EDSD 2024–2025: Demographic Forecasting's Assignment

Sanan Abdulayev Gilbert Habaasa Thiago C. Almeida

2025-06-11

Table of contents

1 Introduction

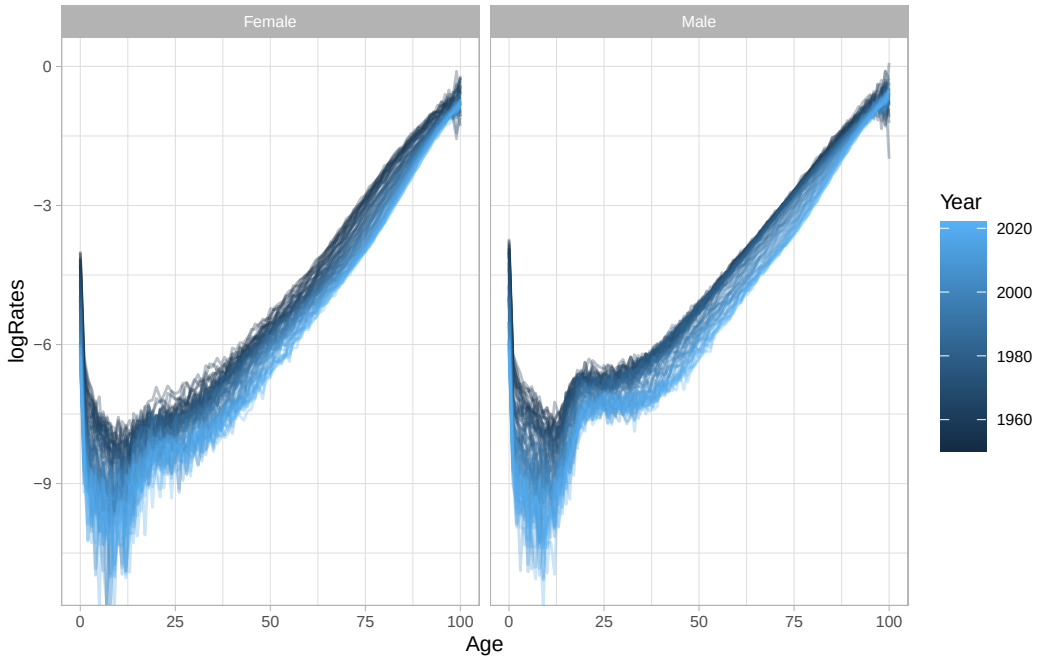
For this assignment, we will use data from Sweden, with mortality rates from 1771 to 2022 by age and sex.

```
load("../../../data/MortSWE.Rdata")
```

A general trend of these mortality rates from 1950 onwards can be seen in the graphic below.

```
MORT.SWE |>
  filter(Year >= 1950, Age <= 100) |>
  ggplot() +
  aes(x = Age, y = logRates, color = Year, group = interaction(Year, Year)) +
```

```
  geom_line(alpha = .3) +
  facet_wrap(.~Sex) +
  theme_light(base_size = graph_size)
```



2 Exercise 1

Q: *Load the mortality data MORTSWE.Rdata, and focus on male mortality from 1950 onwards for newborns (age 0). Fit and forecast mortality up to 2050 using two separate models:*

- *a linear model for death rates*
- *a generalized linear model for deaths, using exposures as an offset*

  Plot the fitted and forecast rates on a normal and log scale. What difference do you see between the two approaches?

For this exercise, we will work with male's mortality from 1950 onward for newborns (age 0).

```
mort_m0 <- MORT.SWE |>
  filter(
    Year >= 1950,
    Sex == "Male",
    Age == 0
```

```
  )
```

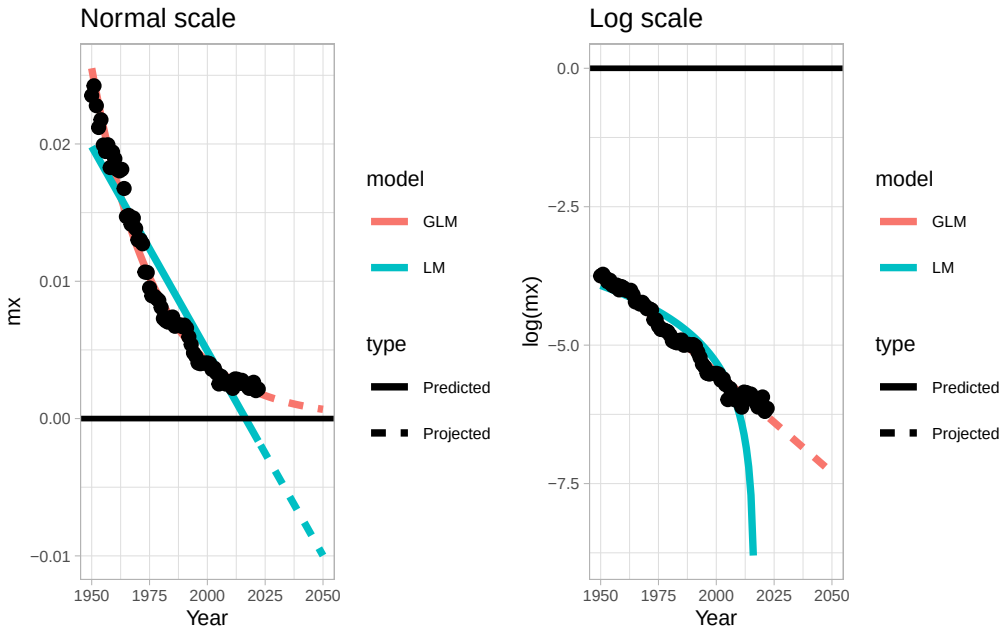
Now, let's fit and forecast mortality from 2022 up to 2050 using two different models:

- a linear model for death rates;
- a generalized linear model for deaths, using exposures as an offset.

```
### MODEL 1 - LM for death rates

mod1 <- lm(Rates ~ Year, data = mort_m0)

# summary(mod1)

## predicting values and extrapolating it until 2050

# dimensions of the problem
t <- mort_m0$Year
y <- mort_m0$Rates

# fit and focast of the model
tF <- min(t):2050 # creating the rage of years

df_forecast <- tibble(Year = tF) # creating df with rage of years

# forecasting years

yF <- predict(object = mod1, newdata = df_forecast)

# adding it to df

df_forecast <- df_forecast |>
  bind_cols(yF) |>
  select(Year, mx_lm = `...2`) |>
  mutate(
    type = case_when(Year <= max(t) ~ "Predicted", TRUE ~ "Projected")
  )

### MODEL 2 - GLM for deaths with exposure

# dimensions of the problem
t <- mort_m0$Year
y <- mort_m0$Deaths
e <- mort_m0$Exposures
tF <- min(t):2050 # creating the rage of years

# fit the model
```

```r
mod1.glm <- glm(y ~ Year, data = mort_m0, offset = log(e), family = poisson())

# summary(mod1.glm)

# forecasting years

df_forecast <- df_forecast |>
  mutate(
    mx_glm = exp(coef(mod1.glm)[1] + coef(mod1.glm)[2] * tF)
  ) |>
  rename(
    "LM" = mx_lm,
    "GLM" = mx_glm
  ) |>
  pivot_longer(
    cols = c(LM, GLM),
    values_to = "mx",
    names_to = "model"
  )

# plotting it...

plot1 <- ggplot() +
  geom_line(
    data = df_forecast,
    aes(x = Year, y = mx, linetype = type, color = model),
    linewidth = 1.3
  ) +
  geom_point(
    data = mort_m0,
    aes(x = Year, y = Rates),
    color = "black",
    size = 2
  ) +
  labs(title = "Normal scale") +
  geom_hline(yintercept = 0, color = "black", linewidth = 1) +
  theme_light(base_size = graph_size)

plot2 <- ggplot() +
  geom_line(
    data = df_forecast,
    aes(x = Year, y = log(mx), linetype = type, color = model),
    linewidth = 1.3
  ) +
  geom_point(
    data = mort_m0,
    aes(x = Year, y = logRates),
    color = "black",
```

```
    size = 2
  ) +
    labs(title = "Log scale") +
  geom_hline(yintercept = 0, color = "black", linewidth = 1) +
  theme_light(base_size = graph_size)

plot1 + plot2
```



For the LM, we are extrapolating our rates based on a linear relationship between mortality rates and time. However, in a scenario that the mortality is decreasing, this assumption makes our forecast estimates negative for mortality rates. This can seen in the left side of the normal scale, which is impossible to observe in real life. However, using log of mortality rates as dependent variable tackles this issue in GLM and forces the forecasted rates to be always over 0. As a result, we forecast the decreasing trends of mortality, as the observed values have shown, but with a lower pace and without becoming lower than zero.

On the other hand, for the GLM, the log of mortality rates has a linear trend over time, as we can see on the right figure above, while log rates predicted by LM declines rapidly towards negative infinity in 2023 as the value of corresponding mx gets closer to 0, and produce NAs afterwards as logarithm of negative values cannot be computed and do not appear in figure on the right.

3 Exercise 2

Q: *Load the mortality data MORTSWE.Rdata, and focus on male mortality from 1950 onwards for those aged 0-100 (i.e. exclude the 101-110+ age groups). Fit and forecast mortality up to 2050 using a generalized linear model for deaths, using exposures as an offset, for all age groups available (a for loop is a convenient way to do so). Compare the observed age-pattern of mortality against the fitted and forecast one. What differences do you observe?*

For this exercise, we will use not only the age 0, but all the range age between 0 and 100.

```
mort <- MORT.SWE |>
  filter(
    Year >= 1950,
    Sex == "Male",
    between(Age, 0, 100)
  )
```

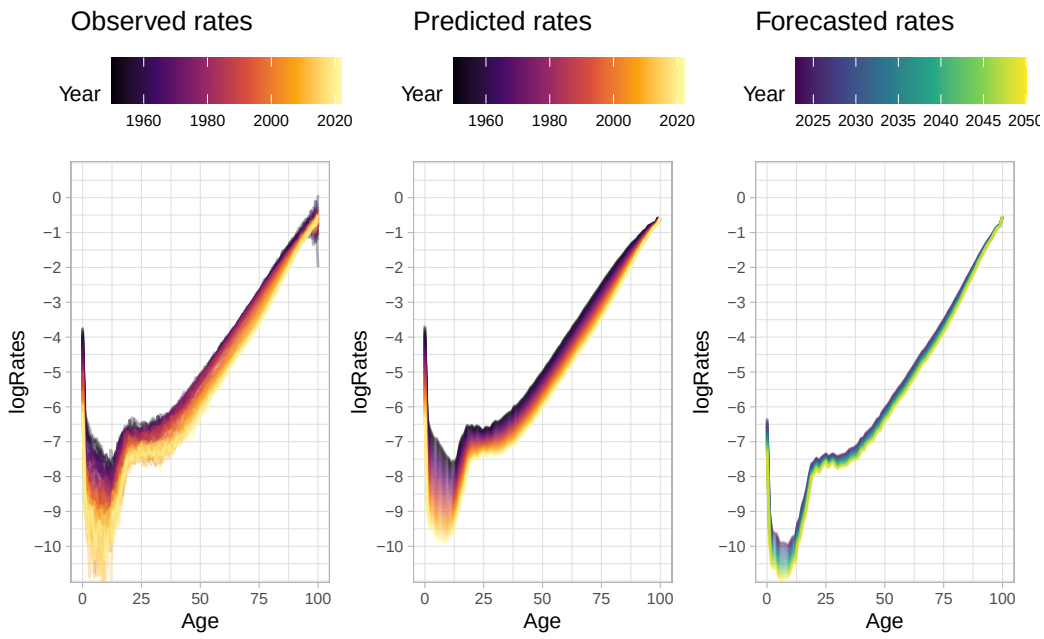
Let's fit a GLM model for deaths, using an offset, for all age groups available.

```
### MODEL - GLM for deaths with exposure

# dimensions of the problem
t <- unique(mort$Year)
tF <- min(t):2050 # creating the rage of years
age <- unique(mort$Age)
n <- length(age)
m <- length(tF)

# creating object for forecasting
mF <- matrix(data = NA, nrow = n, ncol = m)

# loop for predicted and forecasting mortality rates
i = 1
for(i in seq_along(age)){
  # setting remain dimensions of the problem
  y <- mort$Deaths[mort$Age == age[i]]
  e <- mort$Exposures[mort$Age == age[i]]
  # fit the model
  mod.glm <- glm(y ~ t, offset = log(e), family = poisson())
  #predict model
  mF[i,] <- exp(coef(mod.glm)[1] + coef(mod.glm)[2] * tF)
}
```

In order to compare the predicted values and the observed ones, we can see the panel of plots below.

```r
# Plot 1 - observed rates

plot1 <- mort |>
  ggplot() +
  # aes(x = Year, y = logRates, color = Age, group = interaction(Age, Age)) +
  aes(x = Age, y = logRates, color = Year, group = interaction(Year, Year)) +
  geom_line(alpha = .3) +
  scale_color_viridis_c(option = "B") +
  coord_cartesian(ylim = c(-10.5,0.5)) +
  scale_y_continuous(breaks = seq(-10,0,1)) +
  labs(title = "Observed rates") +
  theme_light(base_size = graph_size) +
  theme(
    legend.position = "top"
  )

# Plot 2 - predicted rates

colnames(mF) <- 1950:2050

plot2 <- mF |>
  as_tibble() |>
  mutate(
    Age = age
  ) |>
  pivot_longer(
    `1950`:`2050`,
    names_to = "Year",
    values_to = "mx_hat"
  ) |>
  mutate(Year = as.numeric(Year)) |>
  filter(Year <= 2022) |>
  mutate(
    logRates = log(mx_hat)
  ) |>
  ggplot() +
  aes(x = Age, y = logRates, color = Year, group = interaction(Year, Year)) +
  geom_line(alpha = .3) +
  scale_color_viridis_c(option = "B") +
  coord_cartesian(ylim = c(-10.5,0.5)) +
  scale_y_continuous(breaks = seq(-10,0,1)) +
  labs(title = "Predicted rates") +
  theme_light(base_size = graph_size) +
  theme(
    legend.position = "top"
  )

# Plot 3 - predicted and projected
```

```
plot3 <- mF |>
  as_tibble() |>
  mutate(
    Age = age
  ) |>
  pivot_longer(
    `1950`:`2050`,
    names_to = "Year",
    values_to = "mx_hat"
  ) |>
  mutate(Year = as.numeric(Year)) |>
  filter(Year > 2022) |>
  mutate(
    logRates = log(mx_hat)
  ) |>
  ggplot() +
  aes(x = Age, y = logRates, color = Year, group = interaction(Year, Year)) +
  geom_line(alpha = .3) +
  scale_color_viridis_c(option = "D") +
  coord_cartesian(ylim = c(-10.5,0.5)) +
  scale_y_continuous(breaks = seq(-10,0,1)) +
  labs(title = "Forecasted rates") +
  theme_light(base_size = graph_size) +
  theme(
    legend.position = "top"
  )

plot1 + plot2 + plot3
```



We observe that the fitted values are reflecting the observed rates quite precisely, while with much more smoother trend and less fluctuations particularly, in some young and very old ages. It is expected to see the decreasing trend of mortality for the forecasted rates, as the figure above suggests. However, further decrease for example among the very young ages, may not be plausible in real life since the current rates are already very low.

4 Exercise 3

Q: *Load the mortality data MORTSWE.Rdata, and focus on mortality from 1950 to 2000 for males aged 20 years. Fit and forecast mortality up to 2022 using two separate models:*

- *a random walk with drift model for log-rates the best possible*
- *ARIMA model for log-rates Plot the forecast rates*

  Plot the forecast rates against the subsequently observed data in 2001-2022. Which model seems to forecast mortality better?

We will work with the same data set, but with male mortality rates from 1950 to 2000 for age 20.

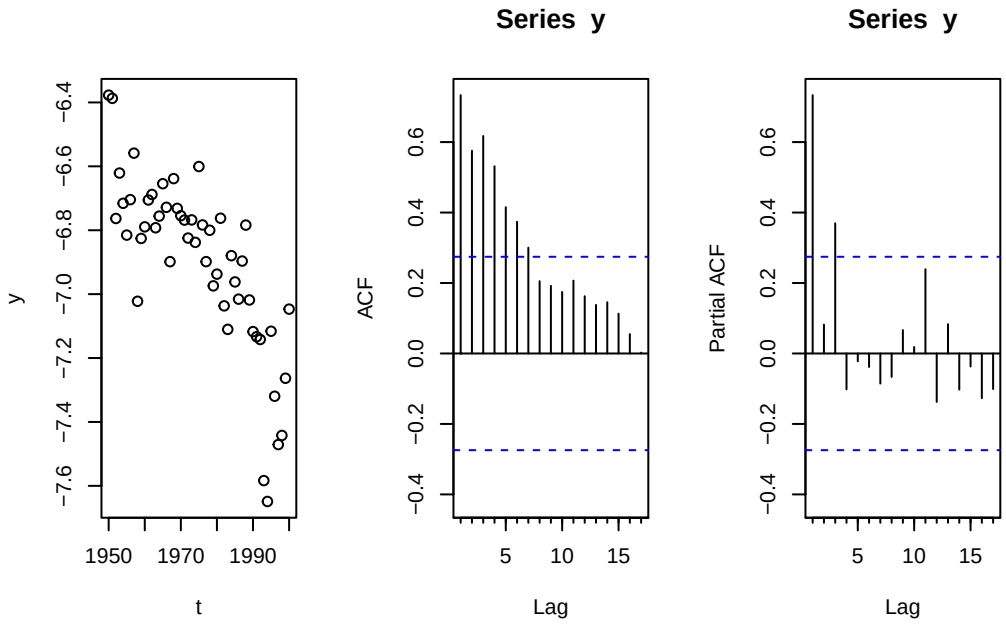
```
mort_m20 <- MORT.SWE |>
  filter(
    Year >= 1950,
    Sex == "Male",
    Age == 20
  )
```

Let's fit and forecast mortality up to 2022 using two different models

- a random walk with drift model for log-rates
- the best possible ARIMA model for log-rates

The first exercise is to describe the data, trying to interpret which kind of model will fit better in this case. To do so, we can use ACF plot and test for stationarity (KPSS).

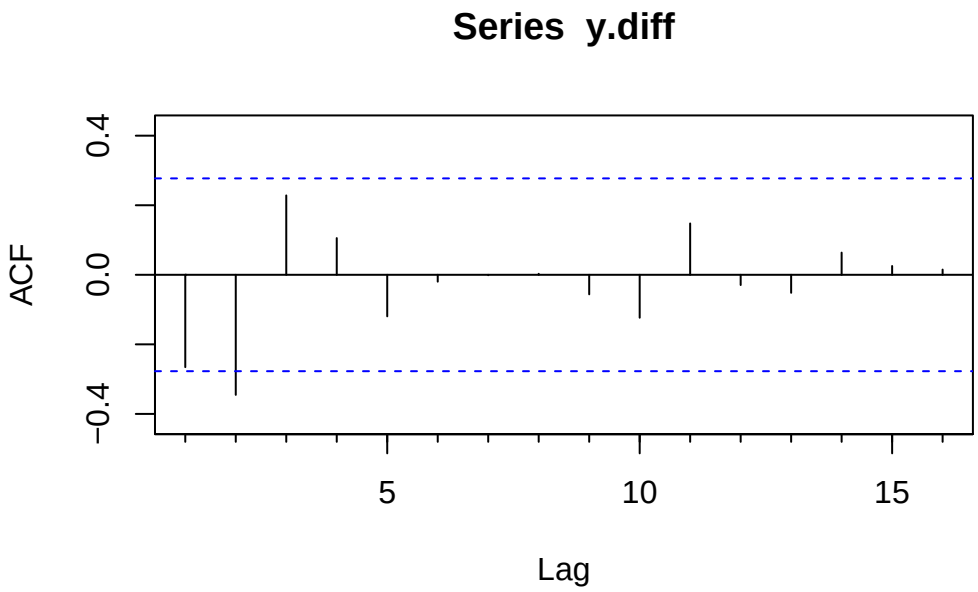
```
### MODEL 1 - Random walk with drift

# dimensions of the problem
t <- mort_m20$Year[mort_m20$Year <= 2000]
y <- mort_m20$logRates[mort_m20$Year <= 2000]
tF <- min(t):2022 # creating the rage of years

# plotting it...

par(mfrow = c(1,3))
# description of the data
plot(t,y)
# ACF
Acf(y)
# PACF
Acf(y,type = "partial")
```



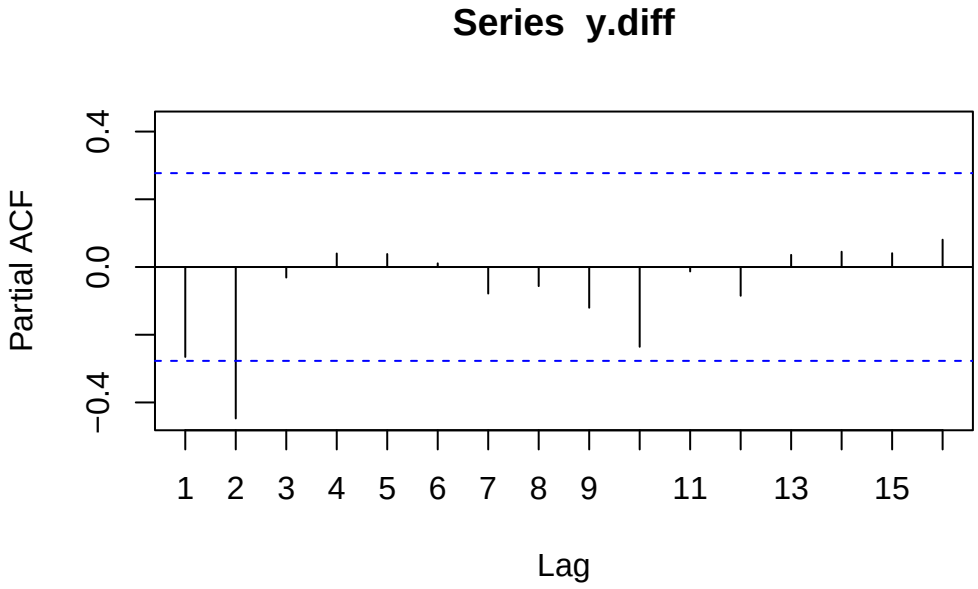

```
par(mfrow = c(1,1))
```

It seems that we are dealing with a non-stationary series. To understand better if we can make the series become stationary, let's take the first difference of the rates and plot the same graphs again.

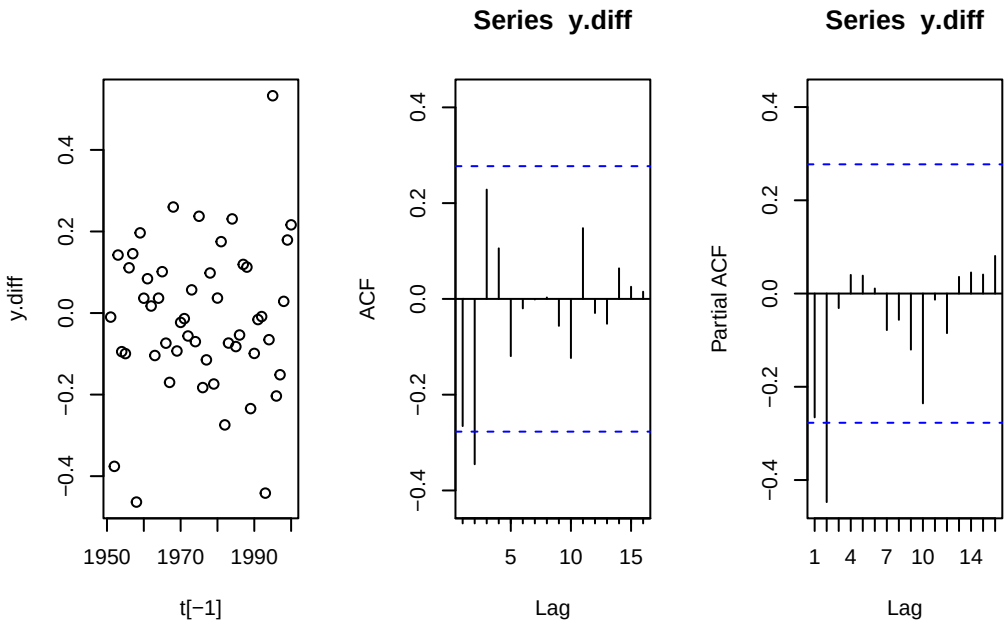
```
# taking the first-order diff...
y.diff <- diff(y)

# ACF
Acf(y.diff)
```




```
# PACF
Acf(y.diff,type = "partial")
```



```
# plotting it...

par(mfrow = c(1,3))
# description of the data
plot(t[-1],y.diff)
Acf(y.diff)
# PACF
Acf(y.diff,type = "partial")
```




```
par(mfrow = c(1,1))
```

Based on the first difference, we observe that the rates are more stationary over time. Also, based on the Partial ACF, we can see that it seems that there is an order three of dependency of previous values, which can be associated with a MA(3) model.

To be sure about the stationarity of the model considering the first difference, let's calculate the KPSS test for stationarity and seasonality.

```
# tests
tseries::kpss.test(y)
```

```
    KPSS Test for Level Stationarity

data:  y
KPSS Level = 1.0866, Truncation lag parameter = 3, p-value = 0.01
```

```
tseries::kpss.test(y.diff)
```

```
    KPSS Test for Level Stationarity

data:  y.diff
KPSS Level = 0.1031, Truncation lag parameter = 3, p-value = 0.1
```

So, we conclude that the first order difference would be a better option to guarantee the stationarity of this series.

```
## modeling it...

# RWD

mod.RWD <- Arima(
  y = y,
  order = c(0,1,0),
  include.drift = TRUE
)

# AR

mod.ar <- auto.arima(y)

# comparing models
modelsummary(list(mod.RWD,mod.ar),output = "markdown")
```

```
## comparing forecasts
t_ends = 2022
tF = t_ends - max(t)

mF.RWD <- forecast(mod.RWD, h = tF) |>
  as_tibble() |>
  mutate(year = seq(max(t)+1,t_ends,1))

mF.AR <- forecast(mod.ar, h = tF) |>
  as_tibble() |>
  mutate(year = seq(max(t)+1,t_ends,1))
```

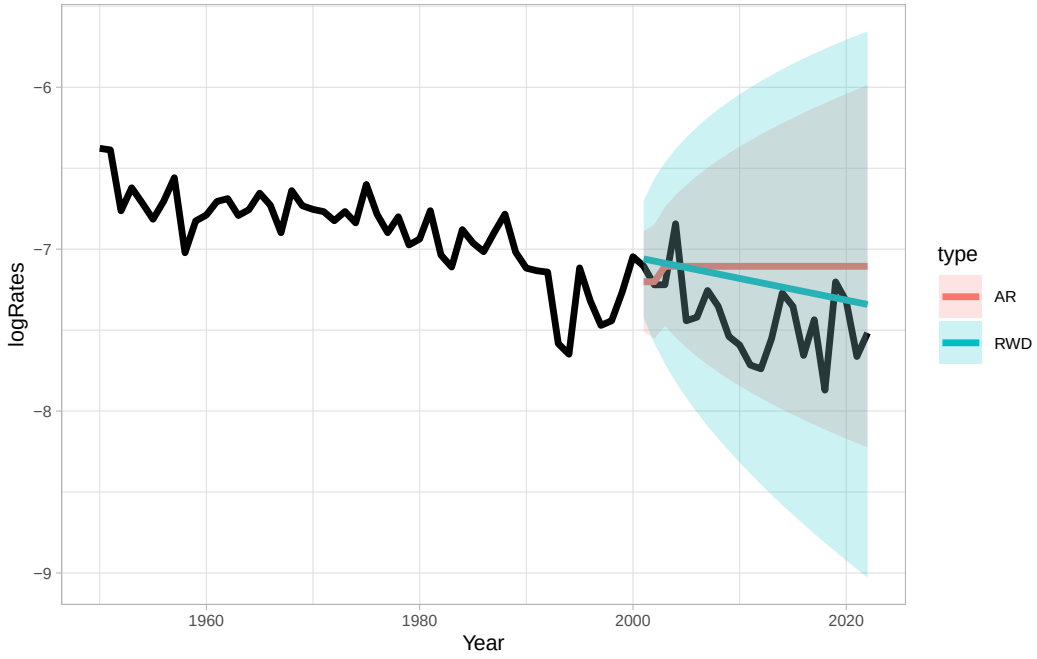
	(1)	(2)
drift	-0.013	
	(0.026)	
ma1		-0.448
		(0.131)
ma2		-0.233
		(0.143)
ma3		0.464
		(0.176)
Num.Obs.	50	50
AIC	-24.7	-36.8
BIC	-20.9	-29.1
RMSE	0.18	0.15
x		0.707717441673797

After modelling and comparing these models, we can see that adding three lagged observations to model the next one – ARIMA (0,1,3) – does not make the model more parsimonious than modeling it as a Random Walk with Drift (RWD) considering the first difference.

Now, let's plot our observed and fitted (predicted and forecasted values).

```
# creating a dataframe with all the estimates together

mF <- mF.RWD |>
  mutate(type = "RWD") |>
  bind_rows(
    mF.AR |>
  mutate(type = "AR")
  )


ggplot() +
  geom_line(
    data = mort_m20,
    aes(x = Year, y = logRates),
    linewidth = 1.2,
    color = "black"
  ) +
  geom_line(
    data = mF,
    aes(x = year, y = `Point Forecast`, color = type),
```

```
    linewidth = 1.2
  ) +
  geom_ribbon(
    data = mF,
    aes(
      x = year,
      ymin = `Lo 95`, ymax = `Hi 95`,
      fill = type
    ),
    alpha = 0.2
  ) +
  theme_light(base_size = graph_size)
```



The figure above shows that, in fact, the trend of decreasing mortality from 2000 onwards is better captured using RWD, based on the point estimates (solid lines). However, the RWD model also considers a higher uncertainty in these estimates when compared with ARIMA one, which can be a drawback of choosing the RWD model.

5 Exercise 4

Q: *Load the mortality data MORTSWE.Rdata, and focus on male mortality from 1950 onwards for those aged 0-100 (i.e. exclude the 101-110+ age groups). Fit and forecast mortality up to 2050 using the best possible ARIMA model for log-rates for all age groups available (a for*

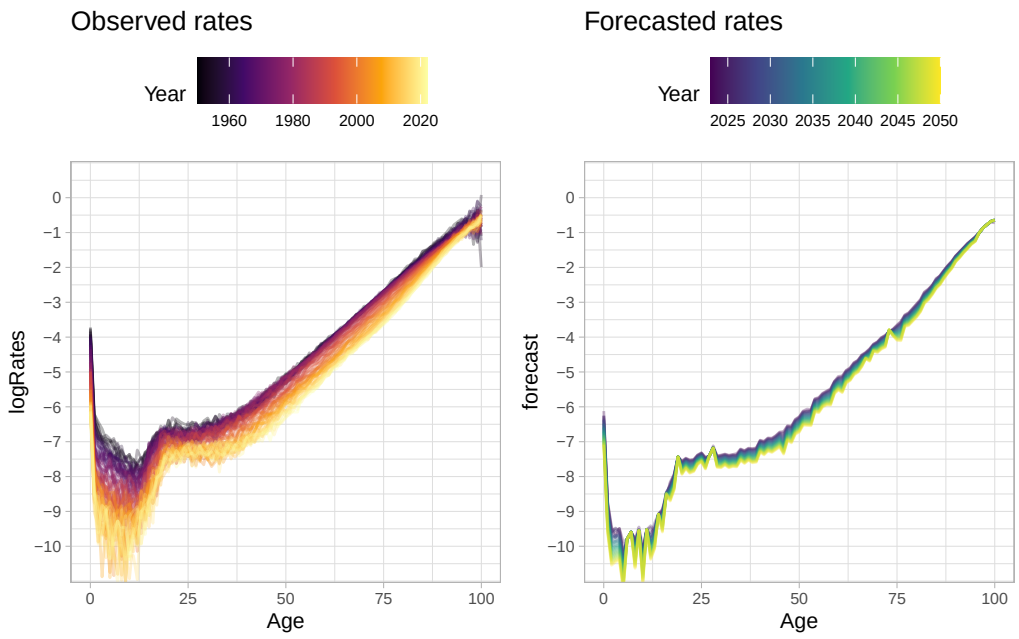
loop is a convenient way to do so). Compare the observed age-pattern of mortality against the forecast one. Does it seem like a reasonable pattern of mortality? [Hint: you will need to replace minus infinite values of log death rates with NA for some age groups]

```
#Filtering male mortality from 1950 onwards for those aged 0-100 (i.e. exclude the
 ↪  101-110+ age groups)

mort <- MORT.SWE |>
  filter(
    Year >= 1950,
    Sex == "Male",
    between(Age, 0, 100)
  )
```

```
mort <- mort |> mutate(yt=case_when(
  logRates== -Inf ~ -9.9772116 ,
  .default = logRates
))

# dimensions of the problem
t <- unique(mort$Year)
tF <- min(t):2050 # creating the rage of years
age <- unique(mort$Age)
n <- length(age)
m <- length(tF)
h <- length(tF) - length(t)

# creating object for forecasting
mF <- matrix(data = NA, nrow = n, ncol = m)

# loop for predicted and forecasting mortality rates
i = 1
for(i in seq_along(age)){

  # setting remain dimensions of the problem
  y <- mort$yt[mort$Age == age[i]]

  # fit the model
  mod.best <- auto.arima(y)

  lmort <- forecast(mod.best,h=h)
  #predict model
  mF[i,] <- c(y, lmort$mean)
}
```

Plotting it...

```r
# Plot 1 - observed rates

plot1 <- mort |>
  ggplot() +
  # aes(x = Year, y = logRates, color = Age, group = interaction(Age, Age)) +
  aes(x = Age, y = logRates, color = Year, group = interaction(Year, Year)) +
  geom_line(alpha = .3) +
  scale_color_viridis_c(option = "B") +
  coord_cartesian(ylim = c(-10.5,0.5)) +
  scale_y_continuous(breaks = seq(-10,0,1)) +
  labs(title = "Observed rates") +
  theme_light(base_size = graph_size) +
  theme(
    legend.position = "top"
  )


colnames(mF) <- 1950:2050

# Plot 2 - forecasted

plot2 <- mF |>
  as_tibble() |>
  mutate(
    Age = age
  ) |>
  pivot_longer(
    `1950`:`2050`,
    names_to = "Year",
    values_to = "forecast"
  ) |>
  mutate(Year = as.numeric(Year)) |>
  filter(Year > 2022) |>
  ggplot() +
  aes(x = Age, y = forecast, color = Year, group = interaction(Year, Year)) +
  geom_line(alpha = .3) +
  scale_color_viridis_c(option = "D") +
  coord_cartesian(ylim = c(-10.5,0.5)) +
  scale_y_continuous(breaks = seq(-10,0,1)) +
  labs(title = "Forecasted rates") +
  theme_light(base_size = graph_size) +
  theme(
    legend.position = "top"
  )

plot1 + plot2
```



Comparing the observed and forecasted rates, we can see that there is a continuation of decreasing trend of mortality across all ages, which we would expect based on the observed rates.

6 Exercise 5

Q: *Following up on Exercise 4, compute simulations for the future paths of all log death rate, and combine them together to compute a forecast of life expectancy. Use the provided function LifetableMX.R to construct a life table from your (log) rates. [Hint: you will need to create an array, rather than a matrix, where to store your simulations of log death rates]*

```
## function for constructing a classic (& rather general) lifetable
lifetable.mx <- function(x, mx, sex="M", ax=NULL){
  m <- length(x)
  n <- c(diff(x), NA)
  if(is.null(ax)){
    ax <- rep(0,m)
    if(x[1]!=0 | x[2]!=1){
      ax <- n/2
      ax[m] <- 1 / mx[m]
    }else{
      if(sex=="F"){
        if(mx[1]>=0.107){
```

```
          ax[1] <- 0.350
        }else{
          ax[1] <- 0.053 + 2.800*mx[1]
        }
      }
      if(sex=="M"){
        if(mx[1]>=0.107){
          ax[1] <- 0.330
        }else{
          ax[1] <- 0.045 + 2.684*mx[1]
        }
      }
      ax[-1] <- n[-1]/2
      ax[m] <- 1 / mx[m]
    }
  }
  qx  <- n*mx / (1 + (n - ax) * mx)
  qx[m] <- 1
  px  <- 1-qx
  lx  <- cumprod(c(1,px))*100000
  dx  <- -diff(lx)
  Lx  <- n*lx[-1] + ax*dx
  lx <- lx[-(m+1)]
  Lx[m] <- lx[m]/mx[m]
  Lx[is.na(Lx)] <- 0 ## in case of NA values
  Lx[is.infinite(Lx)] <- 0 ## in case of Inf values
  Tx  <- rev(cumsum(rev(Lx)))
  ex  <- Tx/lx
  return.df <- data.frame(x, n, mx, ax, qx, px, lx, dx, Lx, Tx, ex)
  return(return.df)
}

## function for constructing a classic (& rather general) lifetable
e0.mx <- function(x, mx, sex="M", ax=NULL){
  lt <- lifetable.mx(x,mx, sex,ax)
  return.ex <- lt$ex[1]
  return(return.ex)
}
```

```
n_sim <- 100  # number of simulations
set.seed(1)
tF <- 1950:2050
t <- 1950:2022
age <- unique(mort$Age)
h <- length(tF) - length(t)

# Creating array for simulation paths of dimensions [sim, age, year]
sim_paths <- array(NA, dim = c(n_sim, length(age), h))
```

```r
# 2. Simulate 100 future logrates for each age

for (i in seq_along(age)) {
  y <- mort$yt[mort$Age == age[i]]
  fit <- auto.arima(y)
  for (simn in 1:n_sim) {
  sim_result <- simulate(fit, nsim = h, future = TRUE, bootstrap = TRUE)
  sim_paths[simn, i, ] <- sim_result
  }
}


# 3. Compute life expectancy for each simulation using the provided code

e0_forecast <- matrix(NA, nrow = n_sim, ncol = h)

for (sim in 1:n_sim) {
  for (year in 1:h) {
    logmx <- sim_paths[sim, , year]
    mx <- exp(logmx)
    e0_forecast[sim, year] <- e0.mx(age, mx, sex = "M")
  }
}

# 4. Compute observed life expectancy from 1950 to 2022

e0_obs <- numeric(length(t))
for (i in seq_along(t)) {
  logmx <- mort$yt[mort$Year == t[i]]
  mx <- exp(logmx)
  e0_obs[i] <- e0.mx(age, mx, sex = "M")
}

# 5. Calculate median and 95% CI

colnames(e0_forecast)<-2023:2050
years_forecast <- (max(t) + 1):max(tF)

e0_sim <- data.frame(
  Year = years_forecast,
  median = apply(e0_forecast, 2, median),
  lower = apply(e0_forecast, 2, quantile, prob = 0.025),
  upper = apply(e0_forecast, 2, quantile, prob = 0.975)
)


# 6. Plotting
```

```r
df_obs <- data.frame(Year = t, e0 = e0_obs)

median <- e0_sim$median
lower <- e0_sim$lower
upper <- e0_sim$upper
t.all <- 1950:2050
tF <- t.all[!t.all%in%t]

plot(t,e0_obs,ylim=range(e0_obs,lower,upper),xlim=range(t,tF))

## simulations
lines(tF,median,col=4,lwd=2)
lines(tF,upper,col=4,lwd=2,lty=2)
lines(tF,lower,col=4,lwd=2,lty=2)
```