EDAV Fall 2019 Probem Set 1

Luke Beasley (lcb2165) and Kliment Mamykin (km2770)

Read *Graphical Data Analysis with R*, Ch. 3

Grading is based both on your graphs and verbal explanations. Follow all best practices as discussed in class.

The datasets in this assignment are from the **ucidata** package which can be installed from GitHub. You will first need to install the `devtools` package if you don't have it:

```
install.packages("devtools")
```

then,

```
devtools::install_github("coatless/ucidata")
```

```
library(tidyverse)
library(ggplot2)
library(devtools)
devtools::install_github("coatless/ucidata")
library(ucidata)
library(ggthemes)
library(gridExtra)
library(nullabor)

# Declare color variables to be used in all plots
theme.fill = "lightskyblue2"
theme.color = "grey10"
```

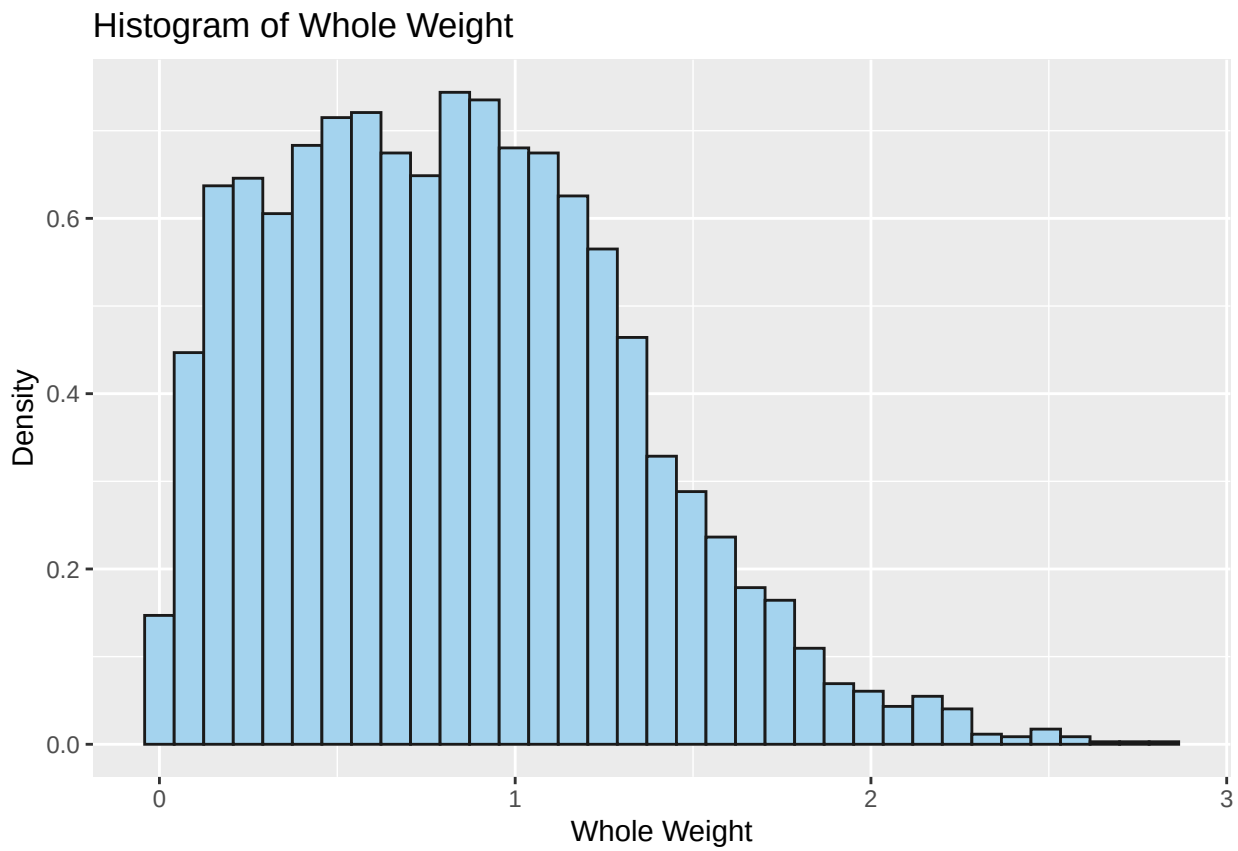
1. Abalone

[18 points]

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

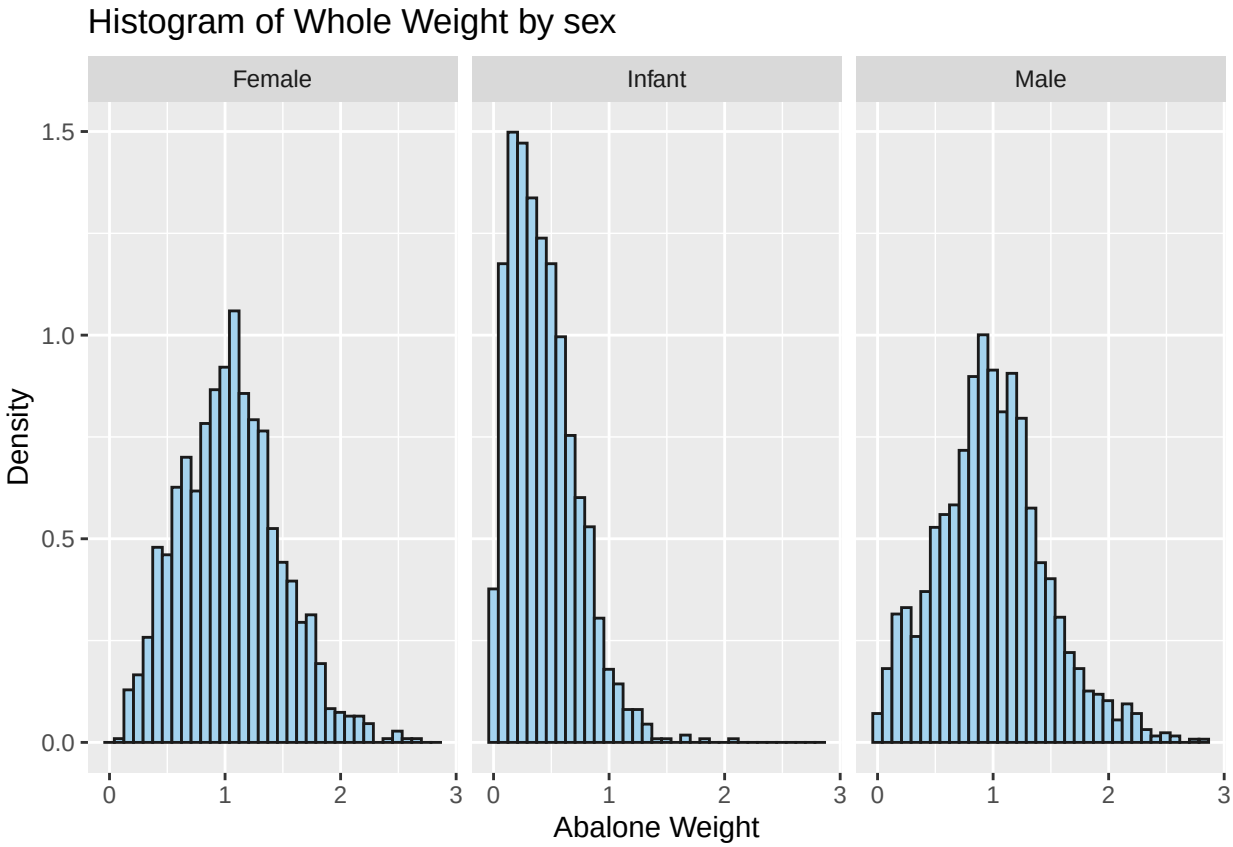
a) Plot a histogram of the variable.

```
ggplot(data = abalone, aes(x=whole_weight, y =..density..)) +
  geom_histogram(bins = 35, color = theme.color, fill = theme.fill) +
  labs(x = "Whole Weight", y = "Density") +
  ggtitle("Histogram of Whole Weight")
```



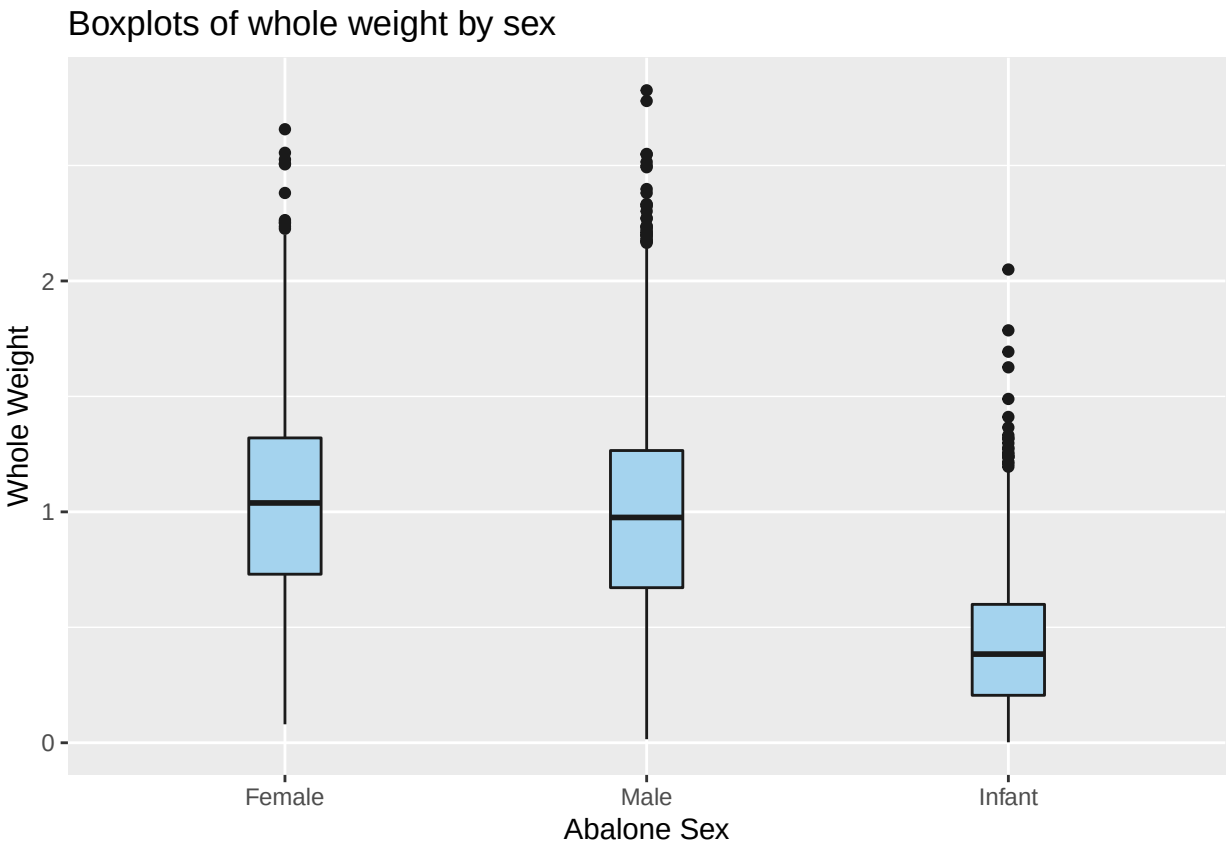
b) Plot histograms, faceted by `sex`, for the same variable.

```
sex.labs <- c("Female", "Infant", "Male")
names(sex.labs) <- c("F", "I", "M")

ggplot(data = abalone, aes(x=whole_weight, y =..density..)) +
  geom_histogram(bins = 35, color = theme.color, fill = theme.fill) +
  facet_wrap(~sex, labeller = labeller(sex = sex.labs)) +
  labs(x = 'Abalone Weight', y = "Density") +
  ggtitle("Histogram of Whole Weight by sex")
```



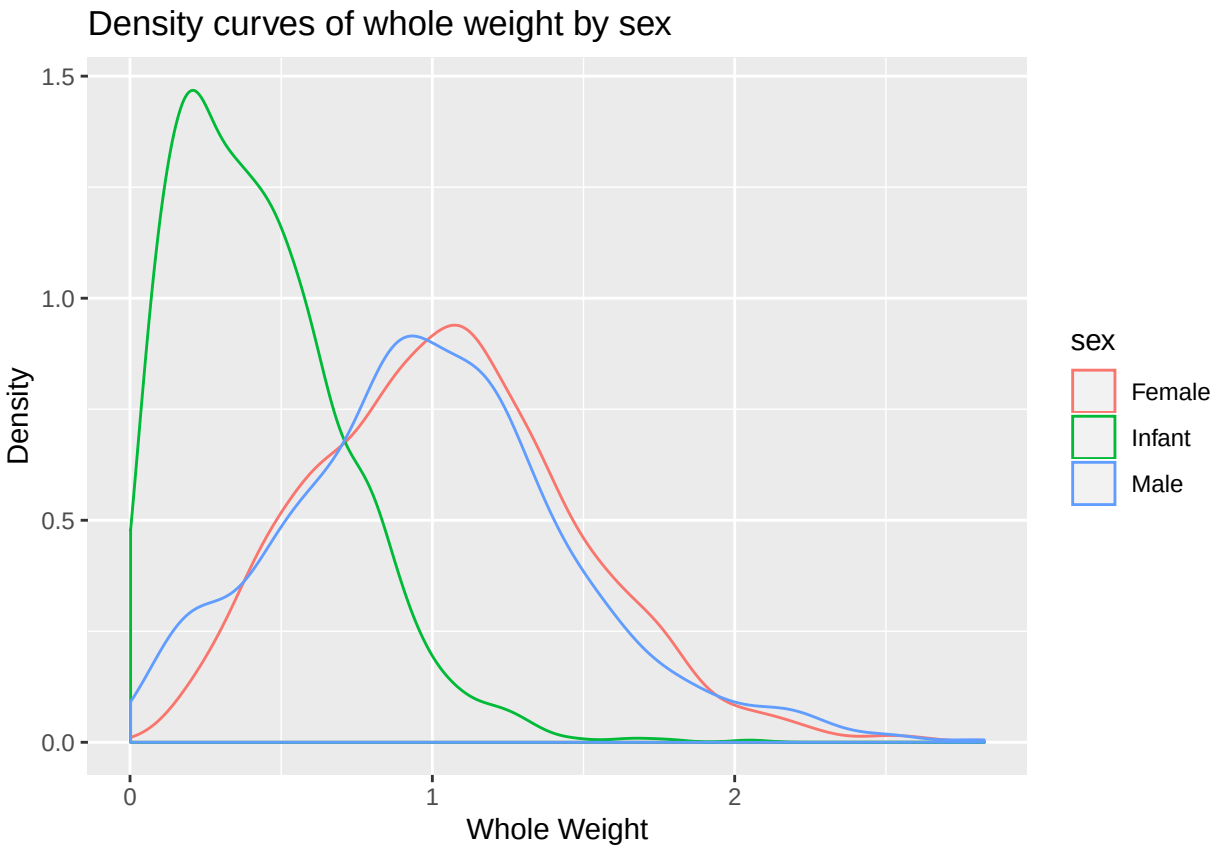
c) Plot multiple boxplots, grouped by `sex` for the same variable. The boxplots should be ordered by decreasing median from left to right.

```
ggplot(abalone, aes(x = reorder(sex, -whole_weight, median),y = whole_weight)) +
        geom_boxplot( width = 0.2, color = theme.color, fill = theme.fill) +
  labs(x='Abalone Sex', y = 'Whole Weight') +
  scale_x_discrete(labels = c("Female", "Male", "Infant")) +
  ggtitle("Boxplots of whole weight by 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(abalone) +
  geom_density(aes(x=whole_weight, y = ..density.., color=sex)) +
  labs(x="Whole Weight", y = "Density") +
  scale_color_discrete(labels = c("Female", "Infant", "Male")) +
  ggtitle("Density curves of whole weight by sex")
```



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?

Whole weight does not vary much between male and female abalones. However, infants weight is significantly lower than both male and female. Additionally, the infants weight varies less than both male and female. Due to biology, a lower weight for an infant is clearly to be expected. Less variance amongst infants also makes sense from the fact that infants have had less years to start to vary from each other.

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

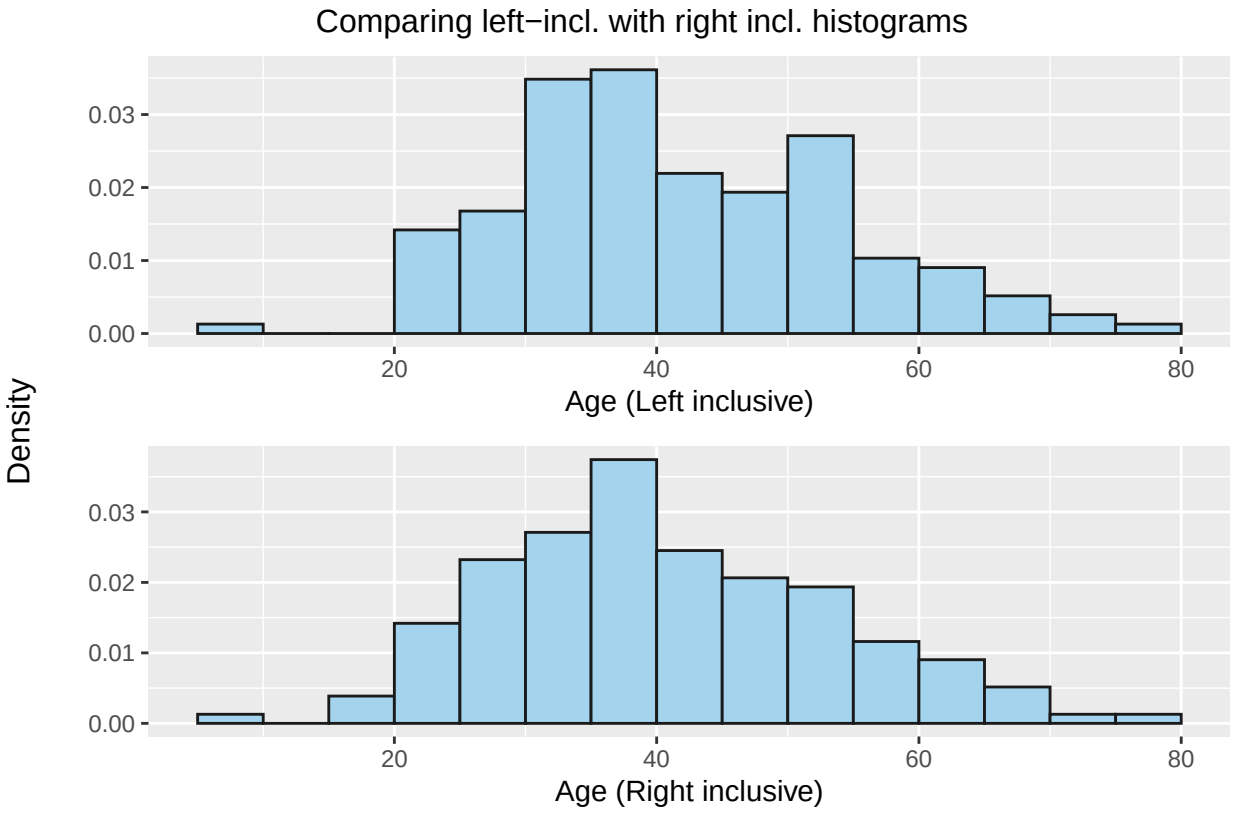
The measurements do not appear correct. Abalone shells appear to be much bigger than the data shows. Images of abalone shells will confirm that the shells are roughly similar size as a human hand. However, the data measure the shells in terms of millimeters. Most of the measurements in the dataset would have abalone shells be about the size of a penny. It may be that the data is just labeled wrong or the measurements may be wrong altogether.

2. Hepatitis

[6 points]

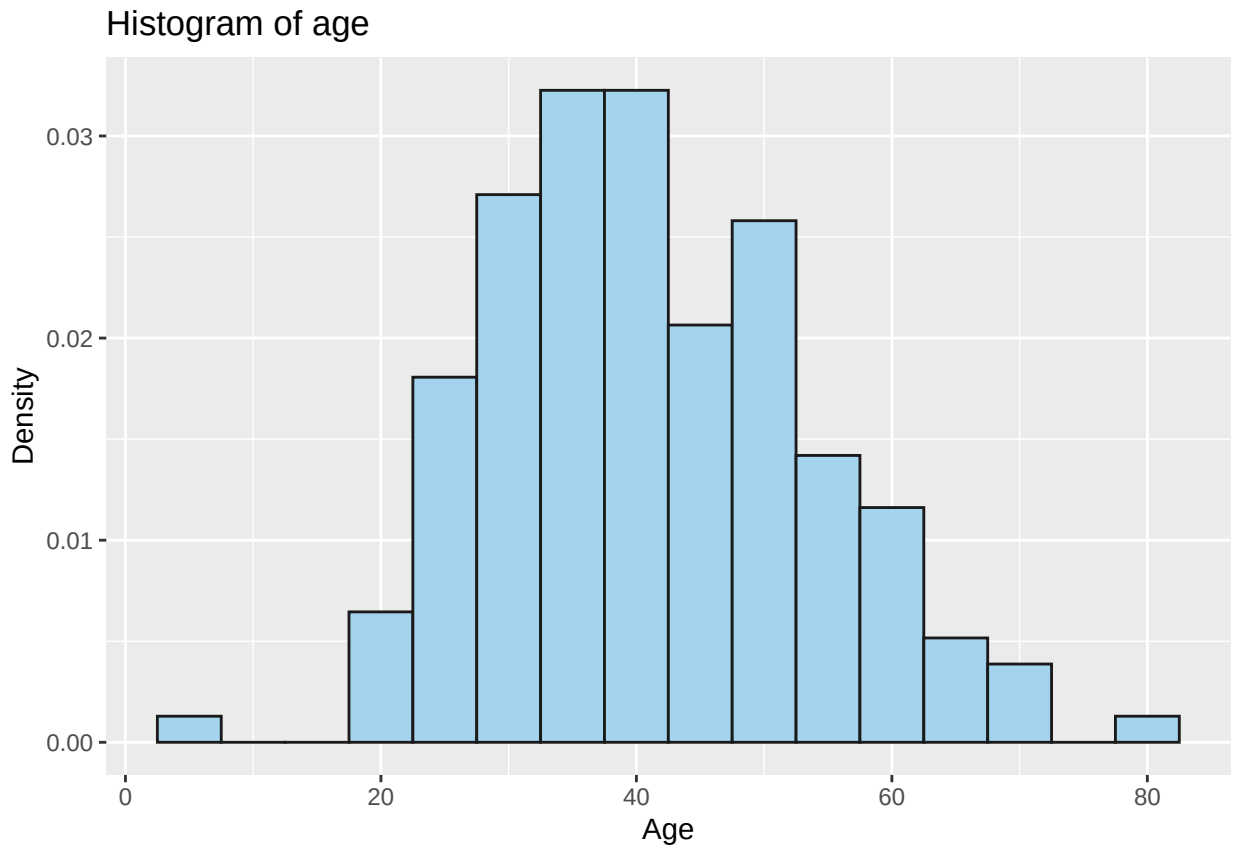
a) Draw two histograms of the age variable in the `hepatitis` dataset in the **ucidata** package, with binwidths of 5 years and `boundary = 0`, one right open and one right closed. How do they compare? one right open and one right closed. How do they compare?

```
g1 <- ggplot(hepatitis, aes(x=age, y=..density..)) +
  geom_histogram(binwidth =  5, closed = "left", boundary = 0, color = theme.color, fill = theme.fill) +
  labs(x='Age (Left inclusive)',y='')

g2 <- ggplot(hepatitis, aes(x=age, y=..density..)) +
  geom_histogram(binwidth  = 5, closed = "right", boundary = 0, color = theme.color, fill = theme.fill) +
  labs(x='Age (Right inclusive)',y='')

grid.arrange(g1,g2, top ='Comparing left-incl. with right incl. histograms', left = 'Density', bottom =
```



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(hepatitis, aes(x=age, y=..density..)) +
  geom_histogram(binwidth=5, boundary=2.5, color = theme.color, fill = theme.fill) +
  labs(x = "Age", y = "Density") +
  ggtitle("Histogram of age")
```



The parameters were chosen in order to smooth out any confusion about the inclusion of boundary ages. After examining the data, no data exists for ages under 2.5. Therefore, moving the boundary up to 2.5 and keeping the bins at 5 allows for a clear picture of the data.

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:

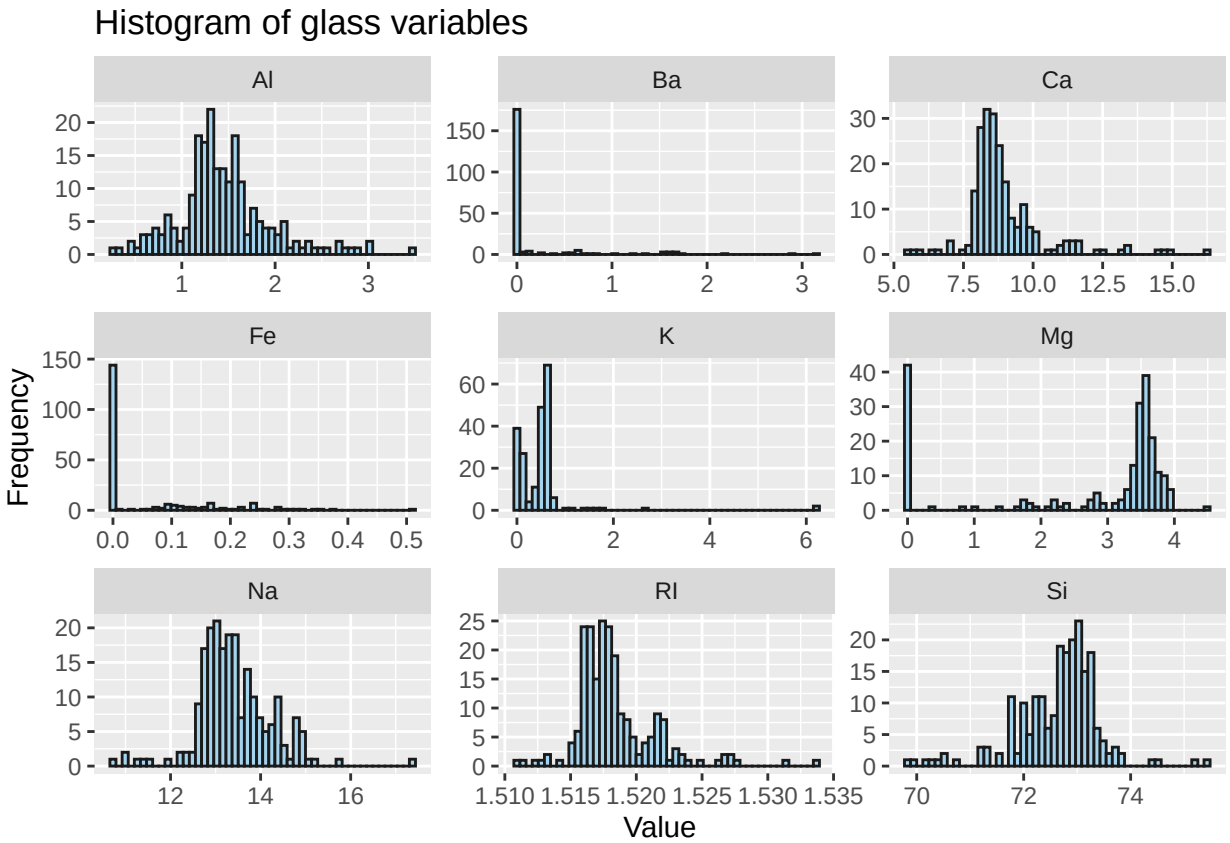
```
  variable   value
1       RI 1.52101
2       RI 1.51761
3       RI 1.51618
4       RI 1.51766
5       RI 1.51742
6       RI 1.51596
```

```
glass_variables = colnames(glass)[2:10]
glass_values <- glass %>%
  gather(key = "variable", value = "value", glass_variables) %>%
  select("variable", "value")
head(glass_values)
```

```
##   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?

```
ggplot(data = glass_values) +
  geom_histogram(bins = 50, aes(x = value), color = theme.color, fill = theme.fill) +
  facet_wrap(~variable, scales = "free") +
  labs(x = 'Value', y = "Frequency") +
  ggtitle("Histogram of glass variables")
```



It was hard to observe any pattern initially when using `facet_wrap()` without the `scales = "free"` parameter, due to the large range of values across features individual histograms were squeezed and did not properly visualize the undelying distribution of values.

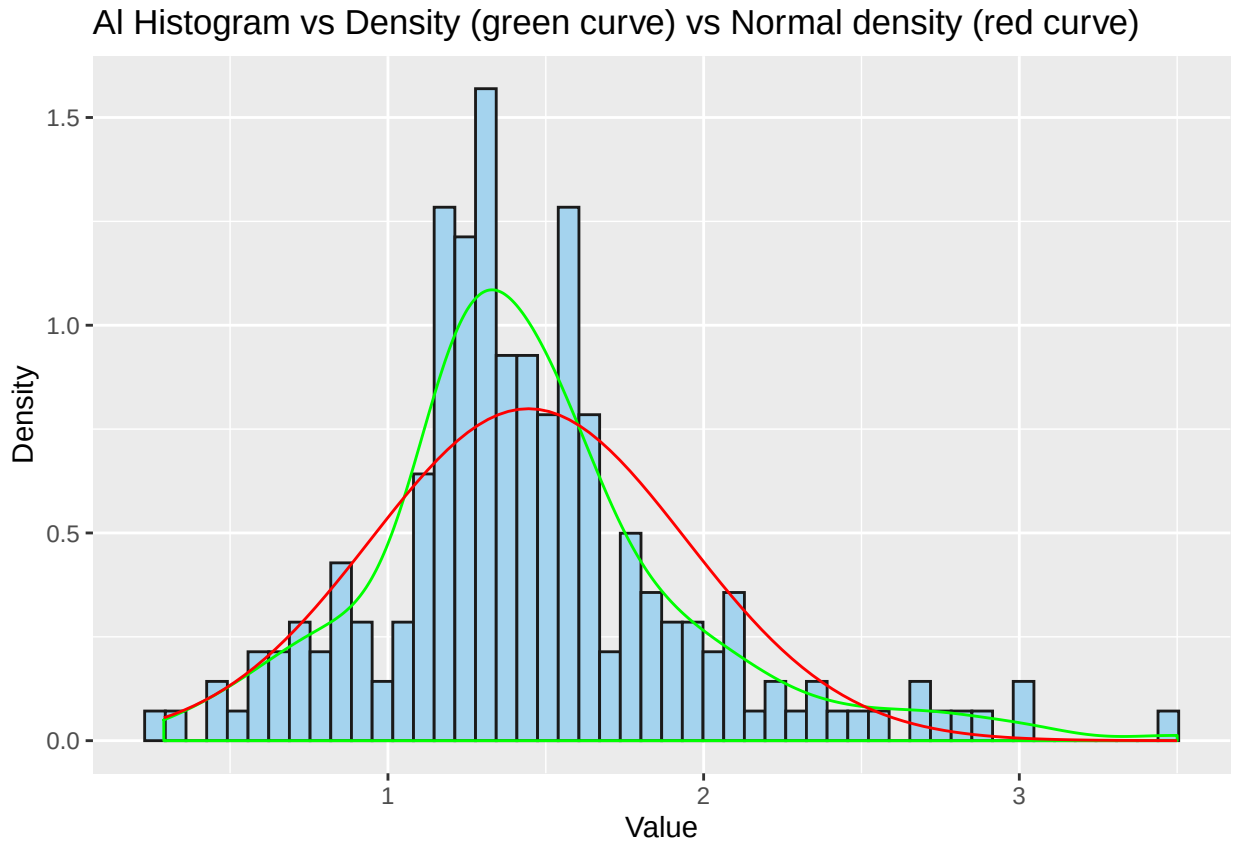
With `scales = "free"` parameter, the facets are scaled individually and the distribution of values is better visualized.

We can observe that variables Al, Ca, Na, Ri and Si are probably distributed normally, variables Ba and Fe tend to have value 0 with a long tail, variables K and Mg display dual mode.

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?

```
glass_al = filter(glass_values, variable == "Al")
ggplot(data = glass_al) +
  geom_histogram(bins = 50, aes(x = value, y = ..density..), color = theme.color, fill = theme.fill) +
  geom_density(bw = 0.15, aes(x = value, y = ..density..), color="green") +
  stat_function(fun = dnorm, args=list(mean=mean(glass_al$value), sd=sd(glass_al$value)), colour = "red
  labs(x = 'Value', y = "Density") +
  ggtitle("Al Histogram vs Density (green curve) vs Normal density (red curve)")
```



We observe that the distribution of Al values roughly follows normal distribution, however the density of the values around the mean is higher and the density on the slopes is lower then the normal density.

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

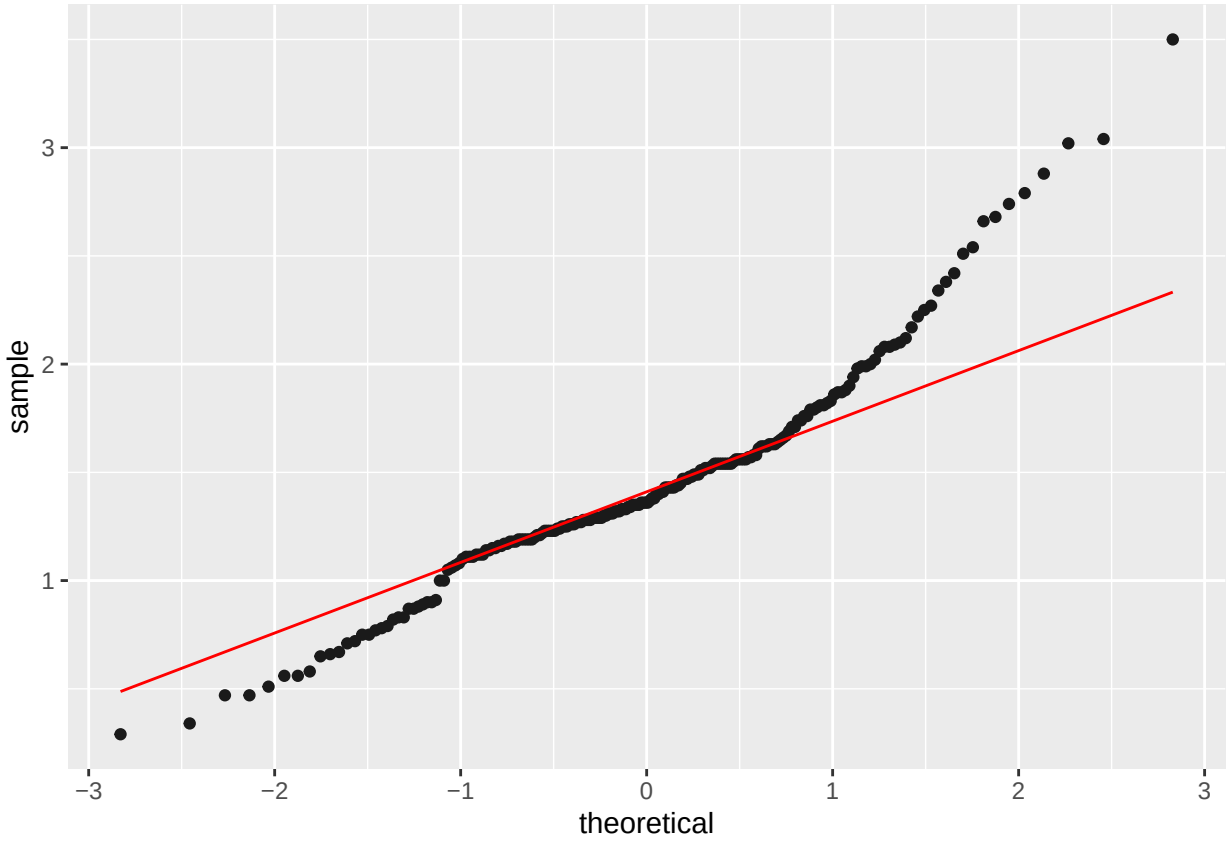
```
shapiro.test(glass_al$value)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  glass_al$value
## W = 0.94341, p-value = 2.083e-07
```

p-value is significantly less then an adequate threshold of 0.1, and therefore we reject the null hypothesis that the data is normally distributed.

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

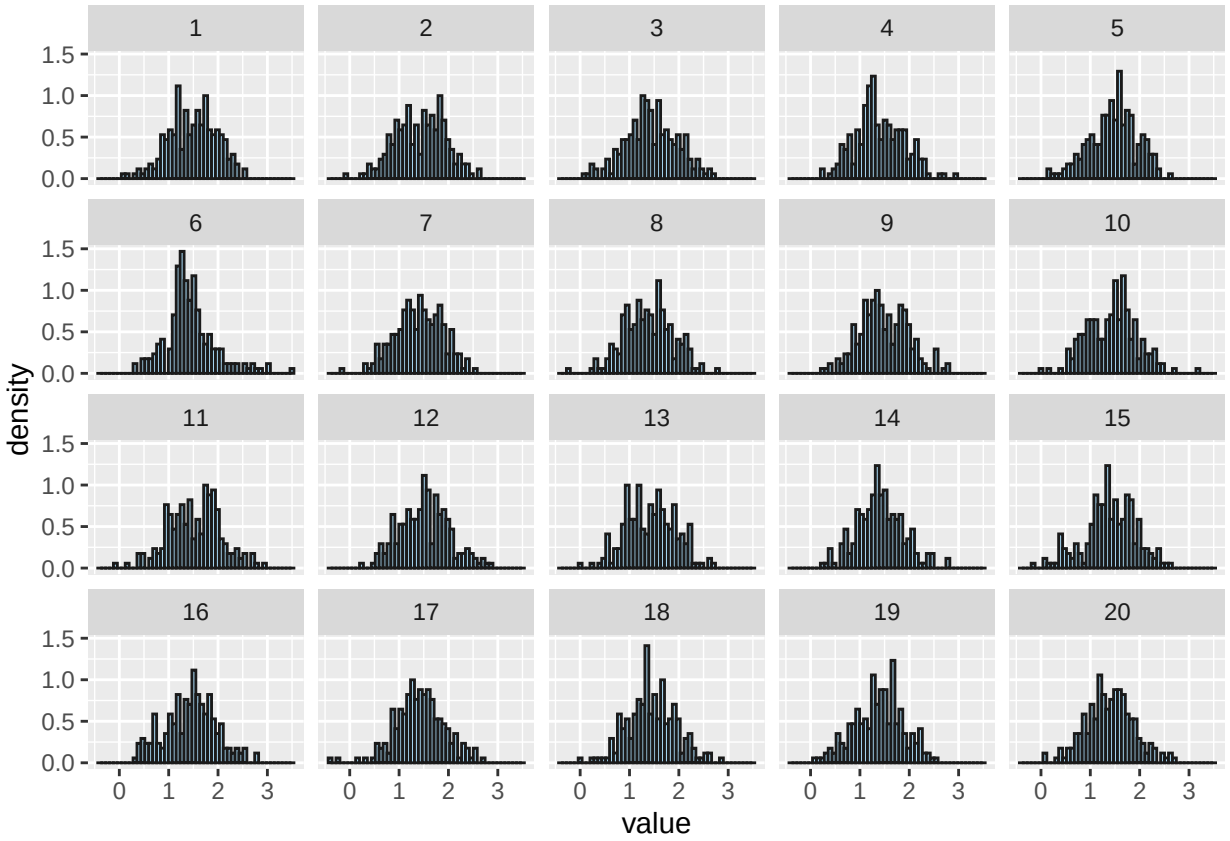
```
ggplot(data = glass_al, aes(sample = value)) +
  geom_qq(color = theme.color) +
  geom_qq_line(color = "red")
```



It is apparent from the plot that the sample is not normally distributed. The tails of the sample are off.

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?

```
d <- lineup(null_dist("value", dist = "normal"), glass_al)
ggplot(data = d, aes(x = value, y = ..density..)) +
  geom_histogram(bins = 50, color = theme.color, fill = theme.fill) +
  facet_wrap(~ .sample)
```



I have been able to correctly select the real data histogram from several lineups. The null hyposesis normal histograms are spread out more vs real histogram narrow peak in the middle and low sholders.

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?

A third-party observer was able to correctly identify the real data distribution. That is another indication that the distribution of the variable is different from the normal distribution.

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

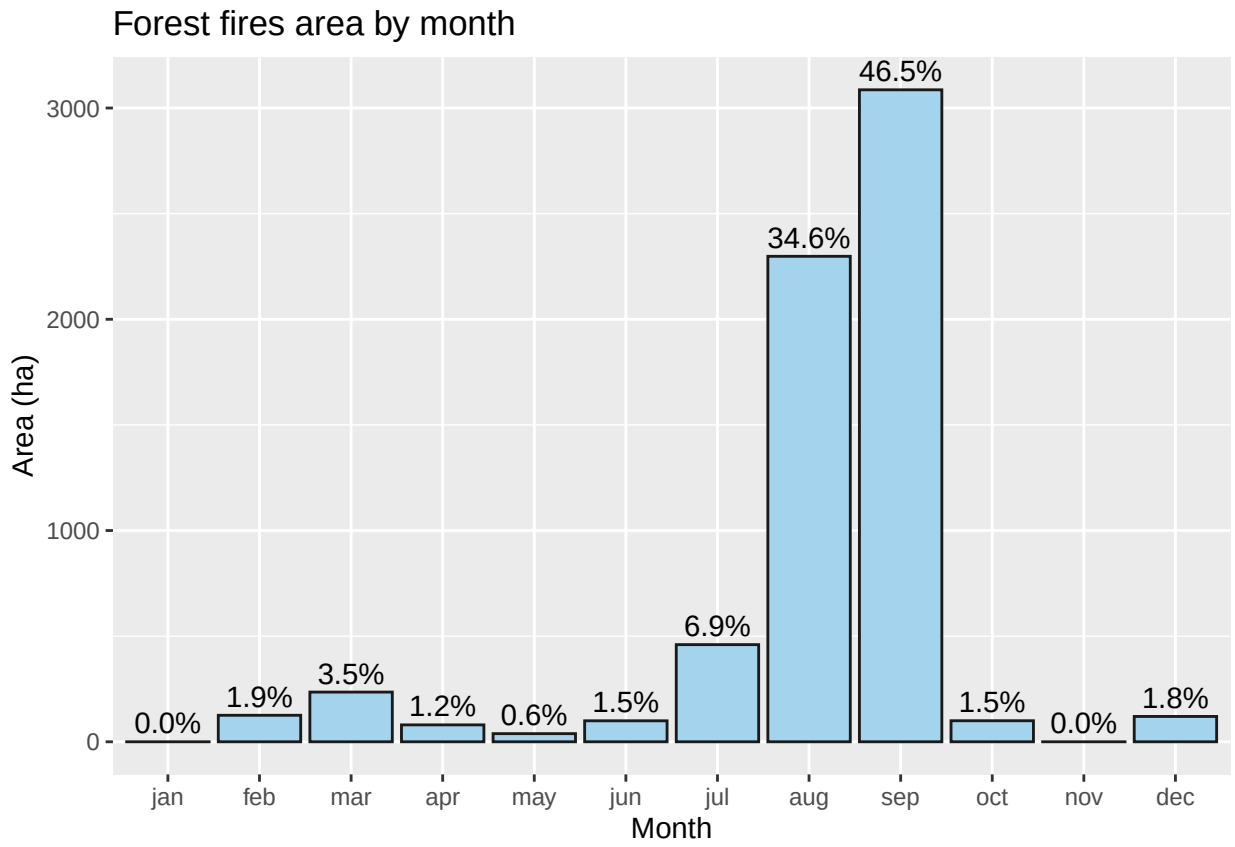
All methods produce the same result, which is that we can not make an assumption that the Al variable of the dataset is normally distributed. By the initial visual inspection it appeared to be the most promising candidate for a normal distribution. The histogram with a normal curve overlay seems to be the least reliable method to test for normality, as it's hard to judge if deviation from the curve are artifacts from binning, or small sample. Shapiro-Wilk normality test, QQ-plot and the lineup must be used to supplement the histogram in determination if the normality assumption can be made.

4. Forest Fires

[8 points]

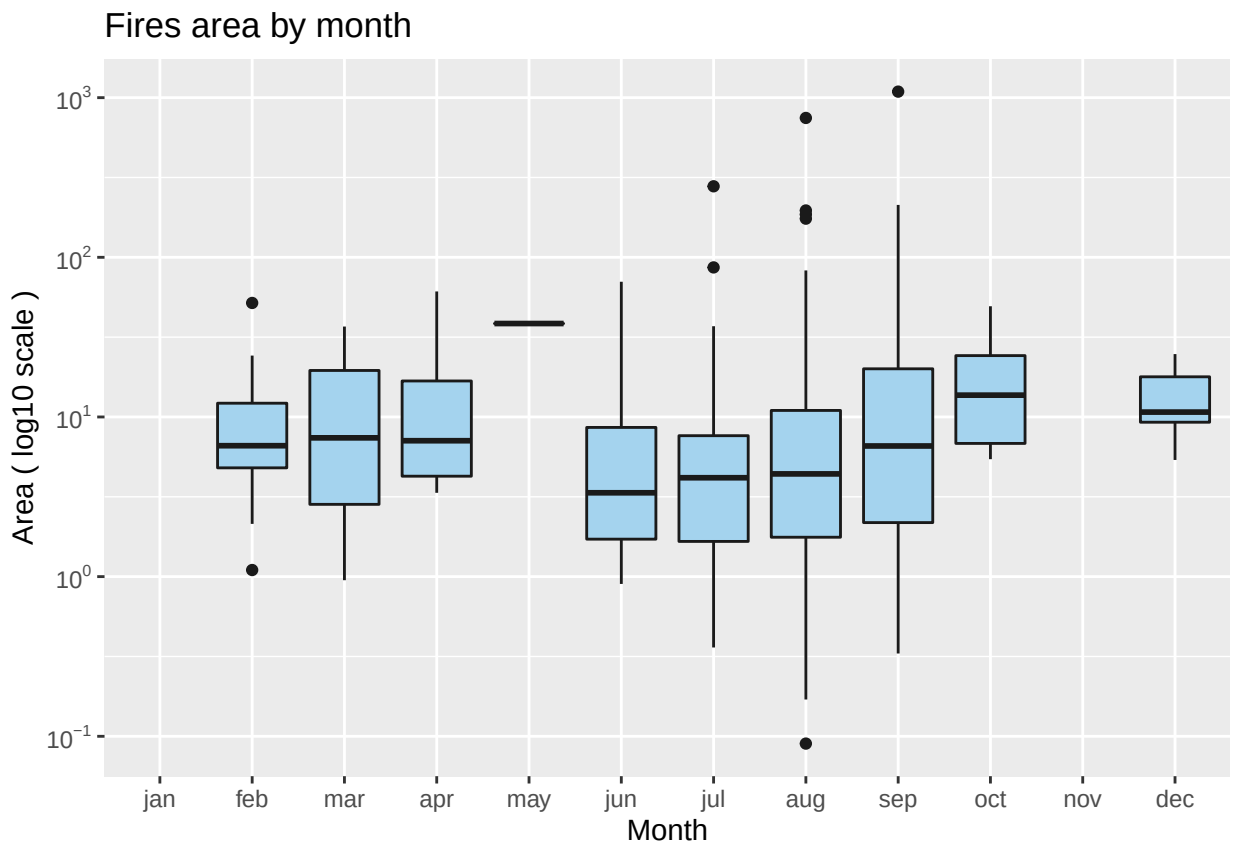
Using the `forest_fires` dataset in the **ucidata** package, analyze the burned area of the forest by month. Use whatever graphical forms you deem most appropriate. Describe important trends.

```r
# Re-order month and day factors in the natural (not alpha) order to make sense in visualizations
forest_fires <- forest_fires %>%
  mutate(month = factor(forest_fires$month, levels = c("jan", "feb", "mar", "apr", "may", "jun", "jul",
  # day is not actually used in this analysis, but was useful in the exploration of the dataset
  mutate(day = factor(forest_fires$day, levels = c("mon", "tue", "wed", "thu", "fri", "sat", "sun")))

ffstats <- forest_fires %>%
  # Calculate stats per month
  group_by(month) %>%
  summarise(
    count = n(), # also not used in this particular visualization
    area = sum(area),
    pct_of_total = sum(area) / sum(forest_fires$area)
  )

ggplot(data = ffstats, aes(x = month, y = area)) +
  geom_bar(stat = "identity", color = theme.color, fill = theme.fill) +
  geom_text(aes(label = scales::percent(pct_of_total)), position = position_dodge(width = 0.9),  vjust =
  labs(x = "Month", y = "Area (ha)") +
  ggtitle("Forest fires area by month")
```



The bar chart highlights that during the months of August and September the forests are most affected by fires, and about 81% of the total burned area is burned during those months.

```
ggplot(data = forest_fires) +
  geom_boxplot(aes(x = month, y = area), color = theme.color, fill = theme.fill) +
  scale_y_log10(labels = scales::trans_format("log10", scales::math_format(10^.x))) +
  labs(x = "Month", y = "Area ( log10 scale )") +
  ggtitle("Fires area by month")
```



This boxplot gives a better understanding of the distribution of fire areas, but the downside is that we have to use a log10 scale since the median of all fires is 0.52ha and the outliers that boxplot is designed to highlight are $>$ 100 ha.

It appears that the months of July, August and Septenber were affected by large outliers, but on average the median size of fire is smaller then (for example) in October. Depending on the goal of analysis we may need to exclude the outliers to obtain a clear picture of the distribution and correlation of fires.

Appendix

Below is a standard pairplot of all variables in the forest_fires dataset.

```
plot(forest_fires)
```