07-1 Pushing Data to the Cloud via Sheets

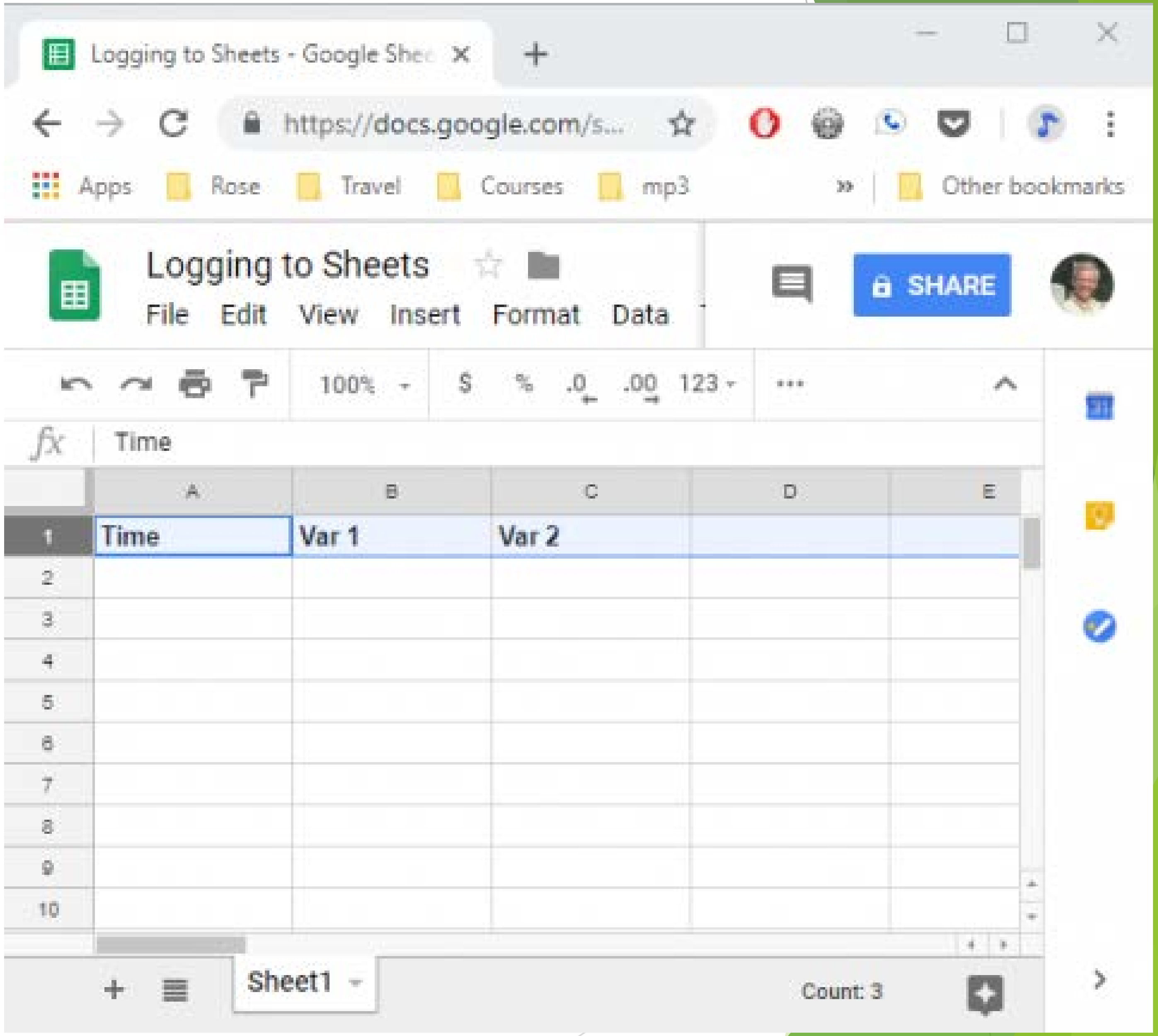
Using Google Sheets with the Bone

Steps to and from the Cloud

- Create a new Sheet
- Get Credentials
- Run the script
- Plot the data

Create a Sheet

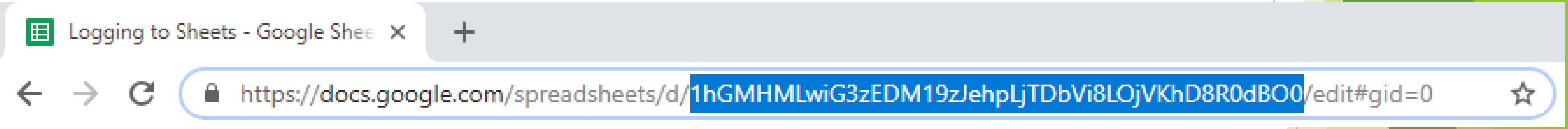
- Go to https://sheets.google.com
- Make a new sheet
- Rename it
- Add some column headings



fx Time

	A	B	C	D	E
1	Time	Var 1	Var 2		
2					
3					
4					
5					
6					
7					
8					
9					
10					

Note the sheetID

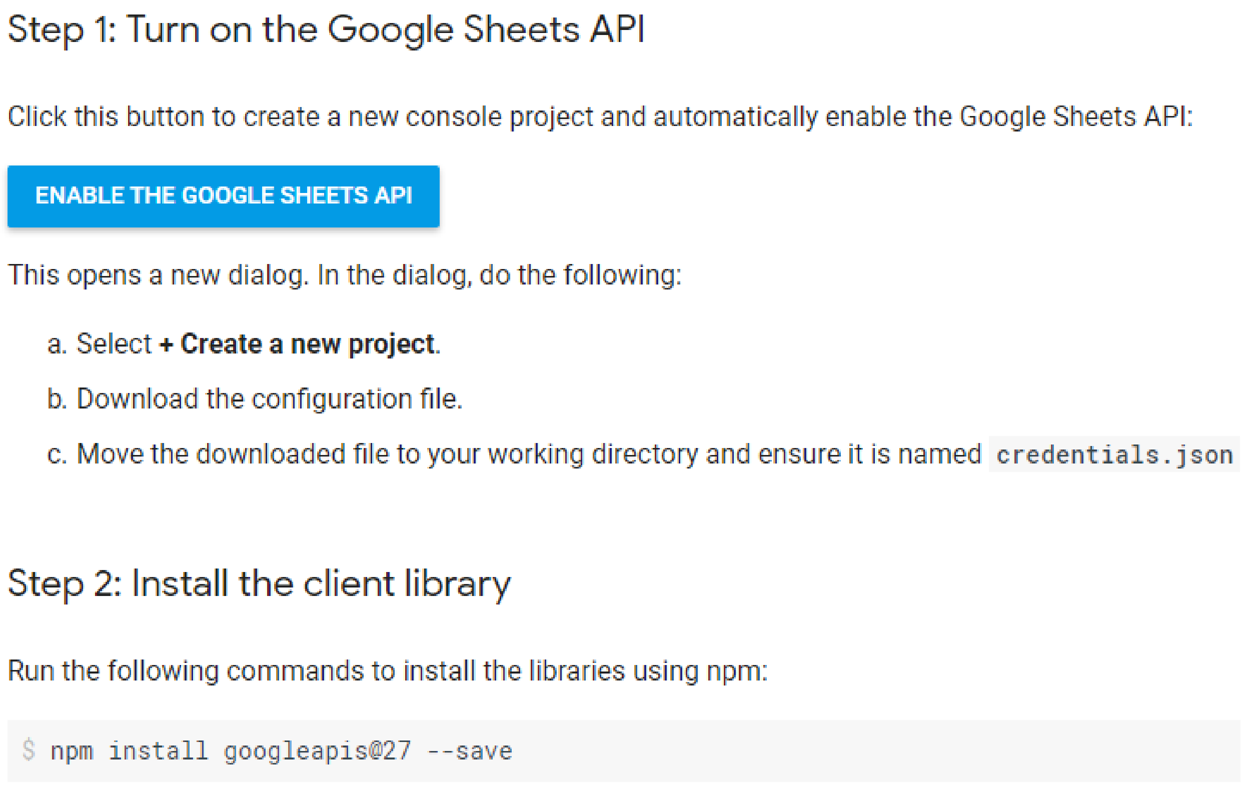


https://docs.google.com/spreadsheets/d/1hGMHMLwiG3zEDM19zJehpLjTDbVi8LOjVKhD8R0dBO0/edit#gid=0

Get Credentials

bone$ **cd exercises/iot/google/sheets**

- JavaScript: https://developers.google.com/sheets/api/quickstart/nodejs
- Python: https://developers.google.com/sheets/api/quickstart/python



Step 1: Turn on the Google Sheets API

Click this button to create a new console project and automatically enable the Google Sheets API:

This opens a new dialog. In the dialog, do the following:

a. Select **+ Create a new project**.

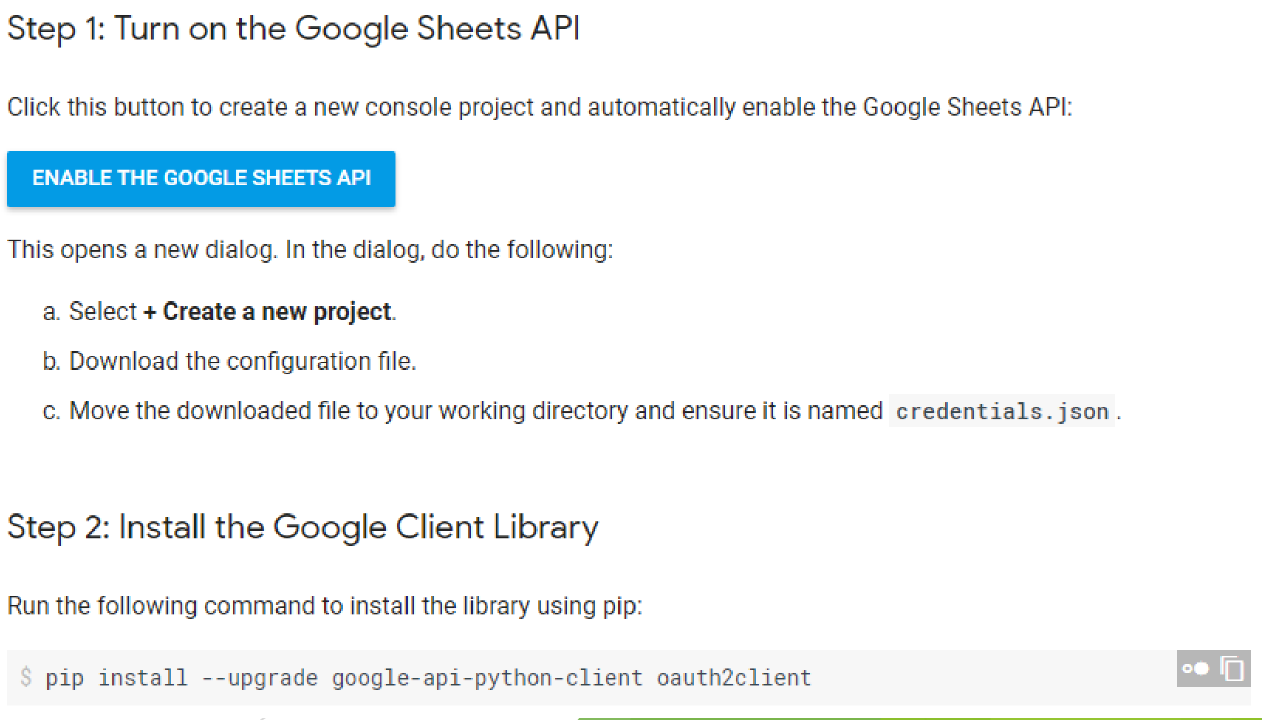
b. Download the configuration file.

c. Move the downloaded file to your working directory and ensure it is named `credentials.json`

Step 2: Install the client library

Run the following commands to install the libraries using npm:

```
$ npm install googleapis@27 --save
```



Step 1: Turn on the Google Sheets API

Click this button to create a new console project and automatically enable the Google Sheets API:

This opens a new dialog. In the dialog, do the following:

a. Select **+ Create a new project**.

b. Download the configuration file.

c. Move the downloaded file to your working directory and ensure it is named `credentials.json`.

Step 2: Install the Google Client Library

Run the following command to install the library using pip:

```
$ pip install --upgrade google-api-python-client oauth2client
```

Get Credentials

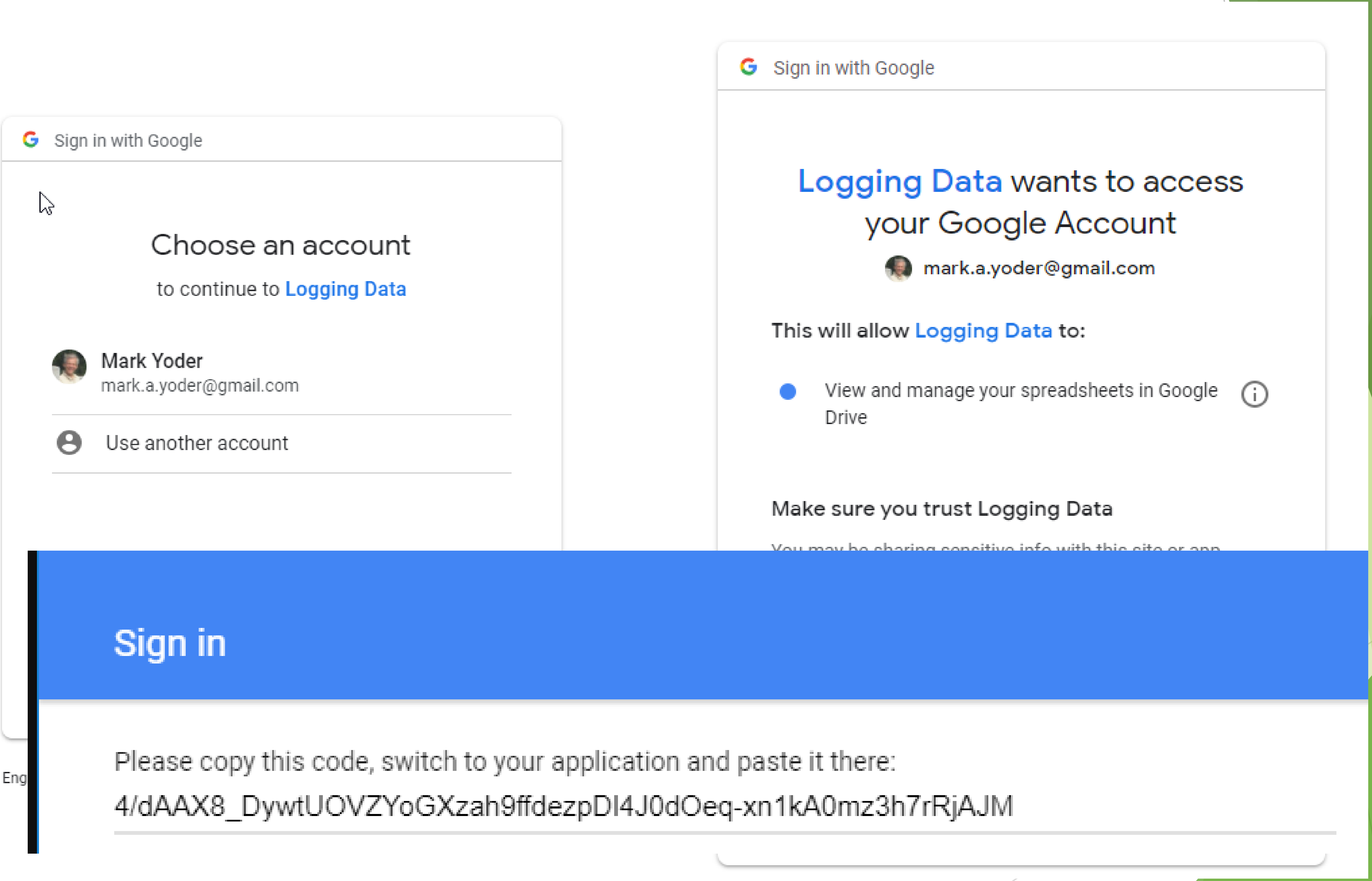
- You will download the file **credentials.json**
- Put it in the sheets directory

Run the example

- To log 3.14 and 10 run:

```
bone$ ./demo.js 3.14 10

Authorize this app by visiting this url:
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&response_type=code&client_id=834314057708-fucjrsdnm1nvmvvbqkfr9008g63d7k2r.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob

Enter the code from that page here:
```

- Paste the URL into your browser and follow the directions.



Sign in with Google

Choose an account

to continue to Logging Data

Mark Yoder
mark.a.yoder@gmail.com

Use another account

Eng

Sign in with Google

Logging Data wants to access your Google Account

mark.a.yoder@gmail.com

This will allow Logging Data to:

- View and manage your spreadsheets in Google Drive

Make sure you trust Logging Data

You may be sharing sensitive info with this site or app

Sign in

Please copy this code, switch to your application and paste it there:

4/dAAX8_DywtUOVZYoGXzah9ffdezpDI4J0dOeq-xn1kA0mz3h7rRjAJM

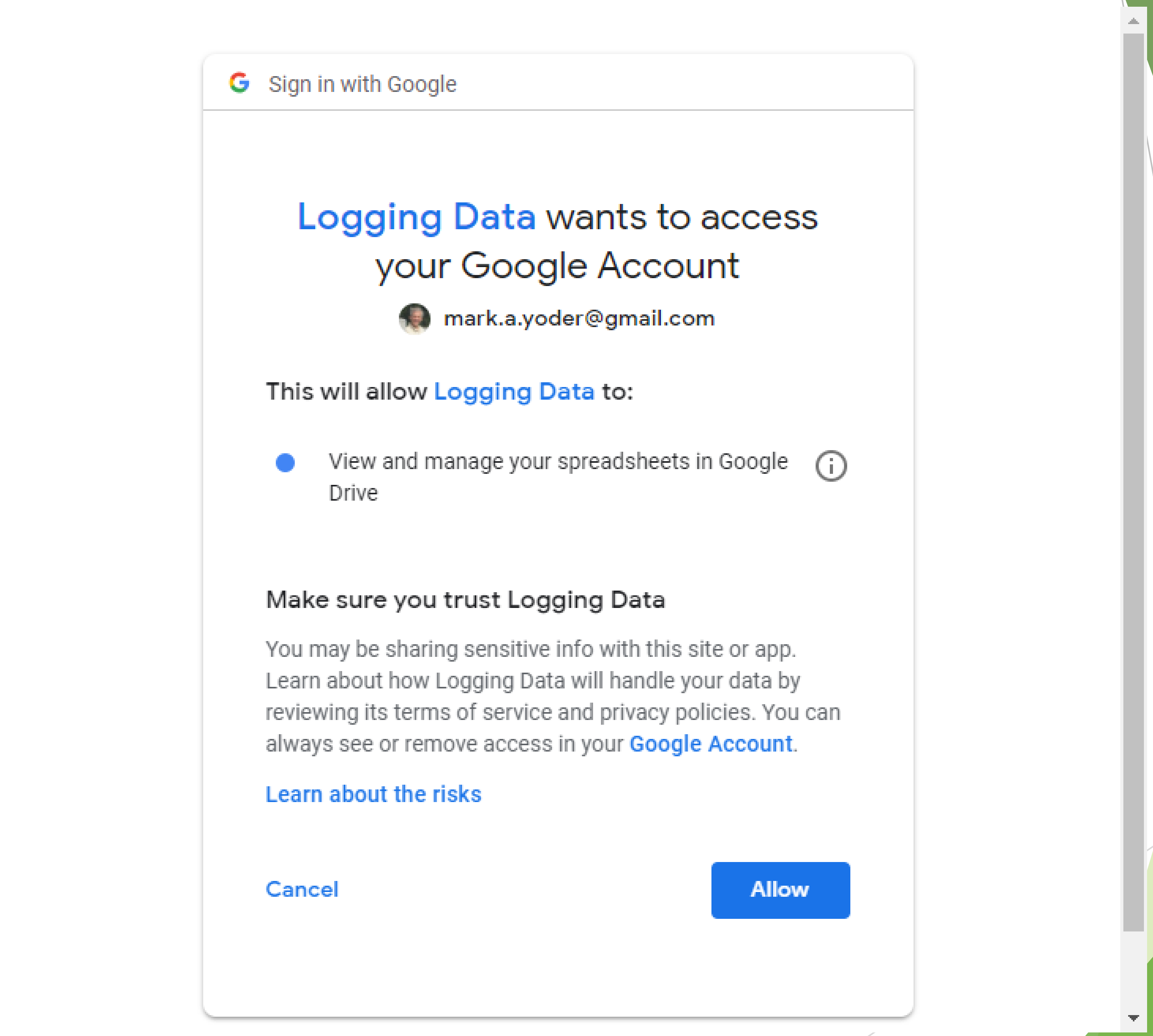
Run the example

- To log 3.14 and 10 run:

```
bone$ ./demo.js 3.14 10

Authorize this app by visiting this url:
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&response_type=code&client_id=834314057708-fucjrsdnm1nvmvvbqkfr9008g63d7k2r.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob

Enter the code from that page here:
```

- Paste the URL into your browser and follow the directions.

Run the e



Sign in with Google

Logging Data wants to access your Google Account

mark.a.yoder@gmail.com

This will allow Logging Data to:

- View and manage your spreadsheets in Google Drive

Make sure you trust Logging Data

You may be sharing sensitive info with this site or app. Learn about how Logging Data will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks

Cancel Allow

Run the example

- To log 3.14 and 10 run:

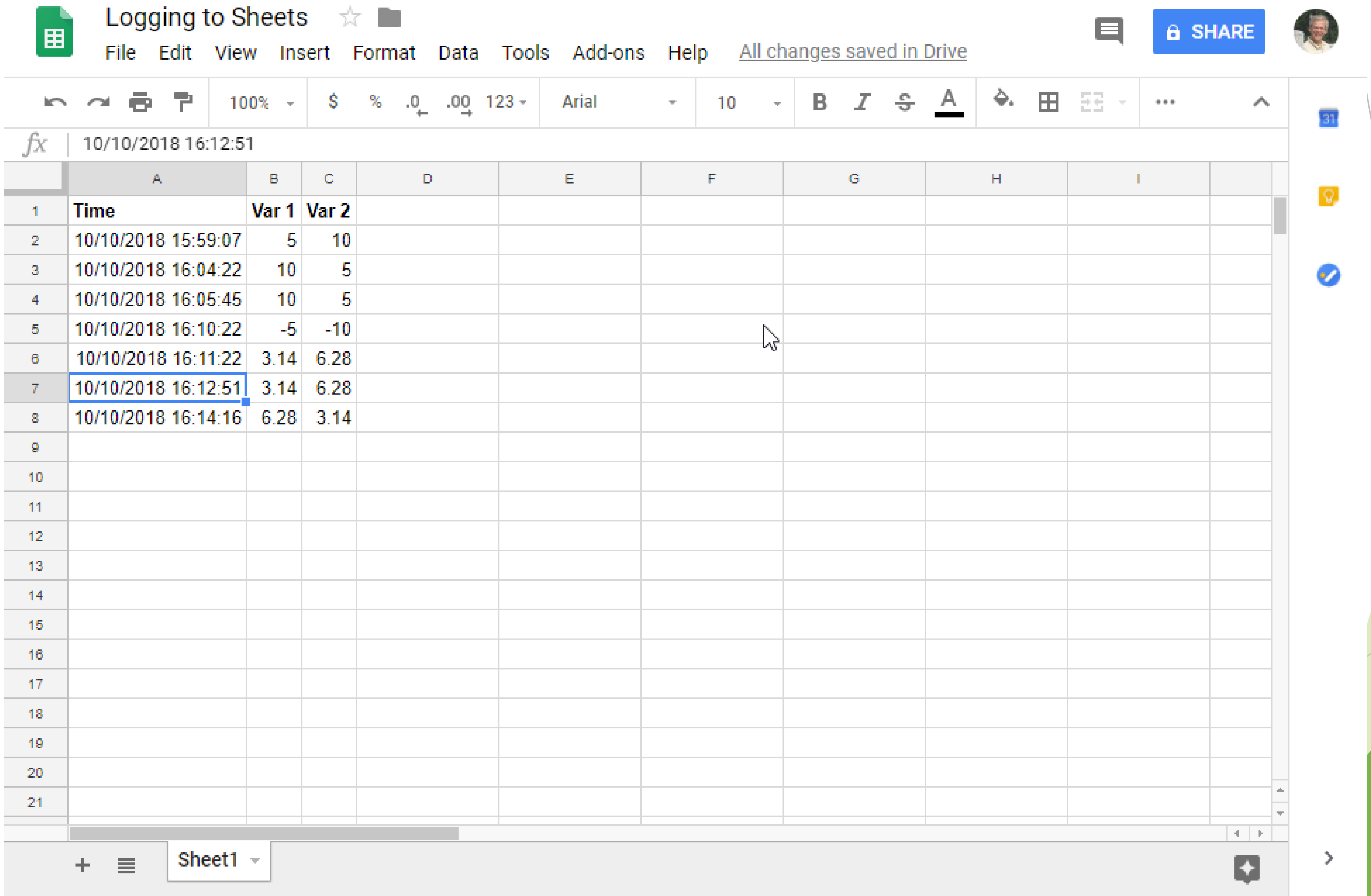
```
bone$ ./demo.js 3.14 10

Authorize this app by visiting this url:
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&response_type=code&client_id=834314057708-fucjrsdnm1nvmvvbqkfr9008g63d7k2r.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob

Enter the code from that page here:
```

- Paste the URL into your browser and follow the directions.

token.json

- A file **token.json** is created
- Don’t have to authorize again

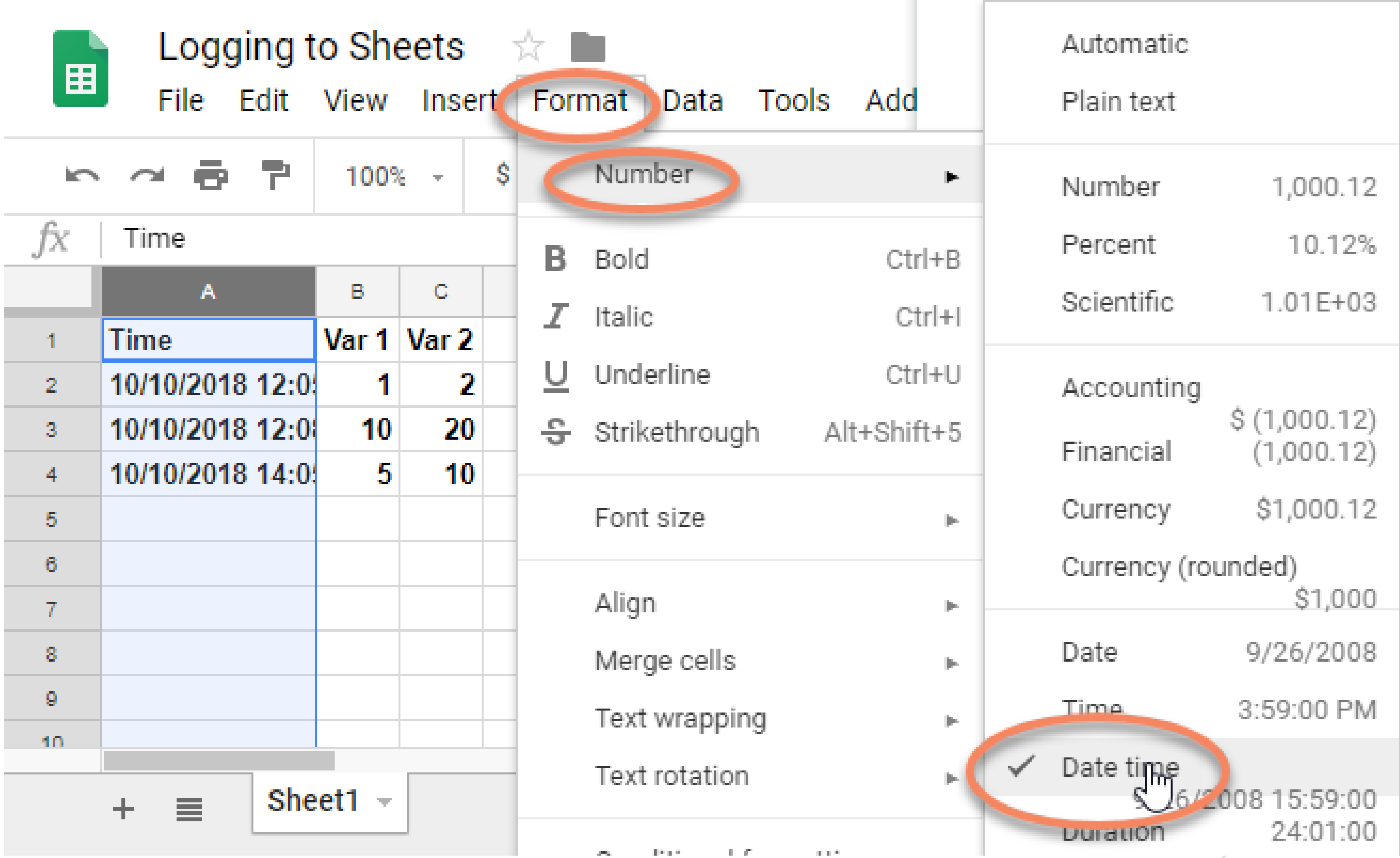
View the Data



fx 10/10/2018 16:12:51

Time	Var 1	Var 2
10/10/2018 15:59:07	5	10
10/10/2018 16:04:22	10	5
10/10/2018 16:05:45	10	5
10/10/2018 16:10:22	-5	-10
10/10/2018 16:11:22	3.14	6.28
10/10/2018 16:12:51	3.14	6.28
10/10/2018 16:14:16	6.28	3.14

Plot the data



Logging to Sheets

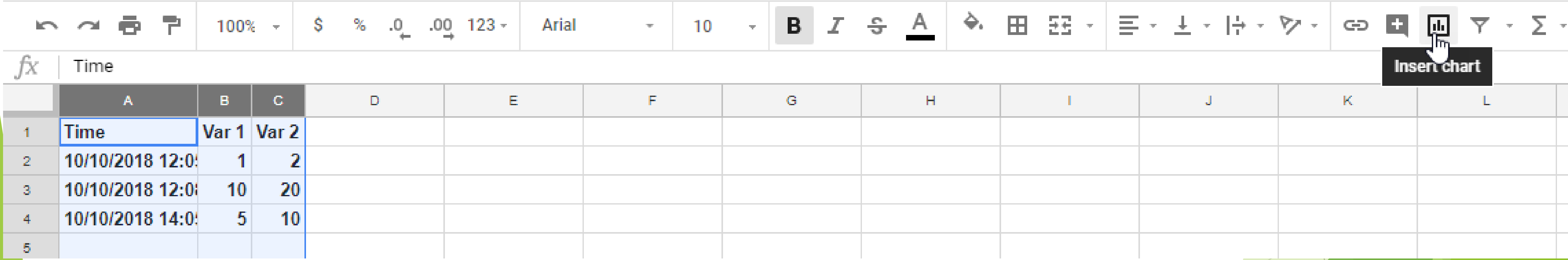
fx Time

	A	B	C
1	Time	Var 1	Var 2
2	10/10/2018 12:0	1	2
3	10/10/2018 12:0	10	20
4	10/10/2018 14:0	5	10

Format > Number > ✓ Date time

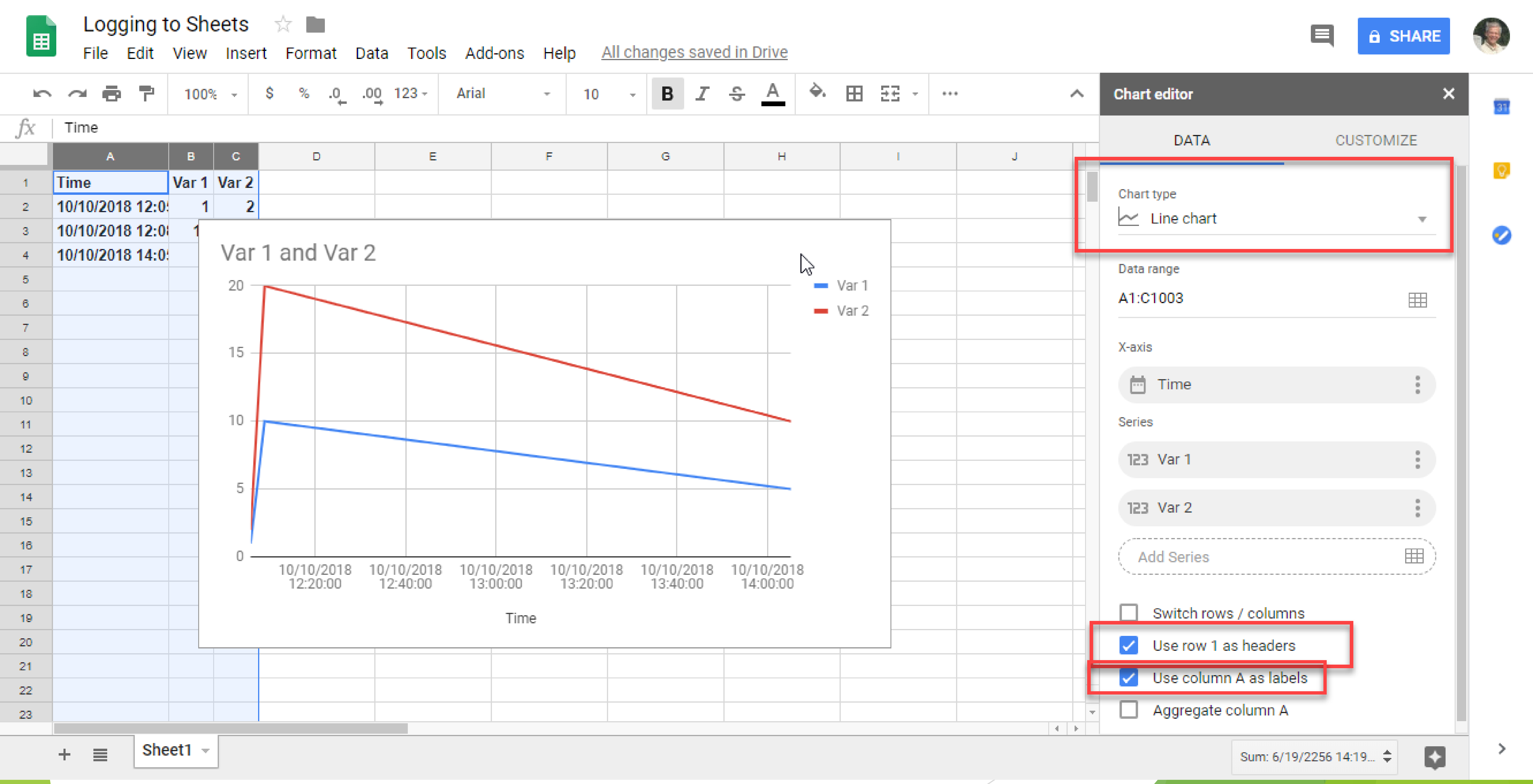
Plot the Data

- Select all columns
- Select **Insert chart**



fx Time

	A	B	C
1	Time	Var 1	Var 2
2	10/10/2018 12:0:	1	2
3	10/10/2018 12:0:	10	20
4	10/10/2018 14:0:	5	10
5			



fx Time

	A	B	C
1	Time	Var 1	Var 2
2	10/10/2018 12:0!	1	2
3	10/10/2018 12:0!	1	
4	10/10/2018 14:0!		

Chart type
Line chart

Data range
A1:C1003

X-axis
Time

Series
Var 1
Var 2

□ Switch rows / columns

☑ Use row 1 as headers

☑ Use column A as labels

□ Aggregate column A

Plot the Data

- Your data should be plotted
- Running demo.js again will add more data

The code - JavaScript

- Lot’s of code for authentication

```
#!/usr/bin/env node

// From:
https://developers.google.com/sheets/api/quickstart/nodejs#step_3_set_up_the_sample

const fs = require('fs');

const util = require('util');

const readline = require('readline');

const {google} = require('googleapis');


const sheetID = '1hGMHMLwiG3zEDM19zJehpLjTDbVi8LOjVKhD8R0dBO0';


// If modifying these scopes, delete credentials.json.

const SCOPES = ['https://www.googleapis.com/auth/spreadsheets'];

const TOKEN_PATH = 'token.json';
```

The code - JavaScript

```
function recordTemp(auth) {
  const sheets = google.sheets({version: 'v4', auth});
  var date = new Date();
    // Read the variable data
  var var1 = process.argv[2];
  var var2 = process.argv[3];
  console.log('Got: ' + var1 + ", " + var2);
```

The code - JavaScript

```
sheets.spreadsheets.values.append({
  spreadsheetId: sheetID,
  range: 'A2',
  // How the input data should be interpreted.
  valueInputOption: 'USER_ENTERED’,
  resource: {
    values: [   // getTime returs ms.  Convert to days.
        [
          date.getTime()/1000/60/60/24 + 25569 - 4/24,
          var1,
          var2
        ]
      ]
```

The code - JavaScript

```
  sheets.spreadsheets.values.append({
…
}, (err, res) => {
    if (err) return console.log('The API returned an error: '
+ err);
    console.log("res: " + util.inspect(res.data.tableRange));
  });


}
```

The code - Python

```
    # Call the Sheets API
    # Compute a timestamp and pass the first two arguments
    values = [ [time.time()/60/60/24+ 25569 - 4/24, sys.argv[1], sys.argv[2]]]
    body = { 'values': values }
    result =
service.spreadsheets().values().append(spreadsheetId=SPREADSHEET_ID,
                            range=RANGE_NAME,
                            #  How the input data should be interpreted.
                            valueInputOption='USER_ENTERED',
                            # How the input data should be inserted.
                            # insertDataOption='INSERT_ROWS'
                            body=body
                            ).execute()
```