6.4.2 Find the Correlation Between Latitude and Maximum Temperature

Now that you are familiar with generating linear regression lines and equations, you can put this knowledge to work by generating a regression line for latitude and maximum temperature for the Northern and Southern Hemispheres.

You send Jack a quick text to let him know you'll be back to APIs soon—and you're excited to fill him in on what you've learned about this side project.

Using the data from the Northern and Southern Hemispheres, we are going to perform linear regression on all four weather parameters: maximum temperature, humidity, cloudiness, and wind speed.

We have an algorithm that performs the linear regression; returns the equation of the regression line, and correlation coefficient, and *p* value; and adds the regression line to a scatter plot of city weather data. Below, the code looks like what we have used before.

```
# Perform linear regression.
(slope, intercept, r_value, p_value, std_err) = linregress(x_values, y_value
```

```
# Calculate the regression line "y values" from the slope and intercept.
regress_values = x_values * slope + intercept

# Get the equation of the line.
line_eq = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))

# Create a scatter plot of the x and y values.
plt.scatter(x_values,y_values)
# Plot the regression line with the x-values and the y coordinates based on
plt.plot(x_values,regress_values,"r")
# Annotate the text for the line equation and add its coordinates.
plt.annotate(line_eq, (10,40), fontsize=15, color="red")
plt.title(title)
plt.xlabel('Latitude')
plt.ylabel('Temp')
plt.show()
```

We will reuse this code with minor changes for each weather parameter in each hemisphere. The variables for each graph are as follows:

1. The *x* values, the latitudes
2. The *y* values, each of the four weather parameters
3. The *y* label, the weather parameter being plotted
4. The x- and y-values given as a tuple, `(10,40)`, for the regression line equation to be placed on the scatter plot.

With only four small changes to the code, this is a great time to convert our linear regression calculation and plotting to a function! In the function, we can add these four parameters as variables, and when we call the function, pass values to those variables.

Create a Linear Regression Function

In a new cell of our `WeatherPy.ipynb` Jupyter Notebook file, let's create a function, "plot_linear_regression", and add the four parameters inside the parentheses. Our function should look like the following.

```
# Create a function to create perform linear regression on the weather data
# and plot a regression line and the equation with the data.
def plot_linear_regression(x_values, y_values, title, y_label, text_coordina
```

In our function, we have four parameters: `x_values`, `y_values`, `y_label`, and `text_coordinates`, and we will add a fifth parameter for the title, called, `title`. Now, add the algorithm we use to perform the linear regression underneath the function. Our function should look like the following.

```
# Import linregress
from scipy.stats import linregress

# Create a function to create perform linear regression on the weather data
# and plot a regression line and the equation with the data.
def plot_linear_regression(x_values, y_values, title, y_label, text_coordina

    # Run regression on hemisphere weather data.
    (slope, intercept, r_value, p_value, std_err) = linregress(x_values, y_v

    # Calculate the regression line "y values" from the slope and intercept.
    regress_values = x_values * slope + intercept
    # Get the equation of the line.
    line_eq = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2)
    # Create a scatter plot and plot the regression line.
    plt.scatter(x_values,y_values)
    plt.plot(x_values,regress_values,"r")
    # Annotate the text for the line equation.
    plt.annotate(line_eq, text_coordinates, fontsize=15, color="red")
    plt.title(title)
    plt.xlabel('Latitude')
    plt.ylabel(y_label)
    plt.show()
```

If we run this code there will be no output until we call the function with five parameters.

REWIND

To get an output from a function, we need to call the function with the correct number of parameters or arguments for the function.

Create the Hemisphere DataFrames

We will add some code to perform regression analysis on the maximum temperatures in the Northern and Southern Hemispheres. To do this, we will need to create Northern Hemisphere DataFrames from the `city_data_df` DataFrame.

To create a new DataFrame from a current DataFrame, we can use the `loc` method on the current DataFrame. The `loc` method accesses a group of rows and columns in the current DataFrame by an index, labels, or a Boolean array. The syntax to get a specific row from a current DataFrame is `row = df.loc[row_index]`.

Let's apply this method to our `city_data_df` DataFrame by adding the code `index13 = city_data_df.loc[13]` in a cell and running the cell. The output will present all the information at index 13 of the `city_data_df` DataFrame. Note that you may see a different city in your output cell than the one shown in the following image.

```
index13 = city_data_df.loc[13]
index13
```

```
City_ID                      13
City                      Vaini
Country                      IN
Date          2019-08-08 22:26:24
Lat                        15.34
Lng                        74.49
Max Temp                    77.3
Humidity                      91
Cloudiness                   100
Wind Speed                  8.88
Name: 13, dtype: object
```

We can also filter a DataFrame based on a value of a row. For instance, if we wanted to get all Northern Hemisphere latitudes, for latitudes greater than or equal to 0, we can filter the `city_data_df` DataFrame using the code `city_data_df["Lat"] >= 0`. Executing this code will return either "True" or "False" for all the rows that meet these criteria.

```
city_data_df["Lat"] >= 0
```

```
0      True
1     False
2     False
3     False
4     False
5      True
6     False
7     False
8     False
9     False
10    False
```

If we want to return a DataFrame with all data fitting the criteria, for latitudes greater than or equal to 0, we can use the `loc` method on the `city_data_df` DataFrame. Inside the brackets, we would add the conditional filter `city_data_df["Lat"] >= 0` so that our statement would appear as:

```
city_data_df.loc[(city_data_df["Lat"] >= 0)]
```

Also, since this is a DataFrame, we can add the `head()` method at the end to get the first five rows, not counting the row of column headings.

```
city_data_df.loc[(city_data_df["Lat"] >= 0)].head()
```

If we execute this code, our output will be the first five rows of a DataFrame.

	City_ID	City	Country	Date	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed
0	0	Tuktoyaktuk	CA	2019-08-08 22:26:20	69.44	-133.03	42.80	87	75	5.82
5	5	Agadez	NE	2019-08-08 22:26:21	16.97	7.99	98.24	25	0	4.79
13	13	Vaini	IN	2019-08-08 22:26:24	15.34	74.49	77.30	91	100	8.88
15	15	Tasiilaq	GL	2019-08-08 22:26:24	65.61	-37.64	46.40	75	1	4.70
16	16	Dingle	PH	2019-08-08 22:26:24	11.00	122.67	75.43	90	99	11.18

Now assign this DataFrame to the variable `northern_hemi_df` to access the data to perform linear regression.

We can take the same approach to get the cities for the Southern Hemisphere by filtering the `city_data_df` DataFrame for latitudes less than 0.

To create DataFrames for the Northern and Southern Hemispheres' data, add the code to a new cell and run the code.

```
# Create Northern and Southern Hemisphere DataFrames.
northern_hemi_df = city_data_df.loc[(city_data_df["Lat"] >= 0)]
southern_hemi_df = city_data_df.loc[(city_data_df["Lat"] < 0)]
```

Now we can perform linear regression on latitude and maximum temperature from each hemisphere DataFrame.

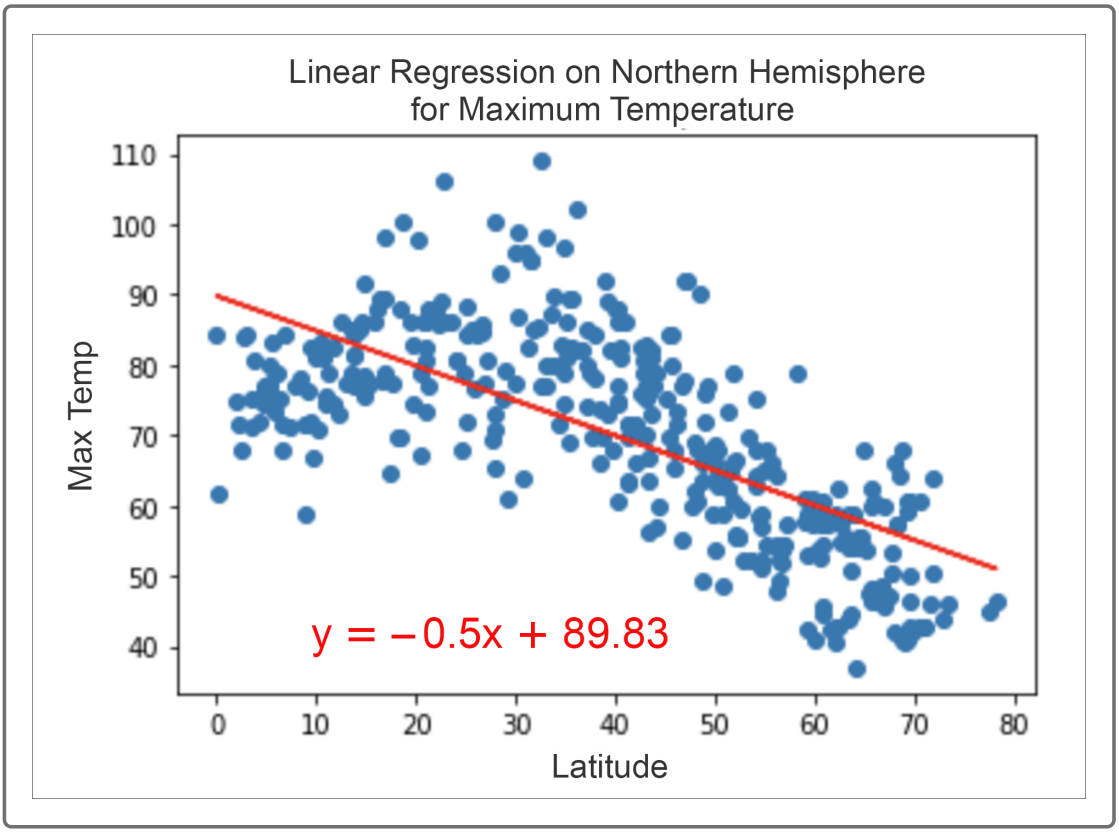
Perform Linear Regression on the Maximum Temperature for the Northern Hemisphere

To generate the linear regression on the maximum temperature for the Northern Hemisphere, we'll need *x* and *y* values. Set the *x* values equal to the latitude column and the *y* values equal to the maximum temperature column from the `northern_hemi_df` DataFrame.

Call the **`plot_linear_regression`** function with the *x* and *y* values, and edit the `title`, `y_label`, and `text_coordinates` for the maximum temperature scatter plot. Add the code to a new cell and run it to generate the linear regression and plot the data.

```
# Linear regression on the Northern Hemisphere
x_values = northern_hemi_df["Lat"]
y_values = northern_hemi_df["Max Temp"]
# Call the function.
plot_linear_regression(x_values, y_values,
                       'Linear Regression on the Northern Hemisphere \
                        for Maximum Temperature', 'Max Temp',(10,40))
```

The scatter plot with the regression line and equation should look like the following.



NOTE

> If the equation for the regression line doesn't show up on your graph, you can change the `text_coordinates` until you see the equation.

Perform Linear Regression on the Maximum Temperature for the Southern Hemisphere

Now we can generate linear regression on the maximum temperature for the Southern Hemisphere.

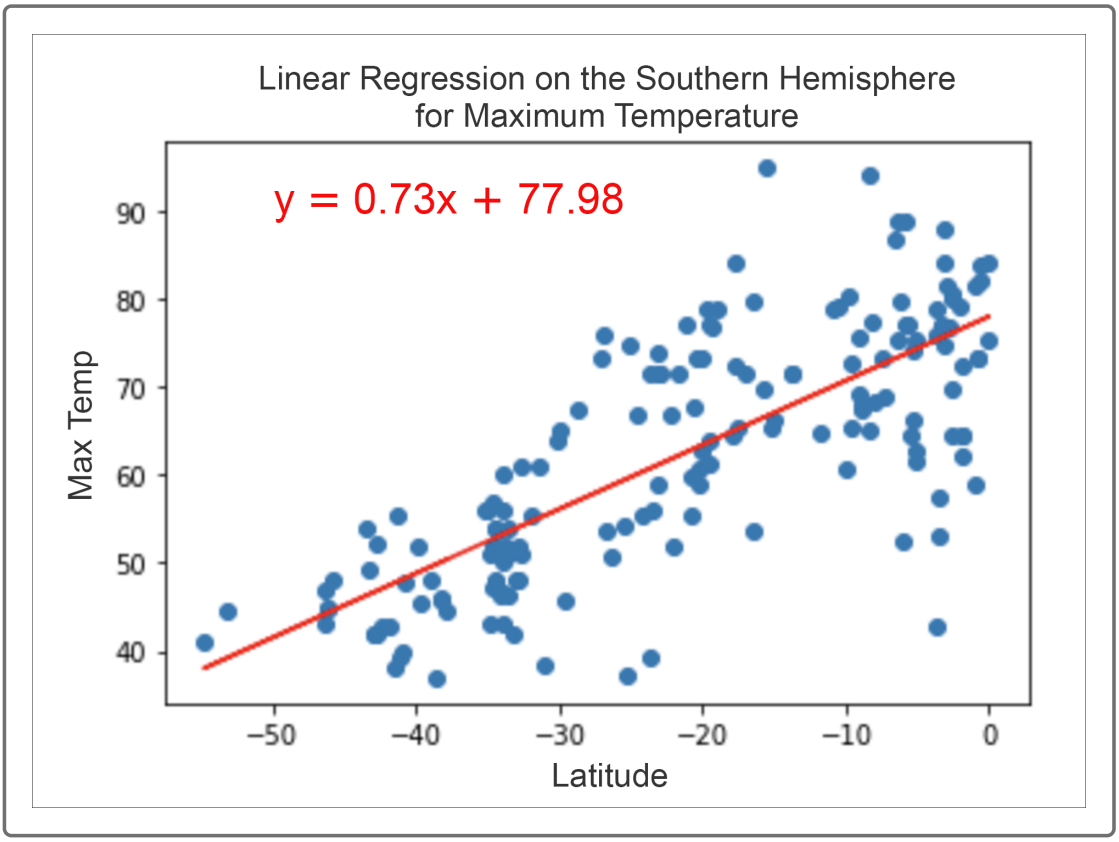
Retake

To generate the linear regression on the maximum temperature for the Southern Hemisphere, reuse the code for the Northern Hemisphere and replace the `northern_hemi_df` DataFrame with the `southern_hemi_df` DataFrame to get the x- and y-values.

Call the `plot_linear_regression` function with the x- and y-values, and edit the `title`, `y_label`, and `text_coordinates` for the maximum temperature scatter plot. Add the code to a new cell and run it to generate the linear regression and plot the data.

```
# Linear regression on the Southern Hemisphere
x_values = southern_hemi_df["Lat"]
y_values = southern_hemi_df["Max Temp"]
# Call the function.
plot_linear_regression(x_values, y_values,
                       'Linear Regression on the Southern Hemisphere \n
                        for Maximum Temperature', 'Max Temp',(-50,90))
```

The scatter plot with the regression line and equation should look like the following.



Congratulations! You have plotted the regression line and equation for latitude and maximum temperature for your Northern and Southern Hemispheres.

FINDING

The correlation between the latitude and the maximum temperature is strong to very strong because the r-value is less than −0.7 for the Northern Hemisphere and greater than 0.7 for the Southern Hemisphere, as shown by the plots here. This means that as we approach the equator, 0° latitude, the temperatures become warmer. And when we are further from the equator the temperatures become cooler. Check the r-values for your plots.