Code ▾

32061412_Assignment2

HongYi

28 April, 2024

Libraries loading

Hide

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.0
```

```
## Warning: package 'ggplot2' was built under R version 4.4.0
```

```
## Warning: package 'tibble' was built under R version 4.4.0
```

```
## Warning: package 'tidyr' was built under R version 4.4.0
```

```
## Warning: package 'readr' was built under R version 4.4.0
```

```
## Warning: package 'purrr' was built under R version 4.4.0
## Warning: package 'dplyr' was built under R version 4.4.0
```

```
## Warning: package 'stringr' was built under R version 4.4.0
```

```
## Warning: package 'forcats' was built under R version 4.4.0
```

```
## Warning: package 'lubridate' was built under R version 4.4.0
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all confli
cts to become errors
```

Hide

```
library(lubridate)
library(stringr)
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.4.0
```

```
## Loading required package: NLP
```

```
## Warning: package 'NLP' was built under R version 4.4.0
```

```
## 
## Attaching package: 'NLP'
## 
## The following object is masked from 'package:ggplot2':
## 
##     annotate
```

Hide

```
library(tokenizers)
```

```
## Warning: package 'tokenizers' was built under R version 4.4.0
```

Hide

```
library(wordcloud2)
```

```
## Warning: package 'wordcloud2' was built under R version 4.4.0
```

Data exploration

Hide

```
# read the news data
news <- read_csv("ireland_news.csv")
```

```
## Rows: 1611495 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (4): publish_date, headline_category, headline_text, news_provider
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
head(news)
```

```
## # A tibble: 6 × 4
##   publish_date                      headline_category headline_text news_provider
##   <chr>                             <chr>             <chr>         <chr>
## 1 Wednesday, 25th of March, 2015    opinion           Renua's plan… Irish Times
## 2 Tuesday, 30th of June, 1998       news              Racism cloud… Irish Examin…
## 3 Thursday, 13th of March, 2014     news.politics.oi… Minister for… RTE News
## 4 Wednesday, 28th of February, 20… opinion.letters   Kaczynski an… RTE News
## 5 Saturday, 17th of October, 2015   opinion           Martyn Turner TheJournal.ie
## 6 Sunday, 28th of January, 2018     business.markets  Chris Johns:… RTE News
```

Hide

```
glimpse(news)
```

```
## Rows: 1,611,495
## Columns: 4
## $ publish_date      <chr> "Wednesday, 25th of March, 2015", "Tuesday, 30th of …
## $ headline_category <chr> "opinion", "news", "news.politics.oireachtas", "opin…
## $ headline_text     <chr> "Renua's plan to publish Attorney General's advice m…
## $ news_provider     <chr> "Irish Times", "Irish Examiner", "RTE News", "RTE Ne…
```

Question 1

What are the earliest and latest articles from Irish Independent, irrespective of headline category? Please also sort the data according to the column publish_date in an ascending manner and display the last 5 records of the data.

Answer

- only publish_date and headline category are the selected columns because we already know the data is from Irish Independent and headline category is irrelevant
- before sorting, filter out the NA publish_date column
- arrange the data based on the publish_date in date type
- show earliest and latest using head() and tail()

Hide

```
# get Irish Independent's articles only
news_irish <- news %>%
  filter(news_provider == "Irish Independent") %>%
  filter(!is.na(publish_date)) %>%
  mutate(publish_date = dmy(publish_date)) %>%
  select(publish_date, headline_text) %>%
  arrange(publish_date)

# show earliest
head(news_irish,1)
```

```
## # A tibble: 1 × 2
##   publish_date headline_text
##   <date>       <chr>
## 1 1996-01-02   Dance Beat tunes up for Ladbroke
```

Hide

```
# show latest
tail(news_irish,1)
```

```
## # A tibble: 1 × 2
##   publish_date headline_text
##   <date>       <chr>
## 1 2021-06-29   Purpose-built Garda facility has no space for specialist units
```

Hide

```
# filter out NA date, change the data type of date, and sort asc based on date
sorted_news <- news %>%
  filter(!is.na(publish_date)) %>%
  mutate(publish_date = dmy(publish_date)) %>%
  arrange(publish_date)

# view data in ascending order
sorted_news
```

```
## # A tibble: 1,611,395 × 4
##    publish_date headline_category headline_text                    news_provider
##    <date>       <chr>             <chr>                            <chr>
##  1 1996-01-02   sport             Dance Beat tunes up for Ladbroke Irish Indepe…
##  2 1996-01-02   business          Jamont plans £5m investment to … Irish Examin…
##  3 1996-01-02   sport             Star of the Sea are on top of t… Irish Indepe…
##  4 1996-01-02   sport             Curran; Dillon brightest of 'St… TheJournal.ie
##  5 1996-01-02   sport             A larger-than-life personality … Irish Examin…
##  6 1996-01-02   sport             From shy hero to Jason of arrog… Irish Times
##  7 1996-01-02   sport             Whelan makes a point at Coventry RTE News
##  8 1996-01-02   sport             O'Halloran comes back to take t… Irish Times
##  9 1996-01-02   sport             Redknapp angry over goalkeeping… TheJournal.ie
## 10 1996-01-02   sport             Collymore hits old friends hard  Irish Times
## # ℹ 1,611,385 more rows
```

Hide

```
# display last 5 records(5 latest data)
tail(sorted_news, 5)
```

```
## # A tibble: 5 × 4
##   publish_date headline_category             headline_text          news_provider
##   <date>       <chr>                         <chr>                  <chr>
## 1 2021-06-30   business.commercial-property  Luxury rental company… <NA>
## 2 2021-06-30   opinion.letters               Polish insult to Holo… Irish Examin…
## 3 2021-06-30   news.politics.oireachtas      Government decision t… TheJournal.ie
## 4 2021-06-30   business.markets              European shares slide… Irish Times
## 5 2021-06-30   news.world.us                 Actor Allison Mack se… Irish Examin…
```

Question 2

How many unique headline_category values are there in the data file? Please consider variations (e.g.: capitalisation, potential inconsistencies) of the headline_category values, when counting them.

How many news category articles contain either the keyword, “Ireland”, “Irish”, “US”, or “USA” along with year digits from 2000 to 2024 in headline_text? For example, you need to search and count articles containing both “Ireland” and the year digits, or containing both “Irish” and the year digits, and so on.

Answer

During data exploration, headline category separated by “_” instead of “.” is found and contains NA data.

Hide

```
capital_exist <- news %>%
  filter(str_detect(headline_category, "[A-Z]+"))

underscore_exist <- news %>%
  filter(str_detect(headline_category, "_"))

na_exist <- news %>%
  filter(is.na(headline_category))

capital_exist
```

```
## # A tibble: 4 × 4
##   publish_date                    headline_category headline_text  news_provider
##   <chr>                           <chr>             <chr>          <chr>
## 1 Friday, 13th of April, 2007     OPINION.LETTERS   Merging of yo… RTE News
## 2 Monday, 12th of October, 2015   business.MARKETS  European mark… Irish Examin…
## 3 Friday, 25th of September, 1998 NEWS              Bishops issue… Irish Examin…
## 4 Saturday, 31th of October, 1998 Opinion.Letters   Speaking For … Irish Indepe…
```

Hide

```
underscore_exist
```

```
## # A tibble: 6 × 4
##   publish_date                     headline_category headline_text news_provider
##   <chr>                            <chr>             <chr>         <chr>
## 1 Saturday, 09th of February, 2013 lifestyle_travel… Sights for b… Irish Examin…
## 2 Friday, 25th of January, 2019    lifestyle_fashion Haute Coutur… RTE News
## 3 Saturday, 30th of January, 2010  culture_books     Thumping goo… RTE News
## 4 Saturday, 08th of August, 2020   opinion_letters   Here's to th… Irish Times
## 5 Monday, 25th of August, 2014     business_economy  GDP to climb… RTE News
## 6 Friday, 13th of May, 2016        news_ireland      Skellig Mich… Irish Times
```

Hide

```
na_exist
```

```
## # A tibble: 191 × 4
##    publish_date                     headline_category headline_text news_provider
##    <chr>                            <chr>             <chr>         <chr>
##  1 Tuesday, 16th of July, 2013      <NA>              Merkel call … Irish Examin…
##  2 Monday, 17th of June, 2013       <NA>              Quinn warns … RTE News
##  3 Monday, 30th of December, 2013   <NA>              Almost half … Irish Examin…
##  4 Tuesday, 22th of October, 2013   <NA>              Facebook lif… RTE News
##  5 Saturday, 26th of April, 2014    <NA>              Q&A: Why spe… Irish Examin…
##  6 Thursday, 21th of November, 20… <NA>              British comp… TheJournal.ie
##  7 Saturday, 03th of May, 2014      <NA>              Ireland: pri… Irish Times
##  8 Monday, 30th of March, 2015      <NA>              World leader… RTE News
##  9 Saturday, 22th of February, 20… <NA>              WhatsApp: wh… Irish Times
## 10 Tuesday, 19th of November, 2013 <NA>              'Selfie' bea… Irish Examin…
## # ℹ 181 more rows
```

Therefore, do counting after replacing “_” with “.” , lower case them and filter out the NA headline_category.

Hide

```
uniq_category <- news %>%
  # transform headline_category column to lower case
  mutate(headline_category = tolower(headline_category)) %>%
  # replace all underscores to dots in headline_category column
  mutate(headline_category = str_replace_all(headline_category,"_",".")) %>%
  # filter na data
  filter(!is.na(headline_category)) %>%
  # get unique rows only
  distinct(headline_category)

nrow(uniq_category)
```

```
## [1] 103
```

Filter rows that are from “news” category and contain “Ireland”, “Irish”, “US”, or “USA” and 2000 to 2024 in headline_text using pure text comparison

Hide

```
news_categories <- news %>%
  filter(headline_category == "news") %>%
  filter(str_detect(headline_text, "US|USA|Irish|Ireland")) %>%
  filter(str_detect(headline_text, "2000|2001|2002|2003|2004|2005|2006|2007|2008|200
         9|2010|2011|2012|2013|2014|2015|2016|2017|2018|2019|2020|2021"))

nrow(news_categories)
```

```
## [1] 229
```

Question 3

Please display the top 10 headline categories with the largest number of articles published on Monday throughout the years. Then, draw a chart showing the total number of articles for the top 10 headline categories (as identified previously) for each year. What can you observe? Please discuss the chart and your findings.

Answer

- filter such that it is Monday, non-NA(like Q2),
- transform headline_category, lower case and inconsistency such as “_” (like Q2)
- group by headline_category
- calculate count for each group
- descending order based on count

Hide

```
top_10 <- news %>%
  # Monday only
  filter(str_detect(publish_date, "Monday")) %>%
  # filter NA headline category
  filter(!is.na(headline_category)) %>%
  # transform headline_category column to lower case
  mutate(headline_category = tolower(headline_category)) %>%
  # replace all underscores to dots in headline_category column
  mutate(headline_category = str_replace_all(headline_category,"_",".")) %>%
  # group by headline category to get the count for each group
  group_by(headline_category) %>%
  # display headline_category with respective count
  summarise(count = n_distinct(headline_text, na.rm = TRUE)) %>%
  ungroup() %>%
  # arrange in desc order
  arrange(desc(count)) %>%
  head(10)

top_10
```

```
## # A tibble: 10 × 2
##    headline_category count
##    <chr>             <int>
##  1 news              83338
##  2 sport             38876
##  3 opinion.letters   11336
##  4 business           9793
##  5 opinion            6657
##  6 sport.soccer       6267
##  7 news.ireland       5246
##  8 news.law           3745
##  9 news.politics      3436
## 10 sport.rugby        3382
```

For second part of the Question 3, additionally,

- filter such that it has date and not NA
- filter such that it is from the top 10 headline_category above
- transform date to year into another column
- group by both headline_category and year

Hide

```
top_10_over_years <- news %>%
  # filter so that date is not unavailable
  filter(!is.na(publish_date)) %>%
  # filter the top 10 headline_category
  filter(headline_category %in% top_10$headline_category) %>%
  mutate(headline_category = tolower(headline_category)) %>%
  mutate(headline_category = str_replace_all(headline_category,"_",".")) %>%
  # change to Date type
  mutate(publish_date = dmy(publish_date)) %>%
  # get year from Date and put into 'year' column
  mutate(year = format(publish_date, "%Y")) %>%
  # additionally, group by year
  group_by(headline_category, year) %>%
  summarise(count = n_distinct(headline_text, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(desc(count))
```

```
## `summarise()` has grouped output by 'headline_category'. You can override using
## the `.groups` argument.
```

Hide

```
# show top 10 each year
top_10_over_years
```

```
## # A tibble: 195 × 3
##    headline_category year  count
##    <chr>             <chr> <int>
##  1 news              2001  45155
##  2 news              2003  40715
##  3 news              2008  37302
##  4 news              2009  36306
##  5 news              2004  36124
##  6 news              2010  36109
##  7 news              2002  35776
##  8 news              2007  35707
##  9 news              2006  35034
## 10 news              2005  33388
## # ℹ 185 more rows
```

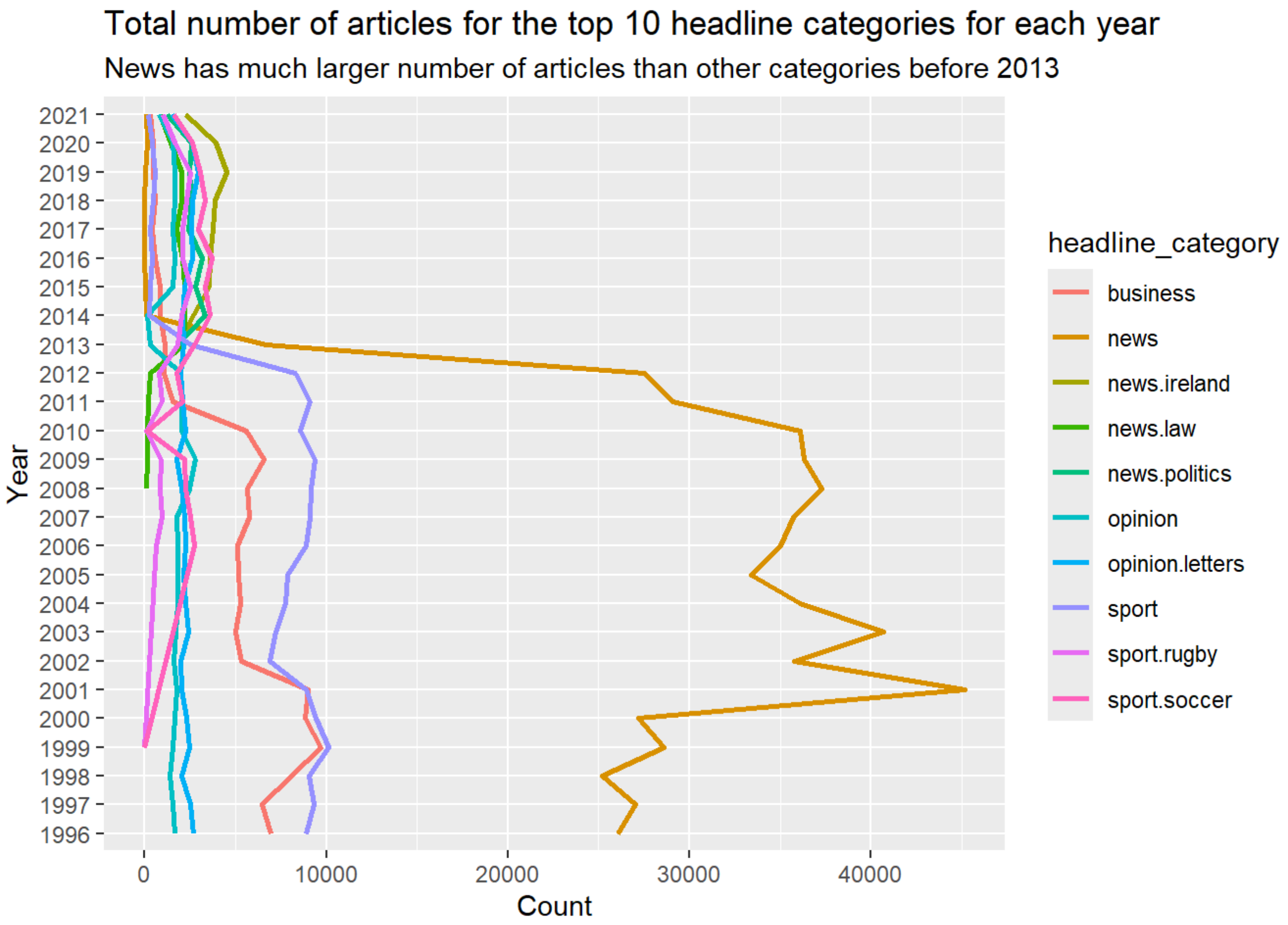
Plot line graph

- The x and y axis are flipped because if x-axis is 'Year', the range of the 'Year' is too large and causes the number to be overlapped and hard to see

Hide

```
# plot
top_10_over_years %>%
  ggplot(aes(x = year, y = count, group = headline_category, color=headline_categor
         y)) +
  geom_line(size=1) +
  xlab("Year") +
  ylab("Count") +
  labs(title="Total number of articles for the top 10 headline categories for each y
         ear", subtitle = "News has much larger number of articles than other catego
         ries before 2013") +
  coord_flip()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Total number of articles for the top 10 headline categories for each year

News has much larger number of articles than other categories before 2013

Findings

- It can be seen as 3 levels number of articles

   1. News has much larger number of articles than others before 2013

   2. Sport and Business maintained roughly between 5000 to 10000 articles before 2010(inclusive) and before 2012(inclusive) respectively
   3. All the other headline categories have below 5000 articles over the years
- There are 3 obvious lines that only have articles since 1999(sport.rugby, sport.soccer) and 2008(news.law)

Question 4

Compute the total number of articles for each headline category and news provider. Then, use a single R function/command to display the statistical information, i.e., Min, Max, and Mean, of the total number of articles (as computed previously) for each news provider. Note: You can use multiple functions/commands to get the desired pre-processed data table, but when you compute and display the statistical information, you need to use a single R function/command.

Answer

- “for each headline category and news provider” indicates group by headline_category and news_provider
- calculate count for each group and put result into a new column

Hide

```
articles_cate_prov <- news %>%
  # filter out NA category and provider
  filter(!is.na(headline_category) & !is.na(news_provider)) %>%
  # for each headline category and news provider
  group_by(headline_category, news_provider) %>%
  # calculate count
  summarise(count = n_distinct(headline_text, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'headline_category'. You can override using
## the `.groups` argument.
```

Hide

```
# show data in desc order
articles_cate_prov %>% arrange(desc(count))
```

```
## # A tibble: 521 × 3
##    headline_category news_provider      count
##    <chr>             <chr>              <int>
##  1 news              Irish Times       199795
##  2 news              Irish Examiner    142198
##  3 news              RTE News          114278
##  4 news              TheJournal.ie      85351
##  5 sport             Irish Times        53777
##  6 sport             Irish Examiner     38129
##  7 business          Irish Times        37286
##  8 sport             RTE News           30783
##  9 news              Irish Independent  28801
## 10 business          Irish Examiner     26832
## # ℹ 511 more rows
```

- use aggregate() function to cast a summary() onto each news provider group

Hide

```
# display statistical info using aggregate() and pass in summary
summary_data <- aggregate(articles_cate_prov$count, list(articles_cate_prov$news_pro
         vider), summary)
# cant be displayed
summary_data
```

```
##             Group.1      x.Min.   x.1st Qu.    x.Median      x.Mean   x.3rd Qu.
## 1    Irish Examiner      1.0000    296.0000    776.0000   3683.4667   2454.0000
## 2 Irish Independent      1.0000     59.5000    172.0000    762.0291    511.0000
## 3       Irish Times      1.0000    423.5000   1093.0000   5196.3077   3455.7500
## 4          RTE News      1.0000    206.0000    628.0000   2894.5701   1855.5000
## 5     TheJournal.ie     19.0000    188.0000    510.5000   2284.6176   1506.0000
##        x.Max.
## 1 142198.0000
## 2  28801.0000
## 3 199795.0000
## 4 114278.0000
## 5  85351.0000
```

Hide

```
# parse it into dataframe so it can be displayed
summary_data <- do.call(data.frame, summary_data)
summary_data
```

```
##             Group.1 x.Min. x.1st.Qu. x.Median    x.Mean x.3rd.Qu. x.Max.
## 1    Irish Examiner      1     296.0    776.0 3683.4667   2454.00 142198
## 2 Irish Independent      1      59.5    172.0  762.0291    511.00  28801
## 3       Irish Times      1     423.5   1093.0 5196.3077   3455.75 199795
## 4          RTE News      1     206.0    628.0 2894.5701   1855.50 114278
## 5     TheJournal.ie     19     188.0    510.5 2284.6176   1506.00  85351
```

Question 5

Please compute the total number of articles for each headline category, news provider, and the day of the week. Then, compute the average number of articles for each news provider and the day of the week, based on the total number of articles computed previously. After that, please display the day of the week with the highest average number of articles for each provider. The output data should be structured in the following format.

Answer

- filter out NA category, publish date and provider
- get weekday label by using wday onto date-converted(dym() function) publish_date
- for each headline category and news provider(group by), calculate count(n_distinct)
    - n_distinct is used here because same article might be published more than once
- display in descending order based on count

Hide

```
articles_cate_prov_day <- news %>%
  # filter out NA category, publish date and provider
  filter(!is.na(headline_category) &
           !is.na(news_provider) &
           !is.na(publish_date)) %>%
  # get weekday label by using wday onto date-converted publish_date
  mutate(weekday = wday(dmy(publish_date), label=TRUE)) %>%
  # for each headline category and news provider
  group_by(headline_category, news_provider, weekday) %>%
  # calculate count
  summarise(count = n_distinct(headline_text, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(desc(count))
```

```
## `summarise()` has grouped output by 'headline_category', 'news_provider'. You
## can override using the `.groups` argument.
```

Hide

```
articles_cate_prov_day
```

```
## # A tibble: 3,359 × 4
##    headline_category news_provider  weekday count
##    <chr>             <chr>          <ord>   <int>
##  1 news              Irish Times    Sat     35898
##  2 news              Irish Times    Wed     33300
##  3 news              Irish Times    Thu     33006
##  4 news              Irish Times    Tue     31511
##  5 news              Irish Times    Fri     31428
##  6 news              Irish Times    Mon     29258
##  7 news              Irish Examiner Sat     25422
##  8 news              Irish Examiner Thu     23763
##  9 news              Irish Examiner Wed     23476
## 10 news              Irish Examiner Tue     22738
## # ℹ 3,349 more rows
```

“Then, compute the average number of articles for each news provider and the day of the week, based on the total number of articles computed previously” indicates to compute average across categories for each news provider and day of the week

- group by news provider and day of the week
- sum up the count calculated in previous ques, get number of articles and calculate average using sum of count/number of articles

Hide

```
articles_cate_prov_day <- articles_cate_prov_day %>%
  group_by(news_provider, weekday) %>%
  summarise(total_articles = sum(count, na.rm = TRUE),
            total_categories = n(),
            average = total_articles/total_categories) %>%
  ungroup() %>%
  arrange(desc(average)) %>%
  select(news_provider, weekday, average)
```

```
## `summarise()` has grouped output by 'news_provider'. You can override using the
## `.groups` argument.
```

- use pure string to set up the column names
- group by provider
- filter out the average is not the max among the days in the week
- floor the average to get integer of the average

Hide

```
articles_cate_prov_day %>%
  group_by(news_provider) %>%
  filter(average == max(average, na.rm=TRUE)) %>%
  summarise("News provider"= news_provider,
            "The day of week (with the highest
average number of articles)" = weekday,
           "The highest average
number of articles"= floor(average)) %>%
  ungroup() %>%
  select("News provider", "The day of week (with the highest
average number of articles)", "The highest average
number of articles")
```

```
## # A tibble: 5 × 3
##   `News provider`   The day of week (with the highest\n…¹ The highest average\…²
##   <chr>             <ord>                                                 <dbl>
## 1 Irish Examiner    Fri                                                     660
## 2 Irish Independent Fri                                                     142
## 3 Irish Times       Fri                                                     916
## 4 RTE News          Fri                                                     528
## 5 TheJournal.ie     Fri                                                     403
## # ℹ abbreviated names:
## #   ¹`The day of week (with the highest\naverage number of articles)`,
## #   ²`The highest average\nnumber of articles`
```

Question 6

Answer

- filter outNA publish date
- change publish date to date type so that can do comparison
- do comparison to get 2019 and 2020 data
- add Period column values by doing comparison as well

Hide

```
add_period <- news %>%
  filter(!is.na(publish_date)) %>%
  mutate(date = dmy(publish_date)) %>%
  filter(date >= '2019-01-01' & date <= '2020-12-31') %>%
  mutate(Period = case_when(date >='2019-01-01'& date<='2019-03-31' ~ "Period 1",
                            date >='2019-04-01'& date<='2019-06-30' ~ "Period 2",
                            date >='2019-07-01'& date<='2019-09-30' ~ "Period 3",
                            date >='2019-10-01'& date<='2019-12-31' ~ "Period 4",
                            date >='2020-01-01'& date<='2020-03-31' ~ "Period 5",
                            date >='2020-04-01'& date<='2020-06-30' ~ "Period 6",
                            date >='2020-07-01'& date<='2020-09-30' ~ "Period 7",
                            date >='2020-10-01'& date<='2020-12-31' ~ "Period 8",))

add_period %>% select(publish_date, date, Period)
```

```
## # A tibble: 112,643 × 3
##    publish_date                        date       Period
##    <chr>                               <date>     <chr>
##  1 Wednesday, 05th of June, 2019      2019-06-05 Period 2
##  2 Sunday, 22th of November, 2020     2020-11-22 Period 8
##  3 Thursday, 25th of July, 2019       2019-07-25 Period 3
##  4 Monday, 01th of June, 2020         2020-06-01 Period 6
##  5 Monday, 15th of April, 2019        2019-04-15 Period 2
##  6 Wednesday, 02th of September, 2020 2020-09-02 Period 7
##  7 Wednesday, 04th of November, 2020  2020-11-04 Period 8
##  8 Tuesday, 21th of May, 2019         2019-05-21 Period 2
##  9 Wednesday, 07th of October, 2020   2020-10-07 Period 8
## 10 Monday, 10th of February, 2020     2020-02-10 Period 5
## # ℹ 112,633 more rows
```

- filter out NA headline_category
- filter such that only top_10 computed in Question 3 are included
- resolve inconsistencies in headline_category similar to Question 2
- “by period and headline category” indicates group by both Period and headline_category
- calculate number of articles using n_distinct()

Hide

```
top10_during_periods <- add_period %>%
  # filter na data
  filter(!is.na(headline_category)) %>%
  # filter the top 10 headline_category
  filter(headline_category %in% top_10$headline_category) %>%
  # lowercase the headline category because inconsistencies
  mutate(headline_category = tolower(headline_category)) %>%
  # replace all underscores to dots in headline_category column
  mutate(headline_category = str_replace_all(headline_category,"_",".")) %>%
  group_by(headline_category, Period) %>%
  summarise(total_articles = n_distinct(headline_text)) %>%
  ungroup() %>%
  arrange(desc(total_articles))
```

```
## `summarise()` has grouped output by 'headline_category'. You can override using
## the `.groups` argument.
```

Hide

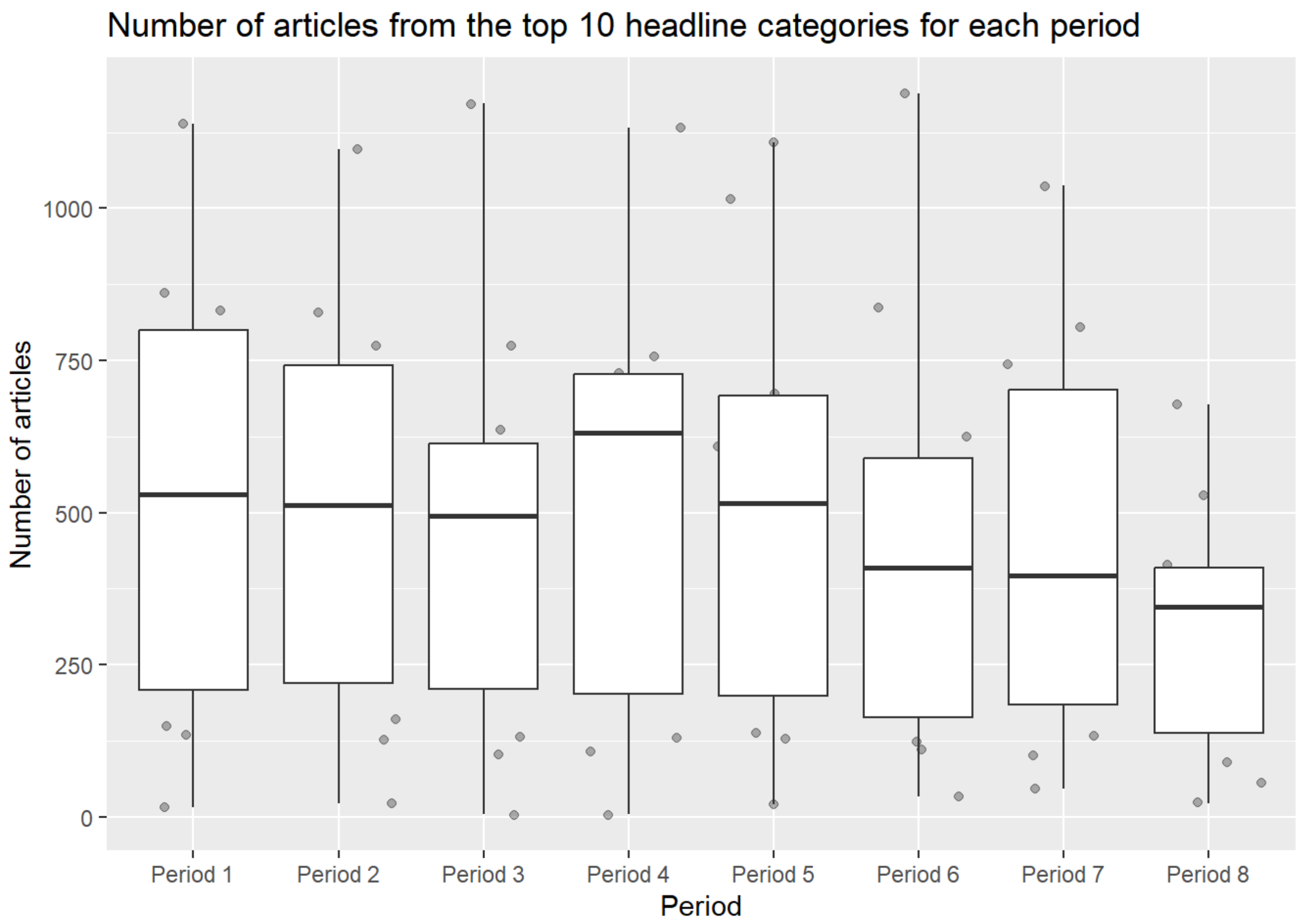
```
top10_during_periods
```

```
## # A tibble: 80 × 3
##    headline_category Period   total_articles
##    <chr>             <chr>             <int>
##  1 news.ireland      Period 6           1189
##  2 news.ireland      Period 3           1172
##  3 news.ireland      Period 1           1139
##  4 news.ireland      Period 4           1133
##  5 news.politics     Period 5           1108
##  6 news.ireland      Period 2           1097
##  7 news.ireland      Period 7           1037
##  8 news.ireland      Period 5           1015
##  9 sport.soccer      Period 1            861
## 10 opinion.letters   Period 6            837
## # ℹ 70 more rows
```

- plot boxplot with jitter

Hide

```
top10_during_periods %>%
  ggplot(aes(x = Period, y = total_articles, group = Period)) +
  geom_jitter(alpha = 0.3) +
  geom_boxplot() +
  ylab("Number of articles") +
  labs(title="Number of articles from the top 10 headline categories for each perio
       d")
```



Question 7

Please sample 1% of the data, conduct the text pre-processing for the values of the headline_text column in the sampled data, and display a portion (the first few columns and rows) of a document-term matrix generated. Then, draw a plot showing the top 10 most frequent words where the x-axis represents the frequency of words and the y-axis represents the words themselves. Additionally, generate a word cloud.

Answer

- Tokenization on every row

Hide

```
# set seed for random sample
set.seed(32061412)

# 1% of the data
one_percent_sample <- news %>%
  sample_frac(0.01)

# apply tokenization for each row
tokenised_sample <-
  lapply(one_percent_sample$headline_text, function(line) {
    unlist(tokenize_words(line))
  })

# check the data
tokenised_sample[1]
```

```
## [[1]]
##  [1] "the"     "day"     "the"     "battle"  "of"      "the"     "reds"   
##  [8] "was"     "all"     "squared"
```

- Create Corpus object that provides structured and efficient framework for text analysis

Hide

```
one_percent_sample$doc_id <- seq(nrow(one_percent_sample))
one_percent_sample <- one_percent_sample %>%
  select(headline_text, doc_id)

names(one_percent_sample)[names(one_percent_sample) == 'headline_text'] <- 'text'
one_percent_sample
```

```
## # A tibble: 16,115 × 2
##    text                                                          doc_id
##    <chr>                                                          <int>
##  1 The day the battle of the reds was all 'squared'                   1
##  2 CIE denies allegation                                              2
##  3 Advertising weapon                                                 3
##  4 Feast at Mullingar                                                 4
##  5 Armed robbers raid shop in Co Antrim                               5
##  6 Draft treaty on banning cluster bombs welcomed                     6
##  7 Ben Arfa's moment of magic helps keep Newcastle's dream alive      7
##  8 Actor says Point cinema will give huge lift to area                8
##  9 Bray Head gets special status after 15-year campaign               9
## 10 All passengers but one survive crash                              10
## # ℹ 16,105 more rows
```

Hide

```
# Create your DataFrameSource
sample_source <- DataframeSource(one_percent_sample)

# Create a Corpus
sample_corpus <- Corpus(sample_source)

# Check corpus
sample_corpus
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
```

- Remove stop words, punctuation, numbers and spaces and case normalisation

Hide

```
# remove stop words
sample_corpus <- tm_map(sample_corpus, removeWords, stopwords("en"))

# remove punctuation
sample_corpus <- tm_map(sample_corpus, removePunctuation)

# remove all numbers
sample_corpus <- tm_map(sample_corpus, removeNumbers)

# remove redundant spaces
sample_corpus <- tm_map(sample_corpus, stripWhitespace)

# case normalisation
sample_corpus <- tm_map(sample_corpus, content_transformer(tolower))
```

- Stemming

Hide

```
# perform stemming to reduce inflected and derived words to their root form
sample_stem <- tm_map(sample_corpus, stemDocument)

# Inspect the stemmed corpus
# inspect(sample_stem[1])
```

- Create document-term matrix

Hide

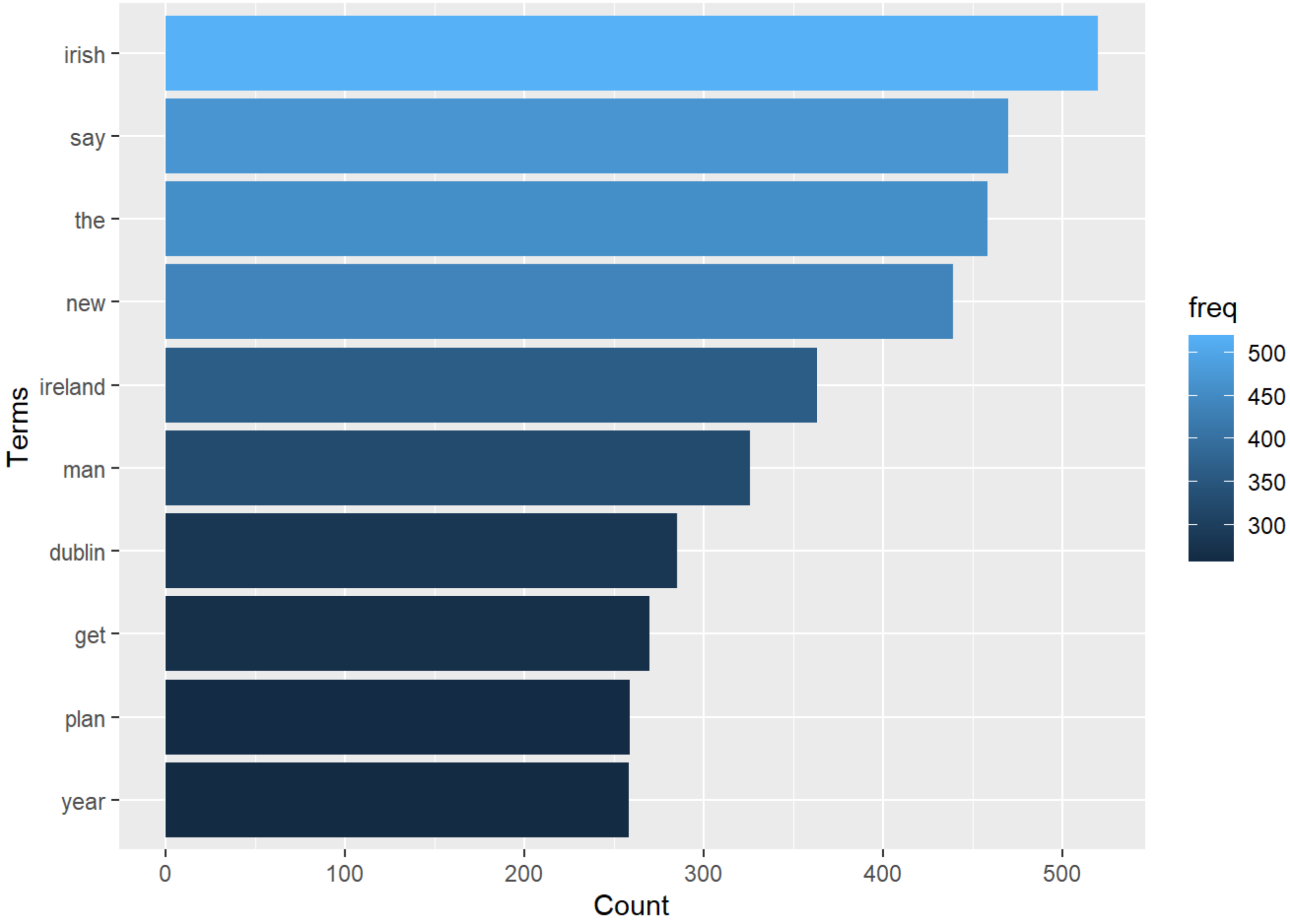
```
#  Create a matrix which its rows are the documents and columns are the words.
sample_dtm <- DocumentTermMatrix(sample_stem)

# check dtm
inspect(sample_dtm)
```

```
## <<DocumentTermMatrix (documents: 1, terms: 12639)>>
## Non-/sparse entries: 12639/0
## Sparsity           : 0%
## Maximal term length: 19
## Weighting          : term frequency (tf)
## Sample             :
##          Terms
## Docs      dublin get ireland irish man new plan say the year
##   1:16115    285 270     363   520 326 439  259 470 458  258
```

- Plot top 10 used words and their frequencies

Hide

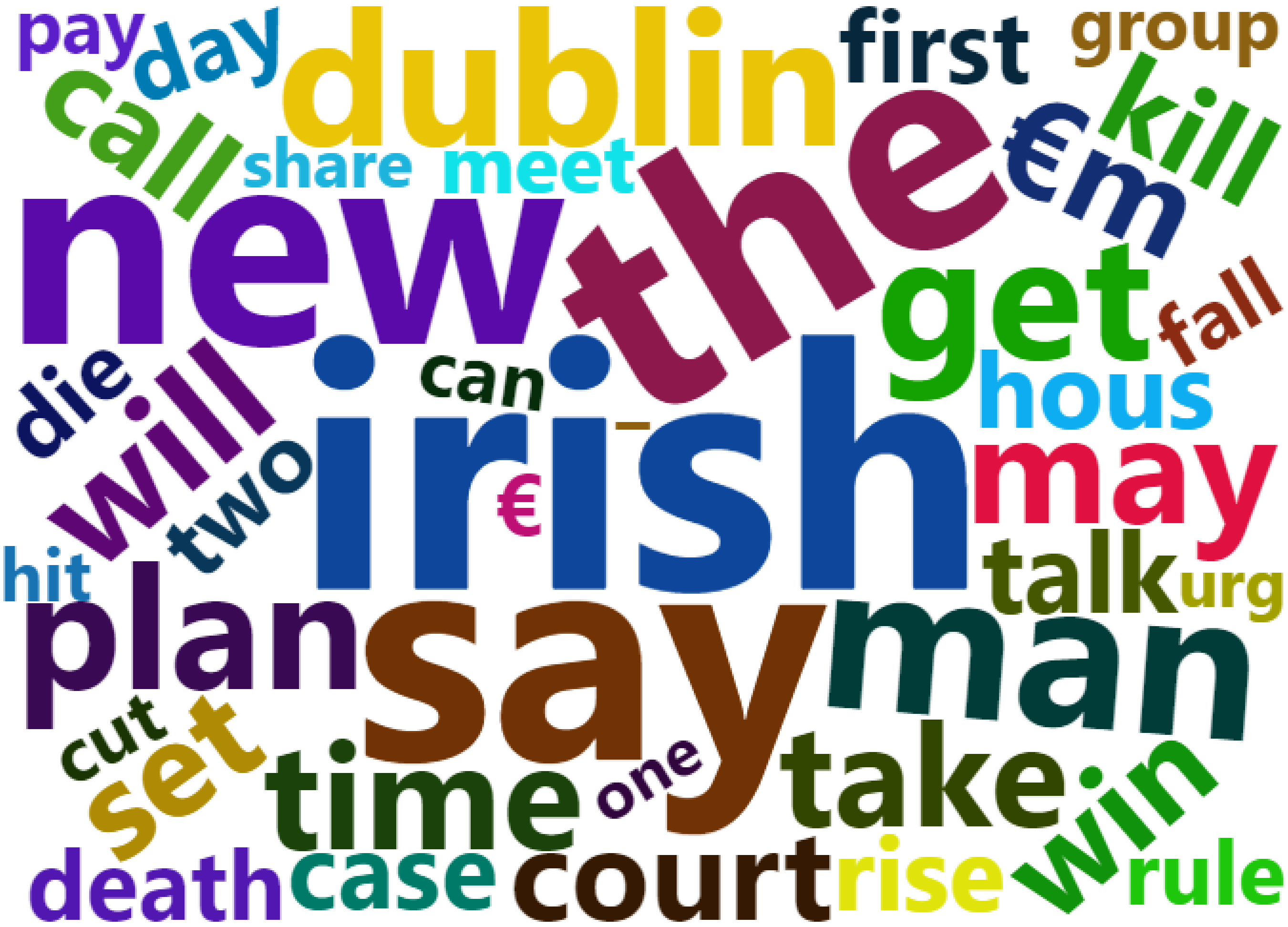
```
# Convert the DocumentTermMatrix into a regular matrix object and calculate term frequencies
term_freq<- colSums(as.matrix(sample_dtm))

# Create a dataframe
df<- data.frame(term = names(term_freq), freq = term_freq)

# Filter terms with a frequency of at least 100
df <- df %>%
  filter(freq>=100) %>%
  arrange(desc(freq))

# Select the top 10 frequent words
df_plot<- df %>%
  top_n(10, freq)

# Plot word frequency
ggplot(df_plot, aes(x = fct_reorder(term, freq), y = freq, fill = freq)) +
  geom_bar(stat = "identity") +
  xlab("Terms") +
  ylab("Count") +
  coord_flip()
```



Wordcloud

Hide

```
wordcloud2(df, color = "random-dark", backgroundColor = "white")
```



Question 8

Answer

Hide

```
irish_times_performance <- news %>%
  filter(!is.na(publish_date)) %>%
  mutate(date = dmy(publish_date)) %>%
  # filter(date >= '2015-01-01' & date <= '2015-12-31') %>%
  mutate(Year = year(date)) %>%
  mutate(Week = week(date)) %>%
  filter(news_provider == "Irish Times") %>%
  group_by(Year ,Week) %>%
  summarise(count = n_distinct(headline_text, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

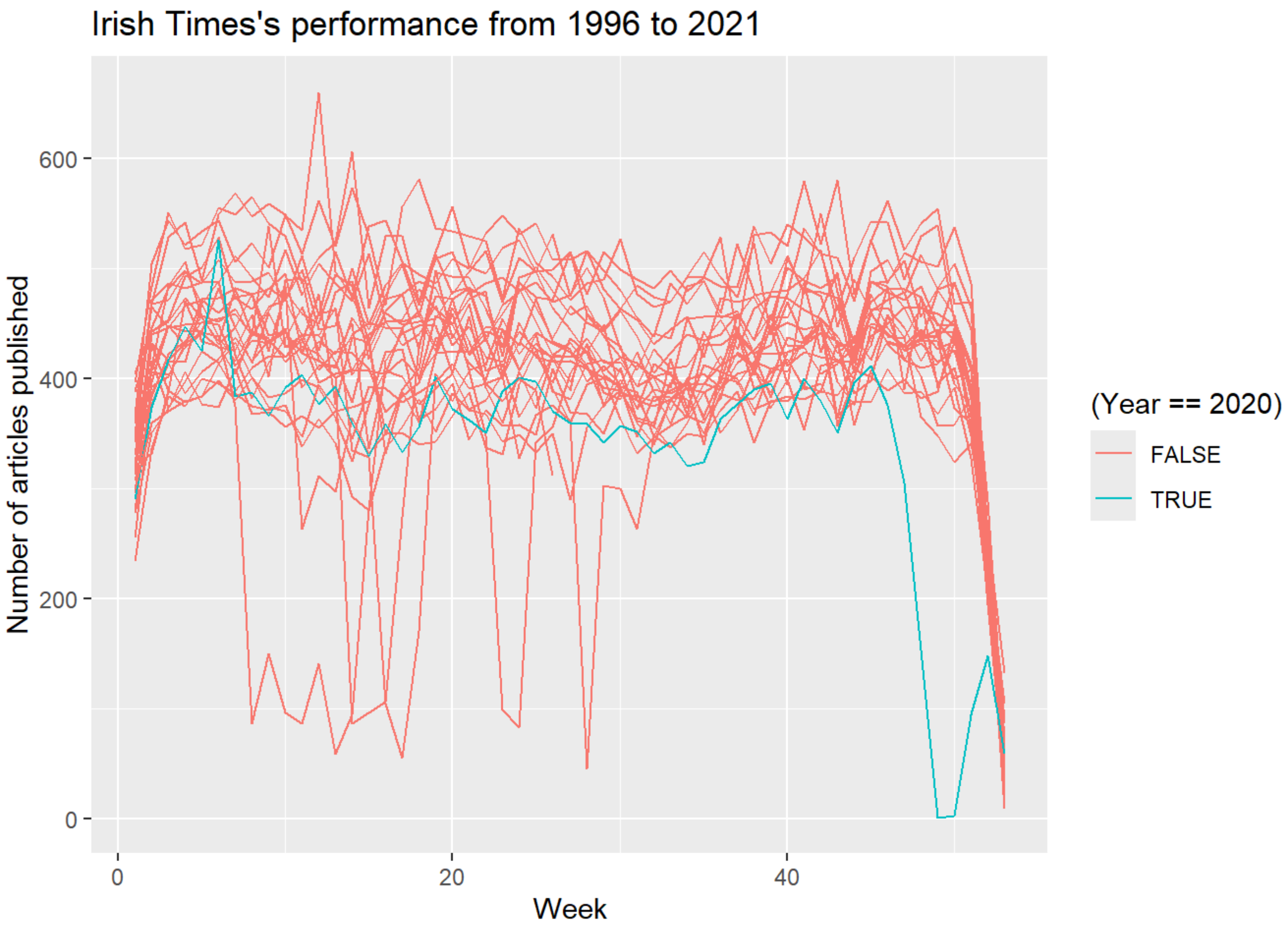
Hide

```
head(irish_times_performance)
```

```
## # A tibble: 6 × 3
##    Year  Week count
##   <dbl> <dbl> <int>
## 1  1996     1   234
## 2  1996     2   336
## 3  1996     3   384
## 4  1996     4   375
## 5  1996     5   400
## 6  1996     6   395
```

Hide

```
irish_times_performance %>%
  ggplot(aes(x = Week, y = count, group = Year, color=(Year==2020))) +
  geom_line() +
  xlab("Week") +
  ylab("Number of articles published") +
  labs(title="Irish Times's performance from 1996 to 2021")
```



Discussion

- If news provider’s performance is based on number of articles within certain time frame then plotting a graph of number of articles vs specific year can be used to observe the news provider’s performance in that year.
- The graph can also be used to monitor the performance of the provider company throughout the year so that downhill performance can be detected early, find out the existing problems within the company and provide corresponding solutions without further deterioration
- Not only single year can be plotted, but also from the year that the company has started operation. By plotting the number of articles published across the year since the year of operation, the company can monitor whether itself has evolved and grown larger or even whether or not the company has been earning money.
- The graph above shows that taking one year of performance and compare to performances in other years. From this graph, we can know that in 2020 the performance is deteriorating and almost lowest among all the other performances, corresponding actions and countermeasures can be come up with by the company directors to prevent their companies from further worsening.