Oxford University Centre for Educational Assessment

UNIVERSITY OF OXFORD

Introduction to R for data analysis

Professional Development and Training Course:
Analyzing International Large-Scale Assessment Data with R

Dr. Daniel Caro & Dr. Christian Bokhove

AERA 2014
Philadelphia, April 2, 2014

UNIVERSITY OF Southampton

Outline

- The R language and environment
- R basics for data analysts
- Practical examples with international assessment data

What is R?

- Programming language:
  - Object-oriented language with broad functionalities
- Environment for statistical analysis:
  - Built in functions and contributed packages
- Open-source project:
  - Access source code, modify it, share it, further develop it
- Community:
  - Many programmers around the world (e.g., Munich, Oxford, Vienna)

Differences between R and SPSS/Stata

R

- Object oriented language
  - An object can be a dataset but also a function, a plot, a character, a list
- Multiple datasets at the same time
- Different kind of data objects
  - Data frames, matrix, lists
- Functions take objects as arguments and produce results
- Outputs can be used as inputs

SPSS/Stata

- Data oriented software
- One single data at a time
- Each row is a case
- Each column is a variable
- Variable labels, value labels, user missing values

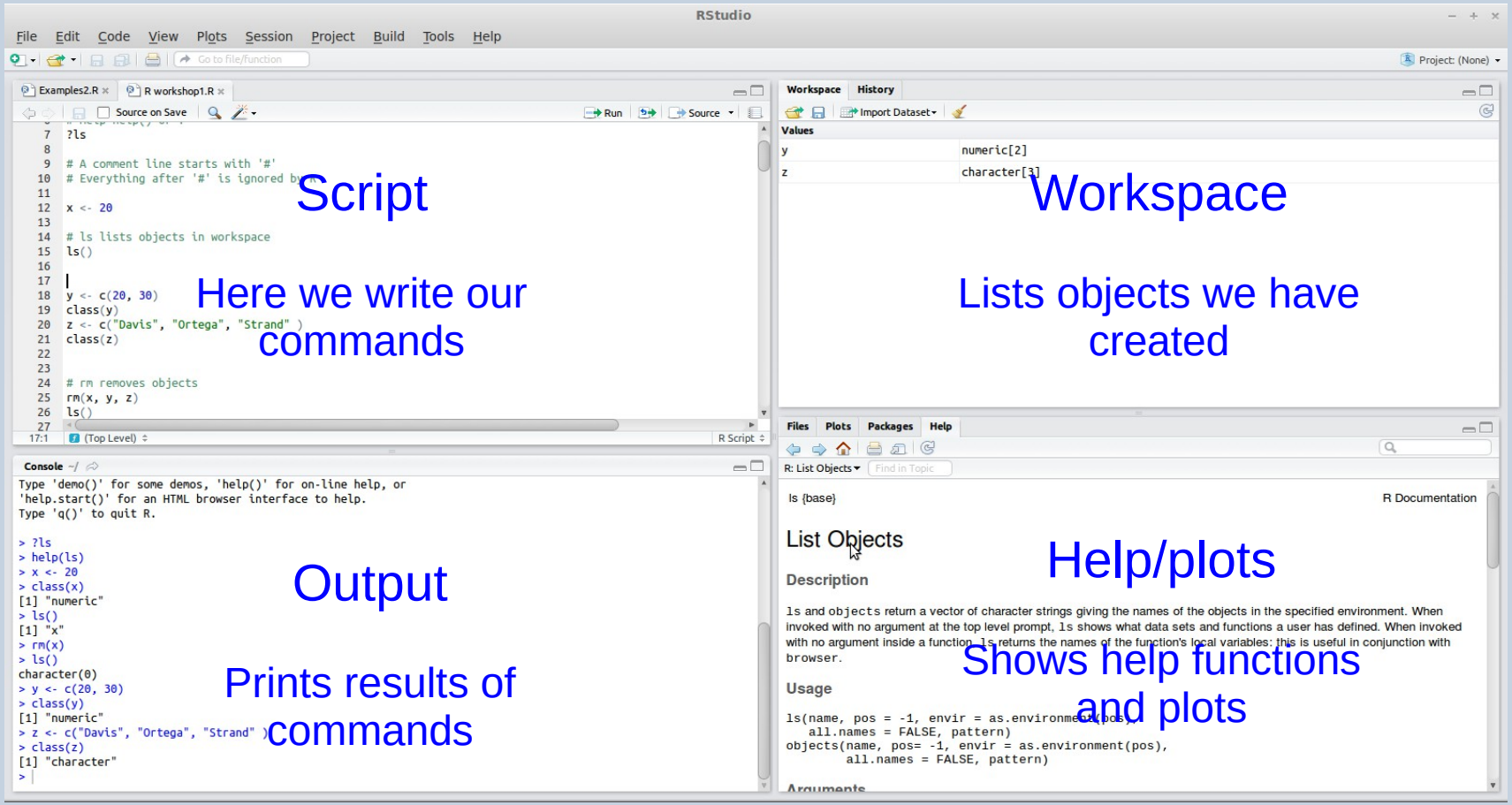
R installation

- R runs on different platforms, including Linux, Mac, and Windows
- Download R from www.r-project.org
- Click on 'CRAN mirror'
- Select mirror closest to your location
- Choose your operating system (e.g., Linux, Mac, Windows)
- For Windows select 'base' and download the executable
- For Mac select the latest version

RStudio

- R is a cross-platform language, but its graphical user interface (GUI) is not
  - Windows and Mac provide different GUIs after installation
  - No GUI in Linux after installation
- RStudio provides a cross-platform GUI for Windows, Mac, and Linux
- Download RStudio from www.rstudio.com

RStudio interface



Outline

- The R language and environment
- R basics for data analysts
- Practical examples with international assessment data

Basic commands: Comments and objects

```
# A comment line is started by '#'
# Everything after '#' is ignored by R
#  R is case sensitive

# Create object ('<-' to assign value to object)
> x <- 20

# Print object
> x
[1] 20

# ls() lists objects in workspace
> ls()
[1] "x"
```

Basic commands: Getting help

Get help for function

> help(ls)

> ?ls

Search for help using keyword

> help.search('regression')

Also in “An Introduction to R”, stackoverflow.com, google, and the R mailing list. The mailing list is organized by SIGs. R bloggers is an excellent blog.

Basic commands: Creating objects

```
# Create numeric object
> x <- c(20, 30)
# Create character object
> y <- c("Ting", "Kane", "Malkeet" )
# Print object class
> class(y); class(x)
[1] "character"
[1] "numeric"
```

Basic commands: Save workspace in directory

```
# Save objects 'x' and 'y' in file 'two_objects.RData'
> save(x, y, file='two_objects.RData')
# Save current workspace (saves all objects)
> save.image(file='two_objects.RData')

# Get working directory
> getwd()
# Change working directory
> setwd(dir= # enter filepath #)
```

Basic commands: Filepaths and OS

filepath is the directory where your data is located

The filepath structure varies by OS, for example:

Windows:

filepath= “C:/My Documents/AERA 2014/R workshop”

 # (note that in R the backslash (\) needs to be replaced for the common slash (/))

With choose.dir() you can also choose a folder by browsing

Mac: filepath =“/Users/eldani/AERA 2014/R workshop”

Linux: filepath ="/home/eldani/Dropbox/Work/Events/AERA 2014/R workshop"

Basic commands: Filepaths and OS

```
# Set path

> setwd(dir=filepath)

# Save two objects in working directory (filepath)

> save(x, y, file='two_objects.RData')

# Equivalent to

> save(x, y, file="/home/eldani/Dropbox/Work/Events/AERA 2014/R workshop/two_objects.RData")

# Or save workspace

> save.image(file="/home/eldani/Dropbox/Work/Events/AERA 2014/R workshop/all_objects.RData")
```

Basic commands: Remove objects

```
# Remove 'x' and 'y'
> rm(x, y)
# Remove all objects
> rm(list=ls())
# Load workspace
> load(file='two_objects.RData')
# Create sequence
> x <- -10:10
```

Basic commands: Descriptive statistics

```
# Calculate mean
> mean(x)
# Calculate standard deviation
> sd(x)
# Calculate min and max
> range(x)
# Produce summary statistics
> summary(x)
# Calculate length of object
> length(x)
```

Basic commands: Dealing with NAs

```
# Add NA element to x

x <- c(x, NA)

[1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7
8   9  10  NA

# Descriptive statistics

> mean(x); sd(x); range(x)

[1] NA

[1] NA

[1] NA NA
```

Basic commands: Dealing with NAs

```
# Descriptive stats removing NAs
> mean(x, na.rm=TRUE)
[1] 0
> sd(x, na.rm=TRUE)
[1] 6.204837
> range(x,  na.rm=TRUE)
[1] -10  10
```

Basic commands: Logical operators and NAs

```
# NA elements
> is.na(x)
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE  TRUE
# Number of NA elements
> sum(is.na(x))
# Percentage of missings
> sum(is.na(x))/length(x)
```

UNIVERSITY OF OXFORD UNIVERSITY OF Southampton

Basic commands: Logical comparisons

```
# Comparisons
> x > 7
> x ==0
> x > -4 & x < 6
# Indices that are TRUE
> which(x > -4 & x < 6)
# Print values that satisfy condition
> x[which(x > -4 & x < 6)]
```

Basic commands: Frequency table

```
# Replicate elements
> x <- rep(0:1, 20)
> x
 [1] 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
# Frequency table
> table(x)
x
 0  1
20 20
```

Basic commands: Draw random sample

```
# Replicate c(0:1) 10,000 times
> x <- rep(0:1, 10000)
# Draw random sample
> sample(x, 20)
[1] 1 1 0 1 0 1 0 0 0 0 1 0 1 1 1 1 0 0 1 1
# Draw sample with replacement
> x <- sample(c(0,1), 20, replace =TRUE)
> x
[1] 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1 0 0 1 0 0
```

Basic commands: Random numbers

```
# Generate random numbers from normal distribution

> norm(10)

 [1]  1.44738709 -0.51542685 -0.01471405 -0.84880873
-1.59386637 -1.18660813  0.64949054 -1.66442462 0.50015844
0.03978030

# Generate random numbers with mean and sd arguments

> y <- rnorm(n=20, mean=500, sd=100)

> y

[1] 494.7246 362.1198 463.6260 569.8073 520.4810 579.4306
513.5433 325.6292 536.8283 557.0928 590.3315 523.1634
514.4952 433.0777 593.3379 622.9004 452.8358 584.6108
808.1443 646.2154
```

Basic commands: Create data frame

```
# Create data frame with 'x' and 'y'
> df <- data.frame(achievement = y, sex = x)
# Print data frame
> df
# Print object class
> class(df)
[1] "data.frame"
# Summary statistics
> summary(df)
```

Basic commands: Dimension of data frame

```
# Data frame dimension
> dim(df)
[1] 20  2
# Number of rows
nrow(df)
[1] 20
# Number of columns
ncol(df)
[1] 2
```

Basic commands: View data

```
# Print first section of dataset

> head(df)

  achievement sex
1    494.7246   1
2    362.1198   0
3    463.6260   0
4    569.8073   1
5    520.4810   0
6    579.4306   0

# Invoke data viewer
> View(df)
```

Basic commands: Data frame attributes

```
# Displays object structure and attributes
> str(df)
> attributes(df)
$names
[1] "achievement" "sex"
$row.names
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
$class
[1] "data.frame"
```

Basic commands: Names are not objects

```
# Print variable names
> names(df)
[1] "achievement" "sex"
# Mean achievement (wrong approach)
> mean(achievement)
Error in mean(achievement) : object 'achievement' not found
# Correct approach
> mean(df$achievement)
> mean(df$achievement) == mean(df[["achievement"]])
```

Basic commands: Statistic by group variable

```
# Print frequency table
> table(df$sex)
# Calculate mean achievement by sex
# 'tapply' applies function by group variable
> with(df, tapply(achievement, sex, mean))
0                1
513.1534    487.2118
```

Basic commands: Statistic by group variable

```
# Calculate mean achievement by sex

# 'by' applies function to data frame by factor

> with(df, by(achievement, sex, mean))

sex: 0
[1] 513.1534
------------------------------------------------------------------------------
sex: 1
[1] 487.2118
```

Basic commands: Using split and lists

```
# Calculate mean achievement by sex
# Using 'split' and 'lapply'
# 'split' data by sex and returns list
> sexl <- split(df, df$sex)


# Print object class and length
> class(sexl)
[1] "list"
> length(sexl)
[1] 2
```

Basic commands: Using split and lists

```
# Calculate mean achievement by sex

# List first element

> sexl[[1]]

> class(sexl[[1]])


# lapply applies function over a list

> lapply(sexl, function(x) mean(x$achievement))
```

Outline

- The R language and environment
- R basics for data analysts
- Practical examples with international assessment data

Reading SPSS data with 'foreign'

Several functions like ?read.table, ?read.csv

For our SPSS data example, we need package 'foreign'

Package installation

> install.packages("foreign")

library loads package

> library(foreign)

read.spss data and converts to data.frame (change backslash for slash)

> pisa.school <- read.spss(file="*filepath*/INT_SCQ12_DEC03.sav", to.data.frame=TRUE)

Examining PISA 2012 school data

```
> class(pisa.school)
> dim(pisa.school)
> head(pisa.school)
> View(pisa.school)
> summary(pisa.school)
> str(pisa.school)
# Print variable names
> names(pisa.school)
# Print variable labels
> attr(pisa.school, "variable.labels")
```

Calculating means with NAs

```
# Quality of school educational resources (SCMATEDU)
# Calculate mean and SD
> mean(pisa.school$SCMATEDU); sd(pisa.school$SCMATEDU)
[1] NA
[1] NA

# Variable SCMATEDU has NAs
> summary(pisa.school$SCMATEDU)
```

Calculating means with NAs

```
# How many?
> sum(is.na(pisa.school$SCMATEDU))
[1] 514
> sum(is.na(pisa.school$SCMATEDU))/length(pisa.school$SCMATEDU)
[1] 0.02833673
# Mean excluding NAs
> mean(pisa.school$SCMATEDU, na.rm=TRUE)
[1] -0.1406119
> sd(pisa.school$SCMATEDU, na.rm=TRUE)
[1] 1.12054
```

Producing results by country

```
# Number of schools by countries

> table(pisa.school$CNT)

# Mean SCMATEDU by country with tapply

> tapply(pisa.school$SCMATEDU, pisa.school$CNT, mean)

# Mean excluding NAs

> tapply(pisa.school$SCMATEDU, pisa.school$CNT, mean, na.rm=TRUE)

# Using 'with'

> with(pisa.school, tapply(SCMATEDU, CNT, mean, na.rm=TRUE))

# Create object with country means

> SCMATEDU.m <- with(pisa.school, tapply(SCMATEDU, CNT, mean,
na.rm=TRUE))
```

Exporting results to spreadsheet

```
# Format results for presentation
> data.frame(SCMATEDU.m)
# Round numbers to 2 decimals
> round(data.frame(SCMATEDU.m), 2)
> SCMATEDU.tab <- round(data.frame(SCMATEDU.m), 2)
# Export table to spreadsheet (.csv)
> write.csv(SCMATEDU.tab, "pisa.table.csv")
> getwd()
```

Thank you!

daniel.caro@education.ox.ac.uk

C.Bokhove@soton.ac.uk