

Data Processing

2023-11-28

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
```

```
## 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)
```

```
train_logs <- read.csv('data/train_logs.csv')
```

```
train_scores <- read.csv('data/train_scores.csv')
```

```
full_data <- merge(train_logs, train_scores, by = "id")
```

```
# remove NULL values
```

```
full_data <- na.omit(full_data)
```

```
set.seed(1)
```

```
unique_ids <- unique(full_data$id)
```

```
train_ids <- sample(unique_ids, size = length(unique_ids) * 0.8)
```

```
# split the dataset into training and testing sets
```

```
train_set <- full_data[full_data$id %in% train_ids, ]
```

```
test_set <- full_data[!full_data$id %in% train_ids, ]
```

```
condensed_data <- train_set %>%
```

```
  group_by(id) %>%
```

```
  mutate(
```

```
    text_change_length = nchar(as.character(text_change)),
```

```
    time_interval = c(NA, diff(down_time)), # Time interval between events
```

```
    cursor_movement = c(NA, diff(cursor_position)) # Movement of cursor
```

```
  ) %>%
```

```
  summarise(
```

```

total_events = n(),
total_input_events = sum(activity == "Input"),
total_nonproduction_events = sum(activity == "Nonproduction"),
total_remove_cut_events = sum(activity == "Remove/Cut"),
total_paste_events = sum(activity == "Paste"),
total_replace_events = sum(activity == "Replace"),
avg_action_time = mean(action_time),
max_action_time = max(action_time),
min_action_time = min(action_time),
sd_action_time = sd(action_time),
final_word_count = max(word_count),
avg_text_change_length = mean(text_change_length, na.rm = TRUE),
total_text_added = sum(text_change_length[activity %in% c("Input", "Paste")], na.rm = TRUE),
total_text_removed = sum(text_change_length[activity %in% c("Remove/Cut", "Replace")], na.rm = TRUE),
avg_time_interval = mean(time_interval, na.rm = TRUE),
max_cursor_movement = max(cursor_movement, na.rm = TRUE),
avg_cursor_movement = mean(cursor_movement, na.rm = TRUE),
sd_cursor_movement = sd(cursor_movement, na.rm = TRUE)
) %>%
left_join(train_scores, by = "id")

head(condensed_data)

```

```

## # A tibble: 6 x 20
##   id          total_events total_input_events total_nonproduction_events
##   <chr>          <int>          <int>          <int>
## 1 001519c8      2557            2010            120
## 2 0042269b      4136            3515            175
## 3 0059420b      1556            1304             99
## 4 0075873a      2531            1942             72
## 5 0093f095      1765            1583             34
## 6 009e23ab      2353            1975            155
## # i 16 more variables: total_remove_cut_events <int>, total_paste_events <int>,
## #   total_replace_events <int>, avg_action_time <dbl>, max_action_time <int>,
## #   min_action_time <int>, sd_action_time <dbl>, final_word_count <int>,
## #   avg_text_change_length <dbl>, total_text_added <int>,
## #   total_text_removed <int>, avg_time_interval <dbl>,
## #   max_cursor_movement <int>, avg_cursor_movement <dbl>,
## #   sd_cursor_movement <dbl>, score <dbl>

```

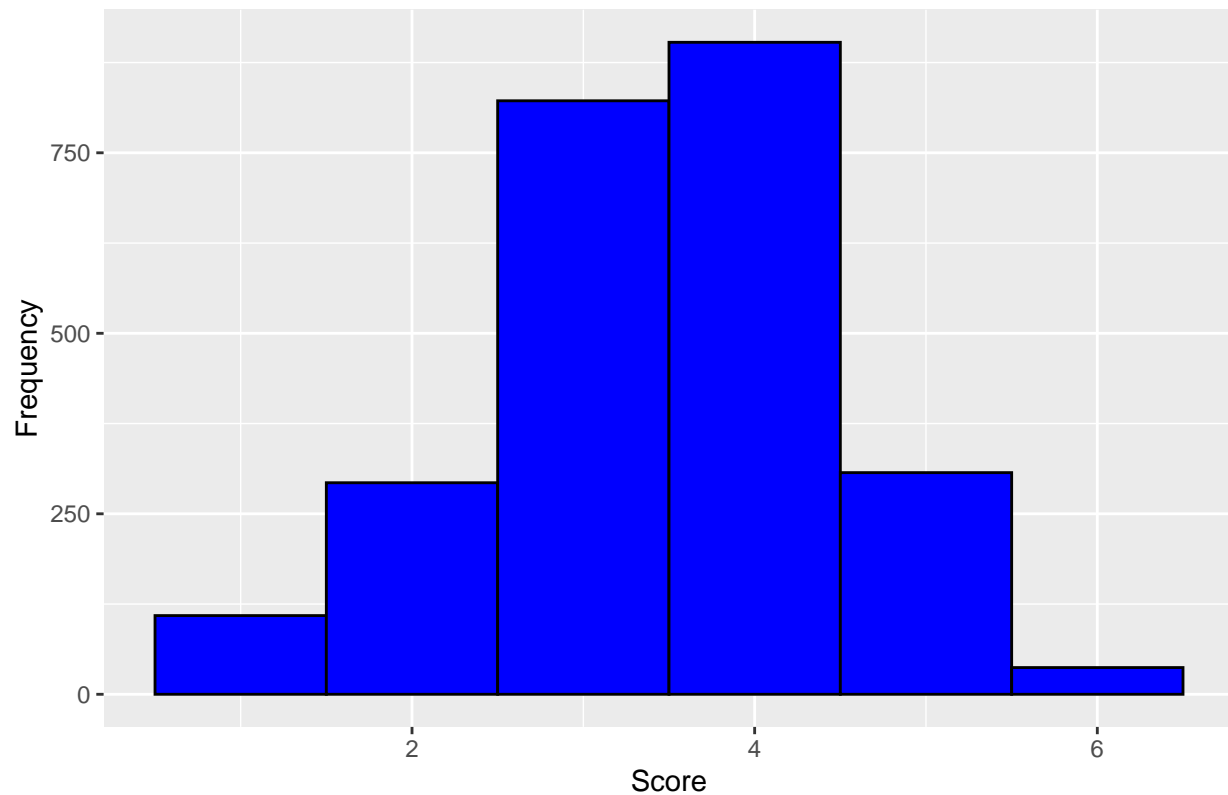
Histogram of Essay Scores

```

ggplot(train_scores, aes(x = score)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  ggtitle("Distribution of Essay Scores") +
  xlab("Score") +
  ylab("Frequency")

```

Distribution of Essay Scores



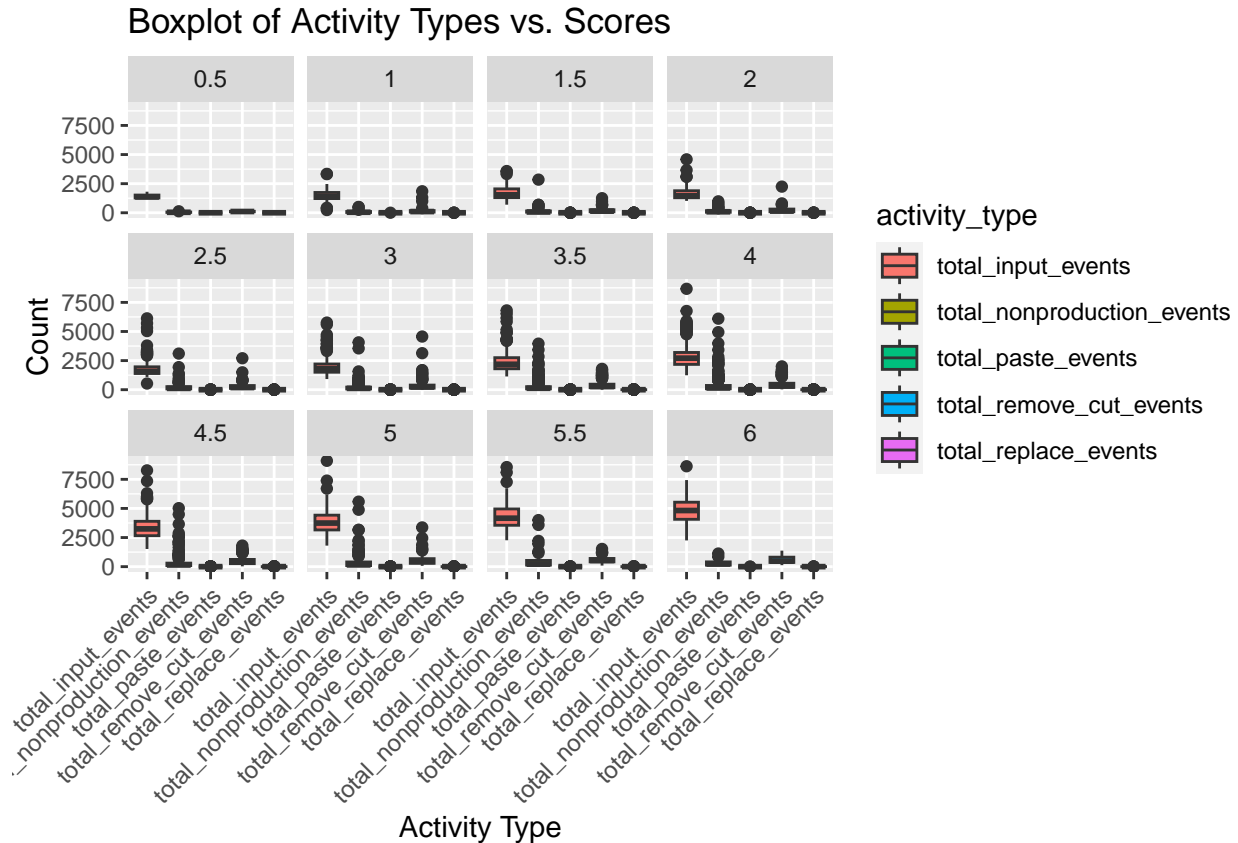
From the plot, we see the distribution is left-skewed, showing that very few essays are rated as either exceptionally poor or excellent.

Boxplots for Each Activity Type vs. Scores

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
condensed_data %>%
  gather(key = "activity_type", value = "count",
    total_input_events, total_nonproduction_events,
    total_remove_cut_events, total_paste_events,
    total_replace_events) %>%
  ggplot(aes(x = activity_type, y = count, fill = activity_type)) +
  geom_boxplot() +
  facet_wrap(~ score) +
  ggtitle("Boxplot of Activity Types vs. Scores") +
  xlab("Activity Type") +
  ylab("Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



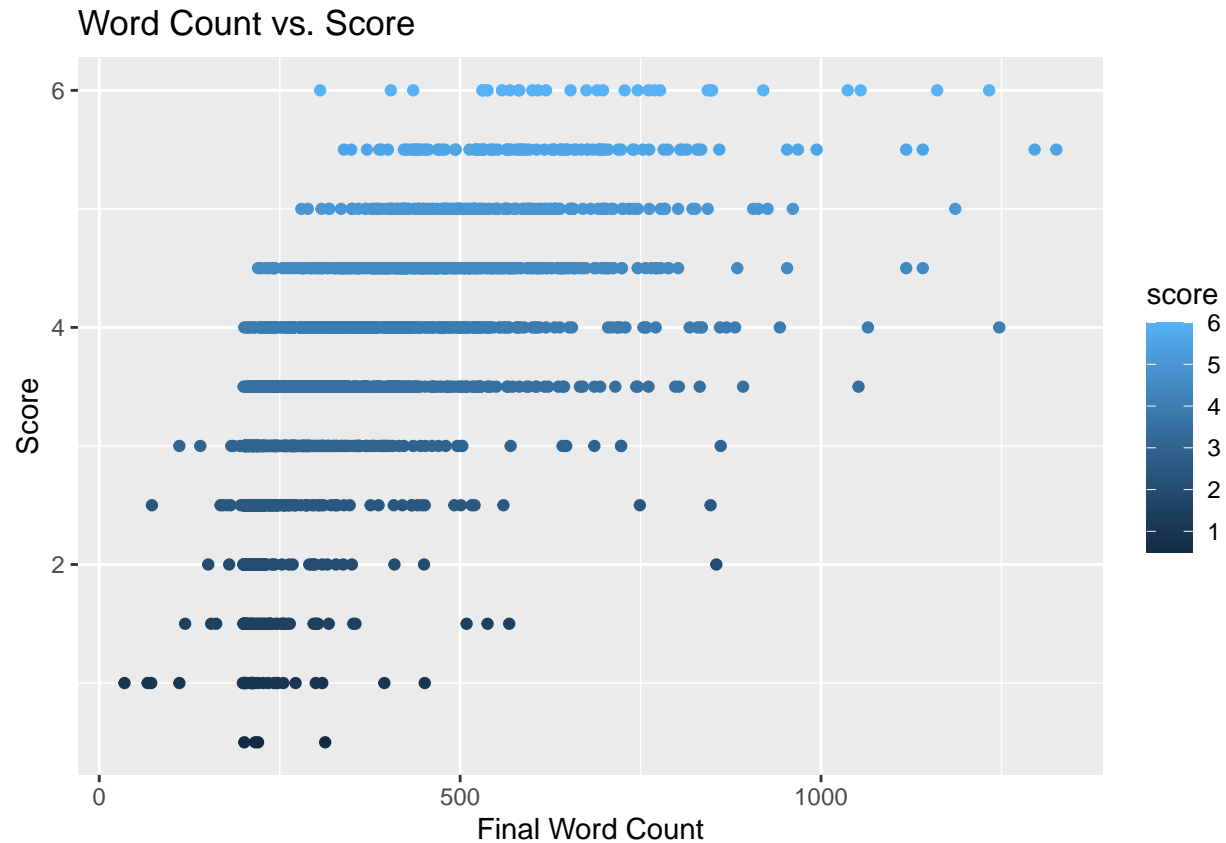
The median counts for most activities seem to increase with the scores up to a certain point, suggesting a correlation between the amount of activity and higher scores.

Some activities, like “total_input_events” and “total_replace_events,” have wide variations at higher scores, indicating that these activities are more variable for essays with higher scores.

“Total_remove_cut_events” and “total_nonproduction_events” appear to have a more stable count across different scores, suggesting these activities may not be as strongly correlated with the scores as others.

Scatter Plot of Word Count vs. Score: if there’s a relationship between the final word count of the essay and its score.

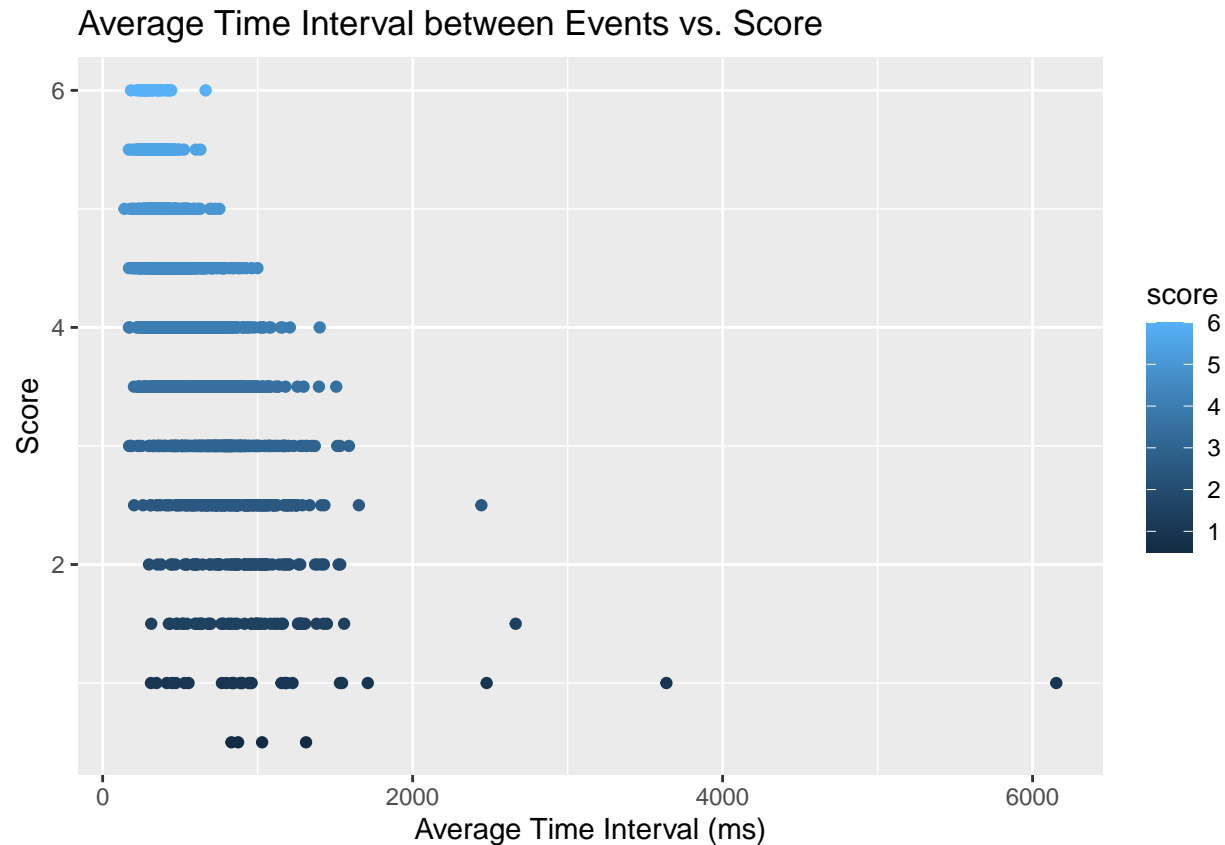
```
ggplot(condensed_data, aes(x = final_word_count, y = score)) +
  geom_point(aes(color = score)) +
  ggtitle("Word Count vs. Score") +
  xlab("Final Word Count") +
  ylab("Score")
```



There is a potential correlation between longer essays and higher scores.

Time Interval vs. Score: understand if there's any pattern in the average time intervals between events for different scores.

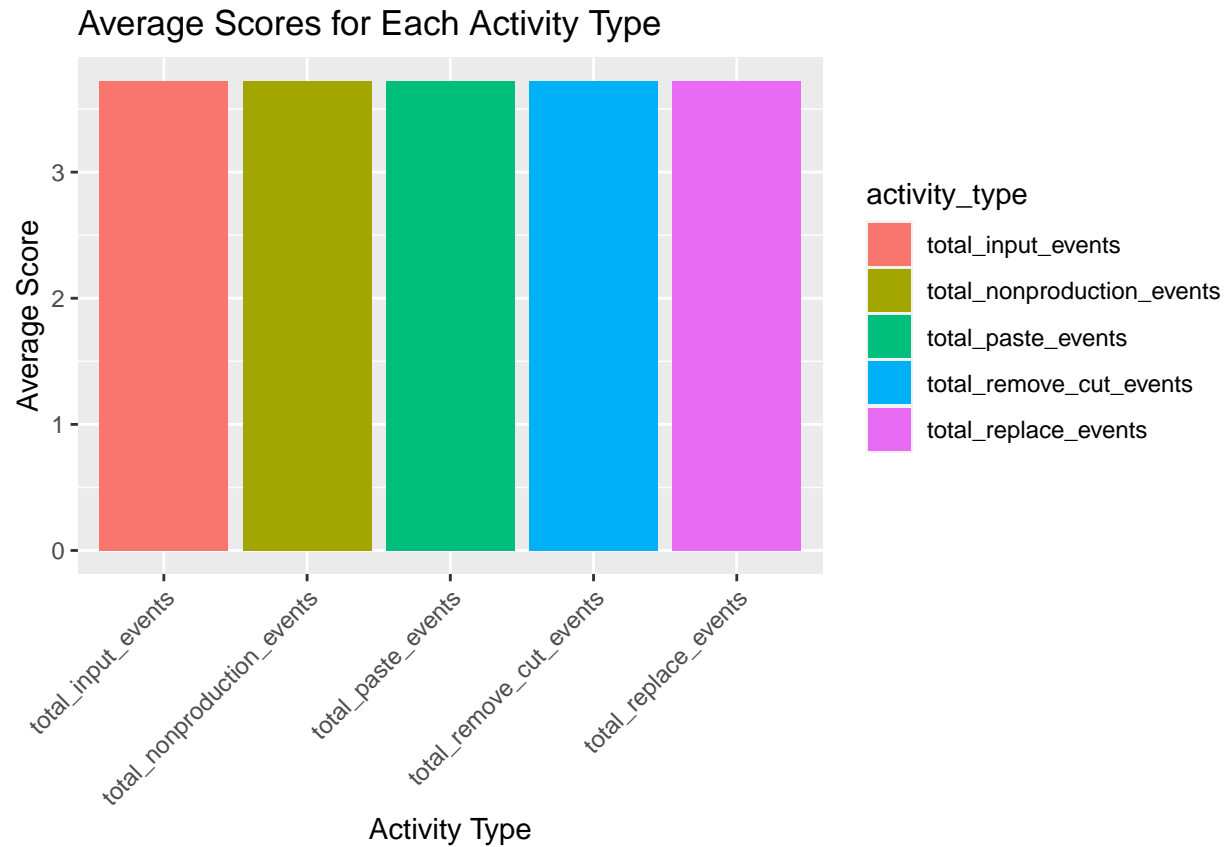
```
ggplot(condensed_data, aes(x = avg_time_interval, y = score)) +
  geom_point(aes(color = score)) +
  ggtitle("Average Time Interval between Events vs. Score") +
  xlab("Average Time Interval (ms)") +
  ylab("Score")
```



The scatter plot may suggest that shorter average time intervals between events are associated with higher scores, but this isn't a super strong correlation

Bar Plot of Average Scores for Each Activity: how the average essay score varies with different levels of activities like input, paste, remove/cut, etc.

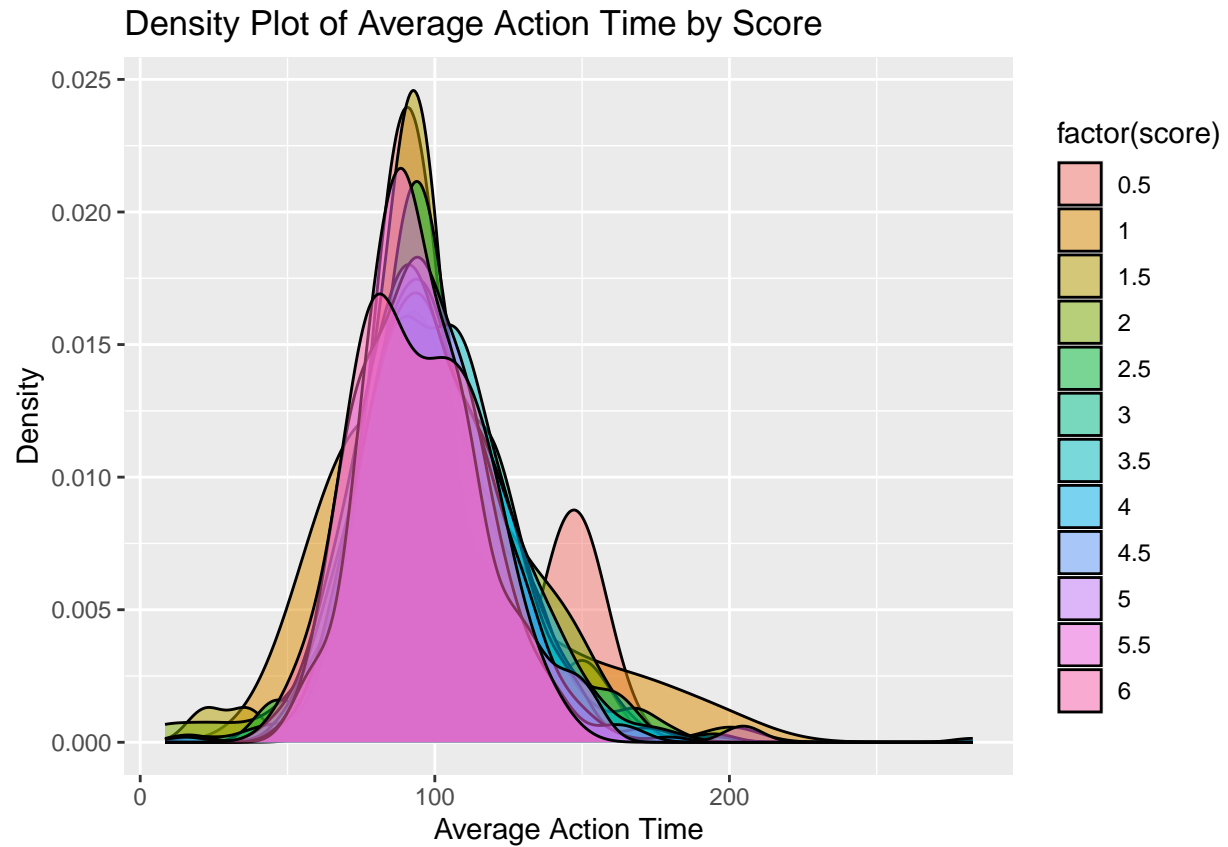
```
condensed_data %>%
  gather(key = "activity_type", value = "count",
    total_input_events, total_nonproduction_events,
    total_remove_cut_events, total_paste_events,
    total_replace_events) %>%
  group_by(activity_type) %>%
  summarise(average_score = mean(score)) %>%
  ggplot(aes(x = activity_type, y = average_score, fill = activity_type)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Average Scores for Each Activity Type") +
  xlab("Activity Type") +
  ylab("Average Score")
```



Not really useful

Density Plot of Action Time: how different durations of actions relate to essay scores.

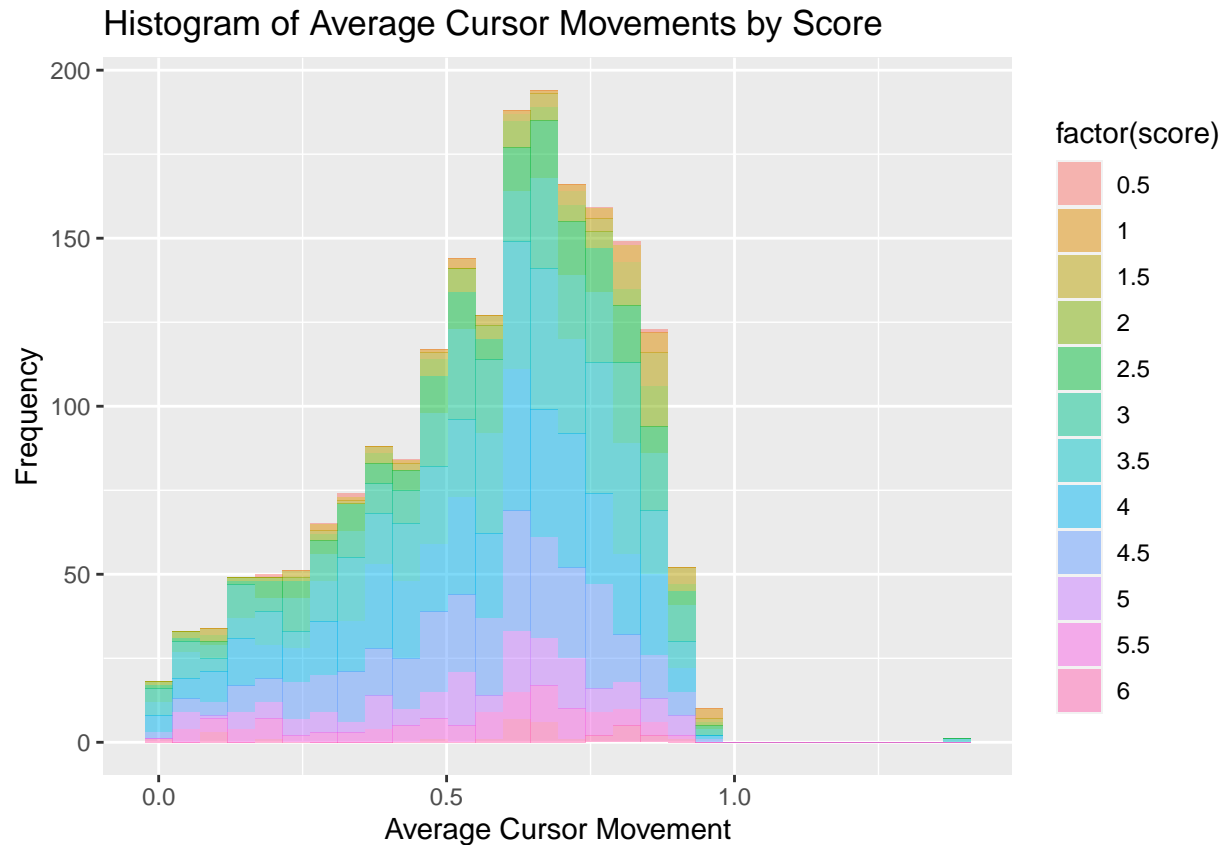
```
ggplot(condensed_data, aes(x = avg_action_time, fill = factor(score))) +
  geom_density(alpha = 0.5) +
  ggtitle("Density Plot of Average Action Time by Score") +
  xlab("Average Action Time") +
  ylab("Density")
```



There seems to be subtle indication that the highest scoring categories tend to have slightly quicker average action times.

Histogram of Cursor Movements: distribution of cursor movements for different scores.

```
ggplot(condensed_data, aes(x = avg_cursor_movement, fill = factor(score))) +
  geom_histogram(bins = 30, alpha = 0.5) +
  ggtitle("Histogram of Average Cursor Movements by Score") +
  xlab("Average Cursor Movement") +
  ylab("Frequency")
```

The visual suggests that there may be an optimal range of average cursor movement that is associated with higher scores, while both very low and very high cursor movements are less frequent among higher-scoring activities.

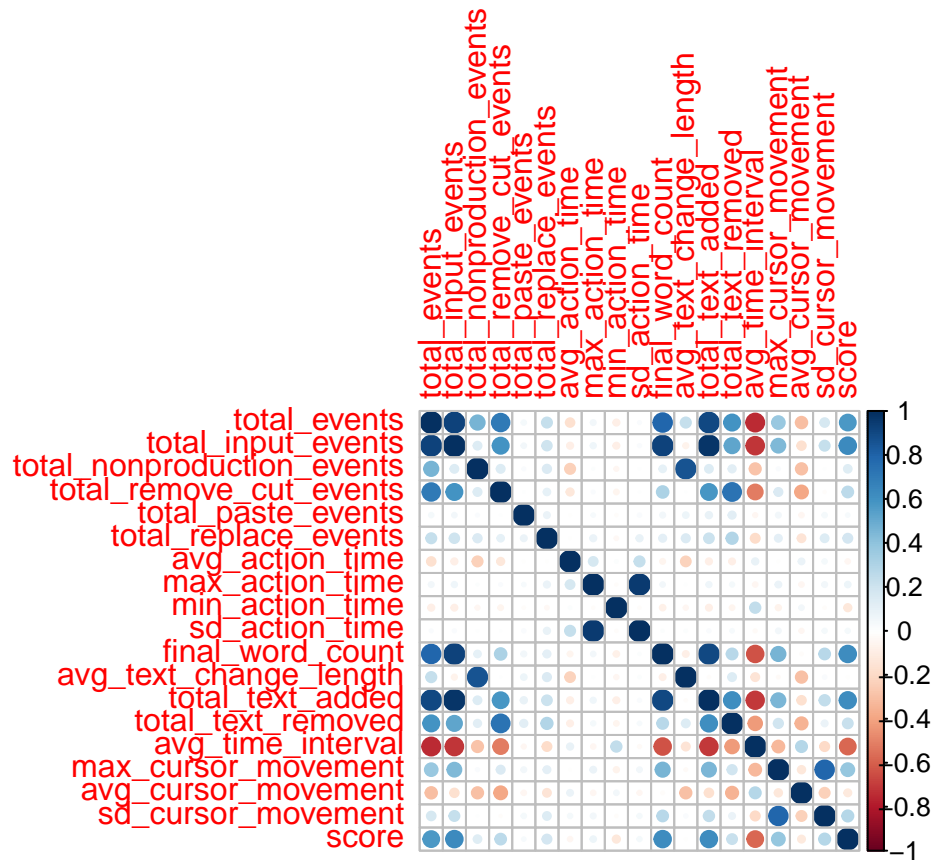
```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
library(dplyr)
```

```
cor_matrix <- cor(condensed_data[, -1], use = "complete.obs")  
corrplot(cor_matrix, method = "circle")
```



```
print(cor_matrix)
```

```
##               total_events total_input_events
## total_events      1.00000000      0.92515698
## total_input_events 0.92515698      1.00000000
## total_nonproduction_events 0.46865728      0.14455405
## total_remove_cut_events 0.70646500      0.59135802
## total_paste_events    0.04769900      0.05109270
## total_replace_events   0.23303201      0.20872189
## avg_action_time       -0.16919024     -0.08862954
## max_action_time        0.05745284      0.06551123
## min_action_time        -0.08813443     -0.08408190
## sd_action_time         0.02220013      0.02534474
## final_word_count       0.79265531      0.92108755
## avg_text_change_length 0.23728230     -0.06175314
## total_text_added       0.90494397      0.97721926
## total_text_removed     0.59253249      0.52551997
## avg_time_interval      -0.74249478     -0.71649445
## max_cursor_movement    0.37274389      0.44220085
## avg_cursor_movement    -0.30339665     -0.16566162
## sd_cursor_movement     0.17376583      0.23133759
## score                 0.57758220      0.63228574
##
##               total_nonproduction_events total_remove_cut_events
## total_events      0.468657276      0.706464995
## total_input_events 0.144554050      0.591358024
```

## total_nonproduction_events	1.000000000	0.147616130	
## total_remove_cut_events	0.147616130	1.000000000	
## total_paste_events	0.022366589	0.004608452	
## total_replace_events	0.158350647	0.102708098	
## avg_action_time	-0.233878711	-0.121043660	
## max_action_time	0.010699878	0.021136081	
## min_action_time	-0.040516252	-0.054372546	
## sd_action_time	0.006829122	0.003196937	
## final_word_count	0.105246743	0.310041819	
## avg_text_change_length	0.861882386	0.001723802	
## total_text_added	0.141194364	0.581690194	
## total_text_removed	0.121599224	0.735719206	
## avg_time_interval	-0.288712507	-0.511539927	
## max_cursor_movement	0.025488358	0.150062763	
## avg_cursor_movement	-0.293941669	-0.384349355	
## sd_cursor_movement	0.006545704	-0.011380875	
## score	0.136664277	0.269310885	
##	total_paste_events	total_replace_events	
## total_events	0.047699002	0.23303201	
## total_input_events	0.051092696	0.20872189	
## total_nonproduction_events	0.022366589	0.15835065	
## total_remove_cut_events	0.004608452	0.10270810	
## total_paste_events	1.000000000	0.08340563	
## total_replace_events	0.083405631	1.00000000	
## avg_action_time	-0.008007783	-0.00665607	
## max_action_time	0.030957439	0.08780628	
## min_action_time	-0.014417073	-0.03198449	
## sd_action_time	0.028602192	0.09191211	
## final_word_count	0.059738108	0.11692914	
## avg_text_change_length	0.074990648	0.15233233	
## total_text_added	0.094609823	0.21887041	
## total_text_removed	0.120087274	0.29326379	
## avg_time_interval	-0.041933344	-0.18389300	
## max_cursor_movement	0.058003824	0.11266638	
## avg_cursor_movement	-0.049423652	-0.14245492	
## sd_cursor_movement	0.049734938	0.14205962	
## score	0.031493171	0.18695037	
##	avg_action_time	max_action_time	min_action_time
## total_events	-0.169190242	0.057452843	-0.08813443
## total_input_events	-0.088629538	0.065511235	-0.08408190
## total_nonproduction_events	-0.233878711	0.010699878	-0.04051625
## total_remove_cut_events	-0.121043660	0.021136081	-0.05437255
## total_paste_events	-0.008007783	0.030957439	-0.01441707
## total_replace_events	-0.006656070	0.087806281	-0.03198449
## avg_action_time	1.000000000	0.179715840	0.01867696
## max_action_time	0.179715840	1.000000000	-0.01243788
## min_action_time	0.018676955	-0.012437882	1.00000000
## sd_action_time	0.235106575	0.954786849	-0.01371775
## final_word_count	-0.056881836	0.050890046	-0.07373801
## avg_text_change_length	-0.222691399	0.007889247	-0.04466051
## total_text_added	-0.088727811	0.063852264	-0.08369687
## total_text_removed	-0.082847548	0.044122898	-0.05239333
## avg_time_interval	0.091356753	-0.049814151	0.24609979
## max_cursor_movement	-0.005543821	0.081059634	-0.06540996

##	avg_cursor_movement	0.080681179	-0.049420012	0.05107261
##	sd_cursor_movement	0.047730600	0.058541322	-0.01418386
##	score	-0.048652402	0.073462264	-0.12099424
##	sd_action_time	final_word_count		
##	total_events	0.022200133	0.79265531	
##	total_input_events	0.025344740	0.92108755	
##	total_nonproduction_events	0.006829122	0.10524674	
##	total_remove_cut_events	0.003196937	0.31004182	
##	total_paste_events	0.028602192	0.05973811	
##	total_replace_events	0.091912113	0.11692914	
##	avg_action_time	0.235106575	-0.05688184	
##	max_action_time	0.954786849	0.05089005	
##	min_action_time	-0.013717747	-0.07373801	
##	sd_action_time	1.000000000	0.01176275	
##	final_word_count	0.011762751	1.00000000	
##	avg_text_change_length	0.017181870	-0.08102893	
##	total_text_added	0.025898365	0.90383609	
##	total_text_removed	0.032221973	0.27477842	
##	avg_time_interval	-0.023731723	-0.63291527	
##	max_cursor_movement	0.055082316	0.46177885	
##	avg_cursor_movement	-0.042782172	-0.01908221	
##	sd_cursor_movement	0.044774566	0.27163354	
##	score	0.044992702	0.62764263	
##	avg_text_change_length	total_text_added		
##	total_events	0.237282297	0.90494397	
##	total_input_events	-0.061753139	0.97721926	
##	total_nonproduction_events	0.861882386	0.14119436	
##	total_remove_cut_events	0.001723802	0.58169019	
##	total_paste_events	0.074990648	0.09460982	
##	total_replace_events	0.152332329	0.21887041	
##	avg_action_time	-0.222691399	-0.08872781	
##	max_action_time	0.007889247	0.06385226	
##	min_action_time	-0.044660508	-0.08369687	
##	sd_action_time	0.017181870	0.02589837	
##	final_word_count	-0.081028930	0.90383609	
##	avg_text_change_length	1.000000000	-0.01423570	
##	total_text_added	-0.014235697	1.00000000	
##	total_text_removed	0.146229486	0.61585010	
##	avg_time_interval	-0.147077762	-0.70095664	
##	max_cursor_movement	-0.020079026	0.45256086	
##	avg_cursor_movement	-0.295870388	-0.16972536	
##	sd_cursor_movement	0.029487851	0.24887751	
##	score	-0.008874247	0.62038712	
##	total_text_removed	avg_time_interval		
##	total_events	0.59253249	-0.74249478	
##	total_input_events	0.52551997	-0.71649445	
##	total_nonproduction_events	0.12159922	-0.28871251	
##	total_remove_cut_events	0.73571921	-0.51153993	
##	total_paste_events	0.12008727	-0.04193334	
##	total_replace_events	0.29326379	-0.18389300	
##	avg_action_time	-0.08284755	0.09135675	
##	max_action_time	0.04412290	-0.04981415	
##	min_action_time	-0.05239333	0.24609979	
##	sd_action_time	0.03222197	-0.02373172	

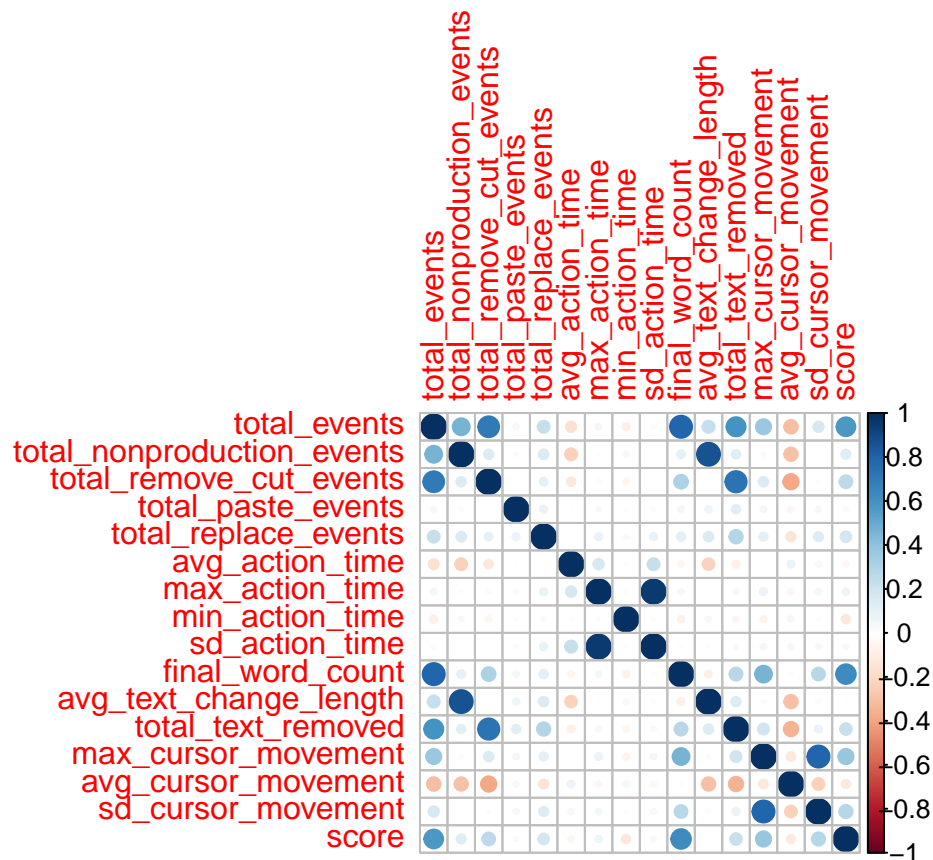
## final_word_count	0.27477842	-0.63291527
## avg_text_change_length	0.14622949	-0.14707776
## total_text_added	0.61585010	-0.70095664
## total_text_removed	1.00000000	-0.42366531
## avg_time_interval	-0.42366531	1.00000000
## max_cursor_movement	0.19698113	-0.32378438
## avg_cursor_movement	-0.34105852	0.28173119
## sd_cursor_movement	0.08366222	-0.19103058
## score	0.22500620	-0.57463613
##	max_cursor_movement	avg_cursor_movement
## total_events	0.372743891	-0.30339665
## total_input_events	0.442200854	-0.16566162
## total_nonproduction_events	0.025488358	-0.29394167
## total_remove_cut_events	0.150062763	-0.38434936
## total_paste_events	0.058003824	-0.04942365
## total_replace_events	0.112666382	-0.14245492
## avg_action_time	-0.005543821	0.08068118
## max_action_time	0.081059634	-0.04942001
## min_action_time	-0.065409959	0.05107261
## sd_action_time	0.055082316	-0.04278217
## final_word_count	0.461778847	-0.01908221
## avg_text_change_length	-0.020079026	-0.29587039
## total_text_added	0.452560865	-0.16972536
## total_text_removed	0.196981134	-0.34105852
## avg_time_interval	-0.323784376	0.28173119
## max_cursor_movement	1.000000000	-0.12760367
## avg_cursor_movement	-0.127603665	1.00000000
## sd_cursor_movement	0.791971463	-0.23021208
## score	0.380343012	-0.11422539
##	sd_cursor_movement	score
## total_events	0.173765829	0.577582200
## total_input_events	0.231337588	0.632285739
## total_nonproduction_events	0.006545704	0.136664277
## total_remove_cut_events	-0.011380875	0.269310885
## total_paste_events	0.049734938	0.031493171
## total_replace_events	0.142059624	0.186950370
## avg_action_time	0.047730600	-0.048652402
## max_action_time	0.058541322	0.073462264
## min_action_time	-0.014183858	-0.120994238
## sd_action_time	0.044774566	0.044992702
## final_word_count	0.271633536	0.627642634
## avg_text_change_length	0.029487851	-0.008874247
## total_text_added	0.248877510	0.620387125
## total_text_removed	0.083662218	0.225006203
## avg_time_interval	-0.191030576	-0.574636127
## max_cursor_movement	0.791971463	0.380343012
## avg_cursor_movement	-0.230212077	-0.114225393
## sd_cursor_movement	1.000000000	0.286148398
## score	0.286148398	1.000000000

There's a decent amount of collinearity. We will remove some variables that are kind of redundant.

```
condensed_data$total_text_added <- NULL
condensed_data$total_input_events <- NULL
condensed_data$avg_time_interval<- NULL
head(condensed_data)
```

```
## # A tibble: 6 x 17
##   id          total_events total_nonproduction_events total_remove_cut_events
##   <chr>          <int>          <int>          <int>
## 1 001519c8        2557            120            417
## 2 0042269b        4136            175            439
## 3 0059420b        1556             99            151
## 4 0075873a        2531             72            517
## 5 0093f095        1765             34            148
## 6 009e23ab        2353            155            222
## # i 13 more variables: total_paste_events <int>, total_replace_events <int>,
## #   avg_action_time <dbl>, max_action_time <int>, min_action_time <int>,
## #   sd_action_time <dbl>, final_word_count <int>, avg_text_change_length <dbl>,
## #   total_text_removed <int>, max_cursor_movement <int>,
## #   avg_cursor_movement <dbl>, sd_cursor_movement <dbl>, score <dbl>
```

```
cor_matrix <- cor(condensed_data[, -1], use = "complete.obs")
corrplot(cor_matrix, method = "circle")
```



```
print(cor_matrix)
```

```
##                total_events total_nonproduction_events
## total_events      1.00000000      0.468657276
## total_nonproduction_events 0.46865728      1.000000000
## total_remove_cut_events    0.70646500      0.147616130
## total_paste_events        0.04769900      0.022366589
## total_replace_events      0.23303201      0.158350647
## avg_action_time          -0.16919024     -0.233878711
## max_action_time          0.05745284      0.010699878
## min_action_time          -0.08813443     -0.040516252
## sd_action_time           0.02220013      0.006829122
## final_word_count         0.79265531      0.105246743
## avg_text_change_length    0.23728230      0.861882386
## total_text_removed        0.59253249      0.121599224
## max_cursor_movement       0.37274389      0.025488358
## avg_cursor_movement       -0.30339665     -0.293941669
## sd_cursor_movement        0.17376583      0.006545704
## score                  0.57758220      0.136664277
##                total_remove_cut_events total_paste_events
## total_events          0.706464995      0.047699002
## total_nonproduction_events 0.147616130      0.022366589
## total_remove_cut_events 1.000000000      0.004608452
## total_paste_events     0.004608452      1.000000000
## total_replace_events    0.102708098      0.083405631
## avg_action_time         -0.121043660     -0.008007783
## max_action_time         0.021136081      0.030957439
## min_action_time         -0.054372546     -0.014417073
## sd_action_time          0.003196937      0.028602192
## final_word_count        0.310041819      0.059738108
## avg_text_change_length  0.001723802      0.074990648
## total_text_removed       0.735719206      0.120087274
## max_cursor_movement      0.150062763      0.058003824
## avg_cursor_movement      -0.384349355     -0.049423652
## sd_cursor_movement       -0.011380875      0.049734938
## score                   0.269310885      0.031493171
##                total_replace_events avg_action_time max_action_time
## total_events          0.23303201     -0.169190242      0.057452843
## total_nonproduction_events 0.15835065     -0.233878711      0.010699878
## total_remove_cut_events 0.10270810     -0.121043660      0.021136081
## total_paste_events     0.08340563     -0.008007783      0.030957439
## total_replace_events    1.00000000     -0.006656070      0.087806281
## avg_action_time        -0.00665607      1.000000000      0.179715840
## max_action_time         0.08780628      0.179715840      1.000000000
## min_action_time        -0.03198449      0.018676955     -0.012437882
## sd_action_time          0.09191211      0.235106575      0.954786849
## final_word_count        0.11692914     -0.056881836      0.050890046
## avg_text_change_length  0.15233233     -0.222691399      0.007889247
## total_text_removed       0.29326379     -0.082847548      0.044122898
## max_cursor_movement      0.11266638     -0.005543821      0.081059634
## avg_cursor_movement     -0.14245492      0.080681179     -0.049420012
## sd_cursor_movement       0.14205962      0.047730600      0.058541322
## score                   0.18695037     -0.048652402      0.073462264
```

##	min_action_time	sd_action_time	final_word_count
## total_events	-0.08813443	0.022200133	0.79265531
## total_nonproduction_events	-0.04051625	0.006829122	0.10524674
## total_remove_cut_events	-0.05437255	0.003196937	0.31004182
## total_paste_events	-0.01441707	0.028602192	0.05973811
## total_replace_events	-0.03198449	0.091912113	0.11692914
## avg_action_time	0.01867696	0.235106575	-0.05688184
## max_action_time	-0.01243788	0.954786849	0.05089005
## min_action_time	1.00000000	-0.013717747	-0.07373801
## sd_action_time	-0.01371775	1.000000000	0.01176275
## final_word_count	-0.07373801	0.011762751	1.00000000
## avg_text_change_length	-0.04466051	0.017181870	-0.08102893
## total_text_removed	-0.05239333	0.032221973	0.27477842
## max_cursor_movement	-0.06540996	0.055082316	0.46177885
## avg_cursor_movement	0.05107261	-0.042782172	-0.01908221
## sd_cursor_movement	-0.01418386	0.044774566	0.27163354
## score	-0.12099424	0.044992702	0.62764263
##	avg_text_change_length	total_text_removed	
## total_events	0.237282297	0.59253249	
## total_nonproduction_events	0.861882386	0.12159922	
## total_remove_cut_events	0.001723802	0.73571921	
## total_paste_events	0.074990648	0.12008727	
## total_replace_events	0.152332329	0.29326379	
## avg_action_time	-0.222691399	-0.08284755	
## max_action_time	0.007889247	0.04412290	
## min_action_time	-0.044660508	-0.05239333	
## sd_action_time	0.017181870	0.03222197	
## final_word_count	-0.081028930	0.27477842	
## avg_text_change_length	1.000000000	0.14622949	
## total_text_removed	0.146229486	1.00000000	
## max_cursor_movement	-0.020079026	0.19698113	
## avg_cursor_movement	-0.295870388	-0.34105852	
## sd_cursor_movement	0.029487851	0.08366222	
## score	-0.008874247	0.22500620	
##	max_cursor_movement	avg_cursor_movement	
## total_events	0.372743891	-0.30339665	
## total_nonproduction_events	0.025488358	-0.29394167	
## total_remove_cut_events	0.150062763	-0.38434936	
## total_paste_events	0.058003824	-0.04942365	
## total_replace_events	0.112666382	-0.14245492	
## avg_action_time	-0.005543821	0.08068118	
## max_action_time	0.081059634	-0.04942001	
## min_action_time	-0.065409959	0.05107261	
## sd_action_time	0.055082316	-0.04278217	
## final_word_count	0.461778847	-0.01908221	
## avg_text_change_length	-0.020079026	-0.29587039	
## total_text_removed	0.196981134	-0.34105852	
## max_cursor_movement	1.000000000	-0.12760367	
## avg_cursor_movement	-0.127603665	1.00000000	
## sd_cursor_movement	0.791971463	-0.23021208	
## score	0.380343012	-0.11422539	
##	sd_cursor_movement	score	
## total_events	0.173765829	0.577582200	
## total_nonproduction_events	0.006545704	0.136664277	

## total_remove_cut_events	-0.011380875	0.269310885
## total_paste_events	0.049734938	0.031493171
## total_replace_events	0.142059624	0.186950370
## avg_action_time	0.047730600	-0.048652402
## max_action_time	0.058541322	0.073462264
## min_action_time	-0.014183858	-0.120994238
## sd_action_time	0.044774566	0.044992702
## final_word_count	0.271633536	0.627642634
## avg_text_change_length	0.029487851	-0.008874247
## total_text_removed	0.083662218	0.225006203
## max_cursor_movement	0.791971463	0.380343012
## avg_cursor_movement	-0.230212077	-0.114225393
## sd_cursor_movement	1.000000000	0.286148398
## score	0.286148398	1.000000000