1. **Outline of Data Processing and Analysis  
   (using OSEMN model)**
2. **OBTAIN:**
   * **Import data, inspect, check for datatypes to convert and null values**  
     + Display header and info
     + Drop any unneeded columns (df.drop(['col1','col2'],axis=1)
3. **SCRUB: cast data types, identify outliers, check for multicollinearity, normalize data**
   * Check and cast data types
     +  Check for #'s that are store as objects (df.info())
       - when converting to #'s, look for odd values (like many 0's), or strings that can't be converted
       - Decide how to deal weird/null values (df.unique(), df.isna().sum(), df.describe()-min/max, etc
     +  Check for categorical variables stored as integers
   *  Check for missing values (df.isna().sum())
     + Can drop rows or colums
     + For missing numeric data with median or bin/convert to categorical
     + For missing categorical data: make NaN own category OR replace with most common category
   *  Check for multicollinearity
     + use seaborn to make correlation matrix plot [Evernote Link](https://www.evernote.com/l/AArNyaEwjA5JUL6I9PazHs_ts_hU-m7ja1I/)
       - Good rule of thumb is anything over 0.75 corr is high, remove the variable that has the most correl with the largest # of variables
   *  Normalize data (may want to do after some exploring)
     + Most popular is Z-scoring (but won't fix skew)
     + Can log-transform to fix skewed data
4. **EXPLORE:Check distributions, outliers, etc**
   *  Check scales, ranges (df.describe())
   *  Check histograms to get an idea of distributions (df.hist()) and dat transformations to perform
     + Can also do kernel density estimates
   *  Use scatterplots to check for linearity and possible categorical variables (df.plot(kind-'scatter')
     + categoricals will look like vertical lines
   *  Use pd.plotting.scatter\_matrix to visualize possible relationships
   *  Check for linearity
5. **FIT AN INITIAL MODEL:**
   * Various forms, detail later...
   * **Assessing the model:**
     + Assess parameters (slope,intercept)
     + Check if the model explains the variation in the data (RMSE, F, R\_square)
     + *Are the coeffs, slopes, intercepts in appropriate units?*
     + *Whats the impact of collinearity? Can we ignore?*
6. **Revise the fitted model**
   * Multicollinearity is big issue for lin regression and cannot fully remove it
   * Use the predictive ability of model to test it (like R2 and RMSE)
   * Check for missed non-linearity
7. **Holdout validation / Train/test split**
   * use sklearn train\_test\_split