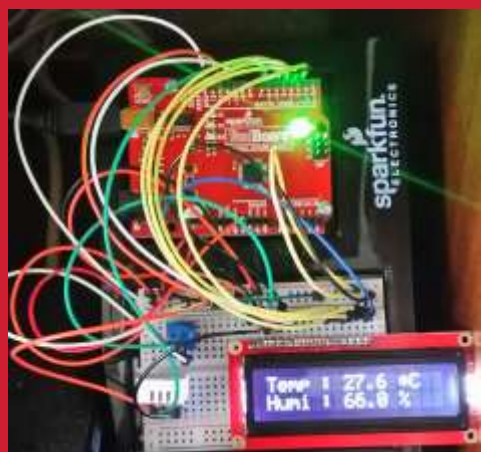




Arduino-IoT

[wk11]

cds_dht22 + node MongoDB Mining



Visualization of Signals using Arduino,
Node.js & storing signals in MongoDB
& mining iot data using Python



Drone-IoT-Comsi, INJE University
2nd semester, 2022

Email : chaos21c@gmail.com



My ID

ID를 확인하고 github에 repo 만들기

AA01	강대진	AA13	박제홍
		AA14	심준혁
AA03	김성우	AA15	이상혁
AA04	김정현	AA16	이승무
		AA17	이승준
AA06	김창연	AA18	이준희
AA07	김창욱	AA19	이현준
AA08	김태화	AA20	임태형
AA09	남승현	AA21	정동현
AA10	류재환		
AA11	박세훈	AA23	정희서
AA12	박신영	AA24	최재형

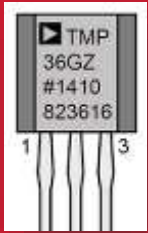
위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 - AAnn

Public, README.md check

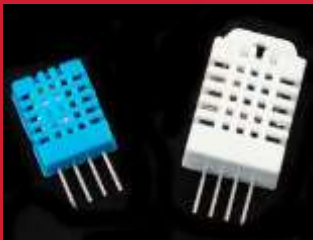


[Practice]



◆ [wk10]

- RT Data storaging with MongoDB
- Multi-sensor circuits (cds-dht22)
- Complete your project
- Upload folder: aann-rpt10



wk10 : Practice : aann-rpt10

◆ [Target of this week]

- Complete your works
- Save your outcomes and upload outputs in github

제출폴더명 : **aann-rpt10**

- 압축할 파일들

- ① **AAnn_iot_mongodb.png**
- ② **AAnn_iot_mongodb_web.png**
- ③ **AAnn_iot_json.png**
- ④ **AAnn_iot_client.png**
- ⑤ **All *.ino**
- ⑥ **All *.js**
- ⑦ **All *.html in public folder**
- ⑧ **Delete 'node_modules' folder**

Purpose of AA

주요 수업 목표는 다음과 같다.

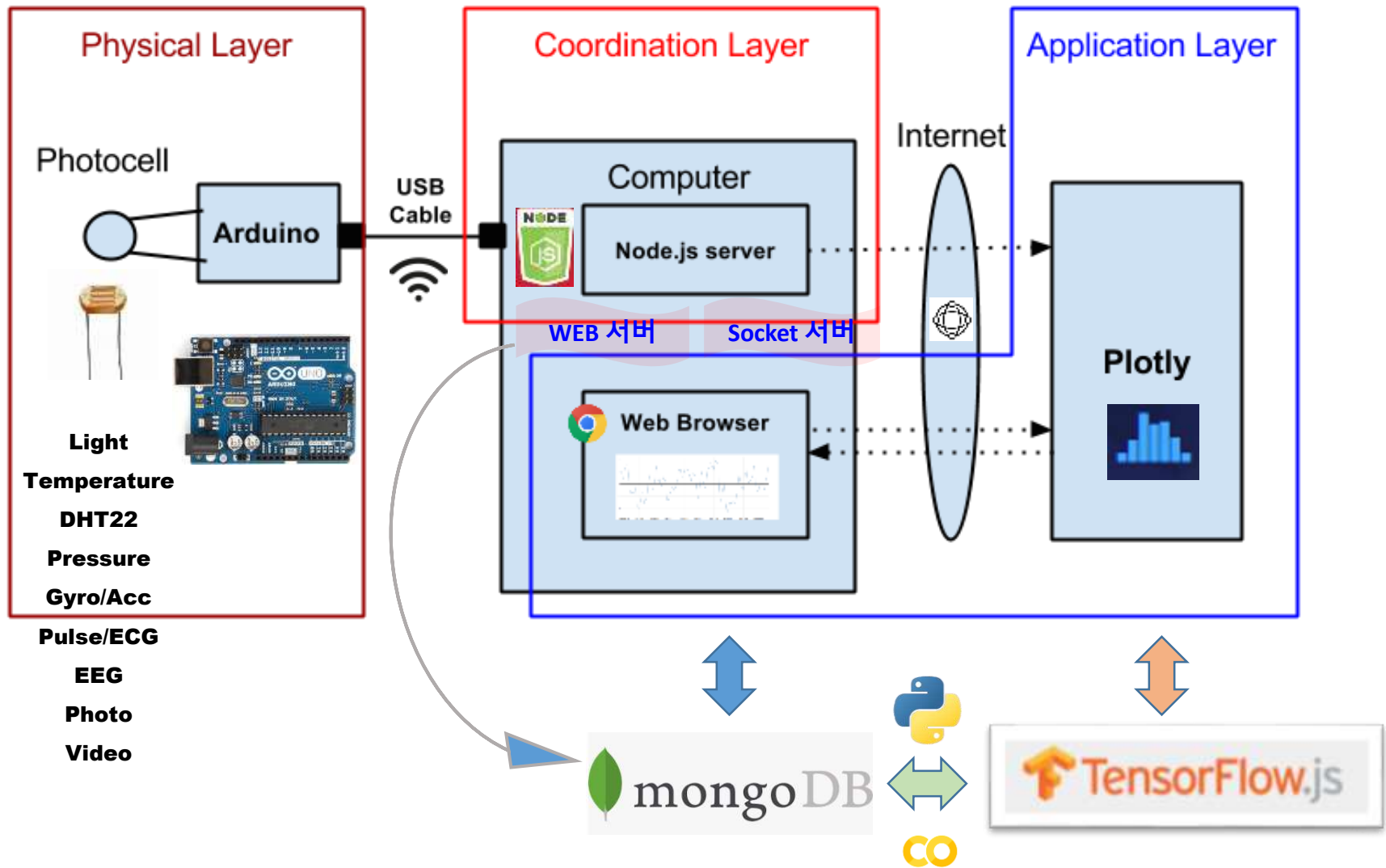
1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리



4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)



Layout [H S C]





DHT22 + CdS + Node.js + MongoDB

Network socket server : port=3000

```
D:\aann\aann-rpt12\cds_dht22>node cds_dht22_mongodb
serial port open
mongo db connection OK.
iotInfo: Current date: 2021-11-17 10:15:23.935, Temp: 21.1, Humi: 33.2, Lux: 87
iotInfo: Current date: 2021-11-17 10:15:26.213, Temp: 21.1, Humi: 33.2, Lux: 87
iotInfo: Current date: 2021-11-17 10:15:28.490, Temp: 21.1, Humi: 33.2, Lux: 87
iotInfo: Current date: 2021-11-17 10:15:30.767, Temp: 21.1, Humi: 33.2, Lux: 87
iotInfo: Current date: 2021-11-17 10:15:33.045, Temp: 21.1, Humi: 33.2, Lux: 87
iotInfo: Current date: 2021-11-17 10:15:35.322, Temp: 21.1, Humi: 33.2, Lux: 87
iotInfo: Current date: 2021-11-17 10:15:37.605, Temp: 21.1, Humi: 33.2, Lux: 87
```

█

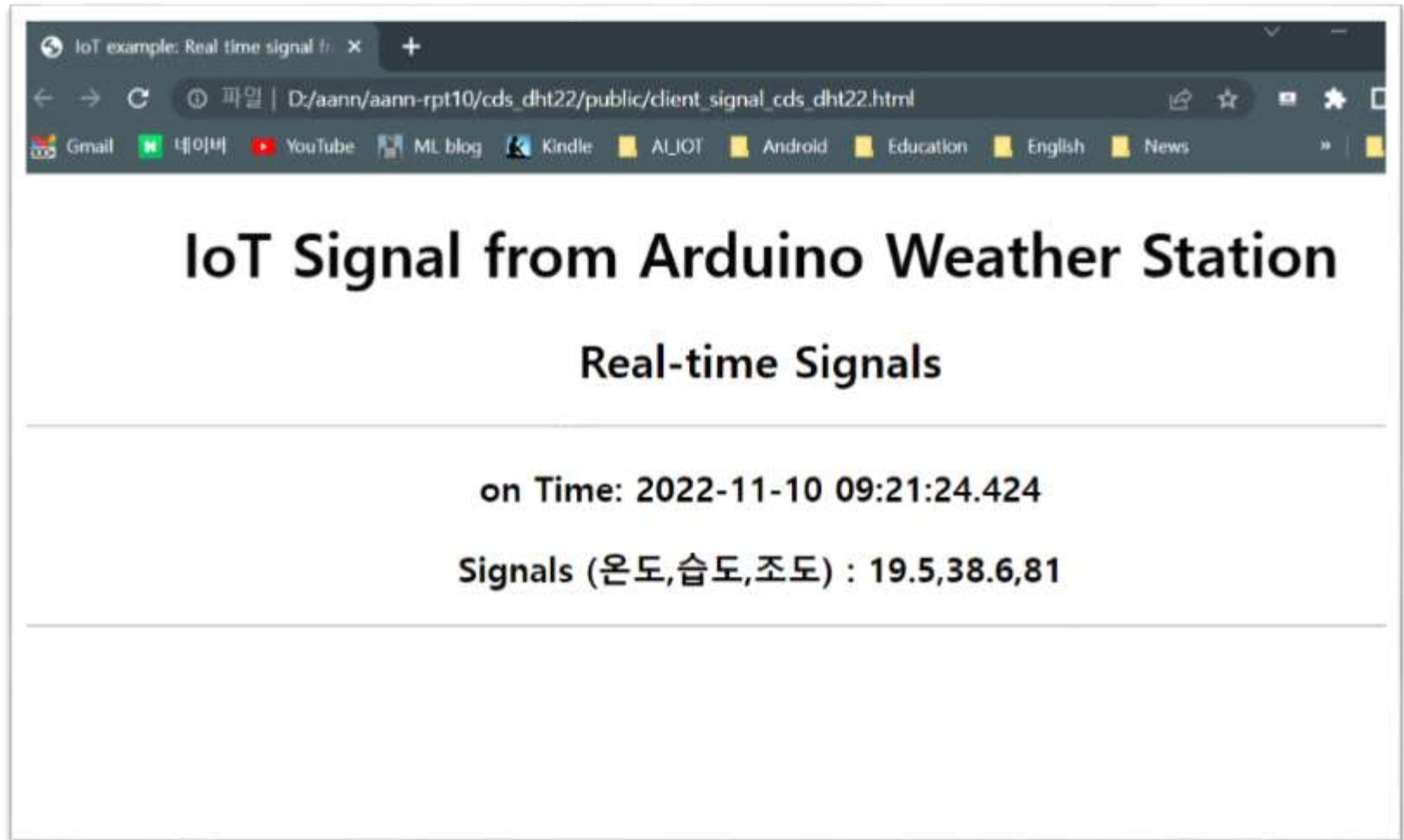
Express server : port=3030

```
D:\aann\aann-rpt12\cds_dht22>node cds_dht22_express
Express_IOT with CORS is running at port:3030
mongo db connection OK.
```

█



on WEB monitoring Arduino data



The screenshot shows a web browser window with a single tab titled "IoT example: Real time signal". The address bar displays the file path "D:/aann/aann-rpt10/cds_dht22/public/client_signal_cds_dht22.html". The browser's toolbar includes icons for Gmail, 네이버 (Naver), YouTube, ML blog, Kindle, AL_IOT, Android, Education, English, and News. The main content area features the heading "IoT Signal from Arduino Weather Station" in a large, bold, black font, followed by the subheading "Real-time Signals" in a slightly smaller, bold, black font. Below these headings, the text "on Time: 2022-11-10 09:21:24.424" is displayed. At the bottom, the text "Signals (온도,습도,조도) : 19.5,38.6,81" is shown, indicating the current temperature, humidity, and light intensity readings.

IoT example: Real time signal

← → ↻ Ⓞ 파일 | D:/aann/aann-rpt10/cds_dht22/public/client_signal_cds_dht22.html

Gmail 네이버 YouTube ML blog Kindle AL_IOT Android Education English News

IoT Signal from Arduino Weather Station

Real-time Signals

on Time: 2022-11-10 09:21:24.424

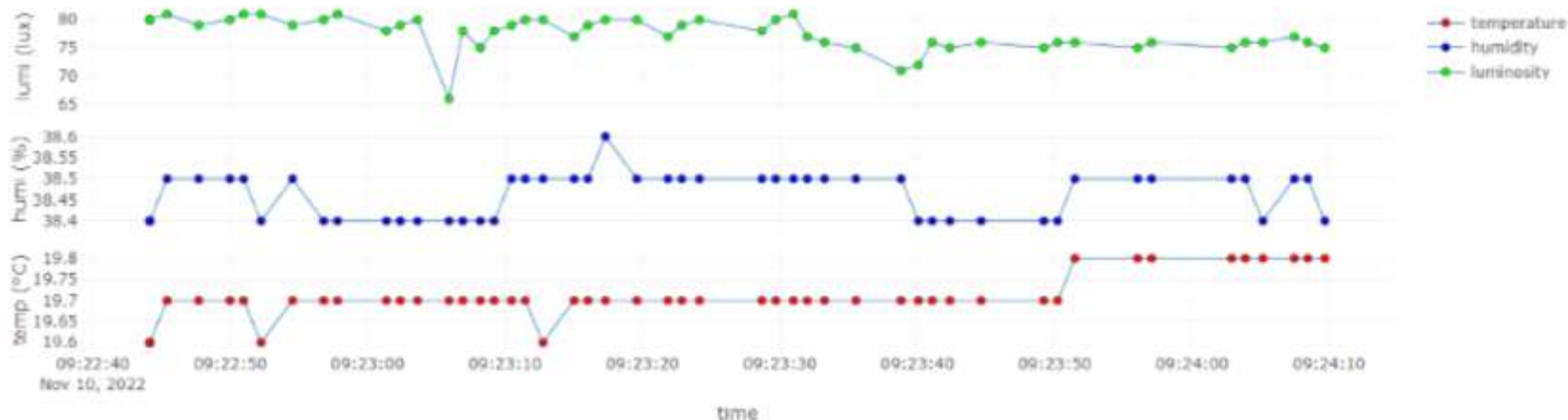
Signals (온도,습도,조도) : 19.5,38.6,81

Web monitoring: http://localhost:3030/client_cds_dht22.html

Real-time Weather Station from sensors



on Time: 2022-11-10 09:24:09.708



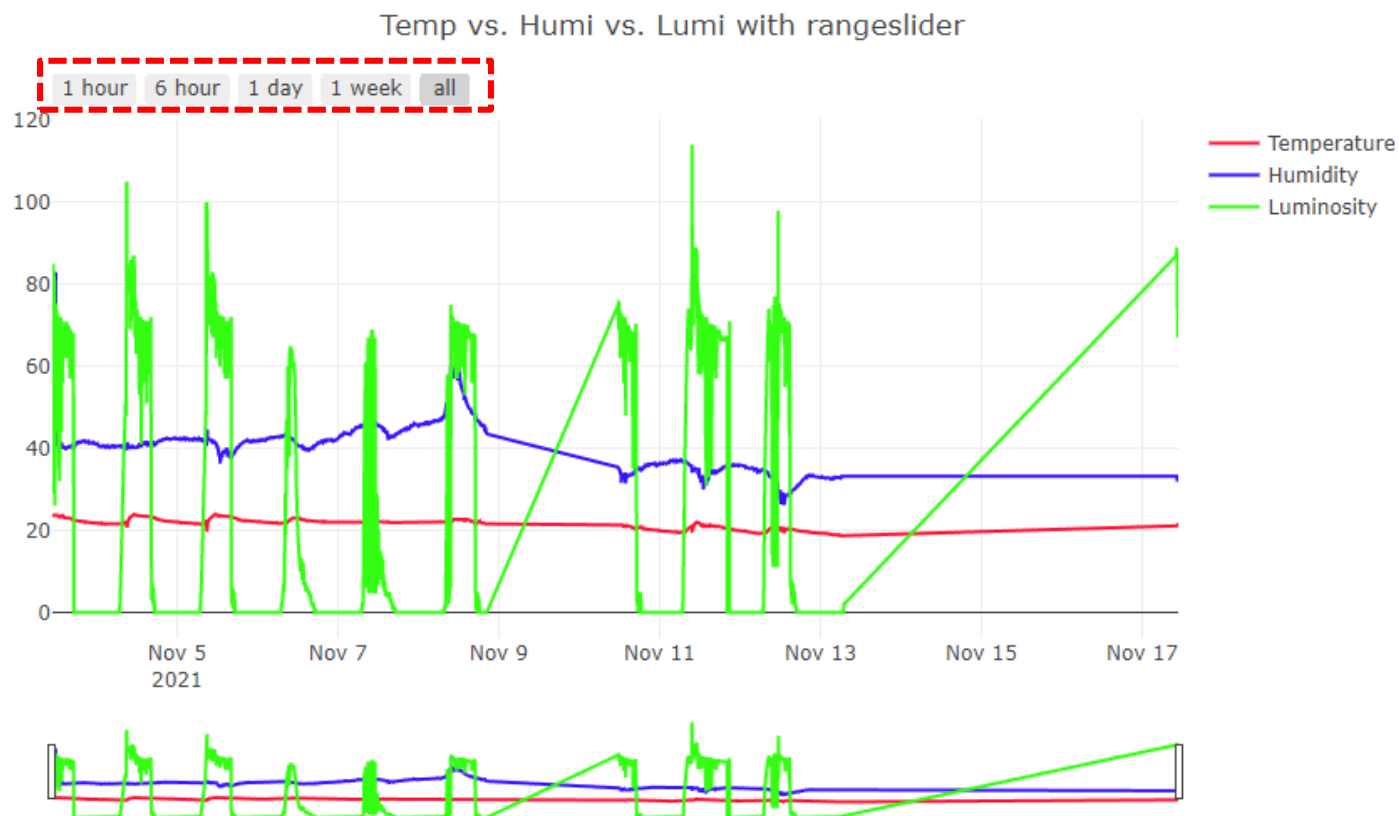
MongoDB shell: mongo

```
> show dbs
admin 0.000GB
config 0.000GB
iot 0.013GB
iot10 0.000GB
local 0.000GB
test 0.000GB
> use iot
switched to db iot
> show collections
sensors
> db.sensors.find().sort({'_id':-1}).limit(10)
{ "_id" : ObjectId("619466cc0374269b9cc2cd3b"), "date" : "2021-11-17 11:19:56.573", "temperature" : "21.6", "humidity" : "32.4", "luminosity" : "85",
  "_v" : 0 }
{ "_id" : ObjectId("619466ca0374269b9cc2cd39"), "date" : "2021-11-17 11:19:54.296", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "86",
  "_v" : 0 }
{ "_id" : ObjectId("619466c80374269b9cc2cd37"), "date" : "2021-11-17 11:19:52.018", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "85",
  "_v" : 0 }
{ "_id" : ObjectId("619466c50374269b9cc2cd35"), "date" : "2021-11-17 11:19:49.741", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "85",
  "_v" : 0 }
{ "_id" : ObjectId("619466c30374269b9cc2cd33"), "date" : "2021-11-17 11:19:47.463", "temperature" : "21.6", "humidity" : "32.3", "luminosity" : "85",
  "_v" : 0 }
{ "_id" : ObjectId("619466c10374269b9cc2cd31"), "date" : "2021-11-17 11:19:45.186", "temperature" : "21.6", "humidity" : "32.3", "luminosity" : "86",
  "_v" : 0 }
{ "_id" : ObjectId("619466be0374269b9cc2cd2f"), "date" : "2021-11-17 11:19:42.908", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "85",
  "_v" : 0 }
{ "_id" : ObjectId("619466bc0374269b9cc2cd2d"), "date" : "2021-11-17 11:19:40.630", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "86",
  "_v" : 0 }
{ "_id" : ObjectId("619466ba0374269b9cc2cd2b"), "date" : "2021-11-17 11:19:38.348", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "85",
  "_v" : 0 }
{ "_id" : ObjectId("619466b80374269b9cc2cd29"), "date" : "2021-11-17 11:19:36.071", "temperature" : "21.7", "humidity" : "32.4", "luminosity" : "85",
  "_v" : 0 }
```

Web monitoring: http://localhost:3030/client_iotdb.html

MongoDB database visualization by AA00

Time series : Multi sensor data





A5. Introduction to IoT service

System (Arduino, sDevice, ...)



Data (signal, image, sns, ...)



Visualization & monitoring



Data storing & Mining



Service



MongoDB data management

- Query in mongo shell
- Export & import MongoDB
- Using and understanding iot data with Python (or R)



A5.9.8 MongoDB management

1. Query in Mongo shell (문서 검색)

`db.sensors.count()` → sensors collection에 있는 도큐먼트 (문서)의 수

`db.sensors.find().sort({_id: 1}).limit(10)` → 오래된 document 10개 추출

`db.sensors.find().sort({_id: -1}).limit(10)` → 최근 document 10개 추출

`db.sensors.find({date: {$gt: "2022-11-10 14:36:05"}})` → 특정 시간 이후 document 추출

`db.sensors.find({temperature: {$gt: "25"}})` → 온도가 25도를 넘는 document 추출

<https://docs.mongodb.com/manual/tutorial/query-documents/>



A5.9.8 MongoDB management

2. Import or export MongoDB (VSCode 터미널에서 실행)

- **mongoimport** --d=dbName --c=collectionName --type=csv --headerline --file= fileName.csv
- **mongoexport** --d=dbName --c=collectionName --fields=<field1,field2,...> --limit=nn --type=csv --out=fileName.csv

json 또는 csv 파일로 import/export

[Help] mongoimport --help

<https://docs.mongodb.com/manual/reference/program/mongoimport/>

<https://docs.mongodb.com/manual/reference/program/mongoexport/>



A5.9.8 MongoDB Database Tool

TOOLS

MongoDB Command Line Database Tools Download

The MongoDB Database Tools are a collection of command-line utilities for working with a MongoDB deployment. These tools release independently from the MongoDB Server schedule enabling you to receive more frequent updates and leverage new features as soon as they are available. See the [MongoDB Database Tools](#) documentation for more information.

Version

100.6.1



Platform

Windows x86_64



Package

zip



Download

More Options ...

➤ D:\mongodb 에
db tool의
'bin' folder를 복사하고
path 설정

[참고] mongodb 4.4부터 db tool 분리됨.

<https://www.mongodb.com/try/download/database-tools>



A5.9.8 MongoDB management

➤ **mongoimport** /db:dbName /collection:collectionName /type:csv /headerline /file: fileName.csv

```
D:\aann\aann-rpt11\cds_dht22\public\data>mongoimport /db:iot10 /collection:sensors /type:csv
/headerline /file:sensor10.csv
2022-11-07T18:47:51.681+0900    connected to: mongodb://localhost/
2022-11-07T18:47:51.740+0900    10 document(s) imported successfully. 0 document(s) failed to
import.
```

Import sensor10.csv to MongoDB: iot10

```
date,temperature,humidity,luminosity
2018-03-12 04:34:40.662,16.6,24.9,0
2018-03-12 04:34:50.923,16.6,24.9,0
2018-03-12 04:35:01.168,16.6,24.9,0
2018-03-12 04:35:11.429,16.6,24.9,0
2018-03-12 04:35:21.674,16.6,24.9,0
2018-03-12 04:35:31.935,16.6,24.9,0
2018-03-12 04:35:42.180,16.6,24.9,0
2018-03-12 04:35:52.441,16.6,24.9,0
2018-03-12 04:36:02.686,16.6,25.0,0
2018-03-12 04:36:12.947,16.6,25.0,0
```



```
> show dbs
admin    0.000GB
config  0.000GB
iot      0.009GB
iot10    0.000GB
local    0.000GB
test     0.000GB
> use iot10
switched to db iot10
```



A5.9.8 MongoDB management

[Tip] **iot db**의 최근 데이터 **500**개를 **csv** 파일 (**iot_s500.csv**)로 저장할 때,

➤ **mongoexport /db:iot /collection:sensors /sort:"{_id: -1}" /limit:500 /fields:date,temperature,humidity,luminosity /type:csv /out:iot_s500.csv**

```
D:\aann\aann-rpt11\cds_dht22\public\data>mongoexport /db:iot /collection:sensors
/sort:"{_id: -1}" /limit:500 /fields:date,temperature,humidity,luminosity /type
:csv /out:iot_s500.csv
2021-11-10T13:08:45.875+0900    connected to: mongodb://localhost/
2021-11-10T13:08:45.941+0900    exported 500 records
```

```
D:\aann\aann-rpt11\cds_dht22\public\data>dir
```

D 드라이브의 볼륨: DATA

볼륨 일련 번호: 82D1-4852

D:\aann\aann-rpt11\cds_dht22\public\data 디렉터리

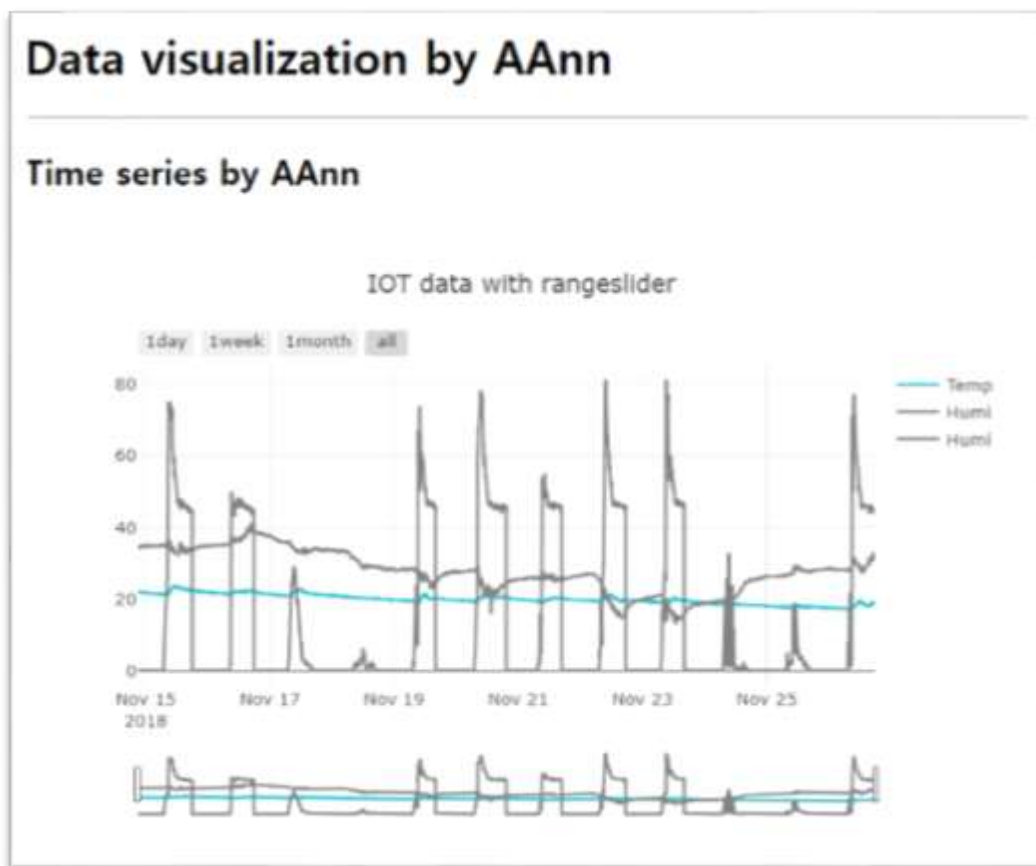
2021-11-10	오후 01:08	<DIR>	.
2021-11-10	오후 01:08	<DIR>	..
2017-11-16	오전 09:58	60,220	aapl.csv
2018-11-26	오후 05:50	3,628,861	iot_chaos.csv
2021-11-10	오후 01:08	18,537	iot_s500.csv
2017-11-16	오후 01:18	135,008	ppg5k.csv
2018-05-26	오후 12:52	397	sensor10.csv
5개 파일		3,843,023	바이트
2개 디렉터리		2,410,432,798,720	바이트 남음



A5.9.8 MongoDB management

[DIY]

1. **iot db**의 최근 데이터 2000개를 **csv** 파일 ([AAnn_s2000.csv](#))로 저장하시오.
2. 저장된 **AAnn_s2000.csv** 파일을 **public/data** 폴더에 복사.
3. **csv** 파일을 이용하는 **Rangeslider**가 포함된 웹 클라이언트 [client_iot.html](#) 파일을 만드시오.
4. **localhost:3030/client_iot.html** 로 실행하고 확인.



iot_chaos.html로

client_iot.html로

이름 변경해서 코드를 완성하시오.



3. How to use and understand iot data? → Python(or R) in Colab/Jupyter lab

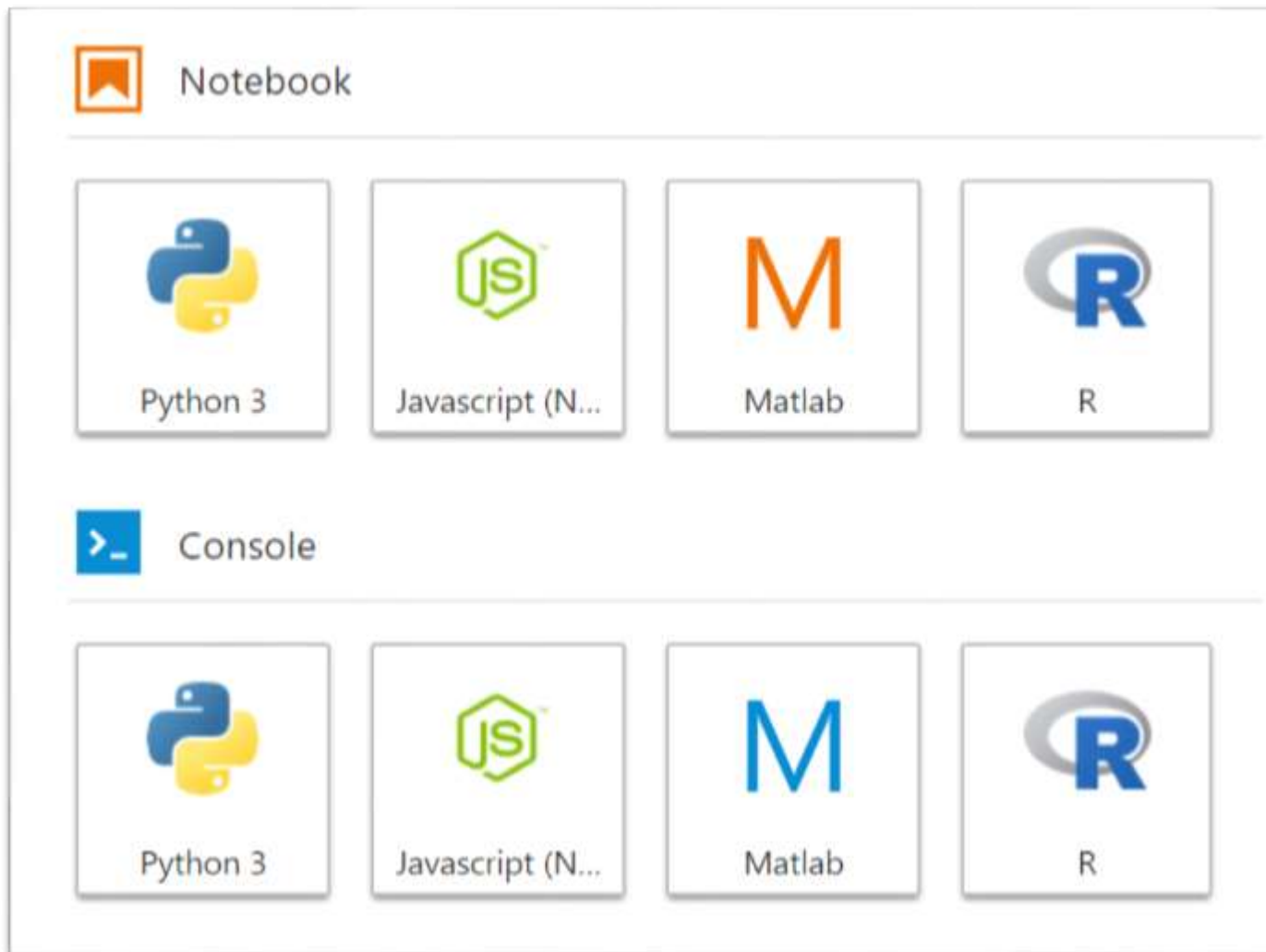


IoT Data Mining



IoT data mining

3. How to use and understand iot data? → Python(or R) in Colab/Jupyter lab





IoT data mining

How to use and understand iot data? → [Google Colab](#)

 Open in Colab

Pandas: access to the remote json from MongoDB

- The json file is generated on the fly from the express server of Node.js.
- The data stored in MongoDB are saved in the json file.
- The data are composed of three time series; temperature, humidity, and luminosity.

```
In [0]: import pandas as pd
```

```
In [0]: # loading json file from MongoDB via web (CORS, port=3030)
url="http://chaos.inje.ac.kr:3030/iot"
df=pd.read_json(url)
print('Large data was retrieved successfully from MongoDB!')
```

```
In [0]: df.head()
```



A5.9.8 IOT data mining

3.1 How to use and understand iot data? → [iot_csv.ipynb](#), [iot_json.ipynb](#)

Redwoods / Arduino

Code Issues 0 Pull requests 0 Projects 0 Wiki

Branch: master ▾ **Arduino / ar-iot / py-pandas /**

Redwoods Add files via upload

..

data	Add files via upload
iot_csv.ipynb	Colaboratory를 통해 생성됨
iot_json.ipynb	Colaboratory를 통해 생성됨



A5.9.8 MongoDB management

3.2 Loading data ... → `iot_json.ipynb`

```
[1] 1 | import pandas as pd
```

```
[2] 1 | # loading json file from MongoDB via web (CORS, port=3030)  
2 | url="http://chaos.inje.ac.kr:3030/iot"  
3 | j1=pd.read_json(url)
```

1. Express 서버에서 MongoDB에 접속한다.
2. 아두이노에서 만들어져 전송되어 MongoDB에 저장되고 있는 센서 데이터를 json 파일로 가져온다.

```
[3] 1 | j1.head()
```



	__v	_id	date	humidity	luminosity	temperature
0	0	5bce24218d1ec32774d781a9	2018-10-23 04:25:21.349	39.7	0	23.2
1	0	5bce242b8d1ec32774d781aa	2018-10-23 04:25:31.594	39.7	0	23.2
2	0	5bce24358d1ec32774d781ab	2018-10-23 04:25:41.855	39.7	0	23.2
3	0	5bce24408d1ec32774d781ac	2018-10-23 04:25:52.100	39.7	0	23.2
4	0	5bce244a8d1ec32774d781ad	2018-10-23 04:26:02.360	39.7	0	23.2



A5.9.8 IOT data mining

3.3 Make dataframe from json data

▼ Dataframe with date and three sensor values(temperature, humidity, luminosity)

```
[ ] 1 | iot_data = j1[['date', 'temperature', 'humidity', 'luminosity']]
```

```
[ ] 1 | iot_data.shape
```

Json 객체에서 필요한 항목을
선택해서 **pandas의 dataframe**을
구성한다.

(340230, 4)

```
[ ] 1 | iot_data.head()
```



	date	temperature	humidity	luminosity
0	2018-10-23 04:25:21.349	23.2	39.7	0
1	2018-10-23 04:25:31.594	23.2	39.7	0
2	2018-10-23 04:25:41.855	23.2	39.7	0
3	2018-10-23 04:25:52.100	23.2	39.7	0
4	2018-10-23 04:26:02.360	23.2	39.7	0



A5.9.8 IOT data mining

3.4.1 Plot iot data (time series)

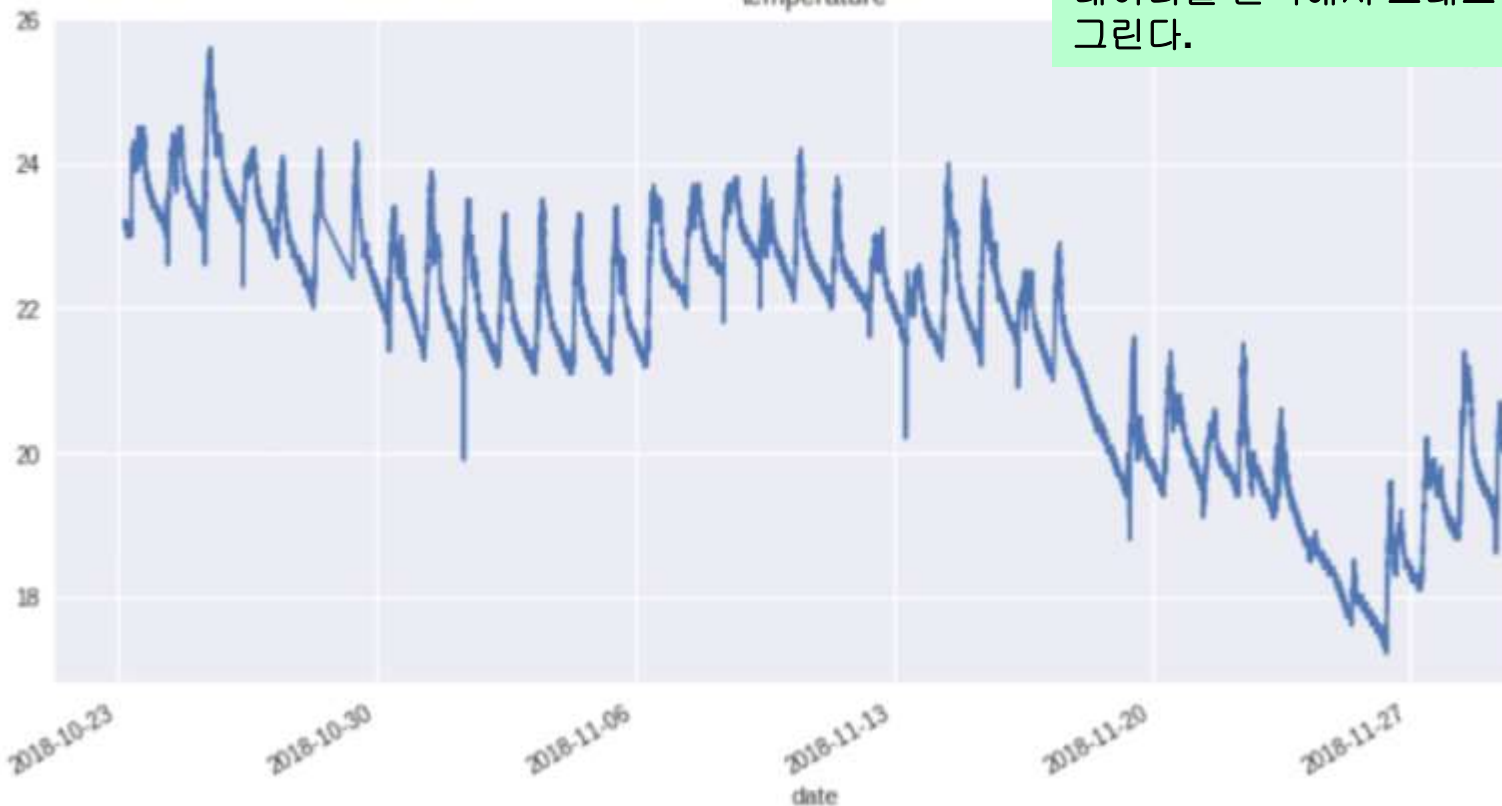
Plot time series of sensor data

```
[ ] | iot_data.plot(x='date', y='temperature', figsize=(12,6), title='temperature')
```



<matplotlib.axes._subplots.AxesSubplot at 0x7f5b2596e438>

temperature



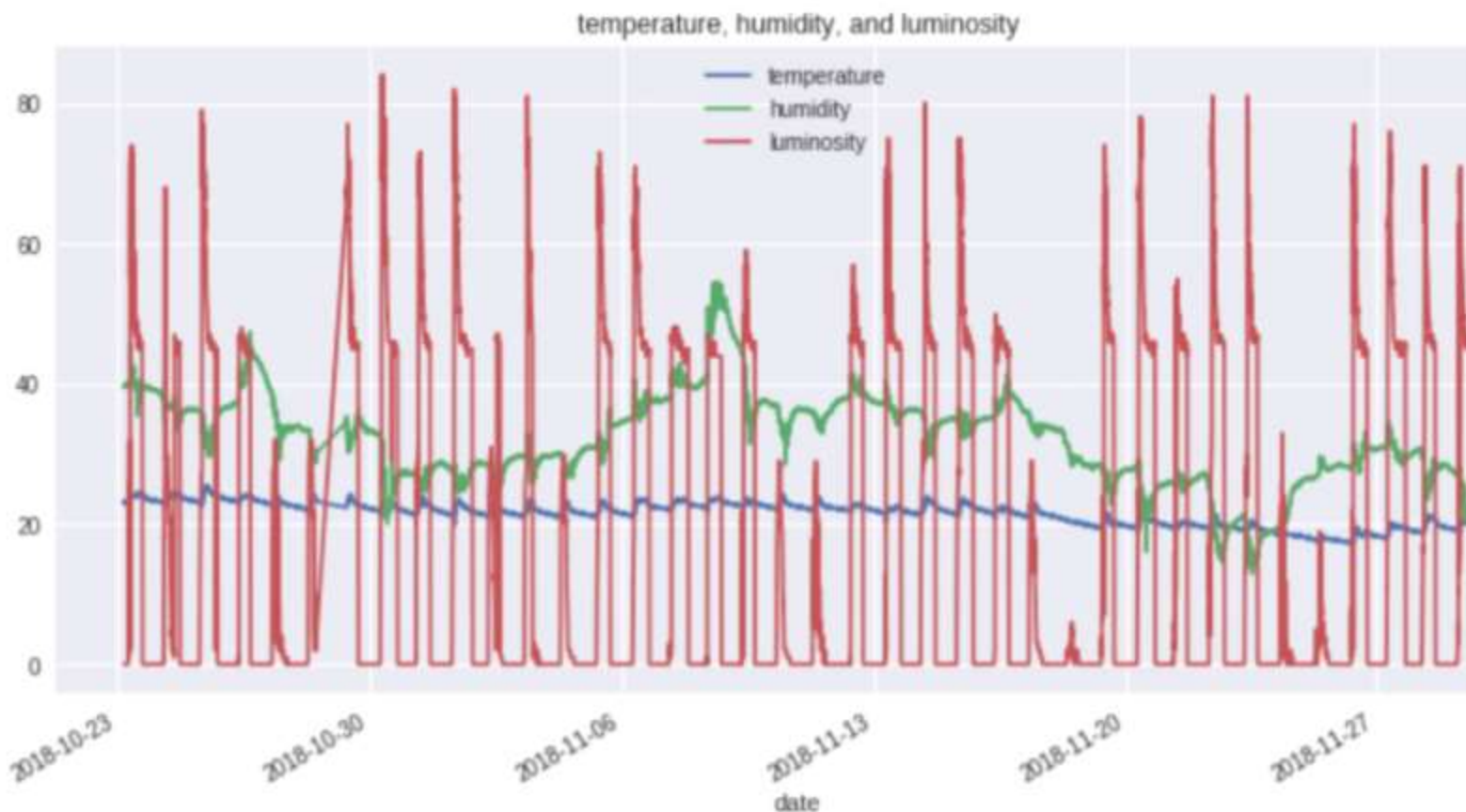
Dataframe에서 시간과 온도 데이터를 선택해서 그래프를 그린다.

3.4.2 Plot iot data (time series)

```
1 # Plot of ['temperature', 'humidity', 'luminosity']
2 iot_data.plot(x='date', y=['temperature', 'humidity', 'luminosity'], figsize=(12,6),
3               title='temperature, humidity, and luminosity')
```

```
/usr/local/lib/python3.6/dist-packages/pandas/plotting/_core.py:1716:
  series.name = label
<matplotlib.axes._subplots.AxesSubplot at 0x7f5b28813128>
```

Dataframe에서 시간과 세 개의 센서 데이터를 전부 선택해서 그래프를 그린다.





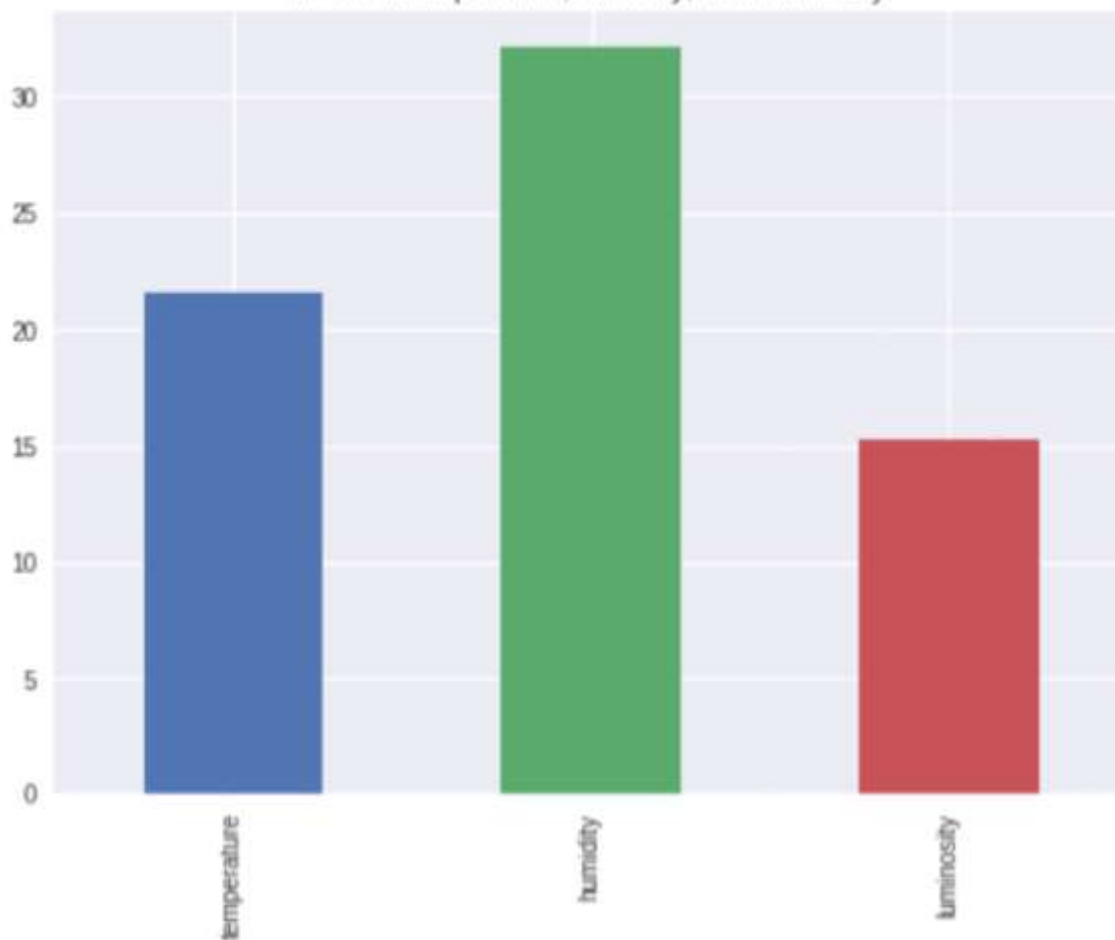
A5.9.8 IOT data mining

3.5 Plot mean of sensor data

```
1 iot_data[['temperature', 'humidity', 'luminosity']].mean().plot.bar(figsize=(8,6),  
2 title="Mean of temperature, humidity, and luminosity")
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f5b297d9470>

Mean of temperature, humidity, and luminosity



Dataframe에서 세 개의 센서 데이터의 평균을 구해서 그래프를 그린다.



A5.9.8 IOT data mining

3.6.1 Plot the change of sensor data over various time spans.

Set date as index of timestamp

```
[ ] 1 | iot_data.set_index('date', inplace=True)
```

```
[ ] 1 | iot_data.info() # timestamp index
```

```
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 307849 entries, 2018-10-23  
Data columns (total 3 columns):  
temperature    307849 non-null float64  
humidity       307849 non-null float64  
luminosity     307849 non-null int64  
dtypes: float64(2), int64(1)  
memory usage: 9.4 MB
```

```
1 | iot_data.head()
```

	temperature	humidity	luminosity
date			
2018-10-23 04:25:21.349	23.2	39.7	0
2018-10-23 04:25:31.594	23.2	39.7	0
2018-10-23 04:25:41.855	23.2	39.7	0
2018-10-23 04:25:52.100	23.2	39.7	0
2018-10-23 04:26:02.360	23.2	39.7	0

시간(date)을 **timestamp** 형태의
Index로 변경해서 데이터를
재구성한다.



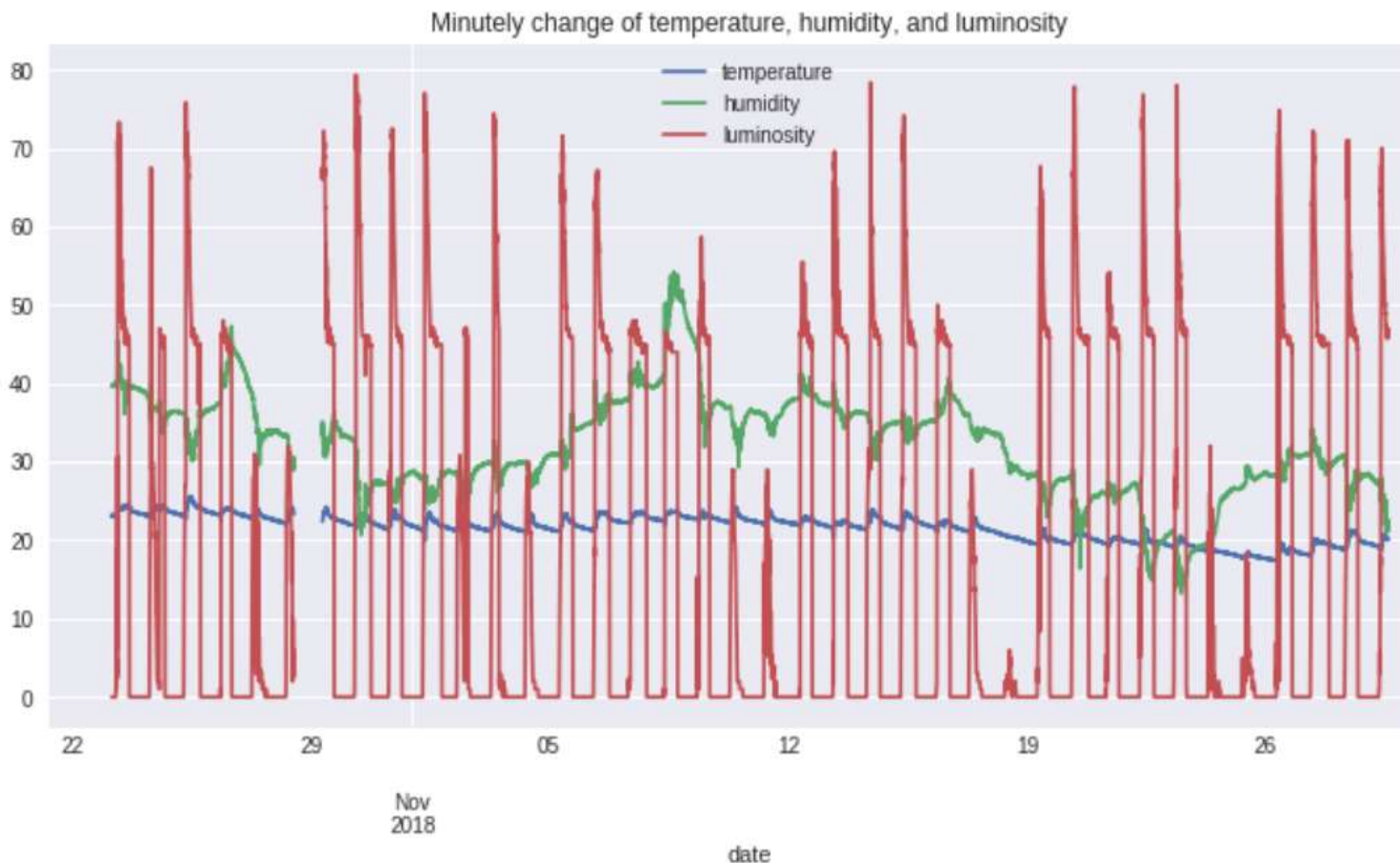
A5.9.8 IOT data mining

3.6.2 Plot the change of sensor data over various time spans.

1 분당 평균 그래프

```
1 # Plot mean of the iot data per every minute  
2 iot_data.resample('60S').mean().plot(figsize=(12,6),  
3 title='Minutely change of temperature, humidity, and lumi
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f5b2b57c630>

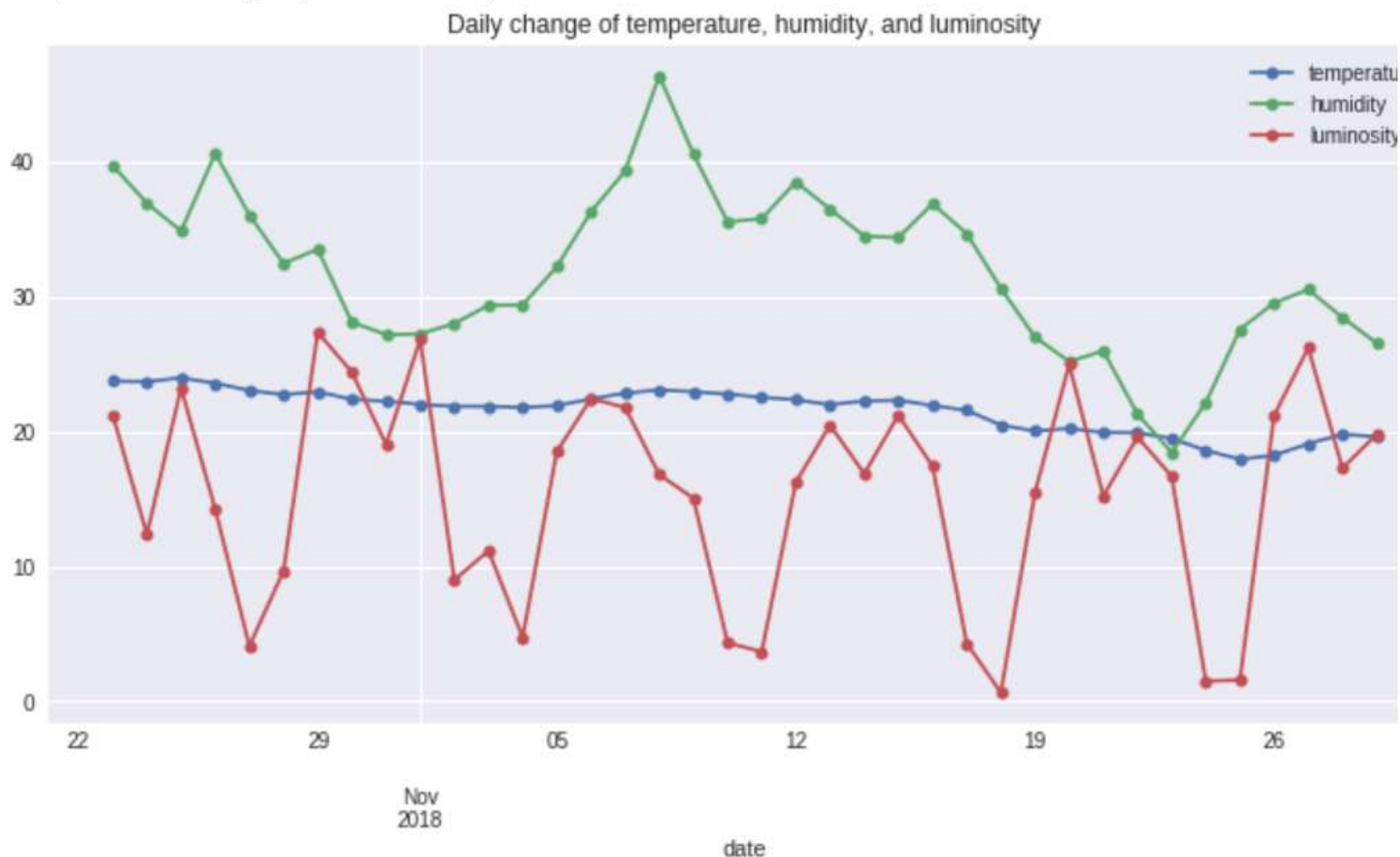


3.6.3 Plot the change of sensor data over various time spans.

1 일당 평균 그래프

```
1 # Plot mean of the iot data per every day
2 iot_data.resample('D').mean().plot(kind='line', marker='o', ms=6, figsize=(12,6),
3                                     title='Daily change of temperature, humidity, and luminosit
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f5b2c7fb7f0>





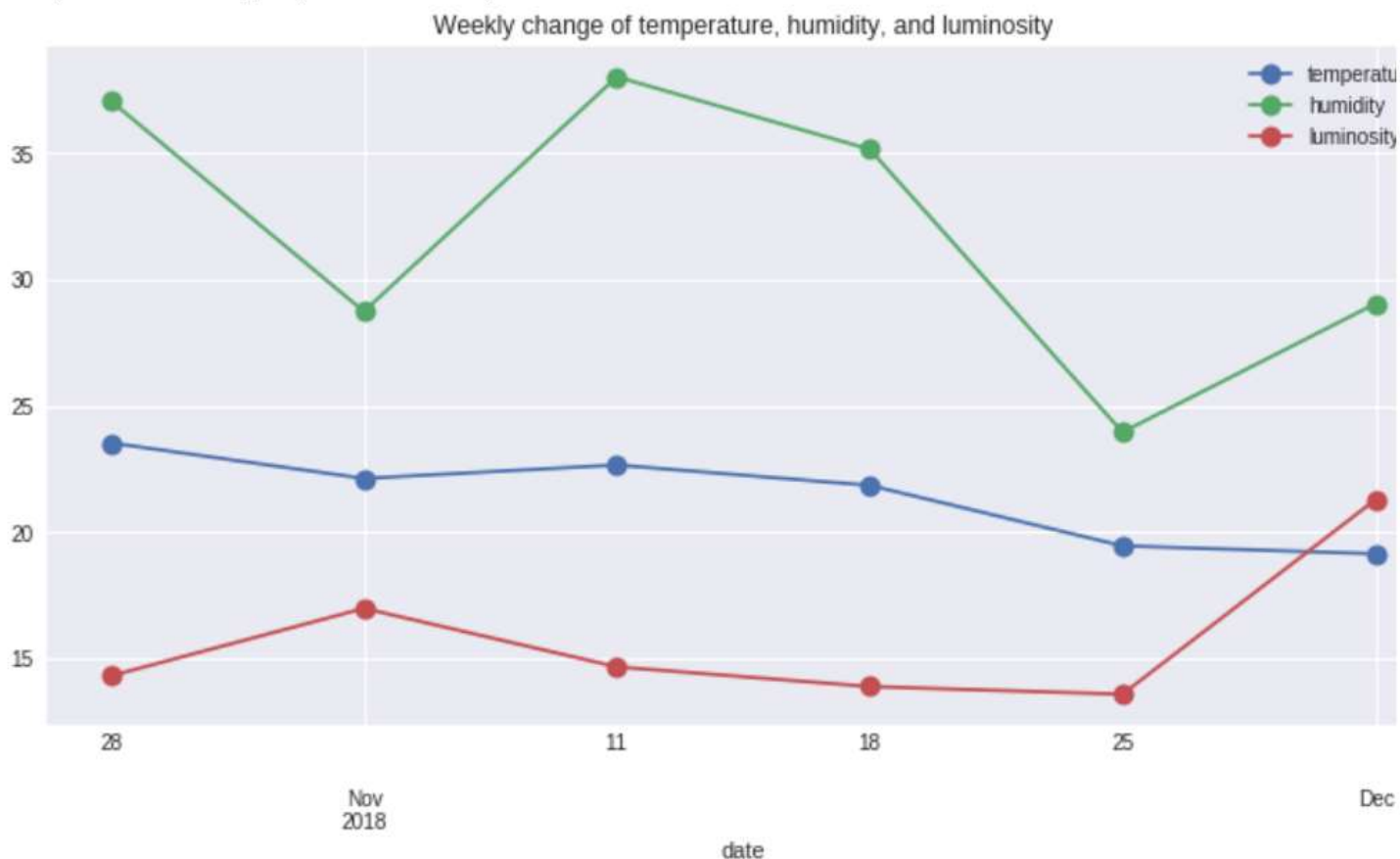
A5.9.8 IOT data mining

3.6.3 Plot the change of sensor data over various time spans.

1 주당 평균 그래프

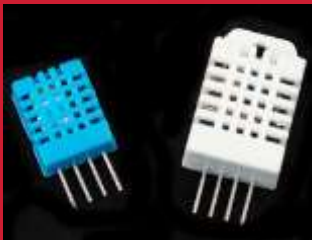
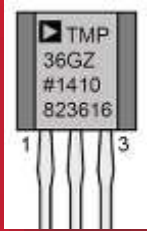
```
1 # Plot mean of the iot data per every week
2 iot_data.resample('W').mean().plot(kind='line', marker='o', ms=10,
3                                     figsize=(12,6),
4                                     title='Weekly change of temperature, humidity, and luminosi
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f5b2c8f8748>





[Practice]



◆ [wk11]

- Data Mining of IoT Data
- Multi-sensor circuits (cds-dht22)
- Complete your project
- Upload folder: aann-rpt11
- Use repo “aann” in github

wk11 : Practice : aann-rpt11

◆ [Target of this week]

- Complete your works
- Save your outcomes and upload outputs in github

제출폴더명 : **aann-rpt11**

- 제출할 파일들

- ① **iot_csv.ipynb**
- ② **iot_ison.ipynb**
- ③ **All *.js**
- ④ **public/All *.html**
- ⑤ **client_iot.html**
- ⑥ **public/data/All data (*.csv)**
- ⑦ **AAnn_s2000.csv**
- ⑧ **Don't upload node_modules subfolder**

● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <https://plot.ly/> plotly
- ✓ <https://www.mongodb.com/> MongoDB
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub

Target of this class

Real-time Weather Station from nano 33 BLE sensors



on Time: 2020-09-09 10:27:17.321

