Introduction to Python 090718

Started on	Friday, 28 September 2018, 8:42 AM
State	Finished
Completed on	Sunday, 30 September 2018, 6:21 PM
Time taken	2 days 9 hours
Grade	**0.00** out of 1.00 (**0**%)

Information

Load the MovieData.csv dataset into a pandas DataFrame as described in this week's lesson, and use it to find the following values:

For each question submit code and answer, either in the textbox or by submitting a file (at the bottom). Later you will be asked to do further analysis, **so save your work.**

When you are satisfied with your answer for the questions, click **Finish attempt.** To see the model answer**, click Submit all and finish**.

Question 1

Not answered

Marked out of 1.00

a. What is the median profit of movies with budgets of over $50M?

b. How many movies were released by each film distributor? Output the results to a csv file.

JJenkins_Week 3_Exercise #1_and_#2.ipynb

Solution:

First, we load the dataset into a pandas DataFrame, just like in the lesson:

In [1]:

```
import pandas
import datetime as dt
```

In [2]:

```
def make_date(date_str):
    '''
    Turn a MM/DD/YY string into a datetime object
    '''
    m, d, y = date_str.split("/")
    m = int(m)
    d = int(d)
    y = int(y)
    if y > 13:
        y += 1900
    else:
        y += 2000
    return dt.datetime(y, m, d)
```

In [3]:

```
movies = pandas.read_csv("MovieData.csv", sep='\t',
                         na_values=["Unknown", "Unkno"], parse_dates=[0], date_parser=make_dat
e)
```

Make sure the data looks correct:

In [4]:

```
movies
```

In [5]:

```
# Replace missing values with zeros
movies.fillna(0, inplace=True)
```

In [6]:

```
# Fill in Worldwide Gross when it's missing
movies["Worldwide Gross"][movies["Worldwide Gross"]==0] = movies["US Gross"]
```

a)

Create a new Profit column, which will be the difference between Worldwide Gross and the budget.

In [7]:

```
movies["Profit"] = movies["Worldwide Gross"] -  movies["Budget"]
```

In [8]:

```
movies.head()
```

We only care about a subset of the data: movies with budgets greater than $50,000,000, so we create a new DataFrame with only the rows that fit our criteria:

In [9]:

```
bigger_budget = movies[movies.Budget > 50000000]
```

Finally, use the *.median()* method on the Profit column to find the median profit in the new DataFrame

In [10]:

```
bigger_budget.Profit.median()
```

b)

To answer this question, we can use *groupby(...)*, grouping by distributor. To count the number of rows associated with each distributor, we aggregate with the *len* function, to get the length of (number of entries in) each group. Then get a column -- any column will do, since the count should be the same for all of them.

In [11]:

```
distributors = movies.groupby("Distributor").aggregate(len)
distributor_count = distributors["Movie"]
```

Notice that we're using only the function's name, without the parentheses: *len*, not *len()*. Putting the parentheses in actually calls the function -- passing the name is telling *aggregate* what function to use on each group.

In [12]:

```
distributor_count
```

To output this Series to a CSV, just use its built-in *to_csv(...)* method.

In [13]:

```
distributor_count.to_csv("Distributor Counts.csv")
```

◀ Lesson 3 Lecture (Jupyter Notebook)

Jump to...

Exercise 2 ▶