



```
In [ ]: '''
SQL
structure and unstructure data store in container that can be store in data ba
database is container where data can be stored.
structure db example --> sql, mysql, oracle,etc
unstructure database example --> mongo db, cassandar,habse, etc
cloud db
***vector db --> vector db using llm model , generative ai
text will convert to vectore
ex -->
pinecone, milvus, qdrant, faiss, chromadb***

server : collection of databases (application server and production server)
All servers must has to install in premises(list of database is available)

cloud : cloude does not required pysical in premises
All cloude can store in datacenter

relational db - data analytics
data engineer()
1- key value db --> every data store as a key value pair (redis db , amzon dy
2-document db --> mongo db , couch db
3-graph db --> storing grph lik sturcture (Neo4j, )
4-wide column db --> store in columns rather than row (Apache casandra, hbase)
5-search engine db --.>
6-timeseries db

'''
```

```
In [ ]: import warnings
warnings.filterwarnings("ignore", category=FutureWarnings)
```

```
-----
NameError                                Traceback (most recent call last)
/tmp/ipython-input-2368578896.py in <cell line: 0>()
      1 import warnings
----> 2 warnings.filterwarnings("ignore", category=FutureWarnings)

NameError: name 'FutureWarnings' is not defined
```

```
In [ ]: import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [ ]: sns.get_dataset_names()
```

```
Out[ ]: ['anagrams',
        'anscombe',
        'attention',
        'brain_networks',
        'car_crashes',
        'diamonds',
        'dots',
        'dowjones',
        'exercise',
        'flights',
        'fmri',
        'geyser',
        'glue',
        'healthexp',
        'iris',
        'mpg',
        'penguins',
        'planets',
        'seaice',
        'taxi',
        'tips',
        'titanic']
```

```
In [ ]: tips = sns.load_dataset("tips")
        sns.set_theme(style='darkgrid')
```

```
In [ ]: tips
```

```
Out[ ]:
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

```
In [ ]: tips.to_csv("tips_dataset.csv", index=False)
```

```
In [ ]: import os
os.getcwd()
```

```
Out[ ]: '/content'
```

```
In [ ]: plt.figure(figsize=(8,6))
```

```
Out[ ]: <Figure size 800x600 with 0 Axes>
<Figure size 800x600 with 0 Axes>
```

```
In [ ]: sns.scatterplot(data = tips, x = "total_bill", y="tip", hue="time", size="size")
plt.title("ScatterPlot : Total bill vs Tips by the Time and Size")
plt.show()
```

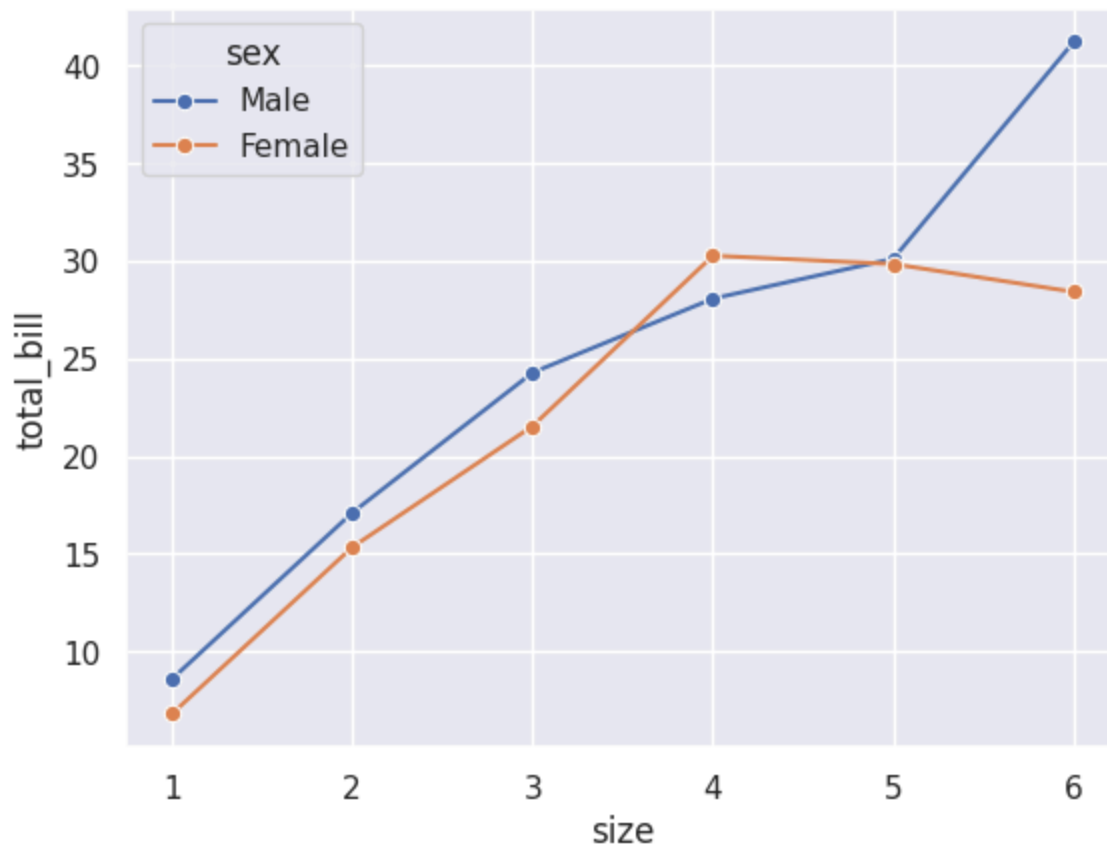


```
In [ ]: sns.lineplot(data=tips, x="size", y="total_bill", hue='sex', ci=None, marker="o")
plt.show()
```

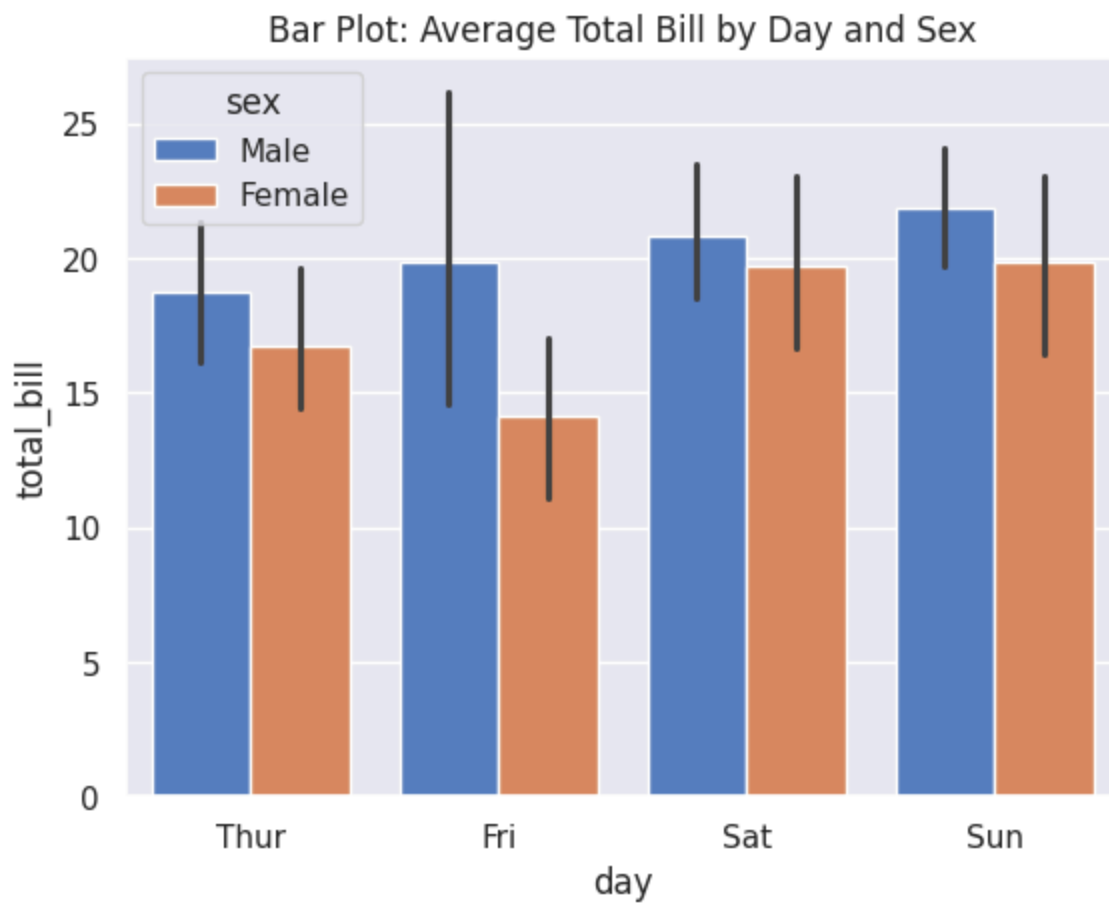
/tmp/ipython-input-2049012410.py:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

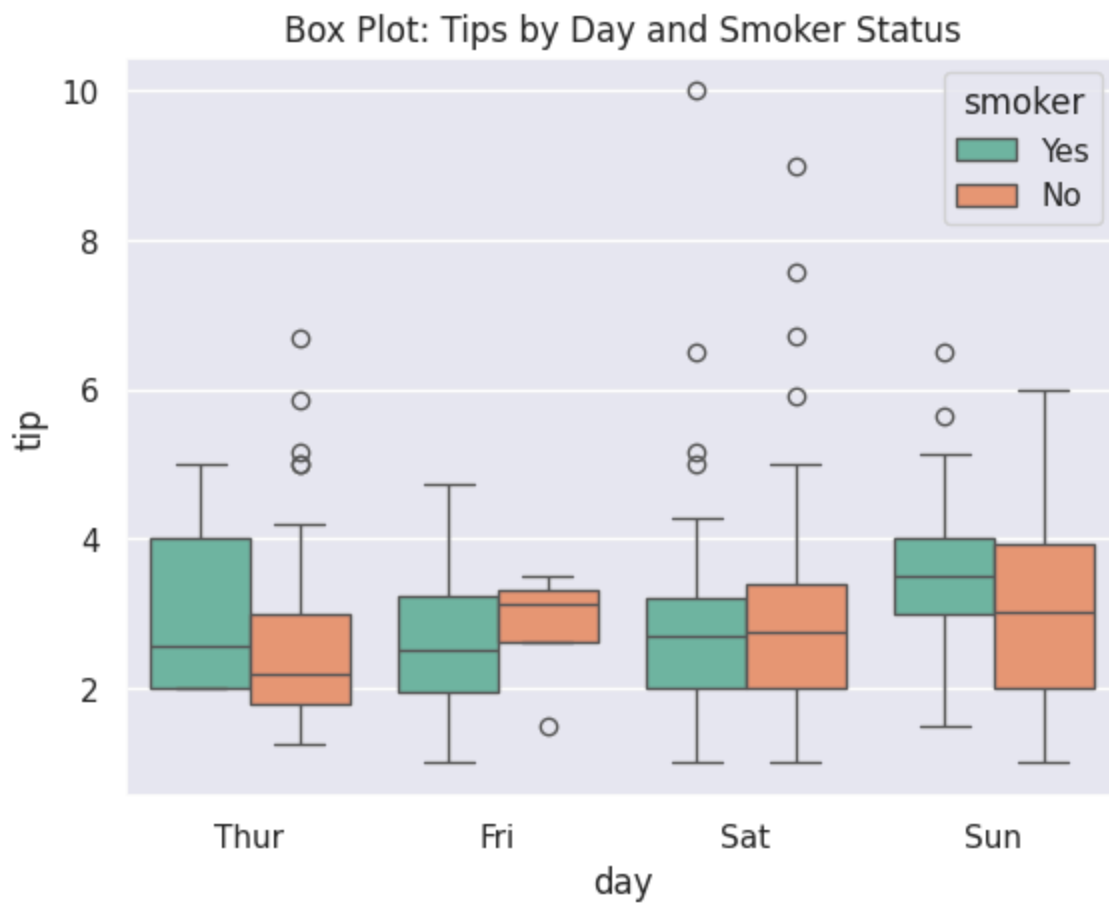
```
sns.lineplot(data=tips, x="size", y="total_bill", hue='sex', ci=None, marker="o")
```



```
In [ ]: sns.barplot(data=tips, x="day", y="total_bill", hue="sex", palette="muted")
plt.title("Bar Plot: Average Total Bill by Day and Sex")
plt.show()
```

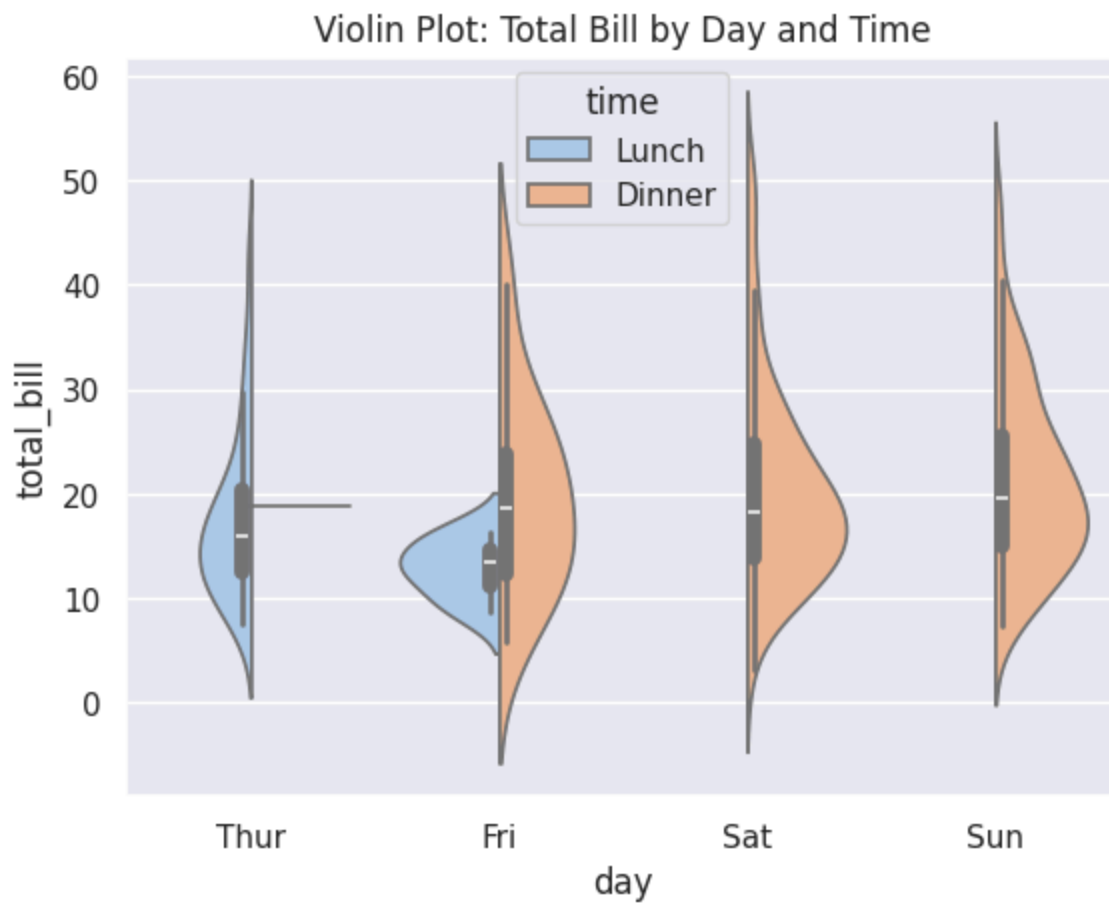


```
In [ ]: sns.boxplot(data=tips, x="day", y="tip", hue="smoker", palette="Set2")
plt.title("Box Plot: Tips by Day and Smoker Status")
plt.show()
```

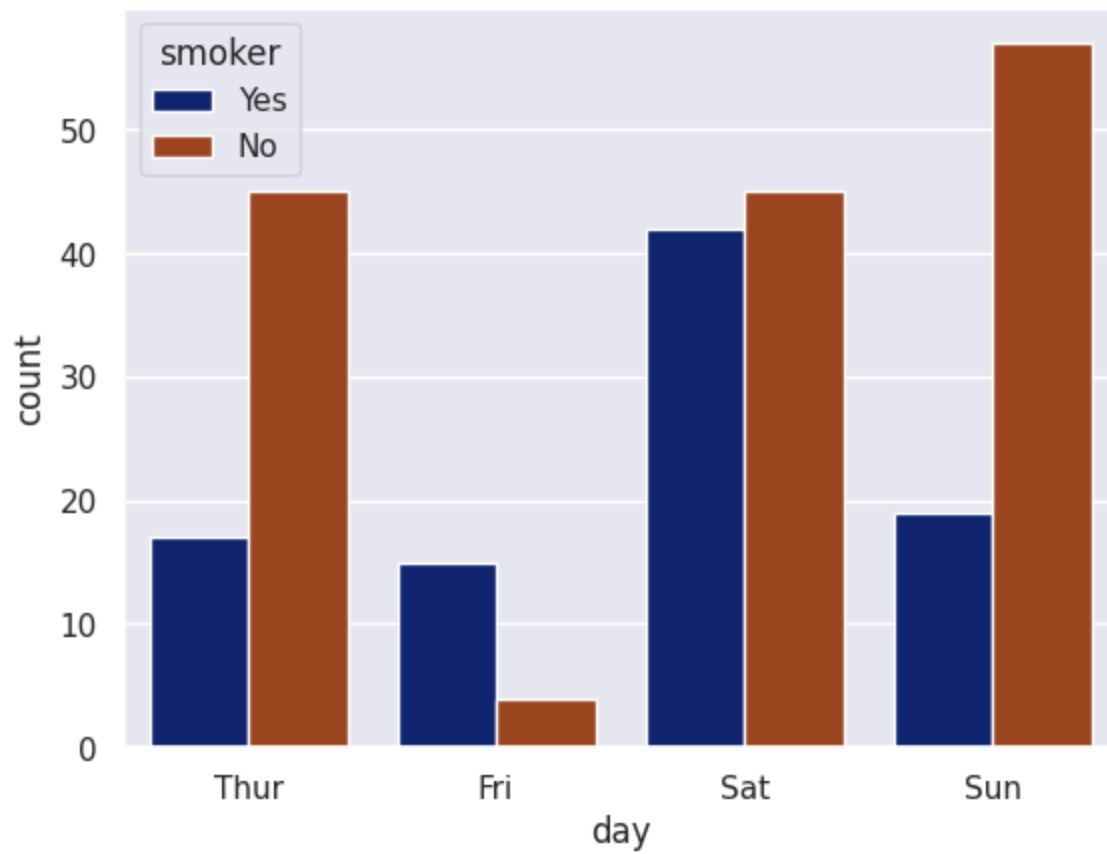


```
In [ ]: # 5. Violin Plot: Total Bill by Day, split by Time

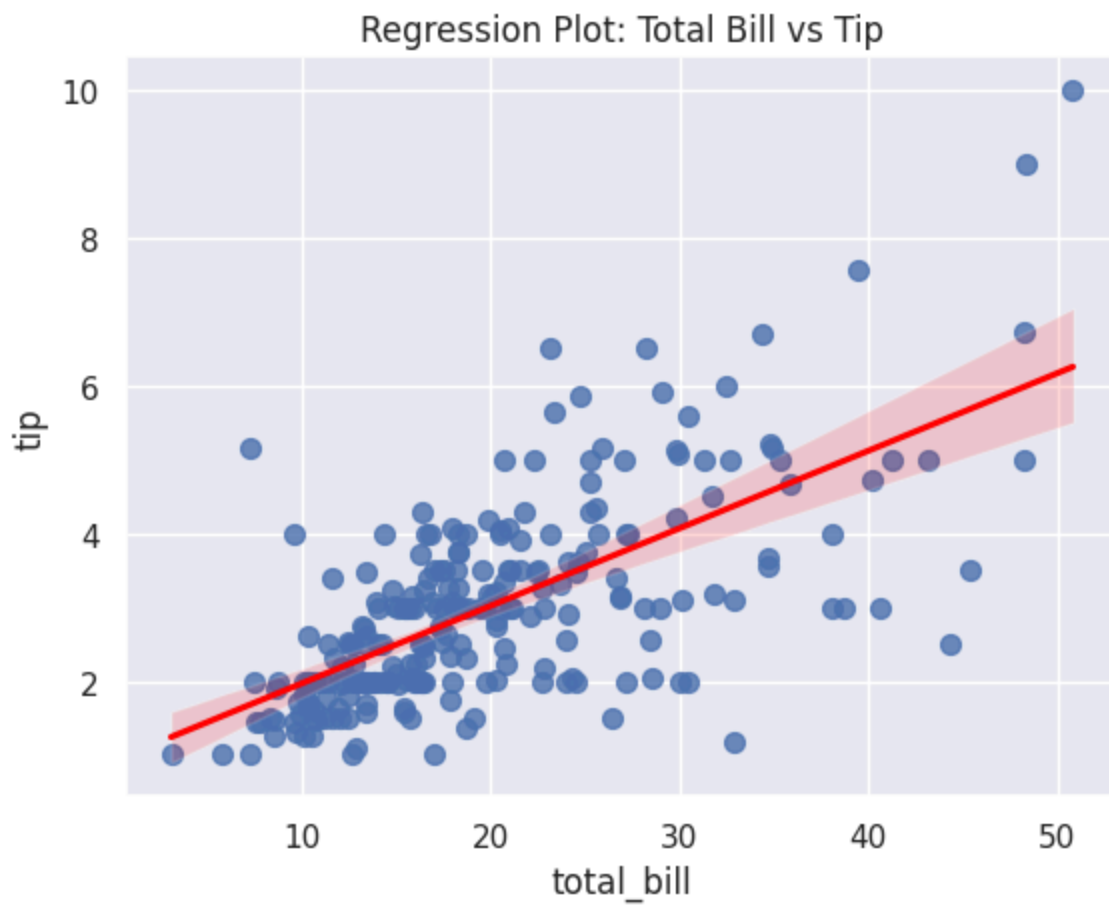
sns.violinplot(data=tips, x="day", y="total_bill", hue="time", split=True, pal
plt.title("Violin Plot: Total Bill by Day and Time")
plt.show()
```



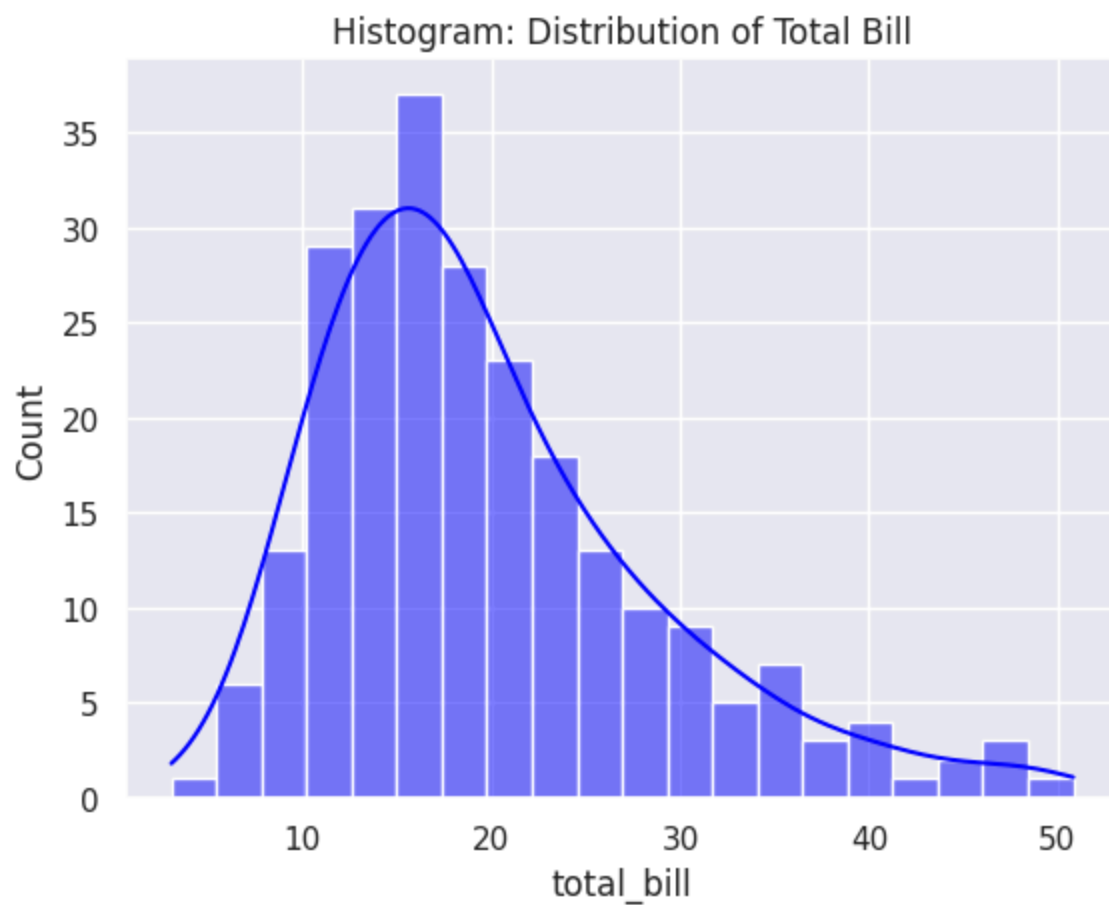
```
In [ ]: sns.countplot(data=tips, x='day', hue='smoker', palette='dark')  
plt.show()
```



```
In [ ]: # 7. Regression Plot: Total Bill vs Tip with regression line
sns.regplot(data=tips, x="total_bill", y="tip", scatter_kws={"s": 50}, line_kws={"color": "red"})
plt.title("Regression Plot: Total Bill vs Tip")
plt.show()
```

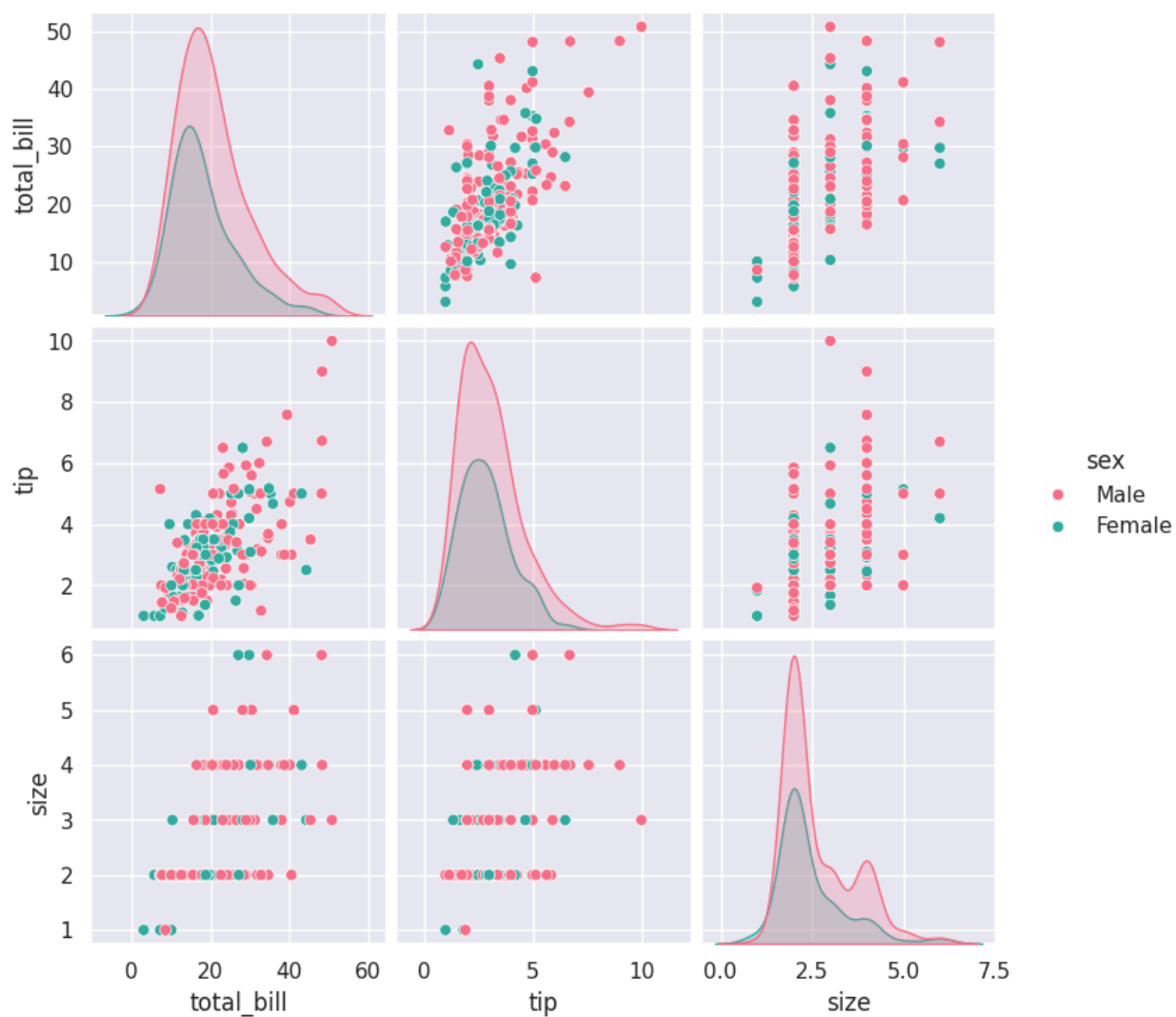
```
In [ ]: #8. Histogram (Distribution Plot): Total Bill with KDE
sns.histplot(data=tips, x="total_bill", kde=True, bins=20, color="blue")
plt.title("Histogram: Distribution of Total Bill")
plt.show()
```



In []: *#9. Pair Plot: Relationships between numerical variables*

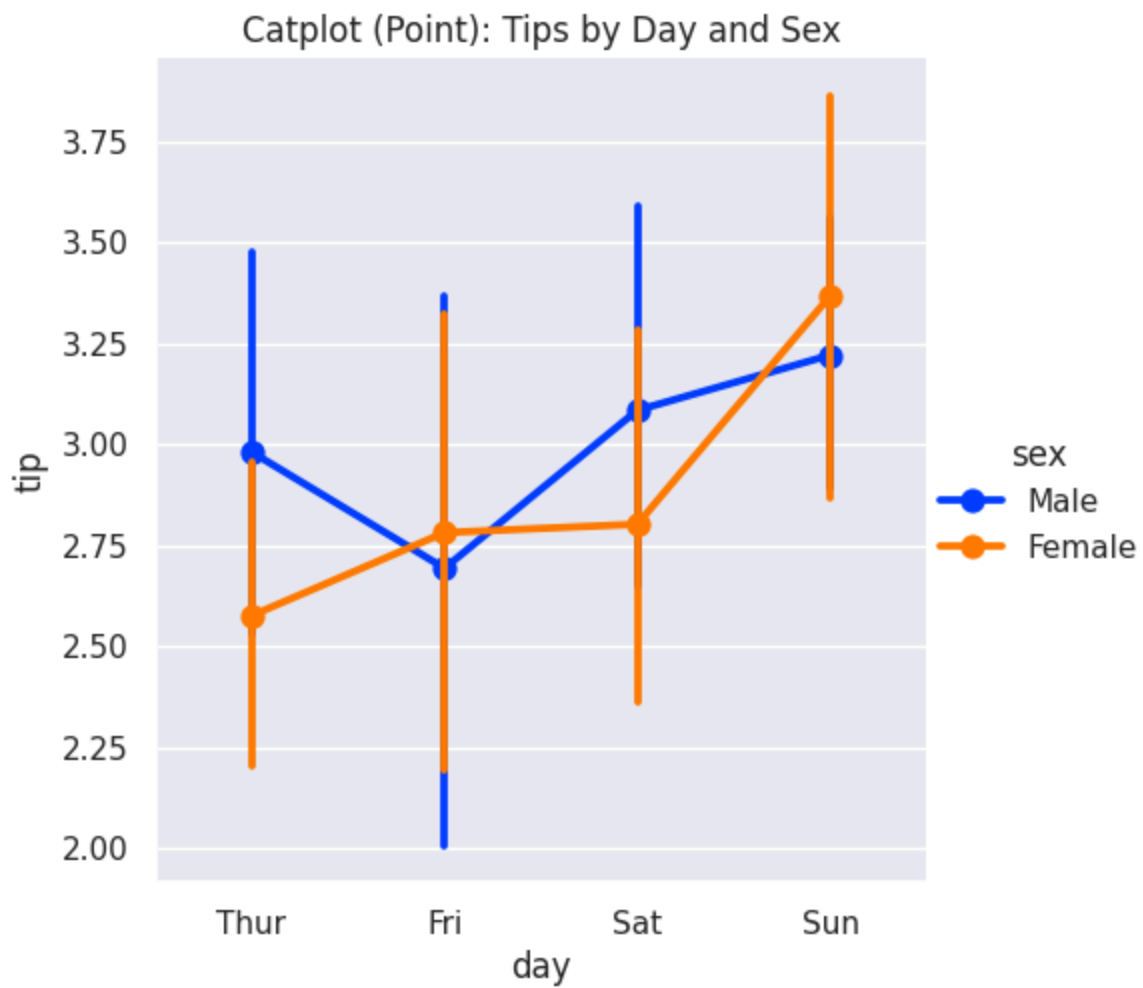
```
sns.pairplot(tips, hue="sex", vars=["total_bill", "tip", "size"], palette="hus  
plt.suptitle("Pair Plot: Numerical Variables by Sex", y=1.02)  
plt.show()
```

Pair Plot: Numerical Variables by Sex



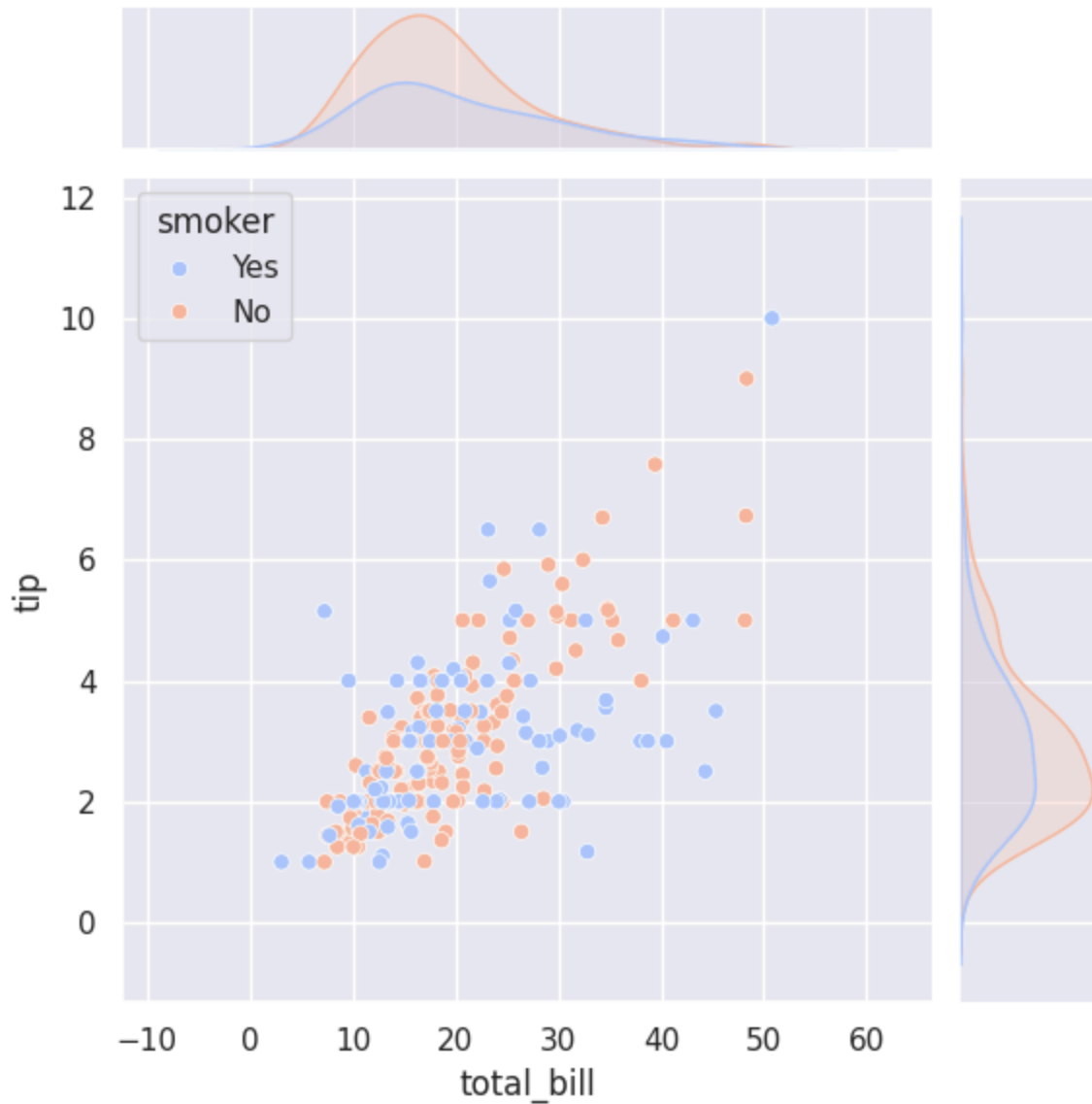
```
In [ ]: # 10. Catplot (Point Plot): Tip by Day and Sex

sns.catplot(data=tips, x="day", y="tip", hue="sex", kind="point", palette="bright")
plt.title("Catplot (Point): Tips by Day and Sex")
plt.show()
```

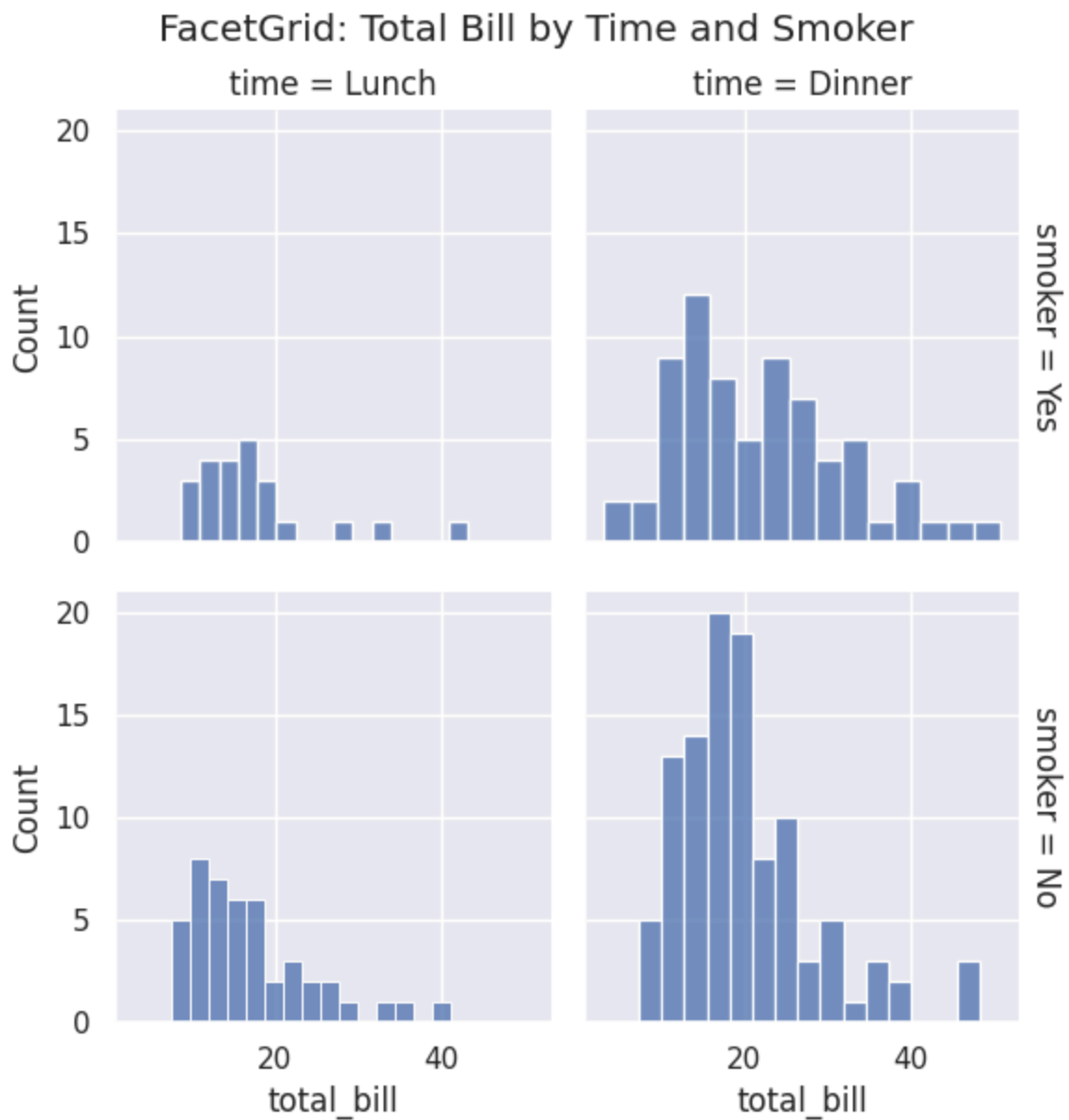


```
In [ ]: # 11. Joint Plot: Total Bill vs Tip with marginal distributions
sns.jointplot(data=tips, x="total_bill", y="tip", kind="scatter", hue="smoker")
plt.suptitle("Joint Plot: Total Bill vs Tip by Smoker", y=1.02)
plt.show()
```

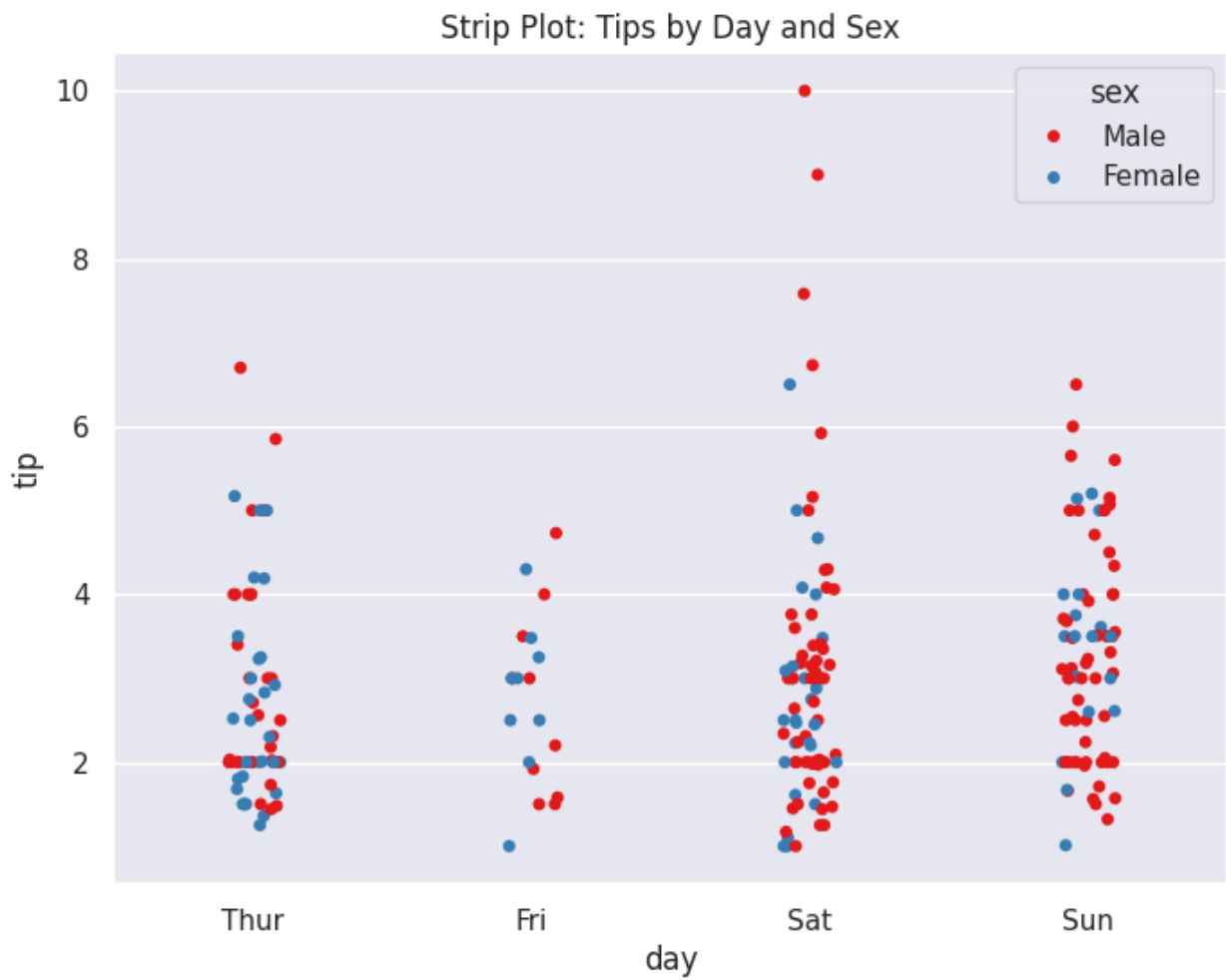
Joint Plot: Total Bill vs Tip by Smoker



```
In [ ]: # 12. FacetGrid: Total Bill by Day, faceted by Time and Smoker
g = sns.FacetGrid(tips, col="time", row="smoker", margin_titles=True)
g.map(sns.histplot, "total_bill", bins=15)
g.fig.suptitle("FacetGrid: Total Bill by Time and Smoker", y=1.02)
plt.show()
```



```
In [ ]: # 13. Strip Plot: Tips by Day, colored by Sex
plt.figure(figsize=(8, 6))
sns.stripplot(data=tips, x="day", y="tip", hue="sex", palette="Set1", jitter=True)
plt.title("Strip Plot: Tips by Day and Sex")
plt.show()
```



```
In [ ]: # 14. KDE Plot: Total Bill density by Sex
plt.figure(figsize=(8, 6))
sns.kdeplot(data=tips, x="total_bill", hue="sex", fill=True, palette="tab10")
plt.title("KDE Plot: Total Bill Density by Sex")
plt.show()
```

