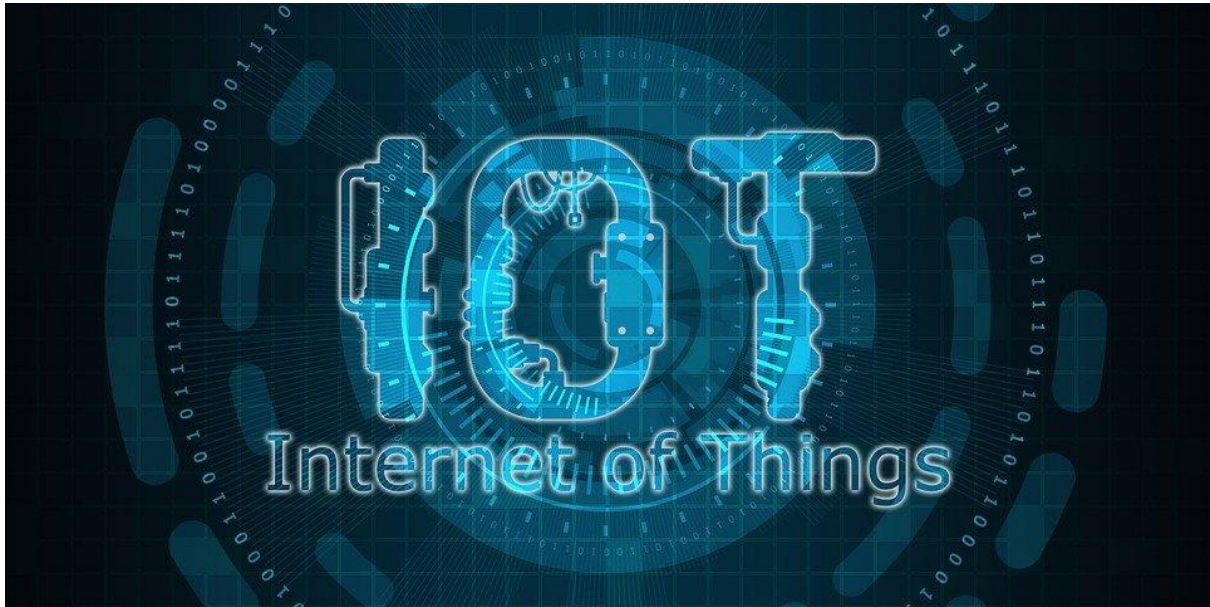


# IoT Group Assignment Report



**Course and Unit Details:2020-HS1-SWE30011-IoT Programming-H1**

**Group 12 Tutor: Shalmoly Mondal**

Group Members:

YiChen Guan \*\*\*\*\*

Bong Siaw Zhen \*\*\*\*\*

Zhong Wang \*\*\*\*\*

## Executive Summary

This report specifically describes how we use the Google sheet API to upload data, and automate data management, data analysis, data visualization and send an email to the executor by using Google apps script API. After the executor receives the notification, the executor will download the data from the google drive cloud storage servers and control the actuator based on the data.

## Table of Contents:

### 1.Sensing system with Code snippet

- 1.1 Arduino settings
- 1.2 Code for Arduino getting data
- 1.3 Python read data from Arduino and upload data to database
- 1.4 Python gets data from database, performs data analysis then operates actuator at third place.

### 2. API details

- 2.1 Google sheet Api V4
  - 2.1.1 What is Google sheet Api
  - 2.1.2 How do we set up and use google sheet api
  - 2.1.3 Use Google sheet Api for data upload in the project
  - 2.1.4 Use Google sheet Api for data download in the project
- 2.2 Google apps script api
  - 2.2.1 What is Google apps script and Google apps script api
  - 2.2.2 How do we use google apps script api
  - 2.2.3 What we do by using this in project

### 3 Statistical data analysis

- 3.1 Introduction
- 3.2 Calculation Formulas
- 3.3 Explanation with code:

### 4 Data visualization

- 4.1 Introduction
  - 4.2.1 Create Line chart and Regression analysis chart for different users based on their temperature data :
  - 4.2.2 Set the mean value, standard deviation value and variance value above the charts then set background color to identify the user (green background: Zhong's data. yellow background: Yichen's data):

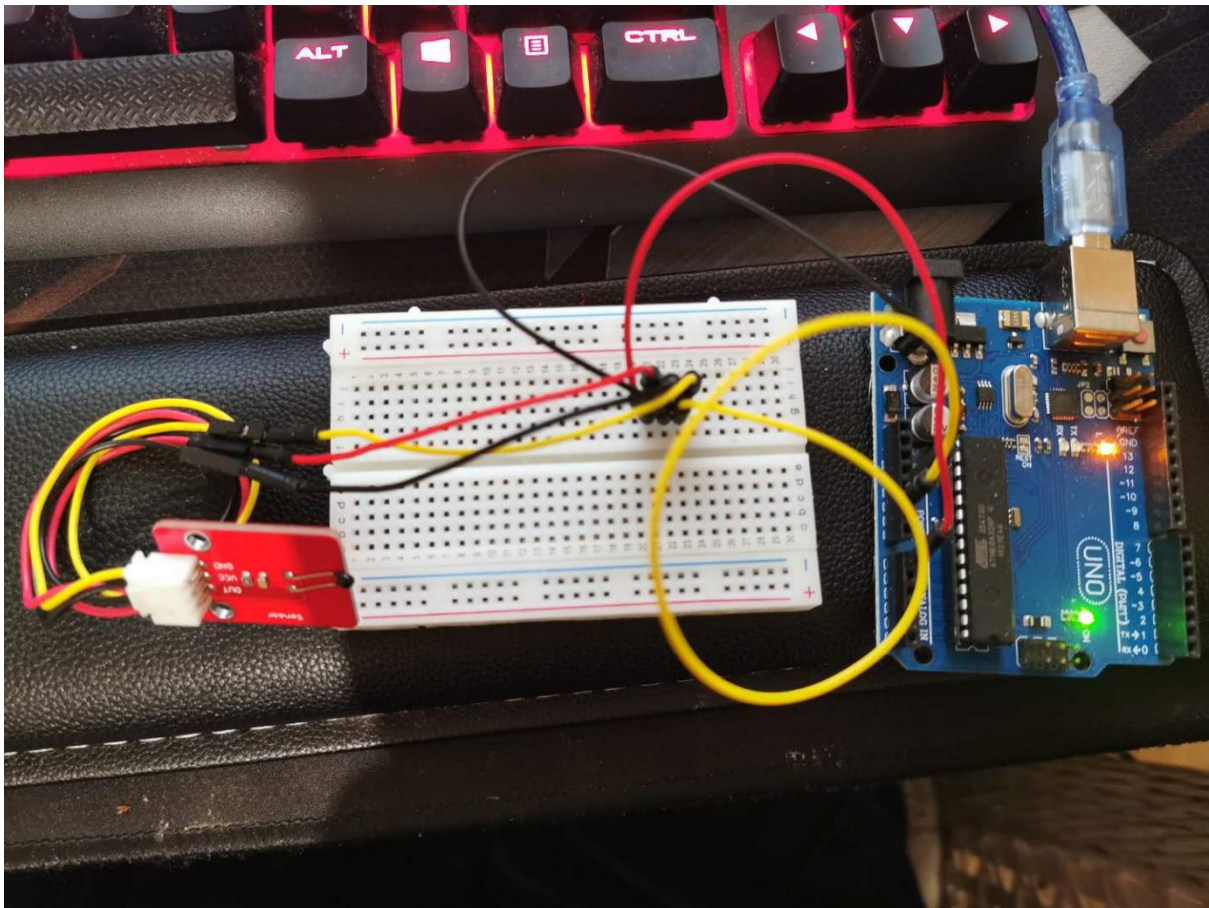
### 5 Implementation and User Manual

## 6. Reference

## 7. Appendix

## 1.Sensing system with Code snippet (Yichen and Zhong)

### 1.1 Arduino settings (Yichen and Zhong)



Arduino compatible temperature sensor

GND-----GND

VCC-----5v

S-----analog A0

## 1.2 Code for Arduino getting data (Yichen and Zhong)

```
#include <math.h>
double Thermister(int RawADC) {
double Temp;
Temp = log(((10240000/RawADC) - 10000));
Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp )) * Temp );
Temp = Temp - 273.15;
return Temp;
}
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println(Thermister(analogRead(0)));
  delay(500);
}
```

## 1.3 Python read data from Arduino and upload data to database (Yichen and Zhong)

```
1 import serial
2 import gspread
3 from oauth2client.service_account import ServiceAccountCredentials
4 auth_json_path = 'zhong.json'
5 gss_scopes = ['https://spreadsheets.google.com/feeds']
6
7 credentials = ServiceAccountCredentials.from_json_keyfile_name(auth_json_path,gss_scopes)
8 gss_client = gspread.authorize(credentials)
9
10 spreadsheet_key = '1sv2Tw9opLRLi19SFLdihbLhn1tBBIDc_eQoVa3V3cnM'
11
12 sheet = gss_client.open_by_key(spreadsheet_key).sheet1
13 device="/dev/ttyACM0"
14 arduino=serial.Serial(device,9600)
15
16 for num in range(0,10):
17     data=arduino.readline()
18     newdata= float(data)
19     listtitle=["Zhong",newdata]
20     sheet.append_row(listtitle) |
21
22
23
```



## 1.4 Python gets data from database, performs data analysis then operates actuator at third place. (Bong Siaw Zhen)

```
import gspread
import pyfirmata
import time
from oauth2client.service_account import ServiceAccountCredentials
auth_json_path = 'zhen.json'
gss_scopes = ['https://spreadsheets.google.com/feeds']
credentials = ServiceAccountCredentials.from_json_keyfile_name(auth_json_path, gss_scopes)
gss_client = gspread.authorize(credentials)
spreadsheet_key = '1sv2Tw9opLRLi19SFLdihbLhn1tBBIDc_eQoVa3V3cnM'
sheet = gss_client.open_by_key(spreadsheet_key).sheet1
value_listdata = sheet.get_all_values()

zhong = []
chen = []
z = 0
c = 0

zYellow = 2 #Zhong's yellow led
zRed = 4 #Zhong's red led
cRed = 6 #Chen's red led
cYellow = 8 #Chen's yellow led

port = '/dev/cu.usbmodem14201'
board = pyfirmata.Arduino(port)

def blink_led(x,y,z):
    if(x>=5):
        board.digital[y].write(0)
        board.digital[z].write(1)
        time.sleep(1)
    elif(x<5):
        board.digital[y].write(1)
        board.digital[z].write(0)
        time.sleep(1)

for a in range(0,len(value_listdata)):
    name = value_listdata[a][0]
    data = value_listdata[a][1]
    if(name == "Zhong"):
        zhong.append(float(data))
        if(float(data)<=25):
            blink_led(z,zYellow,zRed)
        if(float(data)>25):
            z+=1
            blink_led(z,zYellow,zRed)
    if(name == "yichen"):
        chen.append(float(data))
        if(float(data)<=25):
            blink_led(c,cYellow,cRed)
        if(float(data)>25):
            c+=1
            blink_led(c,cYellow,cRed)

print("Zhong's temperature > 25 have",z)
print(zhong)
print("yichen's temperature > 25 have",c)
print(chen)
```

## 2. API details (Yichen)

About demonstrated understanding of API, we use Google sheet api and http request base on apps script restful api.

### 2.1 Google sheet Api V4

#### 2.1.1 What is Google sheet Api

Google sheet api allows the user to read, write and format data in sheets by programming.

#### 2.1.2 How do we set up and use google sheet api


Step 1 : We create a new cloud platform project.

The screenshot displays the Google Cloud Platform (GCP) dashboard. At the top, a blue navigation bar contains the 'Google Cloud Platform' logo, a 'Select a project' dropdown, a search bar, and user account controls. Below the navigation bar, a 'Dashboard' section features a message: 'To view this page, select a project.' with links for 'SELECT PROJECT' and 'CREATE PROJECT'. The 'Select a recent project' section lists five projects: 'IOTsheet', 'apps', 'My Project', 'My Project 18810', and 'My First Project'. Each project card shows its ID, organization, and last access time. At the bottom, the 'New Project' form is visible, including a warning about project quotas, a 'Project name' field with a dropdown showing 'My Project 67276', a 'Project ID' field with the value 'molten-gasket-280108', a 'Location' dropdown set to 'No organization', and 'CREATE' and 'CANCEL' buttons.

Step 2 : Enable the Google sheet API

Google Cloud Platform My Project 67276 Search products and resources

API Library



## Google Sheets API

Google

The Sheets API gives you full control over the content and appearance of your spreadsheet data.

[ENABLE](#) [TRY THIS API](#)

Type

[APIs & services](#)

Last updated

12/10/19, 11:36 AM

Category

Overview

Reads and writes Google Sheets.

About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through

### Step 3: Create a credential for Service Accounts

Google Cloud Platform My Project 67276 Search products and resources

APIs & Services Google Sheets API

Credentials [+ CREATE CREDENTIALS](#) [DELETE](#)

#### Credentials compatible with this API

To view all credentials or create new credentials visit [Credentials in APIs & Services](#)

Remember to configure the OAuth consent screen with information about your application. [CONFIGURE CONSENT SCREEN](#)

#### OAuth 2.0 Client IDs

<input type="checkbox"/>	Name	Creation date ↓	Type	Client ID
No OAuth clients to display				

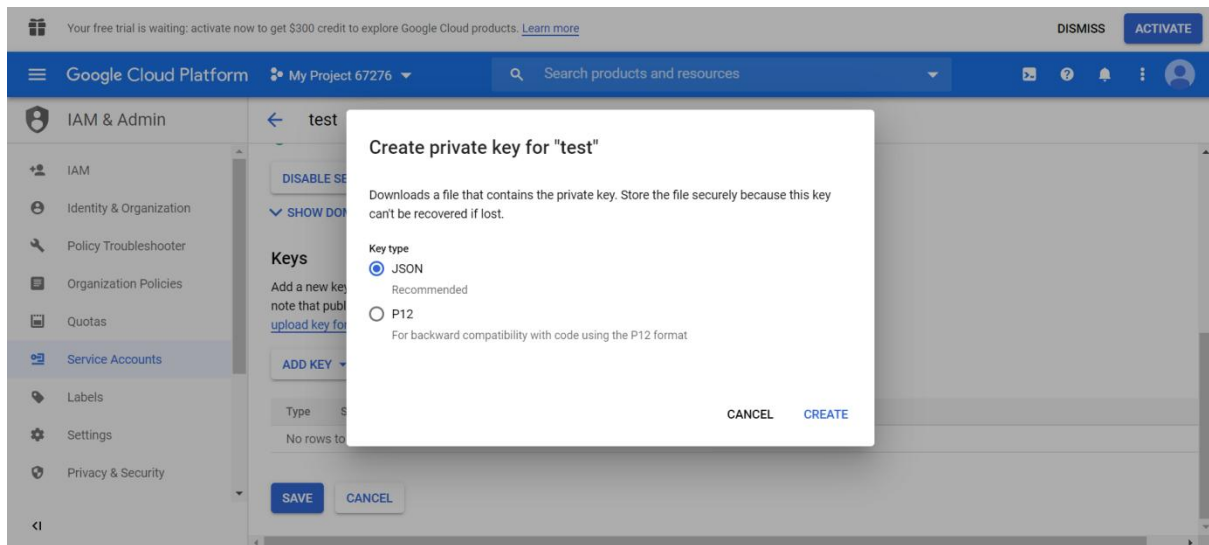
#### Service Accounts

[Manage service accounts](#)

<input type="checkbox"/>	Email	Name	Usage with this service (last 30 days) ? ↓	Usage with all services (last 30 days) ?
No service accounts to display				

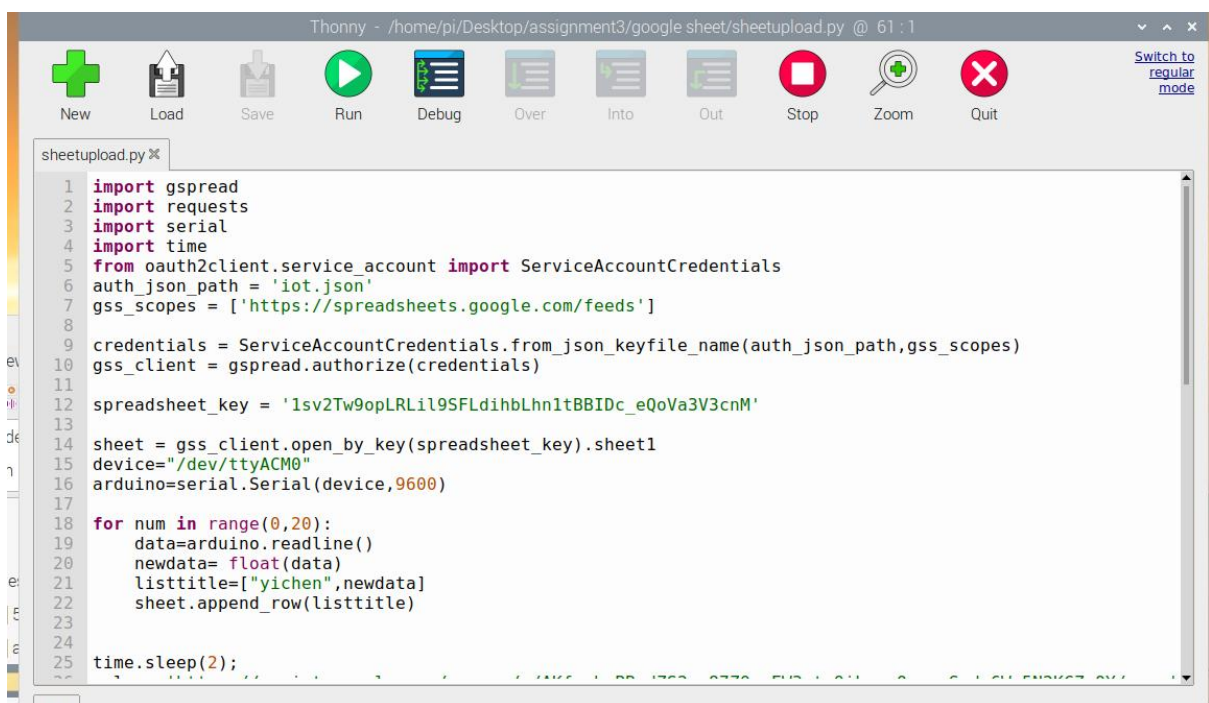
### Step 4: Download client configuration and save as json file





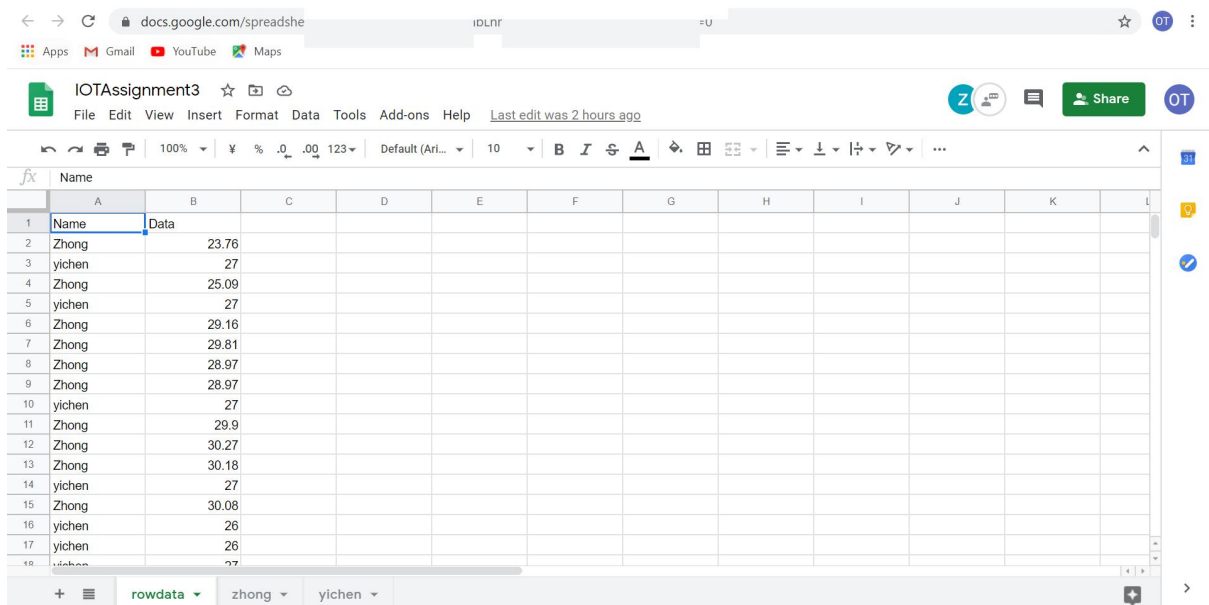
Step 5: Install the Google Client Library and gspread library

### 2.1.3 Use Google sheet Api for data upload in the project



```
zhong.json - Mousepad
File Edit Search View Document Help
{
  "type": "service_account",
  "project_id": "iotsheet-279609",
  "private_key_id": "ccd8c90ef0ea4e5fa3724d9f5415a55cd251d59f",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQDwV",
  "client_email": "wangzhong@iotsheet-279609.iam.gserviceaccount.com",
  "client_id": "110015592056970603126",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/wangzhong%40iotsheet-27"
}
```

Import the gspread library provided by Google sheet api, add the scope to set up the access scope for the google. Add the json file use to perform identity verification and obtain the corresponding permissions. Define the specific sheet by key, which is provided from the URL of the Sheet. And use `gss_client.open_by_key()` function to open the sheet.



The screenshot shows a Google Sheet titled "IOTAssignment3" with two columns: "Name" and "Data". The "Name" column contains alternating entries of "Zhong" and "yichen", while the "Data" column contains numerical values ranging from 23.76 to 30.08. The sheet is displayed in a web browser window with various Google services (Apps, Gmail, YouTube, Maps) visible in the top navigation bar.

	Name	Data
1	Name	Data
2	Zhong	23.76
3	yichen	27
4	Zhong	25.09
5	yichen	27
6	Zhong	29.16
7	Zhong	29.81
8	Zhong	28.97
9	Zhong	28.97
10	yichen	27
11	Zhong	29.9
12	Zhong	30.27
13	Zhong	30.18
14	yichen	27
15	Zhong	30.08
16	yichen	26
17	yichen	26
18	yichen	27

use the `append_row()` function to upload the data to the sheet.

## 2.1.4 Use Google sheet Api for data download in the project

same with data upload, but we use `get_all_values()` function to download the data.

```

import gspread
import pyfirmata
import time
from oauth2client.service_account import ServiceAccountCredentials
auth_json_path = 'zhen.json'
gss_scopes = ['https://spreadsheets.google.com/feeds']
credentials = ServiceAccountCredentials.from_json_keyfile_name(auth_json_path, gss_scopes)
gss_client = gspread.authorize(credentials)
spreadsheet_key = '1sv2Tw9opLRli19SFLdihbLhn1tBBIDc_eQoVa3V3cnM'
sheet = gss_client.open_by_key(spreadsheet_key).sheet1
value_listdata = sheet.get_all_values()

zhong = []
chen = []
z = 0
c = 0

zYellow = 2 #Zhong's yellow led
zRed = 4 #Zhong's red led
cRed = 6 #Chen's red led
cYellow = 8 #Chen's yellow led

```

## 2.2 Google apps script api

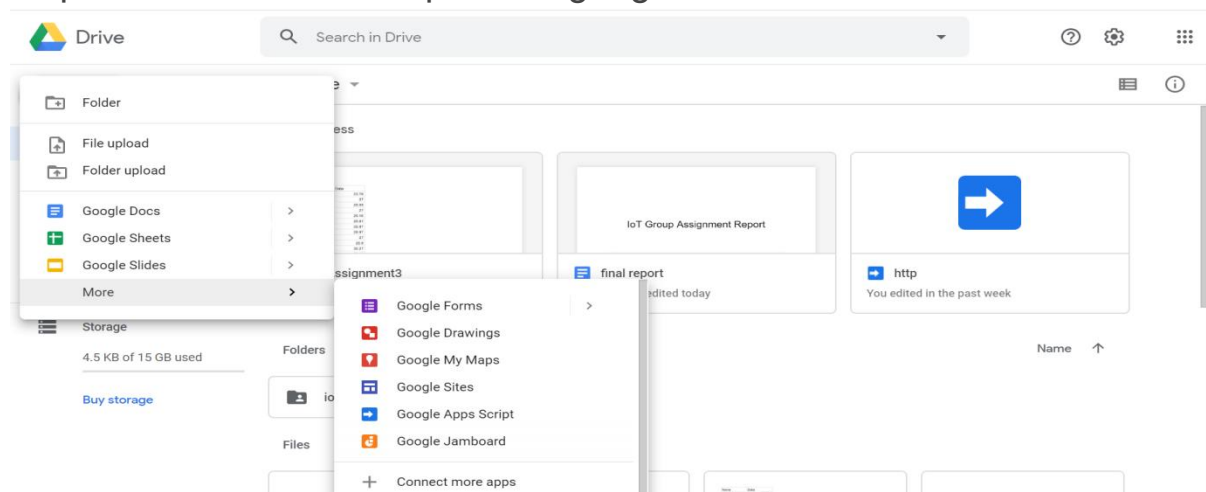
### 2.2.1 What is Google apps script and Google apps script api

Google apps script is a fast application development based on the cloud. the code editor right in the browser, and the script runs on the Google servers.

Google apps script api provides a function to remote execute the script. and also allow the users to deploy their script as web apps and use http request to remote execute the script. This API is like a combination of rest and restful.

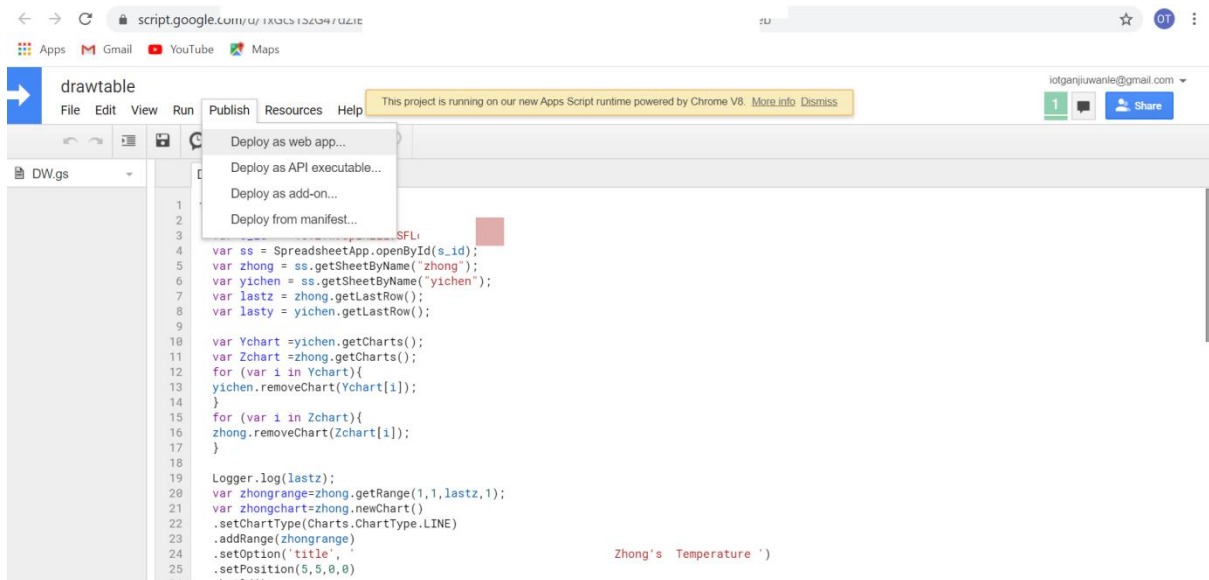
### 2.2.2 How do we use google apps script api

step 1 : create a new script in the google drive.

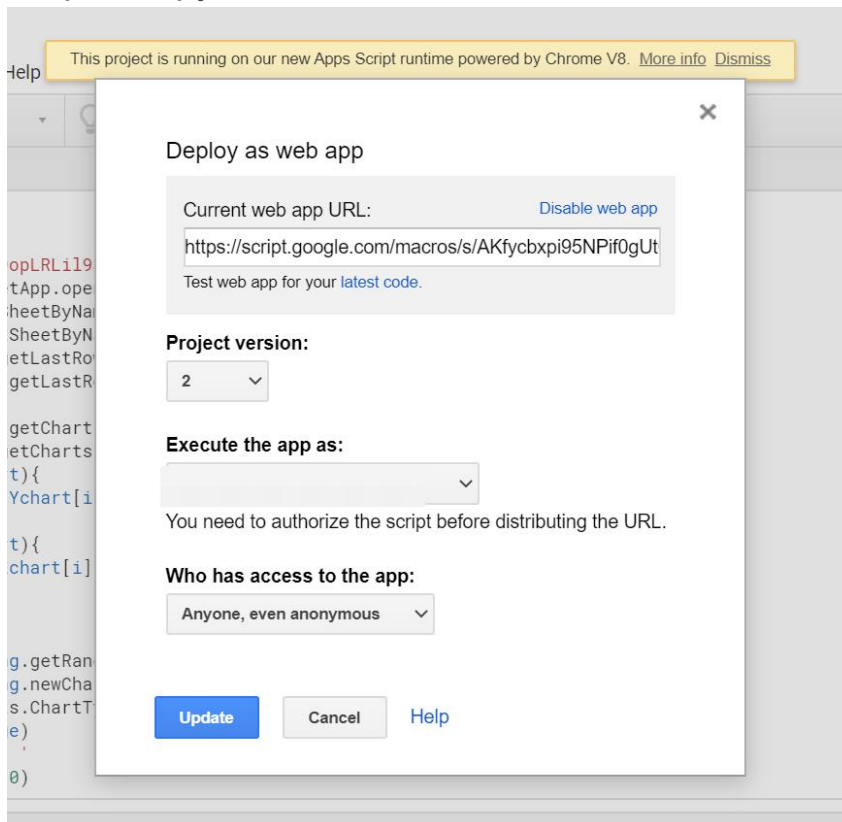


Step 2: write the code for script

Step 3: publish the script as a web apps

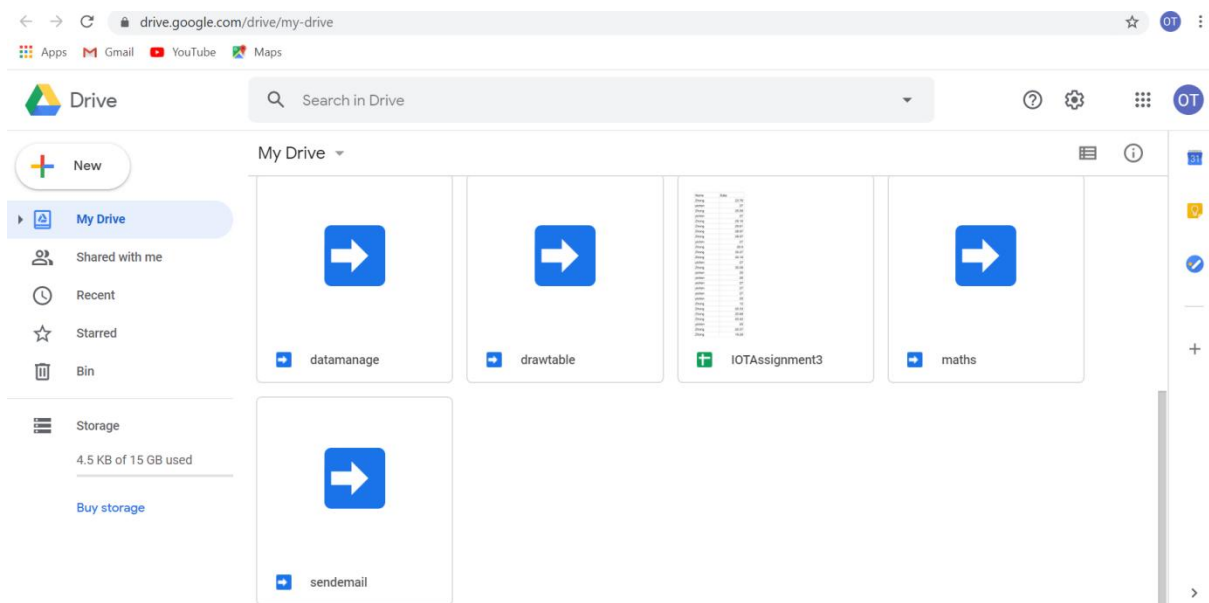


## Step 4: copy the link

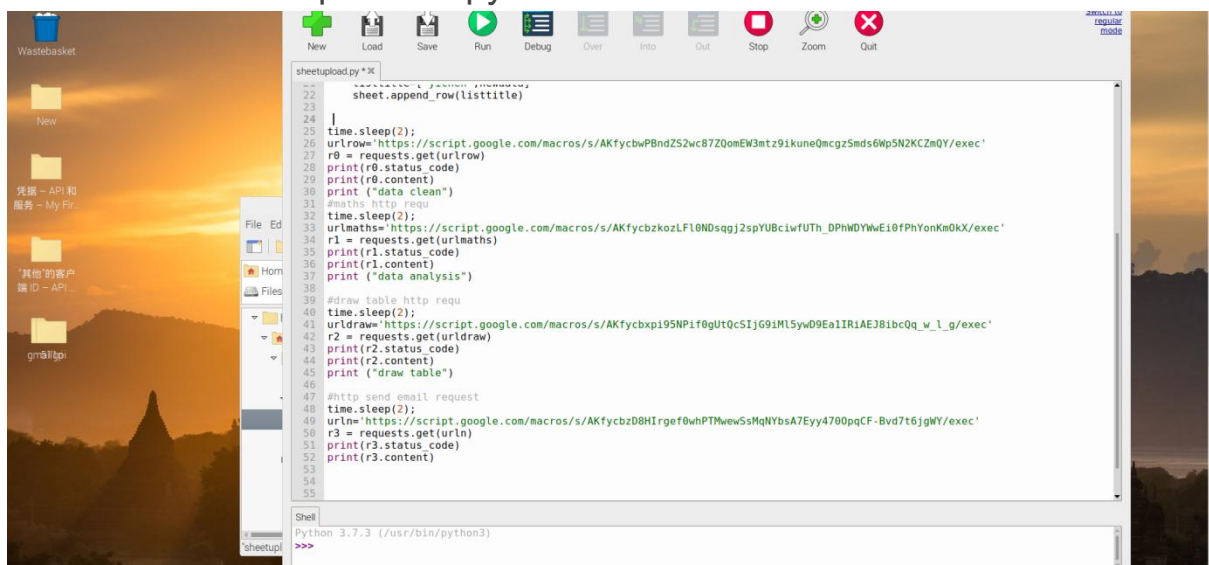


## 2.2.3 What we do by using this in project

We create four scripts and deploy them as web apps. Which are used to separate data, calculate the mean, standard deviation and variance values, one script is used to draw diagrams and the last one is used to send email.

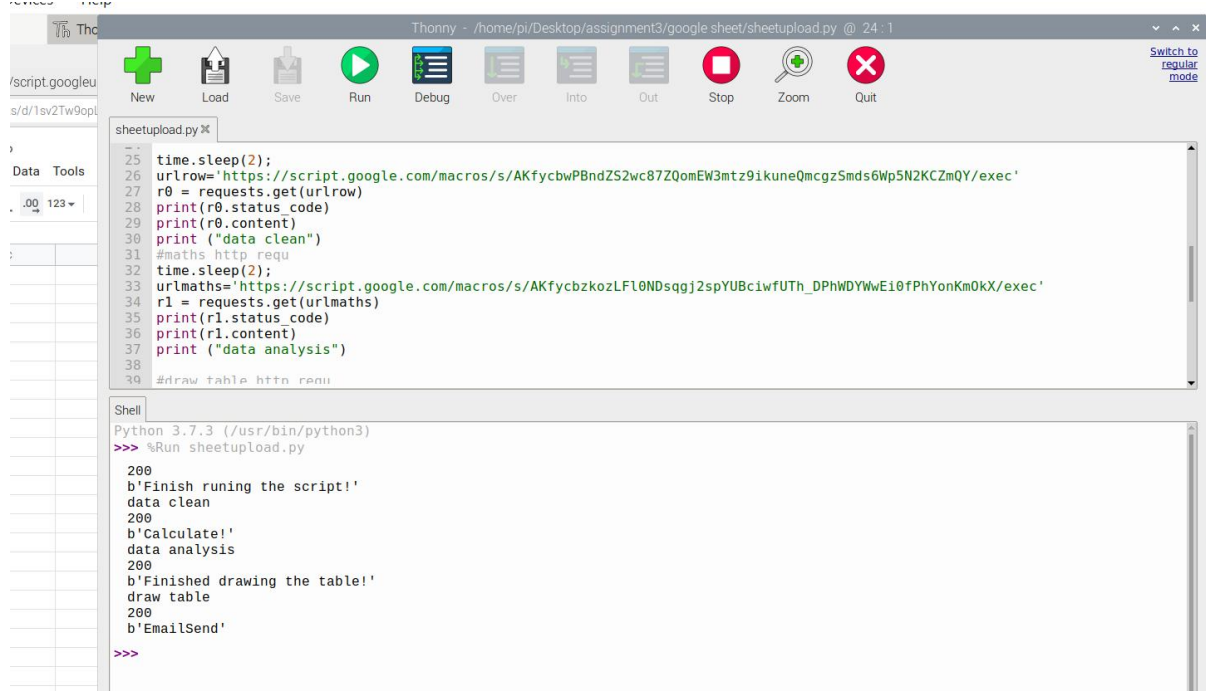


We send HTTP requests in python and run them.



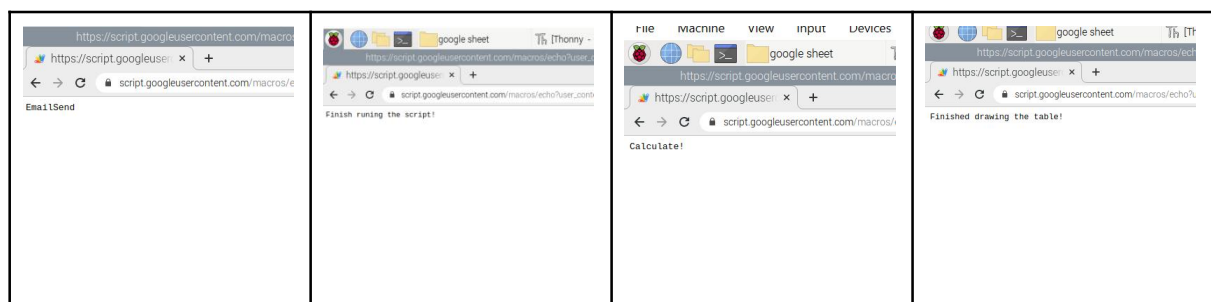
<https://pan.baidu.com/mbox/homepage?action=cloudmanager&type=filefactory&uk=4073099519&errno=0&errmsg=Auth%20Login%20Sucess&>

[bduss=&ssnerror=0&traceid=#share/type=session](#)



```
sheetupload.py %
25 time.sleep(2);
26 urlrow='https://script.google.com/macros/s/AKfycbwPBndZS2wc87ZQomEW3mtz9ikuneQmcgzSmds6Wp5N2KCZmQY/exec'
27 r0 = requests.get(urlrow)
28 print(r0.status_code)
29 print(r0.content)
30 print("data clean")
31 #maths http requ
32 time.sleep(2);
33 urlmaths='https://script.google.com/macros/s/AKfycbzk0zLF10NDsqgJ2spYUBciwfUTH_DPhWDYwWfE10fPhYonKm0kX/exec'
34 r1 = requests.get(urlmaths)
35 print(r1.status_code)
36 print(r1.content)
37 print("data analysis")
38
39 #draw table http requ

Shell
Python 3.7.3 (/usr/bin/python3)
>>> %Run sheetupload.py
200
b'Finish runing the script!'
data clean
200
b'Calculate!'
data analysis
200
b'Finished drawing the table!'
draw table
200
b'EmailSend'
>>>
```

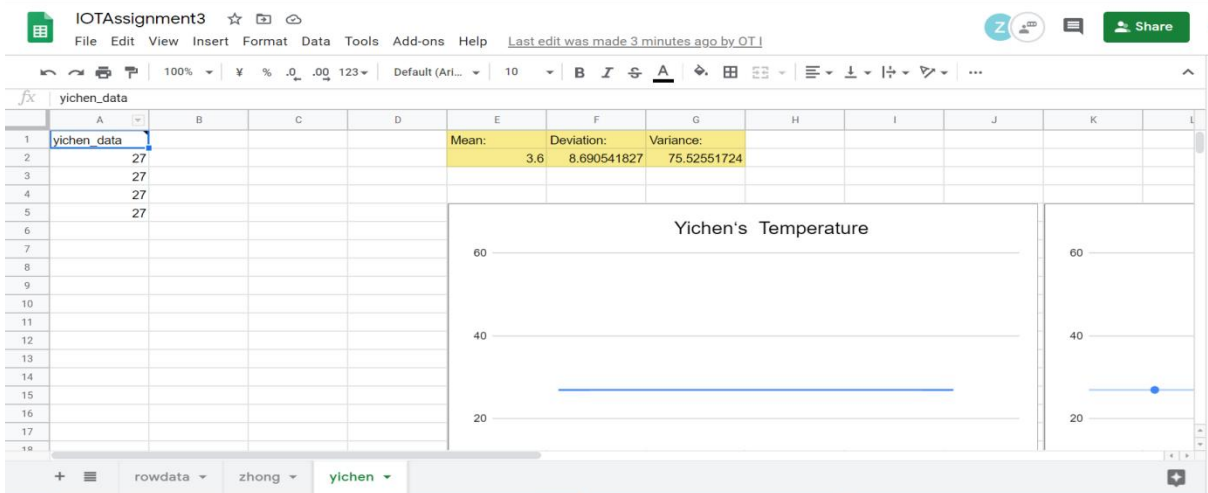
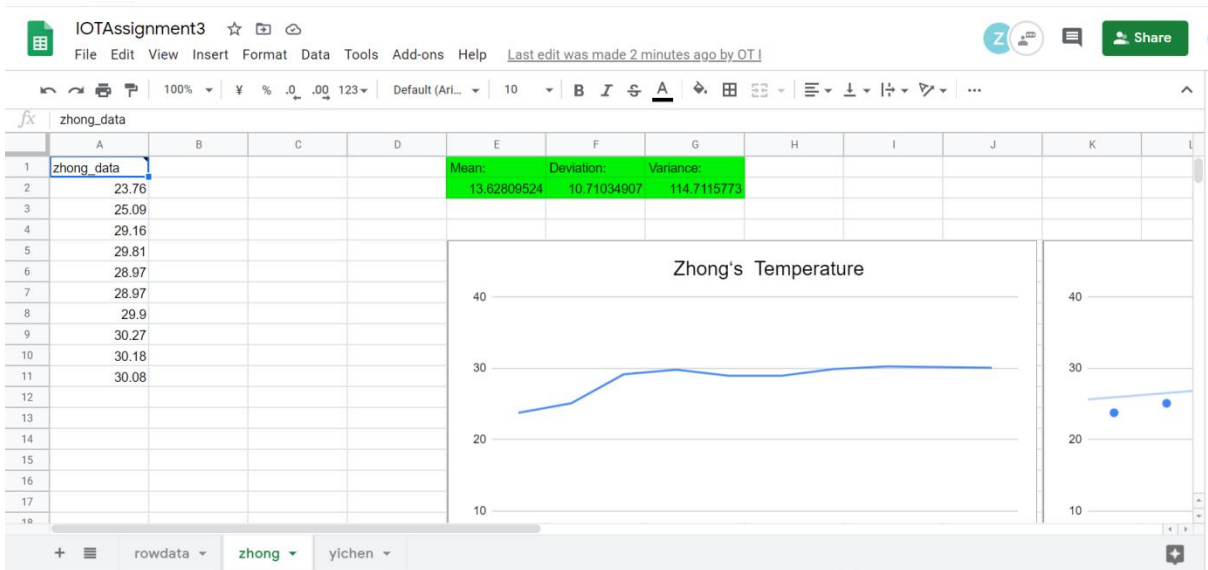
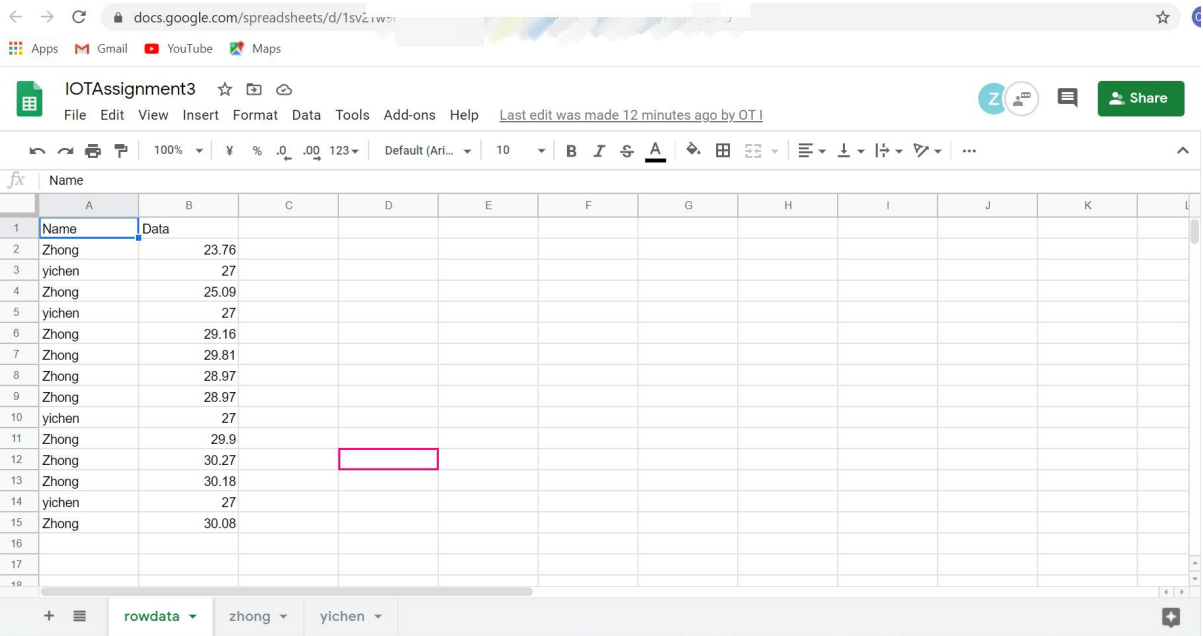


Renderings for script (hold code is in appendix)

Datamanage script:

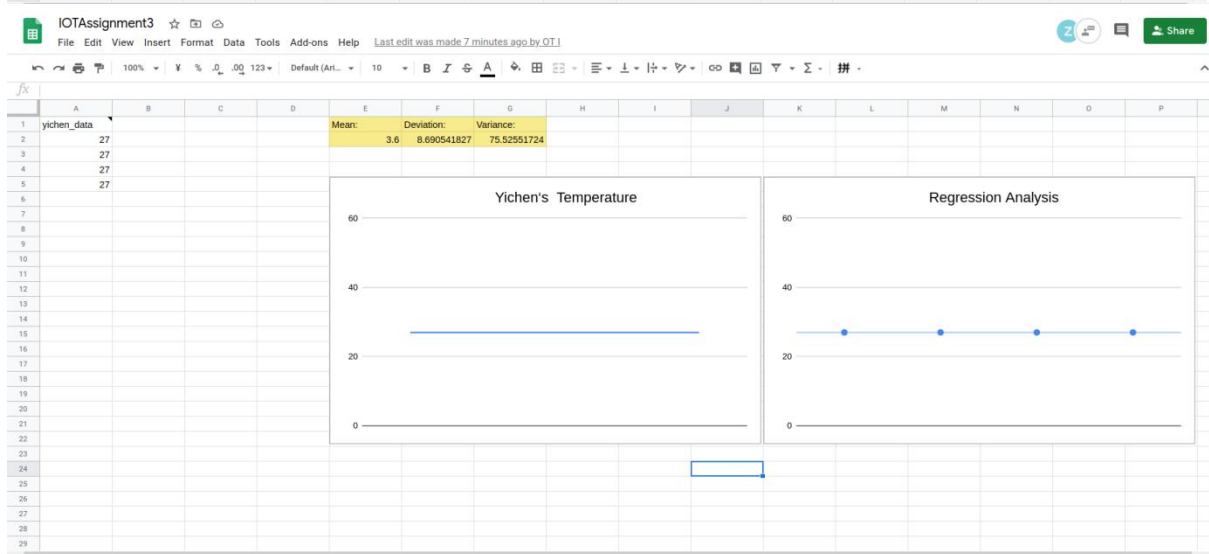
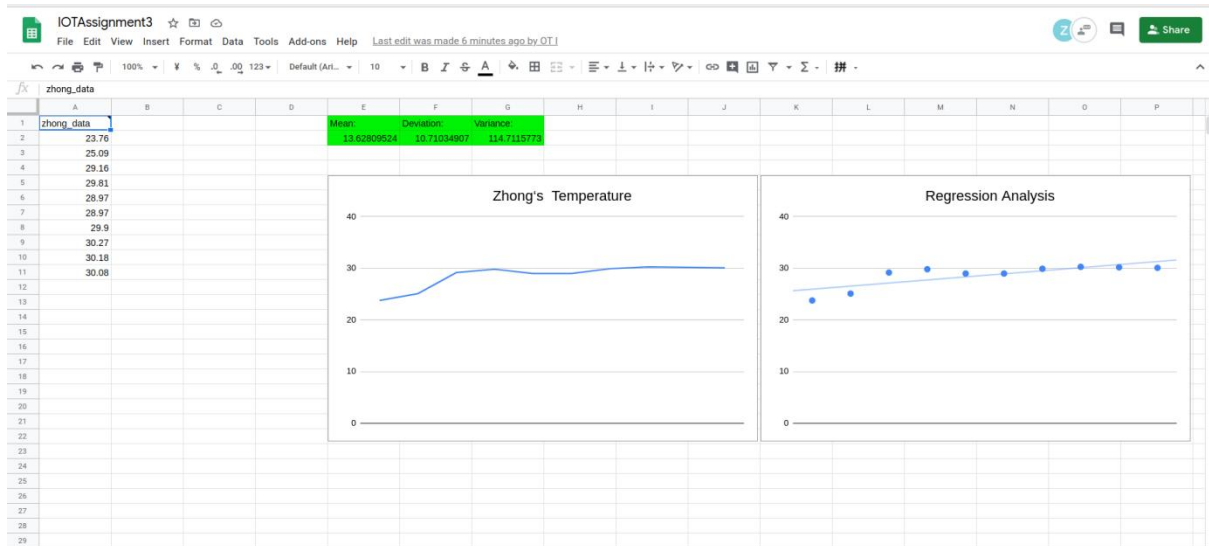
The data on the rowdata page is mixing together, this script will be split according to the name and stored on the page corresponding to the name.



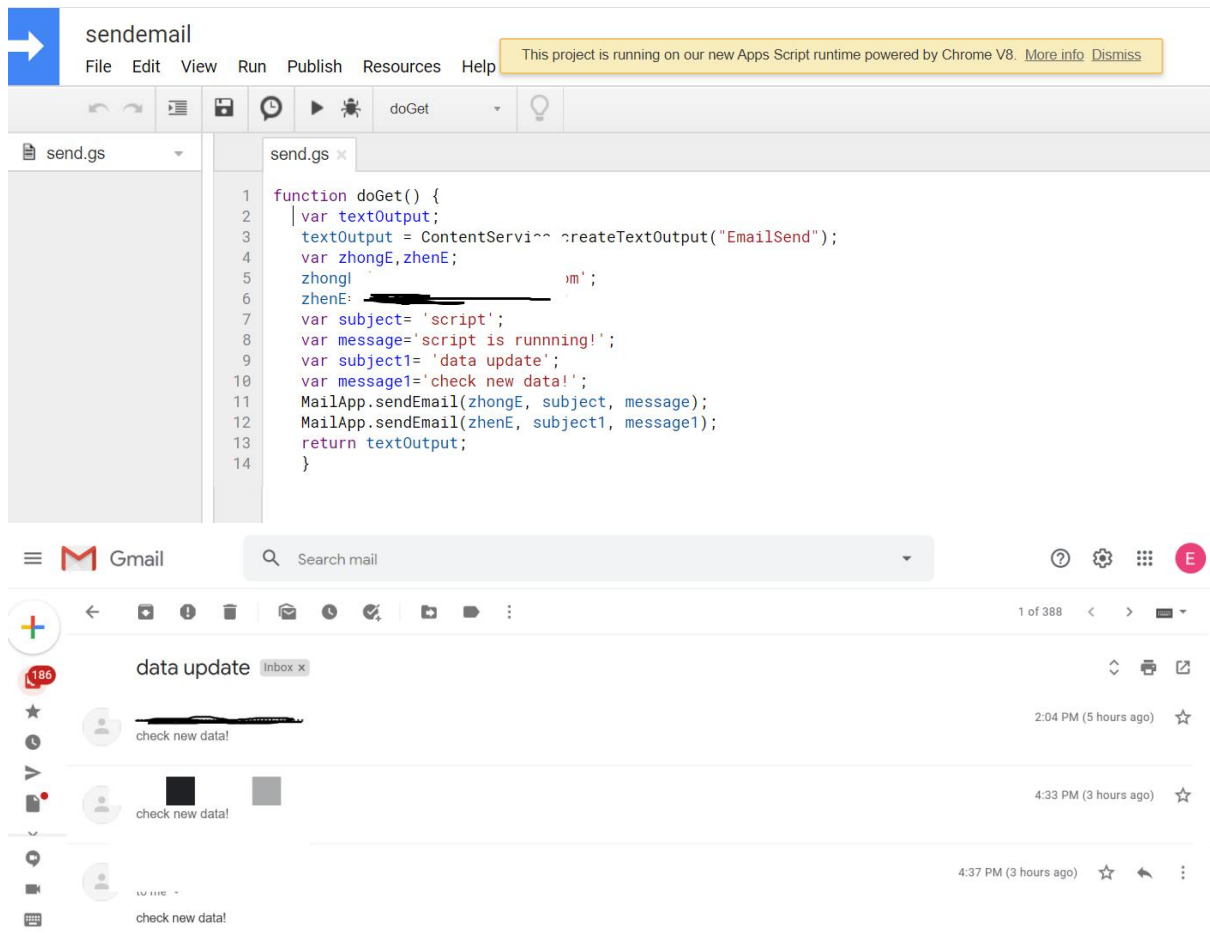


Draw diagrams and Calculation script :

These two scripts will draw diagrams automatically based on the data and calculate the mean, standard deviation and variance values.



send email script:



## 3 Statistical data analysis (Zhong)

### 3.1 Introduction

This part details the difference of data from database which will mainly include Mean value, Standard deviation value and The variance value.

## 3.2 Calculation Formulas

Formula to calculate mean value: sum of data / numbers of data

Formula to calculate standard deviation: 
$$\sigma(r) = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - r)^2}$$

Formula to calculate variance: 
$$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$$
 or (standard deviation^2)

## 3.3 Explanation with code:

sum up all values from two users:

```
28   for (var i=0; i< ynewlength ; i++)
29   {
30       ysum = ynewdata[i][0]+ysum;
31       Logger.log(ysum);
32   }
33   Logger.log(i);
34
35   for (var j=0; j< znewlength; j++)
36   {
37       zsum = znewdata[j][0]+zsum;
38   }
39   Logger.log(j);
40
```

Calculate mean value for two users:

```
44   var meanvalue=ysum/i;
45   Logger.log(meanvalue);
46   var meanvalue1=zsum/j;
47   Logger.log(meanvalue1);
48
```

Calculate Standard deviation value and variance value for two users:

```

50  /*calculate Deviation value*/
51  var S2=0;
52  var S21=0;
53  var variancevalue=0;
54  var variancevalue1=0;
55  var deviationvalue=0;
56  var deviationvalue1=0;
57
58  for (var v=0; v< ynewdata.length; v++)
59  {
60      S2=(ynewdata[v][0]-meanvalue)*(ynewdata[v][0]-meanvalue)+S2;
61      Logger.log(v);
62      Logger.log(S2);
63  }
64
65  for (var b=0; b< znewdata.length; b++)
66  {
67      S21=(znewdata[b][0]-meanvalue1)*(znewdata[b][0]-meanvalue1)+S21;
68      Logger.log(b);
69      Logger.log(S21);
70  }
71
72  variancevalue=(S2/(i-1));
73  variancevalue1=(S21/(j-1));
74  deviationvalue=Math.sqrt(variancevalue);
75  deviationvalue1=Math.sqrt(variancevalue1);
76

```

All code for calculation, By running the function below it will automatically calculate those values.

```

1 function doGet() {
2     var textOutput;
3     textOutput = ContentService.createTextOutput("Calculate!");
4     var s_id = '1sv2Tw9opLRlil9SFLdihbLhn1tBBIDc_eQoVa3V3cnM';
5     var ss = SpreadsheetApp.openById(s_id);
6     var sheet = ss.getSheetByName("rowdata");
7     var zhong = ss.getSheetByName("zhong");
8     var yichen = ss.getSheetByName("yichen");
9     var lastz = zhong.getLastRow();
10    var lasty = yichen.getLastRow();
11    Logger.log(lasty);
12    Logger.log(lastz);
13    var ynewdata = yichen.getRange(2,1,lasty-1).getValues();
14    var znewdata = zhong.getRange(2,1,lastz-1).getValues();
15    Logger.log(ynewdata);
16    Logger.log(znewdata);
17    var ysum = 0;
18    var zsum = 0;
19    var mean = 'Mean: ';
20    var deviation = 'Deviation: ';
21    var variance = 'Variance: ';
22
23    /*calculate mean value*/
24    var ynewlength = ynewdata.length;
25    var znewlength = znewdata.length;
26    Logger.log(ynewlength);
27    Logger.log(znewlength);
28    for (var i=0; i< ynewlength ; i++)
29    {
30        ysum = ynewdata[i][0]+ysum;
31        Logger.log(ysum);
32    }
33    Logger.log(i);
34
35    for (var j=0; j< znewlength; j++)
36    {
37        zsum = znewdata[j][0]+zsum;
38    }
39    Logger.log(j);
40
41    Logger.log(ysum);
42    Logger.log(zsum);
43
44    var meanvalue=ysum/i;
45    Logger.log(meanvalue);
46    var meanvalue1=zsum/j;
47    Logger.log(meanvalue1);
48
49

```



```

50      /*calculate Deviation value*/
51      var S2=0;
52      var S21=0;
53      var variancevalue=0;
54      var variancevalue1=0;
55      var deviationvalue=0;
56      var deviationvalue1=0;
57
58      for (var v=0; v< ynewdata.length; v++)
59      {
60          S2=(ynewdata[v][0]-meanvalue)*(ynewdata[v][0]-meanvalue)+S2;
61          Logger.log(v);
62          Logger.log(S2);
63      }
64
65      for (var b=0; b< znewdata.length; b++)
66      {
67          S21=(znewdata[b][0]-meanvalue1)*(znewdata[b][0]-meanvalue1)+S21;
68          Logger.log(b);
69          Logger.log(S21);
70      }
71
72      variancevalue=(S2/(i-1));
73      variancevalue1=(S21/(j-1));
74      deviationvalue=Math.sqrt(variancevalue);
75      deviationvalue1=Math.sqrt(variancevalue1);
76
77
78      zhong.getRange(1,5).setValue(mean);
79      zhong.getRange(1,6).setValue(deviation);
80      zhong.getRange(1,7).setValue(variance);
81
82      yichen.getRange(1,5).setValue(mean);
83      yichen.getRange(1,6).setValue(deviation);
84      yichen.getRange(1,7).setValue(variance);
85
86
87      zhong.getRange(2,5).setValue(meanvalue1);
88      zhong.getRange(2,6).setValue(deviationvalue1);
89      zhong.getRange(2,7).setValue(variancevalue1);
90      zhong.getRange(1,5,2,3).setBackgroundRGB(0,243,4)
91
92      yichen.getRange(2,5).setValue(meanvalue);
93      yichen.getRange(2,6).setValue(deviationvalue);
94      yichen.getRange(2,7).setValue(variancevalue);
95      yichen.getRange(1,5,2,3).setBackgroundRGB(248,235,140);
96      return textOutput;
97      }

```

Data calculation source code:

[https://script.google.com/d/14ZW\\_L-GyXscNoKhH\\_nb-0lezLS8bqZRYiuwEOTP9AMWO\\_Hwn-YnTnaja/edit?usp=sharing](https://script.google.com/d/14ZW_L-GyXscNoKhH_nb-0lezLS8bqZRYiuwEOTP9AMWO_Hwn-YnTnaja/edit?usp=sharing)

## 4 Data visualization (Zhong)

### 4.1 Introduction

After all data has been added into the database, then it will create line charts for two different spreadsheets based on the data uploaded from different people and set the

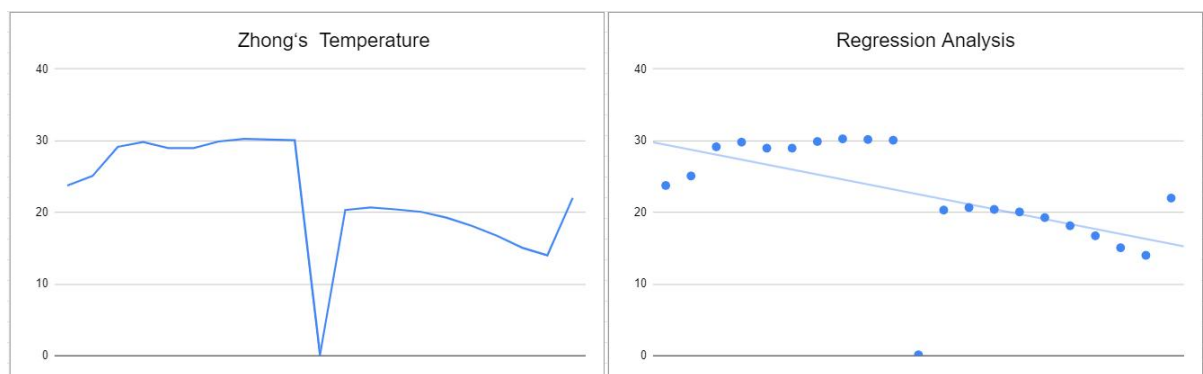
mean value, standard deviation value and variance value above the charts also it will set background color to identify the user .

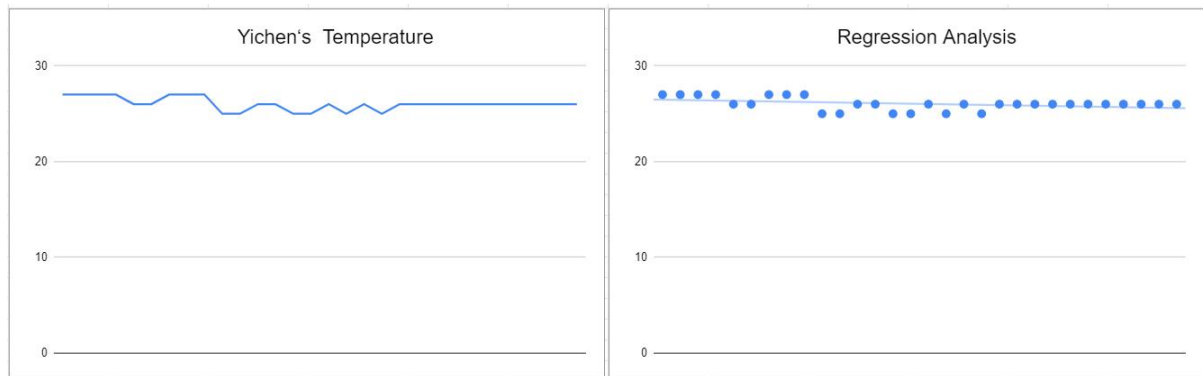
## 4.2 Code to achieve this goal:

### 4.2.1 Create Line chart and Regression analysis chart for different users based on their temperature data :

```
20 var zhongrange=zhong.getRange(1,1,lastz,1);
21 var zhongchart=zhong.newChart()
22 .setChartType(Charts.ChartType.LINE)
23 .addRange(zhongrange)
24 .setOption('title', 'Zhong's Temperature ')
25 .setPosition(5,5,0,0)
26 .build();
27 zhong.insertChart(zhongchart);
28
29 var zhongchart1=zhong.newChart()
30 .setChartType(Charts.ChartType.SCATTER)
31 .addRange(zhongrange)
32 .setOption('title', 'Regression Analysis ')
33 .setPosition(5,11,0,0)
34 .setOption("trendlines", {0: {type: "linear"}})
35 .build();
36 zhong.insertChart(zhongchart1)
37
38
39 Logger.log(zhongrange);
40
41 var chenrange=yichen.getRange(1,1,lasty,1);
42 var chenchart=yichen.newChart()
43 .setChartType(Charts.ChartType.LINE)
44 .addRange(chenrange)
45 .setOption('title', 'Yichen's Temperature ')
46 .setPosition(5,5,0,0).build();
47 yichen.insertChart(chenchart)
48
49 var chenchart1=yichen.newChart()
50 .setChartType(Charts.ChartType.SCATTER)
51 .addRange(chenrange)
52 .setOption('title', 'Regression Analysis ')
53 .setPosition(5,11,0,0)
54 .setOption("trendlines", {0: {type: "linear"}})
55 .build();
56 yichen.insertChart(chenchart1)
```

### Result:





4.2.2 Set the mean value, standard deviation value and variance value above the charts then set background color to identify the user (green background: Zhong's data. yellow background: Yichen's data):

```

78  zhong.getRange(1,5).setValue(mean);
79  zhong.getRange(1,6).setValue(deviation);
80  zhong.getRange(1,7).setValue(variance);
81
82  yichen.getRange(1,5).setValue(mean);
83  yichen.getRange(1,6).setValue(deviation);
84  yichen.getRange(1,7).setValue(variance);
85
86
87  zhong.getRange(2,5).setValue(meanvalue1);
88  zhong.getRange(2,6).setValue(deviationvalue1);
89  zhong.getRange(2,7).setValue(variancevalue1);
90  zhong.getRange(1,5,2,3).setBackgroundRGB(0,243,4)
91
92  yichen.getRange(2,5).setValue(meanvalue);
93  yichen.getRange(2,6).setValue(deviationvalue);
94  yichen.getRange(2,7).setValue(variancevalue);
95  yichen.getRange(1,5,2,3).setBackgroundRGB(248,235,140);

```

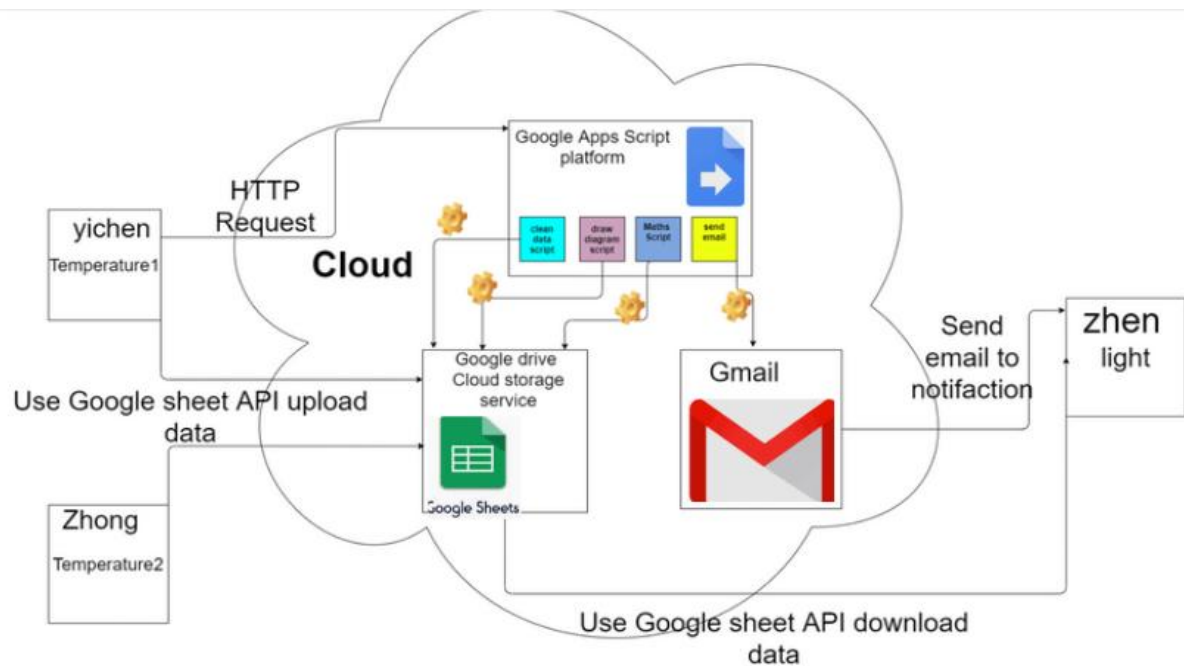
Result:

Mean:	Deviation:	Variance:
22.52619048	7.518557359	56.52870476
Mean:	Deviation:	Variance:
26.03333333	0.6686751355	0.4471264368

Drawtable source code:

<https://script.google.com/d/1xGcs1SzG47dZIEfNiOmf5XLVHJZNPFuKEII2OkfiyoI0qMSvltkVDgIA/edit?usp=sharing>

## 5 Implementation and User Manual (Bong Siaw Zhen, yichen)



Stage1: Use the arduino board and sensor to connect the data to the Raspberry Pi virtual machine through the serial port.

Stage2: upload data to the google drive cloud storage service by using google sheet api.

Stage3: Send Http request to run the script which deploy as web application on the google apps script platform by using python.

Stage4: Scripts will be process data, calculated, analyzed and visualized automatically. When finish process data will send the email to executive(Zhen)

Stage5: The executor receives the email about data update, executor downloads the data using api, and transmits it to the Arduino, and controls the light according to the data.

Logic of the execution (Zhen) section:

The execution section detects the temperature from two different places and operates the actuator at the third place. There are 4 LED lights, two LEDs for each place. The yellow LED will light up when the temperature is lower or equal than 25 degrees. If the system senses that the temperature is higher than 25 degrees up to five times, the red LED will light up.

## 6. Reference

<https://developers.google.com/apps-script/overview>

<https://blog.heron.me/executing-functions-using-the-apps-script-api-47c822681e5d>

<https://www.labnol.org/code/19871-get-post-requests-google-script>

<https://marcelkornblum.com/using-google-apps-scripts-as-a-back-end-2f0d6ba29911>

## 7. Appendix

By using those links below to access the app script code.

1. Datamanage script link:

[https://script.google.com/d/12UhWrnX42Ig9BsLs3vqhonnLh0pcJDBbPyh\\_rT8G032lY6Vgulivb\\_TZ/edit?usp=sharing](https://script.google.com/d/12UhWrnX42Ig9BsLs3vqhonnLh0pcJDBbPyh_rT8G032lY6Vgulivb_TZ/edit?usp=sharing)

2. Draw diagrams script link:

<https://script.google.com/d/1xGcs1SzG47dZIEfNiOmf5XLVHJZNPFuKElI2OkfiyoI0qMSvltkVDglA/edit?usp=sharing>

3. Calculation script link:

[https://script.google.com/d/14ZW\\_L-GyXscNoKhH\\_nb-0IezLS8bqZRyiuwEOTP9AMWO\\_Hwn-YnTnaja/edit?usp=sharing](https://script.google.com/d/14ZW_L-GyXscNoKhH_nb-0IezLS8bqZRyiuwEOTP9AMWO_Hwn-YnTnaja/edit?usp=sharing)

4. Send email script link:

5. Python code link:

[https://docs.google.com/document/d/12MiiyMO\\_H71K5KSOPy8VxFt1Css4mTcoWTdbeQq2TQA/edit?usp=sharing](https://docs.google.com/document/d/12MiiyMO_H71K5KSOPy8VxFt1Css4mTcoWTdbeQq2TQA/edit?usp=sharing)