faster : appendix

Hyeongchan Bae

April 2022

Contents

1. psi

전문가경기서베이조사 원자료 → 보도자료용 부표 만들기

```
# psi

# 22.04.15 updated

# 0. what do you need

FOLDER = paste0('C:/Users/', Sys.info()['user'], '/Documents/GitHub/KIET_831/psi')

YEAR = 2022
MONTH = 3
FILE = '2022년3월_누적시계열_PSI_작성용(잠정).xlsx' # 최신 버전 누적시계열 파일

# 1. setting

setwd(FOLDER)

library(tidyverse) # 데이터 핸들링 패키지
library(readxl) # 엑셀 로드 패키지
```

```r
library(openxlsx) # 엑셀 출력 패키지

ROUND <- function(x, digits = 0) {

  posneg = sign(x)

  z = abs(x)*10^digits
  z = z + 0.5 + sqrt(.Machine$double.eps)
  z = trunc(z)
  z = z/10^digits

  z*posneg

} # 기본 function은 round(0.5) = 0 만들어서, ROUND(0.5) = 1 되는 사용자 함수 생성

# 2. data load

PSI_SIMPLE <- read_xlsx(FILE, '종합(단순평균)', na = c('', NA)) # 누적시계열의 1번 시트
PSI_WEIGHTED <- read_xlsx(FILE, '종합(가중평균)', na = c('', NA)) # 누적시계열의 2번 시트

PSI_SIMPLE_CUT <- PSI_SIMPLE %>% filter(연도 == YEAR, 월 == MONTH) # 필요 시점 데이터만 선택
PSI_WEIGHTED_CUT <- PSI_WEIGHTED %>% filter(연도 == YEAR, 월 == MONTH) # 필요 시점 데이터만 선택

# 3. make table

# 3-1. 업종별 패널 구성(p.2)

temp1 <- PSI_SIMPLE_CUT %>% # 2022년 2월 단순평균 데이터에서 다음 작업을 한 뒤 temp1 임시 객체로 저장
  select(구분, 응답수) %>% # 구분, 응답수 column만 골라서
  mutate(`응답자 구성비(%)` = ROUND(응답수 / PSI_SIMPLE_CUT$응답수[1] * 100, digits = 1)) # 소수점 한 자리로 구성비

APPENDIX.1 <- temp1[c(9, 8, 5, 7, 6, 11, 12, 10, 15, 14, 13, 16, 17, 18), ] %>% # 부표 양식대로 산업 순서 바꾸고
  select(1, 3) # 구분, 응답자 구성비만 저장

# 3-2. 기상도(p.3-4)
```

```r
temp2 <- PSI_SIMPLE_CUT %>% # 단순평균 데이터에서
  select(경기현황, 시장판매현황, 수출현황, 생산수준현황, 투자액현황, 채산성현황) %>% # 항목(변수)을 고른 뒤
  ROUND() %>% # 반올림 해주고
  mutate(구분 = PSI_SIMPLE_CUT$구분) %>% # 반올림 하느라 빼두었던 character variable 다시 넣고
  relocate(구분) # 순서도 맨 앞으로 조정

APPENDIX.2 <- temp2[c(1, 9, 8, 5, 7, 6, 11, 12, 10, 15, 14, 13, 16, 2, 3, 4), ] # 부표 양식 맞춰서 저장

temp3 <- PSI_SIMPLE_CUT %>%
  select(경기전망, 시장판매전망, 수출전망, 생산수준전망, 투자액전망, 채산성전망) %>%
  ROUND() %>%
  mutate(구분 = PSI_SIMPLE_CUT$구분) %>%
  relocate(구분)

APPENDIX.3 <- temp3[c(1, 9, 8, 5, 7, 6, 11, 12, 10, 15, 14, 13, 16, 2, 3, 4), ] # APPENDIX.2와 같은 방식으로 전망 파트

# 3-3. 제조업 및 부문별 통계(p.5-12)

temp4 <- PSI_SIMPLE_CUT %>% # 단순평균 데이터에서
  select(-c(1:4)) %>% # 1~4번째 column(연도, 월, 구분, 응답수) 제외하고, 즉 psi 값만 대상으로
  ROUND() %>% # 반올림 해주고
  mutate(구분 = PSI_SIMPLE_CUT$구분) %>% # 빼두었던 산업 구분 넣고
  relocate(구분) %>% # 맨 앞으로 옮겨준 다음
  filter(구분 %in% c('00_전체', '01_ICT', '02_장비', '03_소재')) %>% # 전체, ICT, 장비, 소재 행만 선택
  select(구분, starts_with(c('경기', '시장', '수출', '생산', '재고', '투자', '채산성', '제품단가'))) # 산업 구분과 부표 파트인 열만

temp4$재고수준현황 <- abs(temp4$재고수준현황 - 200) # 재고수준은 200 빼고 절대값 취해서 계산
temp4$재고수준전망 <- abs(temp4$재고수준전망 - 200)

temp5 <- PSI_WEIGHTED_CUT %>% # 가중평균 데이터에서도 비슷한 작업을 하는데
  select(-c(1:4)) %>%
  ROUND() %>%
  mutate(구분 = PSI_SIMPLE_CUT$구분) %>%
  relocate(구분) %>%
  filter(구분 %in% c('00_전체')) %>% # 제조업(전체)만 필요하니까 하나 선택
  select(구분, starts_with(c('경기', '시장', '수출', '생산', '재고', '투자', '채산성', '제품단가')))
```

```r
temp5$재고수준현황 <- abs(temp5$재고수준현황 - 200)
temp5$재고수준전망 <- abs(temp5$재고수준전망 - 200)

temp5[1, 1] <- '00_전체_가중' # 이름 같으면 헷갈리니까 가중지수는 따로 네이밍

temp6 <- rbind(temp4, temp5) # 단순평균, 가중평균 데이터에서 뽑아낸 4개, 1개 열을 묶고

temp7 <- temp6[c(1, 5, 2, 3, 4), ] %>% # 열 순서 조정한 다음
  t() %>% # 행열 뒤집고 (부표 스타일 보니, 가져다 붙이려면 뒤집는 게 좋아보임)
  as.data.frame() %>% # 데이터 프레임 형식으로 바꾸고 (matrix와 비슷)
  rownames_to_column() %>% # rowname으로 내려온 파트 이름을 아예 column 으로 만들고
  tibble() %>% # 조금 더 정제된 데이터 프레임 형식 전환
  filter(rowname != '구분') # 첫 줄은 거추장스러워서 제거

names(temp7) <- c('파트', '단순지수', '가중지수', 'ICT부문', '기계부문', '소재부문') # 네이밍 수정

APPENDIX.4 <- temp7 %>%
  mutate_at(vars(단순지수:소재부문), as.double) # 변환하면서 character로 입력된 숫자값을 number class로 바꾸고 저장

# 3-4. 세부 업종별 조사 통계(p.13-20)

temp8 <- PSI_SIMPLE_CUT %>% # 대충 비슷하니까 이하는 안써도 되겠지?
  select(-c(1:4)) %>%
  ROUND() %>%
  mutate(구분 = PSI_SIMPLE_CUT$구분) %>%
  relocate(구분) %>%
  filter(구분 %in% c('08_반도체', '07_디스플레이', '06_핸드폰', '05_가전', '10_자동차',
                   '11_조선', '09_기계', '14_화학', '13_철강', '12_섬유', '15_바이오헬스')) %>%
  select(구분, starts_with(c('경기', '시장', '수출', '생산', '재고', '투자', '채산성', '제품단가')))

temp8$재고수준현황 <- abs(temp8$재고수준현황 - 200)
temp8$재고수준전망 <- abs(temp8$재고수준전망 - 200)

temp9 <- temp8 %>%
  t() %>%
```

```
  as.data.frame() %>%
  rownames_to_column() %>%
  tibble()

temp10 <- temp9[-1, c(1, 5, 4, 3, 2, 7, 8, 6, 11, 10, 9, 12)]

names(temp10) <- c('파트', '08_반도체', '07_디스플레이', '06_핸드폰', '05_가전', '10_자동차',
                   '11_조선', '09_기계', '14_화학', '13_철강', '12_섬유', '15_바이오헬스')

APPENDIX.5 <- temp10 %>%
  mutate_at(vars(`08_반도체`:`15_바이오헬스`), as.double)# 세부 업종별 조사 통계

# 5. export

FOLDER.2 <- paste0(FOLDER, '/', YEAR, '-', MONTH, 'M')

dir.create(FOLDER.2)

setwd(FOLDER.2)

write.xlsx(x = list(APPENDIX.1, APPENDIX.2, APPENDIX.3, APPENDIX.4, APPENDIX.5),
           sheetName = c('업종별 패널 구성(p.2)', '현황 PSI(p.3)', '전망 PSI(p.4)',
                         '제조업 및 부문별 통계(p.5-12)',
                         '세부 업종별 조사 통계(p.13-20)'),
           file = paste0(YEAR, '-', MONTH, 'M PSI 보도자료용 부표.xlsx'))
```

2. kosis

통계청 Open API 활용 (미완성)

```
# kosis

# 22.04.15 updated

# 0. what do you need
```

```
FOLDER = paste0('C:/Users/', Sys.info()['user'], '/Documents/GitHub/KIET_831/kosis')

# 1. setting

setwd(FOLDER)

library(tidyverse)
library(jsonlite)
library(openxlsx)
library(lubridate)

BASE <- paste0('https://kosis.kr/openapi/statisticsData.do?method=getList', # 요청
               '&apiKey=시크릿시크릿', # 인증키
               '&format=json&jsonVD=Y', # 포맷 : JSON
               '&userStatsId=bhc5754/') # 방식 : 사용자가 기등록한 자료 로드

# 2. 광업제조업동향조사

# 2-1. 생산출하재고

CORE_생산출하재고 <- c('20220413144648', '20220413170357', '20220414092428') # 항목별 코드 (사용자 URL)

NUMBER_생산출하재고 <- 1:84

DATA_생산출하재고 <- tibble()

for (k in seq_along(CORE_생산출하재고)) {

 temp1 <- tibble()

 for (i in seq_along(NUMBER_생산출하재고)) {

   URL_생산출하재고 <- paste0(BASE,
                        '101/', # 통계청
                        'DT_1F01501/', # 시도/산업별 광공업생산지수(2015=100)
                        '2/', # 시계열
```

```
                    '1/', # 간격 : 1
                    CORE_생산출하재고[k], '_', NUMBER_생산출하재고[i], # URL 나열
                    '&prdSe=', 'M', # 주기 : Month
                    '&newEstPrdCnt=', '1') # 최근 1개 자료

  temp2 <- tryCatch(fromJSON(URL_생산출하재고) %>%
              tibble() %>%
              mutate(번호 = i) %>%
              select(번호, TBL_NM, ITM_ID, ITM_NM, PRD_DE, C1, C1_NM, C2, C2_NM, DT), # 필요 column 선택
              error = function(e) tibble(NULL)) # 오류(데이터 부재) 발생하면 스킵

  temp1 <- rbind(temp2, temp1) # stacking

 }

 temp3 <- temp1 %>%
  arrange(nchar(C2), C2) %>% # 분류값 순으로 정렬
  mutate_at(vars(DT), as.double) # 수치값 class 숫자로 변경

 DATA_생산출하재고 <- rbind(temp3, DATA_생산출하재고) # stacking

}

temp4 <- DATA_생산출하재고 %>% split(as.factor(.$ITM_ID)) # 항목별 data frame 분리

temp5 <- DATA_생산출하재고$ITM_NM %>% unique() # 항목 이름

write.xlsx(temp4, sheetName = temp5, paste0(today(tzone = 'Asia/Seoul'), ' 생산출하재고.xlsx'))
```

3. mzyoon

네이버 상세검색으로 뉴스 링크 수집하기

```
# mzyoon

# 22.04.14 updated
```

```r
# 1. setting

setwd('C:/Users/KIET/Documents/GitHub/KIET/mzyoon') # working directory

library(tidyverse) # 데이터 핸들링
library(rstudioapi) # RStudio Terminal 탭 사용
library(RSelenium) # chrome 자동화
library(rvest) # html 데이터 읽기
library(lubridate) # 날짜
library(openxlsx) # 엑셀 로드, 출력

# 2. selenium

TERM_COMMAND <- 'java -Dwebdriver.gecko.driver="geckodriver.exe" -jar selenium-server-standalone-4.0.0-alpha-1
terminalExecute(command = TERM_COMMAND)
REMDR = remoteDriver(port = 4445, browserName = 'chrome') # 어려운 멘트 다 됐고

REMDR$open() # 크롬 오픈

# 3. crawling

# 3-1. 검색창에서 간보기

TODAY <- today(tzone = 'Asia/Seoul') %>%
  str_replace_all('-', '.') # URL에서 오늘 날짜 설정해주기 위함

URL <- paste0('https://search.naver.com/search.naver',
              '?where=news',
              '&query=경제자유구역', # 검색어
              '&sm=tab_opt',
              '&sort=0',
              '&photo=0',
              '&field=0',
              '&pd=3',
              '&ds=', '1990.01.01', # 시작시점
```

```
          '&de=', TODAY, # 종료시점 = 오늘
          '&docid=',
          '&related=0',
          '&mynews=0',
          '&office_type=0',
          '&office_section_code=0',
          '&news_office_checked=',
          '&nso=so%3Ar%2Cp%3A',
          'from', '19900101', 'to', str_remove_all(TODAY, '₩₩.'), # 시작, 종료시점
          '&is_sug_officeid=0')

REMDR$navigate(URL) # 네이버 검색창으로 가서

temp1 <- REMDR$getPageSource() # 페이지 정보 싹 긁어온 다음

temp2 <- read_html(temp1[[1]]) %>% # html 데이터를 읽는데
 html_elements('div.sc_page_inner') %>% # 페이지 버튼 1~10 있는 파트에서
 html_elements('a.btn') %>% # 버튼 하나하나에 숨겨진
 html_attr('href') # 링크를 따온다

BUTTON_YOUNG <- paste0('https://search.naver.com/search.naver', temp2) # 링크 본체를 붙여서 저장

DATA <- tibble() # 데이터를 담을 빈 그릇

# 3-2. 초반 페이지 스캔

for (i in 1:6) { # 1-6 페이지

 REMDR$navigate(BUTTON_YOUNG[i]) # i번째 버튼을 눌러서

 temp.1 <- REMDR$getPageSource() # 역시 페이지 정보 스캔

 LINK <- read_html(temp.1[[1]]) %>%
  html_elements('div.group_news') %>% # 뉴스 섹션에서
  html_elements('div.news_area') %>% # 개별 뉴스 파트마다 있는
  html_elements('a.news_tit') %>% # 뉴스 타이틀(누르면 접속)을 대상으로
```

```
  html_attr('href') # 링크를 가져오자

 PRESS <- read_html(temp.1[[1]]) %>%
  html_elements('div.group_news') %>%
  html_elements('div.news_area') %>%
  html_elements('a.info.press') %>% # 언론 이름 파트
  html_text() # 그 중에서 텍스트로 된 것만 추출

 DATE <- read_html(temp.1[[1]]) %>%
  html_elements('div.group_news') %>%
  html_elements('div.news_area') %>%
  html_elements('span.info') %>% # 날짜 파트
  html_text()

 DATE_CHECKER <- as.logical(DATE %>% str_detect('~~.') +
                    DATE %>% str_detect('전')) # 간혹 'A면 13단' 처럼 날짜 아닌 텍스트도 잡혀서 거르기 위함

 temp.2 <- tibble(PRESS = PRESS, DATE = DATE[DATE_CHECKER], LINK = LINK) # 모은 데이터를 묶어서

 DATA <- rbind(temp.2, DATA) # 데이터 그릇에 차곡차곡 저장(반복하면 쌓임)

}

# 6페이지에 돌입하자, MAX가 10에서 11로 하나 늘어남
# 서수적으로 보면 6번째 버튼을 계속 클릭하면 되겠다고 판단

# 3-3. 이후 페이지 스캔

for (k in 1:24) { # 7-30 페이지

 temp3 <- REMDR$getPageSource()

 temp4 <- read_html(temp3[[1]]) %>%
  html_elements('div.sc_page_inner') %>%
  html_elements('a.btn') %>%
  html_attr('href')
```

```
  temp.1 <- REMDR$getPageSource()

  LINK <- read_html(temp.1[[1]]) %>%
    html_elements('div.group_news') %>%
    html_elements('div.news_area') %>%
    html_elements('a.news_tit') %>%
    html_attr('href')

  PRESS <- read_html(temp.1[[1]]) %>%
    html_elements('div.group_news') %>%
    html_elements('div.news_area') %>%
    html_elements('a.info.press') %>%
    html_text()

  DATE <- read_html(temp.1[[1]]) %>%
    html_elements('div.group_news') %>%
    html_elements('div.news_area') %>%
    html_elements('span.info') %>%
    html_text()

  DATE_CHECKER <- as.logical(DATE %>% str_detect('₩₩.') + DATE %>% str_detect('전'))

  temp.2 <- tibble(PRESS = PRESS, DATE = DATE[DATE_CHECKER], LINK = LINK)

  DATA <- rbind(temp.2, DATA) # 여기까진 다 비슷한데

  BUTTON_OLD <- paste0('https://search.naver.com/search.naver', temp4) # 버튼 정보를 새로 따와야 함

  REMDR$navigate(BUTTON_OLD[6]) # 매번 6번째 버튼을 눌러서(다음 페이지로) 이동

}

# 4. export

write.xlsx(DATA, 'mzyoon.xlsx') # 엑셀 파일로 저장
```

4. dobby

자동퇴근 프로그램

```
# dobby

# 22.04.19 updated

# 0. what do you need

FOLDER = paste0('C:/Users/', Sys.info()['user'], '/Documents/GitHub/KIET/dobby')

WHEN <- '17:45:00'
ID <- '21032'
PW <- '21032961228'
SHUTDOWN <- 'YES'

# 1. setting

setwd(FOLDER)

library(tidyverse)
library(rstudioapi)
library(RSelenium)
library(rvest)
library(lubridate)

# 2. selenium

TERM_COMMAND <- 'java -Dwebdriver.gecko.driver="geckodriver.exe" -jar selenium-server-standalone-4.0.0-alpha-
terminalExecute(command = TERM_COMMAND)
REMDR = remoteDriver(port = 4445, browserName = 'chrome')

REMDR$open()

# 3. commute check
```

```
for (i in 1:100) {

 # 3-1. time to go

 NOW <- now(tzone = 'Asia/Seoul')
 TODAY <- today(tzone = 'Asia/Seoul')
 GOHOME <- paste0(TODAY, ' ', WHEN) %>% ymd_hms(tz = 'Asia/Seoul')

 if(GOHOME > NOW){

  REMDR$navigate('https://www.kiet.re.kr/kiet_web/main/')

  print(paste0('자동퇴근 프로그램이 작동 중입니다.', ' (현재 : ', i, ' 회차)'))

  print('퇴근까지 남은 시간을 알려드립니다.')

  print(GOHOME - NOW)

 }

 if(GOHOME < NOW){

  # 3-2. login

  REMDR$navigate('https://ep.kiet.re.kr/index.do')

  BUTTON_LOGIN <- REMDR$findElement('xpath', '//*[@id="f_login"]/ul/li[3]')
  TEXT_ID <- REMDR$findElement('xpath', '//*[@id="loginId"]')
  TEXT_PW <- REMDR$findElement('xpath', '//*[@id="pwd"]')

  TEXT_ID$sendKeysToElement(list(ID))
  TEXT_PW$sendKeysToElement(list(PW))
  BUTTON_LOGIN$clickElement()

  # 3-3. leave
```

```
  temp <- REMDR$findElement('name', 'mainFrame')

  REMDR$switchToFrame(temp)

  BUTTON_LEAVE <- REMDR$findElement('xpath', '//*[@id="left"]/div[1]/a[2]')

  BUTTON_LEAVE$clickElement()

  # 3-4. turn off

  ifelse(SHUTDOWN == 'YES', system('shutdown -s'), '전기를 아낍시다.')

  # 3-5. quit

  q()

 }

 Sys.sleep(300 + rnorm(n = 1, mean = 10, sd = 2))

}
```