### Assignment 1 Walkthrough

# 1. Perform Linear Regression

Overview:  
This method performs linear regression on given feature set X and target set y, returning the model's coefficients and intercept.  
  
Detailed Pseudo Code:  
FUNCTION perform\_linear\_regression(X, y)  
 INITIALIZE a LinearRegression model as 'model'  
 FIT 'model' with features 'X' and target 'y'  
 RETRIEVE coefficient of the model into 'coefficients'  
 RETRIEVE intercept of the model into 'intercept'  
 RETURN 'coefficients', 'intercept'  
END FUNCTION  
  
Implementation Guide:  
- Begin with importing LinearRegression from sklearn.linear\_model.  
- Instantiate the LinearRegression model.  
- Fit the model using the .fit() method with X and y as parameters.  
- After fitting the model, extract the coefficients and intercept.  
- Return these values for use in predictions or analysis.

# 2. Calculate Statistics

Overview:  
This function calculates the mean, median, and standard deviation of a dataset.  
  
Detailed Pseudo Code:  
FUNCTION calculate\_statistics(data)  
 SET 'mean' to the mean of 'data'  
 SET 'median' to the median of 'data'  
 SET 'std' to the standard deviation of 'data'  
 CREATE a dictionary 'statistics' with keys 'mean', 'median', 'std', and their respective values  
 RETURN 'statistics'  
END FUNCTION  
  
Implementation Guide:  
- Utilize numpy's np.mean(), np.median(), and np.std() functions to calculate the respective statistical measures.  
- Store these measures in a dictionary for easy access.  
- Return the dictionary containing mean, median, and standard deviation.

# 3. Preprocess DataFrame

Overview:  
This function preprocesses the DataFrame by encoding categorical variables, imputing missing values, and scaling numerical features.  
  
Detailed Pseudo Code:  
FUNCTION preprocess\_dataframe(df)  
 IDENTIFY 'numeric\_features' by selecting 'int64' and 'float64' data types from 'df'  
 IDENTIFY 'categorical\_features' by selecting 'object' data types from 'df'  
 CREATE 'numeric\_transformer' as a Pipeline with SimpleImputer and StandardScaler  
 CREATE 'categorical\_transformer' as a Pipeline with SimpleImputer and OneHotEncoder  
 COMBINE transformers into 'preprocessor' using ColumnTransformer  
 APPLY 'preprocessor' to 'df' and assign result to 'df\_processed'  
 CONVERT 'df\_processed' to a DataFrame  
 RETURN 'df\_processed'  
END FUNCTION  
  
Implementation Guide:  
- Begin by identifying numerical and categorical columns based on their data types.  
- For numerical features, create a pipeline that first imputes missing values with the mean, then scales the data.  
- For categorical features, create a pipeline that imputes missing values with a placeholder ('missing'), then applies one-hot encoding.  
- Combine these pipelines using ColumnTransformer, specifying which transformer applies to which columns.  
- Apply this combined transformer to the DataFrame, converting the output back into a DataFrame format.  
- Return the preprocessed DataFrame.