Descriptive analysis

Field Coordinator Training - R Track

Luiza Andrade, Leonardo Viotti & Rob Marty

June 2018

Outline

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

Introduction

First, let's load the data that is going to be used in the training. Paths should be set in your master file!

Load the data

```
# Load CSV data
lwh <- read.csv(file.path(finalData,"lwh_clean.csv"),
                header = T)
```

Outline

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 *lwh* data frame.

Quick summary statistics

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

```
##     panel_id         hh_code           wave          year
##  Min.   :100103   Min.   :1001   Baseline:213   Min.   :2012
##  1st Qu.:208677   1st Qu.:2087   Endline :307   1st Qu.:2013
##  Median :402755   Median :4028   FUP1&2  :126   Median :2014
##  Mean   :402210   Mean   :4022   FUP3    :345   Mean   :2015
##  3rd Qu.:509629   3rd Qu.:5096   FUP4    :293   3rd Qu.:2016
##  Max.   :712501   Max.   :7125   NA's    :645   Max.   :2018
##  NA's   :645      NA's   :645                   NA's   :645
##     treatment_hh   treatment_site         site_code    gender_hhh
##  Control  :689   Control  :637    Kayanza 15   :279   Female:413
##  Treatment:595   Treatment:647    Kayanza 4    :231   Male  :784
##  NA's     :645   NA's     :645    Rwamangana 2 :159   NA's  :732
##                                   Rwamangana 33:199
##                                   Rwamangana 34:194
##                                   Rwamangana 35:222
##                                   NA's         :645
```

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 *lwh* data frame
2. The variables *gender_hhh* and *year* in the *lwh* data frame, simultaneously

Quick summary statistics

```
# Year of data collection
table(lwh$year)
```

```
##
## 2012 2013 2014 2016 2018
##  213  126  345  293  307
```

Quick summary statistics

```
# Gender of household head per year
table(lwh$gender_hhh, lwh$year)
```

```
##
##             2012 2013 2014 2016 2018
##    Female     61   41  109   94  108
##    Male      152   85  150  199  198
```

Outline

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.

It can also be used to export any data frames you create to LaTeX as a formatted table. To do that, you fist need to construct a data frame object combining vectors (of the same length) with the desired information.

Descriptives tables

If you haven't yet done this in your master, install and load the stargazer package now!

```
# Install stargazer
install.packages("stargazer",
                 dependencies = TRUE)

# Load stargazer
library(stargazer)
```

Descriptives tables

Exercise 3 - `stargazer()` summary statistics table

Use the `stargazer()` function to display summary statistics for the variables in the *lwh* 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(lwh,
          digits = 1,
          type = "text")
```

```
##
## ================================================================================
## Statistic                N      Mean     St. Dev.      Min     Pctl(25)  Pctl(75)      Max
## --------------------------------------------------------------------------------
## panel_id              1,284 402,210.3 202,211.7 100,103.0 208,677.0 509,629.0 712,501.0
## hh_code               1,284  4,022.1   2,022.1   1,001.0   2,086.8   5,096.2   7,125.0
## year                  1,284  2,015.0     2.1     2,012.0   2,013.0   2,016.0   2,018.0
## age_hhh                928    48.1      14.8      20.0      35.0      58.2      93.0
## num_dependents         339     2.1       1.4       0.0       1.0       3.0       6.0
## read_and_write         339     0.5       0.5       0.0       0.0       1.0       1.0
## w_gross_yield_a       1,284 87,599.4  112,690.1    0.0       0.0    129,342.6 483,333.3
## w_gross_yield_b       1,284 88,838.5  128,914.0    0.0       0.0    118,237.6 769,962.0
## expend_food_yearly    1,284 159,650.0 125,232.7    0.0    52,177.5  243,734.1 488,381.4
## expend_food_lastweek  1,284  3,059.7   2,400.1     0.0     1,000.0   4,671.2   9,360.0
## --------------------------------------------------------------------------------
```

Outline

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 `lwh` data set so the table fits in these slides.

Exercise 4

1. Create a vector called `covariates` containing the string names of the variables you want to keep: `age_hhh`, `num_dependents`, `income_total_win`, and `expend_food_yearly`.
2. Use this vector to subset the lwh dataset to contain only these variables. Call the new data frame `lwh_simp`.

Export tables to LaTeX

```
# Vector with covariates to be kept
covariates <- c("age_hhh",
                "num_dependents",
                "income_total_win",
                "expend_food_yearly")

# subset lwh
lwh_simp <- lwh[, 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 `lwh_simp` as the `x` argument this time
- Set the `covariate.labels` argument as the vector you just created

Export tables to LaTeX

```
# Set labels
cov_labels <- c("Age of household head", "Number of dependents",
                "Anual income (winsorized)", "Yearly food expediture")

# Save table to latex
stargazer(lwh_simp,
          covariate.labels = cov_labels,
          #summary.stat = c("n", "mean", "sd", "min", "max"),
          digits = 1,
          out = file.path(rawOutput,"desc_table.tex"))
```

Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Age of household head	928	48.1	14.8	20.0	35.0	58.2	93.0
Number of dependents	339	2.1	1.4	0.0	1.0	3.0	6.0
Anual income (winsorized)	1,929	50,090.5	66,020.0	0	0	81,216	192,000
Yearly food expediture	1,284	159,650.0	125,232.7	0.0	52,177.5	243,734.1	488,381.4

Outline

Descriptives tables - Create tables from scratch

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

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
year_inc_tab <-
  aggregate(x = lwh$income_total_win, # data.frame
            by = list(year = lwh$year, # list
                      treatment = lwh$treatment_hh),
            FUN = mean) # function
```

Note that the `income_total_win` variable is now named `x` in the `income` data frame

Descriptives tables - Create tables from scratch

```
print(year_inc_tab)
```

```
##    year treatment         x
## 1  2012   Control  47958.37
## 2  2013   Control  63482.25
## 3  2014   Control  59232.51
## 4  2016   Control  71106.46
## 5  2018   Control  98294.61
## 6  2012 Treatment  41459.83
## 7  2013 Treatment 104023.15
## 8  2014 Treatment  80672.20
## 9  2016 Treatment  85192.96
## 10 2018 Treatment  99510.29
```

Descriptives tables - Create tables from scratch

`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
year_inc_tab <- reshape(year_inc_tab,
                        idvar = "treatment",
                        timevar = "year",
                        direction = "wide")
```

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

```
print(year_inc_tab)
```

```
##   treatment   x.2012    x.2013   x.2014   x.2016   x.2018
## 1   Control 47958.37  63482.25 59232.51 71106.46 98294.61
## 6 Treatment 41459.83 104023.15 80672.20 85192.96 99510.29
```

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

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

# Create table
stargazer(year_inc_tab,
          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: Total income by treatment status and year

Treatment status	2012	2013	2014	2016	2018
Control	47,958.4	63,482.2	59,232.5	71,106.5	98,294.6
Treatment	41,459.8	104,023.1	80,672.2	85,193.0	99,510.3

Outline

Export tables to Excel

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

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 = lwh)
```

Export regression table

```
# 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

Table 3: Regression table

	Dependent variable:
	Yearly food expenditure (winsorized)
Total income (winsorized)	−0.20**
	(0.10)
Number of dependents	16,745.07***
	(4,536.70)
Constant	86,166.78***
	(12,662.45)
Observations	339
R^2	0.05
Adjusted R^2	0.04
Residual Std. Error	120,006.50 (df = 336)
F Statistic	8.65*** (df = 2; 336)
Note:	*p<0.1; **p<0.05; ***p<0.01

Export regression table

```
# Regression 1
reg1 <- lm(expend_food_yearly ~
           income_total_win + num_dependents,
           data = lwh)

# Reg with year FE
reg2 <- lm(expend_food_yearly ~
           income_total_win + num_dependents + factor(year),
           data = lwh)

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

Export regression table

```
# 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

Table 4: Regression table

	Dependent variable:		
	Yearly food expenditure (winsorized)		
	(1)	(2)	(3)
Total income (winsorized)	-0.20^{**}	0.11	0.07
	(0.10)	(0.07)	(0.07)
Number of dependents	$16{,}745.07^{***}$	4,657.54	$5{,}987.46^{*}$
	(4,536.70)	(3,454.66)	(3,430.74)
Year FE	No	Yes	Yes
Site FE	No	No	Yes
Observations	339	339	339
R^2	0.05	0.47	0.50
Adjusted R^2	0.04	0.47	0.49
F Statistic	8.65^{***} (df = 2; 336)	101.01^{***} (df = 3; 335)	41.04^{***} (df = 8; 330)
Note:			*p<0.1; **p<0.05; ***p<0.01

Outline

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