CODERS' CORNER

csae
CENTRE FOR THE STUDY OF AFRICAN ECONOMIES

Webscraping

Introduction

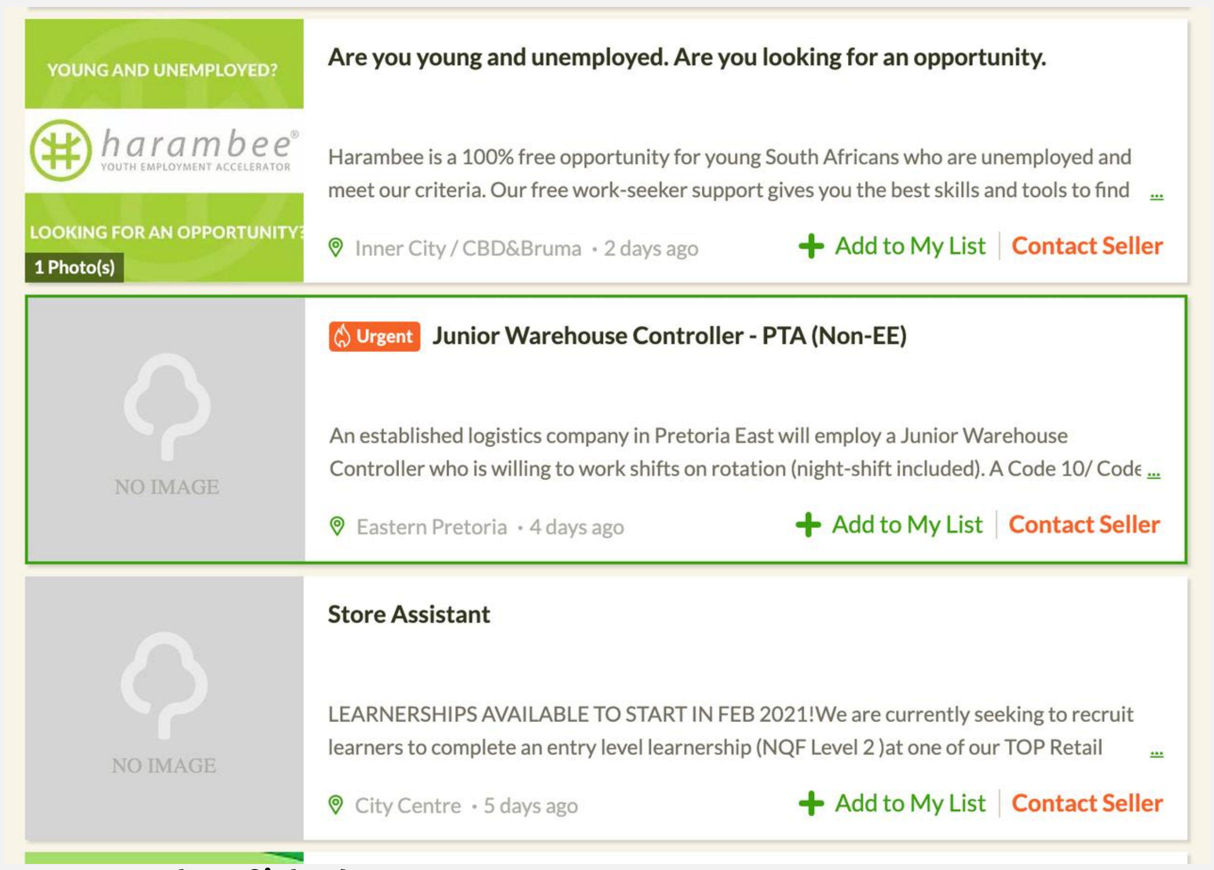
The following post illustrates an approach to "scraping" information from a job posting site where each page on the site contains a number of job post titles, each linking to a full job post**. For more resources, considerations, and examples see** here.

Note: For webscraping tasks, each website is different and solutions will need to be tailored.

Example objective: The code in this example scrapes job post information from "https://www.gumtree.co.za/". The objective is to identify, extract, and aggregate job post information into a data set. We will use an approach called "screen scraping", or HTML parsing. I use a combination of my browser's "inspect tool" and SelectorGadget ("point and click CSS selectors") to figure out what pattern (in the HTML) identifies the piece of information on my screen I'm interested in scraping (in this case job titles, job descriptions, location, &c.). I use the rvest R package ("rvest helps you scrape information from webpages"), and a number of tidyverse tools for data manipulation.

Note: An "element" refers to an HTML element: "An HTML element is defined by a start tag, some content, and an end tag." CSS-selectors "are used to "find" (or select) the HTML elements [...]"

Best practice: (1) Check if the site has a "robots.txt" file which — if a site has one — includes do's and don'ts for scrapers/crawlers. This file can be found at the root of the website host, i.e. "https://www.gumtree.co.za/robots.txt". In the example, I use the robotstxt package to access and check the robots.txt file. (2) Don't go too fast, don't send too many requests to the website per second — be polite.


YOUNG AND UNEMPLOYED?

harambee YOUTH EMPLOYMENT ACCELERATOR

LOOKING FOR AN OPPORTUNITY?

1 Photo(s)

Are you young and unemployed. Are you looking for an opportunity.

Harambee is a 100% free opportunity for young South Africans who are unemployed and meet our criteria. Our free work-seeker support gives you the best skills and tools to find ...

Inner City / CBD&Bruma · 2 days ago + Add to My List | Contact Seller

NO IMAGE

Urgent **Junior Warehouse Controller - PTA (Non-EE)**

An established logistics company in Pretoria East will employ a Junior Warehouse Controller who is willing to work shifts on rotation (night-shift included). A Code 10/ Code ...

Eastern Pretoria · 4 days ago + Add to My List | Contact Seller

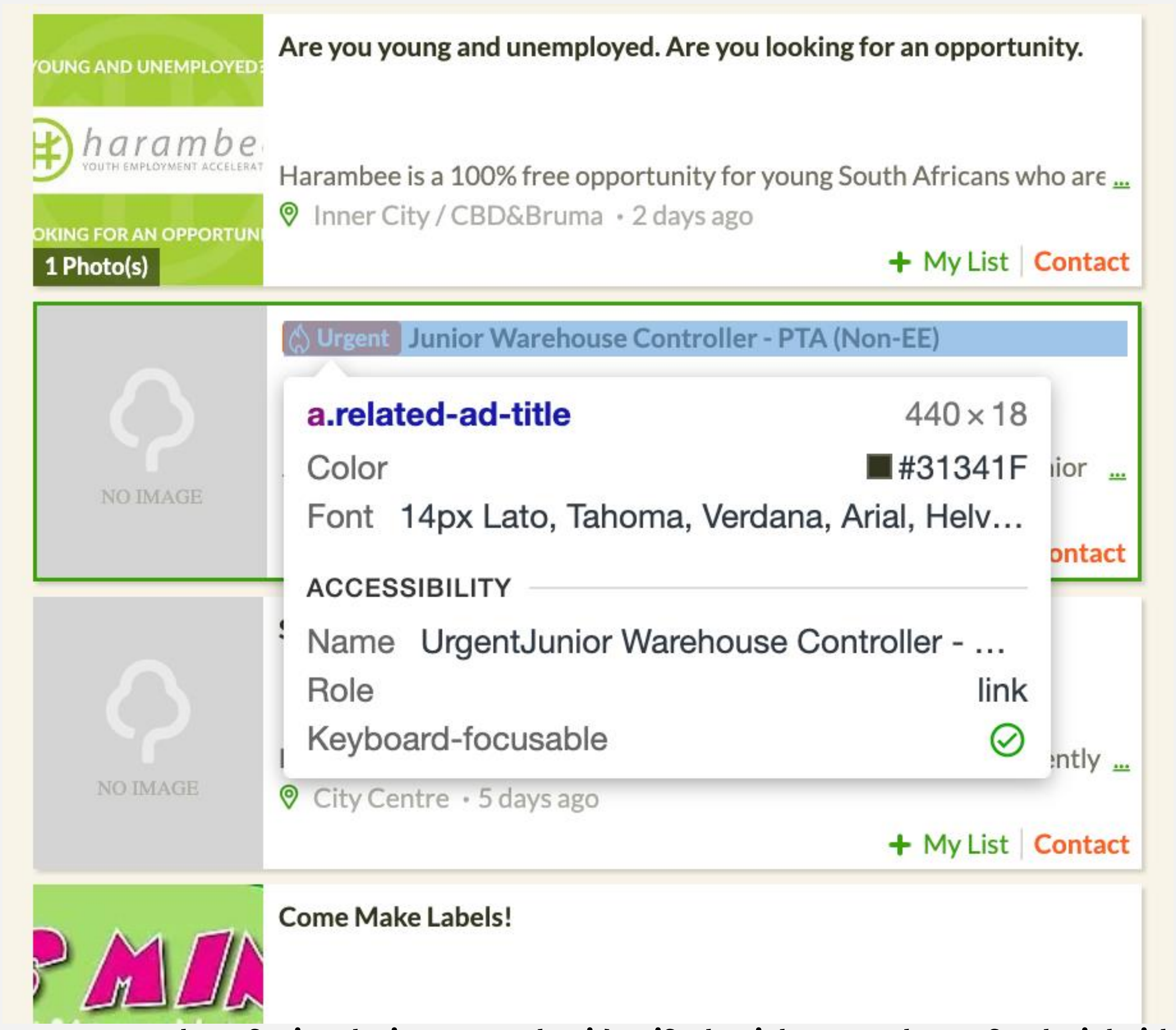
NO IMAGE

Store Assistant

LEARNERSHIPS AVAILABLE TO START IN FEB 2021!We are currently seeking to recruit learners to complete an entry level learnership (NQF Level 2)at one of our TOP Retail ...

City Centre · 5 days ago + Add to My List | Contact Seller

Figure 1: A screenshot of job ads on Gumtree


Are you young and unemployed. Are you looking for an opportunity.

Harambee is a 100% free opportunity for young South Africans who are ...

Inner City / CBD&Bruma · 2 days ago

+ My List | Contact

Urgent Junior Warehouse Controller - PTA (Non-EE)

a.related-ad-title 440 × 18

Color #31341F

Font 14px Lato, Tahoma, Verdana, Arial, Helv…

ACCESSIBILITY

Name UrgentJunior Warehouse Controller - …

Role link

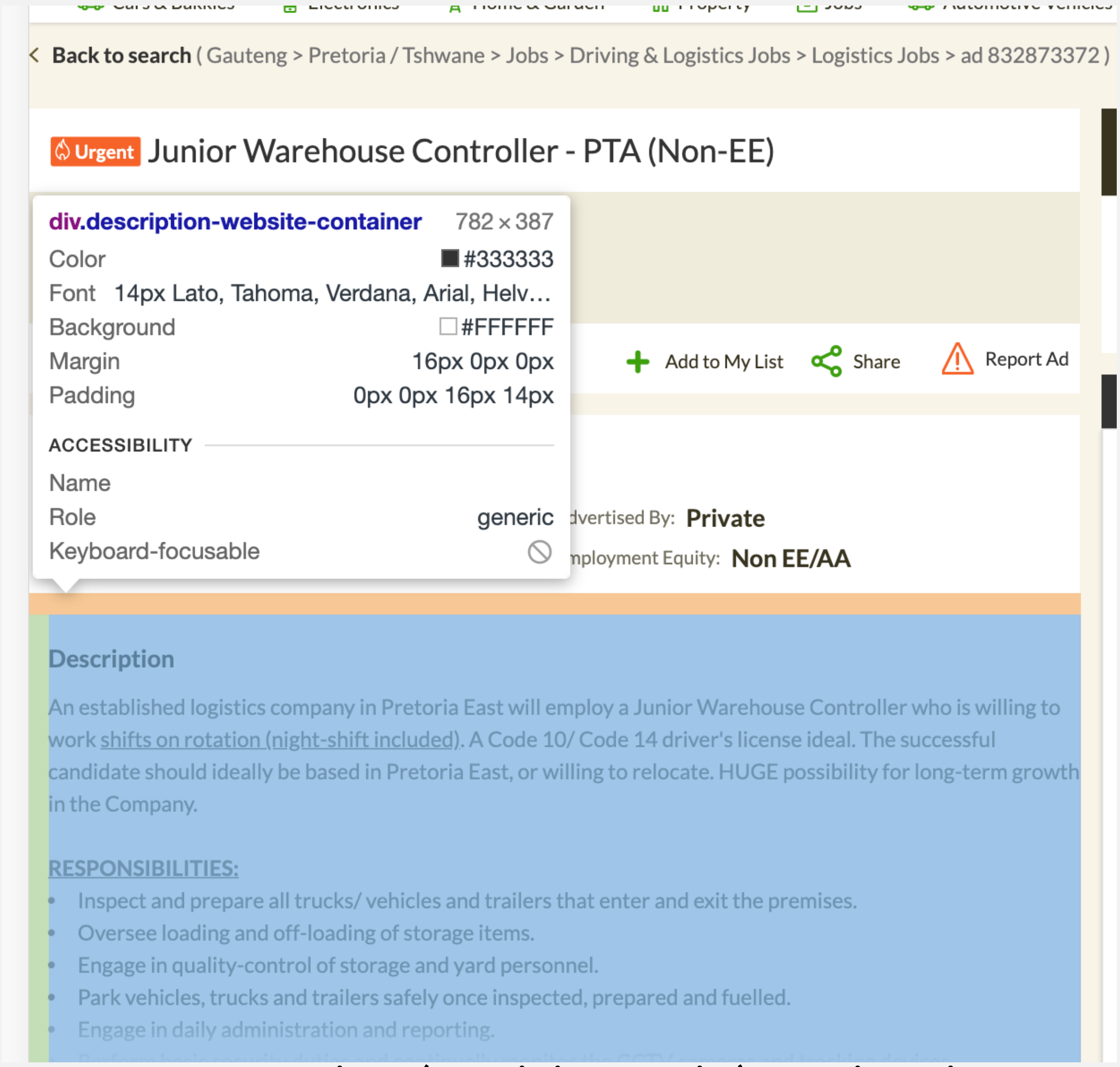
Keyboard-focusable ✓

City Centre · 5 days ago

+ My List | Contact

Come Make Labels!

Figure 2: A screenshot of using the inspect tool to identify the right CSS-selector for the job title (“a.related-ad-title”)


< **Back to search** (Gauteng > Pretoria / Tshwane > Jobs > Driving & Logistics Jobs > Logistics Jobs > ad 832873372)

Urgent Junior Warehouse Controller - PTA (Non-EE)

div.description-website-container 782 × 387

Color	■ #333333
Font	14px Lato, Tahoma, Verdana, Arial, Helv…
Background	□ #FFFFFF
Margin	16px 0px 0px
Padding	0px 0px 16px 14px

ACCESSIBILITY

Name	
Role	generic
Keyboard-focusable	⃠

dvertised By: **Private**

nployment Equity: **Non EE/AA**

Description

An established logistics company in Pretoria East will employ a Junior Warehouse Controller who is willing to work shifts on rotation (night-shift included). A Code 10/ Code 14 driver's license ideal. The successful candidate should ideally be based in Pretoria East, or willing to relocate. HUGE possibility for long-term growth in the Company.

RESPONSIBILITIES:

- Inspect and prepare all trucks/ vehicles and trailers that enter and exit the premises.
- Oversee loading and off-loading of storage items.
- Engage in quality-control of storage and yard personnel.
- Park vehicles, trucks and trailers safely once inspected, prepared and fuelled.
- Engage in daily administration and reporting.

Figure 3: A screenshot of the full job ad (after clicking on the title): Here using the inspect tool to identify the right CSS-selector for the description field (".description-website-container")

Code

Approach: One function to clean an individual element; One function to clean an individual job post (i.e. to clean multiple elements and store the result as a tibble); One loop to extract links to individual job posts over multiple pages and to store these in a list; One loop to iterate over a list of job links, to apply the post cleaning function, and to store the result in a list (a list of tibbles); finally, to collapse the list of tibbles into a single data set, and to export to your chosen location.

Bells and whistles: I use the purrr package's "possibly()" function to deal with errors (so the script doesn't break when a link is broken); I use the progress package to show me, in my console, how far along the scraping is.

Inputs: A number of pages to scrape; a link fragment to increment; CSS-selectors. **Outputs:** A list of job post links to iterate over; a list of tibbles; a dataset of job ads.

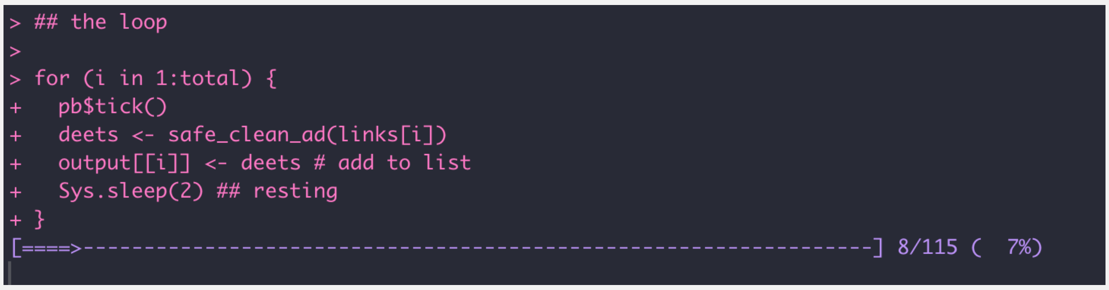
```r
# A. Libraries ####

library(tidyverse) ## for data manipulation (various packages)
library(robotstxt) ## to get robotstxt protocols
library(rvest) ## for scraping
library(textclean) ## some cleaning functions
library(progress) ## for tracking progress in console


# B. Check permissions ####

robotstxt::paths_allowed("https://www.gumtree.co.za/") ## should be TRUE


# C. Function 1: A function to clean an individual element ####
## This function takes the page HTML and the CSS selector, extracts the text, and outputs clean text

clean_element <- function(page_html, selector) {
  output <- page_html %>%
    html_nodes(selector) %>%
    html_text() %>%
    replace_html() %>% ## strip html markup (if any)
    str_squish() ## remove leading and trailing whitespace
  output <- ifelse(length(output) == 0, NA, output) ## in case an element is missing
  return(output)
}


# D. Function 2: A function to clean a job ad ####
## This function extracts information from a single job ad. It takes a job ad link, reads the HTML, and cleans the items you're interested in;
finally it returns a tibble

clean_ad <- function(link) {
  page_html <- read_html(link) ## get HTML
  ## below, the list of items I want in each post. Note ".description-website-container" is an example of the CSS-selector I identified using the
page inspection tool (like SelectorGadget)
  title <- clean_element(page_html, "h1") ## parse HTML
  description <- clean_element(page_html, ".description-website-container")
  jobtype <- clean_element(page_html, ".attribute:nth-child(4) .value")
  employer <- clean_element(page_html, ".attribute:nth-child(3) .value")
  location <- clean_element(page_html, ".attribute:nth-child(1) .value")
  time <- Sys.time() ## current time

  ## I put the selected post info in a tibble
  dat <-
    tibble(
      title,
      description,
      jobtype,
      location,
      time
    )
  return(dat)
}
```

```r
# E. Loop 1: To get a list of job ad links ####
## This loop takes a URL fragment, and increments the web page-number up to a number you've specified. On each page of ads, it extracts the job post link from the job title

pages <- 5 ## let's just do 5 pages (user to set this)
job_list <- list()
link <- "https://www.gumtree.co.za/s-jobs/page-" ## url fragment

for (i in 1:pages) {
  jobs <-
    read_html(paste0(link, i, "/v1c8p", i)) ## using paste0
  links <-
    jobs %>%
    html_nodes(".related-ad-title") %>%
    html_attr("href") ## get links
  job_list[[i]] <- links ## add to list
  Sys.sleep(2) ## rest
}

links <- unlist(job_list) ## make a single list
links <- paste0("https://www.gumtree.co.za", links) ## format


# F. Let's go ####
## We set up the "bells and whistles": A way to see scraping progress in the console, and a way handle errors (when links don't work). We iterate through each job link in our list of links, apply the job-ad cleaning function, and store the resulting tibble in a list; finally we collapse this list of tibbles into our desired data set

## track progress
total <- length(links)
pb <- progress_bar$new(format = "[:bar] :current/:total (:percent)", total = total)

## error handling
safe_clean_ad <- possibly(clean_ad, NA) ## see purrr package

## list
output <- list()

## the loop
for (i in 1:total) {
  pb$tick()
  deets <- safe_clean_ad(links[i])
  output[[i]] <- deets # add to list
  Sys.sleep(2) ## resting
}

# G. Combine all tibbles in the list into a single big tibble ####

all <- output[!is.na(output)] ## remove empty tibbles, if any
all <- bind_rows(all) ## Fab!
glimpse(all)

write_csv(all, "myfile.csv") ## export
```

Results

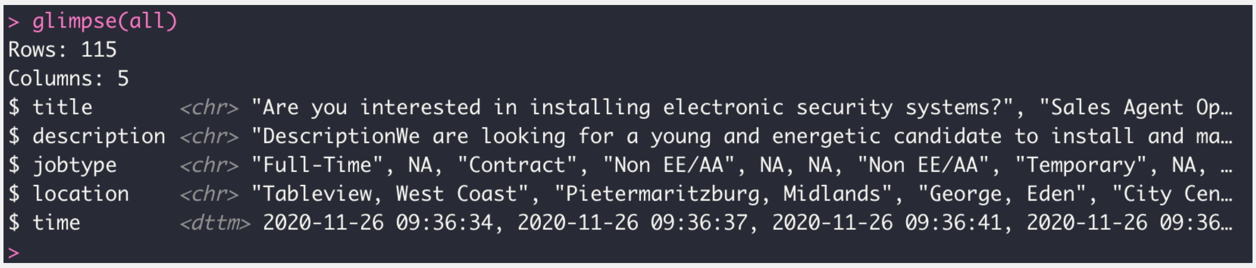
You can see the progress of the script, an example of a job tibble, and the first few observations of the resultant dataset in the screenshots below.

The final output of the above code is shown in Figure 6. The dataset contains variables concerning the job title, description, type, location, and the time it was downloaded.



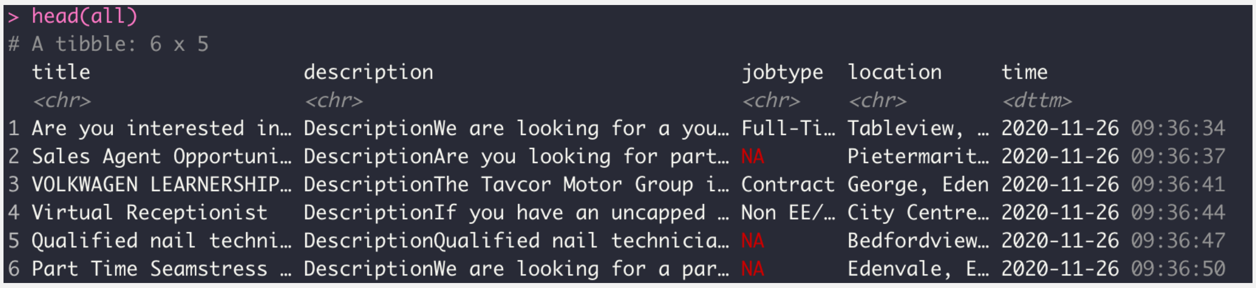
```
> ## the loop
>
> for (i in 1:total) {
+   pb$tick()
+   deets <- safe_clean_ad(links[i])
+   output[[i]] <- deets # add to list
+   Sys.sleep(2) ## resting
+ }
[====>-----------------------------------------------------------------] 8/115 (  7%)
```

Figure 4: A screenshot of the scraping script's progress in the R console



```
> glimpse(all)
Rows: 115
Columns: 5
$ title       <chr> "Are you interested in installing electronic security systems?", "Sales Agent Op…
$ description <chr> "DescriptionWe are looking for a young and energetic candidate to install and ma…
$ jobtype     <chr> "Full-Time", NA, "Contract", "Non EE/AA", NA, NA, "Non EE/AA", "Temporary", NA, …
$ location    <chr> "Tableview, West Coast", "Pietermaritzburg, Midlands", "George, Eden", "City Cen…
$ time        <dttm> 2020-11-26 09:36:34, 2020-11-26 09:36:37, 2020-11-26 09:36:41, 2020-11-26 09:36…
>
```

Figure 5: A screenshot of the resulting job post tibble



```
> head(all)
# A tibble: 6 x 5
  title                  description                          jobtype  location     time
  <chr>                  <chr>                                <chr>    <chr>        <dttm>
1 Are you interested in… DescriptionWe are looking for a you… Full-Ti… Tableview, … 2020-11-26 09:36:34
2 Sales Agent Opportuni… DescriptionAre you looking for part… NA       Pietermarit… 2020-11-26 09:36:37
3 VOLKWAGEN LEARNERSHIP… DescriptionThe Tavcor Motor Group i… Contract George, Eden 2020-11-26 09:36:41
4 Virtual Receptionist   DescriptionIf you have an uncapped … Non EE/… City Centre… 2020-11-26 09:36:44
5 Qualified nail techni… DescriptionQualified nail technicia… NA       Bedfordview… 2020-11-26 09:36:47
6 Part Time Seamstress … DescriptionWe are looking for a par… NA       Edenvale, E… 2020-11-26 09:36:50
```

Figure 6: A screenshot of the first few observations in the job post data set

Wim Louw Research Manager, J-PAL Africa
30 November 2020

Resources

- Mine Dogucu & Mine Çetinkaya-Rundel (2020): "WebScraping in the Statistics and Data Science Curriculum: Challenges and Opportunities", Journal of Statistics Education (FYI, see their use of purrr's "map_dfr()" function!)
- "How we learnt to stop worrying and love web scraping" in the Nature Career Column by Nicholas J. DeVito, Georgia C. Richards & Peter Inglesby
- Chapter 12 in "Automate the Boring Stuff with Python" by Al Sweigart
- "R for data science" by Garrett Grolemund and Hadley Wickham
- See also the "polite" R package