Linear Regression Project on Ecommerce company

You got a contract work with an Ecommerce company based 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. They have hired you on contract to help them figure it out.

Note : the data is not real is just for practices puropse only

In [16]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [8]:
```
customers = pd.read_csv('Ecommerce Customers')
```

In [11]:
```
customers.head()
```

Out[11]:

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

In [12]:
```
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: 31.4+ KB
```

In [16]:
```
customers.describe ()
```

Out[16]:

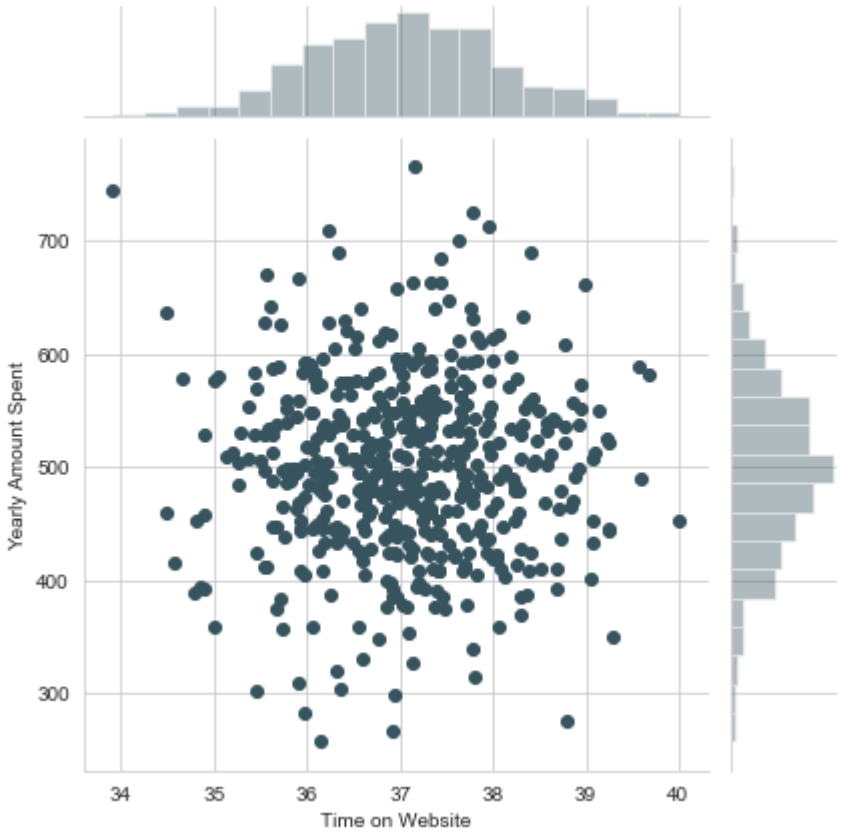
	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
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

Exploratory Data Analysis

In [18]:
```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')
```

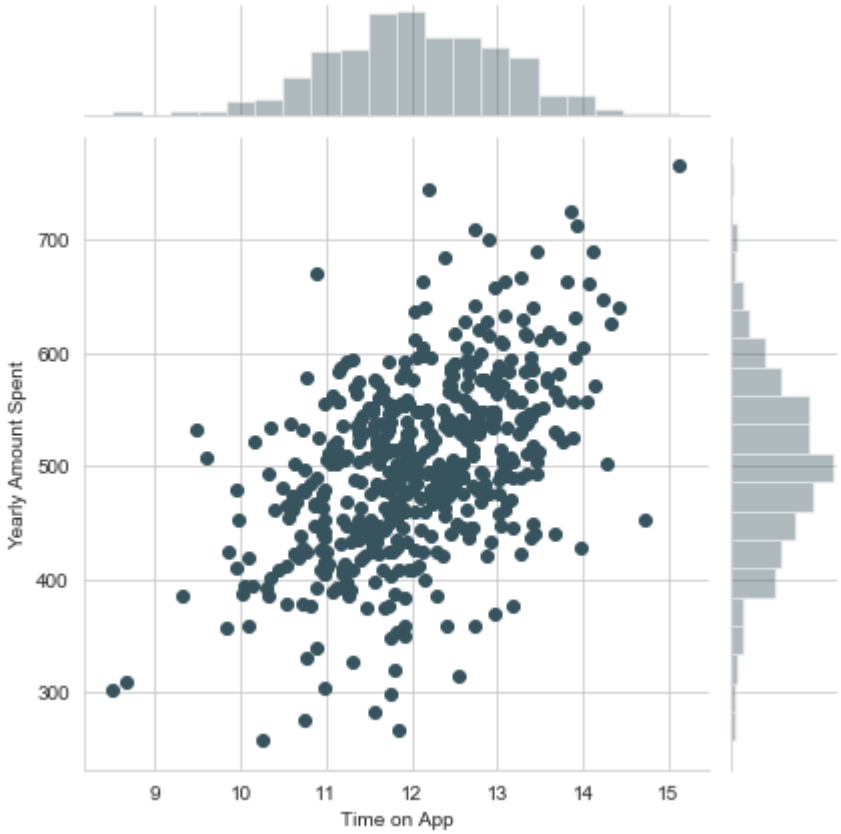
In [21]:
```
sns.jointplot( x='Time on Website', y= 'Yearly Amount Spent', data = customers
)
```

Out[21]: <seaborn.axisgrid.JointGrid at 0x273255ecf08>



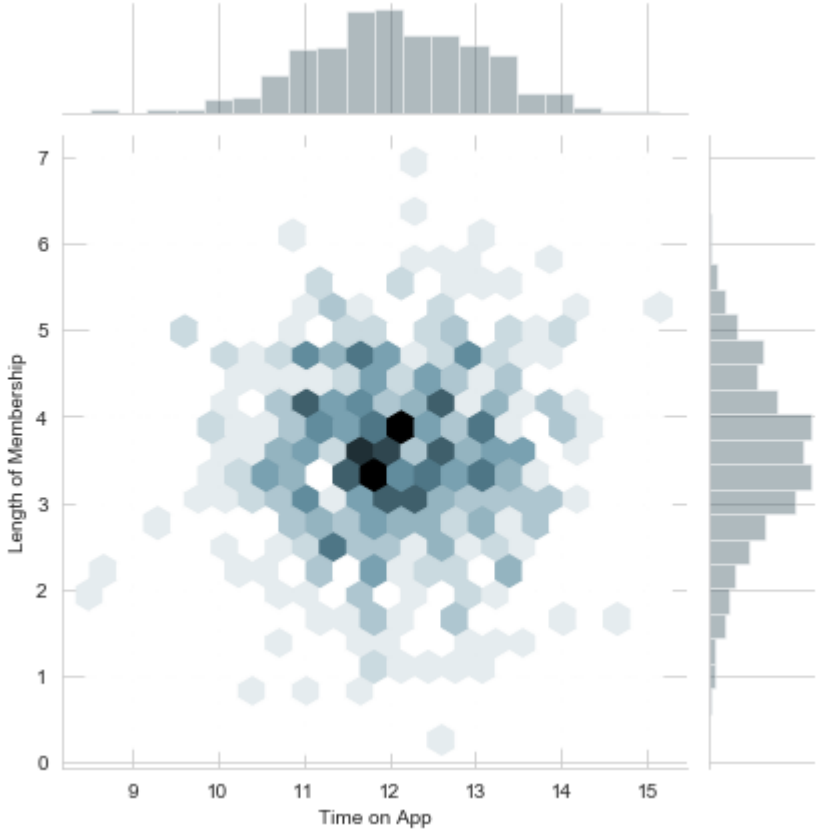
In [22]:
```
sns.jointplot( x='Time on App', y= 'Yearly Amount Spent', data = customers )
```

Out[22]:
```
<seaborn.axisgrid.JointGrid at 0x273259f6ec8>
```



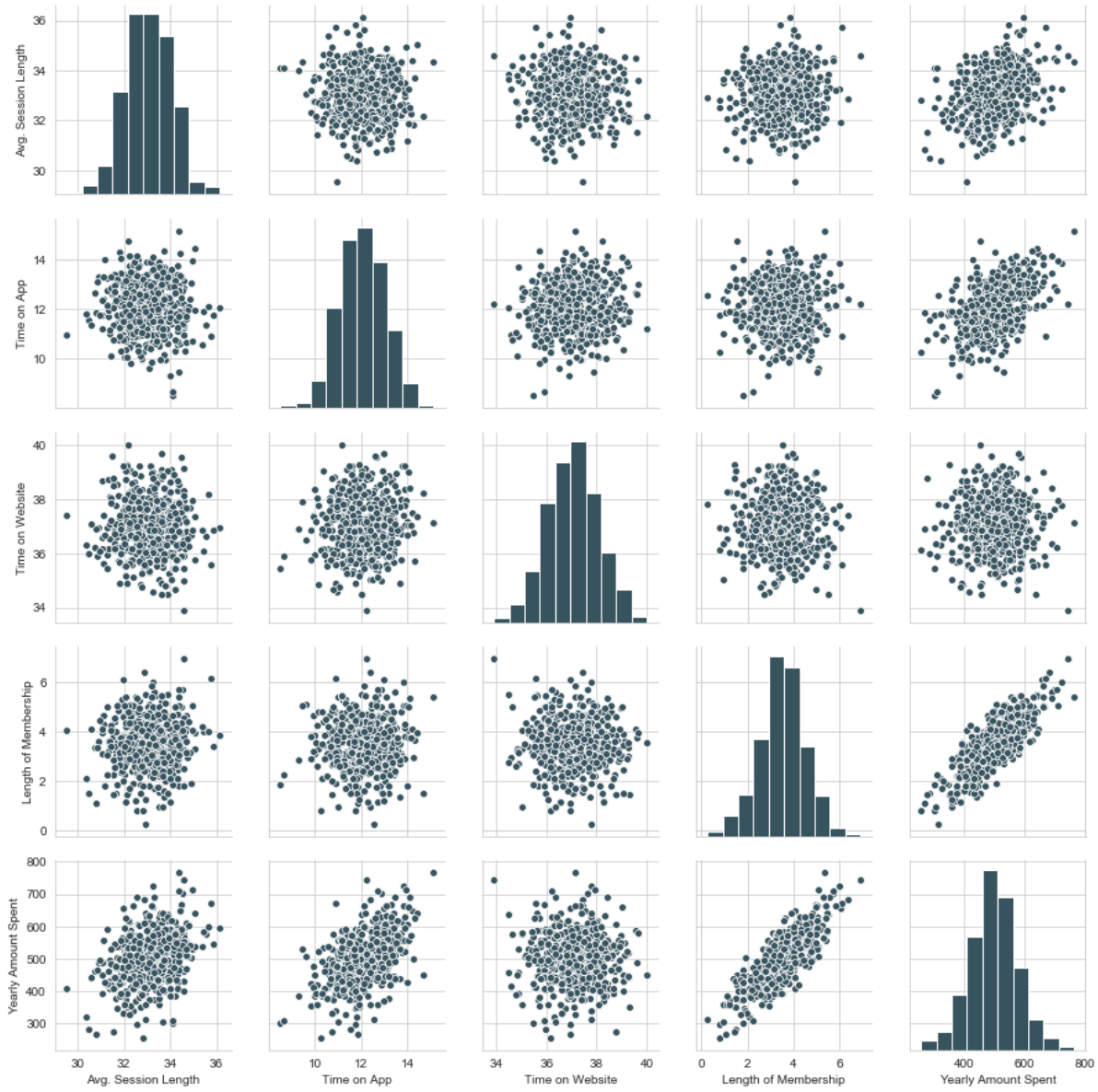
In [23]:
```
sns.jointplot( x='Time on App', y= 'Length of Membership', kind = 'hex', data = customers )
```

Out[23]: <seaborn.axisgrid.JointGrid at 0x27325b9d148>



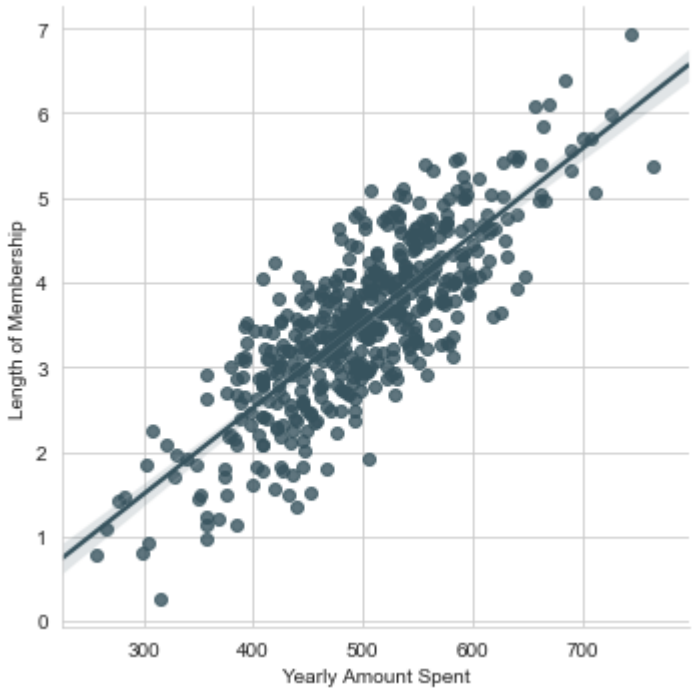
In [28]:
```
sns.pairplot(customers)
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x27325d38c08>
```



In [29]:
```
sns.lmplot(x ='Yearly Amount Spent', y ='Length of Membership', data = customers)
```

Out[29]: <seaborn.axisgrid.FacetGrid at 0x2732697c708>



Training and Testing Data

In [5]:
```
y= customers['Yearly Amount Spent']
```

In [17]:
```
X = customers[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Membership']]
```

In [18]:
```
from sklearn.model_selection import train_test_split
```

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=101)
```

In [20]:
```
from sklearn.linear_model import LinearRegression
```

In [25]:
```
lm = LinearRegression()
```

In [27]:
```
lm.fit(X_train,y_train)
```

Out[27]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [33]:
```
lm.coef_
```

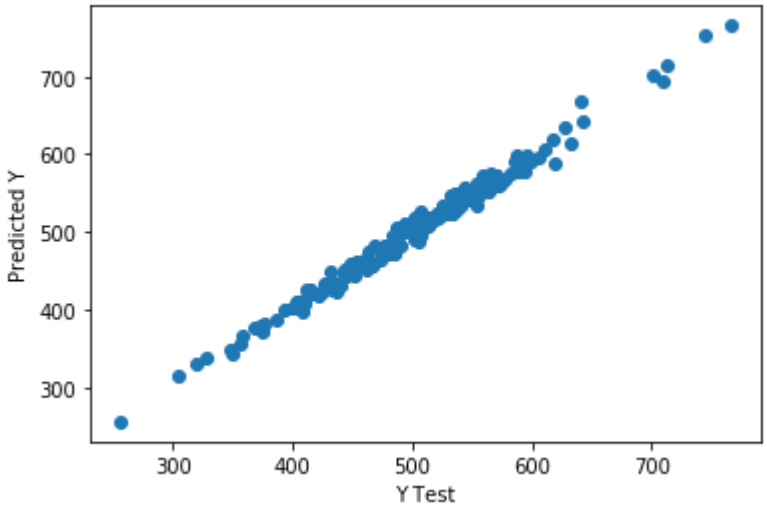
Out[33]:
```
array([25.98154972, 38.59015875,  0.19040528, 61.27909654])
```

Predicting Test Data

In [35]:
```
predictions =lm .predict( X_test)
```

In [36]:
```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[36]: Text(0, 0.5, 'Predicted Y')



Evaluating the Model

evaluating our model performance by calculating the residual sum of squares and the (EXTRA) explained variance score (R^2). Calculate the Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error.

In [37]:
```
from sklearn import metrics
```

In [40]:
```
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: 7.228148653430838
MSE: 79.81305165097461
RMSE: 8.933815066978642
```

In [41]:
```
metrics.explained_variance_score(y_test,predictions)
```

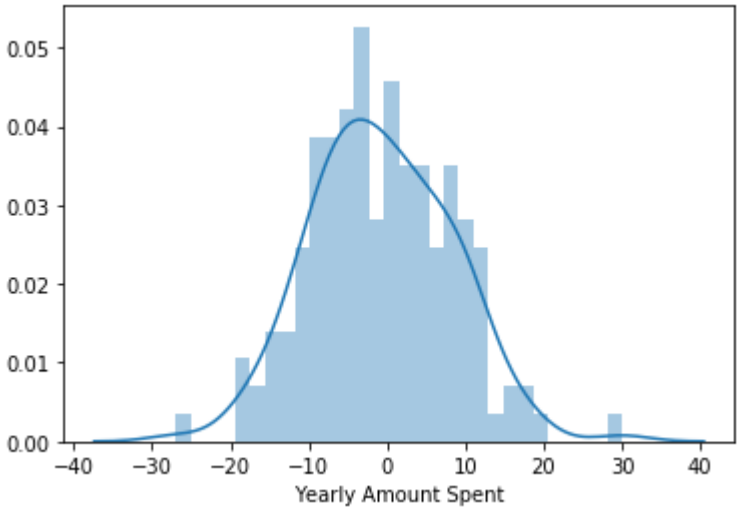
Out[41]: 0.9890771231889606

Residuals

Let explore the residuals to make sure everything was okay with our data.

Ploting a histogram of the residuals and make sure it looks normally distributed.

In [45]:
```
sns.distplot((y_test-predictions),bins=30);
```



Conclusion

We want to figure out the answer to the original question, do we focus our efforst on mobile app or website development? Or maybe that doesn't even really matter, and Membership Time is what is really important.

In [46]:
```
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[46]:

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

Base on the result above from the coefficient when all variable are fix there is a one unit increase on the Avg session length of 26 which apply to all and it shows clearly an increase of 19 cent of the web site.

Base on the question on this project should they spend more time on their mobile app or website, this is based on the management team that will take the final decision on this. However the mobile app is looking good,running and working well while the website is not. This is why the management team have to come in play to give detail of the expensive and the cost of improving the mobile app which is good or the website