# Module 1 Assignment 3

## May 15th, 2020

### Stephen Kiser

#install.packages tidyverse  
  
library(tidyverse)

## -- Attaching packages -------------------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

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

diamonddata <- diamonds  
  
diamonddata

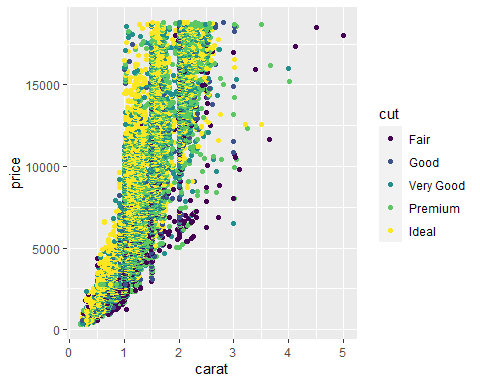
## # A tibble: 53,940 x 10  
## carat cut color clarity depth table price x y z  
## <dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43  
## 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31  
## 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31  
## 4 0.290 Premium I VS2 62.4 58 334 4.2 4.23 2.63  
## 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75  
## 6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48  
## 7 0.24 Very Good I VVS1 62.3 57 336 3.95 3.98 2.47  
## 8 0.26 Very Good H SI1 61.9 55 337 4.07 4.11 2.53  
## 9 0.22 Fair E VS2 65.1 61 337 3.87 3.78 2.49  
## 10 0.23 Very Good H VS1 59.4 61 338 4 4.05 2.39  
## # ... with 53,930 more rows

ggplot(diamonddata, aes(carat, price))+  
 geom\_point()



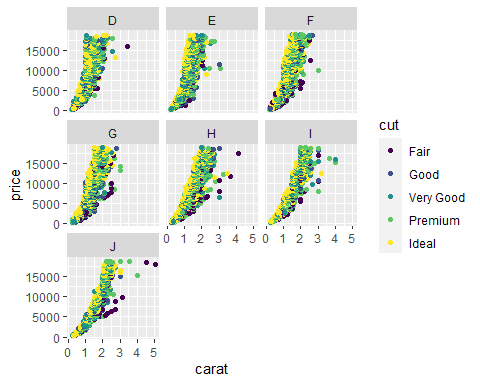
The relation that carat and price is if the carat is larger than the price for the diamond is more. As you can see the majority of diamonds that are less than 1 carat are less than $5000. Then you once you hit more than 1 carat you see the increase in price grow exponentially. There are some other factors that determine the prices of diamonds.

ggplot(diamonddata, aes(carat, price, color = cut))+  
 geom\_point()



We see in this graph the same relationship between carat and price we described in the previous task. However, this time we added the cut of the diamond to see if this had a factor in the price of the diamond. We see that if the diamond was the ideal cut then the price of the diamond would be high. For the most part of the diamonds that had a fair cut were lower priced than the other cuts.

ggplot(diamonddata, aes(carat, price, color = cut))+  
 geom\_point()+  
 facet\_wrap("color")



We see there are seven different colors of diamonds in this data set. Once again carrying on from the previous statements we see that price and carat size of diamonds saw an increasing trend. We see that the colors D, E,and G did not go over 3 carat size. These colors still sold over $15,000 however they did not get any bigger than 3 carat size. Colors F, H, I, and J went over 4 carat size. Color J was the only diamond color that went over 5 carat size.

Inventory <- read\_csv("InventoryData.csv")

## Parsed with 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()  
## )

view(Inventory)  
summary(Inventory)

## 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

inventoryA <- Inventory %>%  
 filter(Supplier== "A")  
  
view(inventoryA)

There are 3695 rows in the new data frame, inventoryA.

inventoryA = mutate(inventoryA, OnHandRatio = `On Hand` / `Annual Demand`)

This code makes a new column called, “OnHandRatio” which calculates the “ON Hand” column divided by the “Annual Demand” colum for each row.

avg\_cost <- inventoryA %>%  
 group\_by(`Item SKU`)%>%  
 summarize(SKUAvgCost = mean(`Cost per Unit ($)`))  
view(avg\_cost)  
summary(avg\_cost)

## Item SKU SKUAvgCost   
## Length:1720 Min. : 0.0   
## Class :character 1st Qu.: 180.7   
## Mode :character Median : 429.8   
## Mean : 507.4   
## 3rd Qu.: 752.8   
## Max. :1969.4

### Task 10

The most challenging topics and concepts was making of the avg\_cost dataframe. I had an idea of how to do it, but no good way to do it. I also did not remember how to do the “facet\_wrap” command.