Proof of concept for health indicators

Julian Flowers

2024-07-12

Table of contents

1 Introduction

Outline an end-to-end process for creating public health indicators and generating public health profiles.

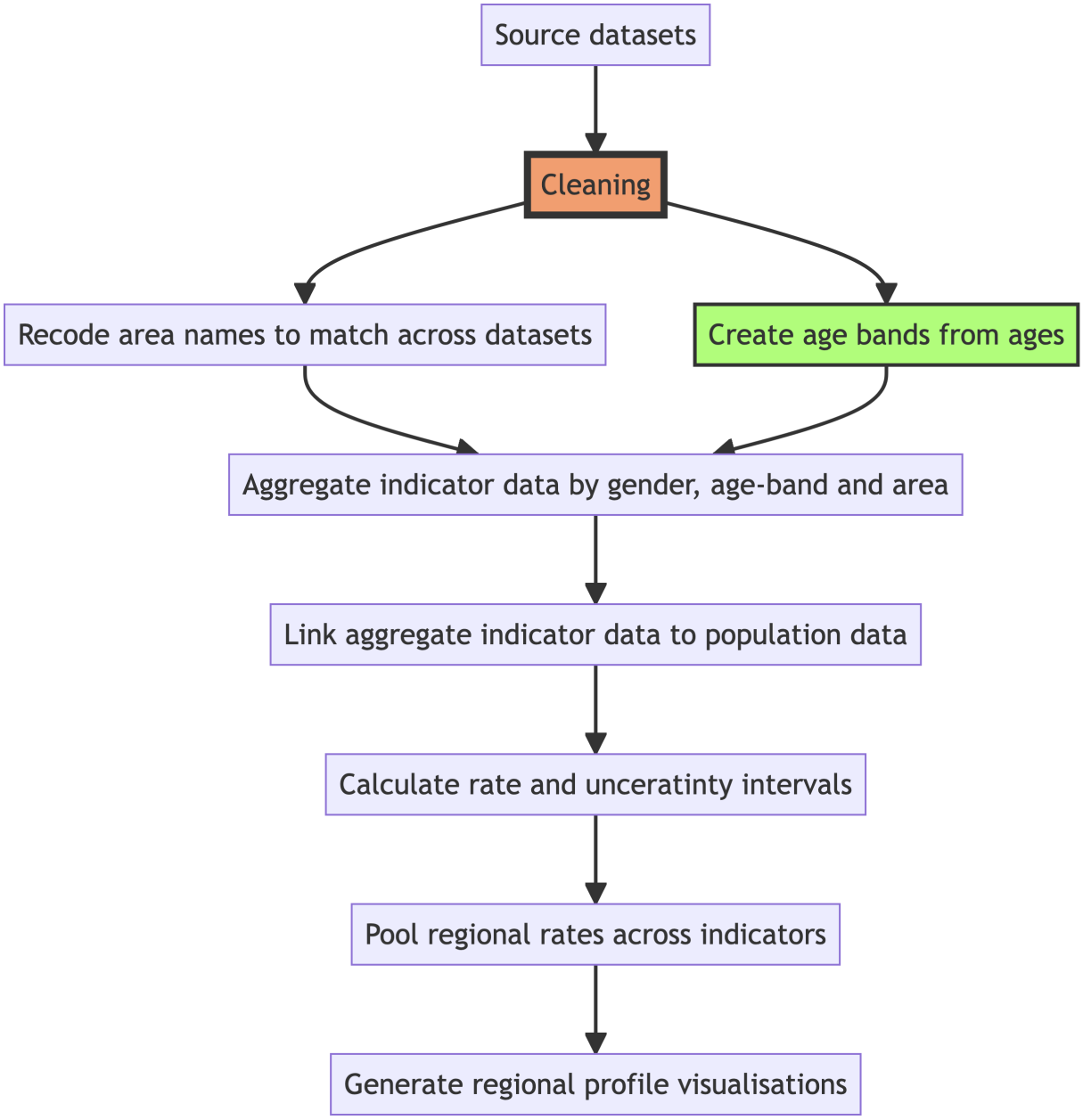


Figure 1.1: Workflow

2 Rapid EDA

A first step is to rapidly evaluate raw data.

In creating regional health indicators and profiles

Store data in a single directory

```
dir <- here("data")

xl_files <- fs::dir_ls(dir, regexp = "xls")

csv_files <- fs::dir_ls(dir, regexp = "csv")

## read_files

xl <- map(xl_files, read_xlsx)
csvs <- map(csv_files, read_csv)
```

```
map(xl, colnames)
map(csvs, colnames)
```

Table 2.1: Area name labels

Dataset	Area field name
AMR	No area variable
Injury	`Region`
Flu	`region_en`
Smoking	`directorate_name`
Populations	`Region`

To facilitate data linkage and creating indicator datasets, area variable names should be consistent between datasets.

Directorate is not equivalent to region.

There are 13 KSA regions and 20 health directorates

2.0.1 Area variable names

```
## rename area variables

csvs$`/Users/julianflowers/poc/data/Flu Vaccine Coverage 2023 updated.csv` <- rename(csvs$`/U

#csvs$`/Users/julianflowers/poc/data/Flu Vaccine Coverage 2023 updated.csv`
```

2.0.2 Area names

```
flu_areas <- csvs$`/Users/julianflowers/poc/data/Flu Vaccine Coverage 2023 updated.csv` |>
    select(Region) |> unique()

smoking_areas <- csvs$`/Users/julianflowers/poc/data/Smoking 2022.csv` |>
    select(directorate_name) |> unique()

pop_areas <- csvs$`/Users/julianflowers/poc/data/Translated_Population_Data_with_Governorates
    select(Region) |> unique()

injury_areas <- xl$`/Users/julianflowers/poc/data/Nonfatal Hospitalizations for Injuries data

n_areas <- data.frame(data = c("flu_areas", "smoking_areas", "pop_areas", "injury_areas"), no
    knitr::kable()
```

The number of unique areas

data	no_areas	area_type
flu_areas	13	region
smoking_areas	20	directorate
pop_areas	13	region
injury_areas	12	region

```
setdiff(flu_areas, pop_areas)
```

```
# A tibble: 9 x 1
  Region
  <chr>
```

```
1 Riyadh
2 Sharqiya
3 Makkah Al Mukarramah
4 Asir
5 madina
6 Al Qassim
7 Hail
8 Al Baha
9 Northern Frontier
```

```
setdiff(pop_areas, injury_areas)
```

```
# A tibble: 10 x 1
   Region
   <chr>
 1 Al Bahah
 2 Al Hudud ash Shamaliyah
 3 Ar Riyadh
 4 Al Qasim
 5 Al Madinah al Munawwarah
 6 Al Mintaqah ash Sharqiyah
 7 Tabuk
 8 Ha'il
 9 'Asir
10 Makkah al Mukarramah
```

```
setdiff(injury_areas, flu_areas)
```

```
# A tibble: 2 x 1
  Region
  <chr>
1 Makkah
2 Madinah
```

3 Creating lookups and mapping geographical areas

3.0.1 Creating a lookup table for KSA regions and health directorates

1. Population estimates by age, gender and region - downloaded from detailed census data 2022. source: https://portal.saudicensus.sa/portal/public/1/15/101464?type=TABLE; translated into English using ChatGPT4o.
2. This gives populations for 13 regions; smoking and injury date is based on health directorates - 20 units.
3. For these analyses aggregated directorates to regions to enable rate calculations
4. To map directorates to regions following steps were undertaken:
    - Shape file for KSA regional boundaries obtained from …
    - Directorate based locations of smoking cessation clinics were scraped from https://www.moh.gov.sa/en/Ministry/Projects/TCP/Pages/default.aspx
    - Locations were spatially joined to KSA regional boundaries to create a region <-> directorate lookup
5. Naming systems differed between datasets so renaming and recoding necessary

```
devtools::install_github("yutannihilation/ggsflabel")
needs(tidyverse, data.table, readxl, myScrapers, sf, curl, ggsflabel)

pops <- fread("/Users/julianflowers/Library/CloudStorage/GoogleDrive-julian.flowers12@gmail.c

region_names <- pops$Region |> unique()

region_names |>
    enframe()
```

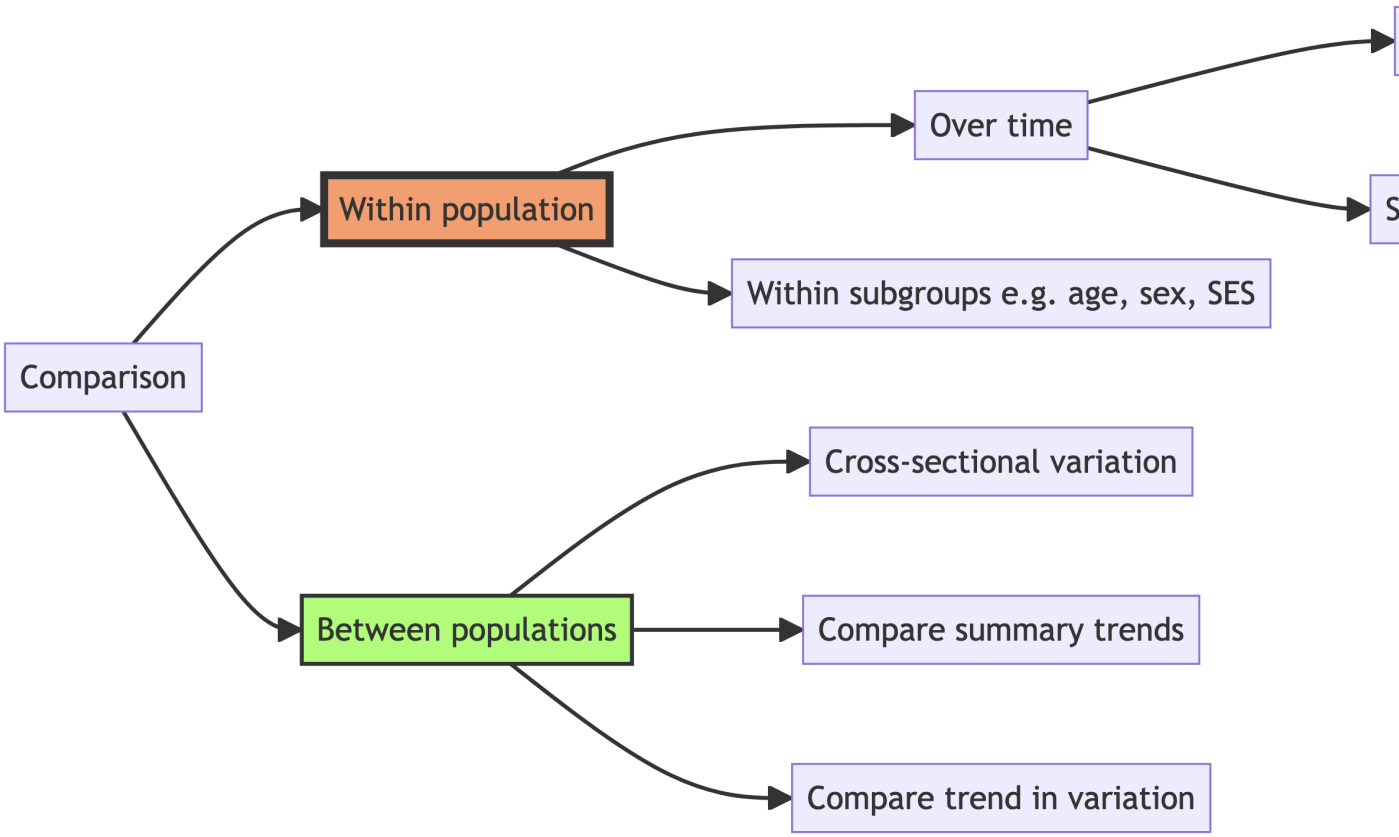


Figure 3.1: Comparative analysis

```
# A tibble: 13 x 2
    name value
   <int> <chr>
 1     1 Al Bahah
 2     2 Al Jawf
 3     3 Al Hudud ash Shamaliyah
 4     4 Ar Riyadh
 5     5 Al Qasim
 6     6 Al Madinah al Munawwarah
 7     7 Al Mintaqah ash Sharqiyah
 8     8 Tabuk
 9     9 Jazan
10    10 Ha'il
11    11 'Asir
12    12 Makkah al Mukarramah
13    13 Najran
```

```
## region names for injury data (NB only 12 names)
df_r <- read_xlsx("/Users/julianflowers/spha/data/fwdatastrategypocpublichealthframeworkindi

## directorate names for smoking data
smok <- read_csv("/Users/julianflowers/spha/data/fwdatastrategypocpublichealthframeworkindic
```

```
url <- "https://www.moh.gov.sa/en/Ministry/Projects/TCP/Pages/default.aspx"

scc_dir <- get_page_links(url) %>%
  .[159:178]

sc_dir_links <- paste0("https://www.moh.gov.sa", scc_dir)

sc_dir_names <- sc_dir_links |>
  basename()

## extract google maps link of scc for each region and create data frame
sc_loc <- map(sc_dir_links, get_page_links) %>%
  map(\(x) x[grepl("https://goo.gl", x)]) %>%
    set_names(., sc_dir_names) |>
  enframe() |>
    mutate(name = str_remove(name, ".aspx"))
```

```r
get_coordinates_from_google_maps <- function(url) {
  # Follow the redirect to get the final URL
  url <- url
  response <- HEAD(url, config(followlocation = TRUE))
  final_url <- response$url

  # Use a regular expression to find the coordinates in the final URL
  match <- str_match(final_url, "@(-?\\d+\\.\\d+),(-?\\d+\\.\\d+)")
  if (!is.na(match[1,2]) && !is.na(match[1,3])) {
    latitude <- as.numeric(match[1,2])
    longitude <- as.numeric(match[1,3])
    return(list(latitude = latitude, longitude = longitude))
  } else {
    return(NULL)
  }
}
```

```r
sc_coords <- sc_loc |>
  unnest(value) |>
  mutate(ll = map(value, get_coordinates_from_google_maps, .progress = TRUE))

## create table of sc clinic locations
sc_ll <- sc_coords |>
    unnest_wider(ll)

## convert to sf file (need to remove missing coordinate values)

sc_ll_sf <- sc_ll |>
    drop_na() |>
    st_as_sf(coords = c("longitude", "latitude"), crs = 4326)
```

```r
sa_shp <- curl_download("https://data.humdata.org/dataset/41ce9023-1d21-4549-a485-94316200aba

tmpd <- tempdir()

sa_shp_1 <- curl_download("https://data.humdata.org/dataset/41ce9023-1d21-4549-a485-94316200a

#sa_pop_d <- curl_download("https://data.humdata.org/dataset/14b288ca-1855-4025-9f01-41cba548

sa_shp <- unzip(sa_shp, exdir = tmpd)

sa_shp_1 <- unzip(sa_shp_1, exdir = tmpd)
```

```r
#sa_tif <- unzip(sa_pop_d, exdir = tmpd)

shps <- fs::dir_ls(tmpd, regexp = "shp$")

## boundary polygon file
sa_bound <- read_sf(shps[2])
```

```r
sa_bound |>
    ggplot() +
    geom_sf(fill = "grey90") +
    geom_sf_label_repel(aes(label = ADM1_EN)) +
    geom_sf(data = sc_ll_sf, aes(colour = name)) +
    theme_void() +
    scale_colour_viridis_d(option = "turbo", name = "Directorates")
```

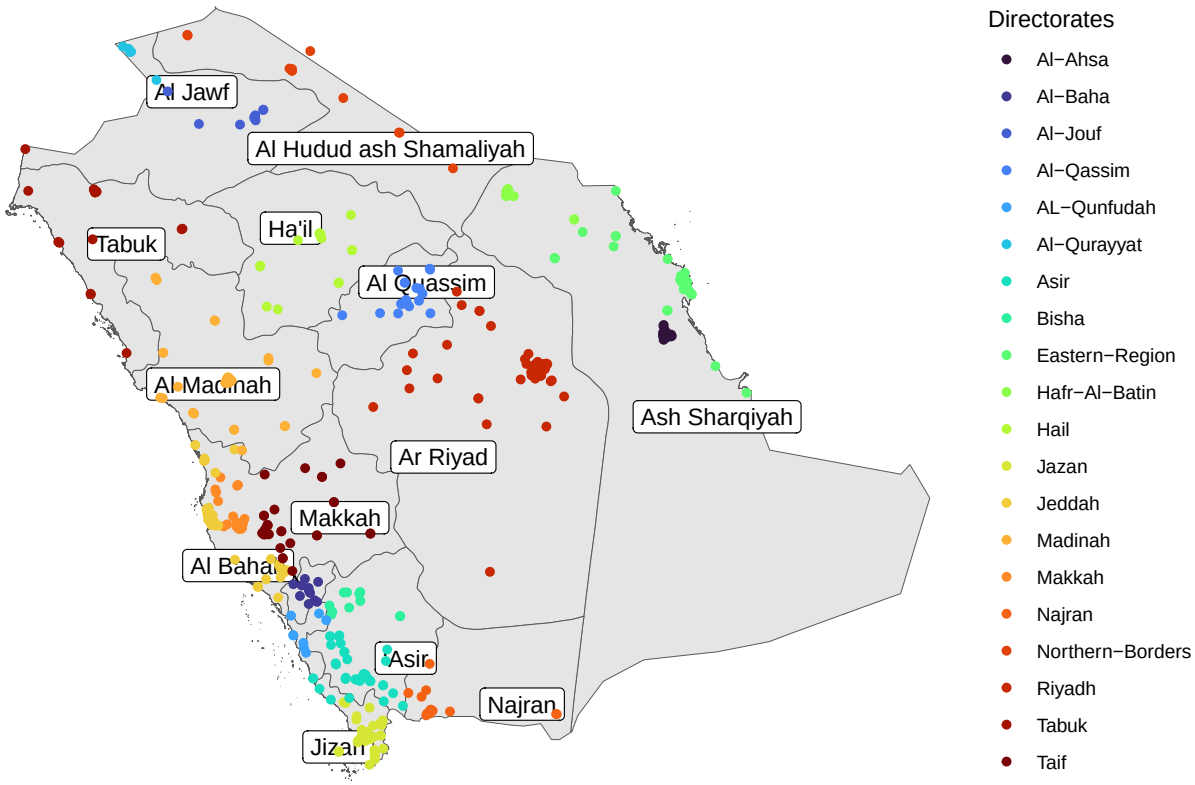


Figure 3.2: SCC location map with regional boundaries

```r
reg_dir_lu <- sa_bound |>
    st_join(sc_ll_sf) |>
    st_drop_geometry() |>
    select(ADM1_EN, name) |>
    group_by(ADM1_EN, name) |>
    summarise(n = n()) |>
    ungroup() |>
    group_by(name) |>
    arrange(name) |>
    filter(n == max(n)) |>
    select(name, everything())
```

Now we want to attach region names tpo the smoking data so we can join with population data in order to calculate attendance rates by age.

```r
pops$Region |>
    unique() |>
    enframe()
```

```
# A tibble: 13 x 2
    name value
   <int> <chr>
 1     1 Al Bahah
 2     2 Al Jawf
 3     3 Al Hudud ash Shamaliyah
 4     4 Ar Riyadh
 5     5 Al Qasim
 6     6 Al Madinah al Munawwarah
 7     7 Al Mintaqah ash Sharqiyah
 8     8 Tabuk
 9     9 Jazan
10    10 Ha'il
11    11 'Asir
12    12 Makkah al Mukarramah
13    13 Najran
```

```r
smok_1 <- smok |>
    mutate(directorate_name = recode(directorate_name, "Qurayyat" = "Al-Qurayyat",
                                     "Qunfotha" = "AL-Qunfudah",
                                     "AlAhsa" = "Al-Ahsa",
                                     "Baha" = "Al-Baha",
```

```
                                    "Eastern" = "Eastern-Region",
                                    "Hafer AlBatin" = "Hafr-Al-Batin",
                                    "Northern Borders" = "Northern-Borders",
                                    "Qassim" = "Al-Qassim",
                                    "Jouf" = "Al-Jouf"
                                    )) |>
    left_join(reg_dir_lu, by = c("directorate_name" = "name"))
    #left_join(pops, by = c("ADM1_EN" = "Region"))
```

```
pops <- pops |>
    mutate(age = parse_number(`Single Age Group`))

pops$Region |>
    unique()
```

```
 [1] "Al Bahah"                  "Al Jawf"
 [3] "Al Hudud ash Shamaliyah"   "Ar Riyadh"
 [5] "Al Qasim"                  "Al Madinah al Munawwarah"
 [7] "Al Mintaqah ash Sharqiyah" "Tabuk"
 [9] "Jazan"                     "Ha'il"
[11] "'Asir"                     "Makkah al Mukarramah"
[13] "Najran"
```

```
smok_pops_region <- smok_1 |>
    mutate(Gender = str_to_title(patient_gender)) |>
    count(ADM1_EN, age, Gender)


## recode region names (ADM1_EN)

# smok_pops_region |>
#     mutate(Region = recode(ADM1_EN,
#                            "`Asir" = "'Asir",
#                            "Ash Sharqiyah" = "Al Hudud ash Sharqiyah",
#                            "Al Madinah" = ))

smok_pops_region <- smok_pops_region |>
    full_join(pops, by = c("ADM1_EN" = "Region", "age", "Gender"))
```

```r
## sense check
smok_pops_region |>
    count(Gender, ADM1_EN, `18-44`) |>
    print(n = 42)
```

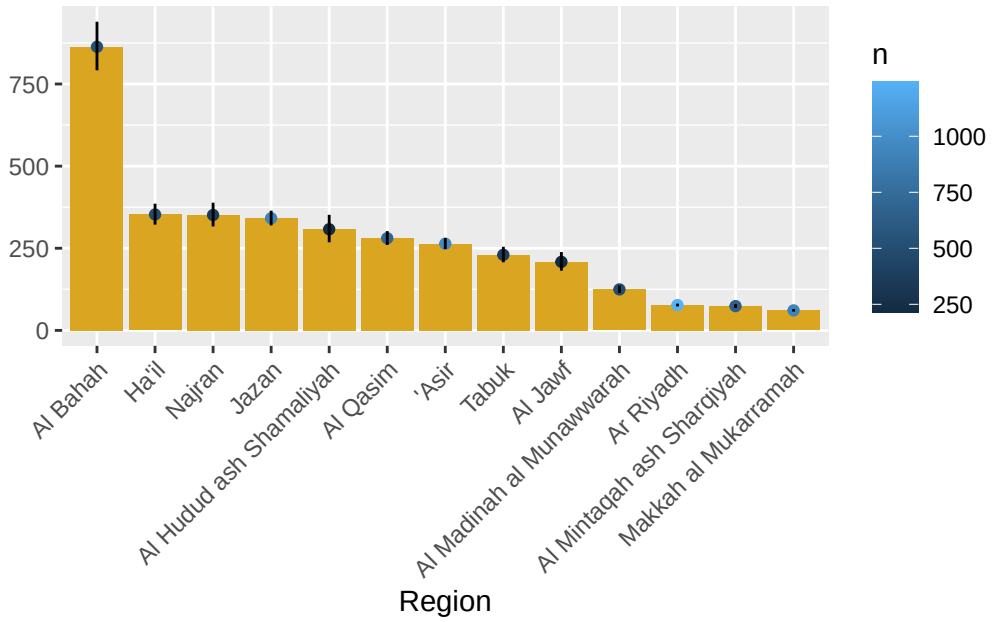
```
# A tibble: 69 x 4
   Gender ADM1_EN                    `18-44`     n
   <chr>  <chr>                      <chr>   <int>
 1 Female 'Asir                      18-44     967
 2 Female 'Asir                      other    2206
 3 Female Al Bahah                   18-44     535
 4 Female Al Bahah                   other    1178
 5 Female Al Hudud ash Shamaliyah    18-44     216
 6 Female Al Hudud ash Shamaliyah    other     522
 7 Female Al Jawf                    18-44     216
 8 Female Al Jawf                    other     522
 9 Female Al Madinah                 <NA>       17
10 Female Al Madinah al Munawwarah   18-44     484
11 Female Al Madinah al Munawwarah   other    1133
12 Female Al Mintaqah ash Sharqiyah  18-44     648
13 Female Al Mintaqah ash Sharqiyah  other    1568
14 Female Al Qasim                   18-44     699
15 Female Al Qasim                   other    1592
16 Female Al Quassim                 <NA>        4
17 Female Ar Riyad                   <NA>       21
18 Female Ar Riyadh                  18-44    1241
19 Female Ar Riyadh                  other    2866
20 Female Ash Sharqiyah              <NA>       19
21 Female Ha'il                      18-44     482
22 Female Ha'il                      other    1045
23 Female Jazan                      18-44     913
24 Female Jazan                      other    2270
25 Female Jizan                      <NA>        5
26 Female Makkah                     <NA>       23
27 Female Makkah al Mukarramah       18-44     917
28 Female Makkah al Mukarramah       other    2224
29 Female Najran                     18-44     372
30 Female Najran                     other     821
31 Female Tabuk                      18-44     376
32 Female Tabuk                      other     876
33 Female `Asir                      <NA>       14
34 Male   'Asir                      18-44     967
```

```
35 Male   'Asir                     other    2293
36 Male   Al Bahah                  18-44     539
37 Male   Al Bahah                  other    1228
38 Male   Al Hudud ash Shamaliyah   18-44     216
39 Male   Al Hudud ash Shamaliyah   other     520
40 Male   Al Jawf                   18-44     216
41 Male   Al Jawf                   other     531
42 Male   Al Jawf                   <NA>        1
# i 27 more rows
```

```
## 18-44 F
smok_18_44 <- smok_pops_region |>
    filter(Gender == "Female", `18-44` == "18-44") |>
    group_by(ADM1_EN) |>
    reframe(n = n(),
            sum_pop = sum(Population),
            rate_100k = 100000 * n / sum_pop)

    smok_18_44_ci <- PHEindicatormethods::phe_rate(smok_18_44, n, sum_pop, multiplier = 10000

smok_18_44_ci |>
    ggplot() +
    geom_col(aes(reorder(ADM1_EN, -rate_100k), rate_100k), fill = "goldenrod") +
    geom_point(aes(reorder(ADM1_EN, -rate_100k), rate_100k, colour = n)) +
    geom_linerange(aes(x = ADM1_EN, ymin = lowercl, ymax = uppercl)) +
    labs(y = "",
         x = "Region
         ") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

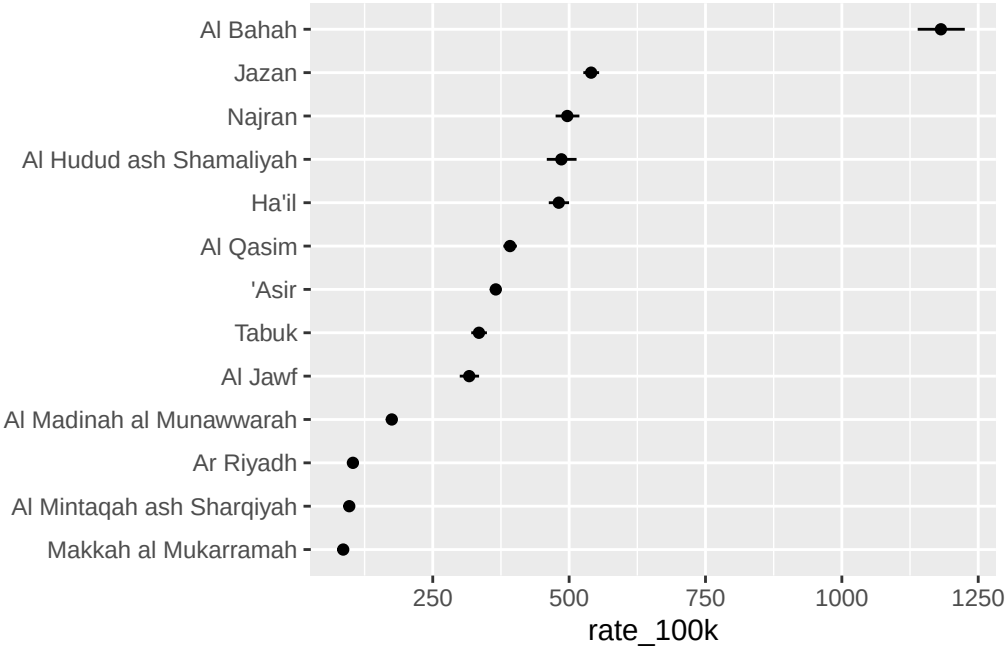



```
## 15+

smok_15_ <- smok_pops_region |>
    filter(`15+` == "15+") |>
    group_by(ADM1_EN) |>
    reframe(n = n(),
            sum_pop = sum(Population),
            rate_100k = 100000 * n / sum_pop)

smok_15_ci <- PHEindicatormethods::phe_rate(smok_15_, n, sum_pop,  multiplier = 100000)

smok_15_ci |>
    ggplot() +
    geom_point(aes(reorder(ADM1_EN, rate_100k), rate_100k)) +
    geom_linerange(aes(x = ADM1_EN, ymin = lowercl, ymax = uppercl)) +
    coord_flip() +
    labs(x = "")
```




```
## AS specific

smok_pops_region |>
    #filter(`15+` == "15+") |>
    group_by(ADM1_EN, `Five-Year Age Group`, Gender) |>
    reframe(n = n(),
            sum_pop = sum(Population),
            rate_100k = 100 * n / sum_pop) |>
   # select(-c(n, sum_pop)) |>
    pivot_wider(-c(n, rate_100k), names_from = c("Gender", "Five-Year Age Group"), values_fr
```

```
# A tibble: 20 x 38
   ADM1_EN    `Female_0-4` `Male_0-4` `Female_10-14` `Male_10-14` `Female_15-19`
   <chr>             <int>      <int>          <int>        <int>          <int>
 1 'Asir             86076      89700          89842        92719          80089
 2 Al Bahah          13905      14292          15738        16437          13866
 3 Al Hudud ~        20196      21493          17737        18266          14648
 4 Al Jawf           35470      36359          29566        30401          23021
 5 Al Madinah           NA         NA             NA           NA             NA
 6 Al Madina~        92536      95669          91346        94512          78252
 7 Al Mintaq~       200425     208376         184743       191018         149977
 8 Al Qasim          54714      56834          57093        58495          50842
 9 Al Quassim           NA         NA             NA           NA             NA
```

```
10 Ar Riyad             NA         NA             NA           NA             NA
11 Ar Riyadh        307991     320698         298933       309250         254888
12 Ash Sharq~           NA         NA             NA           NA             NA
13 Ha'il             32737      33782          32674        33511          27449
14 Jazan             64626      67613          64993        68873          59475
15 Jizan                NA         NA             NA           NA             NA
16 Makkah               NA         NA             NA           NA             NA
17 Makkah al~       281082     292376         299840       314392         270382
18 Najran            31863      33038          27125        28865          22482
19 Tabuk             42296      44012          39399        40646          34513
20 `Asir                NA         NA             NA           NA             NA
# i 32 more variables: `Male_15-19` <int>, `Female_20-24` <int>,
#   `Male_20-24` <int>, `Female_25-29` <int>, `Male_25-29` <int>,
#   `Female_30-34` <int>, `Male_30-34` <int>, `Female_35-39` <int>,
#   `Male_35-39` <int>, `Female_40-44` <int>, `Male_40-44` <int>,
#   `Female_45-49` <int>, `Male_45-49` <int>, `Female_5-9` <int>,
#   `Male_5-9` <int>, `Female_50-54` <int>, `Male_50-54` <int>,
#   `Female_55-59` <int>, `Male_55-59` <int>, `Female_60-64` <int>, ...
```

4 AMR walkthrough

5 Introduction

This document outlines a stepwise approach to calculating AMR indicators from dummy data kindly supplied by PHA.

There are x steps

1. EDA (exploratory data analysis of raw data) - this involves cleaning, visualisation and creation of relevant variables.
2. Review of indicator definitions
    - Numerator
    - Denominator
3. Method for calculating numerator and denominator values from dataset. The outline uses R code for reproducibility and flexibility.
4. Calculating indicator values and uncertainty intervals
5. Suggested indicator visualisations (if appropriate).

6 AMR indicators

6.0.1 MRSA

Percentage of bloodstream infection due to methicillin-resistant Staphylococcus aureus (MRSA)

Numerator: No. of patients with growth of methicillin-resistant S. aureus in tested blood samples

Denominator: Total No. of patients with growth of S. aureus in tested blood samples

6.0.2 E. coli

Percentage of bloodstream infection due to 3rd-generation cephalosporin resistant E. coli

Numerator: No. of patients with growth of 3rd-generation cephalosporin resistant E. coli in tested blood samples

Denominator: Total No. of patients with growth of E. coli in tested blood samples

6.0.3 Import data

```
df <- amr
```

334 observations

6.1 Data preparation

6.1.1 calculate 5-year age bands

```r
amr <- amr[, `:=` (five_year = cut(age_year, breaks = seq(0, 100, 5), right = FALSE))][]

head(amr)
```

```
   record_number sample_no patient_mrn   location
           <num>    <char>      <char>     <char>
1:             1    ######       ##### Outpatient
2:            17    ######       #####  Inpatient
3:            20    ######       #####  Inpatient
4:            25    ######       #####  Inpatient
5:            43    ######       ##### Outpatient
6:            63    ######       ##### Outpatient
                                                  patient_hospitalized
                                                                <char>
1: Patient had NOT been admitted for more than 2 days in the past 30 days
2:                      Patient has been hospitalized for 2 days or less
3:                    Patient has been hospitalized for more than 2 days
4:                      Patient has been hospitalized for 2 days or less
5: Patient had NOT been admitted for more than 2 days in the past 30 days
6: Patient had NOT been admitted for more than 2 days in the past 30 days
     specific_location age_year community_origin   site first_name second_name
                 <char>    <num>           <char> <char>     <char>      <char>
1:      Emergency Room        0 Community Origin  Blood       ####       #####
2: Intensive Care Unit       71 Community Origin  Blood       ####       #####
3: Intensive Care Unit       44  Hospital Origin  Blood       ####       #####
4: Intensive Care Unit       67 Community Origin  Blood       ####       #####
5:      Emergency Room       67 Community Origin  Blood       ####       #####
6:      Emergency Room       92 Community Origin  Blood       ####       #####
   family_name national_iqama_id nationality    pathogen_name minocycline
        <char>            <char>      <char>           <char>      <lgcl>
1:        ####        ##########       ##### Escherichia coli          NA
2:        ####        ##########       ##### Escherichia coli          NA
3:        ####        ##########       ##### Escherichia coli          NA
4:        ####        ##########       ##### Escherichia coli          NA
5:        ####        ##########       ##### Escherichia coli          NA
6:        ####        ##########       ##### Escherichia coli          NA
   tigecycline ampicillin penicillin_g oxacillin cefoxitin cefotaxime
        <lgcl>     <char>       <lgcl>    <char>    <char>     <char>
1:          NA          R           NA      <NA>      <NA>          R
2:          NA          R           NA      <NA>      <NA>         NA
3:          NA          S           NA      <NA>      <NA>          S
```

```
4:          NA          R           NA      <NA>      <NA>          R
5:          NA          R           NA      <NA>      <NA>         NA
6:          NA          R           NA      <NA>      <NA>         NA
   ceftazidime ceftriaxone cefixime cefepime doripenem ertapenem imipenem
        <char>      <char>   <lgcl>   <char>    <char>    <char>   <char>
1:           R           R       NA        R        NA         S        S
2:           S           S       NA        S        NA         S        S
3:           S           S       NA        S        NA         S        S
4:           R           R       NA        R        NA         S        S
5:           I           S       NA        S        NA         S        S
6:           R           R       NA        R         R         S        S
   meropenem co_trimoxazole azithromycin amikacin gentamicin ciprofloxacin
      <char>         <char>       <lgcl>   <lgcl>     <lgcl>        <char>
1:         S              S           NA       NA         NA             S
2:         S              S           NA       NA         NA             S
3:         S              S           NA       NA         NA             S
4:         S              R           NA       NA         NA             S
5:         S              S           NA       NA         NA             S
6:         S              R           NA       NA         NA             R
   levofloxacin colistin spectinomycin five_year
         <char>   <char>        <lgcl>    <fctr>
1:            S       NA            NA     [0,5)
2:            S        S            NA   [70,75)
3:            S       NA            NA   [40,45)
4:            S       NA            NA   [65,70)
5:            S        S            NA   [65,70)
6:            R        S            NA   [90,95)
```

6.1.2 remove non-relevant data

This step removes identifiers (names, record IDs)

```
amr <- amr |> select(-c(family_name, first_name, sample_no, patient_mrn, second_name, nationa
```

6.1.3 create per test file (long data)

- this create a *per test* dataset rather than a per patient sample dataset

```
amr_long <- amr |>
    pivot_longer(names_to = "antibiotic_test", values_to = "resistance", cols = minocycline:s
```

6.1.4 recode 3rd generation cephalosporins

- this step adds a new variable which labels 3rd generation cephalosporins

```
amr_long <- amr_long[, gen_3 := case_when(str_detect(antibiotic_test, "cef") ~ "3rd-gen", TRU
```

6.2 Data summarisation and description (EDA)

- first generate a high level tabular summary

```
gtsummary::tbl_summary(amr)
```

- represent this visually - we'll use decompostion trees

```
amr_freq <- amr_long[pathogen_name == "Escherichia coli", .N, by = .(five_year, gen_3, resist

collapsibleTreeSummary(amr_freq,
                       c("gen_3", "resistance"),
                       root = "E. coli",
                       nodeSize = "N",
                       attribute = "N",
                       fontSize = 16,
                       collapsed = FALSE)
```

6.3 Numerators and denominators

- to calculate indicators we need to calculate
- patients with blood stream infection
- samples with antibiotic resistance

```
amr_long
```

```
    record_number   location
            <num>     <char>
 1:             1 Outpatient
 2:             1 Outpatient
 3:             1 Outpatient
 4:             1 Outpatient
 5:             1 Outpatient
```

```
  ---
7678:          1210  Inpatient
7679:          1210  Inpatient
7680:          1210  Inpatient
7681:          1210  Inpatient
7682:          1210  Inpatient
                                                          patient_hospitalized
                                                                        <char>
   1: Patient had NOT been admitted for more than 2 days in the past 30 days
   2: Patient had NOT been admitted for more than 2 days in the past 30 days
   3: Patient had NOT been admitted for more than 2 days in the past 30 days
   4: Patient had NOT been admitted for more than 2 days in the past 30 days
   5: Patient had NOT been admitted for more than 2 days in the past 30 days
  ---
7678:                     Patient has been hospitalized for more than 2 days
7679:                     Patient has been hospitalized for more than 2 days
7680:                     Patient has been hospitalized for more than 2 days
7681:                     Patient has been hospitalized for more than 2 days
7682:                     Patient has been hospitalized for more than 2 days
       specific_location age_year community_origin   site         pathogen_name
                  <char>    <num>           <char> <char>                <char>
   1:     Emergency Room        0 Community Origin  Blood      Escherichia coli
   2:     Emergency Room        0 Community Origin  Blood      Escherichia coli
   3:     Emergency Room        0 Community Origin  Blood      Escherichia coli
   4:     Emergency Room        0 Community Origin  Blood      Escherichia coli
   5:     Emergency Room        0 Community Origin  Blood      Escherichia coli
  ---
7678: Non Intensive Unit       96  Hospital Origin  Blood Staphylococcus aureus
7679: Non Intensive Unit       96  Hospital Origin  Blood Staphylococcus aureus
7680: Non Intensive Unit       96  Hospital Origin  Blood Staphylococcus aureus
7681: Non Intensive Unit       96  Hospital Origin  Blood Staphylococcus aureus
7682: Non Intensive Unit       96  Hospital Origin  Blood Staphylococcus aureus
      five_year antibiotic_test resistance  gen_3
         <fctr>          <char>     <char> <char>
   1:     [0,5)     minocycline       <NA>  other
   2:     [0,5)     tigecycline       <NA>  other
   3:     [0,5)      ampicillin          R  other
   4:     [0,5)    penicillin_g       <NA>  other
   5:     [0,5)       oxacillin       <NA>  other
  ---
7678:  [95,100)      gentamicin       <NA>  other
7679:  [95,100)   ciprofloxacin          R  other
7680:  [95,100)    levofloxacin          R  other
```

```
7681:  [95,100)        colistin         NA  other
7682:  [95,100)   spectinomycin       <NA>  other
```

6.4 Calculate resistance rates

- calculate proportion of tests resistant
- calculate confidence interval (using Wilsons score method for proportions via the `PHEindicatormethods` R package)

```
amr_long[pathogen_name == "Staphylococcus aureus" & !is.na(resistance), .N, by = .(resistanc
    pivot_wider(names_from = resistance, values_from = N) |>
    rowwise() |>
    mutate(total_tests = sum(c_across(S:I), na.rm = TRUE),
           resistance_rate = R / total_tests)
```

```
# A tibble: 1 x 6
# Rowwise:
      S  `NA`     R     I total_tests resistance_rate
  <int> <int> <int> <int>       <int>           <dbl>
1   947   281   545    11        1784           0.305
```

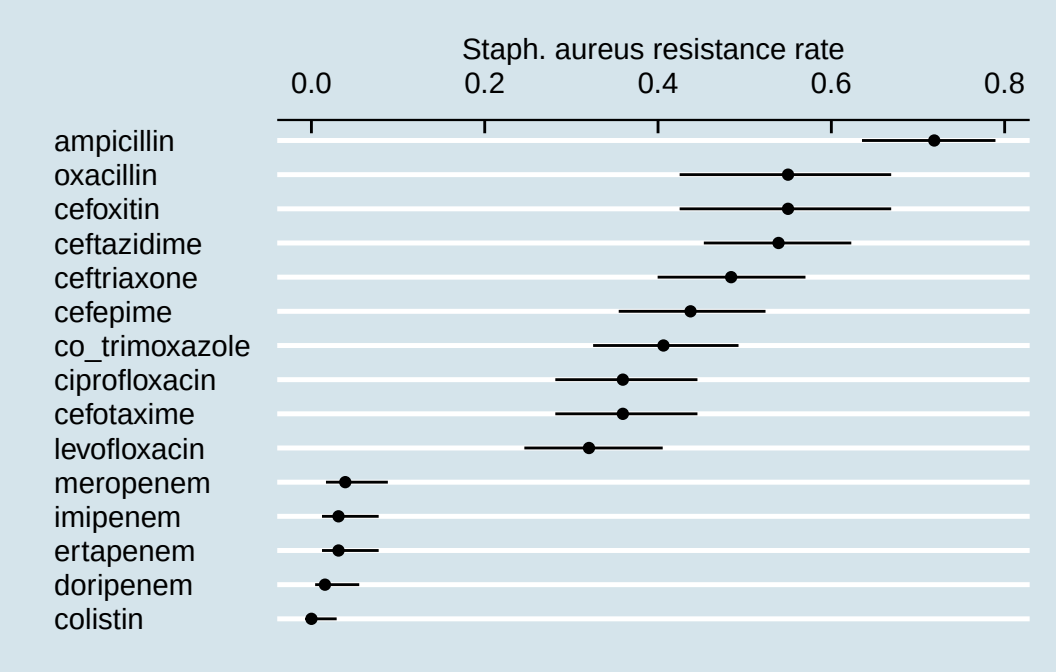
by antibiotic

```
options(digits = 2)

amr_res_ci_sa <- amr_long[pathogen_name == "Staphylococcus aureus" & !is.na(resistance), .N,
    pivot_wider(names_from = resistance, values_from = N, values_fill = 0) |>
    rowwise() |>
    mutate(total_tests = sum(c_across(S:I), na.rm = TRUE),
           resistance_rate = R / total_tests)

phe_proportion(amr_res_ci_sa, R, total_tests) |>
    bind_cols(amr_res_ci_sa) |>
    ggplot() +
    geom_point(aes(reorder(antibiotic_test, value), value)) +
    geom_linerange(aes(antibiotic_test, ymin = lowercl, ymax = uppercl)) +
    coord_flip() +
    labs(y = "Staph. aureus resistance rate", x = "") +
    scale_y_continuous(position = "right")
```

```
New names:
* `R` -> `R...1`
* `total_tests` -> `total_tests...2`
* `R` -> `R...12`
* `total_tests` -> `total_tests...14`
```

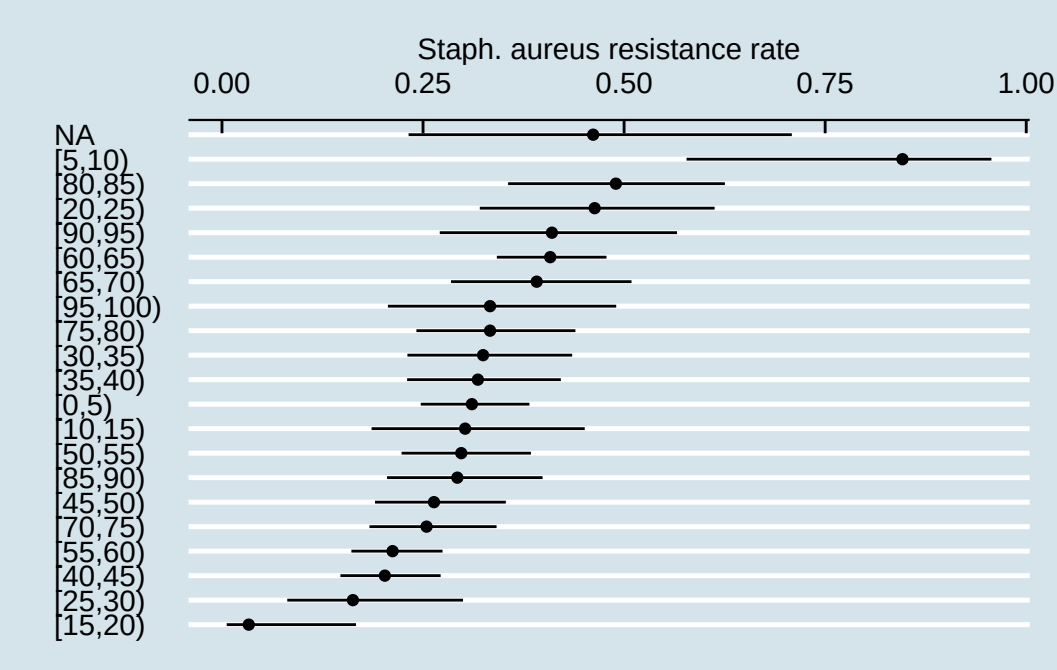


by age

```
amr_res_ci_age <- amr_long[pathogen_name == "Staphylococcus aureus" & !is.na(resistance), .N
    pivot_wider(names_from = resistance, values_from = N, values_fill = 0) |>
    rowwise() |>
    mutate(total_tests = sum(c_across(S:I), na.rm = TRUE),
           resistance_rate = R / total_tests)

phe_proportion(amr_res_ci_age, R, total_tests) |>
    bind_cols(amr_res_ci_age) |>
    ggplot() +
    geom_point(aes(reorder(five_year, value), value)) +
    geom_linerange(aes(five_year, ymin = lowercl, ymax = uppercl)) +
    coord_flip() +
    labs(y = "Staph. aureus resistance rate", x = "") +
    scale_y_continuous(position = "right")
```

```
New names:
* `R` -> `R...1`
* `total_tests` -> `total_tests...2`
* `R` -> `R...12`
* `total_tests` -> `total_tests...14`
```



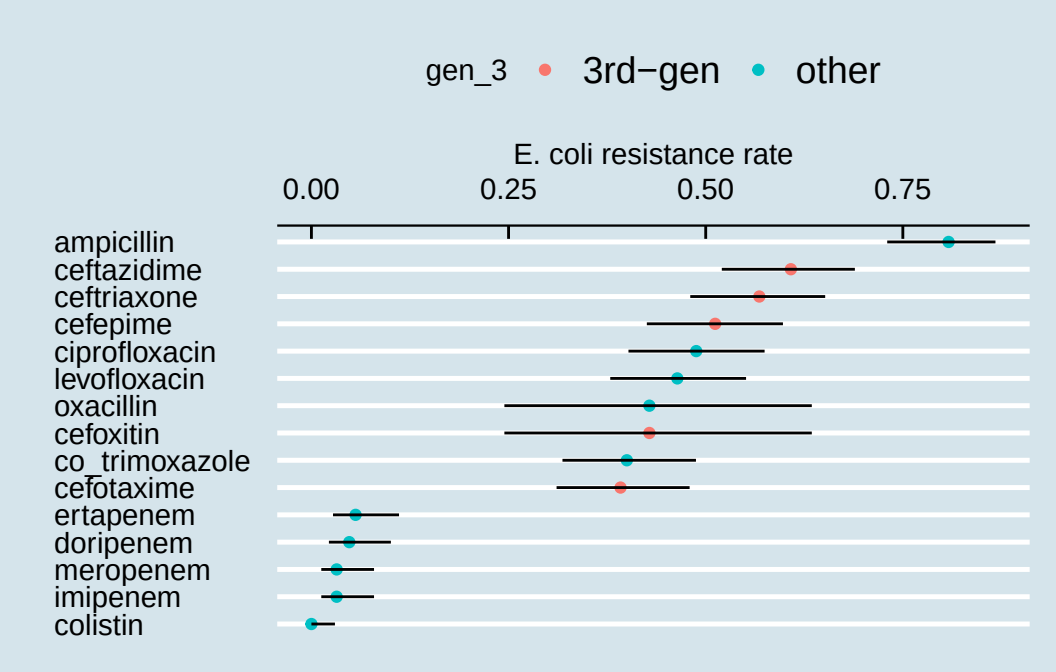
6.5 E. coli

```
amr_res_ci_ec <- amr_long[str_detect(pathogen_name, "coli") & !is.na(resistance), .N, by = .
    pivot_wider(names_from = resistance, values_from = N, values_fill = 0) |>
    rowwise() |>
    mutate(total_tests = sum(c_across(R:I), na.rm = TRUE),
           resistance_rate = R / total_tests)

phe_proportion(amr_res_ci_ec, R, total_tests) |>
    bind_cols(amr_res_ci_ec) |>
    ggplot() +
    geom_point(aes(reorder(antibiotic_test, value), value, colour = gen_3)) +
    geom_linerange(aes(antibiotic_test, ymin = lowercl, ymax = uppercl)) +
```

```
    coord_flip() +
    labs(y = "E. coli resistance rate", x = "") + scale_y_continuous(position = "right")
```

```
New names:
* `R` -> `R...1`
* `total_tests` -> `total_tests...2`
* `R` -> `R...11`
* `total_tests` -> `total_tests...15`
```



6.5.0.1 select methicillin tested samples

```
amr[, .SD, .SDcols = c("record_number", "location", "patient_hospitalized", "specific_locati
```

```
     record_number   location
             <num>     <char>
  1:             1 Outpatient
  2:            17  Inpatient
  3:            20  Inpatient
  4:            25  Inpatient
```

```
  5:            43 Outpatient
 ---
330:          1218  Inpatient
331:          1159 Outpatient
332:          1183 Outpatient
333:          1160 Outpatient
334:          1210  Inpatient
                                                          patient_hospitalized
                                                                       <char>
  1: Patient had NOT been admitted for more than 2 days in the past 30 days
  2:                        Patient has been hospitalized for 2 days or less
  3:                      Patient has been hospitalized for more than 2 days
  4:                        Patient has been hospitalized for 2 days or less
  5: Patient had NOT been admitted for more than 2 days in the past 30 days
 ---
330:                        Patient has been hospitalized for 2 days or less
331: Patient had NOT been admitted for more than 2 days in the past 30 days
332: Patient had NOT been admitted for more than 2 days in the past 30 days
333: Patient had NOT been admitted for more than 2 days in the past 30 days
334:                      Patient has been hospitalized for more than 2 days
          specific_location community_origin   site         pathogen_name
                      <char>           <char> <char>                <char>
  1:         Emergency Room Community Origin  Blood      Escherichia coli
  2:    Intensive Care Unit Community Origin  Blood      Escherichia coli
  3:    Intensive Care Unit  Hospital Origin  Blood      Escherichia coli
  4:    Intensive Care Unit Community Origin  Blood      Escherichia coli
  5:         Emergency Room Community Origin  Blood      Escherichia coli
 ---
330:     Non Intensive Unit Community Origin  Blood      Escherichia coli
331:         Emergency Room Community Origin  Blood Staphylococcus aureus
332: Out Patient Department Community Origin  Blood Staphylococcus aureus
333:         Emergency Room Community Origin  Blood      Escherichia coli
334:     Non Intensive Unit  Hospital Origin  Blood Staphylococcus aureus
     five_year
        <fctr>
  1:     [0,5)
  2:   [70,75)
  3:   [40,45)
  4:   [65,70)
  5:   [65,70)
 ---
330:   [80,85)
331:   [85,90)
```

```
332:   [85,90)
333:   [90,95)
334:  [95,100)
```

7 smoking-poc

7.1

7.1.1 Data preparation

```
smoking <- smoking[, `:=` (five_year = cut(age, breaks = seq(0, 100, 5), right = FALSE), `18-
```

7.1.2 Calculate numerator and denominator

```
smoking[patient_gender == "female" & `18-44` == TRUE, .N, by = .(directorate_name, year)]
```

```
    directorate_name  year     N
              <char> <int> <int>
 1:             Asir  2022     9
 2:            Jazan  2022   145
 3:             Jouf  2022    23
 4:           Najran  2022    82
 5:           Qassim  2022    22
 6:          Eastern  2022   551
 7:           AlAhsa  2022    73
 8:            Tabuk  2022   291
 9: Northern Borders  2022     4
10:          Madinah  2022   205
11:           Riyadh  2022  1830
12:           Jeddah  2022  1569
13:             Baha  2022    28
14:             Taif  2022   100
15:           Makkah  2022   578
16:         Qunfotha  2022    13
17:            Bisha  2022     6
18:             Hail  2022    22
19:    Hafer AlBatin  2022    60
```

```
## probably better - easier to calculate / matching age bands/ more statitsical power NB cru

smoking[patient_gender == "female" & `15+` == TRUE, .N, by = .(directorate_name, year)]
```

```
    directorate_name  year     N
               <char> <int> <int>
 1:             Asir  2022    84
 2:            Jazan  2022   158
 3:             Jouf  2022    28
 4:           Najran  2022    85
 5:           Qassim  2022    48
 6:          Eastern  2022   579
 7:           AlAhsa  2022    93
 8:            Tabuk  2022   436
 9: Northern Borders  2022     7
10:          Madinah  2022   303
11:           Riyadh  2022  1862
12:           Jeddah  2022  1788
13:             Baha  2022    28
14:             Taif  2022   136
15:           Makkah  2022   636
16:         Qunfotha  2022    13
17:            Bisha  2022     6
18:             Hail  2022    22
19:    Hafer AlBatin  2022    60
20:         Qurayyat  2022     4
    directorate_name  year     N
```

```
## denominator
```

7.1.3 Choropleth map

```
sa_shp <- curl_download("https://data.humdata.org/dataset/41ce9023-1d21-4549-a485-94316200aba

tmpd <- tempdir()

sa_shp_1 <- curl_download("https://data.humdata.org/dataset/41ce9023-1d21-4549-a485-9431620

#sa_pop_d <- curl_download("https://data.humdata.org/dataset/14b288ca-1855-4025-9f01-41cba548
```

```r
sa_shp <- unzip(sa_shp_l, exdir = tmpd)

sa_shp_1 <- unzip(sa_shp_1, exdir = tmpd)

sa_tif <- unzip(sa_pop_d, exdir = tmpd)

shps <- fs::dir_ls(tmpd, regexp = "shp")

sa_bound <- read_sf(shps[4])

sa_bound |>
    ggplot() +
    geom_sf(aes(fill = ADM1_EN)) +
    geom_sf(data = read_sf(shps[12])) +
    geom_sf_label(data = read_sf(shps[12]), aes(label = NAME), colour = "blue", size = 3, nu
    theme_void() +
    scale_fill_viridis_d(option = "rocket")

smoking$directorate_name |>
    unique()

sa_bound$ADM1_EN


# source("/Users/julianflowers/Library/CloudStorage/GoogleDrive-julian.flowers12@gmail.com/M
```

8 flu

Injury

```
flu$region_en |> unique()
```

```
 [1] "Riyadh"               "Sharqiya"             "Makkah Al Mukarramah"
 [4] "Asir"                 "madina"               "Tabuk"
 [7] "Jazan"                "Najran"               "Al Qassim"
[10] "Hail"                 "Al Baha"              "Northern Frontier"
[13] "Al Jawf"
```

```
flu[, .N, by = .(Gender, AgeAtAdministration, region_en)]
```

```
      Gender AgeAtAdministration region_en     N
      <char>                <int>    <char> <int>
   1:      M                   23    Riyadh    18
   2:      F                   23    Riyadh    36
   3:      F                   33    Riyadh    93
   4:      M                   33  Sharqiya    79
   5:      M                   33    Riyadh    30
  ---
1019:      M                    6     Jazan     2
1020:      M                   54 Al Qassim     5
1021:      F                   54    Najran     2
1022:      M                   54      Asir     5
1023:      F                   54 Al Qassim     1
```

```
length(flu$region_en |> unique())
```

```
[1] 13
```

```
length(flu$AgeAtAdministration |> unique())
```

```
[1] 89
```

```
max(flu$AgeAtAdministration)
```

```
[1] 118
```

```
flu_reg_names <- pluck(flu, "region_en") |> unique()
pops_reg_names <- pluck(pops, "Region") |> unique()

intersect(flu_reg_names, pops_reg_names)
```

```
[1] "Tabuk"   "Jazan"   "Najran"  "Al Jawf"
```

```
## only 4 names are identical between datasets
## will need to recode region names in flu dataset to pop data names
## also add new variable `region` to facilitate linkage between datasets

flu <- flu[, region := recode(region_en, "Riyadh" = "Ar Riyadh",
                                  "Al Baha" = "Al Bahah",
                                  "Sharqiya"  = "Al Mintaqah ash Sharqiyah",
                                  "Makkah Al Mukarramah" = "Makkah al Mukarramah",
                                  "Al Qassim" = "Al Qasim",
                                  "Hail" = "Ha'il",
                                  "madina" = "Al Madinah al Munawwarah",
                                  "Asir" = "'Asir",
                                  "Northern Frontier" = "Al Hudud ash Shamaliyah")]


## check names match
intersect(unique(flu$region), pops_reg_names)
```

```
 [1] "Ar Riyadh"                 "Al Mintaqah ash Sharqiyah"
 [3] "Makkah al Mukarramah"      "'Asir"
 [5] "Al Madinah al Munawwarah"  "Tabuk"
 [7] "Jazan"                     "Najran"
 [9] "Al Qasim"                  "Ha'il"
[11] "Al Bahah"                  "Al Hudud ash Shamaliyah"
[13] "Al Jawf"
```

```
labels <- unique(pops$`Five-Year Age Group`)

#cut(flu$AgeAtAdministration, breaks = seq(0, max(flu$AgeAtAdministration), 5))
##  first create a terminal age band 80+ to match population data
##
flu <- flu[!is.na(AgeAtAdministration), age := ifelse(AgeAtAdministration >= 80, 85, AgeAtAdm

cut(flu$age, breaks = seq(0, 85, 5)) |> unique()
```

```
 [1] (20,25] (30,35] (35,40] (0,5]   <NA>    (40,45] (25,30] (60,65] (55,60]
[10] (50,55] (65,70] (5,10]  (15,20] (45,50] (10,15] (75,80] (70,75] (80,85]
17 Levels: (0,5] (5,10] (10,15] (15,20] (20,25] (25,30] (30,35] ... (80,85]
```

```
length(pops$`Five-Year Age Group` |> unique())
```

```
[1] 17
```

```
flu <- flu[, age_band := cut(age, breaks = seq(0, 85, 5), labels = labels,  right = TRUE)][]

## count vaccinations by age, gender and region
flu_freq <- flu[, .N, by = .(Gender, region, age_band)][order(region, age_band, Gender)][, ag

flu_freq[]
```

```
     Gender region age_band     N
     <char> <char>   <char> <int>
  1:      F  'Asir      0-4    25
  2:      M  'Asir      0-4     1
  3:      F  'Asir      5-9     7
  4:      M  'Asir      5-9     1
  5:      F  'Asir    10-14     4
 ---
319:      M  Tabuk    65-69     1
320:      F  Tabuk    75-79     1
321:      M  Tabuk      80+     1
322:      F  Tabuk     <NA>    15
323:      M  Tabuk     <NA>    34
```

```
## first remove NAs

flu_freq <- flu_freq[!(is.na(age_band)),]
flu_freq <- flu_freq[, Gender := recode(Gender, "M" = "Male", "F" = "Female")][]

## check age bands match

identical(flu_freq$age_band |> unique(), pops$`Five-Year Age Group` |> unique())
```

```
[1] TRUE
```

```
## join population and aggregated flu data

## first exclude nationality and single age columns from the pop data

pops[, `:=` (`Single Age Group` = NULL, Nationality = NULL)][]
```

```
         Region Five-Year Age Group Gender Population age_numeric    15+  18-44
         <char>               <char> <char>      <int>       <int> <char> <char>
    1: Al Bahah                 0-4 Female        577           0  other  other
    2: Al Bahah                 0-4 Female         58           0  other  other
    3: Al Bahah                 0-4 Female        115           0  other  other
    4: Al Bahah                 0-4 Female          1           0  other  other
    5: Al Bahah                 0-4 Female        364           0  other  other
   ---
54409:   Najran                 80+   Male          1         100    15+  other
54410:   Najran                 80+   Male         15         100    15+  other
54411:   Najran                 80+   Male         42         100    15+  other
54412:   Najran                 80+   Male          8         100    15+  other
54413:   Najran                 80+   Male          1         100    15+  other
```

```
## then calculate 5-year pops by age band, gender and region

pops_agg <- pops[, sum_pop := sum(Population), by = .(Region, `Five-Year Age Group`, Gender)]
    select(Region, Gender, `Five-Year Age Group`, sum_pop)

pops_agg$`Five-Year Age Group` |> unique()
```

```
 [1] "0-4"   "5-9"   "10-14" "15-19" "20-24" "25-29" "30-34" "35-39" "40-44"
[10] "45-49" "50-54" "55-59" "60-64" "65-69" "70-74" "75-79" "80+"
```

```
## Now join aggregate population data to aggregated flu data and replace structural zeros (m

flu_agg <- complete(flu_freq, Gender, region, age_band) |>
    inner_join(pops_agg, by = c("Gender", "region" = "Region",  "age_band" = "Five-Year Age (
    distinct() |>
    mutate(N = ifelse(is.na(N), 0, N)) |>
    setDT()
```

8.1 Check

```
which(is.na(flu_agg[, .(N, sum_pop),  by = .(age_band, Gender, region)])) ## no NAs
```

```
integer(0)
```

```
summary(flu_agg)
```

```
   Gender             region            age_band              N
 Length:442         Length:442         Length:442         Min.   :  0.00
 Class :character   Class :character   Class :character   1st Qu.:  0.00
 Mode  :character   Mode  :character   Mode  :character   Median :  3.00
                                                          Mean   : 19.09
                                                          3rd Qu.: 18.00
                                                          Max.   :253.00

   sum_pop
 Min.   :   696
 1st Qu.:  9340
 Median : 27800
 Mean   : 72795
 3rd Qu.: 73094
 Max.   :776167
```

8.2 Calculate rates

```
flu_agg[, rate := 100000 * N/sum_pop][]
```

```
     Gender region age_band     N sum_pop       rate
     <char> <char>   <char> <num>   <int>      <num>
  1: Female  'Asir      0-4    25   86076  29.044101
  2: Female  'Asir    10-14     4   89842   4.452261
  3: Female  'Asir    15-19    19   80089  23.723607
  4: Female  'Asir    20-24    87   70589 123.248665
  5: Female  'Asir    25-29    75   72715 103.142405
 ---
438:   Male  Tabuk    60-64    14    8327 168.127777
439:   Male  Tabuk    65-69     1    4546  21.997360
440:   Male  Tabuk    70-74     0    2469   0.000000
441:   Male  Tabuk    75-79     0    1467   0.000000
442:   Male  Tabuk      80+     1    1767  56.593096
```

```
## works!
```

8.3 Compare regions

Using KSA population as standard rate

To do this will use the `phe_dsr` function from the `PHEindicatormethods` package from CRAN (see DSR vignette)

```
## first load PHEindicatormethods and epitools

needs(PHEindicatormethods, epitools)

## calculate gender, age-specific populations for KSA
##

ksa_pop <- pops[, ref_pop := sum(Population), by = .(Gender, `Five-Year Age Group`)][, .(`Fi
    distinct() |>
    rename(age_band = `Five-Year Age Group`)

ksa_pop_f <- filter(ksa_pop, Gender == "Female") |> select(-Gender)

##
```

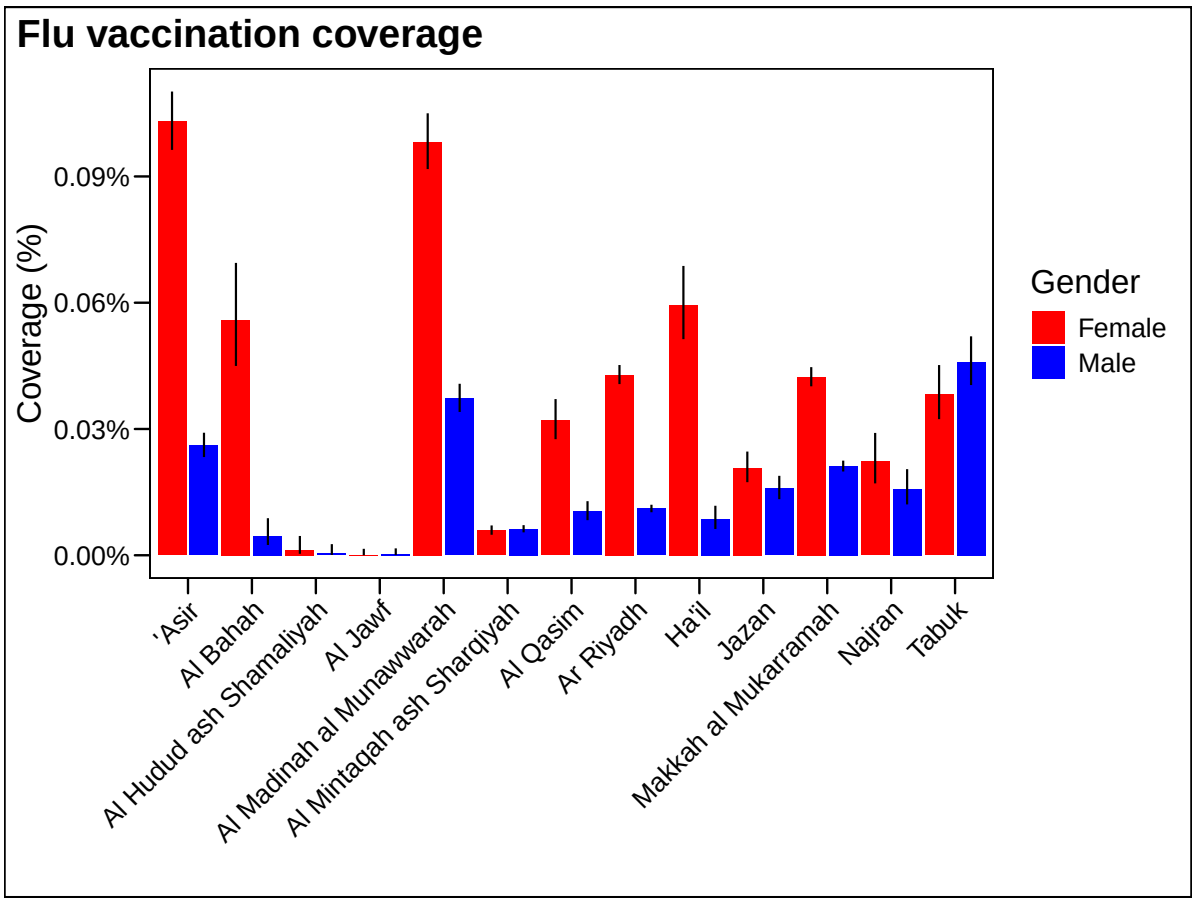
8.4 Calculate coverage

```
flu_rate <- setDT(flu_agg)[, `:=` (tot_obs = sum(N, na.rm = TRUE), tot_pop = sum(sum_pop, na
    distinct()

flu_coverage <- phe_proportion(flu_rate, x = tot_obs, n = tot_pop)
```

8.5 Visualise

```
flu_coverage |>
    ggplot() +
    geom_col(aes(region, value, fill = Gender), position = position_dodge(width = 1)) +
    geom_linerange(aes(region, ymin = lowercl, ymax = uppercl, group = Gender), position = po
    labs(title = "Flu vaccination coverage",
         y = "Coverage (%)",
         x = "") +
    ggthemes::theme_base() +
    theme(plot.title.position = "plot",
          axis.text.x = element_text(angle = 45, hjust = 1, )) +
    scale_y_continuous(label = scales::percent) +
    scale_fill_discrete(type = c("red", "blue"))
```



8.6 Age-standarised coverage

Note

```
flu_agg_std <- flu_agg |>
    left_join(ksa_pop, by = c("age_band", "Gender"))

## which region - gender combinations have data for 17 age bands?
##
##

ksa_pop_f
```

```
    age_band ref_pop
      <char>   <int>
```

```
 1:      0-4 1263917
 2:      5-9 1354766
 3:    10-14 1249029
 4:    15-19 1079884
 5:    20-24 1050547
 6:    25-29 1242388
 7:    30-34 1250860
 8:    35-39 1113283
 9:    40-44  839031
10:    45-49  592256
11:    50-54  463843
12:    55-59  351281
13:    60-64  253046
14:    65-69  156115
15:    70-74   94864
16:    75-79   64100
17:      80+   77419
```

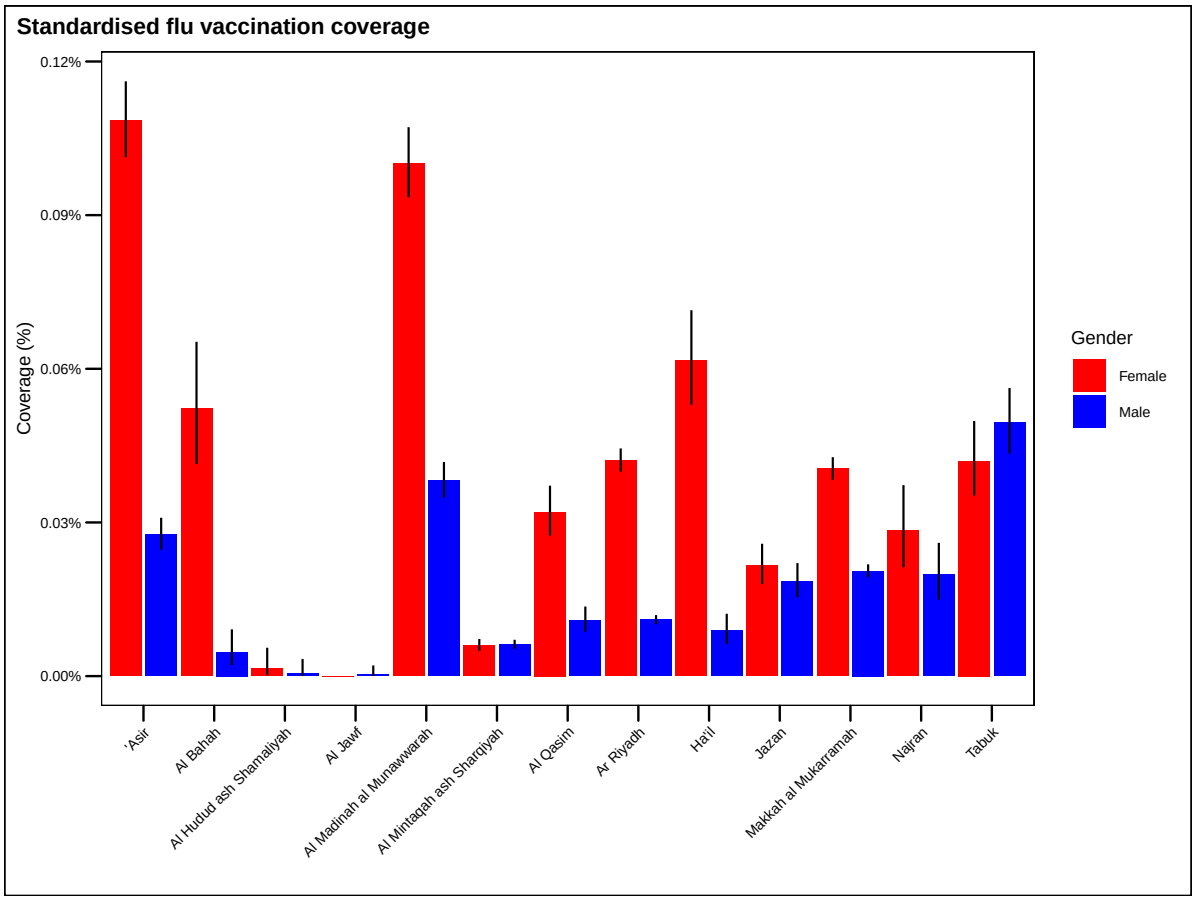
```
gp <- flu_agg_std |>
    mutate(age_band = fct_relevel(as.factor(age_band), "5-9", after = 1)) |>
    arrange(age_band)

gp_nest <- gp |>
    nest_by(region, Gender)

flu_dsrs <- gp_nest |>
    mutate(ds_rates = list(epitools::ageadjust.direct(count = data$N, pop = data$sum_pop, st
    unnest_wider(ds_rates) |>
    select(-data)


flu_dsrs |>
    ggplot() +
    geom_col(aes(region, adj.rate, fill = Gender), position = position_dodge(width = 1)) +
    geom_linerange(aes(region, ymin = lci, ymax = uci, group = Gender), position = position_
    labs(title = "Standardised flu vaccination coverage",
         y = "Coverage (%)",
         x = "") +
    theme(plot.title.position = "plot",
          axis.text.x = element_text(angle = 45, hjust = 1),
          panel.background = element_blank()) +
    scale_fill_discrete(type = c("red", "blue")) +
```

```
    scale_y_continuous(label = scales::percent)
```



9 injury

9.0.1 Injury

10 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```