# Module 1 Assignment 3 - BAN 502

## Kristian Marlowe

### R Refresher

# The following code is to install and load packages  
  
#install.packages(tidyverse)  
library(tidyverse)

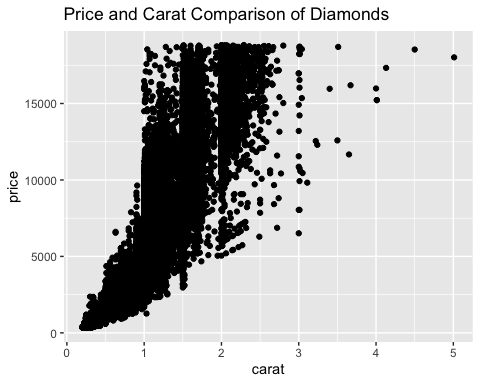
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.0 ✓ dplyr 1.0.5  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

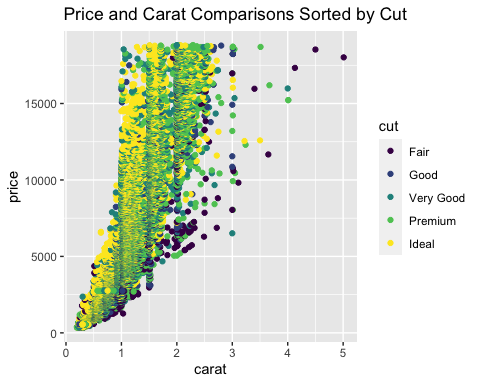
# The following code will read in the diamonds data set  
  
diamonddata = diamonds

# The following code will create a scatter plot comparing carat and price  
ggplot(diamonddata, aes(x=carat, y=price)) +  
 geom\_point() +   
 labs(title= "Price and Carat Comparison of Diamonds")



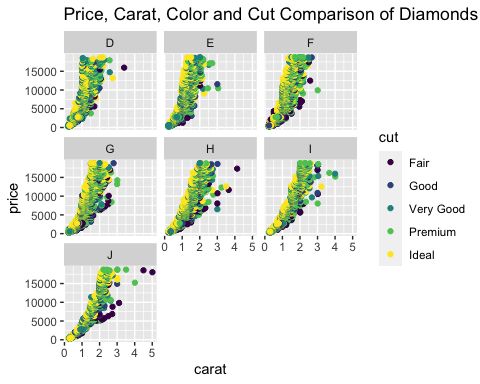
**Generally speaking, as the carat size increases, the price of the diamond increases. We should notice a more exponential/logistic type of growth, with the bulk of the carat sizes ranging from ~ 0.2 carats to ~ 2.5 carats.**

# The following code will create a scatter plot comparing carat and price separated into colors by the cut   
  
ggplot(diamonddata, aes(x=carat, y=price, color= cut)) +  
 geom\_point()+  
 labs(title="Price and Carat Comparisons Sorted by Cut")



**In this scatter plot comparing the carat and price by the different cuts, we can see that typically the ideal cuts are the most expensive per carat, and the fair cuts are the least expensive per carat.**

# The following code will break down the data by carat, price, cut, and color  
ggplot(diamonddata, aes(x=carat, y=price, color= cut)) +  
 facet\_wrap("color")+  
 geom\_point() +   
 labs(title = "Price, Carat, Color and Cut Comparison of Diamonds")



**When breaking down the diamonds by color, we can see that all of the colors follow a similar pattern of an exponential/logarithmic growth. It is interesting to note that worst color quality (J) typically has higher carat diamonds, while the best color quality has a smaller range of carats, with higher prices for the lower carat diamonds. Throughout the different color qualities, we see that the ideal/premium diamonds are typically higher priced, while the fair/good qualities are lower priced.**

# Load in the InventoryData.csv file   
  
InventoryData <- read\_csv("InventoryData.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## `Item SKU` = col\_character(),  
## Store = col\_character(),  
## Supplier = col\_character(),  
## `Cost per Unit ($)` = col\_double(),  
## `On Hand` = col\_double(),  
## `Annual Demand` = col\_double()  
## )

summary(InventoryData)

## Item SKU Store Supplier Cost per Unit ($)  
## Length:13561 Length:13561 Length:13561 Min. : 0.0   
## Class :character Class :character Class :character 1st Qu.: 137.0   
## Mode :character Mode :character Mode :character Median : 377.5   
## Mean : 504.4   
## 3rd Qu.: 775.5   
## Max. :1982.3   
## On Hand Annual Demand   
## Min. : 0.0 Min. : 0.0   
## 1st Qu.: 50.0 1st Qu.: 483.0   
## Median :101.0 Median : 965.0   
## Mean :100.5 Mean : 966.2   
## 3rd Qu.:151.0 3rd Qu.:1448.0   
## Max. :200.0 Max. :2150.0

**Here we can see there are 13,561 observations of 6 variables. The Item SKU, Store and Supplier are all character variables with no summary data provided. The average cost per unit is $504.40, the average on hand is 100.5, and the average annual demand is 966.2. The cost per unit ranges from $0.00 to $1982.30, the on hand ranges from 0 to 200, and the annual demand ranges from 0 to 2150.**

# We will filter data for only inventory from Supplier A  
  
inventoryA <- filter(InventoryData, Supplier=="A")

\*\* In inventory A there are 3695 rows.\*\*

# Explanation of the following code  
inventoryA = mutate(inventoryA, OnHandRatio= `On Hand` / `Annual Demand` )

**This code creates a new variable in the inventoryA file called OnHandRatio which is equal to the ‘On Hand’ variable divided by the ‘Annual Demand’ variable.**

# Create a new data frame by grouping and finding the average  
avg\_cost <- InventoryData %>%  
 group\_by(`Item SKU`) %>%  
 summarize(SKUAvgCost = `Cost per Unit ($)`)

## `summarise()` has grouped output by 'Item SKU'. You can override using the `.groups` argument.

# In my previous R/RStudio experience, I felt that piping and creating more advanced graphs beyond the scatterplot, line graph, histogram, and boxplot. Overall, I enjoyed learning wiht R/RStudio and didn't feel like it was too difficult to learn, and I've enjoyed already taking the online review course to hlep me feel more comfortable with piping and learning more about essique to help with different features of creating graphs.