NAME : Y MURALI KRISHNA

C♥LLEGE :

YEAR :

BRANCH : ECE

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in

import os


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


customers = pd.read_csv('/content/Ecommerce Customers')
display(data)
```

	Email	Address	Avatar	Avg. Session Length	Time
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795
...	...	...	...	...	
495	lewisjessica@craig-evans.com	4483 Jones Motorway Suite 872\nLake Jamiefurt,...	Tan	33.237660	13.566
496	katrina56@gmail.com	172 Owen Divide Suite 497\nWest Richard, CA 19320	PaleVioletRed	34.702529	11.695
		0787 Andrews Ranch			

```
customers.head()
```

	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

```
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
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

```
customers.info()
```

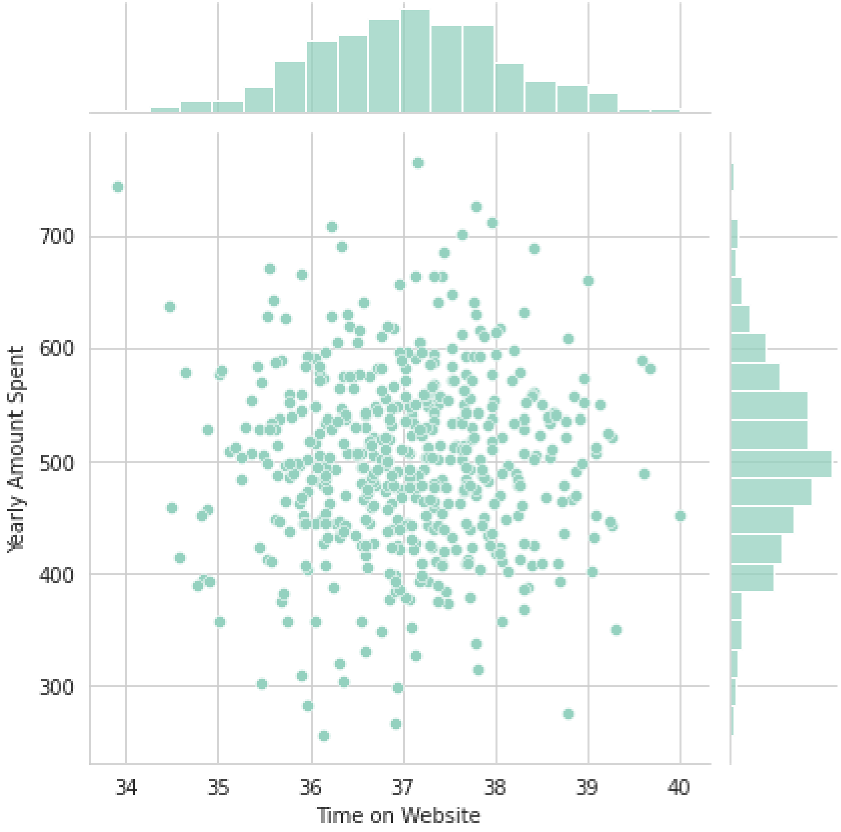
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    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
```

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

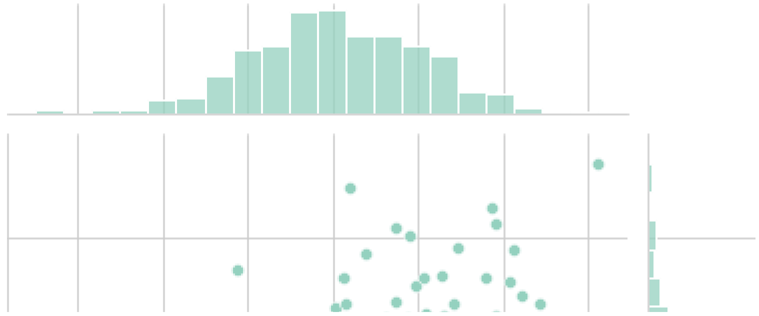
```
sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=customers)
```

```
<seaborn.axisgrid.JointGrid at 0x7feb3ed64c10>
```



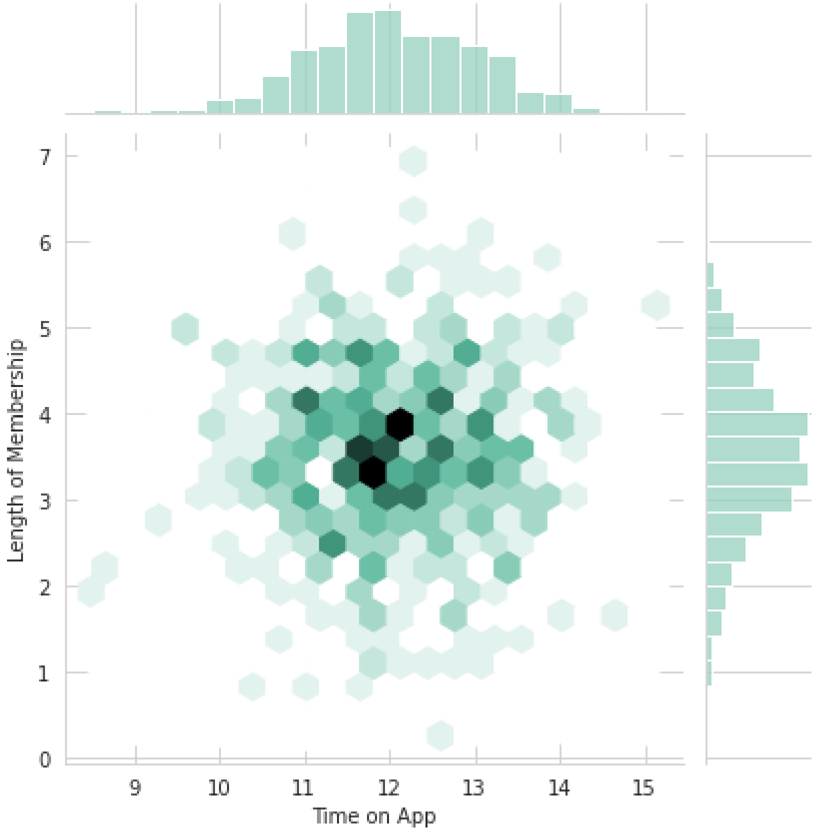
```
sns.jointplot(x='Time on App',y='Yearly Amount Spent',data=customers)
```

```
<seaborn.axisgrid.JointGrid at 0x7feb3be83750>
```



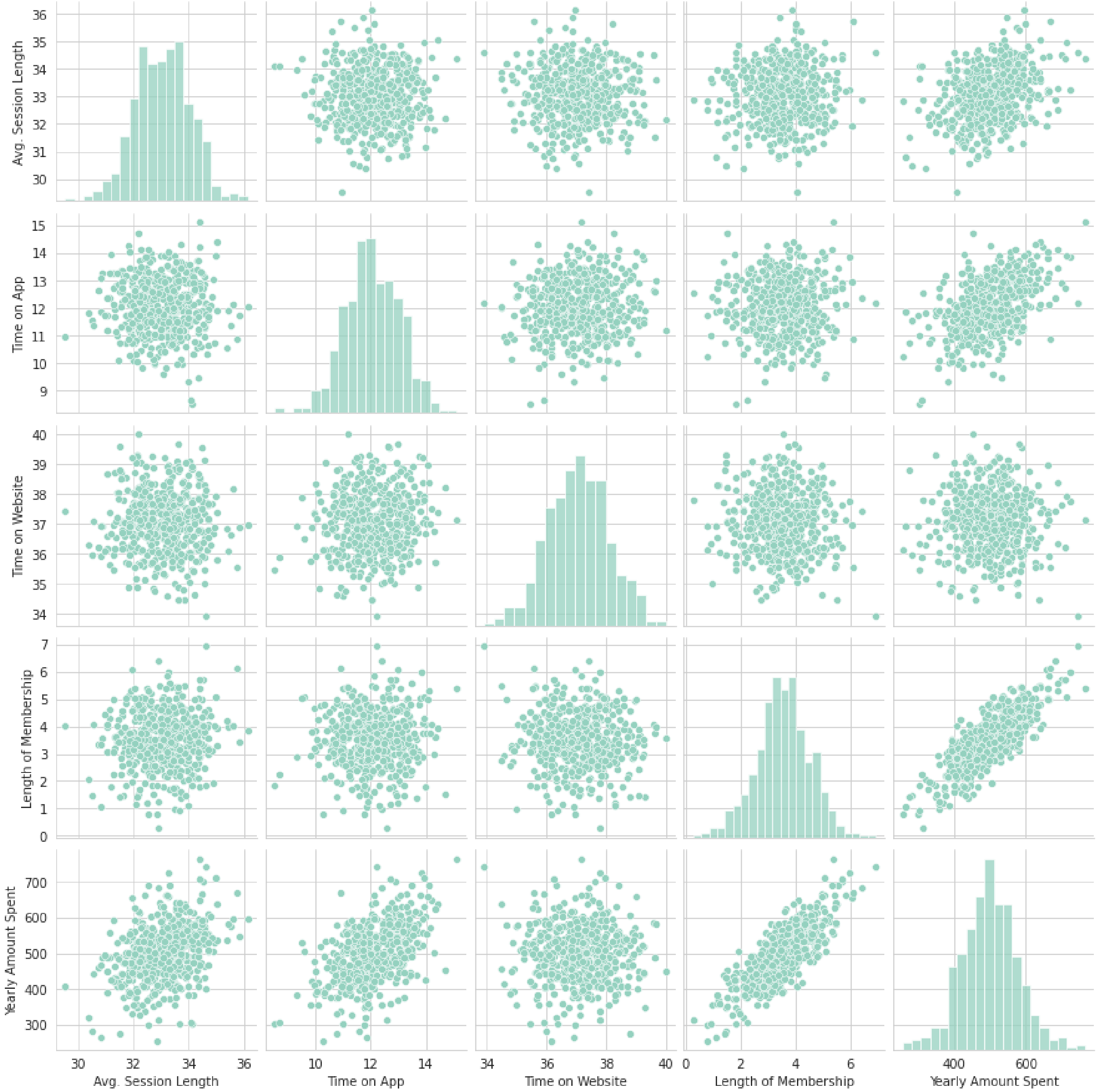
```
sns.jointplot(x='Time on App',y='Length of Membership',kind="hex",data=customers)
```

```
<seaborn.axisgrid.JointGrid at 0x7feb3c345190>
```



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

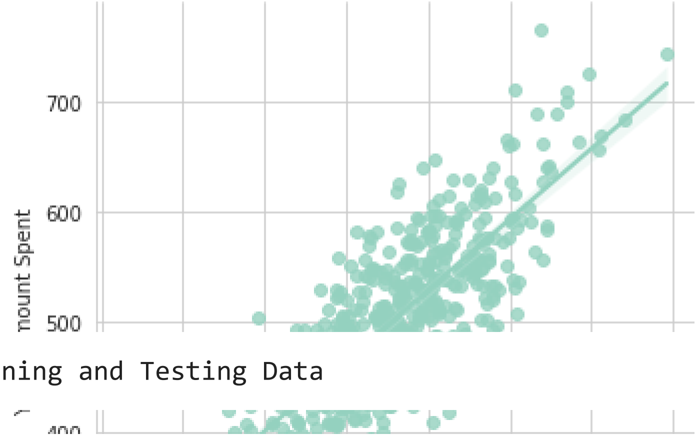
```
<seaborn.axisgrid.PairGrid at 0x7feb3bb8ab10>
```



```
# Length of Membership
```

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

```
     <seaborn.axisgrid.FacetGrid at 0x7feb3b1caf50>
```



```
#Training and Testing Data
```

```
X = customers[['Avg. Session Length','Time on App','Time on Website','Length of Membership']]
y = customers['Yearly Amount Spent']
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)
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

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

```
     LinearRegression()
```

```
lm.coef_
```

```
     array([25.98154972, 38.59015875,  0.19040528, 61.27909654])
```

```
#Predicting test data
```

```
predictions = lm.predict(X_test)
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

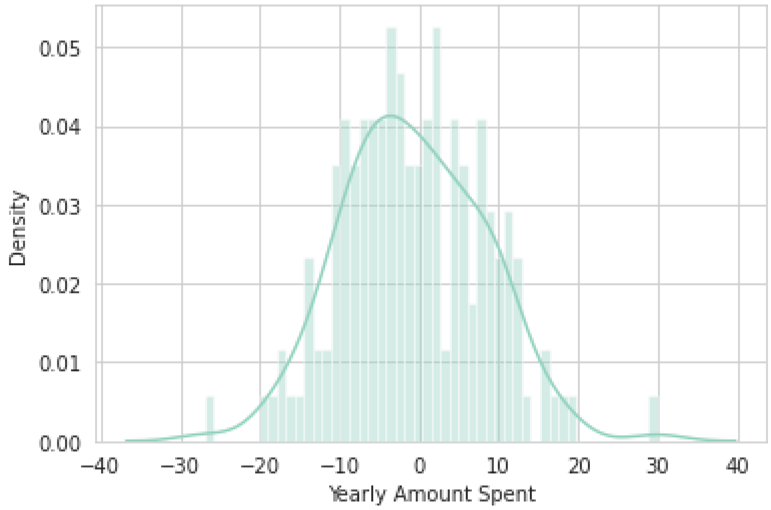
```
Text(0, 0.5, 'Predicted Y')
```

```
from sklearn import metrics
print('MAE :'," ", metrics.mean_absolute_error(y_test,predictions))
print('MSE :'," ", metrics.mean_squared_error(y_test,predictions))
print('RMAE :'," ", np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

```
MAE :   7.228148653430826
MSE :   79.81305165097427
RMAE :   8.933815066978624
```

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7feb38a5cc50>
```



```
#conclusion
```

```
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

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import cv2
import matplotlib.pyplot as plt
#from PIL import Image
# Input data files are available in the "../input/" directory.
```

```
img1 = cv2.imread('/content/7t6t65r5r.jfif')
plt.imshow(img1)
plt.show()
img2 = cv2.imread('/content/abc.jfif')
plt.imshow(img2)
plt.show()
img3 = cv2.imread('/content/komi.jfif')
plt.imshow(img3)
plt.show()
```

New Section

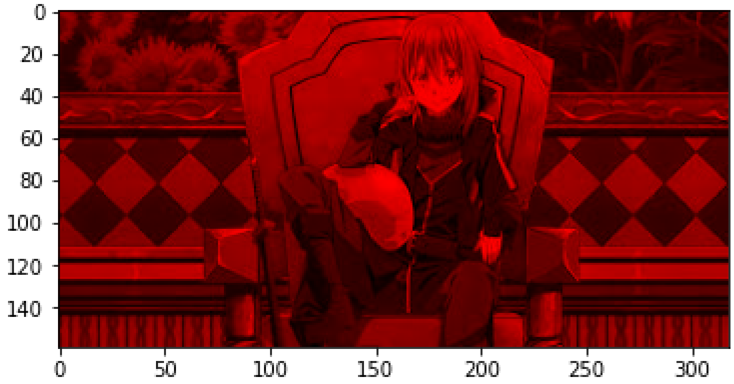
```
plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
plt.show()

b_channel, g_channel, r_channel = cv2.split(img1)
horse_gray = ((0.3 * r_channel) + (0.59 * g_channel)+ (0.11 * b_channel))
#horse_gray = cv2.merge((b_channel, g_channel, r_channel))
print(horse_gray.shape)
plt.imshow(horse_gray)
plt.show()
```

0
20
40
60
80
100
120
140

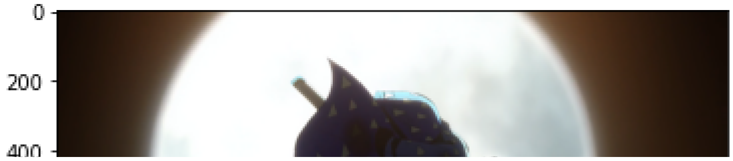
```
#img1
img2[:,:,1] = 0
img2[:,:,0] = 0
#plt.imshow(img2)

plt.imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
plt.show()
```



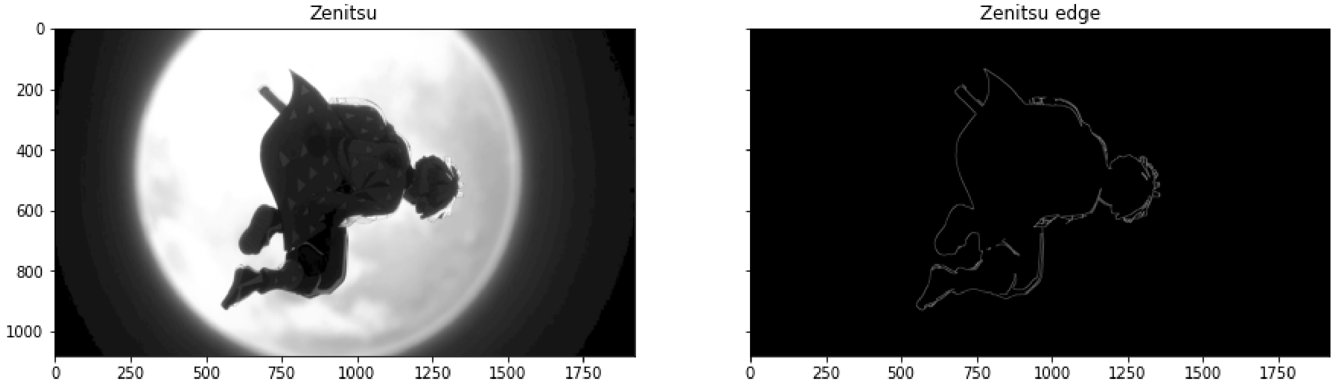
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import cv2
import matplotlib.pyplot as plt
#from PIL import Image
# Input data files are available in the "../input/" directory.


img4 = cv2.imread('/content/Screenshot (18).png')
plt.imshow(img4)
plt.show()
```



```
# function to plot n images using subplots
def plot_image(images, captions=None, cmap=None ):
    f, axes = plt.subplots(1, len(images), sharey=True)
    f.set_figwidth(15)
    for ax,image,caption in zip(axes, images, captions):
        ax.imshow(image, cmap)
        ax.set_title(caption)


img4 = cv2.imread('/content/Screenshot (18).png',0)
edges = cv2.Canny(img4,100,200)
plot_image([img4, edges], cmap='gray', captions = ["Zenitsu", "Zenitsu edge"])
```



0s completed at 10:01 AM