Analysis steps, data sources and a worked example of SII calculation

Authors: Aidan Flatt, Dan Hungerford, David Taylor-Robinson

Uploaded to online repository 13th August 2024

# A detailed explanation of the steps of analysis

The following steps also correspond to the **individual R code files uploaded in the repository**:

## Step 1: Deprivation data curation and cleaning

The data for ascertaining the IMD deprivation scores was collection from the API dashboard within the PHE fingertips dashboard which can be found at this [link](https://fingertips.phe.org.uk/search/deprivation). For further transparency we have also uploaded this dataset into the repository under the file name ‘Full dataset API deprivation scores’.

The data was cleaned to include only the relevant information including GP practice information and the deprivation scores for each practice by the 2019 IMD score given. Deprivation deciles were then assigned to each GP practice 1 through 10, with 1 being the least deprived and 10 being the most deprived.

## Step 2: Merging of the COVER data for the whole time period

COVER data providing childhood vaccination uptake rates for the 5 vaccinations being analysed in this study are available at this [link](https://www.gov.uk/government/collections/vaccine-uptake#cover-of-vaccination-evaluated-rapidly-programme), and is produced at local authority and GP practice levels. Our analysis used GP practice level data which became available on COVER from April 2019. Data is provided on a quarterly (ie 3 monthly) basis for the preceding quarter.

Data for each quarter were downloaded from COVER and renamed appropriately (see section 3 below for naming convention). Sixteen datasets were available at the time of data analysis and spanned sixteen quarter periods from April 2019 to March 2023.

For each file, only the relevant information pertaining to the vaccination being analysed was retained. The number of GP practices for the quarter with a code labelled ‘unknown’ was outputted and these practices were removed. The number of the GP practices where the denominator of children was less than 5 (and therefore rates are not explicitly provided to protect patient confidentiality) was outputted. The hypothetical maximum number of children (ie 5 children) lost if these practices were excluded was outputted.

For rotavirus and MMR vaccinations, a column for each GP practice was added to create the number of children susceptible to infection (ie the number of unvaccinated children registered at the practice.

Finally, all 16 files were merged together to be used in the regression and descriptive analyses. The final merged data contained the vaccination uptake rates for the vaccine being analysed for each of the 16 quarters in the study period.

This process was repeated for each of the 5 vaccinations being studied with a separate R file for each – the file for rotavirus vaccination is uploaded to the repository as an example.

### Step 3: Undertake regression models and create the SII outputs

The quarterly COVER data file and the GP deprivation scores file were merged together by GP practice code. A new column was created with the numerators for vaccinated children to allow the calculation of a weighted uptake mean. Data was manipulated to remove GP practices with missing deprivation data, and local authorities highlighted for removal in the exclusion criteria.

Next, the SII was calculated for each quarter. Please see Section 4 below for a detailed breakdown of the ‘weighted deprivation rank’ value created from the process. This value was then used as a continuous exposure variable in the regression models. Restricted and extended models with random effects for local authority code were tested using ANOVA analysis.

SIIs were created for various time points (including start and end points in the study period; pre-, during and post-pandemic; and at annual intervals). Annual intervals were defined as groups of 4 quarters, making up years 2019-2020, 2020-2021, 2021-2022, and 2022-2023. The SII values for each year were summarised in a table, and plotted on a bar chart in Step 4.

### Step 4: Undertake descriptive analyses and SII plots

For the descriptive analysis, the uptake data for all 5 vaccinations was read in and merged with the deprivation data. As above, data was cleaned to remove excluded local authorities and remove practices without a deprivation score.

Summaries of uptake by quarter and with IMD decile were created, followed by a plot of the mean vaccine uptake % by deprivation, over the study time period. The SII values for each year were plotted on a bar chart with error bars. Finally for MMR and rotavirus vaccinations, the cumulative number of children susceptible to infection through under-vaccination was plotted, using published vaccine efficacy values.

# Data sources and accessibility

The following renaming of data files were undertaken by the authors and correspond to the R codes uploaded in the repository.

### Deprivation data:

‘Full dataset API deprivation scores’ – renamed as ‘GP codes with deprivation.csv’

### Vaccination uptake data:

Example for first quarter, downloads from [COVER](https://www.gov.uk/government/collections/vaccine-uptake#cover-of-vaccination-evaluated-rapidly-programme) as:

* 1. ‘COVER\_GP\_Q1\_2019to2020\_experimental\_data\_tables.ods’
  2. Renamed as ‘1 coverGP Apr19Jun19.ods’

Example for second quarter, downloads from COVER as:

1. ‘COVER\_GP\_Q2\_2019to2020\_experimental\_data\_tables.ods’
2. Renamed as ‘2 coverGP Jul19Sep19.ods’

# A worked example of producing the SII for rotavirus vaccination

To undertake analysis using the Slope Index of Inquality (SII) measure, there was a requirement to create a ‘weighted deprivation rank’ value, which when used as a continuous exposure variable in the regression model produces the SII.

The process of getting the output of ‘weighted deprivation rank’ values is shown below with an example and the relevant R codes for each step. The example below shows an excerpt from (the top of) the data frame produced when the COVER vaccination uptake rates and GP deprivation scores are merged.

GP practices, with their uptake of rotavirus vaccination in Quarter number 1 (April 2019 – June 2019), and their assigned deprivation score, are shown in Frame 1 below. The rank column indicates where in the total ranking amongst the 6500+ GP practices each practice is placed, where rank position 1 would be the least deprived practice for that quarter.

dfQ <- merged\_data[merged\_data$Quarter\_number == 1, ]

order.scores <- order(dfQ$IMD\_2019\_score)

dfQ$rank <- NA

dfQ$rank[order.scores] <- 1:nrow(dfQ)

The ‘percpop’ column is added with the below code, and provides a value to represent the individual GP practice populations expressed as proportion of the total population of children at all GP practices. The sum of these values in the percpop column would equal 1.

total\_denominator <- sum(dfQ$`12m Denom`, na.rm = TRUE)

dfQ <- dfQ %>% mutate(percpop = `12m Denom` / total\_denominator) %>%

filter(!is.na(percpop))

**Frame 1:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Practice code | 12m denom | Rotavacc % uptake | IMD 2019 score | IMD  decile | Quarter number | rank | percpop |
| A81001 | 10 | 100 | 32.3609393 | 8 | 1 | 5240 | 0.00006600 |
| A81002 | 45 | 91.1111111 | 31.9219576 | 8 | 1 | 5174 | 0.00029698 |
| A81004 | 24 | 91.6666667 | 28.0240261 | 7 | 1 | 4541 | 0.00015839 |
| A81005 | 17 | 94.1176471 | 15.3982069 | 4 | 1 | 1953 | 0.00011219 |
| A81006 | 40 | 87.5 | 35.6123871 | 9 | 1 | 5622 | 0.00026398 |
| A81007 | 19 | 100 | 34.4445324 | 9 | 1 | 5497 | 0.00012539 |
| A81009 | 16 | 87.5 | 35.8658492 | 9 | 1 | 5653 | 0.00010559 |
| A81011 | 32 | 100 | 36.308902 | 9 | 1 | 5694 | 0.00021118 |
| A81012 | 11 | 100 | 52.6470209 | 10 | 1 | 6578 | 0.00007259 |
| A81013 | 19 | 94.7368421 | 29.7227528 | 8 | 1 | 4838 | 0.00012539 |
| A81014 | 12 | 91.6666667 | 23.2413016 | 6 | 1 | 3662 | 0.00007259 |
| A81016 | 38 | 89.4736842 | 44.489167 | 10 | 1 | 6280 | 0.00025078 |
| A81017 | 38 | 94.7368421 | 18.9506285 | 5 | 1 | 2726 | 0.00025078 |
| A81018 | 23 | 95.6521739 | 31.6521769 | 8 | 1 | 5134 | 0.00015179 |
| A81019 | 28 | 96.4285714 | 51.0359637 | 10 | 1 | 6538 | 0.00018479 |
| A81020 | 23 | 86.9565217 | 32.5907881 | 8 | 1 | 5273 | 0.00015179 |
| A81021 | 24 | 87.5 | 36.7470315 | 9 | 1 | 5737 | 0.00015839 |
| A81022 | 20 | 85 | 28.1046551 | 7 | 1 | 4554 | 0.00013199 |
| A81023 | 31 | 96.7741935 | 46.7262572 | 10 | 1 | 6378 | 0.00020459 |

Next, the GP practices are sorted into rank order, with Rank 1 at the top of the frame (the least deprived GP practice), seen below in Frame 2.

dfQ <- dfQ[order(dfQ$rank),]

**Frame 2:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Practice code | 12m Denom | Rotavacc % uptake | IMD 2019 score | IMD decile | Quarter number | rank | percpop |
| K82012 | 12 | 100 | 3.40497321 | 1 | 1 | 1 | 0.00007919 |
| K81092 | 38 | 92.1052632 | 3.93329118 | 1 | 1 | 2 | 0.00025078 |
| J82099 | 37 | 83.7837838 | 4.15103119 | 1 | 1 | 3 | 0.00024418 |
| K81028 | 15 | 93.3333333 | 4.15826098 | 1 | 1 | 4 | 0.00009899 |
| K82049 | 24 | 100 | 4.17457488 | 1 | 1 | 5 | 0.00015839 |
| K82051 | 25 | 92 | 4.34925023 | 1 | 1 | 6 | 0.00016499 |
| K81010 | 10 | 100 | 4.36694919 | 1 | 1 | 7 | 0.00006600 |
| E82064 | 11 | 90.9090909 | 4.44444254 | 1 | 1 | 8 | 0.00007259 |
| C88053 | 9 | 100 | 4.51735021 | 1 | 1 | 9 | 0.00005940 |
| K82046 | 45 | 91.1111111 | 4.5662698 | 1 | 1 | 10 | 0.00029698 |
| J82110 | 40 | 95 | 4.60000414 | 1 | 1 | 11 | 0.00026398 |
| N81033 | 19 | 89.4736842 | 4.60072344 | 1 | 1 | 12 | 0.00012539 |
| J82135 | 39 | 84.6153846 | 4.70752641 | 1 | 1 | 13 | 0.00025738 |
| K81070 | 32 | 90.625 | 4.76548024 | 1 | 1 | 14 | 0.00021118 |
| E82077 | 20 | 85 | 4.80682545 | 1 | 1 | 15 | 0.00013199 |
| K81047 | 56 | 94.6428571 | 4.8369766 | 1 | 1 | 16 | 0.00036957 |
| E82106 | 25 | 92 | 4.87409834 | 1 | 1 | 18 | 0.00016499 |
| K82035 | 6 | 100 | 4.91447572 | 1 | 1 | 19 | 0.00003960 |
| J82143 | 40 | 92.5 | 4.95385572 | 1 | 1 | 20 | 0.00026398 |

The cumulative population is then calculated which sums the percpop values moving down the rankings, adding each value consecutively, seen in Frame 3.

dfQ <- dfQ %>% mutate(cumul\_pop = cumsum(percpop))

**Frame 3:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Practice code | 12m Denom | Rotavacc % uptake | IMD 2019 score | IMD decile | Quarter number | rank | percpop | cumul\_pop |
| K82012 | 12 | 100 | 3.40497321 | 1 | 1 | 1 | 0.00007919 | 0.00007919 |
| K81092 | 38 | 92.1052632 | 3.93329118 | 1 | 1 | 2 | 0.00025078 | 0.00032998 |
| J82099 | 37 | 83.7837838 | 4.15103119 | 1 | 1 | 3 | 0.00024418 | 0.00057416 |
| K81028 | 15 | 93.3333333 | 4.15826098 | 1 | 1 | 4 | 0.00009899 | 0.00067315 |
| K82049 | 24 | 100 | 4.17457488 | 1 | 1 | 5 | 0.00015839 | 0.00083154 |
| K82051 | 25 | 92 | 4.34925023 | 1 | 1 | 6 | 0.00016499 | 0.00099653 |
| K81010 | 10 | 100 | 4.36694919 | 1 | 1 | 7 | 0.00006600 | 0.00106252 |
| E82064 | 11 | 90.9090909 | 4.44444254 | 1 | 1 | 8 | 0.00007259 | 0.00113512 |
| C88053 | 9 | 100 | 4.51735021 | 1 | 1 | 9 | 0.00005940 | 0.00119451 |
| K82046 | 45 | 91.1111111 | 4.5662698 | 1 | 1 | 10 | 0.00029698 | 0.00149149 |
| J82110 | 40 | 95 | 4.60000414 | 1 | 1 | 11 | 0.00026398 | 0.00175547 |
| N81033 | 19 | 89.4736842 | 4.60072344 | 1 | 1 | 12 | 0.00012539 | 0.00188087 |
| J82135 | 39 | 84.6153846 | 4.70752641 | 1 | 1 | 13 | 0.00025738 | 0.00213825 |
| K81070 | 32 | 90.625 | 4.76548024 | 1 | 1 | 14 | 0.00021118 | 0.00234943 |
| E82077 | 20 | 85 | 4.80682545 | 1 | 1 | 15 | 0.00013199 | 0.00248142 |
| K81047 | 56 | 94.6428571 | 4.8369766 | 1 | 1 | 16 | 0.00036957 | 0.002851 |
| E82106 | 25 | 92 | 4.87409834 | 1 | 1 | 18 | 0.00016499 | 0.00301598 |
| K82035 | 6 | 100 | 4.91447572 | 1 | 1 | 19 | 0.00003960 | 0.00305558 |
| J82143 | 40 | 92.5 | 4.95385572 | 1 | 1 | 20 | 0.00026398 | 0.00331956 |

The mid-point rank value is then calculated and named the ‘weighted deprivation rank’ (WDR). Using the following code, the column ‘weighted deprivation rank’ is added, shown in Frame 4.

dfQ <- dfQ %>% mutate(weighted\_deprivation\_rank = (dplyr::lag(cumul\_pop) + (percpop / 2)))

**Frame 4:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Practice code |  | rank | percpop | cumul\_pop | weighted\_deprivation\_rank |
| K82012 |  | 1 | 0.00007919 | 0.00007919 | NA |
| K81092 |  | 2 | 0.00025078 | 0.00032998 | 0.00020459 |
| J82099 |  | 3 | 0.00024418 | 0.00057416 | 0.00045207 |
| K81028 |  | 4 | 0.00009899 | 0.00067315 | 0.00062366 |
| K82049 |  | 5 | 0.00015839 | 0.00083154 | 0.00075235 |
| K82051 |  | 6 | 0.00016499 | 0.00099653 | 0.00091403 |
| K81010 |  | 7 | 0.00006600 | 0.00106252 | 0.00102953 |
| E82064 |  | 8 | 0.00007259 | 0.00113512 | 0.00109882 |
| C88053 |  | 9 | 0.00005940 | 0.00119451 | 0.00116482 |
| K82046 |  | 10 | 0.00029698 | 0.00149149 | 0.001343 |
| J82110 |  | 11 | 0.00026398 | 0.00175547 | 0.00162348 |
| N81033 |  | 12 | 0.00012539 | 0.00188087 | 0.00181817 |
| J82135 |  | 13 | 0.00025738 | 0.00213825 | 0.00200956 |
| K81070 |  | 14 | 0.00021118 | 0.00234943 | 0.00224384 |
| E82077 |  | 15 | 0.00013199 | 0.00248142 | 0.00241543 |
| K81047 |  | 16 | 0.00036957 | 0.002851 | 0.00266621 |
| E82106 |  | 18 | 0.00016499 | 0.00301598 | 0.00293349 |
| K82035 |  | 19 | 0.00003960 | 0.00305558 | 0.00303578 |
| J82143 |  | 20 | 0.00026398 | 0.00331956 | 0.00318757 |

Given the Rank 1 practice is used as a lag function, and gives an NA value, the value for the WDR here is calculated as the mid-way point in the cumulative population at this ranking, so the cumulative population is halved, with the value used as the Rank 1 WDR, seen in Frame 5.

dfQ$cumul\_pop\_halved <- (dfQ$cumul\_pop / 2)

dfQ$weighted\_deprivation\_rank[is.na(dfQ$weighted\_deprivation\_rank)] <- as.character(dfQ$cumul\_pop\_halved[is.na(dfQ$weighted\_deprivation\_rank)])

dfQ$weighted\_deprivation\_rank <- as.numeric(dfQ$weighted\_deprivation\_rank)

Frame 5:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Practice code |  | rank | percpop | cumul\_pop | weighted\_deprivation\_rank | cumul\_pop\_halved |
| K82012 |  | 1 | 0.00007919 | 7.91943296859945e-05 | NA | 0.000039597 |
| K81092 |  | 2 | 0.00025078 | 0.00032998 | 0.00020459 | 0.00016499 |
| J82099 |  | 3 | 0.00024418 | 0.00057416 | 0.00045207 | 0.00028708 |
| K81028 |  | 4 | 0.00009899 | 0.00067315 | 0.00062366 | 0.00033658 |
| K82049 |  | 5 | 0.00015839 | 0.00083154 | 0.00075235 | 0.00041577 |
| K82051 |  | 6 | 0.00016499 | 0.00099653 | 0.00091403 | 0.00049826 |
|  |  |  |  |  |  |  |

This process is undertaken as a loop within RStudio, where the process is repeated 16 times for each quarter. The overall code for the loop is shown below:

df\_list <- list()

# Loop through quarters Q1 to Q16

for (quarter\_number in 1:16) {

# Filter data for the current quarter (Q1, Q2, ..., Q16)

dfQ <- merged\_data[merged\_data$Quarter\_number == 1, ]

# Create a column ranking by IMD score

order.scores <- order(dfQ$IMD\_2019\_score)

dfQ$rank <- NA

dfQ$rank[order.scores] <- 1:nrow(dfQ)

# Calculate the sum of '12m Denom' excluding NA values

total\_denominator <- sum(dfQ$`12m Denom`, na.rm = TRUE)

dfQ <- dfQ %>% mutate(percpop = `12m Denom` / total\_denominator) %>%

filter(!is.na(percpop))

# Ordering by rank, calculate the cumulative proportion of the population in each rank

dfQ <- dfQ[order(dfQ$rank),]

dfQ <- dfQ %>% mutate(cumul\_pop = cumsum(percpop))

# Find the mid-point of the range of values for ranks

dfQ <- dfQ %>% mutate(weighted\_deprivation\_rank = (dplyr::lag(cumul\_pop) + (percpop / 2)))

# Rank 1 is NA as used lag function, so need to find a value for this

dfQ$cumul\_pop\_halved <- (dfQ$cumul\_pop / 2)

dfQ$weighted\_deprivation\_rank[is.na(dfQ$weighted\_deprivation\_rank)] <- as.character(dfQ$cumul\_pop\_halved[is.na(dfQ$weighted\_deprivation\_rank)])

dfQ$weighted\_deprivation\_rank <- as.numeric(dfQ$weighted\_deprivation\_rank)

# Add the modified data frame for the current quarter to the list

df\_list[[quarter\_number]] <- dfQ

}

merged\_df <- do.call(rbind, df\_list)