Confidence Intervals for loess fits and linear/segmented regression models

2022-10-18

Aim

This appendix shows the 95% confidence intervals for the loess and linear/segmented models presented in the main paper.

Data preparation

```
# 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()
```

```
library(glue)

# 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"
  )
```

```r
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

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()
```

LOESS smoother, confidence intervals

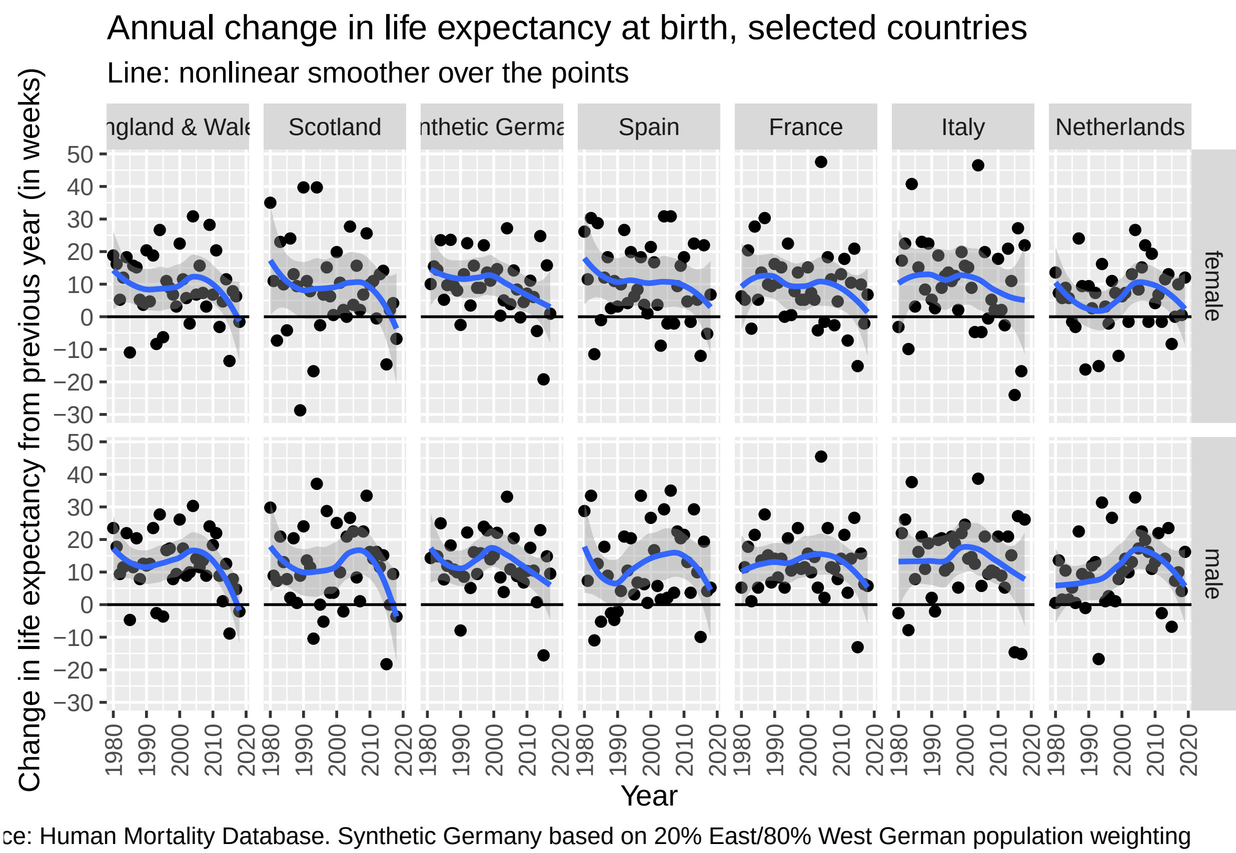
The following shows the LOESS smoother line with confidence intervals for annual changes in life expectancy at birth:

```r
p_0_loess <- change_in_ex_selected_countries %>%
  filter(x == 0) %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  filter(!is.na(country)) %>%
  filter(between(year, 1980, 2020)) %>%
  mutate(delta_ex = delta_ex * 52.25) %>% # Convert to weeks
  ggplot(aes(x = year, y = delta_ex)) +
  geom_point() +
  stat_smooth(se = TRUE) +  # Changed from se = FALSE
  facet_grid(sex~country) +
  geom_hline(yintercept = 0) +
  scale_y_continuous(breaks = seq(-30, 50, by = 10)) +
  theme(
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  ) +
  labs(
    x = "Year",
    y = "Change in life expectancy from previous year (in weeks)",
    title = "Annual change in life expectancy at birth, selected countries",
    subtitle = "Line: nonlinear smoother over the points",
    caption =  "Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German p
  )
```

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

```
p_0_loess
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
## Warning: Removed 2 rows containing missing values (geom_point).
```



And the following shows the same for life expectancy from age 65 years

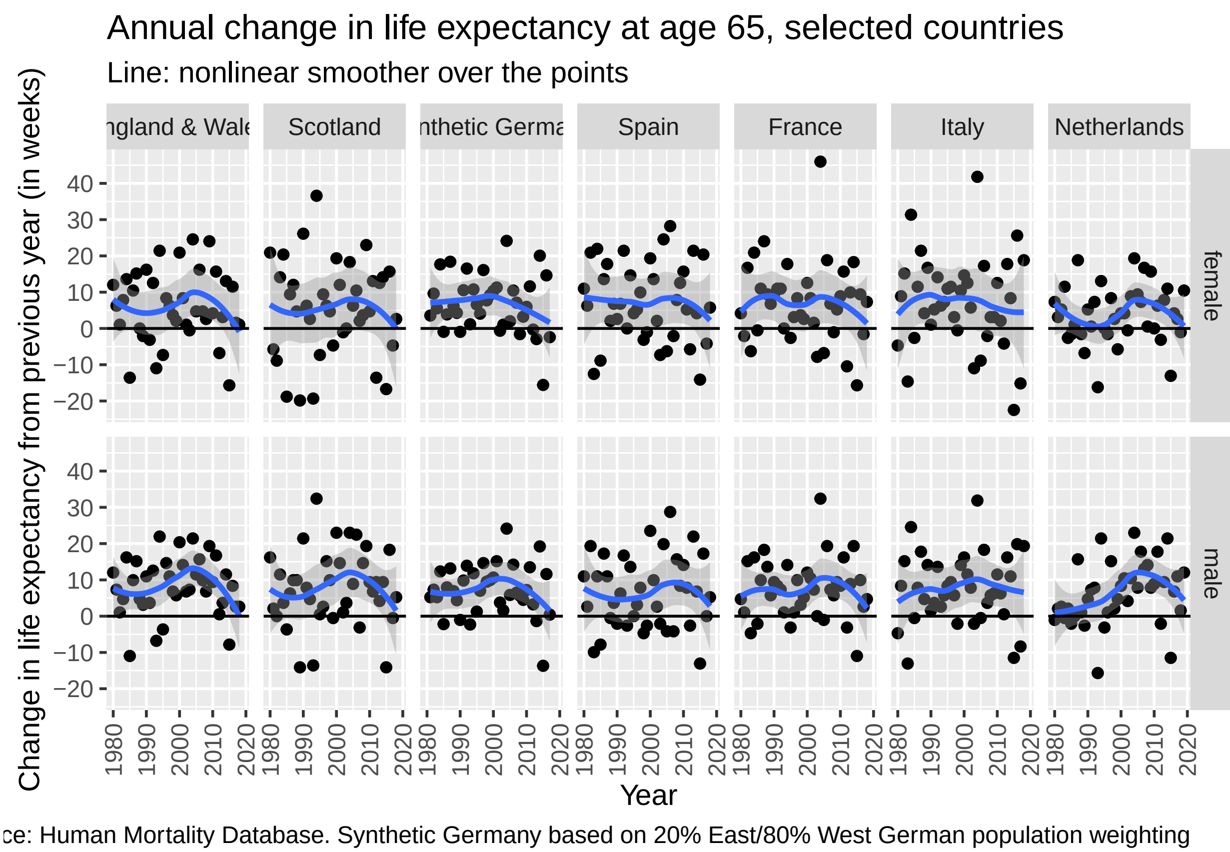
```
p_65_loess <- change_in_ex_selected_countries %>%
  filter(x == 65) %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  filter(!is.na(country)) %>%
  filter(between(year, 1980, 2020)) %>%
  mutate(delta_ex = delta_ex * 52.25) %>% # Convert to weeks
  ggplot(aes(x = year, y = delta_ex)) +
  geom_point() +
  stat_smooth(se = TRUE) + # changed from se = FALSE
  facet_grid(sex~country) +
  geom_hline(yintercept = 0) +
  scale_y_continuous(breaks = seq(-30, 50, by = 10)) +
  theme(
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  ) +
  labs(
```

```
    x = "Year",
    y = "Change in life expectancy from previous year (in weeks)",
    title = "Annual change in life expectancy at age 65, selected countries",
    subtitle = "Line: nonlinear smoother over the points",
    caption =  "Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German po
  )
```

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

```
p_65_loess
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
## Warning: Removed 2 rows containing missing values (geom_point).
```



We can see that the 95% CI on the smoother includes values below zero for a number of populations.

Within the above, and in the main paper, the LOESS smoother is called in ggplot2's `stat_smooth` function, which uses slightly different parameterisation than if the `loess` function in the `stats` packages were called directly. For consistency with the main paper we can extract the point estimate, lower and upper CIs, and standard error, directly from the above ggplot objects. The following function does this:

```
extract_loess_ci <- function(p){
  b <- ggplot_build(p)
  panel_lookup <- b$layout$layout

  # get smoother values
```

```
  p_loess <- b$data[[2]]

  # Join and simplify

  sm <- p_loess %>%
    select(x, y, ymin, ymax, se, PANEL) %>%
    left_join(panel_lookup)

  # find last year for each country

  max_years <- p$data %>%
    group_by(country, sex) %>%
    filter(year == max(year)) %>%
    select(country, sex, max_year = year)

  sm %>%
    select(x, y, ymin, ymax, se, sex, country) %>%
    left_join(max_years) %>%
    filter(x == max_year)
}
```

And the following code applies the function to the two previously created ggplot2 objects

```
loess_lastyear_x0 <-
extract_loess_ci(p_0_loess) %>%
  rename(year = x, y_point = y,
         y_lower = ymin, y_upper = ymax, y_se = se) %>%
  mutate(x = 0)  %>% # x now starting age
  select(year, x, sex, country, everything(), -max_year)
```

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

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Joining, by = "PANEL"

## Joining, by = c("sex", "country")
```

```
loess_lastyear_x65 <-
extract_loess_ci(p_65_loess) %>%
  rename(year = x, y_point = y,
         y_lower = ymin, y_upper = ymax, y_se = se) %>%
  mutate(x = 65)  %>% # x now starting age
  select(year, x, sex, country, everything(), -max_year)
```

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

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Joining, by = "PANEL"
## Joining, by = c("sex", "country")
```

```
loess_lastyear_both <-
  bind_rows(loess_lastyear_x0, loess_lastyear_x65)

loess_lastyear_both
```

```
##    year  x    sex           country     y_point    y_lower   y_upper     y_se
## 1  2018  0 female   England & Wales -1.37085831 -13.530072 10.788356 5.983454
```

```
## 2  2018  0 female          Scotland -3.61707693 -20.509289 13.275135 8.312525
## 3  2017  0 female Synthetic Germany  2.96747333  -8.062811 13.997757 5.414911
## 4  2018  0 female             Spain  2.94139772 -11.305593 17.188389 7.010833
## 5  2018  0 female            France  1.37863909 -12.054318 14.811596 6.610253
## 6  2018  0 female             Italy  5.10750796 -11.526141 21.741157 8.185288
## 7  2019  0 female       Netherlands  2.36725240  -8.638790 13.373295 5.422197
## 8  2018  0   male   England & Wales -1.77259987 -11.851773  8.306573 4.959882
## 9  2018  0   male          Scotland -3.64395051 -17.511375 10.223474 6.824051
## 10 2017  0   male Synthetic Germany  5.96684307  -4.694271 16.627957 5.233681
## 11 2018  0   male             Spain  4.51381577  -9.686210 18.713842 6.987722
## 12 2018  0   male            France  5.32872389  -5.947412 16.604860 5.548898
## 13 2018  0   male             Italy  7.76595659  -6.604990 22.136904 7.071830
## 14 2019  0   male       Netherlands  5.69252174  -5.722116 17.107159 5.623494
## 15 2018 65 female   England & Wales -0.90447135 -12.535858 10.726915 5.723714
## 16 2018 65 female          Scotland -0.06883654 -16.054439 15.916766 7.866390
## 17 2017 65 female Synthetic Germany  1.60189046  -8.061272 11.265053 4.743773
## 18 2018 65 female             Spain  2.19830409 -11.023504 15.420112 6.506348
## 19 2018 65 female            France  1.39777068 -11.805226 14.600767 6.497091
## 20 2018 65 female             Italy  4.47429730 -10.963447 19.912041 7.596793
## 21 2019 65 female       Netherlands  0.65630234  -8.168485  9.481090 4.347588
## 22 2018 65   male   England & Wales  0.03630923  -8.724677  8.797295 4.311213
## 23 2018 65   male          Scotland  1.54944773 -10.702427 13.801322 6.029051
## 24 2017 65   male Synthetic Germany  1.51214525  -6.944740  9.969031 4.151596
## 25 2018 65   male             Spain  2.76986201  -9.287731 14.827455 5.933447
## 26 2018 65   male            France  2.23048158  -7.404687 11.865650 4.741391
## 27 2018 65   male             Italy  6.56582403  -4.980548 18.112196 5.681879
## 28 2019 65   male       Netherlands  4.38609383  -4.768505 13.540692 4.510071
```

The estimates are in weeks of increase per year. The following function produces summary statements for each population

```r
summarise_loess <- function(df, cntry, dp = 2){
  df <- df %>% filter(country == cntry)

  last_year <- df$year[1]

  y_x0_pt_m <- df %>% filter(x == 0, sex == "male") %>% pull("y_point") %>% round(dp)
  y_x65_pt_m <- df %>% filter(x == 65, sex == "male") %>% pull("y_point") %>% round(dp)
  y_x0_pt_f <- df %>% filter(x == 0, sex == "female") %>% pull("y_point") %>% round(dp)
  y_x65_pt_f <- df %>% filter(x == 65, sex == "female") %>% pull("y_point") %>% round(dp)

  y_x0_lwr_m <- df %>% filter(x == 0, sex == "male") %>% pull("y_lower") %>% round(dp)
  y_x65_lwr_m <- df %>% filter(x == 65, sex == "male") %>% pull("y_lower") %>% round(dp)
  y_x0_lwr_f <- df %>% filter(x == 0, sex == "female") %>% pull("y_lower") %>% round(dp)
  y_x65_lwr_f <- df %>% filter(x == 65, sex == "female") %>% pull("y_lower") %>% round(dp)

  y_x0_upr_m <- df %>% filter(x == 0, sex == "male") %>% pull("y_upper") %>% round(dp)
  y_x65_upr_m <- df %>% filter(x == 65, sex == "male") %>% pull("y_upper") %>% round(dp)
  y_x0_upr_f <- df %>% filter(x == 0, sex == "female") %>% pull("y_upper") %>% round(dp)
  y_x65_upr_f <- df %>% filter(x == 65, sex == "female") %>% pull("y_upper") %>% round(dp)


  y_x0_se_m <- df %>% filter(x == 0, sex == "male") %>% pull("y_se") %>% round(dp)
  y_x65_se_m <- df %>% filter(x == 65, sex == "male") %>% pull("y_se") %>% round(dp)
  y_x0_se_f <- df %>% filter(x == 0, sex == "female") %>% pull("y_se") %>% round(dp)
```

```r
  y_x65_se_f <- df %>% filter(x == 65, sex == "female") %>% pull("y_se") %>% round(dp)

  pfall_x0_m <- pnorm(0, y_x0_pt_m, y_x0_se_m) %>% round(dp)
  pfall_x65_m <- pnorm(0, y_x65_pt_m, y_x65_se_m) %>% round(dp)
  pfall_x0_f <- pnorm(0, y_x0_pt_f, y_x0_se_f) %>% round(dp)
  pfall_x65_f <- pnorm(0, y_x65_pt_f, y_x65_se_f) %>% round(dp)

  glue::glue("{cntry} in {last_year}: \nFor males, LOESS estimated annual changes of {y_x0_pt_m} (95% CI
}
```

The following are descriptive summaries of life expectancy trends in the last observed year, as estimated through the LOESS method:

```r
unique(loess_lastyear_both$country) %>% as.character() %>%
  map(summarise_loess, df = loess_lastyear_both)
```

```
## [[1]]
## England & Wales in 2018:
## For males, LOESS estimated annual changes of -1.77 (95% CI -11.85 to 8.31) weeks/year for life expec
## For females, LOESS estimated annual changes of -1.37 (95% CI -13.53 to 10.79) weeks/year for life ex
##
## [[2]]
## Scotland in 2018:
## For males, LOESS estimated annual changes of -3.64 (95% CI -17.51 to 10.22) weeks/year for life expe
## For females, LOESS estimated annual changes of -3.62 (95% CI -20.51 to 13.28) weeks/year for life ex
##
## [[3]]
## Synthetic Germany in 2017:
## For males, LOESS estimated annual changes of 5.97 (95% CI -4.69 to 16.63) weeks/year for life expect
## For females, LOESS estimated annual changes of 2.97 (95% CI -8.06 to 14) weeks/year for life expecta
##
## [[4]]
## Spain in 2018:
## For males, LOESS estimated annual changes of 4.51 (95% CI -9.69 to 18.71) weeks/year for life expect
## For females, LOESS estimated annual changes of 2.94 (95% CI -11.31 to 17.19) weeks/year for life exp
##
## [[5]]
## France in 2018:
## For males, LOESS estimated annual changes of 5.33 (95% CI -5.95 to 16.6) weeks/year for life expecta
## For females, LOESS estimated annual changes of 1.38 (95% CI -12.05 to 14.81) weeks/year for life exp
##
## [[6]]
## Italy in 2018:
## For males, LOESS estimated annual changes of 7.77 (95% CI -6.6 to 22.14) weeks/year for life expecta
## For females, LOESS estimated annual changes of 5.11 (95% CI -11.53 to 21.74) weeks/year for life exp
##
## [[7]]
## Netherlands in 2019:
## For males, LOESS estimated annual changes of 5.69 (95% CI -5.72 to 17.11) weeks/year for life expect
## For females, LOESS estimated annual changes of 2.37 (95% CI -8.64 to 13.37) weeks/year for life expe
```

We can see from the above summaries that the confidence intervals for all populations tend to include negative values, and that the resulting estimated probabilities that the true value in the last observed year (conditional on the model) is below zero, i.e. falling, is substantial (at least 12%) for populations and starting ages considered. For England & Wales, and Scotland (except males from age 65), it is more probable than not that life expectancy fell in the last observed year.

Linear regression and Breakpoint confidence intervals

```r
estimate_breakpoints_and_pval <- function(df){
  null_mdl <- lm(ex ~ year, data = df)
  seg_mdl <- segmented::segmented(null_mdl, seg.Z= ~year, psi = 2010)
  seg2_mdl <- segmented::segmented(null_mdl, seg.z= ~year, psi= c(1985, 2010)) # added to test

  list(
    null = null_mdl,
    seg = seg_mdl,
    seg2 = seg2_mdl
  )
}

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,
      function(x, y){
        if (x == 'seg2') {
          y[["psi"]][2,2]
        } else  {
            NA_real_
          }
      }
      ),
    second_breakpoint_se = map2_dbl(
      mdl_outputs_id, mdl_outputs,
      function(x, y){
        if (x == 'seg2') {
          y[["psi"]][2,3]
        } else  {
            NA_real_
          }
      }
    )
  ) %>%  # let's add predictions too
  mutate(
    pred_data = map(mdl_outputs, make_predictions),
    joined_data = map2(data, pred_data, left_join)
  ) %>%
  select(code, x, sex, joined_data, first_breakpoint:second_breakpoint_se) %>%
  unnest(joined_data)
```

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

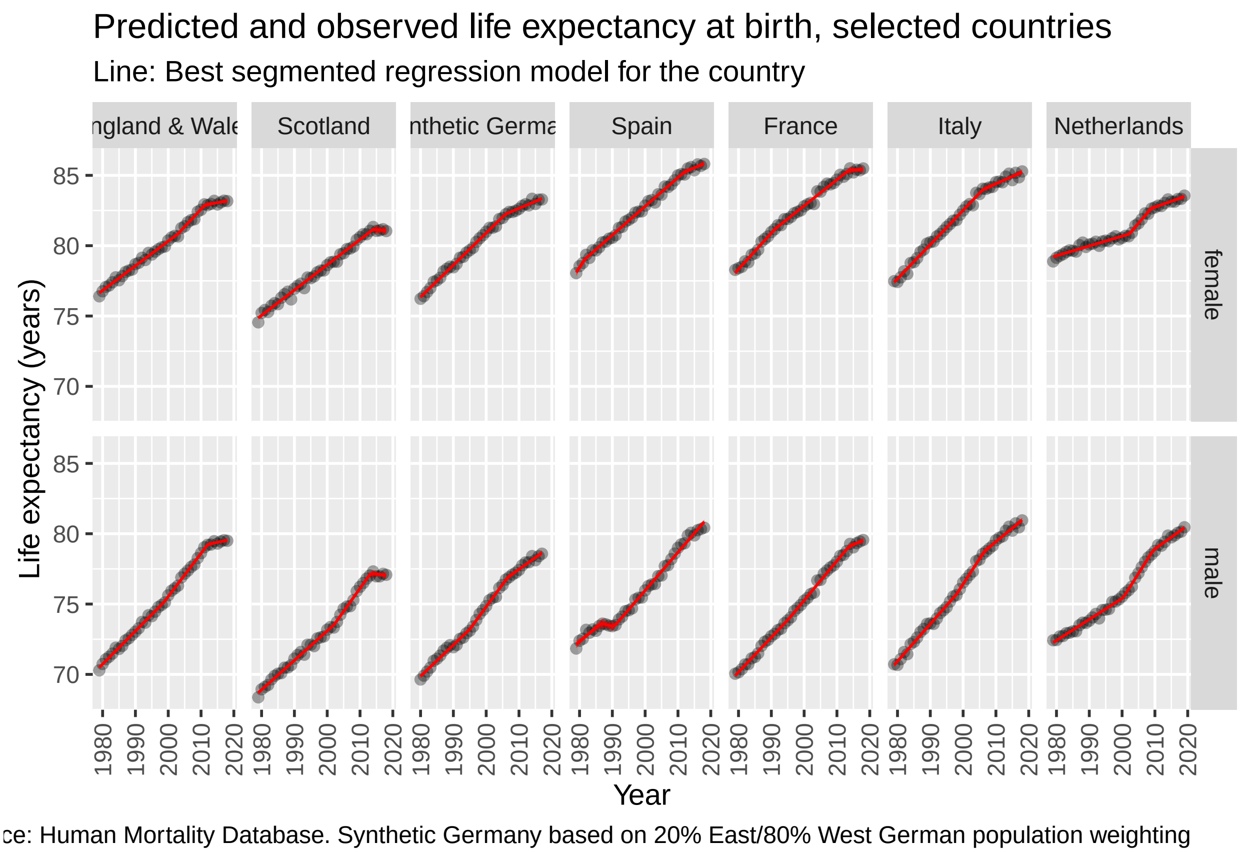
```
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining, by = "year"
## Joining,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

```
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, interval = "confidence", level = 0.95)) %>%
    mutate(data_augmented = map2(data, predictions, bind_cols)) %>%
    select(code, x, sex, data_augmented) %>%
    ungroup() %>%
    unnest(data_augmented)
```

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  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 = fit), colour = "red") +
  geom_ribbon(aes(ymin = lwr, ymax = upr), fill = "red", colour = NA, alpha = 0.5) +
  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 po
  )
```

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

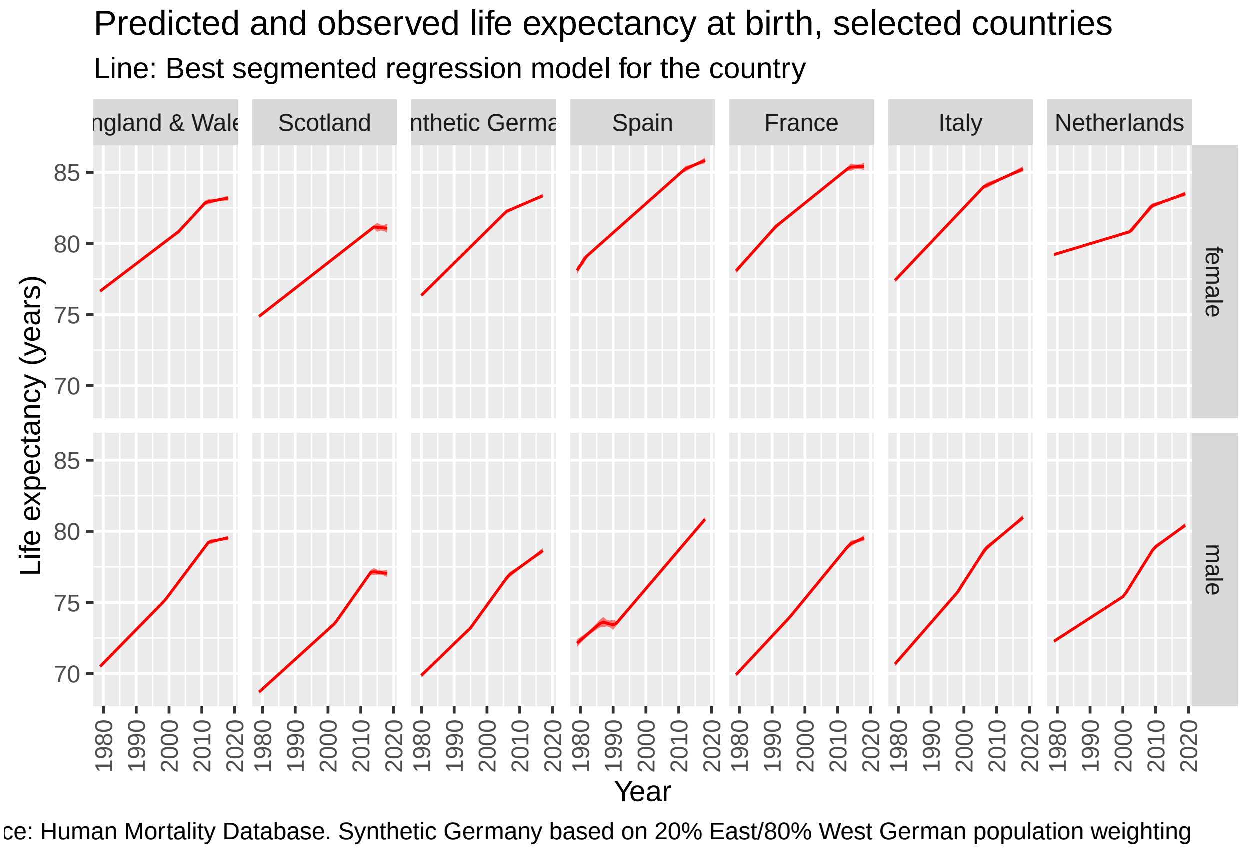


The confidence intervals are quite hard to see given the data points. In the figure below they are shown without the points

```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  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 = fit), colour = "red") +
  geom_ribbon(aes(ymin = lwr, ymax = upr), fill = "red", colour = NA, alpha = 0.5) +
  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 po
```

```
)
```

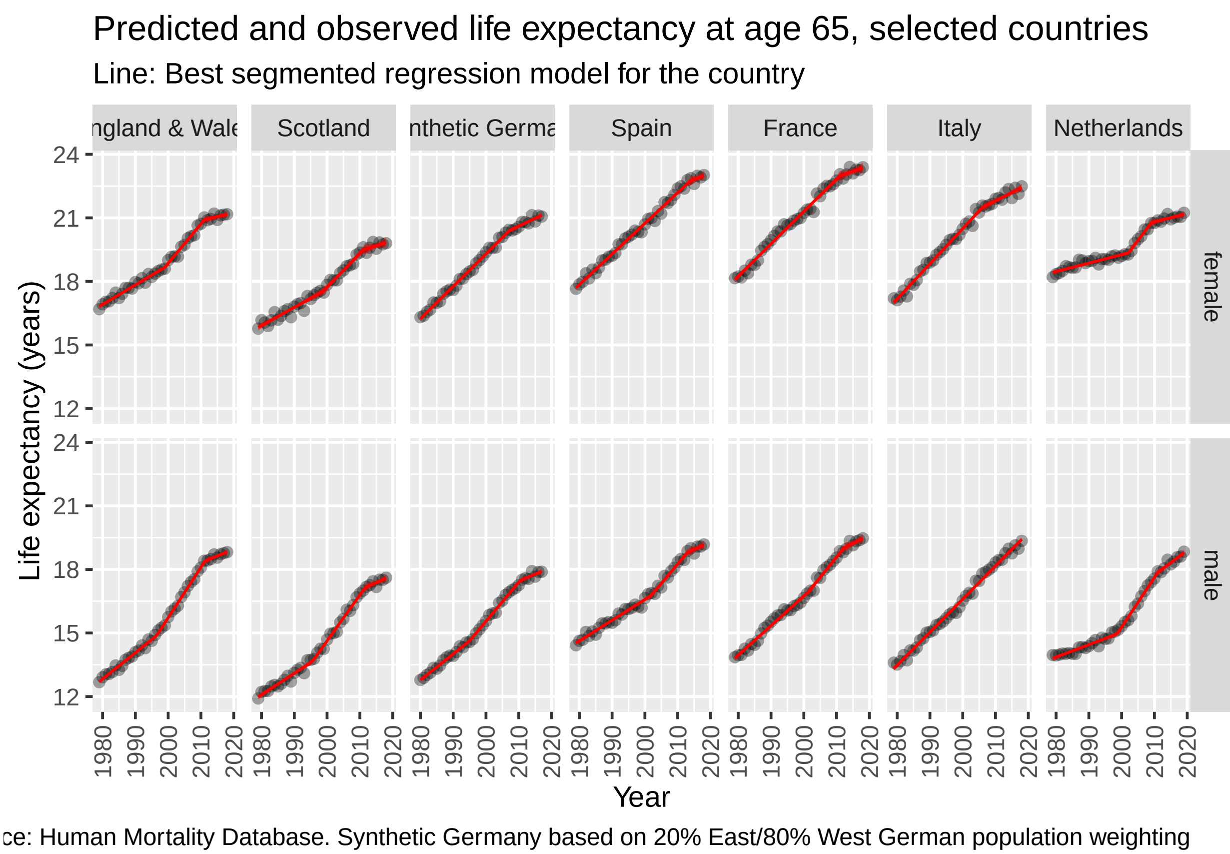
```
## Joining, by = "code"
```



They are repeated, with and then without points, for life expectancy from age 65 below:

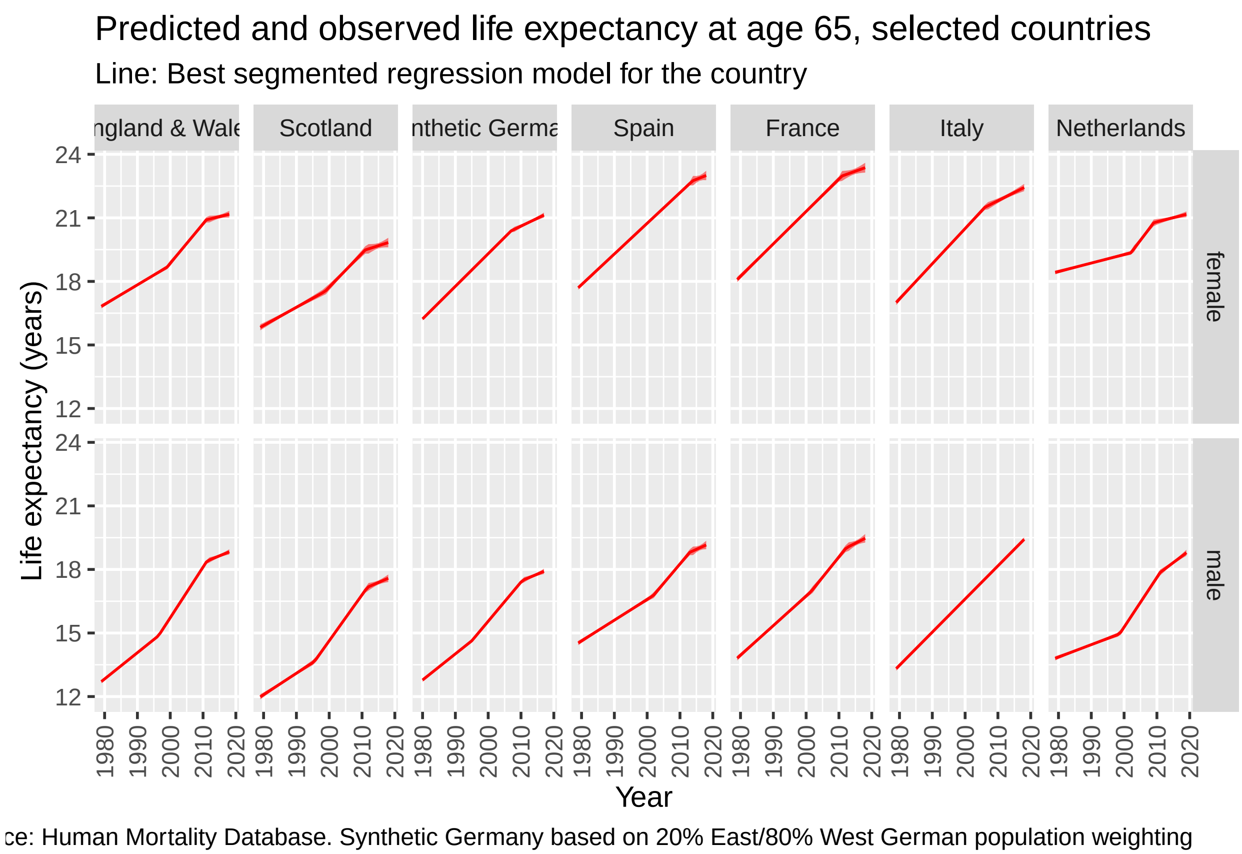
```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  filter(!is.na(country)) %>%
  filter(x == 65) %>%
  ggplot(aes(x = year)) +
  facet_grid(sex ~ country) +
  geom_point(aes(y = ex), alpha = 0.35) +
  geom_line(aes(y = fit), colour = "red") +
  geom_ribbon(aes(ymin = lwr, ymax = upr), fill = "red", colour = NA, alpha = 0.5) +
  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 po
  )
```

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




```
best_breakpoint_model_predictions %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  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 = fit), colour = "red") +
  geom_ribbon(aes(ymin = lwr, ymax = upr), fill = "red", colour = NA, alpha = 0.5) +
  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 p
  )
```

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



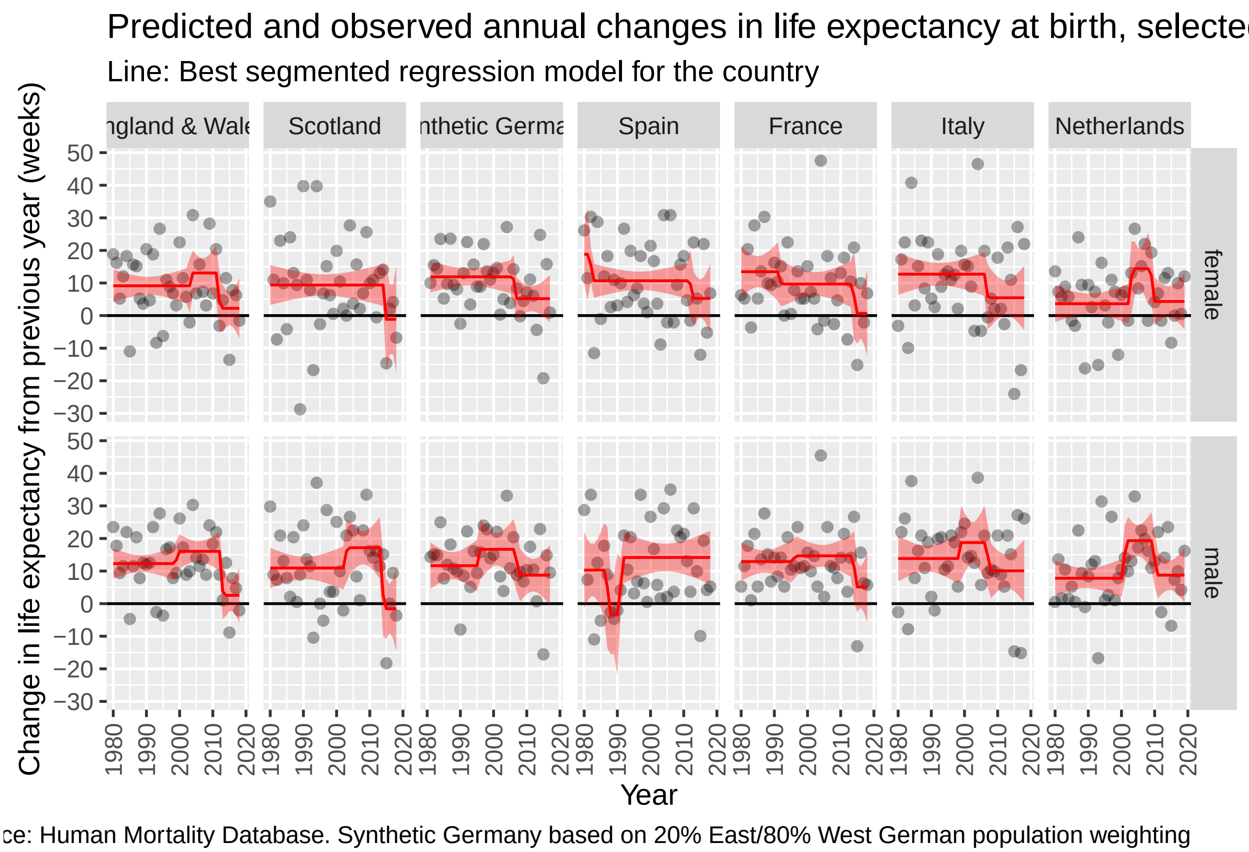

```
pred_differenced <-
  best_breakpoint_model_predictions %>%
    left_join(country_code_lookup) %>%
    mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spa
    filter(!is.na(country)) %>%
    group_by(country, sex, x) %>%
    arrange(year) %>%
    mutate(
      diff_ex = 52.25 * (ex - lag(ex)),
      diff_fit = 52.25 * (fit - lag(fit)),
      adj_upr = 52.25 * (upr - fit),
      adj_lwr = 52.25 * (lwr - fit),
      diff_upr = diff_fit + adj_upr,
      diff_lwr = diff_fit + adj_lwr
    ) %>%
    filter(!is.na(diff_fit))
```

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

The following shows differences with confidence intervals for life expectancy from birth:

```
pred_differenced %>%
  filter(x == 0) %>%
  ggplot(aes(year)) +
  geom_point(aes(y = diff_ex), alpha = 0.35) +
  facet_grid(sex ~ country) +
  geom_hline(yintercept = 0) +
```

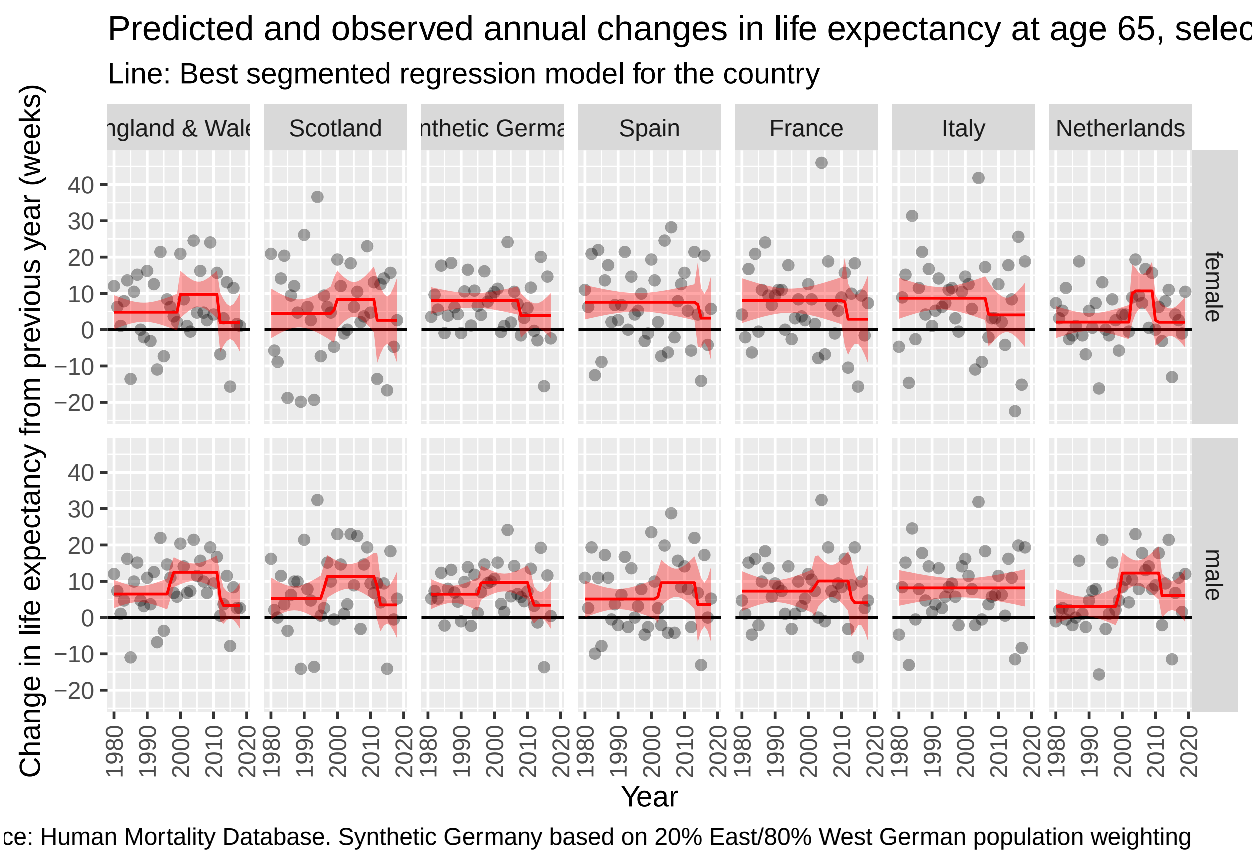
```
  geom_line(aes(y = diff_fit), colour = "red") +
  geom_ribbon(aes(ymin = diff_lwr, ymax = diff_upr), colour = NA, fill = "red", alpha = 0.35) +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  ) +
  scale_y_continuous(breaks = seq(-30, 50, by = 10)) +
  labs(
    x = "Year",
    y = "Change in life expectancy from previous year (weeks)",
    title = "Predicted and observed annual changes in 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 po
  )
```



And the following shows the same for life expectancy changes from age 65

```
pred_differenced %>%
  filter(x == 65) %>%
  ggplot(aes(year)) +
  geom_point(aes(y = diff_ex), alpha = 0.35) +
  facet_grid(sex ~ country) +
  geom_hline(yintercept = 0) +
  geom_line(aes(y = diff_fit), colour = "red") +
  geom_ribbon(aes(ymin = diff_lwr, ymax = diff_upr), colour = NA, fill = "red", alpha = 0.35) +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
```

```
) +
scale_y_continuous(breaks = seq(-30, 50, by = 10)) +
labs(
  x = "Year",
  y = "Change in life expectancy from previous year (weeks)",
  title = "Predicted and observed annual changes in 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 po
)
```



As with the LOESS model, we can can present the point estimates, 95% CIs, and implied probabilities that the values in the last observed year are below zero.

```
summarise_reg_seg <- function(df, cntry, dp = 2){
  df <- df %>%
    filter(country == cntry) %>%
    filter(year == max(year))

  last_year <- max(df$year)

  y_x0_pt_m <- df %>% filter(x == 0, sex == "male") %>% pull("diff_fit")
  y_x65_pt_m <- df %>% filter(x == 65, sex == "male") %>% pull("diff_fit")
  y_x0_pt_f <- df %>% filter(x == 0, sex == "female") %>% pull("diff_fit")
  y_x65_pt_f <- df %>% filter(x == 65, sex == "female") %>% pull("diff_fit")

  y_x0_lwr_m <- df %>% filter(x == 0, sex == "male") %>% pull("diff_lwr")
  y_x65_lwr_m <- df %>% filter(x == 65, sex == "male") %>% pull("diff_lwr")
```

```
  y_x0_lwr_f <- df %>% filter(x == 0, sex == "female") %>% pull("diff_lwr")
  y_x65_lwr_f <- df %>% filter(x == 65, sex == "female") %>% pull("diff_lwr")

  y_x0_upr_m <- df %>% filter(x == 0, sex == "male") %>% pull("diff_upr")
  y_x65_upr_m <- df %>% filter(x == 65, sex == "male") %>% pull("diff_upr")
  y_x0_upr_f <- df %>% filter(x == 0, sex == "female") %>% pull("diff_upr")
  y_x65_upr_f <- df %>% filter(x == 65, sex == "female") %>% pull("diff_upr")


  y_x0_se_m <-  c(y_x0_upr_m - y_x0_pt_m, y_x0_pt_m - y_x0_lwr_m) %>% `/`(1.96) %>% mean() %>% round(dp
  y_x65_se_m <- c(y_x65_upr_m - y_x65_pt_m, y_x65_pt_m - y_x65_lwr_m) %>% `/`(1.96) %>% mean() %>% rounc
  y_x0_se_f <-  c(y_x0_upr_f - y_x0_pt_f, y_x0_pt_f - y_x0_lwr_f) %>% `/`(1.96) %>% mean() %>% round(dp
  y_x65_se_f <- c(y_x65_upr_f - y_x65_pt_f, y_x65_pt_f - y_x65_lwr_f) %>% `/`(1.96) %>% mean() %>% rounc

  y_x0_lwr_m <- y_x0_lwr_m %>% round(dp)
  y_x65_lwr_m <- y_x65_lwr_m %>% round(dp)
  y_x0_lwr_f <- y_x0_lwr_f %>% round(dp)
  y_x65_lwr_f <- y_x65_lwr_f %>% round(dp)

  y_x0_upr_m <- y_x0_upr_m %>% round(dp)
  y_x65_upr_m <- y_x65_upr_m %>% round(dp)
  y_x0_upr_f <- y_x0_upr_f %>% round(dp)
  y_x65_upr_f <- y_x65_upr_f %>% round(dp)

  y_x0_pt_m <- y_x0_pt_m %>% round(dp)
  y_x65_pt_m <- y_x65_pt_m %>% round(dp)
  y_x0_pt_f <- y_x0_pt_f %>% round(dp)
  y_x65_pt_f <- y_x65_pt_f %>% round(dp)

  pfall_x0_m <- pnorm(0, y_x0_pt_m, y_x0_se_m) %>% round(dp)
  pfall_x65_m <- pnorm(0, y_x65_pt_m, y_x65_se_m) %>% round(dp)
  pfall_x0_f <- pnorm(0, y_x0_pt_f, y_x0_se_f) %>% round(dp)
  pfall_x65_f <- pnorm(0, y_x65_pt_f, y_x65_se_f) %>% round(dp)

  glue::glue("{cntry} in {last_year}: \nFor males, segmented/linear estimated annual changes of {y_x0_p
}
```

As before, we can produce summary statements for each country as follows:

```
unique(pred_differenced$country) %>% as.character() %>%
  map(summarise_reg_seg , df = pred_differenced)
```

```
## [[1]]
## Spain in 2018:
## For males, segmented/linear estimated annual changes of 14.16 (95% CI 5.92 to 22.4) weeks/year for li
## For females, segmented/linear estimated annual changes of 5.26 (95% CI -5.23 to 15.76) weeks/year fo
##
## [[2]]
## France in 2018:
## For males, segmented/linear estimated annual changes of 5.12 (95% CI -5.76 to 16) weeks/year for life
## For females, segmented/linear estimated annual changes of 0.68 (95% CI -12.57 to 13.93) weeks/year fo
##
## [[3]]
## Italy in 2018:
```

```
## For males, segmented/linear estimated annual changes of 10.06 (95% CI 0.47 to 19.66) weeks/year for l
## For females, segmented/linear estimated annual changes of 5.46 (95% CI -4.45 to 15.37) weeks/year fo
##
## [[4]]
## Netherlands in 2019:
## For males, segmented/linear estimated annual changes of 8.75 (95% CI 0.38 to 17.11) weeks/year for l
## For females, segmented/linear estimated annual changes of 4.35 (95% CI -4.25 to 12.95) weeks/year fo
##
## [[5]]
## England & Wales in 2018:
## For males, segmented/linear estimated annual changes of 2.55 (95% CI -5.8 to 10.91) weeks/year for l
## For females, segmented/linear estimated annual changes of 2.23 (95% CI -6.95 to 11.41) weeks/year fo
##
## [[6]]
## Scotland in 2018:
## For males, segmented/linear estimated annual changes of -1.57 (95% CI -14.52 to 11.39) weeks/year fo
## For females, segmented/linear estimated annual changes of -1.15 (95% CI -17.55 to 15.25) weeks/year
##
## [[7]]
## Synthetic Germany in 2017:
## For males, segmented/linear estimated annual changes of 8.74 (95% CI -0.57 to 18.05) weeks/year for l
## For females, segmented/linear estimated annual changes of 5.2 (95% CI -2.03 to 12.43) weeks/year for
```