부록 1. 데이터 셋 만들기

### R 코드

* 필요 패키지 설치 및 로딩

if(!require(pacman)) {  
 install.packages('pacman')  
 library(pacman)  
}  
  
pacman::p\_load('tidyverse', 'readxl', 'readr', 'plotly', 'lubridate')

* 코로나 19 데이터 셋

# 1. covid19 원본 데이터 셋 로딩  
# covid19 데이터 로딩(파일을 다운로드 받은 경우)  
df\_covid19 <- read\_csv(file = "데이터저장경로/owid-covid-data.csv",  
 col\_types = cols(Date = col\_date(format = "%Y-%m-%d")  
 )  
 )  
# covid19 데이터 로딩(온라인에서 바로 로딩할 경우)  
df\_covid19 <- read\_csv(file = "https://covid.ourworldindata.org/data/owid-covid-data.csv",  
 col\_types = cols(Date = col\_date(format = "%Y-%m-%d")  
 )  
 )  
  
  
## 2. 전체 데이터셋 중 최근 100일간의 데이터를 필터링한 df\_covid19\_100 생성  
df\_covid19\_100 <- df\_covid19 |>   
 ## 한국 데이터와 각 대륙별 데이터만을 필터링  
 filter(iso\_code %in% c('KOR', 'OWID\_ASI', 'OWID\_EUR', 'OWID\_OCE', 'OWID\_NAM', 'OWID\_SAM', 'OWID\_AFR')) |>  
 ## 읽은 데이터의 마지막 데이터에서 100일전 데이터까지 필터링  
 filter(date >= max(date) - 100) |>  
 ## 국가명을 한글로 변환  
 mutate(location = case\_when(  
 location == 'South Korea' ~ '한국',   
 location == 'Asia' ~ '아시아',   
 location == 'Europe' ~ '유럽',   
 location == 'Oceania' ~ '오세아니아',   
 location == 'North America' ~ '북미',   
 location == 'South America' ~ '남미',   
 location == 'Africa' ~ '아프리카')) |>  
 ## 국가 이름의 순서를 설정   
 mutate(location = fct\_relevel(location, '한국', '아시아', '유럽', '북미', '남미', '아프리카', '오세아니아')) |>  
 ## 날짜로 정렬  
 arrange(date)  
  
  
## 3. df\_covid19\_100을 한국과 각 대륙별열로 배치한 넓은 형태의 데이터프레임으로 변환  
df\_covid19\_100\_wide <- df\_covid19\_100 |>  
 ## 날짜, 국가명, 확진자와, 백신접종완료자 데이터만 선택  
 select(date, location, new\_cases, people\_fully\_vaccinated\_per\_hundred) |>  
 ## 열 이름을 적절히 변경  
 rename('date' = 'date', '확진자' = 'new\_cases', '백신접종완료자' = 'people\_fully\_vaccinated\_per\_hundred') |>  
 ## 넓은 형태의 데이터로 변환  
 pivot\_wider(id\_cols = date, names\_from = location,   
 values\_from = c('확진자', '백신접종완료자')) |>  
 ## 날짜로 정렬  
 arrange(date)  
  
## 4. covid19 데이터를 국가별로 요약한 df\_covid19\_stat 생성  
df\_covid19\_stat <- df\_covid19 |>   
 group\_by(iso\_code, continent, location) |>  
 summarise(인구수 = max(population, na.rm = T),   
 인당GDP = max(gdp\_per\_capita, na.rm = T),  
 전체확진자수 = sum(new\_cases, na.rm = T),  
 전체사망자수 = sum(new\_deaths, na.rm = T),   
 십만명당중환자실 = last(icu\_patients\_per\_million),  
 재생산지수 = last(reproduction\_rate),  
 봉쇄지수 = max(stringency\_index),   
 전체검사자수 = max(total\_tests, na.rm = T),   
 신규검사자수 = sum(new\_tests, na.rm = T),  
 전체백신접종자수 = max(total\_vaccinations, na.rm = T),  
 백신접종자완료자수 = max(people\_fully\_vaccinated, na.rm = T),  
 부스터접종자수 = max(total\_boosters, na.rm = T),  
 인구백명당백신접종완료률 = max(people\_fully\_vaccinated\_per\_hundred, na.rm = T),  
 인구백명당부스터접종자수 = max(total\_boosters\_per\_hundred, na.rm = T)  
 ) |>   
 ungroup() |>  
 mutate(십만명당사망자수 = round(전체사망자수 / 인구수 \*100000, 5),  
 백신접종완료률 = 백신접종자완료자수 / 인구수)

* 취업률 데이터 셋

## R 코드  
  
df\_취업률 <- read\_excel('파일 저장 경로/2020년 학과별 고등교육기관 취업통계.xlsx',   
 ## '학과별' 시트의 데이터를 불러오는데,  
 sheet = '학과별',  
 ## 앞의 13행을 제외하고  
 skip = 13,   
 ## 첫번째 행은 열 이름으로 설정  
 col\_names = TRUE,   
 ## 열의 타입을 설정, 처음 9개는 문자형으로 다음 79개는 수치형으로 설정  
 col\_types = c(rep('text', 9), rep('numeric', 79)))  
  
## df\_취업률에서 첫번째부터 9번째까지의 열과 '계'로 끝나는 열을 선택하여 다시 df\_취업률에 저장  
df\_취업률 <- df\_취업률 |>   
 select(1:9, ends\_with('계'), '입대자')  
  
## df\_취업률에서 졸업자가 500명 이하인 학과 2000개 샘플링  
df\_취업률\_2000 <- df\_취업률 |>   
 filter(졸업자\_계 < 500) |>  
 mutate(id = row\_number()) |>  
 filter(row\_number() %in% seq(from = 1, to = nrow(df\_취업률), by = 4))  
  
## 열 이름을 적절히 설정  
names(df\_취업률\_2000)[10:12] <- c('졸업자수', '취업률', '취업자수')

* 연도별 지역별 신입생 충원율 데이터 셋

df\_충원율\_botton <- read\_excel('파일 저장 경로/고등 주요 01-시도별 신입생 충원율(2010-2022)\_220825y.xlsx',   
 sheet = 'Sheet1',  
 skip = 7,   
 col\_names = FALSE,   
 col\_types = c(rep('text', 2), rep('numeric', 12)))  
  
df\_충원율\_botton <- df\_충원율\_botton |> dplyr::select(1, 2, 5)  
  
colnames(df\_충원율\_botton) <- c('연도', '지역', '신입생충원율')  
  
df\_충원율\_botton <- df\_충원율\_botton |> pivot\_wider(names\_from = '연도', values\_from = '신입생충원율')  
  
df\_충원율\_botton <- as.data.frame(df\_충원율\_botton)

* 여백 설정

margins\_R = list(t = 50, b = 25, l = 25, r = 25)

### python 코드

* 라이브러리 로딩

import pandas as pd  
from datetime import datetime, timedelta  
from pandas.api.types import CategoricalDtype  
from matplotlib import pyplot as plt  
import plotly.graph\_objects as go

* 코로나 19 데이터 셋

df\_covid19 = pd.read\_csv("파일 저장 경로/owid-covid-data\_221203.csv")  
  
df\_covid19['date'] = pd.to\_datetime(df\_covid19['date'], format="%Y-%m-%d")  
  
df\_covid19\_100 = df\_covid19[(df\_covid19['iso\_code'].isin(['KOR', 'OWID\_ASI', 'OWID\_EUR', 'OWID\_OCE', 'OWID\_NAM', 'OWID\_SAM', 'OWID\_AFR'])) & (df\_covid19['date'] >= (max(df\_covid19['date']) - timedelta(days = 100)))]  
  
  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'South Korea', "location"] = '한국'  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'Asia', "location"] = '아시아'  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'Europe', "location"] = '유럽'  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'Oceania', "location"] = '오세아니아'  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'North America', "location"] = '북미'  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'South America', "location"] = '남미'  
df\_covid19\_100.loc[df\_covid19\_100['location'] == 'Africa', "location"] = '아프리카'  
  
ord = CategoricalDtype(categories = ['한국', '아시아', '유럽', '북미', '남미', '아프리카', '오세아니아'], ordered = True)  
  
df\_covid19\_100['location'] = df\_covid19\_100['location'].astype(ord)  
  
df\_covid19\_100 = df\_covid19\_100.sort\_values(by = 'date')  
  
df\_covid19\_100\_wide = df\_covid19\_100.loc[:,['date', 'location', 'new\_cases', 'people\_fully\_vaccinated\_per\_hundred']].rename(columns={'new\_cases':'확진자', 'people\_fully\_vaccinated\_per\_hundred':'백신접종완료자'})  
  
df\_covid19\_100\_wide = df\_covid19\_100\_wide.pivot(index='date', columns='location', values=['확진자', '백신접종완료자']).sort\_values(by = 'date')  
  
df\_covid19\_100\_wide.columns = ['확진자\_한국', '확진자\_아시아', '확진자\_유럽', '확진자\_북미', '확진자\_남미', '확진자\_아프리카','확진자\_오세아니아',  
 '백신접종완료자\_한국', '백신접종완료자\_아시아', '백신접종완료자\_유럽', '백신접종완료자\_북미', '백신접종완료자\_남미', '백신접종완료자\_아프리카','백신접종완료자\_오세아니아']  
   
df\_covid19\_stat = df\_covid19.groupby(['iso\_code', 'continent', 'location'], dropna=False).agg(  
 인구수 = ('population', 'max'),  
 전체확진자수 = ('new\_cases', 'sum'),  
 전체사망자수 = ('new\_deaths', 'sum'),   
 백신접종자완료자수 = ('people\_fully\_vaccinated', 'max'),  
 인구백명당백신접종완료율 = ('people\_fully\_vaccinated\_per\_hundred', 'max'),  
 인구백명당부스터접종자수 = ('total\_boosters\_per\_hundred', 'max')  
).reset\_index()  
  
df\_covid19\_stat['십만명당사망자수'] = round(df\_covid19\_stat['전체사망자수'] / df\_covid19\_stat['인구수'] \*100000, 5)  
  
df\_covid19\_stat['백신접종완료율'] = df\_covid19\_stat['백신접종자완료자수'] / df\_covid19\_stat['인구수']

* 취업률 데이터 셋

df\_취업률 = pd.read\_excel("d:/R/data/2020년 학과별 고등교육기관 취업통계.xlsx",   
 sheet\_name = '학과별',  
 skiprows=(13),   
 header = 0)  
  
df\_취업률 = pd.concat([df\_취업률.iloc[:, 0:8],   
 df\_취업률.loc[:, df\_취업률.columns.str.endswith('계')],   
 df\_취업률.loc[:, '입대자']],   
 axis = 1  
 )  
  
df\_취업률\_2000 = df\_취업률.loc[(df\_취업률['졸업자\_계'] < 500)]  
  
df\_취업률\_2000 = df\_취업률\_2000.iloc[range(0, len(df\_취업률\_2000.index) , 4)]  
  
df\_취업률\_2000 = df\_취업률\_2000.rename(columns = {'졸업자\_계':'졸업자수', '취업률\_계':'취업률', '취업자\_합계\_계':'취업자수'})

* 연도별 지역별 신입생 충원율 데이터 셋

df\_충원율\_control = pd.read\_excel("D:/R/git/datavisualization/plotly/RnPy/chap6/고등 주요 01-시도별 신입생 충원율(2010-2022)\_220825y.xlsx",   
 sheet\_name = 'Sheet1',  
 skiprows=(6),   
 header = 0)  
df\_충원율\_control = df\_충원율\_control.iloc[:, [0, 1, 4]]  
df\_충원율\_control.columns = ('연도', '지역', '신입생충원율')  
df\_충원율\_control = df\_충원율\_control.pivot(index = '지역', columns = '연도', values = '신입생충원율').reset\_index()

* 여백 설정

margins\_P = dict(t = 50, b = 25, l = 25, r = 25)