Ecommerce Customers Data Set

From the below data set, we will derive how the customer experience on ecommerce portal (web page or mobile app) affects the revenue of the ecommerce company. The below data is made up data though.

Lets get started!

Loading the data

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

In [2]:
```
df = pd.read_csv('Ecommerce Customers')
```

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

Out[3]:

	Email	Address	Avatar	Avg. Session Length	-
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12

In [4]:
```
df.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 [5]:
```
df.describe()
```

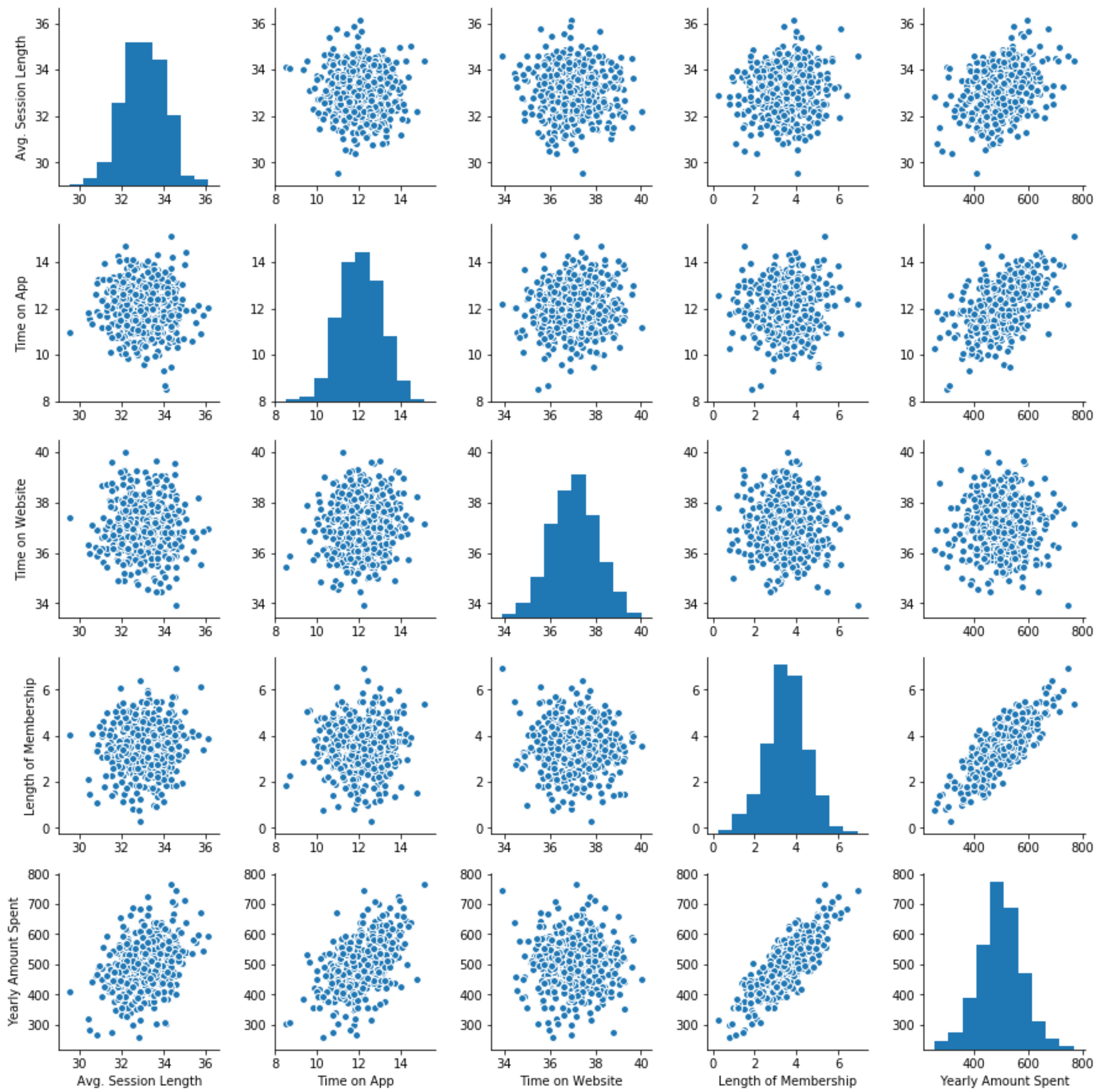
Out[5]:

	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

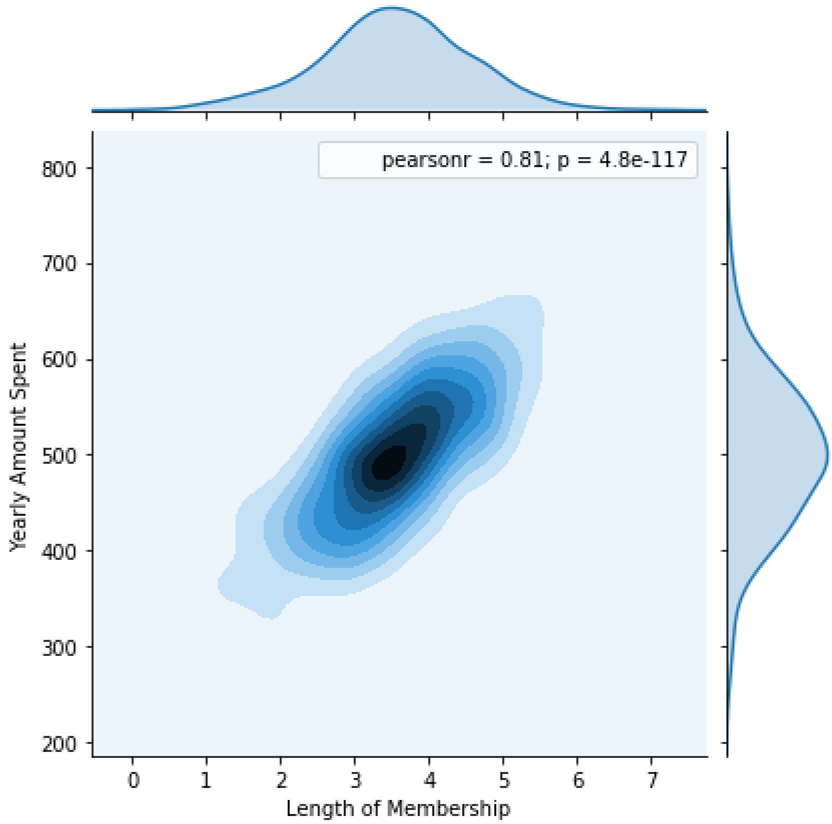
In [6]:
```
sns.pairplot(df)
```

Out[6]: <seaborn.axisgrid.PairGrid at 0xbd632e8>



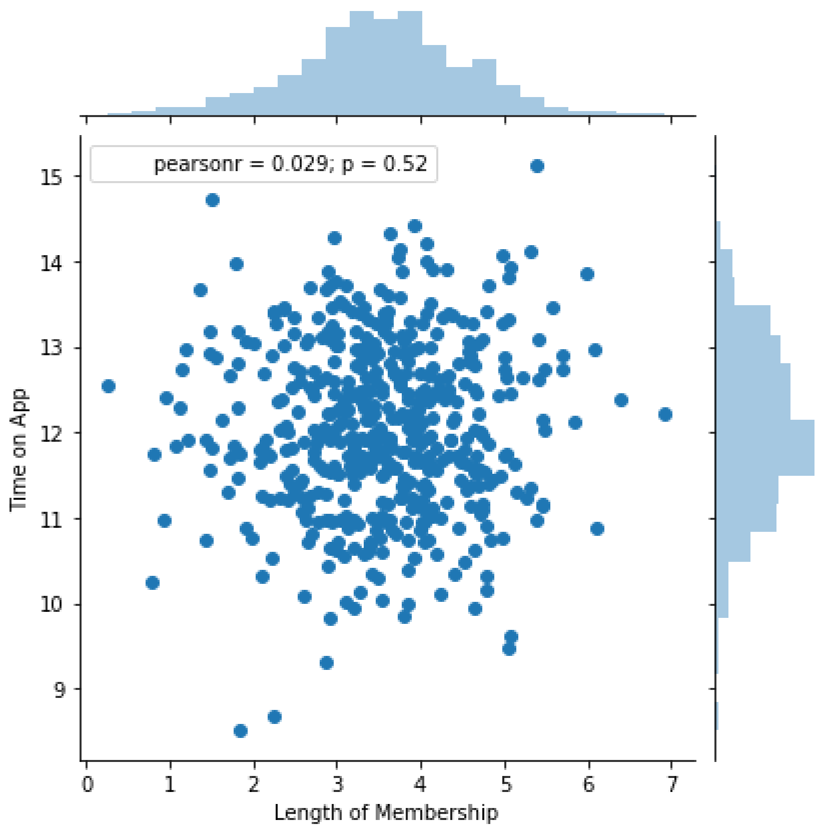
In [8]:
```
sns.jointplot(df['Length of Membership'], df['Yearly Amount Spent'], kind='kd
e')
```

Out[8]: <seaborn.axisgrid.JointGrid at 0xd73a4a8>



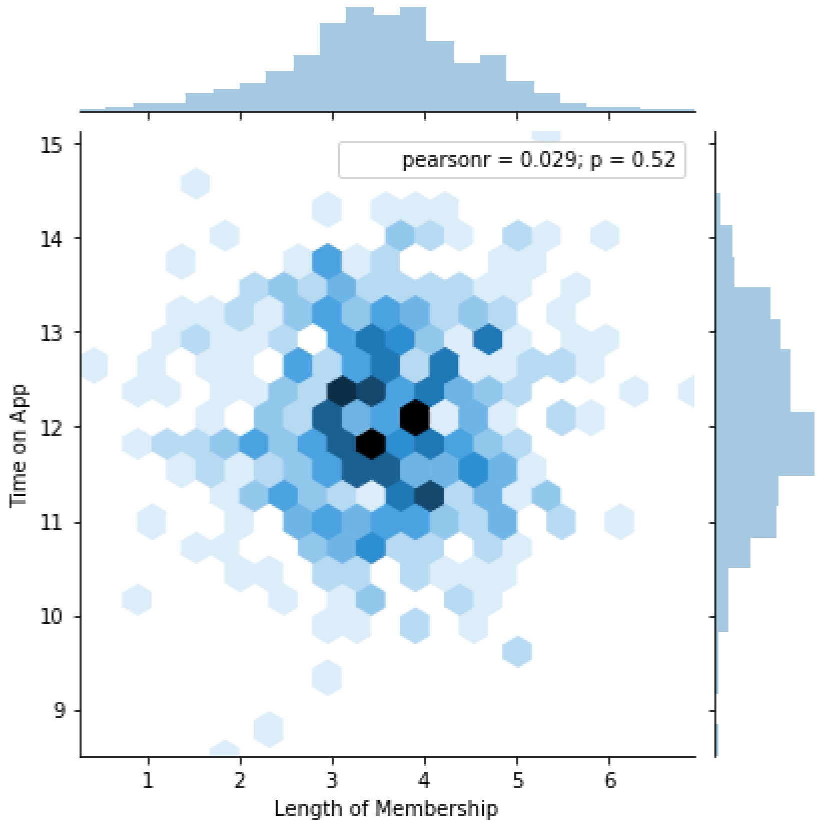
In [9]:
```
sns.jointplot(df['Length of Membership'], df['Time on App'])
```

Out[9]: <seaborn.axisgrid.JointGrid at 0xdefa978>



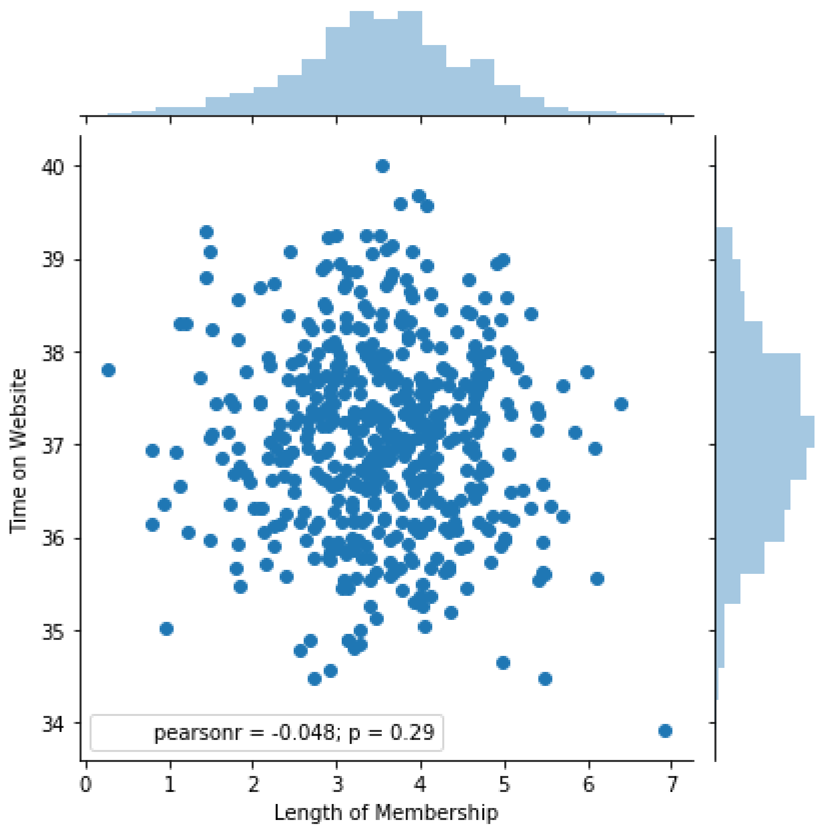
In [14]:
```
sns.jointplot(x=df['Length of Membership'], y=df['Time on App'], data=df, kind
='hex')
```

Out[14]: <seaborn.axisgrid.JointGrid at 0xf1afc88>



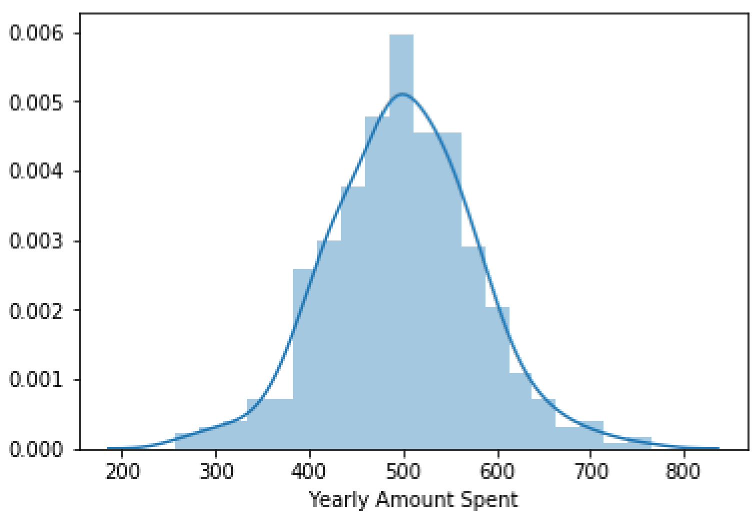
In [15]:
```
sns.jointplot(df['Length of Membership'], df['Time on Website'])
```

Out[15]: <seaborn.axisgrid.JointGrid at 0xf4c22b0>



In [16]:
```
sns.distplot(df['Yearly Amount Spent'])
```

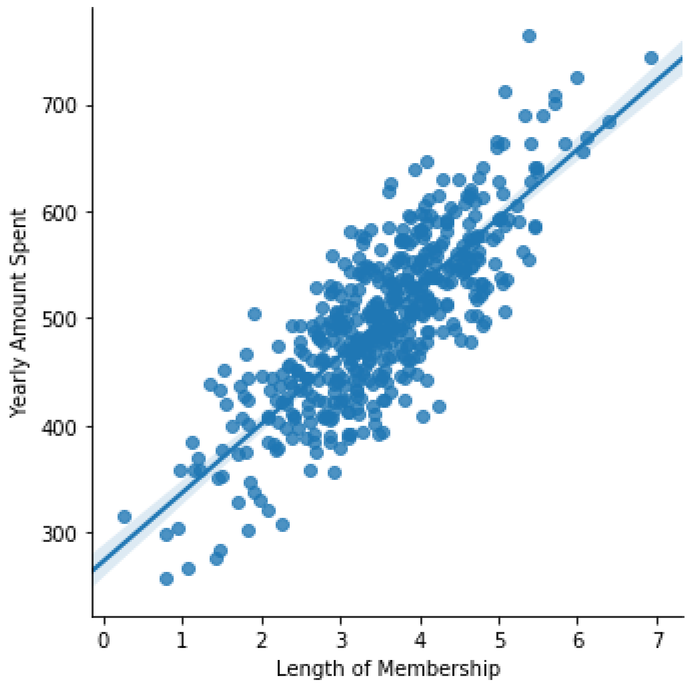
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0xf6cc828>



Checking the linear relationsip with graph

In [21]:
```
sns.lmplot(x='Length of Membership', y='Yearly Amount Spent', data=df)
```

Out[21]: <seaborn.axisgrid.FacetGrid at 0xf77b240>



In [23]:
```
sns.heatmap(df.corr(), annot=True, cmap='viridis')
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1016ce10>

Training and Testing Data

Now that we've explored the data a bit, let's go ahead and split 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 [24]:
```
from sklearn.model_selection import train_test_split
```

In [25]:
```
df.columns
```

Out[25]:
```
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',
       'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

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 [34]:
```
X_train, X_test, y_train, y_test = train_test_split(df.drop(['Yearly Amount Sp
ent','Email','Address','Avatar'], axis=1), df['Yearly Amount Spent'],
                                                    test_size=0.3, random_state
=101)
```

Training the Model

Now its time to train our model on our training data!

Import LinearRegression from sklearn.linear_model

In [35]:
```
from sklearn.linear_model import LinearRegression
```

In [36]:
```
lm = LinearRegression()
```

In [37]:
```
lm.fit(X_train, y_train)
```

Out[37]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Print out the coefficients of the model

In [40]:
```
print("Coefficients:", lm.coef_)
```

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

Predicting Test Data

Now that we have fit our model, let's evaluate its performance by predicting off the test values!

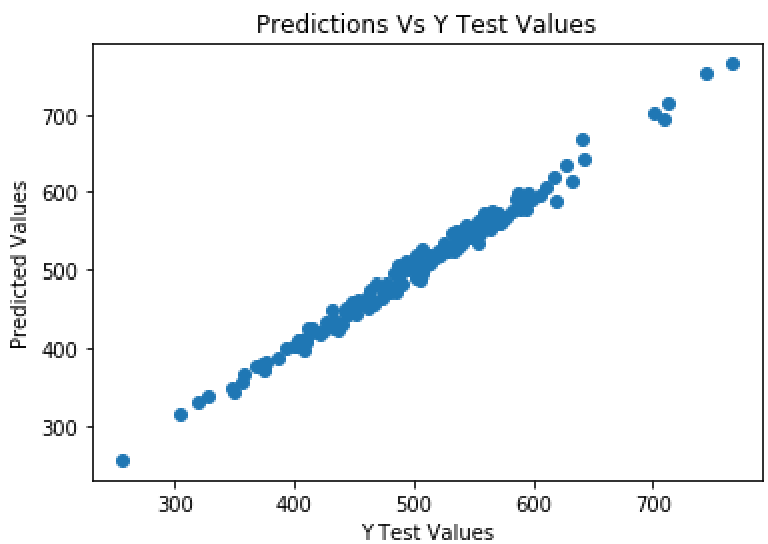
Use lm.predict() to predict off the X_test set of the data.

In [41]:
```
predict = lm.predict(X_test)
```

Create a scatterplot of the real test values versus the predicted values.

In [46]:
```
plt.scatter(y_test, predict)
plt.ylabel('Predicted Values')
plt.xlabel('Y Test Values')
plt.title('Predictions Vs Y Test Values')
```

Out[46]: `Text(0.5,1,'Predictions Vs Y Test Values')`



Evaluating the Model

Let's evaluate our 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. Refer to the lecture or to Wikipedia for the formulas

In [ ]:
```
# calculate these metrics by hand!
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)))
```

In [49]:
```
from sklearn import metrics
```

In [50]:
```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predict))
print('Mean Square Error:', metrics.mean_squared_error(y_test, predict))
print('Root Mean Square Error:', np.sqrt(metrics.mean_squared_error(y_test, pr
edict)))
```

```
Mean Absolute Error: 7.22814865343
Mean Square Error: 79.813051651
Root Mean Square Error: 8.93381506698
```

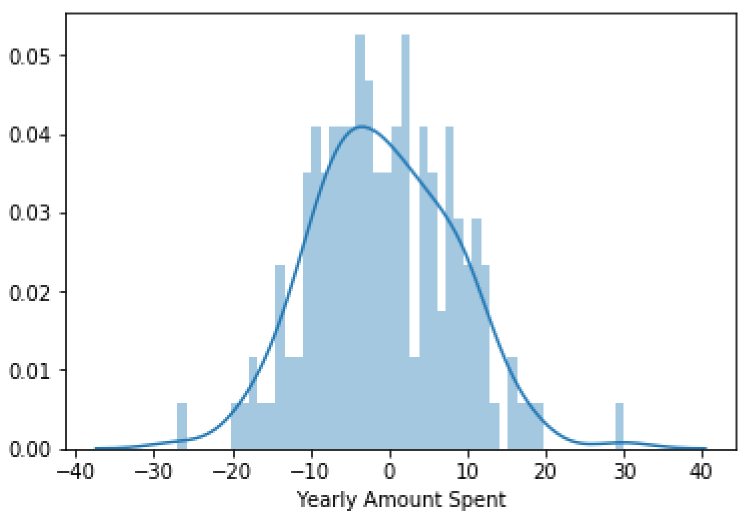
Residuals

You should have gotten a very good model with a good fit. Let's quickly explore the residuals to make sure everything was okay with our data.

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

In [52]:
```
sns.distplot((y_test-predict), bins=50)
```

Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x11aa4828>



Conclusion

We still 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. Let's see if we can interpret the coefficients at all to get an idea.

Recreate the dataframe below.

In [53]:
```
df.columns
```

Out[53]:
```
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',
       'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

In [58]:
```
coefficients = pd.DataFrame(lm.coef_, columns=['Coefficients'],index=['Avg. Se
ssion Length', 'Time on App',
       'Time on Website', 'Length of Membership'],)
```

In [59]:
```
coefficients
```

Out[59]:

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

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**.