R Workshop

Introduction

by Connie Zabarovskaya
Center for Biomedical Informatics
Washington University in St. Louis


DON'T
PANIC
The Hitchhiker's Guide to the Galaxy

You Will Learn

- Why R?
- R and RStudio
- Working Environment
- Understanding Data Types in R
- Assignment
- Reading in data
- Slicing and Dicing Data
- Basic Functions
- Basic Plots

What is R?

- open-source programming language
- massively contributed to for over 20 years by 2 million users and thousands of developers worldwide
- over 5000 packages in statistics, data management and analysis, connecting to databases, websites and software, data visualization and more
- download from http://cran.wustl.edu/

Why R?

- automating data manipulations
- reproducible research
- sophisticated graphs and stat analyses
- distributed computing on large datasets
- publishing research results as a web app online

As opposed to other stat software

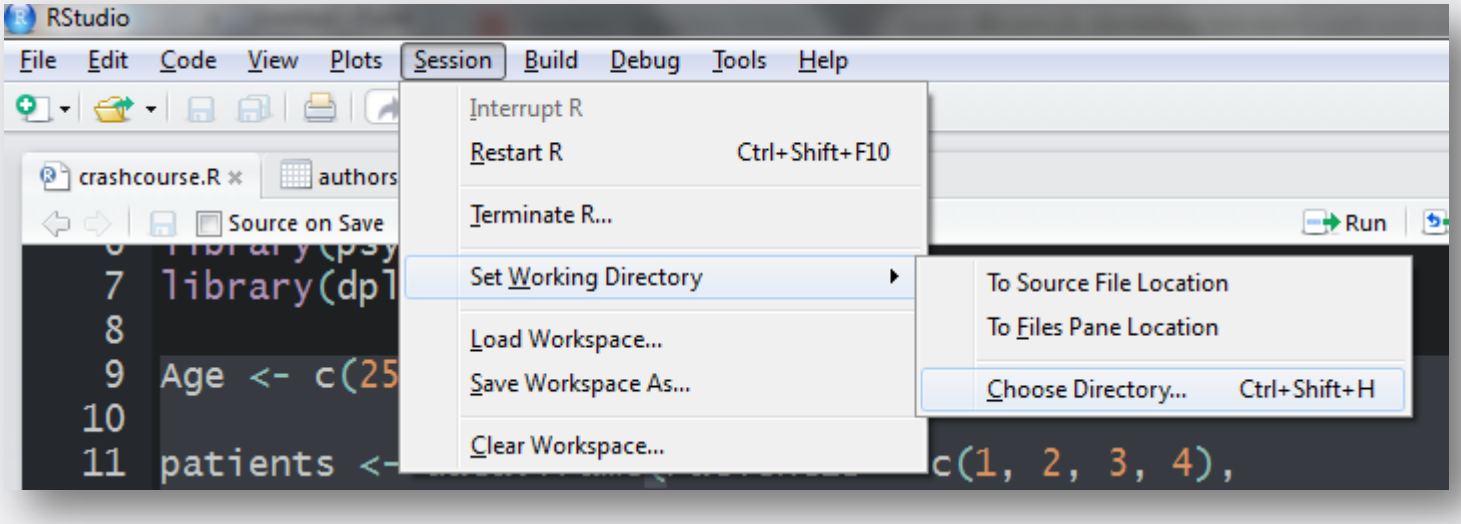
- freedom of control over functions and data manipulations
- access to all loaded data elements at all times

RStudio

- user interface for R
- to change appearance: Tools/Global Options
- download from http://www.rstudio.com/products/rstudio/download/

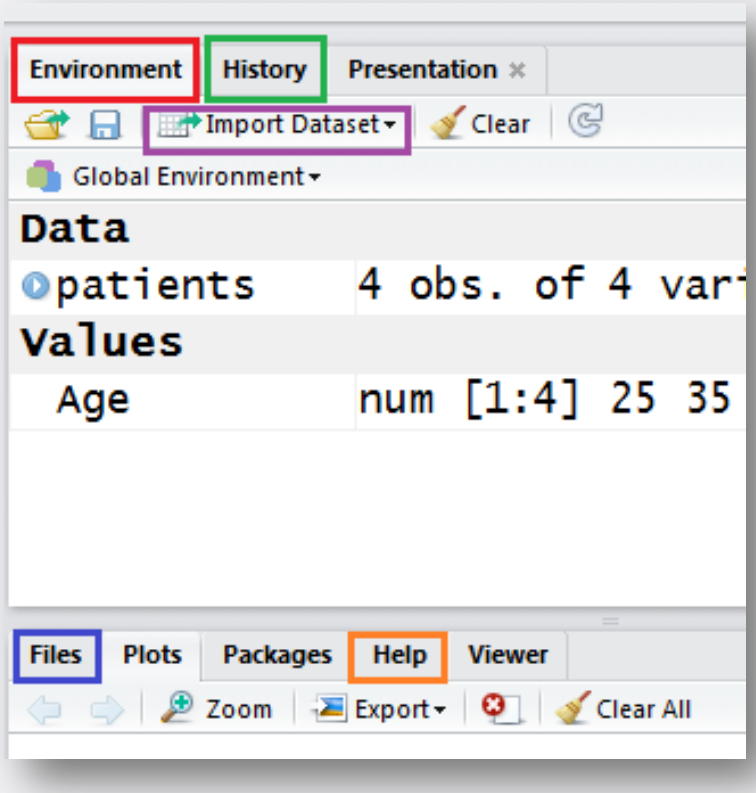
Working Environment

Working directory – your location on the “computer map”



Working Environment

- Getting help:
  - help(),
  - ?c,
  - Help tab in RStudio



Working Environment

- Comments:

  # comment here

- Installing and Loading packages:

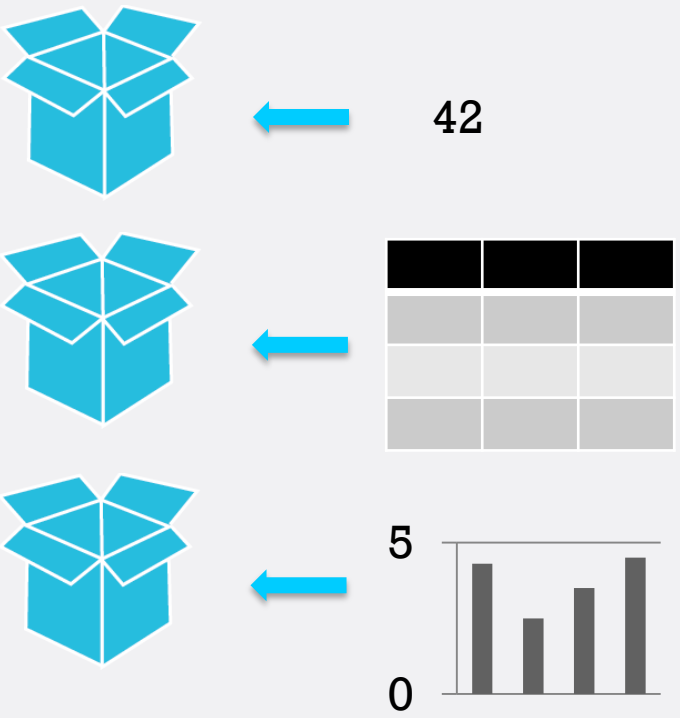
  install.packages(“ggplot2”)

  install.packages(“dplyr”)

  library(ggplot2)

  library(dplyr)

Variables and Functions



Assignment

- For commands use : <- (“greater than” & “dash”)
- Inside functions use: = (when assigning values to arguments)
- If you don’t assign – command result will be dumped in console; or can get error

Understanding Data in R

Index	1	2	3	4
Age	25	35	40	12

Vector:

```
Age <-  c(25, 35, 40, 12)
```

```
length(Age)
4
```

Data Frame:

```
patients <- data.frame(PatientID = c(1, 2, 3, 4),
                       Age = c(25, 35, 40, 12),
                       Gender = c("F", "M", "F", "M"),
                       Diabetes = c("Y", "N", "N", "N"))
```

PatientID	Age	Gender	Diabetes
1	25	F	Y
2	35	M	N
3	40	F	N
4	12	M	N

Reading in Datasets

- R can read in perhaps all possible file formats
- To check datasets in working directory: dir()


```
file <- read.csv("filename.csv")
```


```
library(foreign)
file <- read.dta("file.dta")
# no support after v. 12
```

```
library(xlsx)
file <- read.xlsx("myfile.xlsx", sheetName = "Sheet1")
# special package for Excel files needs to be installed, by
calling install.packages("xslx")
```

Practice Exercise

- Read in this dataset and assign it to var “nycflights”

```
nycflights <- read.csv(url("http://statadata.gwb.wustl.edu/nycfl13sample.csv"))
```

- View it from the RStudio Environment
- Can we tell how many variables and observations the data has?

Metadata

Functions to describe the dataset

```
# num of cols, rows, var name and type, some values
    str(patients)
# statistics on continuous variables, freq on factors
    summary(patients)
# column names
    names(patients)
# number of columns, number of rows
    ncol(patients)
    nrow(patients)
```

Practice Exercise

- Answer these questions about the dataset:
  - name two number vars in the dataset
  - name two factor vars in the dataset
  - what is the median and mean of air time (min)?
  - what is the frequency of origin categories?
- Try to write all the functions from previous slide

Understanding Data Types

- Numbers

```
c(3, 20, 10) + c(30, 10, 15)
[1] 33 30 25
```

- Strings

```
c('Ami', 'Paul', c('Angie', 'Pat'))
 [1] Ami Paul Angie Pat
```

Understanding Data Types

Factor (nominal/categorical vars with labels). Caution!

```
Gender = c("F", "M", "F", "M")
```

Range of strings: "F" "M"

Stored Integers: 1 2

```
levels(patients$Gender)
[1]    "F"     "M"
```

Understanding Data Types

- Logical: TRUE, FALSE
  - used for comparisons of data – vector to vector, single value to single value, to filter data for example, or verify a condition

```
is.na(c(3, 4, NA, 10))
[1] FALSE FALSE TRUE FALSE

c(3, 4, NA, 10) != 4
[1] FALSE TRUE NA FALSE
```

<	less than
>	great than
<=	less than or equal
>=	greater than or equal
==	equal to
!=	not equal to
|	entry wise or
||	or
!	not
&	entry wise and
&&	and

Practice Exercise

- Name some of the levels of airline variable.
- Does the dep_delay (departure delay) variable contain any NAs?*
- Is there ever distance over 4000 miles?*

Describing Data

- crosstables
  - one-way (frequencies)

```
table(patients$Gender)
```

  - two-way

```
table(patients$Gender, patients$Diabetes)
```

- distribution

```
summary(patients$Age)
describe(patients)
```

psych

```
fivenum(patients$Age) vs. quantile(patients$Age)
# fivenum() returns Tukey's five number summary
# quantile() returns quantiles corresponding to the given probabilities
```

Slicing and Dicing

```
IDs <- patients$PatientID
IDs <- patients[,1]
```

```
row2 <- subset(patients, PatientsID==2)
row2<- patients[2,]
```

PatientID	Age	Gender	Diabetes
1	25	F	Y
2	35	M	N
3	40	F	N
4	12	M	N

Conditional (logical) subsetting:

```
patients[patients$Age > 12,]   !
subset(patients, Age > 12)

patients[patients$Age > 12 & patients$Gender =="M",] !

head(patients)
tail(patients,2)
```

```
singlecell <- patients[4,3]
singlecell <- patients$Gender[4]
```

Practice Exercise

- Practice two ways to select carrier variable (7th col).
- Practice two ways to select flights with carrier UA . Practice how to select only the 1st row.
- Practice two ways to select 20th row, 7th column (carrier var). What's the value?
- How would you select all the flights with positive departure delay (dep_delay var)? (use subset())
- Create a crosstable of airline and origin. What airline only flies from Newark airport?

Creating New Variables

To create a new variable in an existing dataset, pretend as if it's already there.

PatientID	Age	Gender	Diabetes	BloodSugar
1	25	F	Y	140
2	35	M	N	135
3	40	F	N	126
4	12	M	N	112

```
patients$BloodSugar <- c(140, 135, 126, 112)
```

Practice Exercise: create air_hours variable, based on air_time variable, dividing it by 60

Basic Functions

- Math functions:
  - **sqrt(patients$Age)**
  - **log(patients$Age)**
  - **mean(patients$Age)**
  - **max(patients$Age)**
  - **sum(patients$Age)**

Data Wrangling

- unique values:
  - **length(unique(patients$PatientID))**
  - **n_distinct(patients$PatientID)** dplyr
- remove duplicate rows
  - **distinct(patients)** dplyr
- filtering columns
  - **select(patients, contains("a"))** dplyr
- filtering rows
  - **filter(patients, Age > 12)** dplyr

Data Wrangling

- aggregating (collapsing)

```
aggregate(patients$Age, by =
   list(Gender=patients$Gender), FUN=mean)
```

```
patients %>% group_by(Gender) %>%
   summarise(avg=mean(Age))
```

dplyr

- merging

```
newdf <- merge(data1, data2, by="ID")
```

Basic Plots

Base R

- barplot

```
plot(patients$Gender) # has to be factor var
plot(patients$Gender, main = “Gender Freq”, xlab =
     “Gender”, ylab = “Frequency”)
```

- histogram

```
hist(patients$Age)
```

- scatterplot

```
plot(patients$Age, patients$BloodSugar)
```

Basic Plots

Using ggplot2 (functions: ggplot() and qplot())

```
qplot(x = Age, y = BloodSugar, color = Gender, data =
      patients, geom = "point")
```

Exporting Data

- Easiest way in a csv

```
write.csv(patients, “patients.csv", row.names
   = F)
```

Practice Exercise

- Plot a scatterplot using qplot() function, to look at relationship between dep_delay and visib (visibility). Use origin for color groups.

Putting It All Together

- Delay Prediction App: https://conniez.shinyapps.io/delayPredictor/

Limitations of R

Limitation	How to Overcome
Memory limits	Packages for parallelization and memory management. Check out HP Distributed R, Biglm, ff, etc.
Lack of official support	Google, StackOverflow. If package is new – submit issue to GitHub repo

Where to learn more

Interactive Learning

http://tryr.codeschool.com/

https://www.datacamp.com/

https://www.teamleada.com/tutorials/introduction-to-statistical-programming-in-r

MOOC Course

https://www.coursera.org/course/rprog

R Playground online (select R Programming)

http://www.tutorialspoint.com

R Cheat Sheets

http://www.rstudio.com/resources/cheatsheets/

More Resources

- HP Distributed R http://www.vertica.com/hp-vertica-products/hp-vertica-distributed-r/
- Handling Big Data: packages - RODBC, Biglm, ff, bigmemory, snow, etc.

Thank You! Questions?