Experimenting with EDA Techniques

Logan Cones

June 1, 2019

# Introduction

We are going to take a look at a couple Kaggle Datasets as a refresher of EDA techniques in R.

setwd('C:\\Users\\Logan Cones\\Desktop\\2019\\R Book\\')  
df <- read.csv('cereal.csv')  
head(df)

## name mfr type calories protein fat sodium fiber  
## 1 100% Bran N C 70 4 1 130 10.0  
## 2 100% Natural Bran Q C 120 3 5 15 2.0  
## 3 All-Bran K C 70 4 1 260 9.0  
## 4 All-Bran with Extra Fiber K C 50 4 0 140 14.0  
## 5 Almond Delight R C 110 2 2 200 1.0  
## 6 Apple Cinnamon Cheerios G C 110 2 2 180 1.5  
## carbo sugars potass vitamins shelf weight cups rating  
## 1 5.0 6 280 25 3 1 0.33 68.40297  
## 2 8.0 8 135 0 3 1 1.00 33.98368  
## 3 7.0 5 320 25 3 1 0.33 59.42551  
## 4 8.0 0 330 25 3 1 0.50 93.70491  
## 5 14.0 8 -1 25 3 1 0.75 34.38484  
## 6 10.5 10 70 25 1 1 0.75 29.50954

# Exploratory Data Analysis

Many different types of metrics to get the middle location or descripitive statistics

* Simple Mean
* Trimmed Mean helps with outliers and is different than the median
* Tried and True Center value or Median

#Simple Mean  
mean(df$calories)

## [1] 106.8831

#Trimmed Mean  
mean(df$calories, trim=.1)

## [1] 107.3016

#Median  
median(df$calories)

## [1] 110

#This is similiar to describe() in python gives you a lot of descriptive statistics  
#summary(df)

#could also experiment with a weighted mean  
library(matrixStats)  
library(dplyr)  
weighted.mean(df$calories, w = df$protein)

## [1] 107.0408

weightedMedian(df$calories, w = df$protein)

## [1] 110

## Variability

You also want to be concerned with the variability in the data \* Variance is the sum of squared deviations from the mean divided by n - 1 \* Standard deviation is the square root of the variance \* IQR 25th percentile and 75th percentile

sd(df$calories)

## [1] 19.48412

IQR(df$calories)

## [1] 10

#Mean Absolute Deviation from the Median  
mad(df$calories)

## [1] 14.826

### Why do we use Degrees of Freedom?

It's all about wanting to make estimates about a population and not based on a sample Essnetially if you divide by n you will underestimate the true value of the variance and the standard deviation of the population. This is referred to as a biased estimate N-1 results in unbiased estimate.

## Exploring the Distrubutions

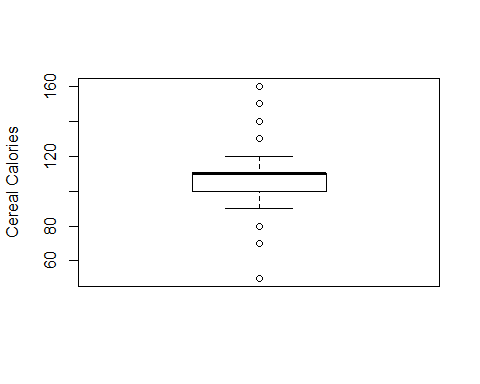
Easy Techniques for doing this

* Frequency Table
* Boxplot (my personal fav)
* Density Plot
* Histogram

quantile(df$calories, p=c(.01, .05, .25, .5, .75, .95, .99))

## 1% 5% 25% 50% 75% 95% 99%   
## 50.0 70.0 100.0 110.0 110.0 140.0 152.4

boxplot(df$calories, ylab = "Cereal Calories")



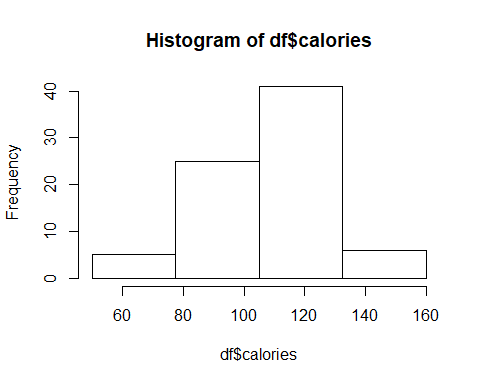
Frequency Tables are pretty cool

breaks <- seq(from= min(df$calories),  
 to = max(df$calories), length = 5)  
pop\_freq <- cut(df$calories, breaks = breaks, right = TRUE, include.lowest = TRUE)  
table(pop\_freq)

## pop\_freq  
## [50,77.5] (77.5,105] (105,132] (132,160]   
## 5 25 41 6

Histogram

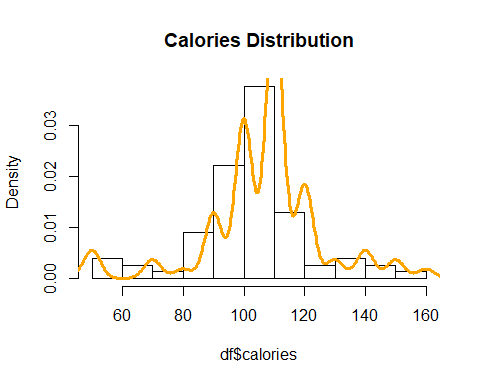
hist(df$calories, breaks = breaks)



Density Plots

* Skewness refers to whether the data is skewed to larger or smaller values
* Kurtosis indicates the propensity of the data to have extreme values

hist(df$calories, freq = FALSE, main = 'Calories Distribution')  
lines(density(df$calories), lwd = 3, col = 'orange')



### Exploring Categorical Data

* Use a bar chart which is different than a histogram because a histogram represents values of a single variable on a numeric scale and a bar chart represents different categories of a factor variable

### Correlation

* High correlation means that if high values of X go with high values of Y and low values of x go with low values of Y, it is vice versa then it is negatively corrleated

#### Key Terms

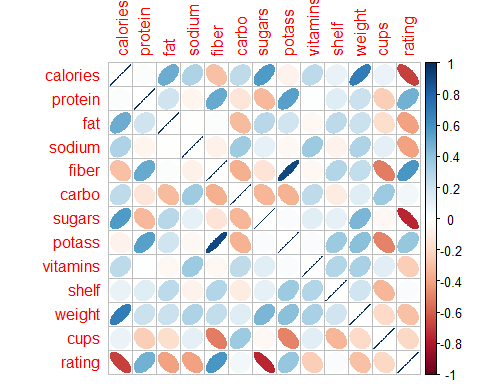
* Correlation Coefficient: A metric that measures the extent to which the numeric vairables are assoicated with one another 1 or -1 is perfect and often a problem in modeling \*\* If relationship not linear this is pretty useless
* Correlation Matrix: A table where the variables are shown on both rows and columns and the values are the correlations between the variables. Super super helpful techniques if limited features
* Scatterplot: A plot in which the x-axis is the value of one variable and the y-axis is the value of another

#### Correlation in Action

Make a new dataframe of all numeric variables and we what the correlations are

Somewhat surprising more calories is negatively correlated with higher ratings! Must have surveys some mature adults :)

library(corrplot)  
library(dplyr)  
nums <- select\_if(df, is.numeric)  
corrplot(cor(nums), method = 'ellipse')

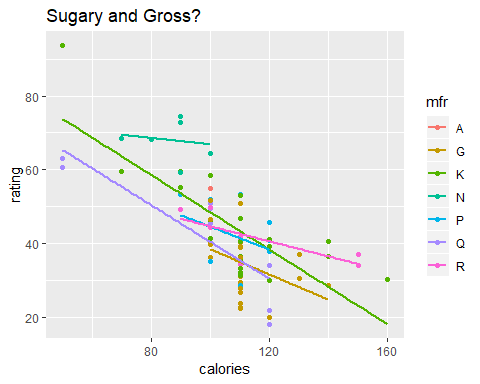


The above graph is based on Pearson's Correlation which is based purely on value and is not robust to outliers. Other Correlations such as Spearman's Rho and Kendall's Tau use Rank of the data. This could be useful in certain situtations.

### Scatterplot

Let's take another look at the Rating and Carolies combo

library(ggplot2)  
ggplot(data = df, aes(x = calories, y = rating, color=mfr), xlab = 'Calories', ylab = 'Rating') + ggtitle('Sugary and Gross?') + geom\_point() + geom\_smooth(method=lm, se = FALSE)



## Exploring Two or More Variables

Key Terms

* Contingency Tables: A tally of counts between two or more categorical variables
* Hexagonal Binning: A plot of two numeric tables with records binned into hexagons
* Contour Plots: A plot showing the density of two numeric variables like topographical mapping
* Violin Plots Similar to a boxplor but showing the density estimate. These are beautiful in the python package seaborn.

I'm going to load in a new dataset to do these tests with for the Cereal data is too small.

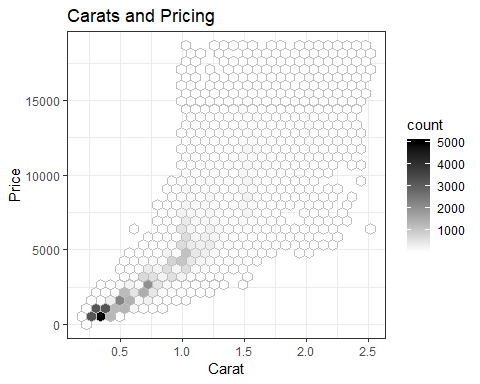
## Diamond Dataset

I recently got engaged to my now fiance so we will take a look at the diamond dataset to learn more about these techniques. This was pulled from everyone's favorite site Kaggle

df2 <- read.csv('diamonds.csv')  
head(df2)

## X carat cut color clarity depth table price x y z  
## 1 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43  
## 2 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31  
## 3 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31  
## 4 4 0.29 Premium I VS2 62.4 58 334 4.20 4.23 2.63  
## 5 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75  
## 6 6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48

#let's get rid of the crazy people that have more than 2.5 Carats cause that's insane  
df2 <- subset(df2, carat < 2.5)  
  
ggplot(df2, (aes(x=carat, y = price)))+ ggtitle('Carats and Pricing' ) + stat\_bin\_hex(colour= 'gray') + theme\_bw() + scale\_fill\_gradient(low='white', high = 'black') + labs(x="Carat", y = 'Price')



Not sure this next technique is going to be super useful here but it's an interesting approach to exploration as is. Take two categorical variables and make a contingency table of them

library(descr)  
CrossTable(df2$clarity, df2$cut, prop.c = FALSE, prop.chisq=FALSE, prop.t = FALSE)

## Cell Contents   
## |-------------------------|  
## | N |   
## | N / Row Total |   
## |-------------------------|  
##   
## ==================================================================  
## df2$cut  
## df2$clarity Fair Good Ideal Premium Very Good Total  
## ------------------------------------------------------------------  
## I1 196 94 141 193 80 704  
## 0.278 0.134 0.200 0.274 0.114 0.013  
## ------------------------------------------------------------------  
## IF 9 71 1212 230 268 1790  
## 0.005 0.040 0.677 0.128 0.150 0.033  
## ------------------------------------------------------------------  
## SI1 408 1560 4278 3570 3238 13054  
## 0.031 0.120 0.328 0.273 0.248 0.243  
## ------------------------------------------------------------------  
## SI2 452 1066 2582 2919 2086 9105  
## 0.050 0.117 0.284 0.321 0.229 0.169  
## ------------------------------------------------------------------  
## VS1 170 648 3588 1988 1775 8169  
## 0.021 0.079 0.439 0.243 0.217 0.152  
## ------------------------------------------------------------------  
## VS2 261 977 5070 3355 2591 12254  
## 0.021 0.080 0.414 0.274 0.211 0.228  
## ------------------------------------------------------------------  
## VVS1 17 186 2047 616 789 3655  
## 0.005 0.051 0.560 0.169 0.216 0.068  
## ------------------------------------------------------------------  
## VVS2 69 286 2606 870 1235 5066  
## 0.014 0.056 0.514 0.172 0.244 0.094  
## ------------------------------------------------------------------  
## Total 1582 4888 21524 13741 12062 53797  
## ==================================================================

Then of course my favorite and the one that seems most useful in my current role is box plots of cateogrical variables. Let's make some Violin plots with clarity and price. Even though I just bought a ring I didn't dive this deep into the process. WISH I DID THIS BEFORE HAND! Doesn't really seem like splitting it up by cut moves the needle as much as I would have thought.

ggplot(df2, aes(clarity, price, color= clarity)) + geom\_violin() + labs(x = 'Clarity', y = 'Cost') + facet\_wrap('cut') + ggtitle('Clairty vs Price by Cut')

