Whatsapp chat analysis

Usama Bin Haider

12/10/2021

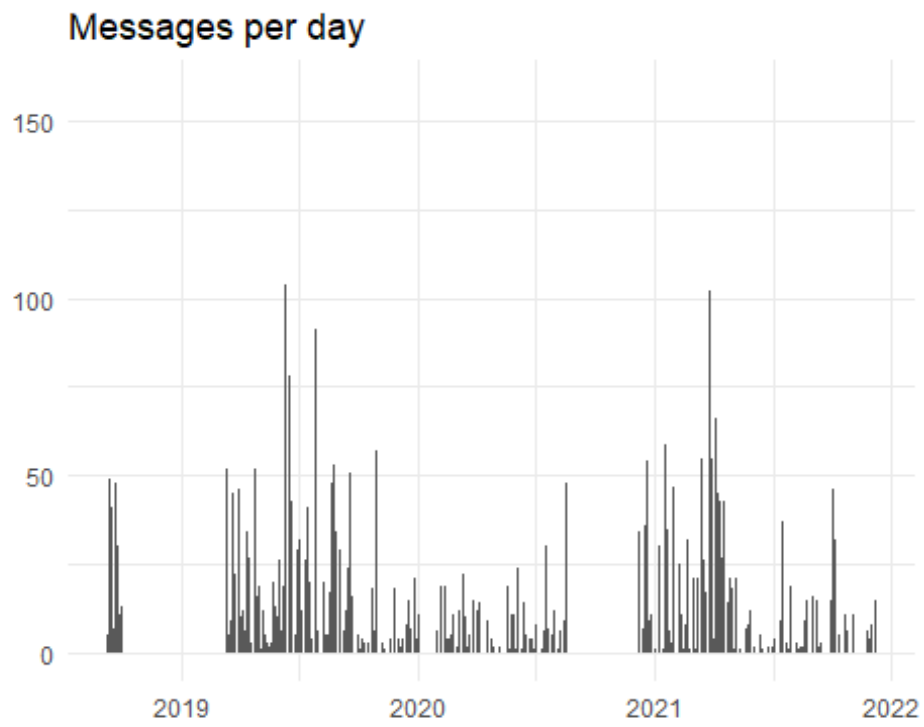
R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

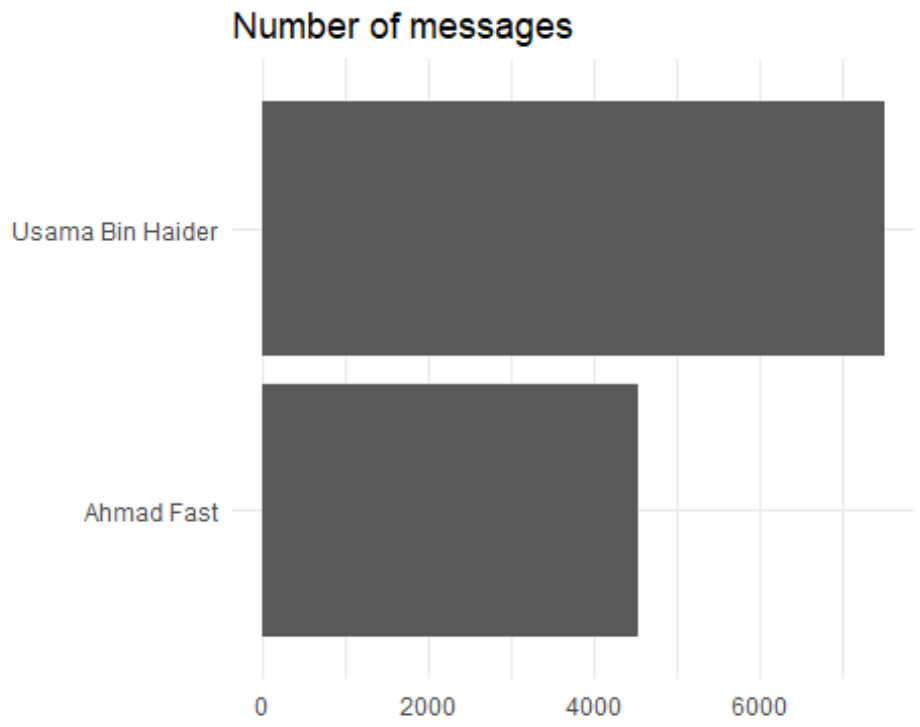
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
history <- system.file("extdata", "sample.txt", package = "rwhatsapp")
library("rwhatsapp")

## Warning: package 'rwhatsapp' was built under R version 4.1.2

library("dplyr")

## Warning: package 'dplyr' was built under R version 4.1.2

##
## 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

chat <- rwa_read("WhatsApp Chat with Ahmad Fast.txt") %>%
  filter(!is.na(author)) # remove messages without author
library("ggplot2"); theme_set(theme_minimal())
library("lubridate")

## Warning: package 'lubridate' was built under R version 4.1.2

##
## Attaching package: 'lubridate'

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

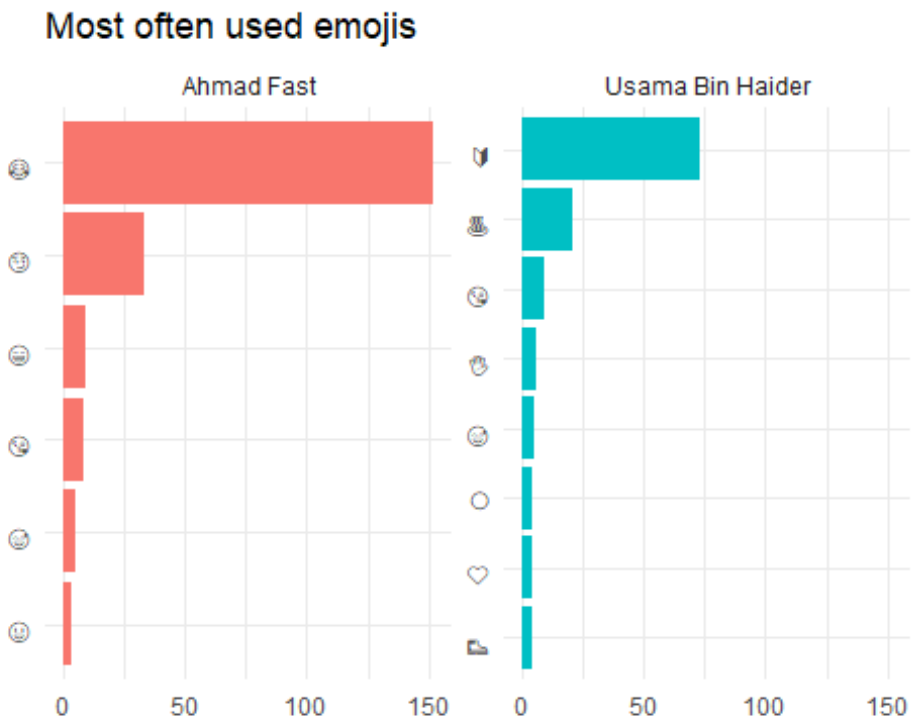
```
chat %>%
  mutate(day = date(time)) %>%
  count(day) %>%
  ggplot(aes(x = day, y = n)) +
  geom_bar(stat = "identity") +
  ylab("") + xlab("") +
  ggtitle("Messages per day")
```



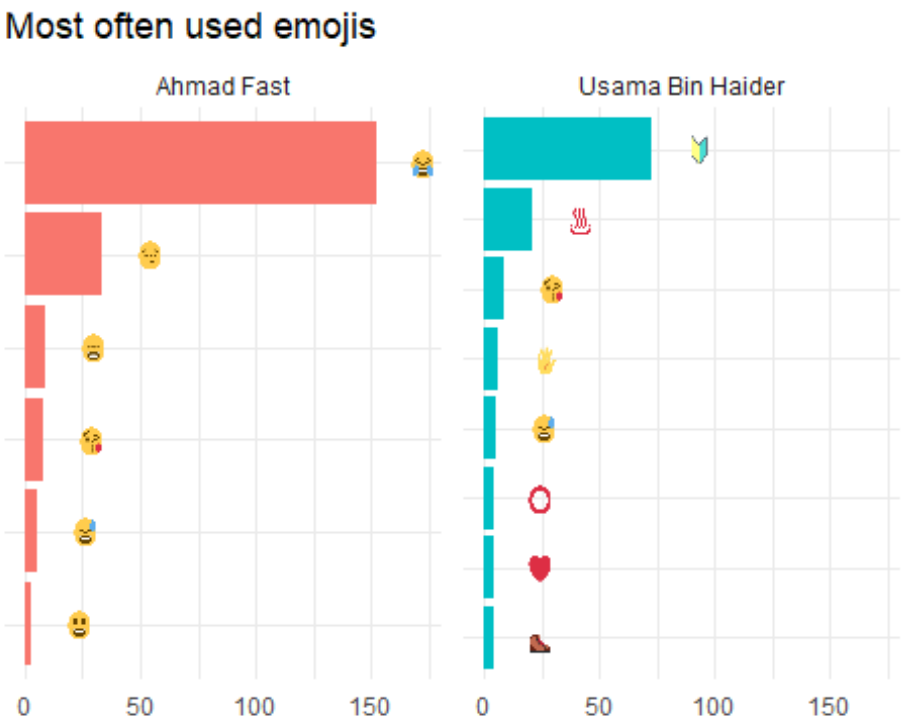

```
chat %>%
  mutate(day = date(time)) %>%
  count(author) %>%
  ggplot(aes(x = reorder(author, n), y = n)) +
  geom_bar(stat = "identity") +
  ylab("") + xlab("") +
  coord_flip() +
  ggtitle("Number of messages")
```



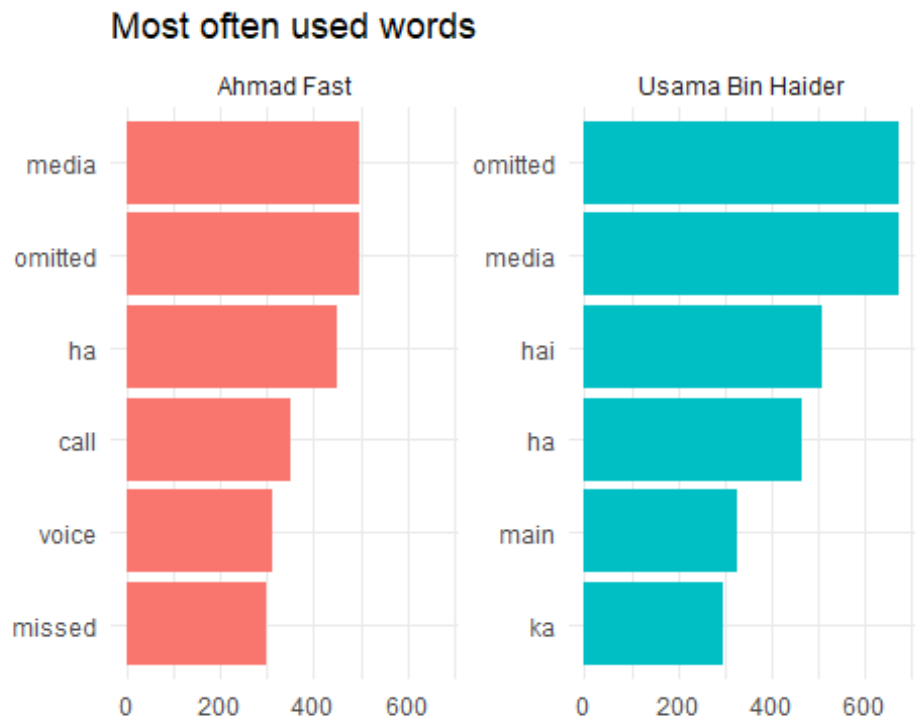

```
library("tidyr")

## Warning: package 'tidyr' was built under R version 4.1.2

chat %>%
  unnest(emoji) %>%
  count(author, emoji, sort = TRUE) %>%
  group_by(author) %>%
  top_n(n = 6, n) %>%
  ggplot(aes(x = reorder(emoji, n), y = n, fill = author)) +
  geom_col(show.legend = FALSE) +
  ylab("") +
  xlab("") +
  coord_flip() +
  facet_wrap(~author, ncol = 2, scales = "free_y")  +
  ggtitle("Most often used emojis")
```




```
library("ggimage")

## Warning: package 'ggimage' was built under R version 4.1.2

emoji_data <- rwhatsapp::emojis %>% # data built into package
  mutate(hex_runes1 = gsub("\\s[[:alnum:]]+", "", hex_runes)) %>% # ignore
combined emojis
  mutate(emoji_url = paste0("https://abs.twimg.com/emoji/v2/72x72/",
                            tolower(hex_runes1), ".png"))

chat %>%
  unnest(emoji) %>%
  count(author, emoji, sort = TRUE) %>%
  group_by(author) %>%
  top_n(n = 6, n) %>%
  left_join(emoji_data, by = "emoji") %>%
  ggplot(aes(x = reorder(emoji, n), y = n, fill = author)) +
  geom_col(show.legend = FALSE) +
  ylab("") +
  xlab("") +
  coord_flip() +
  geom_image(aes(y = n + 20, image = emoji_url)) +
  facet_wrap(~author, ncol = 2, scales = "free_y") +
  ggtitle("Most often used emojis") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank())
```

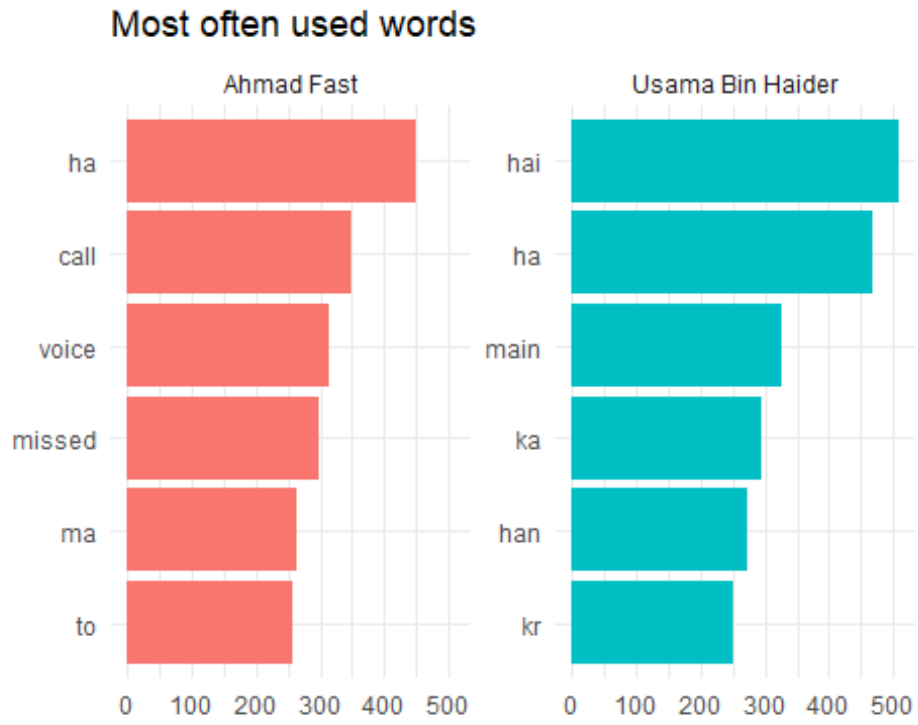



```
library("tidytext")

## Warning: package 'tidytext' was built under R version 4.1.2

chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  count(author, word, sort = TRUE) %>%
  group_by(author) %>%
  top_n(n = 6, n) %>%
  ggplot(aes(x = reorder_within(word, n, author), y = n, fill = author)) +
  geom_col(show.legend = FALSE) +
  ylab("") +
  xlab("") +
  coord_flip() +
  facet_wrap(~author, ncol = 2, scales = "free_y") +
  scale_x_reordered() +
  ggtitle("Most often used words")
```

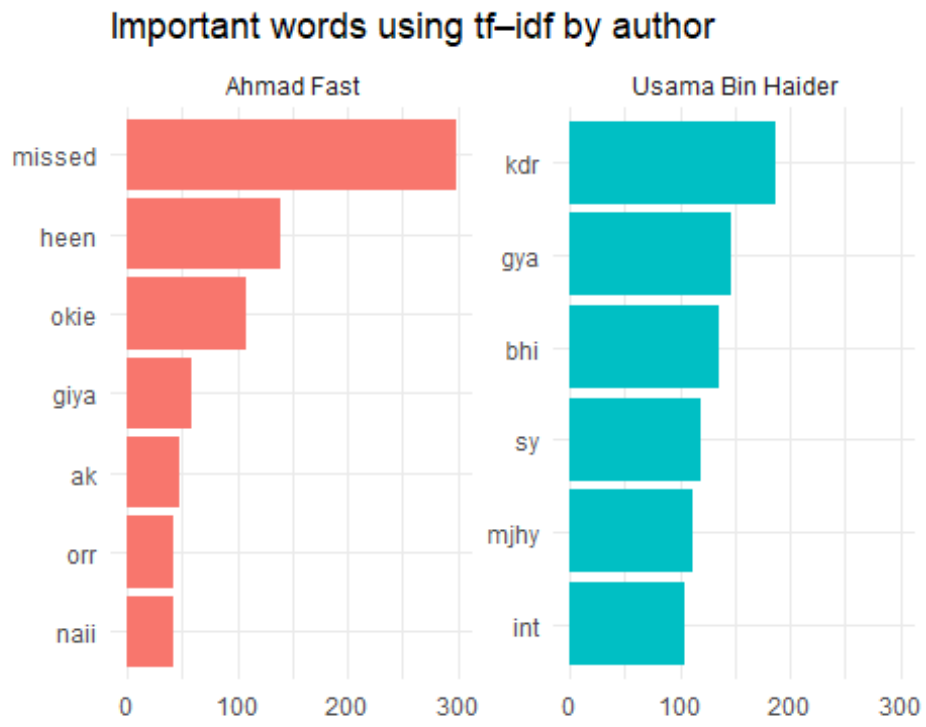



```
library("stopwords")

## Warning: package 'stopwords' was built under R version 4.1.2

to_remove <- c(stopwords(language = "de"),
               "media",
               "omitted",
               "ref",
               "dass",
               "schon",
               "mal",
               "android.s.wt")

chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(!word %in% to_remove) %>%
  count(author, word, sort = TRUE) %>%
  group_by(author) %>%
  top_n(n = 6, n) %>%
  ggplot(aes(x = reorder_within(word, n, author), y = n, fill = author)) +
  geom_col(show.legend = FALSE) +
  ylab("") +
  xlab("") +
  coord_flip() +
  facet_wrap(~author, ncol = 2, scales = "free_y") +
```

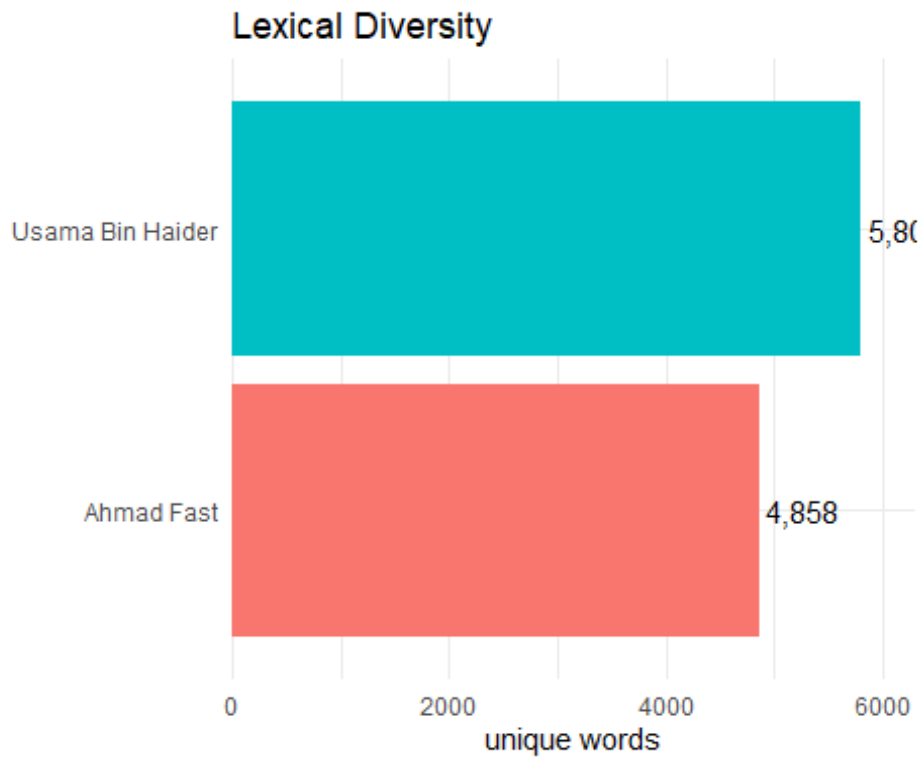
```
  scale_x_reordered() +
  ggtitle("Most often used words")
```



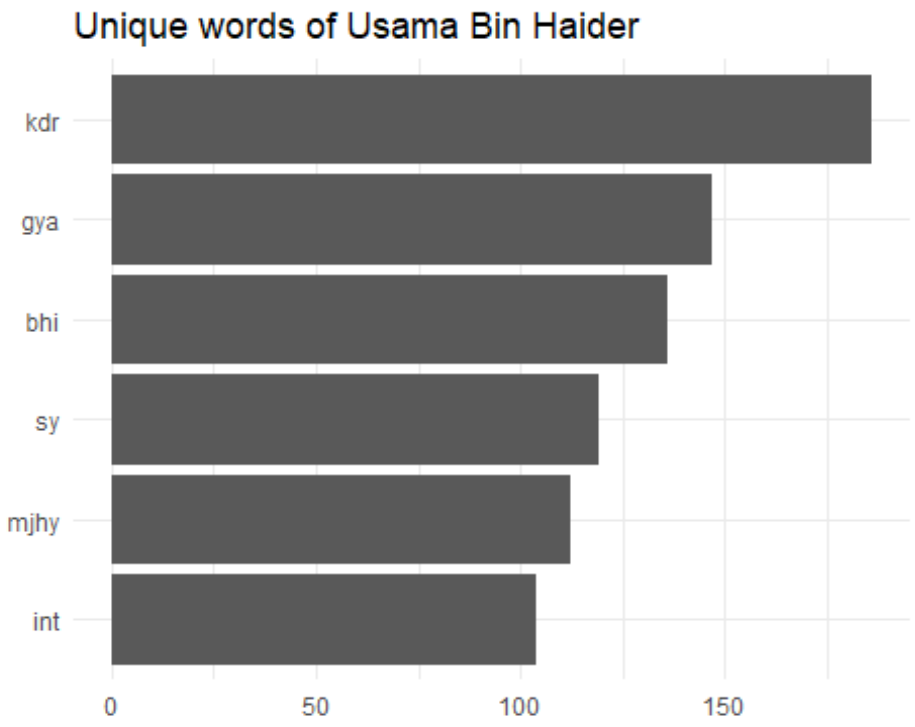

```
chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  select(word, author) %>%
  filter(!word %in% to_remove) %>%
  mutate(word = gsub(".com", "", word)) %>%
  mutate(word = gsub("^gag", "9gag", word)) %>%
  count(author, word, sort = TRUE) %>%
  bind_tf_idf(term = word, document = author, n = n) %>%
  filter(n > 10) %>%
  group_by(author) %>%
  top_n(n = 6, tf_idf) %>%
  ggplot(aes(x = reorder_within(word, n, author), y = n, fill = author)) +
  geom_col(show.legend = FALSE) +
  ylab("") +
  xlab("") +
  coord_flip() +
  facet_wrap(~author, ncol = 2, scales = "free_y") +
  scale_x_reordered() +
  ggtitle("Important words using tf-idf by author")
```



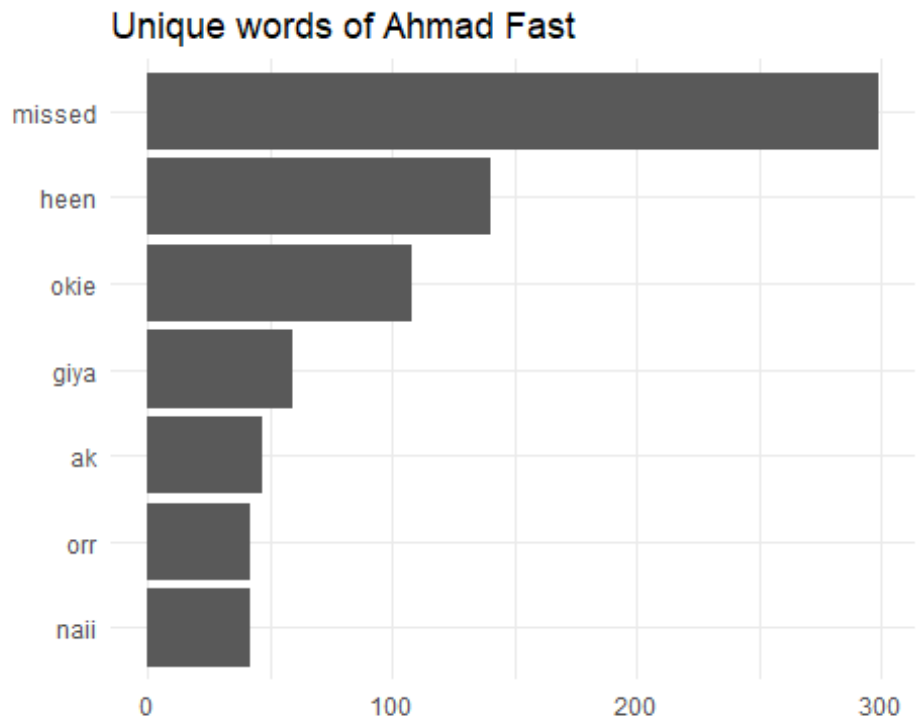

```
chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(!word %in% to_remove) %>%
  group_by(author) %>%
  summarise(lex_diversity = n_distinct(word)) %>%
  arrange(desc(lex_diversity)) %>%
  ggplot(aes(x = reorder(author, lex_diversity),
             y = lex_diversity,
             fill = author)) +
  geom_col(show.legend = FALSE) +
  scale_y_continuous(expand = (mult = c(0, 0, 0, 500))) +
  geom_text(aes(label = scales::comma(lex_diversity)), hjust = -0.1) +
  ylab("unique words") +
  xlab("") +
  ggtitle("Lexical Diversity") +
  coord_flip()
```




```r
o_words <- chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(author != "Usama Bin Haider") %>%
  count(word, sort = TRUE)

chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(author == "Usama Bin Haider") %>%
  count(word, sort = TRUE) %>%
  filter(!word %in% o_words$word) %>% # only select words nobody else uses
  top_n(n = 6, n) %>%
  ggplot(aes(x = reorder(word, n), y = n)) +
  geom_col(show.legend = FALSE) +
  ylab("") + xlab("") +
  coord_flip() +
  ggtitle("Unique words of Usama Bin Haider")
```



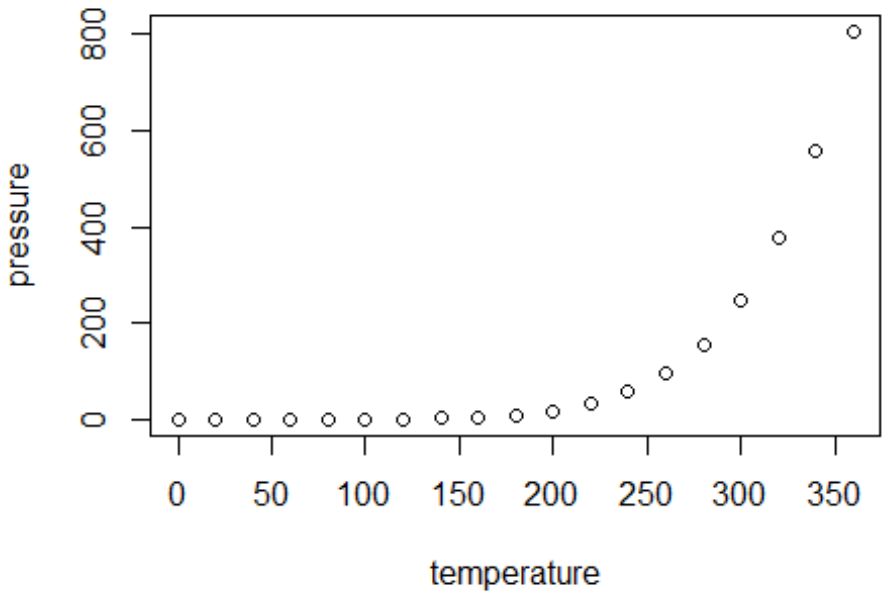

```
o_words <- chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(author != "Ahmad Fast") %>%
  count(word, sort = TRUE)

chat %>%
  unnest_tokens(input = text,
                output = word) %>%
  filter(author == "Ahmad Fast") %>%
  count(word, sort = TRUE) %>%
  filter(!word %in% o_words$word) %>% # only select words nobody else uses
  top_n(n = 6, n) %>%
  ggplot(aes(x = reorder(word, n), y = n)) +
  geom_col(show.legend = FALSE) +
  ylab("") + xlab("") +
  coord_flip() +
  ggtitle("Unique words of Ahmad Fast")
```




```
summary(cars)

##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.