**Week 8 Regression Clinic**: Your instructor will supply data that you can read into R and which will appear in the environment as a data frame entitled “london.” This data frame contains a list of n=203 countries that participated in the 2012 summer Olympics. In addition to the country, each row contains the number of gold, silver, and bronze medals, as well as a measure of average personal income (in $10,000s) and a measure of overall population size (in billions of people). The research question is to ascertain whether a country’s wealth and population size predict its Olympic medal attainment.This first exercise includes: 1) creating a measure of medal attainment; 2) transforming skewed measurements; and 3) creating predictive models. You may wish to use the provided R Markdown file, which does much of the setup described in the first four steps.

**Phase 1**: Read in the dataset and examine it. Each row contains the counts of gold, silver, and bronze medals. Combine those counts into a single measure of Olympic achievement.

1. Place the c\_london.Rdata file on your computer. Read it in using the load() command or with open dialog under the Environment tab in the upper right pane of R-Studio. You will have a new data frame object, called “london,” in your working environment. Review the data and add a comment noting what you see. Run histograms and summaries on the variables in the dataset. Describe any issues in a comment.
2. There are many different ways of combining count measurements such as the ones stored in columns 2, 3, and 4 of the London data. Here are three of them:  
     
   \* Simple sum: Add together the gold, silver, and bronze medals to create a count of the total medals. Advantages: easy to understand. Disadvantages: does not give any additional weight to gold or silver medals.  
     
   \* Borda count: 18th century mathematician Jean-Charles de Borda developed this voting system, which gives higher weight to the first ranked object, etc. In our case, every gold medal would count for 3, every silver for 2, and every bronze for 1.  
     
   \* Dowdall system: 20th century politician Desmond Dowdall created a variation on the Borda count which treats the first rank as 1 and then reduces the weight of every subsequent rank. In our case, every gold medal would count for 1, every silver for 1/2, and every bronze for 1/3.  
     
   Calculate and store these measures of Olympic achievement in a new data frame. Run summary() and hist() on the new variable(s) and comment on the result.
3. Create and interpret a pairs plot. You can remove the categorical/factor variable from the analysis with the square brackets notation like this: *pairs(london[,-1])*
4. Both the predictors and the outcome variable in this data set are substantially positively skewed. Try some transformations to reduce skewness. Test different transformations on your variables to see if you can reduce the skewness. Here are some transformations:  
    *sqrt(X) # The square root of the value  
    log(X) # The natural log of the value   
    asin(X) # Arc sine, the inverse of the sine function  
    atan(X) # Arc tangent, the inverse of the tangent function  
    1/X # The reciprocal of the value*  
     
   Try each transformation on one of the skewed variables. List all of the values to the console so you can see if anything goes wrong. Also examine the results with histograms to see if you have gotten any reduction in skewness.

Now conduct regression analysis on your new dataset. Run as many sensible variations as you can (e.g., try the different counting methods as your outcome variables and the different transformations as your predictors).

1. Run lm() model.s Write a brief comment documenting the R-squared value, whether it was significant, the B-weights, whether they were significant, and anything else of interest in the output.
2. Create lmBF() models with at least 10,000 posterior estimates. Write a brief comment reviewing the HDIs for all relevant parameters.
3. Using the posterior distribution of sig2, calculate and histogram a posterior distribution of R-squared values for the lmBF() model. Here’s a reminder of that calculation:  
   rsqList <- 1 - (lmBFout[,"sig2"] / var(myDependentVar))
4. Integrate results to create a unified interpretation. Answer the research question: how well can a country’s wealth and population size predict its Olympic medal attainment?

**Regression Diagnostics**: This exercise uses the built-in “anscombe” data set. Run *str(anscombe)* to examine the variables. The x and y variables work in pairs: x1 goes with y1, etc. These small data sets were composed by British statistician Francis Anscombe (1981-2001) to underscore the value of graphing data before jumping in with numeric analysis. We will use them to illustrate some regression diagnostics.

1. Run four lm() models and save the resulting fit objects. Here’s the code for the first one:  
   *lmOut1 <- lm(y1 ~ x1, data=anscombe)*  
   Run summary on each fit object and take note of each R-squared value in a comment.
2. Create four bivariate (x-y) plots (raw data) and add comments describing what you see.
3. Create four histograms of residuals. Residuals should be normally distributed around zero. Note any anomalies in a comment. Here’s the first one:  
   *hist(lmOut1$residuals)*
4. Create four bivariate plots of fitted values versus residuals. Ideally there should be no obvious pattern to the plot. The amount of deviation from the zero line should be about the same, regardless of what neighborhood you check. Here’s the first one:  
   *plot(lmOut1$fitted.values, lmOut1$residuals)*

*abline(h=0) # Residuals should center around zero*  
Add comments describing any anomalies.

1. Using the car package (companion to applied regression) run “component plus residual” plots for each of the four fit objects. Don’t forget to install and library car before trying to run the procedure. Here’s the code for the first one:  
   *crPlots(lmOut1)*There’s a lot going on here. The X-axis has the original predictor. The Y axis shows each point as the sum of the predicted value and the residual. The dotted blue line is the OLS regression line. The solid pink line is a “smoothed” prediction curve (using loess polynomial regression). To the extent that the pink curve deviates from the blue line, there is distortion in the distribution of the residuals. Add comments describing what you see for each plot.
2. Use the outlierTest() procedure (from the car package) to analyze the fit object from each of your four regressions. This procedure is running significance tests on the studentized residuals (should be distributed as t) using a Bonferroni correction. If the Bonferroni p value is significant, the flagged observation is an outlier with high leverage.
3. Run lmBF() on each of the four pairs of X and Y and generate posterior distributions. Evaluate the slope on each X and the R-squared. Is anything different from the OLS?
4. According to the package documentation, the “glvma” package, “Perform[s] a single global test to assess the linear model assumptions, as well as perform[ing] specific directional tests designed to detect skewness, kurtosis, a nonlinear link function, and heteroscedasticity.” Install and library the glvma package.
5. Using the fit object from each of the four OLS regressions, run the glvma() procedure and examine the results. Add a comment indicating which tests failed for which data sets. Here’s the first one:  
   *summary(gvlma(lmOut1))*
6. With all of these different diagnostics and tests, one of the big things we have not examined yet is collinearity among the predictors. The most common diagnostic for this is the variance inflation factor (VIF). VIF is calculated for each predictor by temporarily making that predictor the outcome variable and using all of the other predictors to predict it. Obviously, you need two or more predictors for this to work. Large values of VIF indicate that a predictor is redundant (highly correlated) with other predictors.  
     
   The car package contains a vif() function which you can use to evaluate the output object from the following regression that uses synthesized data:  
     
   *myY <- rnorm(32) # A random Y variable*

*myX1 <- rnorm(32) # A random X variable*

*myX2 <- rnorm(32) # Another random X variable*

*myX3 <- myX1 + myX2 + rnorm(32) # An X variable highly correlated with other Xs*

*myData <- data.frame(myY, myX1, myX2, myX3)*

*lmOutV <- lm(myY ~ myX1 + myX2 + myX3, data=myData)*  
  
Ideally, VIF would be about 1 for each predictor. Values between 1 and 5 show a modest amount of overlap among predictors. Values greater than 5 are said to be very highly collinear. Some people use a more generous cutoff of 10 before they will consider removing a predictor because of its redundancy.   
  
An important paradox: In a regression where some predictors have high VIF, this can actually suppress the size of some B/beta weights as the redundant predictors “fight” one other to predict the outcome variable.