Student name: Hoang Long Tran

Student ID: s223128143

SIT225: Data Capture Technologies

Activity 5.1: Firebase Realtime database

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Data is stored as JSON and synchronized in real-time to every connected client. In this activity, you will set up and perform operations such as queries and updates on the database using Python programming language.

Hardware Required

No hardware is required.

Software Required

Firebase Realtime database
Python 3

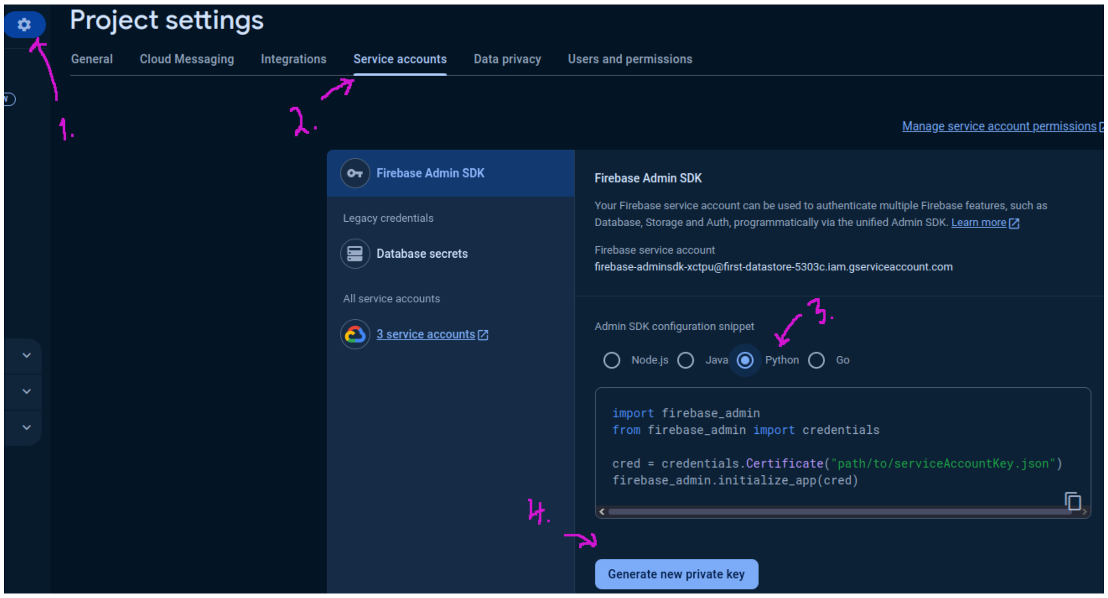
Steps

Step	Action
1	**Create an Account**: First, you will need to create an account in the Firebase console, follow instructions in the official Firebase document (https://firebase.google.com/docs/database/rest/start).
2	**Create a Database**: Follow the above Firebase document to create a database. When you click on Create Database, you have to specify the location of the database and the security rules. Two rules are available – locked mode and test mode; since we will be using the database for reading, writing, and editing, we choose test mode.
3	**Setup Python library for Firebase access**: We will be using Admin Database API, which is available in *firebase_admin* library. Use the below command in the command line to install. You can

follow a Firebase tutorial here (https://www.freecodecamp.org/news/how-to-get-started-with-firebase-using-python).

```
$ pip install firebase_admin
```

Firebase will allow access to Firebase server APIs from Google Service Accounts. To authenticate the Service Account, we require a private key in JSON format. To generate the key, go to project settings, click Generate new private key, download the file, and place it in your current folder where you will create your Python script.



Firebase Admin SDK

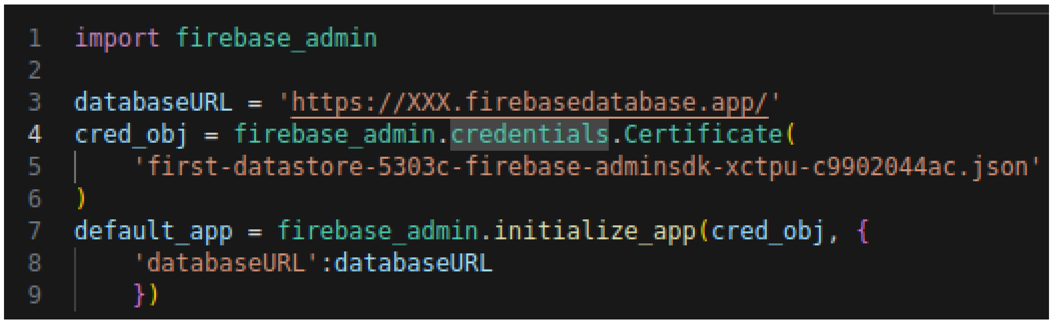
Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. Learn more

Firebase service account
firebase-adminsdk-xctpu@first-datastore-5303c.iam.gserviceaccount.com

Admin SDK configuration snippet
Node.js Java [✓ Python] Go

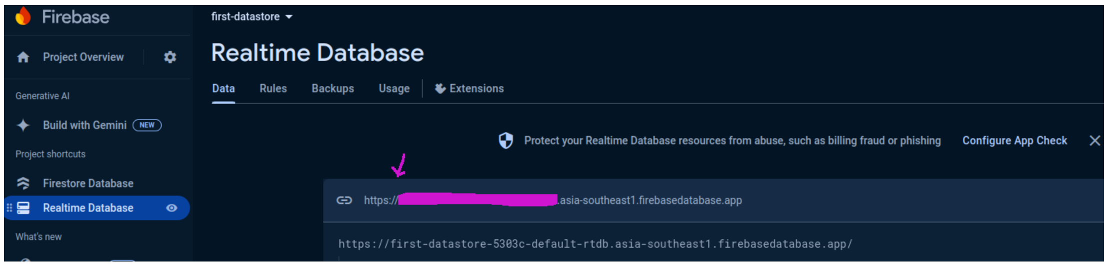
```python
import firebase_admin
from firebase_admin import credentials

cred = credentials.Certificate("path/to/serviceAccountKey.json")
firebase_admin.initialize_app(cred)
```

4 **Connect to Firebase using Python version of Admin Database API**:
A credential object needs to be created to initialise the Python library which can be done using the Python code below. Python notebook can be downloaded here (https://github.com/deakin-deep-dreamer/sit225/blob/main/week_5/firebase_explore.ipynb).



```python
import firebase_admin

databaseURL = 'https://XXX.firebasedatabase.app/'
cred_obj = firebase_admin.credentials.Certificate(
    'first-datastore-5303c-firebase-adminsdk-xctpu-c9902044ac.json'
)
default_app = firebase_admin.initialize_app(cred_obj, {
    'databaseURL':databaseURL
    })
```

The databaseURL is a web address to reach your Firebase database that you have created in step 2. This URL can be found in the Data tab of Realtime Database.



Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://[redacted].asia-southeast1.firebasedatabase.app

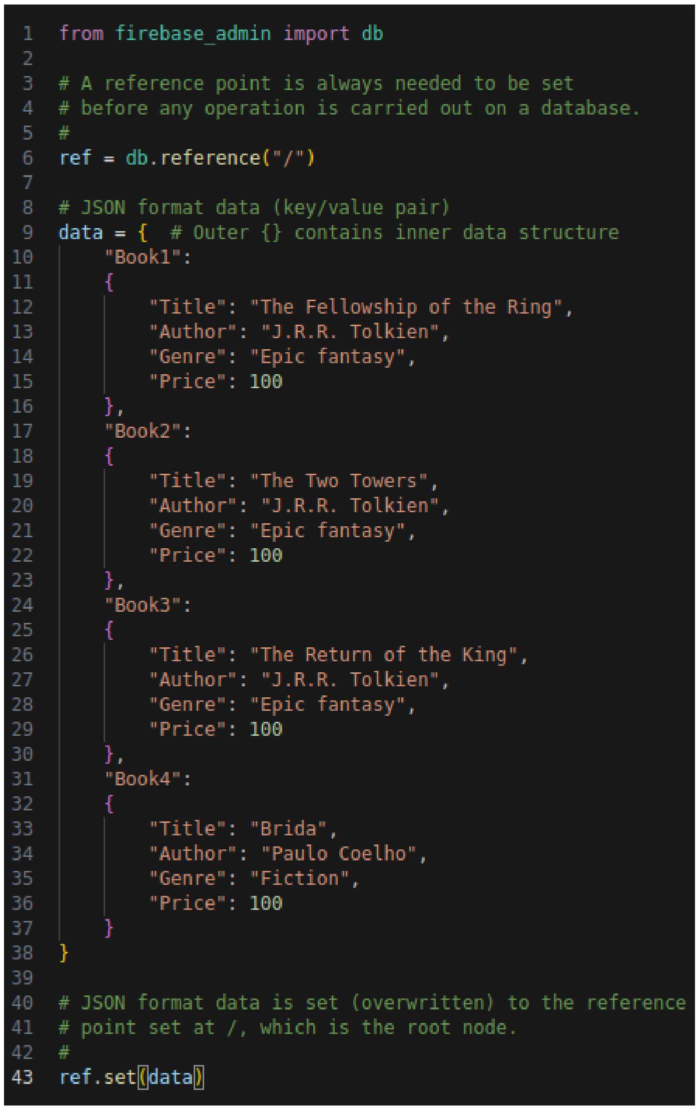
https://first-datastore-5303c-default-rtdb.asia-southeast1.firebasedatabase.app/

If you compile the code snippet above, it should do with no error.

5

Write to database Using the set() Function:

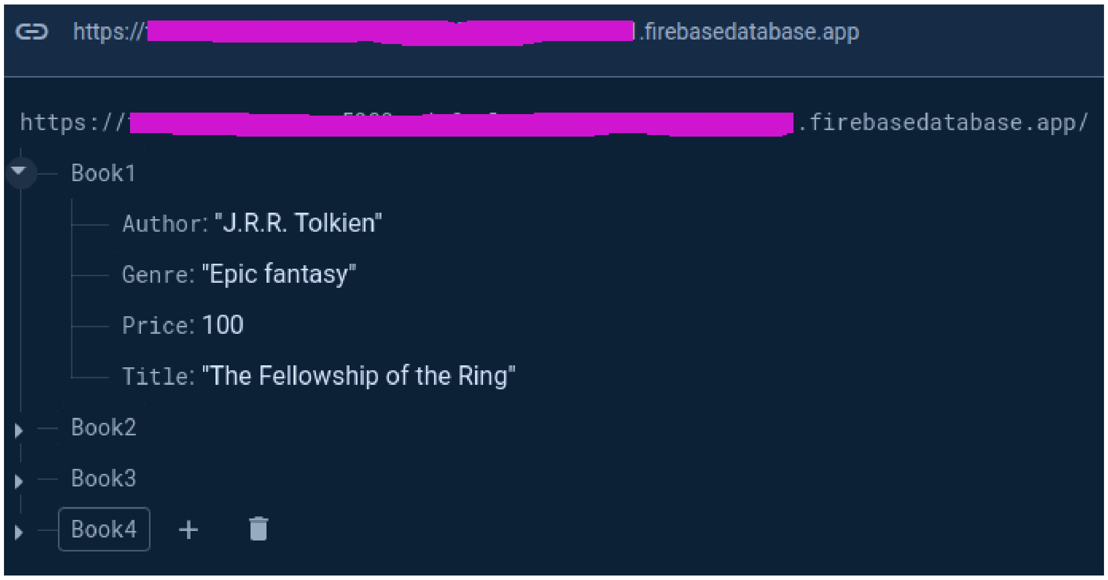
We set the reference to the root of the database (or we could also set it to a key value or child key value). Data needs to be in JSON format as below.



```python
from firebase_admin import db

# A reference point is always needed to be set
# before any operation is carried out on a database.
#
ref = db.reference("/")

# JSON format data (key/value pair)
data = {  # Outer {} contains inner data structure
    "Book1":
    {
        "Title": "The Fellowship of the Ring",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book2":
    {
        "Title": "The Two Towers",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book3":
    {
        "Title": "The Return of the King",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book4":
    {
        "Title": "Brida",
        "Author": "Paulo Coelho",
        "Genre": "Fiction",
        "Price": 100
    }
}

# JSON format data is set (overwritten) to the reference
# point set at /, which is the root node.
#
ref.set(data)
```

A reference point always needed to be set where the data read/write will take place. In the code above, the reference point is set at the root of the NoSQL Document, where consider the database is a JSON tree and / is the root node of the tree). The set() function writes (overwrites) data at the set reference point.

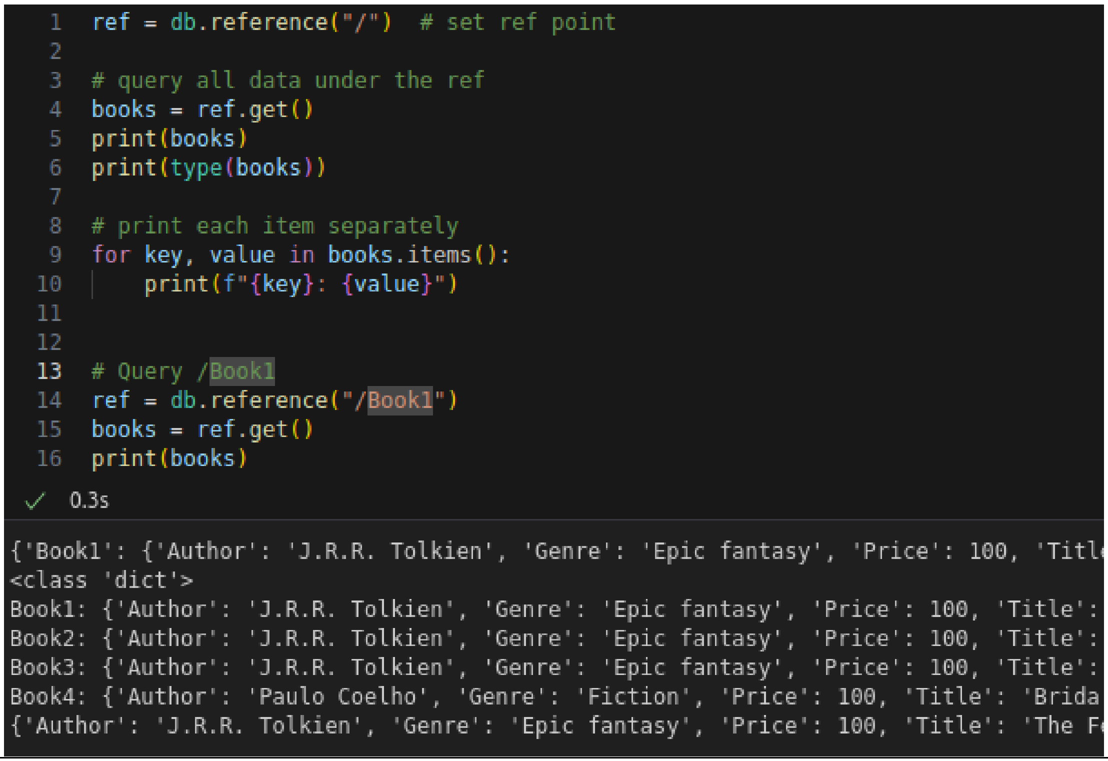
You can visualise the data in the Firebase console as below -



```
https://[redacted].firebasedatabase.app

https://[redacted].firebasedatabase.app/
Book1
    Author: "J.R.R. Tolkien"
    Genre: "Epic fantasy"
    Price: 100
    Title: "The Fellowship of the Ring"
Book2
Book3
Book4
```

6

Read data using get() function:
Data can be read using get() function on the reference set beforehand, as shown below.



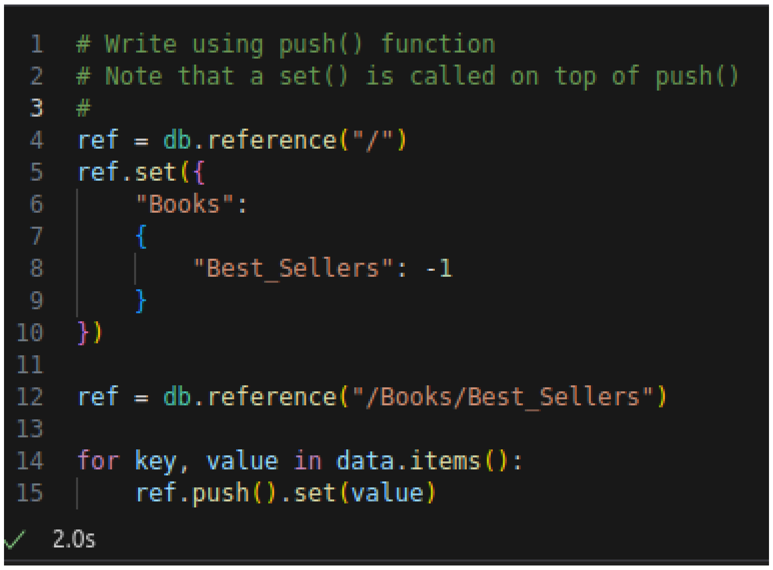
```python
ref = db.reference("/")  # set ref point

# query all data under the ref
books = ref.get()
print(books)
print(type(books))

# print each item separately
for key, value in books.items():
    print(f"{key}: {value}")


# Query /Book1
ref = db.reference("/Book1")
books = ref.get()
print(books)
```

✓ 0.3s

```
{'Book1': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Titl
<class 'dict'>
Book1: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title':
Book2: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title':
Book3: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title':
Book4: {'Author': 'Paulo Coelho', 'Genre': 'Fiction', 'Price': 100, 'Title': 'Brida
{'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The F
```

Consider the reference set in line 1 and the output compared to the reference set at line 14 and the bottom output line to understand the use of db.reference() and ref.get().

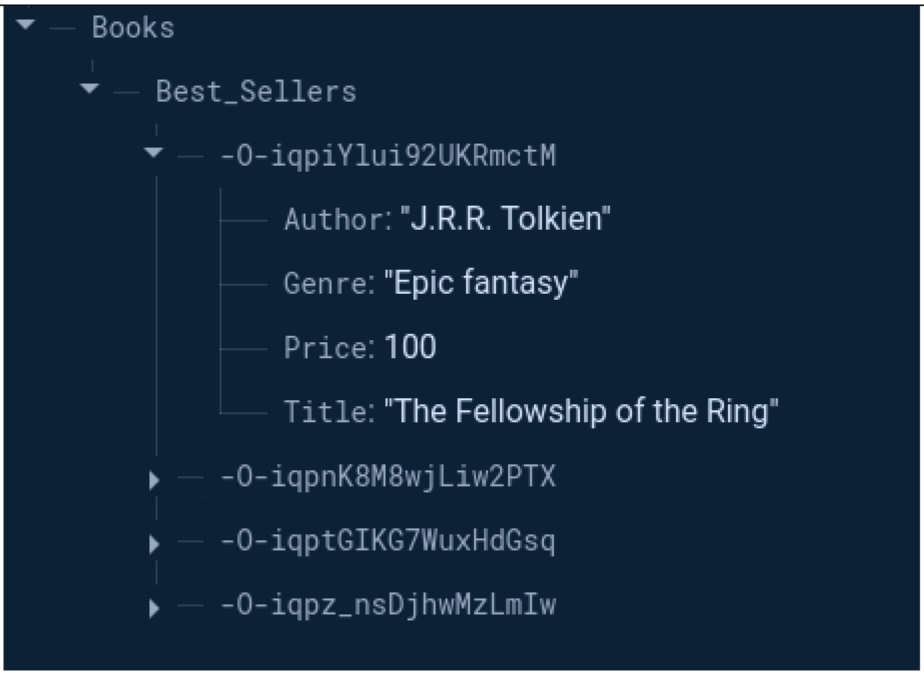
7 **Write to database Using the push() Function**:
The push() function saves data under a *unique system generated key*. This is different than set() where you set the keys such as Book1, Book2, Book3 and Book4 under which the content (author, genre, price and title) appears. Let’s try to push the same data in the root reference. Note that since we already has data under root / symbol, setting (or pushing) in the same reference point will eventually rewrite the original data.



```
# Write using push() function
# Note that a set() is called on top of push()
#
ref = db.reference("/")
ref.set({
    "Books":
    {
        "Best_Sellers": -1
    }
})

ref = db.reference("/Books/Best_Sellers")

for key, value in data.items():
    ref.push().set(value)
```

```
✓ 2.0s
```

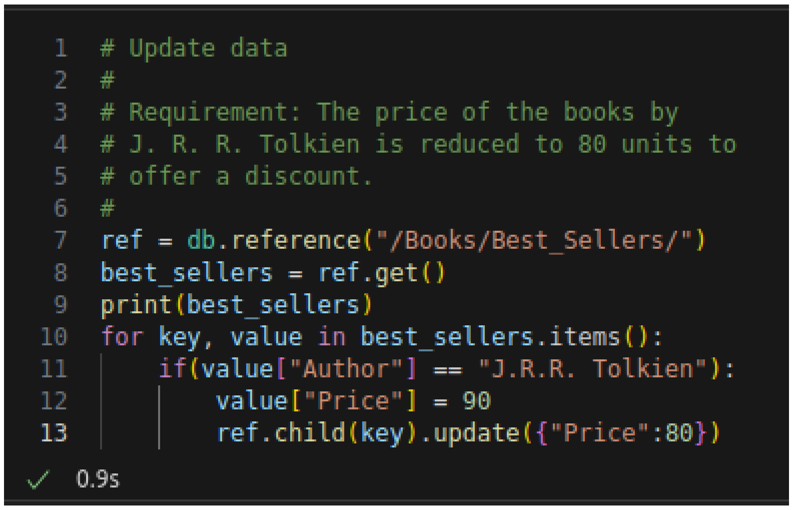
The output will reset the previous data set in / node. The current data is shown below.



```
Books
  Best_Sellers
    -O-iqpiYlui92UKRmctM
      Author: "J.R.R. Tolkien"
      Genre: "Epic fantasy"
      Price: 100
      Title: "The Fellowship of the Ring"
    -O-iqpnK8M8wjLiw2PTX
    -O-iqptGIKG7WuxHdGsq
    -O-iqpz_nsDjhwMzLmIw
```

As you can see, under /Books/Best_Sellers there are 4 nodes where the node head (or node ID) is a randomly generated key which is due to the use of push() function. When data key does not matter, the use of push() function desirable.

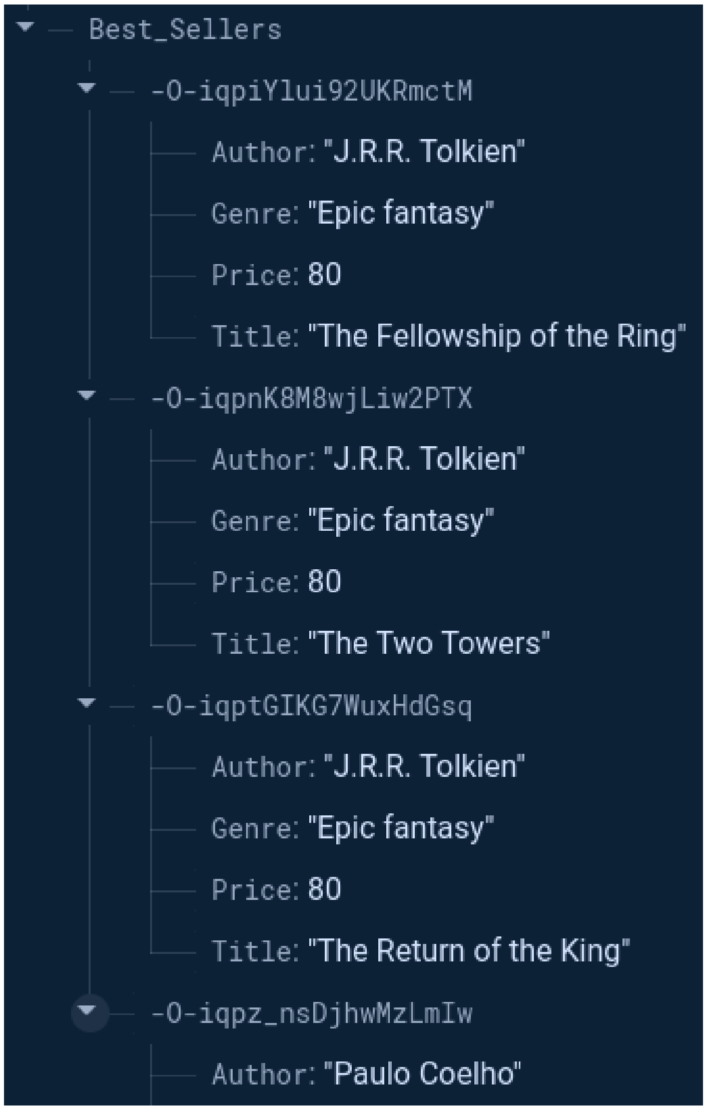
8 **Update data**:
Let's say the price of the books by J. R. R. Tolkien is reduced to 80 units to offer a discount. The first 3 books are written by this author, and we want to apply for a discount on all of them.

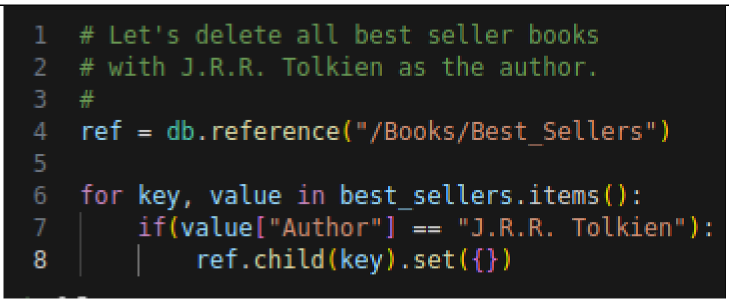
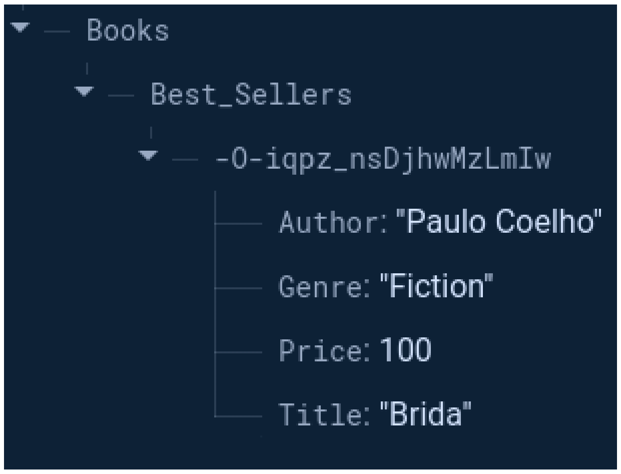
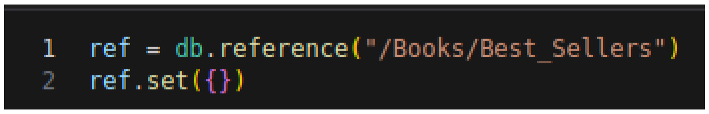


```python
# Update data
#
# Requirement: The price of the books by
# J. R. R. Tolkien is reduced to 80 units to
# offer a discount.
#
ref = db.reference("/Books/Best_Sellers/")
best_sellers = ref.get()
print(best_sellers)
for key, value in best_sellers.items():
    if(value["Author"] == "J.R.R. Tolkien"):
        value["Price"] = 90
        ref.child(key).update({"Price":80})
```

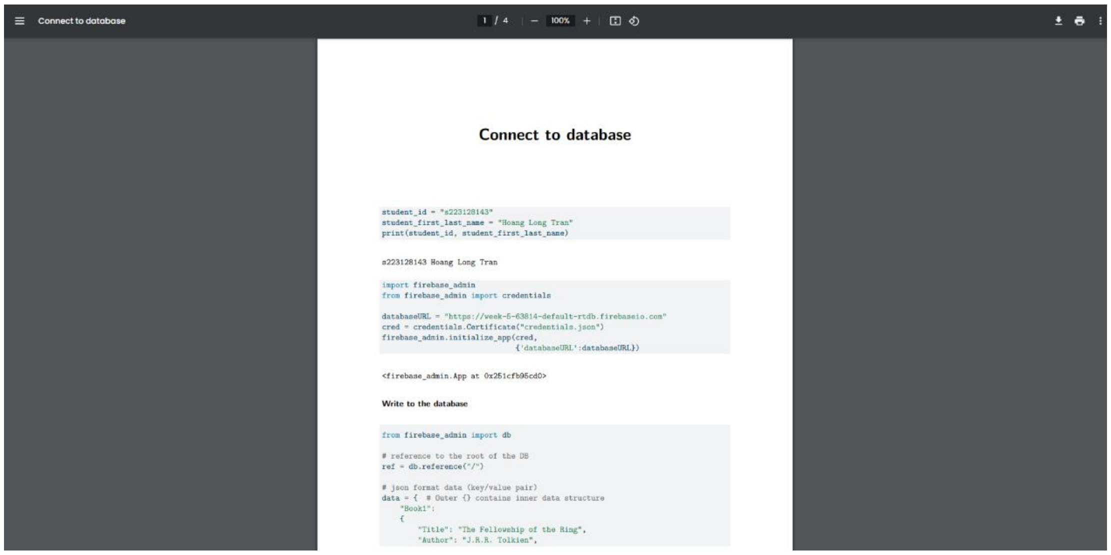
```
✓ 0.9s
```

As you can see, the author name is compared and the new price is set in the best_sellers dictionary and finally, an update() function is called on the ref, however, the current ref is a '/Books/Best_Sellers/', so we need to locate the

	child under the ref node, so ref.child(key) is used in line 13. The output is shown below with a discounted price.  Best_Sellers -O-iqpiYlui92UKRmctM Author: "J.R.R. Tolkien" Genre: "Epic fantasy" Price: 80 Title: "The Fellowship of the Ring" -O-iqpnK8M8wjLiw2PTX Author: "J.R.R. Tolkien" Genre: "Epic fantasy" Price: 80 Title: "The Two Towers" -O-iqptGIKG7WuxHdGsq Author: "J.R.R. Tolkien" Genre: "Epic fantasy" Price: 80 Title: "The Return of the King" -O-iqpz_nsDjhwMzLmIw Author: "Paulo Coelho"
9	**Delete data**: Let's delete all bestseller books with J.R.R. Tolkien as the author. You can locate the node using db.reference() (line 4) and then locate specific record (for loop in line 6) and calling set() with empty data {} as a parameter, such as set({}). The particular child under the ref needs to be located first by using ref.child(key), otherwise, the ref node will be removed – BE CAREFUL.

	
	`# Let's delete all best seller books`
	`# with J.R.R. Tolkien as the author.`
	`#`
	`ref = db.reference("/Books/Best_Sellers")`
	
	`for key, value in best_sellers.items():`
	`    if(value["Author"] == "J.R.R. Tolkien"):`
	`        ref.child(key).set({})`
	This keeps only the other author data, as shown below.
	
	Books
	Best_Sellers
	-O-iqpz_nsDjhwMzLmIw
	Author: "Paulo Coelho"
	Genre: "Fiction"
	Price: 100
	Title: "Brida"
	
	If ref.child() not used, as shown the code below, all data will be removed.
	
	`ref = db.reference("/Books/Best_Sellers")`
	`ref.set({})`
	Now in Firebase console you will see no data exists.

10	Question: Run all the cells in the Notebook you have downloaded in Step 4, fill in the student information at the top cell of the Notebook. Convert the Notebook to PDF and merge with this activity sheet PDF.

	Answer: Convert the Notebook to PDF and merge with this activity sheet PDF. 

Connect to database

1 / 4 100%

Connect to database

```
student_id = "s223128143"
student_first_last_name = "Hoang Long Tran"
print(student_id, student_first_last_name)
```

```
s223128143 Hoang Long Tran
```

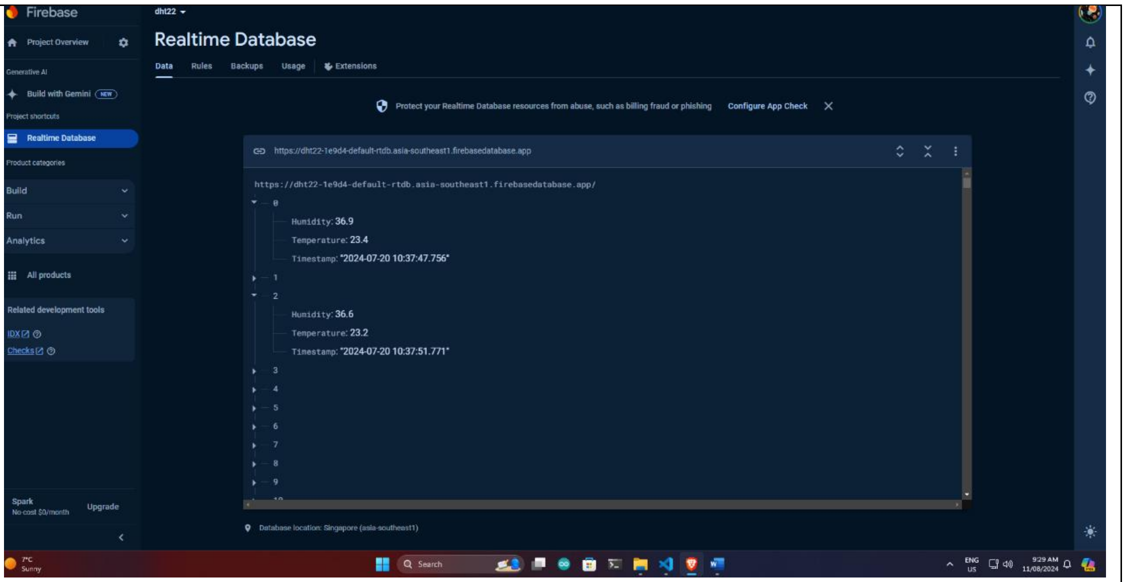
```
import firebase_admin
from firebase_admin import credentials

databaseURL = "https://week-5-63814-default-rtdb.firebaseio.com"
cred = credentials.Certificate("credentials.json")
firebase_admin.initialize_app(cred,
                              {'databaseURL':databaseURL})
```

```
<firebase_admin.App at 0x251cfb95cd0>
```

Write to the database

```
from firebase_admin import db

# reference to the root of the DB
ref = db.reference("/")

# json format data (key/value pair)
data = {  # Outer {} contains inner data structure
    "Book1":
    {
        "Title": "The Fellowship of the Ring",
        "Author": "J.R.R. Tolkien",
```

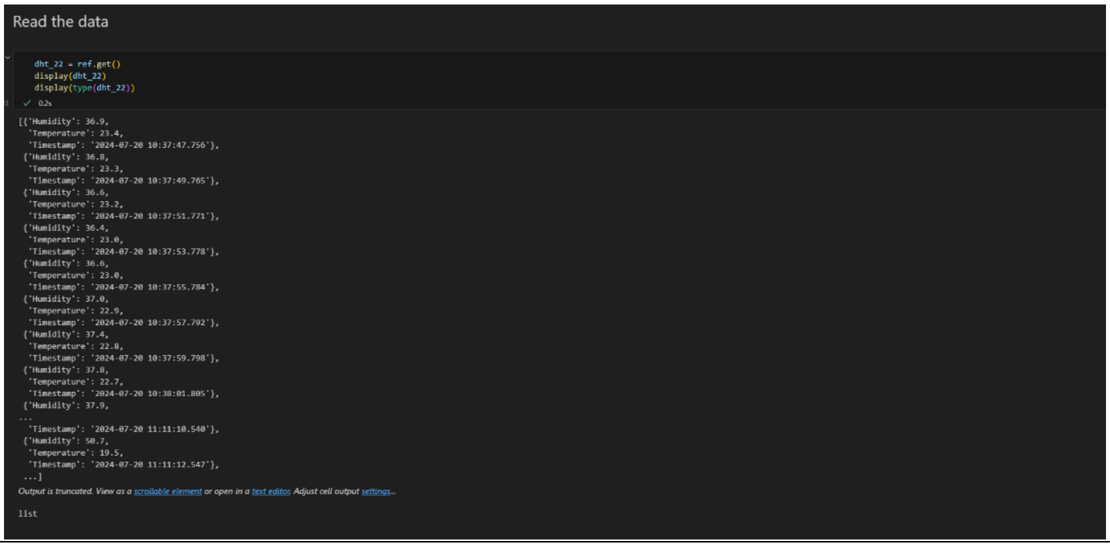
10	Question: Create a sensor data structure for DHT22 sensor which contains attributes such as sensor_name, timestamp, temperature and humidity. Remember there will be other sensors with different sensor variables such as DHT22 has 2 variables, accelerometer sensor has 3. For each such sensor, you will need to gather data over time. Discuss how you are going to handle multiple data values in JSON format? Justify your design. Answer: Since the JSON format is similar to that of a dictionary, I will explain in dictionary terms. There will be keys for each sensor, in each key there will be values of that sensor.
11	Question: Generate some random data for DHT22 sensor, insert data to database, query all data and screenshot the output here. Answer: I am going to reuse my week 2 dht22 data.



Realtime Database

```
https://dht22-1e9d4-default-rtdb.asia-southeast1.firebasedatabase.app/
▾ 0
    Humidity: 36.9
    Temperature: 23.4
    Timestamp: "2024-07-20 10:37:47.756"
▸ 1
▾ 2
    Humidity: 36.6
    Temperature: 23.2
    Timestamp: "2024-07-20 10:37:51.771"
▸ 3
▸ 4
▸ 5
▸ 6
▸ 7
▸ 8
▸ 9
```

Database location: Singapore (asia-southeast1)



Read the data

```python
dht_22 = ref.get()
display(dht_22)
display(type(dht_22))
```

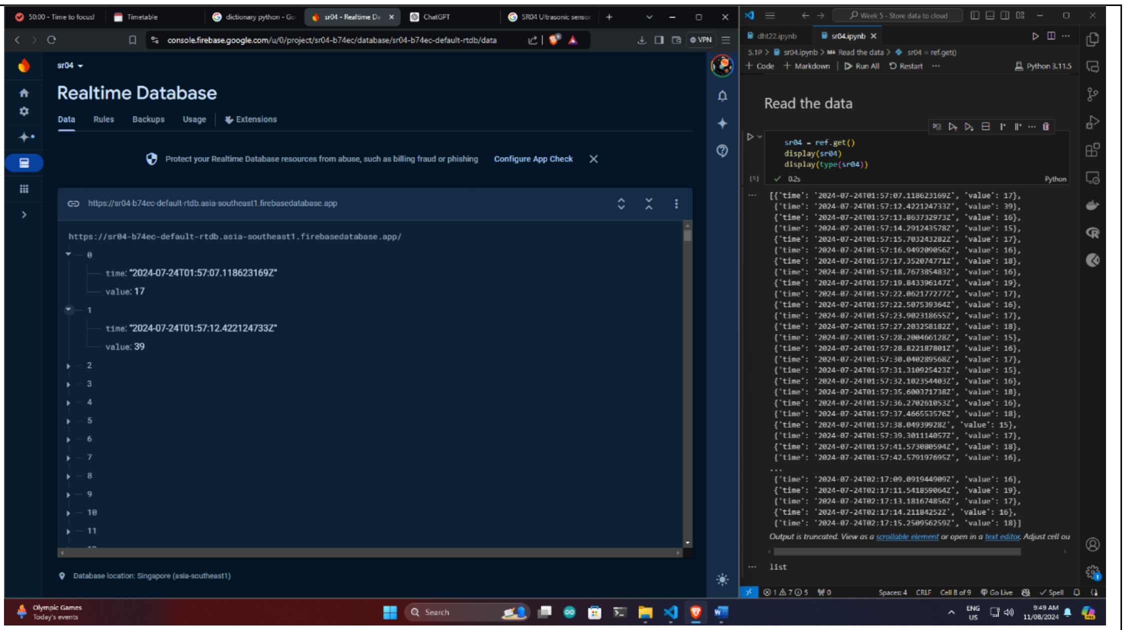
✓ 0.2s

```
[{'Humidity': 36.9,
  'Temperature': 23.4,
  'Timestamp': '2024-07-20 10:37:47.756'},
 {'Humidity': 36.8,
  'Temperature': 23.3,
  'Timestamp': '2024-07-20 10:37:49.765'},
 {'Humidity': 36.6,
  'Temperature': 23.2,
  'Timestamp': '2024-07-20 10:37:51.771'},
 {'Humidity': 36.4,
  'Temperature': 23.0,
  'Timestamp': '2024-07-20 10:37:53.778'},
 {'Humidity': 36.6,
  'Temperature': 23.0,
  'Timestamp': '2024-07-20 10:37:55.784'},
 {'Humidity': 37.0,
  'Temperature': 22.9,
  'Timestamp': '2024-07-20 10:37:57.792'},
 {'Humidity': 37.4,
  'Temperature': 22.8,
  'Timestamp': '2024-07-20 10:37:59.798'},
 {'Humidity': 37.8,
  'Temperature': 22.7,
  'Timestamp': '2024-07-20 10:38:01.805'},
 {'Humidity': 37.9,
...
  'Timestamp': '2024-07-20 11:11:10.540'},
 {'Humidity': 50.7,
  'Temperature': 19.5,
  'Timestamp': '2024-07-20 11:11:12.547'},
 ...]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
list
```

12	Question: Generate some random data for the SR04 Ultrasonic sensor, insert data to database, query all data and screenshot the output here. Answer: I will reuse the SR04 data from week 3

	
13	Question: Firebase Realtime database generates events on data operations. You can refer to section 'Handling Realtime Database events' in the document (https://firebase.google.com/docs/functions/database-events?gen=2nd). Discuss in the active learning session and summarise the idea of database events and how it is handled using Python SDK. Note that these events are useful when your sensors (from Arduino script) store data directly to Firebase Realtime database and you would like to track data update actions from a central Python application such as a monitoring dashboard. Answer: The idea of database event is to handle events without needing to update client code. There are 2 levels to handle Realtime database events. First one being to listen for specific operations like only write, create update or delete events. The second is to listen for any change of any kind to a reference.

```
https://sr04-b74ec-default-rtdb.asia-southeast1.firebasedatabase.app/
0
    time: "2024-07-24T01:57:07.118623169Z"
    value: 17
1
    time: "2024-07-24T01:57:12.422124733Z"
    value: 39
2
3
4
5
6
7
8
9
10
11
```

Database location: Singapore (asia-southeast1)

Read the data

```python
sr04 = ref.get()
display(sr04)
display(type(sr04))
```

```
[{'time': '2024-07-24T01:57:07.118623169Z', 'value': 17},
 {'time': '2024-07-24T01:57:12.422124733Z', 'value': 39},
 {'time': '2024-07-24T01:57:13.863732973Z', 'value': 16},
 {'time': '2024-07-24T01:57:14.291243578Z', 'value': 15},
 {'time': '2024-07-24T01:57:15.703243282Z', 'value': 17},
 {'time': '2024-07-24T01:57:16.949209056Z', 'value': 16},
 {'time': '2024-07-24T01:57:17.352074771Z', 'value': 18},
 {'time': '2024-07-24T01:57:18.767385483Z', 'value': 16},
 {'time': '2024-07-24T01:57:19.843396147Z', 'value': 19},
 {'time': '2024-07-24T01:57:22.062177277Z', 'value': 17},
 {'time': '2024-07-24T01:57:22.507539364Z', 'value': 16},
 {'time': '2024-07-24T01:57:23.902318655Z', 'value': 17},
 {'time': '2024-07-24T01:57:27.203258182Z', 'value': 18},
 {'time': '2024-07-24T01:57:28.200466128Z', 'value': 15},
 {'time': '2024-07-24T01:57:28.822187801Z', 'value': 16},
 {'time': '2024-07-24T01:57:30.040289568Z', 'value': 17},
 {'time': '2024-07-24T01:57:31.310925423Z', 'value': 15},
 {'time': '2024-07-24T01:57:32.102354403Z', 'value': 16},
 {'time': '2024-07-24T01:57:35.600371738Z', 'value': 18},
 {'time': '2024-07-24T01:57:36.270261053Z', 'value': 16},
 {'time': '2024-07-24T01:57:37.466553576Z', 'value': 18},
 {'time': '2024-07-24T01:57:38.04939928Z', 'value': 15},
 {'time': '2024-07-24T01:57:39.301114057Z', 'value': 17},
 {'time': '2024-07-24T01:57:41.573080594Z', 'value': 18},
 {'time': '2024-07-24T01:57:42.579197695Z', 'value': 16},
...
 {'time': '2024-07-24T02:17:09.091944909Z', 'value': 16},
 {'time': '2024-07-24T02:17:11.541859064Z', 'value': 19},
 {'time': '2024-07-24T02:17:13.181674856Z', 'value': 17},
 {'time': '2024-07-24T02:17:14.21184252Z', 'value': 16},
 {'time': '2024-07-24T02:17:15.250956259Z', 'value': 18}]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell ou
list
```

Activity 5.2: Data wrangling

Data wrangling is the process of converting raw data into a usable form. The process includes collecting, processing, analyzing, and tidying the raw data so that it can be easily read and analyzed. In this activity, you will use the common library in python, "pandas".

Hardware Required

No hardware is required.

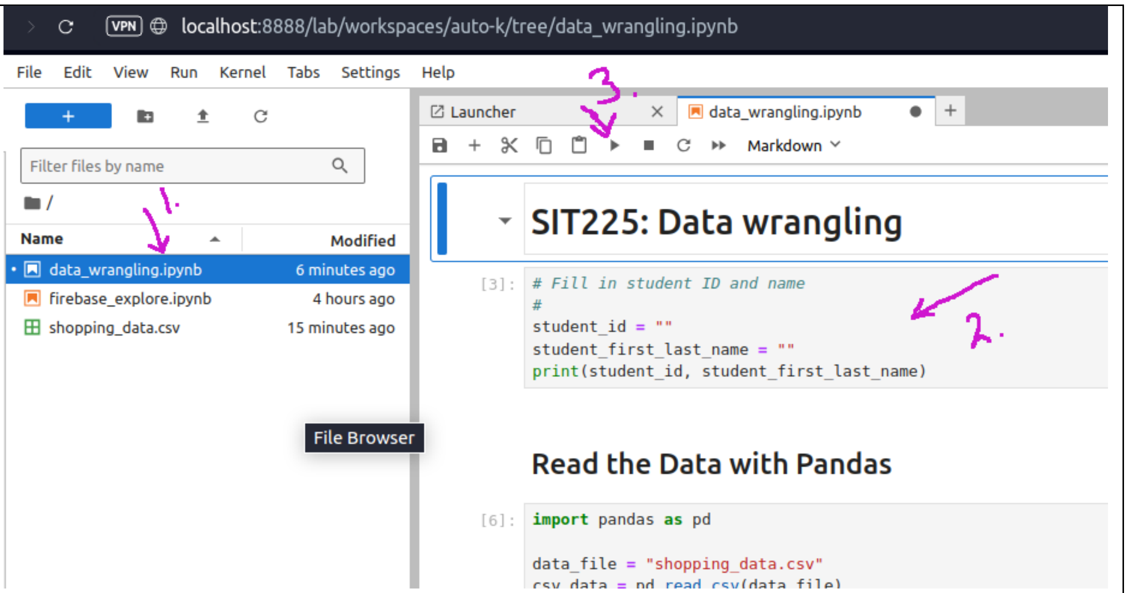
Software Required

Python 3
Pandas Python library

Steps

Step	Action

1	Install Pandas using the command below. Most likely you already have Pandas installed if you have installed Python using Anaconda disribution (https://www.anaconda.com/download).
	
	`$ pip install pandas`
	
	A Python notebook is shared in the GitHub link (https://github.com/deakin-deep-dreamer/sit225/tree/main/week_5). There will be a data_wrangling.ipynb, shopping_data.csv and shopping_data_missingvalue.csv files among others. Download the week_5 folder in your computer, open a command prompt in that folder, and write the command below in the command line:
	
	`$ jupyter lab`
	
	This will open Python Jupyter Notebook where in the left panel you can see the files (labeled as 1 in figure).

	
	 localhost:8888/lab/workspaces/auto-k/tree/data_wrangling.ipynb
	File Edit View Run Kernel Tabs Settings Help
	Filter files by name
	Name Modified
	data_wrangling.ipynb 6 minutes ago
	firebase_explore.ipynb 4 hours ago
	shopping_data.csv 15 minutes ago
	File Browser
	Launcher
	data_wrangling.ipynb
	Markdown
	1.
	2.
	3.
	SIT225: Data wrangling
	`[3]: # Fill in student ID and name`
	`#`
	`student_id = ""`
	`student_first_last_name = ""`
	`print(student_id, student_first_last_name)`
	Read the Data with Pandas
	`[6]: import pandas as pd`
	`data_file = "shopping_data.csv"`
	`csv_data = pd.read_csv(data_file)`
	Each cell contains Python code (labeled as 2 in figure), you can run a cell by clicking on the cell, so the cursor appears in that cell and then click on the play button at the top of the panel (labeled as 3 in the figure).

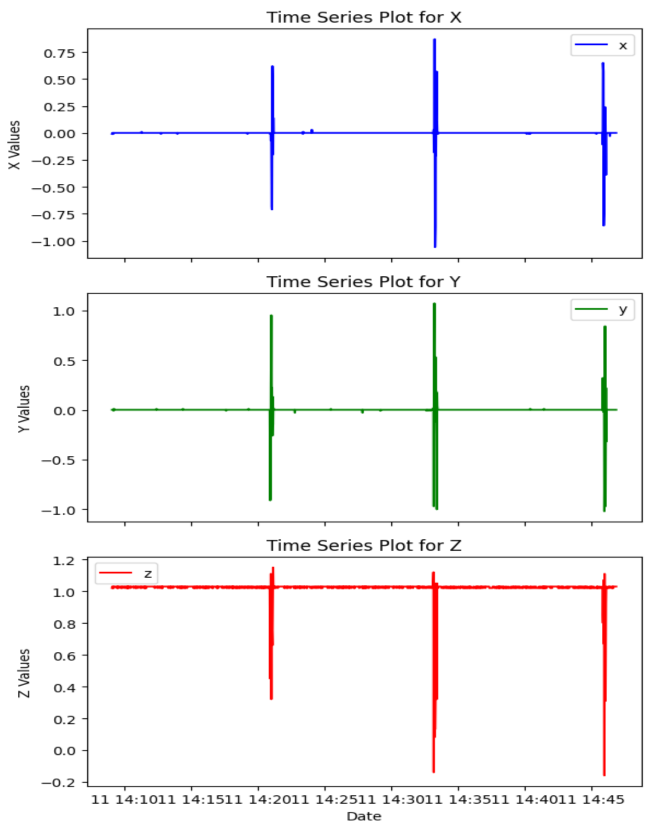
2	Question: Run each cell to produce output. Follow instructions in the notebook to complete codes in some of the cells. Convert the notebook to PDF from menu File > Save and Export Notebook As > PDF. Convert this activity sheet to PDF and merge with the notebook PDF. Answer: There is no answer to write here. You have to answer in the Jupyter Notebook.
3	Question: Once you went through the cells in the Notebook, you now have a basic understanding of data wrangling. Pandas are a powerful tool and can be used for reading CSV data. Can you use Pandas in reading sensor CSV data that you generated earlier? Describe if any modification you think necessary? Answer: Yes, I can, I can modify the header of the csv, I can drop some columns if I feel I don't need it, I can convert the csv to json file to upload to the firebase. There are many things I can do with Pandas, and it depends on the application.
4	Question: What do you understand of the Notebook section called Handling Missing Value? Discuss in group and briefly summarise different missing value imputation methods and their applicability on different data conditions.

	Answer: Handle missing values is when we decide what to do with values that are missing. One simplest method is just to drop it, granted if we have sufficient data. Other basic methods are to fill those missing data with mean and median. The choice between mean and median depends on our data. Mean or average is suitable for normally distributed data but are sensitive to outliers. Median or the middle value, suitable for skewed data but its less sensitive to small changes. The are other advanced techniques which use machine learning, but that will be beyond the scope of this task.

Pass Task: Store data to cloud

Q2.

I am simulating an earthquake at 10 minutes interval. I recorded the data for 37 minutes and here is the graph showing the outcome.

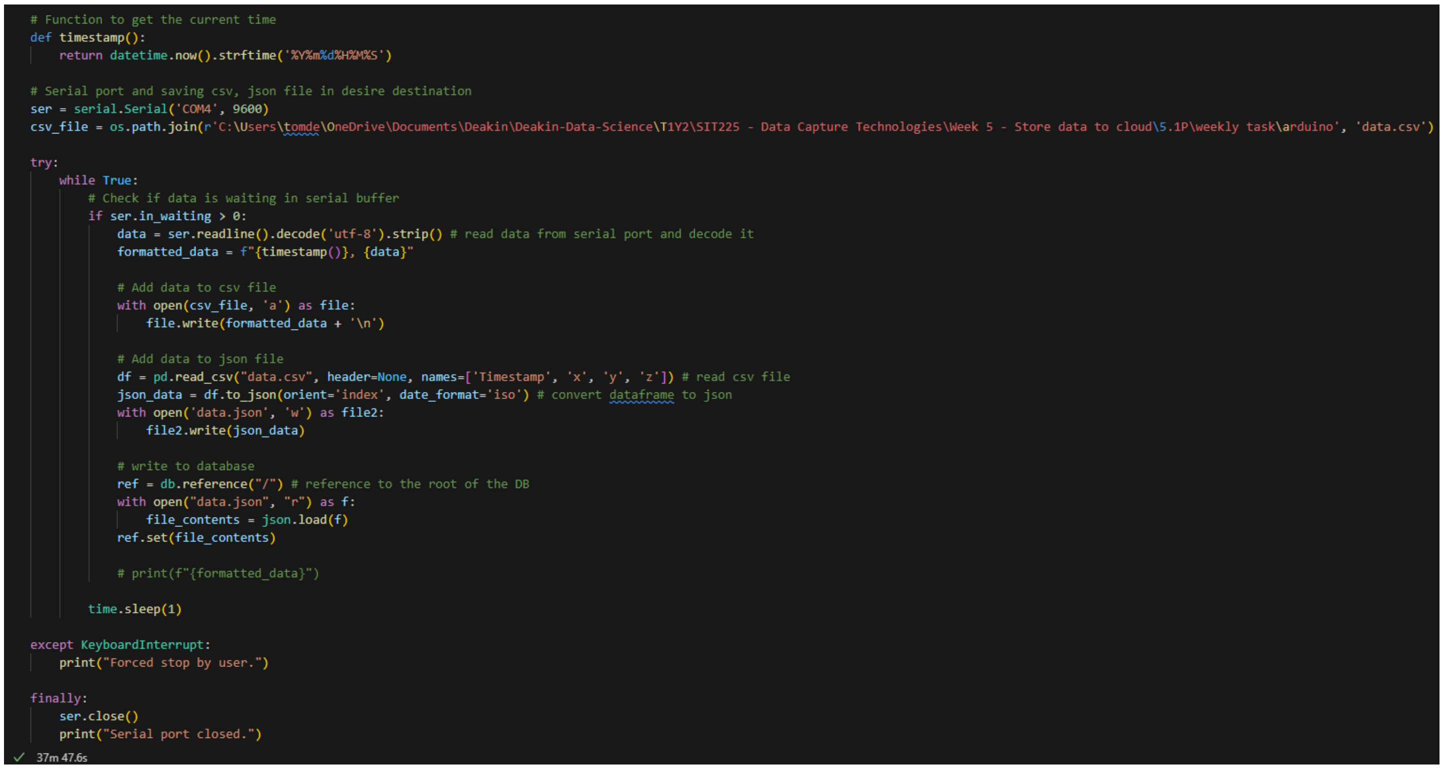


This graph shows the 3 axis or 3 rotations for every time the sensor is detecting large rotations.

Q3.

```
#include <Arduino_LSM6DS3.h>

float x, y, z;

void setup() {
  Serial.begin(9600); // set baud rate
  while (!Serial);  // wait for port to init
//  Serial.println("Started");

  if (!IMU.begin()) {
//    Serial.println("Failed to initialize IMU!");
    while (1);
  }

//  Serial.println(
//    "Accelerometer sample rate = "
//    + String(IMU.accelerationSampleRate()) + " Hz");
}

void loop() {
  // read accelero data
  if (IMU.accelerationAvailable()) {
    IMU.readAcceleration(x, y, z);
  }

  Serial.println(String(x) + "," + String(y) + "," + String(z));
  delay(1000); // delay 1s
}
```

The code above is the Arduino code for reading the rotations of each axis and sending it through the Serial.



```python
# Function to get the current time
def timestamp():
    return datetime.now().strftime('%Y%m%d%H%M%S')

# Serial port and saving csv, json file in desire destination
ser = serial.Serial('COM4', 9600)
csv_file = os.path.join(r'C:\Users\tomde\OneDrive\Documents\Deakin\Deakin-Data-Science\T1Y2\SIT225 - Data Capture Technologies\Week 5 - Store data to cloud\5.1P\weekly task\arduino', 'data.csv')

try:
    while True:
        # Check if data is waiting in serial buffer
        if ser.in_waiting > 0:
            data = ser.readline().decode('utf-8').strip() # read data from serial port and decode it
            formatted_data = f"{timestamp()}, {data}"

            # Add data to csv file
            with open(csv_file, 'a') as file:
                file.write(formatted_data + '\n')

            # Add data to json file
            df = pd.read_csv("data.csv", header=None, names=['Timestamp', 'x', 'y', 'z']) # read csv file
            json_data = df.to_json(orient='index', date_format='iso') # convert dataframe to json
            with open('data.json', 'w') as file2:
                file2.write(json_data)

            # write to database
            ref = db.reference("/") # reference to the root of the DB
            with open("data.json", "r") as f:
                file_contents = json.load(f)
            ref.set(file_contents)

            # print(f"{formatted_data}")

        time.sleep(1)

except KeyboardInterrupt:
    print("Forced stop by user.")

finally:
    ser.close()
    print("Serial port closed.")
```

```
✓ 37m 47.6s
```

This code runs continuously for 37 minutes and 47 seconds. It reads the data (x,y,z) from the Serial, then saves it into a csv and json file along with the timestamp. Using the json file, I continuously upload the data to the Firebase database.

Q4.

https://www.youtube.com/watch?v=eUz0PqXwnzU

Q5.

https://github.com/tomadonna1/SIT225_2024T2

Connect to database

```
student_id = "s223128143"
student_first_last_name = "Hoang Long Tran"
print(student_id, student_first_last_name)
```

```
s223128143 Hoang Long Tran
```

```
import firebase_admin
from firebase_admin import credentials

databaseURL = "https://week-5-63814-default-rtdb.firebaseio.com"
cred = credentials.Certificate("credentials.json")
firebase_admin.initialize_app(cred,
                              {'databaseURL':databaseURL})
```

```
<firebase_admin.App at 0x251cfb95cd0>
```

Write to the database

```
from firebase_admin import db

# reference to the root of the DB
ref = db.reference("/")

# json format data (key/value pair)
data = {  # Outer {} contains inner data structure
    "Book1":
    {
        "Title": "The Fellowship of the Ring",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book2":
    {
        "Title": "The Two Towers",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book3":
    {
        "Title": "The Return of the King",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book4":
    {
        "Title": "Brida",
        "Author": "Paulo Coelho",
        "Genre": "Fiction",
        "Price": 100
    }
}

# overwrite the the reference point at root node
ref.set(data)
```

Read the data

```
# query all data under ref
books = ref.get()
print(books)
print(type(books))
```

```
{'Book1': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The 
<class 'dict'>
```

```
# print each item separately
for key, value in books.items():
    print(f"{key}: {value}")
```

```
Book1: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Fel
Book2: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Two
Book3: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Retu
Book4: {'Author': 'Paulo Coelho', 'Genre': 'Fiction', 'Price': 100, 'Title': 'Brida'}
```

```
# Query /Book1
ref = db.reference("/Book1")
books = ref.get()
print(books)
```

```
{'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Fellowship
```

Write the database using `push()`

This method ensures that if multiple writes are being performed under the same key, they do not overwrite each other. The method uses unique keys for each new child added.

```
# Note that a set() is called on top of a push()
ref = db.reference("/")
ref.set({
    "Books":
    {
        "Best_Sellers": -1
    }
})

ref = db.reference("/Books/Best_Sellers")

for key, value in data.items():
    ref.push().set(value)
```

Update data

J.R.R. Tolkien books are reduced to 80 dollars to offer a discount

```
ref = db.reference("/Books/Best_Sellers/")
best_sellers = ref.get()
print(best_sellers)
```

```
{'-O3UyIWByTLMQkYqVjOR': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100,
```

```
for key, value in best_sellers.items():
    if(value["Author"] == "J.R.R. Tolkien"):
        ref.child(key).update({"Price":80})
```

Delete data

Delete all best seller books with J.R.R. Tolkien as the author

```
ref = db.reference("/Books/Best_Sellers")

for key, value in best_sellers.items():
    if(value["Author"] == "J.R.R. Tolkien"):
        ref.child(key).set({})


# If "ref.child" is not used, all the data will be removed. Like the code below
# ref = db.reference("/Books/Best_Sellers")
# ref.set({})
```

SIT225: Data wrangling

Run each cell to generate output and finally convert this notebook to PDF.

```
# Fill in student ID and name
#
student_id = "s223128143"
student_first_last_name = "Hoang Long Tran"
print(student_id, student_first_last_name)
```

```
s223128143 Hoang Long Tran
```

Read the Data with Pandas

Pandas has a dedicated function read_csv() to read CSV files.

Just in case we have a large number of data, we can just show into only five rows with head function. It will show you 5 rows data automatically.

```
import pandas as pd

data_file = "shopping_data.csv"
csv_data = pd.read_csv(data_file)

print(csv_data)

# show into only five rows with head function
print(csv_data.head())
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male   19                  15                      39
1             2    Male   21                  15                      81
```

```
2              3  Female   20                     16                     6
3              4  Female   23                     16                    77
4              5  Female   31                     17                    40
..           ...     ...  ...                    ...                   ...
195          196  Female   35                    120                    79
196          197  Female   45                    126                    28
197          198    Male   32                    126                    74
198          199    Male   32                    137                    18
199          200    Male   30                    137                    83

[200 rows x 5 columns]
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
```

Access the Column

Pandas has provided function .columns to access the column of the data source.

```
print(csv_data.columns)

# if we want to access just one column, for example "Age"
print("Age:")
print(csv_data["Age"])
```

```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
Age:
0      19
1      21
2      20
3      23
4      31
       ..
195    35
196    45
197    32
```

```
198    32
199    30
Name: Age, Length: 200, dtype: int64
```

Access the Row

In addition to accessing data through columns, using pandas can also access using rows. In contrast to access through columns, the function to display data from a row is the .iloc[i] function where [i] indicates the order of the rows to be displayed where the index starts from 0.

```
# we want to know what line 5 contains

print(csv_data.iloc[5])

print()

# We can combine both of those function to show row and column we want.
# For the example, we want to show the value in column "Age" at the first row
# (remember that the row starts at 0)
#
print(csv_data["Age"].iloc[1])
```

```
CustomerID                     6
Genre                     Female
Age                           22
Annual Income (k$)            17
Spending Score (1-100)        76
Name: 5, dtype: object

21
```

Show Data Based on Range

After displaying a data set, what if you want to display data from rows 5 to 20 of a dataset? To anticipate this, pandas can also display data within a certain range, both ranges for rows only, only columns, and ranges for rows and columns

```
print("Shows data to 5th to less than 10th in a row:")
print(csv_data.iloc[5:10])
```

```
Shows data to 5th to less than 10th in a row:
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
5           6  Female   22                  17                      76
6           7  Female   35                  18                       6
7           8  Female   23                  18                      94
8           9    Male   64                  19                       3
9          10  Female   30                  19                      72
```

Using Numpy to Show the Statistic Information

The describe() function allows to quickly find statistical information from a dataset. Those information such as mean, median, modus, max min, even standard deviation. Don't forget to install Numpy before using describe function.

```
print(csv_data.describe(include="all"))
```

```
        CustomerID   Genre         Age  Annual Income (k$)  \
count   200.000000     200  200.000000          200.000000
unique         NaN       2         NaN                 NaN
top            NaN  Female         NaN                 NaN
freq           NaN     112         NaN                 NaN
mean    100.500000     NaN   38.850000           60.560000
std      57.879185     NaN   13.969007           26.264721
min       1.000000     NaN   18.000000           15.000000
25%      50.750000     NaN   28.750000           41.500000
50%     100.500000     NaN   36.000000           61.500000
75%     150.250000     NaN   49.000000           78.000000
max     200.000000     NaN   70.000000          137.000000

        Spending Score (1-100)
count               200.000000
unique                     NaN
top                        NaN
freq                       NaN
mean                 50.200000
std                  25.823522
min                   1.000000
25%                  34.750000
50%                  50.000000
75%                  73.000000
max                  99.000000
```

Handling Missing Value

```
# For the first step, we will figure out if there is missing value.
print(csv_data.isnull().values.any())
print()
```

```
False
```

```
# We will use another data source with missing values to practice this part.
data_missing = pd.read_csv("shopping_data_missingvalue.csv")
print(data_missing.head())

print()

print("Missing? ", data_missing.isnull().values.any())
```

```
   CustomerID   Genre   Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male  19.0                15.0                    39.0
1           2    Male   NaN                15.0                    81.0
2           3  Female  20.0                 NaN                     6.0
3           4  Female  23.0                16.0                    77.0
4           5  Female  31.0                17.0                     NaN

Missing?  True
```

Ways to deal with missing values.

Follow the tutorial (https://deepnote.com/app/rickyharyanto14-3390/Data-Wrangling-w-Python-e5d1a23e-33cf-416d-ad27-4c3f7f467442). It includes - 1. Delete data * deleting rows * pairwise deletion * delete column 2. imputation * time series problem - Data without trend with seasonality (mean, median, mode, random) - Data with trend and without seasonality (linear interpolation) * general problem - Data categorical (Make NA as multiple imputation) - Data numerical or continuous (mean, median, mode, multiple imputation and linear regression)

Filling with Mean Values

The mean is used for data that has a few outliers/noise/anomalies in the distribution of the data and its contents. This value will later fill in the empty value of the dataset that has a missing value case. To fill in an empty value use the fillna() function

```
data_missing.dtypes
```

```
CustomerID                  int64
Genre                      object
Age                       float64
Annual Income (k$)        float64
Spending Score (1-100)    float64
dtype: object
```

```
# print(data_missing.mean())

"""

Question: This code will generate error. Can you explain why and how it can be solved?
Move on to the next cell to find one way it can be solved.

Answer: Because `Genre` column has object dtype, we need to decode it first, or just drop it

"""
```

```
'\n\nQuestion: This code will generate error. Can you explain why and how it can be solved? 
```

```
# Genre column contains string values and numerial operation mean fails.
# Lets drop Genre column since for numerial calculation.
#
data_missing_wo_genre = data_missing.drop(columns=['Genre'])
print(data_missing_wo_genre.head())
```

```
   CustomerID   Age  Annual Income (k$)  Spending Score (1-100)
0           1  19.0                15.0                    39.0
1           2   NaN                15.0                    81.0
2           3  20.0                 NaN                     6.0
3           4  23.0                16.0                    77.0
4           5  31.0                17.0                     NaN
```

```
print(data_missing_wo_genre.mean())
```

```
CustomerID                 100.500000
Age                         38.939698
Annual Income (k$)          61.005051
Spending Score (1-100)      50.489899
dtype: float64
```

```
print("Dataset with empty values! :")
print(data_missing_wo_genre.head(10))

data_filling=data_missing_wo_genre.fillna(data_missing_wo_genre.mean())
print("Dataset that has been processed Handling Missing Values with Mean :")
print(data_filling.head(10))

# Observe the missing value imputation in corresponding rows.
#
```

```
Dataset with empty values! :
   CustomerID   Age  Annual Income (k$)  Spending Score (1-100)
0           1  19.0                15.0                    39.0
1           2   NaN                15.0                    81.0
2           3  20.0                 NaN                     6.0
3           4  23.0                16.0                    77.0
4           5  31.0                17.0                     NaN
5           6  22.0                 NaN                    76.0
6           7  35.0                18.0                     6.0
7           8  23.0                18.0                    94.0
8           9  64.0                19.0                     NaN
9          10  30.0                19.0                    72.0
Dataset that has been processed Handling Missing Values with Mean :
   CustomerID        Age  Annual Income (k$)  Spending Score (1-100)
0           1  19.000000           15.000000               39.000000
1           2  38.939698           15.000000               81.000000
2           3  20.000000           61.005051                6.000000
3           4  23.000000           16.000000               77.000000
4           5  31.000000           17.000000               50.489899
5           6  22.000000           61.005051               76.000000
6           7  35.000000           18.000000                6.000000
7           8  23.000000           18.000000               94.000000
8           9  64.000000           19.000000               50.489899
9          10  30.000000           19.000000               72.000000
```

Filling with Median

The median is used when the data presented has a high outlier. The median was chosen because it is the middle value, which means it is not the result of calculations involving outlier data. In some cases, outlier data is considered disturbing and often considered noisy because it can affect class distribution and interfere with clustering analysis.

```
print(data_missing_wo_genre.median())
print("Dataset with empty values! :")
print(data_missing_wo_genre.head(10))

data_filling2=data_missing_wo_genre.fillna(data_missing_wo_genre.median())
print("Dataset that has been processed Handling Missing Values with Median :")
print(data_filling2.head(10))

# Observe the missing value imputation in corresponding rows.
#
```

```
CustomerID                 100.5
Age                         36.0
Annual Income (k$)          62.0
Spending Score (1-100)      50.0
dtype: float64
Dataset with empty values! :
   CustomerID   Age  Annual Income (k$)  Spending Score (1-100)
0           1  19.0                15.0                    39.0
1           2   NaN                15.0                    81.0
2           3  20.0                 NaN                     6.0
3           4  23.0                16.0                    77.0
4           5  31.0                17.0                     NaN
5           6  22.0                 NaN                    76.0
6           7  35.0                18.0                     6.0
7           8  23.0                18.0                    94.0
8           9  64.0                19.0                     NaN
9          10  30.0                19.0                    72.0
Dataset that has been processed Handling Missing Values with Median :
   CustomerID   Age  Annual Income (k$)  Spending Score (1-100)
0           1  19.0                15.0                    39.0
1           2  36.0                15.0                    81.0
2           3  20.0                62.0                     6.0
3           4  23.0                16.0                    77.0
4           5  31.0                17.0                    50.0
```

```
5           6  22.0                62.0                    76.0
6           7  35.0                18.0                     6.0
7           8  23.0                18.0                    94.0
8           9  64.0                19.0                    50.0
9          10  30.0                19.0                    72.0
```

Connect to database

```python
import serial
import time
from datetime import datetime
import os
import pandas as pd
from firebase_admin import db
import json
import firebase_admin
from firebase_admin import credentials
import matplotlib.pyplot as plt
```

```python
databaseURL = "https://gyroscope-ca247-default-rtdb.asia-southeast1.firebasedatabase.app/"

cred = credentials.Certificate("key.json")
firebase_admin.initialize_app(cred,
                              {'databaseURL':databaseURL})
```

```
<firebase_admin.App at 0x13e9b083a50>
```

Write to database

This code run continuously for 37 minutes and 47 seconds. The phenomena I want to capture with the Gyroscope sensor is an earth quake simulation (similar to that of the last credit task). During the 37 minutes, there are 3 earthquakes, at minute 10, at minute 20 and minute 30.

A bit of context for the code block below. It reads the data (x,y,z) from the Arduino via serial, then save it into a csv and json file along with the timestamp.

```
# Function to get the current time
def timestamp():
    return datetime.now().strftime('%Y%m%d%H%M%S')

# Serial port and saving csv, json file in desire destination
ser = serial.Serial('COM4', 9600)
csv_file = os.path.join(r'C:\Users\tomde\OneDrive\Documents\Deakin\Deakin-Data-Science\T1Y2\S

try:
    while True:
        # Check if data is waiting in serial buffer
        if ser.in_waiting > 0:
            data = ser.readline().decode('utf-8').strip() # read data from serial port and de
            formatted_data = f"{timestamp()}, {data}"

            # Add data to csv file
            with open(csv_file, 'a') as file:
                file.write(formatted_data + '\n')

            # Add data to json file
            df = pd.read_csv("data.csv", header=None, names=['Timestamp', 'x', 'y', 'z']) # r
            json_data = df.to_json(orient='index', date_format='iso') # convert dataframe to
            with open('data.json', 'w') as file2:
                file2.write(json_data)

            # write to database
            ref = db.reference("/") # reference to the root of the DB
            with open("data.json", "r") as f:
                file_contents = json.load(f)
            ref.set(file_contents)

            # print(f"{formatted_data}")

        time.sleep(1)

except KeyboardInterrupt:
    print("Forced stop by user.")

finally:
    ser.close()
    print("Serial port closed.")
```

```
Forced stop by user.
Serial port closed.
```

Query the Firebase data

```
firebase_data = ref.get()
display(firebase_data)
display(type(firebase_data))
```

```
[{'Timestamp': 20240811140905, 'x': -0.01, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811140906, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140907, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140909, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811140910, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140911, 'x': -0.01, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140912, 'x': 0.0, 'y': -0.01, 'z': 1.03},
 {'Timestamp': 20240811140913, 'x': 0.0, 'y': 0.01, 'z': 1.02},
 {'Timestamp': 20240811140914, 'x': 0.0, 'y': 0.01, 'z': 1.03},
 {'Timestamp': 20240811140915, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140917, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140918, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140919, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140920, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140921, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140922, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140924, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140925, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140926, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140927, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140928, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811140929, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140930, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811140932, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140933, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140934, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811140935, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140936, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140937, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140938, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811140940, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140941, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140942, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140943, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140944, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140945, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811140947, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811140948, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140949, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140950, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140951, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811140952, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140953, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140955, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140956, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140957, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140958, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811140959, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141000, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141002, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141003, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141004, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141005, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141006, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141007, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141008, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141010, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141011, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141012, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141013, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141014, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141015, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141017, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141018, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141019, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141020, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141021, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141022, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141023, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141025, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141026, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141027, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141028, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141029, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141030, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141031, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141033, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141034, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141035, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141036, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141037, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141039, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141040, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141041, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141042, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141043, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141044, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141045, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141047, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141048, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141049, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141050, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141051, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141052, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141054, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141055, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141056, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141057, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141058, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141059, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141100, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141102, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141103, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141104, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141105, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141107, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141108, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141109, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141110, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141111, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141112, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141114, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141115, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141116, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141117, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141118, 'x': 0.01, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141119, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141121, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141122, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141123, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141124, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141125, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141126, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141128, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141129, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141130, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141131, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141132, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141134, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141135, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141136, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141137, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141138, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141139, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141141, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141142, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141143, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141144, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141145, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141146, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141148, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141149, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141150, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141151, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141152, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141153, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141155, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141156, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141157, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141158, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141159, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141200, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141202, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141203, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141204, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141205, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141206, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141207, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141209, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141210, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141211, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141212, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141213, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141214, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141216, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141217, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141218, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141219, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141220, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141222, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141223, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141224, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141225, 'x': 0.0, 'y': 0.01, 'z': 1.03},
{'Timestamp': 20240811141226, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141227, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141229, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141230, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141231, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141232, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141233, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141234, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141236, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141237, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141238, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141239, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141240, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141241, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141243, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141244, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141245, 'x': -0.01, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141246, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141247, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141249, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141250, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141251, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141252, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141253, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141254, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141256, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141257, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141258, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141259, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141300, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141302, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141303, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141304, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141306, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141307, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141308, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141309, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141310, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141312, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141313, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141314, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141315, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141316, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141317, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141319, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141320, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141321, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141322, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141323, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141324, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141326, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141327, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141328, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141329, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141330, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141332, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141333, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141334, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141335, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141336, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141338, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141339, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141340, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141341, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141342, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141344, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141345, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141346, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141347, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141348, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141350, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141351, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141352, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141353, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141354, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141355, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141357, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141358, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141359, 'x': -0.01, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141400, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141401, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141403, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141404, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141405, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141406, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141407, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141408, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141410, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141411, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141412, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141413, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141414, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141416, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141417, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141418, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141419, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141420, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141421, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141423, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141424, 'x': 0.0, 'y': 0.01, 'z': 1.03},
{'Timestamp': 20240811141425, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141426, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141427, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141429, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141430, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141431, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141432, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141433, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141435, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141436, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141437, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141438, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141439, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141440, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141442, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141443, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141444, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141445, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141446, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141448, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141449, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141450, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141451, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141452, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141453, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141455, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141456, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141457, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141458, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141500, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141501, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141502, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141503, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141504, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141506, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141507, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141508, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141509, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141510, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141511, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141513, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141514, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141515, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141516, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141517, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141519, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141520, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141521, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141522, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141523, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141525, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141526, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141527, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141528, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141529, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141530, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141532, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141533, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141534, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141535, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141536, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141538, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141539, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141540, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141541, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141542, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141543, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141545, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141546, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141547, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141548, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141549, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141551, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141552, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141553, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141554, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141555, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141557, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141558, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141559, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141600, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141601, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141602, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141604, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141605, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141606, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141607, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141608, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141610, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141611, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141612, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141613, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141614, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141616, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141617, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141618, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141619, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141620, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141621, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141623, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141624, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141625, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141626, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141627, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141629, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141630, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141631, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141632, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141634, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141635, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141636, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141637, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141638, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141640, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141641, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141642, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141643, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141644, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141645, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141647, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141648, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141649, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141650, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141651, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141653, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141654, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141655, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141656, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141658, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141659, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141700, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141701, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141702, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141704, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141705, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141706, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141707, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141708, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141710, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141711, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141713, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141714, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141715, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141716, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141717, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141719, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141720, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141721, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141722, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141723, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141724, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141726, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141727, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141728, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141729, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141730, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141732, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141733, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141734, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141735, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141736, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141738, 'x': 0.0, 'y': -0.01, 'z': 1.03},
{'Timestamp': 20240811141739, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141740, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141741, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141742, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141743, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141745, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141746, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141747, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141748, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141749, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141751, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141752, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141753, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141754, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141755, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141756, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141758, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141759, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141800, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141801, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141802, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141804, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141805, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141806, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141807, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141808, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141810, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141811, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141812, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141813, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141814, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141815, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141817, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141818, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141819, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141820, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141821, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141823, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141824, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141825, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141826, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141827, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141829, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141830, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141831, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141832, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141833, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141835, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141836, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141837, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141838, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141839, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141840, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141842, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141843, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141844, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141845, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141846, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141848, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141849, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141850, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141851, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141852, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141854, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141855, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141856, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141857, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141858, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141900, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141901, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141902, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141903, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141904, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141905, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141907, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141908, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141909, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141910, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141911, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141912, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141914, 'x': -0.01, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141915, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141916, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141917, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141918, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141919, 'x': 0.0, 'y': 0.01, 'z': 1.03},
{'Timestamp': 20240811141921, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141922, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141923, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141924, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141925, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141927, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141928, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141929, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141930, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141931, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141932, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141934, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141935, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141936, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141937, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141938, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141939, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141941, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141942, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141943, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141944, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141945, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141947, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141948, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141949, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141950, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141951, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141952, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141954, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141955, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811141956, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141957, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811141958, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142000, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142001, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142002, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142003, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142004, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142005, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142007, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142008, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142009, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142010, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142011, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142013, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142014, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142015, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142016, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142018, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142019, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142020, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142021, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142022, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142024, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142025, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142026, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142027, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142028, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142029, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142031, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142032, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142033, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142034, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142035, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142037, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142038, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142039, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142040, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142041, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142042, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142044, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142045, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142046, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142047, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142048, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142050, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142051, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142052, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142053, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142054, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142056, 'x': -0.01, 'y': -0.91, 'z': 0.45},
{'Timestamp': 20240811142057, 'x': -0.01, 'y': -0.9, 'z': 0.47},
{'Timestamp': 20240811142058, 'x': -0.01, 'y': -0.82, 'z': 0.6},
{'Timestamp': 20240811142059, 'x': -0.07, 'y': 0.24, 'z': 1.11},
{'Timestamp': 20240811142101, 'x': 0.0, 'y': 0.95, 'z': 0.32},
{'Timestamp': 20240811142102, 'x': -0.36, 'y': 0.36, 'z': 0.82},
{'Timestamp': 20240811142103, 'x': -0.68, 'y': 0.11, 'z': 0.74},
{'Timestamp': 20240811142104, 'x': -0.71, 'y': 0.22, 'z': 0.75},
{'Timestamp': 20240811142106, 'x': 0.62, 'y': -0.26, 'z': 0.66},
{'Timestamp': 20240811142107, 'x': 0.52, 'y': -0.17, 'z': 0.8},
{'Timestamp': 20240811142108, 'x': 0.1, 'y': 0.13, 'z': 0.99},
{'Timestamp': 20240811142109, 'x': -0.2, 'y': -0.15, 'z': 1.15},
{'Timestamp': 20240811142110, 'x': 0.14, 'y': 0.02, 'z': 1.05},
{'Timestamp': 20240811142111, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142113, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142114, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142115, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142116, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142117, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142119, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142120, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142121, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142122, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142123, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142124, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142126, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142127, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142128, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142129, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142130, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142132, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142133, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142134, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142135, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142136, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142137, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142139, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142140, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142141, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142142, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142143, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142145, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142146, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142147, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142148, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142149, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142150, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142152, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142153, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142154, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142155, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142156, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142158, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142159, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142200, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142201, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142203, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142204, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142205, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142206, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142207, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142208, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142210, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142211, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142212, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142213, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142214, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142216, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142217, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142218, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142219, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142220, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142221, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142223, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142224, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142225, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142226, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142227, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142229, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142230, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142231, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142232, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142233, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142235, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142236, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142237, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142238, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142240, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142241, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142242, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142243, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142244, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142245, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142247, 'x': 0.0, 'y': -0.03, 'z': 1.02},
{'Timestamp': 20240811142248, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142249, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142250, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142251, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142253, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142254, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142255, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142256, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142258, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142259, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142300, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142301, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142302, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142304, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142305, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142306, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142307, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142308, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142310, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142311, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142312, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142313, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142314, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142316, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142317, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142318, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142319, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142321, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142322, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142323, 'x': -0.01, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142324, 'x': 0.01, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142325, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142327, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142328, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142329, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142330, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142332, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142333, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142334, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142335, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142336, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142338, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142339, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142340, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142341, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142342, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142344, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142345, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142346, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142347, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142348, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142350, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142351, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142352, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142353, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142355, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142356, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142357, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142358, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142359, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142401, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142402, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142403, 'x': 0.03, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142404, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142405, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142407, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142408, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142409, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142410, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142412, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142413, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142414, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142415, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142416, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142418, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142419, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142420, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142421, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142423, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142424, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142425, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142426, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142427, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142429, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142430, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142431, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142432, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142433, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142435, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142436, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142437, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142438, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142440, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142441, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142442, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142443, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142444, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142446, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142447, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142448, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142449, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142451, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142452, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142453, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142454, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142456, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142457, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142458, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142459, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142501, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142502, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142503, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142504, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142506, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142507, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142508, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142509, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142510, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142512, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142513, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142514, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142515, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142517, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142518, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142519, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142521, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142522, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142523, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142524, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142525, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142527, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142528, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142529, 'x': 0.0, 'y': 0.01, 'z': 1.03},
{'Timestamp': 20240811142530, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142532, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142533, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142534, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142535, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142536, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142538, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142539, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142540, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142541, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142543, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142544, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142545, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142546, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142547, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142549, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142550, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142551, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142552, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142553, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142555, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142556, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142557, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142558, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142600, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142601, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142602, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142603, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142604, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142606, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142607, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142608, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142609, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142611, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142612, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142613, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142614, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142615, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142617, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142618, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142619, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142620, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142622, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142623, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142624, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142625, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142626, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142628, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142629, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142630, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142631, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142633, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142634, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142635, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142636, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142637, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142639, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142640, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142641, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142642, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142644, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142645, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142646, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142647, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142648, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142650, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142651, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142652, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142653, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142655, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142656, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142657, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142658, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142700, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142701, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142702, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142703, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142705, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142706, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142707, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142708, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142709, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142711, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142712, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142713, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142714, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142716, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142717, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142718, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142719, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142720, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142722, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142723, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142724, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142725, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142727, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142728, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142729, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142730, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142731, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142733, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142734, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142735, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142736, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142738, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142739, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142740, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142741, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142743, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142744, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142745, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142746, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142747, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142749, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142750, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142751, 'x': 0.0, 'y': -0.03, 'z': 1.03},
{'Timestamp': 20240811142752, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142754, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142755, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142756, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142758, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142759, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142800, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142801, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142803, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142804, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142805, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142806, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142807, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142809, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142810, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142811, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142812, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142814, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142815, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142816, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142817, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142818, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142820, 'x': 0.0, 'y': 0.0, 'z': 1.02},
{'Timestamp': 20240811142821, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142822, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142823, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142825, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142826, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142827, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142828, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142830, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142831, 'x': 0.0, 'y': 0.0, 'z': 1.03},
{'Timestamp': 20240811142832, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142833, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142835, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142836, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142837, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142838, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142839, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142841, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142842, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142843, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142844, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142846, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142847, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142848, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142849, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142851, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142852, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142853, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142854, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142855, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142857, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142858, 'x': 0.0, 'y': 0.0, 'z': 1.03},
 {'Timestamp': 20240811142859, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 {'Timestamp': 20240811142900, 'x': 0.0, 'y': 0.0, 'z': 1.02},
 ...]
```

```
list
```

Since I already have a csv file saved when uploading data to the database, I will use it now.

```
df1 = pd.read_csv("data.csv", header=None, names=['Timestamp', 'x', 'y', 'z'])
display(df1.head())
display(df1.info())
```

	Timestamp	x	y	z
0	20240811140905	-0.01	0.0	1.02
1	20240811140906	0.00	0.0	1.03
2	20240811140907	-0.00	0.0	1.03
3	20240811140909	-0.00	0.0	1.02
4	20240811140910	-0.00	0.0	1.03

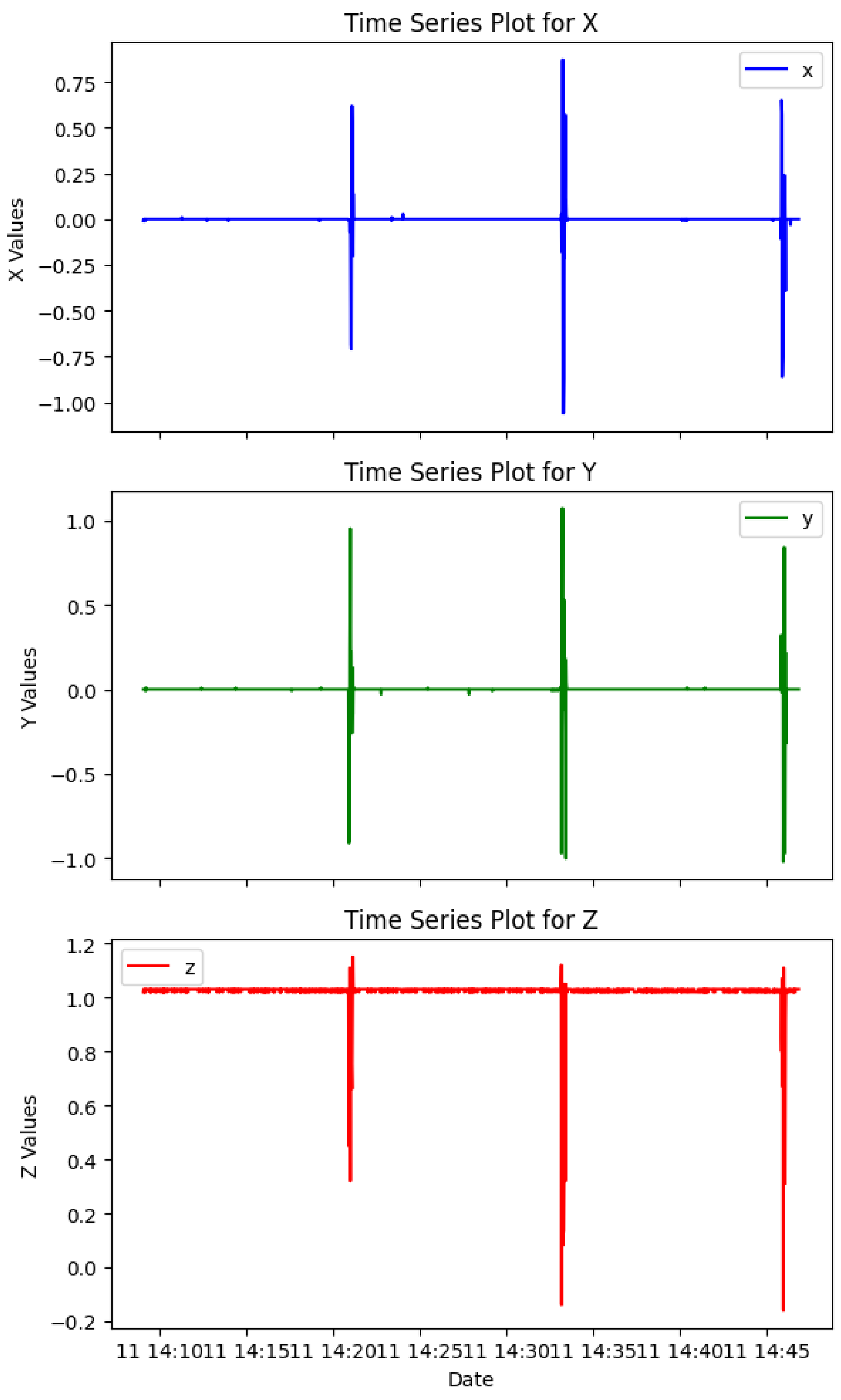
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1844 entries, 0 to 1843
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Timestamp  1844 non-null   int64
 1   x          1844 non-null   float64
 2   y          1844 non-null   float64
 3   z          1844 non-null   float64
dtypes: float64(3), int64(1)
memory usage: 57.8 KB

None
```

```
# Convert `Timestamp` column to time dtype
df['Timestamp'] = pd.to_datetime(df['Timestamp'], format='%Y%m%d%H%M%S')
df.head()
```

	Timestamp	x	y	z
0	2024-08-11 14:09:05	-0.01	0.0	1.02
1	2024-08-11 14:09:06	0.00	0.0	1.03
2	2024-08-11 14:09:07	-0.00	0.0	1.03
3	2024-08-11 14:09:09	-0.00	0.0	1.02
4	2024-08-11 14:09:10	-0.00	0.0	1.03

```
# Subplots
df.set_index('Timestamp', inplace=True)
fig, axs = plt.subplots(3, 1, figsize=(6, 10), sharex=True)

# Plot x
axs[0].plot(df.index, df['x'], label='x', color='b')
axs[0].set_title('Time Series Plot for X')
axs[0].set_ylabel('X Values')
axs[0].legend()

# Plot y
axs[1].plot(df.index, df['y'], label='y', color='g')
axs[1].set_title('Time Series Plot for Y')
axs[1].set_ylabel('Y Values')
axs[1].legend()
```

```python
# Plot z
axs[2].plot(df.index, df['z'], label='z', color='r')
axs[2].set_title('Time Series Plot for Z')
axs[2].set_xlabel('Date')
axs[2].set_ylabel('Z Values')
axs[2].legend()

# Display the plot
plt.tight_layout()
plt.show()
```


Time Series Plot for X
x
X Values
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Time Series Plot for Y
y
Y Values
1.0
0.5
0.0
−0.5
−1.0
Time Series Plot for Z
z
Z Values
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
11 14:1011 14:1511 14:2011 14:2511 14:3011 14:3511 14:4011 14:45
Date

X, Y, Z are rotation on each axis. The rotation of all three axis are mostly static, except at minute 10, 20 and 30, in which I shake the sensor to simulate earthquake. The pattern of earthquake is 10 minute interval, To be specific of the time, we need further analysis.

```
df.describe()
```

	x	y	z
count	1844.000000	1844.000000	1844.000000
mean	-0.001377	-0.000960	1.019555
std	0.062594	0.076862	0.071252
min	-1.060000	-1.020000	-0.160000
25%	-0.000000	0.000000	1.020000
50%	-0.000000	0.000000	1.030000
75%	0.000000	0.000000	1.030000
max	0.870000	1.070000	1.150000

50% percentile is the median value for the 3 rotations. I will filter plus minus 0.2 the median for each rotations.

```
median = df.describe().loc['50%']
median
```

```
x   -0.00
y    0.00
z    1.03
Name: 50%, dtype: float64
```

```
df[
    (df['x'] > (median.iloc[0] + 0.2)) | (df['x'] < (median.iloc[0] - 0.2)) |
    (df['y'] > (median.iloc[1] + 0.2)) | (df['y'] < (median.iloc[1] - 0.2)) |
    (df['z'] > (median.iloc[2] + 0.2)) | (df['z'] < (median.iloc[2] - 0.2))
]
```

	x	y	z
Timestamp			
2024-08-11 14:20:56	-0.01	-0.91	0.45
2024-08-11 14:20:57	-0.01	-0.90	0.47
2024-08-11 14:20:58	-0.01	-0.82	0.60

Timestamp	x	y	z
2024-08-11 14:20:59	-0.07	0.24	1.11
2024-08-11 14:21:01	0.00	0.95	0.32
2024-08-11 14:21:02	-0.36	0.36	0.82
2024-08-11 14:21:03	-0.68	0.11	0.74
2024-08-11 14:21:04	-0.71	0.22	0.75
2024-08-11 14:21:06	0.62	-0.26	0.66
2024-08-11 14:21:07	0.52	-0.17	0.80
2024-08-11 14:33:11	0.03	-0.97	-0.14
2024-08-11 14:33:12	-0.18	0.31	0.78
2024-08-11 14:33:14	-0.02	1.07	0.08
2024-08-11 14:33:15	0.87	-0.12	0.18
2024-08-11 14:33:16	-0.21	0.08	0.99
2024-08-11 14:33:17	-1.06	0.08	0.13
2024-08-11 14:33:19	-0.87	0.53	0.38
2024-08-11 14:33:20	-0.20	0.23	1.05
2024-08-11 14:33:21	-0.21	0.10	0.90
2024-08-11 14:33:24	0.57	-0.25	0.32
2024-08-11 14:33:25	0.05	-1.00	0.50
2024-08-11 14:45:51	-0.11	0.32	0.85
2024-08-11 14:45:52	0.65	-0.01	0.80
2024-08-11 14:45:54	0.57	-0.02	1.07
2024-08-11 14:45:55	-0.86	0.26	0.67
2024-08-11 14:45:57	-0.72	0.01	0.80
2024-08-11 14:45:58	-0.12	-1.02	-0.16
2024-08-11 14:45:59	-0.04	-0.69	1.11
2024-08-11 14:46:01	0.24	0.84	0.39
2024-08-11 14:46:02	-0.00	0.35	0.96
2024-08-11 14:46:03	0.12	-0.97	0.31
2024-08-11 14:46:05	-0.39	0.22	0.92
2024-08-11 14:46:06	-0.02	-0.32	1.03

The first movement is from 14:20:56 to 14:21:07. The second is 14:33:11 to 14:33:25. The last is 14:45:51 to 14:46:06. The interval is not quite exact 10 minutes, but around that.