ExplorePandaSeries

November 17, 2016

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

In [2]: movies = pd.read_csv('http://bit.ly/imdbratings')

In [3]: movies.head()
```

Out[3]:

	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 [4]: movies.dtypes
```

Out[4]:
```
star_rating       float64
title              object
content_rating     object
genre              object
duration            int64
actors_list        object
dtype: object
```

```
In [5]: movies.genre.describe() # shows count, unique, top, frequency of most comm
```

Out[5]:
```
count       979
unique       16
top       Drama
freq        278
Name: genre, dtype: object
```

```
In [6]: movies.genre.value_counts()
        # count of value and freq it appears

Out[6]: Drama        278
        Comedy       156
        Action       136
        Crime        124
        Biography     77
        Adventure     75
        Animation     62
        Horror        29
        Mystery       16
        Western        9
        Sci-Fi         5
        Thriller       5
        Film-Noir      3
        Family         2
        Fantasy        1
        History        1
        Name: genre, dtype: int64

In [7]: movies.genre.value_counts(normalize=True)
        # normalizes values

Out[7]: Drama        0.283963
        Comedy       0.159346
        Action       0.138917
        Crime        0.126660
        Biography    0.078652
        Adventure    0.076609
        Animation    0.063330
        Horror       0.029622
        Mystery      0.016343
        Western      0.009193
        Sci-Fi       0.005107
        Thriller     0.005107
        Film-Noir    0.003064
        Family       0.002043
        Fantasy      0.001021
        History      0.001021
        Name: genre, dtype: float64

In [8]: type(movies.genre.value_counts(normalize=True)) # type is Series

Out[8]: pandas.core.series.Series

In [10]: (movies.genre.value_counts(normalize=True)).head() # chaining commands

Out[10]: Drama        0.283963
         Comedy       0.159346
```

```
Action       0.138917
Crime        0.126660
Biography    0.078652
Name: genre, dtype: float64
```

```
In [11]: movies.genre.unique() # unique values in Series
```

```
Out[11]: array(['Crime', 'Action', 'Drama', 'Western', 'Adventure', 'Biography',
                'Comedy', 'Animation', 'Mystery', 'Horror', 'Film-Noir', 'Sci-Fi',
                'History', 'Thriller', 'Family', 'Fantasy'], dtype=object)
```

```
In [12]: movies.genre.nunique() # number of unique values in Series
```

```
Out[12]: 16
```

```
In [13]: pd.crosstab(movies.genre, movies.content_rating) # count movies belonging
```

Out[13]:

genre	APPROVED	G	GP	NC-17	NOT RATED	PASSED	PG	PG-13	R
Action	3	1	1	0	4	1	11	44	67
Adventure	3	2	0	0	5	1	21	23	17
Animation	3	20	0	0	3	0	25	5	5
Biography	1	2	1	0	1	0	6	29	36
Comedy	9	2	1	1	16	3	23	23	73
Crime	6	0	0	1	7	1	6	4	87
Drama	12	3	0	4	24	1	25	55	143
Family	0	1	0	0	0	0	1	0	0
Fantasy	0	0	0	0	0	0	0	0	1
Film-Noir	1	0	0	0	1	0	0	0	0
History	0	0	0	0	0	0	0	0	0
Horror	2	0	0	1	1	0	1	2	16
Mystery	4	1	0	0	1	0	1	2	6
Sci-Fi	1	0	0	0	0	0	0	1	3
Thriller	1	0	0	0	0	0	1	0	3
Western	1	0	0	0	2	0	2	1	3

genre	TV-MA	UNRATED	X
Action	0	3	0
Adventure	0	2	0
Animation	0	1	0
Biography	0	0	0
Comedy	0	4	1
Crime	0	11	1
Drama	1	9	1
Family	0	0	0
Fantasy	0	0	0
Film-Noir	0	1	0
History	0	1	0

```
Horror              0        5  1
Mystery             0        1  0
Sci-Fi              0        0  0
Thriller            0        0  0
Western             0        0  0
```

```
In [14]: movies.duration.describe()
```

```
Out[14]: count    979.000000
         mean     120.979571
         std       26.218010
         min       64.000000
         25%      102.000000
         50%      117.000000
         75%      134.000000
         max      242.000000
         Name: duration, dtype: float64
```

```
In [15]: movies.duration.mean() # mean of duration
```

```
Out[15]: 120.97957099080695
```

```
In [16]: movies.duration.value_counts() # eg 23 movies with 112 duration (in mins)
```

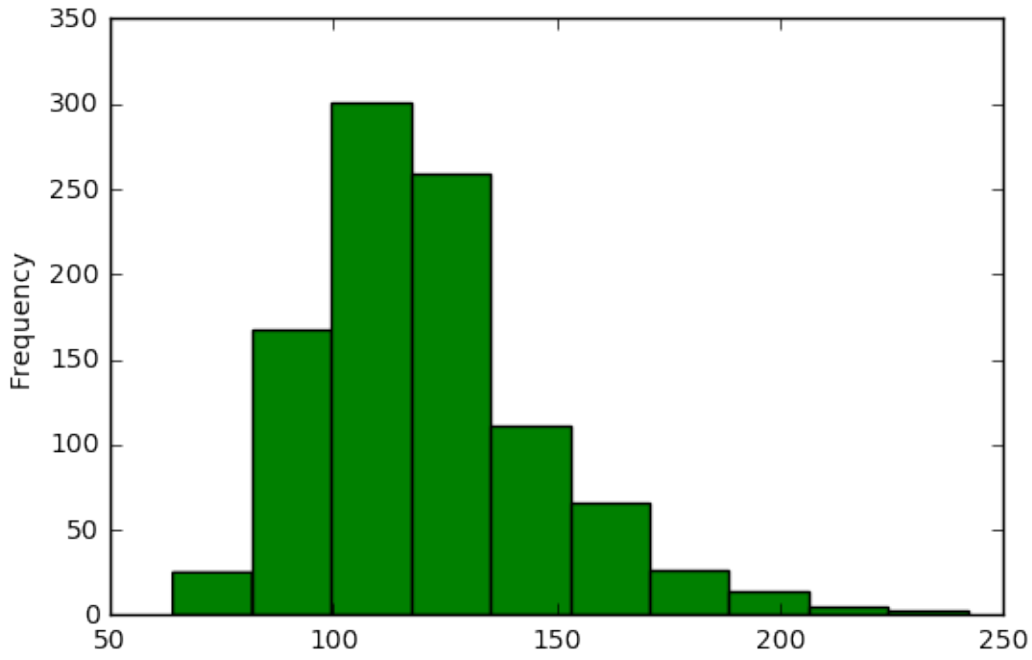
```
Out[16]: 112    23
         113    22
         102    20
         101    20
         129    19
         120    18
         105    18
         126    18
         98     18
         130    18
         100    17
         121    17
         116    17
         124    16
         122    16
         118    16
         115    16
         96     16
         104    16
         110    16
         107    16
         109    16
         119    15
         114    15
         99     15
```

```
108    15
94     14
117    14
106    14
93     14
       ..
70      1
69      1
67      1
66      1
242     1
238     1
195     1
229     1
224     1
220     1
216     1
212     1
207     1
205     1
202     1
201     1
200     1
194     1
159     1
193     1
187     1
186     1
184     1
183     1
182     1
180     1
177     1
168     1
166     1
64      1
Name: duration, dtype: int64
```

```
In [17]: %matplotlib inline
```

```
In [19]: movies.duration.plot(kind='hist', color='green')
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x7f74adeae9b0>
```

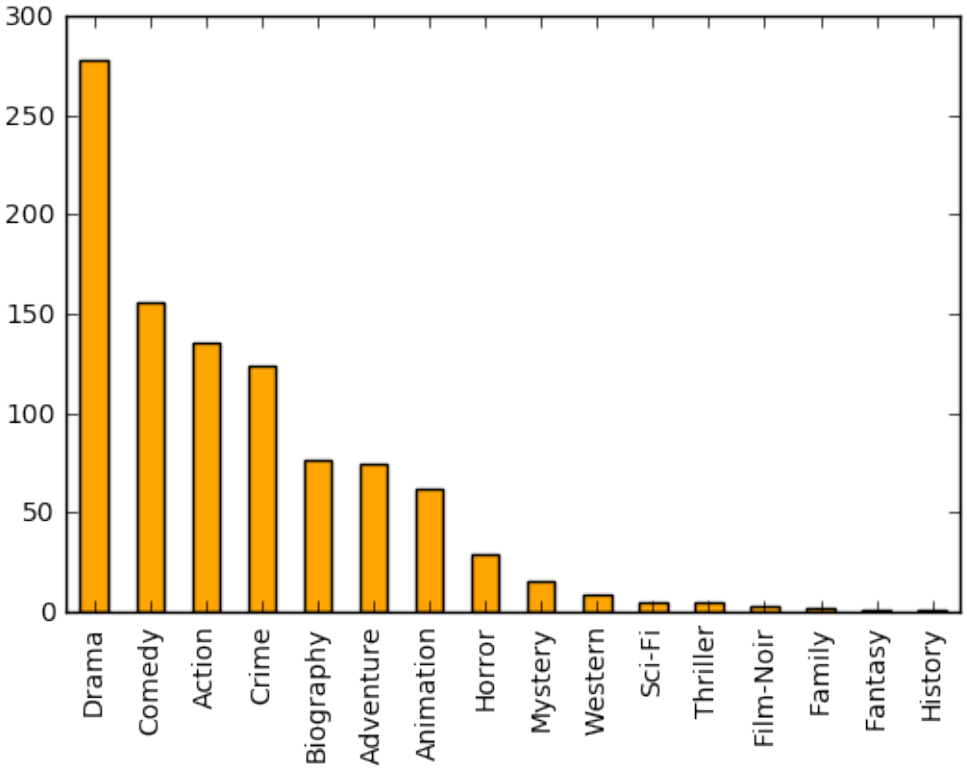



```
In [20]: movies.genre.value_counts().head()

Out[20]: Drama        278
         Comedy       156
         Action       136
         Crime        124
         Biography     77
         Name: genre, dtype: int64

In [22]: movies.genre.value_counts().plot(kind='bar', color='orange')

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7f74a86682b0>
```


300
250
200
150
100
50
0
Drama
Comedy
Action
Crime
Biography
Adventure
Animation
Horror
Mystery
Western
Sci-Fi
Thriller
Film-Noir
Family
Fantasy
History

In [ ]: