# Assignment 1

## BAN 502

### Jennifer Brill

library("tidyverse")

## ── Attaching packages ──────────────────────────────────────────────────────── tidyverse 1.2.1 ──

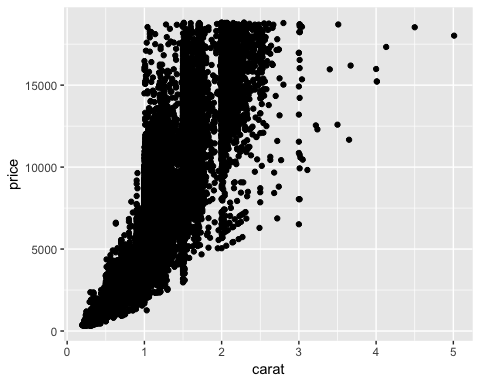
## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

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

diamonddata=diamonds

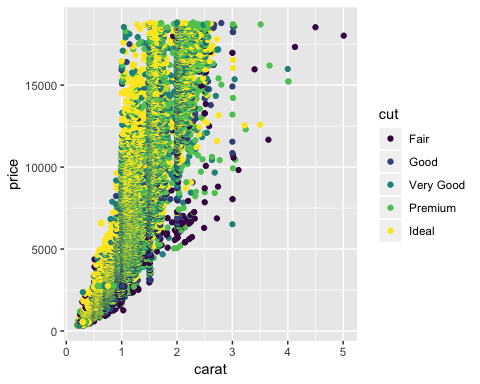
There are 10 columns and 53,940 columns in the diamonddata table.

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



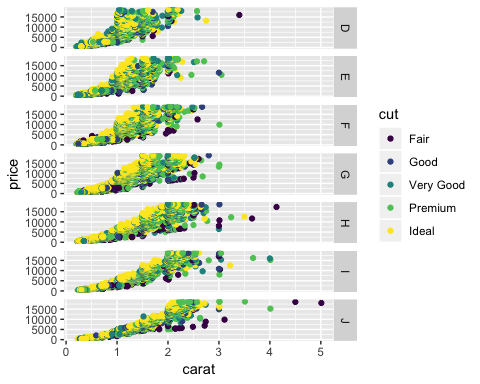
Overall, price increases with carat size. Price increases the most at 1 carat.

ggplot(diamonddata,aes(x=carat,y=price,colour=cut))+  
 geom\_point()



Cut is more strongly related to price than carat size. For each carat size, the ideal cuts are the most expensive diamonds and fair cuts are the cheapest diamonds. There are some fair cuts that are expensive, but they have large carat sizes.

ggplot(diamonddata,aes(x=carat,y=price,colour=cut))+  
 geom\_point()+  
 facet\_grid(color~.)



Colors D through F appear to be the most expensive colors of diamond because smaller carat sizes are correlated with higher prices when compared to the other diamond colors.

library(readr)  
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\_integer(),  
## `Annual Demand` = col\_integer()  
## )

View(inventory)

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

There are 3,695 rows in the dataframe inventoryA.

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

The code above creates a new column named “OnHandRatio” where the value is the amount on hand divided by the annual demand.

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

The one part of R/R Studio that I have found to be most difficult is the line coding. I am most familiar with SQL and coding in queries where spaces and line are not as important so it has been an adjustment having to pay more attention to where I place spaces and start new lines of code.