R code 에 기반한

2021 년 코로나 데이터 분석

2022.10.14

A2 팀(신정훈, 조아진, 임성구, 조성운)

목차

1. 분석 개요 3

2. 원본 데이터 정의 5

3. 데이터 분석 과정 6

3.1 전처리 과정 7

3.2 문제풀이 10

4. 데이터 검정 17

4.1 정규성 검정 17

4.2 비모수 검정 20

5. 데이터 분석 22

5.1 국가별 사망률 22

5.2 월별 발생률 23

5.3 확진자, 사망자간 상관관계 23

6. 결론 24

7. 별첨 25

7.1 소스코드 25

7.2 Readme 원본 28

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 2021 년 일간 데이터에서 python code 로 다음을 수행하고 결과를 팀별로 리포트하시오.

1.1 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. (누적데이터인 경우 누적데이터로 해당 결과를 제시하고, 일별 데이터를 산출하여 총합과 일평균값을 산출하여 결과 비교)

-> 먼저 국가 내의 도(道), 주(州) 등으로 분리된 데이터를 국가명을 기준으로 합산하는 과정이 필요하다. 같은 국가명으로 데이터를 합산한 뒤 21 년 12 월 31 일에서 20 년 12 월 31 일 데이터를 뺄셈하여 21 년의 누적 데이터 결과를 추출할 것이다. 그리고 일별 데이터를 산출하기 위해 21 년 1 월 1 일부터 해당 데이터에서 하루 전 날의 데이터를 뺀 후 별도의 컬럼에 해당 데이터를 저장할 것이다. 이 과정으로 21 년 365 일치의 확진자, 사망자 수를 구하여 일별 데이터의 총합을 구한 후 누적 데이터 간의 결과와 비교할 것이다.

1.2 데이터가 0 인 경우(코로나 환자 0)와 데이터가 없는 경우를 구분하여 전처리하고 전처리 시 data 가 없는 국가는 제외하고 제외된 국가 리스트를 제시하시오.

-> 원본 데이터의 빈 부분이 있는지 확인하기 위하여 1.1 에서 구한 일별 데이터 상의 nan 값의 유무를 확인하고, nan 데이터의 발생 원인을 분석하여 0 으로 전처리 및 해당 국가명 리스트를 추출하여 제시할 것이다.

1.3 2021 년 1 년 동안 코로나 총발생자 수, 총사망자 수, 일평균 발생자 수, 일평균 사망자 수를 기준으로 가장 많은 20 개 국가를 내림차순으로 정렬하고 총발생자수, 총사망자수, 일평균 발생자 수, 일평균 사망자 수를 리포트 하시오. (4 가 지 기준 각각 sorting)

-> 1.1 에서 구한 누적 데이터로부터 평균 값을 구한 후 상위 20 개국의 데이터를 추출하여 제시할 것이다. 이 과정에서 라이브러리 dplyr 패키지에 있는 arrange() 함수가 활용될 것이다.

1.4 2021 년 1 년 동안 대한민국에서 발생한 총 코로나 발생자 수와 총사망자 수 와 일평균 발생자 수와 일평균 사망자 수를 리포트 하시오.

-> 1.1 의 데이터로부터 행 이름이 대한민국(Korea, South)인 부분을 subset() 함수를 이용하여 추출한 후 데이터를 제시할 것이다.

2. 원본 데이터 정의

데이터 분석에 앞서 원본 데이터에 대한 정의를 위해 첨부된 readme 파일을 참조하였으며, 컬럼의 의미 확인 및 분석에 필요한 컬럼을 선별하였다. 선별된 컬럼과 해당 컬럼의 정의는 아래 표와 같다.

컬럼명	의미
Province_State	도(道), 주(州) 또는 자치구 이름
Country_Region	국가, 지역, 독립국의 이름
Confirmed	확진 예상자가 포함된 확진자 명 수(누적)
Deaths	사망 예상자가 포함된 사망자 명 수(누적)

위 표를 통해 데이터가 국가보다 작은 단위로 집계되어 있는 것을 확인할 수 있으므로 해당 데이터를 국가 단위로 그룹화하여 합계 데이터를 산출한 후 분석을 진행할 것이다.

확진자와 사망자 데이터의 경우 readme 파일에 누적 데이터라는 내용은 별도로 기재되어 있지 않으나, 데이터 간의 증감 확인 결과 극소수의 경우를 제외하면 모든 데이터 양이 증가하고 있어 누적 데이터로 정의하였다. 데이터가 감소하는 경우는 소스 데이터의 변경 혹은 입력 착오로 인하여 발생하였음을 readme 파일에 기재된 내용을 통해 알 수 있다.

3. 데이터 분석

변수	주석	변수정의
rawData	3.1.3	2020/12/31 원본 데이터프레임
yesterdayDF	3.1.3	전날 누적 데이터프레임
tmepDF	3.1.3	Merge 용 일별 데이터프레임
test_sub_DF	3.1.4	검정용 데이터(21 년누적-20 년누적)
FN	3.1.5	CSV 파일 리스트
fileName	3.1.5	CSV 파일 리스트중 2021 년도 파일
todayDF	3.1.6	오늘까지 누적 데이터프레임
dailyDF	3.1.6	오늘 일별 데이터프레임
resultDF	3.1.6	일별 데이터 저장하는 데이터프레임
missC	3.1.7	2021 년 일별(확진자) 음수값 개수
missD	3.1.7	2021 년 일별(사망자) 음수값 개수
nullC	3.1.7	2021 년 일별(확진자) NA 데이터 개수
nullD	3.1.7	2021 년 일별(사망자) NA 데이터 개수
confirm	3.1.7	2021 년 일별(확진자) 데이터
death	3.1.7	2021 년 일별(사망자) 데이터
findalDF	3.2.1	최종 21 년(확진자,사망자)두가지 방식 데이터프레임, 이상값,결측값 수 추가

▲ 변수의 데이터 정의

3.1 전처리 과정

전처리 순서
사용할 library 생성
-> function 생성
-> 2020/12/31, 2021/12/31 COVID19 데이터추출
-> 2020/12/31, 2021/12/31 데이터 전처리
-> 2021 년 데이터 파일 리스트 생성
-> 2021 년 일별로 감염자, 발생자 데이터 전처리
-> 음수값, 결측값 전처리
-> 음수값, 결측치 0 처리

3.1.1 사용할 library 생성

```
library(dplyr)
library(stringr)
library(corrgram)
library(ggplot2)
```

3.1.2 function 생성

```
###빼기 함수(NA 값 처리후 뺄셈)###
sub_ <- function(x,y){
        return(ifelse(is.na(x),0,x)-ifelse(is.na(y),0,y))
}
###평균 함수(NA,이상 갯수 만큼 제외한 평균)###
avg_ <- function(x,y,z){
        return(round(x/(365-(y+z)),2))
}
```

-> 사용자 정의 함수로 sub_ 함수를 생성하였다. ifelse 조건문을 사용하여 nan 값을 가지면 0 으로 바꾸어 데이터 누락을 방지하였다. 사용자 정의 함수 avg_ 함수는 365 일에서 데이터가 누락된 일수만큼 뺄셈하여 평균을 구하였다.

3.1.3 2020/12/31, 2021/12/31 COVID19 데이터추출

```
###2020 년 데이터 로드###
rawData <- read.csv("covid19daily/12-31-2020.csv", header = T)
yesterdayDF <- rawData %>% group_by(Country_Region)%>%
               summarise(Confirmed=sum(Confirmed), Deaths=sum(Deaths))
tempDF <- yesterdayDF
```

```
###TEST 데이터####
testData <- read.csv("covid19daily/12-31-2021.csv", header = T)
testData <- testData %>% group_by(Country_Region)%>%
            summarise(Confirmed=sum(Confirmed), Deaths=sum(Deaths))
```

-> Country_Region 이라는 컬럼을 기준으로 그룹화하였다. 국가명을 기준으로 합산하여 국가별 확진자와 사망자 데이터를 구하였다.

3.1.4 2020/12/31, 2021/12/31 데이터 전처리

```
testData <- merge(yesterdayDF,testData, by ='Country_Region', all = TRUE)
test_sub_DF <- testData%>% group_by(Country_Region) %>%
               summarise(Confirmed=sub_(Confirmed.y,Confirmed.x),
               Deaths=sub_(Deaths.y,Deaths.x))
test_sub_DF$T_Mean_C <- round(test_sub_DF$Confirmed/365,2)
test_sub_DF$T_Mean_D <- round(test_sub_DF$Deaths/365,2)
```

-> 데이터끼리 합치기 위해 merge 함수를 사용하였다. 국가가 중간에 늘어나기 때문에 옵션을 통해 데이터프레임 크기가 큰 쪽에 맞춰지도록 하였다.

3.1.5 2021 년 데이터 파일 리스트 생성

```
FN <- list.files(path = 'covid19daily') # 파일 이름
fileName<- unlist(str_extract_all(FN,"[0-9,-]*-2021"))
```

-> 제공된 파일이 저장된 위치를 FN 변수에 담고, fileName 이라는 변수에 날짜 부분 문자열을 추출하여 저장하였다.

3.1.6 2021 년 일별로 확진자, 발생자 데이터 전처리

```
###파일 불러와서 일별로 계산후 데이터프레임에 합치기###
for(i in fileName){
  todayDF <- read.csv(paste0("covid19daily/",i,".csv"),header=T)
  todayDF <- todayDF %>%
             group_by(Country_Region) %>%
             summarise(Confirmed=sum(Confirmed),
             Deaths=sum(Deaths))
  dailyDF <- merge(yesterdayDF,todayDF, by ='Country_Region', all = TRUE)
  dailyDF <- dailyDF%>% group_by(Country_Region) %>%
             summarise(Confirmed=sub_(Confirmed.y,Confirmed.x),
             Deaths=sub_(Deaths.y,Deaths.x))
  date_month <- unlist(str_extract_all(i,"[0-9]{1,2}-[0-9]{1,2}"))
  names(dailyDF) <- c("Country_Region",paste0(date_month,"_확진자"),
                      paste0(date_month,"_사망자"))
  resultDF <- merge(tempDF,dailyDF, by ='Country_Region', all = TRUE)
  yesterdayDF <- todayDF
  tempDF <- resultDF
```

```
}
rDF <- resultDF[,c(-2,-3)]
write.csv(rDF, '일일데이터.csv')
```

-> for 문을 사용해서 21 년 1 월 1 일 파일부터 차례대로 불러왔다. todayDF 변수에 불러온 데이터프레임을 저장한 뒤, 국가명을 기준으로 확진자와 사망자 수를 합하였다. 198 개국 각각의 확진자와 사망자가 저장되다가 특정 날짜부터 국가가 하나 추가되어 199 개국이 된다. 오늘 데이터프레임과 어제 데이터프레임을 merge 함수를 통해 합쳐서 dailyDF 라는 변수명으로 저장하였다. 오늘 누적 데이터(Confirmed.y, Deaths.y) – 어제 누적 데이터(Confirmed.x, Deaths.x)를 하여 Confirmed 와 Death 에 각각 계산된 일별 데이터 값을 저장하였다. date_month 라는 변수에 오늘 날짜를 추출하여 저장한 뒤, dailyDF 컬럼명을 수정할 때 활용하였다. 최종적으로 일별 데이터들을 누적하여 저장한 resultDF 를 얻어내고, 어제 데이터(yesterdayDF)에 오늘 데이터(todayDF)를 넣어주었다.

3.1.7 음수값, 결측값 전처리

```
#####음수값,결측값 갯수 세고 국가 체크##########
confirm <- rDF[,c(1,seq(2, ncol(rDF),2))]
death <- rDF[,c(1,seq(3, ncol(rDF),2))]
missC <- confirm
missC <- apply(confirm[,c(2:ncol(confirm))]<0,1,sum,na.rm=T)
         confirm[missC!=0,1]
missD <- death
missD <- apply(death[,c(2:ncol(death))]<0,1,sum,na.rm=T)
         death[missD!=0,1]
nullC <- apply(is.na(confirm),1,sum)
nullD <- apply(is.na(death),1,sum)
confirm[nullC!=0,1]
death[nullD!=0,1]
```

-> rDF 는 21 년 365 일치의 일별 데이터가 저장된 데이터프레임이다. 날짜마다 확진자, 사망자 순서로 저장되었기 때문에 rDF 의 짝수번 컬럼을 가져오면 확진자 데이터만 얻어낼 수 있다. 사망자는 홀수번 컬럼을 가져와 confirm 과 death 라는 변수에 각각 데이터프레임으로 저장하였다. 이때 음수값을 가지는 컬럼의 수를 계산하여 missC, missD 로 저장하고, nan 값을 가지는 컬럼의 수를 계산하여 nullC, nullD 로 저장하였다.

3.1.8 음수값, 결측치 0 처리

```
####음수값, 결측치 0 으로 대체 #######
rDF <- replace(rDF[,c(1,2:ncol(rDF))],rDF[,c(1,2:ncol(rDF))]<0,0)
rDF[is.na(rDF)] <- 0
```

-> 음수와 nan 을 데이터로 가지면 0 으로 교체하여 rDF 에 저장하였다.

3.2 문제풀이

3.2.1 일별 국가별 코로나 발생자수와 사망자 수를 기준으로 전처리 하시오. 일부 국가는 지역별로 코로나 발생자수와 사망자 수가 분리되어 있으니 국가별로 집계 하고 국가, 총발생자수, 총사망자수, 일평균 발생자수, 일평균 사망자수 리스트를 제시하시오. (누적데이터인 경우 누적데이터로 해당 결과를 제시하고, 일별 데이터를 산출하여 총합과 일평균값을 산출하여 결과 비교)

```
finalDF <- rDF
len <- ncol(finalDF)
finalDF$Confirmed <- apply(finalDF[,seq(2, len,2)],1,sum,na.rm=T)
finalDF$Deaths <- apply(finalDF[,seq(3, len,2)],1,sum,na.rm=T)
finalDF <- subset(finalDF,select=c(Country_Region,Confirmed,Deaths))
finalDF$MeanConfirmed <- avg_(finalDF$Confirmed,missC,nullC)
finalDF$MeanDeaths <- avg_(finalDF$Deaths,missD,nullD)
finalDF$missC <- missC
finalDF$missD <- missD
finalDF$nullC <- nullC
finalDF$nullD <- nullD
finalDF <- merge(finalDF,test_sub_DF, by ='Country_Region', all = TRUE)
print(finalDF)
write.csv(finalDF, '최종 DataFrame.csv')
```

-> 일별 데이터를 finalDF 에 저장하고 합산하여 21 년 총확진자와 총사망자를 구하였다. 또한 일평균 발생자와 사망자를 avg_() 함수로 계산하여 MeanConfirmed, MeanDeaths 라는 컬럼명으로 저장하였다.

일별 데이터 결과표 (국가명순 상위 20 개)

	Country_Region	Confirmed.x	Deaths.x	MeanConfirmed	MeanDeaths
1	Afghanistan	105754	5167	289.74	14.16
2	Albania	151908	2036	416.19	5.58
3	Algeria	118822	3520	325.54	9.64
4	Andorra	15691	56	42.99	0.15
5	Angola	64040	1365	175.45	3.74
6	Antarctica	11	0	0.03	0
7	Antigua and Barbuda	4125	114	11.33	0.31
8	Argentina	4028894	73924	11038.07	202.53
9	Armenia	185521	5151	508.28	14.15
10	Australia	397071	1346	1087.87	3.71
11	Austria	914748	9329	2506.16	25.56
12	Azerbaijan	398247	5717	1091.09	15.66
13	Bahamas	16653	547	45.75	1.5

14	Bahrain	189387	1042	518.87	2.85
15	Bangladesh	1072029	20513	2937.07	56.2
16	Barbados	28182	253	77.21	0.69
17	Belarus	504826	4154	1383.08	11.38
18	Belgium	1458847	8803	3996.84	24.12
19	Belize	22064	358	60.45	0.99
20	Benin	21684	117	59.41	0.32

▲ 노란칠: 누적 데이터와 차이가 존재하는 값

누적 데이터 결과표 (국가명 상위 20 개)					
	Country_Region	Confirmed.y	Deaths.y	T_Mean_C	T_Mean_D
1	Afghanistan	105754	5167	289.74	14.16
2	Albania	151908	2036	416.19	5.58
3	Algeria	118822	3520	325.54	9.64
4	Andorra	15691	56	42.99	0.15
5	Angola	64040	1365	175.45	3.74
6	Antarctica	11	0	0.03	0
7	Antigua and Barbuda	4124	114	11.3	0.31
8	Argentina	4028894	73924	11038.07	202.53
9	Armenia	185521	5149	508.28	14.11
10	Australia	397071	1344	1087.87	3.68
11	Austria	914748	9329	2506.16	25.56
12	Azerbaijan	398247	5717	1091.09	15.66
13	Bahamas	16605	547	45.49	1.5
14	Bahrain	189387	1042	518.87	2.85
15	Bangladesh	1072029	20513	2937.07	56.2
16	Barbados	28182	253	77.21	0.69
17	Belarus	504826	4154	1383.08	11.38
18	Belgium	1458847	8803	3996.84	24.12
19	Belize	22064	354	60.45	0.97
20	Benin	21684	117	59.41	0.32

▲ 노란칠: 일별 데이터와 차이가 존재하는 값

-> 두 결과 사이에 미세한 차이가 존재하는 것을 확인할 수 있다.

3.2.2 데이터가 0 인 경우(코로나 환자 0)와 데이터가 없는 경우를 구분하여 전처리 하고 전처리 시 data 가 없는 국가는 제외하고 제외된 국가 리스트를 제시하시오.

```
print("확진자 이상값 존재 국가 리스트");confirm[missC!=0,1]
print("사망자 이상값 존재 국가 리스트");death[missD!=0,1]
print("확진자 결측값 존재 국가 리스트");confirm[nullC!=0,1]
print("사망자 결측값 존재 국가 리스트");death[nullD!=0,1]
```

-> 이상값과 결측값이 존재하면 국가명을 가져온다.

확진자 이상값 존재 국가 리스트		
"Antigua and Barbuda"	"Bahamas"	"Brunei"
"Czechia"	"Denmark"	"Dominica"
"Ecuador"	"France"	"Germany"
"Iceland"	"Ireland"	"Israel"
"Kazakhstan"	"Liberia"	"Mozambique"
"New Zealand"	"Philippines"	"Seychelles"
"Spain"	"Sudan"	"United Kingdom"
사망자 이상값 존재 국가 리스트		
"Armenia"	"Australia"	"Belize"
"Bosnia and Herzegovina"	"Burma"	"Cabo Verde"
"Canada"	"Chile"	"Czechia"
"Fiji"	"Germany"	"Guatemala"
"Haiti"	"Honduras"	"Ireland"
"Israel"	"Kazakhstan"	"Liberia"
"Montenegro"	"Mozambique"	"Namibia"
"Norway"	"Peru"	"Sao Tome and Principe"
"Sweden"	"Switzerland"	"Uganda"
확진자 결측값 존재 국가 리스트		
"Micronesia"		
사망자 결측값 존재 국가 리스트		
"Micronesia"		

-> 중간부터 추가된 국가인 "Micronesia"에만 결측지가 존재하는 것을 알 수 있다.

3.2.3 2021 년 1 년동안 코로나 총 발생자수, 총 사망자수, 일평균 발생자수, 일평균 사망자수를 기준으로 가장 많은 20 개 국가를 내림차순으로 정렬하고 총 발생자 수, 총 사망자수, 일평균 발생자수, 일평균 사망자 수를 리포트 하시오. (4 가지 기 준 각각 sorting)

```
totConfirmed20 <- finalDF[order(finalDF$Confirmed.x,finalDF$Deaths.x,
                  finalDF$MeanConfirmed,finalDF$
                  MeanDeaths,decreasing = TRUE),]
rownames(totConfirmed20) <- 1 : length(rownames(totConfirmed20))
print(head(totConfirmed20,n=20))
meanConfirmed20 <-finalDF[order(finalDF$MeanConfirmed,finalDF$
                  Confirmed.x,finalDF$Deaths.x,finalDF$
                  MeanDeaths,decreasing = TRUE),]
rownames(meanConfirmed20) <- 1 : length(rownames(meanConfirmed20))
print(head(meanConfirmed20,n=20))
totDeaths20 <- finalDF[order(finalDF$Deaths.x,finalDF$
               Confirmed.x,finalDF$MeanConfirmed,finalDF$
               MeanDeaths,decreasing = TRUE),]
rownames(totDeaths20) <- 1 : length(rownames(totDeaths20))
print(head(totDeaths20,n=20))

meanDeaths20 <- finalDF[order(finalDF$MeanDeaths,finalDF$
                Confirmed.x,finalDF$Deaths.x,finalDF$
                MeanConfirmed,decreasing = TRUE),]
rownames(meanDeaths20) <- 1 : length(rownames(meanDeaths20))
print(head(meanDeaths20,n=20))
write.csv(totConfirmed20, '확진자정렬.csv')
write.csv(meanConfirmed20, '사망자정렬.csv')
write.csv(totDeaths20, '평균확진자정렬.csv')
write.csv(meanDeaths20, '평균사망자정렬.csv')
```

-> 각 컬럼을 기준으로 내림차순 정렬하기 위해 order 함수를 사용하여 옵션으로 decreasing=TRUE 를 지정해주었다. 이때 누적 데이터가 아닌 일별 데이터를 기준으로 진행했다.

확진자 기준 상위 20 개국					
	Country_Region	Confirmed.x	Deaths.x	MeanConfirmed	MeanDeaths
1	US	34645729	475132	94919.81	1301.73
2	India	24574870	332492	67328.41	910.94
3	Brazil	14610807	424262	40029.61	1162.36
4	United Kingdom	10521815	82470	28985.72	225.95
5	France	7748967	59971	21288.37	164.3
6	Turkey	7273898	61480	19928.49	168.44
7	Russia	7193058	246400	19707.01	675.07
8	Germany	5447352	78321	15006.48	215.17
9	Iran	4969259	76383	13614.41	209.27
10	Spain	4440910	38568	12233.91	105.67
11	Argentina	4028894	73924	11038.07	202.53

12	Italy	4018517	63243	11009.64	173.27
13	Indonesia	3519522	121956	9642.53	334.13
14	Colombia	3514665	86729	9629.22	237.61
15	Poland	2813337	68500	7707.77	187.67
16	Ukraine	2760229	82807	7562.27	226.87
17	Malaysia	2645076	31016	7246.78	84.98
18	Mexico	2553629	173621	6996.24	475.67
19	South Africa	2401125	62676	6578.42	171.72
20	Netherlands	2374752	9867	6506.17	27.03

-> 확진자가 많을수록 사망자가 높지만 항상 확진자 순위와 사망자 순위가 동일하지는 않았다.

사망자 기준 상위 20 개국					
	Country_Region	Confirmed.x	Deaths.x	MeanConfirmed	MeanDeaths
1	US	34645729	475132	94919.81	1301.73
2	Brazil	14610807	424262	40029.61	1162.36
3	India	24574870	332492	67328.41	910.94
4	Russia	7193058	246400	19707.01	675.07
5	Mexico	2553629	173621	6996.24	475.67
6	Indonesia	3519522	121956	9642.53	334.13
7	Peru	1281694	110329	3511.49	303.94
8	Colombia	3514665	86729	9629.22	237.61
9	Ukraine	2760229	82807	7562.27	226.87
10	United Kingdom	10521815	82470	28985.72	225.95
11	Germany	5447352	78321	15006.48	215.17
12	Iran	4969259	76383	13614.41	209.27
13	Argentina	4028894	73924	11038.07	202.53
14	Poland	2813337	68500	7707.77	187.67
15	Italy	4018517	63243	11009.64	173.27
16	South Africa	2401125	62676	6578.42	171.72
17	Turkey	7273898	61480	19928.49	168.44
18	France	7748967	59971	21288.37	164.3
19	Romania	1176628	42985	3223.64	117.77
20	Philippines	2369926	42260	6510.79	115.78

일평균 확진자 기준 상위 20 개국					
	Country_Region	Confirmed.x	Deaths.x	MeanConfirmed	MeanDeaths
1	US	34645729	475132	94919.81	1301.73
2	India	24574870	332492	67328.41	910.94
3	Brazil	14610807	424262	40029.61	1162.36
4	United Kingdom	10521815	82470	28985.72	225.95
5	France	7748967	59971	21288.37	164.3
6	Turkey	7273898	61480	19928.49	168.44
7	Russia	7193058	246400	19707.01	675.07
8	Germany	5447352	78321	15006.48	215.17
9	Iran	4969259	76383	13614.41	209.27
10	Spain	4440910	38568	12233.91	105.67
11	Argentina	4028894	73924	11038.07	202.53
12	Italy	4018517	63243	11009.64	173.27
13	Indonesia	3519522	121956	9642.53	334.13
14	Colombia	3514665	86729	9629.22	237.61
15	Poland	2813337	68500	7707.77	187.67
16	Ukraine	2760229	82807	7562.27	226.87
17	Malaysia	2645076	31016	7246.78	84.98
18	Mexico	2553629	173621	6996.24	475.67
19	South Africa	2401125	62676	6578.42	171.72
20	Philippines	2369926	42260	6510.79	115.78

일평균 사망자 기준 상위 20 개국					
	Country_Region	Confirmed.x	Deaths.x	MeanConfirmed	MeanDeaths
1	US	34645729	475132	94919.81	1301.73
2	Brazil	14610807	424262	40029.61	1162.36
3	India	24574870	332492	67328.41	910.94
4	Russia	7193058	246400	19707.01	675.07
5	Mexico	2553629	173621	6996.24	475.67
6	Indonesia	3519522	121956	9642.53	334.13
7	Peru	1281694	110329	3511.49	303.94
8	Colombia	3514665	86729	9629.22	237.61
9	Ukraine	2760229	82807	7562.27	226.87

10	United Kingdom	10521815	82470	28985.72	225.95
11	Germany	5447352	78321	15006.48	215.17
12	Iran	4969259	76383	13614.41	209.27
13	Argentina	4028894	73924	11038.07	202.53
14	Poland	2813337	68500	7707.77	187.67
15	Italy	4018517	63243	11009.64	173.27
16	South Africa	2401125	62676	6578.42	171.72
17	Turkey	7273898	61480	19928.49	168.44
18	France	7748967	59971	21288.37	164.3
19	Romania	1176628	42985	3223.64	117.77
20	Philippines	2369926	42260	6510.79	115.78

-> 총 합산 데이터의 순위와 일평균 데이터의 순위는 비슷하게 나타났지만 완벽하게 일치하지는 않았다.

3.2.4 2021 년 1 년동안 대한민국에서 발생한 총 코로나 발생자수와 총 사망자 수와 일평균 발생자수와 일평균 사망자 수를 리포트 하시오.

```
KOR <- finalDF%>%subset(finalDF$Country=="Korea, South")
write.csv(KOR, '대한민국.csv')
```

-> 국가명이 Korea, South 인 행을 추출하였다. 이때 가져온 데이터는 누적 데이터가 아닌 일별 데이터를 가공한 값들이다.

대한민국 결과표					
	Country_Region	Confirmed.x	Deaths.x	MeanConfirmed	MeanDeaths
1	Korea, South	573484	4708	1571.19	12.9

4. 데이터 검정

4.1. 정규성 검정

Sample1 : 2021 년 12 월 31 일까지 누적 데이터에서 2020 년 12 월 31 일까지 누적데이터를 뺀 확진자 수(2021 년 확진자수)

Sample2 : 누적데이터를 통해서 구한 일일 데이터로 총계 낸 확진자 수(2021 년 확진자수)

Sample1, Sample2 에 대해서

귀무가설(H_0) : 데이터가 정규분포를 따른다.

대립가설(H_1) : 데이터가 정규분포를 따르지 않는다.

4.1.1 과 4.1.2 의 결과로 귀무가설을 기각하고 정규분포를 따르지 않는걸 알 수 있다.

4.1.1 샤피로 윌크 검정

shapiro-wilk test 를 통해서 유의수준 0.05 일때 P-value 가 0.05 이하면 귀무가설은 기각되고 해당 데이터는 정규분포라고 할 수 없다.

```
> shapiro.test(test_sum_DF$Confirmed)
```

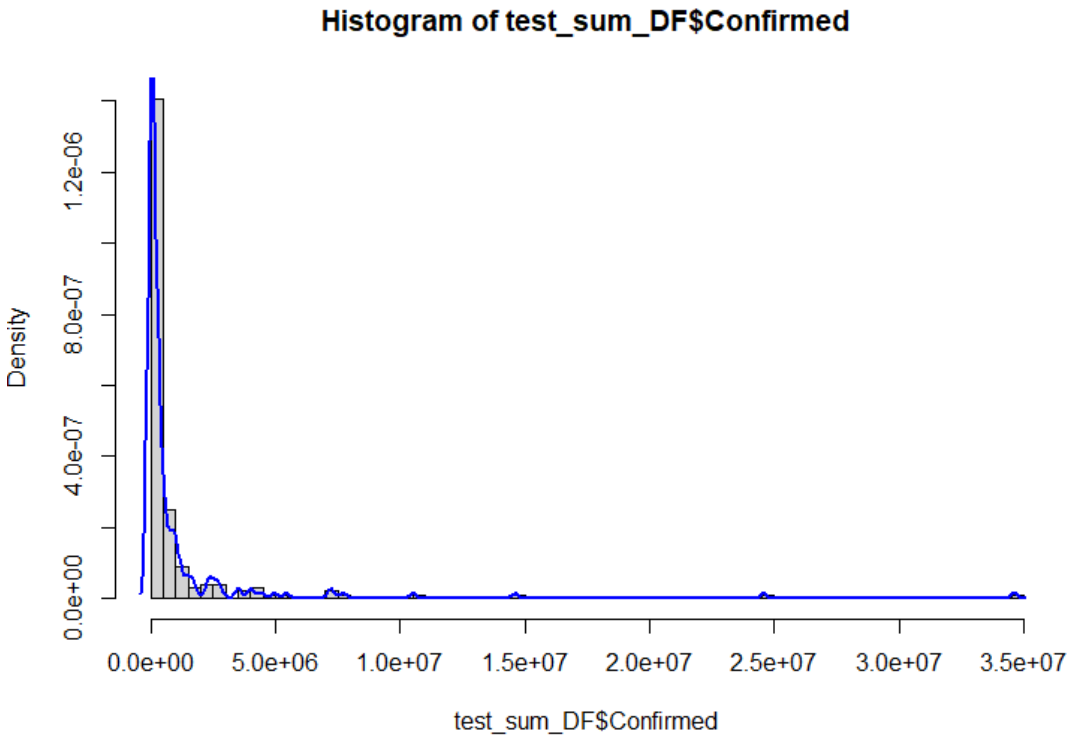
```
        Shapiro-Wilk normality test
data:  test_sum_DF$Confirmed
W = 0.29865, p-value < 2.2e-16
```

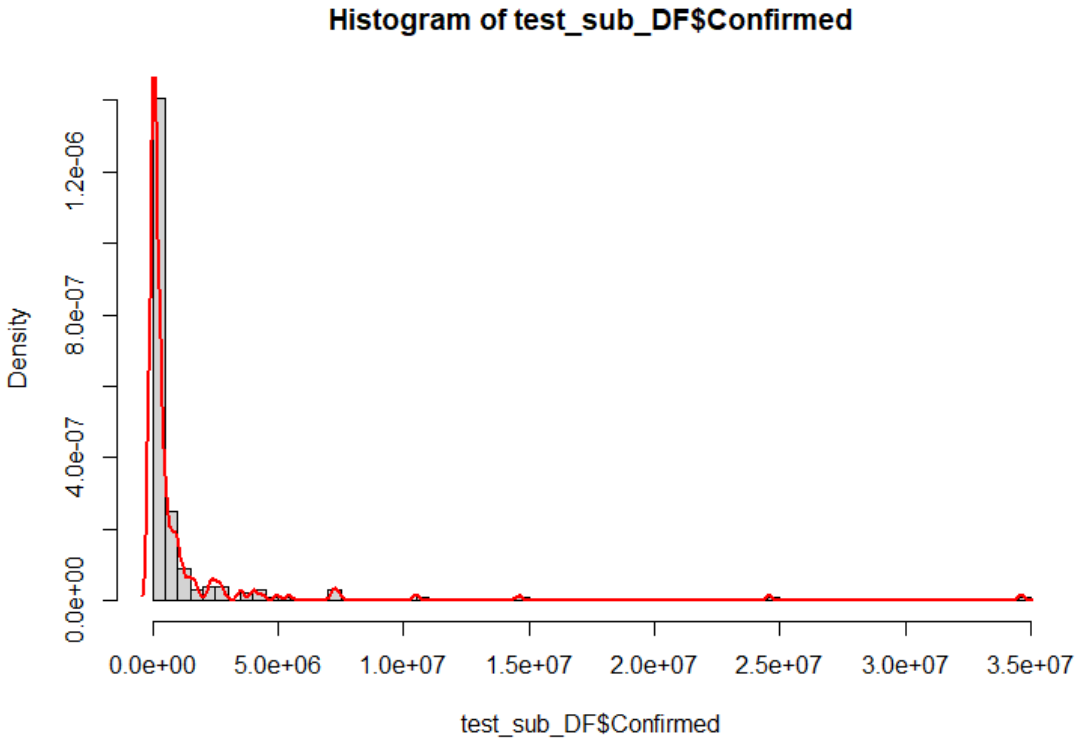
```
> shapiro.test(test_sub_DF$Confirmed)
```

```
        Shapiro-Wilk normality test
data:  test_sub_DF$Confirmed
W = 0.29788, p-value < 2.2e-16
```

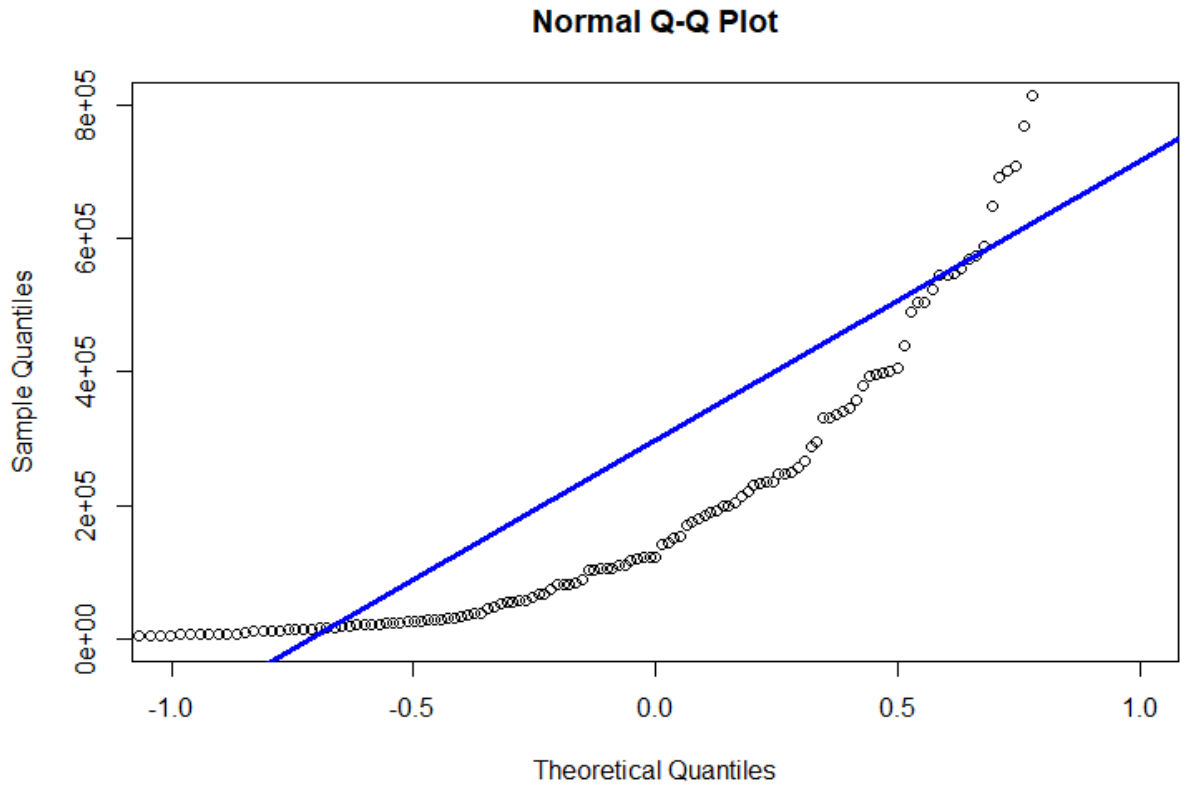
-> 두 데이터 모두 p-value 가 2.2e-16 이하로 귀무가설(H_0)을 기각하고 데이터가 정규분포를 따르지 않는다고 할 수 있다.

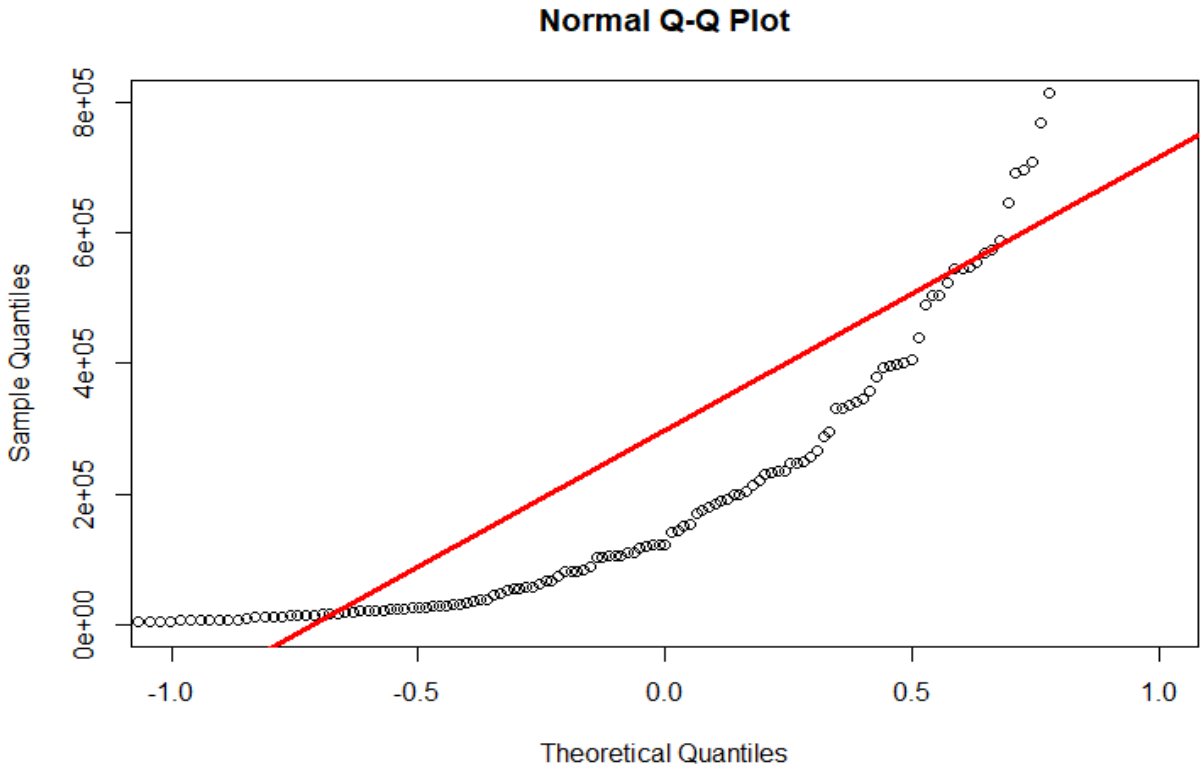
4.1.2 히스토그램, Q-Q(Quantile-Quantile) 시각화





-> 두 데이터가 멱함수 분포를 따르고 있다.





-> Q-Q plot 이 곡선형을 이루고 있다. 그러므로 두 데이터는 정규성을 따르고 있지 않다.

4.2 비모수 검정

두 데이터가 정규분포를 따르지 않기 때문에 비모수검정 중 하나인 Wilcoxon rank sub test 를 확진자데이터에 이용하였다.

4.2.1 Wilcoxon rank sub test

Sample1 : 2021 년 12 월 31 일까지 누적 데이터에서 2020 년 12 월 31 일까지 누적데이터를 뺀 확진자 수(2021 년 확진자수)

Sample2 : 누적데이터를 통해서 구한 일일 데이터로 합계 낸 확진자 수(2021 년 확진자수)

귀무가설(H_0) : 두 집단간 차이가 없다.

대립가설(H_1) : 두 집단간 차이가 있다.

```
>wilcox.test(test_sum_DF$Confirmed,test_sub_DF$Confirmed,alternative='g',conf.int=F,conf.level=0.95)
```

```
        Wilcoxon rank sum test with
        continuity correction

data:  test_sum_DF$Confirmed and test_sub_DF$Confirmed
W = 19815, p-value = 0.4951
alternative hypothesis: true location shift is greater than 0
```

-> 유의수준 5%에서 p-value = 0.4951, p-value=0.5002 로서 0.05 보다 크니 대립가설(H_1)을 기각하고 두 집단은 차이가 없다고 할 수 있다.

4.3 상관관계 분석

```
> cor.test(test_sum_DF$Confirmed,test_sub_DF$Confirmed)
```

```
        Pearson's product-moment correlation

data:  test_sum_DF$Confirmed and test_sub_DF$Confirmed
t = 1892.1, df = 197, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9999636 0.9999792
sample estimates:
      cor
0.9999725
```

-> 95% 신뢰수준:[0.9999636, 0.9999792], 상관관계 0.99997 로 강한 양의 상관관계를 보인다.

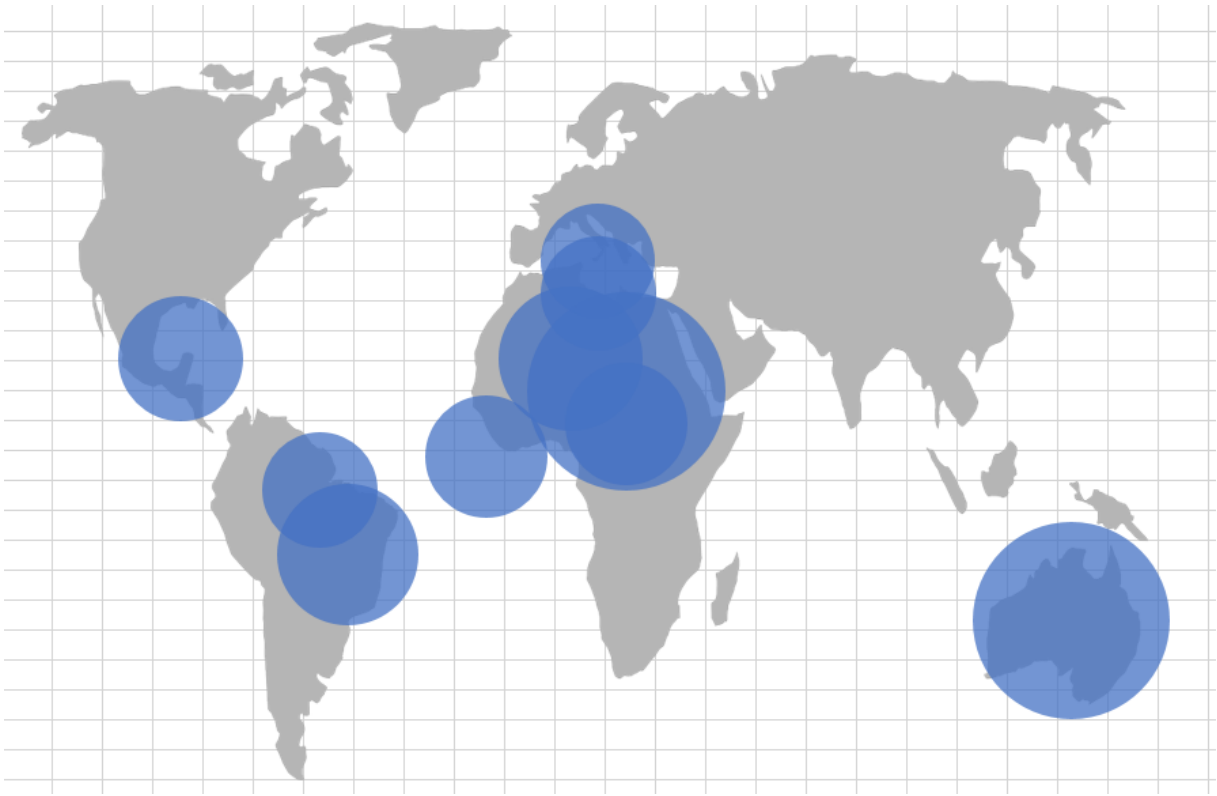
5. 데이터 분석

일별 데이터로 세분화 함으로써 더 많은 특징들을 추출 할 수 있었다.

5.1 사망률

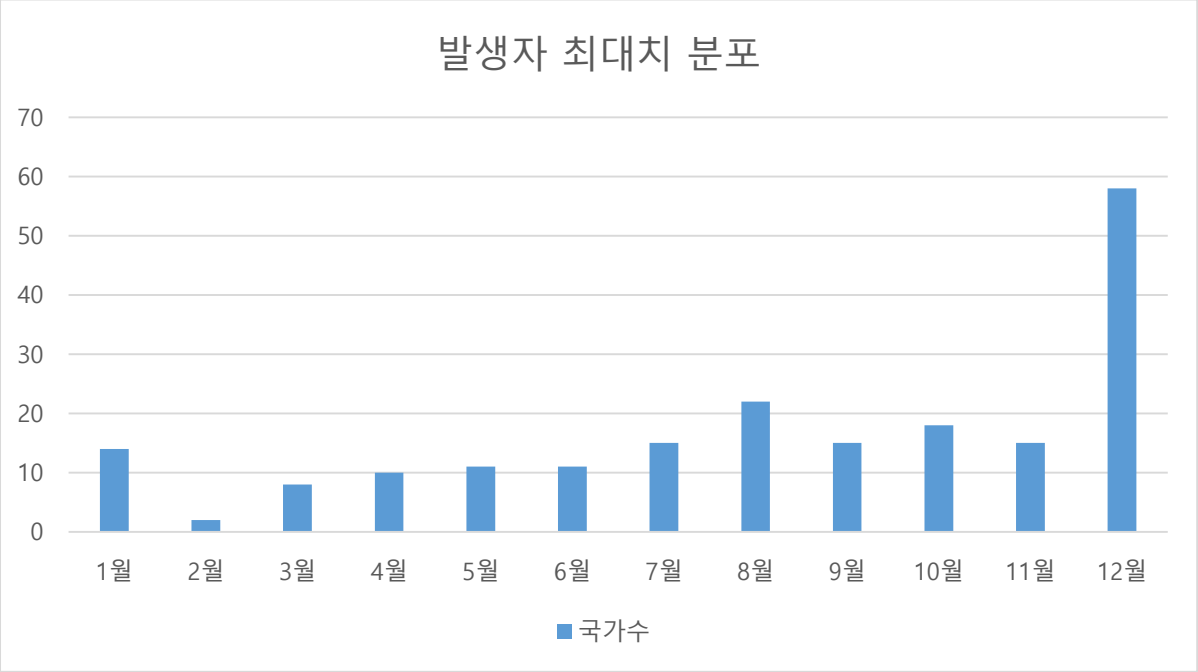
사망률 = (사망자 / 확진자) * 100				
	Country_Region	Confirmed.x	Deaths.x	DeathRate
1	Yemen	8027	1374	17.12
2	Vanuatu	6	1	16.67
3	Sudan	21031	1863	8.86
4	Peru	1281694	110329	8.61
5	Mexico	2553629	173621	6.8
6	Liberia	4660	301	6.46
7	Somalia	18818	1203	6.39
8	Ecuador	337313	19647	5.82
9	Egypt	247513	14121	5.71
10	Syria	38844	2186	5.63

-> 사망률 상위 10 국가



5.2 발생자 최대치 분포

	1 월	2 월	3 월	4 월	5 월	6 월	7 월	8 월	9 월	10 월	11 월	12 월
국가수	14	2	8	10	11	11	15	22	15	18	15	58



5.3 확진자, 사망자간 상관관계

```
> cor.test(test_sum_DF$Confirmed.x,test_sum_DF$Deaths.x,method = 'pearson')
```

```
Pearson's product-moment correlation

data:  test_sum_DF$Confirmed.x and test_sum_DF$Deaths.x
t = 30.098, df = 197, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8778711 0.9283625
sample estimates:
      cor
0.9062992
```

-> 95% 신뢰수준:[0.8778711 0.9283625], 상관관계 0.9062 로 강한 양의 상관관계를 보인다.

6. 결론

4.2, 4.3 에 의해 같은 데이터를 다른 방식으로 구한 두 샘플(Sampe1, Sample2)간 차이가 없다는 것을 알 수 있다. 4.4 에 의해 확진자가 증가하면 사망자도 증가하는 양의 상관 관계를 찾을 수 있었다.

데이터를 통해서 사망률을 추출 할 수 있었는데 상위 국가들이 대부분 중동, 아프리카, 남미에 집중되어 있는 것을 볼 수 있다. 상대적으로 아시아, 북미에 비해 중동, 아프리카, 남미가 의료시설이나 환경이 좋지 않다고 볼 수 있겠다.

또한 각 국가별 확진자수가 최대치 달을 추출해보니 12 월이 가장 많은 것을 확인 할 수 있었다. 2021 년 12 월에 세계적으로 대유행이 시작된다고 판단 할 수 있다.

일별 데이터의 결측치를 확인하기 위하여 결측치 유무를 확인해 보았으며 이 과정에서 결측치 및 0 데이터 이외의 음수를 발견할 수 있었다. 결측치 확인의 최초 목적은 데이터 자체가 비어있는(nan) 값과 0 인 값을 구분하기 위함이었으나, 실제 일일 신규 확진자 데이터 관측 결과 국가가 추가되면서 계산 상 발생한 nan 값 이외의 실제 결측치는 존재하지 않았다.

신규 확진자 값은 음수가 될 수 없으므로 이를 이상치로 처리하는 것이 옳다고 판단하였으며, 0 으로 전처리하여 이상치를 제외한 값으로 새로운 누적 데이터를 도출하였다. 음수 값이 실제 존재하는 데이터가 맞는지 확인하기 위하여 누적 데이터 간의 차이와 음수가 포함된 일별 데이터의 합을 비교 검정해 본 결과 동일함을 확인하였다. 이상치가 입력된 날짜 및 국가의 데이터는 데이터의 오입력 사유를 별첨된 데이터 수정 리스트에서 사유를 확인하기 위한 데이터셋으로 활용할 수 있을 것이다.

7. 별첨

7.1 소스코드

```
library(dplyr)
library(stringr)
library(corrgram)
library(ggplot2)


###빼기 함수(NA 값 처리후 뺄셈)###
sub_ <- function(x,y){
  return(ifelse(is.na(x),0,x)-ifelse(is.na(y),0,y))
}
###평균 함수(NA,이상 갯수 만큼 제외한 평균)###
avg_ <- function(x,y,z){
  return(round(x/(365-(y+z)),2))
}

###2020 년 데이터 로드###
rawData <- read.csv("covid19daily/12-31-2020.csv", header = T)
yesterdayDF <- rawData %>% group_by(Country_Region)%>%
summarise(Confirmed=sum(Confirmed), Deaths=sum(Deaths))
tempDF <- yesterdayDF

###TEST 데이터####
testData <- read.csv("covid19daily/12-31-2021.csv", header = T)
testData <- testData %>% group_by(Country_Region)%>%
summarise(Confirmed=sum(Confirmed), Deaths=sum(Deaths))
testData <- merge(yesterdayDF,testData, by ='Country_Region', all = TRUE)
test_sub_DF <- testData%>% group_by(Country_Region) %>%
summarise(Confirmed=sub_(Confirmed.y,Confirmed.x),
Deaths=sub_(Deaths.y,Deaths.x))
test_sub_DF$T_Mean_C <- round(test_sub_DF$Confirmed/365,2)
test_sub_DF$T_Mean_D <- round(test_sub_DF$Deaths/365,2)
test_sub_DF

###2021 년 파일 리스트 생성###
FN <- list.files(path = 'covid19daily') # 파일 이름
fileName<- unlist(str_extract_all(FN,"[0-9,-]*-2021"))

###파일 불러와서 일별로 계산후 데이터프레임에 합치기###
for(i in fileName){
  todayDF <- read.csv(paste0("covid19daily/",i,".csv"),header=T)
  todayDF <- todayDF %>% group_by(Country_Region) %>%
summarise(Confirmed=sum(Confirmed), Deaths=sum(Deaths))
  dailyDF <- merge(yesterdayDF,todayDF, by ='Country_Region', all = TRUE)
  dailyDF <- dailyDF%>% group_by(Country_Region) %>%
summarise(Confirmed=sub_(Confirmed.y,Confirmed.x),
Deaths=sub_(Deaths.y,Deaths.x))
  date_month <- unlist(str_extract_all(i,"[0-9]{1,2}-[0-9]{1,2}"))
  names(dailyDF) <-
c("Country_Region",paste0(date_month,"_확진자"),paste0(date_month,"_사망자"))
```

```
  resultDF <- merge(tempDF,dailyDF, by ='Country_Region', all = TRUE)
  yesterdayDF <- todayDF
  tempDF <- resultDF
}
rDF <- resultDF[,c(-2,-3)]
write.csv(rDF, '일일데이터.csv')
#####음수값,결측값 갯수 세고 국가 체크##########
confirm <- rDF[,c(1,seq(2, ncol(rDF),2))]
death <- rDF[,c(1,seq(3, ncol(rDF),2))]

missC <- confirm
missC <- apply(confirm[,c(2:ncol(confirm))]<0,1,sum,na.rm=T)
confirm[missC!=0,1]

missD <- death
missD <- apply(death[,c(2:ncol(death))]<0,1,sum,na.rm=T)
death[missD!=0,1]

nullC <- apply(is.na(confirm),1,sum)
nullD <- apply(is.na(death),1,sum)
confirm[nullC!=0,1]
death[nullD!=0,1]


####음수값, 결측치 0으로 대체 #######
rDF <- replace(rDF[,c(1,2:ncol(rDF))],rDF[,c(1,2:ncol(rDF))]<0,0)
rDF[is.na(rDF)] <- 0

rDF
hist(rDF$)

# (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.(누적데이터인 경우 누적데이터로 해당 결과를 제시하고, 일별 데이터
# 를 산출하여 총합과 일평균값을 산출하여 결과 비교)

finalDF <- rDF
len <- ncol(finalDF)
finalDF$Confirmed <- apply(finalDF[,seq(2, len,2)],1,sum,na.rm=T)
finalDF$Deaths <- apply(finalDF[,seq(3, len,2)],1,sum,na.rm=T)
finalDF <- subset(finalDF,select=c(Country_Region,Confirmed,Deaths))
finalDF$MeanConfirmed <- avg_(finalDF$Confirmed,missC,nullC)
finalDF$MeanDeaths <- avg_(finalDF$Deaths,missD,nullD)
finalDF$missC <- missC
finalDF$missD <- missD
finalDF$nullC <- nullC
finalDF$nullD <- nullD
finalDF <- merge(finalDF,test_sub_DF, by ='Country_Region', all = TRUE)
print(finalDF)

write.csv(finalDF, '최종 DataFrame.csv')
# (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
```

```r
print("확진자 이상값 존재 국가 리스트");confirm[missC!=0,1]
print("사망자 이상값 존재 국가 리스트");death[missD!=0,1]
print("확진자 결측값 존재 국가 리스트");confirm[nullC!=0,1]
print("사망자 결측값 존재 국가 리스트");death[nullD!=0,1]



# (3) 2021 년 1 년동안 코로나 총 발생자수, 총 사망자수, 일평균 발생자수, 일평균
# 사망자 수를 기준으로 가장 많은 20 개 국가를 내림차순으로 정렬하고 총 발생
# 자수, 총 사망자수, 일평균 발생자수, 일평균 사망자 수를 리포트 하시오. (4 가
# 지 기준 각각 sorting)

totConfirmed20 <-
finalDF[order(finalDF$Confirmed.x,finalDF$Deaths.x,finalDF$MeanConfirmed,finalD
F$MeanDeaths,decreasing = TRUE),]
rownames(totConfirmed20) <- 1 : length(rownames(totConfirmed20))
print(head(totConfirmed20,n=20))

meanConfirmed20 <-
finalDF[order(finalDF$MeanConfirmed,finalDF$Confirmed.x,finalDF$Deaths.x,finalD
F$MeanDeaths,decreasing = TRUE),]
rownames(meanConfirmed20) <- 1 : length(rownames(meanConfirmed20))
print(head(meanConfirmed20,n=20))

totDeaths20 <-
finalDF[order(finalDF$Deaths.x,finalDF$Confirmed.x,finalDF$MeanConfirmed,finalD
F$MeanDeaths,decreasing = TRUE),]
rownames(totDeaths20) <- 1 : length(rownames(totDeaths20))
print(head(totDeaths20,n=20))

meanDeaths20 <-
finalDF[order(finalDF$MeanDeaths,finalDF$Confirmed.x,finalDF$Deaths.x,finalDF$M
eanConfirmed,decreasing = TRUE),]
rownames(meanDeaths20) <- 1 : length(rownames(meanDeaths20))
print(head(meanDeaths20,n=20))

write.csv(totConfirmed20, '확진자정렬.csv')
write.csv(meanConfirmed20, '사망자정렬.csv')
write.csv(totDeaths20, '평균확진자정렬.csv')
write.csv(meanDeaths20, '평균사망자정렬.csv')

# (4) 2021 년 1 년동안 대한민국에서 발생한 총 코로나 발생자수와 총 사망자 수
# 와 일평균 발생자수와 일평균 사망자 수를 리포트 하시오.

KOR <- finalDF%>%subset(finalDF$Country=="Korea, South")


write.csv(KOR, '대한민국.csv')
```

7.2 Readme 원본

Field description

* FIPS: US only. Federal Information Processing Standards code that uniquely identifies counties within the USA.

* Admin2: County name. US only.

* Province_State: Province, state or dependency name.

* Country_Region: Country, region or sovereignty name. The names of locations included on the Website correspond with the official designations used by the U.S. Department of State.

* Last Update: MM/DD/YYYY HH:mm:ss (24 hour format, in UTC).

* Lat and Long_: Dot locations on the dashboard. All points (except for Australia) shown on the map are based on geographic centroids, and are not representative of a specific address, building or any location at a spatial scale finer than a province/state. Australian dots are located at the centroid of the largest city in each state.

* Confirmed: Counts include confirmed and probable (where reported).

* Deaths: Counts include confirmed and probable (where reported).

* Recovered: Recovered cases are estimates based on local media reports, and state and local reporting when available, and therefore may be substantially lower than the true number. US state-level recovered cases are from [COVID Tracking Project](https://covidtracking.com/). We stopped to maintain the recovered cases (see [Issue #3464](https://github.com/CSSEGISandData/COVID-19/issues/3464) and [Issue #4465](https://github.com/CSSEGISandData/COVID-19/issues/4465)).

* Active: Active cases = total cases - total recovered - total deaths. This value is for reference only after we stopped to report the recovered cases (see [Issue #4465](https://github.com/CSSEGISandData/COVID-19/issues/4465))

* Incident_Rate: Incidence Rate = cases per 100,000 persons.

* Case_Fatality_Ratio (%): Case-Fatality Ratio (%) = Number recorded deaths / Number cases.

* All cases, deaths, and recoveries reported are based on the date of initial report. Exceptions to this are noted in the "Data Modification" and "Retrospective reporting of (probable) cases and deaths" subsections below.

Data modification records

This section will contain any modifications to our datasets as well as the reason for the change. If the error results from an issue on our collection of the data, the error will be listed in the errata.csv in the csse_covid19_time_series folder. If the error results due to a change from the source, the change and reasoning will be listed below.