Regresión Lineal - Ejercicio Propuesto

Acaba de obtener un contrato con una empresa de comercio electrónico con sede en la ciudad de Nueva York que vende ropa en línea, pero también tienen sesiones de asesoramiento sobre vestimenta y estilo en la tienda. Los clientes entran a la tienda, tienen sesiones / reuniones con un estilista personal, luego pueden irse a sus casas y pedir, ya sea en una aplicación móvil o en el sitio web, la ropa que desean.

La compañía está tratando de decidir si enfocar sus esfuerzos en la experiencia de su aplicación móvil o en su sitio web. ¡Te contrataron para ayudarlos a tomar las decisiones! ¡Empecemos!

Simplemente siga los pasos a continuación para analizar los datos de los clientes (son datos inventados, no se preocupe).

Importación de librerias

Importar pandas, numpy, matplotlib,y seaborn. (Importará sklearn a medida que lo necesite.)

In [1]:

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

Recuperar los datos

Trabajaremos con el archivo csv de clientes de comercio electrónico de la compañía. Tiene información del Cliente, como Correo electrónico, Dirección y su color Avatar. También tiene columnas de valores numéricos:

- Avg. Session Length: Promedio de asesoramiento de estilo en la tienda.
- Time on App: Tiempo promedio dedicado a la aplicación en minutos.
- Time on Website: Tiempo promedio dedicado al sitio web en minutos.
- Length of Membership: Cuántos años el cliente ha sido miembro.

Lea en el archivo csv de clientes de comercio electrónico como un DataFrame llamado clientes.

In [2]:

```
clientes = pd.read_csv('Ecommerce Customers')
```

Revise las primeras filas de customers, y reviselas con los métodos info() y describe().

In [3]:

```
1 clientes.head()
```

Out[3]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Tin We
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.57
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.26
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.1
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.72
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.53

In [4]:

```
1 clientes.describe()
```

Out[4]:

	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 [5]:

```
1 clientes.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
```

Análsis de Datos Exploratorios

¡Exploremos los datos!

Para el resto del ejercicio, solo utilizaremos los datos numéricos del archivo csv.

Use seaborn para crear una gráfica conjunta para comparar las columnas Time on Website y Yearly Amount Spent. ¿Tiene sentido la correlación?

In [12]:

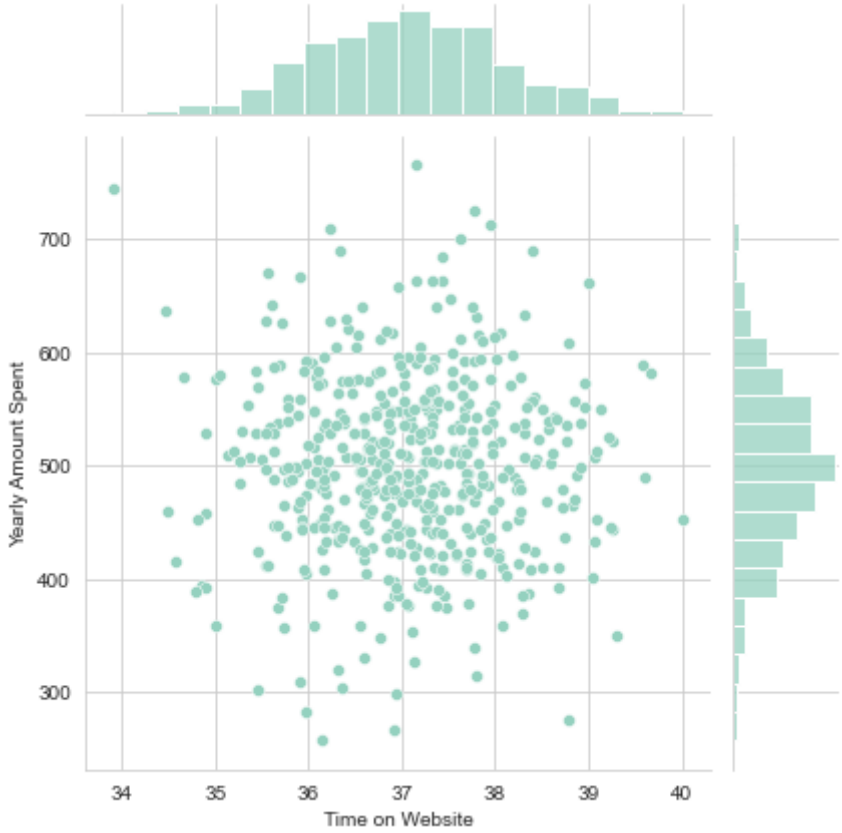
```
1 sns.set_palette("GnBu_d")
2 sns.set_style('whitegrid')
```

In [13]:

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

Out[13]:

```
<seaborn.axisgrid.JointGrid at 0x22e7dc376a0>
```

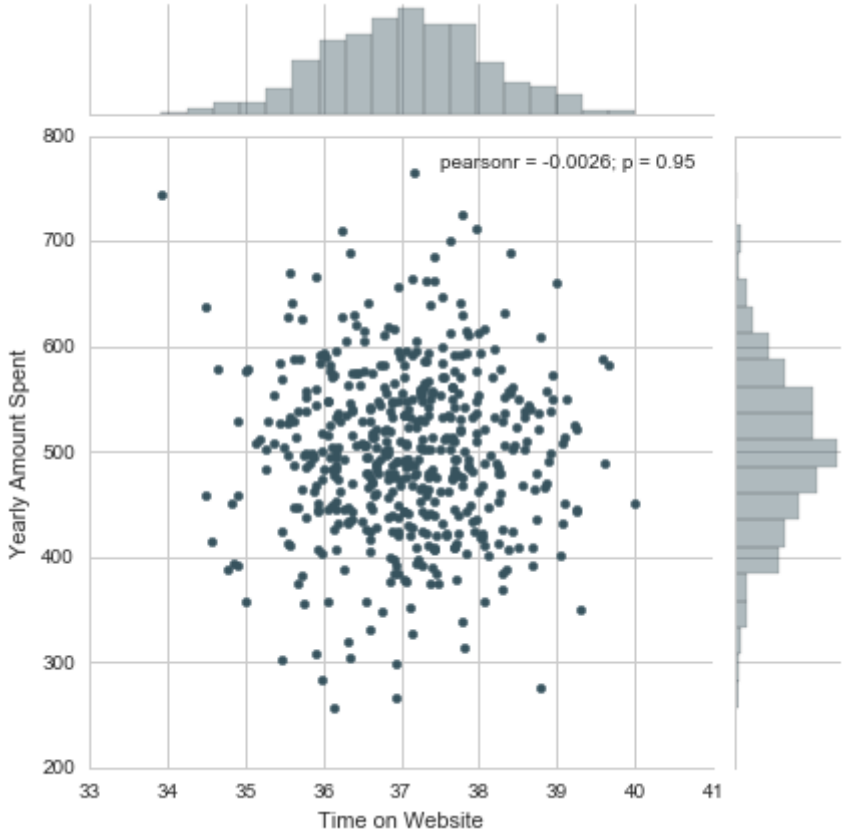


In [281]:

```
1
```

Out[281]:

```
<seaborn.axisgrid.JointGrid at 0x120bfcc88>
```



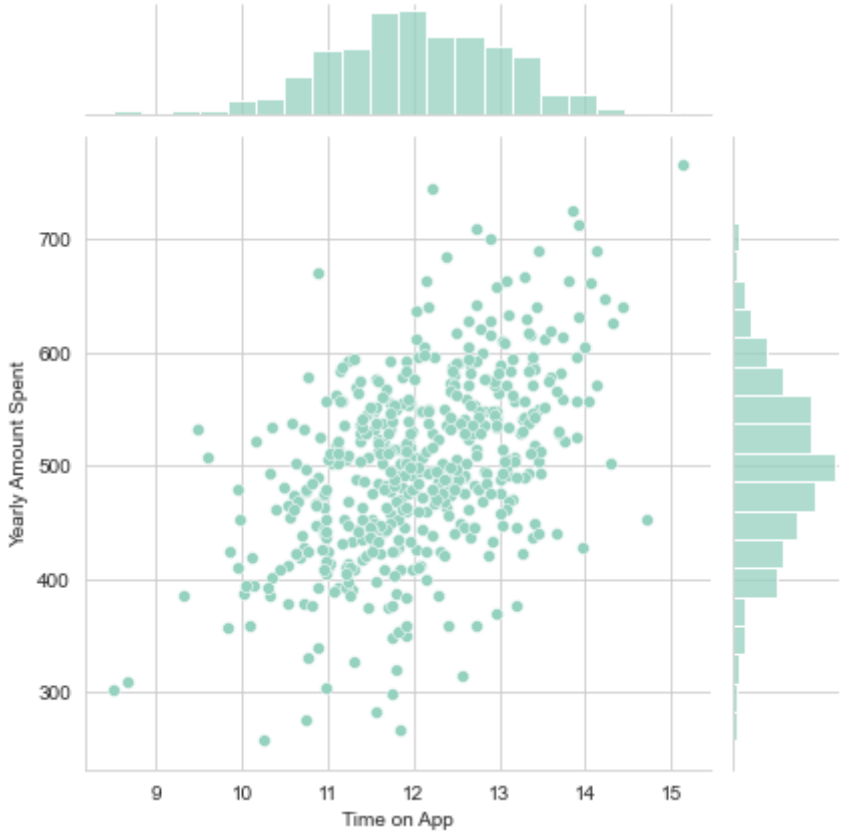
Haz lo mismo pero con la columna Time on App en su lugar.

In [14]:

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

Out[14]:

```
<seaborn.axisgrid.JointGrid at 0x22e7dd41eb0>
```



In [15]:

```
1 clientes.corr()
```

Out[15]:

	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

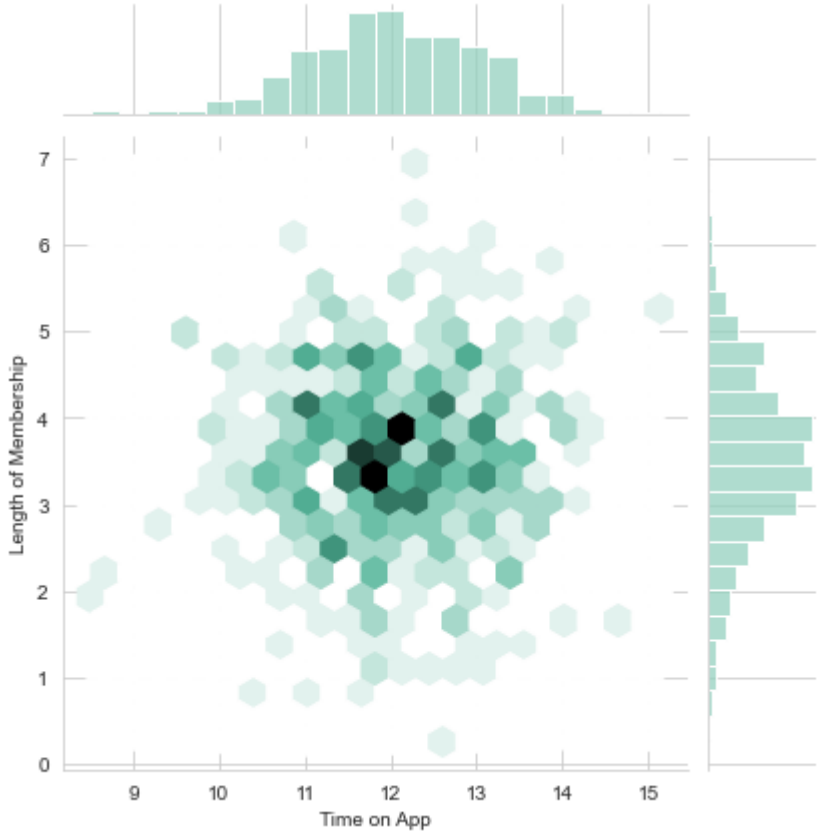
Use jointplot para crear un gráfico 2D hex bin comparando Time on App y Length of Membership.

In [16]:

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

Out[16]:

```
<seaborn.axisgrid.JointGrid at 0x22e7df2a190>
```



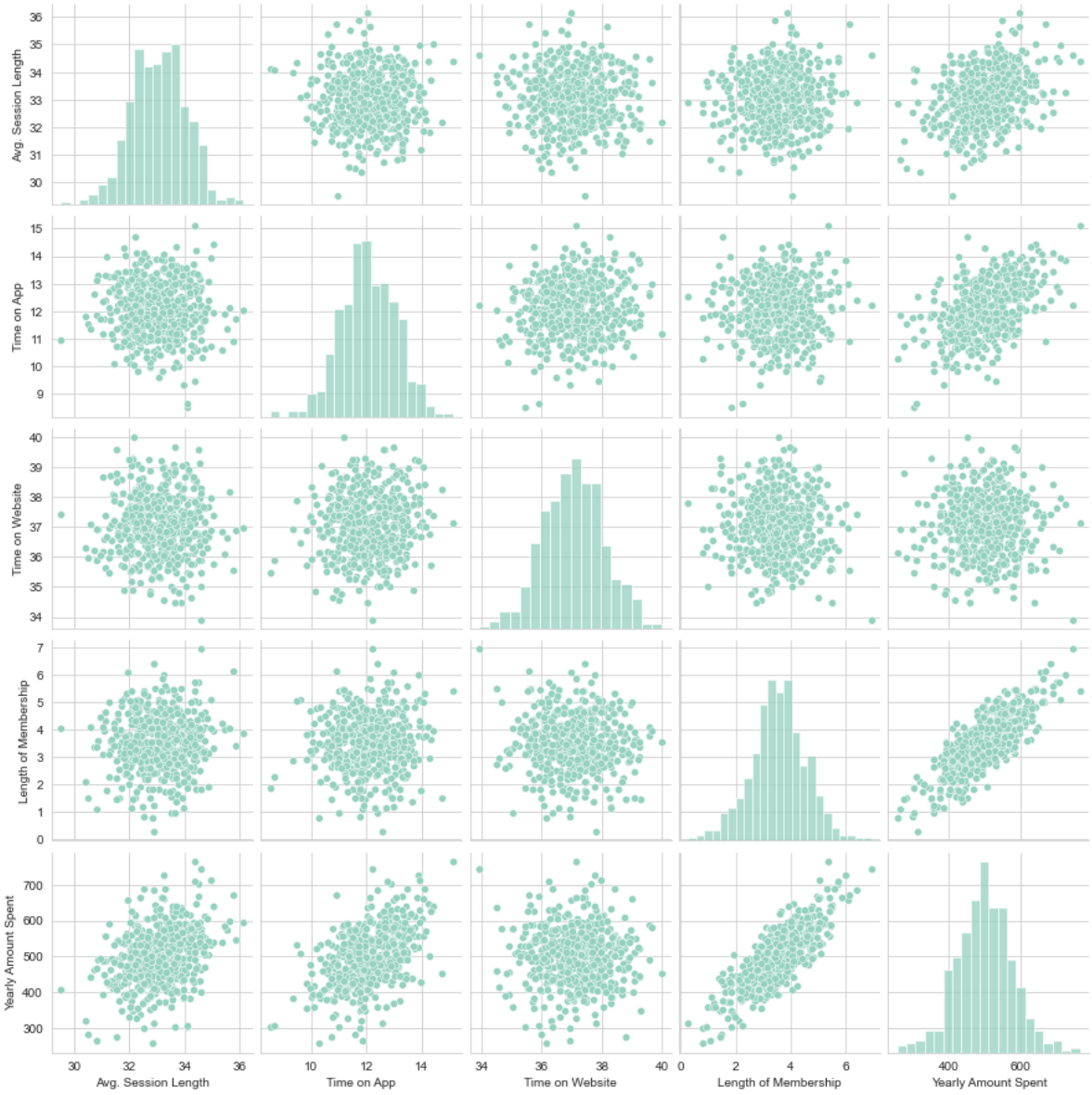
Exploremos este tipo de relaciones en todo el conjunto de datos. Usa pairplot (https://stanford.edu/~mwaskom/software/seaborn/tutorial/axis_grids.html#plotting-pairwise-relationships-with-pairgrid-and-pairplot) para recrear la gráfica de abajo. (No te preocupes por los colores)...Exploremos este tipo de relaciones en todo el conjunto de datos.

In [17]:

```
1 sns.pairplot(clientes)
```

Out[17]:

```
<seaborn.axisgrid.PairGrid at 0x22e7e75c2e0>
```



Basado en esta trama, ¿cuál parece ser la característica más correlacionada con la cantidad anual gastada?

In [18]:

```
1 # Antiguedad de la Membresía (Length of membership)
```

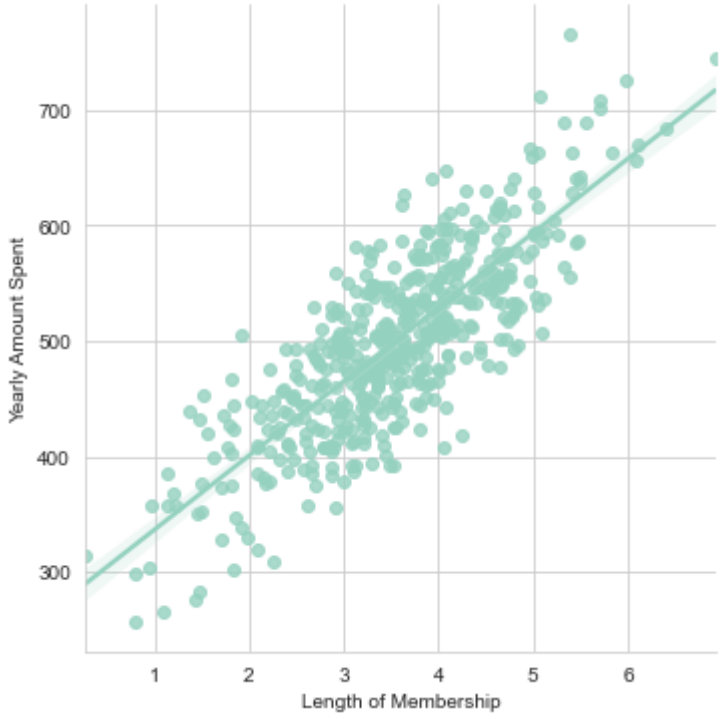
Cree un diagrama de modelo lineal (utilizando lmplot de seaborn) de Yearly Amount Spent vs. Length of Membership.

In [19]:

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

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x22e015ac460>
```



Datos de entrenamiento y prueba

Ahora que hemos explorado un poco los datos, sigamos adelante y dividamos los datos en conjuntos de entrenamiento y prueba. **Establezca una variable X igual a las características numéricas de los clientes y una variable y igual a la columna "Cantidad gastada anual".**

In [287]:

```
1
```

In [288]:

```
1
```

** Use model_selection.train_test_split de sklearn para dividir los datos en el conjunto de entrenamiento y prueba. Establezca test_size=0.3 y random_state=101**

In [289]:

```
1
```

In [290]:

```
1
```

Entrenamiento del modelo

¡Ahora es el momento de entrenar a su modelo con nuestros datos de entrenamiento!

** Importar LinearRegression desde sklearn.linear_model **

In [291]:

```
1
```

Crear una instancia del modelo LinearRegression() llamado lm.

In [292]:

```
1
```

** Entrenar/ajustar lm con los datos de entrenamiento.**

In [293]:

```
1
```

Out[293]:

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

Imprima los coeficientes del modelo

In [294]:

```
1
```

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

Predicción con los datos de prueba

Ahora que hemos ajustado nuestro modelo, ¡evaluemos su rendimiento prediciendo los valores de prueba!

** Use lm.predict () para predecir el conjunto X_test de los datos.**

In [295]:

```
1
```

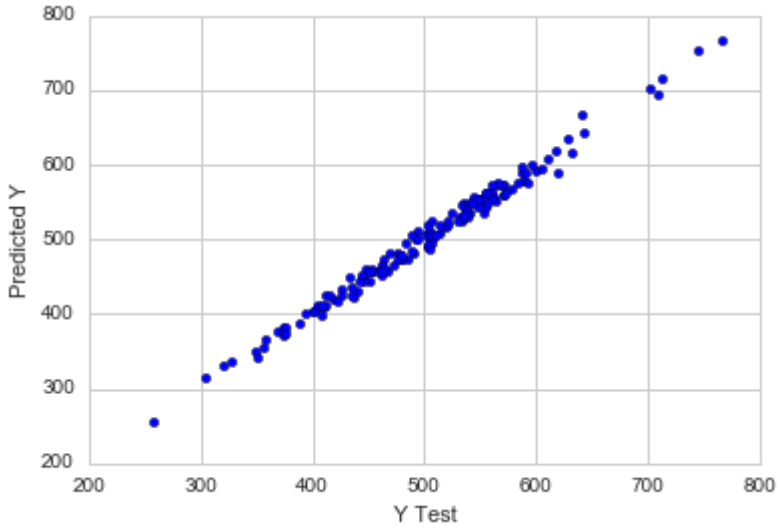
** Cree un diagrama de dispersión de los valores de prueba reales frente a los valores predichos. **

In [296]:

```
1
```

Out[296]:

```
<matplotlib.text.Text at 0x135546320>
```



Evaluación del modelo

Evaluemos el rendimiento de nuestro modelo calculando la suma residual de cuadrados y la puntuación de varianza explicada (R ^ 2).

** Calcule el error absoluto promedio, el error cuadrado promedio y la raíz del error cuadrático promedio.**

In [303]:

```
1
```

```
MAE: 7.22814865343
MSE: 79.813051651
RMSE: 8.93381506698
```

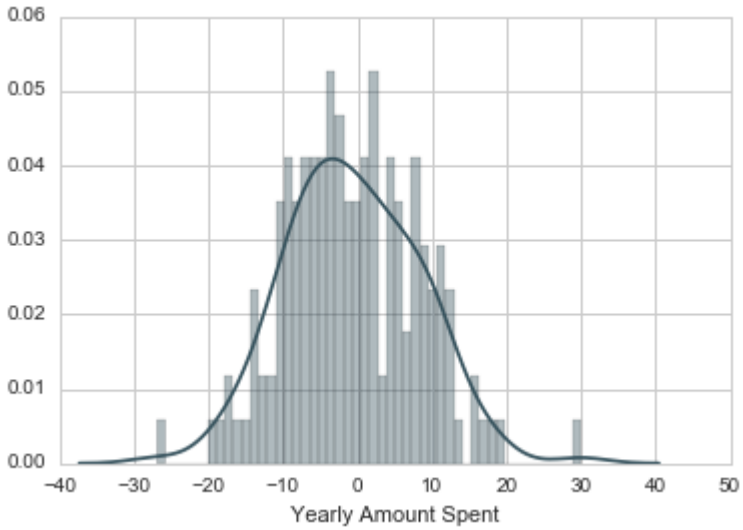
Residuales

Deberías haber obtenido un modelo muy bueno con un buen ajuste. Exploremos rápidamente los residuos para asegurarnos de que todo esté bien con nuestros datos.

Trace un histograma de los residuos y asegúrese de que se vea distribuido normalmente. Utilice ya sea distplot de seaborn o simplemente plt.hist ()

In [317]:

```
1
```



Conclusión

Todavía queremos averiguar la respuesta a la pregunta original, ¿centramos nuestros esfuerzos en el desarrollo de aplicaciones móviles o sitios web? O tal vez eso realmente no importa, y el Tiempo de Membresía es lo que es realmente importante. Veamos si podemos interpretar los coeficientes para obtener una idea.

** Recrea el dataframe de abajo. **

In [298]:

```
1
```

Out[298]:

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

** ¿Cómo puedes interpretar estos coeficientes? **

Type *Markdown* and LaTeX: α^2

¿Crees que la empresa debería centrarse más en su aplicación móvil o en su sitio web?

La respuesta aquí

¡Excelente trabajo!