Data visualization

## step-1 import labirary

```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
```

## Step-2 load data set

```
1   titanic=sns.load_dataset("titanic")
2   titanic.head()
3
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | embark_town | alive | alone |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | male | 22.0 | 1 | 0 | 7.2500 | S | Third | man | True | NaN | Southampton | no | False |
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C | Cherbourg | yes | False |
| **2** | 1 | 3 | female | 26.0 | 0 | 0 | 7.9250 | S | Third | woman | False | NaN | Southampton | yes | True |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C | Southampton | yes | False |
| **4** | 0 | 3 | male | 35.0 | 0 | 0 | 8.0500 | S | Third | man | True | NaN | Southampton | no | True |

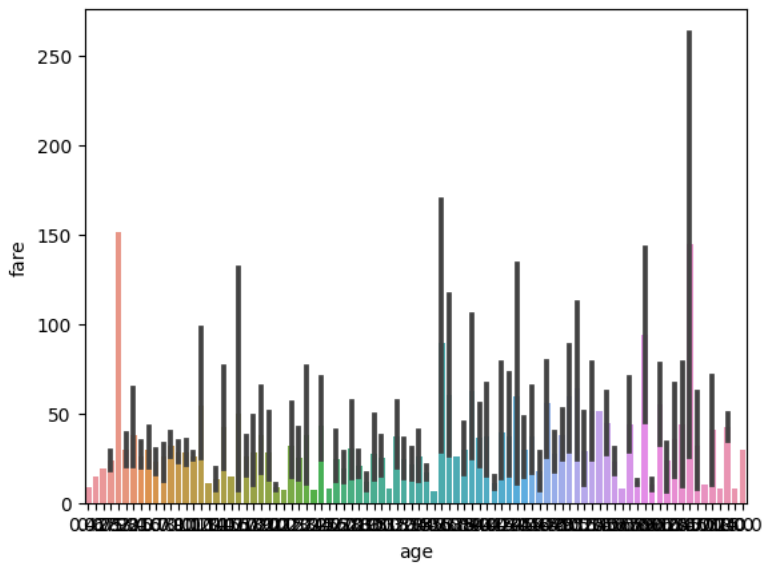
## Step 3 plot a graph

```
1 sns.lineplot(x="fare",y="age",data=titanic)
2 plt.xlim(3)
3 plt.ylim(4)
4 plt.title("Check")
5 plt.show()
```

## Bar Plot

```
1 df = sns.load_dataset("titanic")
2 sns.barplot(data=df, x="age", y="fare")
```

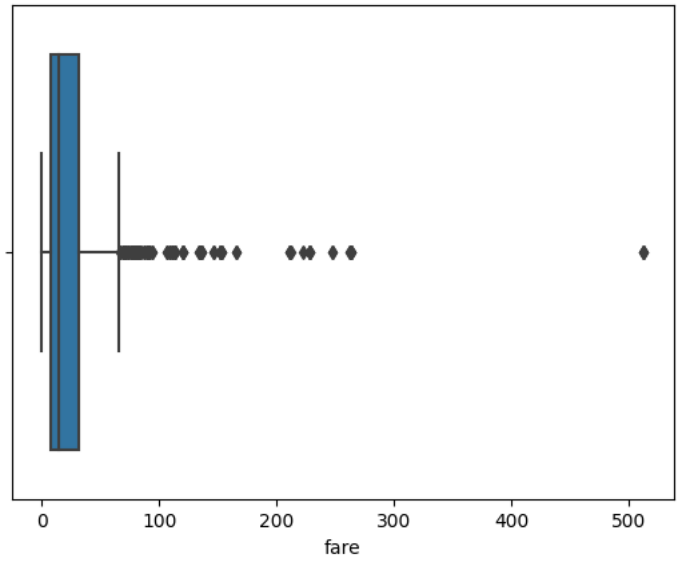
```
<Axes: xlabel='age', ylabel='fare'>
```




## Boxplot

```
df = sns.load_dataset("titanic")
sns.boxplot(x=df["fare"])
```

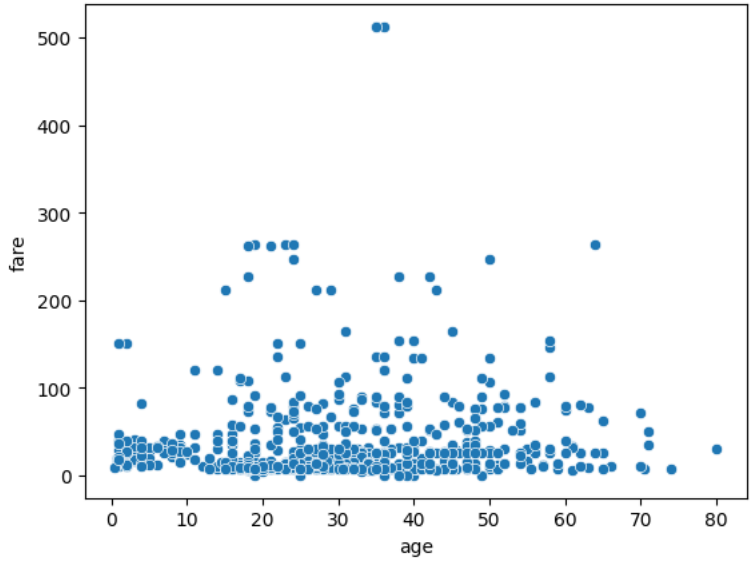
```
<Axes: xlabel='fare'>
```




## Scattor plot

```
sns.scatterplot(data=titanic, x="age", y="fare")
```

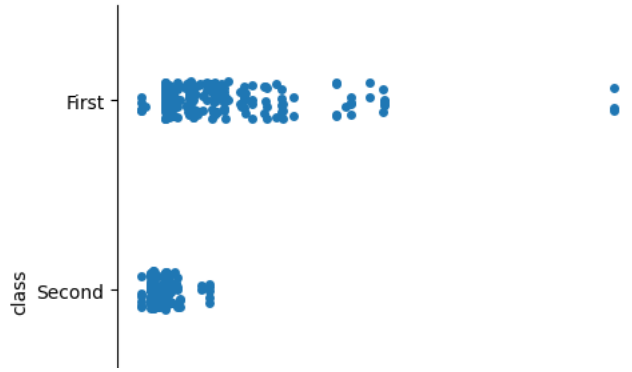
```
<Axes: xlabel='age', ylabel='fare'>
```




## Cat plot

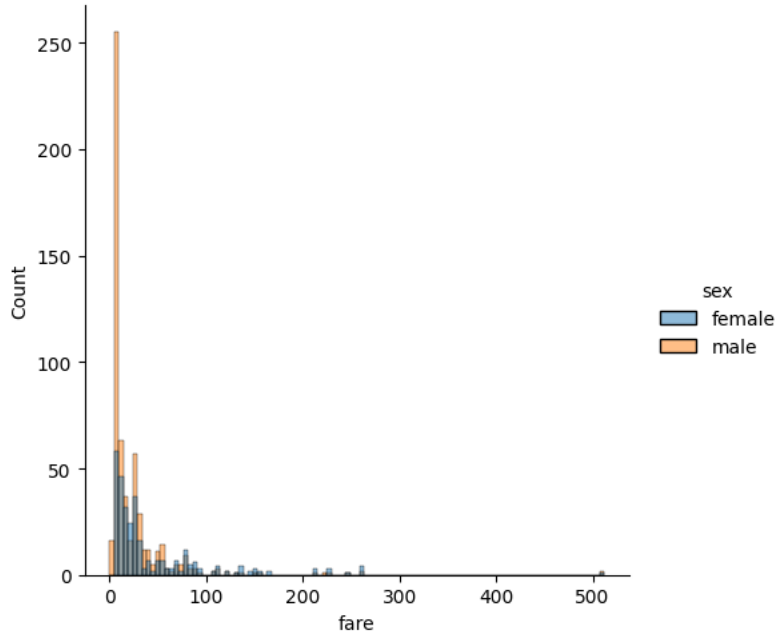
```
df = sns.load_dataset("titanic")
sns.catplot(data=df, x="fare", y="class")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc5bf9a1330>
```




```
1  sns.displot(data = titanic , x = "fare" , hue = "sex", hue_order = ['female', 'male'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc5bf8ffb50>
```

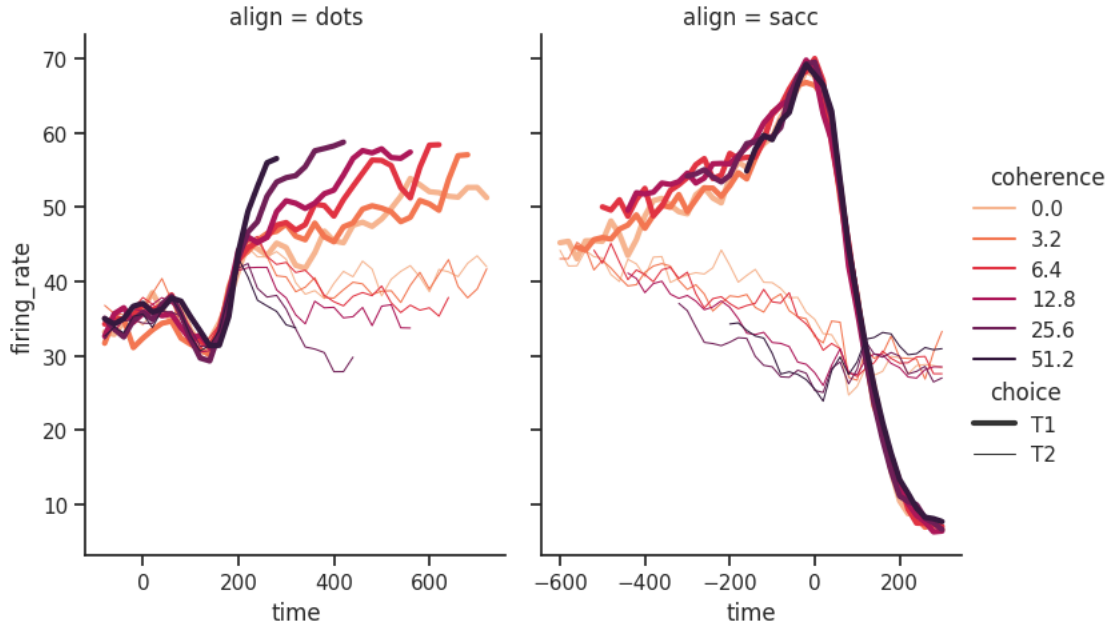



```
1 sns.heatmap(data = titanic.corr())
```

```
<ipython-input-68-f4da9c94fae6>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a futur
  sns.heatmap(data = titanic.corr())
```

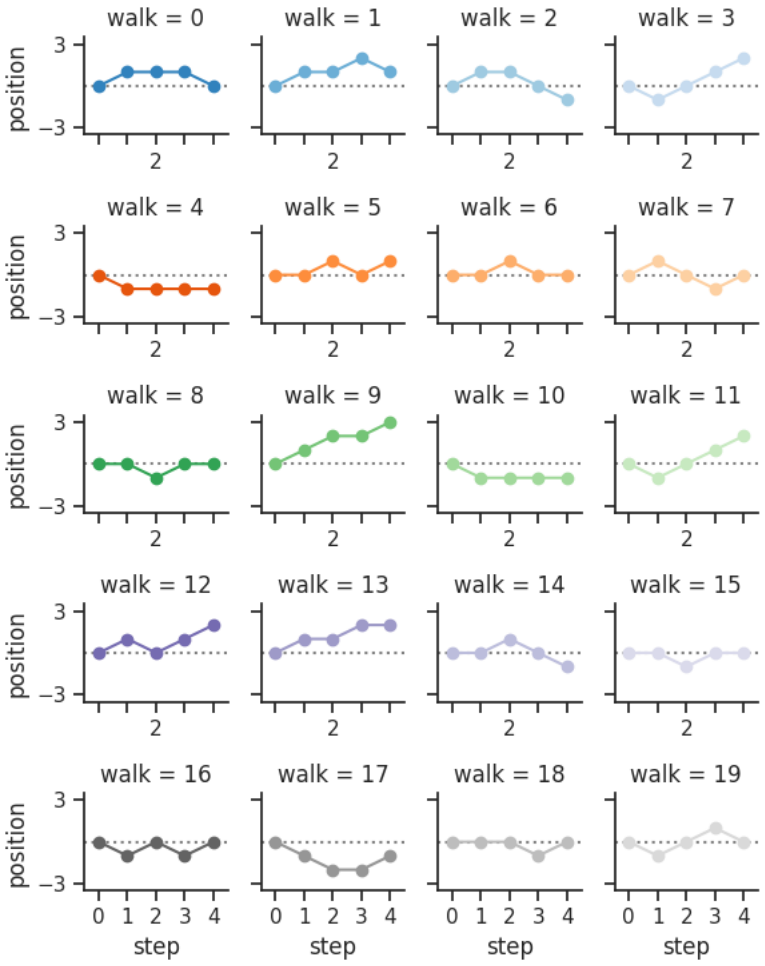
```
import seaborn as sns
sns.set_theme(style="ticks")

dots = sns.load_dataset("dots")

# Define the palette as a list to specify exact values
palette = sns.color_palette("rocket_r")

# Plot the lines on two facets
sns.relplot(
    data=dots,
    x="time", y="firing_rate",
    hue="coherence", size="choice", col="align",
    kind="line", size_order=["T1", "T2"], palette=palette,
    height=5, aspect=.75, facet_kws=dict(sharex=False),
)

```

```
<seaborn.axisgrid.FacetGrid at 0x7fc5bf871b40>
```



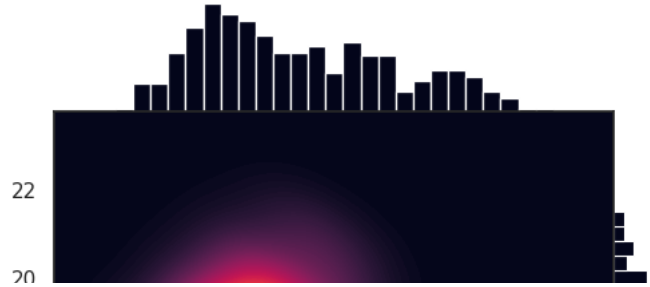

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_theme(style="ticks")

# Create a dataset with many short random walks
rs = np.random.RandomState(4)
pos = rs.randint(-1, 2, (20, 5)).cumsum(axis=1)
pos -= pos[:, 0, np.newaxis]
step = np.tile(range(5), 20)
walk = np.repeat(range(20), 5)
df = pd.DataFrame(np.c_[pos.flat, step, walk],
                  columns=["position", "step", "walk"])

# Initialize a grid of plots with an Axes for each walk
grid = sns.FacetGrid(df, col="walk", hue="walk", palette="tab20c",
                     col_wrap=4, height=1.5)

# Draw a horizontal line to show the starting point
grid.refline(y=0, linestyle=":")

# Draw a line plot to show the trajectory of each random walk
grid.map(plt.plot, "step", "position", marker="o")

# Adjust the tick positions and labels
grid.set(xticks=np.arange(5), yticks=[-3, 3],
         xlim=(-.5, 4.5), ylim=(-3.5, 3.5))
```

```
30
31  # Adjust the arrangement of the plots
32  grid.fig.tight_layout(w_pad=1)
33
```

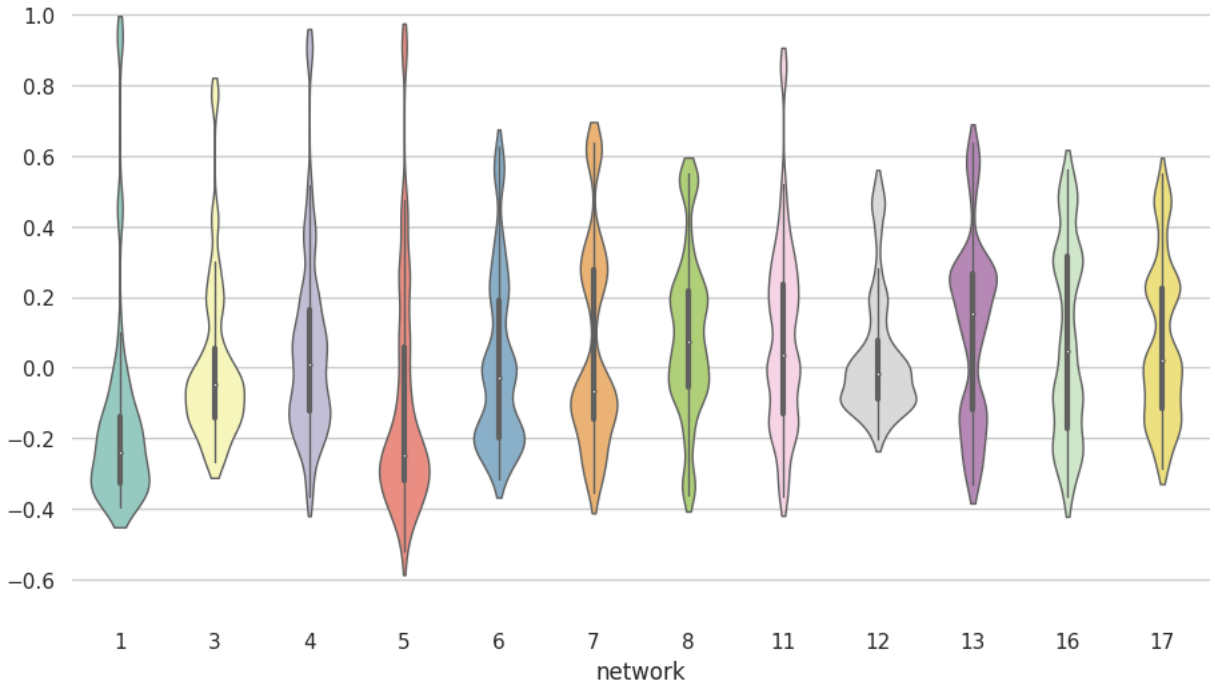



```
 1 import seaborn as sns
 2 sns.set_theme(style="white")
 3
 4 df = sns.load_dataset("penguins")
 5
 6 g = sns.JointGrid(data=df, x="body_mass_g", y="bill_depth_mm", space=0)
 7 g.plot_joint(sns.kdeplot,
 8               fill=True, clip=((2200, 6800), (10, 25)),
 9               thresh=0, levels=100, cmap="rocket")
10 g.plot_marginals(sns.histplot, color="#03051A", alpha=1, bins=25)
11
```

```
<seaborn.axisgrid.JointGrid at 0x7fc5bf4d7f40>
```




```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="whitegrid")

# Load the example dataset of brain network correlations
df = sns.load_dataset("brain_networks", header=[0, 1, 2], index_col=0)

# Pull out a specific subset of networks
used_networks = [1, 3, 4, 5, 6, 7, 8, 11, 12, 13, 16, 17]
used_columns = (df.columns.get_level_values("network")
                          .astype(int)
                          .isin(used_networks))
df = df.loc[:, used_columns]

# Compute the correlation matrix and average over networks
corr_df = df.corr().groupby(level="network").mean()
corr_df.index = corr_df.index.astype(int)
corr_df = corr_df.sort_index().T

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 6))

# Draw a violinplot with a narrower bandwidth than the default
sns.violinplot(data=corr_df, palette="Set3", bw=.2, cut=1, linewidth=1)

# Finalize the figure
ax.set(ylim=(-.7, 1.05))
sns.despine(left=True, bottom=True)

```



✓ 4s completed at 7:54 PM