

Introduction to R

1. Why R?

“R is really important to the point that its hard to overvalue it. It allows statisticians to do very intricate and complicated analyses without knowing the blood and guts of computing systems.” — Daryl Pregibon, a research scientist at Google

Advantages:

- Open source (SAS and Matlab is not)
- Abundant statistical packages
- Simplicity (Don't need to worry too much about variable type, memory allocation, loops. Very easy for beginners. The software that speaks human language.)
- Data visualization (ggplot2, googleVis, rChart...)
- Nice IDE (Rstudio, Eclipse)

Disadvantages: it could be Slow (Need to be smart while coding) * Interpreted language * Memory management

2. Memory Management

How much memory is used?

Function `object_size()` in the 'pryr' package.

The package is built under R 3.1.0. Every vector occupies at least 40 bytes of memory. + metadata, basic type: 4 bytes + 2 pointers to the previous and next object in memory respectively: 8 bytes each. + 1 pointer to the attributes: 8 bytes. On top of this, the object takes more memory to store the data.

2. Memory usage and garbage collection: `mem_used()`, `mem_changed()`

Objects point to the same location does not increase memory usage.

```
require(pryr)
```

```
## Loading required package: pryr
```

```
x <- 1:1e6  
object_size(x)
```

```
## 4 MB
```

```
y <- x  
object_size(y)
```

```
## 4 MB
```

```
object_size(x, y)
```

```
## 4 MB
```

Garbage collection automatically releases memory when the object is no longer used.

```
mem_change(rm(x))
```

```
## 13.6 kB
```

There is no memory change, because y still points to that object.

```
mem_change(rm(y))
```

```
## -4 MB
```

Modification in place

Primitive vs non-primitive functions

```
x <- 1:1e6  
object_size(x)
```

```
## 4 MB
```

```
y <- list(x,x,x)  
object_size(y)
```

```
## 4 MB
```

```
y <- c(x,x,x)  
object_size(y)
```

```
## 12 MB
```

Loops could be slow (Examples on Thursday)

2. Rstudio

Download link (<http://www.rstudio.com/ide/download/>) Or, download the preview version download link (<http://www.rstudio.com/ide/download/preview>)

- Short intro of Rstudio
- Control R version
- vim editor
- Change theme
- Version control with Git or SVN
- Create a project, an R script
- Save workspace and changed script
- Documents
- pdf: LaTeX, Sweave, Knitr
- html: R markdown, Rpres, Notebook
- Web application: Shiny

Useful shortcut for writing R script: - Ctrl+Space or Tab, give you more information of the function. - Tab can also give you the details of the argument. - Ctrl+Shift+c comment out one line or a block of highlighted code. - Ctrl+Shift+c remove the “#” on a block of commented code. - While executing code, environment tab shows the objects in your current environment - Executing the code: Ctrl+R or Ctrl+Enter or Source (useful for editing function: Source on save) - Debugging (more details comes later)

A demo of R markdown: - Insert title “====” - Insert section “#”, subsection “##”, subsection “###”, keep going - Insert bullet points “-” - Insert list “1.” - Insert list within list: Tab+“1.” - Insert inline equation α - Insert display equations

$$Y \sim Normal(\mu, \sigma^2)$$

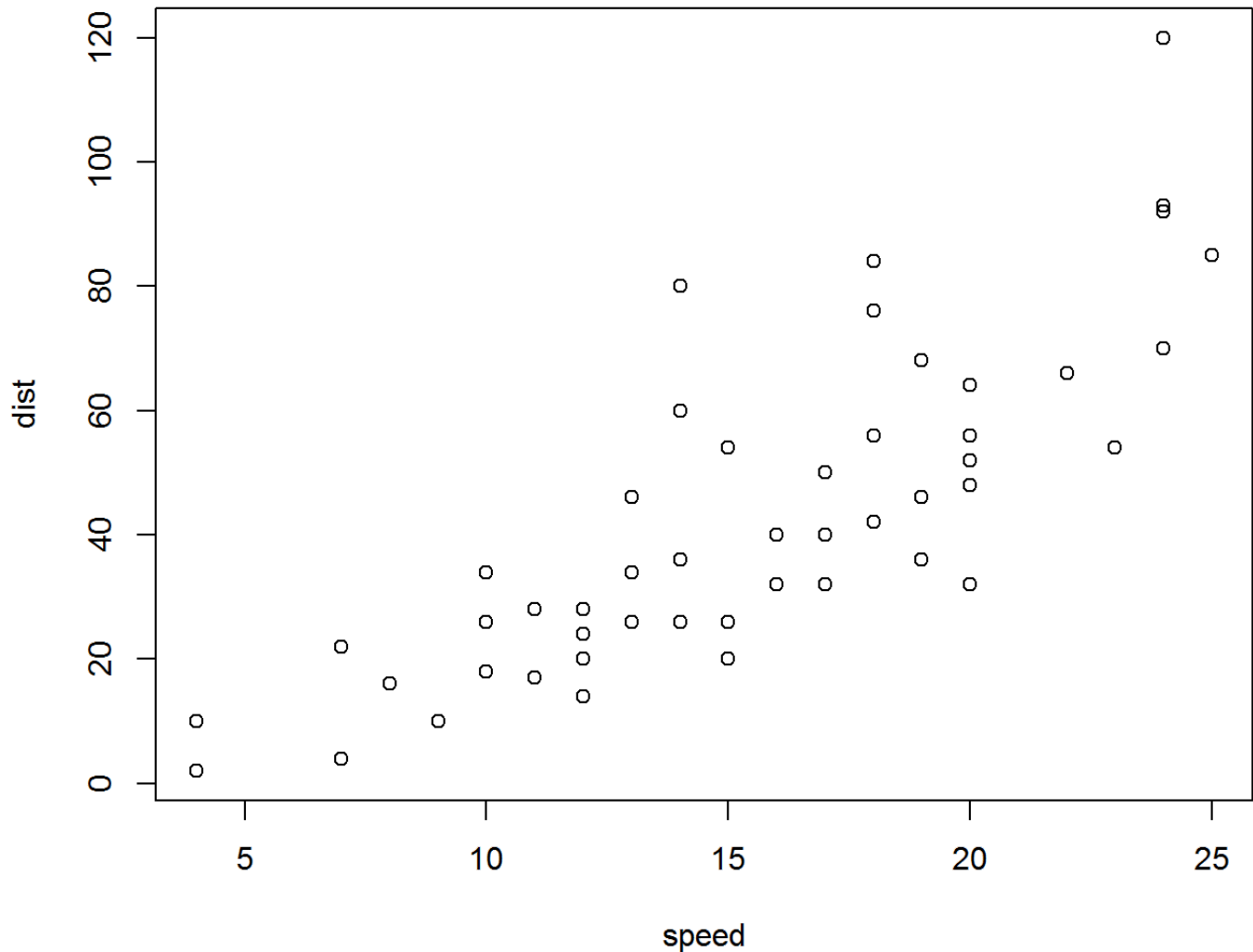
- Insert code { } - Insert R code " {r, cache=TRUE, cache.path="."} ", " " - Insert R plot " {r fig.width=7, fig.height=6} ", "" - Insert and execute R code

```
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0    Min.   :  2
## 1st Qu.:12.0    1st Qu.: 26
## Median :15.0    Median : 36
## Mean   :15.4    Mean   : 43
## 3rd Qu.:19.0    3rd Qu.: 56
## Max.   :25.0    Max.   :120
```

- Insert plot

```
plot(cars)
```



3. Good Coding Habits

- Indentation
- Assignment (use <-, not =)
- Line Length (80 characters preferred)
- Comment your code!!!
- Naming Convention: be consistent!

In all the 4411 packages on CRAN, there is a total of 339032 parameter names and 76176 function names.

– all lower case

```
```searchpaths``` ...
```

– period separated

```
```as.numeric``` , ```read.table``` ...
```

– underscore separated

```
``package_version`` ...
```

– lower camel case (suggested)

```
``colSums``, ``sessionInfo`` ...
```

– Upper camel case

```
``Vectorize``, ``NextMethod`` ...
```