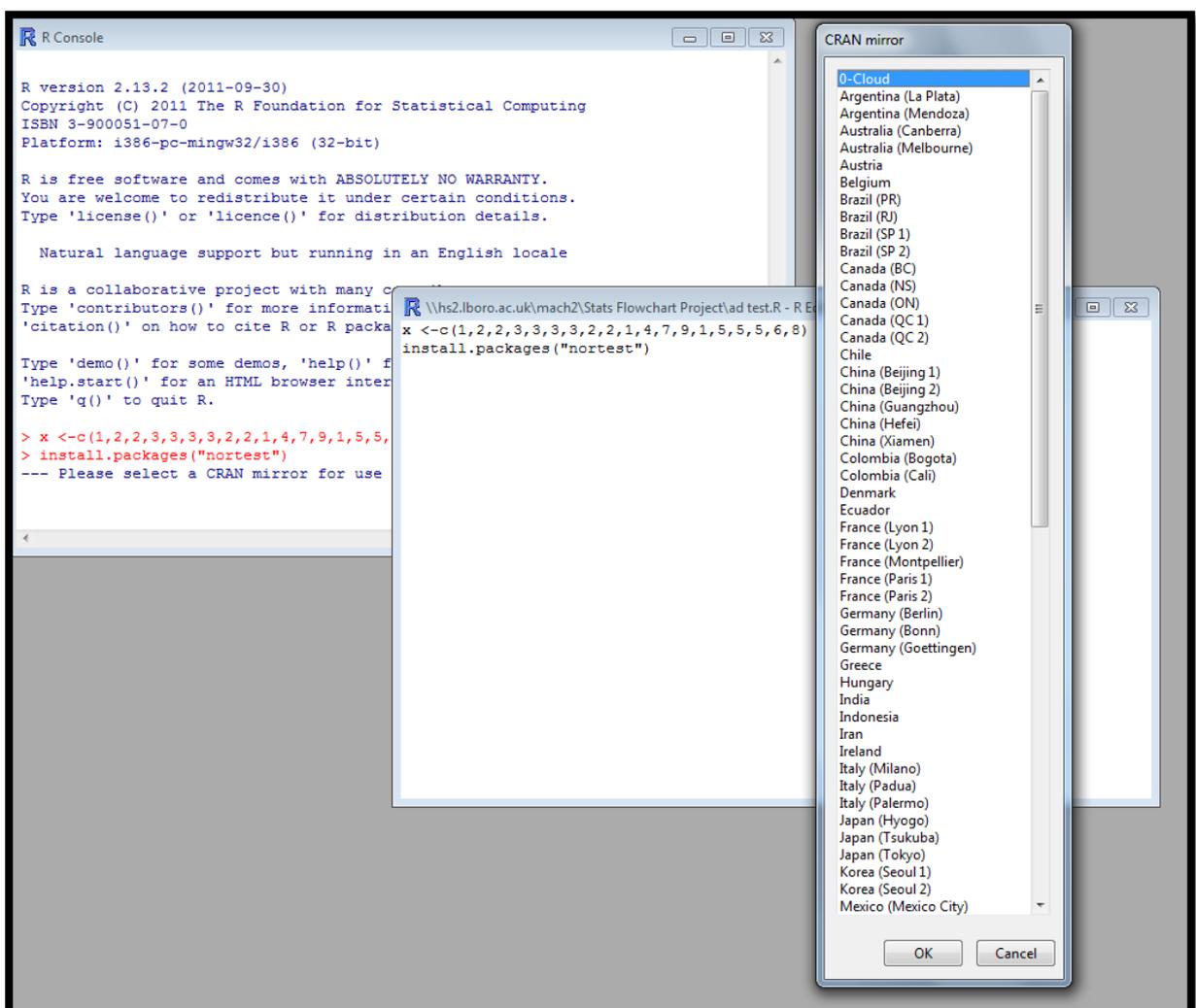


Computing an Anderson–Darling Test in R

- [Inputting data straight into R:](#)

When typing your data straight into R there is a simple command to generate an Anderson–Darling Test. For example if our data is: 1, 2, 2, 3, 3, 3, 3, 2, 2, 1, 4, 7, 9, 1, 5, 5, 5, 6, 8. Then we input the data in R using the `x <- c(1,2,2,3,3,3,3,2,2,1,4,7,9,1,5,5,5,6,8)` command (where x is what you're calling the data). In order to carry out an Anderson–Darling test in R you must first install the package “nortest” using the command `install.packages(“nortest”)`, as follows:



You must then select the country of choice. The following code will then appear:

```
R Console

Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x <-c(1,2,2,3,3,3,3,2,2,1,4,7,9,1,5,5,5,6,8)
> install.packages("nortest")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.ma.imperial.ac.uk/bin/windows/contrib/2.13/nortest_1.0.$
Content type 'application/zip' length 27623 bytes (26 Kb)
opened URL
downloaded 26 Kb

package 'nortest' successfully unpacked and MD5 sums checked

The downloaded packages are in
  C:\Users\mach2\AppData\Local\Temp\RtmpAVQIve\downloaded_packages
> |
```

You must then load the package using the command `library(nortest)` as follows:

```
R Console

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x <-c(1,2,2,3,3,3,3,2,2,1,4,7,9,1,5,5,5,6,8)
> install.packages("nortest")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.ma.imperial.ac.uk/bin/windows/contrib/2.13/nortest_1.0.$
Content type 'application/zip' length 27623 bytes (26 Kb)
opened URL
downloaded 26 Kb

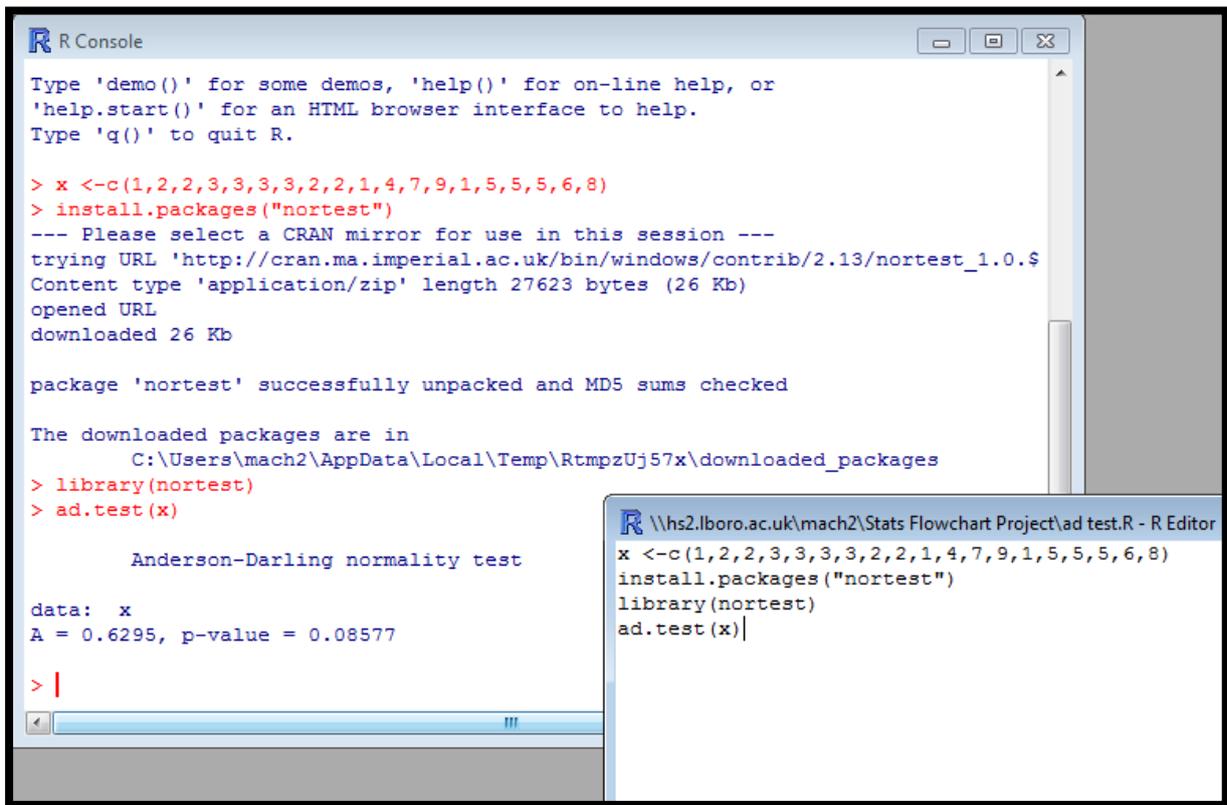
package 'nortest' successful

The downloaded packages are
  C:\Users\mach2\AppData

> library(nortest)
> |
```

```
R \\hs2.lboro.ac.uk\mach2\Stats Flowchart Project\ad test.R - R Editor
x <-c(1,2,2,3,3,3,3,2,2,1,4,7,9,1,5,5,5,6,8)
install.packages("nortest")
library(nortest)|
```

The command `ad.test(x)` can then be used to compute the Anderson–darling test as follows:



The image shows two overlapping windows from an R environment. The background window is the 'R Console', which displays the following text:

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x <-c(1,2,2,3,3,3,3,2,2,1,4,7,9,1,5,5,5,6,8)
> install.packages("nortest")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.ma.imperial.ac.uk/bin/windows/contrib/2.13/nortest_1.0.0.zip'
Content type 'application/zip' length 27623 bytes (26 Kb)
opened URL
downloaded 26 Kb

package 'nortest' successfully unpacked and MD5 sums checked

The downloaded packages are in
  C:\Users\mach2\AppData\Local\Temp\RtmpzUj57x\downloaded_packages
> library(nortest)
> ad.test(x)

      Anderson-Darling normality test

data:  x
A = 0.6295, p-value = 0.08577

> |
```

The foreground window is the 'R Editor', showing the code that was entered into the console:

```
x <-c(1,2,2,3,3,3,3,2,2,1,4,7,9,1,5,5,5,6,8)
install.packages("nortest")
library(nortest)
ad.test(x)
```

In R the test statistic is labelled as A instead of A^2 , and can then be compared to the critical value in order to determine normality. An alternative approach is to examine the p -value in the Anderson–Darling Test. If you are testing at the 95% significance level and our p -value is smaller than 0.05, you reject the null hypothesis which states that the sample data comes from a normal distribution, and hence you can conclude that the data is likely to be non-normal. In this example, at the 95% significance level, our p -value is greater than 0.05 so we do not reject the null hypothesis and we can conclude that our data is likely to follow a normal distribution.

A more detailed example:

Data set: 4, 6, 12, 19, 20, 22, 24, 27, 31, 31, 31, 32, 36, 38, 39, 39, 44, 45, 47, 47, 48, 49, 49, 53, 55, 56, 56, 60, 61, 77.

The image shows two overlapping R windows. The top window is the R Console, and the bottom window is the R Editor.

```
R Console
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x <-c(4,6,12,19,20,22,24,27,31,31,31,32,36,38,39,39,44,45,47,47,48,49,49,53,55)
> install.packages("nortest")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.ma.imperial.ac.uk/bin/windows/contrib/2.13/nortest_1.0.0.zip'
Content type 'application/zip' length 27623 bytes (26 Kb)
opened URL
downloaded 26 Kb

package 'nortest' successfully unpacked and MD5 sums checked

The downloaded packages are in
  C:\Users\mach2\AppData\Local\Temp\RtmpkKKeSd\downloaded_packages
> library(nortest)
> ad.test(x)

Anderson-Darling normality test

data:  x
A = 0.2129, p-value = 0.8386

> |

R Editor
\\hs2.lboro.ac.uk\mach2\Stats Flowchart Project\ad.test2.R - R Editor
x <-c(4,6,12,19,20,22,24,27,31,31,31,32,36,38,39,39,44,45,47,47,48,49,49,53,55,56)
install.packages("nortest")
library(nortest)
ad.test(x)
```

- [Inputting data from an excel file:](#)

When importing data from R we need to use the command `read.csv` as follows:

```
sl<-read.csv("U:/Stats Flowchart Project/Dataset.csv")
```

```
sl
```

```
attach(sl)
```

The text in red is where the excel file can be found in your documents, and 'sl' is what we have called our data. The same commands used in the examples above to install, and load the package "nortest" can then be used, and the command `ad.test(sl)` can then be implemented to compute the Anderson-Darling test.