Best walkthrough of stats meaning from initial data to regression:<http://varianceexplained.org/RData/code/code_lesson3/>

Fitting all models (lm, glm, anova, etc):<https://www.zoology.ubc.ca/~schluter/R/fit-model/>

Generalized Linear Models w/R examples:<http://www.stat.columbia.edu/~martin/W2024/R11.pdf>

Logistic Regression:

Simple complete walkthrough: <http://www.cookbook-r.com/Statistical_analysis/Logistic_regression/>

Another good explanation:<http://www.ats.ucla.edu/stat/r/dae/logit.htm>

Regression Tutorials/Practice Problems:<https://regressionmodeling.wordpress.com/>

Detailed Description of Linear Regression for Prediction using an example:

<https://adcalves.wordpress.com/tag/r/>

Summary:

1. How Find a line of best fit
2. Making Predictions & checking accuracy of predictions with
   1. r-squared (how much of y is explained by x) and
   2. The Standard Error of Residuals is just the standard deviation of the residuals, it tells you how wrong the regression model is, on average, **using the units of the response variable**
3. Multiple Linear Regression. lm() ‘Interactions’ x1:x2
   1. calculate Pearson’s correlation between X1 and X2 (age and height)
      1. cor.test(cars$mpg,cars$wt): cor.test() provides a lot more info than :cor(age, height, method=”pearson”)
      2. A high correlation between X1 and X2 suggests that the effects from age and the effects from height are somewhat bounded together. If there is high collinearity between X1 and X2 that means we SHOULD NOT directly interpret the slope (eg Age) as the the effect of age on lung capacity adjusting/controlling for height.
   2. Test “Interactions” directly using the lmtest() package. You can either use X1:X2, or also use the multiplication sign as well, X1\*X2. (e.g.) model <- lm(Y~X1+X2+X1\*X2) . **This model is sometimes called a multiplicative model, and suggests the the effect that X1 has on Y is dependent on values of X2**.A Liklihood Ratio Test, lrtest(), directly compares 2 models (1 w/interaction included, and 1 w/o interactions) to determine if the interaction between confounding variables is significant. The Null H is that the interaction doesn’t matter, a p-value below .05 would be cause to reject the Null H and say the interaction matters and include y~ X1 + X2 + X1:X2 in our model.
      1. install.packages("lmtest")
      2. library(lmtest)
      3. lrtest(fit\_interaction, fit\_non\_interaction)
      4. fit\_non\_interaction <- lm(mpg ~ cyl + wt, mtcars) # multiple predictors that don’t interact w/eachother
      5. fit\_interaction <- lm(mpg ~ cyl + wt + cyl:wt, mtcars) # multiple predictors that DO interact
         1. # The most important step above here + cly:wt, this is including the interaction between cly & wt
      6. lrtest(fit\_interaction, fit\_non\_interaction)
         1. low p.value means the interaction matters and should be included in the model. High p.value means the opposite
4. Regression Diagnostics. This can be done very pratically w/the 4 plots generated by plot(yourModel). Overall, the four plots can be used to diagnose specific problems. The upper left plot shows whether the wrong model was fitted (e.g., a line versus a parabola). The upper right plot shows whether the residuals are normally distributed. The lower left plot shows whether the data are homoscedastic(does residual value change as a function of x? ie are errors larger, on average, as x increases/decreases). The lower right plot shows whether there are influential outliers
   1. par(mfrow = c(2,2)) #see all 4 plots at once
   2. plot(model)
   3. Here’s a link to the best description/understanding of the plots: <http://strata.uga.edu/6370/rtips/regressionPlots.html>
5. Generalized Linear Models
   1. fit <- glm(F~x1+x2+x3,data=mydata,family=binomial())
   2. The key element is to describe any given model in terms of it's **link function** and it's **variance function**.The variance function describes the relationship between the mean and the variance of the dependent variable. This allows the proper calculation of the variance (and everything that depends on it) under non-normal conditions. The link function describes the (usually) non-linear relationship between the mean of the dependent variable and the linear right hand side.
   3. Types of GLMs/distributions:
      1. **Logistic Regression**: useful when you are predicting a binary outcome from a set of continuous predictor variables. Outcomes that have two possible outcomes are Bernoulli outcomes. These are often measured as 1/0 outcomes for each trial, or the number of "successes" per "number of trials" for different populations at the various levels of X.
         1. There is a direct relationship between the coefficients and the odds ratios. First, let's define what is meant by a logit: A logit is defined as the log base e (log) of the odds, logit(p) = log(odds) = log(p/q)
         2. The end result of all the mathematical manipulations is that the **odds ratio can be computed by raising e to the power of the logistic coefficient**
            1. exp(successVarCoef)/exp(failVarCoef) For binary outcome w/continuous predictors
            2. Or → exp(binaryPredictor1)/exp(binaryPredictor2) for binary outcome w/categorical predictors
      2. **Poisson**: variables that are measured as counts. Such variables cannot have negative values, and hence are not normal. Poisson regression is useful when predicting an outcome variable representing counts from a set of continuous predictor variables.If you have overdispersion (see if residual deviance is much larger than degrees of freedom), you may want to use quasipoisson() instead of poisson().
         1. Count outcomes were, historically, viewed as the outcomes of Poisson processes. Naturally occurring count data, however, often displays overdispersion due to correlated errors in time or space, or other forms of non-independence of the observations.
         2. For such cases, the negative binomial distribution is a widely used alternative. One approach is to use Poisson, but adjust the standard errors. Alternatively, use the negative binomial.
      3. **Analysis of Survival Time:** or time until an event is commonly assumed to have an underlying exponential or gamma distribution (see also other techniques for analyzing survival data).
         1. While generalized linear models are typically analyzed using the glm( ) function, survival analyis is typically carried out using functions from the **survival package()** . Scroll down on this page for examples:<http://www.statmethods.net/advstats/glm.html>
      4. **Transformation of the Chi-square** distribution: analyzing the variance of an outcome under different X, rather than the mean outcome under different X. For such models, a transformation of the Chi-square distribution, where Y is the variance of the outcome, are commonly used.
   4. <http://faculty.ucr.edu/~hanneman/linear_models/c10.html> This site supports tutorial instruction on an linear models, based on Littell, et al. (2002) SAS for Linear Models.The resources for this chapter (10) and others are fantastic. They describe the material well and walk through examples from the book.

As in lm, it may be useful to order the different groups (categories of A) so that a control group is first, followed by treatment groups. This can help interpretation of the model parameters. For example, if there are 4 groups in A and group "c" is the control group, set the order of the group

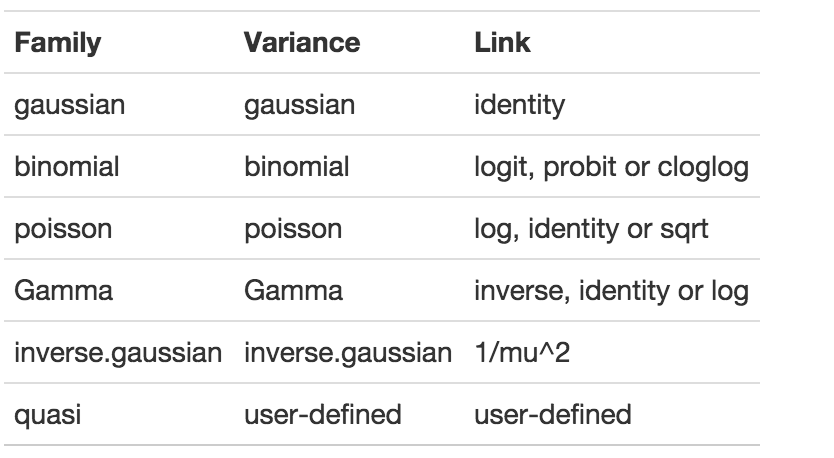
A <- factor(A, levels=c("c","a","b","d"))

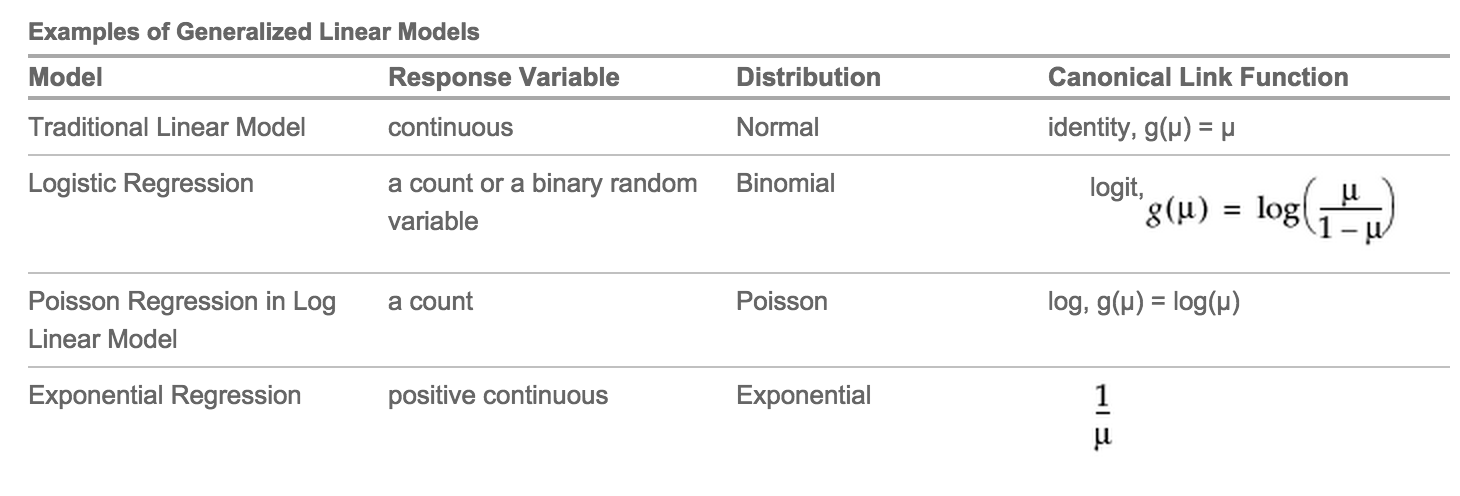
You can also change the way that R models the categorical variable. **For example, a useful approach is to fit a model in which the parameters are the group means (in glm these means are fitted on the logit or log scale before being back-transformed to the original scale). To accomplish this, refit the linear model with a "-1" in the formula, which leaves out an intercept.**

z <- glm(y ~ A - 1, family = binomial(link = "logit"), data = mydata)

z <- glm(y ~ A - 1, family = poisson(link = "log"), data = mydata)

**Mapping Model type to Variance and Link Functions**





**Interpreting Binomial/Logistic Regression:**

<http://www.ats.ucla.edu/stat/sas/faq/oratio.htm>

Let's begin with probability. Let's say that the probability of success is .8, thus p = .8

Then the probability of failure is: q = 1 - p = .2

The odds of success are defined as: odds(success) = p/q = .8/.2 = 4, that is, the odds of success are 4 to 1.

The odds of failure would be odds(failure) = q/p = .2/.8 = .25.

There is a direct relationship between the coefficients and the odds ratios. First, let's define what is meant by a logit: A logit is defined as the log base e (log) of the odds,

[1] logit(p) = log(odds) = log(p/q)

Logistic regression is in reality ordinary regression using the logit as the response variable,

The end result of all the mathematical manipulations is that the **odds ratio can be computed by raising e to the power of the logistic coefficient**

exp(successVarCoef)/exp(failVarCoef) For binary outcome w/continuous predictors

Or → exp(binaryPredictor1)/exp(binaryPredictor2) for binary outcome w/categorical predictors

**Model Selection and Deviance**

An important aspect of generalized linear modeling is the selection of explanatory variables in the model. Changes in goodness-of-fit statistics are often used to evaluate the contribution of subsets of explanatory variables to a particular model. The deviance, defined to be twice the difference between the maximum attainable log likelihood and the log likelihood at the maximum likelihood estimates of the regression parameters, is often used as a measure of goodness of fit. The maximum attainable log likelihood is achieved with a model that has a parameter for every observation.

One strategy for variable selection(which predictive, x-value, variables to include) is to fit a sequence of models, beginning with a simple model with only an intercept term, and then include one additional explanatory variable in each successive model. You can measure the importance of the additional explanatory variable by the difference in deviances or fitted log likelihoods between successive models.

glm(formula, family=familytype(link=linkfunction), data=)

fit <- glm(F~x1+x2+x3,data=mydata,family=binomial())

summary(fit) # display results

confint(fit) # 95% CI for the coefficients

exp(coef(fit)) # exponentiated coefficients

exp(confint(fit)) # 95% CI for exponentiated coefficients

predict(fit, type="response") # predicted values

residuals(fit, type="deviance") # residuals

**Example Questions/Solutions**

From:<http://www.stat.columbia.edu/~martin/W2024/R11.pdf>

0.We are interested in determining the probability that a family purchases a new

car given their income and the age of their oldest car.

#notes: new = 0 or 1, bought car or not. income is family income in thousands of $. age is age of the oldest car in the family.

results <- glm(new ~ income + age, family=binomial)

##Coefficients:

(Intercept) income age

-4.73931 0.06773 0.59863

According to the output, the model is logit( i) = -4.74 + 0.068\*income + 0.60\*age.

#After fitting the model, we can test the overall model fit and hypothesis regarding

a subset of regression parameters using a likelihood ratio test (LRT). we can compare our model with a reduced model that only contains an intercept term.

> results.reduced <- glm(new ~ 1, family=binomial) #only intercept term included

> anova(results.reduced,results, test="Chisq")

#The likelihood ratio test statistic is 2=8.298 with a p-value=0.016. Hence, we have relatively strong evidence in favor of rejecting Null H, income and age matter.

As a next step, we perform tests on the individual regression parameters.

> summary(results)

# Low p.values for a variable means that the variable is significant and should probably be included. High p.values means the opposite.

To compute how the odds of purchasing a car changes as a function of income

use the commands:

> exp(coef(results))

(Intercept) income age

0.008744682 1.070079093 1.819627221

#We see that the odds ratio corresponding to income is 1.070 (95% CI: (1.013,

1.131)). This implies that if we fix the age of the oldest car, increasing family

income by one thousand dollars will increase the odds of purchasing a new car

by 0.07.

1. Categorical Predictor, Categorical/binary Outcome**.**

**You must use glm(outcome ~ predictor -1)** in order to see the coeffecients for both/all groups of the predictor. You will need both of these coeffecients if you want to calculate the Odds Ratio.

**y~0+x or y~ -1 +x or y~ x-1 is linear regression through the origin**

Fit a logistic regression model with autolander (variable auto) use (labeled as "auto" 1) versus not (0) as predicted by wind sign (variable wind). Give the estimated odds ratio for autolander use comparing head winds, labeled as "head" in the variable headwind (numerator) to tail winds (denominator).

# convert outcome to 0 = noauto, 1 = auto

shuttle2<-shuttle

shuttle2$use2

> shuttle2$use <- factor(shuttle$use, levels = c("auto", "noauto"), labels = c(1, 0))

fit <- glm(use ~ **wind - 1**, data = shuttle, family = binomial)

summary(fit)$coef

windhead <- fit$coef[1]

> windtail <- fit$coef[2]

> exp(windtail)/exp(windhead) #Here is the answer

#here’s some other ways of setting things up

<-as.numeric(shuttle2$use=='auto')

#shuttle2$wind2<-as.numeric(shuttle2$wind=='head')

#head(shuttle2)

fit<-glm(use2 ~ **factor(wind) - 1**, family = binomial, data = shuttle2)

2. Multiple Categorical Predictors, Categorical/binary Outcome**.**

Consider the previous problem. Give the estimated odds ratio for autoloader use comparing head winds (numerator) to tail winds (denominator) adjusting for wind strength from the variable magn.

fit<-glm(use2 ~ **factor(wind) + factor(magn) - 1**, family = binomial, data = shuttle2)

summary(fit)$coef

exp(coef(fit))

exp(cbind(OddsRatio = coef(fit), confint(fit)))

3. Count Response, categorical predictor

Consider the insect spray data InsectSprays. Fit a Poisson model using spray as a factor level. Report the estimated relative rate comapring spray A (numerator) to spray B (denominator).

data(InsectSprays)

fit <- glm(count ~ spray - 1, family = "poisson", data = InsectSprays)

exp(fit$coef[1])/exp(fit$coef[2])