Creating publication-ready Word tables in R

Sara Weston and Debbie Yee

12/09/2016

Has this happened to you?

You're working on a draft of a manuscript with your adviser, and one of her edits is something like, "Did we control for time of day? If not we should."

So you rerun the results in R (or whatever) and the coefficients and the standard errors and the t-values change. And you spend 2 hours updating the manuscript.

Has this happened to you?

And then you send it back, and her next edit is, “Oh, we also have to control for gender.”

You scream into the void.

Oh, and then you find that you skipped a row, so all your numbers are off and have to be reentered.

Your time is valuable

You have better things to do than constantly re-update tables in manuscripts and posters and talks.

In fact, it's not worth your time to make any table by hand ever.

You are a human and you will make mistakes

No matter how smart you are, how careful you are, how much coffee you have had to drink, you *will* make mistakes when you create tables by hand.

Sorry, but them's the facts.

Have we mentioned that R is amazing?

You might not have heard that R can do anything.

Before today, you were thinking that anything meant any analyses. Or, if you were generous, any analyses and any graph.

Did you know that R can make your tables for you? Formatted, with stars and everything? And by tables, I mean descriptive tables, correlation matrices, summaries of regressions, summaries of mixed-effects models.

Packages you should download

- sjPlot
- stargazer

These packages are the key to saving time and saving face.

sjPlot

Benefits

- Easier to use
- Can format correlation matrices
- A greater variety of descriptive options
- Colors!
- Also includes functions to plot regression diagnostics and results

Cons

- Less customizable
- Standardization doesn't work

sjPlot

- descriptive tables
- contingency tables
- correlation matrices
- linear models

(These are just some of the table types.)

sjPlot

```r
library(sjPlot)
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Spec
## 1          5.1         3.5          1.4         0.2  set
## 2          4.9         3.0          1.4         0.2  set
## 3          4.7         3.2          1.3         0.2  set
## 4          4.6         3.1          1.5         0.2  set
## 5          5.0         3.6          1.4         0.2  set
## 6          5.4         3.9          1.7         0.4  set
```

sjPlot - descriptive table

If you type the following code into Rstudio, you will see what the table looks like in the Viewer window.

```
sjt.df(iris,
       file="sjt_des.doc") #CRUCIAL - save this as a
                           #word document
```

sjPlot - descriptive table

Variable	*vars*	*n*	*missings*	*missings (percentage)*	*mean*	*sd*	*median*	*trimmed*	*mad*	*min*	*max*	*range*	*skew*	*kurtosis*	*se*
Sepal.Length	1	150	0	0	5.84	0.83	5.8	5.81	1.04	4.3	7.9	3.6	0.31	-0.61	0.07
Sepal.Width	2	150	0	0	3.06	0.44	3	3.04	0.44	2	4.4	2.4	0.31	0.14	0.04
Petal.Length	3	150	0	0	3.76	1.77	4.35	3.76	1.85	1	6.9	5.9	-0.27	-1.42	0.14
Petal.Width	4	150	0	0	1.2	0.76	1.3	1.18	1.04	0.1	2.5	2.4	-0.1	-1.36	0.06
Species*	5	150	0	0	2	0.82	2	2	1.48	1	3	2	0	-1.52	0.07
LongPetal	6	150	0	0	0.42	0.5	0	0.4	0	0	1	1	0.32	-1.91	0.04

Figure 1:

sjPlot - descriptive table

There are lots of ways you can customize the table.

```
sjt.df(iris,
       altr.row.col = T, # this colors the rows
       title = "Descriptive statistics", #always give
                                         #your tables
                                         #titles
       file = "sjt_des_2.doc")
```

sjPlot - descriptive table

Descriptive statistics

Variable	*vars*	*n*	*missings*	*missings (percentage)*	*mean*	*sd*	*median*	*trimmed*	*mad*	*min*	*max*	*range*	*skew*	*kurtosis*	*se*
Sepal.Length	1	150	0	0	5.84	0.83	5.8	5.81	1.04	4.3	7.9	3.6	0.31	-0.61	0.07
Sepal.Width	2	150	0	0	3.06	0.44	3	3.04	0.44	2	4.4	2.4	0.31	0.14	0.04
Petal.Length	3	150	0	0	3.76	1.77	4.35	3.76	1.85	1	6.9	5.9	-0.27	-1.42	0.14
Petal.Width	4	150	0	0	1.2	0.76	1.3	1.18	1.04	0.1	2.5	2.4	-0.1	-1.36	0.06
Species*	5	150	0	0	2	0.82	2	2	1.48	1	3	2	0	-1.52	0.07
LongPetal	6	150	0	0	0.42	0.5	0	0.4	0	0	1	1	0.32	-1.91	0.04

Figure 2:

sjPlot - contingency table

First, create a new variable with me

```r
iris$LongPetal <- ifelse(iris$Petal.Length > 4.5, 1, 0)
```

Now we can create a contingency table to see whether one species of flower is more likely to have long petals.

```r
sjt.xtab(iris$Species, #rows
         iris$LongPetal, #columns
         file = "sjt_contingency.doc")
```

sjPlot - contingency table

Species	*LongPetal*		**Total**
	0	1	
setosa	50	0	50
versicolor	36	14	50
virginica	1	49	50
Total	87	63	150

$X^2=104.598 \cdot df=2 \cdot \Phi_c=.835 \cdot p<.001$

Figure 3:

sjPlot - correlation matrix

Specify which variables you want to see in the correlation matrix using any method of subsetting or indexing you like.

```
sjt.corr(iris[,c("Sepal.Length", "Sepal.Width",
                 "Petal.Length", "Petal.Width")],
         file="sjt_corr.doc")
```

sjPlot - correlation matrix

	Sepal.Length	*Sepal.Width*	*Petal.Length*	*Petal.Width*
Sepal.Length		-0.167*	0.882***	0.834***
Sepal.Width	-0.167*		-0.310***	-0.289***
Petal.Length	0.882***	-0.310***		0.938***
Petal.Width	0.834***	-0.289***	0.938***	
Computed correlation used spearman-method with listwise-deletion.				

Figure 4:

Note: * $= p < .05$, ** $= p < .01$, *** $= p < .001$

sjPlot - correlation matrix

Note the defaults for this are the *spearman* method of calculating correlations and *list-wise* deletion. This is different from other functions you may be familiar with, which often default to the pearson method and pairwise deletion. This will be printed with the table, in case you forget.

```
sjt.corr(iris[,c("Sepal.Length", "Sepal.Width",
                 "Petal.Length", "Petal.Width")],
         corr.method = "pearson",
         na.deletion = "pairwise",
         file="sjt_corr_2.doc")
```

sjPlot - correlation matrices

	Sepal.Length	*Sepal.Width*	*Petal.Length*	*Petal.Width*
Sepal.Length		-0.118	0.872***	0.818***
Sepal.Width	-0.118		-0.428***	-0.366***
Petal.Length	0.872***	-0.428***		0.963***
Petal.Width	0.818***	-0.366***	0.963***	
Computed correlation used pearson-method with pairwise-deletion.				

Figure 5:

a quick side note

One of the more useful but never discussed features of R is the ability to assign attributes to objects. Often you want your column names to be short (this helps keep code clean) but you also want to know exactly what the variable is. Also, you probably want to have some punctuation and spaces in the variable name. You can have both with attributes.

```
# create a fake data frame
new.data.frame <- data.frame(age = c(18, 20, 24, 17, 43),
                             edu = c(1, 3, 1, 5, 2),
                             rt = c(.01, .04, .02,
                                    .10, .06))
#assign label to columns
attr(new.data.frame$age, "label") <- "Age in years"
attr(new.data.frame$edu, "label") <- "Education"
attr(new.data.frame$rt, "label") <- "Reaction time"
```

a quick side note

sjPlot will use the labels as column and row names.

```
sjt.corr(new.data.frame, fade.ns=F,
         corr.method = "pearson",
         file="sjt_corr_3.doc")
```

	Age in years	*Education*	*Reaction time*
Age in years		-0.275	0.083
Education	-0.275		0.910*
Reaction time	0.083	0.910*	

Computed correlation used pearson-method with listwise-deletion.

sjPlot - linear model

Estimate a few linear models.

```
mod1 <- lm(Petal.Length ~ Species,
           data=iris)
mod2 <- lm(Petal.Length ~ Species + Petal.Width,
           data=iris)
mod3 <- lm(Petal.Length ~ Species + Petal.Width +
             Species:Petal.Width,
           data=iris)
```

Put all models into one table simultaneously (just like you might in a manuscript).

```
sjt.lm(mod1, mod2, mod3,
       file="sjt_linear.doc")
```

sjPlot - linear models

	Petal.Length			Petal.Length			Petal.Length		
	B	*CI*	*p*	*B*	*CI*	*p*	*B*	*CI*	*p*
(Intercept)	1.46	1.34 – 1.58	**<.001**	1.21	1.08 – 1.34	**<.001**	1.33	1.07 – 1.59	**<.001**
Species									
Speciesversicolor	2.80	2.63 – 2.97	**<.001**	1.70	1.34 – 2.06	**<.001**	0.45	-0.28 – 1.19	.227
Speciesvirginica	4.09	3.92 – 4.26	**<.001**	2.28	1.72 – 2.83	**<.001**	2.91	2.11 – 3.72	**<.001**
Petal.Width				1.02	0.72 – 1.32	**<.001**	0.55	-0.42 – 1.52	.267
Speciesversicolor:Petal.Width							1.32	0.23 – 2.42	**.019**
Speciesvirginica:Petal.Width							0.10	-0.94 – 1.14	.848
Observations	150			150			150		
R^2 / adj. R^2	.941 / .941			.955 / .954			.959 / .958		

Figure 7:

sjPlot - linear models

```
sjt.lm(mod1, mod2, mod3,
       pred.labels = c("Veriscolor", "Virginica",
                       "Petal Width",
                       "Versicolor x Petal Width",
                       "Virginica x Petal Width"),
       show.ci = F, show.se = T,
       depvar.labels = c("Petal Length",
                         "Petal Length",
                         "Petal Length"),
       file="sjt_linear_2.doc")
```

sjPlot - linear models

	Petal Length			Petal Length			Petal Length		
	B	*std. Error*	*p*	*B*	*std. Error*	*p*	*B*	*std. Error*	*p*
(Intercept)	1.46	0.06	**<.001**	1.21	0.07	**<.001**	1.33	0.13	**<.001**
Species									
Veriscolor	2.80	0.09	**<.001**	1.70	0.18	**<.001**	0.45	0.37	.227
Virginica	4.09	0.09	**<.001**	2.28	0.28	**<.001**	2.91	0.41	**<.001**
Petal Width				1.02	0.15	**<.001**	0.55	0.49	.267
Versicolor x Petal Width							1.32	0.56	**.019**
Virginica x Petal Width							0.10	0.52	.848
Observations	150			150			150		
R^2 / adj. R^2	.941 / .941			.955 / .954			.959 / .958		

Figure 8:

sjPlot - linear models

But what if I'm predicting multiple outcomes with the same models? Go for it!

```
mod4 <- lm(Sepal.Length ~ Species,
           data=iris)
mod5 <- lm(Sepal.Length ~ Species + Petal.Width,
           data=iris)
mod6 <- lm(Sepal.Length ~ Species + Petal.Width +
             Species:Petal.Width,
           data=iris)
```

```
sjt.lm(mod1, mod2, mod3, mod4, mod5, mod6,
       pred.labels = c("Veriscolor", "Virginica",
                       "Petal Width",
                       "Versicolor x Petal Width",
                       "Virginica x Petal Width"),
       show.ci = F, show.se = T,
       file="sjt_linear_3.doc")
```

sjPlot - linear models

	Petal.Length			Petal.Length			Petal.Length			Sepal.Length			Sepal.Length			Sepal.Length		
	B	*std. Error*	*p*	*B*	*std. Error*	*p*	*B*	*std. Error*	*p*	*B*	*std. Error*	*p*	*B*	*std. Error*	*p*	*B*	*std. Error*	*p*
(Intercept)	1.46	0.06	**<.001**	1.21	0.07	**<.001**	1.33	0.13	**<.001**	5.01	0.07	**<.001**	4.78	0.08	**<.001**	4.78	0.17	**<.001**
Species																		
Veriscolor	2.80	0.09	**<.001**	1.70	0.18	**<.001**	0.45	0.37	.227	0.93	0.10	**<.001**	-0.06	0.23	.794	-0.73	0.50	.141
Virginica	4.09	0.09	**<.001**	2.28	0.28	**<.001**	2.91	0.41	**<.001**	1.58	0.10	**<.001**	-0.05	0.36	.889	0.49	0.54	.362
Petal Width				1.02	0.15	**<.001**	0.55	0.49	.267				0.92	0.19	**<.001**	0.93	0.65	.154
Versicolor x Petal Width							1.32	0.56	**.019**							0.50	0.74	.501
Virginica x Petal Width							0.10	0.52	.848							-0.28	0.70	.688
Observations	150			150			150			150			150			150		
R^2 / adj. R^2	.941 / .941			.955 / .954			.959 / .958			.619 / .614			.669 / .663			.677 / .666		

Figure 9:

sjPlot - more models

This package can handle general linear models (e.g., binary models, probit models). To create these tables, use the function sjt.glm.

Also, you can create summaries of mixed effects models using sjt.lmer and sjt.glmer.

stargazer

Pros

- more control over output
- can select which predictors you want in the table, more options for fit statistics
- can import different estiamtes for coefficients, standard errors, confidence intervals and p values (if you want to use odds ratios or adjusted pvalues)
- one function, no matter what you're tabling
- cleaner descriptive table
- works with Sweave and knitr

Cons

- more options = more complicated
- can't create formatted correlation tables
- more limited in terms of descriptives
- can't see output in viewer

stargazer - descriptives

```r
library(stargazer)
stargazer(iris,
          #note you have to specify type
          type = "html",
          #note that the argument is "out" not "file"
          out="star_descriptive.doc")
```

stargazer - descriptive

Statistic	N	Mean	St. Dev.	Min	Max
Sepal.Length	150	5.843	0.828	4.300	7.900
Sepal.Width	150	3.057	0.436	2.000	4.400
Petal.Length	150	3.758	1.765	1.000	6.900
Petal.Width	150	1.199	0.762	0.100	2.500
LongPetal	150	0.420	0.495	0	1

Figure 10:

stargazer - descriptives

```
stargazer(iris,
          digits=2,
          title="Descriptive Statistics",
          type = "html",
          out="star_descriptive_2.doc")
```

stargazer - descriptive

Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Max
Sepal.Length	150	5.84	0.83	4.30	7.90
Sepal.Width	150	3.06	0.44	2.00	4.40
Petal.Length	150	3.76	1.77	1.00	6.90
Petal.Width	150	1.20	0.76	0.10	2.50
LongPetal	150	0.42	0.50	0	1

Figure 11:

stargazer - linear model

```r
stargazer(mod1, mod2, mod3,
          type="html",
          out="star_linear.doc")
```

stargazer - linear model

	Dependent variable:		
	Petal.Length		
	(1)	(2)	(3)
Speciesversicolor	2.798***	1.698***	0.454
	(0.086)	(0.181)	(0.374)
Speciesvirginica	4.090***	2.277***	2.913***
	(0.086)	(0.281)	(0.406)
Petal.Width		1.019***	0.546
		(0.152)	(0.490)
Speciesversicolor:Petal.Width			1.323**
			(0.555)
Speciesvirginica:Petal.Width			0.101
			(0.525)
Constant	1.462***	1.211***	1.328***
	(0.061)	(0.065)	(0.131)
Observations	150	150	150
R^2	0.941	0.955	0.959
Adjusted R^2	0.941	0.954	0.958
Residual Std. Error	0.430 (df = 147)	0.378 (df = 146)	0.361 (df = 144)
F Statistic	1,180.161*** (df = 2; 147)	1,035.992*** (df = 3; 146)	681.915*** (df = 5; 144)
Note:			*p**p***p<0.01

Figure 12:

stargazer - linear model

```
stargazer(mod1, mod2, mod3, mod4, mod5, mod6,
          type="html",
          out="star_linear_2.doc",
          intercept.bottom = F,
          intercept.top = T,
          ci = T, digits=2,
          notes = "This is a caption.",
          model.names = T,
          single.row = T,
          covariate.labels = c("Constant","Veriscolor",
                               "Virginica", "Petal Width",
                               "Versicolor x Petal Width",
                               "Virginica x Petal Width"))
```

stargazer - linear model

	Dependent variable:					
	Petal.Length			Sepal.Length		
	OLS			*OLS*		
	(1)	(2)	(3)	(4)	(5)	(6)
Constant	1.46*** (1.34, 1.58)	1.21*** (1.08, 1.34)	1.33*** (1.07, 1.58)	5.01*** (4.86, 5.15)	4.78*** (4.62, 4.94)	4.78*** (4.44, 5.12)
Veriscolor	2.80*** (2.63, 2.97)	1.70*** (1.34, 2.05)	0.45 (-0.28, 1.19)	0.93*** (0.73, 1.13)	-0.06 (-0.51, 0.39)	-0.73 (-1.70, 0.24)
Virginica	4.09*** (3.92, 4.26)	2.28*** (1.73, 2.83)	2.91*** (2.12, 3.71)	1.58*** (1.38, 1.78)	-0.05 (-0.75, 0.65)	0.49 (-0.56, 1.55)
Petal Width		1.02*** (0.72, 1.32)	0.55 (-0.41, 1.51)		0.92*** (0.54, 1.30)	0.93 (-0.34, 2.20)
Versicolor x Petal Width			1.32** (0.23, 2.41)			0.50 (-0.95, 1.94)
Virginica x Petal Width			0.10 (-0.93, 1.13)			-0.28 (-1.64, 1.08)
Observations	150	150	150	150	150	150
R^2	0.94	0.96	0.96	0.62	0.67	0.68
Adjusted R^2	0.94	0.95	0.96	0.61	0.66	0.67
Residual Std. Error	0.43 (df = 147)	0.38 (df = 146)	0.36 (df = 144)	0.51 (df = 147)	0.48 (df = 146)	0.48 (df = 144)
F Statistic	1,180.16*** (df = 2; 147)	1,035.99*** (df = 3; 146)	681.92*** (df = 5; 144)	119.26*** (df = 2; 147)	98.53*** (df = 3; 146)	60.31*** (df = 5; 144)
Note:						*p**p***p<0.01
						This is a caption.

Figure 13:

stargazer - general linear model

```
mod7 <- glm(LongPetal ~ Species + Petal.Width,
            data=iris,
            family = "binomial")
```

```
stargazer(mod1, mod2, mod7,
          type="html",
          out="star_linear_3.doc",
          intercept.bottom = F,
          intercept.top = T,
          digits=2)
```

stargazer - linear model

	Dependent variable:		
	Petal.Length		LongPetal
	OLS		*logistic*
	(1)	(2)	(3)
Constant	1.46***	1.21***	-22.77
	(0.06)	(0.07)	(2,387.15)
Veriscolor	2.80***	1.70***	11.34
	(0.09)	(0.18)	(2,387.15)
Virginica	4.09***	2.28***	13.05
	(0.09)	(0.28)	(2,387.15)
Petal Width		1.02***	7.61***
		(0.15)	(2.35)
Observations	150	150	150
R^2	0.94	0.96	
Adjusted R^2	0.94	0.95	
Log Likelihood			-26.31
Akaike Inf. Crit.			60.63
Residual Std. Error	0.43 (df = 147)	0.38 (df = 146)	
F Statistic	1,180.16*** (df = 2; 147)	1,035.99*** (df = 3; 146)	
Note:			*p**p***p<0.01

Figure 14: