데이터 전처리

데이터 결합을 위하여, 상권 및 행정동 위치 코드 가져오기

```
library(XML)
loc1 <- xmlToDataFrame(doc = 'http://openapi.seoul.go.kr:8088/6a596b4c4462616235334e454e5a52/xml/TbgisTrdarRelm/1/1000/')
loc2 <- xmlToDataFrame(doc = 'http://openapi.seoul.go.kr:8088/6a596b4c4462616235334e454e5a52/xml/TbgisTrdarRelm/1001/1496/')
var <- 1:2
loc1 <- loc1 %>% slice(-var)
loc2 <- loc2 %>% slice(-var)
sangkwon_loc <- rbind(loc1,loc2)
sangkwon_loc <- sangkwon_loc[,c(6,11)]

#행정동 코드-행정구 파일 읽기
guess_encoding(file = '행정동코드_매핑정보_20200325.xlsx')
sangkwon_gu <- read.xlsx(xlsxFile = '행정동코드_매핑정보_20200325.xlsx',sheet = 1)
sangkwon_gu <- sangkwon_gu %>% slice(-1)
sangkwon_gu <- sangkwon_gu[,c(2,4)]
```

우리마을 상권분석-추정매출 데이터 결합 및 정리

```
library(tidyverse)

setwd('C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터/원본데이터/')
getwd()
list.files()

guess_encoding("서울시 우리마을가게 상권분석서비스(상권-추정매출)_2019.csv")
guess_encoding("서울시 우리마을가게 상권분석서비스(상권-추정매출)_2020.csv")
data1 <- read.csv("서울시 우리마을가게 상권분석서비스(상권-추정매출)_2019.csv")
data2 <- read.csv("서울시 우리마을가게 상권분석서비스(상권-추정매출)_2020.csv")

#상권코드명 인덱스 일치시키기
data2[data2$상권_코드_명=="종로?청계 관광특구",]$상권_코드_명 <- "종로·청계 관광특구"

#2019,2020년도 데이터셋 결합
smallbz_sales <- rbind(data1,data2)

#매출액 컬럼 생성 - 월화수목/금토일 & 0614/1421/2106
smallbz_sales <- smallbz_sales %>%
  mutate(매출_월화수목 = 월요일_매출_금액+화요일_매출_금액+수요일_매출_금액+목요일_매출_금액,
                매출_금토일 = 금요일_매출_금액+토요일_매출_금액+일요일_매출_금액,
                매출_0614 = 시간대_06.11_매출_금액+시간대_11.14_매출_금액,
                매출_1421 = 시간대_14.17_매출_금액+시간대_17.21_매출_금액,
                매출_2106 = 시간대_21.24_매출_금액+시간대_00.06_매출_금액)
vars <- c(1,2,5,8,9,80,81,82,83,84,85)
smallbz_sales <- smallbz_sales[,vars]
colnames(smallbz_sales)[c(1,2,4,5)] <- c("년도","분기","소분류","매출총액")

#매출데이터 행정구 추가
smallbz_total <- merge(x = smallbz_sales, y = sangkwon_loc,
                       by.x = '상권_코드', by.y = 'TRDAR_CD', all.x=T)
smallbz_total <- merge(x = smallbz_total, y = sangkwon_gu,
                       by.x = 'ADSTRD_CD', by.y = '행자부행정동코드', all.x=T)
smallbz_total <- rename(smallbz_total,c('행정구역' = '시군구명'))
```

코로나 데이터 가져온 후 매출 데이터와 결합

```
library(jsonlite) #Fromjson 내장 패키지
library(reshape) #rename 내장 패키지
library(lubridate)

#URL : https://news.seoul.go.kr/api/27/getCorona19Status/get_status_ajax.php?draw=7&start=100&length=100

url <- "https://news.seoul.go.kr/api/27/getCorona19Status/get_status_ajax_pre.php?draw=1"
url <- paste0(url,'&start=0&length=100') #url 지정

#데이터를 저장할 dataframe 생성
covid19<- data.frame()

##1~10000번
```

```
i <- 0
repeat{
  url <- paste0("https://news.seoul.go.kr/api/27/getCorona19Status/get_status_ajax_pre.php?draw=",(i+1))
  url <- paste0(url,'&start=',i*100,'&length=',100)
  data_json <- fromJSON(url)
  data_totalnum <- data_json$recordsTotal
  cat("1~10000번 확진자 크롤링",floor(((i)*10000)/data_totalnum),"% 진행중...\n")
  covid19<- rbind(covid19,as.data.frame(data_json$data))
  if((i*100)>=data_totalnum) break
  i <- i+1
}
#10000번 이후
i <- 0
repeat{
  url <- paste0("https://news.seoul.go.kr/api/27/getCorona19Status/get_status_ajax.php?draw=",(i+1))
  url <- paste0(url,'&start=',i*100,'&length=',100)
  data_json <- fromJSON(url)
  data_totalnum <- data_json$recordsTotal
  cat("10000번 이후 확진자 크롤링",floor(((i)*10000)/data_totalnum),"% 진행중...\n")
  covid19<- rbind(covid19,as.data.frame(data_json$data))
  if((i*100)>=data_totalnum) break
  i <- i+1
}
#컬럼명 재지정
covid19<- rename(covid19,c('V1'='연번','V2'='환자','V3'='확진일','V4'='행정구역',
                                          'V5'='여행력','V6'='접촉력','V7'='퇴원현황'))

#연번 값 변경
num <- c()
for(i in 1:nrow(covid19)){
  string = str_split(string = covid19$연번[i], pattern = "no'>")[[1]][2]
  string = str_split(string, pattern = "</")[[1]][1]
  num = c(num, string)
}
covid19$연번 <- num

#분기 년도 추가
covid19$년도 <- year(ymd(covid19$확진일))
covid19$분기 <- quarter(ymd(covid19$확진일))

#행정구역 & 년도 & 분기별 확진자 수로 변경(19.1~20.3분기)
covid19 <- covid19%>%
  filter( !행정구역 %in% c('기타', '','타시도') &
            년도 %in% c(2019,2020) &
            !(년도 == 2020 & 분기 == 4)) %>%
  group_by(년도,분기,행정구역) %>%
  summarise(확진자수 = n())

#데이터 결합
smallbz_total <- merge(x = smallbz_total, y = covid19,
                       by = c('년도','분기','행정구역'), all.x=T)

#19년도 코로나 확진자 수 NA값을 0으로 대체
smallbz_total[,14] <- ifelse(is.na(smallbz_total$확진자수)==T,0,smallbz_total$확진자수)
```

유동인구 데이터 가져온 후 매출 데이터와 결합

```
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터/원본데이터")
list.files()
guess_encoding("서울시 우리마을가게 상권분석서비스(상권-추정유동인구).csv")
smallbz_pop <- read.csv("서울시 우리마을가게 상권분석서비스(상권-추정유동인구).csv")

#유동인구 데이터에 행정구 기준 추가
smallbz_pop <- merge(x = smallbz_pop, y = sangkwon_loc,by.x = '상권_코드', by.y = 'TRDAR_CD', all.x=T)
smallbz_pop <- merge(x = smallbz_pop, y = sangkwon_gu,by.x = 'ADSTRD_CD', by.y = '행자부행정동코드', all.x=T)

#조작할 컬럼만 선택
vars <- c(3,4,534,8,17:29)
smallbz_pop <- smallbz_pop[,vars]

#컬럼명 및 컬럼 구성 변경
colnames(smallbz_pop)[1:4] <- c("년도","분기","행정구역","총_유동인구수")

smallbz_pop <- smallbz_pop %>%
  mutate(유동인구수_월화수목 = 월요일_유동인구_수+화요일_유동인구_수+수요일_유동인구_수+목요일_유동인구_수,
                   유동인구수_금토일 = 금요일_유동인구_수+토요일_유동인구_수+일요일_유동인구_수,
                   유동인구수_0614 = 시간대_2_유동인구_수+시간대_3_유동인구_수,
                   유동인구수_1421 = 시간대_4_유동인구_수+시간대_5_유동인구_수,
                   유동인구수_2106 = 시간대_6_유동인구_수+시간대_1_유동인구_수,) %>%
  select(년도,분기,행정구역,총_유동인구수,유동인구수_월화수목,유동인구수_금토일, 유동인구수_0614,유동인구수_1421,유동인구수_2106) %>%
  filter(년도 %in% c(2019,2020) & !(년도 == 2020 & 분기 == 4)) %>%
```

```
  group_by(년도,분기,행정구역) %>%
  summarise(총_유동인구수 = sum(총_유동인구수), 유동인구수_월화수목 = sum(유동인구수_월화수목),
                  유동인구수_금토일 = sum(유동인구수_금토일), 유동인구수_0614 = sum(유동인구수_0614),
                  유동인구수_1421 = sum(유동인구수_1421),유동인구수_2106 = sum(유동인구수_2106)) %>% as.data.frame()
smallbz_pop[,1:3] <- map_df(.x = smallbz_pop[,1:3],.f = as.factor)

#매출데이터와 유동인구 데이터 합치기
smallbz_total <- merge(x = smallbz_total, y = smallbz_pop,by = c("년도","분기","행정구역"), all.x=T)
```

우리마을 상권분석 데이터에서 신생기업 생존율 및 업종 구분 컬럼 가져온 후 매출 데이터와 결합

```
smallbz_total$소분류 <- str_replace_all(string = smallbz_total$소분류,
                                        pattern = "/",replacement = "&")

setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
load('우리마을상권분석.rda')
smallbz_data <- smallbz_data[,1:8]
colnames(smallbz_data)[6:8] <- c("생존률_1년차","생존률_3년차","생존률_5년차")
smallbz_data$소분류 <- str_replace_all(string = smallbz_data$소분류,
                                       pattern = '자전거및기타운송장비',
                                       replacement = '자전거 및 기타운송장비')
smallbz_total <- merge(x = smallbz_total, y = smallbz_data,
                       by = c('년도','분기','소분류','행정구역'), all.x=T)
```

상권-집객시설(숙박시설) 데이터 및 지하철 데이터 가져와서 매출 데이터와 합치기

```
#상권-숙박시설 데이터 합치기
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터/원본데이터")
list.files()
guess_encoding("서울시 우리마을가게 상권분석서비스(상권-집객시설).csv")
smallbz_faci <- read.csv("서울시 우리마을가게 상권분석서비스(상권-집객시설).csv")

##NA를 0으로 변환
for(i in 7:26){
  smallbz_faci[,i] <- ifelse(is.na(smallbz_faci[,i])==T,0,smallbz_faci[,i])
}
##행정구 추가
smallbz_faci <- merge(x = smallbz_faci, y = sangkwon_loc,
                      by.x = '상권_코드', by.y = 'TRDAR_CD', all.x=T)
smallbz_faci <- merge(x = smallbz_faci, y = sangkwon_gu,
                      by.x = 'ADSTRD_CD', by.y = '행자부행정동코드', all.x=T)

##필요 컬럼 선택
vars <- c(3,4,28,22)
smallbz_faci <- smallbz_faci[,vars]
colnames(smallbz_faci) <- c("년도","분기","행정구역","숙박시설_수")

smallbz_faci <- smallbz_faci %>%
  filter(is.na(행정구역)!=TRUE & 년도 %in% c(2019,2020) & (년도!=2020 & 분기 != 4)) %>%
 group_by(년도,분기,행정구역) %>% summarise(숙박시설_수 = sum(숙박시설_수))

smallbz_total <- merge(x = smallbz_total, y= smallbz_faci,
                       by = c("년도","분기","행정구역"),all.x =T)
smallbz_total$숙박시설_수 <- ifelse(is.na(smallbz_total$숙박시설_수)==T,
                                    0,smallbz_total$숙박시설_수)
#지하철 개수 추가
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
transportation <- readRDS("지하철역별_행정구 구분.rds")
smallbz_total <- merge(x = smallbz_total, y= transportation, by = c("행정구역"),all.x =T)
```

점포 개수와 매출 데이터 합치기

```
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터/원본데이터")
list.files()
data1 <- read.csv("서울시_우리마을가게_상권분석서비스(상권-점포)_2019년.csv")
data2 <- read.csv("서울시_우리마을가게_상권분석서비스(상권-점포)_2020년.csv")
jeompo <- rbind(data1,data2)

jeompo <- merge(x = jeompo, y = sangkwon_loc,by.x = '상권_코드',
                by.y = 'TRDAR_CD', all.x=T)
jeompo <- merge(x = jeompo, y = sangkwon_gu,by.x = 'ADSTRD_CD',
```

```
                by.y = '행자부행정동코드', all.x=T)

vars <- c(2,3,4,17,9,11)
jeompo <- jeompo[,vars]
colnames(jeompo)[2:6] <- c("년도","분기","행정구역","소분류","점포수")

jeompo$소분류 <- str_replace_all(jeompo$소분류,"/","&")
smallbz_total <- merge(x = smallbz_total, y= jeompo,
                       by=c("상권_코드","년도","분기","행정구역","소분류"),all.x=T)
smallbz_total$점포수 <- ifelse(is.na(smallbz_total$점포수)==T,0,smallbz_total$점포수)
```

데이터 정리

```
#데이터셋 컬럼 정리
vars <- c(2,3,1,21,4,7,8,9,10,11,12,13,15,16,17,18,19,20,14,22,23,24,25,26)
smallbz_total <- smallbz_total[,vars]

#범주형 및 연속형 데이터 정리
vars <- c(1,2,3,4,5)
smallbz_total[,vars] <- map_df(.x = smallbz_total[,vars],.f = as.factor)
smallbz_total[,-vars] <- map_df(.x = smallbz_total[,-vars],.f = as.numeric)

#train # test set 나누기
smallbz_total <- smallbz_total %>% mutate(년분기 = as.factor(paste0(년도,"_",분기)))
trainset <- smallbz_total %>% filter(년분기 !='2020_3')
testset <- smallbz_total %>% filter(년분기 =='2020_3')
```