# Categorical Plots

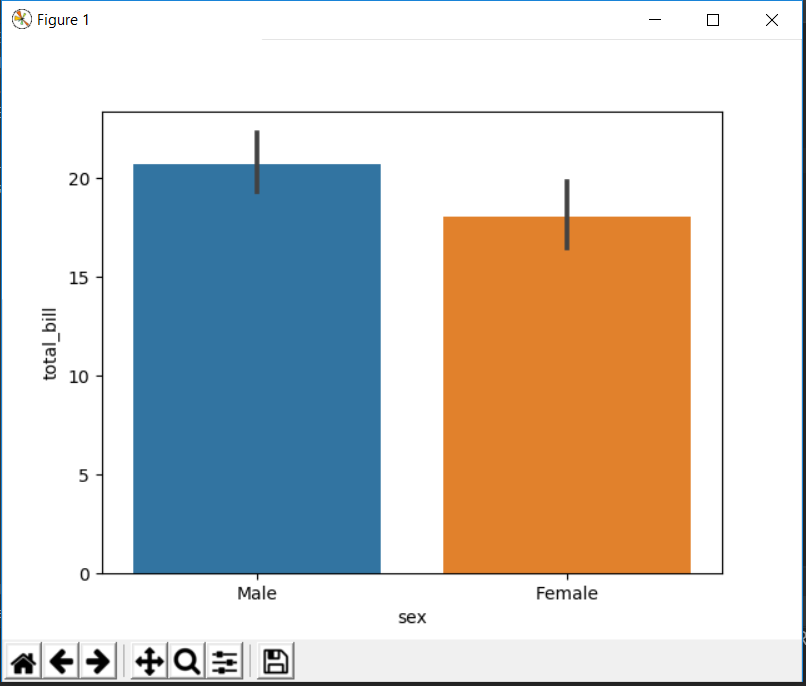
There are a few main plot types for this:

* factorplot
* boxplot
* violinplot
* stripplot
* swarmplot
* barplot
* countplot

## barplot

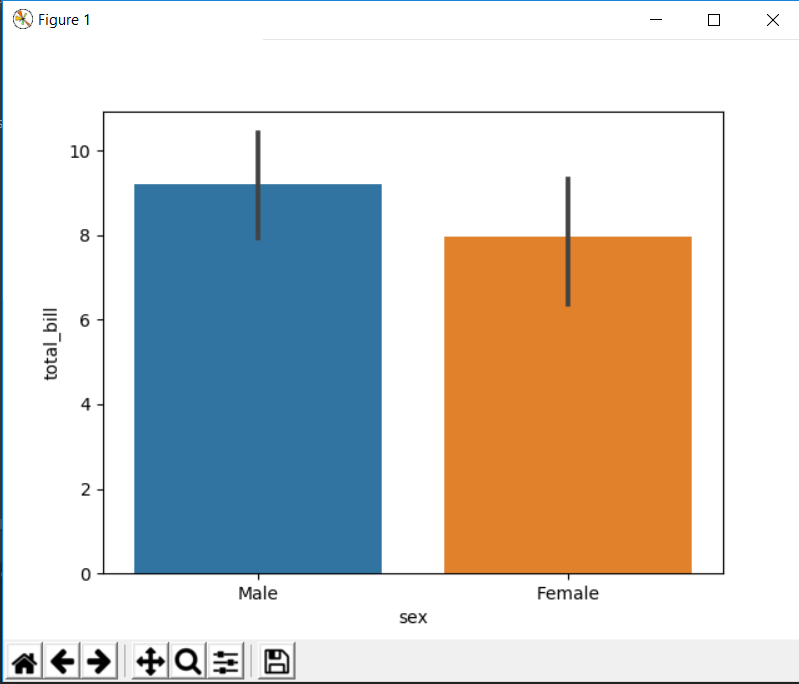
**barplot** is a general plot that allows you to aggregate the categorical data based off some function, by default the mean

import seaborn as sns  
import matplotlib.pyplot as plt  
tips = sns.load\_dataset("tips")  
sns.barplot(x="sex",y="total\_bill",data=tips)  
plt.show()



You can change the estimator object to your own function, that converts a vector to a scalar. You can also apply your own customised function

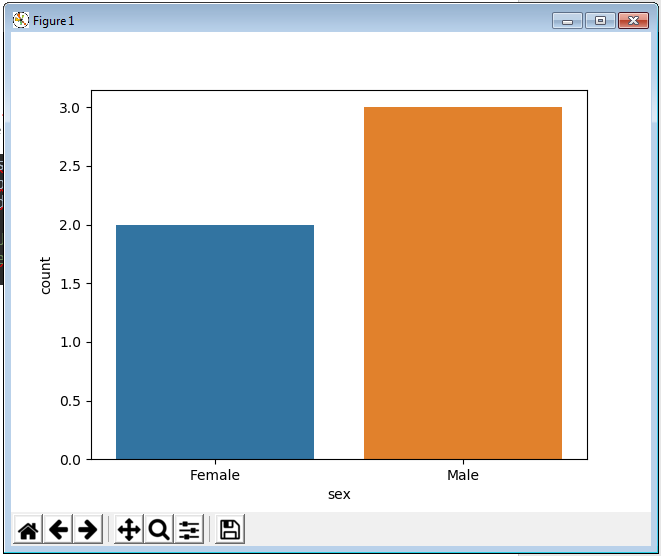
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
tips = sns.load\_dataset("tips")  
sns.barplot(x="sex",y="total\_bill",data=tips,estimator=np.std)  
plt.show()



## countplot

This is essentially the same as barplot except the estimator is explicitly counting the number of occurrences. Which is why we only pass the x value.

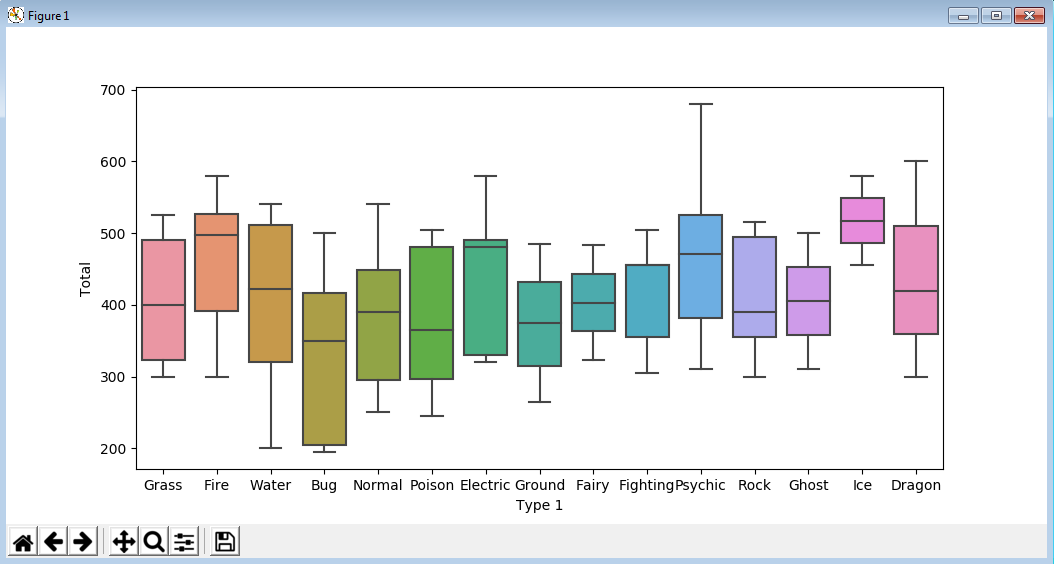
import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.countplot(x="sex",data=tips)  
plt.show()



## boxplot

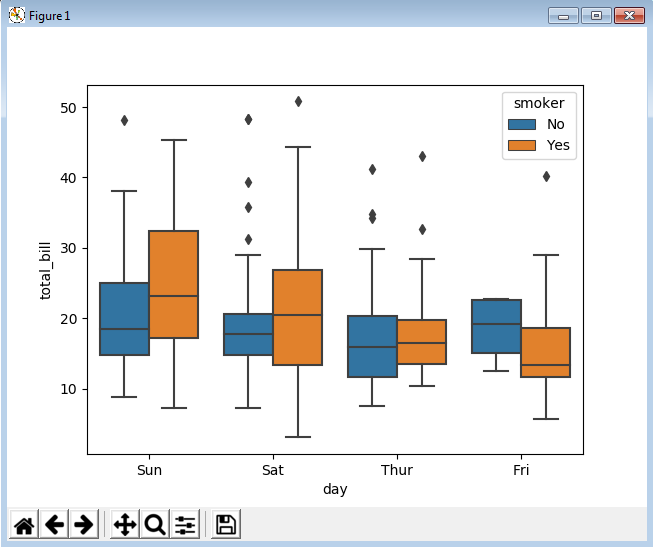
boxplots and violinplots are used to shown the distribution of categorical data. A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/Pokemon.xlsx")  
sns.boxplot(x="Type 1",y="Total",data=tips)  
plt.show()

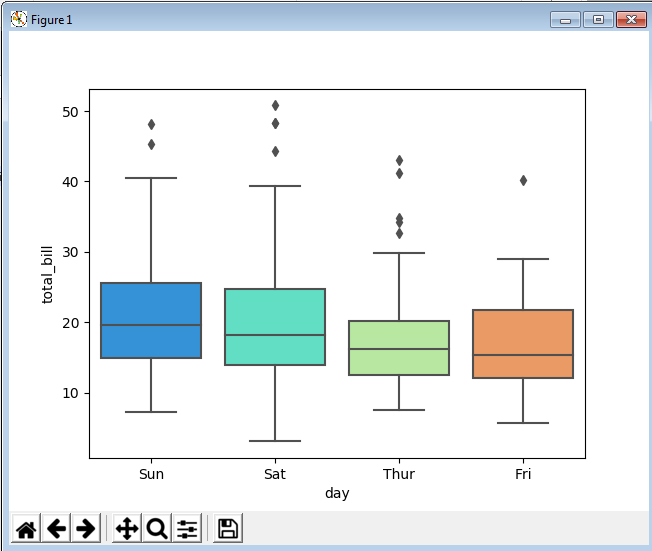


We can add hue argument to quickly analyse the data on some other argument

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.boxplot(x="day",y="total\_bill",data=tips,hue="smoker")  
plt.show()

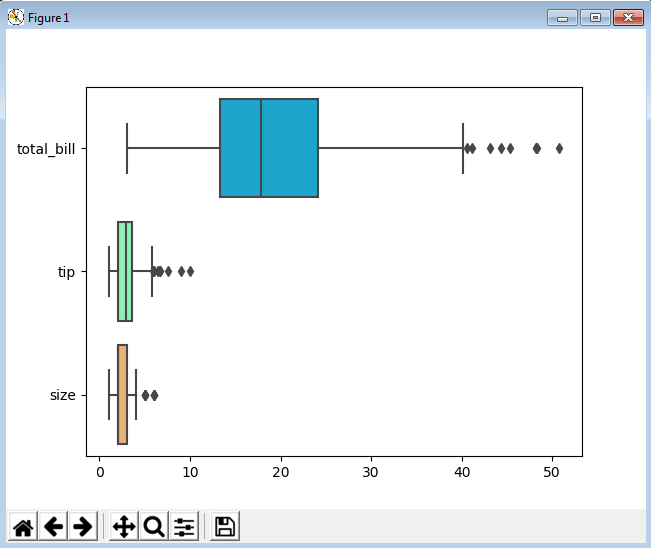


import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.boxplot(x="day",y="total\_bill",data=tips,palette="rainbow")  
plt.show()



We can also draw box plots for whole data wherever possible and even change the orientation

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.boxplot(data=tips,palette="rainbow",orient="h")  
plt.show()

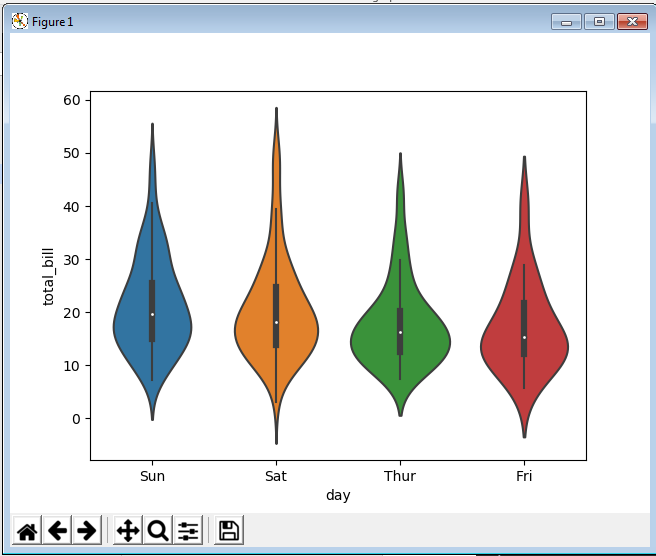


Here we can see, orientation is horizontal and seaborn has picked categories on its own.

## violinplot

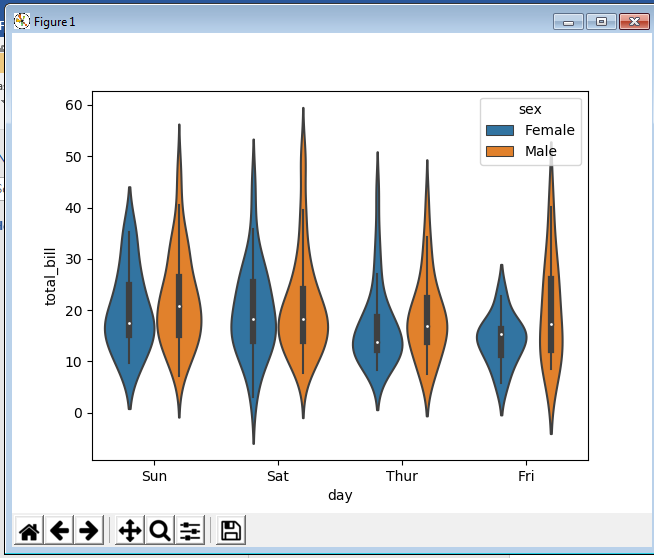
violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. Unlike a box plot, in which all of the plot components correspond to actual datapoints, the violin plot features a kernel density estimation of the underlying distribution. So it gives more information that the box plot but the cost comes at the interpretation since its harder to read as compared to a box plot.

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.violinplot(x="day",y="total\_bill",data=tips)  
plt.show()



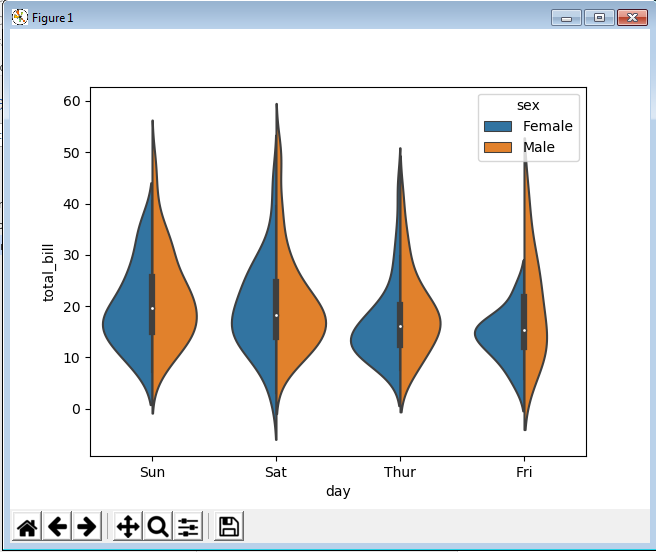
It also has option for hue just as box plot by specifying hue=”sex”

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.violinplot(x="day",y="total\_bill",data=tips,hue="sex")  
plt.show()



We can also merge these violin plots by getting one half from one category and other half from other category.

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.violinplot(x="day",y="total\_bill",data=tips,hue="sex",split=True)  
plt.show()

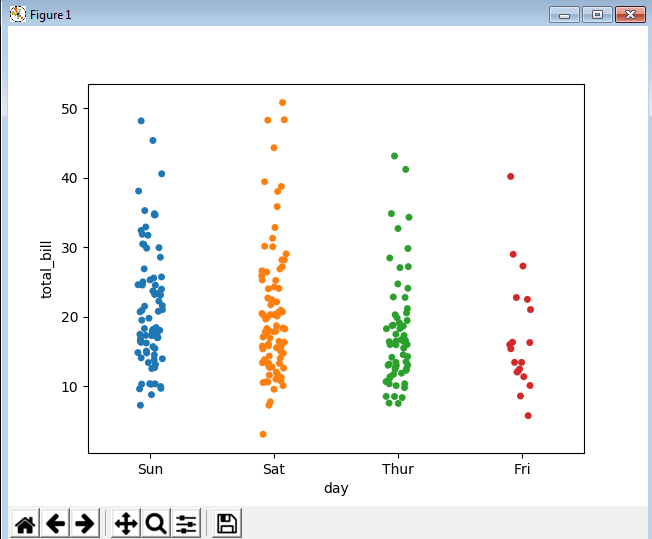


Now, compare the above two plots.

## stripplot

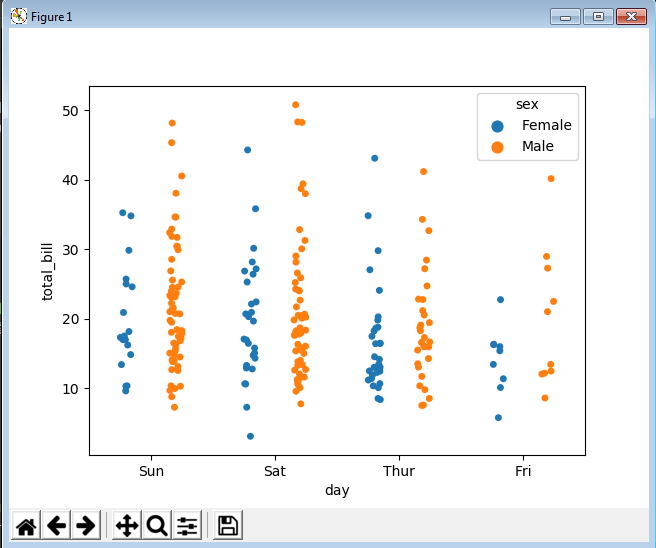
The stripplot will draw a scatterplot where one variable is categorical. A strip plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution.

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.stripplot(x="day",y="total\_bill",data=tips,)  
plt.show()



We can also add some jitter since it sometimes becomes hard to find out how many data points are stacked over each other. We can even add hue and split arguments

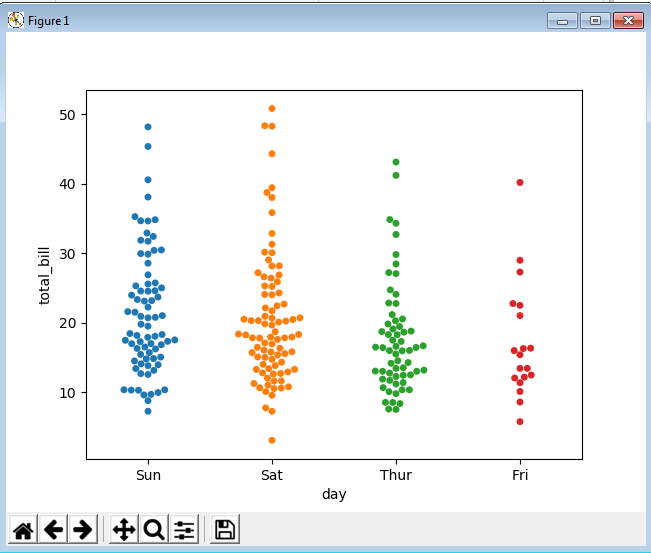
import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.stripplot(x="day",y="total\_bill",data=tips,jitter=True,hue="sex",split=True)  
plt.show()



## swarmplot

The swarmplot is similar to stripplot(), but the points are adjusted (only along the categorical axis) so that they don’t overlap. This gives a better representation of the distribution of values, although it does not scale as well to large numbers of observations (both in terms of the ability to show all the points and in terms of the computation needed to arrange them).

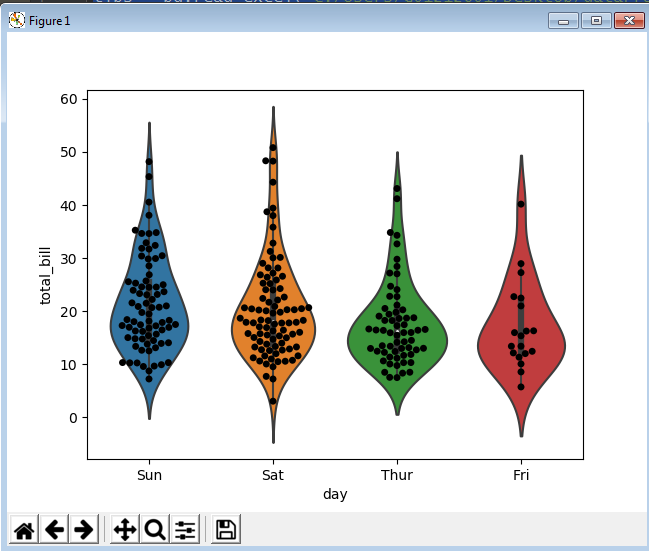
import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.swarmplot(x="day",y="total\_bill",data=tips)  
plt.show()



One drawback of using swarmplot is they behave don’t well with large data sets since they don’t scale well to large numbers and points can get too wide in order not to stack over each other.

Even if we need to use swarmplot, we can use them in conjunction with violin plot which can give large amount of information about the data set.

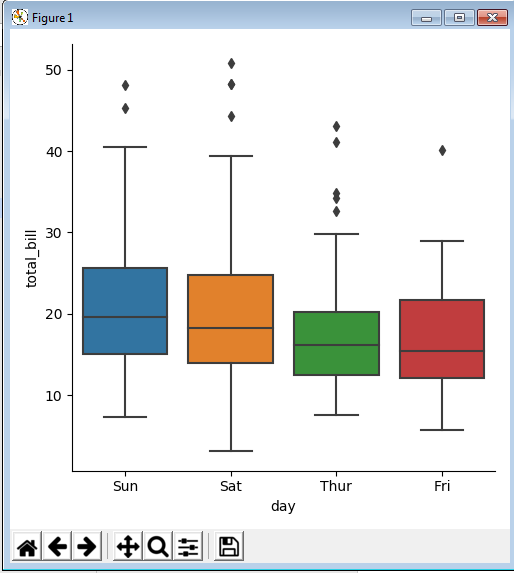
import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.violinplot(x="day",y="total\_bill",data=tips)  
sns.swarmplot(x="day",y="total\_bill",data=tips,color="black")  
plt.show()



## factorplot

factorplot is the most general form of a categorical plot. It can take in a \*\*kind\*\* parameter to adjust the plot type

import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
tips = pd.read\_excel("C:/Users/G01212601/Desktop/dataFrame.xlsx")  
sns.factorplot(x="day",y="total\_bill",data=tips,kind="box")  
plt.show()



Other types can also be added-

sns.factorplot(x="day",y="total\_bill",data=tips,kind="swarm")

sns.factorplot(x="day",y="total\_bill",data=tips,kind="violin")