Mid_Project

Kim In Soo

2021 4 21

Contents Index

Preface

- **Problem Definition**
  - 사람들은 매일 '오늘 뭐 먹지?'라는 고민과 마주합니다. 그들이 음식을 고르는 기준은 다양한데, '맛', '가격', '거리', '시간' 등 여러가지를 고려하여 의사결정을 합니다. 물론, 맛이 좋고 가격이 싸면서 가게와의 거리가 가까운 식당을 찾는 것이 최적의 선택이겠지만, 상황에 따라 그리고 구매자의 성향에 따라 그 기준에는 우선순위가 존재합니다. 하지만, 구매자 입장에서 우선순위가 정해져 있다고 해도 적절한 선택지를 찾기에는 생각보다 많은 시간이 소요됩니다.

    특히, 2030 세대들이 주로 이용하는 SNS 인 '인스타그램'에는 해시태그 기능이 존재하여 해시태그를 입력하면 관련 이미지와 글들을 확인할 수 있습니다. 여기서 문제점은 정보의 양이 너무나도 방대할 뿐만 아니라 정렬기능 조차 존재하지 않아 한번에 가게 정보를 파악하는 데 어려움이 존재합니다. 또한 광고 게시물이 많아 믿을 만한 가게인지 확인하기 어려운 실정입니다.

- **Subject of Project**
  - 따라서 본 프로젝트는 종로구와 성북구 근처에 거주하는 대학생들을 대상으로 종로구와 성북구의 HOT 한 맛집을 추천해주는 프로젝트입니다. 음식점 이용은 크게 '외식'과 '배달음식' 두 가지로 분류할 수 있습니다. 이렇게 두 가지로 구분한 이유는 각 유형에서 강조되는 우선순위가 다르다고 생각하기 때문입니다. 외식에 있어서는 가격이 다소 비싸더라도 가게의 분위기와 음식의 맛이 강조되는 반면, 배달음식에 있어서는 배달시간과 음식의 맛, 그리고 가격이 강조된다고 생각합니다. 따라서 본 프로젝트의 주제는 외식에서 HOT 한 (분위기와 맛을 고려한) 맛집과 배달음식에서 HOT 한 (배달시간,음식의 맛,가격을 고려한) 맛집 추천입니다.

- **Goal of Project**
  - 자연어처리와 이미지처리에 특화된 딥러닝 모델을 기반으로 하여 '외식' 부문에선 인스타에 게시된 음식점의 적절한 평점을 부여하고, '배달음식' 부문에선 배달 어플(요기요 혹은 배달의 민족)에 게시된 음식점의 정보를 학습하여 구매 당사자가 사진을 입력하면 HOT 한 맛집 중 그와 유사한 음식을 추천해주는 것을 목표로 합니다.

About Data

- Source
  - 중간 프로젝트까지는 우선 외식 부문의 데이터를 수집하는데 초점을 맞추었습니다. 최종 프로젝트에선 배달어플 리뷰 데이터(이미지 및 자연어)까지 수집을 목표로 하고 있습니다. 외식 부문의 데이터를 수집하는데 있어,

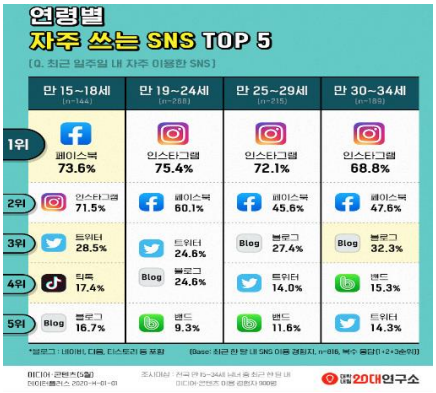
    첫 번째 데이터셋은

    2030 세대들이 가장 많이 이용하는 SNS 인 인스타그램의 데이터를 크롤링하여 정보를 수집하였습니다. 즉, 해시태그 기능을 통해 '성북구맛집', '성북구맛집추천','종로구맛집','종로구맛집추천'을 검색하였고 로드된 데이터들을 크롤링하여 성북구 관련해서 각각 (5852 & 244)개의 데이터와 종로구 관련해서 각각 (9727 & 79)개의 데이터를 수집하였습니다.

    두 번째 데이터셋은

    공공데이터포털의 '소상공인시장진흥공단_상가(상권)정보_서울' 데이터를 다운로드하여 가져왔습니다. 이는 인스타 게시글의 정보에 해당 가게의 장소와 업종분류명이 나타나지 않은 경우가 빈번하여 정확한 장소를 추천하기 위해 이를 해결하고자 추가적으로 이용하였습니다.

    (최종 프로젝트에 이용할 예정입니다.)



(출처 : 대학내일 20 대 연구소)

- Crawling code (ipynb 파일 첨부)

- Description of each data
  - 1. 종로구_맛집 : '성북구맛집'을 해시태그 검색 후,
    "날짜", "좋아요 수","장소", "본문", "태그", "댓글" 정보를 크롤링하여
    얻은 데이터 (총 9727 개의 관측치와 6 개의 변수)
  - 2. 종로구_맛집추천: '성북구맛집추천'을 해시태그 검색 후,
    "날짜", "좋아요 수","장소", "본문", "태그", "댓글" 정보를 크롤링하여
    얻은 데이터 (총 79 개의 관측치와 6 개의 변수)
  - 3. 성북구_맛집 : '종로구맛집'을 해시태그 검색 후,
    "날짜", "좋아요 수","장소", "본문", "태그", "댓글" 정보를 크롤링하여
    얻은 데이터 (총 5852 개의 관측치와 6 개의 변수)
  - 4. 성북구_맛집추천 : '종로구맛집추천'을 해시태그 검색 후,
    "날짜", "좋아요 수","장소", "본문", "태그", "댓글" 정보를 크롤링하여
    얻은 데이터 (총 244 개의 관측치와 6 개의 변수)
    - 결측치처리 :
      날짜(''), 좋아요 수(0), 장소(''), 본문(''), 태그(''), 댓글('')
    - 좋아요 수를 제외하고 모두 blank 로 예외처리 하였습니다.
    - 댓글 수 : 게시물 당 화면에 보이는 댓글만 크롤링하였습니다.

  - 5. 종로구_성북구_상가분류 : '소상공인시장진흥공단_상가(상권)정보_서울' 데이터
    중에서 상권업종대분류명이 음식이고 시군구명이 종로구이거나 성북구인
    관측치들만 추출하고 39 개의 column 중 필요한 변수만 선택.
    변수명 : "상호명", "지점명", "상권업종대분류명", "상권업종중분류명",
    "상권업종소분류코드", "상권업종소분류명", "표준산업분류명", "시군구명",
    "행정동명", "법정동명", "건물명", "신우편번호", "경도", "위도"
    (총 9866 개의 관측치와 14 개의 변수)

- Data Integration
  - 종로구_핫플 : 종로구_맛집 데이터와 종로구_맛집추천 데이터를 통합 및 중복되는 행을 제거하였습니다.
  - 성북구_핫플 : 성북구_맛집 데이터와 성북구_맛집추천 데이터를 통합 및 중복되는 행을 제거하였습니다.

EDA

- Data Pre-processing

'소상공인시장진흥공단_상가(상권)정보_서울' 데이터 전처리

```
shop = shop %>%
  filter(
  (상권업종대분류명 == '음식'& 시군구명 == '성북구') |
  (상권업종대분류명 == '음식'& 시군구명 == '종로구')) %>%
  select(c(2,3,5,7,8,9,11,15,17,19,31,34,38,39))
write.csv(shop, file = '종로구_성북구_상가분류.csv')
```

  - 종로구_데이터 통합 -> 종로구_핫플

```
jr = union(jongro, jongro_) # 중복된 행 제거
dim(jr)
## [1] 9363    6
jr1 = jr %>%
  select(date, like, place, content, tags, comments) %>%
  mutate(date = as.Date(date),
         like = parse_number(like)) %>%
  arrange(date, desc(like)) %>%
  filter(content != 'blank')   # 잘못 크롤링 된 obs  제외

write.csv(jr1, file = '종로구_핫플.csv')
```

- 성북구_데이터 통합 -> 성북구_핫플

```
sb = union(seongbuk, seongbuk_)
dim(sb)
```

```
## [1] 5738    6
```

```
sb1 = sb %>%
  select(date, like, place, content, tags, comments) %>%
  mutate(date = as.Date(date),
         like = parse_number(like)) %>%
  arrange(date, desc(like)) %>%
  filter(content != 'blank')

dim(sb1)
```

```
## [1] 5736    6
```

```
write.csv(sb1, file = '성북구_핫플.csv')

jr1 %>%
  as.data.table() %>%
  arrange(date) %>% select(date) # date 범위 : 2015-11-06 ~ 2021-04-20
```

```
##             date
##    1: 2015-11-06
##    2: 2017-05-24
##    3: 2017-05-28
##    4: 2017-05-31
##    5: 2017-06-01
##   ---
## 9355: 2021-04-20
## 9356: 2021-04-20
## 9357: 2021-04-20
## 9358: 2021-04-20
## 9359: 2021-04-20
```

```
# 결측치가 중복되어 있습니다.

(즉, place 가 NA 인 행이 동시에 다른 변수도 결측치인 행이 존재)

jr_na_place = jr1 %>% filter(is.na(place)) %>% nrow()
jr_na_like = jr1 %>% filter(like == 0) %>% nrow()
jr_na_tags = jr1 %>% filter(tags == '[]') %>% nrow()
jr_na_comments = jr1 %>% filter(comments == '[]') %>% nrow()
```

```
jr_na_df = data.frame(
  var = c('place','like','tags','comments'),
  na_count = c(jr_na_place, jr_na_like, jr_na_tags, jr_na_comments)
) %>% print()
```

```
##        var na_count
## 1    place     5822
## 2     like      445
## 3     tags      871
## 4 comments     4411
```

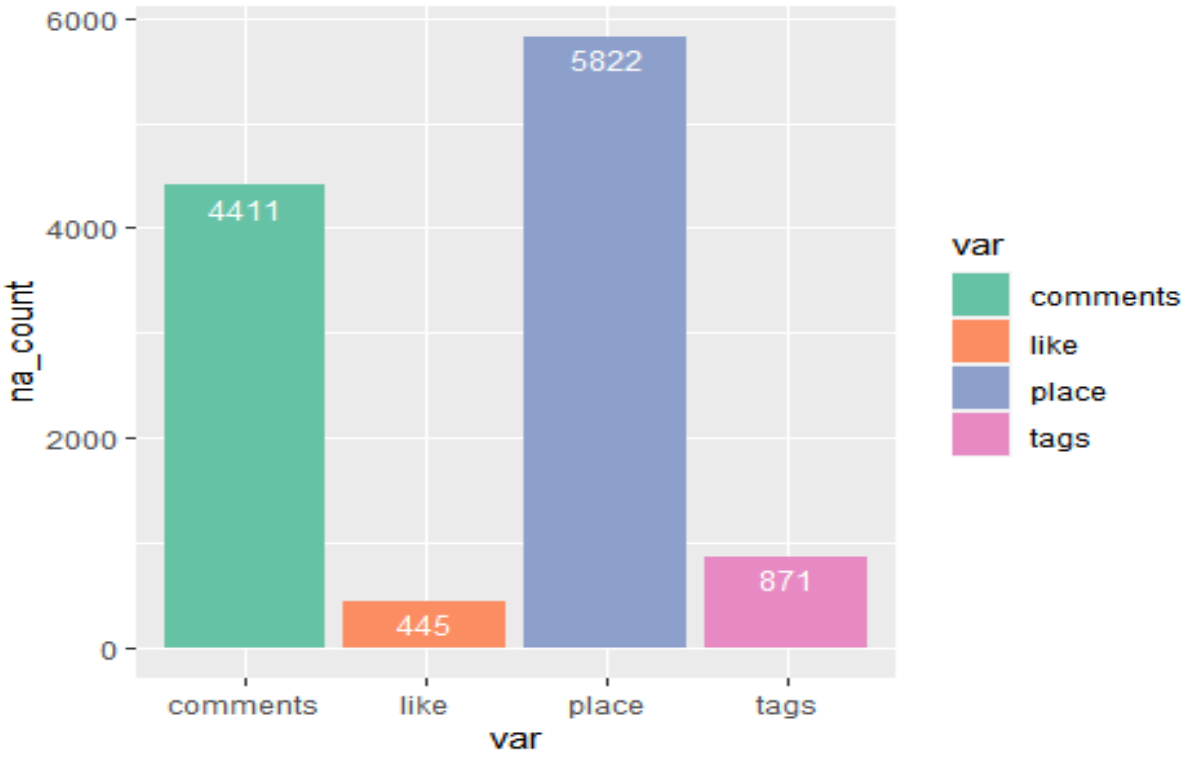
- jr_na_df 의 숫자가 더 큰 것을 확인할 수 있습니다.
- 하지만, 중복되어 있지 않는 경우 경우는 크게 의미 없다고 판단했습니다.

```
# 중복 결측치 경우 제외했을 때
jr1 %>% transmute(missing = case_when(
  like == 0 ~ 'ms_like',
  is.na(place) ~ 'ms_place',
  tags == '[]' ~ 'ms_tags',
  comments == '[]' ~ 'ms_comments',
  TRUE ~ NA_character_)) %>%
  group_by(missing) %>%
  summarise(
    missing_count = n())
```

```
## # A tibble: 5 x 2
##   missing     missing_count
##   <chr>               <int>
## 1 ms_comments           762
## 2 ms_like               445
## 3 ms_place             5475
## 4 ms_tags               636
## 5 <NA>                 2041
```

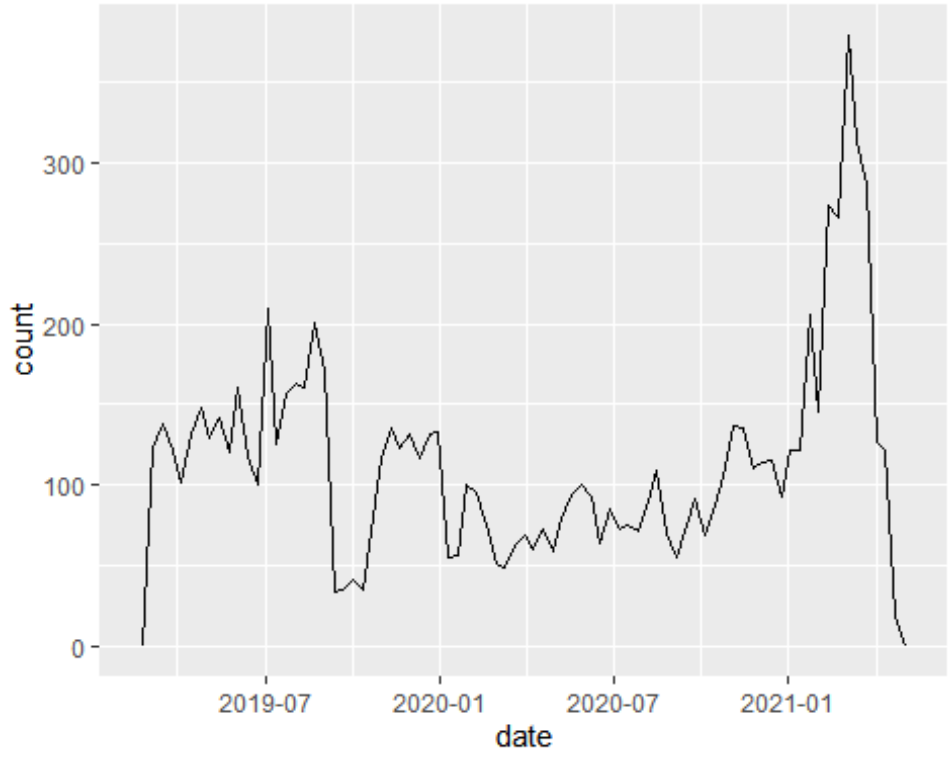
- 결측치 개수가 몇개가 있는지에 대한 bar plot
- 그 결과, place 변수의 결측치가 5822 개로 압도적으로 큰 것을 확인

```
jr_na_df %>% ggplot(aes(x=var, y=na_count, fill=var))+
  geom_bar(stat='identity')+
  scale_fill_brewer(palette = "Set2")+
  geom_text(aes(label = na_count), vjust = 1.5, size = 3.5, color = "white")
```



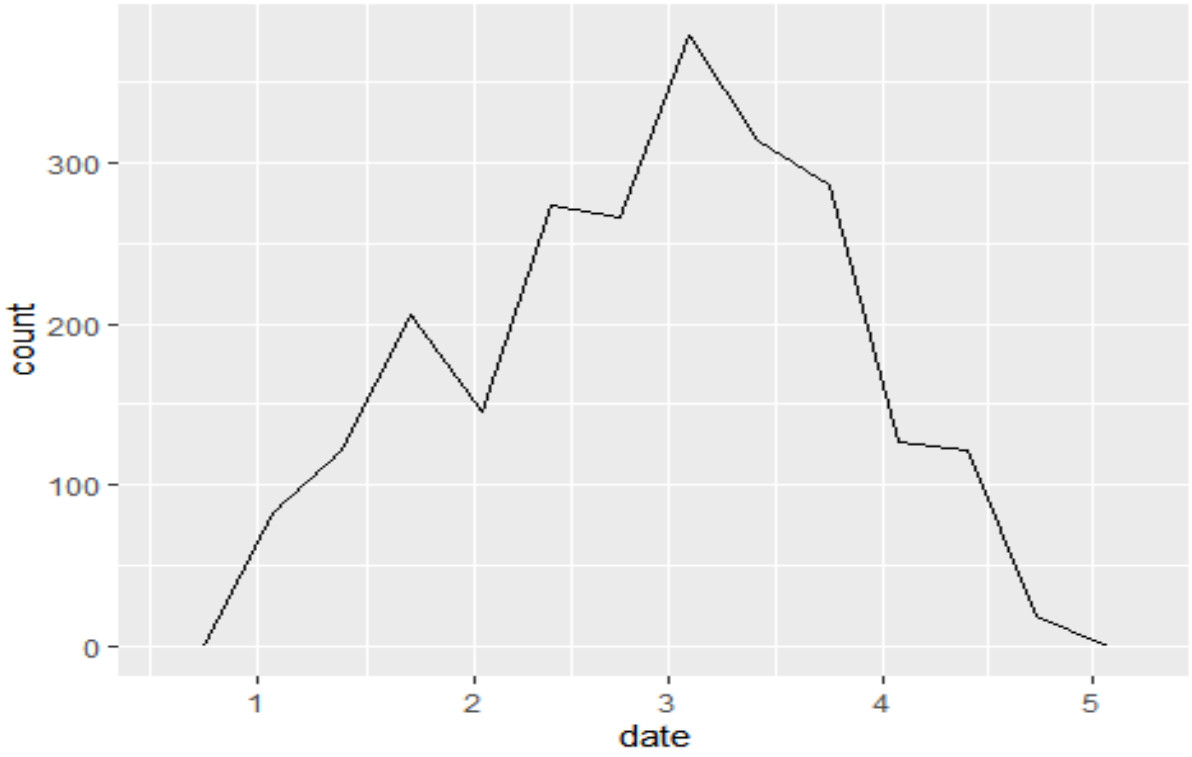
- 종로구 핫플 데이터의 시기별 게시글 개수의 빈도분포를 살펴보았습니다.
- (19/03/02 ~ 20/04/20) 시작날짜를 19 년으로 잡은 이유는 19 년 이전의 데이터의 양이 많지 않을 뿐 아니라 3 월 은 대학생의 개강 달이라는 의미가 있다고 판단하였기 때문입니다. 결과를 보면, 19 년 7 월에 게시글 수가 다소 많았다가 코로나가 시작한 시점(20 년 1 월 말) 이후로 1 년간 100 개 안팎의 범위에 위치해 있는 것을 볼 수 있습니다.

```
# 시간에 따른 게시글 개수  (check 1)
jr1 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20190301)) %>%
  ggplot(aes(date))+
  geom_freqpoly(binwidth = 10*86400)
```



- 21 년 1 월 ~ 4 월까지 중 3 월에 peak 점에 도달하였습니다. 정확한 이유는 이 plot 만 가지고는 알 수 없지만 개강과 봄이 시작되면서 게시글 수가 늘어난 것이 아닐까 추측해보았습니다.

```
jr1 %>% # 21 년 기준
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20210101)) %>%
  ggplot(aes(date))+
  geom_freqpoly(binwidth = 10*86400)
```



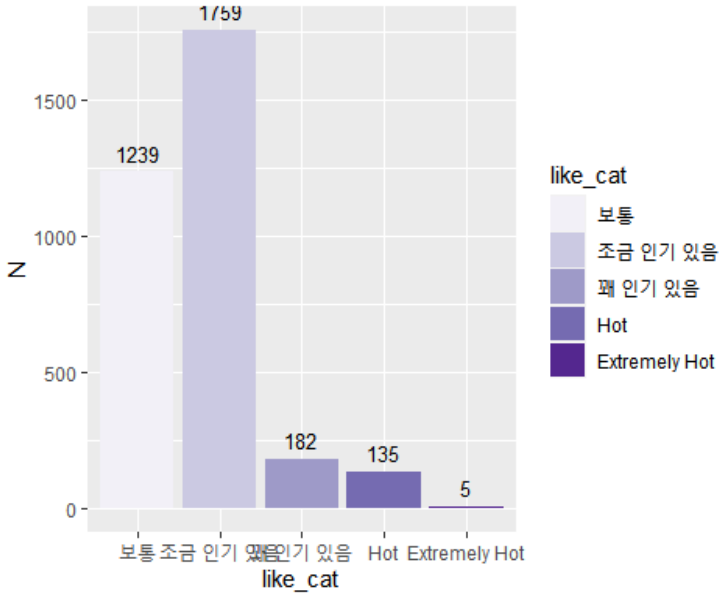
- Descriptive Analysis for the variable 'like' (only numeric)
  - like 값들의 분산이 굉장히 크다는 것을 직관적으로 알 수 있습니다. 그렇다고 outlier 에 영향을 받지 않는 중앙값을 중간지점으로 판단하기에는 무리가 있어보입니다. 인스타그램을 몇년간 이용해본 결과 25 라는 숫자보다 크면 인기 있다고 판단되지 않기 때문입니다.
  -

```
as.data.table(jr1)[, summary(like,na.rm=TRUE)]
```

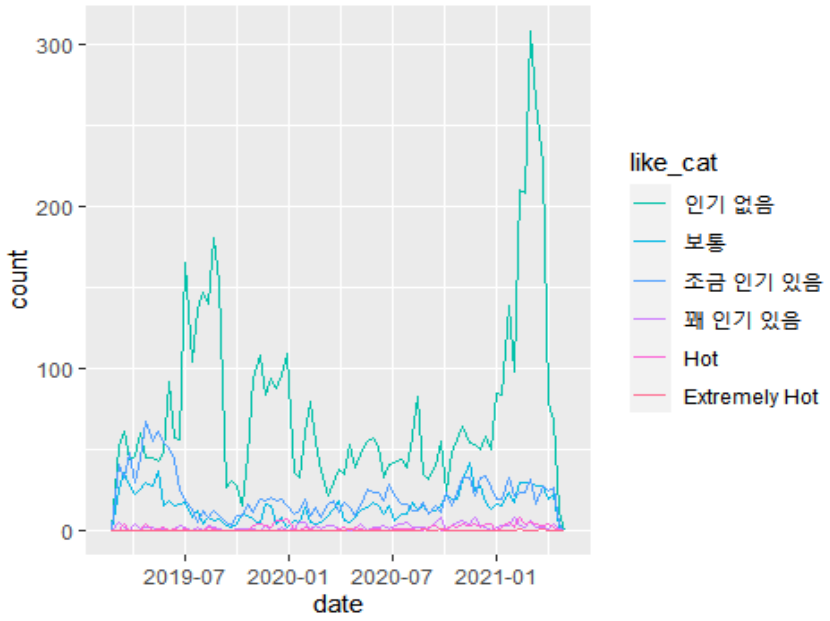
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    8.00   25.00   99.23   84.00 8019.00      21
```

  - 따라서 중앙값이 아닌 평균인 99 보다 큰 숫자의 좋아요 게시물은 인기가 조금 있다고 판단해서 범주형 변수를 생성해보았습니다. 여기서 좋아요가 1000 개 ~ 4999 개를 핫플로, 5000 개 이상은 극단적인 핫플로 설정하였지만 추후 모델을 세울 때 있어서 이 부분은 주의가 필요할 것으로 판단됩니다. 인스타에는 광고 게시물이 굉장히 많기 때문에 이렇게 과도하게 좋아요가 많은 게시물의 경우 그 이유가 정말 해당 가게가 유명해서일 수 있지만 게시물 업로드 계정이 슈퍼계정이기 때문에 가게와 상관없이 좋아요만 많고 '알맹이 없는' 게시물일 가능성이 존재합니다. 이를 주의해서 모델을 세울 예정입니다.

```
jr2 = jr1 %>%
  mutate(like_cat = ifelse(like<50, '인기 없음',
                           ifelse(like<100, '보통',
                                  ifelse(like<500, '조금 인기 있음',
                                         ifelse(like<1000, '꽤 인기 있음',
                                                ifelse(like<5000, 'Hot','Extr
emely Hot'))))))) %>%
  select(date,like,like_cat,place,content,tags,comments) %>%
  arrange(desc(like), like_cat, date, place, content, tags, comments)


jr2 = jr2 %>%
  mutate(like_cat = factor(like_cat, levels = c('인기 없음','보통','조금 인기
있음', '꽤 인기 있음','Hot','Extremely Hot')))

# like_cat bar plot
as.data.table(jr2)[, .(.N), by='like_cat'] %>%
  filter(like_cat!='인기 없음' & !is.na(like_cat)) %>%
  ggplot(aes(x=like_cat, y=N, fill=like_cat))+
  geom_bar(stat='identity')+
  scale_fill_brewer(palette = "Purples",direction = 1)+
  geom_text(aes(label = N), vjust = -0.6, size = 3.5, color = "black")
```

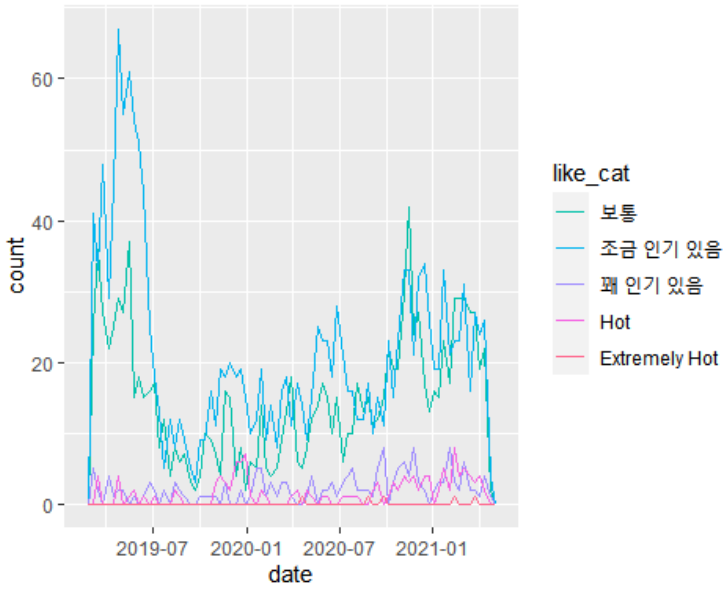


```r
# like_cat 별 좋아요 개수 평균
as.data.table(jr2)[, mean(like, na.rm = TRUE), by='like_cat']

[!is.na(like_cat)] %>%
  mutate(mean = V1) %>% select(-c(V1))

##          like_cat       mean
## 1:  Extremely Hot 6247.20000
## 2:            Hot 1700.33333
## 3:   꽤 인기 있음  674.81868
## 4: 조금 인기 있음  208.79420
## 5:           보통   69.90395
## 6:      인기 없음   14.81040

# 시간에 따른 게시글 개수 (좋아요 카테고리에 따른) (check 2)

# 인기 순에 따라 빈도 순위의 차이가 보임
jr2 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20190301)& !is.na(like_cat)) %>%
  ggplot(aes(date)) + geom_freqpoly(aes(group = like_cat, colour = like_cat),
binwidth = 10*86400) +
  scale_colour_hue(h = c(180, 360))
```



- 예상과 달리, 좋아요가 50 이상인 데이터에 대해선 조금 인기 있는 게시물의 빈도가 보통의 게시물보다 높은 점을 확인할 수 있습니다.

```
# 좋아요 카테고리가 보통이상인 데이터에 대해선? (graph 확대)
jr2 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter( (date > ymd(20190301)) & (like >= 50) ) %>%
  ggplot(aes(date)) + geom_freqpoly(aes(group = like_cat, colour = like_cat),
binwidth = 10*86400) +
  scale_colour_hue(h = c(180, 360))
```




```
# 최소 좋아요가 100 개 이상인 데이터만!
jr3 = jr2 %>% filter((like_cat !='인기 없음') & (like_cat !='보통'))

nrow(jr3) # 2081
```

```
## [1] 2081
```

```
# 댓글이 아예 없는 데이터 제거 (댓글이 최소 1 개인 데이터)
jr4 = jr3 %>% filter(comments != '[]') # 1538


# 장소별 count
dim(jr2) # 9359
```

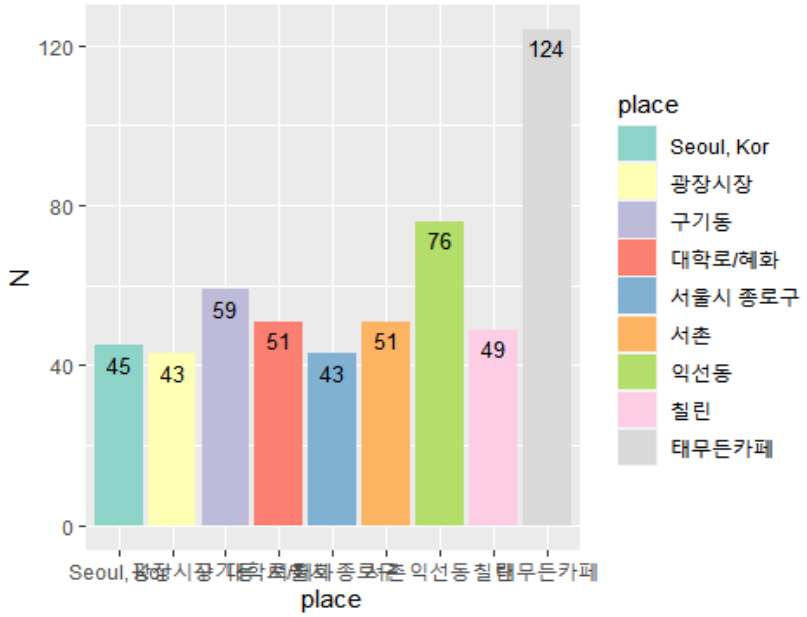
```
## [1] 9359    7
```

```
jr_count_place = as.data.table(jr2)[, .(.N), by='place'][order(-N)]
jr_count_place = jr_count_place %>% mutate(place = str_sub(place,1,10) )
jr_count_place[1:10] # top 10 # N = 43
```

```
##             place    N
##  1:          <NA> 5822
##  2:    태무든카페  124
##  3:        익선동   76
##  4:        구기동   59
##  5:   대학로/혜화   51
##  6:          서촌   51
```

```
##  7:          칠린   49
##  8:    Seoul, Kor   45
##  9:      광장시장   43
## 10: 서울시 종로구   43
```

```
# visualization # 좋아요 수 or 댓글 수와 상관없이
jr_count_place %>%
  filter(N>42 & !is.na(place)) %>%
  ggplot(aes(x=place, y=N, fill=place))+
  geom_bar(stat = 'identity')+
  scale_fill_brewer(palette = "Set3")+
  geom_text(aes(label = N), vjust = 1.5, size = 3.5, color = "black")
```

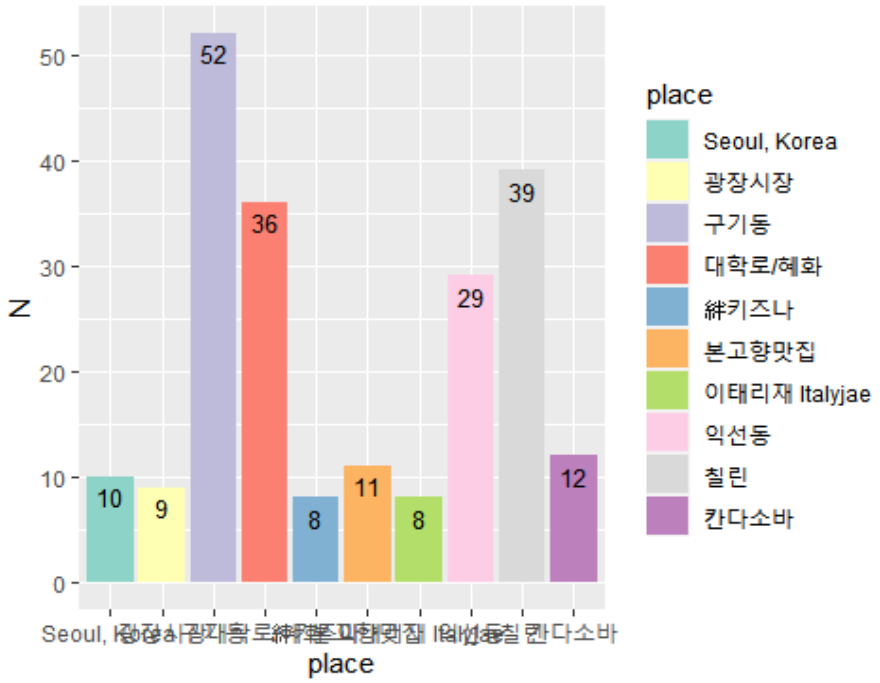



```
# 좋아요 수가 100 개 이상인 2081 개의 데이터 (jr3)
jr_count_place_100 = as.data.table(jr3)[, .(.N), by='place'][order(-N)]

# 517 개의 장소
jr_count_place_100[1:10] # top 10 # N = 8
```

```
##           place    N
##  1:         <NA> 1080
##  2:       구기동   52
##  3:         칠린   39
##  4:  대학로/혜화   36
```

```
##  5:        익선동   29
##  6:      칸다소바   12
##  7:    본고향맛집   11
##  8: Seoul, Korea   10
##  9:      광장시장    9
## 10:      絆키즈나    8
```

```
jr_count_place_100 %>%
  filter(N>7 & !is.na(place)) %>%
  ggplot(aes(x=place, y=N, fill=place))+
  geom_bar(stat = 'identity')+
  scale_fill_brewer(palette = "Set3")+
  geom_text(aes(label = N), vjust = 1.5, size = 3.5, color = "black")
```

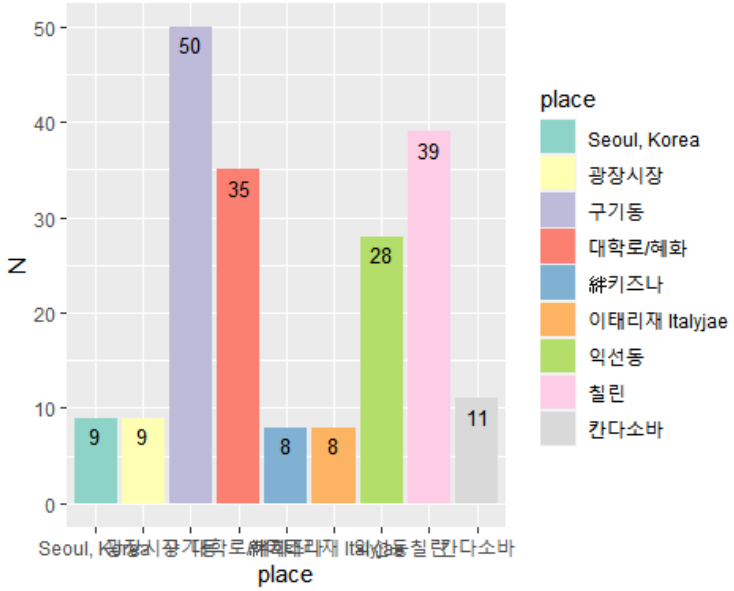



```
# 좋아요 100 개 이상이고 댓글이 1 개라도 있는 1538 개의 데이터 (jr4)
jr_count_place_100_C = as.data.table(jr4)[, .(.N), by='place'][order(-N)]

# 503 개의 장소
jr_count_place_100_C[1:10] # top 10 # N = 8
```

```
##                 place   N
##  1:              <NA> 581
##  2:            구기동  50
##  3:              칠린  39
```

```
##  4:       대학로/혜화  35
##  5:            익선동  28
##  6:          칸다소바  11
##  7:          광장시장   9
##  8:      Seoul, Korea   9
##  9:            絆키즈나   8
## 10: 이태리재 Italyjae   8
```

```
jr_count_place_100_C %>%
  filter(N>7 & !is.na(place)) %>%
  ggplot(aes(x=place, y=N, fill=place))+
  geom_bar(stat = 'identity')+
  scale_fill_brewer(palette = "Set3")+
  geom_text(aes(label = N), vjust = 1.5, size = 3.5, color = "black")
```



- 성북구의 데이터도 종로구와 같은 원리로 EDA 과정을 거쳤습니다.

  아래는 그 코드들 입니다.

```
# 2017-06-18 ~ 2021-04-20
sb1 %>%
  as.data.table() %>%
  arrange(date) %>% select(date)
```

```
##             date
##    1: 2017-06-18
##    2: 2017-06-21
##    3: 2017-10-25
##    4: 2017-10-25
```

```
##    5: 2017-12-13
##   ---
## 5732: 2021-04-20
## 5733: 2021-04-20
## 5734: 2021-04-20
## 5735: 2021-04-20
## 5736: 2021-04-20
```

```
# 중복 결측치 경우 포함
sb_na_place = sb1 %>% filter(is.na(place)) %>% nrow()
sb_na_like = sb1 %>% filter(like == 0) %>% nrow()
sb_na_tags = sb1 %>% filter(tags == '[]') %>% nrow()
sb_na_comments = sb1 %>% filter(comments == '[]') %>% nrow()
```

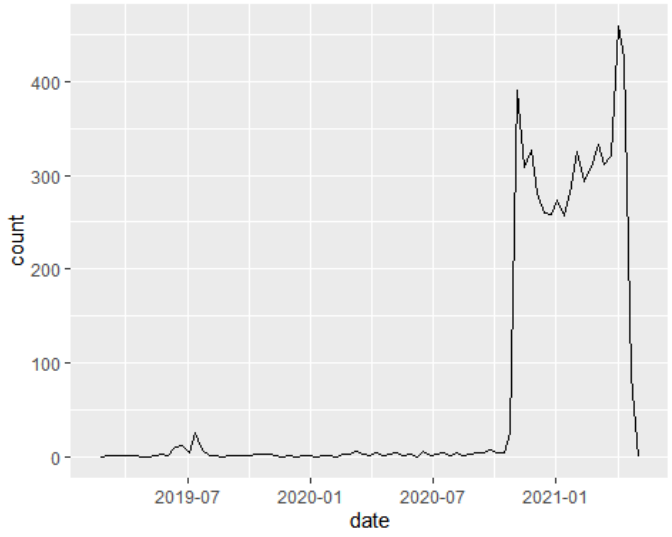
```
# 결측치 df
sb_na_df = data.frame(
  var = c('place','like','tags','comments'),
  na_count = c(sb_na_place, sb_na_like, sb_na_tags, sb_na_comments)
) %>% print()
```

```
##        var na_count
## 1    place     2992
## 2     like      257
## 3     tags      938
## 4 comments     1659
```

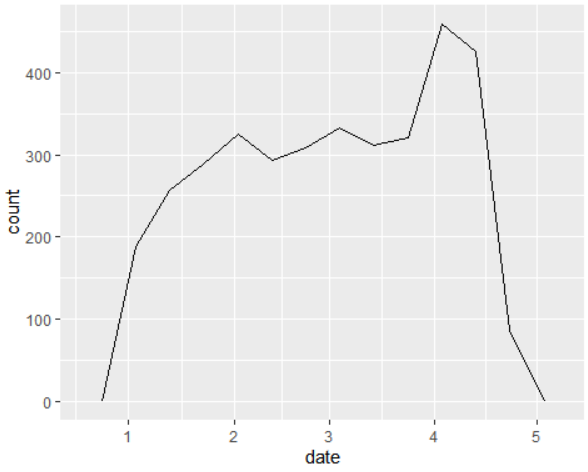
```
sb_na_df %>% ggplot(aes(x=var, y=na_count, fill=var))+
  geom_bar(stat='identity')+
  scale_fill_brewer(palette = "Set2")+
  geom_text(aes(label = na_count), vjust = 1.5, size = 3.5, color = "white")
```

- plot 을 보게 되면 21 년 1 월에 데이터가 몰려 있는 것을 확인할 수 있습니다. 이를 위해 21 년 1 월부터로 기간을 다시 설정하였습니다.

```
# 시간에 따른 게시글 개수  (check 1)
sb1 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20190301)) %>%  # 2 년전 기준
  ggplot(aes(date))+
  geom_freqpoly(binwidth = 10*86400)
```




```
# 21 년 기준
sb1 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20210101)) %>%
  ggplot(aes(date))+
  geom_freqpoly(binwidth = 10*86400)
```

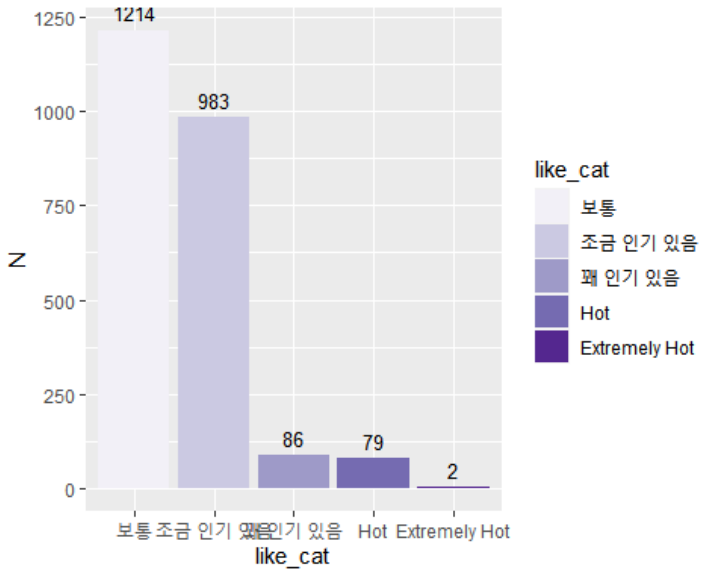


- 중앙값이 38 로써 종로구의 데이터보단 높지만 여전히 인기가 있다고 보기 어려우므로 평균인 99.39 로 기준을 설정하였습니다.

```r
as.data.table(sb1)[, summary(like,na.rm = TRUE)]
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   18.00   38.00   99.39   85.00 5994.00
```

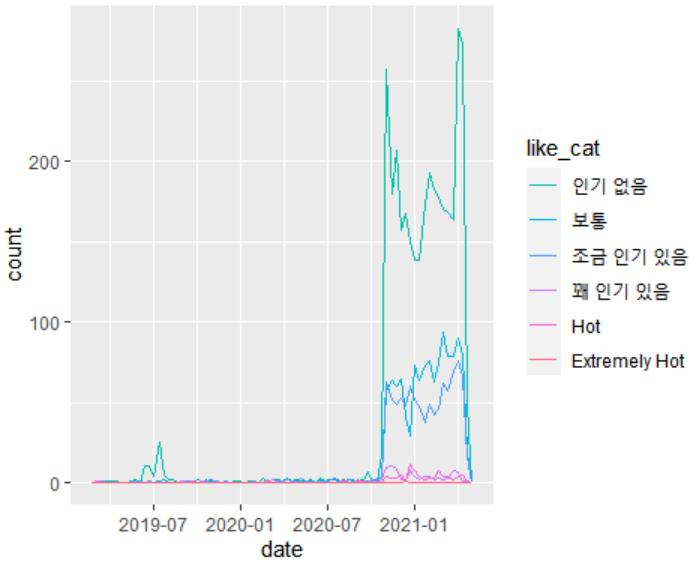
```r
sb2 = sb1 %>%
  mutate(like_cat = ifelse(like<50, '인기 없음',
                           ifelse(like<100, '보통',
                                  ifelse(like<500, '조금 인기 있음',
                                         ifelse(like<1000, '꽤 인기 있음',
                                                ifelse(like<5000, 'Hot','Extr
emely Hot')))))) %>%
  select(date,like,like_cat,place,content,tags,comments) %>%
  arrange(desc(like), like_cat, date, place, content, tags, comments)


sb2 = sb2 %>%
  mutate(like_cat = factor(like_cat, levels = c('인기 없음','보통','조금 인기
있음', '꽤 인기 있음','Hot','Extremely Hot')))

# like_cat bar plot
as.data.table(sb2)[, .(.N), by='like_cat'] %>%
  filter(like_cat!='인기 없음' & !is.na(like_cat)) %>%
  ggplot(aes(x=like_cat, y=N, fill=like_cat))+
  geom_bar(stat='identity')+
  scale_fill_brewer(palette = "Purples",direction = 1)+
  geom_text(aes(label = N), vjust = -0.6, size = 3.5, color = "black")
```



```
# like_cat 별 좋아요 개수 평균
as.data.table(sb2)[, mean(like, na.rm = TRUE), by='like_cat']

[!is.na(like_cat)] %>%
  mutate(mean = V1) %>% select(-c(V1))

##          like_cat       mean
## 1:  Extremely Hot 5563.00000
## 2:            Hot 1926.21519
## 3:   꽤 인기 있음  665.54651
## 4: 조금 인기 있음  193.78637
## 5:           보통   70.84267
## 6:      인기 없음   21.66845
```

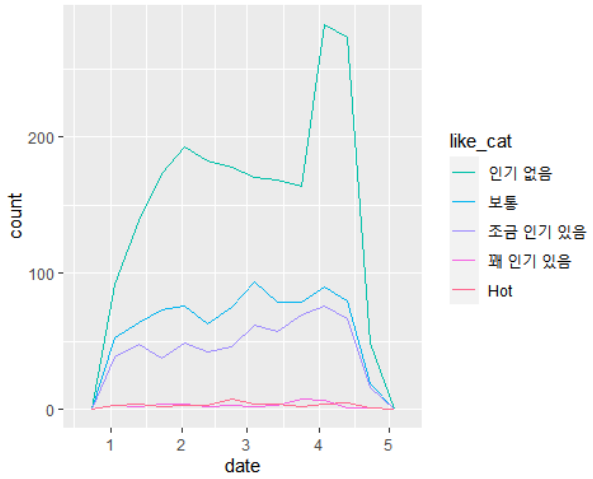
- 좋아요 카테고리 별로 다시 확인해보았습니다.

```
# 시간에 따른 게시글 개수 (좋아요 카테고리에 따른) (check 2)
sb2 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20190301)& !is.na(like_cat)) %>%
  ggplot(aes(date)) + geom_freqpoly(aes(group = like_cat, colour = like_cat),
binwidth = 10*86400) +
  scale_colour_hue(h = c(180, 360))
```

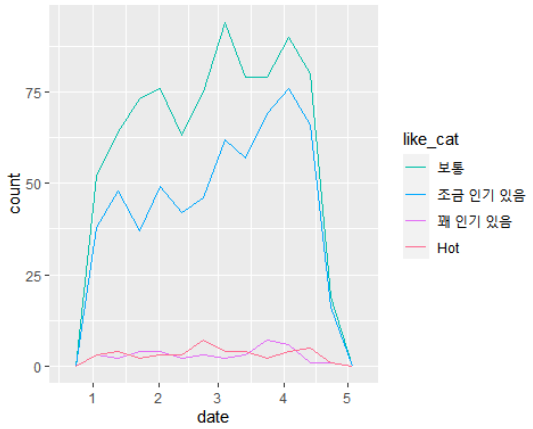


- 21 년 기준 좋아요 카테고리 별 plot
- 확실히 인기 순에 따라 '인기 없음'의 빈도가 가장 높고 'HOT'의 빈도가 가장 낮은 것을 확인할 수 있습니다.
- 성북구의 경우, 19 년 3 월 이후의 Extremely Hot 데이터는 존재하지 않는 것을 확인할 수 있습니다.

```
# 21 년 기준
sb2 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter(date > ymd(20210101)& !is.na(like_cat)) %>%
  ggplot(aes(date)) + geom_freqpoly(aes(group = like_cat, colour = like_cat),
binwidth = 10*86400) +
  scale_colour_hue(h = c(180, 360))
```




```
# 좋아요 카테고리가 보통이상인 데이터에 대해선? (21 년 기준)
sb2 %>%
  mutate(date = as.character(date)) %>%
  separate(date,c('year','month','day')) %>%
  mutate(year = as.integer(year),
         month = as.integer(month),
         day = as.integer(day)) %>%
  mutate(date = make_datetime(year,month,day)) %>%
  filter( (date > ymd(20210101)) & (like >= 50) ) %>%
  ggplot(aes(date)) + geom_freqpoly(aes(group = like_cat, colour = like_cat),
binwidth = 10*86400) +
  scale_colour_hue(h = c(180, 360))
```




```
# 최소 좋아요가 100 개 이상인 데이터만!
sb3 = sb2 %>% filter((like_cat !='인기 없음') & (like_cat !='보통'))

nrow(sb3) # 1150
```

```
## [1] 1150
```

```
# 댓글이 아예 없는 데이터 제거 (댓글이 최소 1 개인 데이터)
sb4 = sb3 %>% filter(comments != '[]')
nrow(sb4) # 1066
```

```
## [1] 1066
```

```
# 장소별 count
dim(sb2) # 5736
```

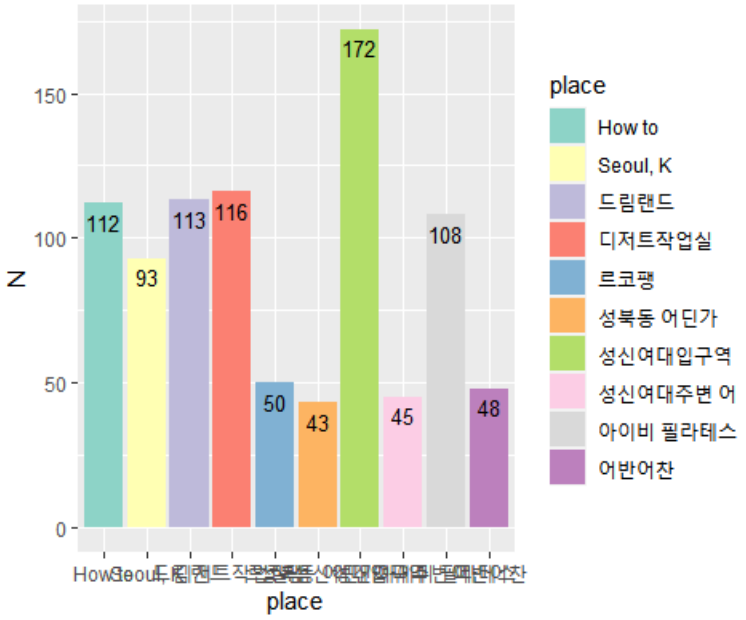
```
## [1] 5736    7
```

```
sb_count_place = as.data.table(sb2)[, .(.N), by='place'][order(-N)]
sb_count_place = sb_count_place %>% mutate(place = str_sub(place,1,8) )
sb_count_place[1:10] # top 10 # N = 43
```

```
##              place    N
##  1:           <NA> 2992
##  2:  성신여대입구역  172
##  3:    디저트작업실  116
##  4:        드림랜드  113
##  5:          How to  112
##  6: 아이비 필라테스  108
##  7:        Seoul, K   93
##  8:          르코팽   50
```

```
##  9:         어반어찬   48
## 10: 성신여대주변 어   45
```

```
# visualization # 좋아요 수 or 댓글 수와 상관없이
sb_count_place %>%
  filter(N>42 & !is.na(place)) %>%
  ggplot(aes(x=place, y=N, fill=place))+
  geom_bar(stat = 'identity')+
  scale_fill_brewer(palette = "Set3")+
  geom_text(aes(label = N), vjust = 1.5, size = 3.5, color = "black")
```

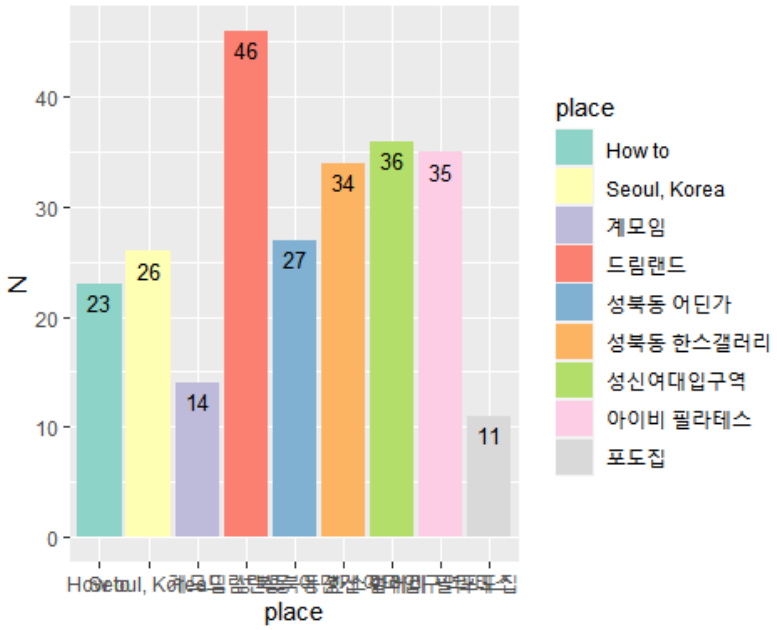



```
# 좋아요 수가 100 개 이상인 1150 개의 데이터 (sb3)
sb_count_place_100 = as.data.table(sb3)[, .(.N), by='place'][order(-N)]

# 289 개의 장소
sb_count_place_100[1:10] # top 10 # N = 8
```

```
##                 place   N
##  1:              <NA> 457
##  2:          드림랜드  46
##  3:    성신여대입구역  36
##  4:   아이비 필라테스  35
##  5: 성북동 한스갤러리  34
```

```
##  6:     성북동 어딘가  27
##  7:      Seoul, Korea  26
##  8:            How to  23
##  9:            계모임  14
## 10:            포도집  11
```

```
sb_count_place_100 %>%
  filter(N>9 & !is.na(place)) %>%
  ggplot(aes(x=place, y=N, fill=place))+
  geom_bar(stat = 'identity')+
  scale_fill_brewer(palette = "Set3")+
  geom_text(aes(label = N), vjust = 1.5, size = 3.5, color = "black")
```

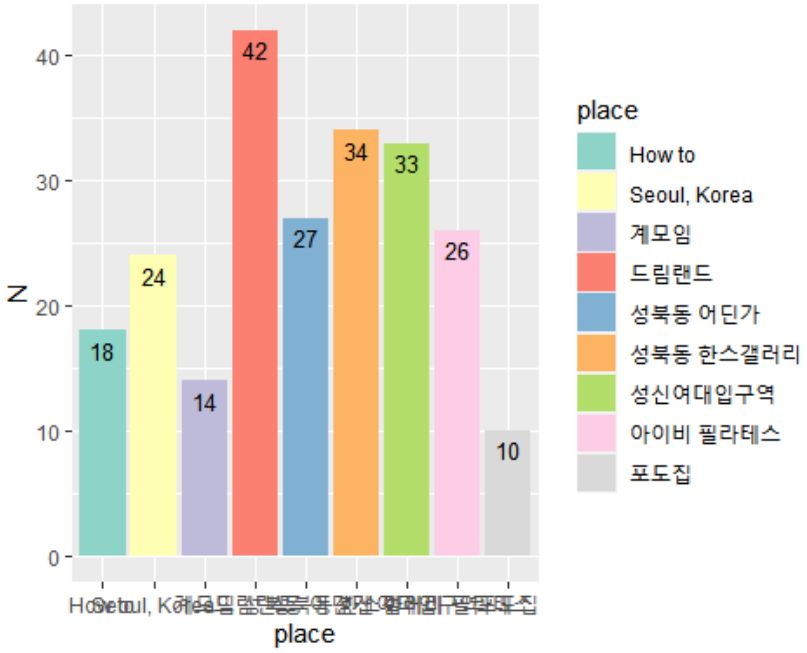


- 성북구의 경우, 댓글이 없는 경우와 댓글이 있는 경우의 장소는 동일하게 9 개 나옵니다.
- 하지만, 댓글 없이 좋아요만 100 개 이상일 경우, '아이비 필라테스'가 3 위로 성북동 ''한스갤러리'보다 순위가 높게 되지만, 댓글을 고려할 경우에는 '아이비 필라테스'가 5 위로 밀려나면서 4 위인 성북동 어딘가보다 낮은 순위를 기록하게 되는 차이가 존재합니다. 이를 통해 광고성이 짙은 게시물의 경우 댓글 없이 좋아요 개수만 많이 존재한다고 생각할 수 있습니다. 또한 댓글 중 음식관련 단어가 들어가는 게시물만 골라낸다면 '아이비 필라테스'와 같이 전혀 무관한 게시물을 걸러낼 수 있을 것이므로 이를 고려할 예정입니다.

```r
# 좋아요 100 개 이상이고 댓글이 1 개라도 있는 1066 개의 데이터 (sb4)
sb_count_place_100_C = as.data.table(sb4)[, .(.N), by='place'][order(-N)]

# 272 개의 장소
sb_count_place_100_C[1:10] # top 10 # N = 8
```

```
##                 place   N
##  1:              <NA> 421
##  2:          드림랜드  42
##  3: 성북동 한스갤러리  34
##  4:    성신여대입구역  33
##  5:     성북동 어딘가  27
##  6:   아이비 필라테스  26
##  7:      Seoul, Korea  24
##  8:            How to  18
##  9:            계모임  14
## 10:            포도집  10
```

```r
sb_count_place_100_C %>%
  filter(N>9 & !is.na(place)) %>%
  ggplot(aes(x=place, y=N, fill=place))+
  geom_bar(stat = 'identity')+
  scale_fill_brewer(palette = "Set3")+
  geom_text(aes(label = N), vjust = 1.5, size = 3.5, color = "black"
```



- Graphical Analysis Results
  - 좋아요 수가 100 개 이상이고 댓글이 1 개 이상인 음식점을 비교적 인기 있는 가게라고 판단하였습니다. 이 두 가지가 인기 지표로서 충분하지 않다고 판단하여 데이터의 빈도 수를 추가로 고려해본 것입니다.

  - 광고 게시글로 인해 좋아요가 무조건 많다고 해서 진정한 맛집이라고 판단하기 어렵습니다. 하지만 빈도 수까지 높은 데이터라면 맛집으로 판단할 가능성이 있어보인다고 판단하였습니다. 물론, 장소 변수가 결측치가 아닌 관측치만 고려해본 것이라 한계는 존재합니다. 이를 개선하기 위해 최종 프로젝트에서는 다른 변수들을 활용하여 장소 결측치를 대체하는 작업을 고려할 예정입니다. 아래는 종로구와 성북구의 핫플레이스 결과입니다.

  - 종로구 : 1538 개의 데이터 중 NA 를 제외하고 502 개의 장소가 존재했으며 빈도가 가장 높은 Top 10 게시물의 장소를 확인해본 결과, 구기동이 1 위로 50 개의 게시물이 존재하나, 구기동은 지명주소입니다. 구기동에 인기가 있는 맛집이 많을 수 있겠다는 가능성은 열어놓고 추후 분석할 예정입니다. 지명과 가게명으로 분류를 해보면 지명의 경우 50 개의 빈도를 지닌 구기동이 1 위, 35 개의 빈도를 지닌 대학로/혜화가 2 위, 28 개의 빈도를 지닌 익선동이 3 위, 9 개의 빈도를 지닌 Seoul,Korea 가 4 위입니다. Seoul,Korea 는 중요하지 않은 데이터로 판단되므로 인스타 데이터를 기반으로 했을 때, 종로구는 구기동 > 대학로/혜화 > 익선동 순으로 핫플레이스가 존재한다고 보여집니다.

    이번엔 가게명의 경우 '칠린','칸다소바','광장시장','키즈나','이태리재'가 존재합니다. 39 개의 빈도를 지닌 2 위 '칠린'은 종로구 명륜 4 가에 위치한 칵테일바입니다. 11 개의 빈도를 지닌 6 위 '칸다소바'는 종로구 대학로에 위치한 일본식라멘집입니다. 사실 '칸다소바'같은 경우, 혜화역 뿐만 아니라 경복궁역 근처에도 지점이 존재합니다. 이런 경우를 확실히 하기 위해 tags 변수와의 관계를 파악하여 최종 프로젝트에 반영할 예정입니다. 9 개의 빈도를 지닌 '광장시장'은 종로구 창경궁로에 위치한 먹거리시장입니다. 8 개의 빈도를 지닌 '키즈나'은 종로구 안국역 부근에 위치한 일식당입니다. 마지막으로 8 개의 빈도를 지닌 '이태리재'는 서울 2021 미쉐린 가이드에 선정된 이탈리아 음식점입니다. 위 5 개의 가게에 대한 네이버 방문자 리뷰 평점은 5 점 만점, 21/04/21 기준 각각 (4.43 / 4.44 / 4.48 / 4.71 / 4.31)으로 모두 나쁘지 않은 평점을 보유하고 있다는 점을 확인할 수 있습니다.

- 성북구 : 1066 개의 데이터 중 NA 를 제외하고 271 개의 장소가 존재했으며 빈도가 가장 높은 Top 10 게시물의 장소를 확인해본 결과, NA 제외 기준 드림랜드가 1 위로 42 개의 게시물이 존재합니다. 드림랜드는 성북구 월계로에 위치한 종합분식점으로 네이버평점 4.28 점입니다. 2 위인 한스갤러리는 성북구 정릉로(북악산)에 위치한 양식전문점으로 네이버평점 4.44 점입니다. 3 위인 성신여대입구역은 지명으로 33 개의 게시물이 존재합니다. 4 위인 성북동 어딘가처럼 SNS 이용자들이 장소를 정확하게 표기하지 않는 경우가 존재하므로 주의해야합니다. 5 위인 아이비 필라테스의 경우 음식점이 아닌데도 불구하고 26 개의 게시물이나 존재하는 것을 확인할 수 있습니다. 이처럼 크롤링 과정에서 광고 게시물을 수집할 가능성이 크기 때문에 추후 tags 와 comments 변수를 활용하여 이러한 데이터는 제외시킬 것입니다. 8 위인 계모임은 성북구 보문로에 위치한 닭요리 전문점으로, 네이버평점 4.62 점입니다. 마지막으로 포도집은 성북구 정릉로에 위치한 양식 전문점으로, 네이버평점 4.55 점입니다.
  마찬가지로 모두 나쁘지 않은 평점을 보유하고 있는 것으로 봐서 15000 여개의 데이터를 어느정도 필터링했다고 판단됩니다. 최종 프로젝트에서는 이를 참고하여 더 탁월한 맛집을 찾아낼 예정입니다.