Markdown_Week13_DataVisualization

Aymeric Collart

1. Prepare the environment

1.1 Load the libraries

```
library(quanteda)
```

```
## Package version: 4.3.1
## Unicode version: 14.0
## ICU version: 71.1
```

```
## Parallel computing: disabled
```

```
## See https://quanteda.io for tutorials and examples.
```

```
library(quanteda.textstats)
library(quanteda.textplots)
library(stringr)
library(ggplot2)
library(showtext)
```

```
## Loading required package: sysfonts
```

```
## Loading required package: showtextdb
```

```
library(scales)

#Sys.setlocale(category = "LC_ALL", locale = "cht")
# Use showtext_auto() to enable automatic font discovery and display Chinese characters
showtext_auto()
```

1.2 Load the files used for the analyses

```
load(file = "ArticleETToday_CorpusCourse_CLEAN.Rdata")
load(file = "ArticleETToday_KWIC_You.Rdata")
load(file = "ArticleETToday_Top100words.Rdata")
```

2. Visualizing KWIC

2.1 Visualizing overall frequency in the "post" context

2.1.1 Word cloud

```
## We need to tranform the tokenized data into a 'dfm' dataset
kwic_post_freq <- dfm(
  tokens(kwic_data$post,
         remove_punct = TRUE,
         remove_numbers = TRUE,
         remove_separators = TRUE))

kwic_post_freq_trim_WithDE <- dfm_trim(kwic_post_freq,
                                       min_termfreq = 100,
                                       verbose = TRUE)
```

```
## dfm_trim() changed from 21,355 features (65,657 documents) to 540 features (65,657 documents)
```

```
kwic_post_freq_trim_WithoutDE <- dfm_trim(kwic_post_freq,
                                       min_termfreq = 100,
                                       max_termfreq = 10000,
                                       verbose = TRUE)
```

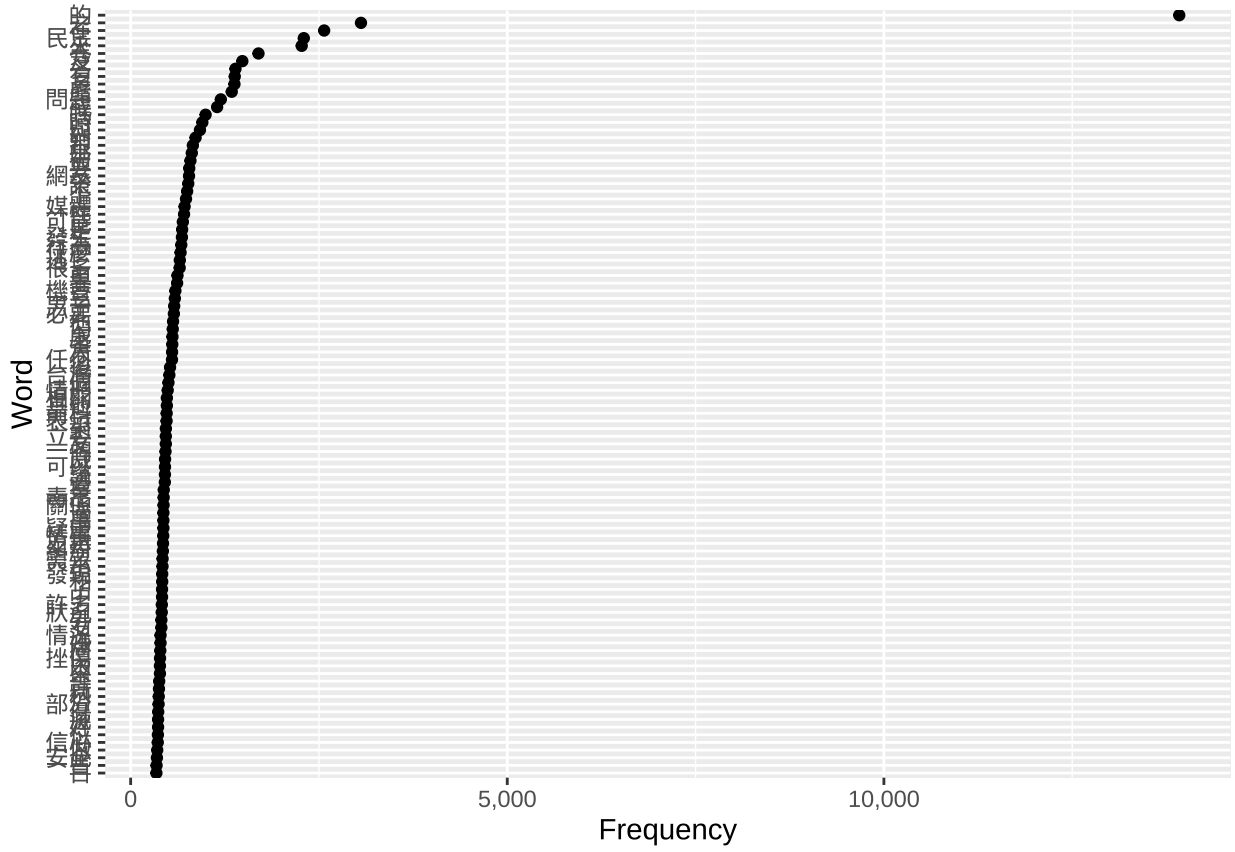
```
## dfm_trim() changed from 21,355 features (65,657 documents) to 539 features (65,657 documents)
```

```
textplot_wordcloud(kwic_post_freq_trim_WithDE,
                   max_words = 100,
                   min_size = 1,
                   max_size = 5,
                   rotation = FALSE,
                   random_order = FALSE,
                   color = c('red', 'green', 'blue', "orange"))
```

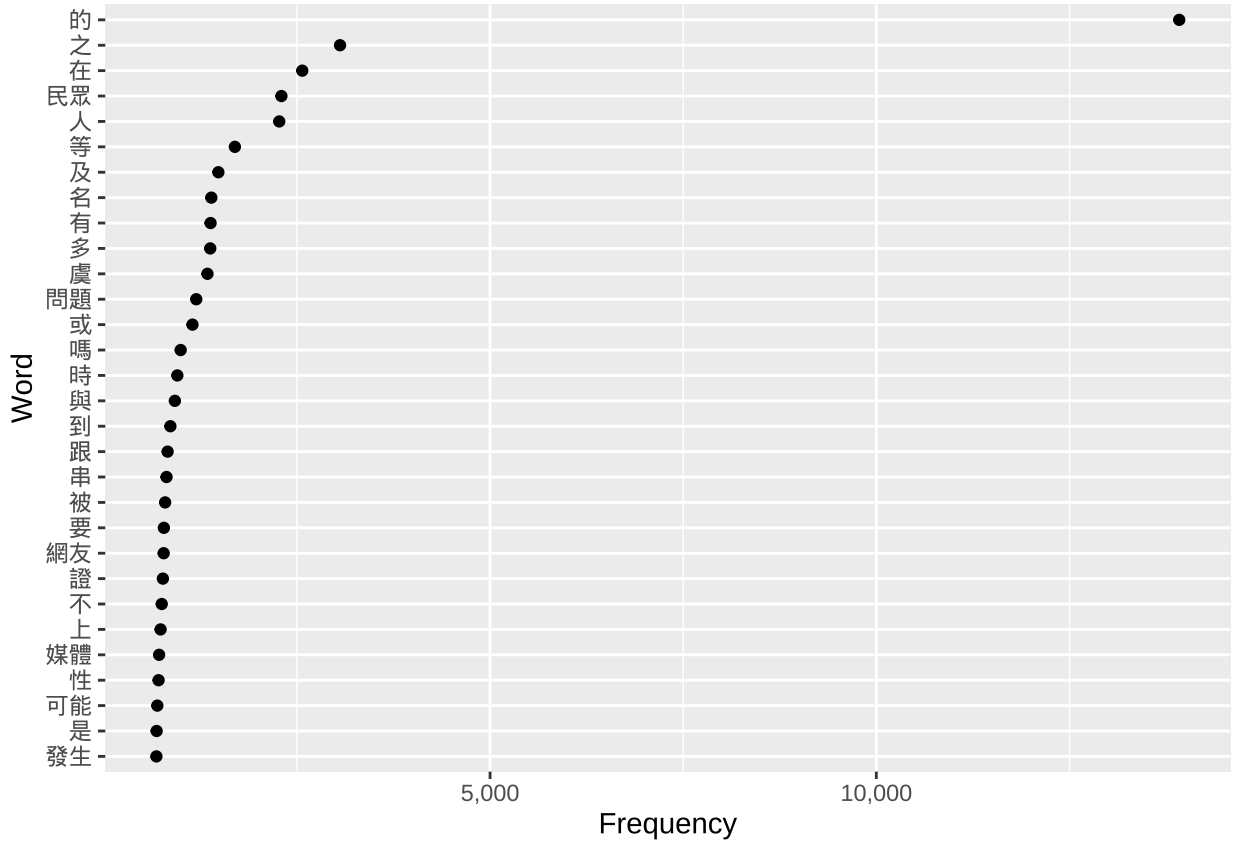

```
textplot_wordcloud(kwic_post_freq_trim_WithoutDE,
                   max_words = 100,
                   min_size = 1,
                   max_size = 5,
                   rotation = FALSE,
                   random_order = FALSE,
                   color = c('red', 'green', 'blue', "orange"))
```

2.1.2 Frequency plots

```
## We need to tranform the tokenized data into a 'dfm' dataset
kwic_data_freq_PostContext <- dfm(
  tokens(kwic_data$post,
         remove_punct = TRUE,
         remove_numbers = TRUE,
         remove_symbols = TRUE)
)

kwic_data_freq_PostContext <- textstat_frequency(kwic_data_freq_PostContext)

## Now plots
# Plot 1
ggplot(head(kwic_data_freq_PostContext, 100),
       aes(x = frequency,
           y = reorder(feature, frequency))) +
  geom_point() +
  labs(x = "Frequency", y = "Word") +
  scale_x_continuous(labels = label_comma())
```




```
# Plot 2
ggplot(head(kwic_data_freq_PostContext, 30),
       aes(x = frequency,
           y = reorder(feature, frequency))) +
  geom_point() +
  labs(x = "Frequency", y = "Word") +
  scale_x_continuous(labels = label_comma())
```



2.2 Visualizing frequency of the first word following the keyword

2.2.1 Word clouds

```
## Extract the first word
kwic_data$post_first_word <- word(kwic_data$post, 1)

## We need to tranform the tokenized data into a 'dfm' dataset
kwic_post_FirstWord_freq <- dfm(
  tokens(kwic_data$post_first_word,
         remove_punct = TRUE,
         remove_numbers = TRUE,
         remove_separators = TRUE))

kwic_postFirstWord_freq_trim <- dfm_trim(kwic_post_FirstWord_freq,
                                         verbose = TRUE)
```

```
## dfm_trim() changed from 7,690 features (65,657 documents) to 7,690 features (65,657 documents)
```

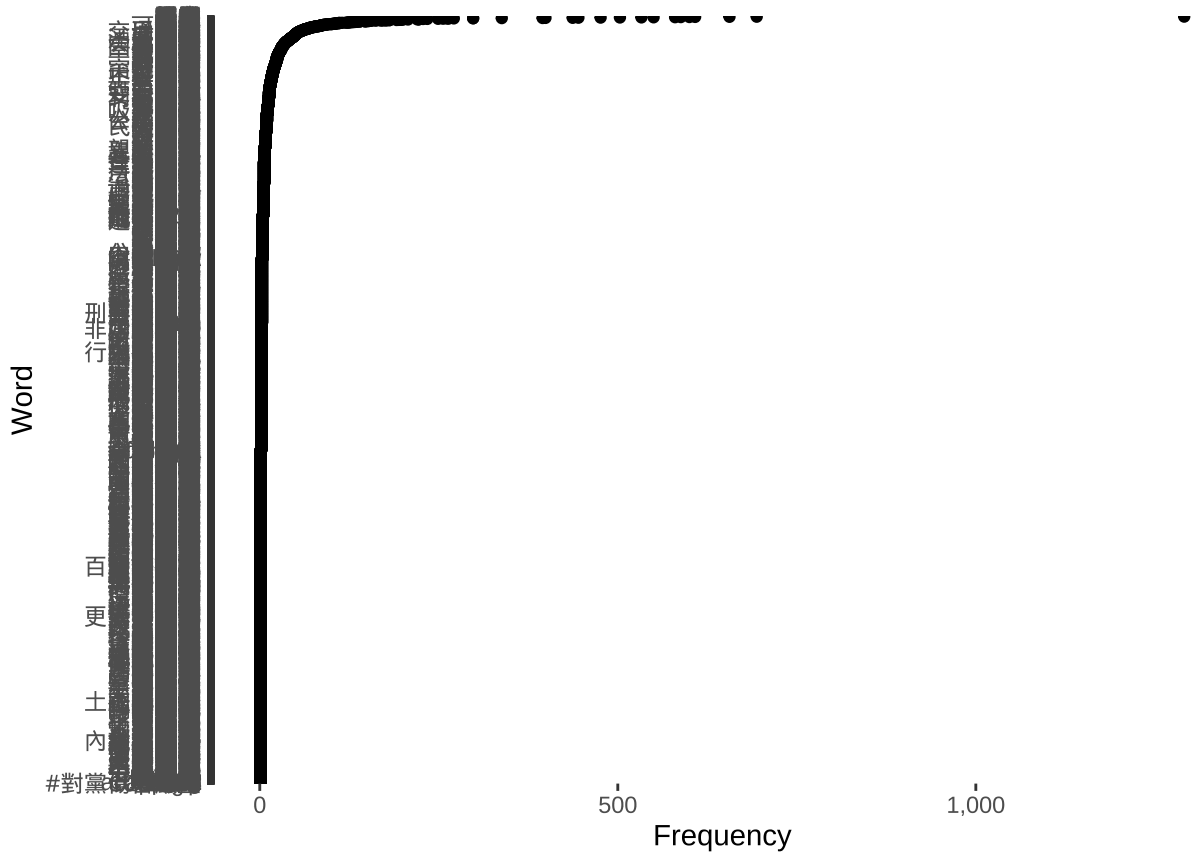
```
textplot_wordcloud(kwic_postFirstWord_freq_trim,
                   max_words = 100,
                   min_size = 1,
                   max_size = 5,
```

```
                  rotation = FALSE,
                  random_order = FALSE,
                  color = c('red', 'green', 'blue', "orange", "cyan"))
```

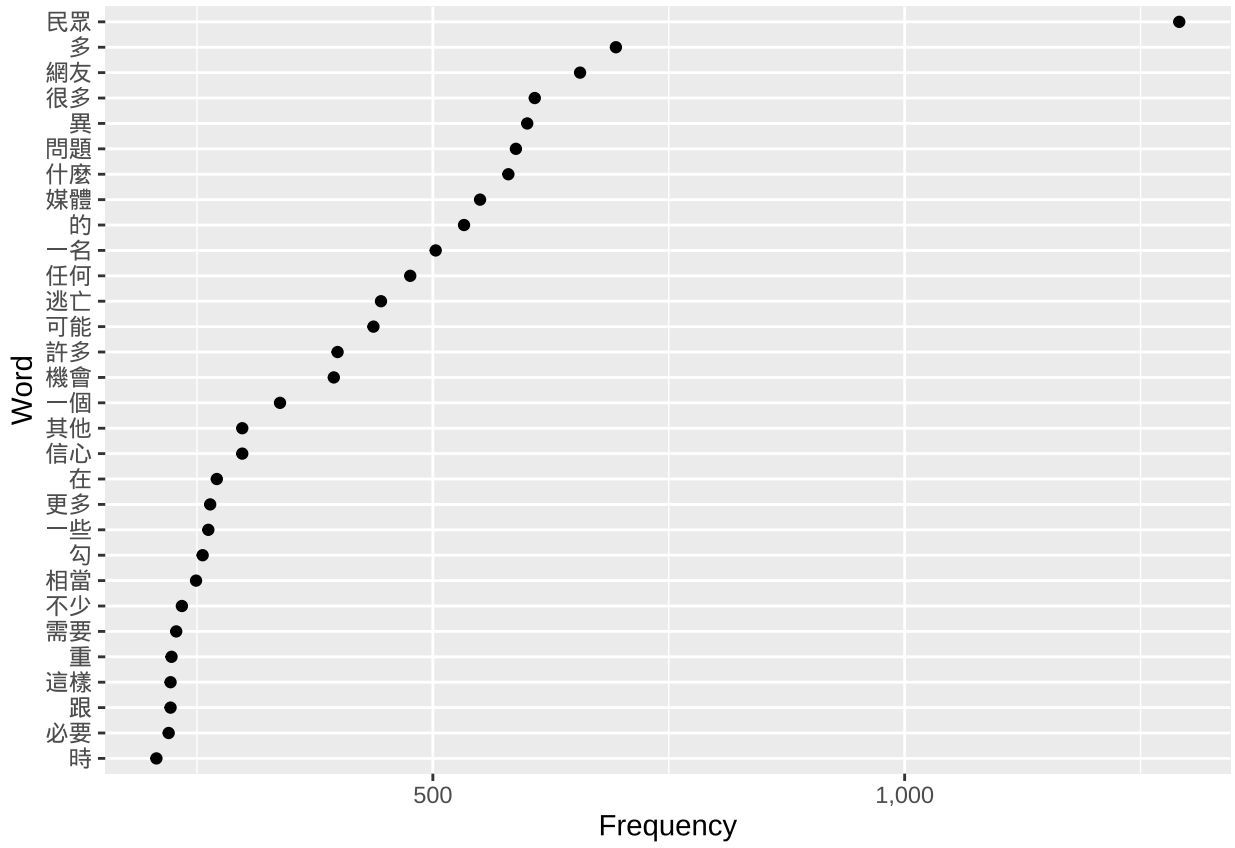
2.2.2 Frequency plots

```
## Extract the first word
kwic_data$post_first_word <- word(kwic_data$post, 1)

## We need to tranform the tokenized data into a 'dfm' dataset
kwic_data_freq_PostFirstWord <- dfm(
  tokens(kwic_data$post_first_word,
         remove_punct = TRUE,
         remove_numbers = TRUE,
         remove_symbols = TRUE)
)

kwic_data_freq_PostFirstWord <- textstat_frequency(kwic_data_freq_PostFirstWord)

## Now plots
# Plot 1
ggplot(kwic_data_freq_PostFirstWord,
       aes(x = frequency,
```

```
          y = reorder(feature, frequency))) +
  geom_point() +
  labs(x = "Frequency", y = "Word") +
  scale_x_continuous(labels = label_comma())
```




```
# Plot 2
ggplot(head(kwic_data_freq_PostFirstWord, 30),
       aes(x = frequency,
           y = reorder(feature, frequency))) +
  geom_point() +
  labs(x = "Frequency", y = "Word") +
  scale_x_continuous(labels = label_comma())
```



3. Frequency tables

3.1 Use word clouds

```
## We need to tranform the tokenized data into a 'dfm' dataset
Article_tokens_frequency <- dfm(
  tokens(Article_total2$body,
         remove_punct = TRUE,
         remove_numbers = TRUE,
         remove_separators = TRUE))

Article_tokens_frequency_trim_WithDE <- dfm_trim(Article_tokens_frequency,
                                          min_termfreq = 10000,
                                          verbose = TRUE)
```

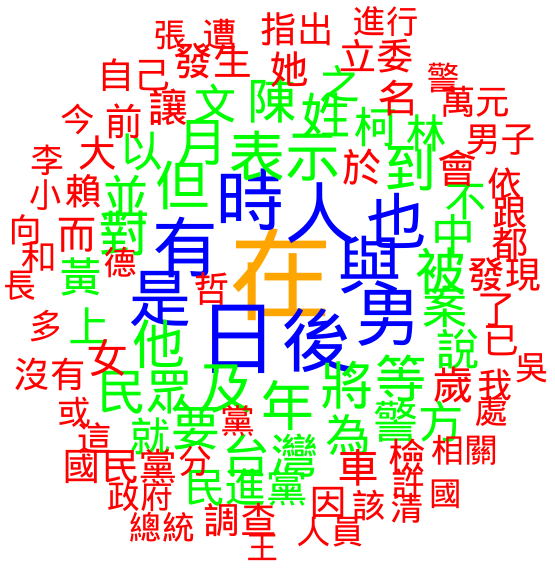
```
## dfm_trim() changed from 69,862 features (247,374 documents) to 205 features (247,374 documents)
```

```
Article_tokens_frequency_trim_WithoutDE <- dfm_trim(Article_tokens_frequency,
                                          min_termfreq = 10000,
                                          max_termfreq = 300000,
                                          verbose = TRUE)
```

```
## dfm_trim() changed from 69,862 features (247,374 documents) to 204 features (247,374 documents)
```

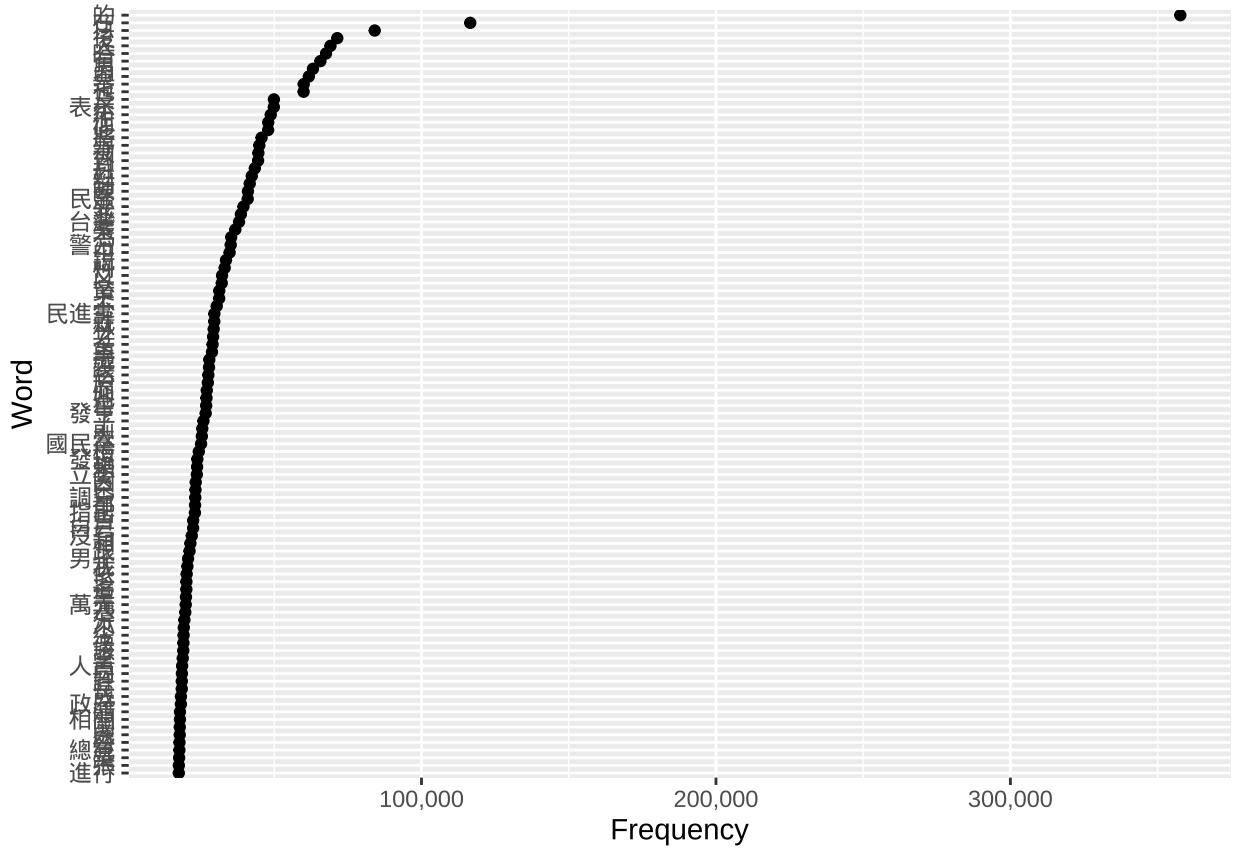
```
textplot_wordcloud(Article_tokens_frequency_trim_WithDE,
                   max_words = 100,
                   min_size = 1,
                   max_size = 5,
                   rotation = FALSE,
                   random_order = FALSE,
                   color = c('red', 'green', 'blue', "orange"))
```


```
textplot_wordcloud(Article_tokens_frequency_trim_WithoutDE,
                   max_words = 100,
                   min_size = 1,
                   max_size = 5,
                   rotation = FALSE,
                   random_order = FALSE,
                   color = c('red', 'green', 'blue', "orange"))
```

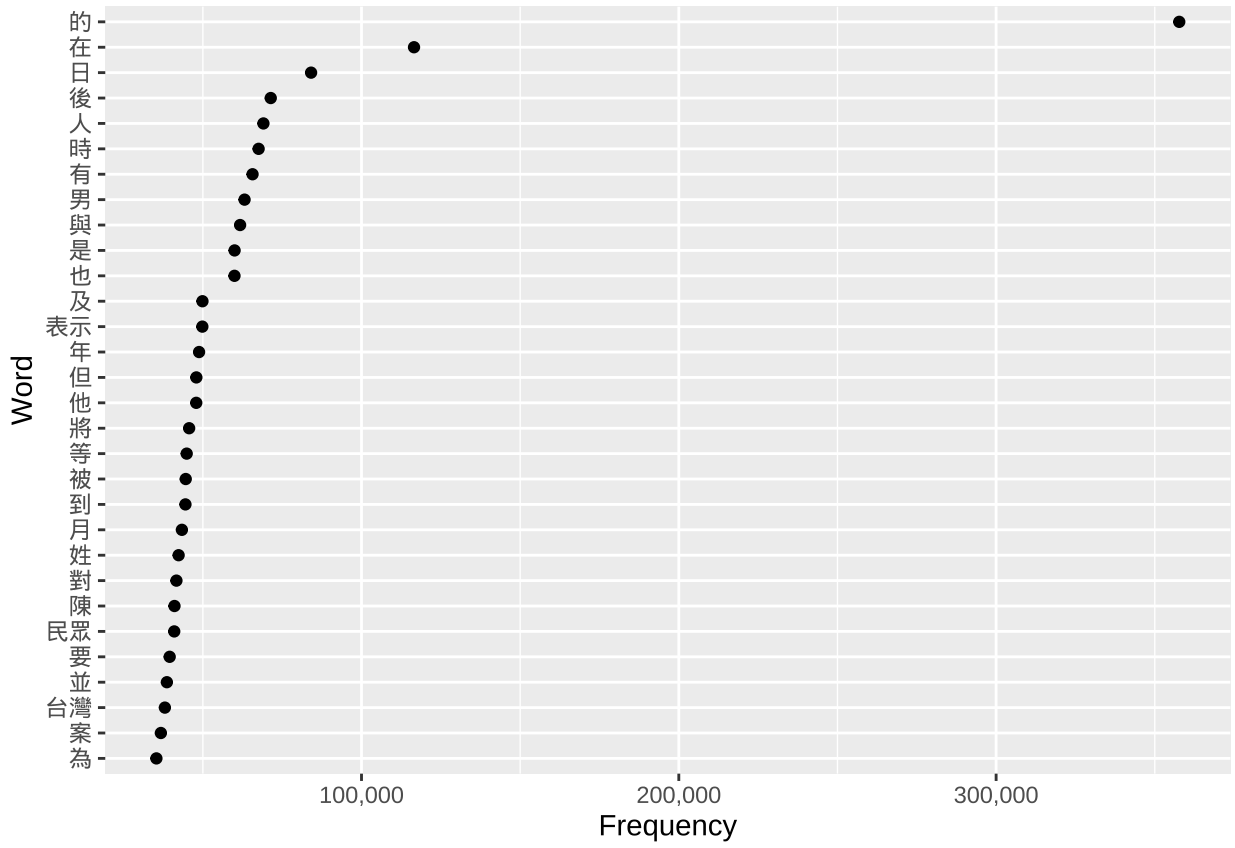


3.2 Frequency plots with ggplot

```
## We can directly use the dataset we created last week
# Plot 1
ggplot(table_FreqWord_Top100,
       aes(x = frequency,
           y = reorder(feature, frequency))) +
  geom_point() +
  labs(x = "Frequency", y = "Word") +
  scale_x_continuous(labels = label_comma())
```




```
# Plot 2; Observe attentively: What's the difference with the plot above?
ggplot(head(table_FreqWord_Top100, 30),
       aes(x = frequency,
           y = reorder(feature, frequency))) +
  geom_point() +
  labs(x = "Frequency", y = "Word") +
  scale_x_continuous(labels = label_comma())
```


的
在
日
後
人
時
有
男
與
是
也
及
表示
年
但
他
將
等
被
到
月
姓
對
陳
民眾
要
並
台灣
案
為
Word
100,000
200,000
300,000
Frequency