Analysis of sales report of a clothes manufacturer

R Project

```r
install.packages("pscl", repos = "https://cran.rstudio.com")
```

```
## Installing package into 'C:/Users/HP/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

```
## package 'pscl' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\HP\AppData\Local\Temp\Rtmp6rplF7\downloaded_packages
```

```r
# import necessary libraries
library(readxl) # to read excel
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(caTools)
library(e1071)
library(caret)
```

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

```
## Loading required package: ggplot2
```

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(rmarkdown)
options(repos=structure(c(CRAN="http://cran.r-project.org")))
attrib1 = read_excel('E:\\Simpli Learn\\6.Data science with R\\R Projects\\1 Analysis of Sales Report o
dresssale1 = read_excel('Dress Sales.xlsx')
#remove Dress_ID column
attrib2 = attrib1[2:14]
dresssale2 = dresssale1[2:24]

# check the unique values for each columns
lapply(attrib2, unique)
```

```
## $Style
##  [1] "Sexy"     "Casual"   "vintage"  "Brief"    "cute"     "bohemian"
##  [7] "Novelty"  "Flare"    "party"    "sexy"     "work"     "OL"
## [13] "fashion"
##
## $Price
## [1] "Low"       "High"      "Average"   "Medium"    "very-high" "low"
## [7] "high"      NA
##
## $Rating
##  [1] 4.6 0.0 4.5 5.0 4.7 4.8 4.3 4.0 4.4 4.9 4.2 3.6 3.7 4.1 3.5 1.0 3.0
##
## $Size
## [1] "M"     "L"     "XL"    "free"  "S"     "small" "s"
##
## $Season
## [1] "Summer" "Automn" "Spring" "Winter" "spring" "winter" NA       "summer"
## [9] "Autumn"
##
## $NeckLine
##  [1] "o-neck"          "v-neck"          "boat-neck"       "peterpan-collor"
##  [5] "ruffled"         "turndowncollor"  "slash-neck"      "mandarin-collor"
##  [9] "open"            "sqare-collor"    "Sweetheart"      "sweetheart"
## [13] NA                "Scoop"           "halter"          "backless"
## [17] "bowneck"         "NULL"
##
## $SleeveLength
##  [1] "sleevless"      "Petal"          "full"           "butterfly"
##  [5] "short"          "threequarter"   "halfsleeve"     "cap-sleeves"
##  [9] "turndowncollor" "threequater"    "capsleeves"     "sleeveless"
## [13] "sleeevless"     "half"           "urndowncollor"  "thressqatar"
## [17] "NULL"           "sleveless"
##
## $waiseline
## [1] "empire"   "natural"  "null"     NA         "princess" "dropped"
##
## $Material
##  [1] "null"          "microfiber"    "polyster"      "silk"
##  [5] "chiffonfabric" "cotton"        "nylon"         "other"
##  [9] "milksilk"      "linen"         "rayon"         "lycra"
## [13] "mix"           "acrylic"       "spandex"       "lace"
## [17] "modal"         "cashmere"      NA              "viscos"
## [21] "knitting"      "sill"          "wool"          "model"
## [25] "shiffon"
##
## $FabricType
##  [1] "chiffon"    "null"       "broadcloth" "jersey"     "other"
##  [6] "batik"      "satin"      "flannael"   "worsted"    "woolen"
## [11] "poplin"     "dobby"      "knitting"   "flannel"    "tulle"
## [16] "sattin"     NA           "organza"    "lace"       "Corduroy"
## [21] "wollen"     "knitted"    "shiffon"    "terry"
##
## $Decoration
##  [1] "ruffles"    "null"       "embroidary" "bow"        "lace"
##  [6] "beading"    "sashes"     "hollowout"  "pockets"    "sequined"
## [11] "applique"   "button"     "Tiered"     "rivet"      "feathers"
## [16] "flowers"    "pearls"     NA           "pleat"      "crystal"
## [21] "ruched"     "draped"     "tassel"     "plain"      "none"
## [26] "cascading"
##
## $`Pattern Type`
##  [1] "animal"    "print"     "dot"       "solid"     "null"      "patchwork"
##  [7] "striped"   "geometric" "plaid"     "leopard"   "floral"    "character"
## [13] NA          "splice"    "leapord"   "none"
##
## $Recommendation
## [1] 1 0
```

```
# values checking
attrib2$Style<-mapvalues(attrib2$Style, c('sexy'), c('Sexy'))
attrib2$Price<-mapvalues(attrib2$Price, c('low','high'), c('Low','High'))
attrib2$Size<-mapvalues(attrib2$Size, c('s','small'), c('S','S'))
attrib2$Season<-mapvalues(attrib2$Season, c('spring','summer','Automn','winter'), c('Spring','Summer','
attrib2$NeckLine<-mapvalues(attrib2$NeckLine, c('sweetheart'), c('Sweetheart'))
attrib2$SleeveLength<-mapvalues(attrib2$SleeveLength, c('sleevless','sleeevless','sleveless','threequat
attrib2$FabricType<-mapvalues(attrib2$FabricType, c('shiffon','sattin','wollen','flannael','knitting'),
attrib2$Decoration<-mapvalues(attrib2$Decoration, c('embroidary','sequined','ruched','none'), c('embroi
attrib2$`Pattern Type`<-mapvalues(attrib2$`Pattern Type`, c('none','leapord'), c('null','leopard'))

# factoring
attrib2$Style <- factor(attrib2$Style,levels=unique(attrib2$Style))
table(attrib2$Style)
```

```
##
##     Sexy   Casual  vintage    Brief     cute bohemian  Novelty    Flare
##       76      232       25       18       45       24        8        2
##    party     work       OL  fashion
##       51       17        1        1
```

```
attrib2$Price <- factor(attrib2$Price,levels=unique(attrib2$Price))
summary(attrib2$Price)
```

```
##       Low      High   Average    Medium very-high      NA's
##       174        21       252        30        21         2
```

```r
attrib2$Size <- factor(attrib2$Size,levels=unique(attrib2$Size))
attrib2$Season <- factor(attrib2$Season,levels=unique(attrib2$Season))
#attrib2$Season <- factor(attrib2$Season,order = TRUE,levels=unique(attrib2$Season))
attrib2$NeckLine <- factor(attrib2$NeckLine,levels=unique(attrib2$NeckLine))
attrib2$SleeveLength <- factor(attrib2$SleeveLength,levels=unique(attrib2$SleeveLength))
attrib2$waiseline <- factor(attrib2$waiseline,levels=unique(attrib2$waiseline))
attrib2$Material <- factor(attrib2$Material,levels=unique(attrib2$Material))
attrib2$FabricType <- factor(attrib2$FabricType,levels=unique(attrib2$FabricType))
attrib2$Decoration <- factor(attrib2$Decoration,levels=unique(attrib2$Decoration))
attrib2$`Pattern Type` <- factor(attrib2$`Pattern Type`,levels=unique(attrib2$`Pattern Type`))
attrib2$Recommendation <- sapply(attrib2$Recommendation, factor)

# count of missing values in attrib2 dataset
colSums(is.na(attrib2))
```

```
##          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
```

```r
# fill missing Value with mode # Categorical data. So, no use of mean & median. So, use Mode. # The mod
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
# fill missing Value with mode
attrib2$Price[is.na(attrib2$Price) ==TRUE] <- getmode(attrib2$Price)
attrib2$Season[is.na(attrib2$Season) ==TRUE] <- getmode(attrib2$Season)
attrib2$NeckLine[is.na(attrib2$NeckLine) ==TRUE] <- getmode(attrib2$NeckLine)
attrib2$waiseline[is.na(attrib2$waiseline) ==TRUE] <- getmode(attrib2$waiseline)
attrib2$Material[is.na(attrib2$Material) ==TRUE] <- getmode(attrib2$Material)
attrib2$FabricType[is.na(attrib2$FabricType) ==TRUE] <- getmode(attrib2$FabricType)
attrib2$Decoration[is.na(attrib2$Decoration) ==TRUE] <- getmode(attrib2$Decoration)
attrib2$`Pattern Type`[is.na(attrib2$`Pattern Type`) ==TRUE] <- getmode(attrib2$`Pattern Type`)


attrib2data <- data.frame(attrib2)
str(attrib2data)
```

```
## 'data.frame':    500 obs. of  13 variables:
##  $ Style         : Factor w/ 12 levels "Sexy","Casual",..: 1 2 3 4 5 6 2 7 8 6 ...
##  $ Price         : Factor w/ 5 levels "Low","High","Average",..: 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 "M","L","XL","free",..: 1 2 2 2 1 1 3 4 4 4 ...
##  $ Season        : Factor w/ 4 levels "Summer","Autumn",..: 1 1 2 3 1 1 1 2 3 1 ...
##  $ NeckLine      : Factor w/ 16 levels "o-neck","v-neck",..: 1 1 1 1 1 2 1 1 2 2 ...
##  $ SleeveLength  : Factor w/ 13 levels "sleeveless","Petal",..: 1 2 3 3 4 1 3 5 5 1 ...
##  $ waiseline     : Factor w/ 5 levels "empire","natural",..: 1 2 2 2 2 1 3 2 1 2 ...
##  $ Material      : Factor w/ 24 levels "null","microfiber",..: 1 2 3 4 5 1 6 3 6 7 ...
##  $ FabricType    : Factor w/ 18 levels "chiffon","null",..: 1 2 2 1 1 2 2 3 3 1 ...
##  $ Decoration    : Factor w/ 24 levels "ruffles","null",..: 1 1 2 3 4 2 2 5 6 2 ...
##  $ Pattern.Type  : Factor w/ 13 levels "animal","print",..: 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
head(dresssale2)
```

```
## # A tibble: 6 x 23
##   `29/8/2013` `31/8/2013` `41314` `41373` `41434` `41495` `41556` `41617`
##         <dbl>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 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
## # ... with 15 more variables: 14/9/2013 <chr>, 16/9/2013 <chr>,
## #   18/9/2013 <chr>, 20/9/2013 <chr>, 22/9/2013 <chr>, 24/9/2013 <dbl>,
## #   26/9/2013 <dbl>, 28/9/2013 <dbl>, 30/9/2013 <dbl>, 41315 <dbl>,
## #   41374 <dbl>, 41435 <dbl>, 40400 <dbl>, 41557 <dbl>, 41618 <dbl>
```

```
names(dresssale2)
```

```
##  [1] "29/8/2013" "31/8/2013" "41314"     "41373"     "41434"     "41495"
##  [7] "41556"     "41617"     "14/9/2013" "16/9/2013" "18/9/2013" "20/9/2013"
## [13] "22/9/2013" "24/9/2013" "26/9/2013" "28/9/2013" "30/9/2013" "41315"
## [19] "41374"     "41435"     "40400"     "41557"     "41618"
```

```
# Update columns name in dresssale2 dataset
# Change any column names you want to, all at once
colnames(dresssale2)[colnames(dresssale2) %in% c("41314","41373","41434","41495","41556","41617","41315
# names(dresssale2) <- c("29/8/2013", "31/8/2013", "2/9/2013",      "41373"  ,   "41434" ,    "41495"  ,
#                        "14/9/2013", "16/9/2013" ,"18/9/2013", "20/9/2013", "22/9/2013", "24/9/2013", "
#                        "30/9/2013", "41315",     "41374",     "41435",     "40400"    , "41557" ,    "

head(dresssale2)
```

```
## # A tibble: 6 x 23
##   `29/8/2013` `31/8/2013` `2/9/2013` `4/9/2013` `6/9/2013` `8/9/2013`
##         <dbl>       <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1        2114        2274       2491       2660       2727       2887
## 2         151         275        570        750        813       1066
## 3           6           7          7          7          8          8
## 4        1005        1128       1326       1455       1507       1621
## 5         996        1175       1304       1396       1432       1559
## 6           4           5         11         13         13         13
## # ... with 17 more variables: 10/9/2013 <dbl>, 12/9/2013 <chr>,
## #   14/9/2013 <chr>, 16/9/2013 <chr>, 18/9/2013 <chr>, 20/9/2013 <chr>,
## #   22/9/2013 <chr>, 24/9/2013 <dbl>, 26/9/2013 <dbl>, 28/9/2013 <dbl>,
## #   30/9/2013 <dbl>, 2/10/2013 <dbl>, 4/10/2013 <dbl>, 6/10/2013 <dbl>,
## #   8/10/2013 <dbl>, 10/10/2013 <dbl>, 12/10/2013 <dbl>
```

```
# Convert all variable types to numeric
dresssale2 <- as.data.frame(apply(dresssale2, 2, as.numeric))
```

```
## Warning in apply(dresssale2, 2, as.numeric): NAs introduced by coercion

## Warning in apply(dresssale2, 2, as.numeric): NAs introduced by coercion

## Warning in apply(dresssale2, 2, as.numeric): NAs introduced by coercion
## Warning in apply(dresssale2, 2, as.numeric): NAs introduced by coercion

## Warning in apply(dresssale2, 2, as.numeric): NAs introduced by coercion

## Warning in apply(dresssale2, 2, as.numeric): NAs introduced by coercion
```

```
# mean row
dresssale2 = as.matrix(dresssale2)
k <- which(is.na(dresssale2), arr.ind=TRUE)
dresssale2[k] <- rowMeans(dresssale2, na.rm=TRUE)[k[,1]]
dresssale2 = as.data.frame(dresssale2)

# sum all values on row on (total sales column)-New column created
dresssale2$total_sales = rowSums(dresssale2)
head(dresssale2)
```

```
##   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 18/9/2013 20/9/2013 22/9/2013 24/9/2013 26/9/2013
## 1      3204      3277      3321      3386      3479      3554      3624
## 2      1756      1878      1985      2106      2454      2710      2942
## 3        10        10        10        10        11        11        11
## 4      1746      1783      1796      1812      1845      1878      1892
## 5      1655      1681      1743      1824      1919      2032      2156
## 6        19        20        20        21        22        25        25
##   28/9/2013 30/9/2013 2/10/2013 4/10/2013 6/10/2013 8/10/2013 10/10/2013
## 1      3706      3746      3795      3832      3897      3923       3985
## 2      3258      3354      3475      3654      3911      4024       4125
## 3        11        11        11        11        11        11         11
## 4      1914      1924      1929      1941      1952      1955       1959
## 5      2252      2312      2387      2459      2544      2614       2693
## 6        26        26        26        26        27        27         27
##   12/10/2013 total_sales
## 1       4048       75979
## 2       4277       52256
## 3         11         223
## 4       1963       39691
## 5       2736       44077
## 6         27         457
```

```
#Merged data
merged_data <- data.frame(attrib2 ,dresssale2)
head(merged_data)
```

```
##      Style   Price Rating Size Season NeckLine SleeveLength waiseline
## 1    Sexy     Low    4.6    M Summer   o-neck   sleeveless    empire
## 2  Casual     Low    0.0    L Summer   o-neck        Petal   natural
## 3 vintage    High    0.0    L Autumn   o-neck         full   natural
## 4   Brief Average    4.6    L Spring   o-neck         full   natural
## 5    cute     Low    4.5    M Summer   o-neck    butterfly   natural
## 6 bohemian     Low    0.0    M Summer   v-neck   sleeveless    empire
##        Material FabricType Decoration Pattern.Type Recommendation X29.8.2013
## 1          null    chiffon    ruffles       animal              1       2114
## 2    microfiber       null    ruffles       animal              0        151
## 3      polyster       null       null        print              0          6
## 4          silk    chiffon embroidery        print              1       1005
## 5 chiffonfabric    chiffon        bow          dot              0        996
## 6          null       null       null        print              0          4
##   X31.8.2013 X2.9.2013 X4.9.2013 X6.9.2013 X8.9.2013 X10.9.2013 X12.9.2013
## 1       2274      2491      2660      2727      2887       2930       3119
## 2        275       570       750       813      1066       1164       1558
## 3          7         7         7         8         8          9         10
## 4       1128      1326      1455      1507      1621       1637       1723
## 5       1175      1304      1396      1432      1559       1570       1638
## 6          5        11        13        13        13         16         18
##   X14.9.2013 X16.9.2013 X18.9.2013 X20.9.2013 X22.9.2013 X24.9.2013 X26.9.2013
## 1       3204       3277       3321       3386       3479       3554       3624
## 2       1756       1878       1985       2106       2454       2710       2942
## 3         10         10         10         10         11         11         11
## 4       1746       1783       1796       1812       1845       1878       1892
## 5       1655       1681       1743       1824       1919       2032       2156
## 6         19         20         20         21         22         25         25
##   X28.9.2013 X30.9.2013 X2.10.2013 X4.10.2013 X6.10.2013 X8.10.2013 X10.10.2013
## 1       3706       3746       3795       3832       3897       3923        3985
## 2       3258       3354       3475       3654       3911       4024        4125
## 3         11         11         11         11         11         11          11
## 4       1914       1924       1929       1941       1952       1955        1959
## 5       2252       2312       2387       2459       2544       2614        2693
## 6         26         26         26         26         27         27          27
##   X12.10.2013 total_sales
## 1        4048       75979
## 2        4277       52256
## 3          11         223
## 4        1963       39691
## 5        2736       44077
## 6          27         457
```

```
str(merged_data)
```

```
## 'data.frame':    500 obs. of  37 variables:
##  $ Style         : Factor w/ 12 levels "Sexy","Casual",..: 1 2 3 4 5 6 2 7 8 6 ...
##  $ Price         : Factor w/ 5 levels "Low","High","Average",..: 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 "M","L","XL","free",..: 1 2 2 2 1 1 3 4 4 4 ...
##  $ Season        : Factor w/ 4 levels "Summer","Autumn",..: 1 1 2 3 1 1 1 2 3 1 ...
##  $ NeckLine      : Factor w/ 16 levels "o-neck","v-neck",..: 1 1 1 1 1 2 1 1 2 2 ...
##  $ SleeveLength  : Factor w/ 13 levels "sleeveless","Petal",..: 1 2 3 3 4 1 3 5 5 1 ...
##  $ waiseline     : Factor w/ 5 levels "empire","natural",..: 1 2 2 2 2 1 3 2 1 2 ...
##  $ Material      : Factor w/ 24 levels "null","microfiber",..: 1 2 3 4 5 1 6 3 6 7 ...
##  $ FabricType    : Factor w/ 18 levels "chiffon","null",..: 1 2 2 1 1 2 2 3 3 1 ...
##  $ Decoration    : Factor w/ 24 levels "ruffles","null",..: 1 1 2 3 4 2 2 5 6 2 ...
##  $ Pattern.Type  : Factor w/ 13 levels "animal","print",..: 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));
```

```
## [1] 350  37
```

```
print(dim(test))
```

```
## [1] 150  37
```

```
#Classification - Predict recommendation
#First model (Naive Bayes): # non-linear model # simple & fast
options(scipen = 999)
naive_model = naiveBayes(Recommendation ~.,data = train) # build model # . means all column
confusionMatrix(train$Recommendation, predict(naive_model,train), positive = '1') # create confusion Ma
```

```
## 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] : 0.000000000001409
##
##                   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
##
```

```
naive_predict = predict(naive_model,test)  # predict test set
table(naive_predict,test$Recommendation)  # create table
```

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

```
# Support vector machine (SVM): # Linear model # complex
svm_model = svm(Recommendation ~.,train) # build model
confusionMatrix(train$Recommendation,predict(svm_model),positive = '1') # create confusion Matrix
```

```
## 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 : <0.0000000000000002
##
##             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
##
```

```
svm_predict = predict(svm_model,test) # predict test set
table(svm_predict,test$Recommendation) # create table
```

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

```
# Third model (Random Forest)
randomForest_model = randomForest(x = train, y = train$Recommendation,ntree =800) # build model
confusionMatrix(train$Recommendation,predict(randomForest_model),positive = '1') # create confusion Mat
```

```
## 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] : < 0.00000000000000022
## 
##                   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
## 
```

```
randomForest_predict = predict(randomForest_model,test) # predict test set
table(randomForest_predict,test$Recommendation ) # create table
```

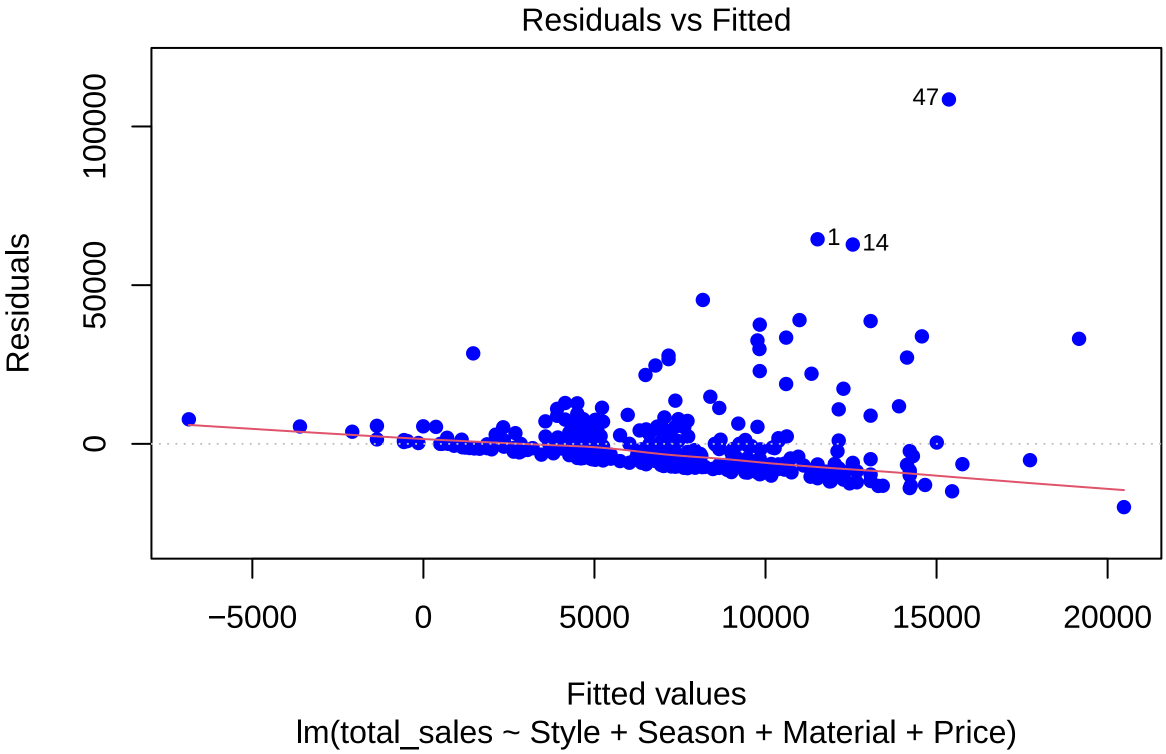
```
## 
## randomForest_predict  1  0
##                    1 63  0
##                    0  0 87
```

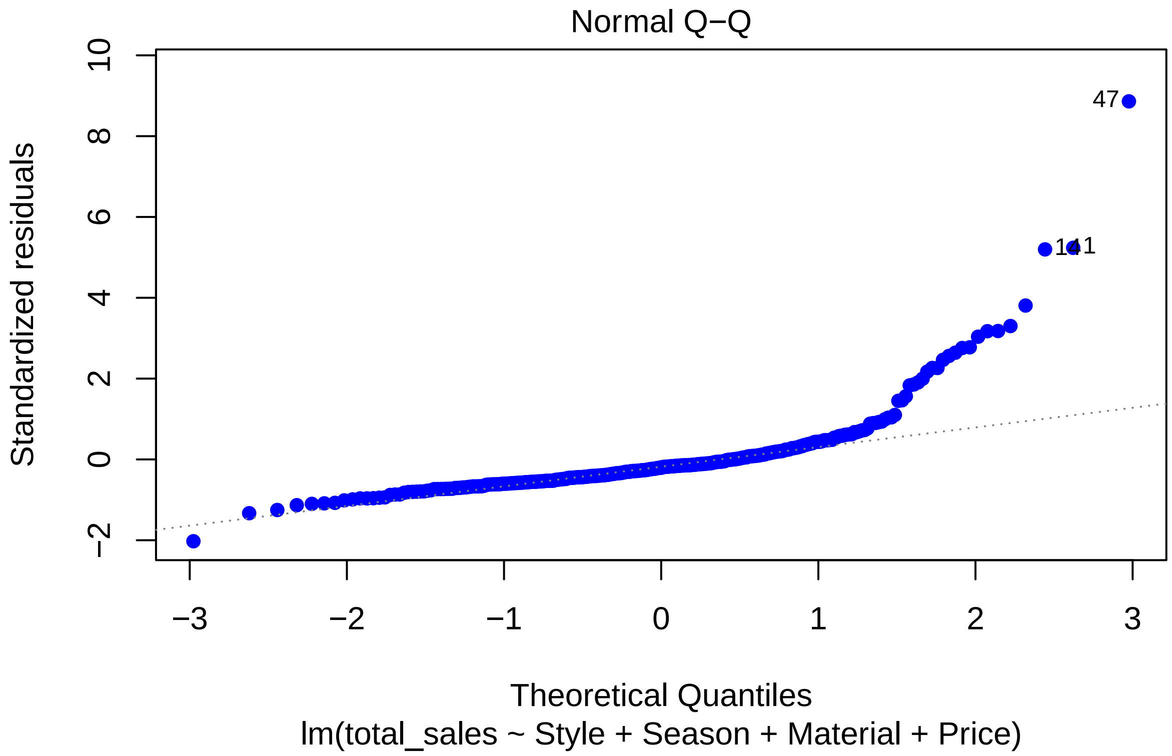
```
# Regresstion model (total sales and (Style+Season+Material+Price))
regressor_Sales = lm(formula = total_sales ~ Style+Season+Material+Price, data = train) # build model
summary(regressor_Sales) # print model summary
```

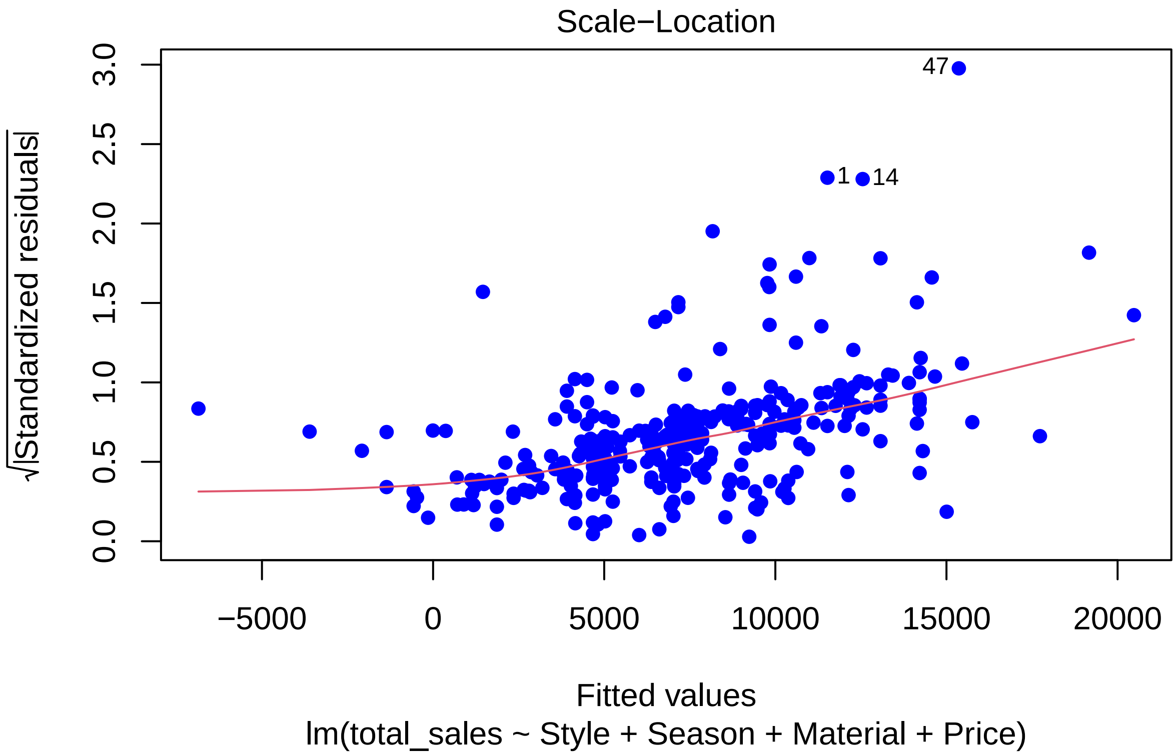
```
## 
## Call:
## lm(formula = total_sales ~ Style + Season + Material + Price, 
##     data = train)
## 
## Residuals:
##    Min     1Q Median     3Q    Max
## -19936  -6113  -2230   1381 108508
##
## Coefficients:
##                        Estimate Std. Error t value   Pr(>|t|)
## (Intercept)            11521.7     2460.7   4.682 0.00000424 ***
## StyleCasual            -4739.3     2018.1  -2.348     0.0195 *
## Stylevintage            2786.7     3731.1   0.747     0.4557
## StyleBrief             -2293.6     3957.3  -0.580     0.5626
## Stylecute              -4514.3     3037.8  -1.486     0.1383
## Stylebohemian          -7057.2     3738.5  -1.888     0.0600 .
## StyleNovelty           -6905.0     6052.5  -1.141     0.2548
## StyleFlare            -11965.4    12808.3  -0.934     0.3509
## Styleparty             -4017.0     3208.8  -1.252     0.2116
## Stylework              -3254.6     4066.9  -0.800     0.4242
## Stylefashion           -9220.8    12763.8  -0.722     0.4706
## SeasonAutumn            -761.5     2339.6  -0.325     0.7450
## SeasonSpring            2694.3     1915.0   1.407     0.1604
## SeasonWinter            -527.7     1944.7  -0.271     0.7863
## Materialmicrofiber     12382.7     7516.0   1.648     0.1005
## Materialpolyster         790.2     2243.6   0.352     0.7249
## Materialsilk           -2454.3     3316.7  -0.740     0.4599
## Materialchiffonfabric   3595.3     3141.2   1.145     0.2533
## Materialcotton         -2111.8     1884.6  -1.121     0.2633
## Materialnylon          -3568.7     5518.6  -0.647     0.5183
## Materialother          -2426.9     9130.3  -0.266     0.7906
## Materialmilksilk       -2429.2     9068.5  -0.268     0.7890
## Materiallinen           2010.3     7588.1   0.265     0.7912
## Materialrayon          -6025.8     5056.8  -1.192     0.2343
## Materiallycra           -637.5     7475.0  -0.085     0.9321
## Materialmix            -1602.8     4285.8  -0.374     0.7087
## Materialacrylic        -2853.4     7500.8  -0.380     0.7039
## Materialspandex        -7213.0    12932.0  -0.558     0.5774
## Materiallace           -2242.1    12843.7  -0.175     0.8615
## Materialmodal          -6416.6    12731.2  -0.504     0.6146
## Materialcashmere       -5257.7     9108.9  -0.577     0.5642
## Materialknitting       -1842.8    12767.8  -0.144     0.8853
## Materialsill           -7843.1    12790.6  -0.613     0.5402
## Materialshiffon        -3215.7     9269.8  -0.347     0.7289
## PriceHigh              -2030.6     3912.1  -0.519     0.6041
## PriceAverage             355.2     1627.8   0.218     0.8274
## PriceMedium            -4172.0     3571.0  -1.168     0.2436
## Pricevery-high         -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
```

```r
plot(regressor_Sales, pch = 16, col = "blue") # Plot the results
```

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

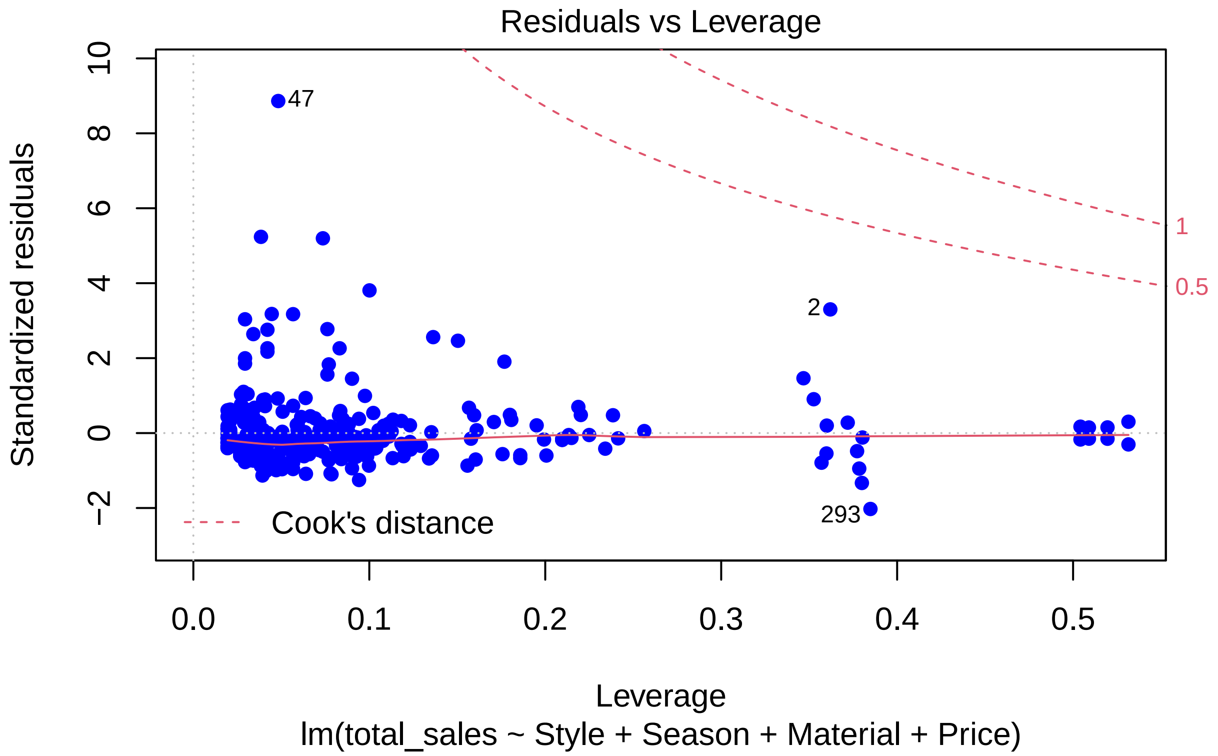

Residuals vs Fitted
Residuals
100000
50000
0
−5000
0
5000
10000
15000
20000
47
1
14
Fitted values
lm(total_sales ~ Style + Season + Material + Price)


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



```
abline(regressor_Sales) # Add regression line
```

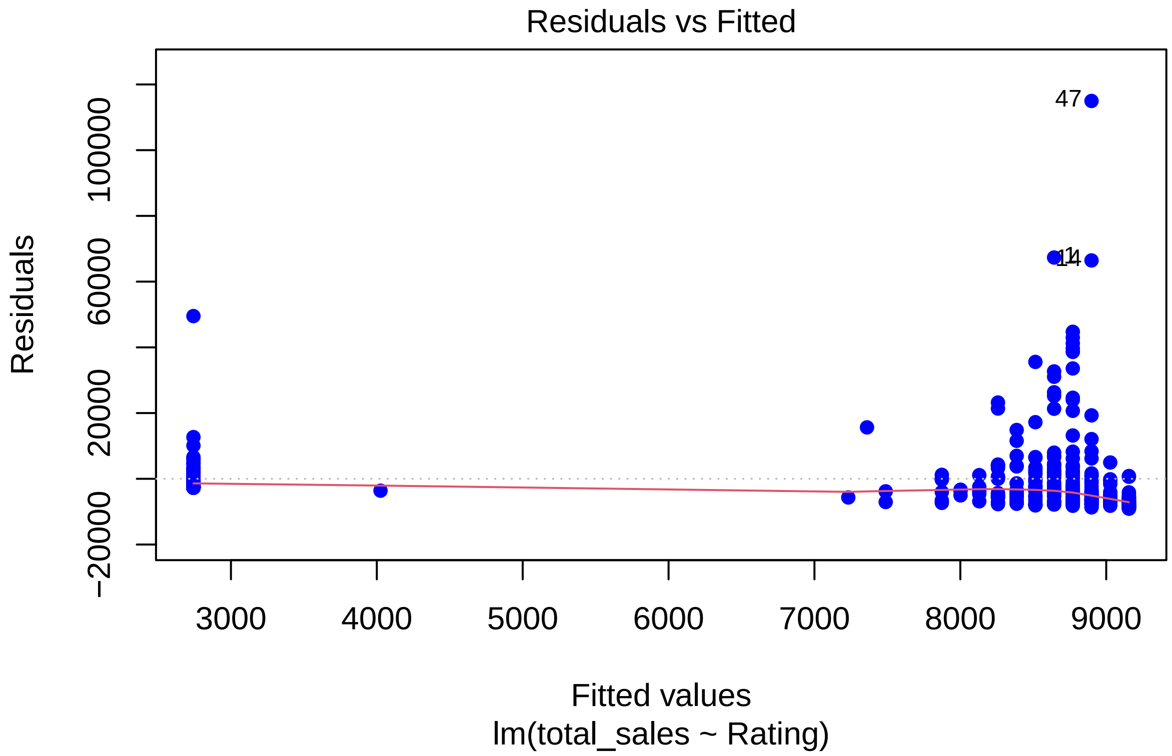
```
## Warning in abline(regressor_Sales): only using the first two of 38 regression
## coefficients
```

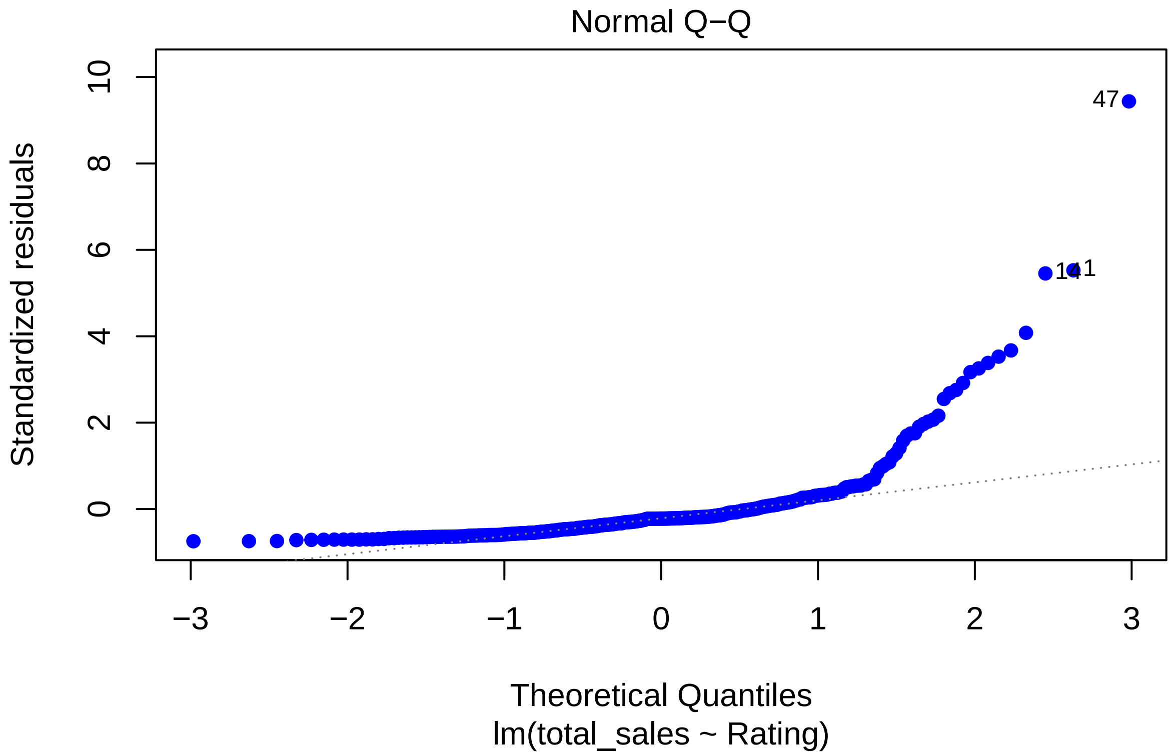


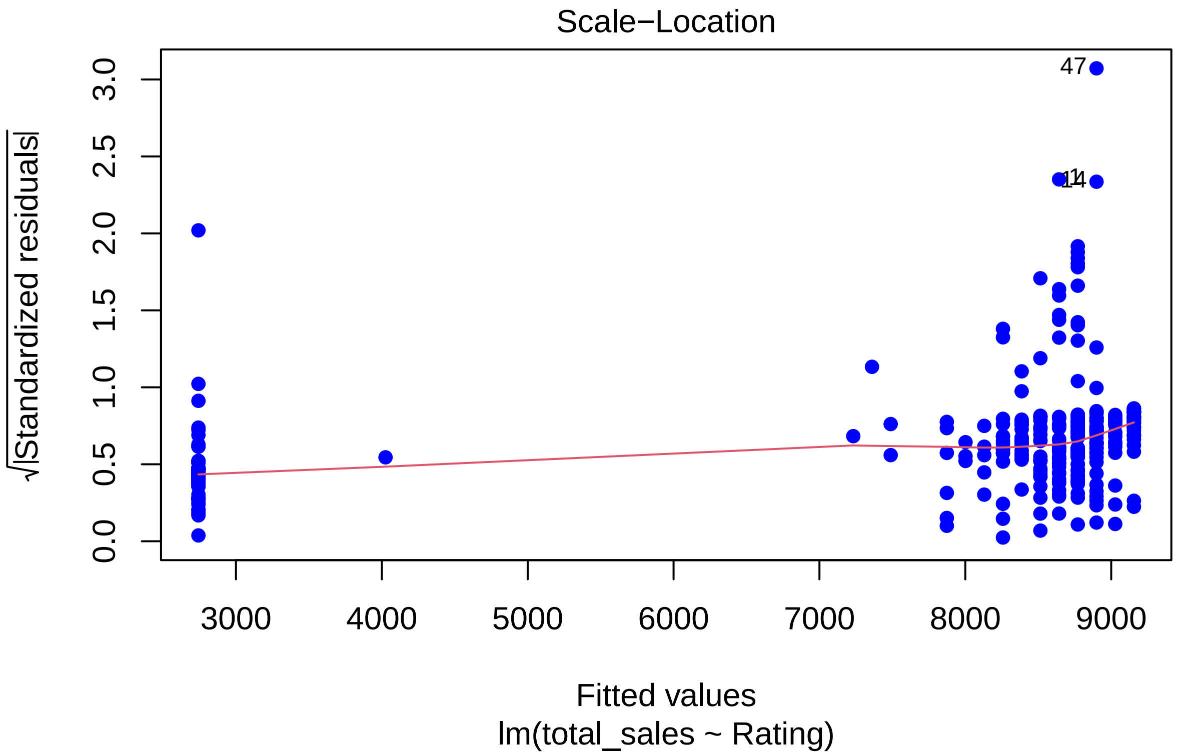
```
# Regression (total sales and Rating)
regressor_Rating = lm(formula = total_sales ~ Rating, data = train) # build model
summary(regressor_Rating) # print model summary
```

```
##
## 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 0.0000902 ***
## ---
## 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: 0.00009022
```

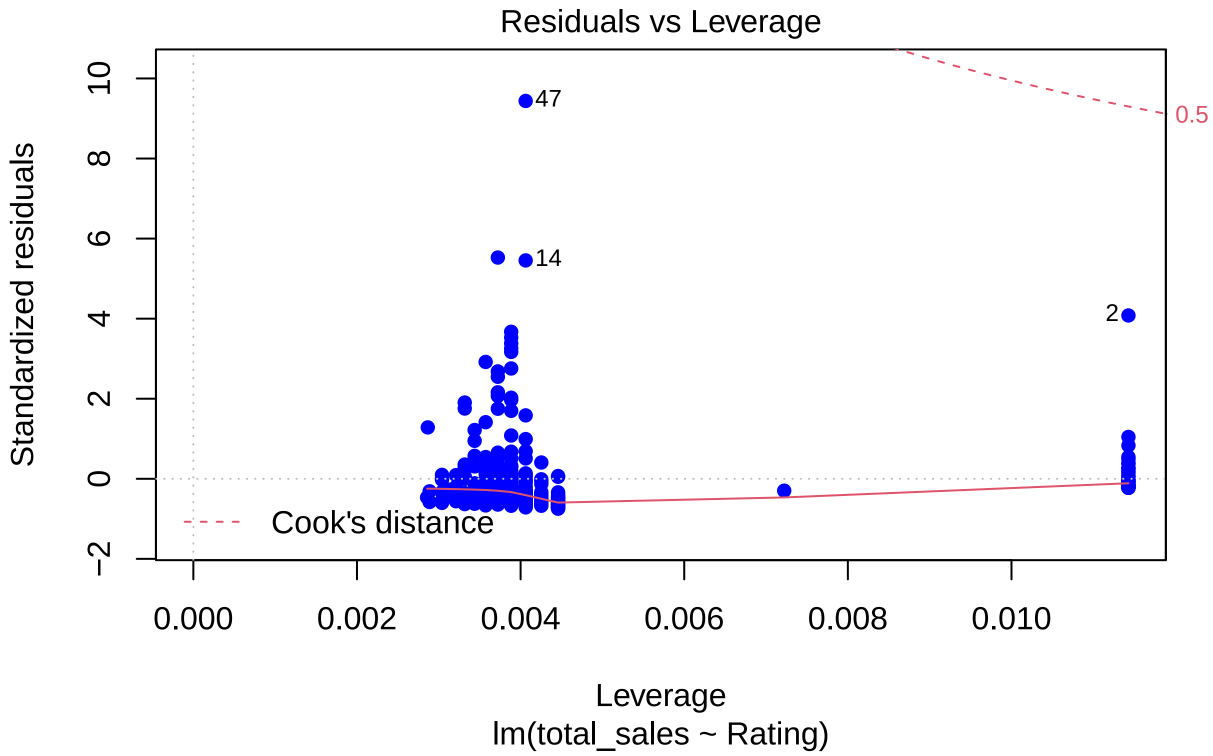
```
plot(regressor_Rating, pch = 16, col = "blue") # Plot the results
```


Residuals vs Fitted
Residuals
−20000
20000
60000
100000
3000
4000
5000
6000
7000
8000
9000
Fitted values
lm(total_sales ~ Rating)
47
11
14


Normal Q−Q
Standardized residuals
10
8
6
4
2
0
−3
−2
−1
0
1
2
3
47
141
14
Theoretical Quantiles
lm(total_sales ~ Rating)



```
abline(regressor_Rating) # Add regression line
```


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