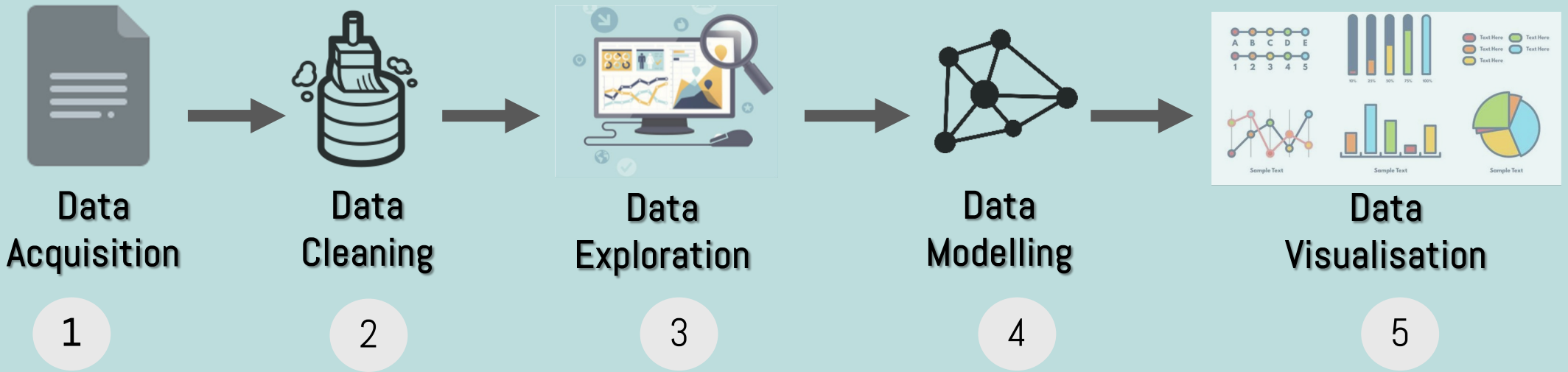
Stelios Sotiriadis

10. Data cleaning and Pandas

Part 1: Data Science Pipelines

Data Analysis steps

- Data Science is about creating pipelines to analyse data
  - What is a Data Science pipeline?
    - A set of steps to approach a problem

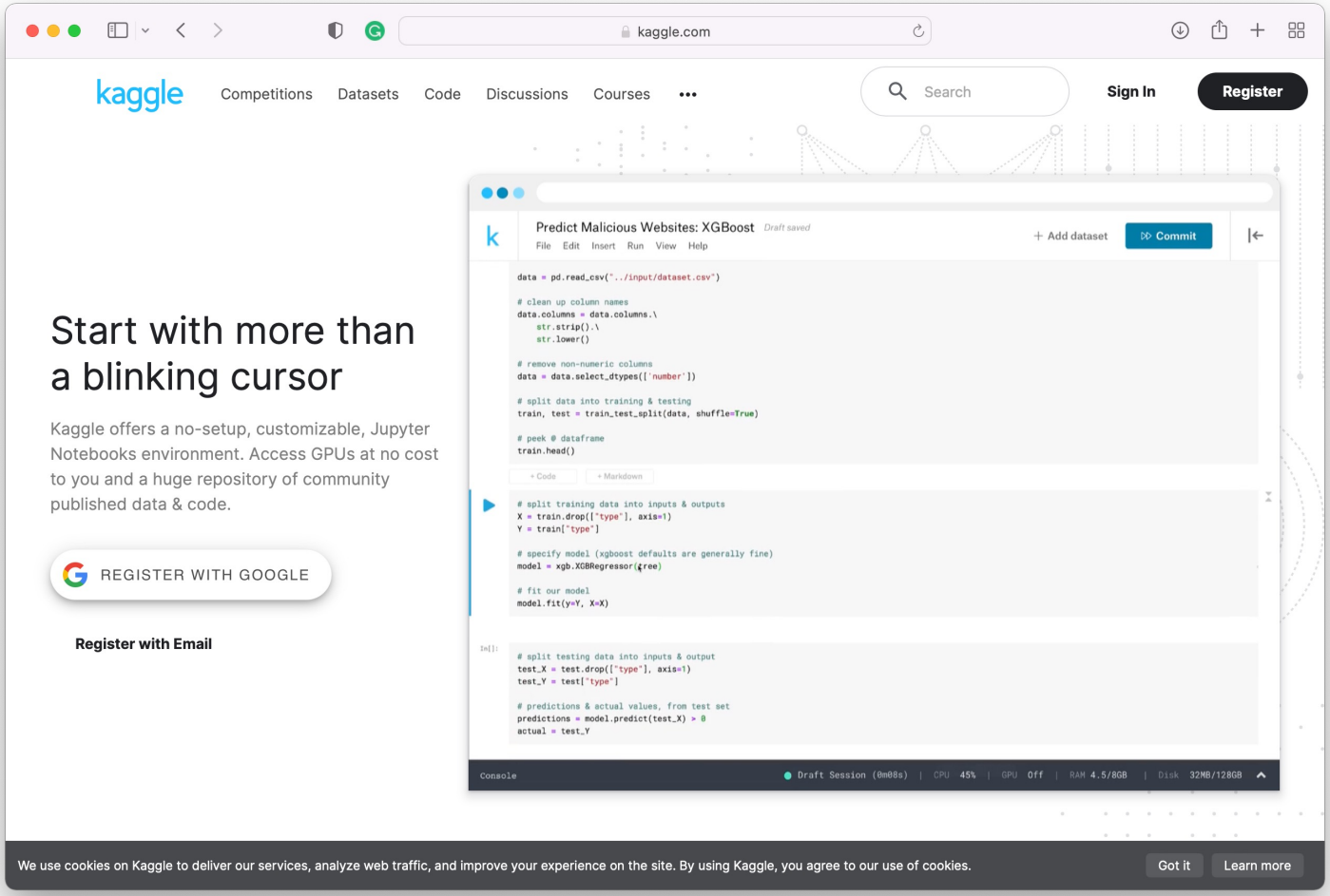


Step 1: Data Acquisition

- Data acquisition is the process of collecting data by from real world use cases.
- The data are usually numeric and can be manipulated.
- Data could be in any kind of format
  - Structured/Un-structured/Semi-structured

Step 1: Where to find data?

- https://www.kaggle.com
- In Kaggle you'll find code & data.
  - There are over 50,000 public datasets and 400,000 public notebooks.



Step 2: Data Cleaning

- The process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate or incomplete data in a dataset
- Data cleaning is important because it improves our data quality
  - All outdated or incorrect information is gone, leaving us with the highest quality information
- Today we will use DataFrames to clean our data and prepare it for analysis

Step 2: Data Cleaning

- Working with dataframes:
  - Extract sample data
  - Clean the data
    - We need to take decisions:
      - Do we drop the incomplete data?
      - Do we fill the missing values with other data?
      - Do we remove duplicated values?

Step 2: Data Cleaning

- Analysis:
  - Detect errors and inconsistencies
    - Manual and automated analysis...
  - Defining transformations and mapping rules
    - Having identified the problems, our next step is to find ways to fix it
  - Verifications:
    - Test and evaluate the transformations
  - Transformation:
    - Apply the transformation

Our focus today...

- Today we focus on:
  - Step 2: Data Cleaning
  - Step 3: Data Exploration
- Data cleaning:
    - We will work with a dataset and clean it (take decisions...)
- Data exploration:
    - We will visualise the dataset and explore the hidden information

Part 2: Data cleaning

2.1 Create a dataframe

- Let us create a dataframe and apply some basic data cleaning processes
- We will use the Movies.csv file

```
import pandas as pd

df = pd.read_csv('Movies.csv')
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
0	You Will Meet a Tall Dark Stranger	Comedy	Independent	35.0	1.211818	43	$26.66	2010
1	Love Happens	Drama	Universal	40.0	2.004444	18	$36.08	2009
2	The Heartbreak Kid	Comedy	Paramount	41.0	2.129444	30	$127.77	2007
3	When in Rome	Comedy	Disney	44.0	0.000000	15	$43.04	2010
4	Killers	Action	Lionsgate	45.0	1.245333	11	$93.40	2010

2.2 Understand the data schema

- Examine the data schema:
  - We have 8 columns

Object means String (Text)

```
[6]  df.dtypes

Film                  object
Genre                 object
Lead Studio           object
Audience score %     float64
Profitability        float64
Rotten Tomatoes %      int64
Worldwide Gross       object
Year                   int64
dtype: object
```

When at least 60% of **reviews** for a movie or TV show are positive, a red **tomato** is displayed to indicate its Fresh status. When less than 60% of **reviews** for a movie or TV show are positive, a green splat is displayed to indicate its **Rotten** status.

2.2 Understand the data schema (cont.)

- Examine the data schema:
  - Let us focus on the numerical fields

```
[104] df.describe()
```

	Audience score %	Profitability	Rotten Tomatoes %	Year
count	74.000000	76.000000	76.000000	76.000000
mean	63.378378	4.637009	46.473684	2009.092105
std	13.707015	8.078562	26.191334	1.358211
min	35.000000	0.000000	3.000000	2007.000000
25%	52.000000	1.775796	26.000000	2008.000000
50%	63.000000	2.645711	44.000000	2009.000000
75%	75.500000	5.163243	63.500000	2010.000000
max	89.000000	66.934000	96.000000	2011.000000

2.3 Examine a sample

- Examine a data sample:
  - Let us print the first 5 rows

```
[7] df.head()
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
0	You Will Meet a Tall Dark Stranger	Comedy	Independent	35.0	1.211818	43	$26.66	2010
1	Love Happens	Drama	Universal	40.0	2.004444	18	$36.08	2009
2	The Heartbreak Kid	Comedy	Paramount	41.0	2.129444	30	$127.77	2007
3	When in Rome	Comedy	Disney	44.0	0.000000	15	$43.04	2010
4	Killers	Action	Lionsgate	45.0	1.245333	11	$93.40	2010

2.3 Examine a sample (cont.)

- Examine a data sample:
  - Or we can print the last 5 rows

```
df.tail()
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
71	My Week with Marilyn	Drama	The Weinstein Company	84.0	0.825800	83	$8.26	2011
72	Midnight in Paris	Romence	Sony	84.0	8.744706	93	$148.66	2011
73	Tangled	Animation	Disney	88.0	1.365692	89	$355.01	2010
74	A Dangerous Method	Drama	Independent	89.0	0.448645	79	$8.97	2011
75	WALL-E	Animation	NaN	89.0	2.896019	96	$521.28	2008

Detect errors and inconsistencies

2.4 Identify problems

- Examine if we have missing or empty records in our dataset
  - We can use the isnull() method.
  - And then the sum() method to sum how many null values.

```
[15] df.isnull().sum()

Film                  0
Genre                 0
Lead Studio           1
Audience score %      2
Profitability         0
Rotten Tomatoes %     0
Worldwide Gross       0
Year                  0
dtype: int64
```

Missing values...

Sparsity: 3/76 = 4%
Density: 73/76 = 96%

2.4 Identify problems (cont.)

- It looks like there is an empty ‘Lead Studio’ record for the Film called ‘WALL-E’.

```
[108] df[df['Lead Studio'].isnull()]
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
75	WALL-E	Animation	NaN	89.0	2.896019	96	$521.28	2008

NaN is empty cell!
NaN: Not a Number

2.5 Decide how to handle it

- **1st way**
  - We can simple add the keyword ‘missing’ if we like to keep the record as it is.

```
[34] df['Lead Studio'].fillna('missing', inplace = True)
```

```
df.tail()
```

We know about it...

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
71	My Week with Marilyn	Drama	The Weinstein Company	84.0	0.825800	83	$8.26	2011
72	Midnight in Paris	Romence	Sony	84.0	8.744706	93	$148.66	2011
73	Tangled	Animation	Disney	88.0	1.365692	89	$355.01	2010
74	A Dangerous Method	Drama	Independent	89.0	0.448645	79	$8.97	2011
75	WALL-E	Animation	missing	89.0	2.896019	96	$521.28	2008

2.5 Decide how to handle it (cont.)

- **2nd way**
  - We can try to fix this by updating the appropriate value.
  - This requires a lot of manual tasks, however it is essential as this column is important for our data analysis
  - In our case, we can update the ‘Lead Studio’ using to ‘Disney’.

```
[122] df.at[75,'Lead Studio'] = 'Disney'
      df.loc[[75]]
```

It’s a manual task…

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year	Studio
75	WALL-E	Animation	Disney	89.0	2.896019	96	$521.28	2008	Disney

2.5 Decide how to handle it (cont.)

- **3rd way**
  - If we don’t miss a lot of data and we do not know the values we can simply delete it, to avoid doing assumptions…
  - Drop the rows where at least one element is missing:

```
[ ]  df.dropna()
```

2.5 Decide how to handle it (cont.)

- **4th way**
  - What if we have numeric data missing and we do not know the values?
  - Let us examine the 'Audience score' data.

```
[70] df[df['Audience score %'].isnull()]
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
37	It's Complicated	Comedy	Universal	NaN	2.642353	56	$224.60	2009
70	Across the Universe	romance	Independent	NaN	0.652603	54	$29.37	2007

There are two empty records!

2.5 Decide how to handle it (cont.)

- **4th Way**
  - Replace with average:
    - Let us replace the empty values with the average audience score (mean).
      - This is the overall average of all the Audience score % records.
    - By printing the data for row 37, the value looks good!

Automated analysis (Good example)

```
[97] df['Audience score %'].fillna(df['Audience score %'].mean(), inplace = True)
     df.iloc[35:40]
```

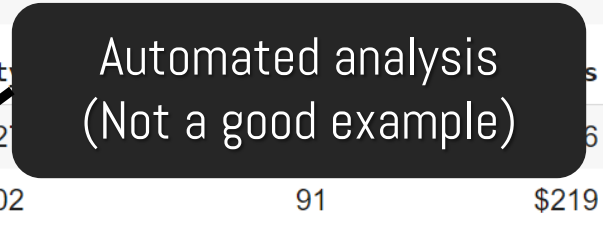
	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
35	Life as We Know It	Comedy	Independent	62.000000	2.530526	28	$96.16	2010
36	Letters to Juliet	Comedy	Summit	62.000000	2.639353	40	$79.18	2010
37	It's Complicated	Comedy	Universal	63.378378	2.642353	56	$224.60	2009
38	No Reservations	Comedy	Warner Bros.	64.000000	3.307180	39	$92.60	2007
39	A Serious Man	Drama	Universal	64.000000	4.382857	89	$30.68	2009

2.5 Decide how to handle it (cont.)

- **4th Way**
  - Let's examine data for value 70…
  - Data does not look good!
    - 63.37 is the average of all scores and looks quite random…

```
df.iloc[68:73]
```

	Film	Genre	Lead Studio	Audience score %	Profitabilit		s	Year
68	Twilight	Romance	Summit	82.000000	10.1		6	2008
69	Knocked Up	Comedy	Universal	83.000000	6.636402	91	$219	2007
70	Across the Universe	romance	Independent	63.378378	0.652603	54	$29.37	2007
71	My Week with Marilyn	Drama	The Weinstein Company	84.000000	0.825800	83	$8.26	2011
72	Midnight in Paris	Romence	Sony	84.000000	8.744706	93	$148.66	2011



2.5 Decide how to handle it (cont.)

- **4th Way**
  - Is there an alternative? Let us examine the dataset once more!
  - It seems that the 'Audience score' records are sorted in an ascending order (from 35 to 89).

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
0	You Will Meet a Tall Dark Stranger	Comedy	Independent	35.000000	1.211818	43	$26.66	2010
1	Love Happens	Drama	Universal	40.000000	2.004444	18	$36.08	2009
2	The Heartbreak Kid	Comedy	Paramount	41.000000	2.129444	30	$127.77	2007
3	When in Rome	Comedy	Disney	44.000000	0.000000	15	$43.04	2010
4	Killers	Action	Lionsgate	45.000000	1.245333	11	$93.40	2010
...	...	...	...	...	...	...	...	...
70	Across the Universe	romance	Independent	63.378378	0.652603	54	$29.37	2007
71	My Week with Marilyn	Drama	The Weinstein Company	84.000000	0.825800	83	$8.26	2011
72	Midnight in Paris	Romence	Sony	84.000000	8.744706	93	$148.66	2011
73	Tangled	Animation	Disney	88.000000	1.365692	89	$355.01	2010
74	A Dangerous Method	Drama	Independent	89.000000	0.448645	79	$8.97	2011

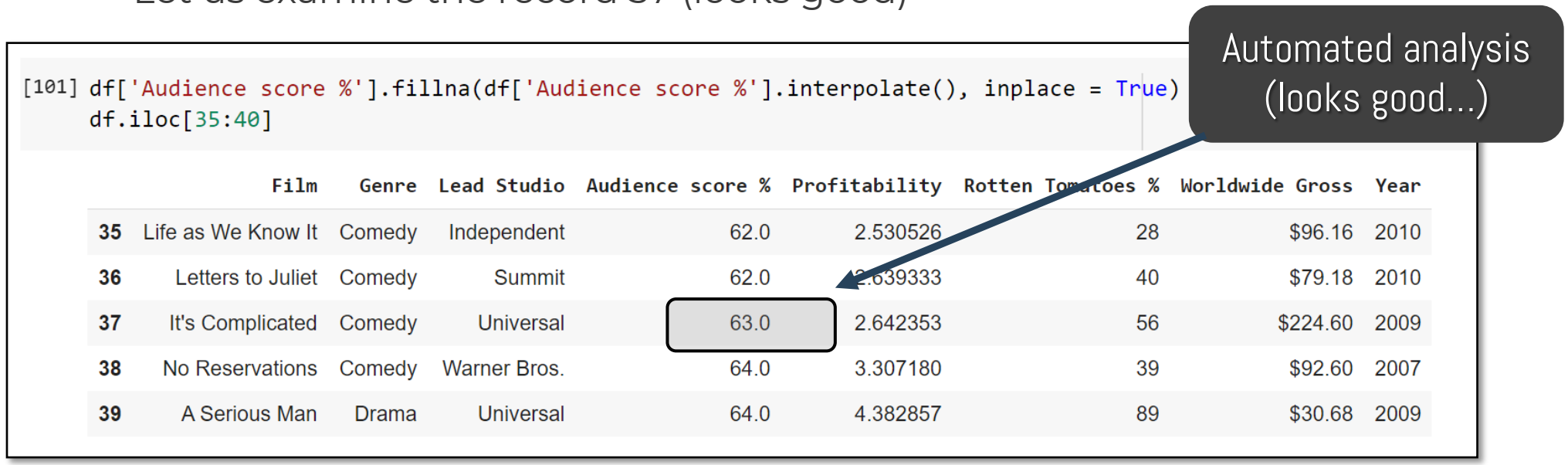
75 rows × 8 columns

2.5 Decide how to handle it (cont.)

- **4th Way**
  - We can use the interpolate method.
    - Interpolation is the process of finding a value between two points on a line or a curve.
    - Let us examine the record 37 (looks good)

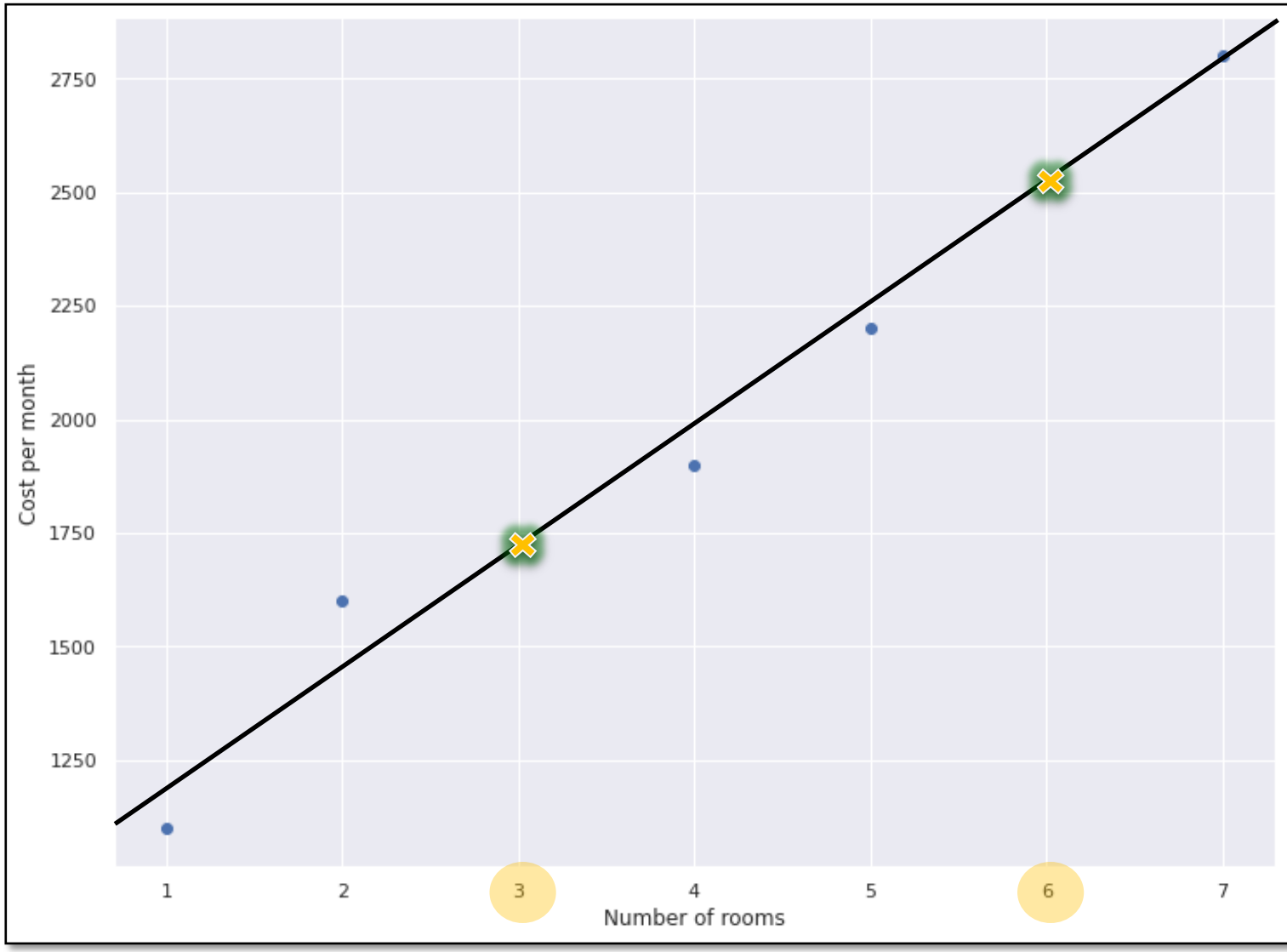
```
[101] df['Audience score %'].fillna(df['Audience score %'].interpolate(), inplace = True)
      df.iloc[35:40]
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
35	Life as We Know It	Comedy	Independent	62.0	2.530526	28	$96.16	2010
36	Letters to Juliet	Comedy	Summit	62.0	2.639333	40	$79.18	2010
37	It's Complicated	Comedy	Universal	63.0	2.642353	56	$224.60	2009
38	No Reservations	Comedy	Warner Bros.	64.0	3.307180	39	$92.60	2007
39	A Serious Man	Drama	Universal	64.0	4.382857	89	$30.68	2009



What is interpolation?

- Consider this example:
  - Number of rooms x = [1,2,4,5,7]
  - Cost per month
    y = [1100,1600,1900,2200,2800]
- We are missing values for some number of rooms:
  - **3 and 6**
- The interpolation will draw a best fit line
- It will help us to define missing values!
  - More to come!



2.5 Decide how to handle it (cont.)

- **Way 4**
  - Let us examine record 69 using the interpolate method.

Automated analysis (looks good…)

```
[102] df.iloc[68:73]
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
68	Twilight	Romance	Summit	82.0	10.180027	49	$376.66	2008
69	Knocked Up	Comedy	Universal	83.0	6.636402	91	$219	2007
70	Across the Universe	romance	Independent	83.5	0.652603	54	$29.37	2007
71	My Week with Marilyn	Drama	The Weinstein Company	84.0	0.825800	83	$8.26	2011
72	Midnight in Paris	Romence	Sony	84.0	8.744706	93	$148.66	2011

Summarising data cleaning steps

- Data cleaning is about identifying problems and errors in the dataset.
  - 2.1 Create a dataframe
  - 2.2 Understand the data schema
  - 2.3 Examine a sample
  - 2.4 Identify problems
  - 2.5 Decide how to handle the problems

Part 3: Visualisations

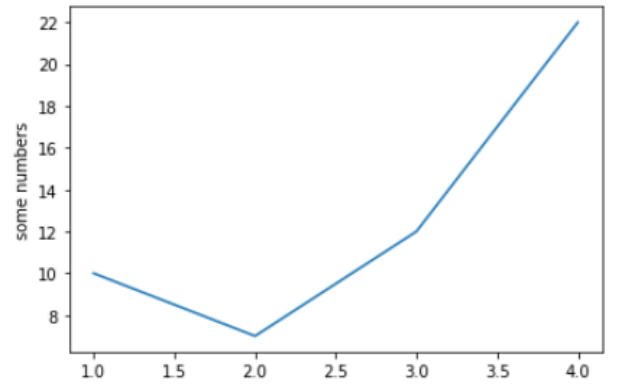
Visualisations

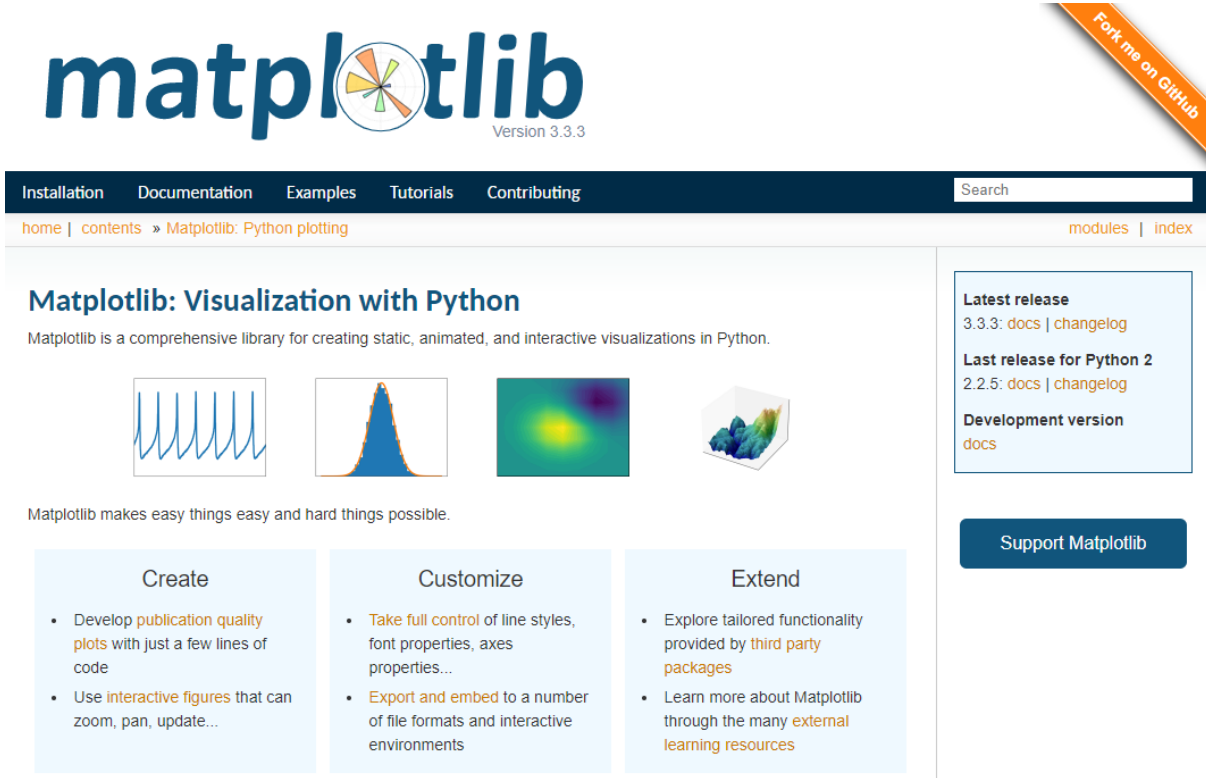
- In many cases we need to visualise datasets in order to explore the hidden relationships
- Python provides powerful visualisation tools:
  - Matplotlib
  - Seaborn

Matplotlib

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python [https://matplotlib.org/]

```
[147] import matplotlib.pyplot as plt
      x = [1, 2, 3, 4]
      y = [10,7,12,22]
      plt.plot(x,y)
      plt.ylabel('some numbers')
      plt.show()
```





Seaborn

- Seaborn is a Python data visualization library based on matplotlib.
- It provides a high-level interface for drawing attractive and informative statistical graphics.

[https://seaborn.pydata.org/]

```python
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f10b1d80e80>
```

3.1 Importing the libraries

- Let us import the appropriate libraries and produce our first visualisations.

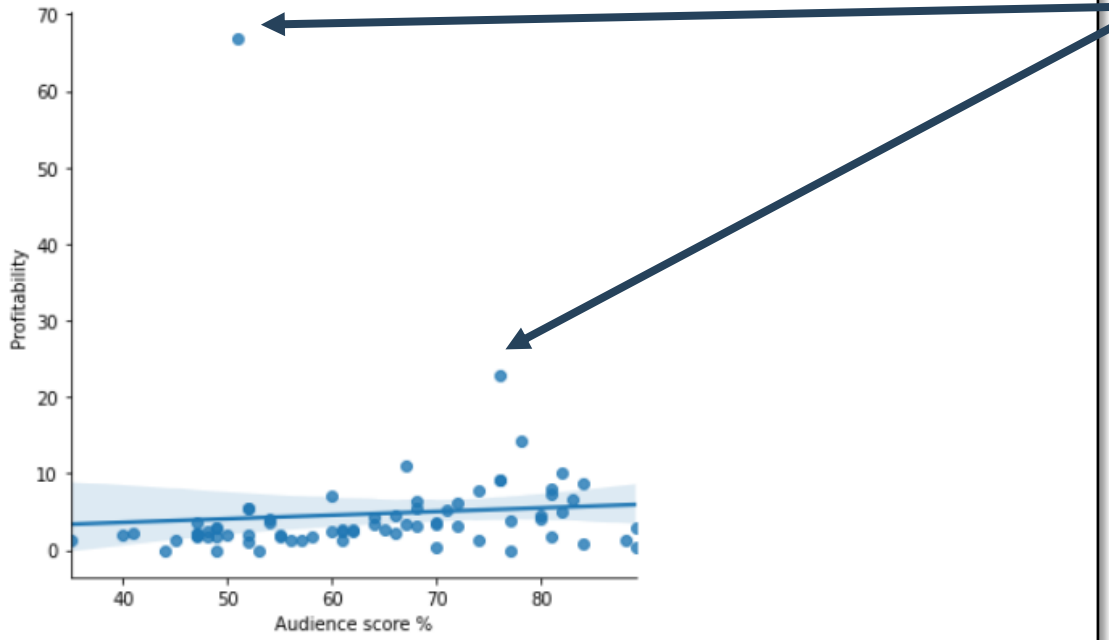
```
[124] import pandas as pd

      # Matplotlib for additional customization
      from matplotlib import pyplot as plt
      %matplotlib inline

      # Seaborn for plotting and styling
      import seaborn as sns
```

3.2 Visualise the data

- Making a scatterplot using the lmplot method
- Examine the visualisation

```
sns.lmplot(x='Audience score %', y='Profitability', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f10b226f860>
```

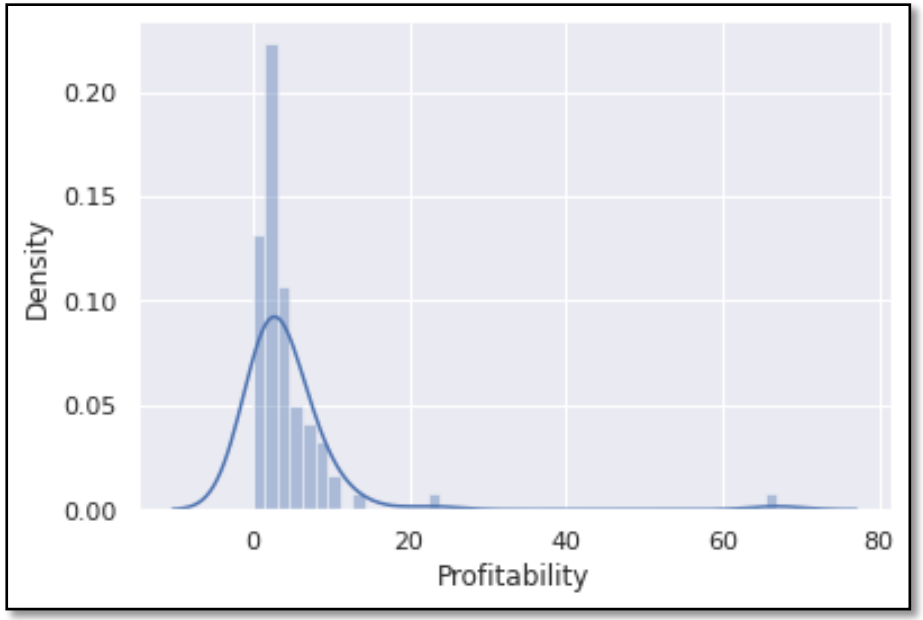


It seems that there are two outliers!

Outlier: An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.

3.2 Let us see the distributions

- This data is skewed!
  - A distribution is said to be skewed when the data points cluster more toward one side of the scale than the other, creating a curve that is not symmetrical.




```
sns.distplot(df['Profitability'])
```

3.2 Visualise the data (cont.)

- Let us identify and remove these outlier so we can have a better visualisation
- It looks that these values are higher than 20 (Profitability)

```
[138] df[df.Profitability >20 ]
```

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide Gross	Year
17	Fireproof	Drama	Independent	51.0	66.934000	40	$33.47	2008
58	High School Musical 3: Senior Year	Comedy	Disney	76.0	22.913136	65	$252.04	2008

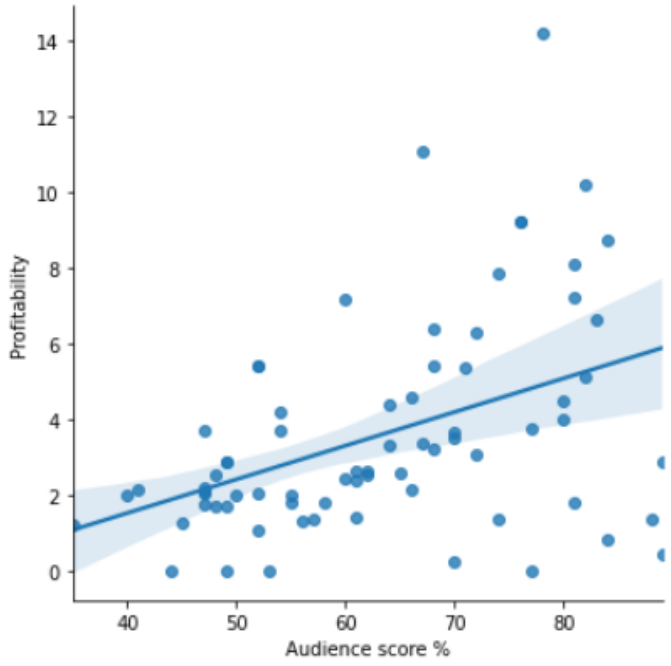
```
[139] df = df.drop([17,58], axis=0)
```

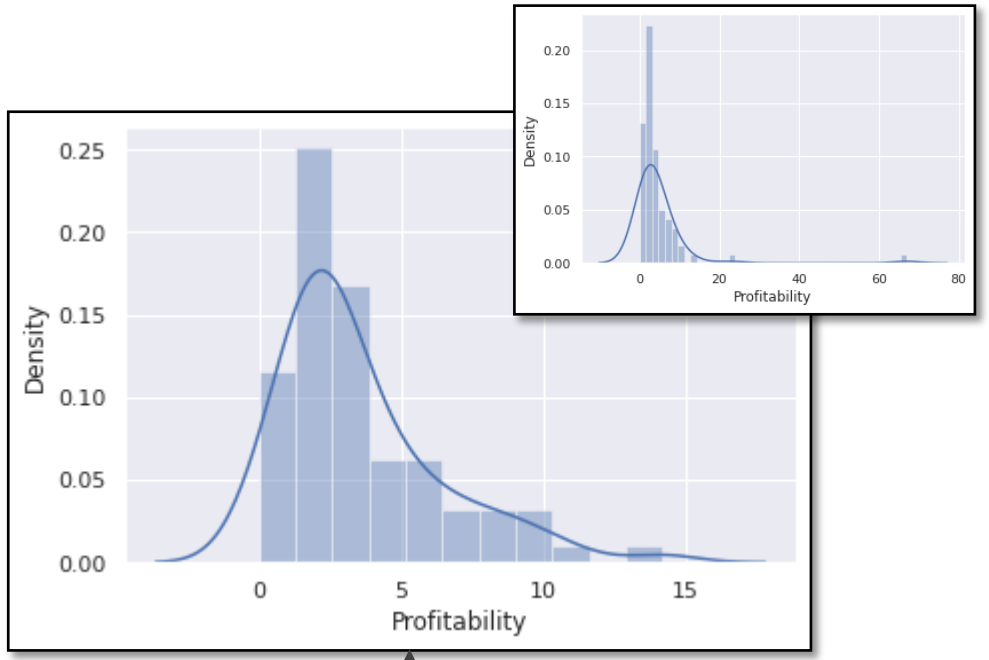
3.2 Visualise the data (cont.)

- Visualise the data once more

```
[140] sns.lmplot(x='Audience score %', y='Profitability', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f10b21ab2b0>
```



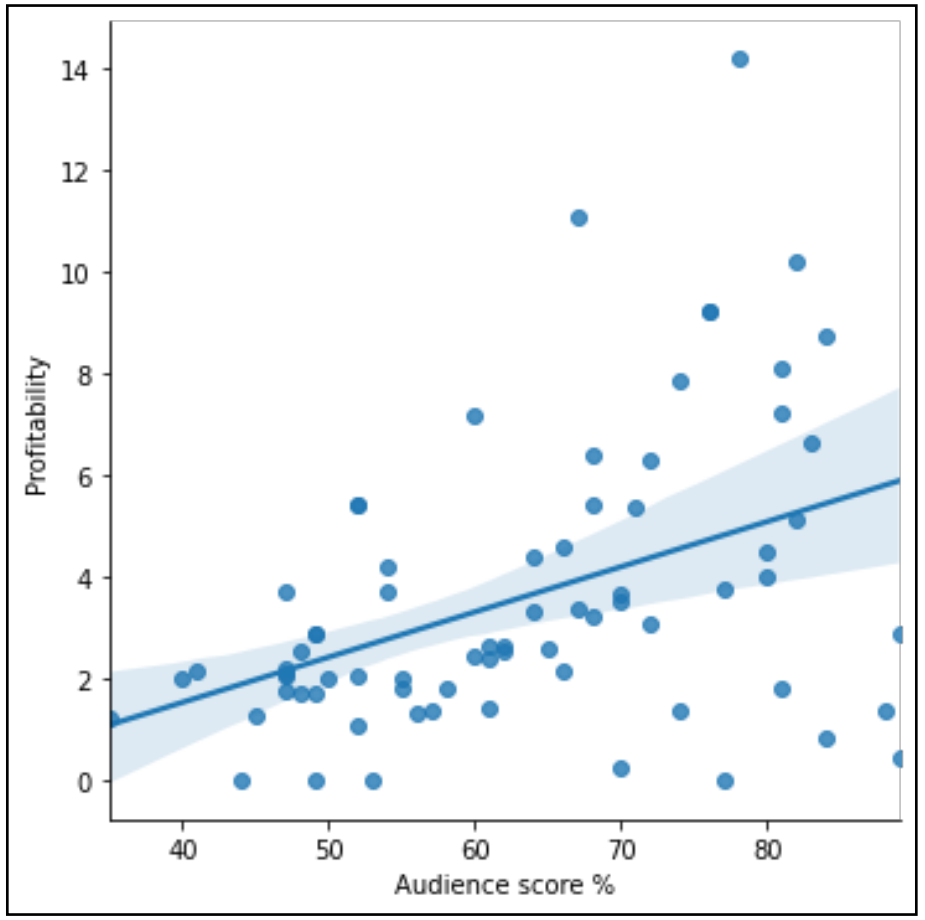


Looks better…

Looks skewed…but better

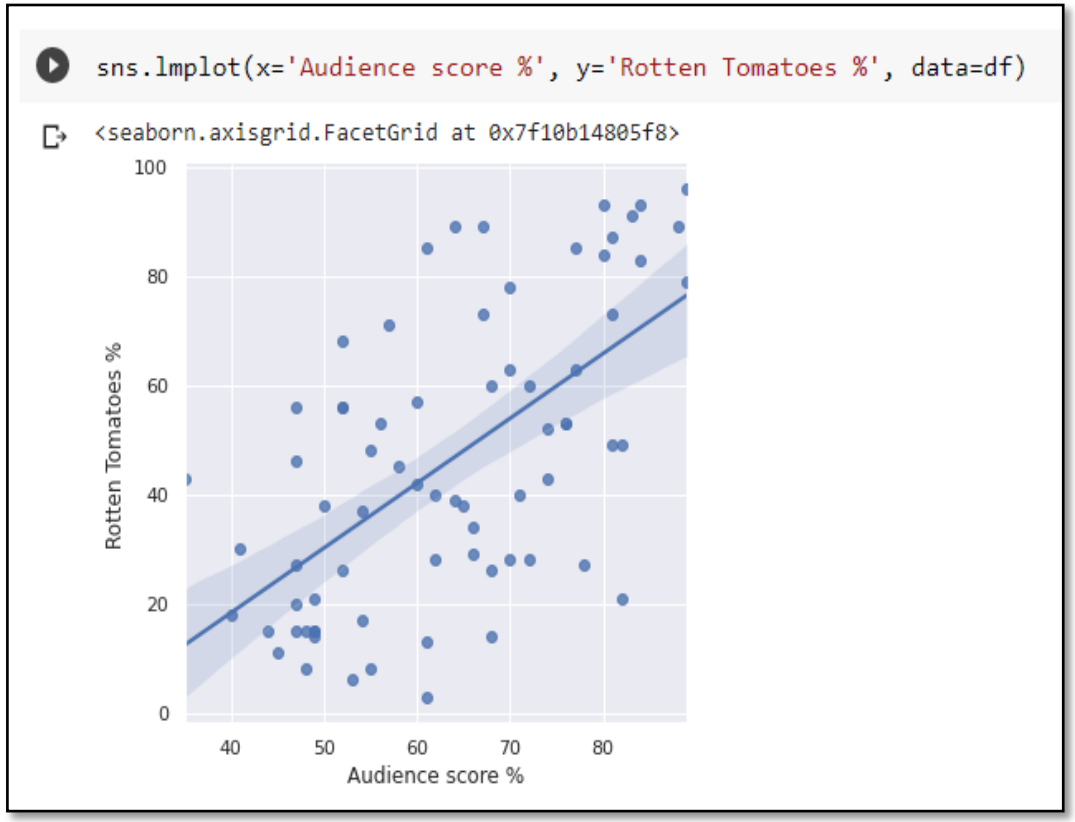
Why visualisations are useful?

- During data examination, visualisations will help us demonstrate the hidden relationships....
- There is a clear relationship between Audience score % and Profitability
- In general the higher the audience score a movie made, the highest the profits (as expected)



Let us study the lmplot method

- This method creates a scatterplot and adds a regression line.
  - A straight line that describes the relationship between x and y points.
- In our case, the line shows a trend
  - The highest the audience score the highest the rotten tomatoes score.
  - We will learn about regression in the next class

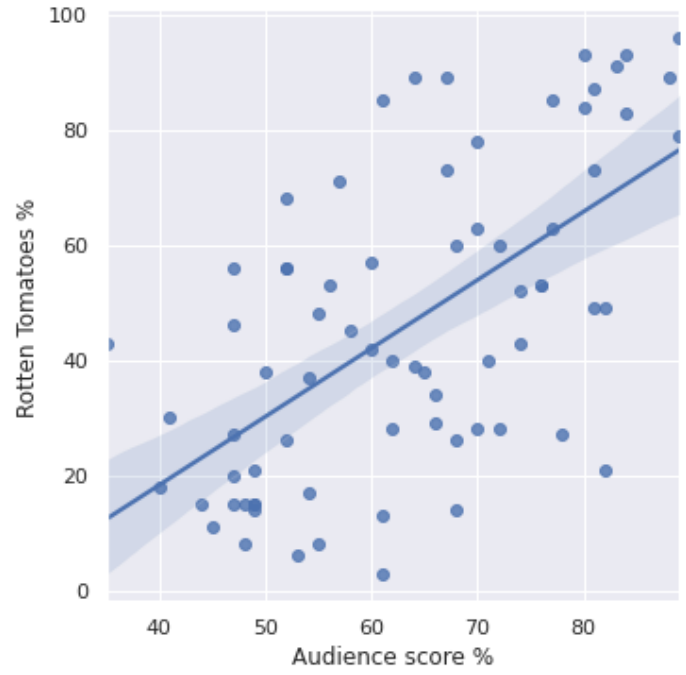


Regression vs Interpolation?

- Regression
  - The process of finding the line that best describes the data
    - (also known as best fit)
- Interpolation
  - The process of using the line of best fit to estimate the value of one variable from the value of another

```
sns.lmplot(x='Audience score %', y='Rotten Tomatoes %', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f10b14805f8>
```

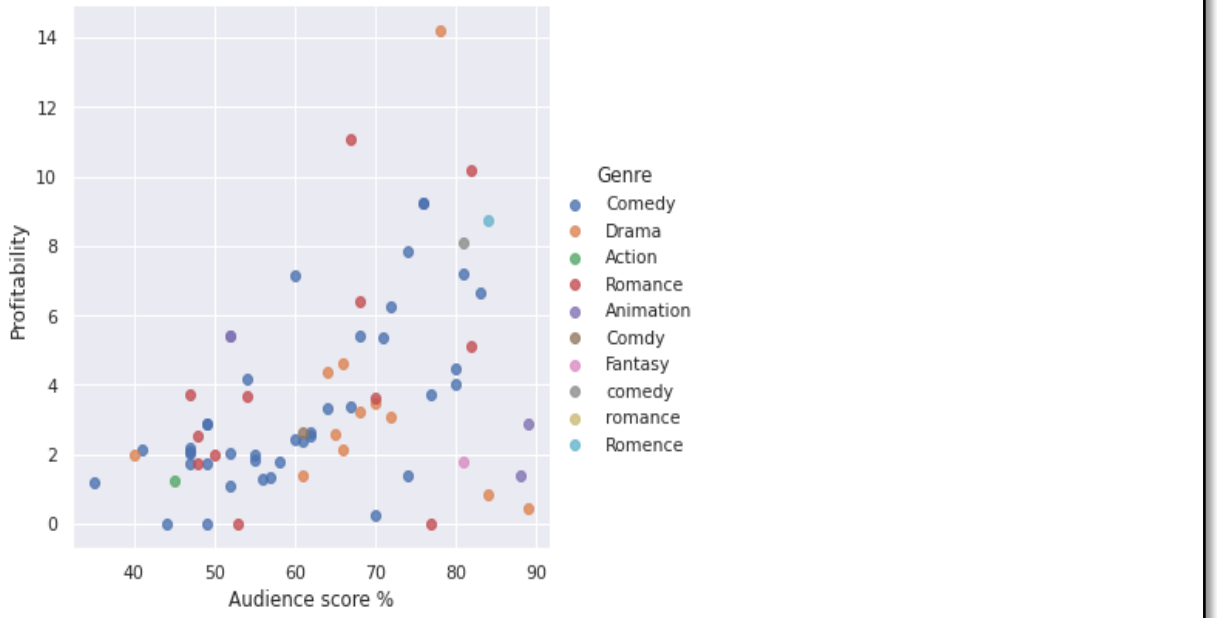


Using the seaborn (Scatterplot with hue)

- Let us produce a scatterplot where colors are according to Genre
  - This scatterplot does not have a regression line

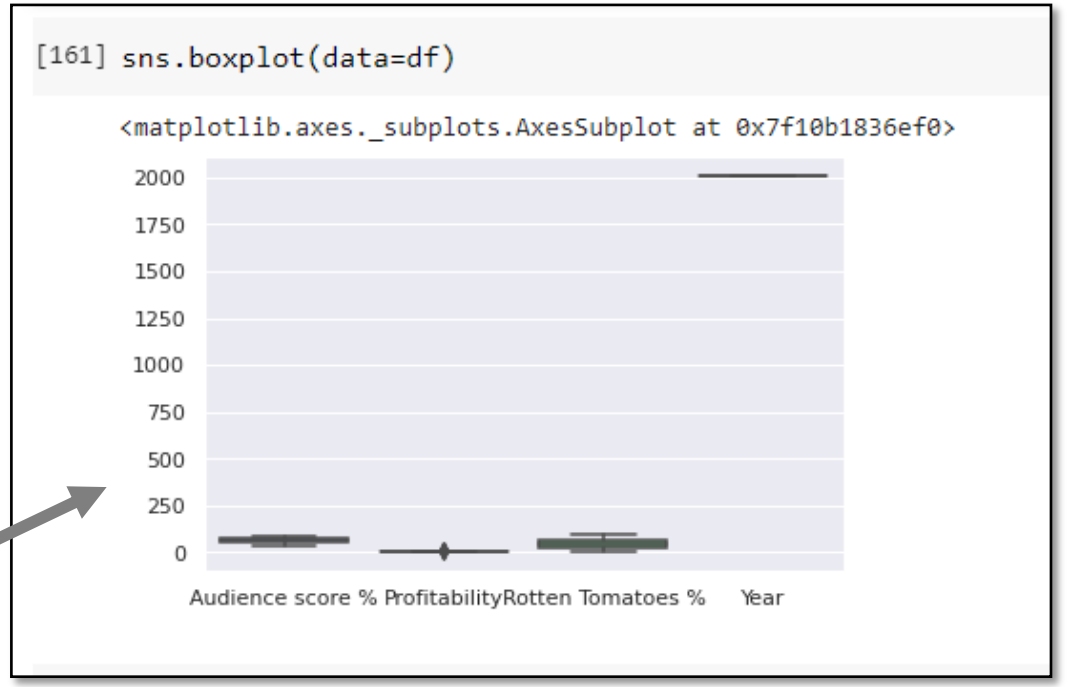
```
# Scatterplot arguments
sns.lmplot(x='Audience score %', y='Profitability', data=df, fit_reg=False, hue='Genre')
```

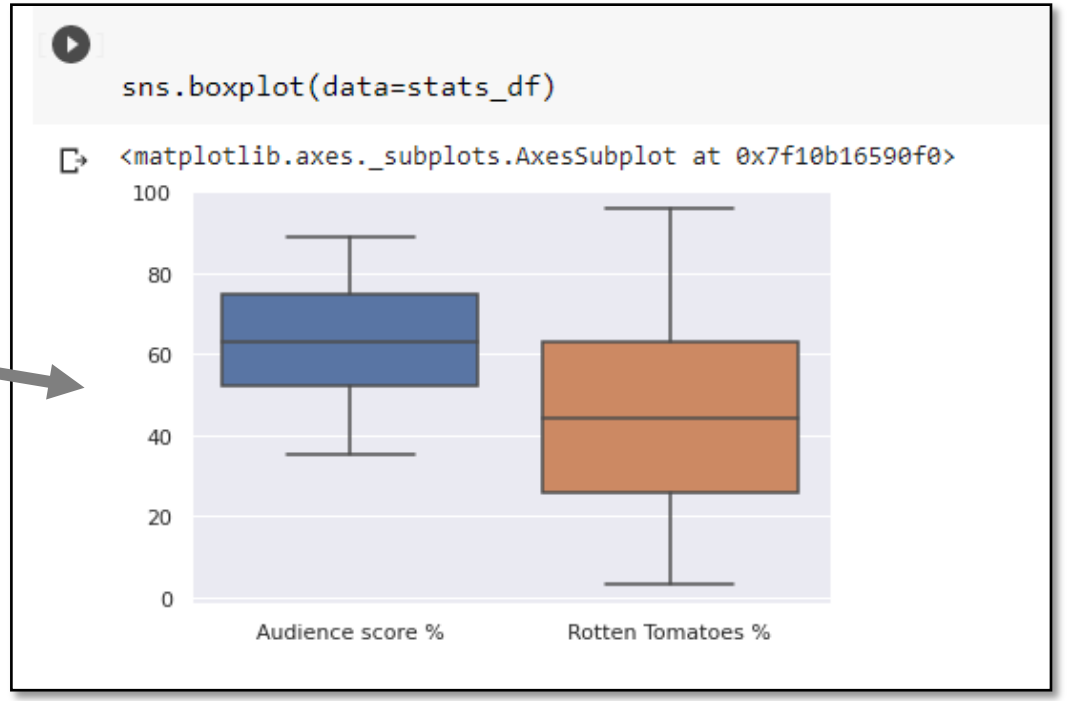
```
<seaborn.axisgrid.FacetGrid at 0x7f10b204f3c8>
```



Using the seaborn (Boxplot)

- Different data scales can make diagrams hard to read
- The boxplot visually show the distribution of numerical data and skewness
- Firstly, we produce a scatterplot for all the data (diagram is hard to read)
- Secondly, we remove the columns that cause problems (e.g., Year and Profitability)
- Then we plot once more!





Lab 10 tasks

- Task:
  - Follow the Lab 10 on how to use Pandas.
  - Download the PWD_Class_10_Questions.ipynb
  - Solutions are online