Hw5

Problem 1: Progresso Soup Sales 1. a)

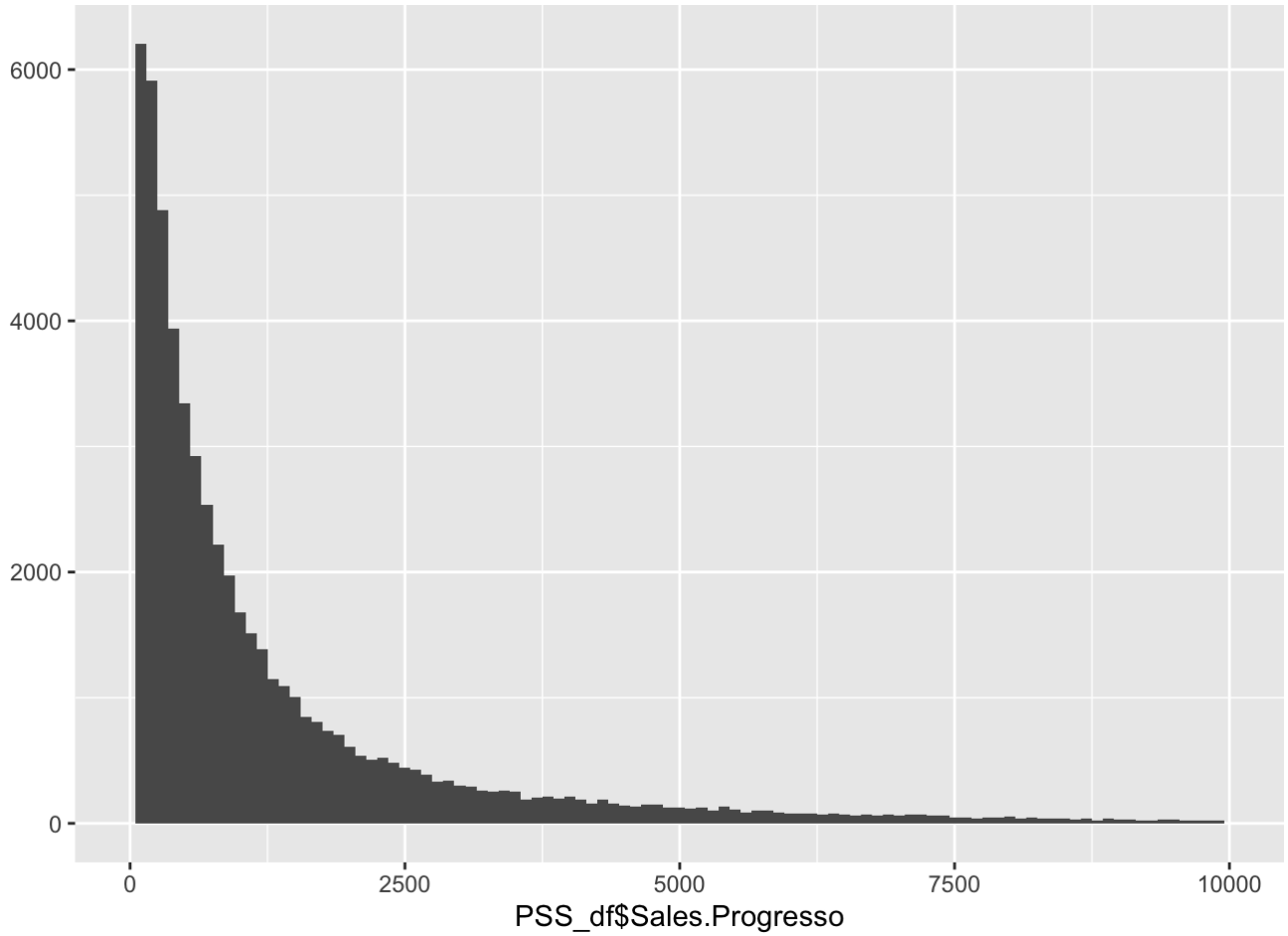
```
url = "https://raw.githubusercontent.com/jcbonilla/BusinessAnalytics/master/BAData/Progr
esso_Soup.csv"
PSS = read.csv(url)
PSS_df = as.data.frame(PSS)
PSS_df$winter <- ifelse(PSS_df$Month==10|PSS_df$Month==11|PSS_df$Month==12|PSS_df$Month=
=1|PSS_df$Month==2,as.logical(1), as.logical(0))
table(PSS_df$winter)
```

```
##
## FALSE  TRUE
## 34342 24758
```

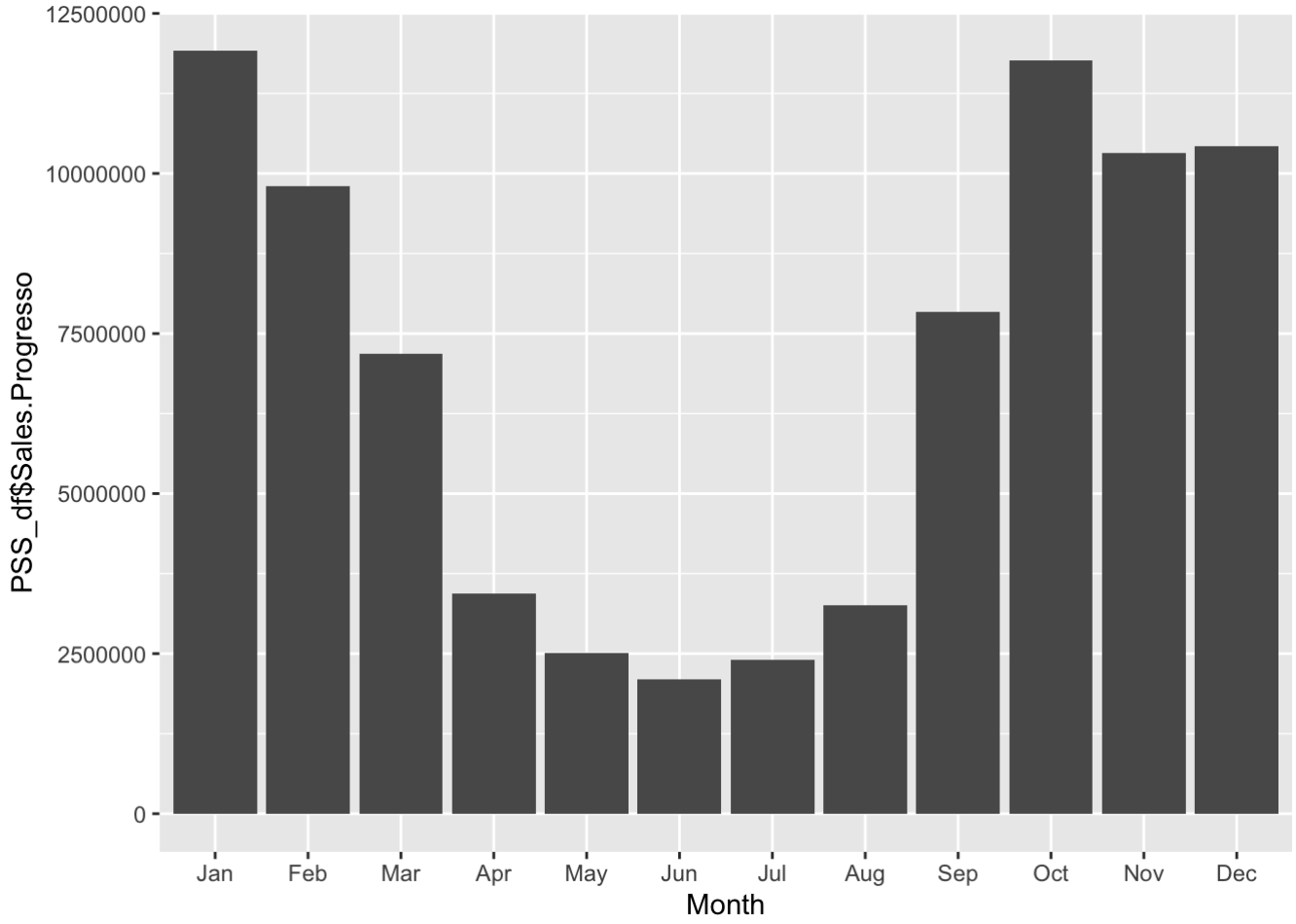
```
PSS_df$Month = factor(PSS_df$Month, levels = 1:12, labels = c("Jan","Feb","Mar","Apr",
                                                              "May","Jun","Jul","Aug",
                                                              "Sep","Oct","Nov","Dec"))

library(ggplot2)
library(dplyr)

qplot(PSS_df$Sales.Progresso, geom="histogram", binwidth=100,xlim=c(0,10000))
```

```
## Warning: Removed 553 rows containing non-finite values (stat_bin).
```

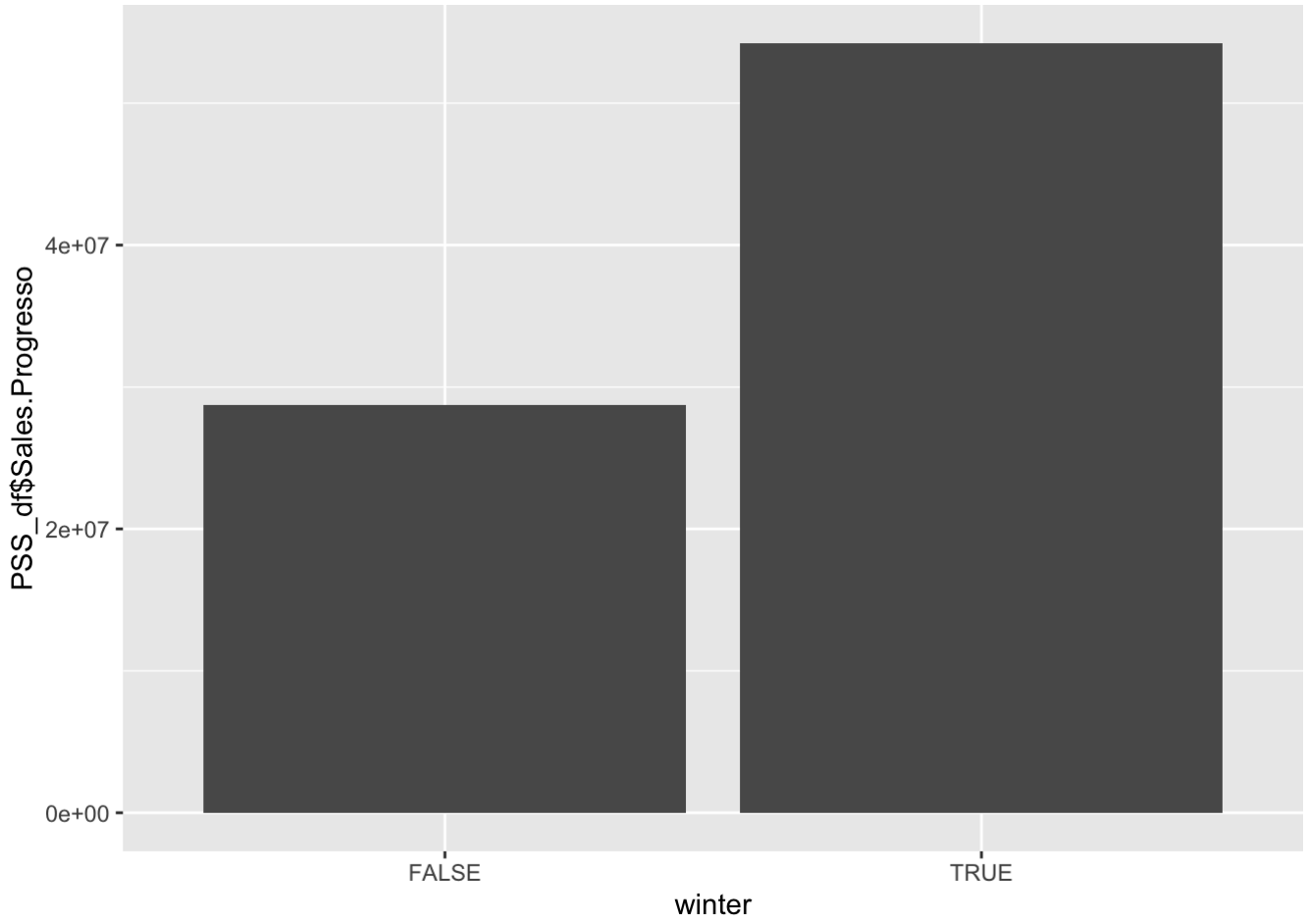
```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



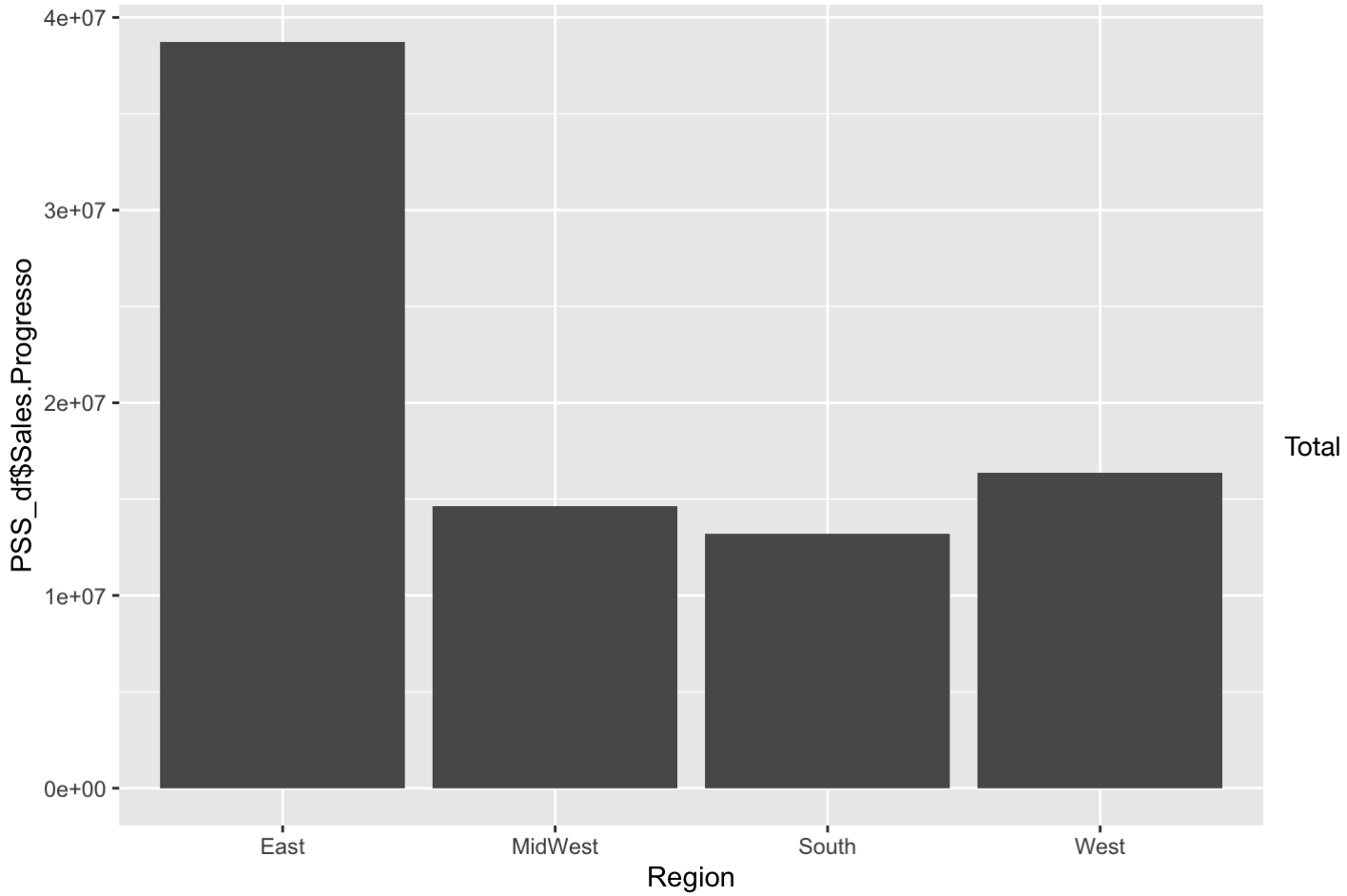

```
ggplot(PSS_df, aes(x=Month, y=PSS_df$Sales.Progresso)) + stat_summary(fun.y="sum"
                                                                       , geom="bar"
)
```




```
ggplot(PSS_df, aes(x=winter, y=PSS_df$Sales.Progresso)) + stat_summary(fun.y="sum"
                                                                       , geom="bar"
)
```




```
ggplot(PSS_df, aes(x=Region, y=PSS_df$Sales.Progresso)) + stat_summary(fun.y="sum"
                                                                       , geom="bar"
)
```



progresso sales are higher during winter time but lower during summer time. Also, east reagion contributed most progresso sales.

b. Winter monthes contributed much higher total progresso sales compared to non-winter monthes.

c.

```
market_share_winter = sum(PSS_df[which(PSS_df$winter),9])/sum(PSS_df[which(PSS_df$winte
r),10])
market_share_winter
```

```
## [1] 0.2841573
```

```
market_share_non_winter = sum(PSS_df[which(!PSS_df$winter),9])/sum(PSS_df[which(!PSS_df
$winter),10])
market_share_non_winter
```

```
## [1] 0.1997194
```

2.

```
model = lm(PSS_df$Sales.Progresso ~ PSS_df$Region+PSS_df$Low_Income+PSS_df$High_Income+P
SS_df$Price.Campbell
+PSS_df$Price.PL+PSS_df$Price.Progresso+PSS_df$winter,data = PSS_df)
summary(model)
```

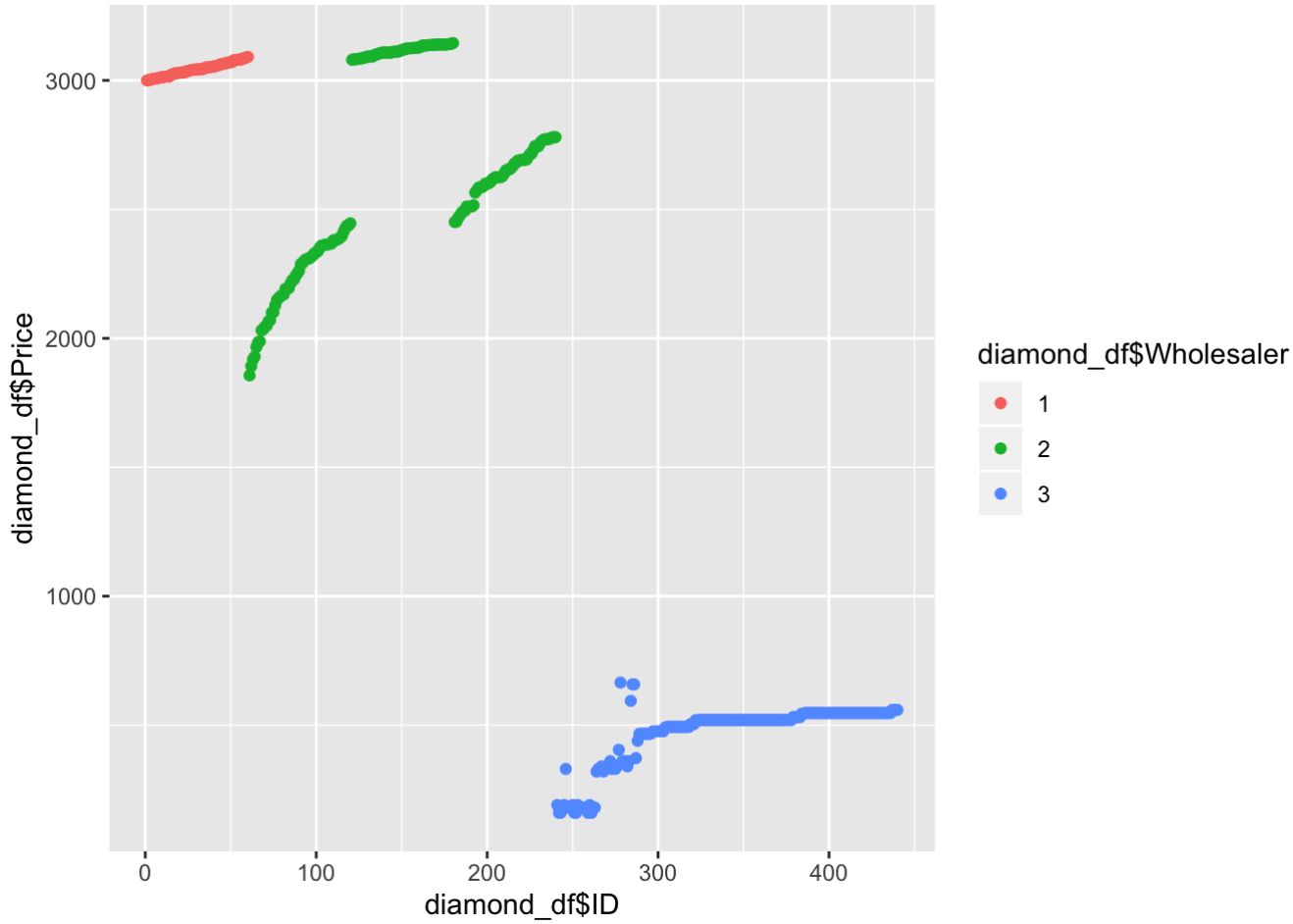
```
##
## Call:
## lm(formula = PSS_df$Sales.Progresso ~ PSS_df$Region + PSS_df$Low_Income +
##     PSS_df$High_Income + PSS_df$Price.Campbell + PSS_df$Price.PL +
##     PSS_df$Price.Progresso + PSS_df$winter, data = PSS_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4225   -833   -173    468  49117
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4293.95      59.91   71.67   <2e-16 ***
## PSS_df$RegionMidWest   -1186.59      22.19  -53.47   <2e-16 ***
## PSS_df$RegionSouth     -1857.59      19.41  -95.68   <2e-16 ***
## PSS_df$RegionWest      -1222.37      21.41  -57.11   <2e-16 ***
## PSS_df$Low_Income       -292.13      17.90  -16.32   <2e-16 ***
## PSS_df$High_Income       361.90      18.10   19.99   <2e-16 ***
## PSS_df$Price.Campbell    922.40      37.00   24.93   <2e-16 ***
## PSS_df$Price.PL          580.68      39.17   14.82   <2e-16 ***
## PSS_df$Price.Progresso -2456.17      23.39 -105.01   <2e-16 ***
## PSS_df$winterTRUE        817.97      15.15   53.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1664 on 59090 degrees of freedom
## Multiple R-squared:  0.394,  Adjusted R-squared:  0.3939
## F-statistic:  4269 on 9 and 59090 DF,  p-value: < 2.2e-16
```

The r-square of this linear regressio model is 0.394, which means 39.4% of the dependent variables (Sales.Progresso) is accounted for the independent variables. Also, since the p-value for each independent variables are small enough, we can keep all of them. Basically, this model indicates that the store located in the East region will decrease the Sales.Progresso by 1186.59 dollars than East region. Sotres in low_income region will decrease the Sales.Progresso by 292.13 dollars. One dollar increases in the price of Campbell will increase the Sales.Progresso by 922.40 dollars

3. Based on this model, we can consider to open some new stores in high income communities in east reagion. Also, we can increase the price of campbell and private label and decrease the price of progresso to increase the sales of progresso.

Problem 2: 1.

```
url2 = "https://raw.githubusercontent.com/jcbonilla/BusinessAnalytics/master/BAData/Diam
onds.csv"
diamond = read.csv(url2)
diamond_df = as.data.frame(diamond)
diamond_df$ID <- seq.int(nrow(diamond_df))
diamond_df$Wholesaler = factor(diamond_df$Wholesaler, levels = 1:3, labels = c("1","2",
"3"))
ggplot(diamond_df, aes(y=diamond_df$Price, x=diamond_df$ID, color=diamond_df$Wholesale
r)) +
  geom_point()
```




```
model = lm(diamond_df$Price ~ diamond_df$Carat + diamond_df$Colour + diamond_df$Clarity
+ diamond_df$Cut.
           + diamond_df$Polish + diamond_df$Symmetry + diamond_df$Certification, data =
diamond_df)
summary(model)
```

```
##
## Call:
## lm(formula = diamond_df$Price ~ diamond_df$Carat + diamond_df$Colour +
##     diamond_df$Clarity + diamond_df$Cut. + diamond_df$Polish +
##     diamond_df$Symmetry + diamond_df$Certification, data = diamond_df)
##
## 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 ***
## diamond_df$Carat            4202.98      51.46  81.677  < 2e-16 ***
## diamond_df$ColourE          -191.18      53.61  -3.566 0.000406 ***
## diamond_df$ColourF          -309.40      53.59  -5.773 1.55e-08 ***
## diamond_df$ColourG          -302.01      54.54  -5.537 5.53e-08 ***
## diamond_df$ColourH          -432.14      53.12  -8.136 5.04e-15 ***
## diamond_df$ColourI          -502.96      53.13  -9.467  < 2e-16 ***
## diamond_df$ColourJ          -637.71      54.69 -11.661  < 2e-16 ***
## diamond_df$ColourK          -987.33      61.03 -16.179  < 2e-16 ***
## diamond_df$ColourL         -1174.60      76.39 -15.377  < 2e-16 ***
## diamond_df$ClarityI2        -777.74      50.84 -15.299  < 2e-16 ***
## diamond_df$ClaritySI1        860.54      43.29  19.877  < 2e-16 ***
## diamond_df$ClaritySI2        731.99      35.63  20.543  < 2e-16 ***
## diamond_df$ClaritySI3        388.67      49.24   7.893 2.77e-14 ***
## diamond_df$ClarityVS1       1027.21      59.06  17.391  < 2e-16 ***
## diamond_df$ClarityVS2        917.77      53.33  17.210  < 2e-16 ***
## diamond_df$ClarityVVS1      1343.74     154.80   8.681  < 2e-16 ***
## diamond_df$ClarityVVS2       931.81     100.69   9.254  < 2e-16 ***
## diamond_df$Cut.G              56.33      41.91   1.344 0.179679
## diamond_df$Cut.I              95.73      40.95   2.338 0.019874 *
## diamond_df$Cut.V              83.85      41.59   2.016 0.044425 *
## diamond_df$Cut.X              57.05      35.73   1.597 0.111084
## diamond_df$PolishG           211.41     106.81   1.979 0.048460 *
## diamond_df$PolishI           460.81     156.95   2.936 0.003514 **
## diamond_df$Polishv           262.28     230.90   1.136 0.256672
## diamond_df$PolishV           226.82     110.71   2.049 0.041119 *
## diamond_df$PolishX           236.09     113.80   2.075 0.038648 *
## diamond_df$SymmetryG         108.22      57.43   1.885 0.060208 .
## diamond_df$SymmetryI             NA         NA      NA       NA
## diamond_df$SymmetryV         117.92      61.33   1.923 0.055225 .
## diamond_df$SymmetryX         111.20      68.36   1.627 0.104589
## diamond_df$CertificationDOW -499.67     223.64  -2.234 0.026008 *
## diamond_df$CertificationEGL -416.84      81.95  -5.087 5.58e-07 ***
## diamond_df$CertificationGIA  -64.40      80.72  -0.798 0.425450
## diamond_df$CertificationIGI   -8.05      95.43  -0.084 0.932812
## ---
## 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
```

```
intercept_ = as.numeric(summary(model)$coefficients[1, 1])
carat_coef = as.numeric(summary(model)$coefficients[2, 1])
cut_very_good_coef = as.numeric(summary(model)$coefficients[21, 1])
color_J_coef = as.numeric(summary(model)$coefficients[8, 1])
ClaritySI2_coef = as.numeric(summary(model)$coefficients[13, 1])
PolishG_coef = as.numeric(summary(model)$coefficients[23, 1])
SymmetryV_coef = as.numeric(summary(model)$coefficients[29, 1])##ignore this due to p-va
lue > 0.05
CertificationGIA_coef = as.numeric(summary(model)$coefficients[33, 1])##ignore this due
 to p-value > 0.05
quota = intercept_+0.9*carat_coef+cut_very_good_coef+color_J_coef+ClaritySI2_coef+Polish
G_coef
quota
```

```
## [1] 2741.848
```

a)2741.848 < 3100, so the diamond is overpriced.

b)One unit increase in carat, the price increases by 4202.98 dollars. Compared to color D, color E decreases the price by 191 dollars. Compared to ClarityFL, diamond with ClarityI2 decreased the price by 777.74 dollars.

c)The r-squre of this model is 0.9739 which means 97.39% of the dependent variables (Price) is accounted for the independent variables. So this is a good model. However, since some independent variables has high p-value such as Cut.G, Cut.X, Polishv and so on, we should ignore these independent variables in our model.

2.

a.

```
diamond_new = subset(diamond_df, diamond_df$Wholesaler!=3)
model1 = lm(diamond_new$Price ~ diamond_new$Carat + diamond_new$Colour + diamond_new$Cla
rity + diamond_new$Cut.
           + diamond_new$Polish + diamond_new$Symmetry + diamond_new$Certification, data
= diamond_new)
summary(model1)
```

```
##
## Call:
## lm(formula = diamond_new$Price ~ diamond_new$Carat + diamond_new$Colour +
##     diamond_new$Clarity + diamond_new$Cut. + diamond_new$Polish +
##     diamond_new$Symmetry + diamond_new$Certification, data = diamond_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -432.53 -112.20  -10.69  110.74  551.95
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   526.711    229.387   2.296 0.022658 *
## diamond_new$Carat            2224.787    181.612  12.250  < 2e-16 ***
## diamond_new$ColourE          -131.893     61.774  -2.135 0.033916 *
## diamond_new$ColourF          -307.649     65.588  -4.691 4.92e-06 ***
## diamond_new$ColourG          -239.704     62.446  -3.839 0.000164 ***
## diamond_new$ColourH          -323.468     63.175  -5.120 6.91e-07 ***
## diamond_new$ColourI          -390.309     63.052  -6.190 3.13e-09 ***
## diamond_new$ColourJ          -505.430     63.955  -7.903 1.54e-13 ***
## diamond_new$ColourK          -770.878     69.417 -11.105  < 2e-16 ***
## diamond_new$ColourL          -952.718     81.144 -11.741  < 2e-16 ***
## diamond_new$ClarityI2        -572.334     51.444 -11.125  < 2e-16 ***
## diamond_new$ClaritySI1        726.365     56.836  12.780  < 2e-16 ***
## diamond_new$ClaritySI2        603.265     41.579  14.509  < 2e-16 ***
## diamond_new$ClaritySI3        309.407     48.226   6.416 9.22e-10 ***
## diamond_new$ClarityVS1        832.316     91.400   9.106  < 2e-16 ***
## diamond_new$ClarityVS2        820.050     89.870   9.125  < 2e-16 ***
## diamond_new$Cut.G              23.092     44.774   0.516 0.606584
## diamond_new$Cut.I              87.321     49.551   1.762 0.079489 .
## diamond_new$Cut.V              36.133     57.122   0.633 0.527708
## diamond_new$Cut.X             121.231     38.078   3.184 0.001676 **
## diamond_new$PolishG            90.589    102.702   0.882 0.378758
## diamond_new$PolishI           181.761    162.655   1.117 0.265079
## diamond_new$Polishv           144.093    223.363   0.645 0.519565
## diamond_new$PolishV           119.910    109.465   1.095 0.274597
## diamond_new$PolishX           122.537    119.582   1.025 0.306685
## diamond_new$SymmetryG         105.981     55.588   1.907 0.057954 .
## diamond_new$SymmetryI              NA         NA      NA       NA
## diamond_new$SymmetryV          99.699     61.592   1.619 0.107022
## diamond_new$SymmetryX          68.277     78.799   0.866 0.387223
## diamond_new$CertificationDOW -315.747    215.317  -1.466 0.144035
## diamond_new$CertificationEGL -263.638     81.777  -3.224 0.001468 **
## diamond_new$CertificationGIA    6.291     79.285   0.079 0.936836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 187 on 209 degrees of freedom
## Multiple R-squared:  0.7735, Adjusted R-squared:  0.741
## F-statistic: 23.79 on 30 and 209 DF,  p-value: < 2.2e-16
```

After dropped wholesaler #3, the r-square of this new model decreases to 0.7735 and the number independent variables with p-values greater than 0.05 incrases, which mean the new model is not fit as good as the previous one. I think the reason is that we have less training samples in the new model so that the new model is not fit that good.

b.

```
intercept_ = as.numeric(summary(model1)$coefficients[1, 1])
carat_coef = as.numeric(summary(model1)$coefficients[2, 1])
cut_very_good_coef = as.numeric(summary(model1)$coefficients[21, 1])##ignore this due to
p-value > 0.05
color_J_coef = as.numeric(summary(model1)$coefficients[8, 1])
ClaritySI2_coef = as.numeric(summary(model1)$coefficients[13, 1])
PolishG_coef = as.numeric(summary(model1)$coefficients[23, 1])##ignore this due to p-val
ue > 0.05
SymmetryV_coef = as.numeric(summary(model1)$coefficients[29, 1])##ignore this due to p-v
alue > 0.05
CertificationGIA_coef = as.numeric(summary(model1)$coefficients[31, 1])##ignore this due
to p-value > 0.05
quota = intercept_+0.9*carat_coef+color_J_coef+ClaritySI2_coef
quota
```

```
## [1] 2626.854
```

```
predict1 = predict(model, diamond_df, se.fit = TRUE)
```

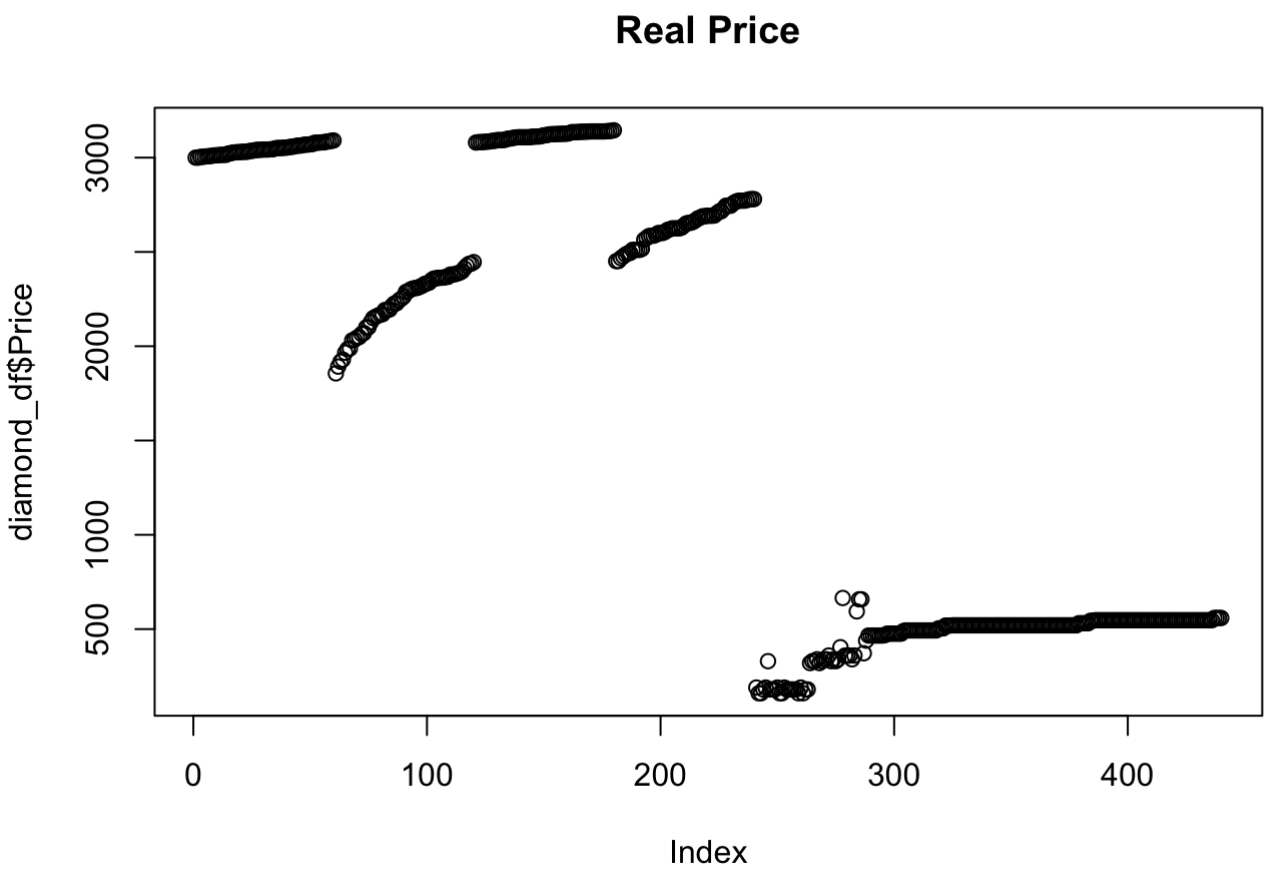
```
## Warning in predict.lm(model, diamond_df, se.fit = TRUE): prediction from a
## rank-deficient fit may be misleading
```

```
predict2 = predict(model1, diamond_df, se.fit = TRUE)
```

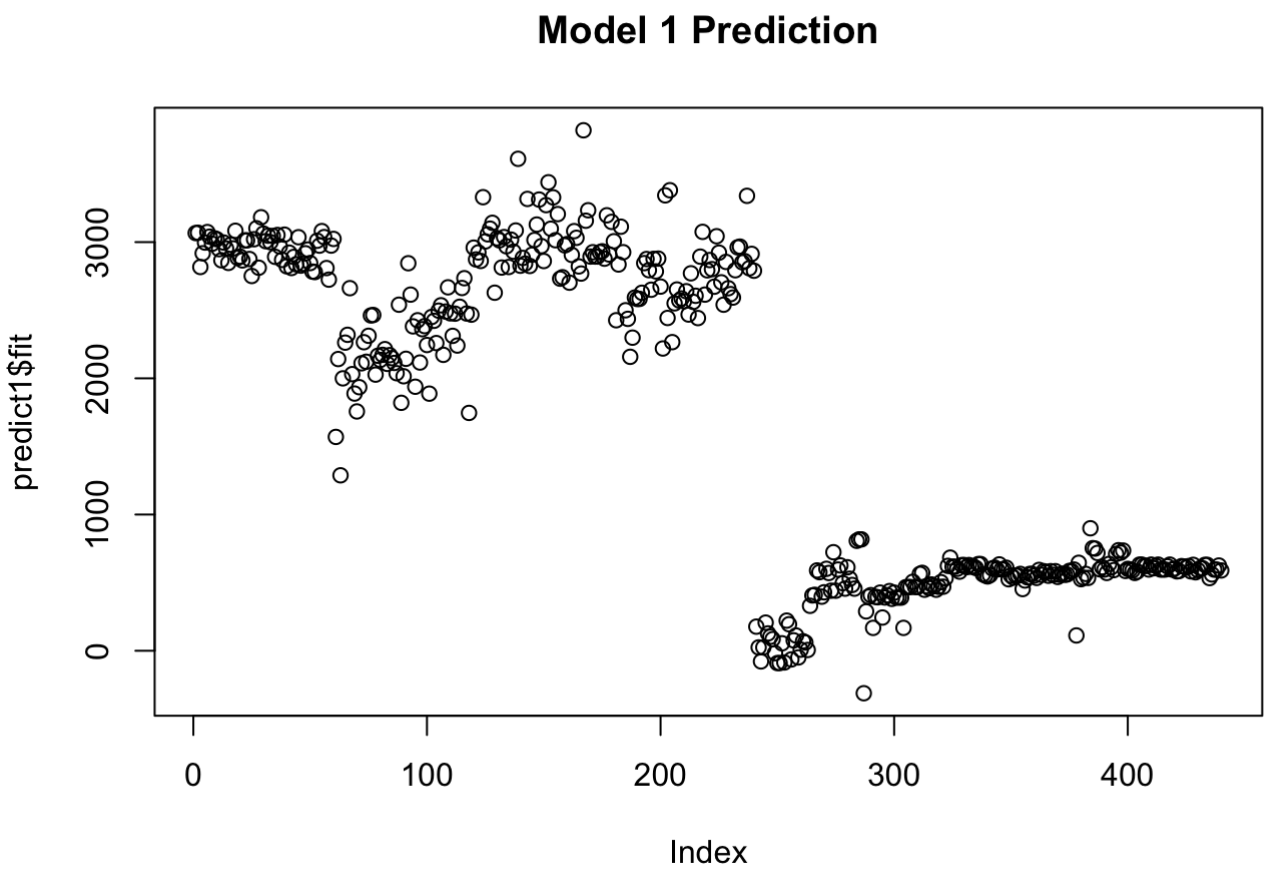
```
## Warning: 'newdata' had 440 rows but variables found have 240 rows
```

```
## Warning in predict.lm(model1, diamond_df, se.fit = TRUE): prediction from a
## rank-deficient fit may be misleading
```

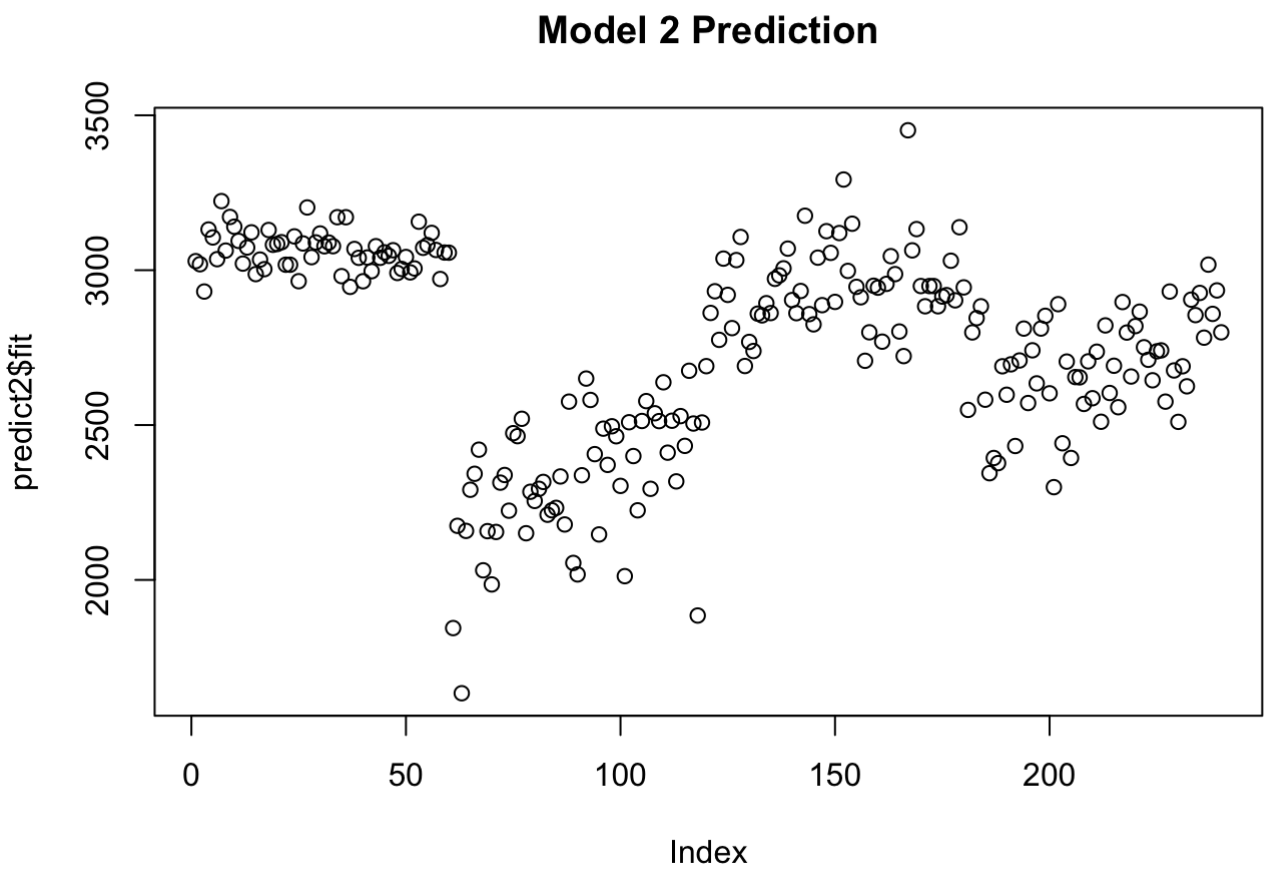
```
plot(diamond_df$Price, main="Real Price")
```




```
plot(predict1$fit, main="Model 1 Prediction")
```




```
plot(predict2$fit, main="Model 2 Prediction")
```



I think model 1 is better and more correct than model 2 because model 1 has higher r-square and the number independent variables with p-values greater than 0.05 less compared to model 2. Also, based on the graphs above, it's not hard to see that the model 1 has better fit shape than model 2.