Advanced Statistics

author: Bernhard Angele date: Class 2 supplement, October 9, 2014

New stuff we did with R this week

- Select a named entry of a list with `$`

```
my_new_list <- list(animal = "cat", ears = "pointy", sound = "meow")
my_new_list$animal
```

```
## [1] "cat"
```

```
my_new_list$sound
```

```
## [1] "meow"
```

More practical example

- Some functions output a list (although that may be cleverly hidden by the `print` command)
- To find out the **structure** of an object, use `str(object_name)`:

```
my_new_list <- list(animal = "cat", ears = "pointy", sound = "meow")
str(my_new_list)
```

```
## List of 3
##  $ animal: chr "cat"
##  $ ears  : chr "pointy"
##  $ sound : chr "meow"
```

Structure of the t.test() output

```
t_results <- t.test( rnorm(n = 10, mean = 3, sd = 1))
str(t_results)
```

```
## List of 9
##  $ statistic  : Named num 13.5
##   ..- attr(*, "names")= chr "t"
##  $ parameter  : Named num 9
##   ..- attr(*, "names")= chr "df"
##  $ p.value    : num 2.74e-07
##  $ conf.int   : atomic [1:2] 2.36 3.3
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num 2.83
##   ..- attr(*, "names")= chr "mean of x"
```

```
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "mean"
##  $ alternative: chr "two.sided"
##  $ method     : chr "One Sample t-test"
##  $ data.name  : chr "rnorm(n = 10, mean = 3, sd = 1)"
##  - attr(*, "class")= chr "htest"
```

Getting the CI from the t.test() output

```
t_results$conf.int
```

```
## [1] 2.357 3.303
## attr(,"conf.level")
## [1] 0.95
```

You can also extract elements by number (slicing):

```
t_results$conf.int[1]
```

```
## [1] 2.357
```

```
t_results$conf.int[2]
```

```
## [1] 3.303
```

Logical tests

- You can use `>`, `<`, `==`, `>=`, `<=` to perform a test. The result of a test is either `true` or `false`
- Is 3 part of the CI?

```
3 > t_results$conf.int[1] & 3 < t_results$conf.int[2]
```

```
## [1] TRUE
```

Count table

`table(variable)` will give you a table with the counts of each unique element in `variable`

```
test_array <- c(1,1,1,1,1,2,3,3,3,3,3,3,4,4)
table(test_array)
```

```
## test_array
## 1 2 3 4
## 5 1 6 2
```

While loops

These loops will keep executing a certain expression until a condition is `FALSE`

Example:

```
my_number <- 1
while(my_number < 4)
  {
  my_number <- my_number + 1
  print(my_number)
  }
```

```
## [1] 2
## [1] 3
## [1] 4
```

Something more practical: Running t-tests

- How do you actually run and report a t-test?
- Let's use a data set that's built into R: Student's sleep data
- Get some information on it by typing `?sleep`

The sleep data set

```
sleep
```

```
##    extra group ID
## 1    0.7     1  1
## 2   -1.6     1  2
## 3   -0.2     1  3
## 4   -1.2     1  4
## 5   -0.1     1  5
## 6    3.4     1  6
## 7    3.7     1  7
## 8    0.8     1  8
## 9    0.0     1  9
## 10   2.0     1 10
## 11   1.9     2  1
## 12   0.8     2  2
## 13   1.1     2  3
## 14   0.1     2  4
## 15  -0.1     2  5
## 16   4.4     2  6
## 17   5.5     2  7
## 18   1.6     2  8
## 19   4.6     2  9
## 20   3.4     2 10
```

What is this data type?

```
str(sleep)
```

```
## 'data.frame':    20 obs. of  3 variables:
##  $ extra: num  0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...
##  $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ID   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

- data.frames are fantastic!
- they combine properties of lists and vectors (or even matrices)
- closest equivalent to your Excel Spreadsheet
- every column has to have the same length (number of rows)
- but each column can be of a different type, e.g. numeric, character, or Factor (discrete variable)

Let's do the two-sample t-test

```
t.test(subset(sleep, group == 1)$extra, subset(sleep, group == 2)$extra)
```

```
##
##  Welch Two Sample t-test
##
## data:  subset(sleep, group == 1)$extra and subset(sleep, group == 2)$extra
## t = -1.861, df = 17.78, p-value = 0.07939
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.3655  0.2055
## sample estimates:
## mean of x mean of y
##      0.75      2.33
```

Let's assume the same patients tried both drugs

Do a pairwise t-test! Note the increase in power.

```
t.test(subset(sleep, group == 1)$extra, subset(sleep, group == 2)$extra, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  subset(sleep, group == 1)$extra and subset(sleep, group == 2)$extra
## t = -4.062, df = 9, p-value = 0.002833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4599 -0.7001
## sample estimates:
## mean of the differences
##                   -1.58
```

A very elegant way of specifying the t-test

```
t.test(formula = extra ~ group, data = sleep, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  extra by group
## t = -4.062, df = 9, p-value = 0.002833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4599 -0.7001
## sample estimates:
## mean of the differences
##                   -1.58
```

See your homework for instructions on how to get R to write your report for you!