## R Refresher exercises

### Underwood, Katie

#### Module 1, Exercise 3

#install.packages(tidyverse)  
library(tidyverse)

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

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

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

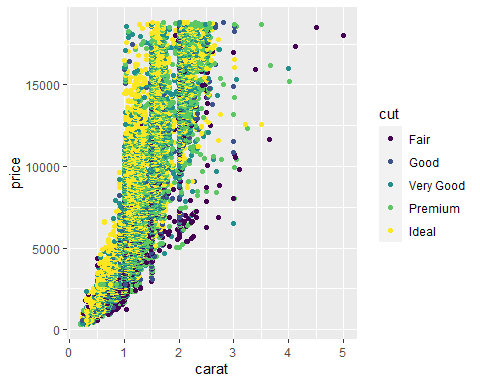
diamonddata=diamonds  
#str(diamonddata)

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



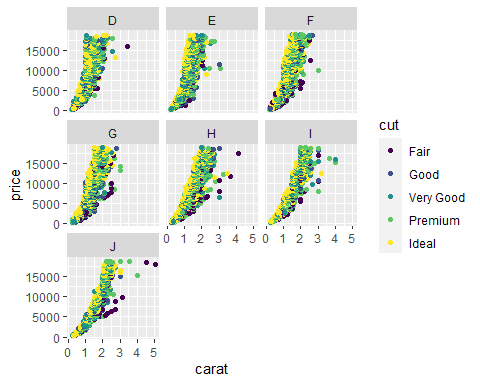
There seems to be a positive relationship between carat and price, the higher the carat, the higer the price. Another interesting observation is that there is a wide range of price point for each carat level, indicating that there must be something other than carat that influences price.

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



Cut seems to have an impact on price for diamonds with the same carat weight. It seems that Ideal and Premium cut are generally priced higher than the other cuts.

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



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

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

#str(inventory)  
#summary(inventory)

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

inventoryA data fream has 3695 rows of data.

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

This line of code creates a new variable called “OnHandRatio” which is the number of parts On Hand compared to the annual demand for that product.

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

## `summarise()` ungrouping output (override with `.groups` argument)

This was a great refresher lesson for me, even just having completed MIS503 last semester.The data camp course in particular was very nice. I hadn’t truly mastered how to combine multiple steps together (using piping) and I feel much more capable now.