Time Series Models

Jon Minton

2022-07-21

Are annual ex time series random walks with drift (RWD) or do they have autocorrelation?

Section X of the main manuscript noted, from figure 1a and 1b, that there is substantial annual variation in annual changes in life expectancy. After taking the first difference in these series (i.e. subtracting for each value in the series its previous value), the series of remaining difference may be purely random. Alternatively, even after such differencing the series may have some degree of autocorrelation (i.e. the differenced value for one year may be informative as to the differenced value for the next one or more observations in the series), or there may be a moving average component to the series.

Different patterns of variation in the time series can be modelled using different specifications of the ARIMA modelling approach. ARIMA stands for Autoregressive (AR), Integrated (I), Moving Average (MA), and different ARIMA specifications are represented by the shorthand ARIMA(p, d, q). The p term indicates the number of autoregressive terms, the d term the number of times the series have been differenced, and the q term the number of moving average terms.

In this appendix we are primarily interested in comparing the ARIMA(0, 1, 0) specification with the ARIMA(1, 1, 0) specification. The ARIMA(0, 1, 0) is known as Random-Walk-with-Drift (RWD) and is the simpler of the two specifications. ARIMA(1, 1, 0) includes one autoregressive term p, meaning for each value in the series there is some correlation with the previous value in the series. For this autoregressive term p, a positive coefficient may be interpreted as indicating some degree of 'stickiness' in the series (a better-than-average value is more likely than by chance to be followed by a better-than-average value, and vice versa), whereas a negative coefficient interpreted as indicating some degree of 'oscillation' in the series (a better-than-average value is more likely than chance to be followed by a worse-than-average value, and so on).

This appendix section will first compare the two ARIMA model specifications ARIMA(0,1,0) (RWD) and ARIMA(1,1,0) (Autoregressive and integrated) for each of the time series. It will then use the auto.arima function from the fable package to consider a wider range of ARIMA model specifications and identify which model specifications are preferred for which datasets.

Preparation

First we load the requisite data, packages, and do the required data processing

```
# This loads the required packages (the pacman package must be installed first, using install.packages(
pacman::p_load(tidyverse, fable, here)

# The following code calculates e0 and e65 for the countries under consideration, using lifetables prev

hmd_lt <- read_rds(here("data", "lifetables.rds"))

# 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(here("R", "make_synthetic_germany_functions.R"))
source(here("R", "make_pop_selection.R"))

series_of_interest <-
  hmd_ex_selected_countries_with_synth %>%
    left_join(country_code_lookup) %>%
    mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spa
    filter(!is.na(country)) %>%
    filter(between(year, 1979, 2020))
```

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

```
series_of_interest
```

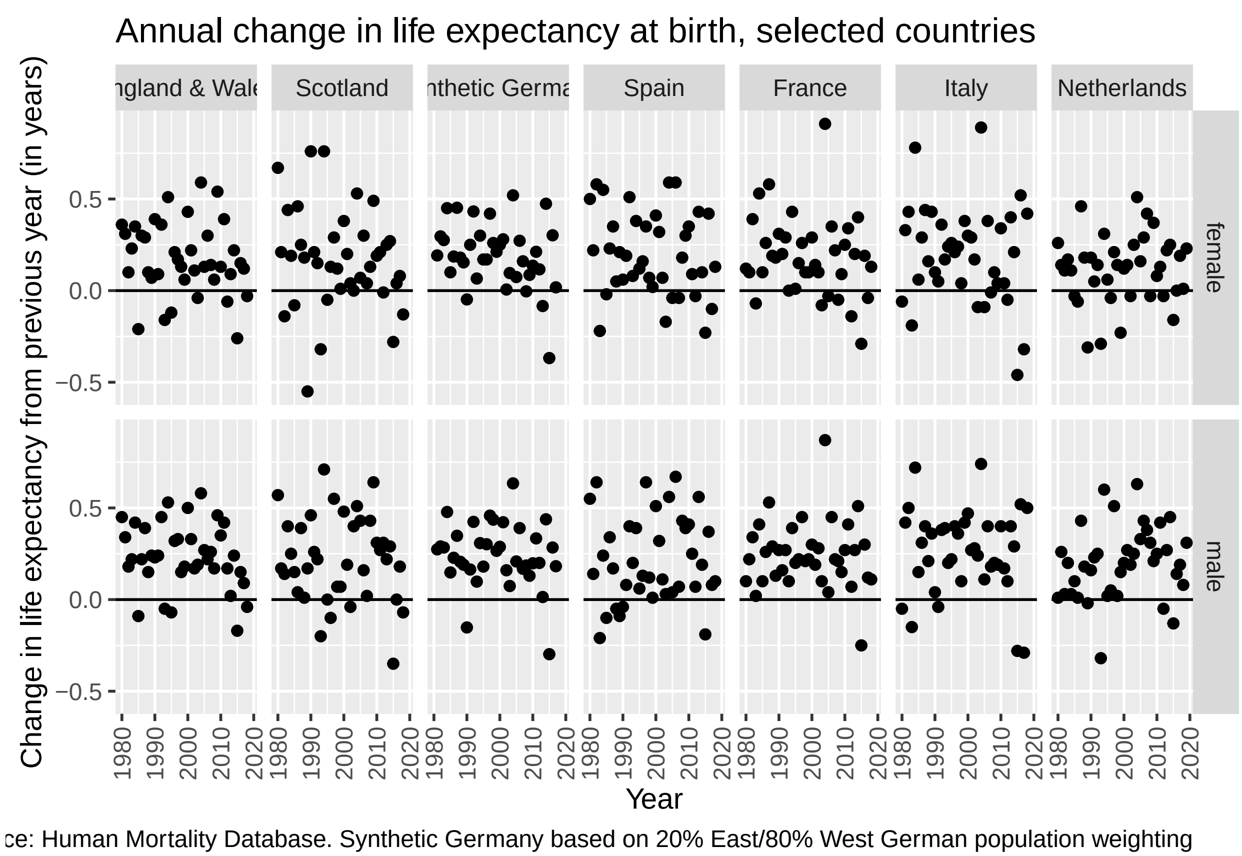
```
## # A tibble: 1,116 x 6
##    code   year     x sex       ex country
##    <chr> <int> <dbl> <chr>  <dbl> <fct>
##  1 ESP    1979     0 female  78.0 Spain
##  2 ESP    1979    65 female  17.6 Spain
##  3 ESP    1980     0 female  78.6 Spain
##  4 ESP    1980    65 female  17.9 Spain
##  5 ESP    1981     0 female  78.8 Spain
##  6 ESP    1981    65 female  18.0 Spain
##  7 ESP    1982     0 female  79.4 Spain
##  8 ESP    1982    65 female  18.4 Spain
##  9 ESP    1983     0 female  79.1 Spain
## 10 ESP    1983    65 female  18.1 Spain
## # ... with 1,106 more rows
```

Our series contains ex for x = 0 (life expectancy at birth) and x=65 (life expectancy at age 65 years), for each of the countries of interest, including 'Synthetic Germany'.

Visualisation

The data series look as follows for x = 0

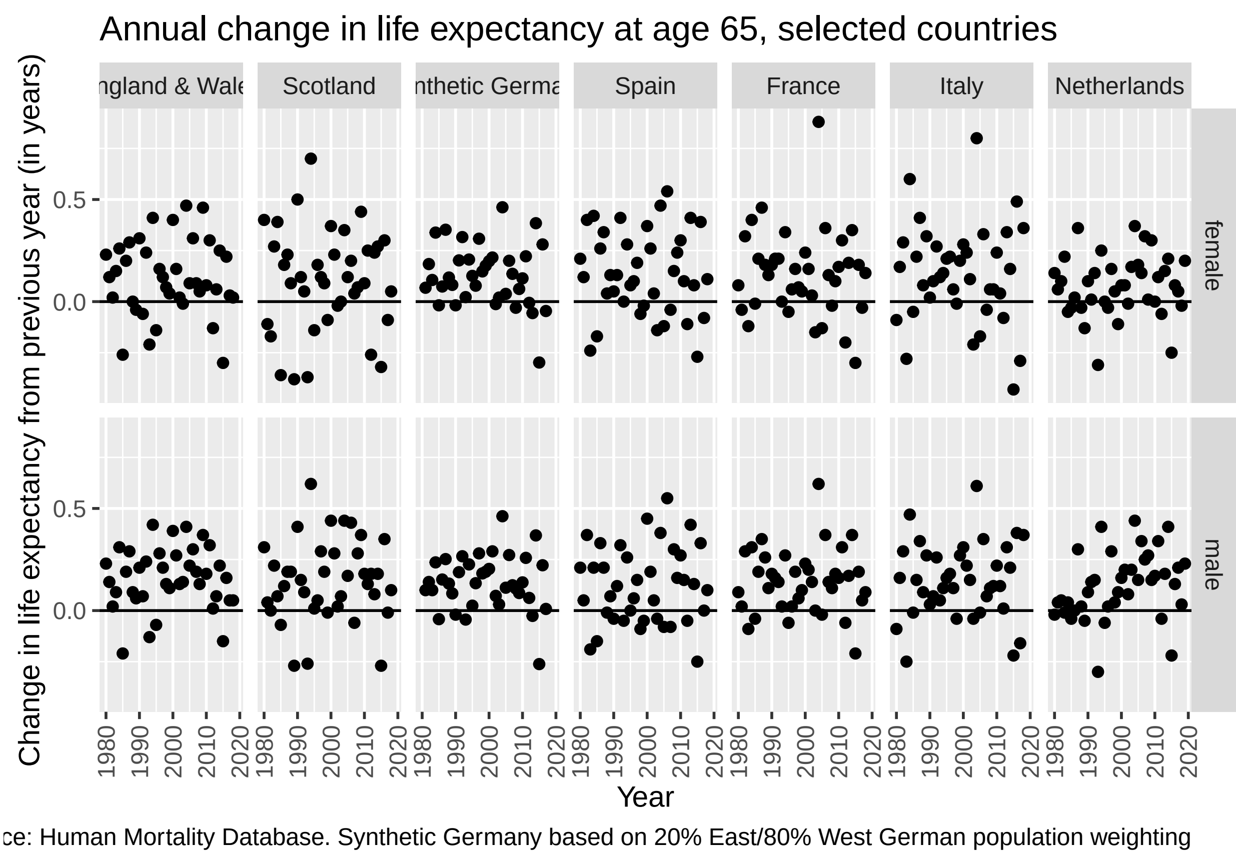
```
series_of_interest %>%
  filter(x == 0) %>%
  group_by(country, sex) %>%
  arrange(year) %>%
  mutate(delta_ex = ex - lag(ex)) %>%
  filter(!is.na(delta_ex)) %>%
  ggplot(aes(x = year, y = delta_ex)) +
  geom_point() +
  facet_grid(sex ~ country) +
  geom_hline(yintercept = 0) +
  theme(
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  ) +
  labs(
    x = "Year",
    y = "Change in life expectancy from previous year (in years)",
    title = "Annual change in life expectancy at birth, selected countries",
    caption =  "Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German po
  )
```



(n.b. the data in this series are shown in years per year, rather than weeks per year as in the main figure)

The equivalent series for e65 is as follows:

```
series_of_interest %>%
  filter(x == 65) %>%
  group_by(country, sex) %>%
  arrange(year) %>%
  mutate(delta_ex = ex - lag(ex)) %>%
  filter(!is.na(delta_ex)) %>%
  ggplot(aes(x = year, y = delta_ex)) +
  geom_point() +
  facet_grid(sex ~ country) +
  geom_hline(yintercept = 0) +
  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 years)",
    title = "Annual change in life expectancy at age 65, selected countries",
    caption =  "Source: Human Mortality Database. Synthetic Germany based on 20% East/80% West German po
  )
```



Calculating and comparing ARIMA(0,1,0) wih ARIMA(1,1,0)

A time series where each observation in the series is simply dependent on the previous value, plus some random variation (which may be negative), can be expressed as an ARIMA(0, 1, 0) model. By contrast, a model where each observation in the series oscillates slightly (i.e. 'worse-than-average' years are more likely than chance to be followed by 'better-than-average' years, and vice versa), is likely to be represented by an ARIMA(1, 1, 0) model, where the coefficient on this first term (called `p`) should be negative rather than positive.

These models can be fit using the `forecast` package, and model fit compared using the AICc metric.

```
fit_arima_model <- function(series, order){
  series %>%
    pull(ex) %>%
    as.ts(start = 1979) %>%
    forecast::Arima(order = order)
}

ts_model_comparisons <-
  series_of_interest %>%
    group_by(country, sex, x) %>%
    nest() %>%
    mutate(
      arima_010 = map(data, fit_arima_model, order = c(1, 1, 0)),
      arima_110 = map(data, fit_arima_model, order = c(1, 1, 0))
    ) %>%
  mutate(
    aicc_arima_010 = map_dbl(arima_010, ~summary(.) %>% pluck("aicc")),
    aicc_arima_110 = map_dbl(arima_110, ~summary(.) %>% pluck("aicc"))
  ) %>%
  mutate(
    which_preferred = if_else(aicc_arima_010 < aicc_arima_110, "Random Walk", "Autocorrelated")
  )
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
ts_model_comparisons %>%
  select(country, sex, x, which_preferred)
```

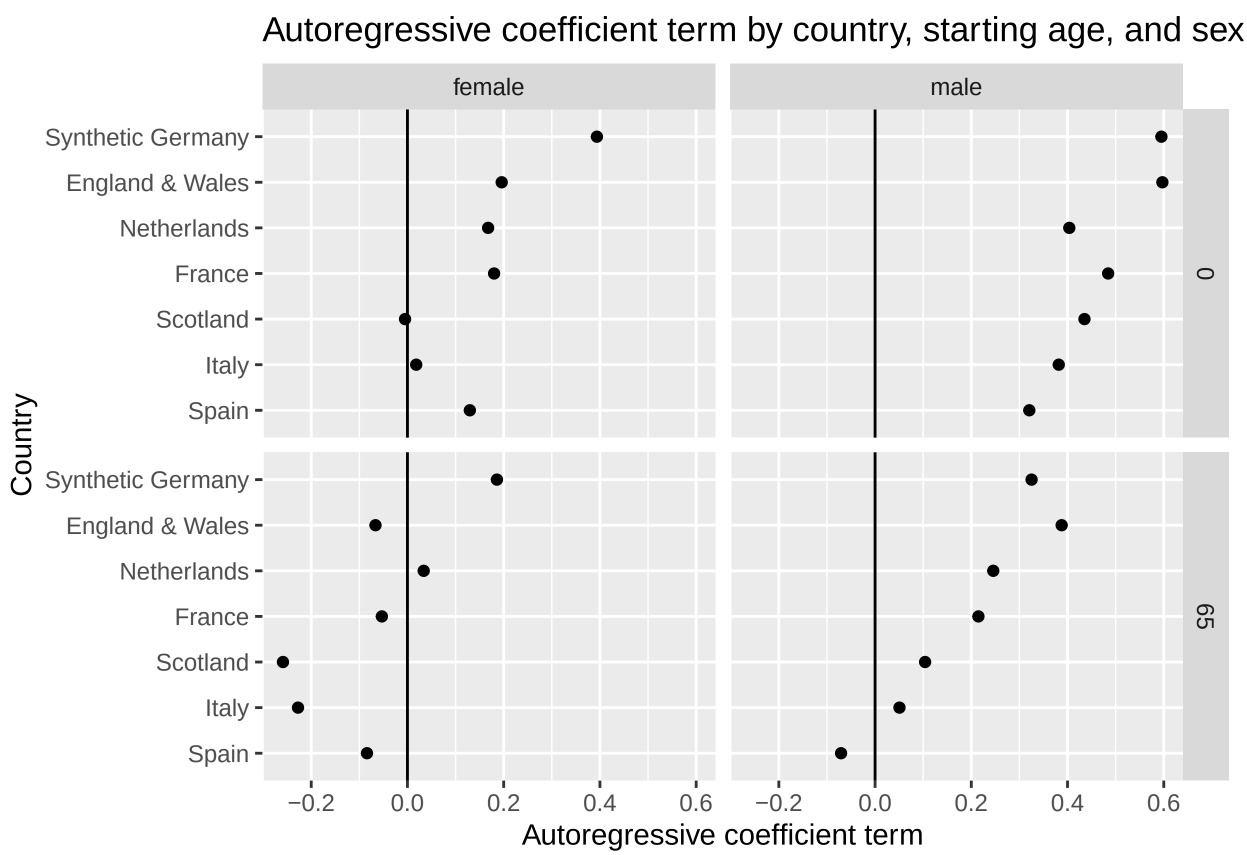
```
## # A tibble: 28 x 4
## # Groups:   country, sex, x [28]
##    country         sex        x which_preferred
##    <fct>           <chr>  <dbl> <chr>
##  1 Spain           female     0 Autocorrelated
##  2 Spain           female    65 Autocorrelated
##  3 France          female     0 Autocorrelated
##  4 France          female    65 Autocorrelated
##  5 Italy           female     0 Autocorrelated
##  6 Italy           female    65 Autocorrelated
##  7 Netherlands     female     0 Autocorrelated
##  8 Netherlands     female    65 Autocorrelated
##  9 England & Wales female     0 Autocorrelated
## 10 England & Wales female    65 Autocorrelated
## # ... with 18 more rows
```

The Autocorrelated specification (ARIMA(1,1,0)) is preferred to RWD (ARIMA(0,1,0)) for all populations.

The following extracts the autocorrelation coefficients and visualises them.

```
get_ar_term_and_se <- function(mdl){
  tibble(
    ar = pluck(mdl, "coef"),
    ar_coef = pluck(mdl, "var.coef")[1,1] %>% sqrt()
  )
}

ts_model_comparisons %>%
  mutate(mdl_terms = map(arima_110, get_ar_term_and_se)) %>%
  select(x, sex, country, mdl_terms) %>%
  unnest_wider(mdl_terms) %>%
  arrange(ar) %>%
  ggplot(aes(ar, fct_reorder(country, ar))) +
  geom_point() +
  facet_grid(x ~ sex) +
  geom_vline(xintercept = 0) +
  labs(x = "Autoregressive coefficient term", y = "Country",
      title ="Autoregressive coefficient term by country, starting age, and sex")
```



The majority of these coefficients are positive, indicating 'stickiness' in the values in the series, rather than oscillation. The exception is for females for conditional life expectancy at age 65, where the coefficients are negative for England & Wales, France, Scotland, Italy, and Spain. This suggests that for older females the life expectancy series tends to 'oscillate' rather than 'stick'.

Alternative approach to establishing if series are autocorrelated

An alternative way to determing whether the series are autocorrelated is to compare two linear regression equations: one in which change in life expectancy is regressed against an intercept; another in which the change in life expectancy is additionally regressed against the change in the previous year

```
alt_approach_autocorrelated <-
  hmd_ex_selected_countries_with_synth %>%
      filter(year >= 1979) %>%
      filter(!(code %in% c("DEUTE", "DEUTW", "DEUTNP"))) %>%  # Using only synthetic germany for longer
      group_by(code, x, sex) %>%
      arrange(year) %>%
      mutate(delta_ex = ex - lag(ex)) %>%
      nest() %>%
      mutate(
        rwd_model = map(data, ~lm(delta_ex ~ 1, data = . )),
        ar_model  = map(data, ~lm(delta_ex ~ lag(delta_ex), data = .))
      ) %>%
      mutate(
        bic_rwd = map_dbl(rwd_model, BIC),
        bic_ar  = map_dbl(ar_model, BIC)
      ) %>%
      mutate(
        diff_bic = bic_ar - bic_rwd
      ) %>%
    mutate(
      which_preferred = ifelse(diff_bic < 0, "Autocorrelated", "RWD")
    ) %>%
      left_join(country_code_lookup) %>%
      mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "S
    filter(!is.na(country))
```

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

```
alt_approach_autocorrelated %>%
  select(country, x, sex, bic_rwd, bic_ar, which_preferred)
```

```
## Adding missing grouping variables: `code`
```

```
## # A tibble: 28 x 7
## # Groups:   code, x, sex [28]
##    code    country             x sex    bic_rwd bic_ar which_preferred
##    <chr>   <fct>           <dbl> <chr>    <dbl>  <dbl> <chr>
##  1 ESP     Spain               0 female    1.99  -7.45 Autocorrelated
##  2 ESP     Spain              65 female   -5.81 -15.4  Autocorrelated
##  3 FRATNP  France              0 female   -2.08  -4.67 Autocorrelated
##  4 FRATNP  France             65 female   -4.73 -10.4  Autocorrelated
##  5 ITA     Italy               0 female   13.4    2.80 Autocorrelated
##  6 ITA     Italy              65 female    6.71  -7.93 Autocorrelated
##  7 NLD     Netherlands         0 female  -15.4  -11.8  RWD
##  8 NLD     Netherlands        65 female  -33.1  -29.5  RWD
##  9 GBRTENW England & Wales     0 female   -8.97 -12.8  Autocorrelated
## 10 GBRTENW England & Wales    65 female  -13.9  -18.8  Autocorrelated
## # ... with 18 more rows
```

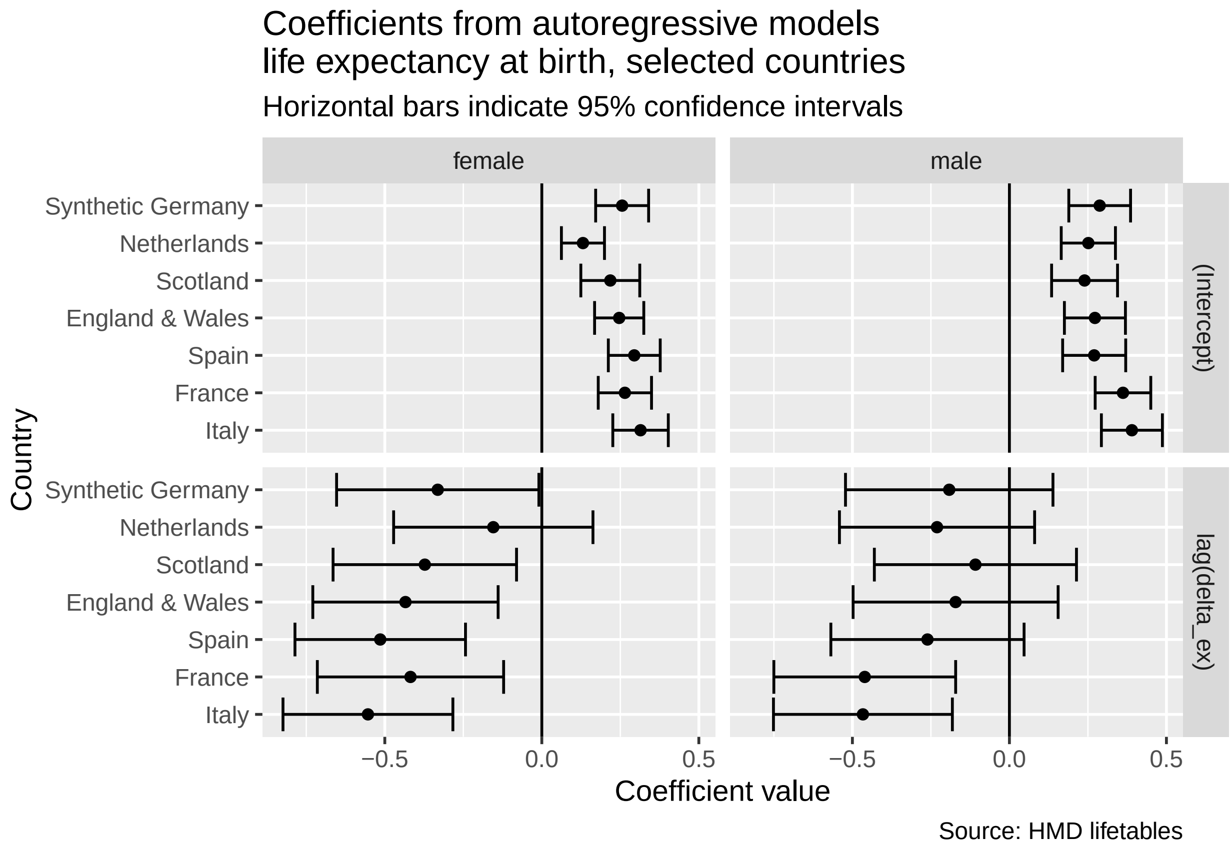
This approach also identifies most, but not all, of the populations to exhibit some degree of autocorrelation. (n.b. the more stringent BIC was used instead of AICc in this example, which may explain discrepancies between approaches)

For each of the above, the intercept (average improvement per year) and autocorrelation term can be extracted as follows:

```
alt_approach_autocorrelated  %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  filter(!is.na(country)) %>%
  select(country, x, sex, ar_model) %>%
  mutate(tidied_model = map(ar_model, broom::tidy)) %>%
  select(-ar_model) %>%
  filter(x == 0) %>%
  unnest(tidied_model) %>%
  ggplot(aes(estimate, reorder(country, estimate))) +
  geom_point() +
  facet_grid(term~sex) +
  geom_vline(xintercept = 0) +
  geom_errorbarh(aes(xmin = estimate - 1.96  * std.error, xmax = estimate + 1.96 * std.error))  +
  labs(
    x = "Coefficient value",
    y = "Country",
    title = "Coefficients from autoregressive models\nlife expectancy at birth, selected countries",
    subtitle = "Horizontal bars indicate 95% confidence intervals",
    caption = "Source: HMD lifetables"

  )
```

```
## Joining, by = c("code", "country")
## Adding missing grouping variables: `code`
```



Here all autoregression coefficients are negative, most of which are statistically significant at $p < 0.05$.

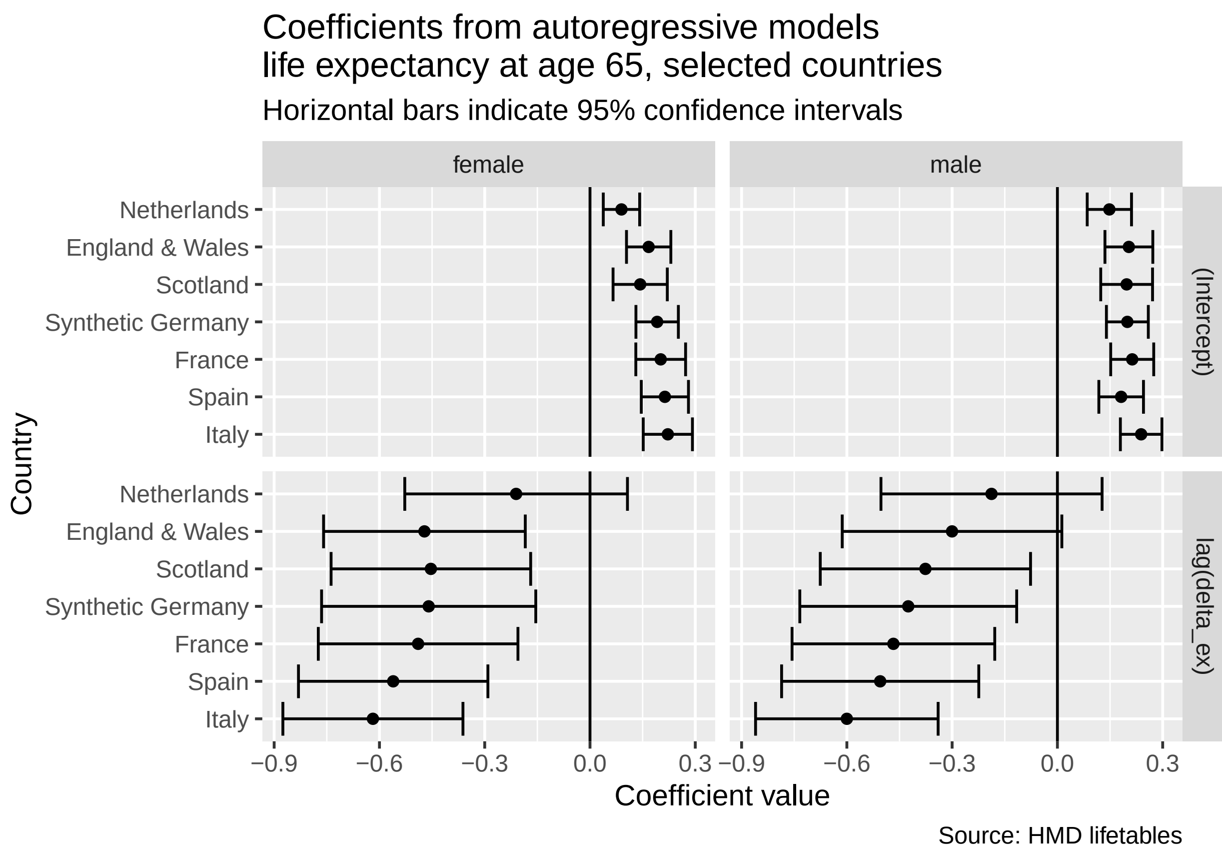
The following shows the equivalent trends for $x = 65$

```
alt_approach_autocorrelated  %>%
  left_join(country_code_lookup) %>%
  mutate(country = factor(country, levels = c("England & Wales", "Scotland", "Synthetic Germany", "Spai
  filter(!is.na(country)) %>%
  select(country, x, sex, ar_model) %>%
  mutate(tidied_model = map(ar_model, broom::tidy)) %>%
  select(-ar_model) %>%
  filter(x == 65) %>%
  unnest(tidied_model) %>%
  ggplot(aes(estimate, reorder(country, estimate))) +
  geom_point() +
  facet_grid(term~sex) +
  geom_vline(xintercept = 0) +
  geom_errorbarh(aes(xmin = estimate - 1.96  * std.error, xmax = estimate + 1.96 * std.error))  +
  labs(
    x = "Coefficient value",
    y = "Country",
    title = "Coefficients from autoregressive models\nlife expectancy at age 65, selected countries",
    subtitle = "Horizontal bars indicate 95% confidence intervals",
    caption = "Source: HMD lifetables"

  )
```

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

```
## Adding missing grouping variables: `code`
```



Here the negative correlations appear even stronger, and of larger magnitude in Italy than elsewhere. It also appears that the two coefficients are negatively associated, with Netherlands having the lowest average annual improvement, but also the smallest (and non-significant) magnitude of negative autocorrelation; Italy appears the converse.

Comparing a wider range of ARIMA models

The `auto.arima` function in the `fable` package allows a larger range of ARIMA-type models to be compared. The following code applies this function to each of the populations.

```
tmp <-
series_of_interest %>%
  as_tsibble(key = c(sex, x, country), index = year) %>%
  model(arima = ARIMA(ex ~ pdq(0:3, 1, 0:3))) %>%
  report() %>%
  select(sex, x, country, ar_roots, ma_roots ) %>%
  arrange(country, x, sex)
```

```
## Warning in report.mdl_df(.): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
tmp
```

```
## # A tibble: 28 x 5
##    sex         x country             ar_roots  ma_roots
##    <chr>   <dbl> <fct>               <list>    <list>
```

```
##  1 female     0 England & Wales   <cpl [1]> <cpl [0]>
##  2 male       0 England & Wales   <cpl [0]> <cpl [0]>
##  3 female    65 England & Wales   <cpl [1]> <cpl [0]>
##  4 male      65 England & Wales   <cpl [1]> <cpl [0]>
##  5 female     0 Scotland          <cpl [0]> <cpl [1]>
##  6 male       0 Scotland          <cpl [0]> <cpl [0]>
##  7 female    65 Scotland          <cpl [2]> <cpl [2]>
##  8 male      65 Scotland          <cpl [0]> <cpl [2]>
##  9 female     0 Synthetic Germany <cpl [1]> <cpl [0]>
## 10 male       0 Synthetic Germany <cpl [0]> <cpl [0]>
## # ... with 18 more rows
```

The length of the vectors ar_roots and ma_roots indicate, respectively, how many ar or ma terms were identified in the best fitting model for the population indicated by sex, x (starting age) and country. For example, for females in England & Wales, from age 0, an ARIMA(1,1,0) model is preferred, whereas for females in Scotland, from age 0, an ARIMA(0, 1, 1) model is preferred.

There are few populations for which the random-walk-with-drift (RWD) model is preferred to more complex models, but also not a single alternative model specification (such as ARIMA(1,1,0)) which is preferred for the majority of populations.