Movies: Relationship between revenue and budget

A story told in data

10m $

110m $

250m $

https://m.media-amazon.com/images/M/MV5BNTdmMDNmYmItOWFmNC00YzdkLWIyZWMtMGRlMTQyZDZmNDU0XkEyXkFqcGdeQXVyMTkxNjUyNQ@@._V1_.jpg, https://www.imdb.com/title/tt12412888/, https://www.amazon.com/Avatar-Movie-Poster-14-21/dp/B09WVSCYPX

Movie budgets:
A grant for success?

Data Cleaning

Python packages

```
import pandas as pd

import numpy as np

import seaborn as sns

from sklearn.linear_model import LinearRegression

from sklearn.metrics import r2_score

import statsmodels.api as sm

from scipy.stats import pearsonr

import scipy
```

Reading the csv file

```
df = pd.read_csv("tmdb_movies_data.csv")

df
```

Columns

df.describe

- 10866 rows and 21 columns

df.columns

- Shows the 21 columns that are present

df.drop([…],axis=1,inpalce=True)

- Drops the unecessary columsna
  - Leaves us with budget, revenue, release_year, budget_adj, and revneue_adj

df.dtype

- Tells us that each column is either a float64 or int64

df

- After removing the necessary columns, we check the data frame
- Budget and revenue columns contains a lot of 0s

	budget	revenue	release_year	budget_adj	revenue_adj
0	0	29355203	2015	0.0	2.700677e+07
1	0	22354572	2015	0.0	2.056620e+07
2	0	45895	2015	0.0	4.222338e+04
3	0	0	2015	0.0	0.000000e+00
4	0	0	2015	0.0	0.000000e+00
...	...	...	...	...	...
10861	270000000	391081192	2006	292050672.7	4.230205e+08
10862	280000000	1405035767	2015	257599886.7	1.292632e+09
10863	300000000	961000000	2007	315500574.8	1.010654e+09
10864	380000000	1021683000	2011	368371256.2	9.904175e+08
10865	425000000	11087569	2010	425000000.0	1.108757e+07

10866 rows × 5 columns

Removing data points

```
df = df.replace([np.inf, -np.inf, 0], np.nan)
```

- Replace infinities and zeros with nan values

```
df = df.dropna(axis = 0)
```

- Drop all nan values

```
df
```

- Shows us the data frame to ensure we haven't messed up

str Check

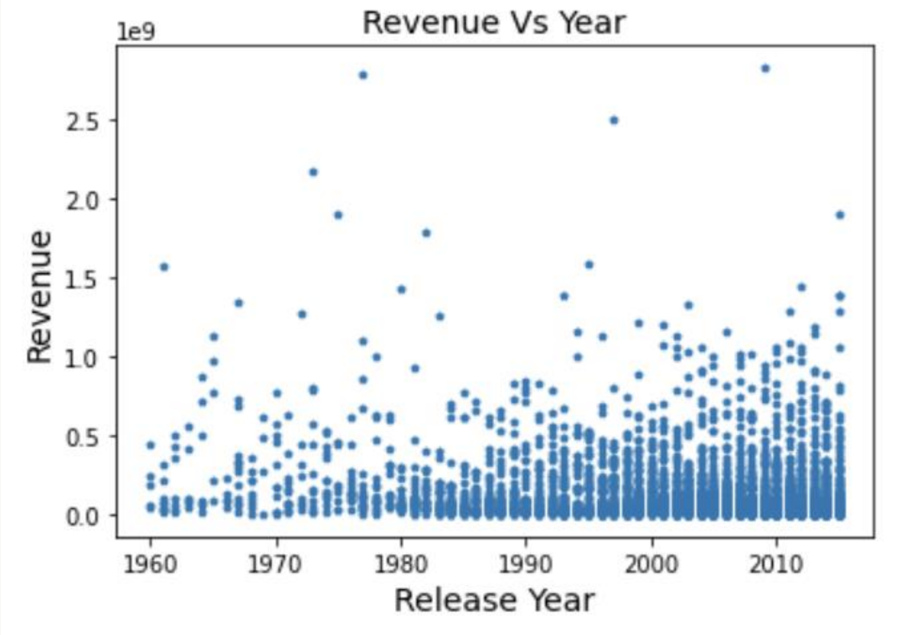
- Function to go through data points and remove rows which contain str values
  + This is done by converting them to nan values and deleting all nan values
- Once everything is done, the word success is printed
- *This is not necessary but adds another layer of safety

```
def str_check(column):
  for i in range(1, len(df.budget)):
    if df.loc(i,) == str:

      df.replace(df.id(i), np.nan)
      df.dropna(how='all', axis=0)
      df

    else:

      return print("success")

def column_list():

  dict_column = {
      1:"id", 2:"budget", 3:"revenue", 4:"release_year", 5:"budget_adj", 6:"revenue_adj"
      }

  for i in range (1,6):
    str_check(dict_column[i])
    return



column_list()

success
```

Data Analysis / Visualization

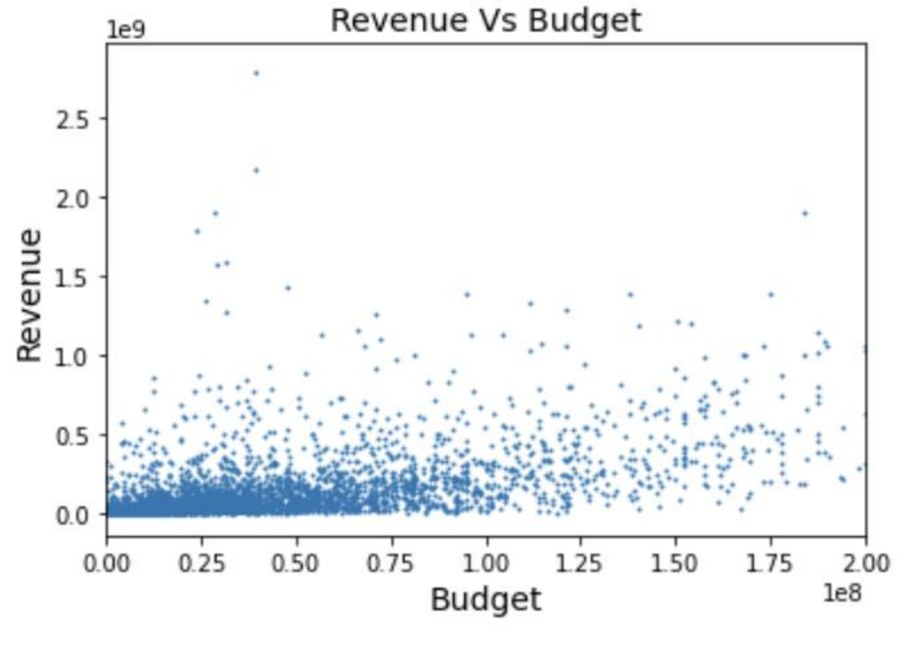
Scatterplot (REV. and Year)

```
plt.scatter(df['release_year'],
df['revenue_adj'], marker='p', s=10)

plt.title('Revenue Vs Year',
fontsize=14)

plt.xlabel('Release Year',
fontsize=14)

plt.ylabel('Revenue', fontsize=14)

plt.show()
```



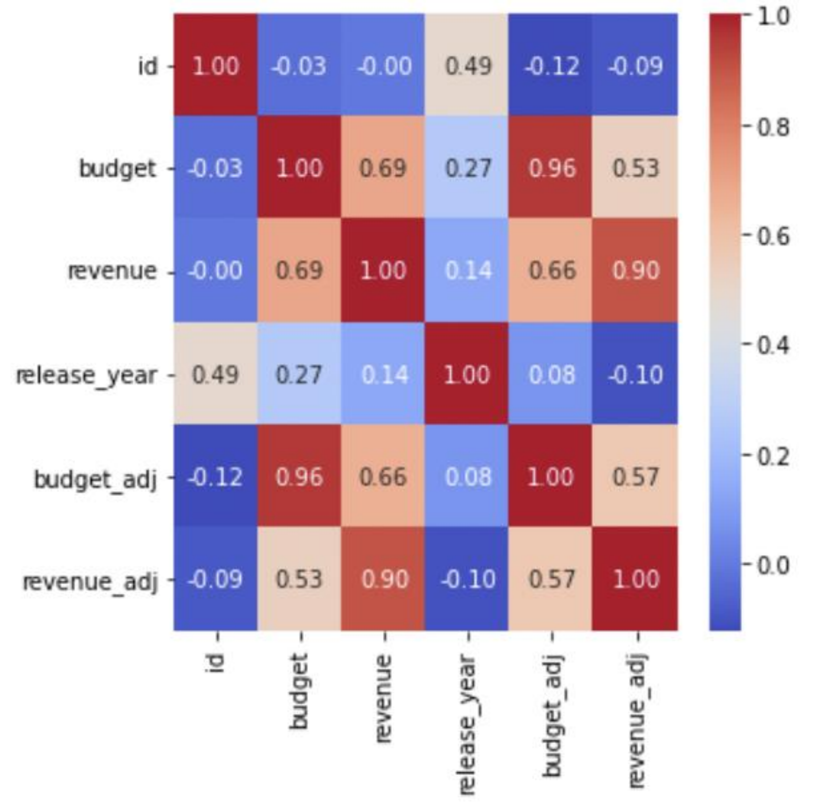
Scatterplot (REV. and Budget)

```
plt.scatter(df['budget_adj'],
df['revenue_adj'], marker='o', s=1)

plt.title('Revenue Vs Budget',
fontsize=14)

plt.xlabel('Budget', fontsize=14)

plt.ylabel('Revenue', fontsize=14)

plt.xlim(0, 2 * (10**8))


plt.show()
```



Correlations

```
corr = df.corr()

corr

figure = plt.figure(figsize=(5,5))

sns.heatmap(corr, cmap="coolwarm",
annot=True, fmt="0.2f")
```



Statistical table

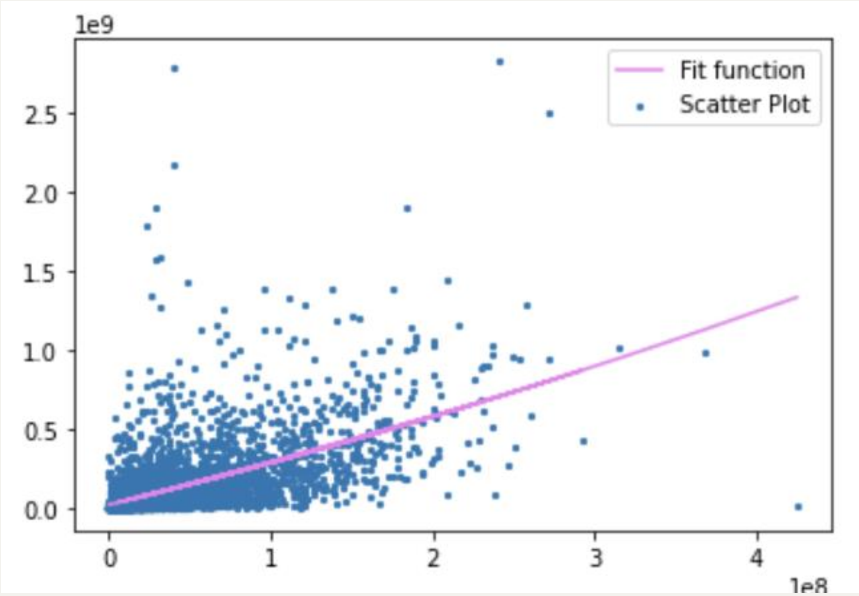
```
x = df["budget_adj"]

y = df["revenue_adj"]


x2 = sm.add_constant(x)

est = sm.OLS(y,x2)

est2 = est.fit()

print(est2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            revenue_adj   R-squared:                       0.325
Model:                            OLS   Adj. R-squared:                  0.325
Method:                 Least Squares   F-statistic:                     1859.
Date:                Tue, 29 Nov 2022   Prob (F-statistic):               0.00
Time:                        16:03:08   Log-Likelihood:                -78693.
No. Observations:                3855   AIC:                         1.574e+05
Df Residuals:                    3853   BIC:                         1.574e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.532e+07   4.02e+06      3.812      0.000    7.44e+06    2.32e+07
budget_adj     2.7514      0.064     43.114      0.000       2.626       2.877
==============================================================================
Omnibus:                     3688.968   Durbin-Watson:                   1.830
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           269025.645
Skew:                           4.433   Prob(JB):                         0.00
Kurtosis:                      42.953   Cond. No.                     8.85e+07
==============================================================================
```

$$Revenue\_adj = 1.532 * 10^7 + 2.7514 * Budget\_adj$$

Plotting the Ordinary Least Squared Regression

```
params = np.polyfit(xs, ys, deg=2)

line_fit = np.polyval(params,xs)

plt.plot(xs, line_fit, label="Fit
function", color= "violet")

plt.scatter(df['budget_adj'],
df['revenue_adj'], marker='o', s=5,
label="Scatter Plot")

plt.legend()

plt.show()
```



Conclusion

- A **third** of all change can be attributed to Budget!
  + **1$** of Budget yields on average **2.75$** more Revenue

- Limitations: We could **not** distinguish budget spent on marketing / operations.
  + **Production** budget

- Further Research Possibilities: **Genres**
  + **Mainstream** movies much more likely to gain an audience by spending 100 mil. $ on exploding horses and making TikToks about it.

Conclusion

OLS Regression Results

Dep. Variable:	revenue_adj	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.325
Method:	Least Squares	F-statistic:	1859.
Date:	Tue, 29 Nov 2022	Prob (F-statistic):	0.00
Time:	16:03:08	Log-Likelihood:	-78693.
No. Observations:	3855	AIC:	1.574e+05
Df Residuals:	3853	BIC:	1.574e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.532e+07	4.02e+06	3.812	0.000	7.44e+06	2.32e+07
budget_adj	2.7514	0.064	43.114	0.000	2.626	2.877

Omnibus:	3688.968	Durbin-Watson:	1.830
Prob(Omnibus):	0.000	Jarque-Bera (JB):	269025.645
Skew:	4.433	Prob(JB):	0.00
Kurtosis:	42.953	Cond. No.	8.85e+07