Notebook 1 Regression

Umar Ali-Salaam, Carolline Osei

2022-10-23

Source: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset

This is a dataset based off of 70,000 records of patient data (Heart Related). Columns (13): ID, Age, Height(cm), Weight(kg), Gender, Systolic Blood Pressure (AP_HIGH), Diastolic Blood Pressure (AP LOW), Cholesterol, Glucose, Smoking, Alcohol Intake, Physical Activity, Presence or Absence of cardiovascular disease.

The .csv file needed to be edited a bit in Microsoft Excel before using it in R. I just performed a split column delimiter function around semicolons, to divide the singular column that existed into 13. Each row had 13 variables in 1 column separated by semicolons, the function I ran split it up into 13 columns, making a 70,000 x 13 table.

https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings

Visit the website above to better understand Systolic and Diastolic Blood Pressure

Cleaning Data

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.1.3
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.1.3
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
# Read in .csv file
heart <- read.csv("cardio_train.csv")


# Clean out any rows that have an unrealistic blood pressure (AP_HIGH & AP_LOW)
# They looked to be input errors by the person who made the data set
s <- subset(heart, AP_HIGH > 50)

s1 <- subset(s, AP_HIGH < 200)

s2 <- subset(s1, AP_LOW > 25)

s3 <- subset(s2, AP_LOW < 200)


# Removing little people(4'10") and Giants(7'3"), values are in cm
s4 <- subset(s3, HEIGHT > 147)

s5 <- subset(s4, HEIGHT < 220)


# Removing anyone below 90 lbs and above 375 lbs, the values are in kg
s6 <- subset(s5, WEIGHT > 40)

h1 <- subset(s6, WEIGHT < 180)

# AGE is in days so to get years i just divide by 365
h1$AGE <- (h1$AGE / 365)

# Removing people under 40
h <- subset(h1, AGE > 39)

# Checking for any NA values
# There is none
colSums(is.na(h))
```

```
##                ID               AGE            GENDER            HEIGHT
##                 0                 0                 0                 0
##            WEIGHT           AP_HIGH            AP_LOW       CHOLESTEROL
##                 0                 0                 0                 0
##           GLUCOSE             SMOKE           ALCOHOL PHYSICAL_ACTIVITY
##                 0                 0                 0                 0
##    CARDIO_DISEASE
##                 0
```

```r
# Everything that should be factored is factored
h$GENDER <- factor(h$GENDER)
h$CHOLESTEROL <- factor(h$CHOLESTEROL)
h$GLUCOSE <- factor(h$GLUCOSE)
h$SMOKE <- factor(h$SMOKE)
h$ALCOHOL <- factor(h$ALCOHOL)
h$PHYSICAL_ACTIVITY <- factor(h$PHYSICAL_ACTIVITY)
h$CARDIO_DISEASE <- factor(h$CARDIO_DISEASE)
```

```
# There is now 67,685 rows

str(h)
```

```
## 'data.frame':    67685 obs. of  13 variables:
##  $ ID                : int  0 1 2 3 4 8 9 12 13 14 ...
##  $ AGE               : num  50.4 55.4 51.7 48.3 47.9 ...
##  $ GENDER            : Factor w/ 2 levels "1","2": 2 1 1 2 1 1 1 2 1 1 ...
##  $ HEIGHT            : int  168 156 165 169 156 151 157 178 158 164 ...
##  $ WEIGHT            : num  62 85 64 82 56 67 93 95 71 68 ...
##  $ AP_HIGH           : int  110 140 130 150 100 120 130 130 110 110 ...
##  $ AP_LOW            : int  80 90 70 100 60 80 80 90 70 60 ...
##  $ CHOLESTEROL       : Factor w/ 3 levels "1","2","3": 1 3 3 1 1 2 3 3 1 1 ...
##  $ GLUCOSE           : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 1 3 1 1 ...
##  $ SMOKE             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ALCOHOL           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PHYSICAL_ACTIVITY: Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 2 2 1 ...
##  $ CARDIO_DISEASE    : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 1 1 ...
```

Train Test Validate

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.1.3
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.3
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:flexclust':
##
##     bclust
```

```
# Splittling data into train and test
# Reducing data size to 10,000 for the sake of the speed of the SVM function

h <- h[1:10000,]

split = sample.split(h, SplitRatio = 0.6)

hTrain = subset(h, split == TRUE)

h2 = subset(h, split == FALSE)
```

```
split1 = sample.split(h2, SplitRatio = 0.5)

hTest = subset(h2, split1 == TRUE)

hVal = subset(h2, split1 == FALSE)


# Showing length of each dataset

cat("Train data has", nrow(hTrain), "rows.")
```

```
## Train data has 5384 rows.
```

```
cat("\nTest data has", nrow(hTest), "rows.")
```

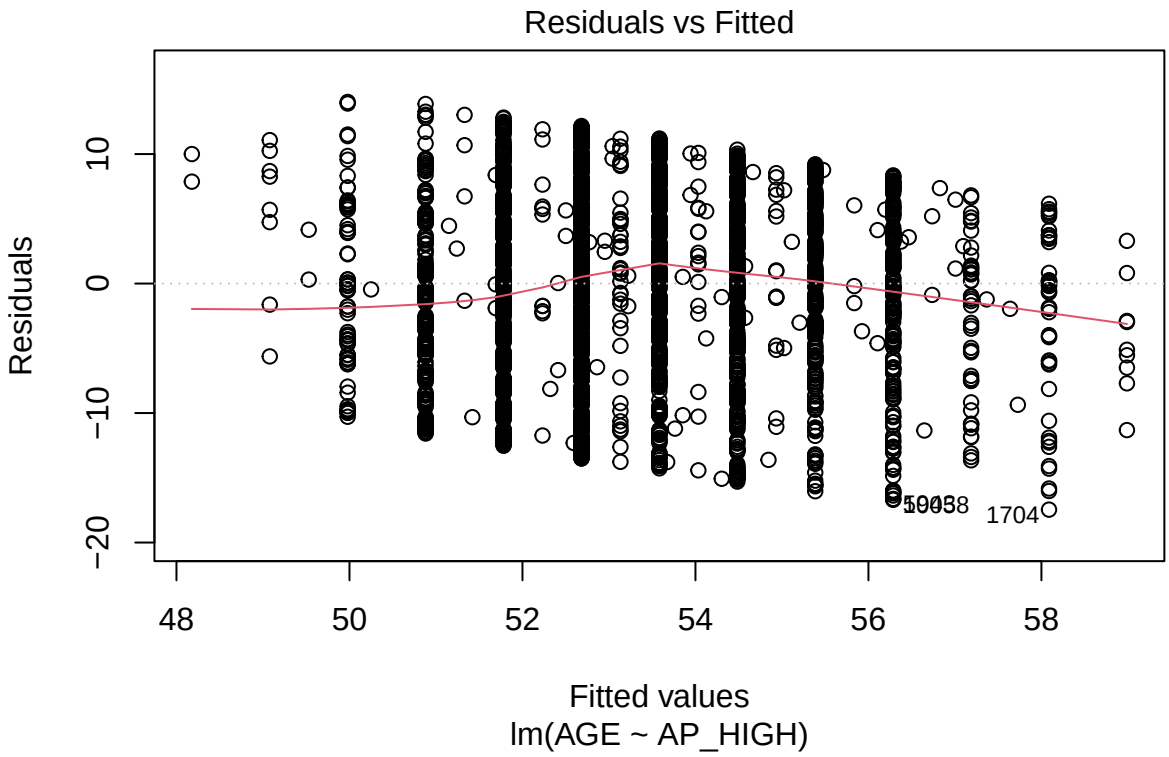
```
##
## Test data has 2130 rows.
```

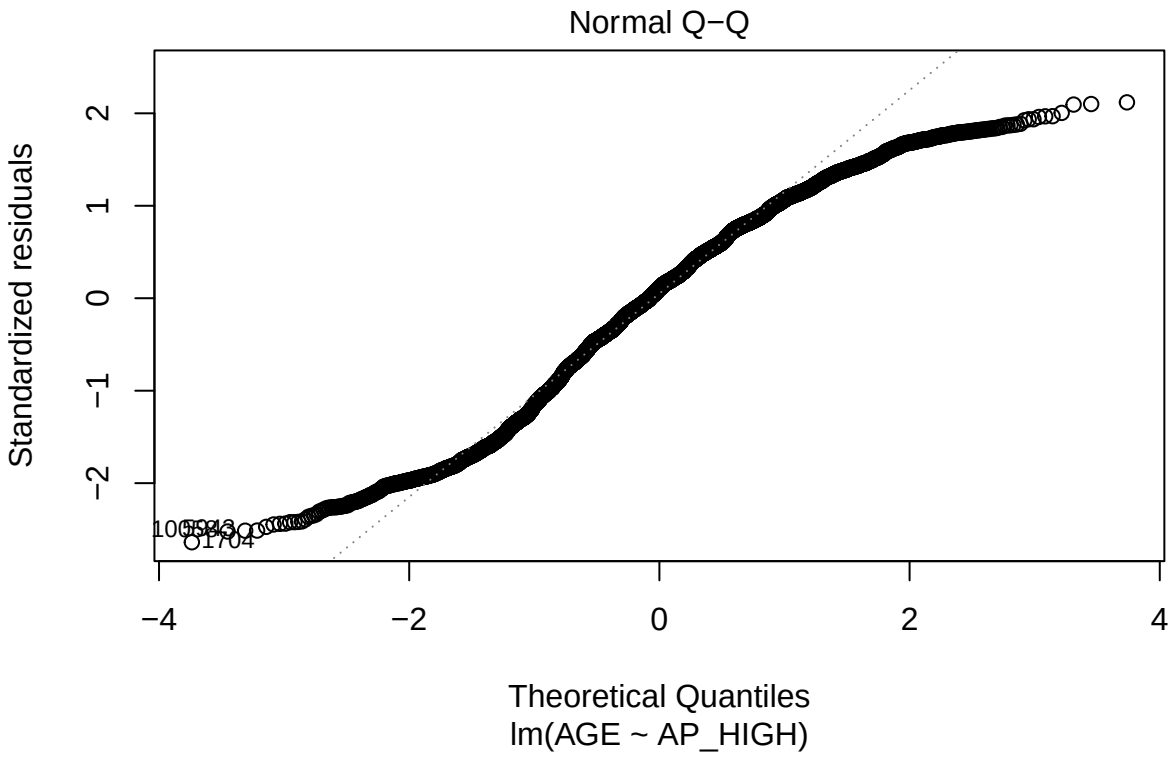
```
cat("\nValidation data has", nrow(hVal), "rows.")
```

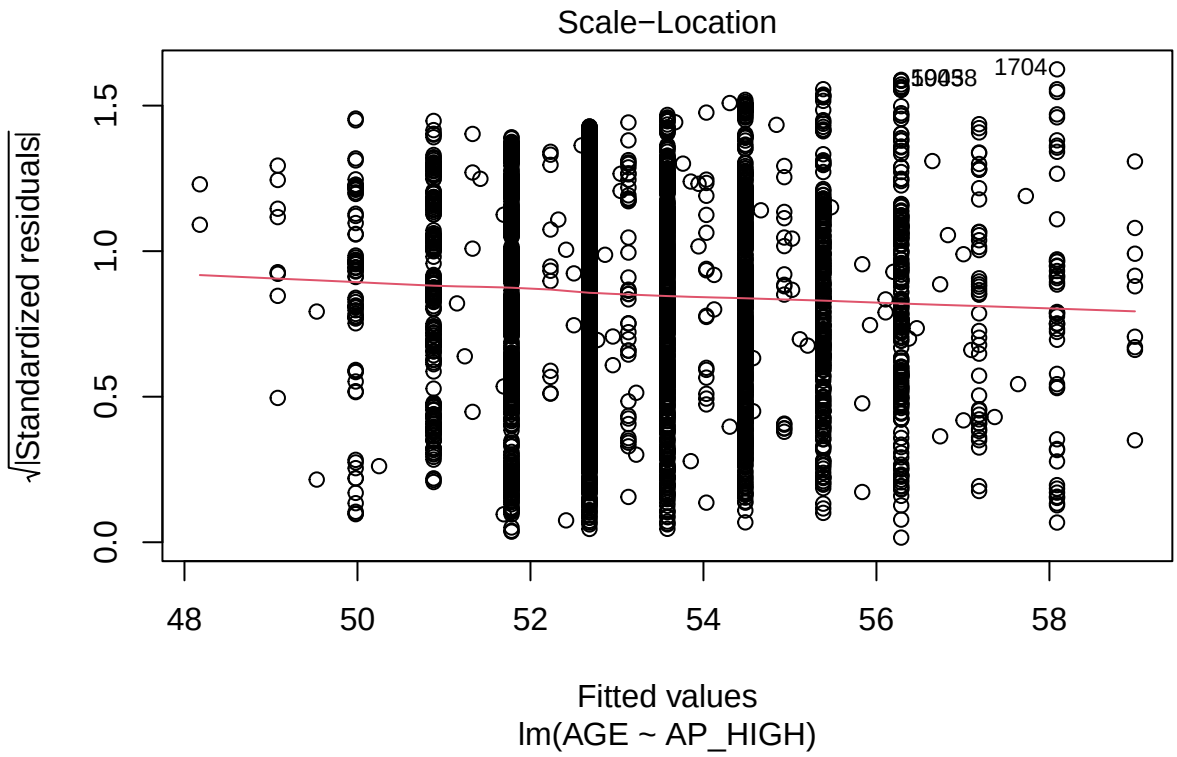
```
##
## Validation data has 2486 rows.
```

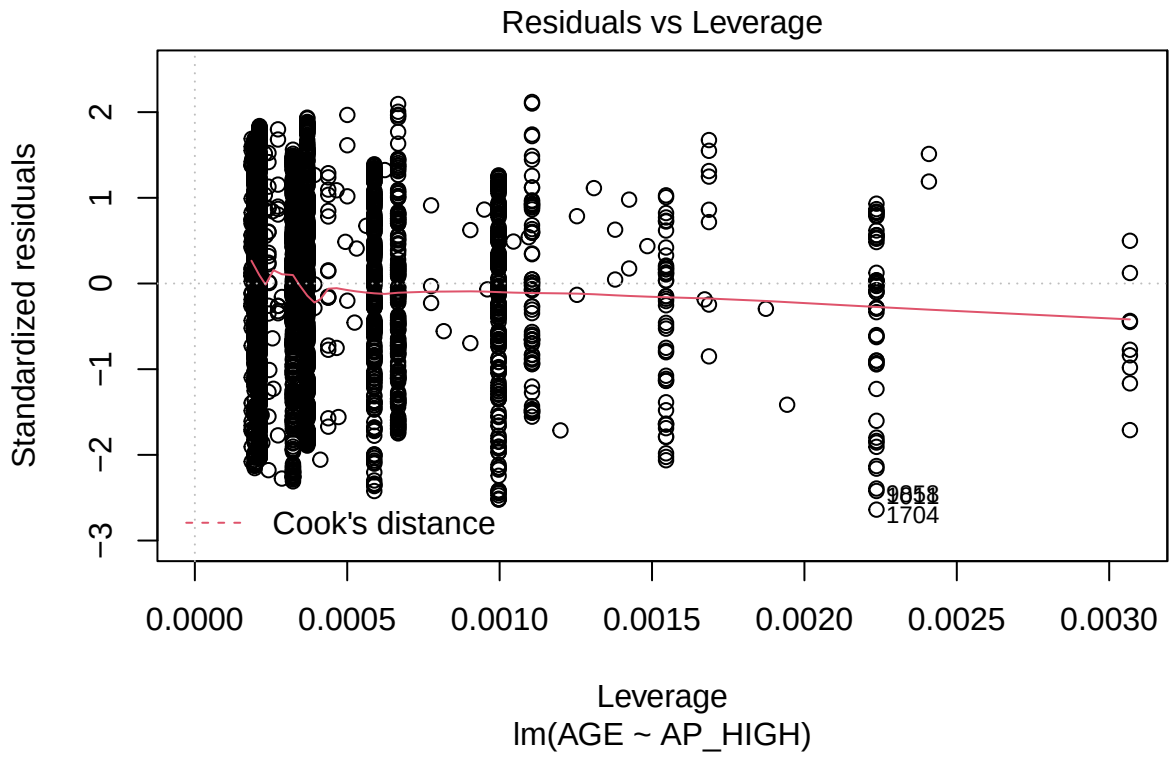
Exploring data

```
# Exploring data through linear regression, correlation test, and histogram
# distribution of age

LM <- lm(AGE ~ AP_HIGH, data = hTrain)

plot(LM)
```


Residuals vs Fitted
Residuals
10
0
−10
−20
48
50
52
54
56
58
1704
Fitted values
lm(AGE ~ AP_HIGH)


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−4
−2
0
2
4
1704
Theoretical Quantiles
lm(AGE ~ AP_HIGH)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
1704
48
50
52
54
56
58
Fitted values
lm(AGE ~ AP_HIGH)




```
cor.test(hTrain$AGE, hTrain$AP_HIGH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hTrain$AGE and hTrain$AP_HIGH
## t = 16.19, df = 5382, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1898811 0.2408266
## sample estimates:
##       cor
## 0.2155005
```

```
hist(hTrain$AGE)
```

Linear Regression

```
# Linear Regression on weight

LM <- lm(WEIGHT ~ ., data = hTrain)

pred <- predict(LM, newdata=hTest)

corLM <- cor(pred, hTest$WEIGHT)

mseLM <- mean((pred-hTest$WEIGHT) ^ 2)
```

SVM Linear Kernel

```
# Performing SVM "Linear" on weight

svmLM10 <- svm(WEIGHT ~ AP_HIGH, data = hTrain, kernel = "linear",
               cost = 10, gamma = 0.5, scale = TRUE)

svmLM1 <- svm(WEIGHT ~ AP_HIGH, data = hTrain, kernel = "linear",
              cost = 1, gamma = 0.1, scale = TRUE)
```

```r
# Summary of each gamma and cost change

summary(svmLM10)
```

```
##
## Call:
## svm(formula = WEIGHT ~ AP_HIGH, data = hTrain, kernel = "linear",
##     cost = 10, gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  0.5
##     epsilon:  0.1
##
##
## Number of Support Vectors:  4836
```

```r
summary(svmLM1)
```

```
##
## Call:
## svm(formula = WEIGHT ~ AP_HIGH, data = hTrain, kernel = "linear",
##     cost = 1, gamma = 0.1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  1
##       gamma:  0.1
##     epsilon:  0.1
##
##
## Number of Support Vectors:  4838
```

```r
# correlation and mse of cost = 10, gamma = 0.5

predLM10 <- predict(svmLM10, newdata = hTest)

corLM10 <- cor(predLM10, hTest$WEIGHT)

mseLM10 <- mean((predLM10 - hTest$WEIGHT) ^ 2)

corLM10
```

```
## [1] 0.2447804
```

```
mseLM10
```

```
## [1] 192.6721
```

```
# correlation and mse of cost = 1, gamma = 0.1

predLM1 <- predict(svmLM1, newdata = hTest)

corLM1 <- cor(predLM1, hTest$WEIGHT)

mseLM1 <- mean((predLM1 - hTest$WEIGHT) ^ 2)

corLM1
```

```
## [1] 0.2447804
```

```
mseLM1
```

```
## [1] 192.6672
```

SVM Polynomial

```
# Performing SVM "Polynomial" on the presence of heart disease

svmP10 <- svm(WEIGHT ~ ., data = hTrain, kernel = "polynomial",
              cost = 10, gamma = 0.5, scale = TRUE)

svmP1 <- svm(WEIGHT ~ ., data = hTrain, kernel = "polynomial",
              cost = 1, gamma = 0.1, scale = TRUE)

# Summary of each gamma and cost change

summary(svmP10)
```

```
##
## Call:
## svm(formula = WEIGHT ~ ., data = hTrain, kernel = "polynomial", cost = 10,
##     gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##       gamma:  0.5
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  4782
```

```r
summary(svmP1)
```

```
##
## Call:
## svm(formula = WEIGHT ~ ., data = hTrain, kernel = "polynomial", cost = 1,
##     gamma = 0.1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##       gamma:  0.1
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  4838
```

```r
# correlation and mse of cost = 10, gamma = 0.5

predP10 <- predict(svmP10, newdata = hTest)

corP10 <- cor(predP10, hTest$WEIGHT)

mseP10 <- mean((predP10 - hTest$WEIGHT) ^ 2)

corP10
```

```
## [1] 0.3427868
```

```r
mseP10
```

```
## [1] 198.0551
```

```r
# correlation and mse of cost = 1, gamma = 0.1

predP1 <- predict(svmP1, newdata = hTest)

corP1 <- cor(predP1, hTest$WEIGHT)

mseP1 <- mean((predP1 - hTest$WEIGHT) ^ 2)

corP1
```

```
## [1] 0.3960201
```

```r
mseP1
```

```
## [1] 175.4163
```

SVM Radial

```
# Performing SVM "Radial" on the presence of heart disease

svmR10 <- svm(WEIGHT ~ ., data = hTrain, kernel = "radial",
              cost = 10, gamma = 0.5, scale = TRUE)

svmR1 <- svm(WEIGHT ~ ., data = hTrain, kernel = "radial",
              cost = 1, gamma = 0.1, scale = TRUE)

# Summary of each gamma and cost change

summary(svmR1)
```

```
##
## Call:
## svm(formula = WEIGHT ~ ., data = hTrain, kernel = "radial", cost = 1,
##     gamma = 0.1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.1
##     epsilon:  0.1
##
##
## Number of Support Vectors:  4806
```

```
summary(svmR10)
```

```
##
## Call:
## svm(formula = WEIGHT ~ ., data = hTrain, kernel = "radial", cost = 10,
##     gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  0.5
##     epsilon:  0.1
##
##
## Number of Support Vectors:  4925
```

```
# correlation and mse of cost = 10, gamma = 0.5

predR10 <- predict(svmR10, newdata = hTest)
```

```r
corR10 <- cor(predR10, hTest$WEIGHT)

mseR10 <- mean((predR10 - hTest$WEIGHT) ^ 2)

corR10
```

```
## [1] 0.255267
```

```r
mseR10
```

```
## [1] 232.9272
```

```r
# correlation and mse of cost = 1, gamma = 0.1

predR1 <- predict(svmR1, newdata = hTest)

corR1 <- cor(predR1, hTest$WEIGHT)

mseR1 <- mean((predR1 - hTest$WEIGHT) ^ 2)

corR1
```

```
## [1] 0.4047972
```

```r
mseR1
```

```
## [1] 172.6749
```

Analysis

```r
# Explaining results

cat("  Changing the gamma and cost on linear SVM seemed to have no effect on the\n
correlation or mse. The correlation is relatively low as well, it doesn't seem like\n
the linear kernel is the best model for SVM.\n\n
  Changing the gamma and cost on polynomial SVM seemed to help it greatly on the\n
correlation and mse. The correlation increased by 0.08 due to the decrease in cost,\n
and the mse decreased which means there were less improperly placed points on either\n
side of the line.It seems like the polynomial kernel is a better model for SVM than\n
linear.\n\n
  Changing the gamma and cost on radial SVM seemed to help it the most on the\n
correlation and mse. The correlation increased by 0.13 due to the decrease in cost.\n
The mse decreased which means there were less improperly placed points on either\n
side of the line. This is the only kernel that actually uses gamma out of the 3, if\n
I used a higher gamma I think the correlation would've improved more, given the nature\n
of the parameter. It seems like the radial kernel is the best model for SVM since\n
it has the highest correlation and lowest mse.")
```

```
##   Changing the gamma and cost on linear SVM seemed to have no effect on the
##
## correlation or mse. The correlation is relatively low as well, it doesn't seem like
##
## the linear kernel is the best model for SVM.
##
##
##   Changing the gamma and cost on polynomial SVM seemed to help it greatly on the
##
## correlation and mse. The correlation increased by 0.08 due to the decrease in cost,
##
## and the mse decreased which means there were less improperly placed points on either
##
## side of the line.It seems like the polynomial kernel is a better model for SVM than
##
## linear.
##
##
##   Changing the gamma and cost on radial SVM seemed to help it the most on the
##
## correlation and mse. The correlation increased by 0.13 due to the decrease in cost.
##
## The mse decreased which means there were less improperly placed points on either
##
## side of the line. This is the only kernel that actually uses gamma out of the 3, if
##
## I used a higher gamma I think the correlation would've improved more, given the nature
##
## of the parameter. It seems like the radial kernel is the best model for SVM since
##
## it has the highest correlation and lowest mse.
```