# Data Cleaning (meta_data)

Juan David Serna Valderrama

*Read .CSV file (meta_data)*

```
#Pandas library
import pandas as pd
import numpy as np

df = pd.read_csv("C:/Users/juand/OneDrive/Escritorio/Recomendation Systems with Python/Data/movies_metadata.csv")
df.head()
```

```
C:\Users\juand\Anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:3145: DtypeWarning: Columns (10) have mixed types.Spec
ify dtype option on import or set low_memory=False.
  has_raised = maxlast self.run_ast_nodes(code_ast.body, cell_name,
```

| | adult | belongs_to_collection | budget | genres | homepage | id | imdb_id | original_language | original_title | overview | ... | release_date | revenue | runtime | spoken_languag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | {'id': 10194, 'name': 'Toy Story Collection', ... | 30000000 | [{'id': 16, 'name': 'Animation'}, ... | http://www.toystory.com/buzz-movie | 862 | tt0114709 | en | Toy Story | Led by Woody, Andy's toys live happily in his room. | ... | 1995-10-30 | 373554033.0 | 81.0 | [{'iso_639_1': 'e 'name': 'Englis |
| 1 | False | Nah | 65000000 | [{'id': 12, 'name': 'Adventure'}, {'id': 14, ... | Nah | 8844 | tt0113497 | en | Jumanji | When siblings Judy and Peter discover an enchan... | ... | 1995-12-15 | 262797249.0 | 104.0 | [{'iso_639_1': 'e 'name': 'Fre Espa |
| 2 | False | {'id': 119050, 'name': 'Grumpy Old Men Collect... | 0 | [{'id': 10749, 'name': 'Romance'}, {'id': 35, ... | Nah | 15602 | tt0113228 | en | Grumpier Old Men | A family wedding reignites the long standing ri... | ... | 1995-12-22 | 0.0 | 101.0 | [{'iso_639_1': 'e 'name': 'Englis |
| 3 | False | Nah | 16000000 | [{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam... | Nah | 31357 | tt0114885 | en | Waiting to Exhale | Cheated on, mistreated and stepped on, the wo... | ... | 1995-12-22 | 81452156.0 | 127.0 | [{'iso_639_1': 'e 'name': 'Englis |
| 4 | False | {'id': 96081, 'name': 'Father of the Bride Coll... | 0 | [{'id': 35, 'name': 'Comedy'}] | Nah | 11062 | tt0113041 | en | Father of the Bride Part II | Just when George thinks has seen it all, he is ... from his ... | ... | 1995-02-10 | 76578911.0 | 106.0 | [{'iso_639_1': 'e 'name': 'Englis |

5 rows × 24 columns

*Output the type of df*

```
type(df)
```

```
pandas.core.frame.DataFrame
```

*The shape of df*

```
df.shape
```

```
(45466, 24)
```

*The columns of df*

```
df.columns
```

```
Index(['adult', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id',
       'imdb_id', 'original_language', 'original_title', 'overview',
       'popularity', 'poster_path', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'video',
       'vote_average', 'vote_count'],
      dtype='object')
```

*Select the second movie in df*

```
second = df.iloc[1]
second
```

```
adult                                                               False
belongs_to_collection                                               Nah
budget                                                              65000000
genres                                  [{'id': 12, 'name': 'Adventure'}, {'id': 14, ...
homepage                                                            Nah
id                                                                  8844
imdb_id                                                             tt0113497
original_language                                                   en
original_title                                                      Jumanji
overview                                When siblings Judy and Peter discover an enchan...
popularity                                                          17.015916
poster_path                                             /vzmL6fP7aPKNKPRTFnZmiUfciyW.jpg
production_companies                    [{'id': 559, 'name': 'TriStar Pictures'}, {'id': 560, ...
production_countries                    [{'iso_3166_1': 'US', 'name': 'United States o...
release_date                                                        1995-12-15
revenue                                                             262797249.0
runtime                                                             104.0
spoken_languages                        [{'iso_639_1': 'en', 'name': 'English'}, {'iso...
status                                                              Released
tagline                                 Roll the dice and unleash the excitement!
title                                                               Jumanji
video                                                               False
vote_average                                                        6.9
vote_count                                                          2413
Name: 1, dtype: object
```

*Change the index to the title*

```
df = df.set_index('title')
```

*Access the movie with title 'Jumanji'*

```
jum = df.loc['Jumanji']
jum
```

```
adult                                                               False
belongs_to_collection                                               Nah
budget                                                              65000000
genres                                  [{'id': 12, 'name': 'Adventure'}, {'id': 14, ...
homepage                                                            Nah
id                                                                  8844
imdb_id                                                             tt0113497
original_language                                                   en
original_title                                                      Jumanji
overview                                When siblings Judy and Peter discover an enchan...
popularity                                                          17.015916
poster_path                                             /vzmL6fP7aPKNKPRTFnZmiUfciyW.jpg
production_companies                    [{'name': 'TriStar Pictures', 'id': 559}, {'na...
production_countries                    [{'iso_3166_1': 'US', 'name': 'United States o...
release_date                                                        1995-12-15
revenue                                                             262797249.0
runtime                                                             104.0
spoken_languages                        [{'iso_639_1': 'en', 'name': 'English'}, {'iso...
status                                                              Released
tagline                                 Roll the dice and unleash the excitement!
title                                                               Jumanji
video                                                               False
vote_average                                                        6.9
vote_count                                                          2413
Name: Jumanji, dtype: object
```

*Revert back to the previous zero-based indexing*

```
df = df.reset_index()
```

*Create a smaller dataframe with a subset of all features*

```
small_df = df[['title', 'release_date', 'budget', 'revenue', 'runtime', 'genres']]
small_df.head(5)
```

| | title | release_date | budget | revenue | runtime | genres |
|---|---|---|---|---|---|---|
| 0 | Toy Story | 1995-10-30 | 30000000 | 373554033.0 | 81.0 | [{'id': 16, 'name': 'Animation'}, {'id': 35, ... |
| 1 | Jumanji | 1995-12-15 | 65000000 | 262797249.0 | 104.0 | [{'id': 12, 'name': 'Adventure'}, {'id': 14, ... |
| 2 | Grumpier Old Men | 1995-12-22 | 0 | 0.0 | 101.0 | [{'id': 10749, 'name': 'Romance'}, {'id': 35, ... |
| 3 | Waiting to Exhale | 1995-12-22 | 16000000 | 81452156.0 | 127.0 | [{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam... |
| 4 | Father of the Bride Part II | 1995-02-10 | 0 | 76578911.0 | 106.0 | [{'id': 35, 'name': 'Comedy'}] |
| 5 | Heat | 1995-12-15 | 60000000 | 187436818.0 | 170.0 | [{'id': 28, 'name': 'Action'}, {'id': 80, 'nam... |
| 6 | Sabrina | 1995-12-15 | 58000000 | 0.0 | 127.0 | [{'id': 35, 'name': 'Comedy'}, {'id': 10749, ... |
| 7 | Tom and Huck | 1995-12-22 | 30000000 | 0.0 | 97.0 | [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam... |
| 8 | Sudden Death | 1995-12-22 | 35000000 | 64350171.0 | 106.0 | [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam... |
| 9 | GoldenEye | 1995-11-16 | 58000000 | 352194034.0 | 130.0 | [{'id': 12, 'name': 'Adventure'}, {'id': 28, ... |
| 10 | The American President | 1995-11-17 | 62000000 | 107879496.0 | 106.0 | [{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam... |
| 11 | Dracula: Dead and Loving It | 1995-12-22 | 0 | 0.0 | 90.0 | [{'id': 35, 'name': 'Comedy'}, {'id': 27, 'nam... |
| 12 | Balto | 1995-12-22 | 0 | 11863264.0 | 78.0 | [{'id': 10751, 'name': 'Family'}, {'id': 16, ... |
| 13 | Nixon | 1995-12-22 | 44000000 | 13967615.0 | 192.0 | [{'id': 36, 'name': 'History'}, {'id': 18, 'na... |
| 14 | Cutthroat Island | 1995-12-22 | 98000000 | 10017322.0 | 119.0 | [{'id': 28, 'name': 'Action'}, {'id': 12, 'nam... |

*Get information of the data types of each feature*

```
small_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   title         45460 non-null  object
 1   release_date  45379 non-null  object
 2   budget        45466 non-null  object
 3   revenue       45460 non-null  float64
 4   runtime       45203 non-null  float64
 5   genres        45466 non-null  object
dtypes: float64(2), object(4)
memory usage: 2.1+ MB
```

*Convert budget to float*

```
df['budget'] = df['budget'].astype('float')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File C:\Users\juand\Anaconda3\Lib\site-packages\pandas\core\generic.py:6324
-> ValueError: could not convert string to float: '/ffRaCqUlaoOk61GZHOwetZ8s4.jpg'

During handling of the above exception, another exception occurred:

ValueEx
ceptionTraceback (most recent call last)
File C:\Users\juand\Anaconda3\Lib\site-packages\pandas\core\generic.py in astype(self, dtype, copy, errors)
   5912         else:
   5913             # else, only a single dtype is given
-> 5914             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors,)
   5915             return self._constructor(new_data).__finalize__(self, method="astype")
   5916

File C:\Users\juand\Anaconda3\Lib\site-packages\pandas\core\internals\managers.py in astype(self, dtype, copy, errors)
    594     def astype(self, dtype, copy: bool = False, errors: str = "raise"
    595         return self.apply("astype", dtype=dtype, copy=copy, errors=errors)
    596

File C:\Users\juand\Anaconda3\Lib\site-packages\pandas\core\internals\managers.py in apply(self, f, align_keys, **kwargs)
    404         applied = b.apply(f, **kwargs)
    405         else:
    406             applied = getattr(b, f)(**kwargs)
    407         result_blocks = _extend_blocks(applied, result_blocks)
    408

File C:\Users\juand\Anaconda3\Lib\site-packages\pandas\core\internals\blocks.py in astype(self, dtype, copy, errors)
    593         values = values.reshape(
    594             shape
    595             )
    596         values = astype_nansafe(values.ravel(), dtype, copy=True)
    597         values = getattr(values, "_data", values)
    598         raise ValueError(msg, TypeError)
    599     # GH 10265
    600     return self.maybe_coerce_values(values)
    601
    602     def convert(self, copy: bool = True, errors: str = "raise"):
    603         """
    604         Attempt to coerce any object types to better types.
    605         Return a copy of the block (if copy = True).
    606         """
    607         return arr.view(dtype)

ValueError: could not convert string to float: '/ffRaCqUlaoOk61GZHOwetZ8s4.jpg'
```

*Function to convert to float manually*

```
def to_float(x):
    try:
        x = float(x)
    except:
        x = np.nan
    return x

#Apply the to_float function to all values in the budget column
small_df['budget'] = small_df['budget'].apply(to_float)

#Try converting to float using pandas astype
small_df['budget'] = small_df['budget'].astype('float')
```

```
<ipython-input-23-f914b9dabf8e>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  small_df['budget'] = small_df['budget'].apply(to_float)
<ipython-input-23-f914b9dabf8e>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  small_df['budget'] = small_df['budget'].astype('float')
```

*Get the data types for all features*

```
small_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   title         45460 non-null  object
 1   release_date  45379 non-null  object
 2   budget        45463 non-null  float64
 3   revenue       45460 non-null  float64
 4   runtime       45203 non-null  float64
 5   genres        45466 non-null  object
dtypes: float64(3), object(3)
memory usage: 2.1+ MB
```

*Convert release_date into pandas datetime format*

```
small_df['release_date'] = pd.to_datetime(small_df['release_date'], errors='coerce')

#Extract year from the datetime
small_df['year'] = small_df['release_date'].apply(lambda x: str(x).split('-')[0]
if x != np.nan else np.nan)

#Divide the DataFrame with the new 'year' feature
small_df.head()
```

```
<ipython-input-25-f91e9bdaaf8e>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  small_df['release_date'] = pd.to_datetime(small_df['release_date'], errors='coerce')
<ipython-input-25-f91e9bdaaf8e>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  small_df['year'] = small_df['release_date'].apply(lambda x: str(x).split('-')[0]
```

| | title | release_date | budget | revenue | runtime | genres | year |
|---|---|---|---|---|---|---|---|
| 0 | Toy Story | 1995-10-30 | 30000000.0 | 373554033.0 | 81.0 | [{'id': 16, 'name': 'Animation'}, {'id': 35, ... | 1995 |
| 1 | Jumanji | 1995-12-15 | 65000000.0 | 262797249.0 | 104.0 | [{'id': 12, 'name': 'Adventure'}, {'id': 14, ... | 1995 |
| 2 | Grumpier Old Men | 1995-12-22 | 0.0 | 0.0 | 101.0 | [{'id': 10749, 'name': 'Romance'}, {'id': 35, ... | 1995 |
| 3 | Waiting to Exhale | 1995-12-22 | 16000000.0 | 81452156.0 | 127.0 | [{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam... | 1995 |
| 4 | Father of the Bride Part II | 1995-02-10 | 0.0 | 76578911.0 | 106.0 | [{'id': 35, 'name': 'Comedy'}] | 1995 |

*Select Top 10 Highest Budget Movies*

```
small_df[['title', 'budget', 'year']].sort_values('budget', ascending=False).head(10)
```

| | title | budget | year |
|---|---|---|---|
| 14531 | Avatar | 2009-12-10 | 237000000.0 | 2009 |
| 26555 | Star Wars: The Force Awakens | 2015-12-15 | 245000000.0 | 2015 |
| 1639 | Titanic | 1997-11-18 | 200000000.0 | 1997 |
| 17818 | The Avengers | 2012-04-25 | 220000000.0 | 2012 |
| 29566 | Jurassic World | 2015-06-09 | 150000000.0 | 2015 |
| 23110 | Frozen | 2013-11-27 | 150000000.0 | 2013 |
| 40202 | Beauty and the Beast | 2017-03-16 | 160000000.0 | 2017 |
| 42255 | The Fate of the Furious | 2017-04-12 | 250000000.0 | 2017 |
| 29830 | Iron Man 3 | 2013-04-18 | 200000000.0 | 2013 |
| 30700 | Minions | 2015-06-17 | 74000000.0 | 2015 |
| 33778 | Captain America: Civil War | 2016-04-27 | 250000000.0 | 2016 |
| 17563 | Transformers: Dark of the Moon | 2011-06-28 | 195000000.0 | 2011 |
| 7000 | The Lord of the Rings: The Return of the King | 2003-12-01 | 94000000.0 | 2003 |
| 3131 | Skyfall | 2012-10-25 | 200000000.0 | 2012 |
| 23617 | Transformers: Age of Extinction | 2014-06-25 | 210000000.0 | 2014 |
| 18252 | The Dark Knight Rises | 2012-07-16 | 250000000.0 | 2012 |
| 16243 | Toy Story 3 | 2010-06-16 | 200000000.0 | 2010 |
| 11006 | Pirates of the Caribbean: Dead Man's Chest | 2006-06-20 | 225000000.0 | 2006 |
| 41489 | Rogue One: A Star Wars Story | 2016-12-14 | 200000000.0 | 2016 |
| 17245 | Pirates of the Caribbean: On Stranger Tides | 2011-05-14 | 380000000.0 | 2011 |
| 38176 | Finding Dory | 2016-06-16 | 200000000.0 | 2016 |
| 14862 | Alice in Wonderland | 2010-03-03 | 200000000.0 | 2010 |
| 26975 | Zootopia | 2016-02-11 | 150000000.0 | 2016 |
| 19971 | The Hobbit: An Unexpected Journey | 2012-11-26 | 200000000.0 | 2012 |
| 44809 | Despicable Me 3 | 2017-06-15 | 80000000.0 | 2017 |
| 12481 | The Dark Knight | 2008-07-18 | 185000000.0 | 2008 |

*Select only those movies which earned more than 1 billion and spent less than 150 million*

```
new2 = small_df[(small_df['revenue'] > 1e9) & (small_df['budget'] < 1.5e8)]
new2
```

| | title | release_date | budget | revenue | runtime | genres | year |
|---|---|---|---|---|---|---|---|
| 17437 | Harry Potter and the Deathly Hallows: Part 2 | 2011-07-07 | 125000000.0 | 1342000.00 | 130.0 | [{'id': 10751, 'name': 'Family'}, {'id': 14, ... | 2011 |
| 30700 | Minions | 2015-06-17 | 74000000.0 | 1156730962.0 | 91.0 | [{'id': 10751, 'name': 'Family'}, {'id': 16, ... | 2015 |
| 7000 | The Lord of the Rings: The Return of the King | 2003-12-01 | 94000000.0 | 1118888979.0 | 201.0 | [{'id': 12, 'name': 'Adventure'}, {'id': 14, ... | 2003 |
| 44809 | Despicable Me 3 | 2017-06-15 | 80000000.0 | 1034799409.0 | 96.0 | [{'id': 28, 'name': 'Action'}, {'id': 16, 'nam... | 2017 |

*Let's check out the shortest- and longest-running movies of all time*

```
runtime = small_df['runtime']

#Print the longest runtime of any movie
print(runtime.max())

#Print the shortest runtime of any movie
print(runtime.min())
```

```
1256.0
0.0
```

*To calculate the mean and median*

```
#Get the budget Series object
budget = small_df['budget']

#Print the mean budget of the movies
print(budget.mean())

#Print the median budget of the movies
print(budget.median())
```

```
42207.0.01847603
0.0
```

```
#The average budget of a movie is $4.2 million and the median budget is $0
#This suggests that at least half the movies in our dataset have no budget at all
```

*What is the revenue that the 90th percentile movie generated?*

```
#Get the revenue Series object
revenue = small_df['revenue']

#Revenue generated by the 90th percentile movie
revenue.quantile(0.90)
```

```
82676168.39999982
```

*Let's find out the number of movies released each year.*

```
#Get number of movies released each year
small_df['year'].value_counts()
```

```
2014    1574
2015    1505
2013    1499
2016    1172
2012    1147
        ...
1893    1
1894    1
1895    1
1896    1
1902    1
Name: year, Length: 136, dtype: int64
```