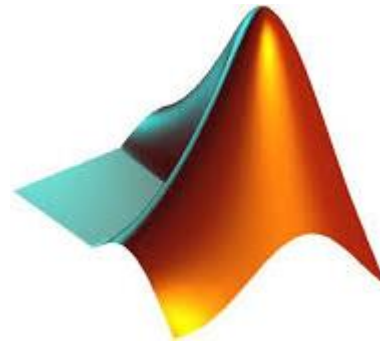


# Getting started With ThingSpeak & MATLAB

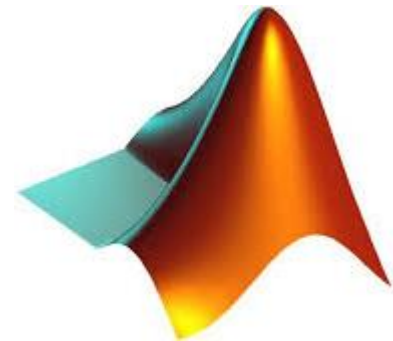




carenuty

# Overview

- Introduction to IoT data logging with ThingSpeak.
- Setting up ThingSpeak account, channels, and fields.
- Connecting sensors to log data.
- Visualizing data with graphs and MATLAB.



# Creating a ThingSpeak Account

Steps:

- Go to *<https://thingspeak.mathworks.com/>*
- Click on Get started for free and Create an Account



Email

No account? [Create one!](#)

By signing in, you agree to our [privacy policy](#).

Next

- Fill in your details (name, email, password)
  - Confirm your email to activate the account
- And set up your password.

Create MathWorks Account



Email Address

**i** To access your organization's MATLAB license, use your school or work email.

Location

United States

First Name

Last Name

Continue

Cancel

# Creating a Channel

## What is a Channel?

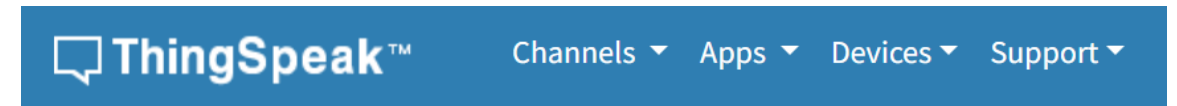
- A channel is where data from your sensor gets stored.

Steps:

1. Navigate to Channels > My Channels > New Channel.



2. Provide a name and description.
3. Create relevant Fields (e.g., Temperature, Humidity).
6. Click Save Channel.



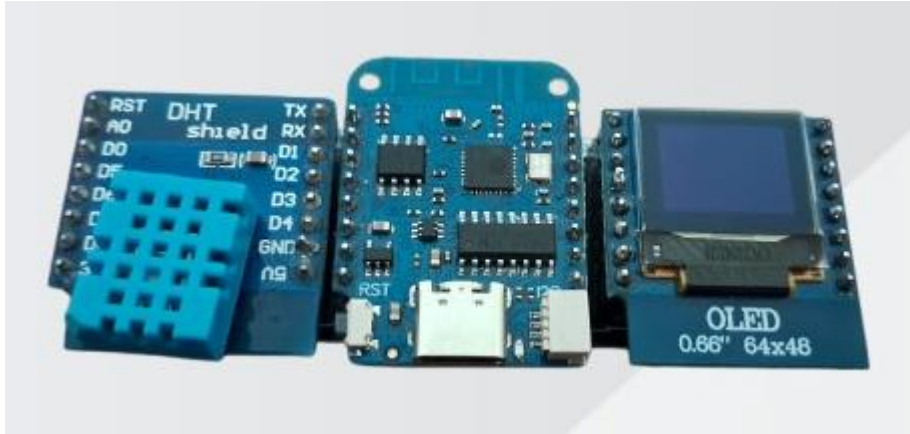
## New Channel

Name	<input type="text" value="Humidity and Temperature monitoring"/>	
Description	<input type="text" value="Monitoring using a DHT11 sensor"/>	
Field 1	<input type="text" value="Temperature"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Humidity"/>	<input checked="" type="checkbox"/>

# Connecting a Sensor to ThingSpeak

## Hardware Requirements:

- Microcontroller (e.g., D1 Mini).
- Sensor (e.g., DHT11 for Temperature and Humidity).
- Display (Optional)
- Internet connection (WiFi).



## Connect Sensor:

1. Program the microcontroller to send data to ThingSpeak via Arduino IDE.

2. Use the **Write API Key** from your channel.

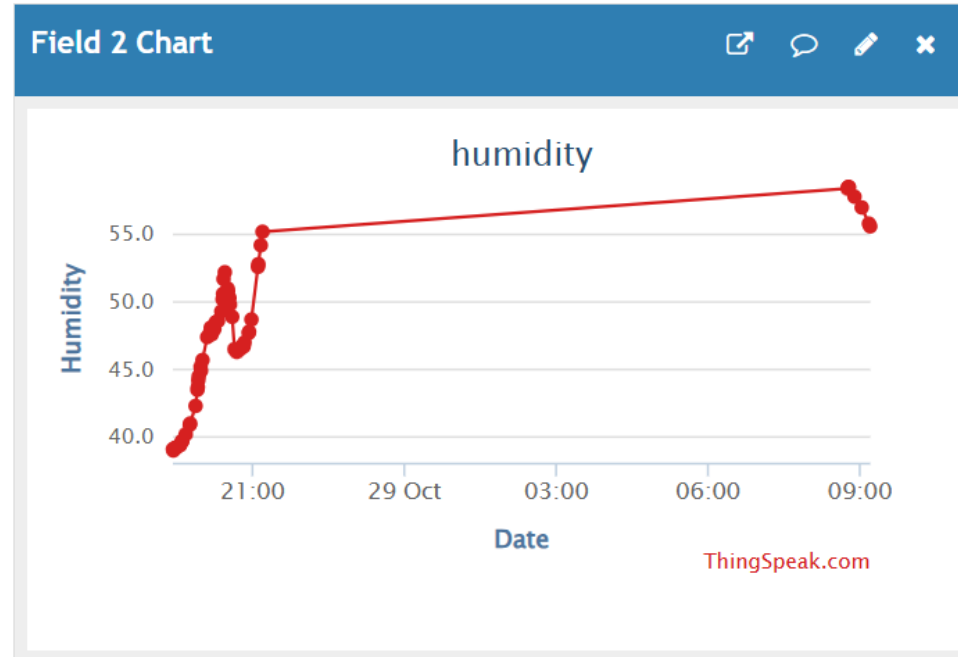
3. Install the necessary libraries

3. Upload the code to the microcontroller.

4. Verify if data is being sent successfully.

```
1  #include <WiFi.h>
2  #include <ThingSpeak.h>
3  #include <Adafruit_Sensor.h>
4  #include <DHT.h>
5
6  const char* ssid = "Lukrasta";
7  const char* password = "Cycy12345";
8
9  #define DHTPIN 4      // Digital pin connected to the DHT sensor
10 #define DHTTYPE DHT11 // DHT 11
11
12 WiFiClient client;
13
14 DHT dht(DHTPIN, DHTTYPE);
15
16 unsigned long myChannelNumber = 1;
17 const char * myWriteAPIKey = "L9RS3ZNCOF6QEUSS";
18
19 // Timer variables
20 unsigned long lastTime = 0;
21 unsigned long timerDelay = 30000; // 60secs
22
23 // Variable to hold temperature readings
24 float temperatureC;
25 float humidity;
```

29	<code>void setup() {</code>	50	<code>// Get a new temperature reading</code>
30	<code>  Serial.begin(115200); //Initialize serial</code>	51	<code>  temperatureC = dht.readTemperature();</code>
31	<code>  dht.begin();</code>	52	<code>  Serial.print("Temperature (°C): ");</code>
32		53	<code>  Serial.println(temperatureC);</code>
33	<code>  WiFi.mode(WIFI_STA);</code>	54	
34		55	<code>  humidity = dht.readHumidity();</code>
35	<code>  ThingSpeak.begin(client); // Initialize Thin</code>	56	<code>  Serial.print("Humidity (%): ");</code>
36	<code>}</code>	57	<code>  Serial.println(humidity);</code>
37		58	
38		59	<code>// set the fields with the values</code>
39	<code>void loop() {</code>	60	<code>  ThingSpeak.setField(1, temperatureC);</code>
40	<code>  if ((millis() - lastTime) &gt; timerDelay) {</code>	61	<code>  ThingSpeak.setField(2, humidity);</code>
41	<code>    // Connect or reconnect to WiFi</code>	62	
42	<code>    if(WiFi.status() != WL_CONNECTED){</code>	63	<code>  int x = ThingSpeak.writeFields(myChannelNumber,</code>
43	<code>      Serial.print("Attempting to connect");</code>	64	
44	<code>      while(WiFi.status() != WL_CONNECTED){</code>	65	<code>    if(x == 200){</code>
45	<code>        WiFi.begin(ssid, password);</code>	66	<code>      Serial.println("Channel update successful.");</code>
46	<code>        delay(5000);</code>	67	<code>    }</code>
47	<code>      }</code>	68	<code>  else{</code>
48	<code>      Serial.println("\nConnected.");</code>	69	<code>    Serial.println("Problem updating channel. HTTP</code>
49	<code>    }</code>	70	<code>  }</code>
50		71	<code>  lastTime = millis();</code>
51	<code>  // Get a new temperature reading</code>	72	<code>}</code>
52	<code>  temperatureC = dht.readTemperature();</code>	73	<code>}</code>
53	<code>  Serial.print("Temperature (°C): ");</code>		





# Visualizing Data on ThingSpeak

## How to visualize data:

1. Open your channel.
2. Go to Private View
3. Scroll to see the generated graphs.
4. Customize the time range or graph style.



## Humidity and Temperature Monitoring

Channel ID: **2677119**

Author: **mwa0000035322476**

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import

[+ Add Visualizations](#)

[+ Add Widgets](#)

[Export recent data](#)

## Channel Stats

Created: 28 days ago

Last entry: 39 minutes ago

Entries: 2037

# MATLAB in ThingSpeak

- MATLAB Analysis: Perform operations on data.
- MATLAB Visualization: Create customized plots.



## MATLAB Visualization

Use interactive ThingSpeak MATLAB® plots to visualize and explore data collected in a channel. You also have access to many more static [MATLAB plots](#).

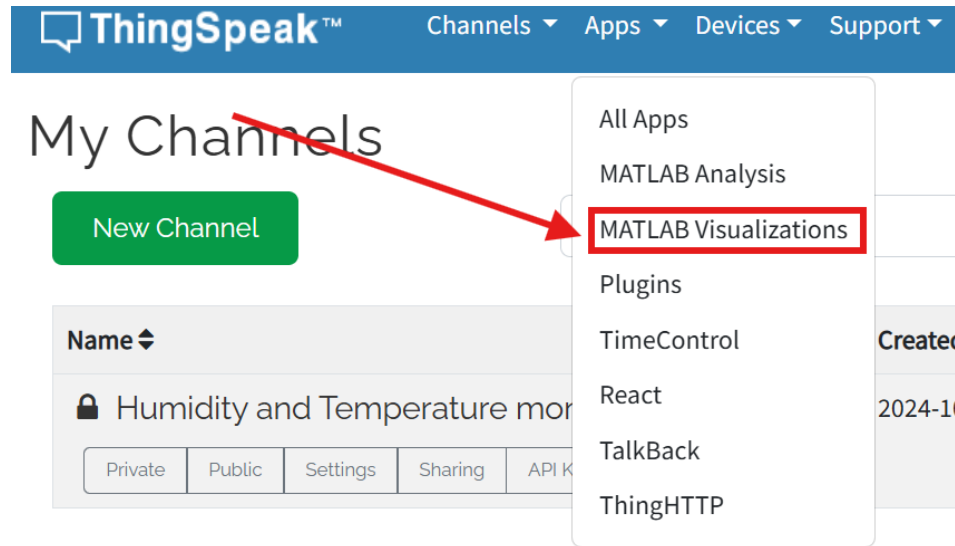
## MATLAB Analysis

- Explore data collected in a channel or scraped from a website
- Find and remove bad data
- Convert data to different units
- Calculate new data
- Build data models

# Writing a MATLAB Script to Visualize Data

Steps :

Go to Apps > MATLAB Visualization.



Click **New** to create a script.



Apps / MATLAB Visualizations

Click **New**, and choose a template to get started. Templates contain sample code.

New

- Select Custom (no starter code)\



Templates:

- ☒ Custom (no starter code)
- ☐ Create a filled area 2-D plot
- ☐ Create a 2-D line plot
- ☐ Create 2-D line plots with y-axes on both left and right side
- ☐ Create a correlated data plot
- ☐ Create a discrete sequence data plot

- Click create and write your MATLAB code (e.g., read and plot data), Save and Run.

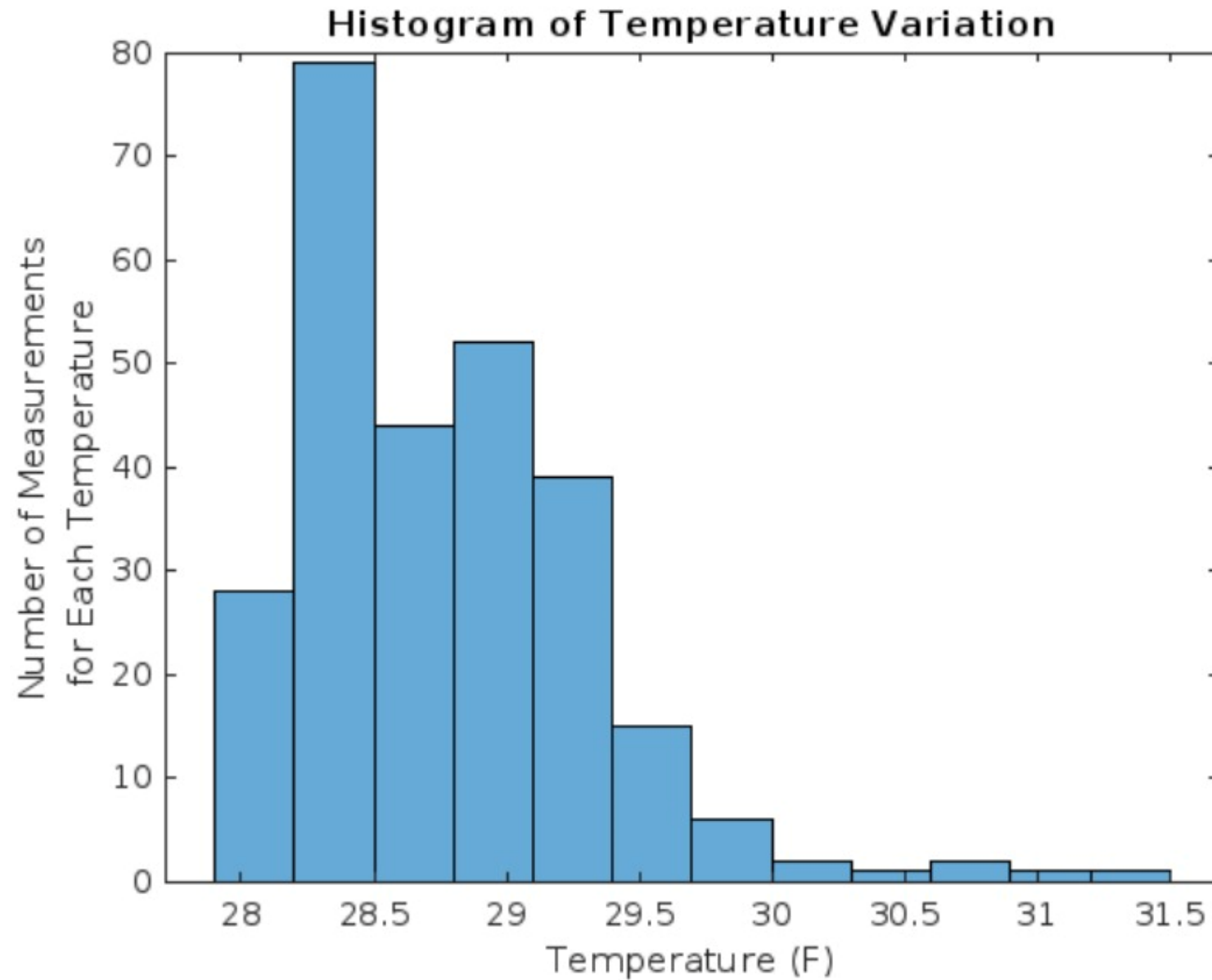
MATLAB Code

```
1 % Read temperature for the last 10 hours from a ThingSpeak channel and
2 % visualize temperature variations using the MATLAB HISTOGRAM function.
3
4 readChannelID = 2677119;
5
6 % Temperature Field ID
7 TemperatureFieldID = 1;
8
9 % Channel Read API Key
10 readAPIKey = 'JX0V8MBSN0YPWPTE';
11
12 tempF = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID,...
13 'NumMinutes',10*60, 'ReadKey',readAPIKey);
14
15 histogram(tempF);
16 xlabel('Temperature (F)');
17 ylabel('Number of Measurements\nnewline for Each Temperature');
18 title('Histogram of Temperature Variation');
```

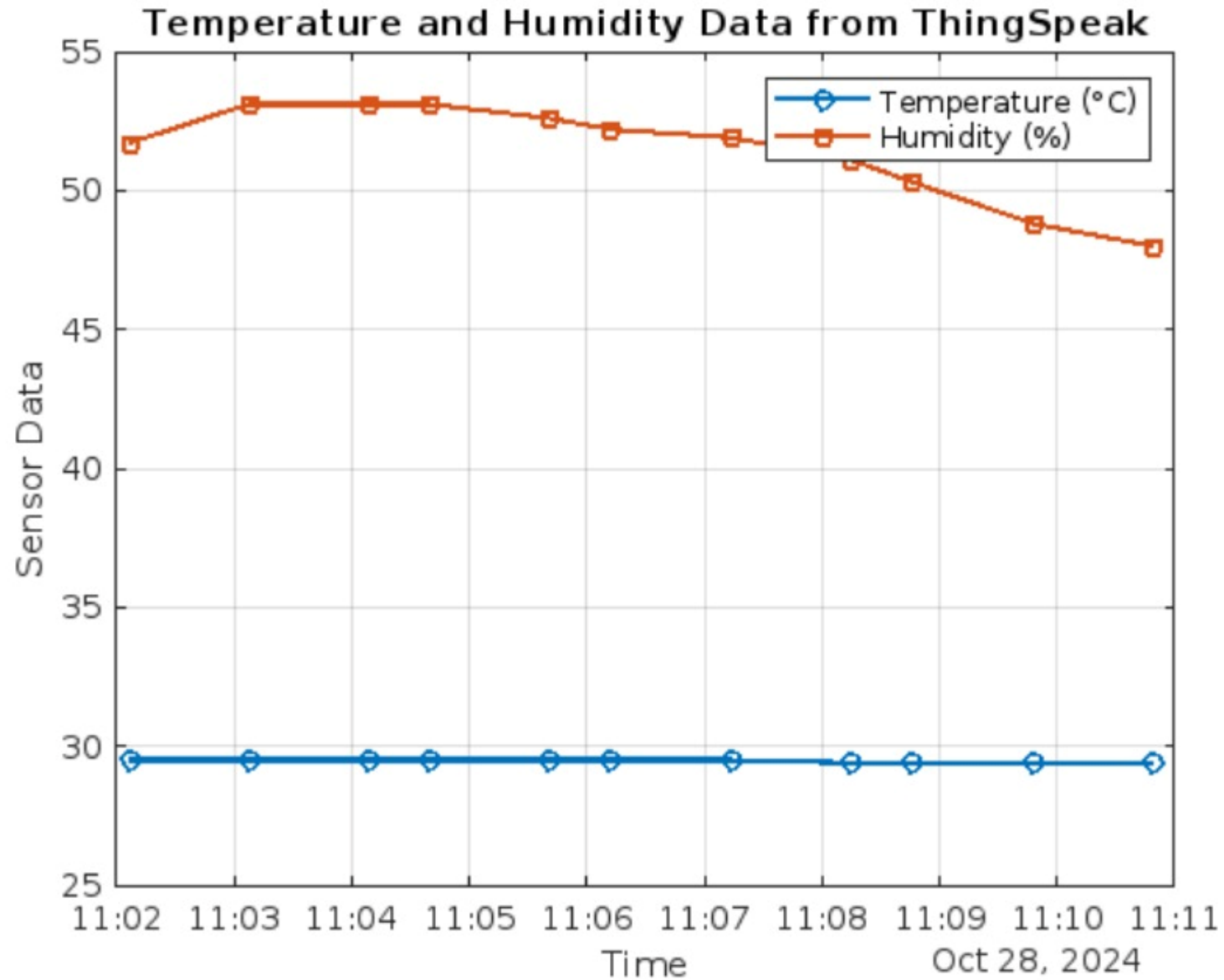
Save and Run

Save\*





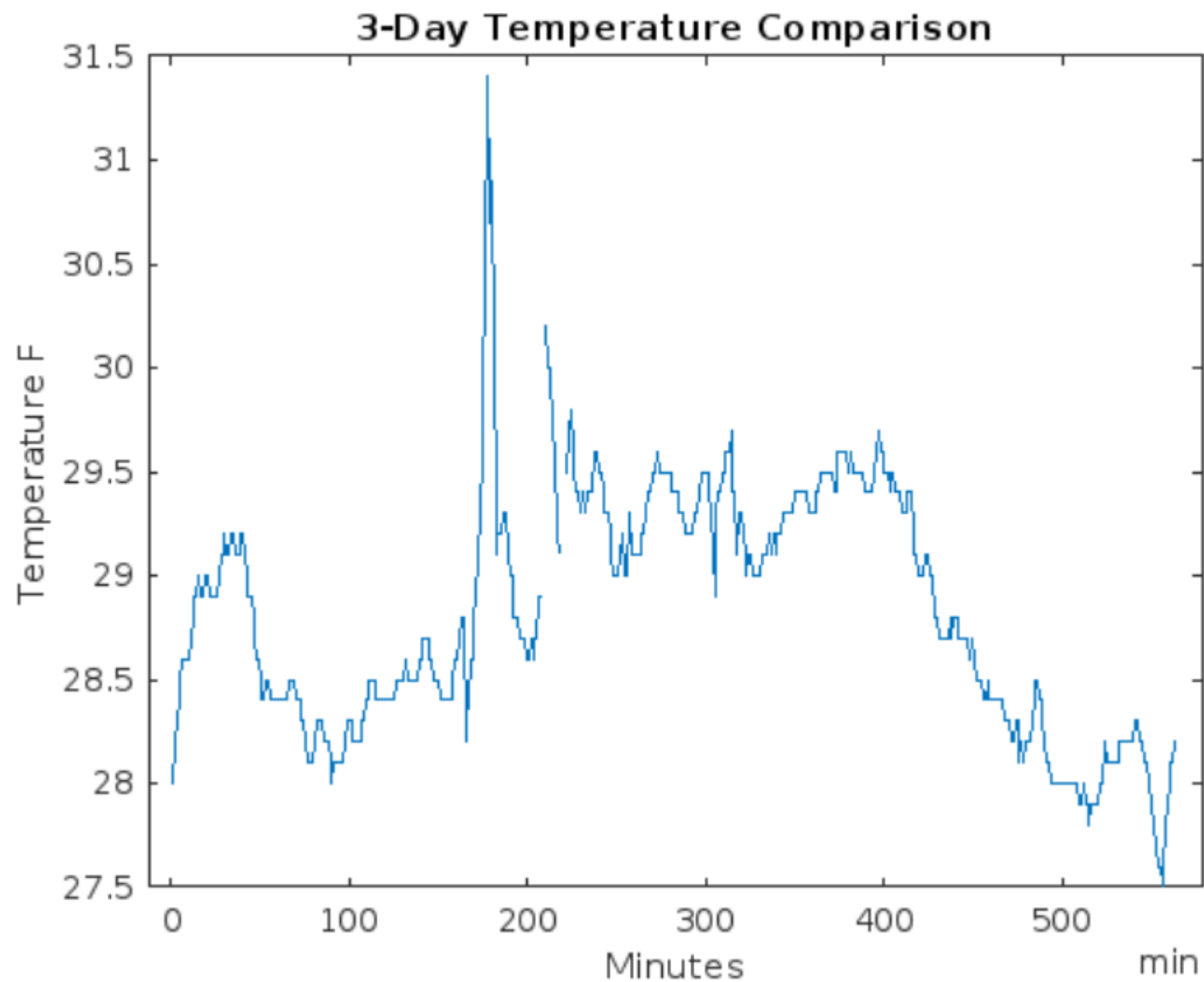
```
1 % Channel ID to read data from
2 readChannelID = 2677119;
3 % Temperature Field ID
4 TemperatureFieldID = 1;
5 % One day date range
6 oneDay = [datetime('yesterday') datetime('today')];
7
8 % Channel Read API Key
9 % If your channel is private, then enter the read API key between the '' below:
10 readAPIKey = 'JX0V8MBSN0YPWPTE';
11
12 % Read Temperature Data. Learn more about the THINGSPEAKREAD function by
13 % going to the Documentation tab on the right side pane of this page.
14 temperatureDay1 = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID, ...
15                                 'dateRange', oneDay, 'ReadKey',readAPIKey);
16 temperatureDay2 = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID, ...
17                                 'dateRange',oneDay-days(1),'ReadKey',readAPIKey);
18 temperatureDay3 = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID, ...
19                                 'dateRange', oneDay-days(2),'ReadKey',readAPIKey);
20
21 % Create array of durations
22 myTimes1 = minutes(1:length(temperatureDay1));
23 myTimes2 = minutes(1:length(temperatureDay2));
24 myTimes3 = minutes(1:length(temperatureDay3));
25
26 % Visualize the data
27 plot(myTimes1,temperatureDay1, myTimes2,temperatureDay2, myTimes3, temperatureDay3);
```

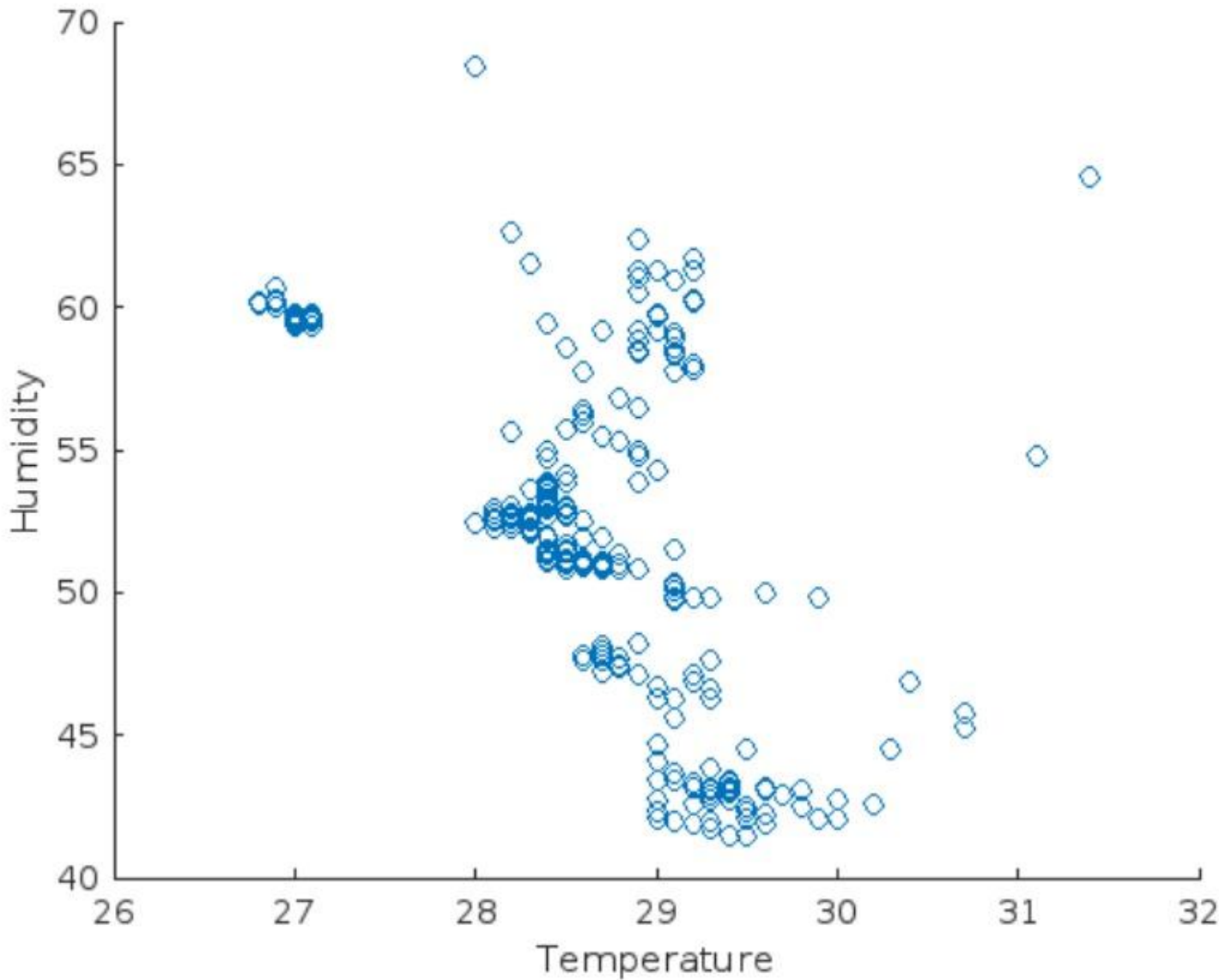




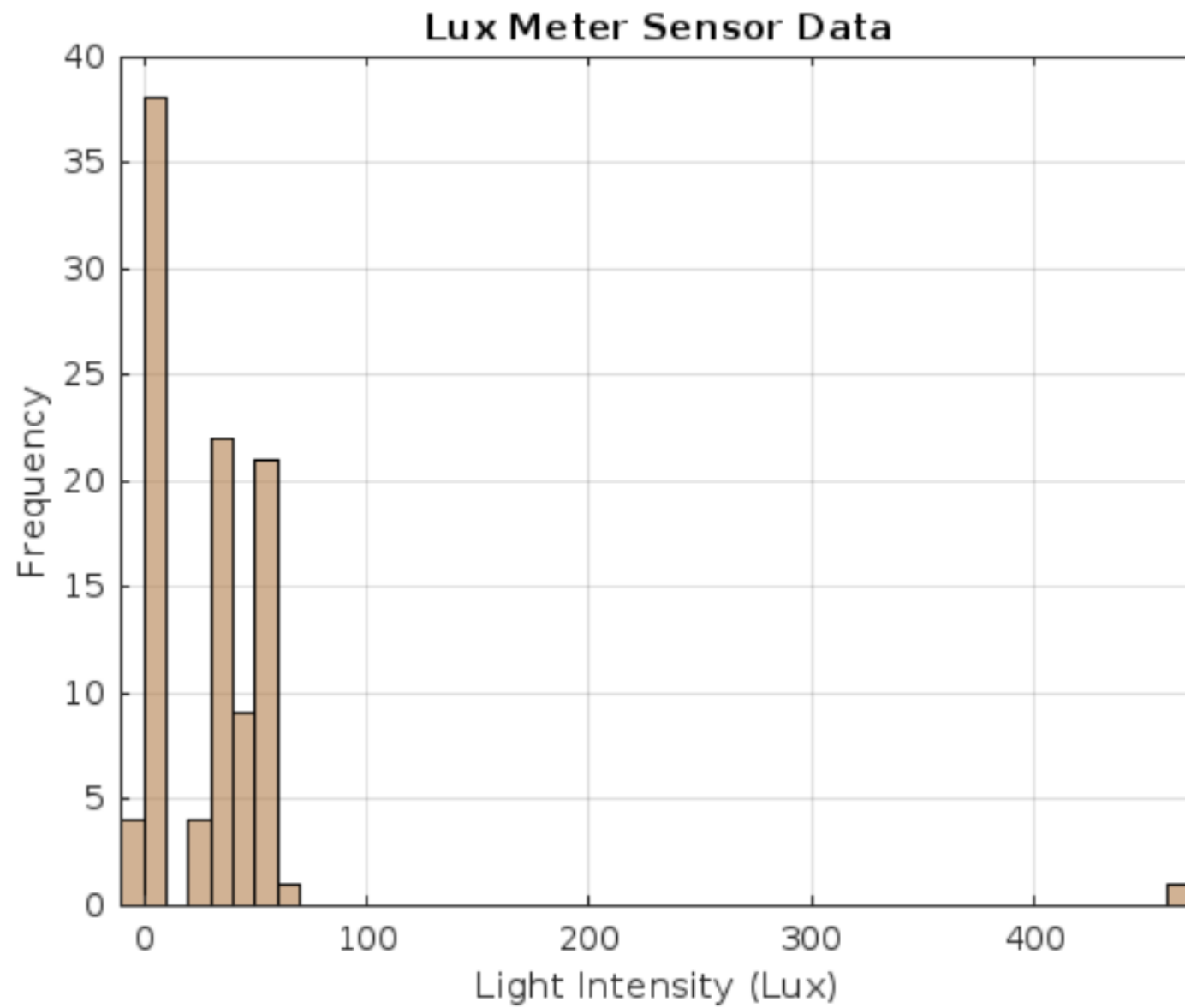


# carenuty

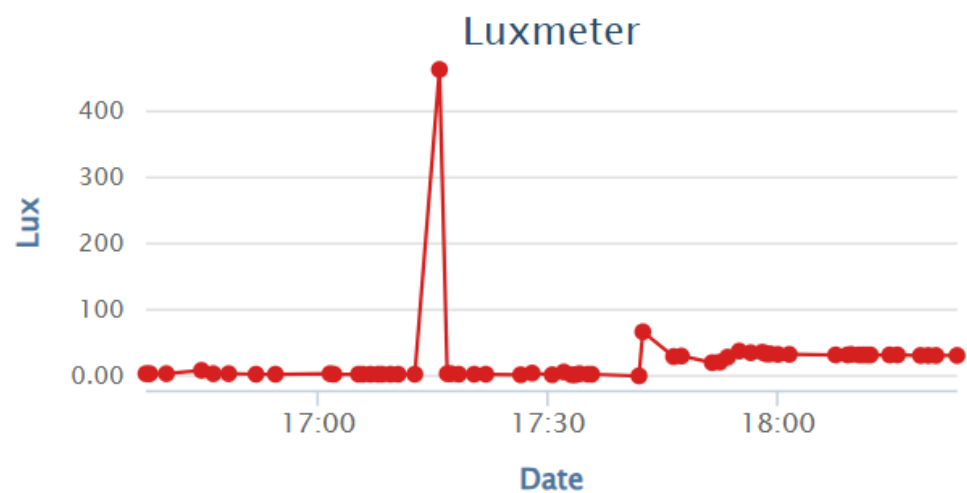




```
1 % Code| read lux meter sensor data from ThingSpeak and plot it in a histogram
2
3 % ThingSpeak Channel ID and Field ID to read lux data
4 readChannelID = 2717729;    % Replace with your ThingSpeak channel ID
5 fieldID = 1;                % Replace with the field ID where lux data is stored
6
7 % Read API Key (if the channel is private)
8 readAPIKey = 'N3TQTSOS0I4AZR1I'; % Replace with your Read API Key (leave blank if public)
9
10 % Fetch lux data from ThingSpeak
11 % Adjust the 'NumPoints' parameter to control the number of data points retrieved
12 luxData = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', 100, 'ReadKey', readAPIKey);
13
14 % Verify that data was retrieved successfully
15 if isempty(luxData)
16     error('No data retrieved. Please check your channel ID, field ID, and API key.');
```

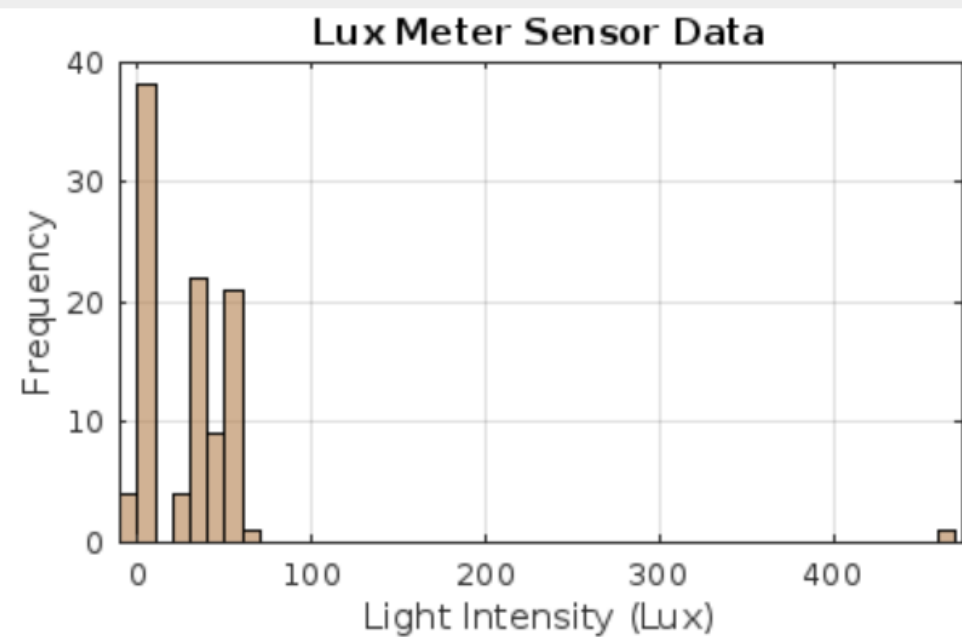


Field 1 Chart

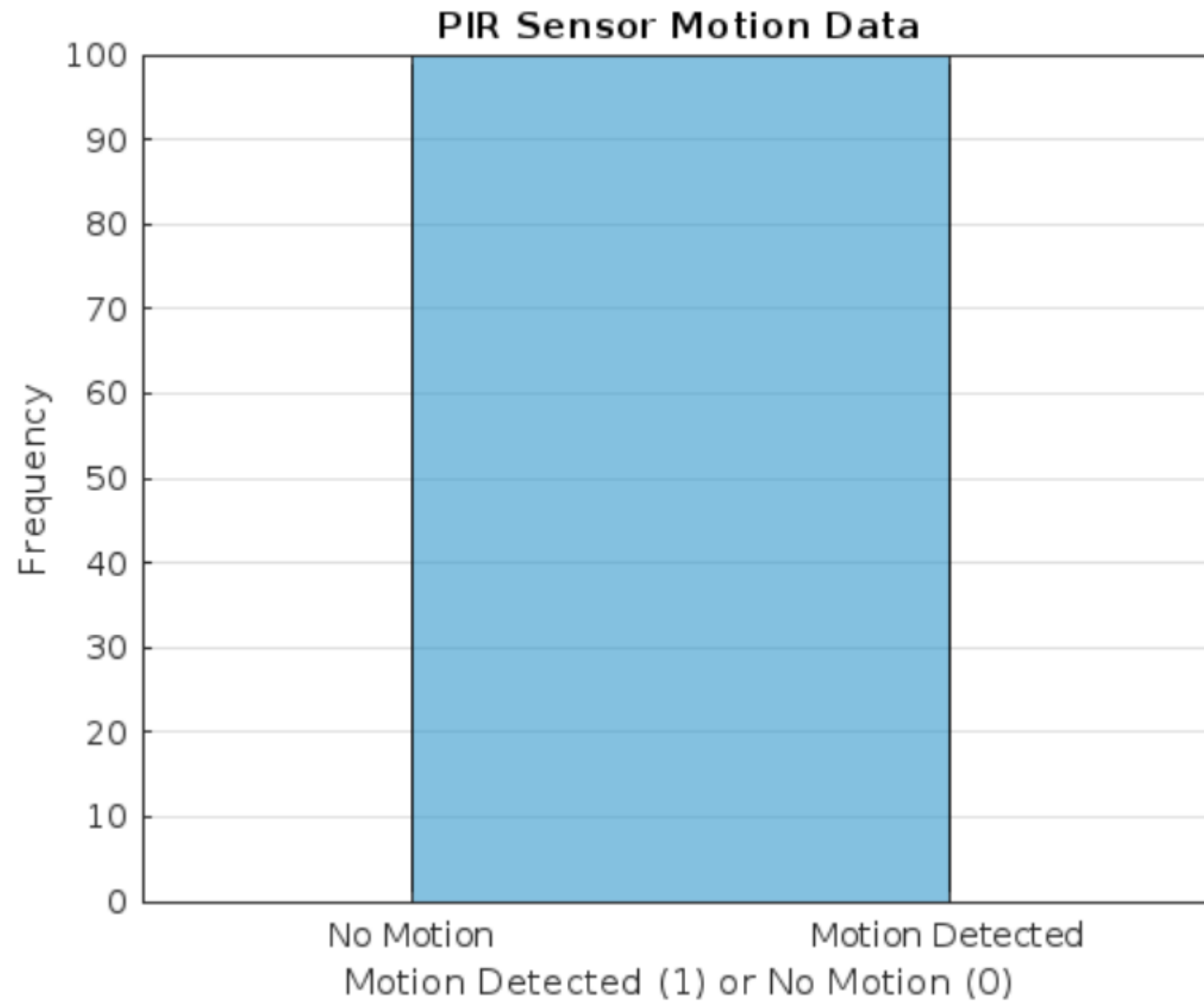


ThingSpeak.com

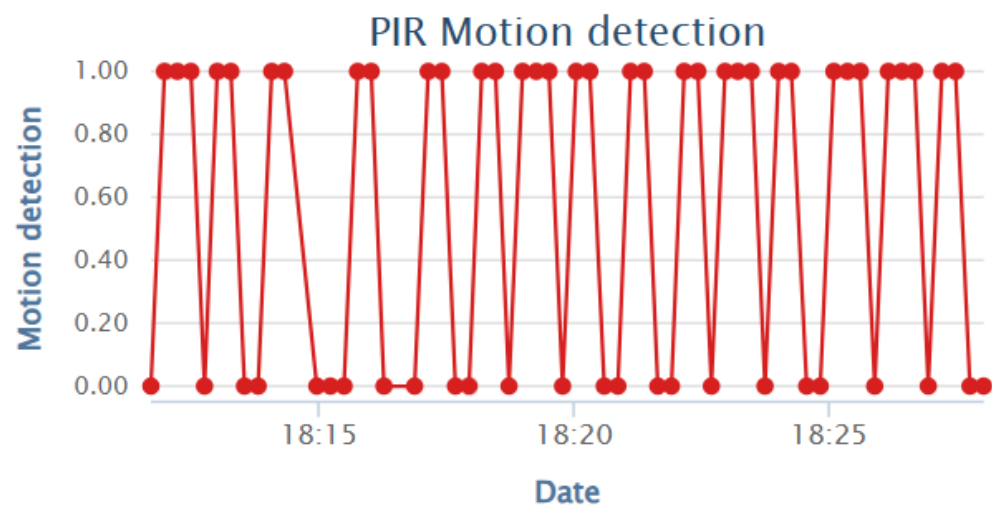
Histogram for Luxmeter



```
1 % Code to read PIR sensor data from ThingSpeak and plot it in a histogram
2 readChannelID = 2725062; % Replace with your ThingSpeak channel ID
3 fieldID = 1; % Replace with the field ID where PIR data is stored
4
5 % Read API Key (if the channel is private)
6 readAPIKey = 'WGUC96BZ0PKUB79C'; % Replace with your Read API Key (leave blank if public)
7
8 % Fetch data from ThingSpeak
9 % Adjust the 'NumPoints' parameter to control the number of data points retrieved
10 pirData = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', 100, 'ReadKey', readAPIKey);
11
12 % Verify that data was retrieved successfully
13 if isempty(pirData)
14     error('No data retrieved. Please check your channel ID, field ID, and API key.');
```

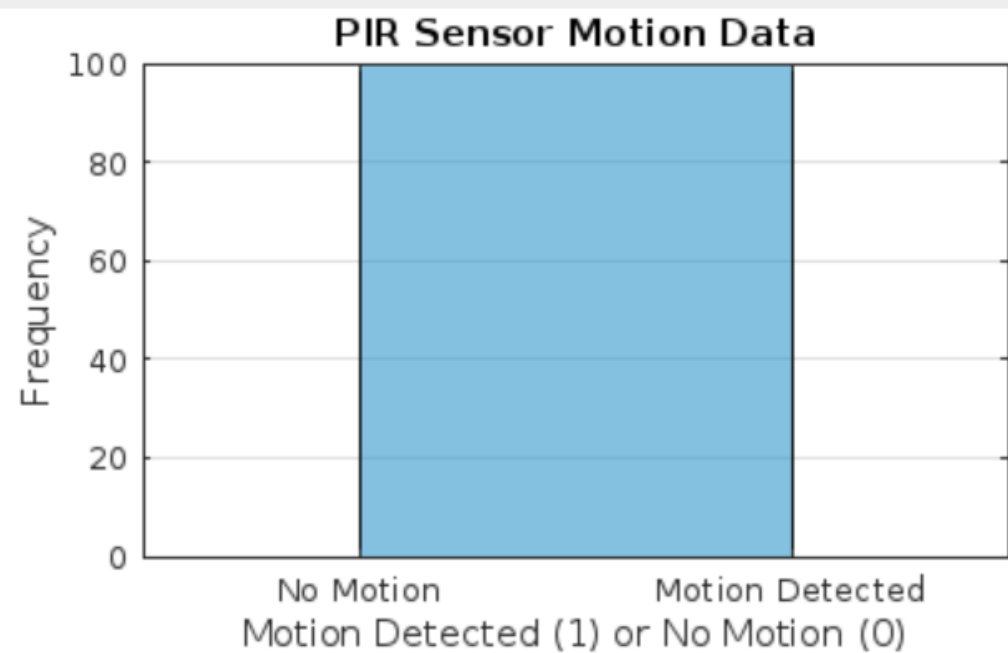


Field 1 Chart



ThingSpeak.com

Histogram for PIR





## CONCLUSION



- MATLAB offers a robust environment for data visualization, enabling the creation of a wide variety of graphs and plots, such as histograms, scatter plots, 3D surfaces, and heatmaps, which provide clear and detailed insights.
- While ThingSpeak's Field graphs are effective for basic real-time data monitoring, they are limited in customization and depth. By importing ThingSpeak data into MATLAB through channels, I am able to enhance visualizations to suit the dataset better, resulting in more tailored and informative views.

## NEXT STEPS



- Next, I will be exploring MATLAB's analytical capabilities, allowing me to perform deeper data analysis on the collected information.
- This shift to MATLAB analysis promises to unlock further insights, turning raw data into actionable information with advanced functions and calculations.
- This workflow, from data collection on ThingSpeak to enhanced visualization and analysis in MATLAB, showcases a powerful and complementary use of both platforms.

THE END