

# We need to talk about R

Jessica Temporal  
<https://github.com/jtemporal/>



# Me and R



[www.r-project.org/](http://www.r-project.org/)



<http://bioconductor.org/>



<http://bioinformatics.fmrp.usp.br/curso2014/>

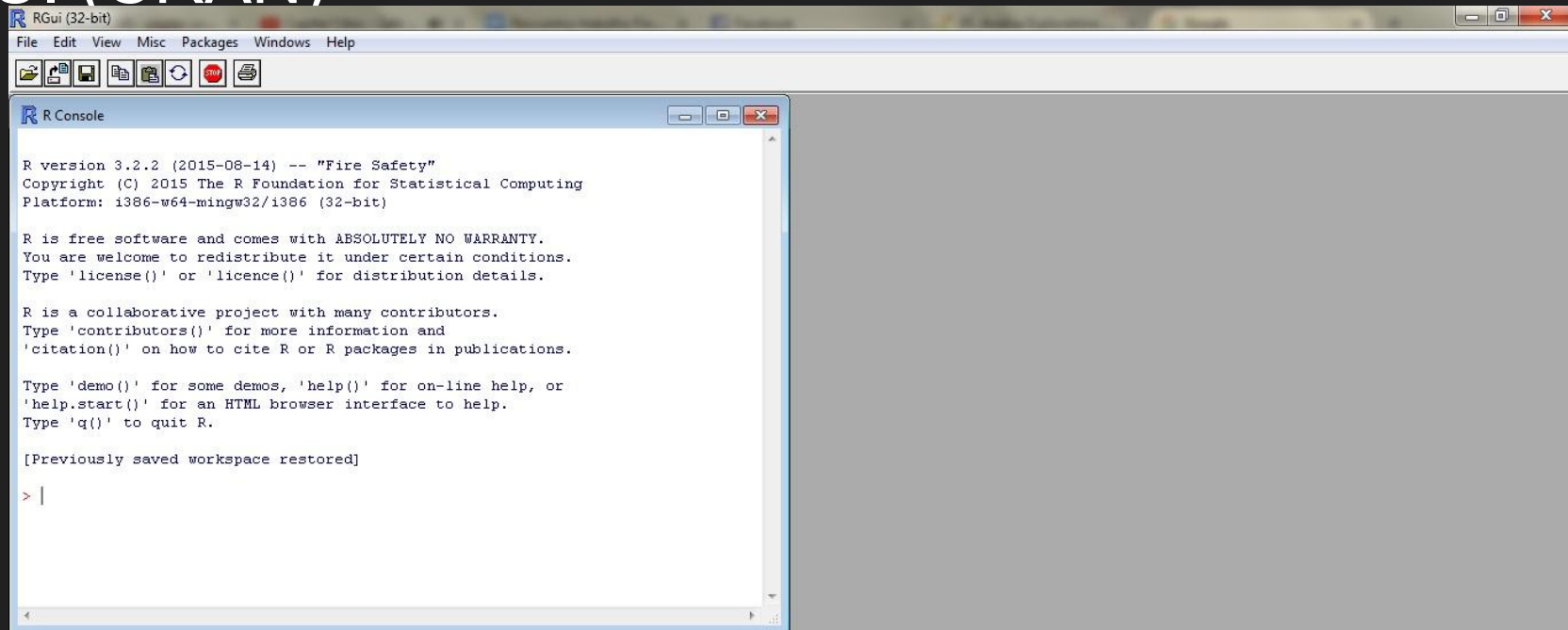
# R

- Powerful calculator
- Interpreted
- Objected-oriented but flexible
- Biologists and statisticians sweetheart :)
- Can integrate with other languages \o/
- Runs in: Linux, Unix, Mac and Windows

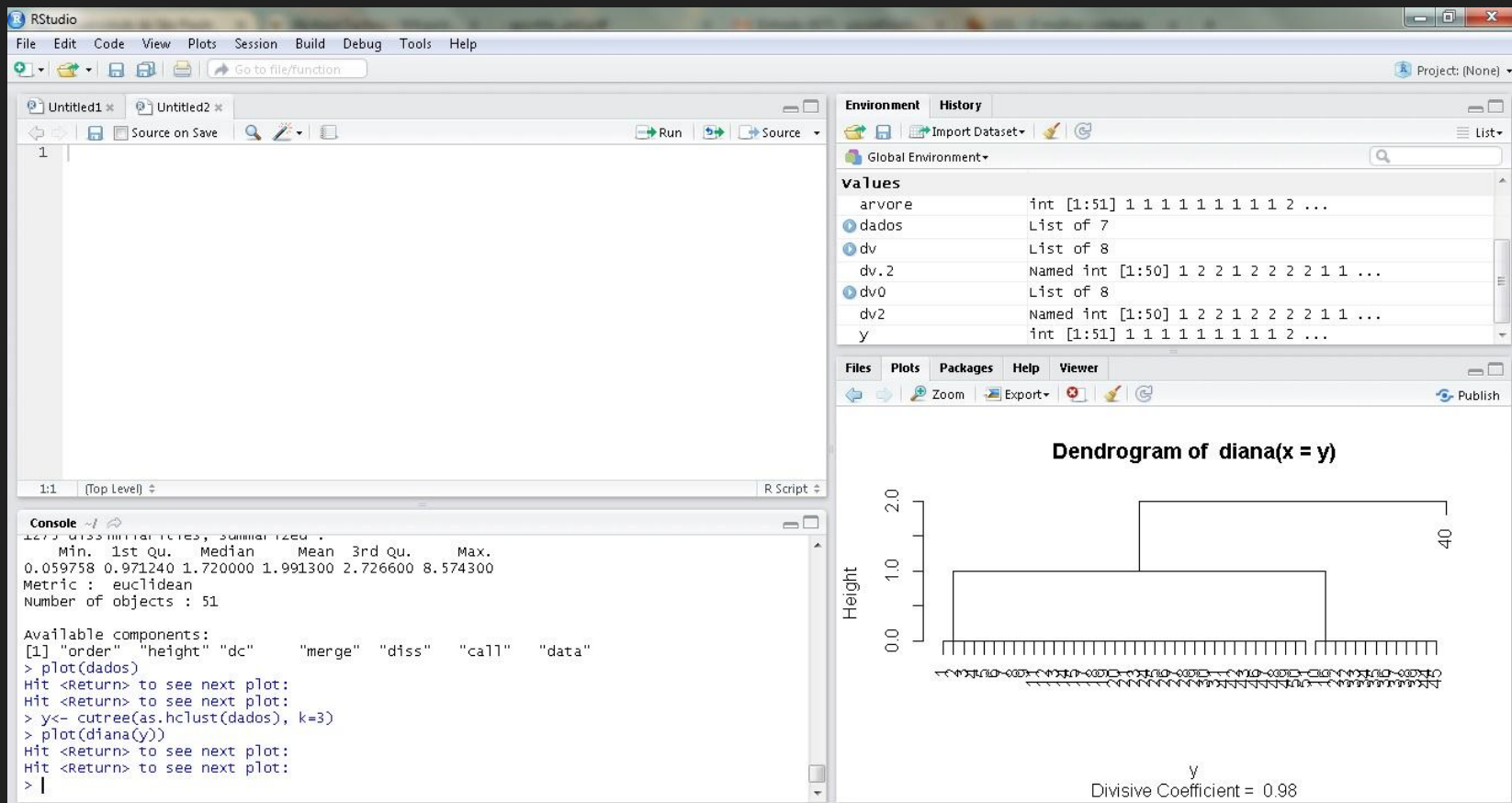
# Interfaces

- R - CRAN
- RStudio
- Terminal

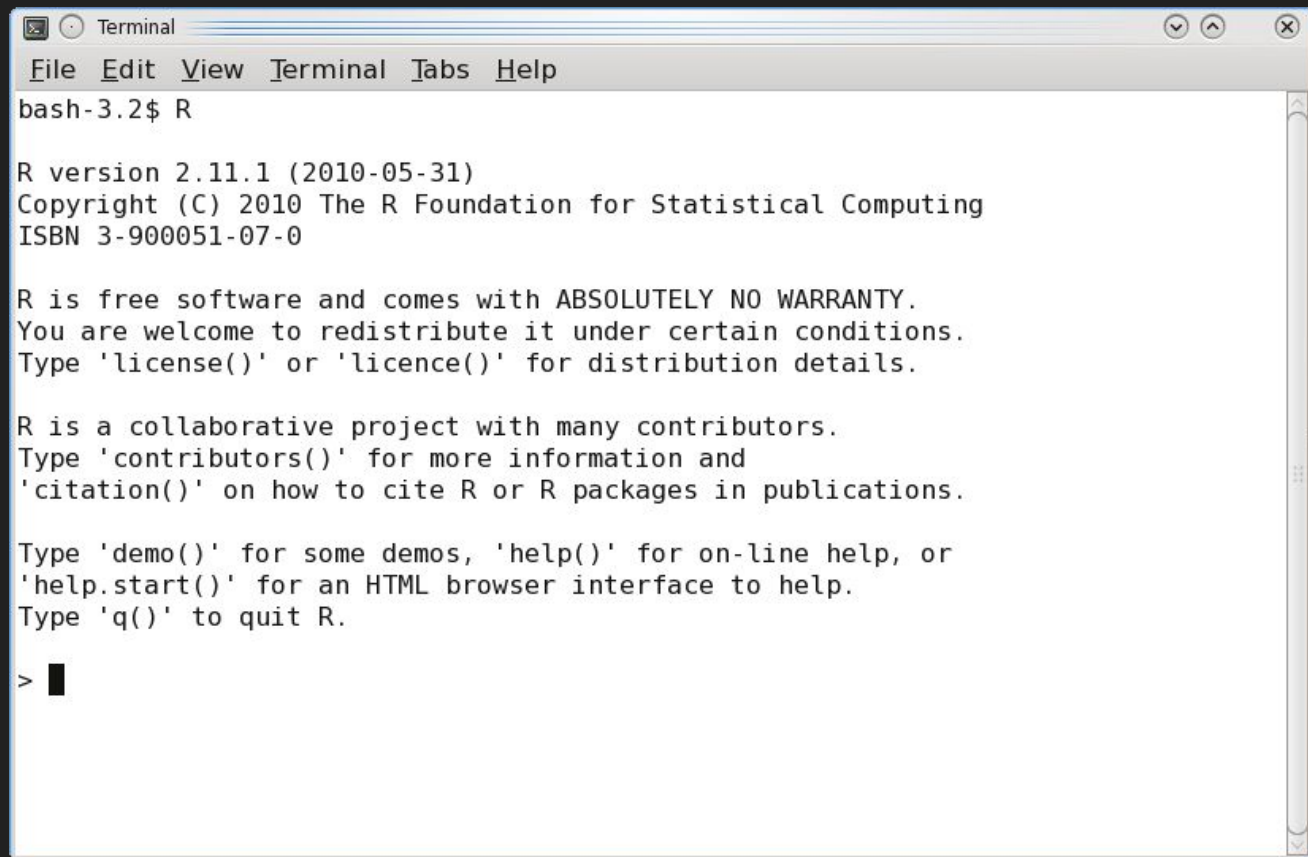
# R-GUI (CRAN)



# RStudio



# Terminal

A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the command "bash-3.2\$ R" followed by the R startup messages. The messages include the R version (2.11.1), copyright information (© 2010 The R Foundation for Statistical Computing), and a disclaimer. It also provides instructions on how to use R, including using "license()", "contributors()", "citation()", "demo()", "help()", "help.start()", and "q()". The prompt ">" is shown at the bottom.

```
bash-3.2$ R

R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █
```

# Data type

- List

```
> x <- matrix(1:20, nrow = 5)
> x
      [,1] [,2] [,3] [,4]
[1,]     1     6    11    16
[2,]     2     7    12    17
[3,]     3     8    13    18
[4,]     4     9    14    19
[5,]     5    10    15    20
> y <- sample(colours(), 3)
> y
[1] "gold4"          "darkorange1" "gray78"
> z <- rep(x=42,5)
> z
[1] 42 42 42 42 42
> 
```



# Data type

- List

```
> x <- matrix(1:20, nrow = 5)
> y <- sample(colours(), 3)
> z <- rep(x=42,5)
> l <- list(x,y,z)
> l
[[1]]
      [,1] [,2] [,3] [,4]
[1,]     1     6    11    16
[2,]     2     7    12    17
[3,]     3     8    13    18
[4,]     4     9    14    19
[5,]     5    10    15    20

[[2]]
[1] "white"           "lightseagreen" "grey28"

[[3]]
[1] 42 42 42 42 42

> █
```

```
> x <- matrix(1:20, nrow = 5)
> x
      [,1] [,2] [,3] [,4]
[1,]     1     6    11    16
[2,]     2     7    12    17
[3,]     3     8    13    18
[4,]     4     9    14    19
[5,]     5    10    15    20
> y <- sample(colours(), 3)
> y
[1] "gold4"           "darkorangel" "gray78"
> z <- rep(x=42,5)
> z
[1] 42 42 42 42 42
> █
```

# Libraries and packages

```
library(downloader)
```

# Libraries and packages

```
library(downloader)
```

```
install.packages("downloader")
```

```
source("yourScript.R")
```

# Pattern Recognition

```
source("Requirements.R")
```

```
source("funcoes/Functions.R")
```

```
# Requirements  
library(XML)  
library(RCurl)  
library(downloader)
```

```
# Functions  
  
# Função getLinkDownloadMatrix  
getLinkDownloadMatrix <- function(gse){  
  l <- list("null")  
  for (i in seq_along(gse)){  
    url <- paste0( "http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=", gse[i])  
    parseURL <- XML::htmlParse(url)  
    links <- XML::xpathSApply(parseURL, "//a/@href")  
    link <- links[grepl("matrix/", links)]  
    mat <- RCurl::getURL(link)  
    mat <- unlist(strsplit(mat, " "))  
    mat <- mat[length(mat)]  
    mat <- unlist(strsplit(mat, "\n"))  
    link <- paste0(link,mat)  
    aux <- c(link,mat)  
    l[[i]] <- aux  
  }  
  return(l)  
}  
  
# Função downloadMatrix  
downloadMatrix <- function(gseList){  
  for (i in 1:length(gseList)){  
    downloader::download(gseList[[i]][1],gseList[[i]][2])  
  }  
  return("Todos os downloads foram concluidos")  
}  
  
# Função findMatrixBegin  
findMatrixBegin <- function(mat){  
  x <- readLines(con = mat)  
  lineNum <- grep("series_matrix_table_begin",x)  
  return(lineNum)  
}  
  
# Função readMyData  
readMyData <- function(gse){  
  tally <- list("null")  
  for (i in seq_along(gse)){  
    files <- list.files(pattern = gse[i])  
    system(paste0("gunzip ",files)) # esse comando só funciona no linux  
    files <- list.files(pattern = gse[i])  
    x <- findMatrixBegin(files[i])  
    data <- read.table(file = files, header = T, skip = x, fill = T, blank.lines.skip = T)
```

```
link <- getLinkDownloadMatrix("GSE51808")
```

```
downloadMatrix(link)
```

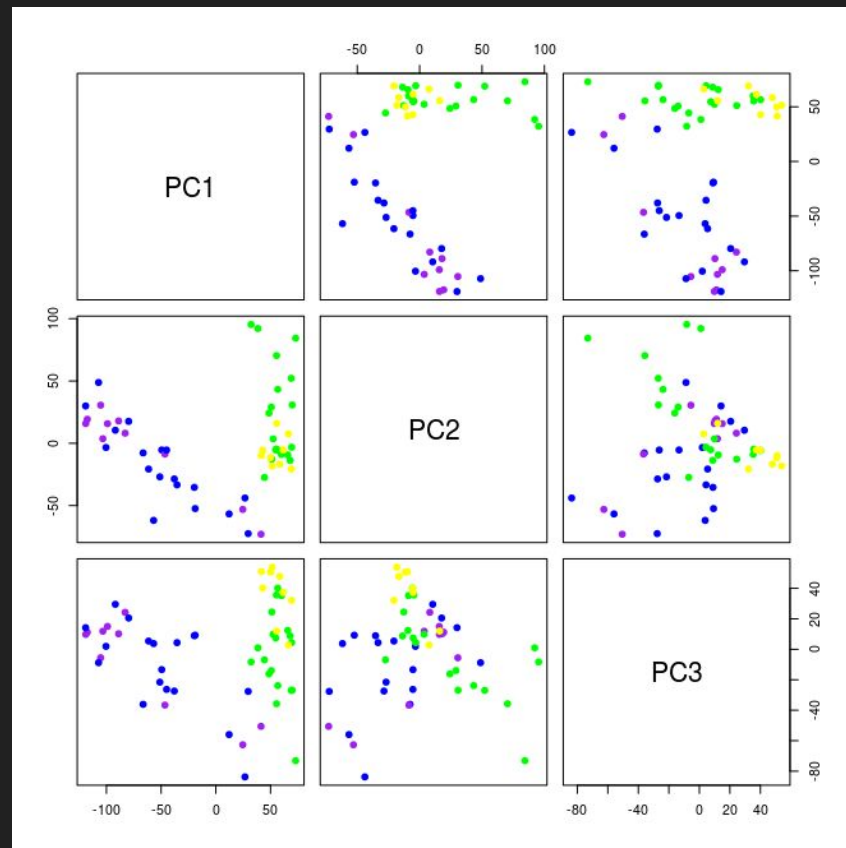
```
dados <- readMyData("GSE51808")
```

```
dados$GSE51808[1:6,1:3]
```

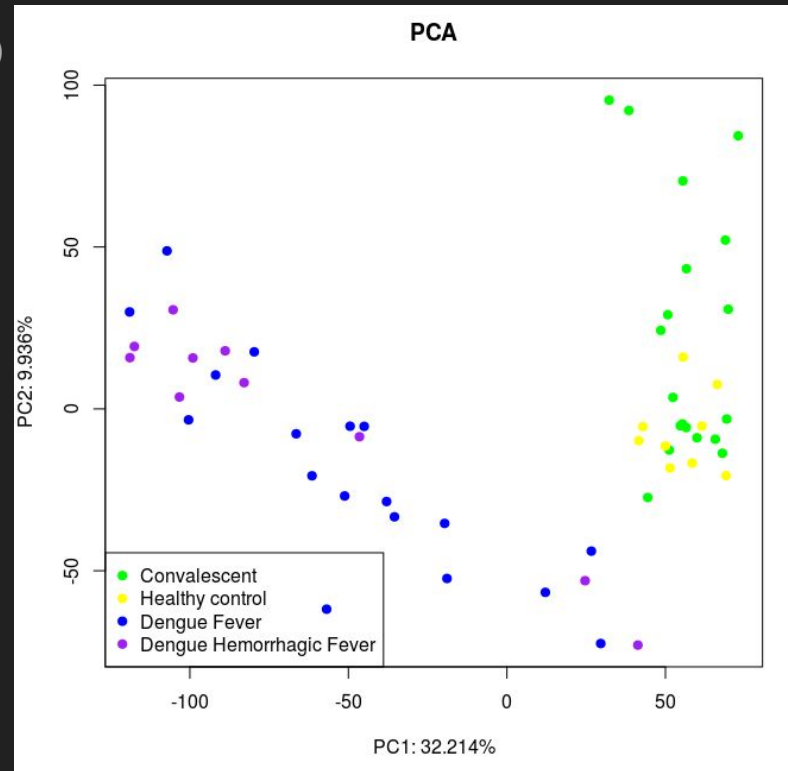
```
> dados$GSE51808[1:6,1:3]
      GSM1253028 GSM1253029 GSM1253030
1007_PM_s_at    5.97419    5.55657    6.51326
1053_PM_at      7.88225    7.75510    8.42061
117_PM_at       10.01840    8.97969    7.61888
121_PM_at        6.24138    6.55102    5.96311
1255_PM_g_at     2.92204    3.18026    2.67519
1294_PM_at       9.43186    9.23699    8.89779
> 
```

```
pca <- prcomp(as.matrix(t(dados$GSE51808)), cor=T, scale=F)
```

```
pairs(pca$x[,1:3], col=dfMeta$col, pch=19)
```



```
plot(  pca$x, col=dfMeta$col, pch=19, main = "PCA",  
      xlab=paste0("PC1: ", summary(pca)$importance[2,1]*100, "%"),  
      ylab=paste0("PC2: ", summary(pca)$importance[2,2]*100, "%"))  
legend(  "bottomleft", pch=rep(19,length(coloring)),  
        col=coloring, legend=categoria)
```





Is it over Jessica?

Yeah it is!