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: 02/07/2022, 11:59 pm

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/pandasdocs/stable/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().

```
import pandas as pd
url= "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
colnames = ['sepal lenght(cm)', "sepal width (cm)", 'petal length (cm)', 'petal width (cm)', 'class']
df = pd.DataFrame(pd.read_csv(url, names= colnames))
```

2. Summarize the dataset

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

```
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 lenght(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)**

```
df.describe()
```

	sepal lenght(cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

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

```
df['class'].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: class, dtype: int64
```

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)**

```
df1 = df.drop('class',axis=1)
```

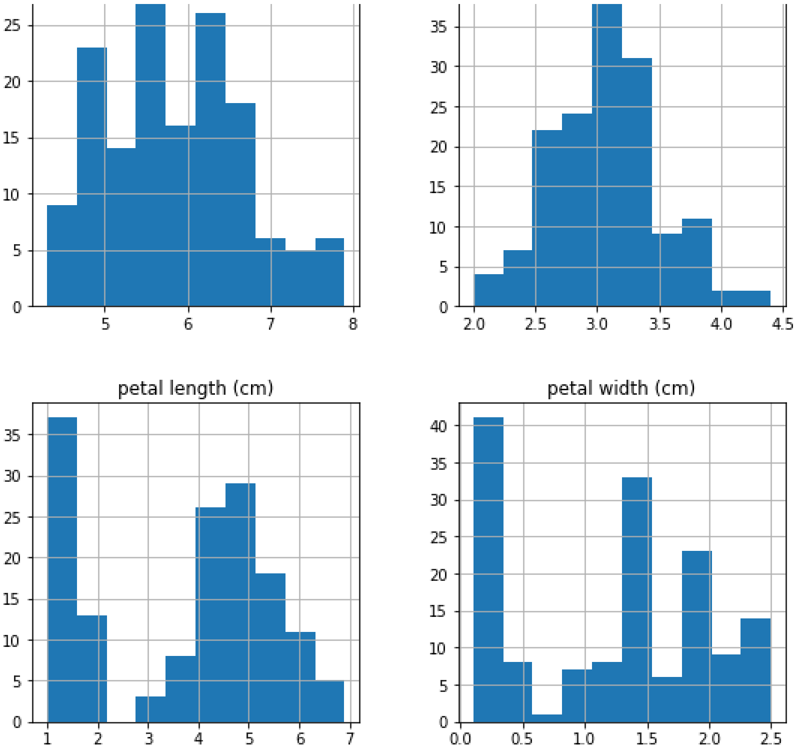
```
df1.head()
```

	sepal lenght(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)**

```
import matplotlib.pyplot as plt
df1.hist(bins = 10,figsize=(9.0,9.0))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f48b360ab10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b35cedd0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f48b358e410>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b3540a10>]],
      dtype=object)
```

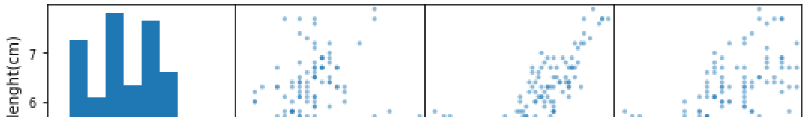


c. Multivariate Plots: plot a scatter plot for each pair of the columns of the new DataFrame using the pandas.plotting.scatter_matrix function**(5 pt)**

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

```
pd.plotting.scatter_matrix(df1, figsize=(9.0,9.0))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f48b3392910>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b3354450>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2ea3bd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2e66210>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2e19810>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2dd0e10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2d934d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2dc5a10>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2dc5a50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2d0a190>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2cf8c90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2cb92d0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2c718d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2c30310>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2be6810>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f48b2b9cd10>]],
      dtype=object)
```



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().

2. Dataset checking and cleaning

```
colnames=['age','workclass','fnlwgt','education','education-num','marital-status','occupation','relationship','race
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
Adult = pd.read_csv(url,names= colnames, index_col = False, na_values=['?',' ?','? ',' ? '])
Adult.shape
```

```
(32561, 15)
```

```
Adult.replace(to_replace=['?',' ?','? ',' ? '], inplace = True)
Adult.isin(['?',' ?','? ',' ? ']).any() #check if the values has been replaced
```

```
age               False
workclass         False
fnlwgt            False
education         False
education-num     False
marital-status    False
occupation        False
relationship      False
race              False
sex               False
capital-gain      False
capital-loss      False
hours-per-week    False
native-country    False
salary            False
dtype: bool
```

```
Adult.head()
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	ca
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	

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)**

```
Adult.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
```

```
Adult.isna().any()
```

```
age                False
workclass           True
fnlwgt             False
education          False
education-num      False
marital-status     False
occupation          True
relationship       False
race               False
sex                False
capital-gain       False
capital-loss       False
hours-per-week     False
native-country      True
salary             False
dtype: bool
```

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

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

```
Adult[Adult.isnull().any(axis=1)]
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gai
14	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	
27	54	NaN	180211	Some-college	10	Married-civ-spouse	NaN	Husband	Asian-Pac-Islander	Male	
38	31	Private	84154	Some-college	10	Married-civ-spouse	Sales	Husband	White	Male	
51	18	Private	226956	HS-grad	9	Never-married	Other-service	Own-child	White	Female	
61	32	NaN	293936	7th-8th	4	Married-spouse-absent	NaN	Not-in-family	White	Male	
...	...	...	...	...	...	...	...	...	...	...	
32530	35	NaN	320084	Bachelors	13	Married-civ-spouse	NaN	Wife	White	Female	
32531	30	NaN	33811	Bachelors	13	Never-married	NaN	Not-in-family	Asian-Pac-Islander	Female	
32539	71	NaN	287372	Doctorate	16	Married-civ-spouse	NaN	Husband	White	Male	
32541	41	NaN	202822	HS-grad	9	Separated	NaN	Not-in-family	Black	Female	
32542	72	NaN	129912	HS-grad	9	Married-	NaN	Husband	White	Male	

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)

```
Adult.dropna(inplace=True)
```

```
Adult.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
memory usage: 3.7+ MB
```

✓ 0s completed at 8:43 PM