

# Databrary Weekly

*Rick O. Gilmore & Andrea Seisler*

*2019-08-09 11:44:31*

## Contents

<b>Institutions &amp; Authorized Users</b>	<b>1</b>
New investigators . . . . .	3
New institutions . . . . .	4
<b>Volumes</b>	<b>4</b>
New volumes . . . . .	5
<b>Data about volumes with videos</b>	<b>6</b>
Number of videos in (shared) volumes . . . . .	7
Number of sessions in (shared) volumes . . . . .	8
Total video hours in (shared) volumes . . . . .	8

## Institutions & Authorized Users

```
new_stats <- databraryapi::get_db_stats()
new_stats$date <- lubridate::as_datetime(new_stats$date)
```

Let's try a Google Sheets-centered workflow. If this is the first time you are rendering this document in your current work session, please run this command to authenticate to Google from the command line, that is **outside of RMarkdown**:

```
db <- googlesheets::gs_title('Databrary-analytics')
#key <- "1tvLIQzULrMtXo97aJu71ljdTmNXkwwpU9e00asVer3g"
db <- gs_title('Databrary-analytics')
```

Now, let's load the data about the number of institutions and investigators.

```
old_stats <- db %>%
  gs_read(ws = 'institutions-investigators')
```

We then update the old stats with new data if `params$update_stats` is TRUE. In the current context, `params$update_stats == FALSE`.

```
# initialize updated_stats
updated_stats <- old_stats
if (as.logical(params$update_stats)) {
  next_entry <- dim(updated_stats)[1] + 1
  updated_stats[next_entry,] = NA
  updated_stats <- updated_stats

  # fill with new data
  updated_stats$date[next_entry] <- new_stats$date
  updated_stats$institutions[next_entry] <- new_stats$institutions
  updated_stats$investigators[next_entry] <- new_stats$investigators
```

```

  updated_stats$affiliates[next_entry] <- new_stats$affiliates
}

```

Now, we plot the data.

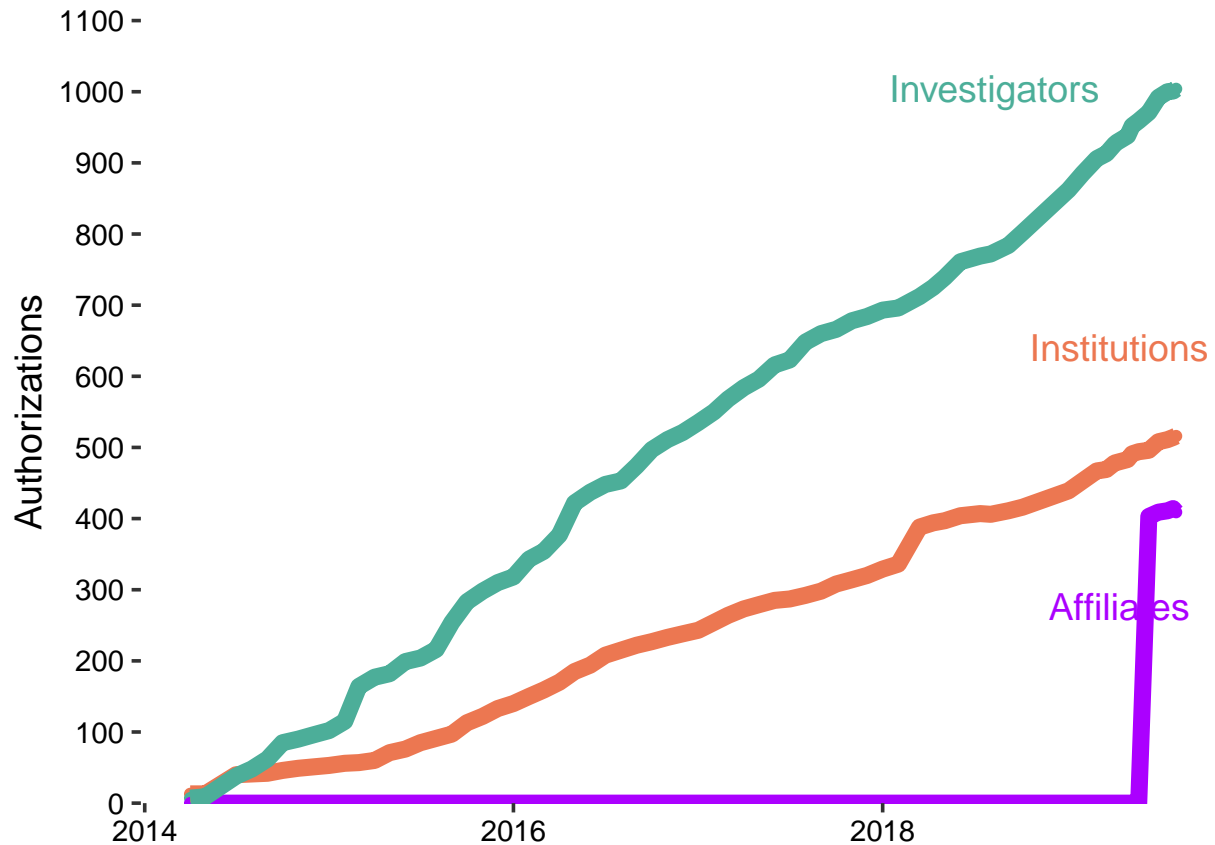
```

# Create a tidy form for plotting both institutions and investigators and affiliates
updated_stats <- updated_stats %>%
  gather(., key = "type", value = "count", -date) %>%
  mutate(date = lubridate::as_date(date)) %>%
  select(date, count, type) %>%
  filter(type %in% c('institutions', 'investigators', 'affiliates')) %>%
  filter(!is.na(count))

# Plot
p <- updated_stats %>%
  ggplot(., aes(x = date, y = count, color = type, group = type)) +
  geom_point() +
  geom_line(size = ln_size) +
  scale_colour_manual(values=c(color_purple, color_orange, color_teal)) +
  ylab("Authorizations") +
  databrary_theme +
  scale_y_continuous(breaks = seq(0, round_any(max(updated_stats$count), 100, ceiling), 100), expand = c(0, 1)) +
  coord_cartesian(ylim = c(0, round_any(max(updated_stats$count), 100, ceiling)))

ggdraw(p) +
  draw_label("Investigators", colour = color_teal, .8, .9) +
  draw_label("Institutions", colour = color_orange, .9, .6) +
  draw_label("Affiliates", colour = color_purple, .9, .3)

```



Next, we update the Google Sheet if `params$update_gs` is TRUE. In the current context, `params$update_gs == FALSE`.

```
if (as.logical(params$update_gs)) {
  db <- db %>%
    gs_add_row(ws = 'institutions-investigators', input = new_stats[,c(1, 4, 2, 3)])
  message("'update_gs' parameter is 'TRUE', so Google Sheet data will be updated.")
} else {
  message("'update_gs' parameter is 'FALSE', so Google Sheet data unmodified.")
}
```

## New investigators

```
new_people <- databraryapi::get_db_stats(type = "people")
new_people %>%
  mutate(url = paste0("https://databrary.org/party/", id)) %>%
  select(., sortname, prename, affiliation, url) %>%
  knitr::kable()
```

sortname	prename	affiliation	url
Lei	Ryan	Haverford College	<a href="https://databrary.org/party/5676">https://databrary.org/party/5676</a>
Kalpidou	Maria	Assumption College	<a href="https://databrary.org/party/5686">https://databrary.org/party/5686</a>
Sutherland	Clare	University of Western Australia	<a href="https://databrary.org/party/2285">https://databrary.org/party/2285</a>
St Loe	Stephanie	Kwantlen Polytechnic University	<a href="https://databrary.org/party/5529">https://databrary.org/party/5529</a>
Bernstein	Michael	Penn State University	<a href="https://databrary.org/party/982">https://databrary.org/party/982</a>

sortname	prename	affiliation	url
----------	---------	-------------	-----

## New institutions

```
new_institutions <- databraryapi::get_db_stats(type = "institutions")
new_institutions %>%
  mutate(db_url = paste0("https://databrary.org/party/", id)) %>%
  select(., sortname, url, db_url) %>%
  knitr::kable()
```

sortname	url	db_url
Haverford College	https://www.haverford.edu/	https://databrary.org/party/5722
Assumption College	https://www.assumption.edu/	https://databrary.org/party/5720
Kwantlen Polytechnic University	https://www.kpu.ca/	https://databrary.org/party/5711

## Volumes

Let's try a new workflow based on Google Sheets. We should already have the spreadsheet loaded.

```
# Read from Google Sheet
old_vols <- db %>%
  gs_read(ws = 'volumes-shared-unshared')
```

Now update from information derived from `databraryapi::db_stats()` if `params$update_stats` is TRUE.

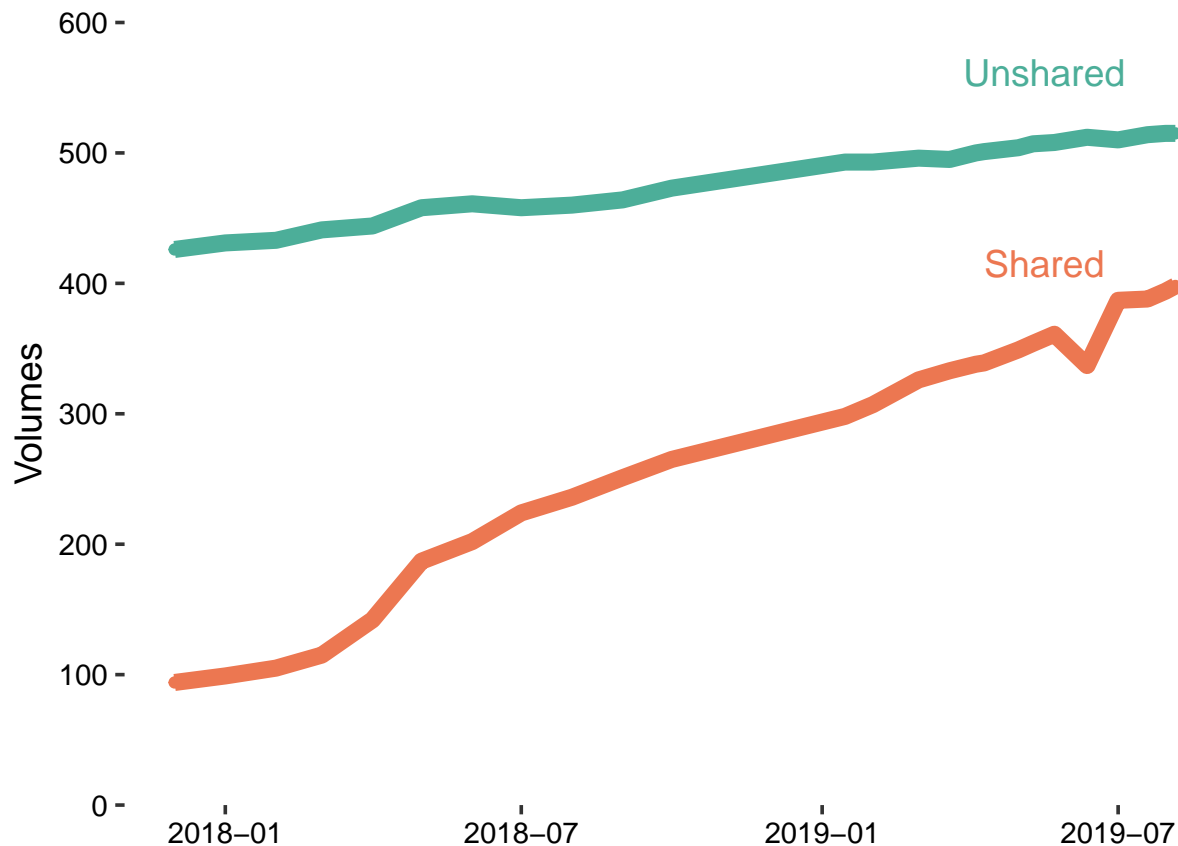
```
updated_vols <- old_vols
if (as.logical(params$update_stats)) {
  next_entry <- dim(updated_vols)[1] + 1
  updated_vols[next_entry,] = NA

  updated_vols$date[next_entry] <- new_stats$date
  if (is.null(new_stats$datasets_shared)) {
    new_stats$datasets_shared = 0
  }
  updated_vols$shared_volumes[next_entry] <- new_stats$datasets_shared
  updated_vols$unshared_volumes[next_entry] <-
    new_stats$datasets_total - new_stats$datasets_shared
}
updated_vols <- updated_vols %>%
  gather(., key = "type", value = "count", -date)
```

```
# Plot
vols_plot <- updated_vols %>%
  ggplot(., aes(x = date, y = count, color = type, group = type)) +
  geom_point() +
  geom_line(size=ln_size) +
  scale_colour_manual(values=c(color_orange, color_tea)) +
  ylab("Volumes") +
  databrary_theme +
  scale_y_continuous(breaks = seq(0, round_any(max(updated_vols$count), 100, ceiling), 100), expand = c
```

```
coord_cartesian(ylim = c(0, round_any(max(updated_vols$count), 100, ceiling)))

ggdraw(vols_plot) +
  draw_label("Unshared", colour = color_teal, .84, .92) +
  draw_label("Shared", colour = color_orange, .84, .70)
```



Next, we update the Google Sheet if `params$update_gs` is TRUE.

```
if (as.logical(params$update_gs)) {
  new_data <- data_frame(date = Sys.Date(),
                        shared_volumes = new_stats$datasets_shared,
                        unshared_volumes = new_stats$datasets_total - new_stats$datasets_shared)

  db <- db %>%
    gs_add_row(ws = 'volumes-shared-unshared', input = new_data)
} else {
  message("'update_gs' parameter is 'false', so Google Sheet data unmodified.")
}
```

## New volumes

```
# define helper functions
new_volumes <- databraryapi::get_db_stats(type = "datasets")
if (is.null(new_volumes)) {
  stop('New volumes data not downloaded.')
```

```

}

unnested_vols <- new_volumes %>%
  unnest(.)
  # rename(., owner_name = name1, owner_id = id1) %>%

unnested_vols$owner_name <- unnested_vols$name1
unnested_vols$owner_id = unnested_vols$name1

unnested_vols <- unnested_vols %>%
  mutate(., url = paste0("https://nyu.databrary.org/volume/", id),
    date_created = lubridate::as_date(creation))

unnested_vols %>%
  select(., name, date_created, owner_name, url) %>%
  knitr::kable()

```

name	date_c
Oh, Behave!	2019-0
Oh, Behave!	2019-0
ELP PLAY 2019	2019-0
ELP PLAY 2019	2019-0
Example Videos - Pupillometry as a Window on the Role of Motionese in Infants' Processing of Dynamic Activity	2019-0
Kosie & Baldwin Video Corpus	2019-0

## Data about volumes with videos

```

# TODO: Fix this. It's hacky and awful.

# List files with metadata
csv_files <- list.files(path = "csv", pattern = "vol_", full.names = TRUE)

# Extract vector of volume numbers, determine the maximum
vol_nums <- stringr::str_match(csv_files, pattern = "_([0-9]+)\\.")
vol_ids <- as.numeric(vol_nums[,2])
last_vol <- max(vol_ids)
vols_to_test <- params$vols_to_test

# Create function to write new data files for each volume
write_vid_csv <- function(vol.id = 1) {
  message(paste0("Getting data for volume ", vol.id))
  vid_dat <- get_video_stats(vol.id)
  # This .Rmd file is already in working/
  if (!is.null(vid_dat)) {
    write.csv(vid_dat, file = paste0("csv/vol_", vol.id, ".csv"),
      row.names = FALSE)
  }
}

vol_files <- list.files("csv", pattern = "vol_[0-9]+", full.names = TRUE)

```

```
# Import the individual csv files
video_data <- lapply(vol_files, read_csv)
video_stats <- Reduce(function(x,y) merge(x, y, all = TRUE), video_data)
```

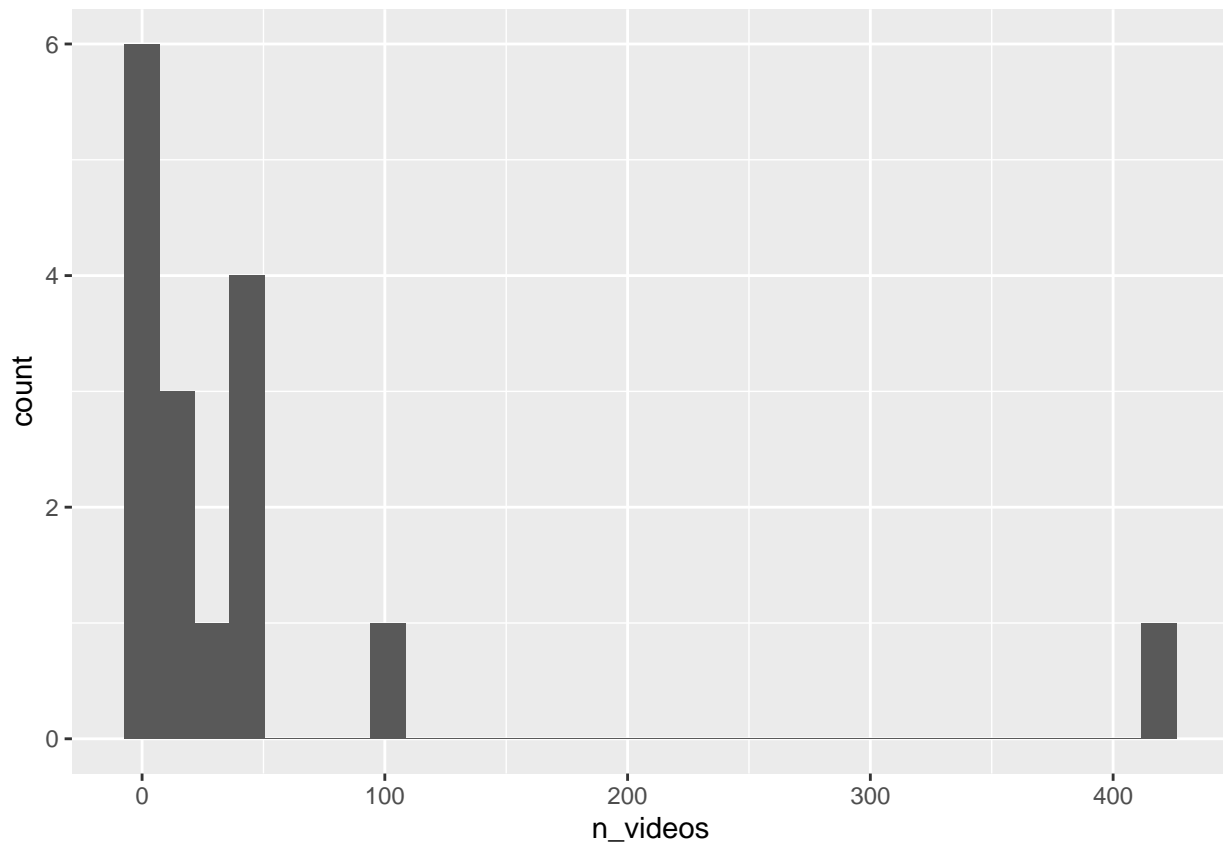
The median number of videos per volume is 16 with a range of [1, 4368].

```
vols <- video_stats$vol_id
vol_info <- lapply(vols, list_volume_metadata)
vols_data <- Reduce(function(x,y) merge(x, y, all = TRUE), vol_info)
vols_joined <- dplyr::left_join(vols_data, video_stats, by = c("vol_id" = "vol_id"))
```

There are 16 volumes with DOIs (shared) that have sessions and at least one video as of today (2019-08-09 11:44:56).

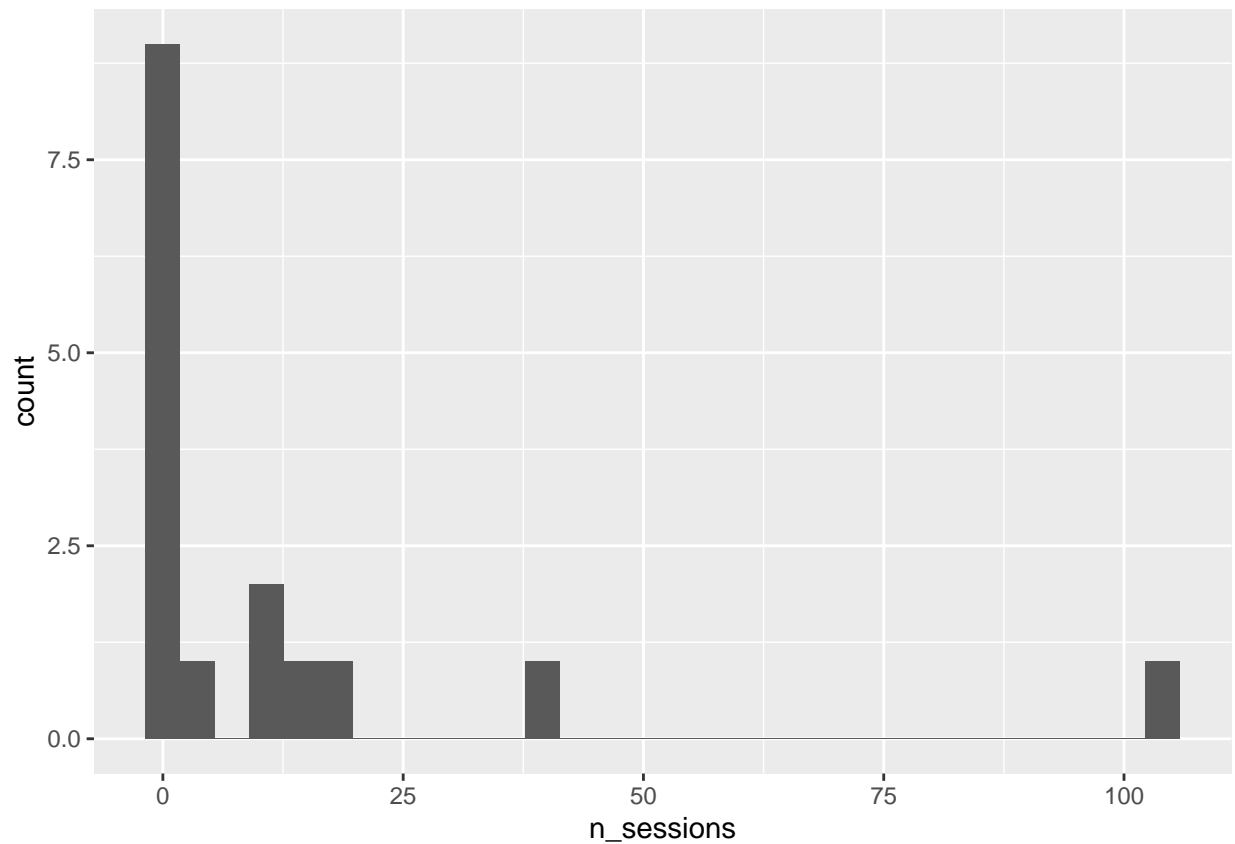
## Number of videos in (shared) volumes

```
vols_joined %>%
  filter(!is.na(doi)) %>%
  ggplot(.) +
  aes(x=n_videos)+
  geom_histogram()
```



## Number of sessions in (shared) volumes

```
vols_joined %>%  
  filter(!is.na(doi)) %>%  
  ggplot(.) +  
  aes(x=n_sessions)+  
  geom_histogram()
```



## Total video hours in (shared) volumes

```
vols_joined %>%  
  filter(!is.na(doi)) %>%  
  ggplot(.) +  
  aes(x=tot_hrs)+  
  geom_histogram()
```



