Day 18: *Plotting Relationships in Python*

After their adventure at Tokyo’s Skytree, Dot continued their globe-spanning journey to their next destination: Bangkok. Dot moved through the city alongside the Chao Phraya River, watching the little boats paddle through the waters towards or away from the Gulf of Thailand. A boatsman gestured to Dot, asking if they would like a ride, but Dot waved him away with a friendly smile. Today wasn’t the day for a boat ride, as they had a destination in mind: the Chatuchak Weekend Market. On the way over to Bangkok, Dot realized the clothes they’d been wearing on the trip were no longer bringing them joy. Dot had been recycling the same few outfits for many days and desperately wanted a change. Plus, they lost a couple of articles of clothing in a hotel room along the way. When Dot arrived in Bangkok, they read something about the Chatuchak market and realized it was the perfect opportunity to snag a fresh new fit.

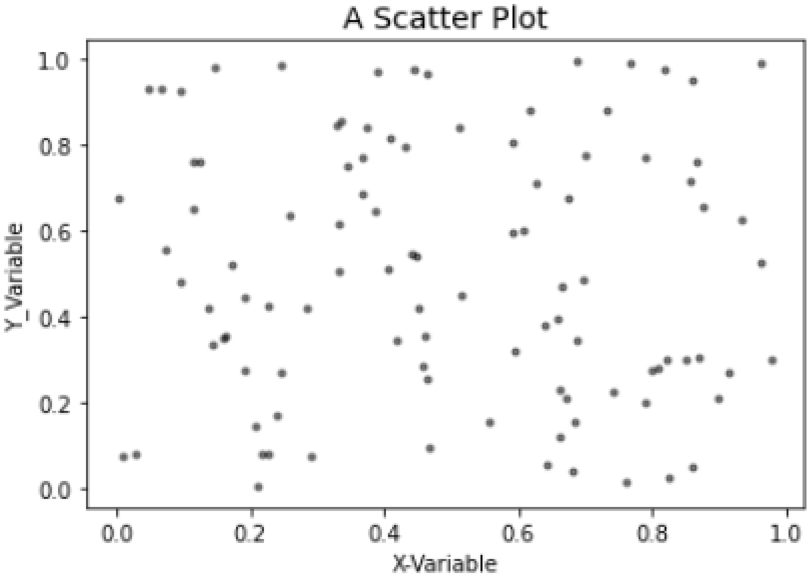
The market stretched on endlessly, the seemingly endless stalls and booths selling all kinds of wares, from clothing to food to books to things Dot couldn’t even name. Standing among the swarming crowds slack-jawed, they didn’t know where to start. Finally, a retailer yelled over at Dot, pulling them out of their consumerist trance. “You there! Do you want to buy this frog?” the salesperson said from behind the counter of their stall. Hypnotized and intrigued, Dot ambled over to him. “What do you mean, a frog?” Dot asked curiously. The salesman opened up his hand, revealing… nothing. Dot raised their eyebrows and scratched their head. “Look closer,” the salesman said patiently. Dot moved closer and looked deeply at the man’s hand. Inside was the tiniest speck of a frog they had ever seen, like a tiny fruit fly letting out faint ribbits. “Well? Do you want to buy it?” the man asked. Dot shook their head and explained that they were simply looking for a fresh new fit. The man clucked his tongue and whisked an outfit out from under the counter. “It’s perfect!” Dot exclaimed. “Wow, this market really does have everything.” Folding the outfit and pushing it into a bag, the salesman explained that the infiniteness of the market was important, as tourism played a big part in strengthening Thailand’s economy. Will you indulge Dot’s curiosity by finding out how tourism impacts the Thai economy?

Tutorial

In the past challenges, we've looked at various methods of data visualization like bar graphs, histograms, and boxplots. Today we'll be covering another visualization method: **scatterplots**.

What is a scatterplot?

A scatterplot is a type of data visualization that shows the relationship between two numerical variables. One variable is plotted on the *x axis*, and the other is plotted on the *y axis*. Each data point is plotted according to (x,y) coordinates related to the two variables' values.



With **scatterplots**, we can visually identify certain patterns or relationships. Typically, we refer to these relationships as correlations.

A positive correlation refers to when y variables increase as x variables increase. Below are two examples of a positive correlation, with the first plot showing a strong positive correlation and the second plot showing a weak positive correlation.

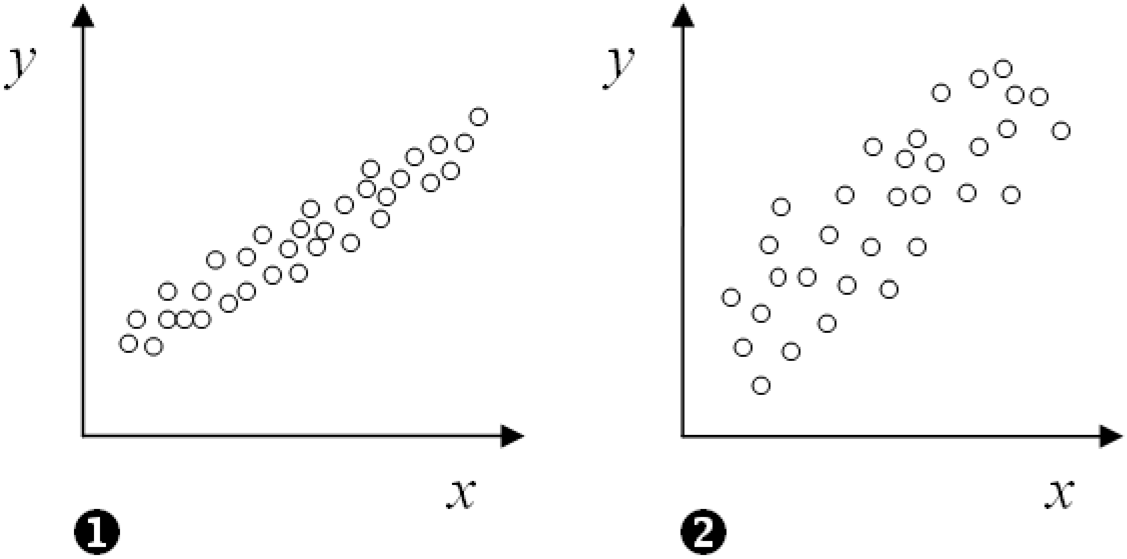


Image Taken from *Wikimedia Commons*

A negative correlation refers to when y variables decrease as x variable increases.

A no correlation refers to when there doesn't seem to be a visible relationship. This can be seen in the first image of this tutorial.

Building a scatterplot with Matplotlib

Here is how a scatterplot can be built using Matplotlib:

```
plt.figure()
plt.scatter(x = df['numerical_data'], y = df['numerical_data_2'])
plt.show()
```

Determining Correlation in Pandas

In addition to creating a scatter plot to view relationships between two numerical ranges,pandas has a built-in function that outputs the correlation value for us:

```
pd.DataFrame.corr()
```

The **correlation coefficient** is given to us from a range of -1 to 1. Values closer to -1 indicate a negative correlation, whereas values closer to 1 indicate a positive correlation.

To learn more abouts correlations with python, read this article.

For the matplotlib user guide, click here. For the pandas user guide, click here.

Note: When two variables are highly correlated we can use one to predict the value of the other. Python has a specific package **scikit-learn** that is used for this purpose. This package contains a ton of different machine learning models and techniques. Even though most of these models are outside of the scope of this challenge, it's good to know about this library because it's arguably the most popular and most often used machine learning library.

Challenge

Play around with the scatterplot and test out different correlations between the numerical variables in the dataset. Then, help Dot by answering these questions:

1. **What kind of correlation is there between the `Receipts (bn $)` and `% of GNP`?**
2. **What is the correlation coefficient between the two columns?**
3. **Which columns are correlated the most?**

Is the correlation visible in a scatterplot as well?

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv('thai_tourism.csv')
```

In [3]:

```
# SOLUTION

df.corr()
```

Out[3]:

	Year	Number of tourists (m)	Receipts (bn $)	% of GNP
Year	1.000000	0.936905	0.930061	0.870395
Number of tourists (m)	0.936905	1.000000	0.994426	0.955270

	Year	Number of tourists (m)	Receipts (bn $)	% of GNP
Receipts (bn $)	0.930061	0.994426	1.000000	0.932489
% of GNP	0.870395	0.955270	0.932489	1.000000

In [4]:

```
df.corr().loc['Receipts (bn $)','% of GNP']
```

Out[4]:

```
0.9324892611735859
```

In [5]:

```
corr = df.corr().replace(1,0) # we replace 1 with 0 so max() function is not affected b
```

In [6]:

```
# maximum is in the row with index "Receipts (bn $)"
corr.max().sort_values()
```

Out[6]:

```
Year                      0.936905
% of GNP                  0.955270
Number of tourists (m)    0.994426
Receipts (bn $)           0.994426
dtype: float64
```

In [7]:

```
corr['Receipts (bn $)'].idxmax()
```

Out[7]:

```
'Number of tourists (m)'
```