



NHS-R
COMMUNITY

NHSDatadictionary – a new
package to automate the
the web scraping of NHS
NHS Data Dictionary
lookups

Gary Hutson – Head of Advanced Analytics
Arden & GEM Commissioning Support Unit

NHSDDataDictionary package benefits



Always up to date, as the web scraping engine brings back the results from the NHS Data Dictionary site in real-time



Rationalisation of NHS lookups and providing consistency to lookups

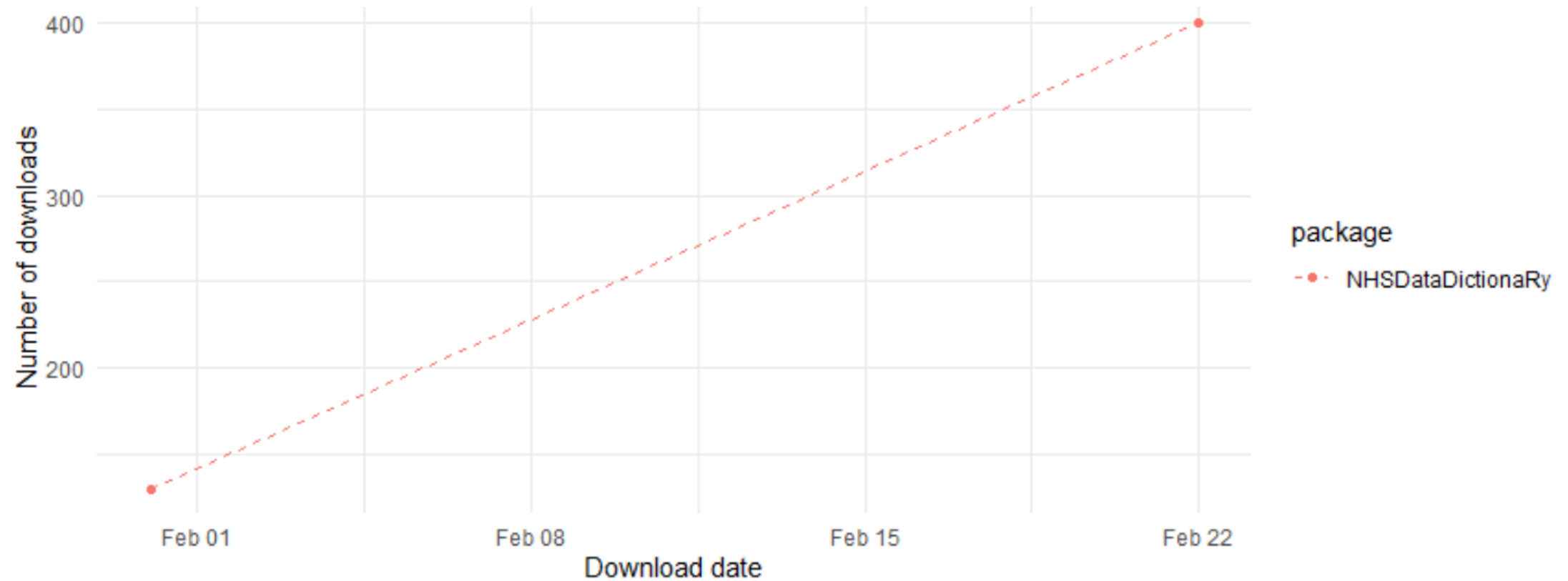


Ability to perform custom scraping of websites and website elements



Open source and maintained by the NHS-R Community

Downloads to date (n=529)



Using the package



Supporting resources



1 Loading the package

2 Accessing the NHS data links

3 Get all current hyperlinks from a webpage using linkScrapR

4 Opening a URL from R into a web browser

5 Working with the NHS R Data Dictionary lookup

5.1 tableR function (utilising scrapeR function)

5.2 Using my lookup with NHS data

5.3 Combine into one function

6 xpathTextR function

6.1 Cleaning the text example

6.2 Manipulating the text with Excel like string functions

7 Working with OpenSafely

8 Wrapping up

NHSDDataDictionaRy package

Gary Hutson - Head of Advanced Analytics

21/04/2021

1 Loading the package

To load the package, you can use the below command:

2 Accessing the NHS data links

This function expects no return and is a way to query the NHS Data Dictionary database to get the most recent list of data elements and their associated lookups. The return of this will provide a tibble of all the links currently on the NHS Data Dictionary website:

```
nhs_tibble <- NHSDDataDictionaRy::nhs_data_elements()
print(head(nhs_tibble))
```

```
## # A tibble: 6 x 6
##   link_name url      full_url xpath_nat_code xpath_default_co~ xpath_also_known
##   <chr>      <chr>    <chr>      <chr>          <chr>          <chr>
## 1 ABBREVIAT~ data_~ https://~ //*[@id=~"ele~  "/*[@id=~"elem~  "/*[@id=~"elem~
## 2 ABDOMINAL~ data_~ https://~ "/*[@id=~"ele~  "/*[@id=~"elem~  "/*[@id=~"elem~
## 3 ABDOMINAL~ data_~ https://~ "/*[@id=~"ele~  "/*[@id=~"elem~  "/*[@id=~"elem~
## 4 ABDOMINAL~ data_~ https://~ "/*[@id=~"ele~  "/*[@id=~"elem~  "/*[@id=~"elem~
## 5 ABLATIVE ~ data_~ https://~ "/*[@id=~"ele~  "/*[@id=~"elem~  "/*[@id=~"elem~
## 6 ABNORMALI~ data_~ https://~ "/*[@id=~"ele~  "/*[@id=~"elem~  "/*[@id=~"elem~
```

This tibble gives a list of all lookups and their associated xpath codes i.e. a direct link to an HTML element, which is the standard way of extracting HTML DOM content. This is where the other functions in the package become powerful.

