
Linear Regression - Project Exercise

You just got some 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've hired you on contract to help them figure it out! Let's get started!

Just follow the steps below to analyze the customer data (it's fake, don't worry I didn't give you real credit card numbers or emails).

Imports

Import pandas, numpy, matplotlib, and seaborn. Then set %matplotlib inline (You'll import sklearn as you need it.)

Get the Data ¶

We'll work with the Ecommerce Customers csv file from the company. It has Customer info, such as 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.

Read in the Ecommerce Customers csv file as a DataFrame called customers.

Check the head of customers, and check out its info() and describe() methods.

In [5]:

Out[5]:

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

In [6]:

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]:

```
<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
```

Exploratory Data Analysis

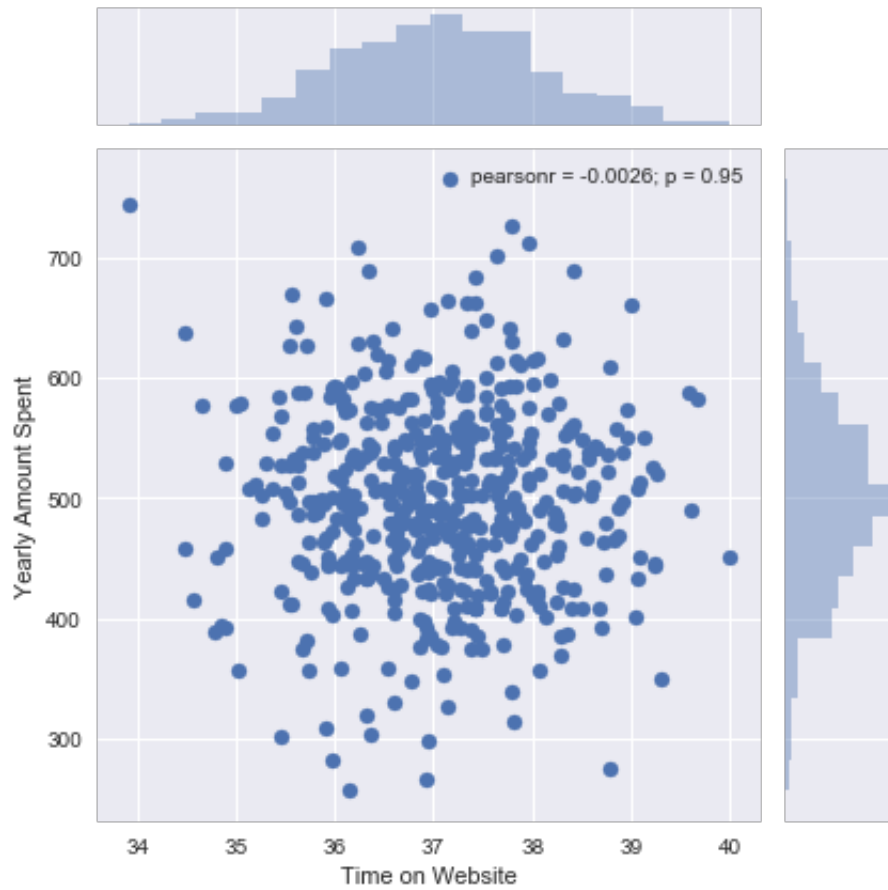
Let's explore the data!

For the rest of the exercise we'll only be using the numerical data of the csv file.

Use seaborn to create a jointplot to compare the Time on Website and Yearly Amount Spent columns. Does the correlation make sense?

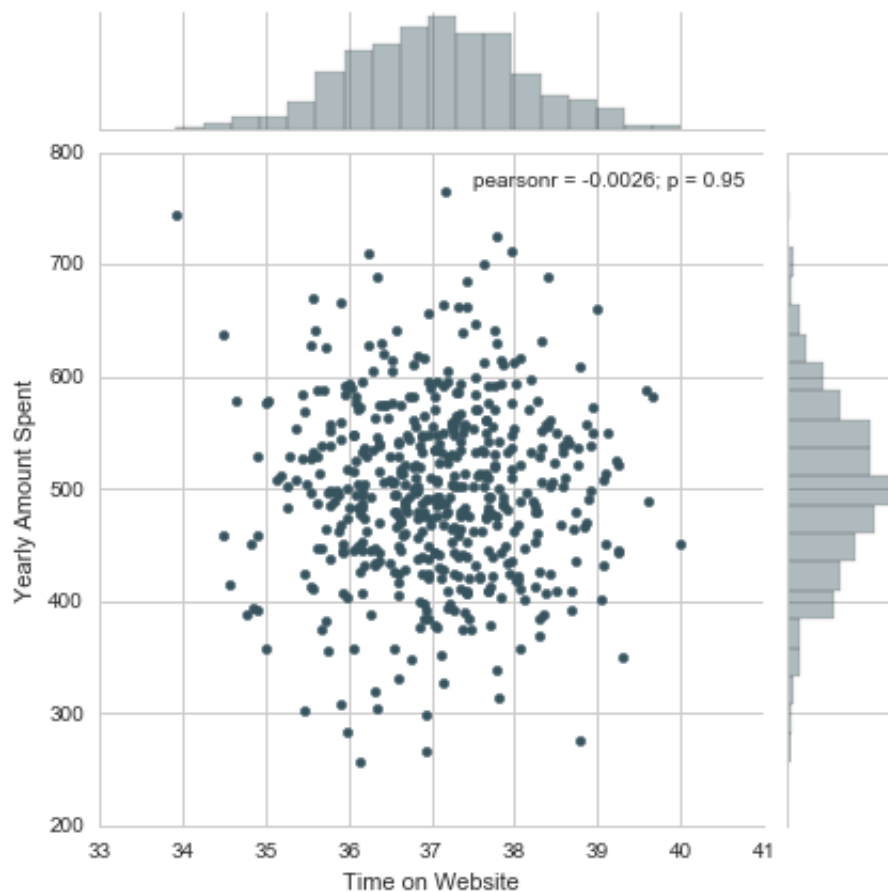
In [9]:

Out[9]: <seaborn.axisgrid.JointGrid at 0x11a2228d0>



In [281]:

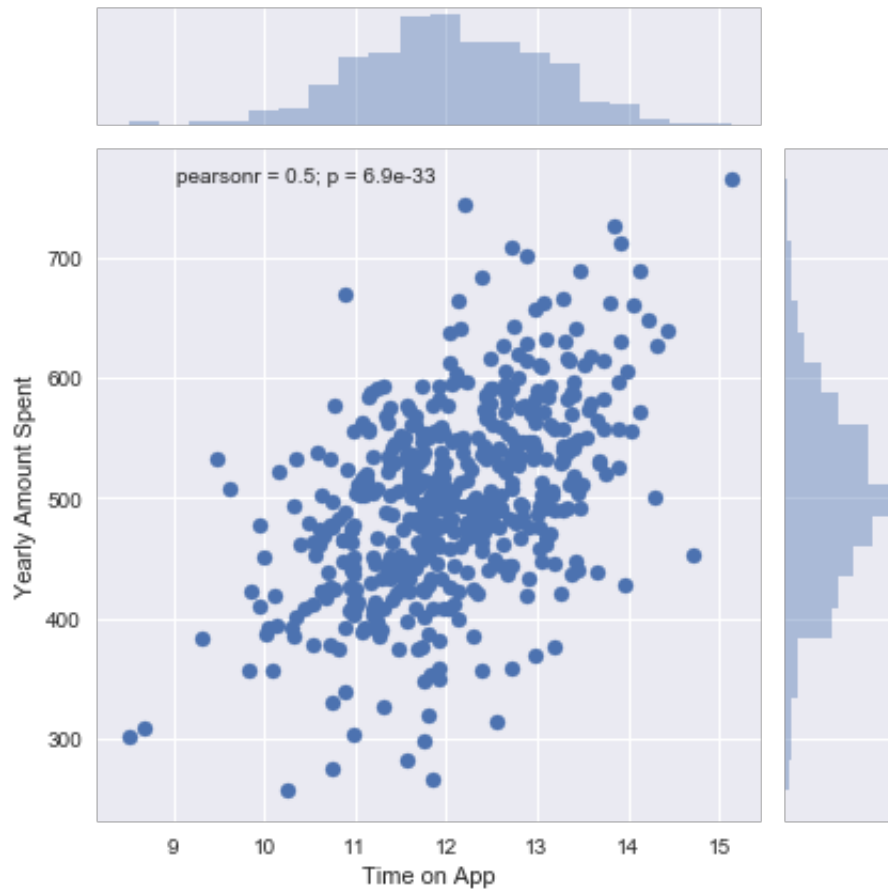
Out[281]: <seaborn.axisgrid.JointGrid at 0x120bfcc88>



Do the same but with the Time on App column instead.

In [11]:

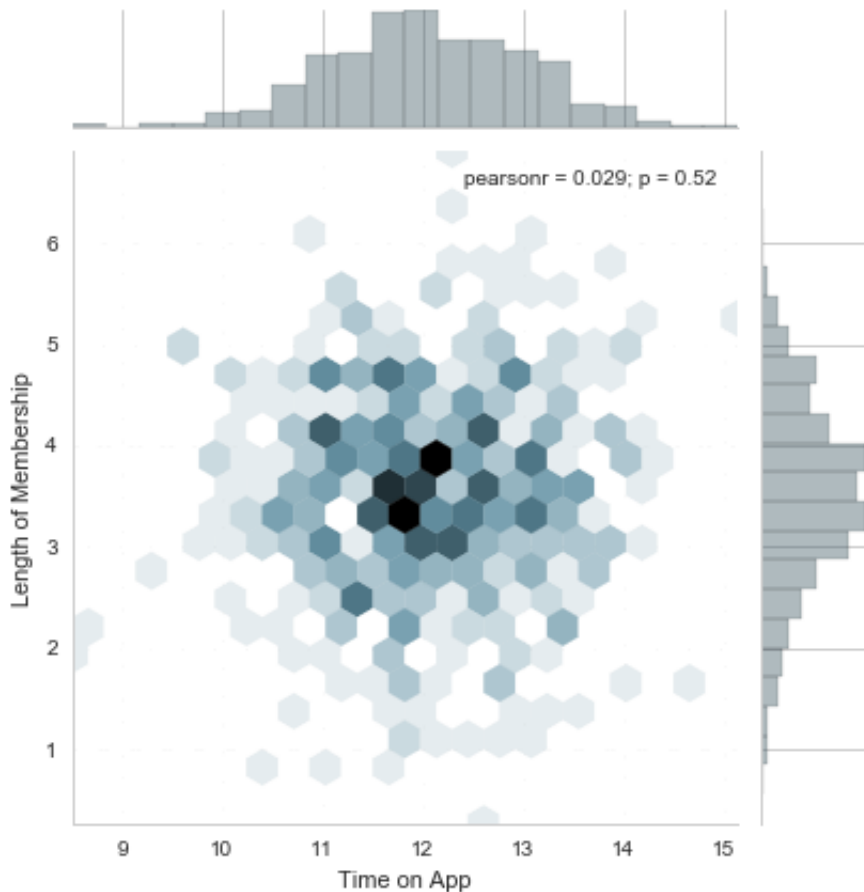
Out[11]: <seaborn.axisgrid.JointGrid at 0x11dcd8b10>



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

In [283]:

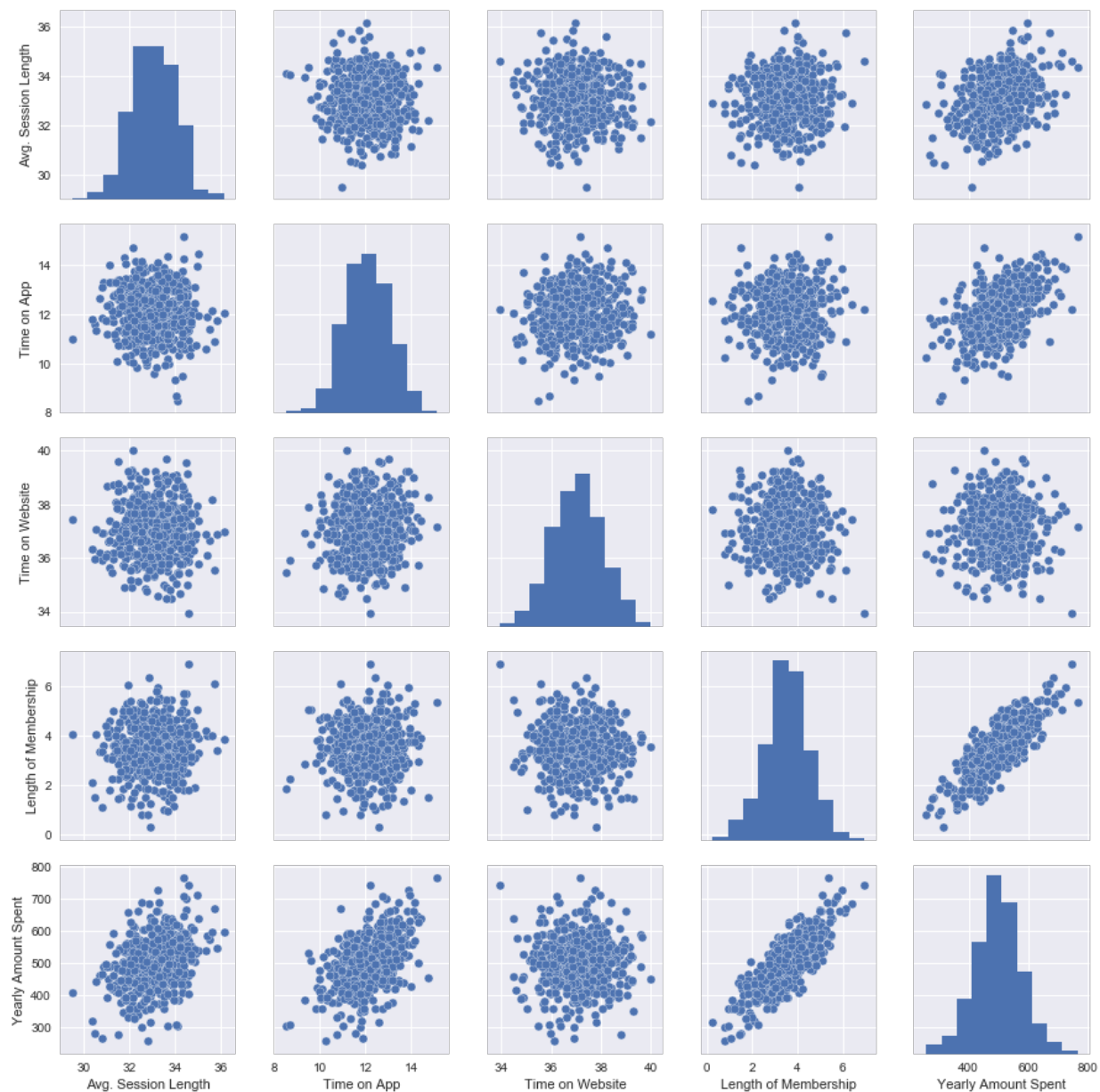
Out[283]: <seaborn.axisgrid.JointGrid at 0x130edac88>



Let's explore these types of relationships across the entire data set. Use [pairplot](https://stanford.edu/~mwaskom/software/seaborn/tutorial/axis_grids.html#plotting-pairwise-relationships-with-pairgrid-and-pairplot) (https://stanford.edu/~mwaskom/software/seaborn/tutorial/axis_grids.html#plotting-pairwise-relationships-with-pairgrid-and-pairplot) to recreate the plot below. (Don't worry about the the colors)

In [12]:

```
Out[12]: <seaborn.axisgrid.PairGrid at 0x11dfe6550>
```

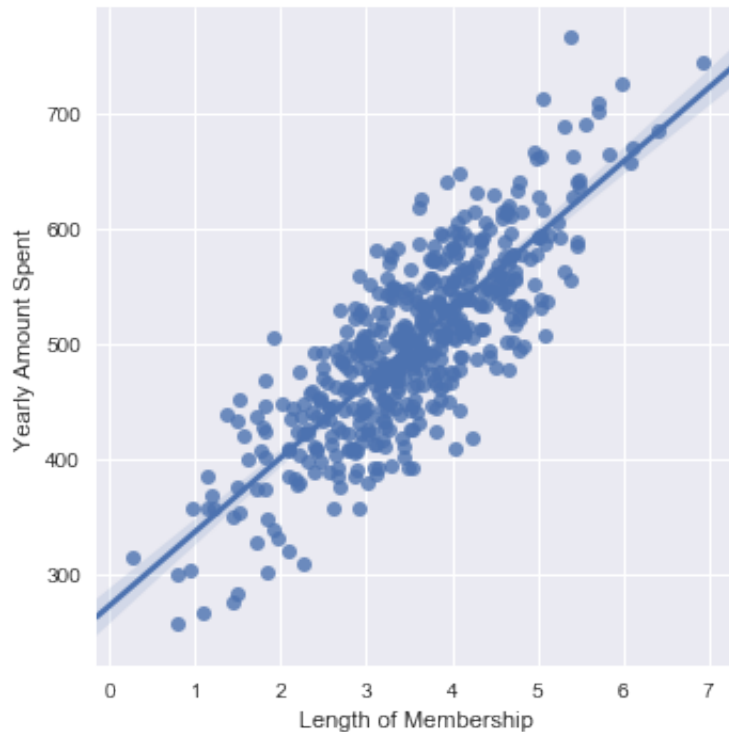


Based off this plot what looks to be the most correlated feature with Yearly Amount Spent?

Create a linear model plot (using seaborn's Implot) of Yearly Amount Spent vs. Length of Membership.

In [16]:

Out[16]: <seaborn.axisgrid.FacetGrid at 0x11f375490>



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.**

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`

Training the Model

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

Import `LinearRegression` from `sklearn.linear_model`

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

Train/fit lm on the training data.

In [23]:

```
Out[23]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Print out the coefficients of the model

In [24]:

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

Predicting Test Data

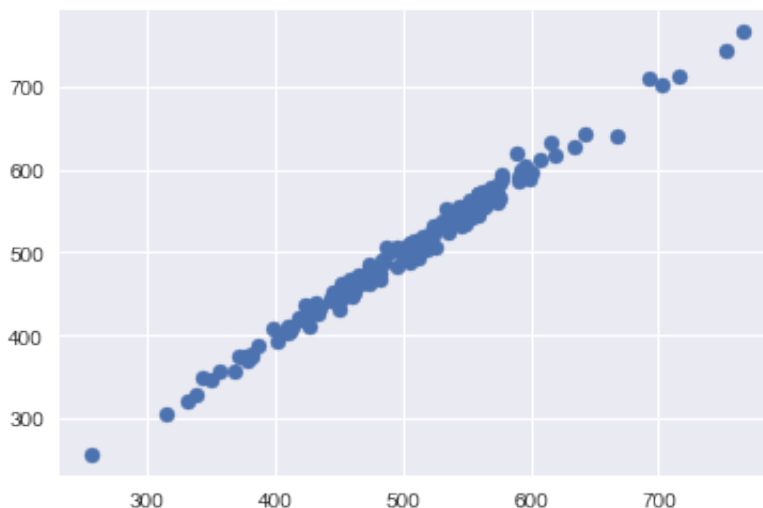
Now that we have fit our model, let's evaluate its performance by predicting off the test values!

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

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

In [27]:

```
Out[27]: <matplotlib.collections.PathCollection at 0x121a4ac10>
```



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

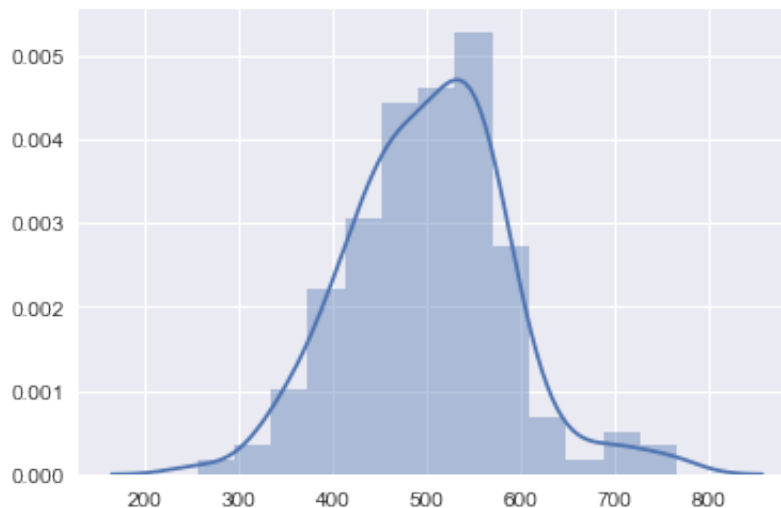
Residuals

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 [32]:

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x121a5a590>



Conclusion

We still want to figure out the answer to the original question, do we focus our effort 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 [33]:

Out[33]:

	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?

Do you think the company should focus more on their mobile app or on their website?