In [ ]:

```
# Name: Meet Hiteshkumar Trivedi
# Student Id: N01520331
```

```
# Lab 7
```

Imports

Import Libraries

In [5]:

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

Import Dataset

Datafile Name: Enrollment Forecast

Number of cases: 29 Variable Names:

- 1.YEAR: 1961 = 1, 1989 = 29
- 2.ROLL: Fall undergraduate enrollment
- 3.UNEM: January unemployment rate (%) for New Mexico
- 4.HGRAD: Spring high schoolgraduates in New Mexico
- 5.INC: Per capita income in Albuquerque (1961 dollars)

In [8]:

```
data = pd.read_csv('E:\Programming\Humber college\Humber Sem 2\Data Analytics\Week-9\enroll
```

Out[8]:

	year	roll	unem	hgrad	inc
0	1	5501	8.1	9552	1923
1	2	5945	7.0	9680	1961
2	3	6629	7.3	9731	1979
3	4	7556	7.5	11666	2030
4	5	8716	7.0	14675	2112

Check the head of customers, and check out its info() and describe() methods.

In [11]:

```
data.head()
```

Out[11]:

	year	roll	unem	hgrad	inc
0	1	5501	8.1	9552	1923
1	2	5945	7.0	9680	1961
2	3	6629	7.3	9731	1979
3	4	7556	7.5	11666	2030
4	5	8716	7.0	14675	2112

In [10]:

```
data.describe()
```

Out[10]:

	year	roll	unem	hgrad	inc
count	29.000000	29.000000	29.000000	29.000000	29.000000
mean	15.000000	12707.034483	7.717241	16528.137931	2729.482759
std	8.514693	3254.076987	1.123155	2926.926676	461.429194
min	1.000000	5501.000000	5.700000	9552.000000	1923.000000
25%	8.000000	10167.000000	7.000000	15723.000000	2351.000000
50%	15.000000	14395.000000	7.500000	17203.000000	2863.000000
75%	22.000000	14969.000000	8.200000	18266.000000	3127.000000
max	29.000000	16081.000000	10.100000	19800.000000	3345.000000

In [12]:

```
data.info()
```

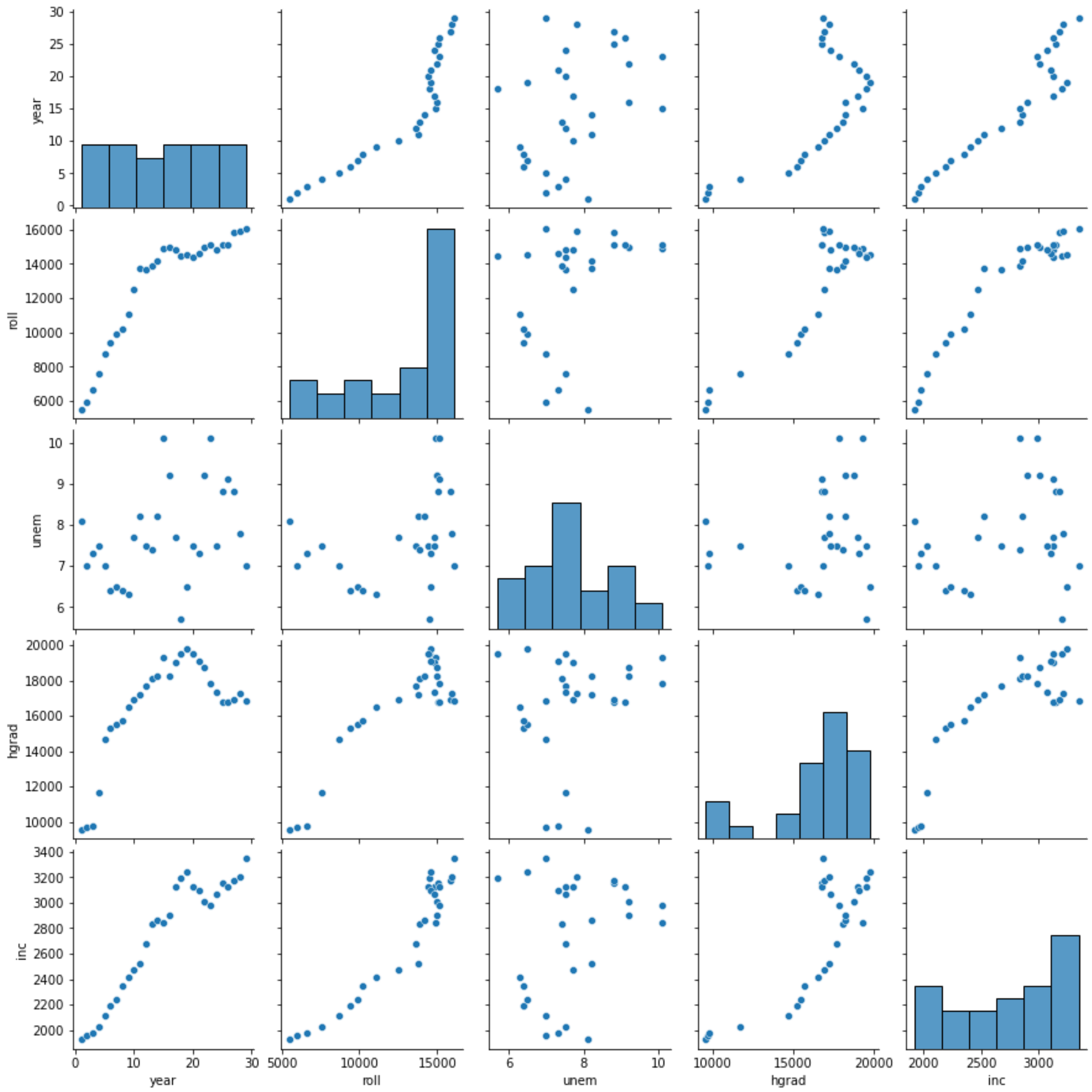
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29 entries, 0 to 28
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   year    29 non-null     int64  
 1   roll    29 non-null     int64  
 2   unem    29 non-null     float64
 3   hgrad   29 non-null     int64  
 4   inc     29 non-null     int64  
dtypes: float64(1), int64(4)
memory usage: 1.3 KB
```

In [13]:

```
sns.pairplot(data)
```

Out[13]:

```
<seaborn.axisgrid.PairGrid at 0x236ff030850>
```



Apply Training and Testing algorithm

X equal to the numerical features of the customers and a variable y equal to the "roll" column.

In [14]:

```
x = data[["year","unem","hgrad", "inc"]]
y= data["roll"]
```

In [29]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)
```

In [30]:

```
from sklearn.linear_model import LinearRegression
```

In [31]:

```
lm = LinearRegression()
lm.fit(X_train,y_train)
```

Out[31]:

```
LinearRegression()
```

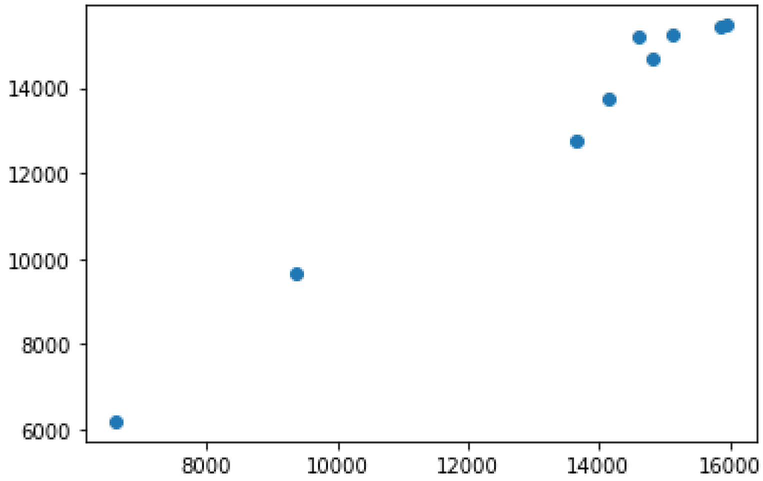
Predicting Test Data

In [32]:

```
predictions = lm.predict(X_test)
plt.scatter(y_test,predictions)
```

Out[32]:

```
<matplotlib.collections.PathCollection at 0x2368209cc40>
```



Evaluating the Model

Let's evaluate our model performance by calculating the residual sum of squares and the explained variance score (R^2).

** Calculate the Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error. Refer to the lecture or to Wikipedia for the formulas**

In [33]:

```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 424.16754520139125
MSE: 233686.20425095654
RMSE: 483.4110096501284
```

In [ ]: