

08-Exercises

December 11, 2024

1 08 - Exercises: Pandas

This week we saw: - Tabular data analysis - The Pandas package

Here are some exercises to help you get comfortable with these concepts :)

1.1 Pandas Series

1.1.1 1. Creating a Series - 3 points

1. Create a Pandas Series with a list of 7 of your favourite foods.
2. Set the index to represent their order of preference.
3. Print the Series.

Example of expected output:

Favourite fruits Series:

```
1      peach
5      banana
3      mandarin
4      cherry
2      apricot
7      melon
6      fig
dtype: object
```

[]:

1.1.2 2. Sorting and filtering - 3 points

1. Sort the Series in alphabetical order.
2. Sort the Series according to increasing index values.
3. Filter out any items whose names start with the letters “A”, “B” or “C”
(**Hint:** https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#boolean-indexing)

Example of expected output

Filtered fruits (not a/b/c):

```
1      peach
3      mandarin
7      melon
```

```
6         fig
dtype: object
```

```
[ ]:
```

1.1.3 3. Statistical analysis - 3 points

1. Create a Series of 50 random integers between 1 and 100 (Hint: use NumPy).
2. Calculate the mean, median, and standard deviation.
3. Find the maximum and minimum values.
4. Find the indexes of the maximum and minimum values.

Example of expected output:

```
Random numbers Series (first 5):
```

```
0    39
1     6
2     4
3    90
4    29
```

```
dtype: int64
```

```
Mean:    52.02
```

```
Median:  50.0
```

```
St.d.:   26.825581759482755
```

```
Max:     98
```

```
Min:     4
```

```
Max value: 98 located at index: 5
```

```
Min value: 4 located at index: 2
```

```
[ ]:
```

1.2 Pandas DataFrame

1.2.1 4. Creating a Series - 3 points

1. Create a DataFrame with data about 7 people (can be fictional or real). Include columns for Name, Age, Favourite_colour, and Number (a random number between 1-100).

```
[ ]:
```

1.2.2 5. Basic manipulations - 3 points

1. Add a new column, Fortune, where the value is `True` if Number is strictly greater than 50, otherwise `False`.
2. Rename the Number column to `Lucky_number`.
3. Print the DataFrame.

Example of expected output:

Added "Fortune" column:

Name	Age	Favourite_colour	Number	Fortune	
0	Lady Gaga	44	Red	73	True
1	Little Simz	22	Yellow	76	True
2	Charlie xcx	33	Green	22	False
3	Joan Thiele	30	Purple	16	False
4	Domiziana	35	Red	68	True
5	Hiromi	55	Blue	58	True
6	Marina	50	Blue	91	True

Renamed "Number" to "Lucky_number":

Name	Age	Favourite_colour	Lucky_number	Fortune	
0	Lady Gaga	44	Red	73	True
1	Little Simz	22	Yellow	76	True
2	Charlie xcx	33	Green	22	False
3	Joan Thiele	30	Purple	16	False
4	Domiziana	35	Red	68	True
5	Hiromi	55	Blue	58	True
6	Marina	50	Blue	91	True

[]:

1.2.3 6. Filtering and querying - 3 points

1. Select rows where the Favorite_Color is "Red".
2. Select only the Name and Lucky_number columns for people who are above 35 years old.

Example of expected output:

People who like Red:

Name	Age	Favourite_colour	Lucky_number	Fortune	
0	Lady Gaga	44	Red	55	True
4	Domiziana	35	Red	12	False

People older than 35:

Name	Lucky_number	
0	Lady Gaga	55
5	Hiromi	22
6	Marina	61

[]:

1.2.4 7. Sorting - 3 points

1. Sort the DataFrame by Age in descending order.
2. Sort the DataFrame by Favourite_colour in alphabetical order **and** by Lucky_number in ascending order (Hint: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_values.html).

Example of expected output:

Sorted by Age (desc):

Name	Age	Favourite_colour	Lucky_number	Fortune	
5	Hiromi	55	Blue	22	False
6	Marina	50	Blue	61	True
0	Lady Gaga	44	Red	55	True
4	Domiziana	35	Red	12	False
2	Charlie xcx	33	Green	38	False
3	Joan Thiele	30	Purple	44	False
1	Little Simz	22	Yellow	81	True

Sorted by Lucky_number and Colour:

Name	Age	Favourite_colour	Lucky_number	Fortune	
5	Hiromi	55	Blue	22	False
6	Marina	50	Blue	61	True
2	Charlie xcx	33	Green	38	False
3	Joan Thiele	30	Purple	44	False
4	Domiziana	35	Red	12	False
0	Lady Gaga	44	Red	55	True
1	Little Simz	22	Yellow	81	True

[]:

1.2.5 8. Statistical analysis - 3 points

1. Calculate the average Lucky_number of all people.
2. Identify the person with the highest Lucky_number.

Example of expected output:

Average Lucky_number: 44.714285714285715

Person with highest Lucky_number:

Name	Little Simz
Age	22
Favourite_colour	Yellow
Lucky_number	81
Fortune	True

Name: 1, dtype: object

[]:

1.2.6 9. Group and aggregate - 3 points

1. Group the DataFrame by Favorite_Color and calculate the average Lucky_number for each color.
2. Count how many people have each Favorite_Color.

Example of expected output:

Average Lucky_number by Favourite_colour:

```
Favourite_colour
Blue      41.5
Green     38.0
Purple    44.0
Red       33.5
Yellow    81.0
Name: Lucky_number, dtype: float64
```

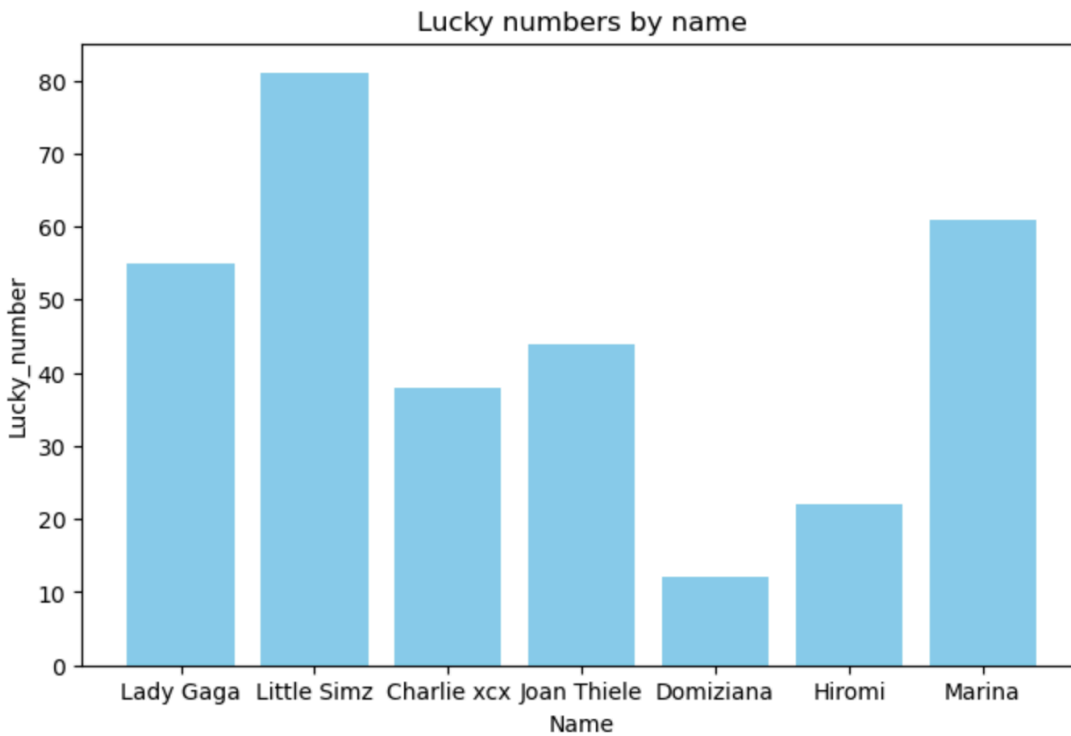
```
Counts of Favourite_colour:
Favourite_colour
Red      2
Blue     2
Yellow   1
Green    1
Purple   1
Name: count, dtype: int64
```

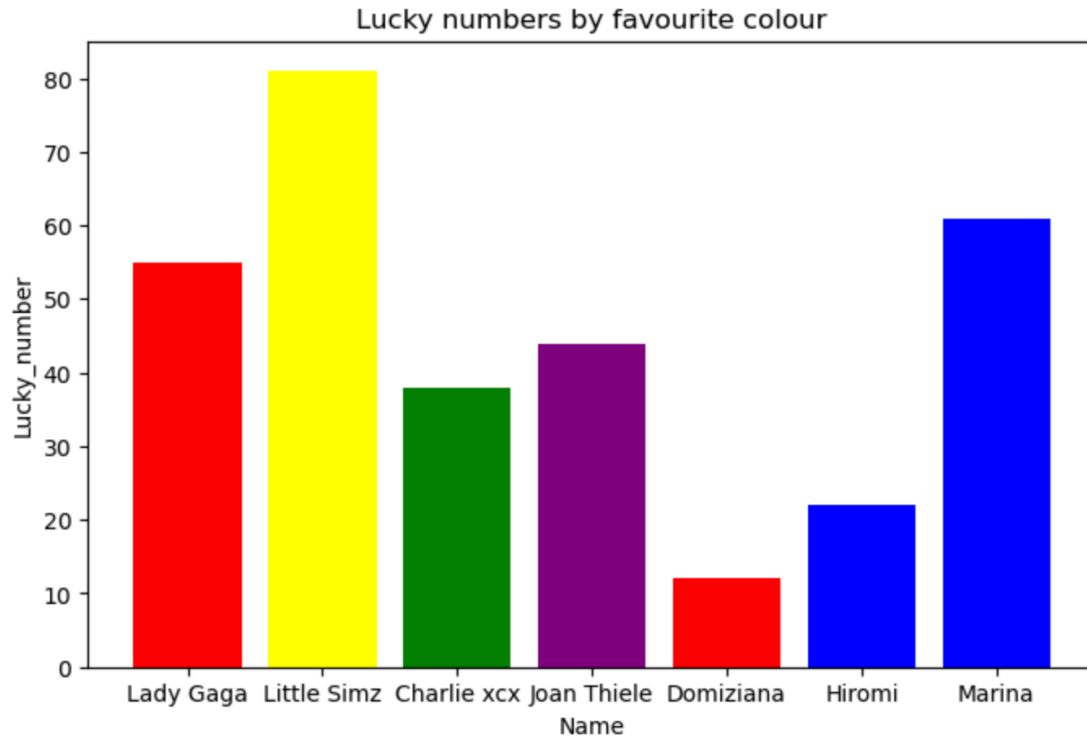
```
[ ]:
```

1.2.7 10. Datavizzz - 3 points

1. Use matplotlib to create a bar plot showing the Lucky_number of each person.
2. Colour each bar according to the Favourite_colour of that person.

Example of expected output:





[]: