Tidyverse Practice

Kyusun

2022-07-24

필요한 Package 가져오기

이번 과제는 Tidyverse 연습을 하는 것입니다. Tidyverse 패키지를 불러와봅시다.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.
3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflict
s() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

데이터 불러오기

같이 드린 reserve.csv 파일은 호텔 예약 정보를 모아둔 데이터셋입니다.

reserve.csv * reserve_id: 예약 ID * hotel_id: 호텔 ID * customer_id: 고객 ID * reserve_datetime: 예약 일시 * checkin_date: 체크인 날짜 * checkin_time: 체크인 시각 * checkout_date: 체크아웃 날짜 * people_num: 숙박 인원 * totel_price: 합계 금액

같이 드린 reserve.csv 파일을 reserve 라는 이름의 tibble 로 불러와봅시다.

```
reserve <- read_csv(file = "reserve.csv")
```

```
## Rows: 4030 Columns: 9
## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (3): reserve_id, hotel_id, customer_id
## dbl  (2): people_num, total_price
## dttm (1): reserve_datetime
```

```
## date (2): checkin_date, checkout_date
## time (1): checkin_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this m
essage.
```

Q1: reserve 데이터는 몇개의 row 가 있나요?

```
nrow(reserve)
```

```
## [1] 4030
```

가장 저렴한 호텔

알린이는 친구들과 갈 때 가장 저렴한 호텔을 찾고 있습니다. 1 일 기준 평균적인 예약가격이 가장 저렴한 호텔을 찾고 있습니다. 알린이는 해킹 능력이 매우 뛰어나 호텔 데이터베이스를 해킹해서 필요한 데이터를 가져왔지만, 데이터 분석은 해본적이 없어서, 가장 저렴한 호텔이 어딘지 계산하는거에 어려움을 겪고 있습니다. 우리가 도와줍시다.

Q2. select() 함수를 사용하여 분석에 필요한 열만을 추출하여 reserve_tb 라는 새로운 tibble 에 저장합시다. 1 일 기준 정보가 필요하니까 체크인 체크아웃 날짜도 필요하겠죠? 우선은 사람이 많아도 방 하나 가격은 같다고 가정하고, people_num 은 계산에서 제외합시다.

```
reserve_tb <- select(reserve, hotel_id, checkin_date, checkout_date, total_pr
ice)
reserve_tb
```

```
## # A tibble: 4,030 x 4
##    hotel_id checkin_date checkout_date total_price
##    <chr>    <date>       <date>              <dbl>
##  1 h_75     2016-03-26   2016-03-29          97200
##  2 h_219    2016-07-20   2016-07-21          20600
##  3 h_179    2016-10-19   2016-10-22          33600
##  4 h_214    2017-03-29   2017-03-30         194400
##  5 h_16     2017-09-22   2017-09-23          68100
##  6 h_241    2017-12-04   2017-12-06          36000
##  7 h_256    2018-01-25   2018-01-28         103500
##  8 h_241    2018-06-08   2018-06-09           6000
##  9 h_217    2016-03-25   2016-03-27          68400
## 10 h_240    2016-07-14   2016-07-17         320400
## # ... with 4,020 more rows
```

tibble 은 자동으로 날짜 형식의 chr 을 date 타입의 변수로 바꿔줍니다. 밑에는 date 타입의 변수끼리의 차이를 계산하는 방법에 대한 예시입니다. as.Date 는 chr 을 date 타입으로 바꿔주는 함수이지만, reserve_tb 는 이미 자동으로 date 타입 변수로 만들어졌지요.

```
as.Date("2016-03-26") - as.Date("2016-03-23")
```

```
## Time difference of 3 days
```

mutate() 함수를 추가해서 num_date 라는, 호텔에 묵은 날짜 변수를 reserve_tb 에 추가하여 새로운 데이터 reserve_tb1 을 만들어봅시다. 날짜 데이터 다루는 법은 따로 다루지 않았기 때문에, 이번만은 정답을 보여드리겠습니다.

```
reserve_tb1 <- mutate(reserve_tb,
                      num_date = as.numeric(checkout_date - checkin_date)
                      )
reserve_tb1
```

```
## # A tibble: 4,030 x 5
##    hotel_id checkin_date checkout_date total_price num_date
##    <chr>    <date>       <date>              <dbl>    <dbl>
##  1 h_75     2016-03-26   2016-03-29          97200        3
##  2 h_219    2016-07-20   2016-07-21          20600        1
##  3 h_179    2016-10-19   2016-10-22          33600        3
##  4 h_214    2017-03-29   2017-03-30         194400        1
##  5 h_16     2017-09-22   2017-09-23          68100        1
##  6 h_241    2017-12-04   2017-12-06          36000        2
##  7 h_256    2018-01-25   2018-01-28         103500        3
##  8 h_241    2018-06-08   2018-06-09           6000        1
##  9 h_217    2016-03-25   2016-03-27          68400        2
## 10 h_240    2016-07-14   2016-07-17         320400        3
## # ... with 4,020 more rows
```

date 끼리의 계산은 difftime 으로 time 변수로 저장됩니다. 3 days, 5 days 이렇게 저장되지요. numeric 변수인 total_price 와 계산하기 위해서는 num_date 도 time 변수형이 아니라, numeric 변수형으로 저장 되어야합니다. 그래서 as.numeric 으로 변수형을 바꾸어주었습니다.

Q3. mutate() 함수를 사용하여 price_per_day 라는 1 일당 숙박 가격 변수를 추가한 새로운 데이터 reserve_tb2 를 만들어봅시다.

```
reserve_tb2 <- mutate(reserve_tb1,
                      price_per_day = total_price/num_date
                      )
reserve_tb2
```

```
## # A tibble: 4,030 x 6
##    hotel_id checkin_date checkout_date total_price num_date price_per_day
##    <chr>    <date>       <date>              <dbl>    <dbl>         <dbl>
##  1 h_75     2016-03-26   2016-03-29          97200        3         32400
##  2 h_219    2016-07-20   2016-07-21          20600        1         20600
##  3 h_179    2016-10-19   2016-10-22          33600        3         11200
##  4 h_214    2017-03-29   2017-03-30         194400        1        194400
##  5 h_16     2017-09-22   2017-09-23          68100        1         68100
##  6 h_241    2017-12-04   2017-12-06          36000        2         18000
##  7 h_256    2018-01-25   2018-01-28         103500        3         34500
##  8 h_241    2018-06-08   2018-06-09           6000        1          6000
##  9 h_217    2016-03-25   2016-03-27          68400        2         34200
## 10 h_240    2016-07-14   2016-07-17         320400        3        106800
## # ... with 4,020 more rows
```

Q4. group_by() 와 summarize() 함수를 이용해서 각 호텔별로 1 일당 평균 숙박 가격을 계산하여 reserve_tb3 에 저장해봅시다.

```
reserve_tb3 <- group_by(reserve_tb2,hotel_id) %>%
  summarize(total_mean=mean(price_per_day,na.rm=TRUE))
reserve_tb3
```

```
## # A tibble: 300 x 2
##    hotel_id total_mean
##    <chr>         <dbl>
##  1 h_1           67860
##  2 h_10          14933.
##  3 h_100         12960
##  4 h_101         33765.
##  5 h_102         16615.
##  6 h_103         46980
##  7 h_104         84400
##  8 h_105         25407.
##  9 h_106         66600
## 10 h_107         56400
## # ... with 290 more rows
```

Q5. arrange() 함수를 이용하여 1 일당 평균 숙박 가격을 오름차순으로 정렬해봅시다. 가장 저렴한 호텔은 어디인가요?

```
arrange(reserve_tb3, total_mean)
```

```
## # A tibble: 300 x 2
##    hotel_id total_mean
##    <chr>         <dbl>
##  1 h_235          8750
##  2 h_35           9406.
##  3 h_197         10133.
```

```
##  4 h_44          10574.
##  5 h_224         10667.
##  6 h_74          10909.
##  7 h_15          11108.
##  8 h_41          11345.
##  9 h_24          11500
## 10 h_50          11769.
## # ... with 290 more rows
```

Q6. 알린이는 형 알령이가 다른 호텔에서 혼자서 4500 원에 묵은 사실을 알고 데이터 분석 결과에 의문을 가졌습니다. 아차! people_num 에 따라 가격이 달라진다는 걸 고려하지 못했네요. 이번에는 1 인 기준으로 다시 summarize()를 해봅시다. 그런데 이번에는 여러 데이터프레임을 계속 저장하지 말고 한번에 계산을 하도록 파이프 (%>%)를 활용하서 한번에 계산을 해보아요!!

```
reserve %>%
  filter(people_num==1) %>%  #1 인 기준 호텔 추리기
  summarize(hotel_id, one_ppl_price=(total_price/as.numeric(checkout_date-che
ckin_date))) %>% #hotel_id, 1 인 기준 1 일 가격
  arrange(one_ppl_price) %>% #오름차순 정리
  unique() #중복데이터 제거
```

```
## # A tibble: 284 x 2
##    hotel_id one_ppl_price
##    <chr>            <dbl>
##  1 h_35              3500
##  2 h_224             4000
##  3 h_197             4000
##  4 h_235             4200
##  5 h_15              4300
##  6 h_24              4600
##  7 h_100             4800
##  8 h_74              4800
##  9 h_44              4900
## 10 h_39              5000
## # ... with 274 more rows
```

호텔 매출 비교

통린이는 호텔 그룹 CEO 입니다. 최근에 코로나로 디폴트 위기에 처하자, 매출이 낮은 호텔부터 처분하기로 하였습니다. 그런데 통린이는 주가가 하락할때부터 데이터분석 팀을 모두 해고해버렸기 때문에, 현재 가장 매출이 낮은 호텔이 어디인지 알수가 없네요.

통린이는 2017 년 6 월 이후 예약 기준으로 총 예약횟수 그리고 총 매출을 낮은순으로 나열한 데이터를 깔끔하게 보고 싶습니다. 통린이를 도와줍시다!! summarize 에서 count(아무 변수)를 사용하면, 데이터프레임 내에서 그 호텔별로 예약수를 구할 수 있습니다. date 비교는 다음과 같이 할 수 있습니다.

```
# install.packages(lubridate)
library(lubridate)                    # 날짜와 시간 계산을 용이하게 해주는 패키지

##
## 다음의 패키지를 부착합니다: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

print(today())                        # today()는 오늘 날짜를 계산하고, print()는 출력을 해줍니다.

## [1] "2022-07-27"

as.Date("2016-03-14") > today()       # as.Date 는 문자열을 date 로 바꿔주고, 숫자처럼, <와 >로 날짜를 비교할 수 있습니다.

## [1] FALSE
```

여기에 파이프를 사용해서 작성해주세요.

```
reserve %>%
  filter(reserve_datetime >= as.Date("2017-06-01")) %>%   #예약일 기준 데이터 추출
  group_by(hotel_id) %>%    #hotel_id 에 따른 group 생성
  summarize(hotel_id, total_reserve=n(), total_sales=sum(total_price)) %>%
#총매출, 총예약 정리
  arrange(total_sales,total_reserve) %>%    #오름차순 정리
  unique()  #중복 데이터 제거

## `summarise()` has grouped output by 'hotel_id'. You can override using the
## `.groups` argument.

## # A tibble: 291 x 3
## # Groups:   hotel_id [291]
```

```
##    hotel_id total_reserve total_sales
##    <chr>            <int>       <dbl>
##  1 h_24                 1        4600
##  2 h_76                 1        9200
##  3 h_229                1       14800
##  4 h_108                1       17200
##  5 h_102                1       18000
##  6 h_265                1       18600
##  7 h_279                2       18900
##  8 h_122                1       20100
##  9 h_208                1       26700
## 10 h_18                 1       27800
## # ... with 281 more rows
```