Python Workshop II

November 26, 2023

1 Python and Data Analysis

1.1 Import Libraries

First, we need to import some common data analysis libraries.

[31]:
```
import pandas as pd
import numpy as np
```

1.2 Import Dataset

Read data set from `.csv` file

[32]:
```
df = pd.read_csv("./data/example.csv", encoding="GBK")
df
```

[32]:

	date	rank	song	artist	last-week	peak-rank	weeks-on-board
0	2021-11-06	1	Easy On Me	Adele	1.0	1	3
1	2021-11-06	2	Stay	The Kid LAROI & Justin Bieber	2.0	1	16
2	2021-11-06	3	Industry Baby	Lil Nas X & Jack Harlow	3.0	1	14
3	2021-11-06	4	Fancy Like	Walker Hayes	4.0	3	19
4	2021-11-06	5	Bad Habits	Ed Sheeran	5.0	2	18
…	…	…	…	…	…	…	…
330082	1958-08-04	96	Over And Over	Thurston Harris	NaN	96	1
330083	1958-08-04	97	I Believe In You	Robert & Johnny	NaN	97	1
330084	1958-08-04	98	Little Serenade	The Ames Brothers	NaN	98	1
330085	1958-08-04	99	I'll Get By (As Long As I Have You)				
330086	1958-08-04	100	Judy				

```
330085                Billy Williams        NaN         99                1
330086              Frankie Vaughan         NaN        100                1

[330087 rows x 7 columns]
```

1.3 Data Match

After we import the dataset, we can easily concentrate on one column. For example, if we want to find the data which "artist" value = "Ed Sheeran", we can

[33]:
```
df[df["artist"] == "Ed Sheeran"]
```

[33]:
```
             date  rank        song      artist  last-week  peak-rank  \
4      2021-11-06     5  Bad Habits  Ed Sheeran        5.0          2
6      2021-11-06     7     Shivers  Ed Sheeran        9.0          7
104    2021-10-30     5  Bad Habits  Ed Sheeran        4.0          2
108    2021-10-30     9     Shivers  Ed Sheeran       10.0          9
203    2021-10-23     4  Bad Habits  Ed Sheeran        5.0          2
…             …   …          …           …          …          …
48084  2012-08-25    85  The A Team  Ed Sheeran       92.0         85
48191  2012-08-18    92  The A Team  Ed Sheeran       99.0         92
48298  2012-08-11    99  The A Team  Ed Sheeran       98.0         95
48397  2012-08-04    98  The A Team  Ed Sheeran       95.0         95
48494  2012-07-28    95  The A Team  Ed Sheeran        NaN         95

       weeks-on-board
4                  18
6                   7
104                17
108                 6
203                16
…                   …
48084               5
48191               4
48298               3
48397               2
48494               1

[397 rows x 7 columns]
```

Similarly, if we need find the song with the largest "weeks-on-board" value,

[34]:
```
df[df["weeks-on-board"] == np.max(df["weeks-on-board"])]
```

[34]:
```
           date  rank             song      artist  last-week  peak-rank  \
919  2021-09-04    20  Blinding Lights  The Weeknd       21.0          1

     weeks-on-board
```

```
919             90
```

1.4 Data Clean

Usually, the dataset can contain some missing values or error values, which will impede our analysis. Therefore, we always need to clean the data first.

1.4.1 Missing Values

We can check the missing values by the following function,

[35]:
```
def check_missing_v1(df):
    for column in df:
        print([df[df[column].isnull()].index, column])
    return


def check_missing_v2(df):
    for column in df:
        if df[df[column].isnull()].index.size > 0:
            print([df[df[column].isnull()].index, column])
    return


check_missing_v1(df)
# check_missing_v2(df)
```

```
[Index([], dtype='int64'), 'date']
[Index([], dtype='int64'), 'rank']
[Index([], dtype='int64'), 'song']
[Index([], dtype='int64'), 'artist']
[Index([    26,     27,     60,     68,     78,     86,     87,     88,     89,
           95,
       …
       330077, 330078, 330079, 330080, 330081, 330082, 330083, 330084, 330085,
       330086],
      dtype='int64', length=32312), 'last-week']
[Index([], dtype='int64'), 'peak-rank']
[Index([], dtype='int64'), 'weeks-on-board']
```

1.4.2 Error Values

We can edit the values or just delete the values in the dataset.

If we need to change a value,

[36]:
```
# df.loc[5, "last-week"] = 7.0
df.loc[5]
```

[36]:
```
date                                        2021-11-06
rank                                                 6
song                                        Way 2 Sexy
artist            Drake Featuring Future & Young Thug
last-week                                          6.0
peak-rank                                            1
weeks-on-board                                       8
Name: 5, dtype: object
```

If we need to add one column “year”,

[37]:
```
df.insert(1, "year", [int(date[:4]) for date in df["date"]])
df
```

[37]:

	date	year	rank	song	artist	last-week	peak-rank	weeks-on-board
0	2021-11-06	2021	1	Easy On Me	Adele	1.0	1	3
1	2021-11-06	2021	2	Stay	The Kid LAROI & Justin Bieber	2.0	1	16
2	2021-11-06	2021	3	Industry Baby	Lil Nas X & Jack Harlow	3.0	1	14
3	2021-11-06	2021	4	Fancy Like	Walker Hayes	4.0	3	19
4	2021-11-06	2021	5	Bad Habits	Ed Sheeran	5.0	2	18
…	…	…	…	…	…	…	…	…
330082	1958-08-04	1958	96	Over And Over	Thurston Harris	NaN	96	1
330083	1958-08-04	1958	97	I Believe In You	Robert & Johnny	NaN	97	1
330084	1958-08-04	1958	98	Little Serenade	The Ames Brothers	NaN	98	1
330085	1958-08-04	1958	99	I'll Get By (As Long As I Have You)	Billy Williams	NaN	99	1
330086	1958-08-04	1958	100	Judy	Frankie Vaughan	NaN	100	1

[330087 rows x 8 columns]

If we don’t need “year” column, then

[38]:
```
df.drop(columns="year")
# df = df.drop(columns="year")
# df
```

[38]:

	date	rank	song	artist	last-week	peak-rank	weeks-on-board
0	2021-11-06	1	Easy On Me	Adele	1.0	1	3
1	2021-11-06	2	Stay	The Kid LAROI & Justin Bieber	2.0	1	16
2	2021-11-06	3	Industry Baby	Lil Nas X & Jack Harlow	3.0	1	14
3	2021-11-06	4	Fancy Like	Walker Hayes	4.0	3	19
4	2021-11-06	5	Bad Habits	Ed Sheeran	5.0	2	18
…	…	…	…	…	…	…	…
330082	1958-08-04	96	Over And Over	Thurston Harris	NaN	96	1
330083	1958-08-04	97	I Believe In You	Robert & Johnny	NaN	97	1
330084	1958-08-04	98	Little Serenade	The Ames Brothers	NaN	98	1
330085	1958-08-04	99	I'll Get By (As Long As I Have You)	Billy Williams	NaN	99	1
330086	1958-08-04	100	Judy	Frankie Vaughan	NaN	100	1

[330087 rows x 7 columns]

If we think *Alan Walker* is not needed, then

[39]:
```
df[df["artist"]=="Alan Walker"]
```

[39]:

	date	year	rank	song	artist	last-week	peak-rank	weeks-on-board
27999	2016-07-02	2016	100	Faded	Alan Walker	92.0	80	8
28091	2016-06-25	2016	92	Faded	Alan Walker	88.0	80	7
28187	2016-06-18	2016	88	Faded	Alan Walker	80.0	80	6
28279	2016-06-11	2016	80	Faded	Alan Walker	97.0	80	5
28396	2016-06-04	2016	97	Faded	Alan Walker	96.0	91	4
28495	2016-05-28	2016	96	Faded	Alan Walker	NaN	91	3
28793	2016-05-07	2016	94	Faded	Alan Walker	91.0	91	2
28890	2016-04-30	2016	91	Faded	Alan Walker	NaN	91	1

```
28495              3
28793              2
28890              1
```

[40]:
```
df_noAW = df.drop(df[df["artist"]=="Alan Walker"].index)
df_noAW[df_noAW["artist"]=="Alan Walker"]
```

[40]:
```
Empty DataFrame
Columns: [date, year, rank, song, artist, last-week, peak-rank, weeks-on-board]
Index: []
```

1.5 Data Exploration

We can use `DataFrame` and `numpy` to do some basic exploration.

[41]:
```
df.shape
```

[41]:
```
(330087, 8)
```

[42]:
```
df.describe()
```

[42]:
```
                year           rank      last-week      peak-rank  \
count  330087.000000  330087.000000  297775.000000  330087.000000
mean     1989.725142      50.500929      47.591631      40.970629
std        18.266426      28.866094      28.054360      29.347481
min      1958.000000       1.000000       1.000000       1.000000
25%      1974.000000      26.000000      23.000000      13.000000
50%      1990.000000      51.000000      47.000000      38.000000
75%      2006.000000      76.000000      72.000000      65.000000
max      2021.000000     100.000000     100.000000     100.000000

       weeks-on-board
count   330087.000000
mean         9.161785
std          7.618264
min          1.000000
25%          4.000000
50%          7.000000
75%         13.000000
max         90.000000
```

[43]:
```
df.dtypes
```

[43]:
```
date              object
year               int64
rank               int64
song              object
artist            object
```

```
last-week         float64
peak-rank           int64
weeks-on-board      int64
dtype: object
```

Here are some additional methods that can give you statistics of a DataFrame or particular column in a DataFrame. - `.mean(axis=0 [will give you the calculated value per column])` - returns the statistical mean - `.median(axis=0 [will give you the calculated value per column])` - returns the statistical median - `.mode(axis=0 [will give you the calculated value per column])` - returns the statistical mode - `.count()` - gives number of total values in column - `.unique()` - returns array of all unique values in that column - `.value_counts()` - returns object containing counts of unique values

[44]:
```
df.artist.unique()
```

[44]:
```
array(['Adele', 'The Kid LAROI & Justin Bieber',
       'Lil Nas X & Jack Harlow', …, "The Daddy-O's", 'Thurston Harris',
       'Frankie Vaughan'], dtype=object)
```

We notice that there are some collaborations, for *Taylor Swift*,

[45]:
```
TS = []
for artist in df["artist"]:
    if "Taylor Swift" in artist:
        TS.append(artist)
print(list(set(TS)))
```

```
['Big Red Machine Featuring Taylor Swift', 'Zayn / Taylor Swift', 'Tim McGraw
With Taylor Swift', 'Taylor Swift Featuring Colbie Caillat', 'Taylor Swift
Featuring Maren Morris', 'Sugarland Featuring Taylor Swift', 'B.o.B Featuring
Taylor Swift', 'Taylor Swift Featuring Kendrick Lamar', 'Taylor Swift', 'Taylor
Swift Featuring HAIM', 'Taylor Swift Featuring Dixie Chicks', 'Taylor Swift
Featuring Ed Sheeran', 'Taylor Swift Featuring Bon Iver', 'Taylor Swift
Featuring The National', 'Taylor Swift Featuring Ed Sheeran & Future', 'Taylor
Swift Featuring The Civil Wars', 'Boys Like Girls Featuring Taylor Swift',
'Taylor Swift Featuring Brendon Urie']
```

1.6 Visualization

Using `Matplotlib` we can easily get some visuals.

Details are in *Keyuan*'s Part.

2 Useful Links

- Python for Data Analysis, 3E -Wes McKinney
- Python Pandas
- Python Matplotlib