DMPM Assignment 3

Name: Rushikesh Jyoti

Division: A

Roll no: 27

SRN: 201901139

Question: Preprocess and clean the given dataset

Code

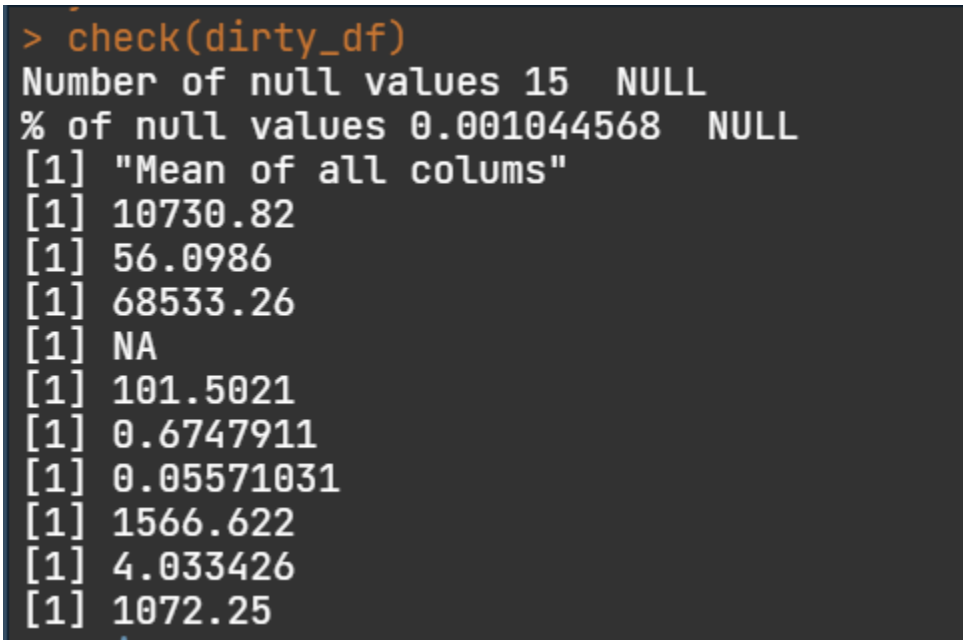
```
# install.packages("tidyverse")
# install.packages("Hmisc")

library(tidyverse)
library(dplyr)
library(Hmisc)

df = read.csv('ToyotaCorolla.csv')
dirty_df = read.csv('ToyotaCorolla - Dirty.csv')



check = function(dataset) {
  print(cat("Number of null values", sum(is.na(dataset)), " "))
  print(cat("% of null values", mean(is.na(dataset)), " "))

  print("Mean of all colums")
  for (i in 1:ncol(dataset)) {
    print(mean(dataset[,i], na.rm = TRUE))
```

```
  }
}

check(dirty_df)

head(rename(dirty_df, Kilometers = KM))

clean_df = na.omit(dirty_df)

head(select(clean_df, -MetColor))

head(arrange(clean_df, Age))

slice(clean_df, 4:17)

head(filter(clean_df, FuelType == 'Petrol'))

glimpse(clean_df)

boxplot(clean_df$Price)

boxplot(clean_df$Age)

boxplot(clean_df$Weight)

print("Outliers of Weight are ")
boxplot.stats(clean_df$Weight)$out
```

```
# Numerical Imputation

dirty_df$Age = impute(dirty_df$Age, fun=mean)


dirty_df$CC = impute(dirty_df$CC, fun=mean)


dirty_df$Weight = impute(dirty_df$Weight, fun=mean)


for (i in 1:ncol(dirty_df)) {
   print(sum(is.na(dirty_df[,i])))
}


print("Phew! No null values anymore!")
```

Output

Null Values of dataset and mean of every column



Fourth column is categorical data so it can’t be `meaned`

Renaming a column

```
> head(rename(dirty_df, Kilometers = KM))
  Price Age Kilometers FuelType HP MetColor Automatic   CC Doors Weight
1 13500  23      46986   Diesel 90        1         0 2000     3   1165
2 13750  23      72937   Diesel 90        1         0 2000     3   1165
3 13950  NA      41711   Diesel 90        1         0 2000     3   1165
4 14950  26      48000   Diesel 90        0         0 2000     3   1165
5 13750  30      38500           90        0         0 2000     3   1170
6 12950  32      61000   Diesel 90        0         0 2000     3   1170
```

Omitting the NA values

```
> clean_df = na.omit(dirty_df)
> sum(is.na(clean_df))
[1] 0
```

Removing a column (MetColor) from dataset

```
> head(select(clean_df, -MetColor))
  Price Age    KM FuelType HP Automatic   CC Doors Weight
1 13500  23 46986   Diesel 90         0 2000     3   1165
2 13750  23 72937   Diesel 90         0 2000     3   1165
4 14950  26 48000   Diesel 90         0 2000     3   1165
5 13750  30 38500           90         0 2000     3   1170
6 12950  32 61000   Diesel 90         0 2000     3   1170
7 16900  27 94612   Diesel 90         0 2000     3   1245
```

Taking a slice of dataset

```
> slice(clean_df, 4:17)
   Price Age    KM FuelType  HP MetColor Automatic   CC Doors Weight
1  13750  30 38500           90        0         0 2000     3   1170
2  12950  32 61000   Diesel  90        0         0 2000     3   1170
3  16900  27 94612   Diesel  90        1         0 2000     3   1245
4  18600  30 75889   Diesel  90        1         0 2000     3   1245
5  21500  27 19700   Petrol 192        0         0 1800     3   1185
6  20950  25 31461   Petrol 192        0         0 1800     3   1185
7  19950  22 43610   Petrol 192        0         0 1800     3   1185
8  19600  25 32189   Petrol 192        0         0 1800     3   1185
9  21500  31 23000   Petrol 192        1         0 1800     3   1185
10 22500  32 34131   Petrol 192        1         0 1800     3   1185
11 22000  28 18739   Petrol 192        0         0 1800     3   1185
12 22750  30 34000           192        1         0 1800     3   1185
13 17950  24 21716   Petrol 110        1         0 1600     3   1105
14 16750  24 25563   Petrol 110        0         0 1600     3   1065
```

Filtering the dataset to get all petrol vehicles

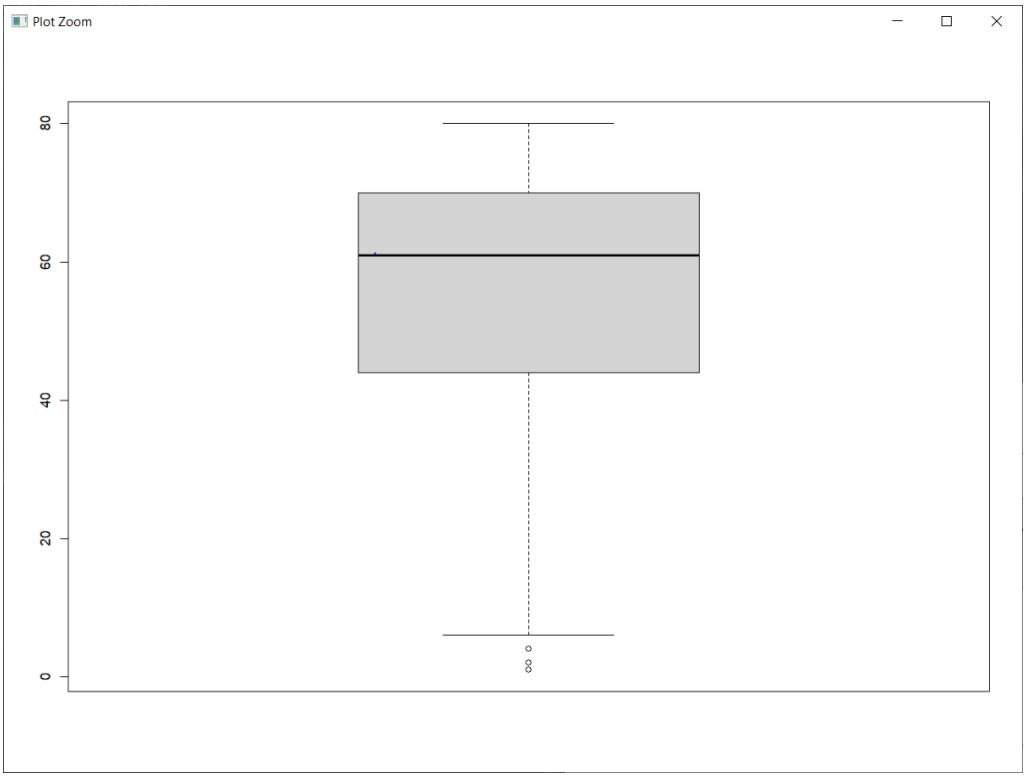
```
> head(filter(clean_df, FuelType == 'Petrol'))
  Price Age    KM FuelType  HP MetColor Automatic   CC Doors Weight
1 21500  27 19700   Petrol 192        0         0 1800     3   1185
2 20950  25 31461   Petrol 192        0         0 1800     3   1185
3 19950  22 43610   Petrol 192        0         0 1800     3   1185
4 19600  25 32189   Petrol 192        0         0 1800     3   1185
5 21500  31 23000   Petrol 192        1         0 1800     3   1185
6 22500  32 34131   Petrol 192        1         0 1800     3   1185
```

Taking a glimpse of our dataset

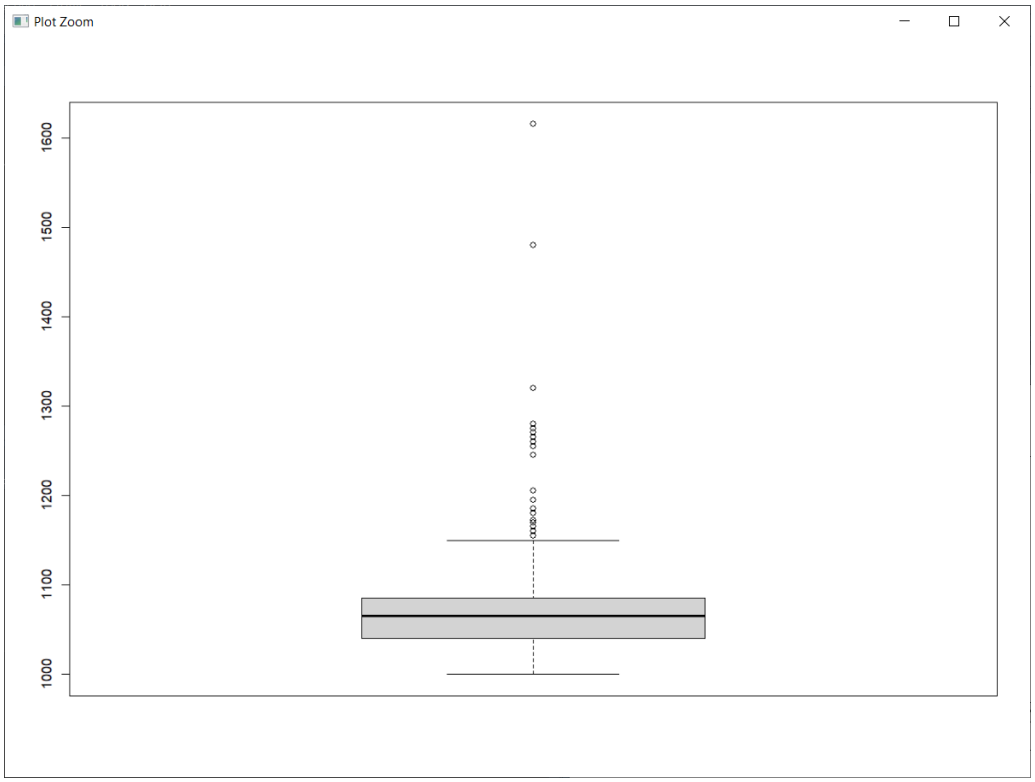
```
> glimpse(clean_df)
Rows: 1,421
Columns: 10
$ Price     <int> 13500, 13750, 14950, 13750, 12950, 16900, 18600, 21500, 20950,~
$ Age       <int> 23, 23, 26, 30, 32, 27, 30, 27, 25, 22, 25, 31, 32, 28, 30, 24~
$ KM        <int> 46986, 72937, 48000, 38500, 61000, 94612, 75889, 19700, 31461,~
$ FuelType  <chr> "Diesel", "Diesel", "Diesel", " ", "Diesel", "Diesel", "Diesel~
$ HP        <int> 90, 90, 90, 90, 90, 90, 90, 192, 192, 192, 192, 192, 192, 192,~
$ MetColor  <int> 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1,~
$ Automatic <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,~
$ CC        <int> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 1800, 1800, 1800, 18~
$ Doors     <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,~
$ Weight    <int> 1165, 1165, 1165, 1170, 1170, 1245, 1245, 1185, 1185, 1185, 11~
```

Boxplot of Price of vehicles

Boxplot of Age of Vehicles



Boxplot of Weight of vehicles



Getting all outliers of weight

```
> boxplot.stats(clean_df$Weight)$out
 [1] 1165 1165 1165 1170 1170 1245 1245 1185 1185 1185 1185 1185 1185 1185 1185
[16] 1170 1255 1255 1270 1255 1195 1255 1180 1195 1165 1180 1275 1180 1180 1245
[31] 1265 1260 1260 1155 1480 1480 1480 1320 1320 1280 1270 1255 1275 1320 1185
[46] 1165 1180 1160 1205 1205 1205 1170 1615 1165 1205 1165 1260 1260 1155 1480
[61] 1172
```

Impute the columns with NA values

```
# Numerical Imputation
dirty_df$Age = impute(dirty_df$Age, fun=mean)

dirty_df$CC = impute(dirty_df$CC, fun=mean)

dirty_df$Weight = impute(dirty_df$Weight, fun=mean)
```

```
> for (i in 1:ncol(dirty_df)) {
+     print(sum(is.na(dirty_df[,i])))
+ }
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
> print("Phew! No null values anymore!")
[1] "Phew! No null values anymore!"
```

The dataset is now clean

Python Notebook for cleaning and analysis of Movie Dataset

Name: Rushikesh Jyoti

Division: A

Roll no: 27

SRN: 201901139

In [81]:

```
import pandas as pd
import seaborn as sb
from matplotlib import pyplot as plot
```

In [82]:

```
df = pd.read_csv('C:\VS_Workshop\Sem 6\Data Mining and Predictive Modelling\Assignments\Ass3\MovieAssignmentData.csv')

df.head()
```

Out[82]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	acto
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	

5 rows × 28 columns

Does the dataset contain any NA values?

In [83]:

```
df.isna().sum()
```

Out[83]:

```
color                        19
director_name               104
num_critic_for_reviews       50
duration                     15
director_facebook_likes     104
actor_3_facebook_likes       23
actor_2_name                 13
actor_1_facebook_likes        7
gross                       884
genres                        0
actor_1_name                  7
movie_title                   0
num_voted_users               0
cast_total_facebook_likes     0
```

```
actor_3_name                23
facenumber_in_poster        13
plot_keywords              153
movie_imdb_link              0
num_user_for_reviews        21
language                    12
country                      5
content_rating             303
budget                     492
title_year                 108
actor_2_facebook_likes      13
imdb_score                   0
aspect_ratio               329
movie_facebook_likes         0
dtype: int64
```

Yup, pretty much every column has null values

Lets remove some columns

In [84]:

```
likes = [col for col in df.columns if 'likes' in col]
likes.extend(['aspect_ratio', 'color', 'facenumber_in_poster', 'movie_imdb_link', 'num_voted_users'])
likes.extend([col for col in df.columns if 'reviews' in col])
df.drop(likes, axis=1, inplace=True)
df.head()
```

Out[84]:

	director_name	duration	actor_2_name	gross	genres	actor_1_name	movie_title	actor_3_nam
0	James Cameron	178.0	Joel David Moore	760505847.0	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	Wes Stu
1	Gore Verbinski	169.0	Orlando Bloom	309404152.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	Jac Davenpo
2	Sam Mendes	148.0	Rory Kinnear	200074175.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	Stephani Sigma
3	Christopher Nolan	164.0	Christian Bale	448130642.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	Josep Gordon-Levi
4	Doug Walker	NaN	Rob Walker	NaN	Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens ...	Na

In [85]:

```
print(df.isna().sum())
```

```
director_name     104
duration           15
actor_2_name       13
gross             884
genres              0
actor_1_name        7
movie_title         0
actor_3_name       23
plot_keywords     153
language           12
country             5
```

```
content_rating     303
budget             492
title_year         108
imdb_score           0
dtype: int64
```

In [86]:

```
df['duration'].fillna(df['duration'].mean(), inplace=True)
df['gross'].fillna(df['gross'].mean(), inplace=True)
df['budget'].fillna(df['budget'].mean(), inplace=True)
df['title_year'].fillna(df['title_year'].mean(), inplace=True)
```

The other columns are categorial data And it doesnt make sense to fill values like `director name` with a mean value so we just omit the NAs

In [87]:

```
df.dropna(inplace=True)
```

In [98]:

```
df.isna().sum()
```

Out[98]:

```
director_name      0
duration           0
actor_2_name       0
gross              0
genres             0
actor_1_name       0
movie_title        0
actor_3_name       0
plot_keywords      0
language           0
country            0
content_rating     0
budget             0
title_year         0
imdb_score         0
dtype: int64
```

YAY!! our dataset is clean now

In [88]:

```
df.columns
```

Out[88]:

```
Index(['director_name', 'duration', 'actor_2_name', 'gross', 'genres',
       'actor_1_name', 'movie_title', 'actor_3_name', 'plot_keywords',
       'language', 'country', 'content_rating', 'budget', 'title_year',
       'imdb_score'],
      dtype='object')
```

In [89]:

```
sb.set(font_scale = 1.8)
```

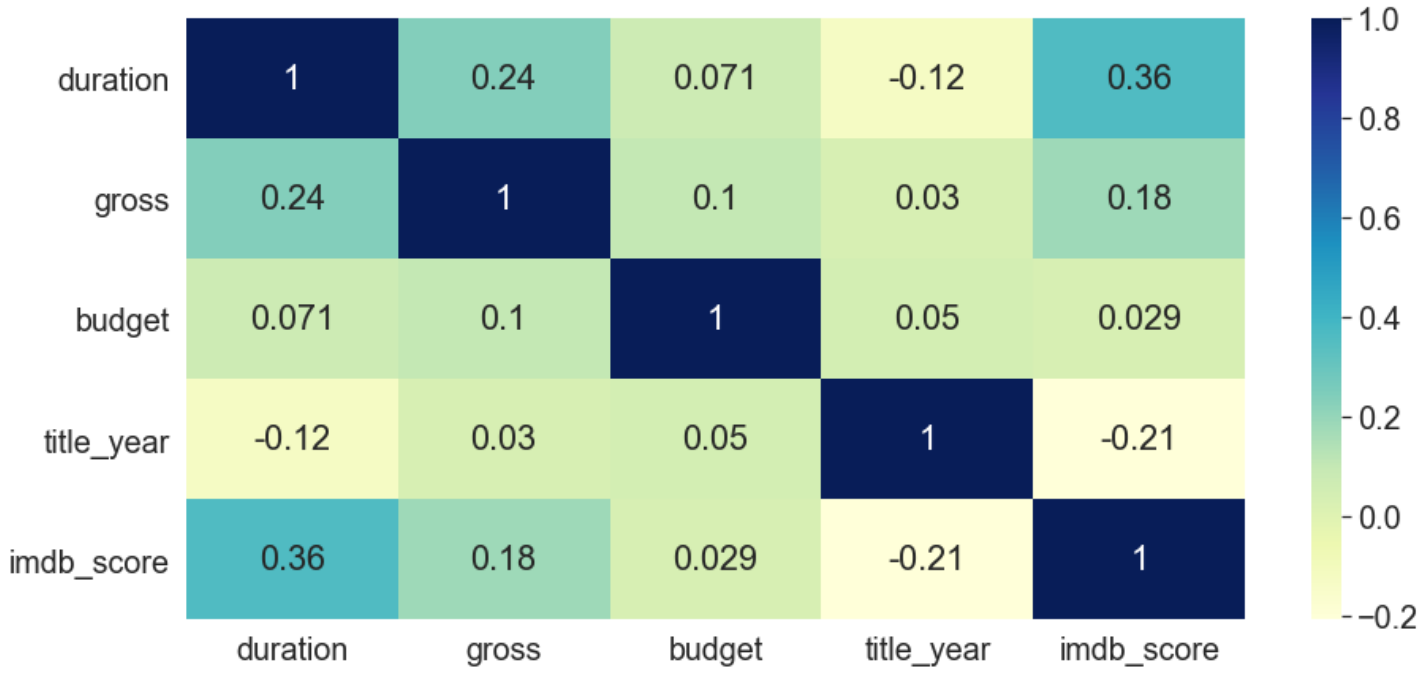
Heatmap of correlation

In [90]:

```
plot.figure(figsize=(15, 7))
sb.heatmap(df.corr(), cmap='YlGnBu', annot=True)
```

Out[90]:

<AxesSubplot:>



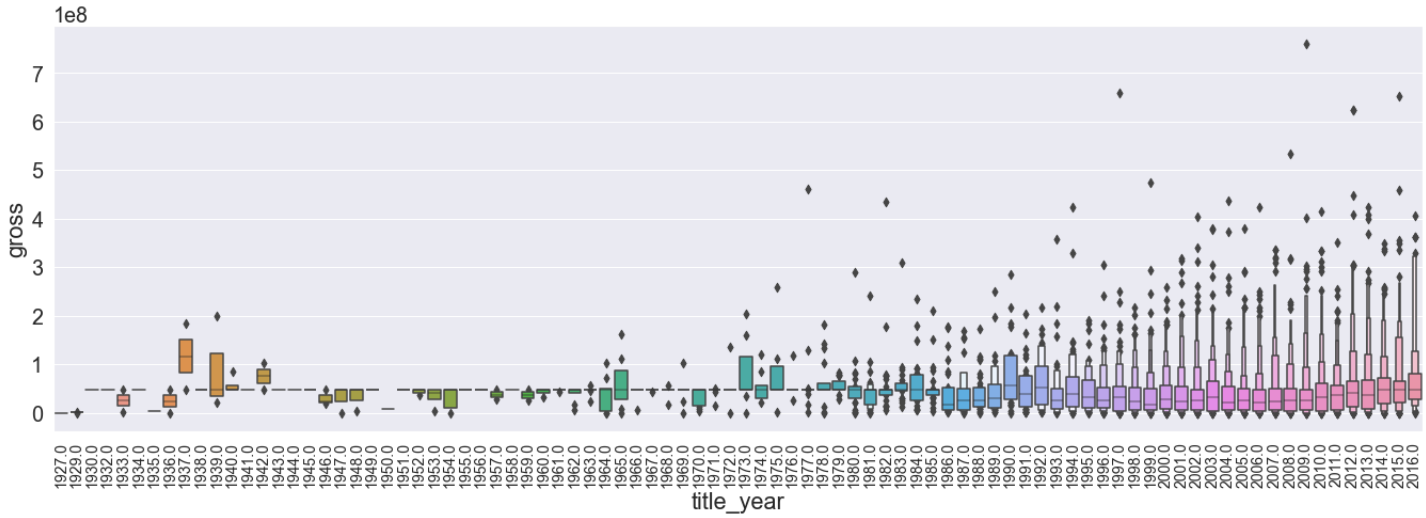
Gross income over the years

In [91]:

```
plot.figure(None, (23, 7))
plot.xticks(rotation=90, fontsize=14)
plot.xlabel("Year")
sb.boxenplot(x='title_year', y='gross', data=df)
```

Out[91]:

<AxesSubplot:xlabel='title_year', ylabel='gross'>



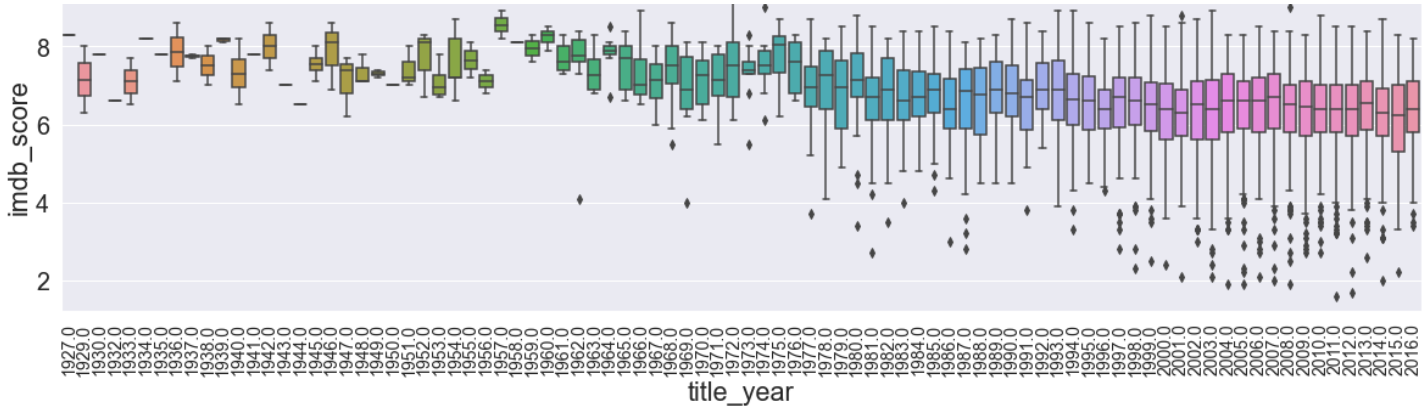
IMDB Scores over the years

In [92]:

```
plot.figure(1, (20, 5))
plot.xticks(rotation=90, fontsize=14)
sb.boxplot(y='imdb_score', x='title_year', data=df)
```

Out[92]:

<AxesSubplot:xlabel='title_year', ylabel='imdb_score'>



In [97]:

```
plot.figure(1, (15, 5))
sb.lineplot(x='gross', y='budget', data=df)
```

Out[97]:

```
<AxesSubplot:xlabel='gross', ylabel='budget'>
```