Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

An R Tutorial for the Non-Coding-Inclined

As empirical, quantitative methods continue to become more commonplace in linguistic research, linguists are increasingly employing various kinds of experimental tasks, such as grammaticality judgment tasks, word/phrase-list readings, perceptual discrimination tasks, matched guise techniques, sociolinguistic interviews, surveys/questionnaires, pre-/post-tests, and corpus analyses. In order to be able to analyze the distinct kinds of data obtained from these and other tasks, minimally, a passive knowledge of inferential statistics and a degree of proficiency with inferential statistics software are required.

Two principal complications present themselves with respect to these requirements: (1) the set of statistical techniques deemed appropriate for specific types of data is inherently non-static, evolving as advances are made in the statistics discipline, effectively resulting in 'waves' of consensus in the Linguistics community regarding the use of particular statistical tests for linguistic data, and; (2) multiple software options exist and continue to be developed for performing inferential statistics, each with unique interfaces and capabilities that evolve alongside the very tests they were designed to perform. Thus, a linguist's endeavor to gain expertise in experimental methodologies, statistical theory, and statistical software packages is much more akin to chasing a perpetually moving target than mastering a finite set of skills.

One software package in particular, R ([3]), has arguably become the new norm for statistical analysis in the Linguistics community, boasting a free and maximally powerful open-source platform that stands in stark contrast to other competitors (e.g. SAS [4], SPSS [1], STATA [6],) that are only accessible through paid (and often University) subscription. Unfortunately, however, R's minimal user interface begs programming language, constituting a daunting and perhaps unintuitive burden for many linguists. Still, as efforts to combat this reliance on user-generated code remain somewhat restricted to Variationist Sociolinguistics (e.g. Rbrul [2], Language Variation Suite [5]), it seems increasingly likely that the newer generations of empirical linguists will be tasked with becoming proficient in R.

My goal is to all but eliminate R's barrier to entry, affording fuller access to statistical analyses to as wide a community of linguists as possible. Reducing user-generated coding to the absolute minimum, namely the typing out of names of independent and dependent variables of interest, this Copy-Paste guide helps enable users to fully perform R analyses with linguistic data. Accompanying files (a version of this guide as an R file to make copy-pasting easier [i.e., no need to click back and forth between R and your PDF-viewer], as well as example datasets and example data organization templates) are available at the bottom of:

https://spanish-portuguese.berkeley.edu/people/justin-davidson/

Please note: This is NOT a guide for or declaration of 'correct' statistical techniques, which should rightly come from the field of statistics, rather than linguistics. R will perform the commands you give it, even if those commands are statistically unsound!

References

[1] IBM Corp. (2019). IBM SPSS Statistics for Windows/Macintosh. Armonk, NY: IBM Corp. <https://www.ibm.com/analytics/spss-statistics-software>.

[2] Johnson, Daniel Ezra (2009). Getting off the Goldvarb standard: Introducing Rbrul for mixed-effects variable rule analysis. *Language and Linguistics Compass*, 3(1): 359-383.

[3] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

[4] SAS Institute Inc. (2019). SAS software. <https://www.sas.com/en_us/software/platform.html>.

[5] Scrivner, Olga & Manuel Díaz Campos (2016). Language Variation Suite: A theoretical and methodological contribution for social and linguistic data analysis. *Linguistic Society of America*. <https://languagevariationsuite.shinyapps.io/Pages/>.

[6] StataCorp (2017). Stata Statistical Software: Release 15. College Station, TX: StataCorp LLC. <https://www.stata.com/>.

TABLE OF CONTENTS

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

The ONLY Coding Knowledge You’ll Need
(A mini-version of this appears at the top of every test, just as a helpful reminder)

The following **colored** abbreviations are the only code you will need to create/write:

V	- Any variable (independent or dependent), i.e., the name of one of your Excel/spreadsheet columns
IV	- An independent variable, i.e., the name of one of your Excel/spreadsheet columns
DV	- A dependent variable, i.e., the name of one of your Excel/spreadsheet columns
#	- Input a number (such as 2 or 3). Specifics on what kind of number to insert appear beside codes that require this.

IVDUMP - Your syntax for fixed main effects and interactions. IVs are separated by a plus sign and interactions are denoted by an asterisk. Some examples appear below, for a hypothetical experiment with the following 3 IVs: **Gender**, **Age**, and **Verb**

Gender + Age	Tests for main effects of gender and age, one at a time.
Gender * Age	Tests for main effects of gender and age, one at a time, and also tests for their interaction (i.e., does a gender effect depend on age?; Does an age effect depend on gender?)
Gender + Age * Verb	Tests for main effects of each IV, 1 at a time, and also tests for 1 interaction (age/verb).
Verb * Age + Gender	No different from the code immediately above this. You can transpose items freely; it's the signs (+ or *) between them that matter.
Gender * Age + Verb * Age	Tests for main effects of each IV, 1 at a time, as well as 2 interactions (age/gender & age/verb)
Gender * Age + Gender	The "**+ Gender**" is redundant, since **Gender * Age** already tests for each IV separately. Delete "**+ Gender**" and reduce to simply "**Gender * Age**".

RIVDUMP - Your syntax for random intercepts/effects and slopes in a mixed effects model. Random effects/intercepts and slopes appear in enclosed parentheses, such as (1|FirstIntercept) + (1|SecondIntercept), each one separated by a plus sign. Use the "1" when no slope is present, and replace the "1" with a within-subjects IV to include it as a slope for the intercept in question. Examples appear below for an experiment with IVs of Gender, Verb, Word, and Participant:

Gender * Verb + (1|Participant)	Fixed main and interaction effects for Gender/Verb, with Participant as a random intercept/effect.
Gender + Verb + (1|Participant) + (1|Word)	Fixed main effects of Gender and Verb, with each of Participant and Word as random intercepts/effects.
Gender + Verb + (Verb|Participant)	Fixed main effects of Gender and Verb, with Participant as a random intercept/effect that has Verb as a random slope.

ROWDUMP – Only applicable when forcing an ANOVA output for a fixed effects logistic regression. See explanation at the top of the fixed effects logistic regression page if you ever want to do this.

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Tips and Terminology

- ColoredWords like this or this indicate labels that you can choose to rename, keeping labels consistent with colors across a given test. Personally, I recommend not modifying/editing them, since the names I've given them are consistent (and it's one less thing you have to worry about [or potentially mess up]!). Occasionally entire lines of text are highlighted in yellow, which is to elicit careful attention.
- For each analysis you run, quit R and open it up anew. Otherwise, you'll find that it remembers objects like DataName and the like from past sessions, which will be problematic. (I.e., if you ever get an error about a Name being 'masked' from prior analyses, this means R remembers this object from a prior session and you need to quit R and start fresh. Clearing your workspace [clickable under the Workspace menu at the top of the screen] once you open up R again should do the trick, especially for RStudio users.)
- RStudio is an alternative software package to R for those that want a more snazzy view of things. Beyond seeing this cheat sheet (assuming you go look at my version of this tutorial as an R file) in the same window as your command-line window, RStudio also can be set to auto-load-in library packages (i.e., every step 2 for each test), which can let you skip some copy-pasting and begin each analysis at step 3.
- Every test begins with a few command lines of *purple italics,* namely *install.packages("SOMETHING")*. You need to be connected to the Internet to successfully run these lines. Luckily, after you run them once, you do NOT have to run them again unless you re-download R (i.e., you update it, or alternatively you trash R and later decide to come crawling back ;) Accordingly, plan on usually skipping to step 2 (and never needing an Internet connection to run your analyses!).
- Tukey post-hoc tests via *emmeans* generate a warning note in red about misleading results. Don't be alarmed when you see it!

Nominal Variable: A variable whose levels are discrete, non-numerical categories or labels, such as "Young" vs. "Old" for Age, or "Absent" vs. "Present" for Copula Use. Since these are discrete categories, you cannot take a meaningful average/mean of them, i.e., it doesn't make sense to ask "What's the average of "Young" and "Old"? IMPORTANT – In your Excel/spreadsheet file, all nominal variables must have cells that begin with a letter (rather than a number). If you have an IV of Verb, with two levels: Type 1 and Type 2, then be sure your cells are **Type1** and **Type2** **(**or even **T1** and **T2**), rather than **1** and **2**.

Continuous Variable: A variable whose levels are numerical values that can be averaged, such as Formant Frequency (250 vs. 252 vs. 300 etc...) and Age (33 vs. 35 vs. 45 etc...). Note that a single variable can be treated as either nominal or continuous by the researcher, such as Age in the examples here (Continuous Age: 33 vs. 45, etc.; Nominal Age: Young vs. Old [where Young is anybody 18-30 and Old is anybody 45-60]). Be careful with nominal IVs that might be associated with numbers, such as Participant Number: Person 1, Person 2, Person 3, etc. You wouldn't want to say that there exists an average participant number of 2 (via [1+2+3]/3) in an experiment with Person 1, Person 2, and Person 3. IMPORTANT – In your Excel/spreadsheet file, all continuous variables must have cells that are numerical (i.e., no letters!). For the example of Participant Number, which should be a nominal IV, do NOT fill cells as **1**, **2**, and **3**, and instead have them start with a letter (e.g. **p1**, **p2**, **p3**).

Between-subjects IV = a nominal IV for which each participant can only contribute a DV response for 1 level. Social IVs (gender, country of origin, age group, language dominance group, etc.) are commonly between-subjects IVs, since a participant is static during an experiment (i.e., a participant can't be classified as "From Argentina" for some of the experiment and "From Uruguay" for another portion of it. Each participant represents or contributes DV data for only 1 level of each between-subjects IV.)

Within-subjects IV = a nominal IV for which each participant contributes a DV response for all/each of the IV's levels. These are often linguistic IVs, such as Phrase Type, whereby every participant offers a DV response for every type of phrase.

Which Test Do I Use??? (Flowchart-esque instructions limited to tests covered here only)

Question 1: What kind of Dependent Variable do you have?

Possible answer 1: Continuous

- Between-subjects ANOVA (if all IVs are nominal and between-subjects, with NO random effects. **Question 2** does not apply.)
- Within-subjects ANOVA (if all IVs are nominal and 1+ is within-subjects, NO random effects. **Question 2** does not apply.)
- Some type of Linear Regression (no restrictions! Math is identical to between-/within-subjects ANOVAs, so you're covered no matter what!)

Possible answer 2: Nominal, Single-level (i.e., counted occurrences of a single, non-varying type of nominal response)

- Chi-Squared Test (if all IVs are nominal, with NO random effects. This test prevents you from running any IV interactions, and moreover can only be run 1 IV at a time, making it more problematic if you have several IVs. **Question 2** does not apply.)
- Some type of Poisson Regression (no restrictions! However, choice between Poisson vs. Zero-Inflated Poisson is discussed in the Poisson coding section, with the empirical test to motivate one vs. the other appearing as code step [3a] in Model Comparison.)

Possible answer 3: Nominal, two-level (henceforth "binary")

- Some type of Logistic Regression (no restrictions!)

Question 2: Do you have any random effects (i.e., random intercepts or random intercepts and slopes)?

Possible answer 1: No!

- Fixed effects model (i.e., whichever regression you're going to run, choose the FIXED effects version of it.)

Possible answer 2: Yes!

- Mixed effects model (i.e., whichever regression you're going to run, choose the MIXED effects version of it.)

Possible answer 3: I don't know... should I?

- Model Comparison (i.e., head to the Model Comparison section and follow the instructions to run code step [3b] on two models that are identical beyond one being FIXED [lacking random effects] and one being MIXED [having 1+ random effect].)

Question 3: Do you already know the IVs you're including and which ones are or are not tested as interaction terms?

Possible answer 1: Yes!

- Peachy! Go run your stats!

Possible answer 2: No! ... or I thought I did, but now your question is making me second-guess myself!

- Model Comparison (i.e., create 2 [or more] models and then compare them to decide which to run stats as normal on.)
- Step-wise Regression (i.e., run a step-wise regression to first identify the best-fit model and then run stats as normal on it.)

Possible answer 3: I'm a Variationist Sociolinguist that wants to do Variable Rule Analysis (Varbrul) a la Goldvarb!

- Variable Rule Analysis (i.e., run a step-wise logistic regression that outputs factor weights for every nominal IV level) – Fixed effects model
- Variable Rule Analysis (i.e., run a step-wise logistic regression that outputs factor weights for every nominal IV level) – Mixed effects model

Understanding an ANOVA output vs. a (non-Varbrul) Regression output

RQ: Do speakers of different ages and languages produce /a/ (F1) differently (i.e., does age and/or language affect /a/ production)?

Independent Variable #1 - Age: Young vs. Middle vs. Old
Independent Variable #2- Language: Spanish vs. Catalan
Dependent Variable: F1 in hertz

Hypothetical results of a production experiment:

Young Spanish:	800hz	Young Catalan:	400 hz
Middle Spanish:	400hz	Middle Catalan:	800 hz
Old Spanish:	400hz	Old Catalan:	400 hz

An ANOVA output is expressed in terms of individual independent variables being statistically significant or not (i.e., do they or do they not affect the dependent variable). An ANOVA output for the aforementioned results might look like the following:

AGE:	p=.0001 ***
LANGUAGE:	p=.15
AGE*LANGUAGE (interaction):	p=.03 *

From this alone, we'd note that age significantly affects /a/ production, language does not, but there is a significant interaction, so it's possible that the age effect depends on language, or likewise that a language effect depends on age. The post-hoc tests on the significant age variable (since it has 3+ levels) and the significant interaction may look like the following:

Young vs. Middle:	p=.8	Young Spanish vs. Young Catalan:	p=.01 *
Young vs. Old:	p=.02 *	Middle Spanish vs. Middle Catalan:	p=.01 *
Middle vs. Old:	p=.001 **	Old Spanish vs. Old Catalan	p=.79

The AGE post-hoc (on the left) reveals that the Young and Middle speakers do not produce /a/ differently from one another, though each produce /a/ differently than the Old speakers. The INTERACTION post-hoc (on the right) reveals that there actually is a language effect; it simply depends on age. For Old speakers, Spanish and Catalan /a/ are not different. For Young and Middle speakers, however, Spanish /a/ is not produced the same as Catalan /a/. A manual inspection of the language difference for Young speakers vs. Middle speakers (i.e., plot the results or envision a graph of the hertz figures above) shows a unique direction of effect: for Young speakers, Catalan /a/ has a lower F1 than Spanish /a/, but for Middle speakers, Catalan /a/ has a higher F1 than Spanish /a/.

Thus, ANOVA outputs are great for showing which independent variables are significant or not, but they don't explicitly show you different directions of effect. To get these, you need to either plot/visualize the results or use the *tapply* command in R.

A regression output differs from an ANOVA in its presentation (i.e., the actual math/results are IDENTICAL). It contains an independent variable intercept/reference level, which is a combination of the alphabetically-/numerically-first level of every independent variable. The p-value for the intercept (row1 below) indicates whether or not the dependent variable (F1 of /a/) is significantly different from a value of 0. The remaining p-values indicate whether or not the level in question is significantly different from the intercept. The results below could be obtained from a regression on the same dataset:

		Estimate/β coefficient	p-value
Row1	INTERCEPT (Middle Catalan)	+800	p=.0001 ***
Row2	Spanish	-400	p=.003 **
Row3	Young	-400	p=.04 *
Row4	Old	-400	p=.004 **
Row5	Young:Spanish	+800	p=.01 *
Row6	Old:Spanish	+400	p=.02 *

We'll interpret row by row, noting again that this result is IDENTICAL to that of the ANOVA, just presented in a unique format.

Row1: The significant intercept means that the F1 of /a/ for Middle-aged Catalan speakers is significantly different from 0 hertz, specifically 800 hertz greater than 0 (hence the positive coefficient), or 800 hertz.

Row2: From the intercept (Middle Catalan) value of 800 hertz, we now look to a possible significant difference in hertz for Middle Spanish speakers (i.e., only language is shifting, with age remaining unchanged from the intercept). The significant p-value suggests that the F1 for Middle Spanish speakers is 400 hertz lower than the intercept, i.e., 800 – 400 = 400 hertz.

Row3: From the intercept (Middle Catalan) value of 800 hertz, we now look to a possible significant difference in hertz for Young Catalan speakers (i.e., only age is shifting, with language remaining unchanged from the intercept). The significant p-value suggests that the F1 for Young Catalan speakers is 400 hertz lower than the intercept, i.e., 800 – 400 = 400 hertz.

Row4: From the intercept (Middle Catalan) value of 800 hertz, we now look to a possible significant difference in hertz for Old Catalan speakers (i.e., only age is shifting, with language remaining unchanged from the intercept). The significant p-value suggests that the F1 for Old Catalan speakers group is 400 hertz lower than the intercept, i.e., 800 – 400 = 400 hertz.

Row5: Calculating the F1 production for Young Spanish speakers, we start from the intercept of 800 and subtract 400 (as row 2 is significant) and then subtract another 400 (as row 3 is significant), yielding 800-400-400 = 0 hertz. At this point, i.e., for Young Spanish speakers, because row5 is significant, we now add 800 (effectively nullifying the prior main effects of age and language) to come to a predicted hertz production for Young Spanish speakers of 800 hertz (or 800-400-400+800).

Row6: Calculating the F1 production for Old Spanish speakers, we start from the intercept of 800 and subtract 400 (as row 2 is significant) and then subtract another 400 (as row 4 is significant), yielding 800-400-400 = 0 hertz. At this point, i.e., for Old Spanish speakers, because row6 is significant, we now add 400 (effectively nullifying the prior main effect of language) to come to a predicted hertz production for Old Spanish speakers of 400 hertz (or 800-400-400+400).

Accordingly, regression outputs are great for noting directions of effect, but require some mental math in order to fully interpret and additionally aren't as transparent as ANOVAs for individual independent variables' statistical significance.

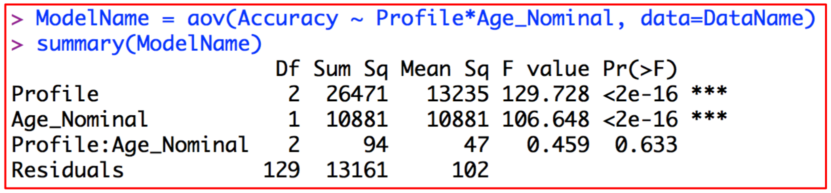
Interpreting and Reporting Results in ANOVA Format

The following example comes from running an ANOVA (via Between-Subjects ANOVA) on the dataset labeled "ContinuousDV_ANOVA_LinearReg" in the Templates and Example Datasets file at the bottom of:

https://spanish-portuguese.berkeley.edu/people/justin-davidson/

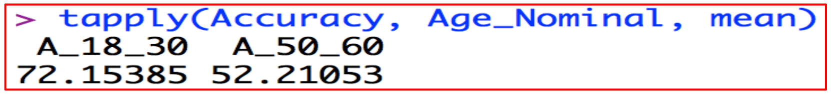
In this hypothetical experiment, we tested for main effects of Profile (Monolingual, Early Bilingual, and Late Bilingual) and Age (18-30 year olds and 50-60 year olds), as well as their interaction, on Accuracy scores.

R Outputs for ANOVA (via Between-Subjects ANOVA)



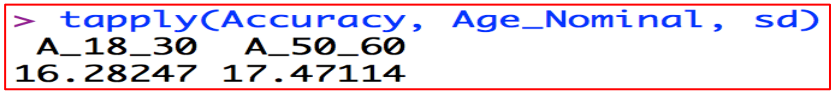
```
> ModelName = aov(Accuracy ~ Profile*Age_Nominal, data=DataName)
> summary(ModelName)
                     Df Sum Sq Mean Sq F value Pr(>F)
Profile               2  26471   13235 129.728 <2e-16 ***
Age_Nominal           1  10881   10881 106.648 <2e-16 ***
Profile:Age_Nominal   2     94      47   0.459  0.633
Residuals           129  13161     102
```

Main effects: Profile/Age = significant (p < .0001 for each)
Interaction effect: not significant (p = 0.633)
Degrees of Freedom: Profile=2, Age=1, Interaction=2, Resid.=129
Test Statistic (F): Profile=129.728, Age=106.648, Interaction=0.459



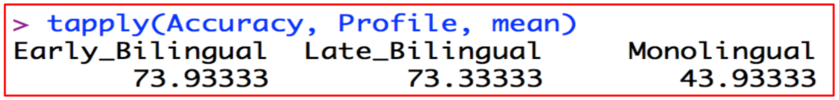
```
> tapply(Accuracy, Age_Nominal, mean)
  A_18_30   A_50_60
72.15385 52.21053
```

Direction of significant main effect: 18-30 year olds have higher accuracy than 50-60 year olds (respectively 72 vs. 52)



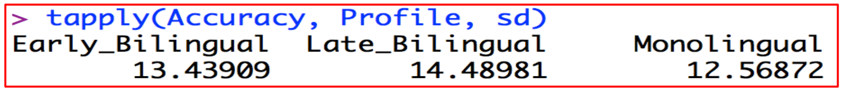
```
> tapply(Accuracy, Age_Nominal, sd)
  A_18_30   A_50_60
16.28247 17.47114
```

Standard Deviation: 18-30 year olds = 16.3 ; 50-60 year olds = 17.5



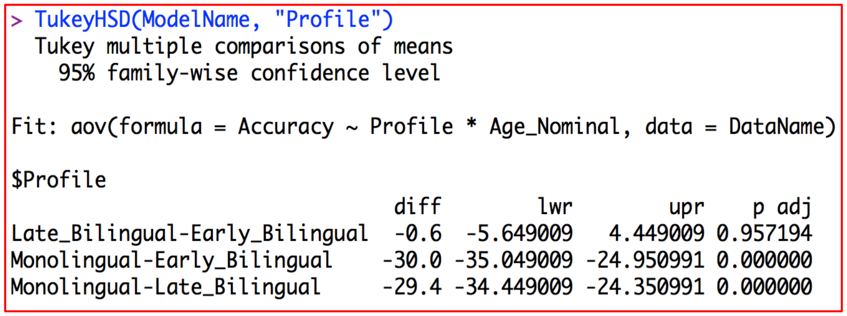
```
> tapply(Accuracy, Profile, mean)
Early_Bilingual  Late_Bilingual     Monolingual
       73.93333        73.33333        43.93333
```

Direction of significant main effect: Observed hierarchy of accuracy scores is Early > Late > Mono.... but a post-hoc test is needed to confirm significant differences therein



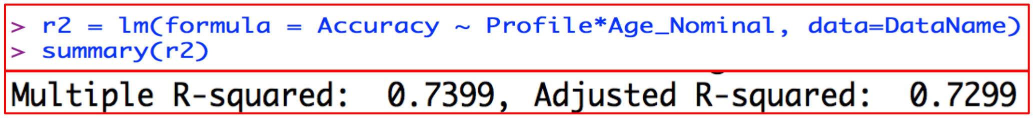
```
> tapply(Accuracy, Profile, sd)
Early_Bilingual  Late_Bilingual     Monolingual
       13.43909        14.48981        12.56872
```

Standard Deviation: Early = 13.4 ; Late = 14.5 ; Mono. = 12.6



```
> TukeyHSD(ModelName, "Profile")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Accuracy ~ Profile * Age_Nominal, data = DataName)

$Profile
                               diff        lwr        upr     p adj
Late_Bilingual-Early_Bilingual -0.6  -5.649009   4.449009 0.957194
Monolingual-Early_Bilingual   -30.0 -35.049009 -24.950991 0.000000
Monolingual-Late_Bilingual    -29.4 -34.449009 -24.350991 0.000000
```

Tukey Post-Hoc on Significant IV (Profile) with 3+ levels:
Late vs. Early: no significant difference (p = .9572)
Mono. vs. Early: significant difference (p < .0001)
Mono. vs. Late: significant difference (p < .0001)



```
> r2 = lm(formula = Accuracy ~ Profile*Age_Nominal, data=DataName)
> summary(r2)
Multiple R-squared:  0.7399, Adjusted R-squared:  0.7299
```

R^2 of the model: 0.7299 (i.e., 72.99% DV variance accounted for by IVs included in model)

When reporting ANOVA format results, often the descriptive statistics appear in a table, with inferential statistics discussed in prose. A descriptive statistics table typically follows the following format, adding additional rows for additional IVs and/or IV levels:

Variable	**Level**	**Observed DV**	**Standard Deviation**
IV1	Level 1	#	#
	Level 2	#	#
IV2	Level 1	#	#
	Level 2	#	#
	Level 3	#	#

Accordingly, the following table would be created in order to display the descriptive statistics of the prior ANOVA analysis:

Variable	**Level**	**Accuracy (in points)**	**Standard Deviation**
Age	18-30 years old	72	16.3
	50-60 years old	52	17.5
Profile	Monolingual	44	12.6
	Early Bilingual	74	13.4
	Late Bilingual	73	14.5

As for the prose description of inferential statistics, main and interaction effects often appear in parentheses using the following notation, whereas post-hoc analyses tend to simply report p values for each relevant comparison:

TestStatisticName[DF#, ResidualsDF#] = TestStatistic# ; p = p#

An example would thus be:

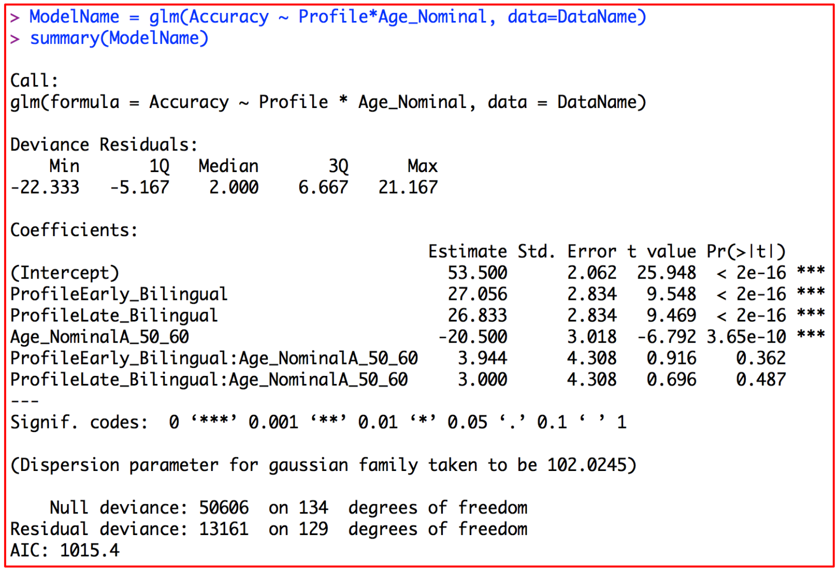
The results of a between-subjects ANOVA (r^2 =.73) testing for main effects of age and profile, as well as an interaction between age and profile, revealed significant main effects of age (F[1,129] = 106.648; p<.0001) and profile (F[2,129] = 129.728; p<.0001). For age, accuracy scores were 20 points higher for 18-30 year olds than 50-60 year olds. For profile, a Tukey post-hoc test showed that whereas early and late bilinguals did not have significantly different accuracy scores (p = .957), both bilingual groups had significantly higher accuracy scores than monolinguals (for early bilinguals, p<.0001 ; for late bilinguals, p<.0001). No significant interaction was obtained (F[2,129] = 0.459; p = .633).

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Interpreting and Reporting Results in Linear Regression Format

*The following example uses the same dataset used for the previous ANOVA interpretation/reporting on the pages prior, with the **IV intercept being coded as 18-30 year olds and Monolinguals**.*

R Outputs for Linear Regression (via Fixed Effects Linear Regression)



```
> ModelName = glm(Accuracy ~ Profile*Age_Nominal, data=DataName)
> summary(ModelName)

Call:
glm(formula = Accuracy ~ Profile * Age_Nominal, data = DataName)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-22.333   -5.167    2.000    6.667   21.167

Coefficients:
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                             53.500      2.062  25.948  < 2e-16 ***
ProfileEarly_Bilingual                  27.056      2.834   9.548  < 2e-16 ***
ProfileLate_Bilingual                   26.833      2.834   9.469  < 2e-16 ***
Age_NominalA_50_60                     -20.500      3.018  -6.792 3.65e-10 ***
ProfileEarly_Bilingual:Age_NominalA_50_60   3.944      4.308   0.916    0.362
ProfileLate_Bilingual:Age_NominalA_50_60    3.000      4.308   0.696    0.487
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 102.0245)

    Null deviance: 50606  on 134  degrees of freedom
Residual deviance: 13161  on 129  degrees of freedom
AIC: 1015.4
```

Model AIC: 1015.4

Intercept (18-30 yr. monolinguals): 53.5 points higher than 0; p<.0001, hence baseline score is treated as **53.5**

Early Bilinguals (18-30 yr.): 27.056 points higher than baseline; p<.0001, hence early bilingual score is treated as **80.556**

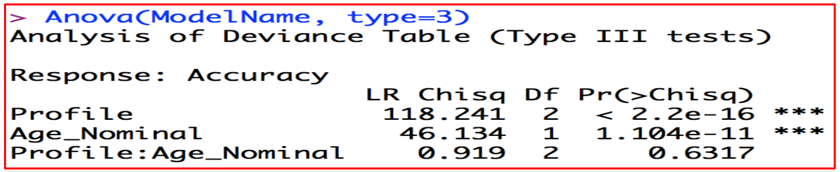
Late Bilinguals (18-30 yr.): 26.833 points higher than baseline; p<.0001, hence late bilingual score is treated as **80.333**

50-60 yr. olds (monolingual): 20.5 points lower than baseline; p<.0001, hence 50-60 yr. old score is treated as **33**

50-60 yr. old Early Bilinguals: 3.944 points higher than score expected for 50-60 yr. old monolinguals (-20.5 from baseline) and 18-30 yr. old Early Bilinguals (+27.056 from baseline); p=.362, hence treated as an adjustment of 0 (i.e., 53.5 + 27.056 – 20.5 + 0 = **60.056**).

50-60 yr. old Late Bilinguals: 3 points higher than score expected for 50-60 yr. old monolinguals (-20.5 from baseline) and 18-30 yr. old Late Bilinguals (+26.833 from baseline); p=.487, hence treated as an adjustment of 0 (i.e., 53.5 + 26.833 – 20.5 + 0 = **59.833**).

Standard Error & Test Statistic (t): For intercept, respectively 2.062 and 25.948. For Early Bilinguals, respectively 2.834 and 9.548. Etc., etc...



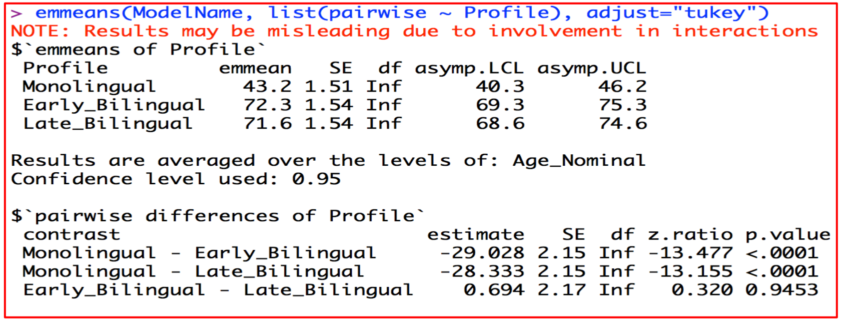
```
> Anova(ModelName, type=3)
Analysis of Deviance Table (Type III tests)

Response: Accuracy
                    LR Chisq Df Pr(>Chisq)
Profile              118.241  2  < 2.2e-16 ***
Age_Nominal           46.134  1  1.104e-11 ***
Profile:Age_Nominal    0.919  2     0.6317
```

Main effects: Profile/Age = significant (p < .0001 for each)

Interaction effect: not significant (p = 0.6317)

Degrees of Freedom: Profile=2, Age=1, Interaction=2, Resid.=129 (appears in the prior output, just above AIC)

Test Statistic (χ^2): Profile=118.241, Age=46.134, Interaction=0.919



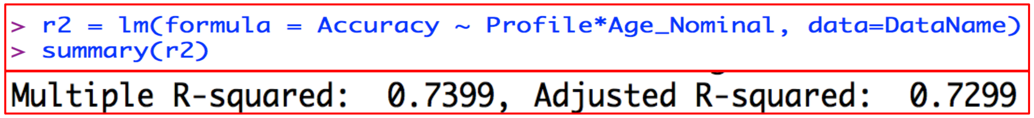
```
> emmeans(ModelName, list(pairwise ~ Profile), adjust="tukey")
NOTE: Results may be misleading due to involvement in interactions
$`emmeans of Profile`
 Profile         emmean   SE  df asymp.LCL asymp.UCL
 Monolingual       43.2 1.51 Inf      40.3      46.2
 Early_Bilingual   72.3 1.54 Inf      69.3      75.3
 Late_Bilingual    71.6 1.54 Inf      68.6      74.6

Results are averaged over the levels of: Age_Nominal
Confidence level used: 0.95

$`pairwise differences of Profile`
 contrast                          estimate   SE  df z.ratio p.value
 Monolingual - Early_Bilingual      -29.028 2.15 Inf -13.477 <.0001
 Monolingual - Late_Bilingual       -28.333 2.15 Inf -13.155 <.0001
 Early_Bilingual - Late_Bilingual     0.694 2.17 Inf   0.320 0.9453
```

Tukey Post-Hoc on Significant IV (Profile) with 3+ levels:

Mono. vs. Early: significant difference (p <.0001)

Mono. vs. Late: significant difference (p <.0001)

Early vs. Late: no significant difference (p = .9453)



```
> r2 = lm(formula = Accuracy ~ Profile*Age_Nominal, data=DataName)
> summary(r2)
```

```
Multiple R-squared:  0.7399, Adjusted R-squared:  0.7299
```

R^2 of the model: 0.7299 (i.e., 72.99% DV variance accounted for by IVs included in model)

When reporting linear regression format results, a table is customary, often following the formatting below:

	β coefficient (in DV units)	Standard Error	Test Statistic (t, F, z, etc.)	p value
Intercept	#	#	#	#
Non-Intercept Level 1	#	#	#	#
Non-Intercept Level 2	#	#	#	#
Non-Intercept Level 3	#	#	#	#
Interaction 1	#	#	#	#
Interaction 2	#	#	#	#

*The intercept for the model is LEVEL, LEVEL, LEVEL, etc...

Accordingly, the following table would be created in order to display the inferential statistics of the prior linear regression analysis:

	β coefficient (points)	Standard Error	t	p value
Intercept	53.500	2.062	25.948	<.0001
Early Bilingual	27.056	2.834	9.548	<.0001
Late Bilingual	26.833	2.834	9.469	<.0001
Age 50-60	-20.500	3.018	-6.792	<.0001
Early Bilingual : 50-60	3.944	4.308	0.916	0.362
Late Bilingual : 50-60	3.000	4.308	0.696	0.487

*The intercept for the model is 18-30 year old monolinguals.

(Alternatively, some describe the intercept levels in the table and omit the asterisked note below the table)

Since regression tables are not the most intuitive to interpret, it is quite frequent to additionally include ANOVA formatting as prose. Main and interaction effects often appear in parentheses using the following notation, whereas post-hoc analyses tend to simply report p values for each relevant comparison:

TestStatisticName[DF#, ResidualsDF#] = TestStatistic# ; p = p#

An example would thus be:

The results of a fixed effects linear regression that included main effects of age and profile, as well as an interaction between age and profile, appear below in table 1. The ANOVA output generated for this regression revealed significant main effects of age (χ^2[1,129] = 46.134; p<.0001) and profile (χ^2[2,129] = 118.241; p<.0001). For age, accuracy scores were 20 points higher for 18-30 year olds than 50-60 year olds. For profile, a Tukey post-hoc test showed that whereas early and late bilinguals did not have significantly different accuracy scores (p = .9453), both bilingual groups had significantly higher accuracy scores than monolinguals (for early bilinguals, p<.0001 ; for late bilinguals, p<.0001). No significant interaction was obtained (χ^2[2,129] = 0.919; p = .6317).

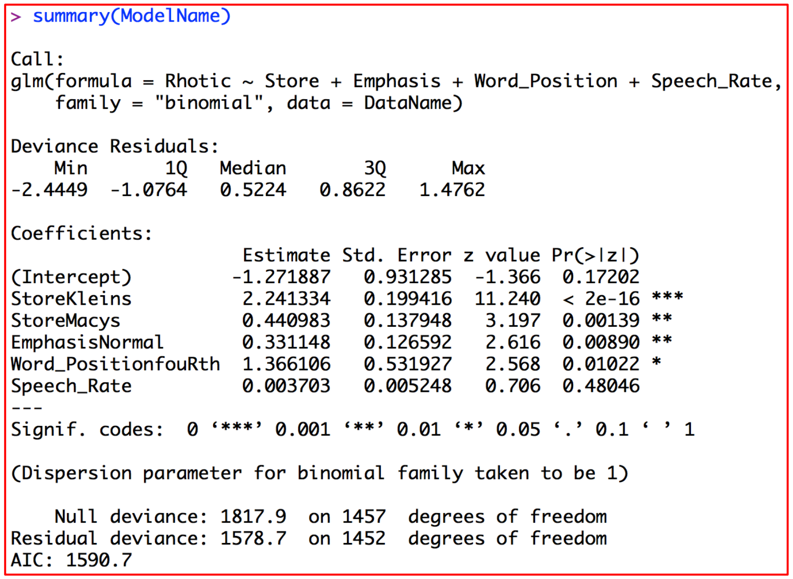
Interpreting and Reporting Results in Logistic Regression Format

The following example comes from running a logistic regression (via fixed effects logistic regression) on the dataset labeled "BinaryDV_LogisticReg" in the Templates and Example Datasets file at the bottom of:

https://spanish-portuguese.berkeley.edu/people/justin-davidson/

In this hypothetical experiment, we tested for main effects of Store (Saks, Kleins, and Macys), Emphasis (Normal and Emphatic), Word Position (fouRth [Word-Internal] and flooR [Word-Final]), and Speech Rate (# of Syllables per Minute) on Rhotic Production (Absent or Present). ***The DV intercept is coded as ABSENT, and the IV intercept is coded as Saks, Emphatic, and flooR.***

R Outputs for Logistic Regression (via Fixed Effects Logistic Regression)



```
> summary(ModelName)

Call:
glm(formula = Rhotic ~ Store + Emphasis + Word_Position + Speech_Rate,
    family = "binomial", data = DataName)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4449  -1.0764   0.5224   0.8622   1.4762

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.271887   0.931285  -1.366  0.17202
StoreKleins         2.241334   0.199416  11.240  < 2e-16 ***
StoreMacys          0.440983   0.137948   3.197  0.00139 **
EmphasisNormal      0.331148   0.126592   2.616  0.00890 **
Word_PositionfouRth 1.366106   0.531927   2.568  0.01022 *
Speech_Rate         0.003703   0.005248   0.706  0.48046
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1817.9  on 1457  degrees of freedom
Residual deviance: 1578.7  on 1452  degrees of freedom
AIC: 1590.7
```

Model AIC: 1590.7

Intercept (Saks, Emphatic, flooR): 1.272 log-odds of ABSENT lower than 0; p=.17202, hence baseline log-odds of ABSENT is treated as **0**

Kleins (Emphatic, flooR): 2.241 log-odds of ABSENT higher than baseline; p<.0001, hence Kleins log-odds of ABSENT is treated as **2.241**

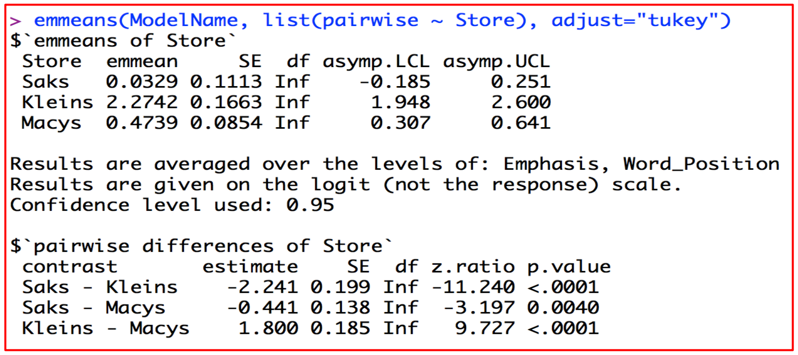
Macys (Emphatic, flooR): 0.441 log-odds of ABSENT higher than baseline; p=.00139, hence Macys log-odds of ABSENT is treated as **0.441**

Normal Emphasis (Saks, flooR): 0.331 log-odds of ABSENT higher than baseline; p=.0089, hence normal emphasis log-odds of ABSENT is treated as **0.331**

fouRth (Saks, Emphatic): 1.366 log-odds of ABSENT higher than baseline; p=.01022, hence fouRth log-odds of ABSENT is treated as **1.366**

Speech Rate: For every 1-unit increase in speech rate (1 syllable per minute), log-odds of ABSENT increases by 0.004; p=.48046, hence change in log-odds of ABSENT for every 1-unit increase in speech rate is treated as **0 (i.e., no change)**

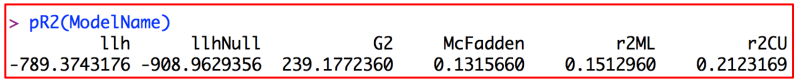
Standard Error & Test Statistic (z): For intercept, respectively 0.931285 and -1.366. For Kleins, respectively 0.199416 and 11.24. Etc., etc...



```
> emmeans(ModelName, list(pairwise ~ Store), adjust="tukey")
$`emmeans of Store`
 Store  emmean     SE  df asymp.LCL asymp.UCL
 Saks   0.0329 0.1113 Inf    -0.185     0.251
 Kleins 2.2742 0.1663 Inf     1.948     2.600
 Macys  0.4739 0.0854 Inf     0.307     0.641

Results are averaged over the levels of: Emphasis, Word_Position
Results are given on the logit (not the response) scale.
Confidence level used: 0.95

$`pairwise differences of Store`
 contrast       estimate    SE  df z.ratio p.value
 Saks - Kleins    -2.241 0.199 Inf -11.240 <.0001
 Saks - Macys     -0.441 0.138 Inf  -3.197 0.0040
 Kleins - Macys    1.800 0.185 Inf   9.727 <.0001
```

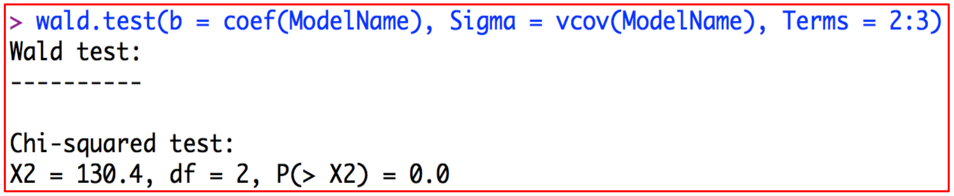
Tukey Post-Hoc on Significant IV (Profile) with 3+ levels:

Saks vs. Kleins: significant difference (p <.0001)
Saks vs. Macys: significant difference (p = .004)
Kleins vs. Macys: significant difference (p <.0001)



```
> pR2(ModelName)
          llh       llhNull            G2      McFadden          r2ML          r2CU
-789.3743176 -908.9629356  239.1772360    0.1315660    0.1512960    0.2123169
```

R^2 of the model: 0.131566 (i.e., 13% DV variance accounted for by IVs included in model)



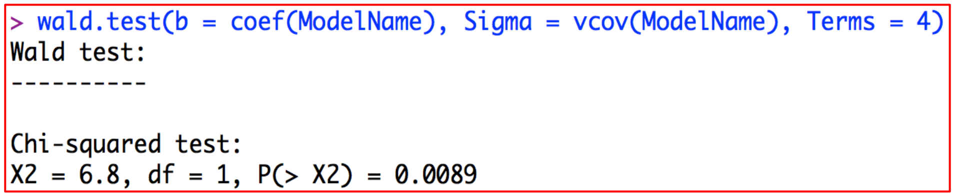
```
> wald.test(b = coef(ModelName), Sigma = vcov(ModelName), Terms = 2:3)
Wald test:
----------

Chi-squared test:
X2 = 130.4, df = 2, P(> X2) = 0.0
```

Wald Test for ANOVA-like Output for Store

Main effect: Store = significant (p < .0001)

Degrees of Freedom: Store = 2, Resid.= 1452 (appears in the prior output, just above AIC)

Test Statistic (χ^2): 130.4



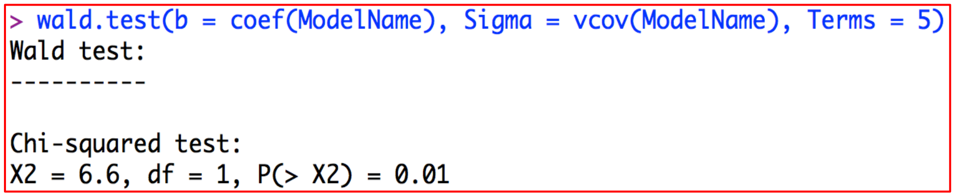
```
> wald.test(b = coef(ModelName), Sigma = vcov(ModelName), Terms = 4)
Wald test:
----------

Chi-squared test:
X2 = 6.8, df = 1, P(> X2) = 0.0089
```

Wald Test for ANOVA-like Output for Emphasis

Main effect: Emphasis = significant (p = .0089)

Degrees of Freedom: Emphasis = 1, Resid.= 1452 (appears in the prior output, just above AIC)

Test Statistic (χ^2): 6.8



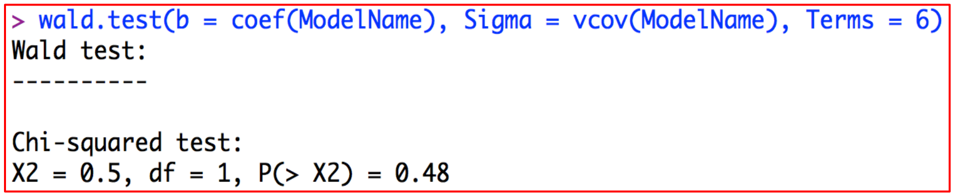
```
> wald.test(b = coef(ModelName), Sigma = vcov(ModelName), Terms = 5)
Wald test:
----------

Chi-squared test:
X2 = 6.6, df = 1, P(> X2) = 0.01
```

Wald Test for ANOVA-like Output for Word Position

Main effect: Word Position = significant (p = .01)

Degrees of Freedom: Word Position = 1, Resid.= 1452 (appears in the prior output, just above AIC)

Test Statistic (χ^2): 6.6



```
> wald.test(b = coef(ModelName), Sigma = vcov(ModelName), Terms = 6)
Wald test:
----------

Chi-squared test:
X2 = 0.5, df = 1, P(> X2) = 0.48
```

Wald Test for ANOVA-like Output for Speech Rate

Main effect: Speech Rate = not significant (p = .48)

Degrees of Freedom: Speech Rate = 1, Resid.= 1452 (appears in the prior output, just above AIC)

Test Statistic (χ^2): 0.5

When reporting logistic regression format results, a table is customary, often following the formatting below:

	β coefficient (in log-odds)	Standard Error	Test Statistic (t, F, z, etc.)	p value	% DV Intercept
Intercept	#	#	#	#	#
Non-Intercept Level 1	#	#	#	#	#
Non-Intercept Level 2	#	#	#	#	#
Non-Intercept Level 3	#	#	#	#	#
Interaction 1	#	#	#	#	#
Interaction 2	#	#	#	#	#

*The intercept for the model is LEVEL, LEVEL, LEVEL, etc...

Accordingly, the following table would be created in order to display the inferential statistics of the prior linear regression analysis:

	β coefficient (log-odds)	Standard Error	z	p value	Rhotic Absence
Intercept	-1.272	0.931	-1.366	0.172	50%
Kleins	2.241	0.199	11.24	<.0001	90.39%
Macys	0.441	0.138	3.197	0.0014	60.85%
Normal	0.331	0.127	2.616	0.0089	58.20%
fouRth	1.367	0.532	2.568	0.0102	79.69%
Speech Rate	0.004	0.005	0.706	0.4805	0% change per 1 syllable-per-minute increase

*The intercept for the model is Saks, Emphatic, and flooR.

(Alternatively, some describe the intercept levels in the table and omit the asterisked note below the table)

Calculations of Predicted % DV Intercept (All β coefficients lacking statistical significance are to be treated as <u>0</u> in calculations **)**

An additional column that is helpful to add to tables of logistic regression is one that expresses each row of β coefficients, which are by default in log-odds, as % of the DV intercept (i.e., what % of Rhotic ABSENCE does the model predict for Kleins vs. Macys, etc.).

For the **<u>intercept row alone</u>**, the predicted % of DV intercept is: exp(β) / (1 + exp(β)) = ## (adjust the decimal two spaces to the right to yield the %)

Accordingly, we'd have the following for the intercept row → exp(**0**)/(1 + exp(**0**)) = .5 ; **50%**

For all other (non-intercept) nominal IV rows, the predicted % of DV intercept is:

exp(Intercept:β + Non-Intercept:β) / (1 + exp(Intercept:β + Non-Intercept:β)) = ## (adjust the decimal two spaces to the right to yield the %)

Accordingly, we'd have the following calculations of predicted rhotic % for each of the remaining nominal IV levels:

Kleins: exp(**0** + **2.241**)/(1 + exp(**0** + **2.241**)) = 90.39%
Macys: exp(**0** + **0.441**)/(1 + exp(**0** + **0.441**)) = 60.85%
Normal: exp(**0** + **0.331**)/(1 + exp(**0** + **0.331**)) = 58.20%
fouRth: exp(**0** + **1.367**)/(1 + exp(**0** + **1.367**)) = 79.69%

For continuous IVs like Speech Rate, the calculation of % change in DV intercept for every 1-unit increase in the IV is simply the difference between 50% (a log-odds of 0) and the % corresponding to the log-odds obtained in the model.

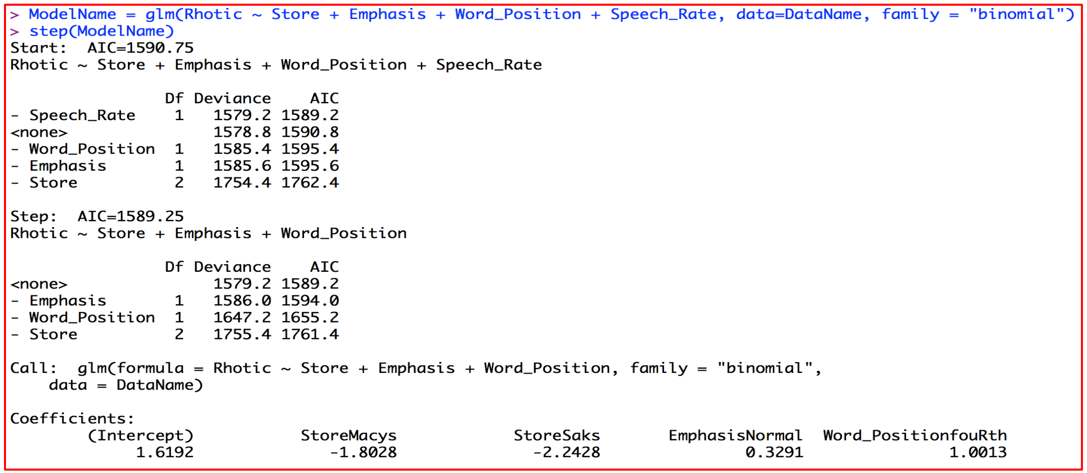
exp(0.004)/(1 + exp(0.004)) = .501; or 50.1%; → 50.1% - 50% = <u>0.1%</u>

This would mean that for every 1 syllable-per-minute increase, rhotic absence would be predicted to increase by 0.1%. HOWEVER: in this model, speech rate was NOT SIGNIFICANT, so the β coefficient is treated NOT as 0.004, but rather as 0. Thus, there is no change in % of rhotic ABSENCE.

Interpreting and Reporting Results in Variable Rule Analysis (Varbrul) Format

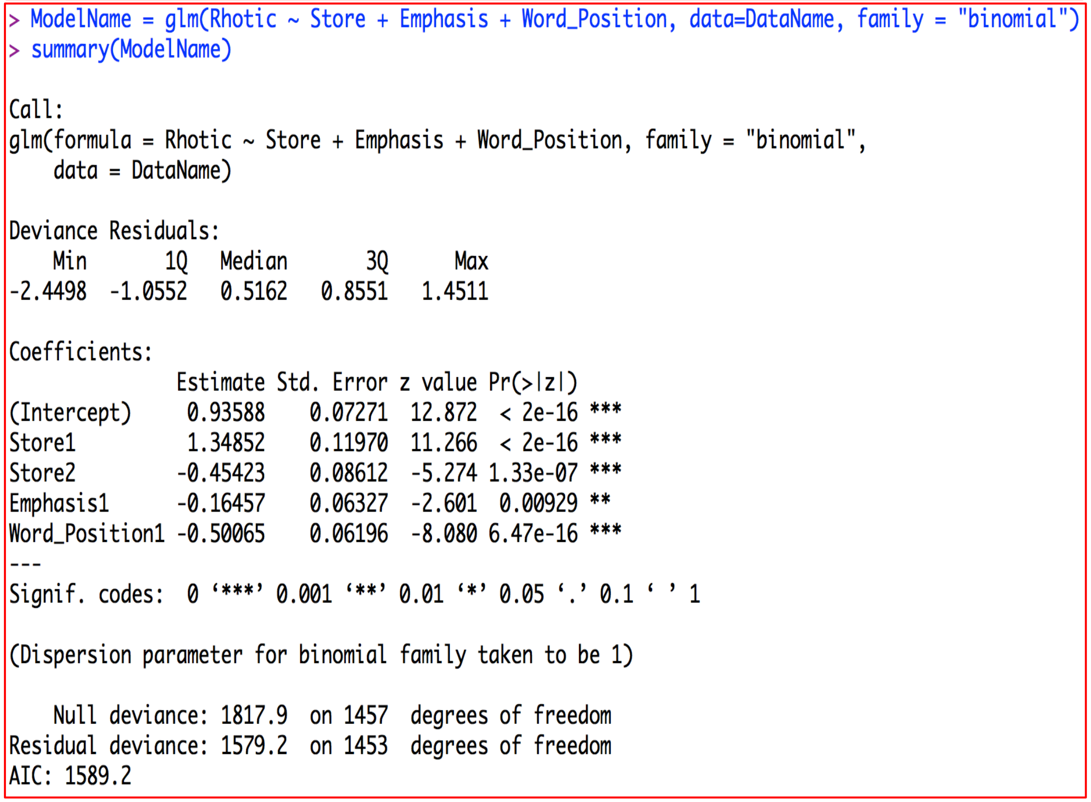
The following example uses the same dataset used for the previous logistic regression interpretation/reporting on the pages prior.

R Outputs for Varbrul Step-wise Logistic Regression (via Varbul – Step-wise Fixed Effects Logistic Regression)



```
> ModelName = glm(Rhotic ~ Store + Emphasis + Word_Position + Speech_Rate, data=DataName, family = "binomial")
> step(ModelName)
Start:  AIC=1590.75
Rhotic ~ Store + Emphasis + Word_Position + Speech_Rate

                Df Deviance    AIC
- Speech_Rate    1   1579.2 1589.2
<none>               1578.8 1590.8
- Word_Position  1   1585.4 1595.4
- Emphasis       1   1585.6 1595.6
- Store          2   1754.4 1762.4

Step:  AIC=1589.25
Rhotic ~ Store + Emphasis + Word_Position

                Df Deviance    AIC
<none>               1579.2 1589.2
- Emphasis       1   1586.0 1594.0
- Word_Position  1   1647.2 1655.2
- Store          2   1755.4 1761.4

Call:  glm(formula = Rhotic ~ Store + Emphasis + Word_Position, family = "binomial",
    data = DataName)

Coefficients:
        (Intercept)           StoreMacys            StoreSaks       EmphasisNormal  Word_PositionfouRth
             1.6192              -1.8028              -2.2428               0.3291               1.0013
```

Call: Best-fit model, via step-wise logistic regression, is one with <u>Store, Emphasis, and Word Position</u> as IVs. (Speech rate was dropped out!)



```
> ModelName = glm(Rhotic ~ Store + Emphasis + Word_Position, data=DataName, family = "binomial")
> summary(ModelName)

Call:
glm(formula = Rhotic ~ Store + Emphasis + Word_Position, family = "binomial",
    data = DataName)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4498  -1.0552   0.5162   0.8551   1.4511

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.93588    0.07271  12.872  < 2e-16 ***
Store1          1.34852    0.11970  11.266  < 2e-16 ***
Store2         -0.45423    0.08612  -5.274 1.33e-07 ***
Emphasis1      -0.16457    0.06327  -2.601  0.00929 **
Word_Position1 -0.50065    0.06196  -8.080 6.47e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1817.9  on 1457  degrees of freedom
Residual deviance: 1579.2  on 1453  degrees of freedom
AIC: 1589.2
```

Model AIC: 1589.2
Model Residual Deviance: 1579.2
Model Degrees of Freedom: 1453
Null Model Deviance: 1817.9
Null Model Degrees of Freedom: 1457
Negative Log Likelihood: (1579.2 / -2) = -789.6

Intercept (ABSENT production in general): 0.93588 log-odds of ABSENT; p<.0001, hence overall log-odds of ABSENT is treated as **0.93588**

Store1 (Kleins [alphabetically 1st]): 1.34852 log-odds of ABSENT; p<.0001, hence Kleins log-odds of ABSENT is treated as **1.34852**

Store2 (Macys [alphabetically 2nd]): -0.45423 log-odds of ABSENT; p<.0001, hence Macys log-odds of ABSENT is treated as **-0.45423**

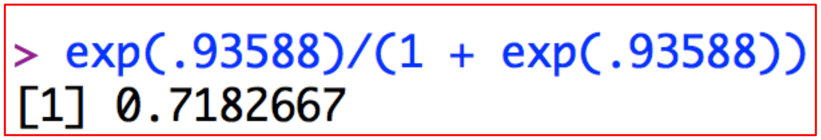
Store3 (Saks [alphabetically last]): Log-odds of ABSENT is the value that sums to 0 with previous Store log-odds.
1.34852 - 0.45423 - **.89429** = 0 ; hence **-.89429**

Emphasis1 (Emphatic [alphabetically 1st]): -0.16457 log-odds of ABSENT; p=.00929, hence Emphatic log-odds of ABSENT is treated as **-0.16457**

Emphasis2 (Normal [alphabetically last]): Log-odds of ABSENT is the value that sums to 0 with previous Emphasis log-odds.
-0.16457 + **0.16457** = 0 ; hence **0.16457**

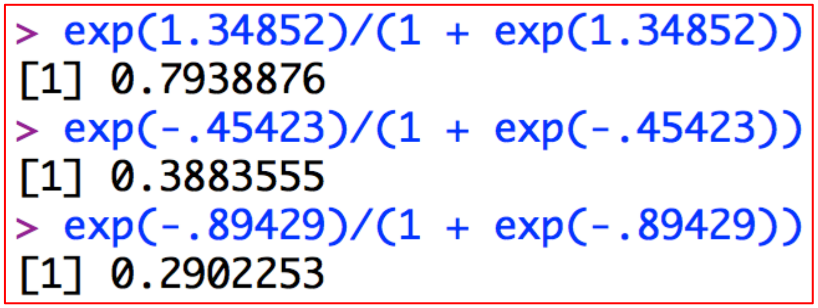
WordPosition1 (flooR [alphabetically 1st]): -0.50065 log-odds of ABSENT; p<.0001, hence flooR log-odds of ABSENT is treated as **-0.50065**

WordPosition2 (fouRth [alphabetically last]): Log-odds of ABSENT is the value that sums to 0 with previous W.P. log-odds.
-0.50065 + **0.50065** = 0 ; hence **0.50065**



```
> exp(.93588)/(1 + exp(.93588))
[1] 0.7182667
```

Corrected Mean / Input: Using log-odds of ABSENT from intercept, corrected mean / input of model = 0.7182667, or rhotic ABSENCE at 72% overall

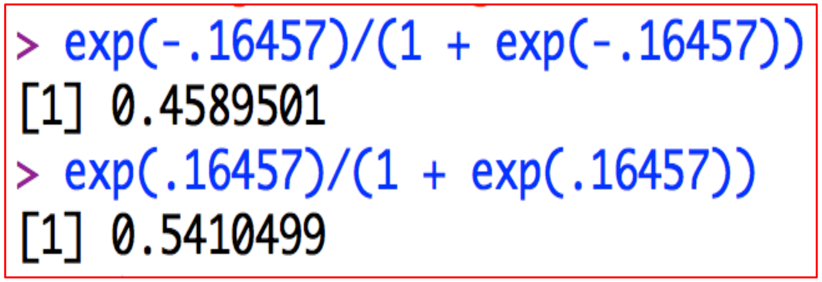


```
> exp(1.34852)/(1 + exp(1.34852))
[1] 0.7938876
> exp(-.45423)/(1 + exp(-.45423))
[1] 0.3883555
> exp(-.89429)/(1 + exp(-.89429))
[1] 0.2902253
```

Factor Weights for each level of Store: Using log-odds of ABSENT from each Store, factor weights are:

Kleins	=	**.79**
Macys	=	**.39**
Saks	=	**.29**

Range for Store = (Maximum F.Weight – Minimum F.Weight) x 100
(.79 - .29) x 100 = **50**

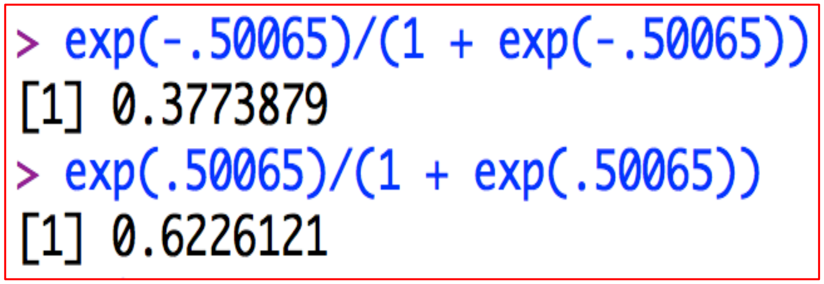


```
> exp(-.16457)/(1 + exp(-.16457))
[1] 0.4589501
> exp(.16457)/(1 + exp(.16457))
[1] 0.5410499
```

Factor Weights for each level of Emphasis: Using log-odds of ABSENT from each Emphasis, factor weights are:

Emphatic	=	**.46**
Normal	=	**.54**

Range for Emphasis = (Maximum F.Weight – Minimum F.Weight) x100
(.54 - .46) x 100 = **8**

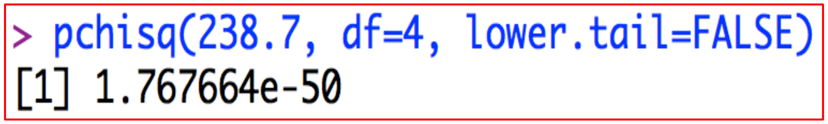


```
> exp(-.50065)/(1 + exp(-.50065))
[1] 0.3773879
> exp(.50065)/(1 + exp(.50065))
[1] 0.6226121
```

Factor Weights for each level of Word Position: Using log-odds of ABSENT from each Word Position, factor weights are:

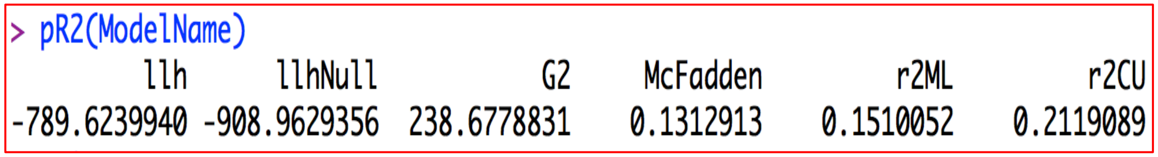
flooR	=	**.38**
fouRth	=	**.62**

Range for Emphasis = (Maximum F.Weight – Minimum F.Weight) x100
(.62 - .38) x 100 = **24**



```
> pchisq(238.7, df=4, lower.tail=FALSE)
[1] 1.767664e-50
```

p value for Model < .0001 (238.7 is the difference between residual and null deviances, and 4 is the difference between null and residual degrees of freedom)



```
> pR2(ModelName)
          llh       llhNull           G2     McFadden         r2ML         r2CU
-789.6239940 -908.9629356  238.6778831    0.1312913    0.1510052    0.2119089
```

R^2 of the model: 0.1312913 (i.e., 13% DV variance accounted for by IVs included in model)

Variationist sociolinguists have a tradition of presenting step-wise logistic regression results in a unique manner (i.e., the results themselves are not different, but rather their presentation is considerably distinct from the previously covered logistic regression). This format is known as **Variable Rule Analysis**, or **Varbrul**. The following table presents a template of Varbrul results:

Factor (aka IV)	Level	Factor Weight	% DV Intercept Observed for this Level in the Dataset	N (Total Instances of this Level in the Dataset)
Factor 1	Name	##	## (N=##)	##
	Name	##	## (N=##)	##
	Name	##	## (N=##)	##
		RANGE: ##		
Factor 2	Name	##	## (N=##)	##
	Name	##	## (N=##)	##
		RANGE: ##		

* Corrected Mean / Input = ## ; Model significance = ## ; Log Likelihood = -## ; Total N (all instances of DV) = ## ; Factors not selected as significant: NAME

Accordingly, the following table would be created in order to display the inferential statistics of the prior linear regression analysis:

Factor	Level	Factor Weight	% Rhotic Absence	N
Store	Kleins	**.79**	90.3% (N=390)	432
	Macys	.39	62.8% (n=422)	672
	Saks	.29	52.5% (n=186)	354
		RANGE: 50		
Emphasis	Normal	**.54**	70.3% (n=644)	916
	Emphatic	.46	65.3% (n=354)	542
		RANGE: 8		
Word Position	fouRth	**.62**	77.2% (n=590)	764
	flooR	.38	58.8% (n=408)	694
		RANGE: 24		

* Corrected Mean / Input = .72 ; Model significance = p<.0001 ; Log Likelihood = -789.6 ; Total N = 1458 ; Factors not selected as significant: Speech Rate

(Factor weights for significant nominal IVs greater than .5 are often **bolded**. Additionally, some authors include all information for nominal factors not selected as significant in the table as well, and simply place [brackets] around all their corresponding factor weights.)

FACTOR WEIGHT is a value computed manually in R and **only applies to nominal IVs**. Values above .5 indicate a favoring of the DV intercept, whereas below .5 indicate a disfavoring of the DV intercept. Range is a measure of factor strength (magnitude of effect), and is simply 100 x the difference between largest factor weight and smallest factor weight. Factor weights are calculated with the following formula: **(** All β coefficients lacking statistical significance are to be treated as 0 in calculations **)**

exp(β) / (1 + exp(β)) = ## (leave the decimal; round to hundredths)

Accordingly, we'd have the following for the Kleins row ➔ exp(1.34852)/(1 + exp(1.34852)) = **.79**

To calculate observed instances and total tokens in your dataset, you can either make use of the qhpvt function (step 3d in most tests), specifying one IV and the DV [EXAMPLE: qhpvt(DataName, "**IV1**", c("**DV**"), "n()")], or filter/pivot-table manually in Excel.

Between-Subjects ANOVA (Results appear as individual IVs that either are or are not significant)

(All IVs are nominal and between-subjects [each participant can only contribute for 1 level], DV = continuous, no random effects)

LEGEND: *Italics* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.
Words - Create your own label, colors show continuity of label
DV ; **IV** ; **V** : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)
IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)
IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)

1a) install.packages("car") [If there's a pop-question about binary versions available and installing
1b) install.packages("pivottabler") sources from a package needing compilation, say **no**]
2a) library(car)
2b) library(pivottabler)
3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]
3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**]

5) leveneTest(**DV** ~ **IV1 * IV2 * IV3**, data=DataName) [Equal between-subjects residuals variance check (aka heteroscedasticity); you want p > 0.05, otherwise you may need to transform DV or use a non-parametric test.]
6) ModelName = aov(**DV** ~ **IVdump**, data=DataName)

7a) Residuals=residuals(ModelName) [Residuals normality check; you want p > 0.05, otherwise you may need
7b) shapiro.test(Residuals) to transform DV or use a non-parametric test.]

8a) summary(ModelName) [Results in ANOVA format (p value for each IV); for direction of effect, see 8b]
8b) tapply(**DV**, **IV**, mean) [Computes DV per IV level; Run this on each significant IV to interpret its effect direction; Change mean to sd to instead/additionally compute standard deviations]

9a) TukeyHSD(ModelName, "**IV1**") [Post-hoc for significant nominal IV with 3+ levels]
9b) TukeyHSD(ModelName, "**IV1**:**IV2**") [Post-hoc for significant interaction between 2 nominal IVs]

10a) r2 = lm(formula = **DV** ~ **IVdump**, data=DataName) [Obtains the $\mathbf{r^2}$ for the model; ignore the output and just search for
10b) summary(r2) the **adjusted** $\mathbf{r^2}$ value. *Higher* r^2 *= more DV variance accounted for by model; .82 = 82%*]

11) boxplot(**DV** ~ **IV1 + IV2 + IV3**) [Quick plot to visualize results]

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Within-Subjects (aka Repeated Measures or Mixed Model) ANOVA (Results appear as individual IVs that are/aren't significant)
(All IVs are nominal and at least 1 is within-subjects [all participants contribute for all levels], DV = continuous, no random effects)

Legend:
Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.
Words - Create your own label, colors show continuity of label
DV ; **IV** ; **V** : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)
IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)
IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)

1a) install.packages("car") [If there's a pop-question about binary versions available and installing sources from a package needing compilation, say **no**]
1b) install.packages("emmeans")
1c) install.packages("ez")
1d) install.packages("pivottabler")

2a) library(car)
2b) library(emmeans)
2c) library(ez)
2d) library(pivottabler)

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]
3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts; permits assessment of data as 100% balanced or not for step 5a)]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**]

5a) Sphericity = ezANOVA(DataName, dv = **DV**, wid = **SubjectNumber**, within = .(**IV1, IV2**), between = .(**IV3, IV4**), type = #)
[Mauchly Sphericity Test, i.e., equal residual variances, for any **within-subjects IVs with 3+ levels**. IVs in the "within" parentheses are within-subjects, IVs in the "between" parentheses are between-subjects, and type is either 2 for 100% balanced cell counts, or 3 for not 100% balanced cell counts (see step 3d). For definitions of within- vs. between-subjects, see green text at top of this page and the prior page]

5b) Sphericity

[Ignore everything and just look at Sphericity test; you want $p > 0.05$, otherwise you may need to transform DV or use a non-parametric test. Remember that this only applies to **within-subjects IVs with 3+ levels**! If yours only has/have 2 levels, then you should skip step 5 entirely!]

6) leveneTest(**DV** ~ **IV1 * IV2 * IV3**, data=DataName)

[Equal between-subjects residuals variance check (aka heteroscedasticity); you want $p > 0.05$, otherwise you may need to transform DV or use a non-parametric test.]

7) ModelName = aov(**DV** ~ **IVdump** + Error(**SubjectNumber**/(**IV1 * IV2**)), data=DataName)

[**IVs** after **SubjectNumber** are all **within-subjects** IVs]

8a) Residuals=residuals(ModelName)
8b) shapiro.test(Residuals)

[Residuals normality check; you want $p > 0.05$, otherwise you may need to transform DV or use a non-parametric test.]

9a) summary(ModelName)

[Results in ANOVA format (p value for each IV); for direction of effect, see 9b]

9b) tapply(**DV**, **IV**, mean)

[Computes DV per IV level; Run this on each significant IV to interpret its effect direction; Change mean to sd to instead/additionally compute standard deviations]

10a) NewName = emmeans(ModelName, ~ **IV**)
10b) pairs(NewName)

[Post-hoc for significant nominal IV with 3+ levels]

11a) YetAnotherName = emmeans(ModelName, pairwise ~ **IV1** | **IV2**)
11b) OneLastName = emmeans(ModelName, pairwise ~ **IV2** | **IV1**)
11c) summary(YetAnotherName)
11d) summary(OneLastName)

[Post-hoc for significant interaction between 2 nominal IVs; note that you basically run things twice, flipping order of IVs around]

12a) r2 = lm(formula = **DV** ~ **IVdump**, data=DataName)
12b) summary(r2)

[Obtains the $\mathbf{r^2}$ for the model; ignore the output and just search for the **adjusted** $\mathbf{r^2}$ value]
[*higher r^2 = more DV variance accounted for by model; .82 = 82%*]

13) boxplot(**DV** ~ **IV1 + IV2 + IV3**)

[Quick plot to visualize results]

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Fixed Effects Linear Regression (Results appear as coefficients for all-1 levels per IV)

(No restriction on IVs, DV = continuous, no random effects)

Reference/Intercept (for the IVs) = Alphabetical, See Step 5 to change!

Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

DV ; **IV** ; **V** : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)

IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)

1a) install.packages("car") [If there's a pop-question about binary versions available and installing sources from a package needing compilation, say **no**]
1b) install.packages("emmeans")
1c) install.packages("lmtest")
1d) install.packages("pivottabler")
1e) install.packages("ggplot2")

2a) library(car)
2b) library(emmeans)
2c) library(lmtest)
2d) library(pivottabler)
2e) library(ggplot2)

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]
3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts (needed for ANOVA output in step 9b)]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**]

5) DataName = within(DataName, **IV** <- relevel(**IV**, ref = "NewIVReferenceLevel")) [**OPTIONAL: For REGRESSION OUTPUT:** Set a not-alphabetically-first level of any one nominal IV to be the reference/intercept in the model (step 9a). Run this code multiple times to change the reference / intercept for various nominal IVs, one at a time]

[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Male" *Changes intercept from **Female** to **Male***]
[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Female" *Does nothing – Intercept was already **Female***]

6) ModelName = glm(**DV** ~ **IVdump**, data=DataName)

7a) Residuals=residuals(ModelName)
7b) shapiro.test(Residuals)

[Residuals normality check; you want $p > 0.05$, otherwise you may need to transform DV or use a non-parametric test.]

8) bptest(ModelName)

[Equal between-subjects residuals variance check (aka heteroscedasticity; you want $p > 0.05$, otherwise you may need to transform DV or use a non-parametric test.]

9a) summary(ModelName)

[Results as a **linear regression** with β coefficients and **Model AIC**]

9b) Anova(ModelName, type=#)

[Results as an **ANOVA** with p values for each IV – type is either 2 for 100% balanced cell counts, or 3 for not 100% balanced cell counts. Refer to step 3d to assess equal counts for all nominal IVs. For direction of effect, see 9c]

[*Lower AIC = better model fit*]

9c) tapply(**DV**, **IV**, mean)

[Computes DV per IV level; Run this on each significant IV to interpret its effect direction; Change mean to sd to instead/additionally compute standard deviations]

10a) emmeans(ModelName, list(pairwise ~ **IV**), adjust="tukey")

[Post-hoc for significant Nominal IV with 3+ levels]

10b) emmeans(ModelName, list(pairwise ~ **IV1*IV2*IV3**), adjust="tukey")

[Post-hoc for significant interaction between 2+ Nominal IVs]

11a) r2 = lm(formula = **DV** ~ **IVdump**, data=DataName)
11b) summary(r2)

[Obtains the $\mathbf{r^2}$ for the model; ignore the output and just search for the **adjusted** $\mathbf{r^2}$ value]

[*higher r^2 = more DV variance accounted for by model, i.e., .82 = 82%*]

12a) ggplot(DataName, aes(**IV1**, **DV**)) + geom_point(size=2, aes(color=**IV2**, shape=**IV3**))

[Quick plot to visualize results where IV1 is continuous and IV2 and 3 are nominal]

12b) boxplot(**DV** ~ **IV1 + IV2 + IV3**)

[Quick plot to visualize results where all IVs are nominal]

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Fixed Effects Logistic Regression (Results appear as coefficients for all minus one levels per IV)

(No restriction on IVs, DV = binary [2 levels] nominal/discrete, no random effects)

Reference/Intercept for <u>DV</u> = <u>REVERSE</u>-Alphabetical, See Step 5 to change!

Reference/Intercept for <u>IVs</u> = Alphabetical, See Step 5 to change!

Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

<u>DV</u> ; **<u>IV</u>** ; **<u>V</u>** : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)

<u>IVdump</u> - Name of the IVs from Excel you want to consider, separated by plus signs: **<u>IV1</u>** + **<u>IV2</u>** + **<u>IV3</u>** + **<u>IV4</u>** (etc...)

<u>IVdump</u> - <u>Interactions</u> between IVs are denoted with an asterisk instead of a plus sign: **<u>IV1</u>** + **<u>IV2 * IV3</u>** + **<u>IV2 * IV4</u>** (etc...)

<u>Rowdump</u> - From output of step 7a, the intercept is row 1, followed by row 2 (a level), row 3 (a level), etc. For an IV with only 2 levels, simply type the row # of the non-intercept level, such as: **5** (meaning this factor has an intercept level and the other level is in row 5 of the summary [step 7a] output). For an IV with 3+ levels, type the row # of the first non-intercept level, followed by a colon (:), then the row # of last level of that factor, such as: **2:5** (meaning this factor has an intercept level and the remaining 3 levels occupy rows 2, 3, 4, and 5 of the summary [step 7a] output).

1a) install.packages("car") [if there's a pop-question about binary versions available and installing sources from a package needing compilation, say **<u>no</u>**]

1b) install.packages("emmeans")

1c) install.packages("pscl")

1d) install.packages("pivottabler")

1e) install.packages("aod") [only if you want to force an ANOVA-like output]

2a) library(car)

2b) library(emmeans)

2c) library(pscl)

2d) library(pivottabler)

2e) library(aod) [only if you want to force an ANOVA-like output]

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]

3b) head(DataName) [Reminds you of **<u>V</u>** names; confirmation of uploading the right file]

3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]

3d) qhpvt(DataName, "**<u>IV1</u>**", c("**<u>IV2</u>**", "**<u>IV3</u>**"), "n()") [Creates table for <u>nominal</u> IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a **<u>V</u>** as DataName$**<u>V</u>**]

5) DataName = within(DataName, V <- relevel(V, ref = "NewIVReferenceLevel/OLDDVReferenceLevel")) [**OPTIONAL**: set a not-alphabetically-first level of any IV or the alphabetically-first level of the DV to be the reference/intercept in the model (step 7a). Run this code multiple times to change the reference/intercept for multiple Vs, one at a time. NOTE THAT A DV REFERENCE CHANGE IS COUNTER-INTUITUVE: YOU WRITE THE DEFAULT/OLD LEVEL TO CHANGE IT]

[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Male" *Changes intercept from **Female** to **Male***]
[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Female" *Does nothing – Intercept was already **Female***]
[EXAMPLE for a DV with "**F**ull" and "**A**spirated" as levels: ref = "Full" *Changes intercept from **Full** to **Aspirated***]
[EXAMPLE for a DV with "**F**ull" and "**A**spirated" as levels: ref = "Aspirated" *Does nothing – Intercept was already **Full***]

6) ModelName = glm(**DV** ~ **IVdump**, data=DataName, family = "binomial")

7a) summary(ModelName) [Results as a **regression** with log-odds coefficients (β) and **model AIC**]
[*Lower AIC = better model fit*]
[Should you want to convert log-odds coefficients into DV intercept %s, copy/paste the following into R, replacing **X** with the β coefficient, and move decimal over to the right by 2 spaces: exp(**X**)/(1 + exp(**X**))
OR, use the code inv.logit(**X**) after running steps 1 & 2 on the package gtools , and move decimal over to the right by 2 spaces]

[The ANOVA-like output of individual IVs as significant or not significant is available for logistic regression, but cumbersome and not often reported. It is common to simply intuit which IVs are significant based on which IVs have levels with significant β coefficients or which interactions are significant.]

7b) wald.test(b = coef(ModelName), Sigma = vcov(ModelName), Terms = **Rowdump**) [ANOVA-like output for an IV that, ignoring the intercept (in row 1), contains the level or levels specified by **Rowdump**]

8a) emmeans(ModelName, list(pairwise ~ **IV**), adjust="tukey") [Post-hoc for significant Nominal IV with 3+ levels]
8b) emmeans(ModelName, list(pairwise ~ **IV1*IV2*IV3**), adjust="tukey") [Post-hoc for significant interaction between 2+ Nominal IVs]

9) pR2(ModelName) [Obtains the **pseudo-r^2** for the model; the value you want is McFadden!]
[*higher r^2 = more DV variance accounted for by model; .82 = 82%*]

10) plot(table(**IV1**, **IV2**, **IV3**, **DV**), col=T, main="InsertTitleHere") [Quick mosaic plot to visualize results for nominal IVs where IV1 levels will be columns and IV2+ levels will be rows; Use this on an individual IV to confirm what you think the DV intercept is based on the + or – β coefficients from step 7a]

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Mixed Effects Linear Regression (Results appear as coefficients for all minus one levels per IV)
(No restriction on IVs, DV = continuous, Presence of 1+ random effects [at least 2 data points per Random IV level])
Reference/Intercept (for the IVs) = Alphabetical, See Step 5 to change!

Legend:
Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.
Words - Create your own label, colors show continuity of label
DV or **IV**; **V**; **RIV**: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column
IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)
IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)
IVRandomdump - For each random intercept, use the following notation with parentheses: (1|**RIV1**) + (1|**RIV2**) (etc...)
IVRandomdump - For a random slope for a specific random intercept, replace the "1" for the random intercept codes above with the within-subjects IV(s) of choice: (**IV1**|**RIV1**) + (**IV1**|**RIV2**) + (**IV2**|**RIV2**) (etc...)

1a) install.packages("afex") [if there's a pop-question about binary versions available and installing
1b) install.packages("lmerTest") sources from a package needing compilation, say **no**]
1c) install.packages("emmeans")
1d) install.packages("r2glmm")
1e) install.packages("pivottabler")
1f) install.packages("ggplot2")

2a) library(afex)
2b) library(lmerTest)
2c) library(emmeans)
2d) library(r2glmm)
2e) library(pivottabler)
2f) library(ggplot2)

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]

3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**]

5) DataName = within(DataName, **IV** <- relevel(**IV**, ref = "NewIVReferenceLevel")) [**OPTIONAL: For REGRESSION OUTPUT:** Set a not-alphabetically-first level of any one nominal IV to be the reference/intercept in the model (step 7a). Run this code multiple times to change the reference / intercept for various nominal IVs, one at a time]

[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Male" *Changes intercept from **Female** to **Male***]

[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Female" *Does nothing – Intercept was already **Female***]

6) ModelName = lmer(**DV** ~ **IVdump** + **IVRandomdump**, data=DataName, REML=FALSE)

[An ‘observations cannot be < groups’ error means there is only 1 data point per RIV level. You can only use a fixed effects model!]

7a) summary(ModelName) [results as a **regression** with β coefficients and **model AIC**]

7b) mixed(ModelName, DataName) [results as an **ANOVA** with p values for each IV; For direction of effect, see 7c]

7c) tapply(**DV**, **IV**, mean) [Computes DV per IV level; Run this on each significant IV to interpret its effect direction; Change mean to sd to instead/additionally compute standard deviations]

8a) emmeans(ModelName, list(pairwise ~ **IV**), adjust="tukey") [Post-hoc for significant Nominal IV with 3+ levels]

8b) emmeans(ModelName, list(pairwise ~ **IV1*IV2*IV3**), adjust="tukey") [Post-hoc for significant interaction between 2+ Nominal IVs]

9) r2beta(ModelName, method ="nsj") [Obtains the **r^2** for the model and for each individual IV]

[*higher r^2 = more DV variance accounted for by model; .82 = 82%*]

10a) ggplot(DataName, aes(**IV1**, **DV**)) + geom_point(size=2, aes(color=**IV2**, shape=**IV3**)) [Quick plot to visualize results where IV1 is continuous and IV2 and 3 are nominal]

10b) boxplot(**DV** ~ **IV1 + IV2 + IV3**) [Quick plot to visualize results where all IVs are nominal]

Mixed Effects Logistic Regression (Results appear as coefficients for all minus one levels per IV)
(No restriction on IVs, DV = binary & nominal/discrete, Presence of 1+ random effects [at least 2 data points per Random IV level])
Reference/Intercept for DV = REVERSE-Alphabetical, See Step 5 to change!
Reference/Intercept for IVs = Alphabetical, See Step 5 to change!

Legend:
Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.
Words - Create your own label, colors show continuity of label
DV or **IV**; **V**; **RIV**: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column
IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)
IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)
IVRandomdump - For each random intercept, use the following notation with parentheses: (1|**RIV1**) + (1|**RIV2**) (etc...)
IVRandomdump - For a random slope for a specific random intercept, replace the "1" for the random intercept codes above with the within-subjects IV(s) of choice: (**IV1**|**RIV1**) + (**IV1**|**RIV2**) + (**IV2**|**RIV2**) (etc...)

1a) install.packages("afex") [if there's a pop-question about binary versions available and installing sources from a package needing compilation, say **no**]
1b) install.packages("lmerTest")
1c) install.packages("emmeans")
1d) install.packages("r2glmm")
1e) install.packages("pivottabler")

2a) library(afex)
2b) library(lmerTest)
2c) library(emmeans)
2d) library(r2glmm)
2e) library(pivottabler)

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]
3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**]

5) DataName = within(DataName, **V** <- relevel(**V**, ref = "NewIVReferenceLevel/OLDDVReferenceLevel")) [**OPTIONAL**: set a not-alphabetically-first level of any IV or the alphabetically-first level of the DV to be the reference/intercept in the model (step 7). Run this code multiple times to change the reference/intercept for multiple Vs, one at a time. NOTE THAT A DV REFERENCE CHANGE IS COUNTER-INTUITUVE: YOU WRITE THE DEFAULT/OLD LEVEL TO CHANGE IT]

[EXAMPLE for an IV with "**M**ale" and "**Female**" as levels: ref = "Male" *Changes intercept from **Female** to **M**ale*]
[EXAMPLE for an IV with "**M**ale" and "**Female**" as levels: ref = "Female" *Does nothing – Intercept was already **Female***]
[EXAMPLE for a DV with "**Full**" and "**A**spirated" as levels: ref = "Full" *Changes intercept from **Full** to **A**spirated*]
[EXAMPLE for a DV with "**Full**" and "**A**spirated" as levels: ref = "Aspirated" *Does nothing – Intercept was already **Full***]

6) ModelName = glmer(**DV** ~ **IVdump** + **IVRandomdump**, data=DataName, family = "binomial")
[An 'observations cannot be < groups' error means there is only 1 data point per RIV level. You can only use a fixed effects model!]

7) summary(ModelName) [Results as a **regression** with log-odds coefficients (β) and **model AIC**]
[*Lower AIC = better model fit*]
[Should you want to convert log-odds coefficients into DV intercept %s, copy/paste the following into R, replacing **X** with the β coefficient, and move decimal over to the right by 2 spaces: exp(**X**)/(1 + exp(**X**))
OR, use the code inv.logit(**X**) after running steps 1 & 2 on the package gtools , and move decimal over to the right by 2 spaces]

[The ANOVA-like output of individual IVs as significant or not significant is available for logistic regression, but cumbersome and not often reported. It is common to simply intuit which IVs are significant based on which IVs have levels with significant β coefficients or which interactions are significant.]

8a) emmeans(ModelName, list(pairwise ~ **IV**), adjust="tukey") [Post-hoc for significant Nominal IV with 3+ levels]
8b) emmeans(ModelName, list(pairwise ~ **IV1*IV2*IV3**), adjust="tukey") [Post-hoc for significant interaction between 2+ Nominal IVs]

9) r2beta(ModelName) [Obtains the **pseudo-r^2** for the model]
[*higher r^2 = more DV variance accounted for by model; .82 = 82%*]

10) plot(table(**IV1**, **IV2**, **IV3**, **DV**), col=T, main="InsertTitleHere") [Quick mosaic plot to visualize results for nominal IVs where IV1 levels will be columns and IV2+ levels will be rows; Use this on an individual IV to confirm what you think the DV intercept is based on the + or – β coefficients from step 7]

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Poisson Regression and Zero-Inflated Poisson Regression (Results appear as coefficients for all minus one levels per IV)

(No restriction on IVs, DV = count data with only 1 level [Your Excel has no DV column! It's just a list of DV occurrences!])

Reference/Intercept for IVs = Alphabetical, See Step 5 to change!

Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

IV; **RIV**: Name of Excel IV column; Name of Excel Random IV column

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)

IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)

IVRandomdump - For each random intercept, use the following notation with parentheses: (1|**RIV1**) + (1|**RIV2**) (etc...)

IVRandomdump - For a random slope for a specific random intercept, replace the "1" for the random intercept codes above with the within-subjects IV(s) of choice: (**IV1**|**RIV1**) + (**IV1**|**RIV2**) + (**IV2**|**RIV2**) (etc...)

1a) install.packages("afex") [if there's a pop-question about binary versions available and installing sources from a package needing compilation, say "no"]
1b) install.packages("lmerTest")
1c) install.packages("emmeans")
1d) install.packages("r2glmm")
1e) install.packages("ggplot2")
1f) install.packages("GLMMadaptive")
1g) install.packages("pscl")

2a) library(afex)
2b) library(lmerTest)
2c) library(emmeans)
2d) library(r2glmm)
2e) library(ggplot2)
2f) library(GLMMadaptive)
2g) library(pscl)

3a) DataName <- read.csv(file.choose(),header=T) [Find your .csv file! This file must consist of a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**]

5) DataName = within(DataName, **IV** <- relevel(**IV**, ref = "NewIVReferenceLevel")) [**OPTIONAL**: Set a not-alphabetically-first level of any one nominal IV to be the reference/intercept in the model (step 8). Run this code multiple times to change the reference / intercept for various nominal IVs, one at a time]

[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Male" *Changes intercept from **Female** to **Male***]
[EXAMPLE for an IV with "**M**ale" and "**F**emale" as levels: ref = "Female" *Does nothing – Intercept was already **Female***]

6a) Distribution = xtabs(~**IV1 + IV2 + IV3**, DataName) [first step toward converting your observations into Frequency Counts]
6b) Distribution [Allows you to see cell counts; “lots” of 0s suggests the use of a Zero-Inflated Poisson Regression]

7a) FreqCounts = as.data.frame(Distribution) [data now has a “Freq” column as the DV, per IV cell]
7b) FreqCounts [shows you the new “Freq” DV per IV cell]
7c) ggplot(FreqCounts, aes(Freq)) + geom_histogram() [visualizes your counts: focus on the left-most (x=0) bar. If this is tallest, you may need a Zero-Inflated P. Regression]

8a) ModelName = glm(Freq ~ **IVdump**, data=FreqCounts, family = "poisson") [Fixed FX Poisson Regression]
8b) ModelName = glmer(Freq ~ **IVdump** + **IVRandomdump**, data=FreqCounts, family = "poisson") [Mixed FX Poisson Reg.]
8c) ModelName = zeroinfl(formula = Freq ~ **IVdump**, data=FreqCounts) [Fixed FX Zero-Inflated Poisson Regression]
8d) ModelName = mixed_model(Freq ~ **IVdump**, random = ~ 1 | RIV, data=FreqCounts, family = zi.poisson(), zi_fixed = ~ **IVdump**, zi_random = ~ 1 | RIV)
[Mixed FX Zero-Inflated Poisson Regression – Replace both “1”s with an IV to affix a random slope. No more than 1 RIV and 1 slope permitted.]

9) summary(ModelName) [Results as a **regression** with log-odds coefficients (β) and **model AIC**]
[*Lower AIC = better model fit*]
[Should you want to convert log-odds coefficients into DV counts, copy / paste the following into R, replacing **X** with the β coefficient: exp(**X**)]

[The ANOVA-like output of individual IVs as significant or not significant is available for logistic regression, but cumbersome and not often reported. It is common to simply intuit which IVs are significant based on which IVs have levels with significant β coefficients or which interactions are significant.]

10a) emmeans(ModelName, list(pairwise ~ **IV**), adjust="tukey") [Post-hoc for significant Nominal IV with 3+ levels]
10b) emmeans(ModelName, list(pairwise ~ **IV1*IV2*IV3**), adjust="tukey") [Post-hoc for significant interaction between 2+ Nominal IVs]

11) r2beta(ModelName) [obtains the **pseudo-r^2** for Poisson Regressions only]
[*higher r^2 = more DV variance accounted for by model; .82 = 82%*]

Model Comparison

(Evaluate which model is better – the one with/without an IV interaction? Random IV? Additional IVs? Poisson or Zero-Inflated Poisson? Etc...)

Legend:
Words - Create your own label, colors show continuity of label
DV or **IV**; **V**; **RIV**: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column

1) For all tests save Within-Subjects ANOVA (for which you should instead just run a mixed effects linear regression), proceed through ModelName = step. Make this your SIMPLER model (fewer IVs, RIVs, interactions, levels for specific IVs, etc.).

2) Hit UP arrow & rename ModelName to ModelNameComplex, adding complexity (more [R]IVs, interactions, zero-inflation, etc.)
[Beyond adjusting the IVs & ModelName, be sure to check if the code before the **DV** needs modification, such as glm vs. lmer, etc.]
[Repeat this step as many times as necessary, i.e., to compare 5 models, complete this step 4 times to create 4 additional models]

3a) vuong(ModelName, ModelNameComplex) [Use to compare Poisson Regression to 0-Inflated-P. Regression: ONLY Fixed FX!]
3b) anova(ModelNameComplex, ModelName, test="Chisq") [Use to compare a Fixed FX to a Mixed FX model: NO 0IP-Reg!]
3c) anova(ModelName, ModelNameComplex, test="Chisq") [Use when both models are either fixed or mixed FX: NO 0IP-Reg!]
[Significant p value - use the more complex / Zero-Inflated model; Non-significant p value - use the simpler / Poisson model]
[If comparing more than 2 models, expand the code by adding in the other model names, separated by commas, before test="Chisq"]

3d) For comparisons involving Zero-Inflated Poisson Regression Models, you must do things manually, via summary(ModelName):

- Calculate the absolute value of the difference between **degrees of freedom (DF)** for each model
- Calculate the absolute value of the difference between **–Log Likelihoods** for each model and multiply by 2
- pchisq(x2LogLik, df=DFdiff, lower.tail=FALSE)

[Significant p value - use the more complex model; Non-significant p value - use the simpler model]

Justin Davidson R Version 4.0.3. justindavidson@berkeley.edu

Variable Rule Analysis (Varbrul, a la Goldvarb) – Step-wise, Fixed Effects Logistic Regression

(Results appear as factor weights for each IV level for each nominal IV, or a log-odds coefficient for any continuous IVs)

(No restriction on IVs, DV = binary [2 levels] nominal/discrete, no random effects)

Reference/Intercept for DV = REVERSE-Alphabetical, See Step 6 to change!

Reference/Intercept for IV = REVERSE-Alphabetical! (Nope, that's not a typo. This ordering is unique to Varbrul.)

Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

DV ; **IV** ; **V** : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)

IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)

0a) Head to **Fixed Effects Logistic Regression** and complete all steps through ModelName = , then return here and continue.

[Create as COMPLEX a model as possible!]

[Complex models have max number of IVs and max interactions]

0b) step(ModelName) [runs the step-wise regression]

[The best-fit model appears at the bottom of the output after "Call:", with interactions noted by **IV1 : IV2** instead of **IV1 * IV2**]

[**IV1 * IV2** is the same as **IV1 + IV2 + IV1:IV2**]

0c) Take note of the **IVdump** for the best-fit model above and use this when you continue here and reach step 7. Quit R and reopen.

1 & 2) Run the set of "library" commands once more on the packages from Fixed Effects Logistic Regression (car, emmeans, etc.)

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]

3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]

3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]

3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**, except in step 5]

5) contrasts(DataName$**IV**) <- contr.sum [Changes default treatment contrasts to sum contrasts in order to obtain valid factor weights for each nominal **IV** level. You must run this code multiple times to include every nominal **IV**, one at a time.]

6) DataName = within(DataName, **DV** <- relevel(**DV**, ref = "OLD**DV**ReferenceLevel")) [**OPTIONAL**: set an alphabetically-first level of the DV to be the reference/intercept in the model (step 8). NOTE THAT THE DV REFERENCE CHANGE IS COUNTER-INTUITUVE: YOU WRITE THE DEFAULT / OLD LEVEL TO CHANGE IT. Additionally, since factor weights will be computed for every nominal IV level, DO NOT reconfigure the IV intercept. Only the DV intercept!

[EXAMPLE for a DV with "**F**ull" and "**A**spirated" as levels: ref = "Full" *Changes intercept from* ***Full*** *to* ***A****spirated*]
[EXAMPLE for a DV with "**F**ull" and "**A**spirated" as levels: ref = "Aspirated" *Does nothing – Intercept was already* ***Full***]

7) ModelName = glm(**DV** ~ **IVdump**, data=DataName, family = "binomial")

8) summary(ModelName) [Results as a **Varbrul regression** with log-odds coefficients (β) and **model AIC**]
[*Lower AIC = better model fit*]

[To convert log-odds coefficients into factor weights, copy/paste the following into R, replacing **X** with the β coefficient: exp(**X**)/(1 + exp(**X**))
OR, use the code inv.logit(**X**) after running steps 1 & 2 on the package gtools]

[The factor weight of the intercept β coefficient is the model's **corrected mean (or input)**, which represents the baseline for DV intercept production (adjust decimal 2 spaces to the right to read it as a %) when no IVs are taken into account. Do not interpret the intercept any further!]

[Nominal IV levels unfortunately appear as a numbered list. To recover what each row name refers to, go in alphabetical order. #1 is the alphabetically-first level for that nominal IV, #2 is second for that nominal IV, etc. The missing level for each nominal IV is alphabetically-last (normally the intercept, but this is a special case.)
To compute the factor weight for each missing level (i.e., do NOT just take the value of the intercept!), calculate the log-odds missing that would make a sum of 0 for that IV, and then compute the factor weight for that value.]
[EXAMPLE: For an IV with levels A, B, and C, with C comprising the intercept and thus being missing, if the β coefficients for A and B are respectively -1.7 and 0.9, then the missing β coefficient of C that would make their total add to 0 is **0.8**. The factor weights for A, B, & C (using the aforementioned formula) are respectively .15 ; .71 ; & .69]

[To compute the p value for the model, note the **NULL** and **RESIDUAL** deviances / Degrees of Freedom at the bottom of this step's output, and:

- Calculate the absolute value of the difference between **degrees of freedom (DF)** for NULL & RESIDUAL
- Calculate the absolute value of the difference between **Deviances** for NULL & RESIDUAL
- Run the following: pchisq(DevianceDiff, df=DFdiff, lower.tail=FALSE)]

[To compute the **negative log likelihood** for the model, take the Residual Deviance and divide it by -2.]
[*Negative Log Likelihood closer to 0 = better model fit*]

9) pR2(ModelName) [Obtains the **pseudo-r²** for the model; the value you want is McFadden!]
[*higher r² = more DV variance accounted for by model; .82 = 82%*]

10) plot(table(**IV1**, **IV2**, **IV3**, **DV**), col=T, main="InsertTitleHere") [Quick mosaic plot to visualize results for nominal IVs where IV1 levels will be columns and IV2+ levels will be rows; Use this on an individual IV to confirm what you think the DV intercept is based on order of β coefficients from step 8]

Variable Rule Analysis (Varbrul, a la Goldvarb) – Step-wise, Mixed Effects Logistic Regression

(Results appear as factor weights for each IV level for each nominal IV, or a log-odds coefficient for any continuous IVs)

(No restriction on IVs, DV = binary [2 levels] nominal/discrete, presence of 1+ random effects)

Reference/Intercept for DV = REVERSE-Alphabetical, See Step 6 to change!

Reference/Intercept for IV = REVERSE-Alphabetical! (Nope, that's not a typo. This ordering is unique to Varbrul.)

Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

IV; **RIV**: Name of Excel IV column; Name of Excel Random IV column

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)

IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2** * **IV3** + **IV2** * **IV4** (etc...)

IVRandomdump - For each random intercept, use the following notation with parentheses: (1|**RIV1**) + (1|**RIV2**) (etc...)

IVRandomdump - For a random slope for a specific random intercept, replace the "1" for the random intercept codes above with the within-subjects IV(s) of choice: (**IV1**|**RIV1**) + (**IV1**|**RIV2**) + (**IV2**|**RIV2**) (etc...)

0a) The step() function does not currently work with mixed effects logistic regression models. Accordingly, what we'll do is run the step() function on a model without random effects (i.e., as a fixed effects logistic regression model), and then with that best model (i.e., that set of IVs), run model comparisons between said fixed effects model and models with the random effects included. To do all this, first complete steps (0a) and (0b) for Varbrul - Step-wise Fixed Effects Logistic Regression (on prior page). Then, head to Model Comparison and complete the steps through (3c) using mixed effects logistic regression models (i.e., the various combinations of your best fixed effects model IVs + all combos of RIVs). Once you have the best combo of Random Effects, finally run Model comparison step (3b) to determine which model of the two to continue with (either the best fixed effects model, or the best random effects model).

0b) Take note of the **IVdump** and **RIVdump** for the best-fit model and use this when you continue ahead and reach step 7. Quit R and reopen.

[The above is complicated, so here's an example. Consider a situation where I have 3 IVs (A, B, and C) and 2 RIVs (X and Y). First, since step() won't work with the random effects, I'll find the best model with just fixed effects. This ends up looking like **step(ModelNameComplex)**, where ModelNameComplex has the IVs as follows: A*B*C. Running this step-wise fixed effects logistic regression could indicate that the best combination of IVs happens to be: A*B + C. Peachy. Now, with this combination of IVs, we'll figure out the best combination for RIVs. Using Model Comparison (step 3c), we'll compare 3 models: RandomX, with A*B+C + (1|X); RandomY, with A*B+C + (1|Y); and lastly RandomXY, with A*B+C + (1|X) + (1|Y). The code would be **anova(RandomX, RandomY, RandomXY, test="Chisq")**. This could tell us that amongst models with random effects, the best one happens to only include Y. Peachy. Now, a final model comparison (step 3b) between the best fixed effects model (A*B+C) and the best mixed effects version (A*B+C+(1|Y)). Whichever model is best, this constitutes the **IVdump** (and possibly **RIVdump**) to write down for later.]

1 & 2) Run the set of "library" commands once more on the packages from Mixed Effects Logistic Regression (afex, lmerTest, etc.)

3a) DataName <- read.csv(file.choose(),header=T, stringsAsFactors = TRUE) [Find your .csv file! File must be a single sheet only!]
3b) head(DataName) [Reminds you of **V** names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY Vs of a #/Quantity have median/mean/max/min stats computed. The remaining nominal Vs should have their levels below them, and how many data points per level there are]
3d) qhpvt(DataName, "**IV1**", c("**IV2**", "**IV3**"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a **V** as DataName$**V**, except in step 5]

5) contrasts(DataName$**IV**) <- contr.sum [Changes default treatment contrasts to sum contrasts in order to obtain valid factor weights for each nominal **IV** level. You must run this code multiple times to include every nominal **IV**, one at a time.]

6) DataName = within(DataName, **DV** <- relevel(**DV**, ref = "OLD**DV**ReferenceLevel")) [**OPTIONAL**: set an alphabetically-first level of the DV to be the reference/intercept in the model (step 8). NOTE THAT THE DV REFERENCE CHANGE IS COUNTER-INTUITUVE: YOU WRITE THE DEFAULT / OLD LEVEL TO CHANGE IT. Additionally, since factor weights will be computed for every nominal IV level, DO NOT reconfigure the IV intercept. Only the DV intercept!
[EXAMPLE for a DV with "**F**ull" and "**A**spirated" as levels: ref = "Full" *Changes intercept from **F**ull to **A**spirated*]
[EXAMPLE for a DV with "**F**ull" and "**A**spirated" as levels: ref = "Aspirated" *Does nothing – Intercept was already **F**ull*]

7) ModelName = glmer(**DV** ~ **IVdump** + **IVRandomdump**, data=DataName, family = "binomial")
[An 'observations cannot be < groups' error means there is only 1 data point per RIV level. You can only use a fixed effects model!]

8) summary(ModelName) [Results as a **Varbrul regression** with log-odds coefficients (β) and **model AIC**]
[*Lower AIC = better model fit*]

[To convert log-odds coefficients into factor weights, copy/paste the following into R, replacing **X** with the β coefficient: exp(**X**)/(1 + exp(**X**))
OR, use the code inv.logit(**X**) after running steps 1 & 2 on the package gtools]

[The factor weight of the intercept β coefficient is the model's **corrected mean (or input)**, which represents the baseline for DV intercept production (adjust decimal 2 spaces to the right to read it as a %) when no IVs are taken into account. Do not interpret the intercept any further!]

[Nominal IV levels unfortunately appear as a numbered list. To recover what each row name refers to, go in alphabetical order. #1 is the alphabetically-first level for that nominal IV, #2 is second for that nominal IV, etc. The missing level for each nominal IV is alphabetically-last (normally the intercept, but this is a special case.)
To compute the factor weight for each missing level (i.e., do NOT just take the value of the intercept!), calculate the log-odds missing that would make a sum of 0 for that IV, and then compute the factor weight for that value.]

[EXAMPLE: For an IV with levels A, B, and C, with C comprising the intercept and thus being missing, if the β coefficients for A and B are respectively -1.7 and 0.9, then the missing β coefficient of C that would make their total add to 0 is **0.8**. The factor weights for A, B, & C (using the aforementioned formula) are respectively .15 ; .71 ; & .69]

[To compute the **negative log likelihood** for the model, take the Residual Deviance and divide it by -2.]

[*Negative Log Likelihood closer to 0 = better model fit*]

[To compute the p value for the model, take note of the **RESIDUAL** deviances / Degrees of Freedom at the top of this step's output. Next, create the following **NULL** model, which notably has NO IVs.

```
ModelNameNULL = glmer(DV ~ IVRandomdump, data=DataName, family = "binomial")
```

Take note once more of the residual deviance and degrees of freedom at the top, but consider them **NULL** values.

- Calculate the absolute value of the difference between **degrees of freedom (DF)** for NULL & RESIDUAL
- Calculate the absolute value of the difference between **Deviances** for NULL & RESIDUAL
- Run the following: `pchisq(DevianceDiff, df=DFdiff, lower.tail=FALSE)`]

9) `pR2(ModelName)` [Obtains the **pseudo-r^2** for the model; the value you want is McFadden!]

[*higher r^2 = more DV variance accounted for by model; .82 = 82%*]

10) `plot(table(IV1, IV2, IV3, DV), col=T, main="InsertTitleHere")` [Quick mosaic plot to visualize results for nominal IVs where IV1 levels will be columns and IV2+ levels will be rows; Use this on an individual IV to confirm what you think the DV intercept is based on the order of β coefficients from step 8]

Step-Wise Regression (Results appear as the best-fit model, indicating which IVs significantly improve the model [vs. their absence])
(Currently only available in R for FIXED effects Linear/Logistic/Poisson/0-Inflated Poisson Models, as well as MIXED Linear])

Legend:
Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.
Words - Create your own label, colors show continuity of label
DV or **IV**; **V**; **RIV**: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column
IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1** + **IV2** + **IV3** + **IV4** (etc...)
IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1** + **IV2 * IV3** + **IV2 * IV4** (etc...)
IVRandomdump - For each random intercept, use the following notation with parentheses: (1|**RIV1**) + (1|**RIV2**) (etc...)
IVRandomdump - For a random slope for a specific random intercept, replace the "1" for the random intercept codes above with the within-subjects IV(s) of choice: (**IV1**|**RIV1**) + (**IV1**|**RIV2**) + (**IV2**|**RIV2**) (etc...)

1) Select the appropriate test and complete all steps through ModelName = [Create as COMPLEX a model as possible!]
[Complex models have max number of IVs, max interactions, max random effects, etc.]

2) step(ModelName) [runs the step-wise regression]

[The best-fit model appears at the bottom of the output after "Call:", with interactions noted by **IV1 : IV2** instead of **IV1 * IV2**]
[**IV1 * IV2** is the same as **IV1 + IV2 + IV1:IV2**]

3) To compute the p value for the best-fit model, note the **NULL** and **RESIDUAL** deviances / Degrees of Freedom at the bottom of the prior step's output, and:

- Calculate the absolute value of the difference between **degrees of freedom (DF)** for NULL & RESIDUAL
- Calculate the absolute value of the difference between **Deviances** for NULL & RESIDUAL
- Run the following: pchisq(DevianceDiff, df=DFdiff, lower.tail=FALSE)]

4a) Hit the UP arrow until the ModelName code (via step 1 above) is displayed [sets up the overwriting of the complex model]
4b) Modify **IVdump** &/or **IVRandomdump** to match the (R)IVs displayed in best-fit model from step 2 [sets up best-fit model]
[Feel free to express interactions via * rather than : . See [note] under step 2, above]

5) Return to the original test steps and finish them as normal, starting with the summary(ModelName) step. [stats on best-fit model]

Chi-Squared Test (Results appear as a single IV with DV proportions that are or are not significantly different across IV levels)
(Nominal/Discrete IV [1 at a time!], DV = count data, no restriction on number of levels, No random effects)

Analysis: Is the distribution/proportion of DV consistent across IV levels (i.e., are all rows equal?) Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

1) install.packages("rcompanion") [if there's a pop-question about binary versions available and installing
2) library(rcompanion) sources from a package needing compilation, say **no**]

3a) **For DVs with 2+ levels**, envision your data according to the following schematic:

	Counts of DV: Level1 (column1)	Counts of DV: Level2 (column2)	Counts of DV: Level3 (column3)
Level1 of IV: (row1)	500 A	400 B	250 C
Level2 of IV: (row2)	550 D	455 E	305 F
Level3 of IV: (row3)	550 G	455 H	305 I
Level4 of IV: (row4)	550 J	455 K	305 L

3b) **For DVs with only 1 level**, envision your data according to the following schematic:

	Level 1 of IV: Level1 (column1)	Level 2 of IV: Level2 (column2)	Level 3 of IV: Level3 (column3)
Observations of DV: (row1)	500 A	400 B	250 C
Evenly Distributed DV (row2)	Sum of row1 values divided by # of columns D (1150/3)	Sum of row1 values divided by # of columns E (1150/3)	Sum of row1 values divided by # of columns F (1150/3)

4) ModelName = matrix(c(**A,B,C,D,E,F,G,H,I,J,K,L**), nrow = MaxRow#, ncol= MaxColumn#, byrow= TRUE, dimnames = list(c("**NameOf1stRow**", "**NameOf2ndRow**", "**NameOf3rdRow**", "**NameOf4thRow**"), c("**NameOfColumn1**", "**NameOfColumn2**", "**NameOfColumn3**")))

[add or remove rows/columns names/values as necessary]

5) ModelName [visualizes your data; checks if your code was correct]

6a) chisq.test(ModelName) [Only use if all cells have at least **5 or more** tokens]
6b) fisher.test(ModelName, workspace = 2e+9) [Only use if any cell has **4 or fewer** tokens; change 9 to different number if there is an error about workspace size]

7) pairwiseNominalIndependence(ModelName, gtest = FALSE) [Post-hoc to compare every pair of rows. Ignore all values without adjustment. For any pair that includes a cell with a token count of **4 or fewer**, use **p.adj.Fisher**. For pairs where all cells have **5 or more** tokens, use **p.adj.Chisq**.]

Unfortunate Problems

As commands are generally introduced to R via the installation and subsequent calling up of packages, the average R-user relies on independent researchers and other coding-inclined folks to constantly be creating and eventually updating them. Beyond the inevitable issue of package authors not keeping their packages up to date as new editions of R are released, another unfortunate problem is that not all packages perform tasks in parallel manners.

To give an example, consider the discrepancy in fetching the r2 for a fixed effects linear regression vs. a mixed effects one. Ideally, one might expect that a single "get r2"-like command might be used to retrieve this statistic for both tests. However, for the former test, the "summary" command won't compute this statistic for a glm object (step [6]), so we must create a dummy "lm" object (steps [11a] and [11b]) simply to have the r2 presented to us. For the latter, an entirely novel package (r2glmm) is required (step [9] – and no, the r2beta command that the r2glmm package offers does not work for fixed effects regressions – that would be too easy!). Overall, this is simply a case of inconvenience rather than a true problem (i.e., while having to juggle multiple creative strategies [codes] to get the same value for different tests is annoying, in the end r2 is still obtainable for each type of test).

True problems are cases where the available packages simply cannot give you what you need, leaving you stuck waiting for somebody to develop a package that does (or I suppose, you could always become a stats/coding expert and create your own package to address the issue!). Below is a list of gaps that I am aware of with this R tutorial, that is, cases where you currently (with this tutorial) CANNOT perform the listed tests. Some of these gaps may be cases where there currently is no package out there with said functionality; others (hopefully more likely) are cases where I simply am not aware that a package exists and could perform the task.

Should you be aware of any packages that allow you to perform any of the following, please let me know and I'll check them out!

- Multinomial regression (for nominal DVs that have three or more levels) [I know packages are readily out there, I simply haven't dabbled enough with them to select which set to showcase here. Suggestions welcome!]
- ANOVA-like output for a mixed effects logistic regression
- ANOVA-like output for any type of Poisson regression
- Mixed effects Zero-Inflated Poisson regression that allows for more than 1 random effect/slope to be included
- Model comparisons involving (mixed effects) Zero-Inflated Poisson regression [not a true gap, since I discuss how to pull the relevant test statistics and manually do some math to compare them... but ideally we should be just as able to use "vuong" or "anova" with these regression models as with all the rest, and currently we can't.]
- Step-wise regression for mixed effects logistic models [I discuss a way for dealing with this under step (0a) of Varbrul: Step-wise Mixed Effects Logistic Regression, though it's still a get-around for a problem that shouldn't exist]
- Step-wise regression for mixed effects Poisson models