Lab 04

Section 1: Group Comparisons with Continuous Data

This exercise will be done in Python.

1. Read the males_ht_wt_cntry.csv file into a data frame
2. Examine the data
    a. Display some rows to make sure it imported correctly
    b. Generate histograms of the heights by country
    c. Generate histograms of the weights by country
3. Conduct a test to determine if the weights differ by nationality and interpret your results. Use this link as a reference.
4. ANOVA won't tell you which sets of weights differ. You will need to compare each group against each other to determine that. Use this link as a reference.
    a. Conduct a test to determine if the weights of the Italian males were significantly different than the Dutch males (from the Netherlands) and interpret your results
    b. Conduct a test to determine if the weights of the Italian males were significantly different than the American males and interpret your results
    c. Conduct a test to determine if the weights of the American males were significantly different than the Dutch males (from the Netherlands) and interpret your results
5. Conducting multiple tests like this increases the odds of getting false significant results. What is the probability one of these t-tests is not actually significant (i.e. false positive)?
6. When comparing these groups, it's better to control the FWER. Use a multiple comparison procedure with a Tukey adjustment. See this link for how to do this in Python.

Section 2: Group Comparisons with Categorical Data

1. Create a new BMI column. Use the Imperial formula $BMI = \frac{Weight*703}{Height^2}$.
2. Create another new column 'Overweight' that is a 1 if BMI >= 25 and 0 otherwise. There are several ways to do this in Python.
3. Create a contingency table and examine it. Describe any differences you see between nationalities.
4. Conduct a test to see if the differences are significant. Explain your findings.

Section 3: Regression

1. Build a linear regression of to see whether height predicts weight. There are two main modules for conducting linear regression in Python. Try both. Explain the results.
    a. Note: When using `sklearn`, the predictor variable must be an (n,1) array.
2. Fit the same regression model using linear algebra. Compare your resultant β's to the ones you obtained earlier.
    a. Note: While `np.dot` can be used to multiply matrices, `np.matmul` is better.