Descriptive Statistics and Regression Tables in Word and LaTeX

Bas Machielsen
a.h.machielsen@uu.nl
github.com/basm92

Utrecht University - R Cafe

December 4, 2019

Possibilities and packages

A small disclaimer..

- There are various possibilities and packages in R to make tidy, well-formatted tables
- I do not claim to know, let alone be familiar with, even a tiny fraction of them
- I do know how to use a few particular ones, most notable `sjPlot` and `stargazer`, which are among the most popular and most-used packages for this purpose, whose features I will demonstrate throughout this lecture
- I will also show a couple of alternatives to circumvent the shortcomings of these packages.
- Stargazer has a vignette available **here**, which is an excellent tutorial in its own right, and sjPlot has a few available tutorials **here**

Why use them?

- What is stargazer?
  stargazer is an R package that creates LATEX code, HTML code and ASCII text for well-formatted regression tables, with multiple models side-by-side, as well as for summary statistics tables, data frames, vectors and matrices.
- Why should you use stargazer? From the vignette:
  Compared to available alternatives, stargazer excels in at least three respects: its ease of use, the large number of models it supports, and its beautiful aesthetics. These advantages have made it the R-to-LATEX package of choice for many satisfied users at research and teaching institutions around the world.

Why use them?

- What is sjPlot?
  [a] collection of plotting and table output functions for data visualization. Results of various statistical analyses (that are commonly used in social sciences) can be visualized using this package, including simple and cross tabulated frequencies, histograms, box plots, (generalized) linear models, mixed effects models, PCA and correlation matrices, cluster analyses, scatter plots, Likert scales, effects plots of interaction terms in regression models, constructing index or score variables and much more.
- sjPlot has way more features than stargazer. We will not focus on these features in this session.

Descriptive statistics - stargazer

We will start out by using stargazer in LaTeXformat. The advantage of a LaTeXformat is that you can use it right away in a presentation, such as this one!

```
#Load an example dataset
#the data comes from Hamermesh and Parker (2005)
#about impact of beauty on teacher evaluations

data(TeachingRatings)

stargazer(TeachingRatings,
          header=FALSE, type='latex',
          summary.stat = c("mean","min","max","n"),
          font.size = "footnotesize")
```

Descriptive statistics - stargazer

- This is the output you get:

Table 1:

Statistic	N	Mean	Min	Max
age	463	48.365	29	73
beauty	463	0.00000	−1.450	1.970
eval	463	3.998	2.100	5.000
students	463	36.624	5	380
allstudents	463	55.177	8	581

Descriptive statistics - stargazer

- The option list of stargazer is very extensive. For example, the same output can be achieved by omitting several (non-default) statistics:

```
#Load an example dataset
#the data comes from a Hamermesh and Parker (2005)
#about impact of beauty on teacher evaluations

data(TeachingRatings)

stargazer(TeachingRatings,
          header=FALSE, type='latex',
          omit.summary.stat = c("sd", "p25", "p75"),
          font.size = "footnotesize",
          title = "Hello, R Cafe!"
          )
```

Descriptive statistics - stargazer

The output you get is the following:

Table 2: Hello, R Cafe!

Statistic	N	Mean	Min	Max
age	463	48.365	29	73
beauty	463	0.00000	−1.450	1.970
eval	463	3.998	2.100	5.000
students	463	36.624	5	380
allstudents	463	55.177	8	581

Please refer to the stargazer vignette **here** to look for the specific summary statistic codes.

Descriptive statistics - stargazer

- Hopefully the advantages have been clear. The syntax is amazingly simple and the number of options is huge (Have a look at `?stargazer`). It automatically filters NA observations. It is also easy to integrate these tables into (R)Markdown and LaTeXdocuments.

Now, a few disadvantages..

- You have to specify the option `header = FALSE` in order to bypass the default output which contains the credit of the package creator (try it!)
- stargazer automatically extracts the variables which are suitable for descriptive statistics out of your dataset. If you want a summarise of, e.g., factor variables, you have to convert them.

Descriptive statistics - stargazer in Word

- Stargazer also supports exporting tables to Microsoft Word.
- It is possible to export tables in one file, and to overwrite or append these documents. (This is also supported for .tex files.)

```
#Let's try to extract another dataset, CPS1988,
#about the determinants of wages
#and summarise the descriptives in a Word table.
data(CPS1988)

stargazer(CPS1988[CPS1988$ethnicity == "cauc",],
          header=FALSE, type='latex',
          omit.summary.stat = c("p25", "p75"),
          font.size = "footnotesize",
          title = "Caucasian")

stargazer(CPS1988[CPS1988$ethnicity == "afam",],
          header=FALSE, type='latex',
          omit.summary.stat = c("p25", "p75"),
          font.size = "footnotesize",
          title = "Caucasian")
```

Descriptive statistics - stargazer in Word

This is the output of the two tables:

Table 3: Caucasian

Statistic	N	Mean	St. Dev.	Min	Max
wage	25,923	617.234	461.210	50.050	18,777.200
education	25,923	13.132	2.902	0	18
experience	25,923	18.153	13.040	−4	63

Table 4: African American

Statistic	N	Mean	St. Dev.	Min	Max
wage	2,232	446.853	312.437	52.230	3,527.340
education	2,232	12.327	2.767	0	18
experience	2,232	18.741	13.513	−2	61

Descriptive statistics - stargazer in Word

- Exporting tables to Word is possible "natively" in 2 ways:
  1. First, output to .html, and copy the resulting tables from your web browser to Word
  2. Second, output each separate stargazer command to different .doc documents
- Stargazer does not yet support appending existing documents with new tables, and automatically overwrites tables.

Syntax:

```
stargazer(CPS1988[CPS1988$ethnicity == "cauc",],
          header=FALSE, omit.summary.stat = c("p25", "p75"),
          font.size = "footnotesize",
          title = "Caucasian",
          type='html', out = "test1.doc") #These 2 arguments are the only 2 you
                                          #need

stargazer(CPS1988[CPS1988$ethnicity == "afam",],
          header=FALSE, omit.summary.stat = c("p25", "p75"),
          font.size = "footnotesize",
          title = "African American",
          type='html', out = "test2.doc") #These 2 arguments are the only 2 you
                                          #need
```

Descriptive statistics - stargazer in Word

- However, if you want to append, you can use a little bit of R's power in the following way:

```
#Step 1, make a list for all the
#partitions of the data, include names
models <-  list(
    Caucasian = CPS1988[CPS1988$ethnicity == "cauc",],
    AfroAmerican = CPS1988[CPS1988$ethnicity == "afam",],
    Northeast = CPS1988[CPS1988$region == "northeast",],
    Midwest = CPS1988[CPS1988$region == "midwest",],
    South = CPS1988[CPS1988$region == "south",],
    West = CPS1988[CPS1988$region == "west",]
)

#Step 2, for loop for every element in the list you want to print
for (m in 1:length(models)){
  a = vector(length = length(models))
  a[m] = (names(models[m]))
  s = capture.output(stargazer(models[m], type = "html", title = a[m]))
  cat(paste(s,"\n"),file="tables.doc",append=TRUE)
  cat(" ",file="tables.doc",append=TRUE)
}
```

Descriptive statistics - stargazer in Word

- One only has to put all the partitions in a list, change the filename, and change the desired descriptive statistics accordingly.
- Let us now look at the **document in which all tables are**! This is very useful when organizing all tables and/or graphs for a paper.

Descriptives in sjPlot

- It is also possible to use `sjPlot` for descriptive statistics.

```
#Let's use a dataset of determinants of GDP Growth
data(GrowthDJ)
descriptives <- descr(GrowthDJ, show =
                        c("type","label","n","mean","sd"))

kable(descriptives, caption = "Descriptive Statistics",
      booktabs = TRUE, row.names = FALSE) %>%
  kable_styling(latex_options = "striped")
```

Descriptives in sjPlot

Table 5: Descriptive Statistics

var	type	label	n	mean	sd
oil	categorical	oil	121	1.809917	0.3939977
inter	categorical	inter	121	1.619835	0.4874457
oecd	categorical	oecd	121	1.181818	0.3872983
gdp60	numeric	gdp60	116	3681.818966	7492.8776368
gdp85	numeric	gdp85	108	5683.259259	5688.6708192
gdpgrowth	numeric	gdpgrowth	117	4.094017	1.8914641
popgrowth	numeric	popgrowth	107	2.279439	0.9987481
invest	numeric	invest	121	18.157025	7.8533096
school	numeric	school	118	5.526271	3.5320372
literacy60	numeric	literacy60	103	48.165048	35.3542568

Descriptives in sjPlot

- `sjPlot` has a lot of advantages. It automatically omits NA observations, and transforms factor variables to numeric variables (although this is dangerous!).
- It can also make contingency tables, which stargazer cannot (readily) do:

```
sjt.xtab(GrowthDJ$oil, GrowthDJ$oecd,
         use.viewer = TRUE, file = "contingency.doc")
```

oil	*oecd*		***Total***
	no	yes	
yes	23	0	23
no	76	22	98
Total	99	22	121

χ^2=4.892 · *df*=1 · φ=0.228 · *Fisher's p*=0.013

Correlation tables

- Both `sjPlot` and `stargazer` can make correlation tables. First, let's try stargazer:

```
data(GrowthDJ)
cormat <- select_if(GrowthDJ, is.numeric)
```

```
stargazer(cor(cormat[5:7],
              use = c('pairwise.complete.obs')),
          header=FALSE, type='latex',
          omit.summary.stat = c("p25", "p75"),
          font.size = "footnotesize",
          title = "Correlation Matrix")
```

- Stargazer has no native stars, a feature which sjPlot does have.

Correlation tables - stargazer

- This is the output:

Table 6: Correlation Matrix

	invest	school	literacy60
invest	1	0.622	0.639
school	0.622	1	0.818
literacy60	0.639	0.818	1

- LaTeX-users: `stargazer` cannot correctly handle negative numbers when generating the LaTeXcode, causing some compilers to have difficulties.
- Word and LaTeX-users: you can also use stargazer's `out`-option to export the tables to Word and .tex.

Correlation tables

- I can also use the package `xtable`, which surpasses the problem of stargazer but creates otherwise identical tables (see .Rmd-file for the code).

	gdp60	gdp85	gdpgrowth	popgrowth	invest	school
gdp60	1.000	0.631	-0.122	0.291	0.091	0.337
gdp85	0.631	1.000	0.139	-0.222	0.581	0.697
gdpgrowth	-0.122	0.139	1.000	0.242	0.351	0.198
popgrowth	0.291	-0.222	0.242	1.000	-0.332	-0.213
invest	0.091	0.581	0.351	-0.332	1.000	0.622
school	0.337	0.697	0.198	-0.213	0.622	1.000

Table 7: Correlation Matrix

- `xtable` also supports output in LaTeXor html format.

Correlation tables - sjPlot

- `sjPlot` has a range of functions to visualize correlations. For example:

```
sjt.corr(cormat[1:6], file = "cortab.doc")
```

	gdp60	*gdp85*	*gdpgrowth*	*popgrowth*	*invest*	*school*
gdp60		0.630***	-0.126	0.290**	0.083	0.351***
gdp85	0.630***		0.132	-0.250*	0.585***	0.723***
gdpgrowth	-0.126	0.132		0.238*	0.378***	0.219*
popgrowth	0.290**	-0.250*	0.238*		-0.339***	-0.206*
invest	0.083	0.585***	0.378***	-0.339***		0.607***
school	0.351***	0.723***	0.219*	-0.206*	0.607***	

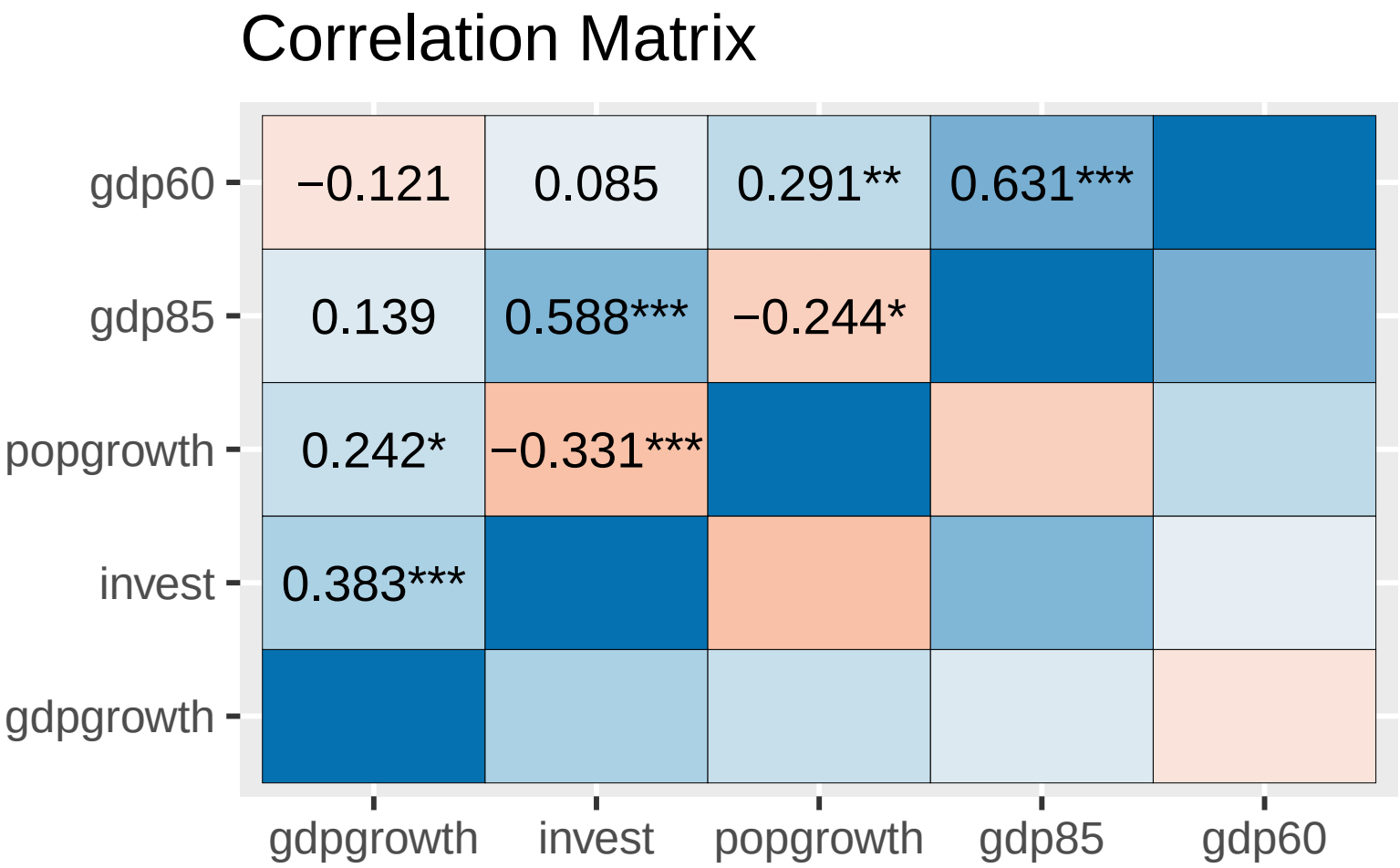
Computed correlation used pearson-method with listwise-deletion.

Figure 2: Correlation table

Correlation tables - sjPlot

- Another example (this uses ggplot2)

```
sjp.corr(cormat[1:5], title = "Correlation Matrix")
```



Correlation tables - with stars

- If you would like a correlation function that prints correlation tables with stars indicating significance, have a look **here**
- Example:

```
xtable(corstarsl(cormat[,1:6]),
       caption = "Correlation Table with Stars")
```

	gdp60	gdp85	gdpgrowth	popgrowth	invest
gdp60					
gdp85	0.63***				
gdpgrowth	-0.12	0.14			
popgrowth	0.29**	-0.22*	0.24*		
invest	0.09	0.58***	0.35***	-0.33***	
school	0.34***	0.70***	0.20*	-0.21*	0.62***

Table 8: Correlation Table with Stars

Regression tables

- Now, finally, we can use stargazer's principal application, regression tables.
- Everything is *very* straightforward: just look at this:

```
data(TeachingRatings)
model1 <- lm(data = TeachingRatings,
             eval ~ beauty)
model2 <- lm(data =TeachingRatings,
             eval ~ beauty + age)
model3 <- lm(data =TeachingRatings,
             eval ~ beauty + age + gender)
model4 <- lm(data =TeachingRatings,
             eval ~ beauty + age + gender + students)

stargazer(model1, model2, model3, model4,
          header = FALSE,
          caption = FALSE,
          font.size = "footnotesize",
          column.sep.width = "0pt",
          omit.stat = c("ll", "F","ser"))
```

Regression tables

Table 9:

	Dependent variable:			
	eval			
	(1)	(2)	(3)	(4)
beauty	0.133***	0.134***	0.140***	0.141***
	(0.032)	(0.034)	(0.033)	(0.034)
age		0.0003	−0.003	−0.003
		(0.003)	(0.003)	(0.003)
genderfemale			−0.211***	−0.212***
			(0.053)	(0.053)
students				−0.0001
				(0.001)
Constant	3.998***	3.984***	4.213***	4.220***
	(0.025)	(0.134)	(0.144)	(0.146)
Observations	463	463	463	463
R^2	0.036	0.036	0.068	0.068
Adjusted R^2	0.034	0.032	0.062	0.060
Note:				*p<0.1; **p<0.05; ***p<0.01

Table 10:

Regression tables

- `stargazer` can also change styles to fit standard formats required by journals in the social sciences, and automatically incorporates different dependent variables.

Table 11:

	Dependent variable:			
	eval		beauty	
	(1)	(2)	(3)	(4)
beauty	0.133^{***}	0.134^{***}		
	(0.032)	(0.034)		
age		0.0003		-0.024^{***}
		(0.003)		(0.004)
eval			0.270^{***}	
			(0.066)	
tenureyes			0.010	
			(0.088)	
Constant	3.998^{***}	3.984^{***}	-1.086^{***}	1.159^{***}
	(0.025)	(0.134)	(0.282)	(0.177)
Observations	463	463	463	463
R^2	0.036	0.036	0.036	0.089
Adjusted R^2	0.034	0.032	0.032	0.087
Note:				*p<0.1; **p<0.05; ***p<0.01

Standard errors

The issue of standard errors is unrelated to the packages that are use to *report* your data. The way to go about this is to change the standard errors in the model list to the appropriate standard errors calculated by another package, in this case, `sandwich`.

```
model1 <- lm(data = GrowthDJ,
             gdpgrowth ~ oil + inter + invest + school)

library(sandwich)

heterosk_vcov <- vcovHC(model1, type = "HC3")
h_ac_vcov <- vcovHAC(model1)

stargazer(model1,
          coeftest(model1, vcov = heterosk_vcov),
          coeftest(model1, vcov = vcovHAC(model1) ))
```

Standard errors

Table 13:

	Dependent variable:		
	gdpgrowth		
	OLS	*coefficient test*	
	(1)	(2)	(3)
oilno	-1.125^{*}	-1.125^{*}	-1.125^{*}
	(0.583)	(0.677)	(0.643)
interyes	1.497^{***}	1.497^{***}	1.497^{***}
	(0.446)	(0.444)	(0.436)
oecdyes	-1.528^{***}	-1.528^{***}	-1.528^{***}
	(0.488)	(0.531)	(0.486)
invest	0.099^{***}	0.099^{***}	0.099^{***}
	(0.026)	(0.035)	(0.032)
school	-0.026	-0.026	-0.026
	(0.062)	(0.076)	(0.069)
Constant	2.711^{***}	2.711^{***}	2.711^{***}
	(0.646)	(0.817)	(0.748)
Observations	115		
R^2	0.262		
Adjusted R^2	0.228		
Note:		*p<0.1; **p<0.05; ***p<0.01	

Conclusion

- Stargazer and sjPlot are two packages that can save you a lot of trouble (as long as you are not looking for correlation tables with stars).
- Supplementary packages, such as xtable and hmisc could also help you in reporting the most common statistical analyses.
- Thank you for your attention! Suggestions? a.h.machielsen@uu.nl!