Assignment #5 – Data Cleaning

To begin, open Jupyter Notebook through your preferred environment such as Google Colab or Anaconda. Once opened, create a new notebook. In Google Colab this is done by selecting **File** then **New Notebook**. Next, carefully write the code provided into each cell of the notebook as shown, including any comments.

Execute each cell by clicking the **Run or Play** button located on the left side of each cell. Should you have an error during execution, review the code within the cell. Pay close attention to potential typos, incorrect spacing, or misspellings, verifying each line against the provided code. Correct and rerun the cell.

Continue this process for all cells. Then save as a pdf by selecting **File** then **Print**.

Submit PDF to Canvas. Code and output must be clearly identified for full credit.

50 points

Intentional Blank Space

data_cleanup_datawrangling.ipynb

```
[ ] # import python packages
    import pandas as pd
    print("import package libraries")
```

```
[ ] # load dataset
    tree_census = pd.read_csv('trees.csv')
    print("load dataset may take long to load")
```

```
[ ] # look at the first five rows
    tree_census.head()
```

```
[ ] # look at the last five rows
    tree_census.tail()
```

```
[ ] # list of column names
    tree_census.columns
```

```
[ ] # identify the size, number of rows and columns in the dataset
    tree_census.shape
```

```
[ ] # summary of the dataset
    tree_census.info()
```

```
[ ] # health status of trees
    tree_census.health.value_counts(dropna=False)
```

```
[ ] # get status on the trees
    tree_census.status.value_counts(dropna=False)
```

```
[ ] # subset of the original, removed columns not interested in
    trees_subset = tree_census[['tree_id', 'tree_dbh',
           'stump_diam', 'curb_loc', 'status', 'health', 'spc_latin', 'spc_common',
           'steward', 'guards', 'sidewalk', 'user_type', 'problems', 'root_stone',
           'root_grate', 'root_other', 'trnk_wire', 'trnk_light', 'trnk_other',
           'brnch_ligh', 'brnch_shoe', 'brnch_othe']]

    # list the first 5 rows of the new subset
    trees_subset.head()
```

```
[ ] # check for any null values
    trees_subset.isna().sum()
```

```
[ ] # show all that are none values in health, alot of missing values NaN
    tree_census.describe()
```

```
[ ] # generate histogram of data distribution
    trees_subset.hist(bins=60, figsize=(20,10))
```

```
[ ] # trees larger than 50
    big_trees = trees_subset[trees_subset['tree_dbh']> 50]
    big_trees.head()
```

```
[ ] # box plot
    tree_census.boxplot(column='tree_dbh', by='stump_diam')
```

```
[ ] # scatter plot
    big_trees[['tree_id', 'tree_dbh']].plot(kind='scatter',x='tree_id', y='tree_dbh')
```