Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exercise 6 - Pandas

Name: Ooi Caaron

IC Number: 990701-07-5837

Date : 26/6/23

Introduction : Pandas can create a data frame for you to view the data structure

Conclusion : Still need to practice more

EXERCISE 6

Pandas

In [1]:

```
import pandas as pd
```

Question 1

UFO (Unidentified Flying Objects) data is from http://www.nuforc.org/webreports.html (http://www.nuforc.org/webreports.html).

Given the UFO sightings data and the path for csv file, read the csv file from the URL.

In [3]:

```
path ='http://bit.ly/uforeports'
```

use the read_csv to read the csv and assign to a variable called 'ufo'

In [70]:

```
#read csv
ufo = pd.read_csv(path)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
ufo
```

Out[70]:

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00
3	Abilene	NaN	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00
5	Valley City	NaN	DISK	ND	9/15/1934 15:30
6	Crater Lake	NaN	CIRCLE	CA	6/15/1935

In [5]:

```
#show the top 5 rows of data
ufo.head()
```

Out[5]:

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00
3	Abilene	NaN	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00

In [7]:

```
#show the last 10 rows of data
ufo.tail(10)
```

Out[7]:

	City	Colors Reported	Shape Reported	State	Time
18231	Pismo Beach	NaN	OVAL	CA	12/31/2000 20:00
18232	Lodi	NaN	NaN	WI	12/31/2000 20:30
18233	Anchorage	RED	VARIOUS	AK	12/31/2000 21:00
18234	Capitola	NaN	TRIANGLE	CA	12/31/2000 22:00
18235	Fountain Hills	NaN	NaN	AZ	12/31/2000 23:00
18236	Grant Park	NaN	TRIANGLE	IL	12/31/2000 23:00
18237	Spirit Lake	NaN	DISK	IA	12/31/2000 23:00
18238	Eagle River	NaN	NaN	WI	12/31/2000 23:45
18239	Eagle River	RED	LIGHT	WI	12/31/2000 23:45
18240	Ybor	NaN	OVAL	FL	12/31/2000 23:59

In [9]:

```
#check the data type
type(ufo)
```

Out[9]:

```
pandas.core.frame.DataFrame
```

In [18]:

```
#check the data type for 'Time' column
time_col = ufo['Time']
time_col.dtype
```

Out[18]:

```
dtype('O')
```

In [27]:

```
#show all rows for the column 'City'
city_col = len(ufo['City'].unique())
city_col
```

Out[27]:

```
6477
```

In [29]:

```
#determine the shape of the data
ufo.shape
```

Out[29]:

```
(18241, 5)
```

In [45]:

```
#show all data for 'City' that starts with 'E'
city_data_starting_with_e = ufo[ufo['City'].str.startswith('E', na=False)]
print(city_data_starting_with_e['City'])
```

```
8               Eklutna
55             Espanola
109           Excelsior
140      East Palestine
179           Evergreen
              ...
18182        Evansville
18215          El Campo
18224           Eufaula
18238       Eagle River
18239       Eagle River
Name: City, Length: 557, dtype: object
```

In [61]:

```
# count number of reported cases for 'CIRCLE'
# and count the number of reported cases for 'LIGHT'
circle = ufo[ufo['Shape Reported'].fillna('').str.contains('CIRCLE', case=False)].shape[0
light = len(ufo[ufo['Shape Reported']=='LIGHT'])
print(circle)
print(light)
```

```
1365
2803
```

In [124]:

```
#determine what are the top three colors reported

ufo['Colors Reported'].value_counts().head(3)
```

Out[124]:

```
RED       780
GREEN     531
ORANGE    528
Name: Colors Reported, dtype: int64
```

In [71]:

```
#count the number of shape reported by state and city
shape_counts = ufo.groupby(['State', 'City'])['Shape Reported'].size()
print(shape_counts)
```

```
State  City
AK     Adak
1
       Alaska
2
       Anchorage
12
       Arctic
1
       Auke Bay
2
       Bering Sea
1
       Bethel
1
       Big Lake
1
       Cantwell
1
```

Question 2

IMDB ratings are from http://www.imdb.com/search/title?groups=top_1000&sort=user_rating&view=simple (http://www.imdb.com/search/title?groups=top_1000&sort=user_rating&view=simple)

Given the IMDB movies dataset and path, use the read_csv to read the data and assign to a variable 'movies'

In [72]:

```
path = 'http://bit.ly/imdbratings'
```

In [73]:

```
#read the dataset
movies = pd.read_csv(path)
movies
```

Out[73]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....

In [74]:

```
#show the top 10 rows
movies.head(10)
```

Out[74]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
5	8.9	12 Angry Men	NOT RATED	Drama	96	[u'Henry Fonda', u'Lee J. Cobb', u'Martin Bals...
6	8.9	The Good, the Bad and the Ugly	NOT RATED	Western	161	[u'Clint Eastwood', u'Eli Wallach', u'Lee Van ...
7	8.9	The Lord of the Rings: The Return of the King	PG-13	Adventure	201	[u'Elijah Wood', u'Viggo Mortensen', u'Ian McK...
8	8.9	Schindler's List	R	Biography	195	[u'Liam Neeson', u'Ralph Fiennes', u'Ben Kings...
9	8.9	Fight Club	R	Drama	139	[u'Brad Pitt', u'Edward Norton', u'Helena Bonh...

In [76]:

```
#show summary of the dataset
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 979 entries, 0 to 978
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   star_rating     979 non-null    float64
 1   title           979 non-null    object 
 2   content_rating  976 non-null    object 
 3   genre           979 non-null    object 
 4   duration        979 non-null    int64  
 5   actors_list     979 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 46.0+ KB
```

In [77]:

```
#determine the dimension of the dataset
movies.shape
```

Out[77]:

```
(979, 6)
```

In [79]:

```
#show the data types for each coloumns
movies.dtypes
```

Out[79]:

```
star_rating       float64
title              object
content_rating     object
genre              object
duration            int64
actors_list        object
dtype: object
```

In [86]:

```
#show all the column names
# movie = [i for i in movies]
movies.columns
# Index(['star_rating', 'title', 'content_rating', 'genre', 'duration',
#        'actors_list'],
#       dtype='object')
```

Out[86]:

```
Index(['star_rating', 'title', 'content_rating', 'genre', 'duration',
       'actors_list'],
      dtype='object')
```

In [87]:

```
# rename the following columns, 'star_rating' as 'stars_rating'
# 'content_rating' as 'content'
movies.rename(columns={
        'star_rating': 'stars_rating',
        'content_rating': 'content'
    }, inplace=True)


movies.columns
```

Out[87]:

```
Index(['stars_rating', 'title', 'content', 'genre', 'duration', 'actors_li
st'], dtype='object')
```

In [89]:

```
#given a list called 'col_names' change all the columns names to 'col_names'
```

Out[89]:

```
Index(['star_rating', 'title', 'content_rating', 'genre', 'duration',
       'actors_list'],
      dtype='object')
```

In [89]:

```
#provide the statistical summary for 'star_rating' and 'duration'
movies.describe()
```

Out[89]:

	stars_rating	duration
count	979.000000	979.000000
mean	7.889785	120.979571
std	0.336069	26.218010
min	7.400000	64.000000
25%	7.600000	102.000000
50%	7.800000	117.000000
75%	8.100000	134.000000
max	9.300000	242.000000

In [91]:

```
#show the data type for column 'star_rating'
star_col = movies['stars_rating']
star_col.dtype
```

Out[91]:

```
dtype('float64')
```

In [94]:

```
#show the 5th row data for column 'content_rating'
movies['content'][4]
```

Out[94]:

```
'R'
```

In [98]:

```
#show all rows for movies duration more than 200 mins
movies[movies['duration'] > 200]
```

Out[98]:

	stars_rating	title	content	genre	duration	actors_list
7	8.9	The Lord of the Rings: The Return of the King	PG-13	Adventure	201	[u'Elijah Wood', u'Viggo Mortensen', u'Ian McK...
17	8.7	Seven Samurai	UNRATED	Drama	207	[u'Toshir\xf4 Mifune', u'Takashi Shimura', u'K...
78	8.4	Once Upon a Time in America	R	Crime	229	[u'Robert De Niro', u'James Woods', u'Elizabet...
85	8.4	Lawrence of Arabia	PG	Adventure	216	[u"Peter O'Toole", u'Alec Guinness', u'Anthony...
142	8.3	Lagaan: Once Upon a Time in India	PG	Adventure	224	[u'Aamir Khan', u'Gracy Singh', u'Rachel Shell...
157	8.2	Gone with the Wind	G	Drama	238	[u'Clark Gable', u'Vivien Leigh', u'Thomas Mit...
204	8.1	Ben-Hur	G	Adventure	212	[u'Charlton Heston', u'Jack Hawkins', u'Stephe...
445	7.9	The Ten Commandments	APPROVED	Adventure	220	[u'Charlton Heston', u'Yul Brynner', u'Anne Ba...
476	7.8	Hamlet	PG-13	Drama	242	[u'Kenneth Branagh', u'Julie Christie', u'Dere...
630	7.7	Malcolm X	PG-13	Biography	202	[u'Denzel Washington', u'Angela Bassett', u'De...
767	7.6	It's a Mad, Mad, Mad, Mad World	APPROVED	Action	205	[u'Spencer Tracy', u'Milton Berle', u'Ethel Me...

In [96]:

```
#what is the average movie duration
movies['duration'].mean()
```

Out[96]:

```
120.97957099080695
```

In [114]:

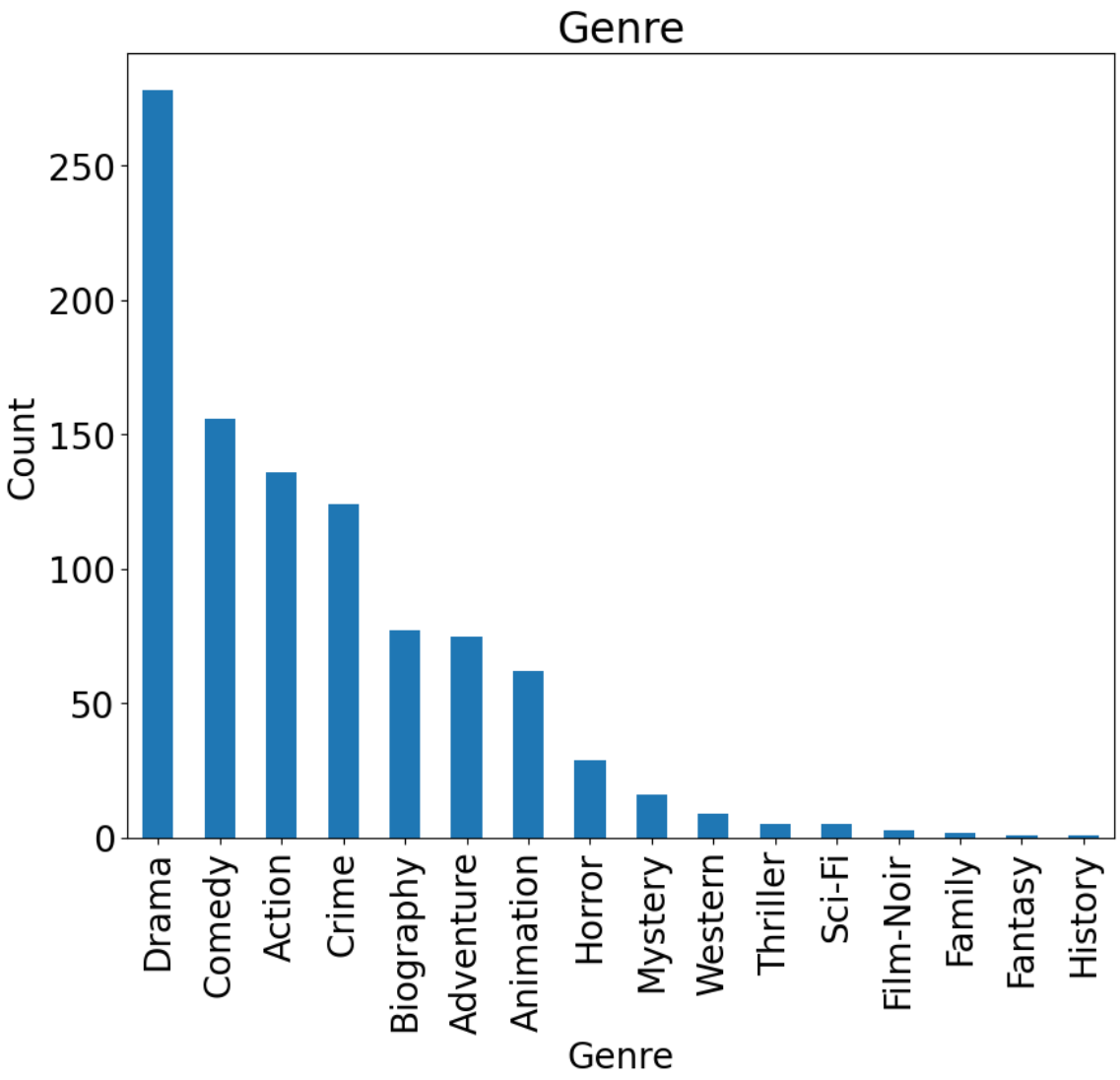
```
#count the number of movies where actor 'Charlton Heston' acted in
len(movies[movies['actors_list'].str.contains('Charlton Heston')])
```

Out[114]:

4

In [121]:

```
#what are the top 5 genre movies
# movieValue = movies.groupby('genre')['genre'].value_counts()
# movieValue.sort_values().iloc[::-1].plot(kind='bar', title='Genre');
genre_counts = movies['genre'].value_counts() #sum the number of each genre
genre_counts.sort_values().iloc[::-1].plot(kind='bar', title='Genre') #sort by descending
plt.xlabel('Genre')
plt.ylabel('Count')
plt.show()
```



In [134]:

```
#which movie has the highest rating
# movies['title'].head(1)
highest_rating_movie = movies[movies['stars_rating'] == movies['stars_rating'].max()]
highest_rating_movie['title']
```

Out[134]:

```
0    The Shawshank Redemption
Name: title, dtype: object
```

In [130]:

```
#which movie has the lowest rating
lowest_rating_movie = movies[movies['stars_rating'] == movies['stars_rating'].min()]
lowest_rating_movie
```

Out[130]:

	stars_rating	title	content	genre	duration	actors_list
930	7.4	Man on the Moon	R	Biography	118	[u'Jim Carrey', u'Danny DeVito', u'Gerry Becker']
931	7.4	Mean Streets	R	Crime	112	[u'Robert De Niro', u'Harvey Keitel', u'David ...
932	7.4	Harry Potter and the Order of the Phoenix	PG-13	Adventure	138	[u'Daniel Radcliffe', u'Emma Watson', u'Rupert...
933	7.4	Beetlejuice	PG	Comedy	92	[u'Alec Baldwin', u'Geena Davis', u'Michael Ke...
934	7.4	Crazy, Stupid, Love.	PG-13	Comedy	118	[u'Steve Carell', u'Ryan Gosling', u'Julianne ...

In [136]:

```
# group by genre and content rating and calculate the mean for duration

movies.groupby(['genre', 'content'])['duration'].mean()
```

Out[136]:

genre	content	
Action	APPROVED	143.333333
	G	178.000000
	GP	144.000000
	NOT RATED	129.500000
	PASSED	98.000000
	PG	119.727273
	PG-13	130.204545
	R	123.850746
	UNRATED	110.666667
Adventure	APPROVED	158.333333
	G	162.000000
	NOT RATED	113.200000
	PASSED	102.000000
	PG	133.952381
	PG-13	143.913043
	R	124.882353
	UNRATED	136.000000
Animation	APPROVED	84.666667
	G	93.150000
	NOT RATED	91.000000
	PG	99.360000
	PG-13	104.200000
	R	101.000000
	UNRATED	89.000000
Biography	APPROVED	111.000000
	G	143.000000
	GP	172.000000
	NOT RATED	96.000000
	PG	126.000000
	PG-13	133.241379
	R	132.138889
Comedy	APPROVED	108.333333
	G	86.000000
	GP	91.000000
	NC-17	95.000000
	NOT RATED	129.875000
	PASSED	83.666667
	PG	100.956522
	PG-13	106.565217
	R	107.561644
	UNRATED	103.750000
	X	84.000000
Crime	APPROVED	102.833333
	NC-17	106.000000
	NOT RATED	109.428571
	PASSED	107.000000
	PG	108.166667
	PG-13	120.750000

In [ ]: