파생변수 생성

2015122026오태환

2020 5 26

1) Import Data

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

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

```
data = read_csv("new_merge_data.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   artist = col_character(),
##   st_day = col_date(format = ""),
##   name = col_character(),
##   production = col_character(),
##   distributor = col_character(),
##   ed_day = col_date(format = ""),
##   name_artist = col_character(),
##     = col_character(),
##      = col_time(format = ""),
##     = col_character(),
```

```
##      = col_character(),
##   month = col_character(),
##   week = col_character(),
##   sex = col_character(),
##   song_type = col_character(),
##   genre = col_character(),
##   runtime_g = col_character(),
##   active_type = col_character(),
##   active = col_character(),
##   active_g = col_character()
##   # ... with 1 more columns
## )

## See spec(...) for full column specifications.
```

```
summary(data)
```

```
##     artist              st_day                X1              id
##  Length:17400       Min.   :2018-04-22   Min.   :    1   Min.   :    1
##  Class :character   1st Qu.:2019-01-20   1st Qu.: 4351   1st Qu.: 4351
##  Mode  :character   Median :2019-06-23   Median : 8700   Median : 8700
##                     Mean   :2019-06-09   Mean   : 8700   Mean   : 8700
##                     3rd Qu.:2019-11-24   3rd Qu.:13050   3rd Qu.:13050
##                     Max.   :2020-04-19   Max.   :17400   Max.   :17400
##
##      name            production        distributor            ed_day
##  Length:17400       Length:17400       Length:17400       Min.   :2018-04-28
##  Class :character   Class :character   Class :character   1st Qu.:2019-01-26
##  Mode  :character   Mode  :character   Mode  :character   Median :2019-06-29
##                                                           Mean   :2019-06-15
##                                                           3rd Qu.:2019-11-30
##                                                           Max.   :2020-04-25
##
##      score               rank        name_artist
##  Min.   : 2288566   Min.   :  1.00   Length:17400       Length:17400
##  1st Qu.: 4015750   1st Qu.: 42.00   Class :character   Class :character
##  Median : 7184503   Median : 83.00   Mode  :character   Mode  :character
##  Mean   : 9682174   Mean   : 90.16
##  3rd Qu.:11855977   3rd Qu.:137.00
##  Max.   :85411467   Max.   :200.00
##
##                                            song_id
##  Length:17400       Length:17400       Length:17400       Min.   :   1
##  Class1:hms         Class :character   Class :character   1st Qu.: 157
##  Class2:difftime    Mode  :character   Mode  :character   Median : 444
##  Mode  :numeric                                           Mean   : 517
##                                                           3rd Qu.: 800
##                                                           Max.   :1447
##
##      rank_g             year          month                day
##  Min.   : 1.000   Min.   :18.00   Length:17400       Min.   : 1.00
##  1st Qu.: 5.000   1st Qu.:19.00   Class :character   1st Qu.: 8.00
##  Median : 9.000   Median :19.00   Mode  :character   Median :15.50
##  Mean   : 7.284   Mean   :18.97                      Mean   :15.53
##  3rd Qu.:10.000   3rd Qu.:19.00                      3rd Qu.:23.00
##  Max.   :10.000   Max.   :20.00                      Max.   :31.00
##
```

```
##       wks             week               sex             song_type
##  Min.   : 1.00   Length:17400       Length:17400       Length:17400
##  1st Qu.:11.00   Class :character   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :24.93
##  3rd Qu.:38.00
##  Max.   :52.00
##
##     genre              runtime       runtime_g          active_type
##  Length:17400       Min.   : 26.0   Length:17400       Length:17400
##  Class :character   1st Qu.:201.0   Class :character   Class :character
##  Mode  :character   Median :221.0   Mode  :character   Mode  :character
##                     Mean   :224.5
##                     3rd Qu.:242.0
##                     Max.   :364.0
##
##     active            active_g            top_freq       top_freq_g
##  Length:17400       Length:17400       Min.   :  1.00   Length:17400
##  Class :character   Class :character   1st Qu.: 14.00   Class :character
##  Mode  :character   Mode  :character   Median : 35.00   Mode  :character
##                                        Mean   : 38.89
##                                        3rd Qu.: 59.00
##                                        Max.   :105.00
##
##   dc_t_number       dc_t_recommend       dc_t_views      dc_m_recommend
##  Min.   :     0.0   Min.   :     0.0   Min.   :      0   Min.   : 0.0000
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:      0   1st Qu.: 0.0000
##  Median :     0.0   Median :     0.0   Median :      0   Median : 0.0000
##  Mean   :   752.9   Mean   :   988.3   Mean   :  56994   Mean   : 0.2425
##  3rd Qu.:     0.0   3rd Qu.:     0.0   3rd Qu.:      0   3rd Qu.: 0.0000
##  Max.   :122450.0   Max.   :219258.0   Max.   :9241868   Max.   :19.0000
##
##    dc_m_views         gg_score         nv_score        ndc_t_numb
##  Min.   :   0.00   Min.   :  0.00   Min.   :  0.00   Min.   :     1
##  1st Qu.:   0.00   1st Qu.: 18.00   1st Qu.:  2.80   1st Qu.:   185
##  Median :   0.00   Median : 33.00   Median :  9.00   Median :  2451
##  Mean   :  22.05   Mean   : 35.72   Mean   : 18.99   Mean   :  6339
##  3rd Qu.:   0.00   3rd Qu.: 49.00   3rd Qu.: 27.90   3rd Qu.:  8040
##  Max.   :3179.50   Max.   :100.00   Max.   :100.00   Max.   :100092
##                                                      NA's   :12553
##    ndc_t_rec          ndc_t_view
##  Min.   :     0.0   Min.   :     43
##  1st Qu.:   233.5   1st Qu.:  32020
##  Median :  2189.0   Median : 217044
##  Mean   :  9549.1   Mean   : 650466
##  3rd Qu.:  9291.0   3rd Qu.: 788900
##  Max.   :212290.0   Max.   :9547158
##  NA's   :12553      NA's   :12553
```

2) 파생변수 만들기

2-1) 이전 곡 순위

```
previous_song = data %>% select(artist, name) %>%
  unique() %>% group_by(artist) %>% mutate(previous_song = lag(name,1)) %>% ungroup()
```

가수별로 그룹핑을 하고 곡 칼럼을 한 칸씩 밀어 이전 곡 칼럼을 생성했다.

```
data = previous_song %>% merge(data, by = c("name", "artist"))


data = data %>% select(artist, name, previous_song, rank) %>%
  group_by(name) %>%
  mutate(top_rank = min(rank)) %>%
  ungroup() %>%
  select(-rank) %>%
  unique() %>%
  group_by(artist) %>%
  mutate(previous_song_rank = lag(top_rank, 1)) %>%
  ungroup() %>%
  merge(data, by = c("artist", "name", "previous_song")) %>%
  select(-c("top_rank", "previous_song"))
```

이전 곡의 최고 랭킹 칼럼을 만들었다.

2-2) 계절 변수 추가

```
data$month = data$month %>% unlist() %>% as.integer()


season = function(x){
  if(any(x == c("3", "4", "5"))){
    return("spring")
  }
  if(any(x == c("6", "7", "8"))){
    return("summer")
  }
  if(any(x == c("9", "10", "11"))){
    return("fall")
  }
  else{
    return("winter")
  }
}


for(i in 1:nrow(data)){
  data$season[i] = season(data$month[i])
}

write.csv(data, "firstmodeldata.csv")
```

3,4,5월은 봄, 6,7,8월은 여름, 9,10,11월은 가을, 12,1,2월은 겨울로 하는 칼럼을 만들었다.