

Term Paper Report

Spencer Pease

6/09/2020

Abstract

The article *Fully Bayesian Benchmarking of Small Area Estimation Models* (2020, Zhang, J.L. and Bryant, J.) presents a new approach to small domain estimation, using larger area estimates and Bayesian hierarchical models to provide small area estimate posterior distributions congruent with neighboring small areas and the larger region encompassing them. This report begins with a summary of the article, followed by an analysis of the methods presented in the article and their recreation for this report. Then these methods are evaluated by (1) recreating the real-world application to estimating mortality rates for local authority districts in England and Wales presented by the authors, and (2) applying the same analysis to mortality rates in Philippine provinces. This report concludes with a discussion of these evaluations.

Contents

1	Article Summary	3
2	Methods Analysis & Implementation	4
2.1	Overview	4
2.2	Limitations	5
3	Methods Evaluation	5
3.1	Replication	5
3.2	Extension	10
4	Discussion	14
5	Appendix	16
5.1	References	16
5.2	Code	16

1 Article Summary

As the need for estimates of small areas defined by social, demographic, and geographic variables increases, so too does the need for methods overcoming the difficulties of small area estimations. Due to unreliability from small sample sizes in these areas, model-based estimates are preferred over direct methods (i.e., calculating a rate by taking number of events over exposure). These small area estimates are frequently compared to aggregate estimates for larger areas that are more easily likely obtained using direct methods (referred to as *benchmarks*). Small area estimation is important to government statistical offices, where reported statistics are used to inform policy and financial decisions. Inconsistencies between the low level and aggregate estimates therefore cause varying degrees of concern depending on the application, so small area estimates are often adjusted to be consistent with aggregate estimates (referred to as *benchmarking*). Currently existing benchmarking methods use benchmarks as constraints in the small area estimation models, but are limited to only providing point estimates for small area parameters.

In their article *Fully Bayesian Benchmarking of Small Area Estimation Models* (2020, Zhang, J.L. and Bryant, J.), Zhang and Bryant propose, implement, and demonstrate a fully Bayesian general approach to benchmarking. Their approach produces a full benchmarked posterior distribution by taking benchmarks as estimates for underlying aggregate parameters and modifying the likelihood function by multiplying it by the probability distribution of the benchmarks. Benefits of this approach include the ability to use multiple benchmarks, benchmarks nonlinearly related to small area estimates, and to specify acceptable discrepancy between benchmarks and model-based estimates.

Throughout the article and this report, some terminology is just that may not be immediately clear, so brief definitions are provided here for reference:

- **Benchmarks** are aggregate (higher level) estimates, generally obtained using direct methods
 - **Internal benchmarks** are calculated from the small area data sources
 - **External benchmarks** are calculated from sources separate from the small area data
- **Benchmarking** methods are techniques for reconciling small area estimates with benchmarks
 - **Exact benchmarking** requires the aggregated small area estimates to exactly match their benchmark

- **Inexact benchmarking** allows some maximum degree of difference between the aggregated areas and the benchmark

2 Methods Analysis & Implementation

2.1 Overview

The authors' methods are based on a conceptual framework that starts with a standard setup for Bayesian estimation of area-level models. The goal is to estimate area-level parameters $\gamma = \{\gamma_1, \dots, \gamma_n\}^T$ from area-level observations $y = \{y_1, \dots, y_n\}^T$ for the n areas defined by unique classification groupings (such as age-sex-region). The hierarchical Bayesian model then becomes:

$$p(\gamma, \phi|y) \propto p(\phi)p(\gamma|\phi)p(y|\gamma)$$

with ϕ representing a vector of hyperparameters, likelihood $p(y|\gamma)$, and prior $p(\phi)p(\gamma|\phi)$.

Extending this framework to include benchmarking requires collecting a set of observed benchmarks $m = \{m_1, \dots, m_d\}^T$ estimating underlying parameters $\psi = \{\psi_1, \dots, \psi_d\}^T$ for $d \ll n$ areas defined by unique aggregate classification groupings (such as age-sex, or age-sex-location where location includes multiple regions). A deterministic benchmarking function $\psi = f(\gamma)$ is defined to set the relationship between the small area and benchmark parameters so that each small area belongs to at most one benchmark area. This benchmarking function does not have to be linear – the later evaluations benchmark small area mortality rate estimates against life expectancy at birth.

To measure agreement with the benchmarks, a probability distribution for the benchmarks conditional on the aggregate parameters is defined as:

$$p^{[m|\psi]}(m|\psi) = p^{[m|\psi]}(m|f(\gamma))$$

This is distribution is then multiplied with the original likelihood to produce:

$$p(y|\gamma) p^{[m|\psi]}(m|f(\gamma))$$

This modified likelihood represents a compromise between the original likelihood and the

benchmarking requirements, where values of γ yielding larger $p^{[m|\psi]}(m|f(\gamma))$ inflate the original likelihood.

With this revised likelihood, the benchmarked posterior distribution becomes:

$$p(\gamma, \phi|y) \propto p(\phi)p(\gamma|\phi)p(y|\gamma)p^{[m|\psi]}(m|f(\gamma))$$

Under exact benchmarking, $p^{[m|\psi]}$ simplifies to 1 when $m = f(\gamma) = \psi$ and 0 otherwise. For inexact benchmarking, $p^{[m|\psi]}$ takes the form of whatever distribution fits the desired definition. One simple case is allowing differences within a given tolerance α , where $p^{[m|\psi]}$ would then become:

$$m_j \sim N(\psi_j, \alpha^2)$$

The authors implement these Bayesian hierarchical models using MCMC Metropolis-Hastings algorithm.

2.2 Limitations

One point the authors make is the difference between the effects of internal and external benchmarking on model performance. Generally, external benchmarking will improve model performance over no benchmarking if $p^{[m|\psi]}$ is correctly specified or not. Internal benchmarking, on the other hand, will hurt performance if $p^{[m|\psi]}$ is correctly specified, since small area data is used twice. However, a misspecified model is more common, which makes it possible for inexact benchmarking to improve performance, though this is not certain and depends on the details of the data.

3 Methods Evaluation

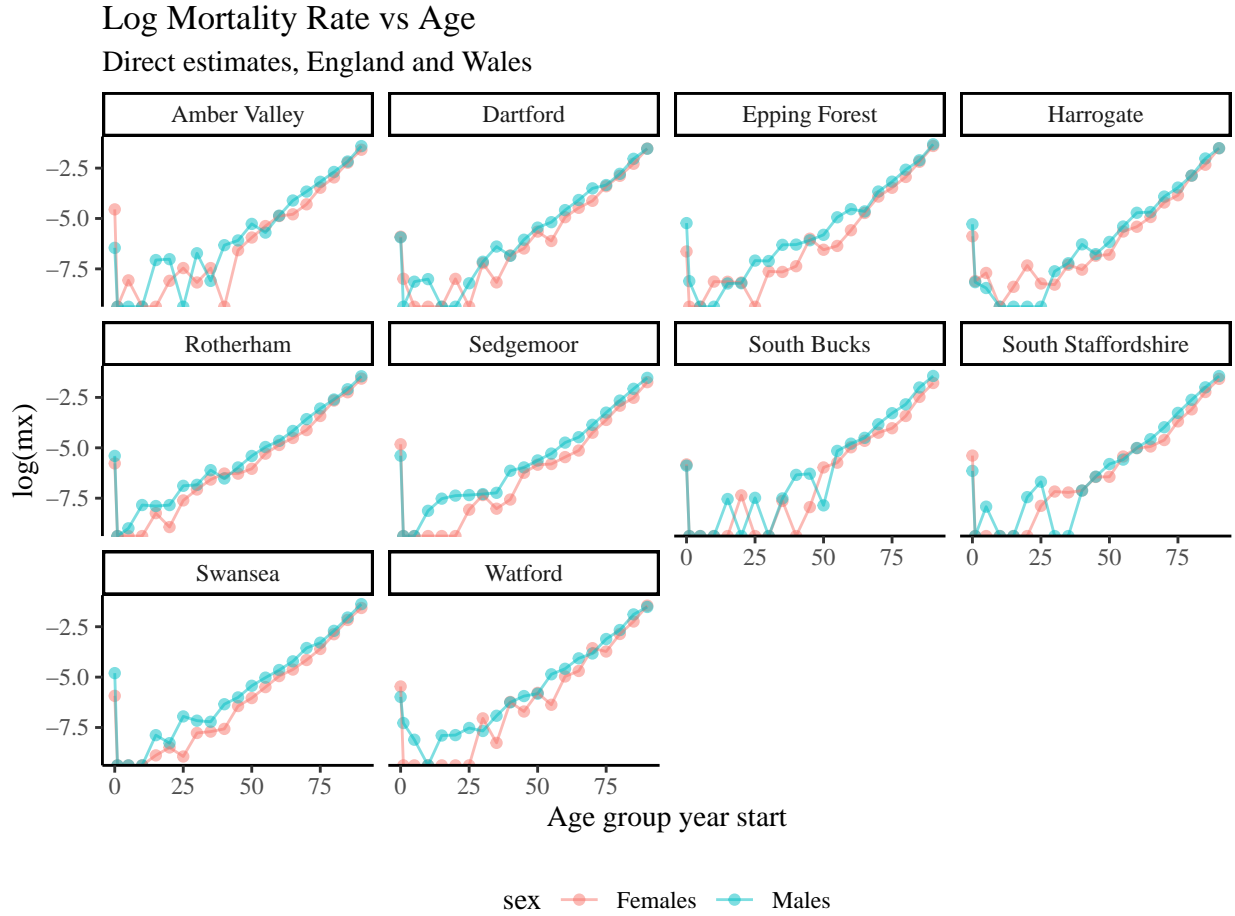
3.1 Replication

Part one of the methods evaluation is to recreate the application presented in the authors' article: estimating age-sex-specific mortality rates in 2014 for local authority districts (LAD) in England and Wales, using as benchmarks sex-specific life expectancies at birth for regions. The data for this analysis is death counts and populations at risk for 20 age groups 0-90+, each sex, and 348 LADs.

Table 1: Summary statistics for England and Wales
LADs

variable	n	missing	min	25%	median	mean	75%	max
deaths	347	0	0	1.00	8	36.05	48.00	1004
population	347	0	17	1984.75	3211	4136.07	5033.25	50381

There are 5.00314×10^5 total deaths among a total population of 5.7408654×10^7 . We can directly estimate mortality rates for each LAD, 10 of which are presented here:



These direct estimates are generally noisy below age 60, suggesting the amount of data at the small area level is not sufficient for accurate results.

The authors let y_{asd} represent observed death counts at each age (a), sex (s), and district (d). With γ_{asd} as the true underlying mortality rate and w_{asd} the corresponding population at risk, they apply the model:

$$y_{\text{asd}} \sim \text{Poisson}(w_{\text{asd}}\gamma_{\text{asd}})$$

$$\log(\gamma_{\text{asd}}) \sim N(\beta^0 + \beta_a^{\text{age}} + \beta_s^{\text{sex}} + \beta_d^{\text{dis}} + \beta_{\text{as}}^{\text{age:sex}}, \sigma^2)$$

Age effects are assumed to follow a random walk with drift:

$$\beta_a^{\text{age}} \sim t_4(\eta_a^{\text{age}}, \tau_{\text{age}}^2)$$

$$\eta_0^{\text{age}} \sim N(0, 10^2)$$

$$\eta_a^{\text{age}} \sim N(\eta_{a-1}^{\text{age}} + \delta_{a-1}^{\text{age}}, \omega^2), \quad a > 0$$

$$\delta_0^{\text{age}} \sim N(0, 1)$$

$$\delta_a^{\text{age}} \sim N(\delta_{a-1}^{\text{age}}, \varphi^2), \quad a > 0$$

The sex effect has a normal prior $\beta_s^{\text{sex}} \sim N(0, 1)$, the district effect has a normal prior $\beta_d^{\text{dis}} \sim N(0, \tau_{\text{dis}}^2)$ with a weakly informative half-t prior on the standard deviation $\tau_{\text{dis}} \sim t_7^+(0, 1)$, and the interaction term has a normal prior $\beta_{\text{as}}^{\text{age:sex}} \sim N(0, \tau_{\text{age:sex}}^2)$ with a weakly informative half-t prior on the standard deviation $\tau_{\text{age:sex}} \sim t_7^+(0, 0.5^2)$. All of the standard deviation terms have $t_7^+(0, 1)$ priors.

These mortality estimate are benchmarked to sex-specific life expectancy at birth at the 10 region levels (one administrative unit up from an LAD) encompassing the 348 LADs. If z_{asr} is age-sex-region mortality rates, then our benchmarking function, f_{life} , is defined as:

$$m_{\text{sr}} = f_{\text{life}}(z_{1\text{sr}}, \dots, z_{A\text{sr}})$$

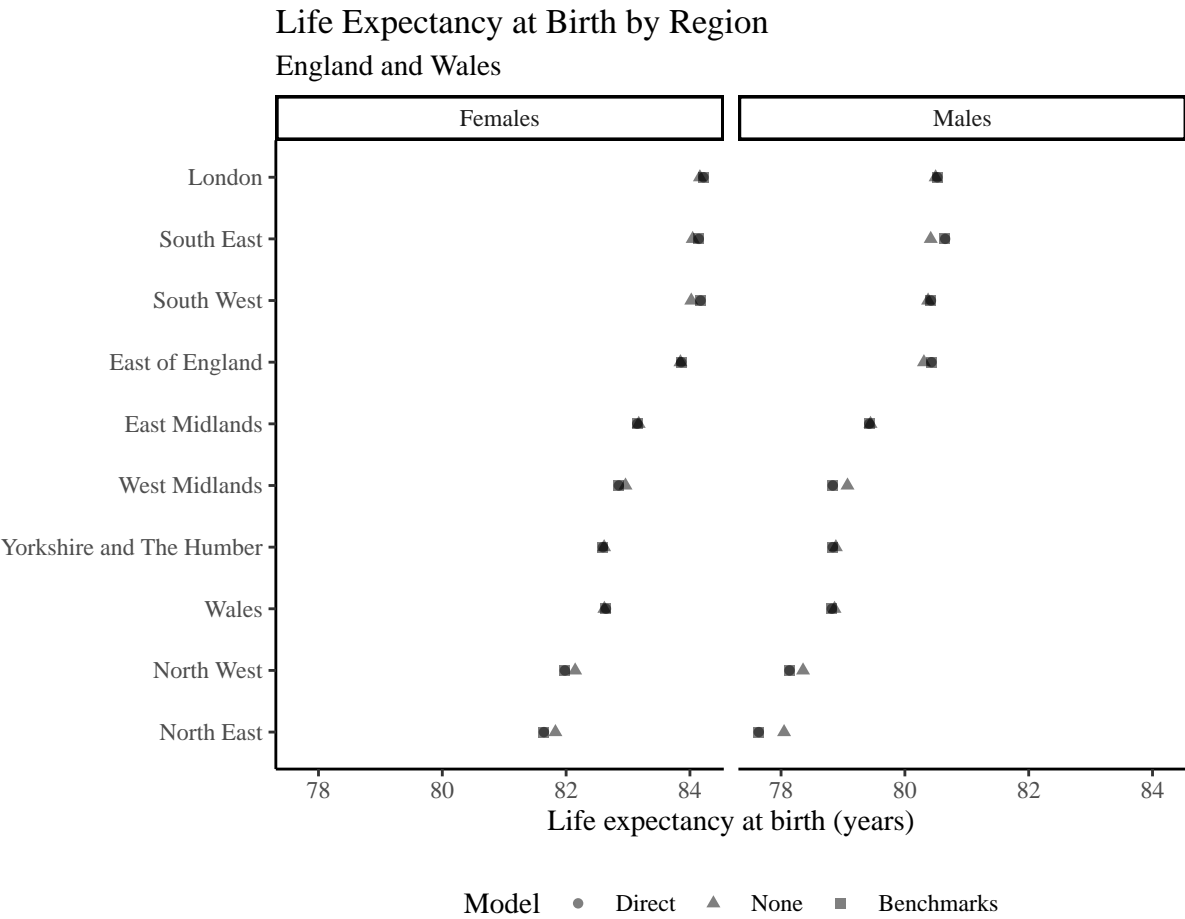
Where $A = 20$ is the number of age groups. Life expectancies are normally reported to two decimal places, so the authors incorporate inexact benchmarking allowing a discrepancy of 0.01. Agreement with the benchmark is then:

$$m_{\text{sr}} \sim N(\psi_{\text{sr}}, 0.005^2)$$

The authors fit this model twice, with and without benchmarking, using four independent chains with 40,000 iterations plus 40,000 burn in. Thinning is set to 80 so 1,600 draws are sampled from the posterior distribution.

After fitting both models, we see that benchmarking increases the agreement between

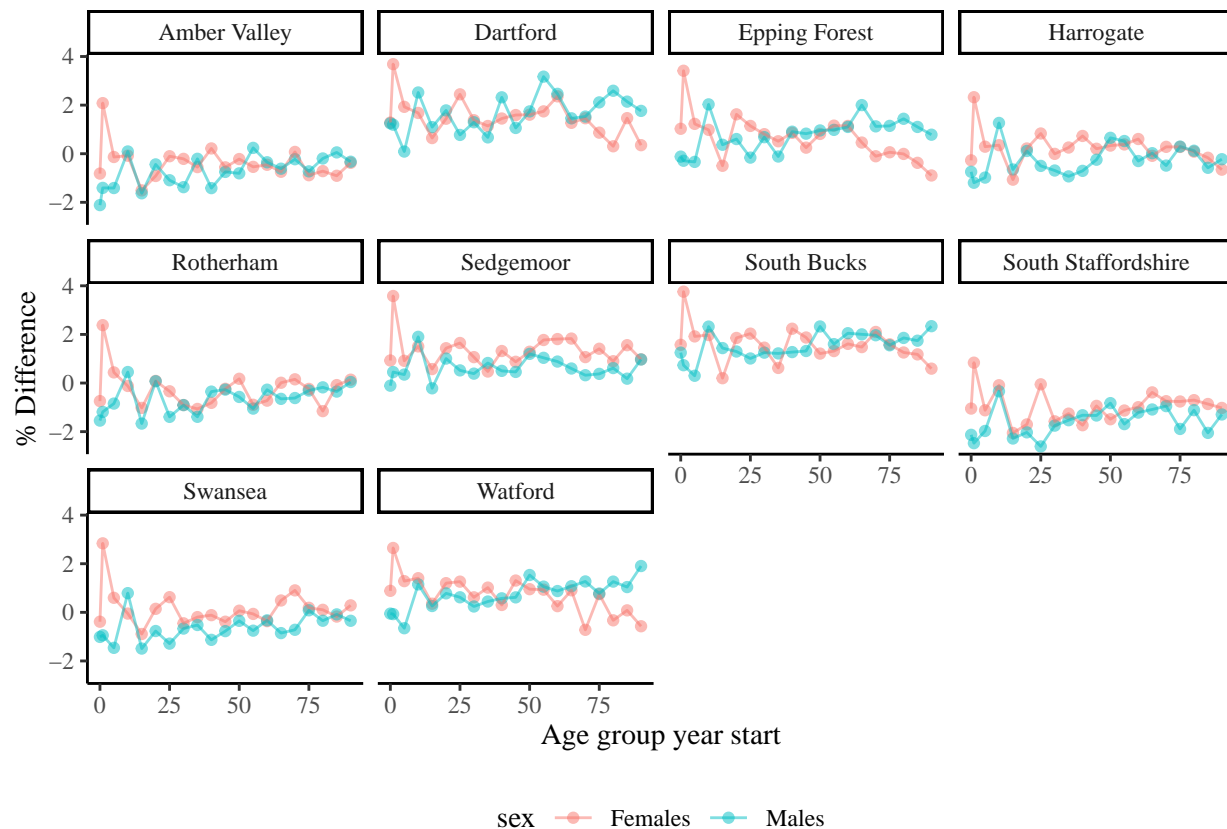
modeled life expectancy at birth and the regional benchmarks, as compared to the non-benchmarked model:



At the LAD level, we can examine the difference between the benchmarked and non-benchmarked models for a sample of LADs:

Percent Difference between Benchmarked and non-Benchmarked Models

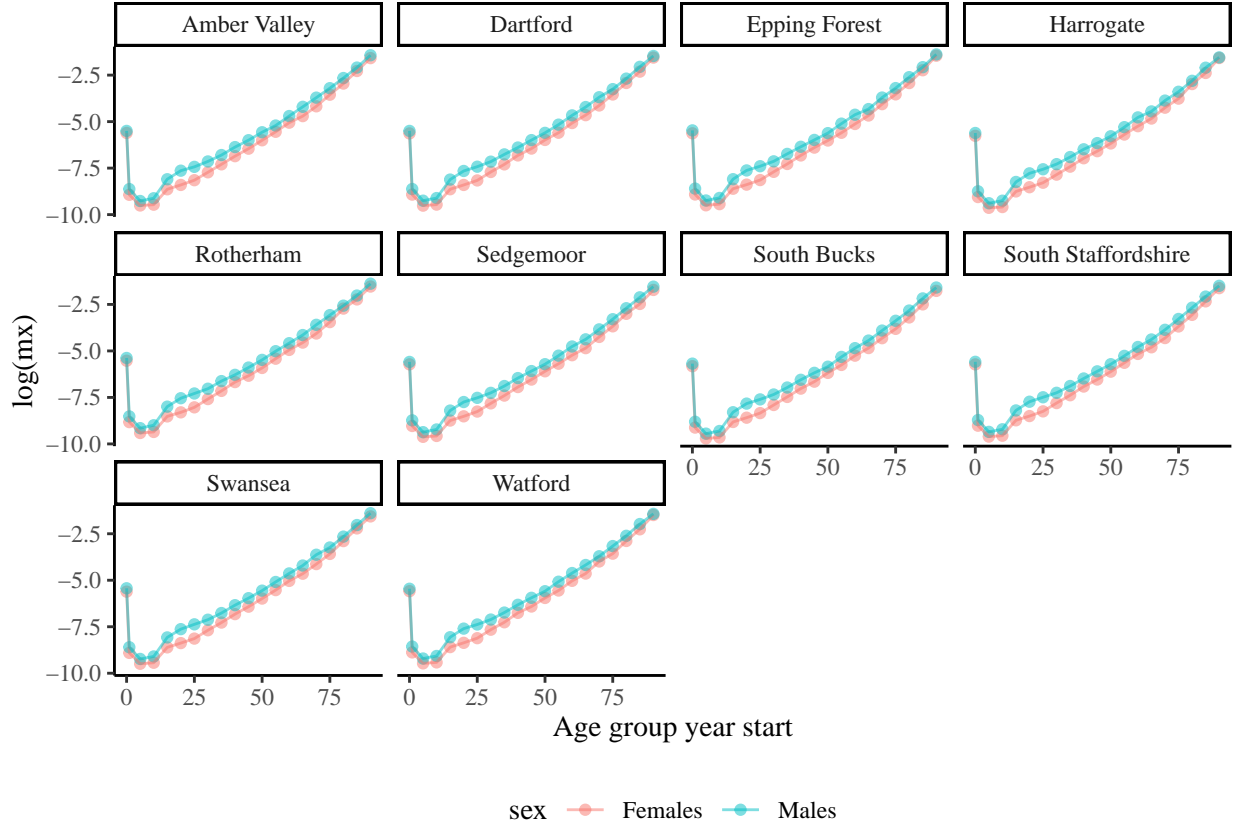
England and Wales



While there are differences between the models, generally that difference stays within 4%. Finally, we look at the modeled mortality rates for the same LADs we included direct estimates of, showing that the model does also has the effect of producing smoother mortality trends.

Log Mortality Rate vs Age

Benchmark model estimates, England and Wales



3.2 Extension

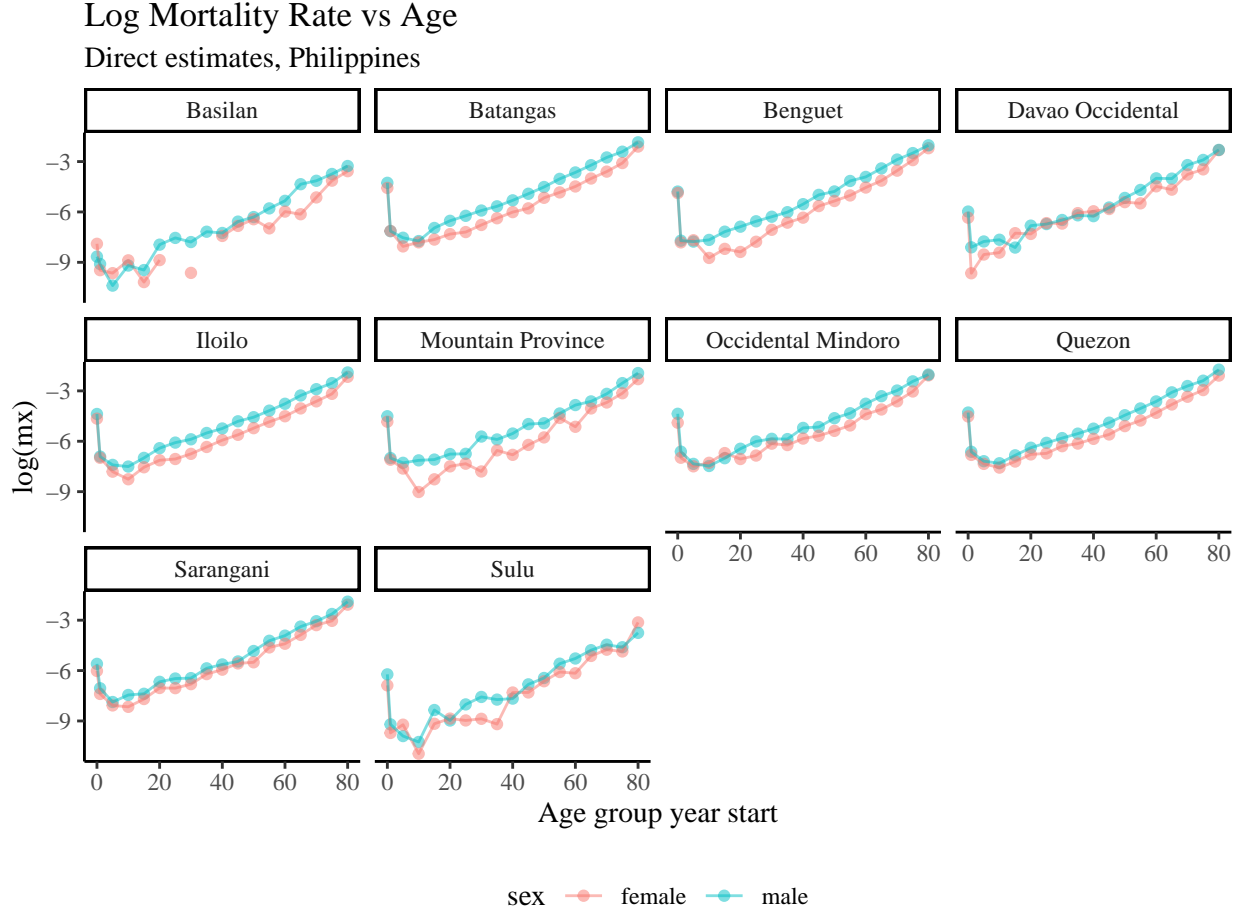
To further evaluate the authors' methods, I applied their same analysis to estimate age-sex-specific mortality rates in 2015 for Philippine provinces, using as benchmarks sex-specific life expectancies at birth for regions. The data for this analysis is death counts and populations at risk for 18 age groups 0-80+, each sex, and 81 provinces.

Table 2: Summary statistics for Philippines Provinces

variable	n	missing	min	25%	median	mean	75%	max
deaths	81	22	0	24.00	69.0	167.51	192.00	3180
population	81	0	103	6228.75	16970.5	30214.64	38628.75	262293

There are 4.84765×10^5 total deaths among a total population of 8.8105903×10^7 . We can

directly estimate mortality rates for each province, five of which are presented here:



Similar to the authors' analysis, I let y_{asp} represent observed death counts at each age (a), sex (s), and province (p). With γ_{asp} as the true underlying mortality rate and w_{asp} the corresponding population at risk, I define the model:

$$y_{asp} \sim \text{Poisson}(w_{asp} \gamma_{asp}) \log(\gamma_{asp}) \sim N(\beta^0 + \beta_a^{\text{age}} + \beta_s^{\text{sex}} + \beta_p^{\text{prv}} + \beta_{as}^{\text{age:sex}}, \sigma^2)$$

Age effects are assumed to follow a random walk analogous to the original analysis, with other effects and standard errors also being equivalent.

The mortality estimates are again benchmarked to regional sex-specific life expectancies at birth for the 16 regions encompassing all of the provinces. A benchmarking function f_{life} is defined as:

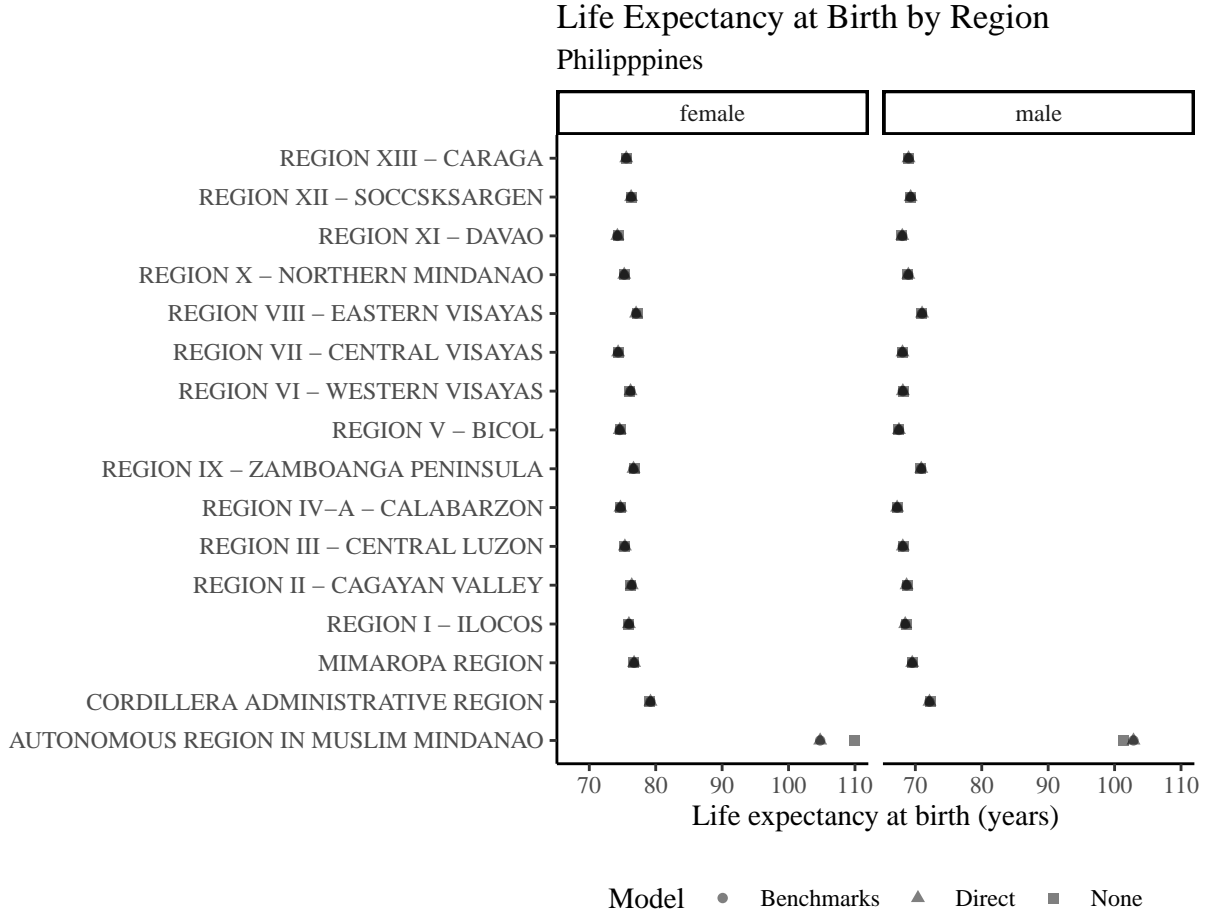
$$m_{sr} = f_{\text{life}}(z_{1sr}, \dots, z_{Asr})$$

With $A = 18$ age groups and z_{asr} age-sex-region morality rates. Life expectancies are normally reported to two decimal places, so the I again incorporate inexact benchmarking allowing a discrepancy of 0.01:

$$m_{\text{sr}} \sim N(\psi_{\text{sr}}, 0.005^2)$$

I also fit this model twice, with and without benchmarking, using four independent chains with 40,000 iterations plus 40,000 burn in. Thinning is set to 80 so 1,600 draws are sampled from the posterior distribution.

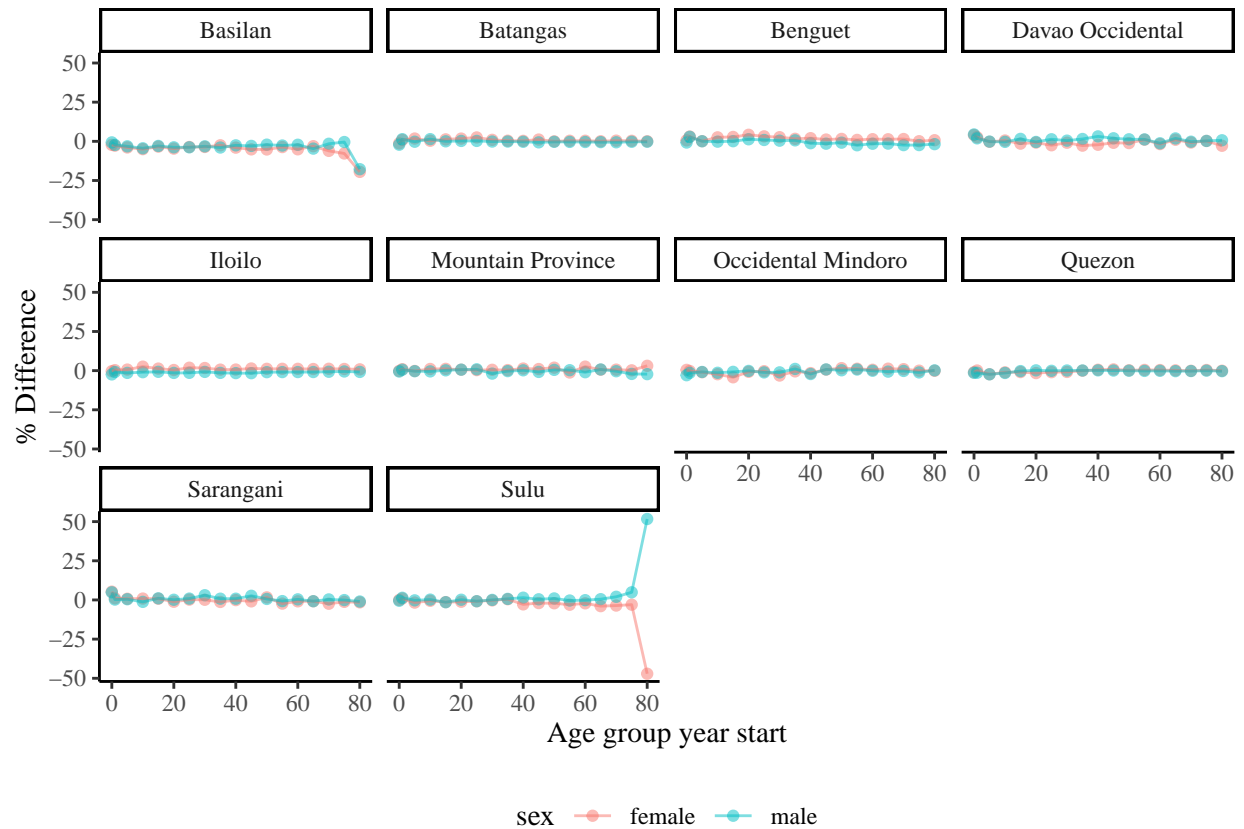
After fitting both models, we see that benchmarking has almost no effect on the agreement between modeled life expectancy at birth and the regional benchmarks, as compared to the non-benchmarked model, except for in the *Autonomous Region in Muslim Mindanao*:



Comparing the models at the province level also shows almost no difference between the benchmarked and non benchmarked model, except for a few outlier points:

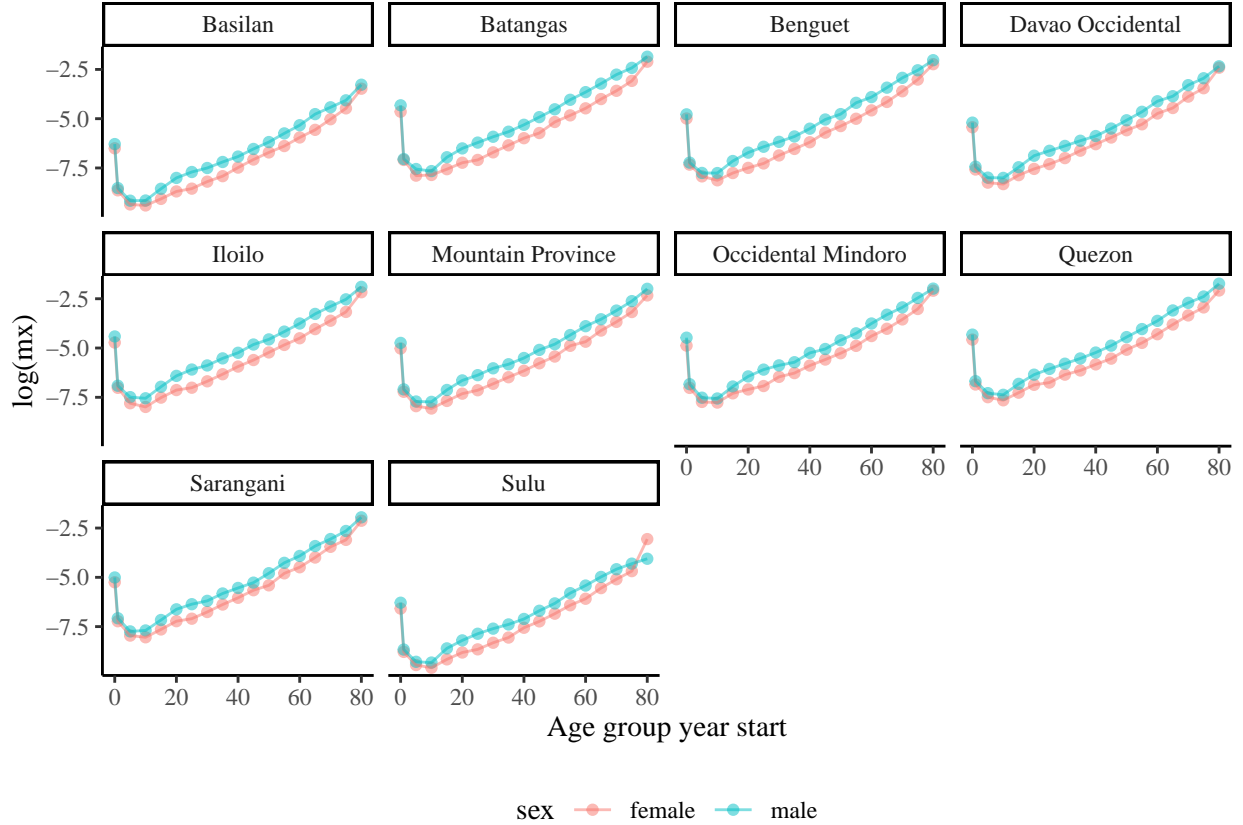
Percent Difference between Benchmarked and non-Benchmarked Models

Philippines



Though the directly estimated mortality rates were smoother than that of England and Wales, the model still does smooth the trend more.

Log Mortality Rate vs Age
Benchmark model estimates, Philippines



4 Discussion

Between these two evaluations, there were some noticeable difference. The England and Wales analysis used almost 3 times as many geographic small areas and 2 more age groups, and a smaller population size than the Philippines analysis. I think this contributed to Philippines having smoother direct estimates than England and Wales, and the general lack of difference in performance between the benchmarked and non-benchmarked Philippines model.

Despite these difference, the methods presented by the authors did behave as expected to produce small area estimates that agreed at a higher level, and were able to produce smoother trends more indicative of the likely underlying mortality parameters.

One general limitation of their methodology is that areas that track data on smaller levels are more likely to have robust data collection systems, reducing the need for benchmarking in

the first place. These methods seem ideal in the case where a location has poorer collection systems, but still collects data at a more granular level.

5 Appendix

5.1 References

Zhang, J.L. and Bryant, J. (2020). [Fully Bayesian Benchmarking of Small Area Estimation Models](#). *Journal of Official Statistics* 36:197-223

Philippines Statistics Authority. [Philippines Vital Statistics System - Deaths 2015](#).

5.2 Code

5.2.1 Report

```
# Prep work -----

# Load libraries
library(dplyr)
library(tidyr)
library(dembase)
library(ggplot2)

# Data
eng_conc <- readRDS("data/britmort/conc.rds") %>%
  as_tibble() %>%
  rename(district = lad, region = rgn) %>%
  filter(!district %in% c("Cornwall", "Isles of Scilly"))

eng_deaths <- readRDS("data/britmort/deaths.rds") %>% as_tibble()
eng_pop <- readRDS("data/britmort/popn.rds") %>% as_tibble()
eng_combined <- eng_pop %>%
  left_join(eng_deaths, by = c("age", "sex", "region")) %>%
  rename(population = count.x, deaths = count.y, district = region) %>%
  mutate(mx = deaths / population) %>%
  right_join(eng_conc, by = "district") %>%
  pivot_longer(
    cols = population:mx,
```



```

    names_to = "variable",
    values_to = "count"
  ) %>%
  extract(age, into = "age_start", remove = FALSE, convert = TRUE)

phl_combined <-
  readRDS("data/combined/phl_2015_all.RDS") %>%
  select(-age_group_years_start) %>%
  rename(age = age_group_name) %>%
  mutate(mx = deaths / population) %>%
  pivot_longer(
    cols = deaths:mx,
    names_to = "variable",
    values_to = "count"
  ) %>%
  extract(age, into = "age_start", remove = FALSE, convert = TRUE)

eng_modeled <- readRDS("data/britmort/mx_modelled.rds") %>%
  collapseIterations(FUN = median) %>%
  as_tibble() %>%
  extract(age, into = "age_start", remove = FALSE, convert = TRUE) %>%
  pivot_wider(names_from = variant, values_from = value) %>%
  mutate(pct_diff = 100 * (None / Benchmarks - 1)) %>%
  pivot_longer(
    cols = None:pct_diff,
    names_to = "variant",
    values_to = "value"
  )

phl_modeled <- readRDS("data/results/model_mx_both.RDS") %>%
  collapseIterations(FUN = median) %>%
  as_tibble() %>%
  extract(age, into = "age_start", remove = FALSE, convert = TRUE) %>%
  pivot_wider(names_from = variant, values_from = value) %>%
  mutate(pct_diff = 100 * (None / Benchmarks - 1)) %>%
  pivot_longer(

```

```

    cols = None:pct_diff,
    names_to = "variant",
    values_to = "value"
  )

eng_ex_compare <- readRDS("data/britmort/life_exp_compare.RDS")
phl_ex_compare <- readRDS("data/results/region_ex_both.RDS")

# Set random samples
set.seed(6789)
sample_size <- 10

eng_name_sample <- sample(unique(eng_combined$district), sample_size)
phl_name_sample <- sample(unique(phl_combined$province), sample_size)

# Helper functions -----

plot_mx <- function(data, ...) {
  ggplot(data, aes(...)) +
    geom_point(alpha = .5) +
    geom_line(alpha = .5) +
    theme_classic() +
    theme(
      text = element_text(family = "serif"),
      legend.position = "bottom"
    )
}

plot_ex <- function(data, ...) {
  ggplot(data, aes(...)) +
    geom_point(alpha = .5) +
    theme_classic() +
    theme(
      text = element_text(family = "serif"),
      legend.position = "bottom"
    )
}

```

```

}

eng_summary_tbl <- eng_combined %>%
  filter(variable != "mx") %>%
  group_by(variable) %>%
  summarise(
    n = length(unique(district)),
    missing = sum(is.na(count)),
    min = min(count),
    `25%` = quantile(count, .25),
    median = median(count),
    mean = mean(count),
    `75%` = quantile(count, .75),
    max = max(count)
  )

eng_n_dis <- length(unique(eng_combined$district))
eng_n_reg <- length(unique(eng_combined$region))

eng_total_pop <- eng_combined %>%
  filter(variable == "population") %>%
  pull(count) %>%
  sum()

eng_total_deaths <- eng_combined %>%
  filter(variable == "deaths") %>%
  pull(count) %>%
  sum()

knitr::kable(
  eng_summary_tbl, booktabs = TRUE, digits = 2,
  caption = "Summary statistics for England and Wales LADs"
)

eng_combined %>%
  filter(
    variable == "mx",

```

```

    district %in% eng_name_sample
  ) %>%
plot_mx(x = age_start, y = log(count), color = sex) +
facet_wrap(vars(district)) +
labs(
  title = "Log Mortality Rate vs Age",
  subtitle = "Direct estimates, England and Wales",
  x = "Age group year start",
  y = "log(mx)"
)
eng_ex_compare %>%
plot_ex(x = value, y = factor(region), shape = variant) +
facet_wrap(vars(sex)) +
labs(
  title = "Life Expectancy at Birth by Region",
  subtitle = "England and Wales",
  x = "Life expectancy at birth (years)",
  y = "",
  shape = "Model"
)
eng_modeled %>%
filter(
  variant == "pct_diff",
  region %in% eng_name_sample
) %>%
plot_mx(x = age_start, y = value, color = sex) +
facet_wrap(vars(region)) +
labs(
  title = "Percent Difference between Benchmarked and non-Benchmarked Models",
  subtitle = "England and Wales",
  x = "Age group year start",
  y = "% Difference"
)
eng_modeled %>%
filter(
  variant == "Benchmarks",

```

```

    region %in% eng_name_sample
  ) %>%
plot_mx(x = age_start, y = log(value), color = sex) +
facet_wrap(vars(region)) +
labs(
  title = "Log Mortality Rate vs Age",
  subtitle = "Benchmark model estimates, England and Wales",
  x = "Age group year start",
  y = "log(mx)"
)
phl_summary_tbl <- phl_combined %>%
  filter(variable != "mx") %>%
  group_by(variable) %>%
  summarise(
    n = length(unique(province)),
    missing = sum(is.na(count)),
    min = min(count, na.rm = TRUE),
    `25%` = quantile(count, .25, na.rm = TRUE),
    median = median(count, na.rm = TRUE),
    mean = mean(count, na.rm = TRUE),
    `75%` = quantile(count, .75, na.rm = TRUE),
    max = max(count, na.rm = TRUE)
  )

phl_n_prv <- unique(length(phl_combined$province))
phl_n_reg <- unique(length(phl_combined$region))

phl_total_pop <- phl_combined %>%
  filter(variable == "population") %>%
  pull(count) %>%
  sum(na.rm = TRUE)

phl_total_deaths <- phl_combined %>%
  filter(variable == "deaths") %>%
  pull(count) %>%
  sum(na.rm = TRUE)

```

```

knitr::kable(
  phl_summary_tbl, booktabs = TRUE, digits = 2,
  caption = "Summary statistics for Philippines Provinces"
)

phl_combined %>%
  filter(
    variable == "mx",
    province %in% phl_name_sample
  ) %>%
  plot_mx(x = age_start, y = log(count), color = sex) +
  facet_wrap(vars(province)) +
  labs(
    title = "Log Mortality Rate vs Age",
    subtitle = "Direct estimates, Philippines",
    x = "Age group year start",
    y = "log(mx)"
  )

phl_ex_compare %>%
  plot_ex(x = value, y = factor(location), shape = variant) +
  facet_wrap(vars(sex)) +
  labs(
    title = "Life Expectancy at Birth by Region",
    subtitle = "Philippines",
    x = "Life expectancy at birth (years)",
    y = "",
    shape = "Model"
  )

phl_modeled %>%
  filter(
    variant == "pct_diff",
    location %in% phl_name_sample
  ) %>%
  plot_mx(x = age_start, y = value, color = sex) +
  facet_wrap(vars(location)) +

```

```

labs(
  title = "Percent Difference between Benchmarked and non-Benchmarked Models",
  subtitle = "Philippines",
  x = "Age group year start",
  y = "% Difference"
)
phl_modeled %>%
  filter(
    variant == "Benchmarks",
    location %in% phl_name_sample
  ) %>%
  plot_mx(x = age_start, y = log(value), color = sex) +
  facet_wrap(vars(location)) +
  labs(
    title = "Log Mortality Rate vs Age",
    subtitle = "Benchmark model estimates, Philippines",
    x = "Age group year start",
    y = "log(mx)"
  )

```

5.2.2 Philippines Mortality Estimation Analysis

Prepare Data

```

# Load libraries -----

library(methods)
library(dplyr)
library(dembase)
library(demdata)

# Load data -----

combined_data <-
  readRDS("data/combined/phl_2015_all.RDS") %>%

```

```

rename(
  age = age_group_name,
  location = province
)

# Prep deaths -----

death_counts <- combined_data %>%
  select(-population) %>%
  xtabs(deaths ~ age + sex + location, data = .) %>%
  Counts() %>%
  setAgeMin(0)

# Prep population -----

pop_counts <- combined_data %>%
  select(-deaths) %>%
  xtabs(population ~ age + sex + location, data = .) %>%
  Counts() %>%
  setAgeMin(0)

# Define aggregate mapping -----

region_concordance <- combined_data %>%
  select(region, location) %>%
  unique() %>%
  Concordance()

# Save prepped data -----

saveRDS(death_counts, "data/prepped/death_counts.RDS")
saveRDS(pop_counts, "data/prepped/pop_counts.RDS")

```



```
saveRDS(region_concordance, "data/prepped/region_concordance.RDS")
```

Calculate Direct Estimates

```
# Load libraries -----

library(dembase)
library(demlife)

# Load data -----

death_counts      <- readRDS("data/prepped/death_counts.RDS")
pop_counts        <- readRDS("data/prepped/pop_counts.RDS")
region_concordance <- readRDS("data/prepped/region_concordance.RDS")

region_deaths <- collapseCategories(
  death_counts,
  dimension = "location",
  concordance = region_concordance
)

region_pop <- collapseCategories(
  pop_counts,
  dimension = "location",
  concordance = region_concordance
)

# Calculate direct estimates -----

# Province level
province_mx <- death_counts / pop_counts
province_life_table <- LifeTable(province_mx)
province_ex <- lifeExpectancy(province_life_table)
```

```

# Region level
region_mx <- region_deaths / region_pop
region_life_table <- LifeTable(region_mx)
region_ex <- lifeExpectancy(region_life_table)

# Save direct estimates -----

saveRDS(region_ex, "data/results/region_ex_direct.rds")

```

Fit Models

```

# Load libraries -----

library(methods)
library(demest)

# Load data -----

death_counts      <- readRDS("data/prepped/death_counts.RDS")
pop_counts        <- readRDS("data/prepped/pop_counts.RDS")
region_concordance <- readRDS("data/prepped/region_concordance.RDS")
region_ex_direct   <- readRDS("data/results/region_ex_direct.rds")

# Modeling prep -----

set.seed(256)

run_model <- function(model_spec, out_file) {

  estimateModel(
    model = model_spec,
    y = death_counts,
    exposure = pop_counts,

```

```

    filename = out_file,
    nBurnin = 40000,
    nSim = 40000,
    nChain = 4,
    nThin = 80
  )
}

gen_model_spec <- function(agg = NULL) {

  Model(
    y ~ Poisson(mean ~ age * sex + location),
    age ~ DLM(damp = NULL, error = Error(robust = TRUE)),
    aggregate = agg,
    jump = 0.11
  )
}

# No benchmark model -----

model_none <- gen_model_spec(agg = NULL)
model_none_filename <- "data/results/model_none.est"
# run_model(model_none, model_none_filename)

# Benchmarked model -----

aggregate <- AgLife(
  value = region_ex_direct,
  sd = 0.005,
  concordances = list(location = region_concordance)
)

```

```

model_bench <- gen_model_spec(agg = aggregate)
model_bench_filename <- "data/results/model_bench.est"
# run_model(model_bench, model_bench_filename)

# Get model summaries -----

model_none_summary <- fetchSummary(model_none_filename)
model_bench_summary <- fetchSummary(model_bench_filename)

# Save modeled outputs -----

model_none_mx <- fetch(
  filename = model_none_filename,
  where = c("model", "likelihood", "rate")
)

model_bench_mx <- fetch(
  filename = model_bench_filename,
  where = c("model", "likelihood", "rate")
)

model_both_mx <- dbind(
  None = model_none_mx,
  Benchmarks = model_bench_mx,
  along = "variant"
)

saveRDS(model_both_mx, "data/results/model_mx_both.RDS")

```

Compare Regions

```

# Load libraries -----

library(dembase)

```

```

library(demlife)
library(dplyr)

# Load data -----

pop_counts  <- readRDS("data/prepped/pop_counts.RDS")
region_conc <- readRDS("data/prepped/region_concordance.RDS")

region_ex_direct <- readRDS("data/results/region_ex_direct.rds")
model_mx        <- readRDS("data/results/model_mx_both.RDS")

# Get modeled region ex -----

region_ex_model <- model_mx %>%
  collapseCategories(
    dimension = "location",
    concordance = region_conc,
    weights = pop_counts
  ) %>%
  LifeTable() %>%
  lifeExpectancy() %>%
  collapseIterations(FUN = median) %>%
  as_tibble()

# Combine ex -----

region_ex_both <- region_ex_direct %>%
  as_tibble() %>%
  mutate(variant = "Direct") %>%
  bind_rows(region_ex_model)

# Save data -----

```

```
saveRDS(region_ex_both, "data/results/region_ex_both.RDS")
```

5.2.3 England and Wales Mortality Estimation Analysis

The Authors' analysis of age-sex-specific mortality rates for local authority districts in England and Wales, using as benchmarks sex-specific life expectancies for regions, was recreated exactly running the code provided on [GitHub](#), and has thus been omitted from this report.