ProjThree

January 8, 2026

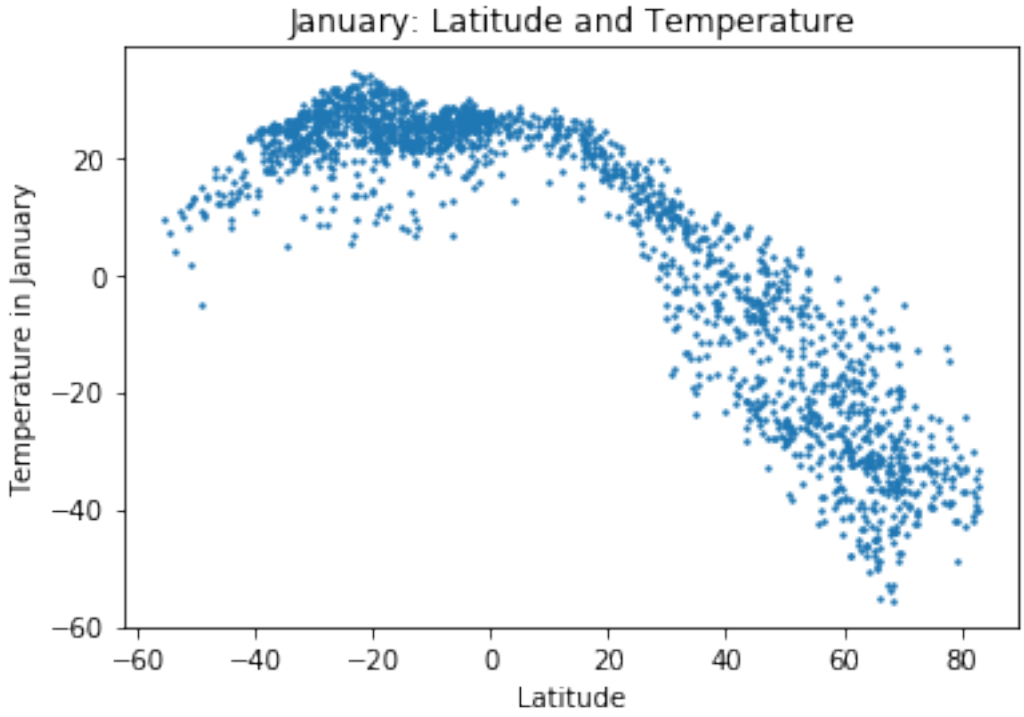
Mapping Global Monthly Temperatures: Patterns Across Latitude and Longitude

Ashley Funes

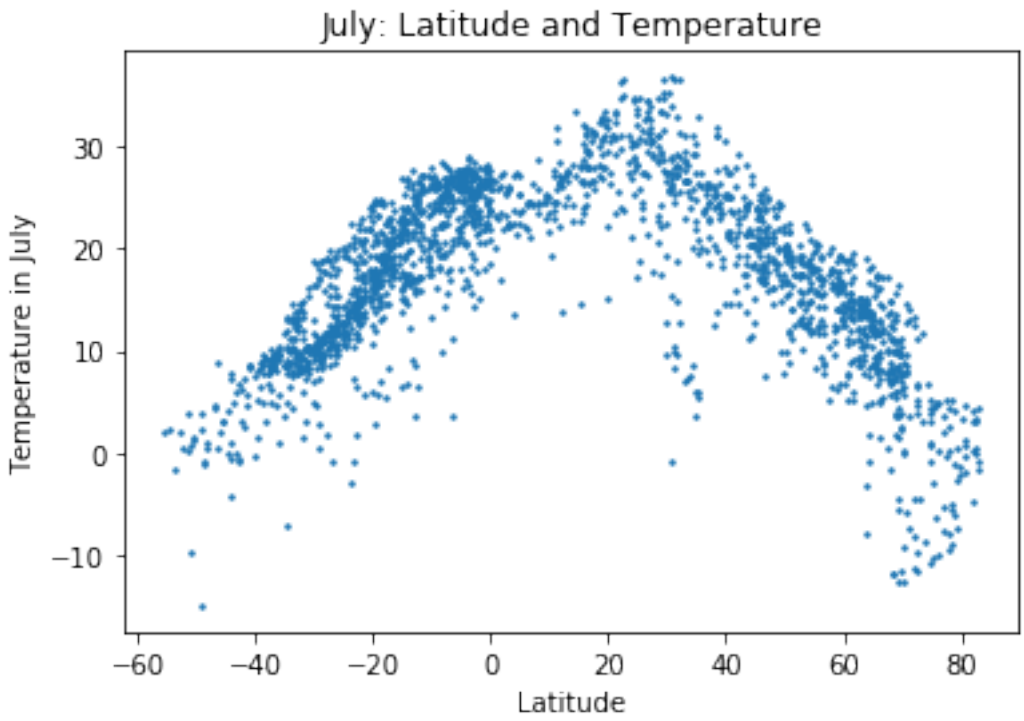
Context: The dataset contains the average monthly temperatures for var ious locations around the world, identified by their latitude and longitude.

[3]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

df = pd.read_csv("shared/data/temperature.csv")

#January:
jan = df[["latitude", "1"]]
#print(jan)

x = jan["latitude"]
y = jan["1"]

plt.scatter(x, y, s=2)
plt.title("January: Latitude and Temperature")
plt.ylabel("Temperature in January")
plt.xlabel("Latitude")
plt.show()

correlation_jan = jan.corr()
print("Correlation")
print(correlation_jan)

print()
print("------------------------------------------------------------")
print()

#July
july = df[["latitude", "7"]]
#print(july)
```

```
x = july["latitude"]
y = july["7"]

plt.scatter(x, y, s=2)
plt.title("July: Latitude and Temperature")
plt.ylabel("Temperature in July")
plt.xlabel("Latitude")
plt.show()

correlation_july = july.corr()
print("Correlation")
print(correlation_july)
```




```
Correlation
           latitude         1
latitude   1.000000 -0.880824
1         -0.880824  1.000000

------------------------------------------------------------
```




```
Correlation
          latitude         7
latitude  1.000000 -0.090811
7        -0.090811  1.000000
```

ANALYSIS:

The justification for wanting to assign an independent and dependent variable is due to our scatter plots. After looking at our plots, it's evident that there is an association between temperature (January and July, respectively) and latitude. Here, I chose my independent variable to be latitude and my dependent variable to be the temperature in January and July (respectively). Latitude measures the distance from a location, and temperature is specific to a given location.

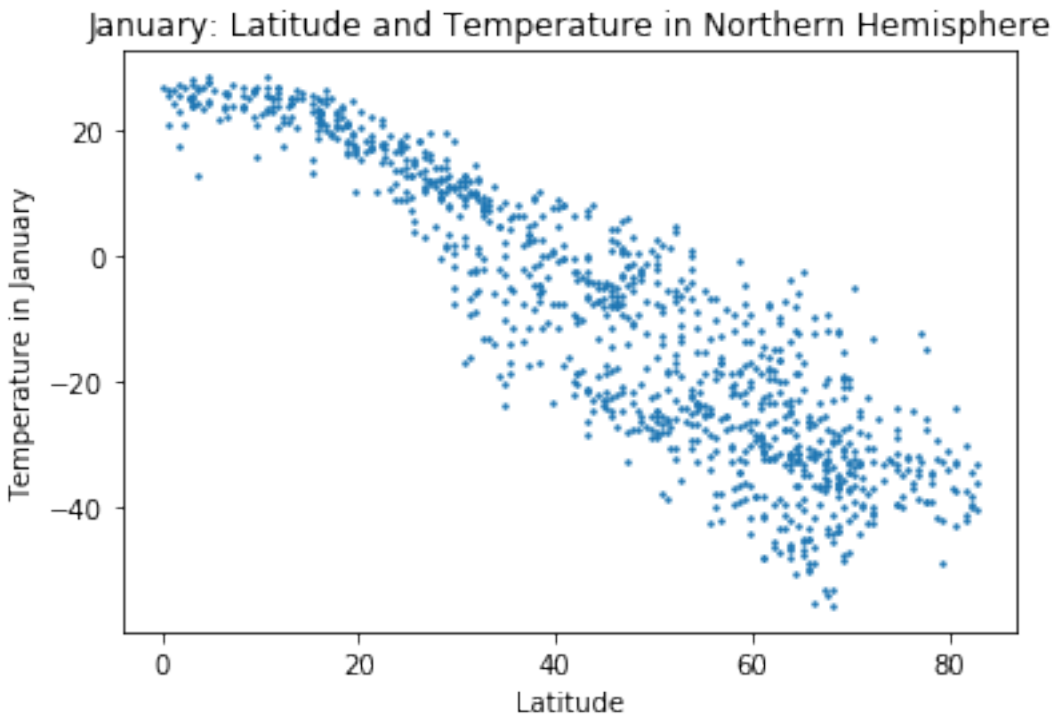
Correlation does not change depending on which variable is treated as X, because correlation measures how strongly BOTH variables move together. Therefore, it wouldn't matter how we assign the variables, because we end up measuring both variables.

The correlation between latitude and January temperature exhibits a strong negative association with a magnitude of -0.880824, whereas the correlation in July shows a weak negative association with a magnitude of -0.090811. In January's scatterplot, we see a strong downward-sloping line. For July, the scatterplot shows a weak downward-sloping line.

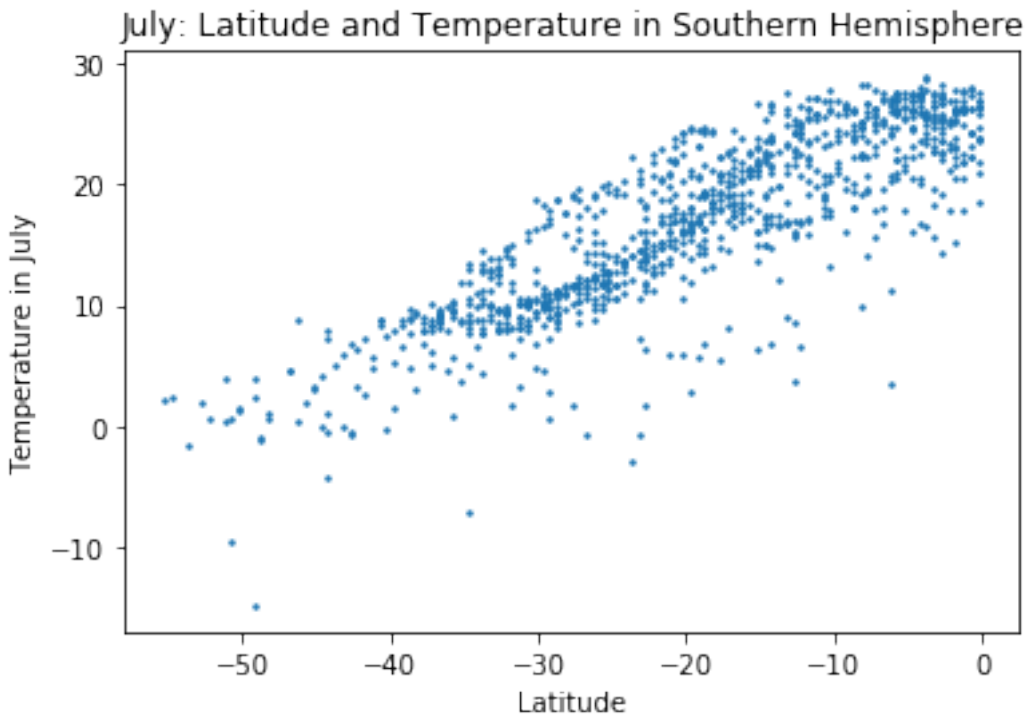
The strong January relationship is caused by the Earth's tilt. The tilt creates a temperature gradient between the cold Northern Hemisphere and the warm Southern Hemisphere. In July, this pattern reverses, and the Northern Hemisphere warms up, and the Southern Hemisphere gets cold. Because in July both the trends in the north and south are opposites, plotting this data looks very

weak.

[4]:
```
#January
jan = df[["latitude", "1"]]
south_jan = jan[df['latitude'] > 0]

#print(south_jan)

x = south_jan["latitude"]
y = south_jan["1"]

plt.scatter(x, y, s=2)
plt.title("January: Latitude and Temperature in Northern Hemisphere")
plt.ylabel("Temperature in January")
plt.xlabel("Latitude")
plt.show()

#correlation
correlation_jan = south_jan.corr()
print("Correlation")
print(correlation_jan)

print()
print("-----------------------------------------------------------------")
print()

#July
july = df[["latitude", "7"]]
south_july = july[df['latitude'] < 0]

#print(south_july)

x = south_july["latitude"]
y = south_july["7"]

plt.scatter(x, y, s=2)
plt.title("July: Latitude and Temperature in Southern Hemisphere")
plt.ylabel("Temperature in July")
plt.xlabel("Latitude")
plt.show()

#correlation
correlation_july = south_july.corr()
print("Correlation")
print(correlation_july)
```

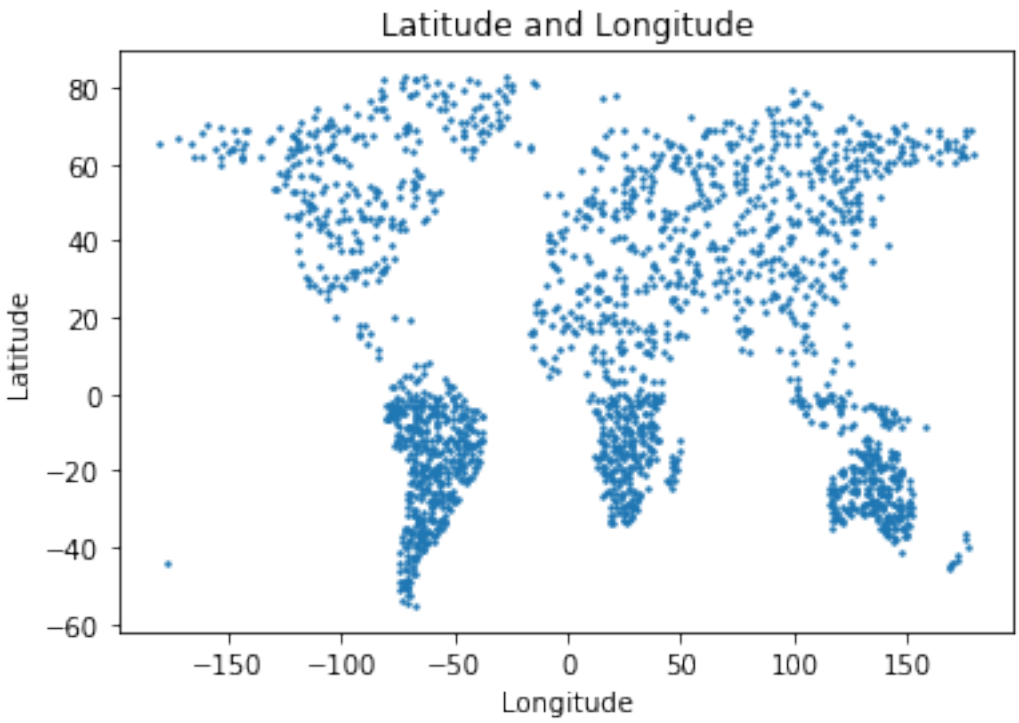



```
Correlation
           latitude         1
latitude   1.000000 -0.904428
1         -0.904428  1.000000

----------------------------------------------------------------
```




```
Correlation
           latitude         7
latitude   1.000000  0.853957
7          0.853957  1.000000
```

[5]:
```
df = pd.read_csv("shared/data/temperature.csv")

y = df["latitude"]
x = df["longitude"]

plt.scatter(x, y, s=2)
plt.title("Latitude and Longitude")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()

#maximum & minimums

min_lat = df["latitude"].min()
max_lat = df["latitude"].max()

print("1. Latitude")
print("Minimum:", min_lat)
print("Maximum:", max_lat)
```



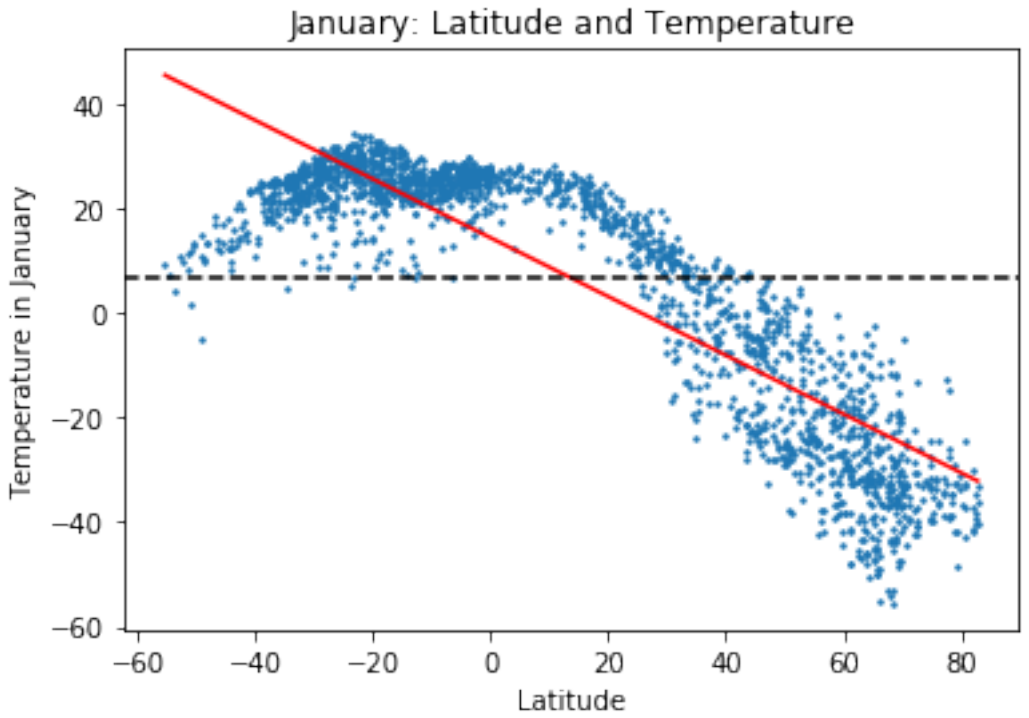

```
1. Latitude
Minimum: -55.25
Maximum: 82.75
```

ANALYSIS:

The distribution of weather stations do not appear to be a random sample of Earth's surface. This land-based sample spans from a minimum latitude of -55.25 to a maximum of 82.75. This range notably excludes the Earth's most extreme polar regions. Thus, our calculations on correlation are an underestimate and the true association between temperature is stronger than expected.

[13]:
```python
# January
jan = df[["latitude", "1"]]
x_jan = jan["latitude"]
y_jan = jan["1"]

# mean and ols
X_jan_const = sm.add_constant(x_jan)
model_jan = sm.OLS(y_jan, X_jan_const).fit()
y_jan_pred = model_jan.predict(X_jan_const)
y_jan_mean = y_jan.mean()

plot_jan_df = pd.DataFrame({'x': x_jan, 'y_pred': y_jan_pred}).sort_values('x')
plt.scatter(x_jan, y_jan, s=2)
plt.plot(plot_jan_df['x'], plot_jan_df['y_pred'], color='red') # ADDED: OLS Line
```

```
plt.axhline(y_jan_mean, color='black', linestyle='--') # ADDED: Mean Line
plt.title("January: Latitude and Temperature")
plt.ylabel("Temperature in January")
plt.xlabel("Latitude")
plt.show()
```



I decided to add an OLS line and the mean for the January scatterplot. Adding the OLS line gives a better visual representation of the negative linear association. The mean line shows the average temperature across all stations. However, whne these lines intersect, we conclude that the stations in the left are warmer than the mean, and the stations to the right are colder than the mean.