РК1

Студент: Ван Чаочао

Группа: ИУ5И-22М

Номер по списку группы (вариант): 16

Вариант задачи №1 - 16

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Бокса-Кокса (Box-Cox transformation).

Вариант задачи №2 - 36

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectKBest для 5 лучших признаков, и метод, основанный на взаимной информации.

Дополнительное задание (по группам)

для произвольной колонки данных построить гистограмму.

Импортирование необходимых библиотек

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_boston
from sklearn.feature_selection import mutual_info_classif, mutual_info_regression
from sklearn.model_selection import train_test_split
color=sns.color_palette()
sns.set_style('darkgrid')
pd.set_option('display.float_format',lambda x: '{:.3f}'.format(x))
%matplotlib inline
```

In [3]:

```
data=pd.read_csv(r'C:\Users\asus\Desktop\iu5\MMO\PK1\data.csv')
df=data.dropna()
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1326 entries, 0 to 1326
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      1326 non-null   int64
 1   Title           1326 non-null   object
 2   Published_date  1326 non-null   object
 3   Views           1326 non-null   int64
 4   Likes           1326 non-null   int64
 5   Desc            1326 non-null   object
 6   Comments        1326 non-null   int64
 7   Duration        1326 non-null   object
 8   Caption         1326 non-null   bool
 9   Month           1326 non-null   object
dtypes: bool(1), int64(4), object(5)
memory usage: 104.9+ KB
```

Out[3]:

	Unnamed: 0	Title	Published_date	Views	Likes	Desc	Comments	Duratio
0	0	MORBIUS Vignette (Telugu) - The Lore of Morbius	2022-02-26	16165	641	From a forbidden Marvel character to a new Mar...	19	PT1M1
1	1	MORBIUS Vignette (Tamil) - The Lore of Morbius	2022-02-26	189376	13830	From a forbidden Marvel character to a new Mar...	264	PT1M1
2	2	Hey Sinamika - Jukebox | Dulquer Salmaan, Adit...	2022-02-25	70182	2540	Achamillai - 00:00\nThozhi - 03:47\nMegham - 0...	148	PT27M42
3	3	Hey Sinamika (Telugu) - Jukebox | Dulquer Salm...	2022-02-25	15461	502	Bhayam Vaddu - 00:00\nAlalegase - 03:47\nManda...	27	PT27M39
4	4	Valimai Kannada - Jukebox | Ajith Kumar | Yuva...	2022-02-23	55125	4657	Naave Bere Thara\nMother Song\nEnno Kore Na\nW...	146	PT13M3

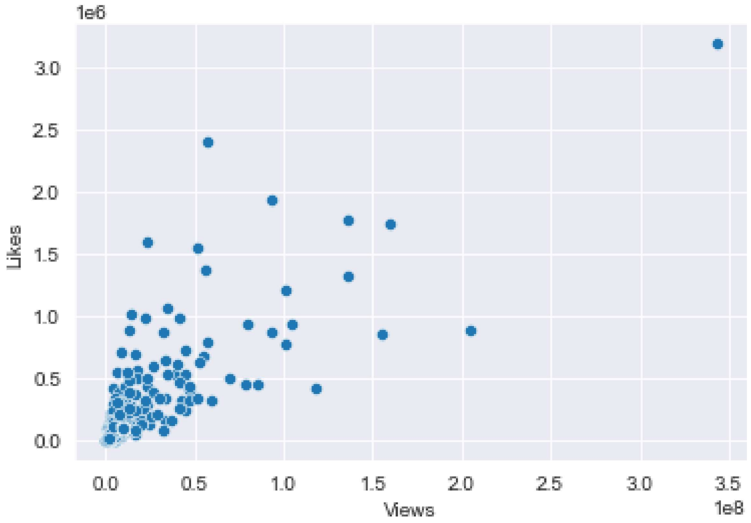
Просмотрите график рассеяния перед масштабированием

In [4]:

```
sns.scatterplot(x='Views',y='Likes',data=df)
```

Out[4]:

```
<AxesSubplot:xlabel='Views', ylabel='Likes'>
```

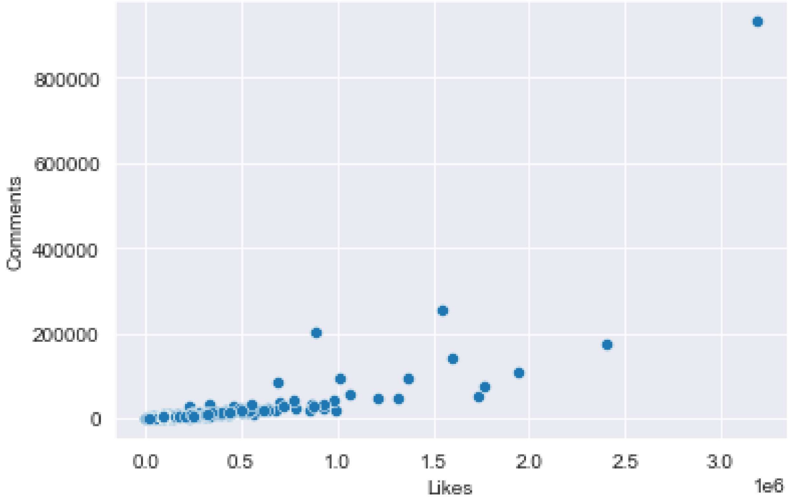


In [6]:

```
sns.scatterplot(x='Likes',y='Comments',data=df)
```

Out[6]:

```
<AxesSubplot:xlabel='Likes', ylabel='Comments'>
```

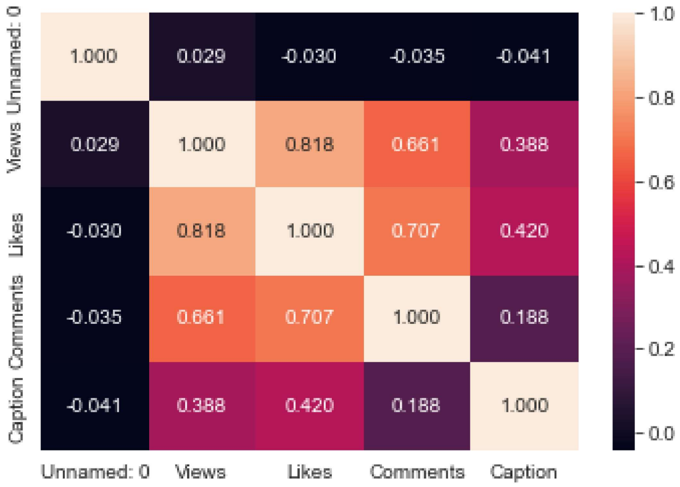


In [7]:

```
sns.heatmap(data.corr(),annot=True, fmt='.3f')
```

Out[7]:

```
<AxesSubplot:>
```



Преобразование Бокса-Кокса

In [8]:

```
import scipy.stats as stats
```

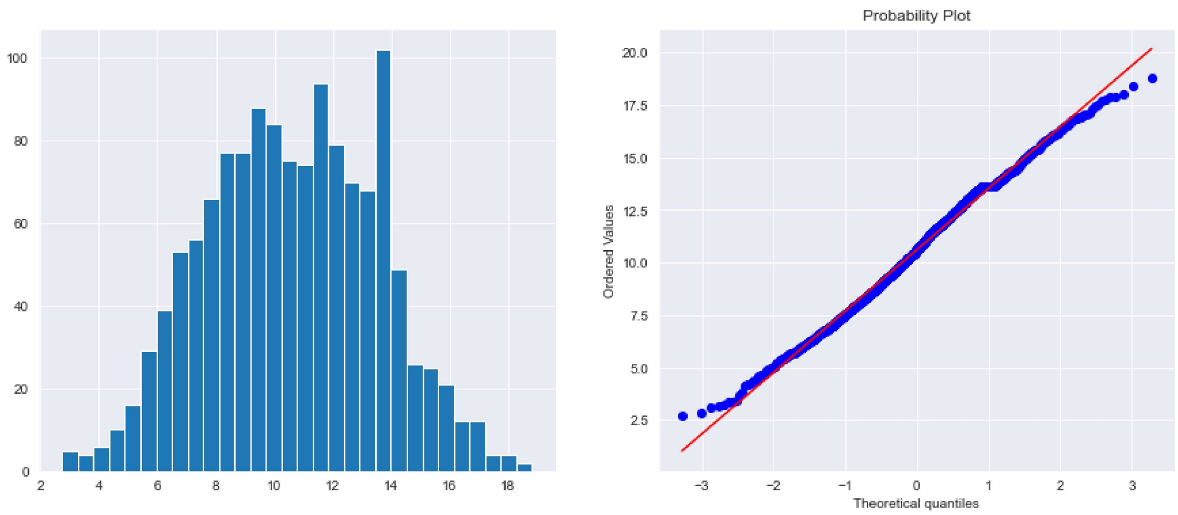
In [9]:

```
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30)
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot = plt)
    plt.show()
```

In [11]:

```
data['Likes_boxcox'], param = stats.boxcox(data['Likes'])
print('Оптимальное значение λ = {}'.format(param))
diagnostic_plots(data, 'Likes_boxcox')
```

```
Оптимальное значение λ = 0.029391982283770254
```



Задача №36.

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectKBest для 5 лучших признаков, и метод, основанный на взаимной информации.

In [27]:

```
data_2=pd.read_csv(r'C:\Users\asus\Desktop\iu5\MMO\PK1\College_Data.csv')
data_2.info()
data_2.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   777 non-null    object 
 1   Private      777 non-null    object 
 2   Apps         777 non-null    int64  
 3   Accept       777 non-null    int64  
 4   Enroll       777 non-null    int64  
 5   Top10perc    777 non-null    int64  
 6   Top25perc    777 non-null    int64  
 7   F.Undergrad  777 non-null    int64  
 8   P.Undergrad  777 non-null    int64  
 9   Outstate     777 non-null    int64  
 10  Room.Board   777 non-null    int64  
 11  Books        777 non-null    int64  
 12  Personal     777 non-null    int64  
 13  PhD          777 non-null    int64  
 14  Terminal     777 non-null    int64  
 15  S.F.Ratio    777 non-null    float64
 16  perc.alumni  777 non-null    int64  
 17  Expend       777 non-null    int64  
 18  Grad.Rate    777 non-null    int64  
dtypes: float64(1), int64(16), object(2)
memory usage: 115.5+ KB
```

Out[27]:

Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Persona
721	23	52	2885	537	7440	3300	450	220
512	16	29	2683	1227	12280	6450	750	150
336	22	50	1036	99	11250	3750	400	116
137	60	89	510	63	12960	5450	450	87
55	16	44	249	869	7560	4120	800	150

In [39]:

```
a=data_2.drop(labels=['Unnamed: 0','Private'],axis=1)
```

In [40]:

```
a.shape
```

Out[40]:

```
(777, 17)
```

In [43]:

```
data_2.describe()
```

Out[43]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outs
count	777.000	777.000	777.000	777.000	777.000	777.000	777.000	777.
mean	3001.638	2018.804	779.973	27.559	55.797	3699.907	855.299	10440.
std	3870.201	2451.114	929.176	17.640	19.805	4850.421	1522.432	4023.
min	81.000	72.000	35.000	1.000	9.000	139.000	1.000	2340.
25%	776.000	604.000	242.000	15.000	41.000	992.000	95.000	7320.
50%	1558.000	1110.000	434.000	23.000	54.000	1707.000	353.000	9990.
75%	3624.000	2424.000	902.000	35.000	69.000	4005.000	967.000	12925.
max	48094.000	26330.000	6392.000	96.000	100.000	31643.000	21836.000	21700.

In [44]:

```
c=data_2.drop(labels=['Outstate'],axis=1)
d=data_2.Outstate
```

In [45]:

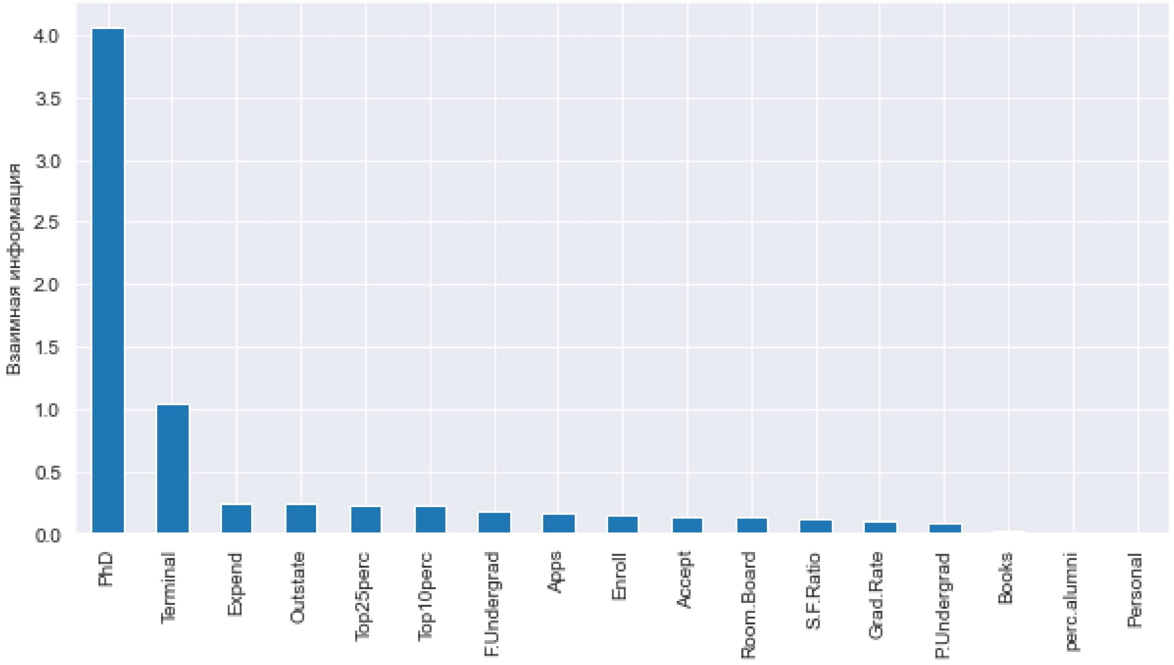
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_regression
```

In [49]:

```
mi = mutual_info_regression(a,b)
mi = pd.Series(mi)
mi.index = a.columns
mi.sort_values(ascending=False).plot.bar(figsize=(10,5))
plt.ylabel('Взаимная информация')
```

Out[49]:

```
Text(0, 0.5, 'Взаимная информация')
```



In [50]:

```
selector=SelectKBest(mutual_info_regression,k=5)
selector.fit(a,b)
X_selected=selector.transform(a)
X_selected.shape
```

Out[50]:

```
(777, 5)
```

In [51]:

```
selector.get_support(indices=False)
```

Out[51]:

```
array([False, False, False,  True, False, False, False,  True, False,
       False, False,  True,  True, False, False,  True, False])
```

In [52]:

```
a.columns[selector.get_support()]
```

Out[52]:

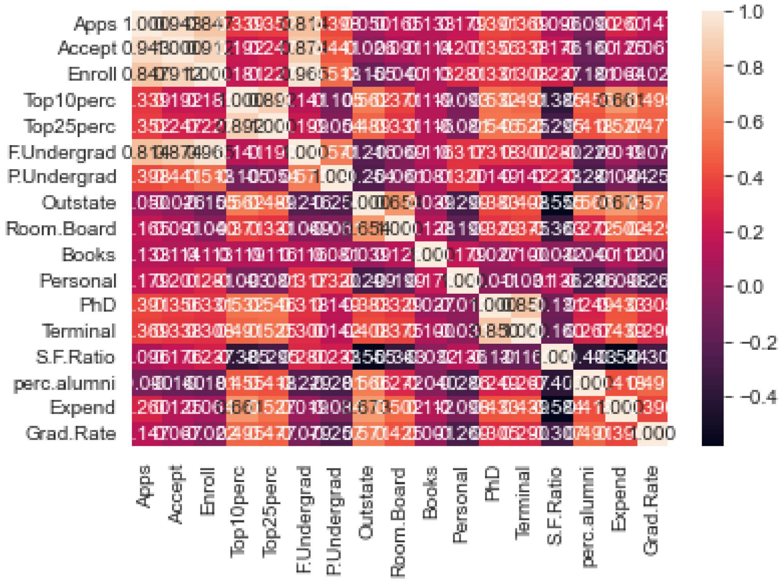
```
Index(['Top10perc', 'Outstate', 'PhD', 'Terminal', 'Expend'], dtype='object')
```

In [53]:

```
sns.heatmap(data_2.corr(),annot=True, fmt='.3f')
```

Out[53]:

```
<AxesSubplot:>
```



Дополнительное задание

для произвольной колонки данных построить гистограмму.

In [61]:

```
out_yes=data_2[data_2['Grad.Rate'] >= 50]
```

In [62]:

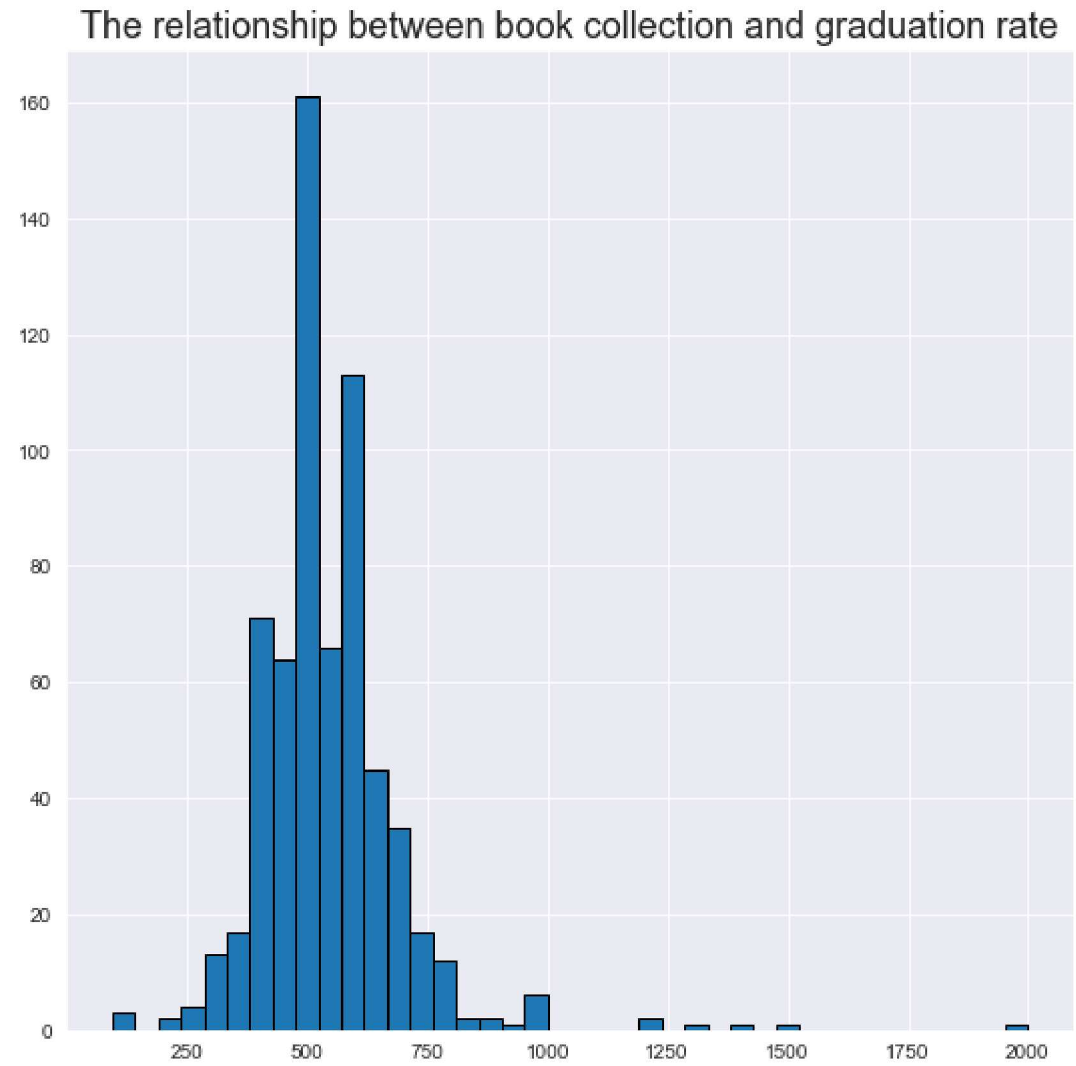
```
out_no=data_2[data_2['Grad.Rate'] <= 50]
```

In [65]:

```
fig, ax = plt.subplots(figsize = (9, 9))
#plot
ax.hist(out_yes["Books"], bins=40, edgecolor="black");
plt.title('The relationship between book collection and graduation rate', fontsize=18)
```

Out[65]:

```
Text(0.5, 1.0, 'The relationship between book collection and graduation rate')
```

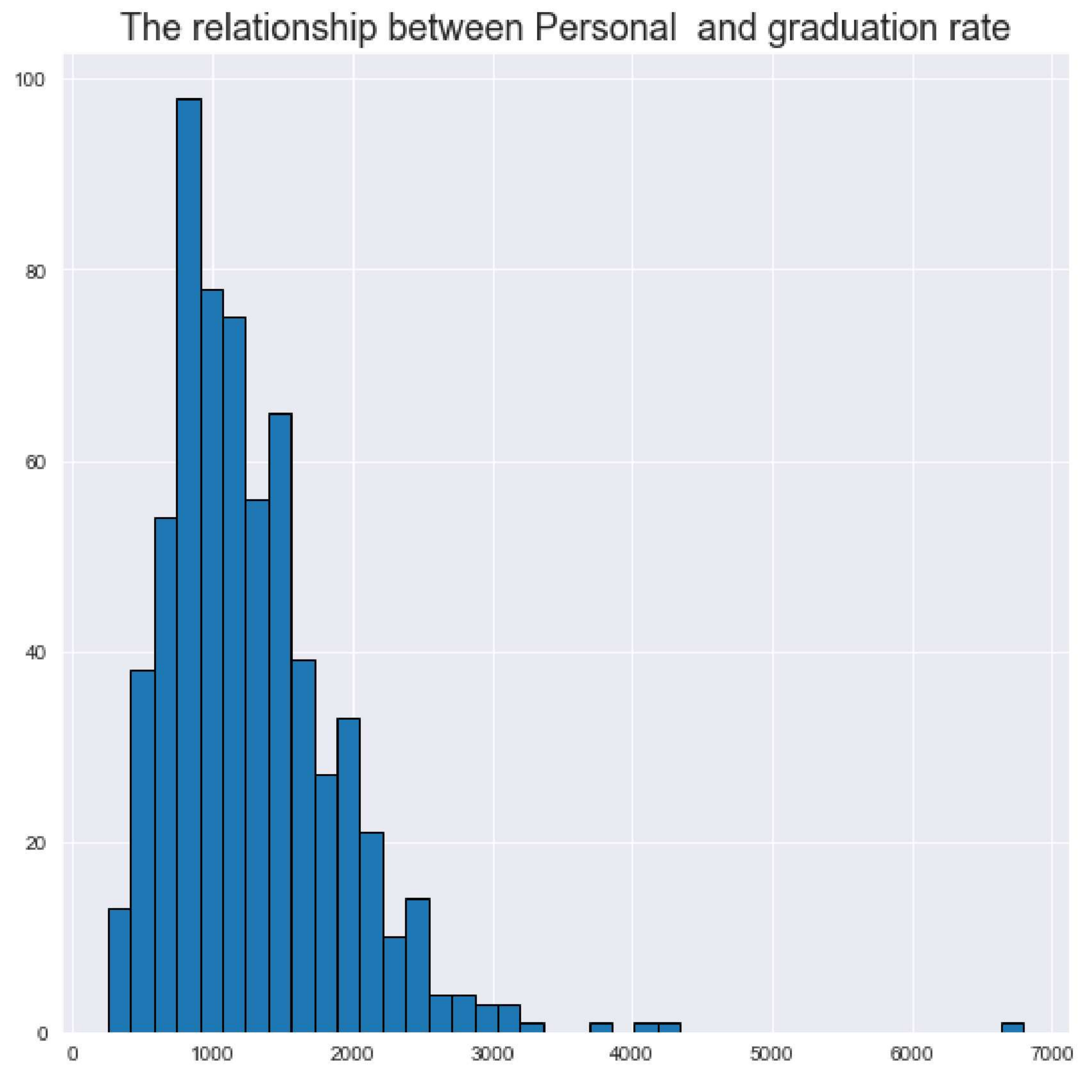


In [68]:

```
fig, ax = plt.subplots(figsize = (9, 9))
#plot
ax.hist(out_yes["Personal"], bins=40, edgecolor="black");
plt.title('The relationship between Personal  and graduation rate', fontsize=18)
```

Out[68]:

```
Text(0.5, 1.0, 'The relationship between Personal  and graduation rate')
```



The relationship between Personal and graduation rate

In [ ]:

```
```