EDAV Fall 2019 Probem Set 1

Harish Visweswaran (hv2197) and Amogh Mishra (am5323)

```
# Loading all necessary packages
library(ucidata)
library(ggplot2)
library(dplyr)
library(nullabor)
# Using this package only where using faceting is cumbersome
library(gridExtra)
```

1. Abalone

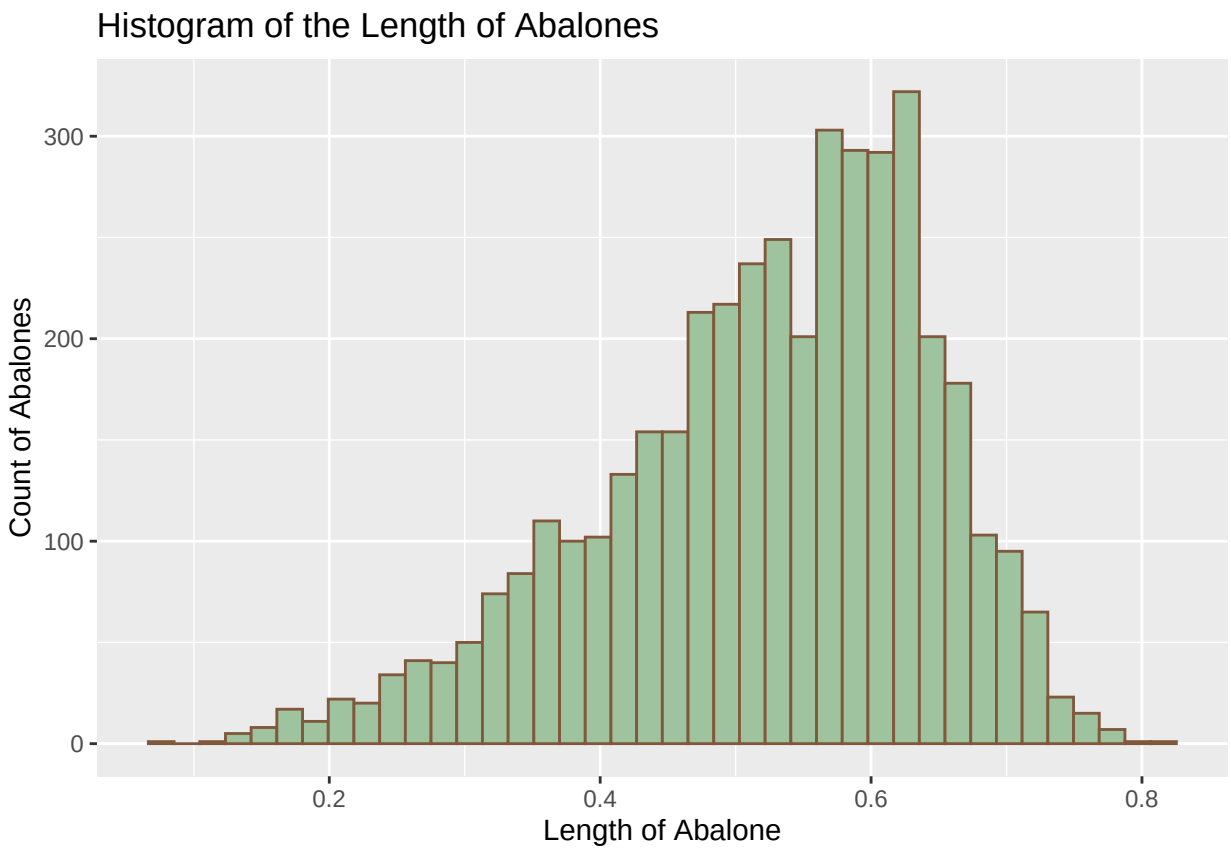
[18 points]

Choose one of the numeric variables in the `abalone` dataset.

a) Plot a histogram of the variable.

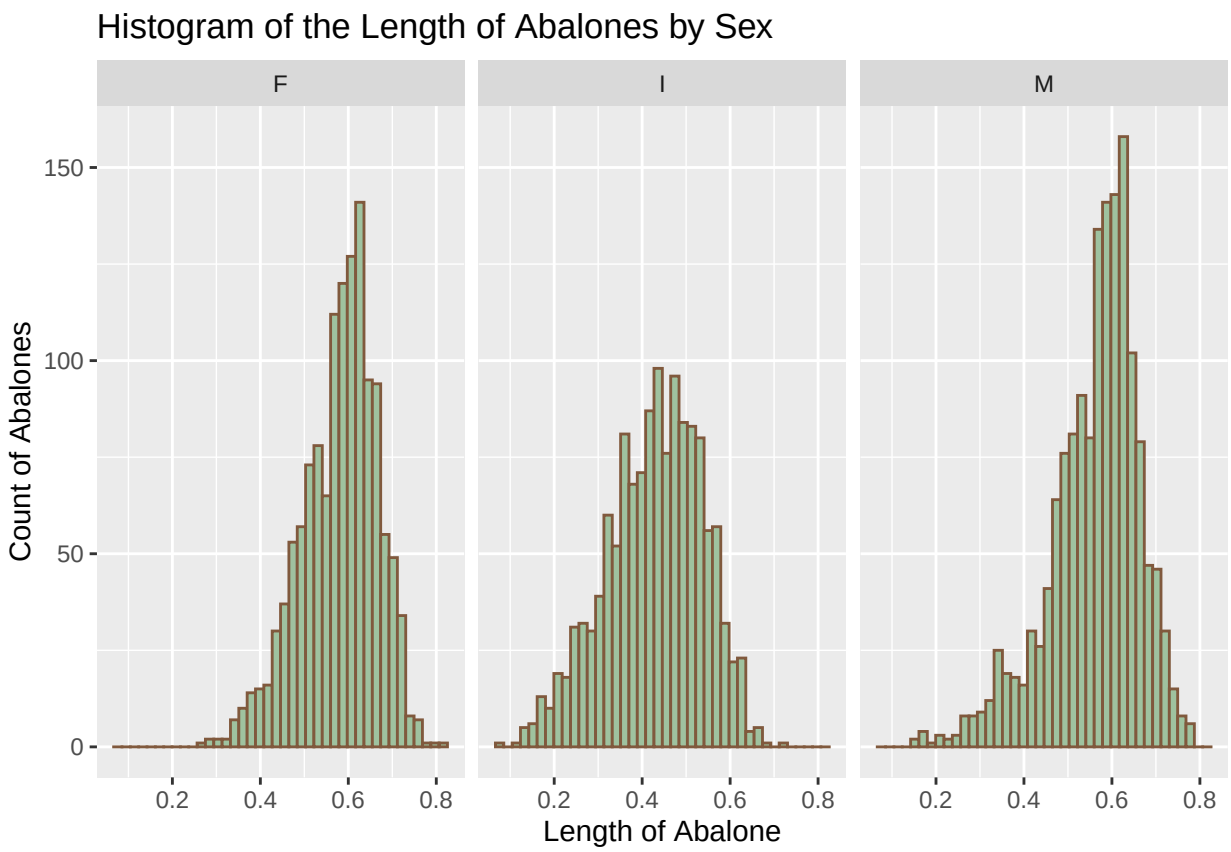
We have chosen the Length Variable to analyse for this question

```
ggplot(data=abalone, aes(x=length)) +
  geom_histogram(fill = "#9FC29F", color = "#80593D", bins=40) +
  labs(x="Length of Abalone",
       y="Count of Abalones",
       title="Histogram of the Length of Abalones")
```



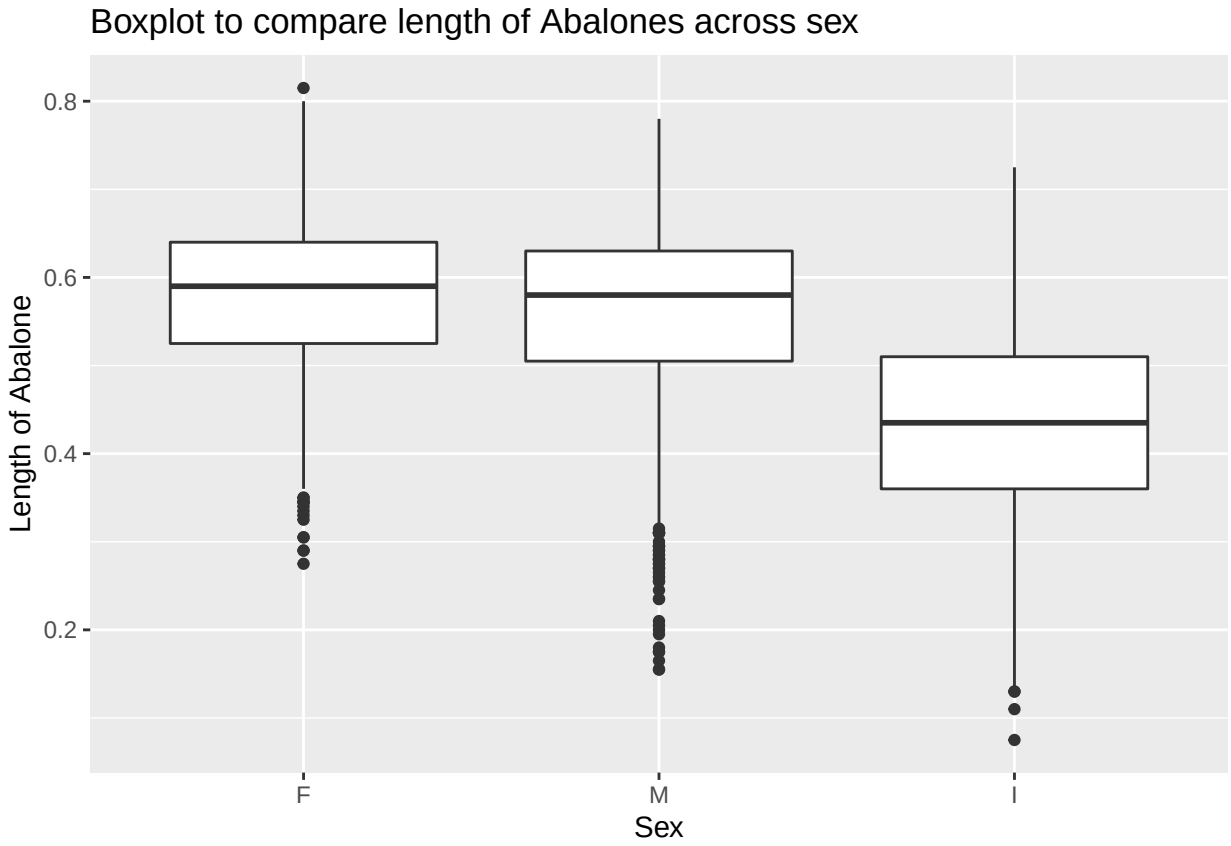
b) Plot histograms, faceted by **sex**, for the same variable.

```
ggplot(data=abalone, aes(x=length)) +
  geom_histogram(fill = "#9FC29F", color = "#80593D", bins=40) +
  facet_wrap(~sex) +
  labs(x="Length of Abalone",
       y="Count of Abalones",
       title="Histogram of the Length of Abalones by Sex")
```



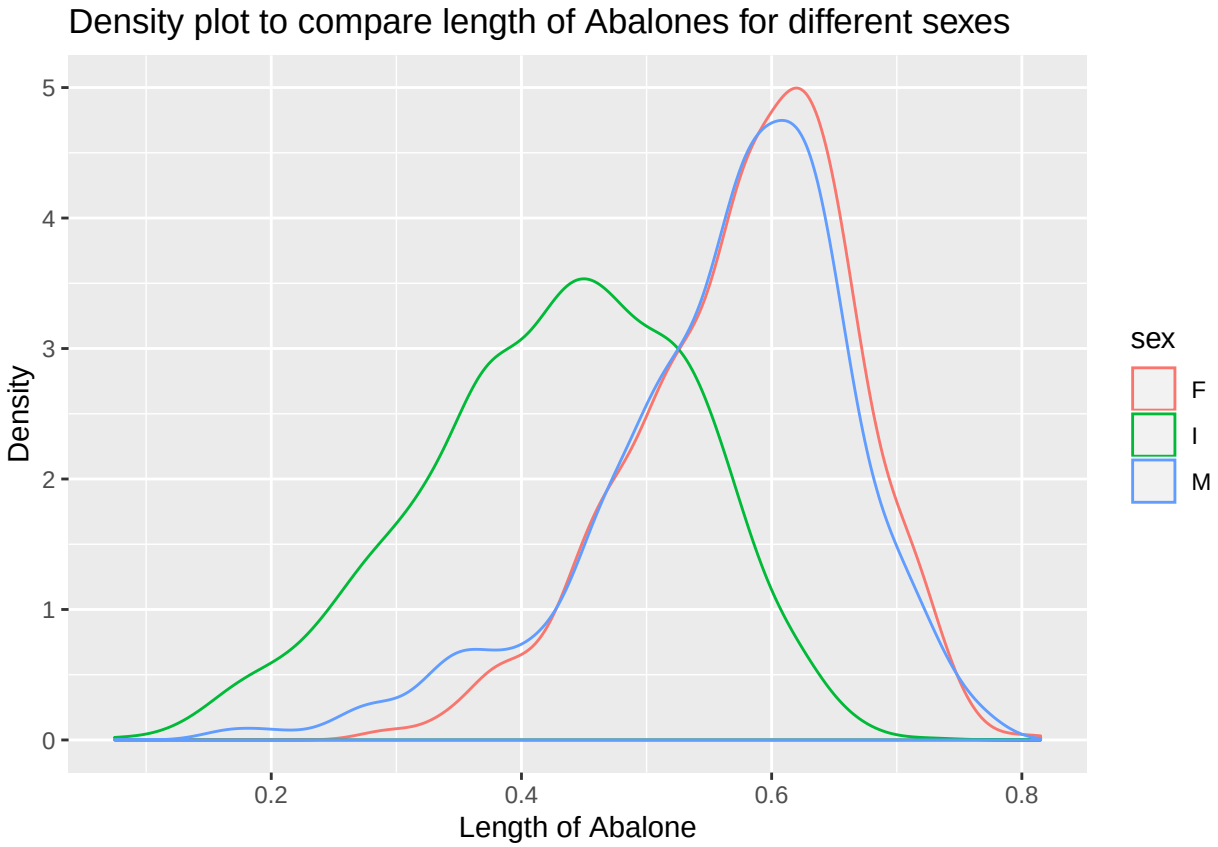
c) Plot multiple boxplots, grouped by **sex** for the same variable. The boxplots should be ordered by decreasing median from left to right.

```
ggplot(data=abalone, aes(x = reorder(sex, -length, median), y = length)) +
  geom_boxplot() +
  labs(x="Sex",
       y="Length of Abalone",
       title="Boxplot to compare length of Abalones across sex")
```



d) Plot overlapping density curves of the same variable, one curve per factor level of `sex`, on a single set of axes. Each curve should be a different color.

```
ggplot(data=abalone, aes(x=length, color=sex)) +
  geom_density() +
  labs(x="Length of Abalone",
       y="Density",
       title="Density plot to compare length of Abalones for different sexes")
```



e) Summarize the results of b), c) and d): what unique information, *specific to this variable*, is provided by each of the three graphical forms?

Observations from the Histograms:

- The histograms give us a clear picture of the distribution of data across the values it takes. The histograms appear to be unimodal across Male, Female and Infants. The histogram for infants does not have a peak that is high like the female and male histograms which intuitively makes sense. There is some left skew (negative skew) across all three histograms. It is slightly more pronounced in the Males histogram.
- The males histogram seems to have some low values (around 0.2) but the females histogram doesn't have any around that value. Potentially, some infants might have been misclassified as male or they could just be outliers which we can observe better in the boxplot

Observations from the Boxplots:

- We can see from the boxplots that the female abalones have the highest median followed by the male abalones and then the infants. The longest abalone also seems to be a female Abalone
- Additionally, we can observe the outliers for Males from the boxplot. We see there are quite a few values below the lower quartile including some values around 0.2
- We can confirm our observation that the data is left skewed by noticing that the median sits slightly above the midpoint of the hinges for all three plots (more so for the Males plot)

Observations from the Density plots:

- The density plots provide a clear picture of the difference in distributions of the length data for Infants and Males/Females because we can see and compare all the densities in one plot (as opposed to the faceted histogram). Males and Females more or less have a similar distribution with Males having a slightly larger left skew but the Infant distribution is quite different from the other two. The infant distribution looks much more symmetrical than the other two and also has a lower peak and seems to

have a higher variance.

f) Look at photos of an abalone. Do the measurements in the dataset seem right? What's the issue?

The max length of the abalone is 0.815 and the min length is 0.075. The description that comes with the help on the abalone dataset mentions that the length is in mm. On checking generic images online, abalones seemed to be about or a little less than the size of a palm. Clearly then, the numbers (which are all less than 1) cannot be raw mm values. It looks like the data has been rescaled.

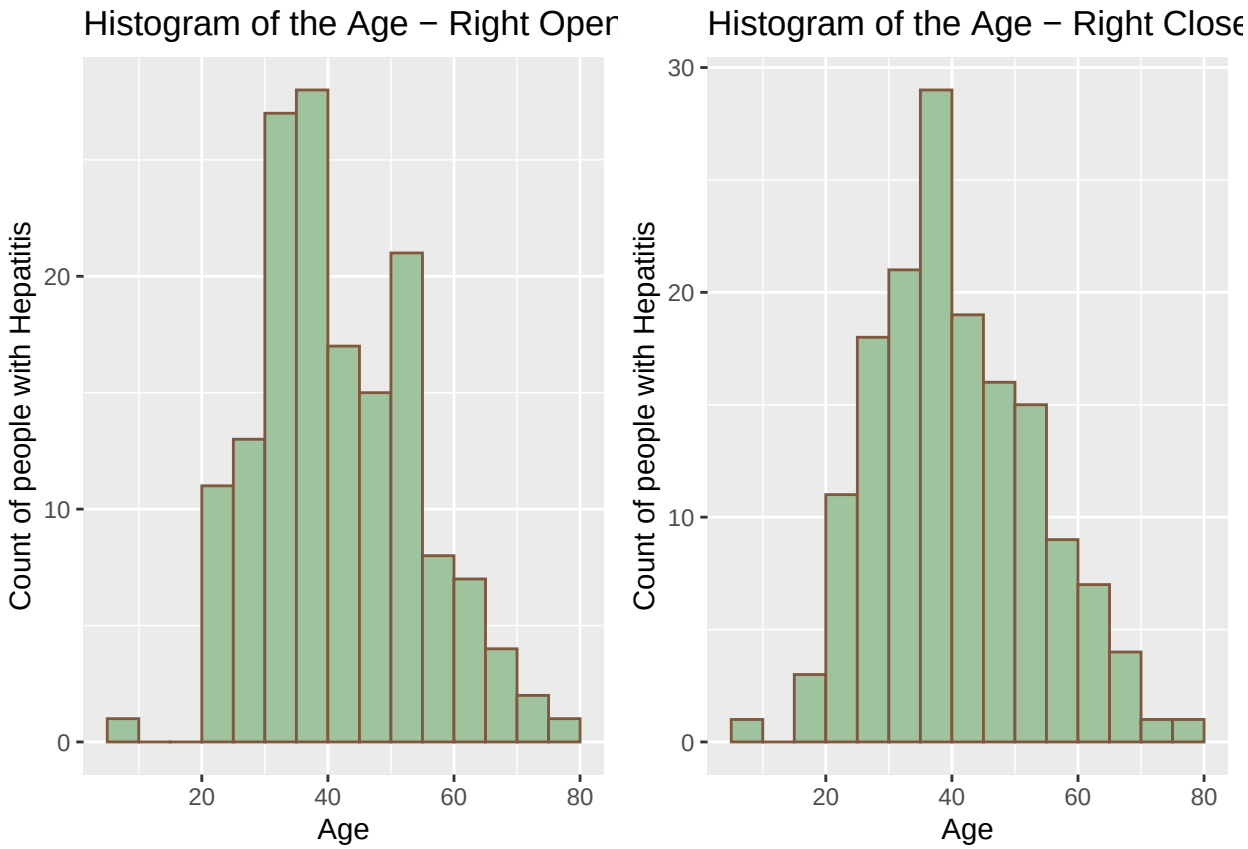
On checking further, we found this on the UCI ML Repository site which was in line with our thoughts: The ranges of the continuous values have been scaled for use with an ANN (by dividing by 200).
https://archive.ics.uci.edu/ml/datasets/Abalone

2. Hepatitis

[6 points]

a) Draw two histograms of the age variable in the `hepatitis` dataset in the **ucidata** package, with binwidths of 5 years and `boundary = 0`, one right open and one right closed. How do they compare?

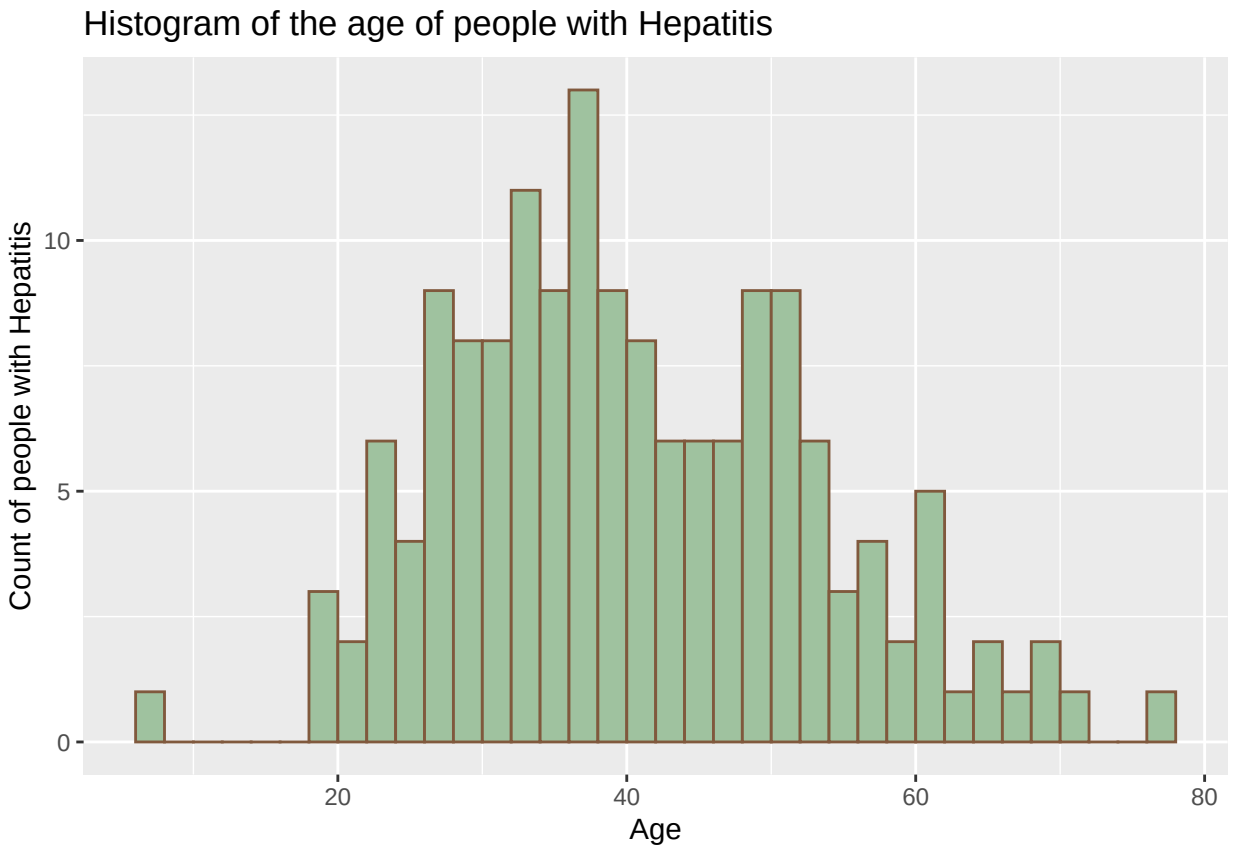
```
right_open <- ggplot(data=hepatitis, aes(x=age)) +
  geom_histogram(fill = "#9FC29F",
                 color = "#80593D",
                 binwidth=5,
                 boundary=0,
                 closed="left") +
  labs(x="Age",
       y="Count of people with Hepatitis",
       title="Histogram of the Age - Right Open")


right_closed <- ggplot(data=hepatitis, aes(x=age)) +
  geom_histogram(fill = "#9FC29F",
                 color = "#80593D",
                 binwidth = 5,
                 boundary = 0,
                 closed = "right")  +
  labs(x="Age",
       y="Count of people with Hepatitis",
       title="Histogram of the Age - Right Closed")

grid.arrange(right_open, right_closed, nrow=1)
```



The right closed histogram seems to have a clear peak between 35 and 40 while the right open histogram seems to have a few bins with high values (30 to 35, 35 to 40 and 50 to 55). The right open histogram almost looks bimodal because of the bin at 50 to 55 as opposed to the right closed histogram which is visibly unimodal. On comparing the two grapsh, we can say that there are at least a few values at age 30. In summary, this is a clear demonstration that the parameter choices when plotting a histogram can make a difference in the observations made about the data.

b) Redraw the histogram using the parameters that you consider most appropriate for the data. Explain why you chose the parameters that you chose.

```
ggplot(data=hepatitis, aes(x=age)) +
  geom_histogram(fill = "#9FC29F",
                 color = "#80593D",
                 boundary = 0,
                 binwidth = 2) +
  labs(x = "Age",
       y = "Count of people with Hepatitis",
       title = "Histogram of the age of people with Hepatitis")
```



We chose a bin width of 2 as we wanted a more granular view of how the data was distributed than using a bin width of 5. We have left the default ggplot setting of right closed histograms. In this we do observe the additional peak in the bins around age 52

3. Glass

[18 points]

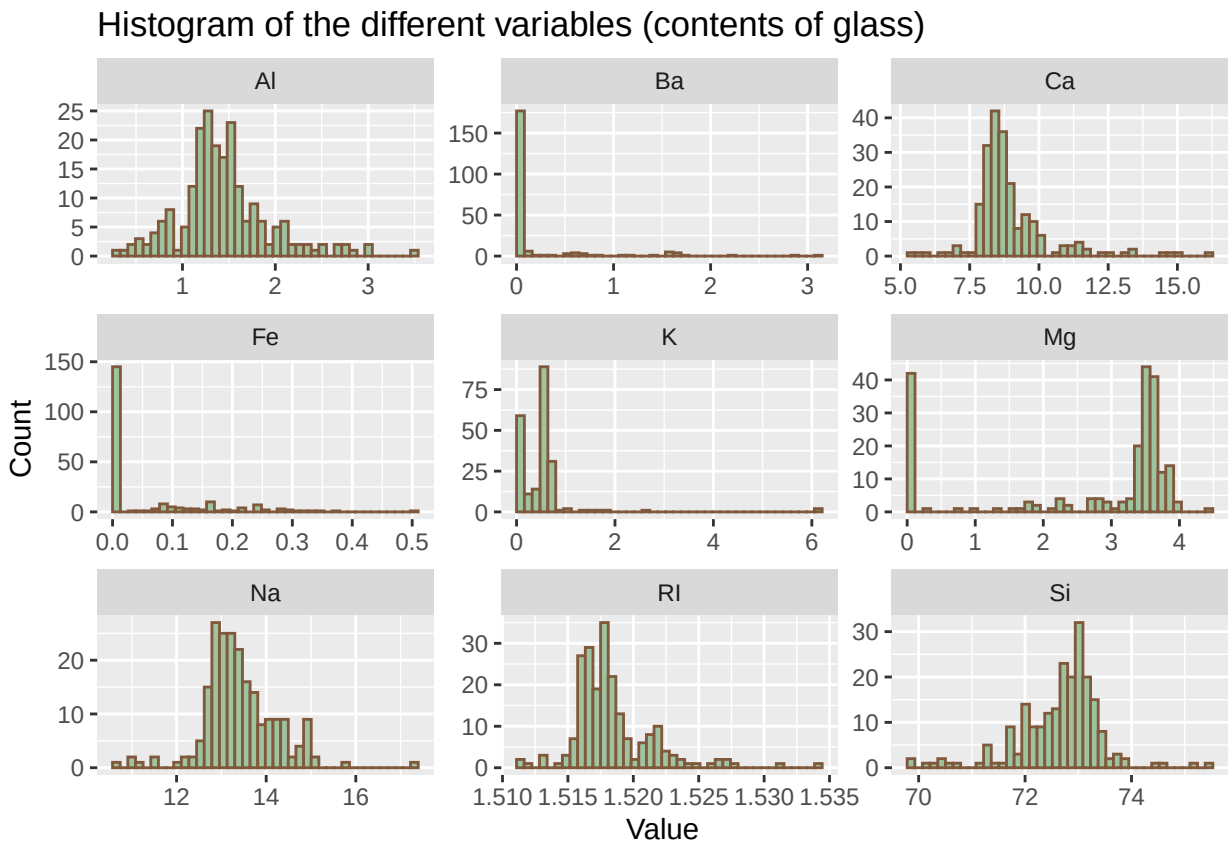
a) Use `tidyr::gather()` to convert the numeric columns in the `glass` dataset in the **ucidata** package to two columns: `variable` and `value`. The first few rows should be:

```
  variable   value
1       RI 1.52101
2       RI 1.51761
3       RI 1.51618
4       RI 1.51766
5       RI 1.51742
6       RI 1.51596
```

Use this form to plot histograms of all of the variables in one plot by faceting on `variable`. What patterns do you observe?

```
# Gathering the data first
glass_numeric = subset(glass, select = -c(Type, ID))
glass_tidy <- tidyr::gather(glass_numeric, key="variable", value="value")

ggplot(glass_tidy, aes(x=value)) +
  geom_histogram(fill = "#9FC29F",
                 color = "#80593D",
```

```
                boundary=0,
                bins=40) +
  facet_wrap(~variable, scales="free") +
  labs(x = "Value",
       y = "Count",
       title = "Histogram of the different variables (contents of glass)")
```



Here, we have let the scales to be free as using a fixed scale did not let us observe the distributions of the data as different variables had very different ranges. We can see from the plots that Al, Ca, Na, RI and Si seem to exhibit a unimodal distribution that is skewed in some cases. Ba and Fe have most of their values at 0 with a long right tail. K has most of its values in the first few bins (below 1) and Mg is interesting because it has a huge spike at 0 and then another spike around 3.5 (seems bimodal).

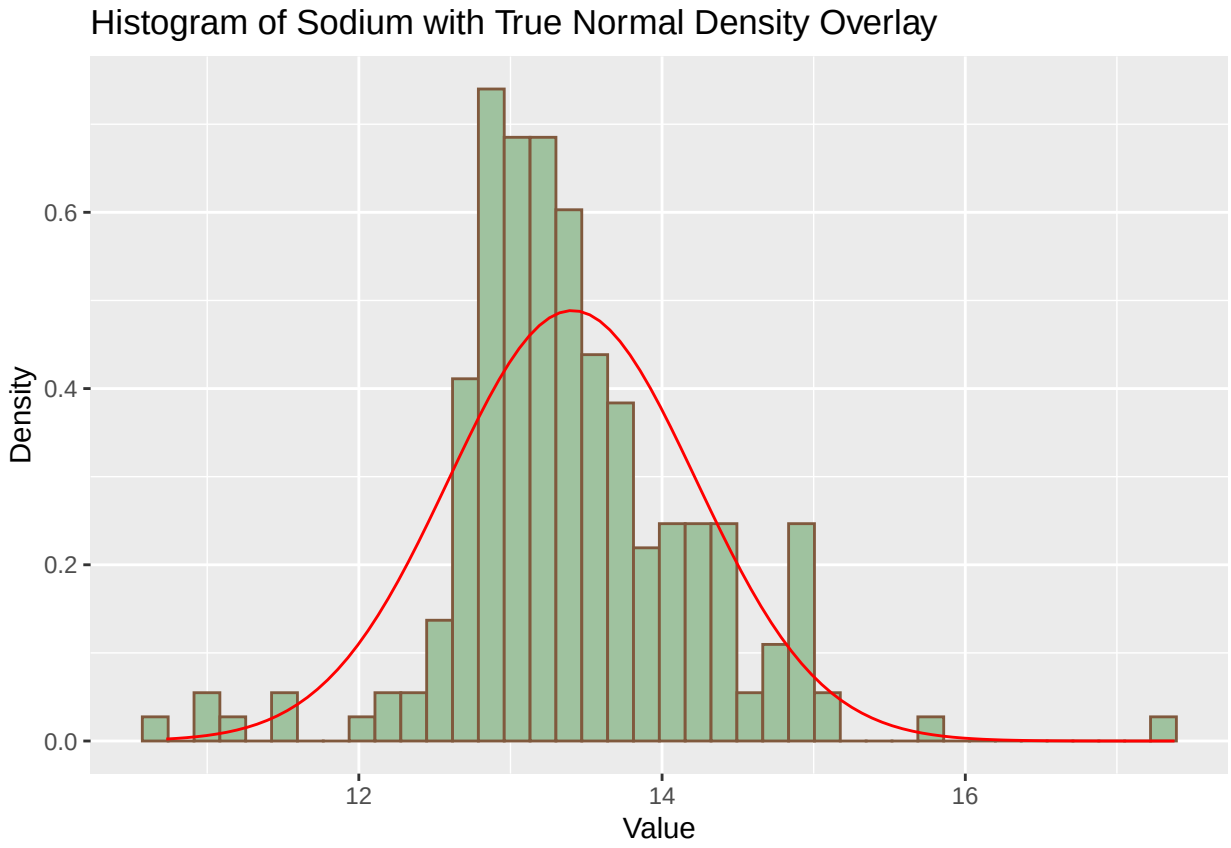
For the remaining parts we will consider different methods to test for normality.

b) Choose one of the variables with a unimodal shape histogram and draw a true normal curve on top on the histogram. How do the two compare?

We Chose Na (sodium) as our variable of interest

```
# First subsetting the data to only get the variable of interest for further use
glass_na <- filter(glass_tidy, variable == "Na")

# Plotting the histogram and density curves
ggplot(glass_na, aes(x=value)) +
  geom_histogram(aes(y=..density..),
                 fill = "#9FC29F",
                 color = "#80593D",
```

```
                 boundary=0,
                 bins=40) +
  stat_function(fun = dnorm,
                color = "red",
                args = list(mean=mean(glass_na$value),
                            sd=sd(glass_na$value))) +
  labs(x = "Value",
       y = "Density",
       title = "Histogram of Sodium with True Normal Density Overlay")
```



Based on the histogram and the true normal density, we can say that the data does not seem to be normally distibuted. We notice the higher peaks and the slight right skew in the data. The data is not symmetrical like a normal distribution. The mean appears to be to the right of the median based on the histogram.

c) Perform the Shapiro-Wilk test for normality of the variable using the `shapiro.test()` function. What do you conclude?

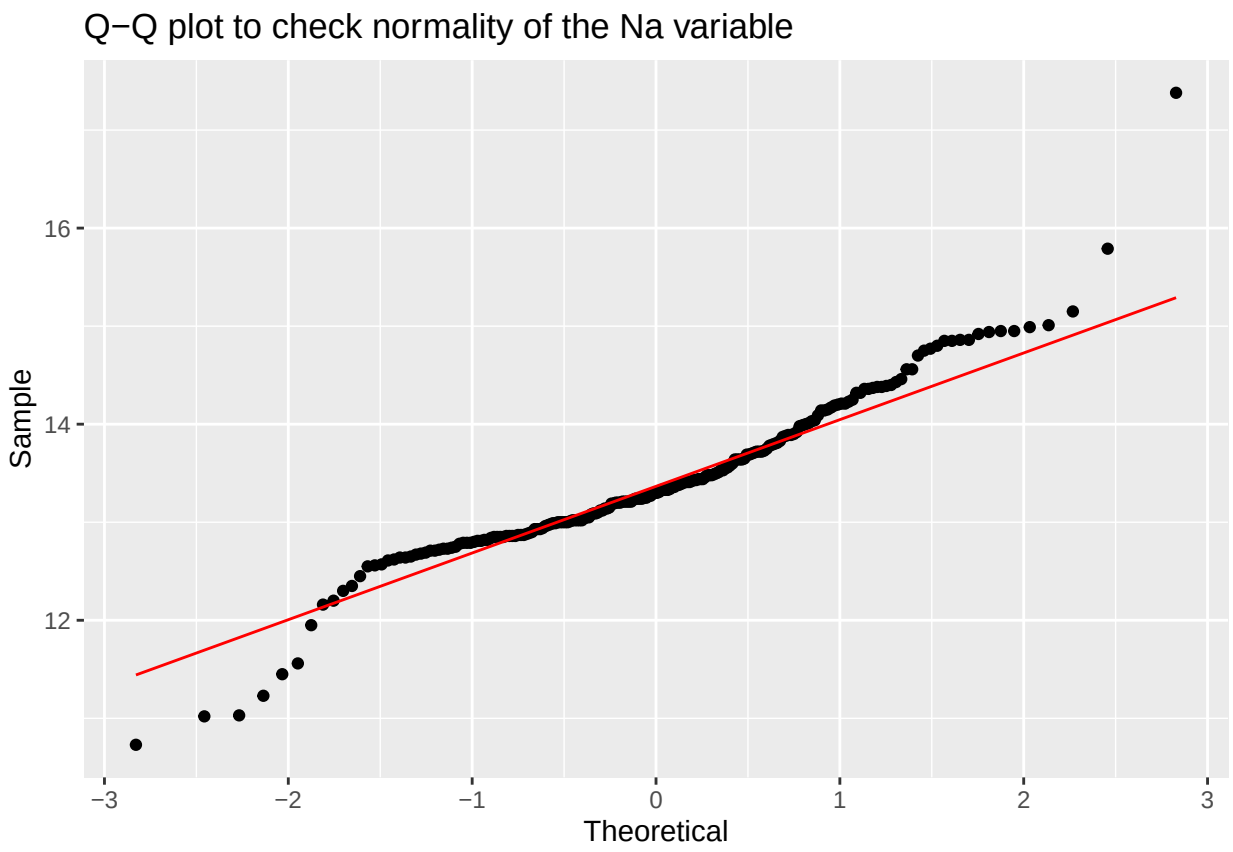
```
shapiro.test(glass_na$value)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  glass_na$value
## W = 0.94576, p-value = 3.466e-07
```

The null hypothesis for the Shapiro-Wilk test is that the data is normal. Here, we get a p-value of 3.46e-7 indicating that the data departs from normality and the null hypothesis can be rejected

d) Draw a quantile-quantile (QQ) plot of the variable. Does it appear to be normally distributed?

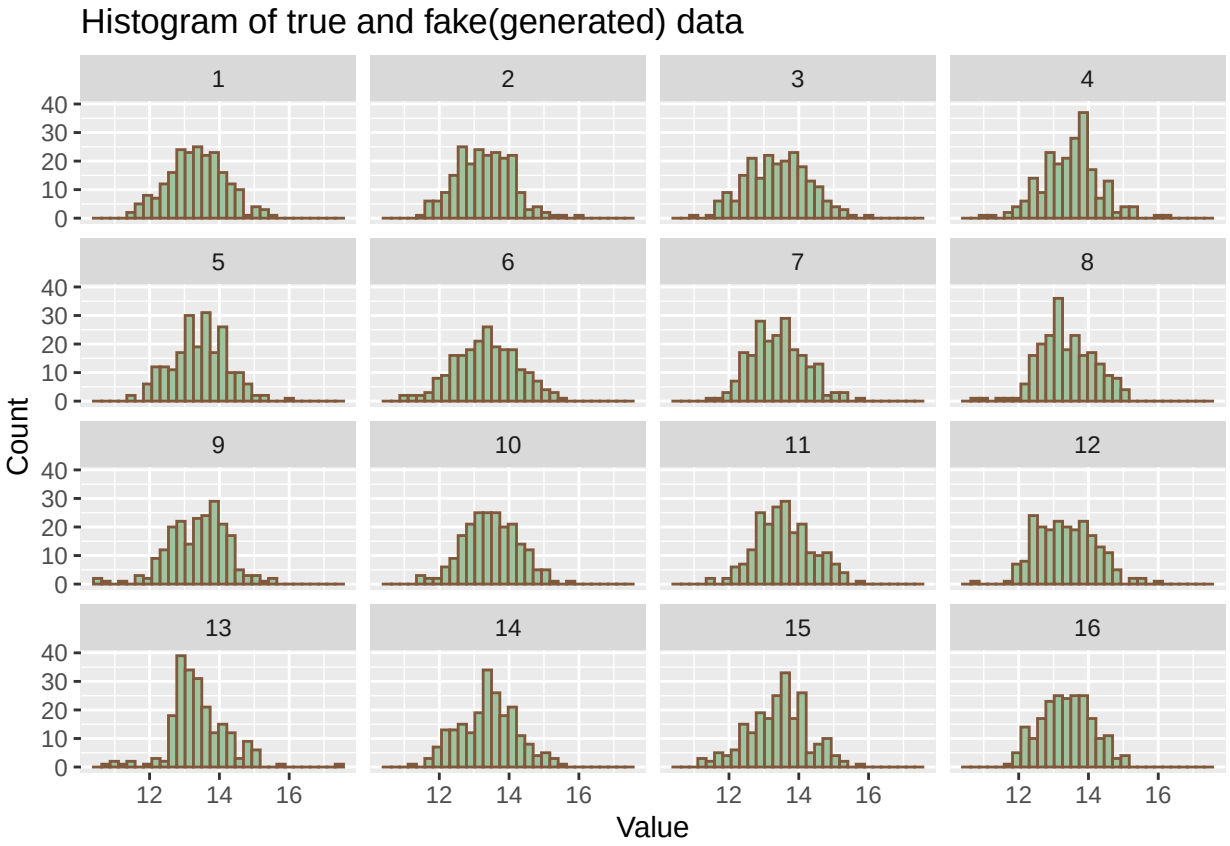
```
ggplot(data=glass_na, aes(sample=value)) +
  stat_qq() +
  stat_qq_line(col="red") +
  labs(x = "Theoretical",
       y="Sample",
       title="Q-Q plot to check normality of the Na variable")
```



The points do not appear to fall on a straight line drawn through the theoretical quantiles again confirming that the data does not appear to be normally distributed

e) Use the **nullabor** package to create a lineup of histograms in which one panel is the real data and the others are fake data generated from a null hypothesis of normality. Can you pick out the real data? If so, how does the shape of its histogram differ from the others?

```
l <- lineup(null_dist("value", "norm"), true=glass_na, n=16)
ggplot(data=l, aes(x=value)) +
  geom_histogram(fill = "#9FC29F", color = "#80593D") +
  facet_wrap(~ .sample) +
  labs(x = "Value",
       y = "Count",
       title = "Histogram of true and fake(generated) data")
```




```
# The plot number for the real data is
attr(l, "pos")
```

```
## [1] 13
```

We can see from the plot that the histogram corresponding to the real data has a higher peak compared to the data generated for the normal null hypothesis. It also does not appear to be as symmetric as the other data generated from a normal null hypothesis. This plot stands out indicating that the data is deviating from normality

f) Show the lineup to someone else, not in our class (anyone, no background knowledge required). Ask them which plot looks the most different from the others. Did they choose the real data?

Yes, she was able to (after I asked her to look closely) as the histogram looked different from the others (she mentioned that it doesn't look as symmetric as the others)

g) Briefly summarize your investigations. Did all of the methods produce the same result?

Yes, all methods produced the same result - some methods appear to be clearer than the others (ex:QQ plot) but almost all methods indicated that the data for sodium was not following a normal distribution

4. Forest Fires

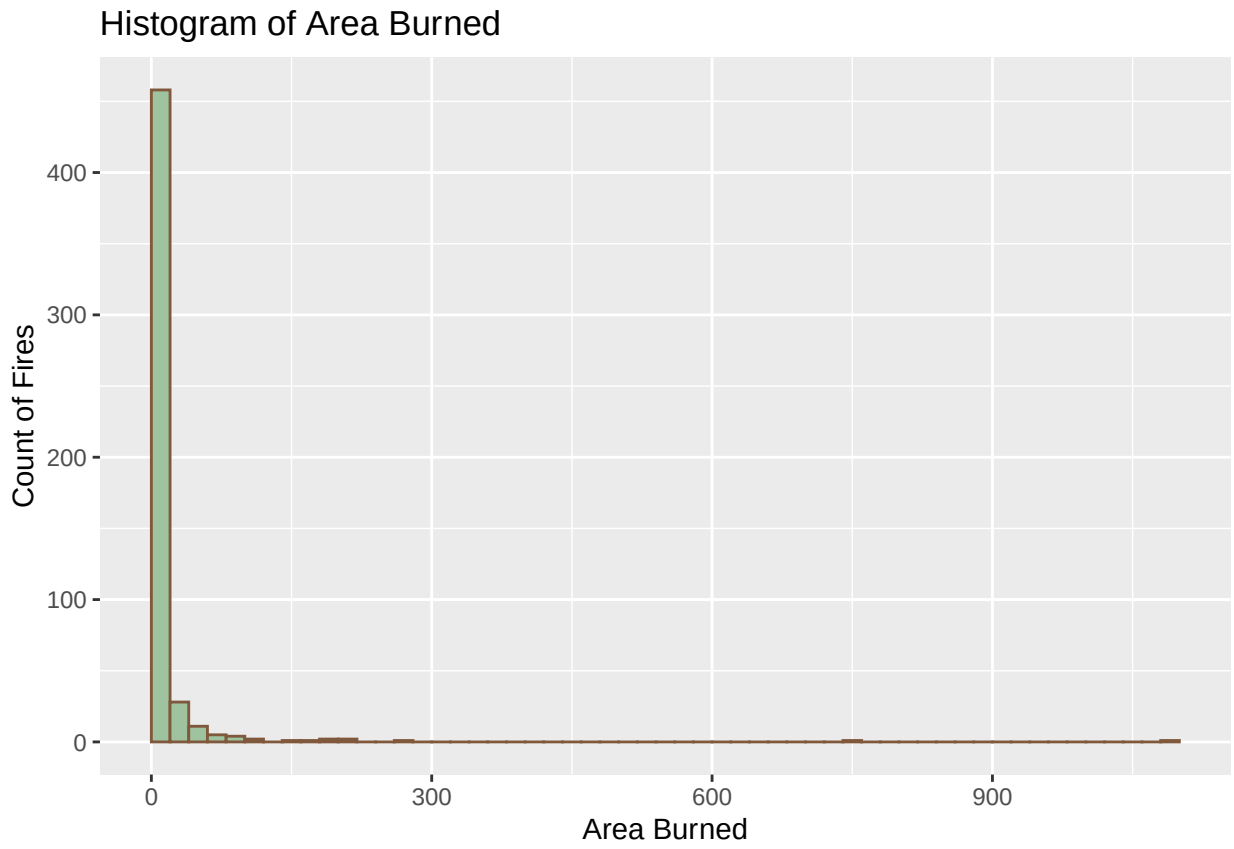
[8 points]

Using the `forest_fires` dataset in the **ucidata** package, analyze the burned area of the forest by month. Use whatever graphical forms you deem most appropriate. Describe important trends.

The data provides information on fires in Portugal with observation indicating the spatial coordinate of the fire along with other information including the area burned. We initially examine

a histogram of the area burned to understand the overall distribution of the data Then, in order to understand trends of burned area across months, we aggregate our data by summarizing the variables of interest for each month. We choose four metrics namely Total Number of Fires, Total Burned Area, Average Burned Area and Median Burned Area. We then plot column charts and line charts to examine trends

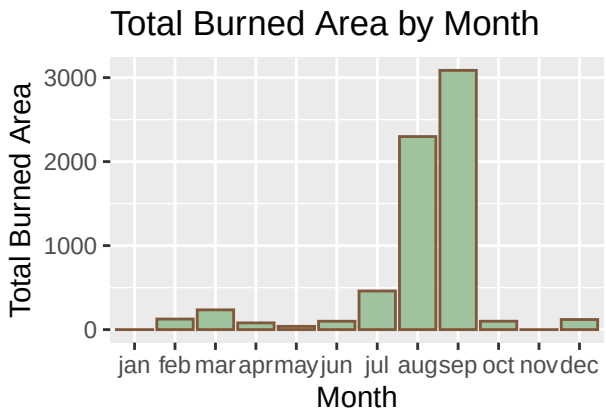
```
# Area Burned Histogram (not by month)
ggplot(data=forest_fires, aes(x=area)) +
  geom_histogram(fill = "#9FC29F",
                 color = "#80593D",
                 boundary=0,
                 binwidth=20) +
  labs(x="Area Burned",
       y="Count of Fires",
       title="Histogram of Area Burned")
```

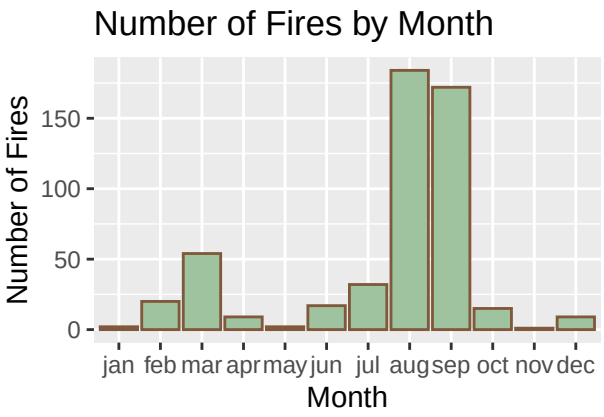


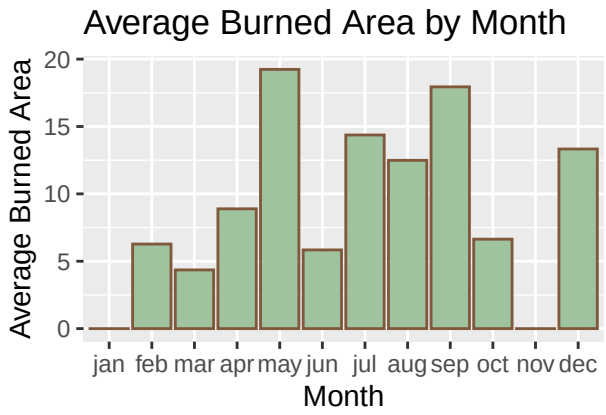
Based on the above histogram, most values seem to be in the first bin of width 20. This indicates that most of the fires affect small areas (relative to all the values of area burned in the data). Since we are interested in the area burned by month, we can summarize our data before plotting.

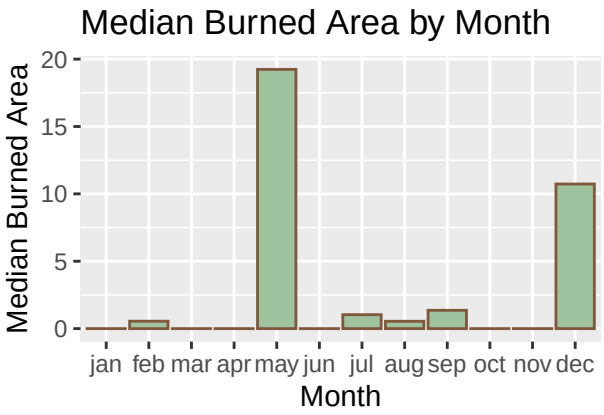
```
# Summarizing monthly fires for plotting and examining trends
month <- group_by(forest_fires, month)
monthly_fires <- summarise(month,
                           num_of_fires = n(),
                           total_burned_area = sum(area, na.rm = TRUE),
                           avg_burned_area = mean(area, na.rm = TRUE),
                           median_burned_area = median(area, na.rm = TRUE))
```

```r
# Creating a month_sort variable for sorting purposes
monthly_fires$month_sort = c(4,8,12,2,1,7,6,3,5,11,10,9)
monthly_fires <- arrange(monthly_fires, month_sort)

# Let's plot the total burned area, total number of fires,
# average burned area and median burned area by month using a column chart
total_burned_col <- ggplot(data=monthly_fires,
                           aes(x=reorder(month, month_sort),
                               y=total_burned_area)) +
  geom_col(fill = "#9FC29F", color = "#80593D") +
  labs(x="Month",
       y="Total Burned Area",
       title="Total Burned Area by Month")

total_num_of_fires_col <- ggplot(data=monthly_fires,
                                 aes(x=reorder(month, month_sort),
                                     y=num_of_fires)) +
  geom_col(fill = "#9FC29F", color = "#80593D") +
  labs(x="Month",
       y="Number of Fires",
       title="Number of Fires by Month")

avg_burned_col <- ggplot(data=monthly_fires,
                         aes(x=reorder(month, month_sort),
                             y=avg_burned_area)) +
  geom_col(fill = "#9FC29F", color = "#80593D") +
  labs(x="Month",
       y="Average Burned Area",
       title="Average Burned Area by Month")

median_burned_col <- ggplot(data=monthly_fires,
                            aes(x=reorder(month, month_sort),
                                y=median_burned_area)) +
  geom_col(fill = "#9FC29F", color = "#80593D") +
  labs(x="Month",
       y="Median Burned Area",
       title="Median Burned Area by Month")

grid.arrange(total_burned_col,
             total_num_of_fires_col,
             avg_burned_col,
             median_burned_col,
             nrow=2,
             ncol=2)
```







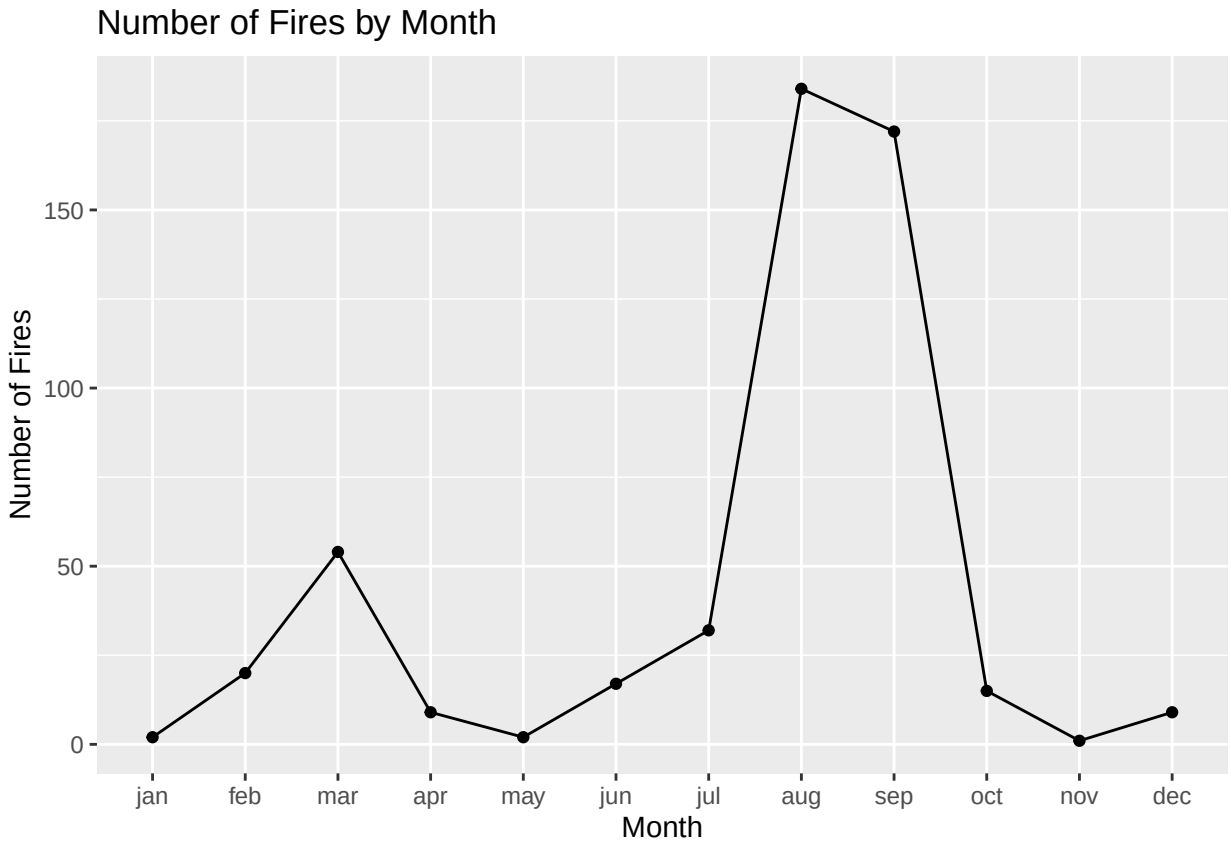


Observations

- The total area burned chart clearly tells us that August and September have the highest burned areas by the fire. This is followed by July and then March.
- When we check the total number of fires by month, we notice that we observe similar spikes in August and September, indicating that there are more fires leading to higher total area burned in those months. We now take a look at the average burned area to see if some months exhibit larger fires that others.
- We see that May has the highest average burned area (even larger than August and September) indicating that the fires in May impact larger areas. However, this could be caused by the much smaller number of fires in May and hence may not be a reliable insight.
- We then take a look at the median burned area. Comparing this plot with the average plot, we can see that a lot of fires have an area burned of 0 (the median is 0 for many months). Also, if we compare December with August and September, we can see from the plot that the fires in December have a higher median burned area than August and September. Although December has a lower number of fires, its median value is higher indicating that while the fires are not common as they are in August and September, they are harder to contain and affect a larger area in December.
- Overall it is clear that most of the fires occur in August and September followed by March.

```
# Since the data is temporal, we can also use a line chart to identify
# any trend in the data over the months. We will just use the total
# number of fires to demonstrate this as we already have shown the
# other two metrics using the column charts

ggplot(data=monthly_fires,
```

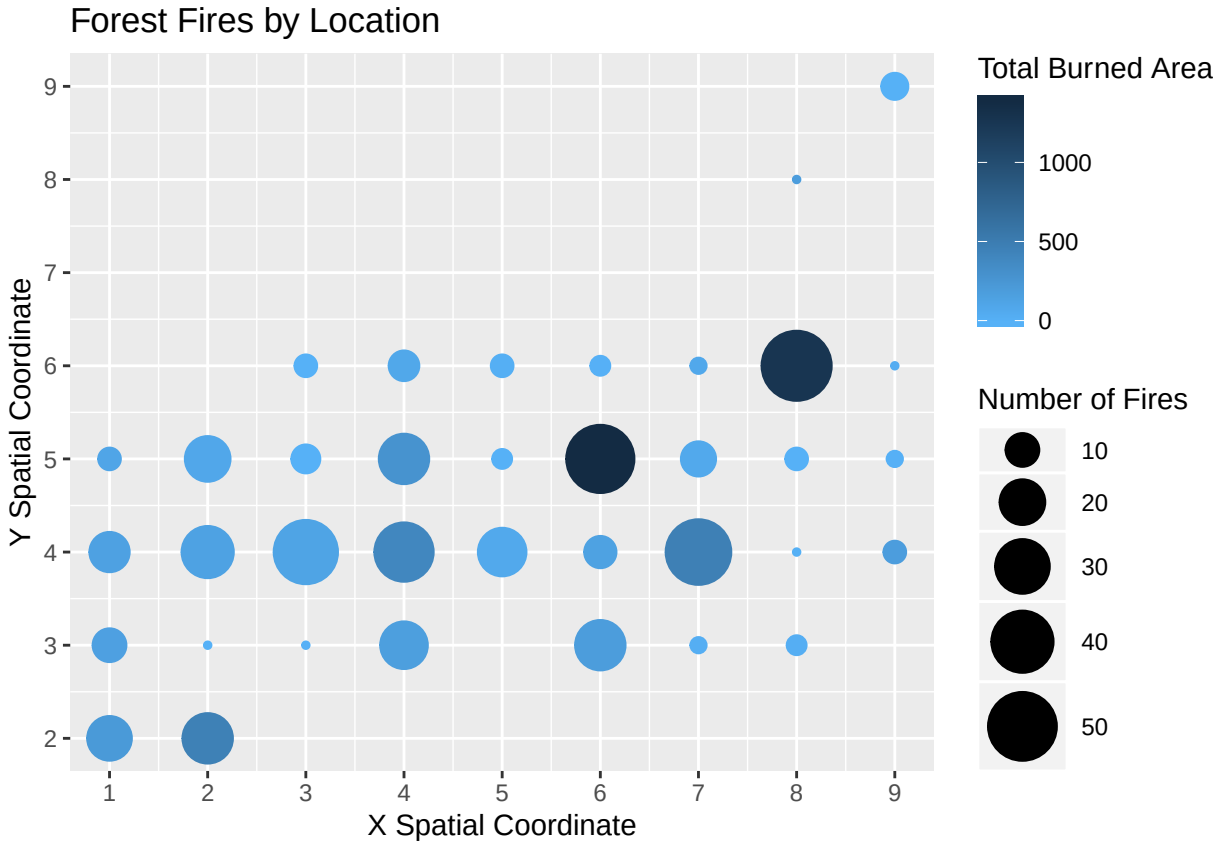
```
      aes(x=reorder(month, month_sort),
          y=num_of_fires)) +
  geom_point() +
  geom_line(group=1) +
  labs(x="Month",
       y="Number of Fires",
       title="Number of Fires by Month")
```



We can see from the line chart that there is small spike in march and then a huge spike/peak in the months of August/September with May, November and Jan hardly seeing any fires. It is clear that there is some seasonality in play here as we see a smooth trend in the data (as opposed to random spikes/ups and downs). Now, let's examine the spatial trends

```
X_Y <- forest_fires %>% group_by(X,Y)
X_Y_fires <- summarise(X_Y,
                       num_of_fires = n(),
                       total_burned_area = sum(area, na.rm = TRUE),
                       avg_burned_area = mean(area, na.rm = TRUE),
                       median_burned_area = median(area, na.rm = TRUE),
                       distinct_month_cnt = n_distinct(area, na.rm=TRUE))

ggplot(data=X_Y_fires, aes(x=X, y=Y)) +
  geom_point(aes(size=num_of_fires, color=total_burned_area)) +
  scale_x_continuous(breaks=1:9) +
  scale_y_continuous(breaks=1:9) +
  scale_size_continuous(range=c(1,12)) +
  scale_color_gradient(high = "#132B43", low = "#56B1F7") +
```

```
labs(title = "Forest Fires by Location",
     x = "X Spatial Coordinate",
     y = "Y Spatial Coordinate",
     size = "Number of Fires",
     color = "Total Burned Area")
```



Although this plot does not talk about the trend across months, this gives us an understanding of where the fires occur spatially and gives us an indication of the total burned area across all months and fires for that spatial location. The size of the bubble indicates the number of fires and the color gradient indicates the total area burned (darker indicates more area). We can see that there are certain areas that have little or no fires while others that have lots of fires. We also observe that while more fires generally indicates more area burned, there are areas where there are many fires but little total area burned (large bubbles that are lighter in color)