Old Faithful Notebook

List datasets available in R

```
data()
```

Loads the Old Faithful Geyser dataset

```
head(faithful)
```

```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

```
rm(list=ls())
```

Display (compactly) the internal structure of the R object - here it is a dataframe with 272 observations and 2 variables

- eruptions: duration of eruption in minutes
- waiting: duration until the next eruption in minutes

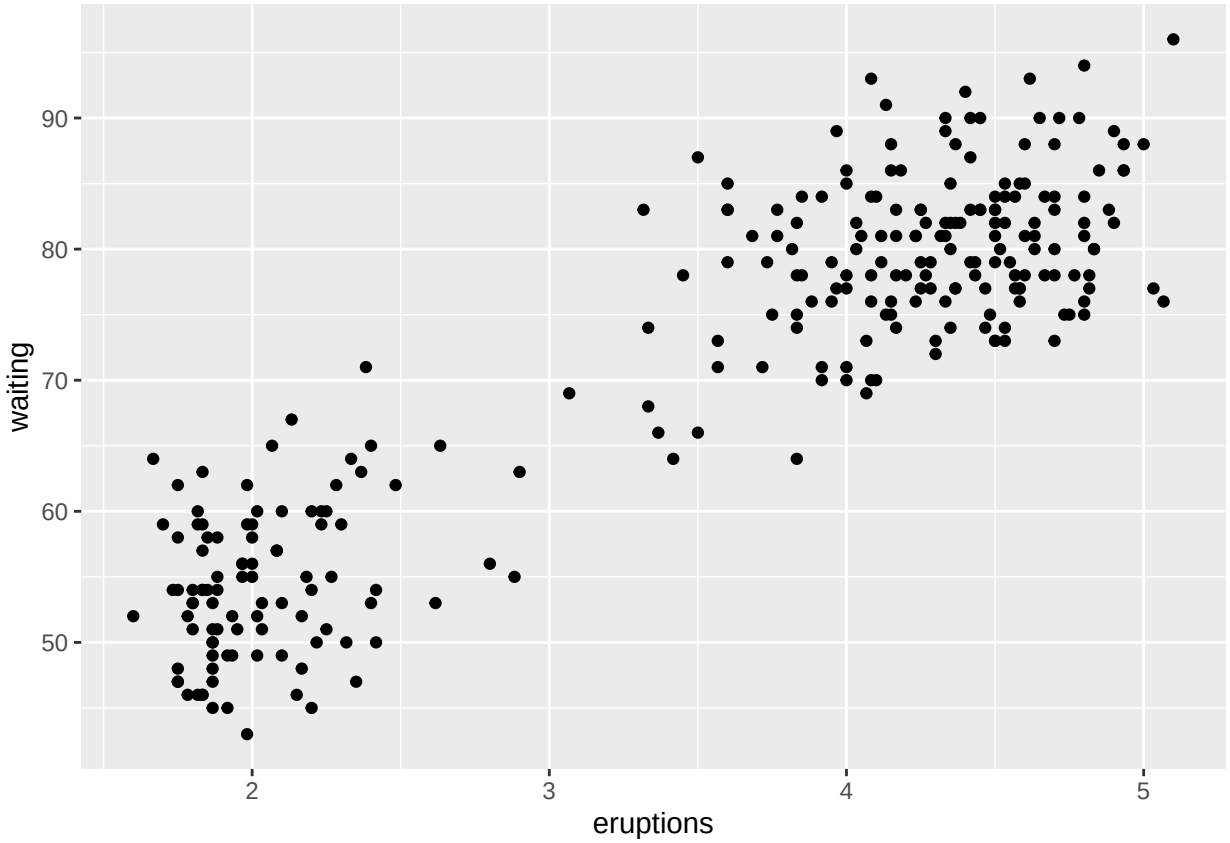
```
str(faithful)
```

```
## 'data.frame':    272 obs. of  2 variables:
##  $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
##  $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
```

The **Old Faithful Geyser** is a hot spring that occasionally becomes unstable and erupts hot water and steam into the air. The Old Faithful Geyser is at Yellowstone Park, Wyoming.

Visualisation *Scatter plot*

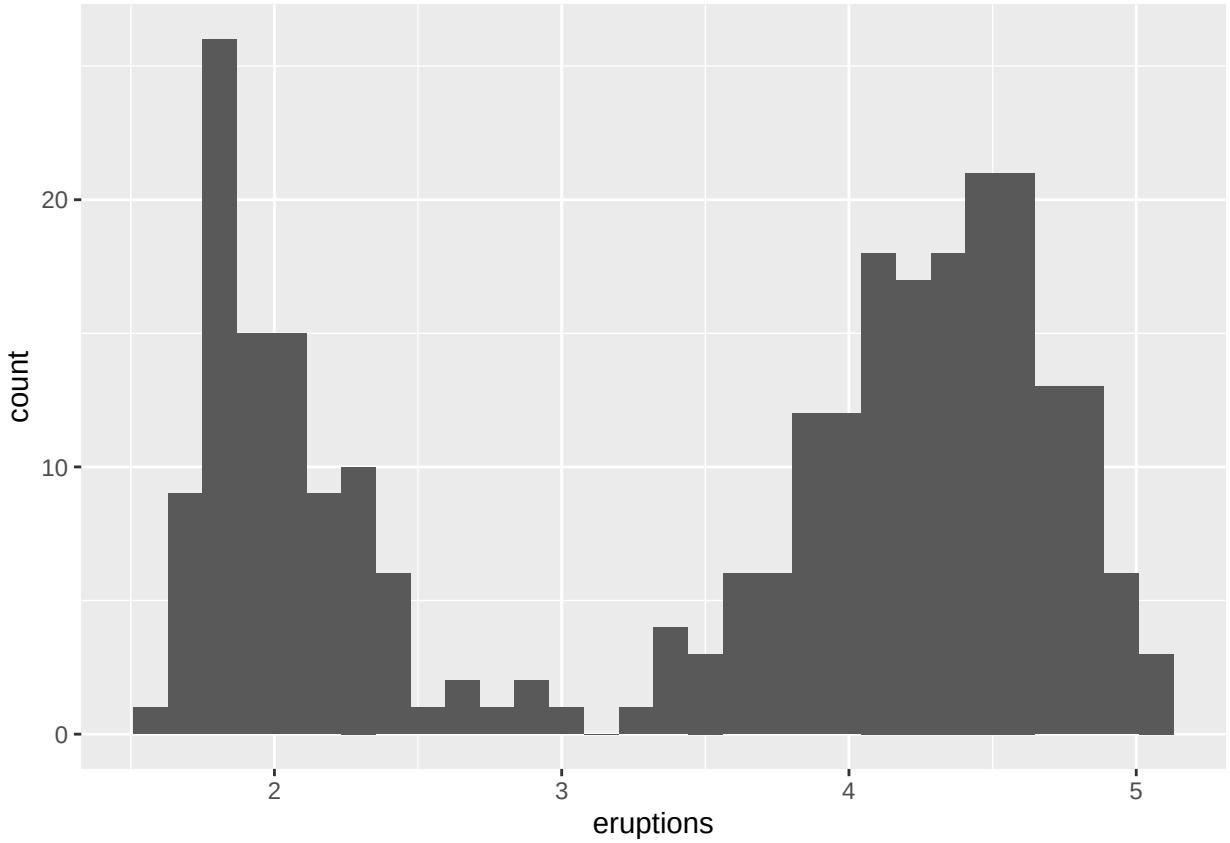
```
library(ggplot2)

ggplot(data = faithful, aes(x = eruptions, y = waiting)) + geom_point()
```



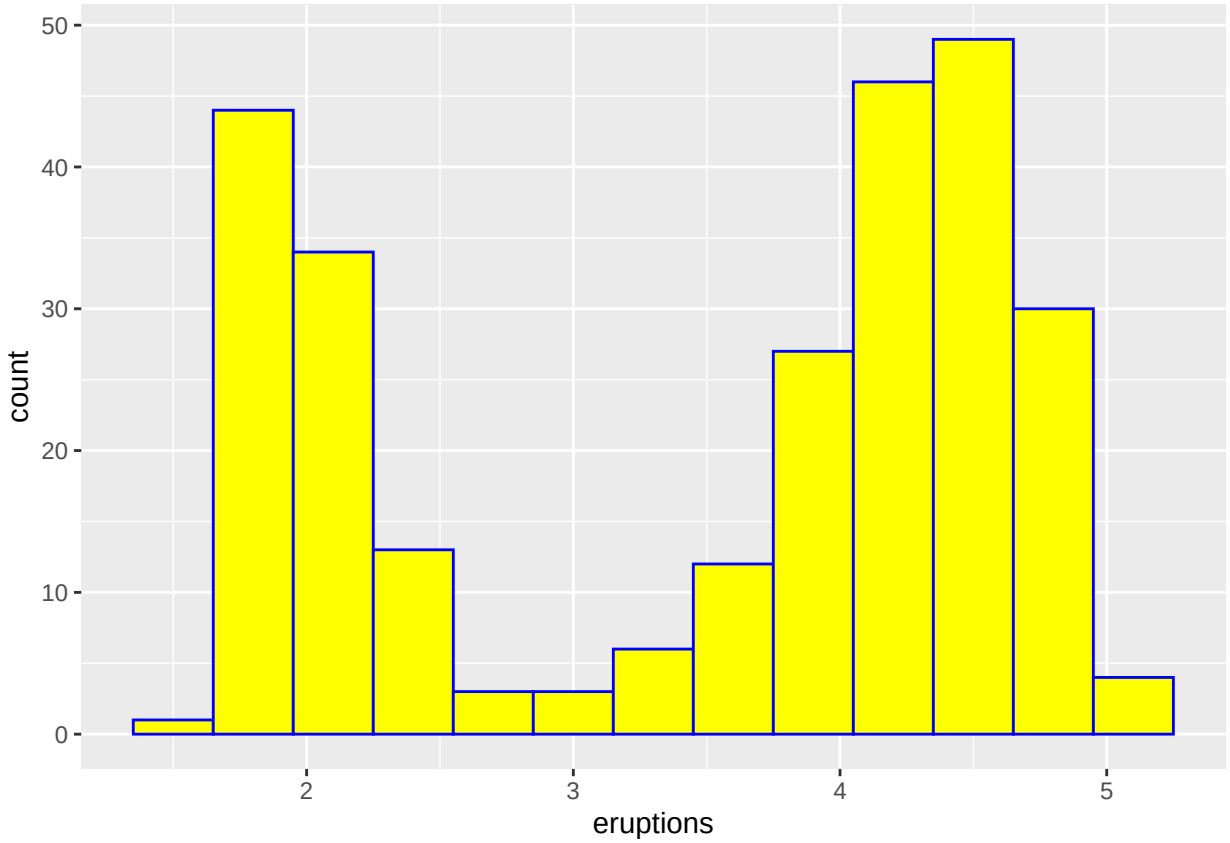
Histogram

```
ggplot(data = faithful, aes(x = eruptions)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

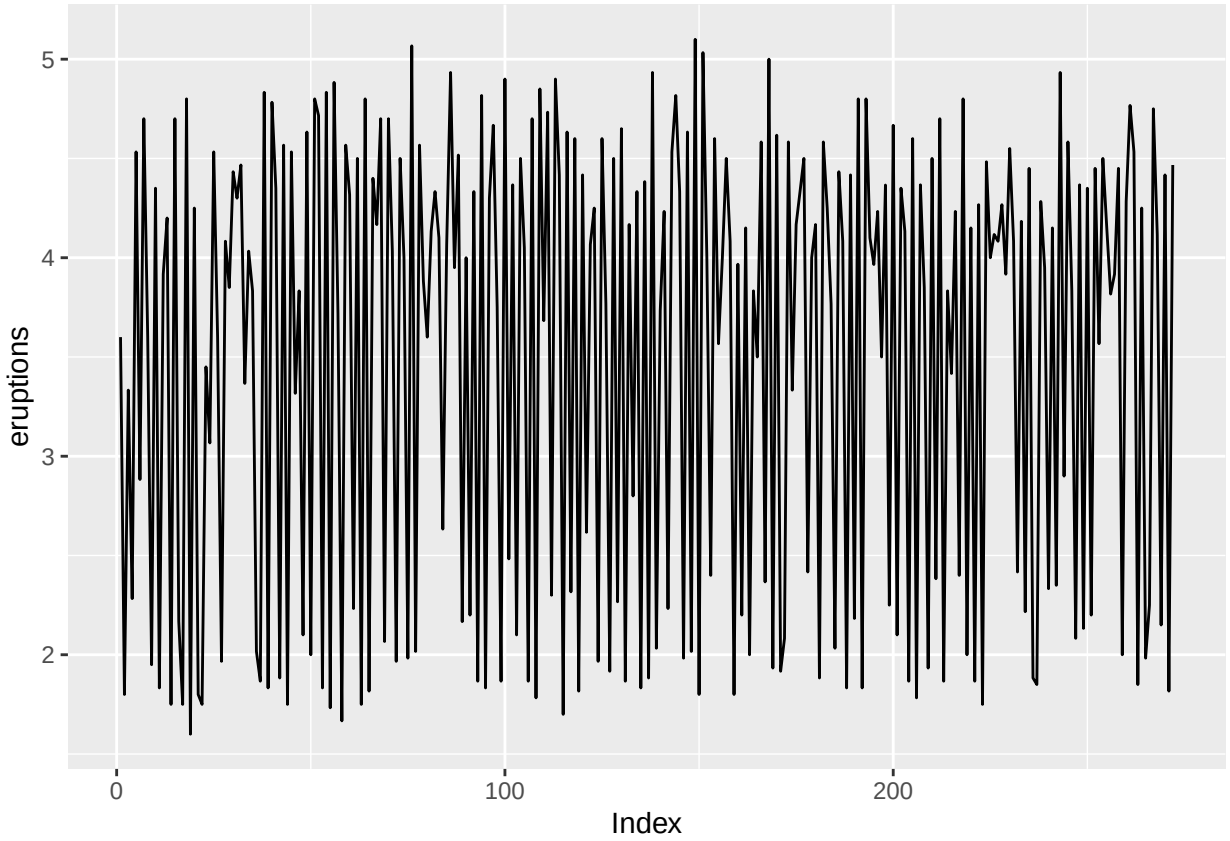



```
ggplot(data = faithful, aes(x = eruptions)) + geom_histogram(binwidth = 0.3, color = "blue", fill = "ye
```

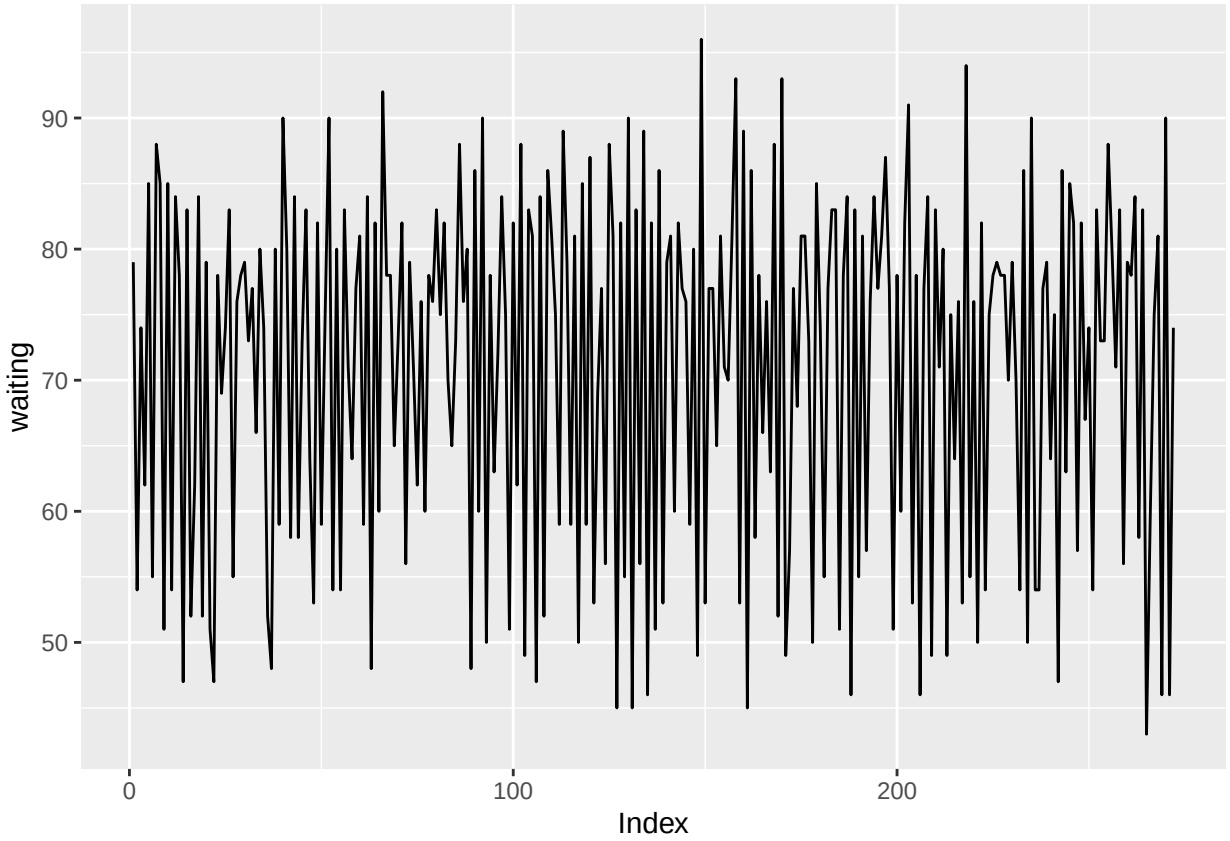


Time series plot

```
# index refers to the sequence of eruption
ggplot(data = faithful) + geom_line(aes(x = 1:length(eruptions), y = eruptions)) + xlab("Index")
```

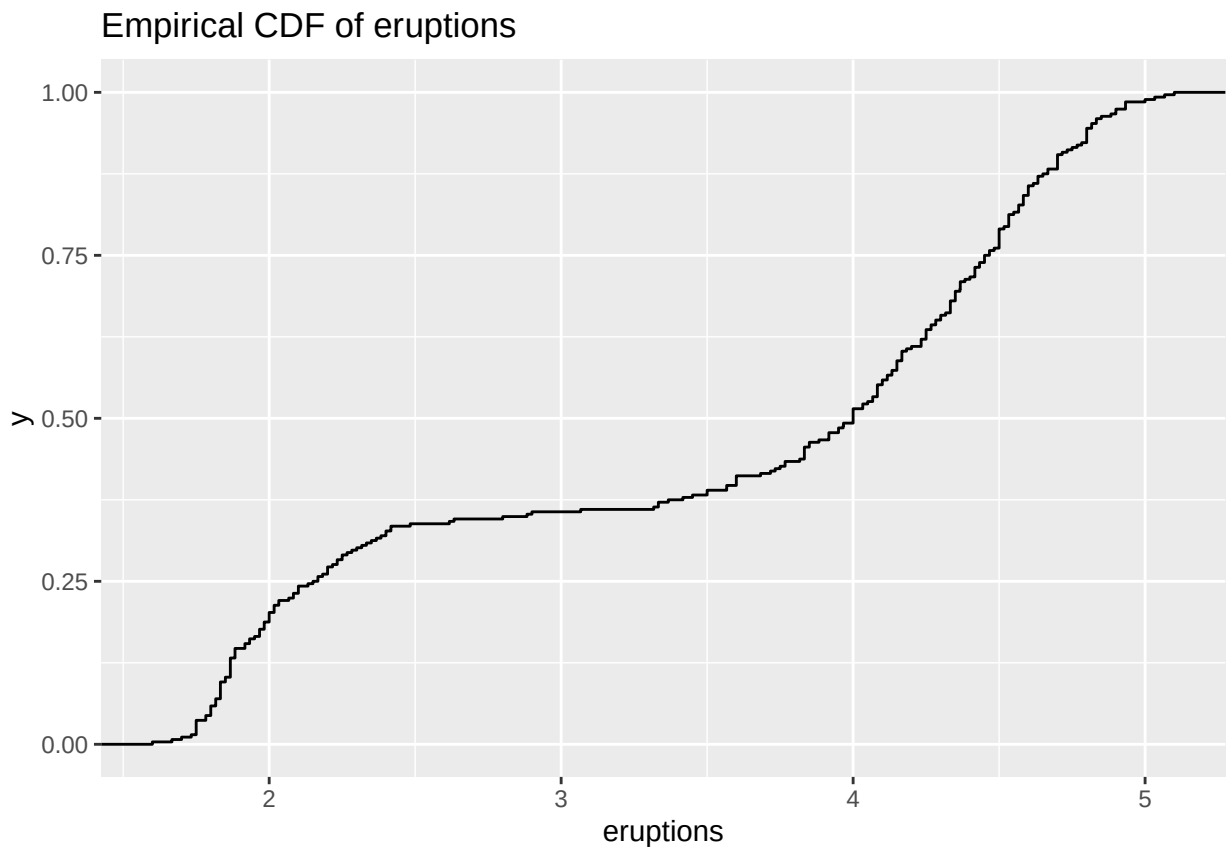



```
# alternative code
# ggplot(data = faithful, aes(x = 1:length(eruptions), y = eruptions)) + geom_line() + xlab("Index")

ggplot(faithful) + geom_line(aes(x = 1:length(waiting), y = waiting)) + xlab("Index")
```



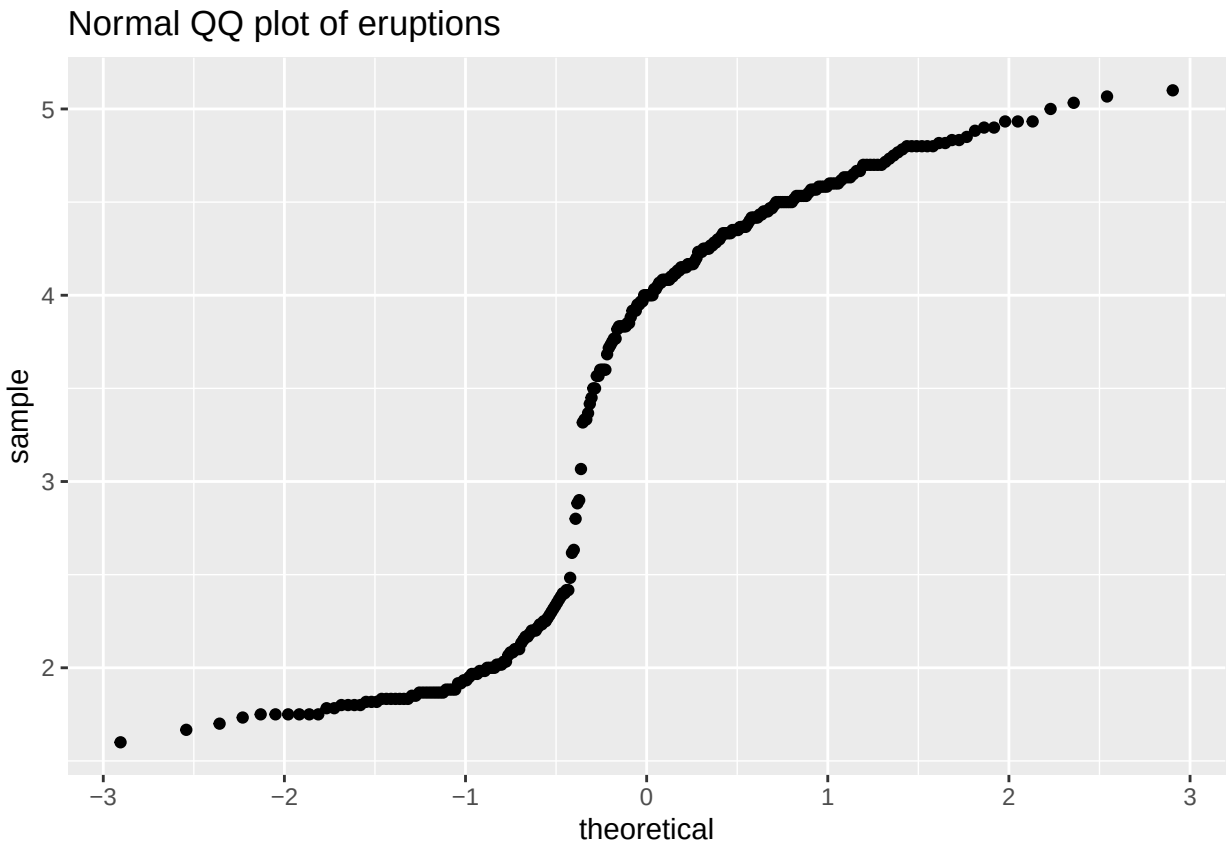
Empirical cdf plot for eruptions

```
# stat_ecdf() function to compute empirical cumulative distribution
ggplot(data = faithful, aes(x = eruptions)) + stat_ecdf() + ggtitle("Empirical CDF of eruptions")
```

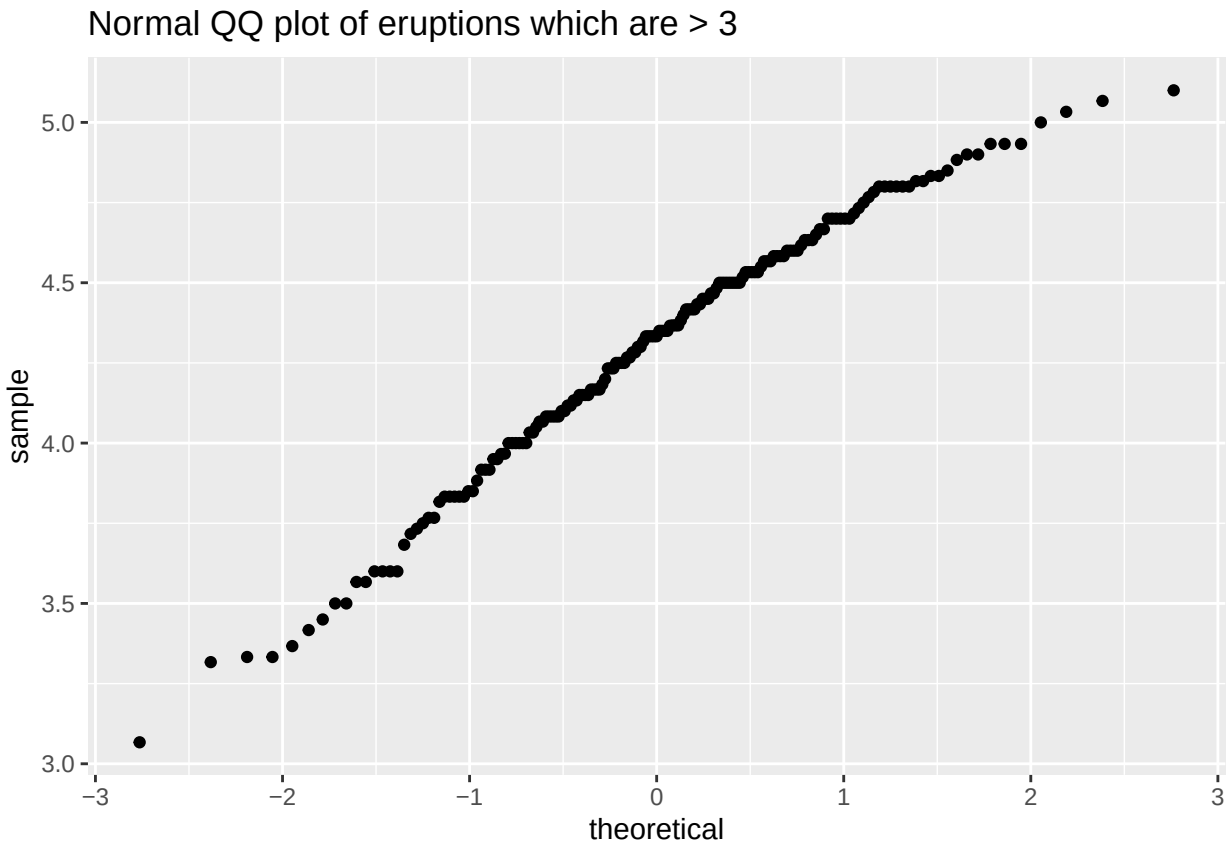


Normal quantile plots

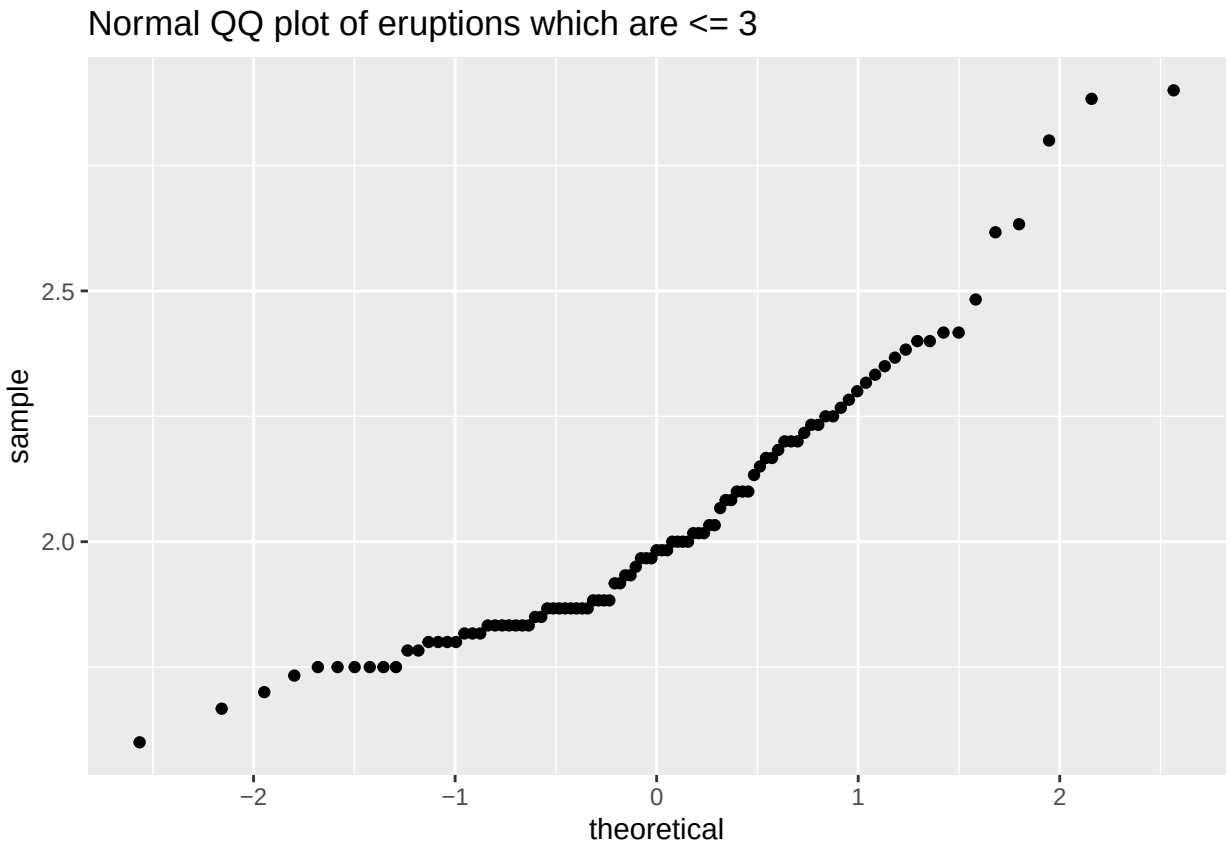
```
# stat_qq() function to generate qq plots
ggplot(data = faithful, aes(sample = eruptions)) + stat_qq() + ggtitle("Normal QQ plot of eruptions")
```




```
ggplot(data = subset(faithful, eruptions > 3), aes(sample = eruptions)) + stat_qq() + ggtitle("Normal Q
```




```
ggplot(data = subset(faithful, eruptions <= 3), aes(sample = eruptions)) + stat_qq() + ggtitle("Normal
```



Certain observations from the plots

1. Eruption times and the waiting times between successive eruptions exhibit highly oscillatory behavior, low followed by high and high followed by low.

- clarification: not relating the eruption and waiting times. individually, both eruption and waiting times exhibit oscillatory behaviour.

2. Eruption times have a bimodal distribution.

- based on empirical CDF plot, we can see that there are two peaks, hence suggesting bimodal distribution.

3. Lower eruption times are followed by lower waiting times. Higher eruption times are followed by higher waiting times. This can be used to predict when the next geyser eruption will occur. For example, during a short eruption, less water and heat are used and so both are restored in shorter time. During longer eruptions, more time is needed to rebuild.

- based on the scatter plot of waiting vs eruption times

T-test and finding confidence intervals

1. Performs a one sample t-test to test the hypothesis that the mean of waiting time mu = mu0 and derives the 95% confidence interval for the mean parameter. Note the goal is more to find the confidence interval here.

- t test: to determine if there is a significant difference between the means of two groups

2. With 95% confidence, if the eruption time < 3, the average waiting time is between 53.3 and 55.67.
3. With 95% confidence, if the eruption time $>= 3$, the average waiting time is between 79.1 and 80.89.

```
# mu0 value
mean_waiting <- mean(faithful$waiting)
# 70.89706

df1 <- subset(faithful, faithful$eruptions < 3)
df2 <- subset(faithful, faithful$eruptions >= 3)

?t.test
t.test(df1$waiting, conf.level = 0.95, mu = mean_waiting)
```

```
##
##  One Sample t-test
##
## data:  df1$waiting
## t = -27.661, df = 96, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 70.89706
## 95 percent confidence interval:
##  53.31781 55.67189
## sample estimates:
## mean of x
##  54.49485
```

```
t.test(df2$waiting, conf.level = 0.95, mu = mean_waiting)
```

```
##
##  One Sample t-test
##
## data:  df2$waiting
## t = 20.064, df = 174, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 70.89706
## 95 percent confidence interval:
##  79.09425 80.88289
## sample estimates:
## mean of x
##  79.98857
```