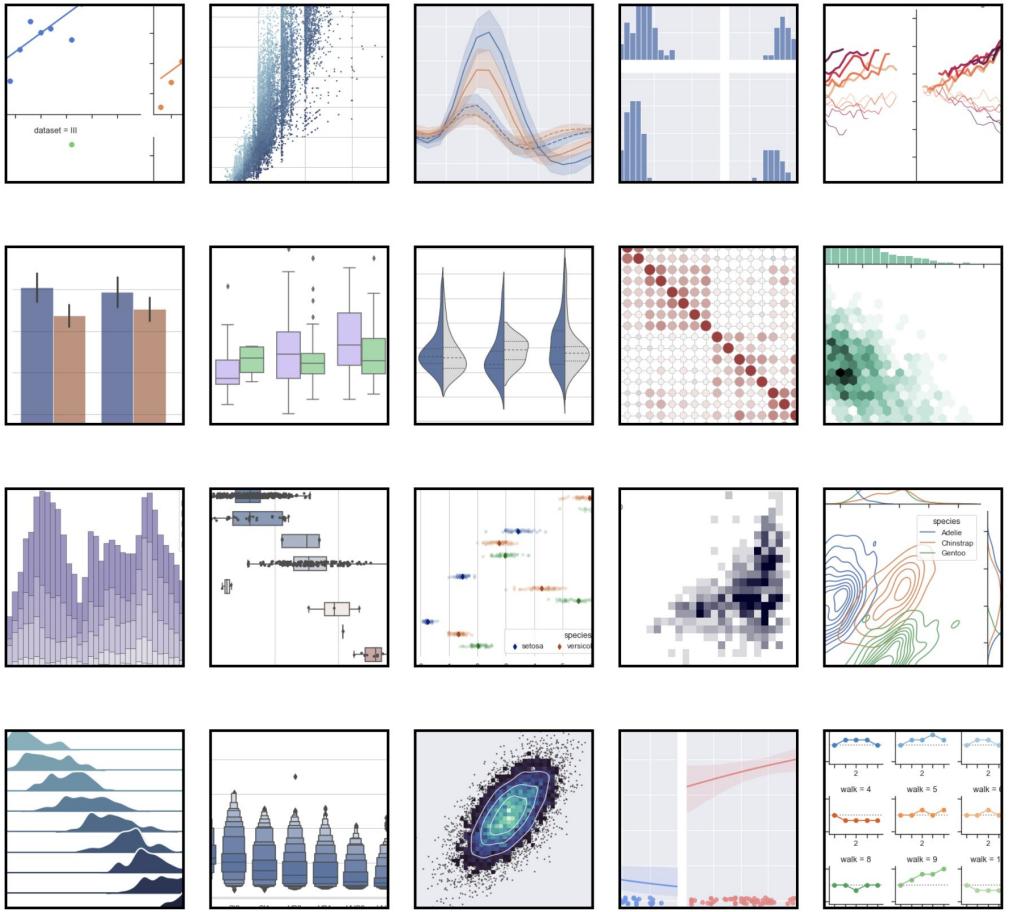
HOCHSCHULE
DER MEDIEN


# Introduction

Prof. Dr. Jan Kirenz
HdM Stuttgart

seaborn helps you
**explore** and understand
your **data**

# seaborn


dataset = III
species
Adelie
Chinstrap
Gentoo
setosa
walk = 4
walk = 5
walk = 8
walk = 9

# Seaborn setup in Anaconda

1) Check your version of seaborn in your terminal:

   **conda list seaborn**

   The newest version is 0.11.0.2.

2) If your are below 0.11.0, first try

   **conda update --all**

3) Again, check the version

   **conda list seaborn**

4) If you are still below 0.11 try:

   **conda install seaborn=0.11.2**

# We use Jupyter Notebooks with this Setup

Magic command:

**%matplotlib inline**

# **Import** the library seaborn as **sns**

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- Seaborn is the only library we need to **import**
- By convention, it is imported with the shorthand **sns**.

Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# Apply the default **theme**

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- If you have a seborn version below 11. use: **sns.set()**
- This will affect your plot look
- There are different seaborn **themes** like darkgrid, whitegrid, dark, white, and ticks.

Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# We load the example **dataset** tips

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- Most examples use **pandas dataframes** (tabular format like spreadsheets)
- Seaborne can use many data structures
- Dataset tips:

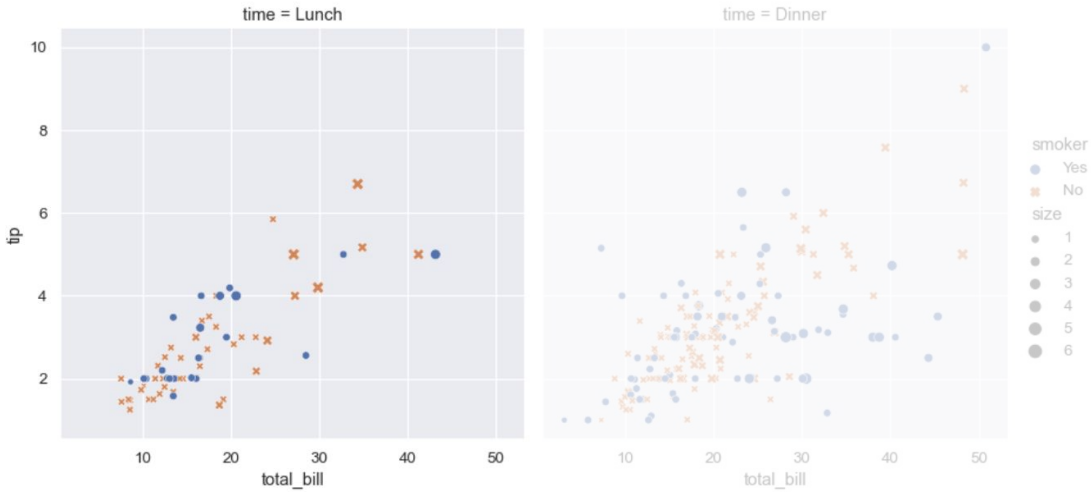
| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |

Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# The **relplot** function plots relationships

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- **relplot** shows the **relationship** between two variables (total_bill and tip)

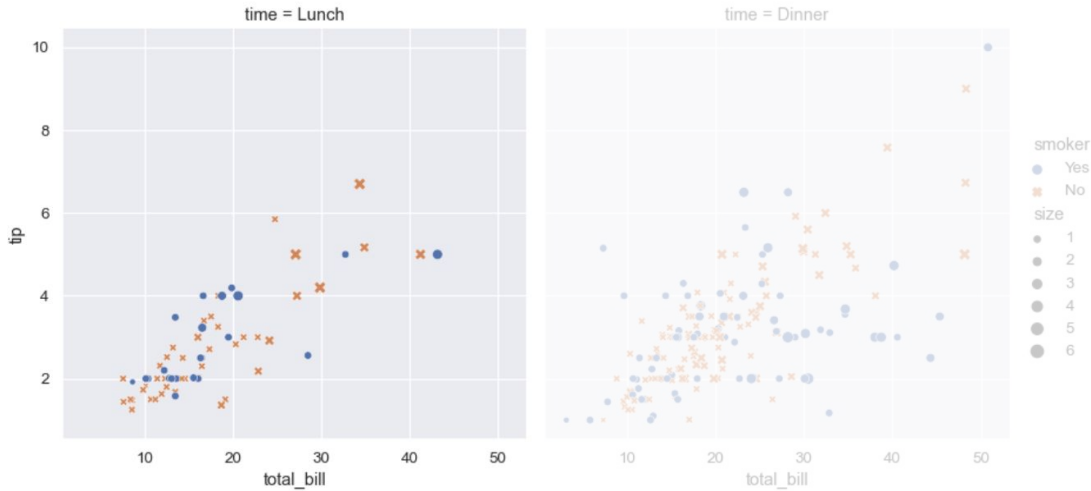



Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# Provide the name of the **dataset**

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- Usually this is the name of your pandas dataframe object




Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# Provide the name of the **x** and **y** variables

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

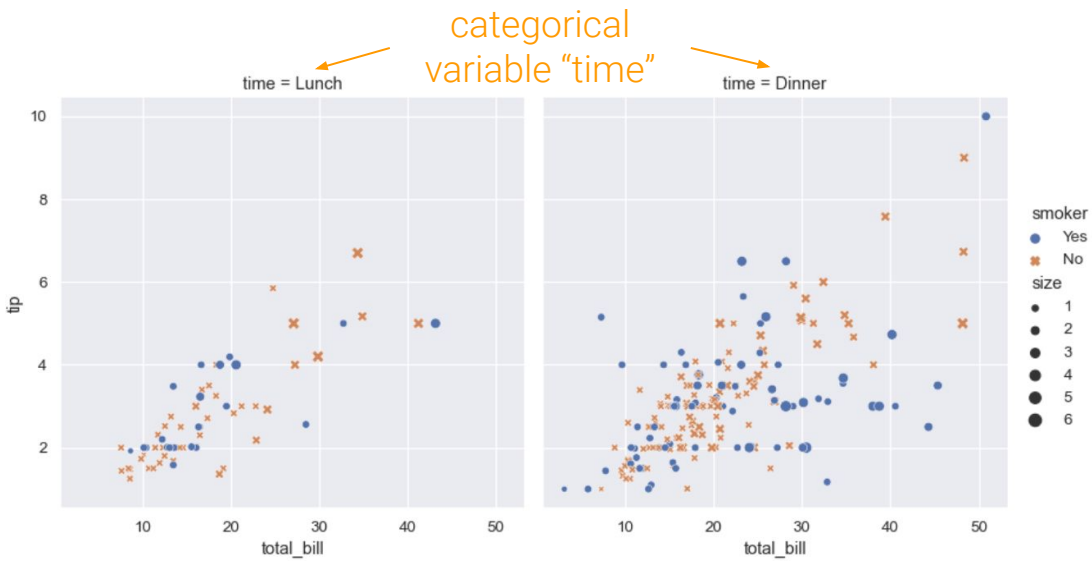
- These are the main variables for our plot: x = total_bill and y = tip


Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# Use **col** to create multiple plots next to each other

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- We can use **col** (column) to include a categorical variable with different conditions



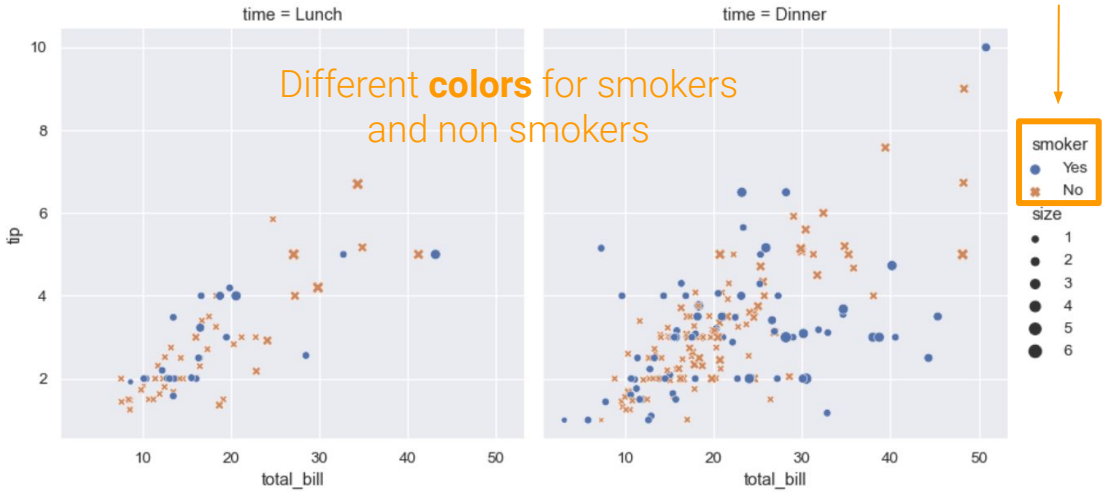

Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# **hue** uses colour encoding

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- Assigning a variable to **hue** will map its levels to the color of the points.

variable "smoker"




Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# **style** changes the markers

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

- Assigning the same variable to **style** will also vary the **markers** and create a more accessible plot (you can also use a new variable)

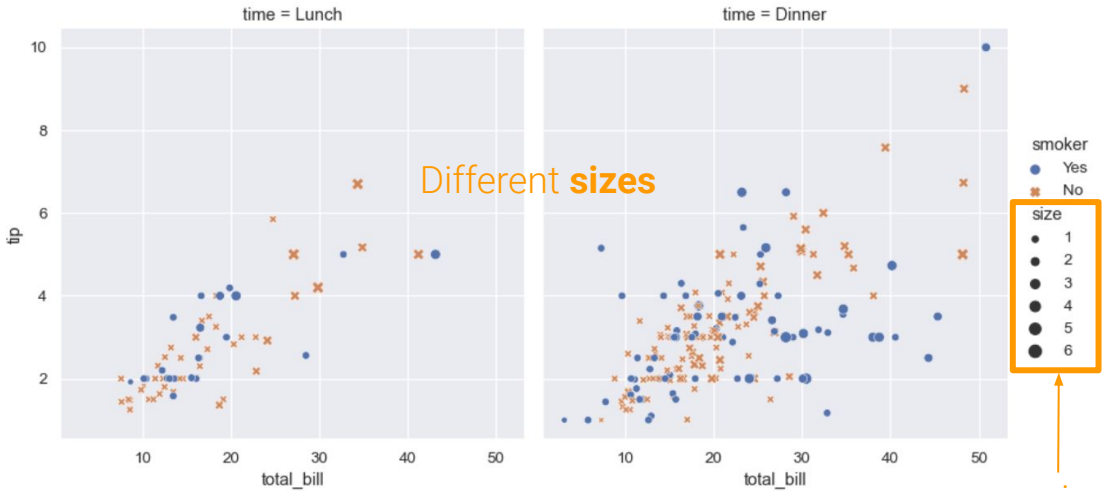
variable "smoker"


Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# **size** changes the size of our markers

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

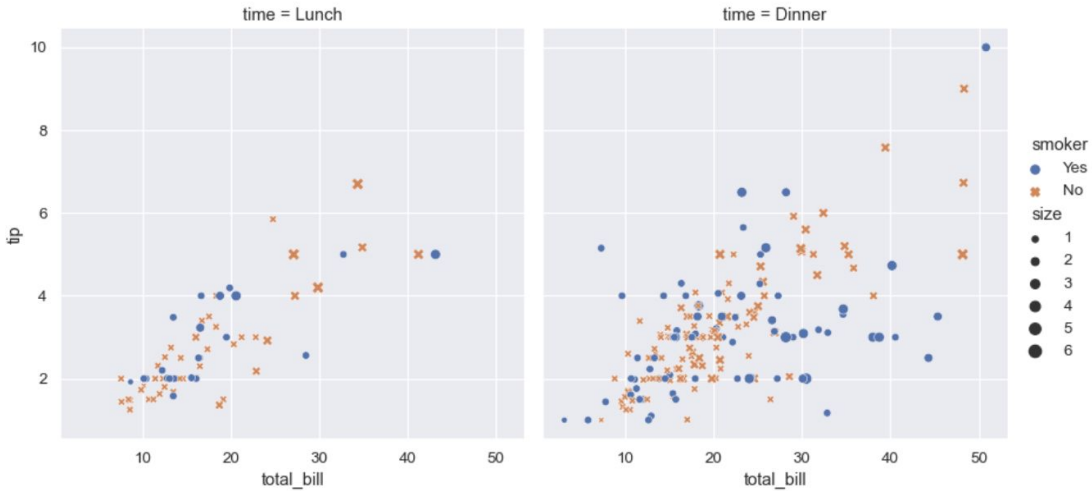
- **Size** uses numerical data to present the observations in different sizes




Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# There are five variables in one plot!

```python
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker" style="smoker", size="size",
)
```

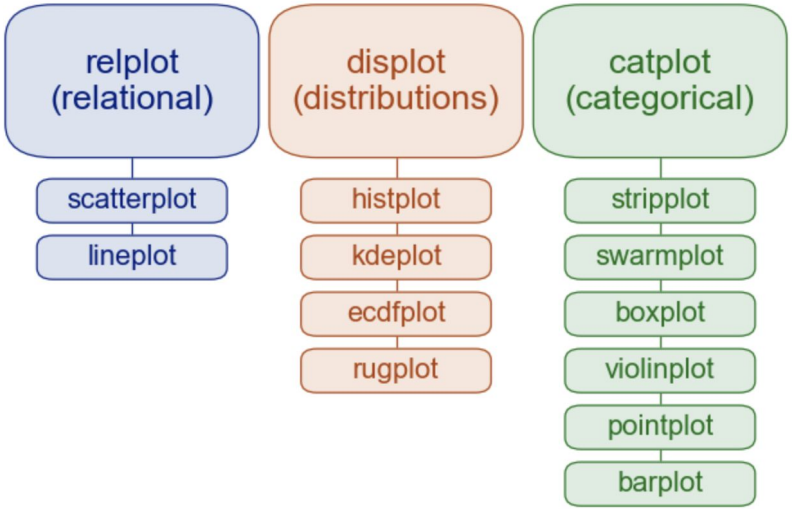



Source: Seaborn documentation http://seaborn.pydata.org/introduction.html

# Similar functions
# for similar tasks

# Similar functions for similar tasks

- Relational plots
- Distribution plots
- Categorical plots
- Regression plots
- Matrix plots
- Multi-plot grids
  - Facet grids (conditional relationships)
  - Pair grids (pairwise relationships)
  - Joint grids




Source: Seaborn documentation https://seaborn.pydata.org/tutorial/function_overview.htm

# Relational plots

# Relational plots

| | |
|---|---|
| `relplot` | Figure-level interface for drawing relational plots onto a FacetGrid. |
| `scatterplot` | Draw a scatter plot with possibility of several semantic groupings. |
| `lineplot` | Draw a line plot with possibility of several semantic groupings. |

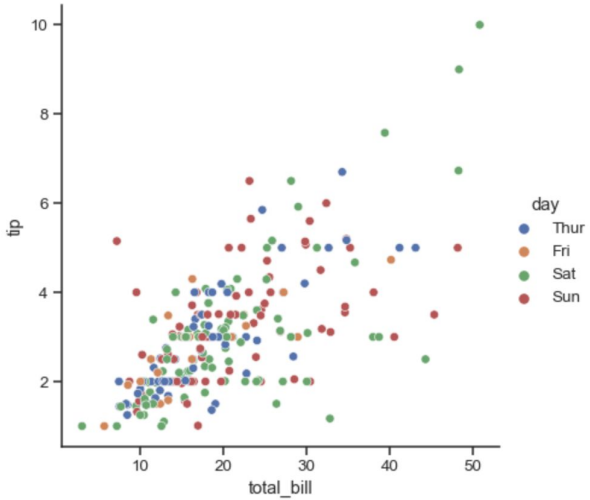
Source: Seaborn documentation: https://seaborn.pydata.org/index.html

# Relational plots

| | |
|---|---|
| `relplot` | Figure-level interface for drawing relational plots onto a FacetGrid. |
| `scatterplot` | Draw a scatter plot with possibility of several semantic groupings. |
| `lineplot` | Draw a line plot with possibility of several semantic groupings. |

| | 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 |

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



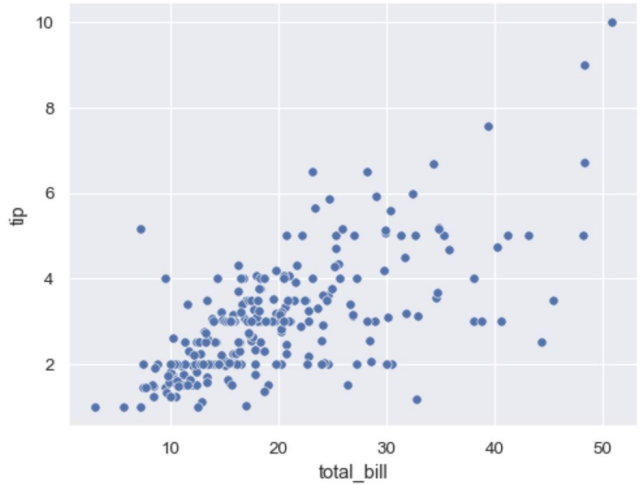

Source: Seaborn documentation: https://seaborn.pydata.org/index.html

# Relational plots

| | |
|---|---|
| `relplot` | Figure-level interface for drawing relational plots onto a FacetGrid. |
| `scatterplot` | Draw a scatter plot with possibility of several semantic groupings. |
| `lineplot` | Draw a line plot with possibility of several semantic groupings. |

| | 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 |

```
sns.scatterplot(data=tips, x="total_bill", y="tip")
```




Source: Seaborn documentation: https://seaborn.pydata.org/index.html

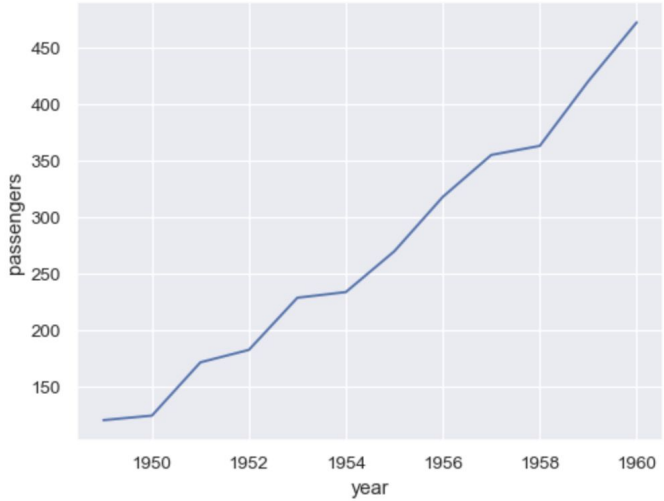
# Relational plots

| | |
|---|---|
| `relplot` | Figure-level interface for drawing relational plots onto a FacetGrid. |
| `scatterplot` | Draw a scatter plot with possibility of several semantic groupings. |
| `lineplot` | Draw a line plot with possibility of several semantic groupings. |

| | year | month | passengers |
|---|---|---|---|
| **0** | 1949 | Jan | 112 |
| **1** | 1949 | Feb | 118 |
| **2** | 1949 | Mar | 132 |
| **3** | 1949 | Apr | 129 |
| **4** | 1949 | May | 121 |

To draw a line plot using long-form data, assign the `x` and `y` variables:

```
may_flights = flights.query("month == 'May'")
sns.lineplot(data=may_flights, x="year", y="passengers")
```




Source: Seaborn documentation: https://seaborn.pydata.org/index.html

# Relational plots

| | |
|---|---|
| `relplot` | Figure-level interface for drawing relational plots onto a FacetGrid. |
| `scatterplot` | Draw a scatter plot with possibility of several semantic groupings. |
| **`lineplot`** | Draw a line plot with possibility of several semantic groupings. |

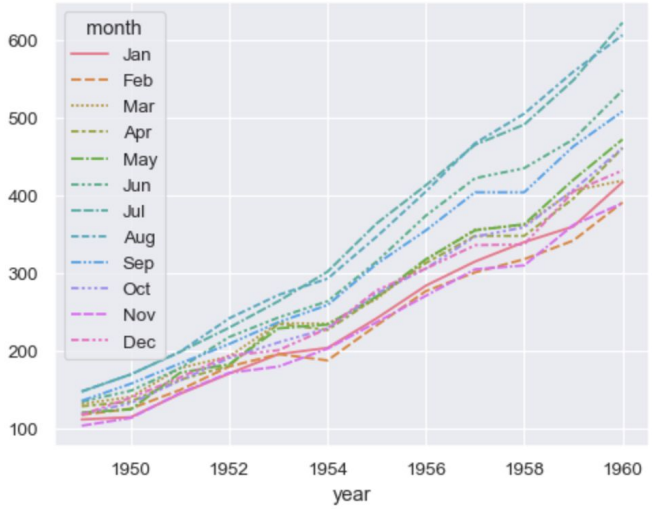
Pivot the dataframe to a wide-form representation:

```
flights_wide = flights.pivot("year", "month", "passengers")
flights_wide.head()
```

| month<br>year | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1949** | 112 | 118 | 132 | 129 | 121 | 135 | 148 | 148 | 136 | 119 | 104 | 118 |
| **1950** | 115 | 126 | 141 | 135 | 125 | 149 | 170 | 170 | 158 | 133 | 114 | 140 |
| **1951** | 145 | 150 | 178 | 163 | 172 | 178 | 199 | 199 | 184 | 162 | 146 | 166 |
| **1952** | 171 | 180 | 193 | 181 | 183 | 218 | 230 | 242 | 209 | 191 | 172 | 194 |
| **1953** | 196 | 196 | 236 | 235 | 229 | 243 | 264 | 272 | 237 | 211 | 180 | 201 |

Passing the entire wide-form dataset to `data` plots a separate line for each column:

```
sns.lineplot(data=flights_wide)
```




Source: Seaborn documentation: https://seaborn.pydata.org/index.html

# Distribution plots

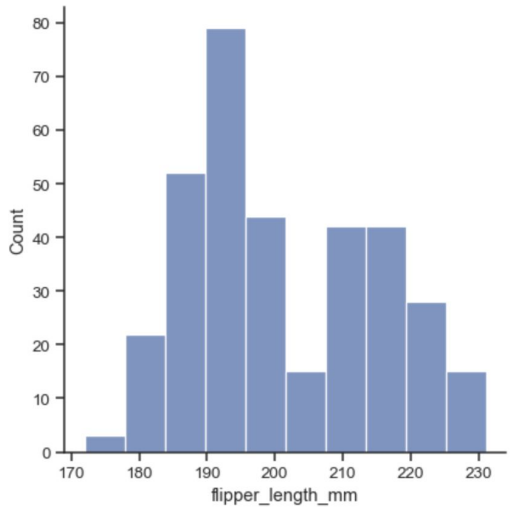
# Distribution plots

| | |
|---|---|
| `displot` | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| `histplot` | Plot univariate or bivariate histograms to show distributions of datasets. |
| `kdeplot` | Plot univariate or bivariate distributions using kernel density estimation. |
| `ecdfplot` | Plot empirical cumulative distribution functions. |
| `rugplot` | Plot marginal distributions by drawing ticks along the x and y axes. |
| `distplot` | DEPRECATED: Flexibly plot a univariate distribution of observations. |

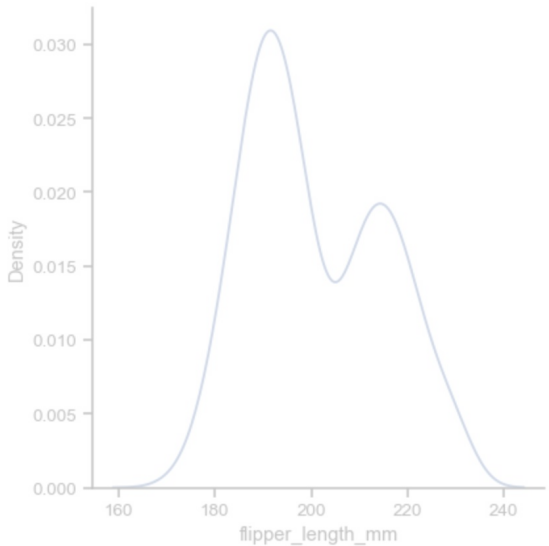
# Distribution plots

| | |
|---|---|
| **displot** | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| **histplot** | Plot univariate or bivariate histograms to show distributions of datasets. |
| **kdeplot** | Plot univariate or bivariate distributions using kernel density estimation. |

```
penguins = sns.load_dataset("penguins")
sns.displot(data=penguins, x="flipper_length_mm")
```




```
sns.displot(data=penguins, x="flipper_length_mm", kind="kde")
```

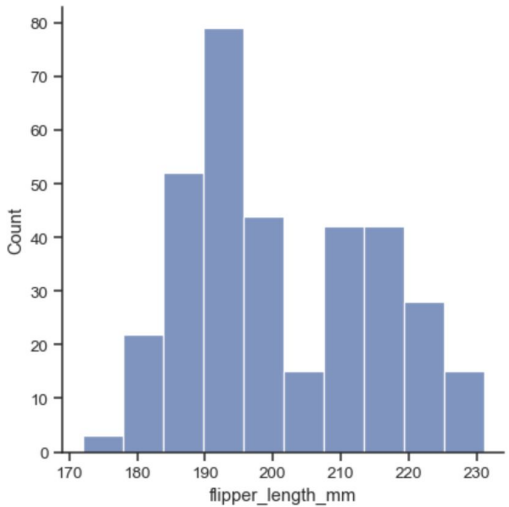



```
sns.displot(data=penguins, x="flipper_length_mm", kde=True)
```

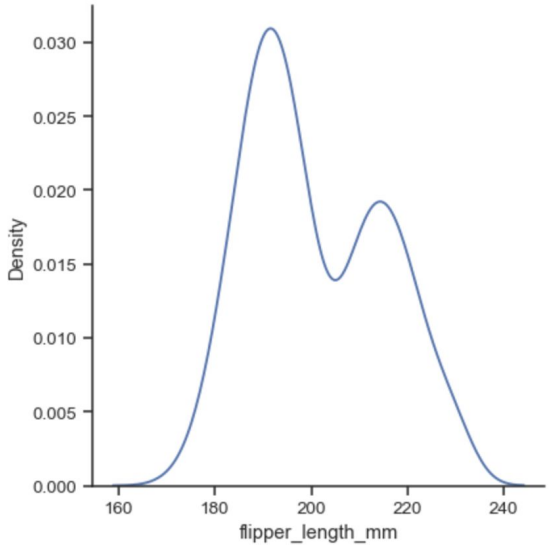
# Distribution plots

| | |
|---|---|
| **displot** | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| **histplot** | Plot univariate or bivariate histograms to show distributions of datasets. |
| **kdeplot** | Plot univariate or bivariate distributions using kernel density estimation. |

```
penguins = sns.load_dataset("penguins")
sns.displot(data=penguins, x="flipper_length_mm")
```




```
sns.displot(data=penguins, x="flipper_length_mm", kind="kde")
```

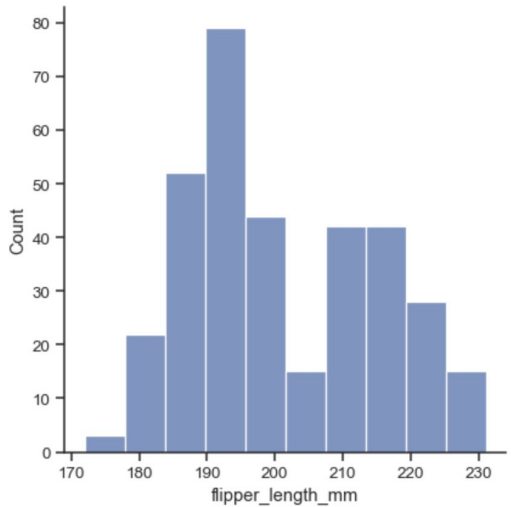



```
sns.displot(data=penguins, x="flipper_length_mm", kde=True)
```

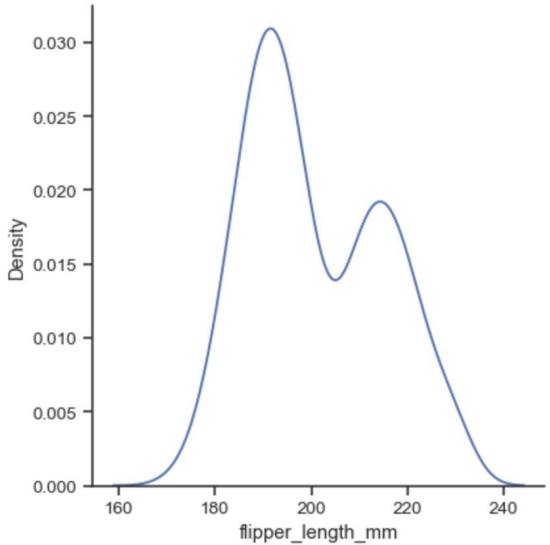
# Distribution plots

| | |
|---|---|
| **displot** | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| **histplot** | Plot univariate or bivariate histograms to show distributions of datasets. |
| **kdeplot** | Plot univariate or bivariate distributions using kernel density estimation. |

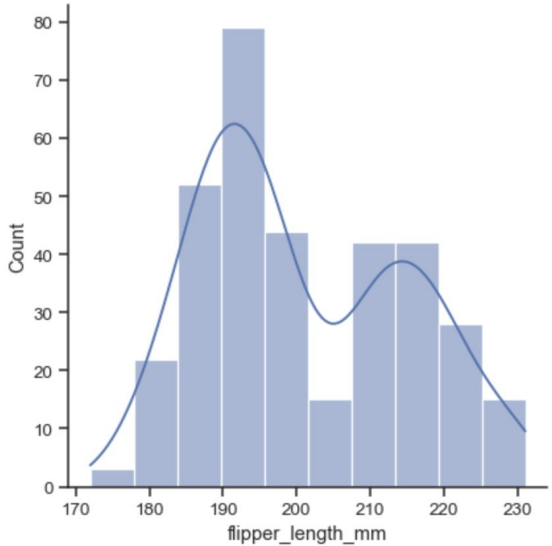
```
penguins = sns.load_dataset("penguins")
sns.displot(data=penguins, x="flipper_length_mm")
```




```
sns.displot(data=penguins, x="flipper_length_mm", kind="kde")
```



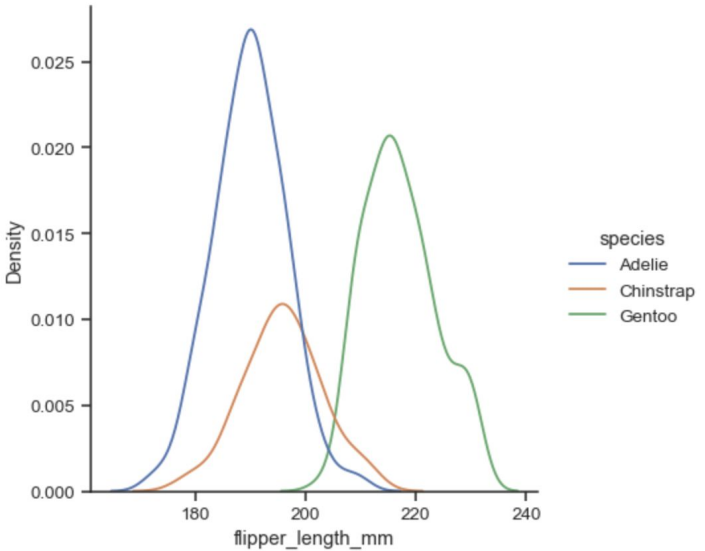

```
sns.displot(data=penguins, x="flipper_length_mm", kde=True)
```



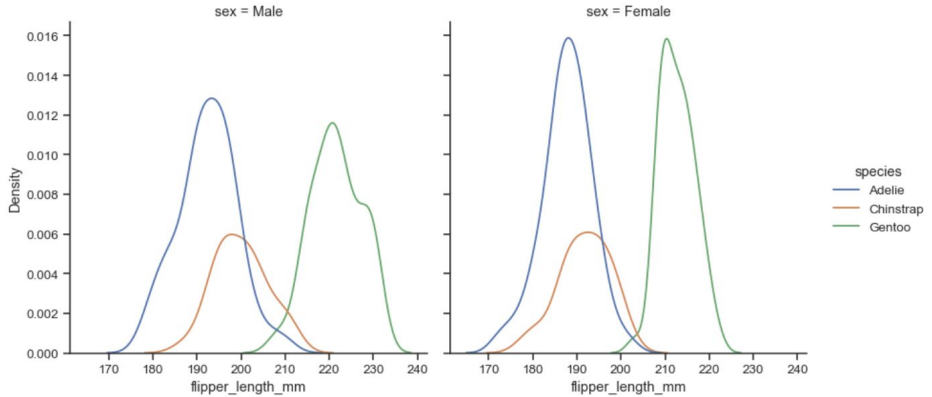
# Distribution plots

| | |
|---|---|
| **displot** | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| **histplot** | Plot univariate or bivariate histograms to show distributions of datasets. |

```
sns.displot(data=penguins, x="flipper_length_mm", hue="species", kind="kde")
```



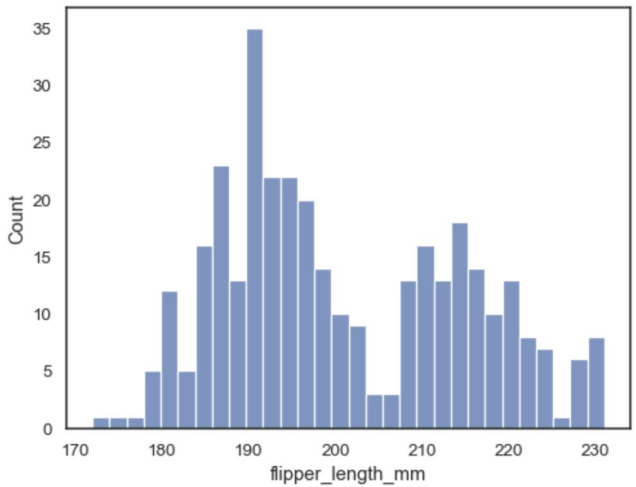

```
sns.displot(data=penguins, x="flipper_length_mm", hue="species", col="sex", kind="kde")
```



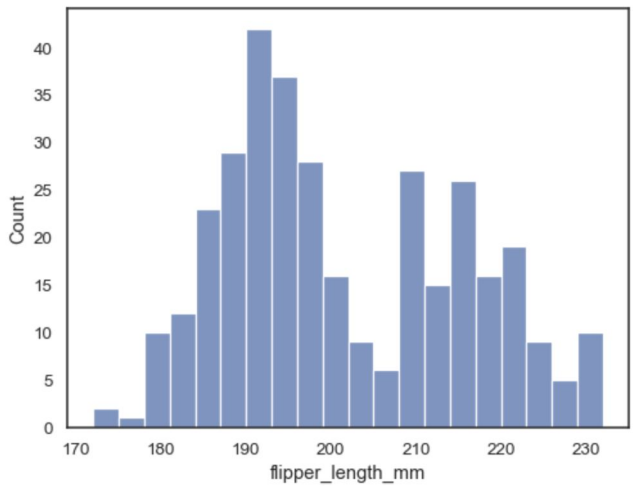
# Distribution plots

| | |
|---|---|
| `displot` | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| `histplot` | Plot univariate or bivariate histograms to show distributions of datasets. |
| `kdeplot` | Plot univariate or bivariate distributions using kernel density estimation. |

```
sns.histplot(data=penguins, x="flipper_length_mm", bins=30)
```



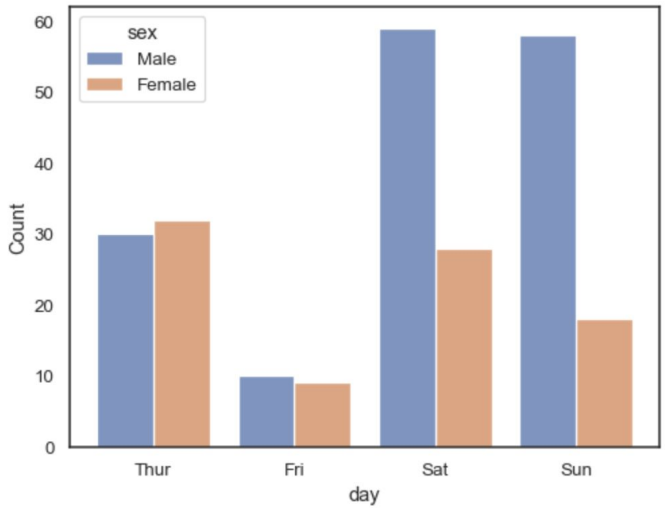

```
sns.histplot(data=penguins, x="flipper_length_mm", binwidth=3)
```



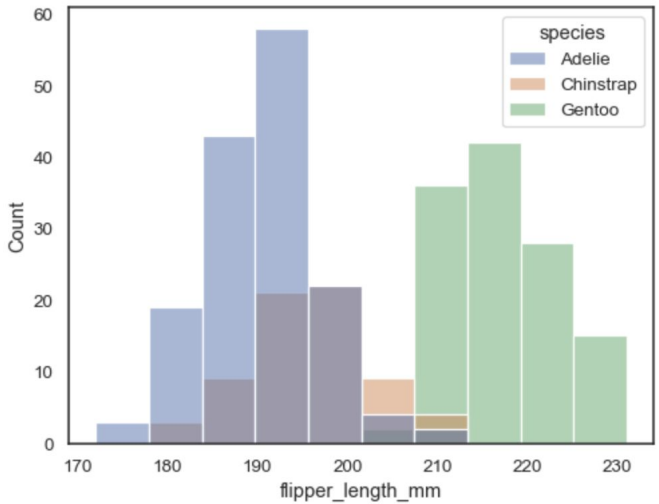
# Distribution plots

| | |
|---|---|
| `displot` | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| `histplot` | Plot univariate or bivariate histograms to show distributions of datasets. |
| `kdeplot` | Plot univariate or bivariate distributions using kernel density estimation. |

```
sns.histplot(data=tips, x="day", hue="sex", multiple="dodge", shrink=.8)
```



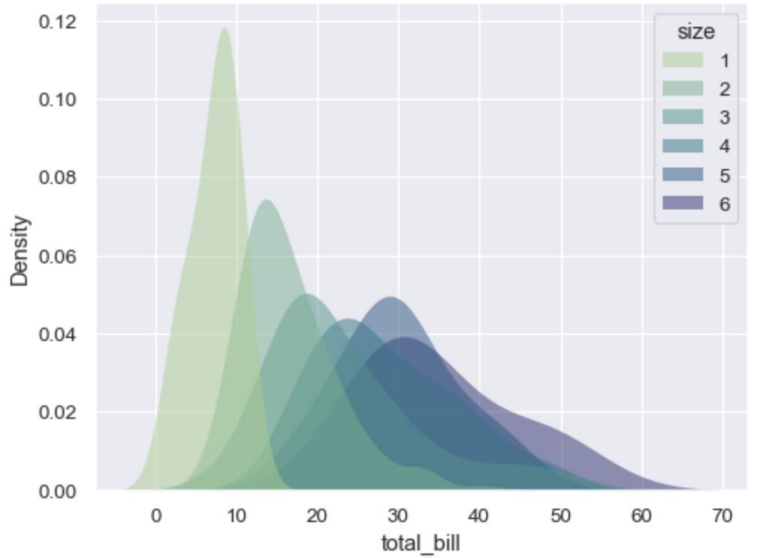

```
sns.histplot(data=penguins, x="flipper_length_mm", hue="species")
```



# Distribution plots

| | |
|---|---|
| `displot` | Figure-level interface for drawing distribution plots onto a FacetGrid. |
| `histplot` | Plot univariate or bivariate histograms to show distributions of datasets. |
| **`kdeplot`** | Plot univariate or bivariate distributions using kernel density estimation. |
| `ecdfplot` | Plot empirical cumulative distribution functions. |
| `rugplot` | Plot marginal distributions by drawing ticks along the x and y axes. |
| `distplot` | DEPRECATED: Flexibly plot a univariate distribution of observations. |

```
sns.kdeplot(
   data=tips, x="total_bill", hue="size",
   fill=True, common_norm=False, palette="crest",
   alpha=.5, linewidth=0,
)
```



# Categorical plots

# Categorical plots

| | |
|---|---|
| `catplot` | Figure-level interface for drawing categorical plots onto a FacetGrid. |
| `stripplot` | Draw a scatterplot where one variable is categorical. |
| `swarmplot` | Draw a categorical scatterplot with non-overlapping points. |
| `boxplot` | Draw a box plot to show distributions with respect to categories. |
| `violinplot` | Draw a combination of boxplot and kernel density estimate. |
| `boxenplot` | Draw an enhanced box plot for larger datasets. |
| `pointplot` | Show point estimates and confidence intervals using scatter plot glyphs. |
| `barplot` | Show point estimates and confidence intervals as rectangular bars. |
| `countplot` | Show the counts of observations in each categorical bin using bars. |