STA141B_HW4: Web Scraping

Introduction

In this assignment, I scraped information from Stack Overflow and returned the results in a data frame format. I constructed many helper and main functions to complete this assignment. For each helper function, I will briefly describe the purpose and how it plays a role in the grand scheme of things. My main functions for this assignments entailed:

- Extracting information surrounding the question
- Extracting information surrounding the answers to each question
- Extracting information surrounding the comments to each answers for a question
- Combining all the question, answers, and comment information into one data frame
- Reading a page of results from search query
  - Extracting the questions based on that page
- Extracting the URL for next page of search query
- Reading a question page containing 50 questions, and extracting the question, answers, and comments info

In each "section", I provide the necessary functions I used in order to complete the task, and a brief description of my thought process. I provided more comments in the R script containing the functions if anything is unclear. I wrote a lot of functions so I know that it may be hard to digest all at once.

With that said, lets start from step 1.

Dependencies/Useragent/Source

```
library(RCurl)
library(XML)
library(httr)
library(rvest)
source("ghFuns.R")

# Need to specify user
useragent = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
 ↪  like Gecko) Chrome/113.0.0.0 Safari/537.36"
# Set WD
setwd("~/Documents/UCD Classes/22-23/Spring Q23 Classes/STA 141B/HW4")
# options(error = recover)
```

The first step was to load the proper packages. Because this is web scraping, I need to make HTTP GET requests from the website. In addition, I needed to include the required HTTP Request Headers, which in this case was the Accept, Cookies, and User Agent. The code for the cookies is not shown however I did

include it in my function, and I extracted it by simply doing readLines() from a txt file that I saved to my specific directory. After I had all the information, I put it into getForm().

Now, I will dive in to the main functions that I needed to create in the order of how they were listed on the assignment.

1.) Read questions from a page

Read_Page() description:

As I stated above for reading a page, I needed to grab the necessary request headers to put into getForm(). Now that I have the headers, the last thing I needed to have was a specific url. Because we are trying to read questions from a page, the url needed for an input was the url based on the search query. Now, if I plug everything into getForm() with the proper parameters in the correct location, I will get the raw output of the HTML file. However, this is not the format we want it in as we want to perform XPath operations later on to extract specific nodes, therefore we need to parse the file into the actual HTML source code. I also included rawToChar before parsing into HTML because I ran into an error with unicode, as mentioned in the Piazza Post @367.

```
# Read a page of results based on query and parse into HTML structure
page =
 ↪  read_page("https://stackoverflow.com/questions/tagged/r?tab=newest&page=1&pagesize=50")
```

The example above reads the first search query page of results on the SO website. Now that I have the parsed HTML of the entire page, the next step was to extract the 50 questions on the page. To do this, I created the function: page_links()

Page_links description:

After using the read_page() with the specified url, we want to find the specific paths associated with each of the 50 questions. From the webpage, it is pretty simple to see that we could find a question page by finding the path and pasting it after the base SO url. For example, if a question in the HTML file had an @href = /questions/1114699/creating-an-adjacency-list-from-a-data-frame, we would get the question page by adding that path to https://stackoverflow.com. For page_links(), I was able to find that exact @href path for each question, and pasted the paths with the base url and returned the urls. Here is the output of the first ten question urls for the first page of SO.

```
# Read all (50) questions from a specific page, uses read_page function
head(page_links("https://stackoverflow.com/questions/tagged/r?tab=newest&page=1&pagesize=50"),10)
```

```
##  [1] "https://stackoverflow.com/questions/76401772/unable-to-install-gtk2-on-windows-using-different
##  [2] "https://stackoverflow.com/questions/76401727/how-to-create-a-continuous-geom-line-across-a-fac
##  [3] "https://stackoverflow.com/questions/76401706/how-to-loop-through-polygons-that-overlap-and-ran
##  [4] "https://stackoverflow.com/questions/76401671/support-of-nanotime-by-rsqlite"
##  [5] "https://stackoverflow.com/questions/76401516/is-it-possible-to-add-a-new-row-after-several-row
##  [6] "https://stackoverflow.com/questions/76401452/how-can-i-use-the-same-fill-colour-for-same-categ
##  [7] "https://stackoverflow.com/questions/76401375/use-lapply-function-to-modify-several-matrices-in
##  [8] "https://stackoverflow.com/questions/76401323/how-to-generate-a-max-variable-for-a-group-exclud
##  [9] "https://stackoverflow.com/questions/76401025/tree-coordinates-matching-on-a-2d-plane-in-r"
## [10] "https://stackoverflow.com/questions/76401017/replace-specific-text-values-in-a-row-of-a-datafra
```

This works for every page on the SO website. Lets try another random page

```
head(page_links("https://stackoverflow.com/questions/tagged/r?tab=newest&page=9814&pagesize=50"),10)
```

```
##  [1] "https://stackoverflow.com/questions/1491124/rounding-output-from-by-function-in-r"
##  [2] "https://stackoverflow.com/questions/1489788/trying-to-return-a-specified-number-of-characters-
##  [3] "https://stackoverflow.com/questions/1489526/trying-to-loop-through-a-dataframe-and-reference-mu
##  [4] "https://stackoverflow.com/questions/1489199/how-can-i-take-multiple-vectors-and-recode-their-da
```

```
##  [5] "https://stackoverflow.com/questions/1487320/blogging-with-r-easy-way-to-embed-r-in-a-blog-post
##  [6] "https://stackoverflow.com/questions/1484904/renaming-rows-and-columns-in-r"
##  [7] "https://stackoverflow.com/questions/1484472/adding-summary-statistics-or-even-raw-data-points-
##  [8] "https://stackoverflow.com/questions/1484307/label-of-log-y-axis-1000-instead-of-1e03"
##  [9] "https://stackoverflow.com/questions/1481032/help-plotting-geographic-data-in-r-using-pbsmapping
## [10] "https://stackoverflow.com/questions/1478758/optimizing-the-computation-of-a-recursive-sequence
```

These functions are able to properly read in a search result page, and return the links to the 50 questions on that page.

2.) Find URL for next page of Search Query Results

Lets say our search query was [r], which took us to the initial page of "https://stackoverflow.com/questions/tagged/r". Now, I want the second page of results. One way to do this manually is to click the next button, or 2 button on the bottom of the page. This would take us to a new link, "https://stackoverflow.com/questions/tagged/r?tab=newest&page=2&pagesize=50".

To do this programatically, I built a nextURL() function.

NextURL() description:

This function takes in a search query page link. In the example above, it would be the tagged/r. Based on this page, I would first parse it into HTML (using read_page() from above) for using Xpath. To get the next page url, we could press on the 'next' button, or the '2' button. For general purposes, I want to simulate the click of the next button, therefore I extracted the path for 'next' within the parsed HTML for the input url. With that said, I was able to find the path "//a[@rel = 'next']", which finds the specific node that contains the url information needed for the next page. To get the next page, I used another function, getNextURL(), within MY nextURL() function, and found the @href of the node that has the path: "//a[@rel = 'next']" RELATIVE TO THAT OF THE INPUT URL, using getRelativeURL(). I also checked that if there was no @rel = 'next' path in the parsed HTML of the input url, the function would return blank, meaning that a next button DNE. Finally, I plugged in my parsed HTML into the getNextURL() within my nextURL function, which retrieved the relative URL. If the description is hard to follow, please refer to my function.

```
# Find URL for next page, uses read_page and getNextURL functions
nextURL("https://stackoverflow.com/questions/tagged/r")
```

```
## [1] "https://stackoverflow.com/questions/tagged/r?tab=newest&page=2&pagesize=50"
```

The function returns exactly what was asked for. The input url was the first search query page link, and it returned the second page link of the search query.

3.) Reading a Question Page

For reading a question page, I extracted the question HEADER (not text), the associated answers, and the comments. To do this, all I needed to do was to find the correct Xpath and add everything into a list of lists. The first list represents the question, second -> answer, and third -> comments given a specific question url.

```
# Reading a question page, extracting question, answer, and comments
pg_info =
 ↪  read_question_pg("https://stackoverflow.com/questions/76366992/how-to-line-up-ui-elements-using-styl
head(pg_info)
```

```
## [[1]]
## [[1]][[1]]
## [1] "How to line up ui elements using style argument in shiny app that uses bslib 5"
##
##
## [[2]]
## [[2]][[1]]
## [1] "We could use style argument: style = \"margin-top: 35px; margin-left: 10px; height: 38px;\"\ndi
## [2] "In your fluidRow, you can add:\nstyle = \"display: flex; flex-wrap: nowrap; width: min-content;"
##
##
## [[3]]
## [[3]][[1]]
## [1] "thanks! could you explain the div() class argument?"
## [2] "thanks for the suggestion. What does this do?"
## [3] "it creates a 'flex' container, sets the alignment of items within it to be placed side by side,
```

From the given question above, I went in manually to check the contents, and from the output we can see that it extracted all the correct information.

4.) Processing 4 Pages of Interest

This section contains all the functions that have not been listed yet in this report. I will explain how they all come together.

Questions

Related Functions: Questions() and Question_info()

Descriptions

Essentially, this function returns all of the information related to a question. This includes:

- Votes, views, text, tags, date, user, user badges, etc.

The **questions() function** only takes in one argument: the url of the page containing all 50 questions. From this url, I read the HTML file using my read_page(), extracted the question urls using my page_links(), and introduced a new function, **parsed_questions()**. Basically, this function just parses all of the question links into HTML for further Xpath evaluation. The function also checks for any potential "blank" questions (length == 0), and returns a character() if so. It also stores the parsed HTMLs for the questions in a list, for easier analysis for Xpath.

To extract the content, I used another function **question_info()**, which takes in 3 arguments, and is used to extract all the specific feature given the Xpath and put it into a list. It also checks whether the path exists in a certain parsed HTML question page, and it will return char(0) if the path was not found (length = 0).

For presenting the questions, I put everything into a data frame. Later when I get to answers and comments, I explain why I did not put it into dfs and instead into a list of lists of lists, and eventually merged them back into the question data frame. Here is a sample of the output using my questions() function:

```
question = questions("https://stackoverflow.com/questions/tagged/r")
lapply(question[1:11], function(x) head(x,5))
```

```
## $Questions
## [1] "https://stackoverflow.com/questions/76401772/unable-to-install-gtk2-on-windows-using-different-
## [2] "https://stackoverflow.com/questions/76401727/how-to-create-a-continuous-geom-line-across-a-face
## [3] "https://stackoverflow.com/questions/76401706/how-to-loop-through-polygons-that-overlap-and-rand
## [4] "https://stackoverflow.com/questions/76401671/support-of-nanotime-by-rsqlite"
## [5] "https://stackoverflow.com/questions/76401516/is-it-possible-to-add-a-new-row-after-several-rows
##
## $`Question Views`
## [1] "3 times"  "5 times"  "8 times"  "5 times"  "14 times"
##
## $`Question Votes`
## [1] "0" "1" "0" "0" "0"
##
## $`Question Text`
## [1] "I am having trouble with installing Gtk2 on Windows following the instructions here: https://ra
## [2] "I'm working on a ggplot code. I have a dataset AirPassenger in tsibble format and I want to cre
## [3] "Is there a way to randomly sample 5 points from individual polygons using st_sample if those po
## [4] "I am looking into what is required to support nanotime objects in RSQLite queries. They are jus
## [5] "My data is like this:\nstructure(list(ACT = c(23.13958, 23.04186967, 24.20995687, 24.40250502, 
##
## $`Question Tags`
## [1] "r, gtk2"                          "r, ggplot2, time-series, x-axis"
```

```
## [3] "r, spatial, sf"                  "r, rsqlite, nanotime"
## [5] "r, dplyr, row"
##
## $`Question Date Posted`
## [1] "2 mins ago"  "6 mins ago"  "12 mins ago" "22 mins ago" "57 mins ago"
##
## $`Question Displayname`
## [1] "Tom"             "Rony Golder"     "Kirsten Fuller"  "Davor Josipovic"
## [5] "A.Mokhtari"
##
## $`Question User Reputation`
## [1] "1"     "75"    "1"     "5,207" "1"
##
## $`Question User Badges`
## [1] ""
## [2] "5 bronze badges"
## [3] ""
## [4] "1 gold badge, 37 silver badges, 55 bronze badges"
## [5] "2 bronze badges"
##
## $`Question Editors`
## [1] "Progman" ""        "Progman" "Progman" "margusl"
##
## $`Edited Question Time`
## [1] "2023-06-04 18:02:32 PDT" NA
## [3] "2023-06-04 17:52:56 PDT" "2023-06-04 17:52:44 PDT"
## [5] "2023-06-04 17:08:28 PDT"
```

Verification

I manually compared the results to the SO webpage, and everything seemed to match. A way to verify that the results were read in correctly is to look at the question post dates. The dates/time should be in order of most recent to older, as the first posts should have more recent times.

```
head(question$`Question Date Posted`,10)
```

```
##  [1] "2 mins ago"  "6 mins ago"  "12 mins ago" "22 mins ago" "57 mins ago"
##  [6] "1 hour ago"  "1 hour ago"  "1 hour ago"  "3 hours ago" "3 hours ago"
```

Since I extracted the very first page of search query results, it is not surprising that we see that the post creation times are pretty recent.

Answers

Related Functions: Answers() and Answer_info()

Descriptions

These functions have similar procedures to the question functions. The main difference is that for every question, there may be multiple answers. Due to this, if there were hypothetically 10 answers for a single question, it would not make sense to put all 10 of those answers in a single cell in that data frame. It would be difficult to extract the information and the data would be messy. Instead, I made the functions return a list of lists of lists, which provides greater flexibility in retrieving answers.

To make the concept of nested lists more transparent, I have a list containing a lists of all the answer information:

- Text
- User
- Reputation/Badges

So those (4) features I listed above are all of type list. They are all contained within a bigger list. Within each feature that is a list, there is one more layer of a list that may contain multiple counts for that feature. For extracting each feature, I found the corresponding Xpaths.

Answer_info() serves a similar function question_info(), the only difference being that I added an additional lapply() to put multiple answers for a specific question into lists.

```
answer = answers("https://stackoverflow.com/questions/tagged/r")
lapply(answer[1:5], function(x) head(x,5))
```

```
## $`Ans Body`
## $`Ans Body`[[1]]
## character(0)
##
## $`Ans Body`[[2]]
## character(0)
##
## $`Ans Body`[[3]]
## character(0)
##
## $`Ans Body`[[4]]
## character(0)
##
## $`Ans Body`[[5]]
## character(0)
##
##
## $`Ans User`
## $`Ans User`[[1]]
## character(0)
##
## $`Ans User`[[2]]
## character(0)
##
## $`Ans User`[[3]]
## character(0)
```

```
##
## $`Ans User`[[4]]
## character(0)
##
## $`Ans User`[[5]]
## character(0)
##
##
## $`Ans Post Date`
## $`Ans Post Date`[[1]]
## character(0)
##
## $`Ans Post Date`[[2]]
## character(0)
##
## $`Ans Post Date`[[3]]
## character(0)
##
## $`Ans Post Date`[[4]]
## character(0)
##
## $`Ans Post Date`[[5]]
## character(0)
##
##
## $`Ans Reputation`
## $`Ans Reputation`[[1]]
## character(0)
##
## $`Ans Reputation`[[2]]
## character(0)
##
## $`Ans Reputation`[[3]]
## character(0)
##
## $`Ans Reputation`[[4]]
## character(0)
##
## $`Ans Reputation`[[5]]
## character(0)
##
##
## $`Ans Badges`
## $`Ans Badges`[[1]]
## character(0)
##
## $`Ans Badges`[[2]]
## character(0)
##
## $`Ans Badges`[[3]]
## character(0)
##
## $`Ans Badges`[[4]]
## character(0)
```

```
##
## $`Ans Badges`[[5]]
## character(0)
```

Verification

The answer dates should also be in order of most recent to older. In addition, it would make sense if the answers came AFTER the question, assuming it was not an edited question. Lets check this

```
# Questions
head(question$`Question Date Posted`,10)
```

```
##  [1] "2 mins ago"  "6 mins ago"  "12 mins ago" "22 mins ago" "57 mins ago"
##  [6] "1 hour ago"  "1 hour ago"  "1 hour ago"  "3 hours ago" "3 hours ago"
```

```
# Answers
head(answer$`Ans Post Date`,10)
```

```
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
##
## [[4]]
## character(0)
##
## [[5]]
## character(0)
##
## [[6]]
## [[6]][[1]]
## [[6]][[1]][[1]]
## [1] "41 mins ago"
##
##
##
## [[7]]
## [[7]][[1]]
## [[7]][[1]][[1]]
## [1] "1 hour ago"
##
##
##
## [[8]]
## [[8]][[1]]
## [[8]][[1]][[1]]
## [1] "59 mins ago"
##
```

```
##
## [[8]][[2]]
## [[8]][[2]][[1]]
## [1] "1 hour ago"
##
##
##
## [[9]]
## character(0)
##
## [[10]]
## [[10]][[1]]
## [[10]][[1]][[1]]
## [1] "2 hours ago"
```

From the results above, the first row represents the time for the questions, while the list of lists represents answer times. We see that for the answers, they came more recently or equal to than the question, meaning that the question did indeed come beforehand, which makes sense.

Comments

Related functions: Comments(), Comment_info(), Parsed_Comments()

Descriptions

These functions are exactly like the answer functions; they take in a list of list of lists, as there may be multiple comments for a particular answer. However, I had to do one extra step for comments, as some comments were hidden under a link saying "Show xyz more comments".

Here's an example: https://stackoverflow.com/questions/2851327/combine-a-list-of-data-frames-into-one-data-frame-by-row

To fetch the comments that were hidden under the link, instead of looking at the node that led to "comment-post" (which gave me all the comments on the page besides the hidden comments under the link), I extracted the path where the node contained "js-show-link comments-link dno". From this, I found that every single question page has this exact node, even if there was no comment, or more surprisingly NO ANSWER. This means that every question/answer on each individual page had a separate page for it comments, and it either contained information or just blank.

Now that I have that down, I had to figure out how to extract the urls for the separate comment pages in order to actually extract all the comments. By playing around with the developer tools and refreshing the network tab, I found that after I clicked on the "Show xyz more comments" link, a new request popped up in the tab, and took me to this page: **https://stackoverflow.com/posts/2851434/comments?_=1685858830012.** We can see that this new url contains all the comments associated with that answer, which gives me the result I want.

The last thing I had to do was extract the proper url. To do this, all I had to do was find the parent node of the node that contained the "js-show-link" mentioned above, as that contained the unique ID I needed for the post. I extracted this by using xmlGetAttr() on the ID of the parent node, and was able to get the number. Finally, I simply pasted the proper url: baseurl/posts + ID + comments, and made a list of lists of lists.

```
comment = comments("https://stackoverflow.com/questions/tagged/r")
lapply(comment[1:3], function(x) head(x,5))
```

```
## $`Comment Text`
## $`Comment Text`[[1]]
## character(0)
##
## $`Comment Text`[[2]]
## character(0)
##
## $`Comment Text`[[3]]
## character(0)
##
## $`Comment Text`[[4]]
## character(0)
##
## $`Comment Text`[[5]]
## character(0)
##
##
## $`Comment User`
## $`Comment User`[[1]]
## character(0)
##
```

```
## $`Comment User`[[2]]
## character(0)
##
## $`Comment User`[[3]]
## character(0)
##
## $`Comment User`[[4]]
## character(0)
##
## $`Comment User`[[5]]
## character(0)
##
##
## $`Comment Date`
## $`Comment Date`[[1]]
## character(0)
##
## $`Comment Date`[[2]]
## character(0)
##
## $`Comment Date`[[3]]
## character(0)
##
## $`Comment Date`[[4]]
## character(0)
##
## $`Comment Date`[[5]]
## character(0)
```

Pages of Interest

Related functions: Pages_search_result(), Last_page, Next_page

Description:

For processing the pages of interest, I wrote a function called pages_search_result that combines all the questions, answers, and comments information given a specific search query page of results. I combined the parts above in a data frame, and I had to use lapply(…,I) in order to add the answers and comments, as they were lists. The "I" basically says that we want to keep the lists as they are in their current form when appending it to the data frame, which makes it a lot easier to add in the lists properly.

I plan to arrange the 4 pages in the following format:

- I'm going to grab the second and third page and all of the results, and display some of the output
- After, I'm going to grab the first and last page, and do some verification to ensure the data seems sensible and was read in fine

Second Page

```
second_pg = pages_search_result(nextURL("https://stackoverflow.com/questions/tagged/r"))
lapply(second_pg[1:19], function(x) head(x,3))
```

```
## $Questions
```

```
## [1] "https://stackoverflow.com/questions/76397278/how-to-add-multiple-column-headers-that-span-speci
## [2] "https://stackoverflow.com/questions/76397124/python-or-r-packages-to-draw-2d-diagrams-of-interm
## [3] "https://stackoverflow.com/questions/76396915/aggregating-model-summaries-from-list-of-models"
##
## $Question.Views
## [1] "35 times" "40 times" "39 times"
##
## $Question.Votes
## [1] "1" "1" "0"
##
## $Question.Text
## [1] "I'm looking for examples of how to add multiple column headers to a table rendered with DT tabl
## [2] "I've docked a large inorganic compound (receptor) with an organic molecule that isn't a protein
## [3] "I am running same models several times on different subset of data, for example,\n  data(\"mtca
##
## $Question.Tags
## [1] "css, r, json, shiny, dt"
## [2] "python, r, diagram, interaction, docking"
## [3] "r, list, apply, plyr, sapply"
##
## $Question.Date.Posted
## [1] "yesterday" "yesterday" "yesterday"
##
## $Question.Displayname
## [1] "Curious Jorge - user9788072" "ginn"
## [3] "bison2178"
##
## $Question.User.Reputation
## [1] "2,286" "95"    "713"
##
## $Question.User.Badges
## [1] "9 silver badges, 20 bronze badges"
## [2] "5 bronze badges"
## [3] "1 gold badge, 8 silver badges, 21 bronze badges"
##
## $Question.Editors
## [1] "" "" ""
##
## $Edited.Question.Time
## [1] NA                          "2023-06-03 18:10:09 PDT"
## [3] "2023-06-03 17:27:34 PDT"
##
## $Ans.Body
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "After synthesizing some of the other SO posts, here's a solution using CSS. Not as easy as XLS
##
##
##
## [[2]]
## character(0)
##
## [[3]]
```

```
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "As stated in the comments, your question is more about having a custom summary function.\nNote t
##
##
##
##
## $Ans.User
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "Curious Jorge - user9788072"
##
##
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "Ricardo Semião e Castro"
##
##
##
##
## $Ans.Post.Date
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "8 hours ago"
##
##
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "yesterday"
##
##
##
##
## $Ans.Reputation
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "2,286"
##
##
##
## [[2]]
```

```
## character(0)
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "3,983"
##
##
##
##
## $Ans.Badges
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "9 silver badges"
##
##
## [[1]][[2]]
## [[1]][[2]][[1]]
## [1] "20 bronze badges"
##
##
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "1 gold badge"
##
##
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "8 silver badges"
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "27 bronze badges"
##
##
##
##
## $Comment.Text
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
##
```

```
##
## $Comment.User
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
##
##
## $Comment.Date
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
```

Third Page

```
third_pg =
 ↪  pages_search_result(nextURL(nextURL("https://stackoverflow.com/questions/tagged/r")))
lapply(third_pg[1:19], function(x) head(x,3))
```

```
## $Questions
## [1] "https://stackoverflow.com/questions/76392902/how-to-launch-second-shiny-modal-based-on-an-event-
## [2] "https://stackoverflow.com/questions/76392893/how-to-format-negative-numbers-with-parenthesis-wi
## [3] "https://stackoverflow.com/questions/76392884/complex-conditional-df-subsetting-with-nested-for-l
##
## $Question.Views
## [1] "13 times" "42 times" "51 times"
##
## $Question.Votes
## [1] "0" "1" "0"
##
## $Question.Text
## [1] "I have a shiny app where an event can and should trigger multiple shinyModals, but it only shows
## [2] "I'm using R to process some data and then OpenXLSX to output to an .xlsx file. I'd like negative
## [3] "I apologize in advance for this headache, and in particular, the minimum amount of data I need t
##
## $Question.Tags
## [1] "r, shiny"                   "r, openxlsx"
## [3] "r, range, nested-for-loop"
##
## $Question.Date.Posted
## [1] "yesterday" "yesterday" "yesterday"
##
## $Question.Displayname
```

```
## [1] "ifoxfoot"     "Laura"        "grace.cutler"
##
## $Question.User.Reputation
## [1] "119" "31"  "61"
##
## $Question.User.Badges
## [1] "6 bronze badges" "4 bronze badges" "4 bronze badges"
##
## $Question.Editors
## [1] ""                      "c(\"r2evans\", \"\")" ""
##
## $Edited.Question.Time
## [1] NA                          "2023-06-02 19:29:56 PDT"
## [3] "2023-06-03 06:48:49 PDT"
##
## $Ans.Body
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "I haven't found a way to make this work without using formal styling (creating a workbook and w
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "Here is a dplyr approach that is much faster than a for loop. One point though, some values in a
##
##
##
##
## $Ans.User
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "Kat"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "Leroy Tyrone"
##
##
##
##
## $Ans.Post.Date
```

```
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "yesterday"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "13 hours ago"
##
##
##
##
## $Ans.Reputation
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "14.7k"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "1,214"
##
##
##
##
## $Ans.Badges
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "3 gold badges"
##
##
## [[2]][[2]]
## [[2]][[2]][[1]]
## [1] "18 silver badges"
##
##
## [[2]][[3]]
## [[2]][[3]][[1]]
```

```
## [1] "51 bronze badges"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "1 gold badge"
##
##
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "14 silver badges"
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "23 bronze badges"
##
##
##
##
## $Comment.Text
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## [1] "Thanks again @LeroyTyrone. Please see my comment re bins above.  I am a little confused by the c
## [2] "In case we are getting different results, here is what I get: drive.google.com/file/d/1OTrcFlVs
##
##
##
## $Comment.User
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## [1] "grace.cutler" "grace.cutler"
##
##
##
## $Comment.Date
## [[1]]
## character(0)
##
## [[2]]
```

```
## character(0)
##
## [[3]]
## [[3]][[1]]
## [1] "57 mins ago" "52 mins ago"
```

First Page

```
first_pg = pages_search_result("https://stackoverflow.com/questions/tagged/r")
lapply(first_pg[1:19], function(x) head(x,3))
```

```
## $Questions
## [1] "https://stackoverflow.com/questions/76401804/how-to-create-a-tibble-from-a-list-with-named-vecto
## [2] "https://stackoverflow.com/questions/76401780/how-can-i-subtract-values-of-each-column-of-my-data
## [3] "https://stackoverflow.com/questions/76401772/unable-to-install-gtk2-on-windows-using-different-i
##
## $Question.Views
## [1] "6 times" "9 times" "7 times"
##
## $Question.Votes
## [1] "0"  "0"  "-1"
##
## $Question.Text
## [1] "I am trying to make an edge df from a dataset that I have for a network I'm building:\n    > hea
## [2] "I have a simple datafrme\ndf <- data.frame(\n  col1 = c(1, 2, 3),\n  col2 = c(4, 5, 6),\n  col3
## [3] "I am having trouble with installing Gtk2 on Windows following the instructions here: https://ra
##
## $Question.Tags
## [1] "r"       "r"       "r, gtk2"
##
## $Question.Date.Posted
## [1] "7 mins ago"  "13 mins ago" "15 mins ago"
##
## $Question.Displayname
## [1] "pandora" "Lara"    "Tom"
##
## $Question.User.Reputation
## [1] "45"  "115" "1"
##
## $Question.User.Badges
## [1] "1 silver badge, 7 bronze badges" "7 bronze badges"
## [3] ""
##
## $Question.Editors
## [1] ""        ""        "Progman"
##
## $Edited.Question.Time
## [1] NA                          NA
## [3] "2023-06-04 18:02:32 PDT"
##
## $Ans.Body
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "If you want to cbind() you need to combine elements with the same number of columns. I think you
```

```
##
##
##
## [[3]]
## character(0)
##
##
## $Ans.User
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "Ben Bolker"
##
##
##
## [[3]]
## character(0)
##
##
## $Ans.Post.Date
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "3 mins ago"
##
##
##
## [[3]]
## character(0)
##
##
## $Ans.Reputation
## [[1]]
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "208k"
##
##
##
## [[3]]
## character(0)
##
##
## $Ans.Badges
## [[1]]
```

```
## character(0)
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "25 gold badges"
##
##
## [[2]][[2]]
## [[2]][[2]][[1]]
## [1] "368 silver badges"
##
##
## [[2]][[3]]
## [[2]][[3]][[1]]
## [1] "451 bronze badges"
##
##
##
## [[3]]
## character(0)
##
##
## $Comment.Text
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
##
##
## $Comment.User
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
##
##
## $Comment.Date
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
```

Last Page

The last page was very cumbersome to retrieve, as the page url was not consistent from time to time. For retrieving the last page, I assume that we started from page 3 of our processed results, and used the url corresponding to page 3 as an input.

My process was that I started from page 3, and extracted the largest page value from that page. The largest value was inconsistent at times (not the same url), so it gave me different last page urls at time. To check whether the page was truly the last, I was able to simulate a click of the "next" button and checked if a page (lets say page 9816 generated from the third page url) had @rel = next. If a page had that, I would go to the next page and keep doing the same process until I found a page with "no questions" on it, and if that was the case I would return the previous url before the blank page.

```
# Third pg
third = nextURL(nextURL("https://stackoverflow.com/questions/tagged/r"))
last = last_page(third)
last_pg = pages_search_result(last)
lapply(last_pg[1:19], function(x) head(x,3))
```

```
## $Questions
## [1] "https://stackoverflow.com/questions/1508889/how-to-count-number-of-numeric-values-in-a-column"
## [2] "https://stackoverflow.com/questions/1508513/selectively-replacing-columns-in-r-with-their-delta
## [3] "https://stackoverflow.com/questions/1504832/help-me-replace-a-for-loop-with-an-apply-function"
##
## $Question.Views
## [1] "43k times" "1k times"  "16k times"
##
## $Question.Votes
## [1] "8" "0" "5"
##
## $Question.Text
## [1] "I have a dataframe, and I want to produce a table of summary statistics including number of vali
## [2] "I've got data being read into a data frame R, by column.  Some of the columns will increase in v
## [3] "...if that is possible\n\nMy task is to find the longest streak of continuous days a user partic
##
## $Question.Tags
## [1] "r, statistics"             "r, diff"
## [3] "r, loops, for-loop, apply"
##
## $Question.Date.Posted
## [1] "Oct 2, 2009 at 10:57" "Oct 2, 2009 at 9:16"  "Oct 1, 2009 at 16:00"
##
## $Question.Displayname
## [1] "PaulHurleyuk" "monch1962"    "gd047"
##
## $Question.User.Reputation
## [1] "7,989" "5,151" "29.6k"
##
## $Question.User.Badges
## [1] "15 gold badges, 54 silver badges, 78 bronze badges"
## [2] "5 gold badges, 30 silver badges, 38 bronze badges"
## [3] "18 gold badges, 106 silver badges, 145 bronze badges"
##
## $Question.Editors
```

```
## [1] "" "" ""
##
## $Edited.Question.Time
## [1] NA NA NA
##
## $Ans.Body
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "These are a few add-on packages that might help (see Quick-R)\n\nUsing the Hmisc package\n\nlibr
##
##
## [[1]][[2]]
## [[1]][[2]][[1]]
## [1] "colSums(!is.na(x)) should work."
##
##
## [[1]][[3]]
## [[1]][[3]][[1]]
## [1] "Can you use something like this?\n\nlength(unique(x))"
##
##
## [[1]][[4]]
## [[1]][[4]][[1]]
## [1] "What are \"blank values\" and \"text values\"? If you have numeric vector then you could have NA
##
##
## [[1]][[5]]
## [[1]][[5]][[1]]
## [1] "Does complete.cases (or sum(complete.cases)) do what you want?"
##
##
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "Without addressing the code in any detail, you're assigning values to column, which is a local v
##
##
## [[2]][[2]]
## [[2]][[2]][[1]]
## [1] "diff calculates the difference between consecutive values in a vector.  You can apply it to each
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "The apply functions are not always (or even generally) faster than a for loop.  That is a remnan
##
##
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "EDIT: Fixed.  I originally assumed that I would have to modify most of rle(), but it turns out c
```

```
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "another option\n\n# convert to Date\nday_table$day <- as.Date(day_table$day, format=\"%Y/%m/%d\"
##
##
## [[3]][[4]]
## [[3]][[4]][[1]]
## [1] "If you've got a really long list of data than it sounds like maybe a clustering problem. Each cl
##
##
## [[3]][[5]]
## [[3]][[5]][[1]]
## [1] "This was Chris's suggestion for how to get the data:\n\ndat <- read.table(textConnection(\n \"da
##
##
##
##
## $Ans.User
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "PaulHurleyuk"
##
##
## [[1]][[2]]
## [[1]][[2]][[1]]
## [1] "meg"
##
##
## [[1]][[3]]
## [[1]][[3]][[1]]
## [1] "Shane"
##
##
## [[1]][[4]]
## [[1]][[4]][[1]]
## [1] "Marek"
##
##
## [[1]][[5]]
## [[1]][[5]][[1]]
## [1] "Jonathan Chang"
##
##
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "Shane"
##
##
## [[2]][[2]]
```

```
## [[2]][[2]][[1]]
## [1] "Richie Cotton"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "Shane"
##
##
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "Matt Parker"
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "gd047"
##
##
## [[3]][[4]]
## [[3]][[4]][[1]]
## [1] "kpierce8"
##
##
##
##
## $Ans.Post.Date
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "Oct 2, 2009 at 11:13"
##
##
## [[1]][[2]]
## [[1]][[2]][[1]]
## [1] "Aug 14, 2010 at 22:08"
##
##
## [[1]][[3]]
## [[1]][[3]][[1]]
## [1] "Oct 2, 2009 at 11:03"
##
##
## [[1]][[4]]
## [[1]][[4]][[1]]
## [1] "Oct 2, 2009 at 15:48"
##
##
## [[1]][[5]]
## [[1]][[5]][[1]]
## [1] "Oct 2, 2009 at 13:55"
##
```

```
##
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "Oct 2, 2009 at 11:01"
##
##
## [[2]][[2]]
## [[2]][[2]][[1]]
## [1] "Oct 2, 2009 at 10:52"
##
##
## [[2]][[3]]
## [[2]][[3]][[1]]
## [1] "Oct 2, 2009 at 11:11"
##
##
## [[2]][[4]]
## [[2]][[4]][[1]]
## [1] "Oct 2, 2009 at 10:56"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "Oct 1, 2009 at 20:45"
##
##
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "Oct 1, 2009 at 16:06"
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "Oct 2, 2009 at 15:12"
##
##
## [[3]][[4]]
## [[3]][[4]][[1]]
## [1] "Oct 1, 2009 at 19:40"
##
##
## [[3]][[5]]
## [[3]][[5]][[1]]
## [1] "Jan 12, 2010 at 7:34"
##
##
## [[3]][[6]]
## [[3]][[6]][[1]]
## [1] "Oct 1, 2009 at 19:48"
##
```

```
##
## [[3]][[7]]
## [[3]][[7]][[1]]
## [1] "May 23, 2017 at 12:08"
##
##
##
##
## $Ans.Reputation
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "7,989"
##
##
## [[1]][[2]]
## [[1]][[2]][[1]]
## [1] "61"
##
##
## [[1]][[3]]
## [[1]][[3]][[1]]
## [1] "98.2k"
##
##
## [[1]][[4]]
## [[1]][[4]][[1]]
## [1] "49.3k"
##
##
## [[1]][[5]]
## [[1]][[5]][[1]]
## [1] "24.4k"
##
##
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "98.2k"
##
##
## [[2]][[2]]
## [[2]][[2]][[1]]
## [1] "118k"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "98.2k"
##
##
```

```
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "26.6k"
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "29.6k"
##
##
## [[3]][[4]]
## [[3]][[4]][[1]]
## [1] "15.8k"
##
##
##
##
## $Ans.Badges
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [1] "15 gold badges"
##
##
## [[1]][[2]]
## [[1]][[2]][[1]]
## [1] "54 silver badges"
##
##
## [[1]][[3]]
## [[1]][[3]][[1]]
## [1] "78 bronze badges"
##
##
## [[1]][[4]]
## [[1]][[4]][[1]]
## [1] "1 silver badge"
##
##
## [[1]][[5]]
## [[1]][[5]][[1]]
## [1] "1 bronze badge"
##
##
## [[1]][[6]]
## [[1]][[6]][[1]]
## [1] "35 gold badges"
##
##
## [[1]][[7]]
## [[1]][[7]][[1]]
## [1] "223 silver badges"
##
##
```

```
## [[1]][[8]]
## [[1]][[8]][[1]]
## [1] "217 bronze badges"
##
##
## [[1]][[9]]
## [[1]][[9]][[1]]
## [1] "15 gold badges"
##
##
## [[1]][[10]]
## [[1]][[10]][[1]]
## [1] "98 silver badges"
##
##
## [[1]][[11]]
## [[1]][[11]][[1]]
## [1] "121 bronze badges"
##
##
## [[1]][[12]]
## [[1]][[12]][[1]]
## [1] "5 gold badges"
##
##
## [[1]][[13]]
## [[1]][[13]][[1]]
## [1] "33 silver badges"
##
##
## [[1]][[14]]
## [[1]][[14]][[1]]
## [1] "33 bronze badges"
##
##
##
## [[2]]
## [[2]][[1]]
## [[2]][[1]][[1]]
## [1] "35 gold badges"
##
##
## [[2]][[2]]
## [[2]][[2]][[1]]
## [1] "223 silver badges"
##
##
## [[2]][[3]]
## [[2]][[3]][[1]]
## [1] "217 bronze badges"
##
##
## [[2]][[4]]
## [[2]][[4]][[1]]
```

```
## [1] "46 gold badges"
##
##
## [[2]][[5]]
## [[2]][[5]][[1]]
## [1] "245 silver badges"
##
##
## [[2]][[6]]
## [[2]][[6]][[1]]
## [1] "359 bronze badges"
##
##
##
## [[3]]
## [[3]][[1]]
## [[3]][[1]][[1]]
## [1] "35 gold badges"
##
##
## [[3]][[2]]
## [[3]][[2]][[1]]
## [1] "223 silver badges"
##
##
## [[3]][[3]]
## [[3]][[3]][[1]]
## [1] "217 bronze badges"
##
##
## [[3]][[4]]
## [[3]][[4]][[1]]
## [1] "6 gold badges"
##
##
## [[3]][[5]]
## [[3]][[5]][[1]]
## [1] "53 silver badges"
##
##
## [[3]][[6]]
## [[3]][[6]][[1]]
## [1] "72 bronze badges"
##
##
## [[3]][[7]]
## [[3]][[7]][[1]]
## [1] "18 gold badges"
##
##
## [[3]][[8]]
## [[3]][[8]][[1]]
## [1] "106 silver badges"
##
```

```
##
## [[3]][[9]]
## [[3]][[9]][[1]]
## [1] "145 bronze badges"
##
##
## [[3]][[10]]
## [[3]][[10]][[1]]
## [1] "2 gold badges"
##
##
## [[3]][[11]]
## [[3]][[11]][[1]]
## [1] "23 silver badges"
##
##
## [[3]][[12]]
## [[3]][[12]][[1]]
## [1] "25 bronze badges"
##
##
##
##
## $Comment.Text
## [[1]]
## [[1]][[1]]
## NULL
##
## [[1]][[2]]
## NULL
##
## [[1]][[3]]
## NULL
##
## [[1]][[4]]
## [1] "+1 Note exactly what the OP asked for, but exactly what I was looking for :-)"
##
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## NULL
##
## [[3]][[2]]
## [1] "Forgot to add: the getSOTable function is from Shane's answer here: stackoverflow.com/questions,
##
## [[3]][[3]]
## NULL
##
## [[3]][[4]]
## NULL
##
```

```
## [[3]][[5]]
## [1] "... yeah, that's probably a bit more sensible.  But I like a little magic in my programming fro
##
##
##
## $Comment.User
## [[1]]
## [[1]][[1]]
## NULL
##
## [[1]][[2]]
## NULL
##
## [[1]][[3]]
## NULL
##
## [[1]][[4]]
## [1] "Shalom Craimer"
##
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## NULL
##
## [[3]][[2]]
## [1] "Matt Parker"
##
## [[3]][[3]]
## NULL
##
## [[3]][[4]]
## NULL
##
## [[3]][[5]]
## [1] "Matt Parker"
##
##
##
## $Comment.Date
## [[1]]
## [[1]][[1]]
## NULL
##
## [[1]][[2]]
## NULL
##
## [[1]][[3]]
## NULL
##
## [[1]][[4]]
## [1] "Jun 12, 2011 at 9:23"
```

```
##
##
## [[2]]
## character(0)
##
## [[3]]
## [[3]][[1]]
## NULL
##
## [[3]][[2]]
## [1] "Oct 1, 2009 at 19:53"
##
## [[3]][[3]]
## NULL
##
## [[3]][[4]]
## NULL
##
## [[3]][[5]]
## [1] "Oct 1, 2009 at 20:18"
```

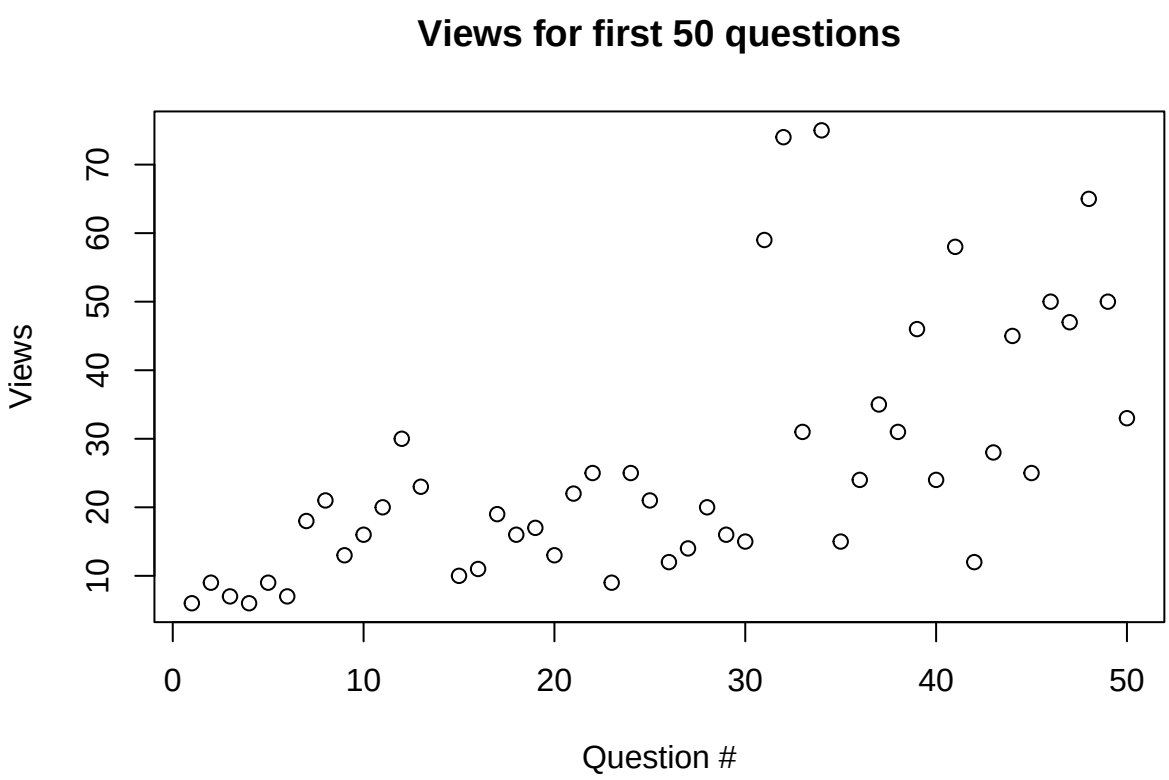
Verification of Results Using First and Last Page

I want to compare the distributions of views and votes for the first and last pages. We should see the distribution of views and votes for the first page to be heavily inclined towards smaller values, whereas for the last page it should be higher.

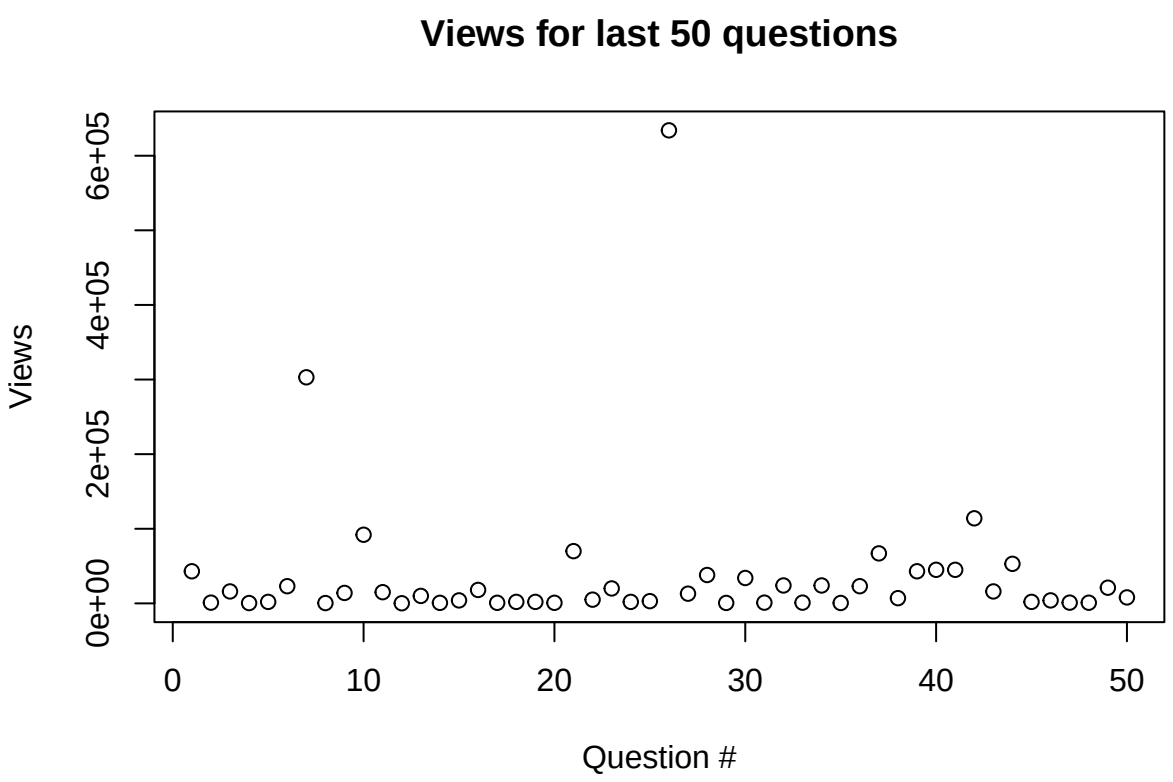
Plots of Views Distributions

```
# First page
plot(as.numeric(gsub("[[:alpha:]]+","",gsub("k","000",first_pg$Question.Views))), xlab =
 ↪  "Question #", ylab = "Views", main = "Views for first 50 questions")
```

```
## Warning in plot(as.numeric(gsub("[[:alpha:]]+", "", gsub("k", "000",
## first_pg$Question.Views))), : NAs introduced by coercion
```




```
# Last Page
plot(as.numeric(gsub("[[:alpha:]]+","",gsub("k","000",last_pg$Question.Views))), xlab =
 ↪  "Question #", ylab = "Views", main = "Views for last 50 questions")
```

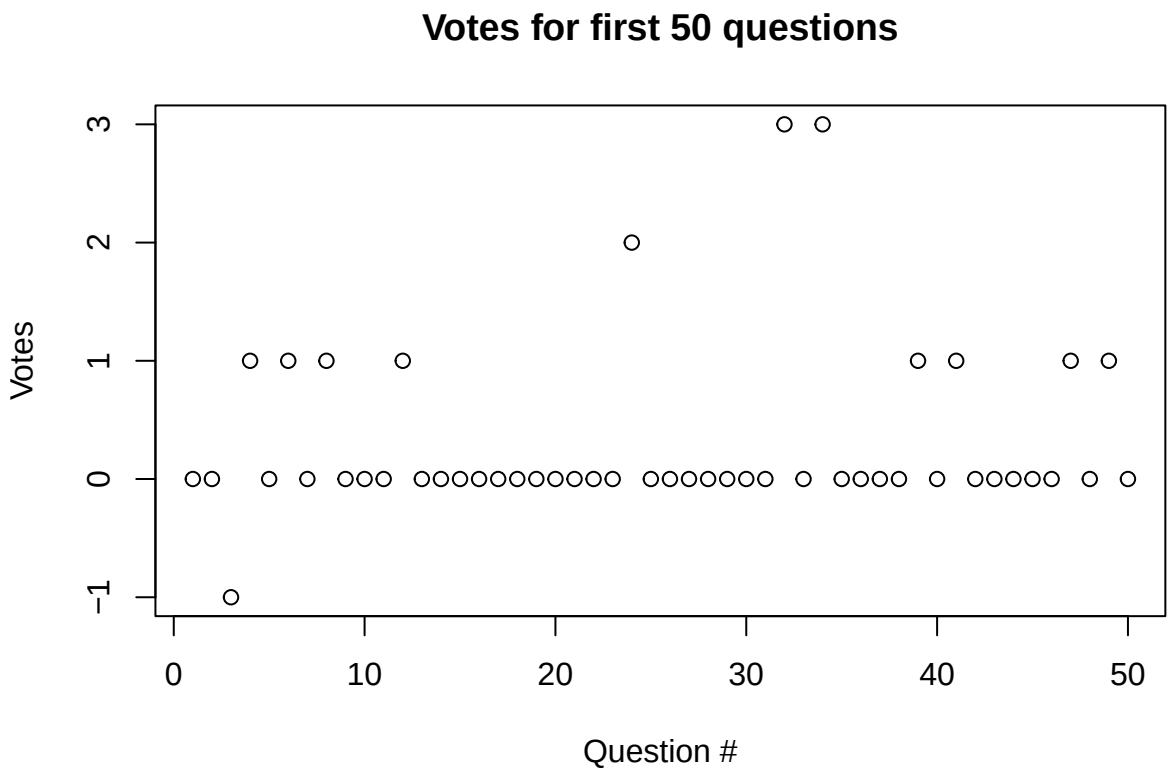


Clearly, we see that there are many more views (in the ten thousands) in the last 50 questions compared to the first 50. This is sensible, as the last pages have been intact since 2009, while the new pages are the most recent.

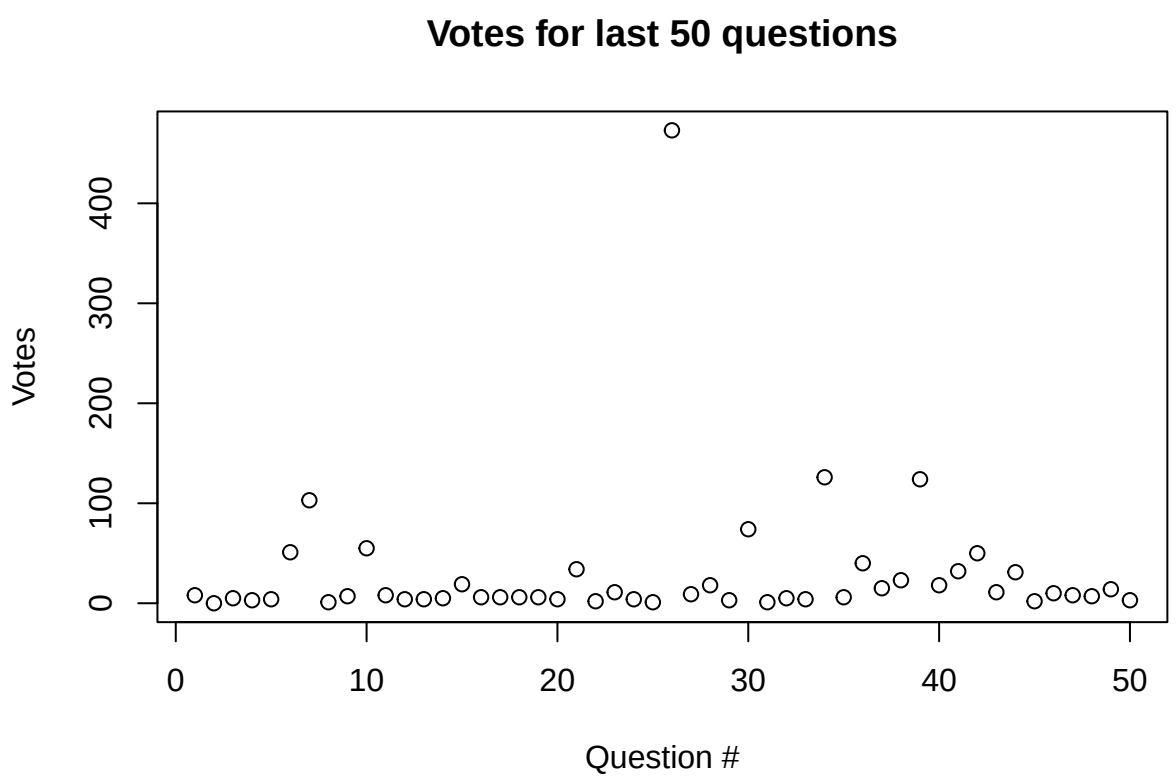
I also plotted the distribution for votes:

Plots of Votes Distributions

```
plot(as.numeric(first_pg$Question.Votes), xlab = "Question #", ylab = "Votes", main =
 ↪  "Votes for first 50 questions")
```




```
plot(as.numeric(last_pg$Question.Votes), xlab = "Question #", ylab = "Votes", main =
↪  "Votes for last 50 questions")
```



We also see many more votes for the last 50 questions compared to the first 50.

Comparing Dates

Another important thing to verify is if the creation dates for the questions on the last 50 pages are all created before the first 50 (excluding edited times).

```
first_pg$Question.Date.Posted
```

```
##  [1] "7 mins ago"   "13 mins ago"  "15 mins ago"  "19 mins ago"  "25 mins ago"
##  [6] "35 mins ago"  "1 hour ago"   "1 hour ago"   "1 hour ago"   "2 hours ago"
## [11] "3 hours ago"  "3 hours ago"  "3 hours ago"  "3 hours ago"  "3 hours ago"
## [16] "3 hours ago"  "4 hours ago"  "4 hours ago"  "4 hours ago"  "5 hours ago"
## [21] "5 hours ago"  "5 hours ago"  "5 hours ago"  "5 hours ago"  "6 hours ago"
## [26] "6 hours ago"  "7 hours ago"  "7 hours ago"  "8 hours ago"  "9 hours ago"
## [31] "9 hours ago"  "11 hours ago" "11 hours ago" "11 hours ago" "12 hours ago"
## [36] "13 hours ago" "16 hours ago" "17 hours ago" "17 hours ago" "18 hours ago"
## [41] "18 hours ago" "18 hours ago" "18 hours ago" "20 hours ago" "20 hours ago"
## [46] "21 hours ago" "22 hours ago" "22 hours ago" "23 hours ago" "23 hours ago"
```

```
last_pg$Question.Date.Posted
```

```
##  [1] "Oct 2, 2009 at 10:57"  "Oct 2, 2009 at 9:16"   "Oct 1, 2009 at 16:00"
##  [4] "Oct 1, 2009 at 15:39"  "Oct 1, 2009 at 14:31"  "Oct 1, 2009 at 9:32"
##  [7] "Sep 30, 2009 at 11:23" "Sep 30, 2009 at 7:51"  "Sep 30, 2009 at 7:29"
```

```
## [10] "Sep 29, 2009 at 17:35" "Sep 29, 2009 at 7:52"  "Sep 28, 2009 at 23:02"
## [13] "Sep 28, 2009 at 21:42" "Sep 28, 2009 at 20:28" "Sep 28, 2009 at 14:23"
## [16] "Sep 28, 2009 at 0:28"  "Sep 27, 2009 at 21:01" "Sep 27, 2009 at 19:54"
## [19] "Sep 26, 2009 at 11:34" "Sep 25, 2009 at 18:01" "Sep 25, 2009 at 17:15"
## [22] "Sep 25, 2009 at 10:36" "Sep 25, 2009 at 8:53"  "Sep 25, 2009 at 5:53"
## [25] "Sep 25, 2009 at 4:01"  "Sep 24, 2009 at 20:51" "Sep 23, 2009 at 23:03"
## [28] "Sep 23, 2009 at 22:25" "Sep 23, 2009 at 18:28" "Sep 23, 2009 at 16:37"
## [31] "Sep 23, 2009 at 1:01"  "Sep 22, 2009 at 2:12"  "Sep 21, 2009 at 22:26"
## [34] "Sep 21, 2009 at 12:11" "Sep 21, 2009 at 8:57"  "Sep 20, 2009 at 22:50"
## [37] "Sep 19, 2009 at 19:11" "Sep 19, 2009 at 13:59" "Sep 18, 2009 at 17:47"
## [40] "Sep 18, 2009 at 12:36" "Sep 17, 2009 at 22:50" "Sep 17, 2009 at 15:15"
## [43] "Sep 17, 2009 at 14:47" "Sep 17, 2009 at 13:58" "Sep 16, 2009 at 22:57"
## [46] "Sep 16, 2009 at 19:06" "Sep 16, 2009 at 17:32" "Sep 16, 2009 at 14:55"
## [49] "Sep 16, 2009 at 13:08" "Sep 16, 2009 at 8:22"
```

The dates seem fine.