Creating and Accessing Pandas DataFrames

Course Code: CPE 031	**Program**: Computer Engineering
Course Title: Visualization and Data Analysis	**Date Performed**: 10/15/2024
Section: CPE21S4	**Date Submitted**:10/15/2024
Name: Sanchez, Christan Ray R.	**Instructor**: Prof. Ma. Rizette Sayo

Intended Learning Outcomes (ILO):

By the end of this laboratory session, learners will be able to

- Construct and manipulate Pandas DataFrames from various data structures (such as lists, dictionaries, and NumPy arrays) while demonstrating an understanding of DataFrame attributes and methods. This includes loading the dataset, creating DataFrames with appropriate column labels and accessing data from rows and columns.

Instructions:

1. **Loading your dataset**: Refer back to your chosen dataset from the PRELIM period. Whether you downloaded it or stored it in your Google Drive, you are required to load it into the Google Colab. Watch this video to learn more about how to read CSV files in Google Colab.**(Take a screenshot to document successful execution.)**

2. **Creating a dataframe from your CSV file:** Once you have successfully loaded your dataset, you need to create a dataframe from your uploaded CSV file.**(Take a screenshot to document successful execution.)**

3. **Creating a dataframe from a dictionary of lists:** Manually create a dictionary where each value is composed of a list from your original dataset, then load it into a dataframe, before printing it. You are required to provide at least five (5) observations in your list. **(Take a screenshot to document successful execution.)**

4. **Creating a dataframe from a list of dictionaries:** Manually create a list of dictionaries from your original dataset, then pass it into a dataframe, before printing it. You are required to provide at least five (5) observations in your list. **(Take a screenshot to document successful execution.)**

5. **Selecting dataframe columns:** Execute a method that would allow you to select a single and multiple dataframe columns. **(Take a screenshot to document successful execution.)**

6. **Selecting dataframe rows:**Execute a method that would allow you to select a single and multiple dataframe rows using panda indexing and python indexing.

Output:

1) **Loading your dataset**:

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  Car_data - ...ed_data.csv
• Car_data - extended_data.csv(text/csv) - 38781 bytes, last modified: 10/15/2024 - 100% done
Saving Car_data - extended_data.csv to Car_data - extended_data.csv
```

2) **Creating a dataframe from your CSV file:**

```
[35] import pandas as pd

     df = pd.read_csv('Car_data - extended_data.csv')

     Car_data = pd.DataFrame(data)

     Car_data
```

	model_year	brand	model	type	miles_per_gallon
0	2016	Toyota	Land Cruiser Base	SUV	13
1	2014	RAM	ProMaster 2500 Window Van High Roof	Van	15
2	2002	Ford	Mustang GT	Coupe	16
3	2012	BMW	428 Gran Coupe i xDrive	Sedan	27
4	2008	Mercedes-Benz	SL-Class SL500 Roadster	Convertible	18
...	...	...	...	...	...
994	2016	Subaru	WRX STI Base	Sedan	18
995	2010	Mercedes-Benz	E-Class E 350 4MATIC	Sedan	20
996	2016	Audi	A6 2.0T Premium Plus	Sedan	30
997	2017	Mercedes-Benz	C-Class C 300 4MATIC	Sedan	25
998	2012	Toyota	4Runner 4WD	SUV	17

999 rows × 5 columns

3) **Creating a dataframe from a dictionary of lists:**

```python
import pandas as pd

data = {
    'year': [2016, 2014, 2002, 2012, 2008],
    'brand': ['Toyota', 'RAM', 'Ford', 'BMW', 'Mercedes-Benz'],
    'model': ['Land Cruiser Base', 'ProMaster 2500 Window Van High Roof', 'Mustang GT', '428 Gran Coupe i xDrive',
    'type': ['SUV', 'Van', 'Coupe', 'Sedan', 'Convertible'],
    'city_miles_per_gallon': [13, 15, 16, 27, 18],
    'highway_miles_per_gallon': [18, 20, 22, 34, 25]
}


df_dict = pd.DataFrame(data)

df_dict
```

	year	brand	model	type	city_miles_per_gallon	highway_miles_per_gallon
0	2016	Toyota	Land Cruiser Base	SUV	13	18
1	2014	RAM	ProMaster 2500 Window Van High Roof	Van	15	20
2	2002	Ford	Mustang GT	Coupe	16	22
3	2012	BMW	428 Gran Coupe i xDrive	Sedan	27	34
4	2008	Mercedes-Benz	SL-Class SL500 Roadster Convertible	Convertible	18	25

4) **Creating a dataframe from a list of dictionaries:**

```python
data = [
    {'First Name': 'Alice', 'Last Name': 'Smith', 'Age': 25, 'City': 'New York'},
    {'First Name': 'Bob', 'Last Name': 'Johnson', 'Age': 30, 'City': 'Los Angeles'},
    {'First Name': 'Charlie', 'Last Name': 'Brown', 'Age': 35, 'City': 'Chicago'},
    {'First Name': 'David', 'Last Name': 'Williams', 'Age': 40, 'City': 'Houston'},
    {'First Name': 'Eve', 'Last Name': 'Davis', 'Age': 45, 'City': 'Phoenix'}
]

import pandas as pd


df_list = pd.DataFrame(data)


df_list
```

	First Name	Last Name	Age	City
0	Alice	Smith	25	New York
1	Bob	Johnson	30	Los Angeles
2	Charlie	Brown	35	Chicago
3	David	Williams	40	Houston
4	Eve	Davis	45	Phoenix

5) **Selecting dataframe columns:**

single dataframe column:

```
df = pd.DataFrame(data)
df['brand']
```

	brand
0	Toyota
1	RAM
2	Ford
3	BMW
4	Mercedes-Benz

dtype: object

multiple dataframe columns:

```
[52] df[['year', 'brand', 'model']]
```

	year	brand	model
0	2016	Toyota	Land Cruiser Base
1	2014	RAM	ProMaster 2500 Window Van High Roof
2	2002	Ford	Mustang GT
3	2012	BMW	428 Gran Coupe
4	2008	Mercedes-Benz	SL-Class SL500

6) **Selecting dataframe rows:**

single dataframe row:

```
[55] single_row_loc = df.loc[2]
     single_row_loc
```

	2
year	2002
brand	Ford
model	Mustang GT
type	Coupe
city_miles_per_gallon	16
highway_miles_per_gallon	22

dtype: object

multiple dataframe rows:

```
multiple_rows_loc = df.loc[[1, 3, 4]]
multiple_rows_loc
```

	year	brand	model	type	city_miles_per_gallon	highway_miles_per_gallon
1	2014	RAM	ProMaster 2500 Window Van High Roof	Van	15	20
3	2012	BMW	428 Gran Coupe	Sedan	27	34
4	2008	Mercedes-Benz	SL-Class SL500	Convertible	18	25