```
######### Anna Hargreaves 2020
################################################################
######### NOTE FOR STUDENTS (delete this part from your scripts once you've read it)
##############

# How to use this script #######
#use in combination with the word doc guide to setting up R script (Annas webpage)
#this code is a starting place and a guide.  It contains the most common commands I use in creating a data analysis script but does not always describe what each command does in detail.  Dont expect this to be all you need to write your own code - You will need to look up commands you are not familiar with in R's help files or online, and look for online blogs/tutorials for examples of how to use them if you need. Good strategies to solve R problems and the order in which to deploy them:
#__1) look up R's helpfile. you can do this by typing one or 2 question marks in front of the command you need help with.  eg ?summary
#__2) search online for the command you are trying to understand  (eg 'R summary examples') or the problem you are trying to solve (eg 'R how to I find unique occurrences in a variable' or 'R how to assess overdispersion poisson glm') or the warning/error message (just cut and paste the general part (ie remove anything specific to your code/data) into a search engine). Expect to have to consult multiple pages, refine your search term, dig a bit.  We all do this all the time when coding in R - its just part of the process.  If you find a page that is really helpful paste the url into the comments in your code in case you want to go back to it (or explain to me why you made a certain coding decision).  NOTE: while doing this you will find lots of people promoting packages that do specific functions. Avoid using extra packages whenever possible - look for a solution in base R first
#__3) there is lots of archived code on my website, and there's a good chance that your project is similar to another project someone else did in the lab.  Find their code and use it as a guide
#__4) if this is your first R code then there are definitely other students in the lab with more coding experience. Ask them (but only after doing your own homework via steps 1-3 above). If there is another student doing similar code organize to work together once a week or something to help trouble shoot and share tips and bond
#__5) if you are stuck on something small, email me the question.  make it as specific as possible
#__6) have a list of coding questions ready for our weekly meeting.  It is normal to spend several full meetings just coding together, but this should be to break log jams and think about tricky issues in analyses (ie it shouldnt be the main way your coding gets done)

# Organizing your R scripts ########
#click on the 'Document outline' button at the top right of this window (bunch or paraellel horizontal lines). This lets you set up primary, secondary and even tertiary headings (dont use more than 3 levels - document outline feature is not THAT good). Having 4 #s or 4 -s in a row creates a heading (ie that line appears on the outline). To make it easy for you to distinguish headings, indent them manually (supposedly R studio does this automatically but that has never worked for me). Use all caps for primary headings, and underscores or some other system to indent secondary and tertiary headings. EG:
# PRIMARY HEADING ######################
#_Secondary heading --------------------
#__Tertiary heading --------------------
################################################################################################
#


################################################################################################
#
## PROJECT TITLE (eg R SCRIPT TEMPLATE FOR HARGREAVES LAB)########
## Author names & Year (eg Pat Smith (Honours thesis) & Anna Hargreaves 1998)
##
## Brief sentence about what this script does (data cleaning? analyses? map generation?)
```

```
## Brief sentence about associated scripts if there are any, including name of associated
script
## Eg This script compiles map shapefiles and calculates the Canadian range area and
proportion range in Canada. Analyses and Figures are done in a separate script called 'R
Smith analyses 98 04 26'
############################################################################## ###

#call in all packages needed throughout the script and say what they're for
library(car) #for Anova tables
library(MuMIn) #for dredge (hopefully you wont need this)...
library(lattice) #for bwplots (you might use ggplot instead, I use lattice cause its easy
and predates ggplot)
library(lme4) #for mixed models
library(emmeans) #for lsmeans
library(visreg) #for plotting model results


# DATA EXPLANATION ##########################################

# Briefy explain structure of the data. Eg. 24 hr seed predation trials on sunflower and oat
seeds.  Some sunflower seeds were caged. 30 depots per site spaced >10 m apart.

#Variable names-----------------------------------------
# column name = explanation
# column name = explanation
# column name = explanation
#etc

#Data notes & Corrections
#(here keep list of any major corrections made to raw data that affect analysis results).
Eg:
#July 2 2019 - realized 2 pages of data from NK-M had been entered as NK-H as page had been
mislabelled.  Fixed in data


# DATA PREP #################################################

#call in data and give it a short but descriptive name.
#If you are going to change it a bunch call it something that makes it obvious that this is
the original data frame.
#call in by file name so you always know which data were used in a given script
#a shortcut to get the file name in RStudio is to go to Files (bottom right panel of
RStudio), chose the file, click 'Import dataset' (which opens a new window), then copy the
filename from the 'Code preview' pane in the bottom right of the new window (dont actually
use this point and click method to open the file)
seed.orig <- read.csv(file="data/LA transplant data raw wtaxonomy 19 07 23.csv")
summary(seed.orig) #always look at data first

#get rid of long columns you don't need---------------
#eg say you are doing a literature review and you have a column where you pasted the full
abstract of each paper. The long text in that column is going to make it hard to see your
data quickly using 'summary', and you'll never need it for analyses themselves, so you can
get rid of it early. once you start getting rid of columns make a new dataframe
seed1 <- seed.orig
seed1$Abstract <- NULL
#you can get rid of multiple columns in one command by stacking them:
seed1$Abstract <- seed1$title <- NULL

#look at any variables that dont make sense ------------
```

```
summary(seed1)
#eg say theres a column whose vaules should be between 0 and 5, and summary shows that the minimum valuw is -1. start by looking at those wierd values
seed1[seed1$variable <0,] #this will print all the rows of the data where variable <0.  if there are just a few thats usually enough to figure out the problem. but if there are a lot, you can create a dummy dataset that you can look at more closely
test <- seed1[seed1$variable <0,]; test <- droplevels(test)
summary(test) #now you can look at which levels of other categories are involved in the wierd data

#make new columns -----------------
#as much as possible have your raw data in excel and your data manipulation in R
#for all examples below look up the command itself for more details on how they work
#you can create derived columns (eg sums of other columns, means of other columns)
seed1$totpred <- seed1$thing1 + seed1$thing2
#you can create summary categorical variables by pasting columns together using 'paste'. #note wrapping in 'as.factor' makes sure the resulting column is seen as a factor with levels
seed1$site.elev <- as.factor(paste(seed1$site, seed1$elev, sep='.'))
#you can create new categorical summary variables using ifelse statements. These have 3 parts: 1) if this thing is true, 2) do this, 3) else do this. Imagine you have birds, mammals, reptiles, insects, and plants, and you want to create a new variable that groups them into Animals or Plants
seed1$higher.taxon <- ifelse(seed1$taxa=='plants', 'plants', 'animals')
#to create more complex variables you can nest ifelse statements, where the 3rd part (the else) is another ifelse. Eg say you want to group the above taxa into vertebrates, invertebrates, or plants
seed1$higher.taxon <- ifelse(seed1$taxa=='plants', 'plants',
                                ifelse(seed1=='insects', 'inverts', 'verts'))

#create final dataset---------------
#usually you will want to work with a final dataset that is clean, with columns ordered logically, with informative but short column names, and no long columns that you dont actually need.  Do this in two parts:
summary(seed1) #remind yourself what your data look like

seed <- seed1[,c('variable1', 'variable2', 'variable 3', etc)] #list the columns you want to keep in the order you want them
dim(seed1); dim(seed2) #make sure the relative sizes of your old vs new datasets look right

colnames(seed) <- c('var1', 'var2', 'var3', etc) #copy and paste the 'c' part of the command above, and adjust any names you want to shorten or clarify
summary(seed) #make sure looks right


# ANALYSES ################################################

#1st write out question as a header (but briefly so makes sense in document outline) ---
#and make notes about model structure (what you're testing), data structure, and data to include/exclude
#the most interesting and trickiest modeling decisions are often about which, if any, interactions to include between predictors. This is something you and I should talk about in our meetings. An interaction means that the effect of one predictor depends on the other. In the first example below, it might mean that abiotic drivers are more supported than biotic drivers at latitudinal range limits, but there's no difference (or the opposite pattern) at elevation RLs. In the second example, including an interaction (height ~ age*sex) means that you expect the difference in height between sexes to depend on the kids age.  eg maybe boys and girls and the same height on average when they are born, but boys grow faster so are taller than girls on average by the time they are 17.  Not including an interaction (height
```

```
~ age + sex) means that you expect height to dpeend on age, and on sex, but that the
difference among the sexes is constant across ages (eg boys are always taller than girls)
#examples:

#Q1) ar cool RLs, are biotic or abiotic drivers supported more often and does this vary bwn
lat & elev RLs? ---------------
#response = 'Driver.supported01' = is a factor supported, Y=1 or N=0 (so need binomial
model)
#predictors = 'Driver.type' (Biotic or Abiotic) and RL.Lat.Elev (either 'latitudinal' or
'elevational')
#data structure: 1 row per taxon X driver
#exclude studies of multiple genera that couldn't be disentangled in data

#or

#Q1) does human height during growing years vary with age and sex? ----------------------
#response = height (inches). try normal distribution
#predictors = age (years, have data from 0 to 17) & sex (category: F or M)
#exclude data collected in 1989 as measurement protocol differed so not comparable


#2nd plot your data.---
#Plot exactly the data you are planning to model (eg if you are excluding data from model,
exclude it from the plot)
#ggplot is great for this but I don't work with it regularly so its more annoying to code
for me, so I use lattice
#examples:

bwplot(response ~ predictor1, data=X) #creates box plot. requires one of your predictors is
a category. specific example:
bwplot(height ~ age | sex, data=kidsize[kidsize$year!=1989])


#3rd write out your full model (all the terms you want to test) ---
model.name <- model.command(response ~ predictors, family=X, data=X)
#chose a model name that is short but informative, ending with a period. After the period
put a 1to2 letter indicator of the distribution the model uses:
#lm = normal (data continuously distributed, can be negative, residuals have 1 peak and are
~ symetrical),
#p  = poisson = for count data (integers, cant be <0). Has characteristic that mean=variance
(often violated in biological data),
#qp = quasi poisson (assumption of mean=variance is relaxed - deals with over or under
dispersion)
#nb = negative binomial = for count data but has extra parameter to estimate dispersion
#b  = binomial = data are from independent trials where each trial has only 2 outcomes (eg 0
or 1, Yes or No). Think coin toss. like poisson has specific expectation re mean and
variance
#qb = quasi binomial (assumption re variance and mean is relaxed - deals with over or under
dispersion)

#As you reduce the model, you will add notes re which predictors have been removed (and
change distribution code if you had to switch distributions)
#examples

coolAvB.b <- glm(Driver.supported01 ~ Driver.type*RL.Lat.Elev,
                 family=binomial,
                 data=allRL[allRL$RL=='cool',])
                #data=allRL, subset=RL=='cool') #another way of coding the line above to
subset the data
```

```
height.lm <- glm(height ~ age*sex, #you will probably never be working with normally
distributed data so we'll stop this example here
                 family=gaussian,
                 data=kidsize[kidsize$year!=1989])

#4th if you're using a poisson or binomial error distribution, you have to check to see
whether your data are meeting their assumptions about how widely dispersed your data are.
see: https://data.princeton.edu/wws509/r/overdispersion It is nto always clear whether you
should do this on your first model, or your final model.  I was taught final model, but if
I'm starting with a simple full model I sometimes test this right away.  If your data are
way overdispersed compared to what the model expects, you can get inflated significance
estimates.  Over dispersion is VERY COMMON in biological data

#4a) FOR GLMS
#if you're using a glm (no random effects), you can eyeball dispersion by comparing the
residual deviance to the residual df in summary (almost at the bottom of output). They
should be roughly equal
coolAvB.b <- glm(Driver.supported01 ~ Driver.type*RL.Lat.Elev,
                 family=binomial,
                 data=allRL[allRL$RL=='cool',])
    summary(coolAvB.b) #resid dev >> resid df so overdispersed

#if you have overdispersion, say so in notes and switch distributions (to quasi poisson or
negative binomial if you have count data, or quasi binomial if you have binary data) #use
quasi binomial distribution to deal with overdispersion (note the residual deviance & df
wont change, so you can no longer use them to assess dispersion)
coolAvB.qb <- glm(Driver.supported01 ~ Driver.type*RL.Lat.Elev,
                 family=quasibinomial,
                 data=allRL[allRL$RL=='cool',])

#4b) FOR MIXED MODELS (glmers)
#if you're using a mixed effects model you can use Ben Bolker's (GLMM guru) formula for
calculating overdispersion. Ben Bolker gives code here:
https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#overdispersion

overdisp_Bolker <- function(model) {
    rdf <- df.residual(model)
    rp <- residuals(model,type="pearson")
    Pearson.chisq <- sum(rp^2)
    prat <- Pearson.chisq/rdf
    pval <- pchisq(Pearson.chisq, df=rdf, lower.tail=FALSE)
    c(chisq=Pearson.chisq,ratio=prat,rdf=rdf,p=pval)
}

coolAvB.mb <- glm(Driver.supported01 ~ Driver.type*RL.Lat.Elev + (1|method), #part in
brackets = random intercept. note use mb to indicate mixed model binomial
                 family=binomial,
                 data=allRL[allRL$RL=='cool',])
    overdisp_Bolker(coolAvB.mb) # if signif means have signif dipersion problem

#you cant use quasi distributions in mixed models (yet), so if you have overdispersion you
have to either switch to a glmer.nb (for count data), or include an individual-level random
effect (look those up)


#5) assess the importance/significance of predictors in your model.
#In some cases you might want to drop non-significant terms. model simplification is
controversial, so this is something you and I should discuss. My general rule of thumbs are:
```

```
#__start with biggest interactions (this is a rule, not a rule of thumb). eg if you have a 3 way interaction sex*age*shoesize you cannot get rid of two-way interactions (eg age*sex) first. You also cant assess the importance of signle terms involved in signif interactions
#__take out NS interactions UNLESS they are the specific thing of interest that I'm testing, as they can complicate assessing the significance of individual predictors in the model, and make your model way more complicated (ie cause convergence problems, use up power)
#__leave in all factors that are integral to the study design, even if they are not significant
#DO NOT use summary to assess term significance - the more complex your model the more impossible it is to understand the summary output or to figure out the estimates. summary is good for getting AIC values and checking dispersion, but thats all I use it for
#Anova: a great tool is Anova from the car package. It is not running an ANOVA on your data, it is showing your model results in an Anova table, calculating sums of squares. Anova sums of squares can be calculated in different ways, generally called type 1, 2, or 3 tests. There are small holy wars about which is better, but I want you to use type3 tests, where the results can handle interactions and dont depend on which factor you write first in your model (if you want to learn more read http://md.psych.bio.uni-goettingen.de/mv/unit/lm_cat/lm_cat_unbal_ss_explained.html). The table will give you a Chi-squared test statistic, its degrees of freedom, and the associated p-value. note that you can't use Anova for mixed models or negative binomial models

#EXAMPLE where Anova is appropriate and interaction is not significant

summary(allRL)
coolAvB.qb <- glm(Driver.supported01 ~ Driver.type*RL.Lat.Elev,
                  family=quasibinomial,
                  data=allRL[allRL$RL=='cool',])
    Anova(coolAvB.qb, type='III')

#here is an example output:
# Analysis of Deviance Table (Type III tests)
#
# Response: Driver.supported01
#                         LR Chisq Df Pr(>Chisq)
# Driver.type               30.385  2  2.524e-07 ***
# RL.Lat.Elev                1.837  1     0.1753
# Driver.type:RL.Lat.Elev    3.890  2     0.xyz     (I made this up)
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# The bottom line (Driver.type:RL.Lat.Elev) shows the interaction is NS, so you could remove it. Let's do that in this case

coolAvB.qbnoX <- glm(Driver.supported01 ~ Driver.type + RL.Lat.Elev, #change model name to indicate whats different. noX = no interaction
                  family=quasibinomial,
                  data=allRL[allRL$RL=='cool',])
    Anova(coolAvB.qbnoX, type='III')
#example Anova output: support varies significantly among drivers but not between elev and latitudinal limits
# Response: Driver.supported01
#             LR Chisq Df Pr(>Chisq)
# Driver.type  29.3913  2  4.147e-07 ***
# RL.Lat.Elev   2.4065  1     0.1208

#post-hoc comparisons for categorical predictors. Driver type has 3 levels - how do you know which are different from each other? Or what levels of support any of them had? You do a post-hoc test. I like least-squared means tests. if you just want to estimate the means you do:
```

```
    lsmeans(coolAvB.qbnoX, ~ Driver.type) #estimates means only
    lsmeans(coolAvB.qbnoX, ~ Driver.type, type='response') #estimates means backtransformed
to scale of original data
    lsmeans(coolAvB.qbnoX, pairwise ~ Driver.type) #estimates means and gives pairwise
significance tests

#plot model results (not raw data). use visreg. this lets you see your model estaimtes for
Driver type while also accounting for effeects of any other predictors in your model
    visreg(coolAvB.qbnoX, xvar='Driver.type', type='conditional')


#EXAMPLE where Anova is not appropriate (eg mixed model, nb models) AND where interaction is
significant.  To assess interaction, do likelihood ratio tests by hand. to do so, run your
full model, run teh model without the term of interest, and compare the two using the
'anova' command and a Chisquare test.  For models where Anova is appropriate, results
between the two methods should be exactly the same.

coolAvB.qb <- glm(Driver.supported01 ~ Driver.type*RL.Lat.Elev,
                  family=quasibinomial,
                  data=allRL[allRL$RL=='cool',])

#check significance of interaction
coolAvB.qbnoX <- glm(Driver.supported01 ~ Driver.type + RL.Lat.Elev, #change model name to
indicate whats different. noX = no interaction
                  family=quasibinomial,
                  data=allRL[allRL$RL=='cool',])

    anova(coolAvB.qb, coolAvB.qbnoX, test='Chisq') #compare the two mdoels using Chisq test
    #Example output - shows that interaction is significant (ie full model is significantly
better than reduced model)
    # Analysis of Deviance Table
    #
    # Model 1: Driver.supported01 ~ Driver.type * RL.Lat.Elev
    # Model 2: Driver.supported01 ~ Driver.type + RL.Lat.Elev
    #   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
    # 1       947     1226.0
    # 2       949     1260.1 -2  -34.105 4.374e-08 ***

    #proceed to lsmeans and visreg. need to incorporate significant interaction into these
    lsmeans(coolAvB.qbnoX, pairwise ~ Driver.type | RL.Lat.Elev) #estimates mean support for
each driver type for each type of RL. constrasts will tell you whether support varies among
driver types at latitudinal limits, and whether support varies among driver types at elev
limits
    visreg(coolAvB.qbnoX, xvar='Driver.type', by ='RL.Lat.Elev', type='conditional')


# PLOTTING ########################################################################

#After your analyses you need a section of code that generates your publication-quality
plots.  We will work on these together as figure design is important. As always, if you're
doing a variation on an already-published figure, start with the code that generated that
figure and adjust it to suit your needs.
```