# Module 1 Assignment 3

## Madison Rauscher

**Task 1**

# install.packages("tidyverse")  
# install.packages("esquisse")  
library(tidyverse)

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

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.5 ✓ dplyr 1.0.3  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

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

library(esquisse)

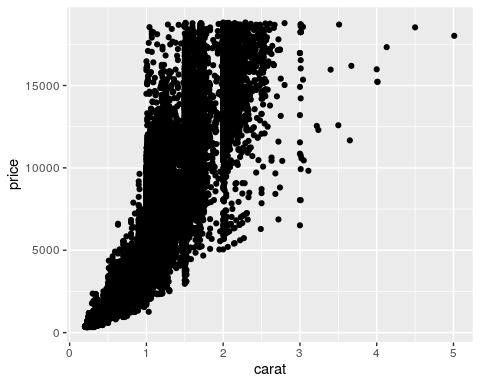
**Task 2**

diamonddata=diamonds  
# str(diamonds)

There are 53,940 rows and 10 columns in this dataset.

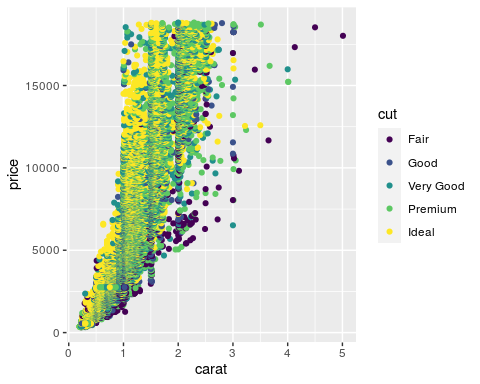
**Task 3**

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



**Task 4**

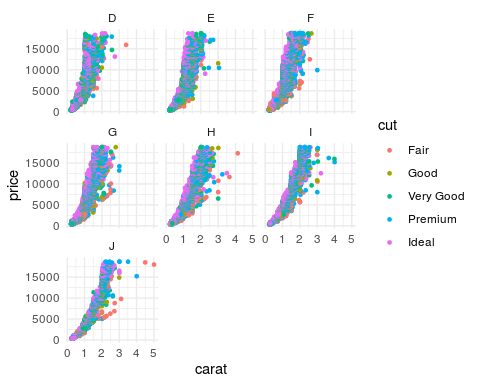
ggplot(diamonddata, aes(x=carat, y=price, color=cut)) +  
 geom\_point()



The larger carat the diamond is, the more expensive it is. At any particular carat, the price varies by cut with fair cuts being the least expensive and ideal cuts being the most expensive.

**Task 5**

ggplot(diamonddata) +  
 aes(x = carat, y = price, colour = cut) +  
 geom\_point(size = 1L) +  
 scale\_color\_hue() +  
 theme\_minimal() +  
 facet\_wrap(vars(color))



With the best color, “D”, the diamonds tend to be about 1 or 2 carats and the prices within each carat range tends to be based on the cut quality with ideal cuts being most expensive. As the color progresses downward in quality, the range of carat size broadens more and more, but the price is still correlated with cut at each carat.

**Task 6**

library(readr)  
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()  
## )

# str(InventoryData)  
# summary(InventoryData)

**Task 7**

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

There are 3695 rows in this new data frame.

**Task 8**

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

This code creates a new column to show the ratio between the “on hand” and “annual demand” columns.

**Task 9**

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

**Task 10**  
The only one that I found particularly challenging on this assignment was task 9 because I had not used this combination of commands often in the past. I also had not used the piping method, but caught on easily thorugh the Data Camp exercise.