Open in Colab

Class 6

Violin plot

A violin plot plays a similar activity that is pursued through whisker or box plot do.

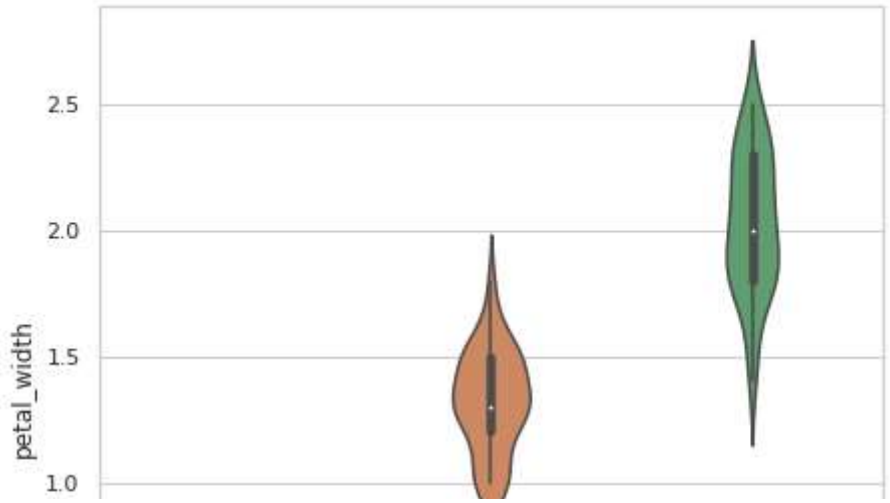
It shows several quantitative data across one or more categorical variables.

It can be an effective and attractive way to show multiple data at several units.

A “wide-form” Data Frame helps to maintain each numeric column which can be plotted on the graph. It is possible to use NumPy or Python objects, but pandas objects are preferable because the associated names will be used to annotate the axes.

Draw a violin plot with species and petal width

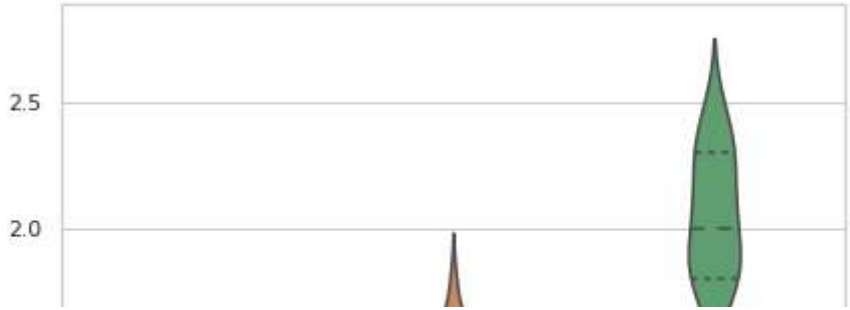
```
import seaborn as sns
import matplotlib.pyplot as plt
iris=sns.load_dataset("iris")
plt.figure(figsize=(7,7))
sns.violinplot(x="species",y="petal_width",  data=iris)
plt.show()
```



Violin plot

Draw a violin plot with species and petal width with quartile markings

```
1 iris=sns.load_dataset("iris")
2 plt.figure(figsize=(7,7))
3 sns.violinplot(x="species",y="petal_width",  data=iris, inner="quartile")
4 plt.show()
```



Violin plot

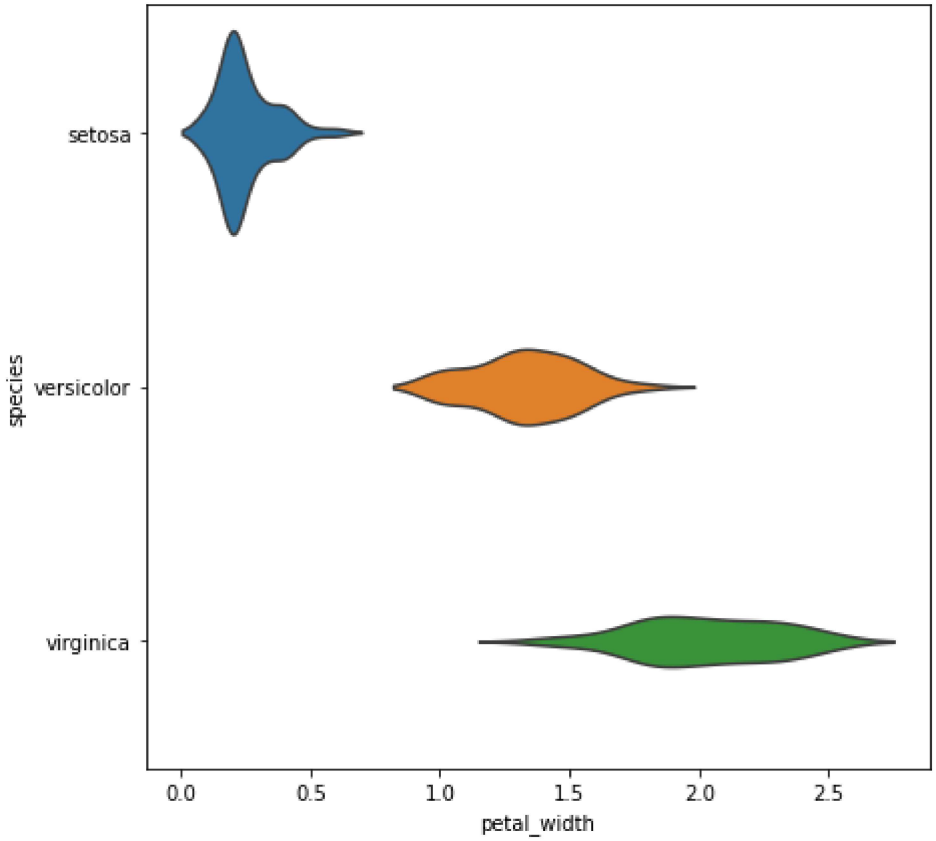
Draw a violin plot with species and petal width with no quartile markings

```
1 iris=sns.load_dataset("iris")
2 plt.figure(figsize=(7,7))
3 sns.violinplot(x="species",y="petal_width",  data=iris, inner=None)
4 plt.show()
```

Violin plot

Draw a violin plot with species in y axis and petal width in x-axis with quartile markings

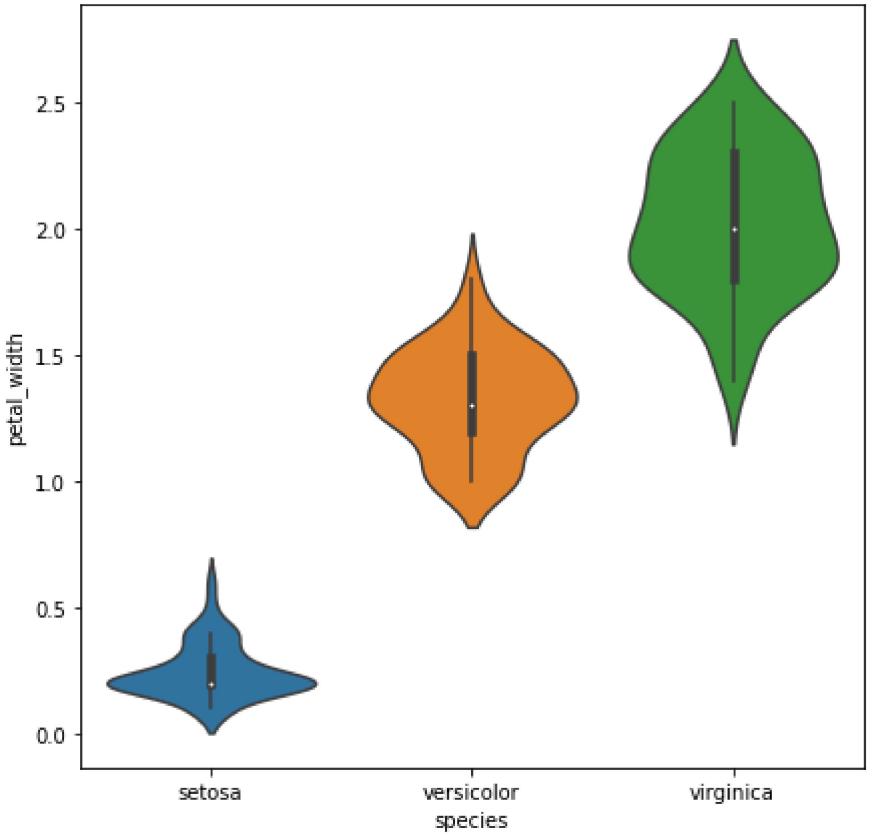
```
1 iris=sns.load_dataset("iris")
2 plt.figure(figsize=(7,7))
3 sns.violinplot(y="species",x="petal_width",  data=iris, inner=None)
4 plt.show()
```



Violin plot

Draw a violin plot with species and petal width with cut and scale parameters

```
1 iris=sns.load_dataset("iris")
2 plt.figure(figsize=(7,7))
3 sns.violinplot(x="species",y="petal_width",  data=iris, cut=2,scale="count")
4 plt.show()
```

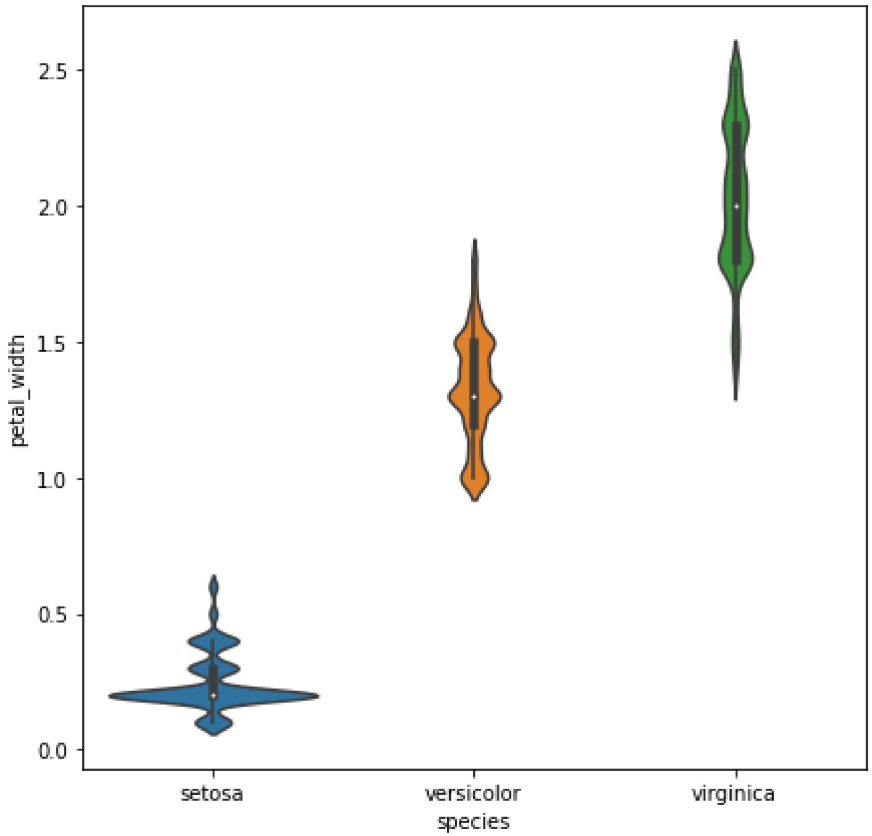


Violin plot

Draw a violin plot with species and petal width with cut and bandwidth parameters

```
1 iris=sns.load_dataset("iris")
```

```
2 plt.figure(figsize=(7,7))
3 sns.violinplot(x="species",y="petal_width",  data=iris, cut=2, bw=0.2)
4 plt.show()
```

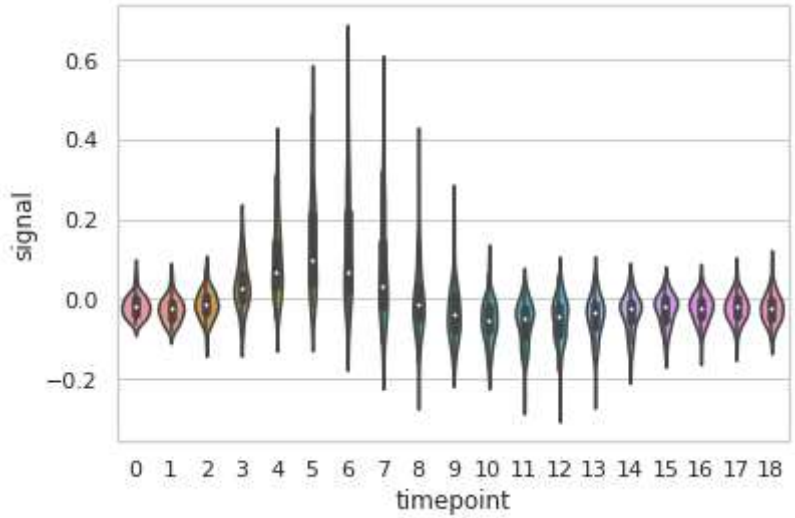


Violin plot

Draw a violin plot using fmri

```
1 sns.set(style = 'whitegrid')
2 fmri = sns.load_dataset("fmri")
3
4 sns.violinplot(x ="timepoint",
5       y ="signal",
6       data = fmri)
7
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff008774ad0>
```

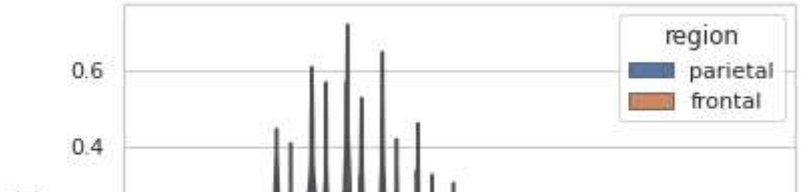



```
1 fmri.columns
```

```
Index(['subject', 'timepoint', 'event', 'region', 'signal'], dtype='object')
```

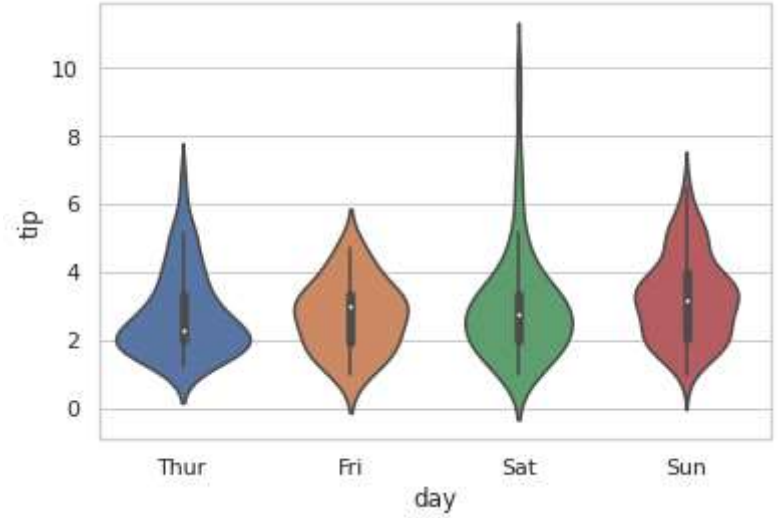
```
1 sns.violinplot(x ="timepoint",
2       y ="signal",
3       hue ="region",
4       style ="event",
5       data = fmri)
6
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff009095a90>
```




```
1 tip = sns.load_dataset('tips')
2 sns.violinplot(x ='day', y ='tip', data = tip)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff008e08310>
```

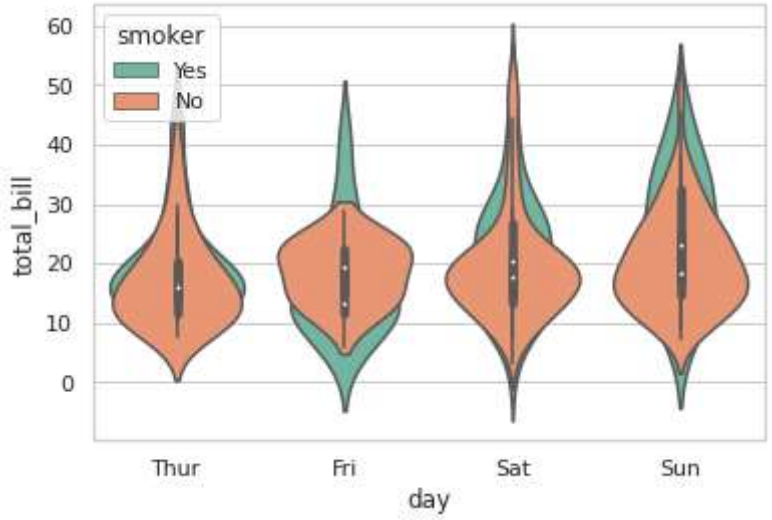



```
1 sns.violinplot(x ='day', y ='tip', data = tip, linewidth = 4)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff008d1d4d0>
```


```
1 tip = sns.load_dataset('tips')
2 sns.violinplot(x="day", y="total_bill", hue="smoker",
3                  data=tip, palette="Set2", dodge=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff008448f10>
```

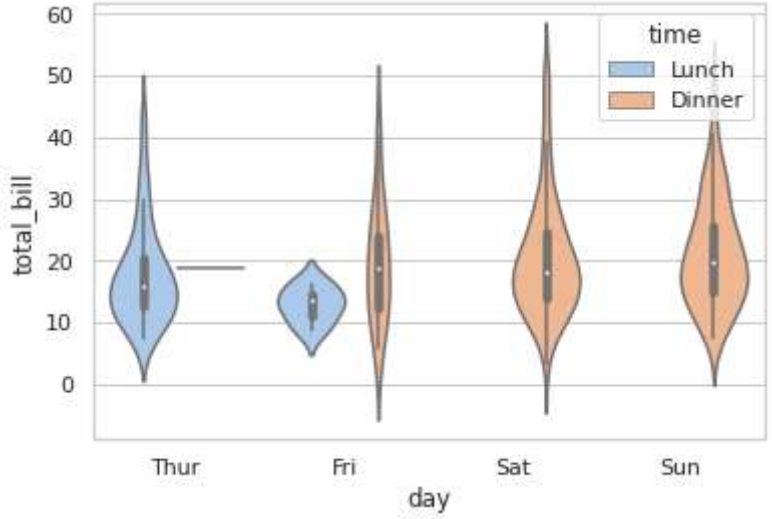



```
1 sns.violinplot(x="time", y="tip", data=tip,
2                  order=["Dinner", "Lunch"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff00831ff50>
```


```
1
2 sns.violinplot(x='day', y='total_bill',
3                    data=tip, hue='time', palette='pastel')
```

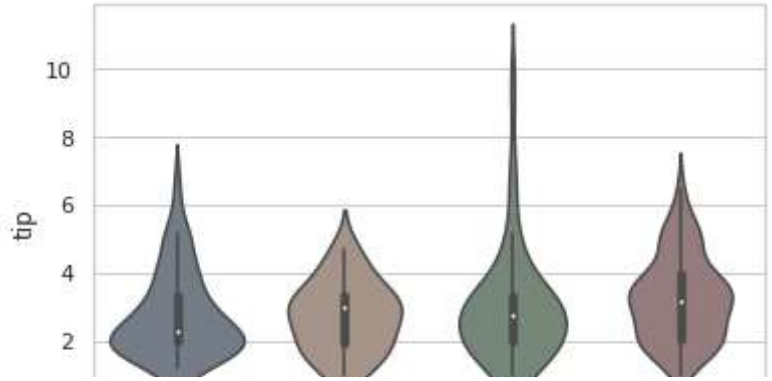
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff0082a4c10>
```



Double-click (or enter) to edit

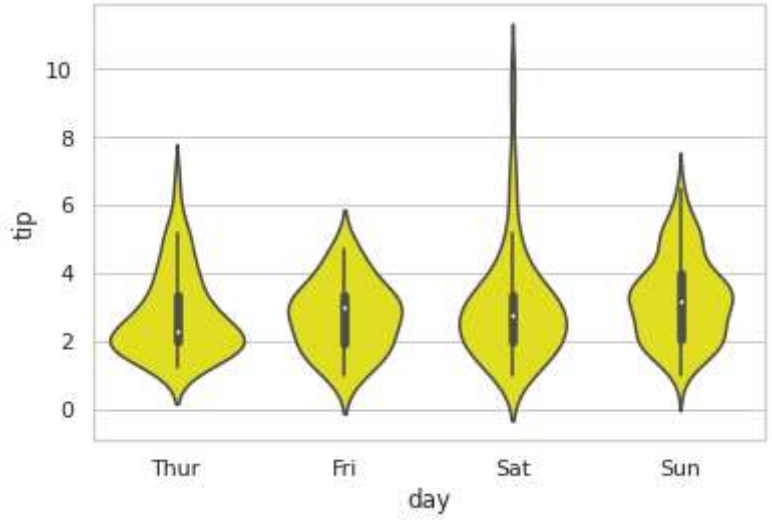
```
1 sns.violinplot(x ='day', y ='tip',
2                    data = tip, saturation =0.2)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff008246dd0>
```




```
1 sns.violinplot(x ='day', y ='tip', data = tip, color = "Yellow")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff008a08f10>
```




```
1 sns.violinplot(x="day", y="total_bill", hue="sex",
2                    data=tip, palette="Set2", split=True,
3                    scale="count")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff009145110>
```