Why R and Python?

- **Purpose**: very general + data analysis, stats.
- **Price**: excludes Matlab and SAS.
- **Popularity** (= more libraries, more help, portability): Ruby, Python, Perl, R, Java, C/C++.
- **High-level** language (“easier”), interactive: Ruby, Python, Perl, R.
  Perl is kind of old-fashioned. We could use Ruby.
- **Functions**: we need R libraries for stats and plots. Another language for anything else.
- **Speed**: choose C only if really necessary. For us usually speed is not an issue.

About execution speed

	Fortran GCC 4.5.1	Julia 12b1d5a7	Python 2.7.3	Matlab R2011a	Octave 3.4	R 2.14.2	JavaScript V8 3.6.6.11
fib	0.28	1.97	46.03	1587.03	2748.74	275.63	2.09
parse_int	9.22	1.72	25.29	846.67	7364.87	353.48	2.55
quicksort	1.65	1.37	69.20	133.46	3341.94	708.76	4.95
mandel	0.76	1.45	34.88	74.61	988.74	184.71	7.62
pi_sum	1.00	1.00	33.64	1.46	457.26	253.45	1.12
rand_mat_stat	2.23	1.95	29.01	7.71	31.04	12.66	5.53
rand_mat_mul	1.14	1.00	1.75	1.08	1.93	9.58	45.82

Figure: benchmark times relative to C (smaller is better).

From http://julialang.org/

- See that languages have specificities
- R is intended to perhaps be replaced by julia later but is the best alternative at the moment for statistics.

Python

```
# script.py

from module import function

f1 = open("file1.txt", "r")
f2 = open("file2.txt", "w")

header = f1.readline()

for line in f1:
  split_line = line.strip("\n").split("\t")
  u = float(split_line[2])
  v = 2*u
  print "value:", v
  f2.write(v)

f1.close(); f2.close()
```

```
help("function")
execfile(script.py)
exit()

$ sudo easy_install package

$ python script.py
```

R

```
# script.R

library(package)

d1 = read.table("file1.txt", header=TRUE)

s1 = d1$score1; s2 = d1$score2


pdf("my_plot.pdf")
  plot(s1,s2)
dev.off()

m1 = mean(s1); m2 = mean(s2)
cat("Mean scores: ", m1, m2)
d2 = data.frame(mean1=m1, mean2=m2)
write.table(d2, "file2.txt", sep="\t")
```

```
help("function"); example("function")
source(script.R)
quit()

install.packages("package")

$ Rscript script.R
```

"$" before a command means it is a console input, not in the language itself.