This exercise introduces some new functions for assessing normality and performing non-parametric tests. We won’t cover new data manipulation methods for a few weeks, so you have time to solidify your understanding of the subset function (and sapply if you want).

**Part 1: Testing for normality**

**Normal quantile-quantile (Q-Q) plots**

Like a histogram, the normal Q-Q plot requires just a single numeric vector variable as an input. That vector variable will often be the column of a data frame or the result of a subset command.

`qqnorm(x)` #where x is the name of the vector variable/column

To add the line passing through the first and third quantiles (which you can use for assessing normality), use:

`qqline(x)` #where x is the same vector from qqnorm

**Shapiro-Wilk test**

Because this method tests whether one variable differs significantly from a normal distribution, it also requires just a single numeric vector variable as the input:

`shapiro.test(x)` #where x is a vector

The output is quite simple: it reports the W statistic (a value of 1 is expected if the data follow a normal distribution and <1 is increasingly different from normal) and the p value.

You should only use the Shapiro-Wilk test if you are specifically examining the hypothesis that the data follow a normal distribution. If you are instead assessing assumptions for a parametric test (t test, F test, or ANOVA), the Shapiro-Wilk test is too sensitive. You should use graphical methods (histogram or Q-Q plot) in that case.

**Part 2: Kolmogorov-Smirnov (K-S) test (two-sample version)**

The K-S test compares the overall shape of the distribution for two datasets, so requires two numeric vector variables as inputs, separated by a comma:

`ks.test(var1, var2)` #where var1 and var2 are the two vectors

The output gives you the D statistic (the maximum difference between the two cumulative distribution functions), the p value, and, possibly, a warning that the p value is not exact because of ties. You can ignore the warning (because there is nothing you can do about the fact the data has tied values); it just means that R estimated the p value from a statistical distribution rather than calculating it exactly.
You can generate the empirical cumulative distribution function using the `ecdf()` function and plot it with `plot()`:

```r
plot(ecdf(x)) # where x is the vector variable
```

The plot command also allows you to vary the color, axis labels, or main title as you did for the histograms.

To add additional data (e.g., from another data set for comparison), you can either use:

```r
plot(ecdf(y), add = T) # where y is the vector variable
```

or

```r
lines(ecdf(y)) # where y is the vector variable
```

You will find that adding new data to a plot doesn’t change the scale of the axes, so some points may end up outside of the current graph. Unfortunately, there’s no way to change the axis scale once the plot has been created, so you’ll have to start again and create a new plot. When you create a new plot, you can specify the x-axis and y-axis limits:

```r
plot(…, xlim = c(1, 10), ylim = c(1.5, 0))
```

This will plot some data … (if you are doing a box-and-whisker plot enter two variables, for CDF enter one) with an x-axis scale from 1 to 10 and a y-axis scale from 1.5 to 0 (note you can have the scale descending by putting the larger number first).

### Part 3: Mann-Whitney U test

Remember that the purpose is to test for differences in the central tendency of two non-normally distributed samples (and only two samples). To run the test, you will need to provide two numeric vector variables as the inputs.

The command is simple and has the same format as the t test or F test functions:

```r
wilcox.test(variable1, variable2)
```

You should almost always do two-tailed tests, but if you have strong prior expectations that lead to you to perform a one-tailed test (H_a is “greater than” or “less than”), this is how:

```r
wilcox.test(variable1, variable2, alternative = "g")
```

Alternative=“g” (or “greater”) is for H_a that variable 1 is greater than variable 2
Alternative=“l” (or “less”) is for H_a that variable 1 is less than variable 2