PROJECT TECHNICAL REPORT

“Diamonds Price Prediction”

Data Science Project using SAS (DSPS)

By

<< Niloofar Zarifi, Student No:0119-02015 >>

Metro College of Technology

2019

TABLE OF CONTENTS

PROJECT OVERVIEW………………………………………………………………………………………………………….. i

LIST OF TABLES ..………………………………………………………………………………………………………………….ii

LIST OF FIGURES ………………………………………………………………………………………………………………….iii

INTRODUCTION …………………………………………………………………………………………………………………..iv

OBJECTIVES ………………………………………………………………………………………………………………………… 1

DATA SCIENCE PROCESS DETAILS …………………………………………………………………………………….. <1>

RESULTS AND DISCUSSIONS ……………………………………………………………………………………………….. <2>

CONCLUSIONS AND RECOMMENDATIONS …………………………………………………………………………. <12>

APPENDIX A. SAS CODE ……………………………………………………………………………………………………… <13>

PROJECT OVERVIEW

A diamond distributor has recently decided to exit the market and has put up a set of 3,000 diamonds up for auction. Seeing this as a great opportunity to expand its inventory, a jewelry company has shown interest in making a bid. To decide how much to bid, the company’s analytics team used a large database of diamond prices to build a model to predict the price of a diamond based on its attributes. As a data analyst, you are assigned to do data analysis and predictive modeling to make a recommendation for how much the company should bid.

LIST OF TABLES

Table 1. The table below describes data attributes………………………………………………………………………………. 2

Table 2. The table below shows pearson correlations …………………………………………………………………………….5

Table 3. Coefficients, standard error, t and P values of Proc reg……………………………………………………………. 9

Table 4. Coefficients, standard error, t and P values of Proc GLM, model B…………………………………………..10

Table 5: The average of predicted price and actual price from existing data set …………………………………. 11

Table 6: The predicted price with model A for specific diamonds …………………………………………………………12

LIST OF FIGURES

Figure 1. Dropbox plot for carat\_color…………………………………………………………………………………………..3

Figure 2. Dropbox plot for Carat\_clarity…………………………………………………………………………………………3

Figure 3. Bar charts of (cut\_clarity), (color\_cut) and (color\_clarity) ……………………………………………….4

Figure 4. Distribution of price, carat, depth, table, x, y and z………………………………………………………….6

Figure 5. Data transformation for price and carat…………………………………………………………………………..7

Figure 6. Fit diagnostic for logprice in model A……………………………………………………………………………….8

INTRODUCTION

Jewelry is a popular piece of item that can either be used for personal use or for giving away as a gift. Jewelry business is highly competitive. The jewelry industry seems poised for a glittering future. Annual global sales of €148 billion are expected to grow at a healthy clip of 5 to 6 percent each year, totaling €250 billion by 2020. Consumer appetite for jewelry, which was dampened by the global recession, now appears more voracious than ever.

But the industry is as dynamic as it is fast growing. Consequential changes are under way, both in consumer behavior as well as in the industry itself. Jewelry players can’t simply do business as usual and expect to thrive; they must be alert and responsive to important trends and developments or else risk being left behind by more agile competitors.

To chart the most likely course of the jewelry sector, data scientists analyze confidential and public data. They study companies’ annual reports and look inside global fine-jewelry and fashion-jewelry industry to keep business profitable. Purchasing the diamond with good deal is very significant for a jewelry company. Based on existing data, data scientist can recommend a winner bid number to buy >1000 diamonds.

OBJECTIVES

This project is meant to predict price of 3000 diamonds so that jewelry company management can make informed decision about how much to bid a large set of diamonds in auction.

DATA SCIENCE PROCESS DETAILS

The project is a part of data science problem domain to quantitatively estimate the unknown price of diamonds by leveraging predictive analytics and extensively utilizing team data science processes to implement supervised algorithm appropriate for Machine Learning to predict diamonds price through data management in SAS environment. In this project two models derived from linear regression will be presented to recommend bid price. The below instruction should be followed to facilitate our analysis

1. **Understanding the data**
2. **Exploratory Data Analysis**
3. **Building the model**
4. **Validation of the model**
5. **Price prediction**

RESULTS AND DISCUSSIONS

1. **Understanding the data**

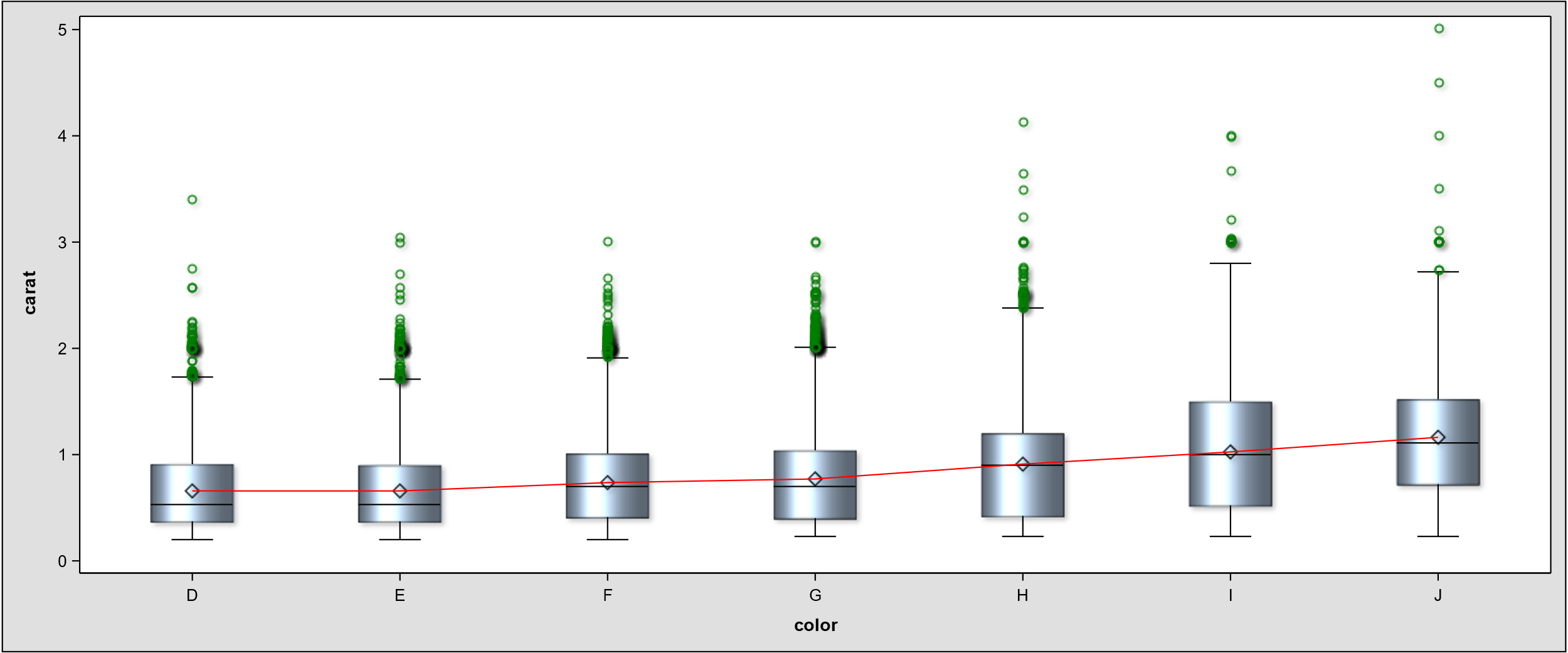
There are two datasets.“​diamonds.csv”​ contains the data used to build the regression model. “​new-diamonds.csv” contains the dataset for the diamonds the jewelry company would like to purchase. Both datasets contain carat, cut, and clarity data for each diamond. Only the “​diamonds.csv” dataset has prices. Use the “​new-diamonds.csv”​ to build the predictive model. The Table 1. Describe the attributes of “​diamonds.csv”​. please note that the diamonds on the market for sell do not have the dimension data including table, depth, x, y and z

Table 1. The table below describes data attributes.

|  |  |
| --- | --- |
| Data element | Description |
| Price | price in US dollars ($326--$18,823) |
| Carat | weight of the diamond (0.2--5.01) |
| Cut | quality of the cut (Fair, Good, Very Good, Premium, Ideal) |
| Color | diamond colour, from D (best) to J (worst) |
| Clarity | measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best)) |
| x | length in mm (0--10.74) |
| y | width in mm (0--58.9) |
| z | depth in mm (0--31.8) |
| Depth | total depth percentage = z / mean(x, y) = 2 \* z / (x + y) (43--79) |
| Table | width of top of diamond relative to widest point (43--95) |

1. **Exploratory Data Analysis**

The boxplots below describe the quintiles and medians of Figure 1. carat\_color and Figure 2. Carat\_clarity. It seems like the poor color and clarity diamonds are generally heavier. Please note that the result from boxplot cannot determine the correlations of color, clarity and carat and it is not conclusive.



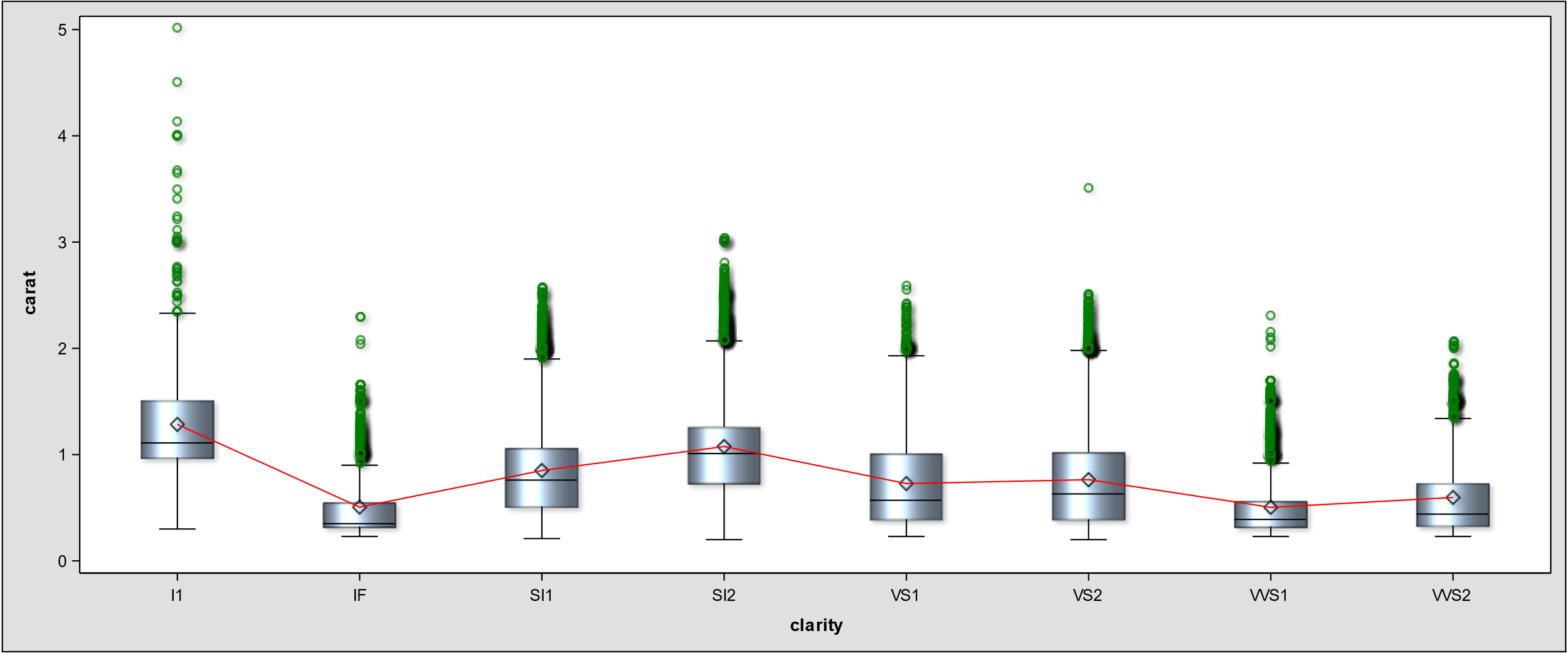
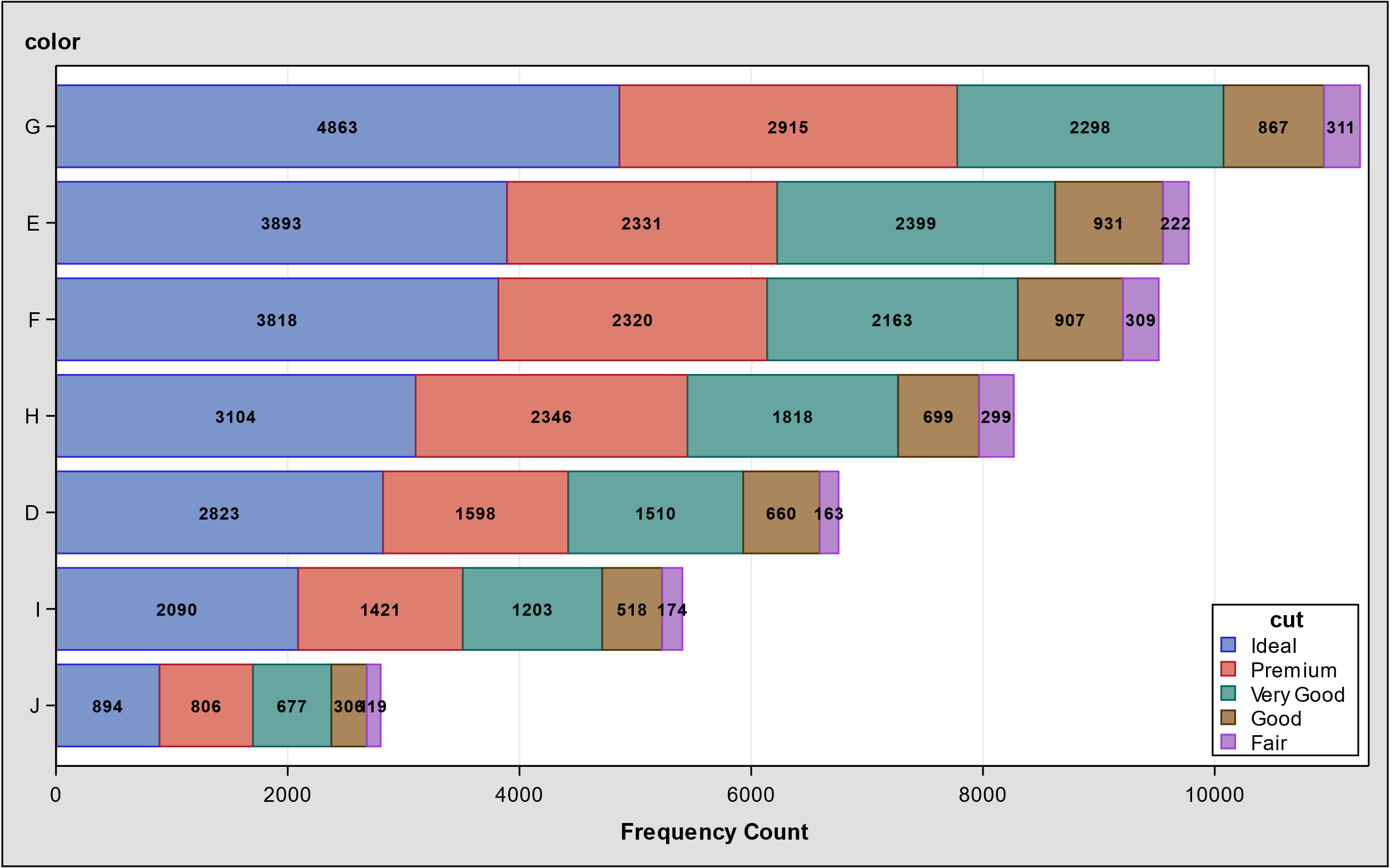
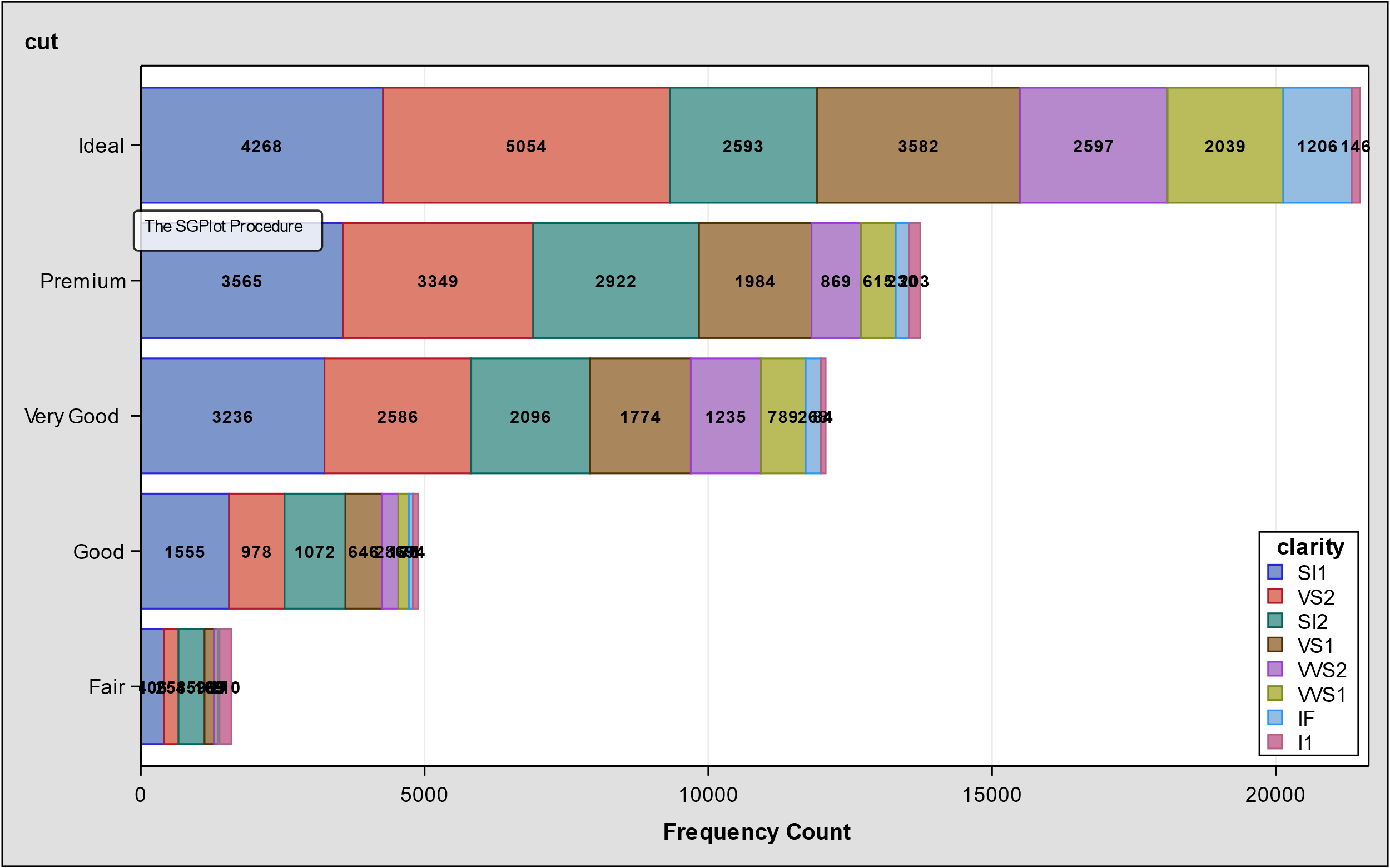


Figure 1. Dropbox plot for carat\_color and Figure 2. Dropbox plot for Carat\_clarity.

Figure 3. simply indicates that good number of diamonds have ideal cut and medium quality color which will likely cause higher bid.



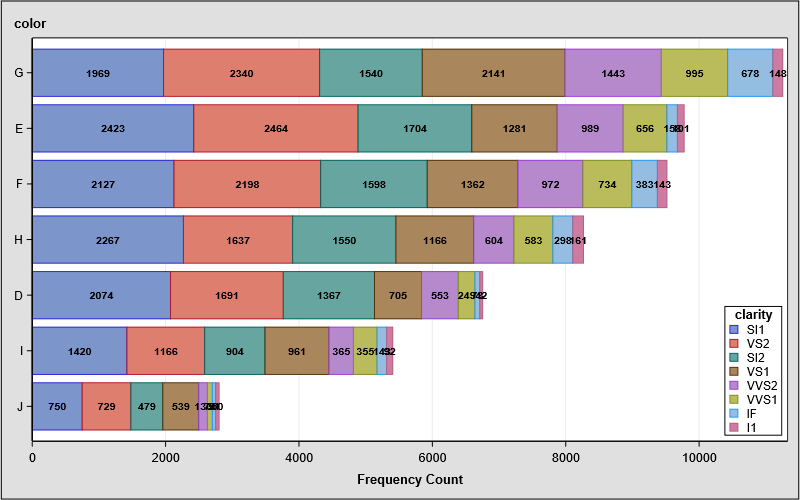
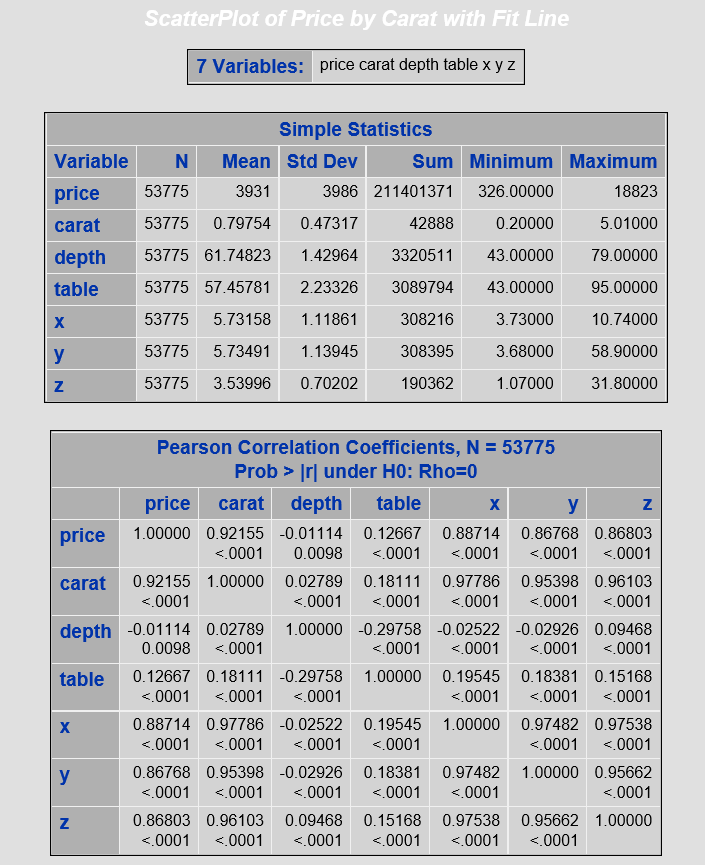
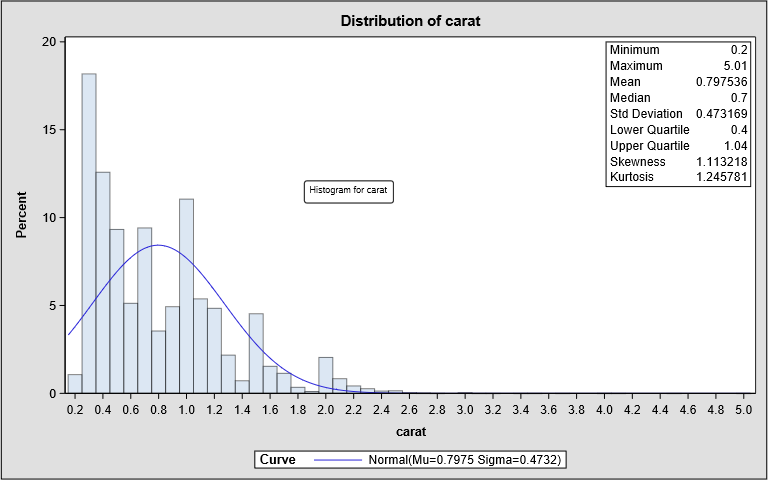
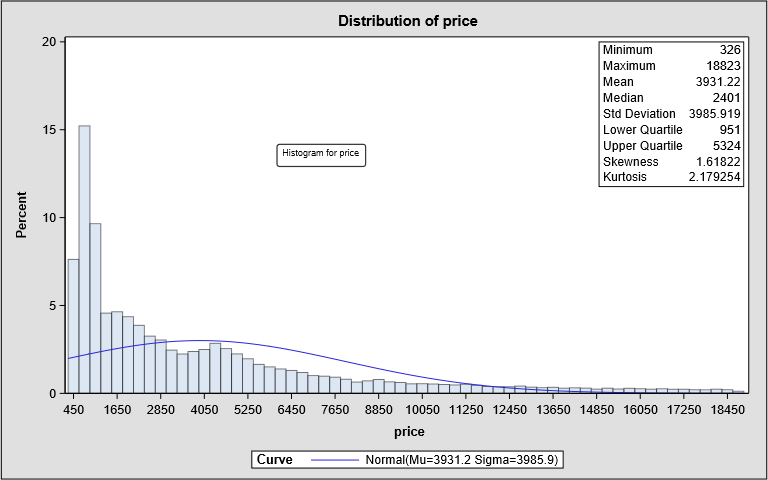


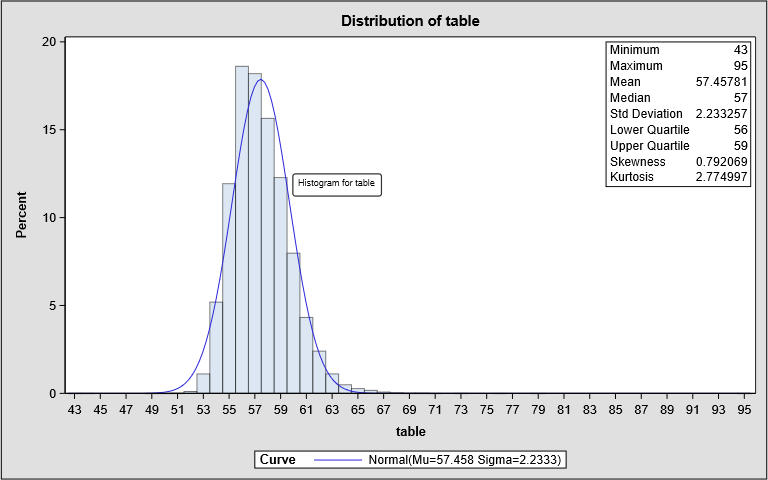
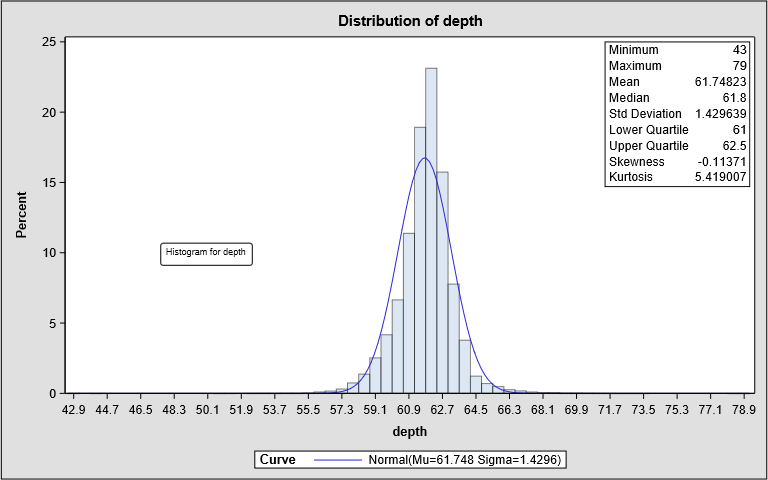
Figure 3. Bar charts of (cut\_clarity), (color\_cut) and (color\_clarity)

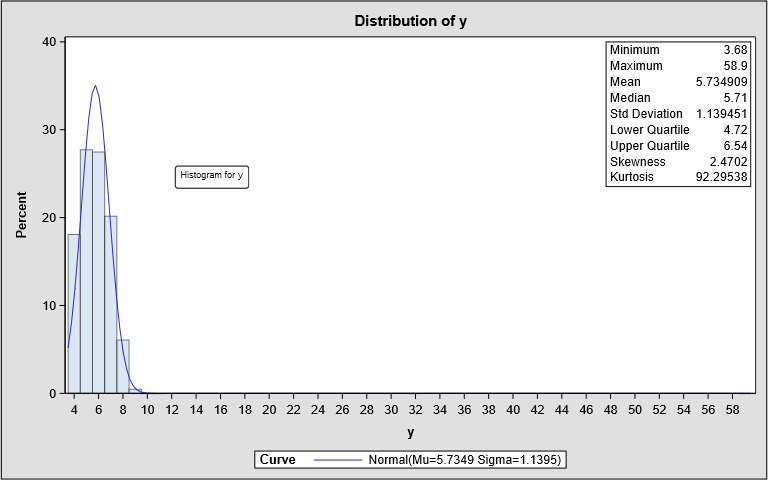
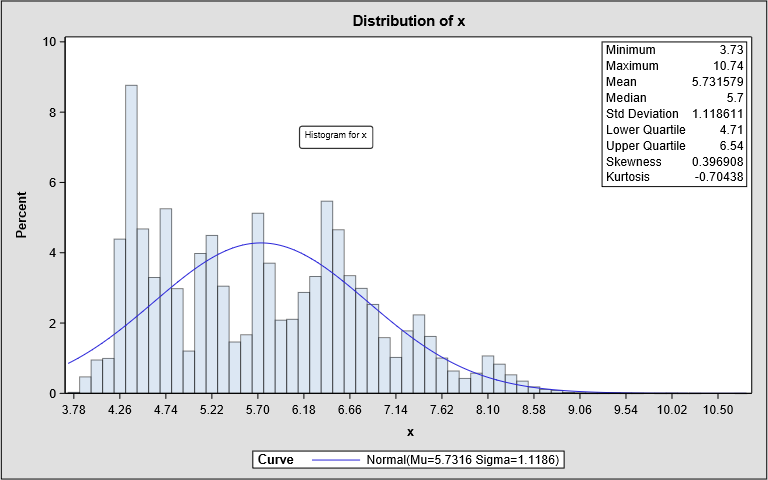
Although x, y and z are highly correlated with diamond’s price, we are not going to apply these parameters for regression model. These information does not exist for unpurchased diamonds.

Table 2. The table below shows pearson correlations









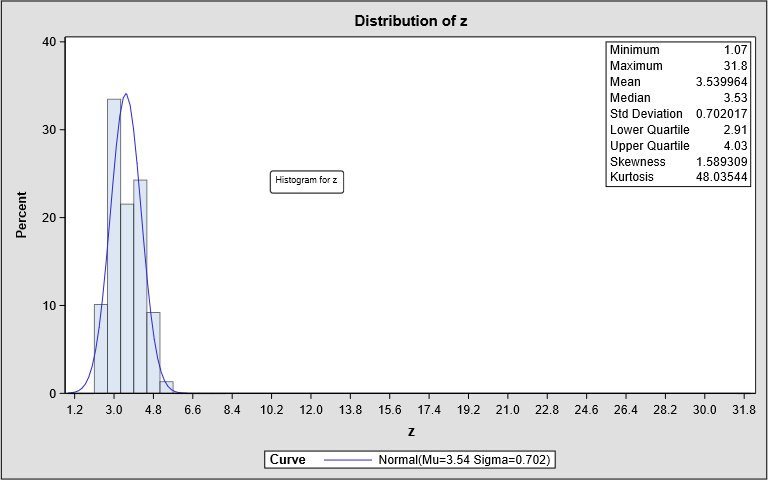


Figure 4. Distribution of price, carat, depth, table, x, y and z.

Price, carat, y and z are right skewed and do not have normal distribution which negatively affect the linear regression. We are not worried about y and z but we apply log(x)+1 function to price and carat to reduce the outliers and make it closer to normal distribution. Figure 5. Show that how outliers get disappear and reduced for price and carat respectively and logprice is bimodal now. The q-q plots were also improved.

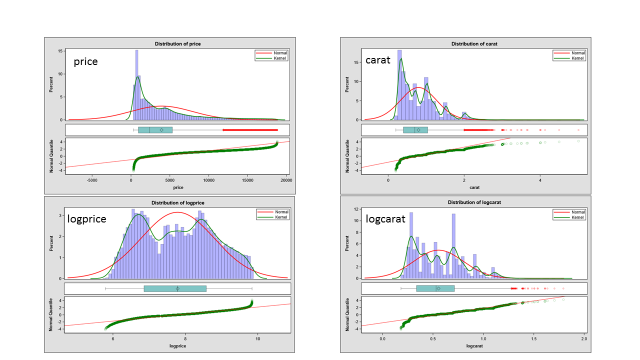


Figure 5. Data transformation for price and carat.

1. **& 4: Building and validation of the models.**

We used proc reg (Model A) and proc GLM (Model B) to build the model. Figure 6. shows fit diagnostic for model A with R\_squred of 0.95 and MSE of 0.04. The accuracy of the Model B is R\_squared of 0.95 and MSE of 0.20.

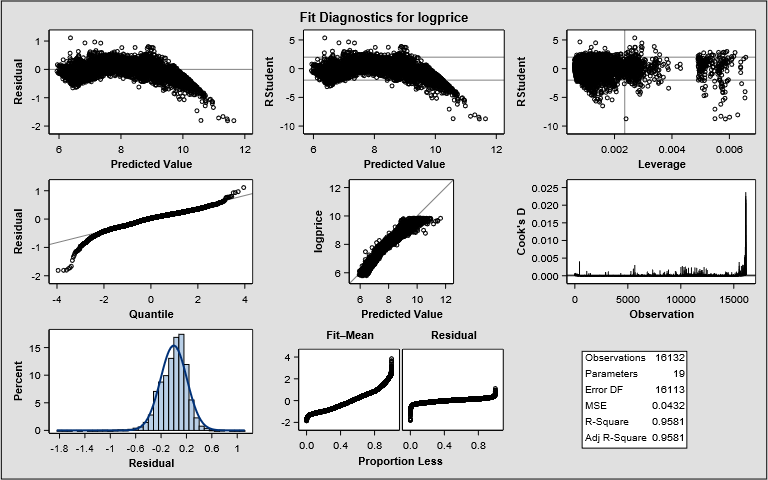
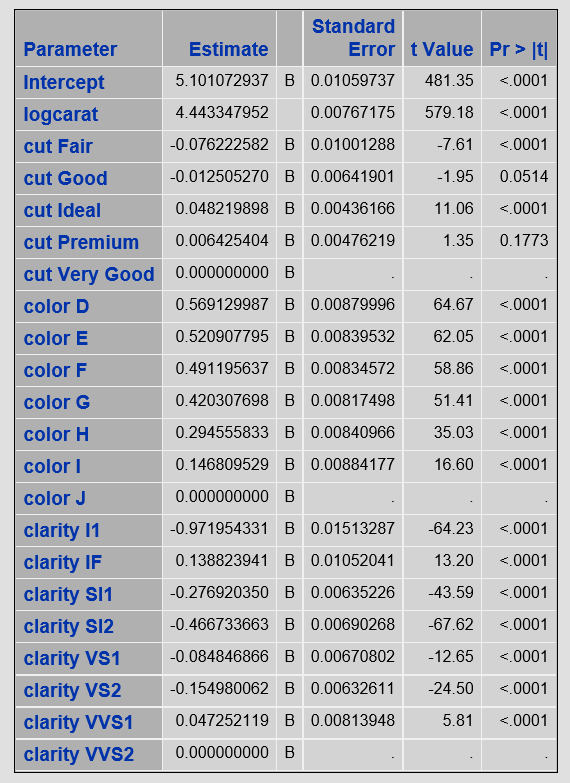


Figure 6. Fit diagnostic for logprice in model A.

Table 3. Coefficients, standard error, t and P values of Proc reg.



Table 4. Coefficients, standard error, t and P values of Proc GLM, model B.



According to model A and B group by diamond cut and categorize carat\_size to predict the average of price. As you can see Model A has been evaluated the heavy diamond 200% higher while the result for model B is reasonable.

Table 5: The average of predicted price and actual price from existing data set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cut | Carat\_size(X) | predicted\_Proc reg $ (model A) | predicted\_GLM  $(model B) | Dataset$ |
| Fair | X<1 | 1601 |  | 2202 |
| Fair | 1<X<2 | 5558 |  | 5274 |
| Fair | 2<X<3 | 32054 | 10567 | 11676 |
| good | X<1 | 1439 |  | 1792 |
| good | 1<X<2 | 6406 |  | 6345 |
| good | 2<X<3 | 36481 | 14622 | 14575 |
| ideal | X<1 | 1202 |  | 1546 |
| ideal | 1<X<2 | 6617 |  | 7898 |
| ideal | 2<X<3 | 34948 | 15489 | 15662 |
| premium | X<1 | 1237 |  | 1597 |
| premium | 1<X<2 | 6646 |  | 7233 |
| premium | 2<cut<3 | 35120 | 14818 | 14907 |
| verygood | cut<1 | 1314 |  | 1732 |
| verygood | 1<cut<2 | 6377 |  | 7270 |
| verygood | 2<cut<3 | 34422 | 15546 | 15120 |

What is the predicted price for diamonds (carat=1.5, cut =Very Good, color = VS2) ?

* 56 diamonds from the dataset has Carat=1.5, Cut=Very Good, Color=VS2 parameters with the average price of $11605.
* 96 diamonds from the dataset has 1.5<Carat<=1.55, Cut=Very Good, Color=VS2 parameters with the average price of $ 11604.

It sounds like the predicted price for model A is reasonable.

Table 6: The predicted price with model A for specific diamonds

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| carat | cut | color | clarity | predicted\_price |
| 1.5 | Very Good | D | VS2 | 10805.4 |
| 1.5 | Very Good | J | VS2 | 10805.4 |
| 1.51 | Very Good | E | VS2 | 10112.04 |
| 1.51 | Very Good | G | VS2 | 8304.48 |
| 1.52 | Very Good | F | VS2 | 9523.83 |
| 1.52 | Very Good | G | VS2 | 8452.59 |

CONCLUSION AND RECOMMENDATIONS

Our model A recommends $12,990,108 for 3000 diamonds and model B is $12520860. The difference of two model is around $470,000 which is significant. I offer further investigation and other powerful algorithms such as random forest.

APPENDIX A: SAS CODE

EDA Codes:

DSPS\_DIAMONDS\_HISTOGRAM\_01

ODS graphics on / reset width=**8**in height=**5**in;

ODS noproctitle;

ODS select histogram ParameterEstimates GoodnessOfFit FitQuantiles;

Title 'Distribution Plot for Diamonds';

**proc** **univariate** data=mylib.clean\_53775;

var price carat depth table x y z;

/\*you can change normal with kernel which is density distribution

with normal solid line we can observe med and some how mean were shifted

which says we do not have normal distribution\*/

histogram price carat depth table x y z / normal(color=salmon);

inset Min Max mean median std Q1 Q3 skewness kurtosis / pos=ne;

**Run**;

Title;

**proc** **univariate** data=mylib.clean\_53775 noprint;

var price carat depth table x y z;

qqplot price / normal(mu=**3933.1** sigma=**3988.1**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (Price)';

qqplot carat / normal(mu=**0.7978** sigma=**0.4734**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (Carat)';

qqplot depth / normal(mu=**61.748** sigma=**1.4299**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (depth)';

qqplot table / normal(mu=**57.458** sigma=**2.2337**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (table)';

qqplot x / normal(mu=**5.7312** sigma=**1.1207**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (x)';

qqplot y / normal(mu=**5.7347** sigma=**1.1412**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (y)';

qqplot z / normal(mu=**3.5387** sigma=**0.705**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (z)';

**run**;

ODS graphics off;

DSPS\_DIAMONDS\_HISTOGRAM\_LOG

ODS graphics on / reset width=**8**in height=**5**in;

ODS noproctitle;

\*ODS select histogram ParameterEstimates GoodnessOfFit FitQuantiles;

ODS select histogram;

Title 'Distribution Plot for Diamonds';

/\*need to generate two cols logprice and logcarat first\*/

**proc** **univariate** data=mylib.clean\_53775\_log;

histogram logprice logcarat / normal;

inset Min Max mean median std Q1 Q3 skewness kurtosis / pos=ne;

**Run**;

Title;

**proc** **univariate** data=mylib.clean\_53775\_log noprint;

qqplot logprice / normal(mu=**7.7878** sigma=**1.0139**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (LogPrice)';

qqplot logcarat / normal(mu=**0.5553** sigma=**0.2446**)

odstitle = 'Normal Quartile-Quartile Plot for Diamonds (LogCarat)';

**run**;

ODS graphics off;

DSPS\_DIAMONDS\_BOXPLOT\_4C

ODS graphics on /reset width=**12**in height=**5**in;

title 'Diamonds Price by Cut';

**proc** **sgplot** data=mylib.clean\_53775;

vbox price / category=cut

dataskin=sheen

outlierattrs=(color=green)

meanattrs=(color=black)

medianattrs=(color=black)

connect=mean connectattrs=(color=red);

**run**;

title 'Diamonds Price by Color';

**proc** **sgplot** data=mylib.clean\_53775;

vbox price / category=color

dataskin=sheen

outlierattrs=(color=green)

meanattrs=(color=black)

medianattrs=(color=black)

connect=mean connectattrs=(color=red);

**run**;

title 'Diamonds Price by Clarity';

**proc** **sgplot** data=mylib.clean\_53775;

vbox price / category=clarity

dataskin=sheen

outlierattrs=(color=green)

meanattrs=(color=black)

medianattrs=(color=black)

connect=mean connectattrs=(color=red);

**run**;

title 'Diamonds Price by Carat';

**proc** **sgplot** data=mylib.clean\_53775;

vbox price / category=carat

dataskin=sheen

nooutliers

connect=mean connectattrs=(color=red);

xaxis display=(novalues);

**run**;

DSPS\_DIAMONDS\_GTL\_Histogram\_Boxplot

**proc** **template**;

define statgraph threepanel;

dynamic \_X \_QUANTILE \_Title \_mu \_sigma;

begingraph;

entrytitle halign=center \_Title;

layout lattice / rowdatarange=data columndatarange=union

columns=**1** rowgutter=**5** rowweights=(**0.6** **0.10** **0.3**);

layout overlay;

histogram \_X / name='histogram'

binaxis=false

datatransparency=**0.7**

fillattrs=(color=blue);

densityplot \_X / name='Normal' normal()

lineattrs=(color=red thickness=**2**);

densityplot \_X / name='Kernel' kernel()

lineattrs=(color=green thickness=**2**);

discretelegend 'Normal' 'Kernel' / border=true

halign=right

valign=top location=inside across=**1**;

endlayout;

layout overlay;

boxplot y=\_X / boxwidth=**0.8**

orient=horizontal

fillattrs=(color=darkcyan)

outlineattrs=(color=black)

medianattrs=(color=black)

meanattrs=(color=black)

outlierattrs=(color=red)

datatransparency=**0.5**

;

endlayout;

layout overlay;

scatterplot x=\_X y=\_QUANTILE / markerattrs=(color=green size=**10**) datatransparency=**0.7**;

lineparm x=\_mu y=**0.0** slope=eval(**1.**/\_sigma) / lineattrs=(color=red) extend=true clip=true;

endlayout;

columnaxes;

columnaxis;

endcolumnaxes;

endlayout;

endgraph;

end;

**run**;

**%macro** ThreePanel(DSName, Var);

%local mu sigma;

/\* 1. sort copy of data \*/

proc sort data=&DSName out=\_MyData(keep=&Var);

by &Var;

run;

/\* 2. Use PROC UNIVARIATE to create Q-Q plot

and parameter estimates \*/

ods exclude all;

proc univariate data=\_MyData;

var &Var;

histogram &Var / normal; /\* create ParameterEstimates table \*/

qqplot &Var / normal;

ods output ParameterEstimates=\_PE QQPlot=\_QQ(keep=Quantile Data rename=(Data=&Var));

run;

ods exclude none;

/\* 3. Merge quantiles with data \*/

data \_MyData;

merge \_MyData \_QQ;

label Quantile = "Normal Quantile";

run;

/\* 4. Get parameter estimates into macro vars \*/

data \_null\_;

set \_PE;

if Symbol="Mu" then call symputx("mu", Estimate);

if Symbol="Sigma" then call symputx("sigma", Estimate);

run;

proc sgrender data=\_MyData template=threepanel;

dynamic \_X="&Var" \_QUANTILE="Quantile" \_mu="&mu" \_sigma="&sigma"

\_title="Distribution of &Var";

run;

**%mend**;

Title;

ODS graphics on / reset width=**8**in height=**5**in;

%***ThreePanel***(mylib.diamonds\_FE, price);

%***ThreePanel***(mylib.diamonds\_FE, logprice);

%\*ThreePanel(MCT.diamonds\_FE, carat);

%\*ThreePanel(MCT.diamonds\_FE, logcarat);

ODS graphics off;

DSPS\_DIAMONDS\_Feature\_Engineer\_02

**Data** mylib.Diamonds\_FE2 (drop=i j k);

Set mylib.clean\_53775;

logprice = log(**1**+price);

logcarat = log(**1**+carat);

Array cut\_grade[**5**] $**9** \_temporary\_ ('Fair' 'Good' 'Very Good' 'Premium' 'Ideal');

Array color\_grade[**7**] $**1** \_temporary\_ ('J' 'I' 'H' 'G' 'F' 'E' 'D');

Array clarity\_grade[**8**] $**4** \_temporary\_ ('I1' 'SI2' 'SI1' 'VS2' 'VS1' 'VVS2' 'VVS1' 'IF');

Array dummy\_cut[\*] cut1 - cut5;

Array dummy\_color[\*] color1 - color7;

Array dummy\_clarity[\*] clarity1 - clarity8;

Do i = **1** to dim(dummy\_cut);

Do j = **1** to dim(dummy\_color);

Do k = **1** to dim(dummy\_clarity);

dummy\_cut(i) = **0**;

dummy\_color(j) = **0**;

dummy\_clarity(k) = **0**;

End;

End;

End;

Do i = **1** to dim(cut\_grade);

If cut\_grade[i] = Strip(cut) Then dummy\_cut[i] = **1**;

End;

Do i = **1** to dim(color\_grade);

If color\_grade[i] = Strip(color) Then dummy\_color[i] = **1**;

End;

Do I = **1** to dim(clarity\_grade);

If clarity\_grade[i] = Strip(clarity) Then dummy\_clarity[i] = **1**;

End;

Select (cut);

when ('Fair') cut\_ord = **1**; /\* Lowest level of fire and brilliance \*/

when ('Good') cut\_ord = **2**;

when ('Very Good') cut\_ord = **3**;

when ('Premium') cut\_ord = **4**;

when ('Ideal') cut\_ord = **5**; /\* Highest level of fire and brilliance \*/

otherwise

;

End;

Select (color);

when ('J') color\_ord = **1**;

when ('I') color\_ord = **2**;

when ('H') color\_ord = **3**;

when ('G') color\_ord = **4**; /\* G-J = Nearly Colorless \*/

when ('F') color\_ord = **5**;

when ('E') color\_ord = **6**;

when ('D') color\_ord = **7**; /\* D-F = Colorless is highest color grade \*/

otherwise

;

End;

Select (clarity);

when ('I1') clarity\_ord = **1**; /\* Inclusions 1 is the worst \*/

when ('SI2') clarity\_ord = **2**; /\* Small Inclusions 1 \*/

when ('SI1') clarity\_ord = **3**; /\* Small Inclusions 2 \*/

when ('VS2') clarity\_ord = **4**; /\* Very Small Inclusions 1 \*/

when ('VS1') clarity\_ord = **5**; /\* Very Small Inclusions 2 \*/

when ('VVS2') clarity\_ord = **6**; /\* Very Very Small Inclusions 1 \*/

when ('VVS1') clarity\_ord = **7**; /\* Very Very Small Inclusions 2 \*/

when ('IF') clarity\_ord = **8**; /\* Internally Flawless is the best \*/

otherwise

;

End;

**Run**;

DSPS\_DIAMONDS\_ScatterPlot\_04

/\* Add a Regression Fit Line \*/

ODS graphics on;

**Proc** **sgplot** data=mylib.clean\_53775 noborder noautolegend;

reg x=carat y=price / lineattrs=(color=red thickness=**3** pattern=**2**)

markerattrs=(color=dodgerblue size=**5**)

cli

cliattrs=(clilineattrs=(color=green thickness=**3** pattern=**20**))

;

title color=white "ScatterPlot of Price by Carat with Fit Line";

footnote color=white "Remark: data has carat <= 3";

xaxis label="carat"

labelattrs=(color=white weight=bold)

values = (**0** **1** **2** **3**)

valueattrs=(color=white)

minor display=(noline)

;

yaxis label="Price of Diamonds"

labelattrs=(color=white weight=bold)

valueattrs=(color=white)

grid

gridattrs=(color=lightgray)

minorgrid

minorgridattrs=(color=lightgray)

display=(noline noticks)

min=**0** max=**20000**

;

format price DOLLAR.;

**Run**;

ODS graphics off;

DSPS\_DIAMONDS\_Correlation\_Heatmap\_Type1

ODS graphics on;

ODS output PearsonCorr=Diamonds\_CORR;

**PROC** **CORR** data=mylib.clean\_53775;

VAR price carat depth table x y z;

**RUN**;

**PROC** **Sort** Data=Diamonds\_CORR;

By variable;

**Run**;

**Proc** **Transpose** Data=Diamonds\_CORR out=Diamonds\_Long1 (rename=(COL1=Correlation))

name = CorrelationID;

var price carat depth table x y z;

By variable;

**Run**;

**Proc** **Transpose** Data=Diamonds\_CORR out=Diamonds\_Long2 (rename=(COL1=p\_value))

name = PvalueID;

var Pprice Pcarat Pdepth Ptable Px Py Pz;

By variable;

**Run**;

**Data** Diamonds\_CORR\_long;

merge Diamonds\_Long1

Diamonds\_Long2;

By variable;

label CorrelationID = "Correlations";

**Run**;

**Proc** **Sort** Data=Diamonds\_CORR\_long;

By variable correlationID;

**Run**;

/\* Create a heat map graph with P-Values in the squares \*/

**proc** **sgplot** data=Diamonds\_CORR\_long noautolegend;

heatmap x=Variable y=CorrelationID / colorresponse=Correlation

name="nope1"

discretex discretey x2axis

colormodel=ThreeColorRamp;

/\*Colorresponse allows discrete squares for each correlation. x2axis bring the label to the top \*/

text x=Variable y=CorrelationID text=p\_value / textattrs=(size=**10**pt)

x2axis name='nope2';

/\*To overlay significance, create a variable that contans that info and set text=VARIABLE \*/

label correlation='Pearson Correlation';

yaxis reverse display=(nolabel);

x2axis display=(nolabel);

gradlegend;

**run**;

DSPS\_DIAMONDS\_Frequency\_Analysis

ODS graphics on;

**proc** **freq** data=mylib.clean\_53775;

tables cut\*(color clarity) / chisq plots=freqplot(orient=horizontal twoway=stacked scale=percent);

tables color\*clarity / chisq plots=freqplot(orient=horizontal twoway=stacked scale=percent);

**Run**;

DSPS\_DIAMONDS\_Stacked Bar

**proc** **freq** data=mylib.clean\_53775 order=freq noprint;

tables color\*cut / out=FreqOut(where=(percent^=**.**));

**run**;

ods graphics /height=**500**px width=**800**px;

title "Counts of Diamonds Color by Cut";

**proc** **sgplot** data=FreqOut;

hbarparm category=Color response=count / group=Cut

seglabel seglabelfitpolicy=none seglabelattrs=(weight=bold);

keylegend / opaque across=**1** position=bottomright location=inside;

xaxis grid;

yaxis labelpos=top;

**run**;

DSPS\_DIAMONDS\_Boxplot\_Carat

ODS graphics on /reset width=**12**in height=**5**in;

title 'Diamonds Carat by Cut';

**proc** **sgplot** data=mylib.clean\_53775;

vbox carat / category=cut

dataskin=sheen

outlierattrs=(color=green)

meanattrs=(color=black)

medianattrs=(color=black)

connect=mean connectattrs=(color=red);

**run**;

title 'Diamonds Carat by Color';

**proc** **sgplot** data=mylib.clean\_53775;

vbox carat / category=color

dataskin=sheen

outlierattrs=(color=green)

meanattrs=(color=black)

medianattrs=(color=black)

connect=mean connectattrs=(color=red);

**run**;

title 'Diamonds Carat by Clarity';

**proc** **sgplot** data=mylib.clean\_53775;

vbox carat / category=clarity

dataskin=sheen

outlierattrs=(color=green)

meanattrs=(color=black)

medianattrs=(color=black)

connect=mean connectattrs=(color=red);

**run**;

-----------------------------------------------------------------------------------------------------------------

1. **SurverSelect:split dataset. Output lists:**

* **DIAMONDS\_TRAIN\_VALID: 53775 rows**
* **DIAMONDSFE\_TRAINING: 37643 rows**
* **DIAMONDSFE\_VALIDATION: 16132 rows**
* **HTML: The SURVEYSELECT Procedure**

1. **Diamond\_Pricing\_Model A.0.1: Proc Reg outputs:**

* **ModelA\_Parameters**
* **HTML: The REG Procedure with target logprice**

1. **Diamonds\_Evaluate\_Model A.0.1: Proc Reg**

* **HTML: Analysis of Variance, Fit Diagnostics plots**

1. **Test\_Diamonds\_FeatureEngineer: generate proper test file**

* **TEST\_DIAMONDS\_FE**

1. **Diamond\_PredictPrice\_Model\_A: three outputs:**

* **MODEL\_A**
* **PRICE\_3000DIAMONDS\_DETAILS**
* **PRICE\_3000DIAMONDS**

1./\* split dataset to 70% training and 30% validation \*/

**proc** **surveyselect** data=mylib.diamonds\_FE2

out=Diamonds\_Train\_Valid

method=SRS

samprate=**0.7** /\* Wanted Training Dataset 70% \*/

seed=**1357924**

outall;

**run**;

**Data** Mylib.DiamondsFE\_Training Mylib.DiamondsFE\_Validation;

Set Diamonds\_Train\_Valid;

If Selected Then

output Mylib.DiamondsFE\_Training;

Else

output Mylib.DiamondsFE\_Validation;

**Run**;

2./\*train the model: Target:logprice

Features: logcarat, dummies for cut, color, clarity

**\*/**

**Proc** **Reg** data=Mylib.DiamondsFE\_Training;

\*PLOTS(MAXPOINTS=None) plots=diagnostics;

\*ODS Select FitStatistics ParameterEstimates;

ODS Output ParameterEstimates=Mylib.Model\_A\_Parameters;

Model\_A: Model logprice = logcarat

cut1-cut5

color1-color7

clarity1-clarity8

;

**Run**;

3. /\*validate the 30% test data\*/

ODS graphics on/reset width=**8**in height=**5**in;

**Proc** **Reg** data=mylib.DiamondsFE\_validation

PLOTS(Maxpoints=None)

PLOTS=diagnostics;

Model logprice = logcarat

cut1-cut5

color1-color7

clarity1-clarity8

;

**Run**;

ODS Graphics off;

4./\* Import 3000 diamonds dataset to predict the price with the model. Apply Feature Engineering To Test Diamonds

-Natural Log Transformation for Carat

-One Hot Encoding for Categorical Variables

-Ordinal Variable for Categorical Variables

\*/

**Data** mylib.TestDiamonds\_FE (drop=i j k);

Set mylib.new\_diamond;

logcarat = log(**1**+carat);

Array cut\_grade[**5**] $**9** \_temporary\_ ('Fair' 'Good' 'Very Good' 'Premium' 'Ideal');

Array color\_grade[**7**] $**1** \_temporary\_ ('J' 'I' 'H' 'G' 'F' 'E' 'D');

Array clarity\_grade[**8**] $**4** \_temporary\_ ('I1' 'SI2' 'SI1' 'VS2' 'VS1' 'VVS2' 'VVS1' 'IF');

Array dummy\_cut[\*] cut1 - cut5;

Array dummy\_color[\*] color1 - color7;

Array dummy\_clarity[\*] clarity1 - clarity8;

Do i = **1** to dim(dummy\_cut);

Do j = **1** to dim(dummy\_color);

Do k = **1** to dim(dummy\_clarity);

dummy\_cut(i) = **0**;

dummy\_color(j) = **0**;

dummy\_clarity(k) = **0**;

End;

End;

End;

Do i = **1** to dim(cut\_grade);

If cut\_grade[i] = Strip(cut) Then dummy\_cut[i] = **1**;

End;

Do i = **1** to dim(color\_grade);

If color\_grade[i] = Strip(color) Then dummy\_color[i] = **1**;

End;

Do I = **1** to dim(clarity\_grade);

If clarity\_grade[i] = Strip(clarity) Then dummy\_clarity[i] = **1**;

End;

Select (cut);

when ('Fair') \_cut\_ord = **1**; /\* Lowest level of fire and brilliance \*/

when ('Good') \_cut\_ord = **2**;

when ('Very Good') \_cut\_ord = **3**;

when ('Premium') \_cut\_ord = **4**;

when ('Ideal') \_cut\_ord = **5**; /\* Highest level of fire and brilliance \*/

otherwise

;

End;

Select (color);

when ('J') \_color\_ord = **1**;

when ('I') \_color\_ord = **2**;

when ('H') \_color\_ord = **3**;

when ('G') \_color\_ord = **4**; /\* G-J = Nearly Colorless \*/

when ('F') \_color\_ord = **5**;

when ('E') \_color\_ord = **6**;

when ('D') \_color\_ord = **7**; /\* D-F = Colorless is highest color grade \*/

otherwise

;

End;

Select (clarity);

when ('I1') \_clarity\_ord = **1**; /\* Inclusions 1 is the worst \*/

when ('SI2') \_clarity\_ord = **2**; /\* Small Inclusions 1 \*/

when ('SI1') \_clarity\_ord = **3**; /\* Small Inclusions 2 \*/

when ('VS2') \_clarity\_ord = **4**; /\* Very Small Inclusions 1 \*/

when ('VS1') \_clarity\_ord = **5**; /\* Very Small Inclusions 2 \*/

when ('VVS2') \_clarity\_ord = **6**; /\* Very Very Small Inclusions 1 \*/

when ('VVS1') \_clarity\_ord = **7**; /\* Very Very Small Inclusions 2 \*/

when ('IF') \_clarity\_ord = **8**; /\* Internally Flawless is the best \*/

otherwise

;

End;

**Run**;

5./\* Predict Each Price of 3000 Diamonds \*/

Options MPRINT MLOGIC SYMBOLGEN;

**Proc** **Transpose** Data=mylib.Model\_A\_Parameters

Out=Model\_A (drop=\_LABEL\_ \_NAME\_)

;

Var Estimate;

**Run**;

**Data** \_NULL\_;

Set Model\_A;

Call Symputx('Beta0', PUT(COL1, **10.5**)); /\* Intercept \*/

Call Symputx('Beta1', PUT(COL2, **10.5**)); /\* logcarat \*/

Call Symputx('Beta2', PUT(COL3, **10.5**)); /\* cut1 \*/

Call Symputx('Beta3', PUT(COL4, **10.5**)); /\* cut2 \*/

Call Symputx('Beta4', PUT(COL5, **10.5**)); /\* cut3 \*/

Call Symputx('Beta5', PUT(COL6, **10.5**)); /\* cut4 \*/

Call Symputx('Beta6', PUT(COL7, **10.5**)); /\* color1 \*/

Call Symputx('Beta7', PUT(COL8, **10.5**)); /\* color2 \*/

Call Symputx('Beta8', PUT(COL9, **10.5**)); /\* color3 \*/

Call Symputx('Beta9', PUT(COL10, **10.5**)); /\* color4 \*/

Call Symputx('Beta10', PUT(COL11, **10.5**)); /\* color5 \*/

Call Symputx('Beta11', PUT(COL12, **10.5**)); /\* color6 \*/

Call Symputx('Beta12', PUT(COL13, **10.5**)); /\* clarity1 \*/

Call Symputx('Beta13', PUT(COL14, **10.5**)); /\* clarity2 \*/

Call Symputx('Beta14', PUT(COL15, **10.5**)); /\* clarity3 \*/

Call Symputx('Beta15', PUT(COL16, **10.5**)); /\* clarity4 \*/

Call Symputx('Beta16', PUT(COL17, **10.5**)); /\* clarity5 \*/

Call Symputx('Beta17', PUT(COL18, **10.5**)); /\* clarity6 \*/

Call Symputx('Beta18', PUT(COL19, **10.5**)); /\* clarity7 \*/

**Run**;

**Data** Price\_3000Diamonds\_Detail;

Set mylib.TestDiamonds\_FE;

Format predicted\_price **10.2**;

LogPrice = &Beta0 +

&Beta1\*logcarat +

&Beta2\*cut1 +

&Beta3\*cut2 +

&Beta4\*cut3 +

&Beta5\*cut4 +

&Beta6\*color1 +

&Beta7\*color2 +

&Beta8\*color3 +

&Beta9\*color4 +

&Beta10\*color5 +

&Beta11\*color6 +

&Beta12\*clarity1 +

&Beta13\*clarity2 +

&Beta14\*clarity3 +

&Beta15\*clarity4 +

&Beta16\*clarity5 +

&Beta17\*clarity6 +

&Beta18\*clarity7

;

Predicted\_Price = exp(logPrice)-**1**;

**Run**;

**Data** Price\_3000Diamonds;

Set Price\_3000Diamonds\_Detail;

Keep rec\_id

carat

clarity

color

cut

predicted\_price

;

**Run**;

**Proc** **Sort** Data=Price\_3000Diamonds;

By carat clarity color cut;

**Run**;

Model Codes:

1. **Proc** **GLM:build the model**

* **HTML: The GLM Procedure**

1. **Prepare 3000 Diamonds (UNKNOWN\_TESTDIAMONDS) for prediction and predict the prices**

* **UNKNOWN\_TESTDIAMONDS : 19132 rows**
* **DIAMONDFE\_VALIDATION: 16132 rows**
* **TESTDIAMONDS\_FE:3000 rows**
* **DIAMONDS\_PREDICTED\_MODEL\_A:19132**
* **DIAMOND\_PREDICTED\_MODEL\_A\_GLM:3000**
* **HTML: The GLM Procedure**

1./\* Proc GLM\*/

Title;

**Proc** **GLM** data=Mylib.DiamondsFE\_Training;

CLASS cut color clarity;

MODEL\_A: Model logprice = logcarat cut color clarity / SOLUTION;

**Run**;

2./\* Prediction process\*/

/\* Prepare Unknown dataset for prediction \*/

**Data** DiamondsFE\_Validation (keep=DS logprice logcarat cut color clarity);

Retain DS logprice logcarat cut color clarity;

Set Mylib.DiamondsFE\_Validation;

Format DS $10.;

Format cut $11.;

Format color $3.;

Format clarity $6.;

DS = "VALIDATION";

**Run**;

**Data** TestDiamonds\_FE (Keep=DS logcarat cut color clarity);

Retain DS logcarat cut color clarity;

Set Mylib.TestDiamonds\_FE;

Format DS $10.;

Format cut $11.;

Format color $3.;

Format clarity $6.;

DS = "TEST";

**Run**;

**Data** Mylib.Unknown\_TestDiamonds;

Set DiamondsFE\_Validation TestDiamonds\_FE;

**Run**;

**Proc** **GLM** data=Mylib.Unknown\_TestDiamonds;

Class Cut Color Clarity;

Model logprice = logcarat cut color clarity / solution;

output out=Diamonds\_Predicted\_Model\_A

predicted=Predicted\_Price\_A;

**Run**;

**Data** Mylib.Diamonds\_Predicted\_Model\_A (drop=logprice);

Set Diamonds\_Predicted\_Model\_A;

If DS = "TEST";

Predicted\_Price\_A = exp(Predicted\_Price\_A)-**1**;

**Run**;