COMPLETED SAMPLE OF WORK DATA SCIENCE (E-Commerce Analysis)

NORIE JEANNE PEREIRA

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!

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.

Linear Regression machine learning in Python on an Ecommerce dataset

1.

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

2.

```
#Read in the Ecommerce Customers csv file as a DataFrame called customers.
customers = pd.read_csv('C:/5 Data Science Project Tuturial/2019MLAI/CSV/Ecommerce Customers.csv')
```

3.

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

out:

Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Lengt Membe
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.5776
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.2689
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.11059
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.72128
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.5366

4.
customers.describe()

out:

	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

5.

```
customers.info()
```

out:

```
<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
```

6.

```
#Exploratory Data Analysis

# we'll only be using the numerical data of the csv file.

#We will use a jointplot to compare the Time on Website and Yearly Amount Spent columns.

customers.corr()
```

out:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
Avg. Session Length	1.000000	-0.027826	-0.034987	0.060247	0.355088
Time on App	-0.027826	1.000000	0.082388	0.029143	0.499328
Time on Website	-0.034987	0.082388	1.000000	-0.047582	-0.002641
Length of Membership	0.060247	0.029143	-0.047582	1.000000	0.809084
Yearly Amount Spent	0.355088	0.499328	-0.002641	0.809084	1.000000

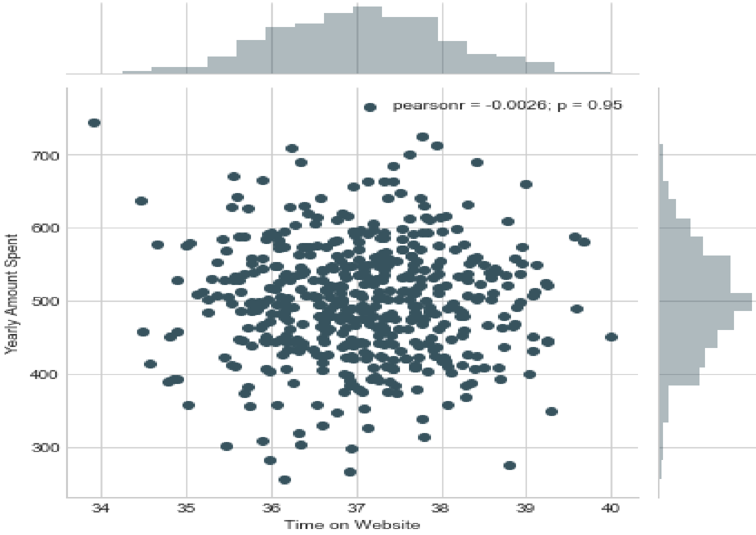
7.

```
sns.set_palette("GnBu_d")
```

8.

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

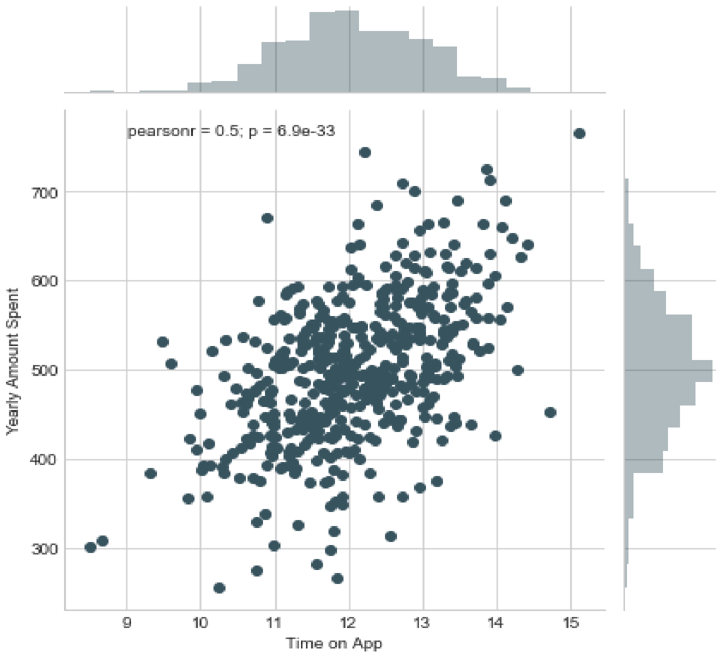
out:



9.

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

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

out:

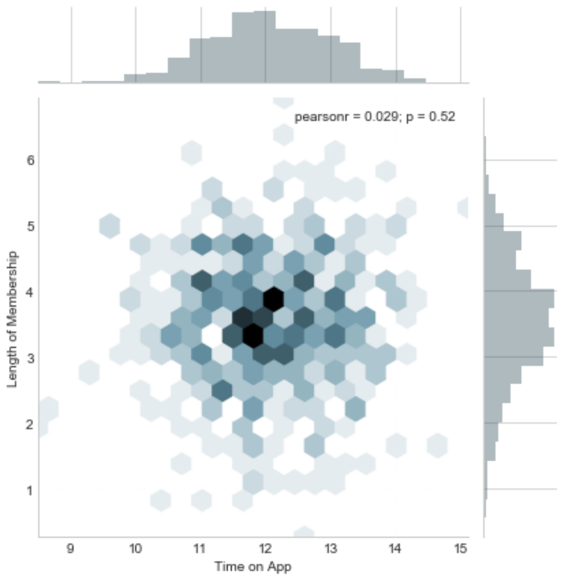


10.

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

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

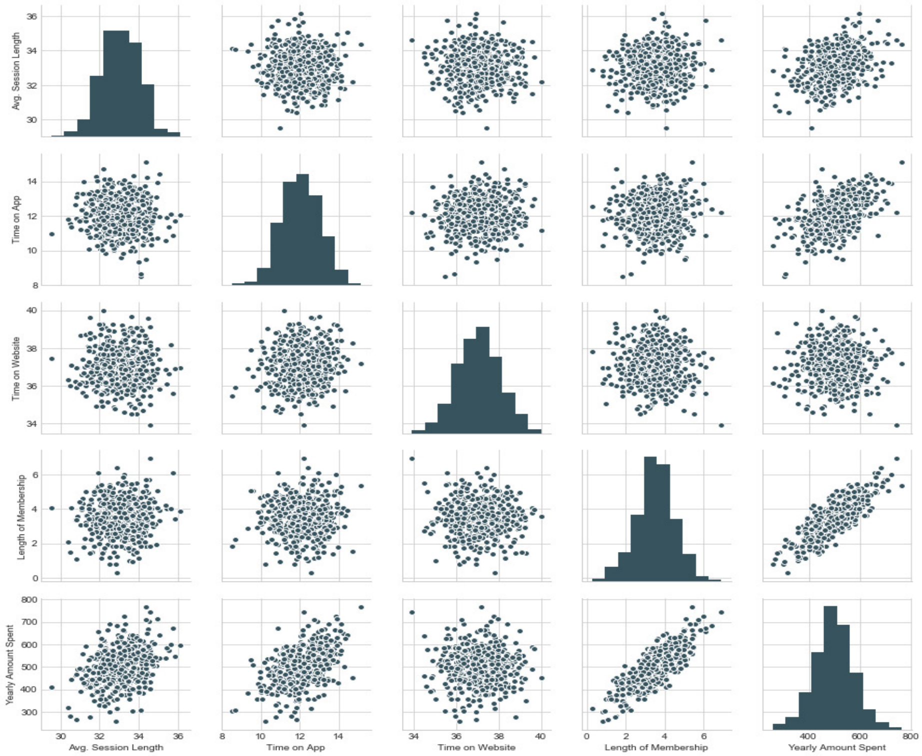
out:



11.

```
#Let's explore these types of relationships across the entire data set. Use pairplot.

sns.pairplot(customers)
```

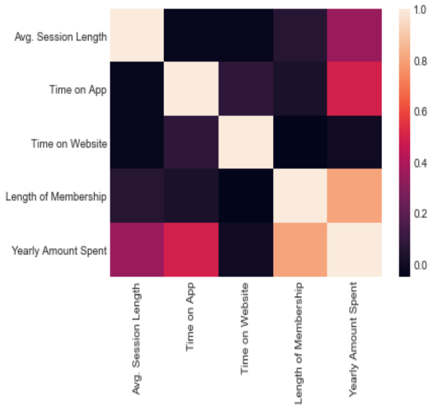
out:



12.

```
#Here we see that Length of Membership and Yearly Amount Spent are most correlated. We also see this in a heatmap.

sns.heatmap(customers.corr())
```

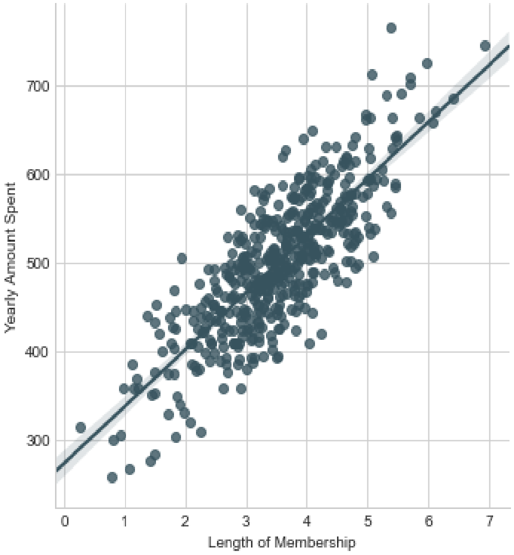
out:



13.

```
#Create a linear model plot of Yearly Amount Spent vs. Length of Membership.

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

out:



14.

```
#Training and Testing Data

#Now that we've explored the data a bit, we will split the data into training and testing sets.

#We set a variable X equal to the numerical features of the customers and a variable y equal to

#the "Yearly Amount Spent" column.

customers.columns
```

out:

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

15.

```
customers.head()
```

out:

Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Lengt Membe
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.5776
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.2689
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.1105
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.7212
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.5366

16.

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

17.

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

18.

```
#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
```

19.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
```

20.

```
#Training the Model

#Now its time to train our model on our training data.

from sklearn.linear_model import LinearRegression
```

21.

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

lm = LinearRegression()
```

22.

```
#Train/fit lm on the training data.

lm.fit(X_train,y_train)
```

out:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

23.

```
#Print out the coefficients of the model

coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficients'])

coeff_df
```

out:

	Coefficients
Avg. Session Length	25.691540
Time on App	37.892600
Time on Website	0.560581
Length of Membership	61.648594

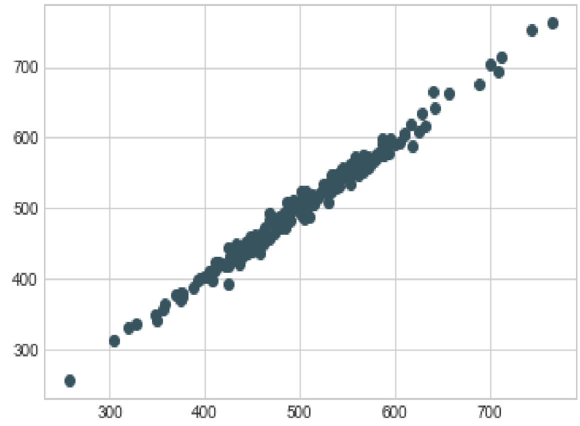
24.

```
#This indicates that one one unit of the quantities in the table, imply an increase in Yearly Amount Spent

#indicated in the table. For instance, one unit increase of Length of Membership induces 61.6 units increase

#of Yearly Amount Spent.


#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.

predictions = lm.predict(X_test)
```

25.

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

plt.scatter(y_test,predictions)
```

out:



26.

```
#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.

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)))
```

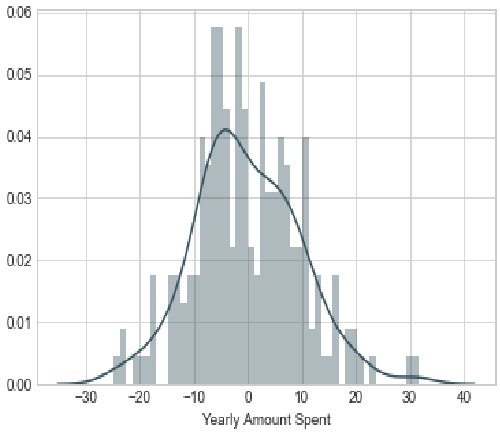
out:

```
MAE: 7.742671285838744

MSE: 93.83297800820097

RMSE: 9.686742383701601
```

27.

```
#Residuals

#Let's quickly explore the residuals to make sure everything was okay with our data.

#Plot a histogram of the residuals

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

out:



28.

```
#Conclusion

#Back to the original question, do we focus our efforts on mobile app or website development?

#Or maybe focussing on Membership length is more fruitful. Let's see if we can interpret the coefficients

#at all to get an idea.

coeff_df
```

out:

	Coefficients
Avg. Session Length	25.691540
Time on App	37.892600
Time on Website	0.560581
Length of Membership	61.648594

2.9.

#How can you interpret these coefficients?

#The coefficients indicate how many units "Yearly Amount Spent" are increased with one unit of the

#quantities given in the table.

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

#According to the data, on average, people spend significantly more time on the website,

#which does not result in spending. The app is more efficient. However, this implies that

#there is much to improve on the website. Improving the flow and usability of the website is

#likely to boost the total amount of spending.