**Python code에 기반한**

**2021년 코로나 데이터 분석**

2022.09.29

C1팀(권호영, 조아진, 임성구, 김현주)

**목차**

1. 분석 개요
2. 팀원 간 의견 수렴
3. 데이터 전처리 및 분석

1) 데이터 전처리 로직

2) 국가별 총확진자, 총사망자, 일평균 확진자, 일평균 사망자

3) data 이상값, 결측값 존재 국가 list

4) 상위 20개 국가 내림차순 정렬

5) 대한민국 리포트

1. 검증
2. 음수 데이터 원인
3. NaN 값 원인
4. 가설 검증
5. 결론
6. 별첨

**1. 분석 개요**

**- 제공된 데이터셋은 Johns’ Hopkins 대학 내 The Center For Systems Science and Engineering(CSSE) 에서 업데이트하는 전 세계 코로나 (COVID-19) 발생 현황 데이터셋이다. 제공된 데이터셋 중 2021년 일간 데이터에서 python code로 다음을 수행하고 결과를 팀별로 리포트하시오.**

**(1) 일별 국가별 코로나 발생자 수와 사망자 수를 기준으로 전처리하시오. 일부 국가는 지역별로 코로나 발생자 수와 사망자 수가 분리되어 있으니 국가별로 집계하고 국가, 총발생자 수, 총사망자 수, 일평균 발생자 수, 일평균 사망자 수 리스트를 제시하시오.**

-> 먼저 지역별로 집계된 국가의 경우 국가명을 기준으로 합산하는 과정이 필요하다. 제공된 데이터는 해당하는 날짜까지 누적된 코로나 발생자와 사망자 수이기 때문에 일별 발생자와 사망자 수를 구하기 위해 해당 날짜와 해당 날짜의 전날을 뺄셈하는 전처리 과정이 필요하다. 이 과정으로 해당 날짜에 늘어난 발생자, 사망자 수를 구할 수 있다. 최종적으로 일별마다 국가명을 기준으로 총발생자 수, 총사망자 수를 가진 데이터 프레임을 구할 수 있다. 이 데이터 프레임을 활용해 일평균 발생자 수와 사망자 수까지 구한다.

**(2) 데이터가 0인 경우(코로나 환자 0)와 데이터가 없는 경우를 구분하여 전처리하고 전처리 시 data가 없는 국가는 제외하고 제외된 국가 리스트를 제시하시오.**

-> 기존에 존재하지 않던 국가가 특정 날짜부터 나타나게 된다면 그 날짜 이전의 값들은 데이터가 없기 때문에 전처리가 필요하다. 또한 전날보다 오늘 데이터값이 감소하는 경우가 존재하기 때문에 음수값이 나타나게 된다. 이럴 때 0으로 치환하여 계산에 지장이 가지 않도록 전처리하였다.

**(3) 2021년 1년 동안 코로나 총발생자 수, 총사망자 수, 일평균 발생자 수, 일평균 사망자 수를 기준으로 가장 많은 20개 국가를 내림차순으로 정렬하고 총발생자수, 총사망자수, 일평균 발생자 수, 일평균 사망자 수를 리포트 하시오. (4가 지 기준 각각 sorting)**

-> 위의 과정을 통해 2021년 1월 1일부터 12월 31일까지 데이터를 활용해 국가의 일별 발생자, 사망자 수를 얻어, 이 값들을 국가마다 모두 더하여 2021년 총발생자와 사망자 수 데이터를 저장한다. 해당 국가가 발생자와 사망자 수를 낸 일수(보통 365일)를 나누어 일평균 발생자와 사망자 수를 구했기 때문에 소트 함수를 활용하여 각각의 기준으로 데이터를 정렬한다.

**(4) 2021년 1년 동안 대한민국에서 발생한 총 코로나 발생자 수와 총사망자 수 와 일평균 발생자 수와 일평균 사망자 수를 리포트 하시오.**

-> 최종적으로 얻어낸 데이터 프레임에서 대한민국이라는 국가명을 가진 데이터 값들을 리포트한다.

**2. 팀원 간 의견 수렴**

각자 코드를 작성 후 피드백을 통해 아이디어를 주고받았다. 피드백된 내용은 아래와 같다.

먼저, 제공된 데이터는 누적된 데이터이기 때문에 2021년 12월 31일 데이터에서 2020년 12월 31일의 데이터를 차감하는 방식으로도 2021년 한 해의 확진자 수와 사망자 수를 구할 수 있었다. 이때 NaN 값을 가진 데이터가 존재하기 때문에 isnull이 True인 경우 0으로 치환해주었다.

21년도 데이터를 얻기 위해 20211231 데이터와 20201231 데이터를 뺄셈하는 방법과 전날 누적 데이터 – 오늘 누적 데이터의 연산 후 365개의 파일을 더하는 방법으로 각각 진행하였다. 이때 일평균 데이터를 구하기 위해 총발생자와 사망자 수를 날짜만큼 나누어야 하는데, 후자의 방법을 활용하면 NaN 값을 가지는 일수만큼 365일에서 빼서 나누기에 용이했다.

하지만 2020년 12월 31일 데이터에는 존재하지 않던 ‘Micronesia’ 국가의 경우 2021년 중간에 확진자 1명이 추가되면서 새로운 인덱스 값으로 중간에 추가되었다. 그렇게 되면 그 이전의 날짜들의 값이 NaN인 행이 나타나게 되고, 어제 데이터와 오늘 데이터를 sub 함수를 통해 차감했을 때 결괏값 역시 NaN이 되어 오차가 발생하게 된다. 따라서 (2)번 문항의 데이터가 없는 경우를 isnull 함수로 빈 데이터를 가진 국가를 추출하고, 이전 날짜들의 값을 연산할 때 미리 0으로 치환하여 계산해 sub 함수에서 null 값이면 0으로 치환하여 계산하는 과정을 추가하였다. 이를 통해 연산 과정 중 데이터값이 누락되지 않도록 하였다.

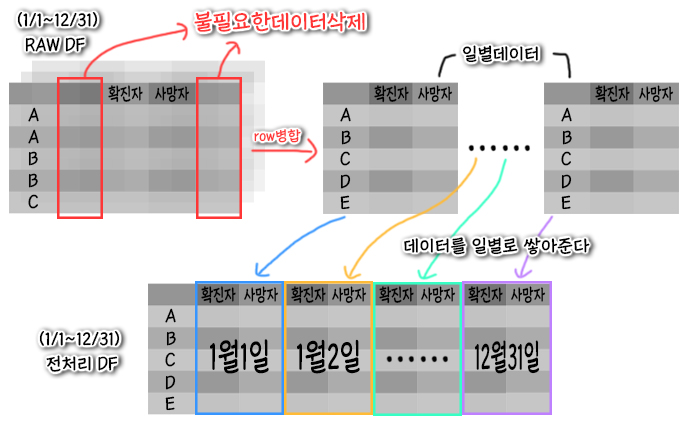
또한 제공된 데이터 자료를 확인하면 전날보다 값이 줄어든 경우가 존재한다. 이때 해당 날짜의 발생자와 사망자를 구하기 위해 전날 데이터 빼기 오늘 데이터를 한다면 음수 값이 나오게 된다. 따라서 음수 값을 가지는 경우 0으로 치환하여 최종 21년도 값을 계산하였다.

**3. 데이터 전처리 및 분석**

1. **데이터 전처리 로직**

텍스트이(가) 표시된 사진

자동 생성된 설명



테이블이(가) 표시된 사진

자동 생성된 설명

* 실제로 저장된 전처리 결과

1. **국가별 총확진자, 총사망자, 일평균 확진자, 일평균 사망자**

테이블이(가) 표시된 사진

자동 생성된 설명

1. **data이상값, 결측값 존재 국가 list**

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **상위20개 국가 내림차순 정렬**

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

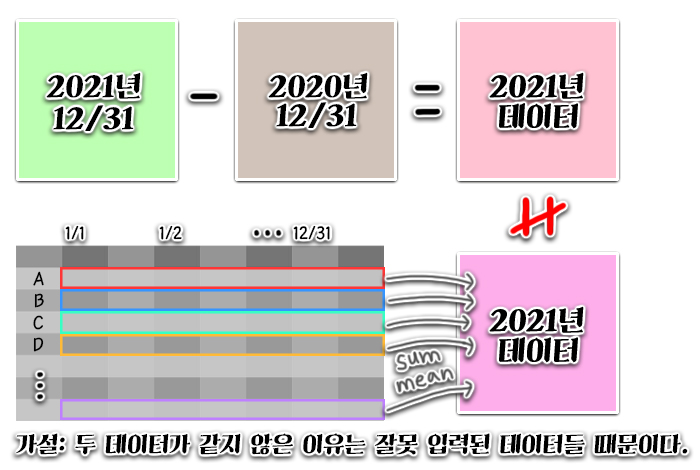
* 상위 20개 국가 정렬 결과

1. **대한민국 리포트**

텍스트이(가) 표시된 사진

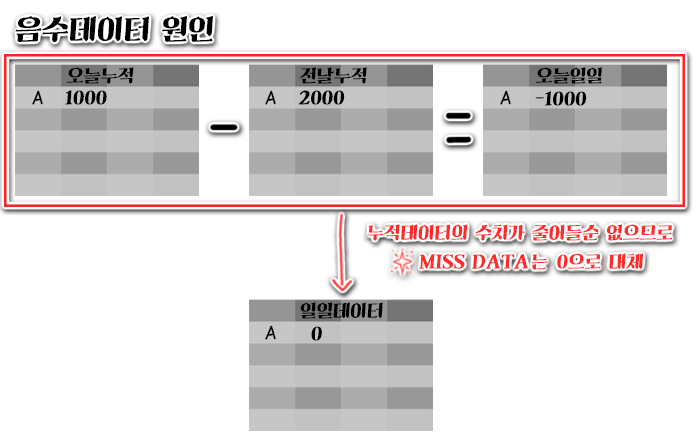
자동 생성된 설명

**4. 검증**

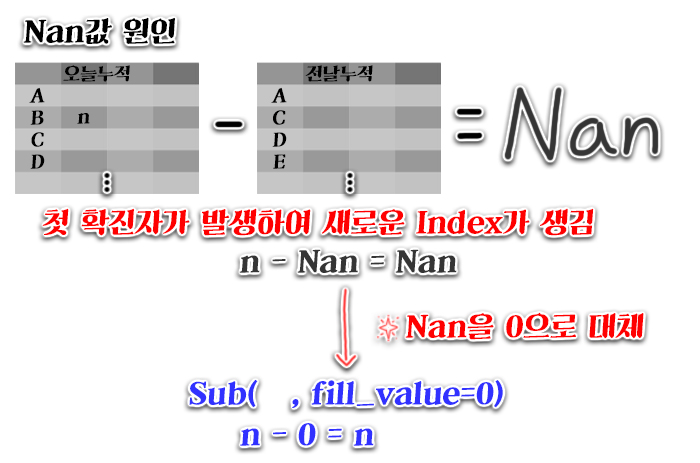
일일 합계로 구한 국가별 통계량과 2021년 누적 데이터에서 2020년 데이터를 단순히 뺀 통계량이 일치하는지 확인해보았다. 하지만 몇몇 국가 데이터들이 일치하지 않았다. 일치하지 않은 이유를 raw 데이터와 전처리 된 데이터를 찾아보다 음수값과 Nan을 발견하였다. ‘통계량이 일치하지 않는 이유는 잘못 입력된(음수값) 데이터 때문이다’라고 가설을 세웠다.

1. **음수 데이터 원인**

누적 데이터이므로 오늘 수치는 전날 수치와 같거나 많아야 한다. 하지만 잘못 입력이 되었는지 오히려 수치가 줄어든 데이터들이 발견되었다. 그 데이터들 때문에 일일 데이터가 음수가 나온 경우가 있었다. 인구수가 음수일 수는 없으므로 0으로 대체하고 카운트해주었다. 일일 평균을 구할 때 365에서 카운트된 숫자만큼 빼서 평균에 영향이 없도록 하였다.



1. **Nan값 원인**

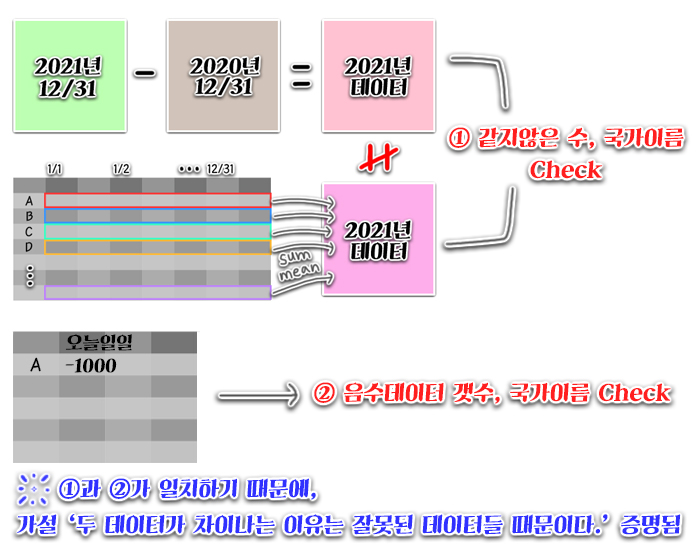


기존 확진자가 0명이어서 데이터가 없는 상태에서 오늘 첫 확진자가 발생하여 숫자에서 Nan을 빼야 하는 상황이 발생했다. Nan을 0으로 대체해 계산하고 카운트해주었다. 일일 평균을 계산할 때 365에서 카운트된 숫자만큼 빼서 평균에 영향 없게 하였다.

1. **가설 검증**

4에서 두 통계량이 차이 나는 부분과 4-(1)에서 구한 음수 데이터들과 비교를 해보았는데 국가들이 정확히 일치했다. 가설이 사실인 게 증명되었다.

|  |  |
| --- | --- |
| 두 통계량이 차이나는 국가들 | 음수데이터 발견된 국가들 |
|  |  |
|  | |



**5. 결론**

잘못 입력된 데이터를 제외하고 두 통계량이 100% 일치한다. 처음 생각한 로직대로 얻은 전처리 데이터가 유효하다는 것을 알 수 있고, 누락되거나 오류가 있는 데이터를 잡아내기에 더 효과적이었다.

이렇게 많은 수의 데이터를 정제 처리해보는 경험은 처음이었다. 이상 값, 결측치들이 예상치 못한 곳에서 발생하여 발견하고 원인을 해결하는 데 오랜 시간을 소비한 것 같다. 그 과정에서 데이터를 가지고 원하는 목적에 맞게끔 전처리 과정을 수행하며 문제 분석의 중요성을 알게 되었고 데이터 프레임에 대한 이해도가 올라가게 되었다.

또한 아이디어 공유를 통해 완성도 있는 코드가 만들어지는 것을 보면서 피드백의 필요성을 깨닫게 되었다. 팀원마다 각자 다른 방법으로 코드를 짠 뒤 더 효율적인 코드가 무엇일지 토론하면서 함께 성장할 수 있었다.

**6. 별첨(소스코드)**

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

**import** **os**

**import** **warnings**

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

warnings.simplefilter(action='ignore', category=**FutureWarning**)

*# import numpy as np*

os.getcwd()

pd.options.display.max\_rows =30

pd.options.display.min\_rows =20

pd.options.display.max\_columns =10

pd.set\_option('display.max.colwidth',15)

*##검토용 데이터프레임*

test=pd.DataFrame()

test1=pd.DataFrame()

testsub=pd.DataFrame()

test20=pd.DataFrame()

test21=pd.DataFrame()

nDf=pd.DataFrame() *#newDataframe 오늘데이터프레임*

tDf=pd.DataFrame() *#tempDataFrame 필요한 컬럼들만 추출한 데이터프레임을 저장하는 곳*

subDf=pd.DataFrame() *#전날누적DataFrame(빼기 하기위한 프레임)*

rs=pd.DataFrame() *#결과DataFrame*

final=pd.DataFrame() *#최종 DataFrame*

filename=os.listdir('covid19daily') *##폴더지정해서 폴더내 파일 이름 리스트에 저장*

df = pd.read\_csv('12-31-2020.csv') *##누적데이터이므로 20년데이터 제외하기 위해 2020년 12월31일 데이터불러옴*

*#데이터 불러와서 필요없는 columns들 제거*

df.drop(labels=['FIPS','Admin2','Province\_State','Last\_Update','Long\_','Lat','Recovered',

'Active','Combined\_Key','Incident\_Rate','Case\_Fatality\_Ratio'], axis=1)

*#국가,확진자, 사망자를 country\_region으로 그룹하고 합계낸 것을 tDf로 객체화*

tDf=df[['Country\_Region', 'Confirmed', 'Deaths']].groupby('Country\_Region').sum()

*#tDf 멀티컬럼명(날짜,확진자,사망자) 설정*

tDf.columns=[['12-31-2020','12-31-2020'],['Confirmed','Deaths']]

*#셀 합산을 위해 멀티컬럼 없는 subDf 객체 (tDf와 내용은 똑같다)*

subDf=df[['Country\_Region', 'Confirmed', 'Deaths']].groupby('Country\_Region').sum()

*###########검토용 데이터(2021년-2020년)#############*

test21temp = pd.read\_csv('covid19daily/12-31-2021.csv')

test21temp.drop(labels=['FIPS','Admin2','Province\_State','Last\_Update','Long\_','Lat',

'Recovered','Active','Combined\_Key','Incident\_Rate','Case\_Fatality\_Ratio'], axis=1)

test21=test21temp[['Country\_Region', 'Confirmed', 'Deaths']].groupby('Country\_Region').sum()

test21

test20.insert(loc=0,column='Confirmed',value=subDf['Confirmed'])

test20.insert(loc=1,column='Deaths',value=subDf['Deaths'])

test20

test=test21.sub(test20,fill\_value=0)

test

*########폴더내에 파일을 하나씩 불러오는 for문*

**for** i **in** range(len(filename)):

df1= pd.read\_csv('covid19daily/'+filename[i]) *##파일 하나씩 불러와서 df1로 객체화*

*#필요없는 columns들 버리기*

df1.drop(labels=['FIPS','Admin2','Province\_State','Last\_Update','Long\_','Lat','Recovered','Active','Combined\_Key','Incident\_Rate','Case\_Fatality\_Ratio'],axis=1)

*#국가,확진자, 사망자를 country\_region으로 그룹하고 합계낸 것을 nDf(오늘누적데이터)로 객체화*

nDf=df1[['Country\_Region','Confirmed','Deaths']].groupby('Country\_Region').sum()

*#nDf에서 subDf를 빼준다(오늘데이터(nDf) = 오늘누적데이터 - 전날누적데이터)*

*멀티컬럼이 있으면 빼기가 안되서 멀티컬럼이 없는 subDf를 따로 만들어준것*

*#값에서 Nan을 빼면 Nan이 나오기 때문에 확진자 없다가 21년에 첫 환자 발생하여 새로 나타난 국가(Micronesia)가 제대로 집계가 안되었었다*

*#1월21일 1명 발생했을때 제대로 집계를 못하고 그 뒤로 누적 확진자 수가 늘지 않아서 오늘누적-전날누적=0인 상황->총 확진자 0명 (실제데이터는 1명)*

*#fill\_value=0으로 결측값을 해결하였다.*

nDf=nDf.sub(subDf,fill\_value=0)

*#nDf(오늘데이터) 멀티컬럼명을 (날짜, confirmed, deaths)로 설정*

nDf.columns = [[filename[i][:-4],filename[i][:-4]],['Confirmed','Deaths']]

*#tDf(임시데이터프레임)와 오늘데이터프레임 국가 기준으로 합칩합으로 합치고 result로 객체화*

result = pd.merge(tDf, nDf, how='outer', left\_index=**True**, right\_index=**True**)

*#오늘누적데이터 subDf로 객체화 (다음 파일 불러와서 뺄셈할때 전날누적데이터로 쓰임)*

subDf = df1[['Country\_Region', 'Confirmed','Deaths']].groupby('Country\_Region').sum()

*#tDf(임시데이터프레임)에 날짜별로 계속 데이터를 쌓는다.*

tDf=result

*#result에 2020년 12월31일 컬럼 삭제하고 deep copy해서 rs로 객체화.*

rs=(result.drop(labels=['12-31-2020'],axis=1,level=0)).copy(deep=**True**)

print(rs)

*#rs 데이터프레임에서 멀티컬럼(날짜만)를 삭제(연산하는데 방해되고 에러뜨기때문에)*

rs.columns=rs.columns.droplevel(0)

*##############오입력 갯수 및 국가 리스트###########*

*#missC : missConfirmed 잘못입력된 확진자(누적데이터인데 전날보다 숫자가 적음)*

*#missD : missDeath 잘못입력된 사망자(누적데이터인데 전날보다 숫자가 적음)*

*#확진자컬럼 데이터중 음수인것들 row기준으로 갯수 세서 새로운 missC컬럼에 표시*

final['missC']=((rs['Confirmed']<0)).sum(axis=1)

*#음수데이터가 나온 index(국가) 추출*

ts=(final['missC']) >0

missC\_Country=final.index[ts]

missC\_Country

*#확진자컬럼내 결측값 row기준으로 갯수 세서 새로운 nullC컬럼에 표시*

final['nullC']=((rs['Confirmed']).isnull()).sum(axis=1)

ts=(final['nullC']) >0

nullC\_Country=final.index[ts]

nullC\_Country

*#사망자컬럼 데이터중 음수인것들 열기준으로 갯수 세서 새로운 missD컬럼에 표시*

final['missD']=(rs['Deaths']<0).sum(axis=1)

*#음수데이터가 나온 index(국가) 추출*

ts=(final['missD']) >0

missD\_Country=final.index[ts]

missD\_Country

*#사망자컬럼내 결측값 row기준으로 갯수 세서 새로운 nullD컬럼에 표시*

final['nullD']=((rs['Deaths']).isnull()).sum(axis=1)

ts=(final['nullD']) >0

nullD\_Country=final.index[ts]

nullD\_Country

*#음수데이터들 0으로 바꿔줌*

*#Nan데이터 0으로 바꿔주고 float64->int64 형변환*

*#(0이면 합계에는 영향없고 평균 구할땐 갯수만 365에서 빼주면된다)*

rs[rs<0]=0

rs.fillna(0, inplace=**True**, downcast='infer')

*'''*

*# (1) 일별 국가별 코로나 발생자수와 사망자 수를 기준으로 전처리 하시오. 일부*

*# 국가는 지역별로 코로나 발생자수와 사망자 수가 분리되어 있으니 국가별로*

*# 집계하고 국가, 총발생자수, 총사망자수, 일평균 발생자수, 일평균 사망자수 리*

*# 스트를 제시하시오.*

*# (난이도: 4, 배점: 30점)*

*'''*

*#####최종 출력#####*

*##confirmed컬럼을 row기준으로 합계내서 새로운 컬럼(총확진자)에 표시*

final.insert(loc=0,column='총확진자',value=rs['Confirmed'].sum(axis=1))

*#confirmed컬럼을 row기준으로 합계내고 [1년(365일)-음수데이터 갯수]로 나눈값을 새로운 컬럼(일평균 확진자)에 표시*

final.insert(loc=1,column='일평균 확진자',value=rs['Confirmed'].sum(axis=1)/(365-(final['missC']+final['nullC'])))

*#deaths컬럼을 row기준으로 합계내서 새로운 컬럼(총사망자)에 표시*

final.insert(loc=2,column='총사망자',value=rs['Deaths'].sum(axis=1))

*#deaths컬럼을 row기준으로 합계내고 [1년(365일)-음수데이터 갯수]로 나눈값을 새로운 컬럼(일평균 사망자)에 표시*

final.insert(loc=3,column='일평균 사망자',value=rs['Deaths'].sum(axis=1)/(365-(final['missD']+final['nullD'])))

*# final[['총확진자', '일평균 확진자', '총사망자', '일평균 사망자','missC', 'nullC', 'missD', 'nullD']]*

print(final)

print(test1)

*# final.to\_csv('result(1).csv')*

final.loc['Micronesia']

*'''*

*# (2) 데이터가 0인 경우(코로나 환자 0)와 데이터가 없는 경우를 구분하여 전처*

*# 리하고 전처리 시 data가 없는 국가는 제외하고 제외된 국가 리스트를 제시하*

*# 시오*

*'''*

*###데이터들이 일별 데이터가 아니라 누적되는 데이터라서 오늘데이터=오늘누적데이터-전날누적데이터로 계산했는데*

*###몇몇데이터들이 잘못 입력이 되었는지 오늘데이터가 음수값으로 나오는 경우가 있었다.*

*###그래서 이상치(음수값) 갯수를 파악하고 해당 국가 리스트도 작성하였다.(1번에서)*

*###음수값은 0으로 처리하여 sum,mean에 영향없게 하였다*

*###null값은 0으로 대체*

Ans = final.loc[final['총확진자']==0]

print("총합계 0인 국가 : 총",len(list(Ans.index)),"개 국가",list(Ans.index))

print('확진자 이상값 존재 국가(음수) : 총',len(missC\_Country),'개 국가',list(missC\_Country))

print('사망자 이상값 존재 국가(음수) : 총',len(missD\_Country),'개 국가',list(missD\_Country))

print('확진자 결측값 존재 국가(Nan) : 총',len(nullC\_Country),'개 국가',list(nullC\_Country))

print('사망자 결측값 존재 국가(Nan) : 총',len(nullD\_Country),'개 국가',list(nullD\_Country))

*# (3) 2021년 1년동안 코로나 총 발생자수, 총 사망자수, 일평균 발생자수, 일평균*

*# 사망자 수를 기준으로 가장 많은 20개 국가를 내림차순으로 정렬하고 총 발생*

*# 자수, 총 사망자수, 일평균 발생자수, 일평균 사망자 수를 리포트 하시오. (4가*

*# 지 기준 각각 sorting)*

*# (난이도: 4, 배점: 30점)*

totConfirmed20 = final.sort\_values('총확진자',ascending=**False**).reset\_index()

print(totConfirmed20.head(20))

meanConfirmed20 = final.sort\_values('일평균 확진자',ascending=**False**).reset\_index()

print(meanConfirmed20.head(20))

totDeaths20 = final.sort\_values('총사망자',ascending=**False**).reset\_index()

print(totDeaths20.head(20))

meanDeaths20 = final.sort\_values('일평균 사망자',ascending=**False**).reset\_index()

print(meanDeaths20.head(20))

*# (4) 2021년 1년동안 대한민국에서 발생한 총 코로나 발생자수와 총 사망자 수*

*# 와 일평균 발생자수와 일평균 사망자 수를 리포트 하시오.*

*# (난이도: 3, 배점: 20점)*

korea=final.loc['Korea, South']

print(final.loc['Korea, South'])

*##최종검토###*

*#일별로 합친것과 단순히 12월31일값만빼서 구한것과 차이가 있는지 보고*

*#그 차이가 음수값(오입력)에 의한 것인지 확인*

*'''*

*####혹시 소수점 데이터 있는지 확인해 본 코드*

*# def is\_int(n):*

*# return n % 1 != 0*

*# ts=is\_int(rs.iloc[:]).sum(axis=1)*

*# ts.sum()*

*'''*

test1.insert(loc=0,column='Confirmed',value=rs['Confirmed'].sum(axis=1))

test1.insert(loc=1,column='Deaths',value=rs['Deaths'].sum(axis=1))

test1

testsub= test1.sub(test,fill\_value=0)

testsub

a=((testsub['Confirmed']!=0))

ts1=(a)==**True**

z=a.index[ts1]

len(a.index[ts1])

print("")

b=((testsub['Deaths']!=0))

ts2=(b)==**True**

z1=b.index[ts2]

len(b.index[ts2])

print("확진자 총 합계 수치가 다른 국가 수:",len(a.index[ts1]),"확진자 음수값 존재 국가 수:",len(missC\_Country))

print("사망자 총 합계 수치가 다른 국가 수:",len(b.index[ts2]),"확진자 음수값 존재 국가 수:",len(missD\_Country))