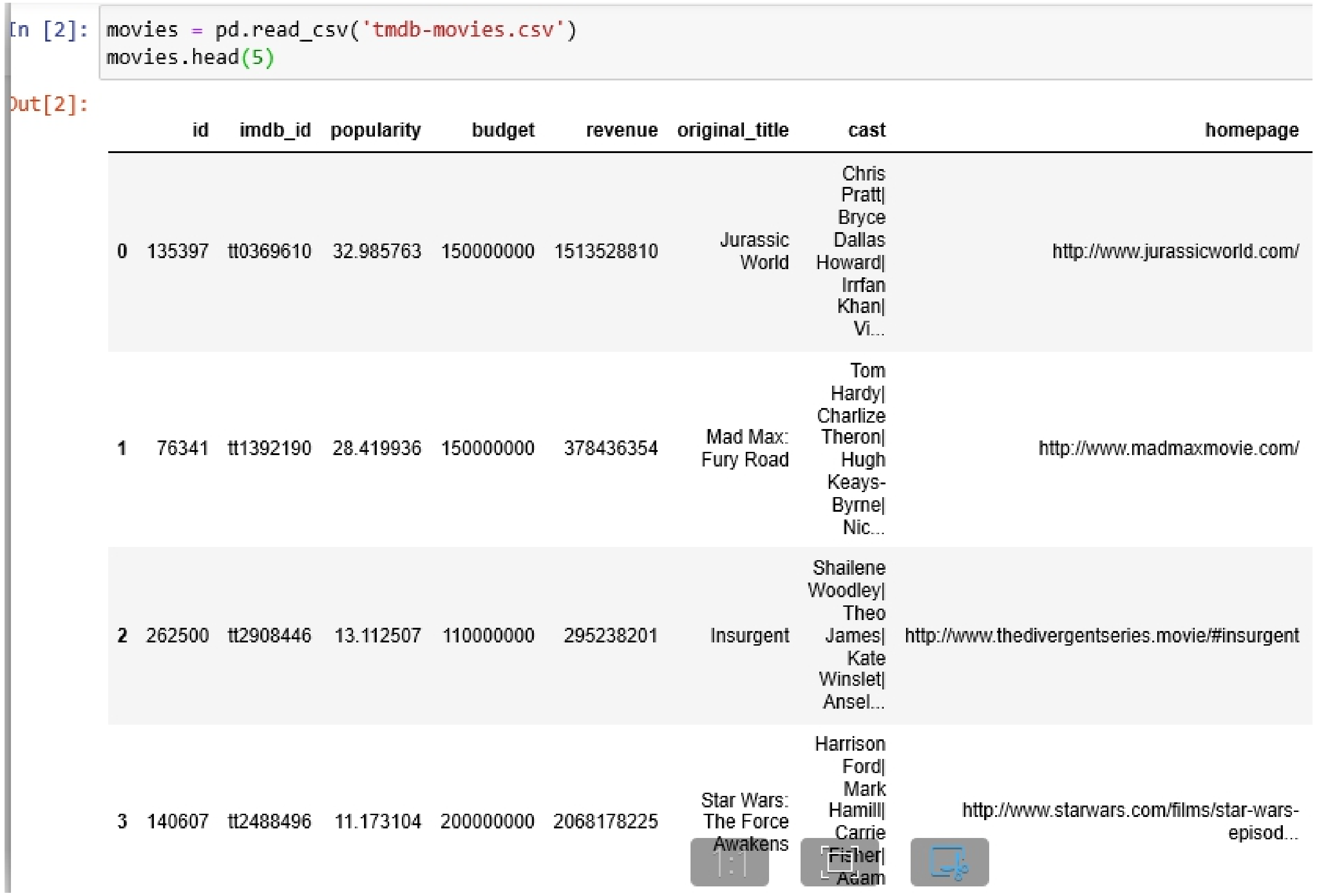
Wrangle_report..:)

1-Gathering and assessing the dataset

First,we obtained a dataset from Kaggel site called “**TMDb Movies Dataset**”, which is investigating dataset contains information about 10k+ movies collected from TMDb divided into 21 columns, **on which the model will be based .**

Second,we assessed the dataset to understand it correctly and to find out the existing issues related to data entry or otherwise to solve them.



```
In [2]: movies = pd.read_csv('tmdb-movies.csv')
        movies.head(5)
```

Out[2]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam	http://www.starwars.com/films/star-wars-episod...

-Data issues:

In such a noisy data,There are many issues which are divided into two types : 1-Tidiness issues . 2- Quality issues.

For tidiness issues we need to:

1- Drop dublicate data.
2- Merge the column of “ralease_year” with “release_date” .....

01

For quality issues we need to:

1-Dealing with missing values.
2-Change the erroneous datatypes.
3-Drop the nused columns....

02

In the next slide,we explain how to deel with data issues in a deeper way...:)

2-Data Cleaning:

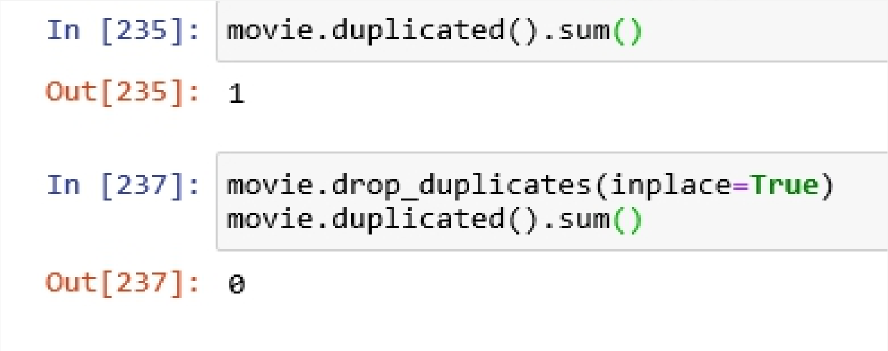
01.

-Drop unused columns that won't be useful.for our model.

02.

-Drop dublicate row

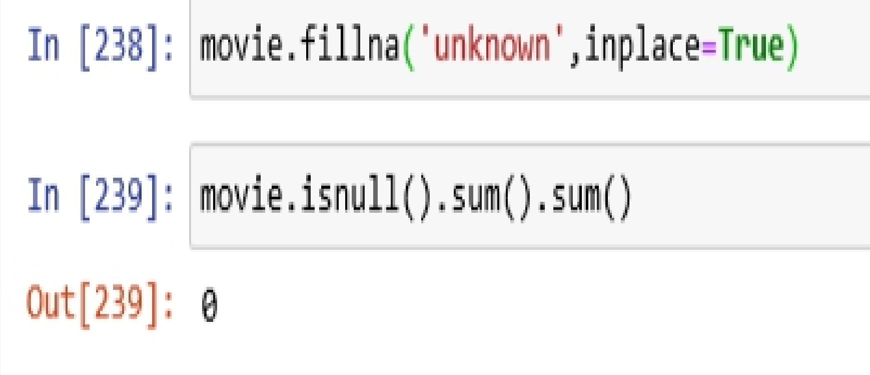
“”There are a lot of duplicated titles which no need to be cleaned.””



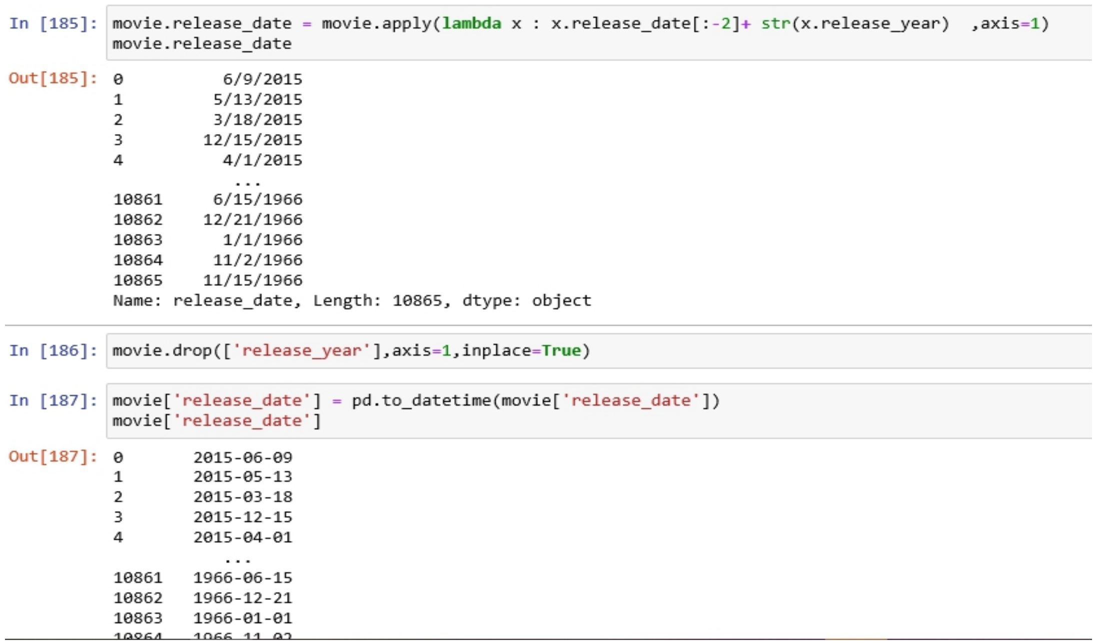
```
In [235]: movie.duplicated().sum()

Out[235]: 1

In [237]: movie.drop_duplicates(inplace=True)
          movie.duplicated().sum()

Out[237]: 0
```

03.

-All the columns which contain null values have “object” data type, so we replaced the null values with “unknown”.



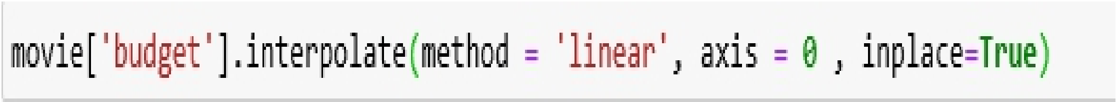
```
In [238]: movie.fillna('unknown',inplace=True)

In [239]: movie.isnull().sum().sum()

Out[239]: 0
```



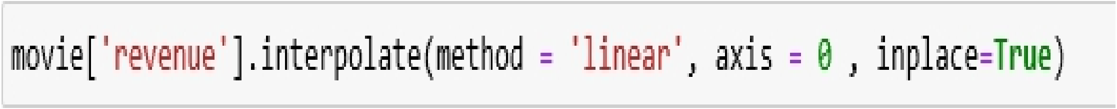
```
In [185]: movie.release_date = movie.apply(lambda x : x.release_date[:-2]+ str(x.release_year)  ,axis=1)
          movie.release_date

Out[185]: 0          6/9/2015
          1         5/13/2015
          2         3/18/2015
          3        12/15/2015
          4          4/1/2015
                      ...
          10861     6/15/1966
          10862    12/21/1966
          10863      1/1/1966
          10864     11/2/1966
          10865    11/15/1966
          Name: release_date, Length: 10865, dtype: object

In [186]: movie.drop(['release_year'],axis=1,inplace=True)

In [187]: movie['release_date'] = pd.to_datetime(movie['release_date'])
          movie['release_date']

Out[187]: 0       2015-06-09
          1       2015-05-13
          2       2015-03-18
          3       2015-12-15
          4       2015-04-01
                     ...
          10861   1966-06-15
          10862   1966-12-21
          10863   1966-01-01
          10864   1966-11-02
```

04.

-Make release date in a better format before converting its type,then remove the unneeded column of release year.



```
movie['budget'].interpolate(method = 'linear', axis = 0 , inplace=True)
```



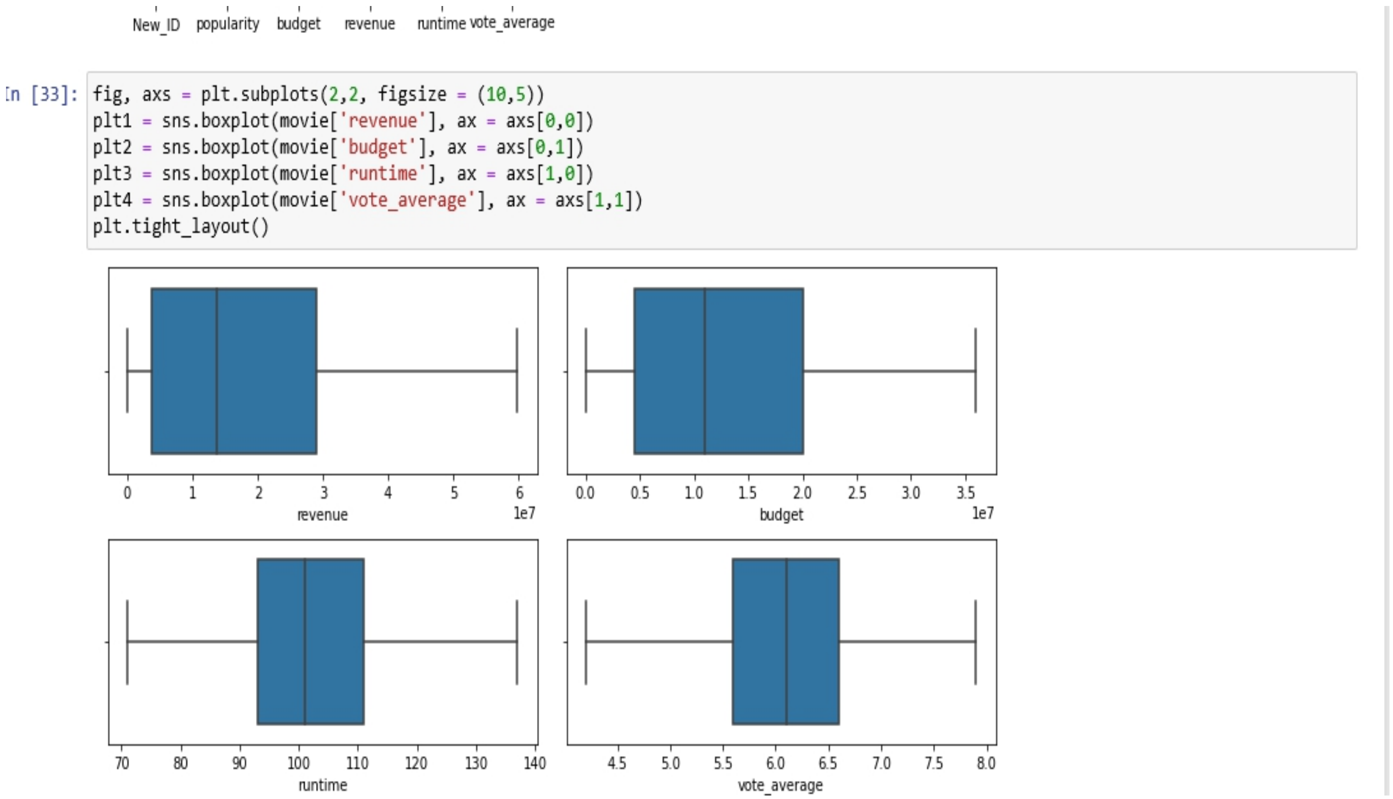
```
movie['revenue'].interpolate(method = 'linear', axis = 0 , inplace=True)
```

05.

-We replace “0” values in “budget, revenue” by using linear interpolation.

06. Removing outliers...:)

- -We used the statistical method of “IQR” to remove outliers.



```
In [33]: fig, axs = plt.subplots(2,2, figsize = (10,5))
         plt1 = sns.boxplot(movie['revenue'], ax = axs[0,0])
         plt2 = sns.boxplot(movie['budget'], ax = axs[0,1])
         plt3 = sns.boxplot(movie['runtime'], ax = axs[1,0])
         plt4 = sns.boxplot(movie['vote_average'], ax = axs[1,1])
         plt.tight_layout()
```