Submission Instruction: Submit a PDF file of your codes and outputs and a public Google Colab shared link to your source file (.ipynb format) to Blackboard (See the submission details on Blackboard).

Due Date: 09/12/2022, 11:59 pm

Name: Ashley Cortez

P1: Write a Python code in Colab using Pandas and Matplotlib libraries to accomplish the following tasks:

1. Import the iris flowers dataset using pandas.read_csv() with the following URL link **(10pt)**; Your DataFrame should have the following column names: 'sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)', and 'class' **(5pt)**; Print the first 5 rows of the resulting DataFrame **(5pt)**.

- Dataset source file: http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
- Dataset description: http://archive.ics.uci.edu/ml/datasets/iris
- https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html
    - You can fetch the data online by inputting the above URL in pandas.read_csv(url = XXX). Downloading the data to a local copy will make the shared Colab code in your homework submission inexecutable.
    - Pay attention to the header and index_col arguments when using read_csv().

```
# write your answer here
import numpy as np
import pandas as pd

url = "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
df = pd.read_csv(url, index_col = False, header = None)

df.columns = ['sepal length (cm)','sepal width (cm)','petal length (cm)','petal width (cm)','class']

df.columns

df.head(5)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

2. Summarize the dataset

a. Print out a concise summary of the DataFrame using .info() and the shape of the DataFrame **(5 pt)**

```
# write your answer here
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   class              150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.shape
```

```
(150, 5)
```

b. Print out the statistics of the continuous columns using .describe() (i.e., the four attribute columns) **(5 pt)**

```
# write your answer here
df.describe()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000

c. Print the number of rows that belong to each class **(5pt)**

```
# write your answer here
df['class'].describe()
```

```
count             150
unique              3
top       Iris-setosa
freq               50
Name: class, dtype: object
```

3. Data Visualization

a. Separate out the first four columns of the original DataFrame into a new DataFrame and print out the first 5 rows of the new DataFrame **(5 pt)**

```
# write your answer here
new_df = df[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', "petal width (cm)"]].copy()

new_df.head(5)
```

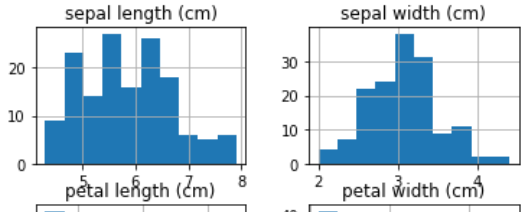
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

b. Univariate Plots: plot a histogram for each column of the new DataFrame **(5 pt)**

```
# write your answer here

new_df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f1b023e2410>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1b0237f750>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1b02303710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1b022b6d10>]],
      dtype=object)
```



c. Multivariate Plots: plot a scatter plot for each pair of the columns of the new DataFrame using the pandas.plotting.scatter_matrix function**(5 pt)**

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.plotting.scatter_matrix.html

```
# write your answer here

pd.plotting.scatter_matrix(new_df)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f1b0219edd0>,
```

P2: Write a Python code in Colab using Pandas and/or Matplotlib libraries to accomplish the following tasks

1. Import the Census Income (Adult) dataset using Pandas, use the 14 attribute names (i.e., “age”, “workclass”, ….., “native-country”) as explained in the dataset description as the first 14 column names and “salary” as the last column name **(5 pt)** , view the strings ‘?’, ‘ ?’, ‘? ’, or ‘ ? ’ as the missing values and replace them with NaN (the default missing value marker in Pandas) **(10 pt)**, and print out the first five rows of the DataFrame. **(5 pt)**

- Dataset source file: http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data
- Dataset description: http://archive.ics.uci.edu/ml/datasets/census+income
- Pay attention to the header and index_col arguments when using pandas.read_csv().

```
# write your answer here

url2 = "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
na = ['?', ' ?', ' ? ', '? ']
df2 = pd.read_csv(url2, index_col= False, header = None,
                  na_values = na)

df2.columns = ['age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capit

df2.head(5)
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K

2. Dataset checking and cleaning

a. Print out a concise summary of the DataFrame and observe if null values exist in each column of the DataFrame by checking the summary**(10pt)**

```
# write your answer here
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             32561 non-null  int64 
 1   workclass       30725 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   education       32561 non-null  object
 4   education-num   32561 non-null  int64 
 5   marital-status  32561 non-null  object
 6   occupation      30718 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital-gain    32561 non-null  int64 
 11  capital-loss    32561 non-null  int64 
 12  hours-per-week  32561 non-null  int64 
 13  native-country  31978 non-null  object
 14  salary          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

b. Find out the rows that contain missing values and print them out **(10pt)**

```
# write your answer here
#df2.isnull().sum()

df2.loc[df2.isnull().any(axis=1)]
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
14	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	0	0	40	NaN	>50K
27	54	NaN	180211	Some-college	10	Married-civ-spouse	NaN	Husband	Asian-Pac-Islander	Male	0	0	60	South	>50K

c. Drop the rows of the DataFrame with missing values and observe if null values still exist in each column by checking the concise summary again **(10 pt)**

```
# write your answer here
df2.dropna(inplace = True)
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30162 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             30162 non-null  int64 
 1   workclass       30162 non-null  object
 2   fnlwgt          30162 non-null  int64 
 3   education       30162 non-null  object
 4   education-num   30162 non-null  int64 
 5   marital-status  30162 non-null  object
 6   occupation      30162 non-null  object
 7   relationship    30162 non-null  object
 8   race            30162 non-null  object
 9   sex             30162 non-null  object
 10  capital-gain    30162 non-null  int64 
 11  capital-loss    30162 non-null  int64 
 12  hours-per-week  30162 non-null  int64 
 13  native-country  30162 non-null  object
 14  salary          30162 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

Colab paid products - Cancel contracts here

✓ 0s completed at 11:28 AM