APA Style Cheat Sheet

Enter Yo' Name

10/1/2017

##install papaja special library  
##you only have to do this part once  
##comment it out when you are done  
#devtools::install\_github("crsh/papaja")  
  
##load papaja  
library(papaja)  
  
##load MOTE for decimals  
library(MOTE)  
  
library(knitr)

Last week we covered how to pull numbers from chunks and report them outside the chunk - using MOTE's apa function. Turns out papaja has that function too! :)

Let's run an example from my class to demonstrate a couple new things today:

##import the data  
library(haven)  
master = read\_spss("c10 viagra.sav")  
  
##run an ANOVA  
library(ez)  
##you must have a participant number for ezANOVA  
master$partno = 1:nrow(master)  
master$dose = as.factor(master$dose)  
options(scipen = 999)  
  
ezoutput = ezANOVA(data = master,  
 dv = libido,  
 between = dose,  
 wid = partno,  
 type = 3,   
 detailed = T)

## Warning: Converting "partno" to factor for ANOVA.

ezoutput

## $ANOVA  
## Effect DFn DFd SSn SSd F p p<.05  
## 1 (Intercept) 1 12 180.26667 23.6 91.661017 0.0000005720565 \*  
## 2 dose 2 12 20.13333 23.6 5.118644 0.0246942895382 \*  
## ges  
## 1 0.8842381  
## 2 0.4603659  
##   
## $`Levene's Test for Homogeneity of Variance`  
## DFn DFd SSn SSd F p p<.05  
## 1 2 12 0.1333333 6.8 0.1176471 0.8900225

##running a one way anova - if Levene's Test is significant  
aovoutput = oneway.test(libido ~ dose, data = master)  
  
aovoutput

##   
## One-way analysis of means (not assuming equal variances)  
##   
## data: libido and dose  
## F = 4.3205, num df = 2.0000, denom df = 7.9434, p-value = 0.05374

##normal LM type output  
lmoutput = lm(libido ~ dose, data = master)  
  
lmoutput

##   
## Call:  
## lm(formula = libido ~ dose, data = master)  
##   
## Coefficients:  
## (Intercept) dose2 dose3   
## 2.2 1.0 2.8

#apa\_print(ezoutput)  
apa\_print(aovoutput)

## Warning in if (tolower(names(x$parameter)) == "df") {: the condition has  
## length > 1 and only the first element will be used

## $estimate  
## NULL  
##   
## $statistic  
## [1] "$F = 4.32$, $p = .054$"  
##   
## $full\_result  
## NULL  
##   
## $table  
## NULL

apa\_print(lmoutput)

## $estimate  
## $estimate$Intercept  
## [1] "$b = 2.20$, 95\\% CI $[0.83$, $3.57]$"  
##   
## $estimate$dose2  
## [1] "$b = 1.00$, 95\\% CI $[-0.93$, $2.93]$"  
##   
## $estimate$dose3  
## [1] "$b = 2.80$, 95\\% CI $[0.87$, $4.73]$"  
##   
## $estimate$modelfit  
## $estimate$modelfit$r2  
## [1] "$R^2 = .46$, 90\\% CI $[0.05$, $0.70]$"  
##   
## $estimate$modelfit$r2\_adj  
## [1] "$R^2\_{adj} = .37$"  
##   
## $estimate$modelfit$aic  
## [1] "$AIC = 57.37$"  
##   
## $estimate$modelfit$bic  
## [1] "$BIC = 60.20$"  
##   
##   
##   
## $statistic  
## $statistic$Intercept  
## [1] "$t(12) = 3.51$, $p = .004$"  
##   
## $statistic$dose2  
## [1] "$t(12) = 1.13$, $p = .282$"  
##   
## $statistic$dose3  
## [1] "$t(12) = 3.16$, $p = .008$"  
##   
## $statistic$modelfit  
## $statistic$modelfit$r2  
## [1] "$F(2, 12) = 5.12$, $p = .025$"  
##   
##   
##   
## $full\_result  
## $full\_result$Intercept  
## [1] "$b = 2.20$, 95\\% CI $[0.83$, $3.57]$, $t(12) = 3.51$, $p = .004$"  
##   
## $full\_result$dose2  
## [1] "$b = 1.00$, 95\\% CI $[-0.93$, $2.93]$, $t(12) = 1.13$, $p = .282$"  
##   
## $full\_result$dose3  
## [1] "$b = 2.80$, 95\\% CI $[0.87$, $4.73]$, $t(12) = 3.16$, $p = .008$"  
##   
## $full\_result$modelfit  
## $full\_result$modelfit$r2  
## [1] "$R^2 = .46$, 90\\% CI $[0.05$, $0.70]$, $F(2, 12) = 5.12$, $p = .025$"  
##   
##   
##   
## $table  
## Predictor $b$ 95\\% CI $t(12)$ $p$  
## 1 Intercept 2.20 $[0.83$, $3.57]$ 3.51 .004  
## 2 Dose2 1.00 $[-0.93$, $2.93]$ 1.13 .282  
## 3 Dose3 2.80 $[0.87$, $4.73]$ 3.16 .008

# Printing when you can use papaja

We talked last week about the apa\_print() function. You can use that with S3 and S4 class objects, meaning lm/aov type answers. - You would use apa\_print(SAVEDOUTPUT), so here that would be apa\_print(aovoutput) - Depending on what is in the output, depends on what you can get out of it.

So we can print the F statistics from the one-way test with , . However, that's not the best, since the *df* are missing. We can explore how to report manually below.

LM outputs work even better. You get lots of options! Mostly, we might consider doing something like this: , 90% CI , , , .

# Printing when you can't use papaja

So, what can we do if we want to dynamically use our numbers but can't use apa\_print? We can figure out how to pull the numbers we want.

I can print the whole ANOVA table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Effect | DFn | DFd | SSn | SSd | F | p | p<.05 | ges |
| (Intercept) | 1 | 12 | 180.26667 | 23.6 | 91.661017 | 0.0000006 | \* | 0.8842381 |
| dose | 2 | 12 | 20.13333 | 23.6 | 5.118644 | 0.0246943 | \* | 0.4603659 |

I'd probably just want to report it APA style. Using my understanding of matrices, I can figure out how to get numbers out I want.

df = 12

F = 5.1186441

p = 0.0246943

eta = 0.4603659

Putting that all together, I can do: *F*(2, 12) = 5.12, *p* = .025, = .46.

(as an aside, you do not need Latex to print to word with the symbols, only PDF).

post = pairwise.t.test(master$libido,  
 master$dose,  
 p.adjust.method = "bonferroni",   
 paired = F,   
 var.equal = T)  
  
post$p.value

## 1 2  
## 2 0.84475161 NA  
## 3 0.02480431 0.1955766

post$p.value[1]; post$p.value[2]; post$p.value[3]; post$p.value[4]

## [1] 0.8447516

## [1] 0.02480431

## [1] NA

## [1] 0.1955766

post$p.value[1,1]; post$p.value[2,1]; post$p.value[1,2]; post$p.value[2,2]

## [1] 0.8447516

## [1] 0.02480431

## [1] NA

## [1] 0.1955766

# Printing use papaja's printnum and printp

We talked about the apa() function in the MOTE library last week. This week, let's cover printnum() and printp() in papaja.

printnum works by: - first, put in the number

* second, gt1 = T or F for greater than 1 (aka the leading zero issue)
* third, zero = T or F for if a number can be zero
* fourth, margin = 1 or 2 for matrices ... 1 for rows, 2 for columns, like the apply function

And a couple of other options, but these are the main ones.

So for F = 5.12, we can use the defaults because F can be greater than one and can also be zero.

But for eta = .46.

You can also use the digits command to get more or less digits: .460.

printp is a separate wrapper with a very handy function that prints the < symbol for numbers less than .001, and automatically assumes three decimals as per APA style.

.025

< .001

# Printing post hoc information

Using that information - let's pull out the post hoc p-values to report.

I could say one versus two was not significant, *p* = .845, while one versus three was significant, *p* = .025, and finally two versus three was not significant, *p* = .196. We'd want to include effect sizes, which is what we covered last week, if you are interested in MOTE and how to report those values.

# Printing a table

I've covered the kable function before for my SEM class, but let's cover apa\_table() in this video.

1. To manually make a table, first create a blank matrix to put your information in.
2. Fill in your table with information you want to print.
3. Use apa\_table!

(#tab:table)*This part goes at the top*

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Column 2 | Column 3 | Column 4 |
| Row 1 | 4.00 | 2.00 | 6.00 |
| Row 2 | .845 | .025 | .196 |
| Row 3 | MORE | ROWS | ETC. |

*Note.* This part goes at the bottom.