Day_38_041223

January 23, 2024

```
[150]: import numpy as np
       import pandas as pd
       movies = pd.read_csv("movies.csv",index_col=0)
       directors = pd.read_csv("directors.csv",index_col=0)
       data = pd.merge(movies,directors,left_on="director_id",right_on='id',how='left')
       data.drop('id_y',axis=1,inplace=True)
       data.rename({"id_x":"movies_id"},axis=1,inplace=True)
       data
```

```
[150]:       movies_id     budget  popularity     revenue  \
       0         43597  237000000         150  2787965087
       1         43598  300000000         139   961000000
       2         43599  245000000         107   880674609
       3         43600  250000000         112  1084939099
       4         43602  258000000         115   890871626
       …             …          …           …           …
       1460      48363          0           3      321952
       1461      48370      27000          19     3151130
       1462      48375          0           7           0
       1463      48376          0           3           0
       1464      48395     220000          14     2040920

                                                title  vote_average  vote_count  \
       0                                       Avatar           7.2       11800
       1     Pirates of the Caribbean: At World's End           6.9        4500
       2                                      Spectre           6.3        4466
       3                        The Dark Knight Rises           7.6        9106
       4                                 Spider-Man 3           5.9        3576
       …                                          …           …           …
       1460                            The Last Waltz           7.9          64
       1461                                    Clerks           7.4         755
       1462                                   Rampage           6.0         131
       1463                                   Slacker           6.4          77
       1464                               El Mariachi           6.6         238

             director_id  year month       day      director_name gender
       0            4762  2009   Dec  Thursday      James Cameron   Male
       1            4763  2007   May  Saturday     Gore Verbinski   Male
```

```
2             4764  2015   Oct    Monday         Sam Mendes   Male
3             4765  2012   Jul    Monday  Christopher Nolan   Male
4             4767  2007   May   Tuesday          Sam Raimi   Male
…             …   …    …      …                  …    …
1460          4809  1978   May    Monday    Martin Scorsese   Male
1461          5369  1994   Sep   Tuesday        Kevin Smith   Male
1462          5148  2009   Aug    Friday           Uwe Boll   Male
1463          5535  1990   Jul    Friday  Richard Linklater   Male
1464          5097  1992   Sep    Friday   Robert Rodriguez    NaN

[1465 rows x 13 columns]
```

1 How the multi indexing works

[151]:
```
data_agg = data.groupby('director_name')[['title','year']].aggregate({'title':
  ↪'count','year':['min','max']})
data_agg
```

[151]:
```
                            title  year
                            count   min   max
director_name
Adam McKay                      6  2004  2015
Adam Shankman                   8  2001  2012
Alejandro González Iñárritu     6  2000  2015
Alex Proyas                     5  1994  2016
Alexander Payne                 5  1999  2013
…                           …    …    …
Wes Craven                     10  1984  2011
Wolfgang Petersen               7  1981  2006
Woody Allen                    18  1977  2013
Zack Snyder                     7  2004  2016
Zhang Yimou                     6  2002  2014

[199 rows x 3 columns]
```

[152]:
```
data.columns
```

[152]:
```
Index(['movies_id', 'budget', 'popularity', 'revenue', 'title', 'vote_average',
       'vote_count', 'director_id', 'year', 'month', 'day', 'director_name',
       'gender'],
      dtype='object')
```

[153]:
```
data_agg.columns
```

[153]:
```
MultiIndex([('title', 'count'),
            ( 'year',   'min'),
            ( 'year',   'max')],
          )
```

2 Changing the Multi index to Single index

[154]:
```
data_agg.columns = ['_'.join(tuple) for tuple in data_agg.columns]
```

[155]:
```
data_agg
```

[155]:
```
                             title_count  year_min  year_max
director_name
Adam McKay                             6      2004      2015
Adam Shankman                          8      2001      2012
Alejandro González Iñárritu            6      2000      2015
Alex Proyas                            5      1994      2016
Alexander Payne                        5      1999      2013
…                                    …         …         …
Wes Craven                            10      1984      2011
Wolfgang Petersen                      7      1981      2006
Woody Allen                           18      1977      2013
Zack Snyder                            7      2004      2016
Zhang Yimou                            6      2002      2014

[199 rows x 3 columns]
```

3 Cleaning the Data using Pandas

- When we have more columns and less rows it is called Fat Data
- When we have more rows and less columns it is called Thin Data

[156]:
```
!gdown 173A59xh2mnpmljCCB9bhC4C5eP2IS6qZ
```

```
Downloading…
From: https://drive.google.com/uc?id=173A59xh2mnpmljCCB9bhC4C5eP2IS6qZ
To: C:\Data\Data_science\Data Science RIA\3 Python\Pandas\Codes\Pfizer_1.csv

  0%|          | 0.00/1.51k [00:00<?, ?B/s]
100%|##########| 1.51k/1.51k [00:00<?, ?B/s]
```

[157]:
```
data = pd.read_csv("Pfizer_1.csv")
```

[158]:
```
data.columns
```

[158]:
```
Index(['Date', 'Drug_Name', 'Parameter', '1:30:00', '2:30:00', '3:30:00',
       '4:30:00', '5:30:00', '6:30:00', '7:30:00', '8:30:00', '9:30:00',
       '10:30:00', '11:30:00', '12:30:00'],
      dtype='object')
```

4 Example of Fat data

```
[159]: data.head()
```

```
[159]:         Date                 Drug_Name    Parameter  1:30:00  2:30:00  \
       0  15-10-2020  diltiazem hydrochloride  Temperature     23.0     22.0
       1  15-10-2020  diltiazem hydrochloride     Pressure     12.0     13.0
       2  15-10-2020       docetaxel injection  Temperature      NaN     17.0
       3  15-10-2020       docetaxel injection     Pressure      NaN     22.0
       4  15-10-2020   ketamine hydrochloride  Temperature     24.0      NaN

          3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
       0      NaN     21.0     21.0       22     23.0     21.0     22.0        20
       1      NaN     11.0     13.0       14     16.0     16.0     24.0        18
       2     18.0      NaN     17.0       18      NaN      NaN     23.0        23
       3     22.0      NaN     22.0       23      NaN      NaN     27.0        26
       4      NaN     27.0      NaN       26     25.0     24.0     23.0        22

          11:30:00  12:30:00
       0      20.0        21
       1      19.0        20
       2      25.0        25
       3      29.0        28
       4      21.0        20
```

```
[160]: data.shape
```

```
[160]: (18, 15)
```

```
[161]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Date        18 non-null     object
 1   Drug_Name   18 non-null     object
 2   Parameter   18 non-null     object
 3   1:30:00     16 non-null     float64
 4   2:30:00     16 non-null     float64
 5   3:30:00     12 non-null     float64
 6   4:30:00     14 non-null     float64
 7   5:30:00     16 non-null     float64
 8   6:30:00     18 non-null     int64
 9   7:30:00     16 non-null     float64
 10  8:30:00     14 non-null     float64
 11  9:30:00     16 non-null     float64
 12  10:30:00   18 non-null     int64
 13  11:30:00   16 non-null     float64
 14  12:30:00   18 non-null     int64
dtypes: float64(9), int64(3), object(3)
memory usage: 2.2+ KB
```

5 To convert the fat data into thin data

- Pandas has a function named melt

[162]:
```
data_melt = pd.melt(data,␣
↪id_vars=['Date','Drug_Name','Parameter'],var_name='Time',value_name='Reading')
```

[163]:
```
data_melt.shape
```

[163]:
```
(216, 5)
```

[164]:
```
data_melt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 216 entries, 0 to 215
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       216 non-null    object
 1   Drug_Name  216 non-null    object
 2   Parameter  216 non-null    object
 3   Time       216 non-null    object
 4   Reading    190 non-null    float64
dtypes: float64(1), object(4)
memory usage: 8.6+ KB
```

6 Change thin data into fat data

- Pandas has a function called Pivot

[165]:
```
data_melt.pivot(index=['Date','Drug_Name','Parameter'],columns = 'Time',values␣
↪= 'Reading').reset_index()
```

[165]:
```
Time        Date                Drug_Name    Parameter  10:30:00  11:30:00  \
0     15-10-2020  diltiazem hydrochloride     Pressure      18.0      19.0
1     15-10-2020  diltiazem hydrochloride  Temperature      20.0      20.0
2     15-10-2020      docetaxel injection     Pressure      26.0      29.0
3     15-10-2020      docetaxel injection  Temperature      23.0      25.0
4     15-10-2020   ketamine hydrochloride     Pressure       9.0       9.0
5     15-10-2020   ketamine hydrochloride  Temperature      22.0      21.0
6     16-10-2020  diltiazem hydrochloride     Pressure      24.0       NaN
7     16-10-2020  diltiazem hydrochloride  Temperature      40.0       NaN
```

```
8     16-10-2020      docetaxel injection     Pressure      28.0      29.0
9     16-10-2020      docetaxel injection  Temperature      56.0      57.0
10    16-10-2020   ketamine hydrochloride     Pressure      16.0      17.0
11    16-10-2020   ketamine hydrochloride  Temperature      13.0      14.0
12    17-10-2020  diltiazem hydrochloride     Pressure      11.0      13.0
13    17-10-2020  diltiazem hydrochloride  Temperature      14.0      11.0
14    17-10-2020      docetaxel injection     Pressure      28.0      29.0
15    17-10-2020      docetaxel injection  Temperature      21.0      22.0
16    17-10-2020   ketamine hydrochloride     Pressure      13.0      14.0
17    17-10-2020   ketamine hydrochloride  Temperature      22.0      23.0

Time  12:30:00  1:30:00  2:30:00  3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  \
0         20.0     12.0     13.0      NaN     11.0     13.0     14.0     16.0
1         21.0     23.0     22.0      NaN     21.0     21.0     22.0     23.0
2         28.0      NaN     22.0     22.0      NaN     22.0     23.0      NaN
3         25.0      NaN     17.0     18.0      NaN     17.0     18.0      NaN
4         11.0      8.0      NaN      NaN      7.0      NaN      9.0     10.0
5         20.0     24.0      NaN      NaN     27.0      NaN     26.0     25.0
6         27.0     18.0     19.0     20.0     21.0     22.0     23.0     24.0
7         42.0     34.0     35.0     36.0     36.0     37.0     38.0     37.0
8         30.0     23.0     24.0      NaN     25.0     26.0     27.0     28.0
9         58.0     46.0     47.0      NaN     48.0     48.0     49.0     50.0
10        18.0     12.0     12.0     13.0      NaN     15.0     15.0     15.0
11        15.0      8.0      9.0     10.0      NaN     11.0     12.0     12.0
12        14.0      3.0      4.0      4.0      4.0      6.0      8.0      9.0
13        10.0     20.0     19.0     19.0     18.0     17.0     16.0     15.0
14        28.0     20.0     22.0     22.0     22.0     22.0     23.0     25.0
15        23.0     12.0     13.0     14.0     15.0     16.0     17.0     18.0
16        15.0      8.0      9.0     10.0     11.0     11.0     12.0     12.0
17        24.0     13.0     14.0     15.0     16.0     17.0     18.0     19.0

Time  8:30:00  9:30:00
0        16.0     24.0
1        21.0     22.0
2         NaN     27.0
3         NaN     23.0
4        11.0     10.0
5        24.0     23.0
6        25.0     25.0
7        38.0     39.0
8        29.0     28.0
9        52.0     55.0
10       15.0      NaN
11       11.0      NaN
12        NaN      9.0
13        NaN     13.0
14       26.0     27.0
```

```
15        19.0      20.0
16        11.0      12.0
17        20.0      21.0
```

7 Removing the NULL Values

```
[166]: data_melt.head()
```

```
[166]:          Date              Drug_Name    Parameter     Time  Reading
       0  15-10-2020  diltiazem hydrochloride  Temperature  1:30:00     23.0
       1  15-10-2020  diltiazem hydrochloride     Pressure  1:30:00     12.0
       2  15-10-2020      docetaxel injection  Temperature  1:30:00      NaN
       3  15-10-2020      docetaxel injection     Pressure  1:30:00      NaN
       4  15-10-2020   ketamine hydrochloride  Temperature  1:30:00     24.0
```

```
[167]: data_tidy = data_melt.
        ↪pivot(index=['Date','Drug_Name','Time'],columns='Parameter',values='Reading').
        ↪reset_index()
```

```
[168]: data_tidy.head()
```

```
[168]: Parameter        Date                Drug_Name      Time  Pressure  \
       0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
       1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
       2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
       3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
       4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0

       Parameter  Temperature
       0                 20.0
       1                 20.0
       2                 21.0
       3                 23.0
       4                 22.0
```

8 Understanding the NULL and None values

```
[169]: type(None)
```

```
[169]: NoneType
```

```
[170]: type(np.nan)
```

```
[170]: float
```

```
[171]: pd.Series([1,np.nan,2])
```

```
[171]: 0    1.0
       1    NaN
       2    2.0
       dtype: float64
```

```
[172]: a = pd.Series(['1','np.nan',2,None])
       type(a[2])
```

```
[172]: int
```

```
[173]: pd.Series([1,2,3,4,5,np.nan])
```

```
[173]: 0    1.0
       1    2.0
       2    3.0
       3    4.0
       4    5.0
       5    NaN
       dtype: float64
```

```
[174]: pd.Series([1,2,3,None])
```

```
[174]: 0    1.0
       1    2.0
       2    3.0
       3    NaN
       dtype: float64
```

9 How to deal with NULL values

9.0.1 Check whether there are null values

```
[175]: data.isnull().sum(axis=1)
```

```
[175]: 0     1
       1     1
       2     4
       3     4
       4     3
       5     3
       6     1
       7     1
       8     1
       9     1
       10    2
       11    2
       12    1
       13    1
```

```
14    0
15    0
16    0
17    0
dtype: int64
```

9.0.2 Dropping the null values

[176]:
```
data.dropna(axis=0)
```

[176]:
```
          Date              Drug_Name    Parameter  1:30:00  2:30:00  \
14  17-10-2020     docetaxel injection  Temperature     12.0     13.0
15  17-10-2020     docetaxel injection     Pressure     20.0     22.0
16  17-10-2020  ketamine hydrochloride  Temperature     13.0     14.0
17  17-10-2020  ketamine hydrochloride     Pressure      8.0      9.0

    3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
14     14.0     15.0     16.0       17     18.0     19.0     20.0        21
15     22.0     22.0     22.0       23     25.0     26.0     27.0        28
16     15.0     16.0     17.0       18     19.0     20.0     21.0        22
17     10.0     11.0     11.0       12     12.0     11.0     12.0        13

    11:30:00  12:30:00
14      22.0        23
15      29.0        28
16      23.0        24
17      14.0        15
```

9.0.3 Filling the null values with 0

[179]:
```
data.fillna(0)
```

[179]:
```
          Date               Drug_Name    Parameter  1:30:00  2:30:00  \
0   15-10-2020  diltiazem hydrochloride  Temperature     23.0     22.0
1   15-10-2020  diltiazem hydrochloride     Pressure     12.0     13.0
2   15-10-2020      docetaxel injection  Temperature      0.0     17.0
3   15-10-2020      docetaxel injection     Pressure      0.0     22.0
4   15-10-2020   ketamine hydrochloride  Temperature     24.0      0.0
5   15-10-2020   ketamine hydrochloride     Pressure      8.0      0.0
6   16-10-2020  diltiazem hydrochloride  Temperature     34.0     35.0
7   16-10-2020  diltiazem hydrochloride     Pressure     18.0     19.0
8   16-10-2020      docetaxel injection  Temperature     46.0     47.0
9   16-10-2020      docetaxel injection     Pressure     23.0     24.0
10  16-10-2020   ketamine hydrochloride  Temperature      8.0      9.0
11  16-10-2020   ketamine hydrochloride     Pressure     12.0     12.0
12  17-10-2020  diltiazem hydrochloride  Temperature     20.0     19.0
13  17-10-2020  diltiazem hydrochloride     Pressure      3.0      4.0
```

```
14  17-10-2020      docetaxel injection  Temperature     12.0     13.0
15  17-10-2020      docetaxel injection     Pressure     20.0     22.0
16  17-10-2020   ketamine hydrochloride  Temperature     13.0     14.0
17  17-10-2020   ketamine hydrochloride     Pressure      8.0      9.0

    3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
0       0.0     21.0     21.0       22     23.0     21.0     22.0        20
1       0.0     11.0     13.0       14     16.0     16.0     24.0        18
2      18.0      0.0     17.0       18      0.0      0.0     23.0        23
3      22.0      0.0     22.0       23      0.0      0.0     27.0        26
4       0.0     27.0      0.0       26     25.0     24.0     23.0        22
5       0.0      7.0      0.0        9     10.0     11.0     10.0         9
6      36.0     36.0     37.0       38     37.0     38.0     39.0        40
7      20.0     21.0     22.0       23     24.0     25.0     25.0        24
8       0.0     48.0     48.0       49     50.0     52.0     55.0        56
9       0.0     25.0     26.0       27     28.0     29.0     28.0        28
10     10.0      0.0     11.0       12     12.0     11.0      0.0        13
11     13.0      0.0     15.0       15     15.0     15.0      0.0        16
12     19.0     18.0     17.0       16     15.0      0.0     13.0        14
13      4.0      4.0      6.0        8      9.0      0.0      9.0        11
14     14.0     15.0     16.0       17     18.0     19.0     20.0        21
15     22.0     22.0     22.0       23     25.0     26.0     27.0        28
16     15.0     16.0     17.0       18     19.0     20.0     21.0        22
17     10.0     11.0     11.0       12     12.0     11.0     12.0        13

    11:30:00  12:30:00
0       20.0        21
1       19.0        20
2       25.0        25
3       29.0        28
4       21.0        20
5        9.0        11
6        0.0        42
7        0.0        27
8       57.0        58
9       29.0        30
10      14.0        15
11      17.0        18
12      11.0        10
13      13.0        14
14      22.0        23
15      29.0        28
16      23.0        24
17      14.0        15
```

9.0.4 Fill the NULL Values with Average

[181]:
```
data['2:30:00'].fillna(data['2:30:00'].mean())
```

[181]:
```
0     22.0000
1     13.0000
2     17.0000
3     22.0000
4     18.8125
5     18.8125
6     35.0000
7     19.0000
8     47.0000
9     24.0000
10     9.0000
11    12.0000
12    19.0000
13     4.0000
14    13.0000
15    22.0000
16    14.0000
17     9.0000
Name: 2:30:00, dtype: float64
```

[ ]:
```
# def replace_nan(x):
#     return x['Drug_Name']['.mean()
```

[ ]:
```

```