Setting-Up R Environment

In [1]:
```
#write('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', file = "~/.Renviron", append = TRUE)
```

In [2]:
```
#Sys.which("make")
## "C:\\rtools40\\usr\\bin\\make.exe"
```

In [3]:
```
#install.packages("jsonlite", type = "source")
```

In [4]:
```
#install.packages("tidyverse",repos = "http://cran.us.r-project.org")
#library(tidyverse)
#install.packages("readxl",repos = "http://cran.us.r-project.org")
#library(readxl)
```

In [5]:
```
#install.packages("plyr")
#library("plyr")
```

In [6]:
```
#install.packages('caTools')
#library(caTools)
```

In [7]:
```
#install.packages('e1071', dependencies=TRUE)
```

In [8]:
```
#install.packages('caret', dependencies = TRUE)
```

Checking R Version

In [9]:
```
version
```

```
               _
platform       x86_64-w64-mingw32
arch           x86_64
os             mingw32
system         x86_64, mingw32
status
major          4
minor          1.3
year           2022
month          03
day            10
svn rev        81868
language       R
version.string R version 4.1.3 (2022-03-10)
nickname       One Push-Up
```

Import necessary libraries

In [10]:
```
library(readxl) # to read excel
library(plyr)
library(caTools)

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

```
Loading required package: ggplot2

Loading required package: lattice

randomForest 4.7-1.1

Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'


The following object is masked from 'package:ggplot2':

    margin
```

Data Understanding

In [11]:
```
# 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

In [12]:
```
# 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
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'
```

In [13]:
```
# factoring

attribset_$Style = factor(attribset_$Style,
                          levels = c('Sexy', 'Casual', 'vintage', 'Brief', 'cute', 'bohemian', 'Novelty', 'Flare
                          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', 'turndowncol
                          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', 'halfs
                          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', 'nylor
                          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', 'flar
                          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', 'hollc
                          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', 'geometr
                          labels = c(0,1,2,3,4,5,6,7,8,9,10,11,12))

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

In [14]:
```
# count of missing values in attribset_ dataset
colSums(is.na(attribset_))
```

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

In [15]:
```
# 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`)
```

In [16]:
```
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

In [17]:
```
# 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'))
```

In [18]:
```r
# 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_)
```

```
Warning message in apply(dresssale_, 2, as.numeric):
"NAs introduced by coercion"
Warning message in apply(dresssale_, 2, as.numeric):
"NAs introduced by coercion"
Warning message in apply(dresssale_, 2, as.numeric):
"NAs introduced by coercion"
Warning message in apply(dresssale_, 2, as.numeric):
"NAs introduced by coercion"
Warning message in apply(dresssale_, 2, as.numeric):
"NAs introduced by coercion"
Warning message in apply(dresssale_, 2, as.numeric):
"NAs introduced by coercion"
```

In [19]:
```
# sum all values on row on (total sales)
dresssale_$total_sales = rowSums(dresssale_)
head(dresssale_)
```

A data.frame: 6 × 24

	29/8/2013	31/8/2013	2/9/2013	4/9/2013	6/9/2013	8/9/2013	10/9/2013	12/9/2013	14/9/2013	16/9/2013	⋯	26/9/2013	28/9/2013	30/9/201
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	⋯	<dbl>	<dbl>	<dbl
1	2114	2274	2491	2660	2727	2887	2930	3119	3204	3277	⋯	3624	3706	374
2	151	275	570	750	813	1066	1164	1558	1756	1878	⋯	2942	3258	335
3	6	7	7	7	8	8	9	10	10	10	⋯	11	11	1
4	1005	1128	1326	1455	1507	1621	1637	1723	1746	1783	⋯	1892	1914	192
5	996	1175	1304	1396	1432	1559	1570	1638	1655	1681	⋯	2156	2252	231
6	4	5	11	13	13	13	16	18	19	20	⋯	25	26	2

Marged data

In [20]:
```
merged_data <- data.frame(attribset_ ,dresssale_)
head(merged_data)
```

A data.frame: 6 × 37

	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	Material	FabricType	⋯	X26.9.2013	X28.9.2013	X30.9.2013	X
	<fct>	<fct>	<dbl>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	⋯	<dbl>	<dbl>	<dbl>	
1	0	0	4.6	0	0	0	0	0	0	0	⋯	3624	3706	3746	
2	1	0	0.0	1	0	0	1	1	1	1	⋯	2942	3258	3354	
3	2	1	0.0	1	1	0	2	1	2	1	⋯	11	11	11	
4	3	2	4.6	1	2	0	2	1	3	0	⋯	1892	1914	1924	
5	4	0	4.5	0	0	0	3	1	4	0	⋯	2156	2252	2312	
6	5	0	0.0	0	0	1	0	0	0	1	⋯	25	26	26	

In [21]:
```
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 ...
```

In [22]:
```
# 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)

In [23]:
```
# 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  67 136

               Accuracy : 0.6914
                 95% CI : (0.6401, 0.7394)
    No Information Rate : 0.5057
    P-Value [Acc > NIR] : 1.409e-12

                  Kappa : 0.3817

 Mcnemar's Test P-Value : 0.01614

            Sensitivity : 0.6127
            Specificity : 0.7684
         Pos Pred Value : 0.7211
         Neg Pred Value : 0.6700
             Prevalence : 0.4943
         Detection Rate : 0.3029
   Detection Prevalence : 0.4200
      Balanced Accuracy : 0.6905

       'Positive' Class : 1


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


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

Second model (Support Vector Machine)

In [24]:
```
# 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)

In [25]:
```
# 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
```

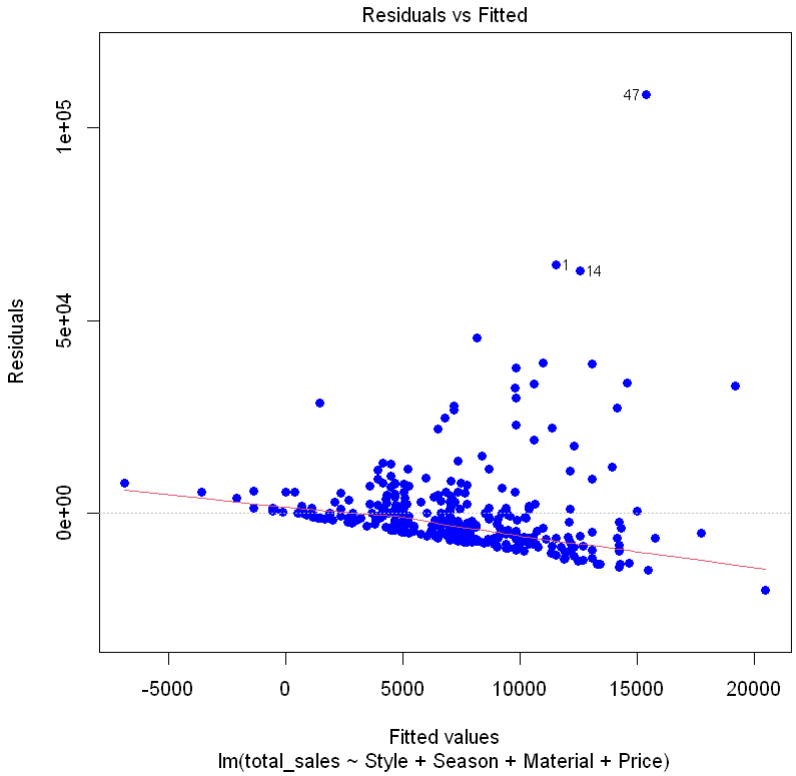
Regresstion model

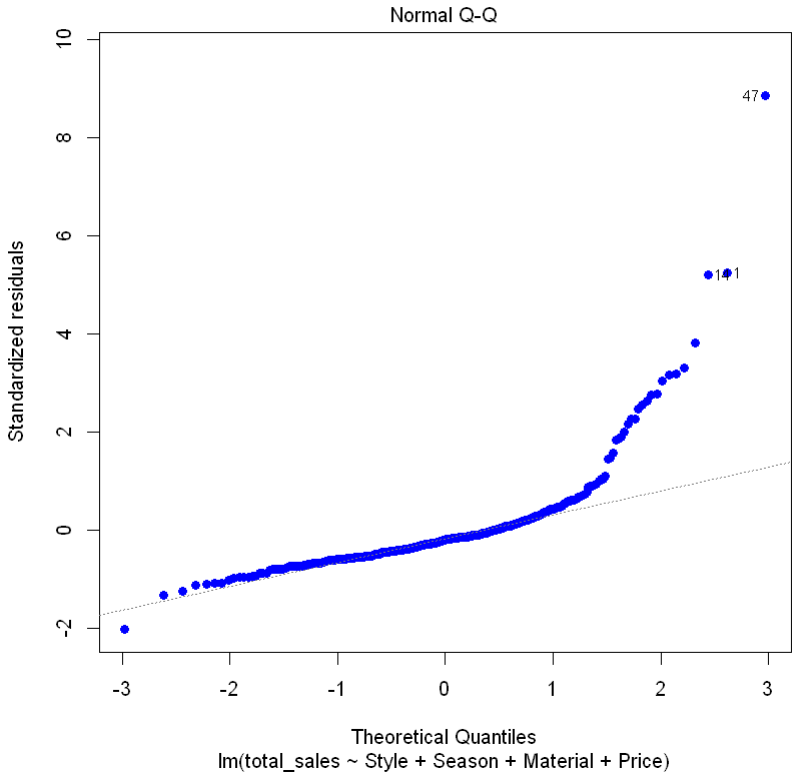
Regression (total sales and (Style+Season+Material+Price))

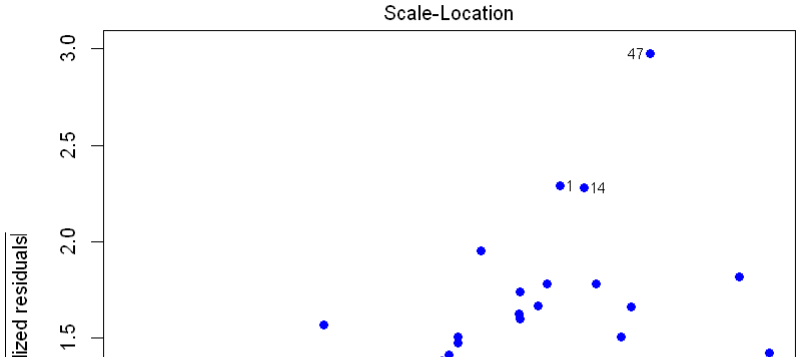
In [26]:
```
# regression (total sales and (Style+Season+Material+Price))
regressor_Sales = lm(formula = total_sales ~ Style+Season+Material+Price, data = train) # build model
summary(regressor_Sales) # print model summary
plot(regressor_Sales, pch = 16, col = "blue") # Plot the results
abline(regressor_Sales) # Add regression line
```

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

Residuals:
   Min     1Q Median     3Q    Max
-19936  -6113  -2230   1381 108508

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11521.7     2460.7   4.682 4.24e-06 ***
Style1       -4739.3     2018.1  -2.348   0.0195 *
Style2        2786.7     3731.1   0.747   0.4557
Style3       -2293.6     3957.3  -0.580   0.5626
Style4       -4514.3     3037.8  -1.486   0.1383
Style5       -7057.2     3738.5  -1.888   0.0600 .
Style6       -6905.0     6052.5  -1.141   0.2548
Style7      -11965.4    12808.3  -0.934   0.3509
Style8       -4017.0     3208.8  -1.252   0.2116
Style9       -3254.6     4066.9  -0.800   0.4242
Style11      -9220.8    12763.8  -0.722   0.4706
Season1       -761.5     2339.6  -0.325   0.7450
Season2       2694.3     1915.0   1.407   0.1604
Season3       -527.7     1944.7  -0.271   0.7863
Material1    12382.7     7516.0   1.648   0.1005
Material2      790.2     2243.6   0.352   0.7249
Material3    -2454.3     3316.7  -0.740   0.4599
Material4     3595.3     3141.2   1.145   0.2533
Material5    -2111.8     1884.6  -1.121   0.2633
Material6    -3568.7     5518.6  -0.647   0.5183
Material7    -2426.9     9130.3  -0.266   0.7906
Material8    -2429.2     9068.5  -0.268   0.7890
Material9     2010.3     7588.1   0.265   0.7912
Material10   -6025.8     5056.8  -1.192   0.2343
Material11    -637.5     7475.0  -0.085   0.9321
Material12   -1602.8     4285.8  -0.374   0.7087
```

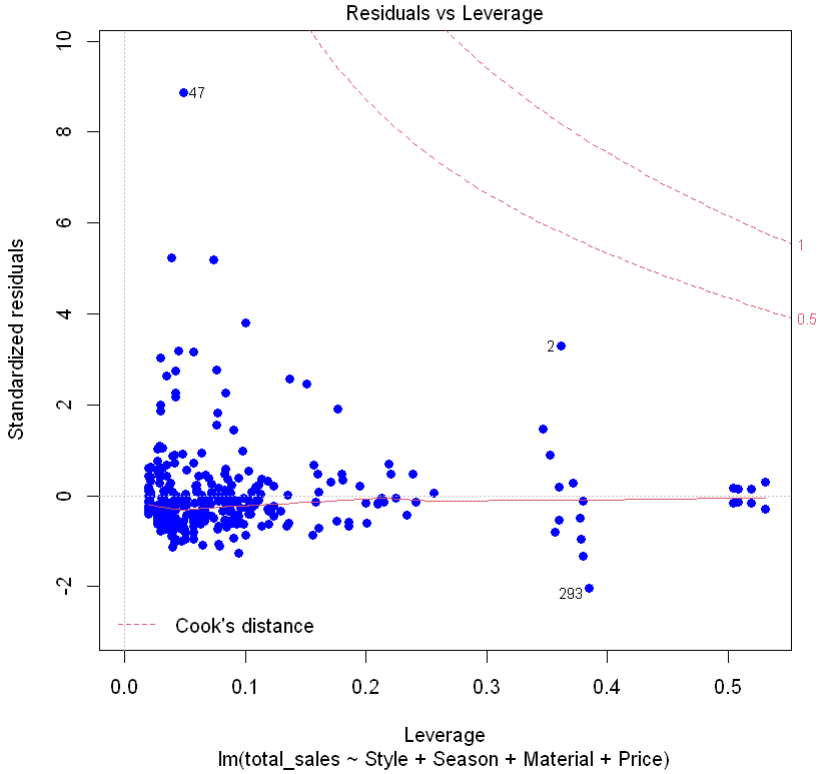
```
Material13   -2853.4     7500.8  -0.380   0.7039
Material14   -7213.0    12932.0  -0.558   0.5774
Material15   -2242.1    12843.7  -0.175   0.8615
Material16   -6416.6    12731.2  -0.504   0.6146
Material17   -5257.7     9108.9  -0.577   0.5642
Material19   -1842.8    12767.8  -0.144   0.8853
Material20   -7843.1    12790.6  -0.613   0.5402
Material23   -3215.7     9269.8  -0.347   0.7289
Price1       -2030.6     3912.1  -0.519   0.6041
Price2         355.2     1627.8   0.218   0.8274
Price3       -4172.0     3571.0  -1.168   0.2436
Price4       -8335.1     4055.2  -2.055   0.0407 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12550 on 312 degrees of freedom
Multiple R-squared:  0.09296,   Adjusted R-squared:  -0.01461
F-statistic: 0.8642 on 37 and 312 DF,  p-value: 0.6971


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


Residuals vs Fitted
Residuals
1e+05
5e+04
0e+00
47
1
14
-5000
0
5000
10000
15000
20000
Fitted values
lm(total_sales ~ Style + Season + Material + Price)


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



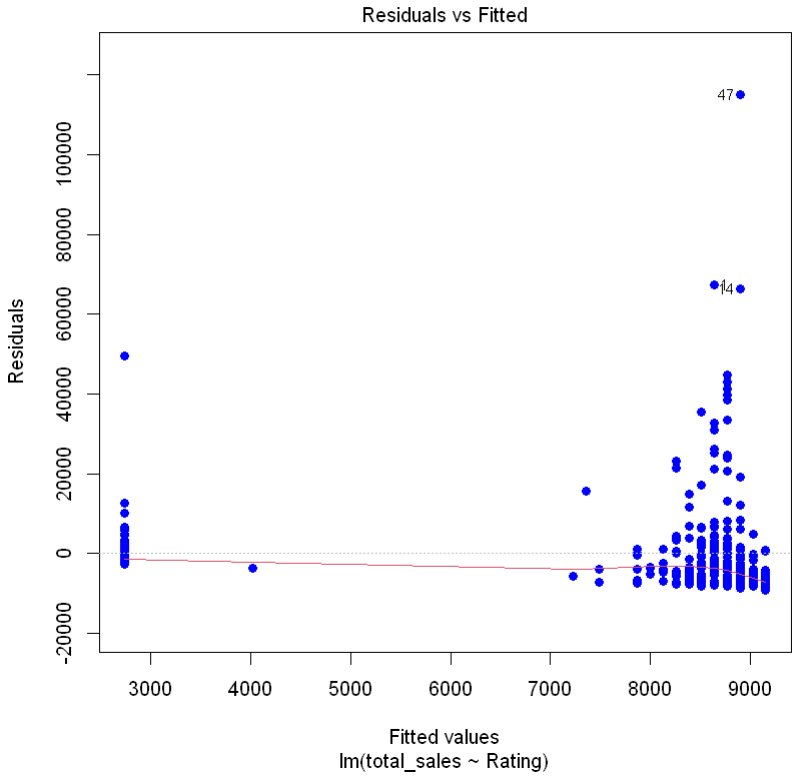

```
Warning message in abline(regressor_Sales):
"only using the first two of 38 regression coefficients"
```

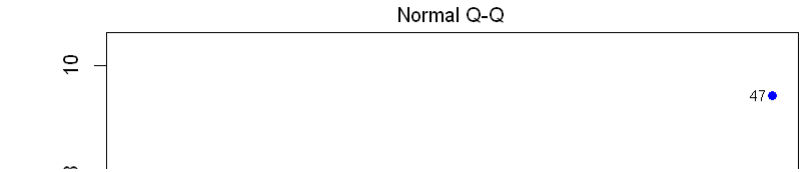


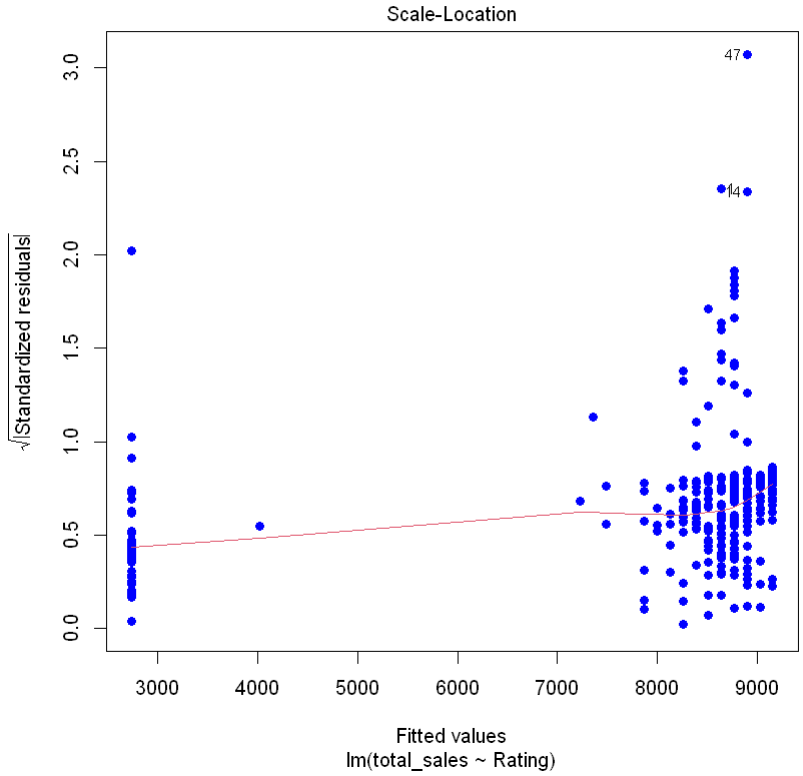
Regression (total sales and Rating)

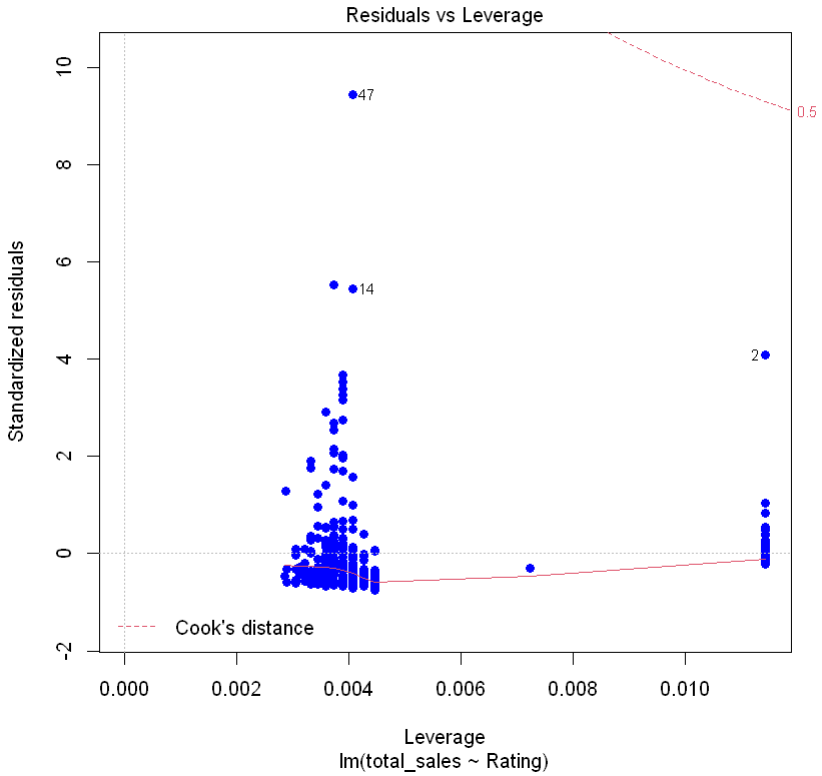
In [27]:
```
# regression (total sales and Rating)
regressor_Rating = lm(formula = total_sales ~ Rating, data = train) # build model
summary(regressor_Rating) # print model summary
plot(regressor_Rating, pch = 16, col = "blue") # Plot the results
abline(regressor_Rating) # Add regression line
```

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


Normal Q-Q
10
47


Scale-Location
√|Standardized residuals|
47
14
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)


Residuals vs Leverage
Standardized residuals
Leverage
lm(total_sales ~ Rating)
Cook's distance
0.5
47
14
2
-2
0
2
4
6
8
10
0.000
0.002
0.004
0.006
0.008
0.010

In [28]:
```
# 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

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
```