CMPSC 448: Machine Learning. Homework 1:
Exploratory Data Analysis with Pandas.
40 points
Due: January 20, 2020

1. Instructions

The purpose of this homework is to introduce you to exploratory data analysis with Pandas, a python library for data manipulation.

- Download the adult dataset from the UCI Machine Learning repository: `https://archive.ics.uci.edu/ml/machine-learning-databases/adult/`
  - adult.data is a csv file containing the data.
  - adult.names provides information about the column names.
- For this homework, you will fill in the appropriate functions in hw1.py and submit it to gradescope.
- You may not look at anyone else's code.
- This homework requires Python 3.5 or higher and the Pandas library. It is easiest to just install the Anaconda Python distribution.

2. Short Pandas Tutorial

You are encouraged to read pandas tutorials, such as `https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/`.

Pandas stores a dataset in a structure called a *dataframe*. It can read a variety of files, including CSV files, from a local drive or from a network. The read_csv function has many options that you can examine through python's help system.

For example, the following code reads the adult.data and adult.test files directly from the uci machine learning repository. Note how the code provides the column names (adult.data does not have a header so column names are needed; meanwhile adult.test does have a header, so we are telling it to skip the first row and use the names we provided). Many datasets also have missing values. In this dataset, they are denoted as '?', so we are telling pandas to treat question marks as missing (i.e NA).

```
import pandas as pd
names = ["age", "workclass", "fnlwgt", "education", "education_num",
         "marital_status", "occupation", "relationship", "race", "sex",
         "capital_gain", "capital_loss", "hours_per_week", "native_country", "income"]
adult_df = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
                       header=None, names=names, sep=",\s+", na_values="?",
                       verbose=True, engine='python')
test_df = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test",
                       header=None, names=names, sep=",\s+", na_values="?", verbose=True,
                       skiprows=[0], engine='python')

# in the test set, there is an extra . at the end of income, we can remove it like this:
test_df["income"] = test_df["income"].apply(lambda x: x[:-1])
```

You can explore the data with the following commands, which you are encouraged to run:

```
print("Columns Information:")
print(adult_df.info())
print("")
print("Summary statistics of Train dataset:")
print(adult_df.describe())
print("")
print("Summary statistics of Test dataset:")
print(test_df.describe())
```

You can also select different parts of the data as follows:

```
# Access age of the first record:
adult_df.at[0,'age']
# select a subset of rows:
adult_df[0:4]
# access specific columns:
adult_df[['age', 'workclass']]
adult_df.age
# get numpy array for a column:
adult_df.age.values
# specific rows and columns:
adult_df[1:4][['age', 'workclass']]
```

You can also perform queries over the data:

```
# get the rows where workclass is missing, note that head(5) gives the first 5 rows of the
    result
adult_df[pd.isna(adult_df.workclass)].head(5)
# get records with age 18 then count how many there are
adult_df[adult_df.age == 18].count()
# group by queries:
adult_df.groupby(['marital_status']).mean()
```

3. Questions

In the following questions, df is a pandas dataframe with the same column names as the adult dataset (e.g., as specified by the **names** variable in the previous section). Note that when we test your code, the contents of df may be different from adult.data (but the column names will be the same).

Question 1. *In hw1.py, fill in the code for the function* `q1(df)`. *This function* ***q1*** *should return the number of records in* ***df*** *that have missing values for* ***education*** *in* ***df****.*

Question 2. *In hw1.py, fill in the code for the function* `q2(df)`. *This function* ***q2*** *should return the number of records in* ***df*** *for which* ***age*** *is between 40 and 60 (inclusive).*

Question 3. *In hw1.py, fill in the code for the function* `q3(df)`. *This function* ***q3*** *should return the average* ***age*** *of males in* ***df****.*

Question 4. *In hw1.py, fill in the code for the function* `q4(df)`. *This function* ***q4*** *should return the number of females in* ***df*** *for whom their* ***native_country*** *is England.*

Question 5. *In hw1.py, fill in the code for the function* `q5(df)`. *This function* ***q5*** *should give standard deviation of ages of people in* ***df*** *whose income is above 50K and live in United-States.*

Question 6. *In hw1.py, fill in the code for the function* `q6(df, a, b)`. *The variables* ***a*** *and* ***b*** *will have possible values for the workclass column. This function should return True if (in* ***df****) the number of people in workclass* ***a*** *who earn more than 50k per year is larger than the number of people in workclass* ***b*** *who earn more than 50k per year. The function should return False otherwise (for example, in the case they are equal).*

Question 7. *In hw1.py, fill in the code for the function* `q7(df)`. *This function* ***q7*** *should return the number of people in* ***df*** *who earn less than 50K per year and work for 20 or more hours per week.*

Question 8. *In hw1.py, fill in the code for the function* `q8(df)`. *This function* ***q8*** *should return the average time of work (hours per week) in* ***df*** *for those who earn more than 50K per year for each of the following countries:* ***United-States, Canada, India, England, Germany****. The answer should be a python dictionary, such as:*

```
{
"United-States": 99,
"Canada": 98.3,
"India": 97.2 ,
"England": 96.1,
"Germany": 94.2
}
```