# Fake_News

*Dada's Lambda*

*5/6/2020*

```
library(data.table)
library(stringr)
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v forcats 0.4.0
## v readr   1.3.1
```

```
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(purrr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(tidytext)
library(widyr)
library(rlang)
```

```
##
## Attaching package: 'rlang'
## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl,
##     flatten_int, flatten_lgl, flatten_raw, invoke, list_along,
##     modify, prepend, splice

## The following object is masked from 'package:data.table':
##
##     :=
```

Historic US Confirmed Cases Cata

```r
cases <- read.csv("https://covidtracking.com/api/v1/us/daily.csv")
```

Gedelt Data

```r
grabRemote <- function(url) {
    temp <- tempfile()
    download.file(url, temp)
    aap.file <- read.csv(gzfile(temp), as.is = TRUE)
    unlink(temp)
    return(aap.file)
}

gdelt_path <- read.table("http://data.gdeltproject.org/blog/2020-coronavirus-narrative/live_onlinenews/l
gdelt_path <- vapply(gdelt_path, as.character, character(nrow(gdelt_path)))
gdelt_path <- as.matrix(gdelt_path[str_detect(gdelt_path, "falsehood")])
gdelt_data <- apply(gdelt_path, 1, grabRemote)
```

Gedelt Data Cleaning

```r
date <- as.Date(substr(gdelt_data[[1]][[1]], 1, 10), "%Y-%m-%d")
for(i in 2:44){
    date <- c(date, as.Date(substr(gdelt_data[[i]][[1]], 1, 10), "%Y-%m-%d"))
}
x <- sort(unique(c(date, as.Date(as.character(cases$date), "%Y%m%d"))))
date_table <- table(date)
news <- ifelse(x %in% as.Date(names(date_table)), date_table[as.character(x)], NA)
news_percentage <- news/max(news, na.rm = TRUE)
cases[[1]] <- as.Date(as.character(cases$date), "%Y%m%d")
positive <- cases$positiveIncrease
names(positive) <- cases$date
case <- ifelse(x %in% cases$date, positive[as.character(x)], NA)
case_percentage <- case/max(case, na.rm = TRUE)
n <- length(case)
```

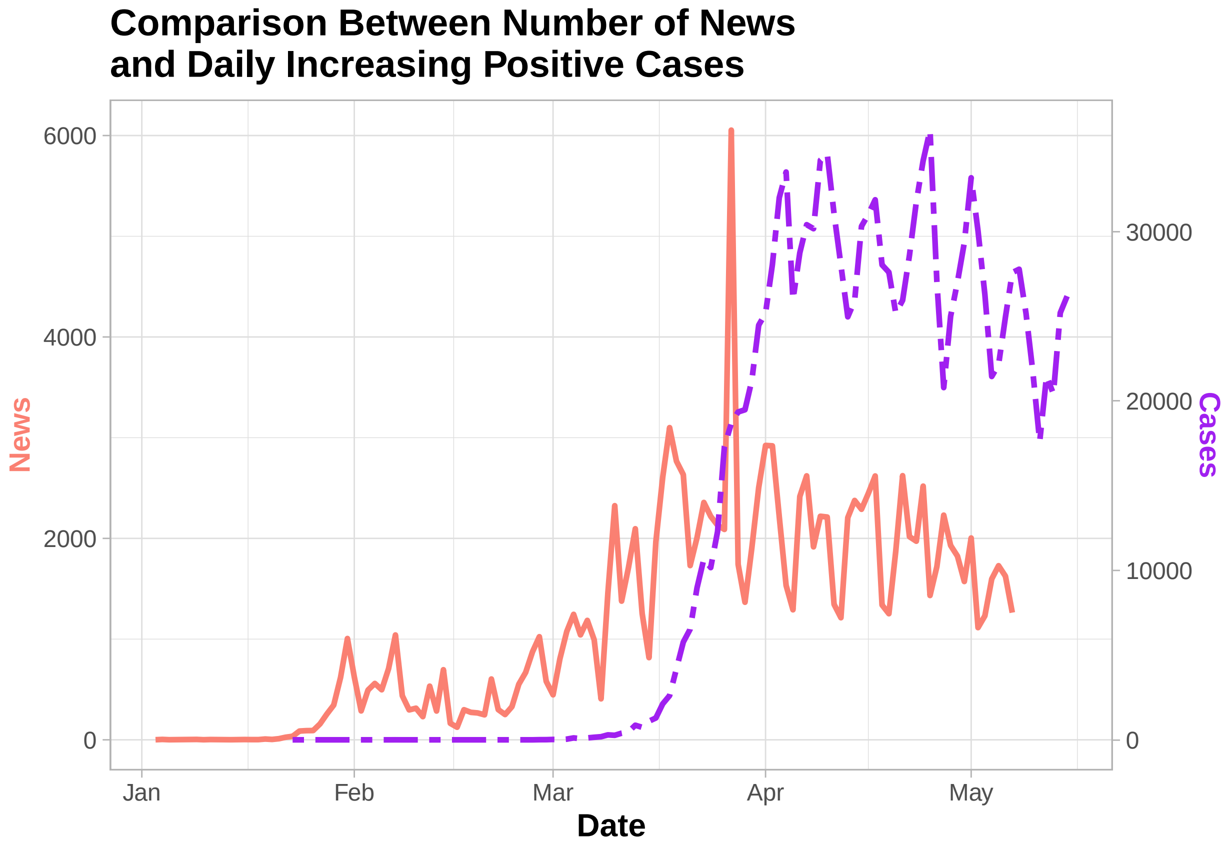
using ggplot

## news and cases

```r
nc <- tibble("Date" = x, news, case, case2 = c(case[6:n], rep(NA, 5)))
coeff <- max(case, na.rm = TRUE)/max(news, na.rm = TRUE)
ggplot(nc, aes(x=Date)) +
    geom_line(aes(y = news), color = "salmon", size = 1) +
    geom_line(aes(y = case / coeff), color="purple", linetype="twodash", size = 1)+
    scale_y_continuous(
        name = "News",
        sec.axis = sec_axis(~.*coeff, name="Cases")
    )+
    theme_light()+
    theme(
        axis.title.y = element_text(color = "salmon", size=11),
        axis.title.y.right = element_text(color = "purple", size=11),
        title =element_text(size=12, face='bold')
    ) +
    ggtitle("Comparison Between Number of News \nand Daily Increasing Positive Cases")
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```

```
## Warning: Removed 14 rows containing missing values (geom_path).
```

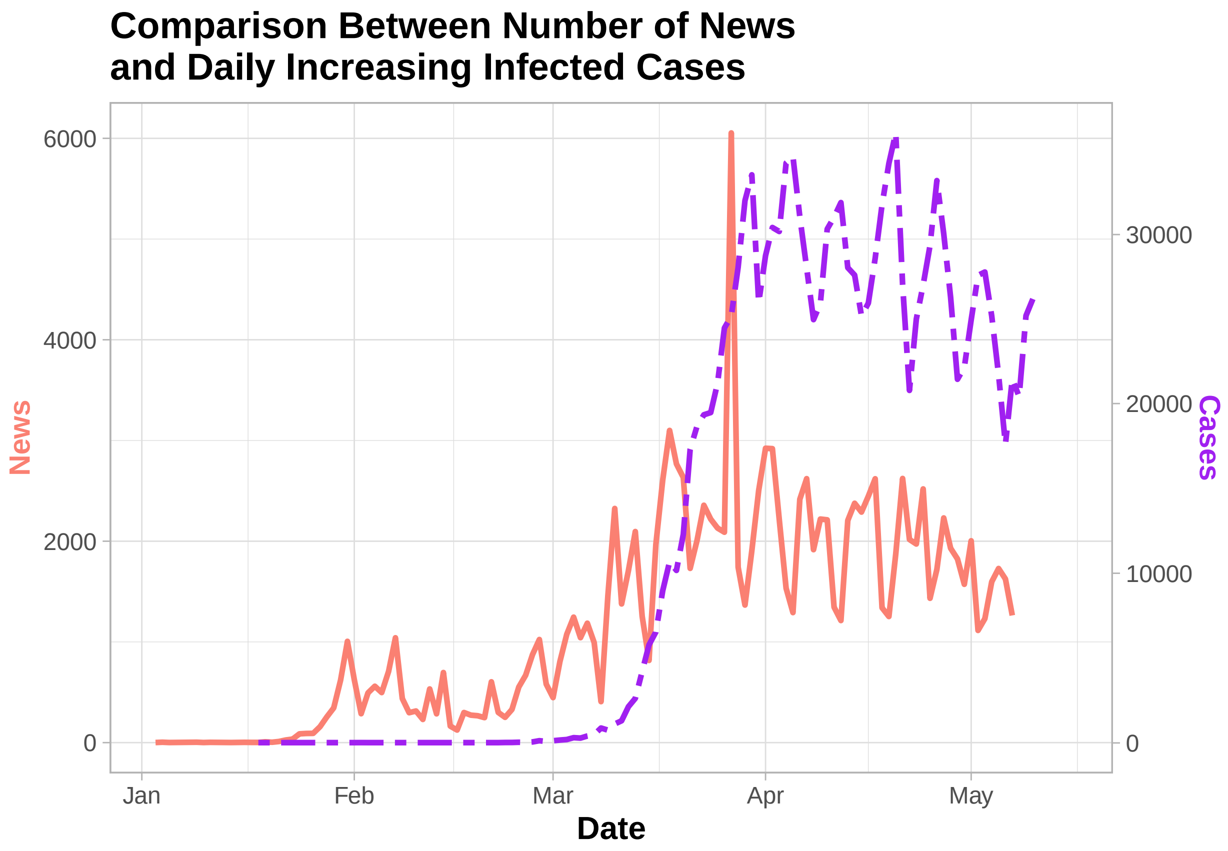


14 days after

```r
ggplot(nc, aes(x=Date)) +
    geom_line(aes(y = news), color = "salmon", size = 1) +
    geom_line(aes(y = case2 / coeff), color="purple", linetype="twodash", size = 1)+
    scale_y_continuous(
        name = "News",
        sec.axis = sec_axis(~.*coeff, name="Cases")
    )+
    theme_light()+
    theme(
        axis.title.y = element_text(color = "salmon", size=11),
        axis.title.y.right = element_text(color = "purple", size=11),
        title =element_text(size=12, face='bold')
    ) +
    ggtitle("Comparison Between Number of News \nand Daily Increasing Infected Cases")
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```

```
## Warning: Removed 14 rows containing missing values (geom_path).
```

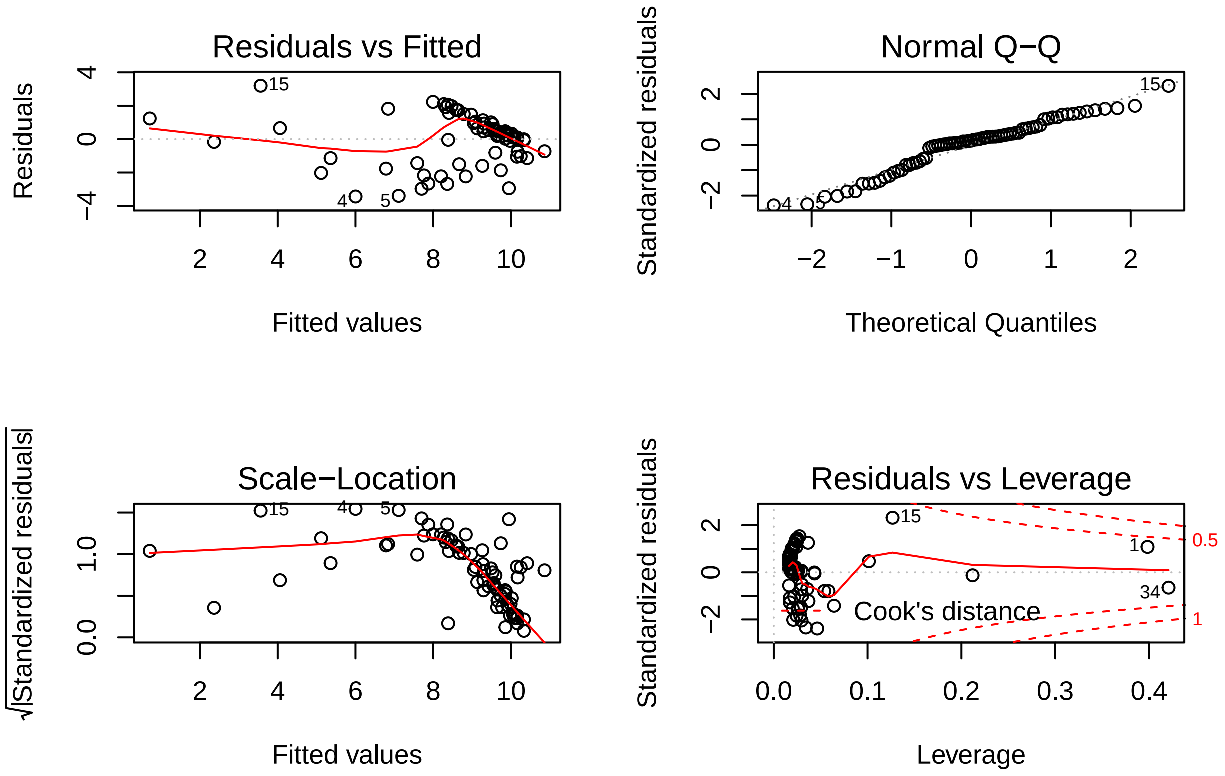


model

```r
model <- lm(log(case[51:125])~poly(log(news[46:120]), 2))
summary(model)
```

```
##
## Call:
## lm(formula = log(case[51:125]) ~ poly(log(news[46:120]), 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4375 -0.9188  0.2364  0.9125  3.1995
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   8.7402     0.1705  51.271  < 2e-16 ***
## poly(log(news[46:120]), 2)1  15.5648     1.4763  10.543 3.01e-16 ***
## poly(log(news[46:120]), 2)2  -4.0963     1.4763  -2.775  0.00703 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.476 on 72 degrees of freedom
## Multiple R-squared:  0.6227, Adjusted R-squared:  0.6123
## F-statistic: 59.43 on 2 and 72 DF,  p-value: 5.74e-16
```

```r
par(mfrow = c(2, 2))
plot(model)
```

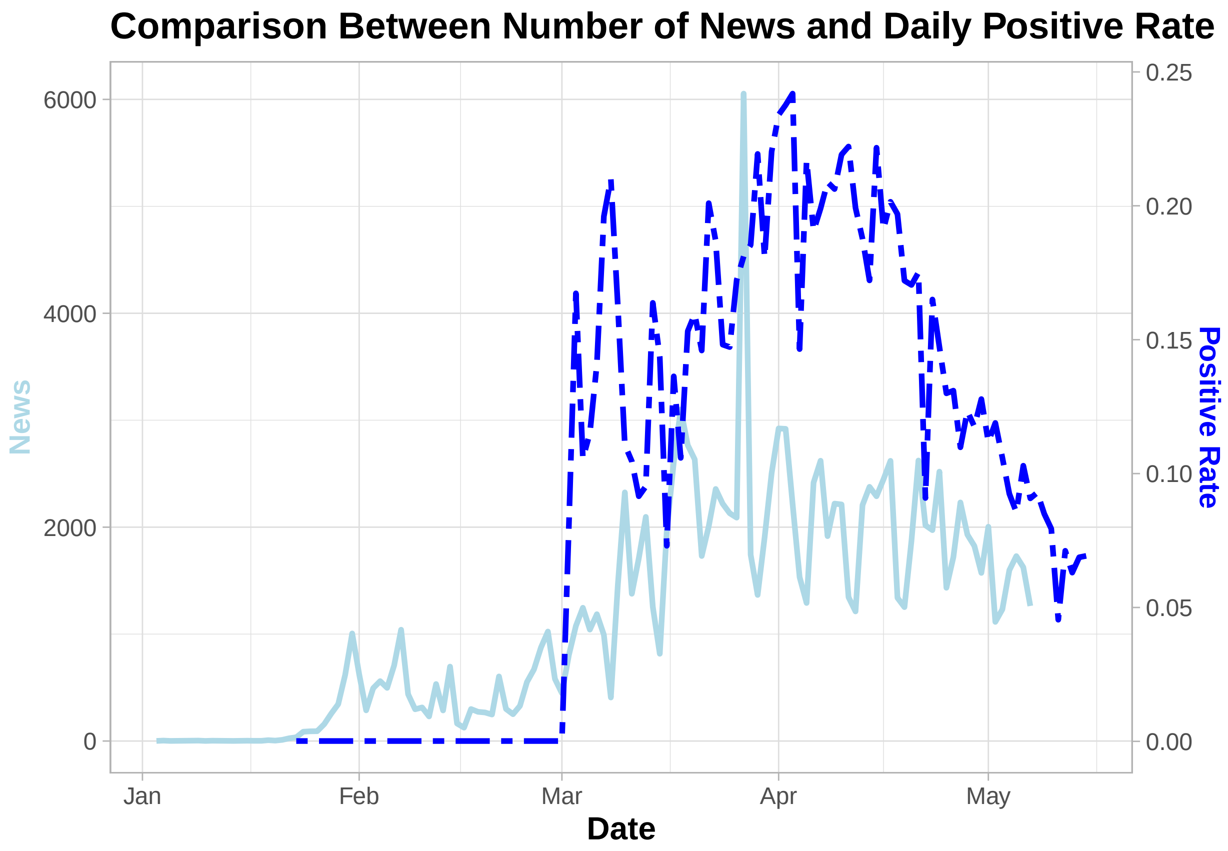


### news and rate

```r
rate <- ifelse(cases$totalTestResultsIncrease < 100, 0, cases$positiveIncrease/cases$totalTestResultsInc
names(rate) <- cases$date
rate <- ifelse(x %in% as.Date(cases$date), rate[as.character(x)], NA)
nr <- tibble("Date" = x, news, rate, rate2 = c(rate[6:n], rep(NA, 5)))
coeff <- max(rate, na.rm = TRUE)/max(news, na.rm = TRUE)
ggplot(nr, aes(x=Date)) +
    geom_line(aes(y = news), color = "lightblue", size = 1) +
    geom_line(aes(y = rate / coeff), color="blue", linetype="twodash", size = 1)+
    scale_y_continuous(
        name = "News",
        sec.axis = sec_axis(~.*coeff, name="Positive Rate")
    )+
    theme_light()+
    theme(
        axis.title.y = element_text(color = "lightblue", size=11),
        axis.title.y.right = element_text(color = "blue", size=11),
        title =element_text(size=12, face='bold')
    ) +
    ggtitle("Comparison Between Number of News and Daily Positive Rate")
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```

```
## Warning: Removed 14 rows containing missing values (geom_path).
```

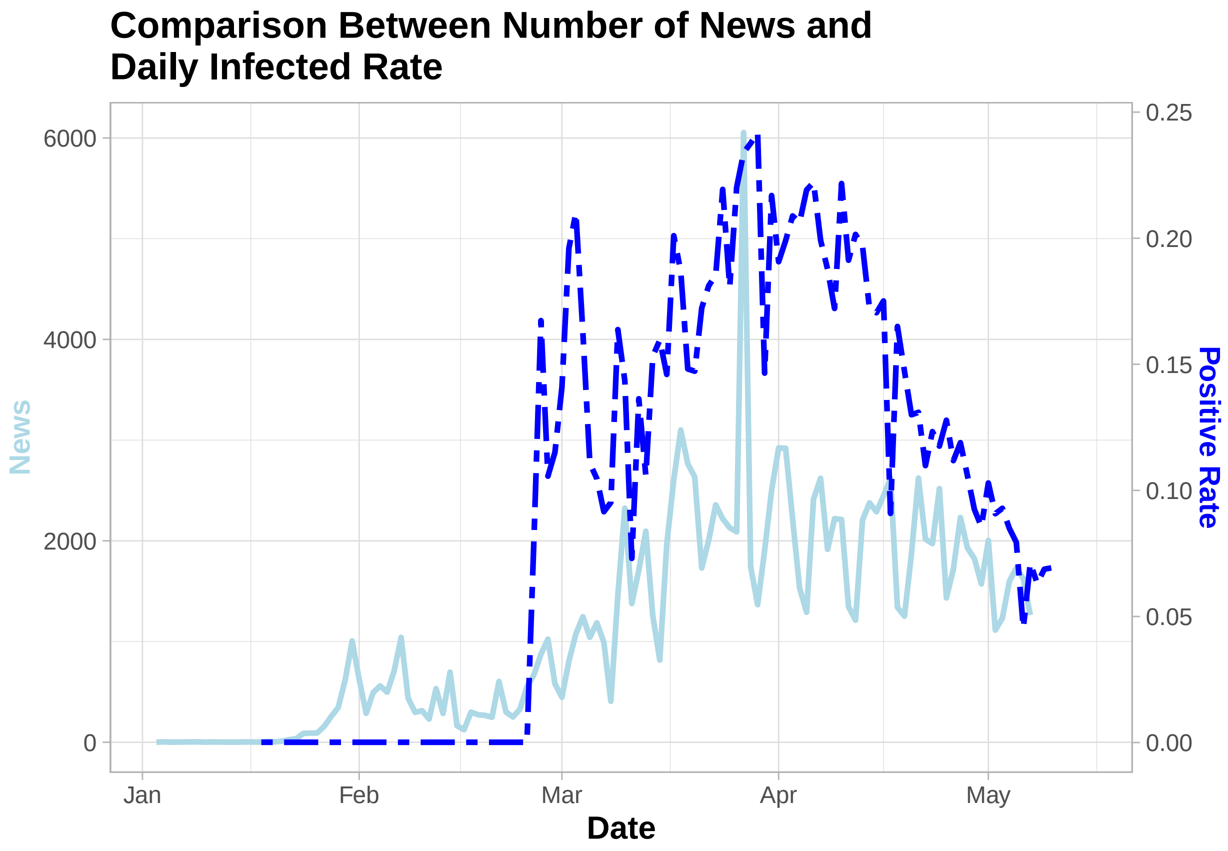


14 days after

```
ggplot(nr, aes(x=Date)) +
    geom_line(aes(y = news), color = "lightblue", size = 1) +
    geom_line(aes(y = rate2 / coeff), color="blue", linetype="twodash", size = 1)+
    scale_y_continuous(
        name = "News",
        sec.axis = sec_axis(~.*coeff, name="Positive Rate")
    )+
    theme_light()+
    theme(
        axis.title.y = element_text(color = "lightblue", size=11),
        axis.title.y.right = element_text(color = "blue", size=11),
        title =element_text(size=12, face='bold')
    ) +
    ggtitle("Comparison Between Number of News and \nDaily Infected Rate")
```

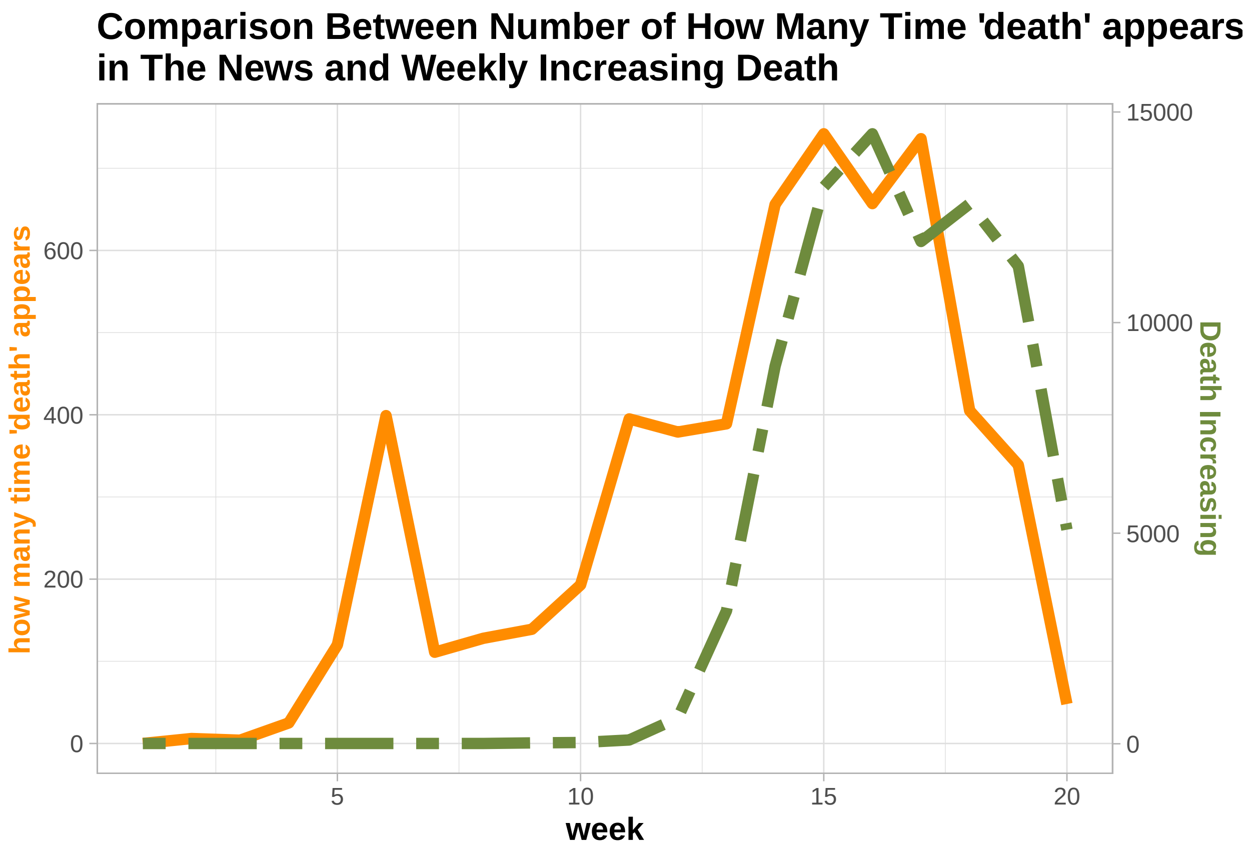
```
## Warning: Removed 8 rows containing missing values (geom_path).
```

```
## Warning: Removed 14 rows containing missing values (geom_path).
```



news and death

```
words <- read.csv("dada.csv")
death_num <- c(0, words$n[words$word == "death"])
death <- tibble(date = cases$date, death = cases$deathIncrease) %>% mutate(week = week(ymd(as.Date(date
death <- tibble(week = c(1, 2, 3, death$week), death = c(0, 0, 0, 0, death$death[-1]), death_num)
coeff <- max(death$death, na.rm = TRUE)/max(death$death_num, na.rm = TRUE)
ggplot(death, aes(x=week)) +
    geom_line(aes(y = death_num), color = "darkorange", size = 2) +
    geom_line(aes(y = death / coeff), color="darkolivegreen4", linetype="twodash", size = 2)+
    scale_y_continuous(
        name = "how many time 'death' appears",
        sec.axis = sec_axis(~.*coeff, name="Death Increasing")
    )+
    theme_light()+
    theme(
        axis.title.y = element_text(color = "darkorange", size=11),
        axis.title.y.right = element_text(color = "darkolivegreen4", size=11),
        title =element_text(size=12, face='bold')
    ) +
    ggtitle("Comparison Between Number of How Many Time 'death' appears \nin The News and Weekly Increa
```

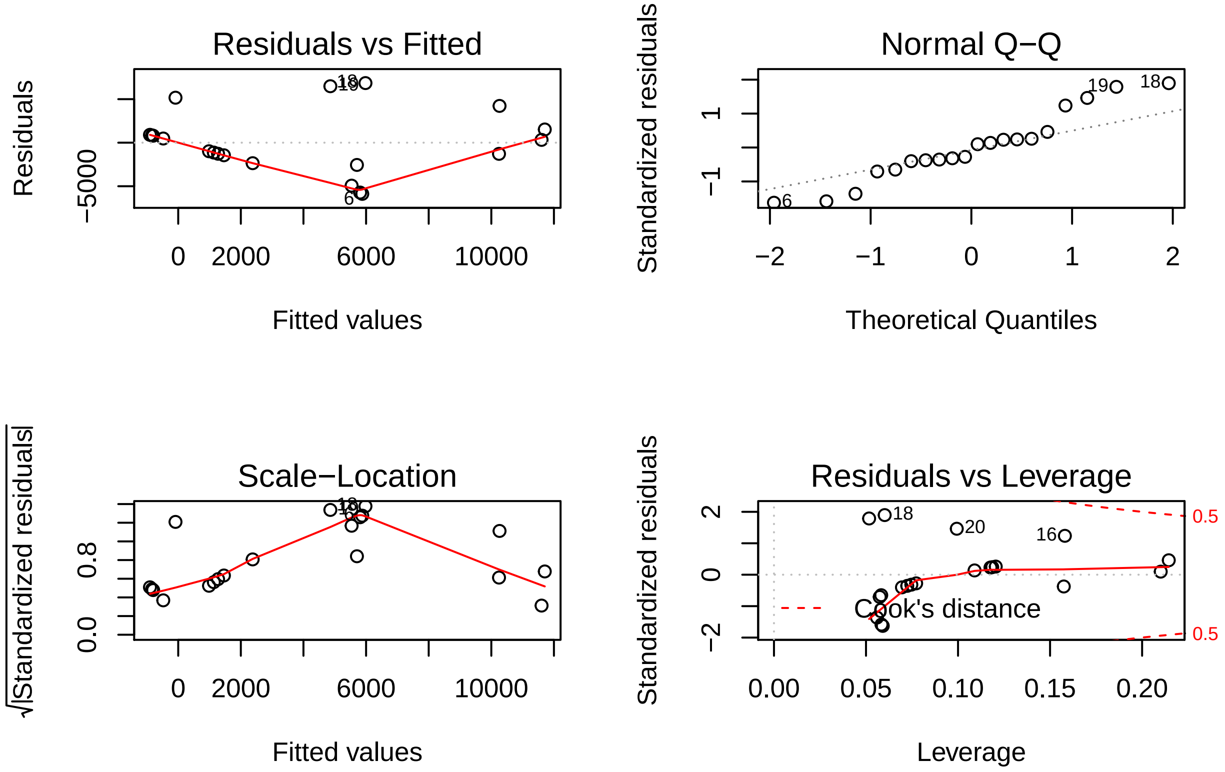


model

```
model2 <- lm(death$death~death$death_num)
summary(model2)
```

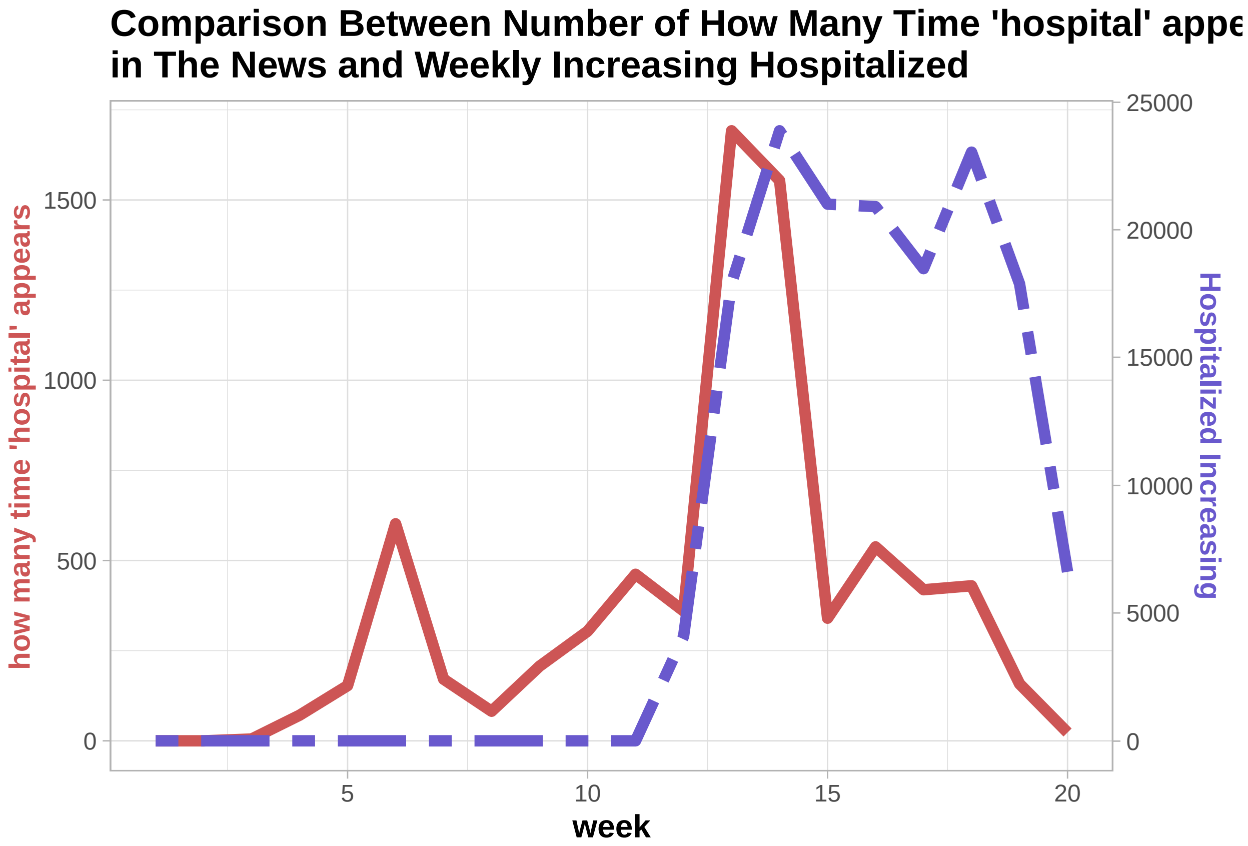
```
##
## Call:
## lm(formula = death$death ~ death$death_num)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5878.5 -1673.2  -330.9  1056.0  6844.5
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -902.207   1292.572  -0.698    0.494
## death$death_num   16.994      3.368   5.046  8.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3724 on 18 degrees of freedom
## Multiple R-squared:  0.5859, Adjusted R-squared:  0.5628
## F-statistic: 25.46 on 1 and 18 DF,  p-value: 8.405e-05
```

```r
par(mfrow = c(2, 2))
plot(model2)
```



### news and hospitalied

```
hos_num <- c(0, 0, words$n[words$word == "hospital"])
hospitalized <- tibble(date = cases$date, hospitalized = cases$hospitalizedIncrease) %>% mutate(week = 
hospitalized <- tibble(week = c(1, 2, 3, hospitalized$week), hospitalized = c(0, 0, 0, 0, hospitalized$
coeff <- max(hospitalized$hospitalized, na.rm = TRUE)/max(hospitalized$hos_num, na.rm = TRUE)
ggplot(hospitalized, aes(x=week)) +
    geom_line(aes(y = hos_num), color = "indianred3", size = 2) +
    geom_line(aes(y = hospitalized / coeff), color="slateblue3", linetype="twodash", size = 2)+
    scale_y_continuous(
        name = "how many time 'hospital' appears",
        sec.axis = sec_axis(~.*coeff, name="Hospitalized Increasing")
    )+
    theme_light()+
    theme(
        axis.title.y = element_text(color = "indianred3", size=11),
        axis.title.y.right = element_text(color = "slateblue3", size=11),
        title =element_text(size=12, face='bold')
    ) +
    ggtitle("Comparison Between Number of How Many Time 'hospital' appears \nin The News and Weekly Inc
```

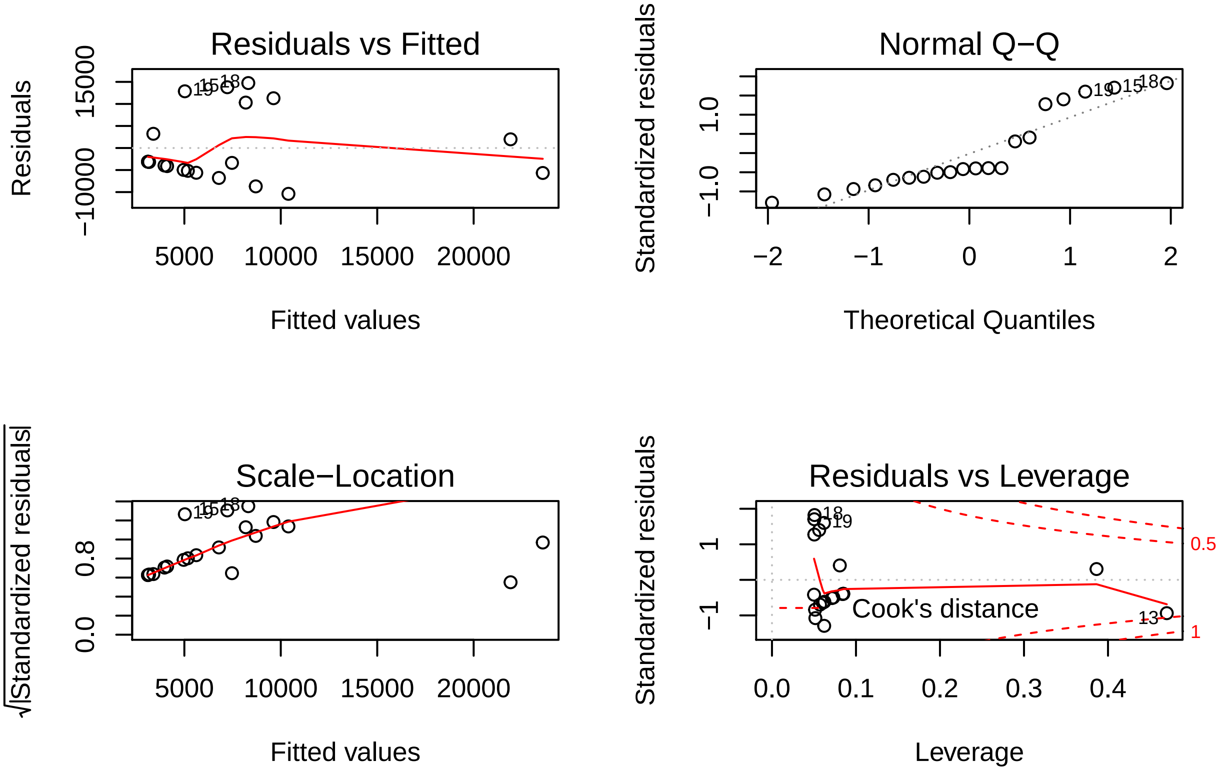


model

```
model3 <- lm(hospitalized$hospitalized~hospitalized$hos_num)
summary(model3)
```

```
##
## Call:
## lm(formula = hospitalized$hospitalized ~ hospitalized$hos_num)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -10396  -5291  -3276   4994  14733
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3113.728   2417.429   1.288  0.21405
## hospitalized$hos_num    12.098      4.094   2.955  0.00847 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8297 on 18 degrees of freedom
## Multiple R-squared:  0.3267, Adjusted R-squared:  0.2893
## F-statistic: 8.733 on 1 and 18 DF,  p-value: 0.008473
```

```
par(mfrow = c(2, 2))
plot(model3)
```



Text Mining

```r
gdelt_clean_data <- pluck(gdelt_data, 1)
colnames(gdelt_clean_data) <- c("date", "url", "title", "misinformation")

for (i in 2:length(gdelt_data)) {
    x <- pluck(gdelt_data, i)
    names(x) <- c("date", "url", "title", "misinformation")
    gdelt_clean_data <- rbind(gdelt_clean_data, x)
}

gdelt_clean_data <- as_tibble(gdelt_clean_data) %>% mutate(date = as.Date(substr(date, 1, 10), "%Y-%m-%
```

Most Common Words Bar Chart Race by Week

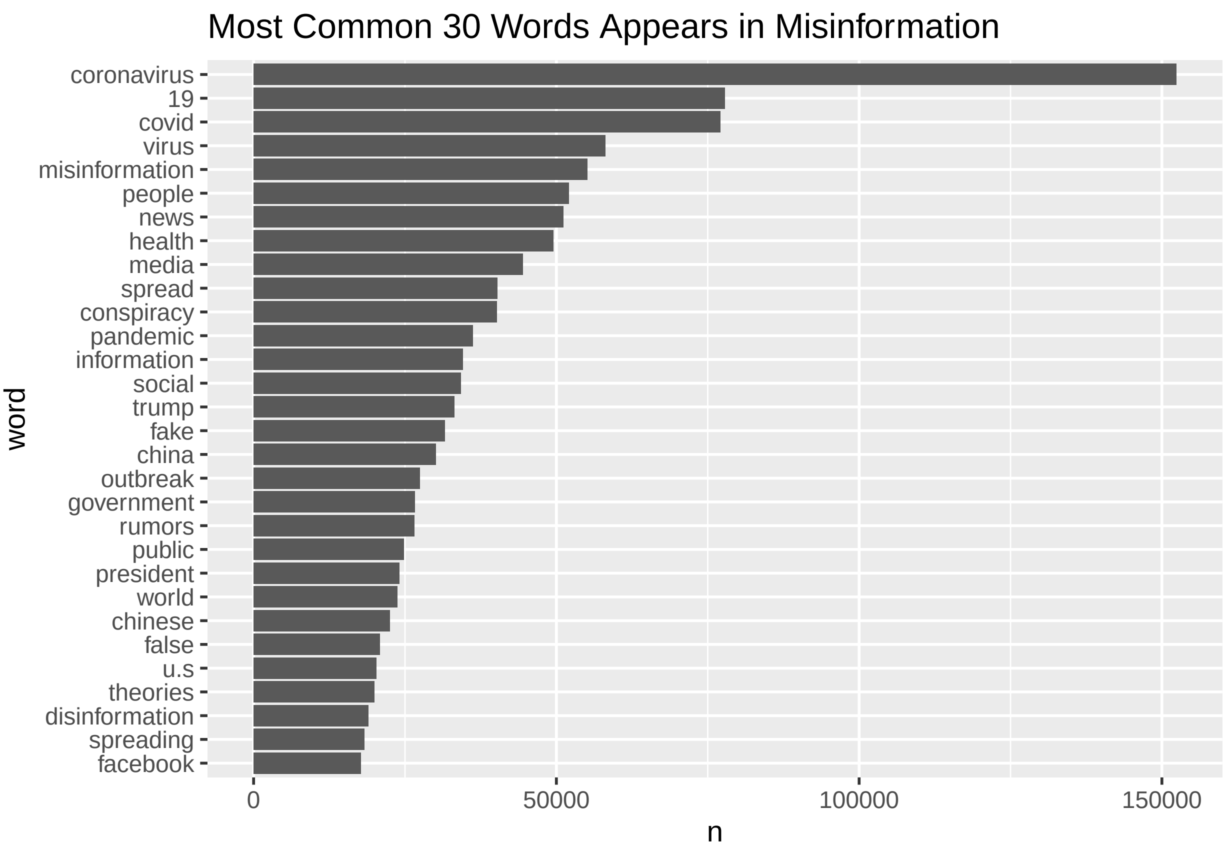
```r
gdelt_clean_data <- gdelt_clean_data %>% mutate("week" = week(ymd(date))) %>% select(date, week, url, ti

gdelt_clean_data_byweek <- gdelt_clean_data %>% unnest_tokens(word, misinformation) %>% select(week, wo
gdelt_clean_data_byweek <- pivot_wider(gdelt_clean_data_byweek, names_from = "week", values_from = "n")

common_words <- character()
for (i in seq(ncol(gdelt_clean_data_byweek) - 1)) {
  words <- gdelt_clean_data_byweek %>% select(word, as.character(i)) %>% arrange(desc(!!rlang::sym(as.cl
  common_words <- c(common_words, words$word)
}
common_words <- unique(common_words)
most_common_words_byweek <- gdelt_clean_data_byweek %>% filter(word %in% common_words)
# write.csv(most_common_words_byweek, "gdelt data")
```

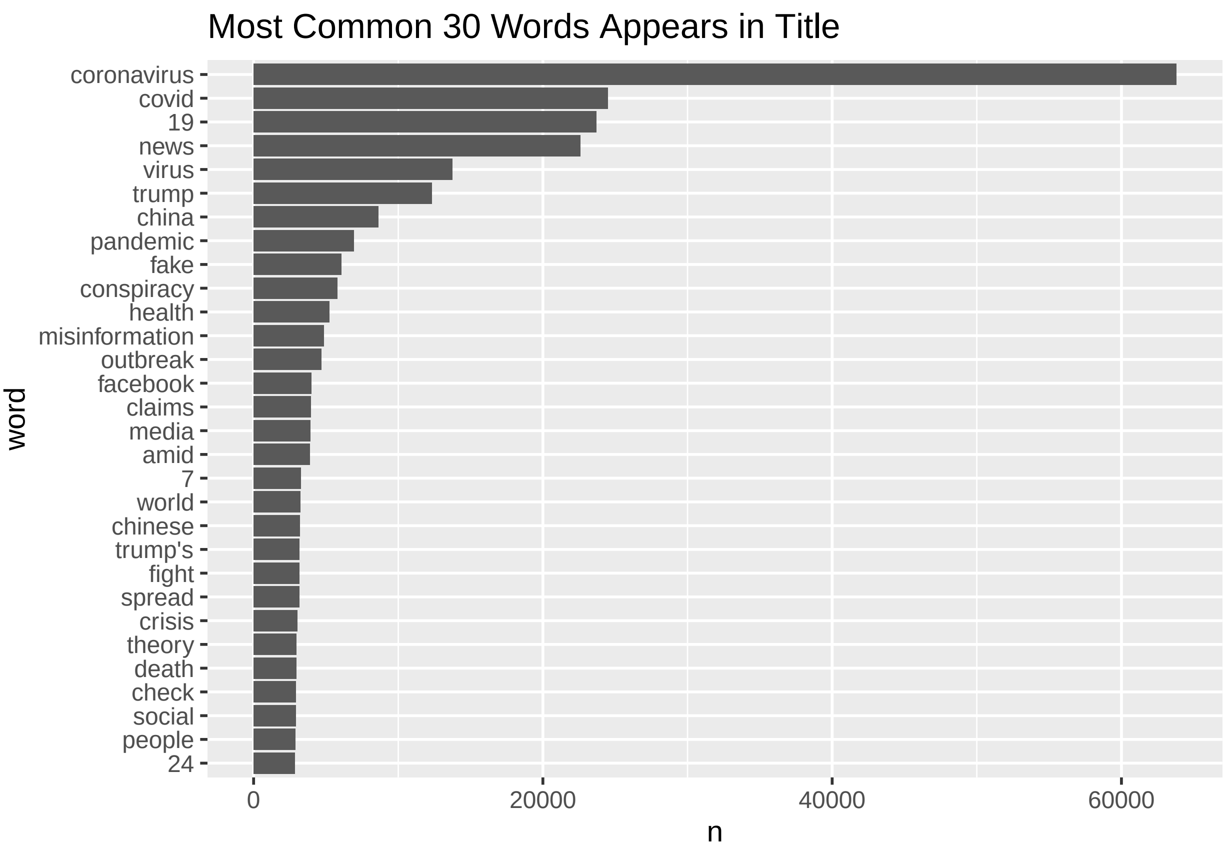
```r
# 30 most common words in misinformation
most_common_words <- gdelt_clean_data %>% unnest_tokens(word, misinformation) %>% count(word, sort = TRU

most_common_words %>% head(30) %>% mutate(word = reorder(word, n)) %>% ggplot(aes(word, n)) + geom_col(
```



```
# 30 most common words in title
most_common_title_words <-gdelt_clean_data %>% unnest_tokens(word, title) %>% count(word, sort = TRUE) %

most_common_title_words %>% head(30) %>% mutate(word = reorder(word, n)) %>% ggplot(aes(word, n)) + geom
```


Most Common 30 Words Appears in Title
coronavirus
covid
19
news
virus
trump
china
pandemic
fake
conspiracy
health
misinformation
outbreak
facebook
claims
media
amid
7
world
chinese
trump's
fight
spread
crisis
theory
death
check
social
people
24
word
0
20000
40000
60000
n