

ANOVA

Weeks 1 – 3

Chris Ives

Contents

- Foundations
 - Importing Data
 - Descriptives
 - Plots
- ANOVA
 - One-way ANOVA
 - Pairwise Comparisons
 - Planned Contrasts
 - Power Analysis
- Appendix
 - Custom Contrast Sets

Installing the **needs** Package

- Can install packages and load them
- No longer need **install.packages()** and **library()**
- It will ask if you would like to load it every time RStudio opens. Select yes.

```
install.packages("needs")
```

Importing the Data

Normally we would use the `import()` function to import our data. However, with the SPSS `.sav` files we don't get factor labels.

```
data_import <- import(here("data/Lab2_Vocab.sav"))  
head(data_import)
```

```
##      idnum vocab instruct  
## 1         1    53         1  
## 2         2    49         1  
## 3         3    47         1  
## 4         4    42         1  
## 5         5    51         1  
## 6         6    34         1
```

Notice how we have 1s down the instruct column.

The Fix

For now, when working with `.sav` files we will use the `read.sav` function from the `misty` package. `use.value.labels = TRUE` tells it to use the labels as the cell values.

```
needs(misty)
l2_data <- misty::read.sav(here("data/Lab2_Vocab.sav"),
                           use.value.labels = TRUE)
head(l2_data)
```

```
##      idnum vocab          instruct
## 1         1    53 physical science
## 2         2    49 physical science
## 3         3    47 physical science
## 4         4    42 physical science
## 5         5    51 physical science
## 6         6    34 physical science
```

Checking The Assumptions of ANOVA

- Independence of observations
- Normality
- Homogeneity of variance

Descriptive Statistics

Make sure the **psych** package is installed and loaded:

```
needs(psych)
```

describe from the **psych** package is one of the more popular functions for descriptive statistics.

```
describe(l2_data)
```

	vars	n	me...	sd	median	trimmed	mad
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
idnum	1	36	18.5	10.5356538	18.5	18.5	13.3434
vocab	2	36	33.5	12.8652355	35.5	34.2	12.6021
instruct*	3	36	2.0	0.8280787	2.0	2.0	1.4826

3 rows | 1–8 of 14 columns

Independence of Observations

- Do participants cross groups?
- We know this if we know the study design
- In this study, each student experienced one and only one lecture

Normality

- We can check the skew and kurtosis values from our `describe()` output.
- 0 ± 2 is a good rule of thumb for a tenable assumption of normality.

	vars	n	skew	kurtosis
	<int>	<dbl>	<dbl>	<dbl>
idnum	1	36	0.0000000	-1.3003629
vocab	2	36	-0.5359932	-0.8726857
instruct*	3	36	0.0000000	-1.5821759

3 rows

- Instruct is a categorical variable, so we can ignore the skew and kurtosis on that.

Visual Inspection for Normality

- For the most part, plots will be wrapped using the `ggplot()` function.

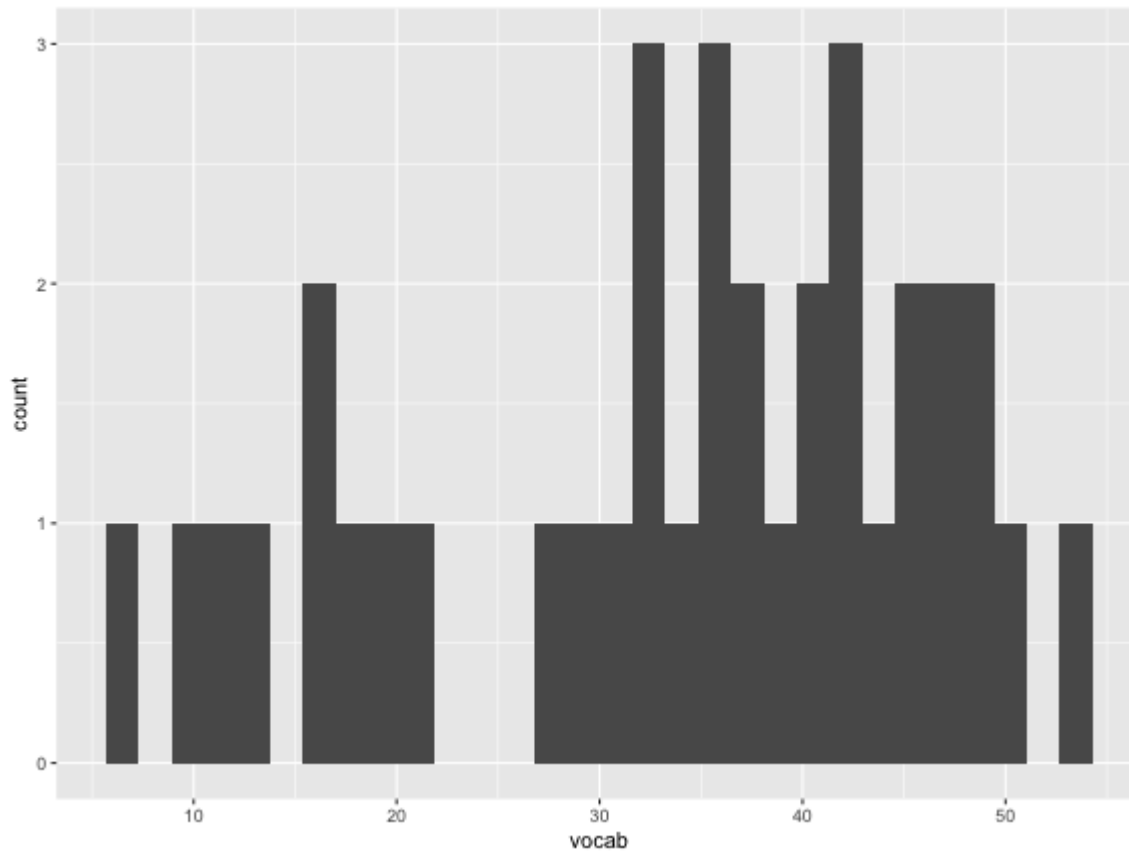
Histogram

```
ggplot(data = l2_data, aes(x = vocab)) +  
  geom_histogram()
```

- `aes()` refers to aesthetics. What are the variables we want represented in our plots? Since we just want counts of a single continuous variable, we just need to specify our `x` (i.e., `x = vocab`).

Histograms

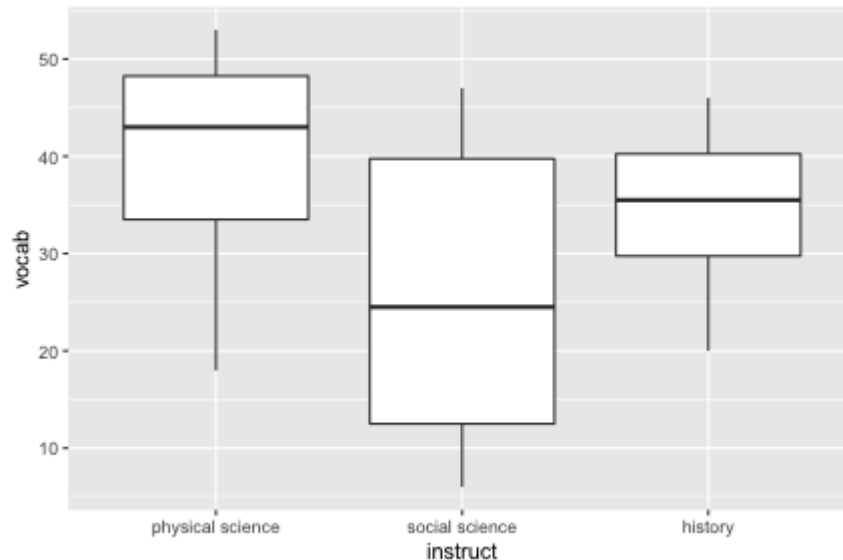
```
ggplot(data = l2_data, aes(x = vocab)) +  
  geom_histogram()
```



Boxplot

- To get boxplots, we just substitute `geom_histogram()` for `geom_boxplot()` and modify our aesthetics.

```
ggplot(data = l2_data, aes(x = instruct, y = vocab)) +  
  geom_boxplot()
```



Homogeneity of Variance

Homogeneity test is a separate analysis. We'll just use the `leveneTest` function for the `car` package. Formula is the same as for the ANOVA we want to run. Specify `center = "mean"` (function's default is median) to match SPSS results.

```
needs(car)
car::leveneTest(vocab ~ instruct, data = l2_data, center = "mean")
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##           Df F value    Pr(>F)
## group      2  7.5054 0.002058 **
##           33
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our significant result shows error variance around the *mean* is not equal across groups.

Running ANOVAs

Example here is from Lab 1 data.

ANOVAs will be run using the `anova_test()` function. Formula is specified as `DV ~ IV`. My convention is to number my model objects (e.g., m1, m2, etc.)

```
needs(rstatix)
m1 <- anova_test(data = l1_data, formula = vocab ~ instruct, deta
```

```
## Coefficient covariances computed by hccm()
```

```
m1
```

```
## ANOVA Table (type II tests)
##
##      Effect  SSn  SSd DFn DFd      F      p p<.05  ges
## 1 instruct 1176 3824   1  22 6.766 0.016      * 0.235
```

Effect Sizes

Effect sizes can be specified using `effect.size = ____` in the `anova.test()` function. Use `effect.size = "ges"` for generalized eta squared or `effect.size = "pes"` for partial eta squared. Default is `"ges"`.

```
anova_test(data = l1_data, formula = vocab ~ instruct,  
            effect.size = "ges")
```



```
anova_test(data = l1_data, formula = vocab ~ instruct,  
            effect.size = "ges")
```

```
## Coefficient covariances computed by hccm()
```

```
## ANOVA Table (type II tests)
```

```
##
```

```
##      Effect DFn DFd      F      p p<.05    ges  
## 1 instruct   1  22 6.766 0.016      * 0.235
```

```
anova_test(data = l1_data, formula = vocab ~ instruct,  
            effect.size = "pes")
```

```
## Coefficient covariances computed by hccm()
```

```
## ANOVA Table (type II tests)
```

```
##
```

```
##      Effect DFn DFd      F      p p<.05    pes  
## 1 instruct   1  22 6.766 0.016      * 0.235
```

Notice they are the same for a one-way between subjects ANOVA.

Posthoc Comparisons | Tukey

If you write out `"data = l2data"` the function will produce an error since it designed for multiple types of objects. Here we just type `"l2data"`.

```
rstatix::tukey_hsd(l2_data, formula = vocab ~ instruct)
```

	term <chr>	group1 <chr>	group2 <chr>	null.value <dbl>
1	instruct	physical science	social science	0
2	instruct	physical science	history	0
3	instruct	social science	history	0

3 rows | 1–5 of 10 columns

Posthoc Comparisons | Games–Howell

Remember, Games–Howell is used if the assumption of variance homogeneity is violated.

```
rstatix::games_howell_test(vocab ~ instruct, data = l2_data,  
                           conf.level = 0.95,  
                           detailed = FALSE)
```

.y.	group1	group2	estimate
<chr>	<chr>	<chr>	<dbl>
1 vocab	physical science	social science	–14.0
2 vocab	physical science	history	–5.5
3 vocab	social science	history	8.5

3 rows | 1–5 of 9 columns

Posthoc Comparisons | Bonferroni

```
pairwise_t_test(data = l2_data, vocab ~ instruct,  
                p.adjust.method = "bonferroni")
```

	.y.	group1	group2	n1	n2	p
	<chr>	<chr>	<chr>	<int>	<int>	<dbl>
1	vocab	physical science	social science	12	12	0.00651
2	vocab	physical science	history	12	12	0.26200
3	vocab	social science	history	12	12	0.08700

3 rows | 1–7 of 10 columns

Specified Contrasts

(Lab 3)

- They are not automatic in R and are a bit difficult. It is not important that you understand the reasoning behind all of the following steps.

Specified Contrasts

Check the order of your levels.

```
levels(l3_data$instruct)
```

```
## [1] "physical science" "social science"   "history"
```

Make sure they are in the order that you would like and code your contrasts accordingly. As an example, If they need to be reordered use:

```
l3_data$instruct <- factor(l3_data$instruct, levels = c("social science", "physical science", "history"))
```

Note: We will be using the original order in the following contrasts.

Step 1: Set your contrasts. We'll do Helmert here.

```
contrast1 <- c(1, -.5, -.5) # physical science vs others
contrast2 <- c(0, 1, -1) #social science vs history
```

Step 2: Bind the vectors into a temporary matrix. Constant should be equal to $1/(\text{length of your vectors})$.

```
mat.temp <- rbind(constant=1/3, contrast1, contrast2)
mat.temp
```

```
##           [,1]      [,2]      [,3]
## constant  0.3333333  0.3333333  0.3333333
## contrast1 1.0000000 -0.5000000 -0.5000000
## contrast2 0.0000000  1.0000000 -1.0000000
```

Step 3: Take the inverse of the matrix using the `solve()` function. Then we are dropping the first column with the constants.

```
mat <- solve(mat.temp)
mat
```

```
##      constant contrast1 contrast2
## [1,]         1  0.6666667         0.0
## [2,]         1 -0.3333333         0.5
## [3,]         1 -0.3333333        -0.5
```

```
mat<- mat[ , -1]
mat
```

```
##      contrast1 contrast2
## [1,]  0.6666667         0.0
## [2,] -0.3333333         0.5
## [3,] -0.3333333        -0.5
```


Step 4: Run your model formula using `lm()` and set contrasts. Here we are linking our "instruct" variable with our contrast matrix. Remember:

- contrast1 = physical science vs others
- contrast2 = social science vs history

```
m_contrasts <- lm(vocab ~ instruct, data=l3_data, contrasts = list(instruct = contrast1, contrast2 = contrast2))
summary(m_contrasts)
```

```
##
## Call:
## lm(formula = vocab ~ instruct, data = l3_data, contrasts = list(instruct = contrast1, contrast2 = contrast2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.000  -10.000    1.750    9.375   21.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.500      1.968  17.026  <2e-16 ***
## instructcontrast1   9.750      4.174   2.336  0.0257 *
## instructcontrast2  -8.500      4.819  -1.764  0.0870 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How do I get the SDs for calculating effect sizes?

We will use the `mutate` function to code new groupings. Then, we will run descriptives on this variable. My `case_when` function reads: If `instruct = social science` or `instruct = history`, code contrast1 as `humanities`. If `instruct = physical science` code as `physical science`.

```
l3_data <- l3_data %>%  
  mutate(contrast1 = case_when(instruct == "social science" | ins  
                              instruct == "physical science" ~ "  
  
contrast1_desc <- describe(l3_data ~ contrast1)
```

```
rmarkdown::paged_table(contrast1_desc$humanities)
```

	vars	n	mean	sd	median	trimmed	
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
idnum	1	24	24.50	7.0710678	24.5	24.50	
vocab	2	24	30.25	12.7696992	34.0	30.85	
instruct*	3	24	2.50	0.5107539	2.5	2.50	
contrast1*	4	24	1.00	0.0000000	1.0	1.00	

4 rows | 1–7 of 14 columns

```
rmarkdown::paged_table(contrast1_desc`physical science`)
```

	vars	n	me...	sd	median	trimmed	mad	
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
idnum	1	12	6.5	3.605551	6.5	6.5	4.4478	
vocab	2	12	40.0	10.795622	43.0	40.9	11.8608	
instruct*	3	12	1.0	0.000000	1.0	1.0	0.0000	
contrast1*	4	12	1.0	0.000000	1.0	1.0	0.0000	

4 rows | 1–8 of 14 columns

Pooling SDs

Here I am saving the second element of each descriptive table (since vocab is the second row in the data frame) to calculate pooled SDs.

```
sd1 <- contrast1_desc$humanities$sd[2]
sd2 <- contrast1_desc$`physical science`$sd[2]

cohens_sd <- sqrt((sd1^2 + sd2^2)/2)
cohens_sd
```

```
## [1] 11.82393
```

```
n1 <- contrast1_desc$humanities$n[2]
n2 <- contrast1_desc$`physical science`$n[2]

hedges_sd <- sqrt(((n1-1)*sd1^2 + (n2-1)*sd2^2)/(n1+n2-2))
hedges_sd
```

```
## [1] 12.16613
```

Calculate Hedges' g

Now that we have our pooled SD and coefficient, we just need to run the calculation for contrast1.

```
coefficients(m_contrasts)
```

```
##           (Intercept) instructcontrast1 instructcontrast2  
##           33.50           9.75           -8.50
```

```
hedges_g <- 9.75/hedges_sd  
hedges_g
```

```
## [1] 0.8014052
```

Calculate Effect Size

For an unbalanced dataset, we will use `pwr.2p2n.test` from the `pwr` package to run a power analysis on our contrast. `h` will be equal to our effect size. We already saved our `n`'s earlier, so `n1` will just be equal to `n1` (same for `n2`).

```
needs(pwr)
```

```
pwr.2p2n.test(h = hedges_g, n1 = n1, n2 = n2, power = NULL, sig.level = 0.05)
```

```
##  
##      difference of proportion power calculation for binomial distribution  
##  
##              h = 0.8014052  
##              n1 = 24  
##              n2 = 12  
##      sig.level = 0.05  
##              power = 0.6204959  
##      alternative = two.sided  
##  
## NOTE: different sample sizes
```

What about my ANOVA results?

To get the anova output, just run

`rstatix::anova_test()` on your `m_contrasts` object.

SPSS uses Type III SS by default, so I am matching it here.

```
contrast_anova <- anova_test(m_contrasts,  
                             type="III",  
                             effect.size = "ges",  
                             detailed = TRUE)
```

```
## Coefficient covariances computed by hccm()
```

```
contrast_anova
```

```
## ANOVA Table (type tests)
```

```
##
```

##	Effect	SSn	SSd	DFn	DFd	F	p	p<.05	ges
## 1	(Intercept)	40401	4599	1	33	289.896	6.57e-18	*	0.898
## 2	instruct	1194	4599	2	33	4.284	2.20e-02	*	0.206

Appendix

Contrasts

Isolated vs. Orthogonal Sets. If you are conducting an isolated contrast, or non-orthogonal set, you will need to use a different process for specifying contrasts. Here I demonstrate doing a single contrast. The primary difference is using `ginv()` from the **MASS** package rather than `solve()`.

```
contrast0 <- c(1, -.5, -.5)
mat.temp <- rbind(constant = 1/2, contrast0)
mat.temp
```

```
##           [,1] [,2] [,3]
## constant   0.5  0.5  0.5
## contrast0  1.0 -0.5 -0.5
```

```
mat <- MASS::ginv(mat.temp)
mat <- mat[, -1]
mat
```

Essentially, R fills in the rest of the matrix. This means that there will be "extra" contrasts in the output that you will need to ignore. Here **instruct1** corresponds to my first and only contrast. **instruct2** can be ignored.

```
m_contrasts <- lm(vocab ~ instruct, data=l3_data, contrasts = list(instruct = "contrasts"))
m_contrasts
```

```
##
## Call:
## lm(formula = vocab ~ instruct, data = l3_data, contrasts = list(instruct = "contrasts"))
##
## Coefficients:
## (Intercept)      instruct1      instruct2
##          33.50           9.75           6.01
```

If you want to make it easier to refer back to your contrasts, you can rename them in your output using the following code. Here I chose to keep the name of the intercept, name my contrast, and make everything else unnamed.

```
attr(m_contrasts$coefficients, "names") <- c("Intercept", "Phys Sci vs. Others", "m_contrasts")
```

```
##  
## Call:  
## lm(formula = vocab ~ instruct, data = l3_data, contrasts = list(instruct = m_contrasts))  
##  
## Coefficients:  
##              Intercept  Phys Sci vs. Others  
##              33.50             9.75             6.01
```