EDA in Practice: Diamonds (are a girl’s best friend)

~~~
---
output: Basic data exploration: Diamonds
  html_document: default
  pdf_document: default
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
### Free memory Functions
# Clear environment
rm(list = ls())

# Clear packages
# pacman::p_unload(rgl)

# Clear plots
# dev.off()  # But only if there IS a plot

# Clear console
cat("\014")  # ctrl+L

## Reading in and basic formatting of data
Start by reading in and formatting the data.
## Reading in diamond dataset from ggplot2

require(ggplot2)
data(diamonds)
## Graphical Methods
~~~

EDA in Practice: Diamonds (are a girl's best friend)

Now that we've checked for missing values and typos and made corrections, we can graphically examine the sample data distribution of our data. Frequency distributions are useful because they can help us visualize the center (e.g., RV) and spread or dispersion (e.g., low, and high) of our data. Typically, in introductory statistics the normal (i.e., Gaussian) distribution is emphasized.

Short Description of Graphical Methods

Plot Types Description

Bar: a plot where each bar represents the frequency of observations for a 'group'

Histogram: a plot where each bar represents the frequency of observations for a 'given range of values'

Density: an estimation of the frequency distribution based on the sample data

Box-Whisker: a visual representation of median, quartiles, symmetry, skewness, and outliers

Scatter & Line: a graphical display of one variable plotted on the x axis and another on the y axis

Quantile-Quantile: a plot of the actual data values against a normal distribution

## Histogram

A histogram is like a bar plot, except that instead of summarizing categorical data, it categorizes a continuous variable like clay content into non-overlapping intervals for the sake of display. The number of intervals can be specified by the user, or can be automatically determined using an algorithm, such as nclass. Sturges(). Since histograms are dependent on the number of bins, for small datasets they're not the best method of determining the shape of a distribution.

# Histogram - base graphics

# Create a histogram of "carat" using a. base graphics and b. using ggplot2 and binwidth=0.5

```{r}
```

# Create a histogram of "carat" using ggplot2 and binwidth=0.5

```{r}
```

## Create an object, smaller, and zoom into only the diamonds with a size of less than three carats and choose a smaller binwidth.

## Use *geom_freqpoly()* to overlay **multiple** 'histograms' in the same plot.

To make it easy to see the unusual values, we need to zoom in to small values of the y-axis with coord_cartesian():

## Filter out unusual values by filtering data where y< 3 or y>20. Arrange the values of y in the dataframe.

Missing Values

## Replace the unusual values with missing values, NA, using mutate() to create a new variable, y, and the ifelse() function.

Explore the distribution of price. Do you discover anything unusual or surprising? (Hint: carefully think about the bin width and make sure you try a wide range of values.)

Try binwidth =10, binwidth=100,

How many diamonds are 0.99 carat?
How many are 1 carat?
What do you think is the cause of the difference?

## Create a histogram of the variable, cut, using ggplot2. Indicate the level of order.

## Create a factor of variable cut.

## Recreate the histogram of the variable, cut, using ggplot2.

## Density Curve

A density estimation, also known as a Kernel density plot, generally provides a better visualization of the shape of the distribution in comparison to the histogram. Compared to the histogram where the y-axis represents the number or percent (i.e., frequency) of observations, the y-axis for the density plot represents the probability of observing any given value, such that the area under the curve equals one. One curious feature of the density curve is the hint of two peaks (i.e. bimodal distribution?). Given that our sample includes a mixture of surface and subsurface horizons, we may have two different populations. However, considering how much the two distributions overlap, it seems impractical to separate them in this instance.

## Create a density plot of the variable, carat, using ggplot2. Fill the columns with a color.

## Density plot using ggplot2

## Box plots

Box plots are a graphical representation of the five number summary, depicting quartiles (i.e. the 25%, 50%, and 75% quantiles), minimum, maximum and outliers (if present). Boxplots convey the shape of the data distribution, the presence of extreme values, and the ability to compare with other variables using the same scale, providing an excellent tool for screening data, determining thresholds for variables and developing working hypotheses.

The parts of the boxplot are shown in the figure below. The "box" of the boxplot is defined as the 1st quartile, (Q1 in the figure) and the 3rd quartile, (Q3 in the figure). The median, or 2nd quartile, is the dark line in the box. The whiskers (typically) show data that is 1.5 * IQR above and below the 3rd and 1st quartile. Any data point that is beyond a whisker is considered an outlier.

That is not to say the outlier points are in error, just that they are extreme compared to the rest of the dataset. However, you may want to evaluate these points to ensure that they are correct.

## Create a boxplot of the variable, carat, using base graphics

## Boxplots with base graphics

## Create a boxplot of the variable, carat, using ggplot2. Center the plot at x=1.

## Create multiple boxplots of the variable, carat by diamond cut.

## Create a violin plot using the variable, carat by the diamond cut.

## Violin Plots using ggplot2

```
require(gridExtra)

p1 <- ggplot(diamonds,aes(y=carat,x=cut)) + geom_point() + geom_violin()

p2 <- ggplot(diamonds,aes(y=carat,x=cut)) + geom_boxplot()

grid.arrange(p1, p2, ncol=2)
```

## Scatter plot

Plotting points of one ratio or interval variable against another is a scatter plot. Plots can be produced for a single or multiple pairs of variables. Many independent variables are often under consideration in soil survey work. This is especially common when GIS is used, which offers the potential to correlate soil attributes with a large variety of raster datasets.

The purpose of a scatterplot is to see how one variable relates to another. With modeling in general the goal is parsimony (i.e., simple). The goal is to determine the fewest number of variables required to explain or describe a relationship. If two variables explain the same thing, i.e., they are highly correlated, only one variable is needed. The scatterplot provides a perfect visual reference for this.

## Create a scatterplot using the variables, carat and price using a. base graphics, b. without using a formula and c. ggplot2 and distinguish the color of the diamond in the legend.

## Scatterplot with base graphics

## Scatterplot without using formula

## Scatterplot using ggplot2

## Create a scatterplot using the variables, carat, and price. A. facet wrap using variable, color and b. facet grid using variables cut and clarity.

## Facet_wrap and facet_grid in ggplot2

## Create a scatterplot using the variables, carat, and price. A. facet wrap using variable, color and b. facet grid using variables cut and clarity.

## Preparing data for multiple line charts

## Theme in ggplot2

There are seven other themes built in to ggplot2 1.1.0:

theme_bw(): a variation on theme_grey() that uses a white background and thin grey grid lines.

theme_linedraw(): A theme with only black lines of various widths on white backgrounds, reminiscent of a line drawing.

theme_light(): similar to theme_linedraw() but with light grey lines and axes, to direct more attention towards the data.

theme_dark(): the dark cousin of theme_light(), with similar line sizes but a dark background. Useful to make thin coloured lines pop out.

theme_minimal(): A minimalistic theme with no background annotations.

theme_classic(): A classic-looking theme, with x and y axis lines and no gridlines.

theme_void(): A completely empty theme.

```
library(ggthemes)
g2 <- ggplot(diamonds, aes(x=carat,y=price)) +
     geom_point(aes(color=color))

## Let’s apply few themes
p1 <- g2 + theme_economist() + scale_color_economist()
p2 <- g2 + theme_excel() + scale_color_excel()
p3 <- g2 + theme_tufte()
p4 <- g2 + theme_wsj()
grid.arrange(p1, p2, p3, p4, nrow=2, ncol=2)

## Find the frequency count of each level in the variable "cut".

## Create a scatterplot of carat and price.

## Create a histogram of carat using the binwidht = 0.75

## Create a frequency count by create bins of "carat" variable by 0.5

## Create a data frame, small, by filtering the carat size less than 3.

## Create a histograms and frequency polygon plot of the variable "carat". Use the freqpoly geometry.
Use the option color to highlight the different "cut"s.
```