

# Grids and Multi-Plot Layouts

## seaborn.FacetGrid() :

FacetGrid class helps in visualizing distribution of one variable as well as the relationship between multiple variables separately within subsets of your dataset using multiple panels.

A FacetGrid can be drawn with up to three dimensions ? row, col, and hue. The first two have obvious correspondence with the resulting array of axes; think of the hue variable as a third dimension along a depth axis, where different levels are plotted with different colors.

FacetGrid object takes a dataframe as input and the names of the variables that will form the row, column, or hue dimensions of the grid. The variables should be categorical and the data at each level of the variable will be used for a facet along that axis.

seaborn.FacetGrid( data, \*\*kwargs)

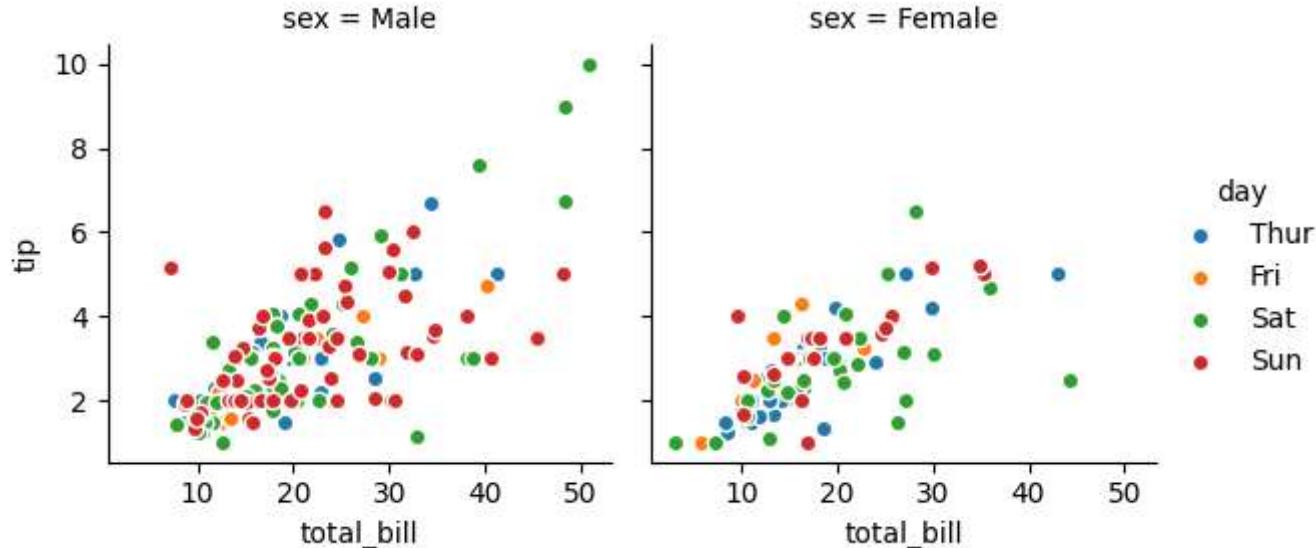
data: Tidy ("long-form") dataframe where each column is a variable and each row is an observation. : DataFrame

row, col, hue: Variables that define subsets of the data, which will be drawn on separate facets in the grid. See the `*_order` parameters to control the order of levels of this variable. : strings

palette : Colors to use for the different levels of the `hue` variable. : palette name, list, or dict, optional

```
In [2]: # importing packages
import seaborn
import matplotlib.pyplot as plt

#Loading of a dataframe from seaborn
df = seaborn.load_dataset('tips')
#Form a facetgrid using columns with a hue
graph = seaborn.FacetGrid(df, col ="sex", hue ="day")
#map the above form facetgrid with some attributes
graph.map(plt.scatter, "total_bill", "tip", edgecolor ="w").add_legend()
# show the object
plt.show()
# hue is used to separate data into different categories using color
```

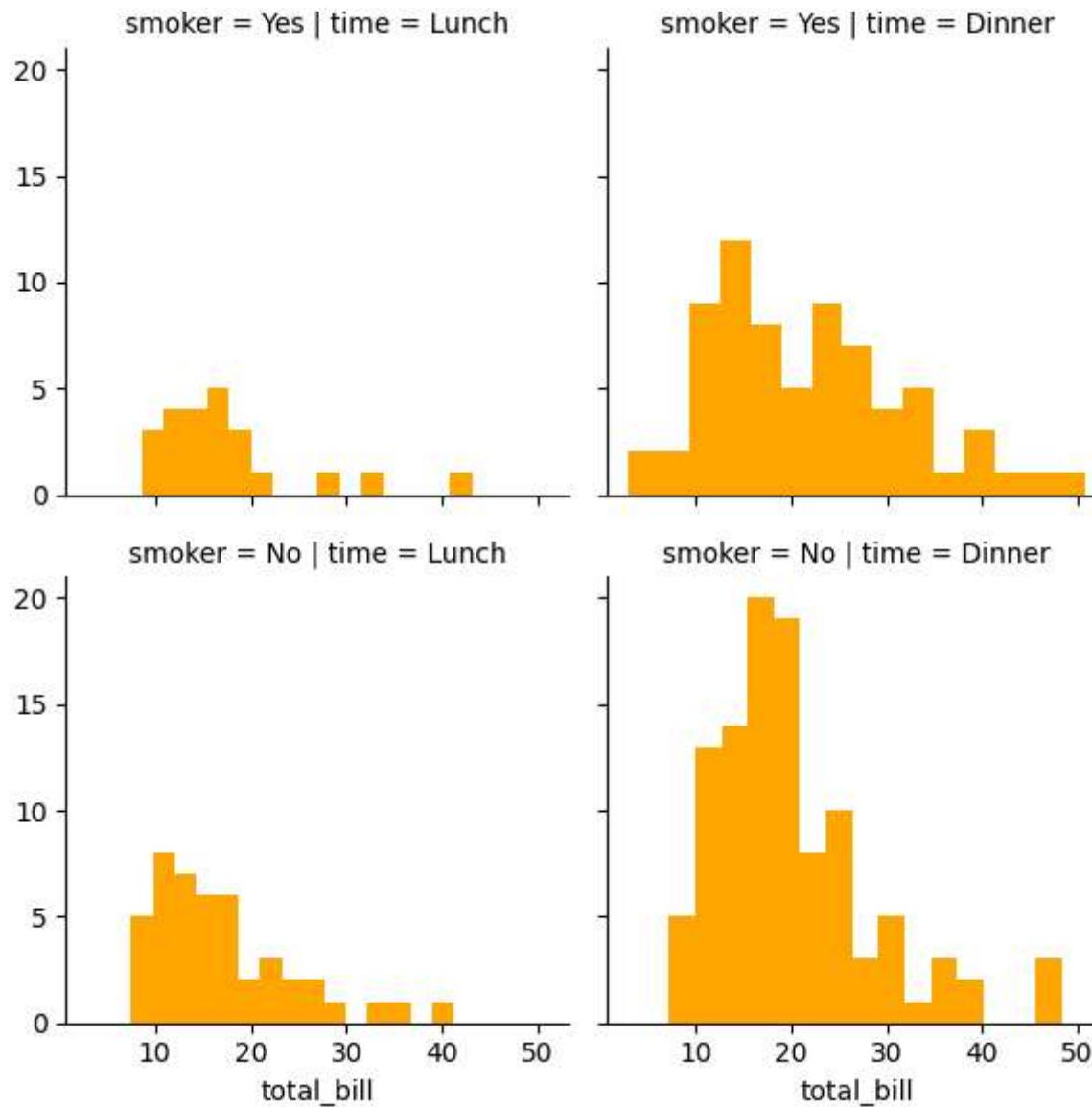


```
In [4]: # importing packages
import seaborn
import matplotlib.pyplot as plt

# Loading of a dataframe from seaborn
df = seaborn.load_dataset('tips')

##### Main Section #####
# Form a facetgrid using columns with a hue
graph = seaborn.FacetGrid(df, row ='smoker', col ='time')
# map the above form facetgrid with some attributes
graph.map(plt.hist, 'total_bill', bins = 15, color ='orange')
# show the object
plt.show()

# This code is contributed by Ratnaprava
```



```
In [7]: # importing packages
import seaborn
import matplotlib.pyplot as plt

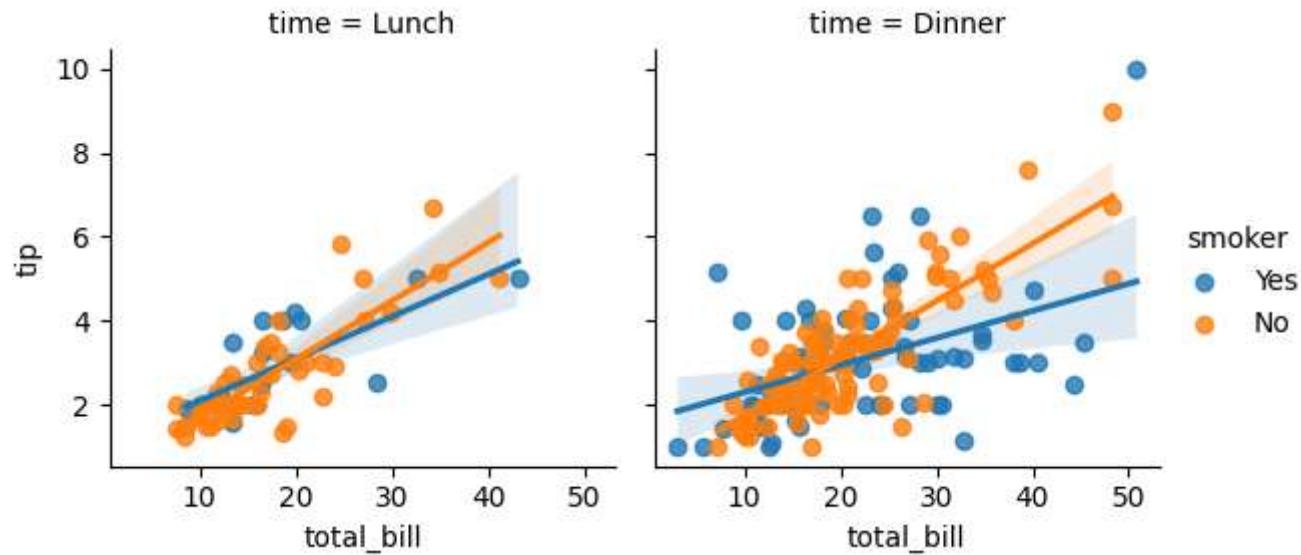
# Loading of a dataframe from seaborn
df = seaborn.load_dataset('tips')
```

```

##### Main Section #####
# Form a facetgrid using columns with a hue
graph = seaborn.FacetGrid(df, col = 'time', hue ='smoker')
# map the above form facetgrid with some attributes
graph.map(seaborn.regplot, "total_bill", "tip").add_legend()
# show the object
plt.show()

# This code is contributed by Ratnaprava
# regplot() is used to plot a scatter plot along with a
# regression line (best-fit line) to show the relationship
# between two numerical variables.

```



`add_legend()` is used to display a legend in Seaborn grids to explain color or style mappings created using `hue`.

**FacetGrid** is used to plot the same relationship across categories, while **PairGrid** is used to visualize pairwise

**relationships among multiple numeric variables.**

## **seaborn.PairGrid() :**

Subplot grid for plotting pairwise relationships in a dataset.

This class maps each variable in a dataset onto a column and row in a grid of multiple axes. Different axes-level plotting functions can be used to draw bivariate plots in the upper and lower triangles, and the marginal distribution of each variable can be shown on the diagonal.

It can also represent an additional level of conditionalization with the hue parameter, which plots different subsets of data in different colors. This uses color to resolve elements on a third dimension, but only draws subsets on top of each other and will not tailor the hue parameter for the specific visualization the way that axes-level functions that accept hue will.

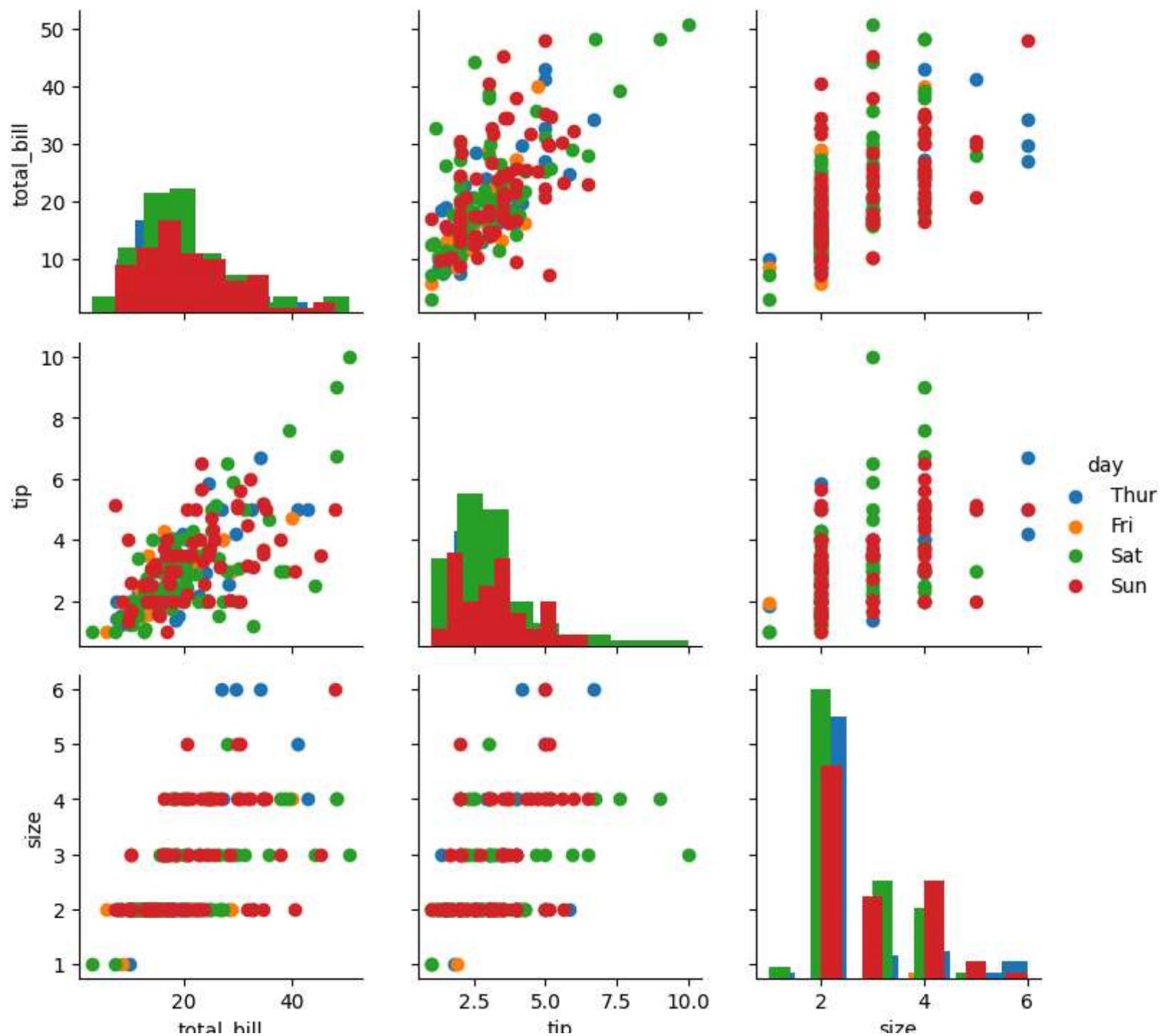
## **seaborn.PairGrid( data, \*\*kwargs)**

Arguments	Description	Value
data	Tidy (long-form) dataframe where each column is a variable and each row is an observation.	DataFrame
hue	Variable in ``data`` to map plot aspects to different colors.	string (variable name), optional
palette	Set of colors for mapping the ``hue`` variable. If a dict, keys should be values in the ``hue`` variable.	dict or seaborn color palette
vars	Variables within ``data`` to use, otherwise use every column with a numeric datatype.	list of variable names, optional
dropna	Drop missing values from the data before plotting.	boolean, optional

```
In [18]: # importing packages
import seaborn
import matplotlib.pyplot as plt

# Loading dataset
df = seaborn.load_dataset('tips')
# PairGrid object with hue
graph = seaborn.PairGrid(df, hue ='day')
# type of graph for diagonal
graph = graph.map_diag(plt.hist) # Uses histogram for each numeric column
# type of graph for non-diagonal
graph = graph.map_offdiag(plt.scatter) # Draws scatter plots for variable pairs
```

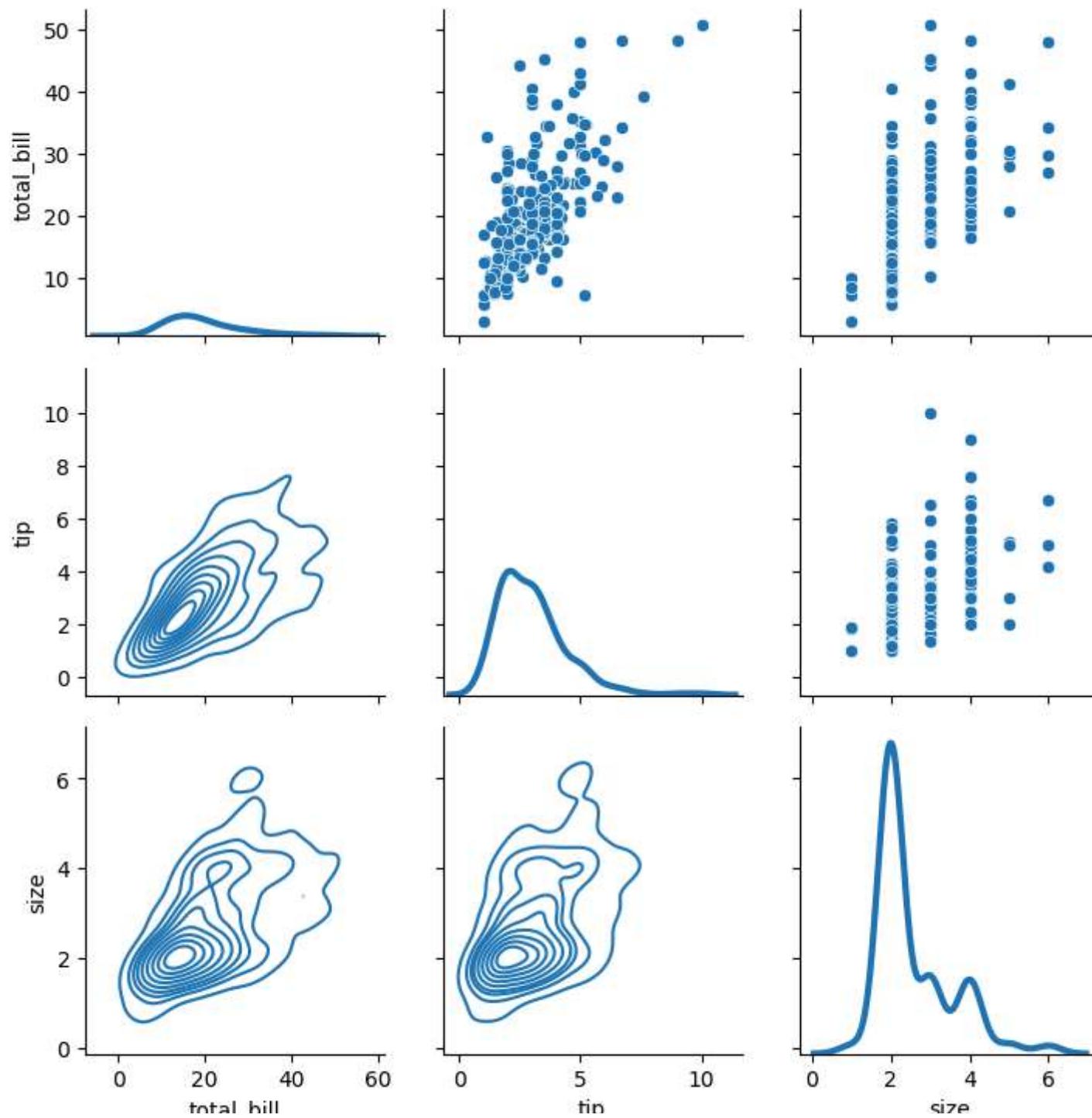
```
# to add legends
graph = graph.add_legend() # Easy interpretation
# to show
plt.show()
# This code is contributed by Ratnaprava
```



```
In [30]: # importing packages
import seaborn as sns
import matplotlib.pyplot as plt

# Loading dataset
df = seaborn.load_dataset('tips')

# PairGrid object with hue
graph = seaborn.PairGrid(df)
# type of graph for non-diagonal(upper part)
graph = graph.map_upper(sns.scatterplot)
# type of graph for non-diagonal(lower part)
graph = graph.map_lower(sns.kdeplot)
# type of graph for diagonal
graph = graph.map_diag(sns.kdeplot, lw = 3)
# to show
plt.show()
# This code is contributed by Ratnaprava
```



Data differentiation → groups

Point differentiation → individual dots

## Relational plots in Seaborn - Part I

Relational plots are used for visualizing the statistical relationship between the data points. Visualization is necessary because it allows the human to see trends and patterns in the data. The process of understanding how the variables in the dataset relate each other and their relationships are termed as Statistical analysis.

Seaborn, unlike to matplotlib, also provides some default datasets. In this article, we will be using a default dataset named 'tips'. This dataset gives information about people who had food at some restaurant and whether they left tip for waiters or not, their gender and whether they do smoke or not, and more.

In [34]:

```
# importing the Library
import seaborn as sns

# reading the dataset
data = sns.load_dataset('tips')

# printing first five entries
print(data.head())
```

	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

To draw the relational plots seaborn provides three functions. These are:

## relplot()

Seaborn.relplot()

This function provides us the access to some other different axes-level functions which shows the relationships between two variables with semantic mappings of subsets.

Syntax :

```
seaborn.relplot(x=None, y=None, data=None, **kwargs)
```

## scatterplot()

## lineplot()

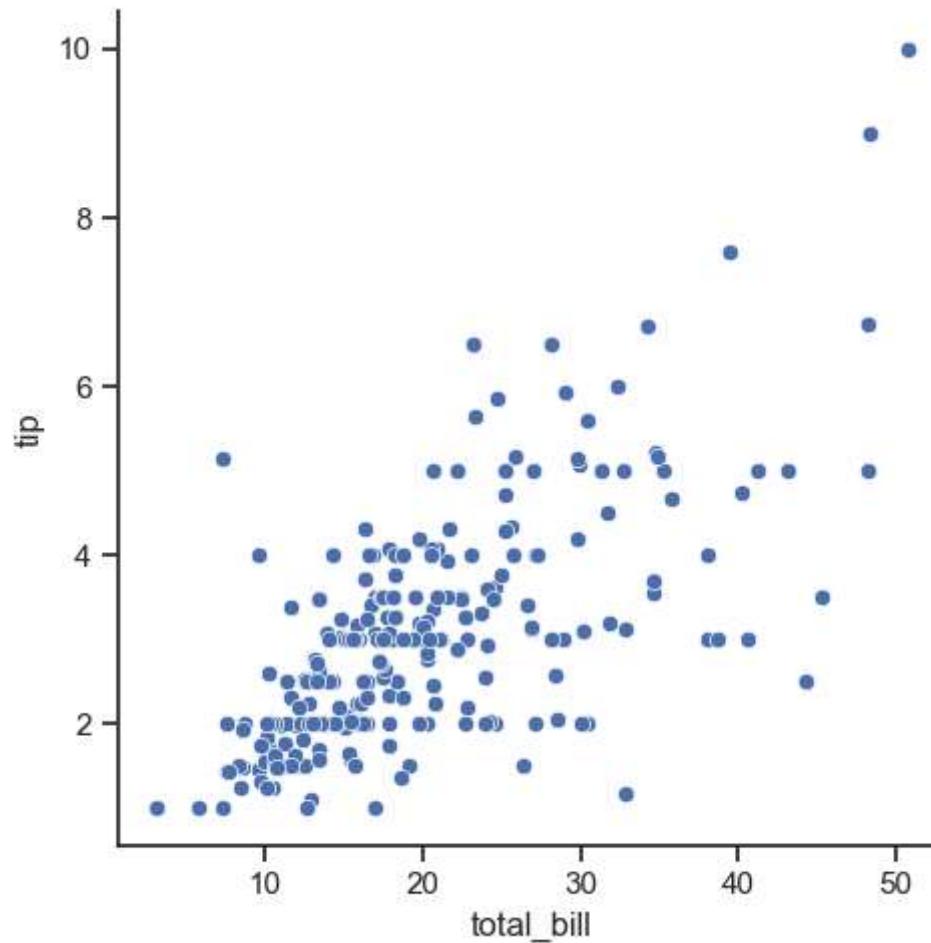
```
In [39]: # importing the Library
import seaborn as sns

# selecting style
sns.set(style ="ticks")

# reading the dataset
tips = sns.load_dataset('tips')

# plotting a simple visualization of data points
sns.relplot(x ="total_bill", y ="tip", data = tips)
```

```
Out[39]: <seaborn.axisgrid.FacetGrid at 0x1b6ab5341a0>
```



```
In [41]: # importing the library
import seaborn as sns

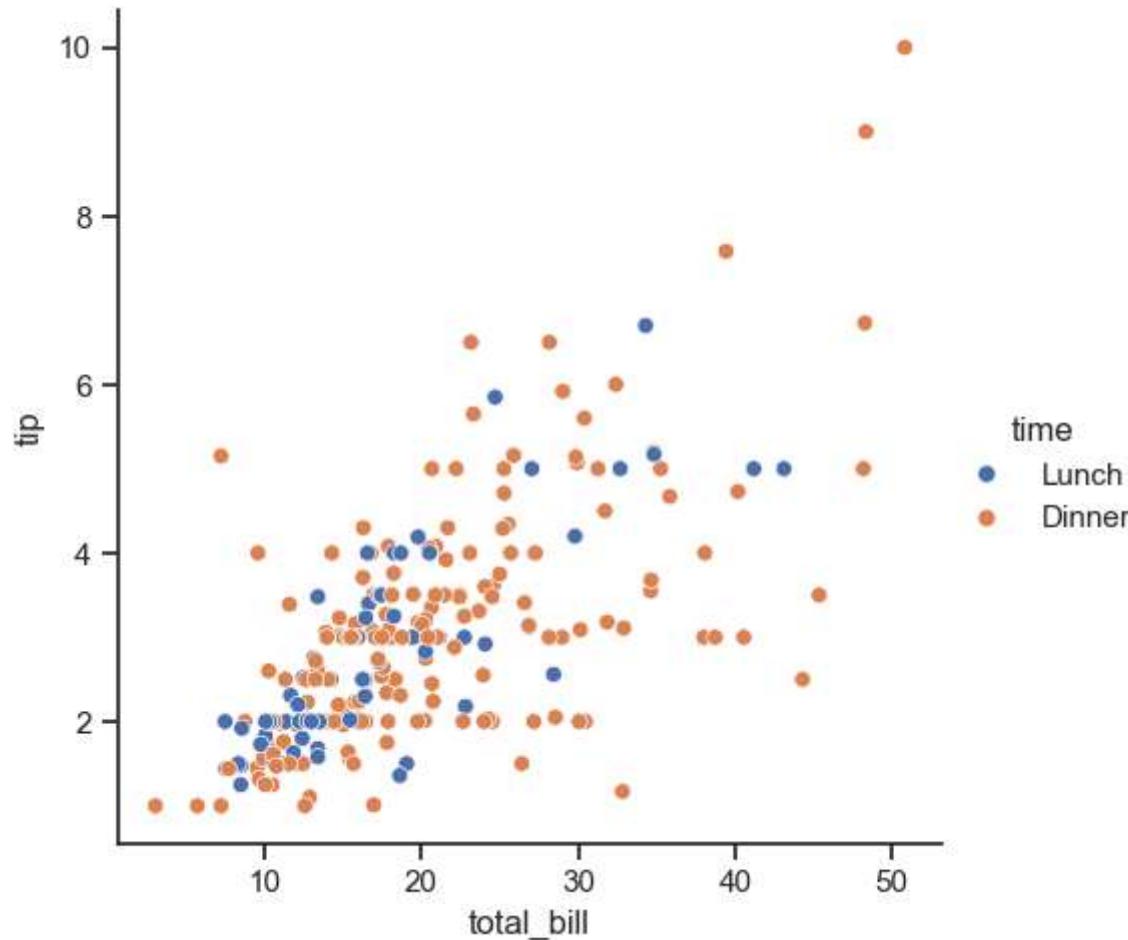
# selecting style
sns.set(style = "ticks")

# reading the dataset
tips = sns.load_dataset('tips')

sns.relplot(x="total_bill",
            y="tip",
```

```
hue="time",  
data=tips)
```

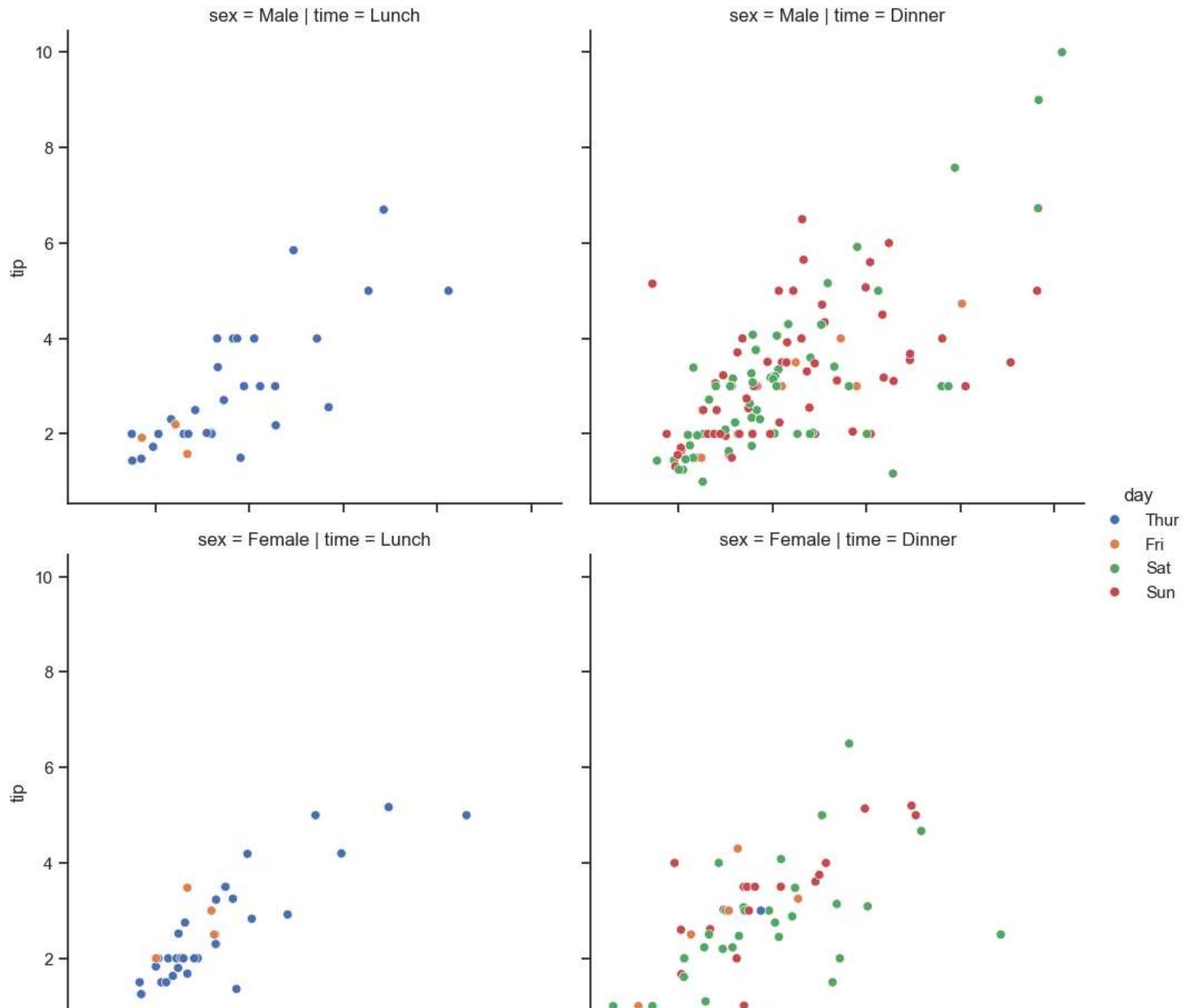
Out[41]: <seaborn.axisgrid.FacetGrid at 0x1b6ab0a6b10>

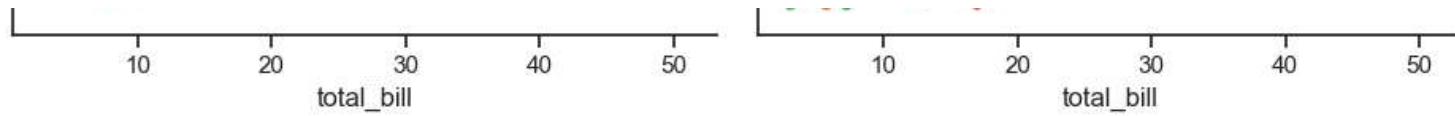


```
# importing the Library  
import seaborn as sns  
  
# selecting style  
sns.set(style ="ticks")  
  
# reading the dataset  
tips = sns.load_dataset('tips')
```

```
sns.relplot(x="total_bill",
             y="tip",
             hue="day",
             col="time",
             row="sex",
             data=tips)
```

Out[43]: <seaborn.axisgrid.FacetGrid at 0x1b6a9559e20>





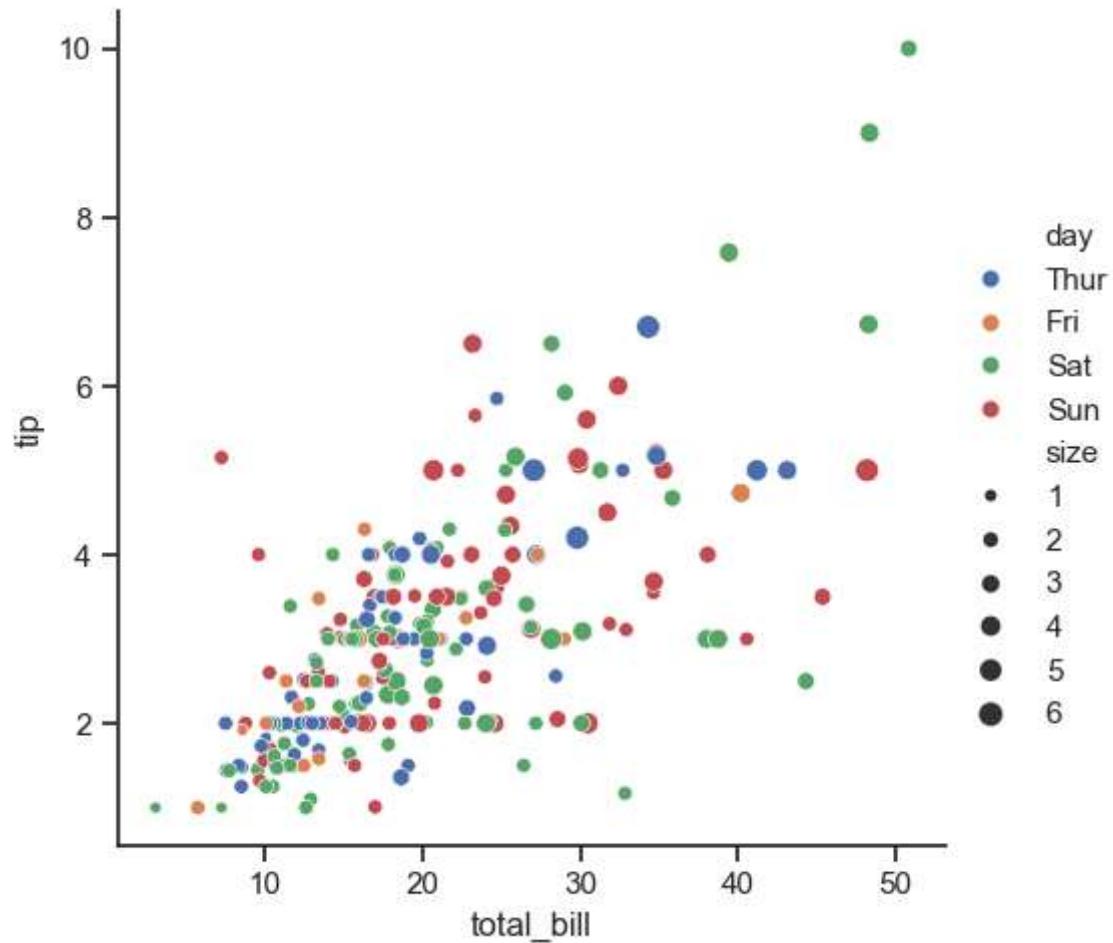
```
In [45]: # importing the library
import seaborn as sns

# selecting style
sns.set(style ="ticks")

# reading the dataset
tips = sns.load_dataset('tips')

sns.relplot(x="total_bill",
            y="tip",
            hue="day",
            size="size",
            data=tips)
```

```
Out[45]: <seaborn.axisgrid.FacetGrid at 0x1b68a2e96d0>
```



In [ ]: