

Algorithm_final

데이터 준비하기

- 5명 모두 유사한 방법을 활용(read.csv 이용 -> csv 파일 가져오기)

1. 약국 운영 정보 가져오기

```
df1 <- read.csv("seoul_pharmacy_final.csv")
```

컬럼명 재설정

```
new_colname <- c("약국명", "주소", "대표전화", "행정구", "시작(월)", "마감(월)", "시작(화)", "마감(화)", "시작(수)", "마감(수)", "시작(목)", "마감(목)", "시작(금)", "마감(금)", "시작(토)", "마감(토)", "시작(일)", "마감(일)", "시작(공휴일)", "마감(공휴일)", "일운영", "공휴일운영", "야간운영(월)", "야간운영(화)", "야간운영(수)", "야간운영(목)", "야간운영(금)", "야간운영(토)", "야간운영(일)", "야간운영(공휴일)")
```

```
colnames(df1) <- new_colname
```

View(df1) # 개발 시 데이터 확인을 위한 코드

2. 2022년 공휴일 정보 가져오기

파일 읽어오기

```
holiday_df <- read.csv("holiday_2022.csv")
```

2022년의 공휴일 날짜 정보를 벡터로 저장

```
holiday_2022 <- holiday_df$날짜
```

View(holiday_df) # 개발 시 데이터 확인을 위한 코드

입력

1. 지역: 사용자에게 '직접' 입력받는다.

```
# 강호은 -----
print("서울시 약국 운영 정보를 알려드립니다!")

## [1] "서울시 약국 운영 정보를 알려드립니다!"

# 차수빈 -----
place <- readline(prompt = "검색하고자 하는 지역을 행정구 단위로 입력해 주세요 : ")

## 검색하고자 하는 지역을 행정구 단위로 입력해 주세요 :

cat("입력하신 지역은 [",place,"] 입니다.")

## 입력하신 지역은 [   ] 입니다.
```

2. 현재 시간: 시스템 시간을 자동으로 읽어온다.

```
# 이수미 -----
curtime = Sys.time() # "2022-05-29 02:38:40 KST"
curtime = as.character(curtime) # "2022-05-29 02:38:40"
curtime = substr(curtime,12,16) # "02:38"
curtime = gsub(":", "", curtime) # "0238"
curtime = as.numeric(curtime) # 238

# 차수빈 -----
# 530 <= (현재 시각) <= 2929 가 되도록 형식 맞춰주기
curtime <- ifelse((curtime<530),curtime + 2400,curtime)
```

3. 현재 요일

- 시스템 시간을 자동으로 읽어온다.
- 530 <= (현재 시각) <= 2929 가 되도록 형식이 맞추어져 있음 => 0시부터 5시 29 까지는 전날에 해당한다.
- 예시> Sys.Date()의 경우 화요일 300 => 해당 프로그램에서는 월요일 2700 으로 처리됨

```

# 차수빈, 강호은, 곽지수(통합) -----
# 현재 날짜
curdate <- ifelse((curtime >= 2400) & (curtime <= 2929),
                  as.character(Sys.Date() - 1), as.character(Sys.Date()))

# 새벽 시간대 요일 처리

# 차수빈 -----
# 현재 요일
curday <- ifelse((curdate%in% holiday_2022),"공휴일",weekdays(as.Date(curdate))) # 공휴일 처리
curday <- ifelse(curday != "공휴일",substr(curday,1,1),"공휴일")
# 요일 저장 형식 처리

# 수정(강호은, 이수미 코드 참고)
# 공휴일 필터링 오류 수정(2022.06.01)

```

4. 제공받을 정보 옵션 선택

```

# 강호은 -----
cat("옵션에 따라 필요한 정보를 선택적으로 얻으실 수 있습니다.", "\n",
    "1) 현재 운영 중인 약국 정보", "\n",
    "2) 요일 별 야간 운영 약국 정보 (20:30 분 이후 운영 약국)", "\n",
    "3) 사용자 주변 약국 정보", "\n")

## 옵션에 따라 필요한 정보를 선택적으로 얻으실 수 있습니다.
## 1) 현재 운영 중인 약국 정보
## 2) 요일 별 야간 운영 약국 정보 (20:30 분 이후 운영 약국)
## 3) 사용자 주변 약국 정보

option = readline(prompt = "원하시는 옵션의 번호를 선택해주세요>> ")

## 원하시는 옵션의 번호를 선택해주세요>>

cat("선택하신 옵션은", option, "입니다.")

## 선택하신 옵션은 입니다.

```

출력

- 사용자가 선택한 옵션에 따라 적절한 정보를 제공한다.

1. 사용자가 입력한 행정구 내의 약국 가져오기

```
# 차수빈, 광지수, 최은빈 -----  
df_gu <- df1[c(which(df1$행정구 == place)),]  
# View(df_gu)
```

2. 운영 여부 필터링

- 요일이 일요일이나 공휴일인 경우 운영시간에 NA 가 존재한다. -> NA 와의 비교 연산의 결과는 항상 NA 이다. -> 이후 과정에서 정확한 필터링이 이루어지지 않을 위험성이 존재한다. -> 일요일/ 공휴일의 경우 운영 여부부터 확인해야 한다.
- 운영 여부를 확인하는 openToday() 함수 선언

```
# 차수빈 -----  
# 일요일 or 공휴일  
openToday <- function(df_gu, day){  
  if((day == "일") | (day == "공휴일")){  
    run_filter = paste0(day, "운영") # "일운영" 또는 "공휴일운영"  
    df_run <- df_gu[c(which(df_gu[,run_filter])),] # 해당 요일에 운영하는 약국  
  }  
  else{ # 요일: 월 ~ 토  
    df_run <- df_gu  
  }  
  return(df_run)  
}
```

3. 선택한 옵션에 따른 필터링

- 각각의 옵션에 대한 기능을 수행하는 함수를 만들어 두고, 최종 알고리즘에서 함수를 호출하여 사용한다.

옵션 1) 현재 운영중인 약국 찾기

```
# 강효은, 차수빈 -----
nowRunning <- function(df_gu,curday){
  df_today <- openToday(df_gu,curday) # 운영 여부 확인
  start = paste0("시작(",curday,")") # 약국 시작
  end = paste0("마감(",curday,")") # 약국 마감
  df_current <- df_today[c(which((df_today[,start] <= curtime) & (df_today[,end] >= curtime))), c("약국명","주소","대표전화",start,end)]
  View(df_current)
}
```

옵션 2) 요일별 야간 운영 정보

```
# 차수빈-----
openByDay <- function(df_gu){
  # 사용자에게 검색하고자 하는 요일을 입력받음

  cat("검색하고자 하는 요일을 입력해 주세요","\n",
      "(예: 월, 화, ... ,일, 공휴일)>> ")
  searchday <- readline()
  cat("입력하신 요일은",searchday,"입니다.")

  # 해당 요일에 여는 약국 찾기
  df_day <- openToday(df_gu,searchday)

  # 약국 운영시간 가져오기
  start = paste0("시작(",searchday,")") # 약국 시작
  end = paste0("마감(",searchday,")") # 약국 마감

  # 차수빈, 최은빈 -----
  # 해당 요일에 여는 약국들 중 야간 운영을 하는 약국을 찾기
  night_filter = paste0("야간운영(",searchday,")")
  df_night <- df_day[c(which(df_day[,night_filter])),]
  df_openday <- df_night[,c("약국명","주소","대표전화",start,end)]

  View(df_openday)
}
```

옵션 3) 근처에 있는 약국의 전체 정보 조회 * 사용자가 입력한 행정구 내의 약국을 보여주면 된다. * 이미 행정구에 대한 처리가 되어 있음 -> 처리된 data frame 을 보여주면 된다.

```
View(df_gu)
```

- switch 문으로 옵션 묶기 -> 메인 실행 코드

```
# 차수빈 -----
switch(option,
  "1" = {nowRunning(df_gu, curday)},
  "2" = {openByDay(df_gu)},
  "3" = {View(df_gu)},
  {print("옵션을 잘못 입력하셨습니다.")})

## [1] "옵션을 잘못 입력하셨습니다."
```