For nearly 50 years, the UK-based independent magazine New Internationalist is bringing us impactful, global stories from voices often unheard. The magazine is dedicated to socially conscious journalism and publishing and is now inviting the public to support their mission and become one of their co-owners. In this short post, I will be looking at the topics covered in the New Internationalist by scraping their website for article titles of past issues and visualizing the results in a word cloud, all using R.

Rationale and disclaimer

The New Internationalist if currently running a fundraiser where you can become a co-owner of the magazine to save its survival in these challenging times for independent journalism. While this fundraiser is the focus of this post, this has certainly been its motivation!

New Internationalist logo.

First a quick disclaimer, though, so we area ll on the same page: I am in no way affiliated with the New Internationalist nor do I have any financial interest in the organization. I am, however, a subscriber of the magazine and truly believe in their approach to journalism, their ethics, their storytelling and their record of supporting issues I deeply care about. I think, it is immensely important to support independent journalism – now more than ever! That's why I decided to have a quick look at which topics the New Internationalist has been covering in the last decade. And maybe, this might convince you to check out their current fundraiser and read up on their mission and goals for the future. With just one week left in their campaign, they are about one-third shy from their ultimate funding goal of 350,000 GBP. So, let me convince you that the topics they're covering make it worthwhile supporting them, if you can.

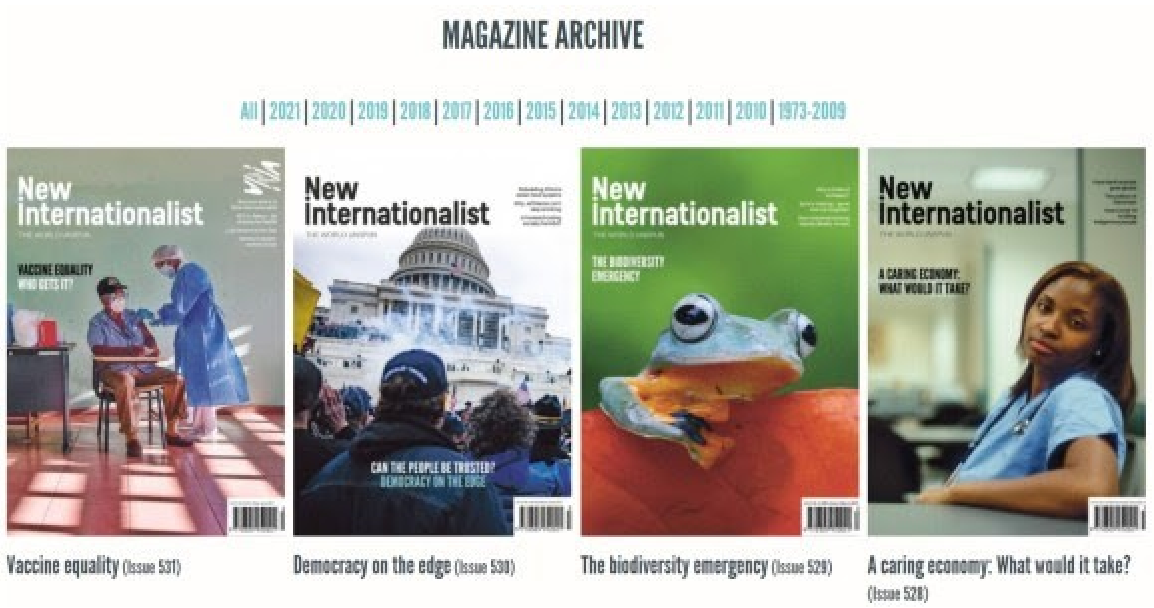
For that, I will be using the R package *rvest* to iteratively scrape their website for all article headlines of the past 11 years that are available online. Once I have all the article headlines, I will visualize them in a word cloud using the *wordcloud2* package.

So, let's get started.

Scraping the New Internationalist website for article headlines

The first step is to scrape the New Internationalist online magazine archive for article headlines. Looking at their website structure, I decided to use the three-step approach to achieve this:

1. Scrape the links from their `/magazines` subpage to the overview of the respective issues for that particular year.
2. Follow these links and scrape the links for each issue's table of content.
3. Follow that link and scrape the article headlines for each issue.


MAGAZINE ARCHIVE
All | 2021 | 2020 | 2019 | 2018 | 2017 | 2016 | 2015 | 2014 | 2013 | 2012 | 2011 | 2010 | 1973-2009
New Internationalist
VACCINE EQUALITY WHO GETS IT?
New Internationalist
CAN THE PEOPLE BE TRUSTED? DEMOCRACY ON THE EDGE
New Internationalist
THE BIODIVERSITY EMERGENCY
New Internationalist
A CARING ECONOMY: WHAT WOULD IT TAKE?
Vaccine equality (Issue 531)
Democracy on the edge (Issue 530)
The biodiversity emergency (Issue 529)
A caring economy: What would it take? (Issue 528)

Screenshot of the New Internationalist magazine archive.

Using the *rvest* package similar to the link-follow-then-scrape method described by Jerid Francom allows us to extract a certain CSS element of the website. Contrary to the Jerid's methods of using a for loop to iterate over all detected elements (i.e. links), I will use the `map()` function to serially apply the html scraping functions of *rvest*.

The CSS patterns to follow were identified using the Selector Gadget browser plugin (I am using the Chrome browser) and are stored in the respective `xxx_selector` variable. As mentioned earlier, I use a step-wise approach to extract the article headlines by first finding the page where all issues for a given year are located, then following the links to the table of content for each issue und finally, extracting the article headlines from all these pages. Thus is how it looks in code:

```
# Load libraries
#===================================

library(tidyverse)
library(stringr)
library(rvest)

# Set variables
#===================================

base_url = "https://newint.org"

year_url = paste0(base_url, "/magazines/")
year_selector = ".magazines-year a"
issue_selector = ".story-card__title a"
article_selector = "#block-views-magazine-articles-block a"

# Find available years
#===================================

newint_years =
    tibble(years_html = map(year_url, read_html)) %>%
    mutate(years_nodes = map(years_html, html_nodes, year_selector),
           year = map(years_nodes, html_text),
           years_url = map(years_nodes, html_attr, "href")) %>%
    unnest(c(year, years_url)) %>%
        #clean data formatting and filter for years after 2010
    mutate(year = str_trim(year)) %>%
    filter(year %in% 2010:2021) %>%
    mutate(years_url = paste0(base_url, years_url)) %>%
    select(year, years_url)

# Find all issues per year
#===================================

newint_issues =
    newint_years %>%
    mutate(issue_html = map(years_url, read_html),
           issue_nodes = map(issue_html, html_nodes, issue_selector),
           issue_title = map(issue_nodes, html_text),
           issue_url = map(issue_nodes, html_attr, "href")) %>%
    unnest(c(issue_title, issue_url)) %>%
    mutate(issue_url = paste0(base_url, issue_url)) %>%
    select(year, issue_title, issue_url)

# Find all article headlines
#===================================

newint_articles =
    newint_issues %>%
    mutate(article_html = map(issue_url, read_html),
           article_nodes = map(article_html, html_nodes,
article_selector),
           article_title = map(article_nodes, html_text)) %>%
    unnest(article_title) %>%
        # remove duplicate article titles
    group_by(article_title) %>%
    mutate(count = n()) %>%
    ungroup() %>%
    filter(count == 1) %>%
    select(year, issue_title, article_title)
```

We now have a data frame which contains the year, issue title and all unique article titles of the New Internationalist from 2010 to 2021. We had to remove duplicate article titles as to not skew the word count data to recurring themes like book, film or music reviews or columns which often have identical titles over several issues.

```
head(newint_articles)
```

```
# A tibble: 6 x 3
  year  issue_title      article_title

1 2021  Vaccine equality The nuts and bolts
2 2021  Vaccine equality How to end vaccine apartheid
3 2021  Vaccine equality A history of vaccines
```

```
4 2021  Vaccine equality Dreams of magic bullets
5 2021  Vaccine equality New Internationalist: the first 50 years – and
the next
6 2021  Vaccine equality 5 very good reasons to invest in New
Internationalist
```

Tokenizing the article titles and creating a word cloud

Before we make the word cloud, we have to tokenize the article titles using the *tidytext* package, which basically means that we split the complete titles into separate words. We also will remove numbers (think titles like *"7 great reasons why you should save independent journalism!"*) and so-called stop words (like "and", "or", "I" etc.) from the titles. After we have tokenized the titles, we also calculate the absolute frequency (i.e. number of occurrences) for each word. This frequency will then determine how large the respective word will be printed in the word cloud.

```
# Load libraries
#===================================

library(tidytext)

# Tokenize article titles
#===================================

newint_words =
    newint_articles %>%
    #remove numbers
    mutate(article_tidy = gsub("[[:digit:]]+", "", article_title)) %>%
    #tokenize text and remove stopwords
    unnest_tokens(word, article_tidy) %>%
    filter(!word %in% stop_words$word) %>%
    ungroup() %>%
        #calclulate frequencies
    group_by(word) %>%
    summarize(freq = n()) %>%
    ungroup()
```

This gives us a data frame with two columns, one for each unique word and one for the respective frequency. Let's look at the top ten:

```
head(newint_words[order(newint_words$freq, decreasing = TRUE), ], 10)
```

```
# A tibble: 10 x 2
   word       freq

 1 country      30
 2 profile      28
 3 climate      17
 4 world        15
 5 interview    14
 6 democracy    13
 7 people       13
 8 letter       12
 9 media        11
```

```
10 time          11
```

For readers of the New Internationalist it won’t come as a surprise that the words *country* and *profile* come in first and second, as most issues feature so-called *Country Profiles* in which a given country is characterized and some facts and figures as well as a assessment of freedom of press and the current political and human rights conditions is being presented.

However, issues like the *climate*, the state of *democracy* and the *media* as well as in depth *interviews* with activists and indigenous land defenders from around the globe are an integral part the New Internationalist’s reporting.

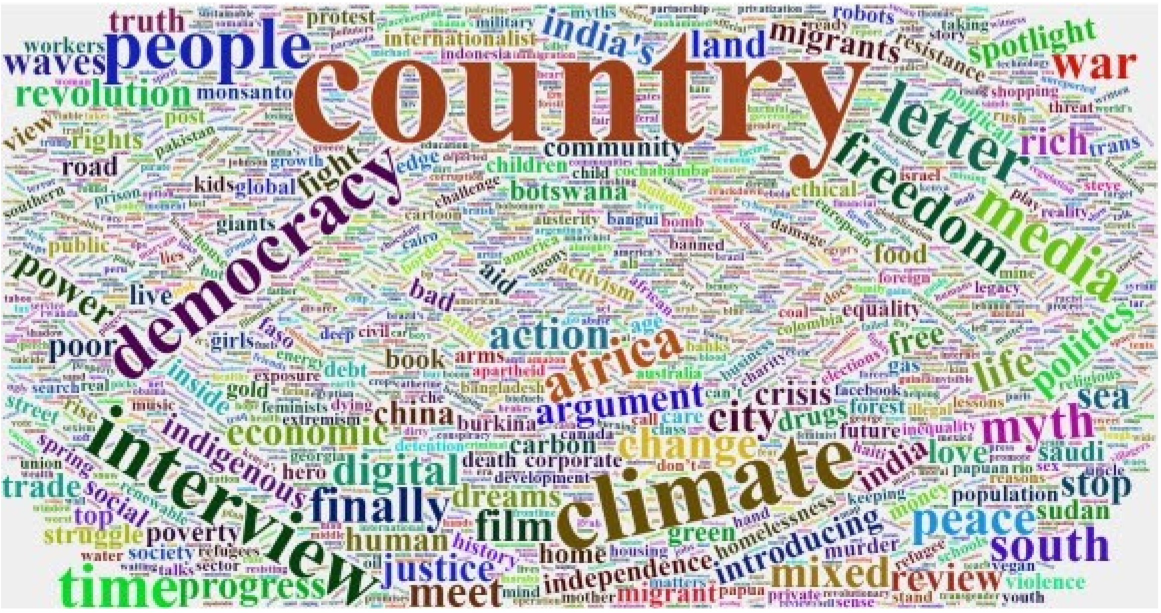
Finally, we can visualize everything in a beautiful word cloud.

```
# Load libraries
#==================================

library(wordcloud2)

# Create word cloud
#==================================

wordcloud2(data = newint_words,
           fontFamily = "serif",
           size = 1.6,
           backgroundColor = "#eeeee",
           color = "random-dark")
```


country
people
democracy
interview
climate
letter
freedom
media
war
truth
waves
revolution
india's
land
migrants
spotlight
rich
politics
life
sea
myth
africa
action
argument
economic
digital
finally
film
time
progress
meet
justice
human
indigenous
power
poor
city
crisis
change
peace
south
review
mixed
introducing
independence
stop
sudan
saudi
love
india
drugs
china
book
fight
community
botswana
children
global
road
rights
trade
struggle
poverty
dreams
carbon
home
green
population
free
equality
food
violence
migrant