

# KEY\_Practice25\_Scatterplots

August 28, 2019

## 1 Scatterplots

Let's start by importing `seaborn` and loading/previewing our iris data

```
[1]: # import seaborn
import seaborn as sns
# set up for inline plotting
%matplotlib inline
```

```
[2]: # load iris and preview the data
iris = sns.load_dataset("iris")
iris.head(10)
```

```
[2]:   sepal_length  sepal_width  petal_length  petal_width  species
0          5.1         3.5         1.4         0.2    setosa
1          4.9         3.0         1.4         0.2    setosa
2          4.7         3.2         1.3         0.2    setosa
3          4.6         3.1         1.5         0.2    setosa
4          5.0         3.6         1.4         0.2    setosa
5          5.4         3.9         1.7         0.4    setosa
6          4.6         3.4         1.4         0.3    setosa
7          5.0         3.4         1.5         0.2    setosa
8          4.4         2.9         1.4         0.2    setosa
9          4.9         3.1         1.5         0.1    setosa
```

In the last lesson we examined the relationship between `sepal_length` and `sepal_width`. Now let's look at this relationship for `petal_length` and `petal_width` using a scatterplot.

```
[3]: # plot petal_length vs petal_width
sns.scatterplot('petal_length', 'petal_width', data=iris)
```

```
[3]: <matplotlib.axes._subplots.AxesSubplot at 0x10e16e0f0>
```



This relationship is definitely more clear without any stratification than our last example in the lesson. Let's create this plot with a **correlation trendline** to visualize the trend even better.

```
[4]: # plot petal_length vs petal_width with trendline
sns.lmplot('petal_length', 'petal_width', data=iris)
```

```
[4]: <seaborn.axisgrid.FacetGrid at 0x103598da0>
```



Now let's *stratify* the plot by the `species` variable, using **both** color and marker shape.

```
[5]: # plot petal_length vs petal_width
sns.scatterplot('petal_length', 'petal_width', hue='species', style =_
↪ 'species', data=iris)
```

```
[5]: <matplotlib.axes._subplots.AxesSubplot at 0x111594cf8>
```



We can very clearly see the separation of our three species across these two variables.

Now, let's color our graph using the `sepal_length` variable (no marker shape). What do you notice about the way the graph is colored now?

```
[6]: # plot petal_length vs petal_width
sns.scatterplot('petal_length', 'petal_width', hue='sepal_length', data=iris)
```

```
[6]: <matplotlib.axes._subplots.AxesSubplot at 0x11181b588>
```



Notice that `sepal_length` is a *continuous* variable, compared to the *categorical* variable `species` we originally used to color our plot. Seaborn can tell the difference by examining the `type` of the stratifying variable - `int` and `float` variables are *continuous* and `string` and `boolean` variables are seen as *categorical*.

It is important to consider variable type when choosing the color palette to use in our plots. *Continuous* variables require *sequential* color palettes (that go from light to dark shades, for example) and *categorical* variables require *qualitative* color palettes. You can find built-in seaborn color palettes here: [https://seaborn.pydata.org/tutorial/color\\_palettes.html](https://seaborn.pydata.org/tutorial/color_palettes.html)

After looking through the link above, choose a new **appropriate** color palette for the plot above.

```
[7]: # plot petal_length vs petal_width
sns.scatterplot('petal_length', 'petal_width', hue='sepal_length',
               ↪palette="BuGn", data=iris)
```

```
[7]: <matplotlib.axes._subplots.AxesSubplot at 0x111860b70>
```



Based on this plot, what can you tell about the relationship of `sepal_length` compared to `petal_length`, `petal_width`?