

Video Analytics

INFO 5200 Learning Analytics: Week 10 Homework

[[Cole Walsh, 4399966]]

In this homework, you will learn how to analyze video interaction data to generate video analytics. You are given timestamped video interaction events for this video from an online course: <https://youtu.be/qKNb8YQYTZg>.

Dataset columns:

- id is a hashed learner id
- time is the exact time of the event
- order counts observed events in temporal for each id
- seconds measures time between events for the same person
- event_type
- video_current_time is the time in the video the event occurred
- video_new/old_speed is for tracking speed_change events
- video_old/new_time is for tracking seek events

Learning Objectives:

1. Exploring the structure of video interaction data
2. Identifying parts of the video with increased activity
3. Deciding what video analytics to report back to learners and instructors

Part 1: Explore video data

Question 1: What event types are in the dataset and how many of each?

```
#####  
##### BEGIN INPUT: Question 1 #####  
#####
```

```
table(vid$event_type)
```

```
##  
##      load_video      pause_video      play_video  
##          3060          7572          12502  
##      seek_video speed_change_video      stop_video  
##          8008          172          587
```

```
#####  
#####
```

Question 2: How many users watched the video at all? How many video events do users have on average?

```
#####
##### BEGIN INPUT: Question 2 #####
#####

paste(length(unique(vid$id)), 'users watched the video', sep = ' ')

## [1] "1544 users watched the video"
```

```
paste('Users have', vid %>%
  group_by(id) %>%
  summarize(N.Events = n()) %>%
  summarize(Avg.Events = round(mean(N.Events), 1)) %>%
  pull(), 'video events, on average', sep = ' ')

## [1] "Users have 20.7 video events, on average"
```

```
#####
#####
```

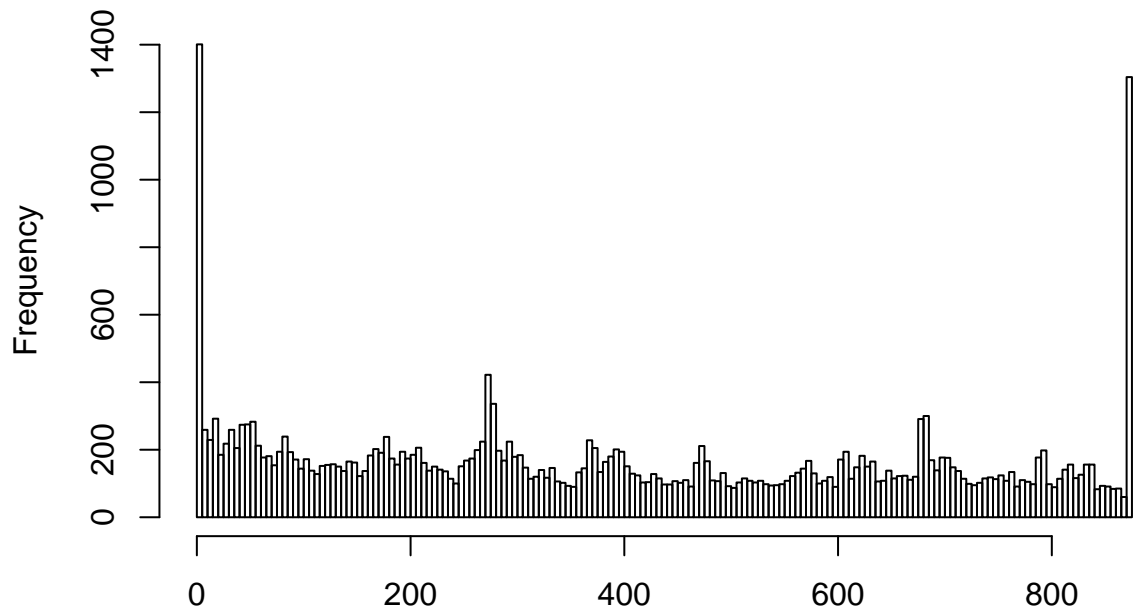
Question 3: Using a histogram, plot how much activity occurred throughout the video in total (x-axis = time in video; y-axis = event count). Tip: Set `breaks=300` in the `hist()` function for better resolution. Note that `seek_video` events have the `video_current_time` field missing so you should use `video_old_time` instead.

You should see a spike at the start, the end, one at around 300, and another around 700.

```
#####
##### BEGIN INPUT: Question 3 #####
#####

hist(ifelse(is.na(vid$video_current_time), vid$video_old_time, vid$video_current_time), breaks = 300)
```

of `ifelse(is.na(vid$video_current_time), vid$video_old_time, vid$video`



`ifelse(is.na(vid$video_current_time), vid$video_old_time, vid$video_current_time)`

```
#####  
#####
```

Question 4: At what point in the video do many learners pause? Using a precision of 10 seconds, report the 3 most common times in seconds. Then go to the video lecture and look up what happens at those times (note that you can type in the seconds in this URL so you don't need to convert it). Say why you think learners paused there.

```
#####  
##### BEGIN INPUT: Question 4 #####  
#####
```

```
library(plyr)
```

```
## -----
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)
```

```
## -----
```

```
##  
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
```

```
## The following object is masked from 'package:purrr':
##
##   compact
```

```
vid %>%
  ungroup() %>%
  filter(event_type == 'pause_video') %>%
  select(video_current_time) %>%
  sapply(., function(x) round_any(x, 10)) %>%
  table(.) %>%
  data.frame(.) %>%
  arrange(desc(Freq)) %>%
  top_n(., 3) %>%
  `colnames<-`(c("Time Paused", "Frequency"))
```

```
## Selecting by Freq
```

```
##   Time Paused Frequency
## 1         870        608
## 2         270        210
## 3         680        199
```

```
#####
#####
```

The video is paused most frequently at 870s, then 270s, then 680s. 870s corresponds to the end of the video; learners likely pause here and either reflect or go back to previous parts of the video. At around 270s, an insertion is performed and the slides flip from the command to the results; learners likely pause here (and may go back) to reflect on the relationship between the code and the result. A similar thing happens at 680s, but with an update command rather than an insert.

Question 5: At what point in the video do many learners seek forwards and when do they seek backwards in the video (ignore for now where they seek to)? Using a precision of 10 seconds, report the 3 most common times in seconds for moving back, 3 most common for moving forward. Then go to the video lecture and look up what happens at those times. Why might learners move?

```
#####
##### BEGIN INPUT: Question 5 #####
#####
```

```
vid %>%
  ungroup() %>%
  filter((event_type == 'seek_video') & (video_new_time > video_old_time)) %>%
  select(video_old_time) %>%
  sapply(., function(x) round_any(x, 10)) %>%
  table(.) %>%
  data.frame(.) %>%
  arrange(desc(Freq)) %>%
```

```
top_n(., 3) %>%
  `colnames<-`(c("Time Seek Forwards", "Frequency"))
```

Selecting by Freq

```
##   Time Seek Forwards Frequency
## 1                   0         169
## 2                  180          90
## 3                   40          83
```

```
vid %>%
  ungroup() %>%
  filter((event_type == 'seek_video') & (video_new_time < video_old_time)) %>%
  select(video_old_time) %>%
  sapply(., function(x) round_any(x, 10)) %>%
  table(.) %>%
  data.frame(.) %>%
  arrange(desc(Freq)) %>%
  top_n(., 3) %>%
  `colnames<-`(c("Time Seek Backwards", "Frequency"))
```

Selecting by Freq

```
##   Time Seek Backwards Frequency
## 1                   280         149
## 2                    50         101
## 3                   190          97
## 4                   690          97
```

```
#####
#####
```

Learners seek forwards most frequently at 0s, then 180s, then 40s. At 0s, the video is just starting, there is an overview of what will be covered, but its over 2min later before any examples appear. At 180s, the first example is run; its relatively straightforward compared to later examples and may be something learners already feel comfortable with. 40s is, again, right in the middle of the video overview; there's no examples here so learners likely realize this a little later and skip it.

Learners seek backwards most frequently at 280s, then 50s, then (tied) 190s and 690s. 280s and 690, as discussed above, are right after the first somewhat complicated insert and update examples are used; learners likely go back seeing these examples to see them again. At 50s, the slides flipped from covering insert to delete statements; learners that needed to see the insert statement overview again likely went back. At 190s, the first example was just run; learners that weren't paying attention after the long intro or needed to see the example again likely seeked backward.

Question 6: At what point in the video do many learners seek backwards in the video and where do they go? Using a precision of 10 seconds, report the 3 most common pairs of times in seconds like this <from, to>. Then go to the video lecture and look up what happens at those times. Why do you think learners moved there?

```
#####
##### BEGIN INPUT: Question 6 #####
#####

vid %>%
  ungroup() %>%
  filter((event_type == 'seek_video') & (video_new_time < video_old_time)) %>%
  select(video_old_time, video_new_time) %>%
  lapply(., function(x) round_any(x, 10)) %>%
  data.frame(.) %>%
  mutate(Time.Pairs = paste(video_old_time, video_new_time, sep = ',')) %>%
  select(Time.Pairs) %>%
  table(.) %>%
  data.frame(.) %>%
  arrange(desc(Freq)) %>%
  top_n(., 3) %>%
  `colnames<-`(c("Pairs of From-To Backward Seeks", "Frequency"))
```

```
## Selecting by Freq
```

```
## Pairs of From-To Backward Seeks Frequency
## 1 280,270 68
## 2 690,680 46
## 3 50,40 38
```

```
#####
#####
```

The most common <From, To> backward seeks are <280s, 270s>, then <690s, 680s>, then <50s, 40s>. We have previously discussed these points in the video. <280s, 270s> and <690s, 680s> denote points where a command was run and then results were shown; learners seeked backward to compare again the code and the output. <50s, 40s> denotes learners who back to the slide giving the overview of the insert command after the slides flipped (maybe because they didn't have enough time to take in that material).

Question 7: Are students more likely to speed the video up or slow it down? Report the proportion speed-ups relative to all speed changes. How do you interpret this?

```
#####
##### BEGIN INPUT: Question 7 #####
#####

paste(vid %>%
  filter(!is.na(video_new_speed)) %>%
  mutate(Speed.Up = video_new_speed > video_old_speed) %>%
  summarize(round(mean(Speed.Up), 3)) %>%
  pull(), 'of all speed changes are speed-ups.', sep = ' ')
```

```
## [1] "0.576 of all speed changes are speed-ups."
```

```
#####
#####
```

Students are more likely to speed the video up than slow it down. This may indicate that the material is boring or that some students are already familiar with it. The relative number of speed changes overall (172) is quite small though, so this should be taken with a grain of salt.

Question 8: Based on the video analytics you have done (or any additional analyses you'd like to do), what information would you give (a) a learner about to watch this lecture video, and (b) the instructor of the lecture video? Make empirically grounded recommendations for each stakeholder.

- **Recommendation to learners:** At 270s and 680s there are two examples that are run with the insert and update commands. Results are quickly shown and so you may need to go back to see the relationship between the code and the output. There is also about 2min of introductory material at the beginning of the video that you may feel like you want to skip.
- **Recommendation to instructor:** Potentially show code and output side by side to reduce the flipping back and forth. One could also give students more time to process examples, but they have the ability to pause and go back so this isn't a big deal. In the intro, give more time for students to understand the insert command syntax.

Self-reflection

Briefly summarize your experience on this homework. What was easy, what was hard, what did you learn?

Straightforward. Much easier and less time consuming than the previous homework. I played around with pulling data from dataframes and pasting output into a string because I thought it looked nicer.

Submit Homework

This is the end of the homework. Please **Knit a PDF report** that shows both the R code and R output and upload it on the EdX platform. Alternatively, you can Knit it as a “doc”, open it in Word, and save that as a PDF.

Important: Be sure that all your code is visible. If the line is too long, it gets cut off. If that happens, organize your code on several lines.