

|  |
| --- |
| COVID-19 데이터 정리 |
| A1조 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제출일 | 2022. 09. 29 |  |  |  |
|  | **이해동** |  |  | **신정훈** |
|  | **김승현** |  |  | **조성운** |

목차

[1. 팀원별 코드](#_Toc59713393)

[1.1 이해동,김승현](#_Toc59713394)

[1.2신정훈,조성운](#_Toc59713395)

[2. 문제및코드분석](#_Toc59713396)

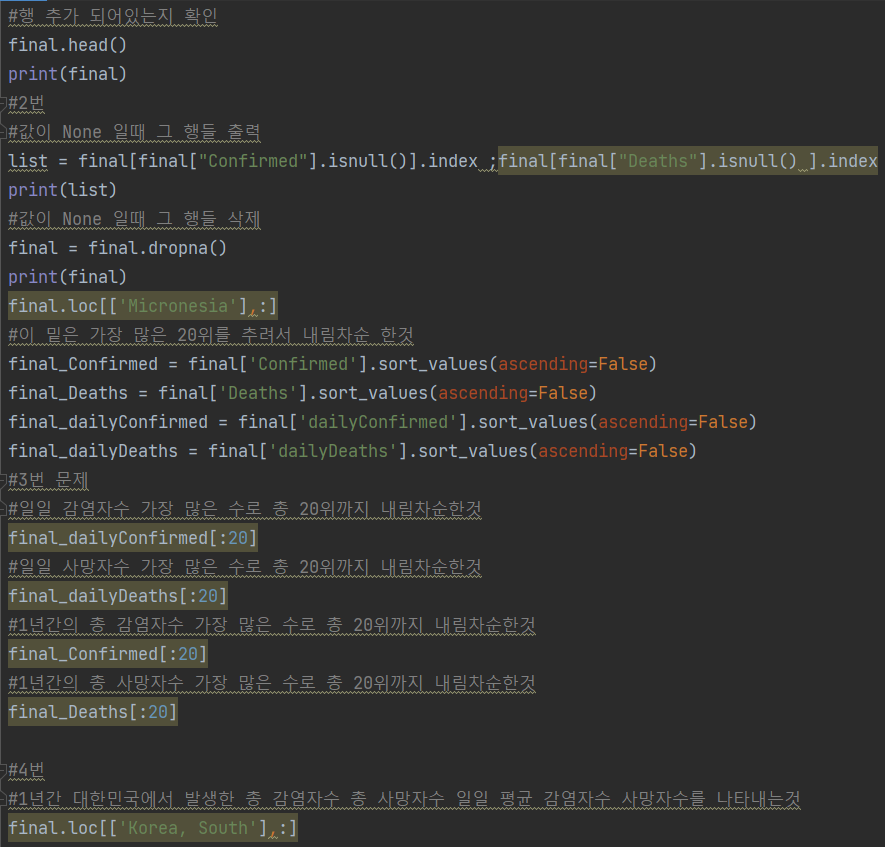
[2.1 코드분석 3](#_Toc59713397)

[2.1.1. 결과데이터 3](#_Toc59713398)

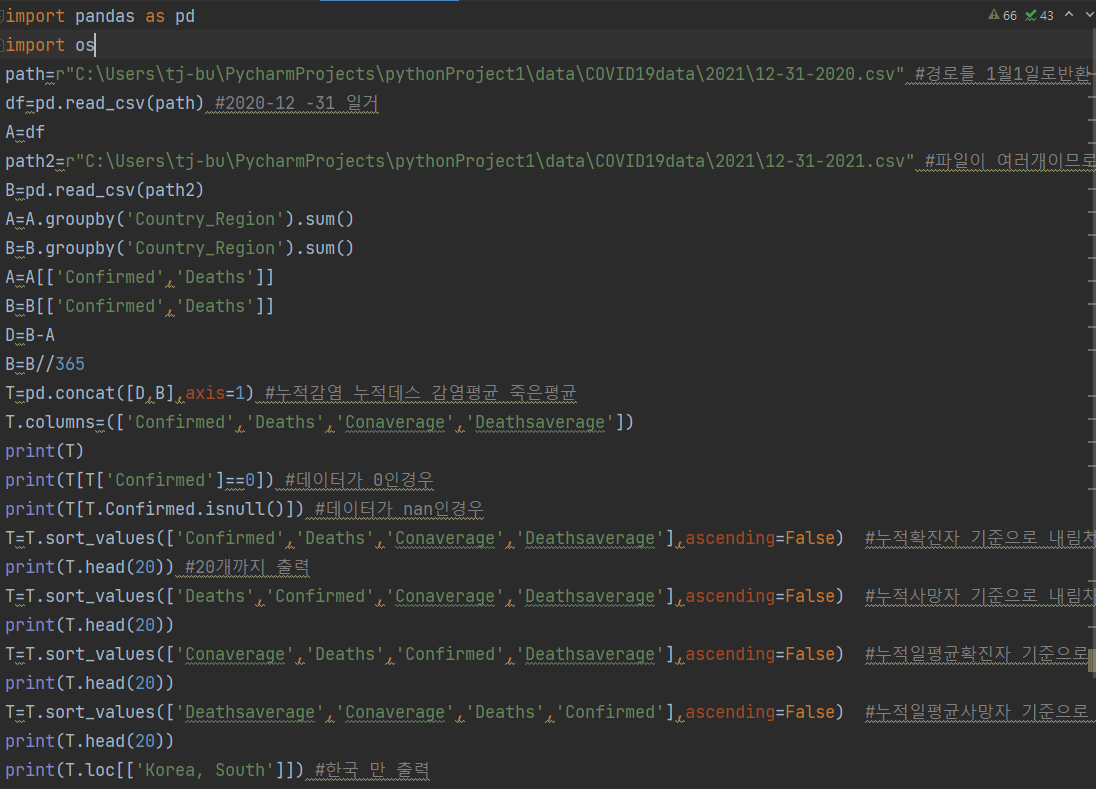
# 1.팀원 별 코드 (사진)

## 이해동

## 



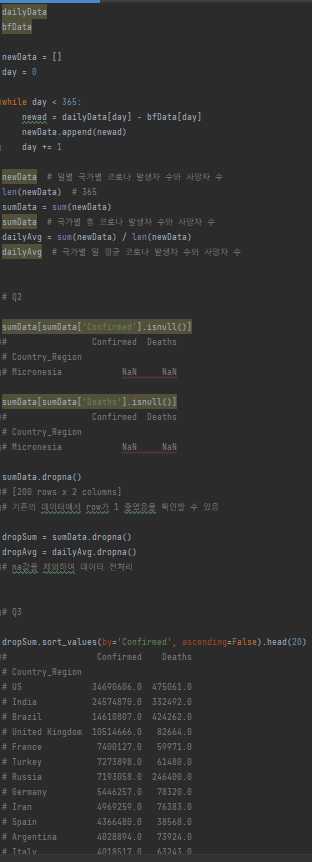
## 김승현

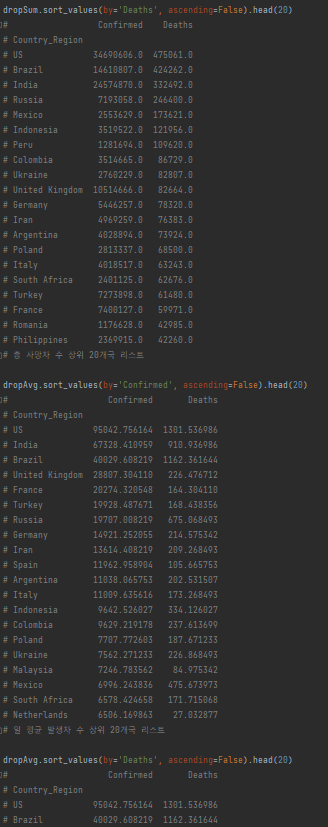


## 신정훈

### 

## 조성운

# 문제 및 코드 분석

## 문제

John’s Hopkis 대학 내 The Center For Sysrems Science and Engineering(CSSE)에서 업데이트 하는

전세계 코로나 발생 현황 데이터 셋을 가지고

1번

일별 국가별 코로나 발생자수와 사망자 수를 기준으로 전처리 하고

일부 국가는 지역별로 나뉘어 있으니 국가별로 집계

총 발생자수,총 사망자수,일평균 발생자수,일평균 사망자수 리스트를 제시

2번

데이터가 0인 경우와 데이터가 없는(None) 경우를 구분 하여 데이터가 없는

국가는 제외하고 그 제외된 국가를 리스트를 제시

3번

1번과 같지만 발생자수,사망자수,일평균 발생자수,사망자수를 기준으로 가장 많은

국가 20개를 내림차순으로 정렬하라

4번

1년간 대한민국에서 발생한 총 코로나 발생자수 총 사망자수 일평균 발생자수 사망자수를

리포트 하라

## Python 코드

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

***#1번 문제***

*#파일 불러오기*

ex1 = pd.read\_csv(r'C:\Users\tj-bu\PycharmProjects\pythonProject1\data\COVID19data\2020\12-31-2020.csv')

*#12-31-2020의 나라와 확진자수와 사망자수 뽑아오기*

ex1db = ex1[['Country\_Region','Confirmed','Deaths']]

*#GroupBy 해서 중복되는 나라 Sum해서 통합식*

ex12 = ex1db.groupby('Country\_Region').sum()

*#파일 불러오기*

ex2 = pd.read\_csv(r'C:\Users\tj-bu\PycharmProjects\pythonProject1\data\COVID19data\2021\12-31-2021.csv')

*#12-31-2021의 나라와 확진자수와 사망자수 뽑아오기*

ex2db = ex2[['Country\_Region','Confirmed','Deaths']]

*#GroupBy 해서 중복되는 나라 Sum해서 통합식*

ex22 = ex2db.groupby('Country\_Region').sum()

*#1년간 발생한 확진자수와 사망자수 계산식*

final = ex22 - ex12

#1번

print(final)

*#일일 평균 확진자 수 계산식*

a1 = final['Confirmed'] // 365

*#일일 평균 사망자 수 계산식*

a2 = final['Deaths'] // 365

*#일일 평균 확진자수 행렬 추가*

final['dailyConfirmed'] = a1

*#일일 평균 사망자수 행렬 추가*

final['dailyDeaths'] = a2

*#행 추가 되어있는지 확인*

final.head()

print(final)

***#2번***

*#값이 None 일때 그 행들 출력*

list = final[final["Confirmed"].isnull()].index ;final[final["Deaths"].isnull() ].index

#제외된 리스트 출력

print(list)

*#값이 None 일때 그 행들 삭제*

final = final.dropna()

#삭제된 이후 확인 용도

print(final)

*#이 밑은 가장 많은 20위를 추려서 내림차순 한것*

final\_Confirmed = final['Confirmed'].sort\_values(ascending=**False**)

final\_Deaths = final['Deaths'].sort\_values(ascending=**False**)

final\_dailyConfirmed = final['dailyConfirmed'].sort\_values(ascending=**False**)

final\_dailyDeaths = final['dailyDeaths'].sort\_values(ascending=**False**)

***#3번 문제***

*#일일 감염자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_dailyConfirmed[:20]

*#일일 사망자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_dailyDeaths[:20]

*#1년간의 총 감염자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_Confirmed[:20]

*#1년간의 총 사망자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_Deaths[:20]

***#4번***

*#1년간 대한민국에서 발생한 총 감염자수 총 사망자수 일일 평균 감염자수 사망자수를 나타내는것*

final.loc[['Korea, South'],:]

### 코드 분석

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

*#1번 문제*

*#파일 불러오기*

ex1 = pd.read\_csv(r'C:\Users\tj-bu\PycharmProjects\pythonProject1\data\COVID19data\2020\12-31-2020.csv')

**pd.read\_csv(경로\사용할 파일명)를 사용하면 해당 csv의 내용을 해당 변수에 넣게 됩니다**

*#12-31-2020의 나라와 확진자수와 사망자수 뽑아오기*

ex1db = ex1[['Country\_Region','Confirmed','Deaths']]

**ex1 에 있는 컬럼들 중 Country Region,Confirmed,Deaths라는**

**컬럼들만 뽑아 쓰기 위해 사용되었습니다**

*#GroupBy 해서 중복되는 나라 Sum해서 통합식*

ex12 = ex1db.groupby('Country\_Region').sum()

**Grouby를 하게되면 중복되는 Country Region컬럼의 값이**

**하나로 합쳐지게 되는데 이 합쳐지는 과정에서 값들을 전부**

**더하기 위해 뒤에 sum()함수를 사용하게 되었습니다**

*#파일 불러오기*

ex2 = pd.read\_csv(r'C:\Users\tj-bu\PycharmProjects\pythonProject1\data\COVID19data\2021\12-31-2021.csv')

*#12-31-2021의 나라와 확진자수와 사망자수 뽑아오기*

ex2db = ex2[['Country\_Region','Confirmed','Deaths']]

*#GroupBy 해서 중복되는 나라 Sum해서 통합식*

ex22 = ex2db.groupby('Country\_Region').sum()

**1년치를 계산하기 위한 2021년도 마지막 날의 파일을 불러오는 차이 말고는 위에 것과 전부 동일합니다**

*#1년간 발생한 확진자수와 사망자수 계산식*

final = ex22 - ex12

**파일의 값이 누적이기 때문에 1년치를 계산하기 위해서는**

**값이 클 수밖에 없는 2021-12-31일 파일을**

**2020-12-31파일과 뺄셈을 하여 1년간 발생한**

**총 확진자,사망자수를 계산하였습니다**

#1번

print(final)

**1년간 발생한 총 확진자,사망자수를 나타내기 위해 사용했습니다**

*#일일 평균 확진자 수 계산식*

a1 = final['Confirmed'] // 365

**final에 있는 Confiremd 컬럼의 값을 365로 나누어**

**일일 평균 발생자 수를 계산 하였습니다**

*#일일 평균 사망자 수 계산식*

a2 = final['Deaths'] // 365

**final에 있는 Deaths 컬럼의 값을 365로 나누어**

**일일 평균 사망자 수를 계산 하였습니다**

*#일일 평균 확진자수 행렬 추가*

final['dailyConfirmed'] = a1

**al의 값을 final에 새로운 행 dailyConfirmed를 생성하여**

**값을 넣기 위해 사용했습니다**

*#일일 평균 사망자수 행렬 추가*

final['dailyDeaths'] = a2

**a2의 값을 final에 새로운 행 dailyDeaths를 생성하여**

**값을 넣기 위해 사용했습니다**

*#행 추가 되어있는지 확인*

final.head()

print(final)

**final의 상위 5개의 행을 출력하기 위해 사용했습니다**

*#2번*

*#값이 None 일때 그 행들 출력*

list = final[final["Confirmed"].isnull()];final[final["Deaths"].isnull()]

#제외된 리스트 출력

print(list)

**리스트에 None 값을 가진 국가를 찾아 LIST에 넣기위해 사용했습니다**

*#값이 None 일때 그 행들 삭제*

final = final.dropna()

**final안에 값이 None인 행을 지우기위해 사용 했습니다**

#삭제된 이후 확인 용도

print(final)

*#이 밑은 가장 많은 20위를 추려서 내림차순 한것*

final\_Confirmed = final['Confirmed'].sort\_values(ascending=**False**)

final\_Deaths = final['Deaths'].sort\_values(ascending=**False**)

final\_dailyConfirmed = final['dailyConfirmed'].sort\_values(ascending=**False**)

final\_dailyDeaths = final['dailyDeaths'].sort\_values(ascending=**False**)

**final안의 컬럼 안에 있는 값을 기준으로 sort 하였습니다**

**ascending=False는 내림차순 하기 위해 사용했습니다**

*#3번 문제*

*#일일 감염자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_dailyConfirmed[:20]

*#일일 사망자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_dailyDeaths[:20]

*#1년간의 총 감염자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_Confirmed[:20]

*#1년간의 총 사망자수 가장 많은 수로 총 20위까지 내림차순한것*

final\_Deaths[:20]

**위에서부터 20개까지 각자 인덱싱하여 표현하였습니다**

*#4번*

*#1년간 대한민국에서 발생한 총 감염자수 총 사망자수 일일 평균 감염자수 사망자수를 나타내는것*

final.loc[['Korea, South'],:]

**loc는 행을 찾기 위해 사용 되었습니다**

**뒤에 : 는 전체 행을 확인 하기 위해 사용했습니다**

### 사용 함수 보충(추가)설명

Pd\_read\_csv:csv 파일을 불러오기 위해 사용

final[[“~~~”,”~~~”]]:final안에 있는 컬럼을 불러오기 위해 사용

final[“~~~”] = 변수 : 이것은 변수에 넣은 값을 final에 새로운 컬럼인 ~~~에 새로 넣기 위해 사용

final[“dailyDeaths”].sort\_values(ascending=False):dailyDeaths 칼럼의 값들을 기준으로

내림차순 하기 위해 사용되었습니다 여기서 ascending을 True로 하면 오름차순이 됩니다

Final\_Confirmed[:20]:final안의 Confirmed 칼럼에서 위에서부터 20개의 row를 출력하기 위해 사용

Final.loc[[‘Korea, South]. :]:이것은 row에서 Korea,South와 같은 행을 찾을 때 사용

뒤에 :는 전체 row를 검색하기 위해 사용

final[final["Confirmed"].isnull()].index:null값을 가지는 행을 출력하려고 사용

final.dropna():final 안에 값이 None이 한 개라도 포함된 row를 제거하기 위해 사용

ex2db.groupby('Country\_Region').sum():동일한 값을 가진것들을 단일로 합치고

그 합칠 때 값들을 전부 더하기 위해 사용

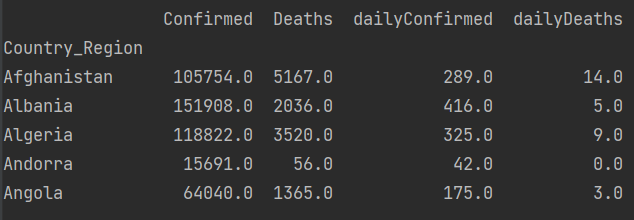
final = ex22 - ex12: 파일들이 그날 발생한 확진자 수를 내놓은게 아닌

그걸 더한 총 누적값이였기에 2021-12-31파일을 2020-12-31에 빼면

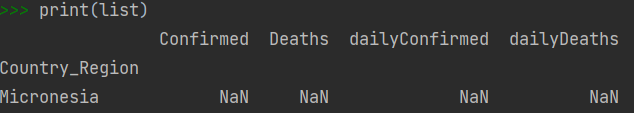
2021년간 발생한 총 확진자 수가 나오기 때문에 이렇게 산수하게 되었다

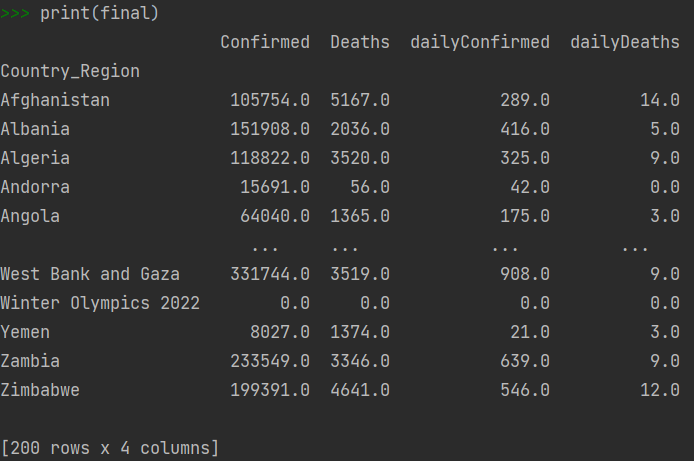
### 결과 데이터

**1)국가별 총 코로나 발생자 수 와 사망자 수**



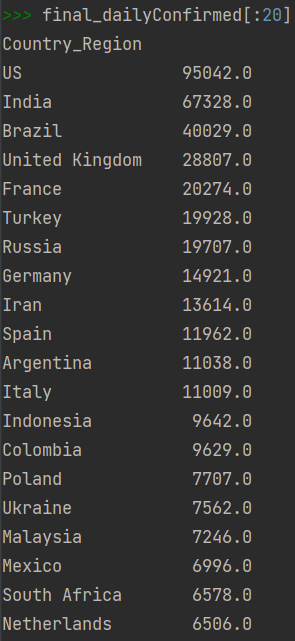
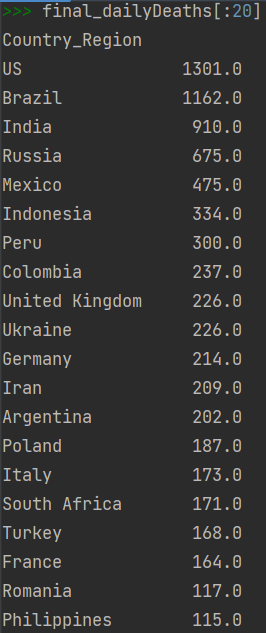
**2)0과 None값을 가지는 행 제거 및 제외된 나라 리스트화**



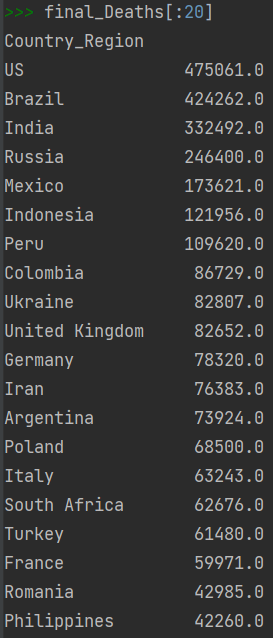
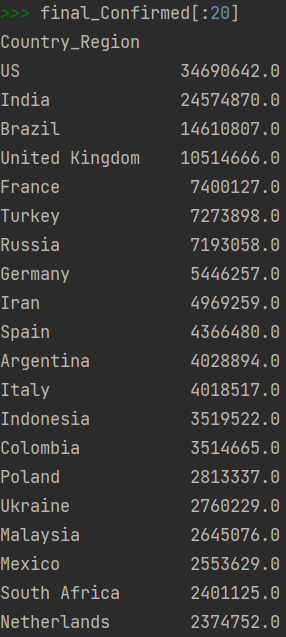


**3)각 항목별 상위 20개 출력하기**

**총 확진자 20위 총 사망자 20위**

**일 평균 확진자 20위 일평균 사망자 20위**



**4)대한민국만 따로 출력해 보기**

