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

How-to and whether-to

MANCHESTER 1824

The University of Manchester

Gabriel Rogers

Senior research fellow

Manchester Centre for Health Economics

R for HTA workshop

York, 2023

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

Driver #1

ACCESSIBILITY OF USEFUL DATA

Cabinet Office

Government Digital Service

Policy paper

Government Transformation Strategy: better use of data

Published 9 February 2017

https://www.gov.uk/government/publications/government-transformation-strategy-2017-to-2020/government-transformation-strategy-better-use-of-data

Policy pape

Gover bette

Published 9 Feb

y:

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

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

NHS 75 Digital

https://digital.nhs.uk/services/supporting-open-data-and-transparency

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

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.

MANCHESTER 1824
The University of Manchester

Driver #2

REPLICABLE WORKFLOW

https://www.tidyverse.org/blog/2017/12/workflow-vs-script/

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 🔥.*

MANCHESTER 1824
The University of Manchester

EXAMPLES

MANCHESTER 1824
The University of Manchester

Sources of web-served data

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

MANCHESTER 1824
The University of Manchester

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

https://www.nomisweb.co.uk/

nomis

official census and labour market statistics

Office for National Statistics

Search...

Home | Reports | Data Sources | Census | Contact us | Sign-in | Settings

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.

Postcode or area name | Search

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.

https://docs.ropensci.org/nomisr/

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.

https://docs.ropensci.org/nomisr/

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,
  ...
)
```

MANCHESTER 1824
The University of Manchester

ONS NOMIS via `nomisr`

What proportion of deaths are due to CVD?

```
tblCoD ← nomisr::nomis_get_data(id          = "NM_161_1",
                                date        = 2013:2020,
                                geography   = 2092957703,
                                sex         = 1:2,
                                age         = 1:20,
                                measure     = 1,
                                measures    = 20100,
                                cause_of_death = c(0,9))
```

MANCHESTER 1824
The University of Manchester

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>      <dbl> <chr>             <chr>          <chr>                    <dbl>               <dbl>
 1  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 2  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 3  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 4  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 5  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 6  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 7  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 8  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
 9  2013      2013      2013 date                  0              0 2092957703 England and Wales K04000001      countries                  499                   0
10  2013      2013      2013 date                  0              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
```

MANCHESTER 1824
The University of Manchester

ONS NOMIS via `nomisr`

What proportion of deaths are due to CVD?

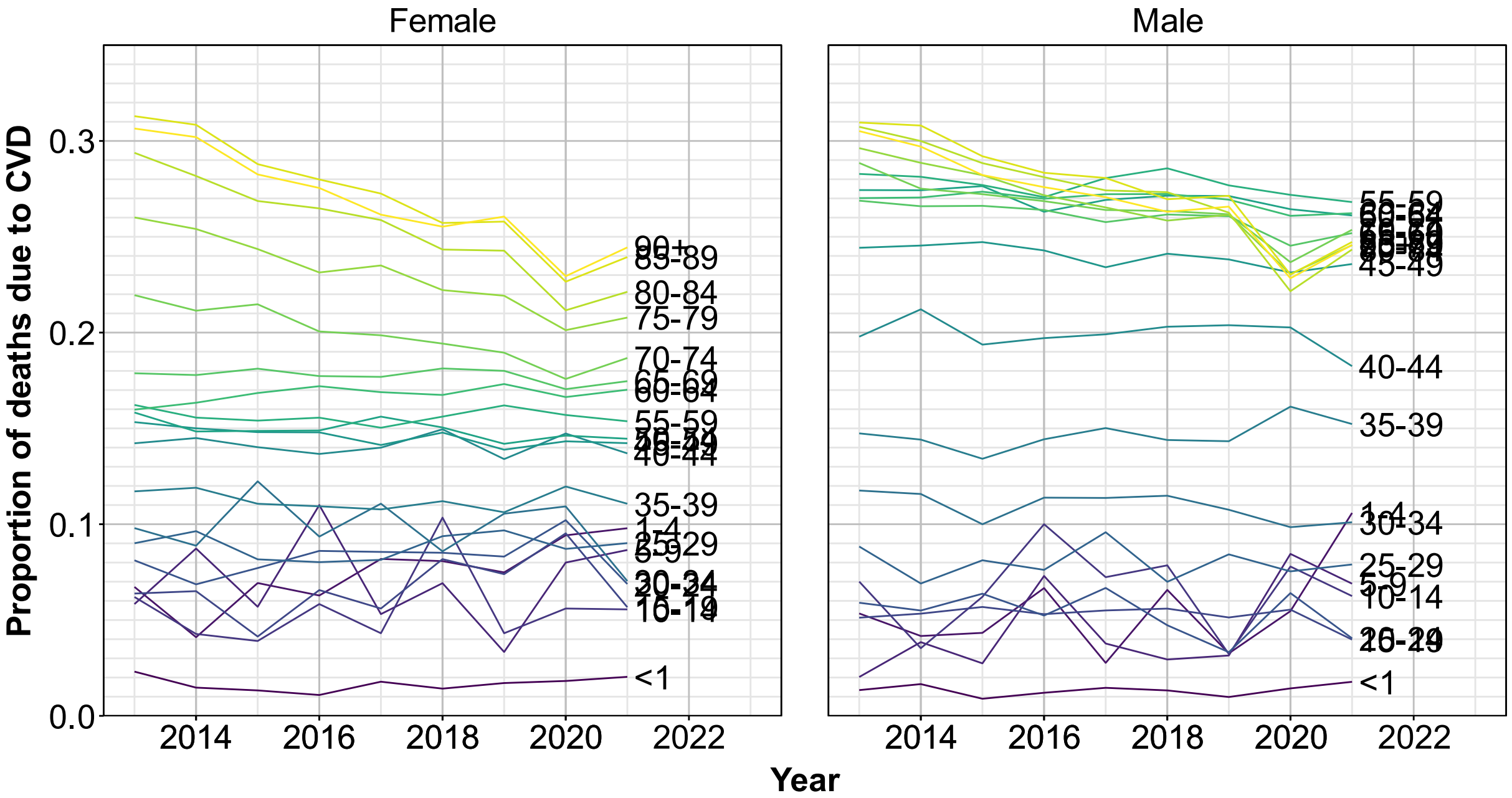
```
tblCoD ← nomisr::nomis_get_data(id          = "NM_161_1",
                                date        = 2013:2020,
                                geography   = 2092957703,
                                sex         = 1:2,
                                age         = 1:20,
                                measure     = 1,
                                measures    = 20100,
                                cause_of_death = c(0,9))

tblCoD ▷
  dplyr::mutate(CoD = ifelse(CAUSE_OF_DEATH==9, "CVD", "All")) ▷
  dplyr::select(Year   = DATE,
                Sex    = GENDER_NAME,
                AgeCat = AGE_NAME,
                AgeC   = AGE_CODE,
                N      = OBS_VALUE,
                CoD) ▷
  tidyr::pivot_wider(values_from = N,
                     names_from  = CoD) ▷
  dplyr::mutate(pCVD  = CVD / All,
                AgeLo = c(0,1,1:18*5)[AgeC],
                AgeHi = c(1,1:18*5,999)[AgeC])
```

MANCHESTER 1824
The University of Manchester

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
```


Female
Male
Proportion of deaths due to CVD
0.0
0.1
0.2
0.3
2014
2016
2018
2020
2022
Year
90+
85-89
80-84
75-79
70-74
65-69
60-64
55-59
50-54
45-49
40-44
35-39
1-4
25-29
5-9
30-34
10-14
15-19
<1

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

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

https://opendata.nhsbsa.net/

NHS
Open Data Portal
Business Services Authority

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

Search datasets

#NHS Prescription Se... #VDPS #NHS Prescriptions #NHS Pensions

NHS
Open Data Portal
Business Services Authority

https://opendata.nhsbsa.net/

Home | Data | News | Documentation | About | Log in

Home / Community Prescribing & Dispensing

Community Prescribing & Dispensing

Datasets on the prescribing and dispensing of medicines in the community read more

Followers 7

Datasets 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

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Accessing English Prescribing Data via `jsonlite`

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Accessing English Prescribing Data via `jsonlite`

```r
strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id=EPD_201401&sql="
strSQL ← paste("SELECT YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",
               "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE,",
               "sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC, sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ",
               "FROM `EPD_201401` WHERE BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",
               "GROUP BY YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR,",
               "BNF_CODE, BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE")

jsonRet ← paste0(strURL, strSQL) ▷
  URLencode() ▷
  jsonlite::fromJSON()

tblSQL ← jsonRet$result$result$records ▷
  dplyr::as_tibble() ▷
  dplyr::arrange(BNF_CODE)

tblSQL
```

```
"https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id=EPD_201401&sql=SELECT%20YEAR_MONTH,%20BNF_CHEMICAL_SUBSTANCE,%20CHEMICAL_SUBSTANCE_BNF_DESCR,%20BNF_CODE,%20BNF_DESCRIPTION,%20BNF_CHAPTER_PLUS_CODE,%20sum(TOTAL_QUANTITY)%20AS%20Quantity,%20sum(NIC)%20AS%20NIC,%20sum(ACTUAL_COST)%20AS%20Cost,%20avg(ADQUSAGE)%20AS%20ADQ%20FROM%20%60EPD_201401%60%20WHERE%20BNF_CHEMICAL_SUBSTANCE%20LIKE%20'060102%25'%20GROUP%20BY%20YEAR_MONTH,%20BNF_CHEMICAL_SUBSTANCE,%20CHEMICAL_SUBSTANCE_BNF_DESCR,%20BNF_CODE,%20BNF_DESCRIPTION,%20BNF_CHAPTER_PLUS_CODE"
```

MANCHESTER 1824
The University of Manchester

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 0601021H0AAAAAA Glibenclamide                Glibenclamide 2.5mg…    54889 34699. 32701.  16.1
# ℹ 118 more rows
# ℹ Use `print(n = ...)` to see more rows
```

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional

```r
fnNHSOpenData ← function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {
  strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="
  res ← paste0(dataset, "_", Year, sprintf("%02d", Month))
  strSQL ← paste("SELECT", strSelect, "FROM `{res}`")
  if (strWhere ≠ "") strSQL ← paste(strSQL, "WHERE", strWhere)
  if (strGroupBy ≠ "") strSQL ← paste(strSQL, "GROUP BY", strGroupBy)

  url ← glue(strURL, strSQL) ▷
    URLencode()

  tibble(Dataset = res) ▷
    bind_cols(fromJSON(url) ▷
                as_tibble())
}
```

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional

```
fnNHSOpenData ← function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {
  strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="
  res ← paste0(dataset, "_", Year, sprintf("%02d", Month))
  strSQL ← paste("SELECT", strSelect, "FROM `{res}`")
  if (strWhere ≠ "") strSQL ← paste(strSQL, "WHERE", strWhere)
  if (strGroupBy ≠ "") strSQL ← paste(strSQL, "GROUP BY", strGroupBy)

  url ← glue(strURL, strSQL) ▷
    URLencode()

  tibble(Dataset = res) ▷
    bind_cols(fromJSON(url) ▷
                as_tibble())
}

fnNHSOpenData(dataset    = "EPD",
              Year       = 2014,
              Month      = 1,
              strSelect  = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE, BNF_DESCRIPTION,",
                                 "BNF_CHAPTER_PLUS_CODE, sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC,",
                                 "sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ"),
              strWhere   = "BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",
              strGroupBy = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",
                                 "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE")) → tblDM
```

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional – calling via `purrr::map`

```r
fnNHSOpenData ← function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {
  strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="
  res ← paste0(dataset, "_", Year, sprintf("%02d", Month))
  strSQL ← paste("SELECT", strSelect, "FROM `{res}`")
  if (strWhere ≠ "") strSQL ← paste(strSQL, "WHERE", strWhere)
  if (strGroupBy ≠ "") strSQL ← paste(strSQL, "GROUP BY", strGroupBy)

  url ← glue(strURL, strSQL) ▷
    URLencode()

  tibble(Dataset = res) ▷
    bind_cols(fromJSON(url) ▷
                as_tibble())
}

crossing(Year       = 2014:2023,
         Month      = 1:12,
         dataset    = "EPD",
         strSelect  = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE, BNF_DESCRIPTION,",
                            "BNF_CHAPTER_PLUS_CODE, sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC,",
                            "sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ"),
         strWhere   = "BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",
         strGroupBy = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",
                            "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE") ▷
  filter(Year < 2023 | Month < 4) ▷
  pmap(.f        = fnNHSOpenData,
       .progress = TRUE) ▷
  list_rbind() → tblAllDM
```

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional

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

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional – calling via `purrr::map`

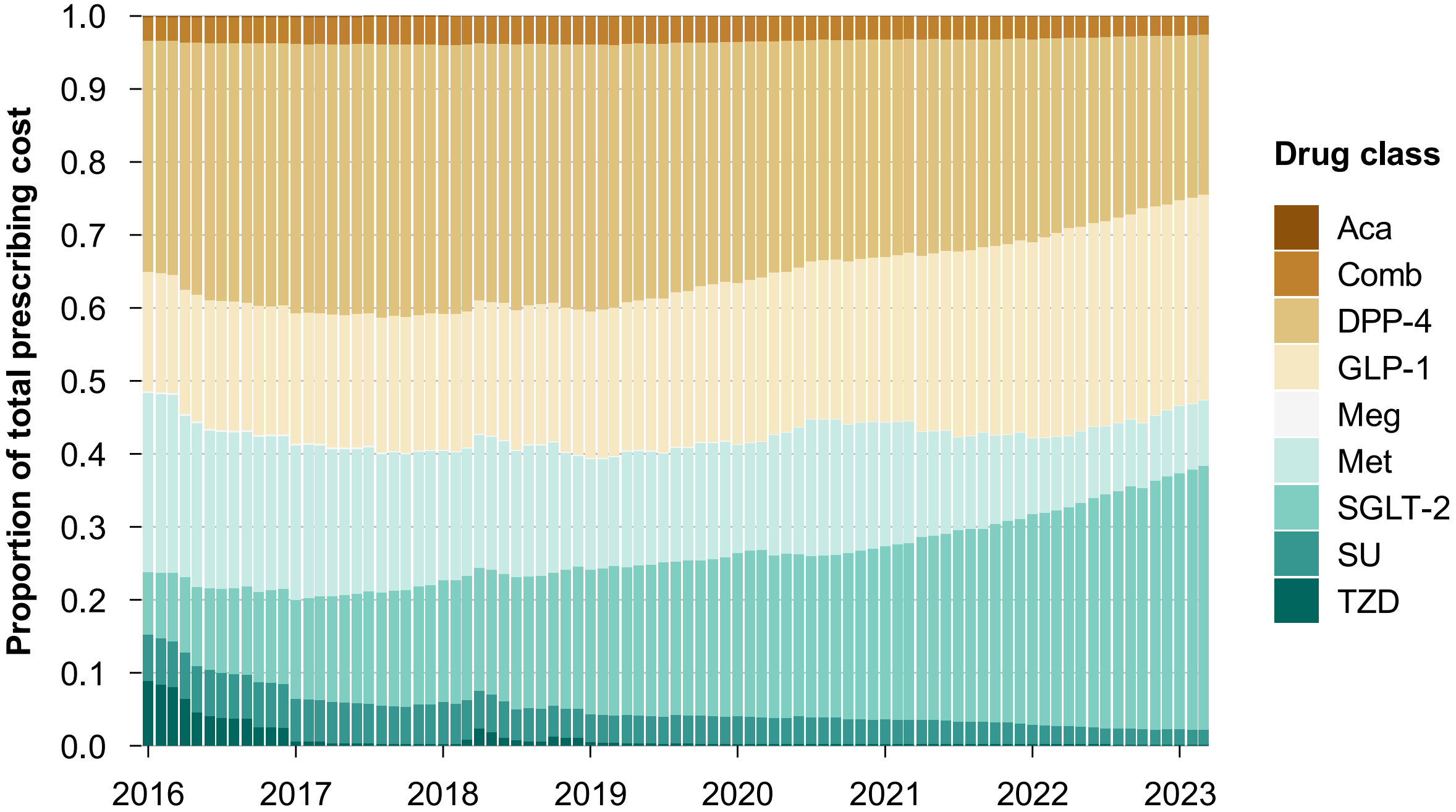
```r
fnNHSOpenData ← function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {
  strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="
  res ← paste0(dataset, "_", Year, sprintf("%02d", Month))
  strSQL ← paste("SELECT", strSelect, "FROM `{res}`")
  if (strWhere ≠ "") strSQL ← paste(strSQL, "WHERE", strWhere)
  if (strGroupBy ≠ "") strSQL ← paste(strSQL, "GROUP BY", strGroupBy)

  url ← glue(strURL, strSQL) ▷
    URLencode()

  tibble(Dataset = res) ▷
    bind_cols(fromJSON(url) ▷
                as_tibble())
}

crossing(Year       = 2014:2023,
         Month      = 1:12,
         dataset    = "EPD",
         strSelect  = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE, BNF_DESCRIPTION,",
                            "BNF_CHAPTER_PLUS_CODE, sum(TOTAL_QUANTITY) AS Quantity, sum(NIC) AS NIC,",
                            "sum(ACTUAL_COST) AS Cost, avg(ADQUSAGE) AS ADQ"),
         strWhere   = "BNF_CHEMICAL_SUBSTANCE LIKE '060102%'",
         strGroupBy = paste("YEAR_MONTH, BNF_CHEMICAL_SUBSTANCE, CHEMICAL_SUBSTANCE_BNF_DESCR, BNF_CODE,",
                            "BNF_DESCRIPTION, BNF_CHAPTER_PLUS_CODE") ▷
  filter(Year < 2023 | Month < 4) ▷
  pmap(.f        = fnNHSOpenData,
       .progress = TRUE) ▷
  list_rbind() → tblAllDM
```

MANCHESTER 1824
The University of Manchester

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>                  <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 0601023AIBBAAAA   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
```


Proportion of total prescribing cost
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2016
2017
2018
2019
2020
2021
2022
2023
Drug class
Aca
Comb
DPP-4
GLP-1
Meg
Met
SGLT-2
SU
TZD

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

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

https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/lifeexpectancies/datasets/nationallifetablesenglandreferencetables

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

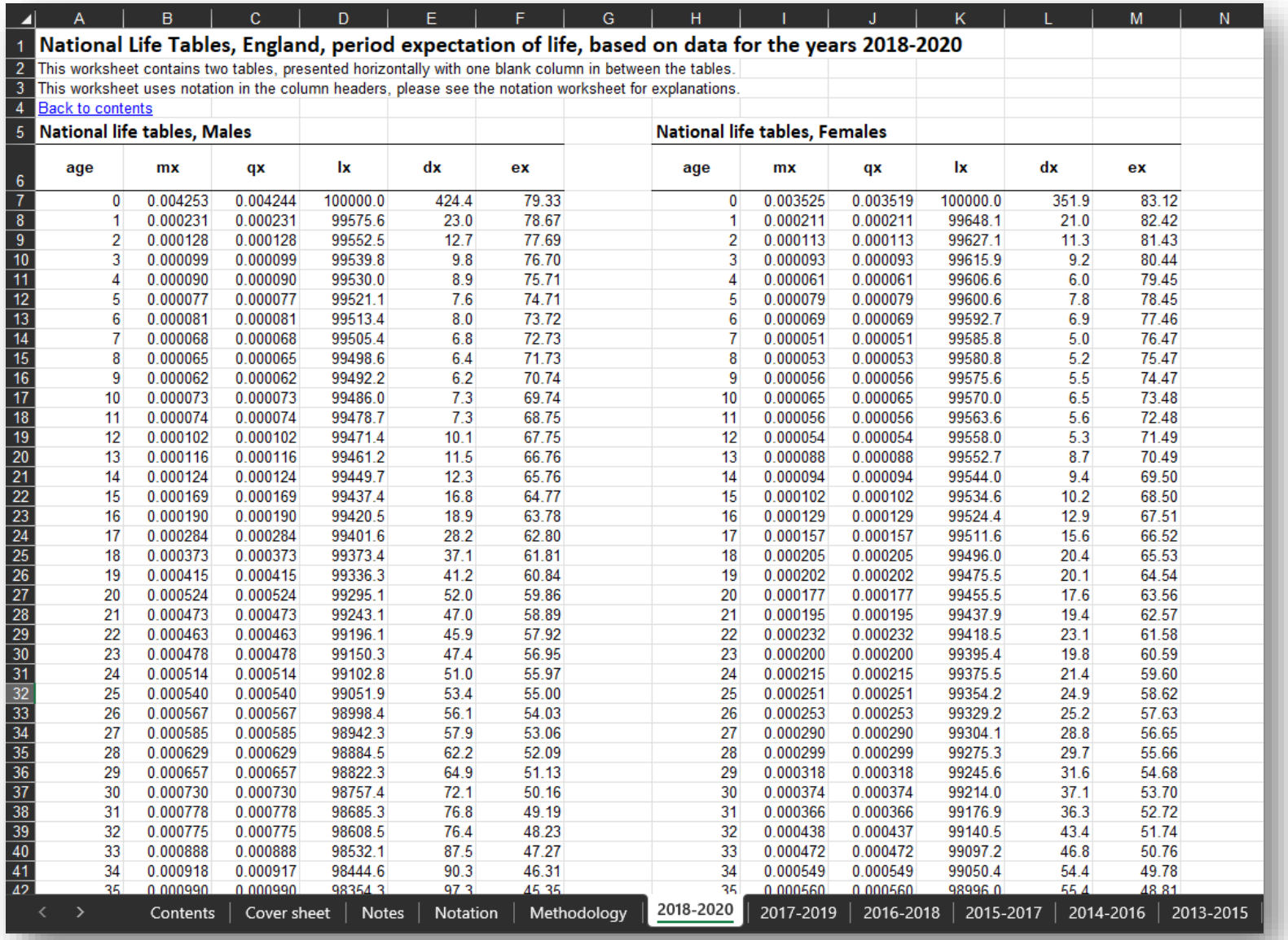
Julian Buxton
pop.info@ons.gov.uk
+44 1329 444661

Publications that use this data

National life tables – life expectancy in the UK

MANCHESTER 1824
The University of Manchester

ONS life tables



National Life Tables, England, period expectation of life, based on data for the years 2018-2020

This worksheet contains two tables, presented horizontally with one blank column in between the tables.
This worksheet uses notation in the column headers, please see the notation worksheet for explanations.
Back to contents

National life tables, Males

age	mx	qx	lx	dx	ex
0	0.004253	0.004244	100000.0	424.4	79.33
1	0.000231	0.000231	99575.6	23.0	78.67
2	0.000128	0.000128	99552.5	12.7	77.69
3	0.000099	0.000099	99539.8	9.8	76.70
4	0.000090	0.000090	99530.0	8.9	75.71
5	0.000077	0.000077	99521.1	7.6	74.71
6	0.000081	0.000081	99513.4	8.0	73.72
7	0.000068	0.000068	99505.4	6.8	72.73
8	0.000065	0.000065	99498.6	6.4	71.73
9	0.000062	0.000062	99492.2	6.2	70.74
10	0.000073	0.000073	99486.0	7.3	69.74
11	0.000074	0.000074	99478.7	7.3	68.75
12	0.000102	0.000102	99471.4	10.1	67.75
13	0.000116	0.000116	99461.2	11.5	66.76
14	0.000124	0.000124	99449.7	12.3	65.76
15	0.000169	0.000169	99437.4	16.8	64.77
16	0.000190	0.000190	99420.5	18.9	63.78
17	0.000284	0.000284	99401.6	28.2	62.80
18	0.000373	0.000373	99373.4	37.1	61.81
19	0.000415	0.000415	99336.3	41.2	60.84
20	0.000524	0.000524	99295.1	52.0	59.86
21	0.000473	0.000473	99243.1	47.0	58.89
22	0.000463	0.000463	99196.1	45.9	57.92
23	0.000478	0.000478	99150.3	47.4	56.95
24	0.000514	0.000514	99102.8	51.0	55.97
25	0.000540	0.000540	99051.9	53.4	55.00
26	0.000567	0.000567	98998.4	56.1	54.03
27	0.000585	0.000585	98942.3	57.9	53.06
28	0.000629	0.000629	98884.5	62.2	52.09
29	0.000657	0.000657	98822.3	64.9	51.13
30	0.000730	0.000730	98757.4	72.1	50.16
31	0.000778	0.000778	98685.3	76.8	49.19
32	0.000775	0.000775	98608.5	76.4	48.23
33	0.000888	0.000888	98532.1	87.5	47.27
34	0.000918	0.000917	98444.6	90.3	46.31
35	0.000990	0.000990	98354.3	97.3	45.35

National life tables, Females

age	mx	qx	lx	dx	ex
0	0.003525	0.003519	100000.0	351.9	83.12
1	0.000211	0.000211	99648.1	21.0	82.42
2	0.000113	0.000113	99627.1	11.3	81.43
3	0.000093	0.000093	99615.9	9.2	80.44
4	0.000061	0.000061	99606.6	6.0	79.45
5	0.000079	0.000079	99600.6	7.8	78.45
6	0.000069	0.000069	99592.7	6.9	77.46
7	0.000051	0.000051	99585.8	5.0	76.47
8	0.000053	0.000053	99580.8	5.2	75.47
9	0.000056	0.000056	99575.6	5.5	74.47
10	0.000065	0.000065	99570.0	6.5	73.48
11	0.000056	0.000056	99563.6	5.6	72.48
12	0.000054	0.000054	99558.0	5.3	71.49
13	0.000088	0.000088	99552.7	8.7	70.49
14	0.000094	0.000094	99544.0	9.4	69.50
15	0.000102	0.000102	99534.6	10.2	68.50
16	0.000129	0.000129	99524.4	12.9	67.51
17	0.000157	0.000157	99511.6	15.6	66.52
18	0.000205	0.000205	99496.0	20.4	65.53
19	0.000202	0.000202	99475.5	20.1	64.54
20	0.000177	0.000177	99455.5	17.6	63.56
21	0.000195	0.000195	99437.9	19.4	62.57
22	0.000232	0.000232	99418.5	23.1	61.58
23	0.000200	0.000200	99395.4	19.8	60.59
24	0.000215	0.000215	99375.5	21.4	59.60
25	0.000251	0.000251	99354.2	24.9	58.62
26	0.000253	0.000253	99329.2	25.2	57.63
27	0.000290	0.000290	99304.1	28.8	56.65
28	0.000299	0.000299	99275.3	29.7	55.66
29	0.000318	0.000318	99245.6	31.6	54.68
30	0.000374	0.000374	99214.0	37.1	53.70
31	0.000366	0.000366	99176.9	36.3	52.72
32	0.000438	0.000437	99140.5	43.4	51.74
33	0.000472	0.000472	99097.2	46.8	50.76
34	0.000549	0.000549	99050.4	54.4	49.78
35	0.000560	0.000560	98996.0	55.4	48.81

Contents | Cover sheet | Notes | Notation | Methodology | 2018-2020 | 2017-2019 | 2016-2018 | 2015-2017 | 2014-2016 | 2013-2015

MANCHESTER 1824
The University of Manchester

ONS life tables

```r
strURL ← paste0("https://www.ons.gov.uk/file?",
                "uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
                "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx")

tmp ← tempfile(fileext = ".xlsx")
httr::GET(strURL, httr::write_disk(tmp))

tblLT ← readxl::read_excel(path  = tmp,
                           sheet = "2018-2020",
                           range = "A6:M107")
```

MANCHESTER 1824
The University of Manchester

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
```

MANCHESTER 1824
The University of Manchester

ONS life tables

```r
strURL ← paste0("https://www.ons.gov.uk/file?",
                "uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
                "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx")

tmp ← tempfile(fileext = ".xlsx")
httr::GET(strURL, httr::write_disk(tmp))

tblLT ← readxl::read_excel(path  = tmp,
                           sheet = "2018-2020",
                           range = "A6:M107")

tblLT ▷
  dplyr::select(!c(...7, age...8)) ▷
  dplyr::rename(age = age...1) ▷
  tidyr::pivot_longer(cols      = !age,
                      names_to  = "tmp",
                      values_to = "val") ▷
  tidyr::separate(col  = tmp,
                  into = c("par", "sex")) ▷
  dplyr::mutate(sex       = ifelse(as.numeric(sex)<7, "Male", "Female")) ▷
  tidyr::pivot_wider(id_cols     = c(age, sex),
                     values_from = val,
                     names_from  = par)
```

MANCHESTER 1824
The University of Manchester

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
```

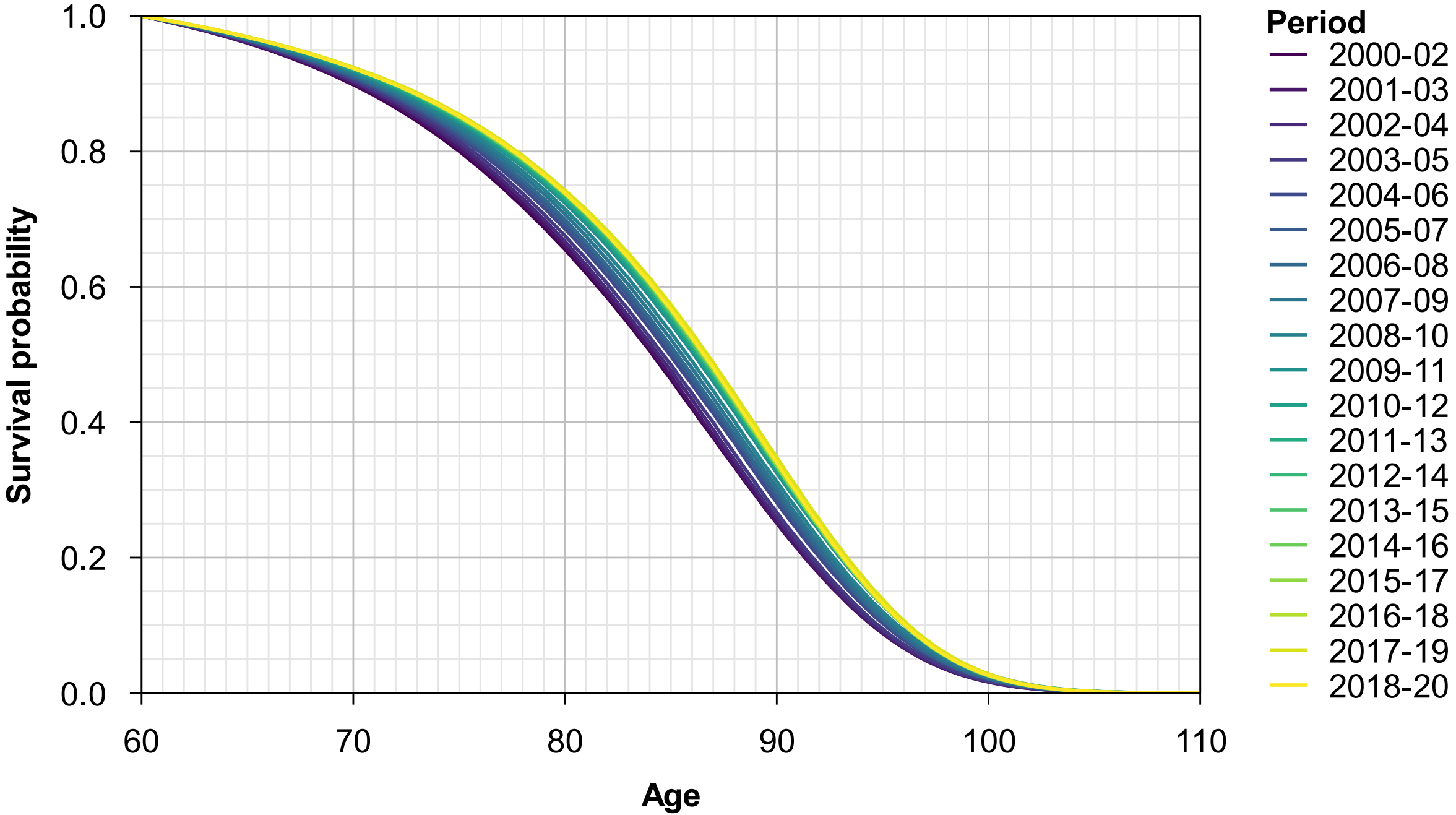
MANCHESTER 1824

The University of Manchester

ONS life tables

Making it functional

```r
fnONSLifeTable← function(years       = 1980:2018,
                         ages        = 0:100,
                         sexes       = c("Male", "Female"),
                         countries   = c("UK", "Eng", "Sco", "Wal", "EngWal"),
                         spans       = c("3-year", "single year"),
                         extrapolate = FALSE,
                         maxExtrap   = 120,
                         widthExtrap = 10) {
  countries ← match.arg(countries)
  spans ← match.arg(spans)
  url ← paste0(switch(spans,
                      "3-year"      = "https://www.ons.gov.uk/file?uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
                      "single year" = "https://www.ons.gov.uk/file?uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fsingleyearlifetablesuk1980to2018%2fsingleyearlifetables"),
               switch(spans,
                      "3-year"      = switch(countries,
                                             "Eng"    = "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx",
                                             "Sco"    = "scotlandreferencetables%2fcurrent/nationallifetables3yrsco.xlsx",
                                             "Wal"    = "walesreferencetables%2fcurrent/nationallifetables3yrwal.xlsx",
                                             "EngWal" = "englandandwalesreferencetables%2fcurrent/nationallifetables3yrew.xlsx",
                                             "unitedkingdomreferencetables%2fcurrent/nationallifetables3yruk.xlsx"),
                      "single year" = switch(countries,
                                             "Eng"    = "england/singleyearlifetablese.xlsx",
                                             "Sco"    = "scotland/singleyearlifetabless.xlsx",
                                             "Wal"    = "wales/singleyearlifetablesw1.xlsx",
                                             "EngWal" = "englandandwales/singleyearlifetablesew.xlsx",
                                             "uk/singleyearlifetablesuk.xlsx")))
  GET(url, write_disk(tmp ← tempfile(fileext = ".xlsx")))
  tblYrs ← tibble(path  = tmp,
                  sheet = tmp %>%
                    excel_sheets() %>%
                    str_subset(pattern = if_else(spans=="3-year", "\\d{4}-\\d{4}", "\\d{4}")),
                  range = "A6:M107")
  tblLT ← tblYrs %>%
    filter(substr(sheet, 1, 4) %in% years) %>%
    pmap_dfr(fnReadLifeTable)
  tblLT %<>%
    dplyr::select(!c(...7, age...8)) %>%
    rename(age = age...1) %>%
    pivot_longer(cols      = !c(age, yr),
                 names_to  = "tmp",
                 values_to = "val") %>%
    separate(col  = tmp,
             into = c("par", "sex")) %>%
    mutate(sex       = ifelse(as.numeric(sex)<7, "Male", "Female"),
           year_from = as.numeric(substr(yr, 1, 4)),
           year_to   = year_from + if_else(spans=="3-year", 2, 0)) %>%
    pivot_wider(id_cols     = c(age, year_from, year_to, sex),
                values_from = val,
                names_from  = par) %>%
    rowwise() %>%
    mutate(lx = ifelse(min(ages)==0, lx, NA),
           dx = ifelse(min(ages)==0, dx, NA)) %>%
    filter(year_from %in% years,
           age       %in% ages,
           sex       %in% sexes) %>%
    arrange(-year_from, sex, age) %>%
    group_by(sex, year_from, year_to)

  if (extrapolate) {
    maxA ← max(tblLT$age)
    tblLT %<>%
      slice_tail(n=widthExtrap) %>%
      nest() %>%
      mutate(lm_obj  = map(data, ~ lm(log(mx)~log(age), data=.x)),
             preddat = list(tibble(age = (maxA+1):(maxExtrap+1))),
             pred    = map2(lm_obj, preddat, predict)) %>%
      unnest(c(preddat, pred)) %>%
      mutate(mx = exp(pred)) %>%
      select(!c(data, lm_obj, pred)) %>%
      bind_rows(tblLT)
  }
  if (extrapolate | min(ages)>0) {
    tblLT %<>%
      arrange(-year_from, sex, age) %>%
      mutate(qx  = coalesce(qx, fnRateToProb(mx)),
             llx = if_else(row_number()==1, 100000, lag(100000*cumprod(1-qx), 1)),
             lx  = coalesce(lx, llx),
             dx  = coalesce(dx, lx*qx),
             Lx  = (lx+lead(lx,1))/2,
             Tx  = rev(cumsum(rev(coalesce(Lx, 0)))),
             Ex  = Tx/lx,
             ex  = coalesce(ex, Ex)) %>%
      filter(age ≤ maxExtrap) %>%
      dplyr::select(!c(llx, Lx, Tx, Ex))
  }
  tblLT %>%
    ungroup()
}
```


Survival probability
1.0
0.8
0.6
0.4
0.2
0.0
60
70
80
90
100
110
Age
Period
2000-02
2001-03
2002-04
2003-05
2004-06
2005-07
2006-08
2007-09
2008-10
2009-11
2010-12
2011-13
2012-14
2013-15
2014-16
2015-17
2016-18
2017-19
2018-20

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

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)

https://www.england.nhs.uk/costing-in-the-nhs/national-cost-collection/

Costing in the NHS
National Cost Collection for the NHS
Costing glossary
Approved Costing Guidance

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.

https://www.england.nhs.uk/costing-in-the-nhs/national-cost-collection/

Home News Public

NHS 75 England

About us Our v

Search

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.

age unit cost of providing
st based on the specific

of work streams including
gramme and NHS
NHS services.

https://www.england.nhs.uk/costing-in-the-nhs/national-cost-collection/

Home News Publications Statistics Blogs Events Contact us

NHS 75 England

About us Our w

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.

Cost
Nati
the N
Cost
Appr

ding
cific

of work streams including
gramme and NHS
NHS services.

https://webarchive.nationalarchives.gov.uk/ukgwa/20200501111106/https://improvement.nhs.uk/resources/reference-costs/

NHS Improvement

Search

About us Improvement Hub Resources Events News and alerts Contact us

Home > Resources > Archived Reference Costs

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.

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.

https://webarchive.nationalarchives.gov.uk/ukgwa/20200507194427/https://www.gov.uk/government/collections/nhs-reference-costs

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

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

https://webarchive.nationalarchives.gov.uk/ukgwa/+/www.dh.gov.uk/en/Managingyourorganisation/NHScostingmanual/DH_129310

DH Department of Health

NHS costing

- 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

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 refernce 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

MANCHESTER 1824

The University of Manchester

NHS Reference Costs

Making it functional

```r
fnGetZipCSV ← function(url, csv, subzip = "") {
  print("Downloading...")
  zip ← tempfile()
  GET(url    = url,
      config = write_disk(zip),
      progress())
  if (subzip ≠ "") {
    print("Unzipping subzip...")
    zip ← unzip(zipfile   = zip,
                files     = subzip,
                exdir     = dirname(tempdir()),
                junkpaths = T)
  }
  print("Unzipping...")
  csv ← unzip(zipfile   = zip,
              files     = csv,
              exdir     = dirname(tempdir()),
              junkpaths = T)
  unlink(zip)
  fread(file = csv)
}
```

```r
fnDownloadParseCSV ← function(url, subzip, csv, Yr) {
  print(Yr)
  fnGetZipCSV(url    = url,
              subzip = subzip,
              csv    = csv) %!>%
    fnParseRefCostsCSV(Yr = Yr)
}
```

```r
fnParseRefCostsCSV ← function(csv, Yr, blnXS = TRUE) {
  print("Parsing...")
  csv %<>%
    clean_names(case = "upper_camel") %>%
    rename(any_of(c(CurrencyCode = "CurrenyCode",
                    Mean         = "NationalMean",
                    DeptCode     = "DepartmentCode",
                    Activity     = "TotalActivity",
                    OrgCode      = "ProviderCode"))) %>%
    mutate(across(.cols = any_of(c("Activity", "Mean", "ExpectedCost")),
                  .fns  = ~ as.numeric(na_if(as.character(.x), "*"))),
           across(.cols = any_of(c("ServiceCode")),
                  .fns  = ~ as.character(.x)),
           DeptCode = str_replace_all(DeptCode,
                                      c("TEI$|EI$"            = "EL",
                                        "TEIXS$|EI_XS$"       = "EL_XS",
                                        "TNEI_L$|NEI_L$"      = "NEL",
                                        "TNEI_L_XS$|NEI_L_XS$" = "NEL_XS",
                                        "TNEI_S$|NEI_S$"      = "NES",
                                        "TDC$"                = "DC",
                                        "TOPROC"              = "OPROC",
                                        "\\GTCL"              = "CL",
                                        "\\GTNCL"             = "NCL"))) %>%
    select(-any_of(c("Memo1", "ScaledMff", "OrganisationName", "DepartmentName", "ServiceName", "CurrencyName"))) %>%
    add_column(Yr = Yr) %>%
    as_tibble()
  if (Yr==2008) {
    csv %<>%
      rename(any_of(c(Activity = "Fce",
                      Activity = "ActivityP1"))) %>%
      select(-any_of(c("Flag", "ActivityP2", "ActivityP3", "ActivityP4")))
    if ("BedDays" %in% names(csv)) {
      csv %<>%
        pivot_longer(cols          = contains(c("UnitCost", "Activity")),
                     names_to      = c("tmp1", "tmp2"),
                     names_pattern = "(.*)(UnitCost|Activity)") %>%
        drop_na(value) %>%
        mutate(DeptCode = paste0(DeptCode, ifelse(tmp1 == "ExcessBedDays", "_XS", "")),
               BedDays  = ifelse(tmp1 == "ExcessBedDays", NA, BedDays)) %>%
        pivot_wider(names_from = tmp2,
                    values_from = value) %>%
        select(-tmp1)
    }
    csv %<>%
      mutate(ActualCost = UnitCost * Activity)
  }
  csv %<>%
    bind_rows(tibble(OrgType = character(),
                     SupplierType = character(),
                     UnitCost = numeric(),
                     BedDays = numeric()))
}
```

MANCHESTER 1824
The University of Manchester

NHS Reference Costs

Making it functional

```
csv %<>%
    clean_names(case = "upper_camel") %>%
    rename(any_of(c(CurrencyCode = "CurrenyCode",
                    Mean         = "NationalMean",
                    DeptCode     = "DepartmentCode",
                    Activity     = "TotalActivity",
                    OrgCode      = "ProviderCode"))) %>%
    mutate(across(.cols = any_of(c("Activity", "Mean", "ExpectedCost")),
                  .fns  = ~ as.numeric(na_if(as.character(.x), "*"))),
           across(.cols = any_of(c("ServiceCode")),
                  .fns  = ~ as.character(.x)),
           DeptCode = str_replace_all(DeptCode,
                                      c("TEI$|EI$"             = "EL",
                                        "TEIXS$|EI_XS$"        = "EL_XS",
                                        "TNEI_L$|NEI_L$"       = "NEL",
                                        "TNEI_L_XS$|NEI_L_XS$" = "NEL_XS",
                                        "TNEI_S$|NEI_S$"       = "NES",
                                        "TDC$"                 = "DC",
                                        "TOPROC"               = "OPROC",
                                        "\\GTCL"               = "CL",
                                        "\\GTNCL"              = "NCL"))) %>%
    select(-any_of(c("Memo1", "ScaledMff", "OrganisationName", "DepartmentName", "ServiceName", "CurrencyName")))
```

MANCHESTER 1824
The University of Manchester

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... ""
```

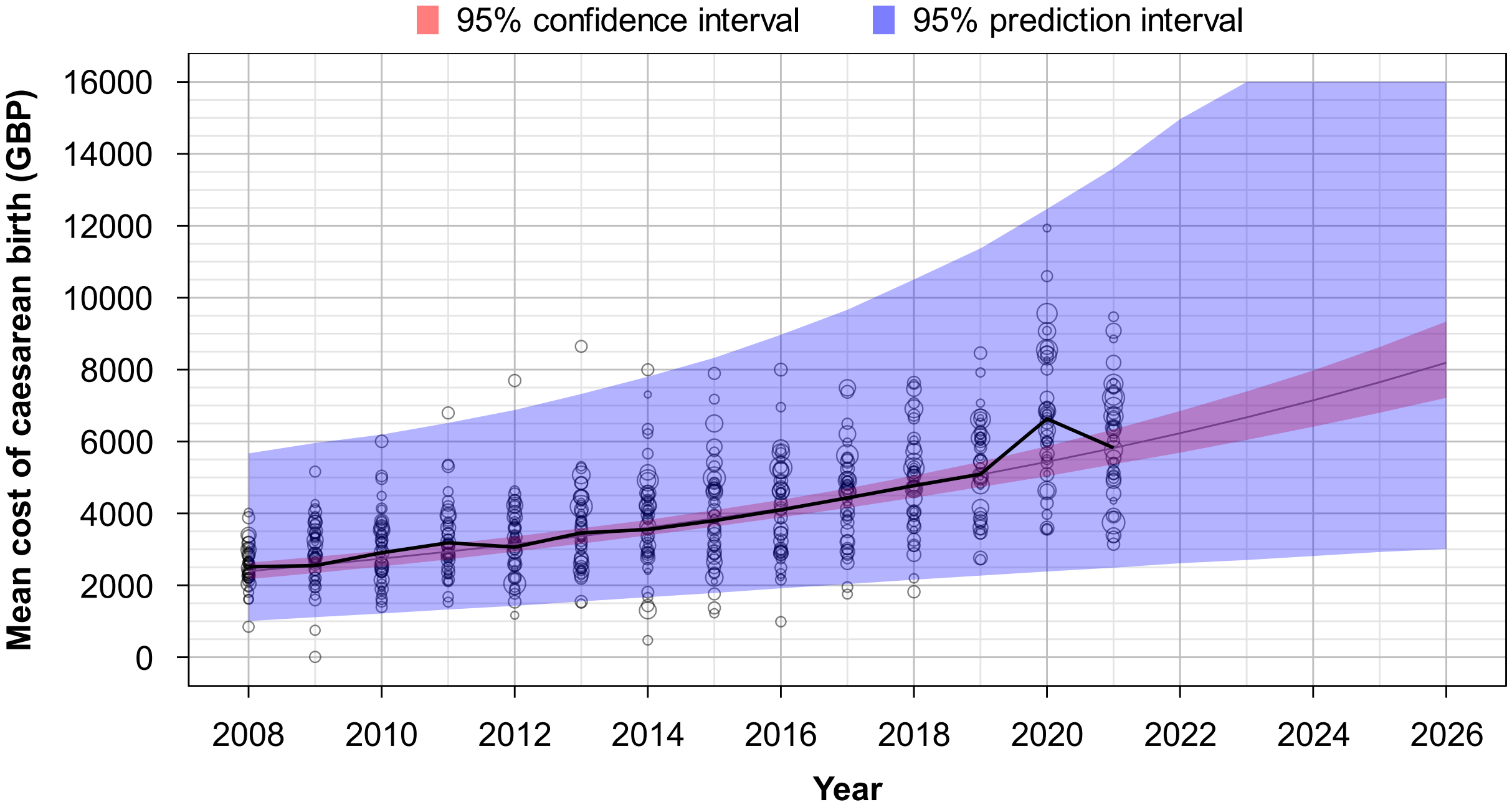
```
tblD ← tblY %>%
  filter(between(Yr, 2008, 2021)) %>%
  pmap(.f        = fnDownloadParseCSV,
       .progress = T) %>%
  list_rbind()
```

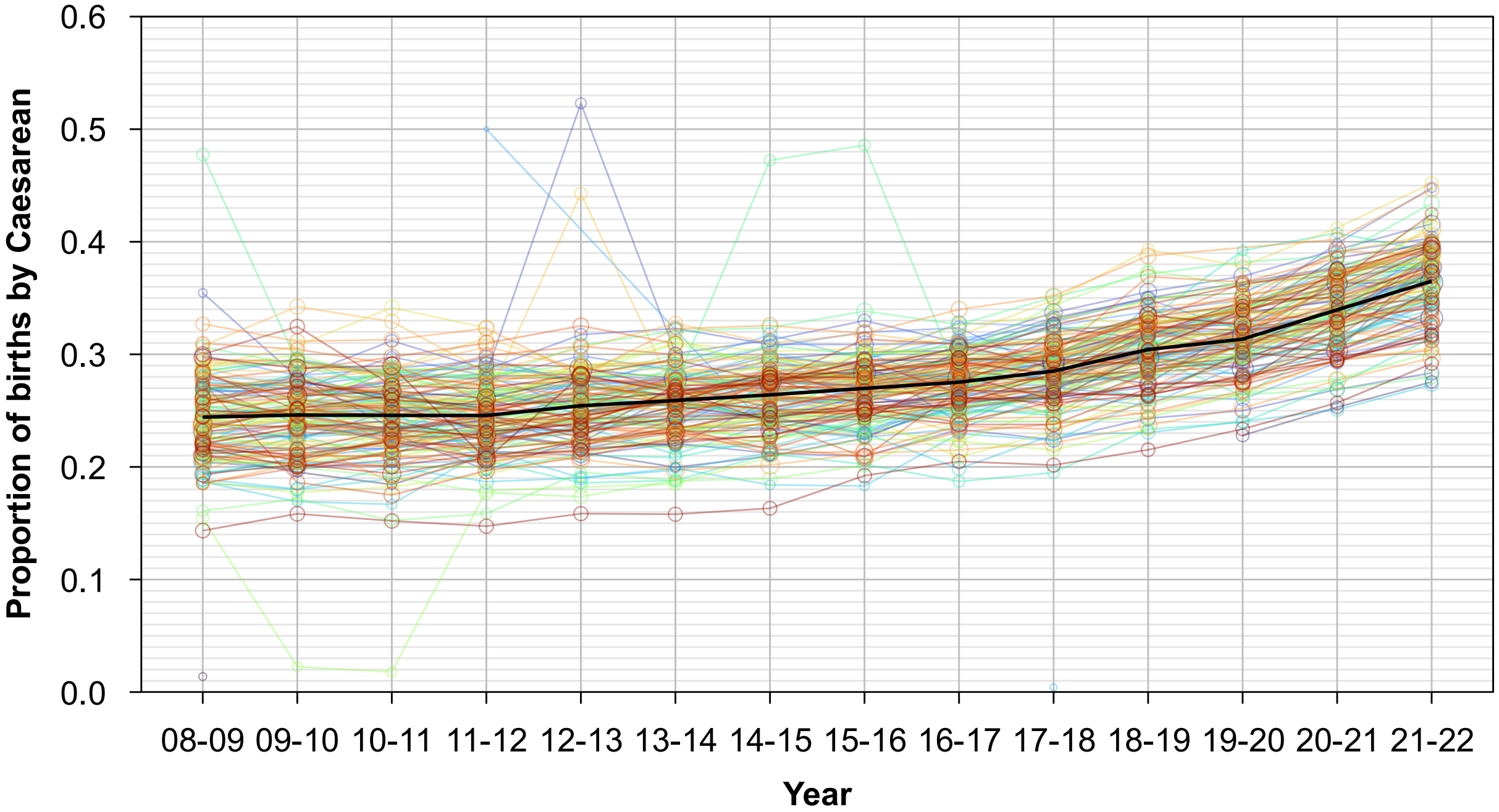
MANCHESTER 1824
The University of Manchester

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> <chr>
 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
```


95% confidence interval
95% prediction interval
Mean cost of caesarean birth (GBP)
16000
14000
12000
10000
8000
6000
4000
2000
0
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
Year


Proportion of births by Caesarean
0.6
0.5
0.4
0.3
0.2
0.1
0.0
08-09
09-10
10-11
11-12
12-13
13-14
14-15
15-16
16-17
17-18
18-19
19-20
20-21
21-22
Year

MANCHESTER 1824
The University of Manchester

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)

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

WHAT COULD *POSSIBLY* GO WRONG?

MANCHESTER 1824
The University of Manchester

What could possibly go wrong?

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

MANCHESTER 1824
The University of Manchester

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... ""
```

```
tblD ← tblY %>%
  filter(between(Yr, 2008, 2021)) %>%
  pmap(.f        = fnDownloadParseCSV,
       .progress = T) %>%
  list_rbind()
```

MANCHESTER 1824
The University of Manchester

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

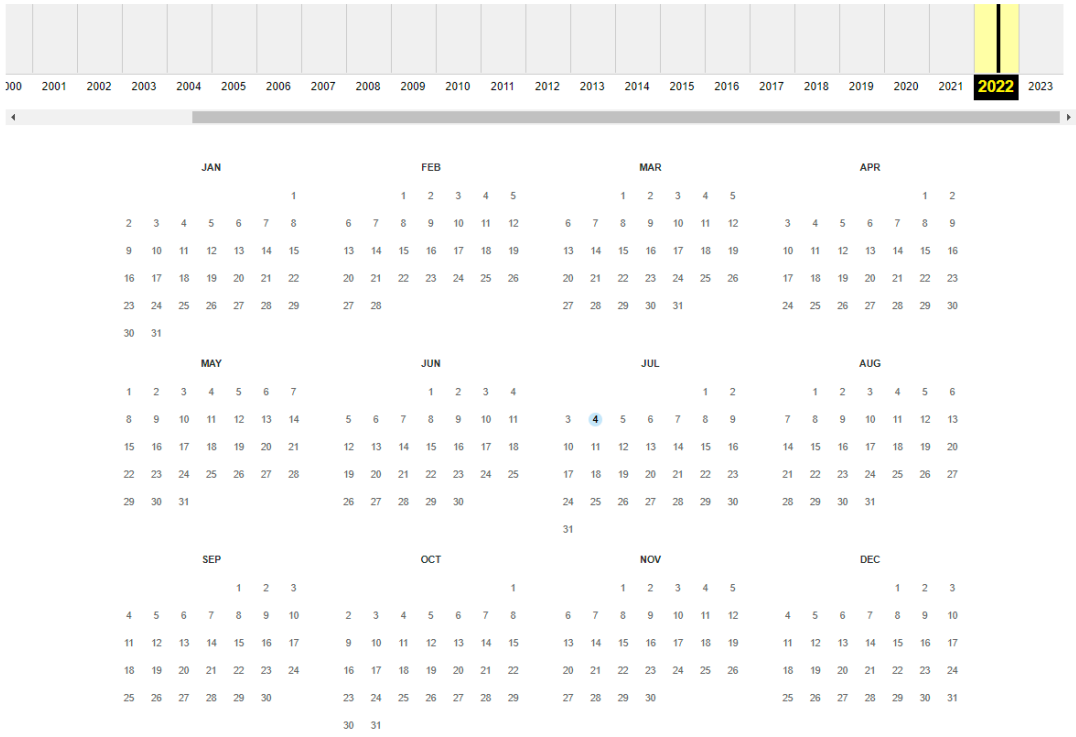
DONATE

INTERNET ARCHIVE
WayBackMachine

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.



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_1.zip** was crawled by the Wayback Machine, *not* how many times the site was actually updated. More info in the FAQ.

INTERNET ARCHIVE

https://web.archive.org/

DONATE

INTERNET ARCHIVE
WayBackMachine

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

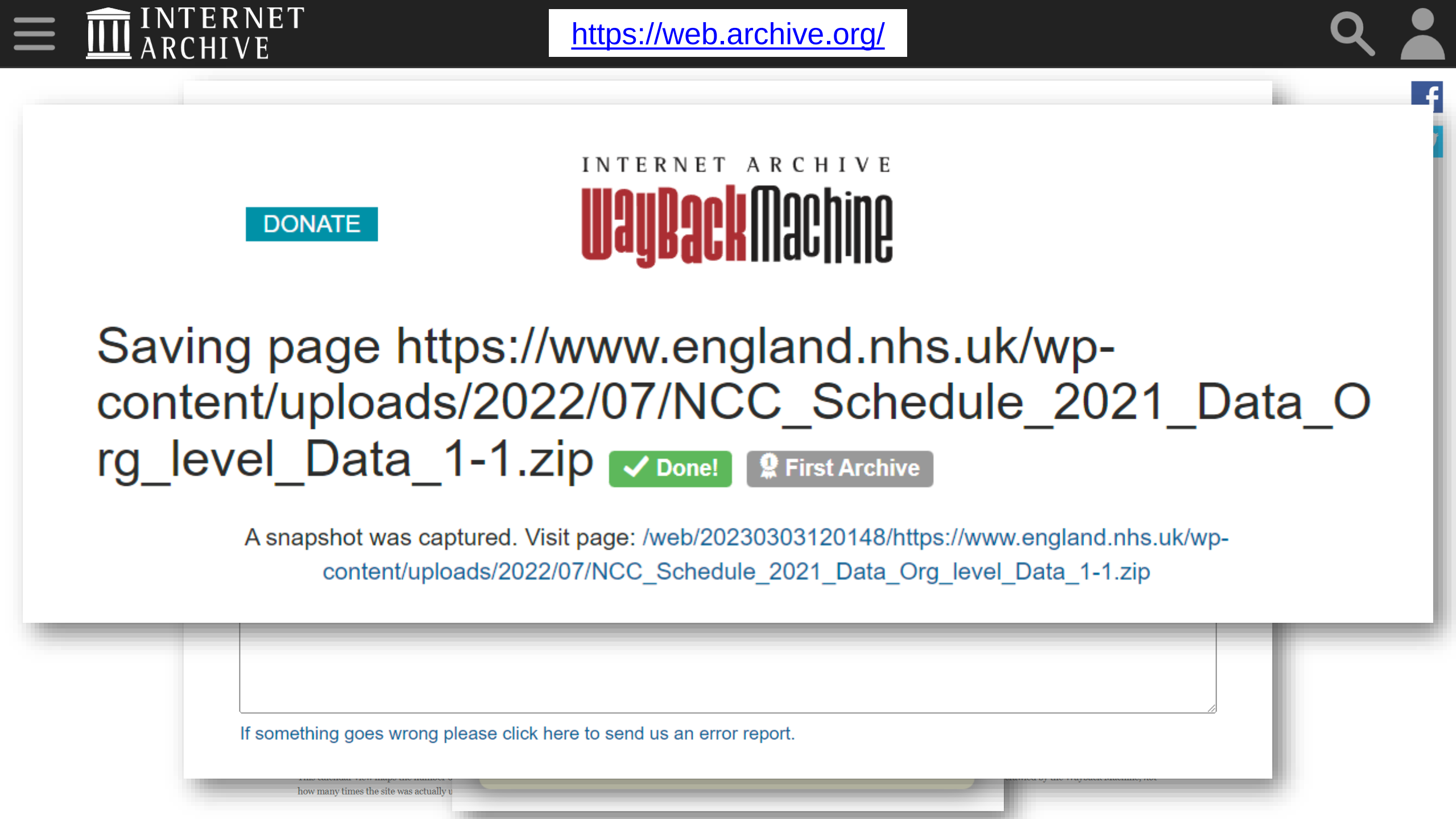
Note

This calendar view maps the number o

how many times the site was actually u

crawled by the Wayback Machine, *not*


INTERNET ARCHIVE
https://web.archive.org/
DONATE
DONATE
INTERNET ARCHIVE
WayBackMachine
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.


INTERNET ARCHIVE
https://web.archive.org/
DONATE
INTERNET ARCHIVE
WayBackMachine
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.

MANCHESTER 1824
The University of Manchester

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 indictive price has now been retired and no further data will be added to that page.

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional

```r
fnNHSOpenData ← function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {
  strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="
  res ← paste0(dataset, "_", Year, sprintf("%02d", Month))
  strSQL ← paste("SELECT", strSelect, "FROM `{res}`")
  if (strWhere ≠ "") strSQL ← paste(strSQL, "WHERE", strWhere)
  if (strGroupBy ≠ "") strSQL ← paste(strSQL, "GROUP BY", strGroupBy)

  url ← glue(strURL, strSQL) ▷
    URLencode()

  tibble(Dataset = res) ▷
    bind_cols(fromJSON(url) ▷
                as_tibble())
}
```

MANCHESTER 1824
The University of Manchester

NHS Open Data Portal API

Making it functional

```
fnNHSOpenData ← function(dataset = c("EPD", "SCMD"), Year, Month, strSelect = "*", strWhere = "", strGroupBy = "") {
  strURL ← "https://opendata.nhsbsa.net/api/3/action/datastore_search_sql?resource_id={res}&sql="
  if (dataset=="SCMD") dataset ← paste0("SCMD_", c("FINAL", "WIP", "PROVISIONAL"))
  res ← paste0(dataset, "_", Year, sprintf("%02d", Month))
  strSQL ← paste("SELECT", strSelect, "FROM `{res}`")
  if (strWhere ≠ "") strSQL ← paste(strSQL, "WHERE", strWhere)
  if (strGroupBy ≠ "") strSQL ← paste(strSQL, "GROUP BY", strGroupBy)

  url ← glue(strURL, strSQL) ▷
    URLencode()
  for (i in 1:length(url)) {
    suppressWarnings(xx ← tryCatch(fromJSON(url[i]), error=function(e) e))
    if (!inherits(xx, "error")) {
      return(tibble(Dataset = res[i]) ▷
               bind_cols(xx$result$result$records ▷
                           as_tibble()))
    }
  }
}
```

MANCHESTER 1824
The University of Manchester

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)

MANCHESTER 1824
The University of Manchester

ONS life tables

```r
strURL ← paste0("https://www.ons.gov.uk/file?",
                "uri=%2fpeoplepopulationandcommunity%2fbirthsdeathsandmarriages%2flifeexpectancies%2fdatasets%2fnationallifetables",
                "englandreferencetables%2fcurrent/nationallifetables3yreng.xlsx")

tmp ← tempfile(fileext = ".xlsx")
httr::GET(strURL, httr::write_disk(tmp))

tblLT ← readxl::read_excel(path  = tmp,
                           sheet = "2018-2020",
                           range = "A6:M107")
```

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

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

https://www.england.nhs.uk/costing-in-the-nhs/national-cost-collection/

Home News Public

NHS 75 England

About us Our v

Search

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.

age unit cost of providing
st based on the specific

of work streams including
gramme and NHS
NHS services.

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

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

MANCHESTER 1824
The University of Manchester

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!

MANCHESTER 1824
The University of Manchester

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!

MANCHESTER 1824
The University of Manchester

All code available at: https://github.com/gbrlrgrs/R-HTA-2023


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


gabriel.rogers@manchester.ac.uk


@gbrlrgrs


github.com/gbrlrgrs