Data Analysis in R

@dustinvtran
Engineering and Applied Sciences @
Harvard University

1. introduction

What is R?

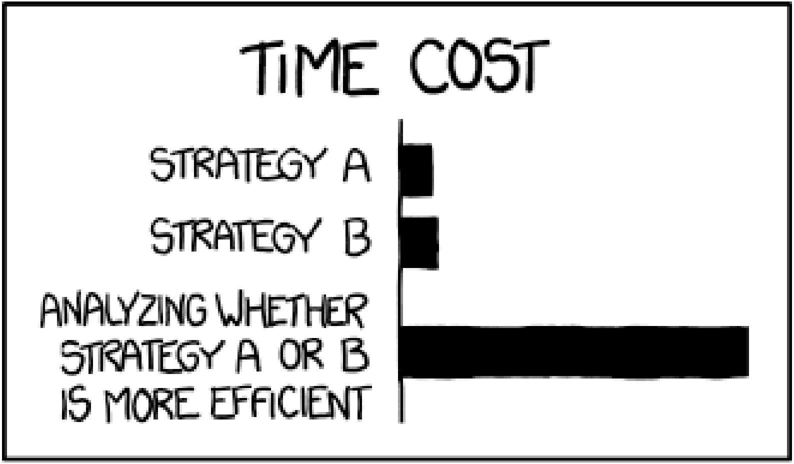
- R is a language developed for statistical computing and visualization

What is R?

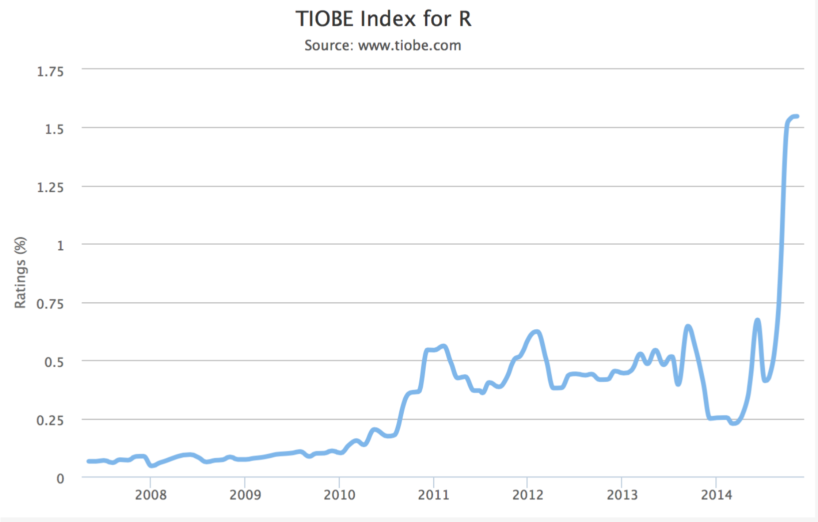
- R is a language developed for statistical computing and visualization
- It is free and open source

What is R?

- R is a language developed for statistical computing and visualization
- It is free and open source
- It is a dynamic, lazy, functional, and object-oriented language



THE REASON I AM SO INEFFICIENT


TIOBE Index for R
Source: www.tiobe.com
Ratings (%)
1.75
1.5
1.25
1
0.75
0.5
0.25
0
2008
2009
2010
2011
2012
2013
2014

R…

- has an enormous number of packages for statistical modelling, machine learning, visualization, and importing and manipulating data

R…

- has an enormous number of packages for statistical modelling, machine learning, visualization, and importing and manipulating data
- is designed to interface with high-performance computing languages such as Fortran and C++.

Bottlenecks

- The biggest bottleneck in data analysis is cognitive.

Bottlenecks

- The biggest bottleneck in data analysis is cognitive.
- You need tools (domain specific languages) to help you define the problem and express solutions programmatically.

2. fundamentals

```
2 + 2
2 * pi
7 + runif(1)
3^4

sqrt(4^4)
log(10)
log(100, base=10)

23 %% 2 # 23 mod 2
23 %/% 2 # floor(23/2)
5e9 * 1e3 # 5000000000 * 1000
```

```r
val <- 3
val
## [1] 3
print(val)
## [1] 3

val = 1:6
val
## [1] 1 2 3 4 5 6
```

R objects

- Vector: vector of some type (all entries are same type)

```r
# numeric
nums <- c(1.1, 3, -5.7)
devs <- rnorm(2)
devs
## [1] 1.8469193  0.4091781

# integer
ints <- c(1L, 5L, -3L)
ints
## [1] 1 5 -3
```

```
# character
chars <- c(’arthur’, ”marvin’s”,
           ”marvin\”s”)
chars
## [1] ”arthur” ”marvin’s” ”marvin\”s”

# logical
bools <- c(TRUE, FALSE, TRUE)
bools
## [1] TRUE FALSE TRUE
```

```r
vals <- seq(2, 12, by=2)
vals
## [1]  2  4  6  8 10 12
vals[3]
## [1]  6
vals[3:5]
## [1]  6  8 10
vals[c(1, 3, 6)]
## [1]  2  6 12
vals[-c(1, 3, 6)]
## [1]  4  8 10
vals[c(rep(TRUE, 3), rep(FALSE, 4))]
## [1]  2  4  6
```

```r
set.seed(42)
vals <- rnorm(3)
vals
## [1]  1.3709584 -0.5646982  0.3631284

vals[1:2] <- 0
vals
## [1] 0.0000000 0.0000000 0.3631284

vals[vals != 0] <- 5
vals
## [1] 0 0 5
```

```
vec1 <- 1:3
vec2 <- 3:5
vec1 + vec2
## [1] 4 6 8
vec1 * vec2
## [1] 3 8 15
vec1 >= vec2
## [1] FALSE FALSE FALSE
vec1 <= 3
## [1] TRUE TRUE TRUE
```

R objects

- Vector: vector of some type (all entries are same type)
- Matrix: matrix of some type (all entries are same type)

```
mat <- matrix(1:9, nrow = 3)
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
dim(mat)
class(mat)
t(mat) %*% mat
```

R objects

- Vector: vector of some type (all entries are same type)
- Matrix: matrix of some type (all entries are same type)
- Data frame: collection of columns (each column can be a different type)

```
dat <- data.frame(ints=1:3,
  chars=c(”hello”, ”world”, ”foo”))
dat
##     ints chars
## 1      1 hello
## 2      2 world
## 3      3   foo
```

R objects

- Vector: vector of some type (all entries are same type)
- Matrix: matrix of some type (all entries are same type)
- Data frame: collection of columns (each column can be a different type)
- List: collection of objects

```
list(stuff = 3,
     mat = matrix(1:4, nrow = 2),
     moreStuff = ”china”,
     list(5, ”bear”))
```

```
help(lm)
?lm
```

3. demo

4. closer

Resources

Guides

- Text: Hadley Wickham's "Advanced R"
- Videos: 2013 R bootcamp at UC Berkeley
- Interactive: DataCamp

Community & Help

- mailing lists
- #rstats
- useR!
- Stack Overflow, Google, Github, ...

Resources

Guides

- Text: Hadley Wickham's "Advanced R"
- Videos: 2013 R bootcamp at UC Berkeley
- Interactive: DataCamp

Help/Community

- mailing lists
- #rstats
- useR!
- Stack Overflow, Google, Github, ...

@dustinvtran • dustinvtran.com • dtran@g.harvard.edu