In [1]:

```
#For this project we will attempt to use KMeans Clustering to cluster Universities into
to two groups, Private and Public.
#Just note We already have labels for the data here, which in the real world would not
 happen while using K-means clustering as it is
#Unsupervised learning algorithm.

# This is basically a test for how K means fares as a unsupervised learning algo.
#We will use a data frame with 777 observations on the following 18 variables.

#Private A factor with levels No and Yes indicating private or public university
#Apps Number of applications received
#Accept Number of applications accepted
#Enroll Number of new students enrolled
#Top10perc Pct. new students from top 10% of H.S. class
#Top25perc Pct. new students from top 25% of H.S. class
#F.Undergrad Number of fulltime undergraduates
#P.Undergrad Number of parttime undergraduates
#Outstate Out-of-state tuition
#Room.Board Room and board costs
#Books Estimated book costs
#Personal Estimated personal spending
#PhD Pct. of faculty with Ph.D.’s
#Terminal Pct. of faculty with terminal degree
#S.F.Ratio Student/faculty ratio
#perc.alumni Pct. alumni who donate
#Expend Instructional expenditure per student
#Grad.Rate Graduation rate
```

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [7]:

```
univ_df=pd.read_csv('College_Data',index_col=0)
```

In [8]:

```
univ_df.head()
```

Out[8]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537
Adelphi University	Yes	2186	1924	512	16	29	2683	1227
Adrian College	Yes	1428	1097	336	22	50	1036	99
Agnes Scott College	Yes	417	349	137	60	89	510	63
Alaska Pacific University	Yes	193	146	55	16	44	249	869

In [9]:

```
univ_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 777 entries, Abilene Christian University to York College of Pennsy
lvania
Data columns (total 18 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Private      777 non-null    object 
 1   Apps         777 non-null    int64  
 2   Accept       777 non-null    int64  
 3   Enroll       777 non-null    int64  
 4   Top10perc    777 non-null    int64  
 5   Top25perc    777 non-null    int64  
 6   F.Undergrad  777 non-null    int64  
 7   P.Undergrad  777 non-null    int64  
 8   Outstate     777 non-null    int64  
 9   Room.Board   777 non-null    int64  
 10  Books        777 non-null    int64  
 11  Personal     777 non-null    int64  
 12  PhD          777 non-null    int64  
 13  Terminal     777 non-null    int64  
 14  S.F.Ratio    777 non-null    float64
 15  perc.alumni  777 non-null    int64  
 16  Expend       777 non-null    int64  
 17  Grad.Rate    777 non-null    int64  
dtypes: float64(1), int64(16), object(1)
memory usage: 115.3+ KB
```

In [11]:

```
univ_df.describe()
```

Out[11]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Ur
count	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	77
mean	3001.638353	2018.804376	779.972973	27.558559	55.796654	3699.907336	85
std	3870.201484	2451.113971	929.176190	17.640364	19.804778	4850.420531	152
min	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	
25%	776.000000	604.000000	242.000000	15.000000	41.000000	992.000000	9
50%	1558.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	35
75%	3624.000000	2424.000000	902.000000	35.000000	69.000000	4005.000000	96
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	2183

In [12]:

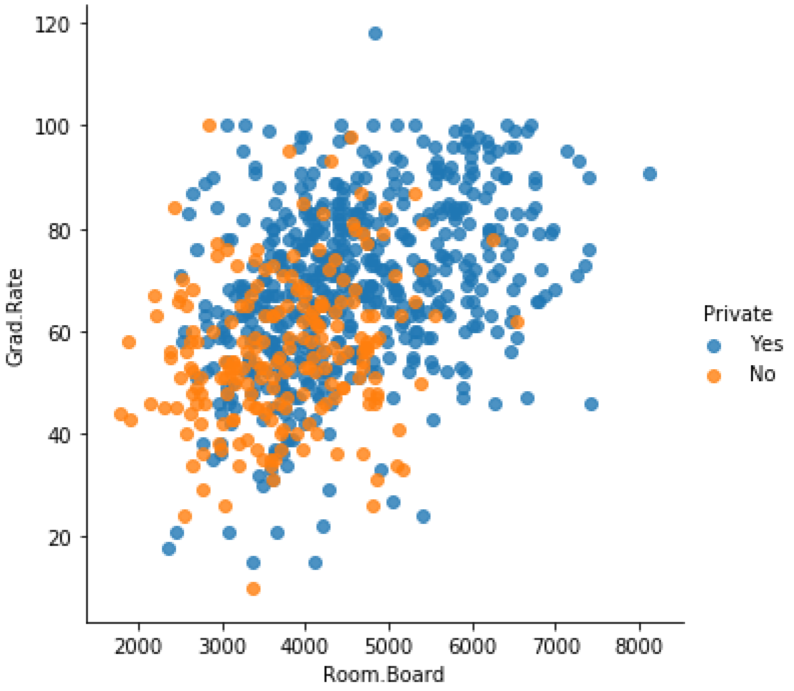
```
#Lets do some exploratory data analysis visualitions
```

In [14]:

```
#Scatter Plot with a hue of Private : Yes or No
#Just used a lmplot plot and got rid of fit_reg to plot something simi;ar as scatter pl
ot of matplotlib
sns.lmplot(x='Room.Board',y='Grad.Rate',data=univ_df,hue='Private',fit_reg=False)
```

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x2a45837dbc8>
```



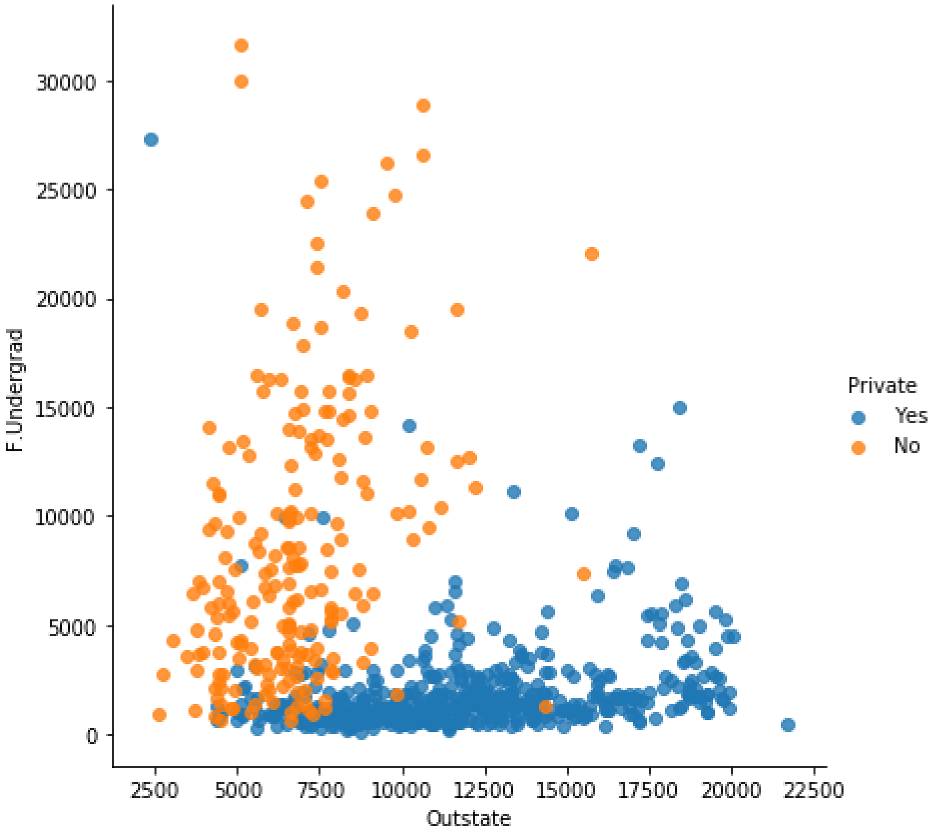
In [16]:

```
#A scatterplot of F.Undergrad versus Outstate where the points are colored by the Private column.
sns.lmplot(x='Outstate',y='F.Undergrad',data=univ_df,hue='Private',fit_reg=False,size=6,aspect=1)
```

```
C:\Users\anike\anaconda3\lib\site-packages\seaborn\regression.py:574: User
Warning: The `size` parameter has been renamed to `height`; please update
your code.
  warnings.warn(msg, UserWarning)
```

Out[16]:

```
<seaborn.axisgrid.FacetGrid at 0x2a458c007c8>
```



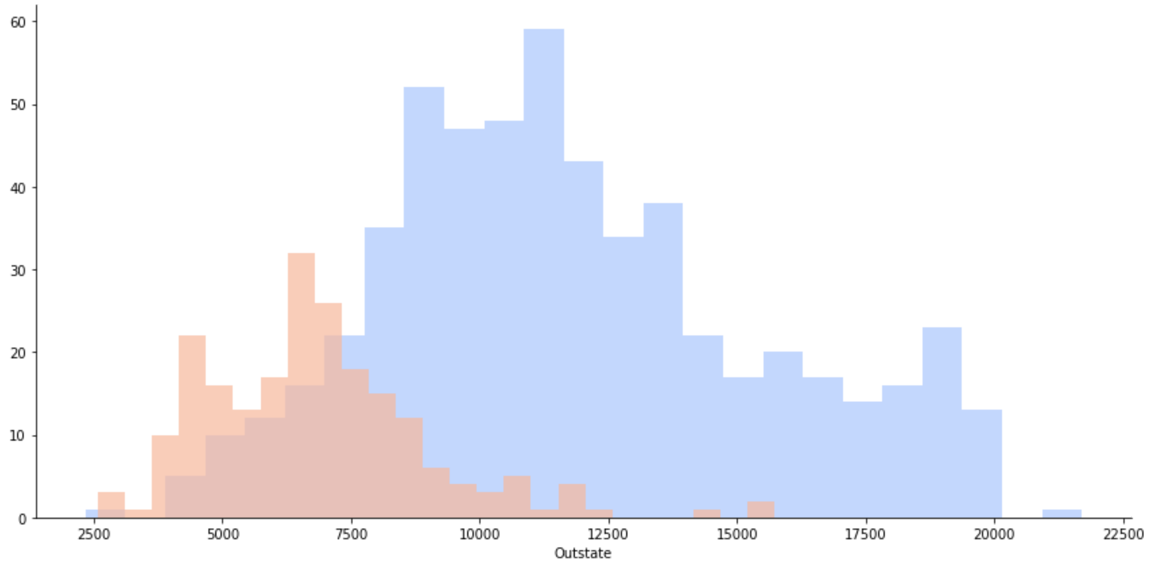
In [17]:

```
#We can already tell that tuition for private schools is way higher
```

In [19]:

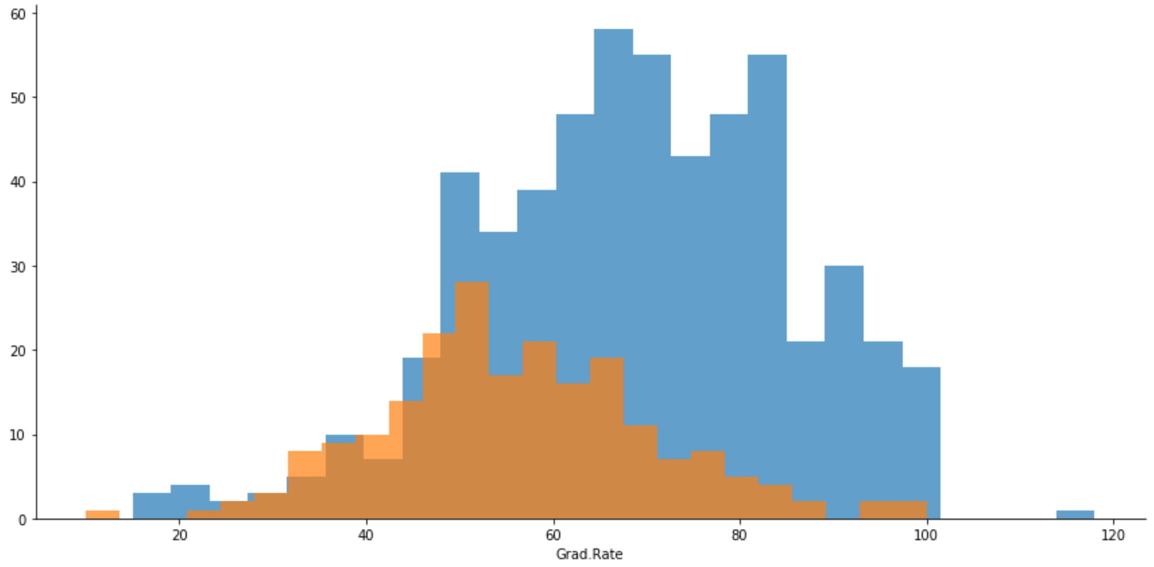
```
#A stacked histogram showing Out of State Tuition based on the Private column.
g=sns.FacetGrid(univ_df,hue='Private',palette='coolwarm',size=6,aspect=2)
g=g.map(plt.hist,'Outstate',bins=25,alpha=0.7)
```

```
C:\Users\anike\anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWa
rning: The `size` parameter has been renamed to `height`; please update yo
ur code.
  warnings.warn(msg, UserWarning)
```



In [20]:

```
g=sns.FacetGrid(univ_df,hue='Private',size=6,aspect=2)
g=g.map(plt.hist,'Grad.Rate',bins=25,alpha=0.7)
```



In [21]:

```
#Seems to be a private school with grad rate greater than 100, that
#isnt possible is it?? Lets find out
univ_df[univ_df['Grad.Rate']>100]
```

Out[21]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad
Cazenovia College	Yes	3847	3433	527	9	35	1010	12

In [23]:

```
#Lets fix this nonsensical number
univ_df['Grad.Rate']['Cazenovia College']=100
```

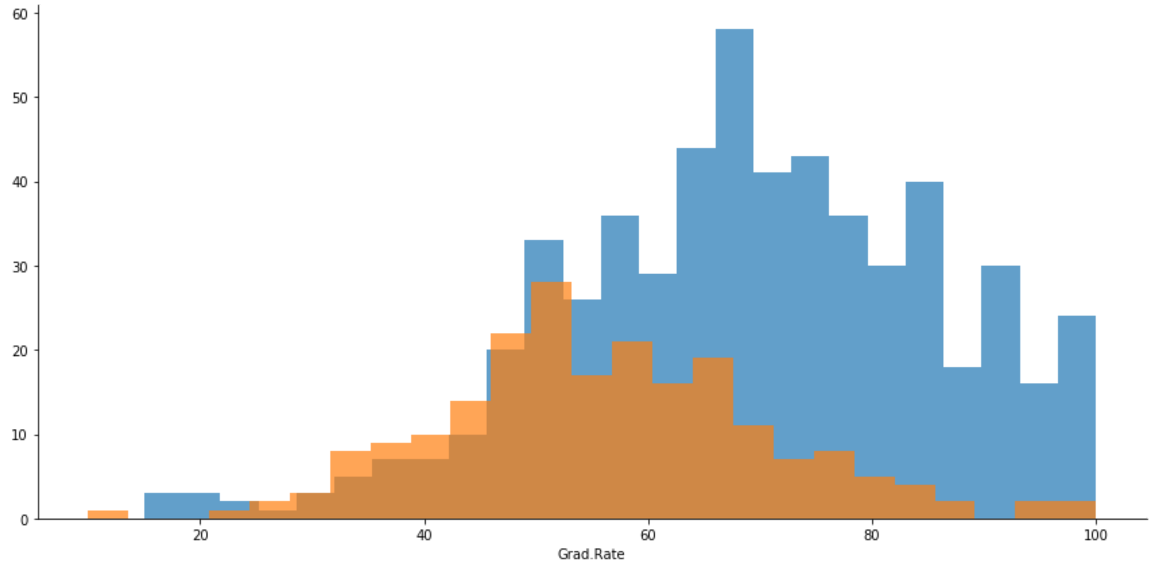
```
C:\Users\anike\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [24]:

```
g=sns.FacetGrid(univ_df,hue='Private',size=6,aspect=2)
g=g.map(plt.hist,'Grad.Rate',bins=25,alpha=0.7)
```

```
C:\Users\anike\anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWa
rning: The `size` parameter has been renamed to `height`; please update yo
ur code.
  warnings.warn(msg, UserWarning)
```



In [25]:

```
#FIXED
```

In [26]:

```
#TIME to create the cluster labels
```

In [27]:

```
from sklearn.cluster import KMeans
```

In [28]:

```
kmeans=KMeans(n_clusters=2)
```

In [30]:

```
kmeans.fit(univ_df.drop('Private',axis=1))
```

Out[30]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [32]:

```
kmeans.cluster_centers_
#dimensions are similar to the number of features on the data set
```

Out[32]:

```
array([[1.81323468e+03, 1.28716592e+03, 4.91044843e+02, 2.53094170e+01,
        5.34708520e+01, 2.18854858e+03, 5.95458894e+02, 1.03957085e+04,
        4.31136472e+03, 5.41982063e+02, 1.28033632e+03, 7.04424514e+01,
        7.78251121e+01, 1.40997010e+01, 2.31748879e+01, 8.93204634e+03,
        6.50926756e+01],
       [1.03631389e+04, 6.55089815e+03, 2.56972222e+03, 4.14907407e+01,
        7.02037037e+01, 1.30619352e+04, 2.46486111e+03, 1.07191759e+04,
        4.64347222e+03, 5.95212963e+02, 1.71420370e+03, 8.63981481e+01,
        9.13333333e+01, 1.40277778e+01, 2.00740741e+01, 1.41705000e+04,
        6.75925926e+01]])
```

In [34]:

```
#There is no perfect way to evaluate clustering if you don't have the labels,we do have
the labels,
#so we take advantage of this to evaluate our clusters, keep in mind,
#you usually won't have this luxury in the real world.

# Creating a new column for df called 'Cluster', which is a 1 for a Private school, and
a 0 for a public school.
```

In [35]:

```
#just converting from yes or no strings to  0 or 1 values
def convert(private):
    if(private=='Yes'):
        return 1
    else:
        return 0
```

In [37]:

```
univ_df['Cluster']=univ_df['Private'].apply(convert)
```

In [38]:

```
univ_df.head()
```

Out[38]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537
Adelphi University	Yes	2186	1924	512	16	29	2683	1227
Adrian College	Yes	1428	1097	336	22	50	1036	99
Agnes Scott College	Yes	417	349	137	60	89	510	63
Alaska Pacific University	Yes	193	146	55	16	44	249	869

In [39]:

```
from sklearn.metrics import confusion_matrix,classification_report
```

In [41]:

```
print(confusion_matrix(univ_df['Cluster'],kmeans.labels_))
print('\n')
print(classification_report(univ_df['Cluster'],kmeans.labels_))
```

```
[[138  74]
 [531  34]]


              precision    recall  f1-score   support

           0       0.21      0.65      0.31       212
           1       0.31      0.06      0.10       565

    accuracy                           0.22       777
   macro avg       0.26      0.36      0.21       777
weighted avg       0.29      0.22      0.16       777
```

In [ ]: