06-23856 Evaluation Methods and Statistics
Autumn Semester 2019

Lab Class 3: The Basics of Using R

1 Open RStudio

Last week we looked at the basics of using R and creating dataframes. For the rest of the lab classes, we will be using RStudio – which uses the same command language but has some useful built-in features which makes it easier to use.

2 Open a data set

In a later lab class, we will explore ways on importing tab-separated data that we have collected from experiments. In this lab, we will use one of the data sets supplied with R. The data set 'airquality' contains a set of measurements from New York over 153 days.

```
>airquality
```

This command identifies the data set and opens it for you. There is a table with 6 headings (Ozone, Solar.R, Wind, Temp, Month, Day) and each line represents a separate reading. Where you see 'NA', this indicates missing values (perhaps there was no measurement taken that day or the sensor was faulty). You can inspect the data set by scrolling up and down, or by looking at the top and bottom of the data.

```
>head(airquality)
```

This shows the first 6 values (you can use the command 'tail' to look at the last 6 values)

To start working with the data set, we want to make sure that R knows that the column headings specifically for this data set will define variables that we want to use.

```
>attach(airquality)
```

Now we can use one of the column labels to look at the data in that column…

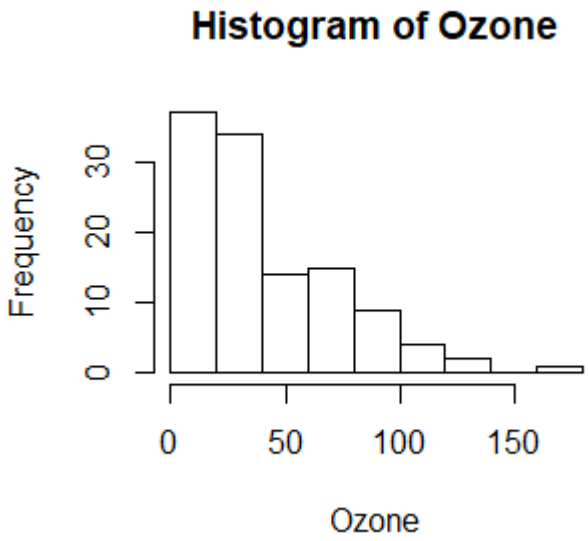
```
>Ozone
```

This will give a table only of values relating to 'Ozone'

It can be easier to see patterns in the data by creating a graph. Last week we used 'plot' to produce a scatterplot, but this would not be appropriate for a single column of data so we can create a Histogram.

```
>hist(Ozone)
```

Notice that this appears in the lower right-hand window (rather than a separate window as last week).



To find out more about creating and using histograms, look at the manual

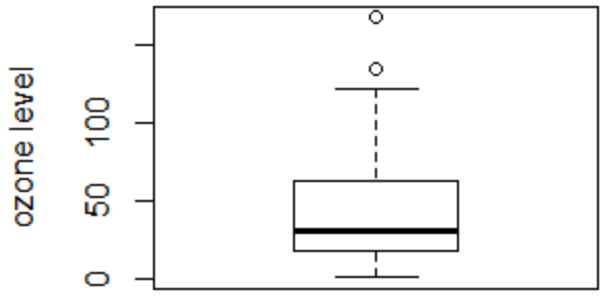
>help(hist)

This replaces the graph with the entry from the help files on 'histograms' (you can return to the graph by selecting the 'Plots' tab above this window). You could, for example, change the bar colours…

> hist(Ozone, col=rgb(0,1,0))

We could also graph the data as a boxplot:

> boxplot(Ozone,ylab="ozone level")



The graphs give us some idea about the nature of the data that we are using. It is always sensible to graph your data before doing any further analysis, just to get a feel for what you are dealing with.

We can summarise these data by asking R to return some summary values.

```
> summary(Ozone)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1.00   18.00   31.50   42.13   63.25  168.00      37
```

If we try to calculate mean and standard deviation, we find a problem.

```
> m = mean(Ozone)
> s = sd(Ozone)
> m
[1] NA
> s
[1] NA
```

This is because there are many values of 'NA' and this would interfere with any further analysis, so let's remove these and create a new set of data to work with.

```
> dataOzone<-Ozone[!is.na((Ozone))]
```

We can check the histogram again:

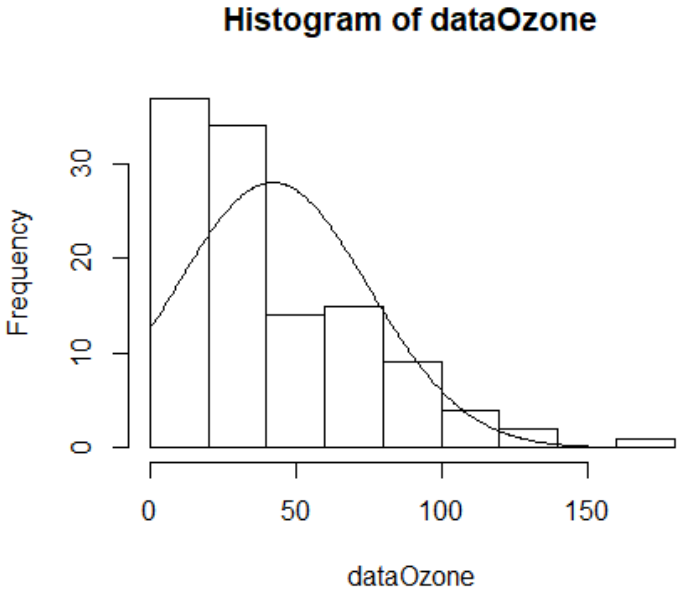
```
> hist(dataOzone)
```

We can also calculate mean and standard deviation correctly.

```
> m=mean(dataOzone)
> m
[1] 42.12931
> s= sd(dataOzone)
> s
[1] 32.98788
```

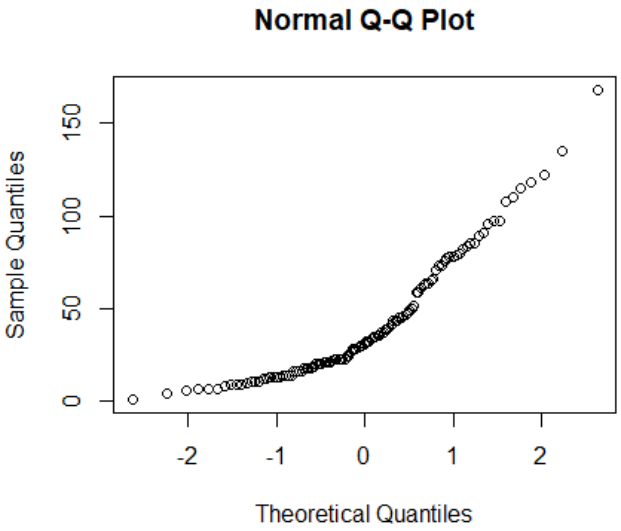
The standard deviation is very large and the histogram is skewed to the right. So, these data do not look as if they follow a normal distribution. We can check this further by plotting a normal distribution curve onto our historgram.

```
> hist(dataOzone)
> xs<-seq(min(dataOzone), max(dataOzone),0.5)
> ys<-dnorm(xs,m,s)*length(dataOzone)*20
> lines(xs,ys)
```



A more useful way of looking at this is as a Q-Q plot, where the quantiles of the data set are plotted against an ideal normal distribution. The closer the points lie to a straight line, the more likely to data have come from a normal distribution.

> qqnorm(dataOzone)



Finally, we can apply a Shapiro-Wilk test to these data. If the statistic is >0.05, then the data are normally distributed. We can see that this is not a normally distributed data set.

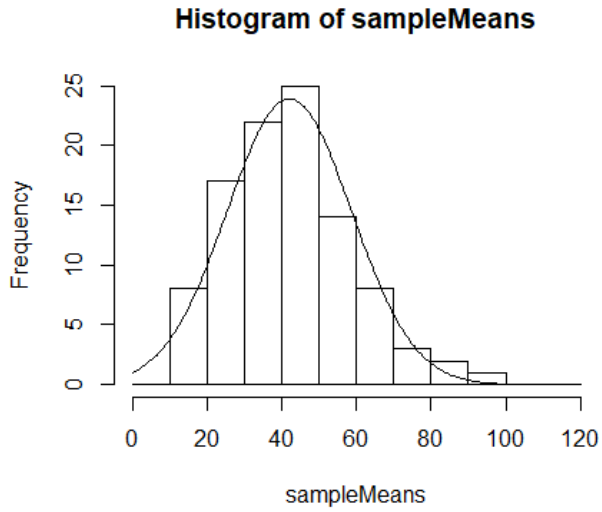
> shapiro.test(dataOzone)

Shapiro-Wilk normality test

data: dataOzone
W = 0.87867, p-value = 2.79e-08

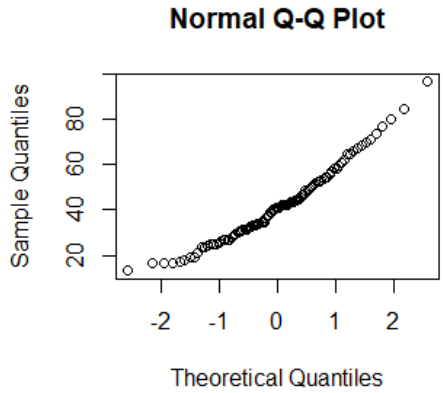
We could take a random sample from these data to see if this could produce a better distribution.

```
> numberOfSamples<-100
> sampleMeans<-numeric(numberOfSamples)
> for(i in 1:numberOfSamples) sampleMeans[i]<-mean(sample(dataOzone, sampleSize))
> hist(sampleMeans, breaks = seq(0, 120, 10), xlim=c(0,120))
> sampleMean<-mean(sampleMeans)
> sampleSD<-sd(sampleMeans)
> xs<-seq(0,100,0.5)
> ys<-dnorm(xs, sampleMean, sampleSD)*numberOfSamples*10
> lines(xs,ys)
```



So, this looks better. But, let's make a Q-Q plot and run a Shapiro Wilk test to check…

```
> qqnorm(sampleMeans)
```




```
> shapiro.test(sampleMeans)
```

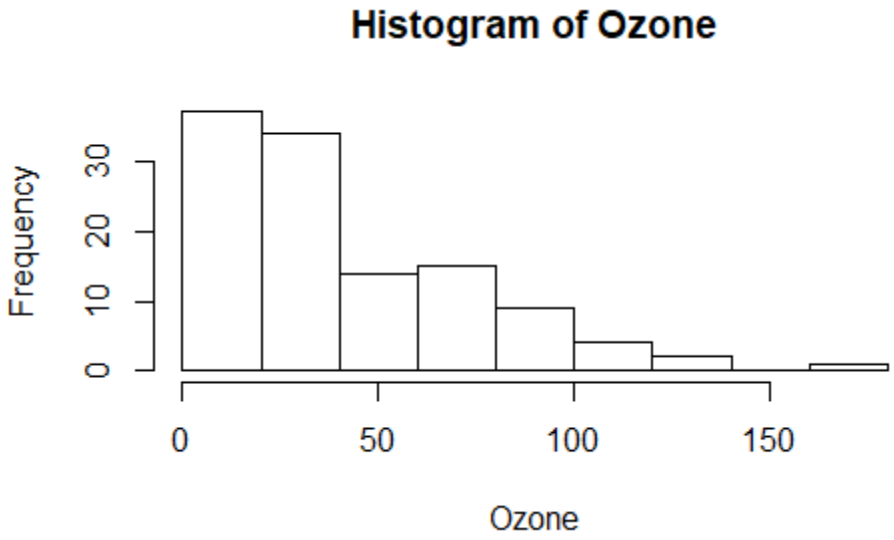
Shapiro-Wilk normality test

data: sampleMeans
W = 0.9655, p-value = 0.01011

TRANSFORMING DATA

If we look at the histogram of the Ozone data set we can get an idea of what is happening:

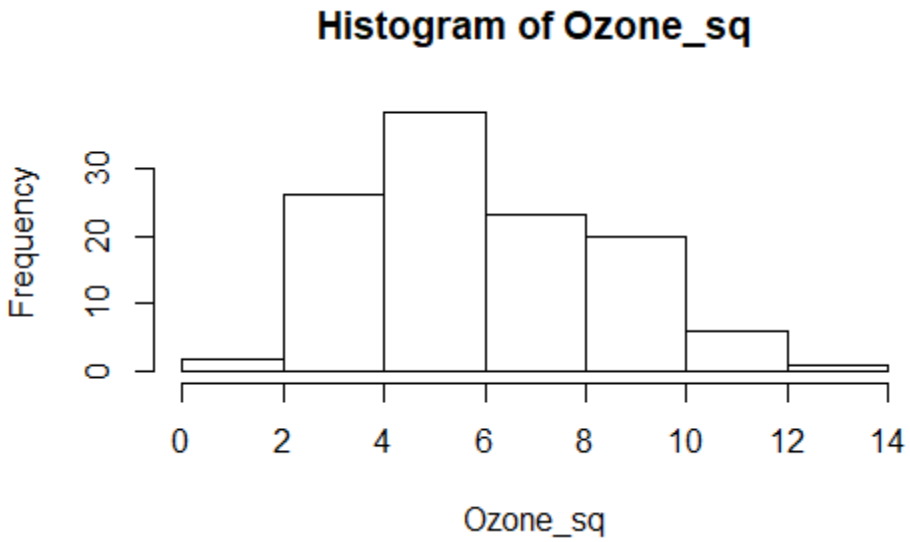
```
> hist(Ozone)
```



Before turning to non-parametric tests, we could transform the data to see if we can convert it into a normal distribution. There are several transforms we could apply using R.

i. Square-root transform
We could take the square root of the data, on the assumption that this might address variance.

```
> Ozone_sq=sqrt(Ozone)
> hist(Ozone_sq)
```



This looks to be an improvement (in terms of bringing this towards a normal distribution).

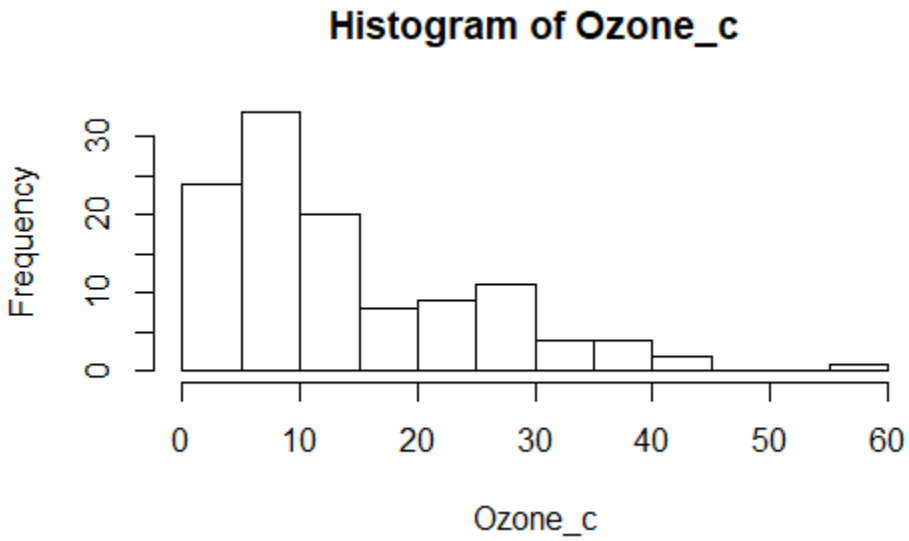
```
> shapiro.test(Ozone_sq)

        Shapiro-Wilk normality test
```

```
data:  Ozone_sq
W = 0.9655, p-value = 0.004408
```

But this fails the test of normal distribution.

iii. Cube-root transform
We could take the cube root…

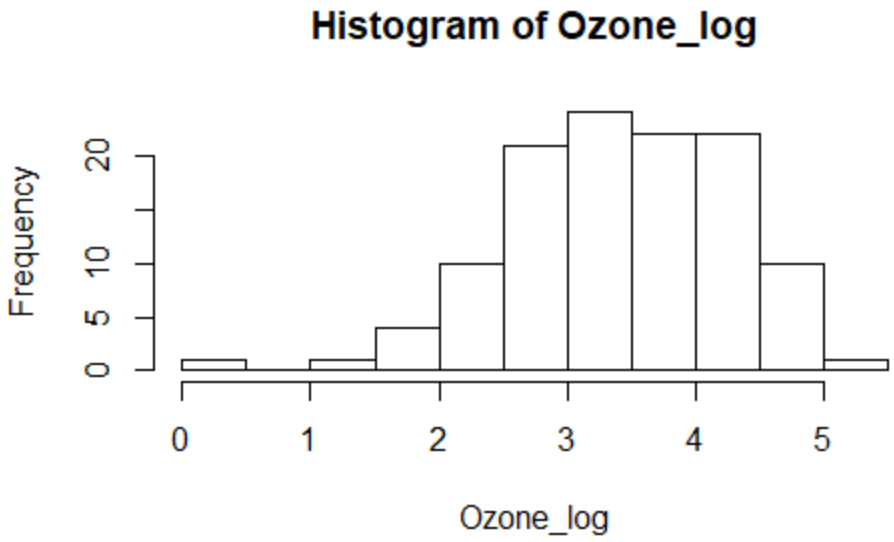
```
> Ozone_c=sign(Ozone)*abs(Ozone^1/3)
> hist(Ozone_c)
```



But that is not helpful…

iii. Log Transform
A Log Transform is fairly easy to imagine (you are simply taking the logarithm, base10, of each value in order to place the data on a uniform scale).

```
> Ozone_log=log(Ozone)
> hist(Ozone_log)
```




```
> shapiro.test(Ozone_log)

        Shapiro-Wilk normality test

data:  Ozone_log
W = 0.97168, p-value = 0.01471
```

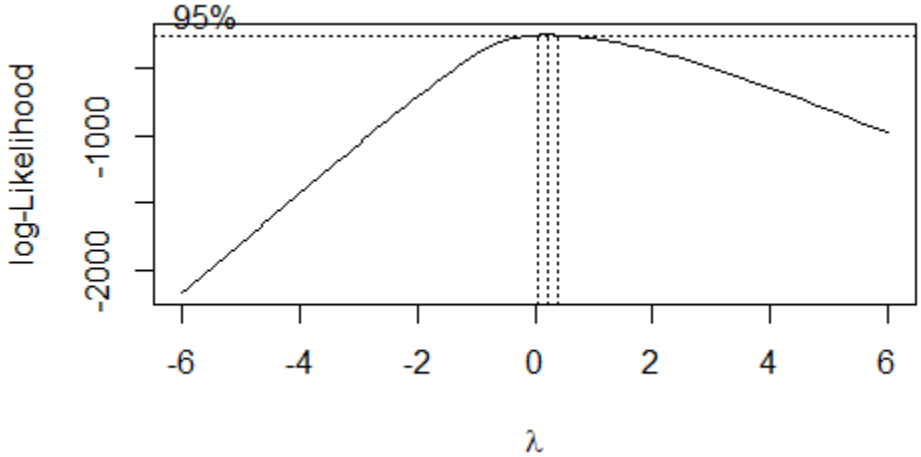
This is a little better, but still doesn't help

iv. Box-Cox transform
A common technique, Box-Cox, uses a function called, in R, from library called MASS.

```
> library(MASS)
```

```
> box=boxcox(Ozone~1, lambda=seq(-6,6,0.1))
```

This makes a single vector bounded by values -6 to +6 and produces a graph to show the distribution. The graph looks like it is following a normal distribution. But we need to apply a transformation factor to produce a better fit.



We need to create a new data frame to work with:

```
> cox = data.frame(box$x, box$y)
```

Now we reorder the data in terms of decreasing values of y:

```
> cox2 = cox [with(cox, order(-cox$box.y)),]
```

Then ask for the highest log-likelihood value in this distribution:

```
> cox2 [1,]
```

```
   box.x     box.y
63   0.2 -255.7666
```

We use this as our new value for lambda:

```
> lambda = cox2[1, "box.x"]
```

And defined the Box-Cox transformed data using this value of lambda applied to our original set:

```
> Ozone_boxcox=(Ozone^lambda-1)/lambda

> shapiro.test(Ozone_boxcox)

        Shapiro-Wilk normality test

data:  Ozone_boxcox
W = 0.98714, p-value = 0.3399
```

This produces a normal distribution, and we can now apply a parametric test to our data.