

KEY_Practice18B_Scatterplots

August 15, 2019

1 Scatterplots

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

```
[16]: # import seaborn
import seaborn as sns
```

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

```
[17]:   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.

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

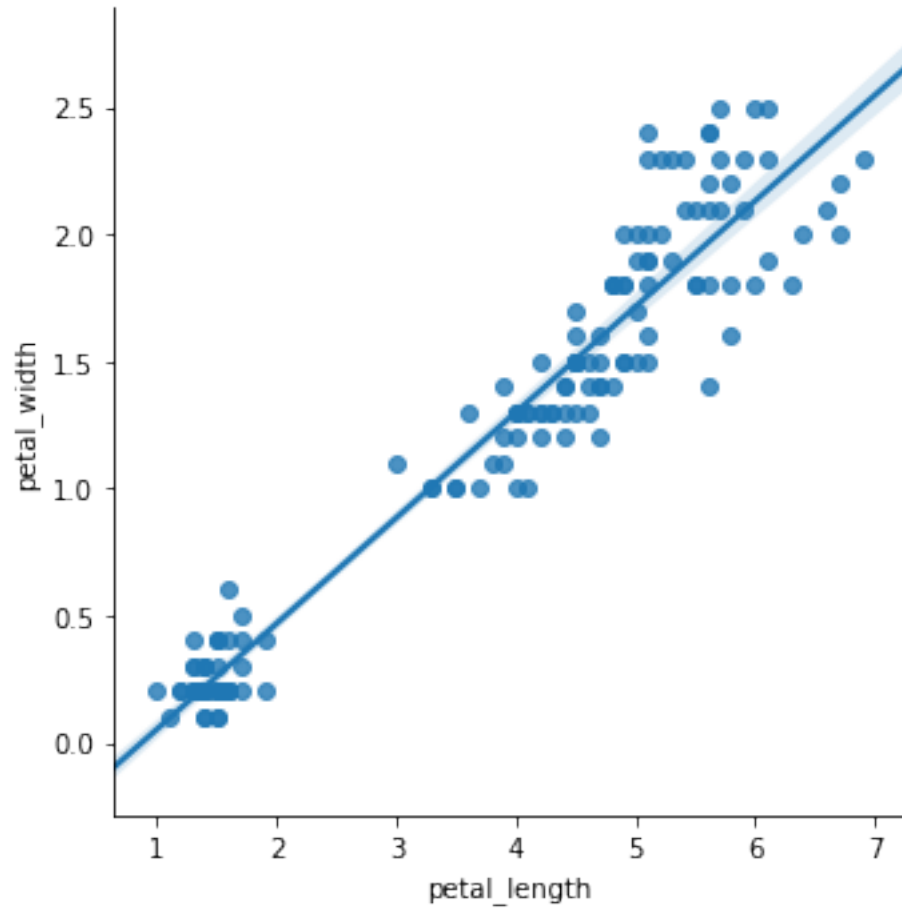
```
[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1a858080>
```



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.

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

```
[19]: <seaborn.axisgrid.FacetGrid at 0x1a1a748c50>
```



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

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

```
[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1aa8bbe0>
```



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?

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

```
[21]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1ab8d080>
```



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

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

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

```
[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1aa8b438>
```



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