HW0-Fangzhou Song

Fangzhou Song

Package loading

```
rm(list = ls())
library(tidyverse)
```

Read data

```
data=read_csv("C:/Users/ArkSong/Desktop/GWU/Stat 6240-Statistical Data Mining/Assignments/HW0/events_log
```

Data processing

```
options(scipen=200)
glimpse(data)
```

```
## Observations: 400,165
## Variables: 9
## $ uuid            <chr> "00000736167c507e8ec225bd9e71f9e5", "00000c69f...
## $ timestamp       <dbl> 20160301103842, 20160307005226, 20160302145305...
## $ session_id      <chr> "78245c2c3fba013a", "c559c3be98dca8a4", "760bf...
## $ group           <chr> "b", "a", "a", "a", "a", "a", "a", "b", "a", "...
## $ action          <chr> "searchResultPage", "searchResultPage", "check...
## $ checkin         <int> NA, NA, 30, 60, 30, 180, 240, NA, 180, 150, NA...
## $ page_id         <chr> "cbeb66d1bc1f1bc2", "eb658e8722aad674", "f99a9...
## $ n_results       <int> 5, 10, NA, NA, NA, NA, NA, 15, NA, NA, 20, NA,...
## $ result_position <int> NA, NA, NA, 10, NA, NA, NA, NA, 1, 1, NA, 1, N...
```

Question 1

1.What is their daily overall clickthrough rate? How does it vary between the groups?

Add variables year, month, day

```
data$timestamp=as.character(data$timestamp)
```

```
data_ymd=mutate(data,
                year=substr(timestamp,1,4),
                month=substr(timestamp,5,6),
                day=substr(timestamp,7,8),
                result_position=factor(result_position)
                )
```

Calculate daily overall clickthrough rate

```
data1=data_ymd %>%
  select(year,month,day,group,session_id,timestamp,action,result_position) %>%
  filter(action=="searchResultPage" |action=="visitPage" ) %>%
  arrange(day,session_id,timestamp) %>%
```

```
  group_by(year,month,day,session_id) %>%
  mutate(
    action_lag=lag(action)
  )
knitr::kable(head(data1,20))
```

year	month	day	group	session_id	timestamp	action	result_position	action_lag
2016	03	01	b	000936ae06d62383	20160301123654	searchResultPage	NA	NA
2016	03	01	b	001544bc03fac3e8	20160301113558	searchResultPage	NA	NA
2016	03	01	b	001544bc03fac3e8	20160301113618	searchResultPage	NA	searchResultPa
2016	03	01	a	001a3950cd4ac6c6	20160301180806	searchResultPage	NA	NA
2016	03	01	a	001a3950cd4ac6c6	20160301180811	searchResultPage	NA	searchResultPa
2016	03	01	a	001a3950cd4ac6c6	20160301180830	searchResultPage	NA	searchResultPa
2016	03	01	a	001a3950cd4ac6c6	20160301180846	searchResultPage	NA	searchResultPa
2016	03	01	a	001a3950cd4ac6c6	20160301180848	searchResultPage	NA	searchResultPa
2016	03	01	b	001e2d0e159172d2	20160301004321	searchResultPage	NA	NA
2016	03	01	b	001e2d0e159172d2	20160301004325	visitPage	2	searchResultPa
2016	03	01	b	0022bba0634595b9	20160301033739	searchResultPage	NA	NA
2016	03	01	a	0024c4506bf92e1c	20160301065221	searchResultPage	NA	NA
2016	03	01	a	0024c4506bf92e1c	20160301065246	searchResultPage	NA	searchResultPa
2016	03	01	a	0024c4506bf92e1c	20160301065249	visitPage	1	searchResultPa
2016	03	01	a	0024c4506bf92e1c	20160301065522	visitPage	NA	visitPage
2016	03	01	a	0024f4f005f34c9d	20160301102517	searchResultPage	NA	NA
2016	03	01	a	0024f4f005f34c9d	20160301102553	searchResultPage	NA	searchResultPa
2016	03	01	a	0024f4f005f34c9d	20160301102712	searchResultPage	NA	searchResultPa
2016	03	01	b	002601319d1a02e1	20160301114301	searchResultPage	NA	NA
2016	03	01	a	0029420a5f8c7d90	20160301204146	searchResultPage	NA	NA

Arrange all record in time order group by each session.

It is easy to conclude that each searchResultPage action is unique and valid but one searchResultPage action may leads several visitPage action records (0,1,2,3,...). Even though someone click more than 1 page after one search, it should be regard as only one valid click. Therefore, I just choose the 1st visitPage record that follows each searchResultPage action as once valid click.

Use lag function to each session's sequential record. If a visitPage record's action_lag is "sequential" , then it must be 1st visitPage action for each session.

```
data2=data1 %>%
  filter(action=="searchResultPage" | (action=="visitPage" & action_lag=="searchResultPage")
  )
```

data2 is the dataset which contains "searchResultPage" and only first "visitPage" action for each session.

```
(data2_overall=data2%>%
  group_by(year,month,day) %>%
  summarise(
    visitPage_count=sum(action=="visitPage"),
    searchResultPage_count=sum(action=="searchResultPage"),
    cr=visitPage_count/searchResultPage_count
  ))
```

```
## # A tibble: 8 x 6
## # Groups:   year, month [?]
##   year  month day   visitPage_count searchResultPage_count    cr
```

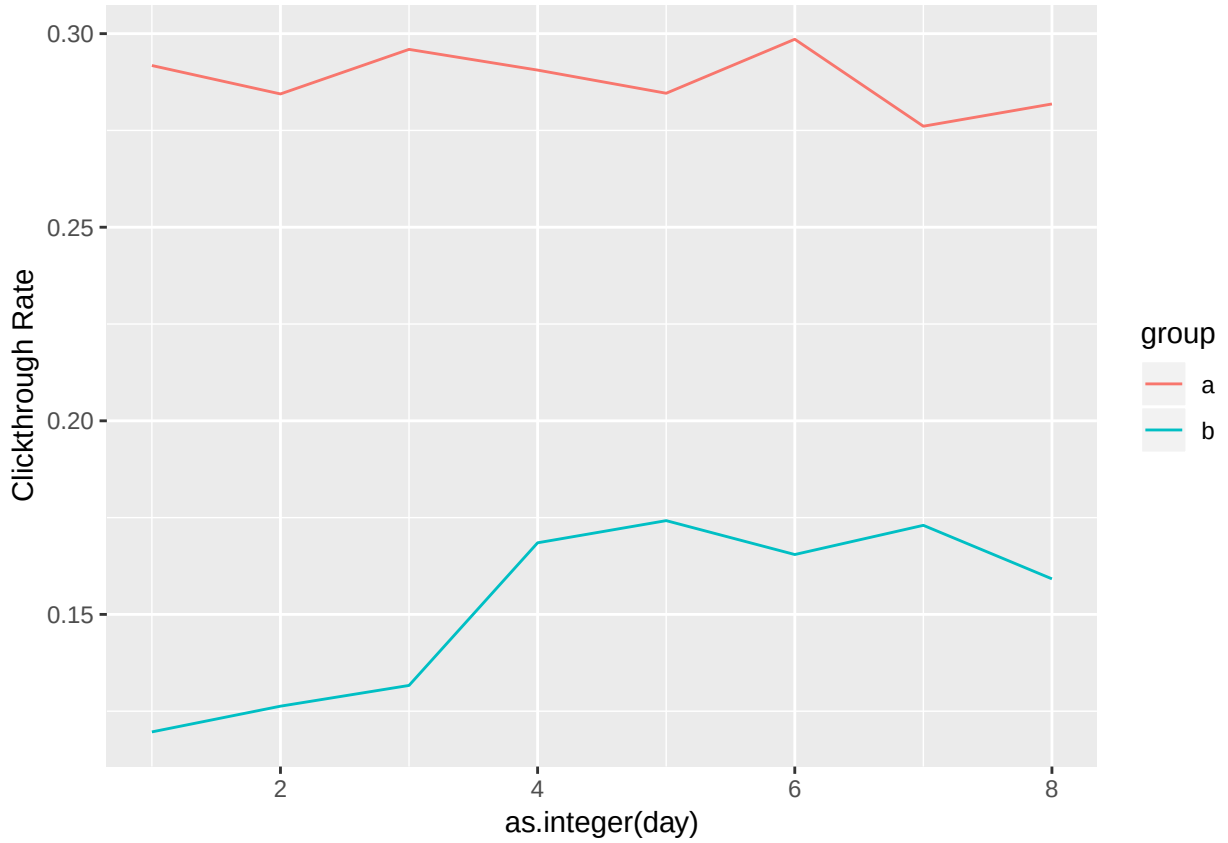
```
##   <chr> <chr> <chr>           <int>                  <int> <dbl>
## 1 2016  03    01               4364                  18374 0.238
## 2 2016  03    02               4476                  18902 0.237
## 3 2016  03    03               4704                  19159 0.246
## 4 2016  03    04               4189                  16675 0.251
## 5 2016  03    05               3251                  13204 0.246
## 6 2016  03    06               3678                  14612 0.252
## 7 2016  03    07               4598                  19011 0.242
## 8 2016  03    08               3932                  16297 0.241
```

Between groups

```
(data2_group=data2 %>%
  group_by(year,month,day,group) %>%
  summarise(
    visitPage_count=sum(action=="visitPage"),
    searchResultPage_count=sum(action=="searchResultPage"),
    cr=visitPage_count/searchResultPage_count
  )
)
```

```
## # A tibble: 16 x 7
## # Groups:   year, month, day [?]
##    year  month day   group visitPage_count searchResultPage_count    cr
##    <chr> <chr> <chr> <chr>           <int>                  <int> <dbl>
##  1 2016  03    01    a                3671                  12582 0.292
##  2 2016  03    01    b                 693                   5792 0.120
##  3 2016  03    02    a                3757                  13209 0.284
##  4 2016  03    02    b                 719                   5693 0.126
##  5 2016  03    03    a                3930                  13280 0.296
##  6 2016  03    03    b                 774                   5879 0.132
##  7 2016  03    04    a                3283                  11298 0.291
##  8 2016  03    04    b                 906                   5377 0.168
##  9 2016  03    05    a                2451                   8612 0.285
## 10 2016  03    05    b                 800                   4592 0.174
## 11 2016  03    06    a                2827                   9469 0.299
## 12 2016  03    06    b                 851                   5143 0.165
## 13 2016  03    07    a                3506                  12699 0.276
## 14 2016  03    07    b                1092                   6312 0.173
## 15 2016  03    08    a                3074                  10907 0.282
## 16 2016  03    08    b                 858                   5390 0.159
```

```
ggplot(data=data2_group)+
  geom_line(mapping = aes(x=as.integer(day),y=cr,color=group))+
  ylab("Clickthrough Rate")
```



It can been seen from the plot that group A has a significant higher clickthrough rate than group B.

Qusetion 2

2.Which results do people tend to try first? How does it change day-to-day?

From the Question1, we have already got the dataset which contains "searchResultPage" and only first "visitPage" action for each session, which is data2. Thus, we just need to count the total number of each "result_position" to see which results people tend to try first using data2.

```
(data_first=data2 %>%
  filter(action=="visitPage") %>%
  ungroup() %>%
  count(result_position))
```

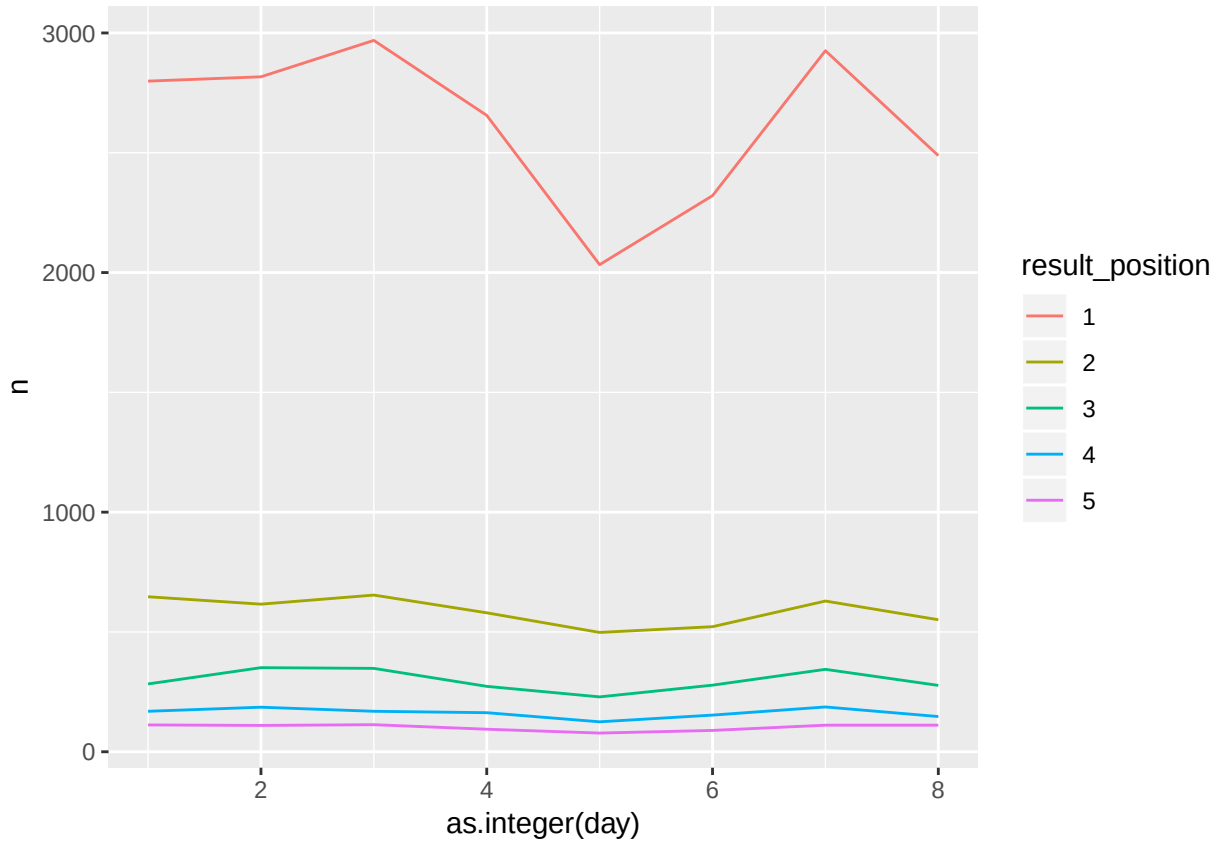
```
## # A tibble: 190 x 2
##    result_position     n
##    <fct>           <int>
##  1 1               21009
##  2 2                4697
##  3 3                2383
##  4 4                1299
##  5 5                 818
##  6 6                 575
##  7 7                 399
##  8 8                 269
##  9 9                 225
```

```
## 10 10                196
## # ... with 180 more rows
```

From the result, we can see that most people try clicking position 1 first. The amount of people decrease as position order number increase.

Top 5 Change day-to-day

```
(day_to_day=data2 %>%
  filter(action=="visitPage",result_position %in% c(1:5)) %>%
  group_by(year,month,day,result_position) %>%
  summarise(
    n=n()
  ) %>%
  ggplot()+
  geom_line(mapping = aes(x=as.integer(day),y=n,color=result_position))
)
```



It can be seen from the plot that even though there is a fluctuation of amount for top 5 result position, the rank doesn't change

Question 3

3.What is their daily overall zero results rate? How does it vary between the groups?

Daily overall zero results rate

```r
(data3=data_ymd %>%
  filter(action=="searchResultPage") %>%
  group_by(year,month,day) %>%
  summarise(
    zero_result=sum(n_results==0),
    total=n(),
    zero_rate=zero_result/total
    ))
```

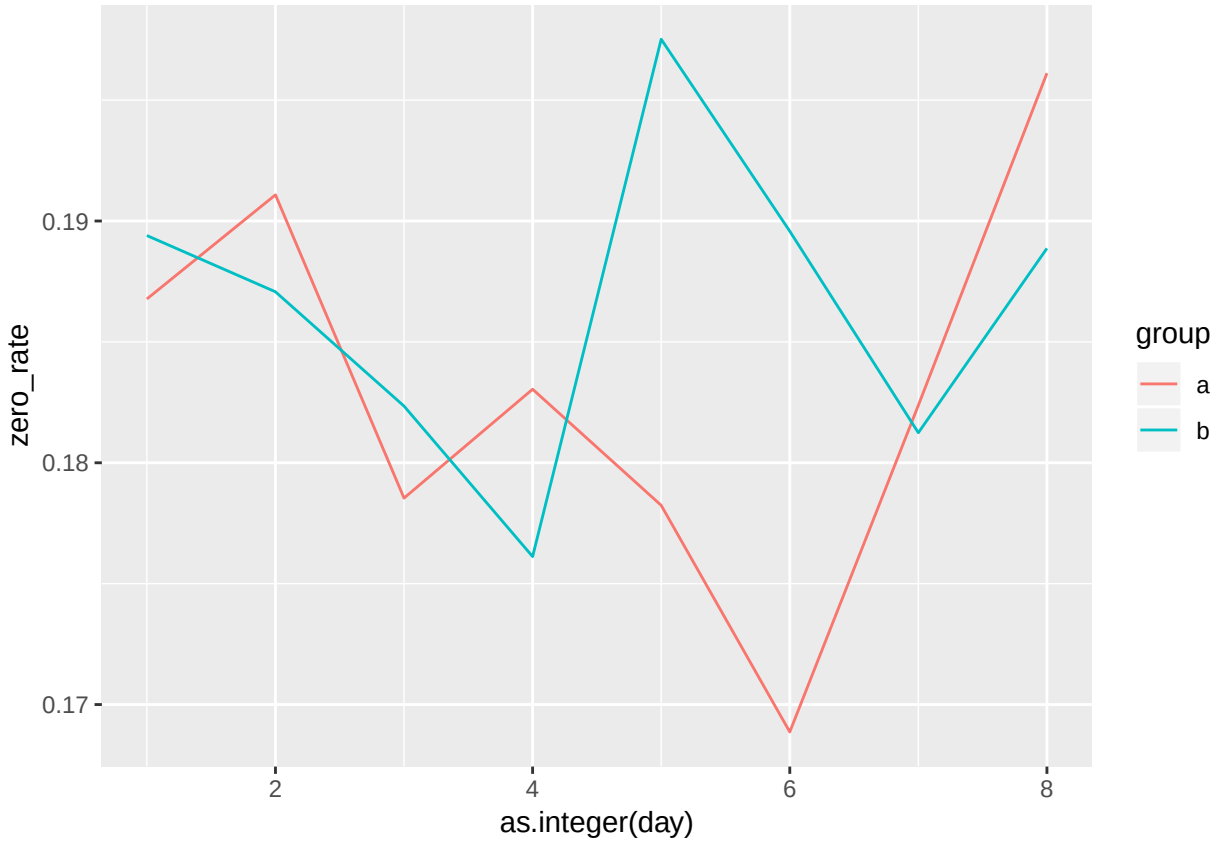
```
## # A tibble: 8 x 6
## # Groups:   year, month [?]
##   year  month day   zero_result total zero_rate
##   <chr> <chr> <chr>       <int> <int>     <dbl>
## 1 2016  03    01           3447 18374     0.188
## 2 2016  03    02           3589 18902     0.190
## 3 2016  03    03           3443 19159     0.180
## 4 2016  03    04           3015 16675     0.181
## 5 2016  03    05           2442 13204     0.185
## 6 2016  03    06           2574 14612     0.176
## 7 2016  03    07           3460 19011     0.182
## 8 2016  03    08           3157 16297     0.194
```

Between groups

```r
(data3_group=data_ymd %>%
  filter(action=="searchResultPage") %>%
  group_by(year,month,day,group) %>%
  summarise(
    zero_result=sum(n_results==0),
    total=n(),
    zero_rate=zero_result/total
    )
)
```

```
## # A tibble: 16 x 7
## # Groups:   year, month, day [?]
##    year  month day   group zero_result total zero_rate
##    <chr> <chr> <chr> <chr>       <int> <int>     <dbl>
##  1 2016  03    01    a            2350 12582     0.187
##  2 2016  03    01    b            1097  5792     0.189
##  3 2016  03    02    a            2524 13209     0.191
##  4 2016  03    02    b            1065  5693     0.187
##  5 2016  03    03    a            2371 13280     0.179
##  6 2016  03    03    b            1072  5879     0.182
##  7 2016  03    04    a            2068 11298     0.183
##  8 2016  03    04    b             947  5377     0.176
##  9 2016  03    05    a            1535  8612     0.178
## 10 2016  03    05    b             907  4592     0.198
## 11 2016  03    06    a            1599  9469     0.169
## 12 2016  03    06    b             975  5143     0.190
## 13 2016  03    07    a            2316 12699     0.182
## 14 2016  03    07    b            1144  6312     0.181
## 15 2016  03    08    a            2139 10907     0.196
## 16 2016  03    08    b            1018  5390     0.189
```

```r
data3_group %>%
  ggplot()+
  geom_line(mapping = aes(x=as.integer(day),y=zero_rate,color=group))
```



Question 4

4.Let session length be approximately the time between the first event and the last event in a session. Choose a variable from the dataset and describe its relationship to session length.

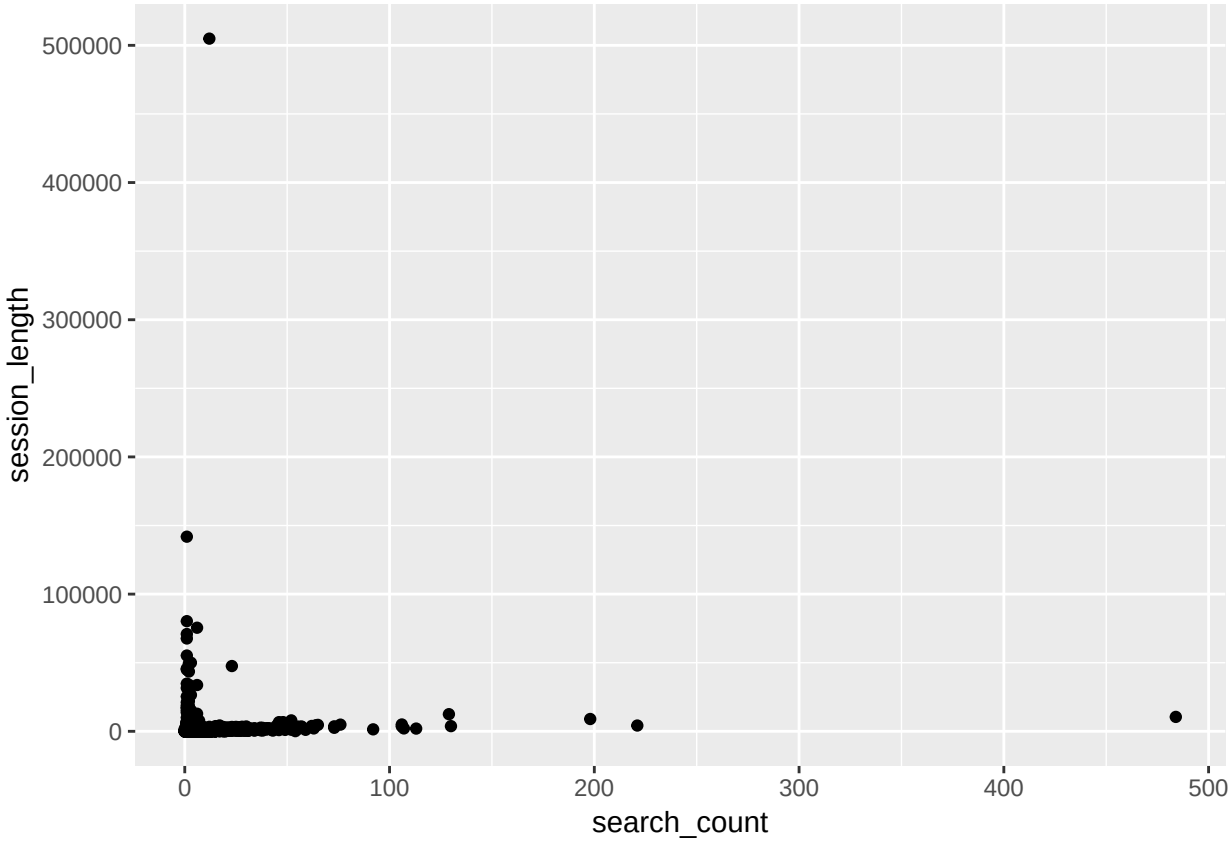
Change timestamp into second unit and calculate session length
Choose total search times for each session as comparable variable

```r
(data4=data %>%
  mutate(
    time=as.integer(substr(timestamp,13,14)) + #second
      as.integer(substr(timestamp,11,12))*60 + #minute
      as.integer(substr(timestamp,9,10))*3600+ #hour
      as.integer(substr(timestamp,7,8))*3600*24 #day
  ) %>%
  arrange(session_id,time) %>%
  group_by(session_id) %>%
  summarise(
    session_length=last(time)-first(time),
    search_count=sum(action=="searchResultPage")
  )
)
```

```
## # A tibble: 68,028 x 3
##    session_id       session_length search_count
##    <chr>                     <dbl>        <int>
##  1 0000cbcb67c19c45            0              1
##  2 0001382e027b2ea4          303              1
##  3 0001e8bb90445cb2          435              1
##  4 000216cf18ae1ab1           58.0            6
##  5 000527f711d50dfc            0              1
##  6 00064fe774048046           43.0            2
##  7 00071a2cf97168df            0              1
##  8 0007582fe23d51e6            0              1
##  9 0007b7f6b575feb6          339              1
## 10 00086b6ff8156928            0              1
## # ... with 68,018 more rows
```

Try scatter plot

```
ggplot(data=data4,mapping = aes(x=search_count,y=session_length))+
  geom_point()
```



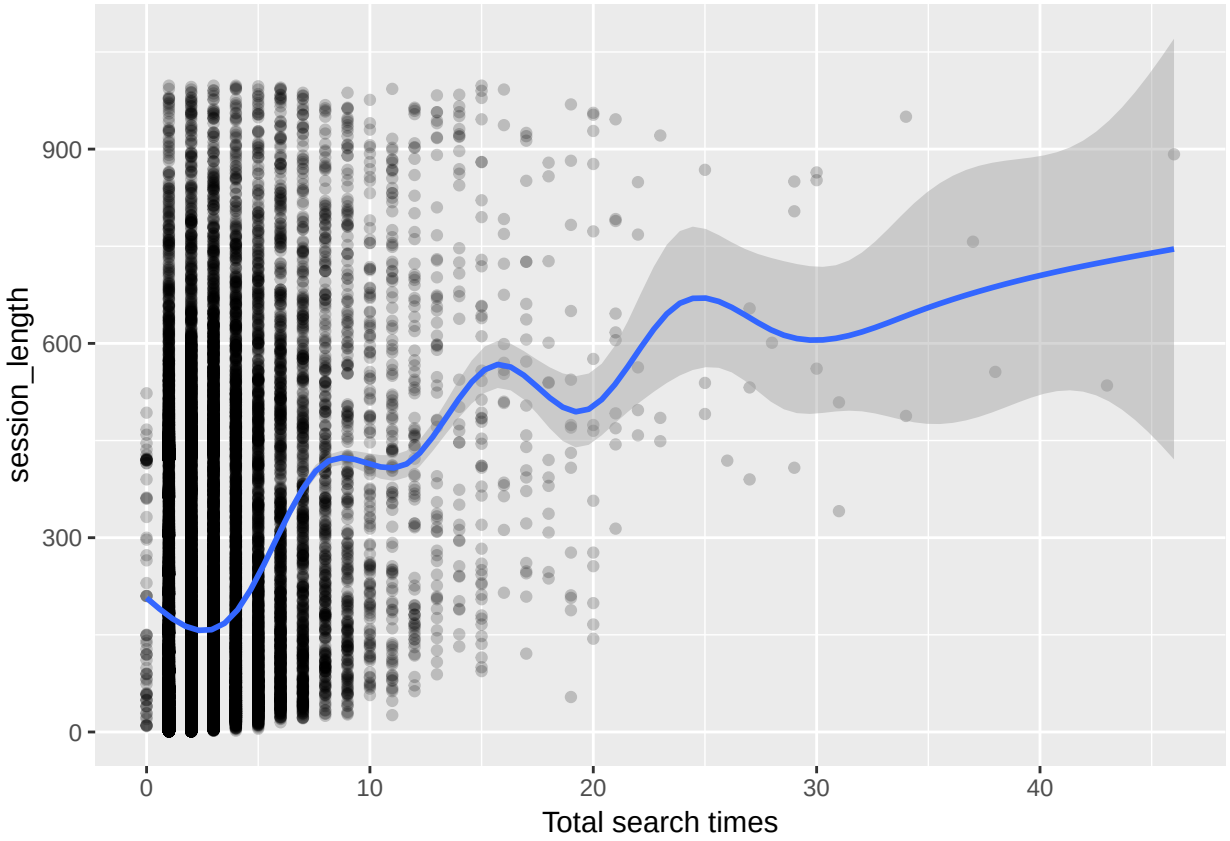
It seems that there are some outlies. Remove them and plot again

```
data4_a=data4 %>%
  filter(session_length < 1000 & session_length!=0,search_count < 50)

(data4_plot=data4_a %>%
  ggplot(mapping = aes(x=search_count,y=session_length))+
  geom_point(alpha=0.2)+
```

```
  geom_smooth()+
  xlab("Total search times"))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



It can be senn from the plot that session length and total search times have a roughly positive relationship

Question 5

Summarize your findings in an executive summary

1.Daily overall clickthrough rate

```
data2_overall
```

```
## # A tibble: 8 x 6
## # Groups:   year, month [?]
##   year  month day   visitPage_count searchResultPage_count    cr
##   <chr> <chr> <chr>           <int>                  <int> <dbl>
## 1 2016  03    01               4364                  18374 0.238
## 2 2016  03    02               4476                  18902 0.237
## 3 2016  03    03               4704                  19159 0.246
## 4 2016  03    04               4189                  16675 0.251
## 5 2016  03    05               3251                  13204 0.246
## 6 2016  03    06               3678                  14612 0.252
## 7 2016  03    07               4598                  19011 0.242
## 8 2016  03    08               3932                  16297 0.241
```

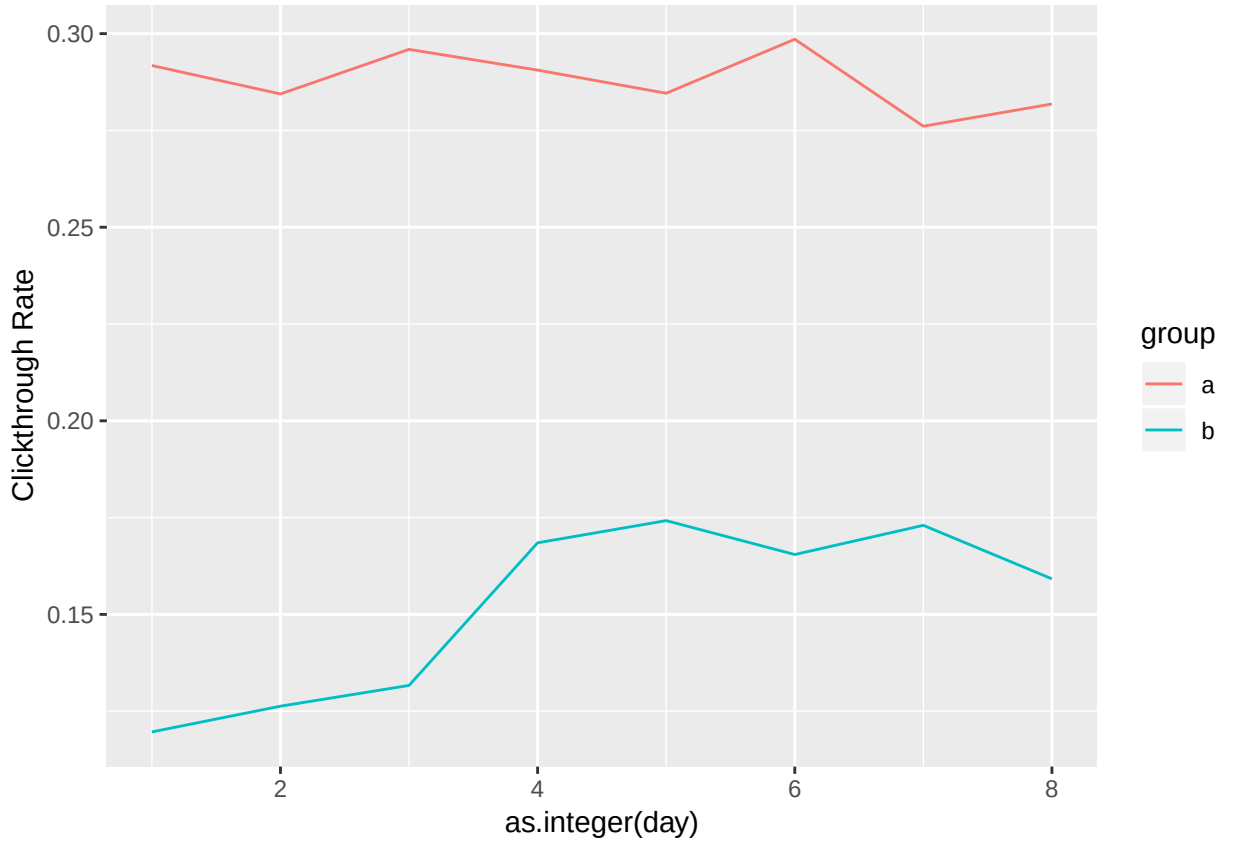
It shows that daily overall clickthrough rate doesn't change a lot over days

Vary between groups

```
data2_group
```

```
## # A tibble: 16 x 7
## # Groups:   year, month, day [8]
##    year  month day   group visitPage_count searchResultPage_count    cr
##    <chr> <chr> <chr> <chr>           <int>                  <int> <dbl>
##  1 2016  03    01    a                3671                  12582 0.292
##  2 2016  03    01    b                 693                   5792 0.120
##  3 2016  03    02    a                3757                  13209 0.284
##  4 2016  03    02    b                 719                   5693 0.126
##  5 2016  03    03    a                3930                  13280 0.296
##  6 2016  03    03    b                 774                   5879 0.132
##  7 2016  03    04    a                3283                  11298 0.291
##  8 2016  03    04    b                 906                   5377 0.168
##  9 2016  03    05    a                2451                   8612 0.285
## 10 2016  03    05    b                 800                   4592 0.174
## 11 2016  03    06    a                2827                   9469 0.299
## 12 2016  03    06    b                 851                   5143 0.165
## 13 2016  03    07    a                3506                  12699 0.276
## 14 2016  03    07    b                1092                   6312 0.173
## 15 2016  03    08    a                3074                  10907 0.282
## 16 2016  03    08    b                 858                   5390 0.159
```

```
ggplot(data=data2_group)+
  geom_line(mapping = aes(x=as.integer(day),y=cr,color=group))+
  ylab("Clickthrough Rate")
```



The clickthrough rate of group a is obviously larger than group b

2.Results that people try first

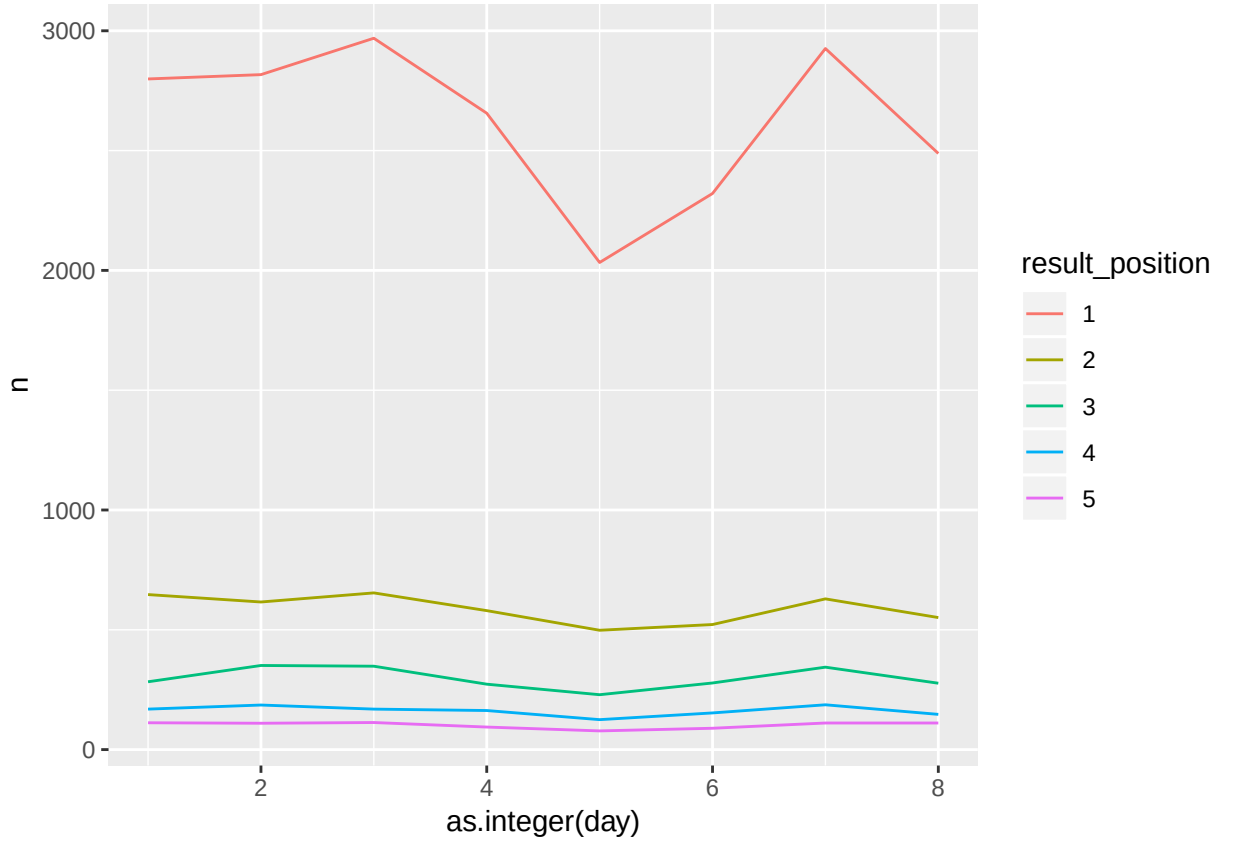
```
head(data_first,10)
```

```
## # A tibble: 10 x 2
##    result_position     n
##    <fct>           <int>
##  1 1               21009
##  2 2                4697
##  3 3                2383
##  4 4                1299
##  5 5                 818
##  6 6                 575
##  7 7                 399
##  8 8                 269
##  9 9                 225
## 10 10                196
```

From the result, we can see that most people try clicking position 1 first. The amount of people decrease as position order number increase.

Change day to day

```
day_to_day
```



It can be seen from the plot that even though there is a fluctuation of amount for top 5 result position, the rank doesn't change

3.Overall zero results rate

```
data3
```

```
## # A tibble: 8 x 6
## # Groups:   year, month [?]
##   year  month day   zero_result total zero_rate
##   <chr> <chr> <chr>       <int> <int>     <dbl>
## 1 2016  03    01           3447 18374     0.188
## 2 2016  03    02           3589 18902     0.190
## 3 2016  03    03           3443 19159     0.180
## 4 2016  03    04           3015 16675     0.181
## 5 2016  03    05           2442 13204     0.185
## 6 2016  03    06           2574 14612     0.176
## 7 2016  03    07           3460 19011     0.182
## 8 2016  03    08           3157 16297     0.194
```

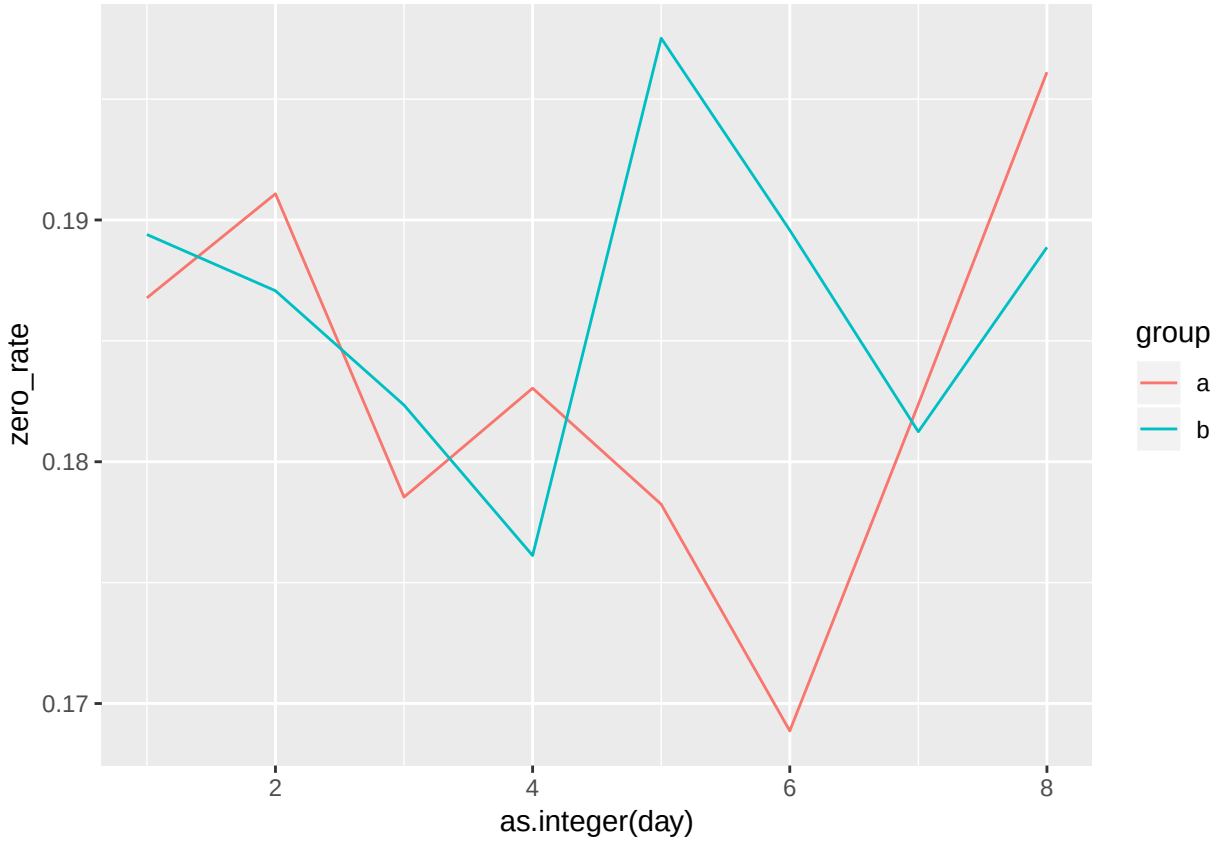
Vary between groups

```
data3_group
```

```
## # A tibble: 16 x 7
## # Groups:   year, month, day [8]
##    year  month day   group zero_result total zero_rate
```

```
##    <chr> <chr> <chr> <chr>       <int> <int>     <dbl>
##  1 2016  03    01    a            2350 12582     0.187
##  2 2016  03    01    b            1097  5792     0.189
##  3 2016  03    02    a            2524 13209     0.191
##  4 2016  03    02    b            1065  5693     0.187
##  5 2016  03    03    a            2371 13280     0.179
##  6 2016  03    03    b            1072  5879     0.182
##  7 2016  03    04    a            2068 11298     0.183
##  8 2016  03    04    b             947  5377     0.176
##  9 2016  03    05    a            1535  8612     0.178
## 10 2016  03    05    b             907  4592     0.198
## 11 2016  03    06    a            1599  9469     0.169
## 12 2016  03    06    b             975  5143     0.190
## 13 2016  03    07    a            2316 12699     0.182
## 14 2016  03    07    b            1144  6312     0.181
## 15 2016  03    08    a            2139 10907     0.196
## 16 2016  03    08    b            1018  5390     0.189
```

```
data3_group %>%
  ggplot()+
  geom_line(mapping = aes(x=as.integer(day),y=zero_rate,color=group))
```

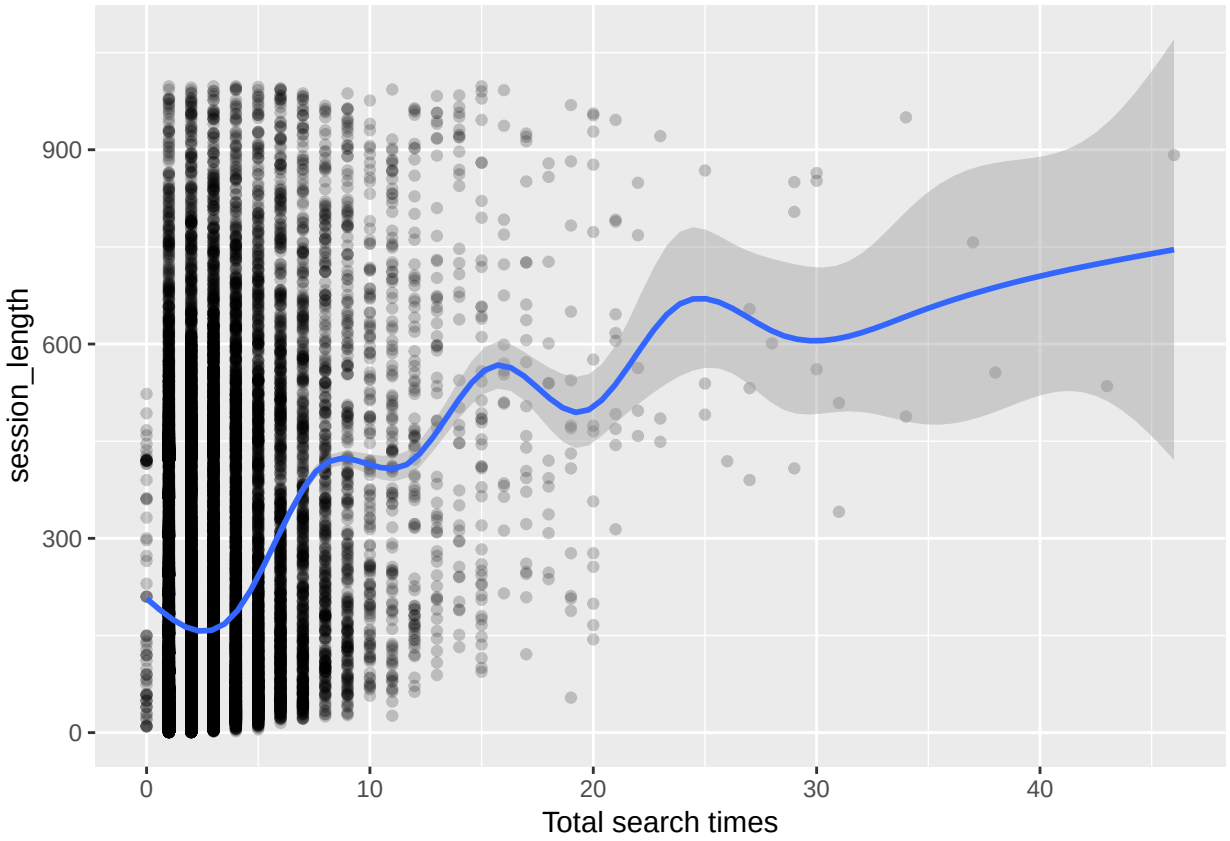


It can be seen that zero rate of two group has a alternating pattern

4.total search times v.s. session length

```
data4_plot
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



It can be senn from the plot that session length and total search times have a roughly positive relationship but not so strong