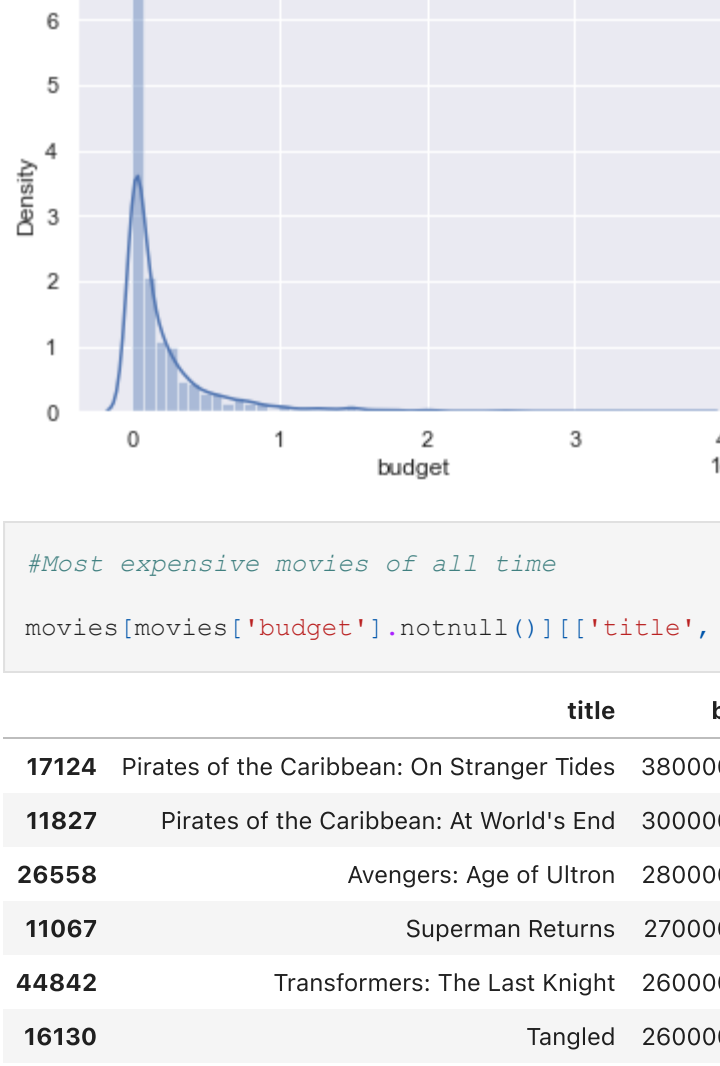


/Users/nicolaescontreras/miniconda3/envs/JupyterEnv/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: 'displot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'histplot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
<AxesSubplot: xlabel='budget', ylabel='Density'>

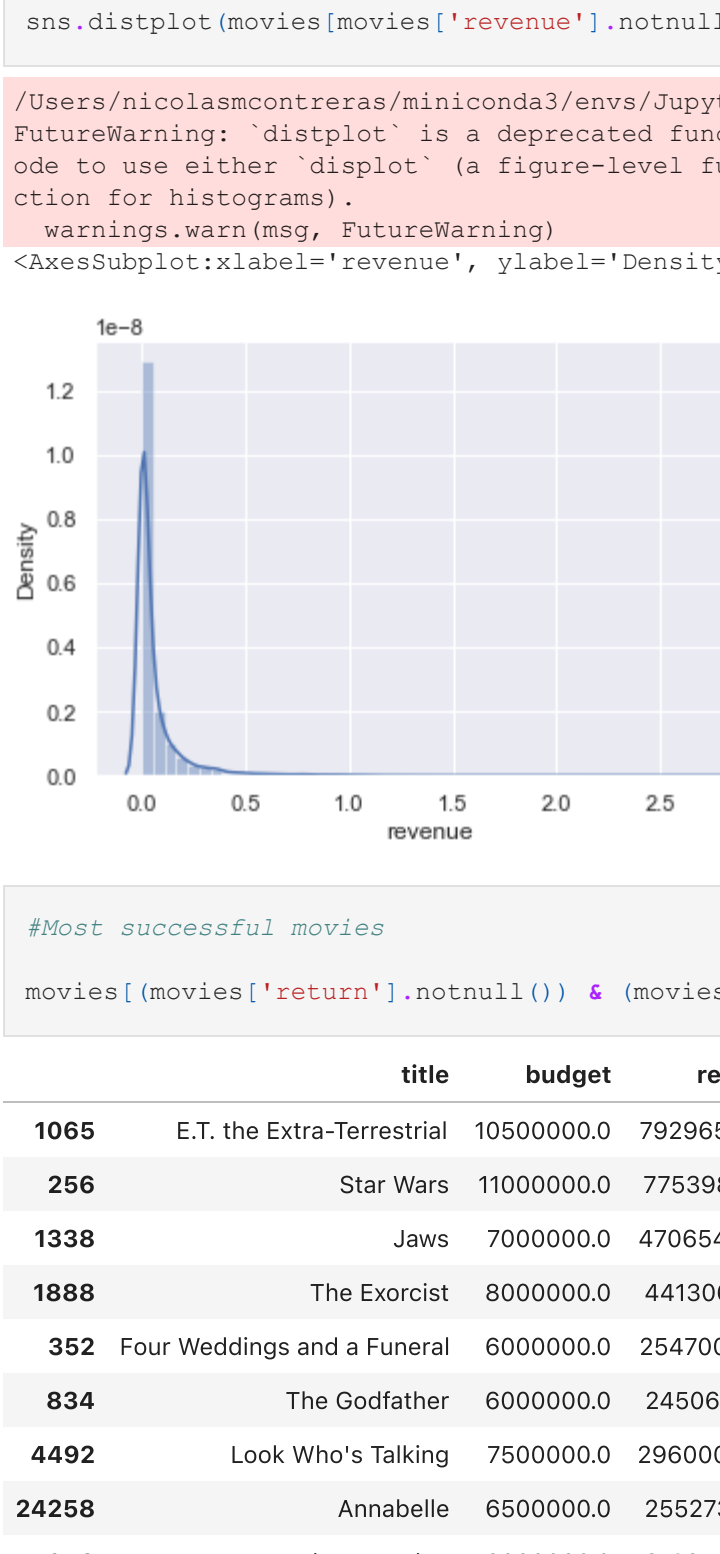


In [94]: #Most expensive movies of all time
movies[movies['budget'].notnull()][['title', 'budget', 'revenue', 'return', 'year']].sort_values('budget', ascending=True)

	title	budget	revenue	return	year
17124	Pirates of the Caribbean: On Stranger Tides	380000000.0	1.045714e+09	2.751678	2011
11827	Pirates of the Caribbean: At World's End	300000000.0	9.610000e+08	3.203333	2007
26558	Avengers: Age of Ultron	280000000.0	1.405404e+09	5.019299	2015
11067	Superman Returns	270000000.0	3.910812e+08	1.448449	2006
44842	Transformers: The Last Knight	260000000.0	6.049421e+08	2.326701	2017
16130	Tangled	260000000.0	5.817949e+08	2.276134	2010
18685	John Carter	260000000.0	2.841391e+08	1.092843	2012
11780	Spider-Man 3	258000000.0	8.908716e+08	3.452991	2007
21175	The Lone Ranger	255000000.0	8.928991e+07	0.350167	2013
22059	The Hobbit: The Desolation of Smaug	250000000.0	9.584000e+08	3.833600	2013

In [96]: #Correlation between budget and revenue
sns.jointplot(x='budget', y='revenue', data=movies[movies['return'].notnull()])

Out[96]: <seaborn.axisgrid.JointGrid at 0x7fc4003c0cd0>



In [98]: #Description of revenue variable
movies['revenue'].describe()

Out[98]: count 7.408000e+03
mean 6.878739e+07
std 1.464203e+08
min 1.000000e+00
25% 2.400000e+06
50% 1.682272e+07
75% 6.722707e+07
max 2.787965e+09
Name: revenue, dtype: float64

In [99]: #Distribution plot of revenue
sns.distplot(movies[movies['revenue'].notnull()]['revenue'])

/Users/nicolaescontreras/miniconda3/envs/JupyterEnv/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
<AxesSubplot: xlabel='revenue', ylabel='Density'>



In [100]: #Most successful movies
movies[(movies['return'].notnull()) & (movies['budget'] > 5e6)][['title', 'budget', 'revenue', 'return', 'year']].sort_values('return', ascending=False)

	title	budget	revenue	return	year
1065	E.T. the Extra-Terrestrial	105000000.0	792965326.0	75.520507	1982
256	Star Wars	110000000.0	775398007.0	70.490728	1977
1338	Jaws	70000000.0	470684000.0	67.236286	1975
1888	The Exorcist	80000000.0	441306145.0	55.163268	1973
352	Four Weddings and a Funeral	60000000.0	254700832.0	42.450139	1994
834	The Godfather	60000000.0	245066411.0	40.844402	1972
4492	Look Who's Talking	75000000.0	296000000.0	39.466667	1989
24258	Annabelle	65000000.0	255273813.0	39.272894	2014
1066	Dirty Dancing	60000000.0	213964274.0	35.659046	1987
1006	The Sound of Music	82000000.0	286214286.0	34.904181	1965

In [102]: #Most successful movies
movies[(movies['return'].notnull()) & (movies['budget'] > 5e6) & (movies['revenue'] > 10000)][['title', 'budget', 'revenue', 'return', 'year']].sort_values('return', ascending=False)

	title	budget	revenue	return	year
11159	Chaos	200000000.0	10399.0	0.000514	2005
19027	5 Days of War	200000000.0	17479.0	0.000874	2011
21034	Special Forces	100000000.0	10759.0	0.001076	2011
25732	Foodfight!	65000000.0	73706.0	0.001134	2012
38388	Term Life	165000000.0	21256.0	0.001289	2016
19505	Laurence Anyways	95000000.0	12250.0	0.001289	2012
12038	The Good Neighbor	150000000.0	20380.0	0.001359	2007
3966	Cherry 2000	100000000.0	14000.0	0.001400	1987
22097	Twice Born	130000000.0	18295.0	0.001407	2012
5651	All The Queen's Men	150000000.0	23000.0	0.001533	2001

In [103]: #Correlation matrix
movies['year'] = movies['year'].replace('NA', np.nan)
movies['year'] = movies['year'].apply(clean_numeric)

sns.set(font_scale=1)
corr = movies.corr()
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)] = True
with sns.axes_style('white'):
 plt.figure(figsize=(8,8))
 ax = sns.heatmap(corr, mask=mask, vmax=.3, square=True, annot=True)



In []:

In []:

In []: