EDA

September 12, 2024

[270]:
```
# import the necessary library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[271]:
```
# loading the data set
df = pd.read_csv('Musician(1).csv')
df.head()
```

[271]:

	Rank	Peak	All Time Peak	Actual gross	Adjusted gross (in 2022 dollars)	Artist	Tour title	Year(s)	Shows	Average gross	Ref.
0	1	1	2	$780,000,000	$780,000,000	Taylor Swift	The Eras Tour †	2023–2024	56	$13,928,571	[1]
1	2	1	7[2]	$579,800,000	$579,800,000	Beyoncé	Renaissance World Tour	2023	56	$10,353,571	[3]
2	3	1[4]	2[5]	$411,000,000	$560,622,615	Madonna	Sticky & Sweet Tour ‡[4][a]	2008–2009	85	$4,835,294	[6]
3	4	2[7]	10[7]	$397,300,000	$454,751,555	Pink	Beautiful Trauma World Tour	2018–2019	156	$2,546,795	[7]
4	5	2[4]	NaN	$345,675,146	$402,844,849	Taylor Swift	Reputation Stadium Tour	2018	53	$6,522,173	[8]

[272]:
```
# checking the columns names in the data set
df.columns
```

[272]:
```
Index(['Rank', 'Peak', 'All Time Peak', 'Actual gross',
       'Adjusted gross (in 2022 dollars)', 'Artist', 'Tour title', 'Year(s)',
       'Shows', 'Average gross', 'Ref.'],
      dtype='object')
```

[ ]:

[273]:
```
# removing unwanted(multiple column) columns in the data set. Note that the␣
 ↪inplace=True removes the column permanently from the data set
df.drop(columns = ['Rank', 'Peak', 'All Time Peak', 'Ref.'], inplace=True)
```

[274]:
```
df.head()
```

[274]:

	Actual gross	Adjusted gross (in 2022 dollars)	Artist	Tour title	Year(s)	Shows	Average gross
0	$780,000,000	$780,000,000	Taylor Swift	The Eras Tour †	2023-2024	56	$13,928,571
1	$579,800,000	$579,800,000	Beyoncé	Renaissance World Tour	2023	56	$10,353,571
2	$411,000,000	$560,622,615	Madonna	Sticky & Sweet Tour ‡[4][a]	2008-2009	85	$4,835,294
3	$397,300,000	$454,751,555	Pink	Beautiful Trauma World Tour	2018-2019	156	$2,546,795
4	$345,675,146	$402,844,849	Taylor Swift	Reputation Stadium Tour	2018	53	$6,522,173

[275]:
```
df.columns
```

[275]:
```
Index(['Actual gross', 'Adjusted gross (in 2022 dollars)', 'Artist',
       'Tour title', 'Year(s)', 'Shows', 'Average gross'],
      dtype='object')
```

[276]:
```
# renaming the column names
col_rename = {'Actual gross': 'gross', 'Adjusted gross (in 2022 dollars)':␣
 ↪'adj_gross', 'Artist': 'artist', 'Tour title': 'tour', 'Year(s)': 'year',␣
 ↪'Shows':'show', 'Average gross':'avg_gross'}
```

[277]:
```
df.rename(columns=col_rename, inplace=True)
```

[278]:
```
df.head()
```

[278]:

	Actual gross	Adjusted gross (in 2022 dollars)	artist	tour	year	show	avg_gross
0	$780,000,000	$780,000,000	Taylor Swift	The Eras Tour †	2023-2024	56	$13,928,571
1	$579,800,000	$579,800,000	Beyoncé	Renaissance World Tour	2023	56	$10,353,571
2	$411,000,000	$560,622,615	Madonna	Sticky & Sweet Tour ‡[4][a]	2008-2009	85	$4,835,294
3	$397,300,000	$454,751,555	Pink				
4	$345,675,146	$402,844,849	Taylor Swift				

```
3  Beautiful Trauma World Tour  2018-2019   156   $2,546,795
4       Reputation Stadium Tour       2018    53   $6,522,173
```

[279]:
```
# some columns name didn't rename because of the columns names are not exact␣
 ↪match hence we need to check the exact column name
df.columns.tolist()
```

[279]:
```
['Actual\xa0gross',
 'Adjusted\xa0gross (in 2022 dollars)',
 'artist',
 'tour',
 'year',
 'show',
 'avg_gross']
```

[280]:
```
# removing the unwanted encoding that made the column name not to change
df.columns= df.columns.str.encode('ascii', 'ignore').str.decode('ascii')
```

[281]:
```
df.columns
```

[281]:
```
Index(['Actualgross', 'Adjustedgross (in 2022 dollars)', 'artist', 'tour',
       'year', 'show', 'avg_gross'],
      dtype='object')
```

[282]:
```
df.rename(columns = {'Actual gross': 'gross', 'Adjusted gross (in 2022␣
 ↪dollars)': 'adj_gross'})
```

[282]:
```
         Actualgross Adjustedgross (in 2022 dollars)        artist  \
0       $780,000,000                    $780,000,000  Taylor Swift
1       $579,800,000                    $579,800,000       Beyoncé
2       $411,000,000                    $560,622,615       Madonna
3       $397,300,000                    $454,751,555          Pink
4       $345,675,146                    $402,844,849  Taylor Swift
5       $305,158,363                    $388,978,496       Madonna
6       $280,000,000                    $381,932,682   Celine Dion
7       $257,600,000                    $257,600,000          Pink
8       $256,084,556                    $312,258,401       Beyoncé
9       $250,400,000                    $309,141,878  Taylor Swift
10  $229,100,000[b]                     $283,202,896       Beyoncé
11      $227,400,000                    $295,301,479     Lady Gaga
12      $204,000,000                    $251,856,802    Katy Perry
13      $200,000,000                    $299,676,265          Cher
14      $194,000,000                    $281,617,035       Madonna
15      $184,000,000                    $227,452,347          Pink
16      $170,000,000                    $213,568,571     Lady Gaga
17      $169,800,000                    $207,046,755       Madonna
18  $167,700,000[e]                     $204,486,106         Adele
```

```
19     $150,000,000                    $185,423,109  Taylor Swift

                                    tour       year  show    avg_gross
0                          The Eras Tour †  2023-2024    56  $13,928,571
1                    Renaissance World Tour       2023    56  $10,353,571
2                Sticky & Sweet Tour ‡[4][a]  2008-2009    85   $4,835,294
3                Beautiful Trauma World Tour  2018-2019   156   $2,546,795
4                    Reputation Stadium Tour       2018    53   $6,522,173
5                              The MDNA Tour       2012    88   $3,467,709
6                  Taking Chances World Tour  2008-2009   131   $2,137,405
7                          Summer Carnival †  2023-2024    41   $6,282,927
8                   The Formation World Tour       2016    49   $5,226,215
9                        The 1989 World Tour       2015    85   $2,945,882
10           The Mrs. Carter Show World Tour  2013-2014   132   $1,735,606
11                  The Monster Ball Tour *  2009-2011   203   $1,118,227
12                     Prismatic World Tour  2014-2015   151   $1,350,993
13  Living Proof: The Farewell Tour ‡[21][a]  2002-2005   325     $615,385
14                           Confessions Tour       2006    60   $3,233,333
15                 The Truth About Love Tour  2013-2014   142   $1,295,775
16                        Born This Way Ball  2012-2013    98   $1,734,694
17                          Rebel Heart Tour  2015-2016    82   $2,070,732
18                           Adele Live 2016  2016-2017   121   $1,385,950
19                              The Red Tour  2013-2014    86   $1,744,186
```

[283]:
```
df.head()
```

[283]:
```
    Actualgross Adjustedgross (in 2022 dollars)        artist  \
0  $780,000,000                    $780,000,000  Taylor Swift
1  $579,800,000                    $579,800,000       Beyoncé
2  $411,000,000                    $560,622,615       Madonna
3  $397,300,000                    $454,751,555          Pink
4  $345,675,146                    $402,844,849  Taylor Swift

                         tour       year  show    avg_gross
0             The Eras Tour †  2023-2024    56  $13,928,571
1       Renaissance World Tour       2023    56  $10,353,571
2  Sticky & Sweet Tour ‡[4][a]  2008-2009    85   $4,835,294
3  Beautiful Trauma World Tour  2018-2019   156   $2,546,795
4      Reputation Stadium Tour       2018    53   $6,522,173
```

[284]:
```
df.columns.tolist()
```

[284]:
```
['Actualgross',
 'Adjustedgross (in 2022 dollars)',
 'artist',
 'tour',
 'year',
```

```
 'show',
 'avg_gross']
```

[285]:
```
df.rename(columns={'Actualgross': 'gross', 'Adjustedgross (in 2022 dollars)':
 ↪'adj_gross'}, inplace=True)
```

[286]:
```
df.head()
```

[286]:
```
          gross     adj_gross        artist                          tour  \
0  $780,000,000  $780,000,000  Taylor Swift             The Eras Tour †
1  $579,800,000  $579,800,000       Beyoncé       Renaissance World Tour
2  $411,000,000  $560,622,615       Madonna  Sticky & Sweet Tour ‡[4][a]
3  $397,300,000  $454,751,555          Pink  Beautiful Trauma World Tour
4  $345,675,146  $402,844,849  Taylor Swift      Reputation Stadium Tour

        year  show    avg_gross
0  2023-2024    56  $13,928,571
1       2023    56  $10,353,571
2  2008-2009    85   $4,835,294
3  2018-2019   156   $2,546,795
4       2018    53   $6,522,173
```

[287]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   gross      20 non-null     object
 1   adj_gross  20 non-null     object
 2   artist     20 non-null     object
 3   tour       20 non-null     object
 4   year       20 non-null     object
 5   show       20 non-null     int64
 6   avg_gross  20 non-null     object
dtypes: int64(1), object(6)
memory usage: 1.2+ KB
```

[288]:
```
 # get info for only one column, sample gives a random sample , note gross is␣
 ↪supposed to be an integar but it is an object data type
df.gross.sample(5)
```

[288]:
```
13    $200,000,000
19    $150,000,000
14    $194,000,000
8     $256,084,556
3     $397,300,000
```

```
Name: gross, dtype: object
```

[289]:
```
# $, [] made the datatype to be a string instead of int hence we need to remove␣
  ↪this
df.gross.str.replace('$','').str.replace(',', '')
```

[289]:
```
0         780000000
1         579800000
2         411000000
3         397300000
4         345675146
5         305158363
6         280000000
7         257600000
8         256084556
9         250400000
10    229100000[b]
11        227400000
12        204000000
13        200000000
14        194000000
15        184000000
16        170000000
17        169800000
18    167700000[e]
19        150000000
Name: gross, dtype: object
```

[290]:
```
# using regular expression we can remove the string from the data set
df .gross.str.replace(r'[^\d.]', '', regex=True)
```

[290]:
```
0     780000000
1     579800000
2     411000000
3     397300000
4     345675146
5     305158363
6     280000000
7     257600000
8     256084556
9     250400000
10    229100000
11    227400000
12    204000000
13    200000000
14    194000000
15    184000000
```

```
16    170000000
17    169800000
18    167700000
19    150000000
Name: gross, dtype: object
```

[291]:
```
#but since we have more than one column to remove the str datatpe we will be␣
 ↪using a loop instead of running the code for seperate columns
col_to_cleaned = ['gross', 'adj_gross', 'avg_gross']
```

[292]:
```
for col in col_to_cleaned:
    df[col] = df[col].str.replace(r'[^\d.]', '', regex=True)
df.head()
```

[292]:

	gross	adj_gross	artist	tour	year	show	avg_gross
0	780000000	780000000	Taylor Swift	The Eras Tour †	2023-2024	56	13928571
1	579800000	579800000	Beyoncé	Renaissance World Tour	2023	56	10353571
2	411000000	560622615	Madonna	Sticky & Sweet Tour ‡[4][a]	2008-2009	85	4835294
3	397300000	454751555	Pink	Beautiful Trauma World Tour	2018-2019	156	2546795
4	345675146	402844849	Taylor Swift	Reputation Stadium Tour	2018	53	6522173

[293]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   gross      20 non-null     object
 1   adj_gross  20 non-null     object
 2   artist     20 non-null     object
 3   tour       20 non-null     object
 4   year       20 non-null     object
 5   show       20 non-null     int64
 6   avg_gross  20 non-null     object
dtypes: int64(1), object(6)
memory usage: 1.2+ KB
```

[294]:
```
# we have change the content to numeric but the  datatype is still showing␣
 ↪object datatype, we need to change this to a numeric data type
#to_numeric convert an object datatype to numeric
```

```
# df.gross.astype (int)
pd.to_numeric(df.gross)
```

[294]:
```
0     780000000
1     579800000
2     411000000
3     397300000
4     345675146
5     305158363
6     280000000
7     257600000
8     256084556
9     250400000
10    229100000
11    227400000
12    204000000
13    200000000
14    194000000
15    184000000
16    170000000
17    169800000
18    167700000
19    150000000
Name: gross, dtype: int64
```

[295]:
```
for col in col_to_cleaned:
    df[col] = pd.to_numeric(df[col])
```

[296]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   gross      20 non-null     int64
 1   adj_gross  20 non-null     int64
 2   artist     20 non-null     object
 3   tour       20 non-null     object
 4   year       20 non-null     object
 5   show       20 non-null     int64
 6   avg_gross  20 non-null     int64
dtypes: int64(4), object(3)
memory usage: 1.2+ KB
```

[297]:
```
df.sample(5)
```

[297]:
```
        gross  adj_gross      artist                             tour  \
10  229100000  283202896     Beyoncé  The Mrs. Carter Show World Tour
7   257600000  257600000        Pink                Summer Carnival †
12  204000000  251856802  Katy Perry             Prismatic World Tour
3   397300000  454751555        Pink      Beautiful Trauma World Tour
1   579800000  579800000     Beyoncé           Renaissance World Tour

          year  show  avg_gross
10  2013-2014   132    1735606
7   2023-2024    41    6282927
12  2014-2015   151    1350993
3   2018-2019   156    2546795
1        2023    56   10353571
```

[298]:
```
#note that the year column is not properly stated a range an a single year
df.year
```

[298]:
```
0     2023-2024
1          2023
2     2008-2009
3     2018-2019
4          2018
5          2012
6     2008-2009
7     2023-2024
8          2016
9          2015
10    2013-2014
11    2009-2011
12    2014-2015
13    2002-2005
14         2006
15    2013-2014
16    2012-2013
17    2015-2016
18    2016-2017
19    2013-2014
Name: year, dtype: object
```

[299]:
```
df.year[2][-4:]
```

[299]:
```
'2009'
```

[300]:
```
df.head()
```

[300]:
```
        gross  adj_gross        artist                          tour       year  \
0  780000000  780000000  Taylor Swift             The Eras Tour †  2023-2024
1  579800000  579800000       Beyoncé       Renaissance World Tour       2023
```

```
2  411000000  560622615       Madonna  Sticky & Sweet Tour ‡[4][a]  2008-2009
3  397300000  454751555          Pink  Beautiful Trauma World Tour  2018-2019
4  345675146  402844849  Taylor Swift      Reputation Stadium Tour       2018

   show  avg_gross
0    56   13928571
1    56   10353571
2    85    4835294
3   156    2546795
4    53    6522173
```

[301]:
```
# hence we need to use slicing to get the
def extract_year(value):

    return value[-4:] if len(value)> 4 else value
```

[302]:
```
df['year']=df.year.apply(extract_year)
```

[303]:
```
df.head()
```

[303]:
```
        gross  adj_gross        artist                          tour  year  \
0  780000000  780000000  Taylor Swift             The Eras Tour †  2024
1  579800000  579800000       Beyoncé       Renaissance World Tour  2023
2  411000000  560622615       Madonna  Sticky & Sweet Tour ‡[4][a]  2009
3  397300000  454751555          Pink  Beautiful Trauma World Tour  2019
4  345675146  402844849  Taylor Swift      Reputation Stadium Tour  2018

   show  avg_gross
0    56   13928571
1    56   10353571
2    85    4835294
3   156    2546795
4    53    6522173
```

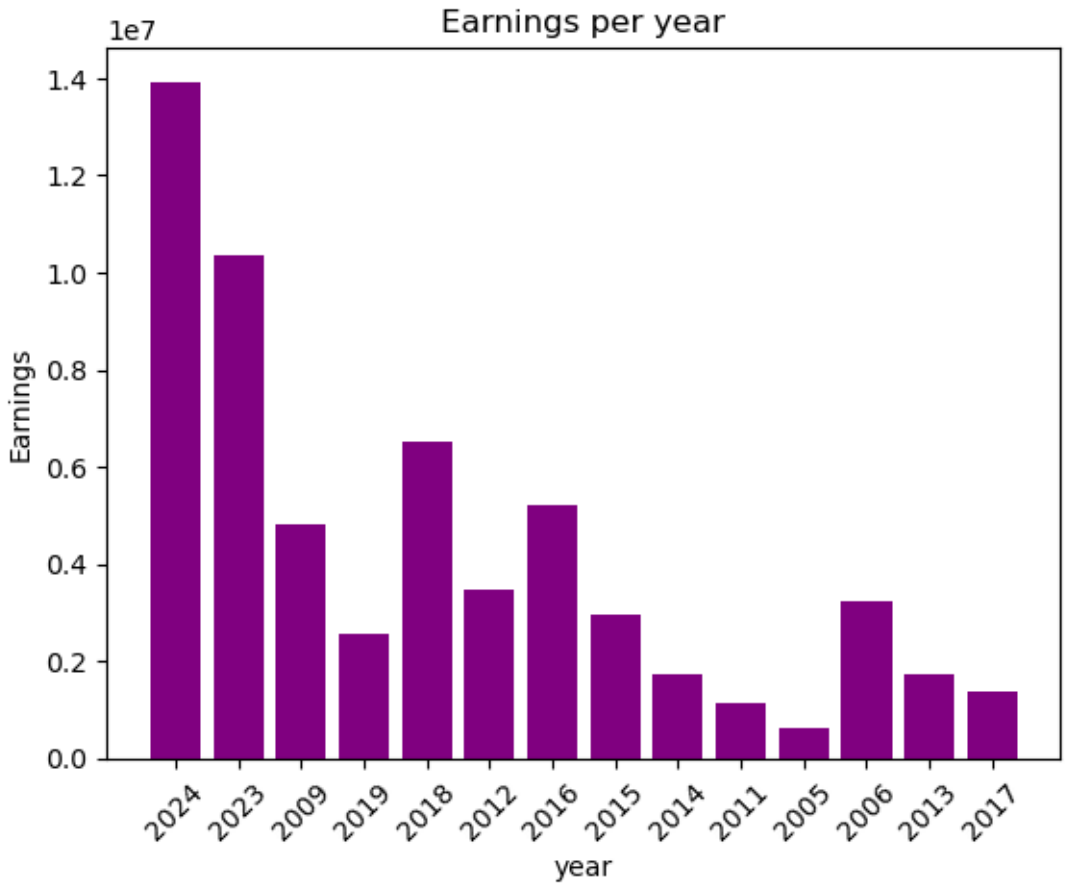
[310]:
```
#sending the cleaned data to excel
#df.to_excel('cleaned.xlsx')
```

[306]:
```
# summary report
df.describe().round(1)
```

[306]:
```
              gross    adj_gross   show   avg_gross
count          20.0         20.0   20.0        20.0
mean   287950903.2  343878092.0  110.0   3726571.2
std    156328421.0  151462683.7   66.5   3393339.6
min    150000000.0  185423109.0   41.0    615385.0
25%    191500000.0  245755688.2   59.0   1647508.0
50%    239750000.0  297488872.0   87.0   2342100.0
```

```
75%     315287558.8  392445084.2  134.5   4933024.2
max     780000000.0  780000000.0  325.0  13928571.0
```

[307]:
```
# what is the earning per year using a graph
x= df.avg_gross
y= df.year
plt.bar(y,x, color='purple')
plt.xticks(rotation=45)
plt.title('Earnings per year')
plt.ylabel('Earnings')
plt.xlabel('year')
plt.show()
```



[ ]:

[308]:
```
# getting the avg earning of each artist
df.groupby('artist')[['avg_gross']].sum().reset_index().
  ↪sort_values(by='avg_gross', ascending=False)
```

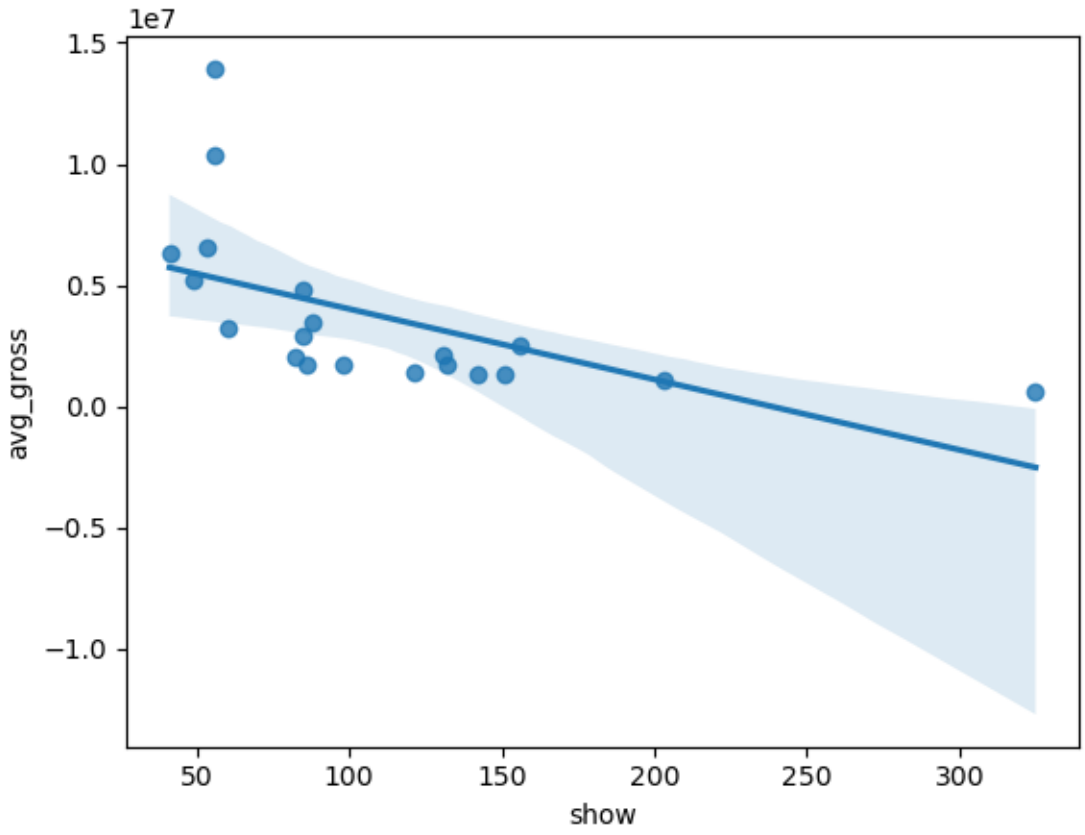
[308]:

	artist	avg_gross
8	Taylor Swift	25140812
1	Beyoncé	17315392
6	Madonna	13607068
7	Pink	10125497
5	Lady Gaga	2852921
2	Celine Dion	2137405
0	Adele	1385950
4	Katy Perry	1350993
3	Cher	615385

[ ]:
```
# to measure relationship use correlation
#round(df.corr(),2)
```

[309]:
```
# impact of shows on earning
sns.regplot(x='show', y='avg_gross', data=df)
```

[309]: <Axes: xlabel='show', ylabel='avg_gross'>



[ ]: