

사물인터넷 / Internet of Things

HW #01 문제 및 보고서

이름	최재원
학번	201724605
소속 학과/대학	부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공
분반	059 (담당교수: 김태운)

<주의사항>

- 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
- PLATO 제출 데드라인: 11/8(화) ~ 11/21(월) 23:50 까지 (데드라인 지나면 0 점 처리) // 2 주간
진행하는 과제입니다.
- PLATO 에 다음 파일을 제출: **보고서 및 소스코드**(문서에 첨부, 또는 별도의 첨부파일로 제출)
 - 보고서
 - 보고서 파일명에 이름과 학번을 입력해 주세요. 그리고, 보고서 첫 페이지에 이름, 학번, 소속 정보를 작성해 주세요.
 - PDF 파일로 변환 후 제출을 권장하나, doc/docx 형식의 워드 문서 제출도 가능
 - 단, pdf/doc/docx 외의 문서 형식으로는 보고서를 제출할 수 없습니다.
 - 코드(소스코드, 스크립트, Makefile 등)를 작성하는 경우, 보고서에서 각각의 문제 하단에 코드를 붙여 넣기 하거나 또는 별도의 첨부파일(zip 압축파일)로 제출
 - 소스코드를 압축하고 별도로 첨부하여 제출하는 경우 문제별로 폴더를 생성하고 폴더별로 코드를 정리한 후 전체 폴더를 압축하여 제출할 것
 - 코드 작성 시, 파이썬 프로그래밍 언어를 사용할 것을 권장함(단, 부득이한 경우에는 다른 프로그래밍 언어를 사용할 수 있음)
 - 부정행위 적발 시 0 점 처리함

<개요>

이번 과제는 1) VirtualBox 를 사용해서 가상 머신을 생성하고, 2) MQTT 를 사용해서 데이터 수집 및 missing value 예측을 수행하는 내용입니다.

<실습 과제 Part 1. 가상머신 및 MQTT 환경 설정>

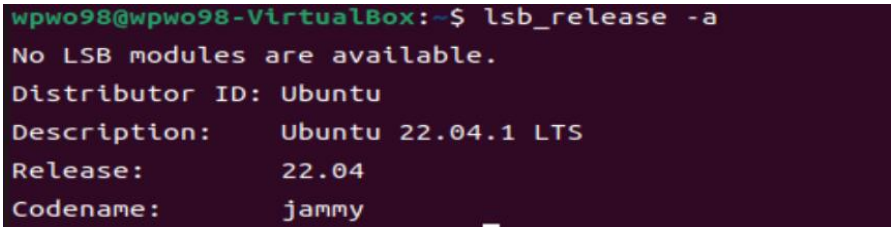
[Q 1] 가상 머신 생성 [30 점]

VirtualBox 에서 가상 머신을 생성하고 우분투 리눅스를 설치하세요. 이 때,

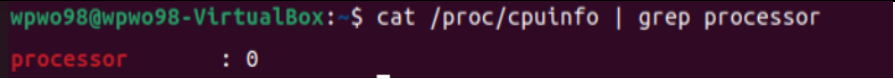
- VirtualBox 는 가급적이면 최신의 버전을 사용하세요 (<https://www.virtualbox.org/wiki/Downloads>)
- 가상머신은 총 2 대를 생성하세요
- 우분투(<https://ubuntu.com/>)는 20.04 또는 최신의 버전을 사용하세요
- 우분투는 Server 또는 Desktop 중 원하는 버전을 사용하세요.
- 우분투 서버 설치용 ISO 이미지는 공식 웹에서 직접 다운로드 받거나, 또는 Torrent 를 사용해서 다운로드 받을 수 있습니다

설치가 완료되면, 가상 머신을 시작하고 우분투에 로그인 한 후 터미널에서 아래의 명령어를 입력하세요. 명령어 실행 후, 터미널 출력 화면을 캡처해서 본 문서에 첨부(=붙여넣기)하세요. 그리고, 각 명령어의 출력 결과가 의미하는 것이 무엇인지 설명하세요.

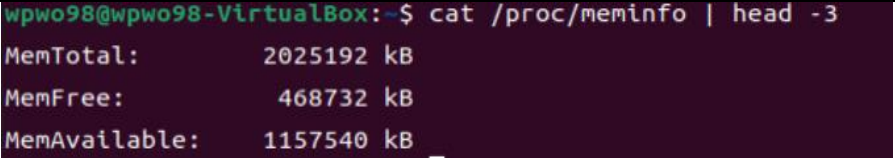
Q 1.1) `$lsb_release -a`

<p>위 명령을 터미널에서 입력하고, 출력 화면을 캡처하여 우측에 첨부하세요</p>	 <pre>wpwo98@wpwo98-VirtualBox:~\$ lsb_release -a No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 22.04.1 LTS Release: 22.04 Codename: jammy</pre>
<p>위 명령의 출력 결과가 의미하는 것이 무엇인지 설명하세요</p>	<p>The lsb_release command prints certain LSB (Linux Standard Base) and Distribution information.</p> <p>여기에 -a 를 추가하면, 모든 information 을 표시한다.</p>

Q 1.2) `$cat /proc/cpuinfo | grep processor`

위 명령을 터미널에서 입력하고, 출력 화면을 캡처하여 우측에 첨부하세요	
위 명령의 출력 결과가 의미하는 것이 무엇인지 설명하세요	Cpu info 중, processor 개수를 표시한다.

Q 1.3) \$cat /proc/meminfo | head -3

위 명령을 터미널에서 입력하고, 출력 화면을 캡처하여 우측에 첨부하세요	
위 명령의 출력 결과가 의미하는 것이 무엇인지 설명하세요	현재 메모리 사용 정보를 확인한다. Head -3 으로 info 중 상위 3 개만 표시.

[Q 2] 가상 머신 NAT 네트워크 설정 [30 점]

두 개의 우분투 가상머신을 VM 1, VM 2 라고 부르겠습니다.

ifconfig 명령어 도구 설치하기

VM 1, 2 가상머신에서 ifconfig 명령을 입력하여 IP 주소를 확인하세요. 만약 “Command ‘ifconfig’ not found, ...” 라는 메시지가 나타나면, 다음의 순서로 net-tools 를 설치한 후, ifconfig 명령을 다시 입력하세요:

- 1) `$sudo apt update` 를 입력하여 패키지 목록 최신화
- 2) `$sudo apt install -y net-tools` 를 입력하여 net-tools 패키지 설치

설치가 완료되면 VM 1, 2 에서 ifconfig 명령을 입력하여 IPv4 주소를 확인하세요. 보통은 10.0.X.X 형식의 주소가 할당되지만, 다른 대역의 주소가 할당될 수도 있습니다. 예를 들어, ifconfig 명령을 입력한 결과가 아래와 같다면, IPv4 주소는 10.10.1.151 입니다.

```
$ ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450
    inet 10.10.1.151 netmask 255.255.255.0 broadcast 10.10.1.255
    inet6 fe80::f816:3eff:fed8:1d14 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:d8:1d:14 txqueuelen 1000 (Ethernet)
    RX packets 235355 bytes 512200432 (512.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 57226 bytes 5382029 (5.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1364 bytes 109906 (109.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1364 bytes 109906 (109.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

VM 1, 2 가 서로 통신 가능한 “NAT 네트워크” 구성하기

VM 1 과 2 가 서로 다른 IPv4 주소를 가지고 있어야 하며, 서로 통신이 가능한 상태로 만들어야 합니다. 이를 위해, VM 1, 2 의 네트워크 구성을 VirtualBox 기본 값인 “NAT”가 아니라, “NAT 네트워크”로 설정하고 서로 통신이 가능하도록 VirtualBox 네트워크를 구성하세요. IP 주소는 DHCP 를 통한 동적 할당을 권장하지만, 부득이한 경우 수동으로 IP 를 구성해도 됩니다. 가상 머신의 IP 주소는 Private IP 대역을 사용하세요: 10.0.0.0 ~ 10.255.255.255 또는 172.16.0.0 ~ 172.31.255.255 또는 192.168.0.0 ~ 192.168.255.255.

설정이 완료되면 VM 1, 2 는 서로 다른 IPv4 주소를 가지고, 각각은 인터넷 연결이 가능하고, 서로 ping 명령을 통한 통신이 가능해야 합니다. 아래를 참고하여 질문에 답하거나 캡처 화면을 첨부하세요.

Q 2.1) NAT 및 NAT 네트워크

NAT(네트워크 주소 변환)의 개념 및 동작방식을 조사해서 우측에 요약하세요.	<p>[1] NAT 의 개념</p> <p>Network Address Translate 의 약자. 네트워크 주소를 변환하는 역할.</p> <p>[2] NAT 의 동작방식</p> <p>NAT 라우터로 들어온 패킷의 Ip 헤더를 체크해 NAT 테이블의 주소로 바꾸고, checksum 을 다시 계산하여 Ip 의 헤더를 바꿈.</p>
VirtualBox 에서 “NAT”와 “NAT 네트워크”의 차이는 무엇인지 답하세요.	<p>NAT 는 gateway 주소 및 기타 설정을 변경할 수 없음 (ex. Ip 고정 등).</p> <p>NAT 네트워크는 반대로 자유롭게 조작 가능함 (ex. Ip 변경 가능).</p>

Q 2.2) VM 1 에서 아래를 수행하고 터미널 출력 화면을 캡처하여 첨부하세요.

<p>VM1의 터미널에서 <code>\$ifconfig</code> 명령을 입력하고, 출력 화면을 우측에 첨부하세요 (= VM 1의 IP 주소 확인하기)</p>	<pre>wpwo98@wpwo98-VirtualBox:~\$ ifconfig enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255 inet6 fe80::1362:c6bc:c431:1314 prefixlen 64 scopeid 0x20<link> ether 08:00:27:49:58:da txqueuelen 1000 (Ethernet) RX packets 162793 bytes 246068137 (246.0 MB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 33316 bytes 2038685 (2.0 MB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 inet6 ::1 prefixlen 128 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) RX packets 187 bytes 18275 (18.2 KB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 187 bytes 18275 (18.2 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0</pre>
<p>VM 1의 IPv4 주소는 무엇인지 우측에 답하시오</p>	<p>10.0.2.15</p>
<p>VM 1의 터미널에서 <code>\$ping google.com</code>을 입력하고, 결과 화면을 우측에 첨부하세요 (= VM 1에서 인터넷 연결이 가능한지 확인하기)</p>	<pre>wpwo98@wpwo98-VirtualBox:~\$ ping google.com PING google.com (172.217.161.238) 56(84) bytes of data. 64 bytes from kix06s05-in-f14.1e100.net (172.217.161.238): icmp_seq=1 ttl=56 time=31.3 ms 64 bytes from kix06s05-in-f14.1e100.net (172.217.161.238): icmp_seq=2 ttl=56 time=31.1 ms 64 bytes from kix06s05-in-f14.1e100.net (172.217.161.238): icmp_seq=3 ttl=56 time=32.0 ms 64 bytes from kix06s05-in-f14.1e100.net (172.217.161.238): icmp_seq=4 ttl=56 time=31.8 ms 64 bytes from kix06s05-in-f14.1e100.net (172.217.161.238): icmp_seq=5 ttl=56 time=32.9 ms 64 bytes from kix06s05-in-f14.1e100.net (172.217.161.238): icmp_seq=6 ttl=56 time=32.6 ms</pre>

<p>VM 1 의 터미널에서 \$ping <VM 2 의 IPv4 주소> 를 입력하고, 결과 화면을 우측에 첨부하세요 (= VM 1 에서 VM 2 로 통신이 가능하지 확인하기)</p>	<pre>wpwo98@wpwo98-VirtualBox:~\$ ping 10.0.2.4 PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data. 64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=1.06 ms 64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=1.25 ms 64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.561 ms 64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.486 ms 64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=0.755 ms 64 bytes from 10.0.2.4: icmp_seq=6 ttl=64 time=0.491 ms</pre>
---	--

Q 2.3) VM 2 에서 아래를 수행하고 터미널 출력 화면을 캡처하여 첨부하세요.

<p>VM2 의 터미널에서 \$ifconfig 명령을 입력하고, 출력 화면을 우측에 첨부하세요 (= VM 2 의 IP 주소 확인하기)</p>	<pre>asdf@asdf-VirtualBox:~\$ ifconfig enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255 inet6 fe80::584e:dcd2:935c:ae33 prefixlen 64 scopeid 0x20<link> ether 08:00:27:d1:23:16 txqueuelen 1000 (Ethernet) RX packets 1376 bytes 1595403 (1.5 MB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 650 bytes 81939 (81.9 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 inet6 ::1 prefixlen 128 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) RX packets 662 bytes 56255 (56.2 KB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 662 bytes 56255 (56.2 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0</pre>
<p>VM 2 의 IPv4 주소는 무엇인지 우측에 답하시오</p>	<p>10.0.2.4</p>

<p>VM2 의 터미널에서 \$ping google.com 을 입력하고, 결과 화면을 우측에 첨부하세요 (= VM 2 에서 인터넷 연결이 가능한지 확인하기)</p>	<pre>asdf@asdf-VirtualBox:~\$ ping google.com PING google.com (172.217.25.174) 56(84) bytes of data. 64 bytes from sin01s16-in-f14.1e100.net (172.217.25.174): icmp_seq=1 ttl=56 time=30.2 ms 64 bytes from syd09s13-in-f14.1e100.net (172.217.25.174): icmp_seq=2 ttl=56 time=27.3 ms 64 bytes from syd09s13-in-f174.1e100.net (172.217.25.174): icmp_seq=3 ttl=56 time=27.6 ms 64 bytes from syd09s13-in-f174.1e100.net (172.217.25.174): icmp_seq=4 ttl=56 time=25.9 ms 64 bytes from syd09s13-in-f174.1e100.net (172.217.25.174): icmp_seq=5 ttl=56 time=27.4 ms 64 bytes from syd09s13-in-f174.1e100.net (172.217.25.174): icmp_seq=6 ttl=56 time=27.2 ms</pre>
<p>VM2 의 터미널에서 \$ping <VM 1 의 IPv4 주소> 를 입력하고, 결과 화면을 우측에 첨부하세요 (= VM 2 에서 VM 1 으로 통신이 가능하지 확인하기)</p>	<pre>asdf@asdf-VirtualBox:~\$ ping 10.0.2.15 PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data. 64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.657 ms 64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=1.49 ms 64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=1.58 ms 64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=1.20 ms 64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=1.37 ms 64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=1.27 ms 64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=0.562 ms</pre>

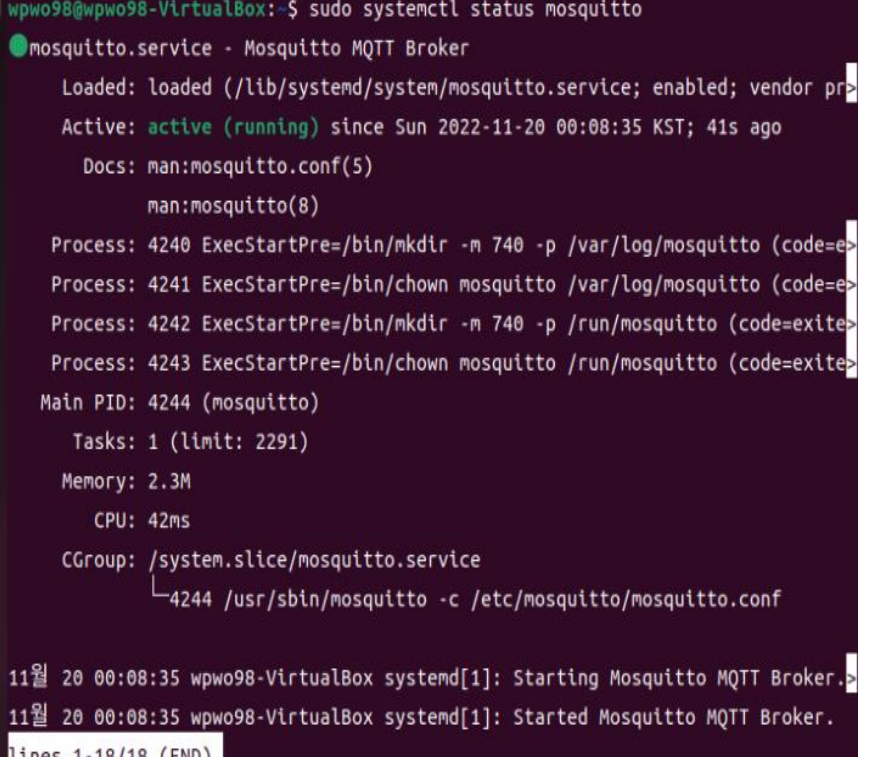
[Q 3] “Mosquitto” MQTT 설치하고 동작 확인하기 [40 점]

가상 머신에 “Mosquitto” MQTT 브로커 및 클라이언트를 설치하고 CLI 기반의 pub/sub 통신을 수행해 보는 과제입니다. 인터넷 검색 또는 아래의 URL 을 참고하여 “Mosquitto” MQTT 를 설치하세요. VM 1 에 MQTT 서버 및 클라이언트를 설치하고, VM 2 에 MQTT 클라이언트를 설치하세요. VM 1 은 MQTT Broker 및 Publisher 의 역할을, VM 2 는 MQTT Subscriber 역할을 수행하도록 환경을 구성하세요.

- “Mosquitto” MQTT 설치 매뉴얼: <https://www.vultr.com/docs/install-mosquitto-mqtt-broker-on-ubuntu-20-04-server/>

아래를 참고하여 질문에 답하거나 캡처 화면을 첨부하세요.

Q 3.1) VM 1 에서 MQTT Broker 동작 상태 확인하기

<p>VM 1 에서 <code>\$sudo systemctl status mosquitto</code> 명령을 입력하고, 출력 화면을 캡처하여 우측에 첨부하세요.</p>	 <pre>wpwo98@wpwo98-VirtualBox:~\$ sudo systemctl status mosquitto ● mosquitto.service - Mosquitto MQTT Broker Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pr> Active: active (running) since Sun 2022-11-20 00:08:35 KST; 41s ago Docs: man:mosquitto.conf(5) man:mosquitto(8) Process: 4240 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=e> Process: 4241 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=e> Process: 4242 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exite> Process: 4243 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exite> Main PID: 4244 (mosquitto) Tasks: 1 (limit: 2291) Memory: 2.3M CPU: 42ms CGroup: /system.slice/mosquitto.service └─4244 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf 11월 20 00:08:35 wpwo98-VirtualBox systemd[1]: Starting Mosquitto MQTT Broker.> 11월 20 00:08:35 wpwo98-VirtualBox systemd[1]: Started Mosquitto MQTT Broker. lines 1-18/18 (END)</pre>
--	---

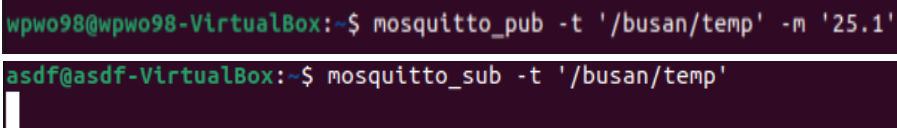
Q 3.2) VM 1 에서 메시지 Publish 하기

<p>VM 1 에서 “/busan/temp” 토픽으로 “25.1” 이라는 값을 publish 하려고 합니다. 어떠한 명령을 터미널에 입력해야 하는지를 우측에 답하세요.</p>	<pre>mosquito_pub -t "/busan/temp" -m "25.1"</pre>
---	--

Q 3.3) VM 2 에서 메시지 Subscribe 하기

VM 2 에서 “/busan/temp” 토픽을 Subscribe 하려고 합니다. 어떠한 명령을 터미널에 입력해야 하는지를 우측에 답하세요.	<code>mosquitto_sub -t “/busan/temp”</code>
---	---

Q 3.4) Pub/Sub 메시지 전달 테스트

VM 2 에서 “/busan/temp” 토픽을 Subscribe 하고 있는 상황에서, VM 1 에서 “/busan/temp” 토픽으로 “25.1” 이라는 값을 Publish 하세요. VM 2 의 터미널 출력을 캡처하여 우측에 첨부하세요.	 <pre>wpwo98@wpwo98-VirtualBox:~\$ mosquitto_pub -t '/busan/temp' -m '25.1' asdf@asdf-VirtualBox:~\$ mosquitto_sub -t '/busan/temp'</pre>
--	---

<실습 과제 Part 2. 온도 센싱 값 수신 및 Missing Value 예측>

[Q 4] 부산시 온도 데이터 plot 하기 [20 점]

첨부한 Busan_2021_Temperature_Record.csv 파일은 2021 년 한 해 동안 부산의 온도 변화를 기록한 CSV 파일 입니다. 이번 과제에서는 CSV 파일의 내용 중 “일시” 및 “평균기온” 값을 사용합니다. 로컬 Python 환경에서 코딩하거나 구글 Colab 환경을 사용하세요. CSV 파일에서 메타 정보를 담고 있는 1~11 행을 삭제하고 과제를 진행해도 됩니다.

아래를 참고하여 질문에 답하거나 캡처 화면을 첨부하세요.

Q 4) Pandas 및 Matplotlib 패키지 사용하여 데이터 도식화하기

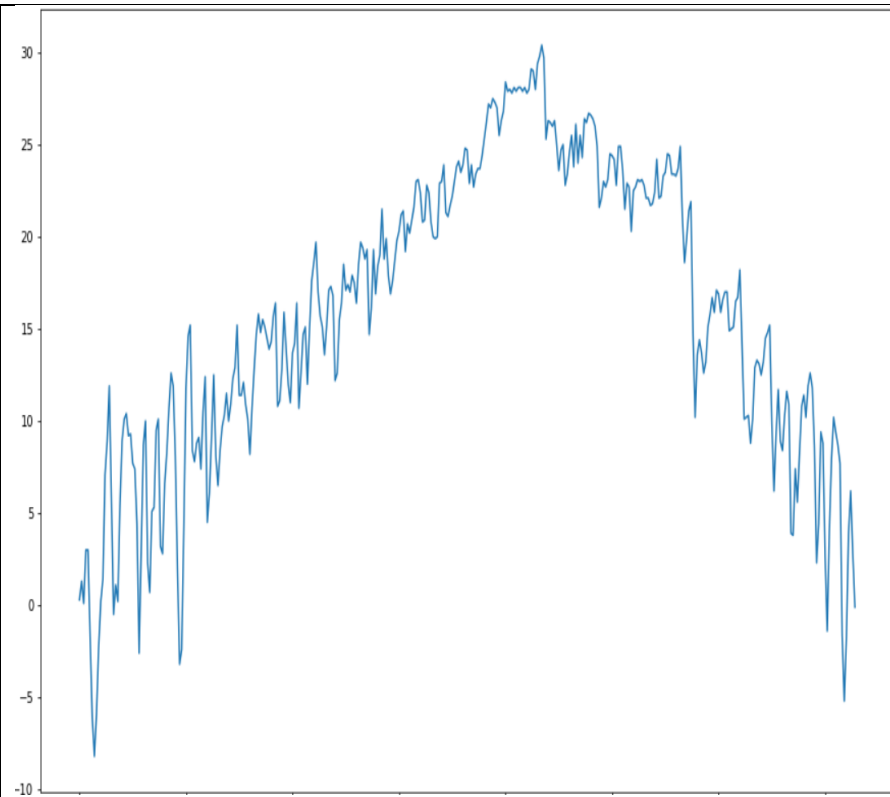
Pandas 및 Matplotlib 를 사용하여 “일시”별로 “평균기온” 을 그래프로 그리는 코드를 작성하고, 코드를 캡처하여 오른쪽에 첨부하세요.

```
[1] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv
```

```
[5] df = pd.read_csv("/content/sample_data/Busan_2021_Temperature_Record.csv", encoding='cp949')
df["일시"] = pd.to_datetime(df["일시"])
#df.head(5)
```

```
[18] #df['일시'] = pd.to_datetime(df['일시'])
plt.figure(figsize=(25,15))
plt.plot(df['평균기온(℃)'])
plt.xlabel('일자')
plt.ylabel('평균 기온')
plt.show()
```

“일시”별 “평균기온” 그래프를
오른쪽에 첨부하세요



[Q 5] MQTT Pub/Sub 으로 부산시 온도 데이터 송수신 하기 [40 점]

VM 1 은 Broker 및 Publisher 를 실행하고, VM 2 는 Subscriber 를 실행합니다. 본 문제 해결을 위해 CLI 및 셸 프로그래밍을 사용해도 되고, 파이썬 코딩을 이용해도 됩니다. 부산시 온도 데이터 CSV 파일 원본 그대로 사용해도 되고, “일시” 및 “평균기온” 데이터만 남기고 나머지 데이터는 모두 삭제한 CSV 파일을 사용해도 됩니다.

아래를 참고하여 질문에 답하거나 캡처 화면을 첨부하세요.

Q 5.1) Publisher 및 확률적 전송

Publisher 가 “/busan/temp” 토픽을 사용하여 “일시,평균기온” 정보를 빠짐없이 1 초 간격으로 Publish 하는 코드를 작성하세요. 그리고, 코드를 캡처하여 오른쪽에 첨부하세요.	ping 은 정상적으로 오는데, mosquito_pub 및 sub 시 아무런 메시지도 받아지지 않아서 5 번 문제는 해결하지 못했습니다.
Publisher 가 매 초마다 95% 확률로 “일시,평균기온” 정보를 Publish 하고, 나머지 5%의 확률로는 Publisher 하지 않고 데이터를 버리도록 코드를 수정하세요. 코드를 캡처하여 오른쪽에 첨부하세요. 참고로, Q6 문제에서는 5%의 확률로 버려지는 값을 missing value 라고 부릅니다.	

Q 5.2) Subscriber

Subscriber 가 “/busan/temp” 토픽을 사용하여 “일시,평균기온” 정보를 수신하여 새로운 CSV	
--	--

파일에 저장하는 코드를 작성하세요. 그리고, 코드를 캡처하여 오른쪽에 첨부하세요.	
Subscriber 가 수신한 데이터를 사용해서 생성한 CSV 파일의 일부를 캡처하여 오른쪽에 첨부하세요.	

[Q 6] Missing Value 예측하기 [40 점]

Publisher 의 확률적 전송 및 확률적 데이터 미전송에 따라 Subscriber 는 일부 데이터가 누락된 채로 “일시,평균기온” 데이터를 수신하게 됩니다. 누락된 값을 missing value 라고 부르겠습니다. 단순 선형 회귀, 선형 기저 함수 모델, LSTM 중 하나를 사용하여 missing value 를 예측하는 코드를 작성하는 문제입니다.

아래를 참고하여 질문에 답하거나 캡처 화면을 첨부하세요.

Q 6) 예측 모델 구현하기

Subscriber 가 수신한 데이터 및 예측한 데이터를 사용하여 “일시”별 “평균기온”을 그래프로 그리는 코드를 작성하세요. 이때, 예측한 “평균기온” 값은 수신한 데이터와 구분이 되도록 그래프에 표시하세요. 프로그램을 실행하여 얻은 그래프를 오른쪽에 첨부하세요.	
본인이 개발한 예측 모델을 사용할 경우, Missing Value 에 대한 MSE(Mean Square Error)가 얼마인지를 계산하고 오른쪽에 답하세요.	

끝! 수고하셨습니다 ☺