Assignment3

Jordan Whitaker

5/17/2020

# Task 1  
install.packages("tidyverse", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Jordan Whitaker/Documents/R/win-library/3.6'  
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Jordan Whitaker\AppData\Local\Temp\RtmpkXbc3T\downloaded\_packages

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.3

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

## v ggplot2 3.3.0 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.5  
## v tidyr 1.0.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'tidyr' was built under R version 3.6.3

## Warning: package 'readr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## Warning: package 'stringr' was built under R version 3.6.3

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

# Task 2   
diamonddata = diamonds  
  
ncol(diamonddata) #10

## [1] 10

nrow(diamonddata) #53940

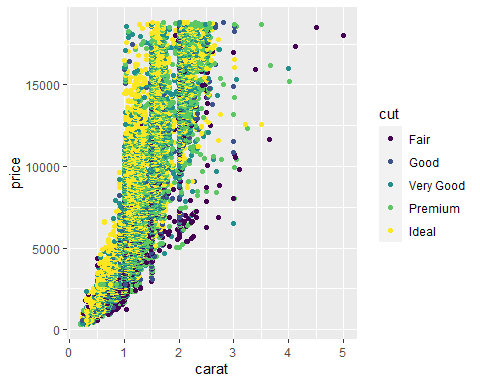
## [1] 53940

# Task 3  
  
ggplot(data = diamonddata, aes(x=carat, y=price)) +  
 geom\_point()



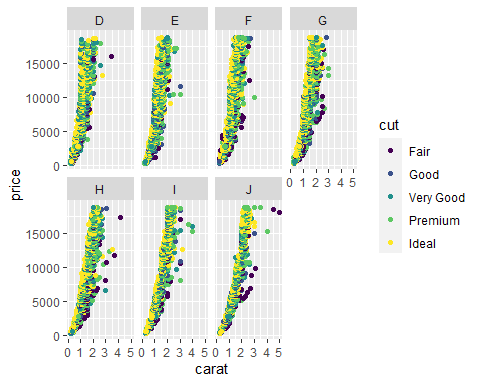
# The relationship shows that price increases as carat increases.

# Task 4   
  
ggplot(data = diamonddata, aes(x=carat, y=price, color = cut)) +  
 geom\_point()



# The relationship between carat, price, and cut shows that the quality of the cut makes a big difference in the overall price. Carat size becomes less important and cut increases in signifigance.

# Task 5  
  
ggplot(data = diamonddata, aes(x=carat, y=price, color = cut)) +  
 geom\_point() +  
 facet\_wrap( ~ color, nrow = 2)



# When adding in the color variable, for the 'colorless' grades their price increase is more steep compared to the color grades that may have more inclusions or flaws.

# Task 6  
#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\_double(),  
## `Annual Demand` = col\_double()  
## )

summary(inventory)

## Item SKU Store Supplier Cost per Unit ($)  
## Length:13561 Length:13561 Length:13561 Min. : 0.0   
## Class :character Class :character Class :character 1st Qu.: 137.0   
## Mode :character Mode :character Mode :character Median : 377.5   
## Mean : 504.4   
## 3rd Qu.: 775.5   
## Max. :1982.3   
## On Hand Annual Demand   
## Min. : 0.0 Min. : 0.0   
## 1st Qu.: 50.0 1st Qu.: 483.0   
## Median :101.0 Median : 965.0   
## Mean :100.5 Mean : 966.2   
## 3rd Qu.:151.0 3rd Qu.:1448.0   
## Max. :200.0 Max. :2150.0

structure(inventory)

## # A tibble: 13,561 x 6  
## `Item SKU` Store Supplier `Cost per Unit ($)` `On Hand` `Annual Demand`  
## <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 0100 003480 A 125. 159 1693  
## 2 0100 01611 B 115. 40 351  
## 3 0100 01611 D 53.6 174 1691  
## 4 0100 020109 B 2.26 176 1559  
## 5 0100 020109 C 60.5 74 733  
## 6 0100 020109 D 53.7 48 496  
## 7 0100 080212 B 107. 6 58  
## 8 011 003480 B 1.33 129 1106  
## 9 011 003480 C 12.9 82 771  
## 10 011 01611 C 5.16 17 172  
## # ... with 13,551 more rows

# Task 7  
  
inventoryA <- filter(inventory, inventory$Supplier == 'A')  
nrow(inventoryA) # 3695

## [1] 3695

# Task 8  
  
inventoryA = mutate(inventoryA, OnHandRatio = `On Hand` / `Annual Demand`)  
  
#This line of code uses the inventoryA dataframe and uses a function called mutate to create a new column in the dataframe that represents the quotient of columns 'On Hand' and 'Annual Demand'.

# Task 9  
  
avg\_cost <- inventory %>%  
 group\_by(`Item SKU`)  
   
avg\_cost <- summarize(avg\_cost, SKUAvgCost = mean(`Cost per Unit ($)`))

# Task 10  
  
# This was a great refresher and helped re-enforce syntax of the tidyverse library.