R-Script Talking Points
German Credit Dataset

Libraries:

Caret - It provides functions designed to simplify common tasks associated with predictive modeling. R has some of functionality built in, but using this package allows you to further analysis. Caret also has sample datasets such as the one I'm using called German Credit which demonstrates CARET's strengths.

Ggplot2 – Ggplot 2 is a R package that allows the user to create complex R data visuals. It is similar to built in R graphs, but allows much more customization through layers. I will use this to customize several graphics to visualize my distributions and explore my data to gain a better understanding

Catools – Catools library is a library that gives the user several functions to analyze a predictive model. In my script I have created a logistic regression to predict 'good' or 'bad' credit data

PVclust – In order to conduct a cluster analysis I searched the web for clustering packages. Pvclust is interesting it allows me to store my cluster analysis as a variable and then plot it, which outputs a very clean looking tree of my clusters

Script Walkthrough:

Misc.

Since the dataset I am using resides in the CARET package, I already have access to it upon loading that library into my script. Using the data() function I tell R I will be using the GermanCredit dataset. Since this is the only dataset in this script, I also run the attach() which allows me to call variable names anywhere in the script.

After that I set the seed of the script which will allow me to generate the exact same random numbers time after time ensuring my results and analysis can be reproduced.

1. **Data Exploration Section**

Before doing any true statistical analysis I wanted to look at some variables that were of interest from initial observations.

First I looked at the structure of my GermanCredit Data using the str() function

Then I calculated summary statistics on the data set to see average, min, max, med, etc… just to see if there are any trends going on

I wanted to see the total count of people by class so I used ggplot to make a bar chart of my class variable. This function is similar to plot() but has some variations in names, especially regarding colors.

Next I was interested in seeing how the Duration variable was distributed so using the hist() I was able to visually see

I was also interested in seeing the distribution of credit amount so I again used the hist() to visualize that

Anova 1: For further exploration I wanted to see if on average there is a difference in Credit duration by class. Using the aov() function that comes with R, I created a model and stored it in a variable. After I created this variable, I ran a summary on it in order to get the anova output. To visualize this I ploted a boxplot of the relationship

Anvoa 2: Again going further into the data I was interested in seeing if the amount of credit varied by class. I assumed that if the class was bad, then the credit amount would be lower. I again create an anova model and stored it in a variable and ran a summary of it and followed that up with a plot.

2. Logistic Regression

Given that my dataset is found in a Machine Learning package, I wanted to utilize some of this functionality. First before creating a model I split up the data into 2 set. One was a training set containing 80% of the data that I will use to develop my model, the other set consisted of the remaining 20% of data that I will use to validate and test my model's performance

I created a split variable which tell me what row of data I should split at. I then use this value to split the data into the two new sets train and test.

Next using the built in glm() functions I create a benchmark logistic model that predicts class based on all variables. I store this model in a model variable so I can use it later. After I run the model I want to see how it did, so I run the summary() to see significant variables

Next I want to save only the coefficient measure from the summary object. I store the summary in a new variable and then extract the coefficients variable. I then transform that into a data frame to make it easier to read. The decimals go out to far and are difficult to read so I round the entire data frame to 4 decimal places which lets me easily see the p-values of each variable.

After seeing the initial models performance I want to use that model to predict my test data. I created a new predicitons object that will store my predicitons from the predict() which I give a model , and what data I want it to predict on.

To find the optimal cutoff point I use the colAuc function from catools() to tell me what is the optimal point to determine the good or bad class I am trying to predict. I then plot this ROC curve and add a green line at the exact cutoff point I should use.

After I find this point I create a new variable called classes that translate the probabilities from my model into 1's and 0's which says 1 if good and 0 if bad. Then I create a confusion matrix of my predicted classes, by the actual classes in my test set to see how well the model did. I create this matrix using the built in table()

3. Cluster Analysis Via Pvclust

Finally I wanted to see if there are similar types of consumers within the first 9 columns of data in my dataset. I found a library called Pvclust which makes this analysis much easier than doing it by myself.

First I create a variable called pvclust.model which will store the pvclust object I will use to plot

After the model is created and stored, I use the plot() to plot the cluster object I just created. The result is a tree like structure of the input data.