iMentor Message Traffic Analysis

Nick Chen

December 14, 2016

I. Introduction and Motivation

The purpose of this analysis is to examine patterns observed in the platform behavior. datasets provided to us by iMentor with the goal of determining first whether there are any distinguishing patterns that could be used to predict dropout and second whether mentors who drop out of the program behave in an appreciably different manner than mentors who do not drop out.

II. Summary of Results

We find that dropout mentors do not seem to behave in an appreciably different manner as observed from platform behavior datasets provided by iMentor. Across a variety of metrics including mentor response time, mentor word count, and mentor ignore rates, we see a definitive time trend indicating decreasing mentor engagement over time.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(xlsx)
```

```
## Loading required package: rJava

## Loading required package: xlsxjars
```

```
library(openxlsx)
```

```
##
## Attaching package: 'openxlsx'

## The following objects are masked from 'package:xlsx':
##
##     createWorkbook, loadWorkbook, read.xlsx, saveWorkbook,
##     write.xlsx
```

```
# ********************************************** Load data
# **********************************************

library(RPostgreSQL)
```

```
## Loading required package: DBI
```

```
# Define driver
pg <- dbDriver("PostgreSQL")

# Define Connection
conn <- dbConnect(pg, user = "awesome_admin", password = "w205.Awesome", host = "34.193.7.196",
    port = 5432, dbname = "awesome")

# ***************** Match history
match.history <- dbGetQuery(conn, "select * from match_history")


# ***************** Match pairs
match.pairs <- dbGetQuery(conn, "select * from master_table")

# ***************** Message Traffic
message.traffic.mentors <- dbGetQuery(conn, "select * from mentor_behavior_2")
```

```r
message.traffic.mentees <- dbGetQuery(conn, "select * from mentee_behavior_2")

# ********************************* Prep data
# *********************************

# Create combined dropout flag in match history dataframe
match.history <- match.history %>% mutate_if(is.factor, as.character)
match.history$combined.dropout.flag <- ifelse(match.history$match_closure_reason_control ==
    "Match Open", "Match Open", match.history$match_closure_reason_super)

# ***************** Message traffic is from 2015 - 2016, so remove mentors
# that were not flagged as active at the bginning of the 15/16 school year.
# Active pairs are flagged as 1 in the mentor active flag from the match
# pair dataset, so remove mentors/mentees flagged as 2

# Merge in active flag from match pairs
message.traffic.mentors <- merge(message.traffic.mentors, match.pairs[, c("most_recent_mentor_persona_id",
    "mentor_active_boy_1516")], by.x = "mentor_persona_id", by.y = "most_recent_mentor_persona_id")
message.traffic.mentees <- merge(message.traffic.mentees, match.pairs[, c("mentee_user_id",
    "mentee_active_boy_1516")], by.x = "mentee_user_id", by.y = "mentee_user_id")

# remove observations from message traffic for mentors / mentees that were
# not flagged as active as of the beginning of the 15/16 school year
message.traffic.mentors <- subset(message.traffic.mentors, message.traffic.mentors$mentor_active_boy_1516 ==
    1)
message.traffic.mentees <- subset(message.traffic.mentees, message.traffic.mentees$mentee_active_boy_1516 ==
    1)

# merge in combined dropout flag and effectively limit to mentors and
# mentees in the match history dataset
match.closure.summary <- match.history %>% group_by(combined.dropout.flag) %>%
    summarise(count = n())

message.traffic.mentors <- merge(message.traffic.mentors, match.history[, c("pair_id",
    "combined.dropout.flag", "match_end_date")], by.x = "pair_id", by.y = "pair_id")
message.traffic.mentees <- merge(message.traffic.mentees, match.history[, c("pair_id",
    "combined.dropout.flag")], by.x = "pair_id", by.y = "pair_id")
```

```r
# ******************** convert end date
message.traffic.mentors$match_end_date_converted <- as.POSIXct(strptime(message.traffic.mentors$match_end_date,
    "%Y-%m-%d", tz = ""), tz = "GMT")

# total writing time
message.traffic.mentors$total.writing.time <- as.numeric(difftime(message.traffic.mentors$user_first_sub,
    message.traffic.mentors$user_begin, units = c("mins")))
message.traffic.mentees$total.writing.time <- as.numeric(difftime(message.traffic.mentees$user_first_sub,
    message.traffic.mentees$user_begin, units = c("mins")))

# calculate curriculum sequence
message.traffic.mentors <- message.traffic.mentors %>% group_by(mentor_persona_id) %>%
    mutate(lesson.rank = dense_rank(curriculum_sequence))
message.traffic.mentees <- message.traffic.mentees %>% group_by(mentee_persona_id) %>%
    mutate(lesson.rank = dense_rank(curriculum_sequence))

message.traffic.mentors <- message.traffic.mentors %>% group_by(mentor_persona_id) %>%
    mutate(max.lesson.rank = max(lesson.rank))
message.traffic.mentees <- message.traffic.mentees %>% group_by(mentee_persona_id) %>%
    mutate(max.lesson.rank = max(lesson.rank))

message.traffic.mentors$lessons.bf.last <- message.traffic.mentors$max.lesson.rank -
    message.traffic.mentors$lesson.rank
message.traffic.mentees$lessons.bf.last <- message.traffic.mentees$max.lesson.rank -
    message.traffic.mentees$lesson.rank

# calculate wordcount zscore
message.traffic.mentors$wordcount.w.zeroes <- ifelse(!is.na(message.traffic.mentors$canvas_word_cnt),
    message.traffic.mentors$canvas_word_cnt, 0)
message.traffic.mentees$wordcount.w.zeroes <- ifelse(!is.na(message.traffic.mentees$canvas_word_cnt),
    message.traffic.mentees$canvas_word_cnt, 0)

message.traffic.mentors <- message.traffic.mentors %>% group_by(mentor_persona_id) %>%
    mutate(mentor.mean.wordcount = mean(canvas_word_cnt, na.rm = TRUE), mentor.sd.wordcount = sd(canvas_word_cnt,
        na.rm = TRUE))
message.traffic.mentors$mentor.zscore.wordcount <- (message.traffic.mentors$canvas_word_cnt -
    message.traffic.mentors$mentor.mean.wordcount)/message.traffic.mentors$mentor.sd.wordcount
```

```r
message.traffic.mentors <- message.traffic.mentors %>% group_by(mentor_persona_id) %>%
    mutate(mentor.mean.wordcount.w.zeroes = mean(wordcount.w.zeroes, na.rm = TRUE),
        mentor.sd.wordcount.w.zeroes = sd(wordcount.w.zeroes, na.rm = TRUE))
message.traffic.mentors$mentor.zscore.wordcount.w.zeroes <- (message.traffic.mentors$wordcount.w.zeroes -
    message.traffic.mentors$mentor.mean.wordcount.w.zeroes)/message.traffic.mentors$mentor.sd.wordcount.w.zeroes

message.traffic.mentees <- message.traffic.mentees %>% group_by(mentee_persona_id) %>%
    mutate(mentee.mean.wordcount = mean(canvas_word_cnt, na.rm = TRUE), mentee.sd.wordcount = sd(canvas_word_cnt,
        na.rm = TRUE))
message.traffic.mentees$mentee.zscore.wordcount <- (message.traffic.mentees$canvas_word_cnt -
    message.traffic.mentees$mentee.mean.wordcount)/message.traffic.mentees$mentee.sd.wordcount

message.traffic.mentees <- message.traffic.mentees %>% group_by(mentee_persona_id) %>%
    mutate(mentee.mean.wordcount.w.zeroes = mean(wordcount.w.zeroes, na.rm = TRUE),
        mentee.sd.wordcount.w.zeroes = sd(wordcount.w.zeroes, na.rm = TRUE))
message.traffic.mentees$mentee.zscore.wordcount.w.zeroes <- (message.traffic.mentees$wordcount.w.zeroes -
    message.traffic.mentees$mentee.mean.wordcount.w.zeroes)/message.traffic.mentees$mentee.sd.wordcount.w.zeroes


# calculate writing time zscore
message.traffic.mentors <- message.traffic.mentors %>% group_by(mentor_persona_id) %>%
    mutate(mentor.mean.writing.time = mean(total.writing.time, na.rm = TRUE),
        mentor.sd.writing.time = sd(total.writing.time, na.rm = TRUE))
message.traffic.mentors$mentor.zscore.writing.time <- (message.traffic.mentors$total.writing.time -
    message.traffic.mentors$mentor.mean.writing.time)/message.traffic.mentors$mentor.sd.writing.time

message.traffic.mentees <- message.traffic.mentees %>% group_by(mentee_persona_id) %>%
    mutate(mentee.mean.writing.time = mean(total.writing.time, na.rm = TRUE),
        mentee.sd.writing.time = sd(total.writing.time, na.rm = TRUE))
message.traffic.mentees$mentee.zscore.writing.time <- (message.traffic.mentees$total.writing.time -
    message.traffic.mentees$mentee.mean.writing.time)/message.traffic.mentees$mentee.sd.writing.time

# add flags for one and four weeks before dropout
message.traffic.mentors$message.within.1.week.dropout <- ifelse(as.numeric(difftime(message.traffic.mentors$match_end_date,
    message.traffic.mentors$user_first_sub, units = c("hours"))) < 168, "Within 1 Week of Dropout",
    "Other Messages")
message.traffic.mentors$message.within.4.weeks.dropout <- ifelse(as.numeric(difftime(message.traffic.mentors$match_end_date,
    message.traffic.mentors$user_first_sub, units = c("hours"))) < 672, "Within 4 Weeks of Dropout",
```

```
    "Other Messages")

# add flags for last message and last four messages
message.traffic.mentors$last.message.flag <- ifelse(message.traffic.mentors$max.lesson.rank ==
    message.traffic.mentors$lesson.rank, "Last Message", "Other Messages")
message.traffic.mentors$last.4.messages.flag <- ifelse(message.traffic.mentors$max.lesson.rank -
    message.traffic.mentors$lesson.rank <= 4, "Last 4 Messages", "Other Messages")

# add flag for blank messages
message.traffic.mentors$blank.message.flag <- ifelse(is.na(message.traffic.mentors$canvas_word_cnt),
    1, 0)
message.traffic.mentees$blank.message.flag <- ifelse(is.na(message.traffic.mentees$canvas_word_cnt),
    1, 0)

# remove combined dropouts that we are not interested in
message.traffic.mentors <- subset(message.traffic.mentors, message.traffic.mentors$combined.dropout.flag !=
    "Program Partnership ended" & message.traffic.mentors$combined.dropout.flag !=
    "")
message.traffic.mentees <- subset(message.traffic.mentees, message.traffic.mentees$combined.dropout.flag !=
    "Program Partnership ended" & message.traffic.mentees$combined.dropout.flag !=
    "")

# Limit to messages that have some content.
message.traffic.mentors.content <- subset(message.traffic.mentors, !is.na(message.traffic.mentors$canvas_word_cnt) &
    message.traffic.mentors$canvas_word_cnt > 0 & message.traffic.mentors$total.writing.time >
    0)
message.traffic.mentees.content <- subset(message.traffic.mentees, !is.na(message.traffic.mentees$canvas_word_cnt) &
    message.traffic.mentees$canvas_word_cnt > 0 & message.traffic.mentees$total.writing.time >
    0)

# drop intermediate columns
message.traffic.mentors <- message.traffic.mentors[, !names(message.traffic.mentors) %in%
    c("mentor.mean.wordcount", "mentor.sd.wordcount", "mentor.mean.writing.time",
        "mentor.sd.writing.time")]
message.traffic.mentees <- message.traffic.mentees[, !names(message.traffic.mentees) %in%
    c("mentee.mean.wordcount", "mentee.sd.wordcount", "mentee.mean.writing.time",
        "mentee.sd.writing.time")]
```

```
message.traffic.mentors.content <- message.traffic.mentors.content[, !names(message.traffic.mentors.content) %in%
    c("max.lesson.rank", "mentor.mean.wordcount", "mentor.sd.wordcount", "mentor.mean.writing.time",
        "mentor.sd.writing.time")]
message.traffic.mentees.content <- message.traffic.mentees.content[, !names(message.traffic.mentees.content) %in%
    c("max.lesson.rank", "mentee.mean.wordcount", "mentee.sd.wordcount", "mentee.mean.writing.time",
        "mentee.sd.writing.time")]
```

III. Mentor and Mentee Separate Message Traffic Analysis

a.Writing time

```
# Summary
summary(message.traffic.mentors.content$total.writing.time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       9      38    4384    5878  531500
```

```
summary(message.traffic.mentees.content$total.writing.time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       7     918    1008    2024    1120  515000
```

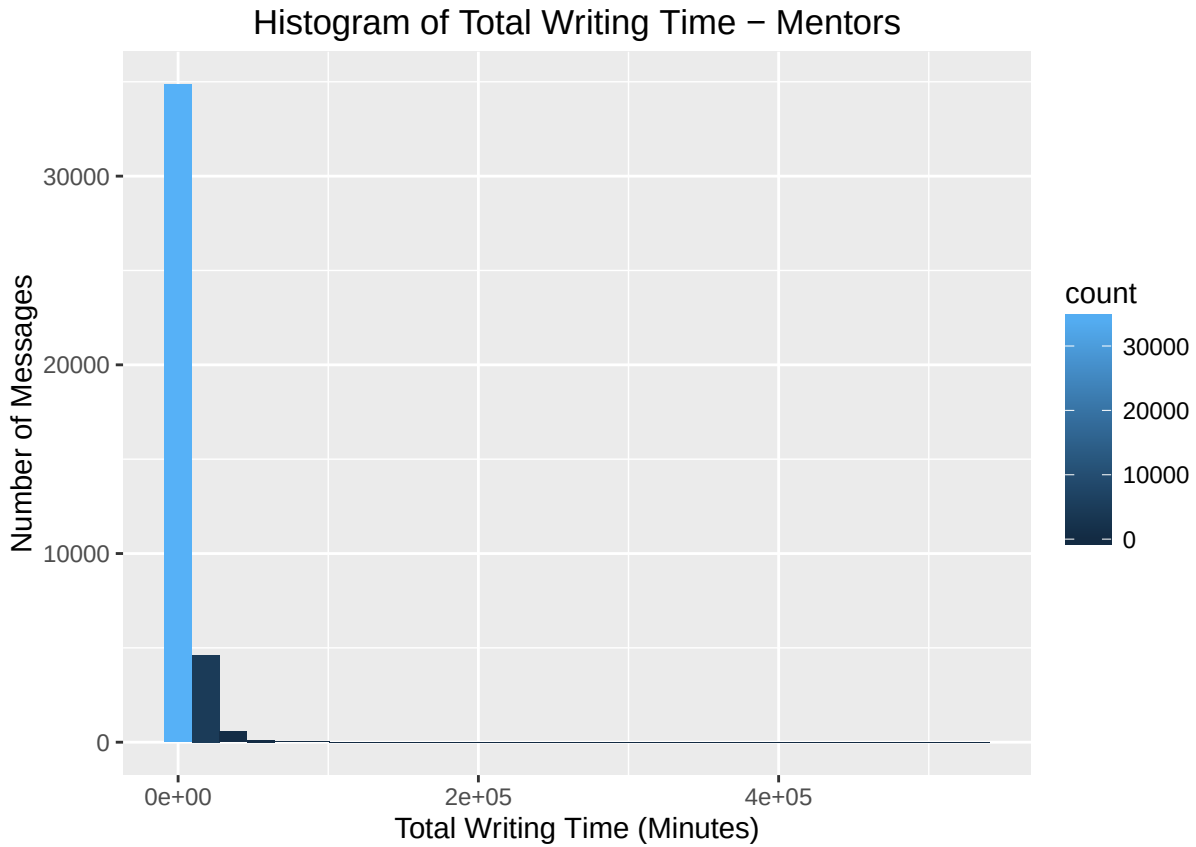
```
# compare writing time across dropout groups
writing.time.by.group.summ <- message.traffic.mentors %>% group_by(combined.dropout.flag) %>%
    summarise(mean.writing.time = mean(total.writing.time, na.rm = TRUE), count.obs = n())
tt = pairwise.t.test(message.traffic.mentors.content$total.writing.time, message.traffic.mentors.content$combined.dropout.flag,
    p.adjust.method = "bonferroni", na.rm = TRUE)
tt
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  message.traffic.mentors.content$total.writing.time and message.traffic.mentors.content$combined.dropout.flag
##
```

```
##                                 Formal closure Match Open
## Match Open                      1              -
## Mentee can no longer participate 1             1
## Mentor can no longer participate 1             1
##                                 Mentee can no longer participate
## Match Open                      -
## Mentee can no longer participate -
## Mentor can no longer participate 1
##
## P value adjustment method: bonferroni
```

```
# look at distribution of response time for messages that actually have
# content. The distribution shows a positive skew and an extreme
# concentration of short time spent writing.
writing.time.hist.mentors <- ggplot(data = message.traffic.mentors.content,
    aes(x = total.writing.time))
writing.time.hist.mentors + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Total Writing Time - Mentors") +
    labs(y = "Number of Messages", x = "Total Writing Time (Minutes)")
```

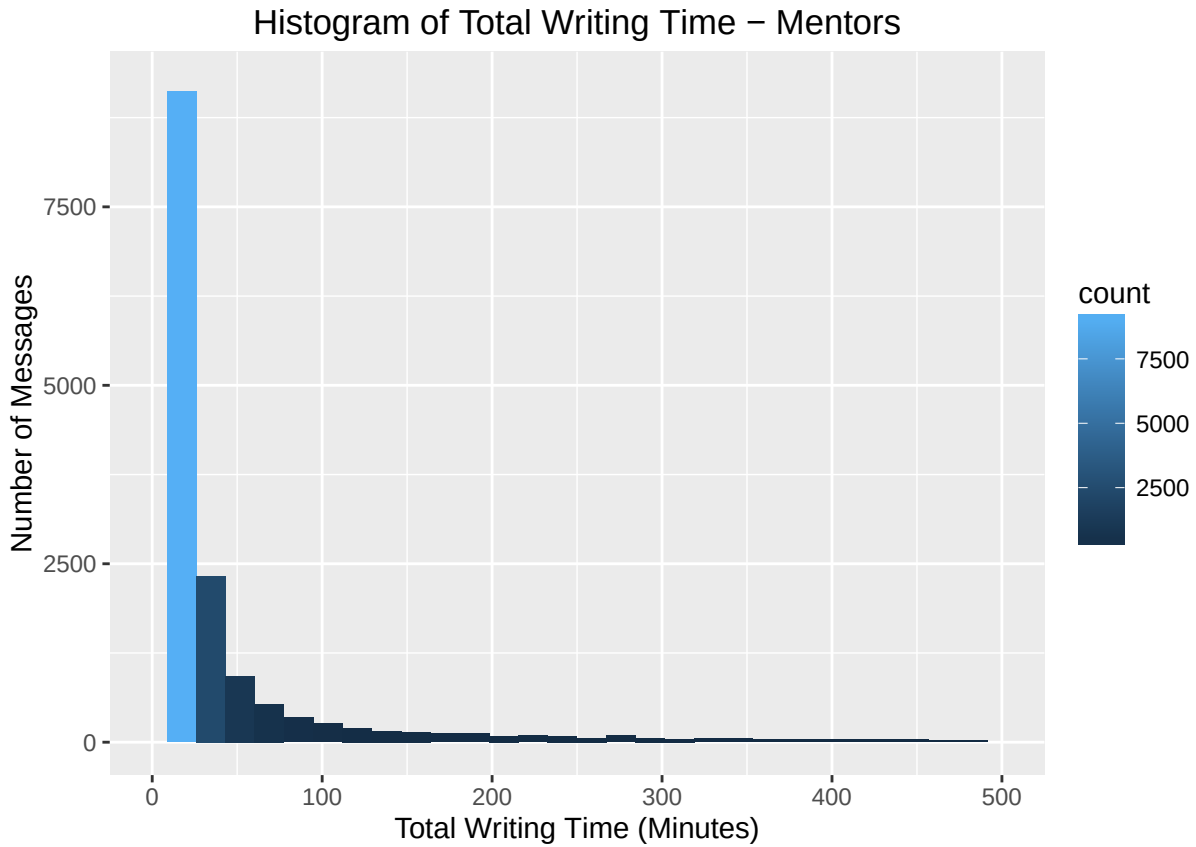
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
writing.time.hist.mentors2 <- ggplot(data = message.traffic.mentors.content,
    aes(x = total.writing.time))
writing.time.hist.mentors2 + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Total Writing Time - Mentors") +
    labs(y = "Number of Messages", x = "Total Writing Time (Minutes)") + xlim(c(0,
    500))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

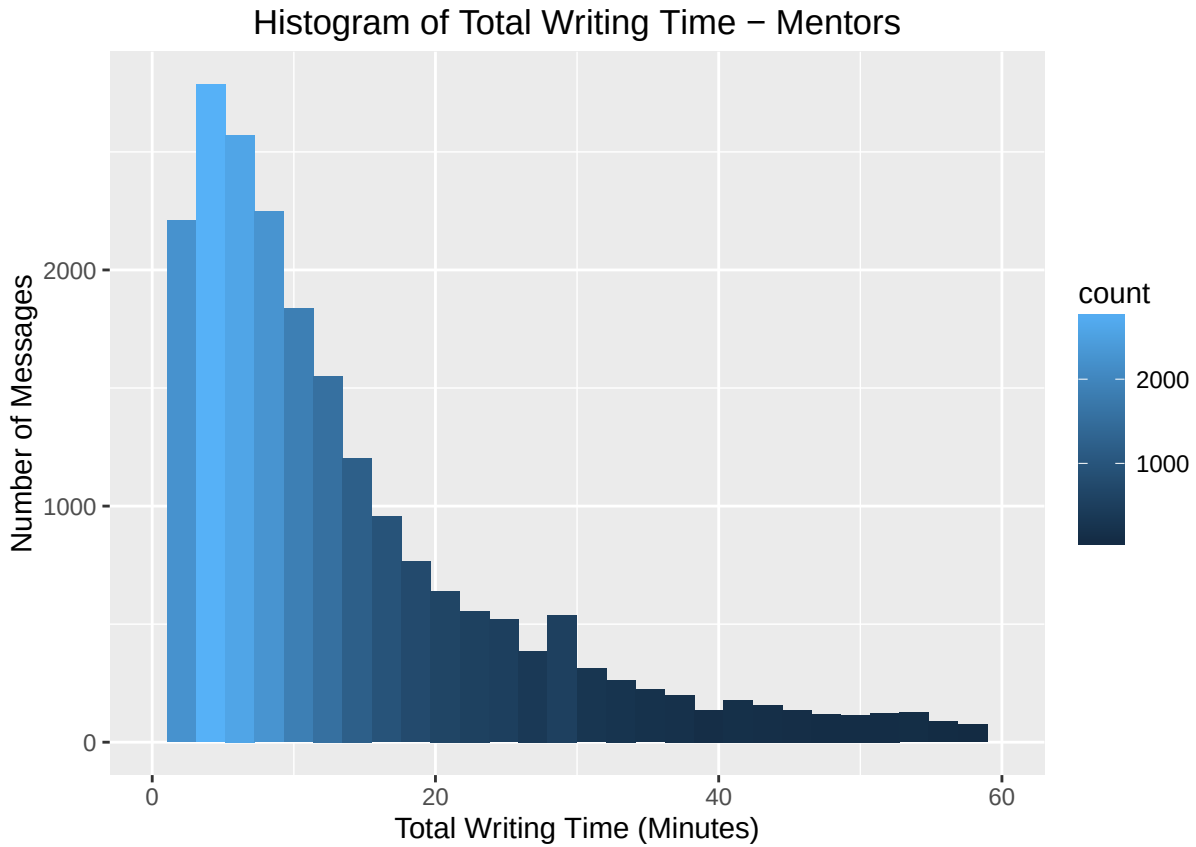
```
## Warning: Removed 15976 rows containing non-finite values (stat_bin).
```




```
writing.time.hist.mentors3 <- ggplot(data = message.traffic.mentors.content,
    aes(x = total.writing.time))
writing.time.hist.mentors3 + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Total Writing Time - Mentors") +
    labs(y = "Number of Messages", x = "Total Writing Time (Minutes)") + xlim(c(0,
    60))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

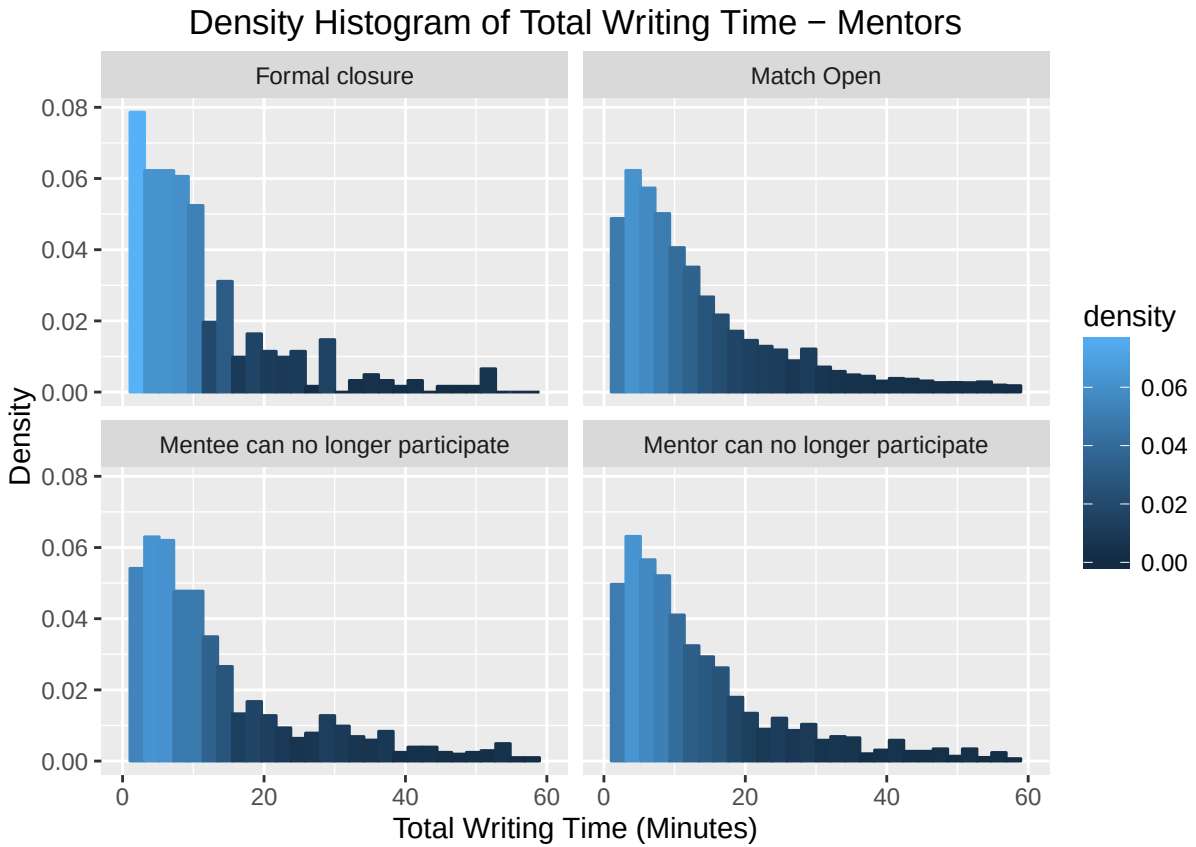
```
## Warning: Removed 18784 rows containing non-finite values (stat_bin).
```




```
# Look at histogram by dropout group - no clear distinction among dropout
# groups.
writing.time.hist.mentors.4 <- ggplot(data = message.traffic.mentors.content,
    aes(x = total.writing.time, color = ..density.., fill = ..density.., y = ..density..))
writing.time.hist.mentors.4 + geom_histogram() + ggtitle("Density Histogram of Total Writing Time - Mentors") +
    labs(y = "Density", x = "Total Writing Time (Minutes)") + xlim(c(0, 60)) +
    facet_wrap(~combined.dropout.flag)
```

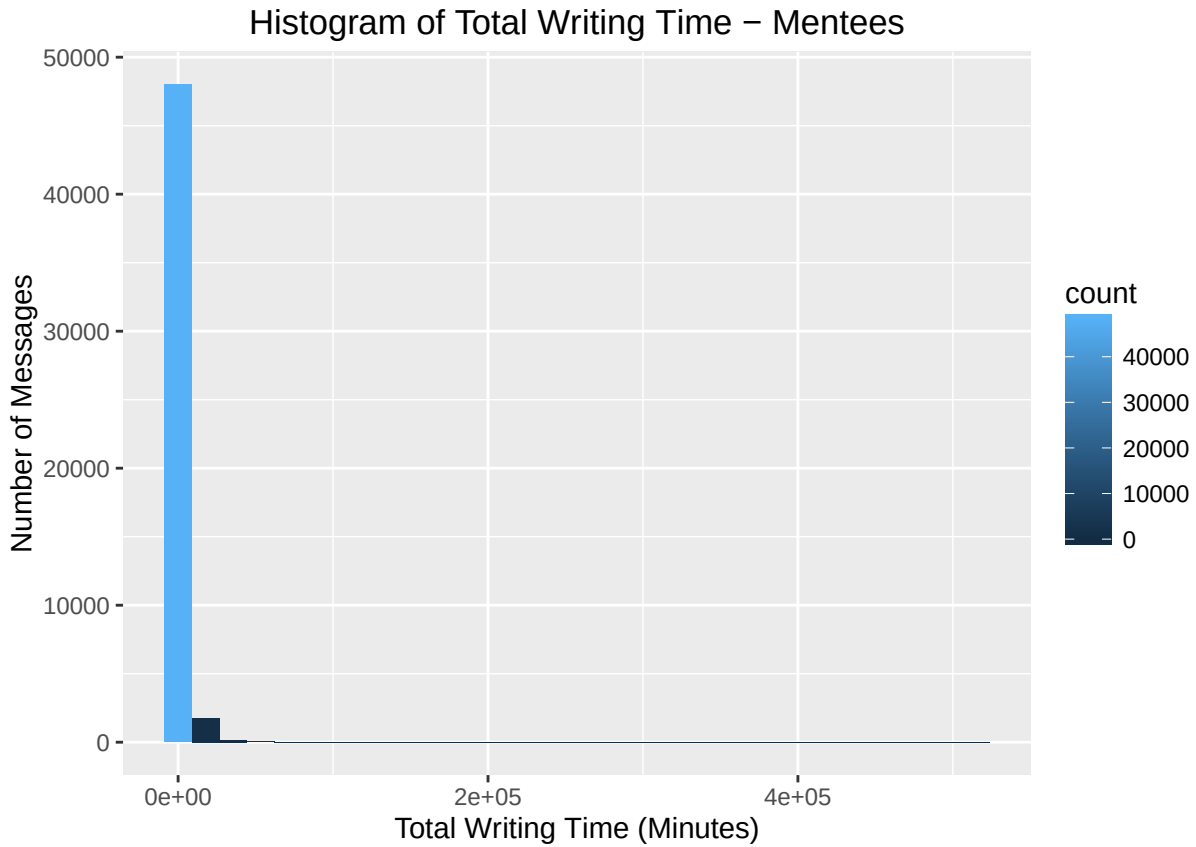
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 18784 rows containing non-finite values (stat_bin).
```




```
# The distribution for mentees also shows a positive skew and large
# concentration of relatively short time spent writing. The right hand tail
# for mentees does not seem to show as many individuals who took longer
# times to write their message.
writing.time.hist.mentees <- ggplot(data = message.traffic.mentees.content,
    aes(x = total.writing.time))
writing.time.hist.mentees + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Total Writing Time - Mentees") +
    labs(y = "Number of Messages", x = "Total Writing Time (Minutes)")
```

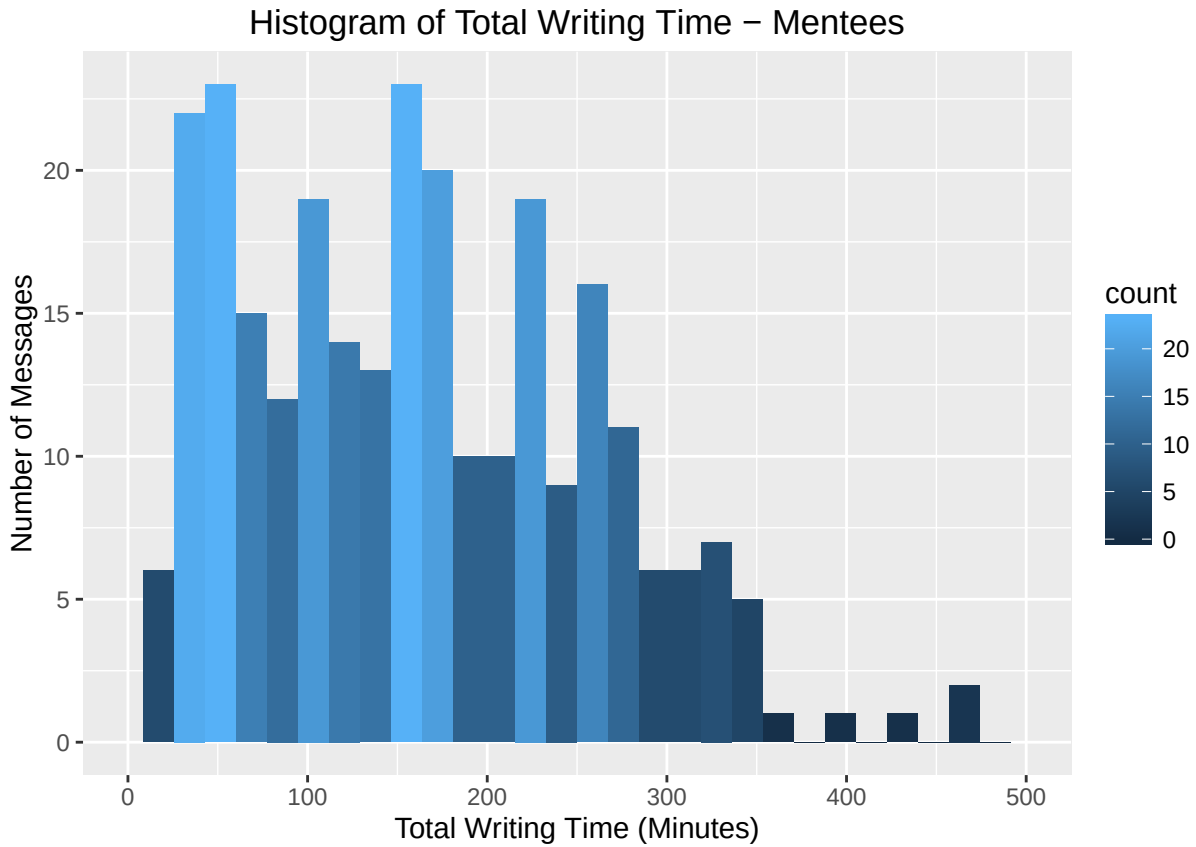
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
writing.time.hist.mentees.2 <- ggplot(data = message.traffic.mentees.content,
    aes(x = total.writing.time))
writing.time.hist.mentees.2 + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Total Writing Time - Mentees") +
    labs(y = "Number of Messages", x = "Total Writing Time (Minutes)") + xlim(c(0,
    500))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

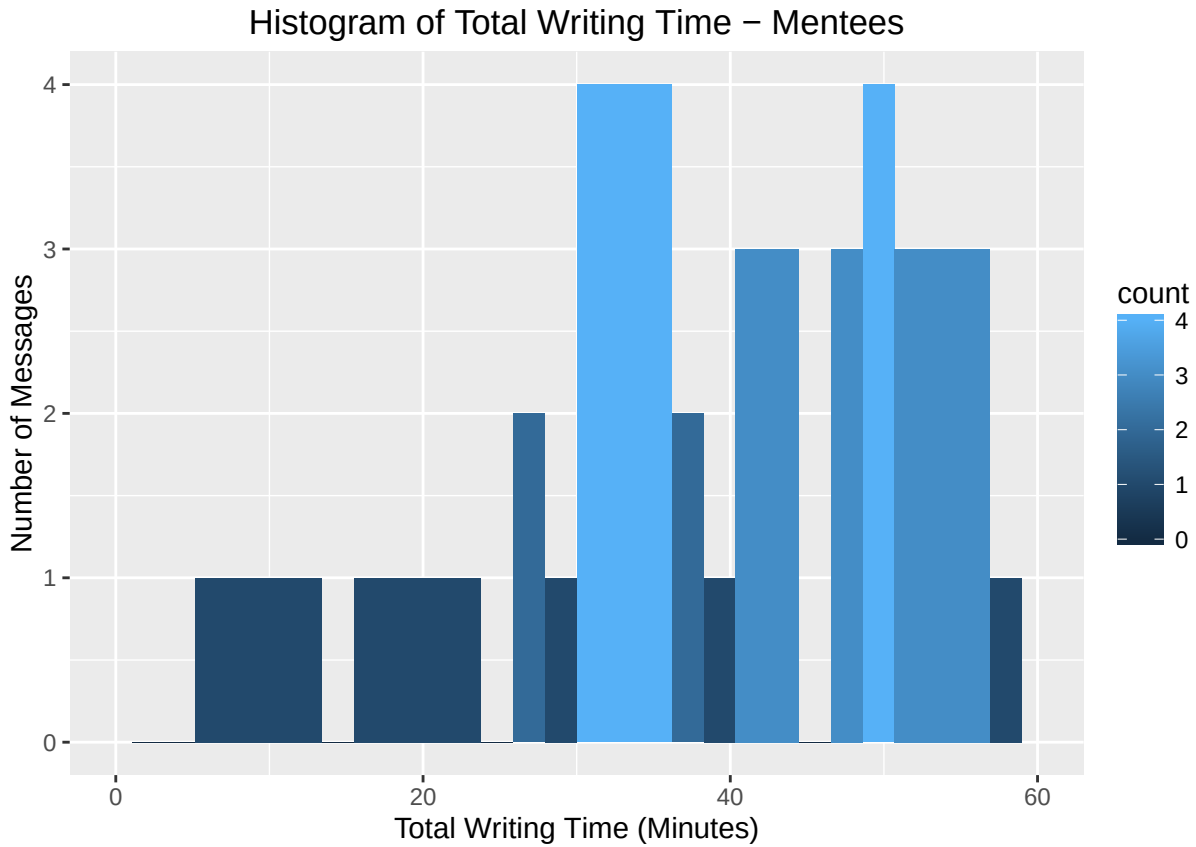
```
## Warning: Removed 49777 rows containing non-finite values (stat_bin).
```




```
writing.time.hist.mentees.3 <- ggplot(data = message.traffic.mentees.content,
    aes(x = total.writing.time))
writing.time.hist.mentees.3 + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Total Writing Time - Mentees") +
    labs(y = "Number of Messages", x = "Total Writing Time (Minutes)") + xlim(c(0,
    60))
```

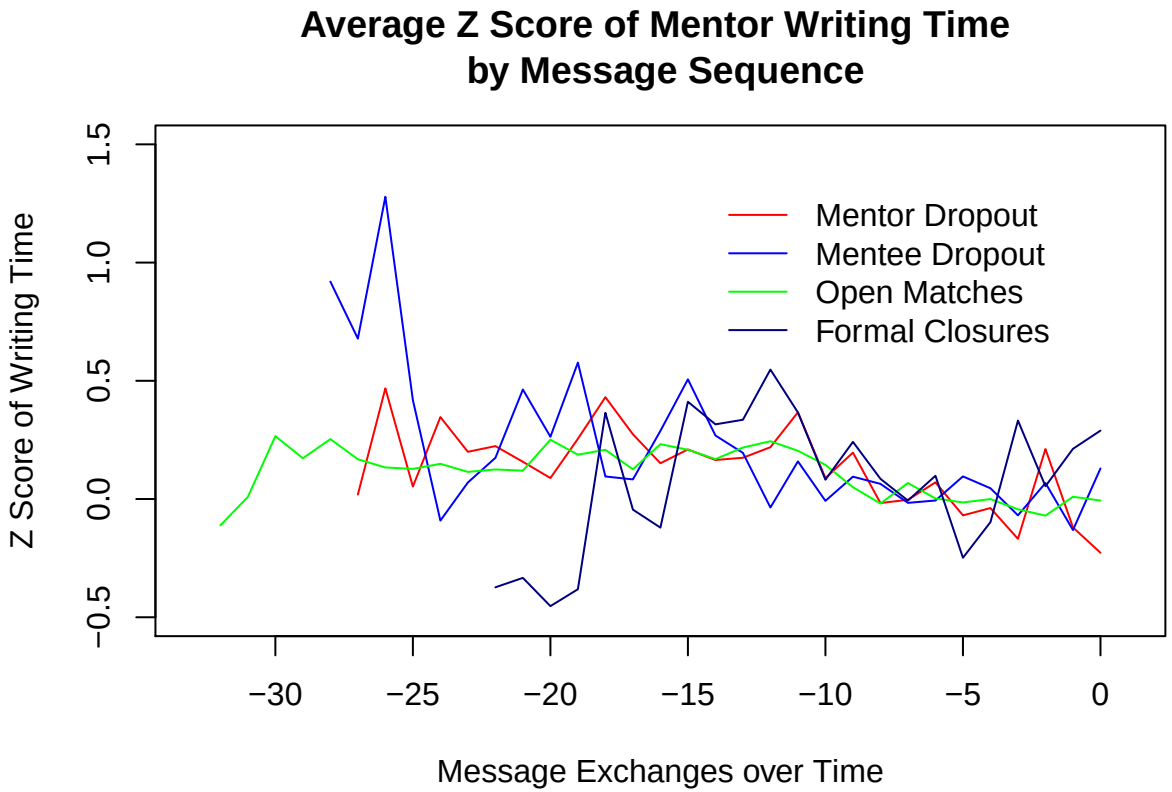
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 49997 rows containing non-finite values (stat_bin).
```




```
# Look at writing time by message sequence
writingtime.by.message.seq <- message.traffic.mentors.content %>% group_by(lessons.bf.last,
    combined.dropout.flag) %>% summarise(mean.mentor.zscore.writing.time = mean(mentor.zscore.writing.time,
    na.rm = TRUE), nobs = n())

# reshape to wide to make it easier to plot
writingtime.by.message.seq <- data.frame(writingtime.by.message.seq)
writingtime.by.message.seq <- reshape(writingtime.by.message.seq, timevar = "combined.dropout.flag",
    idvar = c("lessons.bf.last"), direction = "wide")

writingtime.by.message.seq <- writingtime.by.message.seq[order(writingtime.by.message.seq$lessons.bf.last),
```

```
    ]

# create time variable for lessons before last ascending order to last
# observation
writingtime.by.message.seq$x.for.line.chart <- -1 * writingtime.by.message.seq$lessons.bf.last


par(mfrow = c(1, 1))
plot(x = writingtime.by.message.seq$x.for.line.chart, writingtime.by.message.seq$`mean.mentor.zscore.writing.time.Mentor can no longer par
    main = "Average Z Score of Mentor Writing Time \n by Message Sequence",
    xlab = "Message Exchanges over Time", ylab = "Z Score of Writing Time",
    type = "l", col = "red", xlim = c(-33, 1), ylim = c(-0.5, 1.5))

par(new = TRUE)
plot(x = writingtime.by.message.seq$x.for.line.chart, writingtime.by.message.seq$`mean.mentor.zscore.writing.time.Mentee can no longer par
    main = "", xlab = "", ylab = "", type = "l", col = "blue", xlim = c(-33,
        1), ylim = c(-0.5, 1.5), axes = FALSE)

par(new = TRUE)
plot(x = writingtime.by.message.seq$x.for.line.chart, writingtime.by.message.seq$`mean.mentor.zscore.writing.time.Match Open`,
    main = "", xlab = "", ylab = "", type = "l", col = "green", xlim = c(-33,
        1), ylim = c(-0.5, 1.5), axes = FALSE)

leg.txt <- c("Mentor Dropout", "Mentee Dropout", "Open Matches", "Formal Closures")
legend(inset = c(0.1, 0.1), "topright", legend = leg.txt, lty = c(1, 1, 1, 1),
    col = c("red", "blue", "green", "navy"), bty = "n", cex = 1)

par(new = TRUE)
plot(x = writingtime.by.message.seq$x.for.line.chart, writingtime.by.message.seq$`mean.mentor.zscore.writing.time.Formal closure`,
    main = "", xlab = "", ylab = "", type = "l", col = "navy", xlim = c(-33,
        1), ylim = c(-0.5, 1.5), axes = FALSE)
```



Comment on writing time

The analysis of writing time may show a slight decline in writing time as mentors are in the program for a longer amount of time. There are too many outliers that are driving group means and creating huge variance to see significant differences among groups. Plotting a histogram of writing time grouped by dropout group shows no clear distinction of writing time among dropout groups.

b. Message length

```
# summarize word count
summary(message.traffic.mentors$canvas_word_cnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    94.0   156.0   190.6   242.0  6840.0   31611
```

```
summary(message.traffic.mentees$canvas_word_cnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   40.00   74.00   93.34  123.00 4172.00   23178
```

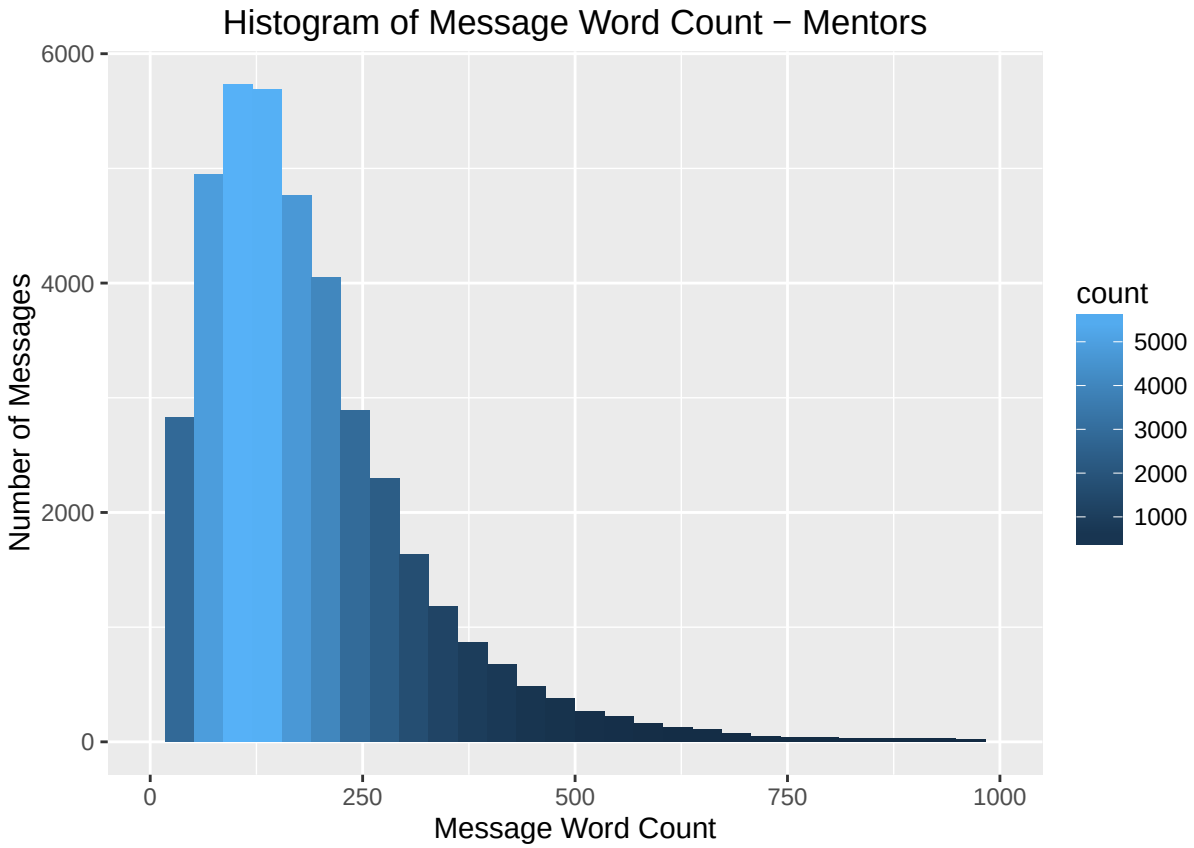
```
# look at message word count by group
message.word.count.by.group.summ <- message.traffic.mentors.content %>% group_by(combined.dropout.flag) %>%
    summarise(mean.word.count.mentor = mean(canvas_word_cnt), count.obs = n())
tt = pairwise.t.test(message.traffic.mentors.content$canvas_word_cnt, message.traffic.mentors.content$combined.dropout.flag,
    p.adjust.method = "bonferroni", na.rm = TRUE)
tt
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  message.traffic.mentors.content$canvas_word_cnt and message.traffic.mentors.content$combined.dropout.flag
##
##                                 Formal closure Match Open
## Match Open                      0.0033         -
## Mentee can no longer participate 0.6901        0.0038
## Mentor can no longer participate 0.0701        0.5256
##                                 Mentee can no longer participate
## Match Open                      -
## Mentee can no longer participate -
## Mentor can no longer participate 0.7808
##
## P value adjustment method: bonferroni
```

```
# word count shows a positively skewed distribution for both mentees and
# mentors.
word.count.mentors.hist <- ggplot(data = message.traffic.mentors.content, aes(x = canvas_word_cnt))
word.count.mentors.hist + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Message Word Count - Mentors") +
    xlim(0, 1000) + labs(x = "Message Word Count", y = "Number of Messages")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

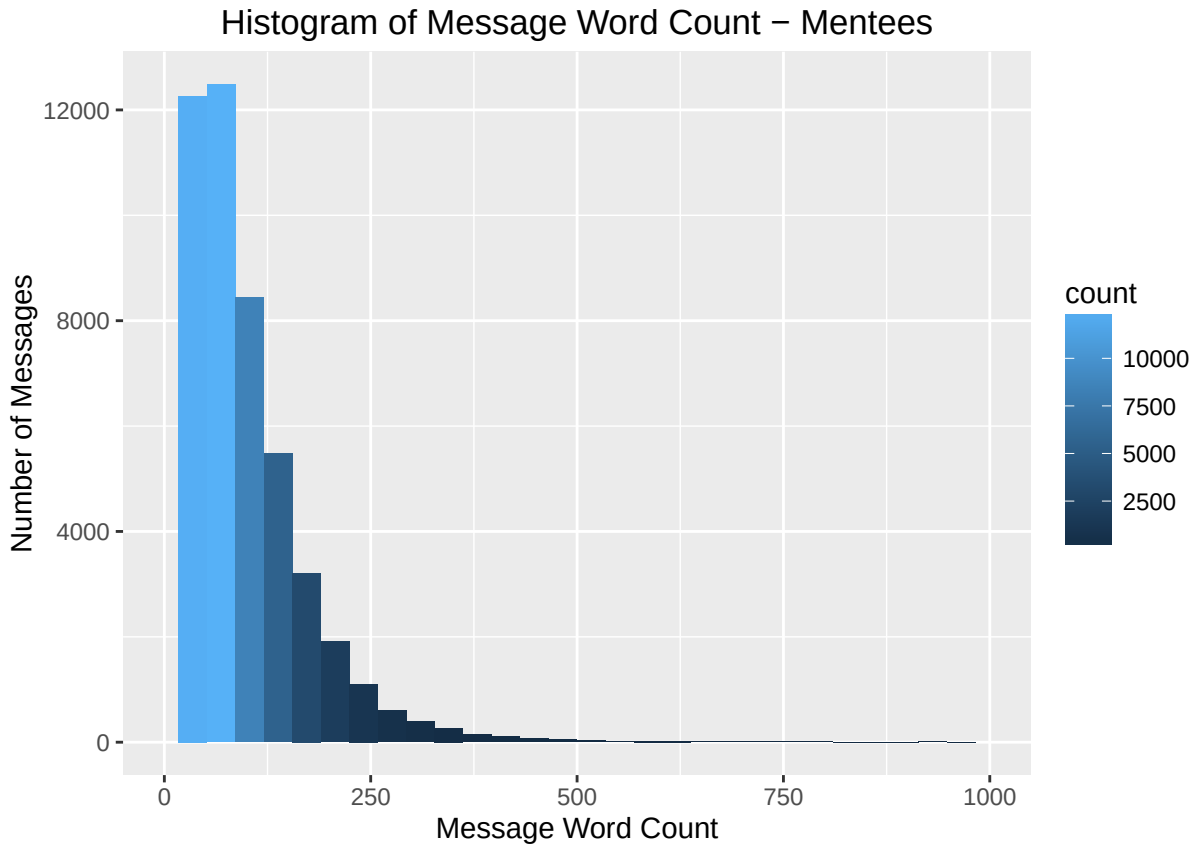
```
## Warning: Removed 156 rows containing non-finite values (stat_bin).
```




```
word.count.mentees.hist <- ggplot(data = message.traffic.mentees.content, aes(x = canvas_word_cnt))
word.count.mentees.hist + geom_histogram(aes(fill = ..count..)) + ggtitle("Histogram of Message Word Count - Mentees") +
    xlim(0, 1000) + labs(x = "Message Word Count", y = "Number of Messages")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

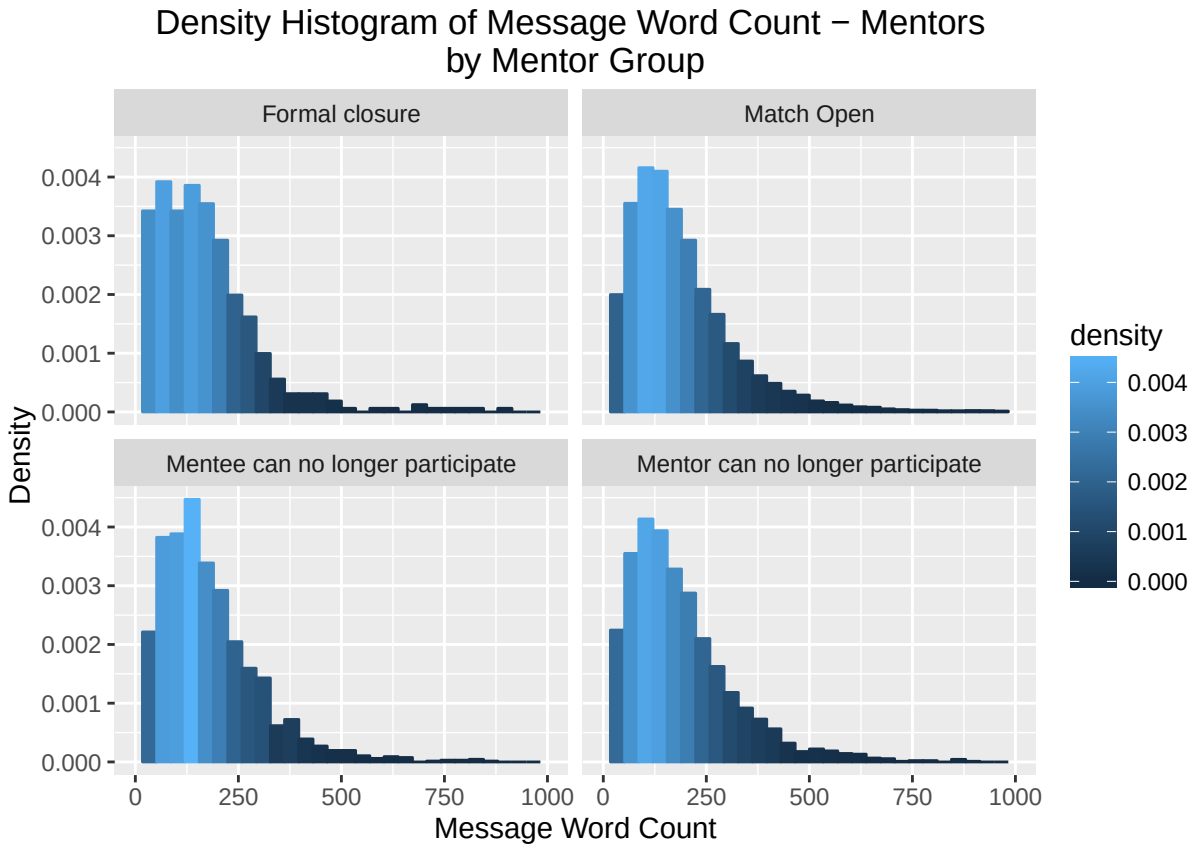
```
## Warning: Removed 31 rows containing non-finite values (stat_bin).
```




```
# look at word count histogram by dropout group.
word.count.hist.mentors.2 <- ggplot(data = message.traffic.mentors.content,
    aes(x = canvas_word_cnt, color = ..density.., y = ..density..))
word.count.hist.mentors.2 + geom_histogram(aes(fill = ..density..)) + ggtitle("Density Histogram of Message Word Count - Mentors \nby Ment
    labs(y = "Density", x = "Message Word Count") + xlim(c(0, 1000)) + facet_wrap(~combined.dropout.flag)
```

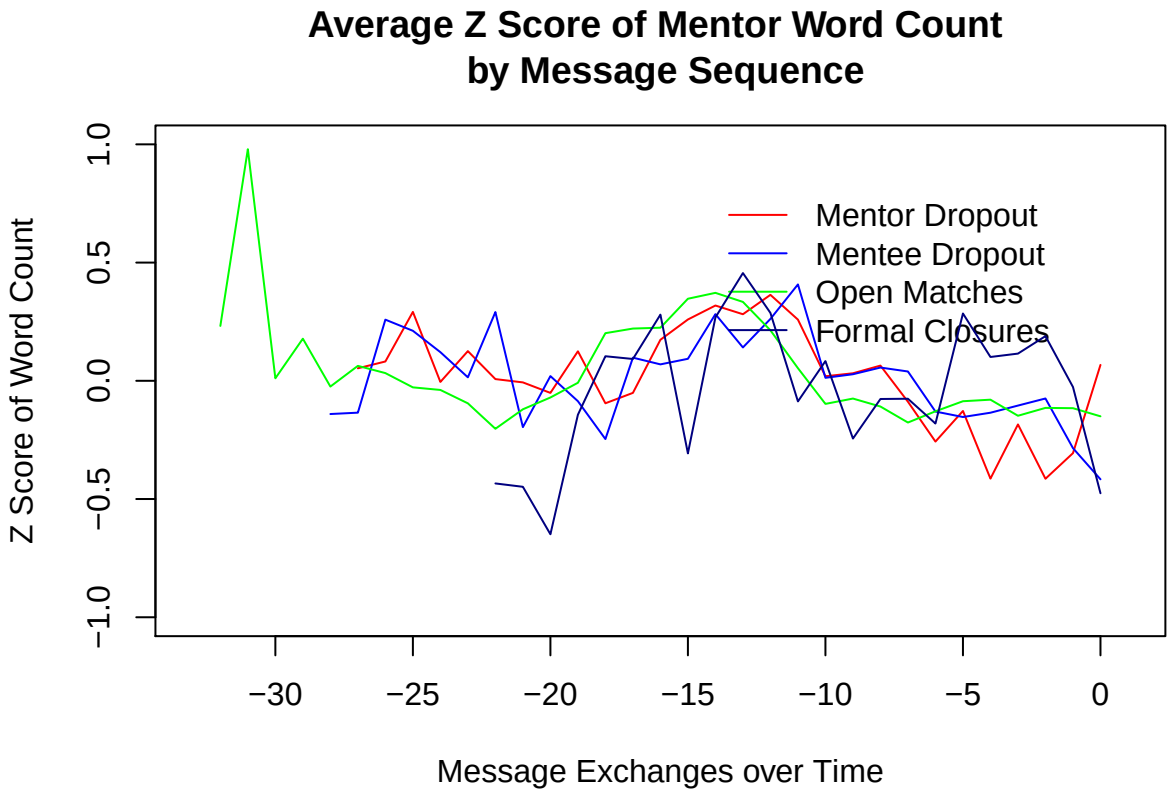
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 156 rows containing non-finite values (stat_bin).
```



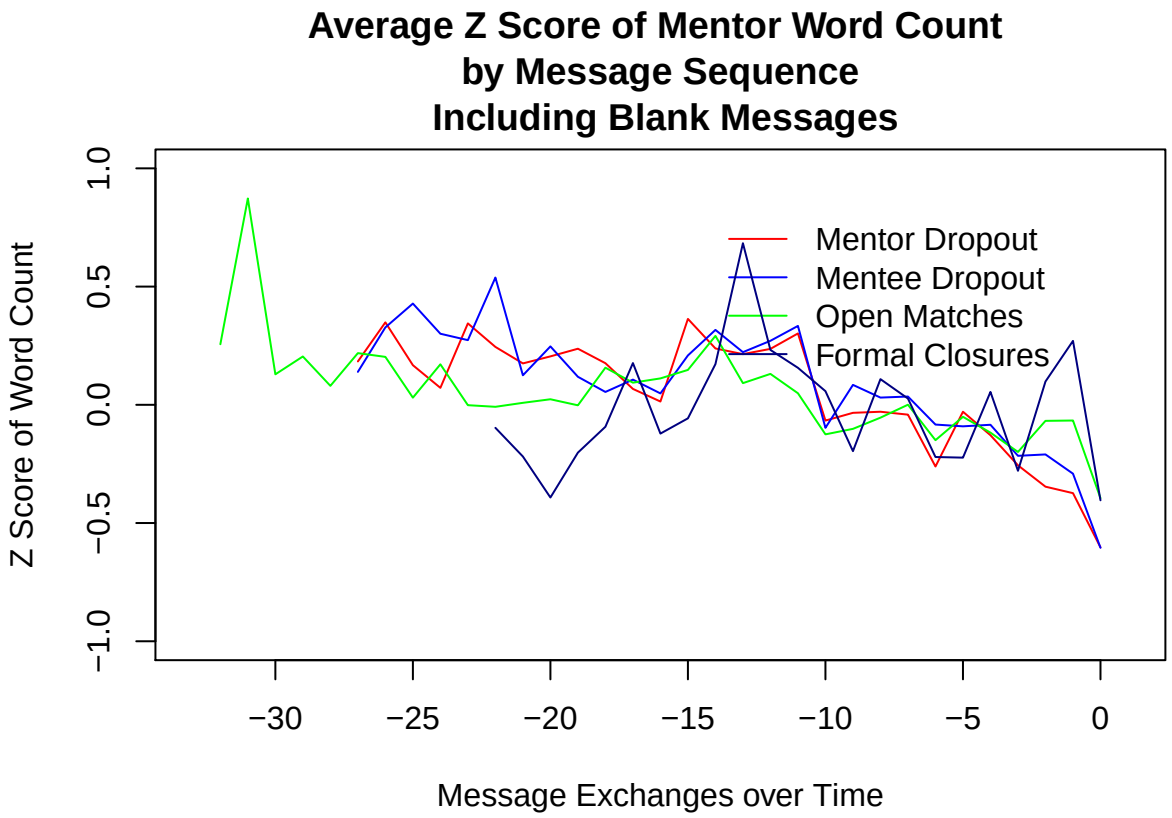

```
# message length over time - only messages that have content z score of word
# count by message sequence
wordcount.by.message.seq <- message.traffic.mentors.content %>% group_by(lessons.bf.last,
    combined.dropout.flag) %>% summarise(mean.mentor.zscore.wordcount = mean(mentor.zscore.wordcount,
    na.rm = TRUE), nobs = n())

# reshape to wide to make it easier to plot
wordcount.by.message.seq <- data.frame(wordcount.by.message.seq)
wordcount.by.message.seq <- reshape(wordcount.by.message.seq, timevar = "combined.dropout.flag",
    idvar = c("lessons.bf.last"), direction = "wide")
```

```r
wordcount.by.message.seq <- wordcount.by.message.seq[order(wordcount.by.message.seq$lessons.bf.last),
    ]

# create time variable for lessons before last ascending order to last
# observation
wordcount.by.message.seq$x.for.line.chart <- -1 * wordcount.by.message.seq$lessons.bf.last



par(mfrow = c(1, 1))
plot(x = wordcount.by.message.seq$x.for.line.chart, wordcount.by.message.seq$`mean.mentor.zscore.wordcount.Mentor can no longer participat
    main = "Average Z Score of Mentor Word Count \n by Message Sequence", xlab = "Message Exchanges over Time",
    ylab = "Z Score of Word Count", type = "l", col = "red", xlim = c(-33, 1),
    ylim = c(-1, 1))

par(new = TRUE)
plot(x = wordcount.by.message.seq$x.for.line.chart, wordcount.by.message.seq$`mean.mentor.zscore.wordcount.Mentee can no longer participat
    main = "", xlab = "", ylab = "", type = "l", col = "blue", xlim = c(-33,
        1), ylim = c(-1, 1), axes = FALSE)

par(new = TRUE)
plot(x = wordcount.by.message.seq$x.for.line.chart, wordcount.by.message.seq$`mean.mentor.zscore.wordcount.Match Open`,
    main = "", xlab = "", ylab = "", type = "l", col = "green", xlim = c(-33,
        1), ylim = c(-1, 1), axes = FALSE)

leg.txt <- c("Mentor Dropout", "Mentee Dropout", "Open Matches", "Formal Closures")
legend(inset = c(0.1, 0.1), "topright", legend = leg.txt, lty = c(1, 1, 1, 1),
    col = c("red", "blue", "green", "navy"), bty = "n", cex = 1)

par(new = TRUE)
plot(x = wordcount.by.message.seq$x.for.line.chart, wordcount.by.message.seq$`mean.mentor.zscore.wordcount.Formal closure`,
    main = "", xlab = "", ylab = "", type = "l", col = "navy", xlim = c(-33,
        1), ylim = c(-1, 1), axes = FALSE)
```




```
# message length over time - all messages including blank messages as length
# zero. z score of word count by message sequence
wordcount.by.message.seq.w.zeroes <- message.traffic.mentors %>% group_by(lessons.bf.last,
    combined.dropout.flag) %>% summarise(mean.mentor.zscore.wordcount.w.zeroes = mean(mentor.zscore.wordcount.w.zeroes,
    na.rm = TRUE), nobs = n())

# reshape to wide to make it easier to plot
wordcount.by.message.seq.w.zeroes <- data.frame(wordcount.by.message.seq.w.zeroes)
wordcount.by.message.seq.w.zeroes <- reshape(wordcount.by.message.seq.w.zeroes,
    timevar = "combined.dropout.flag", idvar = c("lessons.bf.last"), direction = "wide")

wordcount.by.message.seq.w.zeroes <- wordcount.by.message.seq.w.zeroes[order(wordcount.by.message.seq.w.zeroes$lessons.bf.last),
    ]
```

```r
# create time variable for lessons before last ascending order to last
# observation
wordcount.by.message.seq.w.zeroes$x.for.line.chart <- -1 * wordcount.by.message.seq.w.zeroes$lessons.bf.last

# set z scores to NA for data points with less than 5 messages.
wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Formal closure` <- ifelse(wordcount.by.message.seq.w.zeroes$`nobs
    5, NA, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Formal closure`)
wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Mentee can no longer participate` <- ifelse(wordcount.by.message.
    5, NA, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Mentee can no longer participate`)
wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Mentor can no longer participate` <- ifelse(wordcount.by.message.
    5, NA, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Mentor can no longer participate`)



par(mfrow = c(1, 1))
plot(x = wordcount.by.message.seq.w.zeroes$x.for.line.chart, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Ment
    main = "Average Z Score of Mentor Word Count \n by Message Sequence \n Including Blank Messages",
    xlab = "Message Exchanges over Time", ylab = "Z Score of Word Count", type = "l",
    col = "red", xlim = c(-33, 1), ylim = c(-1, 1))

par(new = TRUE)
plot(x = wordcount.by.message.seq.w.zeroes$x.for.line.chart, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Ment
    main = "", xlab = "", ylab = "", type = "l", col = "blue", xlim = c(-33,
        1), ylim = c(-1, 1), axes = FALSE)

par(new = TRUE)
plot(x = wordcount.by.message.seq.w.zeroes$x.for.line.chart, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Matc
    main = "", xlab = "", ylab = "", type = "l", col = "green", xlim = c(-33,
        1), ylim = c(-1, 1), axes = FALSE)

leg.txt <- c("Mentor Dropout", "Mentee Dropout", "Open Matches", "Formal Closures")
legend(inset = c(0.1, 0.1), "topright", legend = leg.txt, lty = c(1, 1, 1, 1),
    col = c("red", "blue", "green", "navy"), bty = "n", cex = 1)

par(new = TRUE)
plot(x = wordcount.by.message.seq.w.zeroes$x.for.line.chart, wordcount.by.message.seq.w.zeroes$`mean.mentor.zscore.wordcount.w.zeroes.Form
    main = "", xlab = "", ylab = "", type = "l", col = "navy", xlim = c(-33,
        1), ylim = c(-1, 1), axes = FALSE)
```



Comment on message length

Again, no apparent distinction between dropout groups in either raw word count or pattern of wordcount over time. When only looking at messages that include any content, there doesn't appear to be any significant time trend of message length as mentors are in the program for a longer period of time. When we include messages with no content, however, there is quite a distinct downward trend in message length as mentors are in the program for a longer period of time, signaling declining mentor engagement over time.

c. Blank messages

```
# count messages with no content
sum(is.na(message.traffic.mentors$canvas_word_cnt))/nrow(message.traffic.mentors)
```

```
## [1] 0.4348562
```

```
sum(is.na(message.traffic.mentees$canvas_word_cnt))/nrow(message.traffic.mentees)
```

```
## [1] 0.3061985
```

```
summary(message.traffic.mentors$canvas_word_cnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    94.0   156.0   190.6   242.0  6840.0   31611
```

```
# get share of blank messages by dropout group
blank.message.share.by.group.summ <- message.traffic.mentors %>% group_by(combined.dropout.flag) %>%
    summarise(share.of.blank.messages = mean(blank.message.flag), count.obs = n())

blank.message.share.by.group.summ
```

```
## # A tibble: 4 x 3
##              combined.dropout.flag share.of.blank.messages count.obs
##                              <chr>                   <dbl>     <int>
## 1                   Formal closure               0.6611111      1440
## 2                       Match Open               0.4067220     60518
## 3 Mentee can no longer participate               0.5901673      4902
## 4 Mentor can no longer participate               0.5403737      5833
```

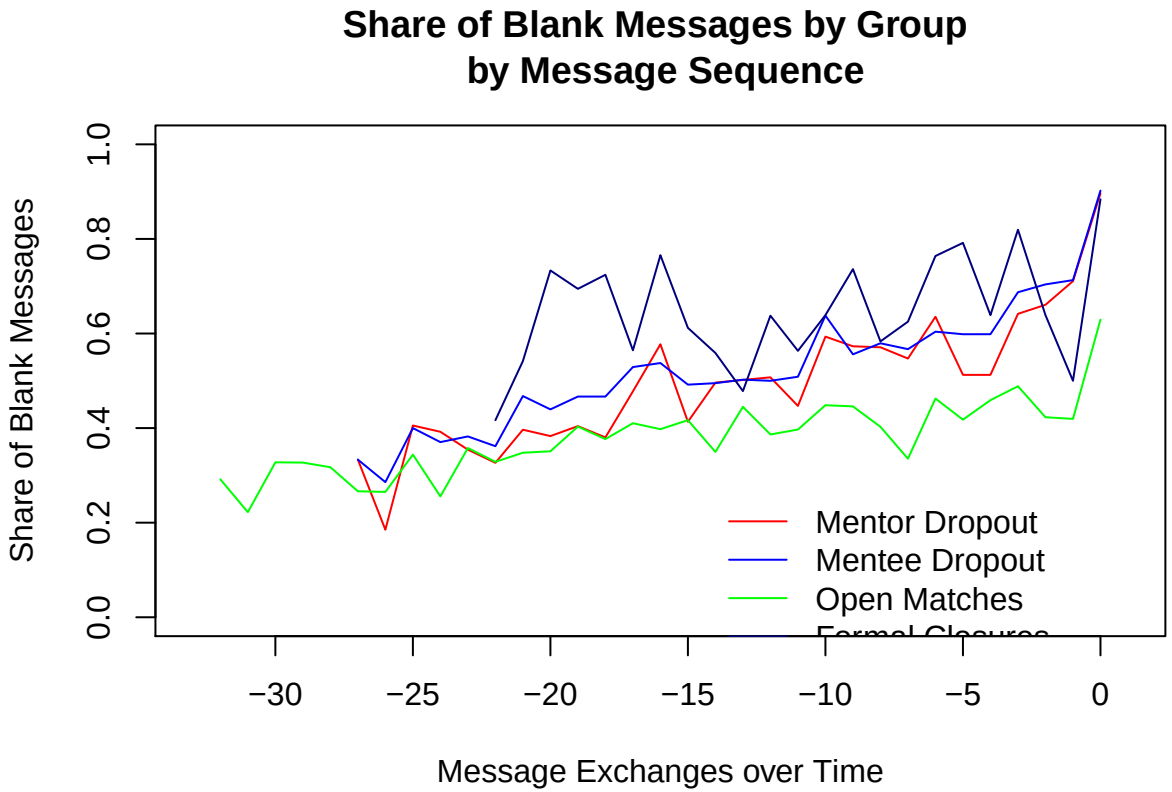
```
tt = pairwise.t.test(message.traffic.mentors$blank.message.flag, message.traffic.mentors$combined.dropout.flag,
    p.adjust.method = "bonferroni", na.rm = TRUE)
tt
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  message.traffic.mentors$blank.message.flag and message.traffic.mentors$combined.dropout.flag
##
##                                Formal closure Match Open
## Match Open                     < 2e-16        -
```

```
## Mentee can no longer participate 8.8e-06        < 2e-16
## Mentor can no longer participate 4.2e-16        < 2e-16
##                                  Mentee can no longer participate
## Match Open                       -
## Mentee can no longer participate -
## Mentor can no longer participate 1.0e-06
##
## P value adjustment method: bonferroni
```

```
# strangely formal closures seem to have a statistically significantly
# higher prevalence of blank messages.

# Look at blank message prevalence over time
blank.message.pct.by.message.seq <- message.traffic.mentors %>% group_by(lessons.bf.last,
    combined.dropout.flag) %>% summarise(blank.message.pct = mean(blank.message.flag,
    na.rm = TRUE), nobs = n())

# reshape to wide to make it easier to plot
blank.message.pct.by.message.seq <- data.frame(blank.message.pct.by.message.seq)
blank.message.pct.by.message.seq <- reshape(blank.message.pct.by.message.seq,
    timevar = "combined.dropout.flag", idvar = c("lessons.bf.last"), direction = "wide")

blank.message.pct.by.message.seq <- blank.message.pct.by.message.seq[order(blank.message.pct.by.message.seq$lessons.bf.last),
    ]

# create time variable for lessons before last ascending order to last
# observation
blank.message.pct.by.message.seq$x.for.line.chart <- -1 * blank.message.pct.by.message.seq$lessons.bf.last

# Remove observations at the beginning of the series with less than five
# observations.
blank.message.pct.by.message.seq$`blank.message.pct.Formal closure` <- ifelse(blank.message.pct.by.message.seq$`nobs.Formal closure` <
    5, NA, blank.message.pct.by.message.seq$`blank.message.pct.Formal closure`)
blank.message.pct.by.message.seq$`blank.message.pct.Mentee can no longer participate` <- ifelse(blank.message.pct.by.message.seq$`nobs.Men
    5, NA, blank.message.pct.by.message.seq$`blank.message.pct.Mentee can no longer participate`)
blank.message.pct.by.message.seq$`blank.message.pct.Mentor can no longer participate` <- ifelse(blank.message.pct.by.message.seq$`nobs.Men
    5, NA, blank.message.pct.by.message.seq$`blank.message.pct.Mentor can no longer participate`)
```

```r
# Graph over time
par(mfrow = c(1, 1))
plot(x = blank.message.pct.by.message.seq$x.for.line.chart, blank.message.pct.by.message.seq$`blank.message.pct.Mentor can no longer parti
    main = "Share of Blank Messages by Group \n by Message Sequence", xlab = "Message Exchanges over Time",
    ylab = "Share of Blank Messages", type = "l", col = "red", xlim = c(-33,
        1), ylim = c(0, 1))

par(new = TRUE)
plot(x = blank.message.pct.by.message.seq$x.for.line.chart, blank.message.pct.by.message.seq$`blank.message.pct.Mentee can no longer parti
    main = "", xlab = "", ylab = "", type = "l", col = "blue", xlim = c(-33,
        1), ylim = c(0, 1), axes = FALSE)

par(new = TRUE)
plot(x = blank.message.pct.by.message.seq$x.for.line.chart, blank.message.pct.by.message.seq$`blank.message.pct.Match Open`,
    main = "", xlab = "", ylab = "", type = "l", col = "green", xlim = c(-33,
        1), ylim = c(0, 1), axes = FALSE)

leg.txt <- c("Mentor Dropout", "Mentee Dropout", "Open Matches", "Formal Closures")
legend(inset = c(0.1, 0.7), "topright", legend = leg.txt, lty = c(1, 1, 1, 1),
    col = c("red", "blue", "green", "navy"), bty = "n", cex = 1)

par(new = TRUE)
plot(x = blank.message.pct.by.message.seq$x.for.line.chart, blank.message.pct.by.message.seq$`blank.message.pct.Formal closure`,
    main = "", xlab = "", ylab = "", type = "l", col = "navy", xlim = c(-33,
        1), ylim = c(0, 1), axes = FALSE)
```



Comment on blank messages

In mentor message traffic limited to mentors that appear in the match history file and who were flagged as active at the beginning of the 15/16 school year, there is an extremely high prevalenceof blank messages (>40%). Graphing the prevalence of blank messages over time shows an increasing prevalence of blank messages over time across all groups, indicating declining mentor engagement over time. Note have removed a few early observations with less than five observations that showed 100% blank messages.

IV. Merged mentor and mentee message traffic analysis

a. Merge and Prep Data for Analysis

```
# First merge the two dataframes
message.traffic.merged <- merge(message.traffic.mentors, message.traffic.mentees,
    by = c("pair_id", "curriculum_sequence"), suffixes = c(".mentor", ".mentee"))

# Get response time response time and lesson length
message.traffic.merged$response.time <- as.numeric(difftime(message.traffic.merged$user_first_sub.mentor,
    message.traffic.merged$user_first_sub.mentee, units = c("mins")))
message.traffic.merged$lesson.length <- as.numeric(difftime(message.traffic.merged$lesson_close.mentor,
    message.traffic.merged$lesson_launch.mentor, units = c("mins")))
summary(message.traffic.merged$response.time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -518800    1486    7602    8519    9839  332700   28005
```

```
# create indicator for non-response, i.e. where a mentee sent a message and
# the mentor never responded
message.traffic.merged$mentor.ignore.mentee.flag <- ifelse(message.traffic.merged$blank.message.flag.mentee ==
    0 & message.traffic.merged$blank.message.flag.mentor == 1, 1, 0)

# remove program partnership ended observations
message.traffic.merged <- subset(message.traffic.merged, message.traffic.merged$combined.dropout.flag.mentor !=
    "Program Partnership ended" & message.traffic.merged$combined.dropout.flag.mentor !=
    "")

# limit to valid response times (less than the length of the lesson and
# responses that occurred AFTER the mentee's initial message).
message.traffic.merged.valid.resp <- subset(message.traffic.merged, message.traffic.merged$response.time <
    message.traffic.merged$lesson.length)
message.traffic.merged.valid.resp <- subset(message.traffic.merged, message.traffic.merged$response.time <=
    14400 & message.traffic.merged$response.time > 0)
```

b. Ignore Rates

```
# get the rate of non response by mentors overall.
sum(message.traffic.merged$mentor.ignore.mentee.flag)/nrow(message.traffic.merged)
```

```
## [1] 0.2254615
```

```
# compare ignore rate by group
ignore.rates.by.group.summ <- message.traffic.merged %>% group_by(combined.dropout.flag.mentor) %>%
    summarise(ignore.rate.by.group = mean(mentor.ignore.mentee.flag, na.rm = TRUE),
        count.obs = n())

ignore.rates.by.group.summ
```

```
## # A tibble: 4 x 3
##       combined.dropout.flag.mentor ignore.rate.by.group count.obs
##                              <chr>                <dbl>     <int>
## 1                   Formal closure            0.2341904      1439
## 2                       Match Open            0.2243848     60267
## 3 Mentee can no longer participate            0.1630114      4742
## 4 Mentor can no longer participate            0.2871042      5653
```

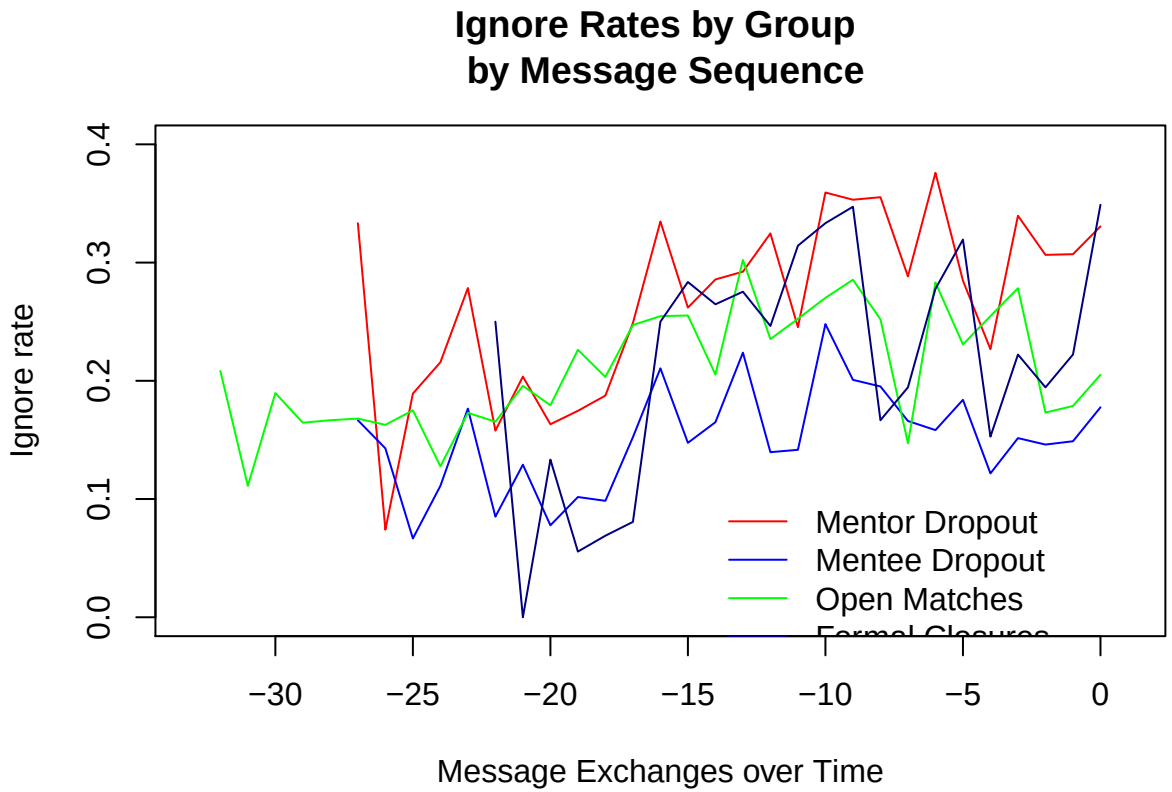
```
tt <- pairwise.t.test(message.traffic.merged$mentor.ignore.mentee.flag, message.traffic.merged$combined.dropout.flag.mentor,
    p.adjust.method = "bonferroni")
tt
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  message.traffic.merged$mentor.ignore.mentee.flag and message.traffic.merged$combined.dropout.flag.mentor
##
##                                 Formal closure Match Open
## Match Open                      1e+00          -
## Mentee can no longer participate 8.7e-08       < 2e-16
## Mentor can no longer participate 1e-04         < 2e-16
##                                 Mentee can no longer participate
```

```
## Match Open                        -
## Mentee can no longer participate -
## Mentor can no longer participate < 2e-16
##
## P value adjustment method: bonferroni
```

```
# Look at ignore rates over time by group
ignore.rate.by.message.seq <- message.traffic.merged %>% group_by(lessons.bf.last.mentor,
    combined.dropout.flag.mentor) %>% summarise(ignore.rate = mean(mentor.ignore.mentee.flag,
    na.rm = TRUE), nobs = n())

# reshape to wide to make it easier to plot
ignore.rate.by.message.seq <- data.frame(ignore.rate.by.message.seq)
ignore.rate.by.message.seq <- reshape(ignore.rate.by.message.seq, timevar = "combined.dropout.flag.mentor",
    idvar = c("lessons.bf.last.mentor"), direction = "wide")

ignore.rate.by.message.seq <- ignore.rate.by.message.seq[order(blank.message.pct.by.message.seq$lessons.bf.last),
    ]

# create time variable for lessons before last ascending order to last
# observation
ignore.rate.by.message.seq$x.for.line.chart <- -1 * ignore.rate.by.message.seq$lessons.bf.last

# Remove observations at the beginning of the series with less than five
# observations.
ignore.rate.by.message.seq$`ignore.rate.Formal closure` <- ifelse(ignore.rate.by.message.seq$`nobs.Formal closure` <
    5, NA, ignore.rate.by.message.seq$`ignore.rate.Formal closure`)
ignore.rate.by.message.seq$`ignore.rate.Mentee can no longer participate` <- ifelse(ignore.rate.by.message.seq$`nobs.Mentee can no longer
    5, NA, ignore.rate.by.message.seq$`ignore.rate.Mentee can no longer participate`)
ignore.rate.by.message.seq$`ignore.rate.Mentor can no longer participate` <- ifelse(ignore.rate.by.message.seq$`nobs.Mentor can no longer
    5, NA, ignore.rate.by.message.seq$`ignore.rate.Mentor can no longer participate`)

# Plot ignore rate over time
par(mfrow = c(1, 1))
plot(x = ignore.rate.by.message.seq$x.for.line.chart, ignore.rate.by.message.seq$`ignore.rate.Mentor can no longer participate`,
    main = "Ignore Rates by Group \n by Message Sequence", xlab = "Message Exchanges over Time",
    ylab = "Ignore rate", type = "l", col = "red", xlim = c(-33, 1), ylim = c(0,
        0.4))
```

```r
par(new = TRUE)
plot(x = ignore.rate.by.message.seq$x.for.line.chart, ignore.rate.by.message.seq$`ignore.rate.Mentee can no longer participate`,
    main = "", xlab = "", ylab = "", type = "l", col = "blue", xlim = c(-33,
        1), ylim = c(0, 0.4), axes = FALSE)

par(new = TRUE)
plot(x = ignore.rate.by.message.seq$x.for.line.chart, ignore.rate.by.message.seq$`ignore.rate.Match Open`,
    main = "", xlab = "", ylab = "", type = "l", col = "green", xlim = c(-33,
        1), ylim = c(0, 0.4), axes = FALSE)

leg.txt <- c("Mentor Dropout", "Mentee Dropout", "Open Matches", "Formal Closures")
legend(inset = c(0.1, 0.7), "topright", legend = leg.txt, lty = c(1, 1, 1, 1),
    col = c("red", "blue", "green", "navy"), bty = "n", cex = 1)

par(new = TRUE)
plot(x = ignore.rate.by.message.seq$x.for.line.chart, ignore.rate.by.message.seq$`ignore.rate.Formal closure`,
    main = "", xlab = "", ylab = "", type = "l", col = "navy", xlim = c(-33,
        1), ylim = c(0, 0.4), axes = FALSE)
```



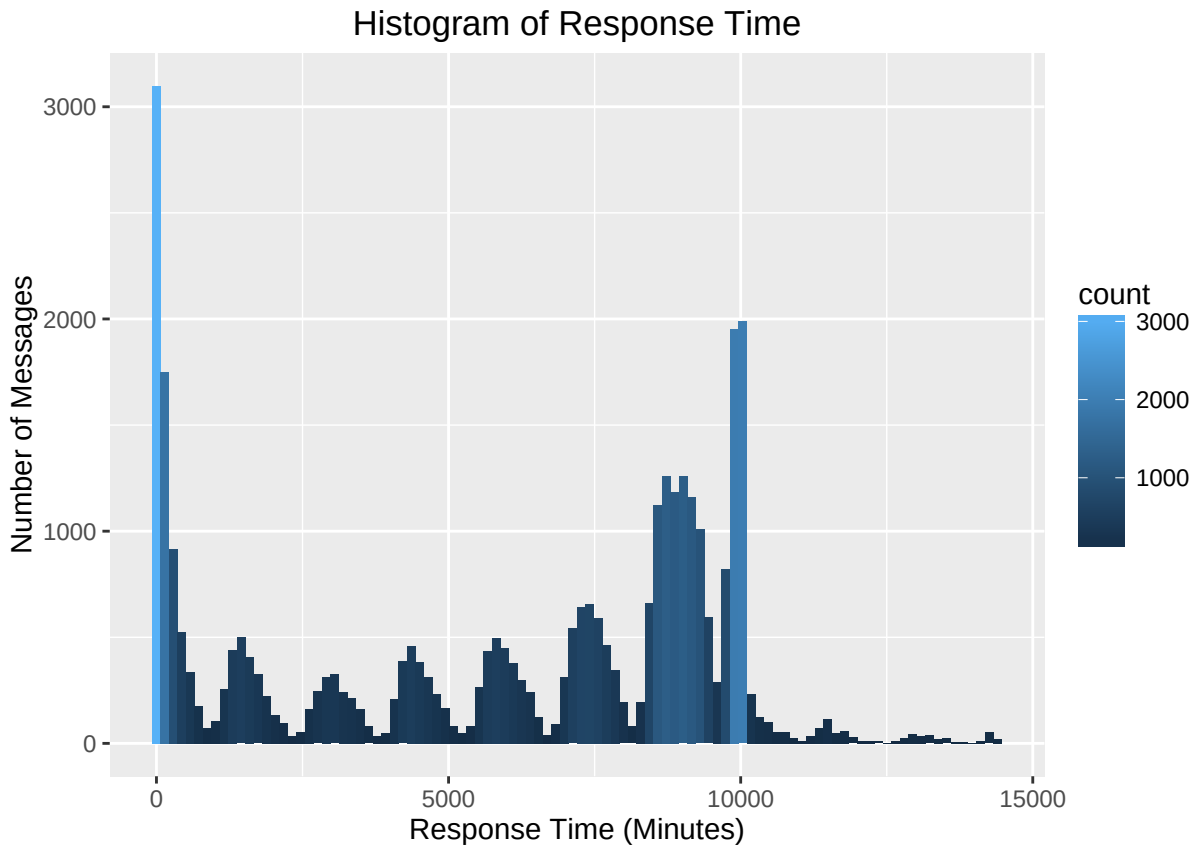
Comment on ignore rate

Ignore rate is the first instance thus far where we have seen a statistically significant difference between mentor dropouts and formal closures. Mentor dropouts do seem to have a statisticcally significantly higher ignore rate than formal closures. All mentor groups appear to show an increasing ignore rate over time, yet anothe signal of declining mentor engagement. Note data points on this line graph of less than five messages have been removed to remove erratic and high ignore rates at the beginning of each series.

c. Response times

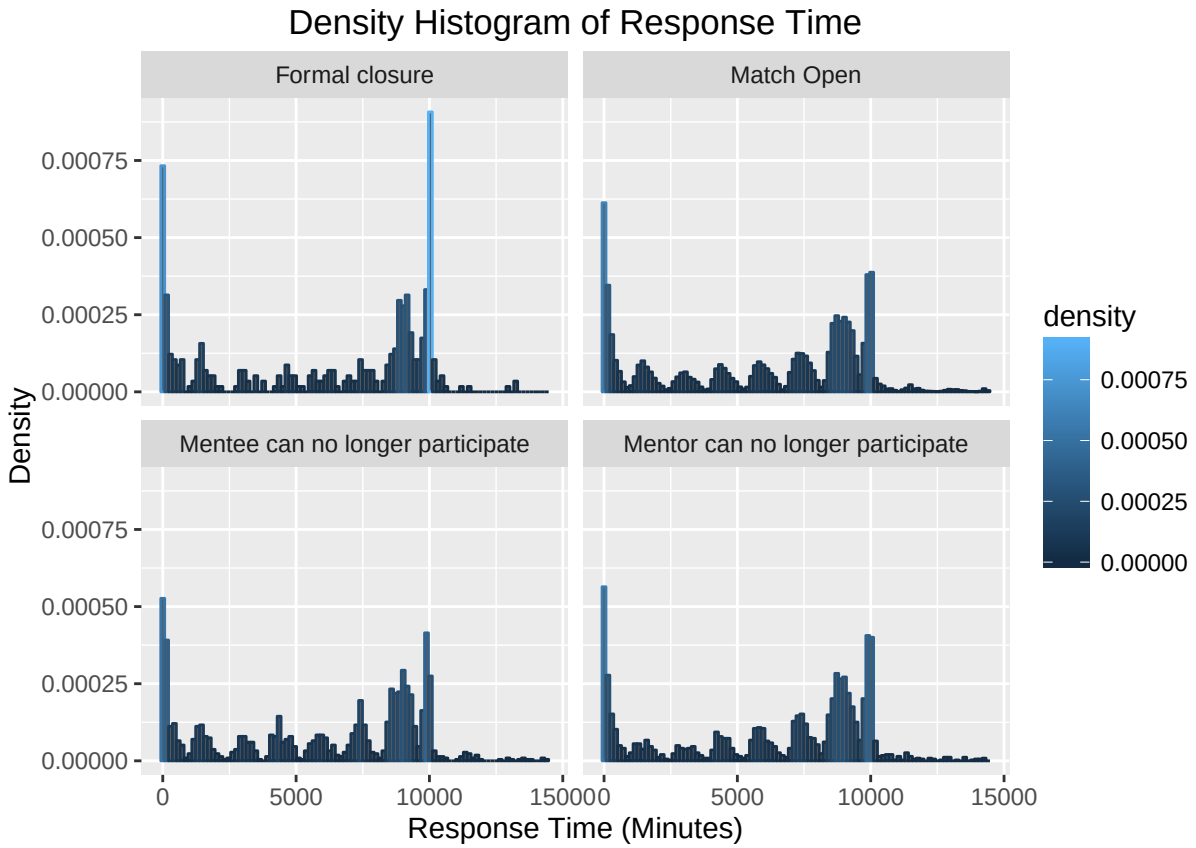
```
# look at distribution of valid response times. response time has a very
# interesting distribution. There are a lot of very timely responses and a
```

```
# lot of 'oh crap' response times where the mentor responded at the very end
# of the lesson. Additionally, each day of the week seems to have its own
# normal distribution of response times.
response.time.hist <- ggplot(data = message.traffic.merged.valid.resp, aes(x = response.time))
response.time.hist + geom_histogram(aes(fill = ..count..), bins = 100) + ggtitle("Histogram of Response Time") +
    labs(y = "Number of Messages", x = "Response Time (Minutes)")
```




```
# look at the same histogram but grouped by mentor groups
response.time.hist.2 <- ggplot(data = message.traffic.merged.valid.resp, aes(x = response.time,
    color = ..density.., y = ..density..))
response.time.hist.2 + geom_histogram(bins = 100) + ggtitle("Density Histogram of Response Time") +
```

```
    labs(y = "Density", x = "Response Time (Minutes)") + facet_wrap(~combined.dropout.flag.mentor)
```




```
# Compare response time across groups
response.time.by.group.summ <- message.traffic.merged.valid.resp %>% group_by(combined.dropout.flag.mentor) %>%
    summarise(mean.response.time = mean(response.time, na.rm = TRUE), count.obs = n())

response.time.by.group.summ
```

```
## # A tibble: 4 x 3
##       combined.dropout.flag.mentor mean.response.time count.obs
```

```
##                              <chr>             <dbl>     <int>
## 1                   Formal closure          6111.820       394
## 2                       Match Open          5883.239     30851
## 3 Mentee can no longer participate          5817.767      1475
## 4 Mentor can no longer participate          6278.809      2352
```

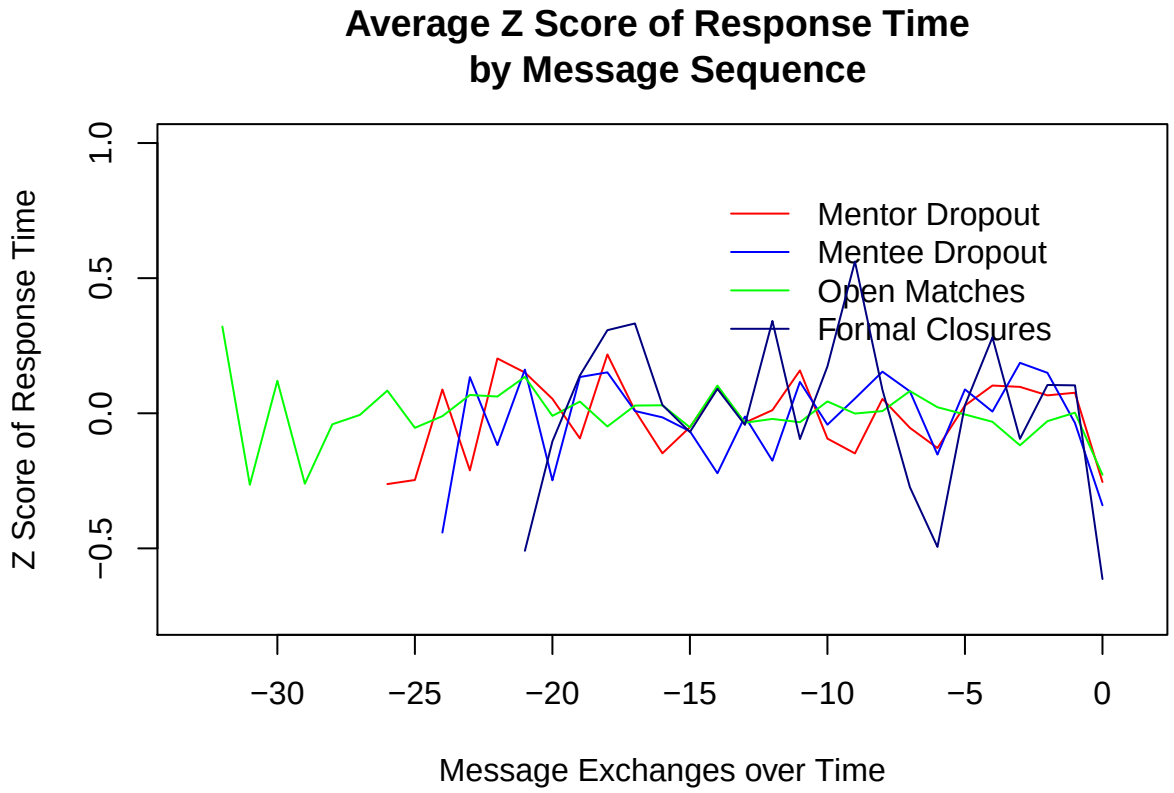
```
tt <- pairwise.t.test(message.traffic.merged.valid.resp$response.time, message.traffic.merged.valid.resp$combined.dropout.flag.mentor,
    p.adjust.method = "bonferroni")
tt
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  message.traffic.merged.valid.resp$response.time and message.traffic.merged.valid.resp$combined.dropout.flag.mentor
##
##                                 Formal closure Match Open
## Match Open                      1.0000         -
## Mentee can no longer participate 1.0000        1.0000
## Mentor can no longer participate 1.0000        6.5e-06
##                                 Mentee can no longer participate
## Match Open                      -
## Mentee can no longer participate -
## Mentor can no longer participate 0.0015
##
## P value adjustment method: bonferroni
```

```
# calculate response time z score
message.traffic.merged.valid.resp <- message.traffic.merged.valid.resp %>% group_by(mentor_persona_id) %>%
    mutate(mentor.mean.response.time = mean(response.time, na.rm = TRUE), mentor.sd.response.time = sd(response.time,
        na.rm = TRUE))
message.traffic.merged.valid.resp$mentor.zscore.response.time <- (message.traffic.merged.valid.resp$response.time -
    message.traffic.merged.valid.resp$mentor.mean.response.time)/message.traffic.merged.valid.resp$mentor.sd.response.time


# look at how response times change as mentors are involved in the program
# for longer periods of time.
response.time.by.message.seq <- message.traffic.merged.valid.resp %>% group_by(lessons.bf.last.mentor,
    combined.dropout.flag.mentor) %>% summarise(mean.response.time.zscore = mean(mentor.zscore.response.time,
```

```
    na.rm = TRUE), nobs = n())

# reshape to wide to make it easier to plot
response.time.by.message.seq <- data.frame(response.time.by.message.seq)
response.time.by.message.seq <- reshape(response.time.by.message.seq, timevar = "combined.dropout.flag.mentor",
    idvar = c("lessons.bf.last.mentor"), direction = "wide")

response.time.by.message.seq <- response.time.by.message.seq[order(blank.message.pct.by.message.seq$lessons.bf.last),
    ]

# create time variable for lessons before last ascending order to last
# observation
response.time.by.message.seq$x.for.line.chart <- -1 * response.time.by.message.seq$lessons.bf.last

# Remove observations at the beginning of the series with less than five
# observations.
response.time.by.message.seq$`mean.response.time.zscore.Formal closure` <- ifelse(response.time.by.message.seq$`nobs.Formal closure` <
    5, NA, response.time.by.message.seq$`mean.response.time.zscore.Formal closure`)
response.time.by.message.seq$`mean.response.time.zscore.Mentee can no longer participate` <- ifelse(response.time.by.message.seq$`nobs.Men
    5, NA, response.time.by.message.seq$`mean.response.time.zscore.Mentee can no longer participate`)
response.time.by.message.seq$`mean.response.time.zscore.Mentor can no longer participate` <- ifelse(response.time.by.message.seq$`nobs.Men
    5, NA, response.time.by.message.seq$`mean.response.time.zscore.Mentor can no longer participate`)

# Plot response time over time
par(mfrow = c(1, 1))
plot(x = response.time.by.message.seq$x.for.line.chart, response.time.by.message.seq$`mean.response.time.zscore.Mentor can no longer parti
    main = "Average Z Score of Response Time \n by Message Sequence", xlab = "Message Exchanges over Time",
    ylab = "Z Score of Response Time", type = "l", col = "red", xlim = c(-33,
        1), ylim = c(-0.75, 1))

par(new = TRUE)
plot(x = response.time.by.message.seq$x.for.line.chart, response.time.by.message.seq$`mean.response.time.zscore.Mentee can no longer parti
    main = "", xlab = "", ylab = "", type = "l", col = "blue", xlim = c(-33,
        1), ylim = c(-0.75, 1), axes = FALSE)

par(new = TRUE)
plot(x = response.time.by.message.seq$x.for.line.chart, response.time.by.message.seq$`mean.response.time.zscore.Match Open`,
    main = "", xlab = "", ylab = "", type = "l", col = "green", xlim = c(-33,
```

```
        1), ylim = c(-0.75, 1), axes = FALSE)

leg.txt <- c("Mentor Dropout", "Mentee Dropout", "Open Matches", "Formal Closures")
legend(inset = c(0.1, 0.1), "topright", legend = leg.txt, lty = c(1, 1, 1, 1),
    col = c("red", "blue", "green", "navy"), bty = "n", cex = 1)

par(new = TRUE)
plot(x = response.time.by.message.seq$x.for.line.chart, response.time.by.message.seq$`mean.response.time.zscore.Formal closure`,
    main = "", xlab = "", ylab = "", type = "l", col = "navy", xlim = c(-33,
        1), ylim = c(-0.75, 1), axes = FALSE)
```



Comment on Response Time

Response time has a very interesting periodic distribution. We observe a lot of very timely responses and a lot of last minute responses times where the mentor responded at the very end of the lesson. In between these two extremes, there are peaks for response times at 24 hour intervals from the beginning of the week. We hoped to see mentors who drop out having relatively more last minute responses than mentors who do not drop out, but again, we did not observe any clear distinctions between message groups. Response time measured by message sequence seems relatively stable over time and does not seem to show an increasing or decreasing trend like some of the other metrics in our analysis.

V. Conclusion

In conclusion, we repeatedly observe distinct time trends across various metrics that seem to signal declining mentor engagement over time. We do not observe any clear distinctions in the time trends for different mentor groups. We do observe some statistically significant distinctions across mentor groups on stationary metrics such as ignore rates, but these results have an unexpected direction that may warrant further investigation. Our main takeaway is that length of mentor involvement in the program appears to be the biggest driver of mentor message behavior.