

Overview: R

Workshop on ggplot

Lokesh Mano • 28-Apr-2021

NBIS, SciLifeLab

Contents

- Demo: I/O
- Special operator
- Some important functions

Reading files

- Errors while reading in files!
- Demo of things that could go wrong when reading in files into R
- Demo on using reserved variables like `T`, `F`, `character` and many others ...
- How can you check if something is a reserved variable?

Special operator

- `%>%`
 - from the `dplyr` package
 - works like a pipe

```
read.table("data/counts_raw.txt", header = T, row.names = 1, sep = "\t") %>%  
  head(6)
```

```
##           Sample_1 Sample_2 Sample_3 Sample_4 Sample_5 Sample_6 Sample_7  
## ENSG00000000000003      321      303      204      492      455      359      376  
## ENSG00000000000005         0         0         0         0         0         0         0  
## ENSG000000000000419      696      660      472      951      963      689      706  
## ENSG000000000000457        59        54        44       109        73        66        60  
## ENSG000000000000460      399      405      236      445      454      374      316  
## ENSG000000000000938         0         0         0         0         0         1         0  
##           Sample_8 Sample_9 Sample_10 Sample_11 Sample_12  
## ENSG00000000000003      523      450      950      760      1436  
## ENSG00000000000005         0         0         0         0         0  
## ENSG000000000000419     1041      796     1036      789     1413  
## ENSG000000000000457      125       74      108      115      174  
## ENSG000000000000460      505      398      141      168      259  
## ENSG000000000000938         0         0         1         0         0
```

Special operator

- `%>%`
 - from the `dplyr` package
 - works like a pipe

```
read.table("data/counts_raw.txt", header = T, row.names = 1, sep = "\t") %>%  
  head(6) %>%  
  rownames_to_column(var = "Gene")
```

```
##           Gene Sample_1 Sample_2 Sample_3 Sample_4 Sample_5 Sample_6  
## 1 ENSG000000000003      321      303      204      492      455      359  
## 2 ENSG000000000005         0         0         0         0         0         0  
## 3 ENSG000000000419      696      660      472      951      963      689  
## 4 ENSG000000000457        59        54        44       109        73        66  
## 5 ENSG000000000460      399      405      236      445      454      374  
## 6 ENSG000000000938         0         0         0         0         0         1  
## Sample_7 Sample_8 Sample_9 Sample_10 Sample_11 Sample_12  
## 1       376       523       450       950       760      1436  
## 2         0         0         0         0         0         0  
## 3       706      1041       796      1036       789      1413  
## 4        60       125        74       108       115       174  
## 5       316       505       398       141       168       259  
## 6         0         0         0         1         0         0
```

Special operator

- `%>%`
 - from the `dplyr` package
 - works like a pipe

```
read.table("data/counts_raw.txt", header = T, row.names = 1, sep = "\t") %>%  
  head(1) %>%  
  rownames_to_column(var = "Gene") %>%  
  gather(Sample_ID, count, -Gene)
```

```
##           Gene Sample_ID count  
## 1 ENSG000000000003 Sample_1  321  
## 2 ENSG000000000003 Sample_2  303  
## 3 ENSG000000000003 Sample_3  204  
## 4 ENSG000000000003 Sample_4  492  
## 5 ENSG000000000003 Sample_5  455  
## 6 ENSG000000000003 Sample_6  359  
## 7 ENSG000000000003 Sample_7  376  
## 8 ENSG000000000003 Sample_8  523  
## 9 ENSG000000000003 Sample_9  450  
## 10 ENSG000000000003 Sample_10 950  
## 11 ENSG000000000003 Sample_11 760  
## 12 ENSG000000000003 Sample_12 1436
```

Tidyr or dplyr functions

- `gather()`
 - converts wide to long format
 - `key` is usually what you measure: `-Gene`
- `select()`
 - you can choose which columns you want,



Thank you. Questions?

R version 4.0.5 (2021-03-31)

Platform: x86_64-pc-linux-gnu (64-bit)

OS: Ubuntu 18.04.5 LTS

Built on: 📅 28-Apr-2021 at 🕒 14:58:59

2021 • SciLifeLab • NBIS