Practice - PANDASMoviesCities

September 27, 2024

1 PANDAS PRACTICE 2

1.0.1 In this practice we will look at weather data from various cities and see how groupby can be used to run some analytics. Add code cells where applicable.

[1]:
```
import pandas as pd
```

Question 1

Let us explore the movie dataset

1. Load in the IMDB movies dataset
2. Display the top 5 and last 5 movies and columns
3. Display information about the columns. What are the datatypes?
4. Append the datframe to itself
5. Display the shape of the dataframe
6. Remove the duplicates
7. Confirm that the shape has been modified

[2]:
```
#reading dataset using read_csv.
imdb=pd.read_csv("IMDB-Movie-Data.csv")
```

[3]:
```
#2. Display the top 5 and last 5 movies and columns

#top 5
imdb.head(5)
```

[3]:
```
   Rank                    Title                       Genre  \
0     1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1     2               Prometheus  Adventure,Mystery,Sci-Fi
2     3                    Split           Horror,Thriller
3     4                     Sing   Animation,Comedy,Family
4     5           Suicide Squad  Action,Adventure,Fantasy

                                         Description             Director  \
0  A group of intergalactic criminals are forced …           James Gunn
1  Following clues to the origin of mankind, a te…         Ridley Scott
2  Three girls are kidnapped by a man with a diag…  M. Night Shyamalan
```

```
3  In a city of humanoid animals, a hustling thea…  Christophe Lourdelet
4  A secret government agency recruits some of th…            David Ayer

                                              Actors  Year  Runtime (Minutes)  \
0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014                121
1  Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012                124
2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016                117
3  Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016                108
4  Will Smith, Jared Leto, Margot Robbie, Viola D…  2016                123

   Rating   Votes  Revenue (Millions)  Metascore
0     8.1  757074              333.13       76.0
1     7.0  485820              126.46       65.0
2     7.3  157606              138.12       62.0
3     7.2   60545              270.32       59.0
4     6.2  393727              325.02       40.0
```

[4]:
```
#last 5
imdb.tail(5)
```

[4]:
```
     Rank                   Title                   Genre  \
995   996    Secret in Their Eyes    Crime,Drama,Mystery
996   997         Hostel: Part II                  Horror
997   998  Step Up 2: The Streets    Drama,Music,Romance
998   999            Search Party       Adventure,Comedy
999  1000              Nine Lives  Comedy,Family,Fantasy

                                          Description          Director  \
995  A tight-knit team of rising investigators, alo…        Billy Ray
996  Three American college students studying abroa…         Eli Roth
997  Romantic sparks occur between two dance studen…        Jon M. Chu
998  A pair of friends embark on a mission to reuni…    Scot Armstrong
999  A stuffy businessman finds himself trapped ins…  Barry Sonnenfeld

                                            Actors  Year  \
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts…  2015
996  Lauren German, Heather Matarazzo, Bijou Philli…  2007
997  Robert Hoffman, Briana Evigan, Cassie Ventura,…  2008
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh…  2014
999  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch…  2016

     Runtime (Minutes)  Rating  Votes  Revenue (Millions)  Metascore
995                111     6.2  27585                NaN       45.0
996                 94     5.5  73152              17.54       46.0
997                 98     6.2  70699              58.01       50.0
998                 93     5.6   4881                NaN       22.0
999                 87     5.3  12435              19.64       11.0
```

[5]:
```
#using info to understand the dataset and datatypes
imdb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Rank                1000 non-null   int64
 1   Title               1000 non-null   object
 2   Genre               1000 non-null   object
 3   Description         1000 non-null   object
 4   Director            1000 non-null   object
 5   Actors              1000 non-null   object
 6   Year                1000 non-null   int64
 7   Runtime (Minutes)   1000 non-null   int64
 8   Rating              1000 non-null   float64
 9   Votes               1000 non-null   int64
 10  Revenue (Millions)  872 non-null    float64
 11  Metascore           936 non-null    float64
dtypes: float64(3), int64(4), object(5)
memory usage: 93.9+ KB
```

[6]:
```
#Append the datframe to itself

#using concat to append dataframe with itself
imdb_append= pd.concat([imdb,imdb])
imdb_append
```

[6]:

	Rank	Title	Genre	\
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	
1	2	Prometheus	Adventure,Mystery,Sci-Fi	
2	3	Split	Horror,Thriller	
3	4	Sing	Animation,Comedy,Family	
4	5	Suicide Squad	Action,Adventure,Fantasy	
..	…	…	…	
995	996	Secret in Their Eyes	Crime,Drama,Mystery	
996	997	Hostel: Part II	Horror	
997	998	Step Up 2: The Streets	Drama,Music,Romance	
998	999	Search Party	Adventure,Comedy	
999	1000	Nine Lives	Comedy,Family,Fantasy	

	Description	Director	\
0	A group of intergalactic criminals are forced …	James Gunn	
1	Following clues to the origin of mankind, a te…	Ridley Scott	
2	Three girls are kidnapped by a man with a diag…	M. Night Shyamalan	
3	In a city of humanoid animals, a hustling thea…	Christophe Lourdelet	

```
4    A secret government agency recruits some of th…           David Ayer
..                                                 …                  …
995  A tight-knit team of rising investigators, alo…           Billy Ray
996  Three American college students studying abroa…            Eli Roth
997  Romantic sparks occur between two dance studen…           Jon M. Chu
998  A pair of friends embark on a mission to reuni…      Scot Armstrong
999  A stuffy businessman finds himself trapped ins…    Barry Sonnenfeld

                                                Actors  Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014
1    Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012
2    James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016
3    Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016
4    Will Smith, Jared Leto, Margot Robbie, Viola D…  2016
..                                                 …     …
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts…  2015
996  Lauren German, Heather Matarazzo, Bijou Philli…  2007
997  Robert Hoffman, Briana Evigan, Cassie Ventura,…  2008
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh…  2014
999  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch…  2016

     Runtime (Minutes)  Rating   Votes  Revenue (Millions)  Metascore
0                  121     8.1  757074              333.13       76.0
1                  124     7.0  485820              126.46       65.0
2                  117     7.3  157606              138.12       62.0
3                  108     7.2   60545              270.32       59.0
4                  123     6.2  393727              325.02       40.0
..                 …     …     …                   …         …
995                111     6.2   27585                 NaN       45.0
996                 94     5.5   73152               17.54       46.0
997                 98     6.2   70699               58.01       50.0
998                 93     5.6    4881                 NaN       22.0
999                 87     5.3   12435               19.64       11.0

[2000 rows x 12 columns]
```

[7]:
```
#5. Display the shape of the dataframe
imdb_append.shape
#there are 2000 rows and 12 columns in the dataframe imdb_append
```

[7]:
```
(2000, 12)
```

[8]:
```
#6. Remove the duplicates

#using drop_duplicates function to drop duplicates in the dataframe imdb_append.
imdb_new=imdb_append.drop_duplicates()
#checking the new dataframe to ensure that duplicates are dropped
```

```
imdb_new.shape

#as the shape is 1000 rows and 12 columns, which is same as original dataset,␣
 ↪this ensures that the duplicates are dropped
```

[8]: (1000, 12)

Question 2

Let us explore another dataset. This time the weather dataset

1. Create the data frame from the given csv file
2. Display the first 10 rows
3. Display the last 5 rows
4. Display the datatypes
5. Display statistics for a numerical column

[9]:
```
#reading dataset using read_csv.
weather_dataset=pd.read_csv("weather_by_cities.csv")
weather_dataset
```

[9]:

	day	city	temperature	windspeed	event
0	1/1/2017	new york	32	6	Rain
1	1/2/2017	new york	36	7	Sunny
2	1/3/2017	new york	28	12	Snow
3	1/4/2017	new york	33	7	Sunny
4	1/1/2017	mumbai	90	5	Sunny
5	1/2/2017	mumbai	85	12	Fog
6	1/3/2017	mumbai	87	15	Fog
7	1/4/2017	mumbai	92	5	Rain
8	1/1/2017	paris	45	20	Sunny
9	1/2/2017	paris	50	13	Cloudy
10	1/3/2017	paris	54	8	Cloudy
11	1/4/2017	paris	42	10	Cloudy

[10]:
```
#Display the first 10 rows
weather_dataset.head(10)
```

[10]:

	day	city	temperature	windspeed	event
0	1/1/2017	new york	32	6	Rain
1	1/2/2017	new york	36	7	Sunny
2	1/3/2017	new york	28	12	Snow
3	1/4/2017	new york	33	7	Sunny
4	1/1/2017	mumbai	90	5	Sunny
5	1/2/2017	mumbai	85	12	Fog
6	1/3/2017	mumbai	87	15	Fog
7	1/4/2017	mumbai	92	5	Rain

```
 8  1/1/2017     paris           45          20   Sunny
 9  1/2/2017     paris           50          13  Cloudy
```

[11]:
```
#Display the last 5 rows
weather_dataset.tail(5)
```

[11]:
```
          day    city  temperature  windspeed   event
7   1/4/2017  mumbai           92          5    Rain
8   1/1/2017   paris           45         20   Sunny
9   1/2/2017   paris           50         13  Cloudy
10  1/3/2017   paris           54          8  Cloudy
11  1/4/2017   paris           42         10  Cloudy
```

[12]:
```
#Display the datatypes
weather_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   day          12 non-null     object
 1   city         12 non-null     object
 2   temperature  12 non-null     int64 
 3   windspeed    12 non-null     int64 
 4   event        12 non-null     object
dtypes: int64(2), object(3)
memory usage: 612.0+ bytes
```

[13]:
```
#5. Display statistics for a numerical column
#from the above result, windspeed and temperature are the int datatypes

#determining the statistical values for the windspeed column
weather_dataset['windspeed'].describe()
```

[13]:
```
count    12.000000
mean     10.000000
std       4.572646
min       5.000000
25%       6.750000
50%       9.000000
75%      12.250000
max      20.000000
Name: windspeed, dtype: float64
```

[14]:
```
#determining the statistical values for the temperature column
weather_dataset['temperature'].describe()
```

[14]:
```
count    12.000000
mean     56.166667
std      25.044808
min      28.000000
25%      35.250000
50%      47.500000
75%      85.500000
max      92.000000
Name: temperature, dtype: float64
```

Question 3 For this dataset let us determine the following. We will explore splitting your dataset in smaller groups and then applying an operation (such as min or max) to get aggregate result is called Split-Apply-Combine approach.

[15]:
```
# Groupby city and print the data for all the groups
# Get the data group for Mumbai
# Get the max temp for all cities
# What is the avarage temperature and windspeed
# Display all the analytics for the data
# Let us do a rudimentary plot. See code below
'''
%matplotlib inline # load matplotlib
variable.plot()

'''
```

[15]:
```
'\n%matplotlib inline # load matplotlib\nvariable.plot()\n\n'
```

[16]:
```
# Groupby city and print the data for all the groups
cities = weather_dataset.groupby('city').apply(lambda x: x)
cities
```

[16]:

		day	city	temperature	windspeed	event
city						
mumbai	4	1/1/2017	mumbai	90	5	Sunny
	5	1/2/2017	mumbai	85	12	Fog
	6	1/3/2017	mumbai	87	15	Fog
	7	1/4/2017	mumbai	92	5	Rain
new york	0	1/1/2017	new york	32	6	Rain
	1	1/2/2017	new york	36	7	Sunny
	2	1/3/2017	new york	28	12	Snow
	3	1/4/2017	new york	33	7	Sunny
paris	8	1/1/2017	paris	45	20	Sunny
	9	1/2/2017	paris	50	13	Cloudy
	10	1/3/2017	paris	54	8	Cloudy
	11	1/4/2017	paris	42	10	Cloudy

[18]:
```
# Get the data group for Mumbai
#applying lambda function to apply the filter of city = mumbai
mumbai_weather = weather_dataset.groupby('city').apply(lambda x:␣
 ↪x[x['city']=='mumbai'])
mumbai_weather
```

[18]:

		day	city	temperature	windspeed	event
city						
mumbai	4	1/1/2017	mumbai	90	5	Sunny
	5	1/2/2017	mumbai	85	12	Fog
	6	1/3/2017	mumbai	87	15	Fog
	7	1/4/2017	mumbai	92	5	Rain

[34]:
```
# Get the max temp for all cities

#using lambda function to apply the max function on temperature.
grouped_cities = weather_dataset.groupby('city').apply(lambda x:␣
 ↪x[x['temperature'] == x['temperature'].max()])
grouped_cities
```

[34]:

		day	city	temperature	windspeed	event
city						
mumbai	7	1/4/2017	mumbai	92	5	Rain
new york	1	1/2/2017	new york	36	7	Sunny
paris	10	1/3/2017	paris	54	8	Cloudy

[20]:
```
# What is the avarage temperature and windspeed
avg_temp = weather_dataset['temperature'].mean()
avg_windspeed = weather_dataset['windspeed'].mean()

print('Average Temperature: ',avg_temp,'\n','Average Windspeed: ',avg_windspeed)
```

```
Average Temperature:  56.166666666666664
 Average Windspeed:  10.0
```

[27]:
```
# Display all the analytics for the data
#using describe to get the statistics of the dataset for each city
weather_analytics=weather_dataset.groupby('city').describe()
weather_analytics
```

[27]:

	temperature								\
	count	mean	std	min	25%	50%	75%	max	
city									
mumbai	4.0	88.50	3.109126	85.0	86.50	88.5	90.50	92.0	
new york	4.0	32.25	3.304038	28.0	31.00	32.5	33.75	36.0	
paris	4.0	47.75	5.315073	42.0	44.25	47.5	51.00	54.0	

	windspeed							
	count	mean	std	min	25%	50%	75%	max
city								
mumbai	4.0	9.25	5.057997	5.0	5.00	8.5	12.75	15.0
new york	4.0	8.00	2.708013	6.0	6.75	7.0	8.25	12.0
paris	4.0	12.75	5.251984	8.0	9.50	11.5	14.75	20.0

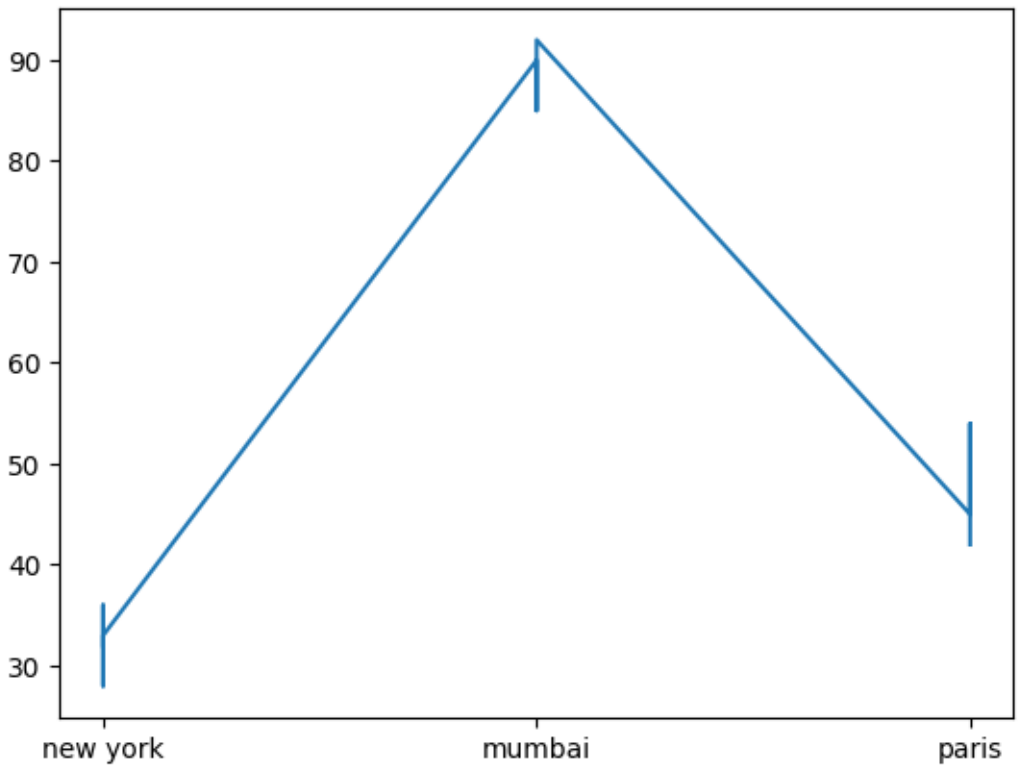
[35]:
```
#overall analytics of the dataset
grouped_cities.describe()
```

[35]:

	temperature	windspeed
count	3.000000	3.000000
mean	60.666667	6.666667
std	28.589042	1.527525
min	36.000000	5.000000
25%	45.000000	6.000000
50%	54.000000	7.000000
75%	73.000000	7.500000
max	92.000000	8.000000

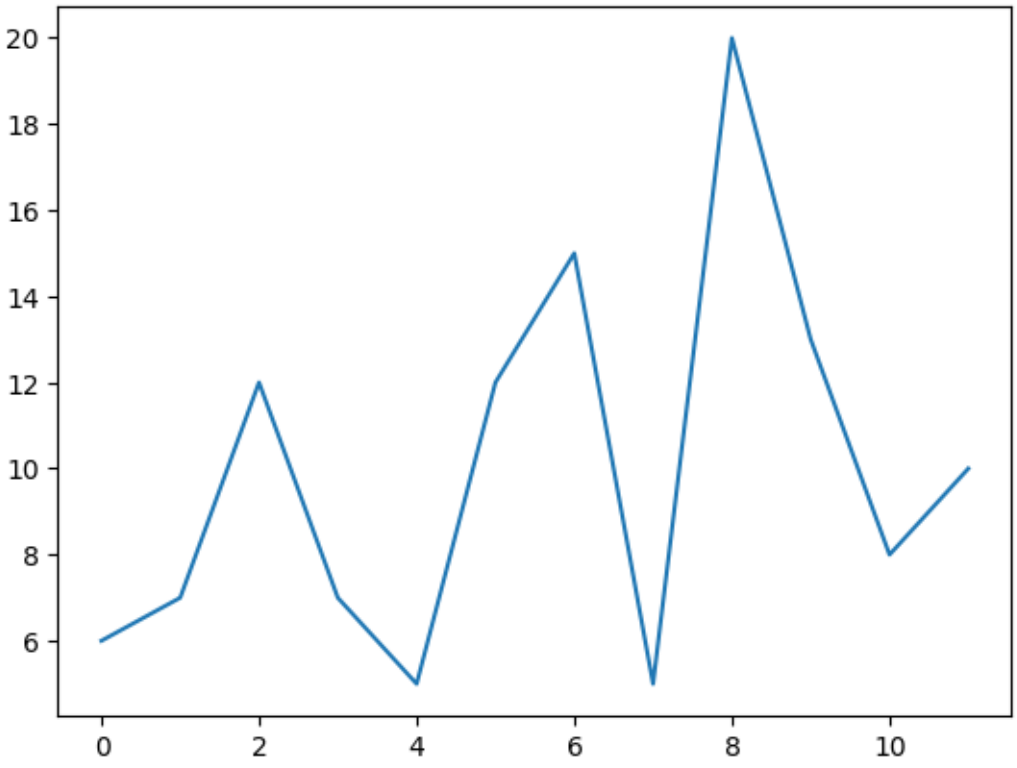
[36]:
```
# Let us do a rudimentary plot. See code below
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(weather_dataset['city'],weather_dataset['temperature'])
```

[36]:
```
[<matplotlib.lines.Line2D at 0x7fa5e15be4d0>]
```



[37]:
```
weather_dataset['windspeed'].plot()
```

[37]: <Axes: >


20
18
16
14
12
10
8
6
0
2
4
6
8
10