

Categorical Data Plots

- `swarmplot()` and `stripplot()` that show each observation at each level of the categorical variable.
- `boxplot()` and `violinplot()` that show an abstract representation of each distribution of observations.
- `barplot()`, `pointplot()` and `countplot()` that apply a statistical estimation to show a measure of Central Tendency and Confidence Interval
- `fatorplot()` is the most general form of a categorical plot. It can take in a `kind` parameter to adjust the plot type.

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import warnings
%matplotlib inline
# warnings.filterwarnings("ignore")
sns.set(rc={"figure.figsize": (5, 3)})
sns.set_style('whitegrid')
```

swarm plot

This plot is a scattered representation of non-overlapping points of the three species of Iris flower with difference in values of their sepal length, sepal width, petal length and petal width.

```
In [ ]: # Loading dataset
iris=sns.load_dataset('iris')
# Melt dataset to 'Long-form' or 'tidy' representation:
iris = pd.melt(iris, "species", var_name="measurement")
```

```
In [ ]: # Drawing a categorical scatterplot to show each observation
sns.swarmplot(x="measurement", y="value", hue="species", data=iris,palette='de
```

changing order

```
In [ ]: sns.swarmplot(x="measurement", y="value", hue="species", data=iris,order=['sep
```

strip plot

- It displays the distribution of a continuous variable for each category by placing individual data points along a axis.

```
In [ ]: tips = sns.load_dataset("tips")
sns.stripplot(x=tips["total_bill"], size=4, color="green")
```

```
In [ ]: sns.stripplot(x="day", y="total_bill", data=tips,palette='rocket')
```

```
In [ ]: sns.stripplot(x="day", y="total_bill", data=tips, jitter=True, hue='sex', pale
```

split the dots by category

```
In [ ]: sns.stripplot(x="day", y="total_bill", data=tips, jitter=True, hue='sex', pale
```

```
In [ ]: iris = sns.load_dataset("iris")
iris = pd.melt(iris, "species", var_name="measurement")
sns.stripplot(x="measurement", y="value", hue="species",data=iris, palette="ic
```

box plot()

depicts group of numerical data through their quartiles in Descriptive statistics

```
In [ ]: # Plotting basic Box Plot:
sns.boxplot(x="day", y="total_bill", data=tips)
```

```
In [ ]: sns.boxplot(x='day', y='total_bill', hue='smoker', data=tips, palette='viridis
```

box plot and stripplot together

```
In [ ]: sns.boxplot(x="day", y="total_bill", data=tips,palette='Set3')
sns.stripplot(x="day", y="total_bill", data=tips, color='b')
```

violin plot

it combines boxplot and kde

```
In [ ]: # Plotting basic Violin Plot horizontally:
sns.violinplot(x = tips["total_bill"], palette="coolwarm")
```

- thick bar in the centre is the interquartile range
- the thin line represents the confidence interval
- the white dot is median
- the spread forms the violin shape is kde which shows distribution- wider means higher probability

```
In [ ]: # Loading built-in Tips dataset:
tips = sns.load_dataset("tips")
# Draw a vertical violinplot grouped by a categorical variable:
sns.violinplot(x='day', y='total_bill', data=tips)
```

```
In [ ]: # Draw split violins to compare the across the hue variable:
sns.violinplot(x='day', y='total_bill', hue='sex', data = tips, palette='muted',
plt.legend())
```

scale on basis of count

```
In [ ]: # Scale the violin width by the number of observations in each bin:
sns.violinplot(x = 'day', y='total_bill', hue='sex', data = tips, palette='muted',
plt.legend())
```

drawing quartiles

```
In [ ]: sns.violinplot(x='day', y='total_bill', hue='sex', data=tips, inner='quartile',
plt.legend())
```

reduce bandwidth to reduce smoothing

```
In [ ]: sns.violinplot(x="day", y="total_bill",data=tips, bw=.2)
```

Bar Plot

```
In [ ]: # Loading built-in Tips dataset:
titanic = sns.load_dataset("titanic")
sns.barplot(x="class", y="fare", data=titanic)
# the lines represent confidence interval
```

adding standard deviation

```
In [ ]: sns.barplot(x="embark_town", y="age", data=titanic, hue="sex", errorbar="sd",
plt.legend())
```

count plot

A Count Plot can be thought of as a Histogram across a Categorical variable, instead of a Quantitative variable.

```
In [ ]: sns.countplot(x='class', data=titanic)
```

point plot

A point plot represents an estimate of central tendency for a numeric variable by the position of the dot and provides some indication of the uncertainty around that estimate using error bars.

```
In [ ]: sns.pointplot(x="sex", y="survived", hue="class", data=titanic, palette="rocke
```

```
In [ ]:
```