```r
# import necessary libraries

library(readxl) # to read excel
library(plyr)
library(caTools)

library(e1071)
library(caret)
library(randomForest)
```

Data Understanding

```r
# load data
attribset = read_excel('Attribute DataSet.xlsx')
dresssale = read_excel('Dress Sales.xlsx')

#remove Dress_ID column
attribset_ = attribset[2:14]
dresssale_ = dresssale[2:24]
```

Data Preparation

Attributes dataset

```r
# check the unique values for each columns
#lapply(attribset[2:14], unique)

# values checking
# style
attribset_$Style[attribset_$Style == 'sexy'] = 'Sexy'

# Price
attribset_$Price[attribset_$Price == 'low'] = 'Low'
attribset_$Price[attribset_$Price == 'high'] = 'High'

# Size
attribset_$Size[attribset_$Size == 's'] = 'S'
attribset_$Size[attribset_$Size == 'small'] = 'S'

# Season
attribset_$Season[attribset_$Season == 'spring'] = 'Spring'
attribset_$Season[attribset_$Season == 'summer'] = 'Summer'
attribset_$Season[attribset_$Season == 'Automn'] = 'Autumn'
attribset_$Season[attribset_$Season == 'winter'] = 'Winter'

# NeckLine
attribset_$NeckLine[attribset_$NeckLine == 'sweetheart'] =
'Sweetheart'

# SleeveLength
```

```r
attribset_$SleeveLength[attribset_$SleeveLength == 'sleevless'] =
'sleeveless'
attribset_$SleeveLength[attribset_$SleeveLength == 'sleeevless'] =
'sleeveless'
attribset_$SleeveLength[attribset_$SleeveLength == 'sleveless'] =
'sleeveless'
attribset_$SleeveLength[attribset_$SleeveLength == 'threequater'] =
'threequarter'
attribset_$SleeveLength[attribset_$SleeveLength == 'thressqatar'] =
'threequarter'
attribset_$SleeveLength[attribset_$SleeveLength == 'urndowncollor'] =
'turndowncollar'

# FabricType
attribset_$FabricType[attribset_$FabricType == 'shiffon'] = 'chiffon'
attribset_$FabricType[attribset_$FabricType == 'sattin'] = 'satin'
attribset_$FabricType[attribset_$FabricType == 'wollen'] = 'woolen'
attribset_$FabricType[attribset_$FabricType == 'flannael'] = 'flannel'
attribset_$FabricType[attribset_$FabricType == 'knitting'] = 'knitted'


# Decoration
attribset_$Decoration[attribset_$Decoration == 'embroidary'] =
'embroidery'
attribset_$Decoration[attribset_$Decoration == 'sequined'] = 'sequins'
attribset_$Decoration[attribset_$Decoration == 'ruched'] = 'ruche'
attribset_$Decoration[attribset_$Decoration == 'none'] = 'null'

# Pattern Type
attribset_$'Pattern Type'[attribset_$'Pattern Type' == 'none'] =
'null'
attribset_$'Pattern Type'[attribset_$'Pattern Type' == 'leapord'] =
'leopard'

# factoring

attribset_$Style = factor(attribset_$Style,
                          levels = c('Sexy', 'Casual', 'vintage',
'Brief', 'cute', 'bohemian', 'Novelty', 'Flare', 'party', 'work',
'OL', 'fashion'),
                          labels = c(0,1,2,3,4,5,6,7,8,9,10,11))

attribset_$Price = factor(attribset_$Price,
                          levels = c('Low', 'High', 'Average',
'Medium', 'very-high'),
                          labels = c(0,1,2,3,4))

attribset_$Size = factor(attribset_$Size,
                          levels = c('M', 'L', 'XL', 'free', 'S'),
                          labels = c(0,1,2,3,4))
```

```
attribset_$Season = factor(attribset_$Season,
                          levels = c('Summer', 'Autumn', 'Spring',
'Winter'),
                          labels = c(0,1,2,3))

attribset_$NeckLine = factor(attribset_$NeckLine,
                          levels = c('o-neck', 'v-neck', 'boat-neck',
'peterpan-collor', 'ruffled', 'turndowncollor', 'slash-neck',
'mandarin-collor', 'open', 'sqare-collor', 'Sweetheart', 'Scoop',
'halter', 'backless', 'bowneck', 'NULL'),
                          labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))

attribset_$SleeveLength = factor(attribset_$SleeveLength,
                          levels = c('sleeveless', 'Petal', 'full',
'butterfly', 'short', 'threequarter', 'halfsleeve', 'cap-sleeves',
'turndowncollor', 'capsleeves', 'half', 'turndowncollar', 'NULL'),
                          labels = c(0,1,2,3,4,5,6,7,8,9,10,11,12))

attribset_$waiseline = factor(attribset_$waiseline,
                          levels = c('empire', 'natural', 'null',
'princess', 'dropped'),
                          labels = c(0,1,2,3,4))

attribset_$Material = factor(attribset_$Material,
                          levels = c('null', 'microfiber', 'polyster',
'silk', 'chiffonfabric', 'cotton', 'nylon', 'other', 'milksilk',
'linen', 'rayon', 'lycra', 'mix', 'acrylic', 'spandex', 'lace',
'modal', 'cashmere', 'viscos', 'knitting', 'sill', 'wool', 'model',
'shiffon'),
                          labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23))

attribset_$FabricType = factor(attribset_$FabricType,
                          levels = c('chiffon', 'null', 'broadcloth',
'jersey', 'other', 'batik', 'satin', 'flannel', 'worsted', 'woolen',
'poplin', 'dobby', 'knitted', 'tulle', 'organza', 'lace', 'Corduroy',
'terry'),
                          labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17))

attribset_$Decoration = factor(attribset_$Decoration,
                          levels = c('ruffles', 'null', 'embroidery',
'bow', 'lace', 'beading', 'sashes', 'hollowout', 'pockets', 'sequins',
'applique', 'button', 'Tiered', 'rivet', 'feathers', 'flowers',
'pearls', 'pleat', 'crystal', 'ruche', 'draped', 'tassel', 'plain',
'cascading'),
                          labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23))
```

```
attribset_$`Pattern Type` = factor(attribset_$`Pattern Type`,
                          levels = c('animal', 'print', 'dot',
'solid', 'null', 'patchwork', 'striped', 'geometric', 'plaid',
'leopard', 'floral', 'character', 'splice'),
                          labels = c(0,1,2,3,4,5,6,7,8,9,10,11,12))

attribset_$Recommendation = sapply(attribset_$Recommendation, factor)

# count of missing values in attribset_ dataset
colSums(is.na(attribset_))

         Style          Price         Rating           Size
Season
             0              2              0              0
2
      NeckLine   SleeveLength      waiseline       Material
FabricType
             1              0              1              1
1
    Decoration   Pattern Type Recommendation
             1              1              0

# Create the function.
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

# fill missing Value with mode
attribset_$Price[is.na(attribset_$Price) ==TRUE] <-
getmode(attribset_$Price)
attribset_$Season[is.na(attribset_$Season) ==TRUE] <-
getmode(attribset_$Season)
attribset_$NeckLine[is.na(attribset_$NeckLine) ==TRUE] <-
getmode(attribset_$NeckLine)
attribset_$waiseline[is.na(attribset_$waiseline) ==TRUE] <-
getmode(attribset_$waiseline)
attribset_$Material[is.na(attribset_$Material) ==TRUE] <-
getmode(attribset_$Material)
attribset_$FabricType[is.na(attribset_$FabricType) ==TRUE] <-
getmode(attribset_$FabricType)
attribset_$Decoration[is.na(attribset_$Decoration) ==TRUE] <-
getmode(attribset_$Decoration)
attribset_$`Pattern Type`[is.na(attribset_$`Pattern Type`) ==TRUE] <-
getmode(attribset_$`Pattern Type`)

attribset_data <- data.frame(attribset_)
str(attribset_data)
```

```
'data.frame':   500 obs. of  13 variables:
 $ Style         : Factor w/ 12 levels "0","1","2","3",..: 1 2 3 4 5 6
2 7 8 6 ...
 $ Price         : Factor w/ 5 levels "0","1","2","3",..: 1 1 2 3 1 1
3 3 3 1 ...
 $ Rating        : num  4.6 0 0 4.6 4.5 0 0 0 0 0 ...
 $ Size          : Factor w/ 5 levels "0","1","2","3",..: 1 2 2 2 1 1
3 4 4 4 ...
 $ Season        : Factor w/ 4 levels "0","1","2","3": 1 1 2 3 1 1 1 2
3 1 ...
 $ NeckLine      : Factor w/ 16 levels "0","1","2","3",..: 1 1 1 1 1 2
1 1 2 2 ...
 $ SleeveLength  : Factor w/ 13 levels "0","1","2","3",..: 1 2 3 3 4 1
3 5 5 1 ...
 $ waiseline     : Factor w/ 5 levels "0","1","2","3",..: 1 2 2 2 2 1
3 2 1 2 ...
 $ Material      : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 1
6 3 6 7 ...
 $ FabricType    : Factor w/ 18 levels "0","1","2","3",..: 1 2 2 1 1 2
2 3 3 1 ...
 $ Decoration    : Factor w/ 24 levels "0","1","2","3",..: 1 1 2 3 4 2
2 5 6 2 ...
 $ Pattern.Type  : Factor w/ 13 levels "0","1","2","3",..: 1 1 2 2 3 2
4 5 4 5 ...
 $ Recommendation: Factor w/ 2 levels "1","0": 1 2 2 1 2 2 2 2 1 1 ...
```

Dresses dataset

```
# Update columns name in dresssale_ dataset

dresssale_ = rename(dresssale_,c('41314'='2/9/2013'))
dresssale_ = rename(dresssale_,c('41373'='4/9/2013'))
dresssale_ = rename(dresssale_,c('41434'='6/9/2013'))
dresssale_ = rename(dresssale_,c('41495'='8/9/2013'))
dresssale_ = rename(dresssale_,c('41556'='10/9/2013'))
dresssale_ = rename(dresssale_,c('41617'='12/9/2013'))
dresssale_ = rename(dresssale_,c('41315'='2/10/2013'))
dresssale_ = rename(dresssale_,c('41374'='4/10/2013'))
dresssale_ = rename(dresssale_,c('41435'='6/10/2013'))
dresssale_ = rename(dresssale_,c('40400'='8/10/2013'))
dresssale_ = rename(dresssale_,c('41557'='10/10/2013'))
dresssale_ = rename(dresssale_,c('41618'='12/10/2013'))

The following `from` values were not present in `x`: 41314
The following `from` values were not present in `x`: 41373
The following `from` values were not present in `x`: 41434
The following `from` values were not present in `x`: 41495
The following `from` values were not present in `x`: 41556
The following `from` values were not present in `x`: 41617
The following `from` values were not present in `x`: 41315
The following `from` values were not present in `x`: 41374
The following `from` values were not present in `x`: 41435
The following `from` values were not present in `x`: 40400
The following `from` values were not present in `x`: 41557
The following `from` values were not present in `x`: 41618

# Convert all variable types to numeric
dresssale_ <- as.data.frame(apply(dresssale_, 2, as.numeric))

# mean row
dresssale_ = as.matrix(dresssale_)
k <- which(is.na(dresssale_), arr.ind=TRUE)
dresssale_[k] <- rowMeans(dresssale_, na.rm=TRUE)[k[,1]]
dresssale_ = as.data.frame(dresssale_)

# sum all values on row on (total sales)
dresssale_$total_sales = rowSums(dresssale_)
head(dresssale_)

  29/8/2013 31/8/2013 2/9/2013 4/9/2013 6/9/2013 8/9/2013 10/9/2013
12/9/2013
1 2114      2274      2491     2660     2727     2887     2930
3119
2  151       275       570      750      813     1066     1164
1558
3    6         7         7        7        8        8        9
10
4 1005      1128      1326     1455     1507     1621     1637
1723
5  996      1175      1304     1396     1432     1559     1570
1638
6    4         5        11       13       13       13       16
18
  14/9/2013 16/9/2013 ... 26/9/2013 28/9/2013 30/9/2013 2/10/2013
4/10/2013
1 3204      3277      ... 3624      3706      3746      3795      3832

2 1756      1878      ... 2942      3258      3354      3475      3654

3   10        10      ...   11        11        11        11        11

4 1746      1783      ... 1892      1914      1924      1929      1941

5 1655      1681      ... 2156      2252      2312      2387      2459

6   19        20      ...   25        26        26        26        26

  6/10/2013 8/10/2013 10/10/2013 12/10/2013 total_sales
1 3897      3923      3985       4048       75979
2 3911      4024      4125       4277       52256
3   11        11        11         11         223
```

```
4 1952      1955      1959       1963       39691
5 2544      2614      2693       2736       44077
6   27        27        27         27         457
```

Marged data

```
merged_data <- data.frame(attribset_ ,dresssale_)
head(merged_data)

  Style Price Rating Size Season NeckLine SleeveLength waiseline
Material
1 0     0     4.6    0    0      0        0            0         0

2 1     0     0.0    1    0      0        1            1         1

3 2     1     0.0    1    1      0        2            1         2

4 3     2     4.6    1    2      0        2            1         3

5 4     0     4.5    0    0      0        3            1         4

6 5     0     0.0    0    0      1        0            0         0

  FabricType ... X26.9.2013 X28.9.2013 X30.9.2013 X2.10.2013
X4.10.2013
1 0          ... 3624       3706       3746       3795       3832

2 1          ... 2942       3258       3354       3475       3654

3 1          ...   11         11         11         11         11

4 0          ... 1892       1914       1924       1929       1941

5 0          ... 2156       2252       2312       2387       2459

6 1          ...   25         26         26         26         26

  X6.10.2013 X8.10.2013 X10.10.2013 X12.10.2013 total_sales
1 3897       3923       3985        4048        75979
2 3911       4024       4125        4277        52256
3   11         11         11          11          223
4 1952       1955       1959        1963        39691
5 2544       2614       2693        2736        44077
6   27         27         27          27          457

str(merged_data)

'data.frame':   500 obs. of  37 variables:
 $ Style        : Factor w/ 12 levels "0","1","2","3",..: 1 2 3 4 5 6
2 7 8 6 ...
 $ Price        : Factor w/ 5 levels "0","1","2","3",..: 1 1 2 3 1 1
```

```
3 3 3 1 ...
 $ Rating        : num  4.6 0 0 4.6 4.5 0 0 0 0 0 ...
 $ Size          : Factor w/ 5 levels "0","1","2","3",..: 1 2 2 2 1 1
3 4 4 4 ...
 $ Season        : Factor w/ 4 levels "0","1","2","3": 1 1 2 3 1 1 1 2
3 1 ...
 $ NeckLine      : Factor w/ 16 levels "0","1","2","3",..: 1 1 1 1 1 2
1 1 2 2 ...
 $ SleeveLength  : Factor w/ 13 levels "0","1","2","3",..: 1 2 3 3 4 1
3 5 5 1 ...
 $ waiseline     : Factor w/ 5 levels "0","1","2","3",..: 1 2 2 2 2 1
3 2 1 2 ...
 $ Material      : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 1
6 3 6 7 ...
 $ FabricType    : Factor w/ 18 levels "0","1","2","3",..: 1 2 2 1 1 2
2 3 3 1 ...
 $ Decoration    : Factor w/ 24 levels "0","1","2","3",..: 1 1 2 3 4 2
2 5 6 2 ...
 $ Pattern.Type  : Factor w/ 13 levels "0","1","2","3",..: 1 1 2 2 3 2
4 5 4 5 ...
 $ Recommendation: Factor w/ 2 levels "1","0": 1 2 2 1 2 2 2 2 1 1 ...
 $ X29.8.2013    : num  2114 151 6 1005 996 ...
 $ X31.8.2013    : num  2274 275 7 1128 1175 ...
 $ X2.9.2013     : num  2491 570 7 1326 1304 ...
 $ X4.9.2013     : num  2660 750 7 1455 1396 ...
 $ X6.9.2013     : num  2727 813 8 1507 1432 ...
 $ X8.9.2013     : num  2887 1066 8 1621 1559 ...
 $ X10.9.2013    : num  2930 1164 9 1637 1570 ...
 $ X12.9.2013    : num  3119 1558 10 1723 1638 ...
 $ X14.9.2013    : num  3204 1756 10 1746 1655 ...
 $ X16.9.2013    : num  3277 1878 10 1783 1681 ...
 $ X18.9.2013    : num  3321 1985 10 1796 1743 ...
 $ X20.9.2013    : num  3386 2106 10 1812 1824 ...
 $ X22.9.2013    : num  3479 2454 11 1845 1919 ...
 $ X24.9.2013    : num  3554 2710 11 1878 2032 ...
 $ X26.9.2013    : num  3624 2942 11 1892 2156 ...
 $ X28.9.2013    : num  3706 3258 11 1914 2252 ...
 $ X30.9.2013    : num  3746 3354 11 1924 2312 ...
 $ X2.10.2013    : num  3795 3475 11 1929 2387 ...
 $ X4.10.2013    : num  3832 3654 11 1941 2459 ...
 $ X6.10.2013    : num  3897 3911 11 1952 2544 ...
 $ X8.10.2013    : num  3923 4024 11 1955 2614 ...
 $ X10.10.2013   : num  3985 4125 11 1959 2693 ...
 $ X12.10.2013   : num  4048 4277 11 1963 2736 ...
 $ total_sales   : num  75979 52256 223 39691 44077 ...

# spliting dataset
set.seed(100)

spl = sample.split(merged_data$Recommendation, SplitRatio = 0.7)
```

```
train = subset(merged_data, spl==TRUE)
test = subset(merged_data, spl==FALSE)

print(dim(train)); print(dim(test))

[1] 350  37
[1] 150  37
```

Classification - Predict recommendation

First model (Naive Bayes)

```
# naive bayes model
naive_model = naiveBayes(Recommendation ~.,data = train) # build model
confusionMatrix(train$Recommendation,predict(naive_model,train),positive = '1') # create confusion Matrix
print('---------------')
naive_predict = predict(naive_model,test) # predict test set
table(naive_predict,test$Recommendation) # create table

Confusion Matrix and Statistics

          Reference
Prediction   1   0
         1 106  41
         0  66 137

               Accuracy : 0.6943
                 95% CI : (0.6431, 0.7422)
    No Information Rate : 0.5086
    P-Value [Acc > NIR] : 1.352e-12

                  Kappa : 0.3869

 Mcnemar's Test P-Value : 0.02033

            Sensitivity : 0.6163
            Specificity : 0.7697
         Pos Pred Value : 0.7211
         Neg Pred Value : 0.6749
             Prevalence : 0.4914
         Detection Rate : 0.3029
   Detection Prevalence : 0.4200
      Balanced Accuracy : 0.6930

       'Positive' Class : 1


[1] "---------------"
```

```
naive_predict  1  0
            1 33 37
            0 30 50
```

Second model (Support Vector Machine)

```r
# Support vector machine
svm_model = svm(Recommendation ~.,train) # build model
confusionMatrix(train$Recommendation,predict(svm_model),positive =
'1')# create confusion Matrix
print('---------------')
svm_predict = predict(svm_model,test) # predict test set
table(svm_predict,test$Recommendation) # create table
```

```
Confusion Matrix and Statistics

          Reference
Prediction   1   0
         1   6 141
         0   0 203

               Accuracy : 0.5971
                 95% CI : (0.5437, 0.6489)
    No Information Rate : 0.9829
    P-Value [Acc > NIR] : 1

                  Kappa : 0.047

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.00000
            Specificity : 0.59012
         Pos Pred Value : 0.04082
         Neg Pred Value : 1.00000
             Prevalence : 0.01714
         Detection Rate : 0.01714
   Detection Prevalence : 0.42000
      Balanced Accuracy : 0.79506

       'Positive' Class : 1


[1] "---------------"


svm_predict  1  0
          1  0  2
          0 63 85
```

Third model (Random Forest)

```r
# Random Forest
randomForest_model = randomForest(x = train, y = train$Recommendation,ntree =800)# build model
confusionMatrix(train$Recommendation,predict(randomForest_model),positive = '1') # create confusion Matrix
print('---------------')
randomForest_predict = predict(randomForest_model,test) # predict test set
table(randomForest_predict,test$Recommendation )# create table
```

```
Confusion Matrix and Statistics

          Reference
Prediction   1   0
         1 147   0
         0   0 203
                                          
               Accuracy : 1          
                 95% CI : (0.9895, 1)
    No Information Rate : 0.58       
    P-Value [Acc > NIR] : < 2.2e-16  
                                          
                  Kappa : 1          
                                          
 Mcnemar's Test P-Value : NA         
                                          
            Sensitivity : 1.00       
            Specificity : 1.00       
         Pos Pred Value : 1.00       
         Neg Pred Value : 1.00       
             Prevalence : 0.42       
         Detection Rate : 0.42       
   Detection Prevalence : 0.42       
      Balanced Accuracy : 1.00       
                                          
       'Positive' Class : 1          
                                          

[1] "---------------"


randomForest_predict  1  0
                   1 63  0
                   0  0 87
```

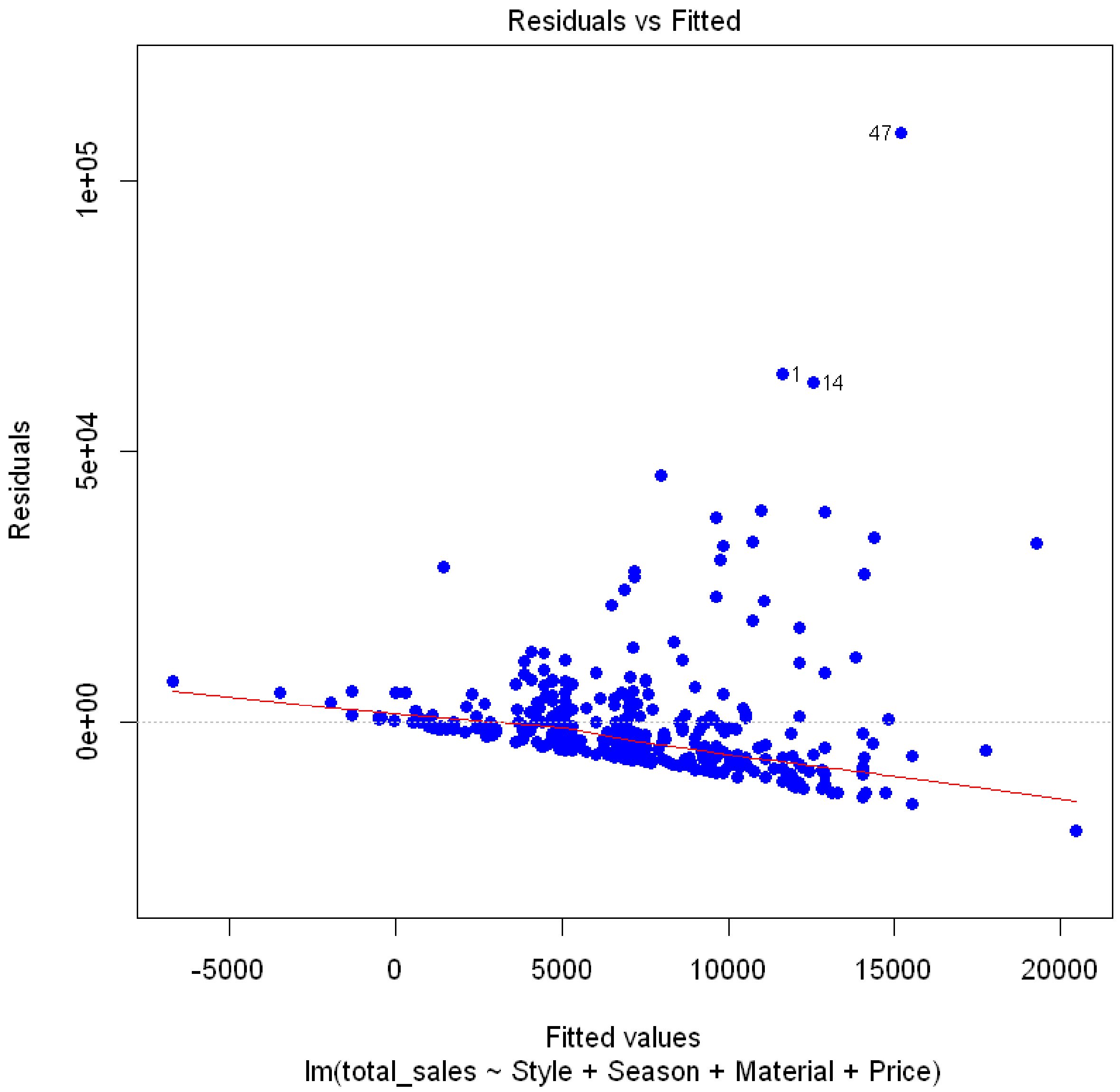
Regresstion model

Regression (total sales and (Style+Season+Material+Price))

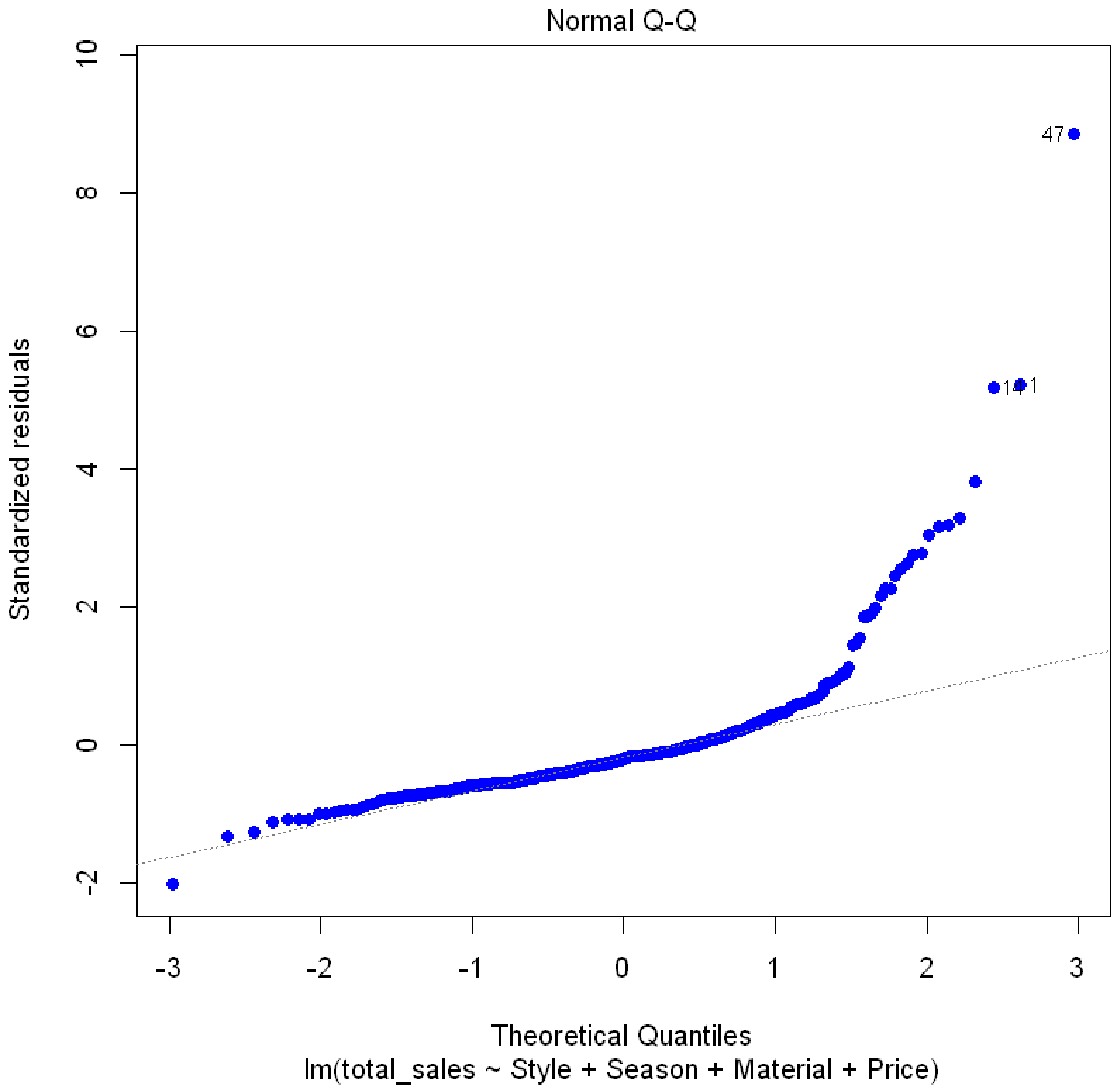
```
# regression (total sales and (Style+Season+Material+Price))
regressor_Sales = lm(formula = total_sales ~
Style+Season+Material+Price, data = train) # build model
summary(regressor_Sales) # print model summary
plot(regressor_Sales, pch = 16, col = "blue") # Plot the results
abline(regressor_Sales) # Add regression line
```

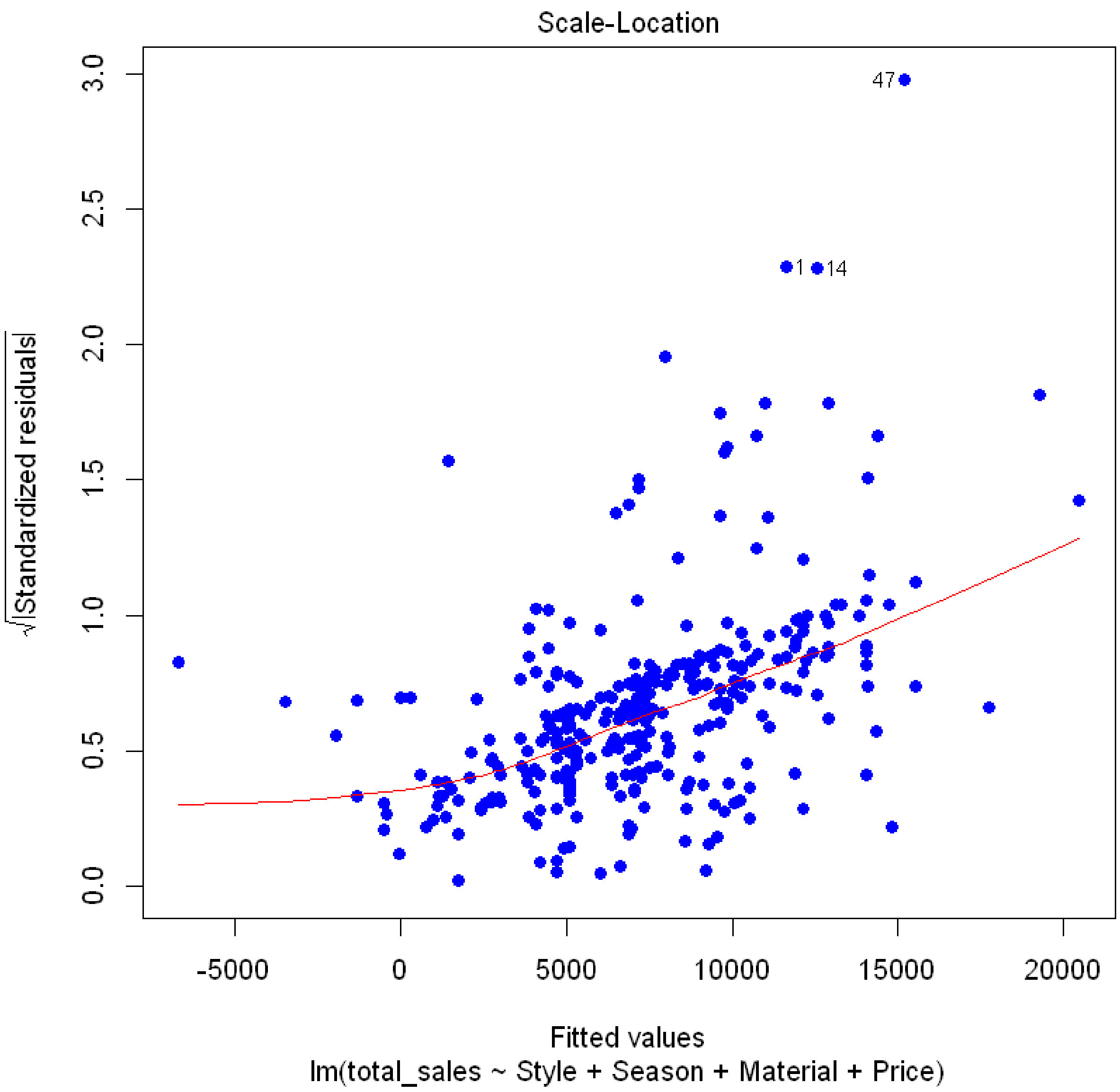
```
Call:
lm(formula = total_sales ~ Style + Season + Material + Price,
    data = train)

Residuals:
   Min      1Q Median      3Q     Max
-19944  -6145  -2112   1336 108668

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11642.9     2468.6   4.716 3.63e-06 ***
Style1       -4756.9     2019.4  -2.356   0.0191 *
Style2        2720.4     3733.0   0.729   0.4667
Style3       -2242.7     3960.1  -0.566   0.5716
Style4       -4555.6     3039.3  -1.499   0.1349
Style5       -7081.8     3740.8  -1.893   0.0593 .
Style6       -6932.0     6056.4  -1.145   0.2533
Style7      -11758.8    12815.6  -0.918   0.3596
Style8       -4119.9     3213.0  -1.282   0.2007
Style9       -3267.7     4069.5  -0.803   0.4226
Style11      -9315.9    12772.5  -0.729   0.4663
Season1       -873.4     2346.7  -0.372   0.7100
Season2       2392.6     1910.8   1.252   0.2115
Season3       -651.9     1953.2  -0.334   0.7388
Material1    12390.3     7521.3   1.647   0.1005
Material2      795.6     2245.2   0.354   0.7233
Material3    -2392.4     3317.6  -0.721   0.4714
Material4     3645.2     3142.3   1.160   0.2469
Material5    -2154.0     1885.9  -1.142   0.2543
Material6    -3599.0     5523.1  -0.652   0.5151
Material7    -2414.3     9136.3  -0.264   0.7918
Material8    -2431.6     9074.4  -0.268   0.7889
Material9     1963.7     7593.4   0.259   0.7961
Material10   -6027.6     5060.2  -1.191   0.2345
Material11    -592.8     7479.6  -0.079   0.9369
Material12   -1596.6     4288.7  -0.372   0.7099
Material13   -2840.7     7505.8  -0.378   0.7053
Material14   -7007.3    12938.8  -0.542   0.5885
```

```
Material15   -2251.5    12852.1  -0.175   0.8611
Material16   -6536.3    12741.0  -0.513   0.6083
Material17   -5259.4     9114.9  -0.577   0.5644
Material19   -1822.2    12776.2  -0.143   0.8867
Material20   -7964.3    12800.2  -0.622   0.5343
Material23   -3010.3     9273.1  -0.325   0.7457
Price1       -1941.8     3912.9  -0.496   0.6201
Price2         371.2     1628.8   0.228   0.8199
Price3       -4102.3     3571.7  -1.149   0.2516
Price4       -8178.1     4050.9  -2.019   0.0444 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12560 on 312 degrees of freedom
Multiple R-squared:  0.09177,	Adjusted R-squared:  -0.01594 
F-statistic: 0.852 on 37 and 312 DF,  p-value: 0.7162

Warning message:
"not plotting observations with leverage one:
  8, 68, 153, 162, 202, 257, 271"
```

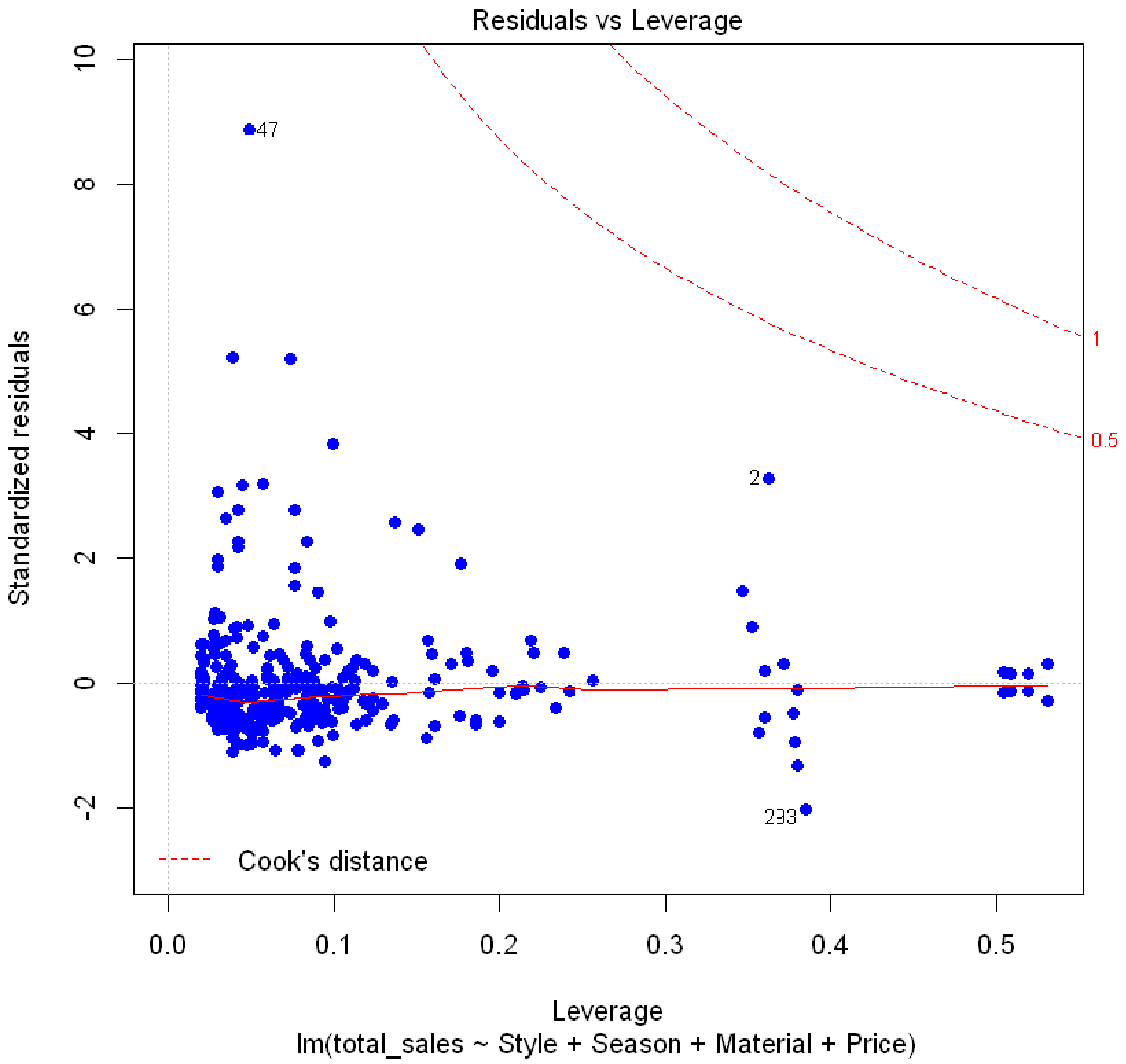


```
Warning message:
"not plotting observations with leverage one:
  8, 68, 153, 162, 202, 257, 271"
```


Normal Q-Q
Standardized residuals
10
8
6
4
2
0
-2
47
141
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(total_sales ~ Style + Season + Material + Price)



```
Warning message in abline(regressor_Sales):
"only using the first two of 38 regression coefficients"
```

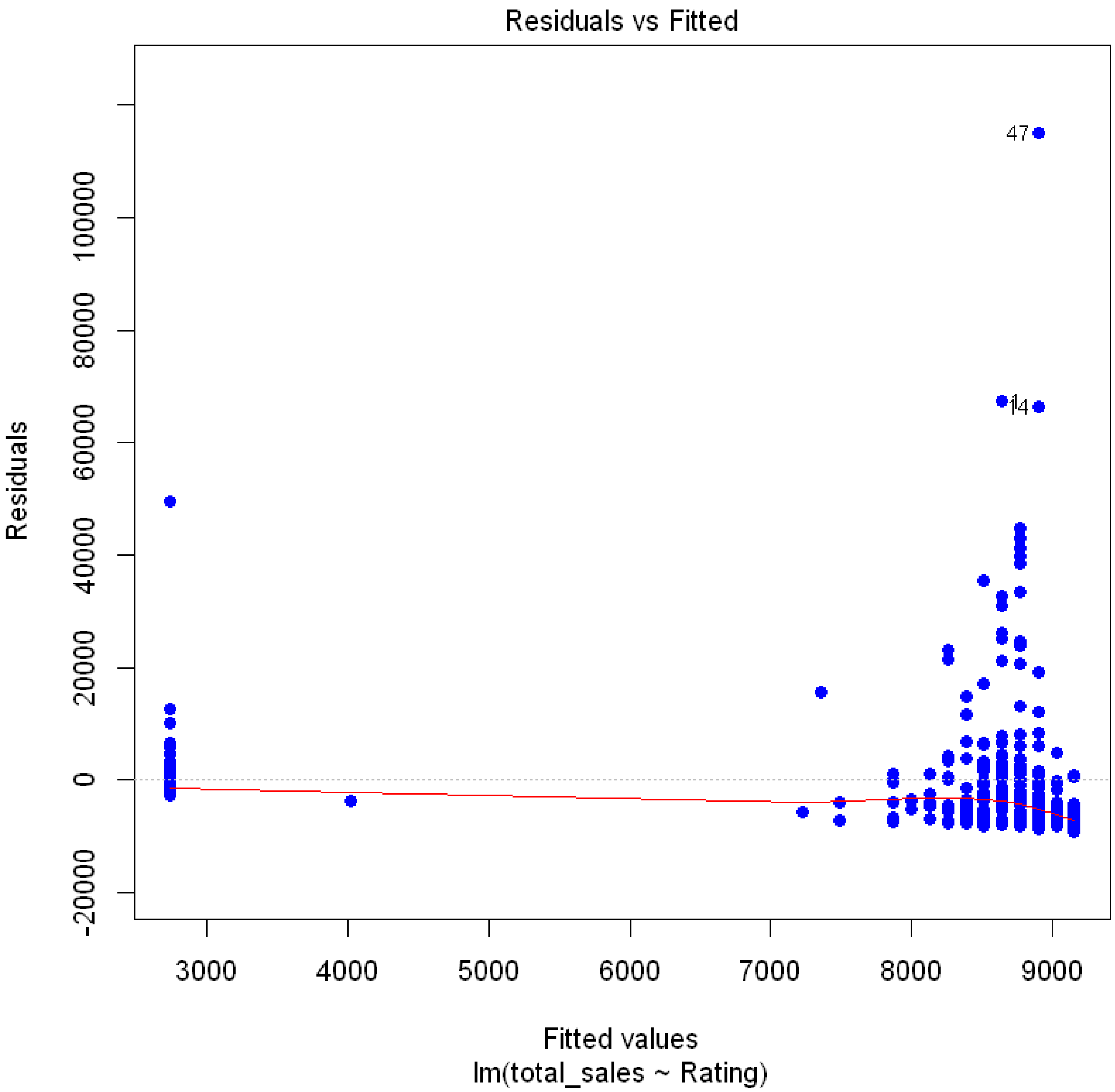


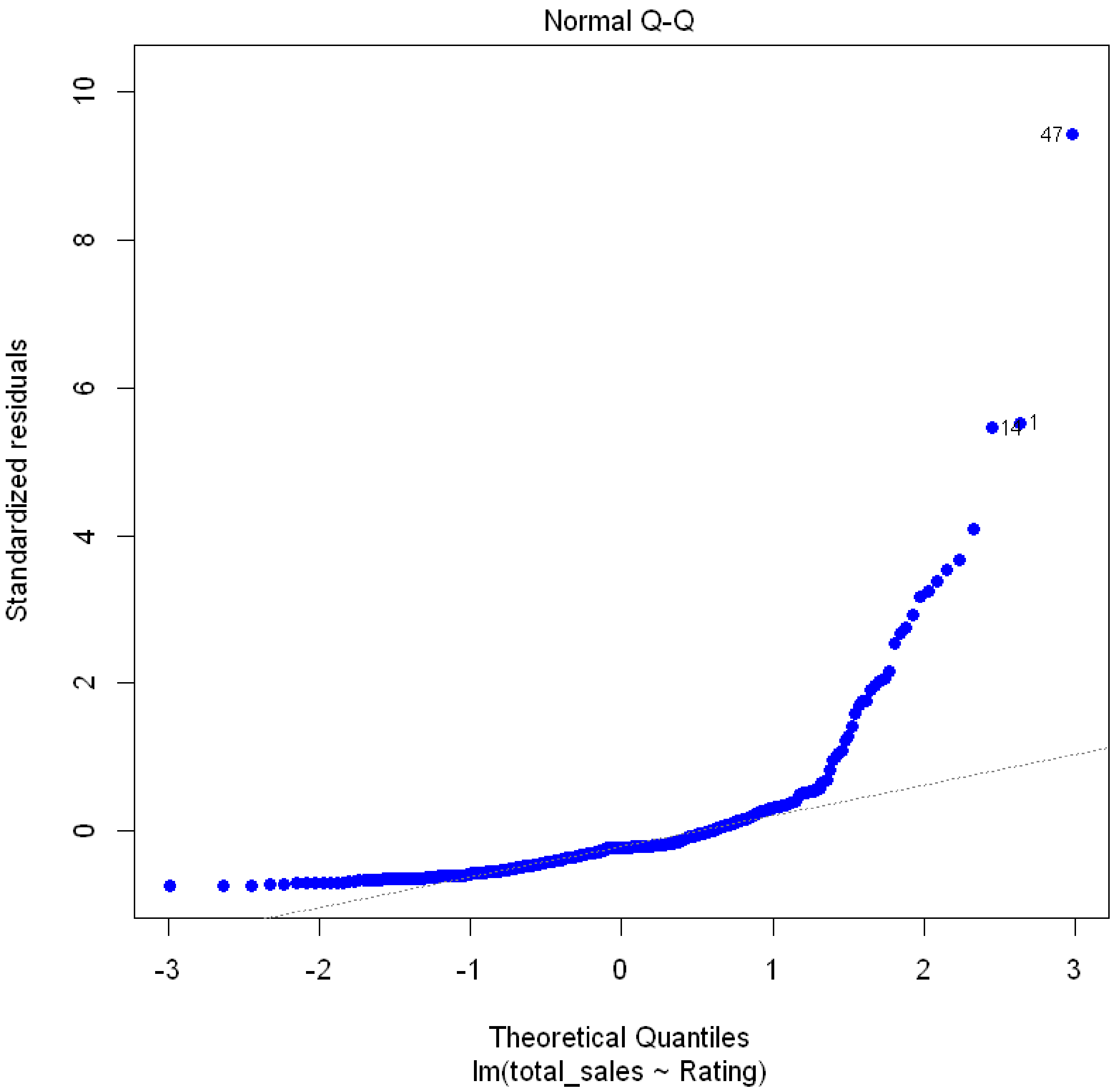
Regression (total sales and Rating)

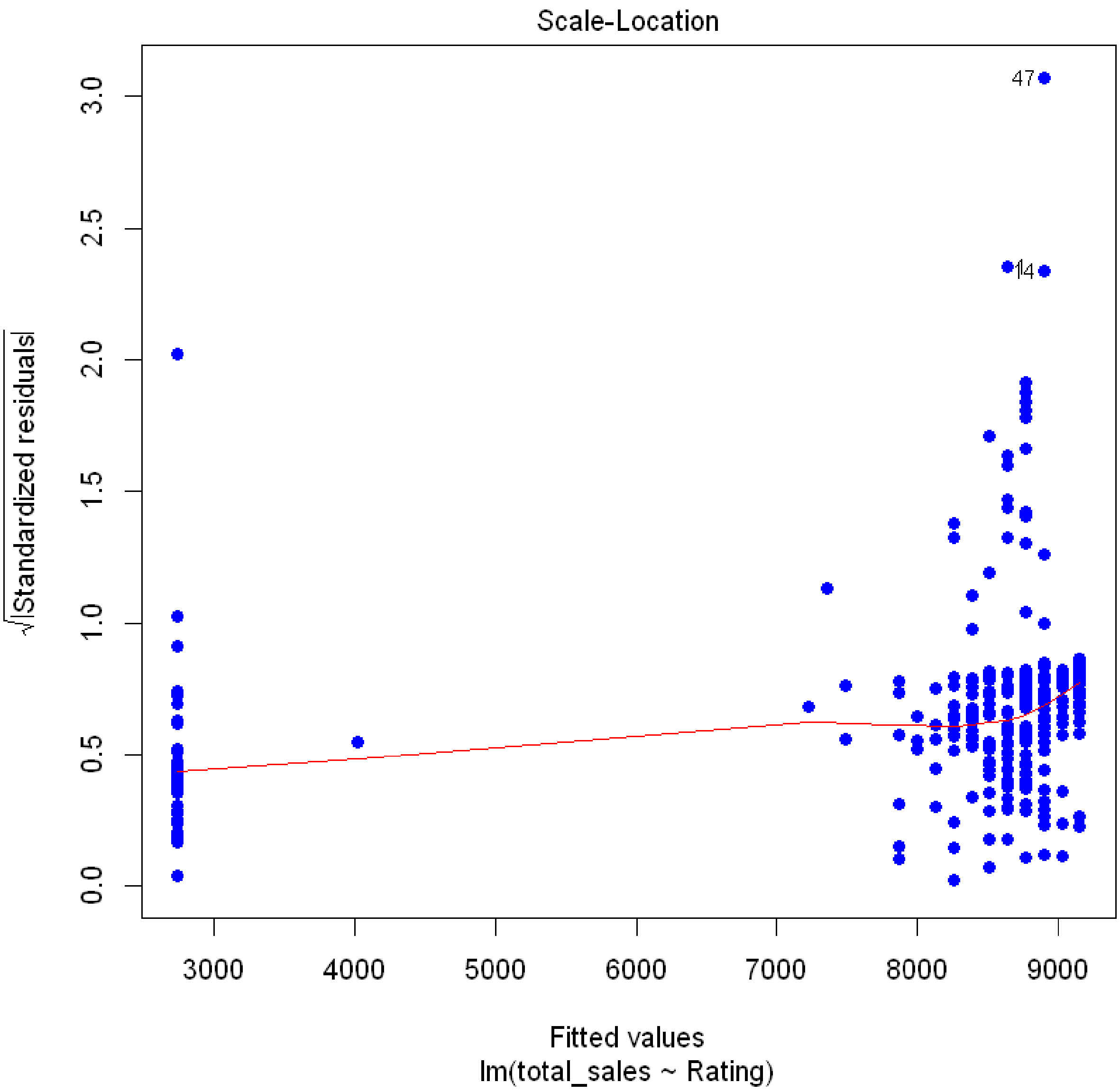
```
# regression (total sales and Rating)
regressor_Rating = lm(formula = total_sales ~ Rating, data = train) # build model
summary(regressor_Rating) # print model summary
plot(regressor_Rating, pch = 16, col = "blue") # Plot the results
abline(regressor_Rating) # Add regression line
```

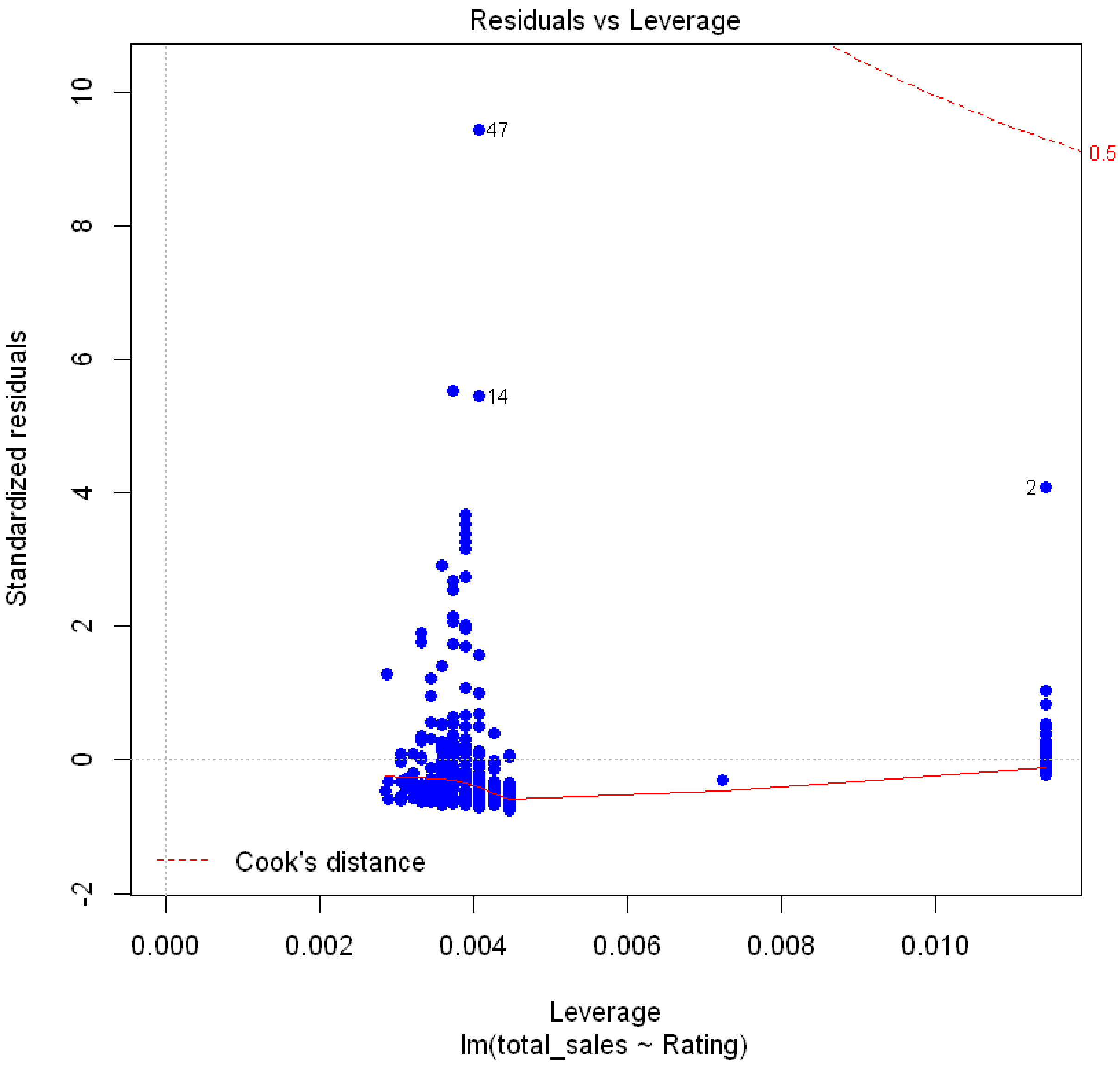
```
Call:
lm(formula = total_sales ~ Rating, data = train)
```

```
Residuals:
   Min     1Q Median     3Q    Max
 -9076  -6020  -2686    812 114971

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2742.8     1305.0   2.102   0.0363 *
Rating        1282.6      323.7   3.962 9.02e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12210 on 348 degrees of freedom
Multiple R-squared:  0.04316,    Adjusted R-squared:  0.04041
F-statistic:  15.7 on 1 and 348 DF,  p-value: 9.022e-05
```


Residuals vs Fitted
Residuals
-20000
0
20000
40000
60000
80000
100000
3000
4000
5000
6000
7000
8000
9000
Fitted values
lm(total_sales ~ Rating)
47
14


Normal Q-Q
Standardized residuals
0
2
4
6
8
10
-3
-2
-1
0
1
2
3
47
14
1
Theoretical Quantiles
lm(total_sales ~ Rating)


Scale-Location
47
14
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3000
4000
5000
6000
7000
8000
9000
Fitted values
lm(total_sales ~ Rating)



```
# evaluation
original = test$total_sales
pred = predict(regressor_Rating,test)
predicted = pred
d = original-predicted

mse = mean((d)^2) # MSE
mae = mean(abs(d)) # MAE
rmse = sqrt(mse) # RMSE
R2 = 1-(sum((d)^2)/sum((original-mean(original))^2)) # R^2
```

```r
cat(" MAE:", mae, "\n", "MSE:", mse, "\n", "RMSE:", rmse, "\n", "R-squared:", R2)
```

```
 MAE: 7784.569 
 MSE: 274959077 
 RMSE: 16581.89 
 R-squared: 0.04042806
```