MANJESH D K

6ISE2

20191ISE0095

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

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
customers = pd.read_csv('/Ecommerce Customers.csv')
```

```
customers.head()
```

	Email	Address	Avatar	Avg. Session Length	Tim
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.65
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.10
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.33

```
customers.describe()
```

	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

```
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

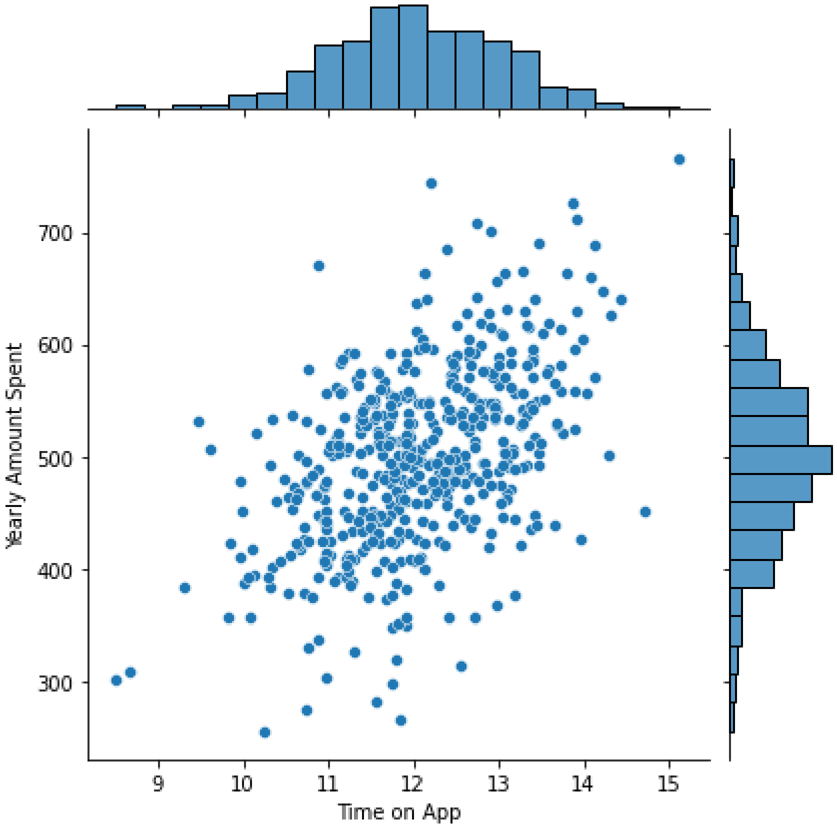
```
import seaborn as sns
```

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

```
<seaborn.axisgrid.JointGrid at 0x7f86d2621090>
```

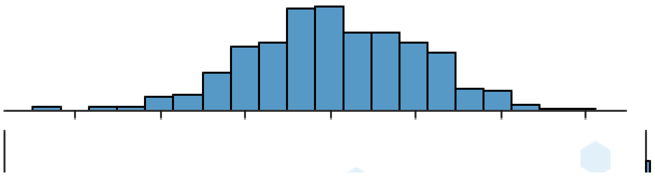
```
sns.jointplot(customers['Time on App'],customers['Yearly Amount Spent'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
  FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f86cf320090>
```



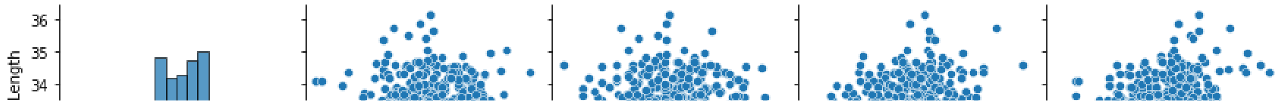
```
sns.jointplot(customers['Time on App'],customers['Yearly Amount Spent'],kind='hex')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f86d24a36d0>
```



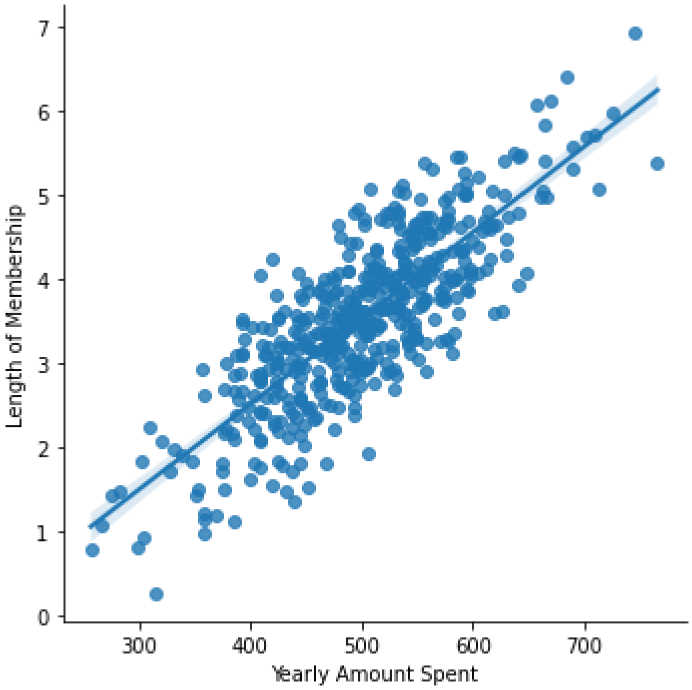
```
sns.pairplot(customers)
```

```
<seaborn.axisgrid.PairGrid at 0x7f86cf1bf7d0>
```



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

```
<seaborn.axisgrid.FacetGrid at 0x7f86ce75dc90>
```



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.

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

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

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

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

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

Training the Model

Import LinearRegression from sklearn.linear_model

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

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

```
lm = LinearRegression()
```

Train/fit lm on the training data.

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

```
     LinearRegression()
```

Print out the coefficients of the model

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

```
     Coefficients: 
      [25.98154972 38.59015876  0.19040527 61.27909654]
```

Predicting Test Data

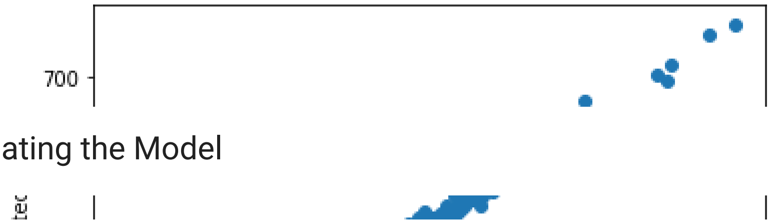
Use lm.predict() to predict off the X_test set of the data.

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

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

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

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



Evaluating the Model

```
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)))
```

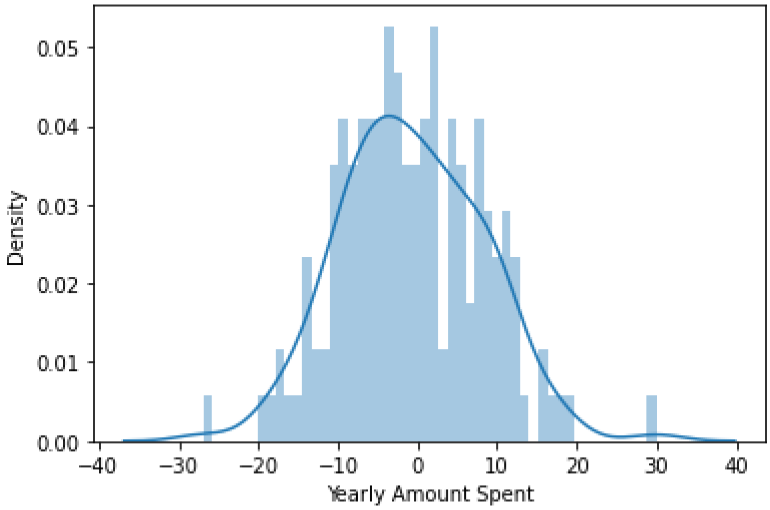
```
MAE: 7.228148667775292
MSE: 79.81305181284631
RMSE: 8.933815076038137
```

Residuals

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

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
```



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.

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

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