

PSYC 6380 Psychological Applications of Multivariate Analysis

David Stanley

2023-05-30

Table of contents

Preface	4
1 One-way ANOVA via Regression	5
1.1 Using Contrasts	5
1.2 Treatement/Dummy Coding	5
1.2.1 Original Data	6
1.2.2 Set Factor with Reference Group	7
1.2.3 Regression with Treatment Contrast	7
1.2.4 Treatment Contrasts Explained	7
1.2.5 ANOVA Summary Information	9
1.3 Sum Contrast / Effect Contrast	10
1.3.1 Behind the scenes	12
1.3.2 ANOVA values	14
1.4 Helmert Contrast	15
1.5 Summary: Contrast Types	15
2 Two-way ANOVA via Regression	18
2.1 Conducting a 2-way ANOVA	18
2.1.1 Activate Packages	18
2.1.2 Load Data	18
2.1.3 Inspect Data	18
2.1.4 Make Factors	20
2.1.5 Linear Model	20
2.2 Regression Becoming ANOVA	21
2.3 Contrasts for Categorical Variables	22
2.3.1 Gender Contrasts	22
2.3.2 Alcohol Contrasts	24
2.3.3 Interaction Contrasts	25
2.4 Regression command (i.e., lm) overview	26
2.5 Full and Restricted Models	28
2.5.1 Full Model	28
2.5.2 Restricted Models	28
2.6 Logic: Model Comparison	29
2.7 Explanation 1: Comparing Models	29
2.7.1 Sex	29

2.7.2	Alcohol	30
2.7.3	Interaction	31
2.8	Degrees of Freedom	31
2.9	Explanation 2: Sum of Squares to ANOVA	32
2.9.1	Intercept	33
2.9.2	Sex	33
2.9.3	Alcohol	33
2.9.4	Interaction	34
2.9.5	Error	34
2.9.6	SS to ANOVA	35

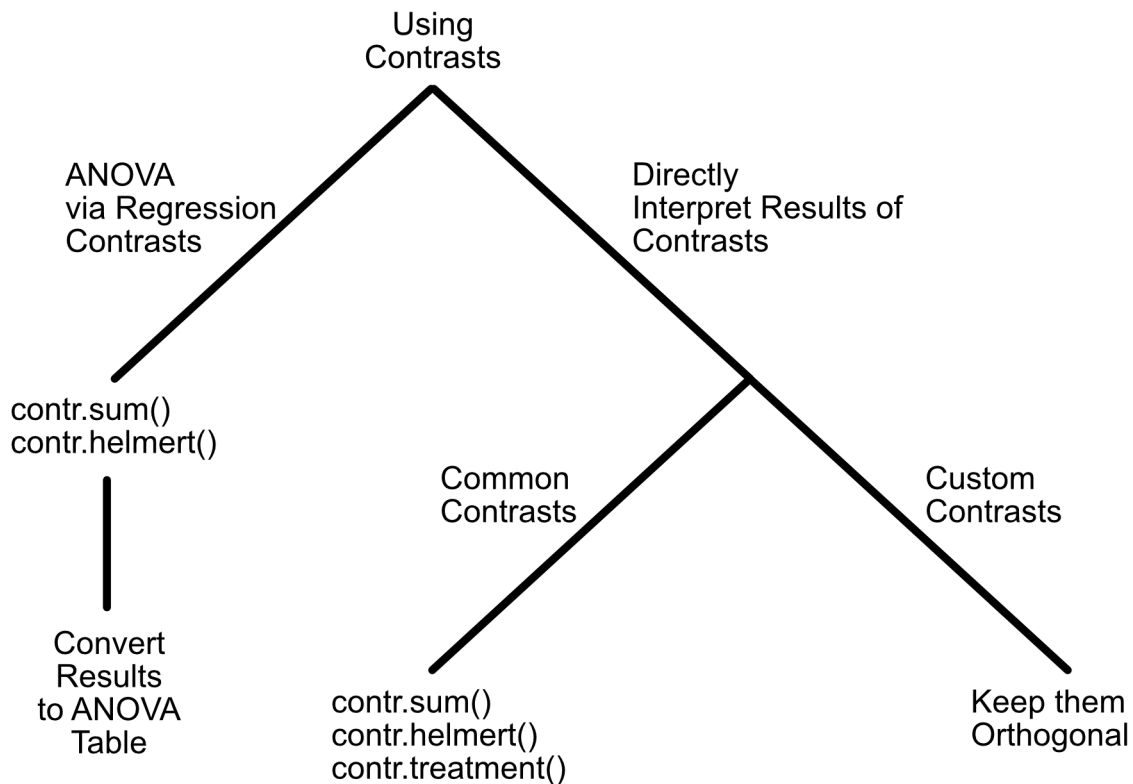
References	36
-------------------	-----------

Preface

This is a Quarto book designed to supplement PSYC 6380.

1 One-way ANOVA via Regression

1.1 Using Contrasts



1.2 Treatment/Dummy Coding

Use the Treatment contrast ONLY when you are interested in the contrast itself. DO NOT use if you are interested in typical ANOVA results (main effect, main effect, interaction, etc.).

This approach is the default approach in R unless you specify otherwise. In most cases, this is NOT what you want in Psychology analyses.

Comparisons are to one specific level of the Independent Variables that we call the reference group.

1.2.1 Original Data

```
print(viagra)
```

	libido	dose
1	3	placebo
2	2	placebo
3	1	placebo
4	1	placebo
5	4	placebo
6	5	low_dose
7	2	low_dose
8	4	low_dose
9	2	low_dose
10	3	low_dose
11	7	high_dose
12	4	high_dose
13	5	high_dose
14	3	high_dose
15	6	high_dose

Note the means for the three groups are:

```
viagra %>% group_by(dose) %>% summarise(group_mean = mean(libido))
```

```
# A tibble: 3 x 2
  dose      group_mean
  <fct>      <dbl>
1 placebo      2.2
2 low_dose     3.2
3 high_dose     5
```

1.2.2 Set Factor with Reference Group

```
viagra <- viagra %>%  
  mutate(dose = as_factor(dose)) %>%  
  mutate(dose = relevel(dose, ref = "placebo"))
```

1.2.3 Regression with Treatment Contrast

The computer will always use contrasts when there are categorical variables. So you should set the contrast you want. Here we set the contrast as Treatment (or Dummy) Coding. We use treatment contrasts when we are interested in directly interpreting the regression results.

```
options(contrasts = c("contr.treatment", "contr.poly"))  
  
lm_viola <- lm(libido ~ dose + 1, data = viagra)  
  
tidy(lm_viola)
```

term	estimate	std.error	statistic	p.value
(Intercept)	2.2	0.6271629	3.507860	0.0043189
doselow__dose	1.0	0.8869423	1.127469	0.2815839
dosehigh__dose	2.8	0.8869423	3.156913	0.0082681

What is going on here? The single dose column has disappeared. Instead we get *b*-weights for doselow__dose and dosehigh__dose. How do you interpret that information?

1.2.4 Treatment Contrasts Explained

When we used the treatment contrast we converted the use the rules below:

```
contr.treatment(3)
```

```
  2 3  
1 0 0  
2 1 0  
3 0 1
```

- The first contrast row (0 0) indicates the mean of group 1 (placebo) is the intercept plus the 0 times first slope and 0 times the second slope.

$$placebo - mean = 2.20 + 0(1.00) + 0(2.80) = 2.20$$

- The second contrast row (1 0) indicates the mean of group 2 (low dose) is the intercept plus the 1 times first slope and 0 times the second slope.

$$low - dose - mean = 2.20 + 1(1.00) + 0(2.80) = 3.20$$

- The third contrast row (0 1) indicates the mean of group 3 (high dose) is the intercept plus the 0 times first slope and 1 times the second slope.

$$low - dose - mean = 2.20 + 0(1.00) + 1(2.80) = 5.00$$

Recall the levels of the dose variable:

```
levels(viagra$dose)
```

```
[1] "placebo"    "low_dose"    "high_dose"
```

We set the order of the levels previously with the `fct_relevel()` command.

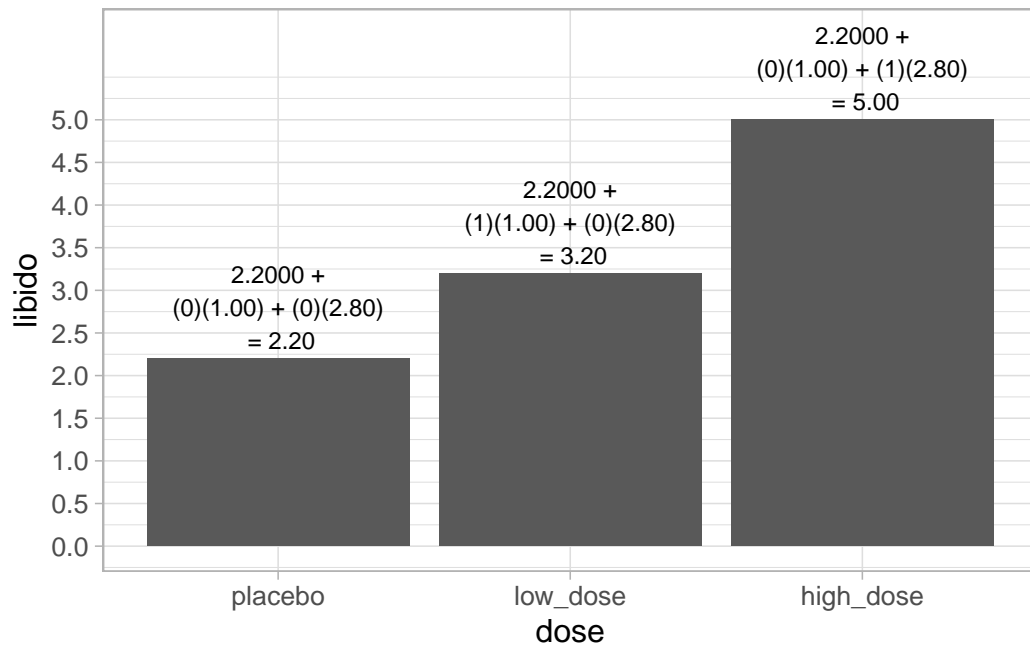
When we ran the regression we effectively use the predictors below

```
print(viagra_dummy_coded)
```

```
# A tibble: 15 x 4
  libido intercept dose_low_dose dose_high_dose
  <int>      <dbl>      <dbl>      <dbl>
1     3         1         0         0
2     2         1         0         0
3     1         1         0         0
4     1         1         0         0
5     4         1         0         0
6     5         1         1         0
7     2         1         1         0
8     4         1         1         0
9     2         1         1         0
10    3         1         1         0
```


11	7	1	0	1
12	4	1	0	1
13	5	1	0	1
14	3	1	0	1
15	6	1	0	1

Examine the weights in the above table and see how they can be used to recreate the group means.



1.2.5 ANOVA Summary Information

With a one-way ANOVA, it's easy to extract ANOVA information from the Regression output.

```
glance(lm_viagra)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.460365	0.3704268	1.402375	5.118644	0.02469432	2	-24.68305	57.36616	60.1983	23.6	12	15

From this output you can see that for this one-way ANOVA, $F(2, 12) = 5.119$, $p = .025$. In a one-way ANOVA the effect size is $\eta^2 = \eta^2_{\text{partial}} = R^2 = .46$. Note that in a one-way ANOVA, $\eta^2 = \eta^2_{\text{partial}}$ but this is not the case when you move to N-way ANOVA.

```
summary(lm_viagra)
```

Call:

```
lm(formula = libido ~ dose + 1, data = viagra)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.0	-1.2	-0.2	0.9	2.0

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.2000	0.6272	3.508	0.00432 **
doselow_dose	1.0000	0.8869	1.127	0.28158
dosehigh_dose	2.8000	0.8869	3.157	0.00827 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.402 on 12 degrees of freedom

Multiple R-squared: 0.4604, Adjusted R-squared: 0.3704

F-statistic: 5.119 on 2 and 12 DF, p-value: 0.02469

From this output you can (AGAIN) see that for this one-way ANOVA, $F(2, 12) = 5.119$, $p = .025$. In a one-way ANOVA the effect size is $\eta^2 = \eta^2_{\text{partial}} = R^2 = .46$.

Note: A good exam question would be to present a table like this and then ask you the mean for each group. With treatment/dummy coding the regression weight indicate for the reference group the mean of that group. For the other groups, the regression weights indicate the difference from the reference group.

1.3 Sum Contrast / Effect Contrast

We typically use Sum Coding or Effect Coding when we are not interested in the directly interpreting the regression results. Although the results can be directly interpreted this is not commonly done. Instead we use Sum Coding or Effect Coding when we use regression to run an ANOVA. Here we will directly interpret the results, however, to show that it can be done.

With sum coding, the contrasts create b -weight represent comparisons of each group mean to the to the grand mean. The contrasts we use for each group are presented below:

```
contr.sum(3)
```

```
  [,1] [,2]
1     1     0
2     0     1
3    -1    -1
```

We run the regression with the code below. Notice we take care to set the contrast to sum before the regression.

```
# Check levels
levels(viagra$dose)
```

```
[1] "placebo" "low_dose" "high_dose"
```

```
# select sum coding
options(contrasts = c("contr.sum", "contr.poly"))

lm_viagra_sum_coded <- lm(libido ~ dose + 1,
                          data = viagra)
```

We see the results are below. We use `as.data.frame()` just to see all the decimals.

```
tidy(lm_viagra_sum_coded) %>% as.data.frame()
```

	term	estimate	std.error	statistic	p.value
1	(Intercept)	3.466667	0.3620927	9.5739760	5.720565e-07
2	dose1	-1.266667	0.5120764	-2.4735893	2.930022e-02
3	dose2	-0.266667	0.5120764	-0.5207556	6.120112e-01

In the results above the intercept corresponds to the grand mean.

Recall the sum contrast below:

```
contr.sum(3)
```

	[,1]	[,2]
1	1	0
2	0	1
3	-1	-1

- The first contrast row (1 0) indicates the mean of group 1 (placebo) is the intercept plus the 1 times first slope and 0 times the second slope.

$$placebo - mean = 3.4666667 + 1(-1.2666667) + 0(-0.2666667) = 2.20$$

- The second contrast row (0 1) indicates the mean of group 2 (low dose) is the intercept plus the 0 times first slope and 1 times the second slope.

$$low - dose - mean = 3.4666667 + 0(-1.2666667) + 1(-0.2666667) = 3.20$$

- The third contrast row (-1 -1) indicates the mean of group 3 (high dose) is the intercept plus -1 times first slope and -1 times the second slope.

$$low - dose - mean = 3.4666667 + (-1)(-1.2666667) + (-1)(-0.2666667) = 5.00$$

1.3.1 Behind the scenes

What is the grand mean? It's just the mean of the dependent variable column across all conditions.

```
summary_stat = viagra %>%
  summarise(grand_mean = mean(libido))

print(summary_stat)
```

```
grand_mean
1    3.466667
```

In the above analysis, when we used this code block:

```
options(contrasts = c("contr.sum", "contr.poly"))

lm_viagra_sum_coded <- lm(libido ~ dose + 1,
  data = viagra)
```

We were doing a regression with the predictors below:

```
print(viagra_sum_coded)
```

	libido	intercept	dose1	dose2
1	3	1	1	0
2	2	1	1	0
3	1	1	1	0
4	1	1	1	0
5	4	1	1	0
6	5	1	0	1
7	2	1	0	1
8	4	1	0	1
9	2	1	0	1
10	3	1	0	1
11	7	1	-1	-1
12	4	1	-1	-1
13	5	1	-1	-1
14	3	1	-1	-1
15	6	1	-1	-1

That is, when we specified this:

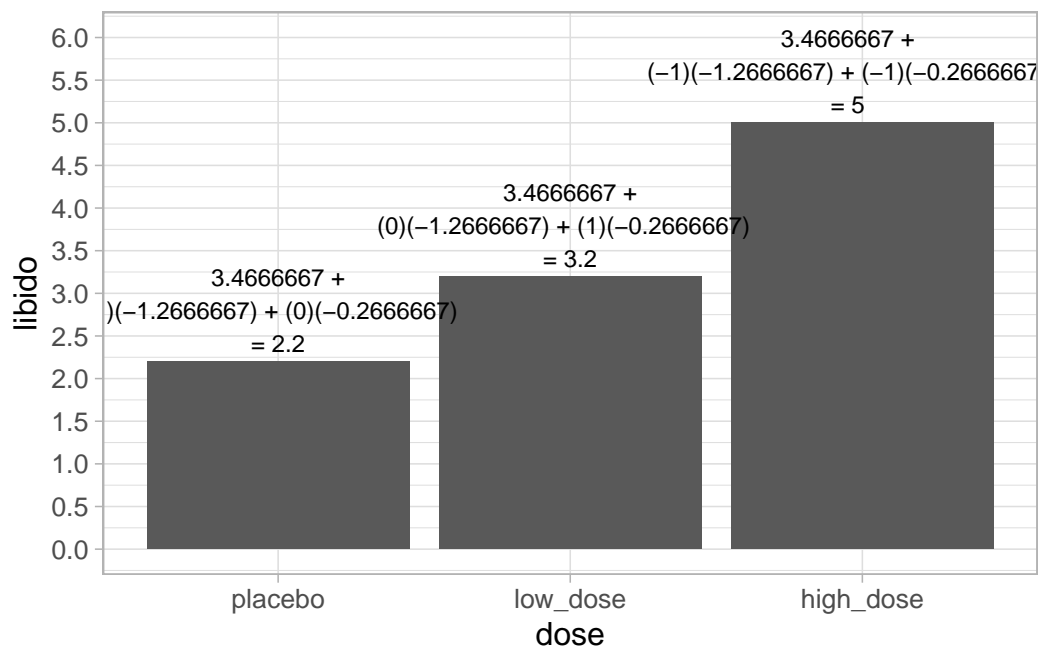
```
lm_viagra <- lm(libido ~ dose + 1, data = viagra)
```

The computer actually ran this:

```
lm_viagra_sum <- lm(libido ~ dose1 + dose2 + 1,  
                    data = viagra_sum_coded)
```

```
tidy(lm_viagra_sum)
```

term	estimate	std.error	statistic	p.value
(Intercept)	3.466667	0.3620927	9.5739760	0.0000006
dose1	-1.266667	0.5120764	-2.4735893	0.0293002
dose2	-0.266667	0.5120764	-0.5207556	0.6120112



1.3.2 ANOVA values

```
glance(lm_viagra_sum)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	hobs
0.4603659	0.3704268	1.4023795	5.1186440	0.0246943	2	-24.68305	57.366160	60.1983	23.6	12	15

From this output you can see that for this one-way ANOVA, $F(2,12) = 5.118644$, $p = 0.0246943$.

```
summary(lm_viagra_sum)
```

Call:

```
lm(formula = libido ~ dose1 + dose2 + 1, data = viagra_sum_coded)
```

Residuals:

```
Min      1Q  Median      3Q      Max
```

```

-2.0   -1.2   -0.2    0.9    2.0

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.4667     0.3621   9.574 5.72e-07 ***
dose1         -1.2667     0.5121  -2.474  0.0293 *
dose2         -0.2667     0.5121  -0.521  0.6120
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.402 on 12 degrees of freedom
Multiple R-squared:  0.4604,    Adjusted R-squared:  0.3704
F-statistic: 5.119 on 2 and 12 DF,  p-value: 0.02469

```

From this output you can (again) see that for this one-way ANOVA, $F(2,12) = 5.118644$, $p = 0.0246943$.

Note: A good exam question would be to present a table like this and then ask you the mean for each group. With treatment/dummy coding the regression weight indicate for the reference group the mean of that group. For the other groups, the regression weights indicate the difference from the reference group.

1.4 Helmert Contrast

Use the Helmert Contrast if you are interested in typical ANOVA results (main effect, main effect, interaction, etc.).

1.5 Summary: Contrast Types

Name/Synonym	R command Example	Nature of Comparison	Note
Treatment Contrast / Dummy Contrast	<pre>> contr.treatment(4) 2 3 4 1 0 0 0 2 1 0 0 3 0 1 0 4 0 0 1</pre>	<p>Intercept is the reference group mean.</p> <p>Unstandardized weights indicate difference between a group's mean and reference group mean. In this example, group 1 is the reference group. Notice the first contrast column is for the second group - because group 1 is the intercept.</p>	Default in R. Do NOT use for ANOVA.
Sum Contrast / Effect Contrast	<pre>> contr.sum(4) [,1] [,2] [,3] 1 1 0 0 2 0 1 0 3 0 0 1 4 -1 -1 -1</pre>	<p>Intercept is the grand mean.</p> <p>Unstandardized weights indicate difference between a group's mean and last group's mean (Group 4 here).</p>	USE for ANOVA. This works because the intercept is the grand mean.

Name/Synonym	R command Example	Nature of Comparison	Note
Helmert Contrast	<pre> > contr.helmert(4) [,1] [,2] [,3] 1 -1 -1 -1 2 1 -1 -1 3 0 2 -1 4 0 0 3 </pre>	<p>Intercept is the grand mean. First contrast, unstandardized weight indicate difference between Group 2 and Group 1 means. Second contrast, unstandardized weight indicate difference between Group 3 mean and the average of the Group 1 and Group 2 means. Third contrast, unstandardized weight indicate difference between Group 4 mean and the average of the Group 1, Group 2, and Group 3 means. And so on.</p>	<p>USE for ANOVA. This works because the intercept is the grand mean.</p>

2 Two-way ANOVA via Regression

On this page we illustrate what is happening “under the hood” when you run an ANOVA using regression.

2.1 Conducting a 2-way ANOVA

2.1.1 Activate Packages

```
library(tidyverse)
library(janitor)
library(pracma)
library(recipes)
library(forcats)
library(tidymodels)
library(apaTables)
```

2.1.2 Load Data

```
gdata = read_csv("gdata.csv")
```

2.1.3 Inspect Data

This example is from the Andy Field book. The example uses alcohol(0, 2 pints, 4 pints) and sex(male,female) as predictors of attractiveness. See the Discovering Statistics Using R (2012) for the complete example. We use this example to illustrate sum contrasts in an ANOVA context.

```
glimpse(gdata)
```

```

Rows: 41
Columns: 3
$ attractiveness <dbl> 60, 60, 55, 60, 55, 70, 65, 60, 70, 65, 60, 60, 50, 55,~
$ gender          <chr> "female", "female", "female", "female", "female", "fema~
$ alcohol         <chr> "none", "none", "none", "none", "none", "pint2", "pint2~

```

```
print(gdata)
```

	attractiveness	gender	alcohol
1	60	female	none
2	60	female	none
3	55	female	none
4	60	female	none
5	55	female	none
6	70	female	pint2
7	65	female	pint2
8	60	female	pint2
9	70	female	pint2
10	65	female	pint2
11	60	female	pint2
12	60	female	pint2
13	50	female	pint2
14	55	female	pint4
15	65	female	pint4
16	70	female	pint4
17	55	female	pint4
18	55	female	pint4
19	60	female	pint4
20	50	female	pint4
21	50	female	pint4
22	50	male	none
23	55	male	none
24	80	male	none
25	65	male	none
26	70	male	none
27	75	male	none
28	75	male	none
29	65	male	none
30	45	male	pint2
31	60	male	pint2
32	85	male	pint2

```

33          65   male   pint2
34          70   male   pint2
35          70   male   pint2
36          80   male   pint2
37          60   male   pint2
38          30   male   pint4
39          30   male   pint4
40          30   male   pint4
41          55   male   pint4

```

2.1.4 Make Factors

```

gdata <- gdata %>%
  mutate(gender = as_factor(gender)) %>%
  mutate(alcobol = as_factor(alcobol))

glimpse(gdata)

```

```

Rows: 41
Columns: 3
$ attractiveness <dbl> 60, 60, 55, 60, 55, 70, 65, 60, 70, 65, 60, 60, 50, 55,~
$ gender          <fct> female, female, female, female, female, female, female,~
$ alcobol         <fct> none, none, none, none, none, pint2, pint2, pint2, pint~

```

2.1.5 Linear Model

Because we want to run an ANOVA we use a sum contrast.

```

options(contrasts = c("contr.sum", "contr.poly"))

lm_output <- lm(attractiveness ~ gender*alcobol,
  data = gdata)

```

But note that when we run the command above with `gender*alcobol` we need to realize is is a shortcut convention for the code below - which implicitly includes an intercept:

```

lm_output <- lm(attractiveness ~ gender+ alcobol + gender:alcobol,
  data = gdata)

```

Which is in turn a shortcut for the code below which explicitly includes the intercept:

```
lm_output <- lm(attractiveness ~ gender+ alcohol + gender:alcohol + 1,
               data = gdata)
```

```
table1 <- apa.aov.table(lm_output, table.number = 1)
apa.save("table1aov.doc", table1)
```

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[-.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[-.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[-.11, .48]
Error	2992.50	35	85.50				

The table above is helpful for interpreting the data, but, how did we obtain it? There is a lot that happens to get the above table. Especially when you remember that if we look at the results of the regression itself it looks quite different:

```
tidy(lm_output)
```

```
# A tibble: 6 x 5
```

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	58	1.50	38.6	2.76e-30
2	gender1	1.33	1.50	0.888	3.81e- 1
3	alcohol1	4.44	2.14	2.08	4.54e- 2
4	alcohol2	6.69	2.01	3.33	2.07e- 3
5	gender1:alcohol1	-5.77	2.14	-2.70	1.06e- 2
6	gender1:alcohol2	-3.52	2.01	-1.75	8.85e- 2

2.2 Regression Becoming ANOVA

A bit of magic seems to happen in the above. We conduct a regression and then somehow get an ANOVA table out at the end. How does that work? The key is understanding that when we specify the regression with factors in it - we are really giving the computer a set of instructions and a starting point - rather than an actual analysis. The computer does a few things “under the hood”:

- 1) Factors are turned into contrast columns
- 2) Multiple columns may be required for a single factor
- 3) The number of contrast columns required for a single factor column is equal to the number of levels minus one.

- 4) The “actual analysis” is conducted using the contrast columns - not the factor column from your data set.

2.3 Contrasts for Categorical Variables

Let’s load a new data set that has some contrast columns created already.

```
gdata <- read_csv("gdata_contrasts.csv")
```

2.3.1 Gender Contrasts

In R when you use the line:

```
options(contrasts = c("contr.sum", "contr.poly"))
```

It effectively runs the `contr.sum()` command on each factor column, when a regression is run, and creates contrasts based on the number of levels of each factor. For example, the sex factor, with levels, causes the command below to be run.

```
contr.sum(2)
```

```
[,1]  
1    1  
2   -1
```

These rules are applied to the gender column. We create a new column called sex where this rule has been applied. In the output below, I have already applied this rule and put the result in the sex column. Normally this happens “under the hood” and you don’t see it. Notice how every female is coded 1 in the sex column where as males are coded -1 in the sex column; consistent with the `contr.sum(2)` command.

```
gdata %>%  
  select(attractiveness, gender, sex) %>%  
  as.data.frame()
```

```
  attractiveness gender sex  
1             60 female  1  
2             60 female  1
```

3	55 female	1
4	60 female	1
5	55 female	1
6	70 female	1
7	65 female	1
8	60 female	1
9	70 female	1
10	65 female	1
11	60 female	1
12	60 female	1
13	50 female	1
14	55 female	1
15	65 female	1
16	70 female	1
17	55 female	1
18	55 female	1
19	60 female	1
20	50 female	1
21	50 female	1
22	50 male	-1
23	55 male	-1
24	80 male	-1
25	65 male	-1
26	70 male	-1
27	75 male	-1
28	75 male	-1
29	65 male	-1
30	45 male	-1
31	60 male	-1
32	85 male	-1
33	65 male	-1
34	70 male	-1
35	70 male	-1
36	80 male	-1
37	60 male	-1
38	30 male	-1
39	30 male	-1
40	30 male	-1
41	55 male	-1

2.3.2 Alcohol Contrasts

```
contr.sum(3)
```

```
  [,1] [,2]
1     1   0
2     0   1
3    -1  -1
```

In the output below, I have already applied this rule and put the result in the alc1 and alc2 columns. Normally this happens “under the hood” and you don’t see it. Notice how every levels of alcohol are coded using this scheme; consistent with the `contr.sum(3)` command.

```
gdata %>%
  select(attractiveness, alcohol, alc1, alc2) %>%
  as.data.frame()
```

	attractiveness	alcohol	alc1	alc2
1	60	none	1	0
2	60	none	1	0
3	55	none	1	0
4	60	none	1	0
5	55	none	1	0
6	70	pint2	0	1
7	65	pint2	0	1
8	60	pint2	0	1
9	70	pint2	0	1
10	65	pint2	0	1
11	60	pint2	0	1
12	60	pint2	0	1
13	50	pint2	0	1
14	55	pint4	-1	-1
15	65	pint4	-1	-1
16	70	pint4	-1	-1
17	55	pint4	-1	-1
18	55	pint4	-1	-1
19	60	pint4	-1	-1
20	50	pint4	-1	-1
21	50	pint4	-1	-1
22	50	none	1	0
23	55	none	1	0

24	80	none	1	0
25	65	none	1	0
26	70	none	1	0
27	75	none	1	0
28	75	none	1	0
29	65	none	1	0
30	45	pint2	0	1
31	60	pint2	0	1
32	85	pint2	0	1
33	65	pint2	0	1
34	70	pint2	0	1
35	70	pint2	0	1
36	80	pint2	0	1
37	60	pint2	0	1
38	30	pint4	-1	-1
39	30	pint4	-1	-1
40	30	pint4	-1	-1
41	55	pint4	-1	-1

2.3.3 Interaction Contrasts

We also need contrasts for the interaction. We create the interaction contrasts by multiplying the columns for sex, alc1, and alc2. You can see how we do so in the code below.

```
gdata <- gdata %>%
  mutate(int1 = sex*alc1,
         int2 = sex*alc2)
```

You can see these new interaction columns below:

```
print(gdata)
```

	attractiveness	sex	alc1	alc2	int1	int2
1	60	1	1	0	1	0
2	60	1	1	0	1	0
3	55	1	1	0	1	0
4	60	1	1	0	1	0
5	55	1	1	0	1	0
6	70	1	0	1	0	1
7	65	1	0	1	0	1
8	60	1	0	1	0	1

9	70	1	0	1	0	1
10	65	1	0	1	0	1
11	60	1	0	1	0	1
12	60	1	0	1	0	1
13	50	1	0	1	0	1
14	55	1	-1	-1	-1	-1
15	65	1	-1	-1	-1	-1
16	70	1	-1	-1	-1	-1
17	55	1	-1	-1	-1	-1
18	55	1	-1	-1	-1	-1
19	60	1	-1	-1	-1	-1
20	50	1	-1	-1	-1	-1
21	50	1	-1	-1	-1	-1
22	50	-1	1	0	-1	0
23	55	-1	1	0	-1	0
24	80	-1	1	0	-1	0
25	65	-1	1	0	-1	0
26	70	-1	1	0	-1	0
27	75	-1	1	0	-1	0
28	75	-1	1	0	-1	0
29	65	-1	1	0	-1	0
30	45	-1	0	1	0	-1
31	60	-1	0	1	0	-1
32	85	-1	0	1	0	-1
33	65	-1	0	1	0	-1
34	70	-1	0	1	0	-1
35	70	-1	0	1	0	-1
36	80	-1	0	1	0	-1
37	60	-1	0	1	0	-1
38	30	-1	-1	-1	1	1
39	30	-1	-1	-1	1	1
40	30	-1	-1	-1	1	1
41	55	-1	-1	-1	1	1

2.4 Regression command (i.e., lm) overview

To get ANOVA results that are consistent with what are typically used in psychology you need to 1) Specify the `contr.sum()` contrast 2) Calculate the Sum of Squares using the logic for Type III Sum of Squares

When you run an ANOVA using the command:

```
options(contrasts = c("contr.sum", "contr.poly"))

lm_output <- lm(attractiveness ~ gender + alcohol + gender:alcohol,
               data = gdata)
```

You actually run the regression below:

```
lm_output <- lm(attractiveness ~ gender1 + alcohol1 + alcohol2 + gender:alcohol1 + gender:
               data = gdata)
```

Which uses these columns - notice there is a column of all ones for the intercept:

```
print(gdata)
```

	attractiveness	intercept	sex	alc1	alc2	int1	int2
1	60	1	1	1	0	1	0
2	60	1	1	1	0	1	0
3	55	1	1	1	0	1	0
4	60	1	1	1	0	1	0
5	55	1	1	1	0	1	0
6	70	1	1	0	1	0	1
7	65	1	1	0	1	0	1
8	60	1	1	0	1	0	1
9	70	1	1	0	1	0	1
10	65	1	1	0	1	0	1
11	60	1	1	0	1	0	1
12	60	1	1	0	1	0	1
13	50	1	1	0	1	0	1
14	55	1	1	-1	-1	-1	-1
15	65	1	1	-1	-1	-1	-1
16	70	1	1	-1	-1	-1	-1
17	55	1	1	-1	-1	-1	-1
18	55	1	1	-1	-1	-1	-1
19	60	1	1	-1	-1	-1	-1
20	50	1	1	-1	-1	-1	-1
21	50	1	1	-1	-1	-1	-1
22	50	1	-1	1	0	-1	0
23	55	1	-1	1	0	-1	0
24	80	1	-1	1	0	-1	0
25	65	1	-1	1	0	-1	0
26	70	1	-1	1	0	-1	0
27	75	1	-1	1	0	-1	0

28	75	1	-1	1	0	-1	0
29	65	1	-1	1	0	-1	0
30	45	1	-1	0	1	0	-1
31	60	1	-1	0	1	0	-1
32	85	1	-1	0	1	0	-1
33	65	1	-1	0	1	0	-1
34	70	1	-1	0	1	0	-1
35	70	1	-1	0	1	0	-1
36	80	1	-1	0	1	0	-1
37	60	1	-1	0	1	0	-1
38	30	1	-1	-1	-1	1	1
39	30	1	-1	-1	-1	1	1
40	30	1	-1	-1	-1	1	1
41	55	1	-1	-1	-1	1	1

2.5 Full and Restricted Models

R actually runs a whole series of regressions for you and combines them into the single output table you saw above. These models fall into two categories Full and Restricted Models.

2.5.1 Full Model

First, the Full Model is run that includes all of the predictor columns:

```
lm_full <- lm(attractiveness ~ sex + alc1 + alc2 + int1 + int2,
              data = gdata)
```

2.5.2 Restricted Models

Next a series of restricted models are run that excluded an effect of interest for each restricted model.

```
lm_restricted_no_sex <- lm(attractiveness ~ alc1 + alc2 + int1 + int2,
                           data = gdata)

lm_restricted_no_alcohol <- lm(attractiveness ~ sex + int1 + int2,
                               data = gdata)

lm_restricted_no_interaction <- lm(attractiveness ~ sex + alc1 + alc2,
```

```

data = gdata)

lm_restricted_no_intercept <- lm(attractiveness ~ sex + alc1 + alc2 + int1 + int2 - 1,
data = gdata)

```

2.6 Logic: Model Comparison

The ANOVA table is created by comparing each of these restricted models to the full model. For example, to determine the main effect for gender we compare the model `lm_restricted_no_sex` to the model `lm_full`. If `lm_full` accounts for substantially more variance than `lm_restricted_no_sex` it is significant.

This is effectively identical to when we looked at comparing two regression models previous. Using that logic, we could just write the code:

```

# try this, it works!
library(apaTables)
apa.reg.table(lm_restricted_no_sex, lm_restricted_all)

```

The result would tell us if the main effect of sex is significant. The logic of calculating things this way is the Type III Sum of Squares logic. HOWEVER, we don't tend to use the output in the form provided in this type of table. Rather the output is reformatted to be consistent with the way ANOVA's are typically presented. The next few sections show you how the table below is created:

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[-.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[-.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[-.11, .48]
Error	2992.50	35	85.50				

2.7 Explanation 1: Comparing Models

2.7.1 Sex

```

lm_restricted_no_sex <- lm(attractiveness ~ alc1 + alc2 + int1 + int2,
data = gdata)

lm_full <- lm(attractiveness ~ sex + alc1 + alc2 + int1 + int2,

```

```
data = gdata)

anova(lm_restricted_no_sex, lm_full)
```

Analysis of Variance Table

```
Model 1: attractiveness ~ alc1 + alc2 + int1 + int2
Model 2: attractiveness ~ sex + alc1 + alc2 + int1 + int2
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1      36 3059.9
2      35 2992.5  1    67.368 0.7879 0.3808
```

Compare the F – and p –values in the above output to those in the table below.

Predictor	SS	df	MS	F	p	$\eta^2_{partial}$	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[-.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[-.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[-.11, .48]
Error	2992.50	35	85.50				

2.7.2 Alcohol

```
lm_restricted_no_alcohol <- lm(attractiveness ~ sex + int1 + int2,
                               data = gdata)

lm_full <- lm(attractiveness ~ sex + alc1 + alc2 + int1 + int2,
              data = gdata)

anova(lm_restricted_no_alcohol, lm_full)
```

Analysis of Variance Table

```
Model 1: attractiveness ~ sex + int1 + int2
Model 2: attractiveness ~ sex + alc1 + alc2 + int1 + int2
  Res.Df    RSS Df Sum of Sq    F    Pr(>F)
1      37 5223.3
2      35 2992.5  2    2230.8 13.045 5.843e-05 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Compare the F – and p -values in the above output to those in the table below.

Predictor	SS	df	MS	F	p	$\eta^2_{partial}$	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.7.3 Interaction

```
lm_restricted_no_interaction <- lm(attractiveness ~ sex + alc1 + alc2,
                                   data = gdata)

lm_full <- lm(attractiveness ~ sex + alc1 + alc2 + int1 + int2,
              data = gdata)

anova(lm_restricted_no_interaction, lm_full)
```

Analysis of Variance Table

Model 1: attractiveness ~ sex + alc1 + alc2

Model 2: attractiveness ~ sex + alc1 + alc2 + int1 + int2

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	37	4501.2				
2	35	2992.5	2	1508.7	8.8225	0.0007896 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Compare the F – and p -values in the above output to those in the table below.

Predictor	SS	df	MS	F	p	$\eta^2_{partial}$	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.8 Degrees of Freedom

When you look at the the columns in the above output notice the number of columns we use for each predictor corresponds the degrees of freedom for that predictor.

Predictor	df	Number of contrast columns	Contrast column names
sex	$df_a = a - 1 = 2 - 1 = 1$	1	sex
alcohol	$df_b = b - 1 = 3 - 1 = 2$	2	alc1, alch2
sex by alcohol	$df_{int} = df_a * df_b = (a - 1)(b - 1) = 1(2) = 2$	2	int1, int2

2.9 Explanation 2: Sum of Squares to ANOVA

Notice the first column of the ANOVA table above is the Sum of Squares column. How are those values calculated? Let's consider the example of alcohol as a predictor. We want to determine the Sum of Squares for alcohol.

We begin by calculating predicted scores for the Full Model (lm_full):

$$\hat{y}_{\text{full}} = b_0 + b_1 \text{sex} + b_2 \text{alc1} + b_3 \text{alc2} + b_4 \text{int1} + b_5 \text{int2}$$

Next, we calculate the predicted scores for the alcohol restricted model (lm_restricted_no_alcohol)

$$\hat{y}_{\text{restricted no alcohol}} = b_0 + b_1 \text{sex} + b_2 \text{int1} + b_3 \text{int2}$$

Then we calculated the difference between these two sets of predicted scores:

$$\hat{y}_{\text{difference}} = \hat{y}_{\text{full}} - \hat{y}_{\text{restricted no alcohol}}$$

Then we square these values and add them up.

$$SS_{\text{alcohol}} = \sum \hat{y}_{\text{difference}}^2$$

We follow this process below for each predictor (including the intercept).

2.9.1 Intercept

```
## Sum of squares intercept  
sum( ( predict(lm_full) - predict(lm_restricted_no_intercept) )^2 )
```

```
[1] 127477.9
```

Compare the Sum of Squares values in the above output to the one in the table below.

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.9.2 Sex

```
## Sum of squares sex  
sum( ( predict(lm_full) - predict(lm_restricted_no_sex) )^2 )
```

```
[1] 67.36842
```

Compare the Sum of Squares values in the above output to the one in the table below.

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.9.3 Alcohol

```
## Sum of squares alcohol  
sum( ( predict(lm_full) - predict(lm_restricted_no_alcohol) )^2 )
```

```
[1] 2230.757
```

Compare the Sum of Squares values in the above output to the one in the table below.

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.9.4 Interaction

```
## Sum of squares interaction
sum( ( predict(lm_full) - predict(lm_restricted_no_interaction) )^2 )
```

```
[1] 1508.651
```

Compare the Sum of Squares values in the above output to the one in the table below.

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.9.5 Error

```
## Sum of squares error
sum( lm_full$residuals^2 )
```

```
[1] 2992.5
```

Compare the Sum of Squares values in the above output to the one in the table below.

Predictor	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>p</i>	η^2_{partial}	90% CI
(Intercept)	127477.89	1	127477.89	1490.97	<.001		
gender	67.37	1	67.37	0.79	.381	.02	[.00, .15]
alcohol	2230.76	2	1115.38	13.05	<.001	.43	[.19, .56]
gender x alcohol	1508.65	2	754.33	8.82	.001	.34	[.11, .48]
Error	2992.50	35	85.50				

2.9.6 SS to ANOVA

Based on the above analyses we know the Sum of Squares and the degrees of freedom for everything. We can put that information in the table below.

Predictor	SS	df	$MS = \frac{SS}{df}$	$F = \frac{MS}{MS_{error}}$	p
(Intercept)	127477.89	1			
sex	67.37	1			
alcohol	2230.76	2			
sex by alcohol	1508.65	2			
Error	2992.5	35			

A few hand calculations, and an F to p -value look-up table, provides us with the rest of the information we need:

Predictor	SS	df	$MS = \frac{SS}{df}$	$F = \frac{MS}{MS_{error}}$	p
(Intercept)	127477.89	1	$\frac{127477.89}{1} = 127477.89$	$\frac{127477.89}{85.5} = 1490.97$	<.001
sex	67.37	1	$\frac{67.37}{1} = 67.37$	$\frac{67.37}{85.5} = 0.79$.381
alcohol	2230.76	2	$\frac{2230.76}{2} = 1115.38$	$\frac{1115.38}{85.5} = 13.05$	<.001
sex by alcohol	1508.65	2	$\frac{1508.65}{2} = 754.325$	$\frac{754.325}{85.5} = 8.82$.001
Error	2992.5	35	$\frac{2992.5}{35} = 85.5$		

References