

Module 4 Exercises - Data Visualization

Exercise 1:

From the datasets folder, load the "dvddata.xlsx" file as a dataframe. Then rename the following columns (new column name is in parentheses):

- # of Clients (Clients)
- Visits to each Client per Month (Visits)
- Calls to each Client per month (Calls)
- Emails to each Client per month (Emails)
- # of businesses in district (Business)

```
In [1]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt # graphic Library
import seaborn as sns

# Allow "Plot to show on Jupyter Notebook" created graphic in Jupyter, not on sep
%matplotlib inline
```

```
In [2]: df = pd.read_excel("datasets/dvddata.xlsx", encoding = "ISO-8859-1") #read EXCEL
#1st column needs [encoding] to read pound/hashtag(#) symbol

df.head()

#df = pd.read_csv("datasets/Pokemon.csv", encoding = "ISO-8859-1") #read CSV.F
#1st column needs [encoding] to read pound/hashtag(#) symbol

#df.head()
```

Out[2]:

	Salesperson	District	Region	Sales	# of Clients	Visits to each Client per Month	Calls to each Client per month	Emails to each Client per month	# of businesses in district
0	Addie Miller	52	A	393	16	2	4	30	500
1	Addie Miller	67	A	267	11	2	1	14	627
2	Addie Miller	104	A	148	4	1	1	17	778
3	Alexandra Jones	49	A	317	17	0	7	1	530
4	Alonzo Tennant	46	A	276	16	2	2	22	770

```
In [3]: #This is data cleaning code
# .rename (this (#) column)
# { 'oldname': 'new name', 'oldname' : 'newname'} #give them dictionary. using key
df.rename(columns={'# of Clients': 'Clients'}, inplace =True)
df.head()
```

Out[3]:

	Salesperson	District	Region	Sales	Clients	Visits to each Client per Month	Calls to each Client per month	Emails to each Client per month	# of businesses in district
0	Addie Miller	52	A	393	16	2	4	30	500
1	Addie Miller	67	A	267	11	2	1	14	627
2	Addie Miller	104	A	148	4	1	1	17	778
3	Alexandra Jones	49	A	317	17	0	7	1	530
4	Alonzo Tennant	46	A	276	16	2	2	22	770

```
In [6]: df.columns
```

Out[6]: Index(['Salesperson', 'District', 'Region', 'Sales', 'Clients',
 'Visits to each Client per Month', 'Calls to each Client per month',
 'Emails to each Client per month', '# of businesses in district'],
 dtype='object')

```
In [4]: df.rename(columns = {'Visits to each Client per Month': 'Visit', 'Calls to each  
 'Emails to each Client per month': 'Email',  
 '# of businesses in district': 'Business'},  
 inplace = True)
```

```
In [5]: df.head()
```

Out[5]:

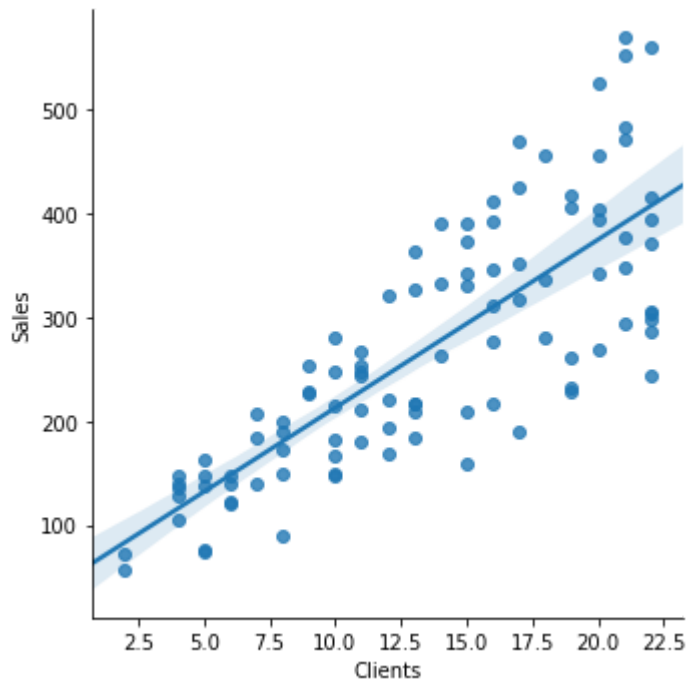
	Salesperson	District	Region	Sales	Clients	Visit	Call	Email	Business
0	Addie Miller	52	A	393	16	2	4	30	500
1	Addie Miller	67	A	267	11	2	1	14	627
2	Addie Miller	104	A	148	4	1	1	17	778
3	Alexandra Jones	49	A	317	17	0	7	1	530
4	Alonzo Tennant	46	A	276	16	2	2	22	770

Exercise 2:

Using the dataframe from the previous exercise, use the Seaborn library to create a scatterplot of the number of clients compared to the sales.

```
In [9]: sns.lmplot(x='Clients', y='Sales', data=df)
```

```
Out[9]: <seaborn.axisgrid.FacetGrid at 0x1f711f3a710>
```



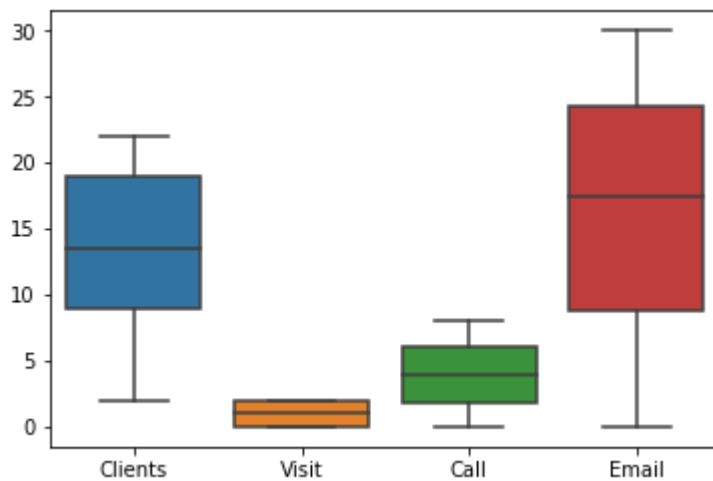
Exercise 3:

Using the dataframe in the previous exercise, use the Seaborn library to make a boxplot of the Clients, Visits, Calls, and Emails columns. (**Hint:** Make a dataframe that only contains those columns)

```
In [10]: newdf = df[['Clients', 'Visit', 'Call', 'Email']]
```

```
In [11]: sns.boxplot(data=newdf)
#sns.boxplot(data=[['Clients', 'Visits', 'Calls', 'Emails']]) #have to fix above
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1f713943b00>
```

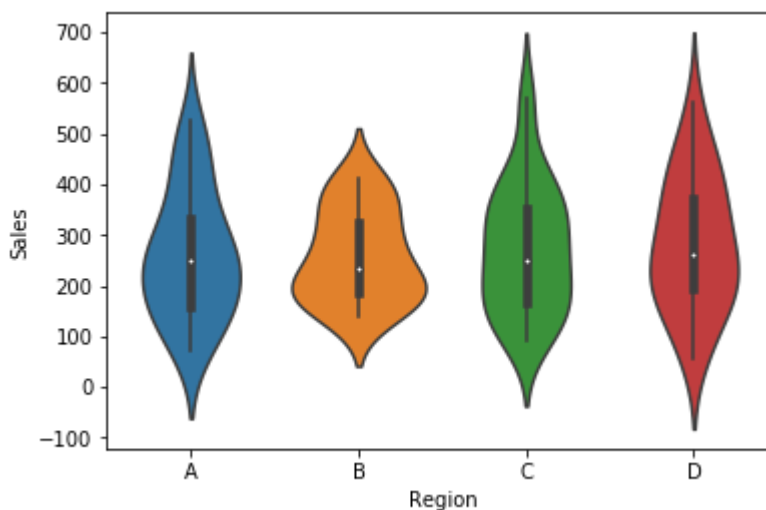


Exercise 4:

Using the dataframe from Exercise 3, use the Seaborn library to make a violin plot of sales per region.

```
In [27]: sns.violinplot(x='Region', y='Sales', data=df)
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1a17fbacf8>
```

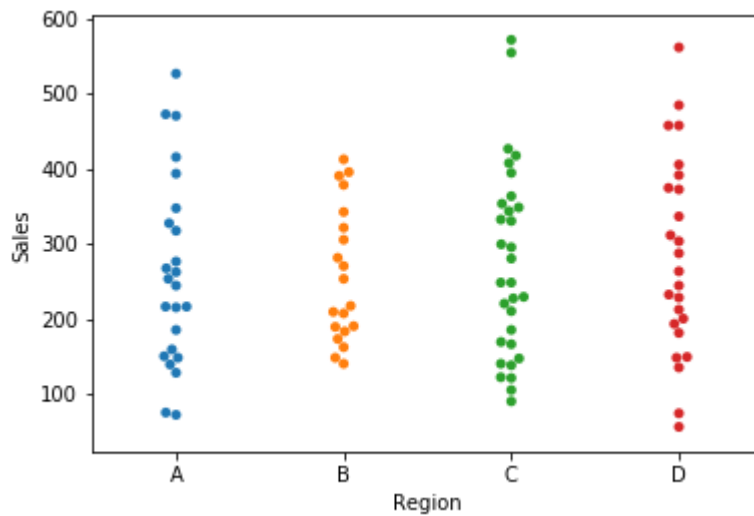


Exercise 5:

Using the dataframe in the previous exercise, create a swarm plot of sales per region.

```
In [30]: sns.swarmplot(x='Region', y='Sales', data=df)
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x1a18155c50>
```



Exercise 6:

Using the dataframe in the previous exercise, make a correlation heatmap.

```
In [33]: corr = df.corr()
sns.heatmap(corr, vmin=-1, annot=True)
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1a18191b00>
```

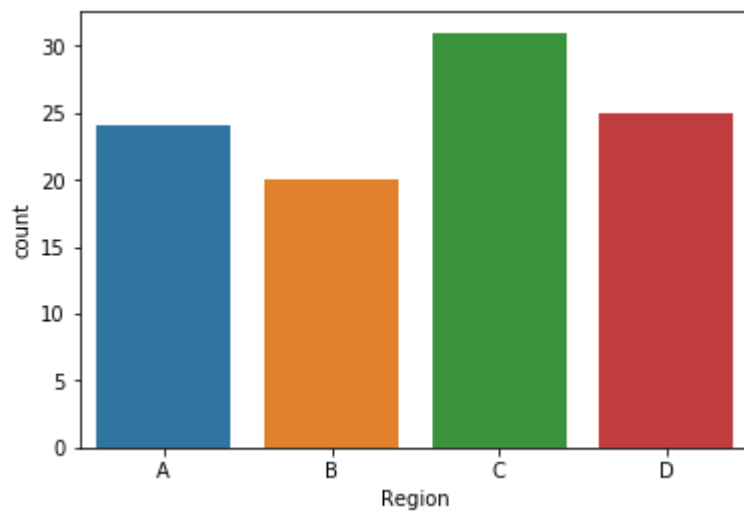


Exercise 7:

Using the dataframe in the previous exercise, make a bar chart (count plot) of the regions.

```
In [35]: sns.countplot(x='Region', data=df)
```

```
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x1a18450860>
```



Bonus

Feel free to explore any other data visualizations using the Seaborn library.

Reference: https://en.wikipedia.org/wiki/Box_plot (https://en.wikipedia.org/wiki/Box_plot)

```
In [ ]:
```