IB R-Workshop: base R plotting

Keith Bouma-Gregson

January 2018

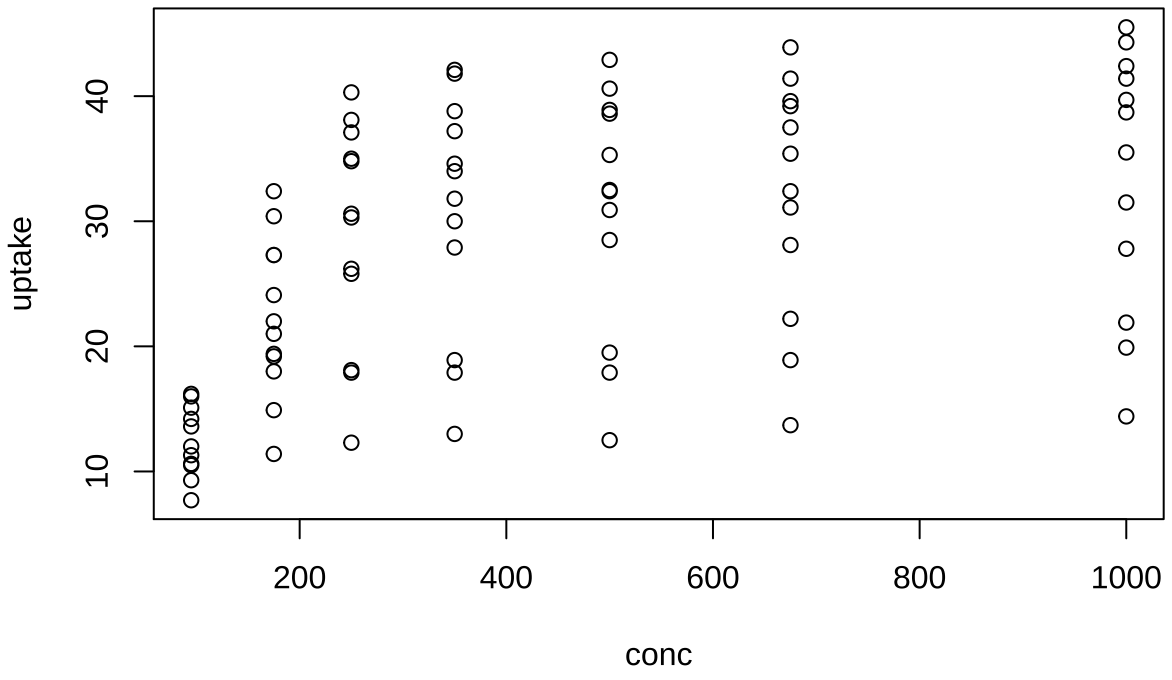
Plotting in base R

The basic plotting function is `plot()`, which takes a formula `plot(y ~ x)` or explicit definitions of the x and y axis `plot(x= , y= )`. The inputs can be vectors or columns in a data frame. If columns in a data frame, then the data frame must also be defined, `plot(y ~ x, data= ?)` or `plot(dataframe$y ~ dataframe$x)`. Let's make some plots with `CO2` data frame, which shows different CO2 uptake rates for plants.

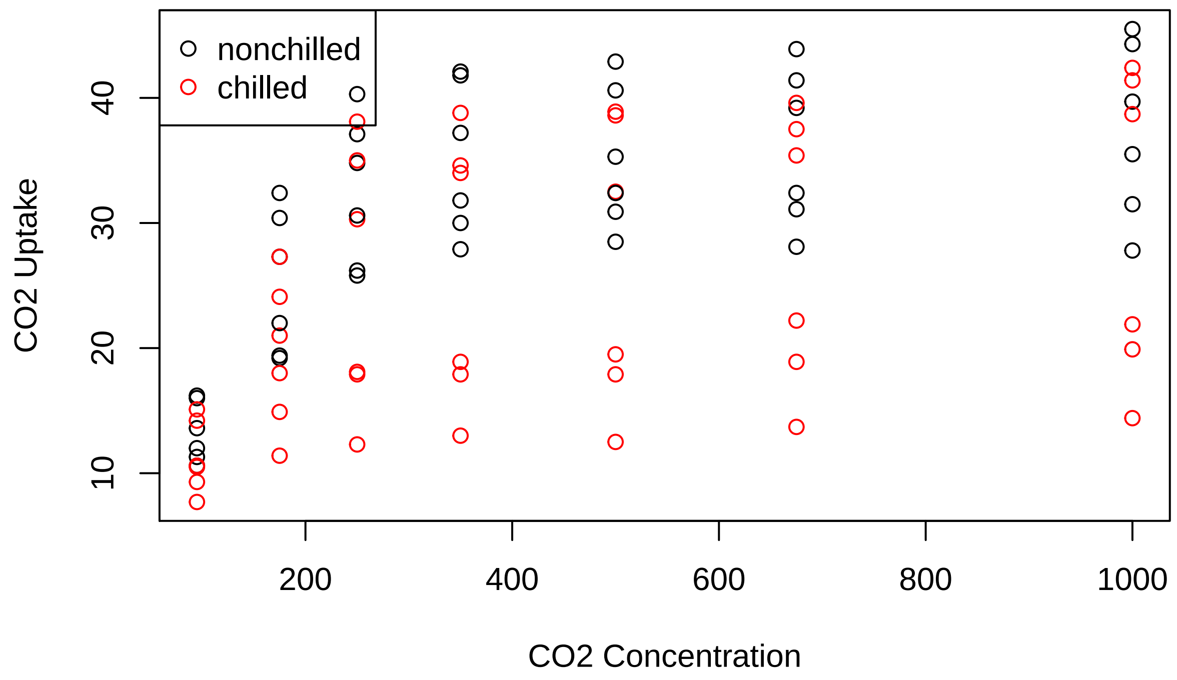
```
head(CO2)
```

```
##   Plant   Type  Treatment conc uptake
## 1   Qn1 Quebec nonchilled   95   16.0
## 2   Qn1 Quebec nonchilled  175   30.4
## 3   Qn1 Quebec nonchilled  250   34.8
## 4   Qn1 Quebec nonchilled  350   37.2
## 5   Qn1 Quebec nonchilled  500   35.3
## 6   Qn1 Quebec nonchilled  675   39.2
```

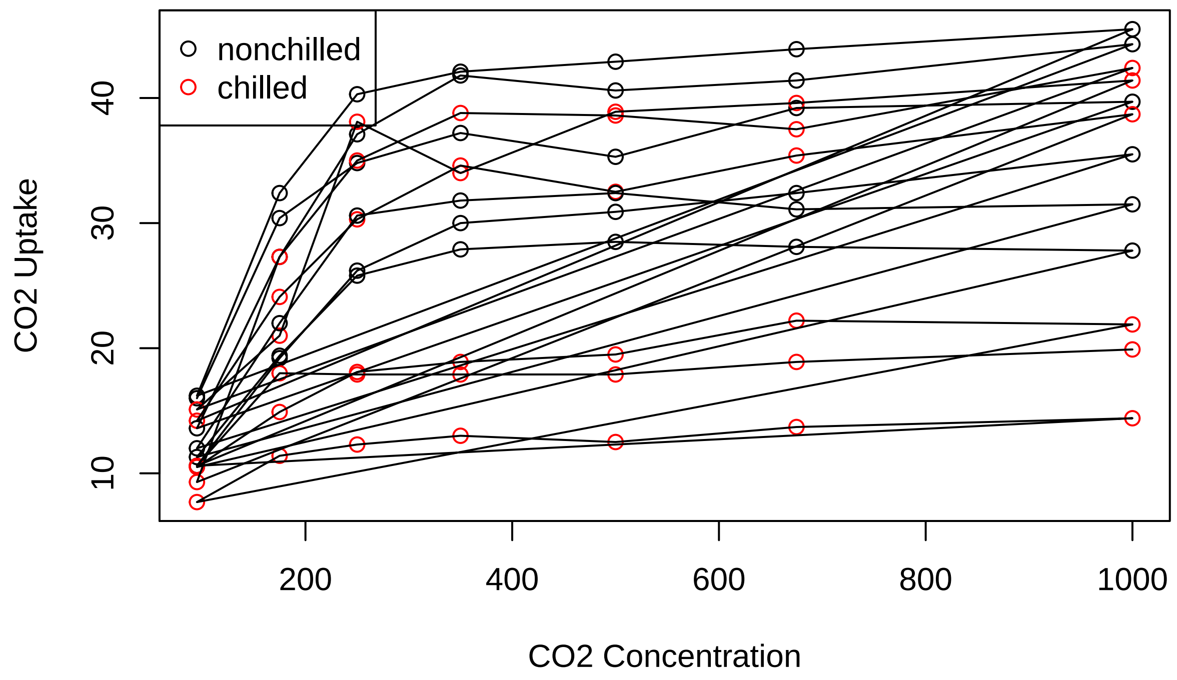
```
#  Plot of uptake vs. CO2 concentration

# plot(CO2$uptake ~ CO2$conc)
# or
plot(uptake ~ conc, data= CO2)
```



```
# Add color, labels, and legend
plot(uptake ~ conc, data= CO2,
     col= Treatment,
     xlab= "CO2 Concentration",
     ylab= "CO2 Uptake")
legend("topleft", legend= levels(CO2$Treatment), pch= 1, col= c("black", "red"))
```



```
#  Add lines to the points
plot(uptake ~ conc, data= CO2,
     col= Treatment,
     xlab= "CO2 Concentration",
     ylab= "CO2 Uptake")
legend("topleft", legend= levels(CO2$Treatment), pch= 1, col= c("black", "red"))
lines(uptake ~ conc, data= CO2,
      col= Treatment)
```

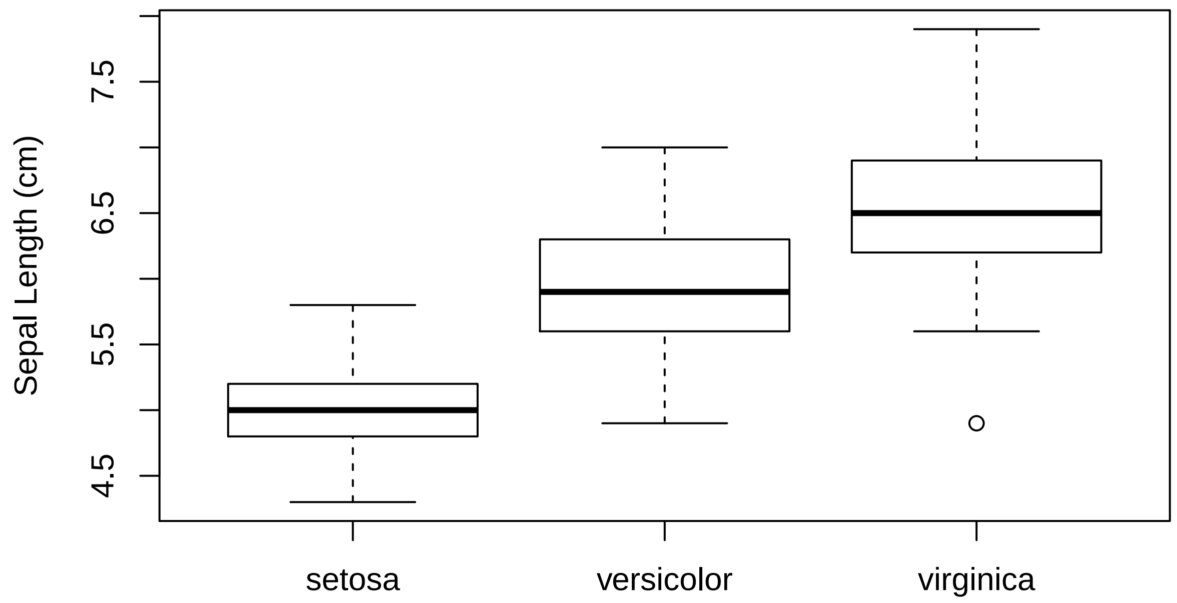


Boxplots

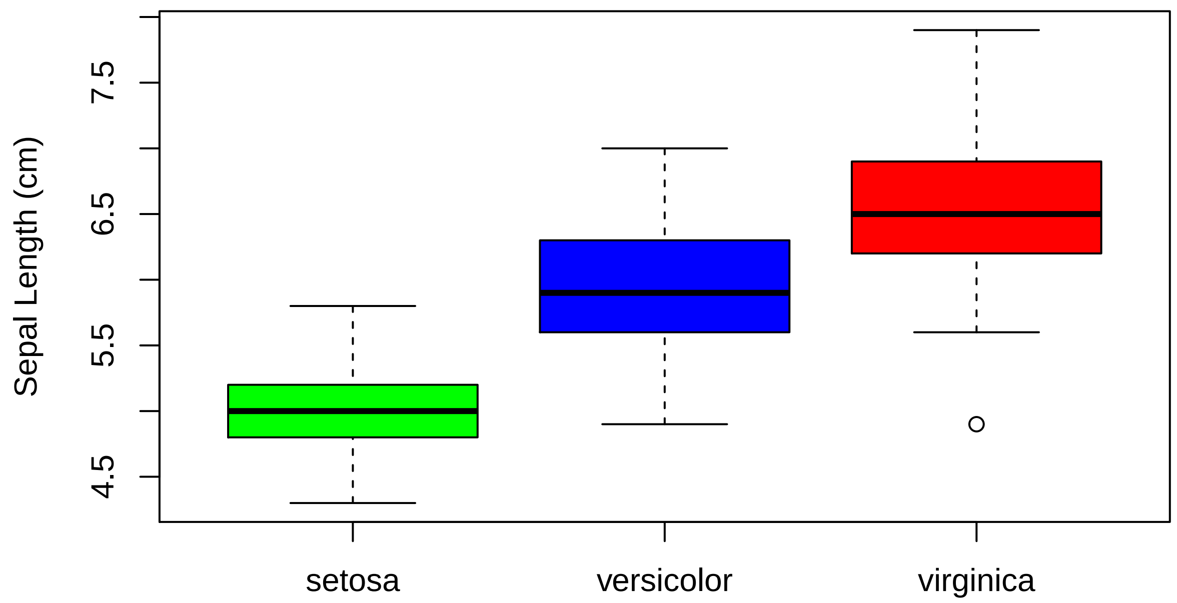
```r
# Iris data sets includes sepal and petal length for 3 different species of Iris
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
# A boxplot of the same data
boxplot(Sepal.Length ~ Species, data= iris,
        ylab= "Sepal Length (cm)")
```



```r
# Customize colors
boxplot(Sepal.Length ~ Species, data= iris,
        col= c("green", "blue", "red"),
        ylab= "Sepal Length (cm)")
```



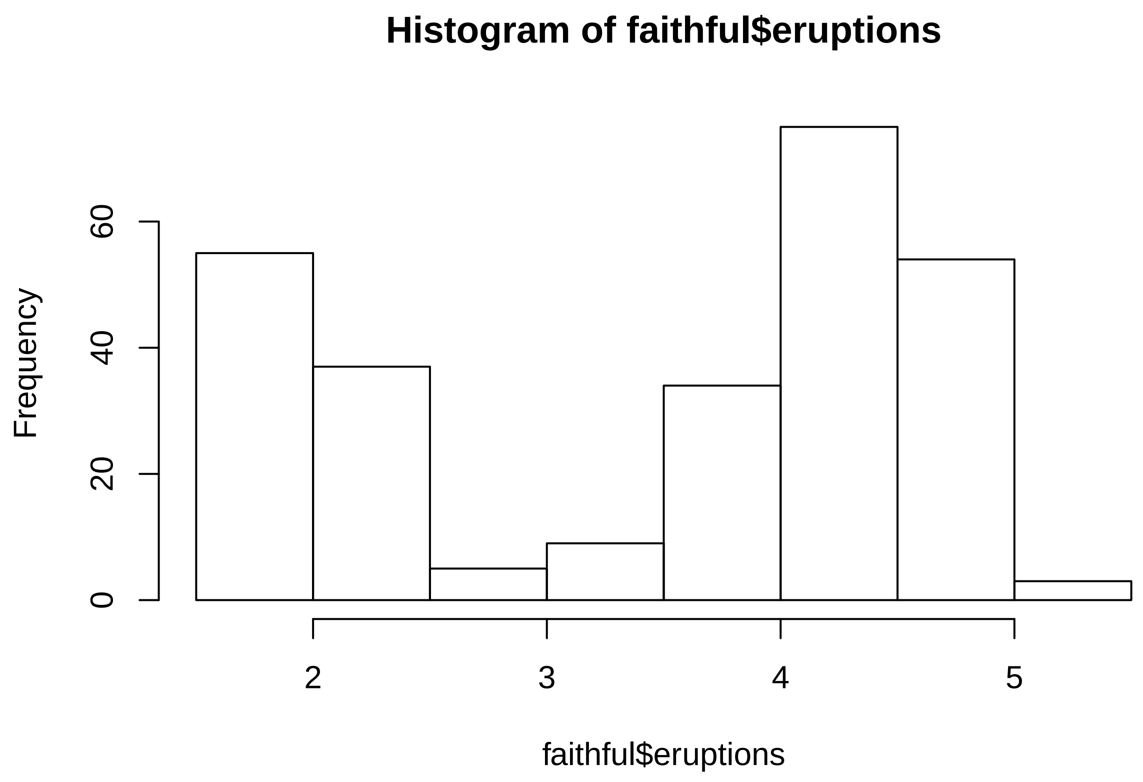
Histograms

Histograms are created with the `hist()` function. The argument `breaks` specifies how many bins the data will be aggregated into.

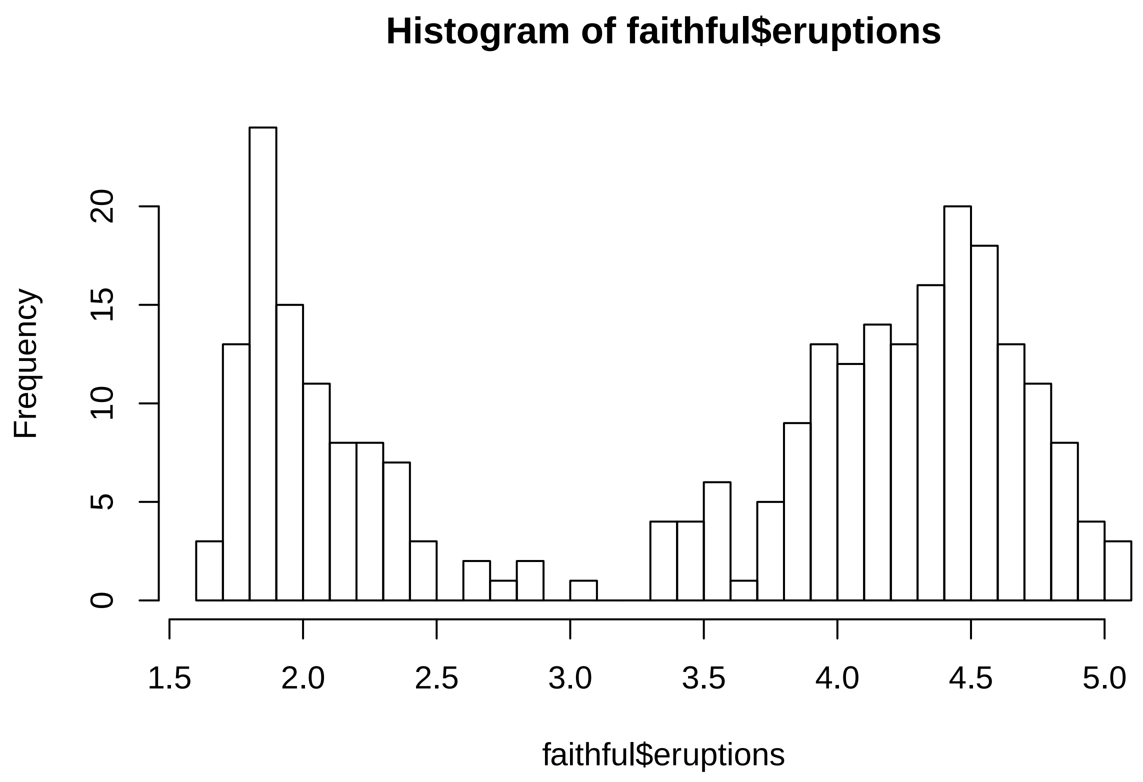
```r
# Duration of eruptions (min) and waiting time between eruptions (min)
# for Old Faithful geyser in Yellowstone National Park
head(faithful)
```

```
##   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
```

```
hist(faithful$eruptions) # default breaks
```



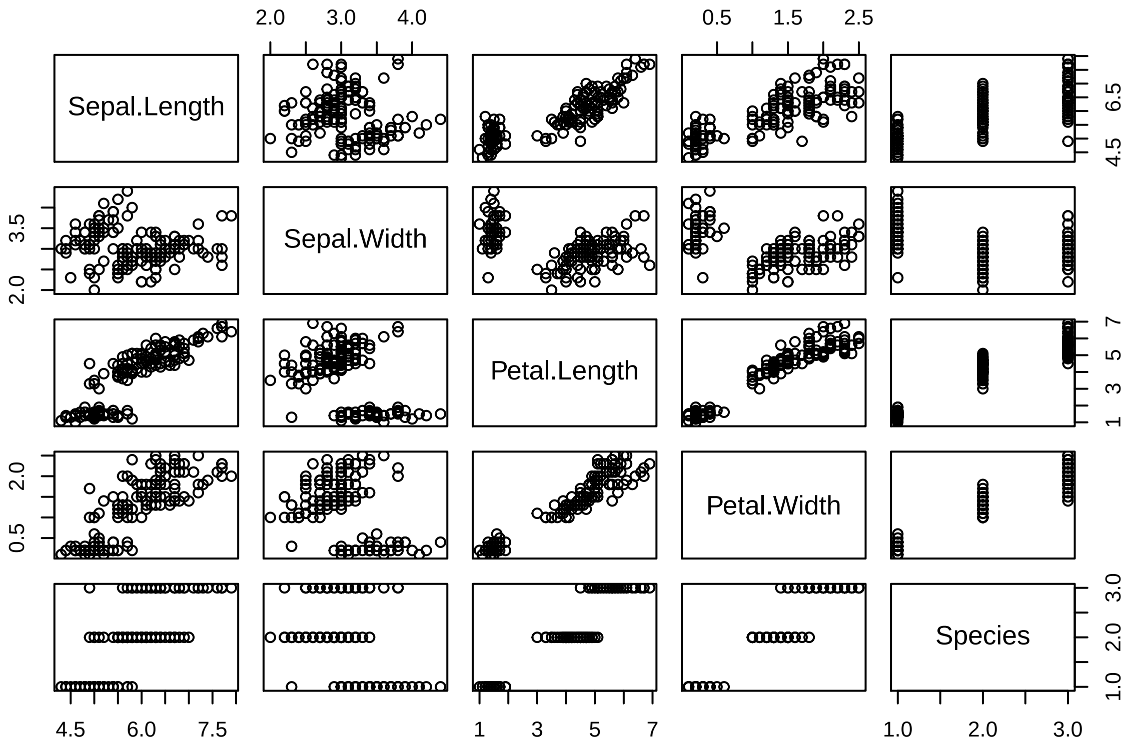
```
hist(faithful$eruptions, breaks= 40)
```



Plotting pairwise combinations of variables

The `pairs` command can be helpful when looking for patterns among variables. This is especially helpful when looking for correlation among explanatory variables in a statistical model.

```
## Using the iris data set again
pairs(iris)
```



Using plot with different R objects

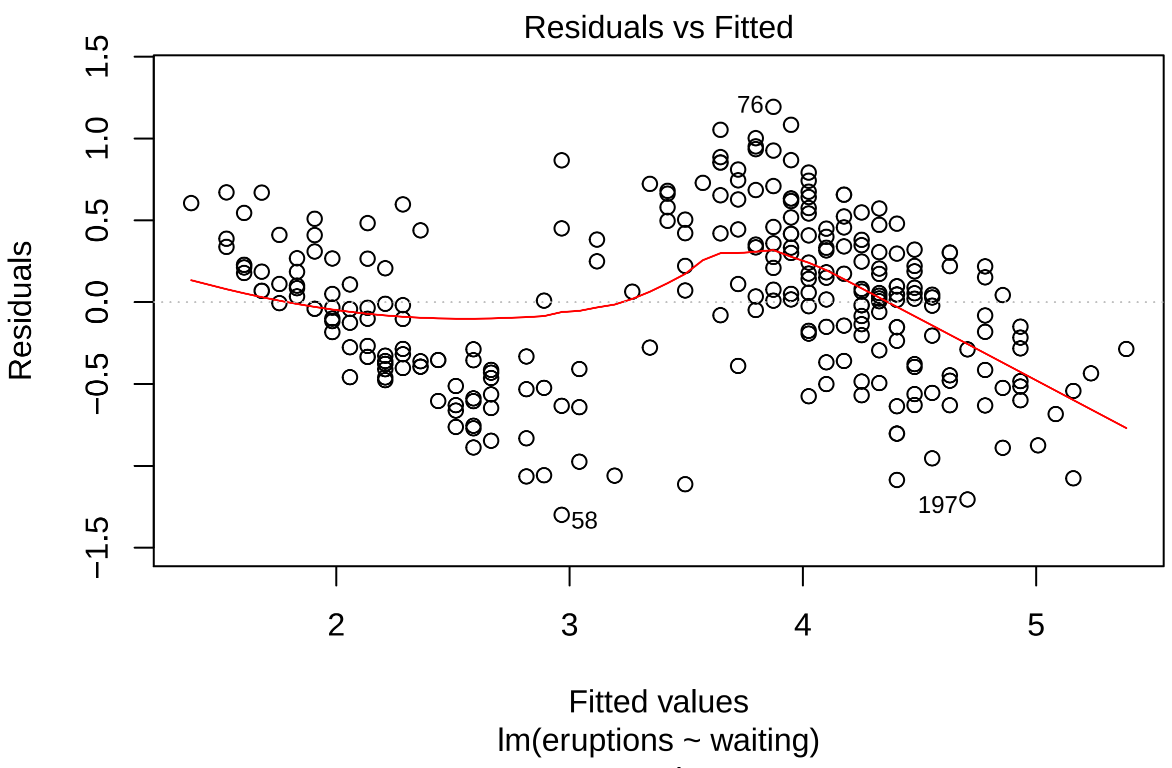
More complex objects can also be plotted using `plot` and many package developers will use `plot` to work with different R objects.

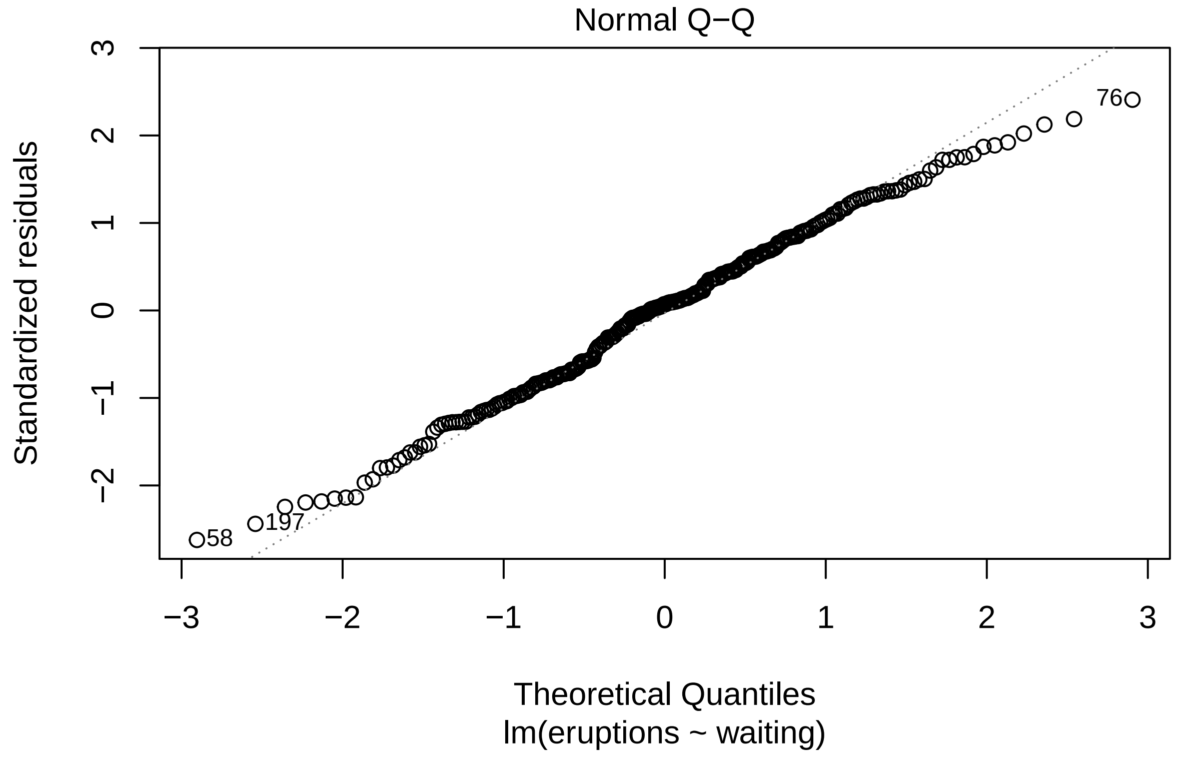
Statistical models We can make a linear regression of our eruptions vs. waiting. Using `plot` on the statistical model automatically generates 4 plots to help evaluation of the model

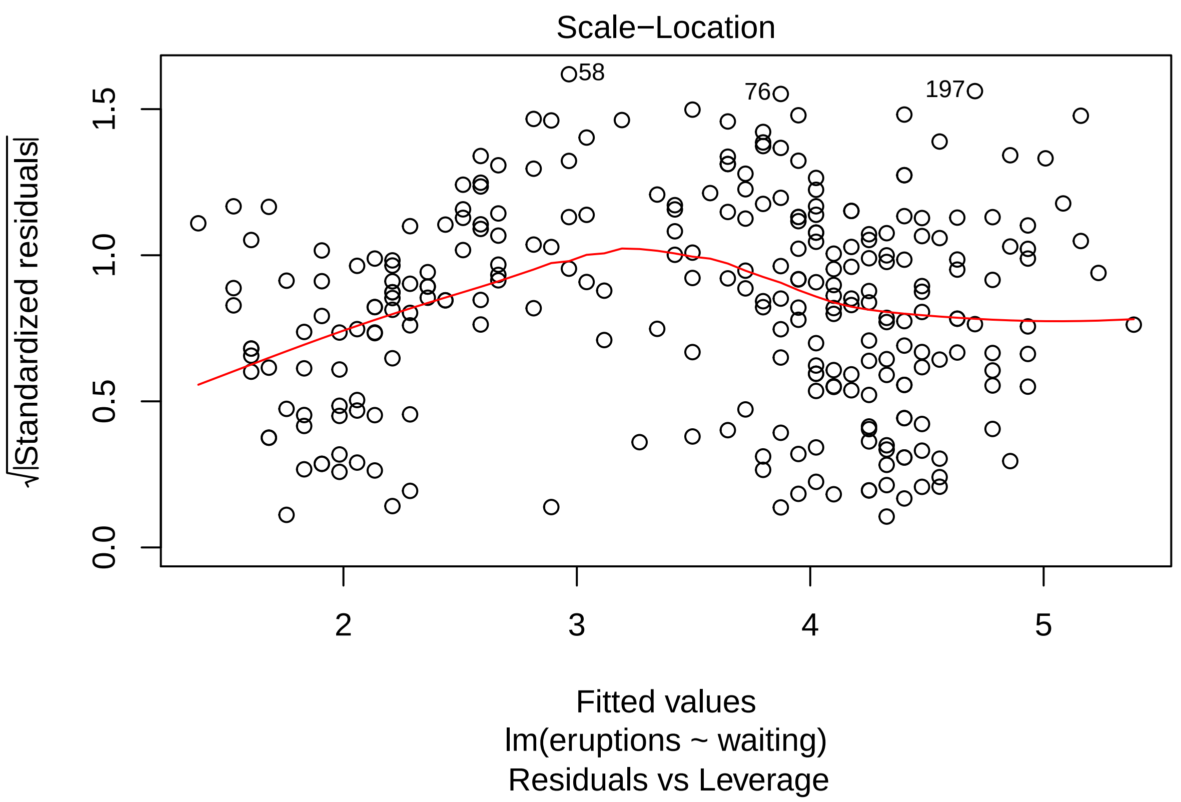
```
lm.faith <- lm(eruptions ~ waiting, data= faithful)
lm.faith
```

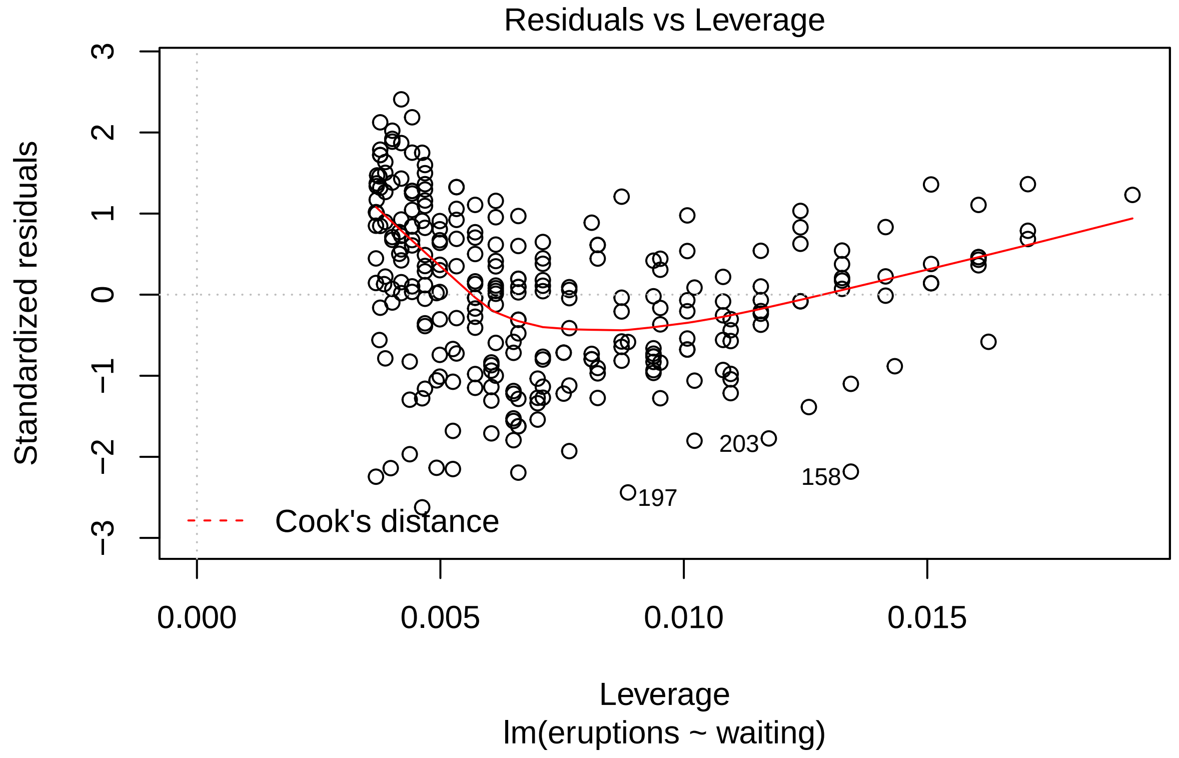
```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Coefficients:
## (Intercept)      waiting
##    -1.87402      0.07563
```

```
plot(lm.faith)
```


Residuals vs Fitted
Residuals
Fitted values
lm(eruptions ~ waiting)
76
58
197
1.5
1.0
0.5
0.0
−0.5
−1.5
2
3
4
5


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(eruptions ~ waiting)
76
58
197
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

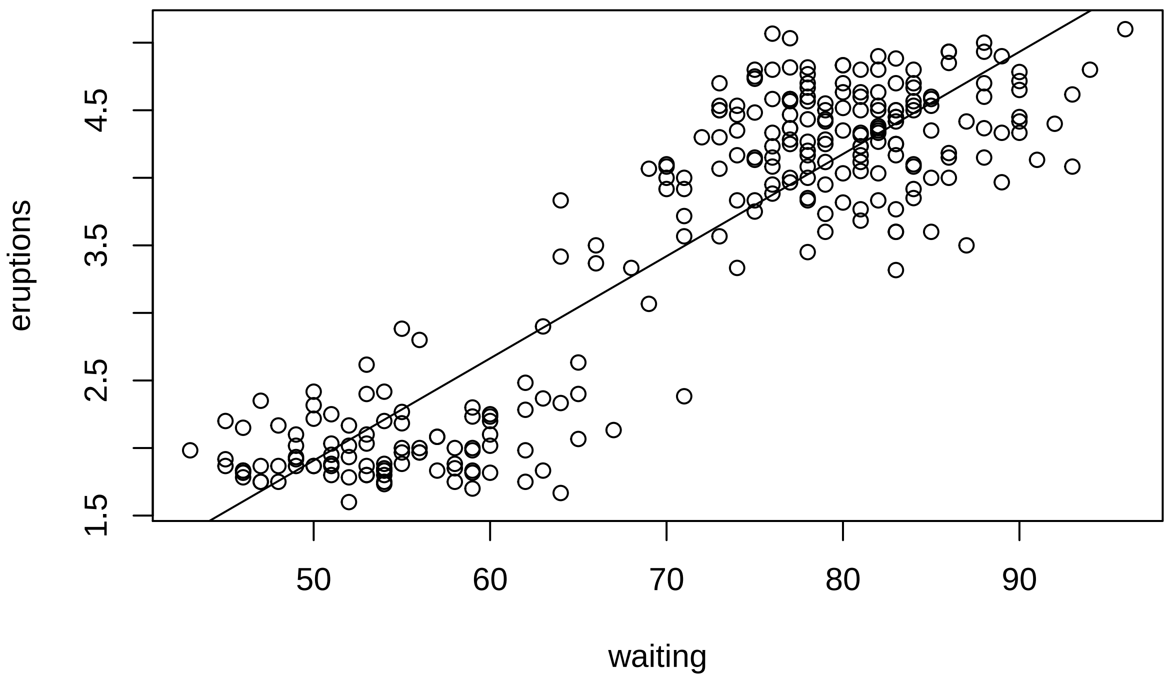




\

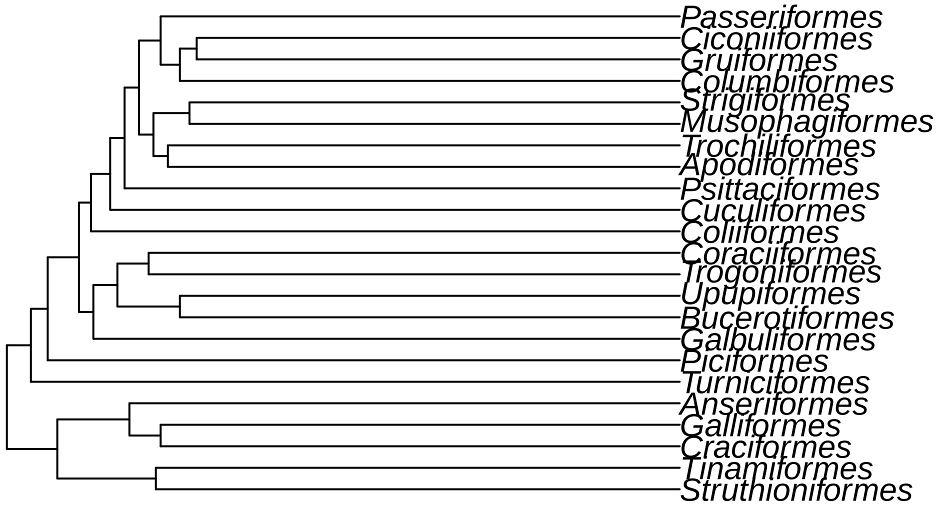
We can use abline to add the regression line to our scatterplot

```r
plot(eruptions ~ waiting, data= faithful)
abline(lm.faith)
```



Phylogenetic Trees

```r
# install.packages("ape")
library(ape)

data(bird.orders)
plot(bird.orders)
```



Passeriformes
Ciconiiformes
Gruiformes
Columbiformes
Strigiformes
Musophagiformes
Trochiliformes
Apodiformes
Psittaciformes
Cuculiformes
Coliiformes
Coraciiformes
Trogoniformes
Upupiformes
Bucerotiformes
Galbuliformes
Piciformes
Turniciformes
Anseriformes
Galliformes
Craciformes
Tinamiformes
Struthioniformes