## Module 1 Assignment 2 - R Refresher

#install.packages("tidyverse")  
library(tidyverse) #Loads Tidyverse Package

## Warning: package 'tidyverse' was built under R version 3.5.2

## -- Attaching packages -------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

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

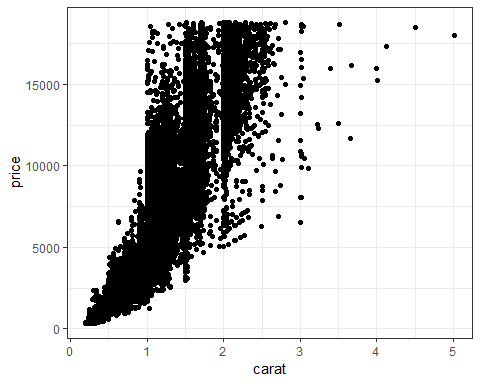
diamonddata = diamonds  
nrow(diamonddata)

## [1] 53940

ncol(diamonddata)

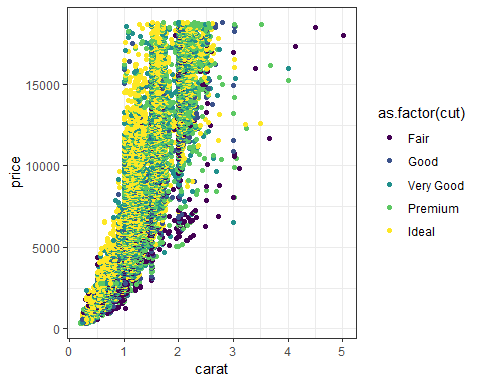
## [1] 10

ggplot(diamonddata, aes(x = carat, y = price)) + geom\_point() + theme\_bw()



In looking at the scatterplot created comparing the cara and price variables within the diamonddata dataset, it appears that tehre is a non-linear relationship between the number of carats and price. As the carat size increases so does the price at an increasing rate.

ggplot(diamonddata, aes(x = carat, y = price, color = as.factor(cut))) + geom\_point() + theme\_bw()



When running the same line of code, but adding in the cut variable, we are able to gain a clearer picture into the the factors that influence the price of a diamond. The better the cut the more quickly the price of the diamon increases. For example, the most right point is a fair cut diamond that is above 5 carats, but a very good cut 2 carat diamond is the same price.

library(readr)  
Inventory <- read\_csv("C:/Users/Evan/Desktop/BAN 502/Module 1/Assignment 2/InventoryData.csv")

## Parsed with column specification:  
## cols(  
## `Item SKU` = col\_integer(),  
## Store = col\_integer(),  
## Supplier = col\_character(),  
## `Cost per Unit ($)` = col\_double(),  
## `On Hand` = col\_integer(),  
## `Annual Demand` = col\_integer()  
## )

View(Inventory)

The invenory data set includes 6 variables: Item SKU, Store, Supplies, Cost Per Unit, On Hand and Annual Demand. Along with the 6 variables there are 13,561 observations.

InventoryA = Inventory %>% filter(Supplier == 'A')  
InventoryA = mutate(InventoryA, OnHandRatio = `On Hand` / `Annual Demand`)

The new InventoryA data frame has 3,695 rows of observations.  
The line of code shown in Task 8 creates a new variable that is a ratio by dividing two current variables against one another and adds it to the InventoryA dataframe.

avg\_cost = InventoryA %>% group\_by(`Item SKU`) %>% summarize (SKUAvgCost = mean(`Cost per Unit ($)`))  
avg\_cost

## # A tibble: 1,720 x 2  
## `Item SKU` SKUAvgCost  
## <int> <dbl>  
## 1 6 6.59  
## 2 7 11.4   
## 3 11 12.3   
## 4 13 14.3   
## 5 14 9.22  
## 6 30 26.3   
## 7 44 37.3   
## 8 46 33.5   
## 9 53 26.3   
## 10 55 46.6   
## # ... with 1,710 more rows

The only previous R experience I have is from completing MIS503, during the course I did not run into much difficulty. My biggest concern was that I would be rusty coming into this class, or any other class / job that requires R experience as I do not find mysel in R while working on school work. But this was a good refresher and I did not find myself getting lost. It did help to look back through my old course notes. I am looking forward to getting stronger in R through further applications.