datavyu R package update

Joshua Rosenberg, Mekenzie Meadows, and the Human Analysis Team

$T(CA)^2$

2020/08/10

Outline

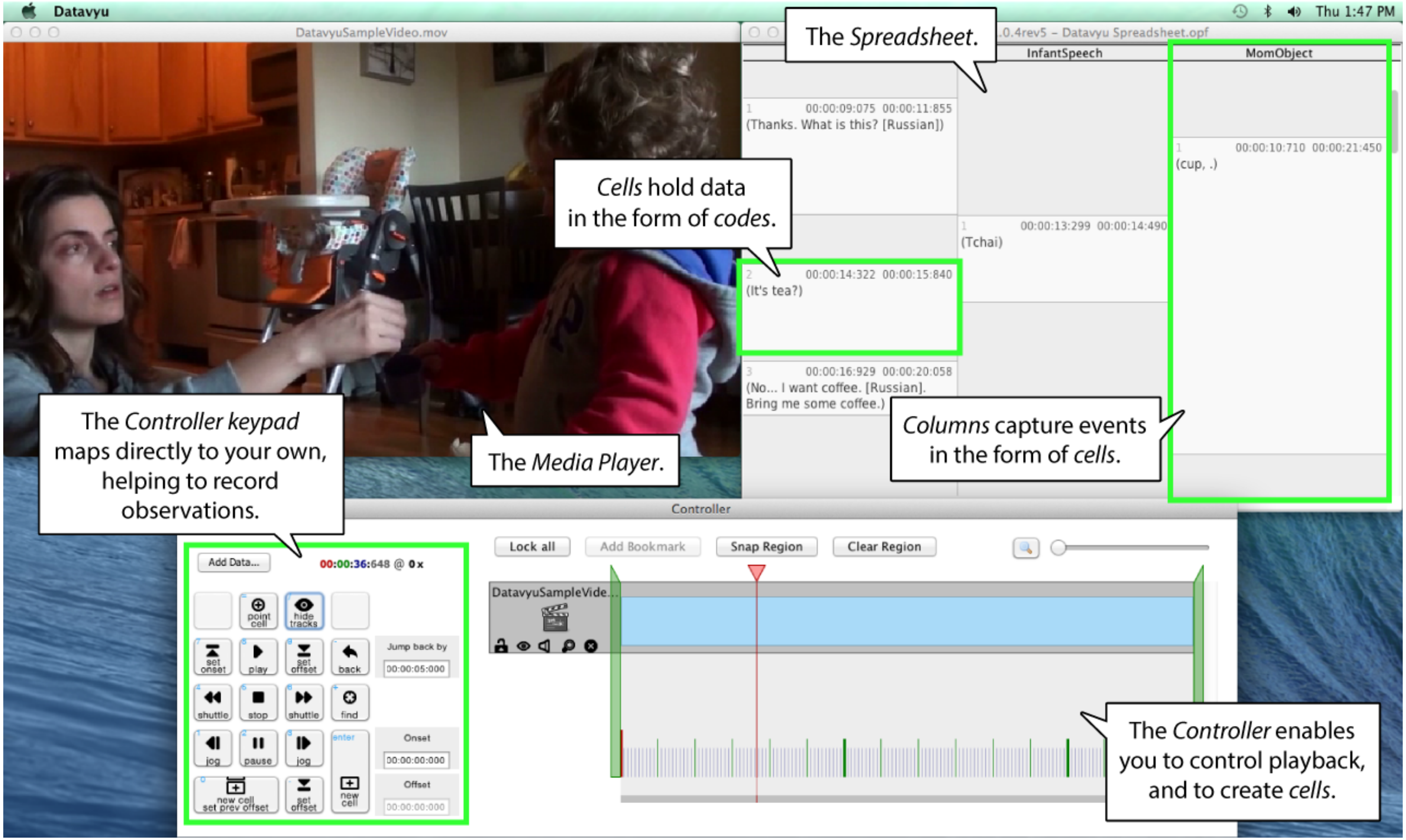
1. Background
2. Summarizing a column
3. Plotting a column summary
4. Preparing and plotting time series data
5. Next steps & discussion

1. Background

What we've been doing

- Learning about using datavyu
- Learning about an already-existing R package, {datavyur}
- Developing a new R package focused on preparing datavyu output for subsequent analyses and summarizing and plotting the prepared data ({datavyu})

A look at datavyu (the qualitative audiovisual coding software)



1 00:00:09:075 00:00:11:855
(Thanks. What is this? [Russian])

2 00:00:14:322 00:00:15:840
(It's tea?)

3 00:00:16:929 00:00:20:058
(No... I want coffee. [Russian]. Bring me some coffee.)

InfantSpeech

1 00:00:13:299 00:00:14:490
(Tchai)

MomObject

1 00:00:10:710 00:00:21:450
(cup, .)

The *Spreadsheet*.

Cells hold data in the form of *codes*.

Columns capture events in the form of *cells*.

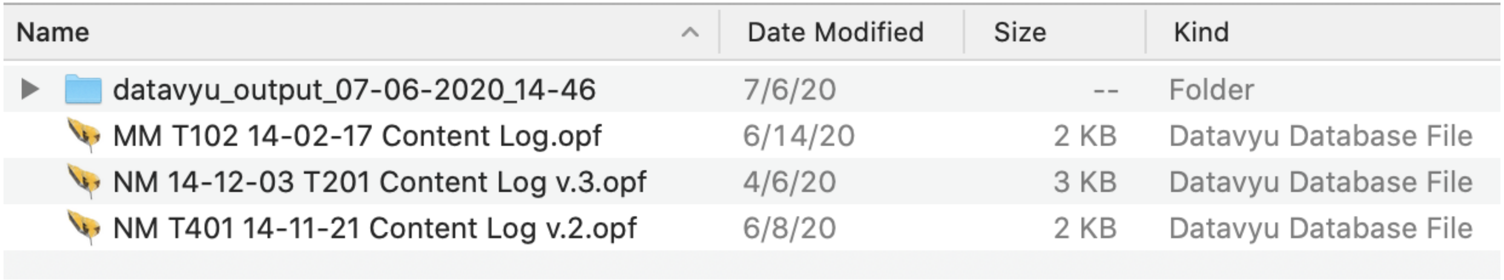
The *Media Player*.

The *Controller keypad* maps directly to your own, helping to record observations.

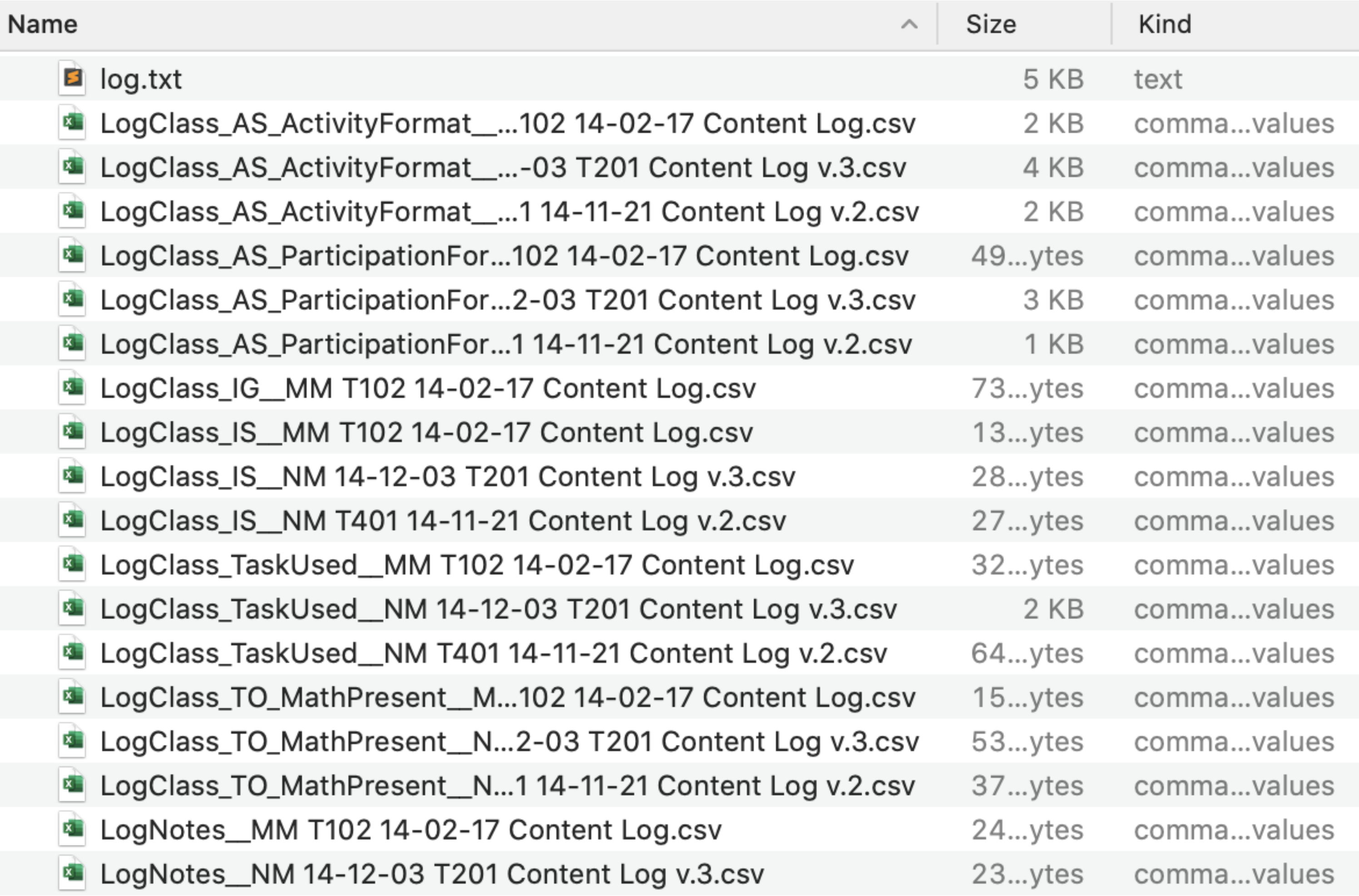
The *Controller* enables you to control playback, and to create *cells*.

A short (2 min.) video: https://datavyu.org/user-guide/guide.html

A look at exported datavyu data



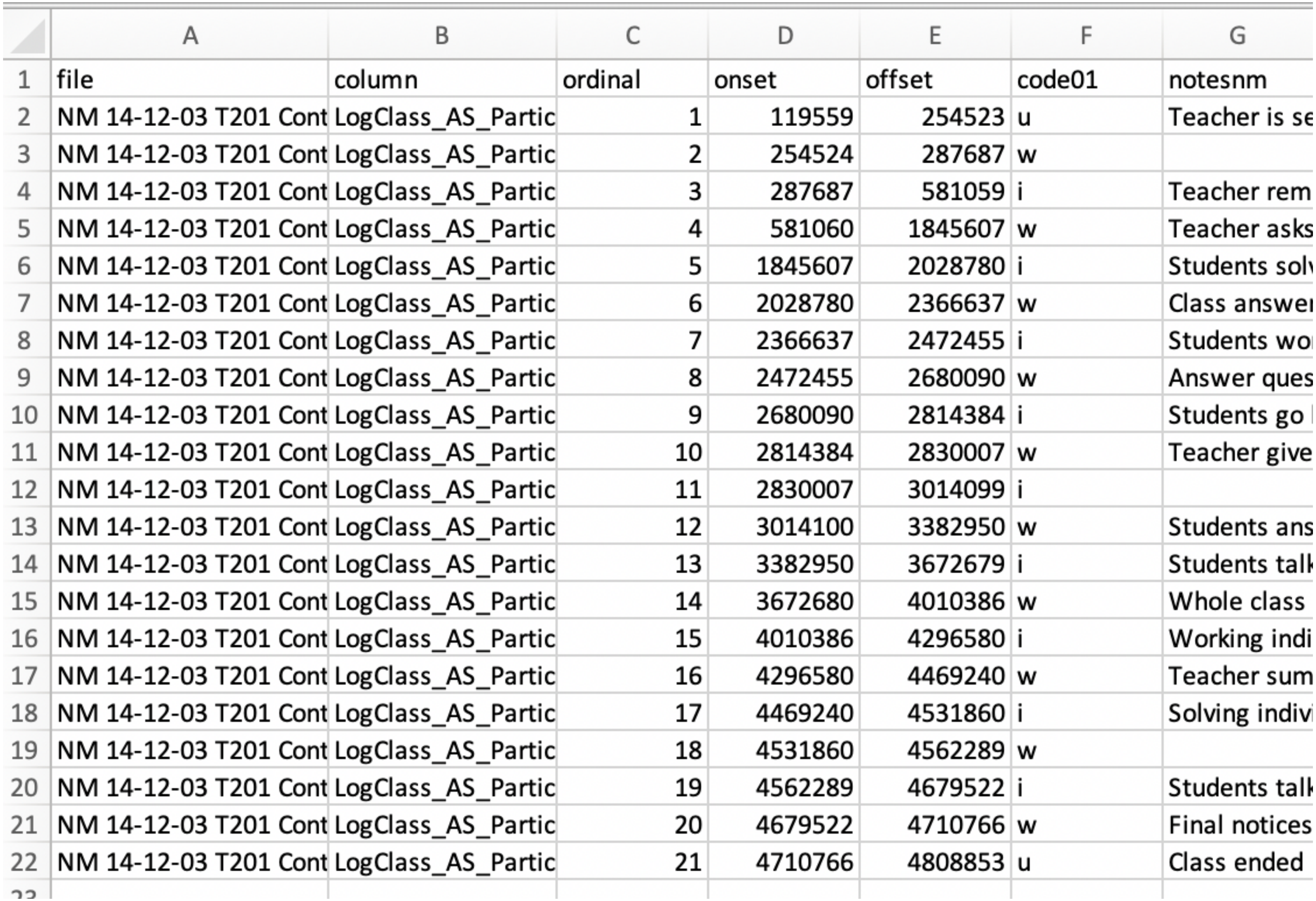
Name	Date Modified	Size	Kind
▶ datavyu_output_07-06-2020_14-46	7/6/20	--	Folder
MM T102 14-02-17 Content Log.opf	6/14/20	2 KB	Datavyu Database File
NM 14-12-03 T201 Content Log v.3.opf	4/6/20	3 KB	Datavyu Database File
NM T401 14-11-21 Content Log v.2.opf	6/8/20	2 KB	Datavyu Database File



Name	Size	Kind
log.txt	5 KB	text
LogClass_AS_ActivityFormat__...102 14-02-17 Content Log.csv	2 KB	comma...values
LogClass_AS_ActivityFormat__...-03 T201 Content Log v.3.csv	4 KB	comma...values
LogClass_AS_ActivityFormat__...1 14-11-21 Content Log v.2.csv	2 KB	comma...values
LogClass_AS_ParticipationFor...102 14-02-17 Content Log.csv	49...ytes	comma...values
LogClass_AS_ParticipationFor...2-03 T201 Content Log v.3.csv	3 KB	comma...values
LogClass_AS_ParticipationFor...1 14-11-21 Content Log v.2.csv	1 KB	comma...values
LogClass_IG__MM T102 14-02-17 Content Log.csv	73...ytes	comma...values
LogClass_IS__MM T102 14-02-17 Content Log.csv	13...ytes	comma...values
LogClass_IS__NM 14-12-03 T201 Content Log v.3.csv	28...ytes	comma...values
LogClass_IS__NM T401 14-11-21 Content Log v.2.csv	27...ytes	comma...values
LogClass_TaskUsed__MM T102 14-02-17 Content Log.csv	32...ytes	comma...values
LogClass_TaskUsed__NM 14-12-03 T201 Content Log v.3.csv	2 KB	comma...values
LogClass_TaskUsed__NM T401 14-11-21 Content Log v.2.csv	64...ytes	comma...values
LogClass_TO_MathPresent__M...102 14-02-17 Content Log.csv	15...ytes	comma...values
LogClass_TO_MathPresent__N...2-03 T201 Content Log v.3.csv	53...ytes	comma...values
LogClass_TO_MathPresent__N...1 14-11-21 Content Log v.2.csv	37...ytes	comma...values
LogNotes__MM T102 14-02-17 Content Log.csv	24...ytes	comma...values
LogNotes__NM 14-12-03 T201 Content Log v.3.csv	23...ytes	comma...values

See more in this vignette on how to make this data:
https://github.com/tca2/datavyu/blob/master/vignettes/preparing-data.Rmd

How can it be easier to use datavyu output?



	A	B	C	D	E	F	G
1	file	column	ordinal	onset	offset	code01	notesnm
2	NM 14-12-03 T201 Cont	LogClass_AS_Partic	1	119559	254523	u	Teacher is se
3	NM 14-12-03 T201 Cont	LogClass_AS_Partic	2	254524	287687	w	
4	NM 14-12-03 T201 Cont	LogClass_AS_Partic	3	287687	581059	i	Teacher rem
5	NM 14-12-03 T201 Cont	LogClass_AS_Partic	4	581060	1845607	w	Teacher asks
6	NM 14-12-03 T201 Cont	LogClass_AS_Partic	5	1845607	2028780	i	Students sol
7	NM 14-12-03 T201 Cont	LogClass_AS_Partic	6	2028780	2366637	w	Class answe
8	NM 14-12-03 T201 Cont	LogClass_AS_Partic	7	2366637	2472455	i	Students wo
9	NM 14-12-03 T201 Cont	LogClass_AS_Partic	8	2472455	2680090	w	Answer ques
10	NM 14-12-03 T201 Cont	LogClass_AS_Partic	9	2680090	2814384	i	Students go
11	NM 14-12-03 T201 Cont	LogClass_AS_Partic	10	2814384	2830007	w	Teacher give
12	NM 14-12-03 T201 Cont	LogClass_AS_Partic	11	2830007	3014099	i	
13	NM 14-12-03 T201 Cont	LogClass_AS_Partic	12	3014100	3382950	w	Students ans
14	NM 14-12-03 T201 Cont	LogClass_AS_Partic	13	3382950	3672679	i	Students talk
15	NM 14-12-03 T201 Cont	LogClass_AS_Partic	14	3672680	4010386	w	Whole class
16	NM 14-12-03 T201 Cont	LogClass_AS_Partic	15	4010386	4296580	i	Working indi
17	NM 14-12-03 T201 Cont	LogClass_AS_Partic	16	4296580	4469240	w	Teacher sum
18	NM 14-12-03 T201 Cont	LogClass_AS_Partic	17	4469240	4531860	i	Solving indiv
19	NM 14-12-03 T201 Cont	LogClass_AS_Partic	18	4531860	4562289	w	
20	NM 14-12-03 T201 Cont	LogClass_AS_Partic	19	4562289	4679522	i	Students talk
21	NM 14-12-03 T201 Cont	LogClass_AS_Partic	20	4679522	4710766	w	Final notices
22	NM 14-12-03 T201 Cont	LogClass_AS_Partic	21	4710766	4808853	u	Class ended

Exploring the columns and files in the data

First, let's load the package.

```r
devtools::install_github("tca2/datavyu") # only have to do once
```

```r
library(datavyu)
```

Using {datavyu}, you can find the unique columns across all of the files in a directory.

```r
find_unique_columns("ex-data/datavyu_output_07-06-2020_14-46")
```

columns
LogClass_AS_ActivityFormat
LogClass_AS_ParticipationFormat
LogClass_IG
LogClass_TO_MathPresent
LogClass_IS
LogNotes
LogClass_TaskUsed

You can also find unique files

```
find_unique_files("ex-data/datavyu_output_07-06-2020_14-46")
```

files
MM T102 14-02-17 Content Log
NM 14-12-03 T201 Content Log v.3
NM T401 14-11-21 Content Log v.2

2. Big task #1: Summarizing a column

Summarizing a column

{datavyu} can help to summarize a column. It defaults to summarizing the frequency of codes for a specified column.

```
summarize_column(column = "LogClass_AS_ActivityFormat",
                 directory = "ex-data/datavyu_output_07-06-2020_14-4
## # A tibble: 8 x 3
##   log_class_as_activity_format_code01     n percent
## * <chr>                               <dbl>   <dbl>
## 1 l                                       7  0.318
## 2 sp                                      7  0.318
## 3 a                                       2  0.0909
## 4 o                                       2  0.0909
## 5 aw                                      1  0.0455
## 6 class discussion?                       1  0.0455
## 7 class discussion? lecture?              1  0.0455
## 8 l??                                     1  0.0455
```

Setting an option

We'll be typing that folder file path a number of times.

You can set an option that will mean that the folder file path you set will be used *by default*, though you can over-ride it any time you like.

```
options(directory = "ex-data/datavyu_output_07-06-2020_14-46")
```

We can also explore the frequencies *by file* by changing the `by_file` argument to `TRUE`.

```
summarize_column(column = "LogClass_AS_ActivityFormat",
                 by_file = TRUE) %>%
  dplyr::select(-file)
## # A tibble: 13 x 3
##    log_class_as_activity_format_code01     n percent
##    <chr>                               <dbl>   <dbl>
##  1 aw                                      1   0.1  
##  2 l                                       3   0.3  
##  3 sp                                      6   0.6  
##  4 a                                       1   0.333
##  5 l                                       1   0.333
##  6 o                                       1   0.333
##  7 a                                       1   0.111
##  8 class discussion?                       1   0.111
##  9 class discussion? lecture?              1   0.111
## 10 l                                       3   0.333
## 11 l??                                     1   0.111
## 12 o                                       1   0.111
## 13 sp                                      1   0.111
```

To summarize durations (instead of frequencies) by changing the `summary` argument, which defaults to `"frequency"`, but can be changed to `"duration"`:

```
summarize_column(column = "LogClass_AS_ActivityFormat",
                 summary = "duration")
## # A tibble: 8 x 3
##   log_class_as_activity_format_code01 duration     percent
## * <chr>                               <chr>          <dbl>
## 1 l                                   00:52:00:316  0.327
## 2 a                                   00:27:16:305  0.172
## 3 sp                                  00:25:18:250  0.159
## 4 class discussion?                   00:20:39:356  0.130
## 5 o                                   00:13:01:093  0.0820
## 6 aw                                  00:10:08:256  0.0638
## 7 l??                                 00:06:06:588  0.0385
## 8 class discussion? lecture?          00:04:20:950  0.0274
```

Columns of durations can also be summarized by file:

```
summarize_column(column = "LogClass_AS_ActivityFormat",
                 summary = "duration",
                 by_file = TRUE) %>%
    dplyr::select(-file)
## # A tibble: 13 x 3
##    log_class_as_activity_format_code01 duration     percent
##    <chr>                               <chr>          <dbl>
##  1 l                                   00:46:17:990  0.576
##  2 sp                                  00:23:59:473  0.298
##  3 aw                                  00:10:08:256  0.126
##  4 a                                   00:04:53:373  0.898
##  5 o                                   00:00:25:134  0.0770
##  6 l                                   00:00:08:029  0.0246
##  7 a                                   00:22:22:932  0.307
##  8 class discussion?                   00:20:39:356  0.283
##  9 o                                   00:12:35:959  0.173
## 10 l??                                 00:06:06:588  0.0837
## 11 l                                   00:05:34:297  0.0763
## 12 class discussion? lecture?          00:04:20:950  0.0596
## 13 sp                                  00:01:18:777  0.0180
```

3. Big task #2: Plotting a column summary

Plotting the results of a summary of a column

{datavyu} can also help to plot the summary of a column:

```r
freq_summary <- summarize_column(column = "LogClass_AS_ActivityFormat

plot_column_summary(freq_summary)
```

This also works by file-so long as the column is summarized by file:

```r
freq_summary <-
  summarize_column(column = "LogClass_AS_ActivityFormat",
                   summary = "duration",
                   by_file = TRUE)

plot_column_summary(freq_summary)
```

Similarly, if the output is for the duration, rather than the frequency, the durations are plotted:

```
duration_summary <-
  summarize_column(column = "LogClass_AS_ActivityFormat",
                   summary = "duration")

plot_column_summary(duration_summary)
```

Like for frequency, these can be plotted by file:

```r
duration_summary_by_file <-
  summarize_column(column = "LogClass_AS_ActivityFormat",
                   summary = "duration",
                   by_file = TRUE)

plot_column_summary(duration_summary_by_file)
```

Output can be passed between functions with the pipe operator:

```r
library(dplyr)

summarize_column(column = "LogClass_AS_ActivityFormat",
                 summary = "duration",
                 by_file = TRUE) %>%
  plot_column_summary()
```

4. Big task #3: Preparing and plotting time series data

Time series preparation and plot

```
prepared_time_series <-
  prep_time_series(column = "LogClass_AS_ActivityFormat",
                   specified_file = "MM T102 14-02-17 Content Log")

prepared_time_series
## # A tibble: 4,849 x 2
##        ts code
##     * <dbl> <chr>
##   1   235 aw
##   2   236 aw
##   3   237 aw
##   4   238 aw
##   5   239 aw
##   6   240 aw
##   7   241 aw
##   8   242 aw
##   9   243 aw
## 10   244 aw
## # … with 4,839 more rows
```

The units argument defaults to "s", but can be changed to "m" (to round the data to minutes) or "ms" (to not round the data and to retain the units as milliseconds).

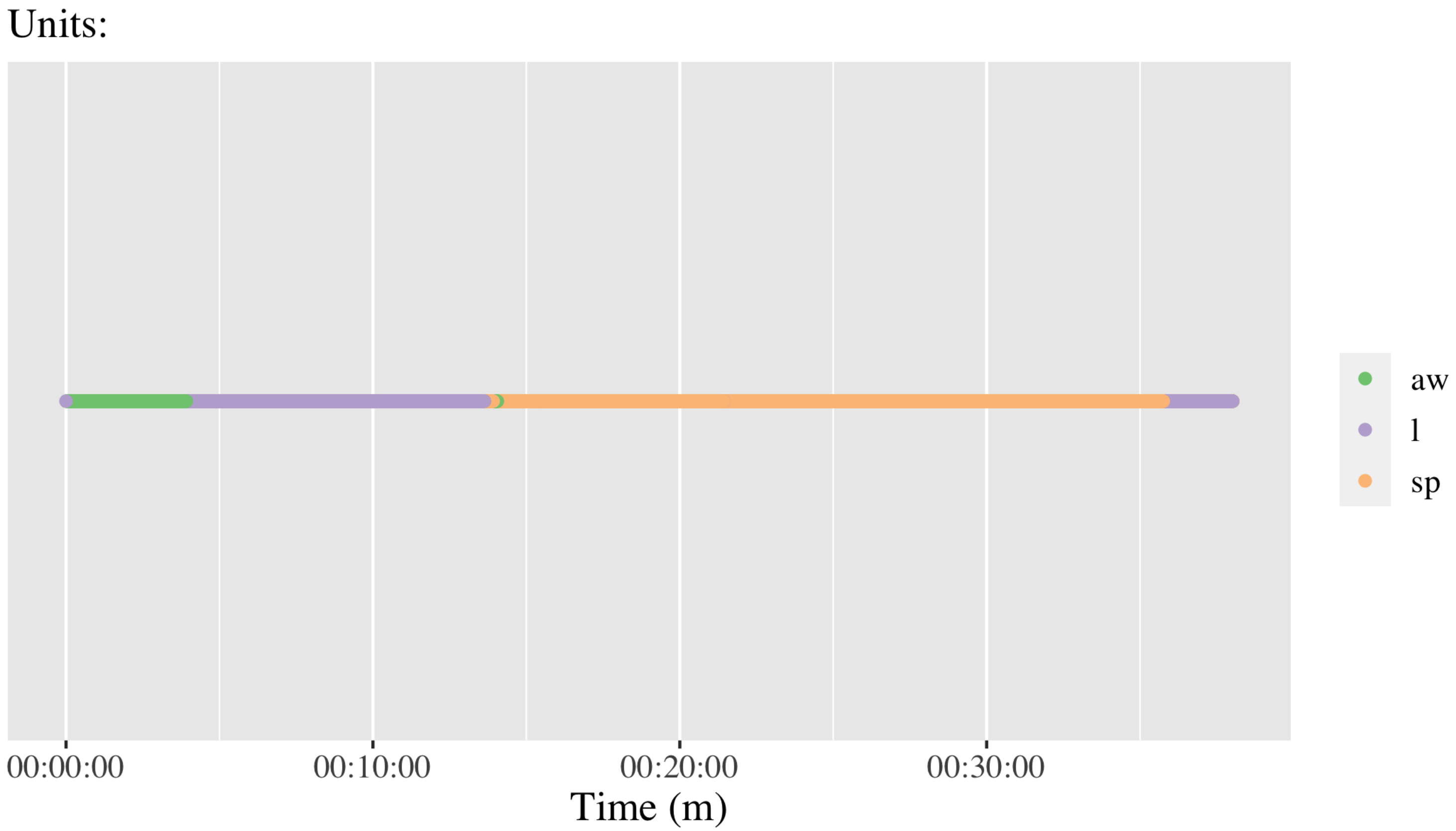
We can see how using milliseconds increases the number of data points:

```
prepared_time_series_ms <-
  prep_time_series(column = "LogClass_AS_ActivityFormat",
                   specified_file = "MM T102 14-02-17 Content Log",
                   units = "ms") # takes around .8s to run

prepared_time_series_ms
## # A tibble: 4,825,743 x 2
##         ts code
##   *  <int> <chr>
##   1 235026 aw
##   2 235027 aw
##   3 235028 aw
##   4 235029 aw
##   5 235030 aw
##   6 235031 aw
##   7 235032 aw
##   8 235033 aw
##   9 235034 aw
## 10 235035 aw
## # … with 4,825,733 more rows
```

This time series data can then be plotted (using the data with the units as seconds):

```
plot_time_series(prepared_time_series)
```



5. Next steps and discussion

Next steps

- Improving time series preparation to work by file
- Improving time series plotting
- Addressing many issues: https://github.com/tca2/datavyu/issues
- Currying along other variables (e.g., teacher ID)
- Reliability plots and statistics
- Improving plot theming
- Documenting and testing the package
- Preparing the package for CRAN submission
- Working with the creator of {datavyur} so that both packages can be on CRAN

We welcome your feedback and advice

https://github.com/tca2/datavyu

This presentation was created with {xaringan}