Linear Regression.Project

October 26, 2016

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```

** Read in the Ecommerce Customers csv file as a DataFrame called customers.** * Avg. Session Length: Average session of in-store style advice sessions. * Time on App: Average time spent on App in minutes * Time on Website: Average time spent on Website in minutes * Length of Membership: How many years the customer has been a member.

```
In [2]: customers = pd.read_csv("Ecommerce Customers")
```

```
In [3]: customers.head()
```

```
Out[3]:                            Email  \
        0      mstephenson@fernandez.com
        1              hduke@hotmail.com
        2               pallen@yahoo.com
        3        riverarebecca@gmail.com
        4  mstephens@davidson-herman.com

                                                     Address            Avatar  \
        0      835 Frank Tunnel\nWrightmouth, MI 82180-9605            Violet
        1    4547 Archer Common\nDiazchester, CA 06566-8576         DarkGreen
        2  24645 Valerie Unions Suite 582\nCobbborough, D...            Bisque
        3   1414 David Throughway\nPort Jason, OH 22070-1220       SaddleBrown
        4  14023 Rodriguez Passage\nPort Jacobville, PR 3...  MediumAquaMarine

           Avg. Session Length  Time on App  Time on Website  Length of Membership  \
        0            34.497268    12.655651        39.577668              4.082621
        1            31.926272    11.109461        37.268959              2.664034
        2            33.000915    11.330278        37.110597              4.104543
        3            34.305557    13.717514        36.721283              3.120179
        4            33.330673    12.795189        37.536653              4.446308

           Yearly Amount Spent
        0           587.951054
        1           392.204933
        2           487.547505
        3           581.852344
        4           599.406092
```

```
In [4]: customers.describe()
```

```
Out[4]:        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 [279]: customers.info()
```

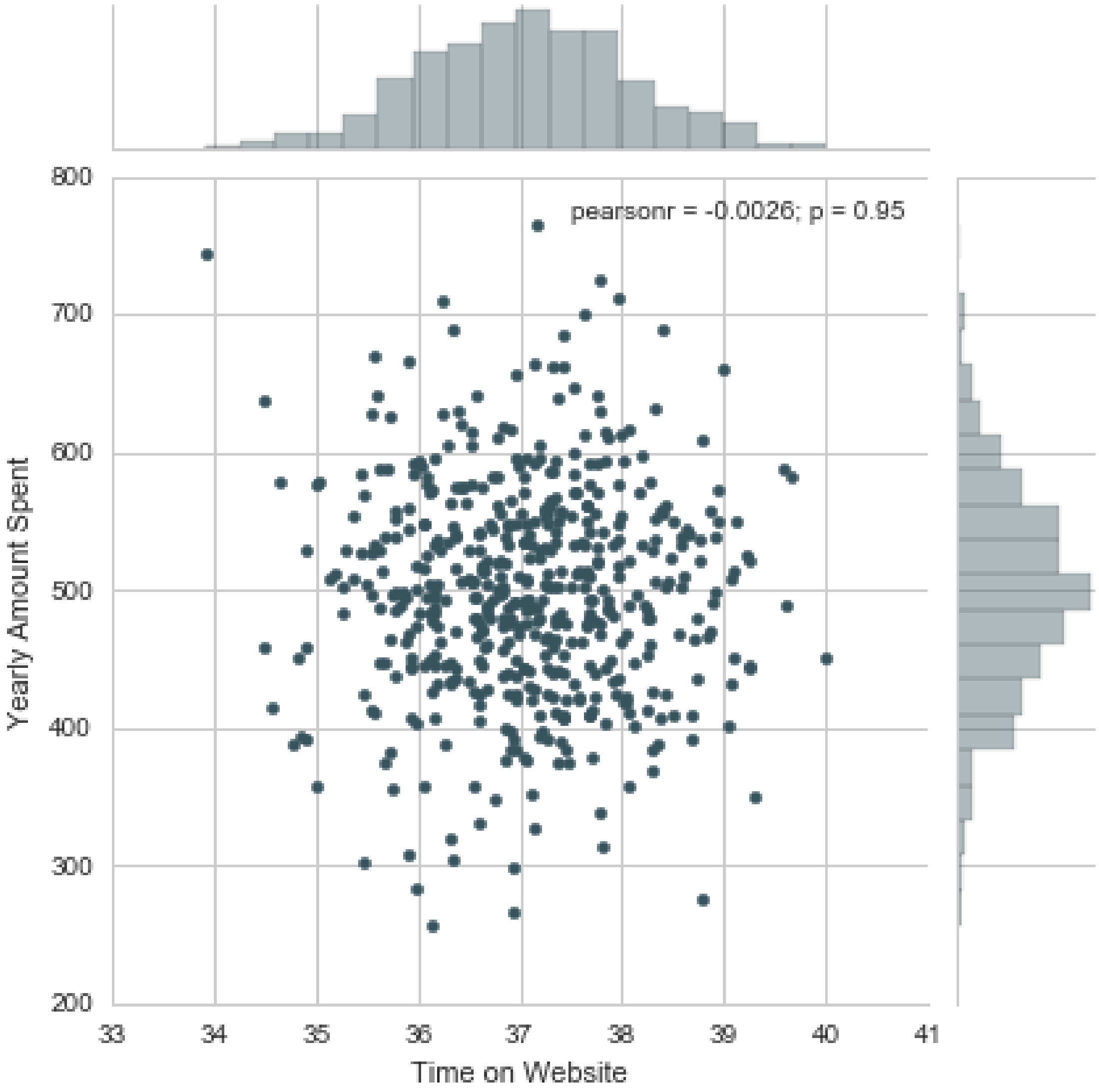
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
Email                   500 non-null object
Address                 500 non-null object
Avatar                  500 non-null object
Avg. Session Length     500 non-null float64
Time on App             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
```

0.1 Exploratory Data Analysis

```
In [5]: sns.set_palette("GnBu_d")
        sns.set_style('whitegrid')
```

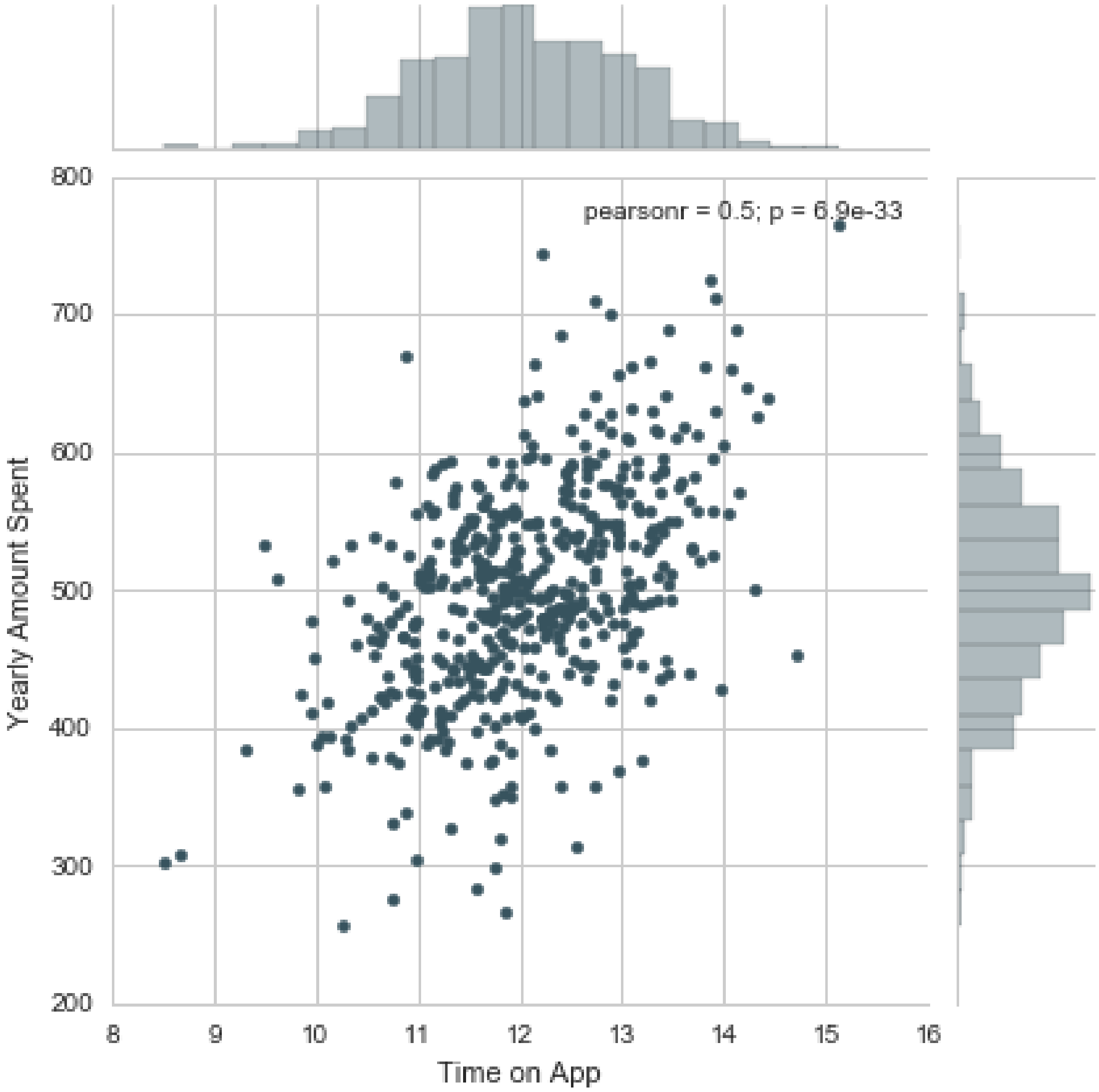
```
In [281]: sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=customers)
```

```
Out[281]: <seaborn.axisgrid.JointGrid at 0x120bfcc88>
```



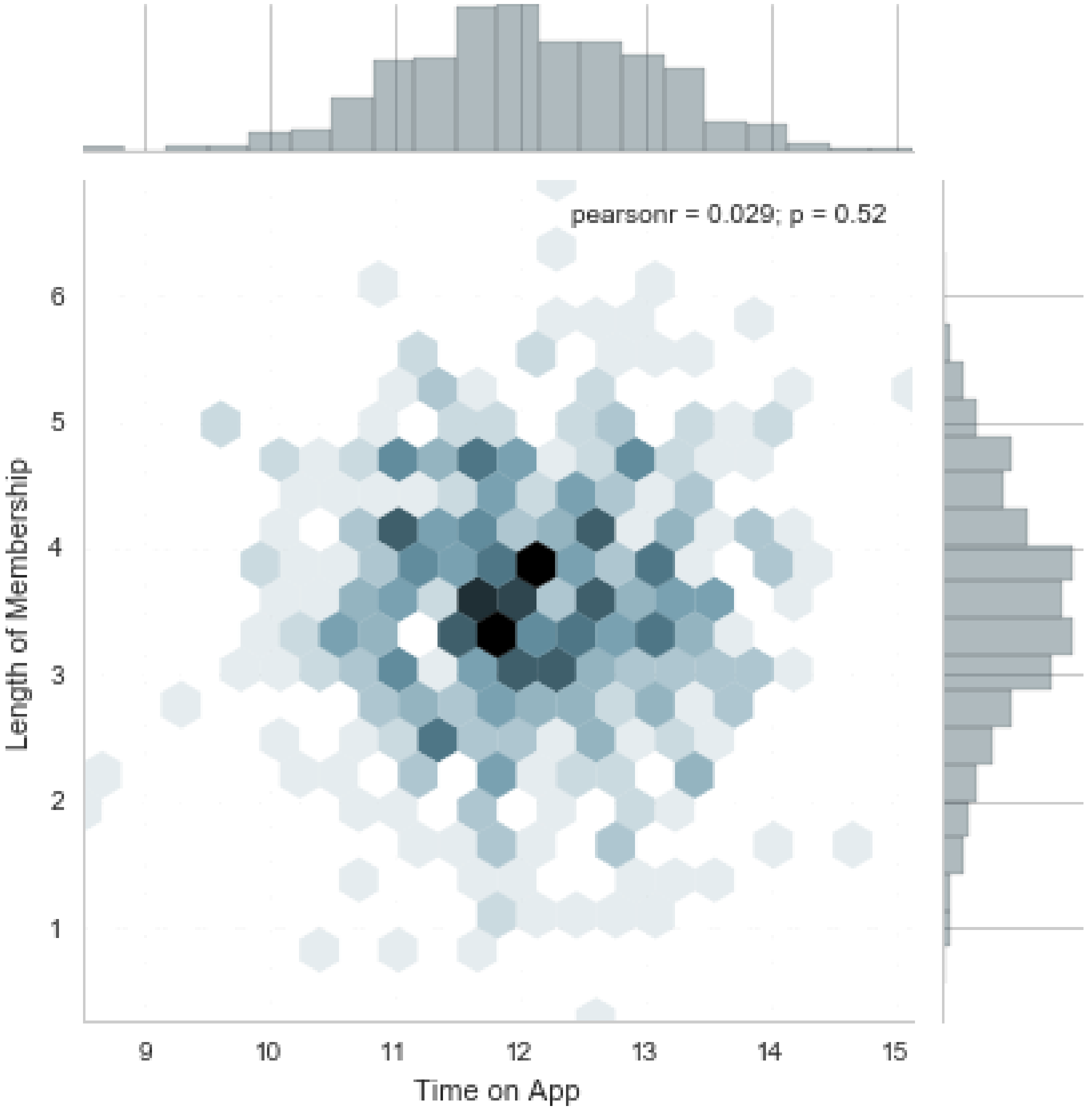
```
In [6]: sns.jointplot(x='Time on App',y='Yearly Amount Spent',data=customers)
```

```
Out[6]: <seaborn.axisgrid.JointGrid at 0x1f4f95f198>
```



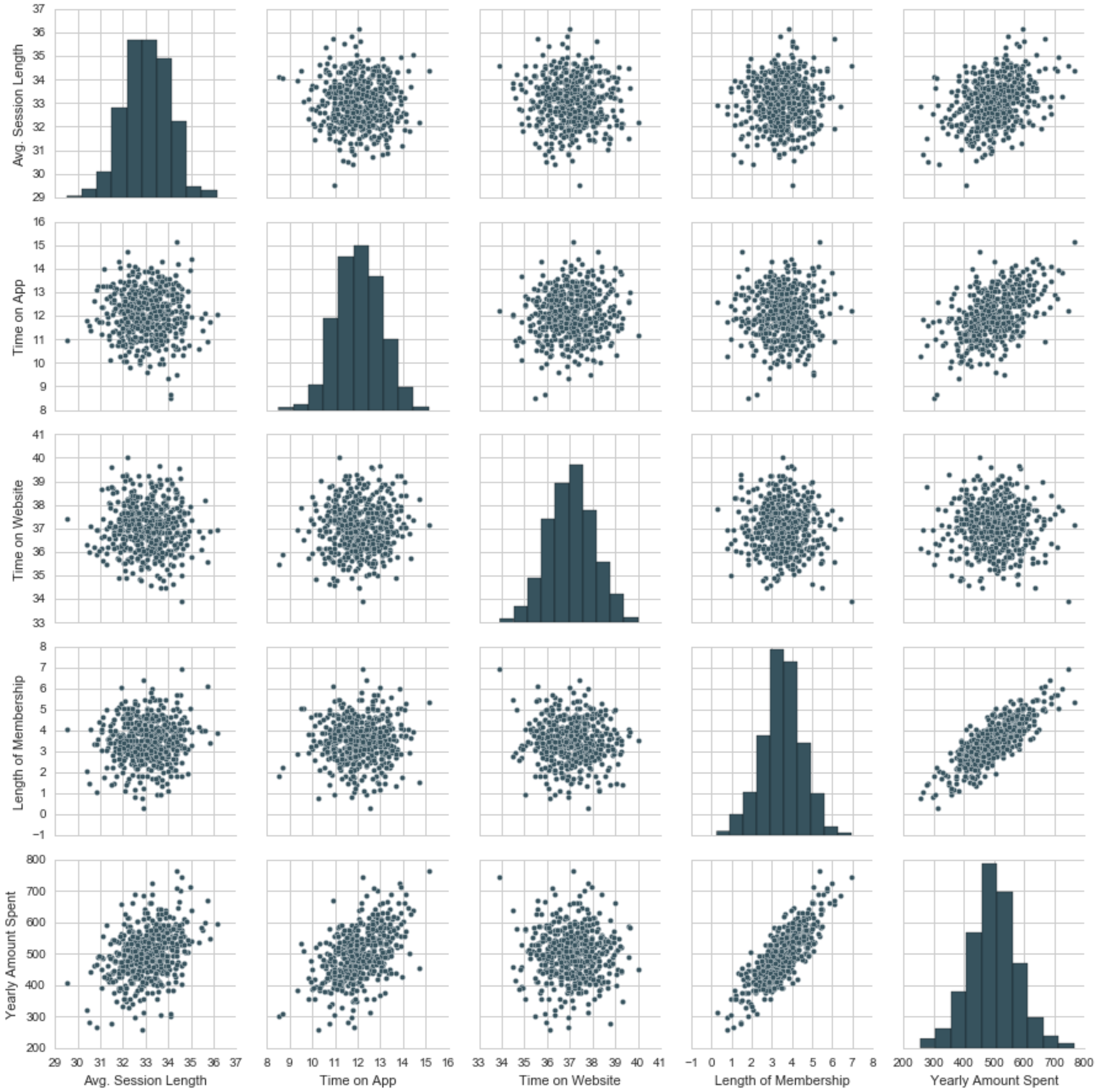
```
In [7]: sns.jointplot(x='Time on App',y='Length of Membership',kind='hex',data=customers)

Out[7]: <seaborn.axisgrid.JointGrid at 0x1f50373c88>
```



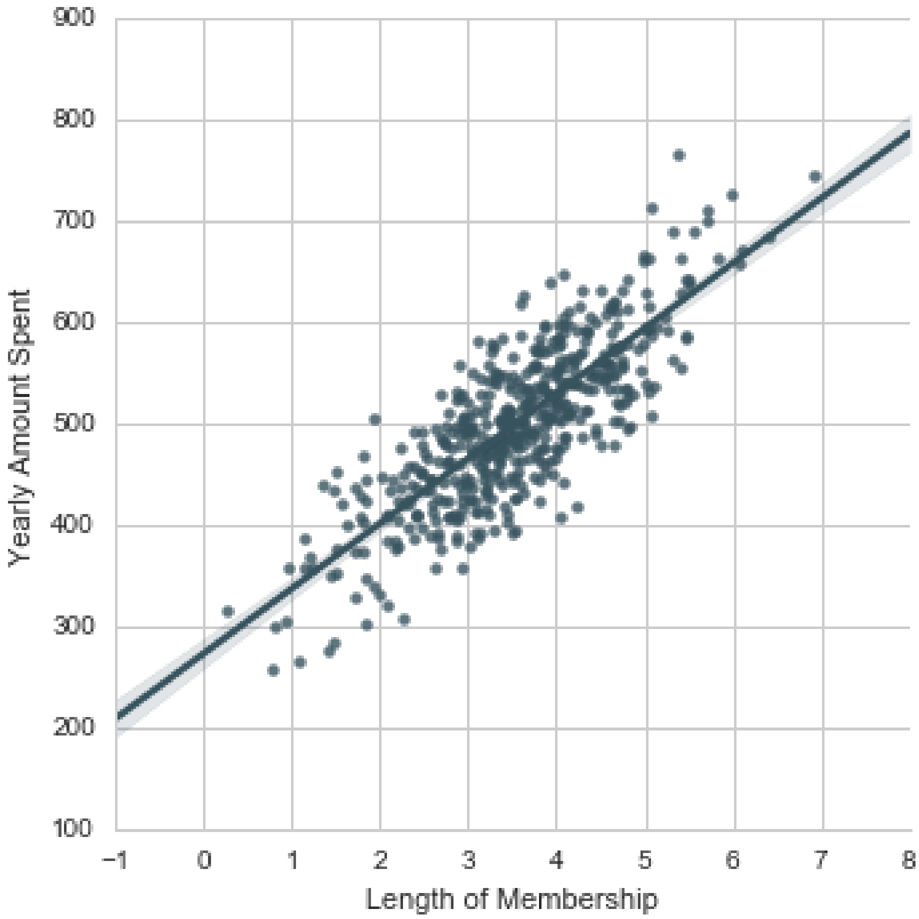
```
In [284]: sns.pairplot(customers)

Out[284]: <seaborn.axisgrid.PairGrid at 0x132fb3da0>
```



```
In [8]: sns.lmplot(x='Length of Membership',y='Yearly Amount Spent',data=customers)

Out[8]: <seaborn.axisgrid.FacetGrid at 0x1f5052aac8>
```



0.2 Training and Testing Data

```
In [9]: y = customers[’Yearly Amount Spent’]
```

```
In [10]: X = customers[[’Avg. Session Length’, ’Time on App’,’Time on Website’, ’Length of Membership’]]
```

** Use cross_validation.train_test_split from sklearn to split the data into training and testing sets. Set test_size=0.3 and random_state=101**

```
In [11]: from sklearn.cross_validation import train_test_split
```

```
In [12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

0.3 Training the Model

```
In [13]: from sklearn.linear_model import LinearRegression
```

```
In [14]: lm = LinearRegression()
```

** Train/fit lm on the training data.**

```
In [15]: lm.fit(X_train,y_train)
```

```
Out[15]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Print out the coefficients of the model

```
In [16]: print('Coefficients: \n', lm.coef_)
```

```
Coefficients: 
 [ 25.98154972  38.59015875   0.19040528  61.27909654]
```

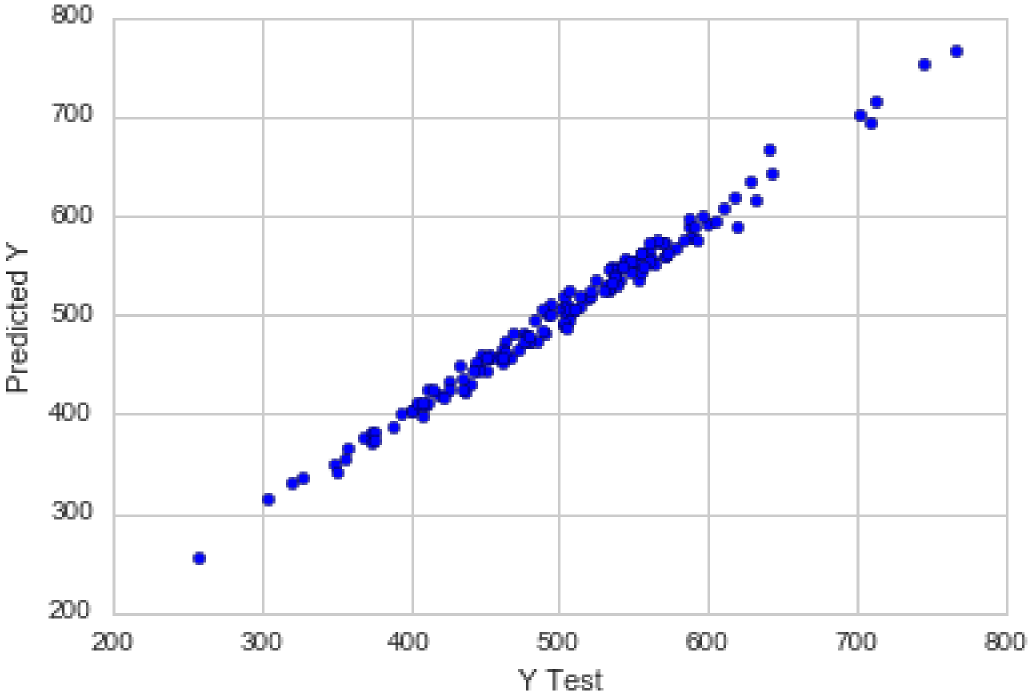
0.4 Predicting Test Data

```
In [17]: predictions = lm.predict( X_test)
```

0.5 Scatter plot

```
In [18]: plt.scatter(y_test,predictions)
         plt.xlabel('Y Test')
         plt.ylabel('Predicted Y')
```

```
Out[18]: <matplotlib.text.Text at 0x1f515fd0b8>
```



0.6 Evaluating the Model

```
In [19]: 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: 7.22814865343
MSE: 79.813051651
RMSE: 8.93381506698
```

```
In [22]: metrics.explained_variance_score(y_test,predictions)
```

```
Out[22]: 0.98907712318896057
```

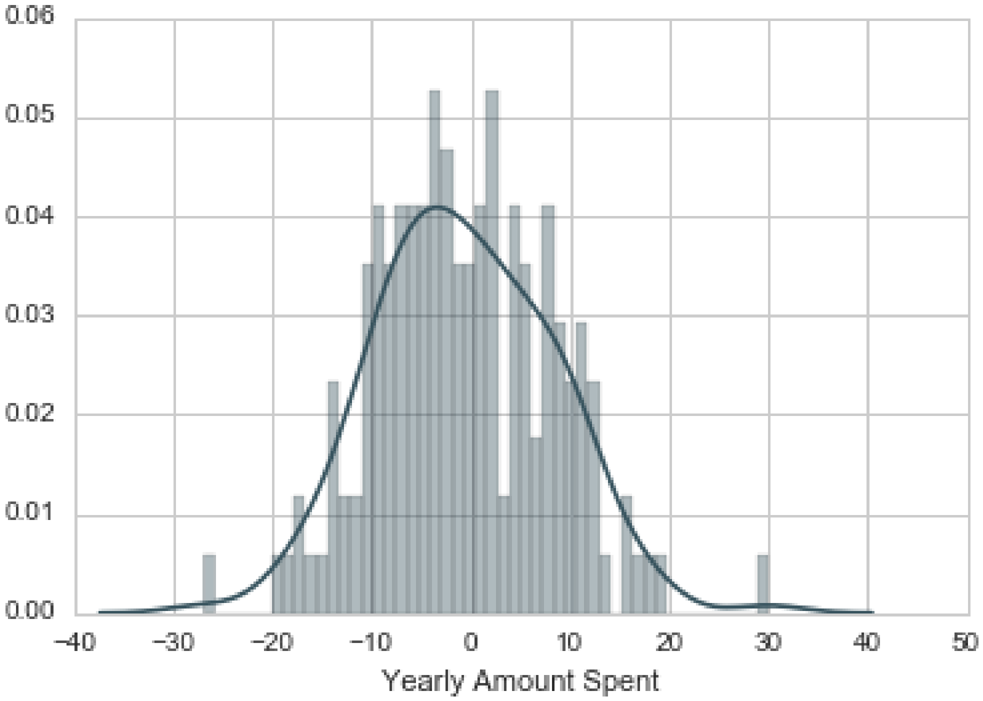
```
In [23]: lm.score(X,y)
```

```
Out[23]: 0.98427271423360208
```

0.7 Residuals

Plot a histogram of the residuals and make sure it looks normally distributed. Use either seaborn distplot, or just plt.hist().

```
In [317]: sns.distplot((y_test-predictions),bins=50);
```



0.8 Conclusion

** Recreate the dataframe below. **

```
In [298]: coeffecients = pd.DataFrame(lm.coef_,X.columns)
          coeffecients.columns = ['Coeffecient']
          coeffecients
```

```
Out[298]:                       Coeffecient
          Avg. Session Length     25.981550
          Time on App             38.590159
          Time on Website          0.190405
          Length of Membership    61.279097
```

```
In [24]: lm.score(X,y)
```

```
Out[24]: 0.98427271423360208
```

Interpreting the coefficients:

- Holding all other features fixed, a 1 unit increase in **Avg. Session Length** is associated with an **increase of 25.98 total dollars spent**.
- Holding all other features fixed, a 1 unit increase in **Time on App** is associated with an **increase of 38.59 total dollars spent**.
- Holding all other features fixed, a 1 unit increase in **Time on Website** is associated with an **increase of 0.19 total dollars spent**.
- Holding all other features fixed, a 1 unit increase in **Length of Membership** is associated with an **increase of 61.27 total dollars spent**.