# Module 1 Assignment 3

## Matthew Woody

### Task 1

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

### Task 2

view(diamonds)  
diamonddata<-(diamonds)  
sample(diamonddata)

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

**The Data set “diamonddata” contains 53,940 observations (rows) and 10 variables (columns).**

### Task 3

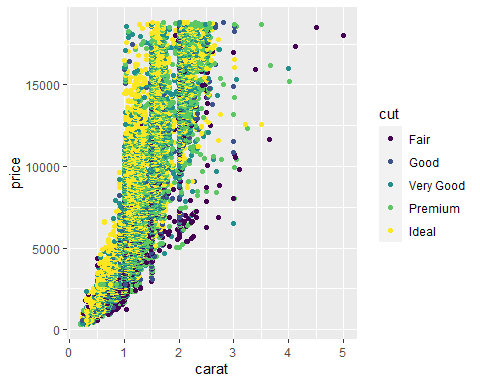
ggplot(diamonddata, aes(x=carat, y=price))+geom\_point()



**According to the scatter plot of the variables carat as related to price, it appears that higher carat diamonds sell for a higher price.**

### Task 4

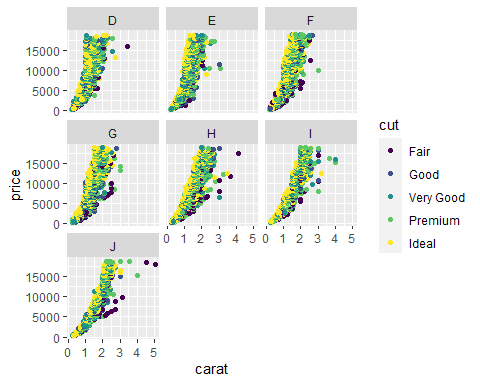
ggplot(diamonddata, aes(x=carat, y=price, color=cut))+geom\_point()



**The carat of the diamond does not seem to have any bearing on the cut of the diamond. It does seem that diamonds with a cut rated as premium or ideal do possibly command a higher price than fair or good rated diamond cuts.**

### Task 5

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



**It appears that as the color class ascends from class “D” to “J” the carat size gradually increases. It does not seem that cut has any bearing on color. It does seem that both increased carat size and ascending color class may both have an impact on price.**

### Task 6

library(readr)  
Inventory <- read\_csv("InventoryData.csv")  
View(Inventory)  
sample(Inventory)

## # A tibble: 13,561 x 6  
## `Annual Demand` Store Supplier `Cost per Unit ($)` `On Hand` `Item SKU`  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 1693 3480 A 125. 159 100  
## 2 351 1611 B 115. 40 100  
## 3 1691 1611 D 53.6 174 100  
## 4 1559 20109 B 2.26 176 100  
## 5 733 20109 C 60.5 74 100  
## 6 496 20109 D 53.7 48 100  
## 7 58 80212 B 107. 6 100  
## 8 1106 3480 B 1.33 129 11  
## 9 771 3480 C 12.9 82 11  
## 10 172 1611 C 5.16 17 11  
## # ... with 13,551 more rows

**Data Frame contains 6 columns and 13,561 observations (rows).**

### Task 7

InventoryA <- Inventory %>%  
 filter(Supplier == "A")  
sample(InventoryA)

## # A tibble: 3,695 x 6  
## Supplier `Item SKU` `On Hand` `Annual Demand` Store `Cost per Unit ($)`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 A 100 159 1693 3480 125.   
## 2 A 11 173 1695 20109 12.3   
## 3 A 113 166 1496 31779 208.   
## 4 A 113 157 1654 80212 187.   
## 5 A 122 34 290 3480 68.5   
## 6 A 122 77 680 20109 120.   
## 7 A 122 133 1239 31779 56.6   
## 8 A 13 28 277 3480 19.1   
## 9 A 13 103 962 20109 22.7   
## 10 A 13 29 297 31779 1.13  
## # ... with 3,685 more rows

**The InventoryA data frame contains 3,695 rows.**

### Task 8

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

## # A tibble: 3,695 x 7  
## Store OnHandRatio `Cost per Unit ~ Supplier `Item SKU` `Annual Demand`  
## <dbl> <dbl> <dbl> <chr> <dbl> <dbl>  
## 1 3480 0.0939 125. A 100 1693  
## 2 20109 0.102 12.3 A 11 1695  
## 3 31779 0.111 208. A 113 1496  
## 4 80212 0.0949 187. A 113 1654  
## 5 3480 0.117 68.5 A 122 290  
## 6 20109 0.113 120. A 122 680  
## 7 31779 0.107 56.6 A 122 1239  
## 8 3480 0.101 19.1 A 13 277  
## 9 20109 0.107 22.7 A 13 962  
## 10 31779 0.0976 1.13 A 13 297  
## # ... with 3,685 more rows, and 1 more variable: `On Hand` <dbl>

**The mutate code above divides the On Hand data by the Annual Demand to create a new column called OnHandRatio.**

### Task 9

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

## # A tibble: 1,720 x 2  
## `Item SKU` SKUAvgCost  
## <dbl> <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

### Task 10

**I found the various ways to format charts to be perhaps the most challenging part of working in R and RStudio. While I was able to eventually complete assignments, I did find that it required greater thought and more time to fully grasp the language for formatting charts.**