

# Response to Reviewer 1 Comments

Jon Minton

2022-10-03

## Introduction

This document presents additional analyses performed in response to comments from Referee 1.

## The queries

- 1) It could prove relevant to verify whether the residuals of the proposed models are white noise. Have you made a model residuals analysis?
- 2) In addition to the BIC and AIC information criteria, could you compare the proposed models using also AICc
- 3) In view of the results obtained, would it be possible to use an intervention variable to fit the ARIMA model?

## Query 1: Residuals analysis

A residuals analysis was not conducted, but will be within this appendix. As mentioned elsewhere, we have an understanding that, for many populations, the series are better represented by models where each value in the series is autocorrelated with the previous value. However for simplicity no such term is included in the models being compared. This does mean we may expect residuals which are not entirely white noise, however.

To look at the residuals, we first load and process the dataset:

```
# load packages

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.7      v dplyr  1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

# load data
hmd_lt <- read_rds("https://github.com/JonMinton/change-in-ex/blob/main/data/lifetables.rds?raw=true")

# Labels for codes
country_code_lookup <-
```

```

tribble(
  ~code, ~country,
  "DEUTNP", "Germany",
  "DEUTE", "East Germany",
  "DEUTW", "West Germany",
  "ESP", "Spain",
  "FRATNP", "France",
  "ITA", "Italy",
  "GBRTENW", "England & Wales",
  "GBR_SCO", "Scotland",
  "DEUTSYNTH", "Synthetic Germany",
  "NLD", "Netherlands"
)

countries_of_interest <- c(
  "GBRTENW",
  "GBR_SCO",
  "GBR_UK",
  "FRATNP",
  "ESP",
  "ITA",
  "DEUTNP",
  "DEUTE",
  "DEUTW",
  "NLD"
)

source("https://raw.githubusercontent.com/JonMinton/change-in-ex/main/R/make_synthetic_germany_function.R")
source("https://raw.githubusercontent.com/JonMinton/change-in-ex/main/R/make_pop_selection.R")

change_in_ex_selected_countries <-
  hmd_ex_selected_countries_with_synth %>%
  group_by(code, x, sex) %>%
  arrange(year) %>%
  mutate(delta_ex = ex - lag(ex)) %>%
  ungroup()

```

Then we create and run some convenience functions...

```

estimate_breakpoints_and_pval <- function(df){
  null_md1 <- lm(ex ~ year, data = df)
  seg_md1 <- segmented::segmented(null_md1, seg.Z= ~year, psi = 2010)
  seg2_md1 <- segmented::segmented(null_md1, seg.z= ~year, psi= c(1985, 2010)) # added to test

  list(
    null = null_md1,
    seg = seg_md1,
    seg2 = seg2_md1
  )
}

segmented_breakpoints_models <-

```

```

hmd_ex_selected_countries_with_synth %>%
  filter(code != "DEUTNP") %>%
  filter(year >= 1979) %>%
  group_by(code, x, sex) %>%
  nest() %>%
  mutate(
    mdl_outputs = map(data, estimate_breakpoints_and_pval)
  ) %>%
  unnest_longer(mdl_outputs)

# Now let's get the BIC for each model

make_predictions <- function(mdl, dta = tibble(year = 1980:2020)){
  tibble(
    year = dta$year,
    ex_pred = predict(mdl, newdata = dta)
  )
}

best_model_predictions_descriptions <-
segmented_breakpoints_models %>%
  mutate(bic = map_dbl(mdl_outputs, BIC)) %>%
  group_by(code, x, sex) %>%
  mutate(rank_bic = rank(bic)) %>%
  filter(rank_bic == 1) %>%
  mutate(
    best_model = case_when(
      mdl_outputs_id == 'seg2' ~ "Two breakpoints",
      mdl_outputs_id == 'seg' ~ "One breakpoint",
      mdl_outputs_id == 'null' ~ "No breakpoints"
    )
  ) %>%
  mutate(
    first_breakpoint = map2_dbl(
      mdl_outputs_id, mdl_outputs,
      function(x, y){
        if (x == 'null' ) {NA_real_} else {
          y[["psi"]][1,2]
        }
      }
    ),
    first_breakpoint_se = map2_dbl(
      mdl_outputs_id, mdl_outputs,
      function(x, y){
        if (x == 'null' ) {NA_real_} else {
          y[["psi"]][1,3]
        }
      }
    ),
    second_breakpoint = map2_dbl(
      mdl_outputs_id, mdl_outputs,

```



```
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
```

We can produce predicted values as follows...

```
best_breakpoint_model_predictions <-
  segmented_breakpoints_models %>%
    mutate(bic = map_dbl mdl_outputs, BIC)) %>%
    group_by(code, x, sex) %>%
    filter(bic == min(bic)) %>%
    mutate(predictions = map2(mdl_outputs, data, predict)) %>%
    mutate(data_augmented = map2(data, predictions, ~ .x %>% mutate(prediction = .y))) %>%
    select(code, x, sex, data_augmented) %>%
    ungroup() %>%
    unnest(data_augmented)
```

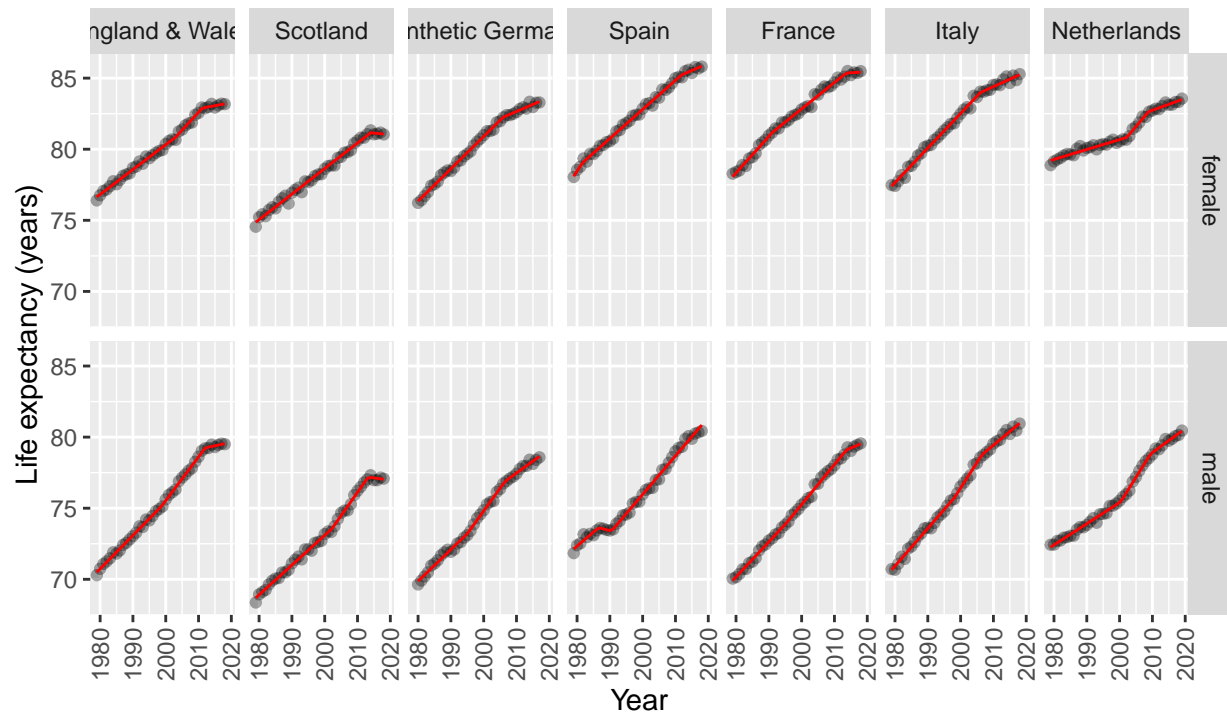
The following shows the predicted values (as lines) and observed values (as points) for x = 0:

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spain"))) %>%
  filter(!is.na(country)) %>%
  filter(x == 0) %>%
  ggplot(aes(x = year)) +
  facet_grid(sex ~ country) +
  geom_point(aes(y = ex), alpha = 0.35) +
  geom_line(aes(y = prediction), colour = "red") +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  ) +
  labs(
    x = "Year",
    y = "Life expectancy (years)",
    title = "Predicted and observed life expectancy at birth, selected countries",
    subtitle = "Line: Best segmented regression model for the country",
    caption = "Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German population"
  )
```

```
## Joining, by = "code"
```

## Predicted and observed life expectancy at birth, selected countries

Line: Best segmented regression model for the country



Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German population weighting

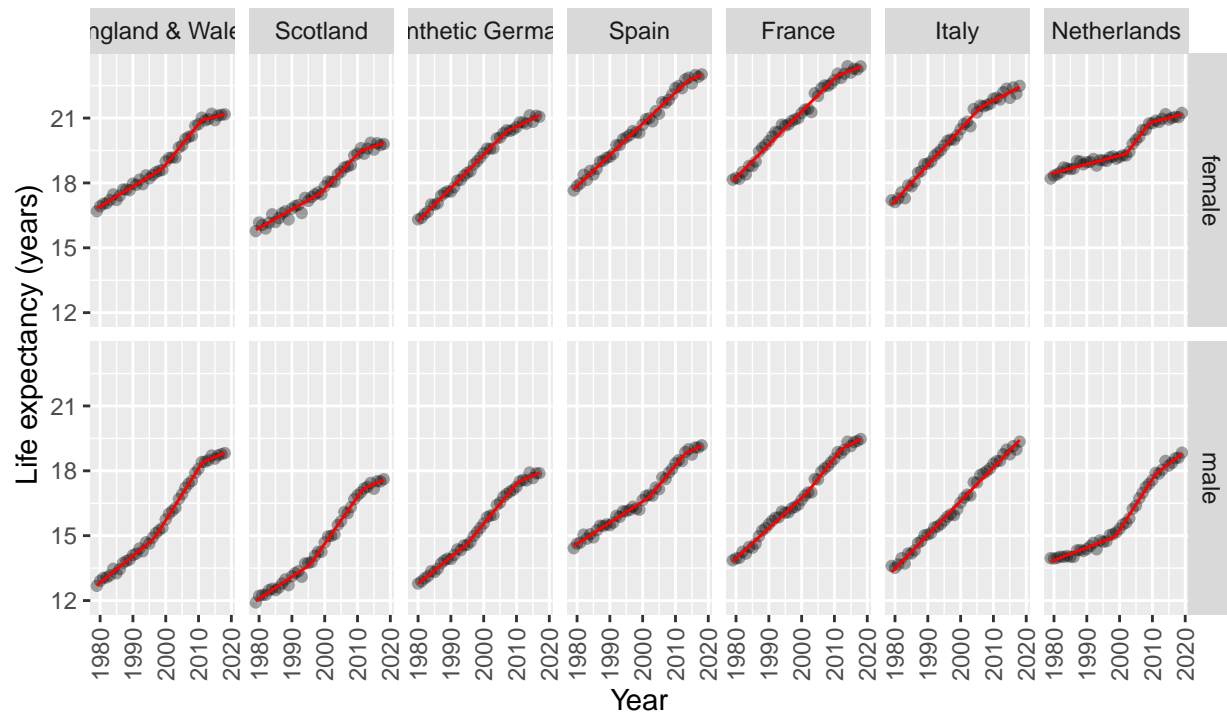
And the following figure shows the same for  $x = 65$ :

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spain", "France", "Italy", "Netherlands"))) %>%
  filter(!is.na(country)) %>%
  filter(x == 65) %>%
  ggplot(aes(x = year)) +
  facet_grid(sex ~ country) +
  geom_point(aes(y = ex), alpha = 0.35) +
  geom_line(aes(y = prediction), colour = "red") +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  ) +
  labs(
    x = "Year",
    y = "Life expectancy (years)",
    title = "Predicted and observed life expectancy at age 65, selected countries",
    subtitle = "Line: Best segmented regression model for the country",
    caption = "Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German population weighting"
  )
```

```
## Joining, by = "code"
```

## Predicted and observed life expectancy at age 65, selected countries

Line: Best segmented regression model for the country



Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German population weighting

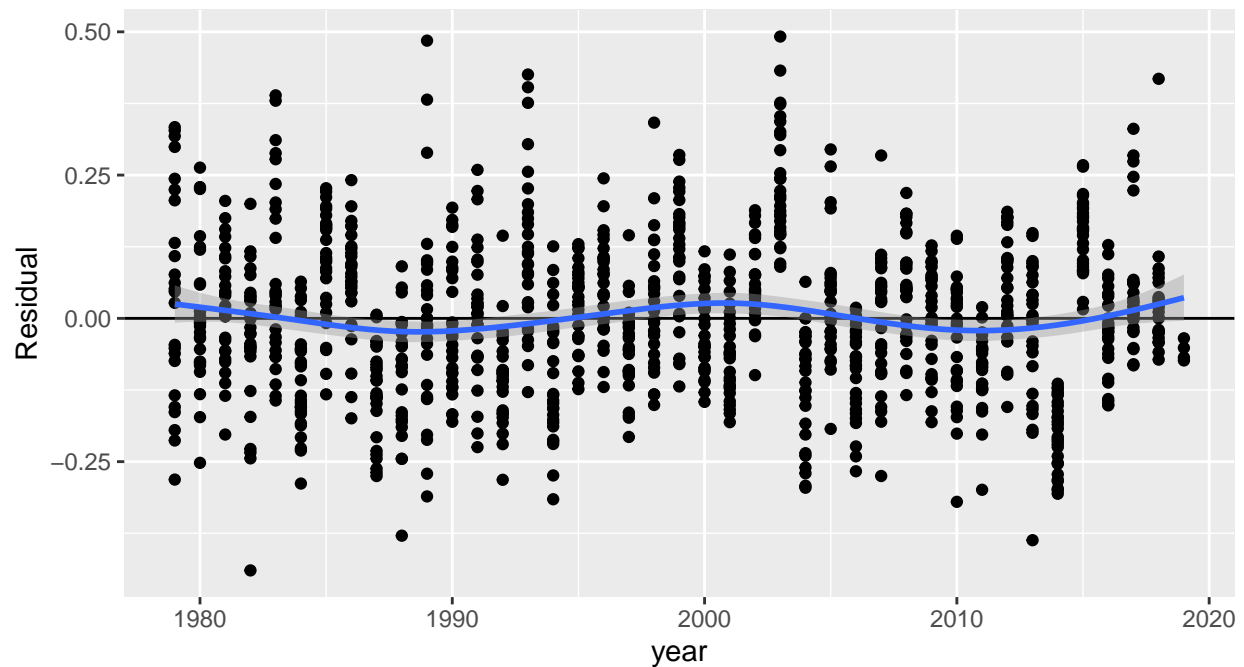
The residuals for all populations (ages, sexes, countries) combined is shown in the following figure:

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spain", "France", "Italy", "Netherlands"))) %>%
  filter(!is.na(country)) %>%
  mutate(
    residual = prediction - ex
  ) %>%
  ggplot(aes(year, residual)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  stat_smooth() +
  labs(
    x = "year", y = "Residual",
    title = "Difference between observed and predicted life expectancies",
    subtitle = "Both starting ages (0 and 65), both sexes and all countries",
    caption = "Predictions based on best-performing 0, 1 or 2 breakpoint models as assessed using BIC\nA GAM smoother line has been added to indicate any systematic change in residuals over time"
  )

## Joining, by = "code"
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

## Difference between observed and predicted life expectancies

Both starting ages (0 and 65), both sexes and all countries



Predictions based on best-performing 0, 1 or 2 breakpoint models as assessed using BIC

A GAM smoother line has been added to indicate any systematic change in residuals over time

And the following figure shows the residuals separated by starting age:

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spain")))
  filter(!is.na(country)) %>%
  mutate(
    residual = prediction - ex
  ) %>%
  ggplot(aes(year, residual)) +
  geom_point() +
  facet_wrap(~x) +
  geom_hline(yintercept = 0) +
  stat_smooth() +
  labs(
    x = "year", y = "Residual",
    title = "Difference between observed and predicted life expectancies",
    subtitle = "Separated by starting age (0 and 65), both sexes and all countries",
    caption = "Predictions based on best-performing 0, 1 or 2 breakpoint models as assessed using BIC\n
    A GAM smoother line has been added to indicate any systematic change in residuals over time"
  )
```

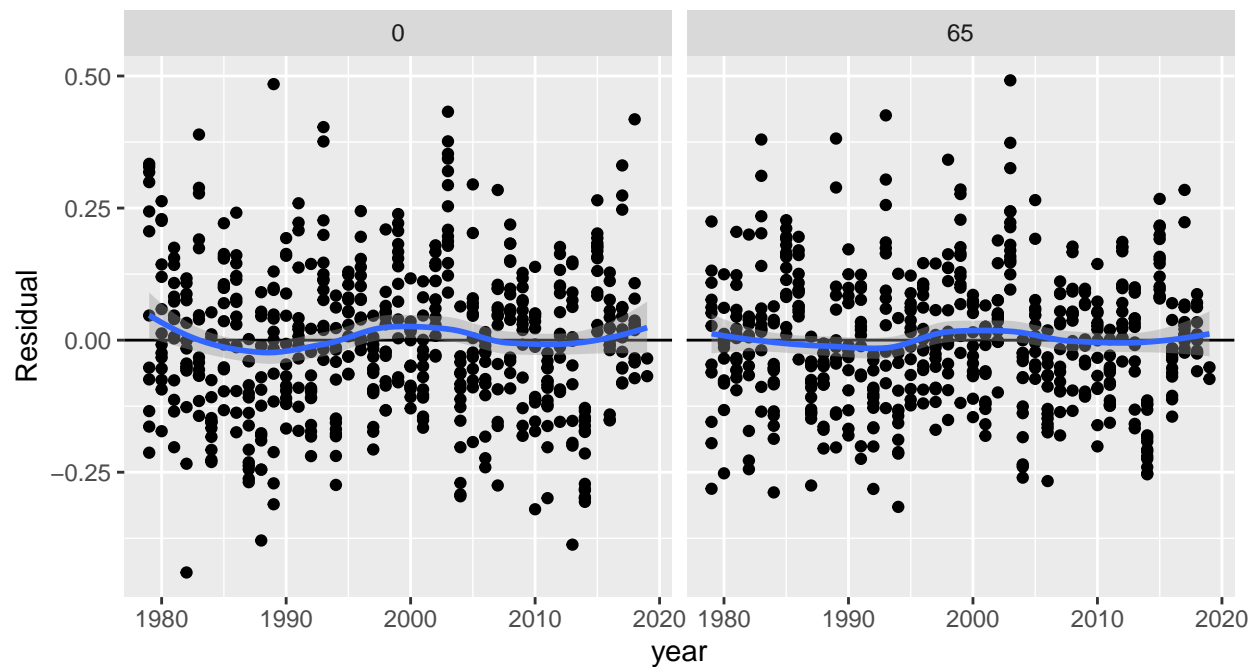
```
## Joining, by = "code"
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



## Difference between observed and predicted life expectancies

Separated by starting age (0 and 65), both sexes and all countries



Predictions based on best-performing 0, 1 or 2 breakpoint models as assessed using BIC

A GAM smoother line has been added to indicate any systematic change in residuals over time

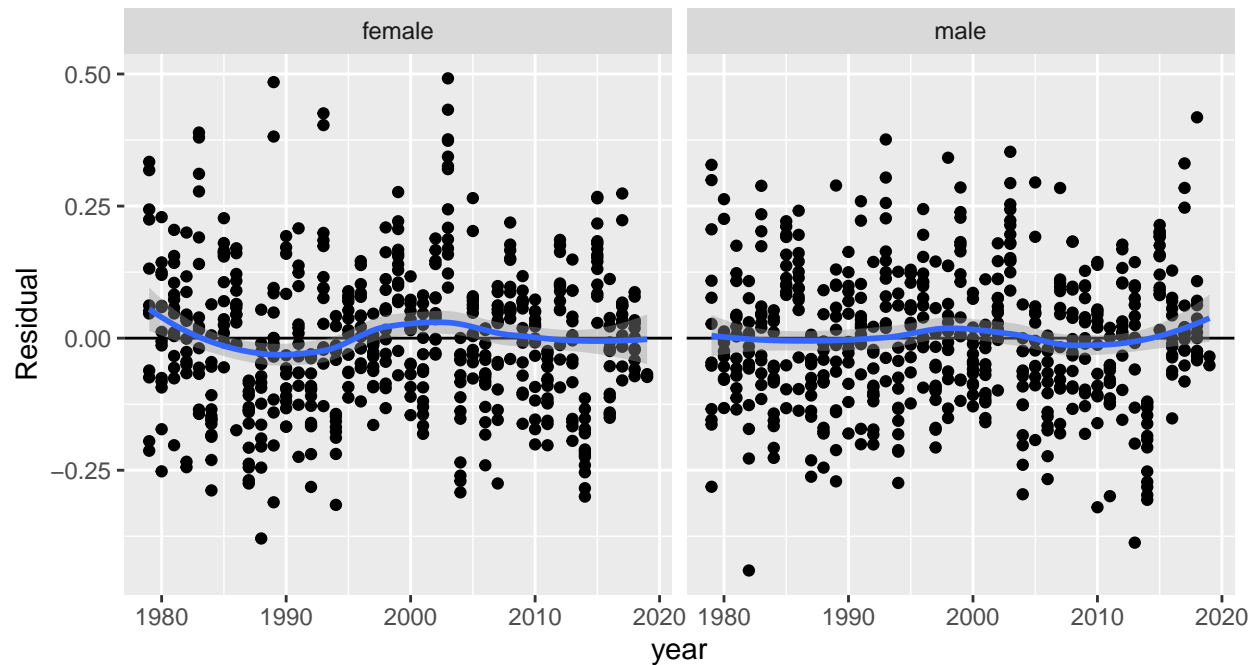
The following shows residuals separately by sex:

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spain")))
  filter(!is.na(country)) %>%
  mutate(
    residual = prediction - ex
  ) %>%
  ggplot(aes(year, residual)) +
  geom_point() +
  facet_wrap(~sex) +
  geom_hline(yintercept = 0) +
  stat_smooth() +
  labs(
    x = "year", y = "Residual",
    title = "Difference between observed and predicted life expectancies",
    subtitle = "Separated by sex, both starting ages and all countries",
    caption = "Predictions based on best-performing 0, 1 or 2 breakpoint models as assessed using BIC\n
    A GAM smoother line has been added to indicate any systematic change in residuals over time"
  )
```

```
## Joining, by = "code"
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

## Difference between observed and predicted life expectancies Separated by sex, both starting ages and all countries



Predictions based on best-performing 0, 1 or 2 breakpoint models as assessed using BIC

A GAM smoother line has been added to indicate any systematic change in residuals over time

## Discussion

Though there are some years where residuals tend to be very high or very low, these tend to be consistent across all countries, despite differences in the preferred model specification for each country. Key examples include 2003 and 2004, and 2014 and 2015. Both pairs of adjacent year comprise one year with systematically high (or low) residuals, followed by another with systematically low (or high) residuals. These are likely to reflect the somewhat oscillatory behaviour of annual mortality, likely driven by phenomena like forward mortality displacement.

## Query 2: Model estimation by AICc

As the reviewer notes, the models and model selection processes make use of BIC to select from a series of non-nested models. Other information criteria could also be used. The reviewer asks about AICc, a variation of AIC adapted for small sample sizes.

BIC is currently used to select between a zero breakpoint model, the best performing one breakpoint model, and the best performing two breakpoint model.

The identification of the best performing one and two breakpoint models is performed within the `segmented::segmented` function, which selects the model which minimises minus log likelihood (See the `fn.obj` argument definition in the `segmented::seg.control` help file). As the number of parameters and observations will not vary when selecting between one and two breakpoint model specifications for a single population, there does not appear to be any advantage to rerunning this stage with a different objective function.

This means AICc should replace BIC as follows. AICc has been calculated using the `AICcmodavg::AICc` package and function.

```
# Now let's get the AICc for each model
```

```
compare_aicc_bic <-
  segmented_breakpoints_models %>%
    mutate(bic = map_dbl mdl_outputs, BIC)) %>%
    mutate(aicc = map_dbl mdl_outputs, AICcmodavg::AICc)) %>%
    group_by(code, x, sex) %>%
    mutate(rank_bic = rank(bic)) %>%
    mutate(rank_aicc = rank(aicc)) %>%
    ungroup() %>%
    mutate(both_equal = rank_bic == rank_aicc)
```

```
compare_aicc_bic
```

```
## # A tibble: 108 x 11
##   code      x sex  data    mdl_outputs mdl_outputs_id    bic  aicc rank_bic
##   <chr> <dbl> <chr> <list> <named list> <chr>      <dbl> <dbl>    <dbl>
## 1 DEUTE    0 female <tibble> <lm>      null         74.3  70.0      3
## 2 DEUTE    0 female <tibble> <segmentd> seg         53.2  46.7      2
## 3 DEUTE    0 female <tibble> <segmentd> seg2        -12.1 -20.1      1
## 4 DEUTE   65 female <tibble> <lm>      null         42.1  37.8      3
## 5 DEUTE   65 female <tibble> <segmentd> seg         31.4  24.9      2
## 6 DEUTE   65 female <tibble> <segmentd> seg2        -36.7 -44.7      1
## 7 DEUTW    0 female <tibble> <lm>      null         28.4  24.1      3
## 8 DEUTW    0 female <tibble> <segmentd> seg         -9.01 -15.5      2
## 9 DEUTW    0 female <tibble> <segmentd> seg2        -35.4 -43.5      1
## 10 DEUTW   65 female <tibble> <lm>      null        -15.6 -19.9      3
## # ... with 98 more rows, and 2 more variables: rank_aicc <dbl>,
## #   both_equal <lgl>
```

The rank order of AICc and BIC is identical for 104 of the 108 rows above. The differences are as follows:

```
compare_aicc_bic %>%
  filter(!both_equal)
```

```
## # A tibble: 4 x 11
##   code      x sex  data    mdl_outputs mdl_outputs_id    bic  aicc rank_bic
##   <chr> <dbl> <chr> <list> <named lis> <chr>      <dbl> <dbl>    <dbl>
## 1 ESP     65 fema~ <tibble> <lm>      null        -28.2 -32.6      2
## 2 ESP     65 fema~ <tibble> <segmentd> seg2        -25.9 -34.2      3
## 3 DEUTSYNTH 0 fema~ <tibble> <segmentd> seg         -36.0 -42.3      1
## 4 DEUTSYNTH 0 fema~ <tibble> <segmentd> seg2        -34.8 -42.5      2
## # ... with 2 more variables: rank_aicc <dbl>, both_equal <lgl>
```

For Spain, x=65, females: BIC ranked the 0 breakpoint model 2nd, AICc ranked it 3rd; BIC ranked the 2 breakpoint model 3rd. For Germany, x = 0, females: BIC ranked the 1 breakpoint model 1st, AICc ranked it 2nd; BIC ranked the 2 breakpoint model second, AICc ranked it 1st.

## Discussion

Adopting AICc in place of BIC would only have led to one difference in terms of the ‘best performing model’, out of the 36 datasets (comprising country, sex and starting age combination) for which the three classes of model were each estimated and compared. This was for Germany, for life expectancy from birth (x = 0), and for females. For this population AICc indicated a two breakpoint model be preferred; by contrast BIC suggested the one breakpoint model should be preferred.

### Query 3: Intervention variable to fit the ARIMA model

ARIMA models were not used in the analyses presented in the main manuscript, although they are used within one of the appendices.

Both the `forecast::Arima` function and the `stats::arima` function include an argument `xreg`, for ‘external regressors’. This means intervention or control variables can be included in ARIMA model specifications. We know from appendix X that ARIMA(1, 1, 0) models (with autoregression) are often preferred to ARIMA(0, 1, 0) (no autoregression) specifications for many populations. We also know from the response to the first reviewer query that the residuals for the years 2003 and 2004, and 2014 and 2015, in particular, appear somewhat incompatible with a white noise error term assumption. Given both of those observations, an ARIMA(1, 1, 0) model specification with at least one external regressor may be reasonable. Examples of such regressors may include number of days where temperatures were either above or below optimal ranges.