CM515 Assignment 5

Teagan Rockwood

Instructions

The aim of this assignment is to provide you with an opportunity to sharpen your skills in using ggplot(). While some of the plots you create may resemble those we covered in class, it is essential that they are original.

Rules for the Assignment:

1.All data sets except iris are fair game. That includes past data sets used in the class, sets built into R, your own data, and even online data.

2. All graphs must include axis labels, plot title, a theme of your choice, and a brief description/interpretation of the plot (2-3 sentences).

It's worth noting that ggplot() is a widely-used tool, and there are numerous online resources available for you to explore. We strongly recommend that you take advantage of these resources to deepen your understanding of ggplot().

Knit the document into a PDF and submit it to Canvas by 02/28/2024 at 11:59 pm.

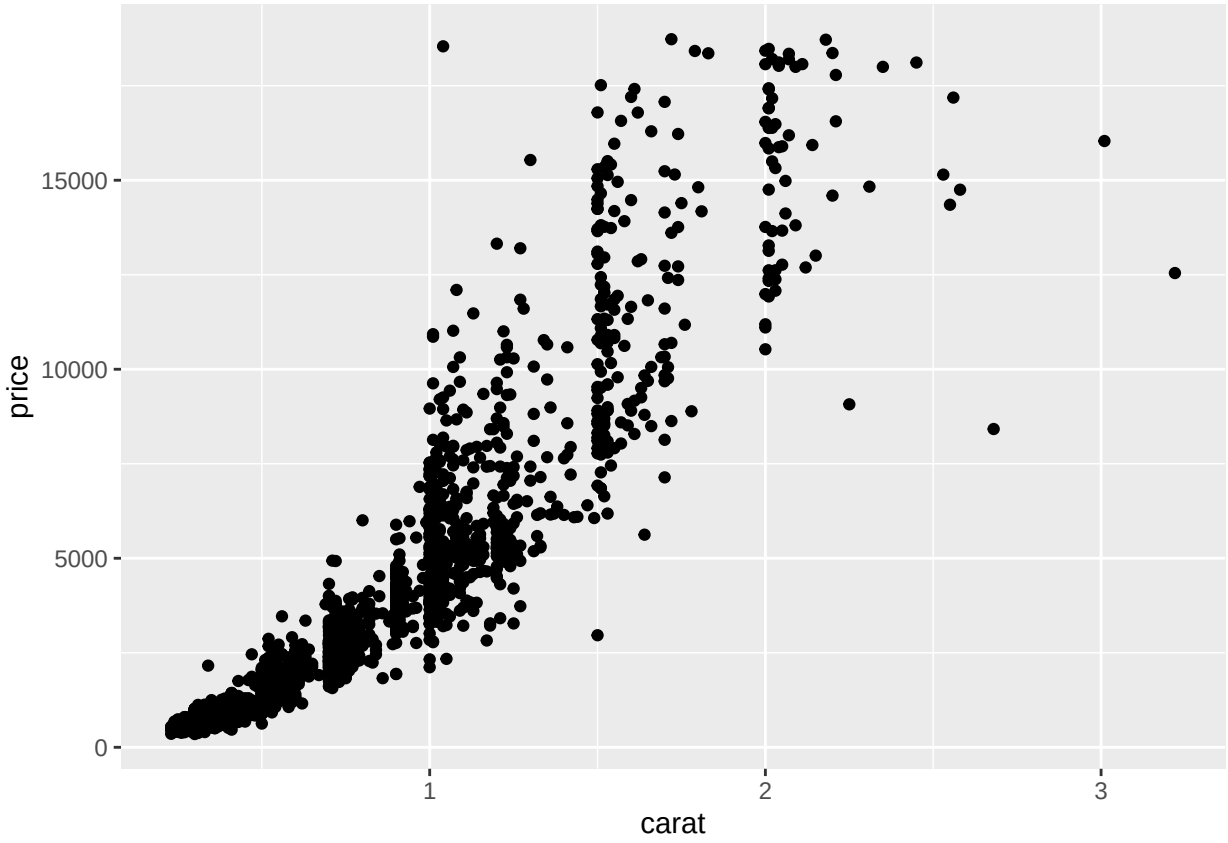
Load Packages and Data

---

1. Make a Scatter Plot With Customized Point Size and Transparency (3pts)

Description and Interpretation: The diamonds dataset is nice to work with for this assignment since it has a mix of continuous and discrete variables. It is, however, enormous and trimming it to only 2000 samples makes it look nicer in charts and still can give a representative view of the dataset. Particularly when you use random sampling. In order to ensure a nice spread that isn't as noisy, I plotted two continuous variables (carat

and price) and the resulting scatterplot shows how the size (carat) of a diamond has a positive relationship

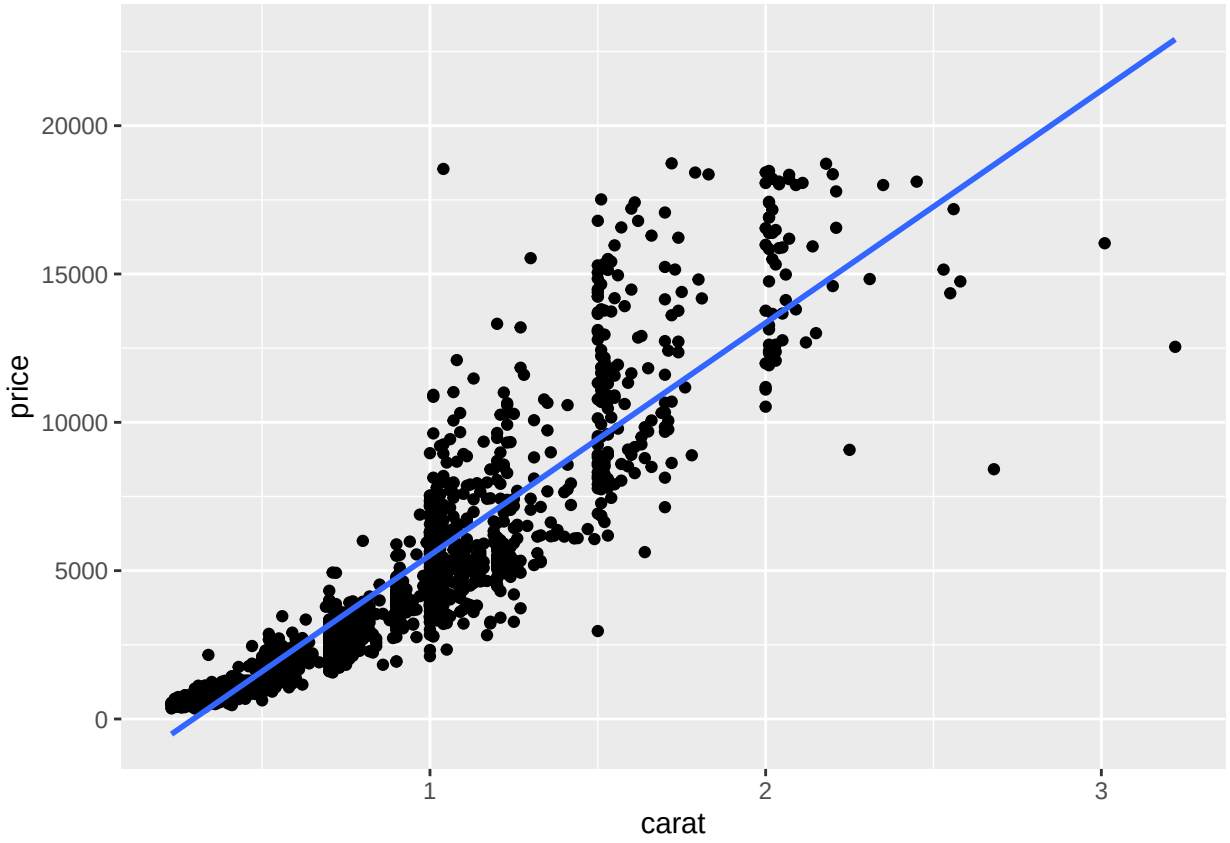


with the price.

---

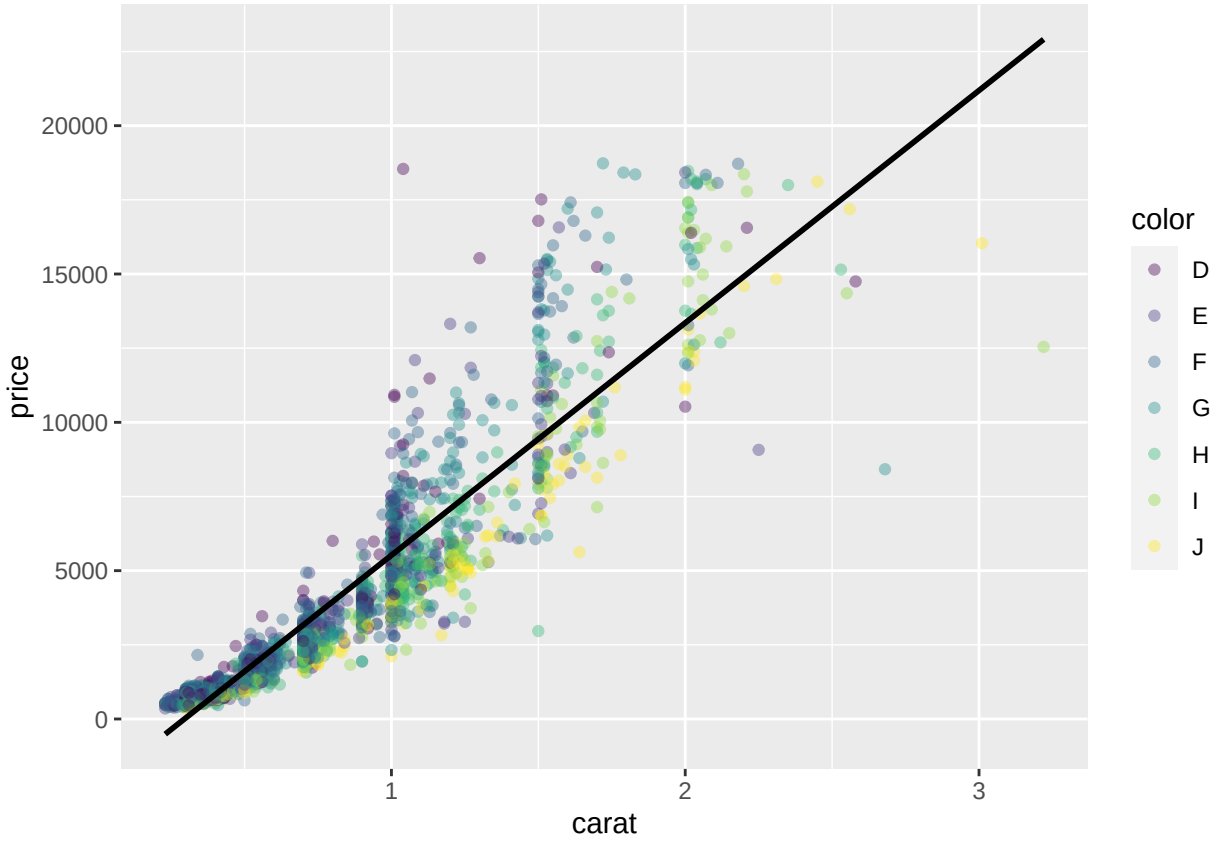
2. Fit a Line Through Your Scatter Plot From 1. (3pts)

Description and Interpretation: Same plot as before but using geom_smooth with the method "lm" to draw a line of best fit through the data.



3. Set the transparency, shape, and color of your graph from 1 based on expressions. Scale the color. (3pts)

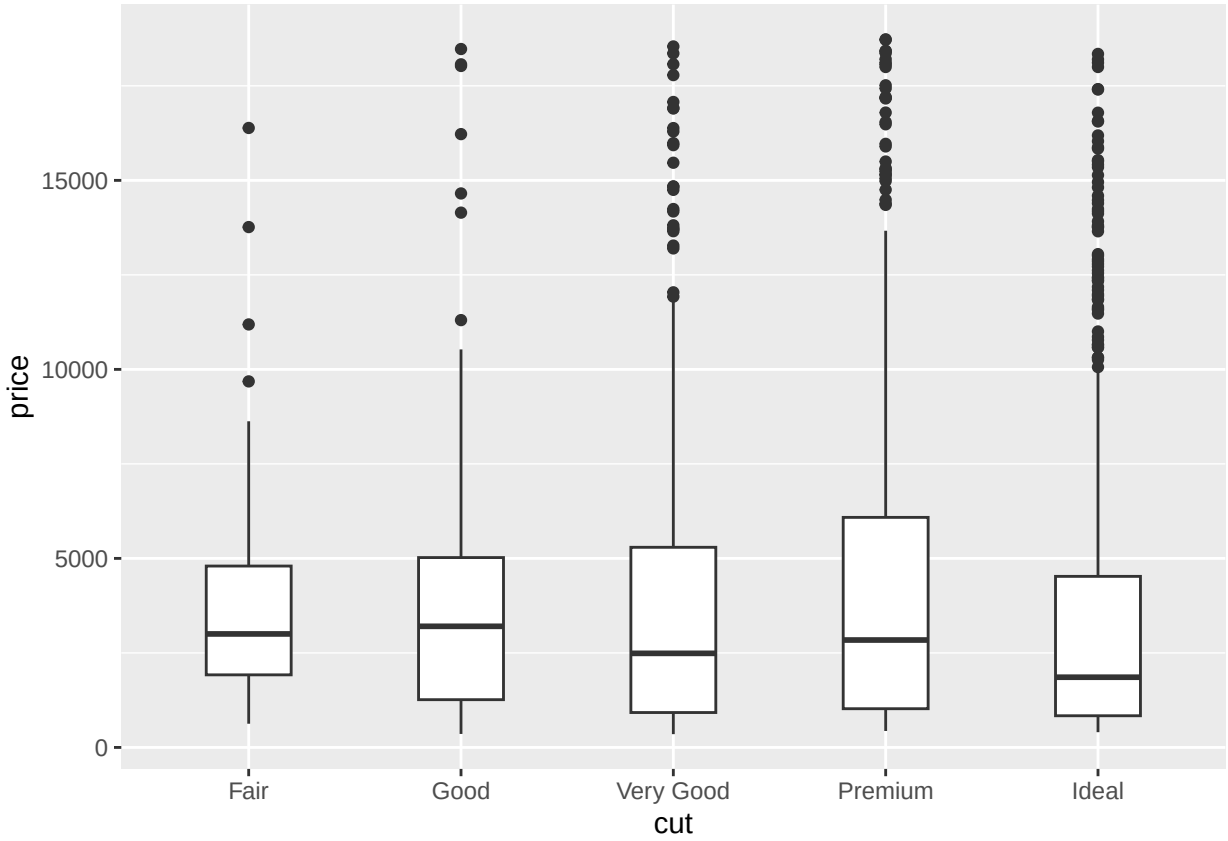
Description and Interpretation: Changing the alpha to a lower value can help make the datapoints easier to see (particularly overlaps). Having the color of the points change based on the color quality color of the diamonds adds another dimension that makes the graph more meaningful. I think circles still look the nicest so I just kept them. You can see how the quality of the color affects the price now, where D,E and other colors are more pricey even at lower carats.



---

4. Make a Boxplot With Customized Boxplot Width (3pts)

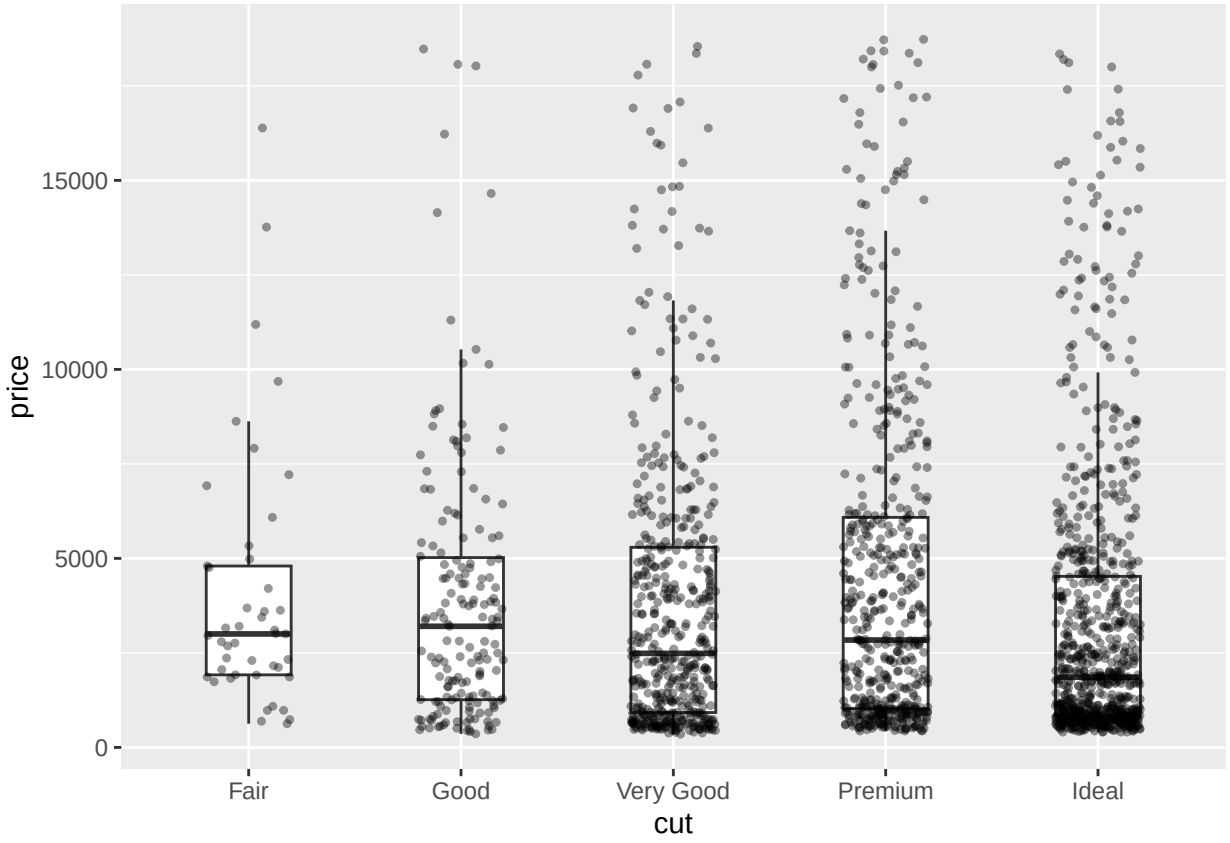
Description and Interpretation: Boxplots are useful for looking at a discrete variable on the x-axis and a continuous variable on the y-axis. So I changed out carat for cut. The box plot shows that the cut of the diamond is very loosely related to the price.



---

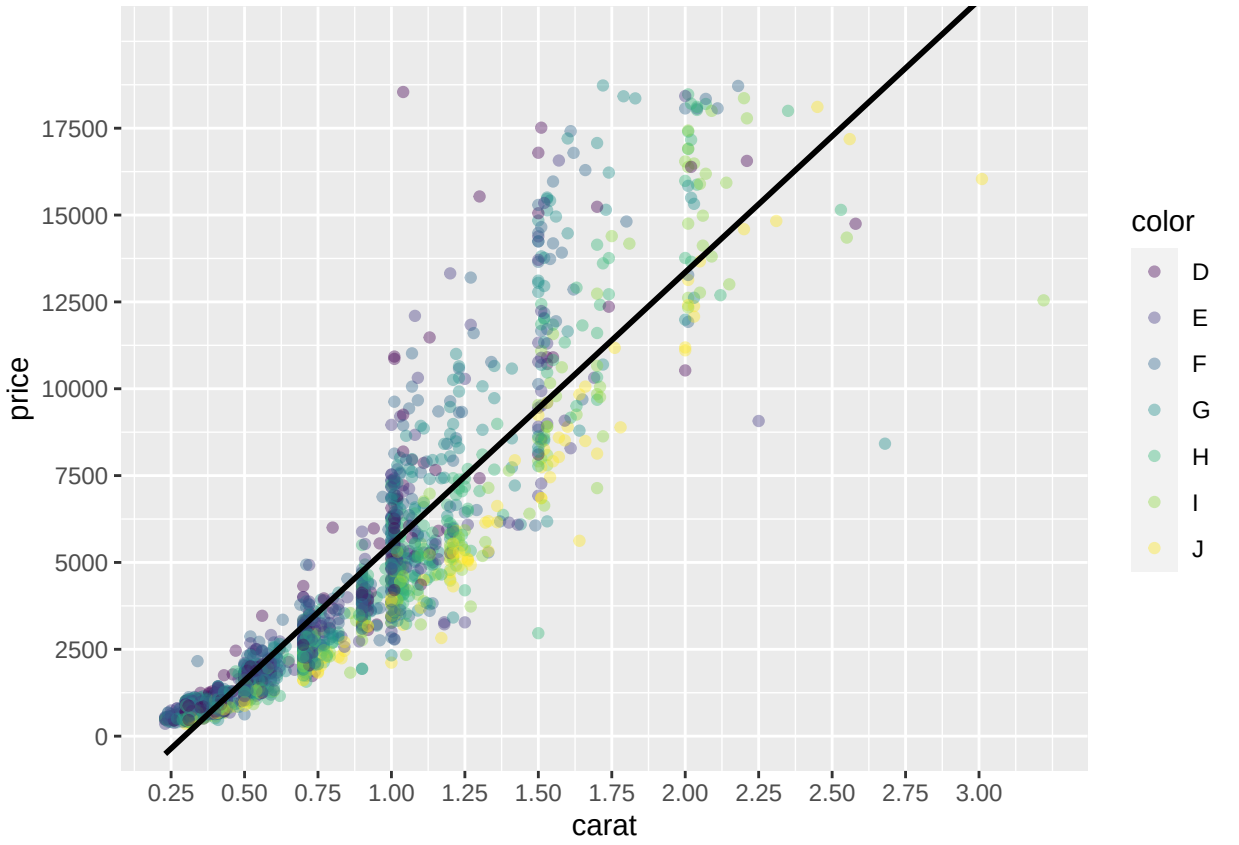
5. Overlay the Individual Points Over Your Box Plot From 3. and Adjust the Point Size and Transparency as Needed (3pts)

Description and Interpretation: Adding the individual points in the box plot helps give a better view of the spread of the data as a whole. Because of the amount of data points in this data set, it makes more sense to do a jitter plot that helps spread the points out. From there I also removed the outlier points from the boxplot (outlier.shape = NA) so the only points come from the jitter. Again changing the alpha, size, and width of the jitter plot can help it look easier to read. In particular, the width being the same as the boxplot is aesthetically pleasing.



6. Make a plot and modify the scale in a beneficial way using scale_x_continuous(), scale_y_continuous(), and coordinate_cartesian(). Explain how your modifications to the scales improved the plot. (4pts)

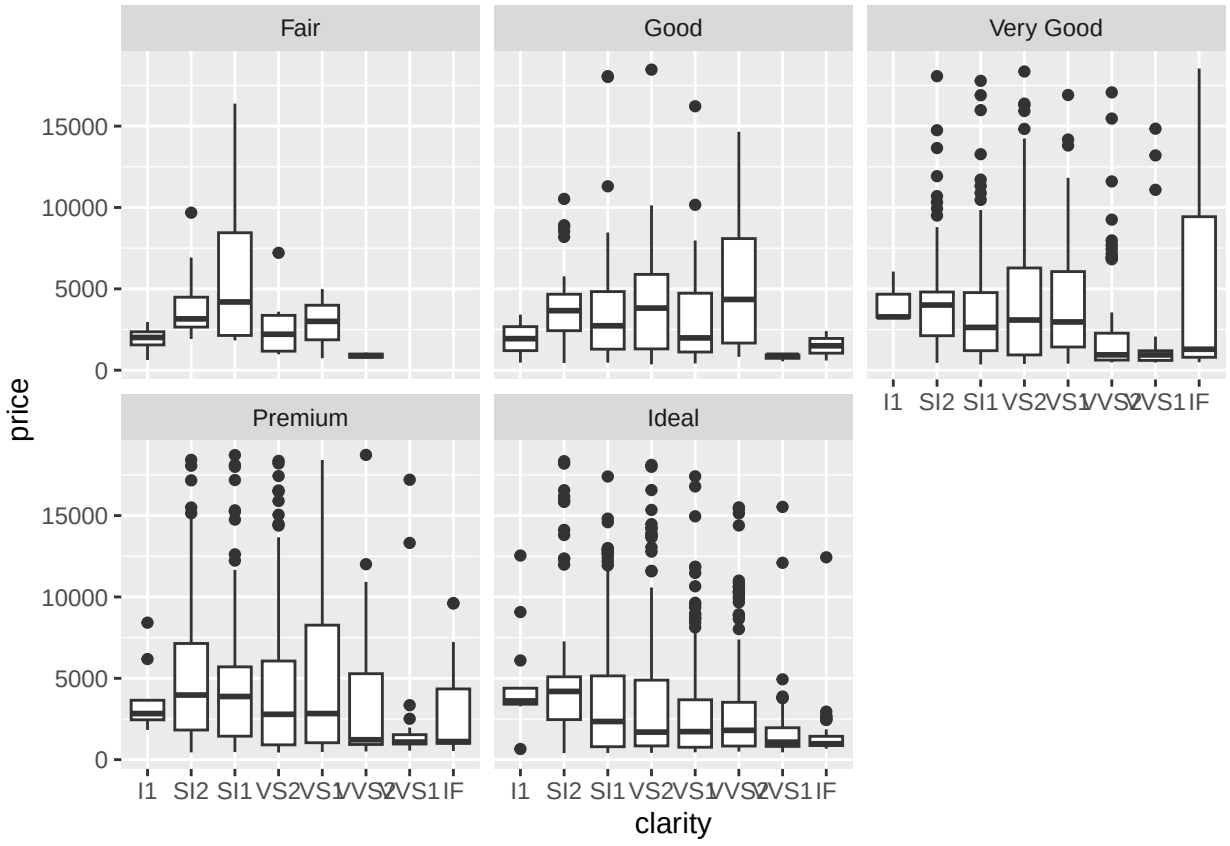
Description and Interpretation: Changed the x-scale to start at 0 and end at the max of the carat value and step by 0.25. This makes the scale more granular which helps what carat a particular point falls at. Same deal with the y-scale. The ylim in coord_cartesian helps trim the top of the graph to make it look nicer, since the scale_y_continous makes it look kind of funky.



---

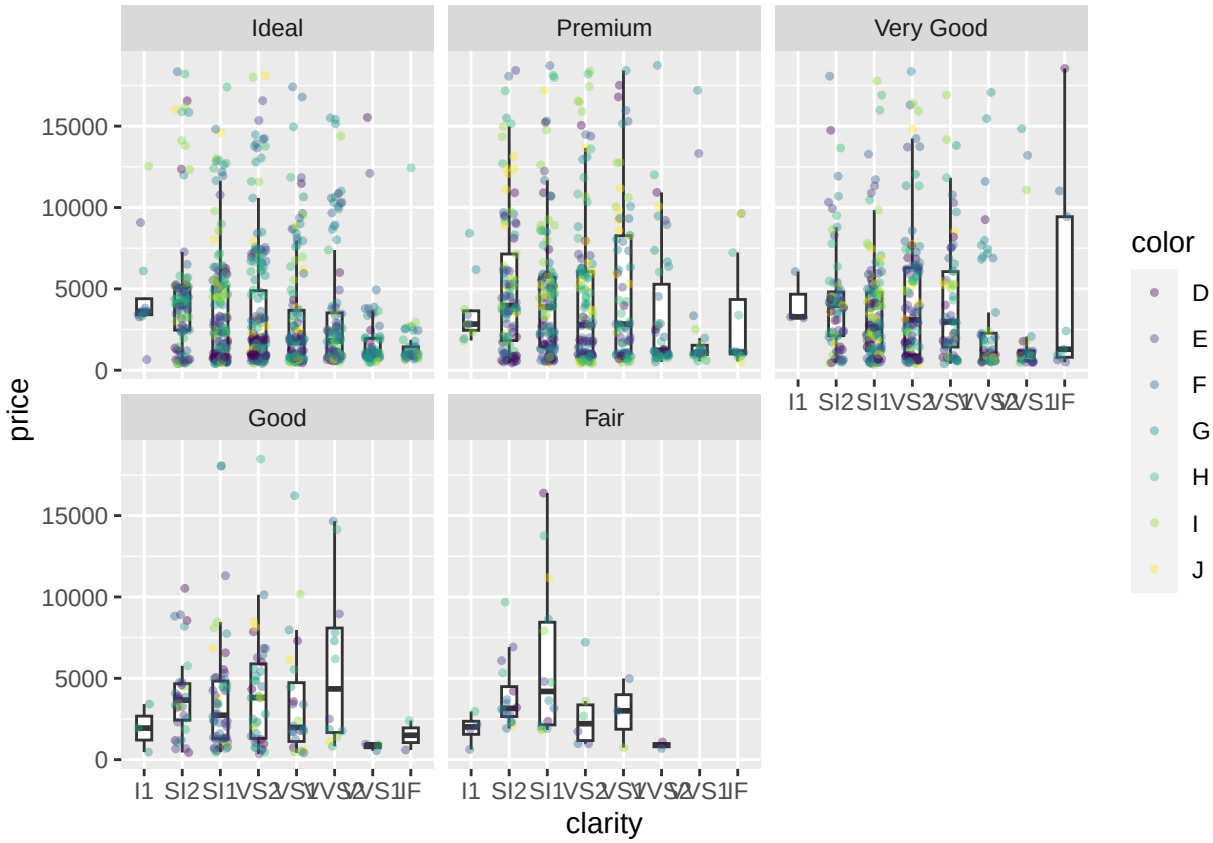
7. Make a New (don't use any plots from 1-4) Plot with facet_wrap() (4pts)

Description and Interpretation: Making several box plots where the clarity is on the x-axis and the price is on the y-axis. Then facet-wrapping by the cut to make 5 different graphs. The main thing that sticks out to me about the data is that there aren't really any Internally flawless (IF) diamonds in the Fair cut quality.



8. Using the plot from 5, scale the colors and reorder your facet to reverse order. (4pts)

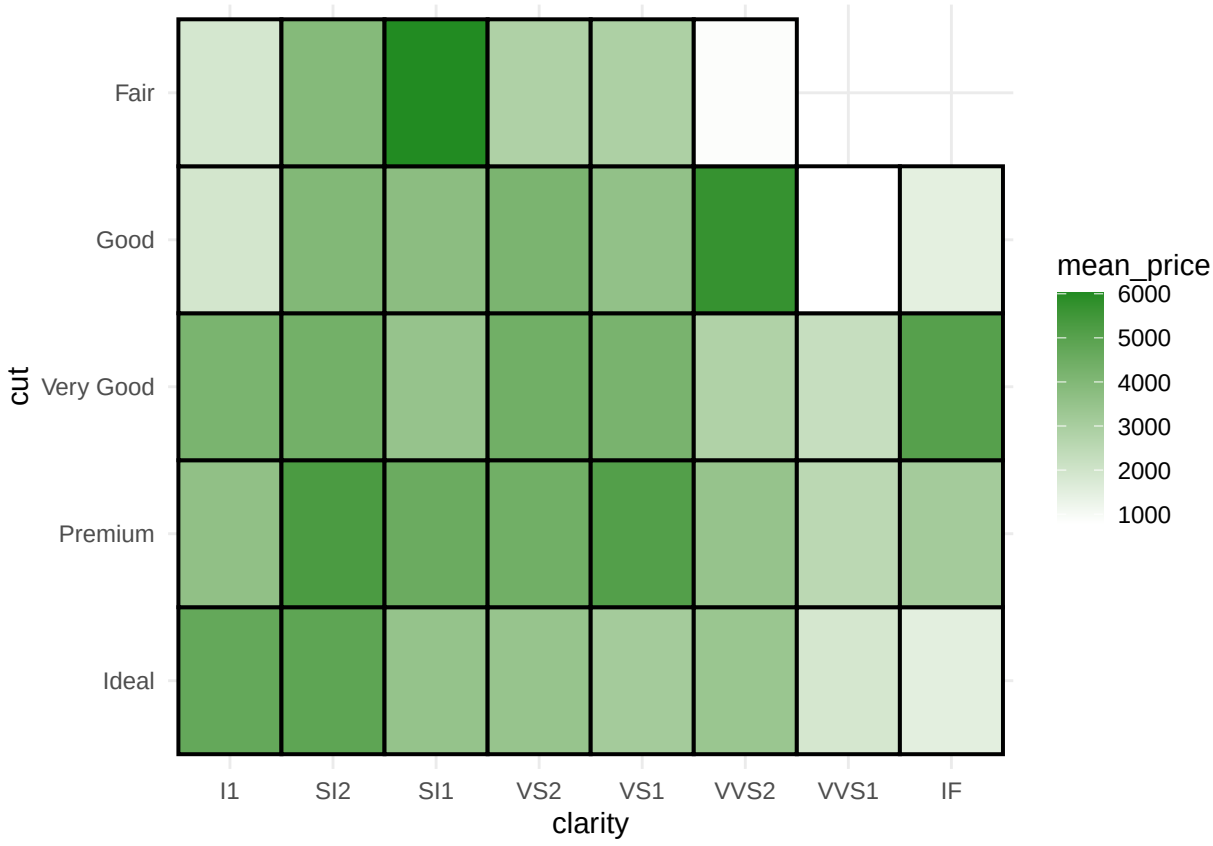
Description and Interpretation: I reordered the factor outside of the ggplot and then had it facet_wrap again by cut. I once again scaled the color by the diamond color.



---

9. Make a Plot Using a Geometry That Was Not Covered in Class (4pts)

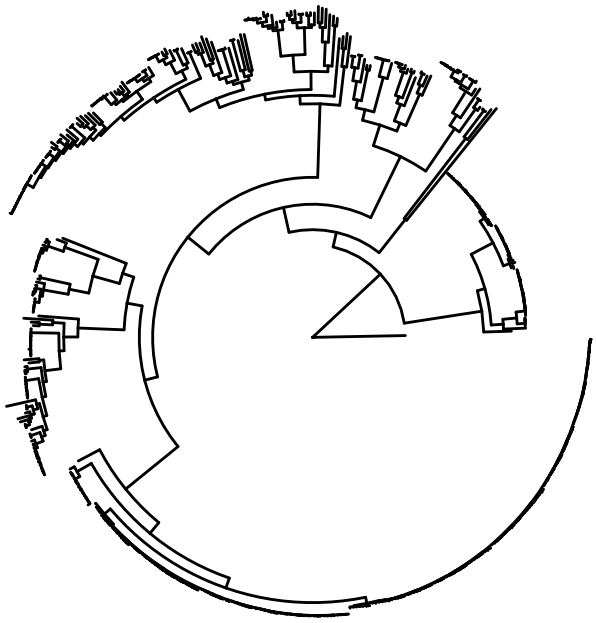
Description and Interpretation: Heatmap where the clarity and cut are grouped and then the average price for each group is found and used as the fill for the tiles. Changed the gradient and added a black border for aesthetics

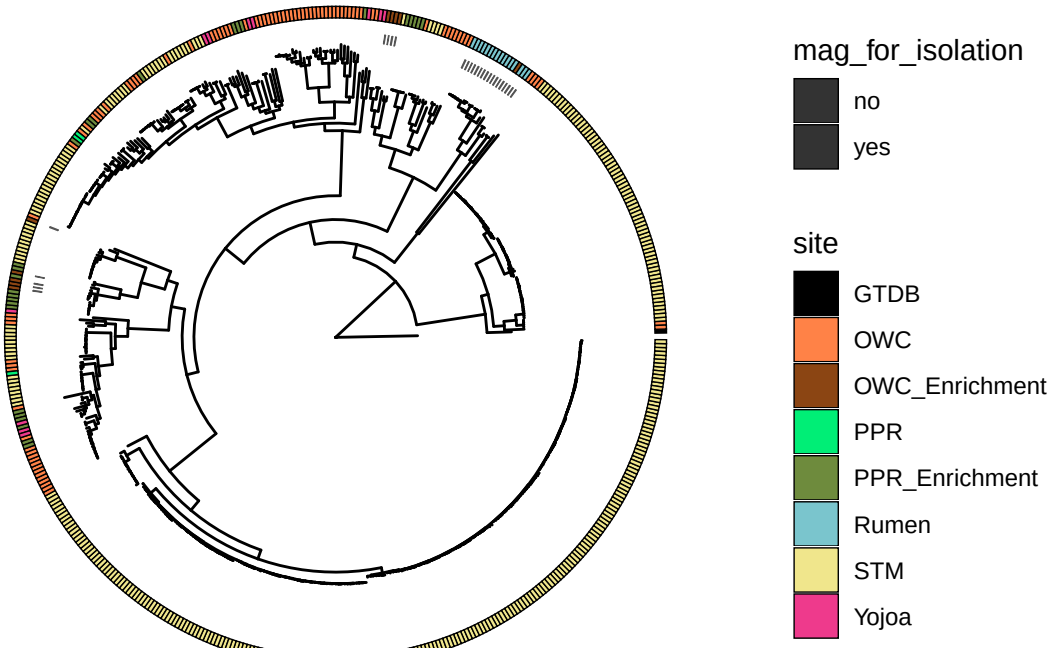


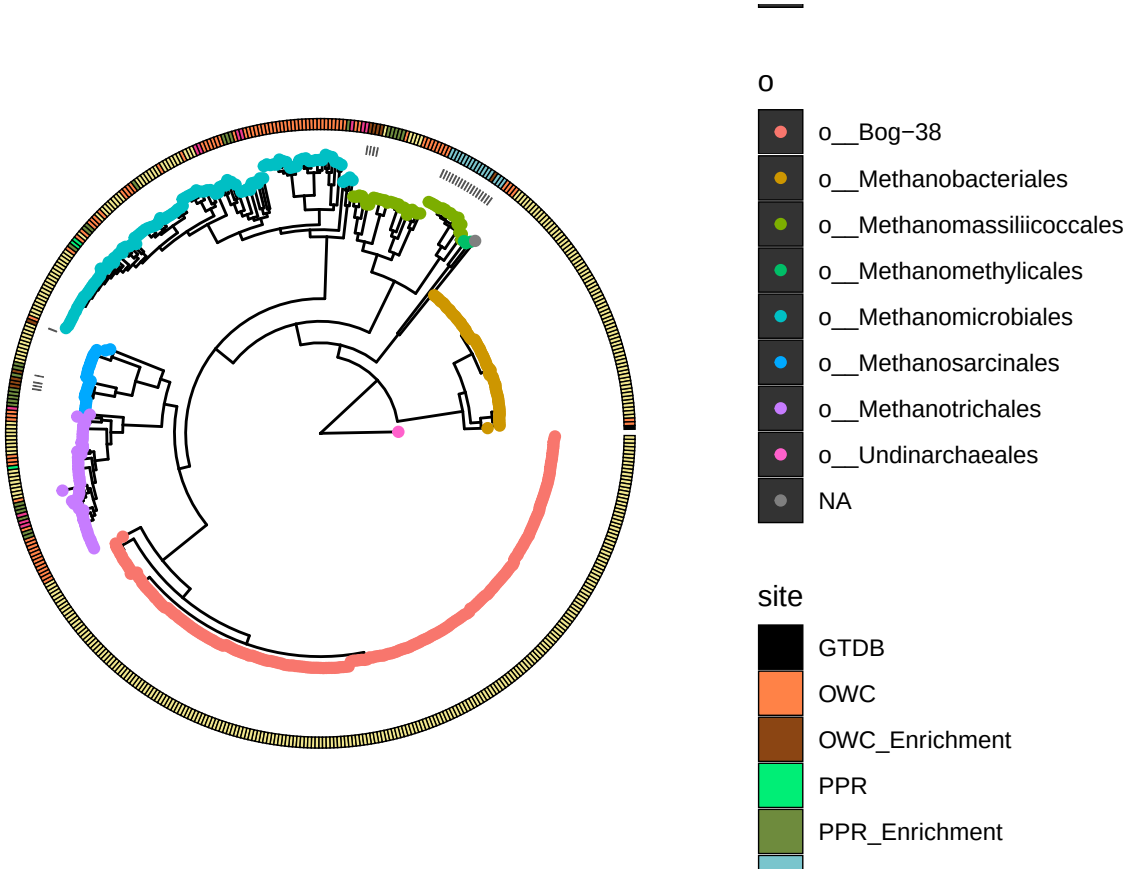
10. Make a Plot Using a ggplot Extension That Was Not Covered in Class. Explain what the extension does. (4pts)

Description and Interpretation: ggtree and ggtreeExtra allows for generation of phylogenetic trees using ggplot. This is using data from my current rotation in the Wrighton lab and generates a phylogenetic tree of methanogenic MAGs.

```
##
## Phylogenetic tree with 523 tips and 522 internal nodes.
##
## Tip labels:
##   GCA_002495465.1_ASM249546v1_genomic, OWC_0005, PLTK01, 20160700_E25_7, 20130700_C2M_14, PLKI01, ..
## Node labels:
##   d__Archaea, 0.99, '1.0:p__Methanobacteriota; c__Methanobacteria; o__Methanobacteriales; f__Methanol
##
## Rooted; includes branch lengths.
```




mag_for_isolation
no
yes
site
GTDB
OWC
OWC_Enrichment
PPR
PPR_Enrichment
Rumen
STM
Yojoa



Appendix

```
knitr::opts_chunk$set(echo = F, message = F, warning = F)
####
#00#
####
library(knitr)
library(tidyverse)

####
#01#
####

#Randomly sample 2000 rows to get a decent representative sample, but not overwhelming amount of data
diamonds_trim <- diamonds %>% sample_n(2000, replace = FALSE)
#Scatterplot of the trimmed dataframe comparing the carat to the price
ggplot(diamonds_trim) +
  geom_point(aes(x= carat, y=price))

####
#02#
```

```
####
#your code here
#Same scatter plot as above, but geom_smooth adds a line of best fit with method = "lm"
ggplot(diamonds_trim, aes(x= carat, y=price)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)

####
#03#
####
#Colors the points according to the diamond color (obviously not equivalent colors)
#Set the transparency to 0.4, makes it look nicer
  ggplot(diamonds_trim) +
  geom_point( aes(x= carat, y=price, color = color), alpha = 0.4, shape = "circle")+
  geom_smooth(aes(x= carat, y=price), method = "lm", se = FALSE, color = "black")

####
#04#
####
#Plot the cut quality of the diamond set the width to 0.4
ggplot(diamonds_trim)+
  geom_boxplot(aes(x=cut, y=price), width = 0.4)
####
#05#
####
#your code here
ggplot(diamonds_trim)+
  geom_boxplot(aes(x=cut, y=price), width = 0.4, outlier.shape = NA)+
  geom_jitter(aes(x=cut, y =price), alpha = 0.4, size = 0.9, width = 0.2)

####
#06#
####
#your code here
ggplot(diamonds_trim) +
  geom_point( aes(x= carat, y=price, color = color), alpha = 0.4, shape = "circle")+
  geom_smooth(aes(x= carat, y=price), method = "lm", se = FALSE, color = "black")+
  scale_x_continuous(breaks= seq(min(0),max(diamonds_trim$carat), by = 0.25))+
  scale_y_continuous(breaks= seq(min(0),max(diamonds_trim$price), by = 2500))+
  coord_cartesian(ylim= c(0,20000))

#07#
####
#your code here

ggplot(diamonds_trim)+
  geom_boxplot(aes(x=clarity, y=price)) +
  facet_wrap(~cut)

#08#
####
#your code here
diamonds_trim$cut <- factor(diamonds_trim$cut, levels=c("Ideal", "Premium", "Very Good", "Good", "Fair"
```

```r
ggplot(diamonds_trim)+
  geom_boxplot(aes(x=clarity, y=price), width = 0.4, outlier.shape = NA)+
  geom_jitter(aes(x=clarity, y =price, color = color), alpha = 0.4, size = 0.9, width = 0.2)+
  facet_wrap(~cut)

####
#09#
####
#your code here
diamonds_mean <- diamonds_trim %>% group_by(clarity, cut) %>% summarise(mean_price = mean(price))
ggplot(diamonds_mean)+
  geom_tile(aes(x=clarity, y=cut,fill=mean_price), color = "black", size = 0.7)+
  scale_fill_gradient(low = "white", high = "forestgreen", na.value = "white")+
  theme_minimal()
####
#10#
####
#your code here

library(ape)
library(ggplot2)
library(ggtree)
library(ggtreeExtra)
library(readxl)
library(dplyr)
library(tidyr)
library(ggnewscale)
library(cowplot)
library(TDbook)
library(treeio)

# read in GTDB trees (from data_processing)
setwd("C:/Users/teaga/Downloads/Grad School/WrightonLab/Presentation/MAG_tree/teagan_MAG_tree" )
arc_tree = read.tree("gtdbtk.ar53.decorated.tree")
arc_tree

# read in taxonomy (from data_processing)
annotation = read.delim("classification_wOutgroup.txt",sep="\t",header = FALSE)
colnames(annotation)=c("MAG","tax")
annotation=annotation%>%separate(tax,into=c("d","p","c","o","f","g","s"),sep=";",remove=FALSE)

#read in data on linking MAG to source
site <- read.delim('site.txt')

#read in data on alling MAGs isolates
mag_for_isolation<- read.delim('isolate.txt')

# make archaeal tree
arc_dat=as.data.frame(arc_tree$tip.label)
colnames(arc_dat)=c("MAG")
arc_dat=left_join(arc_dat,annotation,by="MAG")

# undecorated tree - section 4.2.2 of https://yulab-smu.top/treedata-book/chapter4.html
```

```
mgen_tree <- ggtree(arc_tree, color="black", size=0.5, linetype="solid", layout="circular",  branch.leng
mgen_tree

#color palette for sites -
Sourcecol <- c("GTDB"="black", "OWC"="sienna1", "STM"="khaki", 'PPR'="springgreen2",
               "PPR_Enrichment"="darkolivegreen4", "OWC_Enrichment"="chocolate4",
               "Rumen"="cadetblue3", "Yojoa"="violetred2")
IsolateCol <- c("no"="white", "yes"="grey33")




#script with dummy data for pangenome analyses
arc=mgen_tree +
  geom_fruit(data=mag_for_isolation,geom=geom_tile,
             mapping=aes(y=MAG,x=0,fill=mag_for_isolation),width = 0.05,offset=0.03,show.legend=TRUE,col
  scale_fill_manual(values=IsolateCol)+
  new_scale_fill()+
  geom_fruit(data=site,geom=geom_tile,
                mapping=aes(y=MAG,x=0,fill=site),width = 0.05,offset=0.08,show.legend=TRUE,color = "bla
  scale_fill_manual(values=Sourcecol)

 arc


 #add taxonomy to tree

 arc <- arc %<+% annotation + new_scale_fill() +
   geom_tippoint(mapping=aes(color= o))
arc
```