

🔗 Muhammad Tahir

K214503

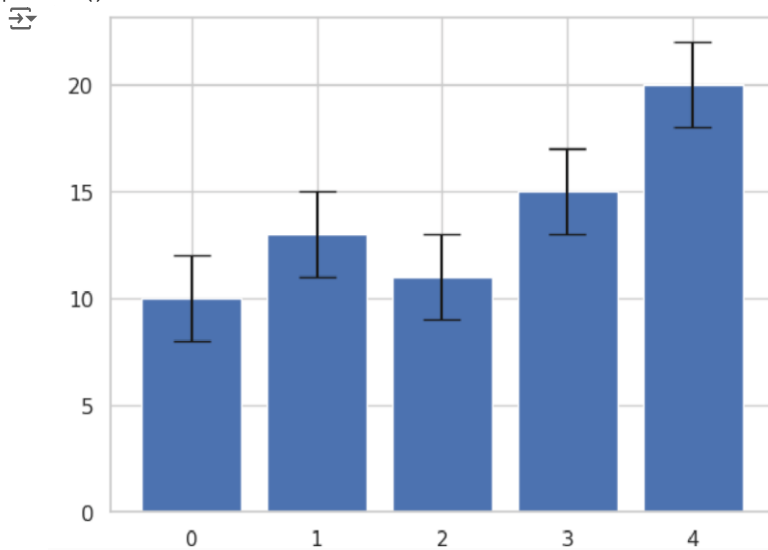
Data Science Visualization Lab

Part 02 & Part 03

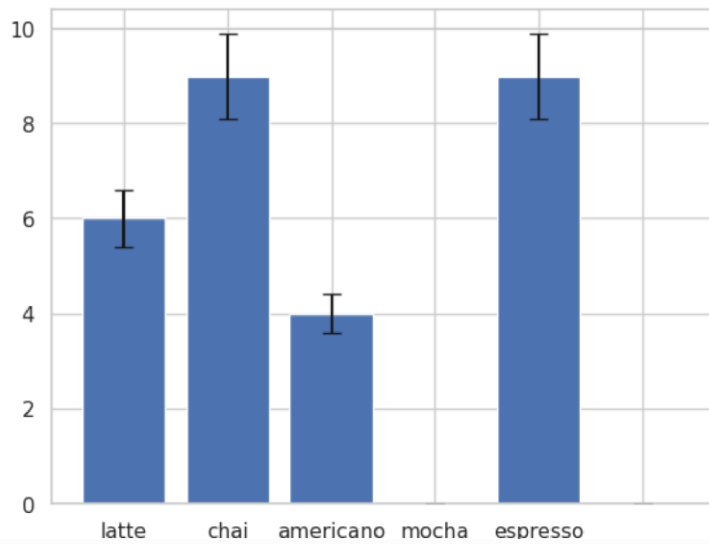
🔗 Error Bar Lab

```
import matplotlib.pyplot as plt
```

```
values = [10, 13, 11, 15, 20] yerr = 2  
plt.bar(range(len(values)), values, yerr=yerr, capsize=10)  
plt.show()
```

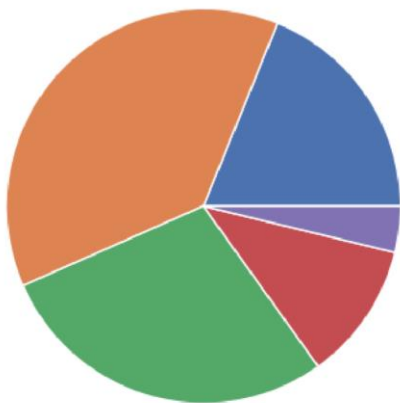


```
from matplotlib import pyplot as plt  
  
drinks = ["cappuccino", "latte", "chai", "americano", "mocha", "espresso"]  
ounces_of_milk = [6, 9, 4, 0, 9, 0] error = [0.6, 0.9, 0.4, 0, 0.9, 0]  
fig, ax = plt.subplots() ax.bar(range(len(drinks)), ounces_of_milk,  
yerr=error, capsize=5)  
  
ax.set_xticklabels(drinks)  
plt.show() <ipython-input-21-793d857f3832>:11: UserWarning:  
set_ticklabels() should only be  
used with a fixed number of ticks,  
i.e. after set_t  
ax.set_xticklabels(drinks)
```



```
budget_data = [500, 1000, 750, 300, 100]
```

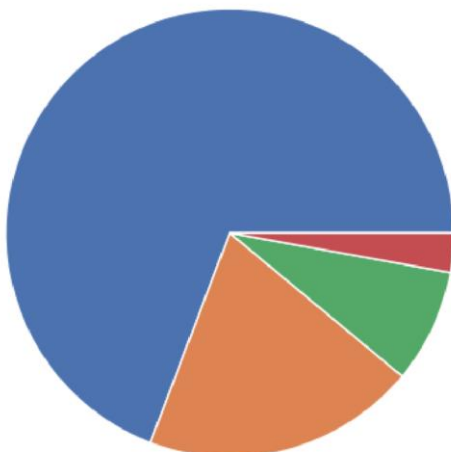
```
plt.pie(budget_data)  
plt.show()
```



```
from matplotlib import pyplot as plt  
import numpy as np
```

```
payment_method_names = ["Card Swipe", "Cash", "Apple Pay", "Other"]  
payment_method_freqs = [270, 77, 32, 11]
```

```
plt.pie(payment_method_freqs)  
plt.axis('equal') plt.show()
```

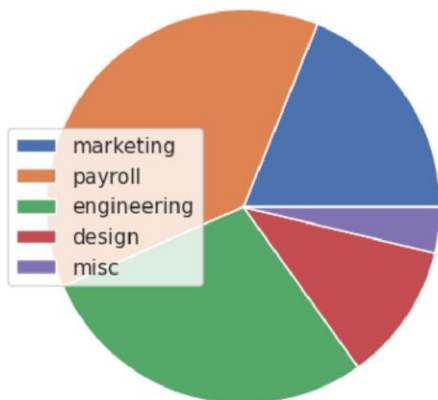




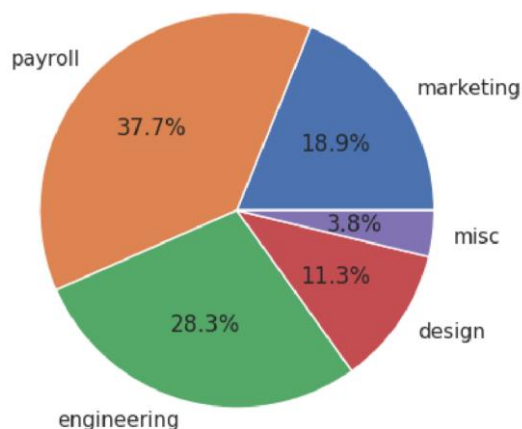
```
budget_data = [500, 1000, 750, 300, 100] budget_categories = ['marketing',  
'payroll', 'engineering', 'design', 'misc'] plt.pie(budget_data)  
plt.legend(budget_categories)
```



```
<matplotlib.legend.Legend at 0x7b580c7cf940>
```



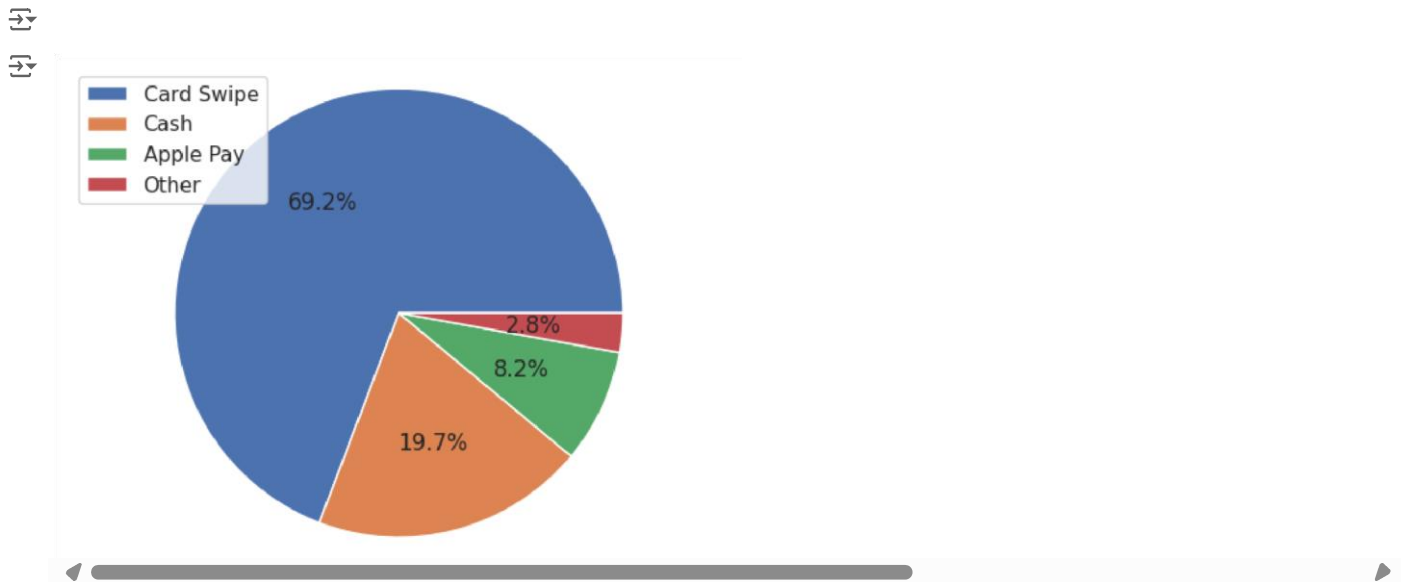
```
plt.pie(budget_data, labels=budget_categories, autopct='%0.1f%%')  
([<matplotlib.patches.Wedge at 0x7b5811617550>,  
<matplotlib.patches.Wedge at 0x7b5811616500>,  
<matplotlib.patches.Wedge at 0x7b5811616dd0>,  
<matplotlib.patches.Wedge at 0x7b5811616f80>,  
<matplotlib.patches.Wedge at 0x7b5811614700>],  
[Text(0.9123462485674189, 0.6145114504425102, 'marketing'),  
Text(-0.7892580098194788, 0.7662061040841398, 'payroll'),  
Text(-0.2899458620567297, -1.0610991457334136, 'engineering'),  
Text(0.9123462485674185, -0.6145114504425107, 'design'),  
Text(1.0922792053850405, -0.13010048994306064, 'misc')],  
[Text(0.49764340830950116, 0.3351880638777328, '18.9%'), Text(-  
0.430504368992443, 0.41793060222771256, '37.7%'),  
Text(-0.1581522883945798, -0.5787813522182255, '28.3%'),  
Text(0.49764340830950093, -0.3351880638777331, '11.3%'),  
Text(0.5957886574827493, -0.0709639036053058, '3.8%')])
```



```
from matplotlib import pyplot as plt
```

```
payment_method_names = ["Card Swipe", "Cash", "Apple Pay",  
"Other"] payment_method_freqs = [270,  
77, 32, 11]
```

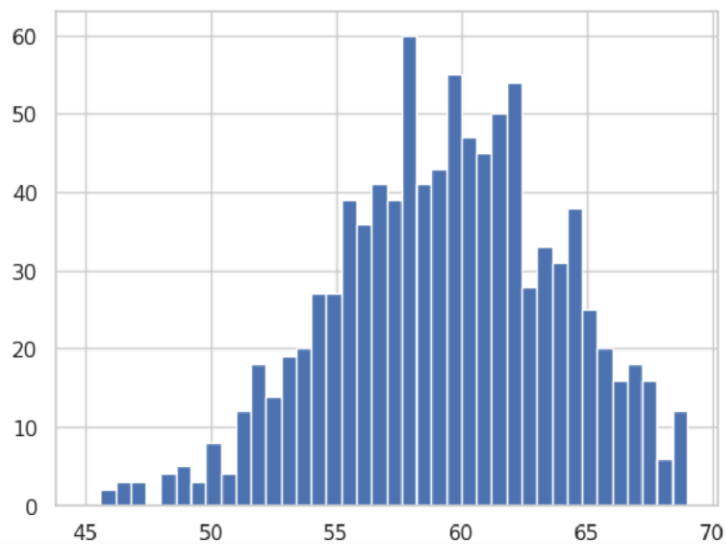
```
plt.pie(payment_method_freqs, autopct="%0.1f%%")  
plt.axis('equal')  
plt.legend(payment_method_names) plt.show()
```



histogram

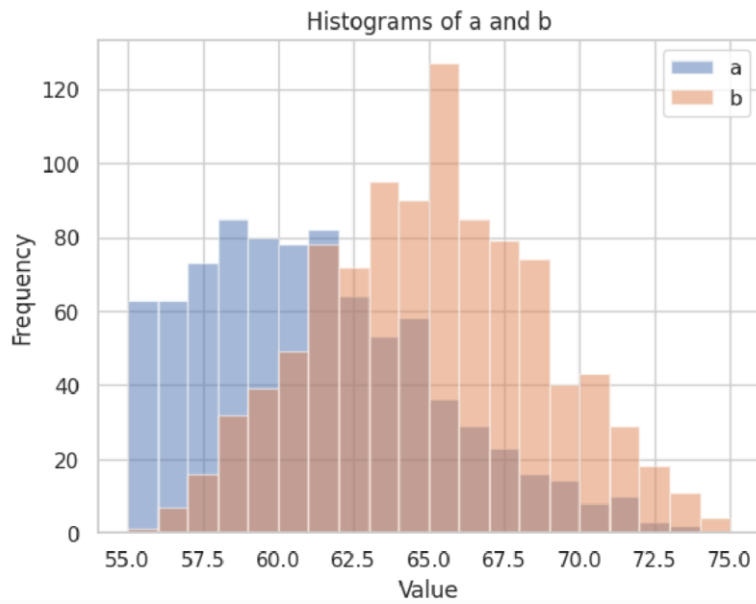
```
import matplotlib.pyplot as plt
import numpy as np
```

```
np.random.seed(0) data = np.random.normal(loc=60,
scale=5, size=1000) plt.hist(data, range=(45,69),
bins=40) plt.show()
```



```
np.random.seed(0) a = np.random.normal(loc=60,
scale=5, size=1000) b = np.random.normal(loc=65,
scale=4, size=1000)
```

```
plt.hist(a, range=(55, 75), bins=20, alpha=0.5, label='a')
plt.hist(b, range=(55, 75), bins=20, alpha=0.5, label='b')
plt.legend() plt.xlabel('Value') plt.ylabel('Frequency')
plt.title('Histograms of a and b') plt.show()
```



Seaborn

```
import csv

data = [      ['col1',
'col2'],
        [1, 'A'],
        [2, 'B'],
        [3, 'C'],
        [4, 'D'],
        [5, 'E'],
]

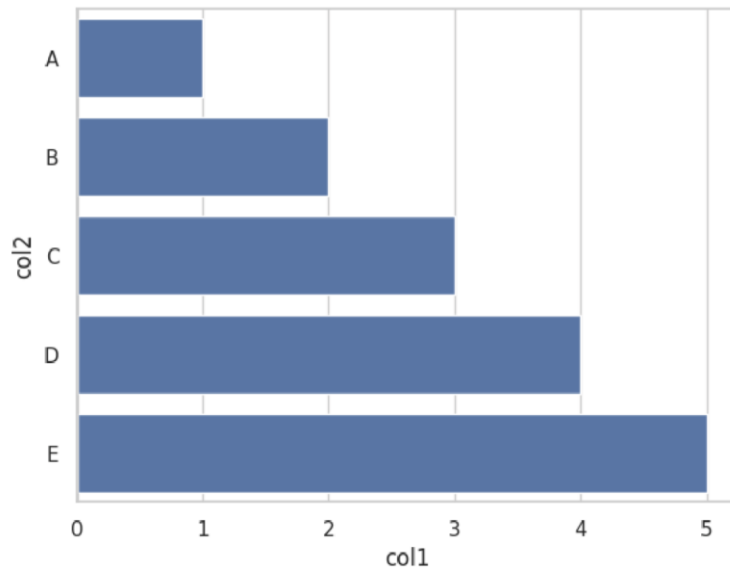
with open('result.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(data)

import warnings
warnings.filterwarnings('ignore')
```

```
import pandas as pd from matplotlib
import pyplot as plt import seaborn
as sns df =
pd.read_csv('result.csv')
```

```
sns.barplot(
data= df, x=
"col1",
y= "col2")
```

<Axes: xlabel='col1', ylabel='col2'>



```
import csv
```

```
data = [
    ['student', 'grade'],
    [1, 'A'],
    [2, 'B'],
    [3, 'C'],
    [4, 'D'],
    [5, 'B'],
]
```

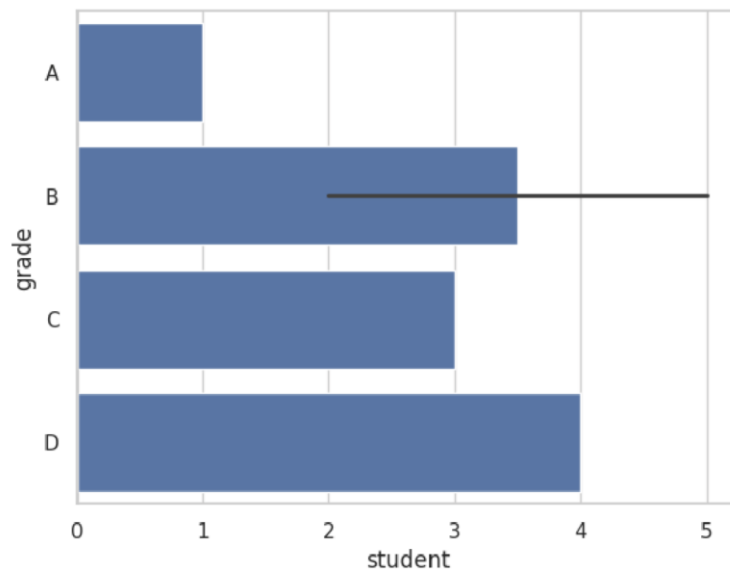
```
with open('gradebook.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(data)
```

```
import pandas as pd from matplotlib import pyplot
```

```
as plt gradebook = pd.read_csv("gradebook.csv")
```

```
sns.barplot(data=gradebook, x="student", y="grade")
```

<Axes: xlabel='student', ylabel='grade'>



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

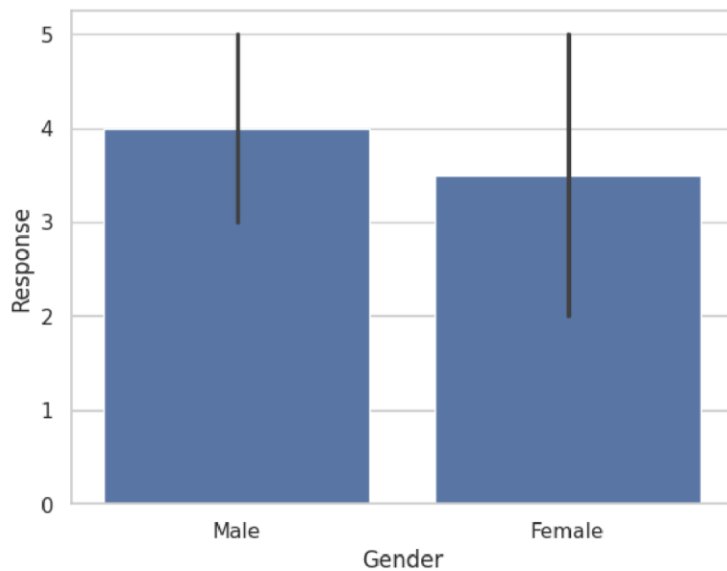
data = [
    ['Gender', 'Response', 'Age Range'],
    ['Male', 4, '20-29'],
    ['Female', 3, '30-39'],
    ['Male', 5, '20-29'],
    ['Female', 2, '40-49'],
    ['Male', 4, '20-29'],
    ['Female', 5, '30-39'],
    ['Male', 3, '20-29'],
    ['Female', 4, '40-49']
]

with open('survey.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(data)

warnings.filterwarnings('ignore')

df = pd.read_csv("survey.csv")

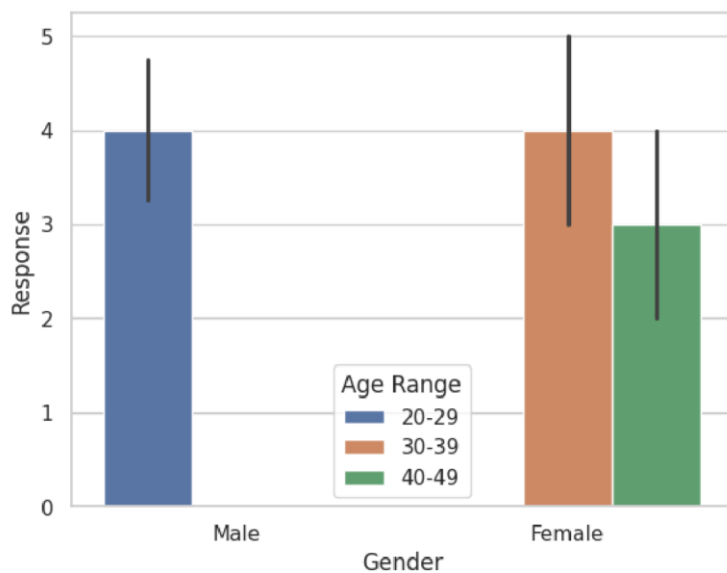
sns.barplot(data=df,
x='Gender',
y='Response',
estimator=np.median)
plt.show()
```



```
sns.barplot(data=df,  
x="Gender",  
y="Response",  
hue="Age Range")
```

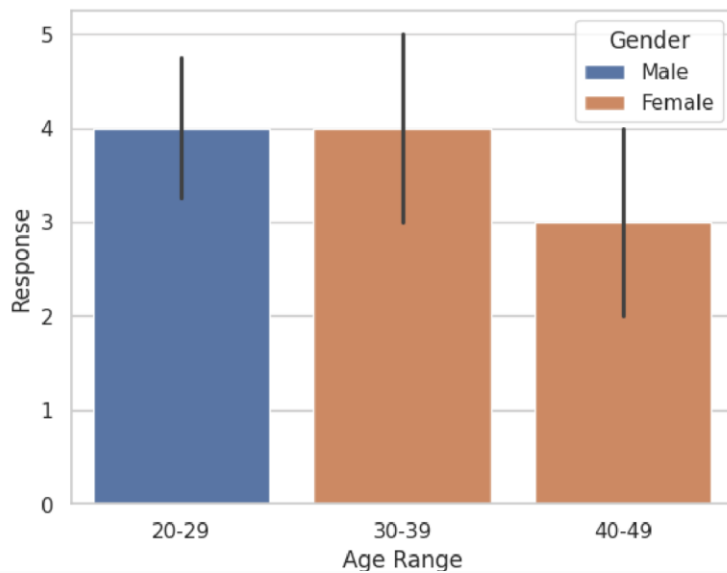


<Axes: xlabel='Gender', ylabel='Response'>



```
df = pd.read_csv("survey.csv")
```

```
sns.barplot(data=df,  
x="Age Range",  
y="Response",  
hue="Gender")  
plt.show()
```

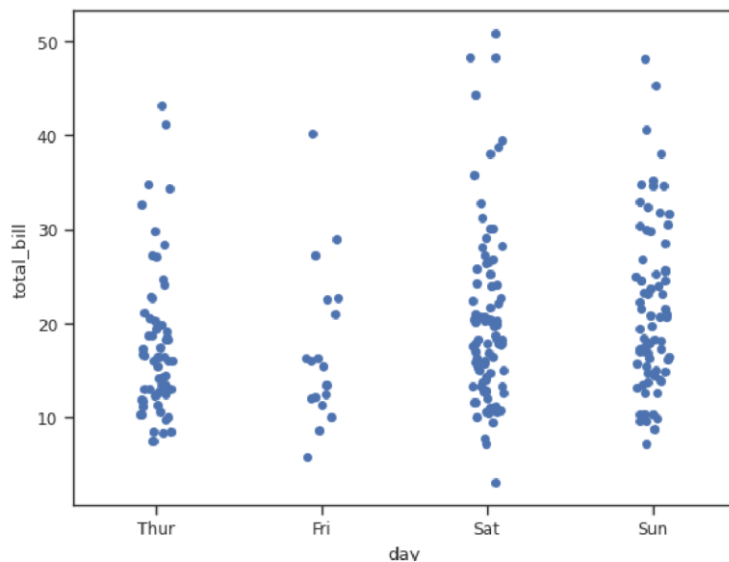



? Load_dataset with Seaborn

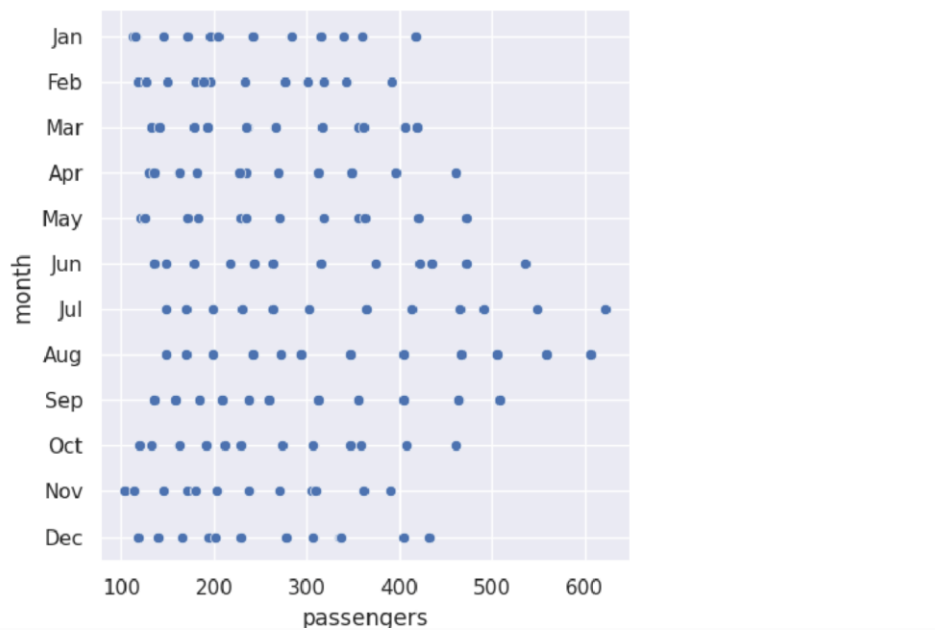
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.set_style("ticks")
tips = sns.load_dataset("tips")
sns.set_context("paper")
sns.stripplot(x="day", y="total_bill", data=tips)
```

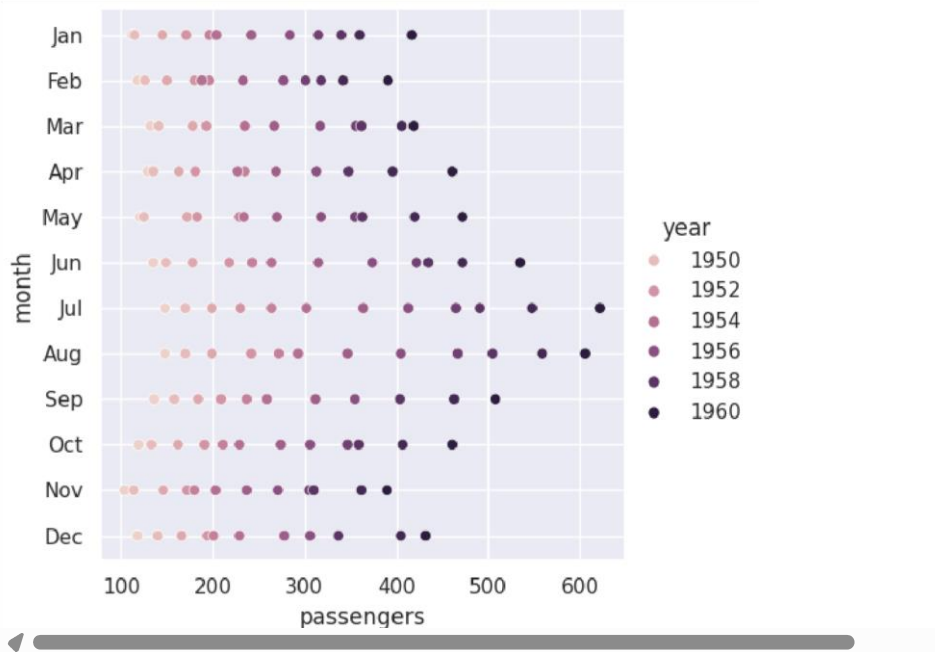
<Axes: xlabel='day', ylabel='total bill'>



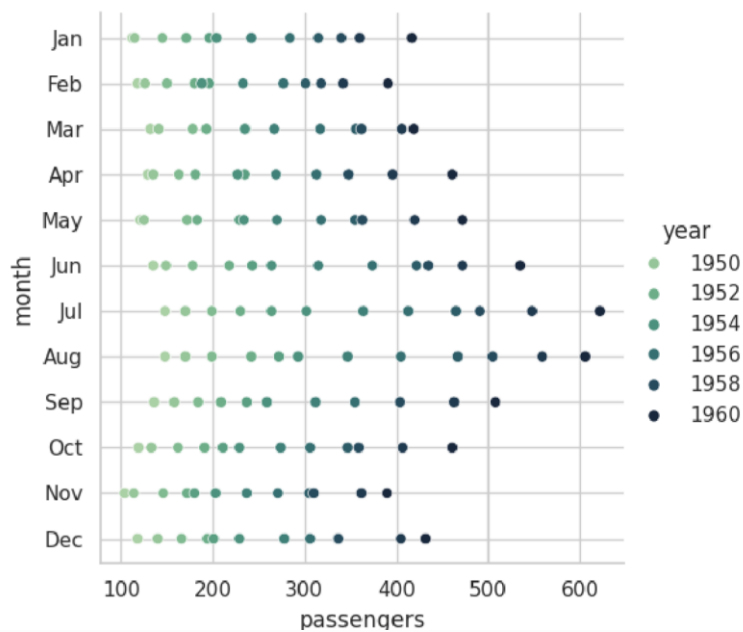
```
f = sns.load_dataset("flights")
sns.set(style="darkgrid")
sns.relplot(x="passengers", y="month", data=f)
<seaborn.axisgrid.FacetGrid at 0x7b580c43bf70>
```



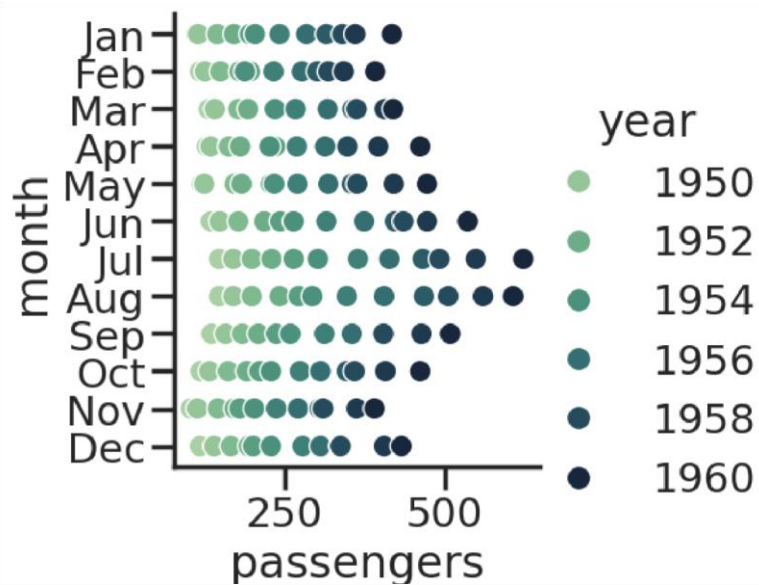
```
f = sns.load_dataset("flights") sns.set(style="darkgrid")
sns.relplot(x="passengers", y="month", hue="year", data=f);
```



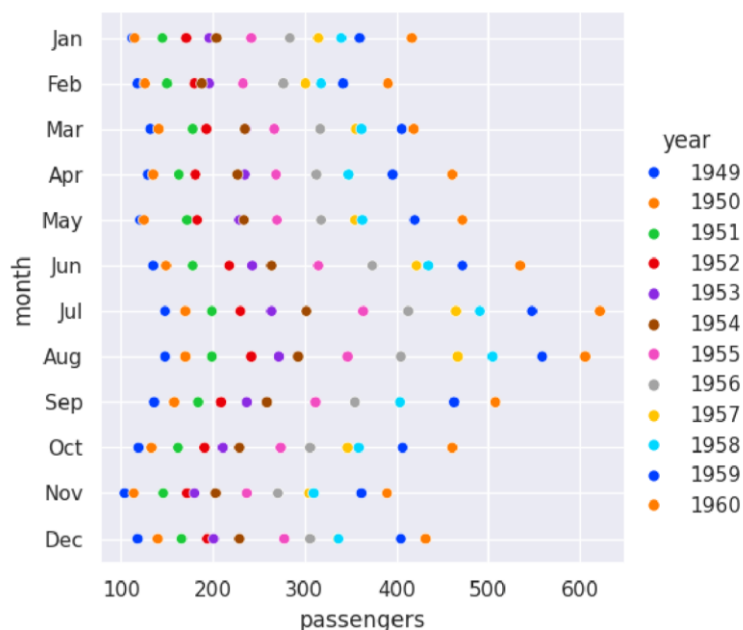
```
sns.set(style="whitegrid") sns.relplot(x="passengers", y="month", hue="year",
palette="ch:r=-.5,l=.75", data=f);
```



```
sns.set_style("ticks") sns.set_context("poster") sns.relplot(x="passengers",  
y="month", hue="year", palette="ch:r=-.5,l=.75", data=f);
```

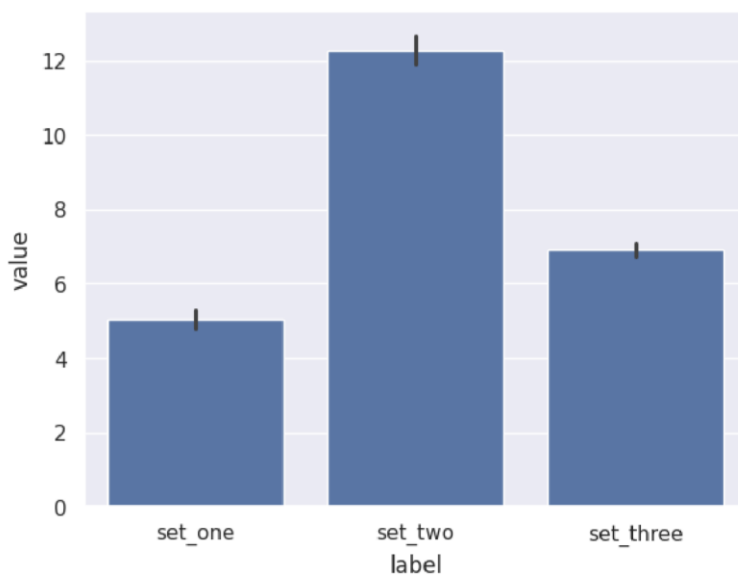


```
sns.set(style="darkgrid") f = sns.load_dataset("flights")  
sns.relplot(x="passengers", y="month", hue="year", palette="bright", data=f);
```



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
n = 500
#Load data from a text file
dataset1 = np.genfromtxt("dataset1.csv", delimiter=",")
dataset2 = np.genfromtxt("dataset2.csv", delimiter=",")
dataset3 = np.genfromtxt("dataset3.csv", delimiter=",")
df = pd.DataFrame({
    "label": ["set_one"] * n + ["set_two"] * n + ["set_three"] * n,
    "value": np.concatenate([dataset1, dataset2, dataset3])
})
```

```
sns.set()
sns.barplot(data=df, x='label', y='value')
plt.show()
```



KDE Plots

```
import numpy as np
```

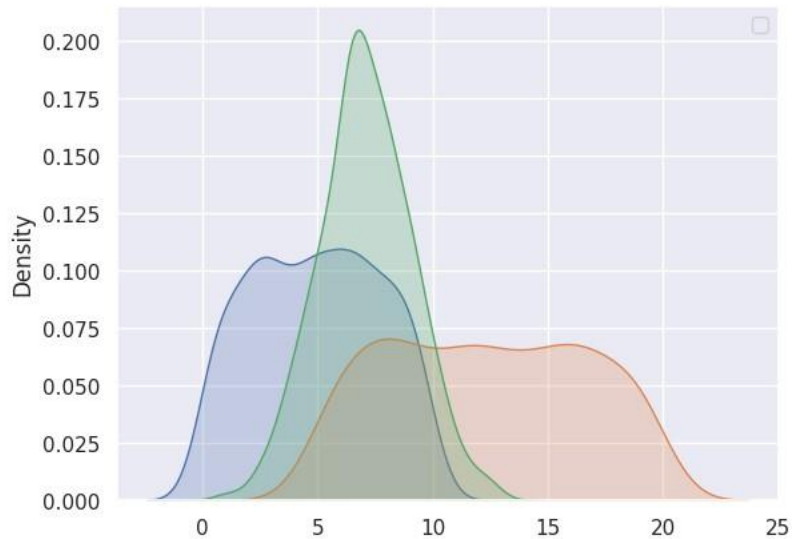
```
n = 500
dataset1 = np.random.rand(n) * 10
np.savetxt("dataset1.csv", dataset1, delimiter=",")
```

```
dataset2 = np.random.rand(n) * 15 + 5
np.savetxt("dataset2.csv", dataset2, delimiter=",")
```

```
dataset3 = np.random.normal(loc=7, scale=2, size=n)
np.savetxt("dataset3.csv", dataset3, delimiter=",")
```

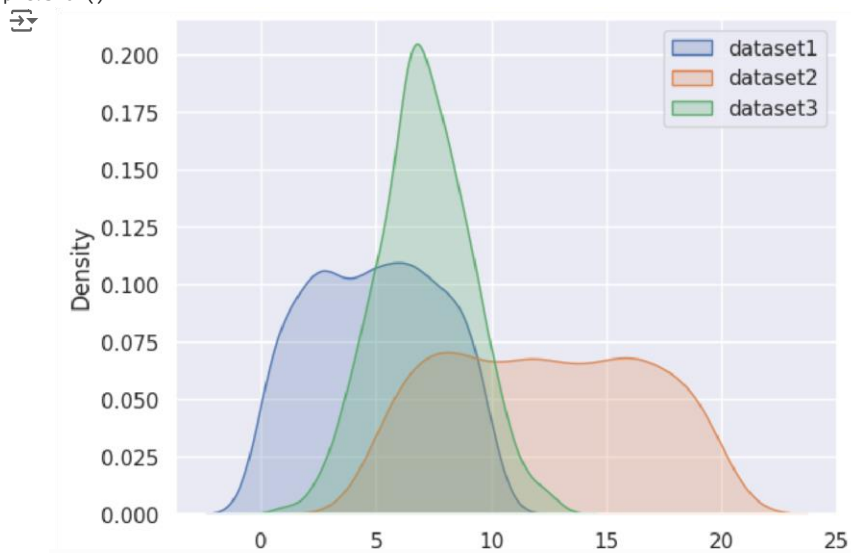
```
sns.kdeplot(dataset1, shade=True)
sns.kdeplot(dataset2, shade=True)
sns.kdeplot(dataset3, shade=True)
plt.legend()
plt.show()
```

⚠ WARNING:matplotlib.legend:No artists with labels found to put in legend. Note that artists whose label start with an underscore are



```
sns.kdeplot(dataset1, shade=True, palette="dark", label="dataset1")
sns.kdeplot(dataset2, shade=True, palette="bright", label="dataset2")
sns.kdeplot(dataset3, shade=True, palette="muted", label="dataset3")
```

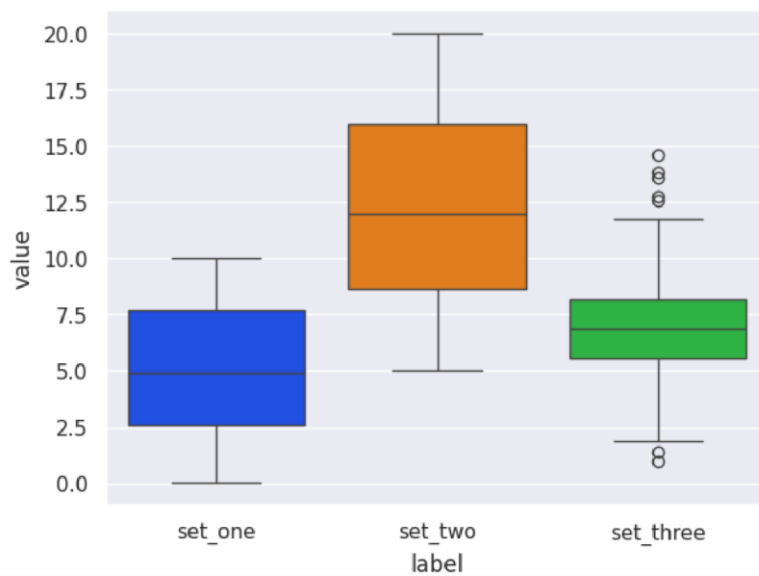
```
plt.legend()
plt.show()
```





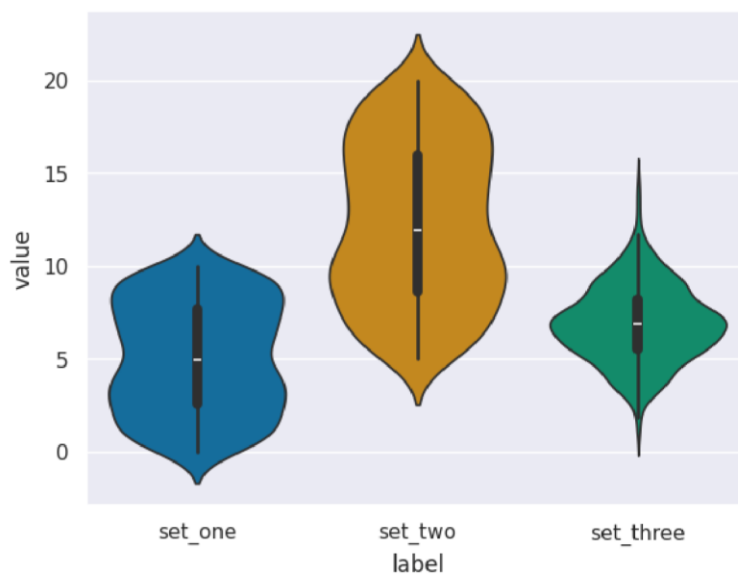
Box Plots

```
sns.boxplot(data=df, x='label', palette="bright", y='value')  
plt.show()
```



Violin plots

```
sns.violinplot(data=df, x="label", palette="colorblind", y="value")  
plt.show()
```



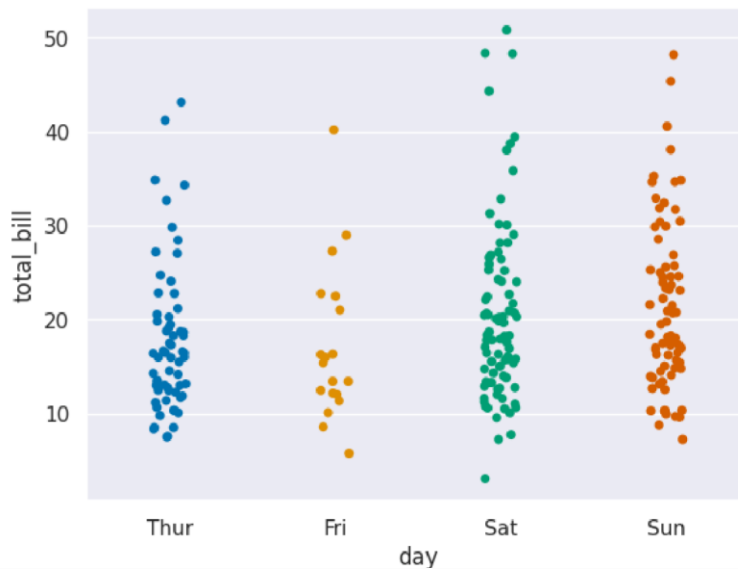
Seaborn Styling, Part 1: Figure Style and Scale

1. Built-in Themes:

```

sns.set_style("darkgrid")
sns.stripplot(x="day", y="total_bill", palette="colorblind", data=tips)

```



```

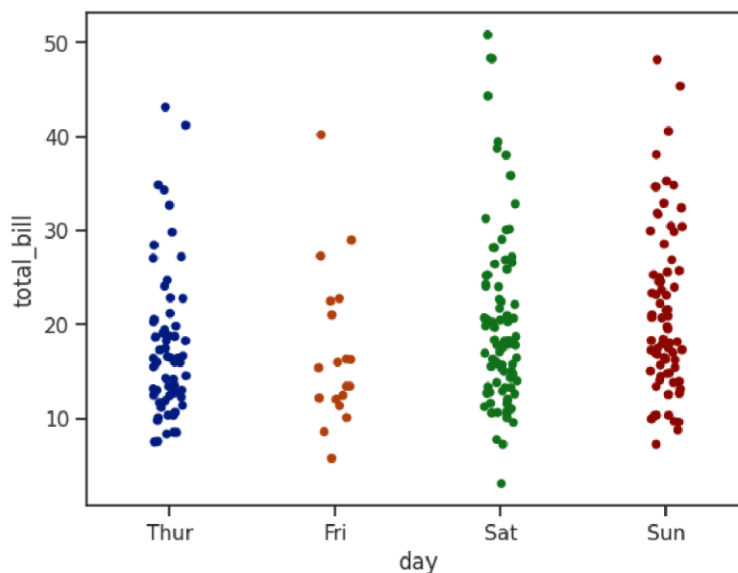
sns.set_style("ticks")
sns.stripplot(x="day", y="total_bill", palette="dark", data=tips)

```

```

<Axes: xlabel='day', ylabel='total bill'>

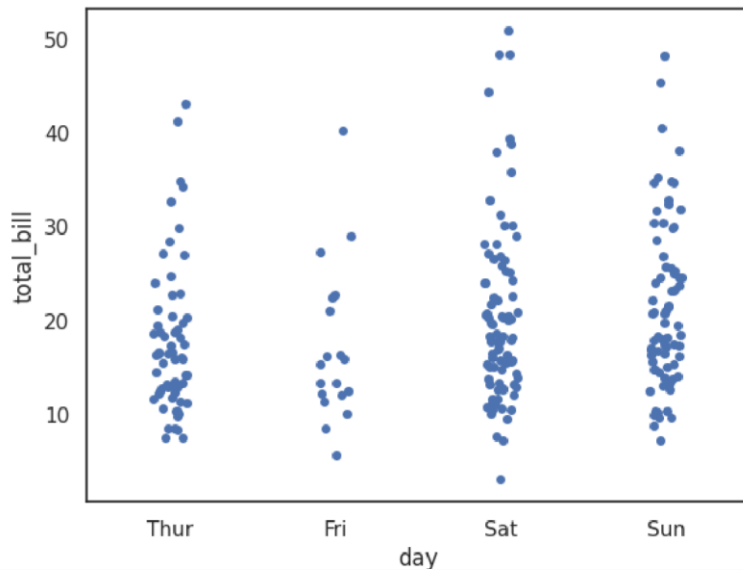
```



```

import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="white", color_codes=True)
a = sns.load_dataset("tips")
sns.boxplot(x="day", y="total_bill", palette="bright", data=a)

```

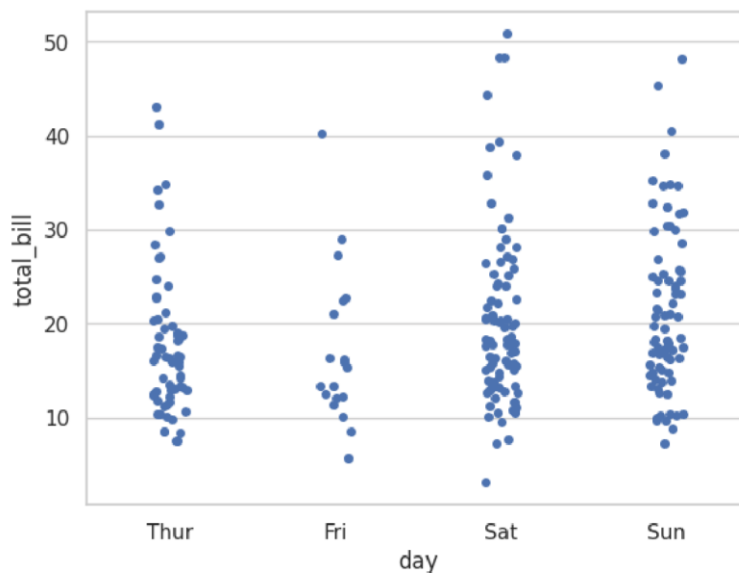


2. Grids:

```
sns.set_style("whitegrid")
sns.stripplot(x="day", y="total_bill", data=a)
```



<Axes: xlabel='day', ylabel='total bill'>

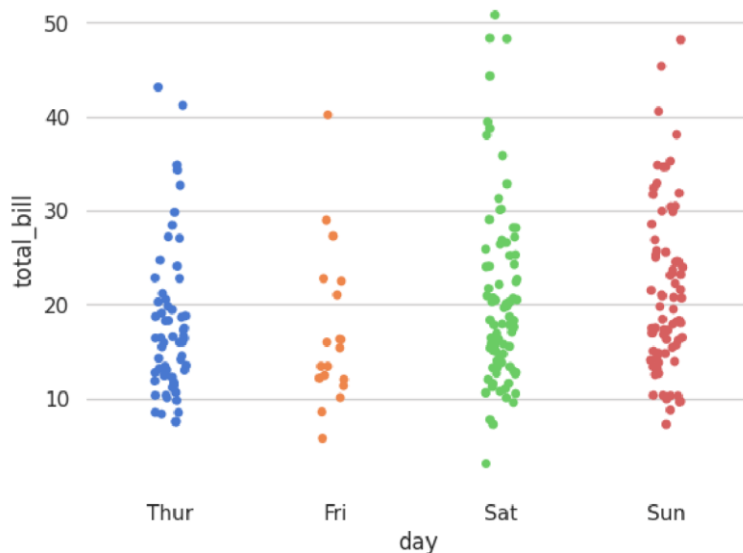


```
sns.set_style("white")
sns.stripplot(x="day", y="total_bill", data=a)
```

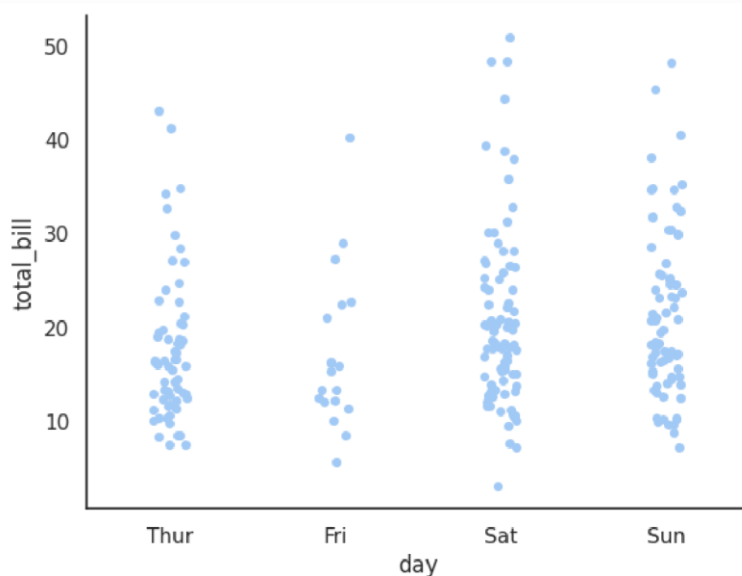
3. Despine

```
sns.set_style("white") sns.stripplot(x="day", y="total_bill", data=a) sns.despine()
```


[↵]



[↵]



```
sns.set_style("whitegrid") sns.stripplot(x="day", y="total_bill",palette="muted", data=a) sns.despine(left=True, bottom=True)
```

4. Working with Palettes

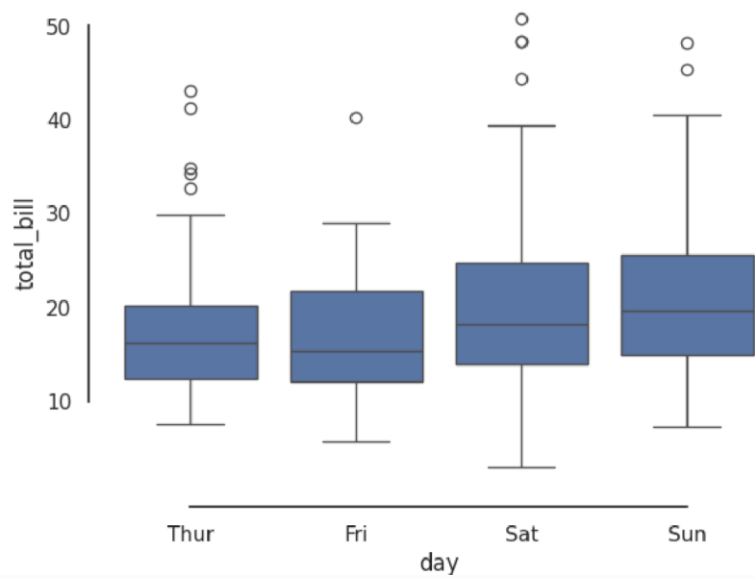
```
# Save a palette to a variable: palette = sns.color_palette("bright")
```

```
# Use palplot and pass in the variable: sns.palplot(palette)
```

[↵]




```
import seaborn as sns import matplotlib.pyplot as plt sns.set(style="white", color_codes=True) a = sns.load_dataset("tips") sns.boxplot(x="day", y="total_bill", data=a); sns.despine(offset=10, trim=True);
```

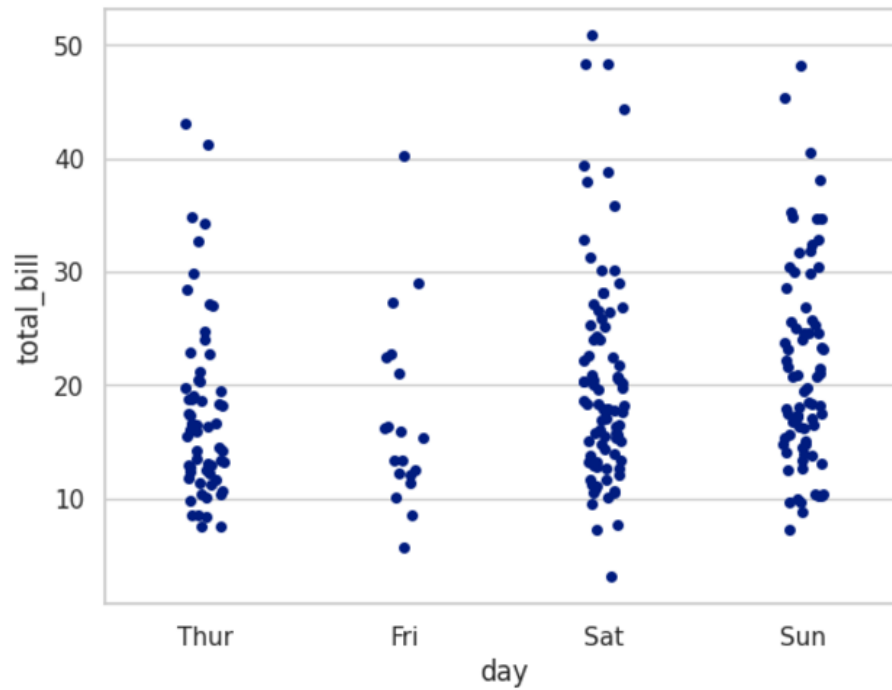




4. Working with Palettes

```
[ ] # Set the palette using the name of a palette:  
sns.set_palette("dark")  
  
sns.stripplot(x="day", y="total_bill", data=tips)
```

 <Axes: xlabel='day', ylabel='total_bill'>



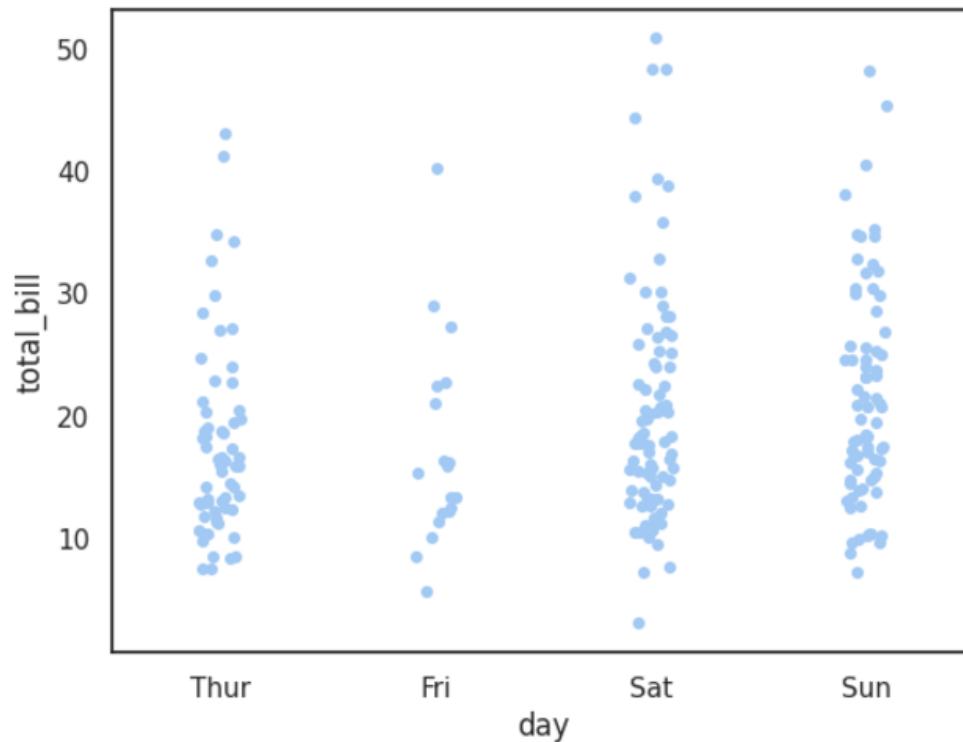


```
# Set the palette to the "pastel" default palette:
sns.set_palette("pastel")

# plot using the "pastel" palette
sns.stripplot(x="day", y="total_bill", data=tips)
```



```
<Axes: xlabel='day', ylabel='total_bill'>
```

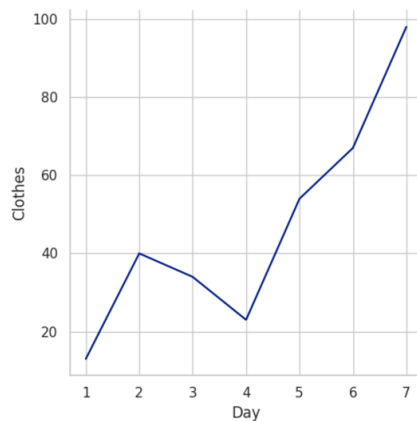


lineplot():

```
[ ] a=pd.DataFrame({'Day':[1,2,3,4,5,6,7], 'Grocery':[30,80,45,23,51,46,76], 'Clothes':[13,40,34,23,54,67,98], 'Utensils':[12,32,27,56,87,54,34]}, index=[1,2,3,4,5,6,7])
sns.relplot(x="Day", y="Clothes", kind="line", data=a)
```



```
<seaborn.axisgrid.FacetGrid at 0x7b581236b670>
```

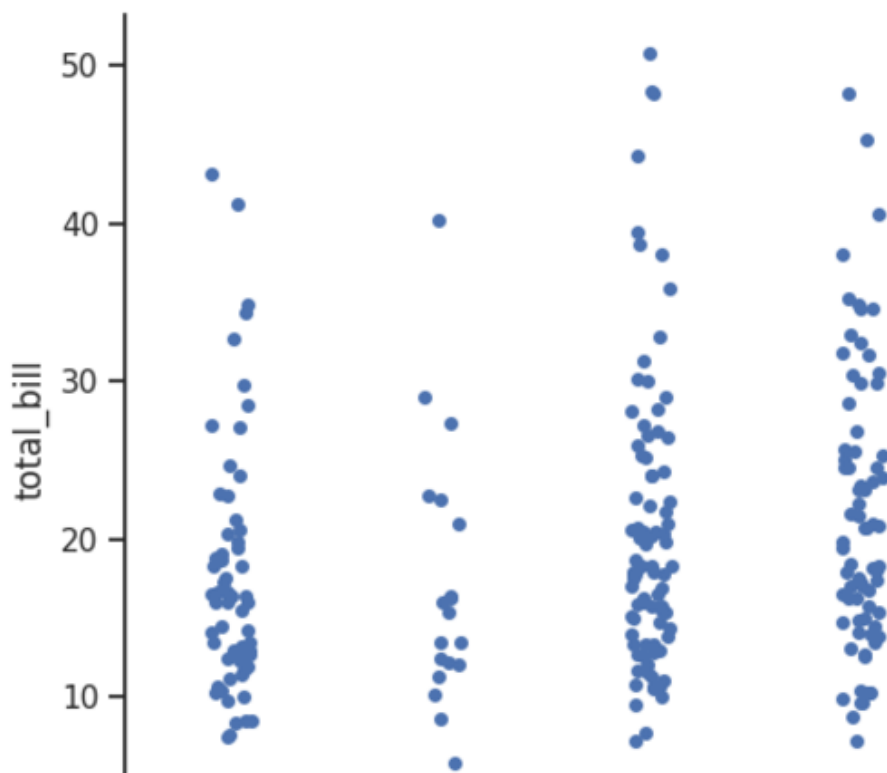




✓ Plotting with Categorical Data:

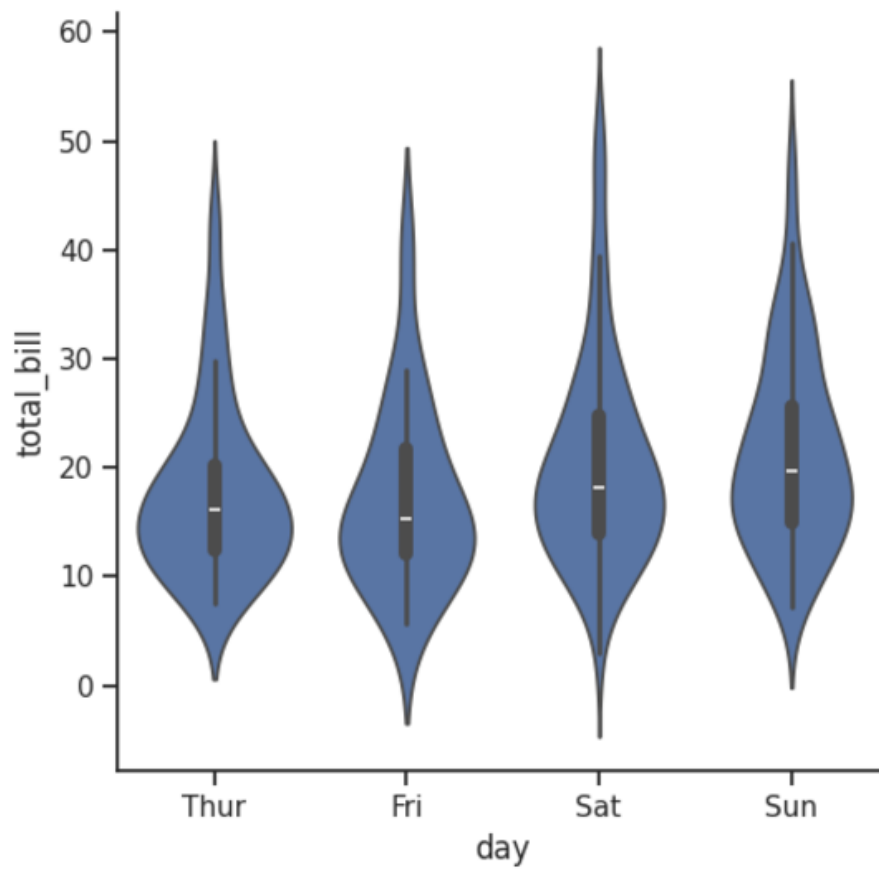
catplot():

```
[ ] import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="ticks", color_codes=True)
a = sns.load_dataset("tips")
sns.catplot(x="day", y="total_bill", data=a);
```





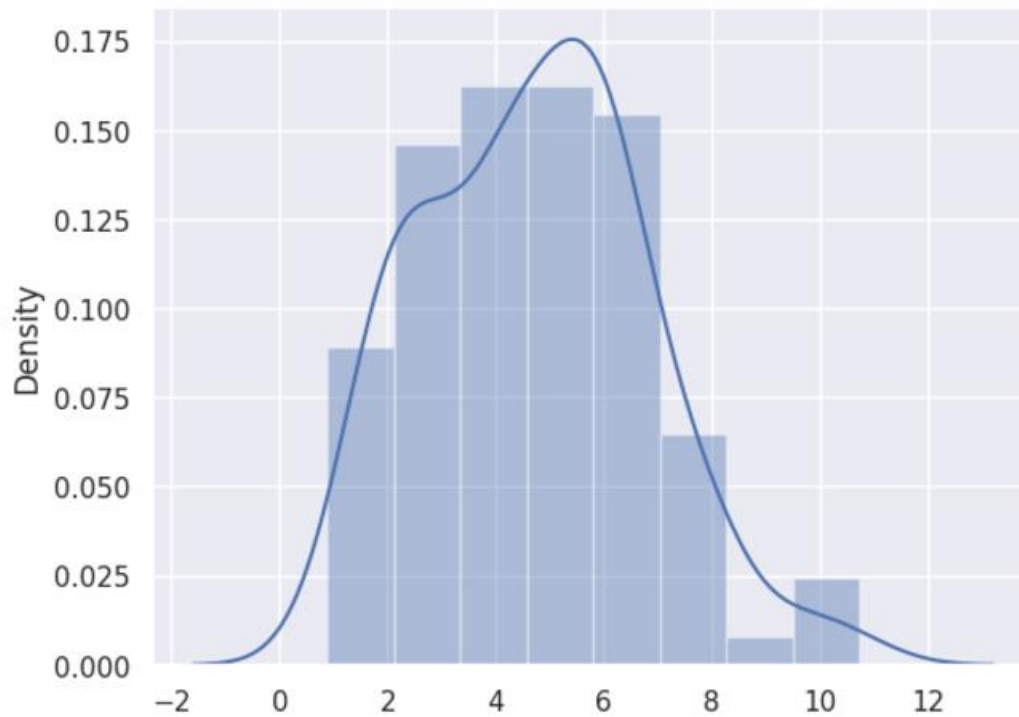
```
[ ] import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="ticks", color_codes=True)
a = sns.load_dataset("tips")
sns.catplot(x="day", y="total_bill", kind="violin", data=a);
```





```
[ ] import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
sns.set(color_codes=True)

a = np.random.normal(loc=5,size=100,scale=2)
sns.distplot(a);
```





✓ Class Task: Draw the heatmap

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

np.random.seed(0)
n = 100
col1 = np.random.rand(n)
col2 = 2 * col1 + np.random.normal(0, 0.5, n)
col3 = col1 + col2 + np.random.normal(0, 1, n)

data = {'col1': col1, 'col2': col2, 'col3': col3}
df = pd.DataFrame(data)

correlation_matrix = df.corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True)
plt.title('Correlation Matrix Heatmap')
plt.show()
```

