gig_platform_log

- readr::read_csv : 대용량 csv 파일 읽어들일 때 좋음
- stringr::str_detect : 문자열 찾는 함수
- TTR::SMA : 시계열 smoothing 함수

Data

```
conv <- read_csv('C:/data/gig/conversion.csv')
```

```
## Parsed with column specification:
## cols(
##   eventcategory = col_character(),
##   isfirstactivity = col_logical(),
##   apppackagename = col_character(),
##   appversion = col_character(),
##   devicetype = col_character(),
##   devicemanufacturer = col_character(),
##   osversion = col_character(),
##   canonicaldeviceuuid = col_character(),
##   sourcetype = col_character(),
##   channel = col_character(),
##   params_campaign = col_character(),
##   params_medium = col_character(),
##   params_term = col_character(),
##   inappeventcategory = col_character(),
##   inappeventlabel = col_double(),
##   eventdatetime = col_datetime(format = ""),
##   rowuuid = col_character(),
##   isfirstgoalactivity = col_logical(),
##   event_rank = col_logical()
## )
```

```
attr(conv, 'spec') <- NULL
funnel <- read_csv('C:/data/gig/funnel.csv')
```

```
## Parsed with column specification:
## cols(
##   Lv2 = col_double(),
##   viewid = col_character(),
##   `viewid desc` = col_character(),
##   Lv1 = col_double(),
##   `funnel name` = col_character(),
##   `funnel desc` = col_character()
## )
```

```
attr(funnel, 'spec') <- NULL
category <- read_csv('C:/data/gig/category.csv')
```

```
## Parsed with column specification:
## cols(
##   depth = col_double(),
##   categoryid = col_double(),
##   categoryname = col_character(),
##   cat1_id = col_double(),
##   cat2_id = col_double(),
##   cat3_id = col_double(),
##   cat1 = col_character(),
##   cat2 = col_character(),
##   cat3 = col_character()
## )
```

```
attr(category, 'spec') <- NULL
```

```
head(conv)
```

```
## # A tibble: 6 x 19
##   eventcategory isfirstactivity apppackagename appversion devicetype
##   <chr>         <lgl>           <chr>          <chr>      <chr>
## 1 goal          FALSE           com.kmong.iOS  4.0.4      iPhone
## 2 goal          FALSE           com.kmong.kmo~ 3.3.5      SM-N935S
## 3 goal          FALSE           com.kmong.iOS  4.0.4      iPhone
## 4 foreground    NA              com.kmong.iOS  4.0.4      iPhone
## 5 goal          FALSE           com.kmong.iOS  4.0.4      iPhone
## 6 goal          FALSE           com.kmong.kmo~ 3.3.5      SM-G955N
## # ... with 14 more variables: devicemanufacturer <chr>, osversion <chr>,
## #   canonicaldeviceuuid <chr>, sourcetype <chr>, channel <chr>,
## #   params_campaign <chr>, params_medium <chr>, params_term <chr>,
## #   inappeventcategory <chr>, inappeventlabel <dbl>, eventdatetime <dttm>,
## #   rowuuid <chr>, isfirstgoalactivity <lgl>, event_rank <lgl>
```

```
str(conv)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 434244 obs. of  19 variables:
##  $ eventcategory      : chr  "goal" "goal" "goal" "foreground" ...
##  $ isfirstactivity    : logi  FALSE FALSE FALSE NA FALSE FALSE ...
##  $ apppackagename     : chr  "com.kmong.iOS" "com.kmong.kmong" "com.kmong.iOS" "com.kmong.iOS" ...
##  $ appversion         : chr  "4.0.4" "3.3.5" "4.0.4" "4.0.4" ...
##  $ devicetype         : chr  "iPhone" "SM-N935S" "iPhone" "iPhone" ...
##  $ devicemanufacturer : chr  "Apple" "samsung" "Apple" "Apple" ...
##  $ osversion          : chr  "iOS11.4.1" "Android7.0" "iOS12.0" "iOS11.4.1" ...
##  $ canonicaldeviceuuid: chr  "F36FAA62-ADAC-4AA5-9B00-1FD6CB7EE957" "8a871e50-0717-4aed-9bad-04ac3c3
##  $ sourcetype         : chr  "unattributed" "unattributed" "unattributed" NA ...
##  $ channel            : chr  "unattributed" "unattributed" "unattributed" NA ...
##  $ params_campaign    : chr  NA NA NA NA ...
##  $ params_medium      : chr  NA NA NA NA ...
##  $ params_term        : chr  NA NA NA NA ...
##  $ inappeventcategory : chr  "home.view" "gig_detail.view" "inbox_detail.view" NA ...
##  $ inappeventlabel    : num  NA 41201 NA NA NA ...
##  $ eventdatetime      : POSIXct, format: "2018-09-27 15:00:00" "2018-09-27 15:00:00" ...
##  $ rowuuid            : chr  "fd2a188c-bc9b-4702-9c47-b546b2614817" "e62dccef-dd70-4415-8a33-c8324dd
##  $ isfirstgoalactivity: logi  FALSE FALSE FALSE NA FALSE FALSE ...
##  $ event_rank         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
```

```
head(funnel)
```

```
## # A tibble: 6 x 6
##     Lv2 viewid        `viewid desc`        Lv1 `funnel name` `funnel desc`
##   <dbl> <chr>         <chr>               <dbl> <chr>         <chr>
## 1  1100 home           ( )                11 home
## 2  1210 category_list      ( )    12 category
## 3  1200 category_gig    -       12 category
## 4  1300 search                            13 search
## 5  1301 search_gig     -           13 search
## 6  1302 search_seller  -            13 search
```

```
str(funnel)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 53 obs. of  6 variables:
##  $ Lv2        : num  1100 1210 1200 1300 1301 ...
##  $ viewid     : chr  "home" "category_list" "category_gig" "search" ...
##  $ viewid desc: chr  "  ( )" "      ( )" "  -   " " " ...
##  $ Lv1        : num  11 12 12 13 13 13 14 14 14 15 ...
##  $ funnel name: chr  "home" "category" "category" "search" ...
##  $ funnel desc: chr  " " "   " "   " " " ...
```

```
head(category)
```

```
## # A tibble: 6 x 9
##   depth categoryid categoryname  cat1_id cat2_id cat3_id cat1        cat2  cat3
##   <dbl>      <dbl> <chr>           <dbl>   <dbl>   <dbl> <chr>       <chr> <chr>
## 1     1          1                  1      NA      NA         <NA>  <NA>
## 2     1          2                  2      NA      NA         <NA>  <NA>
## 3     1          3  ·            3      NA      NA  ·    <NA>  <NA>
## 4     1          4                  4      NA      NA        <NA>  <NA>
## 5     1          6 IT·          6      NA      NA IT·   ~ <NA>  <NA>
## 6     1          7              7      NA      NA      <NA>  <NA>
```

```
str(category)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 245 obs. of  9 variables:
##  $ depth       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ categoryid  : num  1 2 3 4 6 ...
##  $ categoryname: chr  "  " "  " "  " . " "  " ...
##  $ cat1_id     : num  1 2 3 4 6 7 9 10 11 99 ...
##  $ cat2_id     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cat3_id     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cat1        : chr  "  " "  " "  " · " "  " ...
##  $ cat2        : chr  NA NA NA NA ...
##  $ cat3        : chr  NA NA NA NA ...
```

Q1 로그 정렬

```
colnames(conv)[colnames(conv) == 'canonicaldeviceuuid'] <- 'user_id'
user <- conv %>% group_by(user_id, rowuuid)
user <- arrange(user, user_id, desc(eventdatetime))
```

R에서는 group_by가 큰 의미가 없어 보인다.
sort가 오래 걸리는 함수이니까 미리 정렬해놓고 merge하는 것이 나을 것 같다.

column 정리

```
user$inappeventcategory[user$inappeventcategory == ""] <- 'nothing.view'
user$inappeventcategory[is.na(user$inappeventcategory)] <- 'nothing.nothing'
split_point <- regexpr('[.]', user$inappeventcategory) %>% as.vector()
user$viewid <- substr(user$inappeventcategory, 1, split_point-1)
user$viewaction <- substr(user$inappeventcategory, split_point+1, nchar(user$inappeventcategory))
```

inappeventcategory를 '.'을 기준으로 viewid와 viewaction으로 나눈다.

```
colnames(user)[colnames(user) == 'inappeventlabel'] <-'categoryid'
```

category 데이터와 컬럼명을 동일하게 변경한다.

merge

```
log <- merge(user, funnel, by = 'viewid', all.x = T)
log <- merge(log, category, by = 'categoryid', all.x = T)
str(log)
```

```
## 'data.frame':    434244 obs. of  34 variables:
##  $ categoryid         : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ viewid             : chr  "category_gig" "category_gig" "category_gig" "category_gig" ...
##  $ eventcategory      : chr  "goal" "goal" "goal" "goal" ...
##  $ isfirstactivity    : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
##  $ apppackagename     : chr  "com.kmong.iOS" "com.kmong.iOS" "com.kmong.iOS" "com.kmong.iOS" ...
##  $ appversion         : chr  "4.0.4" "4.0.4" "4.0.4" "4.0.4" ...
##  $ devicetype         : chr  "iPhone" "iPhone" "iPhone" "iPhone" ...
##  $ devicemanufacturer : chr  "Apple" "Apple" "Apple" "Apple" ...
##  $ osversion          : chr  "iOS12.0" "iOS11.2.6" "iOS12.0" "iOS11.4.1" ...
##  $ user_id            : chr  "BBE5870B-D697-467B-807C-1A097107B8F1" "D20ACEB3-AAE0-4FEB-B3D3-43B0FE1
##  $ sourcetype         : chr  "unattributed" "viral" "viral" "unattributed" ...
##  $ channel            : chr  "unattributed" "WEB" "WEB" "unattributed" ...
##  $ params_campaign    : chr  NA NA NA NA ...
##  $ params_medium      : chr  NA NA NA NA ...
##  $ params_term        : chr  NA NA NA NA ...
##  $ inappeventcategory : chr  "category_gig.view" "category_gig.view" "category_gig.view" "category_g
##  $ eventdatetime      : POSIXct, format: "2018-09-28 07:50:06" "2018-09-28 16:35:35" ...
##  $ rowuuid            : chr  "73c6a0ad-b5d4-47cf-9e08-bab913964cd7" "b74ce4e0-6923-493c-9b14-d6a736f
##  $ isfirstgoalactivity: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ event_rank         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ viewaction         : chr  "view" "view" "view" "view" ...
##  $ Lv2                : num  1200 1200 1200 1200 1200 1200 1200 1200 1200 1200 ...
##  $ viewid desc        : chr  "  -  " "  -  " "  -  " "  -  " ...
##  $ Lv1                : num  12 12 12 12 12 12 12 12 12 12 ...
##  $ funnel name        : chr  "category" "category" "category" "category" ...
##  $ funnel desc        : chr  "  " "  " "  " "  " ...
```

```
##  $ depth              : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ categoryname       : chr  "  " "  " "  " "  " "  " ...
##  $ cat1_id            : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ cat2_id            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cat3_id            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cat1               : chr  "  " "  " "  " "  " "  " ...
##  $ cat2               : chr  NA NA NA NA ...
##  $ cat3               : chr  NA NA NA NA ...
```

funnel 데이터, category 데이터와 merge한다.

```
log$ostype[str_detect(log$osversion, 'iOS')] <- 'iOS'
log$ostype[str_detect(log$osversion, 'Android')] <- 'Android'
log$osversion <- sub('iOS|Android','', log$osversion)
```

osversion을 ostype과 osversion(= version number)으로 나눈다.

```
colnames(log)[colnames(log) == 'funnel desc'] <- 'funnel_desc'
colnames(log)[colnames(log) == 'funnel name'] <- 'funnel_name'
colnames(log)[colnames(log) == 'viewid desc'] <- 'viewid_desc'
```

컬렴명 사이에 여백이 있어 사용하기 어렵다.
연결해주자.

```
colnames(log)
```

```
##  [1] "categoryid"          "viewid"              "eventcategory"
##  [4] "isfirstactivity"     "apppackagename"      "appversion"
##  [7] "devicetype"          "devicemanufacturer"  "osversion"
## [10] "user_id"             "sourcetype"          "channel"
## [13] "params_campaign"     "params_medium"       "params_term"
## [16] "inappeventcategory"  "eventdatetime"       "rowuuid"
## [19] "isfirstgoalactivity" "event_rank"          "viewaction"
## [22] "Lv2"                 "viewid_desc"         "Lv1"
## [25] "funnel_name"         "funnel_desc"         "depth"
## [28] "categoryname"        "cat1_id"             "cat2_id"
## [31] "cat3_id"             "cat1"                "cat2"
## [34] "cat3"                "ostype"
```

```
log <- log[c(10, 18, 17, 3, 25, 26, 2, 21, 23, 24,
             22, 4, 19, 7, 8, 35, 9, 6, 5, 11,
             12, 1, 28, 27, 32, 29, 33, 30, 34, 31,
             14, 13, 15, 16, 20)]
str(log)
```

```
## 'data.frame':    434244 obs. of  35 variables:
##  $ user_id            : chr  "BBE5870B-D697-467B-807C-1A097107B8F1" "D20ACEB3-AAE0-4FEB-B3D3-43B0FE1
##  $ rowuuid            : chr  "73c6a0ad-b5d4-47cf-9e08-bab913964cd7" "b74ce4e0-6923-493c-9b14-d6a736f
##  $ eventdatetime      : POSIXct, format: "2018-09-28 07:50:06" "2018-09-28 16:35:35" ...
##  $ eventcategory      : chr  "goal" "goal" "goal" "goal" ...
##  $ funnel_name        : chr  "category" "category" "category" "category" ...
```

```
##  $ funnel_desc        : chr  "  " "  " "  " "  " ...
##  $ viewid             : chr  "category_gig" "category_gig" "category_gig" "category_gig" ...
##  $ viewaction         : chr  "view" "view" "view" "view" ...
##  $ viewid_desc        : chr  "  -  " "  -  " "  -  " "  -  " ...
##  $ Lv1                : num  12 12 12 12 12 12 12 12 12 12 ...
##  $ Lv2                : num  1200 1200 1200 1200 1200 1200 1200 1200 1200 1200 ...
##  $ isfirstactivity    : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
##  $ isfirstgoalactivity: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ devicetype         : chr  "iPhone" "iPhone" "iPhone" "iPhone" ...
##  $ devicemanufacturer : chr  "Apple" "Apple" "Apple" "Apple" ...
##  $ ostype             : chr  "iOS" "iOS" "iOS" "iOS" ...
##  $ osversion          : chr  "12.0" "11.2.6" "12.0" "11.4.1" ...
##  $ appversion         : chr  "4.0.4" "4.0.4" "4.0.4" "4.0.4" ...
##  $ apppackagename     : chr  "com.kmong.iOS" "com.kmong.iOS" "com.kmong.iOS" "com.kmong.iOS" ...
##  $ sourcetype         : chr  "unattributed" "viral" "viral" "unattributed" ...
##  $ channel            : chr  "unattributed" "WEB" "WEB" "unattributed" ...
##  $ categoryid         : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ categoryname       : chr  "  " "  " "  " "  " "  " ...
##  $ depth              : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ cat1               : chr  "  " "  " "  " "  " "  " ...
##  $ cat1_id            : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ cat2               : chr  NA NA NA NA ...
##  $ cat2_id            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cat3               : chr  NA NA NA NA ...
##  $ cat3_id            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ params_medium      : chr  NA NA NA NA ...
##  $ params_campaign    : chr  NA NA NA NA ...
##  $ params_term        : chr  NA NA NA NA ...
##  $ inappeventcategory : chr  "category_gig.view" "category_gig.view" "category_gig.view" "category_gi
##  $ event_rank         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
```

자주 사용할 컬럼 위주로 재배열해보자.
더 쉬운 방법이 없나 고민해보자.
컬럼이 많아서 일일이 하기에 비효율적이다.

```
log[log$user_id == 'fff40190-d751-425a-9813-2284072e47f5',
    colnames(log) %in% c('eventdatetime', 'eventcategory', 'viewid', 'viewaction', 'funnel_desc', 'viewi
```

```
##              eventdatetime   eventcategory funnel_desc       viewid viewaction
## 192835 2018-09-27 23:50:22            goal        inbox_detail       view
## 210208 2018-09-28 00:02:41            goal        inbox_detail       view
## 210330 2018-09-28 00:03:16            goal        inbox_detail       view
## 211350 2018-09-27 23:51:08            goal        inbox_detail       view
## 316749 2018-09-27 23:51:25      background        <NA>      nothing    nothing
## 316751 2018-09-27 23:50:22          launch        <NA>      nothing    nothing
## 421529 2018-09-28 00:02:41 launchInSession        <NA>      nothing    nothing
## 424780 2018-09-28 00:03:19      background        <NA>      nothing    nothing
##            viewid_desc
## 192835    -
## 210208    -
## 210330    -
## 211350    -
## 316749            <NA>
```

```
## 316751            <NA>
## 421529            <NA>
## 424780            <NA>
```

```
log[log$user_id == 'fff40190-d751-425a-9813-2284072e47f5' & log$viewid %in% c("home", "inbox_detail", "
```

```
##                                   user_id
## 192835 fff40190-d751-425a-9813-2284072e47f5
## 210208 fff40190-d751-425a-9813-2284072e47f5
## 210330 fff40190-d751-425a-9813-2284072e47f5
## 211350 fff40190-d751-425a-9813-2284072e47f5
##                                   rowuuid       eventdatetime eventcategory
## 192835 a773ad20-fb9c-4c6d-96fd-1861dec0cea8 2018-09-27 23:50:22          goal
## 210208 457da225-eaf2-4ffe-94e0-d3ffb7b93490 2018-09-28 00:02:41          goal
## 210330 f55223e0-52f5-41c2-8f0b-41c562baa3fc 2018-09-28 00:03:16          goal
## 211350 a9d0ac01-b715-46f2-b92d-8570dfbc2cca 2018-09-27 23:51:08          goal
##        funnel_name funnel_desc       viewid viewaction     viewid_desc Lv1  Lv2
## 192835       inbox         inbox_detail       view    -   16 1610
## 210208       inbox         inbox_detail       view    -   16 1610
## 210330       inbox         inbox_detail       view    -   16 1610
## 211350       inbox         inbox_detail       view    -   16 1610
##        isfirstactivity isfirstgoalactivity devicetype devicemanufacturer
## 192835           FALSE               FALSE   SM-G950N            samsung
## 210208           FALSE               FALSE   SM-G950N            samsung
## 210330           FALSE               FALSE   SM-G950N            samsung
## 211350           FALSE               FALSE   SM-G950N            samsung
##         ostype osversion appversion  apppackagename   sourcetype      channel
## 192835 Android     8.0.0      3.3.3 com.kmong.kmong unattributed unattributed
## 210208 Android     8.0.0      3.3.3 com.kmong.kmong unattributed unattributed
## 210330 Android     8.0.0      3.3.3 com.kmong.kmong unattributed unattributed
## 211350 Android     8.0.0      3.3.3 com.kmong.kmong unattributed unattributed
##        categoryid categoryname depth cat1 cat1_id cat2 cat2_id cat3 cat3_id
## 192835         NA         <NA>    NA <NA>      NA <NA>      NA <NA>      NA
## 210208         NA         <NA>    NA <NA>      NA <NA>      NA <NA>      NA
## 210330         NA         <NA>    NA <NA>      NA <NA>      NA <NA>      NA
## 211350         NA         <NA>    NA <NA>      NA <NA>      NA <NA>      NA
##        params_medium params_campaign params_term inappeventcategory event_rank
## 192835          <NA>            <NA>        <NA>  inbox_detail.view       TRUE
## 210208          <NA>            <NA>        <NA>  inbox_detail.view       TRUE
## 210330          <NA>            <NA>        <NA>  inbox_detail.view       TRUE
## 211350          <NA>            <NA>        <NA>  inbox_detail.view       TRUE
```

잘 정리되었는지 확인.

function

```
find_log <- function(user_id = c(), columns = c(), viewid = c()){
  if (is.null(user_id) & is.null(columns) & is.null(viewid)) {

    return (log)

  } else if (!is.null(user_id) & is.null(columns) & is.null(viewid)){
```

```
    return (log[log$user_id %in% user_id,])

  } else if (is.null(user_id) & !is.null(columns) & is.null(viewid)) {

    return (log[colnames(log) %in% columns])

  } else if (is.null(user_id) & is.null(columns) & !is.null(viewid)) {

    return (log[log$viewid %in% viewid,])

  } else if (!is.null(user_id) & !is.null(columns) & is.null(viewid)) {

    return (log[log$user_id %in% user_id, colnames(log) %in% columns])

  } else if (!is.null(user_id) & is.null(columns) & !is.null(viewid)) {

    return (log[log$user_id %in% user_id & log$viewid %in% viewid,])

  } else if (is.null(user_id) & !is.null(columns) & !is.null(viewid)) {

    return (log[log$viewid %in% viewid, colnames(log) %in% columns])

  } else {

    return(log[log$user_id %in% user_id & log$viewid %in% viewid, colnames(log) %in% columns])
  }
}
```

로그데이터를 찾는 함수 find_log를 만든다.
좀더 깔끔하고 합리적인 방법이 없을까 고민해봐야 할 것 같다.

```
find_log(user_id = 'fff40190-d751-425a-9813-2284072e47f5' , columns = c('user_id', 'eventcategory', 'dev
```

```
##                                     user_id   eventcategory       viewid
## 192835 fff40190-d751-425a-9813-2284072e47f5           goal inbox_detail
## 210208 fff40190-d751-425a-9813-2284072e47f5           goal inbox_detail
## 210330 fff40190-d751-425a-9813-2284072e47f5           goal inbox_detail
## 211350 fff40190-d751-425a-9813-2284072e47f5           goal inbox_detail
## 316749 fff40190-d751-425a-9813-2284072e47f5     background      nothing
## 316751 fff40190-d751-425a-9813-2284072e47f5         launch      nothing
## 421529 fff40190-d751-425a-9813-2284072e47f5 launchInSession      nothing
## 424780 fff40190-d751-425a-9813-2284072e47f5     background      nothing
##        devicetype
## 192835   SM-G950N
## 210208   SM-G950N
## 210330   SM-G950N
## 211350   SM-G950N
## 316749   SM-G950N
## 316751   SM-G950N
## 421529   SM-G950N
## 424780   SM-G950N
```

함수가 잘 만들어졌나 확인.

Q2 그래프 그리기

```
users <- aggregate(log$rowuuid, list(format(log$eventdatetime, '%Y-%m-%d %H:%M'), log$devicetype,log$os
colnames(users) <- c('eventdatetime', 'devicetype', 'ostype', 'osversion', 'appversion', 'count')
head(users)
```

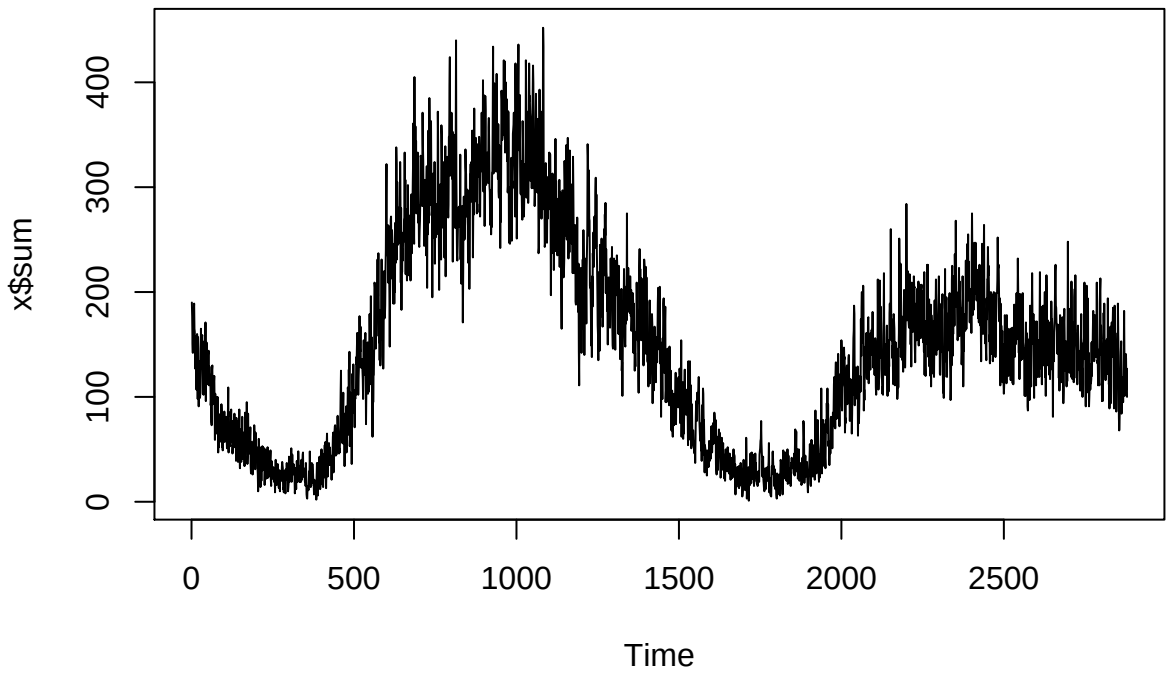
```
##      eventdatetime devicetype  ostype osversion appversion count
## 1 2018-09-28 00:19   SM-A530N Android     8.0.0     2.19.3     1
## 2 2018-09-28 00:20   SM-A530N Android     8.0.0     2.19.3     1
## 3 2018-09-28 22:42   SM-A530N Android     8.0.0     2.19.3     2
## 4 2018-09-28 13:50   SM-G930S Android     8.0.0     2.19.3     4
## 5 2018-09-28 09:11   SM-N920S Android       7.0     2.19.4     1
## 6 2018-09-28 09:16   SM-N920S Android       7.0     2.19.4     1
```

사용자의 주요 특징인 devicetype, ostype, osversion, appversion에 대해서 미리 개수를 세놓는다.
시간 단위로 하기에는 데이터가 너무 개략적으로 나오는 것 같아서 분 단위로 표현한다.

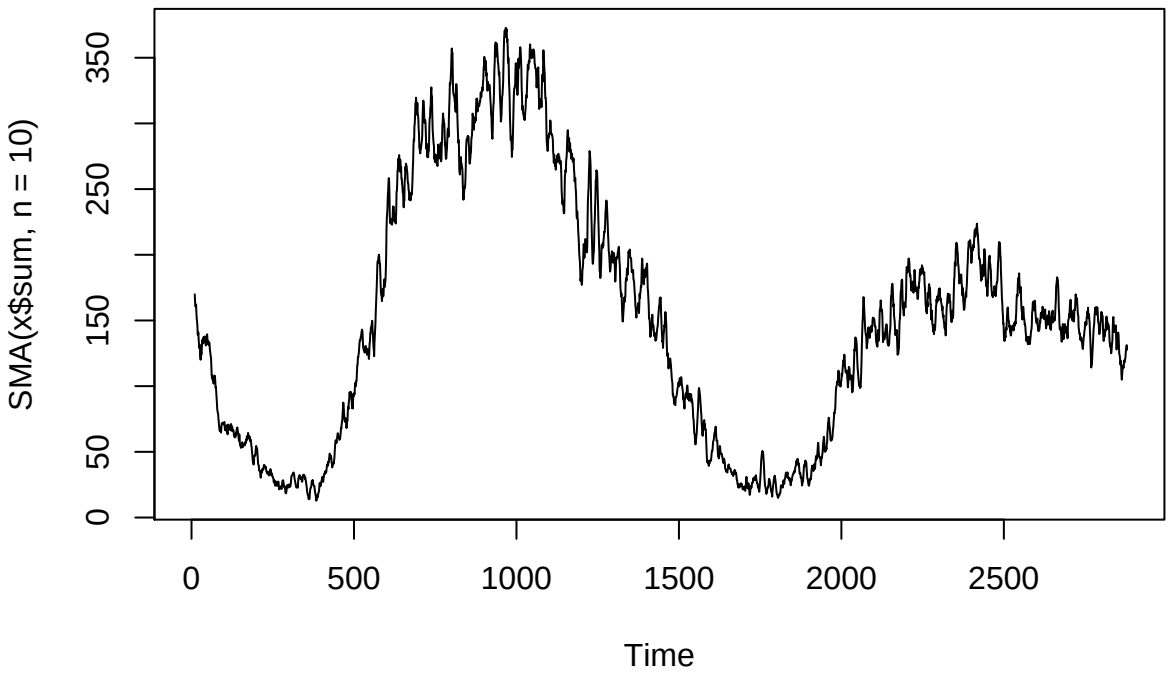
```
x <- users %>% group_by(eventdatetime) %>% summarise(sum = sum(count))
head(x, 20)
```

```
## # A tibble: 20 x 2
##    eventdatetime      sum
##    <chr>            <int>
##  1 2018-09-27 15:00   190
##  2 2018-09-27 15:01   177
##  3 2018-09-27 15:02   151
##  4 2018-09-27 15:03   142
##  5 2018-09-27 15:04   170
##  6 2018-09-27 15:05   171
##  7 2018-09-27 15:06   182
##  8 2018-09-27 15:07   189
##  9 2018-09-27 15:08   165
## 10 2018-09-27 15:09   162
## 11 2018-09-27 15:10   127
## 12 2018-09-27 15:11   148
## 13 2018-09-27 15:12   158
## 14 2018-09-27 15:13   149
## 15 2018-09-27 15:14   107
## 16 2018-09-27 15:15   129
## 17 2018-09-27 15:16   160
## 18 2018-09-27 15:17   156
## 19 2018-09-27 15:18    98
## 20 2018-09-27 15:19   157
```

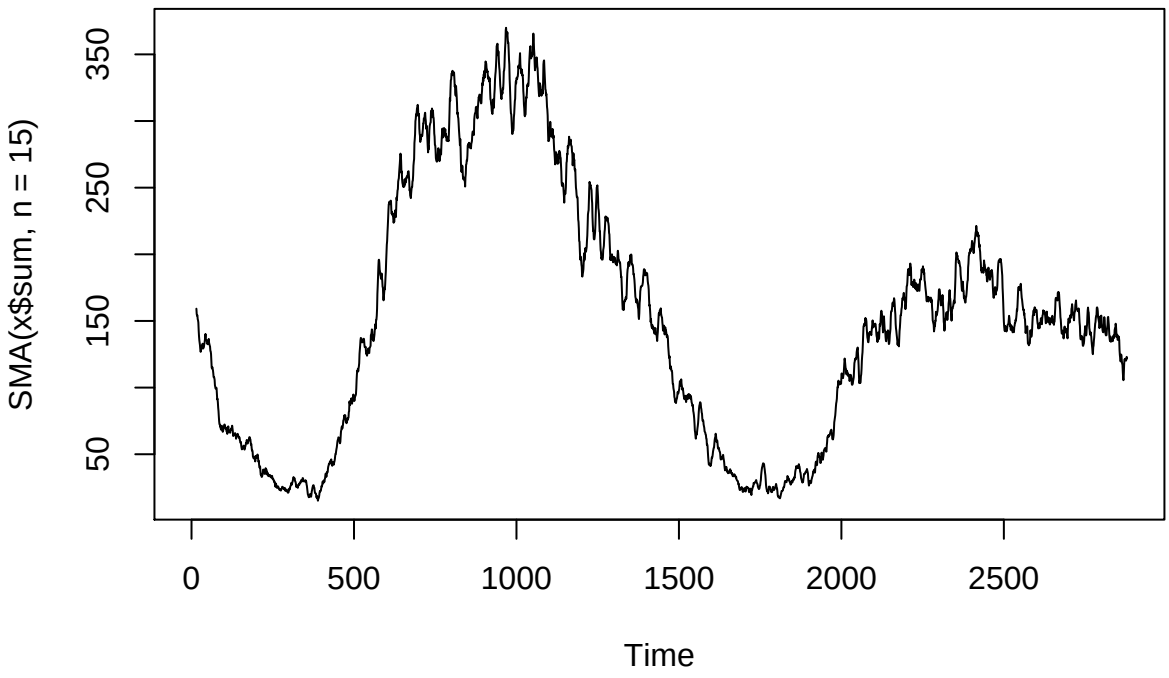
```
plot.ts(x$sum, type = 'l')
```



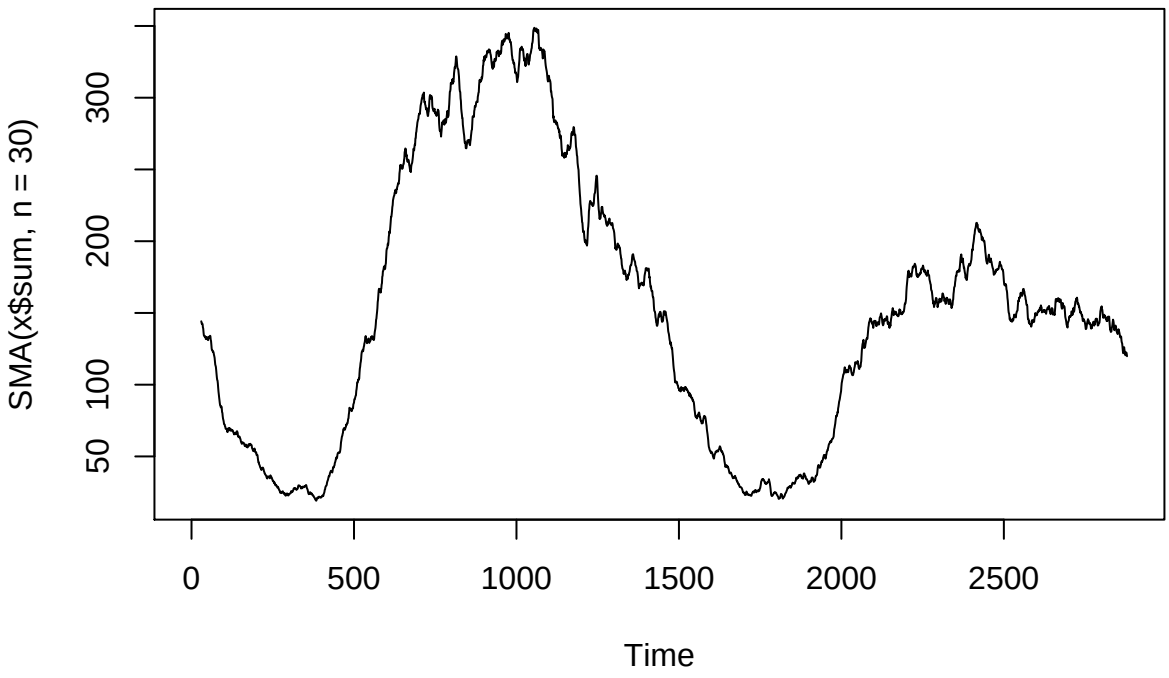

```
plot.ts(SMA(x$sum, n = 10), type = 'l')
```




```
plot.ts(SMA(x$sum, n = 15), type = 'l')
```




```
plot.ts(SMA(x$sum, n = 30), type = 'l')
```



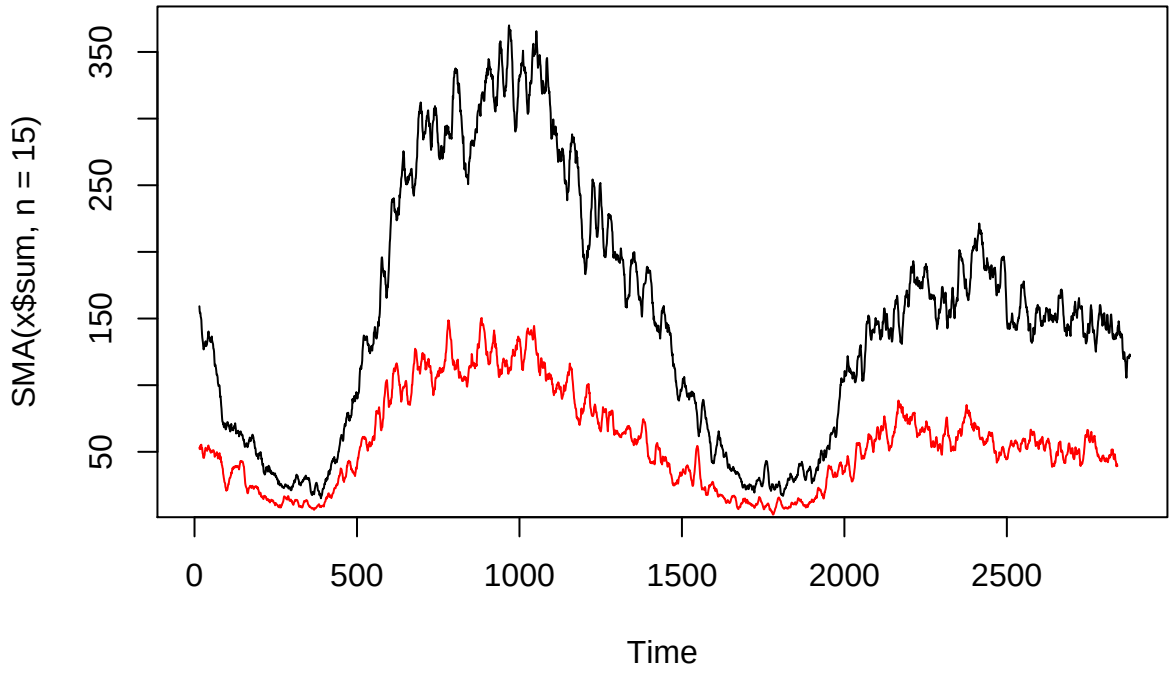
전체에 대해 확인할 때는 eventdatetime에 대해서 그룹핑하여 그래프를 그린다.
분 단위로 표현했더니 그래프가 너무 지저분하다.
시계열그래프임을 고려하여 smoothing을 진행한다.
우선 10분 단위로 smoothing을 하여 그래프를 그려본다.
15분, 30분 단위로도 smoothing을 해보니 15분 정도가 적절해보인다. (개인적인 판단)
아래 그래프들로 15분 단위로 smoothing하여 그래프를 그리기로 한다.

```
y <- users %>% filter(devicetype %in% c('iPhone')) %>% group_by(eventdatetime) %>% summarise(sum = sum(
head(y, 20)
```

```
## # A tibble: 20 x 2
##    eventdatetime      sum
##    <chr>            <int>
##  1 2018-09-27 15:00    64
##  2 2018-09-27 15:01    42
##  3 2018-09-27 15:02    27
##  4 2018-09-27 15:03    39
##  5 2018-09-27 15:04    42
##  6 2018-09-27 15:05    55
##  7 2018-09-27 15:06    82
##  8 2018-09-27 15:07    46
##  9 2018-09-27 15:08    73
## 10 2018-09-27 15:09    74
## 11 2018-09-27 15:10    49
## 12 2018-09-27 15:11    63
```

```
## 13 2018-09-27 15:12    45
## 14 2018-09-27 15:13    56
## 15 2018-09-27 15:14    40
## 16 2018-09-27 15:15    48
## 17 2018-09-27 15:16    41
## 18 2018-09-27 15:17    67
## 19 2018-09-27 15:18    33
## 20 2018-09-27 15:19    53
```

```
plot.ts(SMA(x$sum, n = 15), type = 'l')
lines(SMA(y$sum, n=15), type = 'l', col = 'red')
```



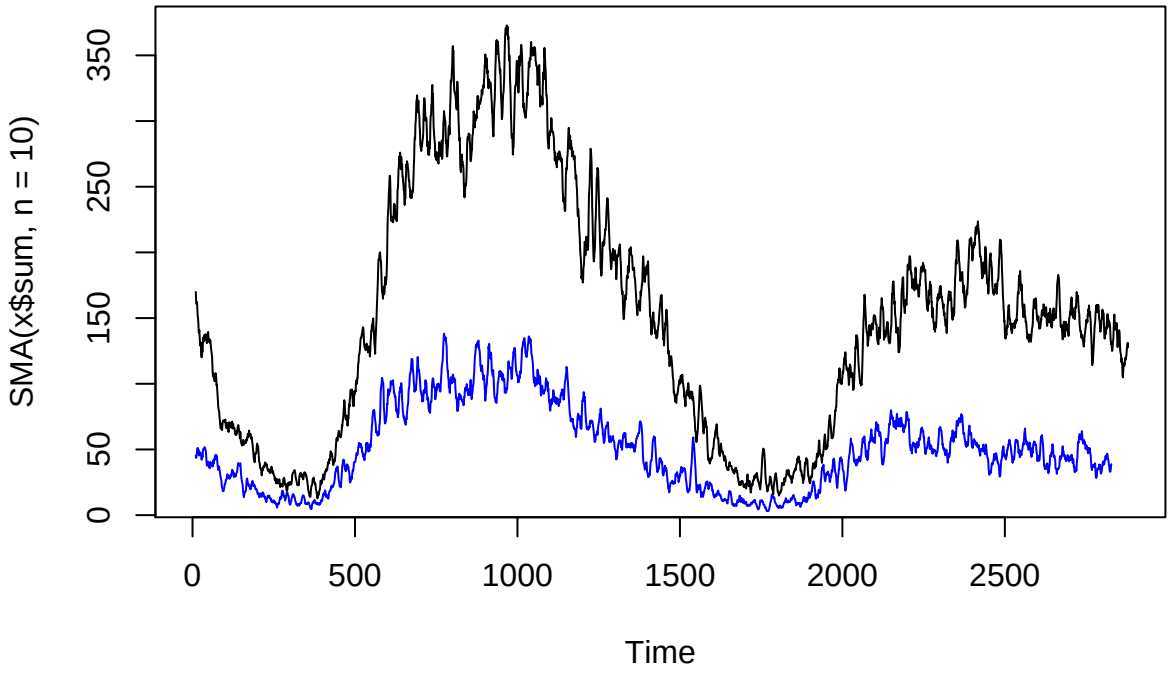
devicetype이 iPhone인 데이터에 대해서만 확인할 때는 devicetype으로 filter해서 시간별로 그룹핑하여 그래프를 그린다.
전체 그래프에 lines로 추가하여 비교하기로 한다.

```
z <- users  %>% filter(users$appversion == '4.0.4') %>% group_by(eventdatetime) %>% summarise(sum = sum
head(z, 20)
```

```
## # A tibble: 20 x 2
##    eventdatetime      sum
##    <chr>            <int>
##  1 2018-09-27 15:00    54
##  2 2018-09-27 15:01    35
##  3 2018-09-27 15:02    25
##  4 2018-09-27 15:03    28
```

```
##  5 2018-09-27 15:04    23
##  6 2018-09-27 15:05    39
##  7 2018-09-27 15:06    75
##  8 2018-09-27 15:07    41
##  9 2018-09-27 15:08    67
## 10 2018-09-27 15:09    65
## 11 2018-09-27 15:10    37
## 12 2018-09-27 15:11    47
## 13 2018-09-27 15:12    41
## 14 2018-09-27 15:13    53
## 15 2018-09-27 15:14    38
## 16 2018-09-27 15:15    48
## 17 2018-09-27 15:16    39
## 18 2018-09-27 15:17    66
## 19 2018-09-27 15:18    32
## 20 2018-09-27 15:19    53
```

```r
plot.ts(SMA(x$sum, n = 10), type = 'l')
lines(SMA(z$sum, n=10), type = 'l', col = 'blue')
```



appversion이 4.0.4인 데이터에 대해서만 확인할 때는 appversion으로 filter해서 시간별로 그룹핑하여 그래프를 그린다.
전체 그래프에 lines로 추가하여 비교하기로 한다.

Q3 자유 주제

1) 변수 추가

```
log$eventdate <- str_sub(log$eventdatetime, 1, 10)
log$eventtime <- str_sub(log$eventdatetime, 12,)
```

date와 time로 나눠보자

```
library(lubridate)
```

요일을 추가해보자

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

```
log$eventdate <- as.Date(log$eventdate)
log$event_day <- wday(log$eventdate, week_start = 1, label = T)
```

```
log$ap <- ifelse(str_sub(log$eventtime, 1, 2) < 12, 'AM', 'PM') %>% as.factor()
```

오전/오후를 추가해보자

```
log$event_h10m <- paste0(str_sub(log$eventtime, 1, 4),'0')
```

10분 단위 시간을 추가해보자

2) Imputation

user_id가 여러 번 등장하기 때문에 다른 rowuuid에서 정보를 얻을 수 있을 것이다.
imputation을 하는 것이 적절한지에 대해 확실치 않으므로
log0을 따로 만들어서 impuation을 진행하고 log와 비교해보자.

```
log0 <- log
colSums(is.na(log))
```

```
##             user_id             rowuuid       eventdatetime       eventcategory
##                   0                   0                   0                   0
##         funnel_name         funnel_desc              viewid          viewaction
##              129694              129694                   0                   0
##         viewid_desc                 Lv1                 Lv2     isfirstactivity
##              129694              129694              129694              103079
## isfirstgoalactivity          devicetype  devicemanufacturer              ostype
##              103079                   0                   0                   0
```

```
##           osversion          appversion      apppackagename          sourcetype
##                   0                   0                   0              103079
##             channel          categoryid        categoryname               depth
##              103079              373410              373410              373410
##                cat1             cat1_id                cat2             cat2_id
##              373410              373410              384094              384094
##                cat3             cat3_id       params_medium     params_campaign
##              418846              418846              406325              424926
##         params_term  inappeventcategory         event_rank           eventdate
##              432303                   0                   0                   0
##           eventtime           event_day                  ap          event_h10m
##                   0                   0                   0                   0
```

categoryid (373410) / categoryname (373410) / depth (373410)

```
log0$user_id[is.na(log$categoryname)] %>% unique() %>% head(10)
```

```
##  [1] "6cbfee08-70f1-40d2-9af3-d4e349621a51"
##  [2] "dc0a0ab6-ea19-4f03-822e-3ca44d65695f"
##  [3] "321424FC-860D-40D2-9C83-5176EDC55798"
##  [4] "184146c1-0e75-41ba-8a26-f72dd68c6f0e"
##  [5] "530855fd-1989-4ad0-92eb-dbaf4c549aa9"
##  [6] "8EEEC9E7-5A78-4156-8B33-4088E0163DBF"
##  [7] "ABF45E91-946F-43C9-9FAA-FD21624D5BDC"
##  [8] "05B38D4A-F228-46F4-9837-571E452DBCE1"
##  [9] "1C04F1E0-EEDD-4869-9376-EC550717A252"
## [10] "8faa8633-14f2-4f3c-890d-7a40541dcfe3"
```

```
log0$categoryname[log$user_id == "6cbfee08-70f1-40d2-9af3-d4e349621a51"] %>% as.factor() %>% summary()
```

```
## 3D   ·
##               1               9               1               1               4
##      ·             NA's
##              10              16
```

유저별로 자주 사용하는 카테고리로 대체해보려고 했으나 너무 다양하다.
다른 방법을 찾아보자.

isfirstactivity (103079) / isfirstgoalactivity (103079)

```
summary(log0$isfirstactivity)
```

```
##    Mode   FALSE    TRUE    NA's
## logical  290633   40532  103079
```

```
log0$isfirstactivity[is.na(log0$isfirstactivity)] <- FALSE

summary(log0$isfirstgoalactivity)
```

```
##    Mode   FALSE    TRUE    NA's
## logical  312140   19025  103079
```

```
log0$isfirstgoalactivity[is.na(log0$isfirstgoalactivity)] <- FALSE
```

first(goal)activity인지 아닌지 알 수 없다는 것은 아닌 것으로 봐도 무방하다고 생각한다.

대부분의 활동 로그가 first(goal)activity가 아니기 때문에 최빈값으로 대체한다.

sourcetype (103079) / channel (103079)

```
log0$sourcetype %>% as.factor() %>% summary()
```

```
##   app-market         paid     stranger unattributed        viral         NA's
##        19973         4773         5630       270834        29955       103079
```

```
log0$channel %>% as.factor() %>% summary()
```

```
##            (not set)      apple.searchads             facebook
##                 2444                  697                  126
##               google          google-play       google.adwords
##                 3186                19973                 4076
##               m_daum              m_naver m_naverpowercontents
##                  966                  849                  148
##             pc_naver         unattributed                  WEB
##                  227               270834                27639
##                NA's
##               103079
```

```
log0$sourcetype[log0$channel == '(not set)'] %>% unique()
```

```
## [1] "stranger" NA
```

```
us <- log0$user_id[is.na(log0$channel)] %>% unique()
log0$channel[log0$user_id %in% us] %>% unique()
```

```
##  [1] "unattributed"         "WEB"                  "google-play"
##  [4] "(not set)"            "m_naver"              "google"
##  [7] "google.adwords"       "m_daum"               "apple.searchads"
## [10] "m_naverpowercontents" "facebook"             "pc_naver"
## [13] NA
```

다른 사용로그에서 유입경로가 기록된 경우도 있다 -> user_id를 이용해서 채워보자

```
us_unattr <- log0$user_id[log0$user_id %in% us & log0$channel == 'unattributed'] %>% unique()
us_web <- log0$user_id[log0$user_id %in% us & log0$channel == 'WEB'] %>% unique()
us_gp <- log0$user_id[log0$user_id %in% us & log0$channel == 'google-play'] %>% unique()
us_ns <- log0$user_id[log0$user_id %in% us & log0$channel == '(not set)'] %>% unique()
us_nv <- log0$user_id[log0$user_id %in% us & log0$channel == 'm_naver'] %>% unique()
us_gg <- log0$user_id[log0$user_id %in% us & log0$channel == 'google'] %>% unique()
us_ga <- log0$user_id[log0$user_id %in% us & log0$channel == 'google.adwords'] %>% unique()
us_dm <- log0$user_id[log0$user_id %in% us & log0$channel == 'm_daum'] %>% unique()
us_apl <- log0$user_id[log0$user_id %in% us & log0$channel == 'apple.searchads'] %>% unique()
us_nvp <- log0$user_id[log0$user_id %in% us & log0$channel == 'm_naverpowercontents'] %>% unique()
us_fb <- log0$user_id[log0$user_id %in% us & log0$channel == 'facebook'] %>% unique()
us_pcnv <- log0$user_id[log0$user_id %in% us & log0$channel == 'pc_naver'] %>% unique()

log0$channel[log0$user_id %in% us_unattr] <- 'unattributed'
log0$channel[log0$user_id %in% us_web] <- 'WEB'
log0$channel[log0$user_id %in% us_gp] <- 'google-play'
log0$channel[log0$user_id %in% us_ns] <- '(not set)'
log0$channel[log0$user_id %in% us_nv] <- 'm_naver'
```

```
log0$channel[log0$user_id %in% us_gg] <- 'google'
log0$channel[log0$user_id %in% us_ga] <- 'google.adwords'
log0$channel[log0$user_id %in% us_dm] <- 'm_daum'
log0$channel[log0$user_id %in% us_apl] <- 'apple.searchads'
log0$channel[log0$user_id %in% us_nvp] <- 'm_naverpowercontents'
log0$channel[log0$user_id %in% us_fb] <- 'facebook'
log0$channel[log0$user_id %in% us_pcnv] <- 'pc_naver'

log0$channel[is.na(log0$channel)] <- '(not set)'
```

```
log0$sourcetype[log0$channel == 'unattributed'] %>% unique()
```

```
## [1] "unattributed" NA
```

```
# unattributed
log0$sourcetype[log0$channel == 'unattributed'] <- 'unattributed'

log0$sourcetype[log0$channel == 'WEB'] %>% unique()
```

```
## [1] "viral"        "unattributed" NA
```

```
log0$sourcetype[log0$channel == 'WEB'] %>% as.factor() %>% summary()
```

```
## unattributed        viral         NA's
##          573        27621         7841
```

```
# viral 27621, unattributed 573
log0$sourcetype[log0$channel == 'WEB'] <- 'viral'

log0$sourcetype[log0$channel == 'google-play'] %>% unique()
```

```
## [1] "app-market"   "unattributed" "viral"        NA
```

```
log0$sourcetype[log0$channel == 'google-play'] %>% as.factor() %>% summary()
```

```
##   app-market unattributed        viral         NA's
##        19973          495           18         4631
```

```
# app-market 19973, unattributed 495, viral 18
log0$sourcetype[log0$channel == 'google-play'] <- 'app-market'

log0$sourcetype[log0$channel == '(not set)'] %>% unique()
```

```
## [1] "stranger" NA
```

```
# stranger
log0$sourcetype[log0$channel == '(not set)'] <- 'stranger'

log0$sourcetype[log0$channel == 'm_naver'] %>% unique()
```

```
## [1] "viral"        "unattributed" NA
```

```
log0$sourcetype[log0$channel == 'm_naver'] %>% as.factor() %>% summary()
```

```
## unattributed        viral         NA's
##           21          849          254
```

```
# viral 849, unattributed 21
log0$sourcetype[log0$channel == 'm_naver'] <- 'viral'
```

```
log0$sourcetype[log0$channel == 'google'] %>% unique()
```

```
## [1] "stranger"     "unattributed" NA
```

```
log0$sourcetype[log0$channel == 'google'] %>% as.factor() %>% summary()
```

```
##     stranger unattributed         NA's
##         3186           52          654
```

```
# stranger 3186, unattributed 52
log0$sourcetype[log0$channel == 'google'] <- 'stranger'

log0$sourcetype[log0$channel == 'google.adwords'] %>% unique()
```

```
## [1] "paid"         "unattributed" NA
```

```
log0$sourcetype[log0$channel == 'google.adwords'] %>% as.factor() %>% summary()
```

```
##         paid unattributed         NA's
##         4076           26         1013
```

```
# paid 4076, unattributed 26
log0$sourcetype[log0$channel == 'google.adwords'] <- 'paid'

log0$sourcetype[log0$channel == 'm_daum'] %>% unique()
```

```
## [1] "viral" NA
```

```
# viral
log0$sourcetype[log0$channel == 'm_daum'] <- 'viral'

log0$sourcetype[log0$channel == 'apple.searchads'] %>% unique()
```

```
## [1] "paid"         NA             "unattributed"
```

```
log0$sourcetype[log0$channel == 'apple.searchads'] %>% as.factor() %>% summary()
```

```
##         paid unattributed         NA's
##          697            1          101
```

```
# paid 697, unattributed 1
log0$sourcetype[log0$channel == 'apple.searchads'] <- 'paid'

log0$sourcetype[log0$channel == 'm_naverpowercontents'] %>% unique()
```

```
## [1] "viral" NA
```

```
# viral
log0$sourcetype[log0$channel == 'm_naverpowercontents'] <- 'viral'

log0$sourcetype[log0$channel == 'facebook'] %>% unique()
```

```
## [1] "viral" NA
```

```
# viral
log0$sourcetype[log0$channel == 'facebook'] <- 'viral'

log0$sourcetype[log0$channel == 'pc_naver'] %>% unique()
```

```
## [1] "viral" NA
```

```
# viral
log0$sourcetype[log0$channel == 'pc_naver'] <- 'viral'

sum(is.na(log0$channel))
```

```
## [1] 0
```

```
sum(is.na(log0$sourcetype))
```

```
## [1] 0
```

각 channel별로 sourcetype이 하나이면 그 값으로, 여러 개이면 최빈값으로 대체한다.

viewid (0) / viewid_desc (129694) / funnel_name (304550) / funnel_desc (304550)

```
log0$viewid[is.na(log$funnel_name)] %>% unique()
```

```
## [1] "buyer_order_history_filter"    "buyer_payment_history_filter"
## [3] "category_gig_filter"           "inbox_detail_filter"
## [5] "nothing"                       "seller_selling_history_filter"
## [7] "search_gig_filter"
```

```
log0$funnel_name[log$viewid == 'buyer_order_history_filter'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'buyer_order_history'] %>% unique()
```

```
## [1] "transaction_history"
```

```
log0$funnel_name[log$viewid == 'buyer_payment_history_filter'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'buyer_payment_history'] %>% unique()
```

```
## [1] "transaction_history"
```

```
log0$funnel_name[log$viewid == 'category_gig_filter'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'category_gig'] %>% unique()
```

```
## [1] "category"
```

```
log0$funnel_name[log$viewid == 'inbox_detail_filter'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'inbox_detail'] %>% unique()
```

```
## [1] "inbox"
```

```
log0$funnel_name[log$viewid == 'nothing'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'seller_selling_history_filter'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'seller_selling_history'] %>% unique()
```

```
## [1] "transaction_history"
```

```
log0$funnel_name[log$viewid == 'search_gig_filter'] %>% unique()
```

```
## [1] NA
```

```
log0$funnel_name[log$viewid == 'search_gig'] %>% unique()
```

```
## [1] "search"
```

filter를 제외한 viewid가 있어서 그것을 바탕으로 대체하면 될 것 같다.

```
log0$funnel_name[log0$viewid == 'buyer_order_history_filter'] <- 'transaction_history'
log0$funnel_name[log0$viewid == 'buyer_payment_history_filter'] <- 'transaction_history'
log0$funnel_name[log0$viewid == 'category_gig_filter'] <- 'category'
log0$funnel_name[log0$viewid == 'inbox_detail_filter'] <- 'inbox'
log0$funnel_name[log0$viewid == 'nothing'] <- 'nothing'
log0$funnel_name[log0$viewid == 'seller_selling_history_filter'] <- 'transaction_history'
log0$funnel_name[log0$viewid == 'search_gig_filter'] <- 'search'

log0$funnel_desc[log0$funnel_name == 'transaction_history'] <- '  '
log0$funnel_desc[log0$funnel_name == 'category'] <- '  '
log0$funnel_desc[log0$funnel_name == 'inbox'] <- '  '
log0$funnel_desc[log0$funnel_name == 'nothing'] <- '  '
log0$funnel_desc[log0$funnel_name == 'search'] <- ' '

log0$viewid_desc[log0$viewid == 'buyer_order_history_filter'] <- '  -  '
log0$viewid_desc[log0$viewid == 'buyer_payment_history_filter'] <- '  -  '
log0$viewid_desc[log0$viewid == 'category_gig_filter'] <- '  -  '
log0$viewid_desc[log0$viewid == 'inbox_detail_filter'] <- '   - '
log0$viewid_desc[log0$viewid == 'seller_selling_history_filter'] <- '  -  '
log0$viewid_desc[log0$viewid == 'search_gig_filter'] <- ' -  '

log0$viewid_desc[log0$viewid == 'nothing'] <- '  '
```

```
colSums(is.na(log0))
```

```
##             user_id             rowuuid       eventdatetime       eventcategory
##                   0                   0                   0                   0
##         funnel_name         funnel_desc              viewid          viewaction
##                   0                   0                   0                   0
##         viewid_desc                 Lv1                 Lv2     isfirstactivity
##                   0              129694              129694                   0
## isfirstgoalactivity          devicetype  devicemanufacturer              ostype
##                   0                   0                   0                   0
##           osversion          appversion      apppackagename          sourcetype
##                   0                   0                   0                   0
##             channel          categoryid        categoryname               depth
##                   0              373410              373410              373410
##                cat1             cat1_id                cat2             cat2_id
##              373410              373410              384094              384094
##                cat3             cat3_id       params_medium     params_campaign
##              418846              418846              406325              424926
##         params_term  inappeventcategory         event_rank           eventdate
```

```
##             432303                  0                  0                  0
##          eventtime          event_day                 ap         event_h10m
##                  0                  0                  0                  0
```

category 관련 (categoryid/categoryname/depth/cat1/cat2/cat3/…), params 관련 (params_medium/params_campaign/p
제외하고는 imputation을 완료했다.

3) 구매 고객의 특징에 대해서 기본적인 시각화를 진행해보자.

```
library(ggplot2)
```

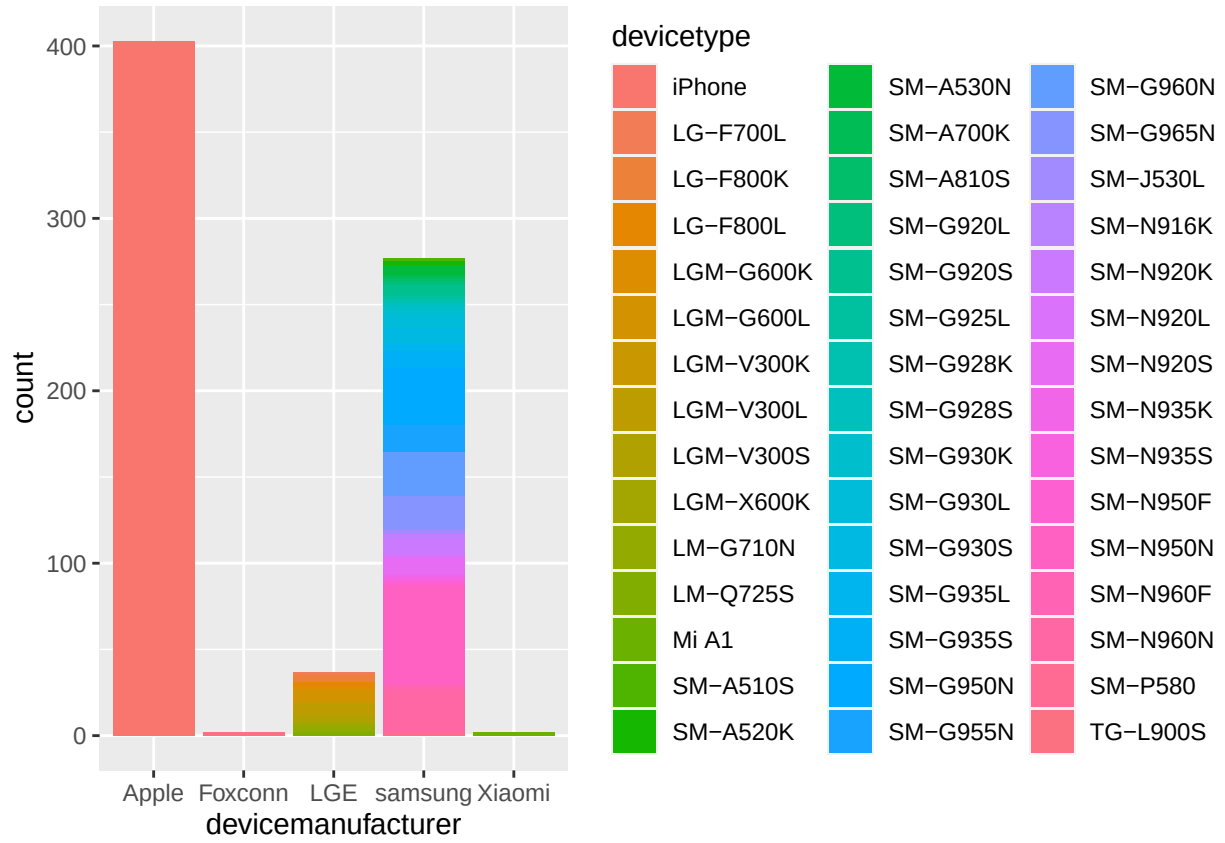
```
trg <- log$user_id[log$funnel_name == 'thankyou'] %>% unique()
trg_log <- log[log$user_id %in% trg,
               c('user_id', 'devicetype', 'devicemanufacturer', 'ostype', 'osversion','appversion', 'sou

trg_log0 <- log0[log0$user_id %in% trg,
                 c('user_id', 'devicetype', 'devicemanufacturer', 'ostype', 'osversion','appversion', 'so
```

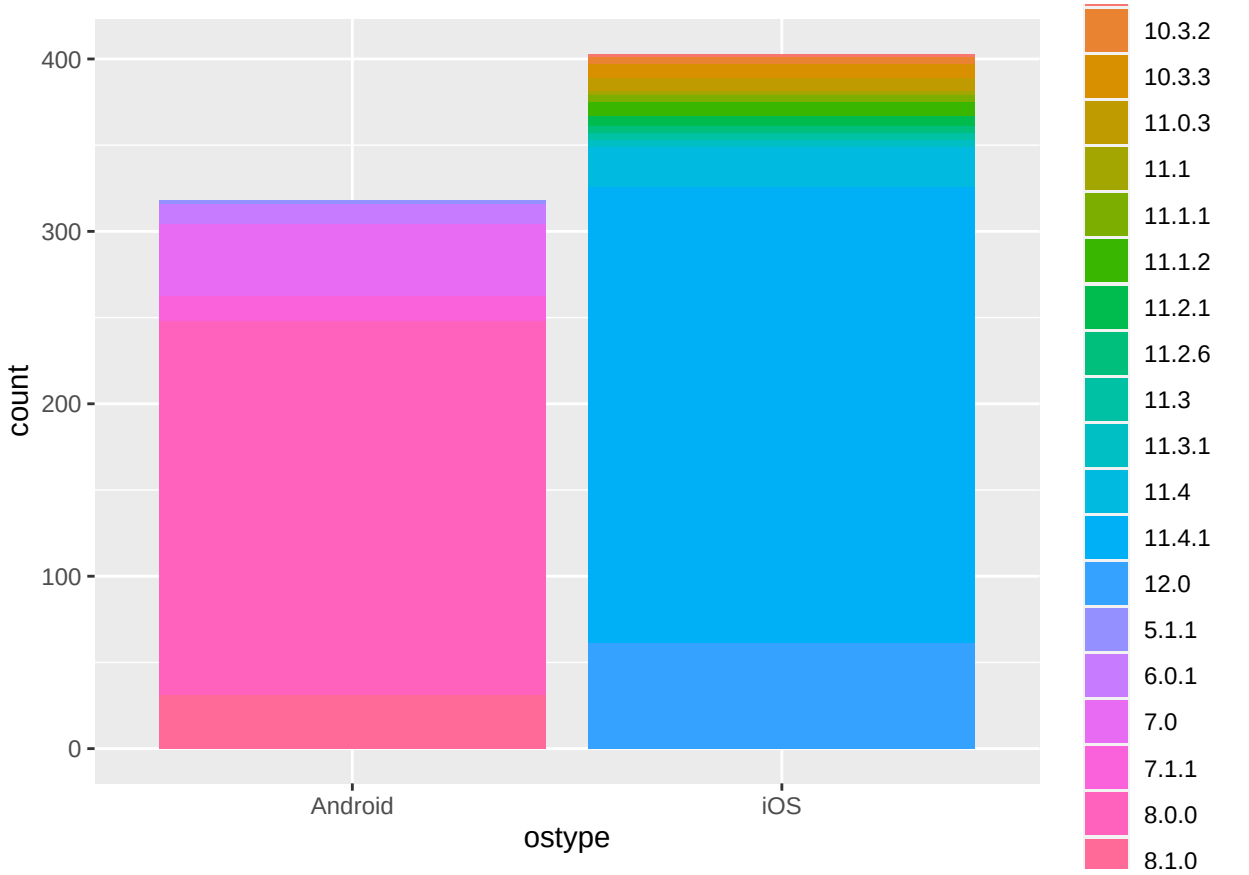
sourcetype과 channel의 imputation 여부에 따라 unique한 user가 다르게 나온다.
둘 다 해보자

imputation 하지 않은 데이터 기반 시각화 유저의 특징별로 barplot을 그려보자

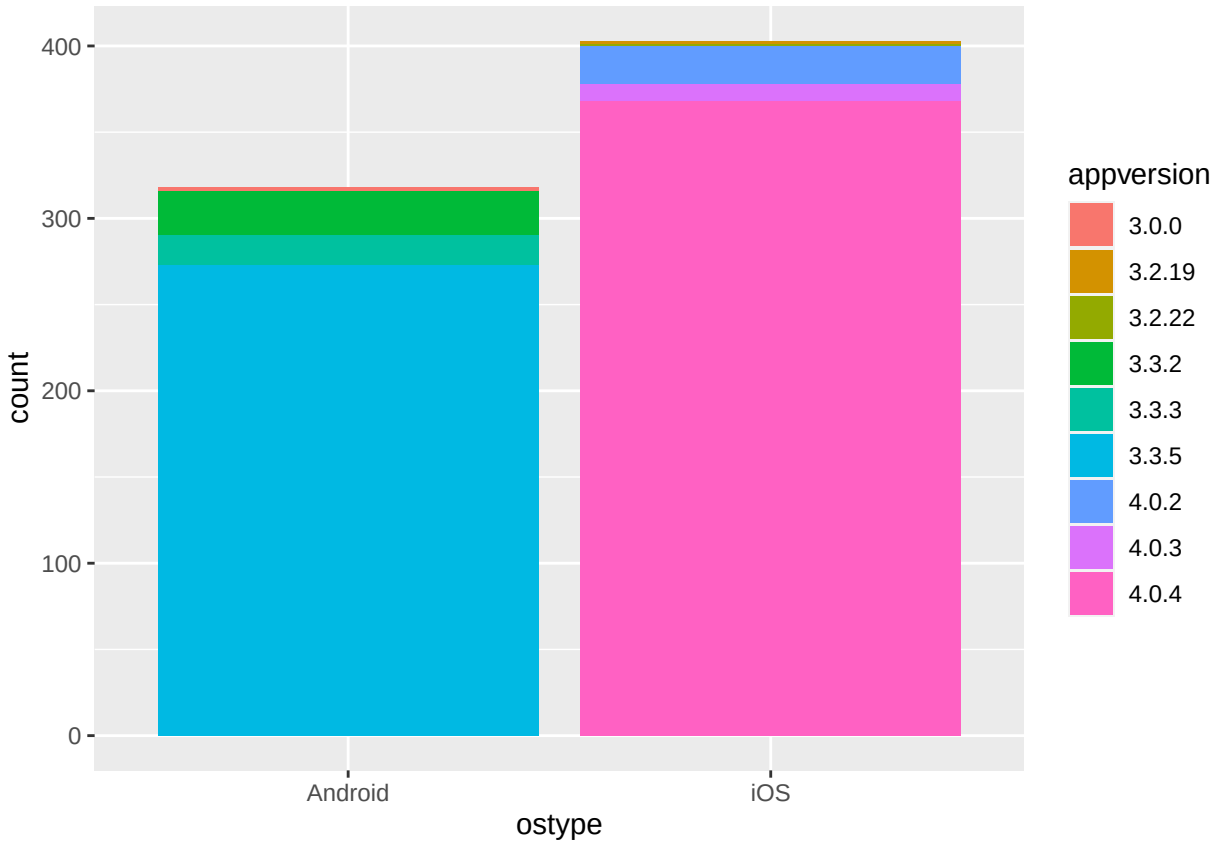
```
ggplot(trg_log, aes(x = devicemanufacturer, fill = devicetype)) + geom_bar()
```



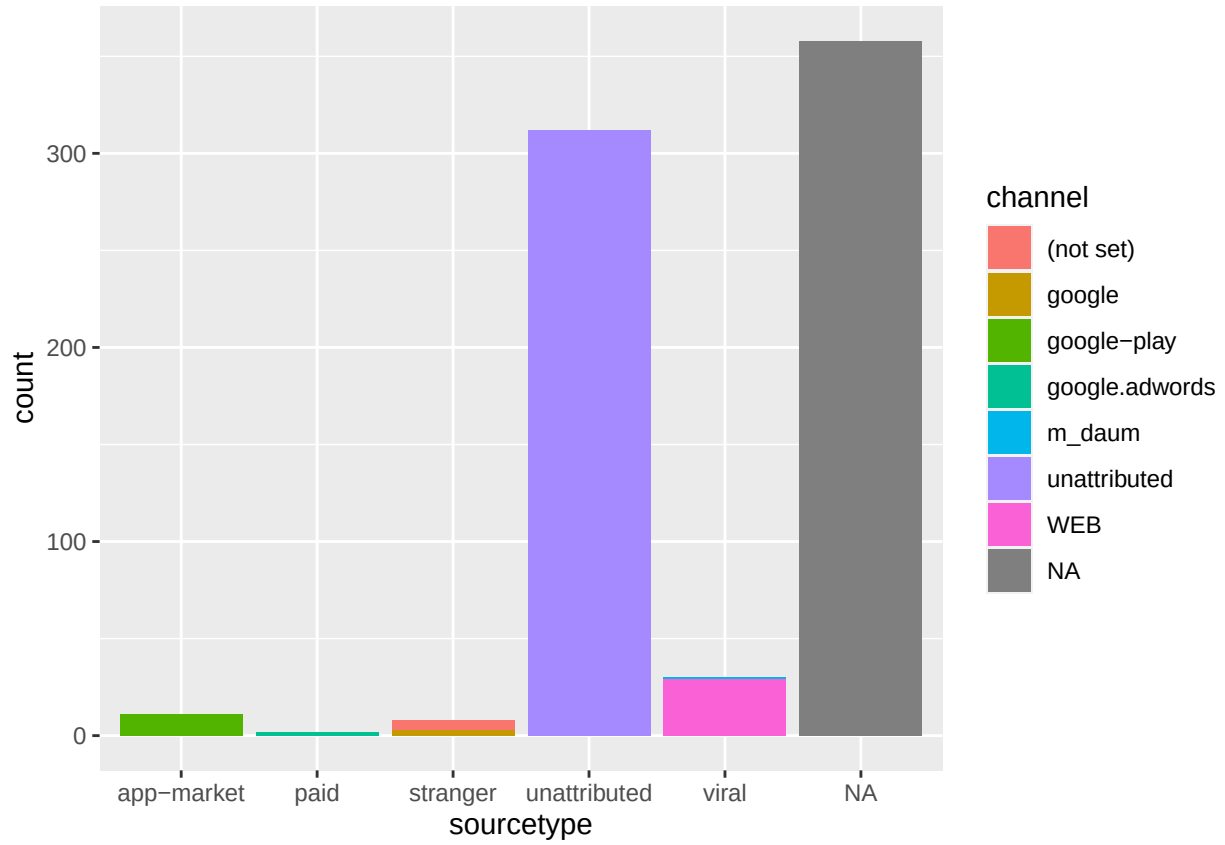

```
ggplot(trg_log, aes(x = ostype, fill = osversion)) + geom_bar()
```




```
ggplot(trg_log, aes(x = ostype, fill = appversion)) + geom_bar()
```

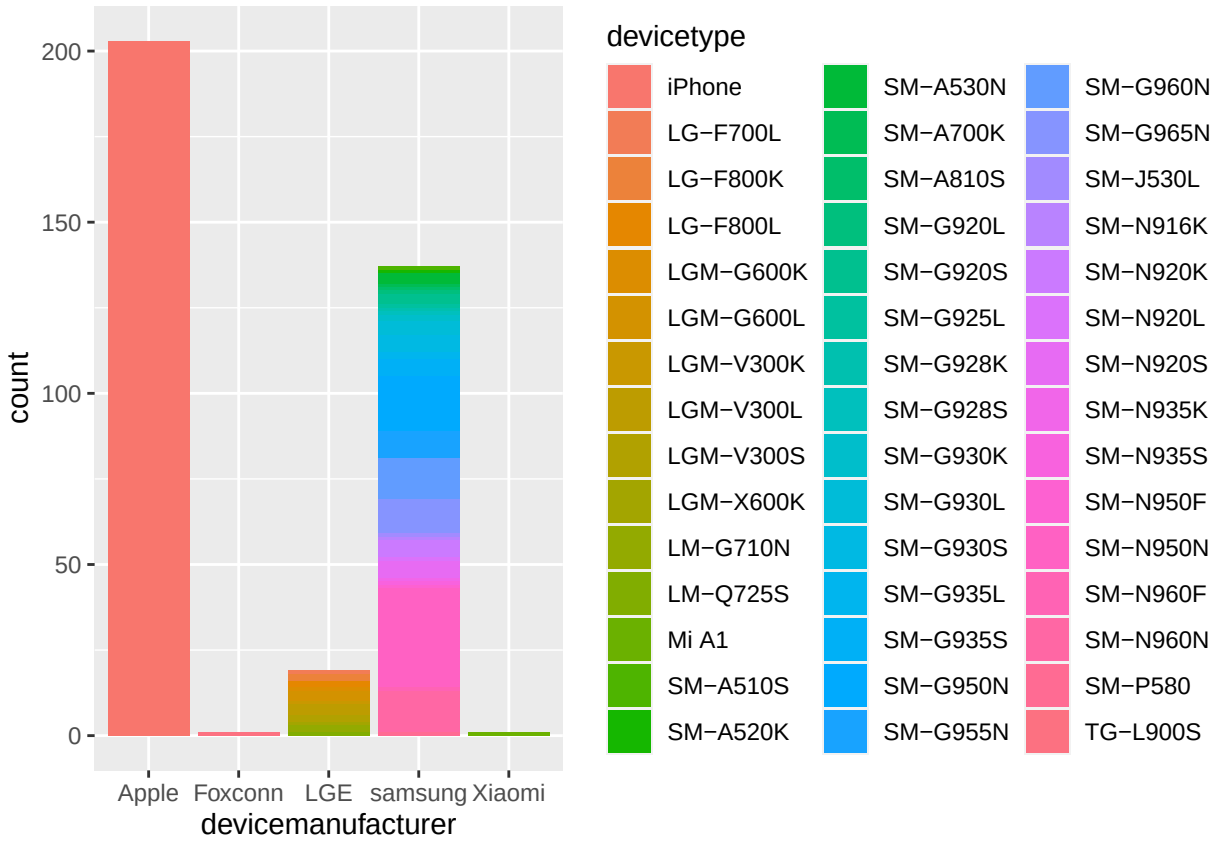



```
ggplot(trg_log, aes(x = sourcetype, fill = channel)) + geom_bar()
```

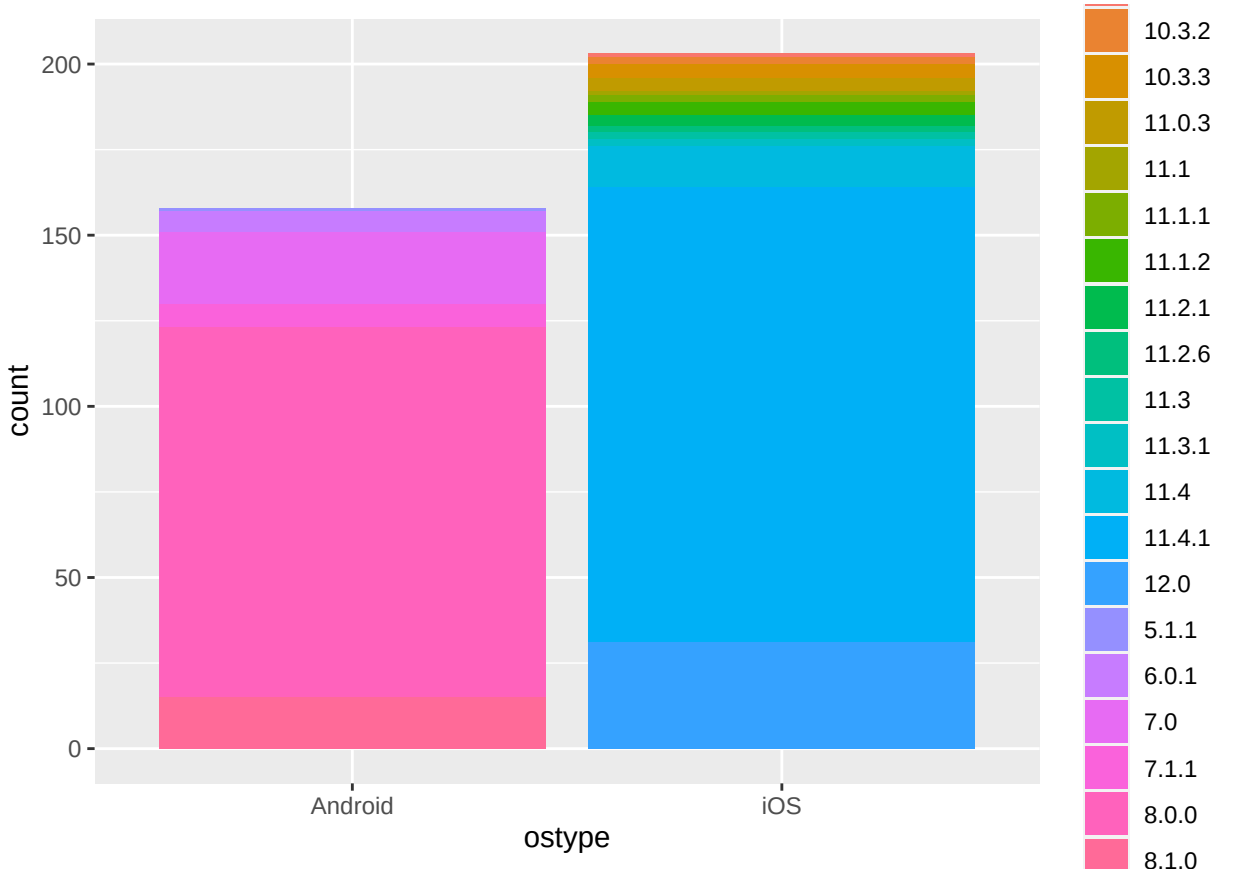


imputation 진행한 데이터 기반 시각화 유저의 특징별로 barplot을 그려보자

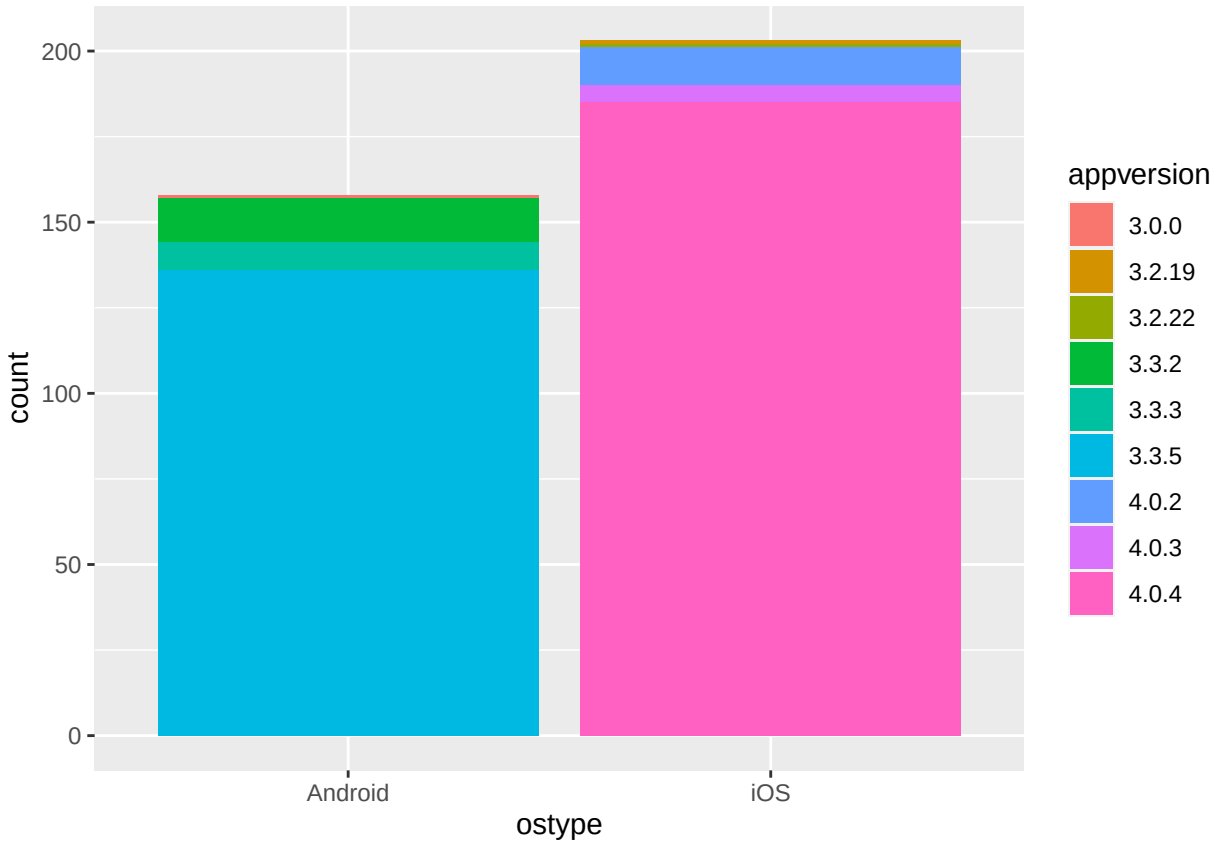
```
ggplot(trg_log0, aes(x = devicemanufacturer, fill = devicetype)) + geom_bar()
```



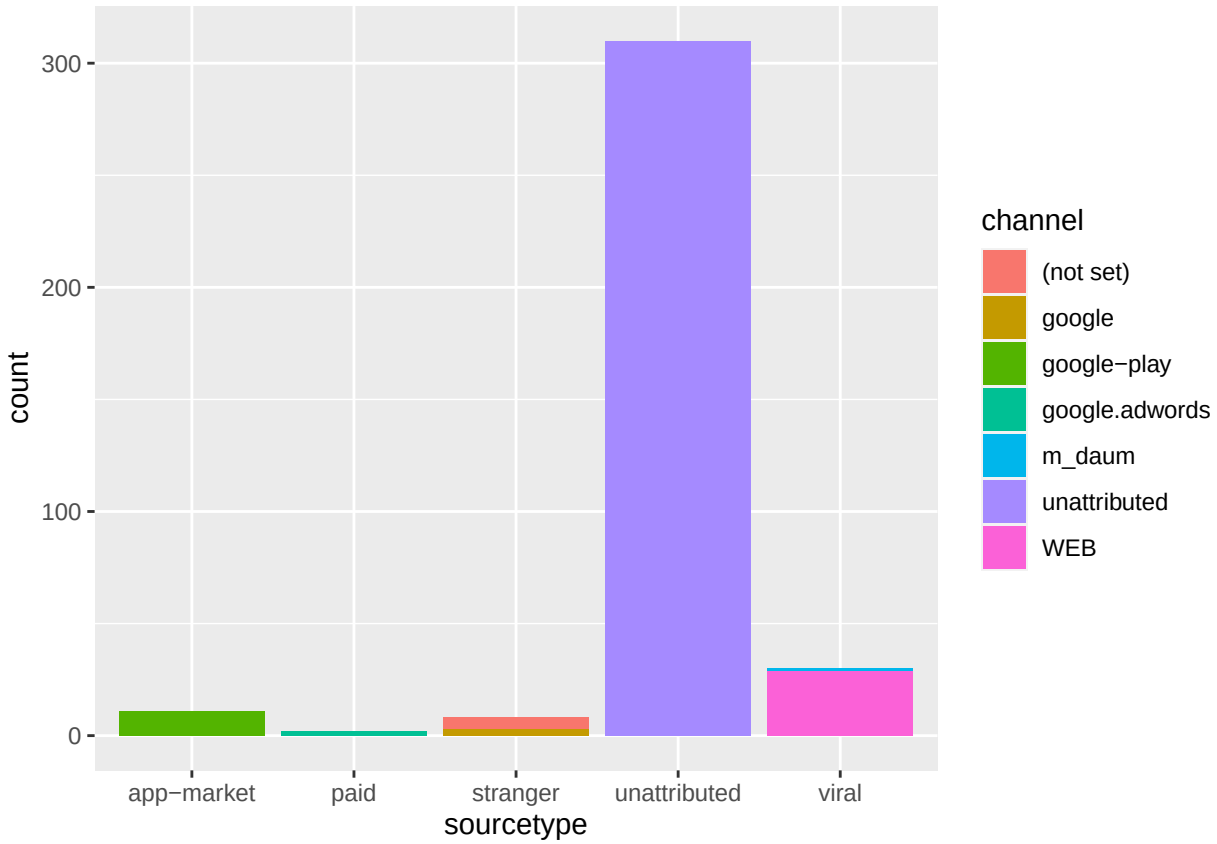

```
ggplot(trg_log0, aes(x = ostype, fill = osversion)) + geom_bar()
```




```
ggplot(trg_log0, aes(x = ostype, fill = appversion)) + geom_bar()
```




```
ggplot(trg_log0, aes(x = sourcetype, fill = channel)) + geom_bar()
```



4) modelling

가장 기본적인 의사결정나무 모형을 적용해보자.

```
library(C50)
library(gmodels)
set.seed(2020)
```

```
log1 <- log
log1$thankyou <- ifelse(log1$user_id %in% trg, 'yes', 'no') %>% as.factor()

log1$eventcategory <- as.factor(log1$eventcategory)
log1$funnel_name <- as.factor(log1$funnel_name)
log1$viewid <- as.factor(log1$viewid)
log1$viewaction <- as.factor(log1$viewaction)
log1$isfirstactivity <- as.factor(log1$isfirstactivity)
log1$isfirstgoalactivity <- as.factor(log1$isfirstgoalactivity)
log1$devicetype <- as.factor(log1$devicetype)
log1$devicemanufacturer <- as.factor(log1$devicemanufacturer)
log1$ostype <- as.factor(log1$ostype)
log1$osversion <- as.factor(log1$osversion)
log1$appversion <- as.factor(log1$appversion)
```

```
log1$sourcetype <- as.factor(log1$sourcetype)
log1$channel <- as.factor(log1$channel)
log1$categoryname <- as.factor(log1$categoryname)
```

```
train = sample(1:dim(log1)[1], 0.6*dim(log1)[1])
log1_train <- log1[train,]
log1_test <- log1[-train,]

log1_train[c(4,5,7,8,12,13,14,15,16,17,18,20,21,23)] %>% colnames()
```

imputation을 하지 않은 기본 데이터로 모델을 만들어보자

```
##  [1] "eventcategory"       "funnel_name"         "viewid"
##  [4] "viewaction"          "isfirstactivity"     "isfirstgoalactivity"
##  [7] "devicetype"          "devicemanufacturer"  "ostype"
## [10] "osversion"           "appversion"          "sourcetype"
## [13] "channel"             "categoryname"
```

```
crd <- C5.0(log1_train[c(4,5,7,8,12,13,14,15,16,17,18,20,21)], log1_train$thankyou,
            control = C5.0Control(minCases = 10))
crd
```

```
##
## Call:
## C5.0.default(x = log1_train[c(4, 5, 7, 8, 12, 13, 14, 15, 16, 17, 18, 20,
##  21)], y = log1_train$thankyou, control = C5.0Control(minCases = 10))
##
## Classification Tree
## Number of samples: 260546
## Number of predictors: 13
##
## Tree size: 265
##
## Non-standard options: attempt to group attributes, minimum number of cases: 10
```

```
# summary(crd)
crd.pred <- predict(crd, log1_test)
CrossTable(log1_test$thankyou, crd.pred,
           prop.chisq = F, prop.c = F, prop.r = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  173698
##
##
##                     | crd.pred
## log1_test$thankyou |        no |       yes | Row Total |
## -------------------|-----------|-----------|-----------|
```

```
##                  no |    150129 |      1419 |    151548 |
##                     |     0.864 |     0.008 |           |
## -------------------|-----------|-----------|-----------|
##                 yes |     17342 |      4808 |     22150 |
##                     |     0.100 |     0.028 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |    167471 |      6227 |    173698 |
## -------------------|-----------|-----------|-----------|
##
##
```

```
F1_Score(y_pred = crd.pred, y_true = log1_test$thankyou, positive = "yes")
```

```
## [1] 0.338866
```

precision = 4808/6227 = 0.7721214
recall = 4808/22150 = 0.2170655

no의 비율이 높기 때문에 yes를 예측하는 데 sensitivity가 높지 않은 편이다.
no의 비율을 좀 낮춘 sample을 추출하여 모델을 만들어보자.

```
cus_yes <- log1[log1$thankyou == 'yes',]
cus_no <- log1[log1$thankyou == 'no',]
cus_samp <- rbind(cus_yes, cus_no[sample(1:dim(cus_no)[1], dim(cus_yes)[1]*2),])
prop.table(table(cus_samp$thankyou))
```

```
##
##        no       yes
## 0.6666667 0.3333333
```

```
train_samp = sample(1:dim(cus_samp)[1], 0.6*dim(cus_samp)[1])
cus_train <- cus_samp[train_samp,]
cus_test <- cus_samp[-train_samp,]

crd_samp <- C5.0(cus_train[c(4,5,7,8,12,13,14,15,16,17,18,20,21)], cus_train$thankyou,
                 control = C5.0Control(minCases = 10))
crd_samp
```

```
##
## Call:
## C5.0.default(x = cus_train[c(4, 5, 7, 8, 12, 13, 14, 15, 16, 17, 18, 20,
##  21)], y = cus_train$thankyou, control = C5.0Control(minCases = 10))
##
## Classification Tree
## Number of samples: 100728
## Number of predictors: 13
##
## Tree size: 309
##
## Non-standard options: attempt to group attributes, minimum number of cases: 10
```

```
# summary(crd_samp)
crd_samp.pred <- predict(crd_samp, cus_test)
CrossTable(cus_test$thankyou, crd_samp.pred,
           prop.chisq = F, prop.c = F, prop.r = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  67152
##
##
##                    | crd_samp.pred
## cus_test$thankyou |        no |       yes | Row Total |
## ------------------|-----------|-----------|-----------|
##                no |     39748 |      5035 |     44783 |
##                   |     0.592 |     0.075 |           |
## ------------------|-----------|-----------|-----------|
##               yes |     11539 |     10830 |     22369 |
##                   |     0.172 |     0.161 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |     51287 |     15865 |     67152 |
## ------------------|-----------|-----------|-----------|
##
##
```

```
F1_Score(y_pred = crd_samp.pred, y_true = cus_test$thankyou, positive = "yes")
```

```
## [1] 0.5665115
```

precision = 10830/15865 = 0.6826347
recall = 10830/22369 = 0.4841522

yes와 no의 비율을 1:2로 조정한 샘플을 통해 분류를 진행해보니
precision이 감소했으나 recall이 꽤 증가한 것이 보인다.
F1 score도 증가한 것을 보아 샘플을 통해 만든 모델이 더 나을 것이라고 할 수 있겠다.
그렇다고 분류 성능 평가 지표값들이 충분히 높은 것은 아니기 때문에 데이터 비대칭을 해결하고
hyperparameter tuning을 통해 더 좋은 모델을 만들 수 있는 방법을 고민해볼 필요가 있을 것 같다.

```
log2 <- log0
log2$thankyou <- ifelse(log1$user_id %in% trg, 'yes', 'no') %>% as.factor()

log2$eventcategory <- as.factor(log2$eventcategory)
log2$funnel_name <- as.factor(log2$funnel_name)
log2$viewid <- as.factor(log2$viewid)
log2$viewaction <- as.factor(log2$viewaction)
log2$isfirstactivity <- as.factor(log2$isfirstactivity)
log2$isfirstgoalactivity <- as.factor(log2$isfirstgoalactivity)
log2$devicetype <- as.factor(log2$devicetype)
log2$devicemanufacturer <- as.factor(log2$devicemanufacturer)
log2$ostype <- as.factor(log2$ostype)
```

```
log2$osversion <- as.factor(log2$osversion)
log2$appversion <- as.factor(log2$appversion)
log2$sourcetype <- as.factor(log2$sourcetype)
log2$channel <- as.factor(log2$channel)
log2$categoryname <- as.factor(log2$categoryname)
```

```
train2 = sample(1:dim(log2)[1], 0.6*dim(log2)[1])
log2_train <- log2[train2,]
log2_test <- log2[-train2,]

crd2 <- C5.0(log2_train[c(4,5,7,8,12,13,14,15,16,17,18,20,21)], log2_train$thankyou, control = C5.0Cont
crd2
```

imputation을 진행한 데이터로 모델을 만들어보자

```
##
## Call:
## C5.0.default(x = log2_train[c(4, 5, 7, 8, 12, 13, 14, 15, 16, 17, 18, 20,
##  21)], y = log2_train$thankyou, control = C5.0Control(minCases = 10))
##
## Classification Tree
## Number of samples: 260546
## Number of predictors: 13
##
## Tree size: 207
##
## Non-standard options: attempt to group attributes, minimum number of cases: 10
```

```
# summary(crd2)
crd2.pred <- predict(crd2, log2_test)
CrossTable(log2_test$thankyou, crd2.pred,
           prop.chisq = F, prop.c = F, prop.r = F)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  173698
##
##
##                     | crd2.pred
## log2_test$thankyou |        no |       yes | Row Total |
## -------------------|-----------|-----------|-----------|
##                 no |    150258 |      1174 |    151432 |
##                    |     0.865 |     0.007 |           |
## -------------------|-----------|-----------|-----------|
##                yes |     17513 |      4753 |     22266 |
##                    |     0.101 |     0.027 |           |
## -------------------|-----------|-----------|-----------|
```

```
##       Column Total |    167771 |      5927 |    173698 |
## -------------------|-----------|-----------|-----------|
##
##
```

```
F1_Score(y_pred = crd2.pred, y_true = log2_test$thankyou, positive = "yes")
```

```
## [1] 0.3371759
```

precision = 4753/5927 = 0.8019234
recall = 4753/22266 = 0.2134645

위의 imputation을 하지 않은 모델(precision = 0.7721214 & recall = 0.2170655)과 비교하면,
precision은 소폭 상승했고, recall은 소폭 감소했다.
imputation이 유의미한지는 잘 알 수 없다.

no의 비율이 높기 때문에 yes를 예측하는 데 sensitivity가 높지 않은 편이다.
no의 비율을 좀 낮춘 sample을 추출하여 모델을 만들어보자.

```
cus2_yes <- log2[log2$thankyou == 'yes',]
cus2_no <- log2[log2$thankyou == 'no',]
cus2_samp <- rbind(cus2_yes, cus2_no[sample(1:dim(cus2_no)[1], dim(cus2_yes)[1]*2),])
prop.table(table(cus2_samp$thankyou))
```

```
##
##        no       yes
## 0.6666667 0.3333333
```

```
train2_samp = sample(1:dim(cus2_samp)[1], 0.6*dim(cus2_samp)[1])
cus2_train <- cus2_samp[train2_samp,]
cus2_test <- cus2_samp[-train2_samp,]

crd2_samp <- C5.0(cus2_train[c(4,5,7,8,12,13,14,15,16,17,18,20,21)], cus2_train$thankyou,
                  control = C5.0Control(minCases = 10))
crd2_samp
```

```
##
## Call:
## C5.0.default(x = cus2_train[c(4, 5, 7, 8, 12, 13, 14, 15, 16, 17, 18, 20,
##  21)], y = cus2_train$thankyou, control = C5.0Control(minCases = 10))
##
## Classification Tree
## Number of samples: 100728
## Number of predictors: 13
##
## Tree size: 284
##
## Non-standard options: attempt to group attributes, minimum number of cases: 10
```

```
# summary(crd_samp)
crd2_samp.pred <- predict(crd2_samp, cus2_test)
CrossTable(cus2_test$thankyou, crd2_samp.pred,
           prop.chisq = F, prop.c = F, prop.r = F)
```

```
##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  67152
##
##
##                     | crd2_samp.pred
## cus2_test$thankyou |        no |       yes | Row Total |
## -------------------|-----------|-----------|-----------|
##                 no |     40425 |      4294 |     44719 |
##                    |     0.602 |     0.064 |           |
## -------------------|-----------|-----------|-----------|
##                yes |     11925 |     10508 |     22433 |
##                    |     0.178 |     0.156 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |     52350 |     14802 |     67152 |
## -------------------|-----------|-----------|-----------|
##
##
```

```
F1_Score(y_pred = crd2_samp.pred, y_true = cus2_test$thankyou, positive = "yes")
```

```
## [1] 0.5644152
```

precision = 10508/14802 = 0.7099041
recall = 10508/22433 = 0.4684171

위의 imputation을 하지 않은 모델(precision = 0.6826347 & recall = 0.4841522)과 비교하면,
precision은 소폭 상승했고, recall은 소폭 감소했다.
imputation이 유의미한지는 잘 알 수 없다.

yes와 no의 비율을 1:2로 조정한 샘플을 통해 분류를 진행해보니
precision이 감소했으나 recall이 꽤 증가한 것이 보인다.
F1 score도 증가한 것을 보아 샘플을 통해 만든 모델이 더 나을 것이라고 할 수 있겠다.
그렇다고 분류 성능 평가 지표값들이 충분히 높은 것은 아니기 때문에 데이터 비대칭을 해결하고
hyperparameter tuning을 통해 더 좋은 모델을 만들 수 있는 방법을 고민해볼 필요가 있을 것 같다.

앞으로 프로세스 마이닝을 추가 학습하여 적용해보자.