Assignment6

Yiming Ge

10/29/2020

Problem1 Progresso Soup Sales read&check data

```
Soup <- read.csv("https://raw.githubusercontent.com/jcbonilla/BusinessAnalytics/master/BAData/Progresso_
head(Soup)
```

```
##   IRI_KEY Month  Region Low_Income High_Income Price.Campbell Price.PL
## 1  289854    12   South          0           0          1.494    1.102
## 2  534863     2 MidWest          0           1          1.594    1.053
## 3  683960    12   South          0           1          1.445    0.906
## 4  257658     6 MidWest          0           0          1.670    1.169
## 5  210202     2   South          1           0          1.278    0.833
## 6  534005    12   South          1           0          1.294    0.982
##   Price.Progresso Sales.Progresso Category_Sales
## 1           1.718         921.716      5663.5959
## 2           1.306        1309.433      3700.0734
## 3           1.700         911.686     10253.6026
## 4           2.051          18.844       806.6891
## 5           1.692         142.470      2837.7123
## 6           1.654         323.972      3717.7864
```

```
dim(Soup)
```

```
## [1] 59100    10
```

```
str(Soup)
```

```
## 'data.frame':    59100 obs. of  10 variables:
##  $ IRI_KEY        : int  289854 534863 683960 257658 210202 534005 279734 650517 1130099 684023 ...
##  $ Month          : int  12 2 12 6 2 12 6 3 8 3 ...
##  $ Region         : chr  "South" "MidWest" "South" "MidWest" ...
##  $ Low_Income     : int  0 0 0 0 1 1 0 1 0 0 ...
##  $ High_Income    : int  0 1 1 0 0 0 0 0 1 0 ...
##  $ Price.Campbell : num  1.49 1.59 1.45 1.67 1.28 ...
##  $ Price.PL       : num  1.102 1.053 0.906 1.169 0.833 ...
##  $ Price.Progresso: num  1.72 1.31 1.7 2.05 1.69 ...
##  $ Sales.Progresso: num  921.7 1309.4 911.7 18.8 142.5 ...
##  $ Category_Sales : num  5664 3700 10254 807 2838 ...
```

```r
summary(Soup)
```

```
##     IRI_KEY            Month           Region            Low_Income
##  Min.   : 200039   Min.   : 1.000   Length:59100       Min.   :0.0000
##  1st Qu.: 239069   1st Qu.: 4.000   Class :character   1st Qu.:0.0000
##  Median : 265932   Median : 7.000   Mode  :character   Median :0.0000
##  Mean   : 359738   Mean   : 6.747                      Mean   :0.2019
##  3rd Qu.: 291654   3rd Qu.:10.000                      3rd Qu.:0.0000
##  Max.   :8032406   Max.   :12.000                      Max.   :1.0000
##   High_Income     Price.Campbell     Price.PL     Price.Progresso
##  Min.   :0.0000   Min.   :0.418   Min.   :0.259   Min.   :0.721
##  1st Qu.:0.0000   1st Qu.:1.280   1st Qu.:0.989   1st Qu.:1.382
##  Median :0.0000   Median :1.417   Median :1.096   Median :1.640
##  Mean   :0.1981   Mean   :1.432   Mean   :1.118   Mean   :1.648
##  3rd Qu.:0.0000   3rd Qu.:1.573   3rd Qu.:1.224   3rd Qu.:1.890
##  Max.   :1.0000   Max.   :2.519   Max.   :3.180   Max.   :2.962
##  Sales.Progresso    Category_Sales
##  Min.   :    1.16   Min.   :  115.2
##  1st Qu.:  261.56   1st Qu.: 2484.3
##  Median :  659.40   Median : 4194.1
##  Mean   : 1403.62   Mean   : 5662.6
##  3rd Qu.: 1605.13   3rd Qu.: 7102.7
##  Max.   :53857.27   Max.   :81870.1
```

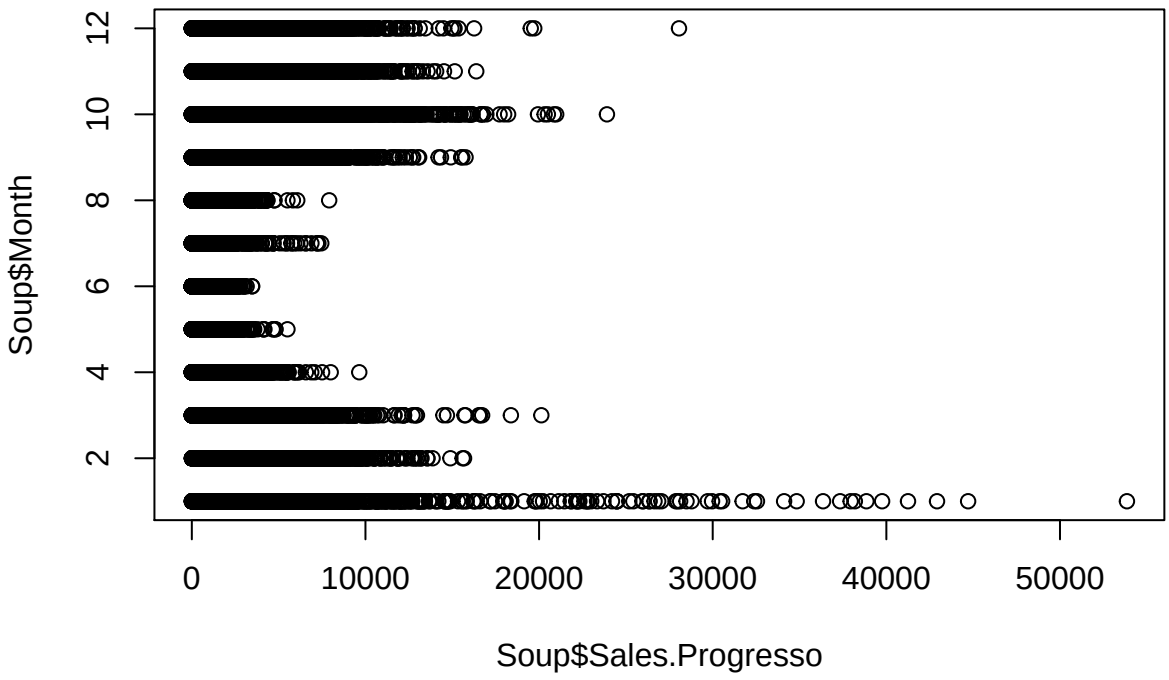
creat winter dummy variable

```r
Soup$winter <- ifelse(Soup$Month == 10|Soup$Month == 11|Soup$Month == 12|Soup$Month == 1|Soup$Month == 
table(Soup$winter)
```

```
##
##     0     1
## 34342 24758
```

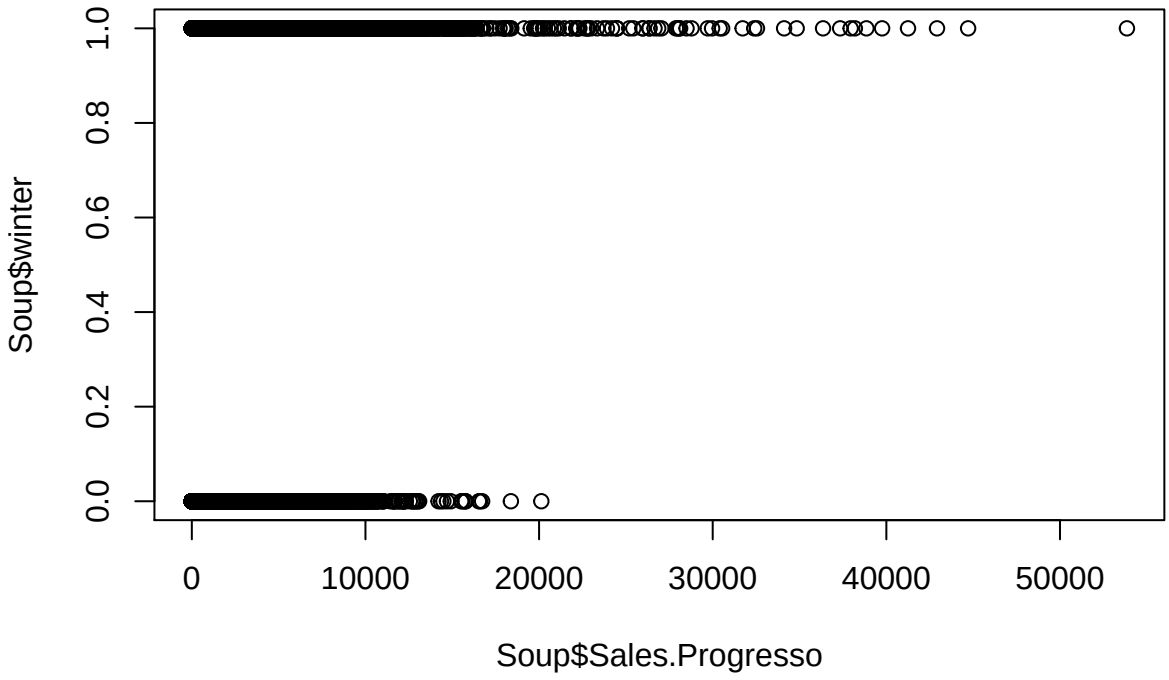
We have 24758 winter month data and 34342 non-winter month data

```r
plot(Soup$Sales.Progresso,Soup$Month)
```



Sales decreased from Jan to Jun and increase from Jun to Dec.

```
plot(Soup$Sales.Progresso,Soup$winter)
```



Sales are much higher in winter month.

```
aggdata <-aggregate(Soup$Sales.Progresso, by=list(Soup$winter),
  FUN=sum, na.rm=TRUE)
winter_percent = aggdata[2,2]/(aggdata[1,2]+aggdata[2,2])
nonwinter_percent = aggdata[1,2]/(aggdata[1,2]+aggdata[2,2])
winter_percent
```

```
## [1] 0.653823
```

```
nonwinter_percent
```

```
## [1] 0.346177
```

winter is 0.653823 and nonwinter is 0.346177

```
soupModel<-lm(Soup$Sales.Progresso~Soup$Month+Soup$Region+Soup$Low_Income+Soup$High_Income+Soup$Price.Ca
summary(soupModel)
```

```
##
## Call:
## lm(formula = Soup$Sales.Progresso ~ Soup$Month + Soup$Region +
##     Soup$Low_Income + Soup$High_Income + Soup$Price.Campbell +
##     Soup$Price.PL + Soup$Price.Progresso + Soup$Category_Sales +
##     Soup$winter)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10002.5   -482.6    -32.2    423.7  30569.8
##
## Coefficients:
##                          Estimate Std. Error  t value Pr(>|t|)
## (Intercept)          -8.561e+02  4.272e+01  -20.041   <2e-16 ***
## Soup$Month           -2.564e+01  1.299e+00  -19.731   <2e-16 ***
## Soup$RegionMidWest   -1.374e+03  1.411e+01  -97.400   <2e-16 ***
## Soup$RegionSouth     -6.585e+02  1.299e+01  -50.696   <2e-16 ***
## Soup$RegionWest      -7.359e+02  1.370e+01  -53.718   <2e-16 ***
## Soup$Low_Income      -4.629e+00  1.141e+01   -0.406    0.685
## Soup$High_Income      2.241e+02  1.151e+01   19.474   <2e-16 ***
## Soup$Price.Campbell   1.991e+03  2.378e+01   83.723   <2e-16 ***
## Soup$Price.PL         8.625e+02  2.490e+01   34.634   <2e-16 ***
## Soup$Price.Progresso -1.570e+03  1.516e+01 -103.574   <2e-16 ***
## Soup$Category_Sales   3.158e-01  1.073e-03  294.259   <2e-16 ***
## Soup$winter           9.917e+01  1.017e+01    9.752   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1057 on 59088 degrees of freedom
## Multiple R-squared:  0.7555, Adjusted R-squared:  0.7555
## F-statistic: 1.66e+04 on 11 and 59088 DF,  p-value: < 2.2e-16
```

From summary we can see R square is 0.755 which means almost 76% data fit in the regression model. We should drop low_income since it is not significant(no * and very large p-value) Realtionship:(keep other variables unchanged) Month:given the month number, the sales will drop the given number times 25.64 dollar RegionMidWest: If the store locates in MidWest, the sales will drop 1374 dollar RegionSouth:If the store locates in South, the sales will drop 658.5 dollar RegionWest:If the store locates in West, the sales will drop 735.9 dollar HighIncome:If the store locates in HighIncome area,the sales will increase 224.1dollar Price.campbell:one dollar increase in campbell, the sales will increase 1991 dollar price.PL: one dollar increase in PL, the sales will increase 862.5 dollar price.progresso: one dollar increase in progresso, the sales will drop 1570 dollar Category_sales: one dollar increase in category_sales will increase the sales 0.3158 dollar winter: If right now is winter season, the sales will increase 99.17 dollar

```
#split the data into training and test data
library(dplyr)
```

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

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

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

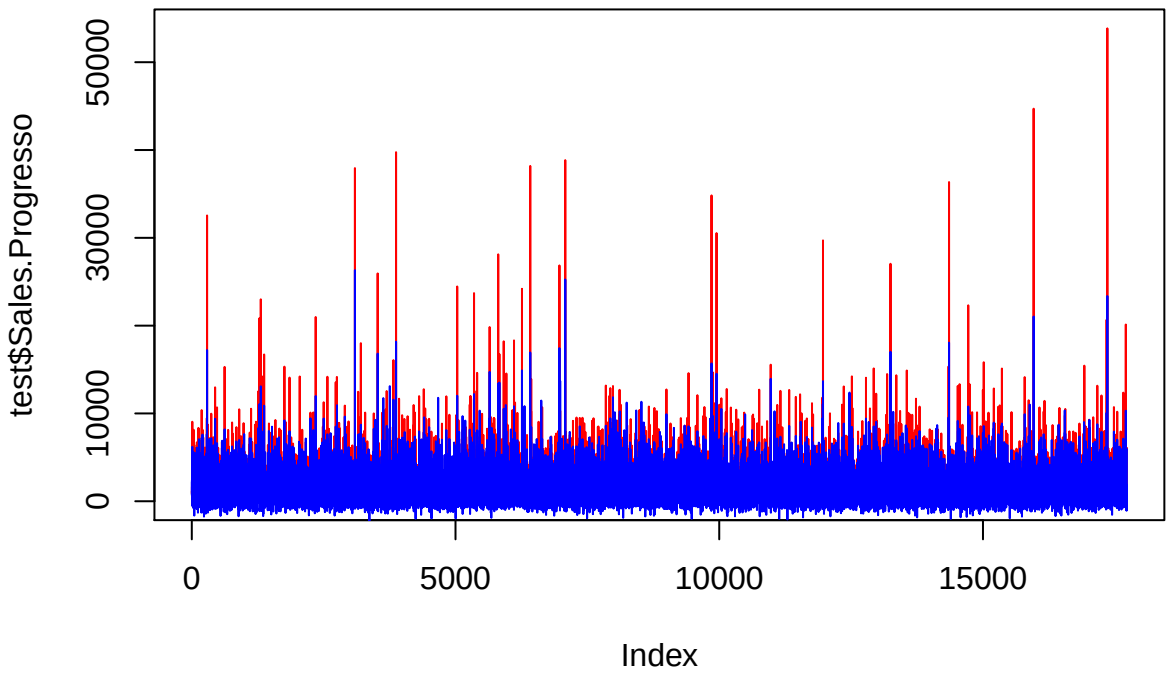
```r
train<-sample_frac(Soup, 0.7)
sid<-as.numeric(rownames(train)) # because rownames() returns character
test<-Soup[-sid,]
#train
train.model<- lm(Sales.Progresso~.,data=train)
summary(train.model)
```

```
##
## Call:
## lm(formula = Sales.Progresso ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9169.1  -485.6   -32.7   428.1 25279.6
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -8.441e+02  5.131e+01 -16.453  < 2e-16 ***
## IRI_KEY         -1.130e-04  1.464e-05  -7.718 1.21e-14 ***
## Month           -2.522e+01  1.554e+00 -16.227  < 2e-16 ***
## RegionMidWest   -1.380e+03  1.687e+01 -81.820  < 2e-16 ***
## RegionSouth     -6.528e+02  1.555e+01 -41.989  < 2e-16 ***
## RegionWest      -7.390e+02  1.636e+01 -45.166  < 2e-16 ***
## Low_Income      -1.187e+01  1.370e+01  -0.866    0.386
## High_Income      2.310e+02  1.377e+01  16.770  < 2e-16 ***
## Price.Campbell   2.016e+03  2.865e+01  70.359  < 2e-16 ***
## Price.PL         8.589e+02  2.994e+01  28.687  < 2e-16 ***
## Price.Progresso -1.580e+03  1.818e+01 -86.928  < 2e-16 ***
## Category_Sales   3.174e-01  1.284e-03 247.298  < 2e-16 ***
## winter           1.039e+02  1.217e+01   8.542  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1059 on 41357 degrees of freedom
## Multiple R-squared:  0.757,  Adjusted R-squared:  0.7569
## F-statistic: 1.073e+04 on 12 and 41357 DF,  p-value: < 2.2e-16
```

```r
#prediction
pred<-predict(train.model,test)
head(pred)
```

```
##     41371     41372     41373     41374     41375     41376
## 2349.6598  676.9060 2180.3074 -411.8449 1244.9445 3655.8584
```

```r
#compare predicted value and actual value
plot(test$Sales.Progresso,type='l',lty=1.8,col='red')
lines(pred,type='l',col='blue')
```




```
#find accuracy
#find accuracy
rmse<-sqrt(mean(pred-Soup$Sales.Progresso)^2)
```

```
## Warning in pred - Soup$Sales.Progresso: longer object length is not a multiple
## of shorter object length
```

```
rmse #is the square root of the average of squared differences between the prediction and the actual ob
```

```
## [1] 6.278593
```

Problem2 Diamond Quotes read data

```
Diamonds <- read.csv("https://raw.githubusercontent.com/jcbonilla/BusinessAnalytics/master/BAData/Diamon
head(Diamonds)
```

```
##   Carat Colour Clarity Cut. Certification Polish Symmetry Price Wholesaler
## 1  0.92      I     SI2    G           AGS      V        V  3000          1
## 2  0.92      I     SI2    V           AGS      G        G  3000          1
## 3  0.82      F     SI2    I           GIA      X        X  3004          1
## 4  0.81      G     SI1    I           GIA      X        V  3004          1
## 5  0.90      J     VS2    V           GIA      V        V  3006          1
## 6  0.87      F     SI2    I           AGS      G        V  3007          1
```

```r
dim(Diamonds)
```

```
## [1] 440   9
```

```r
str(Diamonds)
```

```
## 'data.frame':    440 obs. of  9 variables:
##  $ Carat        : num  0.92 0.92 0.82 0.81 0.9 0.87 0.8 0.84 0.8 0.8 ...
##  $ Colour       : chr  "I" "I" "F" "G" ...
##  $ Clarity      : chr  "SI2" "SI2" "SI2" "SI1" ...
##  $ Cut.         : chr  "G" "V" "I" "I" ...
##  $ Certification: chr  "AGS" "AGS" "GIA" "GIA" ...
##  $ Polish       : chr  "V" "G" "X" "X" ...
##  $ Symmetry     : chr  "V" "G" "X" "V" ...
##  $ Price        : int  3000 3000 3004 3004 3006 3007 3008 3010 3012 3012 ...
##  $ Wholesaler   : int  1 1 1 1 1 1 1 1 1 1 ...
```

```r
summary(Diamonds)
```

```
##      Carat           Colour            Clarity             Cut.          
##  Min.   :0.0900   Length:440         Length:440         Length:440        
##  1st Qu.:0.3000   Class :character   Class :character   Class :character  
##  Median :0.8100   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :0.6693                                                           
##  3rd Qu.:1.0100                                                           
##  Max.   :1.5800                                                           
##  Certification         Polish            Symmetry            Price      
##  Length:440         Length:440         Length:440         Min.   : 160  
##  Class :character   Class :character   Class :character   1st Qu.: 520  
##  Mode  :character   Mode  :character   Mode  :character   Median :2169  
##                                                           Mean   :1717  
##                                                           3rd Qu.:3012  
##                                                           Max.   :3145  
##    Wholesaler   
##  Min.   :1.000  
##  1st Qu.:2.000  
##  Median :2.000  
##  Mean   :2.318  
##  3rd Qu.:3.000  
##  Max.   :3.000
```

bulid the model

```r
diamonds_model<-lm(Price~.,data=Diamonds)
summary(diamonds_model)
```

```
##
## Call:
## lm(formula = Price ~ ., data = Diamonds)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -579.01  -76.75  -12.12   69.83  542.76
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         77.84     176.47   0.441 0.659388
## Carat             3166.60      95.78  33.062  < 2e-16 ***
## ColourE           -153.33      46.02  -3.332 0.000942 ***
## ColourF           -241.78      46.23  -5.230 2.73e-07 ***
## ColourG           -206.97      47.36  -4.370 1.58e-05 ***
## ColourH           -313.18      46.52  -6.732 5.76e-11 ***
## ColourI           -365.07      46.89  -7.786 5.82e-14 ***
## ColourJ           -433.72      49.74  -8.721  < 2e-16 ***
## ColourK           -685.24      57.84 -11.846  < 2e-16 ***
## ColourL           -894.94      69.33 -12.909  < 2e-16 ***
## ClarityI2         -586.07      46.29 -12.661  < 2e-16 ***
## ClaritySI1         590.38      43.20  13.667  < 2e-16 ***
## ClaritySI2         506.65      35.68  14.200  < 2e-16 ***
## ClaritySI3         264.04      43.39   6.085 2.70e-09 ***
## ClarityVS1         689.26      57.69  11.948  < 2e-16 ***
## ClarityVS2         637.47      51.14  12.466  < 2e-16 ***
## ClarityVVS1        927.50     136.90   6.775 4.40e-11 ***
## ClarityVVS2        651.49      89.25   7.300 1.54e-12 ***
## Cut.G               64.65      35.90   1.801 0.072431 .
## Cut.I               93.45      35.07   2.665 0.008012 **
## Cut.V               67.53      35.64   1.895 0.058855 .
## Cut.X              113.17      30.94   3.657 0.000289 ***
## CertificationDOW  -275.84     192.41  -1.434 0.152460
## CertificationEGL   -63.33      75.94  -0.834 0.404815
## CertificationGIA    89.08      70.27   1.268 0.205658
## CertificationIGI   105.36      82.26   1.281 0.200947
## PolishG            182.63      91.51   1.996 0.046619 *
## PolishI            376.89     134.59   2.800 0.005352 **
## Polishv            120.75     198.09   0.610 0.542487
## PolishV            192.44      94.85   2.029 0.043132 *
## PolishX            200.20      97.50   2.053 0.040689 *
## SymmetryG           74.00      49.26   1.502 0.133810
## SymmetryI              NA         NA      NA       NA
## SymmetryV           83.76      52.60   1.592 0.112084
## SymmetryX           72.94      58.63   1.244 0.214201
## Wholesaler        -395.49      32.45 -12.187  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 169 on 405 degrees of freedom
## Multiple R-squared:  0.9809, Adjusted R-squared:  0.9793
## F-statistic:   613 on 34 and 405 DF,  p-value: < 2.2e-16
```

Keep other variables unchanged Carat : Every increase in 1 carat the price increases by $3166.60 Color E: Price decreases by $153.33 with respect to category D Color F: Price decreases by $241.78 with respect to category D Color G: Price decreases by $206.97 with respect to category D Color H: Price decreases by $313.18 with respect to category D Color I: Price decreases by $365.07 with respect to category D Color J: Price decreases by $433.72 with respect to category D Color K: Price decreases by $685.24 with respect to category D Color L: Price decreases by $894.94 with respect to category D Clarity I2: Price decreases by

$586.07 with respect to Clarity I1 Clarity SI1: Price increases by $590.38 with respect to Clarity I1 Clarity SI2: Price increases by $506.65 with respect to Clarity I1 Clarity SI3: Price increases by $264.04 with respect to Clarity I1 Clarity VS1: Price increases by $689.26 with respect to Clarity I1 Clarity VS2: Price increases by $637.47 with respect to Clarity I1 Clarity VVS1: Price increases by $927.50 with respect to Clarity I1 Clarity VVS2: Price increases by $651.49 with respect to Clarity I1 Cut.Good: Price increases by $64.65 with respect to Cut Fair Cut.Ideal: Price increases by $93.45 with respect to Cut Fair Cut.Very Good: Price increases by $67.53 with respect to Cut Fair Cut.Excellent: Price increases by $113.17 with respect to Cut Fair CertificationDOW: Price decreases by $275.84 with respect to Certification AGS CertificationEGL: Price decreases by $63.33 with respect to Certification AGS CertificationGIA: Price increases by $89.08 with respect to Certification AGS CertificationIGI: Price decreases by $105.36 with respect to Certification AGS PolishGood: Price increases by $182.63 with respect to Polish Fair PolishIdeal: Price increases by $376.89 with respect to Polish Fair PolishVery Very good: Price increases by $120.75 with respect to Polish Fair PolishVery good: Price increases by $192.44 with respect to Polish Fair PolishExcellent: Price increases by $200.20 with respect to Polish Fair SymmetryGood: Price increases by $74.00 with respect to Symmetry Fair SymmetryVery Good: Price increases by $83.76 with respect to Symmetry Fair SymmetryExcellent: Price increases by $72.94 with respect to Symmetry Fair Wholesaler: Given the number of wholesaler, the price will decrease the given number times 395.49

```
diamond1<-data.frame(Price=3100,Carat=0.9,Cut. ='V',Colour = 'J', Clarity = 'SI2',Polish='G',Symmetry='
diamond2<-data.frame(Price=3100,Carat=0.9,Cut. ='V',Colour = 'J', Clarity = 'SI2',Polish='G',Symmetry='
diamond3<-data.frame(Price=3100,Carat=0.9,Cut. ='V',Colour = 'J', Clarity = 'SI2',Polish='G',Symmetry='

predict(diamonds_model,diamond1)
```

```
## Warning in predict.lm(diamonds_model, diamond1): prediction from a rank-
## deficient fit may be misleading
```

```
##        1
## 3028.221
```

```
predict(diamonds_model,diamond2)
```

```
## Warning in predict.lm(diamonds_model, diamond2): prediction from a rank-
## deficient fit may be misleading
```

```
##        1
## 2632.732
```

```
predict(diamonds_model,diamond3)
```

```
## Warning in predict.lm(diamonds_model, diamond3): prediction from a rank-
## deficient fit may be misleading
```

```
##        1
## 2237.244
```

The r-squared is 0.98 which means almost 98% data fit in the regression model. Since we do not have information with wholsealer, we try all the three conditions. We find no matter what wholesaler we choose, the diamond is always overpriced.

```r
diamonds_model2<-lm(Price~Carat+Colour+Clarity+Cut.+Certification+Polish+Symmetry,data=Diamonds)
summary(diamonds_model2)
```

```
##
## Call:
## lm(formula = Price ~ Carat + Colour + Clarity + Cut. + Certification +
##     Polish + Symmetry, data = Diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -760.88  -83.67  -18.01  101.68  690.91
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1430.38     146.89  -9.737  < 2e-16 ***
## Carat             4202.98      51.46  81.677  < 2e-16 ***
## ColourE           -191.18      53.61  -3.566 0.000406 ***
## ColourF           -309.40      53.59  -5.773 1.55e-08 ***
## ColourG           -302.01      54.54  -5.537 5.53e-08 ***
## ColourH           -432.14      53.12  -8.136 5.04e-15 ***
## ColourI           -502.96      53.13  -9.467  < 2e-16 ***
## ColourJ           -637.71      54.69 -11.661  < 2e-16 ***
## ColourK           -987.33      61.03 -16.179  < 2e-16 ***
## ColourL          -1174.60      76.39 -15.377  < 2e-16 ***
## ClarityI2         -777.74      50.84 -15.299  < 2e-16 ***
## ClaritySI1         860.54      43.29  19.877  < 2e-16 ***
## ClaritySI2         731.99      35.63  20.543  < 2e-16 ***
## ClaritySI3         388.67      49.24   7.893 2.77e-14 ***
## ClarityVS1        1027.21      59.06  17.391  < 2e-16 ***
## ClarityVS2         917.77      53.33  17.210  < 2e-16 ***
## ClarityVVS1       1343.74     154.80   8.681  < 2e-16 ***
## ClarityVVS2        931.81     100.69   9.254  < 2e-16 ***
## Cut.G               56.33      41.91   1.344 0.179679
## Cut.I               95.73      40.95   2.338 0.019874 *
## Cut.V               83.85      41.59   2.016 0.044425 *
## Cut.X               57.05      35.73   1.597 0.111084
## CertificationDOW  -499.67     223.64  -2.234 0.026008 *
## CertificationEGL  -416.84      81.95  -5.087 5.58e-07 ***
## CertificationGIA   -64.40      80.72  -0.798 0.425450
## CertificationIGI    -8.05      95.43  -0.084 0.932812
## PolishG            211.41     106.81   1.979 0.048460 *
## PolishI            460.81     156.95   2.936 0.003514 **
## Polishv            262.28     230.90   1.136 0.256672
## PolishV            226.82     110.71   2.049 0.041119 *
## PolishX            236.09     113.80   2.075 0.038648 *
## SymmetryG          108.22      57.43   1.885 0.060208 .
## SymmetryI              NA         NA      NA       NA
## SymmetryV          117.92      61.33   1.923 0.055225 .
## SymmetryX          111.20      68.36   1.627 0.104589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 197.3 on 406 degrees of freedom
```

```
## Multiple R-squared:  0.9739, Adjusted R-squared:  0.9718
## F-statistic: 459.9 on 33 and 406 DF,  p-value: < 2.2e-16
```

```
diamond <-data.frame(Price=3100,Carat=0.9,Cut. ='V',Colour = 'J', Clarity = 'SI2',Polish='G',Symmetry='
predict(diamonds_model2,diamond)
```

```
## Warning in predict.lm(diamonds_model2, diamond): prediction from a rank-
## deficient fit may be misleading
```

```
##       1
## 2795.37
```

The diamond is still overpriced. From the summary data we can say the first model have better 'goodness of fit' since it has higher r squared but we cannot say it is more 'correctness' because regression is a statistical relationship which does not imply causation.