

## Python Chilla Pandas Assignment

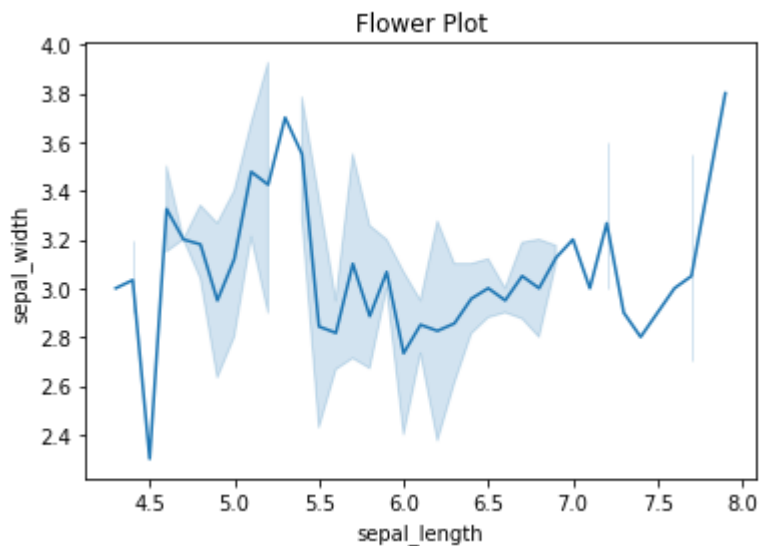
Title= "Mr"\ Name= "Ali Nawaz"\ email = "nawazk99@gmail.com"\ whatsapp  
= "03358043653"\ Artificial Intelligence Engineer at NUST\ Education : Master  
in Software Engineering

```
In [ ]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [ ]: pholl = sns.load_dataset("titanic")
pholl.head()
```

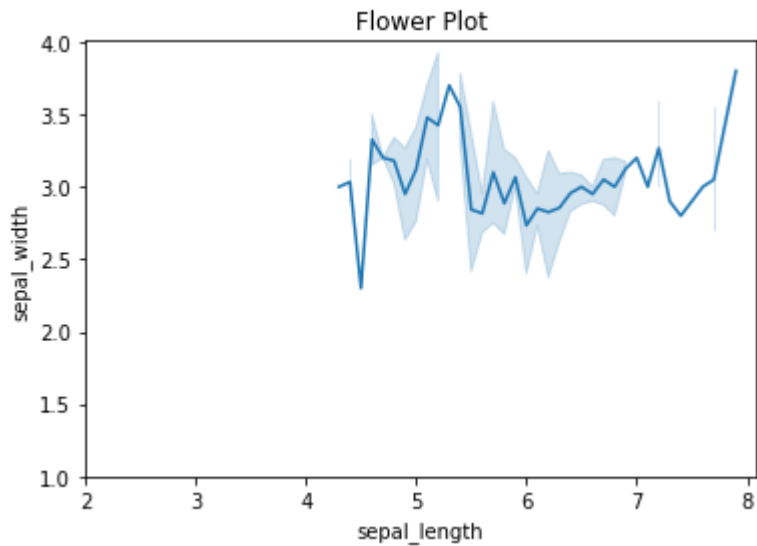
```
Out[ ]:   survived  pclass    sex  age  sibsp  parch    fare  embarked  class  who  adult_male  deck  e
0         0        3  male  22.0     1     0   7.2500         S  Third   man         True  NaN  S
1         1        1 female  38.0     1     0  71.2833         C   First  woman        False    C
2         1        3 female  26.0     0     0   7.9250         S  Third  woman        False  NaN  S
3         1        1 female  35.0     1     0  53.1000         S   First  woman        False    C  S
4         0        3  male  35.0     0     0   8.0500         S  Third   man         True  NaN  S
```

```
In [ ]: sns.lineplot(x='sepal_length', y = "sepal_width", data=pholl)
plt.title("Flower Plot")
plt.show()
```



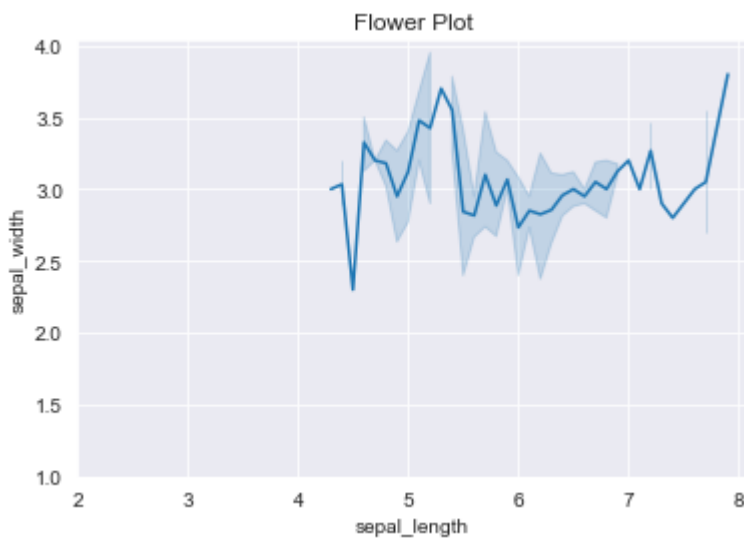
```
In [ ]: sns.lineplot(x='sepal_length', y = "sepal_width", data=pholl)
plt.title("Flower Plot")
plt.xlim(2)
```

```
plt.ylim(1)
plt.show()
```



In [ ]:

```
sns.set_style(style= "darkgrid")
sns.lineplot(x='sepal_length', y = "sepal_width", data=pholl)
plt.title("Flower Plot")
plt.xlim(2)
plt.ylim(1)
plt.show()
```



In [ ]:

```
# fig size
pholl = sns.load_dataset("iris")
plt.figure(figsize=(8,6))
sns.set_style(style= "darkgrid")
sns.lineplot(x='sepal_length', y = "sepal_width", data=pholl)
plt.title("Flower Plot")
plt.show()
```



In [ ]:

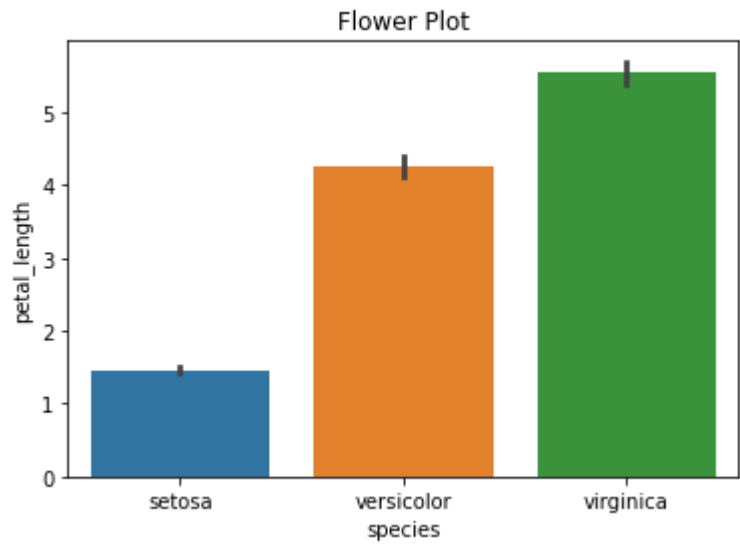
In [ ]:

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
pholl = sns.load_dataset("iris")
pholl.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [ ]:

```
pholl = sns.load_dataset("iris")
sns.barplot(x='species', y = "petal_length", data=pholl)
plt.title("Flower Plot")
plt.show()
```



```
In [ ]: pholl = sns.load_dataset("iris")
sns.barplot(x='species', y = "sepal_width", data=pholl)
plt.title("Flower Plot")
plt.show()
```

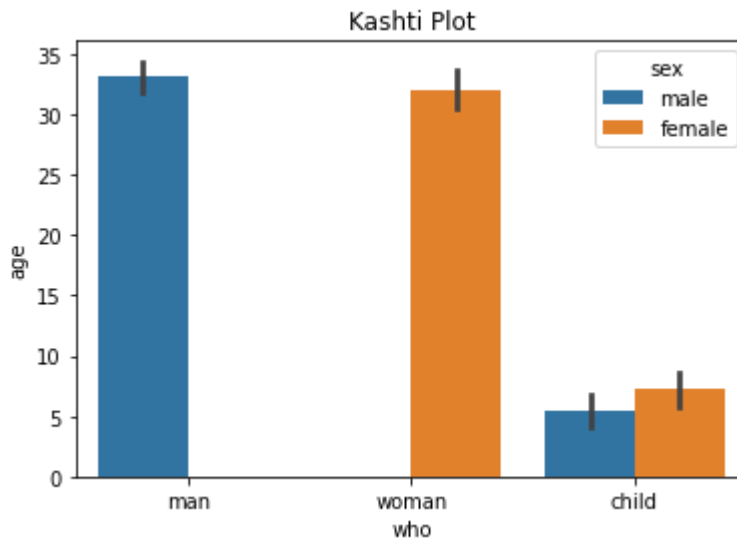


```
In [ ]: kashti = sns.load_dataset("titanic")
kashti.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	9
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	9
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	9
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	9

```
In [ ]:
```

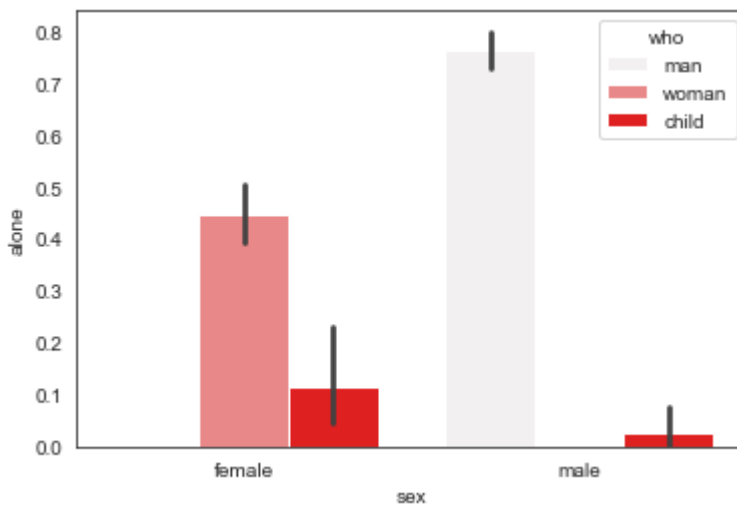
```
sns.barplot(x='who', y = "age", hue='sex', data=kashti)
plt.title("Kashti Plot")
plt.show()
```



In [ ]:

```
# color change with ordering
kashti = sns.load_dataset("titanic")

sns.barplot(x='sex', y = "alone", hue='who', data =kashti, order=["female", "male"], co
plt.show()
```

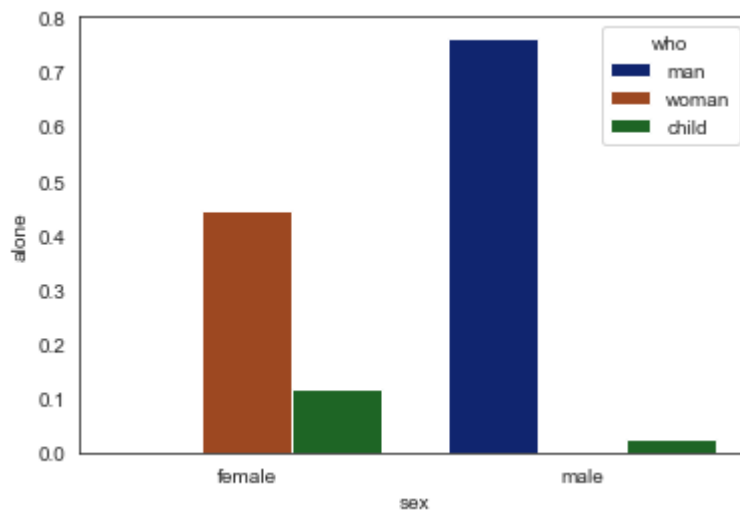


In [ ]:

```
# color pallets with ordering
# deep , muted , pastel , bright , dark , and colorblind

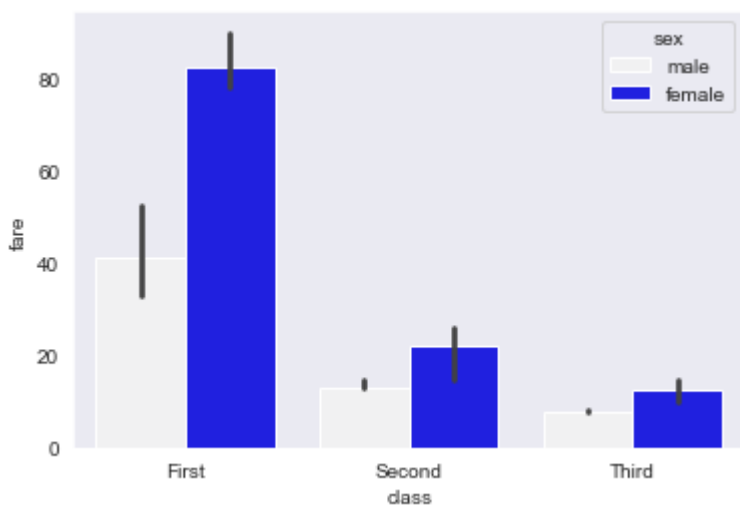
kashti = sns.load_dataset("titanic")

sns.barplot(x='sex', y = "alone", hue='who', data =kashti, order=["female", "male"], co
plt.show()
```



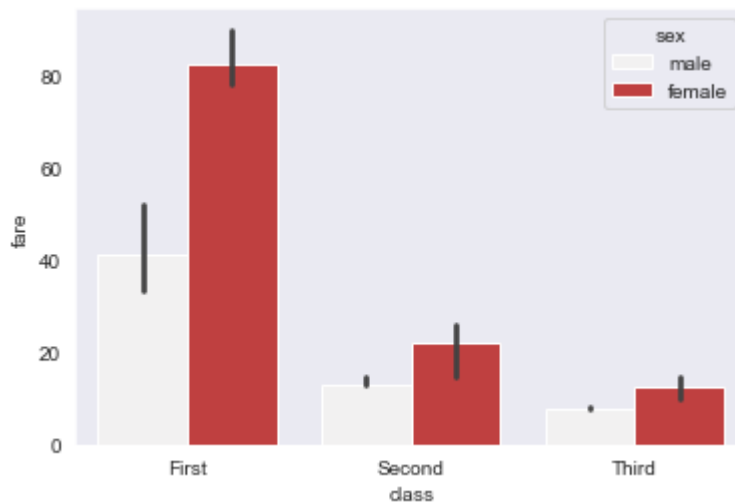
```
In [ ]: # estimators by using numpy and importing from that

sns.barplot(x='class', y = "fare", hue='sex', data =kashti, color="blue", estimator= np
plt.show()
```



```
In [ ]: # by adding saturation we can increase and decrease contrast

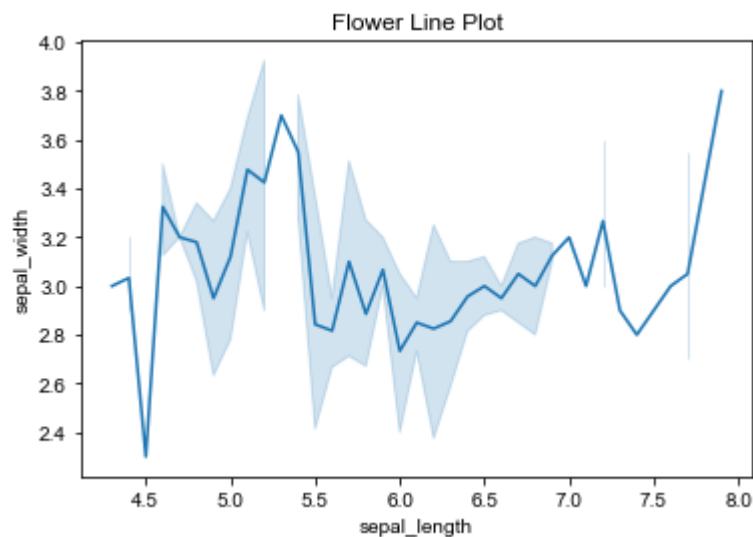
sns.set_style(style='dark')
sns.barplot(x='class', y = "fare", hue='sex', data =kashti, color="red", estimator= np.
plt.show()
```



```
In [ ]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [ ]: pholl = sns.load_dataset("iris")

sns.lineplot(x='sepal_length', y = "sepal_width", data=pholl)
sns.set_style('dark')
plt.title("Flower Line Plot")
plt.show()
```

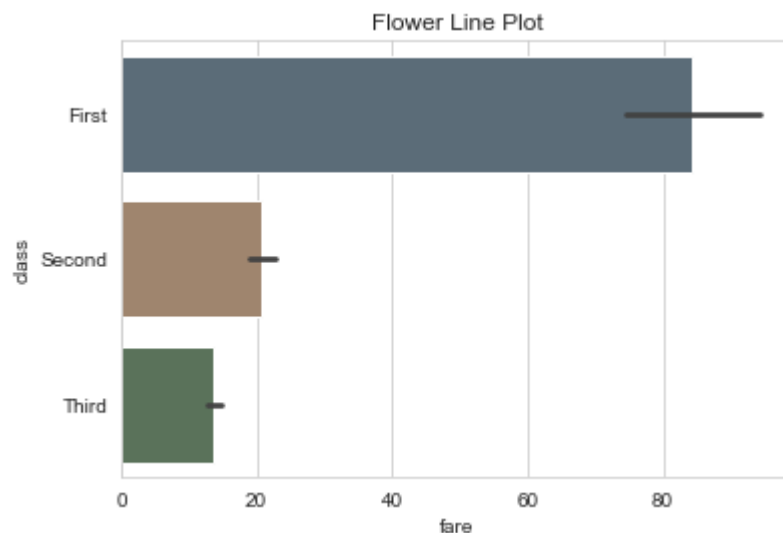


```
In [ ]: kashti = sns.load_dataset("titanic")
print(kashti.head(5))
sns.barplot(x='fare', y = "class", data=kashti, saturation=0.2)
sns.set_style('dark')
plt.title("Flower Line Plot")
plt.show()
```

survived	pclass	sex	age	sibsp	parch	fare	embarked	class
0	0	3	male	22.0	1	0	7.2500	S Third

1	1	1	female	38.0	1	0	71.2833	C	First
2	1	3	female	26.0	0	0	7.9250	S	Third
3	1	1	female	35.0	1	0	53.1000	S	First
4	0	3	male	35.0	0	0	8.0500	S	Third

	who	adult_male	deck	embark_town	alive	alone
0	man	True	NaN	Southampton	no	False
1	woman	False	C	Cherbourg	yes	False
2	woman	False	NaN	Southampton	yes	True
3	woman	False	C	Southampton	yes	False
4	man	True	NaN	Southampton	no	True



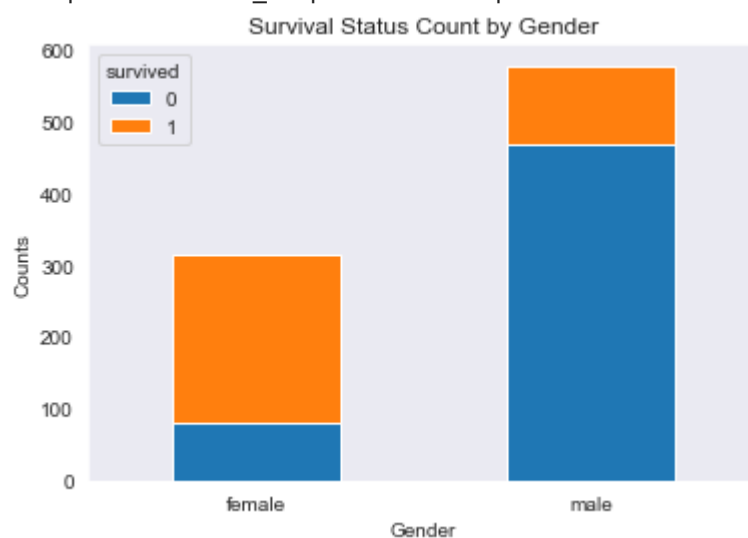
In [ ]:

```
# Load the two necessary column
df = sns.load_dataset('titanic').loc[:, ['sex', 'survived']]

# create a pivot table
dfp = df.pivot_table(index='sex', columns=['survived'], aggfunc=len)

# plot the dataframe
dfp.plot(kind='bar', stacked=True, ylabel='Counts', xlabel='Gender',
         title='Survival Status Count by Gender', rot=0)
```

&lt;matplotlib.axes.\_subplots.AxesSubplot at 0x2c5a2ba4860&gt;



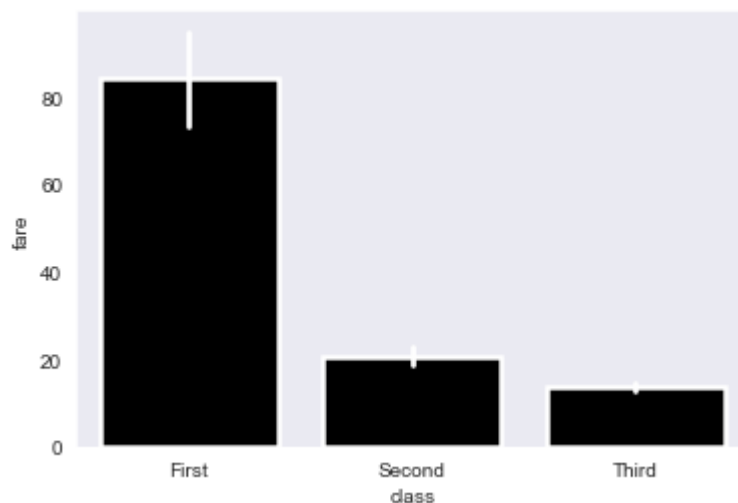
In [ ]:

```
# errcolor='0.2' 0 balck and 1 is white
```



```
# edgecolor = '0.2' 0 to one
sns.barplot(x='class', y = "fare", data=kashti, linewidth = 2.5, facecolor = (0,0,0,1),
```

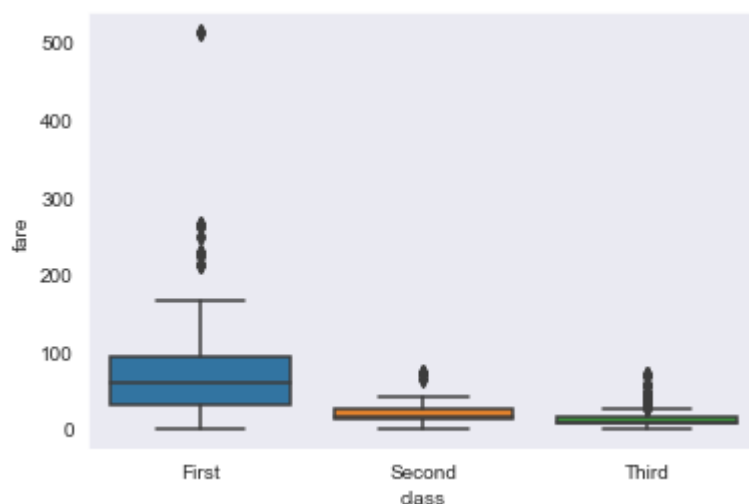
<matplotlib.axes.\_subplots.AxesSubplot at 0x2c5a31987f0>



## Box Plot

```
In [ ]: sns.boxplot(x='class', y = "fare", data=kashti)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x2c5a299efd0>

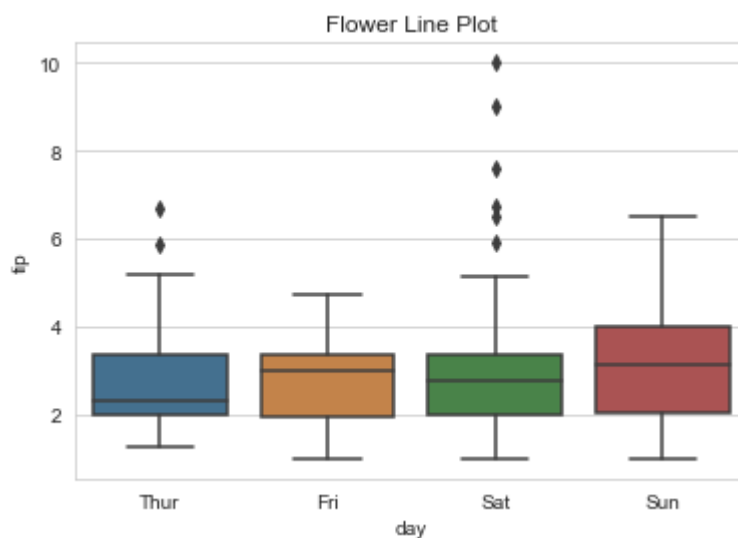


```
In [ ]: df = sns.load_dataset("tips")
df.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
In [ ]: df = sns.load_dataset("tips")

sns.boxplot(x='day', y = "tip", data=df, saturation=0.5)
sns.set_style(style='whitegrid')
plt.title("Flower Line Plot")
plt.show()
```

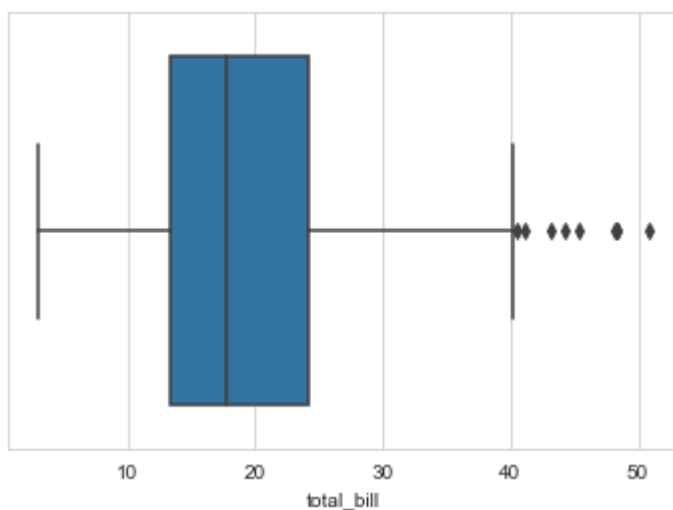


```
In [ ]: df = sns.load_dataset("tips")
df.describe()
```

	total_bill	tip	size
<b>count</b>	244.000000	244.000000	244.000000
<b>mean</b>	19.785943	2.998279	2.569672
<b>std</b>	8.902412	1.383638	0.951100
<b>min</b>	3.070000	1.000000	1.000000
<b>25%</b>	13.347500	2.000000	2.000000
<b>50%</b>	17.795000	2.900000	2.000000
<b>75%</b>	24.127500	3.562500	3.000000
<b>max</b>	50.810000	10.000000	6.000000

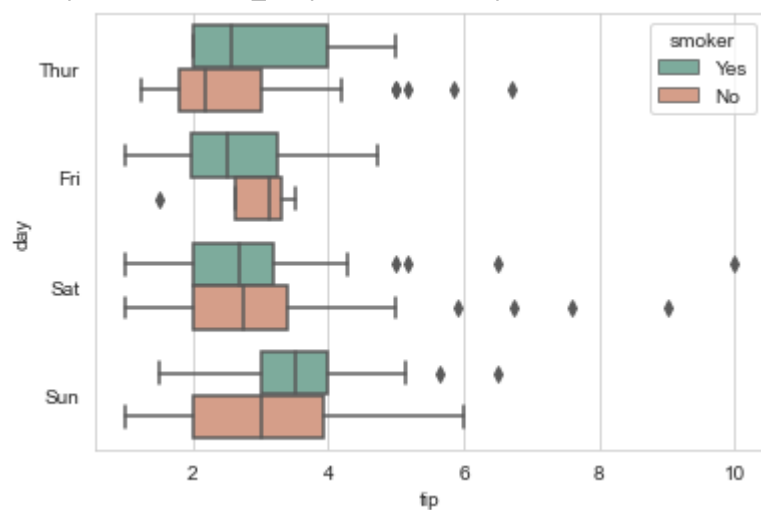
```
In [ ]: # single variable box plot only work with numerical and will not add any other stuff
sns.boxplot(x=df['total_bill'])

<matplotlib.axes._subplots.AxesSubplot at 0x2c5a33343c8>
```



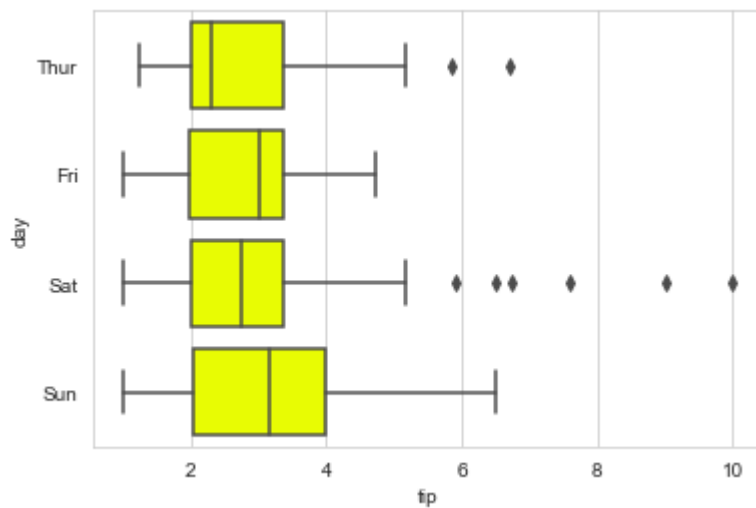
```
In [ ]: df = sns.load_dataset("tips")
# dodge is used to chakker laga kr nikal jana
# not give tip of thess tip
sns.boxplot(x='tip', y = "day", data=df, hue='smoker', palette= 'Set2', dodge=True, sa
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x2c5a48c4278>



```
In [ ]: df = sns.load_dataset("tips")
# u can change the color to add any code of hexa color piker on google
#
sns.boxplot(x='tip', y = "day", data=df, saturation=1, color='#e8fc03')
#
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x2c5a4b17518>



In [ ]:

## Start of Chilla Students Data with plotting

Name : Ali Nawaz

Education : MS in Software Engineering

Current Status : Artificial Intelligence Engienerg at NUST

From : KPK

In [ ]:

```
# Improting Libraries
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import StrMethodFormatter
```

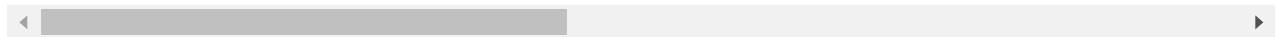
In [ ]:

```
# import dataset of the chilla
df = pd.read_csv('Chilla_data.csv')
df.head()
```

Gender	Location	Age	Qualification_completed	field_of_study	Purpose_for_chilla	What are you?	Blood group
--------	----------	-----	-------------------------	----------------	--------------------	---------------	-------------

	Gender	Location	Age	Qualification_completed	field_of_study	Purpose_for_chilla	What are you?	Blood group
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemployed	B+
1	Male	Pakistan	26-30	Bachelors	CS/IT	to boost my skill set	Student	B+
2	Male	Pakistan	31-35	Masters	Engineering	Switch my field of study	Employed	B+
3	Female	Pakistan	31-35	Masters	CS/IT	to boost my skill set	Employed	O+
4	Female	Pakistan	26-30	Masters	Engineering	to boost my skill set	Student	A

5 rows × 23 columns



In [ ]:

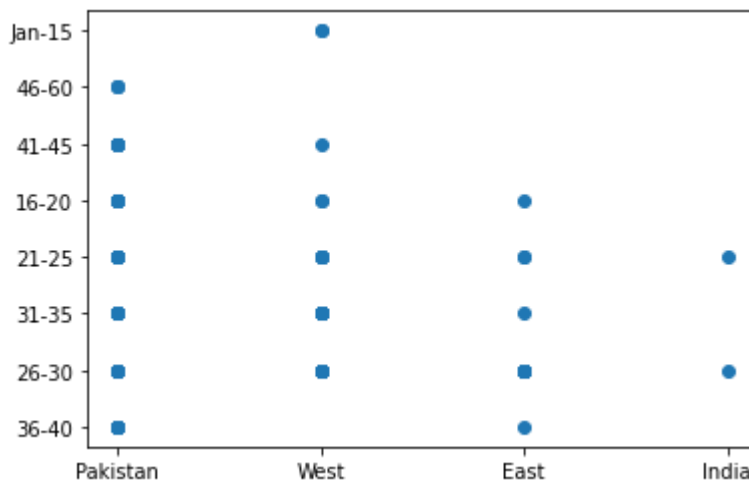
```
# I want the location and age wise desnistry of the dataset provided by the students
df_new = df[['Location', 'Age', 'What are you?']]
print("Length of the Data is :", len(df_new))
df_new.head()
```

Length of the Data is : 375

	Location	Age	What are you?
0	Pakistan	36-40	Unemployed
1	Pakistan	26-30	Student
2	Pakistan	31-35	Employed
3	Pakistan	31-35	Employed
4	Pakistan	26-30	Student

In [ ]:

```
x = df_new.Location
y = df_new.Age
plt.scatter(x, y)
# I can also do by single line but it was my start so that why I just do this
plt.show() # or plt.savefig("name.png")
```

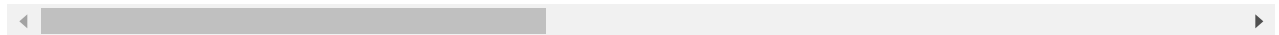


In [ ]:

```
df.head(2)
```

	Gender	Location	Age	Qualification	Subject	purpose	Employment	Blood_group	SIM_company
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemployed	B+	U-fone
1	Male	Pakistan	26-30	Bachelors	CS/IT	to boost my skill set	Student	B+	U-fone

2 rows × 23 columns



## Cleaning Data

In [ ]:

```
# rename_col_name
df.rename(columns={'Qualification_completed': 'Qualification', 'field_of_study': 'Subject',
'Purpose_for_chilla': 'purpose', 'What are you?': 'Employment', 'Blood group ': 'Blood', 'Where do you live?': 'living_place', 'Research/Working experience (Float/Int) years': 'experience_years', 'Your Weight in kg? (float)': 'Weight', 'Height in cm? Freelancer- (Float)': 'Height', 'How Light kitni der band hti hy? int': 'Loadshedding'}, inplace = True)
```

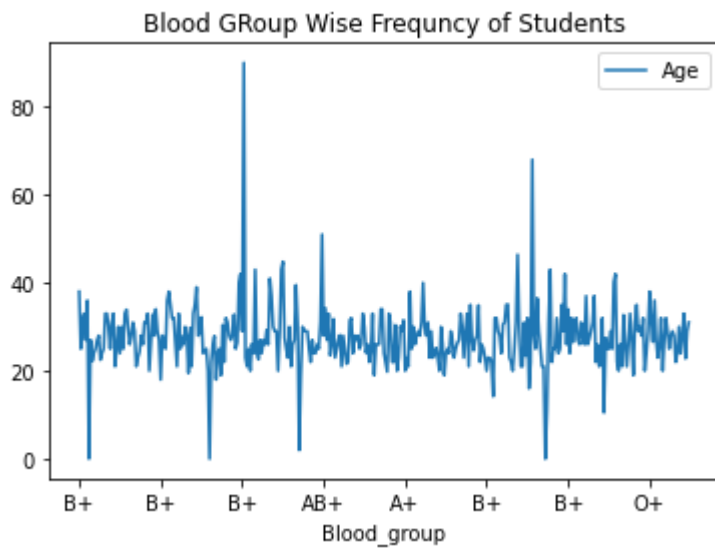
In [ ]:

```
df.to_csv('cleaned_chilla_data.csv')
```

## Different Plots to understand what data need for each plot and how it's works

In [ ]:

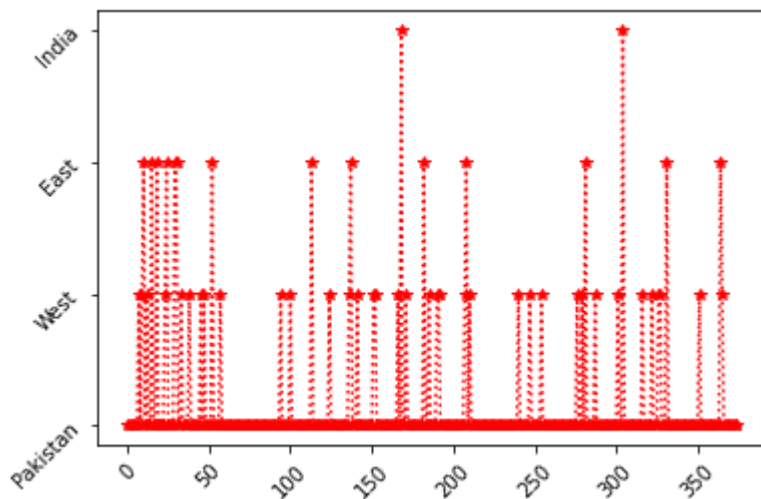
```
# plot for finding the relation between two variable
dff = df[['Blood_group', 'Age', 'Location']]
dff.set_index('Blood_group', 'Location').plot()
plt.title("Blood Group Wise Frequency of Students")
plt.show()
```



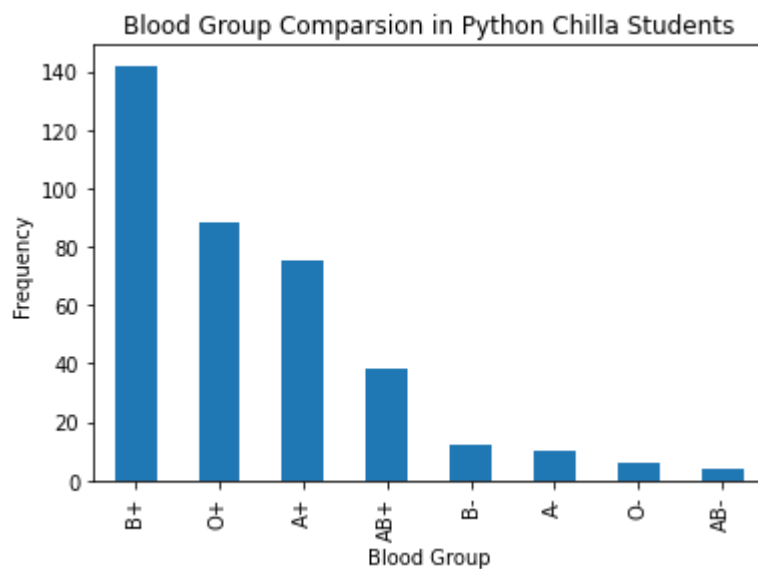
```
In [ ]: plt.plot(df.index, df['Location'], '*', ls='dotted', color='red')

plt.tick_params(rotation=45)

plt.show()
```

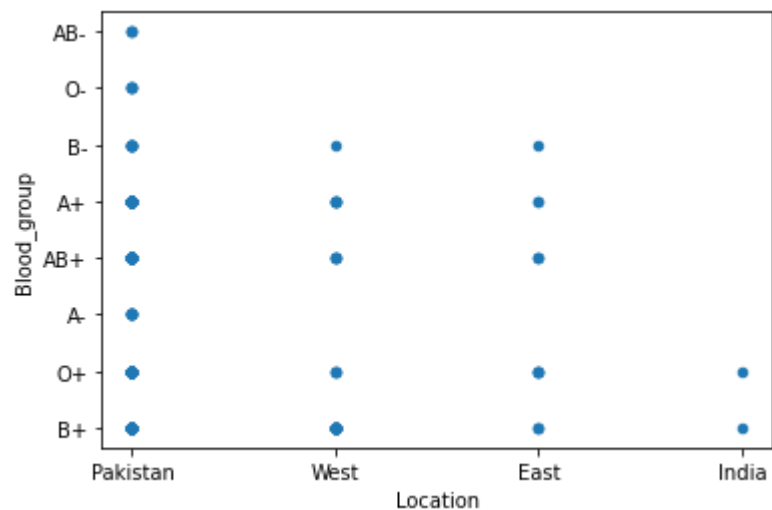


```
In [ ]: fig, ax = plt.subplots()
plt.title('Blood Group Comparsion in Python Chilla Students')
df['Blood_group'].value_counts().plot(ax=ax, kind='bar', xlabel='Blood Group', ylabel='
plt.show()
```



```
In [ ]: # scatter plot the dataframe
df.plot.scatter(x="Location", y="Blood_group")

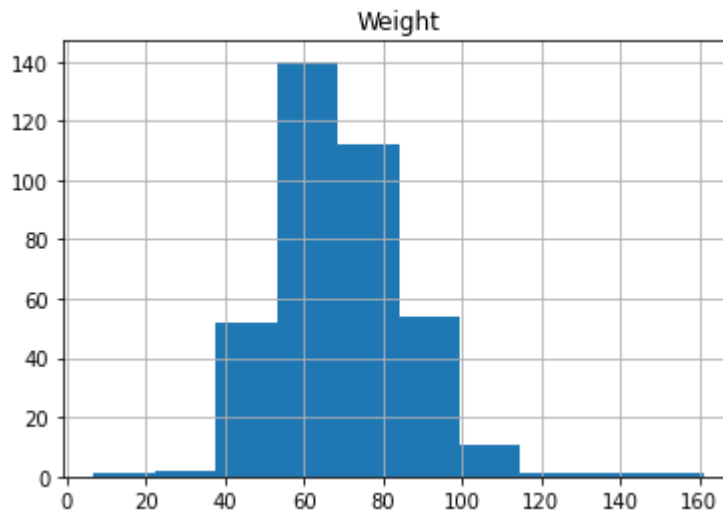
# displaying scatter plot
plt.show()
```



```
In [ ]: # Histogram for a dataframe for some numeric value
df.hist(column='Weight')

plt.show()
```

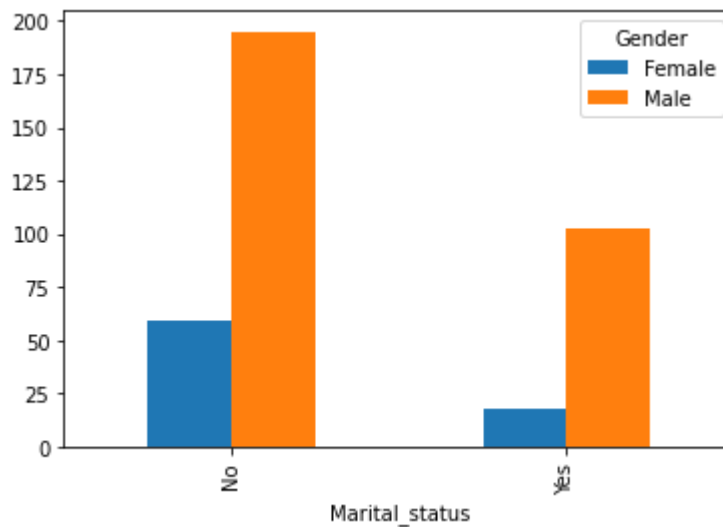




In [ ]:

```
# To Check how much peoples are married and unmarried
gender_column = 'Gender'
df.groupby(["Marital_status", gender_column]).size().unstack(level=1).plot(kind='bar')
```

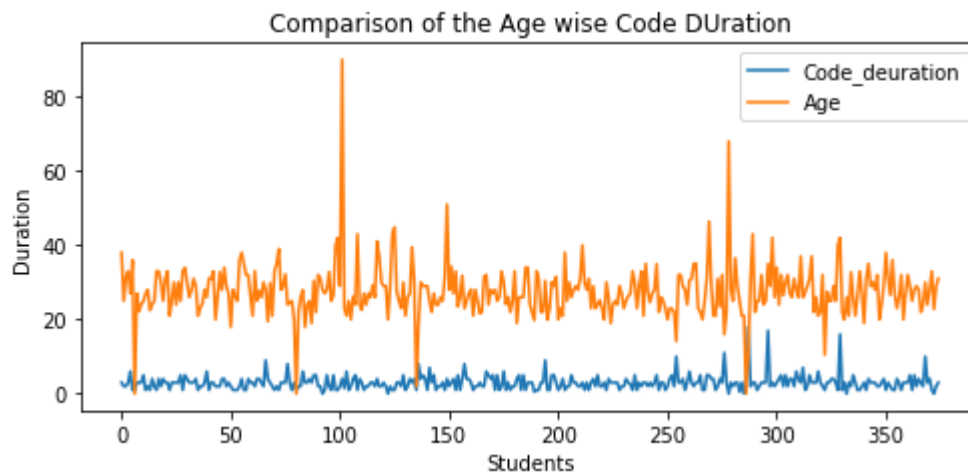
<AxesSubplot:xlabel='Marital\_status'>



In [ ]:

```
# Plot the lines
# Set the figure size
dff = df[['Code_deuration', 'Age']]
plt.rcParams["figure.figsize"] = [7.00, 3.50]
plt.rcParams["figure.autolayout"] = True
dff.plot()
plt.title("Comparison of the Age wise Code DURATION")
plt.ylabel('Duration')
plt.xlabel('Students')

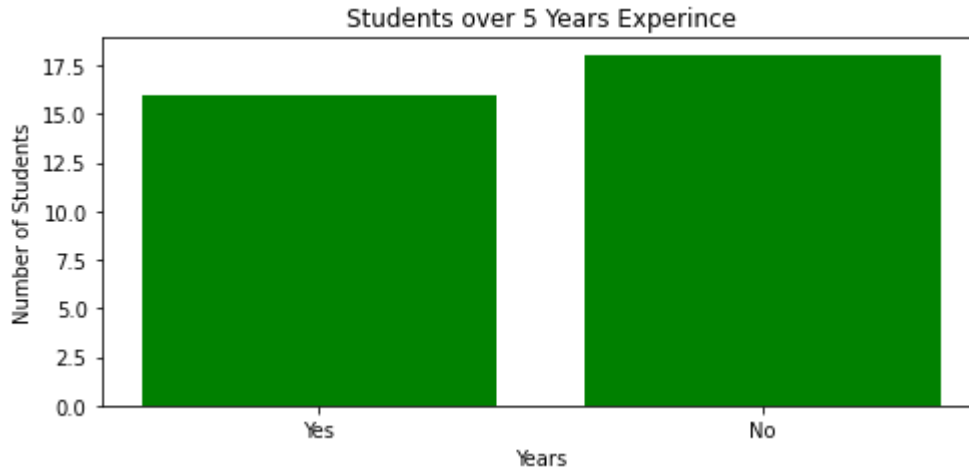
plt.show()
```



```
In [ ]: # Bar plot to show that how many are married and how many or not in 5 year of experinve
Y = list(df.Code_deuration)
X = list(df.Marital_status)

# Plot the data using bar() method
plt.bar(X, Y, color='g')
plt.title("Students over 5 Years Experinve")
plt.xlabel("Years")
plt.ylabel("Number of Students")

# Show the plot
plt.show()
```



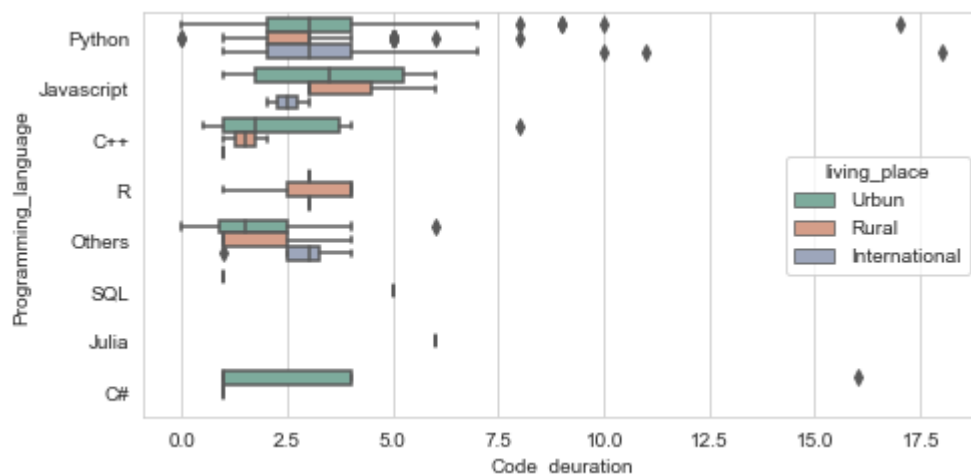
```
In [ ]: df.head(2)
```

	Gender	Location	Age	Qaulification	Subject	purpose	Employment	Blood_group	SIM_company
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemployed	B+	U-fone
1	Male	Pakistan	26-30	Bachelors	CS/IT	to boost my skill set	Student	B+	U-fone

2 rows × 23 columns

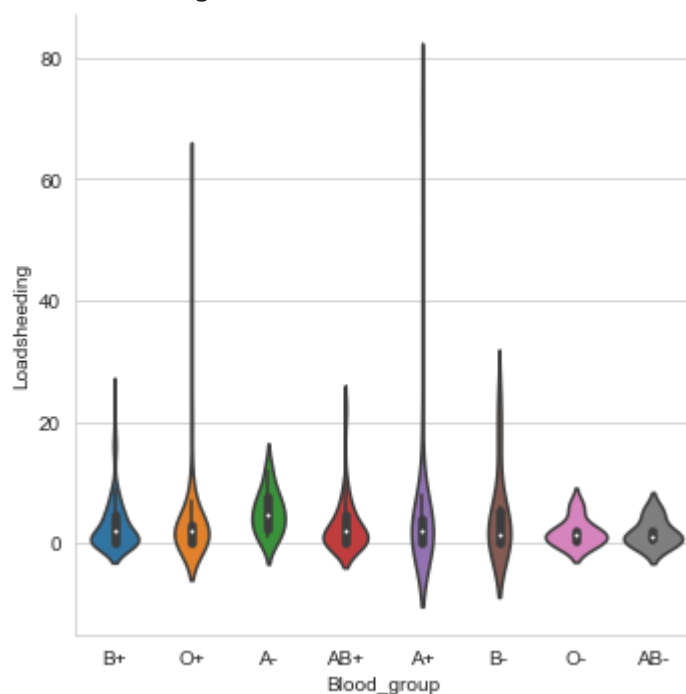
```
In [ ]: # dodge is used to chakker laga kr nikal jana
# not give tip of thess tip
sns.boxplot(x='Code_deuration', y = "Programming_language", data=df, hue='living_place')
```

<AxesSubplot:xlabel='Code\_deuration', ylabel='Programming\_language'>



```
In [ ]: sns.catplot(x="Blood_group", y="Loadshedding",
kind="violin", split=True, data=df)
```

<seaborn.axisgrid.FacetGrid at 0x22a835ee6a0>



```
In [ ]: titanic = sns.load_dataset("titanic")
titanic.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	5

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	9
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	9
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	9

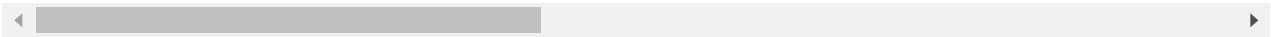


In [ ]:

```
df.head(2)
```

	Gender	Location	Age	Qaulification	Subject	purpose	Employment	Blood_group	SIM_company
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemployed	B+	U-fone
1	Male	Pakistan	26-30	Bachelors	CS/IT	to boost my skill set	Student	B+	U-fone

2 rows × 23 columns

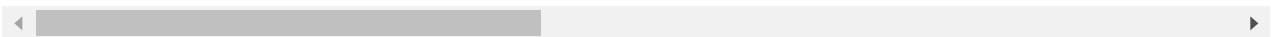


In [ ]:

```
df.head(2)
```

	Gender	Location	Age	Qaulification	Subject	purpose	Employment	Blood_group	SIM_company
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemployed	B+	U-fone
1	Male	Pakistan	26-30	Bachelors	CS/IT	to boost my skill set	Student	B+	U-fone

2 rows × 23 columns



In [ ]:

```
diamonds = sns.load_dataset("diamonds")
diamonds.head(2)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31

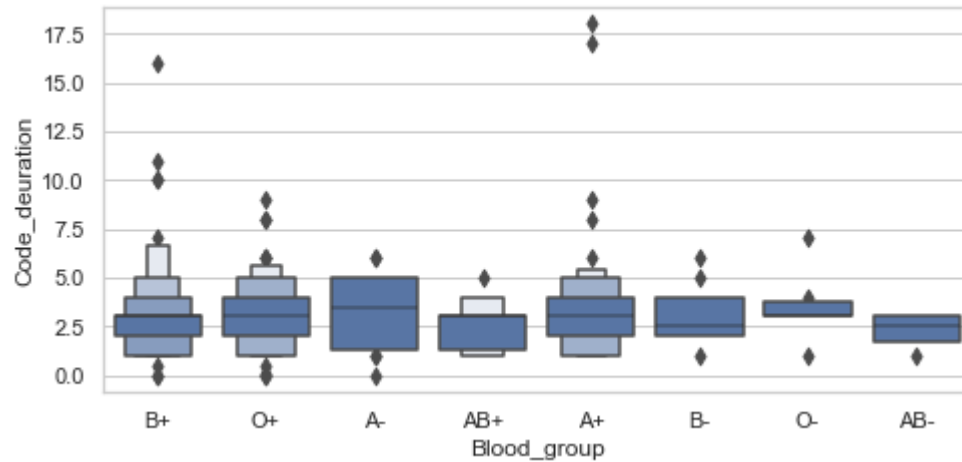
In [ ]:

```
sns.set_theme(style="whitegrid")

# diamonds = sns.load_dataset("diamonds")
clarity_ranking = ["I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"]
```

```
sns.boxenplot(x="Blood_group", y="Code_deuration",
              color="b",
              scale="linear", data=df)
```

<AxesSubplot:xlabel='Blood\_group', ylabel='Code\_deuration'>



In [ ]:

In [ ]:

In [ ]:

```
#package

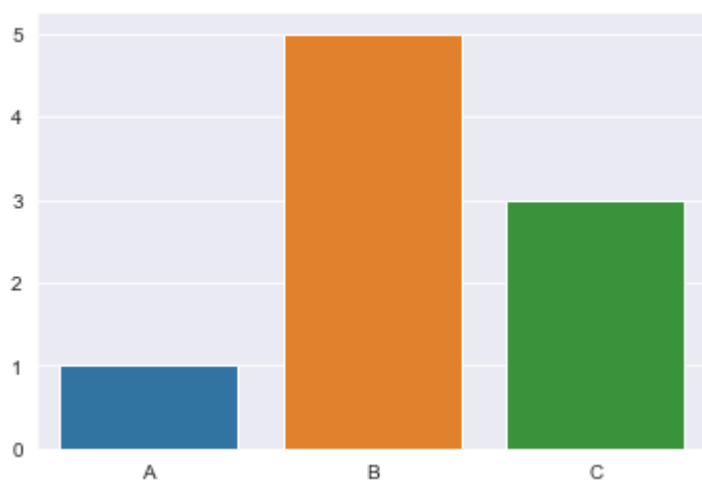
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')

x = ['A', 'B', 'C']
y = [1, 5, 3]

sns.barplot(x, y)
plt.show()
```

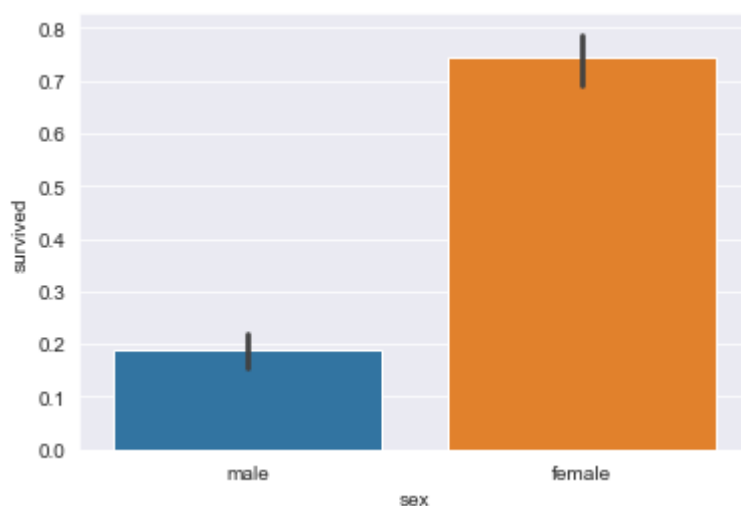
C:\Users\Ali\anaconda3\envs\python-chilla\lib\site-packages\seaborn\\_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



```
In [ ]: # Set Seaborn style
sns.set_style('darkgrid')
# Import Data
titanic_dataset = sns.load_dataset("titanic")

# Construct plot
sns.barplot(x = "sex", y = "survived", data = titanic_dataset)
plt.show()
```



```
In [ ]: print(titanic_dataset.head())
```

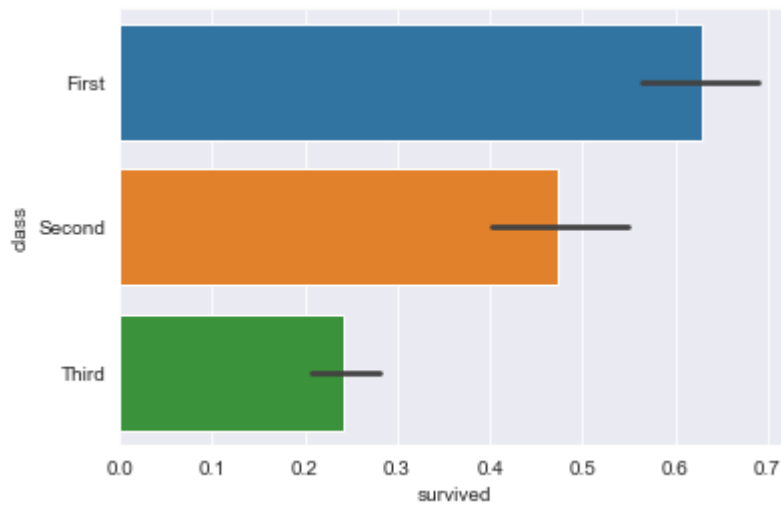
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
0	0	3	male	22.0	1	0	7.2500	S	Third	
1	1	1	female	38.0	1	0	71.2833	C	First	
2	1	3	female	26.0	0	0	7.9250	S	Third	
3	1	1	female	35.0	1	0	53.1000	S	First	
4	0	3	male	35.0	0	0	8.0500	S	Third	

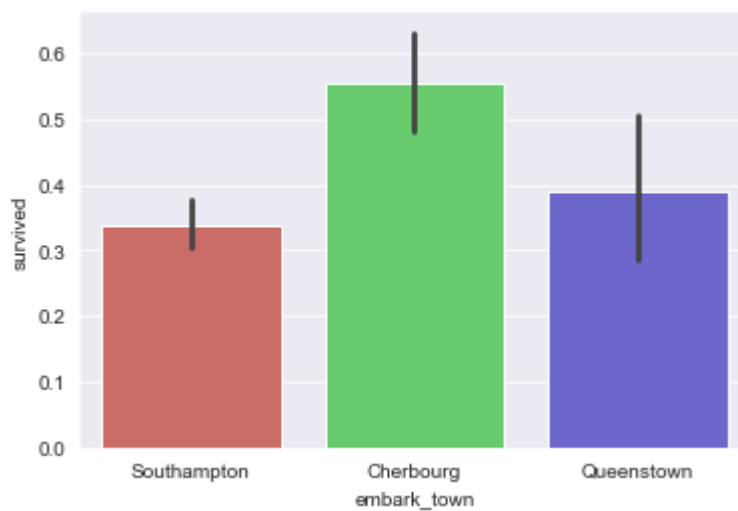
	who	adult_male	deck	embark_town	alive	alone
0	man	True	NaN	Southampton	no	False
1	woman	False	C	Cherbourg	yes	False
2	woman	False	NaN	Southampton	yes	True
3	woman	False	C	Southampton	yes	False
4	man	True	NaN	Southampton	no	True

```
In [ ]:
```

```
# horizontal_bar plot
sns.barplot(x = "survived", y = "class", data = titanic_dataset)
plt.show()
```

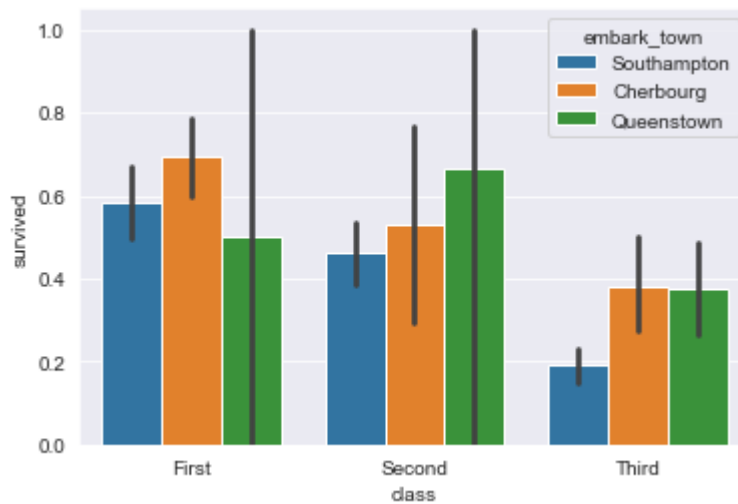


```
In [ ]: # color changing
sns.barplot(x = "embark_town", y = "survived", palette = 'hls', data = titanic_dataset)
plt.show()
```



```
In [ ]: # Plot Grouped Bar Plot in Seaborn

sns.barplot(x = "class", y = "survived", hue = "embark_town", data = titanic_dataset)
plt.show()
```

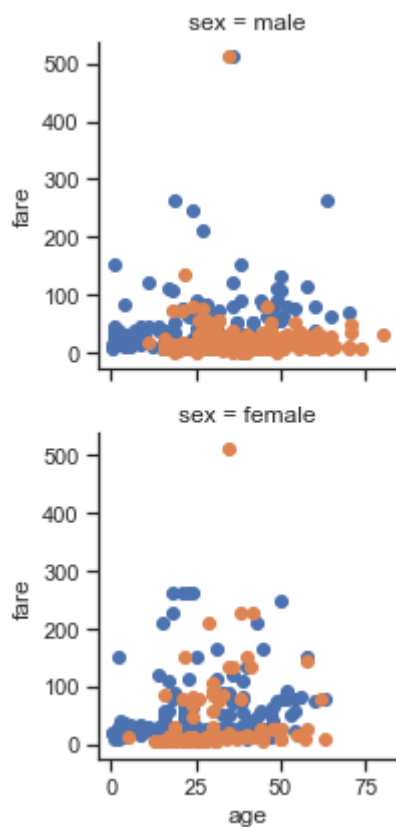


```
In [ ]: # create plot
sns.countplot(x = 'class', hue = 'who', data = titanic_dataset, palette = 'magma')
plt.title('Survivors')
plt.show()
```



```
In [ ]: # for continous data we use scatter
sns.set_theme(style='ticks', color_codes=True)
g = sns.FacetGrid(titanic_dataset, row='sex', hue='alone')
g = (g.map(plt.scatter, "age", "fare").add_legend)
plt.show()
```

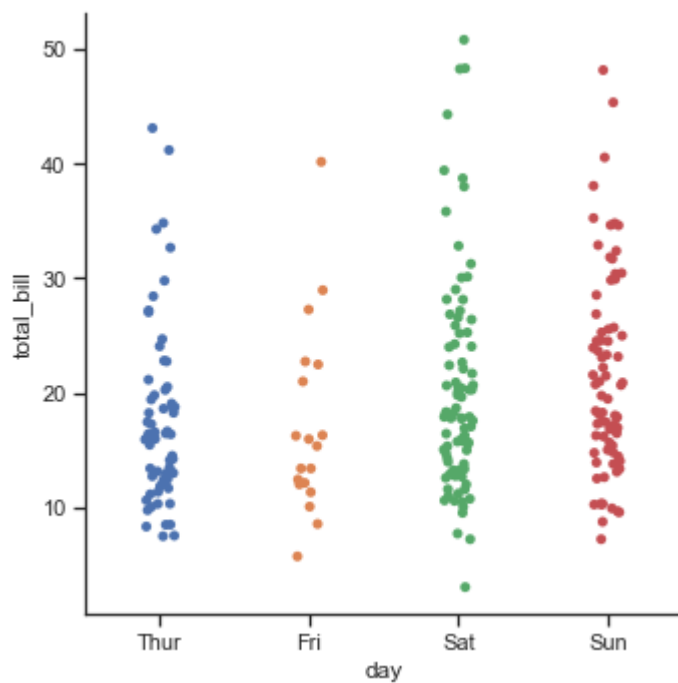




In [ ]:

```
sns.set_theme(style="ticks", color_codes=True)
tips = sns.load_dataset("tips")
sns.catplot(x="day", y="total_bill", data=tips)
```

<seaborn.axisgrid.FacetGrid at 0x1da92e5e278>



In [ ]:

```
sns.catplot(x="day", y="total_bill", jitter=False, data=tips)
```

<seaborn.axisgrid.FacetGrid at 0x1da92ed78d0>

