Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

Retake the assignment in **7h 50m**

GRADE
100%

Analyze Box Office Data with Plotly and Python

LATEST SUBMISSION GRADE
100%

1. You are analyzing a pandas data frame **df** with multiple rows and columns. You notice that most entries in the **income** column are missing. How would you detect and return only the non-missing entries in **df['income']**?

1 / 1 point

- ○
```
df.loc[df['income'].isnull() == True, 'income']
```
- ○
```
df.loc[df['income'].isnull() == True]
```
- ◉
```
df.loc[df['income'].isnull() == False, 'income']
```
- ○
```
df.loc[df['income'].isnull() == False]
```

Correct
Correct!

2. While answering Question 1, you notice that the result contains a thousand rows. You only want to look at the first five entries. How would display only the fist five entries in **df**? Select all that apply.

1 / 1 point

- ☑
```
df[0:5]
```

Correct
Correct!

- ☐
```
df.tail()
```
- ☑
```
df.head()
```

Correct
Correct!

- ☑
```
df[:5]
```

Correct
Correct!

3. Plotly's **go.Scatter** function from **plotly.graph_objects** can be used both for plotting points (markers) or lines, depending on the value of the **mode** argument.

1 / 1 point

- ◉ True

False

Correct
Correct!

4. What change needs to be made to the following code chunk to convert Fig 1. to Fig 2? **1 / 1 point**

```
data = [go.Scatter(x=d1.index, y=d1.values, name='train', mode='lines'),
        go.Scatter(x=d2.index, y=d2.values, name='test', mode='lines')]
```

Fig 1

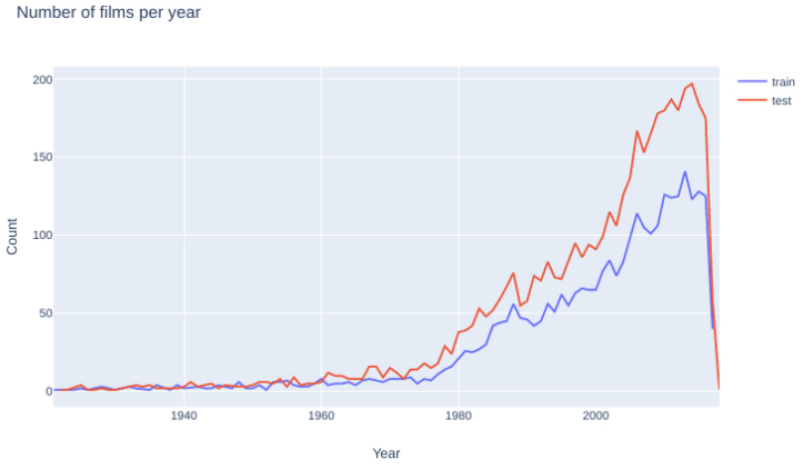
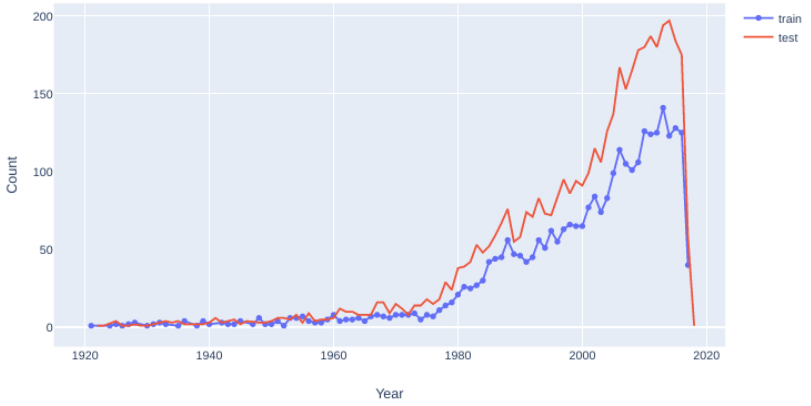


Fig 2




- ```
  data = [go.Scatter(x=d1.index, y=d1.values, name='train', mode='markers'),
          go.Scatter(x=d2.index, y=d2.values, name='test', mode='lines')]
  ```
- ```
  data = [go.Scatter(x=d1.index, y=d1.values, name='train', mode='lines'),
          go.Scatter(x=d2.index, y=d2.values, name='test', mode='lines+markers')]
  ```
- ```
  data = [go.Scatter(x=d1.index, y=d1.values, name='train', mode='lines+markers'),
          go.Scatter(x=d2.index, y=d2.values, name='test', mode='lines')]
  ```

Correct
Terrific!

5. You can use GUI controls to zoom, pan and scale plots generated with Plotly **1 / 1 point**
  ```

- ○ False
- ◉ True

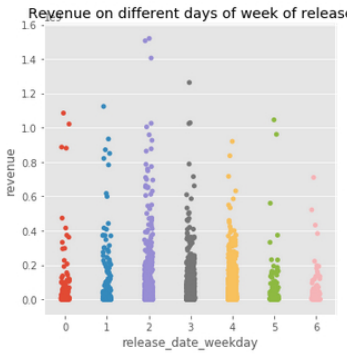
✓ **Correct**
Correct!

6. You are exploring a data frame **train** and want to visualize the **budget** column using a histogram. How can you use Seaborn to create the plot? **1 / 1 point**

- ○
```
1 sns.scatterplot(train['budget'])
```
- ○
```
1 sns.distplot(train['budget'], kde=True)
```
- ◉
```
1 sns.distplot(train['budget'], kde=False)
```

✓ **Correct**
Correct!

7. The following plot was generated using Seaborn's **sns.catplot()** function: **1 / 1 point**



What is a more accurate way of describing the plot?

- ○ Categorical histogram
- ◉ Categorical scatter plot

✓ **Correct**
Correct!