Anthony J. Damico

Analyzing European social survey data with R

Kaunas University of Technology

2014 09 15

European Social Survey

Why use R?

- R is a language and environment for statistical computing and graphics
  - More than just another statistical analysis software package (SAS, SPSS, Stata)
  - Less than programming language (C++, Perl, Python)
  - Combination of both → 'one-of-a-kind'!
- Disadvantages:
  - Hard to learn → programming, console, scripts
  - Obscure terms, intimidating manuals, odd symbols, inelegant output
  - Too demanding for simple tasks (Excel, SPSS etc.)
- Advantages:
  - Open-source (FREE!) → learn once – use forever
  - Best for reproducible research (coming standard!)
  - Cross platform (WIN/MAC/LIN)
  - Almost every analysis (including most advanced and inovative) is possible in R (3000+ packages)
  - Beautiful visualizations :-)

European Social Survey

Why use R?

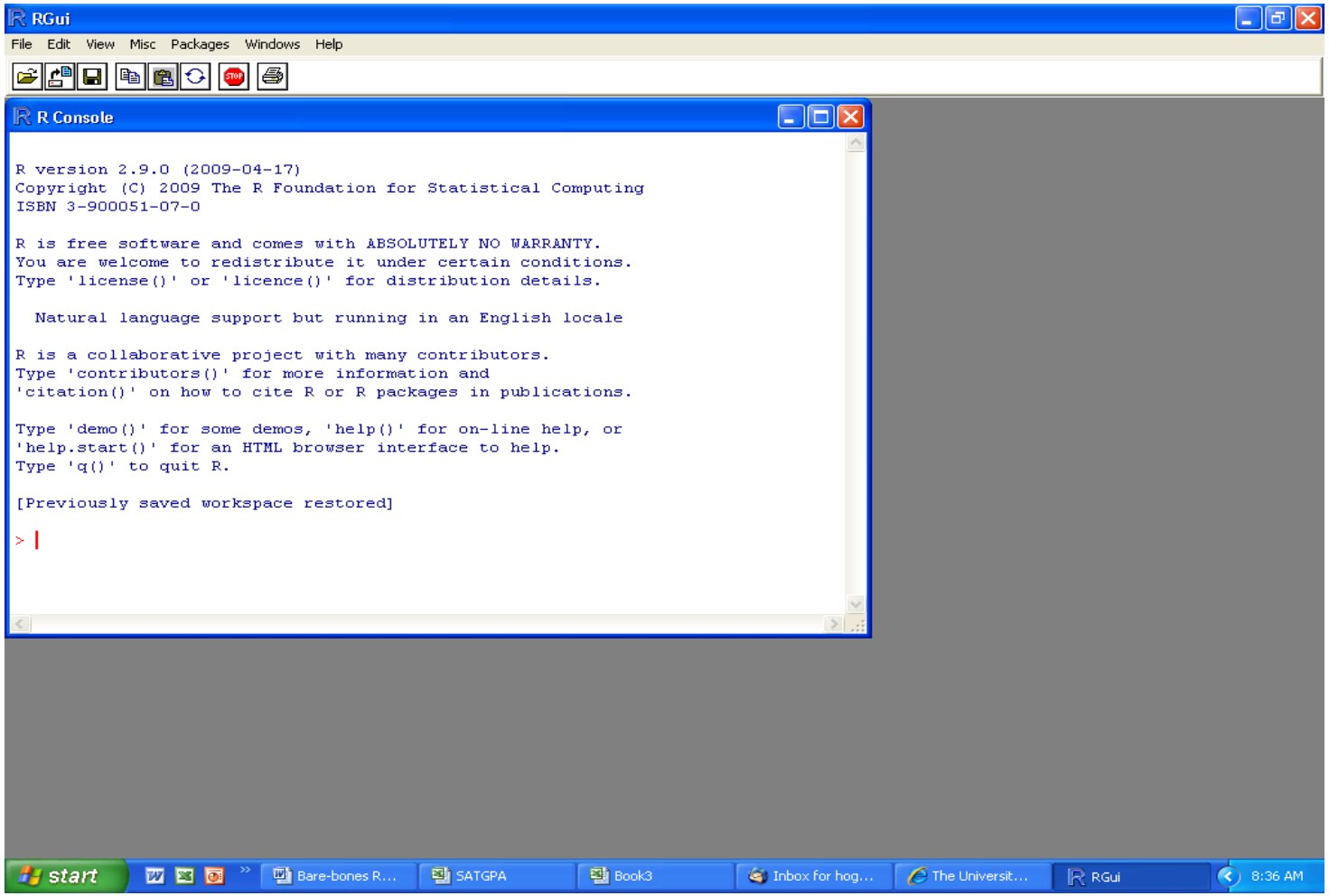
European Social Survey

Installing and running

- Installing R
  - Windows/Mac
    - Go to cran.r-project.org
    - R for Windows/Mac screen, click “base”
  - Linux
    - Install with your Linux installer (platform dependent)
- Running R
  - Find icon in program menus and run :-)

Installing and running

- Get some ugly looks



European Social Survey

Installing and running

- Awful prompt:
  - > This is the so-called 'R prompt'
  - If cursor after it is blinking this tells that R is ready to take a command and execute it
  - Only console (no drop down menus, no 'point-and-click')
  - Type and run commands
  - Not very convenient...

European Social Survey

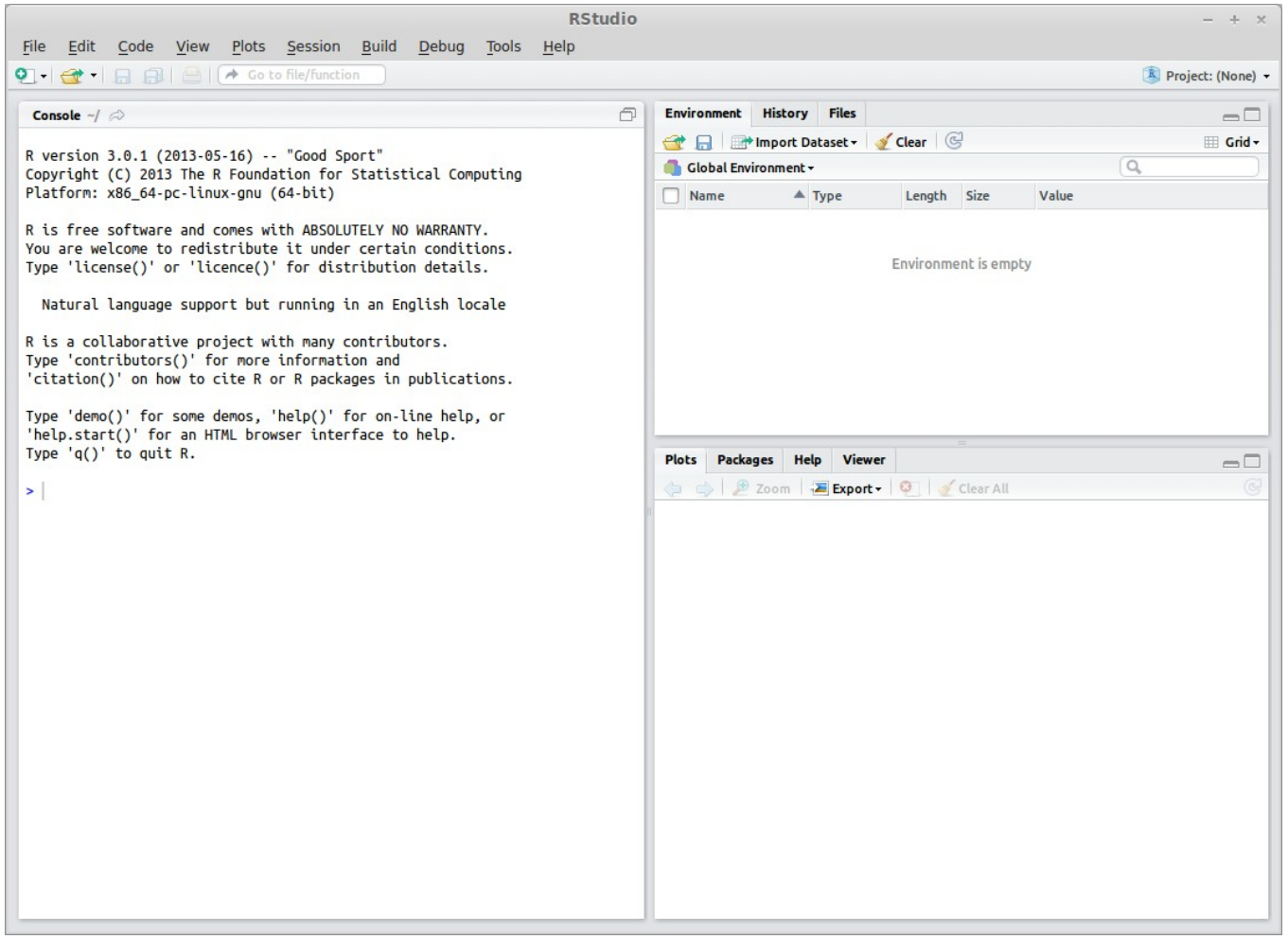
Installing and running

- RStudio is a graphical user interface (GUI) for R base (download and install from: www.rstudio.com)
  - Many GUIs: www.sciviews.org/_rgui
    - RStudio – 'best of the bad'
      - For most important things it has 4 separate windows
        - Scripts
        - Console
        - Files, objects etc.
        - Plots, packages, help

European Social Survey

Installing and running

- More pleasant look



European Social Survey

Installing and running

- Main windows:
  - Scripts
    - Write commands here and execute in Console
    - Save and reuse
    - Reproducible research
    - Write comments
  - Console
    - Commands executed
    - Text/numeric results
  - Objects (→ data and results) and files
  - Packages, help and pictures

European Social Survey

Installing and running

- Play with RStudio from www.rstudio.com/ide/docs
  - Using RStudio
    - Working in the Console
    - Editing and Executing Code
    - Code Folding and Sections
    - Navigating Code
    - Using Projects
    - Command History
    - Working Directories and Workspaces
    - Customizing RStudio
    - Keyboard Shortcuts
  - Advanced Topics
    - Character Encoding

Working with R: basics

- Using R as calculator
  - Enter these after the prompt (copy and paste), observe output
    - `2+3`
    - `2^3+(5)`
    - `6/2+(8+5)`
    - `2^3+(5)`
    - `2 ^ 3 + ( 5 )`
  - Use # at end of command (on separate line) for comments/notes
    - `(22+34+18+29+36)/5 # Calculating mean`
  - R as calculator: not very useful

European Social Survey

Working with R: basics

- R is about executing data operations (functions), getting results, saving them and reusing
- Function → function name + parentheses
  - library(survey)
- Any kind of result → object (data, variable, analysis result)
- You save objects with '<-'
- Creating a Data Object ('free floating objects' → most awesome thing in R, not available in other statistical packages like SAS, SPSS, Stata etc.)
  - `Scores <- c(22, 34, 18, 29, 36)`
    - c is short for 'concatenate'...
    - in plain English 'treat as data set'
  - Then do → `Scores`
    - R will print the data set

Working with R: basics

- Object naming conventions
  - Object names are case sensitive
  - No blank spaces in names
    - (can use _ or . to join words, but not -)
  - Always start with a letter (cap or lc)
- Create SCORES
  - `SCORES<-c(122, 134, 118, 129, 124)`
  - SCORES different from Scores
- Check results typing and executing
  - `SCORES`
  - `Scores`

Working with R: basics

- Non-numeric data
  - Enclose in quotes: single or double
  - Always separate entries with comma
  - Example:
    - `Names <- c("Mary", "Tom", "Ed", "Dan", "Meg")`
    - `Names <- c('Mary', 'Tom', 'Ed', 'Dan', 'Meg')`

European Social Survey

Working with R: basics

- R functions
  - Thousands of them
  - R's biggest strength, most common use
  - Function help
    - help(function) → help(library)
    - example(function) → example(library)
    - ?function → ?library
    - ??keyword → ??mean
  - Reminder: function names case sensitive

European Social Survey

Working with R: basics

- R functions
  - Simple examples
    - Functions for mean, standard deviation, summary
      - NB: function names case sensitive!
      - `mean(Scores)`
      - `sd(Scores)`
      - `summary(Scores)`
    - Function for correlation
      - `cor(Scores,SCORES)`
    - TAB key → invokes possible endings for data objects and functions

European Social Survey

Analyzing ESS with R

- Not possible to have census data frequently → researchers use survey data
  - Surveys use samples of respondents drawn from the population to infer something about the population (eg. trust in police)
    - Simple random samples too expensive → complex survey sample designs
  - ESS uses complex probability sample designs which are different in all the countries covered
    - In order to analyze data collected using complex sample designs researchers need to include sample design information (stratification, clustering, selection probabilities) in their analyses
    - www.asdfree.com - is a website dedicated to the analysis of different popular complex sample design surveys with R

Analyzing ESS with R

- www.asdfree.com – is a website dedicated to the analysis of different popular complex sample design surveys with R:
  - ESS is one of those surveys
  - Others include WVS, PISA, ANES
- How to analyze ESS data with R → 2 steps (1):
  - Download data → script 1:
    - Register for an account and plop 'your.email' at the top of this script and let 'er rip
    - Automatically log in and determine which countries and rounds are currently available
    - For each round available, cycle through each file available, download, unzip, and import it.
    - Save everything on the local disk as a convenient data.frame object

European Social Survey

Analyzing ESS with R

- www.asdfree.com – is a website dedicated to the analysis of different popular complex sample design surveys with R:
  - ESS is one of those surveys
  - Others include WVS, PISA, ANES
- How to analyze ESS data with R → 2 steps (2):
  - Analyze data → script 2:
    - Load a country-specific data set, merge on the survey design data file, remove unnecessary columns (optional)
    - Construct a survey design object producing Taylor series linearized standard errors
    - Use that survey design object to run examples of any summary statistical analysis you'll need (with correct estimates and theirs standard errors)

Analyzing ESS with R

- Example with Lithuanian data from Round 5 (1):
  - Run a script to download data (download all microdata.R)
    - Register for an account
    - Input 'your.email' at the top of this script
    - Change working directory at the top of this script
    - Install any required libraries
    - Have a coffee (it takes some time to download all data to your computer and prepare it for analysis)

Analyzing ESS with R

- Example with Lithuanian data from Round 5 (2):
  - Analyze data with the provided script (adapted to Lithuanian data) analysis examples LT.R (1)
    - Input directory where ESS data was downloaded (line 57)
    - Load necessary libraries:
      - `library(survey) # load survey package (analyzes complex design surveys)`
      - `library(downloader) # downloads and then runs the source() function on scripts from github`
    - Since Lithuanian ESS round 5 data has some PSU with single observations line 69 is uncommented:
      - `options( survey.lonely.psu = "adjust" )`

Analyzing ESS with R

- Example with Lithuanian data from Round 5 (2):
  - Analyze data with the provided script (adapted to Lithuanian data) analysis examples LT.R (2)
    - Load ESS LT R5 main and supplementary questionnaire data (line 120):
      - `load( "./2010/LT/ESS5.rda" )`
    - Load ESS LT R5 sample design data (line 129):
      - `load( "./2010/LT/ESS5__SDDF.rda" )`
    - Merge these files into one:
      - `ess5.lt <- merge( ess5.lt.ms , ess5.lt.sddf , by=c("cntry", "idno") , all = TRUE )`

Analyzing ESS with R

- Example with Lithuanian data from Round 5 (2):
  - Analyze data with the provided script (adapted to Lithuanian data) analysis examples LT.R (3)
    - Optional → keep only those variables that are needed in the analysis
      - Lines 164-199
      - Selected variables: TV watching (tvtot) + Children living at home (chldhm) + gender of respondents (gender) + Complex sample survey design variables (psu, stratify, prob)
    - Create survey design for Taylor-series linearization
      - `ess5.lt.design <- svydesign( ids = ~psu , strata = ~stratify , probs = ~prob , data = x )`
      - Notice the 'ess5.lt.design' object used in all subsequent analysis commands

Analyzing ESS with R

- Example with Lithuanian data from Round 5 (2):
  - Analyze data with the provided script (adapted to Lithuanian data) analysis examples LT.R (4)
    - Run summary statistics and results export examples:
      - Lines 226-429
      - Run them line by line (CRTL + Enter) if you want to see what is going on
      - Change to the directory of results at line 413
    - Draw a barplot
      - `barplot( 100*female.rate.by.chldhm[ , 2 ] , main = "% of 15+ Year old Lithuanians who are Female, by Child Living at Home" , sub="Design wighted results", names.arg = c( "Child at Home" , "No Child in Household" ) , ylim = c( 0 , 100 ), col = "darkgreen")`

Analyzing ESS with R

- Example with Lithuanian data from Round 5 (2):
  - Analyze data with the provided script (adapted to Lithuanian data) analysis examples LT.R (5)
    - Plug-in our own variables and rerun analyses
      - You only need to change variable names where appropriate
      - Yes, it is that simple! :-)

Good luck!