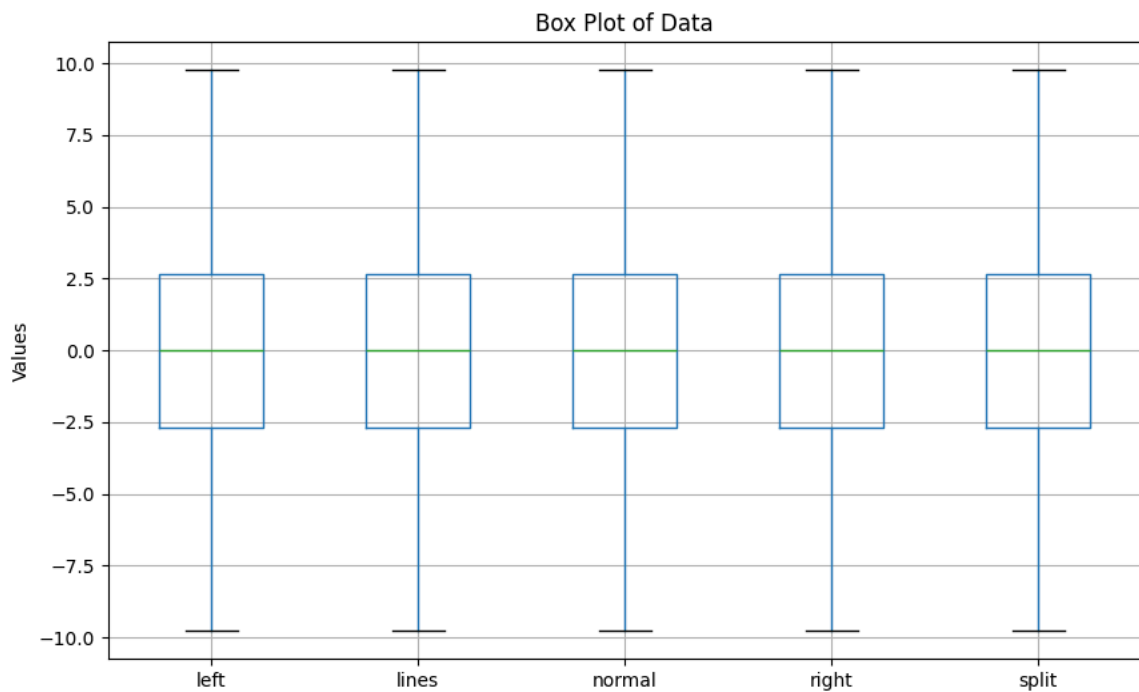


```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
df = pd.read_csv('BoxPlots.tsv', sep='\t', index_col=0)

# Basic Box Plot
plt.figure(figsize=(10, 6))
df.boxplot()
plt.title("Box Plot of Data")
plt.ylabel("Values")
plt.show()
```



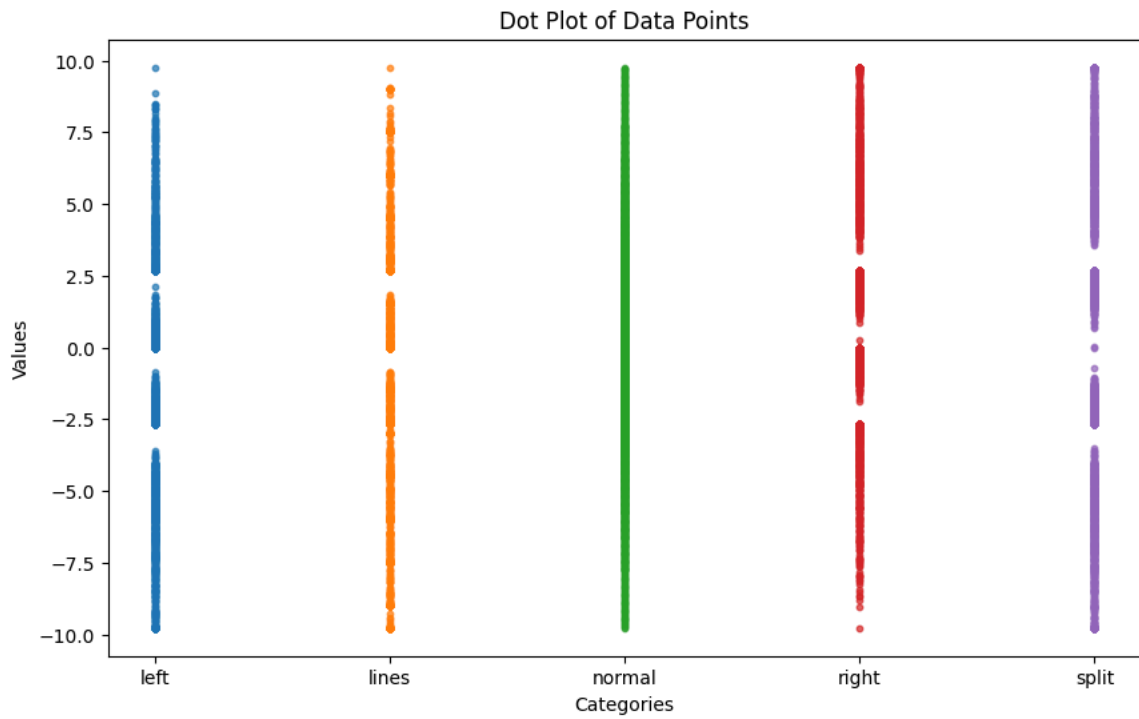
```
import numpy as np

plt.figure(figsize=(10, 6))

for i, column in enumerate(df.columns):
    x = np.full(df[column].shape, i + 1) # Assign x-coordinate based on category index
    plt.scatter(x, df[column], alpha=0.7, s=10) # Small, slightly transparent dots

plt.xticks(ticks=range(1, len(df.columns) + 1), labels=df.columns)
plt.title("Dot Plot of Data Points")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```

[4]

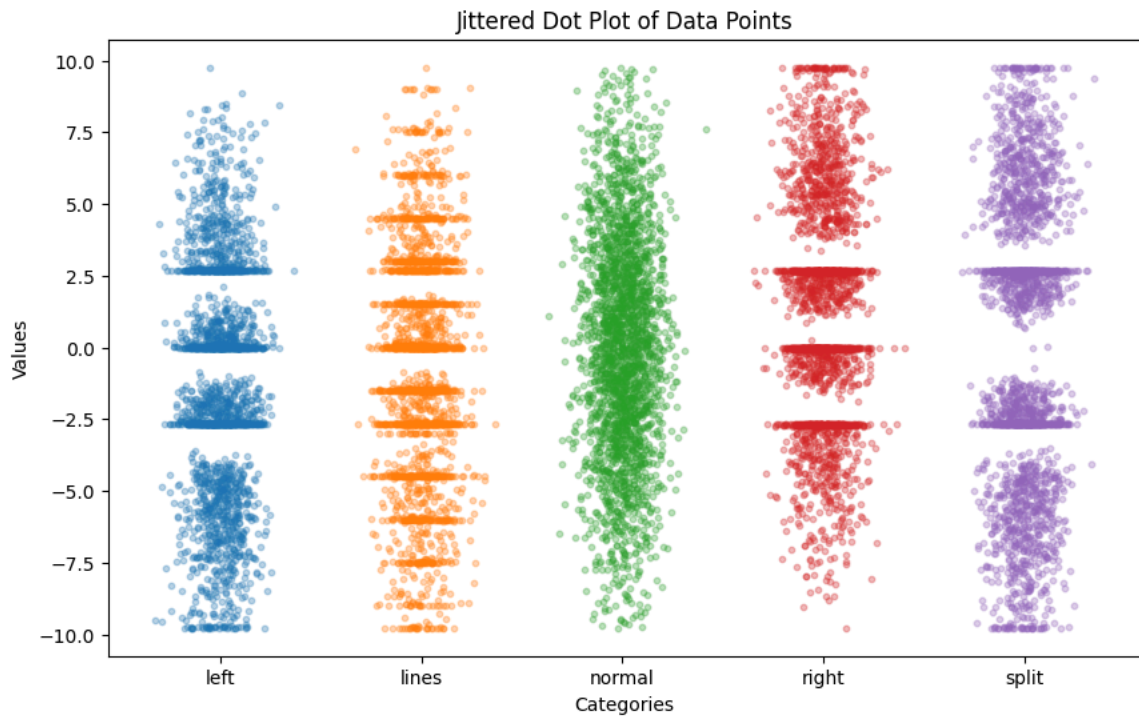


```
plt.figure(figsize=(10, 6))

for i, column in enumerate(df.columns):
    x = np.random.normal(i + 1, 0.1, size=df[column].shape) # Add jitter to x-coordinates
    plt.scatter(x, df[column], alpha=0.3, s=10)

plt.xticks(ticks=range(1, len(df.columns) + 1), labels=df.columns)
plt.title("Jittered Dot Plot of Data Points")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```

[4]

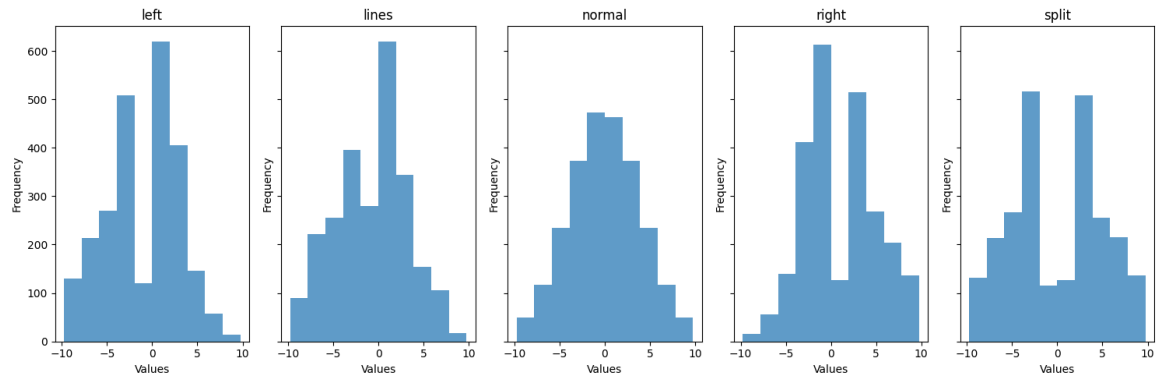


```
num_cols = len(df.columns)
fig, axes = plt.subplots(1, num_cols, figsize=(15, 5), sharey=True)

for i, column in enumerate(df.columns):
    axes[i].hist(df[column], bins=10, alpha=0.7)
    axes[i].set_title(column)
    axes[i].set_xlabel("Values")
    axes[i].set_ylabel("Frequency")

plt.tight_layout()
plt.show()
```

(4)

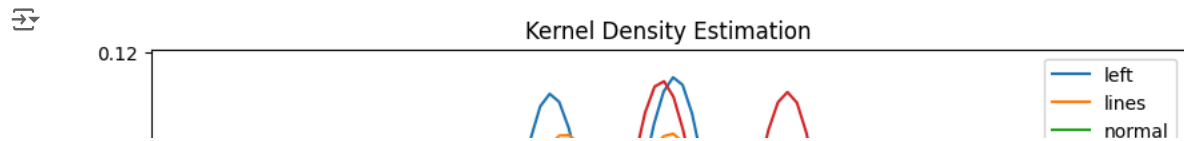


```
from scipy.stats import gaussian_kde

plt.figure(figsize=(10, 6))

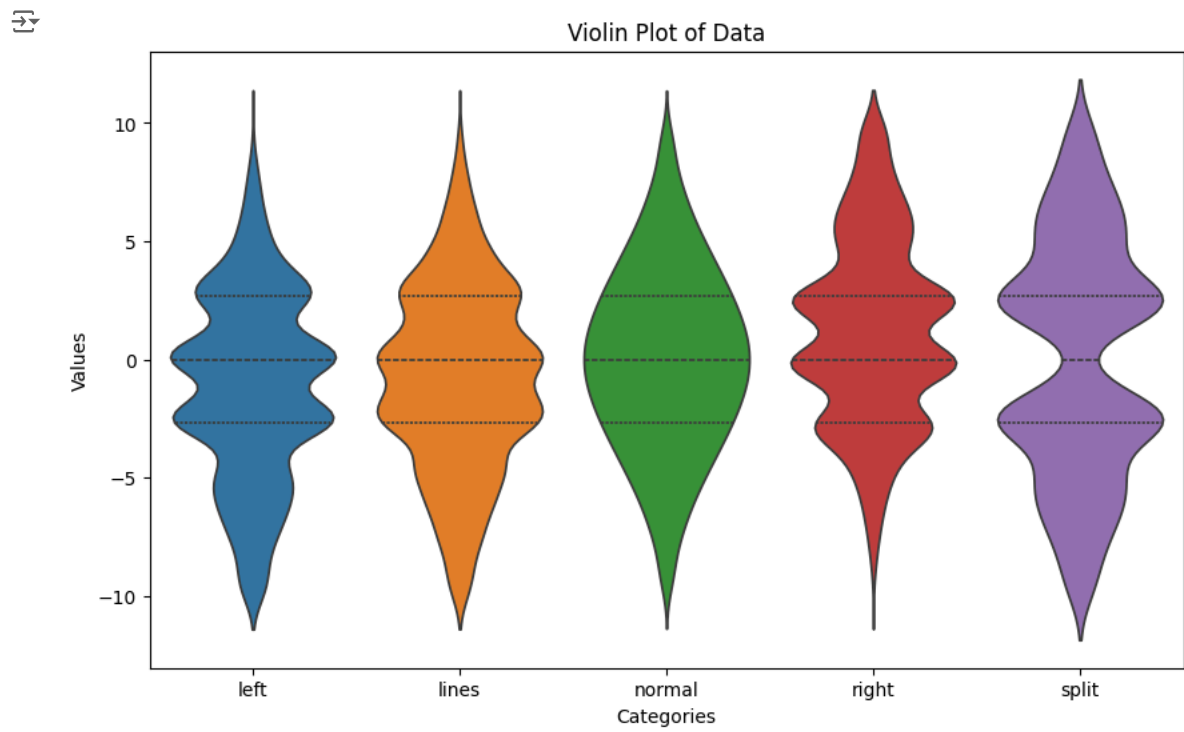
for column in df.columns:
    kde = gaussian_kde(df[column])
    x_vals = np.linspace(df[column].min(), df[column].max(), 100)
    plt.plot(x_vals, kde(x_vals), label=column)

plt.title("Kernel Density Estimation")
plt.xlabel("Values")
plt.ylabel("Density")
plt.legend()
plt.show()
```



```
import seaborn as sns
```

```
plt.figure(figsize=(10, 6))
sns.violinplot(data=df, inner="quartile")
plt.title("Violin Plot of Data")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```



```
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)
```

```
# Box Plot
df.boxplot(ax=axes[0])
axes[0].set_title("Box Plot")
axes[0].set_ylabel("Values")
```

```
# Jittered Dot Plot
for i, column in enumerate(df.columns):
    x = np.random.normal(i + 1, 0.1, size=df[column].shape)
    axes[1].scatter(x, df[column], alpha=0.7, s=10)
axes[1].set_title("Jittered Dot Plot")
```