In [10]:
```python
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
```

In [11]:
```python
data = pd.read_csv('Admission_Predict (1).csv')
```

In [12]:
```python
data.head()
```

Out[12]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

In [13]:
```python
data.shape
```

Out[13]:
```
(400, 9)
```

In [14]:
```python
data.info()
```

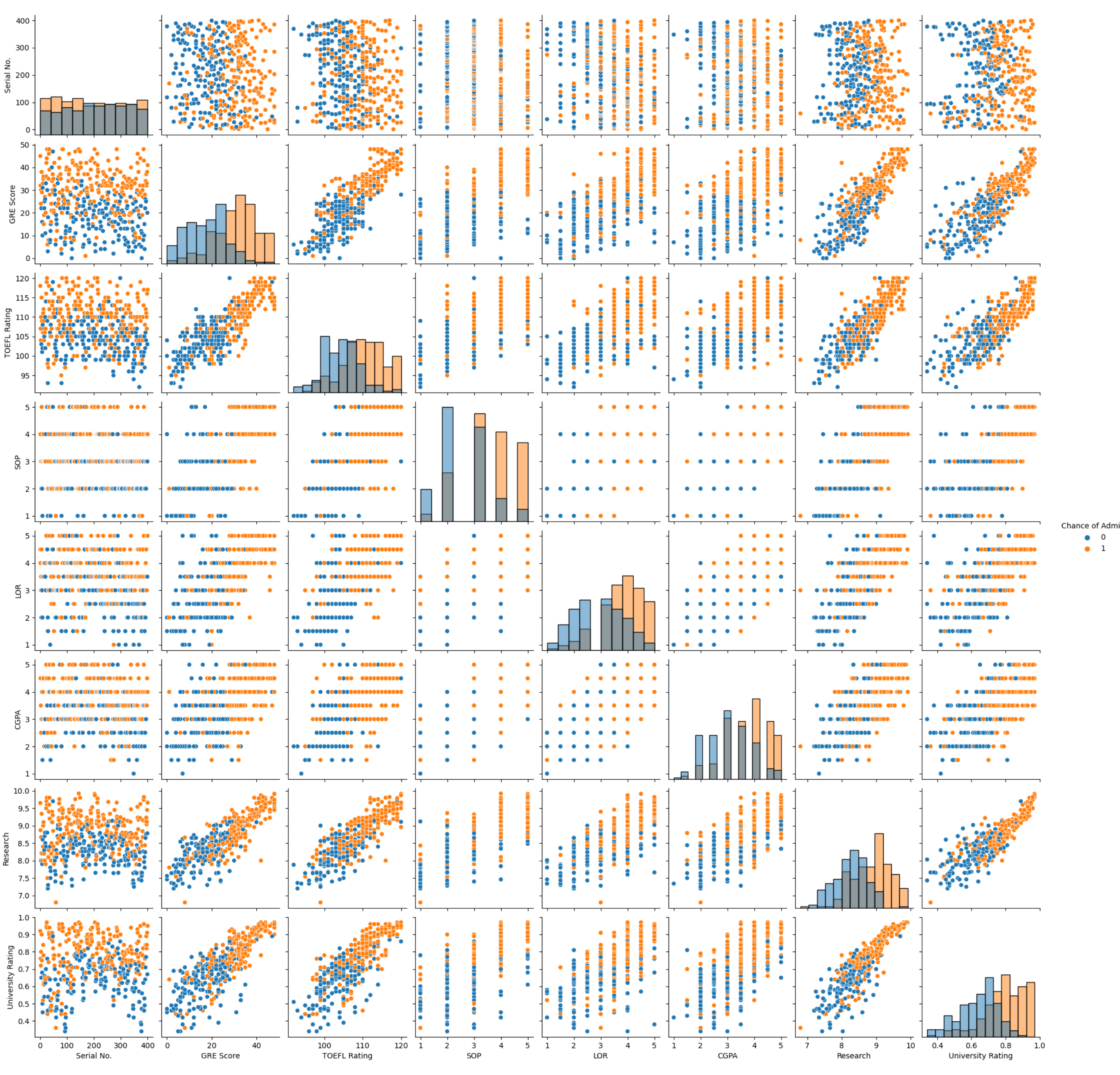
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial No.         400 non-null    int64  
 1   GRE Score          400 non-null    int64  
 2   TOEFL Score        400 non-null    int64  
 3   University Rating  400 non-null    int64  
 4   SOP                400 non-null    float64
 5   LOR                400 non-null    float64
 6   CGPA               400 non-null    float64
 7   Research           400 non-null    int64  
 8   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
```

In [15]:
```python
data.isnull().sum()
```

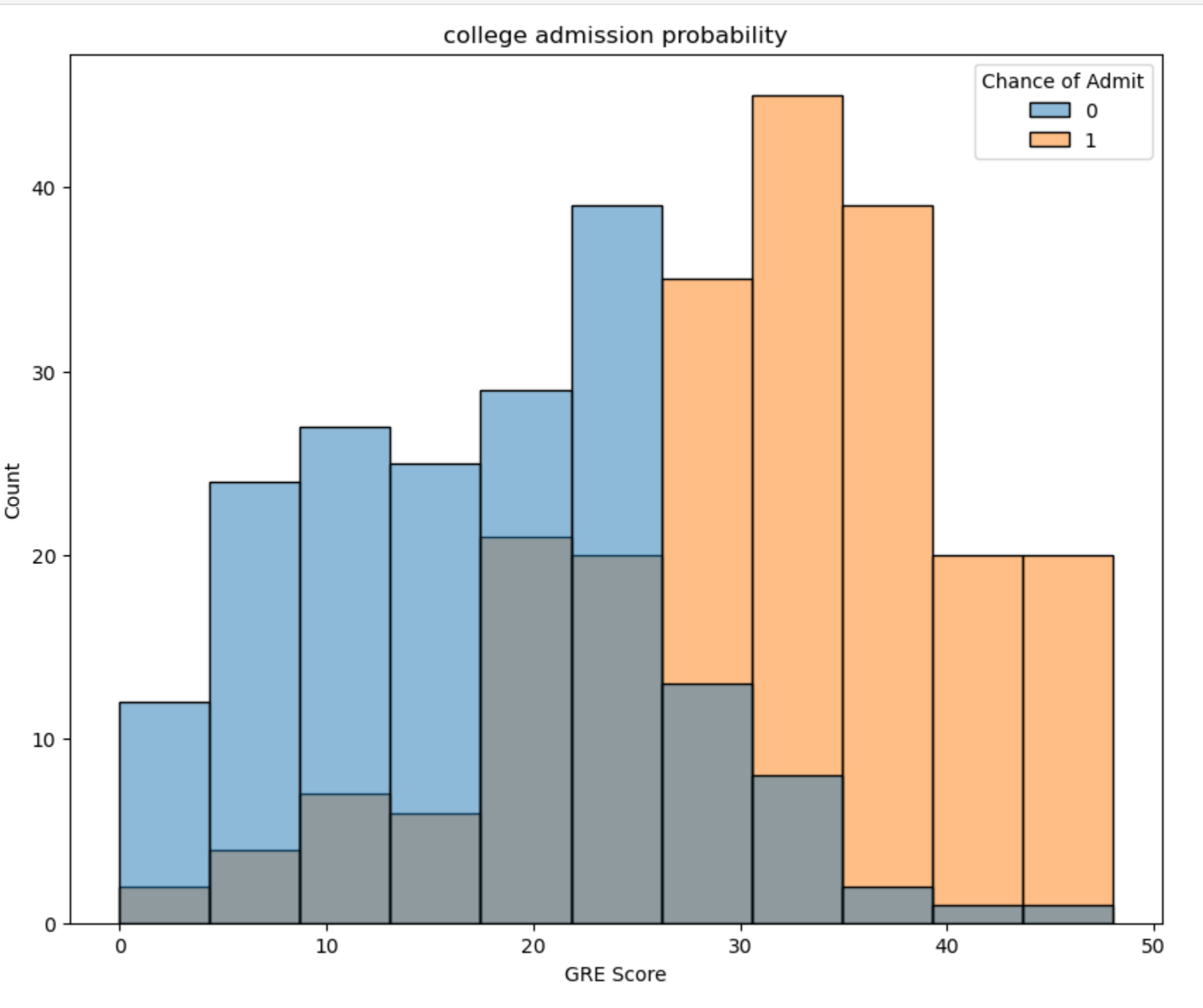
Out[15]:
```
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

In [44]:
```python
sns.pairplot(data,hue="Chance of Admit",diag_kind="hist")
```

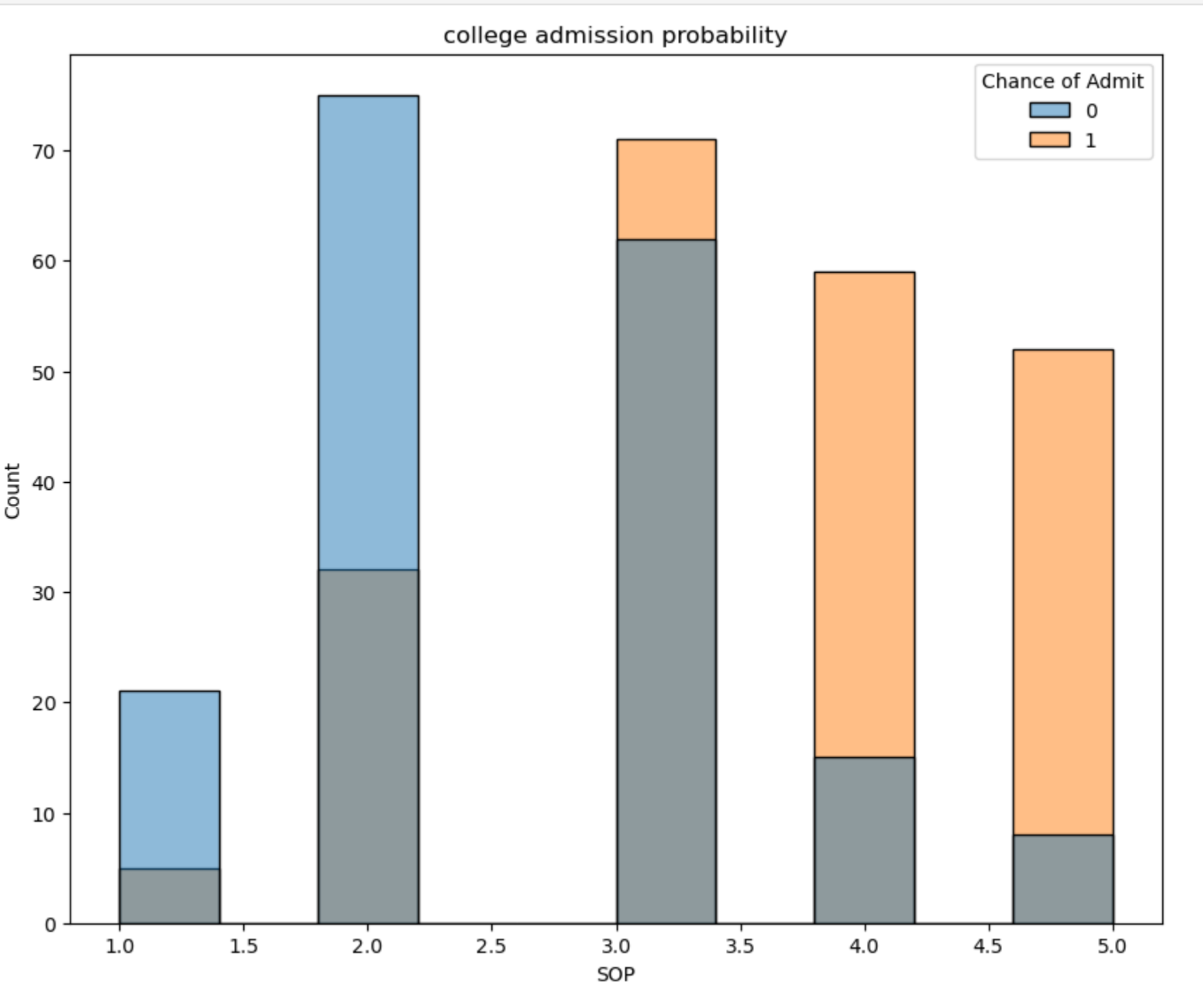
Out[44]:
```
<seaborn.axisgrid.PairGrid at 0x2a89b17a110>
```



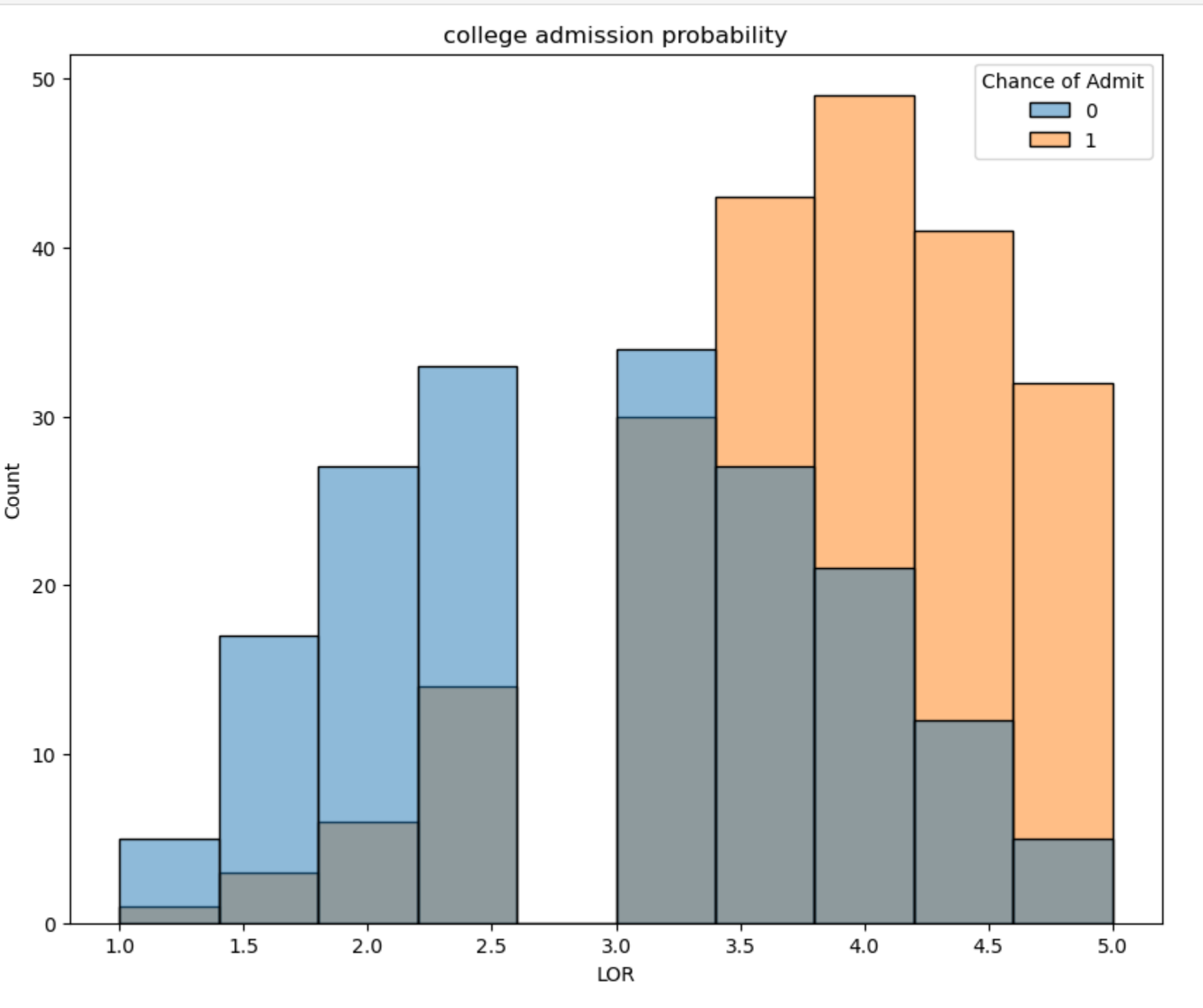
In [37]:
```python
data.columns= ["Serial No.","GRE Score","TOEFL Rating","SOP","LOR","CGPA","Research","Chance of Admit","University Rating"]

plt.figure(figsize=(10, 8))
plt.title("college admission probability")
sns.histplot(x="GRE Score", hue="Chance of Admit", data=data)
plt.show()
```



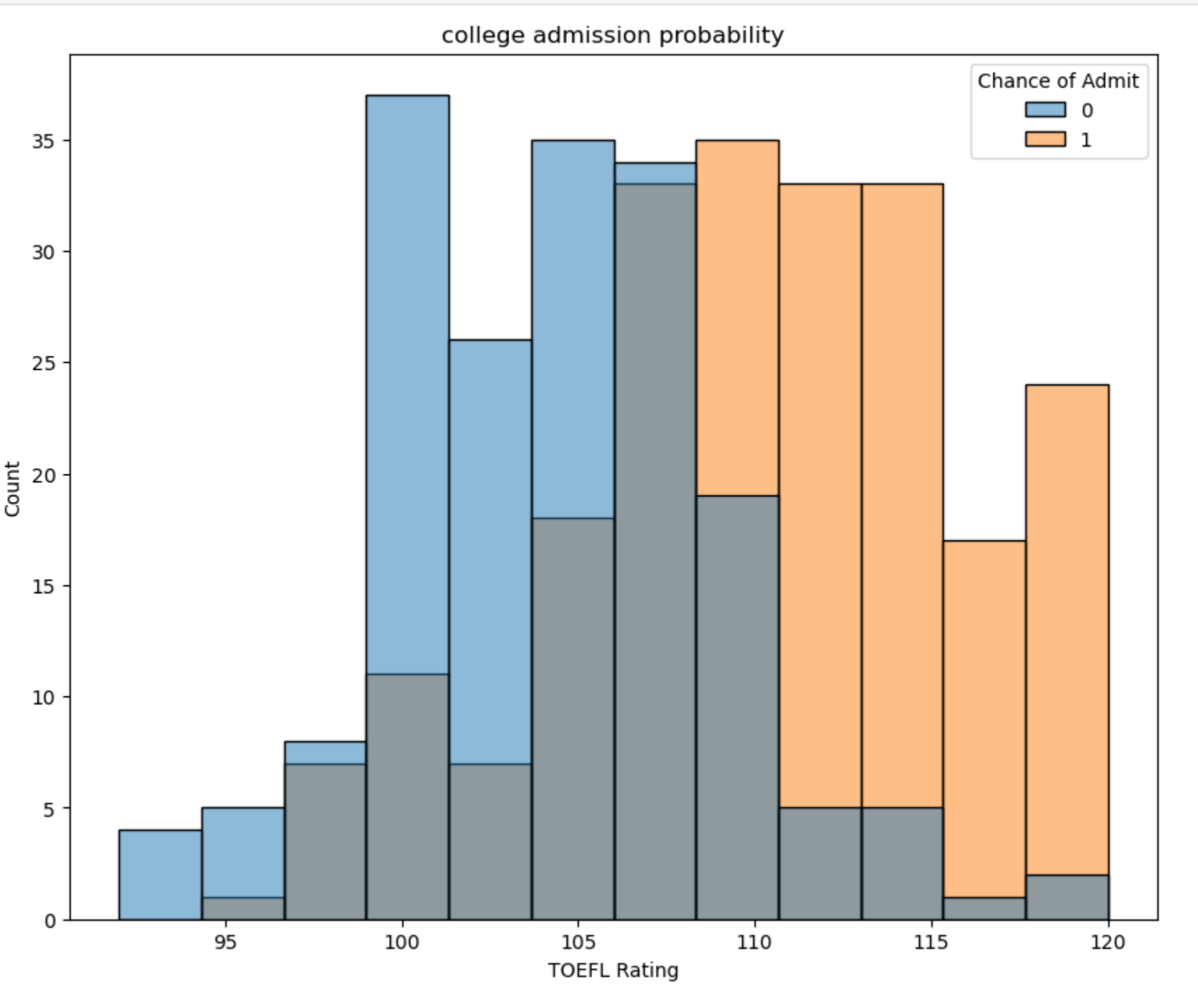
In [38]:
```python
plt.figure(figsize=(10, 8))
plt.title("college admission probability")
sns.histplot(x="SOP", hue="Chance of Admit", data=data)
plt.show()
```



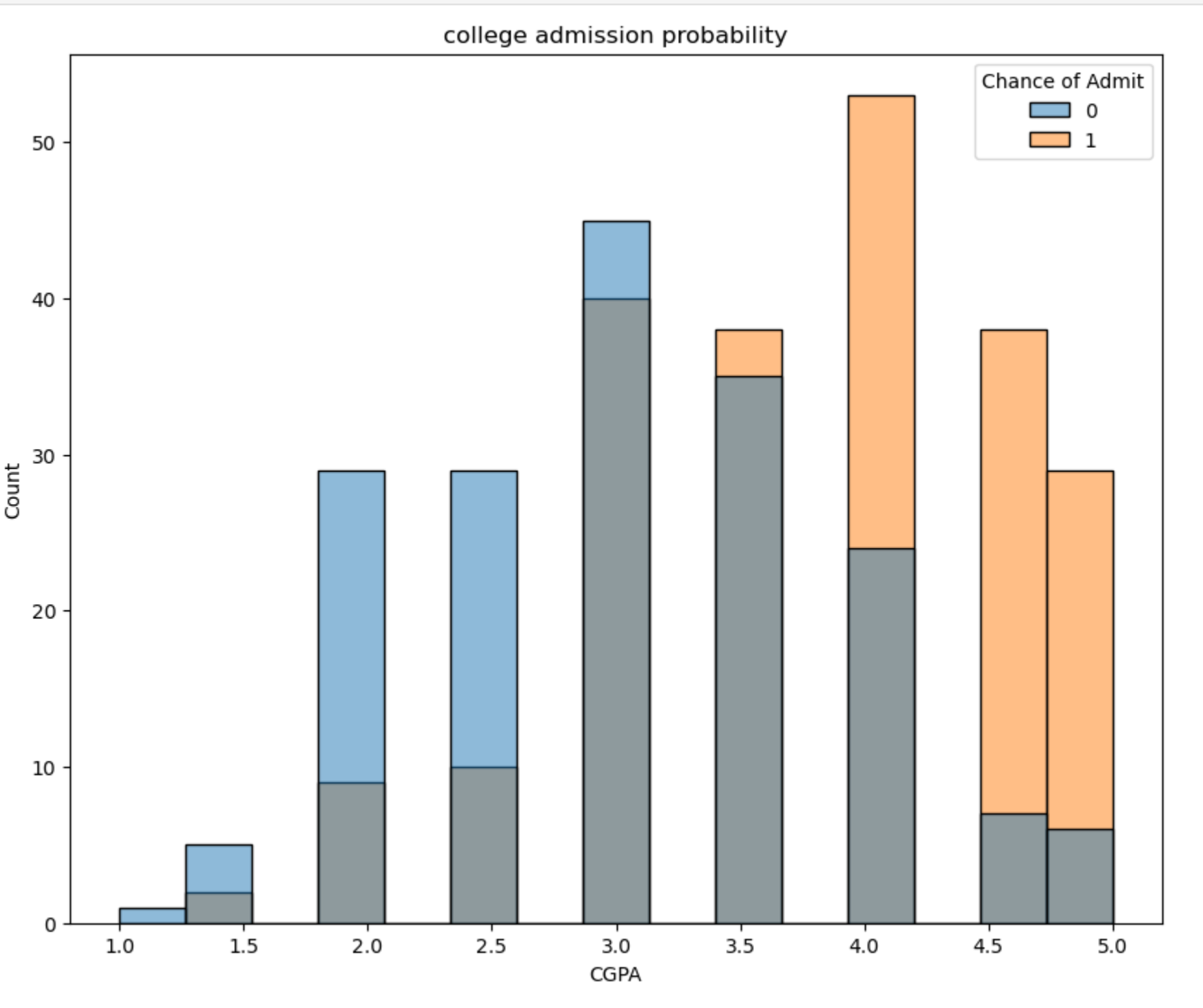
In [39]:
```python
plt.figure(figsize=(10, 8))
plt.title("college admission probability")
sns.histplot(x="LOR", hue="Chance of Admit", data=data)
plt.show()
```



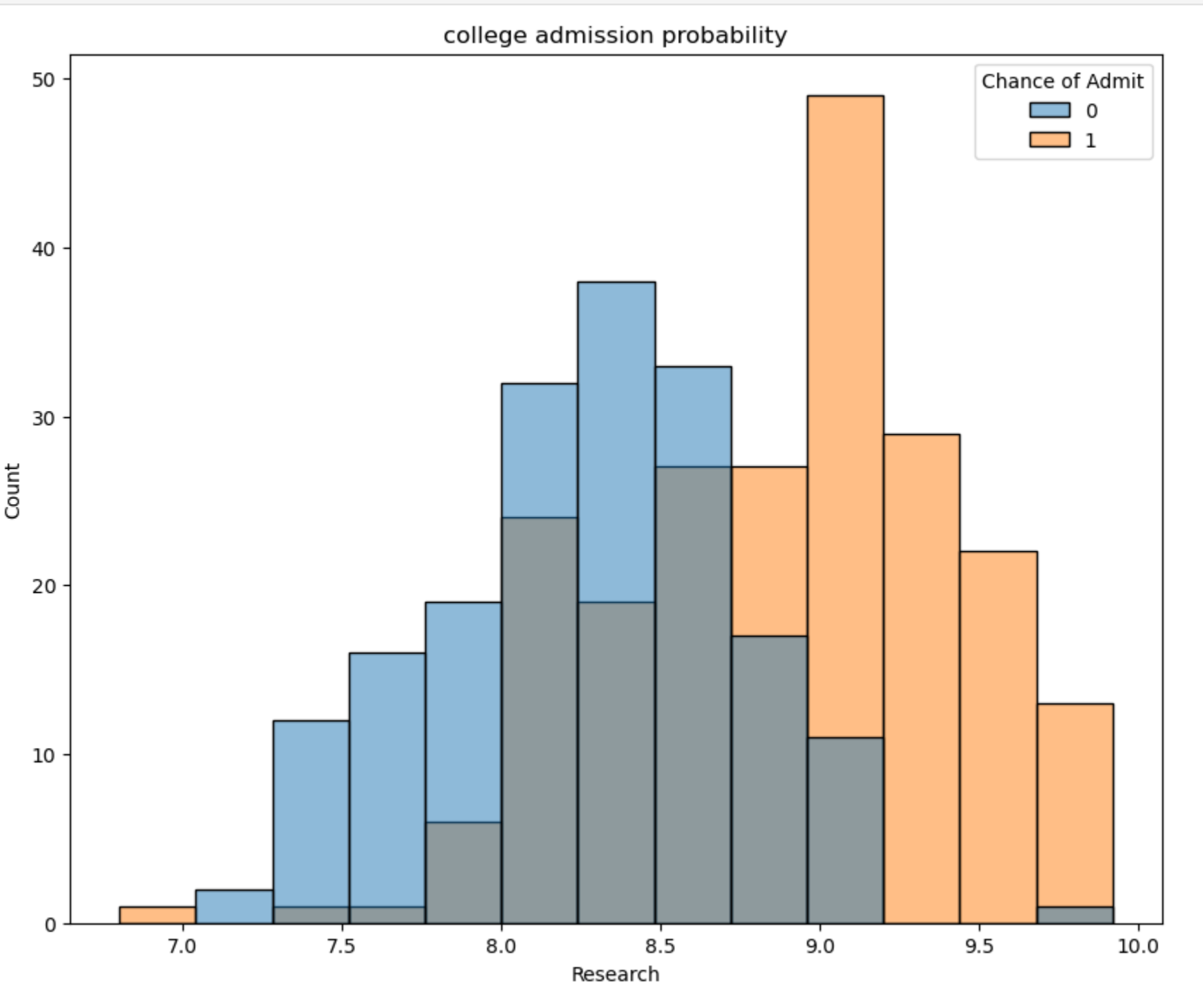
In [40]:
```python
plt.figure(figsize=(10, 8))
plt.title("college admission probability")
sns.histplot(x="TOEFL Rating", hue="Chance of Admit", data=data)
plt.show()
```



In [41]:
```python
plt.figure(figsize=(10, 8))
plt.title("college admission probability")
sns.histplot(x="CGPA", hue="Chance of Admit", data=data)
plt.show()
```



In [42]:
```python
plt.figure(figsize=(10, 8))
plt.title("college admission probability")
sns.histplot(x="Research", hue="Chance of Admit", data=data)
plt.show()
```



In [ ]:

In [21]:
```python
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
data.iloc[:,1]=encoder.fit_transform(data.iloc[:,1].values)
```

In [22]:
```python
data.iloc[:,1]
```

Out[22]:
```
0      45
1      32
2      24
3      30
4      22
       ..
395    32
396    33
397    38
398    20
399    41
Name: GRE Score, Length: 400, dtype: int64
```

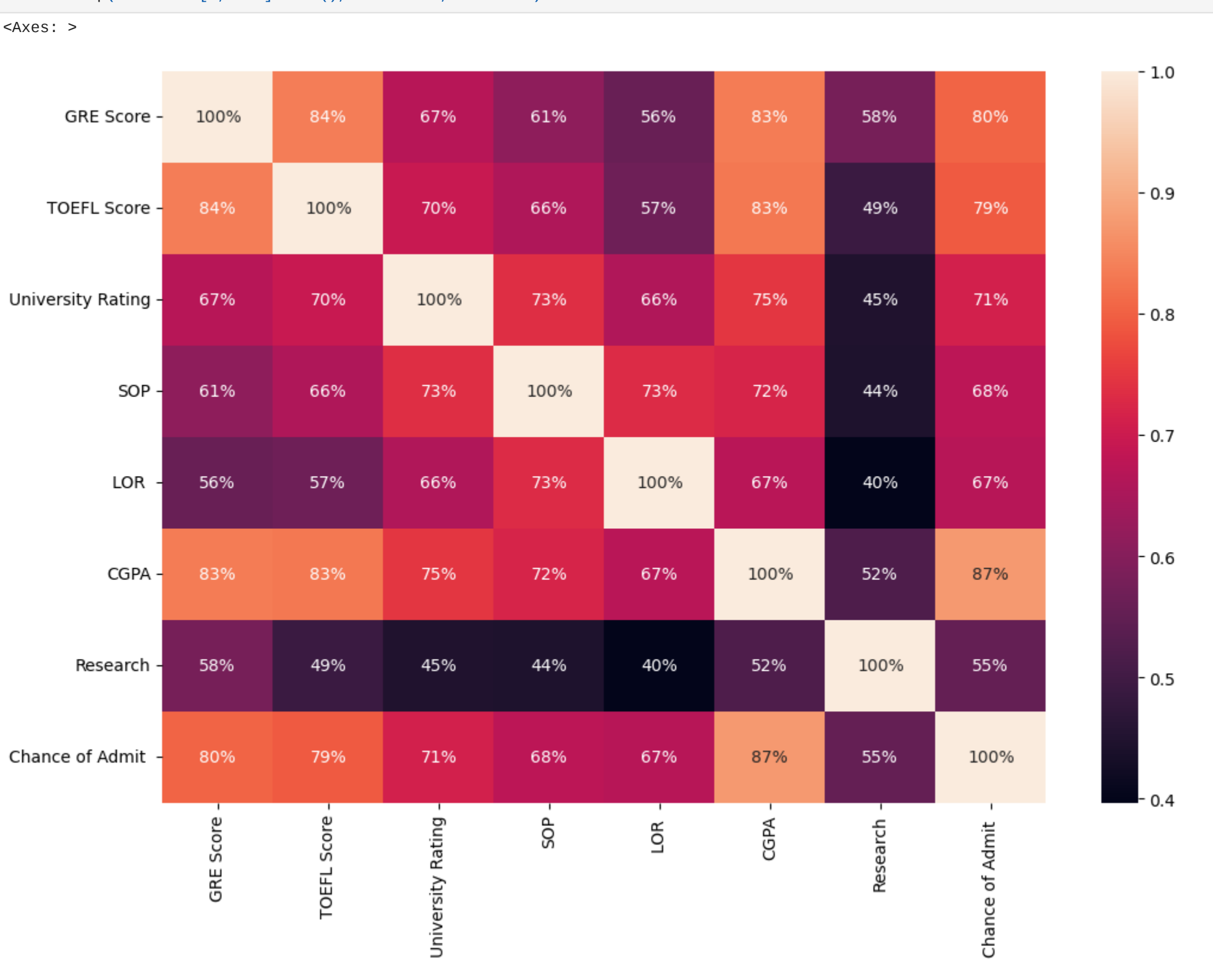
In [23]:
```python
data.iloc[:,1:14].corr()
```

Out[23]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
GRE Score	1.000000	0.836845	0.669845	0.612604	0.557482	0.833104	0.580573	0.802570
TOEFL Score	0.836845	1.000000	0.695590	0.657981	0.567721	0.828417	0.489858	0.791594
University Rating	0.669845	0.695590	1.000000	0.734523	0.660123	0.746479	0.447783	0.711250
SOP	0.612604	0.657981	0.734523	1.000000	0.729593	0.718144	0.444029	0.675732
LOR	0.557482	0.567721	0.660123	0.729593	1.000000	0.670211	0.396859	0.669889
CGPA	0.833104	0.828417	0.746479	0.718144	0.670211	1.000000	0.521654	0.873289
Research	0.580573	0.489858	0.447783	0.444029	0.396859	0.521654	1.000000	0.553202
Chance of Admit	0.802570	0.791594	0.711250	0.675732	0.669889	0.873289	0.553202	1.000000

In [24]:
```python
plt.figure(figsize=(12,8))
sns.heatmap(data.iloc[:,1:13].corr(),annot=True,fmt='.0%')
```

Out[24]:
```
<Axes: >
```



In [47]:
```python
X = data.iloc[:,2:31].values
y = data.iloc[:,1].values
```

In [48]:
```python
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
```

In [49]:
```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

In [50]:
```python
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train,y_train)
lr.score(X_train,y_train)
```

Out[50]:
```
0.24333333333333335
```

In [51]:
```python
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(X_train,y_train)
dtc.score(X_train,y_train)
```

Out[51]:
```
1.0
```

In [52]:
```python
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X_train,y_train)
rfc.score(X_train,y_train)
```

Out[52]:
```
1.0
```

In [ ]:

In [ ]: