

Seaborn Workshop

Plotting pairwise relationships with PairGrid and pairplot()

- ▶ PairGrid also allows you to quickly draw a grid of small subplots using the same plot type to visualize data in each.
- ▶ In a PairGrid, each row and column is assigned to a different variable, so the resulting plot shows each pairwise relationship in the dataset.
- ▶ This style of plot is sometimes called a scatterplot matrix, as this is the most common way to show each relationship, but PairGrid is not limited to scatterplots.

Seaborn Workshop

It's important to understand the differences between a FacetGrid and a PairGrid. In the former, each facet shows the same relationship conditioned on different levels of other variables. In the latter, each plot shows a different relationship (although the upper and lower triangles will have mirrored plots). Using PairGrid can give you a very quick, very high-level summary of interesting relationships in your dataset.

Seaborn Workshop

The basic usage of the class is very similar to `FacetGrid`. First you initialize the grid, then you pass plotting function to a `map` method and it will be called on each subplot. There is also a companion function, `pairplot()` that trades off some flexibility for faster plotting.

Seaborn Workshop

```
iris = sns.load_dataset("iris")  
g = sns.PairGrid(iris)  
g.map(plt.scatter);  
../_images/axis_grids_42_0.png
```

Seaborn Workshop

Its possible to plot a different function on the diagonal to show the univariate distribution of the variable in each column. Note that the axis ticks wont correspond to the count or density axis of this plot, though.

```
g = sns.PairGrid(iris)
g.map_diag(plt.hist)
g.map_offdiag(plt.scatter);
../_images/axis_grids_44_0.png
```

Seaborn Workshop

A very common way to use this plot colors the observations by a separate categorical variable. For example, the iris dataset has four measurements for each of three different species of iris flowers so you can see how they differ.

```
g = sns.PairGrid(iris, hue="species")
g.map_diag(plt.hist)
g.map_offdiag(plt.scatter)
g.add_legend();
../_images/axis_grids_46_0.png
```

By default every numeric column in the dataset is used, but you can focus on particular relationships if you want.

Seaborn Workshop

```
g = sns.PairGrid(iris, vars=["sepal_length", "sepal_width"])  
g.map(plt.scatter);  
../_images/axis_grids_48_0.png
```

Seaborn Workshop

Its also possible to use a different function in the upper and lower triangles to emphasize different aspects of the relationship.

```
g = sns.PairGrid(iris)
g.map_upper(plt.scatter)
g.map_lower(sns.kdeplot, cmap="Blues_d")
g.map_diag(sns.kdeplot, lw=3, legend=False);
../_images/axis_grids_50_0.png
```


Seaborn Workshop

The square grid with identity relationships on the diagonal is actually just a special case, and you can plot with different variables in the rows and columns.

```
g = sns.PairGrid(tips, y_vars=["tip"], x_vars=["total_bill"])
g.map(sns.regplot, color=".3")
g.set(ylim=(-1, 11), yticks=[0, 5, 10]);
../_images/axis_grids_52_0.png
```

Seaborn Workshop

Of course, the aesthetic attributes are configurable. For instance, you can use a different palette (say, to show an ordering of the hue variable) and pass keyword arguments into the plotting functions.

```
g = sns.PairGrid(tips, hue="size", palette="GnBu_d")
g.map(plt.scatter, s=50, edgecolor="white")
g.add_legend();
../_images/axis_grids_54_0.png
```

Seaborn Workshop

PairGrid is flexible, but to take a quick look at a dataset, it can be easier to use `pairplot()`. This function uses scatterplots and histograms by default, although a few other kinds will be added (currently, you can also plot regression plots on the off-diagonals and KDEs on the diagonal).

```
sns.pairplot(iris, hue="species", size=2.5);  
../_images/axis_grids_56_0.png
```

Seaborn Workshop

- ▶ You can also control the aesthetics of the plot with keyword arguments, and it returns the PairGrid instance for further tweaking.

```
g = sns.pairplot(iris, hue="species", palette="Set2", c  
../_images/axis_grids_58_0.png
```