Descriptive analysis

Field Coordinator Training - R Track

Luiza Andrade, Leonardo Viotti & Rob Marty

June 2018

1. Introduction

2. Getting started

3. Using packages

4. Quick summary statistics

5. Descriptives tables

6. Export tables to LaTeX

7. Descriptives tables - Create tables from scratch

8. Export tables to Excel

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Introduction

Descriptive statistics are used to represent the basic features of data. When we talk about descriptive analysis, it usually means that we're not making any assumptions, and we're not using probability theory to infer anything beyond the immediate data.

This session is mostly focused on how to implement descriptive analysis in R. We will not go in depth into these concepts, but you can find some useful references at the end of this presentation.

Introduction

This session will cover two topics:

1. Quick ways to extract summary information from your data
2. How to use this information to create and export tables

Outline

1. Introduction
2. **Getting started**
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Getting started

First, let's load the data that is going to be used in the training.

Load the data

```
#### Load CSV data

# Replace with where your data is
FolderPath <- file.path("YOUR FOLDER PATH HERE")

whr <- read.csv(file.path(FolderPath,"whr_panel.csv"),
                header = T)
```

Getting started

Before starting, let's install the packages we'll use in this session since it might take a while.

We'll get into what packages are in a momment.

Install packages

```
  install.packages(c("stargazer",
                     "tidyverse"),
                   dependencies = T) # dependencies argument
#installs all other packages that it may depend upon to run
```

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Using packages

- Since there is a lot of people developing for R, it can have many different functionalities
- To make it simpler, these functionalities are bundled into packages
- A package is the fundamental unit of shareable code

Using packages

- It may contain new functions, but also more complex functionalities, such as a Graphic User Interface (GUI) or settings for parallel processing (similar to Stata MP)
- They can be shared through R's official repository - CRAN (10,000+ packages reviewed and tested) and many other online sources
- There are many other online sources such as Github, but it's important to be careful, as these probably haven't gone through a review process as rigorous as those in CRAN

Using packages

- To install and use packages you can either do it with the user interface or by the command prompt.

```r
# Installing a package
install.packages("stargazer",
                 dependencies = T)
# the dependencies argument also installs all other packages
# that it may depend upon to run
```

Using packages

- You only have to install a package once, but you have to load it every new session. To load a package type:

```r
library(stargazer)
```

Using packages

Once a package is loaded, you can use its features and functions. Here's a list of some useful and cool packages:

- Rcmdr - Easy to use GUI
- swirl - An interactive learning environment for R and statistics.
- ggplot2 - beautiful and versatile graphics (the syntax is a pain, though)
- stargazer - awesome latex regression and summary statistics tables
- foreign - reads dtas and other formats from inferior statistical software
- zoo - time series and panel data manipulation useful functions
- data.table - some functions to deal with huge data sets
- sp and rgeos - spatial analysis
- multiwayvcov and sandwich - clustered and robust standard errors
- RODBC, RMySQL, RPostgresSQL, RSQLite - For relational databases and using SQL in R.

Using packages

Exercise 1

Now load the packages we just installed. Note that the `library` function only accepts one argument, so you will need to load each of them separately.

Using packages

```r
library(stargazer)
library(tidyverse)
```

Warinings vs Errors

update

Outline

1. Introduction
2. Getting started
3. Using packages
4. **Quick summary statistics**
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Quick summary statistics

`summary(x, digits)` - equivalent to Stata's *summarize*, displays summary statistics. Its arguments are:

- **x:** the object you want to summarize, usually a vector or data frame
- **digits:** the number of decimal digits to be displayed

Exercise 1

Use the `summary()` function to display summary statistics for the *whr* data frame.

Quick summary statistics

```r
# Summary statistics
summary(whr)
```

```
##         country                                region        year     
##  Afghanistan:  3   Sub-Saharan Africa             :117   Min.   :2015  
##  Albania    :  3   Central and Eastern Europe     : 87   1st Qu.:2015  
##  Algeria    :  3   Latin America and Caribbean    : 68   Median :2016  
##  Angola     :  3   Western Europe                 : 63   Mean   :2016  
##  Argentina  :  3   Middle East and Northern Africa: 58   3rd Qu.:2017  
##  Armenia    :  3   Southeastern Asia              : 26   Max.   :2017  
##  (Other)    :452   (Other)                        : 51                 
##    happy_rank      happy_score        gdp_pc           family      
##  Min.   :  1.00   Min.   :2.693   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.: 40.00   1st Qu.:4.509   1st Qu.:0.6053   1st Qu.:0.7930  
##  Median : 79.00   Median :5.282   Median :0.9954   Median :1.0257  
##  Mean   : 78.83   Mean   :5.371   Mean   :0.9278   Mean   :0.9903  
##  3rd Qu.:118.00   3rd Qu.:6.234   3rd Qu.:1.2524   3rd Qu.:1.2287  
##  Max.   :158.00   Max.   :7.587   Max.   :1.8708   Max.   :1.6106  
## 
```

Quick summary statistics

`table()` - equivalent to `tabulate` in Stata, creates a frequency table. Its main arguments are the objects to be tabulated.

Exercise 2

Use the `table()` function to display frequency tables for:

1. The variable *year* in the *whr* data frame
2. The variables *region* and *year* in the *whr* data frame, simultaneously

Quick summary statistics

```r
# Year of data collection
table(whr$year)
```

```
##
## 2015 2016 2017
##  158  157  155
```

Quick summary statistics

```
# Gender of household head per year
table(whr$region, whr$year)
```

```
##
##                                    2015 2016 2017
##   Australia and New Zealand           2    2    2
##   Central and Eastern Europe         29   29   29
##   Eastern Asia                        6    6    6
##   Latin America and Caribbean        22   24   22
##   Middle East and Northern Africa    20   19   19
##   North America                       2    2    2
##   Southeastern Asia                   9    9    8
##   Southern Asia                       7    7    7
##   Sub-Saharan Africa                 40   38   39
##   Western Europe                     21   21   21
```

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Descriptives tables

We can also use the `stargazer()` function to quickly display a nice-looking descriptives table.

Stargazer was originally developed to export beautiful regression tables to LATEX or html, but it also allows you to generate summary statistics.

Descriptives tables

Exercise 3 - `stargazer()` summary statistics table

Use the `stargazer()` function to display summary statistics for the variables in the *whr* data frame.

The `stargazer()` function accepts **a lot** of arguments, most of which are beyond the scope of this session. Here are the arguments you'll need for this specific table:

- **x:** the object you want to summarize – in this case a vector or data frame
- **type:** the output format – "text" to just display, "latex" (the default) to save as a LaTeX table, and "html" for, yes, html
- **digits:** the number of decimal digits to be displayed

Descriptives tables

```
# A descriptive table with stargazer
stargazer(whr,
          digits = 1,
          type = "text")
```

```
##
## =================================================================
## Statistic       N   Mean   St. Dev.  Min  Pctl(25) Pctl(75)  Max
## -----------------------------------------------------------------
## year           470 2,016.0   0.8    2,015  2,015    2,017   2,017
## happy_rank     470  78.8     45.3     1      40      118     158
## happy_score    470   5.4     1.1     2.7    4.5      6.2     7.6
## gdp_pc         470   0.9     0.4     0.0    0.6      1.3     1.9
## family         470   1.0     0.3     0.0    0.8      1.2     1.6
## health         470   0.6     0.2     0.0    0.4      0.8     1.0
## freedom        470   0.4     0.2     0.0    0.3      0.5     0.7
## trust_gov_corr 470   0.1     0.1     0.0    0.1      0.2     0.6
## generosity     470   0.2     0.1     0.0    0.2      0.3     0.8
## dystopia_res   470   2.1     0.6     0.3    1.7      2.5     3.8
## -----------------------------------------------------------------
```

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Export tables to LaTeX

To export the table to LaTeX, we will use a couple of additional arguments of the `stargazer()` function:

- **out:** where to save the table, i.e., the file path, including the file name
- **covariate.labels:** a vector of variable labels

Export tables to LaTeX

But first, let's pick a few variables of interest in the whr data set so the table fits in these slides.

Exercise 4

UPDATE

1. Create a vector called covariates containing the string names of the variables you want to keep: happy_score, gdp_pc, family, and trust_gov_corr.
2. Use this vector to subset the whr dataset to contain only these variables. Call the new data frame whr_simp.

Export tables to LaTeX

```r
# Vector with covariates to be kept
covariates <- c("happy_score",
                "gdp_pc",
                "family",
                "trust_gov_corr")

# subset whr
whr_simp <- whr[, covariates]
```

Export tables to LaTeX

Exercise 4

1. Create a vector called `cov_labels` containing the labels to the covariates, in the same order as in the `covariates` vector.
2. Now use the stargazer function as in the previous exercise:

- Set `whr_simp` as the `x` argument this time
- Set the `covariate.labels` argument as the vector you just created

Export tables to LaTeX

```r
# Set labels
cov_labels <- c("Happy score", "GDP per capita",
                "Family", "Trust in gornment and currption")

# Save table to latex
stargazer(whr_simp,
          covariate.labels = cov_labels,
          #summary.stat = c("n", "mean", "sd", "min", "max"),
          digits = 2,
          out = file.path(rawOutput,"desc_table.tex"))
```

Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Happy score	470	5.37	1.14	2.69	4.51	6.23	7.59
GDP per capita	470	0.93	0.42	0.00	0.61	1.25	1.87
Family	470	0.99	0.32	0.00	0.79	1.23	1.61
Trust in gornment and currption	470	0.13	0.11	0.00	0.06	0.17	0.55

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Descriptives tables - Create tables from scratch

UPDATE

In R, it is relatively easy to construct any table you can think of by manipulating objects. To construct a table from scratch, we will use two functions:

- `aggregate()` - Similar to collapse in Stata, it can compute statistics of a variable based on the values of other variable
- `reshape()` - Reshapes data sets from long to wide and vice-versa

Descriptives tables - Create tables from scratch

`aggregate(X, by, FUN)`:

- **x**: a data frame or column
- **by**: a list of grouping variables
- **FUN**: a function to compute statistics

Descriptives tables - Create tables from scratch

UPDATE

Exercise 5

Use the aggregate function to create a data frame called year_inc_tab with the mean of the total income per year and treatment status. The syntax of the aggregate function is very similar to that of the collapse function in Stata.

```
# Aggregate income by year and treatment status
happy_table <-
  aggregate(happy_score ~ year + region,
            data = whr,
            FUN = mean)
```

Descriptives tables - Create tables from scratch

```
print(happy_table)
```

```
##    year                          region happy_score
## 1  2015       Australia and New Zealand    7.285000
## 2  2016       Australia and New Zealand    7.323500
## 3  2017       Australia and New Zealand    7.299000
## 4  2015      Central and Eastern Europe    5.332931
## 5  2016      Central and Eastern Europe    5.370690
## 6  2017      Central and Eastern Europe    5.409931
## 7  2015                    Eastern Asia    5.626167
## 8  2016                    Eastern Asia    5.624167
## 9  2017                    Eastern Asia    5.646667
## 10 2015     Latin America and Caribbean    6.144682
## 11 2016     Latin America and Caribbean    6.101750
## 12 2017     Latin America and Caribbean    5.957818
## 13 2015 Middle East and Northern Africa    5.406900
## 14 2016 Middle East and Northern Africa    5.386053
## 15 2017 Middle East and Northern Africa    5.369684
## 16 2015                   North America    7.273000
## 17 2016                   North America    7.254000
## 18 2017                   North America    7.154500
## 19 2015               Southeastern Asia    5.317444
## 20 2016               Southeastern Asia    5.338889
## 21 2017               Southeastern Asia    5.444875
```

Descriptives tables - Create tables from scratch

UPDATE

`reshape(data, varying, idvar, timevar, direction):`

- **data**: a data frame
- **idvar**: the variables that identify the group in the wide data set
- **timevar**: the variable in long format that differentiates multiple records from the same group or individual

Descriptives tables - Create tables from scratch

Exercise 6

Use the `reshape` function to make the `year_inc_tab` data frame wide per treatment status.

```
# Aggregate income by year and treatment status
happy_table <-
  spread(happy_table,
         key = year,
         value = happy_score)
```

For comparison, here's how you'd do it in Stata:

```
reshape wide x, i(year) j(treatment_hh)
```

Descriptives tables - Create tables from scratch

```r
print(happy_table)
```

```
##                              region     2015     2016     2017
## 1         Australia and New Zealand 7.285000 7.323500 7.299000
## 2        Central and Eastern Europe 5.332931 5.370690 5.409931
## 3                      Eastern Asia 5.626167 5.624167 5.646667
## 4       Latin America and Caribbean 6.144682 6.101750 5.957818
## 5   Middle East and Northern Africa 5.406900 5.386053 5.369684
## 6                     North America 7.273000 7.254000 7.154500
## 7                 Southeastern Asia 5.317444 5.338889 5.444875
## 8                     Southern Asia 4.580857 4.563286 4.628429
## 9                Sub-Saharan Africa 4.202800 4.136421 4.111949
## 10                   Western Europe 6.689619 6.685667 6.703714
```

Descriptives tables - Create tables from scratch

With a data frame as input, stargazer by default tries to summarize it. So, to export this table we must specify one additional argument: summary = F.

Exercise 7

Print the year_inc_tab table you created in exercise 6 using stargazer. If you want, you can also save it using the out option.

Descriptives tables - Create tables from scratch

UPDATE

```
# Label variables
column_lab <- c("Treatment status", "2012", "2013", "2014", "2016", "2018")

# Create table
stargazer(happy_table,
          summary = F,
          # Some extra formatting:
          #covariate.labels = column_lab,
          #title = "Total income by treatment status and year",
          header = F,
          digits = 1,
          rownames = F)
```

Table 2:

region	2015	2016	2017
Australia and New Zealand	7.3	7.3	7.3
Central and Eastern Europe	5.3	5.4	5.4
Eastern Asia	5.6	5.6	5.6
Latin America and Caribbean	6.1	6.1	6.0
Middle East and Northern Africa	5.4	5.4	5.4
North America	7.3	7.3	7.2
Southeastern Asia	5.3	5.3	5.4
Southern Asia	4.6	4.6	4.6
Sub-Saharan Africa	4.2	4.1	4.1
Western Europe	6.7	6.7	6.7

Outline

1 Introduction

2 Getting started

3 Using packages

4 Quick summary statistics

5 Descriptives tables

6 Export tables to LaTeX

7 Descriptives tables - Create tables from scratch

8 **Export tables to Excel**

9 Export regression table

10 References and recommendations

11 Appendix

Export tables to Excel

UPDATE

To export a table to excel we'll use the `write.table()` function. It takes a data frame object as input and saves it as a `.csv` file

`write.table()` is the most basic function, but there are many other functions that allow you to export formatted tables to Microsoft Excel, Word or PowerPoint. Here are some examples:

- ReporteRs
- Flextable
- r2excel (only available in GitHub).

Export tables to Excel

```
write.table(x, file = "", sep = " ", row.names = TRUE)
```

- **x**: the object to be written
- **file**: where to save the table, i.e., the file path including the file name
- **sep**: the field separator of the csv, Excel's default is comma
- **row.names**: either a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written
- **row.names**: same as `row.names` for columns

Export tables to Excel

Exercise 8

Use the `write.table()` function to save the `year_inc_tab` you table created in Exercise 6 into a csv file.

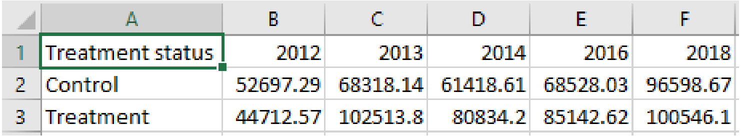
1. Set `x` arugment as *year_inc_tab*.
2. Set `row.names` as `FALSE`
3. Set `col.names` as a vector of labels
4. Set `file` as the folder path to your output folder plus a name for a file plus ".csv"
5. Set `sep` as `","`.

Tips:

- Make sure to save it in the *Raw Ouput* folder. You can you the function `file.path` to do it
- Use the help function to check syntax if needed

Export tables to Excel

```
# write.table(year_inc_tab,
#             sep = ",",
#             row.names = F,
#             col.names = c("Treatment status",
#                           "2012", "2013", "2014", "2016", "2018"),
#             file = file.path(rawOutput, "year_inc_tab.csv"))
```



	A	B	C	D	E	F
1	Treatment status	2012	2013	2014	2016	2018
2	Control	52697.29	68318.14	61418.61	68528.03	96598.67
3	Treatment	44712.57	102513.8	80834.2	85142.62	100546.1

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Export regression table

This is a session on *descriptive* analysis, so regressions are beyond its scope. But since you'll probably ask, here's how you run a regression and how you export a very simple regression table to LaTeX using `stargazer`:

```
# Run a Regression
# reg1 <- lm(expend_food_yearly ~
#             income_total_win + num_dependents,
#             data = whr)
```

Export regression table

```r
# Export a regression table

depvar_label <- "Yearly food expenditure (winsorized)"
covar_labels <- c("Total income (winsorized)",
                  "Number of dependents")

# stargazer(reg1,
#           title = "Regression table",
#           dep.var.labels = depvar_label,
#           covar_labels = covar_labels,
#           digits = 2,
#           header = F)
```

Export regression table

Export regression table

```
# Regression 1
# reg1 <- lm(expend_food_yearly ~
#             income_total_win + num_dependents,
#             data = whr)
#
# # Reg with year FE
# reg2 <- lm(expend_food_yearly ~
#             income_total_win + num_dependents + factor(year),
#             data = whr)
#
# # Reg with year and site FE
# reg3 <- lm(expend_food_yearly ~
#             income_total_win + num_dependents + factor(year) + factor(site_code),
#             data = whr)
```

Export regression table

```r
# Labels
depvar_label <- "Yearly food expenditure (winsorized)"
covar_labels <- c("Total income (winsorized)", "Number of dependents")

# Table
# stargazer(reg1,
#           reg2,
#           reg3,
#           font.size = "tiny",
#           title = "Regression table",
#           keep = c("ncome_total_win","num_dependents"),
#           dep.var.labels = depvar_label,
#           covariate.labels = covar_labels,
#           add.lines = list(c("Year FE", "No", "Yes", "Yes"),
#                            c("Site FE", "No", "No", "Yes")),
#           omit.stat = c("ser"),
#           digits = 2,
#           header = F)
```

Export regression table

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

References and recommendations

- Johns Hopkins Exploratory Data Analysis at Coursera: https://www.coursera.org/learn/exploratory-data-analysis
- Udacity's Data Analysis with R: https://www.udacity.com/course/data-analysis-with-r--ud651
- Jake Russ stargazer cheat sheet: https://www.jakeruss.com/cheatsheets/stargazer/

References and recommendations

Since we talked about LaTeX so much...

- DIME LaTeX templates and trainings: https://github.com/worldbank/DIME-LaTeX-Templates
- All you need to know about LaTeX: https://en.wikibooks.org/wiki/LaTeX

Outline

1. Introduction
2. Getting started
3. Using packages
4. Quick summary statistics
5. Descriptives tables
6. Export tables to LaTeX
7. Descriptives tables - Create tables from scratch
8. Export tables to Excel
9. Export regression table
10. References and recommendations
11. Appendix

Appendix - Formulas

update