Analyze the Sales Report of a Clothes Manufacturing Outlet

Business Analytic Foundation with R Tools- Solutions

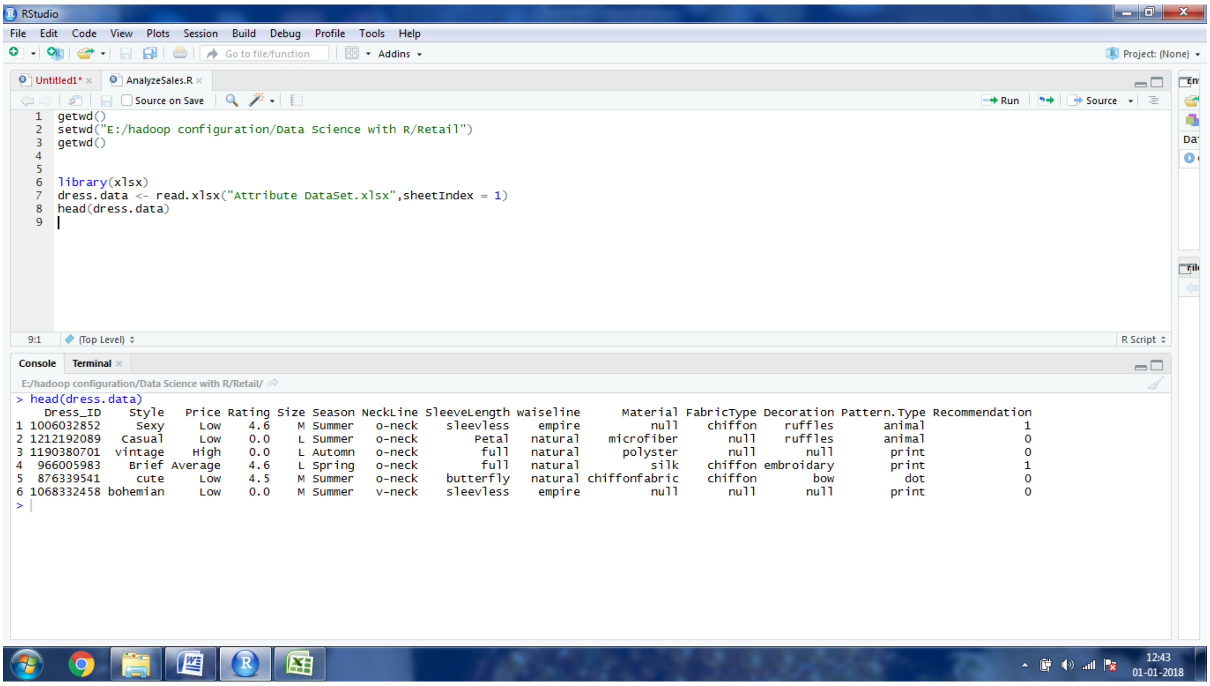
1. To automate the process of recommendations, the store needs to analyze the given attributes of the product, like style, season, etc., and come up with a model to predict the recommendation of products (in binary output – 0 or 1) accordingly.

The **DressRecommendation.csv** file contains all the attributes and the recommendations for each dress in binary. For binary output models, logistic regression can be used to build a suitable model to recommend products (as mentioned earlier, a value of 1 denotes a positive recommendation and 0 denotes a negative recommendation). To perform logistic regression on a given dataset, we need to decide two major attributes of the model – the dependent and independent variables. The required values are:
Dependent variable: Recommendation
Independent Variables: All other variables, except the Dress_ID, since it is only an identifier.
Outlier treatments: None

Code:

```
dress.data <- read.csv("DressRecommendation.csv",header = T)

head(dress.data)
```



Code:

attach(dress.data)

model <- glm(Recommendation ~ Style + Price + Rating + Size + Season + NeckLine +SleeveLength + waiseline+Decoration + Material + FabricType + Pattern.Type, data = dress.data)

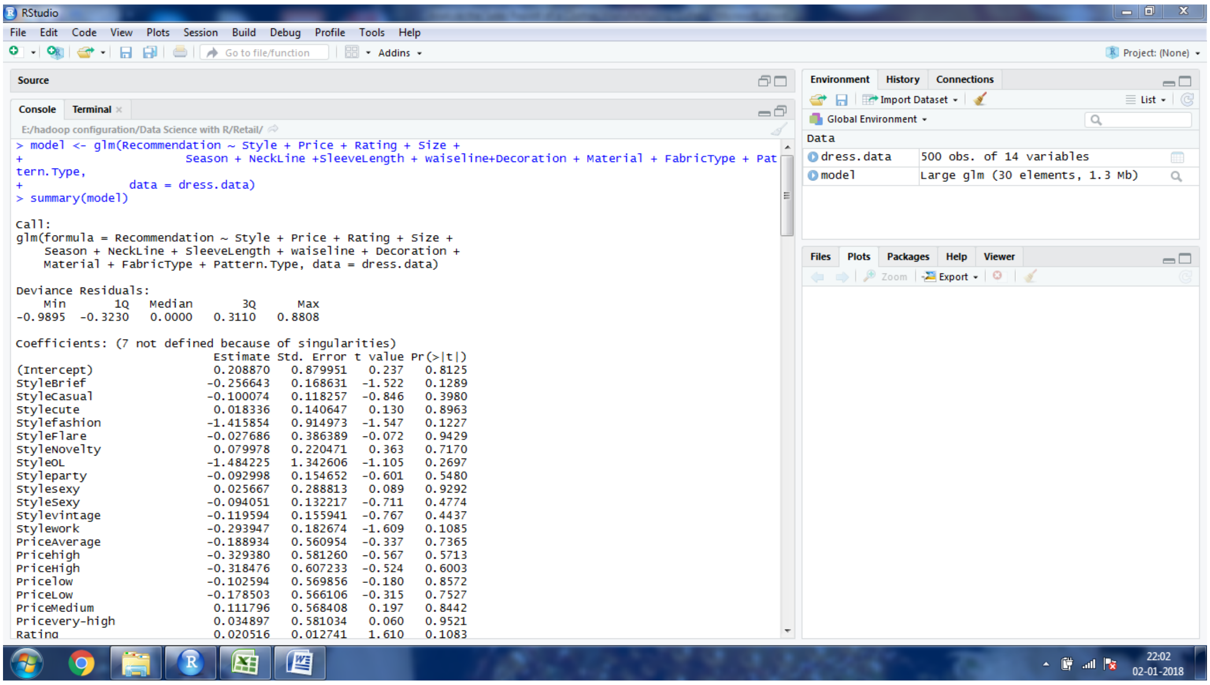
summary(model)

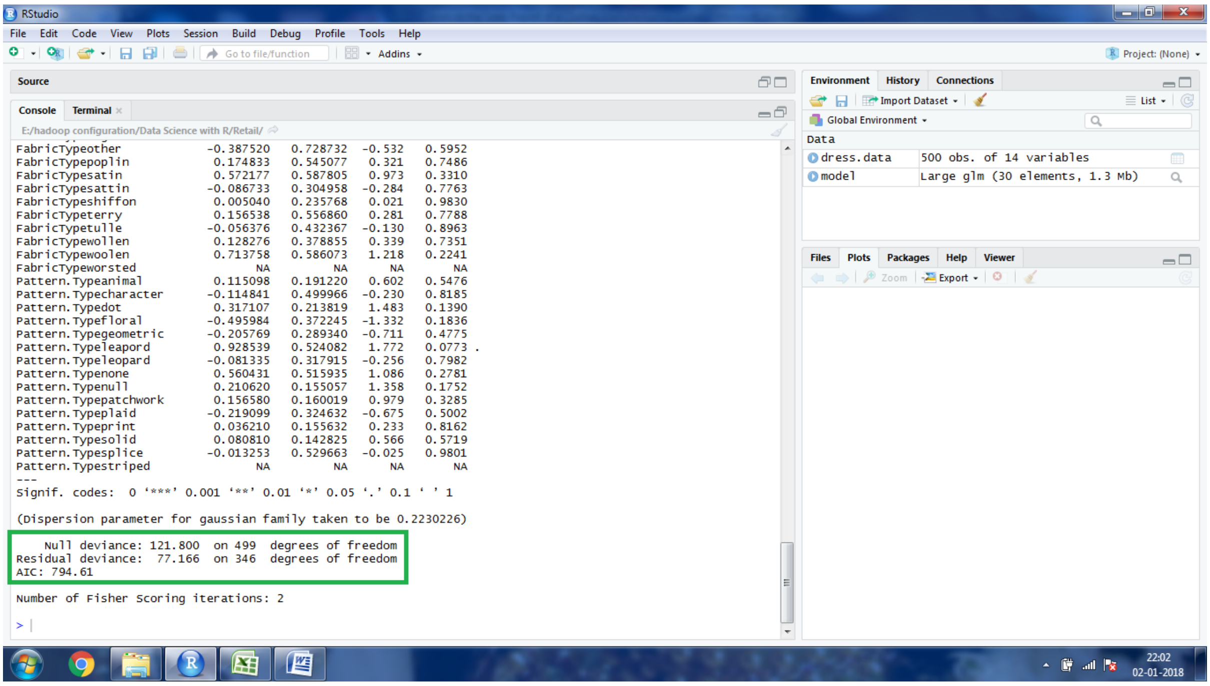
Result:

From the significance codes for each attribute, we can see that Pattern Type and Sleeve Length make an impact on the recommendation, both positively affecting the recommendation. Other than that, we can see that the increased number of factors and comparatively lesser number of entries make the predictions slightly difficult.

However, the residual deviance is lower than the null deviance, which implies that using the independent variables makes it closer to predicting the actual values of recommendation.

With the given model, the new data or attributes can be fed into the model to get recommendations.



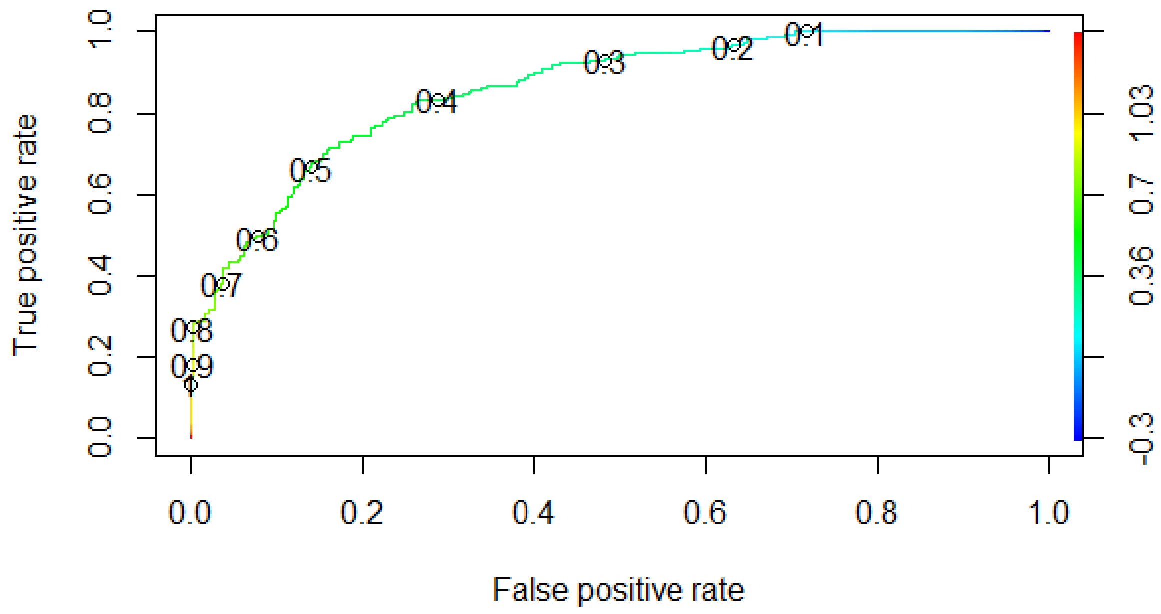


Code:

```
mddress.data <- dress.data[,c(-1,-14)]
result <- predict(model,mddress.data,type = "response")
library(ROCR)
rocrPredic <- prediction(result,Recommendation)
rocrperf  <- performance(rocrPredic,"tpr","fpr")
plot(rocrperf,colorize=TRUE, print.cutoffs.at=seq(0.1,by=0.1))
```

Result :

From that plot we can see the best threshold point is 0.4



Code :

train_PredSurvived <- ifelse(result > 0.4,1,0)
table(predicted=train_PredSurvived,actualdata=Recommendation)

```
Number of Fisher Scoring iterations: 2

> mddress.data <- dress.data[,c(-1,-14)]
> result <- predict(model,mddress.data,type = "response")
> library(ROCR)
> rocrPredic <- prediction(result,Recommendation)
> rocrperf  <- performance(rocrPredic,"tpr","fpr")
> plot(rocrperf,colorize=TRUE, print.cutoffs.at=seq(0.1,by=0.1))
> train_PredSurvived <- ifelse(result > 0.4,1,0)
> table(predicted=train_PredSurvived,actualdata=Recommendation)
         actualdata
predicted   0   1
        0 207  35
        1  83 175
> |
```

Code :

library(caret)
confusionMatrix(train_PredSurvived,Recommendation)

Result :

From confusion Matrix. We can see that Accuracy is 76%
And Sensitivity is 71% and Specificity is 83%

```
> library(caret)
> confusionMatrix(train_PredSurvived,Recommendation)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 207  35
         1  83 175

               Accuracy : 0.764
                 95% CI : (0.7243, 0.8006)
    No Information Rate : 0.58
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5304
 Mcnemar's Test P-Value : 1.514e-05

            Sensitivity : 0.7138
            Specificity : 0.8333
         Pos Pred Value : 0.8554
         Neg Pred Value : 0.6783
             Prevalence : 0.5800
         Detection Rate : 0.4140
   Detection Prevalence : 0.4840
      Balanced Accuracy : 0.7736

       'Positive' Class : 0

> |
```

2. In order to stock the inventory, the store wants to analyze the sales data and predict the trend of total sales for each dress for an extended period of three or more alternative days.

For this question, we will use the **Dress Sales.xlsx** file. Since the time series should be a vector, we will work on the total sales per day (of all items). The total sales is calculated in Excel (using the SUM command) and saved in a file named **totalSales3.csv**. (The file is provided here for verification).
The Auto.arima function is used to predict the trend for three more days. Note that any of the given time series functions can be used in the prediction of sales.

Code :

```
sales <- read.csv("totalSales3.csv")
sales
```

```
> sales <- read.csv("totalSales3.csv")
> sales
       X X29.08.2013 X31.08.2013 X09.02.2013 X09.04.2013 X09.06.2013 X09.08.2013 X09.10.2013 X09.12.2013
1 Totals       94883      100483      107081      149336      151829      157647      159391      165962
  X14.09.2013 X16.09.2013 X18.09.2013 X20.09.2013 X22.09.2013 X24.09.2013 X26.09.2013 X28.09.2013 X30.09.2013
1      169101      171726      174360      179037      183261      185616       77934      193734       55412
  X10.02.2013 X10.04.2013 X10.06.2013 X10.08.2010 X10.10.2013 X10.12.2013
1       56395       57405      206334       59816       60757      215533
> |
```

Code :

```
mm <- as.matrix(sales[,-1])
numeric.vector <-  as.numeric(as.vector(mm))
timeseries <- ts(numeric.vector, start = 1,frequency = 7)
fit <- auto.arima(timeseries)
summary(fit)
```

```
> mm <- as.matrix(sales[,-1])
> numeric.vector <-  as.numeric(as.vector(mm))
> timeseries <- ts(numeric.vector, start = 1,frequency = 7)
> fit <- auto.arima(timeseries)
> summary(fit)
Series: timeseries
ARIMA(1,1,0)(0,1,0)[7]

Coefficients:
          ar1
      -0.5754
s.e.   0.1953

sigma^2 estimated as 4.68e+09:  log likelihood=-187.97
AIC=379.93   AICc=380.93   BIC=381.35

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -4326.728 53371.64 31649.38 -17.82185 33.33584 0.5248468 -0.1716439
> |
```

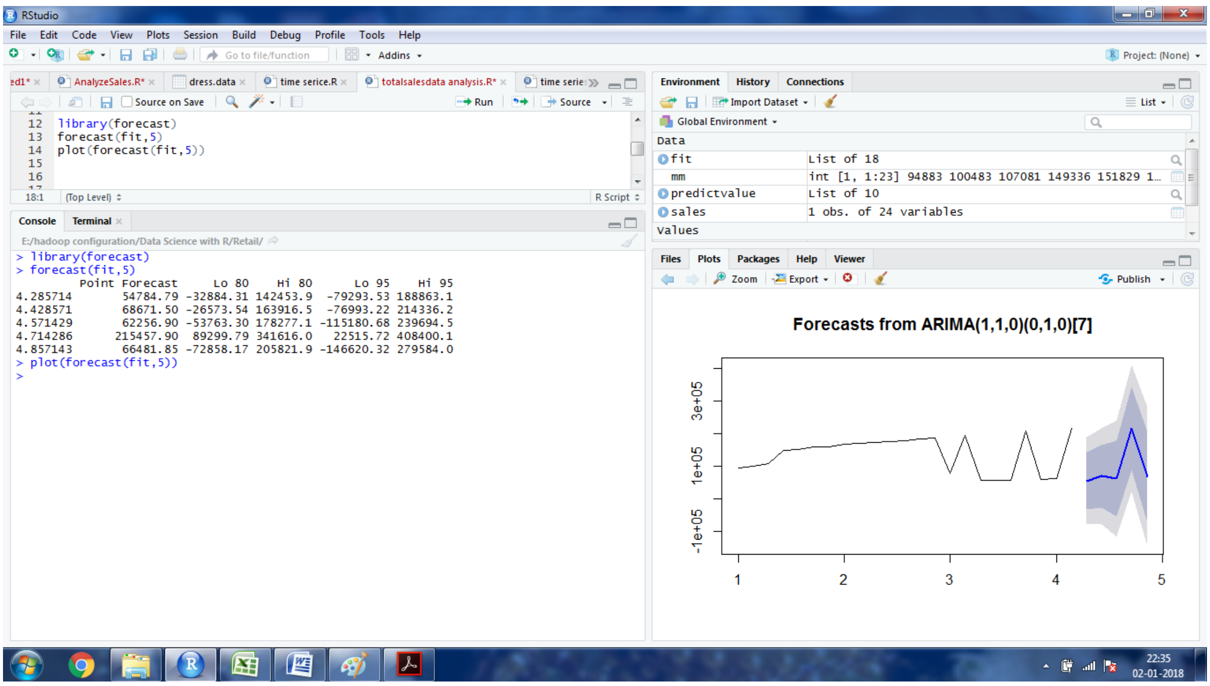
Code :

```
library(forecast)
forecast(fit,5)
plot(forecast(fit,5))
```

Result:

The model built by the Auto.arima function is ARIMA(1,1,0)(0,1,0). The coefficients, information criterion (used in comparing different models), and the error terms are given.

The forecasted values are 54784, 68671, 62256, 215456 and 66481 respectively for the next five dates. A plot of the forecasted values show that there is a lot of fluctuation in the total sales, and hence we can see that the low and high values, which are 80% and 95%, have huge differences for the predicted values (depicted by the light grey and dark grey areas in the plot).



Similarly, the forecasting used for each dress to make individual predictions for the inventory.

Code :

```
sales <- read.csv("csvdresssale.csv")
modifydata <- sales[,c(-25:-36,-1)]
modifydata[is.na(modifydata)] <- 0

i <- 0
n <- nrow(modifydata)
#n <- 450
tmp <- 0

while(i < n) {
  mm <- as.matrix(modifydata[i+1,])
  vv <- as.vector(as.numeric(mm))
  timeseries <- ts(vv, start = 1,frequency = 7)
  fit <- auto.arima(timeseries)
  predictedvalue <-  as.data.frame(forecast(fit,3))
  productwithpredictedvalue <- data.frame(dressid=sales[i+1,1],forecast1stday=predictedvalue$`Point Forecast`[1],
                              forecast2ndday=predictedvalue$`Point Forecast`[2],
                              forecast3rdday=predictedvalue$`Point Forecast`[3])

  tmp <- rbind(tmp,productwithpredictedvalue)
  i <- i+1
  print(tmp)
}

View(tmp)
```

Result :

Created a new data frame where Dressid for each Dress Id and three attribute is forecasted values for next three dates like forecast1stday means forecasting 1st date . Similarly, forecast2ndday and forecast3rdday for forecasting 2nd date and 3rd date.

	dressid	forecast1stday	forecast2ndday	forecast3rdday
1	0	0.000000	0.000000	0.000000
2	1006032852	4110.327664	4173.107363	4235.583143
3	1212192089	4464.545455	4652.090909	4839.636364
4	1190380701	11.050948	11.101896	11.152844
5	966005983	1967.000000	1971.000000	1975.000000
6	876339541	2825.053038	2905.799641	2988.207532
7	1068332458	27.345484	27.690967	28.036451
8	1220707172	559.000000	580.000000	597.000000
9	1219677488	265.947553	273.742428	284.220962
10	1113094204	34.272727	35.545455	36.818182
11	985292672	14.227273	14.454545	14.681818
12	1117293701	142.331598	148.103728	153.875858
13	898481530	210.136364	218.272727	226.409091
14	957723897	3074.087817	3159.535855	3247.612924
15	749031896	4246.000000	4322.000000	4398.000000
16	1055411544	53.442044	53.688685	54.021702
17	1162628131	232.321698	242.405332	253.607999
18	624314841	2577.427526	2618.563837	2661.283412
19	830467746	19.033629	19.033629	19.033629
20	840857118	17.681818	18.363636	19.045455
21	1113221101	647.000000	667.000000	680.000000
22	861754372	411.071295	424.119916	437.168537
23	[illegible]	[illegible]	[illegible]	[illegible]

Showing 1 to 23 of 71 entries

3. To decide the pricing for various upcoming clothes, the store wishes to find how the style, season, and material affect the sales of a dress and if the style of the dress is more influential than its price.

a. Firstly, calculate the total sales per dress ID (like the previous question) and save it along with the Attribute DataSet file (with column name as Total.Sales). We need to find how style, season, and material affect the sale of a dress. Since they are categorical, let us first use the analysis of variances to see if the different types make an impact.

The required variables are:

Independent variable: Total.Sales

Dependent Variables: Style, Season, and Material

Code :

extractotalsales <- read.csv("extractotalsales.csv",header = T)
extractotalsales[,25]
dress.data <- cbind(dress.data,Total.Sales=extractotalsales[-501,25])
attach(dress.data)

TestForStyle <- aov(Total.Sales ~ Style)
summary(TestForStyle)
TestForSeason <- aov(Total.Sales ~ Season)
summary(TestForSeason)
TestForMaterial <- aov(Total.Sales ~ Material)
summary(TestForMaterial)

Result:

From the p-values we can see that out of the three, only season has a high p-value, thus showing that different seasons have different impact on the sales.

```
> TestForStyle <- aov(Total.Sales ~ Style)
> summary(TestForStyle)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Style        12 2.738e+09 228195052   1.527  0.111
Residuals   487 7.278e+10 149450545
> TestForSeason <- aov(Total.Sales ~ Season)
> summary(TestForSeason)
             Df    Sum Sq   Mean Sq F value  Pr(>F)
Season        8 3.700e+09 462453725   3.162 0.00167 **
Residuals   491 7.182e+10 146275206
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TestForMaterial <- aov(Total.Sales ~ Material)
> summary(TestForMaterial)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Material     24 3.221e+09 134204184   0.882  0.628
Residuals   475 7.230e+10 152210222
> |
```

Next, we will try linear regression with Style, Season and Material to find out which of the factors affect the sales more. The required variables are -
Dependent Variable: Total.Sales
Independent Variables: Style, Season, Material

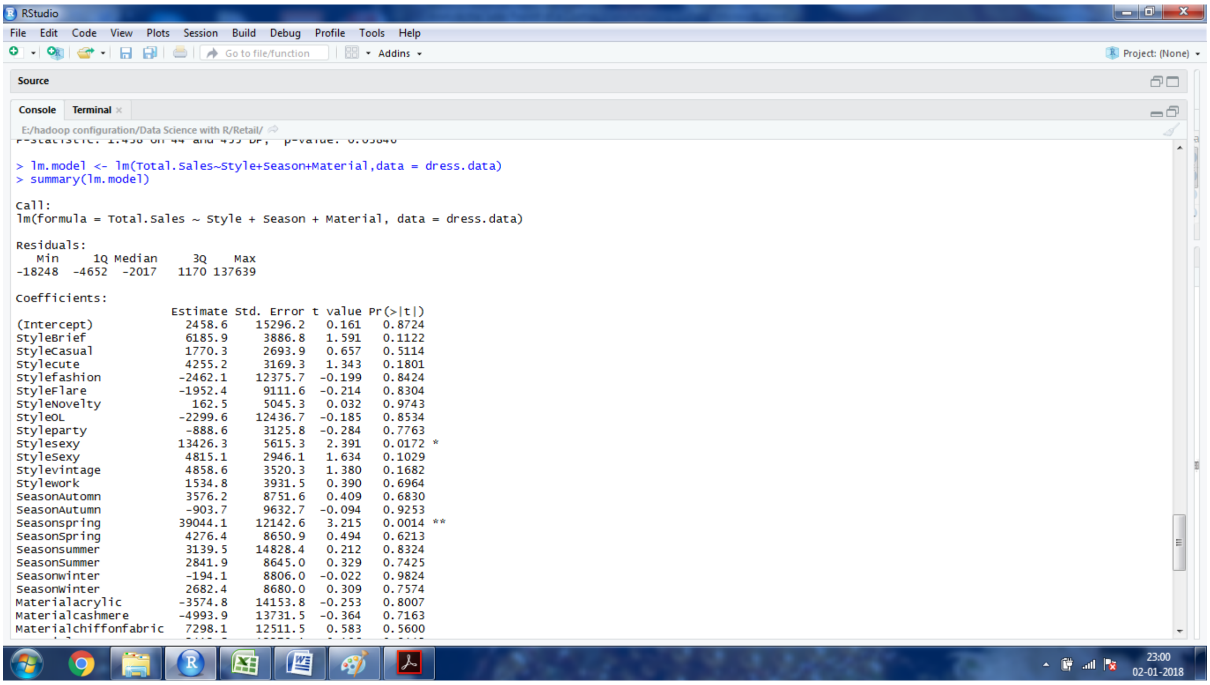
Code:

head(dress.data)

lm.model <- lm(Total.Sales~Style+Season+Material,data = dress.data)

summary(lm.model)

```
> head(dress.data)
    Dress_ID    Style   Price Rating Size Season NeckLine SleeveLength waiseline      Material FabricType Decoration Pattern.Type Recommendation Total.Sales
1 1006032852     Sexy     Low    4.6    M Summer   o-neck    sleevless    empire          null    chiffon    ruffles       animal              1       75979
2 1212192089   Casual     Low    0.0    L Summer   o-neck        Petal   natural    microfiber       null    ruffles       animal              0       52256
3 1190380701  vintage    High    0.0    L Automn   o-neck         full   natural      polyster       null       null        print              0         223
4  966005983    Brief Average    4.6    L Spring   o-neck         full   natural          silk    chiffon embroidary        print              1       39691
5  876339541     cute     Low    4.5    M Summer   o-neck    butterfly   natural chiffonfabric    chiffon        bow          dot              0       44077
6 1068332458 bohemian     Low    0.0    M Summer   v-neck    sleevless    empire          null       null       null        print              0         457
```

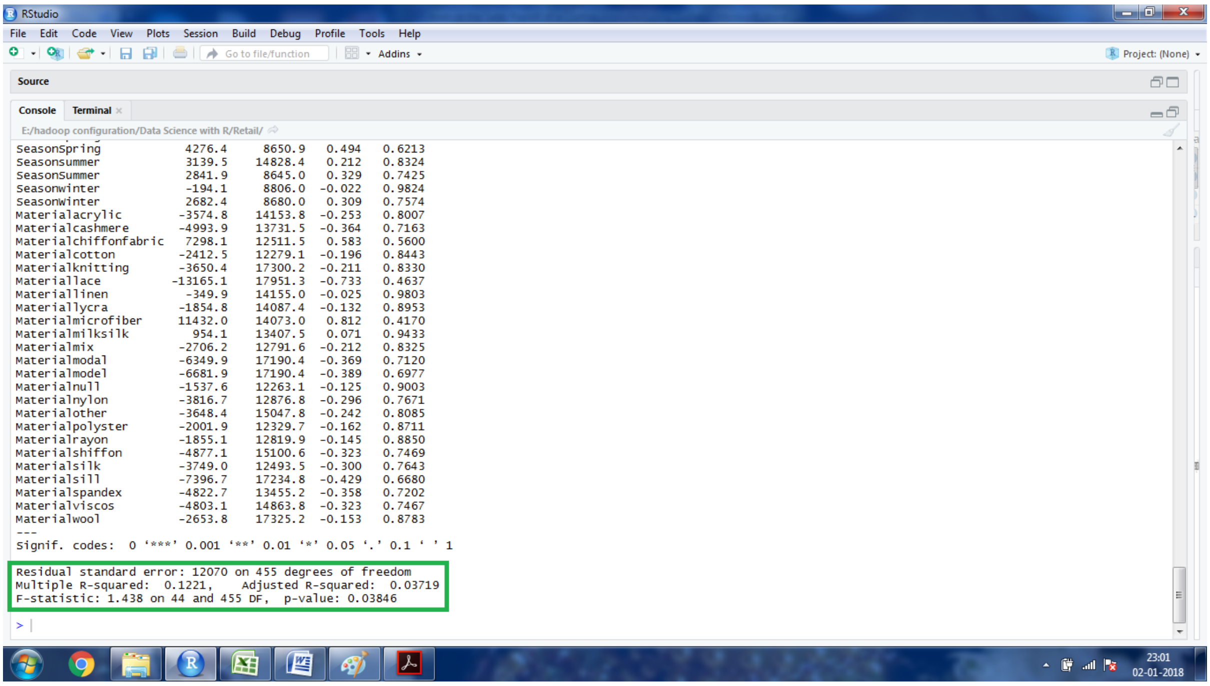


```
> lm.model <- lm(Total.Sales~Style+Season+Material,data = dress.data)
> summary(lm.model)

Call:
lm(formula = Total.Sales ~ Style + Season + Material, data = dress.data)

Residuals:
   Min     1Q Median     3Q    Max 
-18248  -4652  -2017   1170 137639 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)   
(Intercept)             2458.6    15296.2   0.161   0.8724   
StyleBrief              6185.9     3886.8   1.591   0.1122   
StyleCasual             1770.3     2693.9   0.657   0.5114   
Stylecute               4255.2     3169.3   1.343   0.1801   
Stylefashion           -2462.1    12375.7  -0.199   0.8424   
StyleFlare             -1952.4     9111.6  -0.214   0.8304   
StyleNovelty             162.5     5045.3   0.032   0.9743   
StyleOL                -2299.6    12436.7  -0.185   0.8534   
Styleparty              -888.6     3125.8  -0.284   0.7763   
Stylesexy              13426.3     5615.3   2.391   0.0172 * 
StyleSexy               4815.1     2946.1   1.634   0.1029   
Stylevintage            4858.6     3520.3   1.380   0.1682   
Stylework               1534.8     3931.5   0.390   0.6964   
SeasonAutomn            3576.2     8751.6   0.409   0.6830   
SeasonAutumn            -903.7     9632.7  -0.094   0.9253   
Seasonspring           39044.1    12142.6   3.215   0.0014 **
SeasonSpring            4276.4     8650.9   0.494   0.6213   
Seasonsummer            3139.5    14828.4   0.212   0.8324   
SeasonSummer            2841.9     8645.0   0.329   0.7425   
Seasonwinter            -194.1     8806.0  -0.022   0.9824   
SeasonWinter            2682.4     8680.0   0.309   0.7574   
Materialacrylic        -3574.8    14153.8  -0.253   0.8007   
Materialcashmere       -4993.9    13731.5  -0.364   0.7163   
Materialchiffonfabric   7298.1    12511.5   0.583   0.5600   
```



```
SeasonSpring            4276.4     8650.9   0.494   0.6213   
Seasonsummer            3139.5    14828.4   0.212   0.8324   
SeasonSummer            2841.9     8645.0   0.329   0.7425   
Seasonwinter            -194.1     8806.0  -0.022   0.9824   
SeasonWinter            2682.4     8680.0   0.309   0.7574   
Materialacrylic        -3574.8    14153.8  -0.253   0.8007   
Materialcashmere       -4993.9    13731.5  -0.364   0.7163   
Materialchiffonfabric   7298.1    12511.5   0.583   0.5600   
Materialcotton         -2412.5    12279.1  -0.196   0.8443   
Materialknitting       -3650.4    17300.2  -0.211   0.8330   
Materiallace          -13165.1    17951.3  -0.733   0.4637   
Materiallinen           -349.9    14155.0  -0.025   0.9803   
Materiallycra          -1854.8    14087.4  -0.132   0.8953   
Materialmicrofiber     11432.0    14073.0   0.812   0.4170   
Materialmilksilk         954.1    13407.5   0.071   0.9433   
Materialmix            -2706.2    12791.6  -0.212   0.8325   
Materialmodal          -6349.9    17190.4  -0.369   0.7120   
Materialmodel          -6681.9    17190.4  -0.389   0.6977   
Materialnull           -1537.6    12263.1  -0.125   0.9003   
Materialnylon          -3816.7    12876.8  -0.296   0.7671   
Materialother          -3648.4    15047.8  -0.242   0.8085   
Materialpolyster       -2001.9    12329.7  -0.162   0.8711   
Materialrayon          -1855.1    12819.9  -0.145   0.8850   
Materialshiffon        -4877.1    15100.6  -0.323   0.7469   
Materialsilk           -3749.0    12493.5  -0.300   0.7643   
Materialsill           -7396.7    17234.8  -0.429   0.6680   
Materialspandex        -4822.7    13455.2  -0.358   0.7202   
Materialviscos         -4803.1    14863.8  -0.323   0.7467   
Materialwool           -2653.8    17325.2  -0.153   0.8783   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12070 on 455 degrees of freedom
Multiple R-squared:  0.1221,	Adjusted R-squared:  0.03719 
F-statistic: 1.438 on 44 and 455 DF,  p-value: 0.03846

> 
```

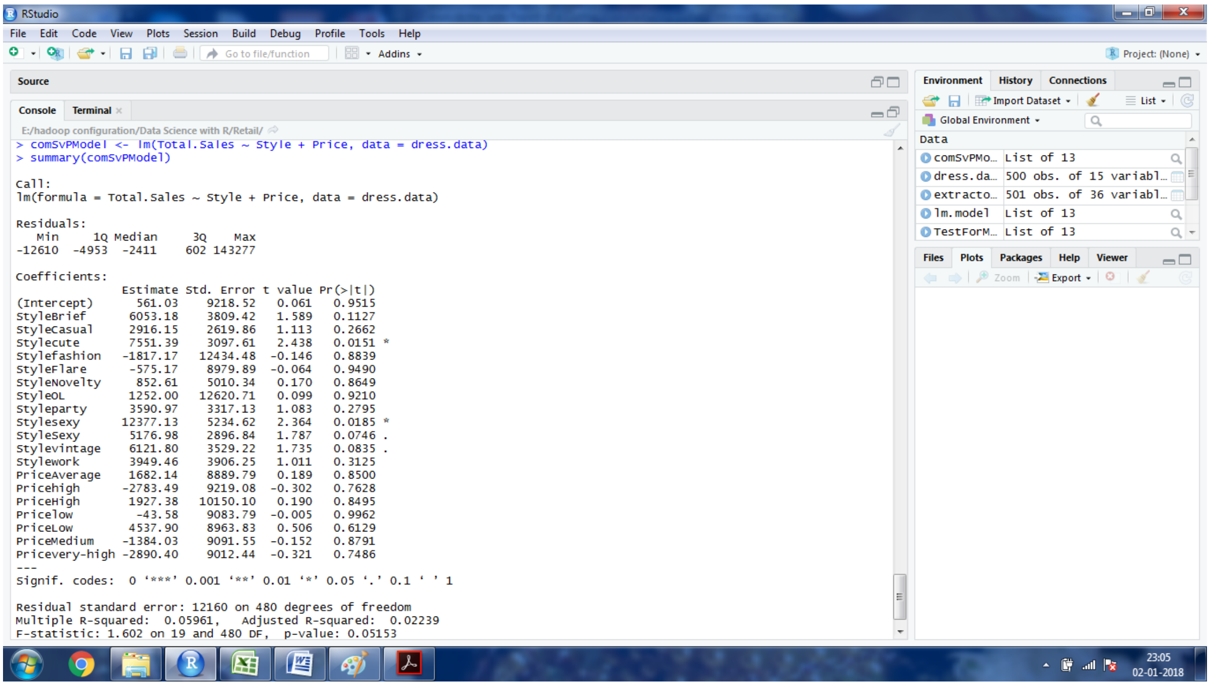
b. To check if style is more influential than the price, let us construct a linear regression model as before, with only the attributes style and price.

Dependent Variable: Total.Sales
Independent Variables: Style, Price

Code:

```
comSvPModel <- lm(Total.Sales ~ Style + Price, data = dress.data)
summary(comSvPModel)
```

Result:
We can see that the style affects more compared to the price range. The price values hardly have any significance whereas the cute, sexy, and vintage style dresses positively affect the sales. The style '**Sexy**' has a positive coefficient of 12377, from which we can safely conclude that sexy dresses are a safe bet when looking at the total sales. The p-value is almost 0.05 and hence we can conclude that these variables do affect the sales linearly.
However, the R-squared value is very low, specifying that these variables do not completely explain the significant changes in sales. That is, style and price cannot completely be used in predicting the total sales of dresses.



4. Also, to increase the sales, the management wants to analyze the attributes of dresses and find which are the leading factors affecting the sales of a dress.

We will again use linear regression, however, with all attribute variables to find which variables are most significant.
Dependent variable: Total.Sales
Independent variables: All the other variables, except Dress_ID

Code :

```
modeluseAll <- lm(Total.Sales~.-Dress_ID,data = dress.data)
summary(modeluseAll)
```

Result :

Since linear regression uses dummy variables for categorical variables, we have a lot of variables affecting the sales. Checking the output, we can make the following observations: –

a. Sexy style makes a positive impact

b. Rating is of a very high significance

c. Large size clothes are sold more

d. Spring season clothes make a positive impact on sales

e. Ruffled neckline clothes have a very significant positive impact

f. Sleeve length is significant, however, it affects the sale negatively (sold less)

g. Ruched and Beaded clothes make a negative impact on sale (sold less)

The multiple and adjusted R-squared values are 58% and 39% respectively. We can conclude that the model is quite robust and the p-value of almost 0 also suggests that there is definitely an impact of these variables on the total sales of the dresses.

```
> modeluseAll <- lm(Total.Sales~.-Dress_ID,data = dress.data)
> summary(modeluseAll)

Call:
lm(formula = Total.Sales ~ . - Dress_ID, data = dress.data)

Residuals:
   Min     1Q Median     3Q    Max 
-24359  -3794      0   1919  59098 

Coefficients: (7 not defined because of singularities)
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)               24704.90   17862.52   1.383  0.16754    
StyleBrief                 4874.30    3434.28   1.419  0.15671    
StyleCasual                2096.88    2402.84   0.873  0.38345    
Stylecute                  1600.70    2854.88   0.561  0.57537    
Stylefashion             -10126.75   18636.09  -0.543  0.58721    
StyleFlare                 2043.71    7842.89   0.261  0.79457    
StyleNovelty               1737.32    4475.92   0.388  0.69815    
StyleOL                   15385.85   27300.01   0.564  0.57340    
Styleparty                 2172.36    3140.72   0.692  0.48961    
Stylesexy                  5348.73    5862.32   0.912  0.36220    
StyleSexy                  5628.23    2685.67   2.096  0.03684 *  
Stylevintage               6121.19    3167.94   1.932  0.05415 .  
Stylework                  1804.85    3721.72   0.485  0.62802    
PriceAverage               6754.38   11387.99   0.593  0.55349    
Pricehigh                  7296.15   11803.76   0.618  0.53690    
PriceHigh                  5407.89   12330.37   0.439  0.66124    
```


```
Pricehigh                  7296.15   11803.76   0.618  0.53690    
PriceHigh                  5407.89   12330.37   0.439  0.66124    
```


```
Pricelow                   4451.73   11567.35   0.385  0.70058    
PriceLow                   8196.48   11492.35   0.713  0.47620    
PriceMedium                2799.93   11538.06   0.243  0.80841    
Pricevery-high             6631.61   11793.75   0.562  0.57428    
Rating                     1236.47     259.59   4.763 2.81e-06 ***
SizeL                      4104.71    1520.05   2.700  0.00727 ** 
SizeM                      1186.31    1355.53   0.875  0.38210    
Sizes                    -11441.93   10707.61  -1.069  0.28601    
SizeS                      1361.12    2112.62   0.644  0.51982    
Sizesmall                 -2138.72   11223.29  -0.191  0.84898    
SizeXL                     -178.18    3086.05  -0.058  0.95399    
SeasonAutomn              -1325.97   10030.47  -0.132  0.89491    
SeasonAutumn              -6846.28   10661.47  -0.642  0.52120    
Seasonspring              30907.99   12291.74   2.515  0.01237 *  
SeasonSpring              -1250.99   10004.19  -0.125  0.90056    
Seasonsummer               1324.23   13914.98   0.095  0.92424    
SeasonSummer              -2429.08    9961.45  -0.244  0.80749    
Seasonwinter              -7313.53   10098.55  -0.724  0.46942    
SeasonWinter              -2132.17    9997.57  -0.213  0.83124    
NeckLinebackless           3251.82   28887.56   0.113  0.91044    
NeckLineboat-neck         10384.41   25014.62   0.415  0.67830    
NeckLinebowneck           11068.04   25101.45   0.441  0.65954    
NeckLinehalter            12718.49   28886.01   0.440  0.66000    
NeckLinemandarin-collor         NA         NA      NA       NA    
NeckLineNULL               4820.35   26187.90   0.184  0.85407    
NeckLineo-neck            11931.11   24815.93   0.481  0.63097    
NeckLineopen              19152.01   29539.88   0.648  0.51719    
NeckLinepeterpan-collor    9690.80   25403.58   0.381  0.70309    
```

```
NeckLineNULL                    4820.35   26187.90   0.184  0.85407
NeckLineo-neck                 11931.11   24815.93   0.481  0.63097
NeckLineopen                   19152.01   29539.88   0.648  0.51719
NeckLinepeterpan-collor         9690.80   25403.58   0.381  0.70309
NeckLineruffled               150944.44   26865.54   5.619 3.97e-08 ***
NeckLineScoop                   8122.53   28961.38   0.280  0.77929
NeckLineslash-neck              8527.08   24942.76   0.342  0.73266
NeckLinesqare-collor            5682.05   24620.26   0.231  0.81762
NeckLinesweetheart             13752.76   29058.81   0.473  0.63632
NeckLineSweetheart             12628.88   24653.51   0.512  0.60880
NeckLineturndowncollor         14496.57   25084.01   0.578  0.56369
NeckLinev-neck                 11237.33   24864.52   0.452  0.65159
SleeveLengthcap-sleeves       -35827.44   13222.66  -2.710  0.00707 **
SleeveLengthcapsleeves        -33486.85   11988.62  -2.793  0.00551 **
SleeveLengthfull              -31076.84   10695.68  -2.906  0.00390 **
SleeveLengthhalf              -28542.05   18382.71  -1.553  0.12142
SleeveLengthhalfsleeve        -29590.10   10755.18  -2.751  0.00625 **
SleeveLengthNULL              -23154.81   13742.64  -1.685  0.09291 .
SleeveLengthPetal              12969.91   18298.38   0.709  0.47893
SleeveLengthshort             -32132.46   10530.82  -3.051  0.00246 **
SleeveLengthsleeevless        -41724.65   12768.75  -3.268  0.00119 **
SleeveLengthsleeveless        -34322.26   11836.31  -2.900  0.00397 **
SleeveLengthsleevless         -32975.66   10616.89  -3.106  0.00205 **
SleeveLengthsleveless         -38905.07   14646.61  -2.656  0.00827 **
SleeveLengththreequarter      -23959.51   11007.41  -2.177  0.03018 *
SleeveLengththreequater       -38754.71   17350.23  -2.234  0.02615 *
SleeveLengththressqatar       -33444.60   11122.03  -3.007  0.00283 **
SleeveLengthturndowncollor    -34870.83   14654.40  -2.380  0.01788 *
SleeveLengthurndowncollor     -39645.37   14621.42  -2.711  0.00703 **
```

```
FabricTypetulle                 3541.06    8776.30   0.403  0.68685
FabricTypewollen               -4959.61    7691.19  -0.645  0.51946
FabricTypewoolen                7088.33   11921.44   0.595  0.55251
FabricTypeworsted                    NA         NA      NA       NA
Decorationapplique            -21074.89   11871.32  -1.775  0.07673 .
Decorationbeading             -25944.45   11862.27  -2.187  0.02940 *
Decorationbow                 -19811.70   11810.51  -1.677  0.09436 .
Decorationbutton              -22991.52   12038.63  -1.910  0.05699 .
Decorationcascading           -16928.13   15219.10  -1.112  0.26679
Decorationcrystal             -19369.51   15275.51  -1.268  0.20565
Decorationdraped              -20105.50   15235.98  -1.320  0.18784
Decorationembroidary          -17507.98   12694.60  -1.379  0.16874
Decorationfeathers            -24561.89   13762.62  -1.785  0.07519 .
Decorationflowers             -25288.59   13450.45  -1.880  0.06093 .
Decorationhollowout           -22947.00   11736.28  -1.955  0.05136 .
Decorationlace                -21155.65   11541.54  -1.833  0.06766 .
Decorationnone                -21462.70   13654.65  -1.572  0.11691
Decorationnull                -22178.38   11465.58  -1.934  0.05389 .
Decorationpearls              -21041.40   15726.74  -1.338  0.18180
Decorationplain               -25071.10   15177.87  -1.652  0.09948 .
Decorationpleat               -33444.01   18586.91  -1.799  0.07284 .
Decorationpockets             -23620.37   12250.13  -1.928  0.05465 .
Decorationrivet               -21703.97   13017.56  -1.667  0.09637 .
Decorationruched              -26759.17   13279.98  -2.015  0.04468 *
Decorationruffles             -17410.62   11824.78  -1.472  0.14183
Decorationsashes              -21381.24   11472.89  -1.864  0.06322 .
Decorationsequined            -21648.81   11811.62  -1.833  0.06769 .
Decorationtassel              -24952.86   15532.44  -1.607  0.10908
DecorationTiered                     NA         NA      NA       NA
```

```
Decorationruffles            -17410.62   11824.78  -1.472  0.14183
Decorationsashes             -21381.24   11472.89  -1.864  0.06322 .
Decorationsequined           -21648.81   11811.62  -1.833  0.06769 .
Decorationtassel             -24952.86   15532.44  -1.607  0.10908
DecorationTiered                    NA         NA      NA       NA
Pattern.Typeanimal             6236.87    3883.37   1.606  0.10918
Pattern.Typecharacter          1099.33   10148.98   0.108  0.91381
Pattern.Typedot                 677.12    4353.83   0.156  0.87650
Pattern.Typefloral             5483.85    7575.11   0.724  0.46960
Pattern.Typegeometric         10551.01    5877.26   1.795  0.07349 .
Pattern.Typeleapord          -10334.57   10685.84  -0.967  0.33416
Pattern.Typeleopard             689.72    6453.58   0.107  0.91495
Pattern.Typenone              -1705.51   10490.17  -0.163  0.87094
Pattern.Typenull               -533.61    3155.70  -0.169  0.86582
Pattern.Typepatchwork          -785.16    3252.53  -0.241  0.80939
Pattern.Typeplaid             -2296.55    6593.65  -0.348  0.72783
Pattern.Typeprint              1437.73    3159.24   0.455  0.64933
Pattern.Typesolid              2140.53    2900.38   0.738  0.46101
Pattern.Typesplice            -1221.36   10750.99  -0.114  0.90962
Pattern.Typestriped                 NA         NA      NA       NA
Recommendation                 1371.84    1091.22   1.257  0.20954
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9586 on 345 degrees of freedom
Multiple R-squared:  0.5802,	Adjusted R-squared:  0.3929
F-statistic: 3.097 on 154 and 345 DF,  p-value: < 2.2e-16
```

5. To regularize the rating procedure and find its efficiency, the store wants to find if the rating of the dress affects the total sales.

To find the relation between rating and total sales (both are numerical variables), perform a correlation of the two attributes.

Code:

```
attach(dress.data)
cor.test(Total.Sales,Rating)
```

Result:

It can be clearly seen from the result that there is almost no correlation between the two variables. The correlation value is 0.2, which shows a very weak positive association, that is, a higher rating correlates with higher sales. Thus, the rating process has to be regularized.