Linear Regression Project

It is a data of Ecommerce company in New York City that sells clothing online but they also have in-store style and clothing advice sessions. Customers come in to the store, have sessions/meetings with a personal stylist, then they can go home and order either on a mobile app or website for the clothes they want.

The company is trying to decide whether to focus their efforts on their mobile app experience or their website.

Imports

In [1]:

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

Data

- **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')
```

Details. (Data check karlo)

In [3]:

```
customers.head()
```

Out[3]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.577668	4.082621	58
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.268959	2.664034	39
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.110597	4.104543	48
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	3.120179	58
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	4.446308	59

In [4]:

```
customers.describe()
```

Out[4]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
25%	32.341822	11.388153	36.349257	2.930450	445.038277
50%	33.082008	11.983231	37.069367	3.533975	498.887875
75%	33.711985	12.753850	37.716432	4.126502	549.313828
max	36.139662	15.126994	40.005182	6.922689	765.518462

In [5]:

```
customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    object 
 2   Avatar                500 non-null    object 
 3   Avg. Session Length   500 non-null    float64
 4   Time on App           500 non-null    float64
 5   Time on Website       500 non-null    float64
 6   Length of Membership  500 non-null    float64
 7   Yearly Amount Spent   500 non-null    float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB
```

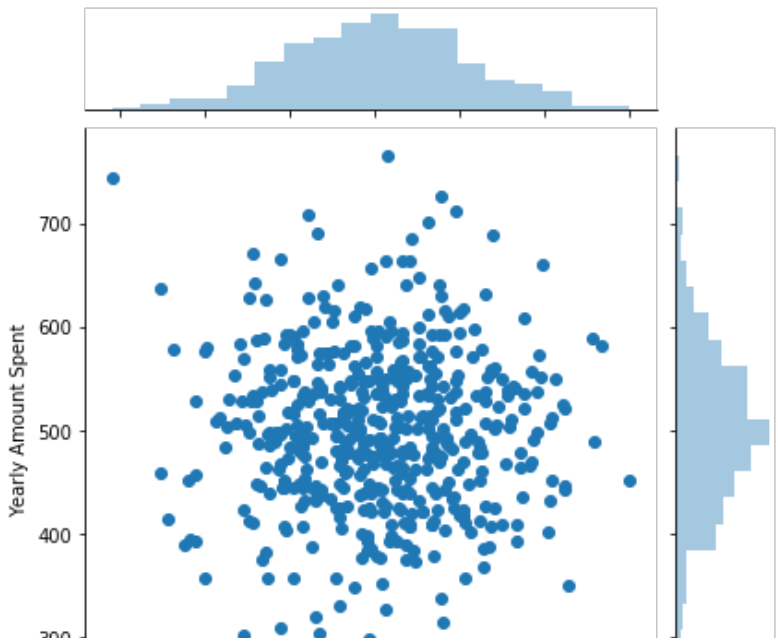
Data Analysis

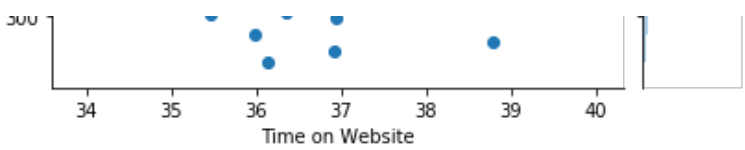
In [6]:

```
sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=customers)
```

Out[6]:

```
<seaborn.axisgrid.JointGrid at 0x7f36607a1950>
```



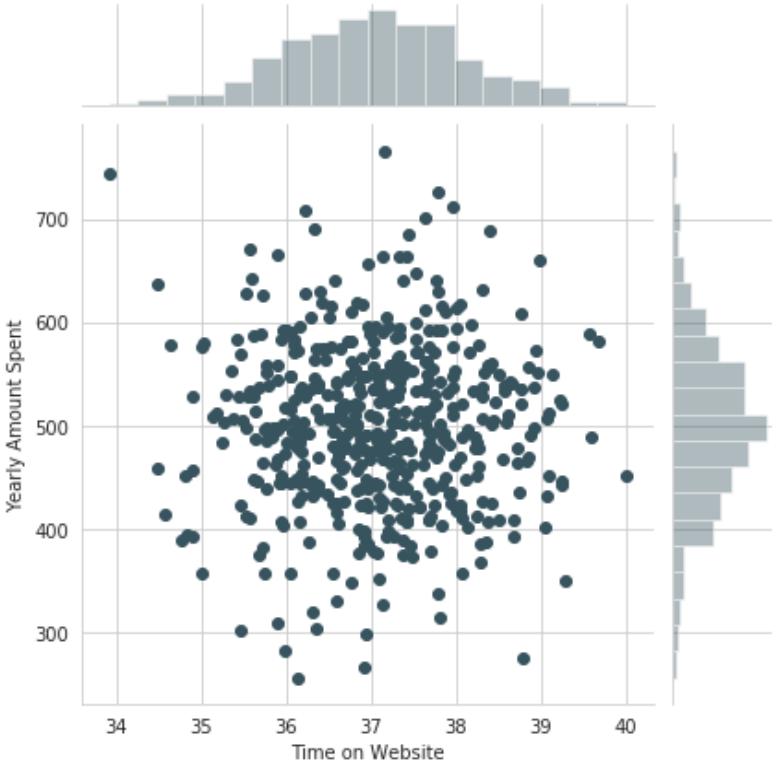


In [7]:

```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')
sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=customers)
```

Out[7]:

```
<seaborn.axisgrid.JointGrid at 0x7f365fe7c790>
```

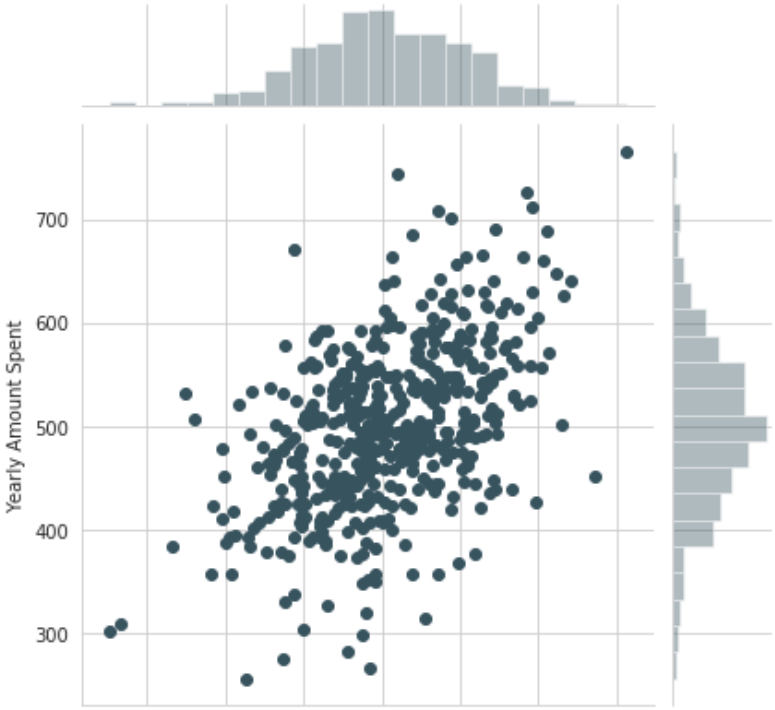


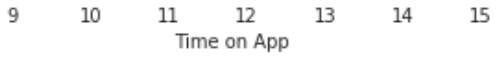
In [8]:

```
sns.jointplot(x='Time on App',y='Yearly Amount Spent',data=customers)
```

Out[8]:

```
<seaborn.axisgrid.JointGrid at 0x7f365ff3ae10>
```



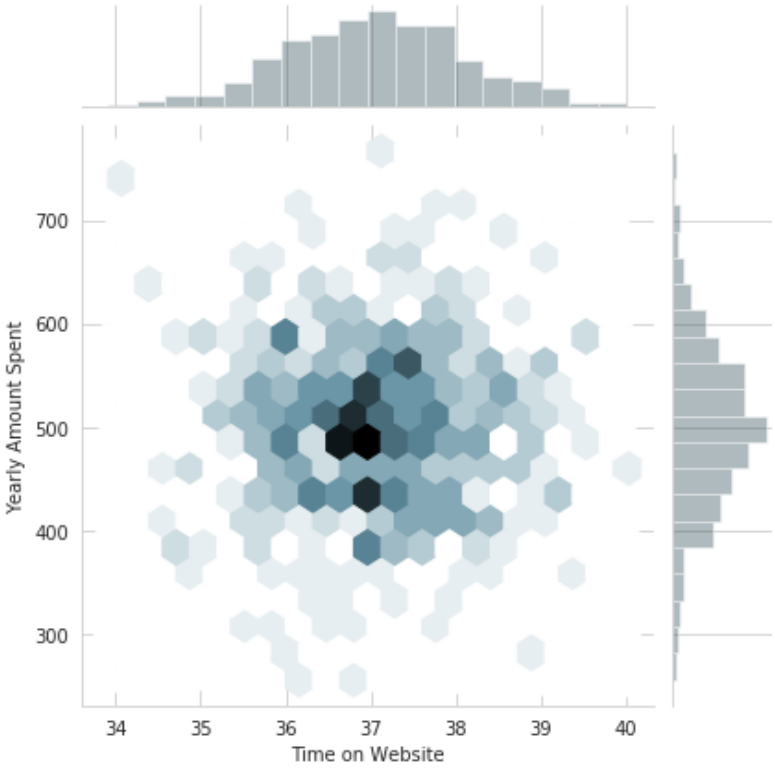


In [9]:

```
sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=customers,kind='hex')
```

Out[9]:

```
<seaborn.axisgrid.JointGrid at 0x7f365fb2c9d0>
```

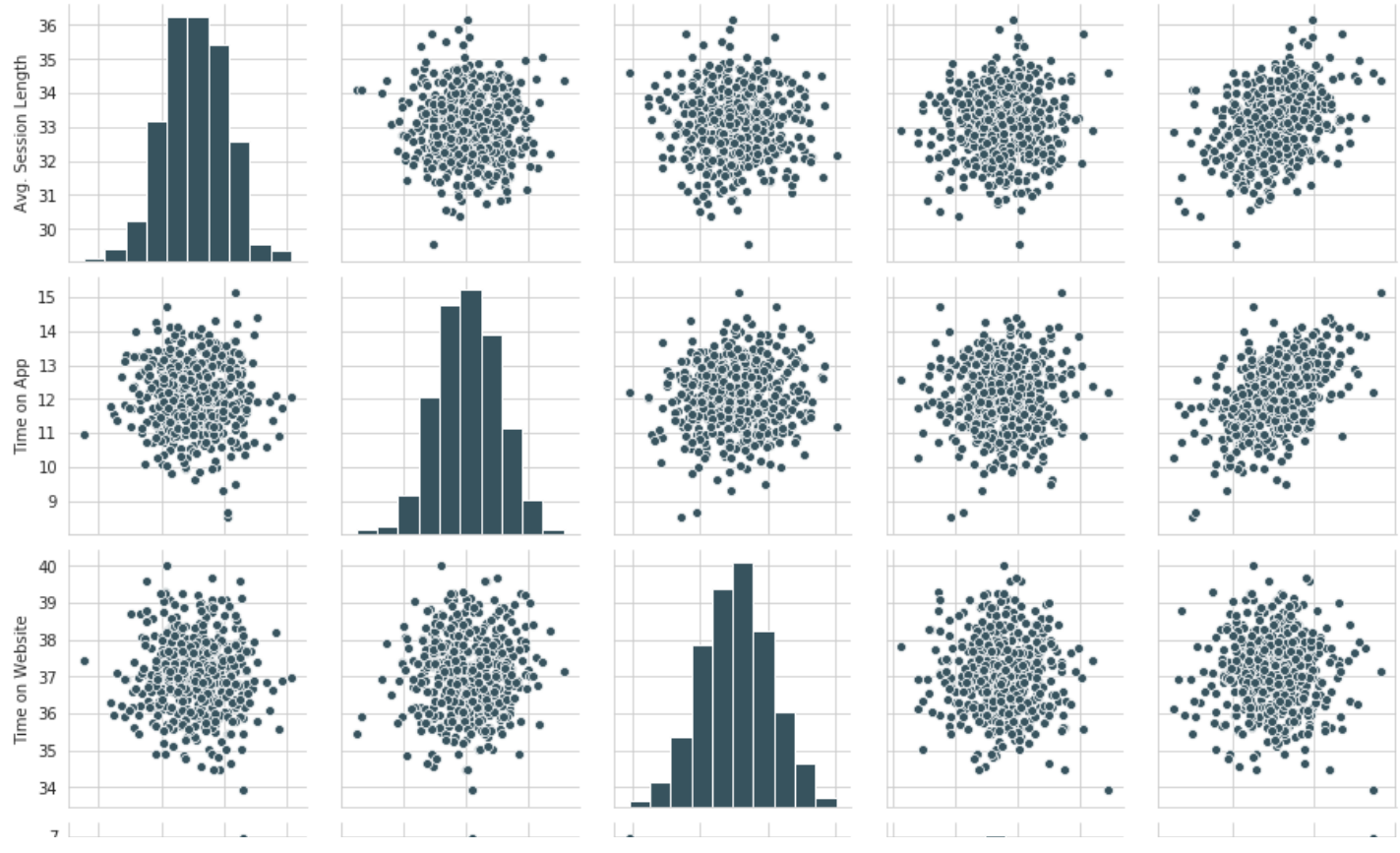


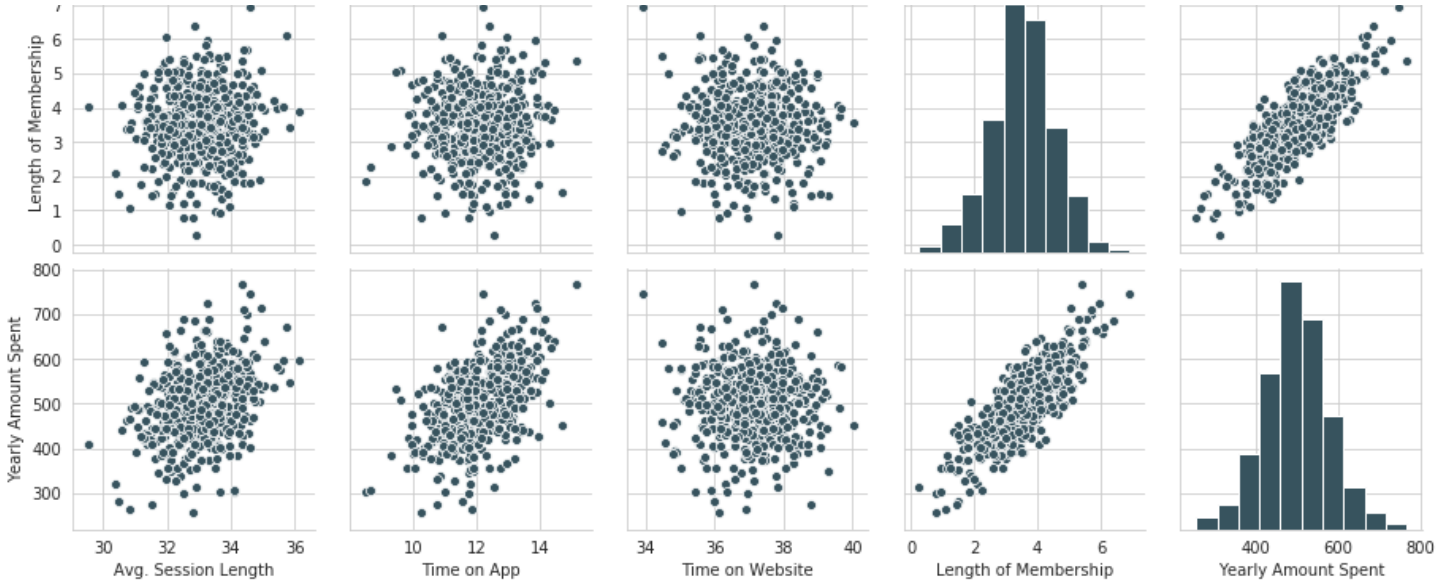
In [10]:

```
sns.pairplot(customers)
```

Out[10]:

```
<seaborn.axisgrid.PairGrid at 0x7f365f981090>
```



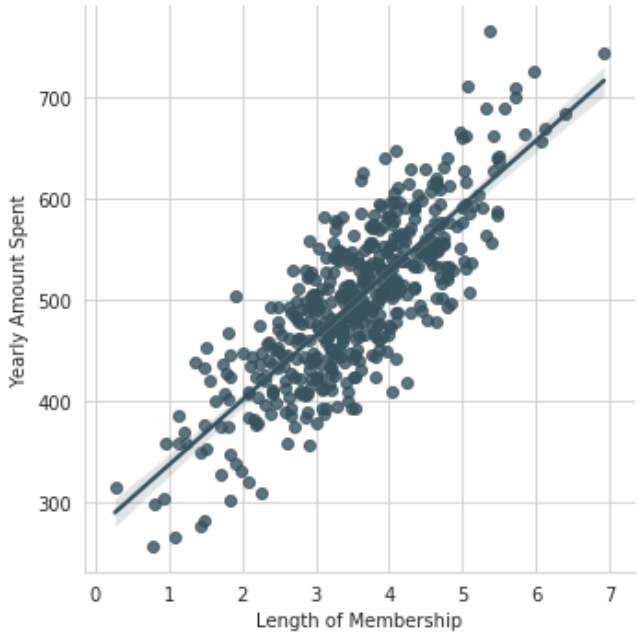


In [11]:

```
sns.lmplot(x='Length of Membership',y='Yearly Amount Spent',data=customers)
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x7f365ec64550>
```



Training and Testing Data

In [12]:

```
customers[1:3]
```

Out[12]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
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

In [13]:

```
X = customers[['Avg. Session Length','Time on App','Time on Website','Length of Membership
']]
y = customers['Yearly Amount Spent']
```

test_size=0.3

In [14]:

```
from sklearn.model_selection import train_test_split
```

In [15]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101
)
```

Training the Model

In [16]:

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

In [17]:

```
lm = LinearRegression()
```

In [18]:

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

Out[18]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Coefficients of the model

In [19]:

```
print(lm.coef_)
```

```
[25.98154972 38.59015875  0.19040528 61.27909654]
```

Predicting Test Data

In [20]:

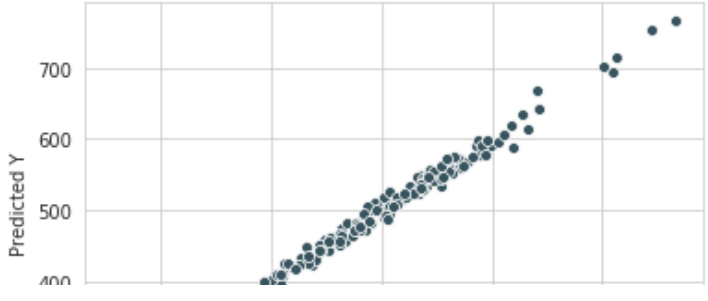
```
predictions = lm.predict(X_test)
```

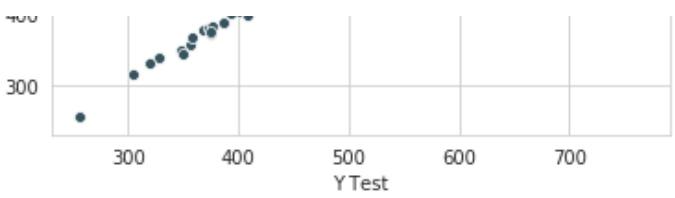
In [21]:

```
sns.scatterplot(x=y_test,y=predictions)
#plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[21]:

```
Text(0, 0.5, 'Predicted Y')
```





Evaluating the Model

Our model performance by calculating the residual sum of squares and the explained variance score (R^2).

In [22]:

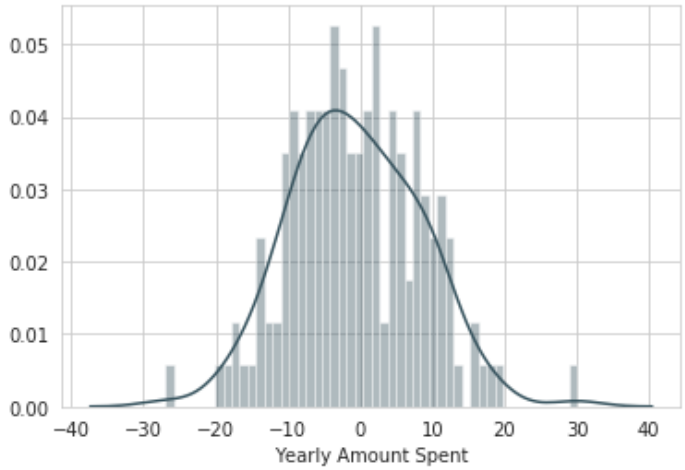
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test,predictions))
print('MSE:', metrics.mean_squared_error(y_test,predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

```
MAE: 7.228148653430853
MSE: 79.81305165097487
RMSE: 8.933815066978656
```

Residuals

In [23]:

```
#customers[['Yearly Amount Spent']].plot(kind='hist')
sns.distplot((y_test-predictions),bins=50);
# error = y_test - predictions
```



Conclusion

In [24]:

```
coeffecients = pd.DataFrame(data=lm.coef_,index=X.columns,columns=['Coeffecient'])
coeffecients.head()
```

Out[24]:

	Coeffecient
Avg. Session Length	25.981550
Time on App	38.590159
Time on Website	0.190405
Length of Membership	61.279097

In [25]:

```
metrics.explained_variance_score(y_test,predictions)
```

Out[25]:

```
0.9890771231889606
```

In [26]:

```
metrics.r2_score(y_test,predictions)
```

Out[26]:

```
0.9890046246741233
```