



Summary & Pilot Study

Midterm Exam

Hwang Jihyun
02221081



Understanding Big Data

(Week 02)



1. About Compiler

Install Python3 program

```
(base) ari@ari-com:~$ which python  
/home/ari/anaconda3/bin/python
```

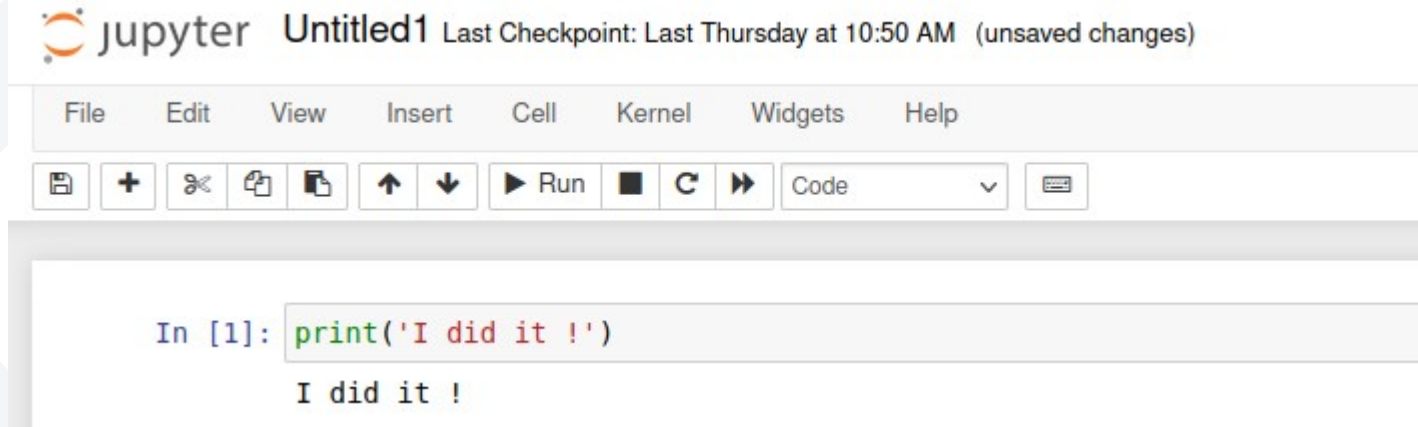
Run Python

```
(base) ari@ari-com:~$ python  
Python 3.9.13 (main, Aug 25 2022, 23:26:10)  
[GCC 11.2.0] :: Anaconda, Inc. on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> a=1  
>>> b=5  
>>> c = a + b  
>>> c  
6  
>>> 
```

Install Jupyter Lab

```
(base) ari@ari-com:~$ which jupyter  
/home/ari/anaconda3/bin/jupyter
```

Run Jupyter Notebook





Make my First Python code

(Week 03)

- 
- 
- 1. ABC of Python**
 - 2. How to crawl from COVID-19 sources**
 - 3. Database management system**

Grammars

1. input() & print()

```
In [1]: # input
imVariable = input()
```

How are u?

```
In [2]: # input2
imVariable2 = input('Typr Here : ')
```

Typr Here : How are you .. ?

```
In [3]: # print
print(imVariable2)
```

How are you .. ?

Grammars

2. if ~ else

```
In [10]: # if - elif - else
imVariable7 = input('Type here: ')
answer4 = 'Fine thank you, and you?'
if imVariable7 == 'How are you?':
    print(answer4)
elif imVariable7 == 'how are you?':
    print(answer4)
else:
    print('Wrong question, sorry')
```

Type here: how do you do?
Wrong question, sorry

Grammars

3. open() & for ~ in ~

```
In [12]: # open
         handle = open('readMe.txt', 'r')
         handle.close()
```

```
In [13]: # for~ in~
         handle2 = open('readMe.txt', 'r')
         for line in handle2 :
             print(line)
         handle2.close()
```

This is a file for reading practice in big data class.

The sentence you are looking for is..

Sample Script

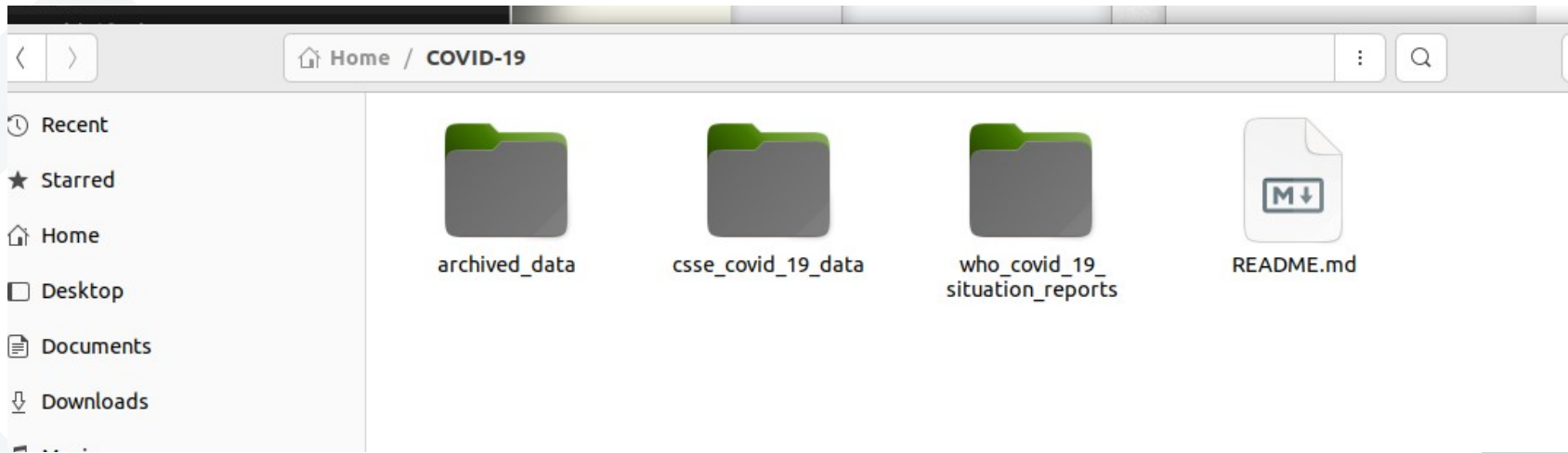
```
In [16]: # Sample script
a = 'Hello ari !'
handle3 = open('readMe.txt', 'r')
for line in handle3 :
    line = line.strip()
    if line == a :
        print('line : ' + line + ' matched to a : ' + a)
handle3.close()
```

```
line : Hello ari ! matched to a : Hello ari !
```

Install Git

```
(base) ari@ari-com:~$ which git  
/usr/bin/git
```

Git clone COVID-19 Vaccination data



Install MySQL & Make Database

```
(base) ari@ari-com:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.30-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| ari_DB   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
```



Supercomputer & Metaverse

(Week 04)

1. Pandas DataFrame

Create DataFrame

```
In [3]: # DataFrame - Using List
import pandas as pd
frame = pd.DataFrame([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
frame
```

```
Out[3]:
```

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

```
In [4]: # DataFrame - Using Dictionary
import pandas as pd
data = {
    'age' : [29, 33, 39],
    'height' : [169, 170, 183],
    'weight' : [65, 74, 81]
}
indexName = pd.Series(['ari', 'insung', 'thanin'])
frame2 = pd.DataFrame(data, index = indexName)
frame2
```

```
Out[4]:
```

	age	height	weight
ari	29	169	65
insung	33	170	74
thanin	39	183	81

Search from DataFrame

```
In [7]: # Search from DataFrame
# row
print(frame2.loc['ari'])
print(frame2.iloc[2])
# column
print(frame2['age'])
print(frame2.age)
```

```
age      29
height   169
weight    65
Name: ari, dtype: int64
age      39
height   183
weight    81
Name: thanin, dtype: int64
ari      29
insung   33
thanin   39
Name: age, dtype: int64
ari      29
insung   33
thanin   39
Name: age, dtype: int64
```

Add a new column & row

```
In [10]: # add a new column
frame2_col_added = pd.DataFrame(frame2, columns = ['age', 'height', 'weight', 'now_col'])
frame2_col_added
```

```
Out[10]:
```

	age	height	weight	now_col
ari	29	169	65	NaN
insung	33	170	74	NaN
thanin	39	183	81	NaN

```
In [13]: # add a new row
frame2_row_added = frame2_col_added.copy()
frame2_row_added.loc['inguk'] = [37, 180, 68, 'new1']
frame2_row_added
```

```
Out[13]:
```

	age	height	weight	now_col
ari	29	169	65	NaN
insung	33	170	74	NaN
thanin	39	183	81	NaN
inguk	37	180	68	new1

Save DataFrame as csv format & Read csv as DataFrame

```
In [17]: # save DataFrame as csv format
frame2_row_added.to_csv('df_to_csv.csv')
```

```
In [18]: # read csv as DataFrame
df_from_csv = pd.read_csv('df_to_csv.csv')
df_from_csv
```

```
Out[18]:
```

	Unnamed: 0	age	height	weight	now_col
0	ari	29	169	65	NaN
1	insung	33	170	74	NaN
2	thanin	39	183	81	NaN
3	inguk	37	180	68	new1

Set index name

```
In [20]: # set index name  
df_from_csv.set_index('Unnamed: 0', inplace = True)  
df_from_csv
```

```
Out[20]:
```

	age	height	weight	now_col
Unnamed: 0				
ari	29	169	65	NaN
insung	33	170	74	NaN
thanin	39	183	81	NaN
inguk	37	180	68	new1



Stock Market data & MySQL

(Week 05)

- 1. About Stock Market data**
- 2. Connecting MySQL @ Python**
- 3. Make my own table in MySQL**

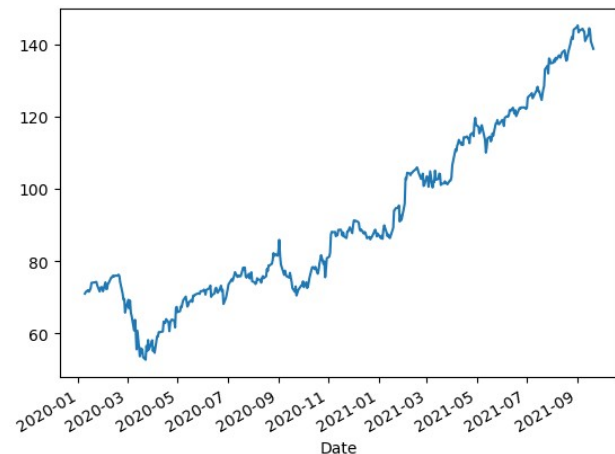
using Yahoo Finance @ python

```
import pandas_datareader as data
import datetime
start_date = datetime.datetime(2020,1,10)
end_date = datetime.datetime(2021,9,20)
google_data = data.DataReader('GOOGL','yahoo', start_date, end_date)
print(google_data.head(9))
```

Date	High	Low	Open	Close	Volume	Adj Close
2020-01-09	71.433998	70.510498	71.096497	70.989502	33200000	70.989502
2020-01-10	71.747002	70.980003	71.473503	71.447998	26258000	71.447998
2020-01-13	72.073997	71.268501	71.762497	72.001503	30730000	72.001503
2020-01-14	72.131500	71.388496	72.000000	71.529503	26076000	71.529503
2020-01-15	72.039001	71.583000	71.651001	71.959999	21550000	71.959999
2020-01-16	72.535004	72.000000	72.272499	72.508003	26080000	72.508003
2020-01-17	74.027496	72.827499	73.126999	73.975998	52424000	73.975998
2020-01-21	74.494003	73.510498	73.949997	74.112503	48930000	74.112503
2020-01-22	75.028999	74.133003	74.486504	74.193497	28458000	74.193497

```
In [6]: google_data['Close'].plot()
```

```
Out[6]: <AxesSubplot:xlabel='Date'>
```



```
In [31]: google_data.to_csv("yahoofinance.csv", index = False)
```

Download the KOSPI List

```
In [18]: # downlooad kospi list
import pandas as pd
kospicode = pd.read_html('https://kind.krx.co.kr/corpgeneral/corpList.do?method=download&marketType=stockMkt', header
print(kospicode.head(9))
```

	회사명	종목코드	업종	\
0	DL	210	기타 금융업	
1	DRB동일	4840	고무제품 제조업	
2	DSR	155660	1차 비철금속 제조업	
3	GS	78930	기타 금융업	
4	HDC현대산업개발	294870	건물 건설업	
5	KG케미칼	1390	기초 화학물질 제조업	
6	KPX케미칼	25000	기초 화학물질 제조업	
7	KSS해운	44450	해상 운송업	
8	KTis	58860	기타 정보 서비스업	

	주요제품	상장일	결산월	\
0	지주회사	1976-02-02	12월	
1	고무벨트(V벨트, 콘베이어벨트, 평벨트), 플라스틱제품	제조, 판매	1976-05-21	12월
2	합성섬유로프	2013-05-15	12월	
3	지주회사/부동산 임대	2004-08-05	12월	
4	외주주택, 자체공사, 일반건축, 토목 등	2018-06-12	12월	
5	콘크리트혼화제, 비료, 친환경농자재, 수처리제	1989-08-25	12월	
6	PPG, PU RESIN, 우레탄수지, 대향막박리재, 반도체 CM PAD, Polyether...	1994-12-27	12월	
7	특수화학해상운송(액화가스, LPG, 암모니아, VCM, 석유화학제품 등), 선박대여	2007-10-26	12월	
8	114전화번호안내, 고객센터	2010-12-17	12월	

	대표자명	홈페이지	지역
0	전병욱	http://www.dlholdings.co.kr	서울특별시
1	류영식	http://drbworl.com	부산광역시
2	홍석빈	http://www.dsr.com	부산광역시
3	허태수, 홍순기 (각자 대표이사)	NaN	서울특별시
4	최익훈, 정익희, 김희연 (각자 대표이사)	http://www.hdc-dvp.com	서울특별시
5	곽정현, 김재익	http://www.kgchem.co.kr	울산광역시
6	양준영, 최재호, 이찬수 (각자대표이사)	http://www.kpxchemical.com	서울특별시
7	이승우	http://www.kssline.com	서울특별시
8	윤경근	http://www.ktis.co.kr	서울특별시

Connect to MySQL in Python

```
(base) ari@ari-com:~$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.0.2-py3-none-any.whl (43 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

```
: import pymysql
connection = pymysql.connect(host='localhost', user='root', password='[REDACTED]', db='ari_DB', charset='utf8', autocommit=True)
cursor = connection.cursor()
sql = 'CREATE TABLE ' + tableName + '(id int, country text, year int, month int, day int, confirmed int, deaths int,
cursor.execute(sql)
connection.close()
```

Create Connection & MySQL Table

```
mysql> show tables;
```

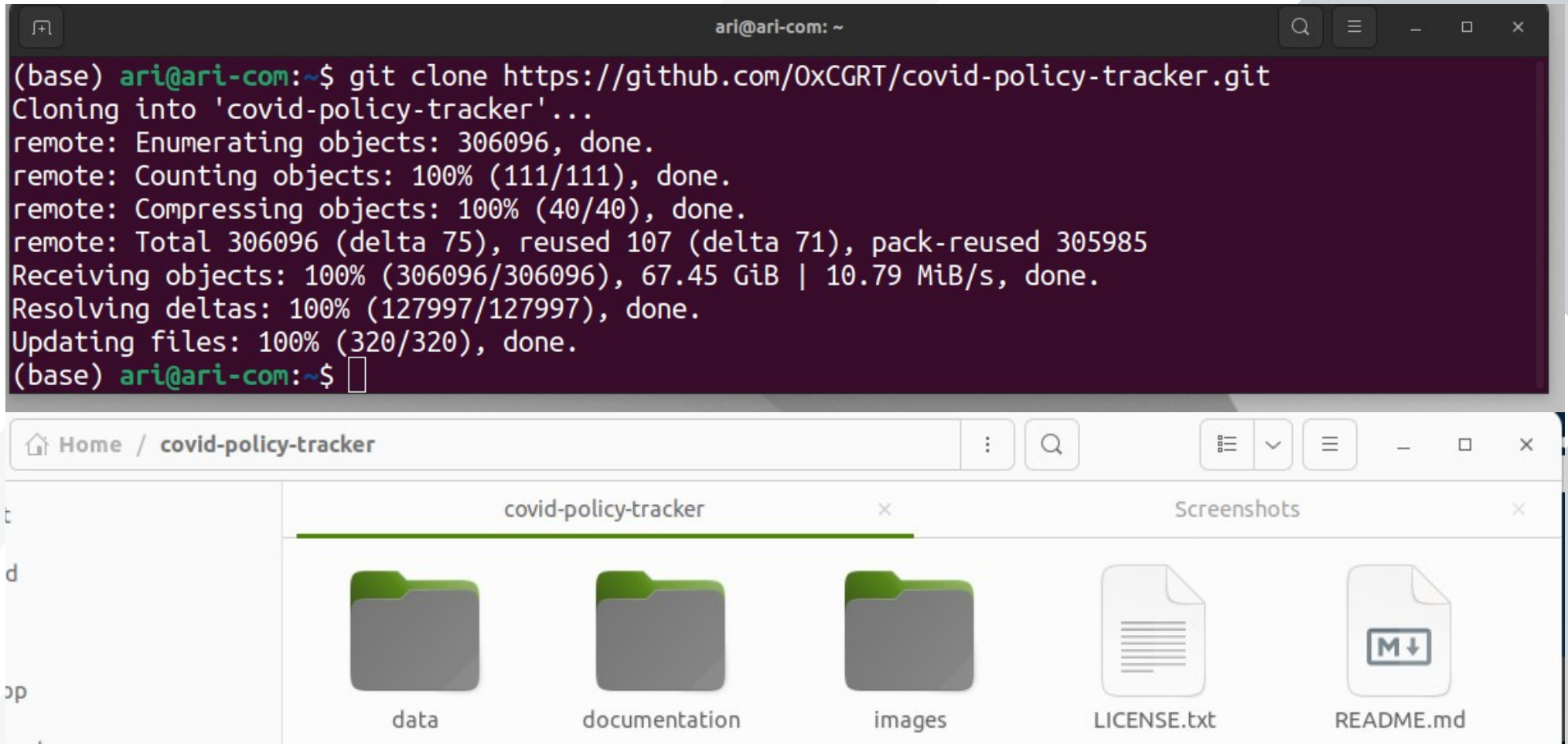
```
+-----+  
| Tables_in_ari_DB |  
+-----+  
| time_series_covid19_confirmed_global |  
| time_series_covid19_deaths_global |  
| time_series_covid19_recovered_global |  
| stock_market_data |  
+-----+
```



COVID-19 & Government response

- 1. Pandas DataFrame**
- 2. Extract target information**

Download Data sets of OxCGRT



The image shows a terminal window and a file explorer. The terminal window, titled 'ari@ari-com: ~', displays the command to clone the repository and its output. The file explorer, titled 'Home / covid-policy-tracker', shows the contents of the cloned repository, including folders for 'data', 'documentation', and 'images', and files for 'LICENSE.txt' and 'README.md'.

```
(base) ari@ari-com:~$ git clone https://github.com/OxCGRT/covid-policy-tracker.git
Cloning into 'covid-policy-tracker'...
remote: Enumerating objects: 306096, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 306096 (delta 75), reused 107 (delta 71), pack-reused 305985
Receiving objects: 100% (306096/306096), 67.45 GiB | 10.79 MiB/s, done.
Resolving deltas: 100% (127997/127997), done.
Updating files: 100% (320/320), done.
(base) ari@ari-com:~$
```

Home / covid-policy-tracker

covid-policy-tracker

data documentation images LICENSE.txt README.md

Perform a simple correlation analysis : Failed... :(

```
In [47]: import pymysql
connection = pymysql.connect(host='localhost', user='root', password='[REDACTED]', db='ari_DB', charset='utf8', autocommit=True)
cursor = connection.cursor()
sql = 'CREATE TABLE ' + 'OxCGRT_vaccines_full.csv' + '( CountryName text, CountryCode text, PRIMARY KEY (CountryName))'

cursor.execute(sql)
connection.close()
```

```
File "/tmp/ipykernel_15112/431406267.py", line 4
    sql = 'CREATE TABLE ' + 'OxCGRT_vaccines_full.csv' + '( CountryName text, CountryCode text, PRIMARY KEY (CountryName));'
    ^
SyntaxError: EOL while scanning string literal
```

```
the right syntax to use near 'CHARACTER SET UTF8' at line 2
mysql> LOAD DATA LOCAL INFILE 'OxCGRT_vaccines_full.csv'
-> INTO TABLE centers
-> CHARACTER SET UTF8
-> FIELDS TERMINATED BY ',' IGNORE 1 ROWS
-> (col1, col2, col3, ..., coln);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '..., coln)' at line 5
mysql> LOAD DATA LOCAL INFILE 'C:/Users/KD/Desktop/data.csv'
```

```
In [43]: SELECT year, month FROM time_series_covid19_confirmed_global
        INTO OUTFILE 'select_test.csv'
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n';
```

```
File "/tmp/ipykernel_15112/3119045308.py", line 1
    SELECT name, dept_cd, phone, address FROM class.select_test
    ^
SyntaxError: invalid syntax
```

A rough plan of the pilot study

- 1. Study how to learn more about tasks that you couldn't complete before and use them in a variety of ways**
- 2. Analysis of changes in the global exchange rate after COVID-19 and future prospects**
- 3. Comparison of COVID-19 and Flu Trends**

Thank You

