

Using the Earth Mover's Distance (EMD) for mortality analyses

Mathematical description

To describe the Wasserstein Distance or Earth Mover's Distance (EMD), we follow the equations and the example provided by (Weng 2019). The EMD can be interpreted “as the minimum energy cost of moving and transforming a pile of dirt in the shape of one probability distribution to the shape of the other distribution. The cost is quantified by: the amount of dirt moved \times the moving distance” (Weng 2019, pp. 8-9). For instance, if we have two distributions P and Q with a total amount of dirt of 10 each,

$$\begin{aligned} P_1 &= 3, P_2 = 2, P_3 = 1, P_4 = 4 \\ Q_1 &= 1, Q_2 = 2, Q_3 = 4, Q_4 = 3, \end{aligned}$$

and we want to make both distributions equal. We would need to change the distributions so that P_i matches Q_i . After defining the cost to make $P_i = Q_i$ as δ_i , we can calculate δ_i as $\delta_{i-1} = \delta_i + P_i - Q_i$. Accordingly, the cost in this example is,

$$\begin{aligned} \delta_0 &= 0 \\ \delta_1 &= 0 + 3 - 1 = 2 \\ \delta_2 &= 2 + 2 - 2 = -2 \\ \delta_3 &= 2 + 1 - 4 = -1 \\ \delta_4 &= -1 + 4 - 3 = 0. \end{aligned}$$

Hence, the EMD is given by summing up the cost,

$$EMD = \sum_{i=1}^n |\delta_i| = 5.$$

In continuous time, Weng (2019) defines the EMD as,

$$EMD(p_r, p_g) = \inf_{\gamma} \gamma \sim \Pi(p_r, p_g) \mathbb{E}_{(x,y) \sim \gamma} [|x - y|],$$

where $\Pi(p_r, p_g)$ is the set of all possible joint probability distributions between p_r and p_g and $\gamma(x, y)$ denotes the percentage of dirt that needs to be transported from x to y in order to make x follow the same probability distribution as y . Further, the travelling distance of piles of dirt is given by $|x - y|$ and hence, the expected cost averaged across all (x, y) pairs is computed by $\sum_{x,y} \gamma(x, y) |x - y| = \mathbb{E}_{(x,y) \sim \gamma} |x - y|$. Since we are only interested in “best” transport plan, i.e., the one with the smallest cost, the equation above uses \inf to mathematically describe the EMD.

In the one-dimensional setting, EMD can be calculated as the difference between the two corresponding cumulative distribution functions (CDF), P and Q ,

$$EMD(P, Q) = \int_{-\infty}^{+\infty} |P - Q|.$$

Please refer to Ramdas et al. (2015) for more information on the equivalence of both definitions.

In a life table with the radix equal to one, the age distribution of death can be seen as the probability density function, $f(x)$. Then, the survival function, $S(x)$, is the complement of the cumulative distribution function, $F(x)$,

$$\begin{aligned} S(x) &= 1 - \int_{-\infty}^x f(u) du \\ &= 1 - F(x). \end{aligned}$$

Hence, we can calculate EMD between two survival functions $S_A(x)$ and $S_B(x)$ as,

$$EMD(S_A(x), S_B(x)) = \int_{-\infty}^{+\infty} |S_A(x) - S_B(x)| dx$$

Implementation in R using two life tables from the Human Mortality Database

The data can be downloaded from mortality.org. Here, I am using the period life tables for Germany and the US in the year 1990 (women and men combined).

```
#read in data
Germany <- read.csv("bltper_1x1_Germany.txt", header = TRUE, skip = 2, sep="")
USA <- read.csv("bltper_1x1_USA.txt", header = TRUE, skip = 2, sep="")
```

```

#select year
qx_1990_Germany <- Germany$qx[Germany$Year == 1990]
qx_1990_USA <- USA$qx[USA$Year == 1990]
#get the life table survival function, lx
lx_1990_Germany <- c(1,
                      cumprod(1-qx_1990_Germany)[1:length(qx_1990_Germany)-1])

lx_1990_USA <- c(1,
                  cumprod(1-qx_1990_USA)[1:length(qx_1990_USA)-1])

#get the age distribution of deaths, dx
dx_1990_USA <- c(-diff(lx_1990_USA), lx_1990_USA[length(lx_1990_USA)])
dx_1990_Germany <- c(-diff(lx_1990_Germany),
                       lx_1990_Germany[length(lx_1990_Germany)])

```

The mean of the $d(x)$ function gives the average age at death or life expectancy at birth.

```
#check e0 result for the US
sum(dx_1990_USA * c(0:110+0.5)) /sum(dx_1990_USA)
```

```
[1] 75.39955
```

```
USA$ex[USA$Year == 1990][1]
```

```
[1] 75.4
```

```
#check e0 result for Germany
sum(dx_1990_Germany * c(0:110+0.5)) /sum(dx_1990_Germany)
```

```
[1] 75.35225
```

```
Germany$ex[Germany$Year == 1990][1]
```

```
[1] 75.35
```

Now, we calculate the EMD using the Wasserstein Distance function from the “transport” package.

```

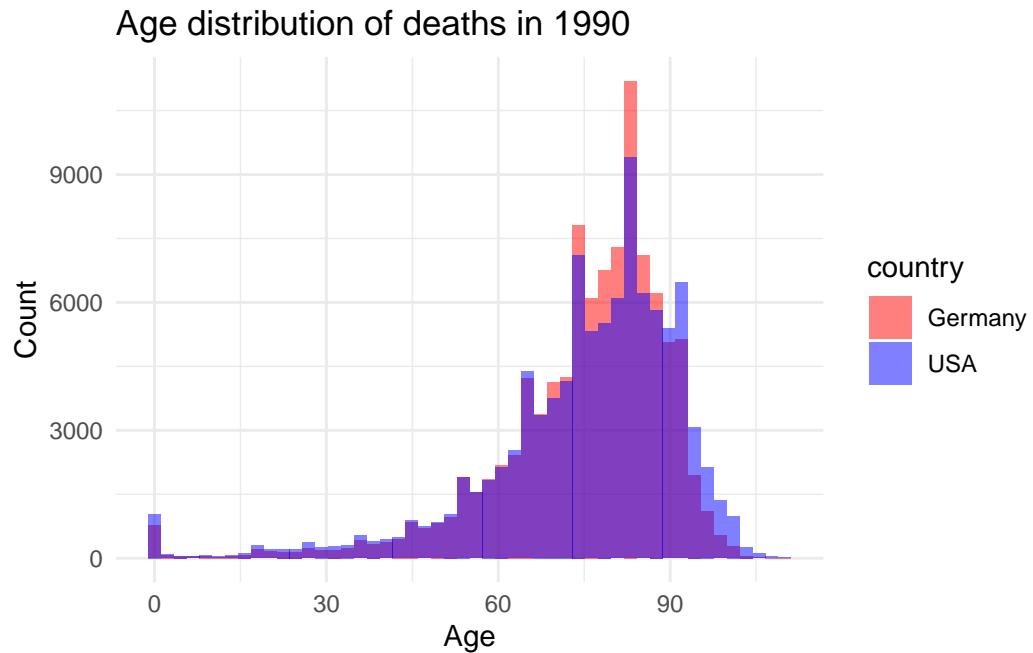
library(tidyr)
library(dplyr)
library(ggplot2)
library(transport)

sample_USA <- rep(0:110, times = round(dx_1990_USA * 100000))
sample_Germany <- rep(0:110, times = round(dx_1990_Germany * 100000))

df <- data.frame(
  age = c(sample_USA, sample_Germany),
  country = c(rep("USA", length(sample_USA)),
             rep("Germany", length(sample_Germany))))
)

#plot
ggplot(df, aes(x = age, fill = country)) +
  geom_histogram(position = "identity", alpha = 0.5, bins = 50) +
  labs(title = "Age distribution of deaths in 1990", x = "Age", y = "Count") +
  theme_minimal() +
  scale_fill_manual(values = c("USA" = "blue", "Germany" = "red"))

```



```
EMD_from_dx <- wasserstein1d(sample_USA, sample_Germany)
```

In the next step, we derive the same result by comparing the two survival functions. Please note that we have single-age groups in this example. If we had 5-year age groups, we would need to adjust for the age intervals in our calculation (see R code below).

```
CDF_1990_USA <- cumsum(dx_1990_USA)
Sx_1990_USA <- 1 - CDF_1990_USA
sum(Sx_1990_USA)+0.5 #again, e0 for the US in 1990
```

```
[1] 75.39955
```

```
CDF_1990_Germany <- cumsum(dx_1990_Germany)
Sx_1990_Germany <- 1 - CDF_1990_Germany
sum(Sx_1990_Germany) +0.5 #again, e0 for Germany in 1990
```

```
[1] 75.35225
```

```
#the difference in e0
round(USA$ex[USA$Year == 1990][1] - Germany$ex[Germany$Year == 1990][1], 2)
```

```
[1] 0.05
```

```
#can be defined as the difference in Sx
round(sum(Sx_1990_USA - Sx_1990_Germany), 2)
```

```
[1] 0.05
```

```
#EMD, however, is given by the absolute difference
abs_diff_CDF <- abs(CDF_1990_USA - CDF_1990_Germany)
#In case we do not have single ages,
#we need to multiply by the distance between ages.
#abs(CDF_1990_USA - CDF_1990_Germany) * distance_between_ages
abs_diff_Sx <- abs(Sx_1990_USA - Sx_1990_Germany)

EMD_from_CDF <- sum(abs_diff_CDF)
```

```
EMD_from_Sx <- sum(abs_diff_Sx)
```

```
EMD_from_dx
```

```
[1] 1.561458
```

```
EMD_from_CDF
```

```
[1] 1.562148
```

```
EMD_from_Sx
```

```
[1] 1.562148
```

Is there an upper bound for the EMD?

In principle, there is no upper bound for the EMD because the distance depends on the provided steps on the x-axis. The larger the step (here age intervals), the larger distance that needs to travel between the distributions. However, if we assume single-ages and an upper age limit of 100, the EMD's maximum value is 100. In this scenario, we compare the two distributions with highest dissimilarity. That is, the age at death distribution where everyone dies at the age of zero with the age at death distribution where everyone dies at the age of 100.

```
ages <- 0:100  
dx1 <- c(rep(0,length(ages)-1), 1)  
dx2 <- c(1, rep(0,length(ages)-1))
```

```
length(dx1) == length(dx2)
```

```
[1] TRUE
```

```
sample1 <- rep(ages, times = dx1 * 100000)  
sample2 <- rep(ages, times = dx2 * 100000)
```

```
wasserstein1d(sample1, sample2)
```

```
[1] 100
```

```
px1 <- 1-c(rep(0,length(ages)-1), 1)
px2 <- 1-c(1, rep(0,length(ages)-1))

lx1 <- c(1, (cumprod(px1))[1:(length(px1)-1)])
lx2 <- c(1, (cumprod(px2))[1:(length(px2)-1)])

dx1 <- c(-diff(lx1), lx1[length(lx1)])
dx2 <- c(-diff(lx2), lx2[length(lx2]))

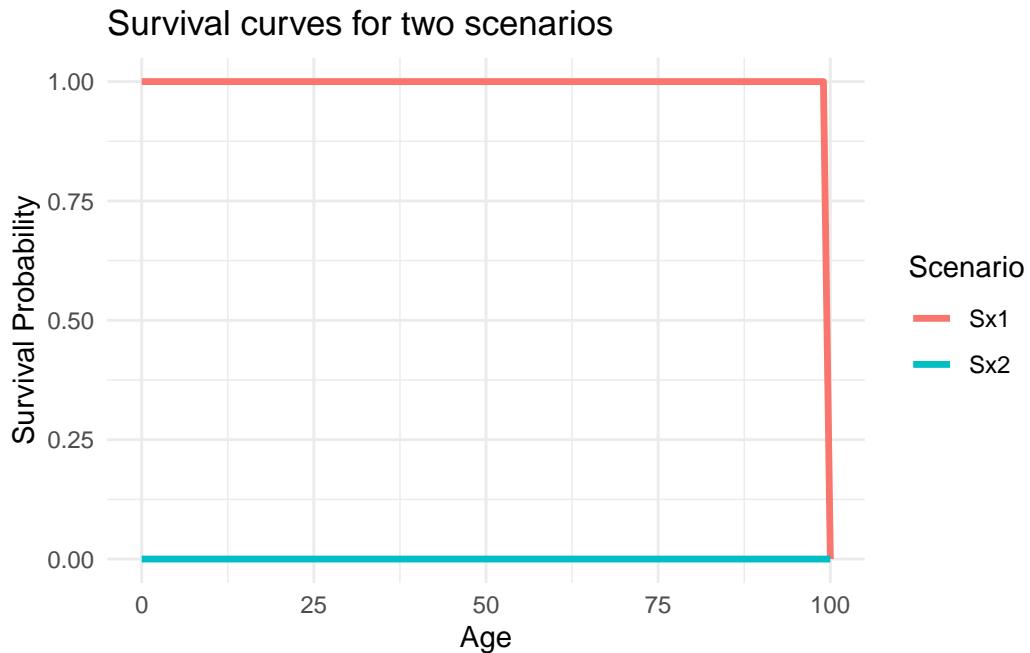
CDF1 <- cumsum(dx1)
CDF2 <- cumsum(dx2)

Sx1 <- 1 - CDF1
Sx2 <- 1 - CDF2

df <- data.frame(
  age = ages,
  Sx1 = Sx1,
  Sx2 = Sx2
)

df_long <- pivot_longer(df, cols = c("Sx1", "Sx2"),
                        names_to = "Scenario", values_to = "Survival")

#plot
ggplot(df_long, aes(x = age, y = Survival, color = Scenario)) +
  geom_line(linewidth = 1.2) +
  ylim(0, 1) +
  labs(title = "Survival curves for two scenarios",
       x = "Age", y = "Survival Probability") +
  theme_minimal()
```



```
#compute EMD
abs_diff_Sx <- abs(Sx1 - Sx2)
EMD <- sum(abs_diff_Sx)
EMD
```

```
[1] 100
```

In empirical populations, the highest observed EMD values are about 65. In the following example, we compare the EMD between the populations showing the highest life expectancy at birth values with the populations showing lowest life expectancy at birth values.

```
library(HMDHFDplus)
library(transport)

get_EMD <- function(qx1, qx2) {

  lx1 <- c(1,
          cumprod(1-qx1)[1:length(qx1)-1])
  dx1 <- c(-diff(lx1), lx1[length(lx1)])

  lx2 <- c(1,
```

```

        cumprod(1-qx2)[1:length(qx2)-1])
dx2 <- c(-diff(lx2), lx2[length(lx2)])

if (abs(sum(dx1) - 1) > 1e-6) {
  warning("dx1 do not sum to 1")
}

if (abs(sum(dx2) - 1) > 1e-6) {
  warning("dx2 do not sum to 1")
}

sample1 <- rep(0:110, times = round(dx1 * 100000))
sample2 <- rep(0:110, times = round(dx2 * 100000))
EMD <- wasserstein1d(sample1, sample2)

return(EMD)
}

#Here I load my HMD username and password
HMD_data <- read.table("my_HMD_data.txt",sep=",", header=TRUE)
my_username <- HMD_data$Name
my_password <- HMD_data$PW

compare_best_vs_worst_e0 <- function(item = "bltper_1x1",
                                         my_username, my_password,
                                         min_year=1751, rank_nr=100) {

  countries <- getHMDcountries()
  LT_df_country <- do.call(rbind,
                             lapply(countries$CNTRY,
                                    function(country_code){
  e0_df <- readHMDweb(country_code,
                        item = item,
                        my_username,
                        my_password)

  e0_df$country_code <- country_code
  e0_df[complete.cases(e0_df), ]
}))}

e0_df_country <- subset(LT_df_country, Age==0)

```

```

e0_selected <- subset(e0_df_country, Year>=min_year)
e0_selected <- e0_selected[,c("Year", "country_code", "ex")]
e0_selected <- e0_selected[complete.cases(e0_selected), ]
e0_summary <- unique(e0_selected[, c("Year", "country_code", "ex")])

sorted_e0 <- e0_summary[order(e0_summary$ex), ]
worst_cases <- sorted_e0[1:rank_nr, ]
best_cases <- head(sorted_e0[rev(1:nrow(sorted_e0)), ], n = rank_nr)

results_list <- list()

for (i in 1:rank_nr) {
  worst_row <- worst_cases[i, ]
  best_row <- best_cases[i, ]

  qx1 <- LT_df_country[LT_df_country$Year==worst_row$Year &
    LT_df_country$country_code==worst_row$country_code, ]

  qx2 <- LT_df_country[LT_df_country$Year==best_row$Year &
    LT_df_country$country_code==best_row$country_code, ]

  qx1 <- qx1[order(qx1$Age), "qx"]
  qx2 <- qx2[order(qx2$Age), "qx"]

  emd_value <- get_EMD(qx1, qx2)

  results_list[[i]] <- data.frame(
    year_low_rank = worst_row$Year,
    country_low_rank = worst_row$country_code,
    e0_low_rank = worst_row$ex,
    year_high_rank = best_row$Year,
    country_high_rank = best_row$country_code,
    e0_high_rank = best_row$ex,
    EMD = emd_value
  )
}

emd_results <- do.call(rbind, results_list)

return(emd_results)
}

```

```

men_100ranks <- compare_best_vs_worst_e0(item = "mltper_1x1",
                                         my_username, my_password,
                                         min_year=1751, rank_nr=100)

women_100ranks <- compare_best_vs_worst_e0(item = "fltpcr_1x1",
                                              my_username, my_password,
                                              min_year=1751, rank_nr=100)

women_100ranks[c(1:3),]

year_low_rank country_low_rank e0_low_rank year_high_rank country_high_rank
1            1773             SWE        18.83        2023           HKG
2            1882             ISL        19.01        2021           HKG
3            1846             ISL        19.19        2020           JPN
e0_high_rank      EMD
1            87.98 69.10641
2            87.93 68.84739
3            87.75 68.46334

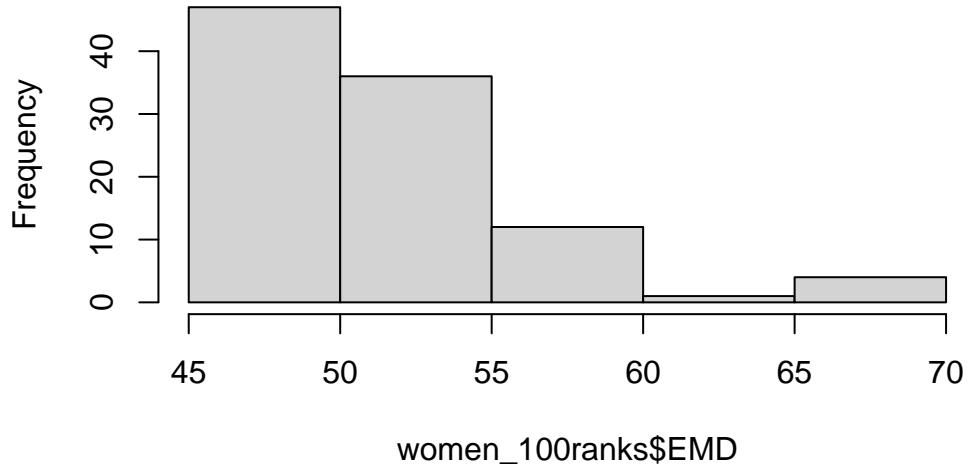
men_100ranks[c(1:3),]

year_low_rank country_low_rank e0_low_rank year_high_rank country_high_rank
1            1882             ISL        16.94        2023           HKG
2            1773             SWE        17.18        2021           HKG
3            1846             ISL        18.22        2023           CHE
e0_high_rank      EMD
1            82.48 65.45773
2            82.48 65.25055
3            82.21 63.88358

hist(women_100ranks$EMD, main="Women")

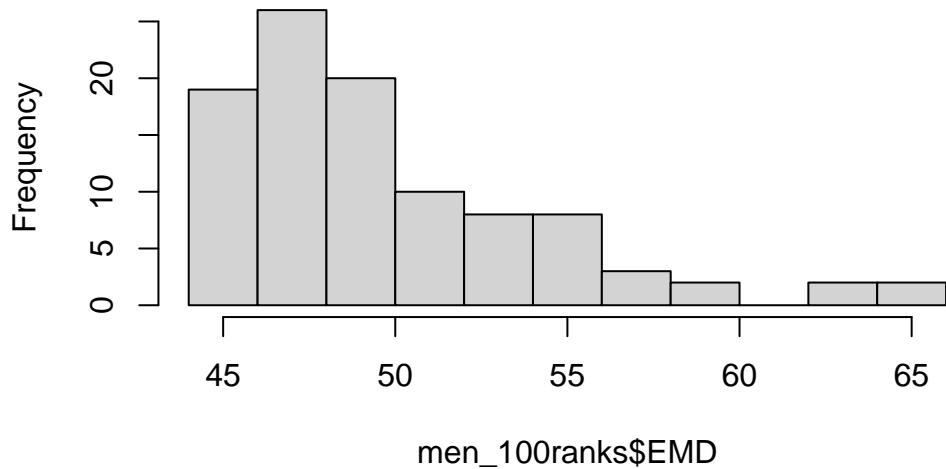
```

Women



```
hist(men_100ranks$EMD, main="Men")
```

Men



How do subpopulation-specific death rates affect the Earth Mover's distance?

As shown above, the EMD -in a one-dimensional setting- can be derived by calculating the difference between two CDFs or between two survival functions. In a life tale, the survival function is given by age-specific mortality rates,

$$S(x) = e^{\int_0^x \mu(a)da},$$

where $\mu(a)$ denotes the age-specific mortality rates. Further, the mortality rate observed in a given population is the weighted-average of group-specific mortality rates,

$$\mu(x) = \sum_{i=1}^n \mu_i(x) \cdot c_i(x),$$

where $\mu_i(x)$ is the age-specific mortality rate for group i and $c_i(x)$ the corresponding population weight.

Demographers have used this relationship to decompose differences in mortality indicators such as life expectancy at birth (e_0) or life disparity at birth (e^\dagger) into subnational contributions (Torres, Canudas-Romo, und Oeppen 2019; Su u. a. 2024). Alternatively, Andreev et al. (2002) have proposed the stepwise-replacement algorithm for solving demographic decomposition problems. The R implementation of the stepwise-replacement algorithm is provided by Riffe's "DemoDecomp" package. In the following, I will apply the stepwise-replacement R function to examine the EMD between Germany's mortality level (measured through the life table survival function) in 1990 and 2019 in terms of its east-west contributions. Data is again taken from [mortality.org](#).

```
#read in age-specific deaths and exposures for
#eastern and western Germany for the years 1990 and 2019
Dx_West <- read.csv("Deaths_1x1_WestGermany.txt",
                     sep="", header=TRUE, skip=2)
Dx_West_1990 <- subset(Dx_West, Year==1990)$Total
Dx_West_2019 <- subset(Dx_West, Year==2019)$Total

Dx_East <- read.csv("Deaths_1x1_EastGermany.txt",
                     sep="", header=TRUE, skip=2)
Dx_East_1990 <- subset(Dx_East, Year==1990)$Total
Dx_East_2019 <- subset(Dx_East, Year==2019)$Total

Ex_West <- read.csv("Exposures_1x1_WestGermany.txt",
                     sep="", header=TRUE, skip=2)
Ex_West_1990 <- subset(Ex_West, Year==1990)$Total
```

```

Ex_West_2019 <- subset(Ex_West, Year==2019)$Total

Ex_East <- read.csv("Exposures_1x1_EastGermany.txt",
                     sep="", header=TRUE, skip=2)
Ex_East_1990 <- subset(Ex_East, Year==1990)$Total
Ex_East_2019 <- subset(Ex_East, Year==2019)$Total

#calculate death rates as Dx/Ex
Mx_East_1990 <- Dx_East_1990 / Ex_East_1990
Mx_East_1990[is.na(Mx_East_1990)] <- 0

Mx_West_1990 <- Dx_West_1990 / Ex_West_1990
Mx_West_1990[is.na(Mx_West_1990)] <- 0

Mx_East_2019 <- Dx_East_2019 / Ex_East_2019
Mx_East_2019[is.na(Mx_East_2019)] <- 0

Mx_West_2019 <- Dx_West_2019 / Ex_West_2019
Mx_West_2019[is.na(Mx_West_2019)] <- 0

#calculate the population weights
Weight_East_1990 <- Ex_East_1990 / c(Ex_East_1990 + Ex_West_1990)
Weight_West_1990 <- Ex_West_1990 / c(Ex_East_1990 + Ex_West_1990)

Weight_East_2019 <- Ex_East_2019 / c(Ex_East_2019 + Ex_West_2019)
Weight_West_2019 <- Ex_West_2019 / c(Ex_East_2019 + Ex_West_2019)

#calculate weighted-death rates for eastern and western Germany
rates_weighted_1990 <- c(Mx_East_1990 * Weight_East_1990,
                           Mx_West_1990 * Weight_West_1990)

rates_weighted_2019 <- c(Mx_East_2019 * Weight_East_2019,
                           Mx_West_2019 * Weight_West_2019)

#function to derive the life table survival function
#from a vector with weighted death rates for eastern
#and western Germany
get_weighted_Sx <- function(mx_weighted_vec) {
  n <- length(mx_weighted_vec) / 2
  mx_east <- mx_weighted_vec[1:n]
  mx_west <- mx_weighted_vec[(n + 1):(2 * n)]
}

```

```

mx_total <- mx_east + mx_west

#convert mx to qx
qx <- mx_total / (1 + 0.5 * mx_total)

#compute life table lx (here denotes as Sx) with radix = 1
Sx <- c(1, cumprod(1 - qx)[1:(length(qx) - 1)])
return(Sx)
}

#load stepwise_replacement function from DemoDecomp
library(DemoDecomp)

#compute survival curves
Sx_1990 <- get_weighted_Sx(rates_weighted_1990)
Sx_2019 <- get_weighted_Sx(rates_weighted_2019)

#Wasserstein distance as distance between survival curves
wasserstein_distance <- sum(abs(Sx_1990 - Sx_2019))

#Decomposition function for stepwise_replacement()
func_total_abs_diff <- function(mx_weighted_vec) {
  Sx <- get_weighted_Sx(mx_weighted_vec)
  sum(abs(Sx - Sx_1990))
}

#Apply weighted death rates for eastern
#and western Germany to the function
contribs <- stepwise_replacement(
  func = func_total_abs_diff,
  pars1 = rates_weighted_1990,
  pars2 = rates_weighted_2019
)

#assign east-west contributions from results vector
n <- length(contribs) / 2
contrib_east <- sum(contribs[1:n])
contrib_west <- sum(contribs[(n + 1):(2 * n)])
sum_contribs <- contrib_east + contrib_west

#compare results

```

```

paste("EMD (total):", round(wasserstein_distance, 6))

[1] "EMD (total): 5.851911"

paste("Sum of east + west contributions:", round(sum_contribs, 6))

[1] "Sum of east + west contributions: 5.851911"

paste("East contribution:", round(contrib_east, 6))

[1] "East contribution: 1.514057"

paste("West contribution:", round(contrib_west, 6))

[1] "West contribution: 4.337854"

```

The exercise reveals that the difference in Germany's survival functions between 1990 and 2019 is mostly driven by western Germany. This is not surprising due to western Germany's higher population weights, i.e., age-specific death rates for eastern Germany are weighted less favorable due to their smaller population size.

The Earth Mover Distance in a two-dimensional setting

We can use the EMD to compare two cause-specific death distributions. To derive the cause-specific life table $d_i(x)$, we use multi-decrement life tables. As described by Preston et al. (2001), $d_i(x)$ can be calculated by applying cause-specific fractions to the total $d(x)$,

$$d_i(x) = d(x) \cdot \frac{m_i(x)}{m(x)}$$

where $m_i(x)$ denotes the cause-specific death rate at age x and $m(x)$ refers to the age-specific death rate for all causes combined.

Accordingly, a multi-decrement life table can be constructed on the basis of cause- and age-specific death rates. First, we use the all-cause death rates to derive conventional $d(x)$. Second, cause-specific fractions are calculated by dividing cause-specific death rates by all-cause death rates. Then, we can use these fractions to derive cause-specific life table deaths. If we set the life table radix to one, the sum of $d_i(x)$ over all causes and ages equals one. Hence, the $d_i(x)$ distribution can be seen as a probability distribution.

```

library(emdist)

get_cause_dx_dist <- function(HMD_file, the_year, the_sex) {

  df <- HMD_file %>%
    mutate(across(starts_with("m"), ~ ifelse(. == ".", "0", .))) %>%
    mutate(across(starts_with("m"), ~ as.numeric(.))) %>%
    filter(year == the_year & sex == the_sex) %>%
    select(-c(list, agf, total)) %>%
    pivot_longer(
      cols = starts_with("m"),
      names_to = "age_group",
      values_to = "mx"
    )
  #convert death rates per 100,000 to per person
  df$mx <- df$mx / 100000

  #clean up age group labels to get numeric age
  df$age_group <- as.numeric(gsub("[a-zA-Z]", "", df$age_group))

  #group causes into very simple categories
  df$cause <- case_when(
    df$cause %in% c("S002") ~ "Cancer",
    df$cause %in% c("S007", "S008", "S009") ~ "CVD",
    df$cause %in% c("S016") ~ "External",
    TRUE ~ "Rest"
  )

  #aggregate death rates by age and cause
  df_agg <- df %>%
    group_by(country, year, sex, cause, age_group) %>%
    summarise(mx = sum(mx), .groups = "drop") %>%
    arrange(cause, age_group)

  #compute total mx for each age group
  total_mx <- df_agg %>%
    group_by(age_group) %>%
    summarise(mx_total = sum(mx), .groups = "drop")

  #merge and compute fraction of each cause
  df_agg <- df_agg %>%

```

```

left_join(total_mx, by = "age_group") %>%
  mutate(fraction_cause = mx / mx_total) %>%
  mutate(fraction_cause = ifelse(is.nan(fraction_cause),
                                0,
                                fraction_cause))

#estimate life table qx, lx, and dx
age_vector <- sort(unique(df_agg$age_group))
n <- c(diff(age_vector), 999)

total_mx <- total_mx %>%
  arrange(age_group) %>%
  mutate(ax = n/2,
        n = n,
        qx = n * mx_total / (1 + (n - ax) * mx_total),
        qx = pmin(qx, 1))

#use radix = 1
total_mx$lx <- cumprod(c(1, head(1 - total_mx$qx, -1)))
total_mx$dx <- total_mx$lx * total_mx$qx

#merge dx into cause-specific df
df_agg <- df_agg %>%
  left_join(total_mx[, c("age_group", "dx")], by = "age_group") %>%
  mutate(dx_cause = dx * fraction_cause)

#prepare output
df_dist <- df_agg %>%
  select(age_group, cause, dx_cause) %>%
  group_by(age_group, cause) %>%
  summarise(prob = sum(dx_cause), .groups = "drop") %>%
  arrange(age_group, cause)

#validate that dx distribution sums to 1
if (abs(sum(df_dist$prob) - 1) > 1e-6) {
  warning("Cause-specific dx do not sum to 1")
}

return(df_dist)
}

```

```

USA_CoD <- read.csv("USA_m_short_idr.csv")
FR_CoD <- read.csv("FRATNP_m_short_idr.csv")

dx_dist_US <- get_cause_dx_dist(USA_CoD, the_year=2019, the_sex=3)
dx_dist_France <- get_cause_dx_dist(FR_CoD, the_year=2019, the_sex=3)
sum(dx_dist_US$prob)

[1] 1

sum(dx_dist_France$prob)

[1] 1

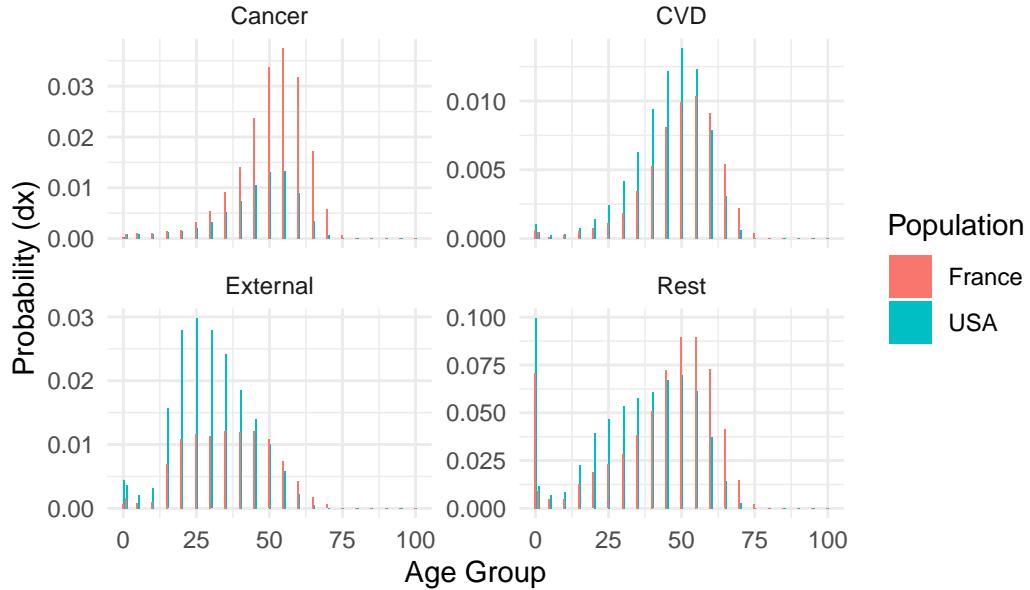
dx_dist_US$population <- "USA"
dx_dist_France$population <- "France"

# Combine both
df_plot <- bind_rows(dx_dist_US, dx_dist_France)

# Plot
ggplot(df_plot, aes(x = age_group, y = prob, fill = population)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ cause, scales = "free_y") +
  labs(
    title = "Cause-specific Life Table Deaths (dx) by Age Group",
    x = "Age Group",
    y = "Probability (dx)",
    fill = "Population"
  ) +
  theme_minimal()

```

Cause-specific Life Table Deaths (d_x) by Age Group



Next, we use the 2-dimensional EMD to examine the difference between the $d_i(x)$ distributions in the US and France for the 2019. The `emd2d()` function requires two matrices - in our case the $d_i(x)$ - and optionally, we can define the corresponding distance matrix. The EMD treats the difference between the $d_i(x)$ distributions as a transport problem. Thus, the distance matrix determines how far the mass between two distributions needs to travel. We set the distance between causes as one, i.e., shifting mass between the causes is constant. However, our life table uses the age-groups (0,1,5,10,15...100+). The different length of age intervals should be reflected in the distance matrix.

```
prepare_distribution <- function(dx_dist_A, dx_dist_B) {
  full_grid <- expand.grid(
    age_group = union(dx_dist_A$age_group, dx_dist_B$age_group),
    cause = union(dx_dist_A$cause, dx_dist_B$cause)
  )

  dx_dist_A_full <- full_grid %>%
    left_join(dx_dist_A, by = c("age_group", "cause")) %>%
    mutate(prob = replace_na(prob, 0))

  dx_dist_B_full <- full_grid %>%
    left_join(dx_dist_B, by = c("age_group", "cause")) %>%
    mutate(prob = replace_na(prob, 0))
}
```

```

    return(list(A = dx_dist_A_full, B = dx_dist_B_full))
}

dx_to_matrix <- function(dx_dist) {
  dx_dist %>%
    pivot_wider(names_from = cause, values_from = prob, values_fill = 0) %>%
    arrange(age_group) %>%
    select(-age_group) %>%
    as.matrix()
}

# get matrix A and B
aligned <- prepare_distribution(select(dx_dist_US, -population),
                                select(dx_dist_France, -population))
A <- dx_to_matrix(aligned$A)
B <- dx_to_matrix(aligned$B)

stopifnot(all(dim(A) == dim(B)))

#ydist: Age distances is 5-year age groups
ages <- sort(unique(dx_dist_US$age_group))
ydist <- diff(c(ages, max(ages) + 5))

#xdist: Cause distances, we assume equal distance over all causes
xdist <- rep(1, 4)
emd_value <- emd2d(A, B, xdist = xdist, ydist = ydist)
paste("2D EMD:", round(emd_value, 6))

```

[1] "2D EMD: 8.794995"

References

- Andreev, Evgeny M., Vladimir Shkolnikov, und Alexander Begun. 2002. „Algorithm for decomposition of differences between aggregate demographic measures and its application to life expectancies, healthy life expectancies, parity-progression ratios and total fertility rates“. <https://www.demographic-research.org/articles/volume/7/14>.
- Ramdas, Aaditya, Nicolas Garcia, und Marco Cuturi. 2015. „On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests“. <https://arxiv.org/abs/1509.02237>.

- Samuel H. Preston, Michel Guillot, Patrick Heuveline. 2001. „Demography: Measuring and Modeling Population Processes“. <https://u.demog.berkeley.edu/~jrw/Biblio/Eprints/%20P-S/palloni.2001.pdf>.
- Su, Wen, Alyson van Raalte, José Manuel Aburto, und Vladimir Canudas-Romo. 2024. „Subnational contribution to life expectancy and life span variation changes: Evidence from the United States“. <https://www.demographic-research.org/articles/volume/50/22>.
- Torres, Catalina, Vladimir Canudas-Romo, und Jim Oeppen. 2019. „The contribution of urbanization to changes in life expectancy in Scotland, 1861-1910“. <https://www.tandfonline.com/doi/full/10.1080/00324728.2018.1549746>.
- Weng, Lilian. 2019. „From GAN to WGAN“. <https://arxiv.org/abs/1904.08994>.