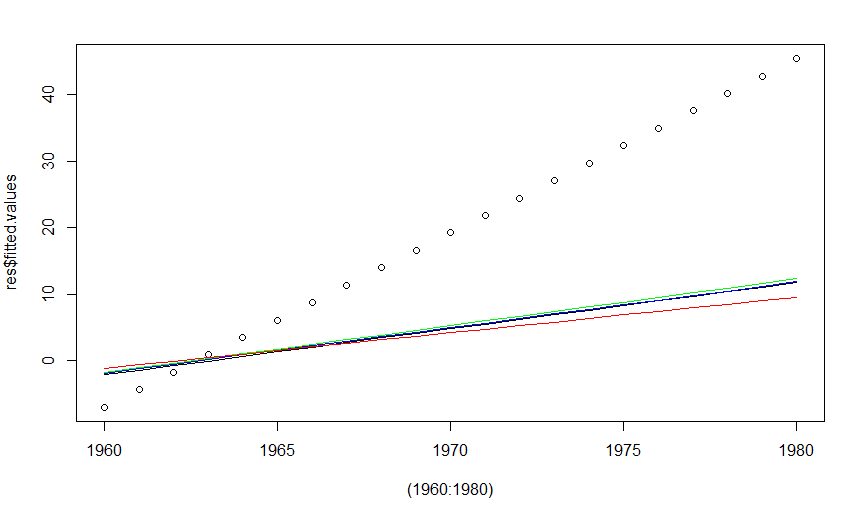
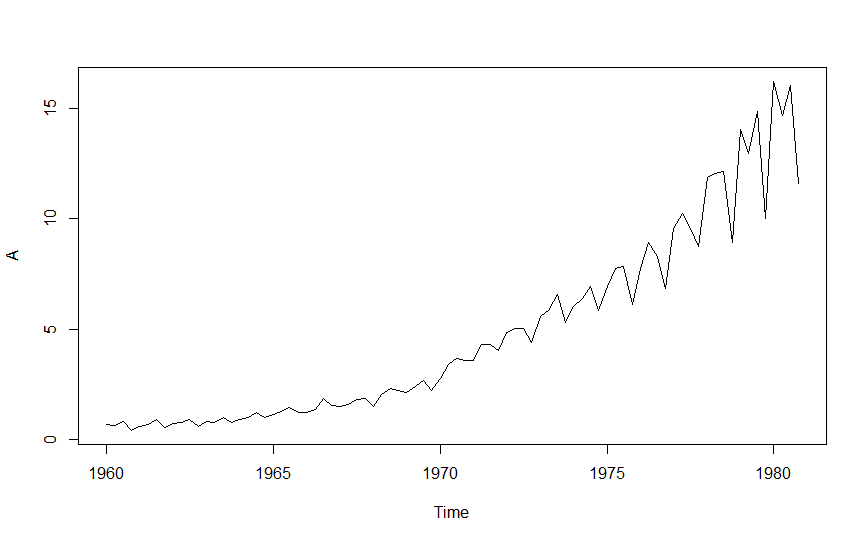
lines(res4$fitted.values, col = "red", x = (1960:1980))

new<- data.frame(X1960.1980 = (2016))

pred <- predict(res1, newdata = new )

print(pred)

**Результаты:**



Прогноз на 2016 год:  
1 квартал – 36.75964;

2 квартал – 36.48945;

3 квартал – 37.65394;

4 квартал – 28.79391;

год – 139.6969;

Лучший показатель у 3-го квартала, худший у 4-го.

# **Седьмое задание:**

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.

**Код программы:**

library(datasets)

A <- sunspot.year

plot(A)

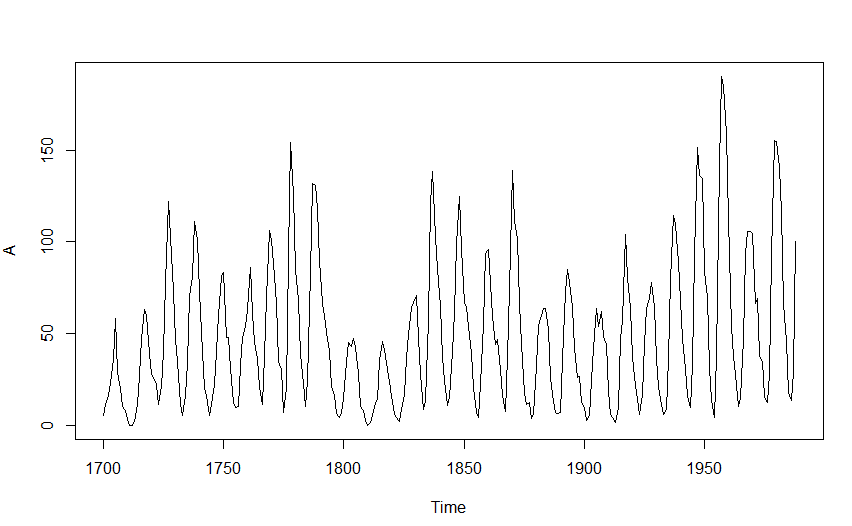
A <- data.frame(A, Year=1700:1988)

res <- lm(A~Year, data = A)

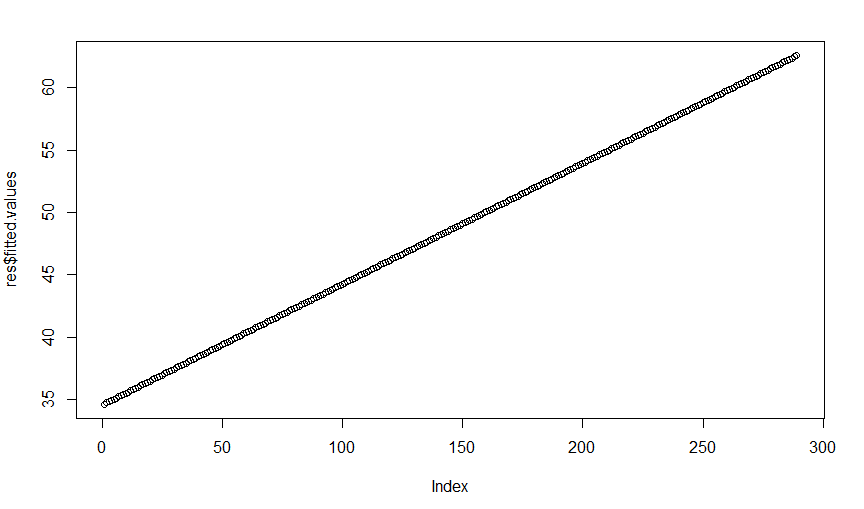
plot(res$fitted.values)

**Результаты:**

Кривая изменения числа солнечных пятен во времени:



Линейная регрессия для данных:



# **Восьмое задание:**

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.

**Код программы:**

library(datasets)

A <- read.csv("UKgas.csv")

i = 1

print(A[i,3])

A.1 <- cbind(A[i,3], A[i+1,3], A[i+2,3], A[i+3, 3])

i = 5

while(i < 108){

Tmp <- cbind(A[i,3], A[i+1,3], A[i+2,3], A[i+3, 3])

A.1 <- rbind(A.1, Tmp)

i <- i + 4

}

A.1 <- data.frame(A.1, Year=1960:1986)

res<-lm(X1+X2+X3+X4~Year, data = A.1)

res1<-lm(X1~Year, data = A.1)

res2<-lm(X2~Year, data = A.1)

res3<-lm(X3~Year, data = A.1)

res4<-lm(X4~Year, data = A.1)

plot(res$fitted.values, x = (1960:1986))

lines(res1$fitted.values, col = "yellow", x = (1960:1986))

lines(res2$fitted.values, col = "blue", x = (1960:1986))

lines(res3$fitted.values, col = "green", x = (1960:1986))

lines(res4$fitted.values, col = "red", x = (1960:1986))

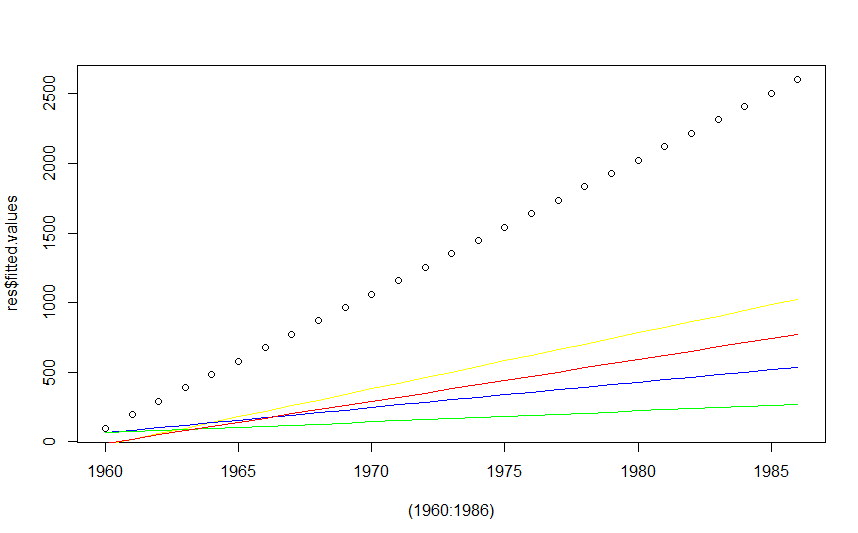
new<- data.frame(Year = (2016))

pred <- predict(res1, newdata = new )

print(pred)

plot(pred, x = (1980:2016))

**Результаты:**



Предсказания:

1 квартал - 2230.936;

2 квартал - 1076.885;

3 квартал - 505.9368;

4 квартал - 1677.392;

Год - 5491.149;

Наиболее удовлетворительный первый квартал

# **Девятое задание:**

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.

**Код программы:**

library(datasets)

A <- cars

plot(A)

res<-lm(dist~speed, data = A)

plot(res$fitted.values)

new <-data.frame(speed=25:40)

dist <- c(predict(res,newdata = new))

new <- cbind(new, dist)

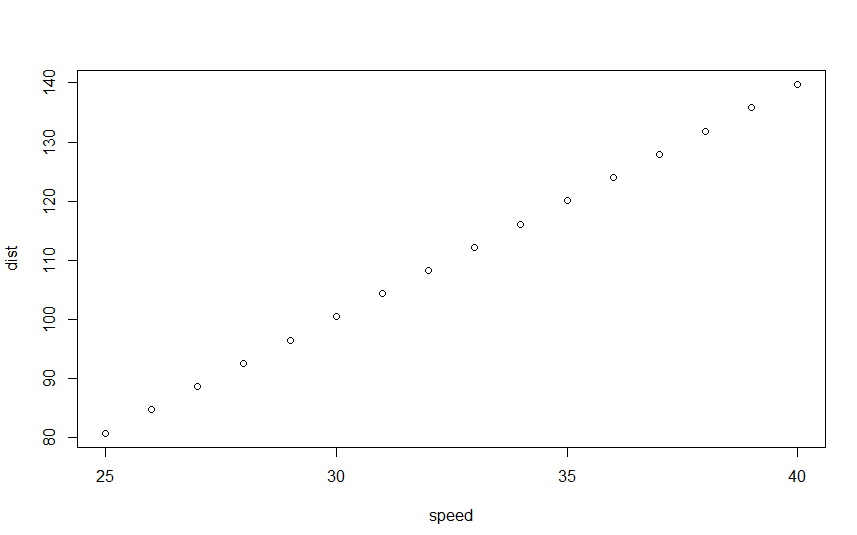
plot(new)

new2 <-data.frame(speed=40)

pred <- predict(res,newdata = new2)

print(pred)

**Результаты:**



Длинна тормозного пути при 40 миль в час - 139.7173 футов.