Methodology statement
Yaoli Mao

Problem: I'm modeling how video features and audience attention features affect audience comprehension about the video content, holding audience attitude as covariates.
Data set: fixation duration/counts of 29 subjects watching two video talks, their comprehension and attitude results towards the content after watching the video, video features including gestures used by the speaker and scene shots used by the video.
Treating the data in two levels:
-Level one: Individual subject level.
-Level two: Timeblock level. when temporal perspective of the video comes into play, treating the individual subject as five time blocks.

A description of the method you plan to employ to analyze your data. A list of assumptions, "gotchas" or other considerations you need to be aware of to use that method.
Method one: Linear regression. Assumptions involves homogeneity test(errors are normally distributed). No missing data.
Method two: Regression Tree. No missing data. (still need to find more)
Model three: Random Forests. (still need to find more)
Other considerations: Variation in the comprehension score is small. Limited sample size.

How to validate?
For linear regression, One way is to report confidence intervals (or their Bayesian equivalents) around the regression coefficients; another popular alternative is to compute them through resampling (e.g., bootstrapping or jackknifing), which makes fewer assumptions about the distribution of errors.
[some confusion: can we use significance test-P value? Cronbach alpha? Coefficient of determination aka R squared(how much variance explained)?]
For regression tree and random forests, select the training and the test. Divide the data set into two subsets. One subset is used to find the linear regression (training subset), another is used to evaluate it (test subset).

Some learning from the internet:
training and test-I don't think this kind of assessment is generally used with simple regression models. What would it tell you that you wouldn't find out from using the entire dataset to generate your regression parameters? Normally the reason to use an evaluation dataset is to prevent overfitting, but that's not an issue when you already know that your model is going to contain just one independent variable.
http://stats.stackexchange.com/questions/43310/how-to-evaluate-results-of-linear-regression
What is the difference between "coefficient of determination" and "mean squared error"?
http://stats.stackexchange.com/questions/32596/what-is-the-difference-between-coefficient-of-determination-and-mean-squared

$R^2 = 1 - \frac{SSE}{SST}$, where SSE is the sum of squared error (residuals or deviations from the regression line) and SST is the sum of squared deviations from the dependent's Y mean.

$MSE = \frac{SSE}{n-m}$, where n is the sample size and m is the number of parameters in the model (including intercept, if any).

R^2 is a standardized measure of degree of predictedness, or fit, in the sample. MSE is the estimate of variance of residuals, or non-fit, in the population. The two measures are clearly related, as seen in the most usual formula for *adjusted* R^2 (the estimate of R^2 for population):

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-m} = 1 - \frac{SSE/(n-m)}{SST/(n-1)} = 1 - \frac{MSE}{\sigma_y^2}.$$

More validation for regression trees:
http://www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf
from arbitrarily setting a minimum number of points q per node. A more successful approach to finding regression trees uses the idea of cross-validation from last time. We randomly divide our data into a training set and a testing set, as in the last lecture (say, 50% training and 50% testing). We then apply the basic tree-growing algorithm to the training data only, with q = 1 and δ = 0 — that is, we grow the largest tree we can. This is generally going to be too large and will over-fit the data. We then use cross-validation to prune the tree. At each pair of leaf nodes with a common parent, we evaluate the error on the testing data, and see whether the sum of squares would be smaller by remove those two nodes and making their parent a leaf. This is repeated until pruning no longer improves the error on the testing data. There are lots of other cross-validation tricks for trees. One cute one is to alternate growing and pruning. We divide the data into two parts, as before, and first grow and then prune the tree. We then exchange the role of the training and testing sets, and try to grow our pruned tree to fit the second half. We then prune again, on the first half. We keep alternating in this manner until the size of the tree doesn't change.

"Cross-validation is not necessary when using random forest, because multiple bagging in process of training random forest prevents over-fitting."