

Incorporating external data into a health economic analysis pipeline in R:

How-to and whether-to

Gabriel Rogers
Senior research fellow
Manchester Centre for Health Economics

MANCHESTER
1824

The University of Manchester

R for HTA workshop
York, 2023



"I know nothing about the subject,
but I'm happy to give you my expert opinion."

Driver #1

ACCESSIBILITY OF USEFUL DATA

[Home](#) > [Government](#) > [Government efficiency, transparency and accountability](#) > [Government spending](#)
> [Government Transformation Strategy 2017 to 2020](#)



[Cabinet Office](#)



[Government Digital Service](#)

Policy paper

Government Transformation Strategy: better use of data

Published 9 February 2017



Policy paper

Government better

Published 9 Feb

We will enable better use of data by addressing the technical, ethical and legal issues, specifically focusing on the following priorities:

- making better use of data as an enabler for public services, particularly where those services cross organisational boundaries
- removing barriers to effective data use by all parts of government through the data sharing provisions of the Digital Economy Bill, once it is passed by Parliament
- make better use of data to improve decision making, by building and expanding data science and analytical capability across government, for analysts and non-analysts alike
- managing and using data securely and appropriately, ensuring that public sector workers understand the ethics of data sharing - including what is and what is not permissible
- building a national data infrastructure of registers (authoritative lists that are held once across government) and ensuring that they are secured appropriately
- opening up government data where appropriate
- continuing to open up government services internally and externally through the use of APIs where appropriate
- improving data discovery tools for users both within and beyond government
- transforming the way that government's major repositories of data are stored and managed



Policy paper

Government better

Published 9 Feb

We will enable better use of data by addressing the technical, ethical and legal issues, specifically focusing on the following priorities:

- making better use of data as an enabler for public services, particularly where those services cross organisational boundaries
- removing barriers to effective data use by all parts of government through the data sharing provisions of the Digital Economy Bill, once it is passed by Parliament
- make better use of data to improve decision making, by building and expanding data science and analytical capability across government, for analysts and non-analysts alike
- managing and using data securely and appropriately, ensuring that public sector workers understand the ethics of data sharing - including what is and what is not permissible
- building a national data infrastructure of registers (authoritative lists that are held once across government) and ensuring that they are secured appropriately
- opening up government data where appropriate
- continuing to open up government services internally and externally through the use of APIs where appropriate
- improving data discovery tools for users both within and beyond government
- transforming the way that government's major repositories of data are stored and managed



Supporting open data and transparency

Open data is data that can be used and shared by anyone, for any purpose. We make this data publicly available to improve transparency in health and care.

Page contents

[Top of page](#)[The benefits of open data](#)[How NHS Digital promotes open data](#)[Access open health data](#)[We make a broad range of data available](#)

The benefits of open data

Open data, put simply, is data that can be accessed, used and shared, by anyone, for any purpose. Many organisations collect a broad range of data in order to perform their tasks: making this data publicly accessible contributes to the transparency agenda, which is important for:

- improving outcomes and productivity in our public services
- promoting higher quality and more efficient services, choice and accountability
- encouraging economic growth - it enables the development of tools to support users, commissioners and providers of public services

Good open data should be easy to find, free to access and made available in a format that promotes its use. It should also be timely, accurate and well described. Organisations like the [Open Data Institute](#) and the [Open Knowledge Foundation](#) provide more information on open data, including ways to help assess quality and reliability.

Driver #2

REPLICABLE WORKFLOW

Project-oriented workflow

📅 2017/12/12

👤 Jenny Bryan

If the first line of your R script is

```
setwd("C: \Users \jenny \path \that \only \I \have")
```

I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.

EXAMPLES

Sources of web-served data

In approximate, subjective, descending order of painfulness...

Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. `nomisr` → ONS cause of death statistics

Welcome to Nomis

Nomis is a service provided by [Office for National Statistics \(ONS\)](#), the UK's largest independent producer of official statistics. On this website, we publish statistics related to population, society and the labour market at national, regional and local levels. These include data from current and previous censuses.

Information for first-time visitors
[Sign-in or Register](#)

Important note

Census 2021 datasets are now available in the query data tool

Labour Market Profiles

View a labour market profile of an area. Includes some of the data from our key datasets on population, employment, unemployment, qualifications, earnings, benefit claimants and businesses.

[Local Authority Profile](#) (district/county areas)

[Local Enterprise Partnerships Profile](#)

[Combined Authority Profile](#)

[Regional and National Profile](#)

[2010 Parliamentary Constituencies Profile](#)

[2011 Ward Profile](#) (England & Wales only)

Local Area Report

View a [report for a local area](#) such as a parish, ward, village or town. Includes information on the characteristics of people and households in the area.

Data Downloads

Create a data download from one of our full range of data sets. Data is available at a very detailed level.

[Query data](#)

Download figures from a single data set.



Census Statistics

[2021 Data catalogue](#)

Browse by table type and number.



[2011 Data catalogue](#)

Browse by table type and number, or view by release.

[2011 Search by topic \(table finder\)](#)

Search by keyword and geography type.

[Earlier censuses](#)

Find data from 2001, 1991, 1981 and 1961 censuses.

nomisr



`nomisr` is for accessing UK official statistics from the [Nomis](#) database through R. Nomis contains data from the Census, the Labour Force Survey, DWP benefit statistics and other economic and demographic data, and is maintained on behalf of the Office for National Statistics by the University of Durham.

The `nomisr` package provides functions to find what data is available, the variables and query options for different datasets and a function for downloading data. `nomisr` returns data in [tibble](#) format. Most of the data available through `nomisr` is based around statistical geographies, with a handful of exceptions.

The package is for demographers, economists, geographers, public health researchers and any other researchers who are interested in geographic factors. The package aims to aid reproducibility, reduce the need to manually download area profiles, and allow easy linking of different datasets covering the same geographic area.

Usage

```
nomis_get_data(  
  id,  
  time = NULL,  
  date = NULL,  
  geography = NULL,  
  sex = NULL,  
  measures = NULL,  
  additional_queries = NULL,  
  exclude_missing = FALSE,  
  select = NULL,  
  tidy = FALSE,  
  tidy_style = "snake_case",  
  query_id = NULL,  
  ...  
)
```

ONS NOMIS via nomisr

What proportion of deaths are due to CVD?

ONS NOMIS via nomisr

What proportion of deaths are due to CVD?

```
# A tibble: 720 × 46
  DATE DATE_NAME DATE_CODE DATE_TYPE DATE_TYPECODE DATE_SORTORDER GEOGRAPHY GEOGRAPHY_NAME  GEOGRAPHY_CODE GEOGRAPHY_TYPE GEOGRAPHY_TYPECODE GEOGRAPHY_SORTORDER
  <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <chr> <chr> <chr> <chr> <dbl> <dbl>
1 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
2 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
3 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
4 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
5 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
6 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
7 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
8 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
9 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
10 2013 2013 2013 date 0 2092957703 England and Wales K04000001 countries 499 0
# i 710 more rows
# i 34 more variables: CAUSE_OF_DEATH <dbl>, CAUSE_OF_DEATH_NAME <chr>, CAUSE_OF_DEATH_CODE <chr>, CAUSE_OF_DEATH_TYPE <chr>, CAUSE_OF_DEATH_TYPECODE <dbl>,
# CAUSE_OF_DEATH_SORTORDER <dbl>, GENDER <dbl>, GENDER_NAME <chr>, GENDER_CODE <dbl>, GENDER_TYPE <chr>, GENDER_TYPECODE <dbl>, GENDER_SORTORDER <dbl>, AGE <dbl>,
# AGE_NAME <chr>, AGE_CODE <dbl>, AGE_TYPE <chr>, AGE_TYPECODE <dbl>, AGE_SORTORDER <dbl>, MEASURE <dbl>, MEASURE_NAME <chr>, MEASURE_CODE <dbl>,
# MEASURE_TYPE <chr>, MEASURE_TYPECODE <dbl>, MEASURE_SORTORDER <dbl>, MEASURES <dbl>, MEASURES_NAME <chr>, OBS_VALUE <dbl>, OBS_STATUS <chr>,
# OBS_STATUS_NAME <chr>, OBS_CONF <lgl>, OBS_CONF_NAME <chr>, URN <chr>, RECORD_OFFSET <dbl>, RECORD_COUNT <dbl>
# i Use `print(n = ...)` to see more rows
```

ONS NOMIS via nomisr

What proportion of deaths are due to CVD?

```
1 tblCoD ← nomisr::nomis_get_data(id      = "NM_161_1",
2                                     date     = 2013:2020,
3                                     geography = 2092957703,
4                                     sex      = 1:2,
5                                     age      = 1:20,
6                                     measure   = 1,
7                                     measures  = 20100,
8                                     cause_of_death = c(0,9))
9
10 tblCoD ▷
11   dplyr::mutate(CoD = ifelse(CAUSE_OF_DEATH==9, "CVD", "All")) ▷
12   dplyr::select(Year    = DATE,
13                 Sex      = GENDER_NAME,
14                 AgeCat   = AGE_NAME,
15                 AgeC     = AGE_CODE,
16                 N        = OBS_VALUE,
17                 CoD) ▷
18   tidyr::pivot_wider(values_from = N,
19                       names_from = CoD) ▷
20   dplyr::mutate(pCVD  = CVD / All,
21                 AgeLo = c(0,1,1:18*5)[AgeC],
22                 AgeHi = c(1,1:18*5,999)[AgeC])
```

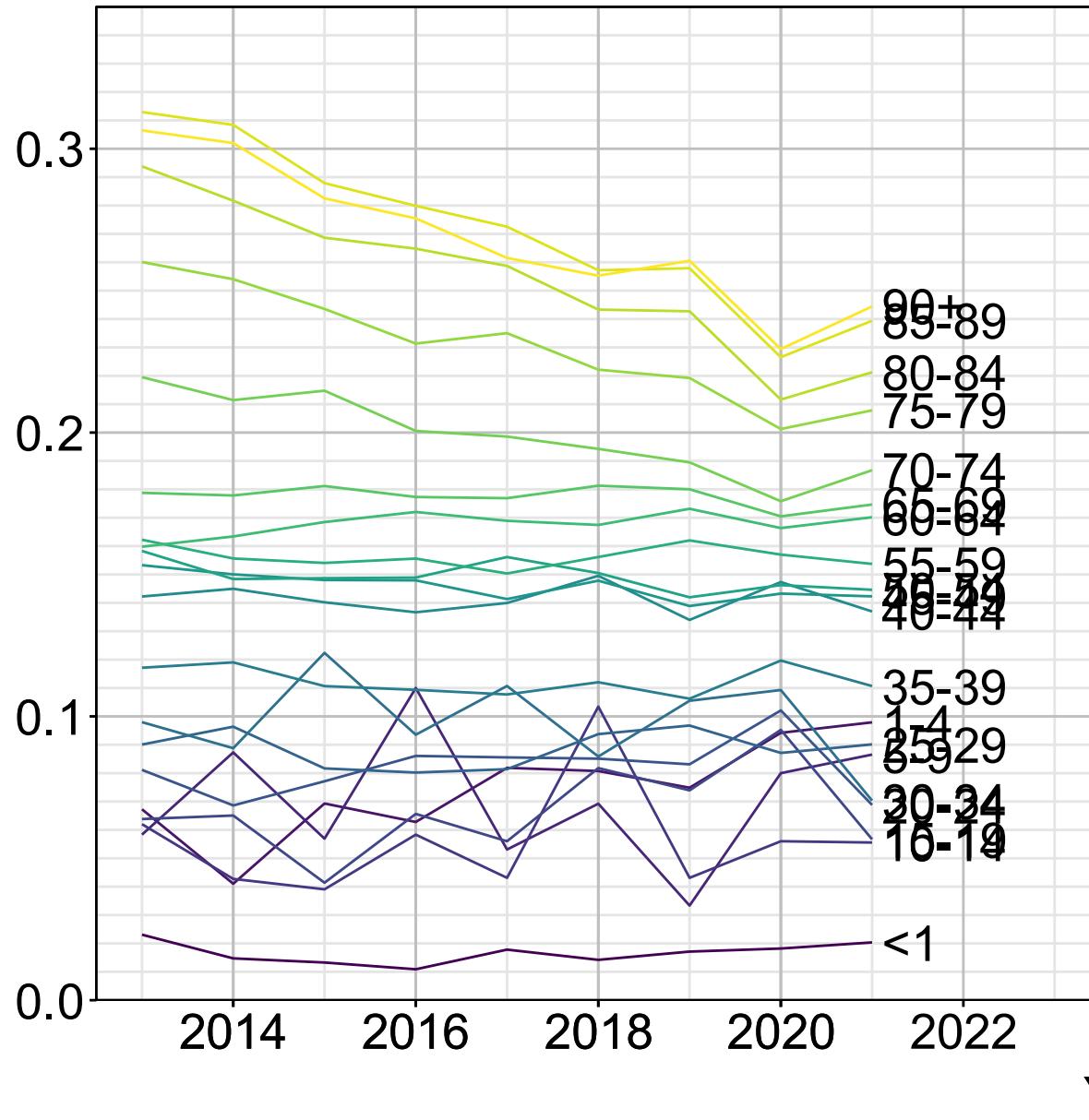
ONS NOMIS via nomisr

What proportion of deaths are due to CVD?

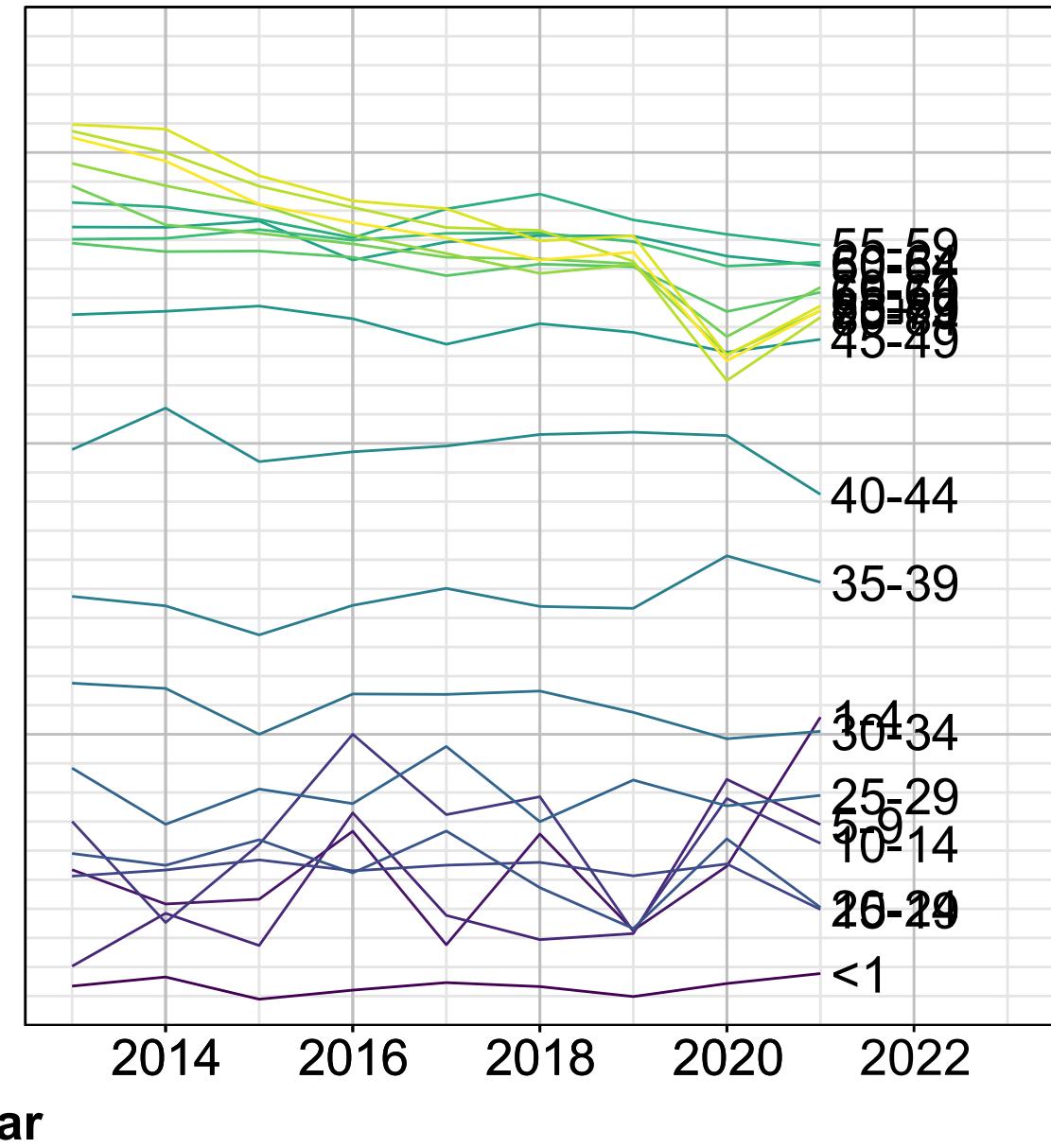
```
# A tibble: 320 × 9
  Year Sex   AgeCat      AgeC   All    CVD    pCVD AgeLo AgeHi
  <dbl> <chr> <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 2013 Male  Aged under 1     1  1563    21 0.0134     0     1
2 2013 Male  Aged 1 to 4     2   262     14 0.0534     1     5
3 2013 Male  Aged 5 to 9     3   148      3 0.0203     5    10
4 2013 Male  Aged 10-14     4   157     11 0.0701    10    15
5 2013 Male  Aged 15-19     5   527     27 0.0512    15    20
6 2013 Male  Aged 20-24     6   881     52 0.0590    20    25
7 2013 Male  Aged 25-29     7  1143    101 0.0884   25    30
8 2013 Male  Aged 30-34     8  1497    176 0.118    30    35
9 2013 Male  Aged 35-39     9  2076    306 0.147    35    40
10 2013 Male  Aged 40-44    10  3371    667 0.198   40    45
# i 310 more rows
# i Use `print(n = ...)` to see more rows
```

Proportion of deaths due to CVD

Female



Male



Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. `nomisr` → ONS cause of death statistics

Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. nomisr → ONS cause of death statistics
- API serving JSON read via jsonlite
 - e.g. NHS Open Data Portal → English Prescribing Dataset

[Home](#)[Data](#)[News](#)[Documentation](#)[About](#)[Log in](#)

BETA

Your feedback is important to us! Help us serve you better by filling out our [short survey](#)!

Welcome to the Open Data Portal

Our home of open data.

- All our data is free for use and re-use
- You can view, filter, and download data and meta data
- Built using open-source tools

[#NHS Prescription Se...](#)[#VDPS](#)[#NHS Prescriptions](#)[#NHS Pensions](#)



Community Prescribing & Dispensing

Datasets on the prescribing and dispensing of medicines in the community [read more](#)

Followers Datasets

7 4

Themes

Community Prescribi...

4

Tags

Datasets

Activity Stream

About

Search datasets...



4 datasets found

Order by:

Name Descending ▾

June 16, 2021, 3:46 PM (UTC+01:00)

[Prescription Cost Analysis \(PCA\) Monthly Administrative Data](#)

CSV

PCA contains information on all prescriptions items dispensed in the community in England on a monthly basis and submitted to the NHSBSA for reimbursement. This data mirrors the...

[Read More](#)

June 8, 2021, 3:10 PM (UTC+01:00)

[Prescription Cost Analysis \(PCA\) Annual Statistics](#)

CSV

The PCA release is an Official Statistic with National Statistic designation. The lowest granularity of data in the publication for both financial and calendar year is added...

[Read More](#)

NHS Open Data Portal API

Accessing English Prescribing Data via jsonlite

NHS Open Data Portal API

Accessing English Prescribing Data via jsonlite

```
1 strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id=EPD_201401&sql="
2 strSQL ← paste("SELECT YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",
3                  "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE,",
4                  "sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC, sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ",
5                  "FROM `EPD_201401` WHERE BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",
6                  "GROUP BY YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR,",
7                  "BNF_CODE, BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE")
8
9 jsonRet ← paste0(strURL, strSQL) ▷
10 URLEncode() ▷
11 jsonlite::fromJSON()
12
13 tblSQL ← jsonRet$result$result$records ▷
14 dplyr::as_tibble() ▷
15 dplyr::arrange(BNF_CODE)
16
17 tblSQL
```

```
"https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id=EPD_201401&sql=SELECT%20YEAR_MONTH,%20BNF_CHEMICAL_SUBSTANCE,%20
CHEMICAL_SUBSTANCE_BNF_DESCR,%20BNF_CODE,%20BNF_DESCRIPTION,%20BNF_CHAPTER_PLUS_CODE,%20sum(TOTAL_QUANTITY)%20AS%20Quantity,%20sum(NIC)%20A
S%20NIC,%20sum(ACTUAL_COST)%20AS%20Cost,%20avg(ADQUSAGE)%20AS%20ADQ%20FROM%20%60EPD_201401%60%20WHERE%20BNF_CHEMICAL_SUBSTANCE%20LIKE%20'06
0102%25'%20GROUP%20BY%20YEAR_MONTH,%20BNF_CHEMICAL_SUBSTANCE,%20CHEMICAL_SUBSTANCE_BNF_DESCR,%20BNF_CODE,%20BNF_DESCRIPTION,%20BNF_CHAPTER_
PLUS_CODE"
```

NHS Open Data Portal API

Accessing English Prescribing Data via jsonlite

```
# A tibble: 128 × 7
  BNF_CODE      CHEMICAL_SUBSTANCE_BNF_DESCR BNF_DESCRIPTION      Quantity      NIC      Cost     ADQ
  <chr>          <chr>                         <chr>                  <int>       <dbl>     <dbl>   <dbl>
1 0601021A0AAAAAA Glimepiride                Glimepiride 2mg tab...    736153  27976. 27516. 104.
2 0601021A0AAABAB Glimepiride                Glimepiride 1mg tab...    587778  21345. 21031. 48.9
3 0601021A0AAACAC Glimepiride                Glimepiride 3mg tab...    394870  76599. 72697. 147.
4 0601021A0AAADAD Glimepiride                Glimepiride 4mg tab...    751773  34587. 34023. 223.
5 0601021A0AAAHAH Glimepiride                Glimepiride 6mg/5ml...     11     577.  543.   6.6
6 0601021A0BBAAAA Glimepiride                Amaryl 2mg tablets      4155    987.  935.  56.9
7 0601021A0BBABAB Glimepiride                Amaryl 1mg tablets      4638    669.  634.  32.2
8 0601021A0BBACAC Glimepiride                Amaryl 3mg tablets      1794    643.  607.  84.1
9 0601021A0BBADAD Glimepiride                Amaryl 4mg tablets      3729    1770. 1670. 98.1
10 0601021H0AAAAAAA Glibenclamide            Glibenclamide 2.5mg...   54889  34699. 32701. 16.1
# i 118 more rows
# i Use `print(n = ...)` to see more rows
```

NHS Open Data Portal API

Making it functional

```
1 fnNHSOpenData <- function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {  
2   strURL <- "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="  
3   res <- paste0(dataset, "_", Year, sprintf("%02d", Month))  
4   strSQL <- paste("SELECT", strSelect, "FROM `", res, "`")  
5   if (strWhere != "") strSQL <- paste(strSQL, "WHERE", strWhere)  
6   if (strGroupBy != "") strSQL <- paste(strSQL, "GROUP BY", strGroupBy)  
7  
8   url <- glue(strURL, strSQL) %>  
9     URLencode()  
10  
11  tibble(Dataset = res) %>  
12    bind_cols(fromJSON(url)) %>  
13      as_tibble()  
14 }
```

NHS Open Data Portal API

Making it functional

```
1 fnNHSOpenData <- function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {  
2   strURL <- "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="  
3   res <- paste0(dataset, "_", Year, sprintf("%02d", Month))  
4   strSQL <- paste("SELECT", strSelect, "FROM `", res, "`")  
5   if (strWhere != "") strSQL <- paste(strSQL, "WHERE", strWhere)  
6   if (strGroupBy != "") strSQL <- paste(strSQL, "GROUP BY", strGroupBy)  
7  
8   url <- glue(strURL, strSQL) %>  
9     URLencode()  
10  
11  tibble(Dataset = res) %>  
12    bind_cols(fromJSON(url)) %>  
13      as_tibble()  
14 }  
15  
16 fnNHSOpenData(dataset      = "EPD",  
17                 Year        = 2014,  
18                 Month       = 1,  
19                 strSelect   = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE, BNF_DESCRIPTION,",  
20                               "BNF_CHAPTER_PLUS_CODE, sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC,",  
21                               "sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ"),  
22                 strWhere    = "BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",  
23                 strGroupBy = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",  
24                               "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE")) %>%tblDM
```

NHS Open Data Portal API

Making it functional - calling via purrr::map

```
1 fnNHSOpenData <- function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {  
2   strURL <- "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="  
3   res <- paste0(dataset, "_", Year, sprintf("%02d", Month))  
4   strSQL <- paste("SELECT", strSelect, "FROM `", res, "`")  
5   if (strWhere != "") strSQL <- paste(strSQL, "WHERE", strWhere)  
6   if (strGroupBy != "") strSQL <- paste(strSQL, "GROUP BY", strGroupBy)  
7  
8   url <- glue(strURL, strSQL) %>  
9     URLencode()  
10  
11  tibble(Dataset = res) %>  
12    bind_cols(fromJSON(url)) %>  
13      as_tibble()  
14 }  
15  
16 crossing(Year      = 2014:2023,  
17           Month     = 1:12,  
18           dataset   = "EPD",  
19           strSelect  = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE, BNF_DESCRIPTION,",  
20                         "BNF_CHAPTER_PLUS_CODE, sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC,",  
21                         "sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ"),  
22           strWhere    = "BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",  
23           strGroupBy = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",  
24                         "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE")) %>  
25  filter(Year < 2023 | Month < 4) %>  
26  pmap(.f      = fnNHSOpenData,  
27        .progress = TRUE) %>  
28  list_rbind() → tblAllDM
```

NHS Open Data Portal API

Making it functional

```
# A tibble: 111 × 6
  Year Month dataset strSelect          strWhere          strGroupBy
  <chr> <chr> <chr>   <chr>
1 2014  1     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
2 2014  2     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
3 2014  3     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
4 2014  4     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
5 2014  5     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
6 2014  6     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
7 2014  7     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
8 2014  8     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
9 2014  9     EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
10 2014 10    EPD    YEAR_MONTH, BNF_CHEMICAL_SUBSTAN... BNF_CHEMICAL_SUBSTANCE LIKE '060...
# i 101 more rows
# i Use `print(n = ...)` to see more rows
```

NHS Open Data Portal API

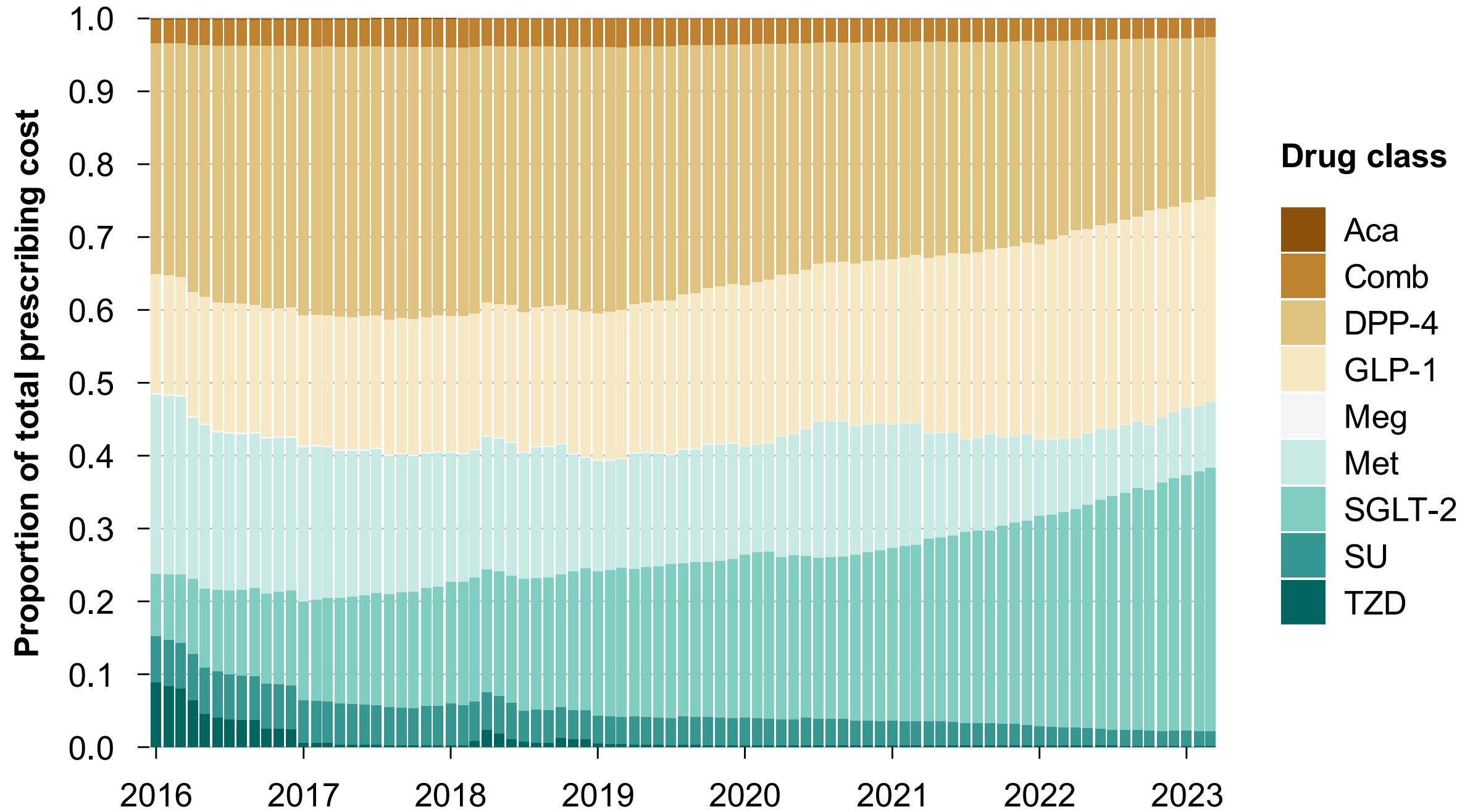
Making it functional - calling via purrr::map

```
1 fnNHSOpenData <- function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {  
2   strURL <- "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="  
3   res <- paste0(dataset, "_", Year, sprintf("%02d", Month))  
4   strSQL <- paste("SELECT", strSelect, "FROM `", res, "`")  
5   if (strWhere != "") strSQL <- paste(strSQL, "WHERE", strWhere)  
6   if (strGroupBy != "") strSQL <- paste(strSQL, "GROUP BY", strGroupBy)  
7  
8   url <- glue(strURL, strSQL) %>  
9     URLencode()  
10  
11  tibble(Dataset = res) %>  
12    bind_cols(fromJSON(url)) %>  
13      as_tibble()  
14 }  
15  
16 crossing(Year      = 2014:2023,  
17           Month     = 1:12,  
18           dataset   = "EPD",  
19           strSelect  = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE, BNF_DESCRIPTION,",  
20                         "BNF_CHAPTER_PLUS_CODE, sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC,",  
21                         "sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ"),  
22           strWhere    = "BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",  
23           strGroupBy = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",  
24                         "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE")) %>  
25  filter(Year < 2023 | Month < 4) %>  
26  pmap(.f      = fnNHSOpenData,  
27        .progress = TRUE) %>  
28  list_rbind() %>tblAllDM
```

NHS Open Data Portal API

Making it functional

```
# A tibble: 19,973 × 11
  Dataset    BNF_CODE      ADQ      NIC CHEMICAL_SUBSTANCE_BNF_DESCR   Cost BNF CHAPTER_PLUS_CODE YEAR_MONTH BNF_CHEMICAL_SUBSTANCE BNF_DESCRIPTION      Quantity
  <chr>      <chr>       <dbl>     <dbl> <chr>                         <dbl> <chr>           <int> <chr>           <chr>           <chr>           <dbl>
1 EPD_201401 0601021M0AAARAR  39.5    381083. Gliclazide                361950. 06: Endocrine System 201401 0601021M0  Gliclazide 40mg ta...  3175699
2 EPD_201401 0601023R0AAABAB  28.7    19485. Repaglinide                 18535. 06: Endocrine System 201401 0601023R0  Repaglinide 1mg ta...  199203
3 EPD_201401 0601023A0AAAAAA  42.8    27890. Acarbose                  26515. 06: Endocrine System 201401 0601023A0  Acarbose 50mg tabl...  266472
4 EPD_201401 0601023AIBBAAA  0       29750. Lixisenatide                28018. 06: Endocrine System 201401 0601023AI  Lyxumia 20microgra...  1099
5 EPD_201401 0601023R0AAAAAA  12.9    13815. Repaglinide                 13158. 06: Endocrine System 201401 0601023R0  Repaglinide 500mic...  149612
6 EPD_201401 0601021M0AAAAAA  260.   1322231. Gliclazide                1262961. 06: Endocrine System 201401 0601021M0  Gliclazide 80mg ta...  35589459
7 EPD_201401 0601021X0AAADAD  49.4    341790. Tolbutamide                321955. 06: Endocrine System 201401 0601021X0  Tolbutamide 500mg ...  370644
8 EPD_201401 0601023B0AAACAC  196.   86610. Pioglitazone hydrochloride  82396. 06: Endocrine System 201401 0601023B0  Pioglitazone 45mg ...  1040542
9 EPD_201401 0601022B0AAASAS  171.   1921122. Metformin hydrochloride  1818374. 06: Endocrine System 201401 0601022B0  Metformin 500mg mo...  20221848
10 EPD_201401 0601023ACAAABAB 31.8   216724. Saxagliptin                 204329. 06: Endocrine System 201401 0601023AC  Saxagliptin 2.5mg ...  192037
# i 19,963 more rows
# i Use `print(n = ...)` to see more rows
```



Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. nomisr → ONS cause of death statistics
- API serving JSON read via jsonlite
 - e.g. NHS Open Data Portal → English Prescribing Dataset

Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. `nomisr` → ONS cause of death statistics
- API serving JSON read via `jsonlite`
 - e.g. NHS Open Data Portal → English Prescribing Dataset
- Archived .XLSXs read via `readxl`
 - e.g. ONS lifetables

Search for a keyword(s) or time series ID



[Home](#) > [People, population and community](#) > [Births, deaths and marriages](#) > [Life expectancies](#)

Dataset

National life tables: England



Contact:
Julian Buxton

Release date:
23 September 2021

Next release:
To be announced

About this Dataset

Period life expectancy by age and sex for England. Each national life table is based on population estimates, births and deaths for a period of three consecutive years. Tables are published annually.

Edition in this dataset

Current edition of this dataset Current



[xlsx \(485.6 KB\)](#)



[Previous versions](#) of this data are available.

[View all data related to life expectancies](#)

Contact details for this dataset

Julian Buxton
pop.info@ons.gov.uk
+44 1329 444661

Publications that use this data

[National life tables – life expectancy in the UK](#)

ONS life tables

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	National Life Tables, England, period expectation of life, based on data for the years 2018-2020													
2	This worksheet contains two tables, presented horizontally with one blank column in between the tables.													
3	This worksheet uses notation in the column headers, please see the notation worksheet for explanations.													
4	Back to contents													
5	National life tables, Males							National life tables, Females						
6	age	mx	qx	lx	dx	ex		age	mx	qx	lx	dx	ex	
7	0	0.004253	0.004244	100000.0	424.4	79.33		0	0.003525	0.003519	100000.0	351.9	83.12	
8	1	0.000231	0.000231	99575.6	23.0	78.67		1	0.000211	0.000211	99648.1	21.0	82.42	
9	2	0.000128	0.000128	99552.5	12.7	77.69		2	0.000113	0.000113	99627.1	11.3	81.43	
10	3	0.000099	0.000099	99539.8	9.8	76.70		3	0.000093	0.000093	99615.9	9.2	80.44	
11	4	0.000090	0.000090	99530.0	8.9	75.71		4	0.000061	0.000061	99606.6	6.0	79.45	
12	5	0.000077	0.000077	99521.1	7.6	74.71		5	0.000079	0.000079	99600.6	7.8	78.45	
13	6	0.000081	0.000081	99513.4	8.0	73.72		6	0.000069	0.000069	99592.7	6.9	77.46	
14	7	0.000068	0.000068	99505.4	6.8	72.73		7	0.000051	0.000051	99585.8	5.0	76.47	
15	8	0.000065	0.000065	99498.6	6.4	71.73		8	0.000053	0.000053	99580.8	5.2	75.47	
16	9	0.000062	0.000062	99492.2	6.2	70.74		9	0.000056	0.000056	99575.6	5.5	74.47	
17	10	0.000073	0.000073	99486.0	7.3	69.74		10	0.000065	0.000065	99570.0	6.5	73.48	
18	11	0.000074	0.000074	99478.7	7.3	68.75		11	0.000056	0.000056	99563.6	5.6	72.48	
19	12	0.000102	0.000102	99471.4	10.1	67.75		12	0.000054	0.000054	99558.0	5.3	71.49	
20	13	0.000116	0.000116	99461.2	11.5	66.76		13	0.000088	0.000088	99552.7	8.7	70.49	
21	14	0.000124	0.000124	99449.7	12.3	65.76		14	0.000094	0.000094	99544.0	9.4	69.50	
22	15	0.000169	0.000169	99437.4	16.8	64.77		15	0.000102	0.000102	99534.6	10.2	68.50	
23	16	0.000190	0.000190	99420.5	18.9	63.78		16	0.000129	0.000129	99524.4	12.9	67.51	
24	17	0.000284	0.000284	99401.6	28.2	62.80		17	0.000157	0.000157	99511.6	15.6	66.52	
25	18	0.000373	0.000373	99373.4	37.1	61.81		18	0.000205	0.000205	99496.0	20.4	65.53	
26	19	0.000415	0.000415	99336.3	41.2	60.84		19	0.000202	0.000202	99475.5	20.1	64.54	
27	20	0.000524	0.000524	99295.1	52.0	59.86		20	0.000177	0.000177	99455.5	17.6	63.56	
28	21	0.000473	0.000473	99243.1	47.0	58.89		21	0.000195	0.000195	99437.9	19.4	62.57	
29	22	0.000463	0.000463	99196.1	45.9	57.92		22	0.000232	0.000232	99418.5	23.1	61.58	
30	23	0.000478	0.000478	99150.3	47.4	56.95		23	0.000200	0.000200	99395.4	19.8	60.59	
31	24	0.000514	0.000514	99102.8	51.0	55.97		24	0.000215	0.000215	99375.5	21.4	59.60	
32	25	0.000540	0.000540	99051.9	53.4	55.00		25	0.000251	0.000251	99354.2	24.9	58.62	
33	26	0.000567	0.000567	98998.4	56.1	54.03		26	0.000253	0.000253	99329.2	25.2	57.63	
34	27	0.000585	0.000585	98942.3	57.9	53.06		27	0.000290	0.000290	99304.1	28.8	56.65	
35	28	0.000629	0.000629	98884.5	62.2	52.09		28	0.000299	0.000299	99275.3	29.7	55.66	
36	29	0.000657	0.000657	98822.3	64.9	51.13		29	0.000318	0.000318	99245.6	31.6	54.68	
37	30	0.000730	0.000730	98757.4	72.1	50.16		30	0.000374	0.000374	99214.0	37.1	53.70	
38	31	0.000778	0.000778	98685.3	76.8	49.19		31	0.000366	0.000366	99176.9	36.3	52.72	
39	32	0.000775	0.000775	98608.5	76.4	48.23		32	0.000438	0.000438	99140.5	43.4	51.74	
40	33	0.000888	0.000888	98532.1	87.5	47.27		33	0.000472	0.000472	99097.2	46.8	50.76	
41	34	0.000918	0.000917	98444.6	90.3	46.31		34	0.000549	0.000549	99050.4	54.4	49.78	
42	35	0.000990	0.000990	98364.3	97.3	45.35		35	0.000560	0.000560	98996.0	55.4	48.81	

ONS life tables

```
1 strURL ← paste0("https://www.ons.gov.uk/file?",
2                     "uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
3                     "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx")
4
5 tmp ← tempfile(fileext = ".xlsx")
6 httr::GET(strURL, httr::write_disk(tmp))
7
8 tbLLT ← readxl::read_excel(path  = tmp,
9                             sheet = "2018-2020",
10                            range = "A6:M107")
```

ONS life tables

```
# A tibble: 101 × 14
  age...1   mx...2   qx...3   lx...4   dx...5   ex...6 ...7   age...8   mx...9   qx...10  lx...11  dx...12  ex...13   yr
  <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <lgl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1     0 0.00425  0.00424  100000   424.    79.3 NA        0 0.00352  0.00352  100000   352.    83.1 2018
2     1 0.000231  0.000231  99576.   23      78.7 NA        1 0.000211  0.000211  99648.   21      82.4 2018
3     2 0.000128  0.000128  99552.   12.7    77.7 NA        2 0.000113  0.000113  99627.   11.3    81.4 2018
4     3 0.000099  0.000099  99540.   9.8     76.7 NA        3 0.000093  0.000093  99616.   9.2     80.4 2018
5     4 0.00009  0.00009   99530    8.9     75.7 NA        4 0.000061  0.000061  99607.   6       79.4 2018
6     5 0.000077  0.000077  99521.   7.6     74.7 NA        5 0.000079  0.000079  99601.   7.8     78.4 2018
7     6 0.000081  0.000081  99513.   8       73.7 NA        6 0.000069  0.000069  99593.   6.9     77.5 2018
8     7 0.000068  0.000068  99505.   6.8     72.7 NA        7 0.000051  0.000051  99586.   5       76.5 2018
9     8 0.000065  0.000065  99499.   6.4     71.7 NA        8 0.000053  0.000053  99581.   5.2     75.5 2018
10    9 0.000062  0.000062  99492.   6.2     70.7 NA        9 0.000056  0.000056  99576.   5.5     74.5 2018
# i 91 more rows
# i Use `print(n = ...)` to see more rows
```

ONS life tables

```
1 strURL ← paste0("https://www.ons.gov.uk/file?",
2                     "uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
3                     "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx")
4
5 tmp ← tempfile(fileext = ".xlsx")
6 httr::GET(strURL, httr::write_disk(tmp))
7
8 tbLLT ← readxl::read_excel(path = tmp,
9                             sheet = "2018-2020",
10                            range = "A6:M107")
11
12 tbLLT ▷
13   dplyr::select(!c(...7, age...8)) ▷
14   dplyr::rename(age = age...1) ▷
15   tidyr::pivot_longer(cols      = !age,
16                         names_to  = "tmp",
17                         values_to = "val") ▷
18   tidyr::separate(col  = tmp,
19                     into = c("par", "sex")) ▷
20   dplyr::mutate(sex      = ifelse(as.numeric(sex)<7, "Male", "Female")) ▷
21   tidyr::pivot_wider(id_cols  = c(age, sex),
22                       values_from = val,
23                       names_from  = par)
```

ONS life tables

```
# A tibble: 202 × 7
  age sex      mx      qx      lx      dx      ex
  <dbl> <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1     0 Male  0.00425  0.00424 100000  424.    79.3
2     0 Female 0.00352  0.00352 100000  352.    83.1
3     1 Male  0.000231 0.000231 99576.   23     78.7
4     1 Female 0.000211 0.000211 99648.   21     82.4
5     2 Male  0.000128 0.000128 99552.   12.7   77.7
6     2 Female 0.000113 0.000113 99627.   11.3   81.4
7     3 Male  0.000099 0.000099 99540.   9.8    76.7
8     3 Female 0.000093 0.000093 99616.   9.2    80.4
9     4 Male  0.00009  0.00009  99530    8.9    75.7
10    4 Female 0.000061 0.000061 99607.   6      79.4
# i 192 more rows
# i Use `print(n = ...)` to see more rows
```

ONS life tables

Making it functional

```

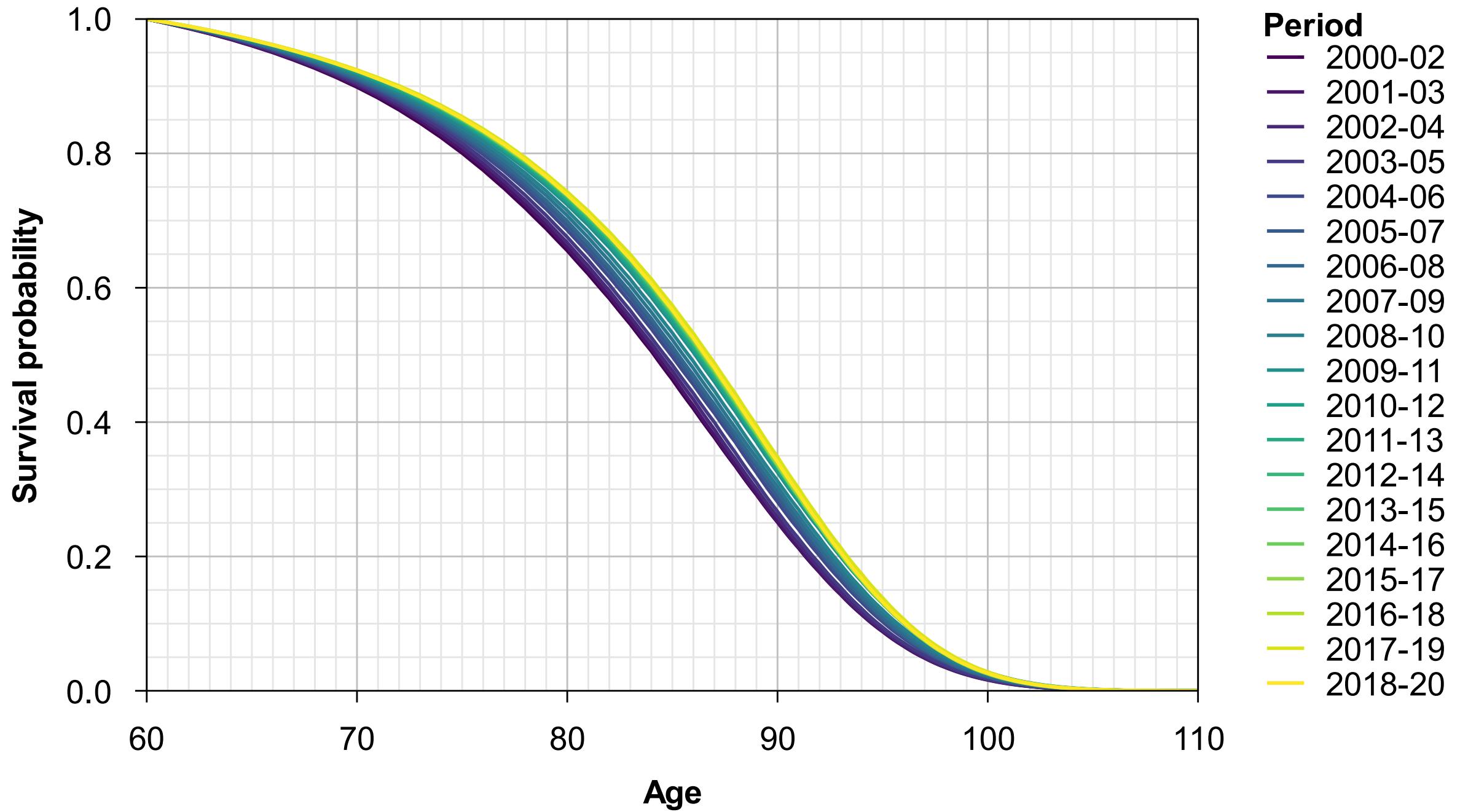
1 fnONSlifeTable<- function(years      = 1980:2018,
2                           ages       = 0:100,
3                           sexes      = c("Male", "Female"),
4                           countries   = c("UK", "Eng", "Sco", "Wal", "EngWal"),
5                           spans       = c("3-year", "single year"),
6                           extrapolate = FALSE,
7                           maxExtrap   = 120,
8                           widthExtrap = 10) {
9
10  countries <- match.arg(countries)
11  spans <- match.arg(spans)
12  url <- paste0(switch(spans,
13                 "3-year"     = "https://www.ons.gov.uk/file?
uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
14                 "single year" = "https://www.ons.gov.uk/file?
uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fsingleyearlifetablesuk1980to2018%2fsingleye
arlfetables"),
15                 switch(spans,
16                     "3-year"     = switch(countries,
17                         "Eng"       = "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx",
18                         "Sco"       = "scotlandreferencetables%2fcurrent/nationallifetables3yrso.xlsx",
19                         "Wal"       = "walesreferencetables%2fcurrent/nationallifetables3yrwal.xlsx",
20                         "EngWal"    = "englandandwalesreferencetables%2fcurrent/nationallifetables3yrew.xlsx",
21                         "unitedkingdomreferencetables%2fcurrent/nationallifetables3yruk.xlsx"),
22                     "single year" = switch(countries,
23                         "Eng"       = "england/singleyearlifetablese.xlsx",
24                         "Sco"       = "scotland/singleyearlifetables.xlsx",
25                         "Wal"       = "wales/singleyearlifetables1.xlsx",
26                         "EngWal"    = "englandandwales/singleyearlifetablesew.xlsx",
27                         "uk/singleyearlifetablesuk.xlsx")))
28
29  GET(url, write_disk(tmp <- tempfile(fileext = ".xlsx")))
30  tblYrs <- tibble(path = tmp,
31                    sheet = tmp %>%
32                      excel_sheets() %>%
33                      str_subset(pattern = if_else(spans=="3-year", "\d{4}-\d{4}", "\d{4}")),
34                      range = "A6:H107")
35
36  tbLLT <- tblYrs %>%
37    filter(substr(sheet, 1, 4) %in% years) %>%
38    pmap_dfr(fnReadLifeTable)

```

```

36  tbLLT %>%
37  dplyr::select(!c(...7, age...8)) %>%
38  rename(age = age...1) %>%
39  pivot_longer(cols      = !c(age, yr),
40               names_to = "tmp",
41               values_to = "val") %>%
42  separate(col = tmp,
43            into = c("par", "sex")) %>%
44  mutate(sex = ifelse(as.numeric(sex)<7, "Male", "Female"),
45        year_from = as.numeric(substr(yr, 1, 4)),
46        year_to = year_from + if_else(spans=="3-year", 2, 0)) %>%
47  pivot_wider(id_cols = c(age, year_from, year_to, sex),
48              values_from = val,
49              names_from = par) %>%
50  rowwise() %>%
51  mutate(lx = ifelse(min(ages)==0, lx, NA),
52        dx = ifelse(min(ages)==0, dx, NA)) %>%
53  filter(year_from %in% years,
54        age    %in% ages,
55        sex    %in% sexes) %>%
56  arrange(-year_from, sex, age) %>%
57  group_by(sex, year_from, year_to)
58
59  if (extrapolate) {
60    maxA <- max(tbLLT$age)
61    tbLLT %>%
62      slice_tail(n=widthExtrap) %>%
63      nest() %>%
64      mutate(lm_obj = map(data, ~ lm(log(mx)~log(age), data=.x)),
65             preddat = list(tibble(age = (maxA+1):(maxExtrap+1))),
66             pred = map2(lm_obj, preddat, predict)) %>%
67      unnest(c(preddat, pred)) %>%
68      mutate(mx = exp(pred)) %>%
69      select(:, data, lm_obj, pred)) %>%
70      bind_rows(tbLLT)
71  }
72  if (extrapolate | min(ages)>0) {
73    tbLLT %>%
74      arrange(-year_from, sex, age) %>%
75      mutate(qx = coalesce(qx, fnRateToProb(mx)),
76             llx = if_else(row_number()==1, 100000, lag(100000*cumprod(1-qx), 1)),
77             lx = coalesce(lx, llx),
78             dx = coalesce(dx, lx*qx),
79             Lx = (lx+lead(lx,1))/2,
80             Tx = rev(cumsum(rev(coalesce(Lx, 0)))),
81             Ex = Tx/lx,
82             ex = coalesce(ex, Ex)) %>%
83      filter(age<=maxExtrap) %>%
84      dplyr::select(!c(llx, Lx, Tx, Ex))
85  }
86  tbLLT %>%
87  ungroup()
88 }

```



Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. `nomisr` → ONS cause of death statistics
- API serving JSON read via `jsonlite`
 - e.g. NHS Open Data Portal → English Prescribing Dataset
- Archived .XLSXs read via `readxl`
 - e.g. ONS lifetables

Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. nomisr → ONS cause of death statistics
- API serving JSON read via jsonlite
 - e.g. NHS Open Data Portal → English Prescribing Dataset
- Archived .XLSXs read via readxl
 - e.g. ONS lifetables
- Archived .CSVs in .ZIPs within .ZIPs
 - e.g. National Cost Collection (FKA NHS Reference Costs)



About us

Our work

Commissioning

Get involved

Coronavirus

Costing in the NHS

National Cost Collection for
the NHS

Costing glossary

Approved Costing Guidance

[Home](#) > [Costing in the NHS](#) > National Cost Collection for the NHS

National Cost Collection for the NHS

The National Cost Collection publication comprises aggregated costs (the average unit cost of providing defined services to NHS patients in England) and patient-level costs/PLICS (a cost based on the specific interactions a patient has, and the events related to their healthcare activity).

NHS providers submit costs annually. These costs are used to inform a number of work streams including the PLICS portal, the Model Hospital, the Getting It Right First Time (GIRFT) programme and NHS Payment Scheme prices. They are a key source of information about the cost of NHS services.



About us

Our work

Costing in the NHS

National Cost Collection for the NHS

Costing glossary

Approved Costing Guidance

2021/22 National Cost Collection data

As in 2020/21, we have not produced a report for this year's National Cost Collection. However, the data itself is published in its entirety, presented in the following ways:

- the national schedules of NHS costs
- the National Cost Collection Index (NCCI)
- the reconciliation statement
- a database of source data

Data spreadsheets

- [National schedule of NHS costs](#) – The main schedule, showing data for the whole range of services provided by provider, including admitted patient care on a finished consultant episode (FCE) basis.
- [The National Cost Collection index \(NCCI\)](#) – The National Cost Collection index (NCCI) is a measure of the relative cost difference between NHS providers.
- [The National Cost Collection index by department and service code](#) – A more granular view of the National Cost Collection index by department and service code.
- [Reconciliation Statement](#) – Shows the adjustments made to get from provider audited financial accounts to their total NCC costs.

2021/22 National Cost Collection data files

- [Organisation level source data part 1](#) – Unadjusted data as submitted by NHS providers in England.
- [Organisation level source data part 2](#) – Market forces factor (MFF) adjusted data.
- [Organisation level source data part 3](#) – Remaining organisational source level data and reference tables.

Please note that amended documentation was issued on 19 May 2023, reflecting updated data quality issues. The [NCC update log](#) gives details of changes made since initial publication.

Search



About us

Our work

Search

2021/22 National Cost Collection data

As in 2020/21, we have not produced a report for this year's National Cost Collection. However, the data itself is published in its entirety, presented in the following ways:

- the national schedules of NHS costs
- the National Cost Collection Index (NCCI)
- the reconciliation statement
- a database of source data

Archived cost collections

Please see the pages [2020/21 National Cost Collection Data Publication](#), [2019/20 National Cost Collection Data Publication](#) and [2018/19 National Cost Collection Data Publication](#) for the data from the past three years.

You can also access reference costs for the financial years 2017/18 and 2016/17 on the page [Archived Reference Costs](#), along with a link to earlier cost collections on the Department of Health website.

- [Organisation level source data part 2](#) – Market forces factor (MFF) adjusted data.
- [Organisation level source data part 3](#) – Remaining organisational source level data and reference tables.

Please note that amended documentation was issued on 19 May 2023, reflecting updated data quality issues. The [NCC update log](#) gives details of changes made since initial publication.

of work streams including
gramme and NHS
NHS services.

We use cookies to help us improve this website. [Find out more about cookies](#)



Search



[About us](#) [Improvement Hub](#) [Resources](#) [Events](#) [News and alerts](#) [Contact us](#)

[Home](#) > [Resources](#) > [Archived Reference Costs](#)

2017/18 Reference Costs



[2017/18 reference costs and guidance](#)

ZIP, 89.0 MB

2016/17 Reference Costs



[2016/17 reference costs and guidance](#)

ZIP, 85.1 MB

Previous years' reference costs



[Reference costs published by the Department of Health & Social Care](#)

from the [Department of Health and Social Care](#) website

The Department of Health & Social Care has collected reference costs from NHS providers for every financial year between 1997/98 and 2015/16.

Archived Reference Costs

Reference costs are the average unit cost to the NHS of providing secondary healthcare to NHS patients.



Add to favourites

(0)

This page contains archived Reference Costs for the financial years 2017/18 and 2016/17, along with a link to earlier cost collections on the Department of Health website. For information on the current (2018/19) collection please go to the page [National Cost Collection for the NHS](#).



Search



Departments Worldwide How government works Get involved
Consultations Statistics News and communications

Reference costs from 1998 to 2009 are available on the [National Archives](#).

[NHS reference costs 2015 to 2016](#)

15 December 2016 Research and analysis

[NHS reference costs 2014 to 2015](#)

27 November 2015 Research and analysis

[NHS reference costs 2013 to 2014](#)

9 March 2015 Policy paper

[NHS reference costs 2012 to 2013](#)

22 January 2015 Policy paper

[NHS reference costs: financial year 2011 to 2012](#)

8 November 2012 Policy paper

[2010-11 reference costs publication](#)

17 November 2011 Policy paper

[NHS reference costs 2009-2010](#)

13 January 2011 Policy paper

[Home](#) > [Health and social care](#) > [National Health Service](#) > [NHS efficiency](#)

Collection

NHS reference costs

Reference costs are the average unit cost to the NHS of providing secondary healthcare to NHS patients.

This was published under the 2010 to 2015 Conservative and Liberal Democrat coalition government

Published 23 January 2014

Last updated 15 December 2016 — [see all updates](#)

From: [Department of Health and Social Care](#)



NHS costing

Health care ▾ Social care ▾ Public health ▾

Management resources ▾

Publications ▾

Consultations ▾

Media Centre ▾

About us ▾

► Management resources

► Finance and planning

► Payment by results

► NHS costing

► NHS Injury Cost Recovery scheme

► Emergency planning

► Workforce

► NHS procurement

► Information policy

► Social enterprise

► Shared Business Services

► Responsible officers

► Death certification

► Health Gateway Reviews

You are here: [Home](#) >> [Management resources](#) >> [NHS costing](#)

 [Email this page](#)

Provides link to previous year's reference costs data publications

Last modified date: 8 August 2011

Provides link to previous year's reference costs data publications.

Reference costs under HRG4.0

► [NHS reference costs 2008-2009](#)

► [NHS reference costs 2007-08](#)

► [NHS reference costs 2006-07](#)

Archived data publications (on the National Archives website). Referenced costs under HRG3.5

► [NHS reference costs 2005-06](#) 

► [NHS reference costs 2004-2005](#) 

► [NHS reference costs 2003-2004](#) 

```

1 fnGetZipCSV <- function(url, csv, subzip = "") {
2   print("Downloading...")
3   zip <- tempfile()
4   GET(url = url,
5        config = write_disk(zip),
6        progress())
7   if (subzip != "") {
8     print("Unzipping subzip...")
9     zip <- unzip(zipfile = zip,
10                  files = subzip,
11                  exdir = dirname(tempdir()),
12                  junkpaths = T)
13   }
14   print("Unzipping...")
15   csv <- unzip(zipfile = zip,
16                 files = csv,
17                 exdir = dirname(tempdir()),
18                 junkpaths = T)
19   unlink(zip)
20   fread(file = csv)
21 }
```

```

1 fnDownloadParseCSV <- function(url, subzip, csv, Yr) {
2   print(Yr)
3   fnGetZipCSV(url = url,
4                subzip = subzip,
5                csv = csv) %!>%
6   fnParseRefCostsCSV(Yr = Yr)
7 }
```

NHS Reference Costs

Making it functional

```

1 fnParseRefCostsCSV <- function(csv, Yr, blnXS = TRUE) {
2   print("Parsing...")
3   csv %>%
4     clean_names(case = "upper_camel") %>%
5     rename(any_of(c(CurrencyCode = "CurrencyCode",
6                   Mean = "NationalMean",
7                   DeptCode = "DepartmentCode",
8                   Activity = "TotalActivity",
9                   OrgCode = "ProviderCode"))) %>%
10    mutate(across(.cols = any_of(c("Activity", "Mean", "ExpectedCost")),
11               .fns = ~ as.numeric(na_if(as.character(.x), "*"))),
12           across(.cols = any_of(c("ServiceCode")),
13                  .fns = ~ as.character(.x)),
14           DeptCode = str_replace_all(DeptCode,
15                                     c("TEI$|EI$" = "EL",
16                                       "TEIXS$|EI_XS$" = "EL_XS",
17                                       "TNEI_L$|NEI_L$" = "NEL",
18                                       "TNEI_L_XS$|NEI_L_XS$" = "NEL_XS",
19                                       "TNEI_S$|NEI_S$" = "NES",
20                                       "TDC$" = "DC",
21                                       "TOPROC" = "OPROC",
22                                       "\\\GTCL" = "CL",
23                                       "\\\GTNCL" = "NCL")))) %>%
24     select(-any_of(c("Memo1", "ScaledMff", "OrganisationName", "DepartmentName", "ServiceName", "CurrencyName"))) %>%
25     add_column(Yr = Yr) %>%
26     as_tibble()
27   if (Yr == 2008) {
28     csv %>%
29       rename(any_of(c(Activity = "Fce",
30                     Activity = "ActivityP1"))) %>%
31       select(-any_of(c("Flag", "ActivityP2", "ActivityP3", "ActivityP4")))
32   if ("BedDays" %in% names(csv)) {
33     csv %>%
34       pivot_longer(cols = contains(c("UnitCost", "Activity")),
35                     names_to = c("tmp1", "tmp2"),
36                     names_pattern = "(.)(UnitCost|Activity)") %>%
37       drop_na(value) %>%
38       mutate(DeptCode = paste0(DeptCode, ifelse(tmp1 == "ExcessBedDays", "_XS", "")),
39             BedDays = ifelse(tmp1 == "ExcessBedDays", NA, BedDays)) %>%
40       pivot_wider(names_from = tmp2,
41                  values_from = value) %>%
42       select(-tmp1)
43   }
44   csv %>%
45     mutate(ActualCost = UnitCost * Activity)
46   }
47   csv %>%
48     bind_rows(tibble(OrgType = character(),
49                SupplierType = character(),
50                UnitCost = numeric(),
51                BedDays = numeric()))
52 }
```

NHS Reference Costs

Making it functional

```
3 csv %>%
4   clean_names(case = "upper_camel") %>%
5   rename(any_of(c(CurrencyCode = "CurrencyCode",
6                 Mean      = "NationalMean",
7                 DeptCode   = "DepartmentCode",
8                 Activity   = "TotalActivity",
9                 OrgCode    = "ProviderCode")))) %>%
10  mutate(across(.cols = any_of(c("Activity", "Mean", "ExpectedCost")),
11             .fns  = ~ as.numeric(na_if(as.character(.x), "*"))),
12        across(.cols = any_of(c("ServiceCode")),
13             .fns  = ~ as.character(.x)),
14        DeptCode = str_replace_all(DeptCode,
15                                  c("TEI$|EI$"           = "EL",
16                                    "TEIXS$|EI_XS$"     = "EL_XS",
17                                    "TNEI_L$|NEI_L$"    = "NEL",
18                                    "TNEI_L_XS$|NEI_L_XS$" = "NEL_XS",
19                                    "TNEI_S$|NEI_S$"    = "NES",
20                                    "TDC$"              = "DC",
21                                    "TOPROC"            = "OPROC",
22                                    "\\GTCL"             = "CL",
23                                    "\\GTNCL"            = "NCL")))) %>%
24  select(-any_of(c("Memo1", "ScaledMff", "OrganisationName", "DepartmentName", "ServiceName", "CurrencyName")))
```

NHS Reference Costs

Making it functional

```
> tbLY

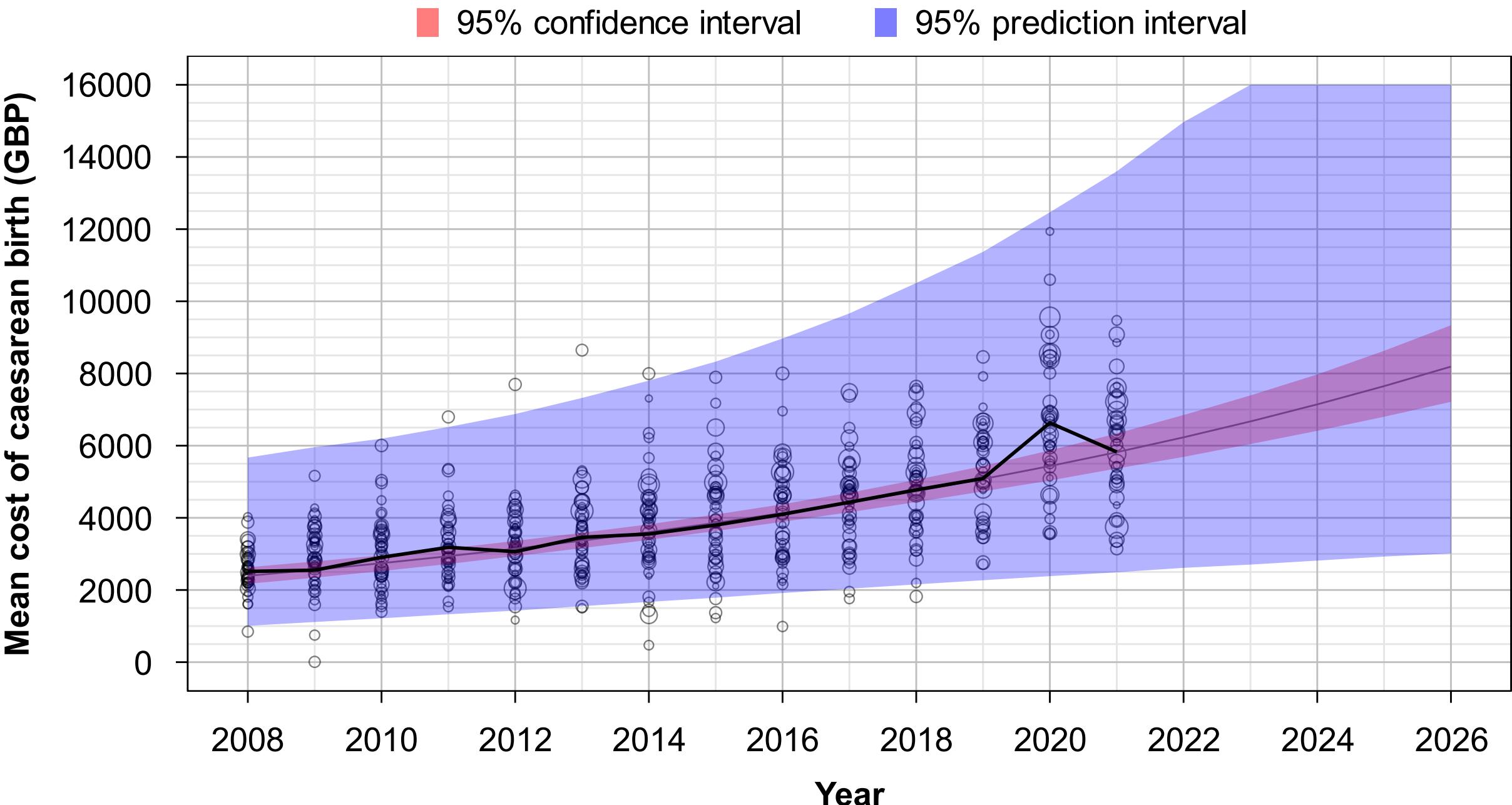
# A tibble: 15 × 4
  Yr      url          csv      subzip
  <chr> <chr>        <chr>    <chr>
1 2008  https://webarchive.nationalarchives.g... tbl_Admitted_Patient_Care.csv  ""
2 2008  https://webarchive.nationalarchives.g... tbl_Non-Admitted_Patient_Care.csv ""
3 2009  https://webarchive.nationalarchives.g... 1 Data.csv                   ""
4 2010  https://webarchive.nationalarchives.g... 1 Data.csv                   ""
5 2011  https://webarchive.nationalarchives.g... 1 Data.csv                   ""
6 2012  https://webarchive.nationalarchives.g... 1 Data.csv                   ""
7 2013  https://webarchive.nationalarchives.g... 1 Data.csv                   ""
8 2014  https://webarchive.nationalarchives.g... 1 Data.csv                   ""
9 2015  https://webarchive.nationalarchives.g... 1a Data.csv                  ""
10 2016 https://webarchive.nationalarchives.g... csv061117NoMff (1).csv     "8 - Organisation level source data pa..."
11 2017 https://webarchive.nationalarchives.g... 1 - Data.csv                 "7 - Organisation level source data pa..."
12 2018 https://www.england.nhs.uk/wp-content... 1 Data V2.csv                ""
13 2019 https://www.england.nhs.uk/wp-content... NCC_Schedule_1920_Org_level_Data_1 v2... ""
14 2020 https://www.england.nhs.uk/wp-content... NCC_Schedule_2021_Data_Org_level_Data... ""
15 2021 https://www.england.nhs.uk/wp-content... NCC_Schedule_2021_Data_Org_level_Data... ""
```

```
1 tbLD ← tbLY %>%
2   filter(between(Yr, 2008, 2021)) %>%
3   pmap(.f      = fnDownloadParseCSV,
4       .progress = T) %>%
5   list_rbind()
```

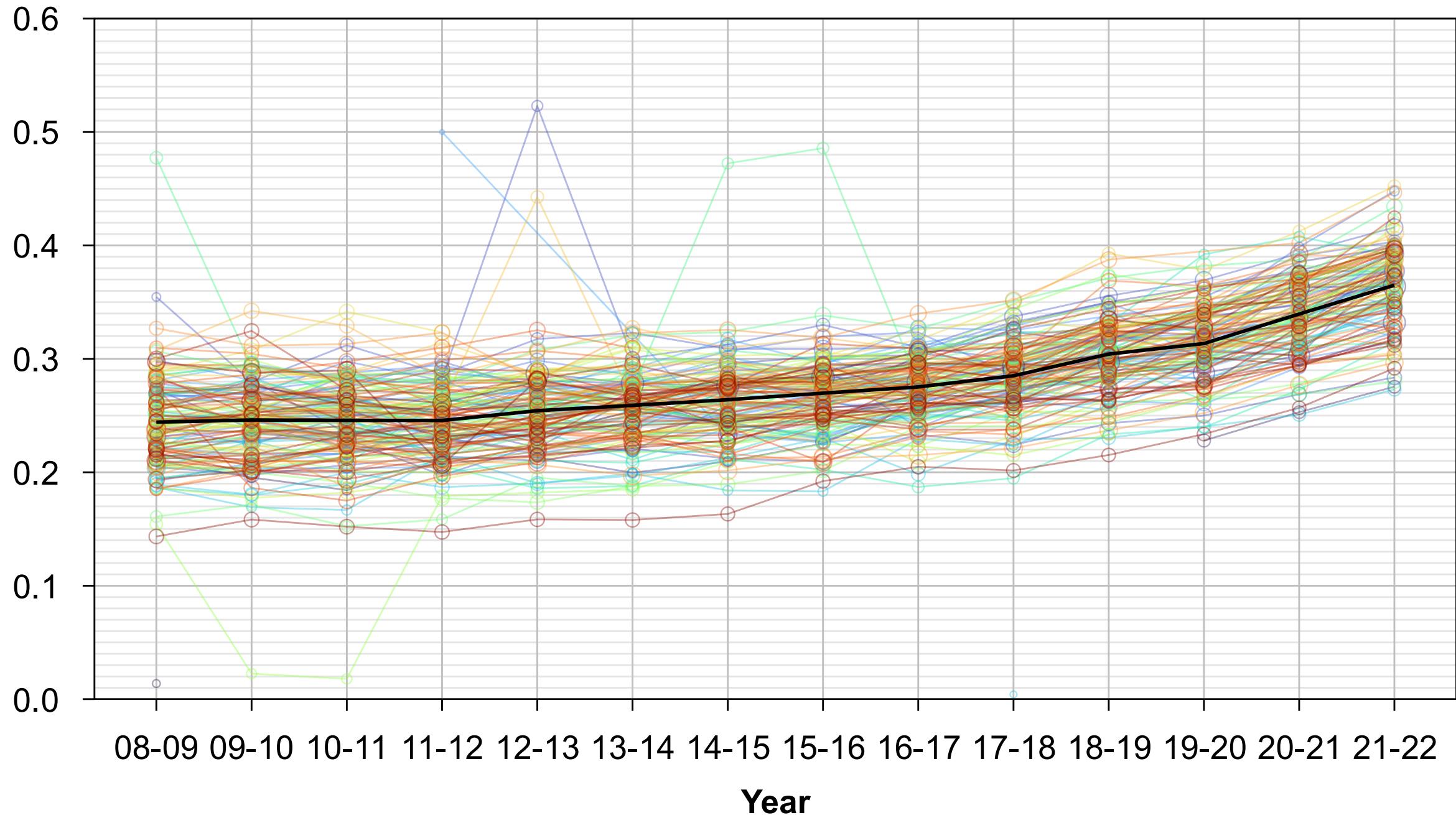
NHS Reference Costs

Making it functional

```
# A tibble: 22,003,039 × 14
  OrgCode SupplierType DeptCode ServiceCode CurrencyCode BedDays    Yr UnitCost Activity ActualCost OrgType  Mean ExpectedCost MappingPot
  <chr>   <chr>      <chr>   <chr>      <chr>       <dbl> <dbl>     <dbl>     <dbl>     <dbl> <chr>   <dbl>   <dbl>   <dbl>
1 RNJ      OWN        NEL      410       HB23C        4  2008     698      1       698 <NA>     NA      NA <NA>
2 RNJ      OWN        NEL_XS   410       HB23C        NA  2008     254      6      1524 <NA>     NA      NA <NA>
3 RNJ      OWN        NEL      410       HB24C        10 2008    2794.     2      5588. <NA>     NA      NA <NA>
4 RNJ      OWN        NEL_XS   410       HB24C        NA  2008     254     16      4064 <NA>     NA      NA <NA>
5 RNJ      OWN        NEL      410       HD21B        6  2008    2032.     2      4064. <NA>     NA      NA <NA>
6 RNJ      OWN        NEL_XS   410       HD21B        NA  2008     254     10      2540 <NA>     NA      NA <NA>
7 RNJ      OWN        NEL      410       HD21C        3  2008     703      1       703 <NA>     NA      NA <NA>
8 RNJ      OWN        NEL_XS   410       HD21C        NA  2008     254      1       254 <NA>     NA      NA <NA>
9 RNJ      OWN        NEL      410       QZ17B        21 2008    1333.     4      5334. <NA>     NA      NA <NA>
10 RNJ     OWN        NEL     410       SA09F       18 2008    2794.     2      5588. <NA>     NA     NA <NA>
# i 22,003,029 more rows
# i Use `print(n = ...)` to see more rows
```



Proportion of births by Caesarean



Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. `nomisr` → ONS cause of death statistics
- API serving JSON read via `jsonlite`
 - e.g. NHS Open Data Portal → English Prescribing Dataset
- Archived .XLSXs read via `readxl`
 - e.g. ONS lifetables
- Archived .CSVs in .ZIPs within .ZIPs
 - e.g. National Cost Collection (FKA NHS Reference Costs)

Sources of web-served data

In approximate, subjective, descending order of painfulness...

- API for which a package already exists
 - e.g. `nomisr` → ONS cause of death statistics
- API serving JSON read via `jsonlite`
 - e.g. NHS Open Data Portal → English Prescribing Dataset
- Archived .XLSXs read via `readxl`
 - e.g. ONS lifetables
- Archived .CSVs in .ZIPs within .ZIPs
 - e.g. National Cost Collection (FKA NHS Reference Costs)
- Anything (even a lovely API) serving XML
 - e.g. DM+D

WHAT COULD ★POSSIBLY★ GO WRONG?

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file

NHS Reference Costs

Making it functional

```
> tbLY

# A tibble: 15 × 4
  Yr      url          csv      subzip
  <chr> <chr>        <chr>    <chr>
1 2008  https://webarchive.nationalarchives.g... tbl_Admitted_Patient_Care.csv  ""
2 2008  https://webarchive.nationalarchives.g... tbl_Non-Admitted_Patient_Care.csv ""
3 2009  https://webarchive.nationalarchives.g... 1 Data.csv                ""
4 2010  https://webarchive.nationalarchives.g... 1 Data.csv                ""
5 2011  https://webarchive.nationalarchives.g... 1 Data.csv                ""
6 2012  https://webarchive.nationalarchives.g... 1 Data.csv                ""
7 2013  https://webarchive.nationalarchives.g... 1 Data.csv                ""
8 2014  https://webarchive.nationalarchives.g... 1 Data.csv                ""
9 2015  https://webarchive.nationalarchives.g... 1a Data.csv               ""
10 2016 https://webarchive.nationalarchives.g... csv061117NoMff (1).csv "8 - Organisation level source data pa..."
11 2017 https://webarchive.nationalarchives.g... 1 - Data.csv              "7 - Organisation level source data pa..."
12 2018 https://www.england.nhs.uk/wp-content... 1 Data V2.csv             ""
13 2019 https://www.england.nhs.uk/wp-content... NCC_Schedule_1920_Org_level_Data_1_v2... ""
14 2020 https://www.england.nhs.uk/wp-content... NCC_Schedule_2021_Data_Org_level_Data... ""
15 2021 https://www.england.nhs.uk/wp-content... NCC_Schedule_2021_Data_Org_level_Data... ""
```

```
1 tbLD ← tbLY %>%
2   filter(between(Yr, 2008, 2021)) %>%
3   pmap(.f      = fnDownloadParseCSV,
4     .progress = T) %>%
5   list_rbind()
```

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file
- ↳ You don't get the data

INTERNET ARCHIVE

[DONATE](#)

Explore more than 790 billion web pages saved over time

Enter a URL or words related to a site's home page

Saved 1 time July 4, 2022.



JAN FEB MAR APR

1	1	2	3	4	5	1	2	3	4	5	1	2
2	3	4	5	6	7	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18
16	17	18	19	20	21	22	20	21	22	23	24	25
23	24	25	26	27	28	29	27	28	29	30	31	
30	31											

MAY JUN JUL AUG

1	2	3	4	5	6	7	1	2	3	4	5	6
8	9	10	11	12	13	14	5	6	7	8	9	10
15	16	17	18	19	20	21	12	13	14	15	16	17
22	23	24	25	26	27	28	19	20	21	22	23	24
29	30	31					26	27	28	29	30	
							24	25	26	27	28	29
							21	22	23	24	25	26
							17	18	19	20	21	22
							10	11	12	13	14	15
							3	4	5	6	7	8
							13	14	15	16	17	18
							10	11	12	13	14	15
							17	18	19	20	21	22
							24	25	26	27	28	29
							21	22	23	24	25	26
							17	18	19	20	21	22
							14	15	16	17	18	19
							10	11	12	13	14	15
							7	8	9	10	11	12
							3	4	5	6	7	8
							13	14	15	16	17	18
							10	11	12	13	14	15
							7	8	9	10	11	12
							4	5	6	7	8	9
							1	2	3	4	5	6
							11	12	13	14	15	16
							18	19	20	21	22	23
							15	16	17	18	19	20
							12	13	14	15	16	17
							9	10	11	12	13	14
							6	7	8	9	10	11
							3	4	5	6	7	8
							1	2	3	4	5	6
							13	14	15	16	17	18
							10	11	12	13	14	15
							7	8	9	10	11	12
							4	5	6	7	8	9
							1	2	3	4	5	6
							11	12	13	14	15	16
							18	19	20	21	22	23
							15	16	17	18	19	20
							12	13	14	15	16	17
							9	10	11	12	13	14
							6	7	8	9	10	11
							3	4	5	6	7	8
							1	2	3	4	5	6

SEP OCT NOV DEC

1	2	3	1	2	3	4	5	1	2	3	4	5	1	2	3
4	5	6	7	8	9	10	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30		23	24	25	26	27	28	29	30	
							27	28	29	30					25
							24	25	26	27	28	29	30	31	
							21	22	23	24	25	26	27	28	
							18	19	20	21	22	23	24	25	
							15	16	17	18	19	20	21	22	
							12	13	14	15	16	17	18	19	
							9	10	11	12	13	14	15	16	
							6	7	8	9	10	11	12	13	
							3	4	5	6	7	8	9	10	
							1	2	3	4	5	6	7	8	

Note

This calendar view maps the number of times https://www.england.nhs.uk/wp-content/uploads/2020/08/Organisation_level_source_data_part_1.zip was crawled by the Wayback Machine, *not* how many times the site was actually updated. More info in the [FAQ](#).

INTERNET ARCHIVE

[DONATE](#)

Explore more than 790 billion web pages saved over time

Enter a URL or words related to a site's home page

<https://www.england.nhs.uk/wp-content/uploads/2022/0>[Latest](#)[Show All](#)

Hrm.

The Wayback Machine has not archived that URL.

This page is available on the web!

Help make the Wayback Machine more complete!

[Save this url in the Wayback Machine](#)**Note**

This calendar view maps the number of times the site was actually crawled by the Wayback Machine, not

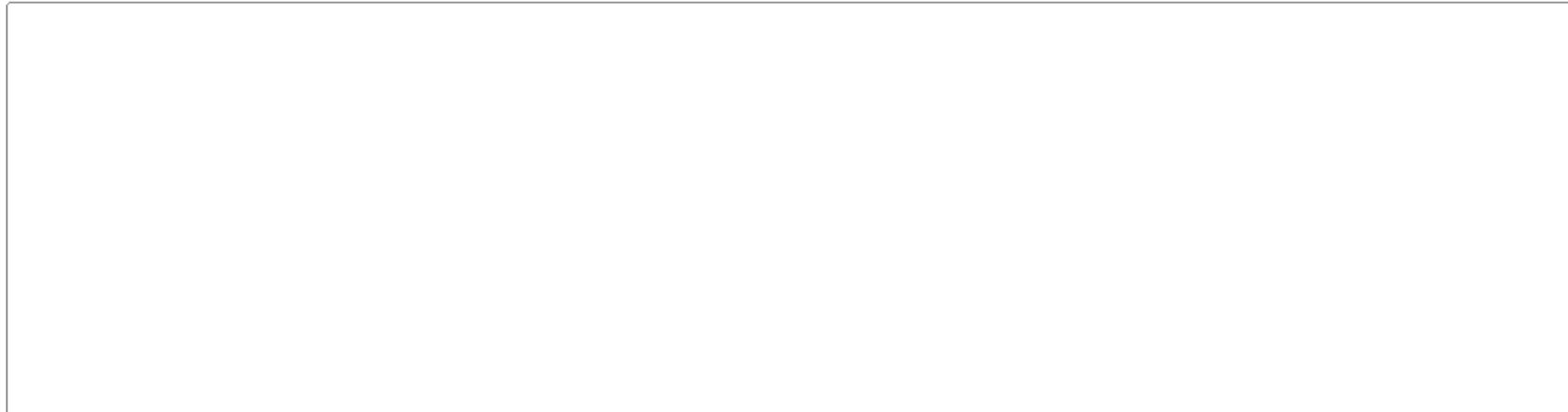
crawled by the Wayback Machine, not

[DONATE](#)[DONATE](#)

INTERNET ARCHIVE

wayBack Machine

Saving page https://www.england.nhs.uk/wp-content/uploads/2022/07/NCC_Schedule_2021_Data_Org_level_Data_1-1.zip

Saving... [If something goes wrong please click here to send us an error report.](#)

[DONATE](#)

INTERNET ARCHIVE



Saving page https://www.england.nhs.uk/wp-content/uploads/2022/07/NCC_Schedule_2021_Data_Org_level_Data_1-1.zip  Done!  First Archive

A snapshot was captured. Visit page: /web/20230303120148/https://www.england.nhs.uk/wp-content/uploads/2022/07/NCC_Schedule_2021_Data_Org_level_Data_1-1.zip

If something goes wrong please click [here](#) to send us an error report.

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file
 - ↳ You don't get the data
- The data-owner might reformat the data
 - Change API syntax

Secondary Care Medicines Data (SCMD) with indicative price

Followers

18

 Theme



[Dataset](#)

[Activity Stream](#)

[Discussions](#)

 [Report an issue](#)

Secondary Care Medicines Data (SCMD) with indicative price

Important Update

We have made changes to the Secondary Care Medicines data (SCMD) we publish.

We previously published Secondary Care Medicines Data in two locations on the Open Data Portal. We have consolidated this so that we will only provide version of the SCMD. The version that we have decided to continue to publish is, this version of SCMD, with an indicative price metric. The dataset with indicative price was in the Experiment Area theme and will be moved into the Hospital & Provider Medicines theme. The SCMD without indicative price has now been retired and no further data will be added to that page.

NHS Open Data Portal API

Making it functional

```
1 fnNHSOpenData <- function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {  
2   strURL <- "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="  
3   res <- paste0(dataset, "_", Year, sprintf("%02d", Month))  
4   strSQL <- paste("SELECT", strSelect, "FROM `", res, "`")  
5   if (strWhere != "") strSQL <- paste(strSQL, "WHERE", strWhere)  
6   if (strGroupBy != "") strSQL <- paste(strSQL, "GROUP BY", strGroupBy)  
7  
8   url <- glue(strURL, strSQL) %>  
9     URLencode()  
10  
11  tibble(Dataset = res) %>  
12    bind_cols(fromJSON(url)) %>  
13      as_tibble()  
14 }
```

NHS Open Data Portal API

Making it functional

```
1 fnNHSOpenData <- function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {  
2   strURL <- "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="  
3   if (dataset=="SCMD") dataset <- paste0("SCMD_", c("FINAL", "WIP", "PROVISIONAL"))  
4   res <- paste0(dataset, "_", Year, sprintf("%02d", Month))  
5   strSQL <- paste("SELECT", strSelect, "FROM `", res, "`")  
6   if (strWhere != "") strSQL <- paste(strSQL, "WHERE", strWhere)  
7   if (strGroupBy != "") strSQL <- paste(strSQL, "GROUP BY", strGroupBy)  
8  
9   url <- glue(strURL, strSQL) ▷  
10  URLencode()  
11  for (i in 1:length(url)) {  
12    suppressWarnings(xx <- tryCatch(fromJSON(url[i]), error=function(e) e))  
13    if (!inherits(xx, "error")) {  
14      return(tibble(Dataset = res[i]) ▷  
15             bind_cols(xx$result$result$records ▷  
16                         as_tibble()))  
17    }  
18  }  
19 }
```

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file
 - ↳ You don't get the data
- The data-owner might reformat the data
 - Change API syntax
 - Change how data are presented (Excel workbooks worst culprits)

ONS life tables

```
1 strURL ← paste0("https://www.ons.gov.uk/file?",
2                     "uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
3                     "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx")
4
5 tmp ← tempfile(fileext = ".xlsx")
6 httr::GET(strURL, httr::write_disk(tmp))
7
8 tbLLT ← readxl::read_excel(path  = tmp,
9                             sheet = "2018-2020",
10                            range = "A6:M107")
```

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file
 - ↳ You don't get the data
- The data-owner might reformat the data
 - Change API syntax
 - Change how data are presented (Excel workbooks worst culprits)
 - ↳ You might not get the data; you might get the data but your code will fail

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file

↳ You don't get the data
- The data-owner might reformat the data
 - Change API syntax
 - Change how data are presented (Excel workbooks worst culprits)

↳ You might not get the data; you might get the data but your code will fail
- The data-owner might change the data



About us

Our work

Costing in the NHS

National Cost Collection for the NHS

Costing glossary

Approved Costing Guidance

2021/22 National Cost Collection data

As in 2020/21, we have not produced a report for this year's National Cost Collection. However, the data itself is published in its entirety, presented in the following ways:

- the national schedules of NHS costs
- the National Cost Collection Index (NCCI)
- the reconciliation statement
- a database of source data

Data spreadsheets

- [National schedule of NHS costs](#) – The main schedule, showing data for the whole range of services provided by provider, including admitted patient care on a finished consultant episode (FCE) basis.
- [The National Cost Collection index \(NCCI\)](#) – The National Cost Collection index (NCCI) is a measure of the relative cost difference between NHS providers.
- [The National Cost Collection index by department and service code](#) – A more granular view of the National Cost Collection index by department and service code.
- [Reconciliation Statement](#) – Shows the adjustments made to get from provider audited financial accounts to their total NCC costs.

2021/22 National Cost Collection data files

- [Organisation level source data part 1](#) – Unadjusted data as submitted by NHS providers in England.
- [Organisation level source data part 2](#) – Market forces factor (MFF) adjusted data.
- [Organisation level source data part 3](#) – Remaining organisational source level data and reference tables.

Please note that amended documentation was issued on 19 May 2023, reflecting updated data quality issues. The [NCC update log](#) gives details of changes made since initial publication.

What could possibly go wrong?

- The data-owner might move the data
 - Change API URL / availability
 - Move/rename file
- ↳ You don't get the data
- The data-owner might reformat the data
 - Change API syntax
 - Change how data are presented (Excel workbooks worst culprits)
- ↳ You might not get the data; you might get the data but your code will fail
- The data-owner might change the data
 - ↳ Everything works in your code, but your answers change

The data-owner might change the data

↳ Everything works in your code, but your answers change

- It's a bug!
 - Suddenly, my beautifully replicable analysis doesn't replicate

The data-owner might change the data

↳ Everything works in your code, but your answers change

- It's a bug!
 - Suddenly, my beautifully replicable analysis doesn't replicate
- It's a feature!
 - If the data-owners changed the data, it was probably some sort of correction; don't you want the correct data?
 - Maybe you want latest, e.g., drug costs every time you run your analysis
 - Haven't you heard? Living HTA is the future!

The data-owner might change the data

↳ Everything works in your code, but your answers change

- It's a bug!
 - Suddenly, my beautifully replicable analysis doesn't replicate
- It's a feature!
 - If the data-owners changed the data, it was probably some sort of correction; don't you want the correct data?
 - Maybe you want latest, e.g., drug costs every time you run your analysis
 - Haven't you heard? Living HTA is the future!
- It depends!

All code available at: <https://github.com/gbirlgrs/R-HTA-2023>



<https://research.manchester.ac.uk/en/persons/gabriel.rogers>



gabriel.rogers@manchester.ac.uk



[@gbirlgrs](https://twitter.com/gbirlgrs)



github.com/gbirlgrs