HW5_IS457_33

Do not remove any of the comments. These are marked by # HW 5 - Due Monday, Oct 22, 2018 on moodle and hardcopy in class. (1). Please upload R script and report to Moodle with filename: HW5_IS457_YourCourseID. (2). Turn in hard copy of your report in class.

Class ID: 33

The grading is based on properties of good graph construction: 1.Data stand out 2.Facilitate comparison 3.Information rich 4.Vocabulary (in titles, axes labels, legend names etc) The grading will be strict, since there are many elements in each plot. The total points for each questions is 15, 10 pts for plotting and 5 for explanation. But there are two bonus questions in the end.

Note: for interpretation questions, you won't get any points only describing the plots. Use relevant technical terms (from lectures/slides) to EXPLAIN your findings/insights. e.g., for normal distribution, think about mean (center), sd(spread), skewness, outliers etc.

Unless we mentioned using external packages, stick with base R commands.

Part 1. Basic plots

Q1. Show the shape of a distribution.

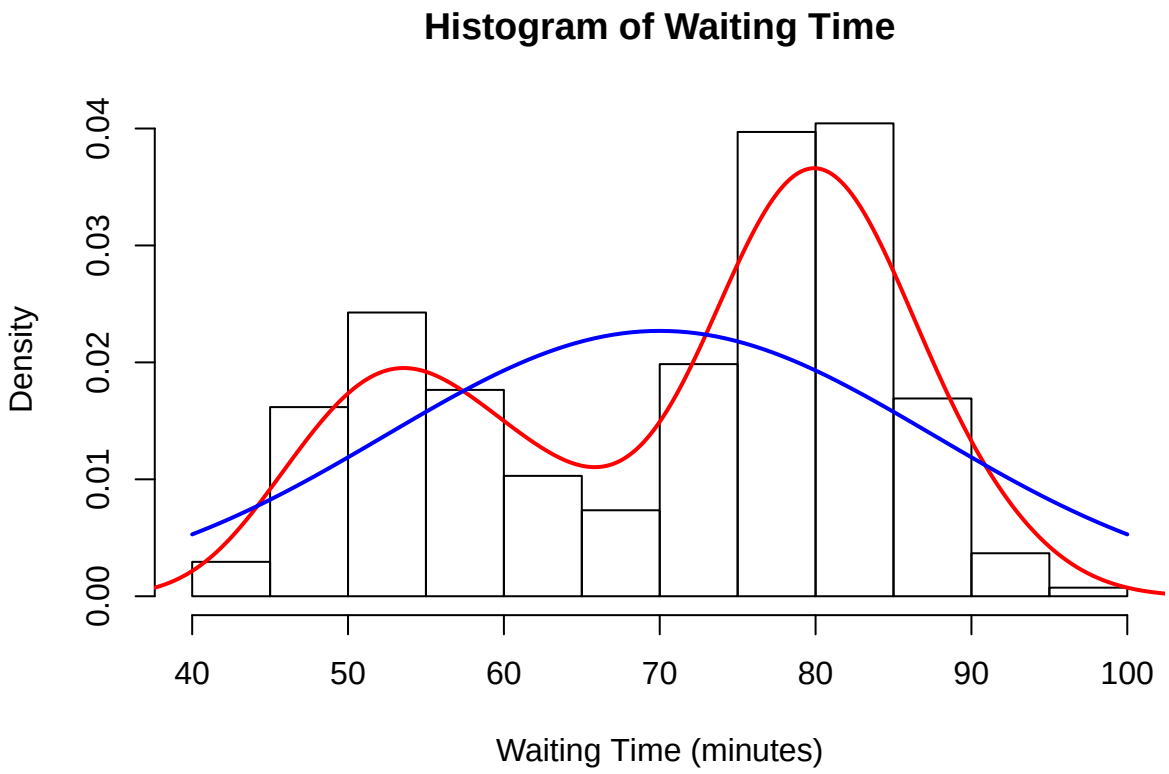
load the data set "faithful" (we've shown many times before how to load data in base R) 1), make a histogram that shows the distribution of variable "waiting". Hint: Adjust arguments of line() to make the line stand out. 3), add a normal distribution curve on the plot with mean and standard deviation of waiting. Hint: curve() may help, also make the newly added line stand out.

What do you see from the histogram? what about after adding the density curve?

and after imposing the normal curve?

Your code here

```
data(faithful)
hist(faithful$waiting, freq = FALSE, xlab = "Waiting Time (minutes)", main = "Histogram of Waiting Time
lines(density(faithful$waiting), lwd=2, col= "red")
x <- faithful$waiting
curve(dnorm(x, mean=mean(x), sd=sd(x)), add=TRUE, lwd=2, col="blue")
```



Your answer

Currently in this histogram is a main mode right around 80, a skew left with a second mode around 55. This shows higher density around these waiting times, with little time in between, however, I would not call this a data gap. We can tell that this data is most likely not a normal distribution.

Q2. Comparing distributions.

generate 3 distributiuons with (sample size, mean, sd) = (200,6,1), (100, 8,1) and (300,10,2). plot them on the same graph, one color each distribution, with rainbow colors. Hint: rgb() function. If your choice of color scheme is correct, overlapping areas should have different/darker colors. set.seed(457) # do not change

Comment on the shape of each distribution (effect of sample size, sd);

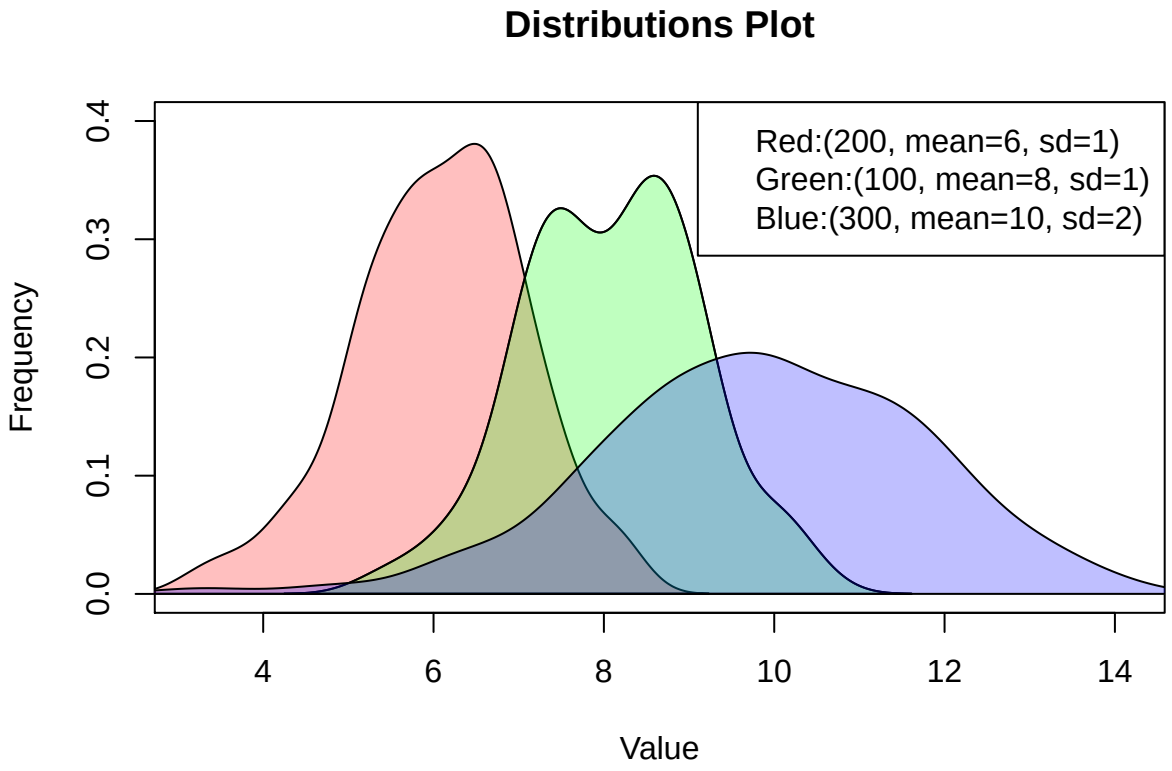
what does the final plot look like, and explain why.

why did the distritbution overlap? is there area overlapped by all three distritbution? if yes, why?

Your code

```
dist1 <- rnorm(200, mean=6, sd=1)
dist2 <- rnorm(100, mean=8, sd=1)
dist3 <- rnorm(300, mean=10, sd=2)

plot(density(dist2), xlim=c(min(dist1), max(dist3)), ylim=c(0, 0.4), main = "Distributions Plot", xlab=
polygon(density(dist1), col=rgb(1,0,0, alpha = .25))
polygon(density(dist2), col=rgb(0,1,0, alpha = .25))
polygon(density(dist3), col=rgb(0,0,1, alpha = .25))
legend("topright", legend=c("Red:(200, mean=6, sd=1)", "Green:(100, mean=8, sd=1)", "Blue:(300, mean=10
```



Your answer

Each distribution is generall centered around the mean value that we give to it. The standard deviation affects the 'width' of the distribution, as we can see by comparing the Blue distribution to the other two. The

number of samples taken affects the shape of the curve. The distributions overlap because they are normal distribution based around similar means and have standard deviations that will cause them to overlap. The three distributions overlap between about 3-9 on the xaxis, as the means and standard deviations place the lines in a place where they will share some overlap.

Q3. Boxplots to display multivariate relationships

We will use the mtcars data set. we've shown you how to use boxplot with one variable with multiple levels in base R command, now let's try with multiple variables using a function from package lattice, look up the manual. make a boxplot to display the variable, mpg, for different values of cylinders, conditioned on am and vs. hint: make sure you read the function documentation of what "condition on" means, your plot should consist of (num. of levels of am X num. of levels of vs) subplots.

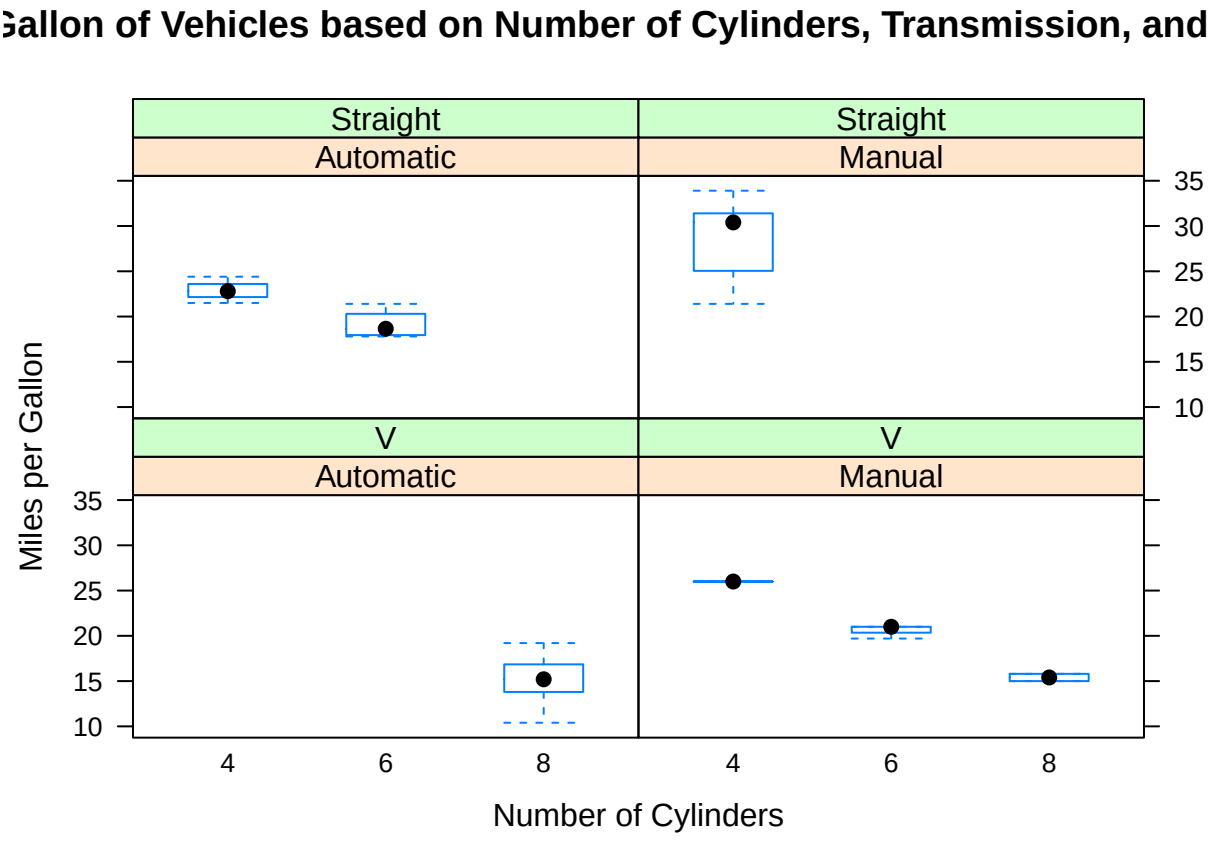
what information do you get from this plot? anything stand out? explain how/why this kind of plot can be useful.

Your code here

```
require(lattice)
```

```
## Loading required package: lattice
```

```
bwplot(mpg~factor(cyl) | factor(am, labels = c("Automatic","Manual"))+factor(vs, labels = c("V", "Straig
```



Your answer

This plot shows up some interesting information about miles per gallon of vehicles based on a number of factors. First, we can see which vehicles we have no data for, because no plot exists (i.e., there are no vehicles with a V engine, automatic transmission, and 4 or 6 cylinders). This plot makes it really easy to compare cars quickly, and see the locations of median and how quartiles compare across cars very quickly.

Q4. Stack bar plots with gradient colors

we will use the diamonds data set from ggplot2: first load the package, then load the data set as before. Using two categorical variables, cut and clarity to create a stacked bar chart. your y axis should be frequency. use the same color with darker shade indicating BETTER cut quality. hint: you can create a contingency table to help you plot

explain what you see from plot in the context of the data set.

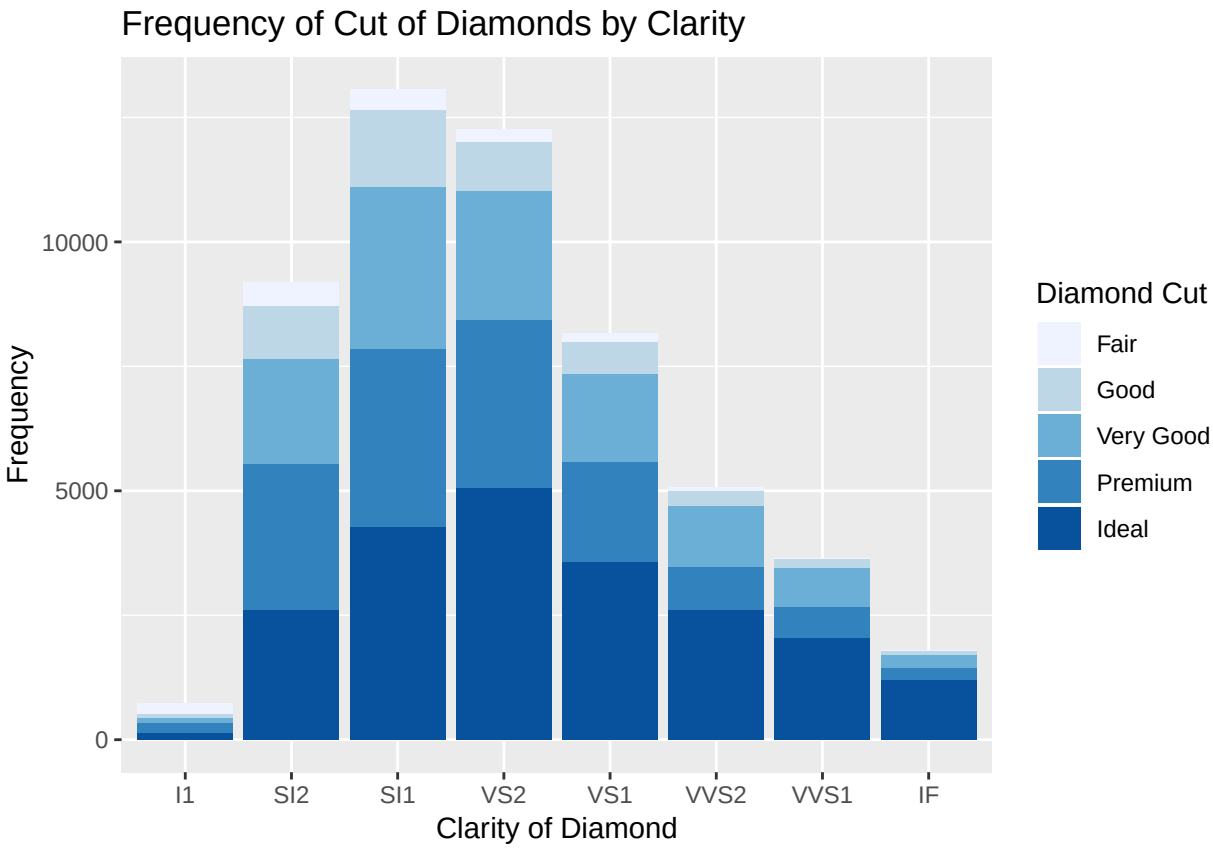
Your code

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
data(diamonds)
ggplot(diamonds, aes(x=clarity, fill=cut))+
  geom_bar()+
  xlab("Clarity of Diamond")+
  ylab("Frequency")+
  ggtitle("Frequency of Cut of Diamonds by Clarity")+
  scale_fill_brewer("Diamond Cut")
```



Your answer

This plot show us the total number of diamonds by clarity, and also breaks that up nicely so we can see how much of each clarity are in each category of cut diamonds. It makes it really easy to see and compare numbers of diamonds by cut across clarity values.

Part 2. Fancy plots with ggplot2 and ggmosaic.

also using diamonds data set for both questions.

Q1. Use ggplot to make a histogram for the carat variable and color it by (levels of) the cut Variable.

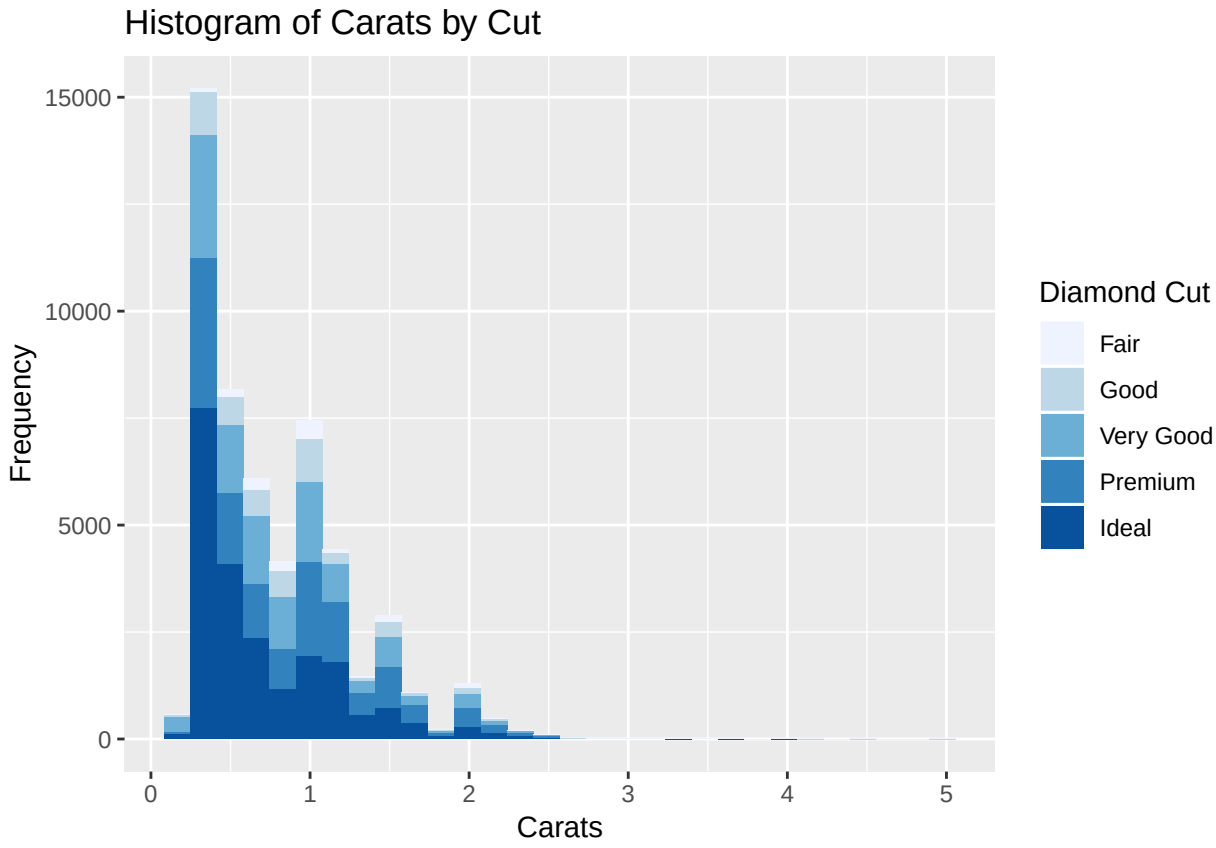
explain what you see from the plot in the context of the data set.

Your code here

```
ggplot(diamonds, aes(x=carat, fill=cut))+
  geom_histogram()+
  xlab("Carats")+
```

```
  ylab("Frequency")+
  ggtitle("Histogram of Carats by Cut")+
  scale_fill_brewer("Diamond Cut")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Your answer

This shows us that we have higher concentrations of diamonds that are between 0-1 carats. Our main mode appears to be somewhere in the .33 range, and the curve would come down with a very long tail as we go out to 5 carats. This means we have the most diamonds between 0-1 carats, and this trails off as we continue up to 5 carats. There are very few diamonds comparatively with more than 2 carats, and they are difficult to observe in this plot due to the size. It may be ideal to breakt this plot up to see a histogram of just diamonds with 2 or more carats.

Q2. Make a mosaic plot by cut and clarity variables.

To create a mosaic plot with ggplot, you will need the ggmosaic package.

explain how to interpret the plot, and what you see in the context of the data set.

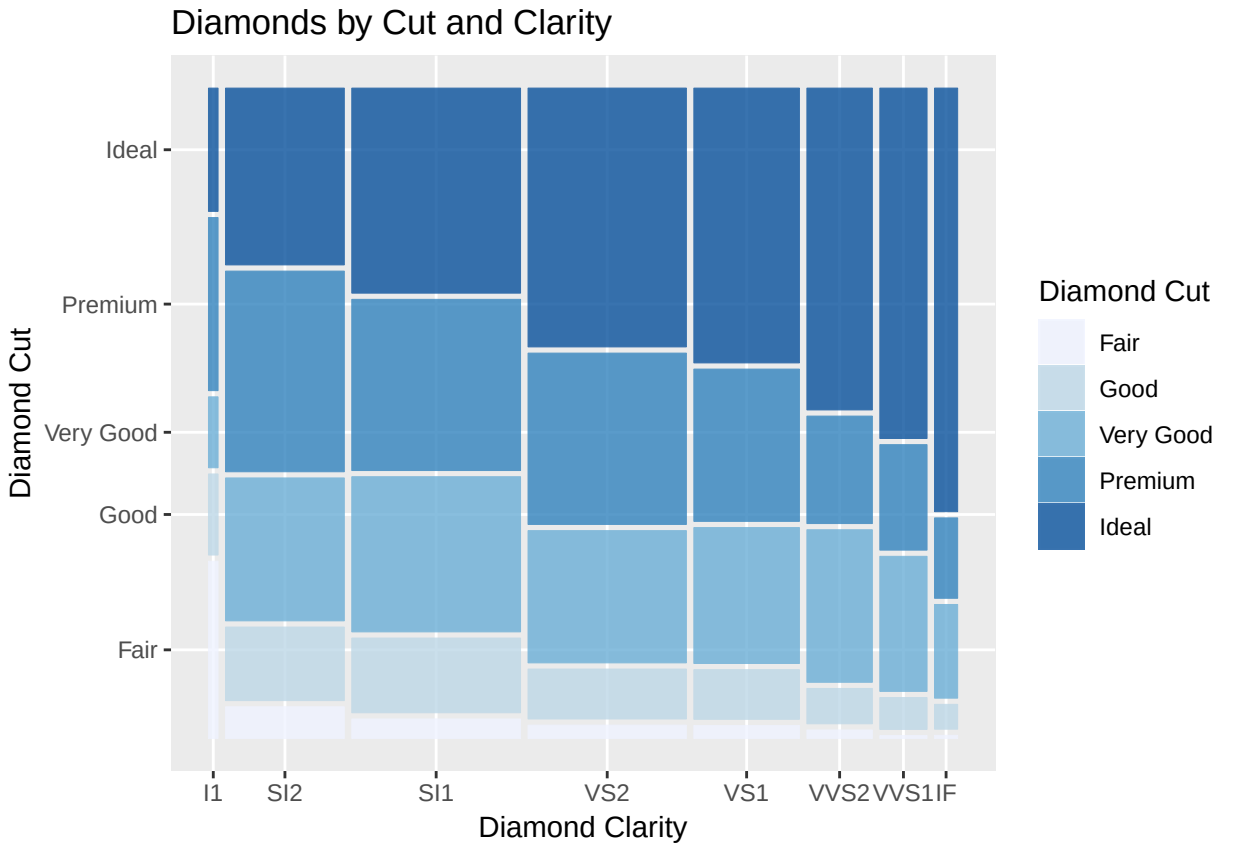
Your code here

```
require(ggmosaic)
```

```
## Loading required package: ggmosaic
```

```
## Warning: package 'ggmosaic' was built under R version 3.4.4
```

```
ggplot(data = diamonds)+
  geom_mosaic(aes(x = product(clarity), fill = cut))+
  xlab("Diamond Clarity")+
  ylab("Diamond Cut")+
  scale_fill_brewer("Diamond Cut")+
  ggtitle("Diamonds by Cut and Clarity ")
```



Your answer

Mosaic plots are interpreted by comparing areas. The width of each Clarity category helps to show how many observations of that clarity we have in proportion to the other categories. The cut factor shows us how many of those diamonds in each category are in each cut category by proportion. For example, it appears that clarity SI1 has the most observations in the data set, however, IF has the higher proportion of ideal cut diamonds in the data set.

Bonus question: Include a URL to a "tale" you created that carries out the code you created for this homework in

RStudio implemented on the WholeTale platform at wholetale.org . A "tale" is the output of a some code and it

includes the code as well. You'll need to log on to Wholetal.org using your UIUC ID. Wholetale is an ongoing research

project at UIUC so it would also be useful to hear about any problems you ran into using Wholetale to implement

your homework code (extra bonus there :)) See https://wholetale.readthedocs.io/users_guide/index.html