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**:
Name: Masangkay, Frederick D.	**Instructor**: Ma'am Maria 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**

Laboratory Manual Activity: Creating and Accessing Panda DataFrames

```
[48] import pandas as pd
```

```
[49] from google.colab import files
```

```
[50] uploaded = files.upload()
```

Choose Files LoL_champ...s - 1.3.3.csv

- **LoL_champions - 1.3.3.csv**(text/csv) - 1872 bytes, last modified: 10/15/2024 - 100% done

Saving LoL_champions - 1.3.3.csv to LoL_champions - 1.3.3 (1).csv

```
data = pd.read_csv("LoL_champions - 1.3.3.csv")
```

2. **Creating a dataframe from your CSV file**

```
[61] LoL_champions = pd.DataFrame(data)
```

```
LoL_champions
```

	Name	Tags	Role	Range type	Resourse type	Base HP	Movement speed	Base armor	Base magic resistance	Attack range	HP regeneration	Attack damage	Attack speed
0	Aatrox	Tank, Fighter	Top	Melee	Blood Well	650	345	38	32	175	3.0	60	0.651
1	Alistar	Tank,Support	Support	Melee	Mana	685	330	47	32	125	8.5	62	0.625
2	Amumu	Tank,Support	Jungle	Melee	Mana	685	335	33	32	125	9.0	57	0.736
3	Blitzcrank	Tank, Support	Support	Melee	Mana	600	325	37	32	125	7.5	62	0.625
4	Braum	Tank, Support	Support	Melee	Mana	610	335	47	32	125	8.5	55	0.644
5	Camille	Tank,Fighter	Top	Melee	Mana	670	340	35	32	125	8.5	68	0.644
6	Cho'Gath	Tank,Fighter	Top	Melee	Mana	644	345	38	32	125	9.0	69	0.625
7	Darius	Tank,Fighter	Top	Melee	Mana	652	340	39	32	175	10.0	64	0.625
8	Dr. Mundo	Tank,Fighter	Top	Melee	NaN	613	345	32	29	125	7.0	61	0.670
9	Galio	Tank, Mage	Middle	Melee	Mana	632	340	24	32	150	8.0	59	0.625
10	Garen	Tank,Fighter	Top	Melee	NaN	690	340	38	32	175	8.0	69	0.625
11	Gnar	Tank,Fighter	Top	Ranged	Rage	540	335	32	30	175	4.5	57	0.625
12	Gragas	Tank,Fighter	Jungle	Melee	Mana	640	330	38	32	125	5.5	64	0.675
13	Renekton	Tank,Fighter	Top	Melee	Fury	660	345	35	28	125	8.0	69	0.665
14	Rengar	Tank,Fighter	Jungle	Melee	Ferocity	590	345	34	32	125	6.0	68	0.667
15	Riven	Tank,Fighter	Top	Melee	NaN	630	340	33	32	125	8.5	64	0.625
16	Sejuani	Tank,Fighter	Jungle	Melee	Mana	630	340	34	32	150	8.5	66	0.688
17	Sett	Tank,Fighter	Top	Melee	Grit	670	340	33	28	125	7.0	60	0.625
18	Shen	Tank,Fighter	Top	Melee	Energy	610	340	34	32	125	8.5	64	0.751
19	Shyvana	Tank,Fighter	Jungle	Melee	Fury	665	350	38	32	125	8.5	66	0.658
20	Singed	Tank,Fighter	Top	Melee	Mana	650	345	34	32	125	9.5	63	0.625
21	Sion	Tank,Fighter	Top	Melee	Mana	655	345	32	32	175	7.5	68	0.679
22	Skarner	Tank,Fighter	Jungle	Melee	Mana	610	335	33	32	150	7.5	60	0.625

3. **Creating a dataframe from a dictionary of lists**

```
# Creating a dictionaries of list from the original dataset
champion_data = {
    "Name": ["Aatrox", "Alistar", "Amumu", "Blitzcrank", "Braum"],
    # observation - Under the variable name the data are "Aatrox", "Alistar", "Amumu", "Blitzcrank", "Braum"
    "Tags": [["Tank", "Fighter"], ["Tank", "Support"], ["Tank", "Support"], ["Tank", "Support"], ["Tank", "Support"]],
    "Role": ["Top", "Support", "Jungle", "Support", "Support"],
    "Range type": ["Melee", "Melee", "Melee", "Melee", "Melee"],
    "Resource type": ["Blood Well", "Mana", "Mana", "Mana", "Mana"],
    "Base HP": [650, 685, 685, 600, 610],
    "Movement speed": [345, 330, 335, 325, 335],
    "Base armor": [38, 47, 33, 37, 47],
    "Base magic resistance": [32, 32, 32, 32, 32],
    "Attack range": [175, 125, 125, 125, 125],
    "HP regeneration": [3, 8.5, 9, 7.5, 8.5],
    "Attack damage": [60, 62, 57, 62, 55],
    "Attack speed": [0.651, 0.625, 0.736, 0.625, 0.644]
    # observation - Under the variable Attack speed the data are 0.651, 0.625, 0.736, 0.625, 0.644
}
```

```
[76] df = pd.DataFrame(champion_data)  # observation - creating avariable to store the data frame from champion_list
```

```
[77] df # Observation - Printing the df
```

	Name	Tags	Role	Range type	Resource type	Base HP	Movement speed	Base armor	Base magic resistance	Attack range	HP regeneration	Attack damage	Attack speed
0	Aatrox	[Tank, Fighter]	Top	Melee	Blood Well	650	345	38	32	175	3.0	60	0.651
1	Alistar	[Tank, Support]	Support	Melee	Mana	685	330	47	32	125	8.5	62	0.625
2	Amumu	[Tank, Support]	Jungle	Melee	Mana	685	335	33	32	125	9.0	57	0.736
3	Blitzcrank	[Tank, Support]	Support	Melee	Mana	600	325	37	32	125	7.5	62	0.625
4	Braum	[Tank, Support]	Support	Melee	Mana	610	335	47	32	125	8.5	55	0.644

4. Creating a dataframe from a list of dictionaries

```
[72] # Creating a list of dictionaries from the original dataset
champion_list = [
    {"Name": "Aatrox", "Tags": ["Tank", "Fighter"], "Role": "Top", "Range type": "Melee", "Resource type": "Blood Well", "Base HP": 650, "Movement speed": 345, "Base armor": 38, "Ba
    # observation - The list was consist of Name, Tags, Role, Range type, Resource type, Base Hp, and so on.
    {"Name": "Alistar", "Tags": ["Tank", "Support"], "Role": "Support", "Range type": "Melee", "Resource type": "Mana", "Base HP": 685, "Movement speed": 330, "Base armor": 47, "Bas
    {"Name": "Amumu", "Tags": ["Tank", "Support"], "Role": "Jungle", "Range type": "Melee", "Resource type": "Mana", "Base HP": 685, "Movement speed": 335, "Base armor": 33, "Base m
    {"Name": "Blitzcrank", "Tags": ["Tank", "Support"], "Role": "Support", "Range type": "Melee", "Resource type": "Mana", "Base HP": 600, "Movement speed": 325, "Base armor": 37, "
    {"Name": "Braum", "Tags": ["Tank", "Support"], "Role": "Support", "Range type": "Melee", "Resource type": "Mana", "Base HP": 610, "Movement speed": 335, "Base armor": 47, "Base
] # observation - The list of dictionaies was created
```

```
df_from_list = pd.DataFrame(champion_list) # observation - creating avariable to store the data frame from champion_list
```

```
[74] df_from_list # Observation - Printing the df_from_list
```

	Name	Tags	Role	Range type	Resource type	Base HP	Movement speed	Base armor	Base magic resistance	Attack range	HP regeneration	Attack damage	Attack speed
0	Aatrox	[Tank, Fighter]	Top	Melee	Blood Well	650	345	38	32	175	3.0	60	0.651
1	Alistar	[Tank, Support]	Support	Melee	Mana	685	330	47	32	125	8.5	62	0.625
2	Amumu	[Tank, Support]	Jungle	Melee	Mana	685	335	33	32	125	9.0	57	0.736
3	Blitzcrank	[Tank, Support]	Support	Melee	Mana	600	325	37	32	125	7.5	62	0.625
4	Braum	[Tank, Support]	Support	Melee	Mana	610	335	47	32	125	8.5	55	0.644

5. Selecting dataframe columns

```
[78] single_column = df_from_list['Name']
```

```
[81] single_column
```

	Name
0	Aatrox
1	Alistar
2	Amumu
3	Blitzcrank
4	Braum

dtype: object

```
[79] multiple_columns = df_from_list[['Name', 'Base HP', 'Attack damage']]
```

```
multiple_columns
```

	Name	Base HP	Attack damage
0	Aatrox	650	60
1	Alistar	685	62
2	Amumu	685	57
3	Blitzcrank	600	62
4	Braum	610	55

6. Selecting data frame rows

```
# Selecting a single row using pandas indexing (loc)
single_row_pandas = df_from_list.loc[2]  # Selecting the 3rd row (Amumu)
single_row_pandas
```

	2
Name	Amumu
Tags	[Tank, Support]
Role	Jungle
Range type	Melee
Resource type	Mana
Base HP	685
Movement speed	335
Base armor	33
Base magic resistance	32
Attack range	125
HP regeneration	9.0
Attack damage	57
Attack speed	0.736

```
# Selecting multiple rows using pandas indexing (loc)
multiple_rows_pandas = df_from_list.loc[1:4]  # Selecting the 2nd to 4th row (Alistar to Blitzcrank)
multiple_rows_pandas
```

	Name	Tags	Role	Range type	Resource type	Base HP	Movement speed	Base armor	Base magic resistance	Attack range	HP regeneration	Attack damage	Attack speed
1	Alistar	[Tank, Support]	Support	Melee	Mana	685	330	47	32	125	8.5	62	0.625
2	Amumu	[Tank, Support]	Jungle	Melee	Mana	685	335	33	32	125	9.0	57	0.736
3	Blitzcrank	[Tank, Support]	Support	Melee	Mana	600	325	37	32	125	7.5	62	0.625
4	Braum	[Tank, Support]	Support	Melee	Mana	610	335	47	32	125	8.5	55	0.644

```python
# Selecting a single row using Python indexing (index-based)
single_row_python = df_from_list[1:2]  # Selecting the 2nd row (Alistar)
single_row_python
```

	Name	Tags	Role	Range type	Resource type	Base HP	Movement speed	Base armor	Base magic resistance	Attack range	HP regeneration	Attack damage	Attack speed
1	Alistar	[Tank, Support]	Support	Melee	Mana	685	330	47	32	125	8.5	62	0.625

```python
[91] # Selecting multiple rows using Python indexing (index-based)
multiple_rows_python = df_from_list[1:4]  # Selecting rows from index 1 to 3 (Alistar to Blitzcrank)
multiple_rows_python
```

	Name	Tags	Role	Range type	Resource type	Base HP	Movement speed	Base armor	Base magic resistance	Attack range	HP regeneration	Attack damage	Attack speed
1	Alistar	[Tank, Support]	Support	Melee	Mana	685	330	47	32	125	8.5	62	0.625
2	Amumu	[Tank, Support]	Jungle	Melee	Mana	685	335	33	32	125	9.0	57	0.736
3	Blitzcrank	[Tank, Support]	Support	Melee	Mana	600	325	37	32	125	7.5	62	0.625