UDACITY

Logout

Return to "Data Analyst Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Investigate a Dataset

REVIEW | HISTORY

Meets Specifications

Code Functionality

All code is functional and produces no errors when run. The code given is sufficient to reproduce the results described.

The project uses NumPy arrays and Pandas Series and DataFrames where appropriate rather than Python lists and dictionaries. Where possible, vectorized operations and built-in functions are used instead of loops.

The analysis makes use of the NumPy and Pandas libraries, vector operators are employed instead of loops and lists.

It is awesome that you make use the function `.info()` or `.describe()` to examine the structure of the entire data, identify missing values and the summary statistics for the numerical features.

Pandas allow you to present the statistics in a simple table, that can be useful when included next to the corresponding figure.

```
df_new1.groupby(['genres' ])[['id']].count()
```

	id
genres	
Comedy	292
Comedy|Drama	116
Documentary	35
Drama	317
Drama|Romance	147

```
df_new1.groupby(['genres' ])[['budget']].sum()
```

	budget
genres	
Comedy	6771216919
Comedy|Drama	2008077010
Documentary	90747355
Drama	5787292732
Drama|Romance	2703887169

The code makes use of functions to avoid repetitive code. The code contains good comments and variable names, making it easy to read.

Quality of Analysis

The project clearly states one or more questions, then addresses those questions in the rest of the analysis.

The report states clear and relevant questions that are being addressed by the following analysis.

Data Wrangling Phase

The project documents any changes that were made to clean the data, such as merging multiple files, handling missing values, etc.

Well Done for identifying the missing values in the dataset and documenting the changes made in the dataset. This is important because it makes it possible for the readers to repeat your analysis if needed.

Exploration Phase

The project investigates the stated question(s) from multiple angles. At least three variables are investigated using both single-variable (1d) and multiple-variable (2d) explorations.

The analysis makes use of both single and multiple variable explorations to investigate different features and the relations between these features in the dataset.

The project's visualizations are varied and show multiple comparisons and trends. Relevant statistics are computed throughout the analysis when an inference is made about the data.

At least two kinds of plots should be created as part of the explorations.

The report makes use of different chart type to explore and depict the insights and the results of the analysis. I strongly encourage you to include the relevant statistics next to each figure. Below I show a few examples of different chart types and the relevant descriptive statistics.

For example, A box plot allows you to visualize the comparison of the distribution between different categories. Please note how I include the relevant statistics next to each figure.

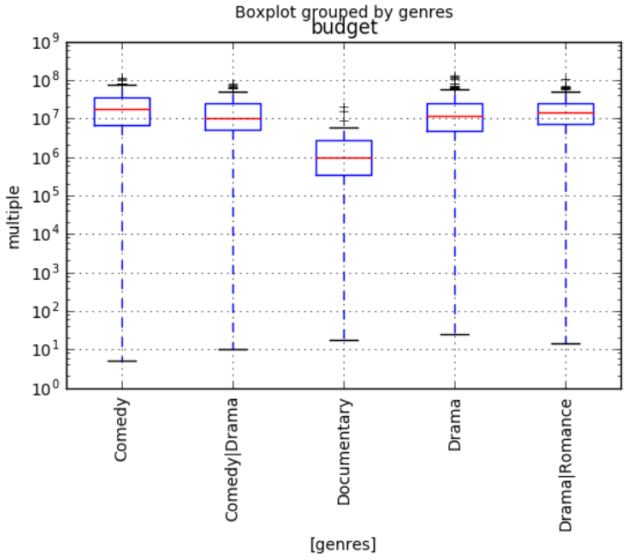
Rate this review

statistics next to each figure;

```
# Sort the data acording to the sample numbers , sellecting the top 5
df_1=df.groupby(['genres' ])[['id']].count().sort_values(by=['id'], ascending=False)[0:5]
df_new = df[df['genres'].isin(df_1.index.values.tolist())]
# remove budget = 0
df_new1= df_new[df_new['budget']>0]

df_new1.boxplot(column=['budget'],by = ['genres'], rot=90).set_yscale('log')
plt.ylabel("multiple")
pd.DataFrame(df_new1.groupby( ['genres'])['budget'].describe().loc[:,['mean','std']])
```

		budget
genres		
Comedy	**mean**	2.318910e+07
	std	2.145831e+07
Comedy|Drama	**mean**	1.731101e+07
	std	1.798798e+07
Documentary	**mean**	2.592782e+06
	std	4.251575e+06
Drama	**mean**	1.825644e+07
	std	1.986337e+07
Drama|Romance	**mean**	1.839379e+07
	std	1.740442e+07



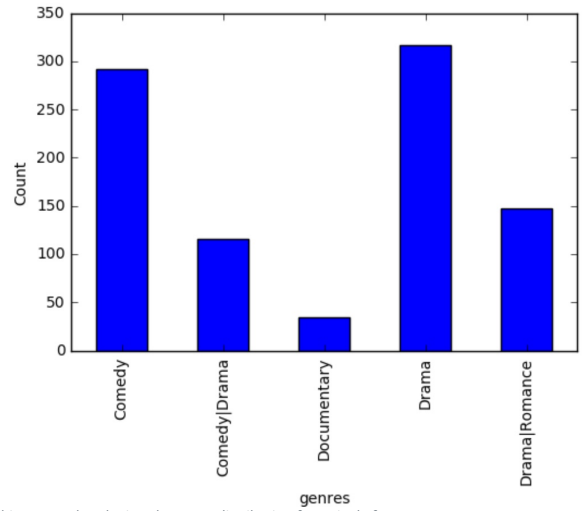
A bar plot that depicts the count distribution or other statistics for different categories.

Rate this review

```
df_new1.groupby([ 'genres'])['id'].count().plot(kind='bar').set_ylabel('Count')
```

```
df_new1.groupby(['genres' ])[['id']].count()
```

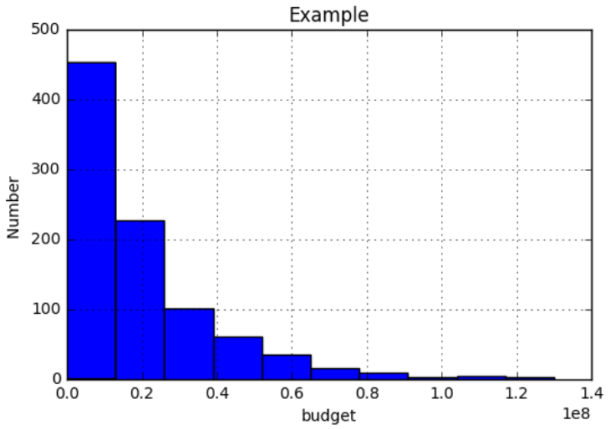
	id
genres	
Comedy	292
Comedy|Drama	116
Documentary	35
Drama	317
Drama|Romance	147



A histogram that depicts the count distribution for a single feature.

```
ax = df_new1['budget'].hist()
ax.set_ylabel('Number ')
ax.set_xlabel('budget')
ax.set_title('Example')
pd.DataFrame(df_new1['budget'].describe())
```

	budget
count	9.070000e+02
mean	1.914137e+07
std	1.981735e+07
min	5.000000e+00
25%	5.000000e+06
50%	1.350000e+07
75%	2.700000e+07
max	1.300000e+08

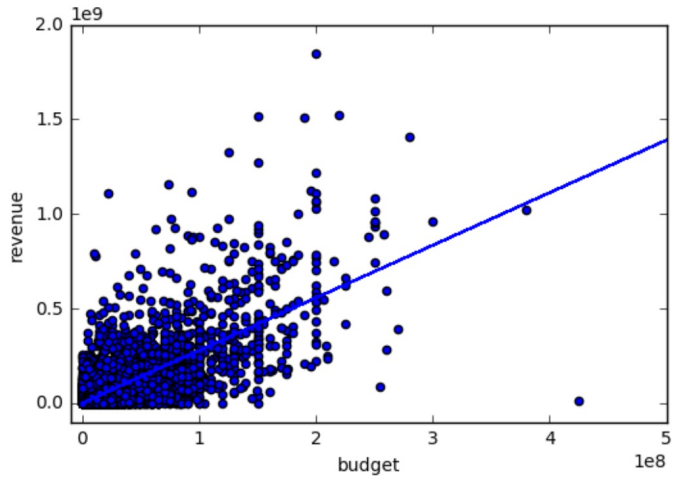


Rate this review

A scatter plot with the relevant correlation value,

```
import statsmodels.api as sm
import scipy
# regress "expression" onto "motifScore" (plus an intercept)
model = sm.OLS(df.revenue, sm.add_constant(df.budget))
p = model.fit().params
# generate x-values for your regression line (two is sufficient)
x = df.revenue
# scatter-plot data
ax = df.plot(x='budget', y='revenue', kind='scatter')
# plot regression line on the same axes, set x-axis limits
ax.plot(x, p.const + p.budget* x)
ax.set_xlim([-10000000,500000000])
ax.set_ylim([-100000000,2000000000])
print ("correlation :",scipy.stats.pearsonr(df.budget, df.revenue) )
```

```
correlation : (0.73490068190761171, 0.0)
```



Conclusions Phase

The results of the analysis are presented such that any limitations are clear. The analysis does not state or imply that one change causes another based solely on a correlation.

Excellent ! the report includes a discussion about the limitations and the shortcomings of the dataset and the analysis.

Communication

Reasoning is provided for each analysis decision, plot, and statistical summary.

The analysis follows a logical flow, the discussion includes reasonings, explanations about the analysis and relevant statistics to quantify the results and insights.

Visualizations made in the project depict the data in an appropriate manner that allows plots to be readily interpreted.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review