Methods for Subnational Estimation of Child Mortality

Demo and Hands-on Exercises

Richard Li

Department of Biostatistics
Yale University

Overview

Overview

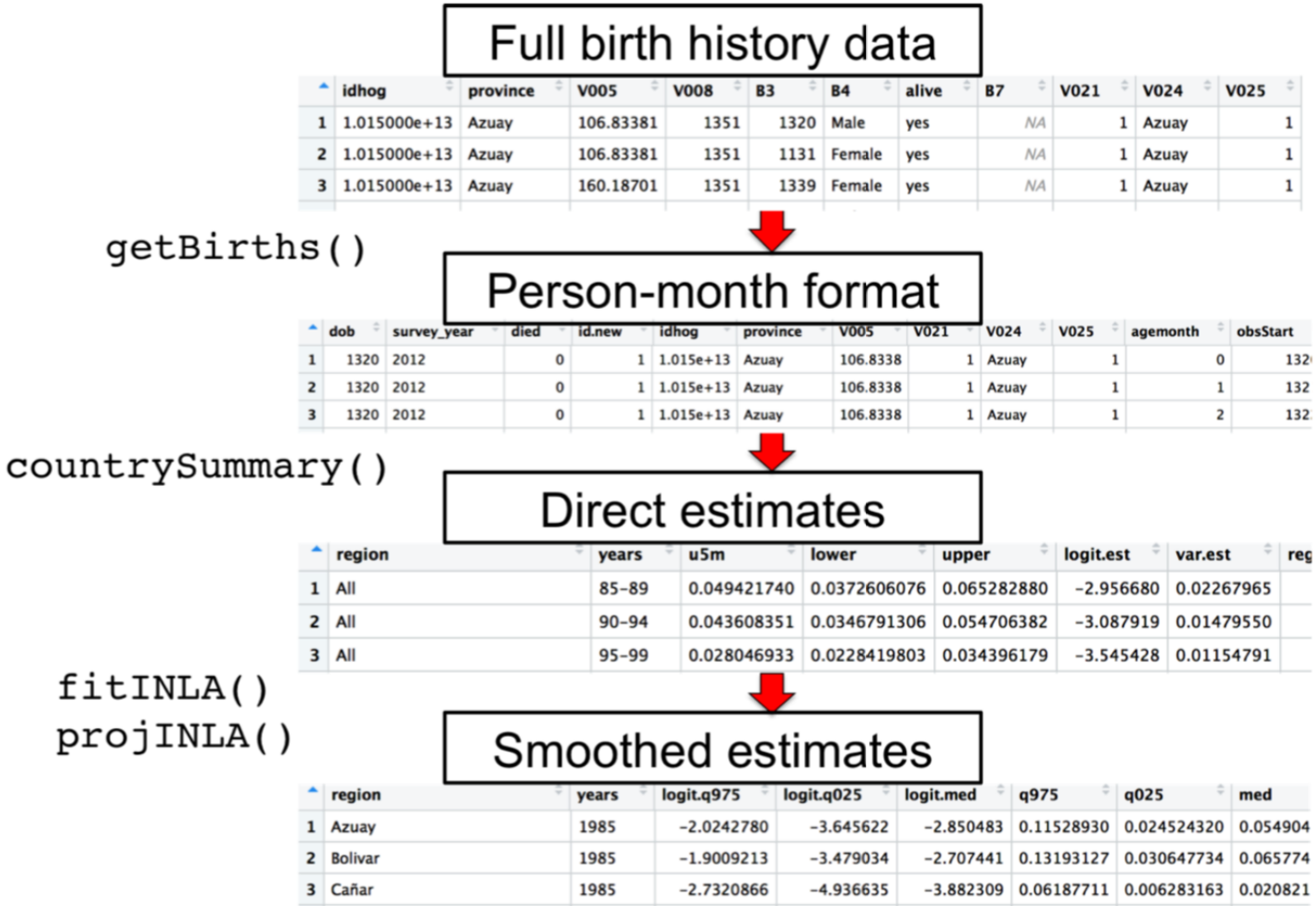
In this session, I will show you

- a full analysis of one Jordan DHS survey;
- codes and some of the output on the slides;
- code script and data available on the website.

Try to follow the analysis as we go through ☺

There is also a short exercise with a built-in small dataset for you to practice ☺

Workflow overview



	idhog	province	V005	V008	B3	B4	alive	B7	V021	V024	V025
1	1.015000e+13	Azuay	106.83381	1351	1320	Male	yes	NA	1	Azuay	1
2	1.015000e+13	Azuay	106.83381	1351	1131	Female	yes	NA	1	Azuay	1
3	1.015000e+13	Azuay	160.18701	1351	1339	Female	yes	NA	1	Azuay	1

	dob	survey_year	died	id.new	idhog	province	V005	V021	V024	V025	agemonth	obsStart
1	1320	2012	0	1	1.015e+13	Azuay	106.8338	1	Azuay	1	0	132
2	1320	2012	0	1	1.015e+13	Azuay	106.8338	1	Azuay	1	1	132
3	1320	2012	0	1	1.015e+13	Azuay	106.8338	1	Azuay	1	2	132

	region	years	u5m	lower	upper	logit.est	var.est	reg
1	All	85-89	0.049421740	0.0372606076	0.065282880	-2.956680	0.02267965	
2	All	90-94	0.043608351	0.0346791306	0.054706382	-3.087919	0.01479550	
3	All	95-99	0.028046933	0.0228419803	0.034396179	-3.545428	0.01154791	

	region	years	logit.q975	logit.q025	logit.med	q975	q025	med
1	Azuay	1985	-2.0242780	-3.645622	-2.850483	0.11528930	0.024524320	0.054904
2	Bolivar	1985	-1.9009213	-3.479034	-2.707441	0.13193127	0.030647734	0.065774
3	Cañar	1985	-2.7320866	-4.936635	-3.882309	0.06187711	0.006283163	0.020821

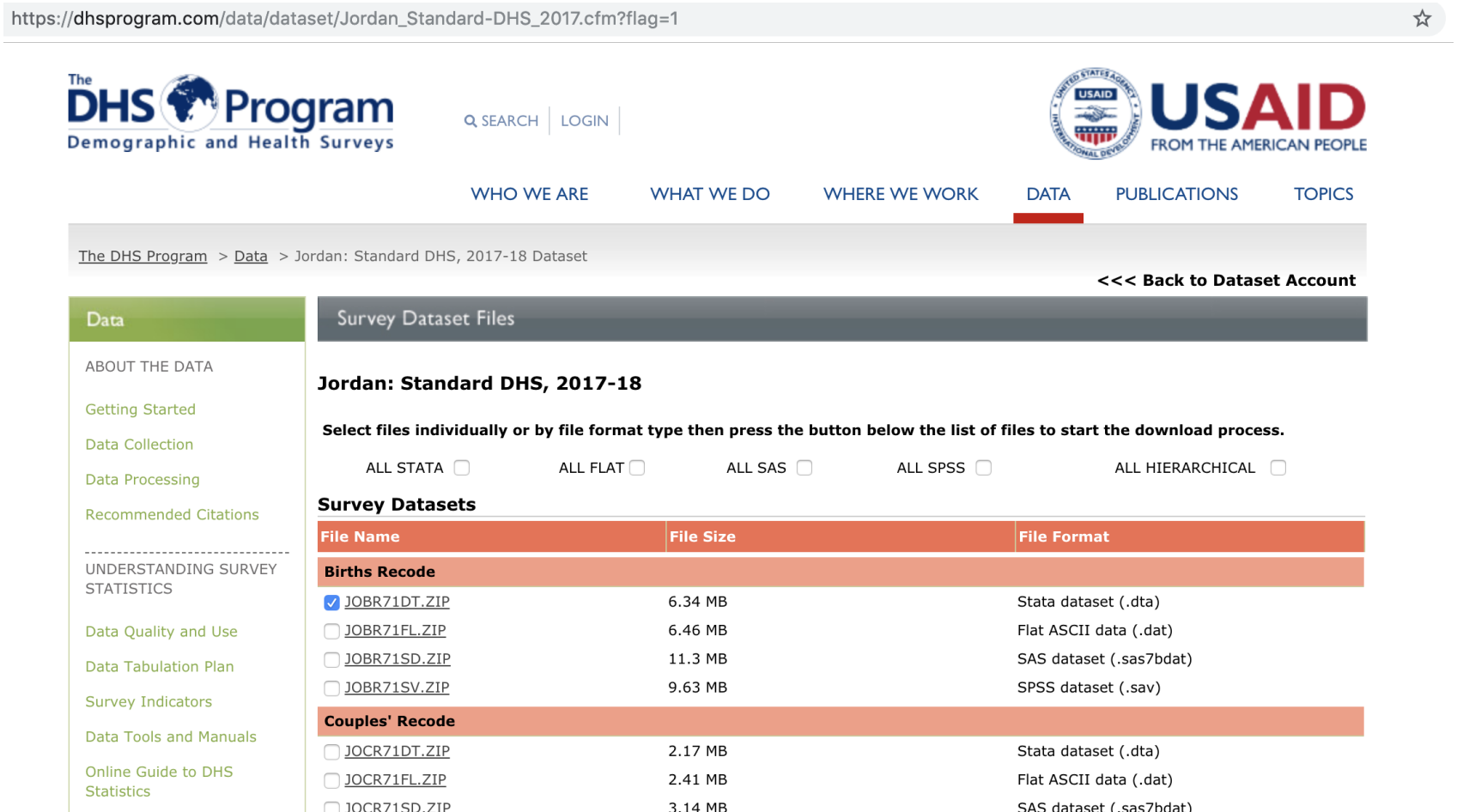
Figure 1: Workflow overview

Subnational Direct Estimates

Getting to know the data

Throughout, will use DHS data as an example.

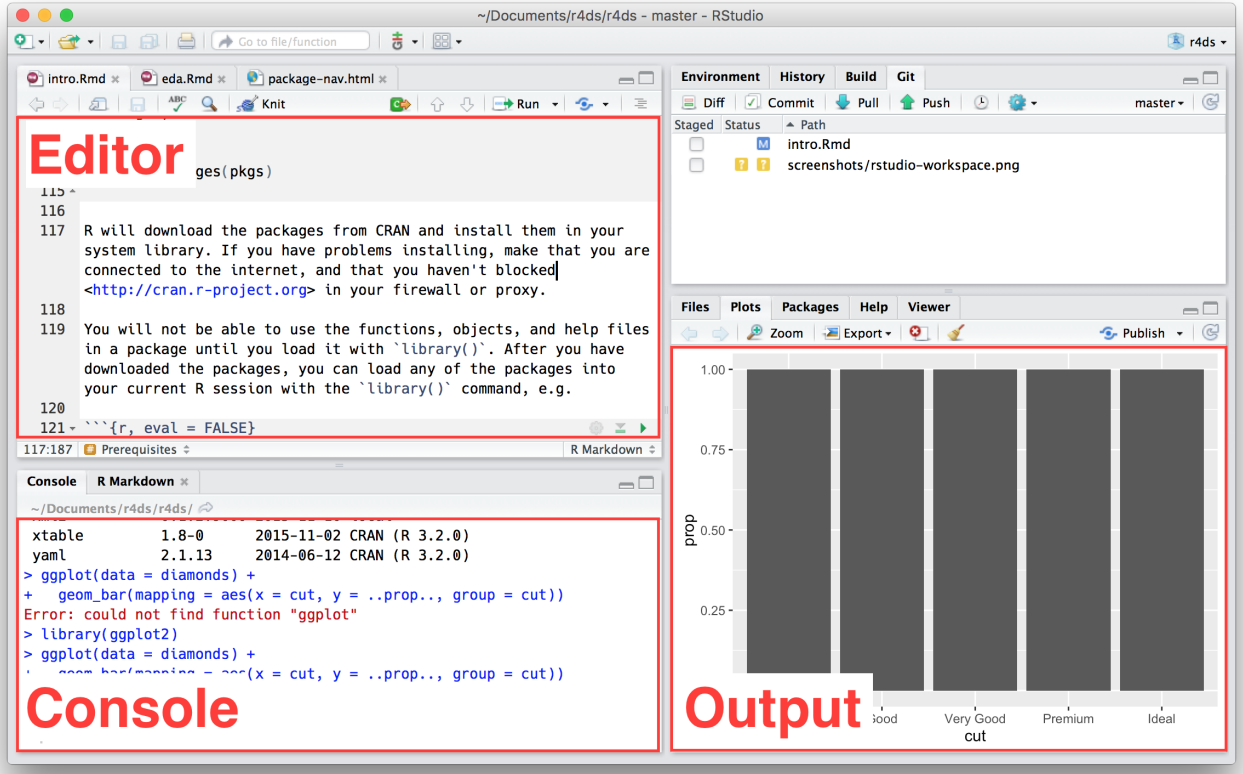
For other types of FBH data, the first few steps of data processing need to change the variable names accordingly



R and R packages

R is a free software environment for statistical computing and graphics.

RStudio is a good integrated development environment.

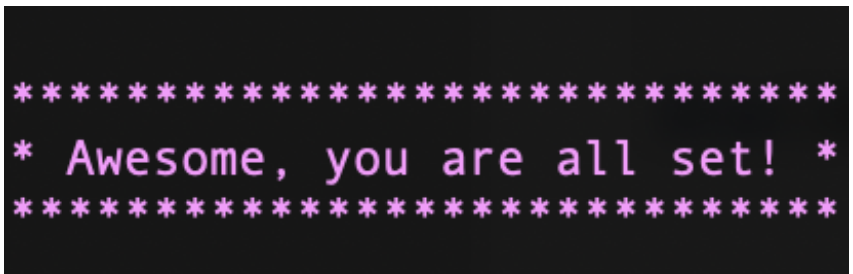


Get Started

I will show R codes and results in the demo.
All the codes, data, and map files I will use are available on the website

http://faculty.washington.edu/jonno/UNICEF-WORKSHOPS.html

To get started, download the zip file from Quick Start Kit Here.
Open Get_started.R file in RStudio. Click *Session* $\rightarrow$ *Set Working Directory* $\rightarrow$ *To Source File Location*.
Click Source button on the top right corner of the code panel. It may take a while to install all packages. At the end, you should see something like



Getting to know the data

For the first part, we will use a simulated dataset that resembles a DHS survey for Jordan. If you have access to the Jordan DHS dataset, just change the file name below:

```
filename <- "../data/JOsim.DTA"
```

We read in the .DTA (Stata format data) into R. Alternatively, for data in spreadsheets or other formats, you can read into R in similar fashions:

```
library(SUMMER)
dat <- readstata13::read.dta13(filename,
                generate.factors = TRUE)
dim(dat)
```

```
## [1] 30000   971
```

FBH data

The SUMMER package can be used to obtain direct estimates from full birth histories. The data will need to organized such that every row corresponds to a birth and columns that contain

- Indicators corresponding to survey design (e.g., strata, cluster, and household)
- Survey weight
- Date of interview in century month codes (CMC) format, i.e., the number of the month since the beginning of 1990
- Date of child's birth in CMC format
- Indicator for death of child
- Age of death of child in months

Reorganize data into person-month format

We then reformat the data into person-months

```
births <- getBirths(filename, surveyyear = 2018,
    strata = "v023")
head(births[, 1:10])
```

```
##    dob survey_year died id.new
## 1 1261        2018    0      1
## 2 1261        2018    0      1
## 3 1261        2018    0      1
## 4 1261        2018    0      1
## 5 1261        2018    0      1
## 6 1261        2018    0      1
##            caseid v001 v002 v004
## 1       438 10  2  438   10  438
## 2       438 10  2  438   10  438
```

Getting to know person-month format

```
table(births$v024, births$age)
```

```
##
##                0  1-11 12-23 24-35 36-47
##    amman    3254 31868 30885 28336 26204
##    balqa    1883 18363 17661 16217 14912
##    zarqua   2979 28840 27851 25514 23618
##    madaba   2089 20385 19559 17858 16632
##    irbid    2810 27029 26023 23862 21924
##    mafraq   3563 34416 33124 30572 28407
##    jerash   2665 25845 24698 22675 20855
##    aljoun   2609 25319 24290 22306 20551
##    karak    1941 18779 18069 16755 15379
##    tafilh   2467 23846 22681 20767 19323
##    maan     1853 17918 17301 16063 14771
##    aqouba   1875 18078 17435 15980 14855
```

Getting to know person-month format

```
table(births$v024, births$age)
```

```
##
##                0  1-11 12-23 24-35 36-47
##    amman    3254 31868 30885 28336 26204
##    balqa    1883 18363 17661 16217 14912
##    zarqua   2979 28840 27851 25514 23618
##    madaba   2089 20385 19559 17858 16632
##    irbid    2810 27029 26023 23862 21924
##    mafraq   3563 34416 33124 30572 28407
##    jerash   2665 25845 24698 22675 20855
##    aljoun   2609 25319 24290 22306 20551
##    karak    1941 18779 18069 16755 15379
##    tafilh   2467 23846 22681 20767 19323
##    maan     1853 17918 17301 16063 14771
##    aqouba   1875 18078 17435 15980 14855
```

What's going on in the getBirths() function

```
births <- getBirths(filename,  surveyyear=2018,
              variables = c("caseid", "v001", "v002",
                  "v004", "v005", "v021", "v022", "v023",
                  "v024", "v025", "v139", "bidx"),
              strata=c("v024", "v025"),
              dob = "b3",
              alive = "b5",
              age = "b7",
              date.interview= "v008",
              month.cut = c(1, 12, 24, 36, 48, 60),
              year.cut = seq(1980, 2020, by = 5))
```

The dob, age, date.interview are in CMC format.

Direct estimates

Now we are ready to calculate the direct estimates for each region and time period.

```
years <- levels(births$time)
print(years)
```

```
## [1] "80-84" "85-89" "90-94" "95-99"
## [5] "00-04" "05-09" "10-14" "15-19"
```

```
years <- c(years[-1], "20-24")
u5m <- countrySummary(births = births, years = years,
    regionVar = "v024", timeVar = "time",
    clusterVar = "~v002 + v001", ageVar = "age",
    weightsVar = "v005", geo.recode = NULL)
```

Similarly, we can extend this type of calculation to other mortality rates as well.

Direct estimates within subpopulation

```
dat <- readstata13::read.dta13(filename,
    generate.factors = TRUE)
table(dat$b4)
dat_male <- dat[dat$b4 == "male", ]
births_male <- getBirths(data = dat_male,
    surveyyear = 2018)
u5m_male <- countrySummary(births = births_male,
    years = years, regionVar = "v024", timeVar = "time",
    clusterVar = "~v002 + v001", ageVar = "age",
    weightsVar = "v005", geo.recode = NULL)
```

Infant mortality

We can change the discrete survival model into having only two bins: $[0, 1)$ and $[1, 12)$.

```
births_infant <- getBirths(data = dat, surveyyear = 2018,
    month.cut = c(1, 12))
u1m <- countrySummary(births = births_infant,
    years = years, regionVar = "v024", timeVar = "time",
    clusterVar = "~v002 + v001", ageVar = "age",
    weightsVar = "v005", geo.recode = NULL)
colnames(u1m)[4] <- "IMR"
```

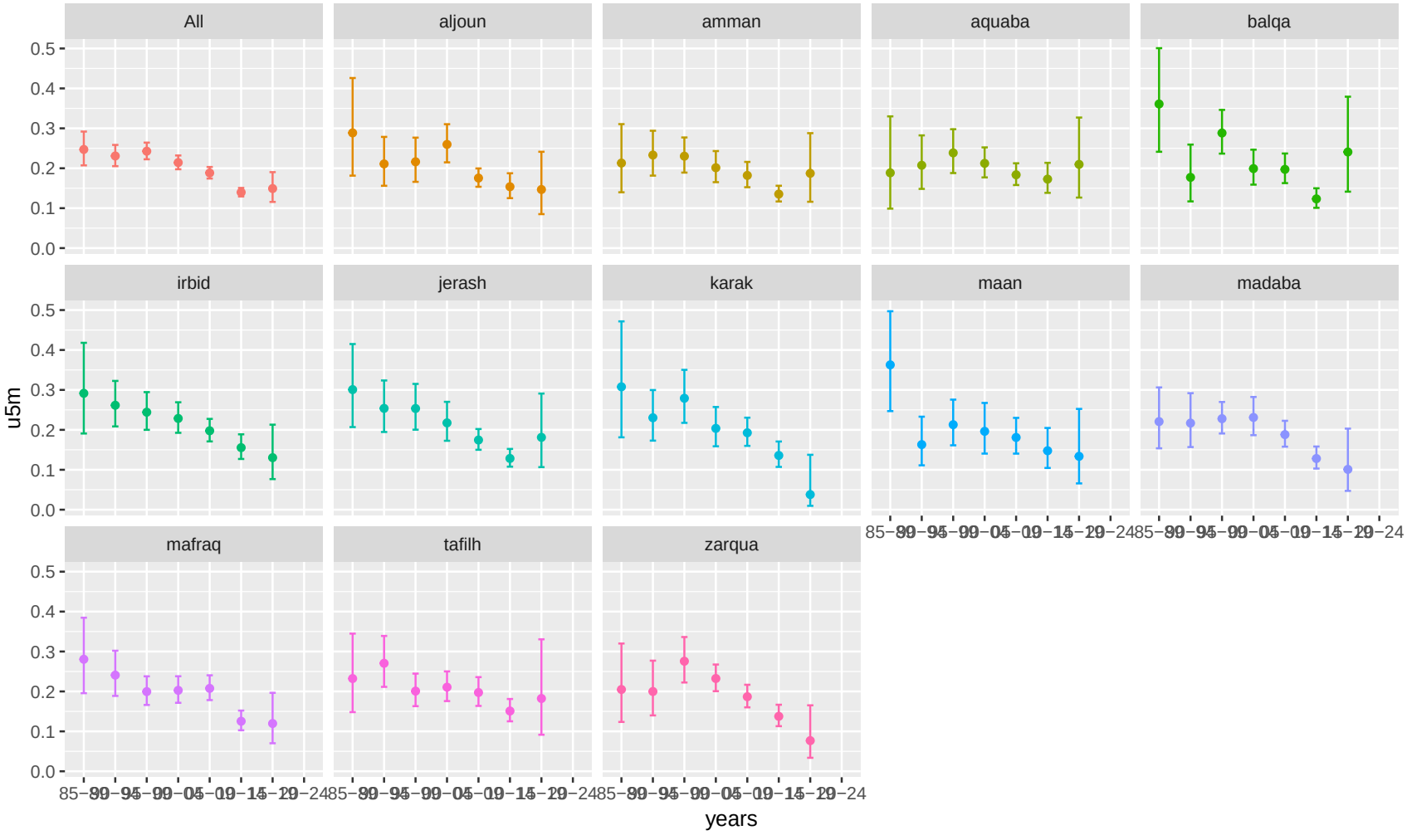
Neonatal mortality

```
births_neonates <- getBirths(data = dat,
    surveyyear = 2018, month.cut = c(1))
nmr <- countrySummary(births = births_neonates,
    years = years, regionVar = "v024", timeVar = "time",
    clusterVar = "~v002 + v001", ageVar = "age",
    weightsVar = "v005", geo.recode = NULL)
colnames(nmr)[4] <- "NMR"
```

Visualize direct estimates

```
library(ggplot2)
u5m$years <- factor(u5m$years, levels = years)
u5m$region <- factor(u5m$region, levels = )
ggplot(u5m, aes(x = years, y = u5m, ymin = lower,
    ymax = upper, color = region)) + geom_point() +
    geom_errorbar(width = 0.2) + facet_wrap(~region,
    ncol = 5) + theme(legend.position = "none")
```

Visualize direct estimates

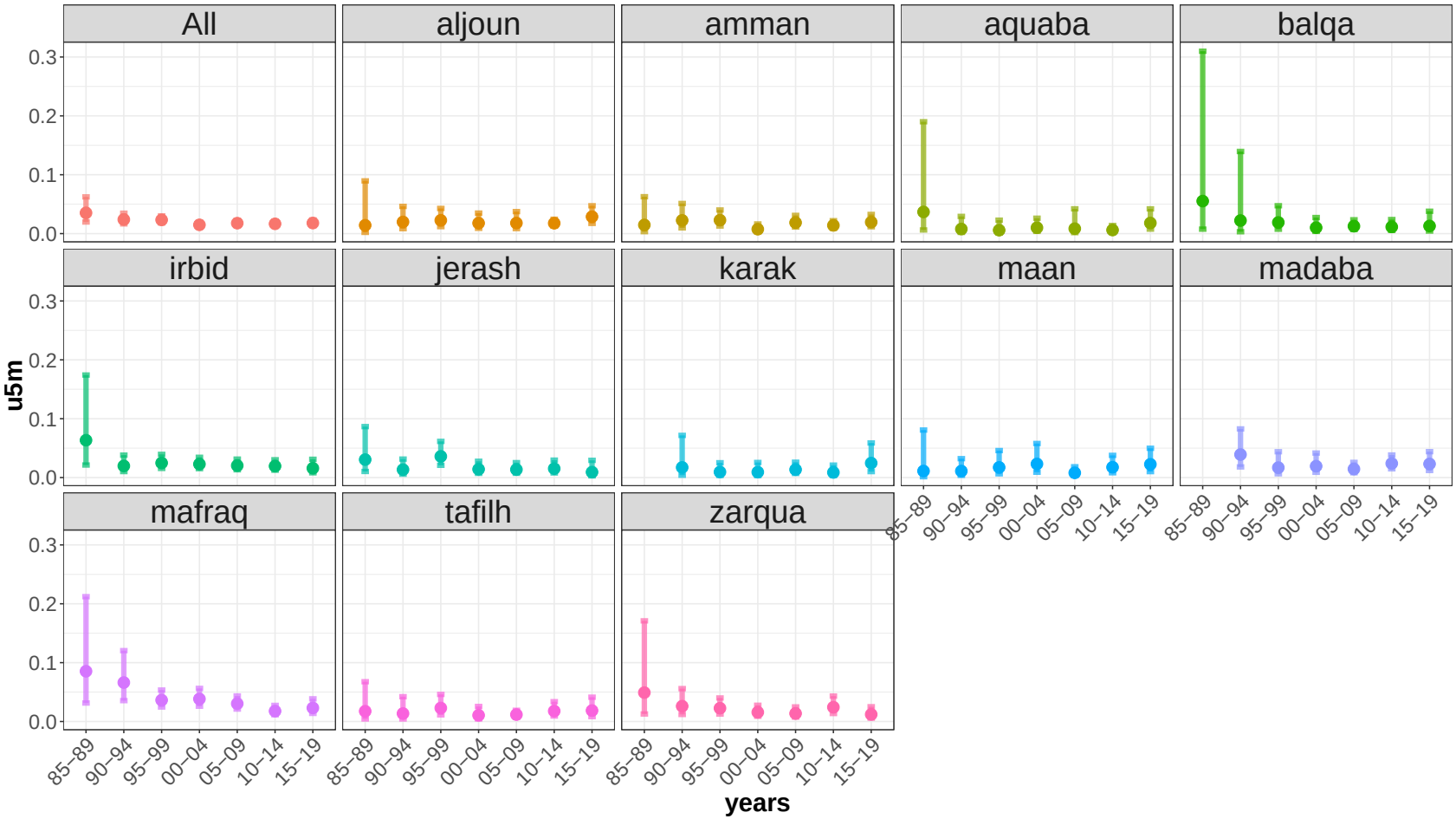


Summary

- SUMMER can help with calculating direct estimates from FBH data.
- Mortality by gender, and of different age groups can be specified with a little more manipulation of the data and model.
- Bias adjustments to the direct estimates can be performed after these steps as well.
- As discussed before, the direct estimates will be used for the smoothing model in the next step.

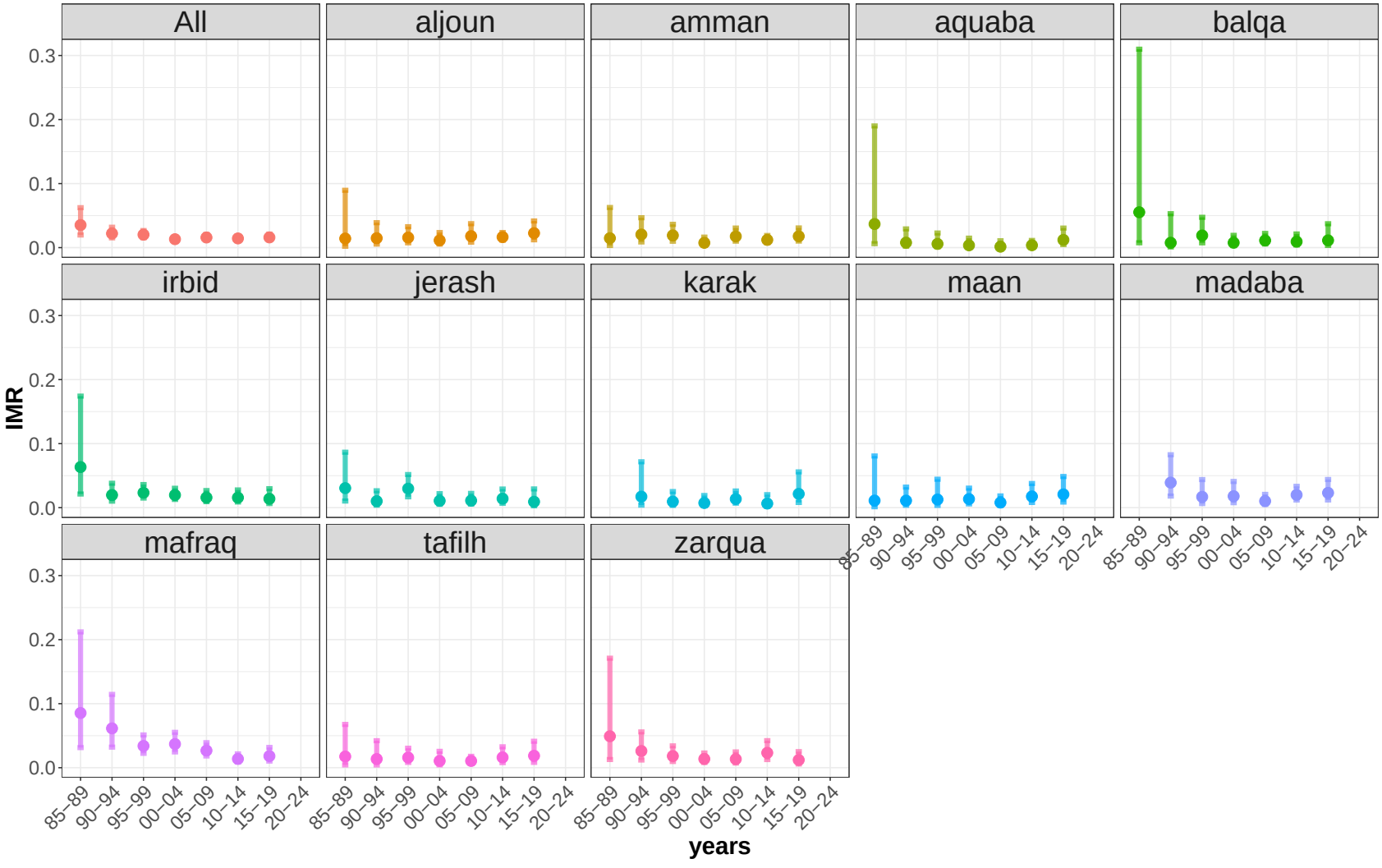
Space-time Smoothing

Actual U5MR estimates from Jordan 2017-2018 DHS

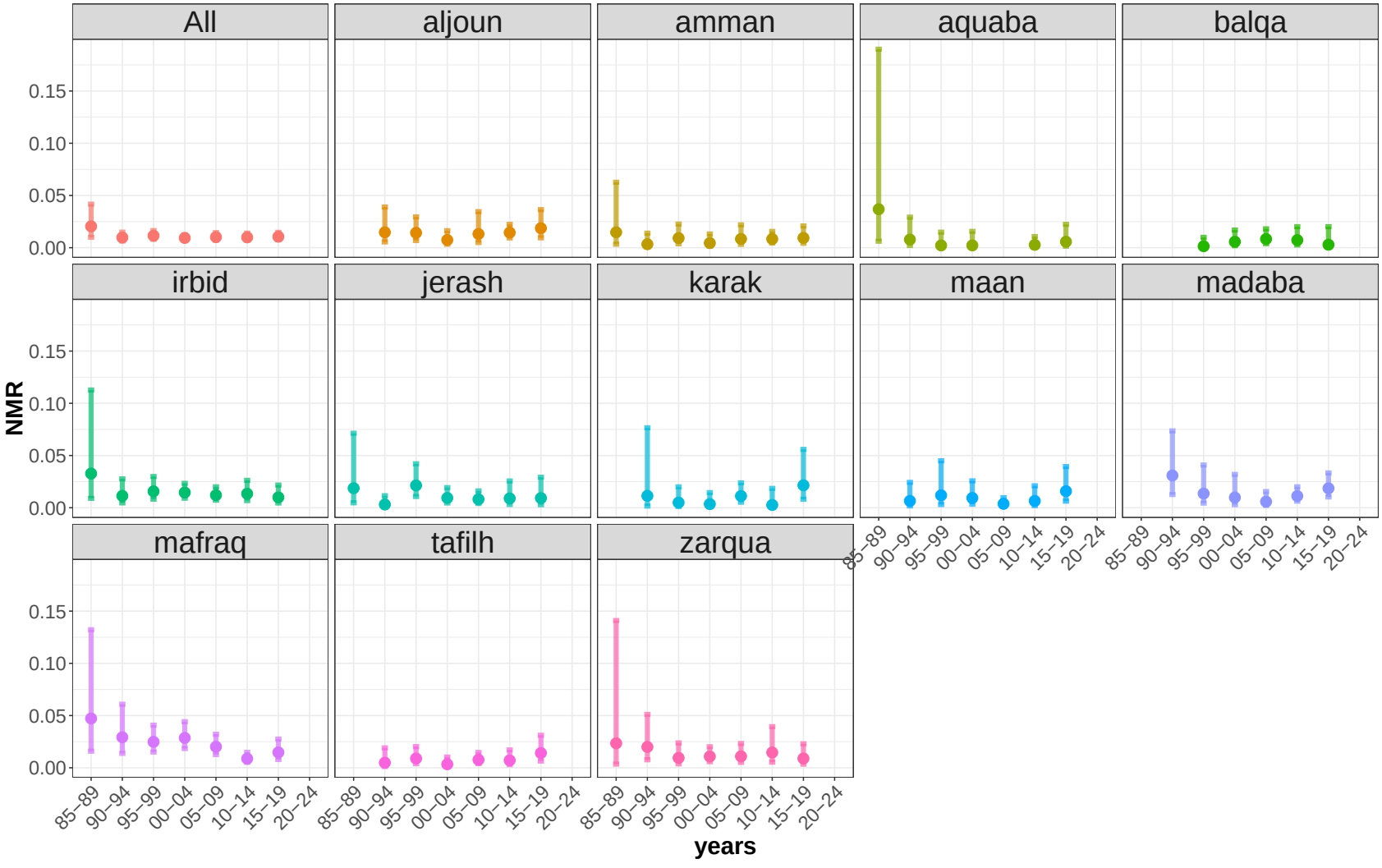


U5MR by sex: Jordan 2017-2018 DHS

IMR: Jordan 2017-2018 DHS



NMR: Jordan 2017-2018 DHS



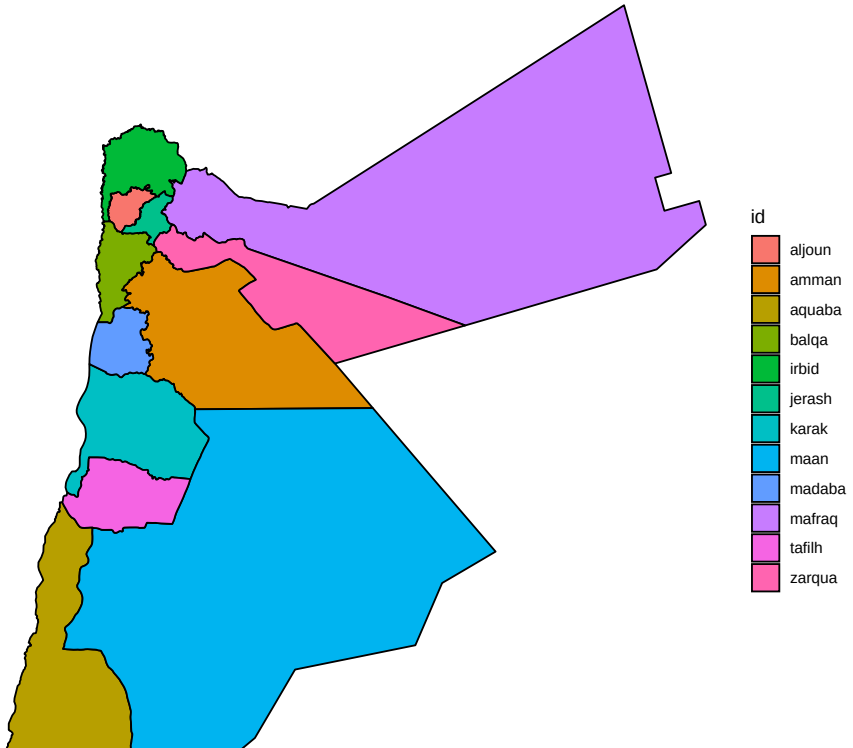
Getting to know the map

- ESRI (a company one of whose products is ArcGIS) shapefiles consist of three files, and this is a common form.
- The first file (.shp) contains the geography of each shape.
- The second file (.shx) is an index file which contains record offsets.
- The third file (.dbf) contains feature attributes with one record per feature.

```
library(spdep)
library(maptools)
f <- "../shapefiles/JO/sdr_subnational_boundaries.shp"
geo <- readShapePoly(f)
geo$REGNAME <- tolower(geo$REGNAME)
```

Getting to know the map

```
library(ggplot2)
g <- ggplot(fortify(geo, region = "REGNAME"))
g <- g + geom_polygon(aes(x = long, y = lat,
    group = group, fill = id), color = "black")
g <- g + theme_void() + coord_map()
print(g)
```



View map as an adjacency matrix

```
nb.r <- poly2nb(geo, queen = F, row.names = geo$REGNAME)
mat <- nb2mat(nb.r, style = "B", zero.policy = TRUE)
regions <- colnames(mat) <- rownames(mat)
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
nreg <- length(regions)
```

View map as an adjacency matrix

```
mat[1:5, 1:5]
```

```
##         aljoun amman aquaba balqa irbid
## aljoun       0     0       0     1     1
## amman        0     0       0     1     0
## aquaba       0     0       0     0     0
## balqa        1     1       0     0     1
## irbid        1     0       0     1     0
```

Bayesian smoothing

We will use INLA for the main workhorse to compute the Bayesian smoothing model.

```
# install.packages('INLA',
# repos=c(getOption('repos'),
# INLA='https://inla.r-inla-download.org/R/testing'),
# dep=TRUE)
library(INLA)
```

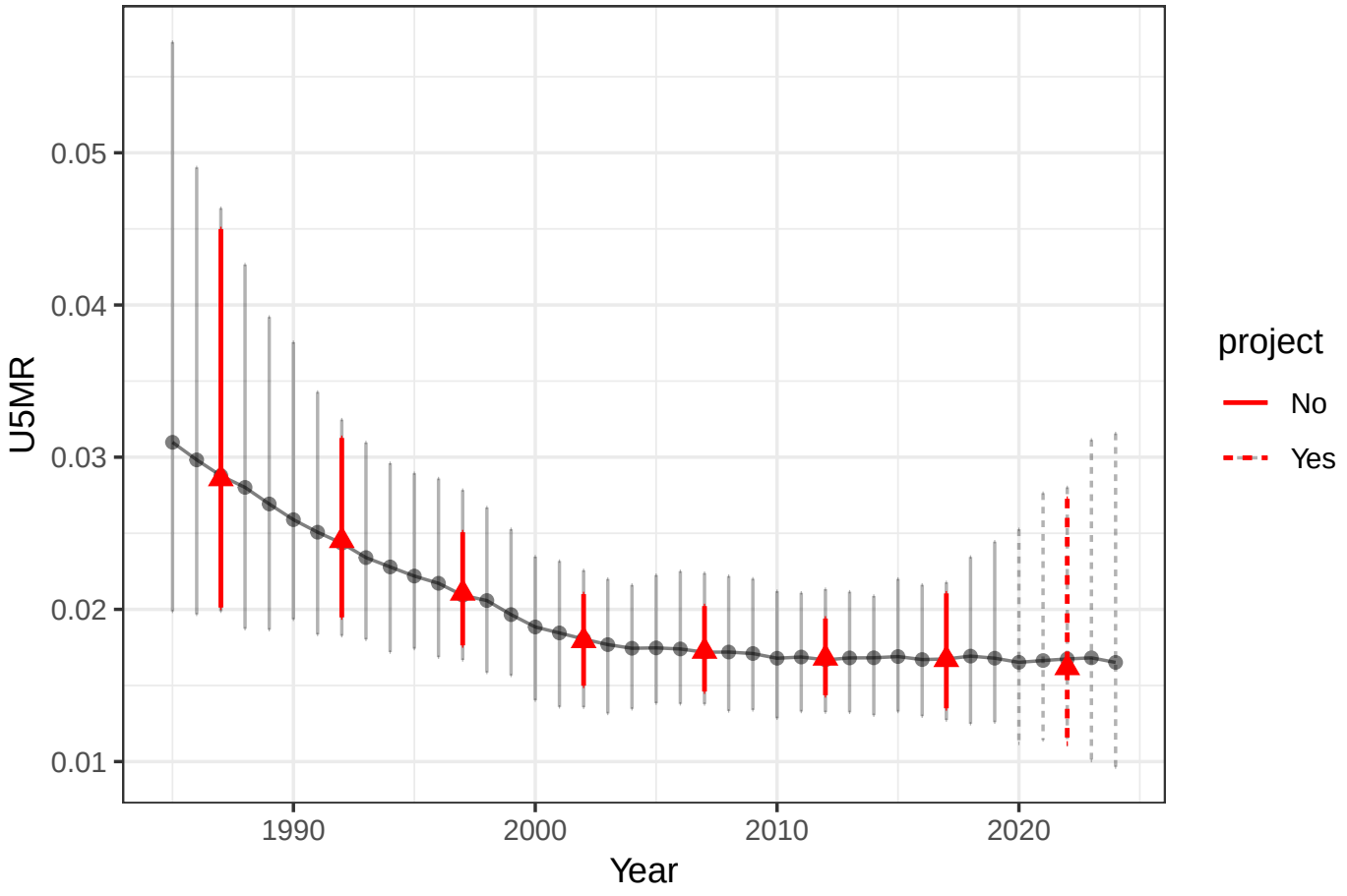
First, we read in the direct estimates (if we were using the actual DHS data, they will be exactly what we just calculated).

```
u5m <- read.csv("../data/u5m.csv")
u1m <- read.csv("../data/u1m.csv")
nmr <- read.csv("../data/nmr.csv")
```

Temporal smoothing of national estimates

```
years <- as.character(unique(u5m$years))
fit.national <- fitINLA(data = u5m, geo = NULL,
    Amat = NULL, year_names = years,
    year_range = c(1985, 2024), rw = 2,
    is.yearly=TRUE, m = 5)
proj.national <- projINLA(fit.national, is.yearly = TRUE,
    year_range = c(1985, 2024), year_label = years)
plot(proj.national, is.subnational = FALSE,
    is.yearly = TRUE, year_label = years,
    year_med = seq(1987, 2022, by=5),
    proj_year = 2020)
```

Temporal smoothing of national estimates



Spatial-temporal smoothing on the yearly scale

```
fit <- fitINLA(data = u5m, geo = geo, Amat = mat,
         year_names = years,
         year_range = c(1985, 2024), is.yearly=TRUE,
         rw = 2, m = 5, type.st = 4)
proj <- projINLA(fit, is.yearly = TRUE,
                      year_range = c(1985, 2024),
                      year_label = years,
                      Amat = mat)
```

The type.st = 4 arguments specifies the type IV space-time interaction model, i.e., the fully structured interaction. Details see Li et al (2019).

Spatial-temporal smoothing on the yearly scale

```
head(proj)
```

```
##    region years logit.q975 logit.q025
## 1 aljoun  1985  -2.985102  -4.649551
## 2  amman  1985  -2.928772  -4.426491
## 3 aquaba  1985  -2.945366  -5.274296
## 4  balqa  1985  -1.921923  -4.234751
## 5  irbid  1985  -2.574660  -4.027918
## 6 jerash  1985  -2.502601  -4.059391
##    logit.med       q975         q025
## 1 -3.839378 0.04850618 0.009447655
## 2 -3.706246 0.05233832 0.012005910
## 3 -4.072060 0.04979278 0.005371703
## 4 -3.120318 0.11558852 0.015822696
## 5 -3.264737 0.07127441 0.018977106
## 6 -3.242272 0.07430587 0.018738035
```

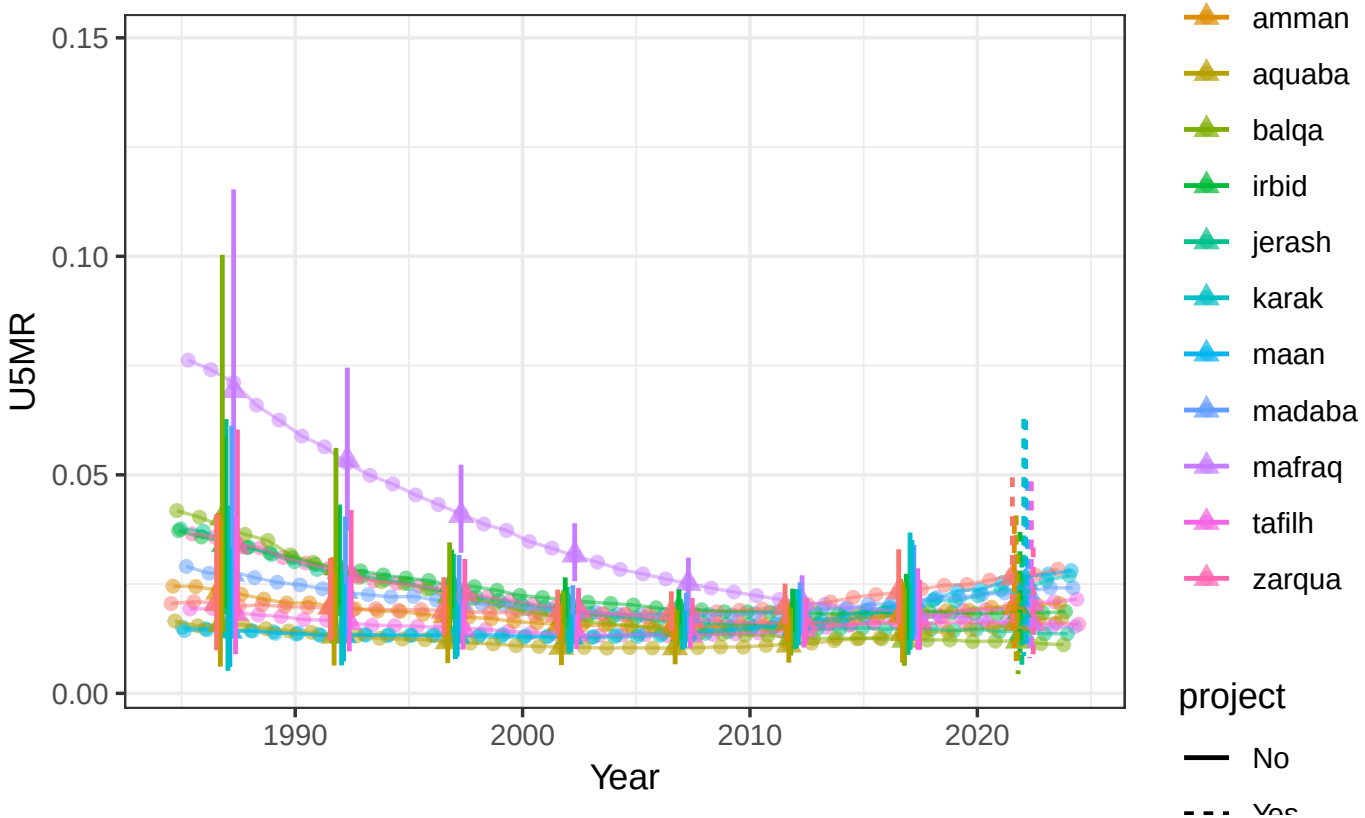
Spatial-temporal smoothing on the yearly scale

```
tail(proj)
```

```
##      region years logit.q975 logit.q025
## 571   karak 20-24  -2.587897  -4.605805
## 572    maan 20-24  -2.737280  -4.495279
## 573 madaba 20-24  -3.006332  -4.491748
## 574 mafraq 20-24  -3.376836  -4.834545
## 575 tafilh 20-24  -2.960538  -4.663739
## 576 zarqua 20-24  -3.379597  -4.865461
##      logit.med       q975        q025
## 571 -3.702694 0.06278819 0.008554950
## 572 -3.631886 0.06246431 0.010929448
## 573 -3.749313 0.04855140 0.011727112
## 574 -4.129658 0.03055193 0.008020667
## 575 -3.869549 0.04848589 0.009115892
```

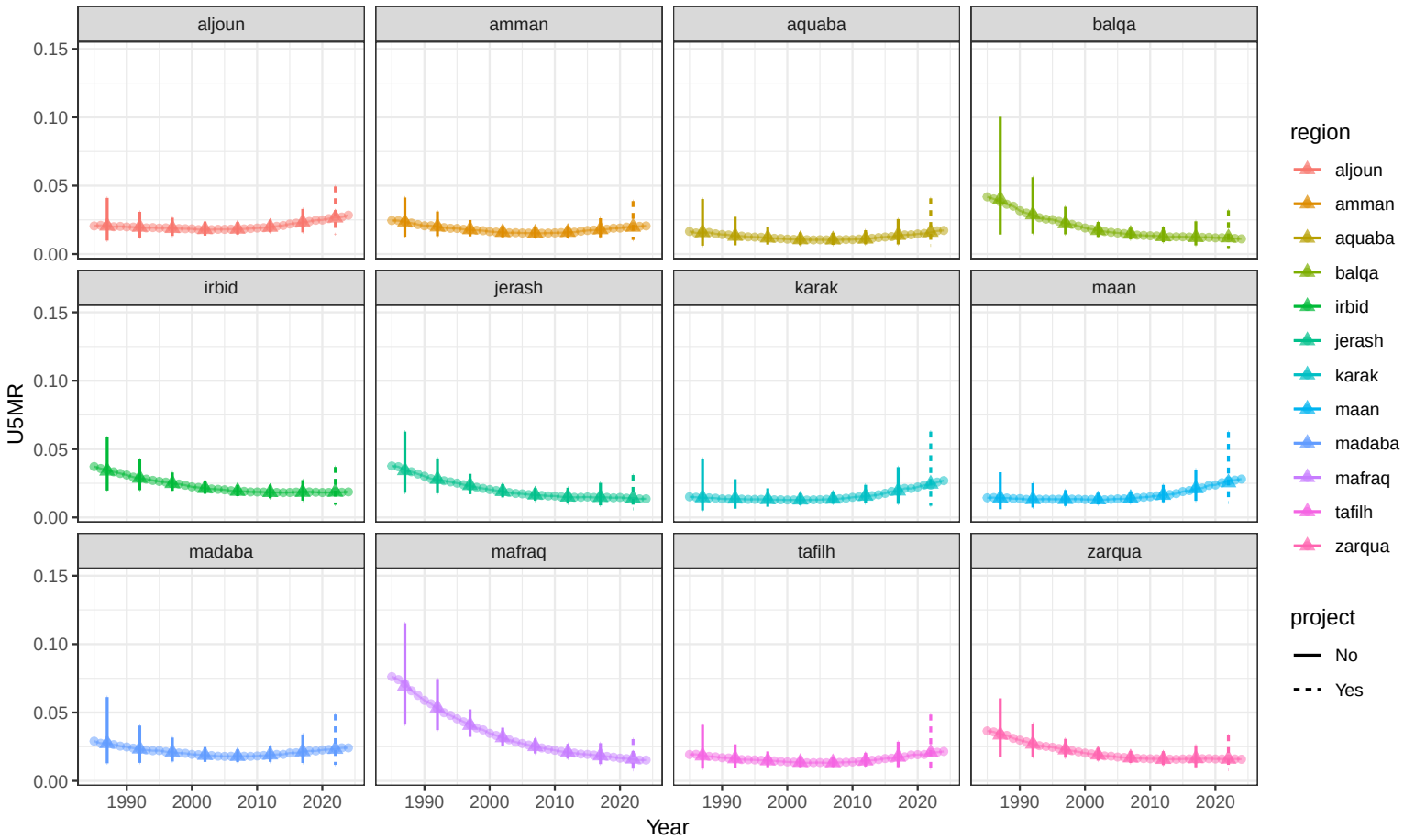
Spatial-temporal smoothing on the yearly scale

```
g <- plot(proj, is.yearly = TRUE, is.subnational = TRUE,
    year_label = years, year_med = seq(1987,
        2022, by = 5), proj_year = 2020)
g
```



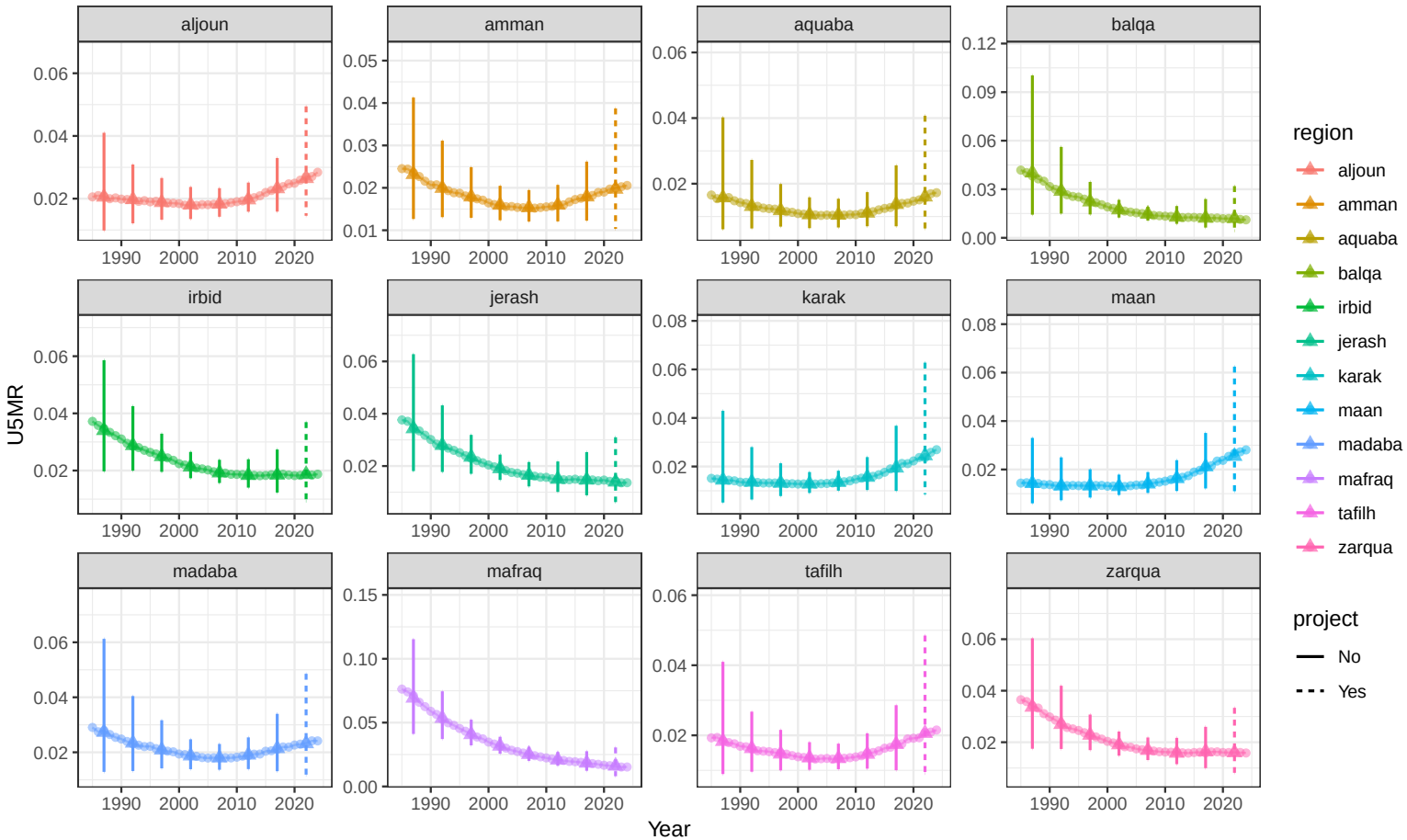
Spatial-temporal smoothing on the yearly scale

```
g + facet_wrap(~region)
```



Spatial-temporal smoothing on the yearly scale

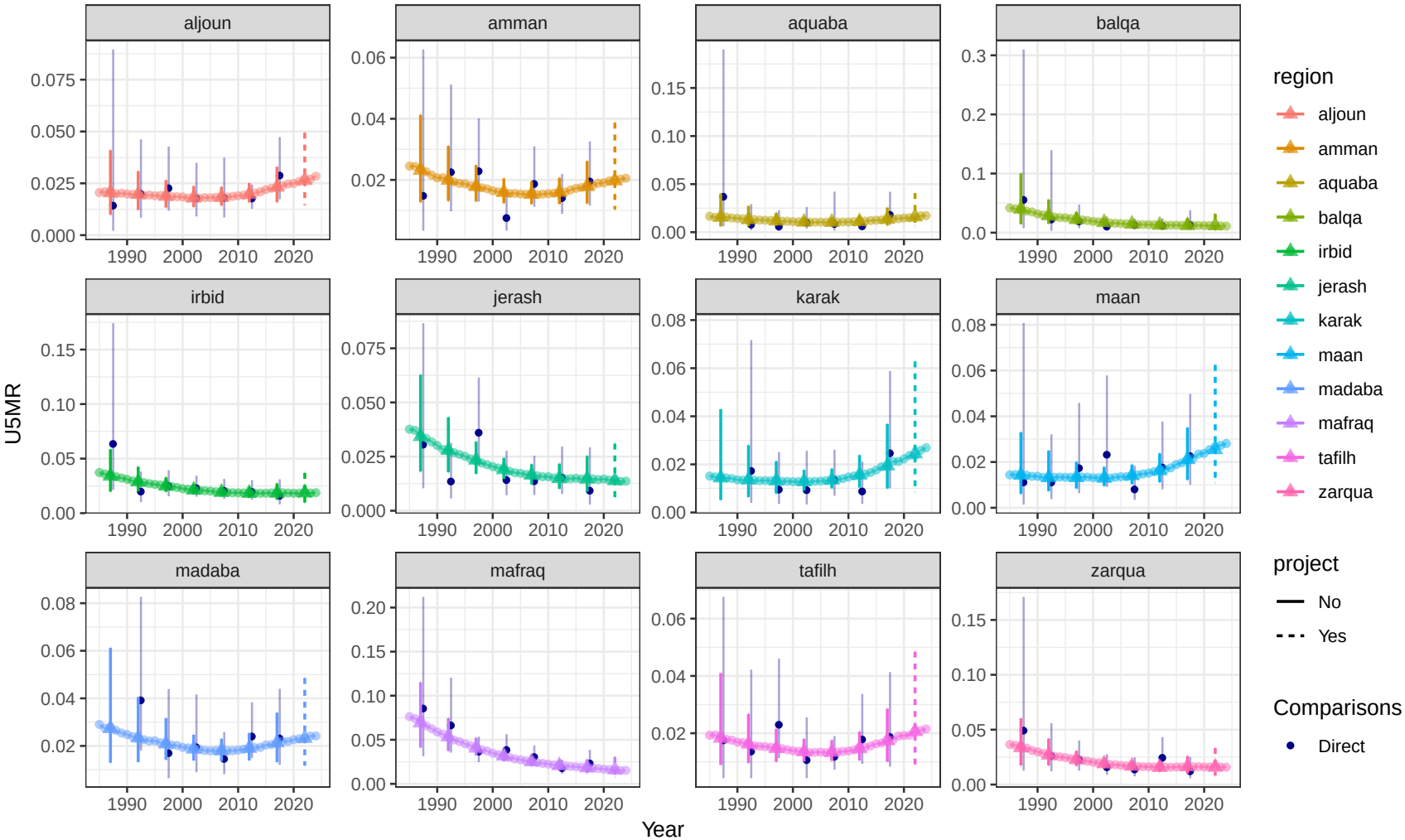
```
g + facet_wrap(~region, scales = "free")
```



Add back the direct estimates for comparison

```
g <- plot(proj, is.yearly = TRUE, is.subnational = TRUE,
    year_label = years, year_med = seq(1987,
        2022, by = 5), proj_year = 2020,
    data.add = u5m, option.add = list(point = "u5m",
        by = "survey", lower = "lower", upper = "upper"),
    color.add = "darkblue")
g + facet_wrap(~region, scales = "free")
```

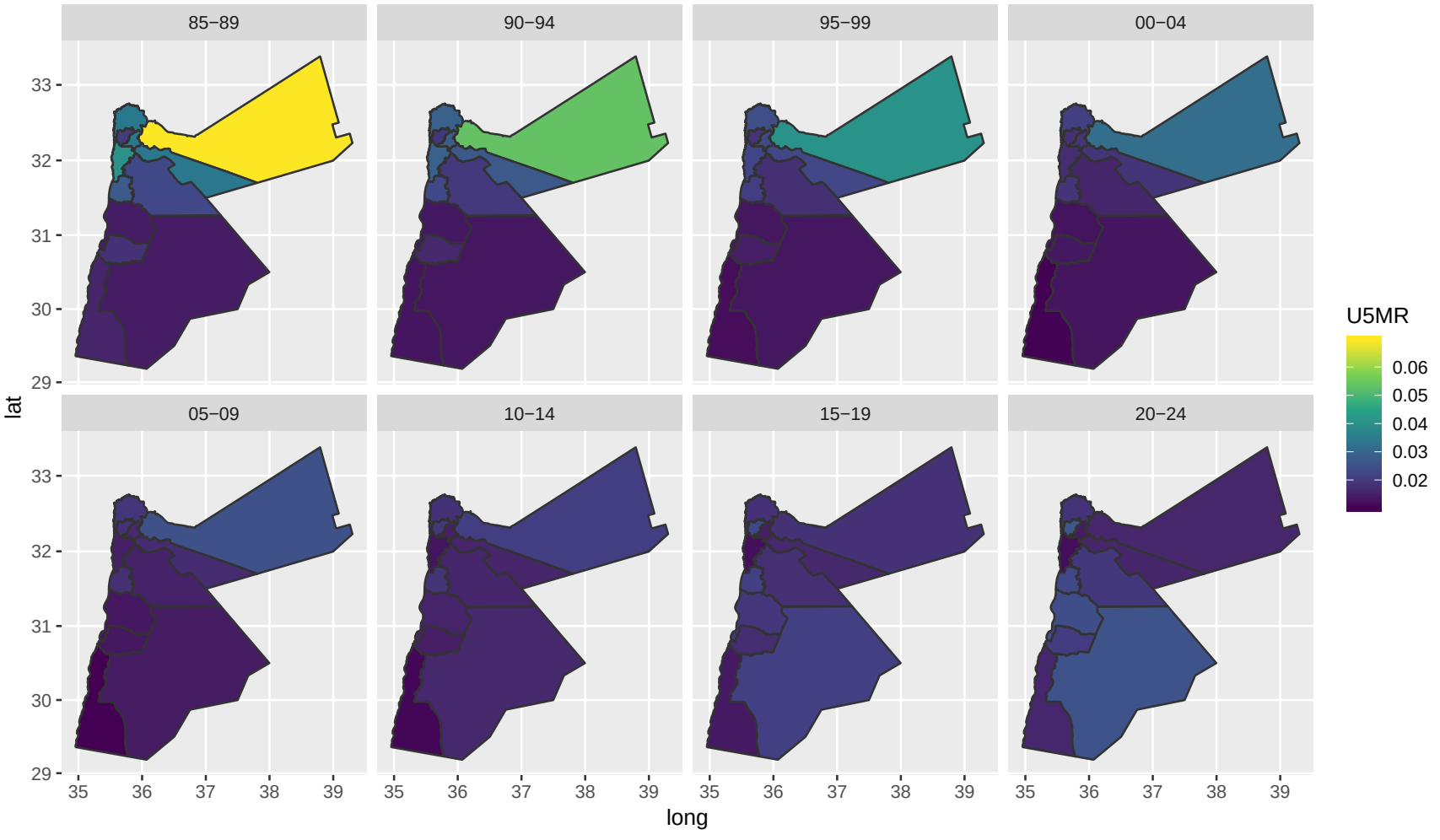
Add back the direct estimates for comparison



Visualize on the map

```
g <- mapPlot(data = subset(proj, is.yearly ==
    F), geo = geo, variables = c("years"),
    values = c("med"), by.data = "region",
    by.geo = "REGNAME", is.long = TRUE, ncol = 4)
g <- g + coord_map() + scale_fill_viridis_c("U5MR")
g
```

Visualize on the map



Summary

- Once data is in the right format, the space-time smoothing can be done relatively easily (and fast) with SUMMER.
- The smoothed estimates typically lead to smaller variance than the direct estimates by borrowing information from all subnational areas.
- We have only looked at one survey so far. The Bayesian hierarchical model framework can also incorporate multiple surveys as well. You will see it in action in the next exercise.
- More new features to come in SUMMER!

Example: Subnational Smoothed Estimates from Multiple Surveys

Hands-on time

We will use the example dataset derived from the DHS model data to calculate U5MR as an exercise.

Exercise

- Follow along the next few slides. Some of the tasks we have seen already.
- This exercise also deals with a new task of combining multiple surveys.
- Many of the customizations we have seen before are not necessary. The default functions calls match the variable name, time periods, etc. of the dataset.
- The DemoData has only 4 regions and can be fit quite quickly.

Read and process data

We will use the following commands to load the built-in person-month dataset, and the following time periods

```
data(DemoData)
years <- levels(DemoData[[1]]$time)
years.all <- c(years, "15-19")
```

This datasets contains 5 surveys (*try* `names(DemoData)`).

Q: Take a look at `DemoData[[1]]`, can you calculate direct estimates based on this survey? Compare with the next slides, where we compute the estimates from all surveys.

Obtain person-month from multiple surveys

```
data <- countrySummary_mult(births = DemoData,
    years = years, regionVar = "region",
    timeVar = "time", clusterVar = "~clustid+id",
    ageVar = "age", weightsVar = "weights",
    geo.recode = NULL)
head(data)
```

```
##   region years       u5m      lower
## 1    All 85-89 0.2373033 0.11954261
## 2    All 90-94 0.3250755 0.13332676
## 3    All 95-99 0.1484044 0.08851810
## 4    All 00-04 0.1410587 0.09294438
## 5    All 05-09 0.2319795 0.17516469
## 6    All 10-14 0.1639098 0.11747183
##       upper  logit.est    var.est
## 1 0.4162288 -1.1675213 0.17900781
```

Obtain map

We will use the built-in map and adjacency matrix instead of reading from shapefiles and calculating by hand:

```
data(DemoMap)
geo <- DemoMap$geo
mat <- DemoMap$Amat
```

Q: Can you visualize the map? (*try* `plot(geo)`)

Combine multiple surveys

We have not seen this function in the example before. It basically calculates a weighted average for each area and time from the multiple surveys.

$$ {}_5\widehat{q}_0^{\,it} = \text{expit}\left(\sum_{s=1}^{S_t} \underbrace{\left[\frac{\widehat{V}_{\text{DES},its}^{-1}}{\sum_{s=1}^{S_t} \widehat{V}_{\text{DES},its}^{-1}} \right]}_{\text{Weight for survey } s} \text{logit}({}_5\widehat{q}_0^{\,its}) \right), $$

and

$$ \widehat{V}_{\text{DES},it} = \frac{1}{\sum_{s=1}^{S_t} \widehat{V}_{\text{DES},its}^{-1}}. $$

```
data_agg <- aggregateSurvey(data)
dim(data_agg)
```

```
## [1] 30 10
```

Calculate subnational smoothed estimates of U5MR

Q: How do we use `fitINLA()` and `projINLA()` to fit the model and calculate the smoothed estimates?

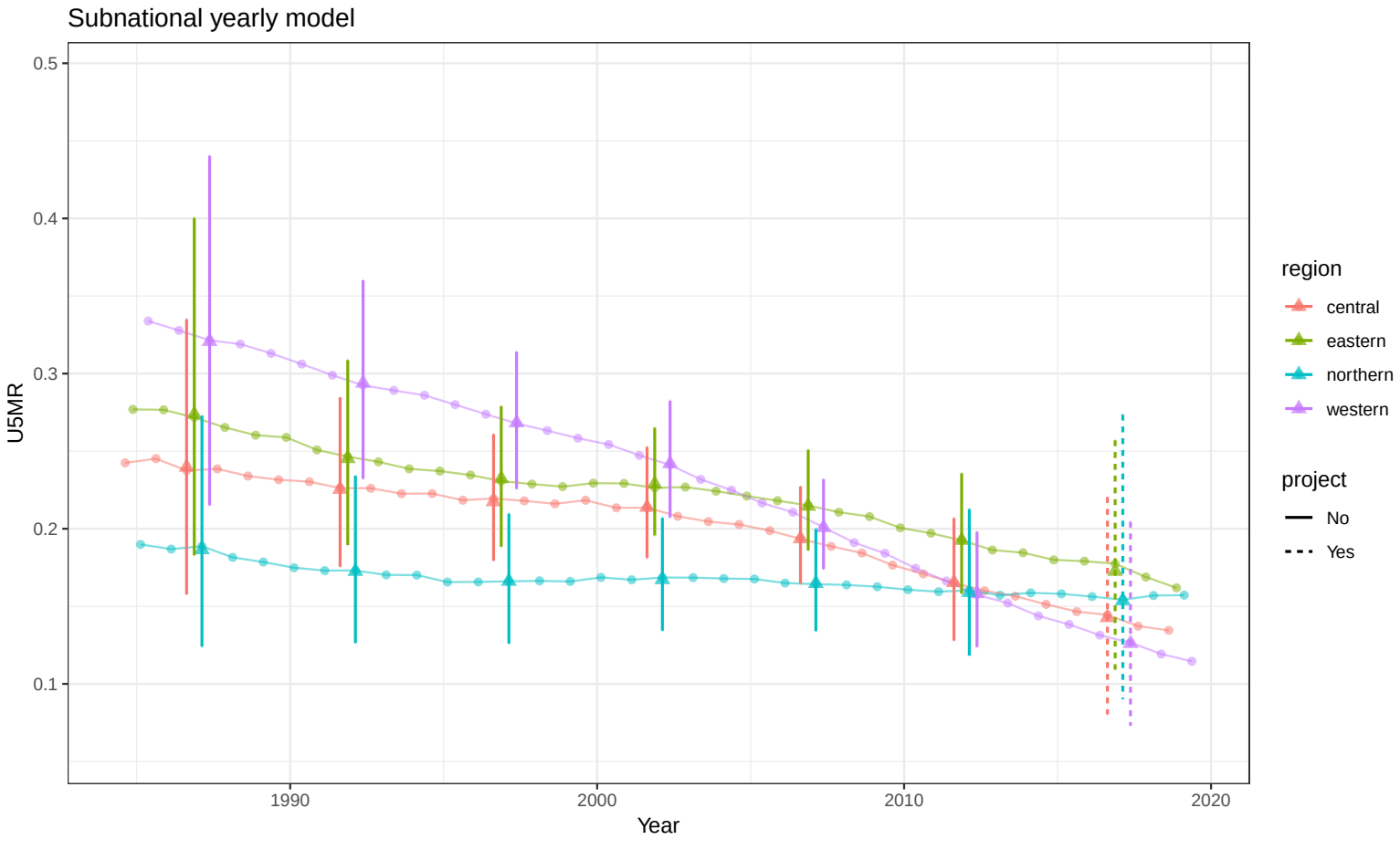
```
fit <- fitINLA(data = data_agg, geo = geo, Amat = mat,
         year_names = years.all, year_range = c(1985, 2019),
         rw = 2, is.yearly=TRUE, m = 5, type.st = 4)
out <- projINLA(fit, Amat = mat, is.yearly = TRUE)
```

Q: What are these arguments?

Q: How to visualize the data frame `out`?

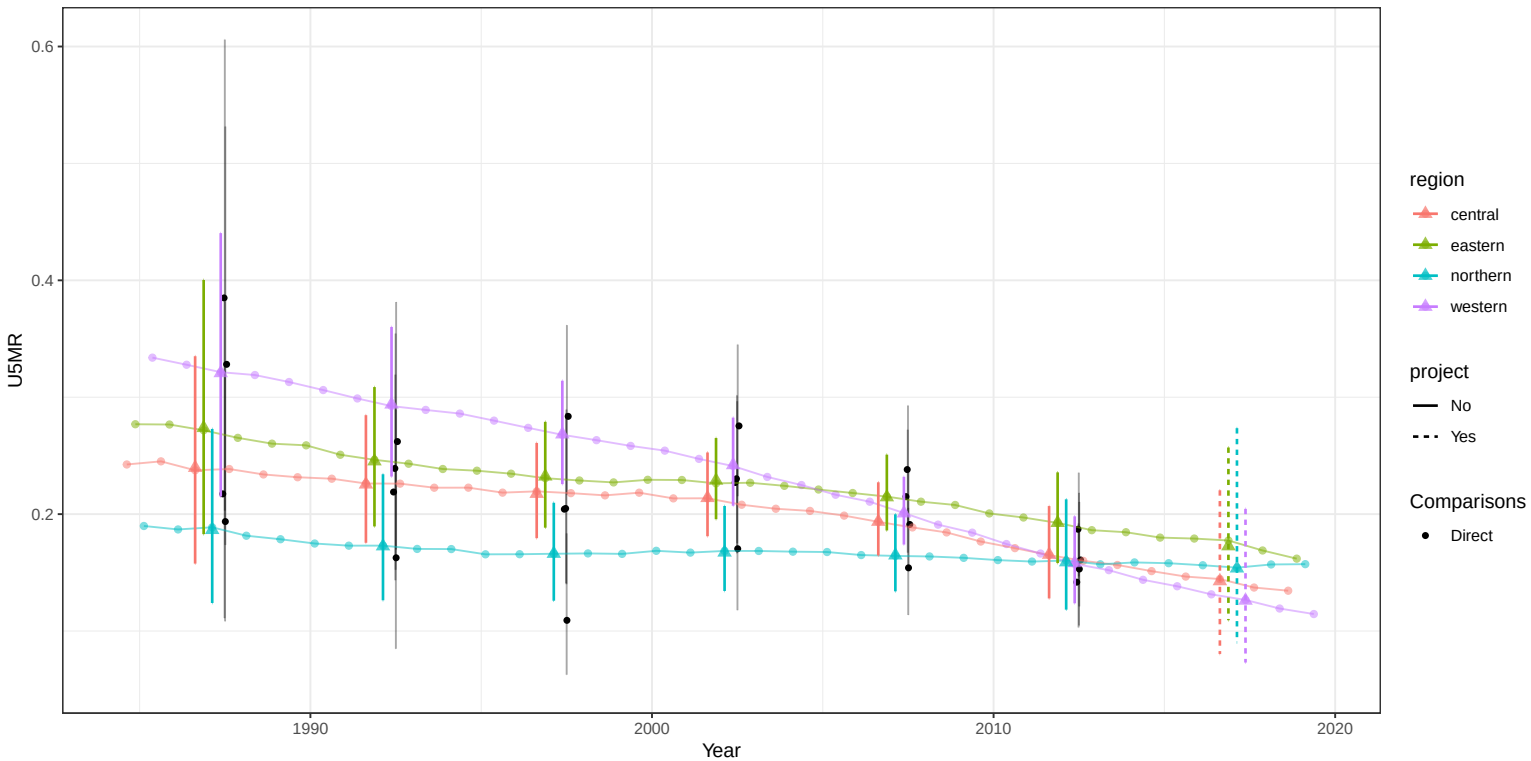
Time-series plot

```
plot(out, is.yearly = TRUE, is.subnational = TRUE) +
    ggplot2::ggtitle("Subnational yearly model")
```



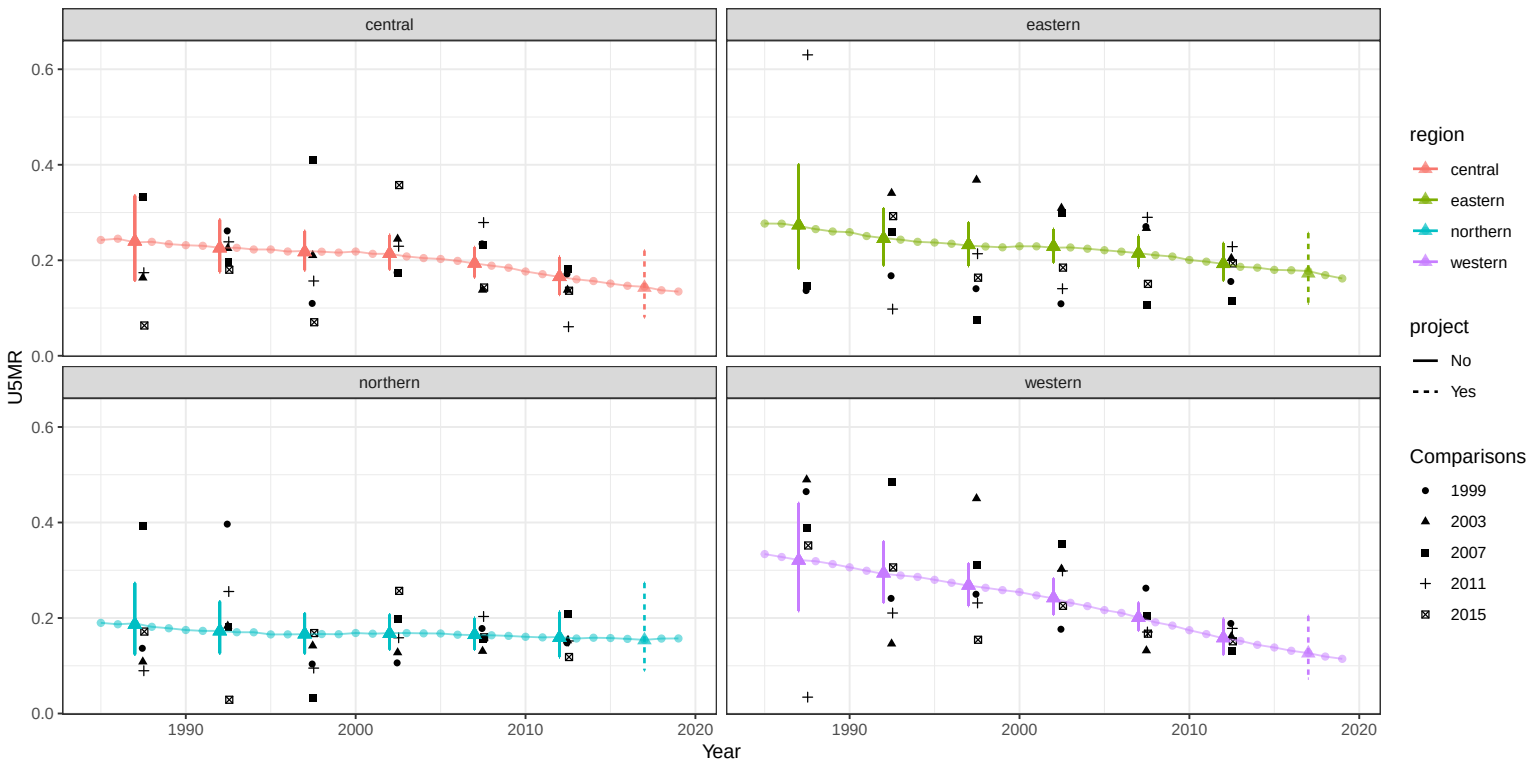
Time-series plot with the direct estimates

```
plot(out, is.yearly = TRUE, is.subnational = TRUE,
     data.add = data_agg, option.add = list(point = "u5m",
          lower = "lower", upper = "upper"))
```



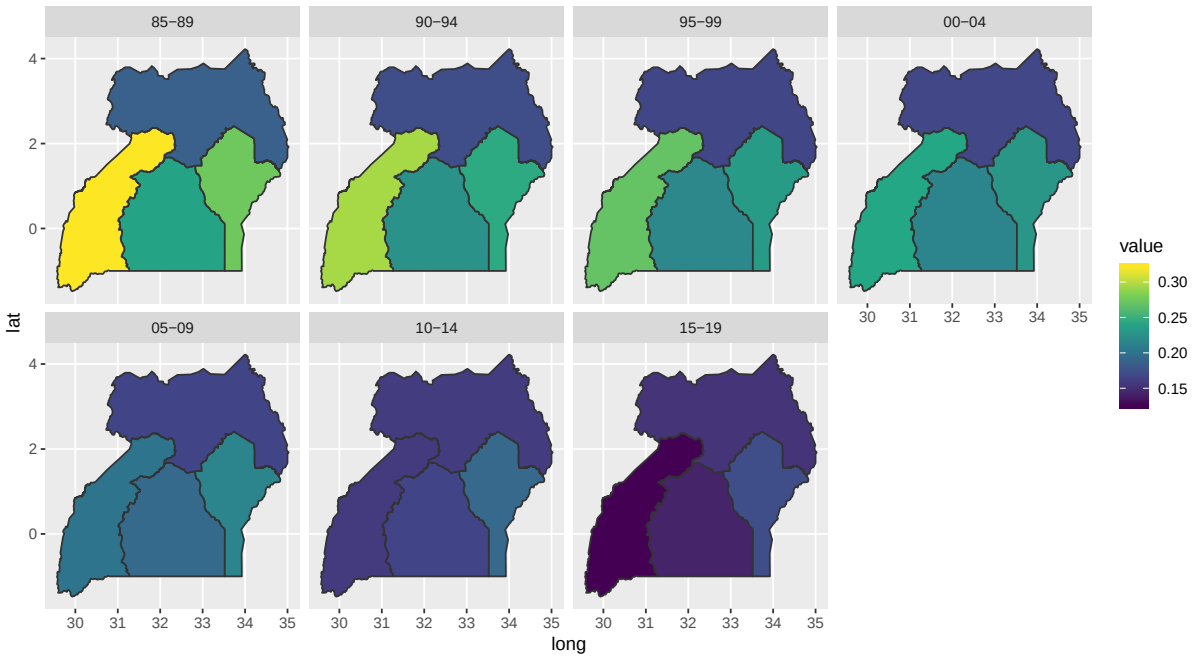
Time-series plot with the direct estimates from each survey

```
plot(out, is.yearly = TRUE, is.subnational = TRUE,
     data.add = data, option.add = list(point = "u5m",
          by = "surveyYears")) + facet_wrap(~region)
```



Plot the changes on the map

```
mapPlot(data = subset(out, is.yearly == F),
    geo = DemoMap$geo, variables = c("years"),
    values = c("med"), by.data = "region",
    by.geo = "NAME_final", is.long = TRUE,
    ncol = 4) + coord_map()
```



Summary

More details about using SUMMER package can be found in the vignette:
https://cran.r-project.org/web/packages/SUMMER/vignettes/summer-vignette.pdf

Questions/Feature requests? Submit an issue at
https://github.com/bryandmartin/SUMMER/issues