Statistical analysis 12.04.17

```r
# Load datasets from R
library(datasets)

# Load the library we are going to use
library(data.table)
library(psych)
library(MASS)
```

Little reminder on comparisons

Element wise comparison, compares each element of the vector.

```r
a = c(1,2,3,4)

# OR is joining - use ONE |
b = a[((a>3) | (a<2))]
print(b)
```

```
## [1] 1 4
```

```r
# AND is the intersection - use ONE &
b = a[((a>3) & (a<2))]
print(b)
```

```
## numeric(0)
```

```r
######## Operators ########
# ==  equals to
# <=  smaller than
# =>  greater than
# !=  different than
# &&  Conditional and
# ||  conditional or
# |   elementwise or
# &   elementwise and
```

Check what your datasets

Go to : http://127.0.0.1:8617/help/library/datasets/html/00Index.html Or write : ?datasets then index bottom of the page

Categorical datasets

```r
print(HairEyeColor)
```

```
## , , Sex = Male
##
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    32   11    10     3
##   Brown    53   50    25    15
##   Red      10   10     7     7
##   Blond     3   30     5     8
##
## , , Sex = Female
##
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    36    9     5     2
##   Brown    66   34    29    14
##   Red      16    7     7     7
##   Blond     4   64     5     8
```

Explore the structure of the data. Here it is a table NOT a data.table

```r
print(str(HairEyeColor))
```

```
##  table [1:4, 1:4, 1:2] 32 53 10 3 11 50 10 30 10 25 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Hair: chr [1:4] "Black" "Brown" "Red" "Blond"
##   ..$ Eye : chr [1:4] "Brown" "Blue" "Hazel" "Green"
##   ..$ Sex : chr [1:2] "Male" "Female"
## NULL
```

Get the marginal count for one variable: the sum of all the counts by this variable

```r
eyes = margin.table(HairEyeColor, 2)
print(eyes)
```

```
## Eye
## Brown  Blue Hazel Green
##   220   215    93    64
```

```r
# convert to proportions
prop.table(eyes)
```

```
## Eye
##     Brown      Blue     Hazel     Green
## 0.3716216 0.3631757 0.1570946 0.1081081
```

```r
# test significantly differ from H0 equal proportions
chisq.test(eyes)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  eyes
## X-squared = 133.47, df = 3, p-value < 2.2e-16
```

```r
# test with an H0 with custom population proportions (p = c(...,...))
chisq.test(eyes, p=c(.41,.32,.15,.12))
```

```
##
##  Chi-squared test for given probabilities
```

```
##
## data:  eyes
## X-squared = 6.4717, df = 3, p-value = 0.09079
```

```
# Different structure for a data.table
dt = data.table(a = rep(23,4))
str(dt)
```

```
## Classes 'data.table' and 'data.frame':   4 obs. of  1 variable:
##  $ a: num  23 23 23 23
##  - attr(*, ".internal.selfref")=<externalptr>
```

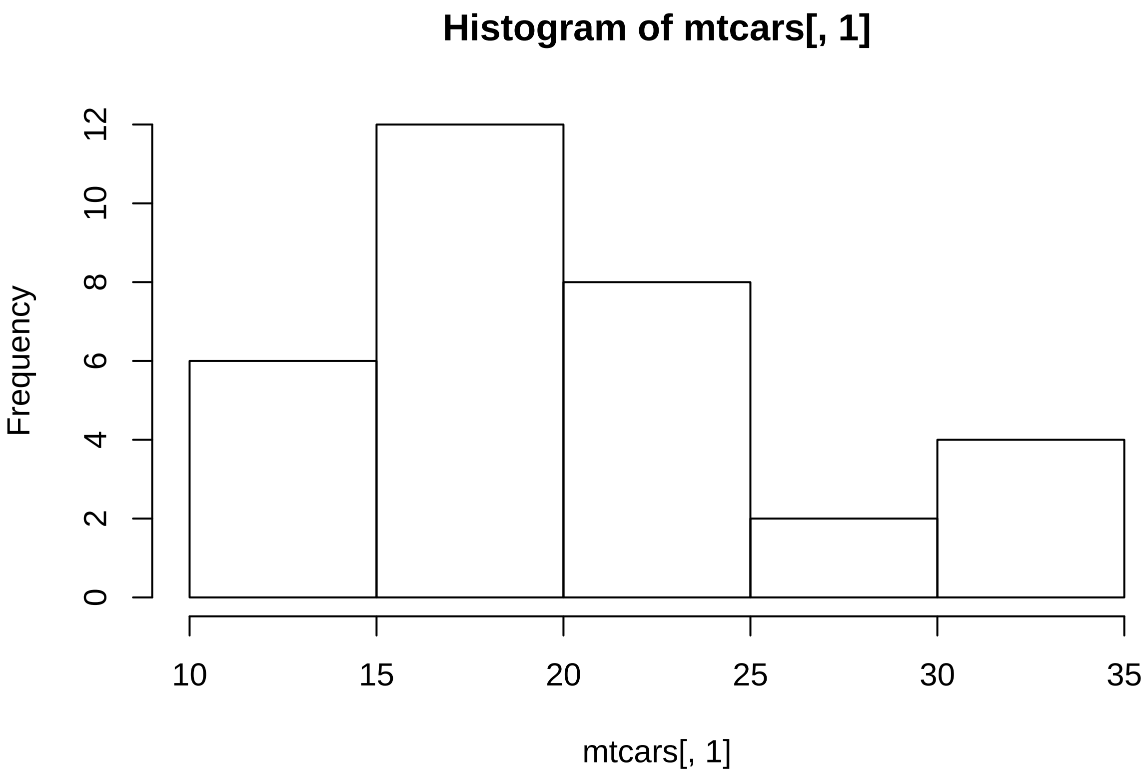
T-Test

SAMPLE MEAN - 2 SAMPLE MEAN - 2 tail - 1 tail

```
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

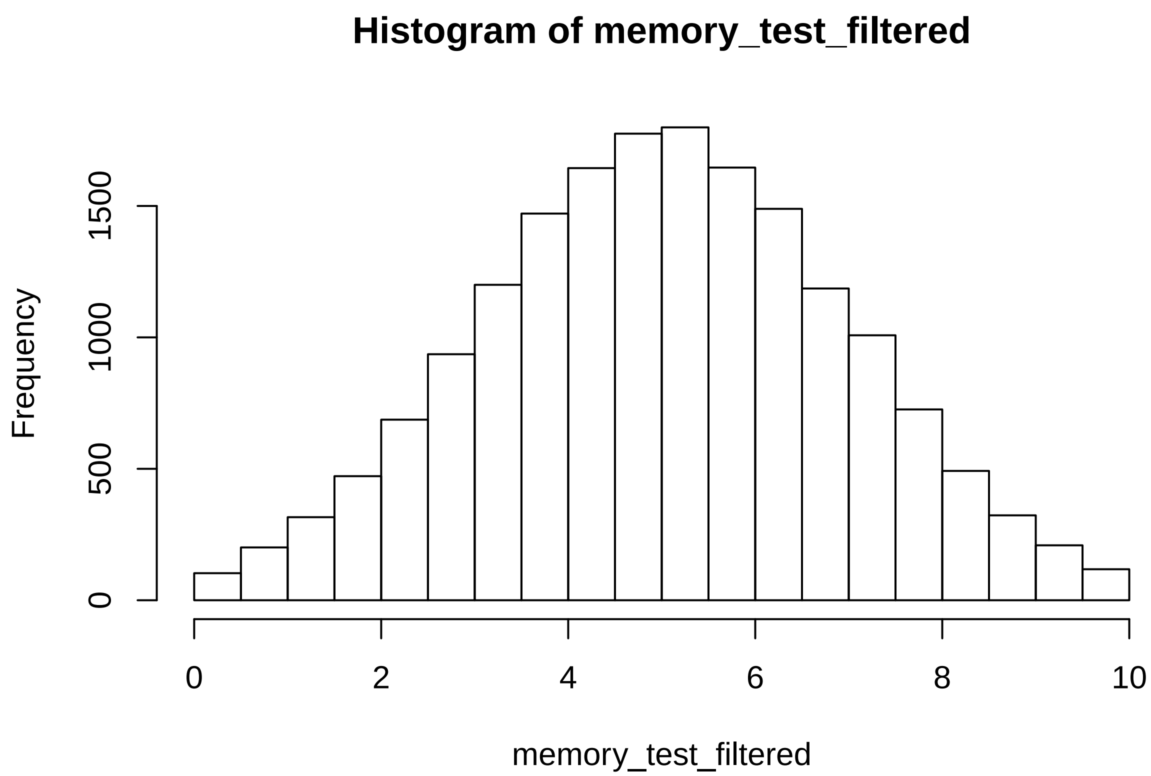
```
# extract mpg
hist(mtcars[,1])
```




```
# test significantly different than 20 -- get confidence interval of 0.8
t.test(mtcars[,1], mu = 20, conf.level = 0.8)
```

```
##
##  One Sample t-test
##
## data:  mtcars[, 1]
## t = 0.08506, df = 31, p-value = 0.9328
## alternative hypothesis: true mean is not equal to 20
## 80 percent confidence interval:
##  18.69549 21.48576
## sample estimates:
## mean of x
##  20.09062
```

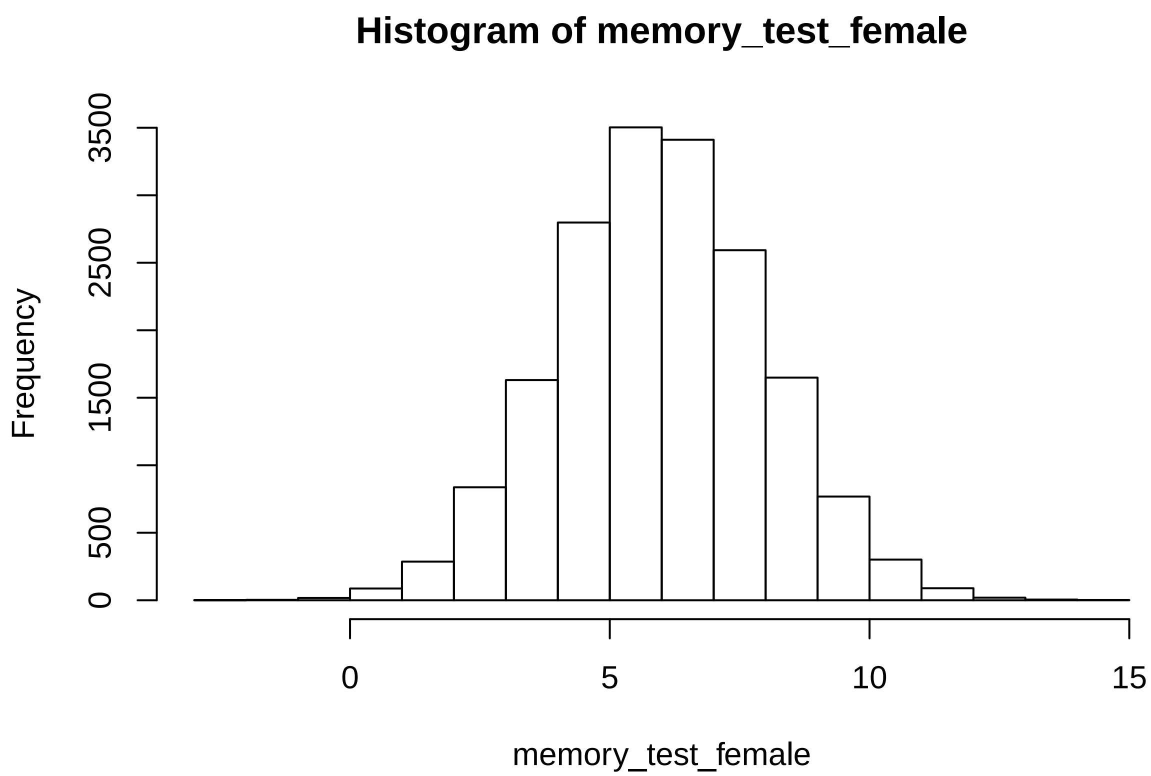
```
# create fake data
# Sample 18000 scores from a normal distribution of mean 5 and sd 2
memory_test = rnorm(18000, mean = 5, sd = 2)

# take only the values between 0 and 10 (you could use also the function ifelse())
memory_test_filtered = memory_test[memory_test>0 & memory_test<10]
hist(memory_test_filtered)
```




```r
# test significantly different than 5
t.test(memory_test_filtered, mu=5)
```

```
##
##  One Sample t-test
##
## data:  memory_test_filtered
## t = 1.8325, df = 17800, p-value = 0.06689
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  4.998163 5.054605
## sample estimates:
## mean of x
##  5.026384
```

```r
# sample the same amount of subjects this time centered on 6
memory_test_female = rnorm(18000, mean = 6, sd = 2)
hist(memory_test_female)
```




```r
describe(memory_test_female)
```

```
##    vars     n mean sd median trimmed mad   min   max range skew kurtosis
## X1    1 18000 5.98  2   5.96    5.97   2 -2.26 14.21 16.47 0.03     0.01
##      se
## X1 0.01
```

```r
# Filter it in the same way take only values between 0 and 10
memory_test_female_filtered = memory_test_female[memory_test_female>0 & memory_test_female<10]
hist(memory_test_female_filtered)
```


```r
# One tailed (alternative / alt = "greater")
t.test(memory_test_female_filtered, memory_test_filtered, alt = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  memory_test_female_filtered and memory_test_filtered
## t = 42.001, df = 35354, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.812925      Inf
## sample estimates:
## mean of x mean of y
##  5.872444  5.026384
```

PAIRED T-TEST

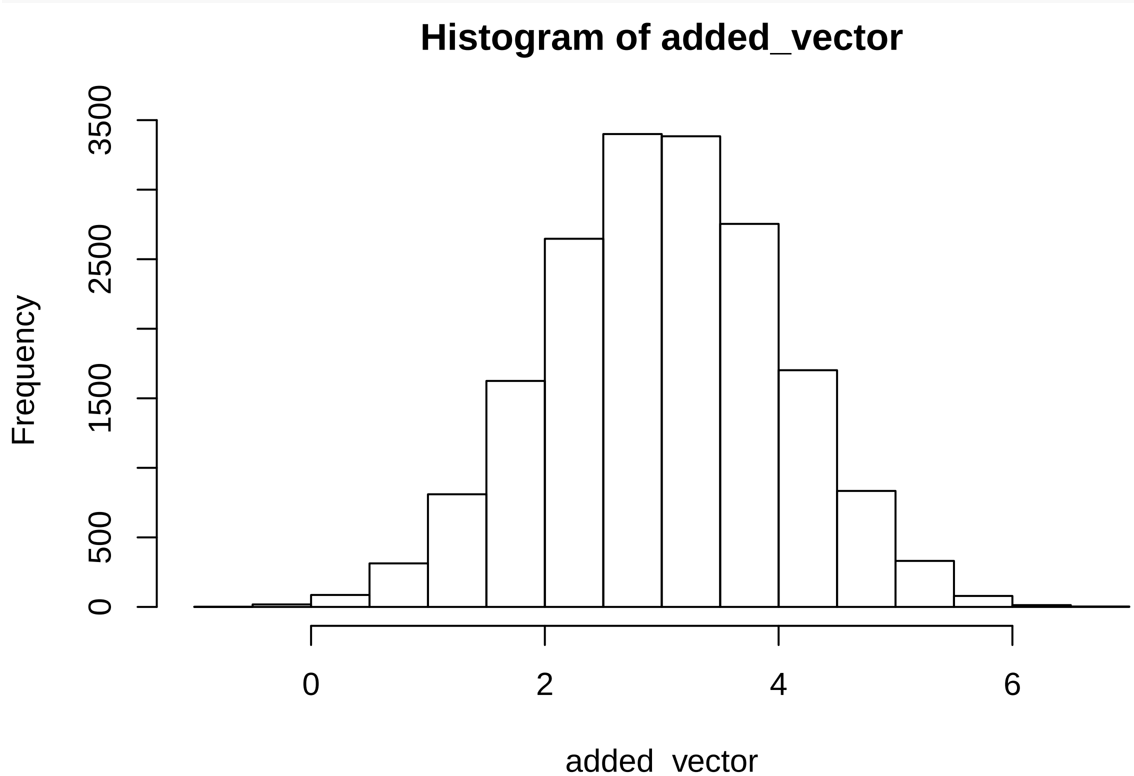
T1 -> Intervention -> T2 (repeated measure)

```r
## GROUP T1 - first test before the intervention
# sample fake data centered on 2
memory_test = rnorm(18000, mean = 2, sd = 1)

# a lot of the data is below 0
# but since we are going to use a paired t-test we want to keep the same amount of
# subjects and not delete them

# We need to REPLACE the values for subjects below 0
# If we replace all the values below 0 by 0 it will strongly alter the shape of the distribution
```

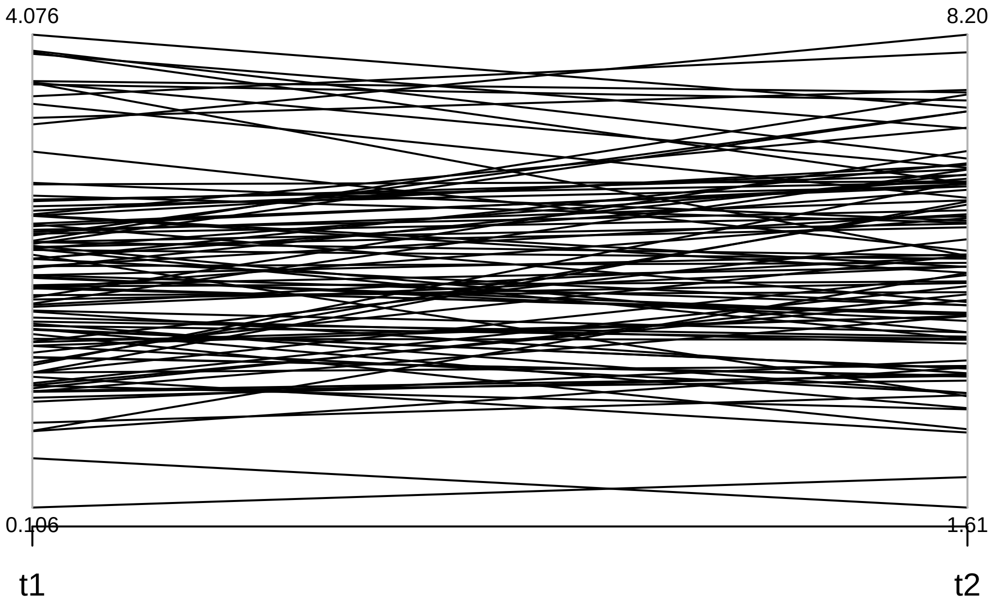
```r
memory_test[memory_test<0] = 0

# look at the first bin on the left is a lot higher than its symetric bin on the other side
# The distribution is skewed, which is not bad in itself, you just have to account for it by using
# robust statistics if the skeweness is too important
hist(memory_test)
```


```r
# Another way, to smooth your distribution a bit could be to replace by 0 + a random value between 0 an
# (this is fake data -- not something you will want to do with real data)
memory_test = rnorm(18000, mean = 2, sd = 1)

# get the number of subjects below 0
size_below_zero = length(memory_test[memory_test<0])

# get a vector of values between 0 and 3
sampled_vector = seq(from = 0, to = 3, by = 0.01)

# sample from that vector the same amount of time that there are subject below 0 in the memory test
sample_for_replacement = sample(sampled_vector, size = size_below_zero,replace = T)

# replace the values
memory_test[memory_test<0] = sample_for_replacement
hist(memory_test)
```


```
## GROUP T2 cw in between !
# We create the second vector by adding a random number sampled from a normal distribution
# with mean of 3 and sd of 1
added_vector = rnorm(length(memory_test), mean = 3, sd = 1)
hist(added_vector)
```



```r
# our intervention has a positive effect so we add this positive vector
memory_test_t2 = copy(memory_test) + added_vector

# we build our data.table with the two samples
pairs = data.table(t1 = memory_test, t2 = memory_test_t2)

# Check visually evolution of the 100 first subjects (parcoord is part of the MASS package)
parcoord(pairs[0:100,], var.label = T)
```




```r
# Paired T test with H0 difference is not significantly different than 0
t.test(memory_test_t2, memory_test , paired = T)
```

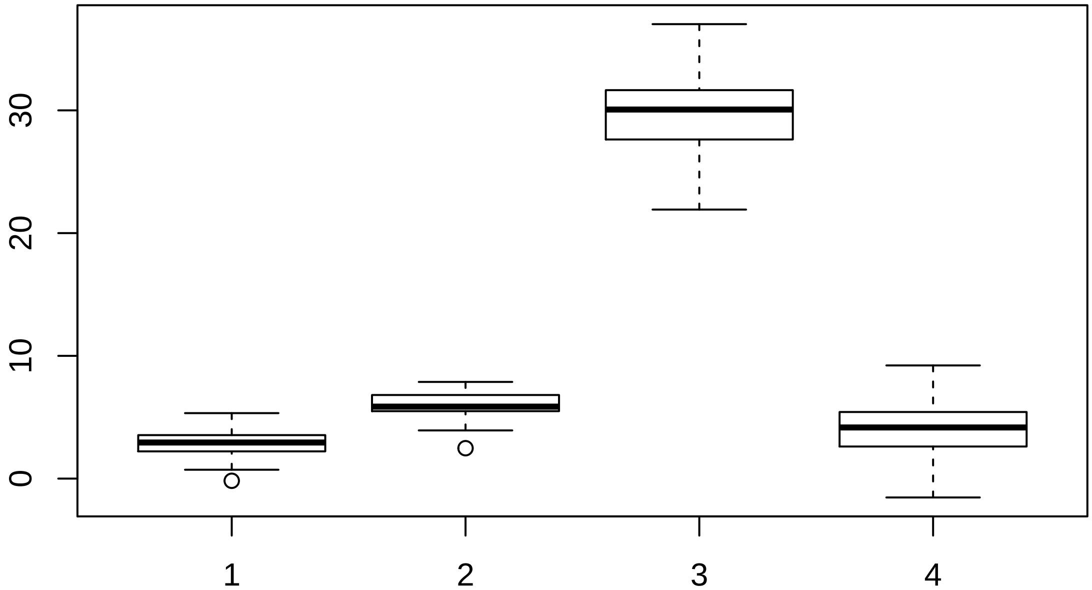
```
##
##  Paired t-test
##
## data:  memory_test_t2 and memory_test
## t = 402.1, df = 17999, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.999473 3.028859
## sample estimates:
## mean of the differences
##                3.014166
```

ANOVA

One variable

```r
# Check that the number of days of holidays between countries is different
chinese = rnorm(n = 150, mean = 3, sd = 1)
japan = rnorm(n = 30, mean = 6, sd = 1)
danemark = rnorm(n = 20, mean = 30, sd = 4)
usa = rnorm(n = 400, mean = 4, sd = 2)
```

```r
boxplot(chinese, japan, danemark, usa)
```




```r
# We need an indicator variable, we build it by hand
group = c(rep('chinese', 150),rep('japanese', 30), rep('danish', 20), rep('american', 400))

# This is our value variable
values = c(chinese, japan, danemark, usa)

# We create a data.table with those two corresponding vector
dt = data.table(g = group, v = values)
print(dt)
```

```
##             g        v
##   1:  chinese 0.832583
##   2:  chinese 3.034556
##   3:  chinese 3.056769
##   4:  chinese 1.475493
##   5:  chinese 3.665782
##  ---
## 596: american 2.864510
## 597: american 5.224677
## 598: american 4.691604
## 599: american 3.713914
## 600: american 6.897616
```

```r
# Now we can perform our anova test
aov_results = aov(dt, formula = v ~ g)
summary(aov_results)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## g             3  13316    4439    1275 <2e-16 ***
## Residuals   596   2075       3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# We can also test group difference one to one using Tukey
# Here they are all significant
tukey_results = TukeyHSD(aov_results)
print(tukey_results)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = v ~ g, data = dt)
##
## $g
##                         diff         lwr         upr p adj
## chinese-american   -1.153972  -1.6142621  -0.6936817 0e+00
## danish-american    25.764004  24.6624503  26.8655575 0e+00
## japanese-american   1.874414   0.9643551   2.7844733 9e-07
## danish-chinese     26.917976  25.7735442  28.0624074 0e+00
## japanese-chinese    3.028386   2.0668713   3.9899010 0e+00
## japanese-danish   -23.889590 -25.2774168 -22.5017625 0e+00
```

```r
# To use the function stack() the number of observation need to be similar
chinese = rnorm(n = 50, mean = 3, sd = 1)
japan = rnorm(n = 50, mean = 6, sd = 1)
danemark = rnorm(n = 50, mean = 30, sd = 4)
usa = rnorm(n = 50, mean = 4, sd = 2)
stack_variable = stack(data.frame(cbind(chinese, japan, danemark, usa)))
result = aov(stack_variable, formula = values ~ ind)
summary(result)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## ind           3  25955    8652    1844 <2e-16 ***
## Residuals   196    920       5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```