Linear Regression - Project Exercise

December 9, 2017

In [100]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [101]:
```
customers = pd.read_csv('Ecommerce Customers')
```

In [102]:
```
customers.head()
```

Out[102]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.577668	4.082621	587.951054
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.268959	2.664034	392.204933
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.110597	4.104543	487.547505
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	3.120179	581.852344
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	4.446308	599.406092

```
In [103]: customers.describe()
```

```
Out[103]:        Avg. Session Length  Time on App  Time on Website  \
          count           500.000000   500.000000       500.000000
          mean             33.053194    12.052488        37.060445
          std               0.992563     0.994216         1.010489
          min              29.532429     8.508152        33.913847
          25%              32.341822    11.388153        36.349257
          50%              33.082008    11.983231        37.069367
          75%              33.711985    12.753850        37.716432
          max              36.139662    15.126994        40.005182

                 Length of Membership  Yearly Amount Spent
          count            500.000000           500.000000
          mean               3.533462           499.314038
          std                0.999278            79.314782
          min                0.269901           256.670582
          25%                2.930450           445.038277
          50%                3.533975           498.887875
          75%                4.126502           549.313828
          max                6.922689           765.518462
```

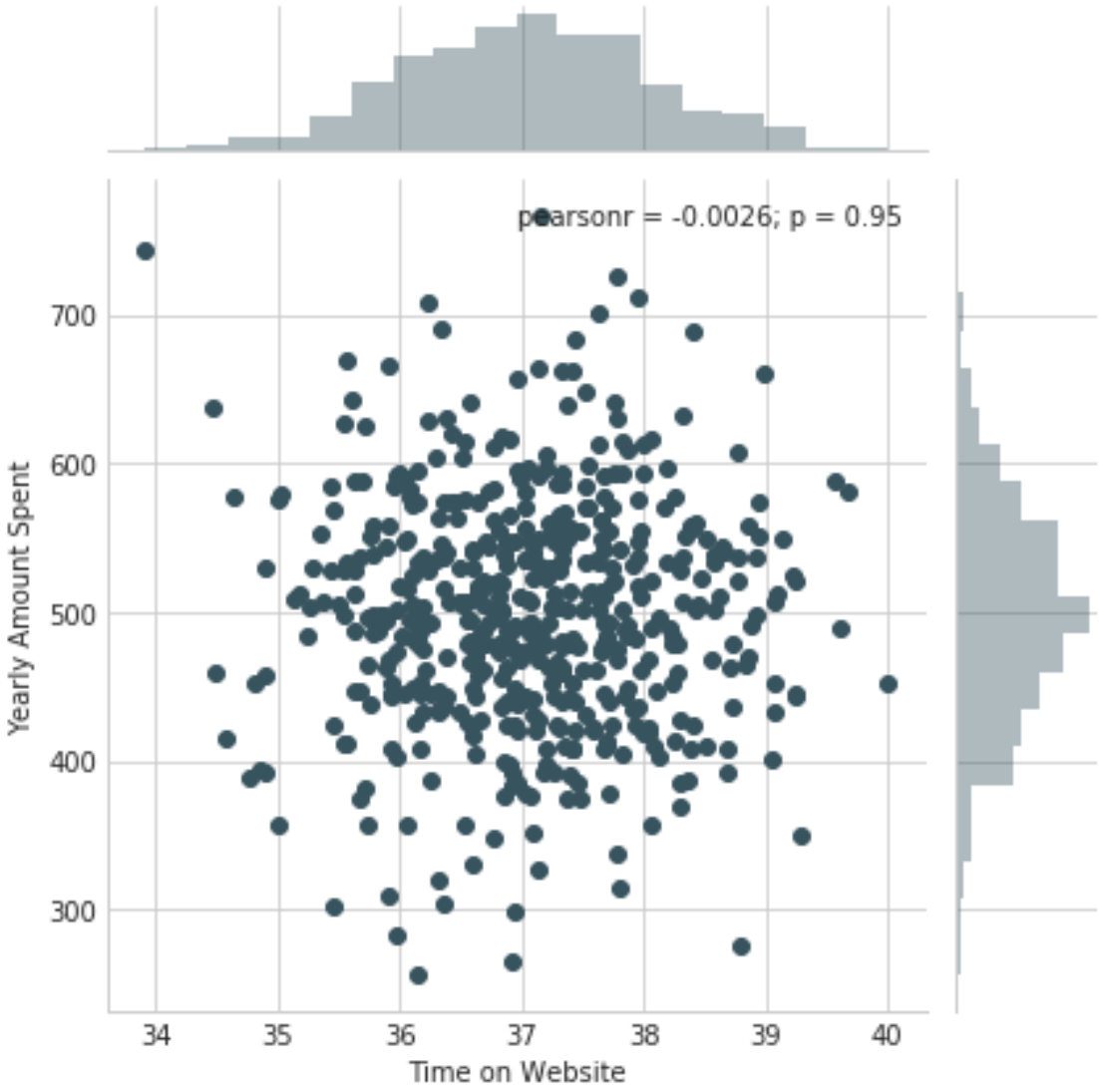
```
In [104]: customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
Email                   500 non-null object
Address                 500 non-null object
Avatar                  500 non-null object
Avg. Session Length     500 non-null float64
Time on App             500 non-null float64
Time on Website         500 non-null float64
Length of Membership    500 non-null float64
Yearly Amount Spent     500 non-null float64
dtypes: float64(5), object(3)
memory usage: 31.3+ KB
```

```
In [105]: sns.set_palette("GnBu_d")
          sns.set_style('whitegrid')
```

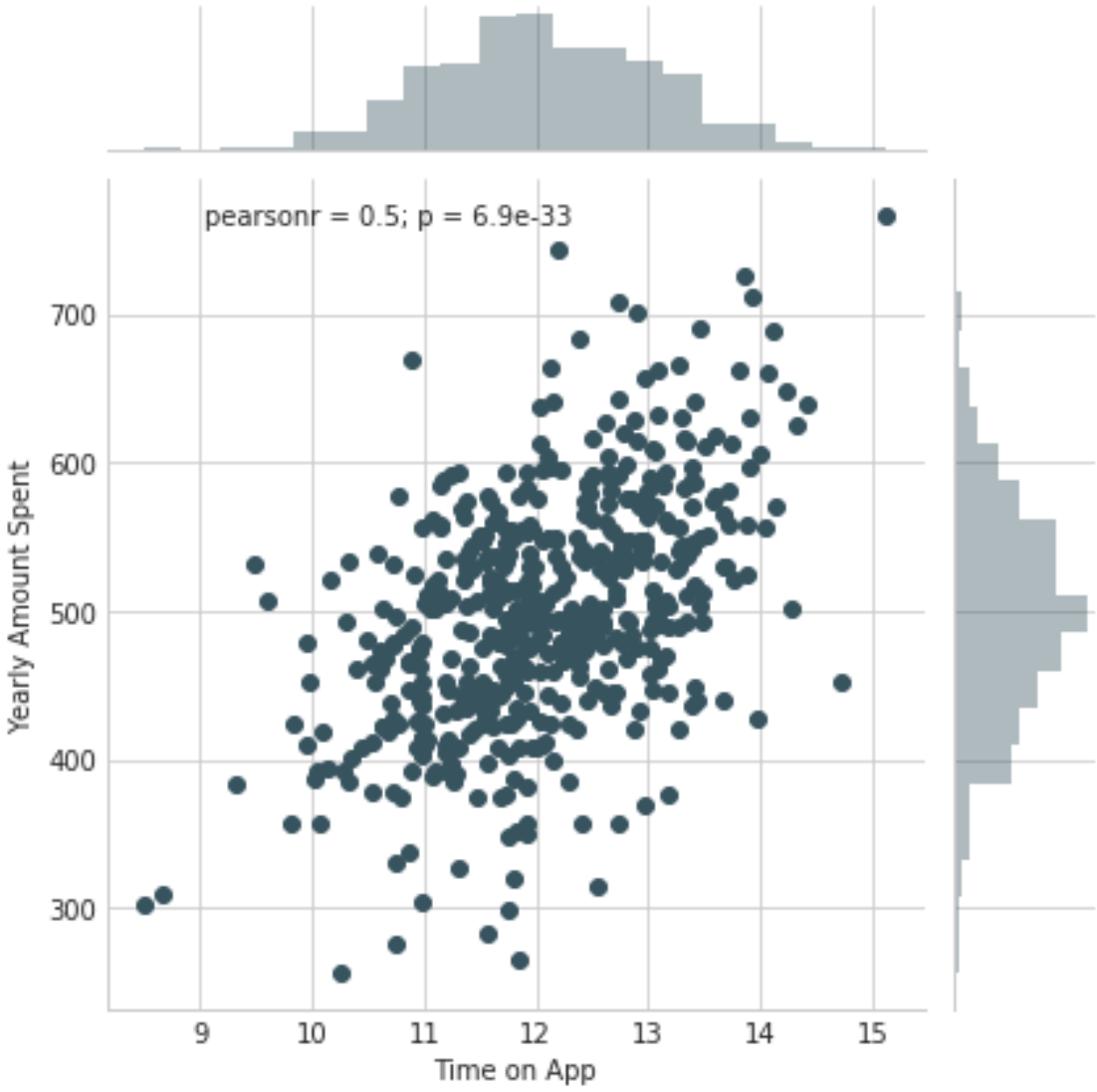
```
In [106]: sns.jointplot('Time on Website' , 'Yearly Amount Spent' , customers)
```

```
Out[106]: <seaborn.axisgrid.JointGrid at 0x7f38d8881810>
```



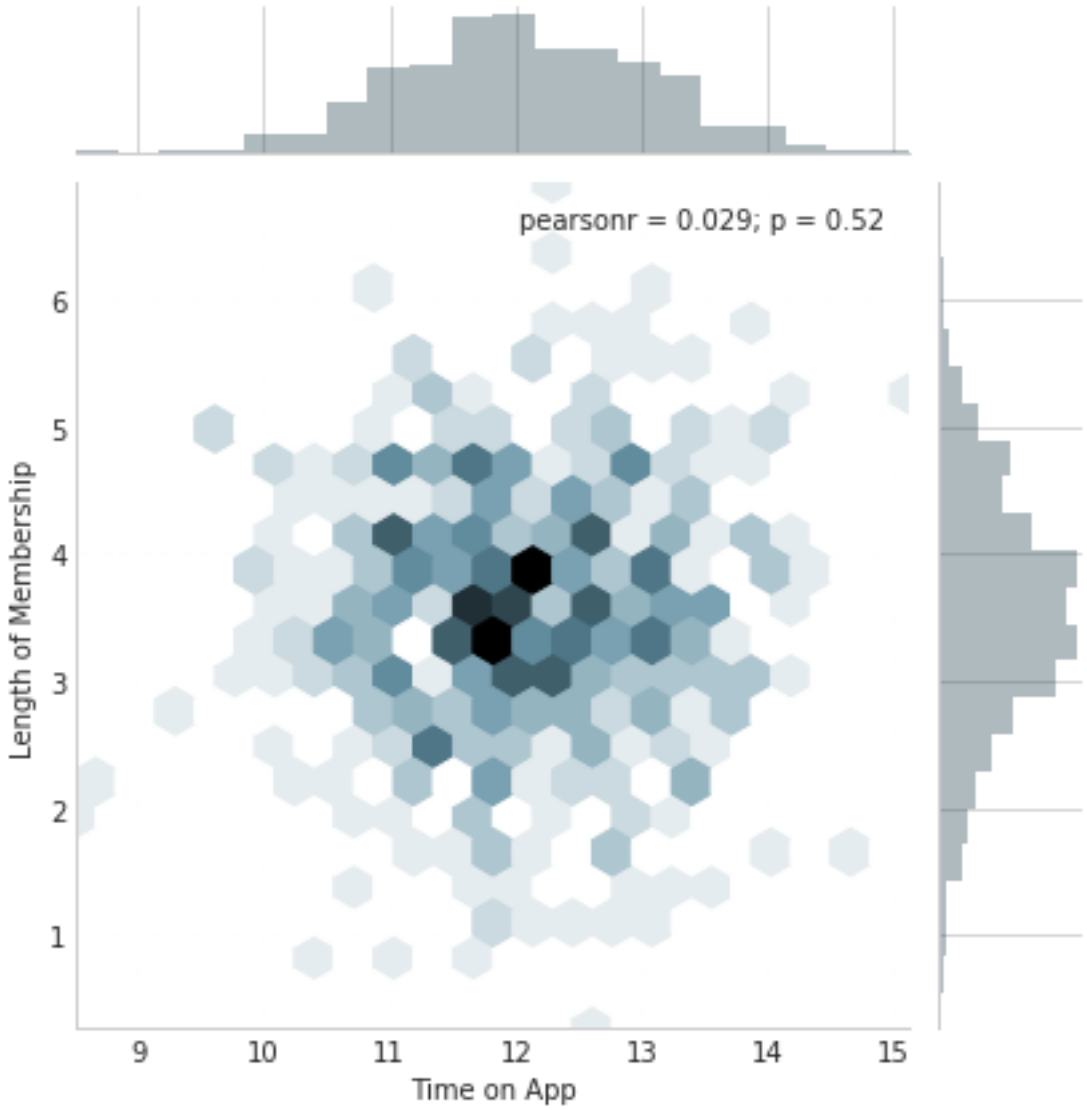
In [107]: sns.jointplot('Time on App' , 'Yearly Amount Spent' , customers)

Out[107]: <seaborn.axisgrid.JointGrid at 0x7f38d885ee90>



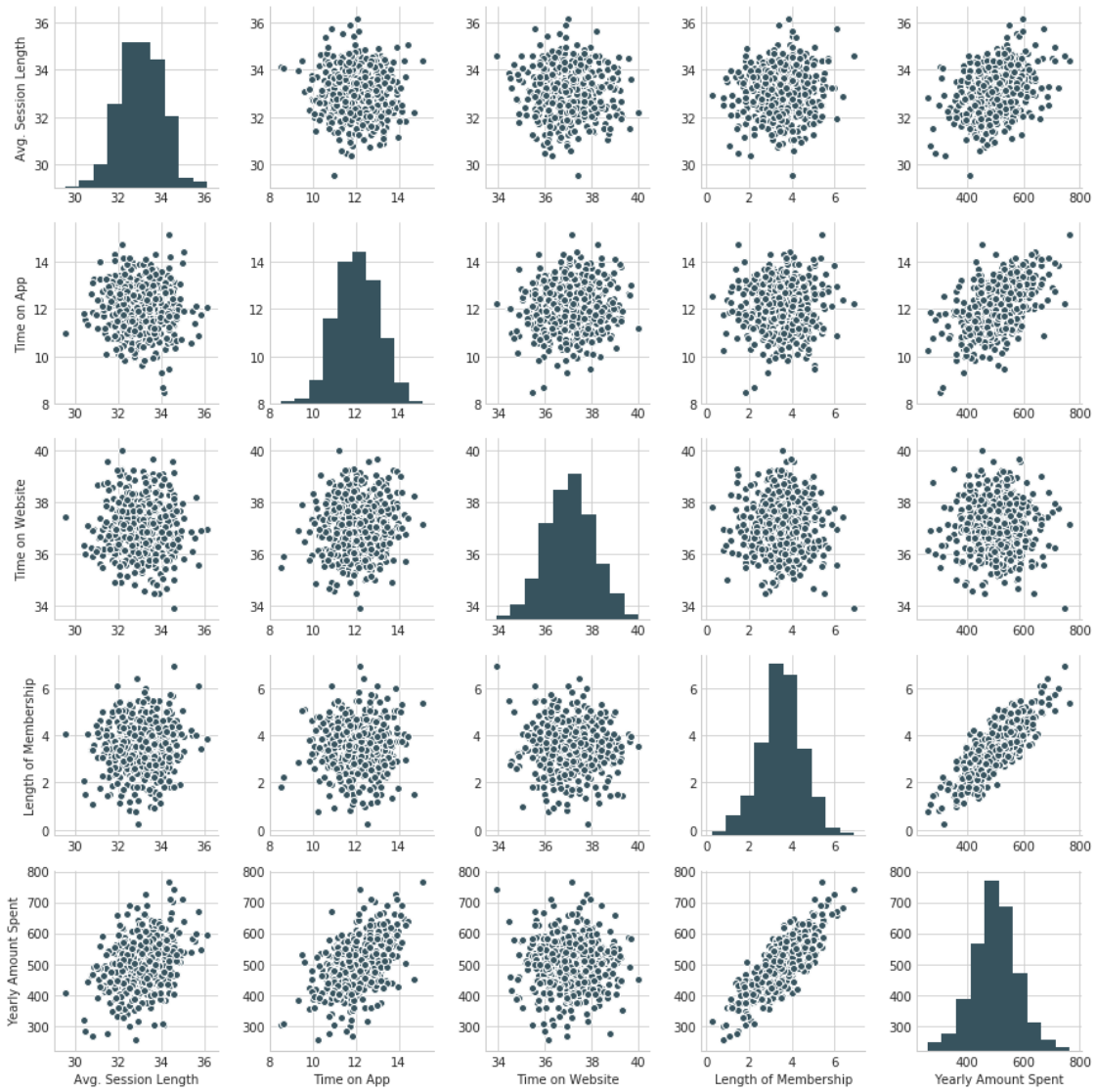

```
In [108]: sns.jointplot('Time on App' , 'Length of Membership' , customers , kind = 'hex')
```

```
Out[108]: <seaborn.axisgrid.JointGrid at 0x7f38d862f290>
```



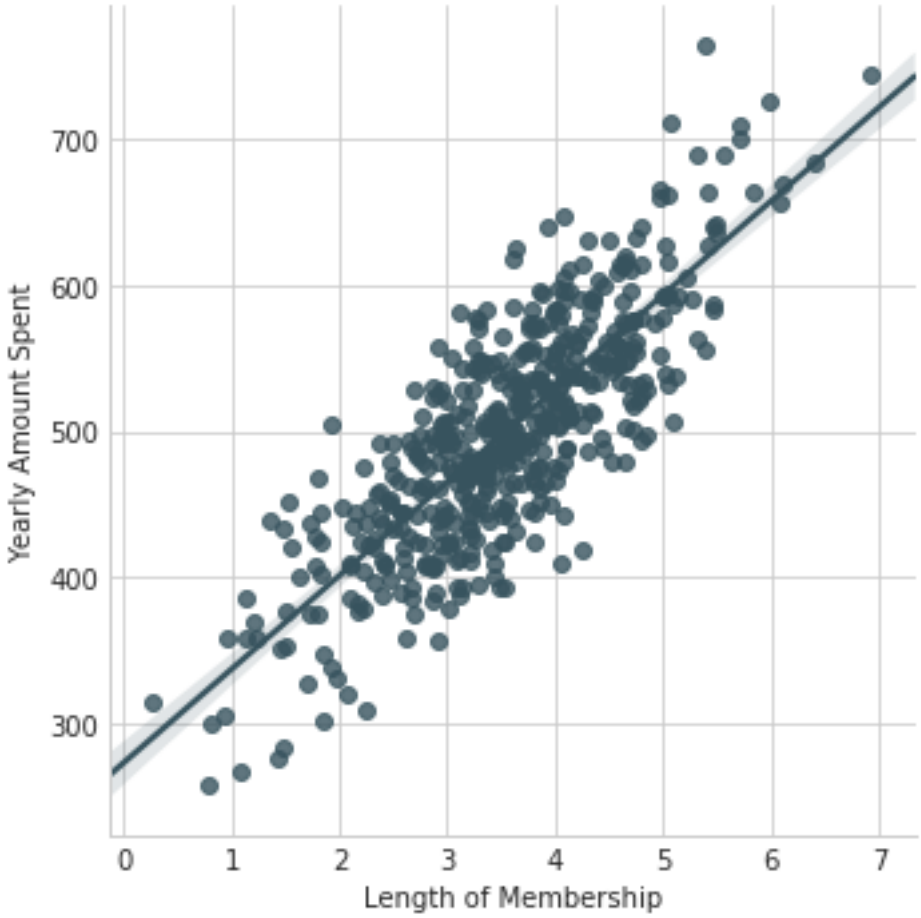

```
In [109]: sns.pairplot(customers)
```

```
Out[109]: <seaborn.axisgrid.PairGrid at 0x7f38d8455090>
```




```
In [110]: sns.lmplot(x= 'Length of Membership', y= 'Yearly Amount Spent', data = customers)
```

```
Out[110]: <seaborn.axisgrid.FacetGrid at 0x7f38d8854950>
```

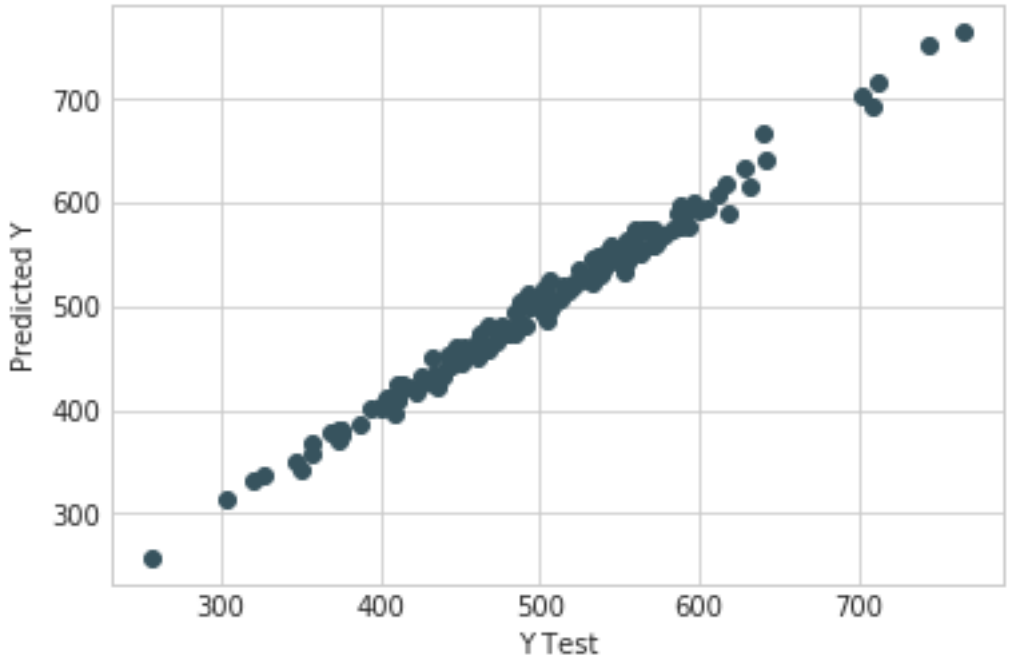



```
In [111]: y = customers['Yearly Amount Spent']

In [112]: X = customers[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Memb

In [113]: from sklearn.model_selection import train_test_split

In [114]: X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state

In [115]: from sklearn.linear_model import LinearRegression

In [116]: lm = LinearRegression()

In [117]: lm.fit(X_train ,y_train)

Out[117]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

In [118]: lm.coef_

Out[118]: array([ 25.98154972,  38.59015875,   0.19040528,  61.27909654])
```

In [119]:
```
predictions = lm.predict(X_test)
```

In [120]:
```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[120]:
```
Text(0,0.5,u'Predicted Y')
```



In [121]:
```
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 :', 7.2281486534308597)
('MSE :', 79.813051650974828)
('RMSE :', 8.9338150669786547)
```

In [122]:
```
lm.coef_
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[122]:

	Coeffecient
Avg. Session Length	25.981550

```
Time on App             38.590159
Time on Website          0.190405
Length of Membership    61.279097
```

In [123]:
```
# Noramlizing the data to reduce the error

from sklearn import preprocessing
customers_new= pd.DataFrame(preprocessing.normalize(customers[['Avg. Session Length',
                                                               ,columns= ['Avg. Session Le
```

In [124]:
```
y = customers_normalized['Yearly Amount Spent']
X = customers_normalized[['Avg. Session Length', 'Time on App','Time on Website', 'Len
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state
lm.fit(X_train ,y_train)

predictions = lm.predict(X_test)
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)))
```

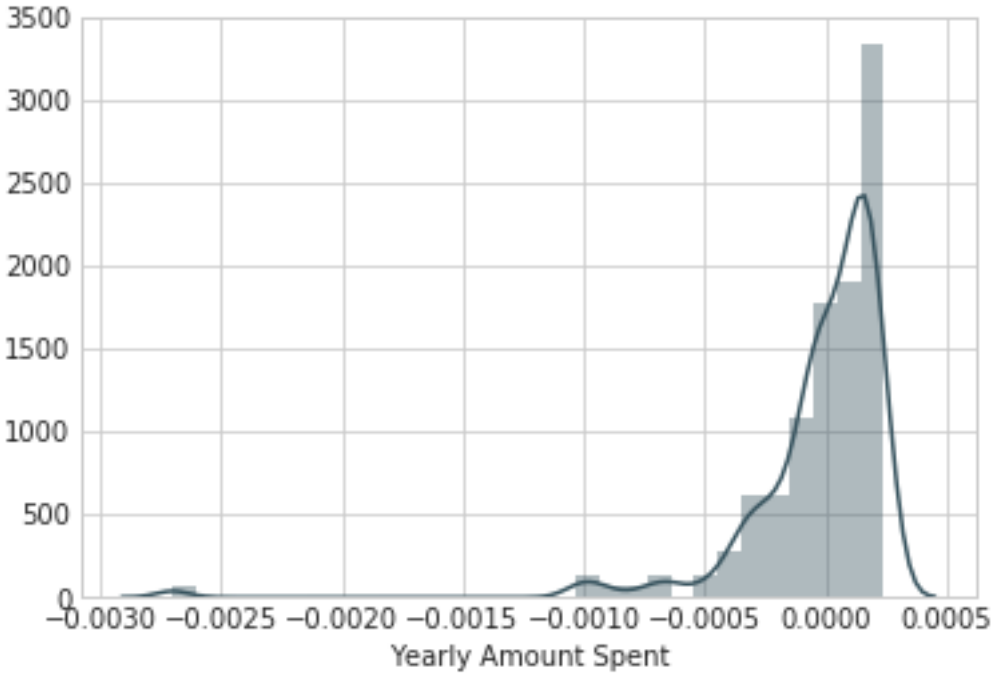
```
('MAE :', 0.00018896340228741944)
('MSE :', 1.0562489642701683e-07)
('RMSE :', 0.00032499984065690988)
```

In [125]:
```
print(lm.coef_)
```

```
[-0.078774   -0.04256498 -0.07913042 -0.02554651]
```

In [126]:
```
sns.distplot((y_test-predictions),bins=30)
```

Out[126]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f38d3520f90>
```



In [127]:
```
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[127]:

	Coeffecient
Avg. Session Length	-0.078774
Time on App	-0.042565
Time on Website	-0.079130
Length of Membership	-0.025547