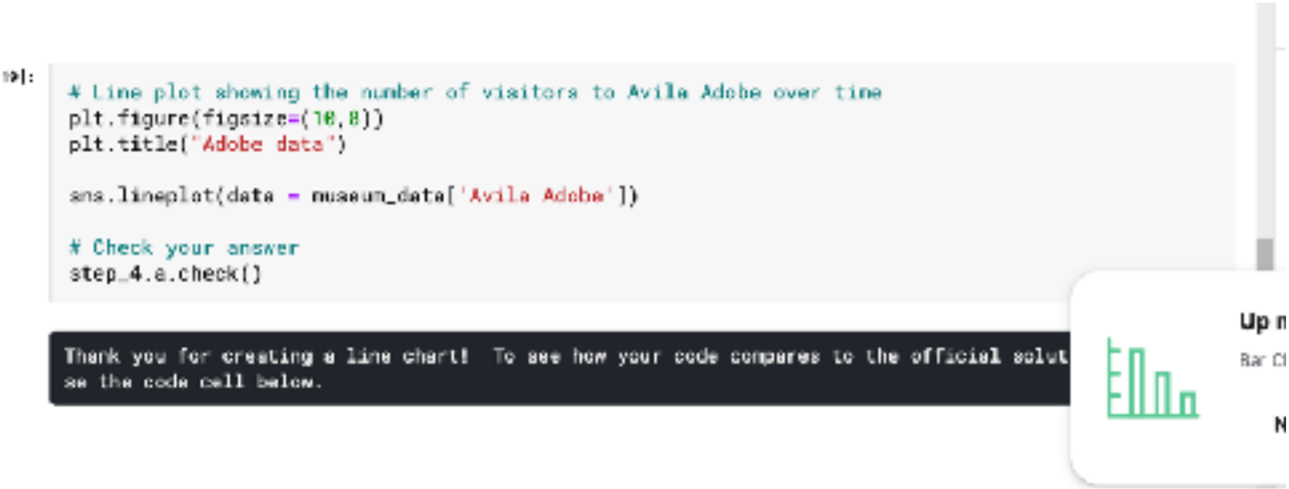
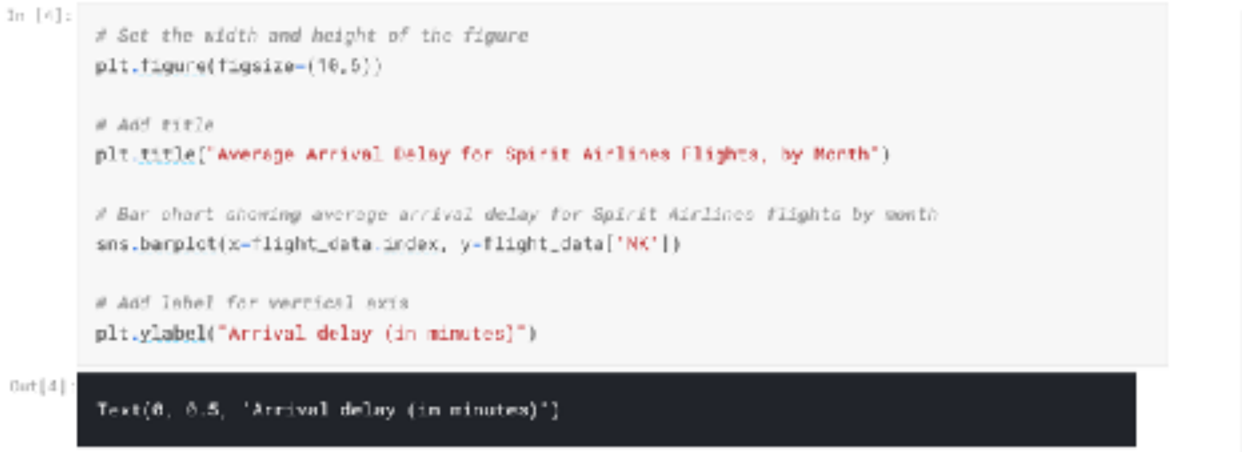
Visualization



19]:

```
# Line plot showing the number of visitors to Avila Adobe over time
plt.figure(figsize=(10,8))
plt.title("Adobe data")

sns.lineplot(data = museum_data['Avila Adobe'])

# Check your answer
step_4.a.check()
```

```
Thank you for creating a line chart!  To see how your code compares to the official solut
se the code cell below.
```

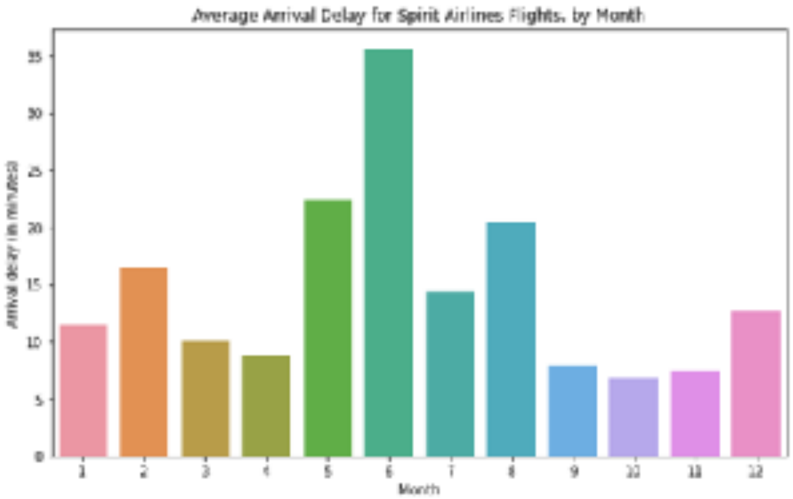


In [4]:

```
# Set the width and height of the figure
plt.figure(figsize=(10,5))

# Add title
plt.title("Average Arrival Delay for Spirit Airlines Flights, by Month")

# Bar chart showing average arrival delay for Spirit Airlines flights by month
sns.barplot(x=flight_data.index, y=flight_data['NK'])

# Add label for vertical axis
plt.ylabel("Arrival delay (in minutes)")
```

Out[4]:

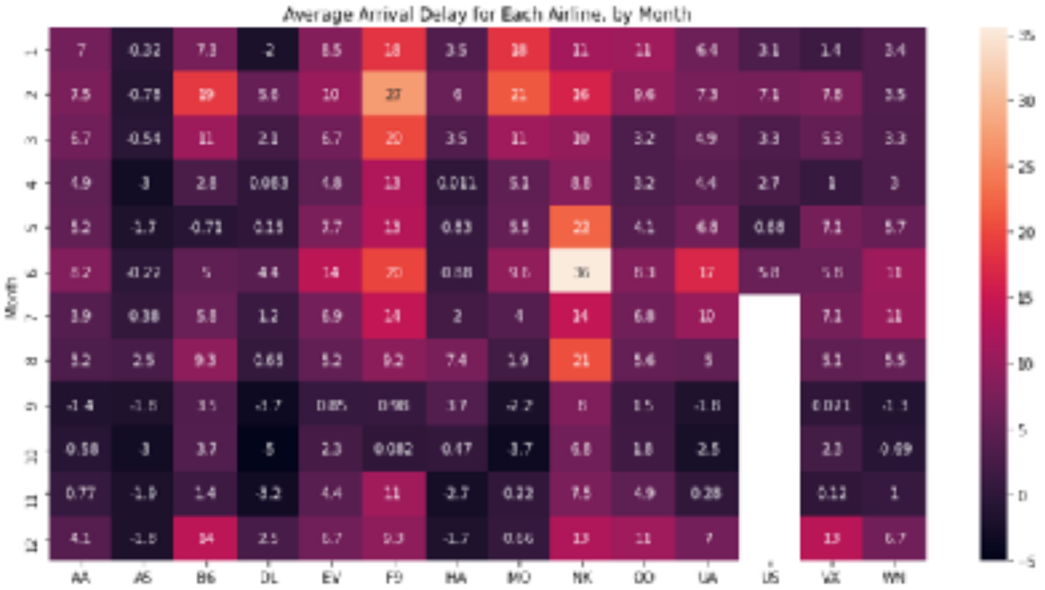
```
Text(0, 0.5, 'Arrival delay (in minutes)')
```



```
# Add title
plt.title("Average Arrival Delay for Each Airline, by Month")

# Heatmap showing average arrival delay for each airline by month
sns.heatmap(data=flight_data, annot=True)

# Add label for horizontal axis
plt.xlabel("Airline")
```

Out[6]:

```
Text(0.5, 42.0, 'Airline')
```



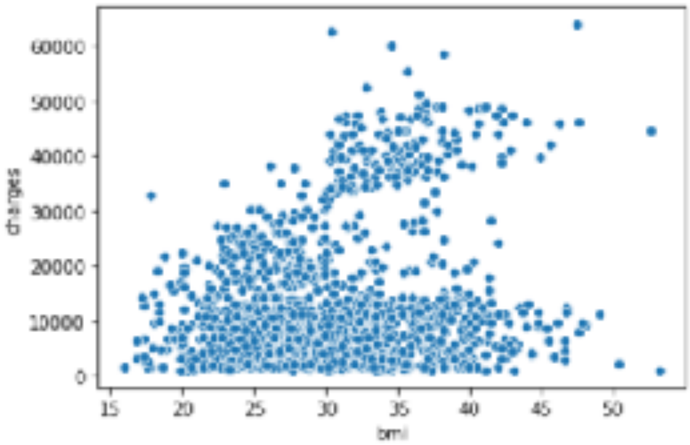
- the horizontal x-axis (`x=insurance_data['bmi']`), and
- the vertical y-axis (`y=insurance_data['charges']`).

In [4]:

```
sns.scatterplot(x=insurance_data['bmi'], y=insurance_data['charges'])
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc446212490>
```



Color-coded scatter plots

We can use scatter plots to display the relationships between *(not two, but...)* three variables! One way of doing this is by color-coding the points.

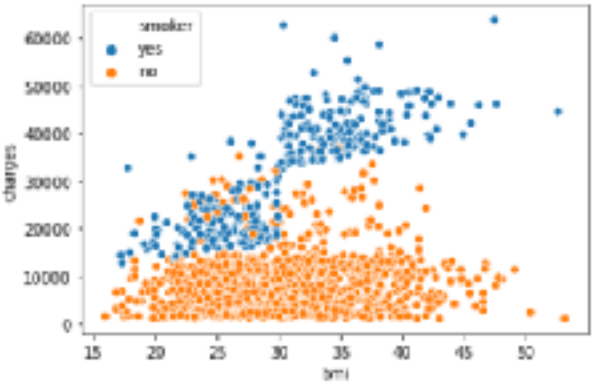
For instance, to understand how smoking affects the relationship between BMI and insurance costs, we can color-code the points by `'smoker'`, and plot the other two columns (`'bmi'`, `'charges'`) on the axes.

[6]:

```
sns.scatterplot(x=insurance_data['bmi'], y=insurance_data['charges'], hue=insuranc
e_data['smoker'])
```

t[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc4435d8790>
```



This scatter plot shows that while nonsmokers to tend to pay slightly more with increasing BMI, smokers pay MUCH more.

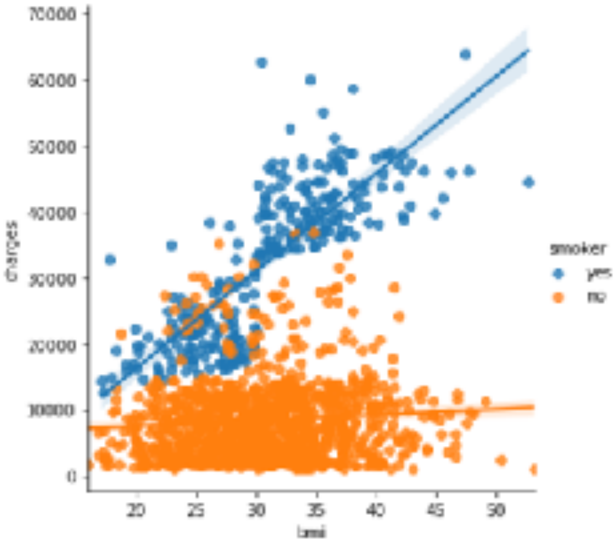
To further emphasize this fact, we can use the `sns.lmplot` command to add two regression lines, corresponding to smokers and nonsmokers. *(You'll notice that the regression line for smokers has a much steeper slope, relative to the line for nonsmokers!)*

[7]:

```
sns.lmplot(x="bmi", y="charges", hue="smoker", data=insurance_data)
```

t[7]:

```
<seaborn.axisgrid.FacetGrid at 0x7fc4435cb3d0>
```



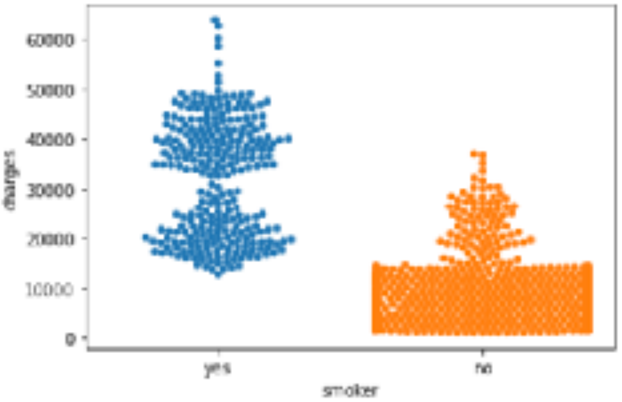
Finally, there's one more plot that you'll learn about, that might look slightly different from how you're used to seeing scatter plots. Usually, we use scatter plots to highlight the relationship between two continuous variables (like `"bmi"` and `"charges"`). However, we can adapt the design of the scatter plot to feature a categorical variable (like `"smoker"`) on one of the main axes. We'll refer to this plot type as a **categorical scatter plot**, and we build it with the `sns.swarmplot` command.

In [8]:

```
sns.swarmplot(x=insurance_data['smoker'],
              y=insurance_data['charges'])
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc441da1d58>
```



Distributions

Data Visualization: 5 of 8

Histograms

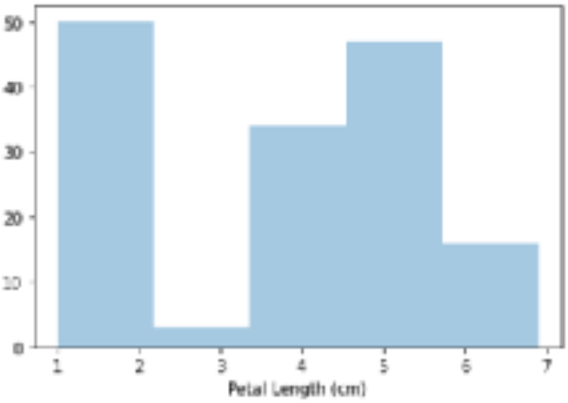
Say we would like to create a **histogram** to see how petal length varies in iris flowers. We can do this with the `sns.distplot` command.

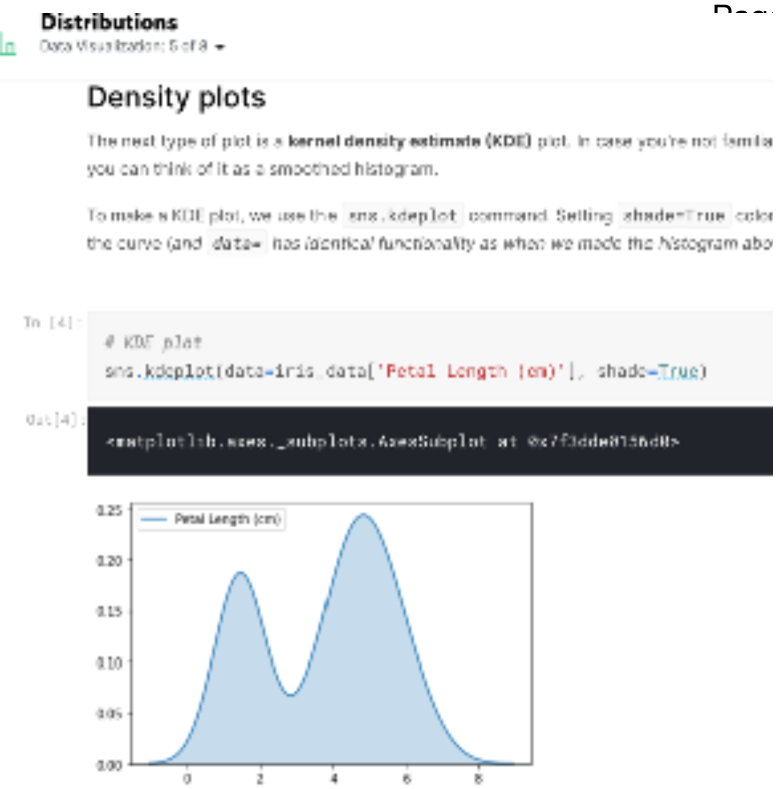
In [3]:

```
# Histogram
sns.distplot(a=iris_data['Petal Length (cm)'], kde=False)
```

Out[3]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3dde14d610>
```





Distributions

Data Visualization: 5 of 8

Density plots

The next type of plot is a **kernel density estimate (KDE)** plot. In case you're not familia
you can think of it as a smoothed histogram.

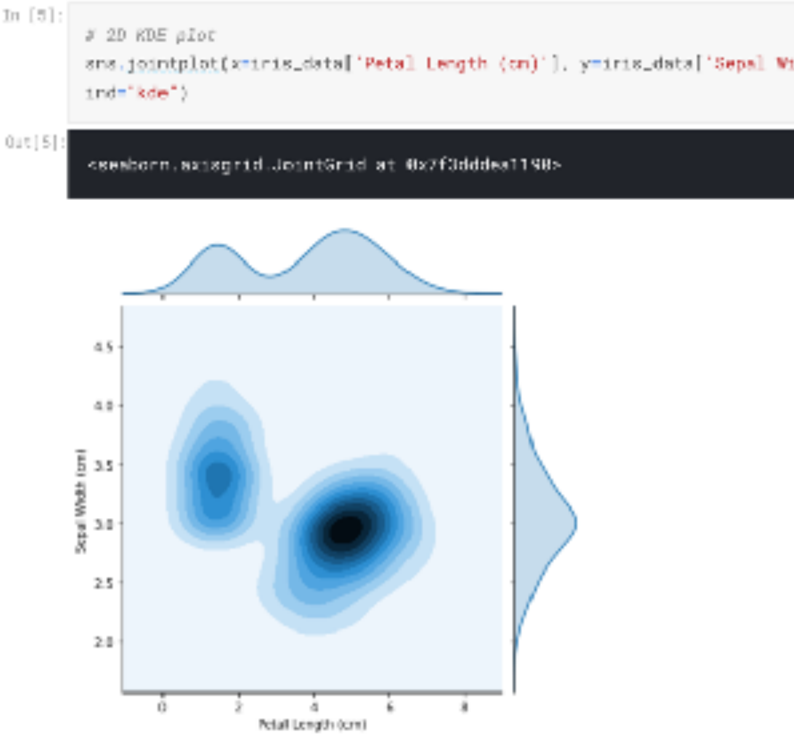
To make a KDE plot, we use the `sns.kdeplot` command. Setting `shade=True` colo
the curve (and `data=` has *identical functionality as when we made the histogram abo*

In [4]:

```
# KDE plot
sns.kdeplot(data=iris_data['Petal Length (cm)'], shade=True)
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3dde0156d0>
```

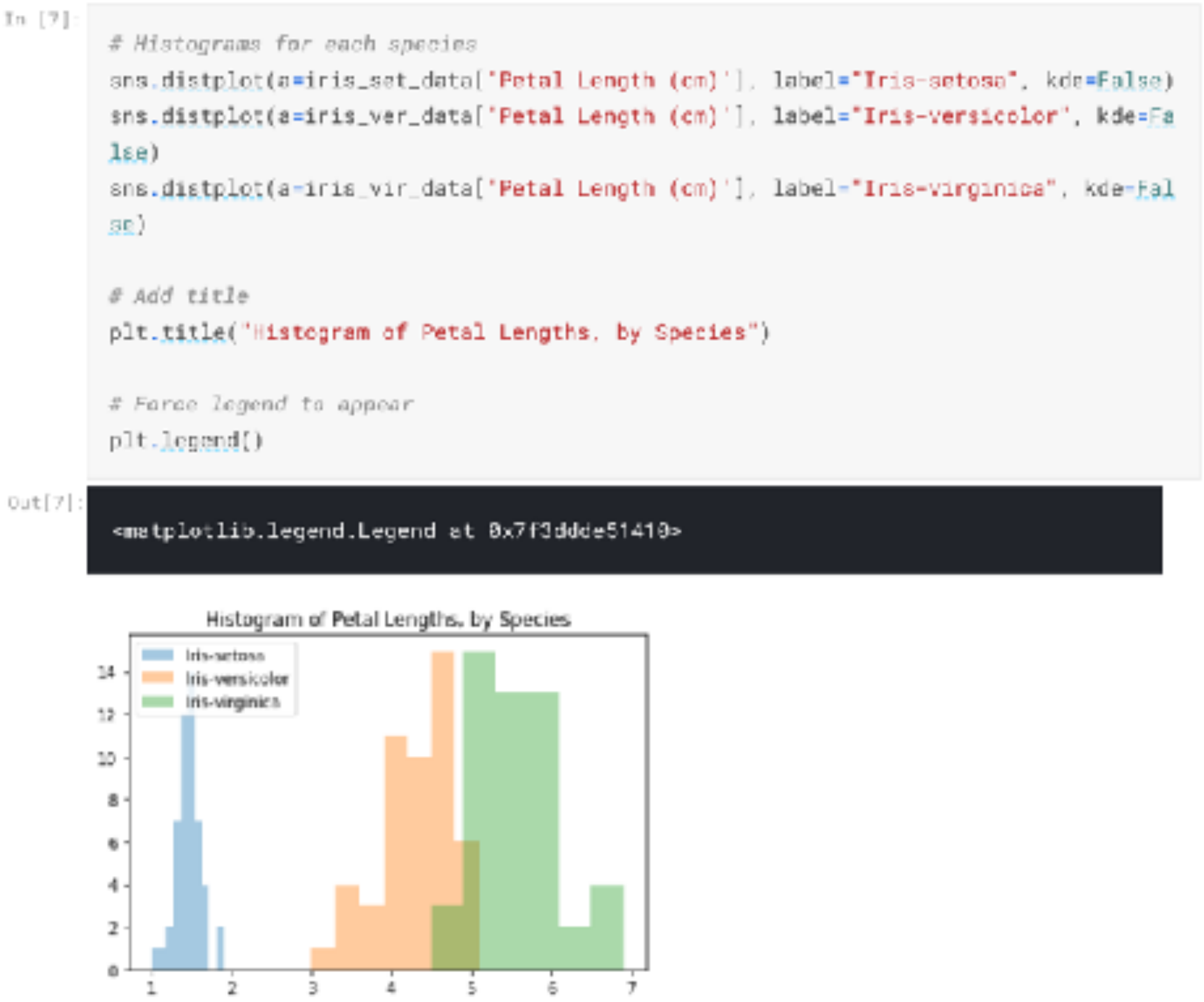


In [5]:

```
# 2D KDE plot
sns.jointplot(x=iris_data['Petal Length (cm)'], y=iris_data['Sepal Wi
ind="kde")
```

Out[5]:

```
<seaborn.axisgrid.JointGrid at 0x7f3dddea1198>
```

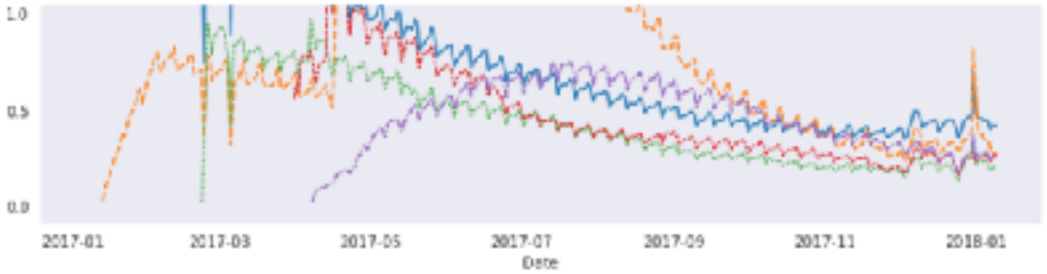


In [7]:

```
# Histograms for each species
sns.distplot(a=iris_set_data['Petal Length (cm)'], label="Iris-setosa", kde=False)
sns.distplot(a=iris_ver_data['Petal Length (cm)'], label="Iris-versicolor", kde=Fa
lse)
sns.distplot(a=iris_vir_data['Petal Length (cm)'], label="Iris-virginica", kde=Fal
se)

# Add title
plt.title("Histogram of Petal Lengths, by Species")

# Force legend to appear
plt.legend()
```

Out[7]:

```
<matplotlib.legend.Legend at 0x7f3ddde51410>
```

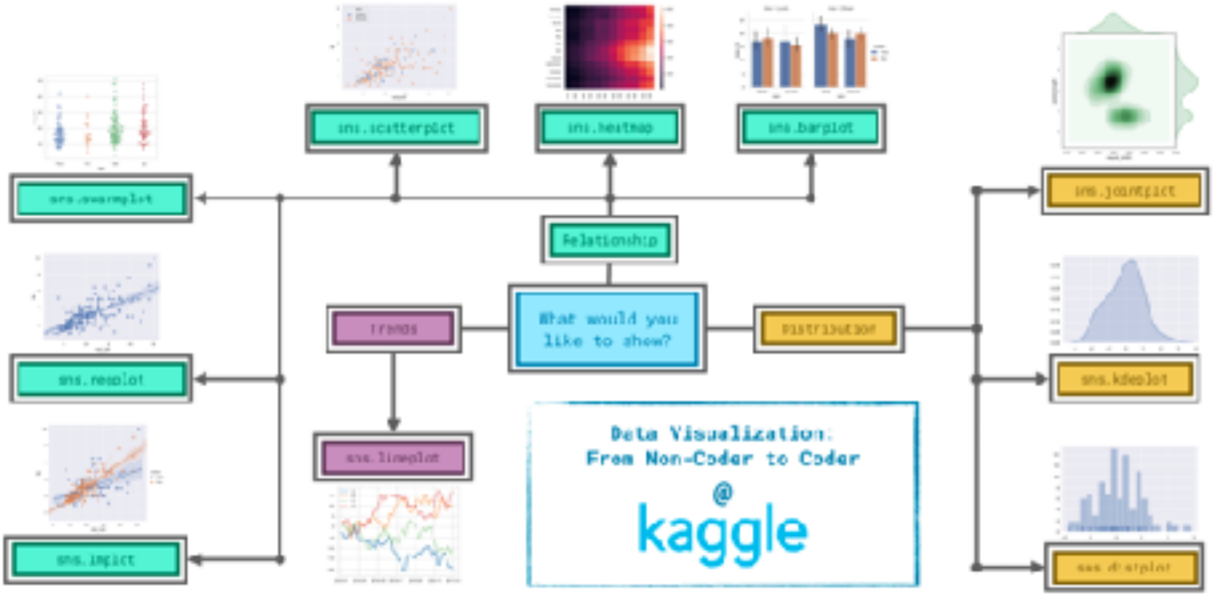


Seaborn has five different themes: (1) `"darkgrid"`, (2) `"whitegrid"`, (3) `"dark"`, (4) `"white"`, and (5) `"ticks"`, and you need only use a command similar to the one in the code cell above (with the chosen theme filled in) to change it.

In the upcoming exercise, you'll experiment with these themes to see which one you like most!

What's next?

Explore seaborn styles in a quick coding exercise!


sns.swarmplot
sns.scatterplot
sns.heatmap
sns.barplot
sns.jointplot
Relationship
sns.regplot
Trends
What would you like to show?
Distribution
sns.kdeplot
sns.lmplot
sns.lineplot
Data Visualization:
From Non-Coder to Coder
@
kaggle
sns.distplot