Assignment – I

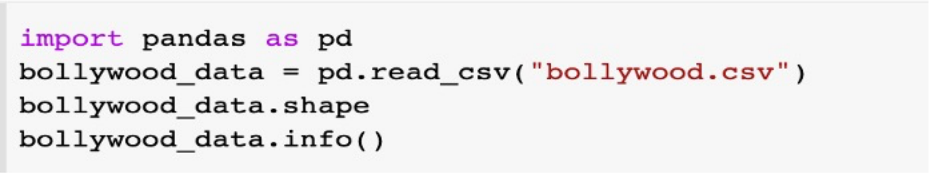
Dataset Description

Use the Bollywood Dataset to answer Questions 1 to 12.
The data file *bollywood.csv* contains box office collection and social media information about movie released in 2013 – 2015 period. Following are the columns and their descriptions.

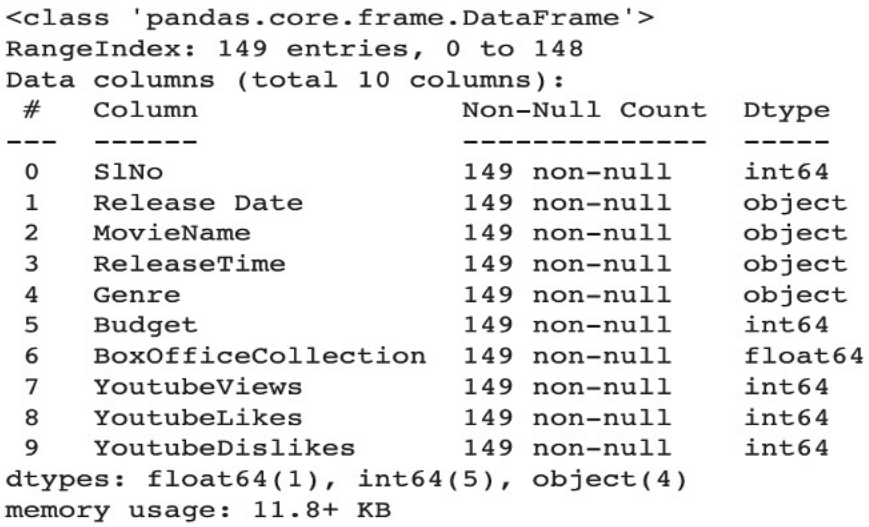
- *SLNo* – Release Date
- *MovieName* – Name of the movie
- *ReleaseTime* – Mentions special time of release. LW (Long Weekend), FS (Festive Seasons), HS (Holiday Season), N (Normal)
- *Genre* – Genre of the film such as Romance, Thriller, Action, Comedy, etc
- *Budget* – Movie creation budget
- *BoxOfficeCollection* - Box Office Collection
- *YoutubeViews* – Number of views of the YouTube trailers
- *YoutubeLikes* – Number of likes of the YouTube trailers
- *YoutubeDislikes* – Number of dislikes of the YouTube trailers

Use Python Code to answer the following questions:

1. How many records are present in the dataset? Print the metadata information of the dataset.

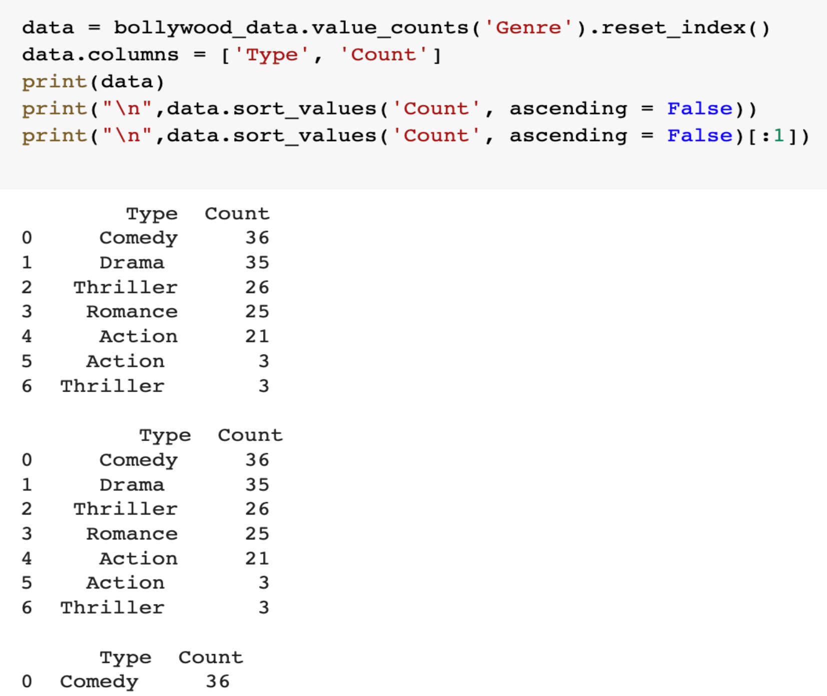


```python
import pandas as pd
bollywood_data = pd.read_csv("bollywood.csv")
bollywood_data.shape
bollywood_data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   SlNo                 149 non-null    int64  
 1   Release Date         149 non-null    object 
 2   MovieName            149 non-null    object 
 3   ReleaseTime          149 non-null    object 
 4   Genre                149 non-null    object 
 5   Budget               149 non-null    int64  
 6   BoxOfficeCollection  149 non-null    float64
 7   YoutubeViews         149 non-null    int64  
 8   YoutubeLikes         149 non-null    int64  
 9   YoutubeDislikes      149 non-null    int64  
dtypes: float64(1), int64(5), object(4)
memory usage: 11.8+ KB
```

2. How many movies got released in each genre? Which genre had highest number of releases? Sort number of releases in each genre in descending order.



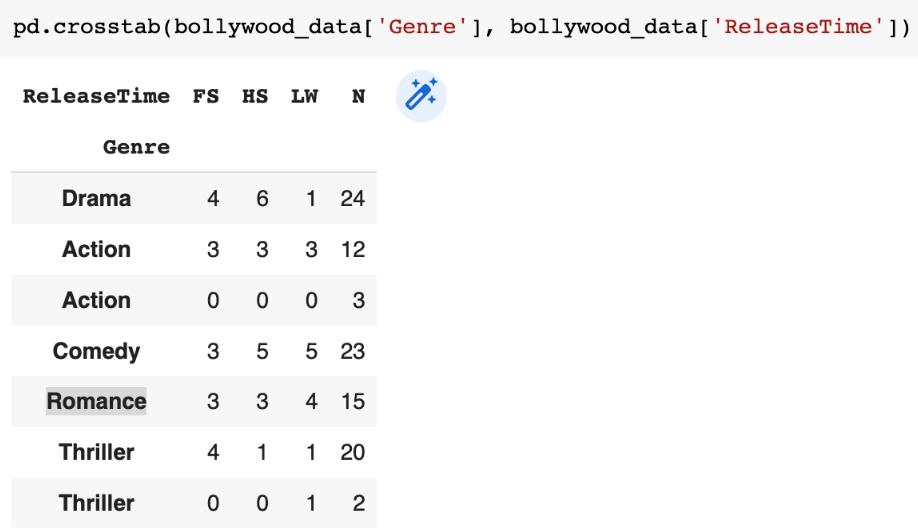
```
data = bollywood_data.value_counts('Genre').reset_index()
data.columns = ['Type', 'Count']
print(data)
print("\n",data.sort_values('Count', ascending = False))
print("\n",data.sort_values('Count', ascending = False)[:1])
```

```
        Type  Count
0     Comedy     36
1      Drama     35
2   Thriller     26
3    Romance     25
4     Action     21
5    Action       3
6  Thriller       3

         Type  Count
0     Comedy     36
1      Drama     35
2   Thriller     26
3    Romance     25
4     Action     21
5    Action       3
6  Thriller       3

      Type  Count
0  Comedy     36
```

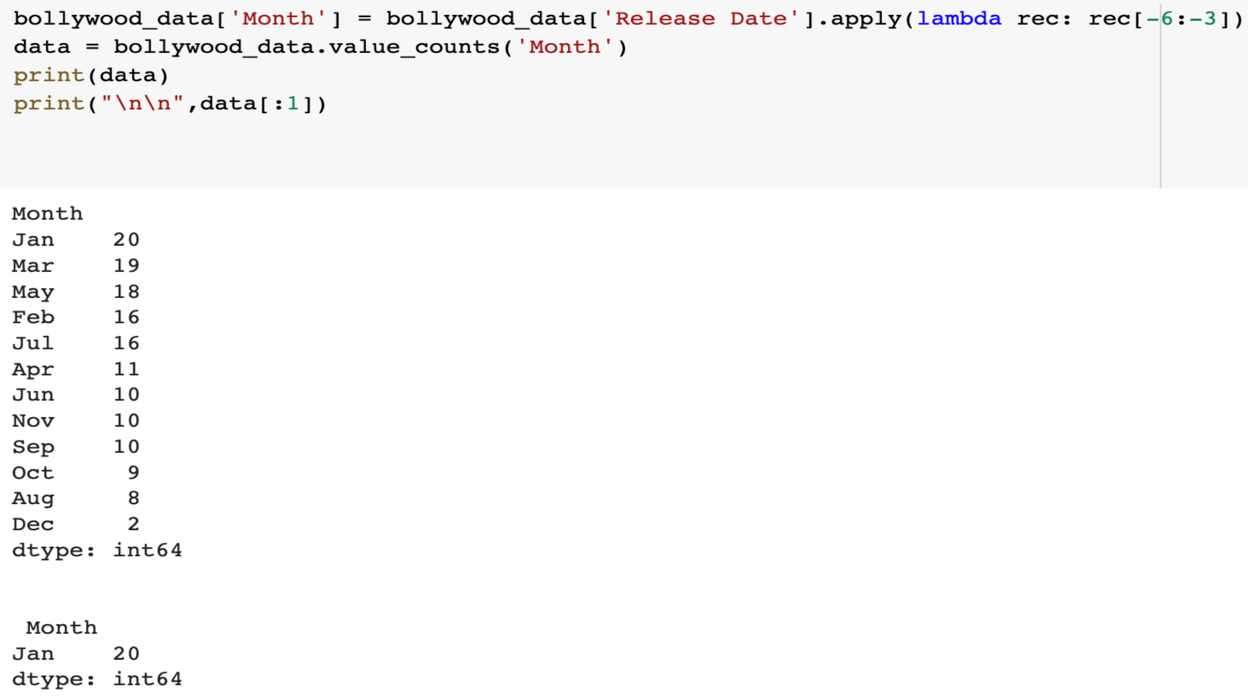
3. How many movies in each genre got released in different release times like long weekend, festive season, etc. (Note: Do a cross tabulation between *Genre* and *ReleaseTime*.)



```
pd.crosstab(bollywood_data['Genre'], bollywood_data['ReleaseTime'])
```

ReleaseTime	FS	HS	LW	N
Genre				
Drama	4	6	1	24
Action	3	3	3	12
Action	0	0	0	3
Comedy	3	5	5	23
Romance	3	3	4	15
Thriller	4	1	1	20
Thriller	0	0	1	2

4. Which month of the year, maximum number movie releases are seen? (Note: Extract a new column called month from *ReleaseDate* column.)

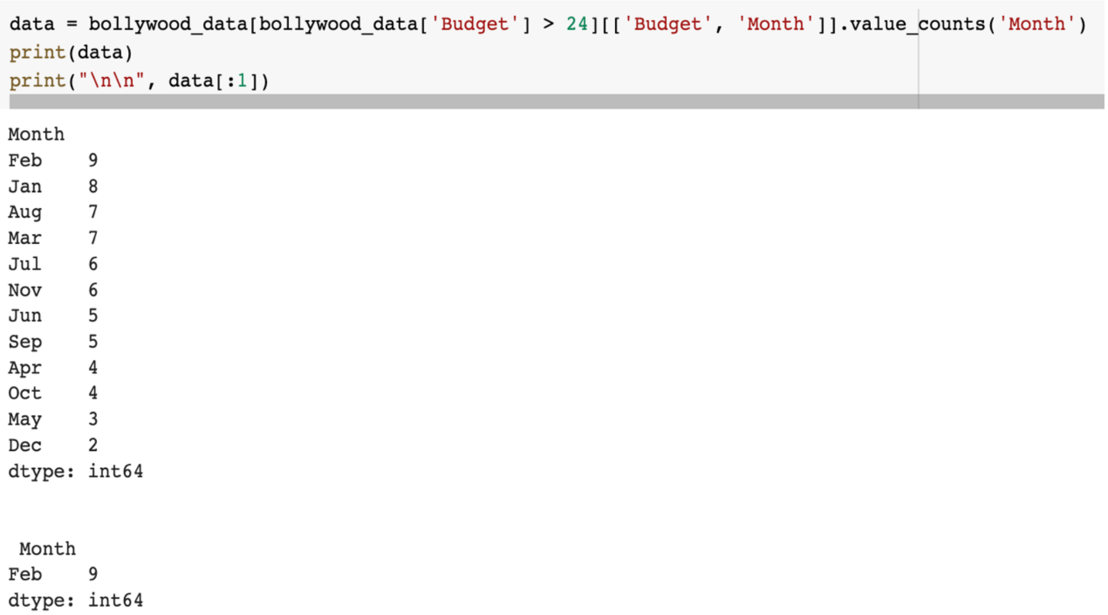


```
bollywood_data['Month'] = bollywood_data['Release Date'].apply(lambda rec: rec[-6:-3])
data = bollywood_data.value_counts('Month')
print(data)
print("\n\n",data[:1])
```

```
Month
Jan    20
Mar    19
May    18
Feb    16
Jul    16
Apr    11
Jun    10
Nov    10
Sep    10
Oct     9
Aug     8
Dec     2
dtype: int64


 Month
Jan    20
dtype: int64
```

5. Which month of the year typically sees most releases of high budgeted movies, that is, movies with budget of 25 crore or more?

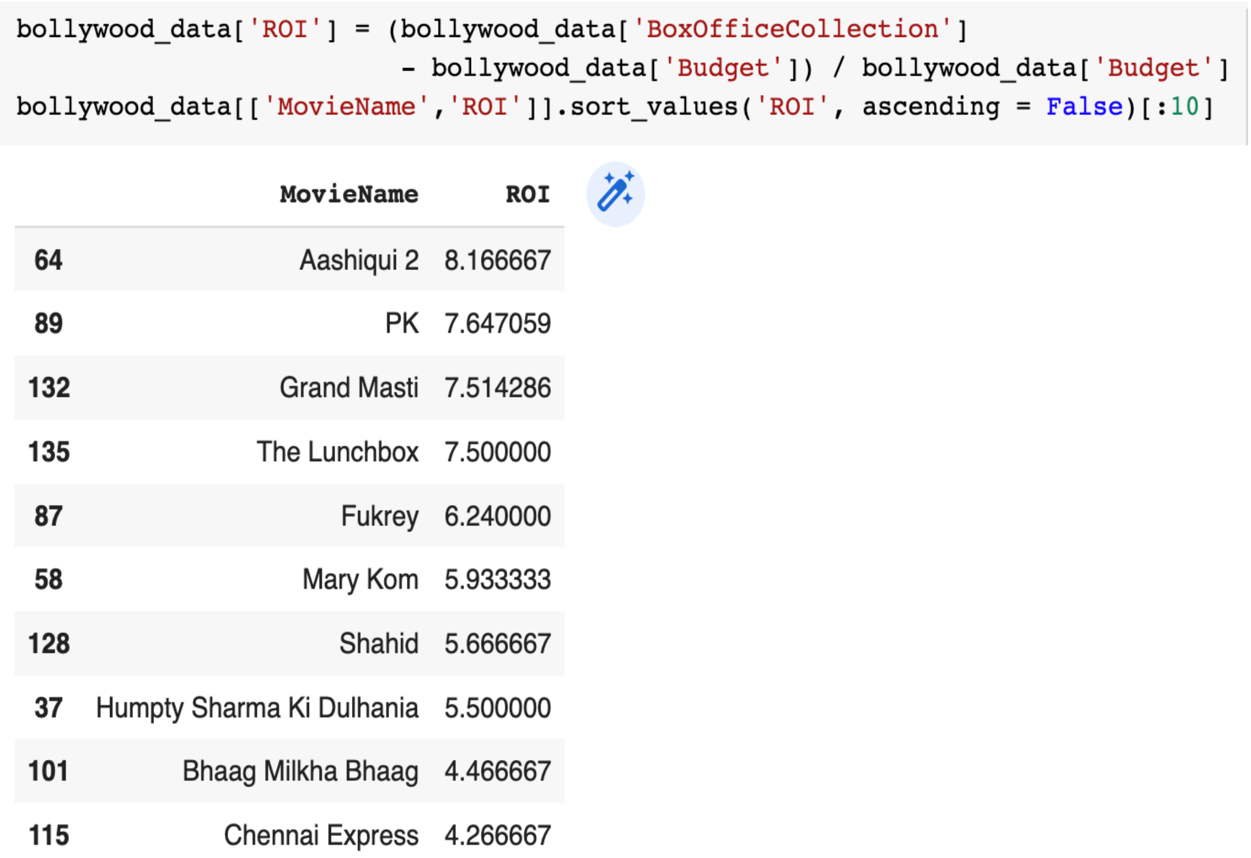


```
data = bollywood_data[bollywood_data['Budget'] > 24][['Budget', 'Month']].value_counts('Month')
print(data)
print("\n\n", data[:1])
```

```
Month
Feb    9
Jan    8
Aug    7
Mar    7
Jul    6
Nov    6
Jun    5
Sep    5
Apr    4
Oct    4
May    3
Dec    2
dtype: int64


 Month
Feb    9
dtype: int64
```

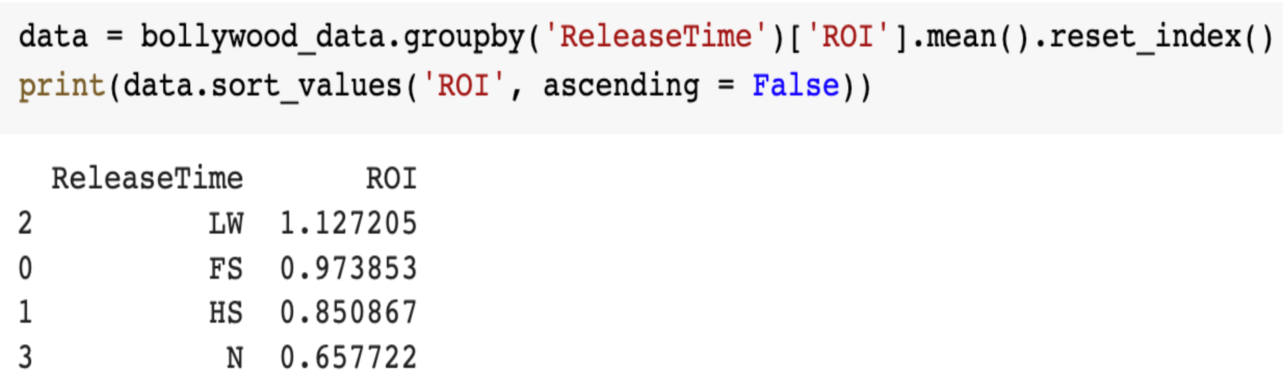
6. Which are the top 10 movies with maximum return on investment (ROI)? Calculate return on investment (ROI) as (*BoxOfficeCollection* – *Budget*) / *Budget*.



```
bollywood_data['ROI'] = (bollywood_data['BoxOfficeCollection']
                         - bollywood_data['Budget']) / bollywood_data['Budget']
bollywood_data[['MovieName','ROI']].sort_values('ROI', ascending = False)[:10]
```

	MovieName	ROI
64	Aashiqui 2	8.166667
89	PK	7.647059
132	Grand Masti	7.514286
135	The Lunchbox	7.500000
87	Fukrey	6.240000
58	Mary Kom	5.933333
128	Shahid	5.666667
37	Humpty Sharma Ki Dulhania	5.500000
101	Bhaag Milkha Bhaag	4.466667
115	Chennai Express	4.266667

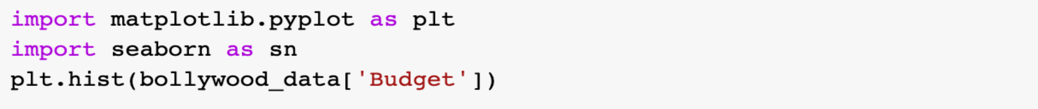
7. Do the movies have higher ROI if they get released on festive seasons or long weekend? Calculate the average ROI for different release times.



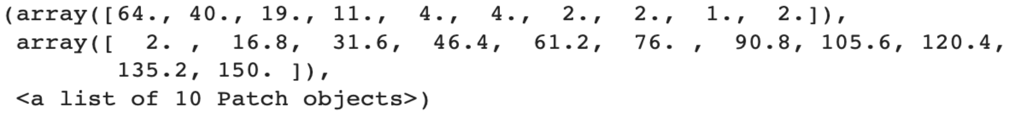
```
data = bollywood_data.groupby('ReleaseTime')['ROI'].mean().reset_index()
print(data.sort_values('ROI', ascending = False))
```

```
  ReleaseTime       ROI
2          LW  1.127205
0          FS  0.973853
1          HS  0.850867
3           N  0.657722
```

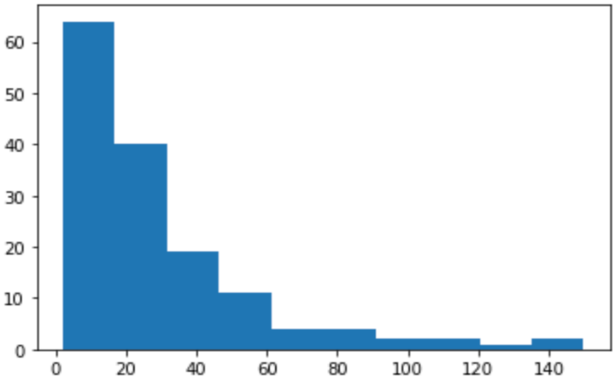
8. Draw a histogram and a distribution plot to find out the distribution of movie budgets. Interpret the plot to conclude if the most movies are high or low budgeted movies.



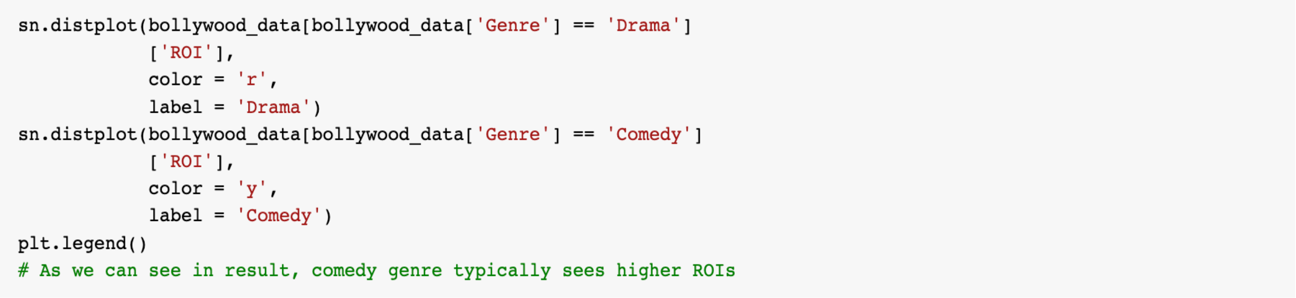
```
import matplotlib.pyplot as plt
import seaborn as sn
plt.hist(bollywood_data['Budget'])
```



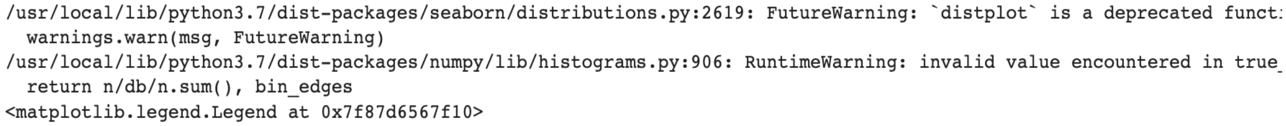
```
(array([64., 40., 19., 11.,  4.,  4.,  2.,  2.,  1.,  2.]),
 array([  2. ,  16.8,  31.6,  46.4,  61.2,  76. ,  90.8, 105.6, 120.4,
        135.2, 150. ]),
 <a list of 10 Patch objects>)
```



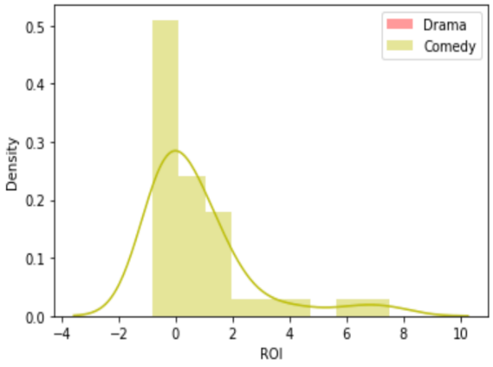
9. Compare the distribution of ROIs between movies with comedy genre and drama. Which genre typically sees higher ROIs?



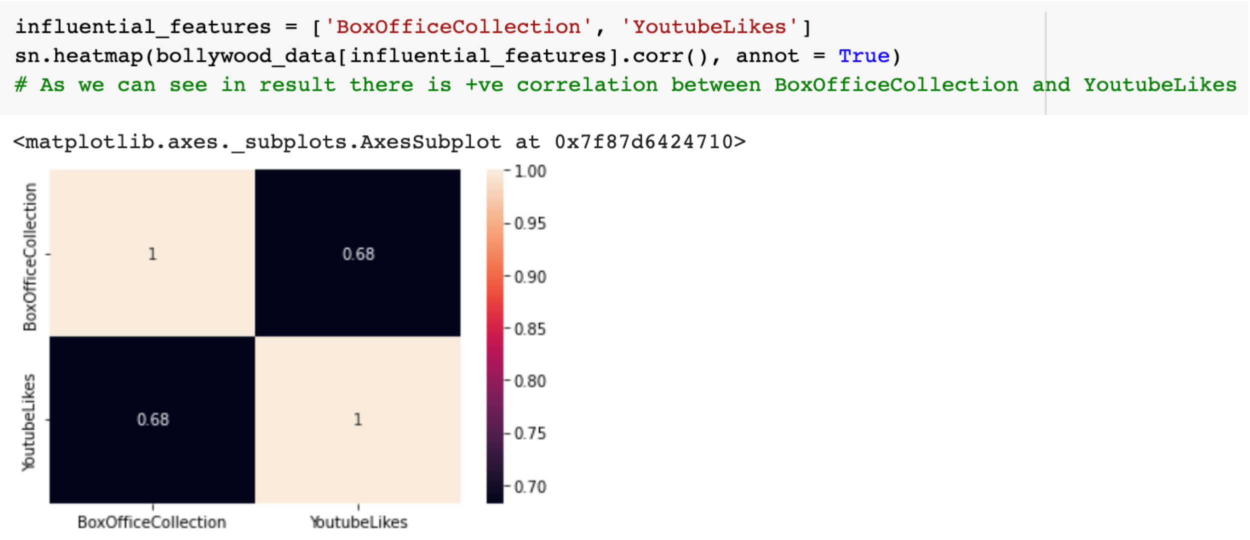
```
sn.distplot(bollywood_data[bollywood_data['Genre'] == 'Drama']
            ['ROI'],
            color = 'r',
            label = 'Drama')
sn.distplot(bollywood_data[bollywood_data['Genre'] == 'Comedy']
            ['ROI'],
            color = 'y',
            label = 'Comedy')
plt.legend()
# As we can see in result, comedy genre typically sees higher ROIs
```



```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated funct:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/numpy/lib/histograms.py:906: RuntimeWarning: invalid value encountered in true_
  return n/db/n.sum(), bin_edges
<matplotlib.legend.Legend at 0x7f87d6567f10>
```



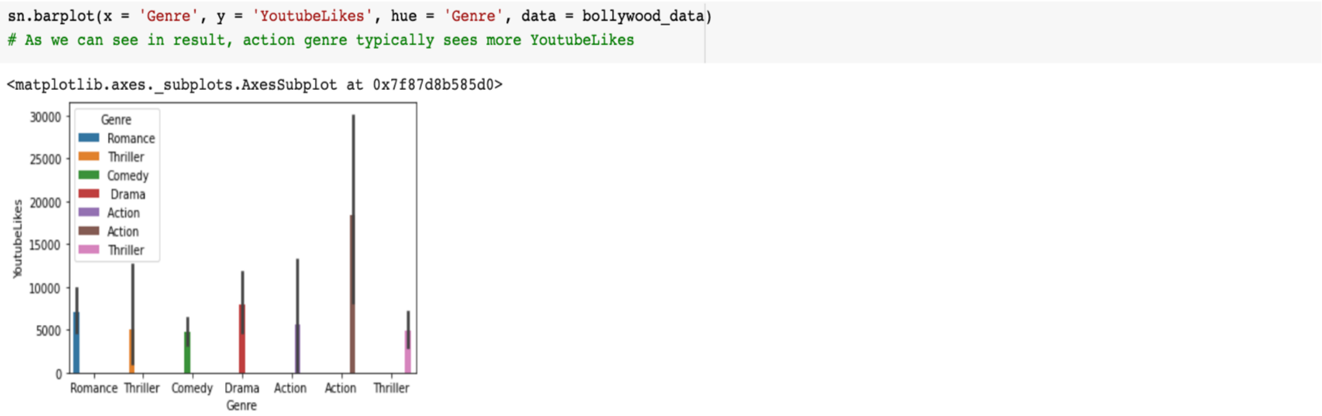
10. Is there a correlation between box office collection and YouTube likes? Is the correlation positive or negative?



```
influential_features = ['BoxOfficeCollection', 'YoutubeLikes']
sn.heatmap(bollywood_data[influential_features].corr(), annot = True)
# As we can see in result there is +ve correlation between BoxOfficeCollection and YoutubeLikes
```

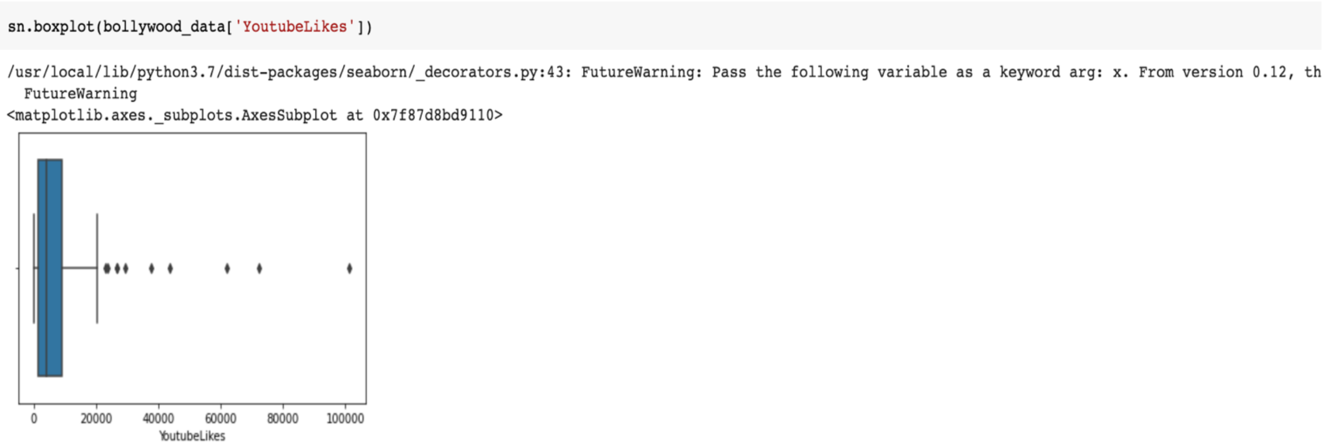
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f87d6424710>
```

11. Which genre of movies typically sees more YouTube likes? Draw boxplots for each genre of movies to compare.



```
sn.barplot(x = 'Genre', y = 'YoutubeLikes', hue = 'Genre', data = bollywood_data)
# As we can see in result, action genre typically sees more YoutubeLikes
```

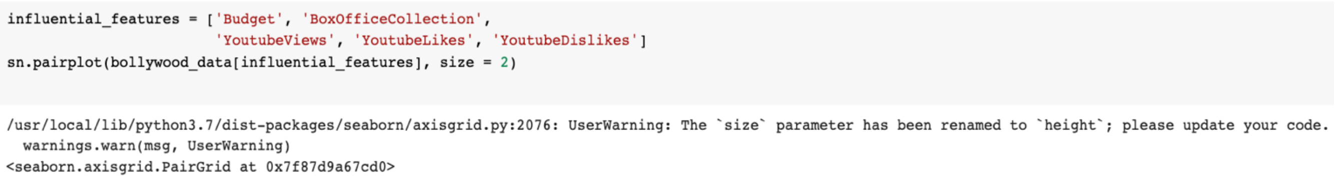
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f87d8b585d0>
```



```
sn.boxplot(bollywood_data['YoutubeLikes'])
```

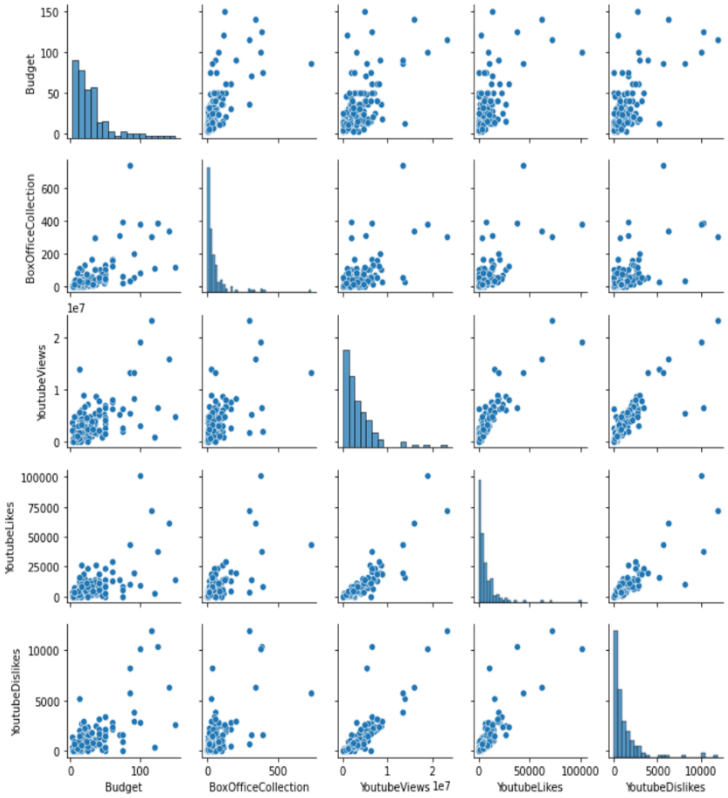
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, th
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f87d8bd9110>
```

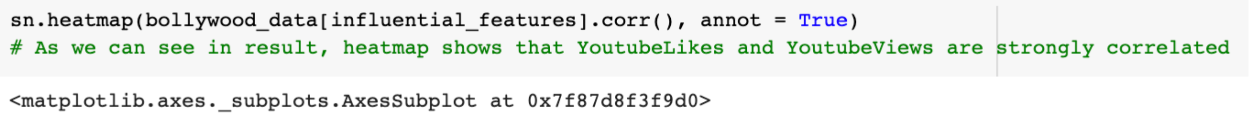
12. Which of the variables among *Budget*, *BoxOfficeCollection*, *YoutubeView*, *YoutubeLikes*, *YoutubeDislikes* are highly correlated? Note: Draw pair plot or heatmap.



```
influential_features = ['Budget', 'BoxOfficeCollection',
                        'YoutubeViews', 'YoutubeLikes', 'YoutubeDislikes']
sn.pairplot(bollywood_data[influential_features], size = 2)
```

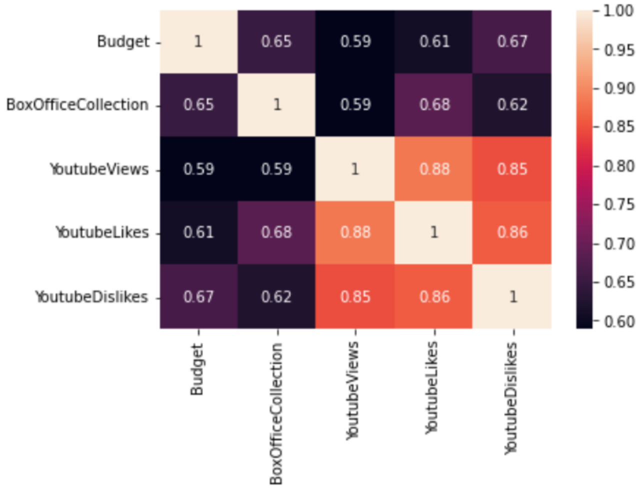
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
<seaborn.axisgrid.PairGrid at 0x7f87d9a67cd0>
```





```
sn.heatmap(bollywood_data[influential_features].corr(), annot = True)
# As we can see in result, heatmap shows that YoutubeLikes and YoutubeViews are strongly correlated
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f87d8f3f9d0>
```



Dataset Description

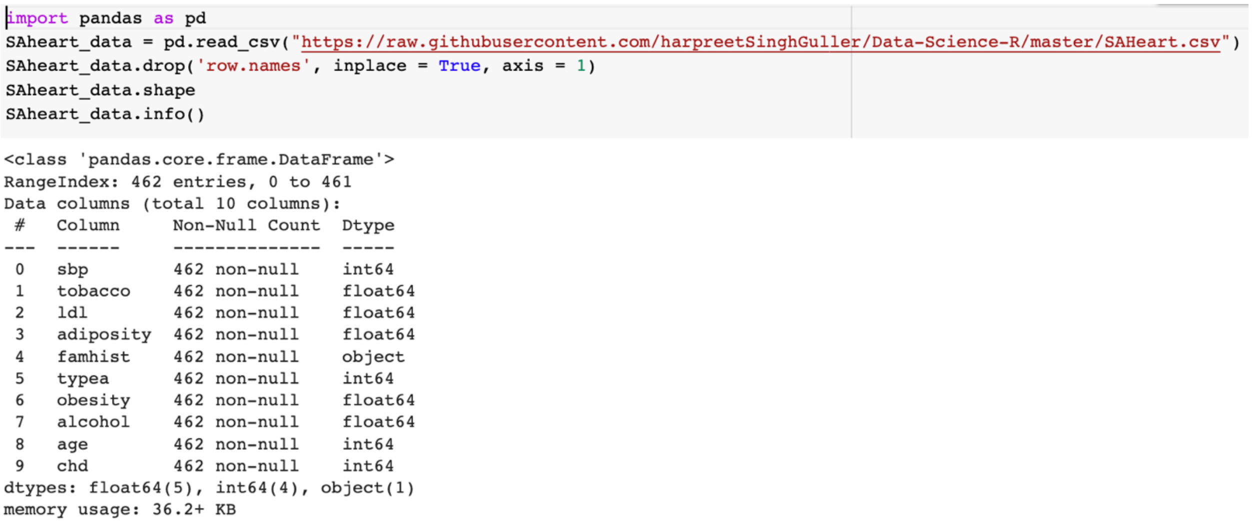
Use the SAheart Dataset to Answer Questions 13 to 20.
The dataset SAheart.data is taken from the link below:

http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data

The dataset contains retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. There are roughly two controls per case of Coronary Heart Disease (CHD). Many of the CHD-positive men have undergone blood pressure reduction treatment and other programs to reduce their risk factors after their CHD event. In some cases, the measurements were made after these treatments. These data are taken from a larger dataset, described in Rousseauw et al. (1983), *South African Medical Journal*. It is a tab separated file (csv) and contains the following columns (*source: http://www-stat.stanford.edu*)

- *sbp* – Systolic blood pressure
- *tobacco* – Cumulative tobacco (kg)
- *ldl* – Low density lipoprotein cholesterol
- *adiposity*
- *famhist* – Family history of heart disease (Present, Absent)
- *typea* – Type-A behaviour
- *obesity*
- *alcohol* – Current alcohol consumption
- *age* – Age at onset
- *chd* – Response, coronary heart disease

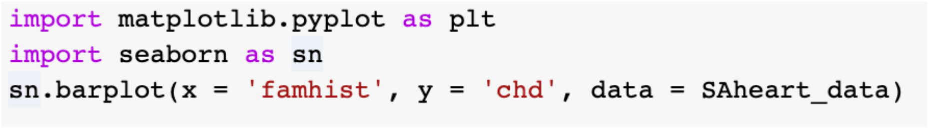
13. How many records are present in the dataset? Print the metadata information of the dataset.



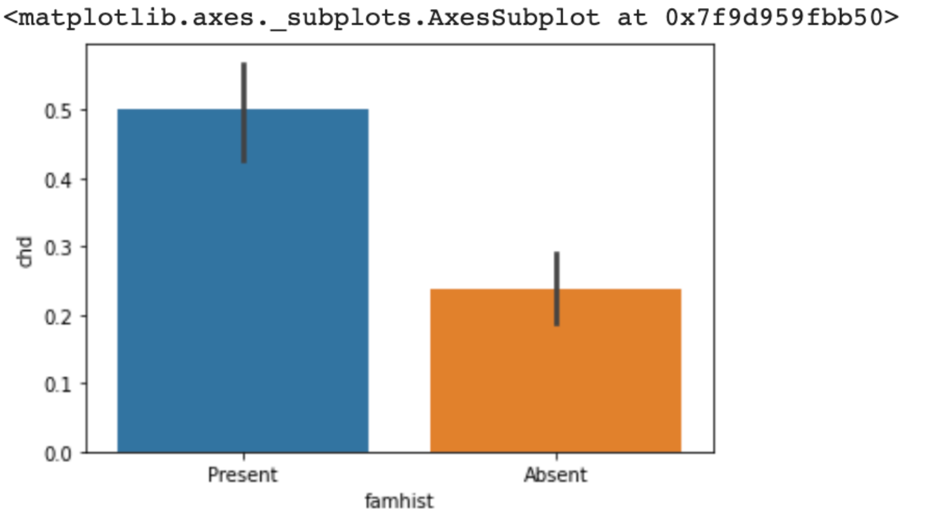
```
import pandas as pd
SAheart_data = pd.read_csv("https://raw.githubusercontent.com/harpreetSinghGuller/Data-Science-R/master/SAHeart.csv")
SAheart_data.drop('row.names', inplace = True, axis = 1)
SAheart_data.shape
SAheart_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 462 entries, 0 to 461
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   sbp        462 non-null    int64  
 1   tobacco    462 non-null    float64
 2   ldl        462 non-null    float64
 3   adiposity  462 non-null    float64
 4   famhist    462 non-null    object 
 5   typea      462 non-null    int64  
 6   obesity    462 non-null    float64
 7   alcohol    462 non-null    float64
 8   age        462 non-null    int64  
 9   chd        462 non-null    int64  
dtypes: float64(5), int64(4), object(1)
memory usage: 36.2+ KB
```

14. Draw a bar plot to show the number of persons having CHD or not in comparison to they having family history of the disease or not.

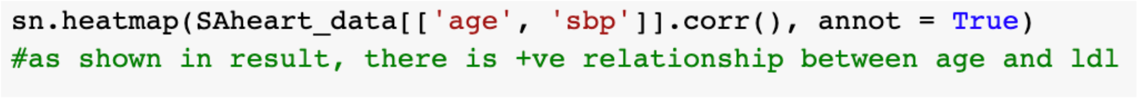


```
import matplotlib.pyplot as plt
import seaborn as sn
sn.barplot(x = 'famhist', y = 'chd', data = SAheart_data)
```

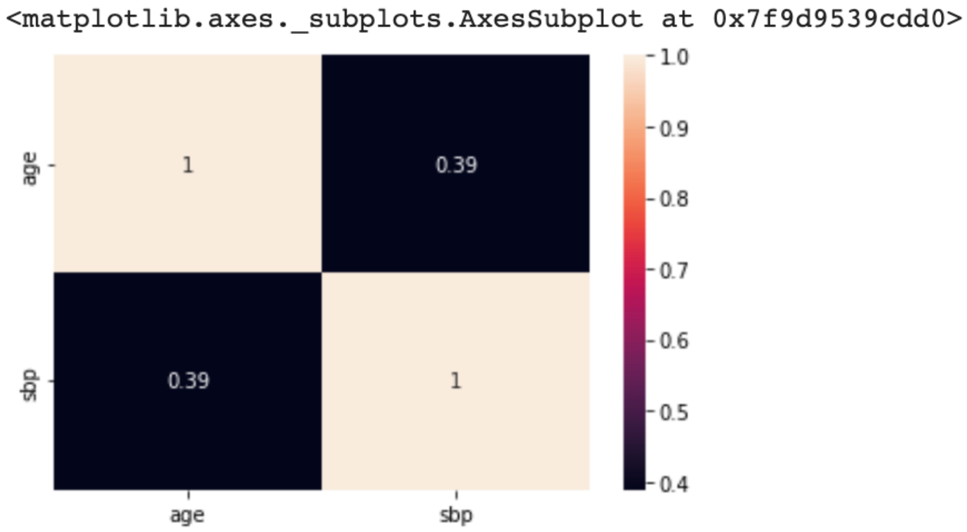


```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9d959fbb50>
```

15. Does age have any correlation with sbp? Choose appropriate plot to show the relationship.



```
sn.heatmap(SAheart_data[['age', 'sbp']].corr(), annot = True)
#as shown in result, there is +ve relationship between age and ldl
```

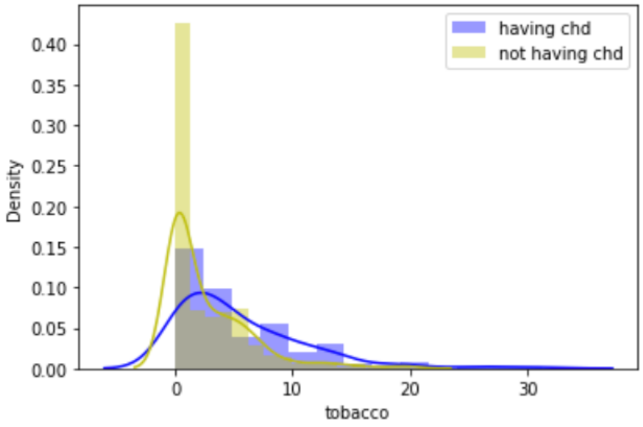


```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9d9539cdd0>
```

16. Compare the distribution of tobacco consumption for persons having CHD and not having CHD. Can you interpret the effect of tobacco consumption on having coronary heart disease?

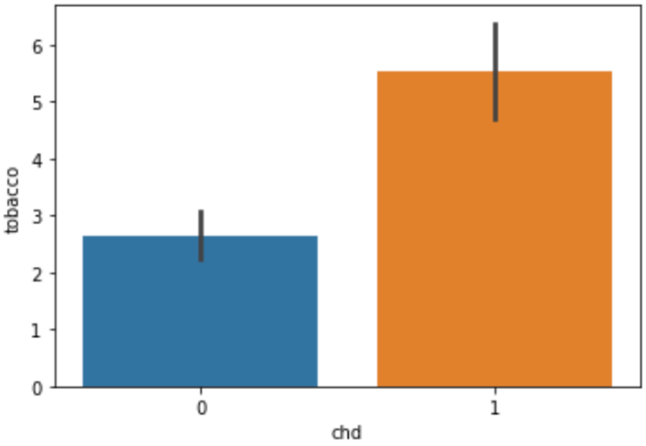
```
sn.distplot(SAheart_data[SAheart_data['chd'] == 1]
            ['tobacco'],
            color = 'b',
            label = 'having chd')
sn.distplot(SAheart_data[SAheart_data['chd'] == 0]
            ['tobacco'],
            color = 'y',
            label = 'not having chd')
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619
  warnings.warn(msg, FutureWarning)
<matplotlib.legend.Legend at 0x7f9d8dddfa10>
```

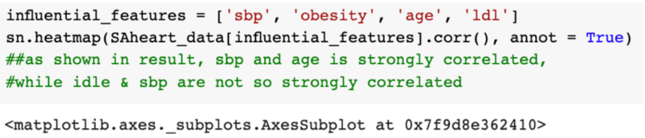


```
sn.barplot(x = 'chd', y = 'tobacco', data = SAheart_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9d8d7abe90>
```

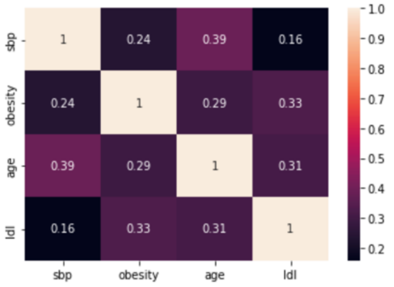


17. How are the parameters sbp, obesity, age and ldl correlated? Choose the right plot to show the relationships.



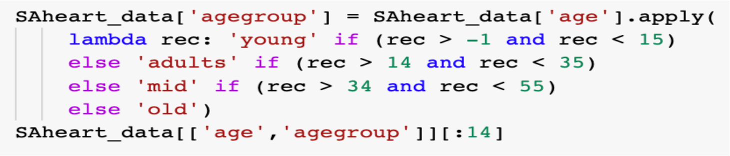
```
influential_features = ['sbp', 'obesity', 'age', 'ldl']
sn.heatmap(SAheart_data[influential_features].corr(), annot = True)
##as shown in result, sbp and age is strongly correlated,
#while idle & sbp are not so strongly correlated
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9d8e362410>
```

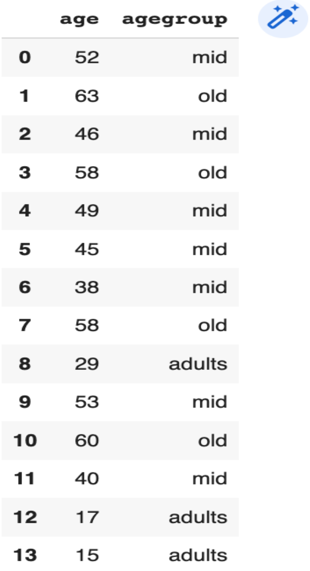


18. Derive a new column called *agegroup* from *age* column where persons falling in different age ranges are categorized as below.

(0-15): young
(15-35): adults
(35-55): mid
(55-): old

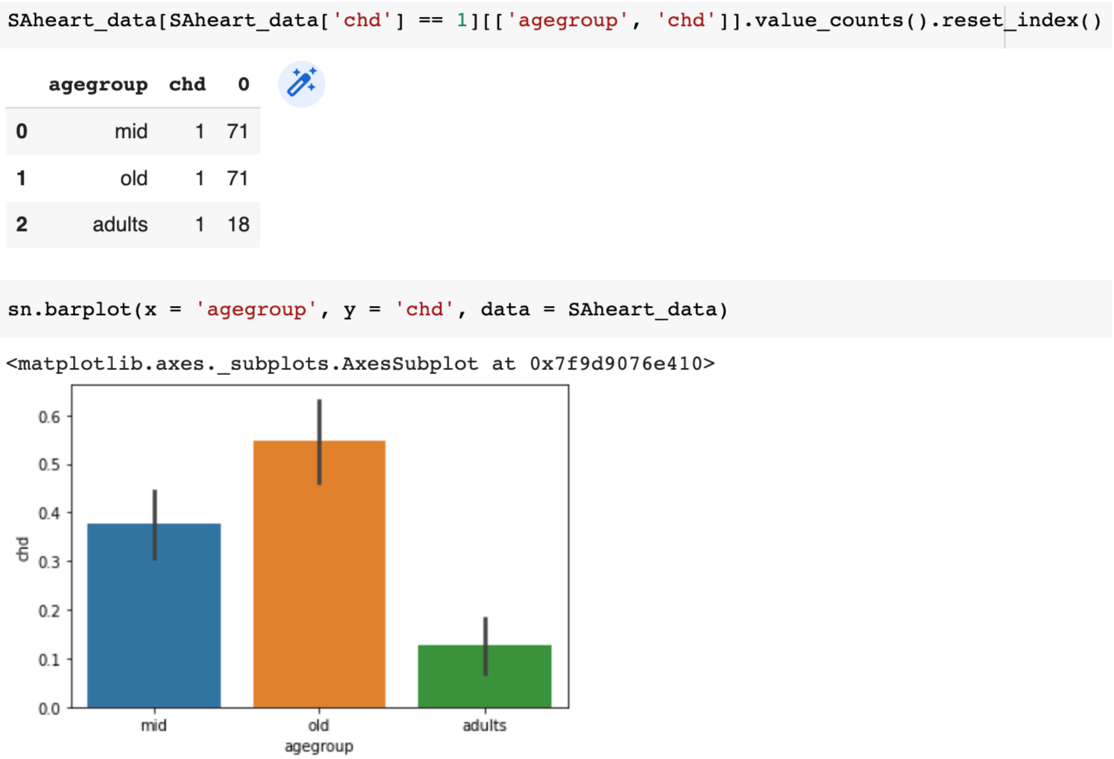


```
SAheart_data['agegroup'] = SAheart_data['age'].apply(
    lambda rec: 'young' if (rec > -1 and rec < 15)
    else 'adults' if (rec > 14 and rec < 35)
    else 'mid' if (rec > 34 and rec < 55)
    else 'old')
SAheart_data[['age','agegroup']][:14]
```



	age	agegroup
0	52	mid
1	63	old
2	46	mid
3	58	old
4	49	mid
5	45	mid
6	38	mid
7	58	old
8	29	adults
9	53	mid
10	60	old
11	40	mid
12	17	adults
13	15	adults

19. Find out the number of CHD cases in different age categories. Do a barplot and sort them in the order of age groups.



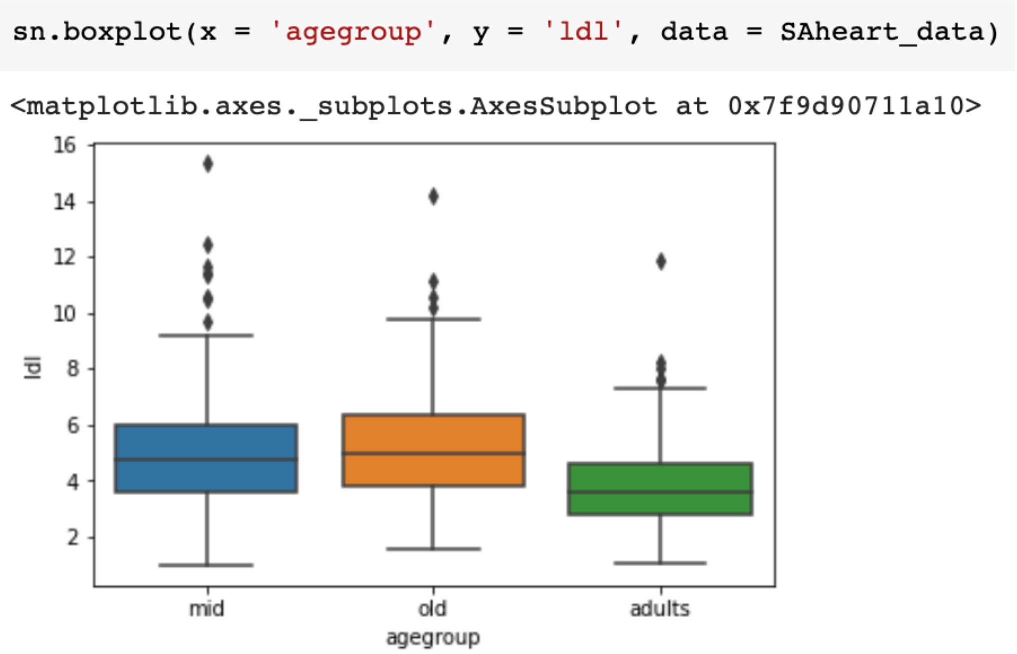
```
SAheart_data[SAheart_data['chd'] == 1][['agegroup', 'chd']].value_counts().reset_index()
```

	agegroup	chd	0
0	mid	1	71
1	old	1	71
2	adults	1	18

```
sn.barplot(x = 'agegroup', y = 'chd', data = SAheart_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9d9076e410>
```

20. Draw a box plot to compare distributions of *ldl* for different age groups.

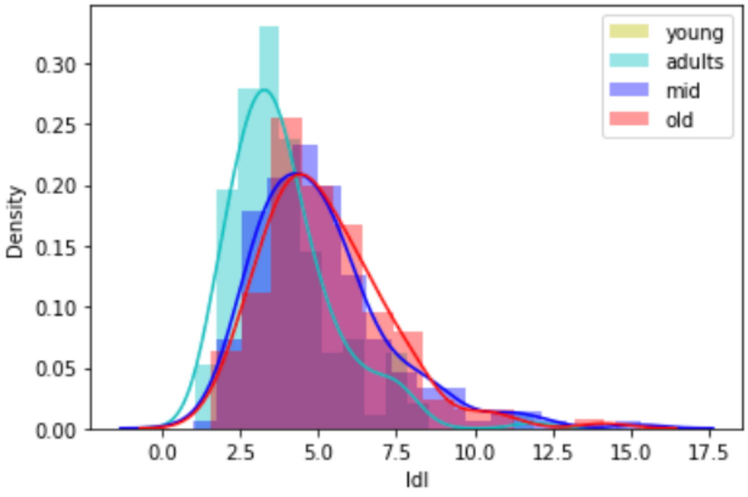


```
sn.boxplot(x = 'agegroup', y = 'ldl', data = SAheart_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9d90711a10>
```

```
sn.distplot(SAheart_data[SAheart_data['agegroup'] == 'young']
            ['ldl'],
            color = 'y',
            label = 'young')
sn.distplot(SAheart_data[SAheart_data['agegroup'] == 'adults']
           ['ldl'],
            color = 'c',
            label = 'adults')
sn.distplot(SAheart_data[SAheart_data['agegroup'] == 'mid']
            ['ldl'],
            color = 'b',
            label = 'mid')
sn.distplot(SAheart_data[SAheart_data['agegroup'] == 'old']
           ['ldl'],
            color = 'r',
            label = 'old')
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/numpy/lib/histograms.py:906: I
  return n/db/n.sum(), bin_edges
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619
  warnings.warn(msg, FutureWarning)
<matplotlib.legend.Legend at 0x7f9d8db60cd0>
```


young
adults
mid
old
Density
ldl