KEY_Lesson26_Scatterplots

February 4, 2020

1 Scatterplots

Scatterplots are used to examine the relationship between two variables.

[1]:
```
# import seaborn, matplotlib
import seaborn as sns
import matplotlib.pyplot as plt
# set up inline figures
%matplotlib inline
```

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

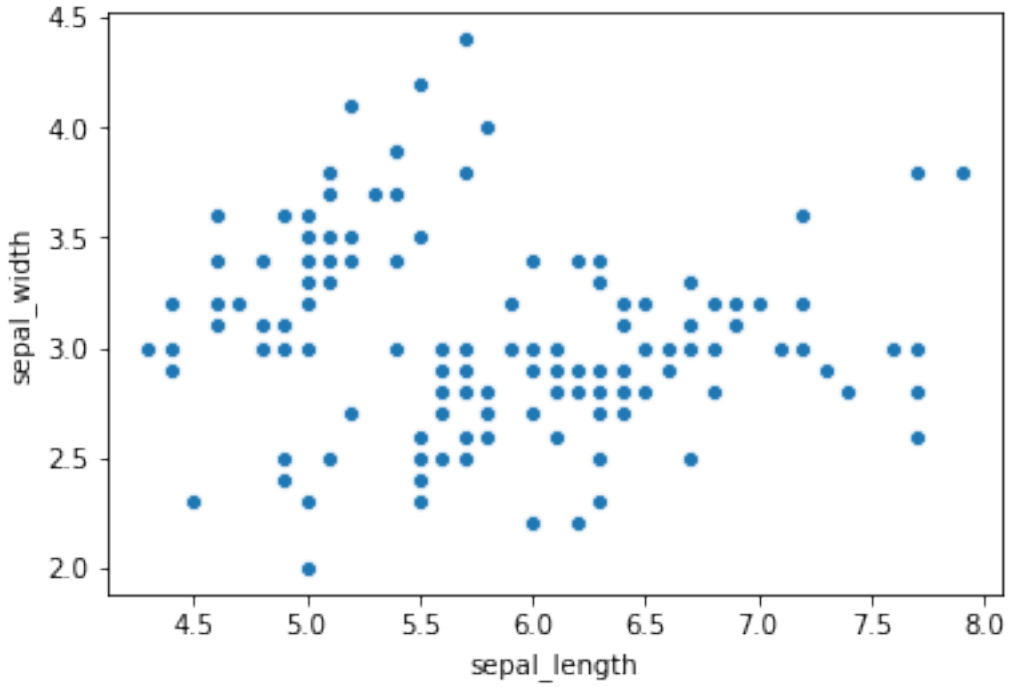
[2]:

	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

Say we want to look at the relationship between `sepal_length` and `sepal_width` within our dataset. We'll use the `sns.scatterplot` function to plot this.

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

[3]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x10ed8a128>
```



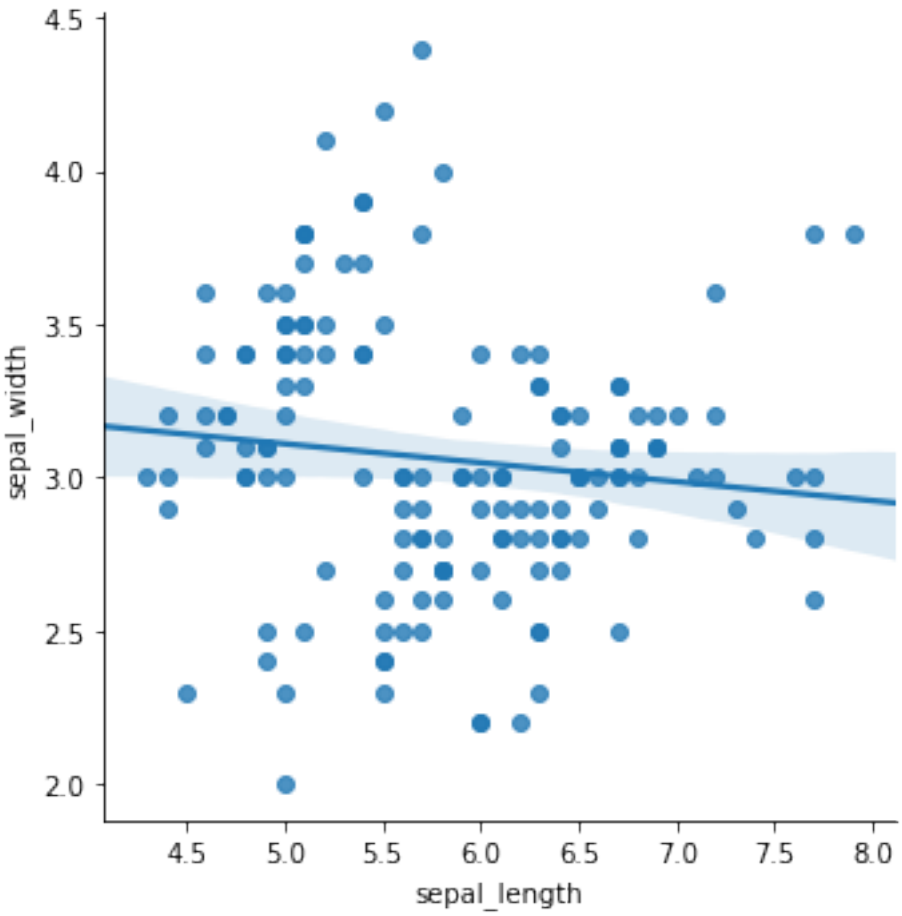
Can you remember what method in the statistics lessons we learned about that tells us about the relationship between two variables?

Correlation!

There is an easy way we can visualize the strength of the correlation on the plot using the `lmplot` function.

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

[4]:
```
<seaborn.axisgrid.FacetGrid at 0x10f0a2908>
```



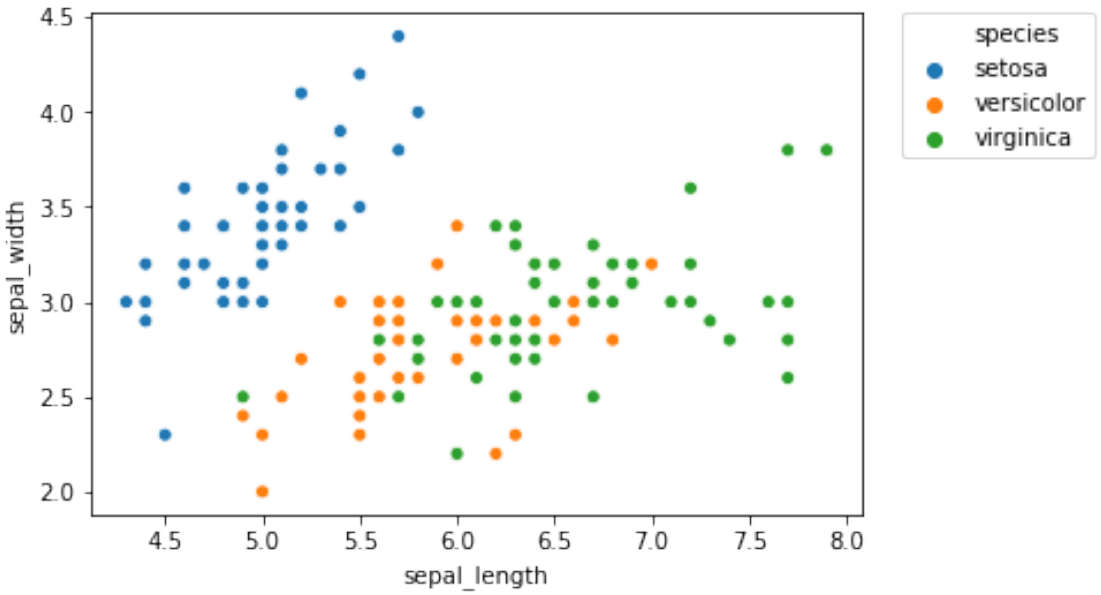
Based on this plot do you think there is a strong relationship between `sepal_length` and `sepal_width` in our data?

This gives us a general idea of the trend between `sepal_length` and `sepal_width`, but what if we wanted to explore the relationship between these variables on a more granular level? For example - if we wanted to see how this relationship might differ between the different species within our dataset? We can separate our plot similar to the way we did in the line graph using the `hue` parameter.

[5]:
```
# plot sepal_length vs sepal_width colored by species
sns.scatterplot('sepal_length', 'sepal_width', data=iris, hue = 'species')

# the line below moves the legend outside of the plot borders
# dont worry about understanding this line of code
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

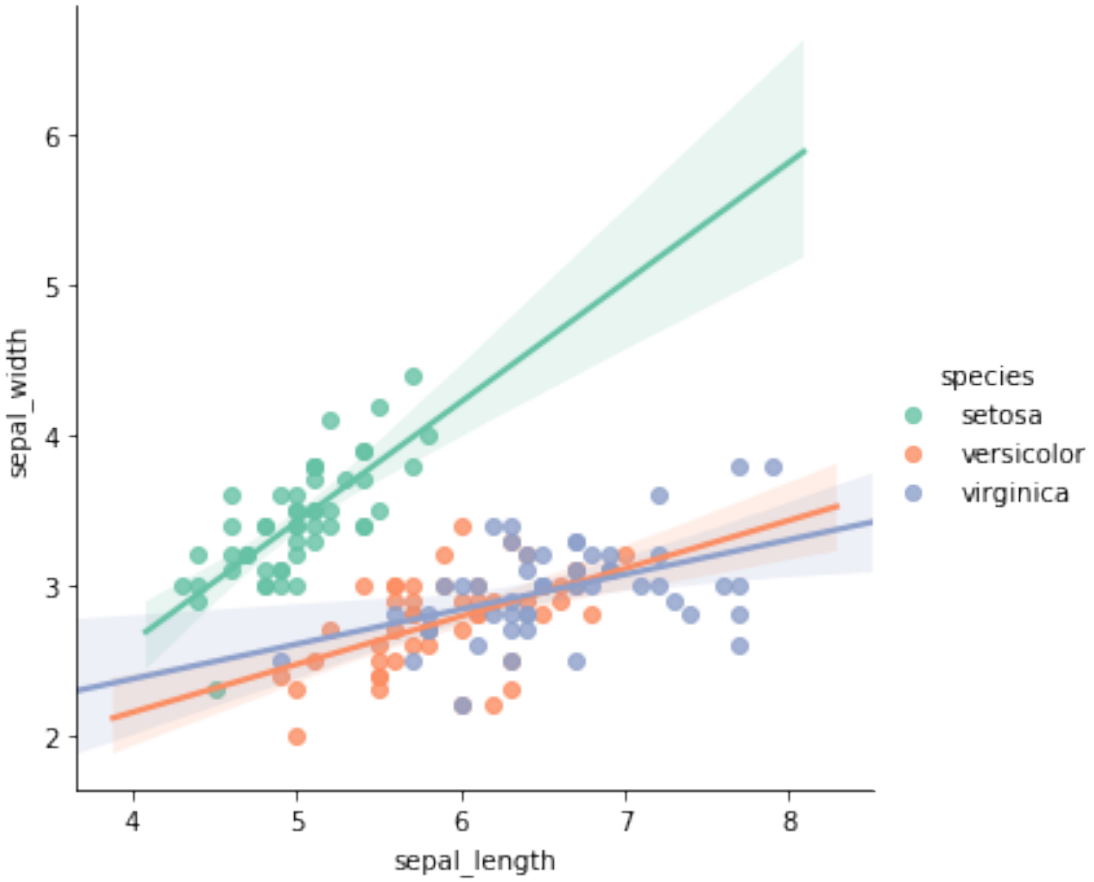
[5]:
```
<matplotlib.legend.Legend at 0x10f1c9ac8>
```



Similarly, we can use the `sns.lmplot` function to add a linear trendline for each species separately. We can also change the color palette using the `palette` parameter

[6]:
```
# plot sepal_length vs sepal_width colored by species
sns.lmplot('sepal_length', 'sepal_width', data=iris, hue = 'species',␣
↪palette="Set2")
```

[6]:
```
<seaborn.axisgrid.FacetGrid at 0x10f2cb3c8>
```



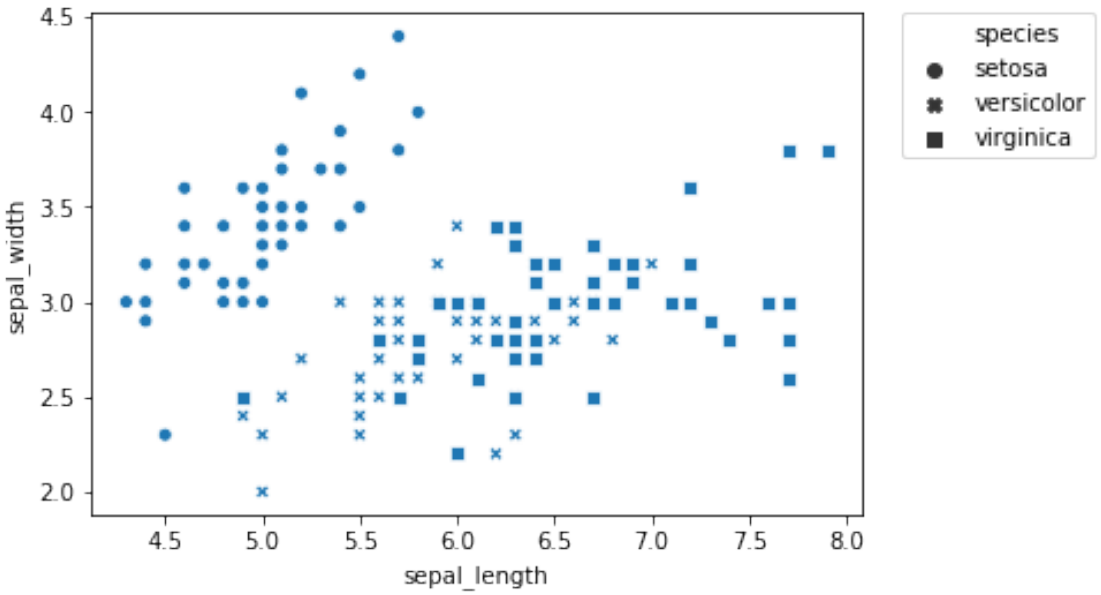
What do you notice about the relationship between our two variables when we separate (i.e. *stratify*) by species?

Instead of stratifying by species using color, we can do so using the marker shape with the `style` parameter.

[7]:
```
# plot sepal_length vs sepal_width colored by species
sns.scatterplot('sepal_length', 'sepal_width', data=iris, style='species',␣
↪palette = 'Set2')

# the line below moves the legend outside of the plot borders
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

[7]:
```
<matplotlib.legend.Legend at 0x10f311780>
```

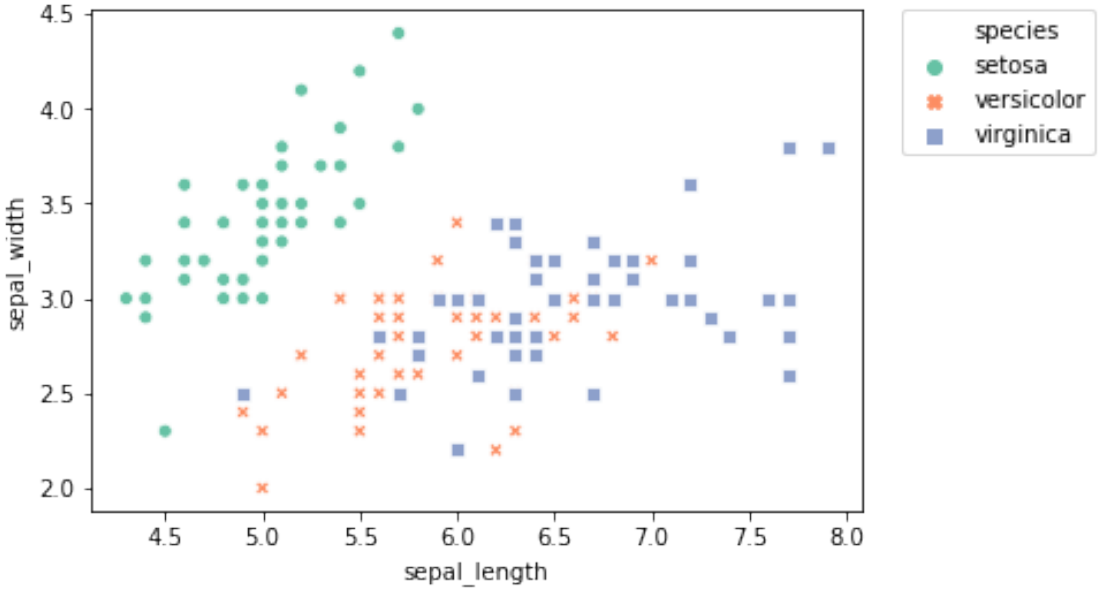


Lastly, we can combine `hue`, `style` and `palette` all together:

[8]:
```
# plot sepal_length vs sepal_width colored by species
sns.scatterplot('sepal_length', 'sepal_width', data=iris, hue = 'species',␣
↪style='species', palette = 'Set2')

# the line below moves the legend outside of the plot borders
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

[8]:
```
<matplotlib.legend.Legend at 0x10f656160>
```



In this lesson we learned: * How to create a scatterplot in `seaborn` * Stratifying a scatterplot by another variable using color (`hue`) * Stratifying a scatterplot by another variable using marker shape (`style`) * Changing the color palette of a stratified plot (`palette`)