HW5_IS457_85

Mon Oct 22, 2018

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 question are 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

(1) 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".
2), add the density curve of 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?

Ans :

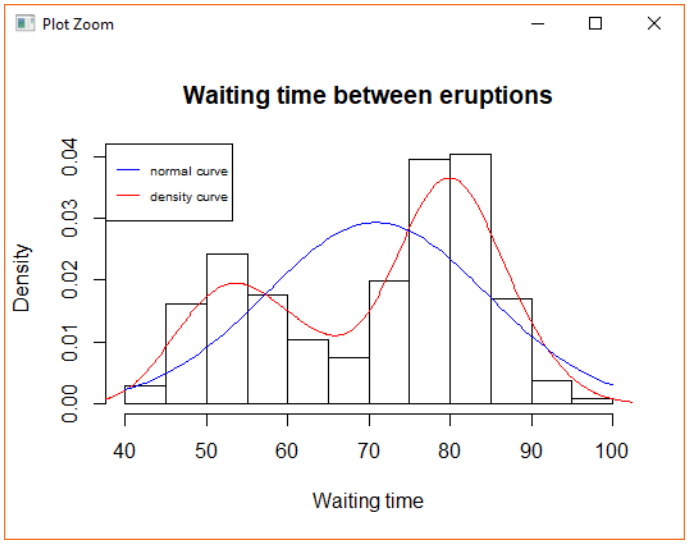
```
data("faithful")

hist(faithful$waiting, freq = FALSE, xlab = "Waiting time", main = "Waiting time between eruptions")

lines(density(faithful$waiting), col = "Red")

curve(dnorm(x,mean = mean(faithful$waiting), sd = sd(faithful$waiting)), from = 40, to = 100, add = TRUE, col = "blue")

legend("topleft", legend = c("normal curve", "density curve"),  col = c("Blue","Red"), lty = 1, lwd = 1.5, cex = 0.55)
```



The histogram shows the frequency of values of waiting time for the geyser eruption. For the histogram we can infer that there might be two modes for this data. This can be more clearly seen by plotting the density graph of the distribution. We can see that the main mode for the data is around the value 80 and the second mode is around the value 55. After imposing the normal distribution, we can observe that the waiting time data can be said to have two normal distributions having different start and end range.

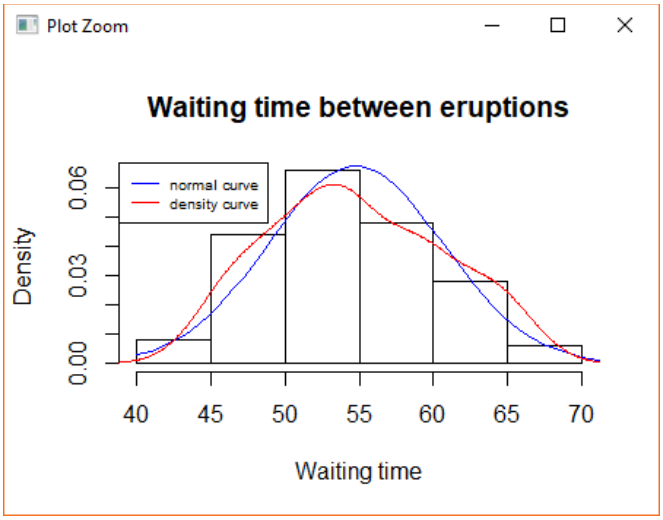
Based on a cursory glance we would split the distribution at value 68 as this is where first distribution ends and second distribution start. We look at whether this assumption holds true. We will generate two waiting time density distribution splitting the dataset having 68 as our cut value.

hist(faithful$waiting[faithful$waiting<68], freq = FALSE, xlab = "Waiting time", main = "Waiting time between eruptions")

lines(density(faithful$waiting[faithful$waiting<68]), col = "Red")

curve(dnorm(x,mean = mean(faithful$waiting[faithful$waiting<68]), sd = sd(faithful$waiting[faithful$waiting<68])), from = 40, to = 100, add = TRUE, col = "blue")

legend("topleft", legend = c("normal curve", "density curve"), col = c("Blue","Red"), lty = 1, lwd = 1.25, cex = 0.60)

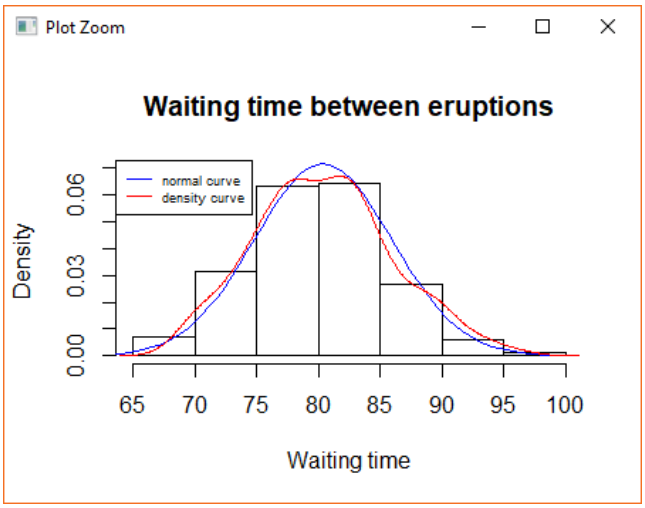


hist(faithful$waiting[faithful$waiting>68], freq = FALSE, xlab = "Waiting time", main = "Waiting time between eruptions", ylim = c(0,0.07))

lines(density(faithful$waiting[faithful$waiting>68]), col = "Red")

curve(dnorm(x,mean = mean(faithful$waiting[faithful$waiting>68]), sd = sd(faithful$waiting[faithful$waiting>68])), from = 40, to = 100, add = TRUE, col = "blue")

legend("topleft", legend = c("normal curve", "density curve"), col = c("Blue","Red"), lty = 1, lwd = 1.25, cex = 0.56)



We can see that although the two distributions do not correspond to normal distribution faithfully they can surely approximate by normal distribution. So, our assumption that the waiting time distribution can be said to contain two normal distribution holds true.

(2) Q2. Comparing distributions.
generate 3 distributions 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 distribution overlap? is there area overlapped by all three distribution? if yes, why?

Ans :

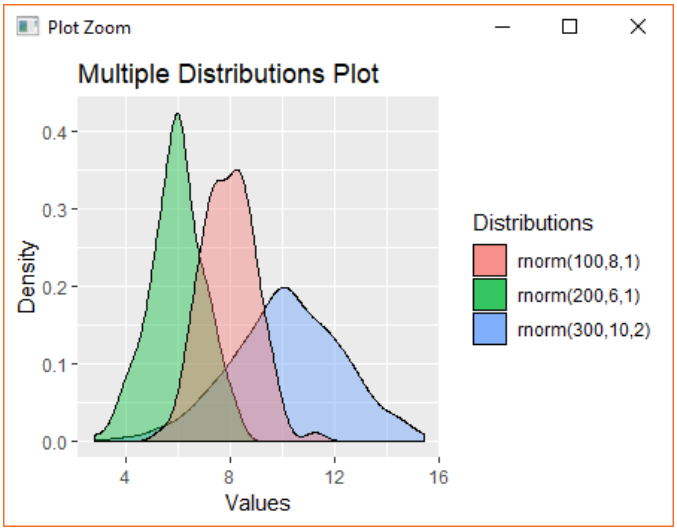
A = data.frame(a1 = rnorm(300,10,2))

B = data.frame(b1 = rnorm(200,6,1))

C = data.frame(c1 = rnorm(100,8,1))

```
rCol = c(rgb(1,0,0),rgb(0,1,0),rgb(0,0,1))

ggplot() + geom_density(data = A, aes(a1, fill = rCol[1]), alpha = 0.4) +
geom_density(data = B, aes(b1, fill = rCol[2]), alpha = 0.4) +
geom_density(data = C, aes(c1, fill = rCol[3]), alpha = 0.4) +
ggtitle("Multiple Distributions Plot") + labs(x = "Values", y = "Density") +
scale_fill_discrete(name = "Distributions", labels = c("rnorm(100,8,1)","rnorm(200,6,1)", "rnorm(300,10,2)"))
```



The above plot shows us 3 normal distributions centered on different means viz. 6, 8, 10. The plot clearly shows the effect of increasing the sample size and standard deviation. Increasing the sample size increases the height of the distribution. We then have a narrow and a tall peak distribution. The standard deviation has the similar effect but with regards to width of the distribution. Increasing the standard deviation increases the width of the distribution thereby we don't get pointed peaks as compared to when we increase the sample size.

There is an overlap between all the three distributions because even though they are centered on different means their standard deviation value makes its possible for an overlap. If we decrease the standard deviation then there it is possible to have no overlap between the distributions.

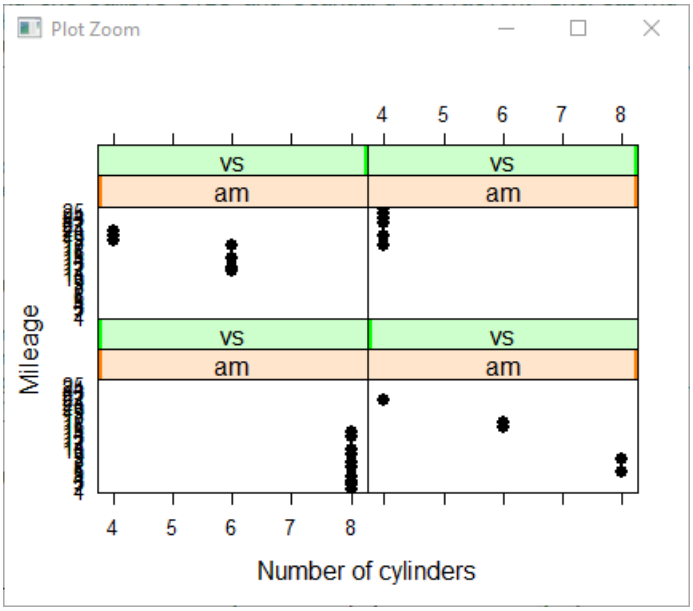
(3) Boxplots to display multivariate relationships
We will use the mtcars data set.
We've shown you how to use boxplot with one variable with multiple levels in base R command, now let's try with multiple variables using a function from package lattice, look up the manual.
make a boxplot to display the 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 stands out? explain how/why this kind of plot can be useful.

Ans :

```
require(lattice)

bwplot(mpg ~ cyl | am * vs, data = mtcars, xlab = "Number of cylinders", ylab = "Mileage")
```



The above box plot compares vs and am variables across their different values of mileage and number of cylinders. Both am (automatic or manual) and vs (v-shaped of straight engine) are binary variables with values 0 and 1. This graph is useful to compare two variables with respect to other variables in the dataset. In the above plot we can see that cars having a v-shaped engine (vs=0), the bottom row in the plot tend to have low mileage as compared to straight engines. Also, straight engines tend to have 4 or 6 cylinders and not 8 cylinders.

(4) 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.

Ans :

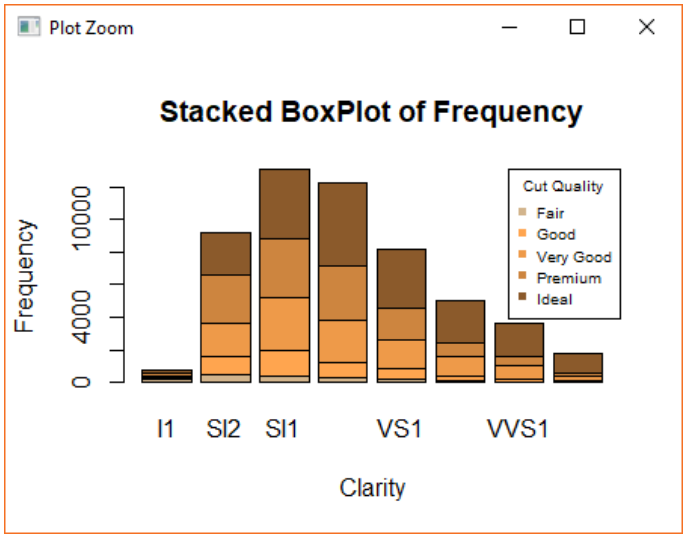
```
require(ggplot2)

contingency_tab = table(diamonds$cut, diamonds$clarity, dnn = c("Cut", "Clarity"))

barplot(contingency_tab, col = c("tan", "tan1", "tan2", "tan3", "tan4"), main = "Stacked BoxPlot of

Frequency", xlab = "Clarity", ylab = "Frequency")

legend("topright", legend = c("Fair", "Good", "Very Good", "Premium", "Ideal"), col = c("tan", "tan1", "tan2",
"tan3", "tan4"), title = "Cut Quality", pch = 25, cex = 0.625)
```



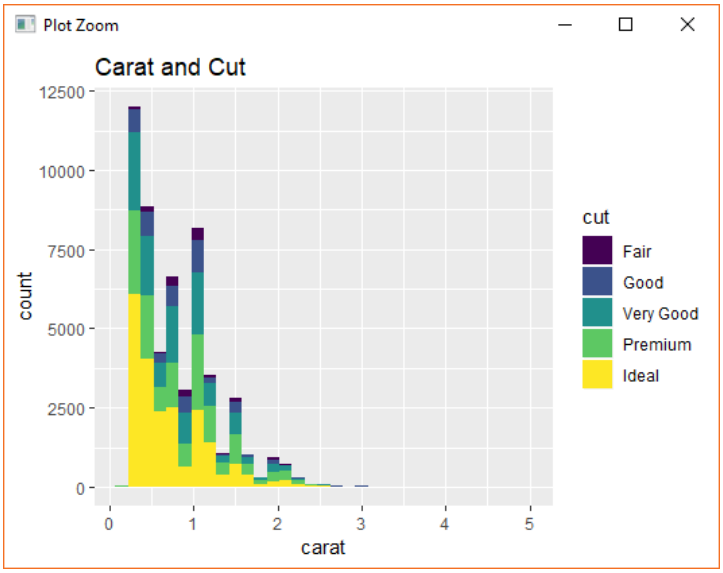
The above plot shows the frequency of diamonds with cut quality across different clarity ranges. We can observe that the number of diamonds with higher cut quality increases as we move from lower clarity of I1 to highest clarity IF.

Part 2. Fancy plots with ggplot2 and ggmosaic also using diamonds data set for both questions.

(1) 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.

Ans :

ggplot(diamonds, aes(carat, fill = cut)) + geom_histogram(binwidth = 0.15) +labs(title = "Price and Cut")



The plot above shows us the counts of carats across different range as well as the distribution of cut quality in that range. From the above plot we can observe that the number of diamonds with higher cut quality decreases

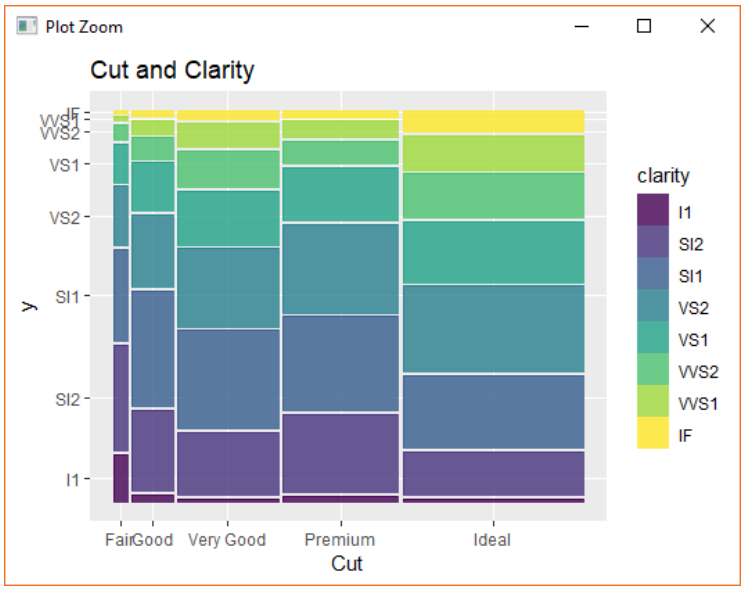
as we increase the carat size. We can say from this data that the diamonds with higher carat size have less cut quality in this dataset.

(2) 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.

Ans :

require(ggmosaic)

ggplot(diamonds)+ geom_mosaic(aes(weight = 1, x = product(cut), fill = clarity)) + labs(x ="Cut", title = "Cut and Clarity")



The above plot compares cut quality and the clarity variables of the diamond. The plot is used to compare the proportion of values between two qualitative variables. In this case both cut, and clarity are qualitative variables. The x axis has the quality of cut variable whereas the y axis has the clarity variable. The above plot shows that lowest cut quality of the diamond (Fair) also has highest share of diamonds with lowest clarity (L1) when compared to the other cut quality values. Also, the diamonds with highest cut quality (Ideal) have more share of the highest clarity (IF) among other diamonds. This helps in comparing the proportions of diamonds across different cut and clarity variables.

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 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

Ans :

https://tmp-ev1kcrvs61lr.prod.wholetale.org/