

# Normal Distribution Tutorial

2022-11-16

## Learning Goals

- Visualize properties of the normal distribution.
- Understand the Central Limit Theorem.
- Calculate sampling properties of sample means.
- Decide whether a data set likely comes from a normal distribution

## Tutorial

This lab mainly focuses on some exercises to better understand the nature of the normal distribution. We will also learn a couple of tools that help us decide whether a particular data set is likely to have come from population with an approximately normal distribution.

Many statistical tests assume that the variable being analyzed has a normal distribution. Fortunately, many of these tests are fairly robust to this assumption—that is, they work reasonably well even when this assumption is not quite true, especially when sample size is large. Therefore it is often sufficient to be able to assess whether the data come from a distribution whose shape is even approximately normal (the bell curve).

## Draw a histogram

A good way to start is to simply visualize the frequency distribution of the variable in the data set by drawing a histogram. Let's use the age of passengers on the Titanic for our example.

```
#read in titanic
titanicData <- read.csv("input_files/titanic.csv", stringsAsFactors = TRUE)
#have a look at the data
head(titanicData)
```

```
##   passenger_class      name    age  embarked
## 1             1st Allen, Miss Elisabeth Walton 29.0000 Southampton
## 2             1st Allison, Miss Helen Loraine  2.0000 Southampton
## 3             1st Allison, Mr Hudson Joshua Creighton 30.0000 Southampton
## 4             1st Allison, Mrs Hudson J.C. (Bessie Waldo Daniels) 25.0000 Southampton
## 5             1st Allison, Master Hudson Trevor  0.9167 Southampton
## 6             1st Anderson, Mr Harry 47.0000 Southampton
##           home_destination  sex survive
## 1           StLouis, MO female    yes
## 2 Montreal, PQ/Chesterville, ON female    no
## 3 Montreal, PQ/Chesterville, ON  male    no
## 4 Montreal, PQ/Chesterville, ON female    no
```

```
## 5 Montreal,PQ/Chesterville,ON   male   yes
## 6                               NewYork,NY   male   yes
```

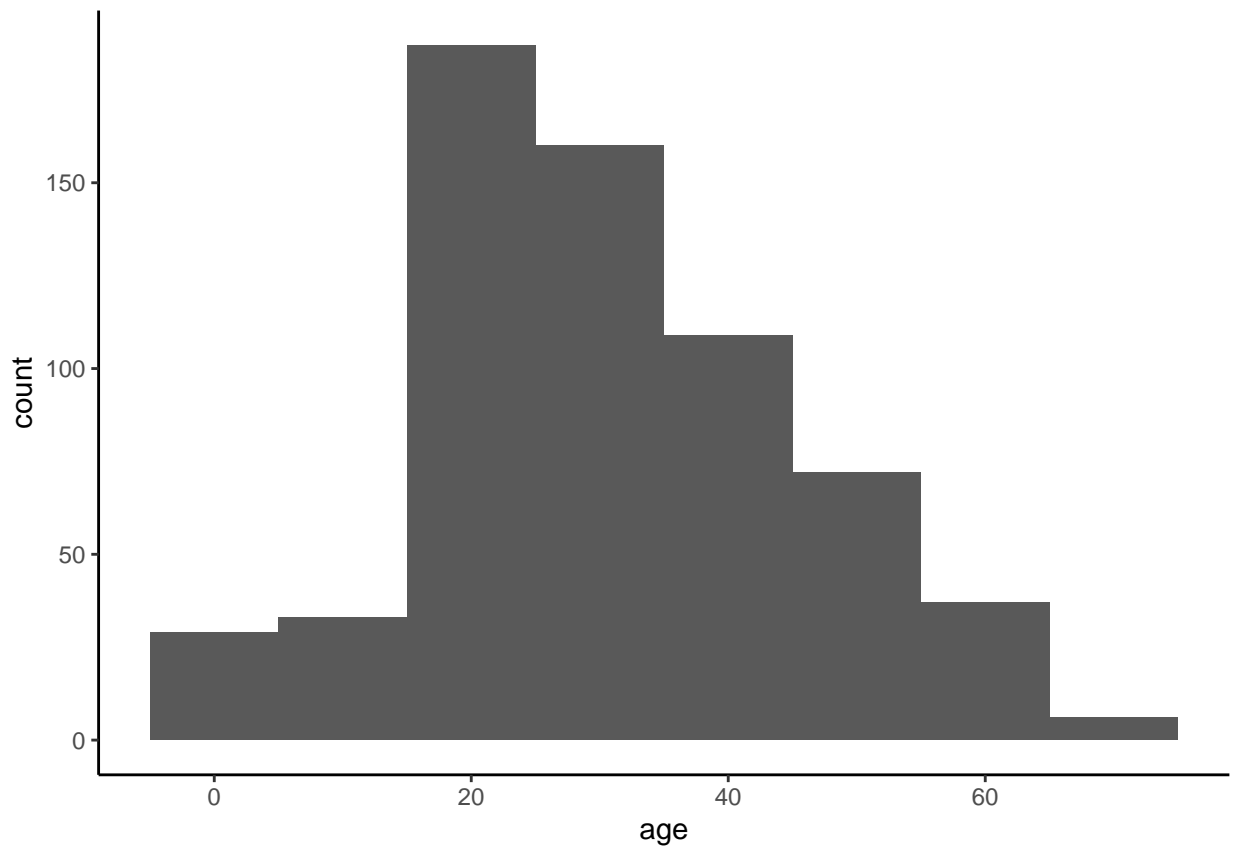
```
str(titanicData)
```

```
## 'data.frame':   1313 obs. of  7 variables:
##  $ passenger_class : Factor w/ 3 levels "1st","2nd","3rd": 1 1 1 1 1 1 1 1 1 1 ...
##  $ name             : Factor w/ 1310 levels "Abbing,MrAnthony",...: 22 25 26 27 24 31 45 46 50 54 ...
##  $ age              : num  29 2 30 25 0.917 ...
##  $ embarked        : Factor w/ 4 levels "", "Cherbourg",...: 4 4 4 4 4 4 4 4 4 2 ...
##  $ home_destination: Factor w/ 372 levels "", "?Havana,Cuba",...: 314 234 234 234 234 241 165 26 24 23 ...
##  $ sex              : Factor w/ 2 levels "female","male": 1 1 2 1 2 2 1 2 1 2 ...
##  $ survive          : Factor w/ 2 levels "no","yes": 2 1 1 1 2 2 2 1 2 1 ...
```

Remember we can use `ggplot()` to draw histograms.

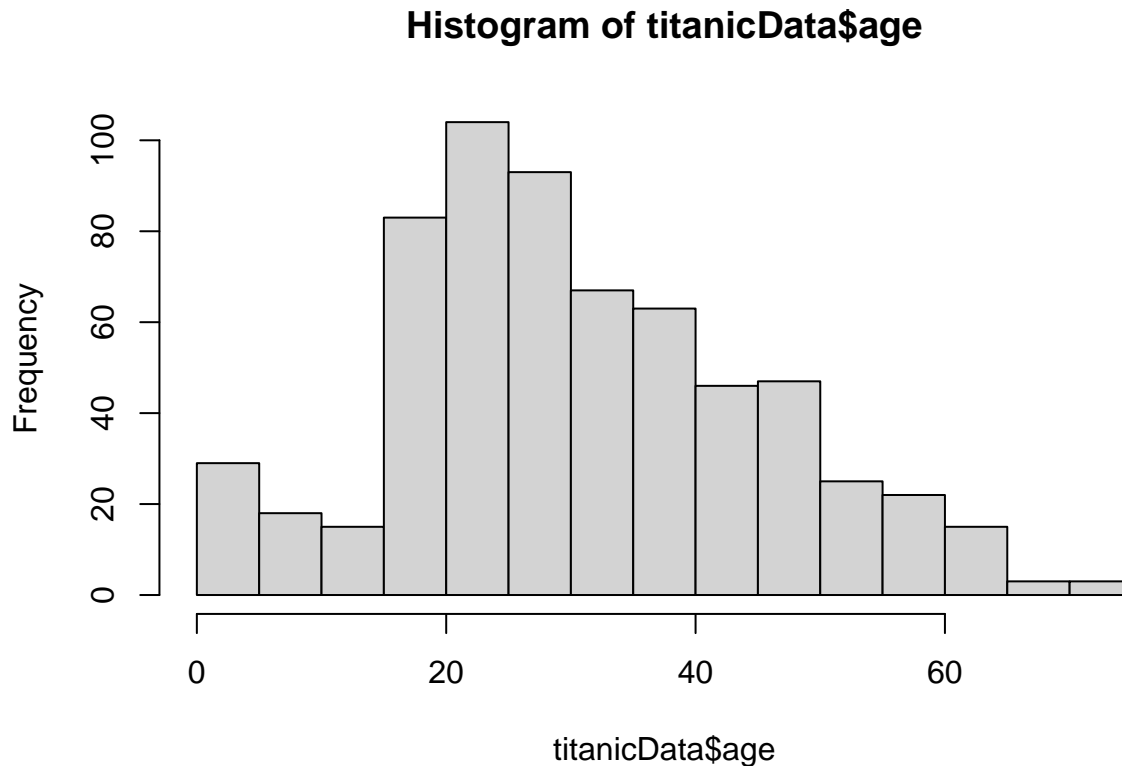
```
#use ggplot to make a histogram of age
ggplot(titanicData, aes(x = age)) +
  geom_histogram(binwidth = 10) +
  theme_classic()
```

```
## Warning: Removed 680 rows containing non-finite values (`stat_bin()`).
```



If we are just drawing a histogram for ourselves to better understand the data, it is even easier to just use a function from base R, `hist()`. Give `hist()` a vector of data as input, and it will print a histogram in the plots window.

```
hist(titanicData$age)
```



Looking at this histogram, we see that the frequency distribution of the variable is not exactly normal; it is slightly asymmetric and there seems to be a second mode near 0. On the other hand, like the normal distribution, the frequency distribution has a large mode near the center of the distribution, frequencies mainly fall off to either side, and there are no outliers. This is close enough to normal that most methods would work fine.

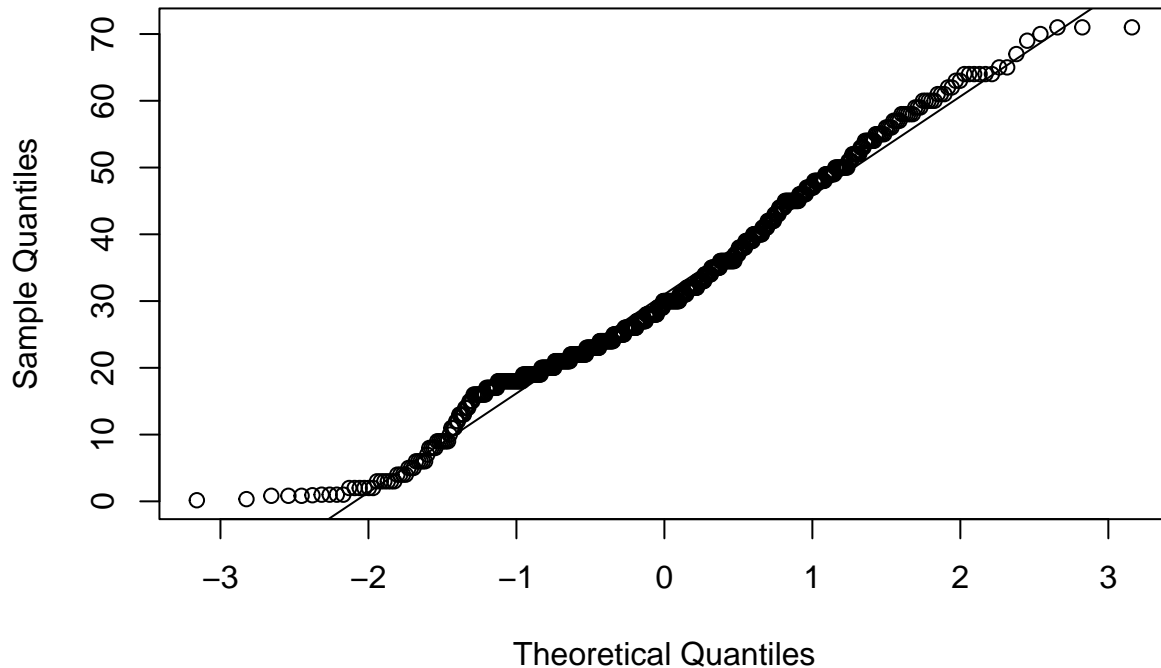
### QQ plot

Another graphical technique that can help us visualize whether a variable is approximately normal is called a quantile plot (or a QQ plot). The QQ plot shows the data on the vertical axis ranked in order from smallest to largest (“sample quantiles” in the figure below). On the horizontal axis, it shows the expected value of an individual with the same quantile if the distribution were normal (“theoretical quantiles” in the same figure). The QQ plot should follow more or less along a straight line if the data come from a normal distribution (with some tolerance for sampling variation).

QQ plots can be made in R using a function called `qqnorm()`. Simply give the vector of data as input and it will draw a QQ plot for you. (`qqline()` will draw a line through that Q-Q plot to make the linear relationship easier to see.)

```
qqnorm(titanicData$age)
qqline(titanicData$age)
```

## Normal Q-Q Plot



This is what the resulting graph looks like for the Titanic age data. The dots do not land along a perfectly straight line. In particular the graph curves at the upper and lower end. However, this distribution definitely would be close enough to normal to use most standard methods, such as the t-test.

It is difficult to interpret QQ plots without experience. One of the goals of today's exercises will be to develop some visual experience about what these graphs look like when the data is truly normal. To do that, we will take advantage of a function built into R to generate random numbers drawn from a normal distribution. This function is called `rnorm()`.

### `rnorm()`

The function `rnorm()` will return a vector of numbers, all drawn randomly from a normal distribution. It takes three arguments:

`n`: how many random numbers to generate (the length of the output vector)

`mean`: the mean of the normal distribution to sample from

`sd`: the standard deviation of the normal distribution

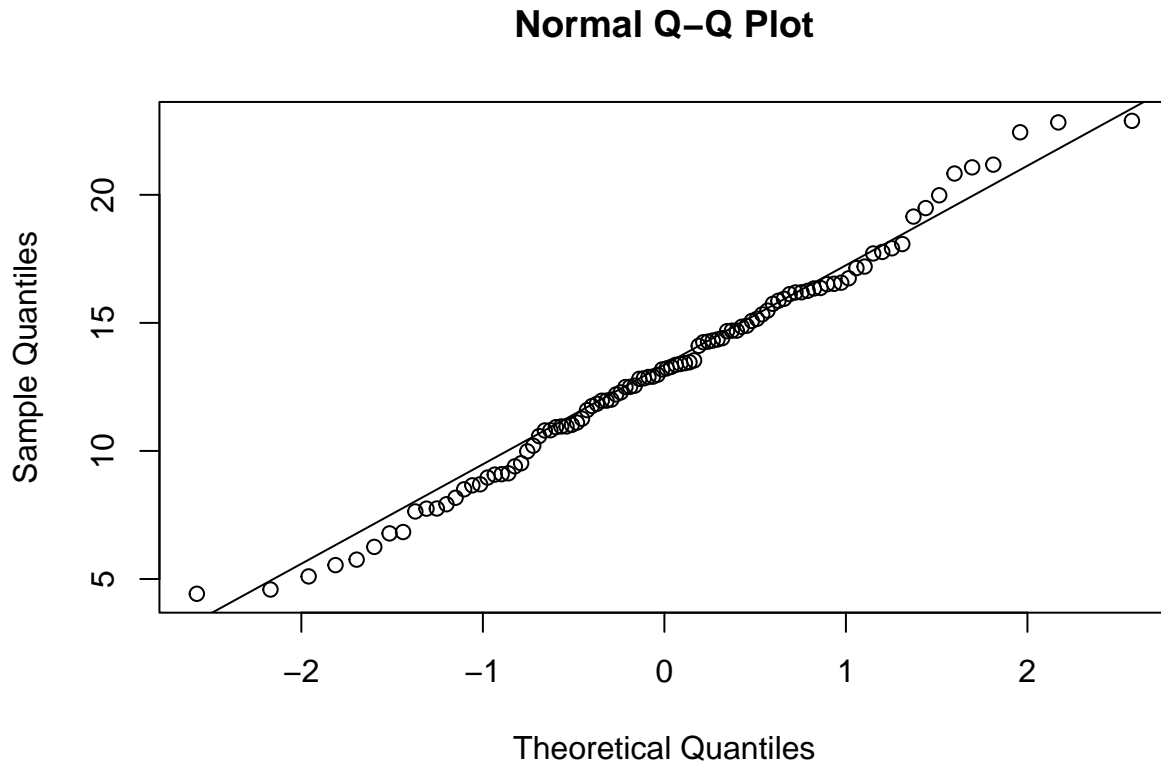
For example, the following command will give a vector of 20 random numbers drawn from a normal distribution with mean 13 and standard deviation 4:

```
rnorm(n = 20, mean = 13, sd = 4)
```

```
## [1]  5.512276 15.385273 13.679546  9.954855 10.062103 22.036931 19.794391
## [8]  5.982245 10.450542 16.120818 11.996709 11.371084 15.015185 15.430626
## [15] 10.963954 15.513197 20.908594 11.098613 11.471749 13.479485
```

Let's look at a QQ plot generated from 100 numbers randomly drawn from a normal distribution:

```
normal_vector <- rnorm(n = 100, mean = 13, sd = 4)
qqnorm(normal_vector)
qqline(normal_vector)
```



These points fall mainly along a straight line, but there is some wobble around that line even though these points were in fact randomly sampled from a known normal distribution. With a QQ plot, we are looking for an overall pattern that is approximately a straight line, but **we do not expect a perfect line**.

In the exercises, we'll simulate several samples from a normal distribution to try to build intuition about the kinds of results you might get.

**When data are not normally distributed, the dots in the quantile plot will not follow a straight line, even approximately.** For example, here is a histogram and a QQ plot for the population size of various counties, from the data in `countries.csv`. These data are very skewed to the right, and do not follow a normal distribution at all.

## Transformations

When data are not normally distributed, we can try to use a simple mathematical transformation on each data point to create a list of numbers that still convey the information about the original question but that may be better matched to the assumptions of our statistical tests. Chapter 13 of Whitlock and Schluter has more info on this, but for now let's learn how to do one of the most common data transformations, the log-transformation.

With a transformation, we apply the same mathematical function to each value of a given numerical variable for individual in the data set. With a log-transformation, we take the logarithm of each individual's value for a numerical variable.

- We can only use the log-transformation if all values are greater than zero.
- Also, it will only improve the fit of the normal distribution to the data in cases when the **frequency distribution of the data is right-skewed**.

To take the log transformation for a variable in R is very simple. We simply use the function `log()`, and apply it to the vector of the numerical variable in question. For example, to calculate the log of age for all passengers on the Titanic, we use the command:

```
#eval=F so we don't print three pages of values in the pdf, but you should run it  
log(titanicData$age)
```

This will return a vector of values, each of which is the log of age of a passenger.

### Normal Distribution exercises

1. Let's use R's random number generator for the normal distribution to build intuition for how to view and interpret histograms and QQ plots. Remember, the lists of values generated by `rnorm()` come from a population that truly have a normal distribution.

- A) Generate a list of 10 random numbers from a normal distribution with mean 15 and standard deviation 3 and save the results to `normal_vector`:

```
#your code
```

- B) Use `hist()` to plot a histogram of these numbers from part a.

```
#your code
```

- C) Plot a QQ plot from the numbers in part a.

```
#your code
```

- D) Repeat a through c several times (at least a dozen times [write a loop or function if you feel adventurous]). For each, look at the histograms and QQ plots. Think about the ways in which these look different from the expectation of a normal distribution (but remember that each of these samples comes from a truly normal population).

```
#your code
```

2. Repeat the procedures of Question 1, except this time have R sample 250 individuals for each sample. (You can use the same command as in Question 1, but now set `n = 250`.) Do the graphs and QQ plots from these larger samples look more like the normal expectations than the smaller sample you already did? Why do you think that this is?

```
#your code
```

## Air Quality Example

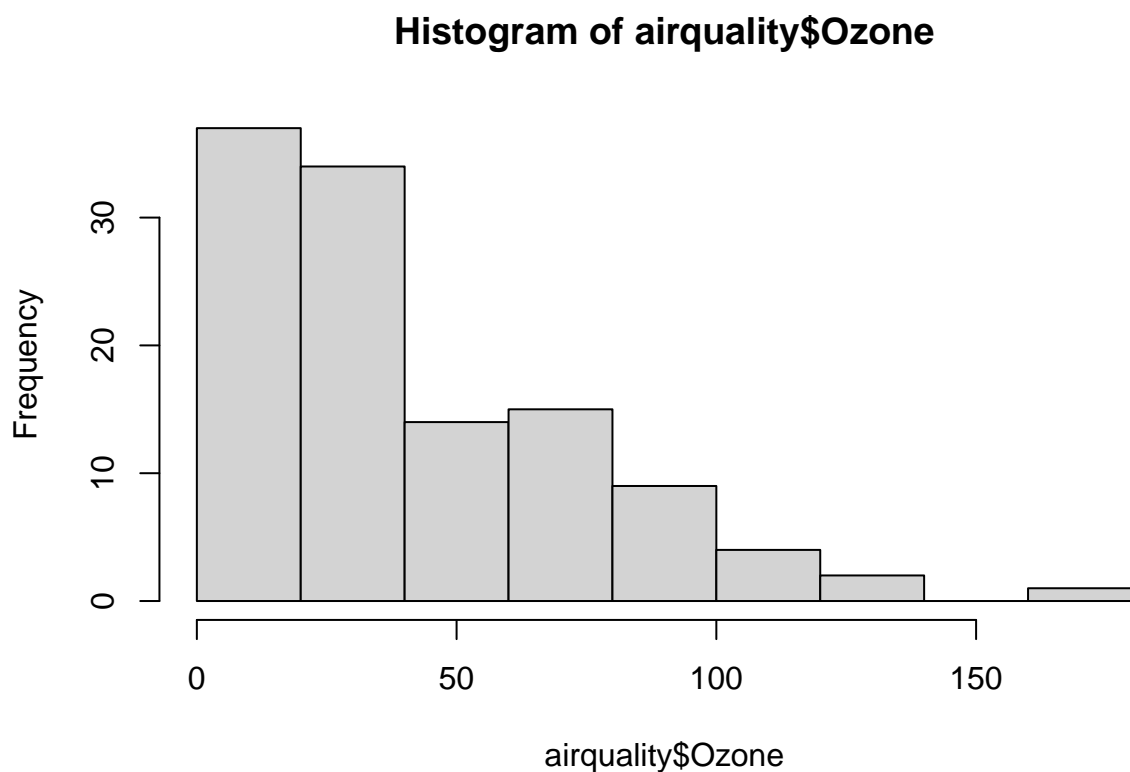
The air quality data set from the `Datasets` package contains 4 measures of air quality over a year. Ozone level, solar radiation, wind speed and temperature. You are going to test the normality of each of these measures. But before you load the data guess which measure you think will look the most and least normal.

```
## Most normal:
```

```
## Least normal:
```

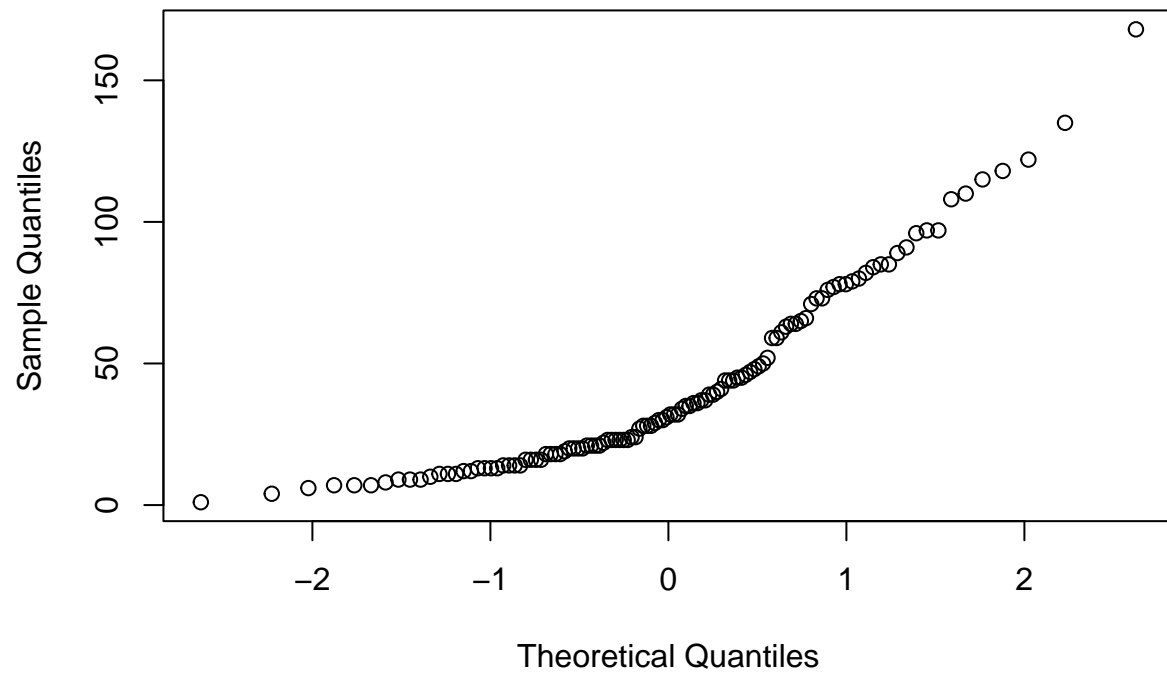
Load then data and make histograms and QQ plots for each of these variables!

```
#Load data from the datasets package
airquality<-datasets::airquality
#Ozone
hist(airquality$Ozone)
```



```
qqnorm(airquality$Ozone)
```

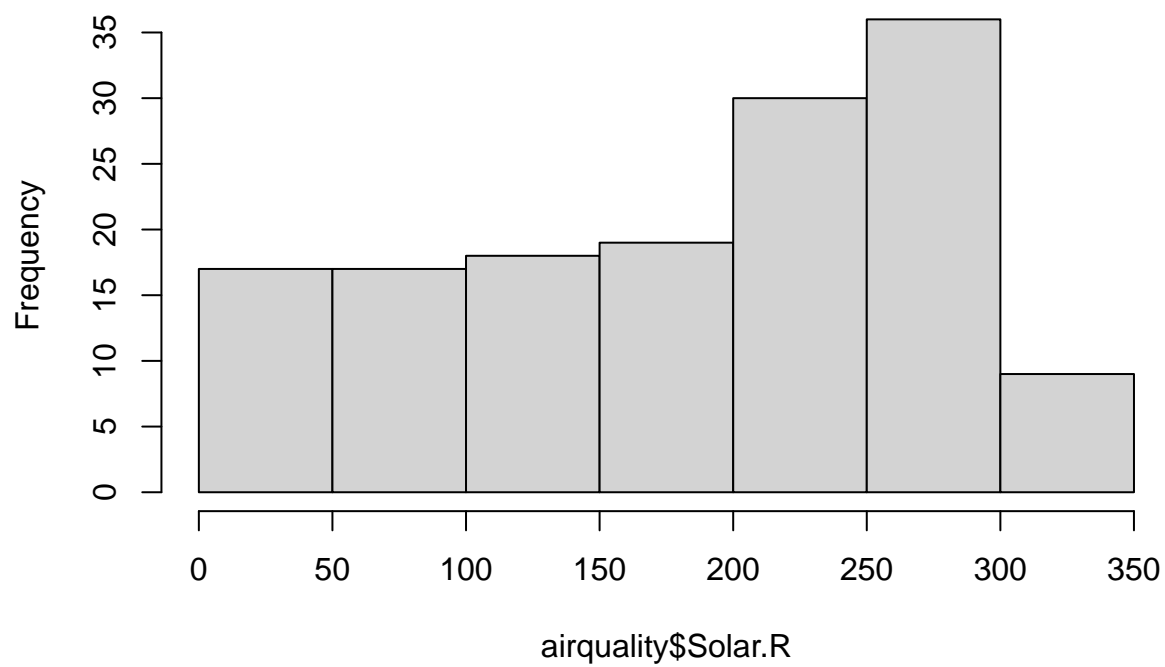
Normal Q-Q Plot



```
#Solar.R  
hist(airquality$Solar.R)
```

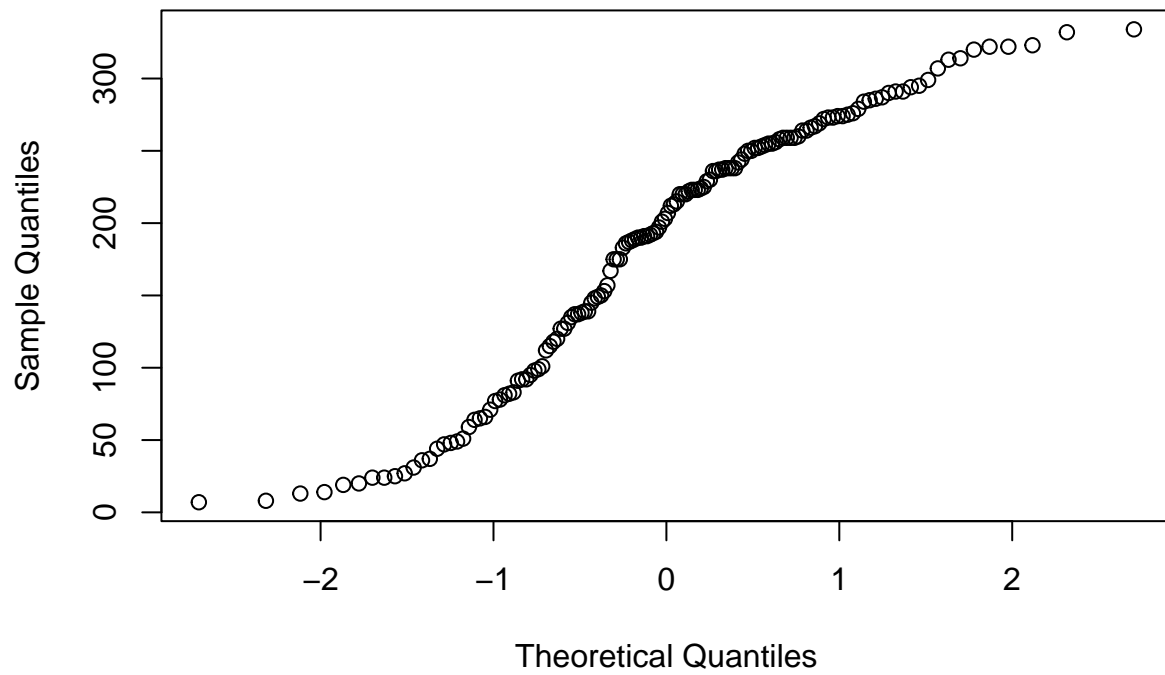


**Histogram of airquality\$Solar.R**



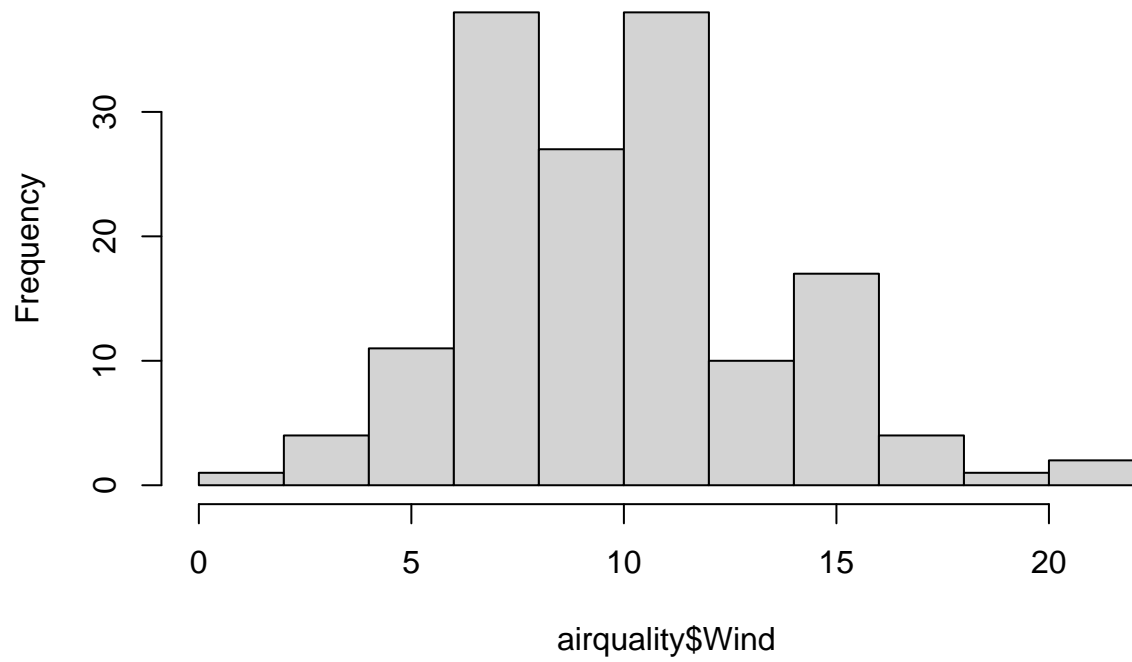
```
qqnorm(airquality$Solar.R)
```

Normal Q-Q Plot



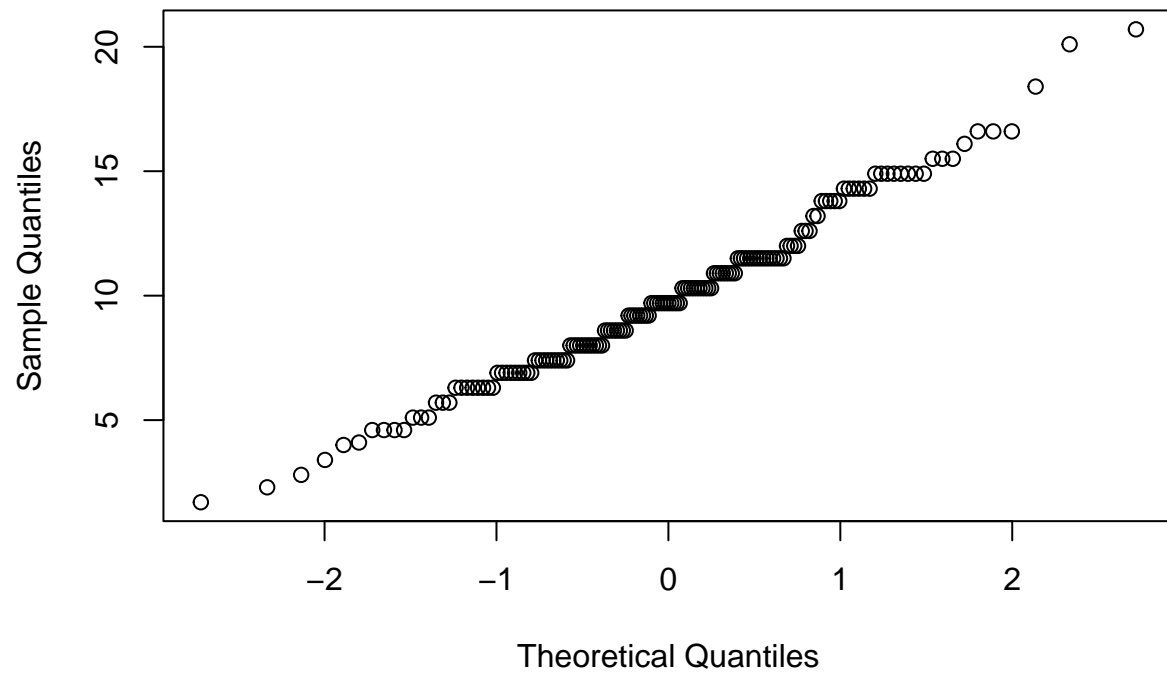
```
#Wind  
hist(airquality$Wind)
```

**Histogram of airquality\$Wind**



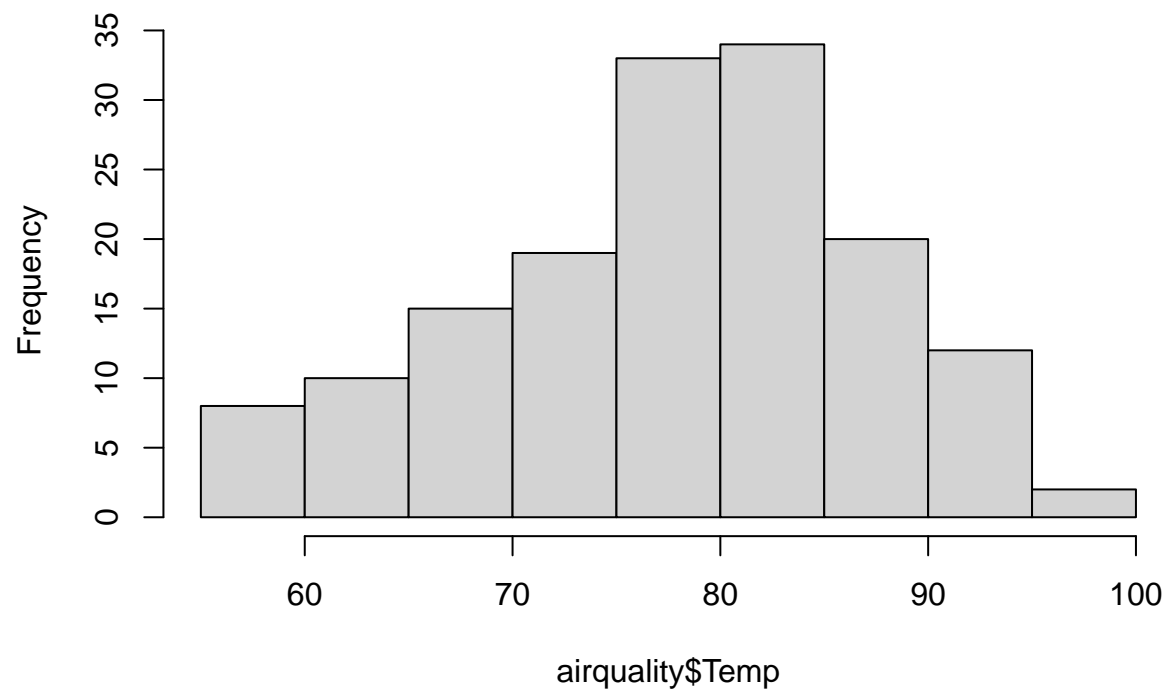
```
qqnorm(airquality$Wind)
```

Normal Q-Q Plot



```
#Temp  
hist(airquality$Temp)
```

**Histogram of airquality\$Temp**



```
qqnorm(airquality$Temp)
```

Normal Q-Q Plot

