Linear Regression Project - Time spent on app

Imports

In [2]:

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

Data

working with the Ecommerce Customers csv file from the company. It has Customer info, suchas Email, Address, and their color Avatar. Then it also has numerical value columns:

- 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 [3]:

```
customers = pd.read_csv("Ecommerce Customers")
```

In [4]:

```
customers.head()
```

Out[4]:

	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 [6]:

```
customers.describe()
```

Out[6]:

	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 [7]:

```
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
```

Exploratory Data Analysis

**seaborn to create a jointplot to compare the Time on Website and Yearly Amount Spent columns.

In [8]:

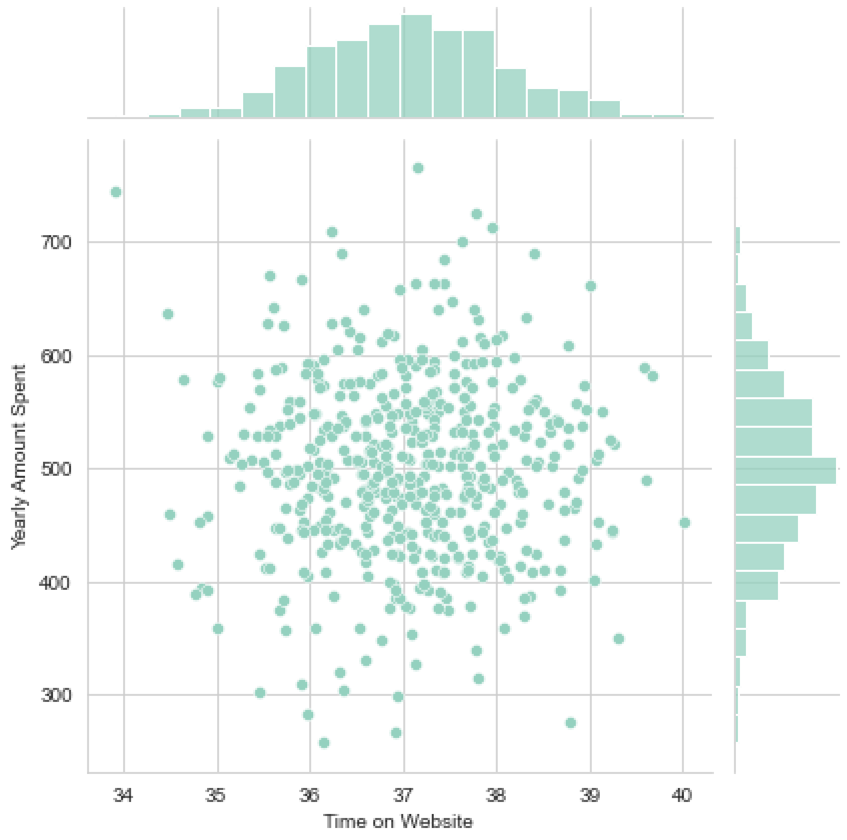
```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')
```

In [9]:

```
# More time on site, more money spent.
sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=customers)
```

Out[9]:

```
<seaborn.axisgrid.JointGrid at 0x7fca22ca32b0>
```

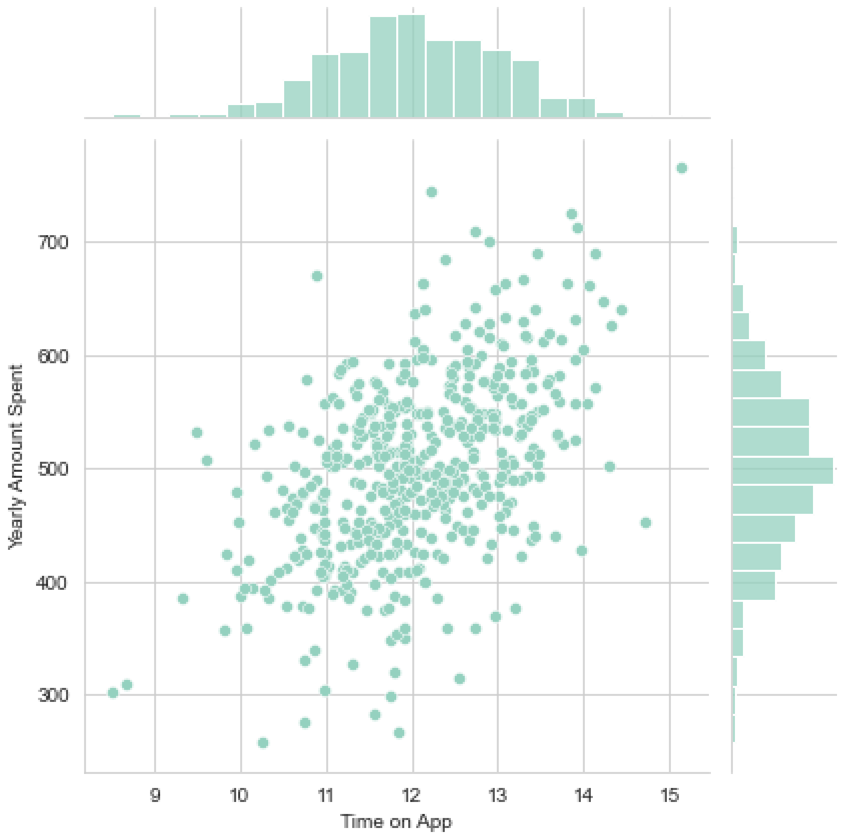


In [10]:

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

Out[10]:

```
<seaborn.axisgrid.JointGrid at 0x7fca23236d68>
```



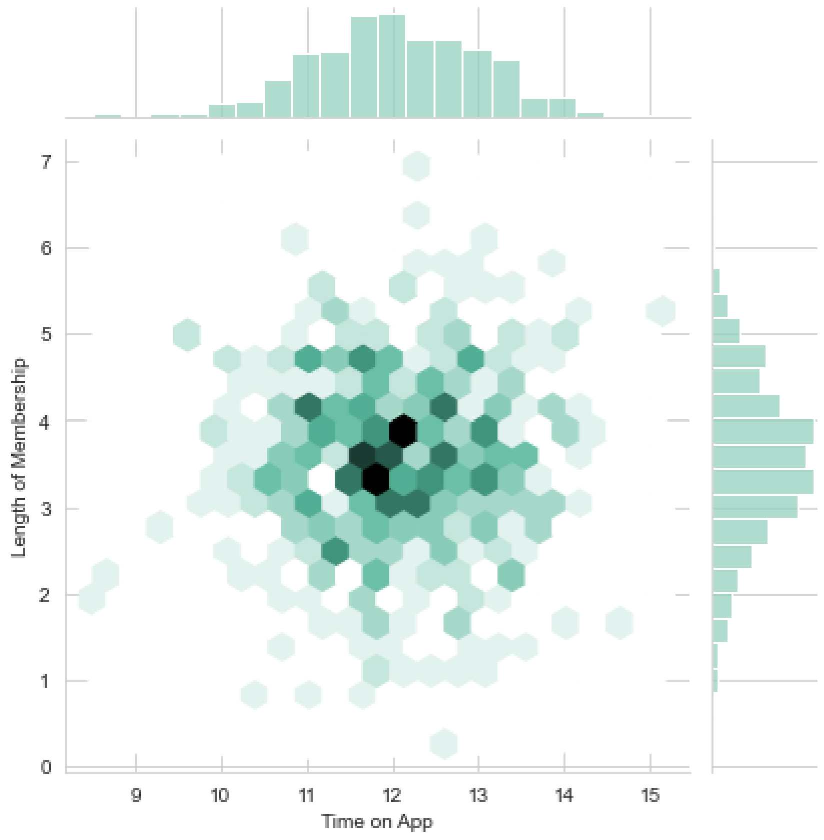
Used jointplot to create a 2D hex bin plot comparing Time on App and Length of Membership.

In [11]:

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

Out[11]:

```
<seaborn.axisgrid.JointGrid at 0x7fca23221780>
```



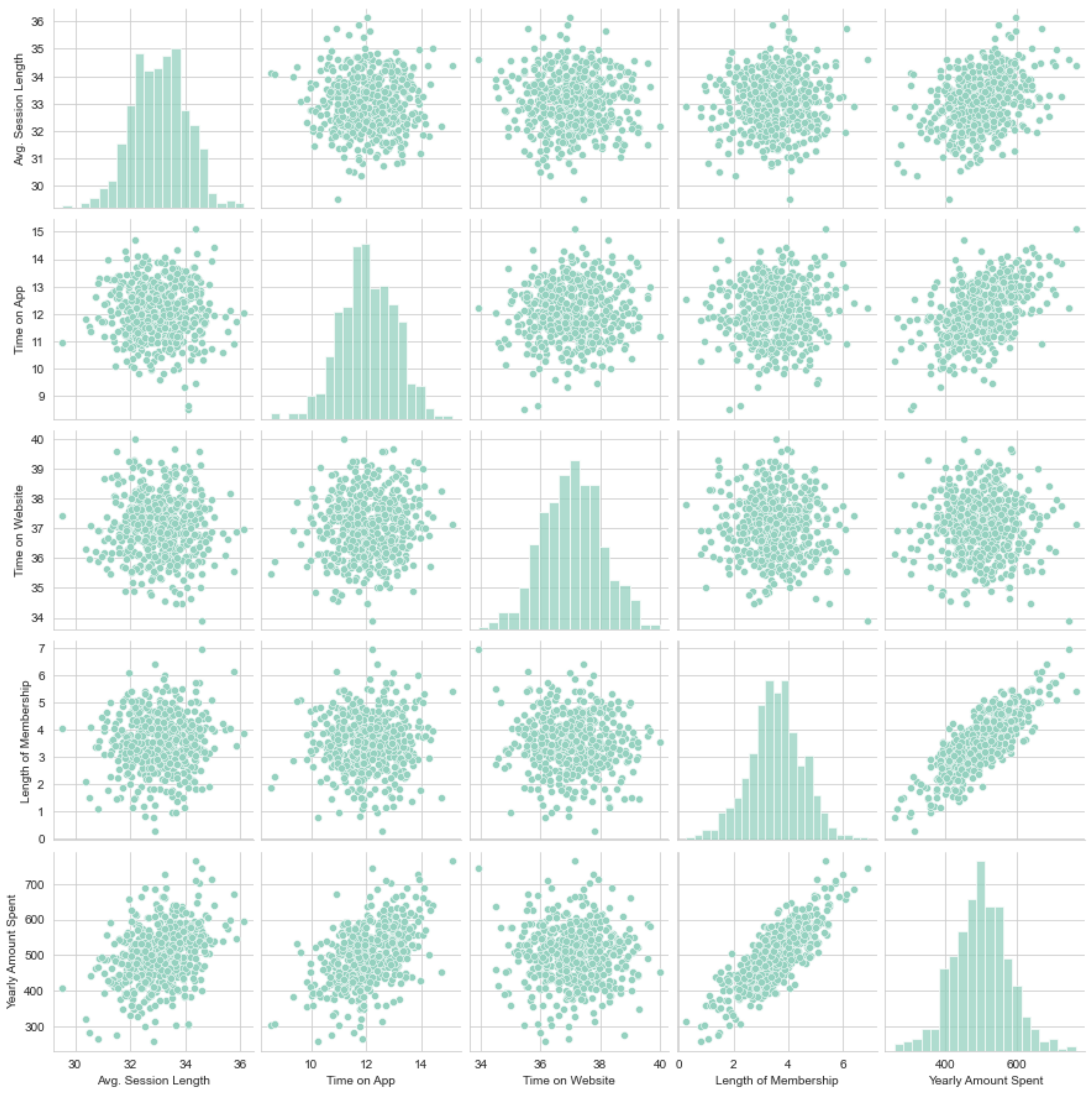
Using Pairplot

In [12]:

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

Out[12]:

```
<seaborn.axisgrid.PairGrid at 0x7fca237300f0>
```



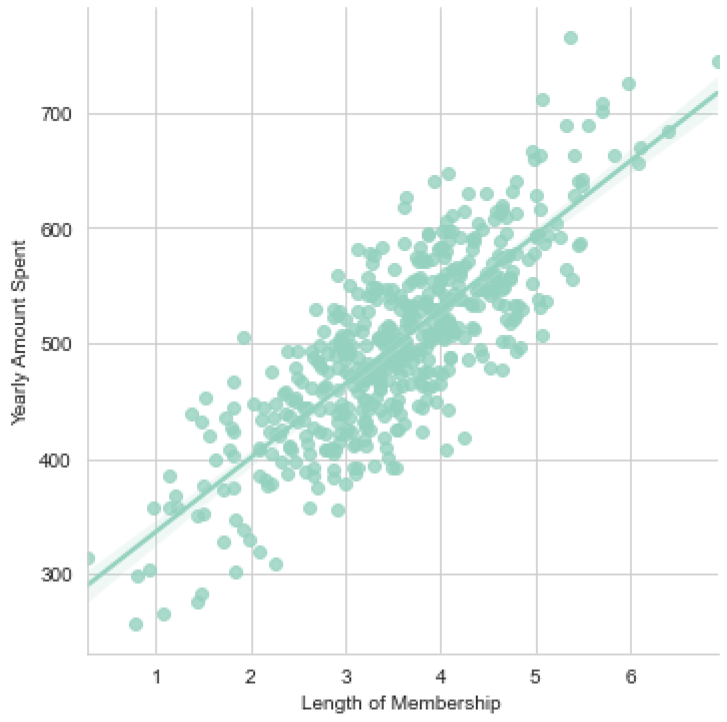
*Created a linear model plot (using seaborn's lmplot) of Yearly Amount Spent vs. Length of Membership. *

In [13]:

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

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x7fca24adf320>
```



Training and Testing Data

spliting the data into training and testing sets. ** Set a variable X equal to the numerical features of the customers and a variable y equal to the "Yearly Amount Spent" column. **

In [14]:

```
y = customers['Yearly Amount Spent']
```

In [15]:

```
X = customers[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Me
```

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

In [16]:

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

In [17]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_stat
```

Training the Model

In [18]:

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

Create an instance of a LinearRegression() model named lm.

In [19]:

```
lm = LinearRegression()
```

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

In [20]:

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

Out[20]:

```
LinearRegression()
```

Print out the coefficients of the model

In [21]:

```
# The coefficients
print('Coefficients: \n', lm.coef_)
```

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

Predicting Test Data

evaluating its performance by predicting off the test values!

In [22]:

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

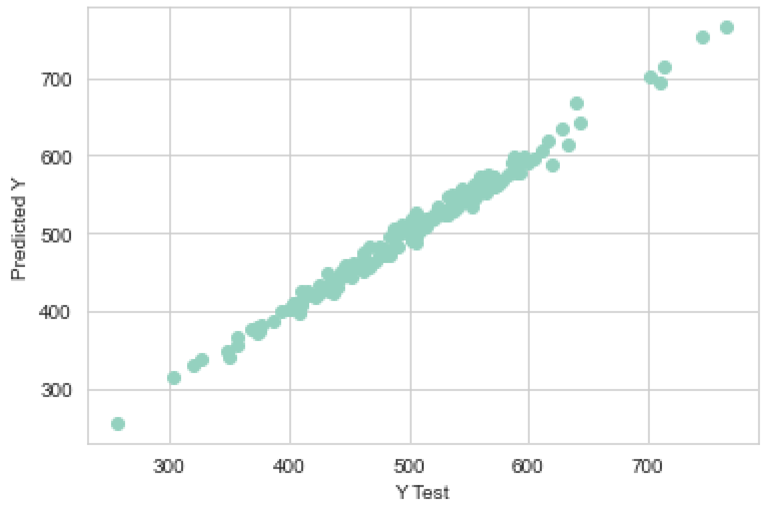
** Created a scatterplot of the real test values versus the predicted values. **

In [23]:

```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[23]:

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



Evaluating the Model

evaluating 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.

In [24]:

```

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: 7.2281486534308295
MSE: 79.8130516509743
RMSE: 8.933815066978626
```

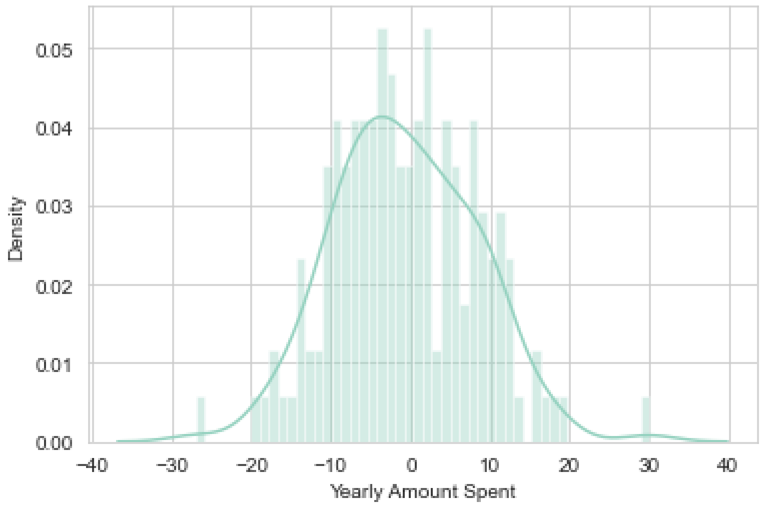
Residuals

Plotted a histogram of the residuals

In [25]:

```
sns.distplot((y_test-predictions),bins=50);
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-p
ackages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please ad
apt your code to use either `displot` (a figure-level function with si
milar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```



Conclusion

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

** How can you interpret these coefficients? **

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.**