## BAN502

## Mandy Hawkins

## Task 1

library(tidyverse) #install.packages

## -- 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()

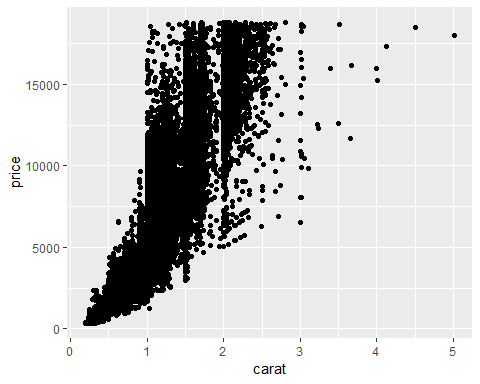
## Task 2

#The “ggplot2” package (part of the tidyverse set of packages) includes a dataset containing data on diamonds. Line of code below will read in this dataset into a data frame called “diamonddata”.  
  
diamonddata = diamonds  
str(diamonddata) #R command to determine how many rows and columns are in this dataset.

## Classes 'tbl\_df', 'tbl' and 'data.frame': 53940 obs. of 10 variables:  
## $ carat : num 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...  
## $ cut : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...  
## $ color : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...  
## $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...  
## $ depth : num 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...  
## $ table : num 55 61 65 58 58 57 57 55 61 61 ...  
## $ price : int 326 326 327 334 335 336 336 337 337 338 ...  
## $ x : num 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...  
## $ y : num 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...  
## $ z : num 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...

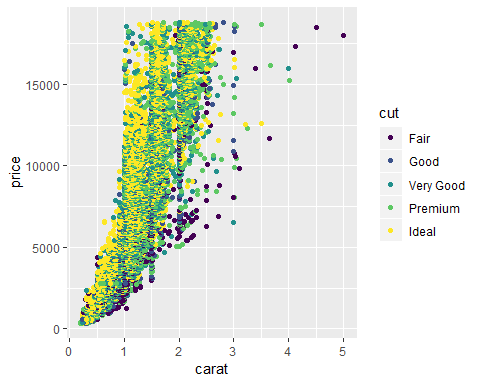
## Task 3

#Using ggplot, create a scatterplot of caret (x axis) versus price (y axis). Brieﬂy describe the relationship between these two variables.  
library(ggplot2)  
ggplot(data = diamonddata) +  
 geom\_point(mapping= aes(x=carat, y=price)) # the larger the carat, the higher the price



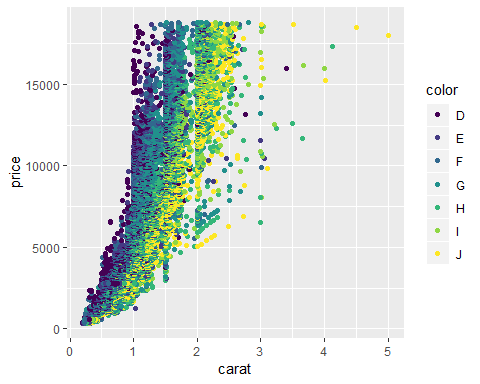
## Task 4

#Repeat Task 3, but in this plot color the scatterplot points by the “cut” variable. Brieﬂy describe the relationship between these three variables (carat, price, and cut).  
ggplot(data = diamonddata) +  
 geom\_point(mapping= aes(x=carat, y=price, color=cut)) #the size impacts the price and the best cuts can significantly vary in price



## Task 5

#Repeat Task 4, but in this plot facet by “color”. Brieﬂy describe the relationship between the four variables (carat, price, cut, and color).  
ggplot(data = diamonddata) +  
 geom\_point(mapping= aes(x=carat, y=price, color=color))#Color does not impact the price. Colors I and J are associated with larger carat sizes.



## Task 6

#Use the “readr” package (part of the tidyverse) to read-in the “InventoryData.csv” ﬁle as a data frame called “inventory”. Examine the structure and summary of the data frame.  
library(readr)  
InventoryData <- 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(InventoryData)  
str(InventoryData)#examine structure

## Classes 'tbl\_df', 'tbl' and 'data.frame': 13561 obs. of 6 variables:  
## $ Item SKU : chr "0100" "0100" "0100" "0100" ...  
## $ Store : chr "003480" "01611" "01611" "020109" ...  
## $ Supplier : chr "A" "B" "D" "B" ...  
## $ Cost per Unit ($): num 125.32 115.12 53.61 2.26 60.51 ...  
## $ On Hand : int 159 40 174 176 74 48 6 129 82 17 ...  
## $ Annual Demand : int 1693 351 1691 1559 733 496 58 1106 771 172 ...  
## - attr(\*, "spec")=List of 2  
## ..$ cols :List of 6  
## .. ..$ Item SKU : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_character" "collector"  
## .. ..$ Store : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_character" "collector"  
## .. ..$ Supplier : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_character" "collector"  
## .. ..$ Cost per Unit ($): list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ On Hand : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ Annual Demand : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## ..$ default: list()  
## .. ..- attr(\*, "class")= chr "collector\_guess" "collector"  
## ..- attr(\*, "class")= chr "col\_spec"

glimpse(InventoryData)

## Observations: 13,561  
## Variables: 6  
## $ `Item SKU` <chr> "0100", "0100", "0100", "0100", "0100", "0...  
## $ Store <chr> "003480", "01611", "01611", "020109", "020...  
## $ Supplier <chr> "A", "B", "D", "B", "C", "D", "B", "B", "C...  
## $ `Cost per Unit ($)` <dbl> 125.32, 115.12, 53.61, 2.26, 60.51, 53.72,...  
## $ `On Hand` <int> 159, 40, 174, 176, 74, 48, 6, 129, 82, 17,...  
## $ `Annual Demand` <int> 1693, 351, 1691, 1559, 733, 496, 58, 1106,...

## Task 7

#Use a ﬁlter to create a new data frame called “inventoryA” containing only inventory from Supplier A. How many rows are in this new data frame? 3695 rows  
  
inventoryA <- InventoryData %>% filter(Supplier == "A")  
inventoryA#display data frame

## # A tibble: 3,695 x 6  
## `Item SKU` Store Supplier `Cost per Unit ($)` `On Hand` `Annual Demand`  
## <chr> <chr> <chr> <dbl> <int> <int>  
## 1 0100 003480 A 125. 159 1693  
## 2 011 020109 A 12.3 173 1695  
## 3 0113 031779 A 208. 166 1496  
## 4 0113 080212 A 187. 157 1654  
## 5 0122 003480 A 68.5 34 290  
## 6 0122 020109 A 120. 77 680  
## 7 0122 031779 A 56.6 133 1239  
## 8 013 003480 A 19.1 28 277  
## 9 013 020109 A 22.7 103 962  
## 10 013 031779 A 1.13 29 297  
## # ... with 3,685 more rows

## Task 8

#What does the line of code shown below do? Note the use of the backtick character (on the tilde key on your computer’s keyboard) to delineate the variable names with spaces in them. inventoryA = mutate(inventoryA, OnHandRatio = `On Hand` / `Annual Demand`) It created a new column named OnHandRatio that calculated the number On Hand divided by the number Annual Demand.  
  
inventoryA = mutate(inventoryA, OnHandRatio = `On Hand` / `Annual Demand`)

## Task 9

# Create a new data frame called “avg\_cost” that contains the average “Cost per Unit ($)” by each “Item SKU” (let this quantity be in a variable called “SKUAvgCost. Hint: Recall the summarize and group\_by functions and the use of the backtick character from Task 8. Your data frame should have only two columns:”Item SKU" and “SKUAvgCost”  
  
avg\_cost = inventoryA %>% group\_by(`Item SKU`) %>% summarise(SKUAvgCost=mean(`Cost per Unit ($)`))  
avg\_cost#display data frame

## # A tibble: 1,720 x 2  
## `Item SKU` SKUAvgCost  
## <chr> <dbl>  
## 1 0100 125.   
## 2 011 12.3   
## 3 0113 197.   
## 4 0122 81.7   
## 5 013 14.3   
## 6 0133 203.   
## 7 0137 165.   
## 8 014 9.22  
## 9 0151 75.7   
## 10 0152 212.   
## # ... with 1,710 more rows

## Task 10

#Given your previous course experience with R/RStudio, what topics/concepts did you ﬁnd to be most challenging? I found/find debugging the code to be challenging. MIS503, the course I took prior to this one, was my first experience with programming languages and R. I find R to be pretty user friendly though.