

Physical inactivity attributable burden of disease on morbidity and mortality across Swiss language regions: a population-based modelling study

Technical report | DOI <https://doi.org/10.57187/5162>

GÜNTERT Lara^{1,2}, TAEYMANS Jan^{1,2}, BAUR Heiner¹, VAN DYCK Marcus³, LUTZ Nathanael^{1,2}

1 Bern University of Applied Sciences, School of Health Professions, Physiotherapy, Bern, Switzerland

2 Vrije Universiteit Brussel, Faculty of Sports and Rehabilitation Sciences, Brussels, Belgium

3 Apollon University for Health Care Management, Bremen, Germany

Preparation

Load libraries

```
library(dplyr)
library(readxl)
library(readr)
library(kableExtra)
library(ggplot2)
library(stringr)
library(tidyr)
library(openxlsx)
library(scales)
library(stats)
library(rtables)
library(tinytex)
```

```
library(tibble)
library(gt)
library(patchwork)
library(gridExtra)
library(devtools)
library(survey)
```

Load raw data

The following raw data will be loaded:

- risk ratios, extracted from literature and prepared in a previous step
- population data for females and males from the Federal Statistical Office (FSO)
- deaths statistics from the FSO
- prevalences for physical inactivity (PIA) from the Swiss Health Survey (SHS) 2022
- burden of disease (Years of Life Lost (YLL) and Years Lived with Disability (YLD)) from the Institute for Health Metrics and Evaluation (IHME)

```
#import risk ratios
riskratios <- read_excel("../data/riskratios_V2.xlsx")

#import population data from BFS
population2021_2023 <- read_excel("../data/population2021-2023.xlsx")

#import deaths BFS
deaths_canton <- read_excel("../data/deaths_canton_2022.xlsx")

ihme_gbd_data <- read_csv("../data/ihme_gbd_data/ihme-gbd-2022.csv")
```

Create dataframes and remove datasets that are not needed any more.

```
#dataframe for risk ratios
d.riskratios <- data.frame(riskratios)

#dataframe for deaths
d.deaths <- data.frame(deaths_canton)
```

```
# create df for prevalences raw data
d.prev_prep_ind <- data.frame(prev_prep_raw_ind)
d.prev_prep_sfb <- data.frame(prev_prep_raw_sfb)
d.prev_prep_tel <- data.frame(prev_prep_raw_tel)

#dataframe for DALYS
d.ihme_gbd_data <- data.frame(ihme_gbd_data)

#dataframe for population
d.population2021_2023 <- as.data.frame(population2021_2023)

n.diseases <- nrow(d.riskratios)

rm(ihme_gbd_data, deaths_canton, population2021_2023, riskratios,
    prev_prep_raw_ind, prev_prep_raw_sfb, prev_prep_raw_tel)
```

Process data

Risk Ratios

Mapping of Disease Names: Short disease labels from the dataset are mapped to their corresponding full names using a named vector. This ensures consistency and clarity in the presentation of results.

```
# create vectors for abbreviation and full name from d.riskratios
shortnames_diseases <- d.riskratios$disease
longnames_diseases <- d.riskratios$disease_full

# mapping full name <-> short name
mapping_disease <- setNames(longnames_diseases, shortnames_diseases)
```

Formatting of Risk Ratios: Risk ratios and their confidence intervals (CI) are combined into a single formatted string (rr_ci) for easier display and interpretation.

```
d.riskratios <- d.riskratios %>%
  mutate(
    rr_ci = paste0(sprintf("%.2f", rr),
      " [" , sprintf("%.2f", rr_lower), " , ",
      sprintf("%.2f", rr_upper), "]" ) )
```

Visualization of Risk Ratios: A point plot with error bars is created to visualize the risk ratios and their 95% CI for each disease for easier interpretation.

```
ggplot(d.riskratios, aes(x = disease, y = rr)) +
  geom_point() +
  geom_errorbar(
    aes(ymin = rr_lower, ymax = rr_upper),
    width = 0.2) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 1))
```

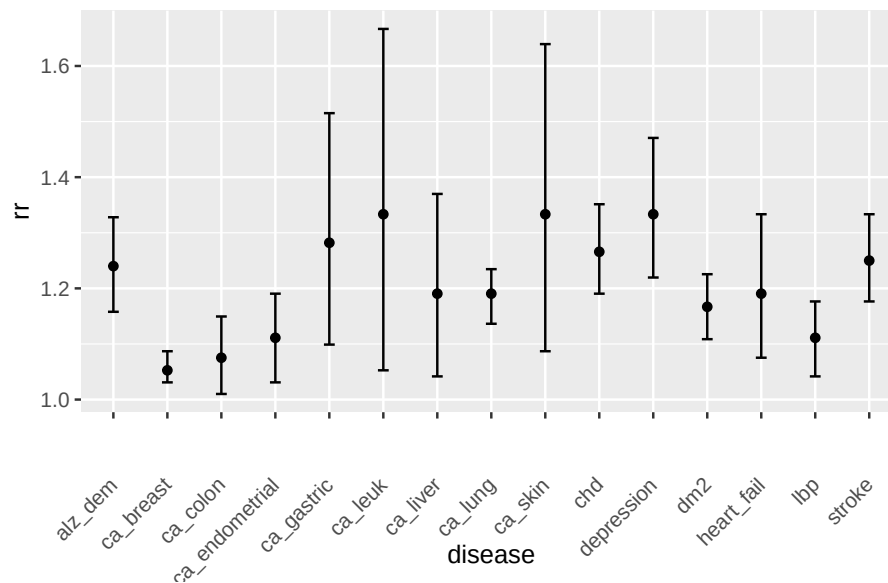


Figure 1: Risk ratios

Prevalences of physical inactivity 2022

Mapping of labels: Labels for sexes, regions and cantons from the dataset are mapped to their corresponding full names using a named vector. This ensures consistency and clarity in the presentation of results.

```
# create vectors for abbreviation and full name from datasets
# assign the vectors using setNames

# sex
```

```

shortnames_sex <- c("1", "2")
longnames_sex <- c("male", "female")
mapping_sex <- setNames(longnames_sex, shortnames_sex)

# language
shortnames_language <- c("1", "2", "3")
longnames_language <- c("german", "french", "italian")
mapping_language <- setNames(longnames_language, shortnames_language)

```

The order and assignment of labels follow the official coding and guidance provided by the Swiss Health Survey (SHS).

```

# canton
shortnames_canton <- c("1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26")

longnames_canton <- c("Zürich", "Bern / Berne", "Luzern", "Uri",
"Schwyz", "Obwalden", "Nidwalden", "Glarus", "Zug", "Fribourg",
"Solothurn", "Basel-Stadt", "Basel-Landschaft", "Schaffhausen",
"Appenzell Ausserrhoden", "Appenzell Innerrhoden", "St. Gallen",
"Grisson", "Aargau", "Thurgau", "Ticino", "Vaud", "Valais",
"Neuchâtel", "Genève", "Jura")

mapping_canton <- setNames(longnames_canton, shortnames_canton)

```

Merging Datasets by Participant ID: Data from the SHS 2022 were merged according to the official guidance provided by the FSO. Three datasets containing different sets of variables were merged using the unique participant identifier (IDNO). Prior to merging, only the relevant variables were retained, and each dataset was sorted by IDNO to ensure consistent alignment. In accordance with FSO recommendations, special attention was given to the assignment of survey weights. For analyses combining the telephone interview dataset and the written questionnaire dataset, the weight of the written survey was used, as advised, rather than the telephone survey weight. After the stepwise merging procedure, temporary objects created during data processing were removed.

```

#identification via IDNO
d.prev_prep_sfb <- d.prev_prep_sfb %>%
  select(IDNO, SALTER, SSEX, KANTON, WECRIT) %>%
  arrange(IDNO)

```

```

d.prev_prep_tel <- d.prev_prep_tel %>%
  select(IDNO, SEX, WGHT) %>%
  arrange(IDNO)

d.prev_prep_ind <- d.prev_prep_ind %>%
  select(IDNO, SPRACHE, ACTPHY4) %>%
  arrange(IDNO)

# Step 1: merge prev_prep_tel and prev_prep_sfb
d.prev_prep_merge1 <- merge(d.prev_prep_tel, d.prev_prep_sfb,
by = "IDNO", all = TRUE)

# Step 2: merge d.prev_prep_merge1 and prev_prep_ind
d.prev_prep_merge2 <- merge(d.prev_prep_merge1, d.prev_prep_ind,
by = "IDNO", all = TRUE)

#remove all datasets not needed anymore
rm(d.prev_prep_merge1, d.prev_prep_ind, d.prev_prep_sfb, d.prev_prep_tel)

```

Data Cleaning and Selection: From the merged SHS 2022 dataset, the variables relevant to this analysis (SSEX, SALTER, SPRACHE, KANTON, ACTPHY4) were selected. The dataset was then cleaned by excluding incomplete, inconsistent, or invalid entries. After applying these criteria, 18 760 of the original 21 930 records remained. For improved clarity and organisation within the analytical environment, the initial merged dataset was subsequently removed from the workspace.

```

#rename final dataframe
d.prev_pia <- d.prev_prep_merge2 %>%

# Select relevant columns
select(SSEX, SALTER, SPRACHE, KANTON, ACTPHY4, WECRIT) %>%

# Filter incomplete entries (no NA values and valid ranges)
filter(
  !is.na(SSEX) &
  !is.na(SALTER) & SALTER >= 0 &
  !is.na(SPRACHE) &
  !is.na(KANTON) &
  !is.na(ACTPHY4) & ACTPHY4 >= 0)

rm(d.prev_prep_merge2)

```

Classification by PIA and Age Groups: The dataset was extended by classifying individuals into activity levels and age groups. PIA was cate-

gorized according to WHO guidelines, with individuals performing less than 3 sessions per week labeled as inactive (ACTPHY2 = 1) and others as active (ACTPHY2 = 0). Additionally, participants were grouped into predefined 5-year age intervals based on their age (SALTER). This allows for stratified analysis by activity status and age.

```
#filter and summarise data
d.prev_pia <- d.prev_pia %>%

#Categorization in dichotomous variables:
#active and inactive according to WHO: 1 = inactive, 0 = active
#create a new variable ACTPHY2
#classify if less than 3 --> value 1 = inactive, else 0 = active
mutate(ACTPHY2 = ifelse(ACTPHY4 < 3, 1, 0)) %>%

#Categorization in age groups
mutate(age_group = case_when(
  SALTER >= 20 & SALTER <= 24 ~ "20-24",
  SALTER >= 25 & SALTER <= 29 ~ "25-29",
  SALTER >= 30 & SALTER <= 34 ~ "30-34",
  SALTER >= 35 & SALTER <= 39 ~ "35-39",
  SALTER >= 40 & SALTER <= 44 ~ "40-44",
  SALTER >= 45 & SALTER <= 49 ~ "45-49",
  SALTER >= 50 & SALTER <= 54 ~ "50-54",
  SALTER >= 55 & SALTER <= 59 ~ "55-59",
  SALTER >= 60 & SALTER <= 64 ~ "60-64",
  SALTER >= 65 & SALTER <= 69 ~ "65-69",
  SALTER >= 70 & SALTER <= 74 ~ "70-74",
  SALTER >= 75 ~ "75_plus",
  TRUE ~ "NA"))
```

Prevalence calculation by sex, age group and region: The prevalence of PIA was estimated using the recommended survey weights from the SHS to ensure representativeness of the Swiss population.

```
#weights
design <- svydesign(
  ids = ~1,
  weights = ~WECRIT,
  data = d.prev_pia)

d.prev_pia <- d.prev_pia %>%
  mutate(
    SSEX = factor(SSEX, labels = c("male", "female")),
    SPRACHE = factor(SPRACHE, labels = c("german", "french", "italian")),
```

```

    ACTPHY2 = factor(ACTPHY2, labels = c("active", "inactive")),
    age_group = factor(age_group,
      levels = c("20-24", "25-29", "30-34", "35-39", "40-44", "45-49",
        "50-54", "55-59", "60-64", "65-69", "70-74", "75_plus")))

design <- svydesign(ids = ~1, weights = ~WECRIT, data = d.prev_pia)

prop_by_group <- svyby(
  ~ACTPHY2,                # variable of interest
  ~SSEX + age_group + SPRACHE, # grouping variables
  design,
  svyciprop,
  method="logit",
  vartype = c("ci"),        # adds 95% confidence intervals
  na.rm = TRUE)

```

Label Mapping and Visualization of PIA: Sex and language codes were mapped to their descriptive labels using predefined mappings (mapping_sex and mapping_language). The variables were then renamed to sex and language for consistency. A stratified plot was created to visualize the prevalence of PIA by region, sex and age group.

```

d.prev_pia <- prop_by_group %>%
  rename(
    sex = SSEX,
    language = SPRACHE,
    prev = ACTPHY2,
    prev_lower = ci_l,
    prev_upper = ci_u
  ) %>%
  select(sex, language, prev, prev_lower, prev_upper, age_group)

ggplot(d.prev_pia, aes(x = age_group, y = prev, color = language)) +
  geom_point(position = position_dodge(width = 0.5), shape = 21) +
  geom_errorbar(
    aes(ymin = prev_lower, ymax = prev_upper),
    width = 0.2,
    position = position_dodge(width = 0.5)
  ) +
  facet_grid(~sex)

```

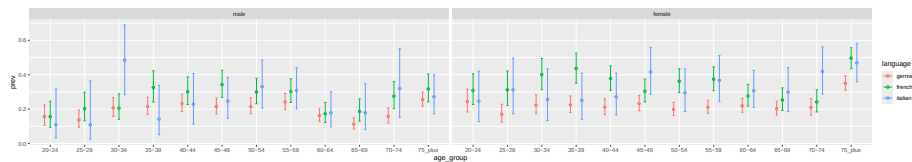


Figure 2: Prevalence of PIA by age, language and sex (2022)

Create global dataframe

A global dataframe (d.global) was created by combining prevalence data with disease-specific risk ratios. The dataset includes all combinations of disease, sex, age group, and language region. Relevant variables from both the prevalence and risk ratio datasets were repeated and merged systematically to ensure alignment across all dimensions. This structure enables comprehensive analyses of disease burden stratified by demographic and regional factors.

```
# create a global dataframe with all values
# Sort data frames using a consistent scheme
d.prev_pia <- d.prev_pia %>%
  arrange(language, sex, age_group)

# Create a dataframe with all entries; repeat values from the
# 'language' column as many times as there are diseases
# in the risk ratio dataset
d.global <- data.frame(
  language = factor(rep(d.prev_pia$language, length(d.riskratios$disease))))

# Fill the dataframe
# each = number of age groups
d.global <- d.global %>%
  mutate(
    disease = factor(rep(d.riskratios$disease,
                        each = nrow(d.global) /
                        length(d.riskratios$disease))),
    disease_group = factor(rep(d.riskratios$disease_group,
                              each = nrow(d.global) /
                              length(d.riskratios$disease))),
    age_group = factor(rep(d.prev_pia$age_group,
                          times = nrow(d.riskratios))))
```

```

sex = factor(rep(d.prev_pia$sex,
                 times = nrow(d.riskratios))),

riskratio = rep(d.riskratios$rr,
                each = nrow(d.prev_pia)),

rr_lower = rep(d.riskratios$rr_lower,
                each = nrow(d.prev_pia)),

rr_upper = rep(d.riskratios$rr_upper,
                each = nrow(d.prev_pia)),

prev = rep(d.prev_pia$prev,
            length(d.riskratios$disease)),

prev_lower = rep(d.prev_pia$prev_lower,
                  length(d.riskratios$disease)),

prev_upper = rep(d.prev_pia$prev_upper,
                  length(d.riskratios$disease))
) %>%

arrange(disease, language, sex, age_group) %>%
select(disease, language, sex, age_group, everything())

```

Visualization of the Global Dataset: A series of plots was created to visually inspect the structure and distribution of key variables in the global dataset. Histograms were generated for the continuous variables riskratio and prevalence of PIA to assess their distribution. Bar charts were produced for the categorical variables disease, age group, sex, and language to examine the frequency of observations across categories.

```

# Histograms
p1 <- ggplot(d.global, aes(x = riskratio)) +
  geom_histogram(binwidth = 0.1, color = "white") +
  labs(title = "Risk Ratio") +
  theme_minimal()

p2 <- ggplot(d.global, aes(x = prev)) +
  geom_histogram(binwidth = 0.05, color = "white") +
  labs(title = "Prevalence") +
  theme_minimal()

# Bar Charts for Categorical Variables
p3 <- ggplot(d.global, aes(x = disease)) +

```

```

geom_bar() +
labs(title = "Disease") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1))

p4 <- ggplot(d.global, aes(x = age_group)) +
geom_bar() +
labs(title = "Age Group") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1))

p5 <- ggplot(d.global, aes(x = sex)) +
geom_bar() +
labs(title = "Sex") +
theme_minimal()

p6 <- ggplot(d.global, aes(x = language)) +
geom_bar() +
labs(title = "Language") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 3)

```

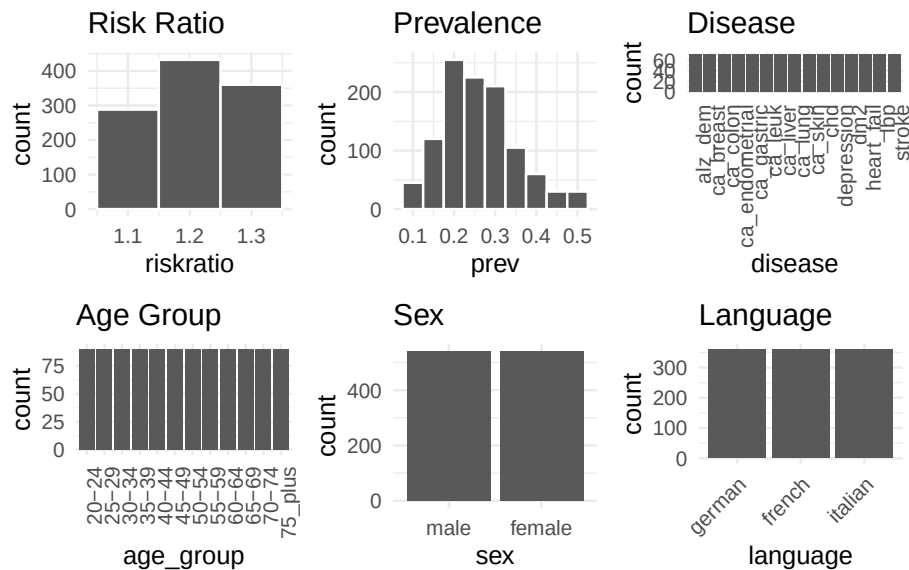


Figure 3: Graphics Global Dataset

Population Attributable Fraction Calculation

The Population Attributable Fraction (PAF) was calculated using Levin's formula, which estimates the proportion of disease cases that can be attributed to a specific risk factor in the population. The formula incorporates both the prevalence of exposure - here prevalence of PIA - and the associated risk ratio. The resulting PAF values were added to the global dataset.

A histogram was then generated to visualize the distribution of PAF values across all disease-region-sex-age combinations.

```
# Levin's formula
PAF2 <- function(prev, riskratio)
{ (prev*(riskratio-1)) / ((prev*(riskratio-1)+1)) }

# add column PAF in d.global
d.global <- d.global %>%
  mutate(paf = PAF2(prev, riskratio))
```

```
ggplot(d.global, aes(x = paf)) +
  geom_histogram(color = "white") +
  theme_minimal()
```

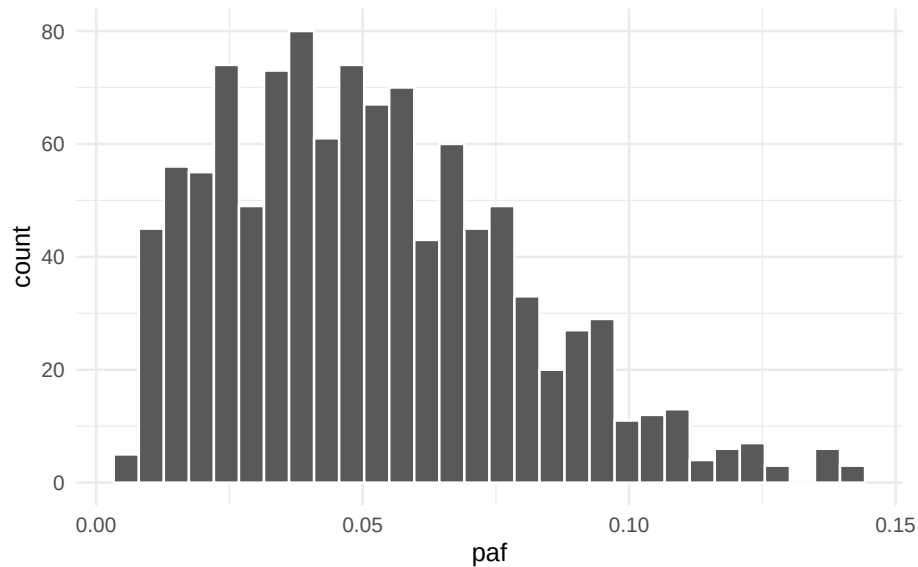


Figure 4: PAF

GBD Data: Preparation and filtering

The GBD (Global Burden of Disease) dataset (`d.ihme_gbd_data`) was pre-processed for integration and analysis.

Numeric Formatting: The `val`, `lower`, and `upper` columns—representing point estimates and confidence intervals—were converted to numeric and reformatted to avoid scientific notation. The `year` column was also cast to numeric.

```
d.ihme_gbd_data$val <- format(as.numeric(d.ihme_gbd_data$val),
                             scientific = FALSE)
d.ihme_gbd_data$lower <- format(as.numeric(d.ihme_gbd_data$lower),
                                scientific = FALSE)
d.ihme_gbd_data$upper <- format(as.numeric(d.ihme_gbd_data$upper),
                                scientific = FALSE)
d.ihme_gbd_data$year <- as.numeric(d.ihme_gbd_data$year)
```

Standardization of Labels: Sex labels were harmonized by recoding “Male” and “Female” to lowercase (“male”, “female”). Disease names (`cause_name`) were mapped to the naming conventions used in the internal dataset (`d.global`) to ensure consistency across data sources.

Filtering: Diseases not relevant to the current analysis (e.g. Rheumatic heart disease, Brain and central nervous system cancer) were excluded.

```
library(dplyr)
#harmonizing sex labels to global environment
d.ihme_gbd_data <- d.ihme_gbd_data %>%
  mutate(sex_name = recode(sex_name,
                           "Male" = "male",
                           "Female" = "female"))

#rename diseases
d.ihme_gbd_data <- d.ihme_gbd_data %>%
  mutate(cause_name = recode(cause_name,
                             "Alzheimer's disease and other dementias" = "alz_dem",
                             "Multiple myeloma" = "ca_skin",
                             "Leukemia" = "ca_leuk",
                             "Ischemic heart disease" = "chd",
                             "Stroke" = "stroke",
                             "Hypertensive heart disease" = "heart_fail",
                             "Stomach cancer" = "ca_gastric",
                             "Liver cancer" = "ca_liver",
                             "Breast cancer" = "ca_breast",
                             "Uterine cancer" = "ca_endometrial",
                             "Colon and rectum cancer" = "ca_colon",
                             "Diabetes mellitus type 2" = "dm2",
                             "Low back pain" = "lbp",
                             "Depressive disorders" = "depression",
                             "Tracheal, bronchus, and lung cancer" = "ca_lung"
  ))

d.ihme_gbd_data <- d.ihme_gbd_data %>%
  filter(cause_name != "Rheumatic heart disease",
         cause_name != "Brain and central nervous system cancer")
```

Subsetting by Burden Type (YLL+YLD): Three separate datasets were created:

- d.yll: Contains Years of Life Lost (YLLs).
- d.yld: Contains Years Lived with Disability (YLDs).

These datasets were filtered by measure_name and metric_name, and then sorted by cause, sex, and age group. This cleaned and harmonized data serves as a basis for further burden of disease calculations.

```

library(dplyr)
#YLL
d.yll <- d.ihme_gbd_data %>%
  filter(measure_name == "YLLs (Years of Life Lost)",
         metric_name == "Number") %>%
  dplyr::select(sex_name, age_name, cause_name, val, upper, lower) %>%
  arrange(cause_name, sex_name, age_name)
# some missing entries, not available on GBD

#YLD
d.yld <- d.ihme_gbd_data %>%
  filter(measure_name == "YLDs (Years Lived with Disability)",
         metric_name == "Number") %>%
  dplyr::select(sex_name, age_name, cause_name, val, upper, lower) %>%
  arrange(cause_name, sex_name, age_name)
# --> ca_endometrial only for females

```

Integration of YLD and YLL Data into the Global Dataset:

The YLD and YLL datasets were cleaned and aligned to match the format of the d.global dataset.

Standardization of Variables: Column names were harmonized (cause_name → disease, sex_name → sex, age_name → age_group). Age group labels were simplified (e.g. “45-49 years” to “45-49”, “75+” to “75_plus”).

Merging: The standardized YLD and YLL datasets were joined to d.global using a left join on disease, sex, and age_group. This added point estimates and confidence intervals for disease burden (columns val, lower, upper), which were then renamed for clarity (e.g., yld_val, yll_val).

Missing Value Handling: All missing values (NA) in the new YLD/YLL columns in d.global were replaced with zeroes to ensure completeness of downstream calculations.

```

# Standardization of Variables
d.yld <- d.yld %>%
  rename(disease = cause_name) %>%
  rename(sex = sex_name) %>%
  rename(age_group = age_name) %>%
  mutate(age_group = gsub(" years", "", age_group)) %>%
  mutate(age_group = gsub("^75\\+?$", "75_plus", age_group))

d.yll <- d.yll %>%

```

```

rename(disease = cause_name) %>%
rename(sex = sex_name) %>%
rename(age_group = age_name)%>%
mutate(age_group = gsub(" years", "", age_group)) %>%
mutate(age_group = gsub("^75\\+\\$", "75_plus", age_group))

# add YLD to d.global
d.global <- d.global %>%
  left_join(d.yld, by = c("disease", "sex", "age_group"))

d.global <- d.global %>%
  rename(
    "yld_val" = val,
    "yld_lower" = lower,
    "yld_upper" = upper)

# add YLL
d.global <- d.global %>%
  left_join(d.yll, by = c("disease", "sex", "age_group"))

d.global <- d.global %>%
  rename(
    "yll_val" = val,
    "yll_lower" = lower,
    "yll_upper" = upper)

# add 0 to NA values
d.global <- d.global %>%
  mutate(across(yld_val:yll_lower, ~ as.numeric(.))) %>%
  mutate(across(yld_val:yll_lower, ~ if_else(is.na(.), 0, .)))

```

Visualization: A faceted scatter plot of YLD values with confidence intervals was generated by disease, sex, and age group. The same was repeated using the merged d.global dataset to verify successful integration.

```

ggplot(d.yld, aes(x = disease, y = (as.numeric(val)), color = age_group)) +
  geom_point(position = position_dodge(width = 0.5), shape = 21) +
  geom_errorbar(
    aes(ymin = as.numeric(lower), ymax = as.numeric(upper)),
    width = 0.2,
    position = position_dodge(width = 0.5)) +
  facet_grid(rows = "sex") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1))

```

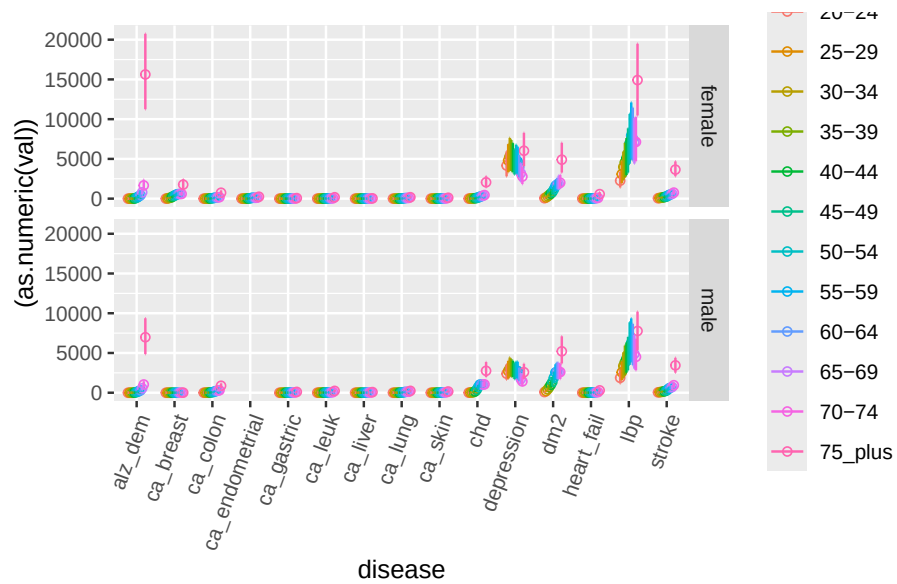


Figure 5: YLD 2022 from GBD

```
ggplot(d.global, aes(x = disease, y = yll_val, color = age_group)) +
  geom_point(position = position_dodge(width = 0.5), shape = 21) +
  geom_errorbar(
    aes(ymin = yll_lower, ymax = yll_upper),
    width = 0.2,
    position = position_dodge(width = 0.5)) +
  facet_grid(rows = "sex") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1))
```

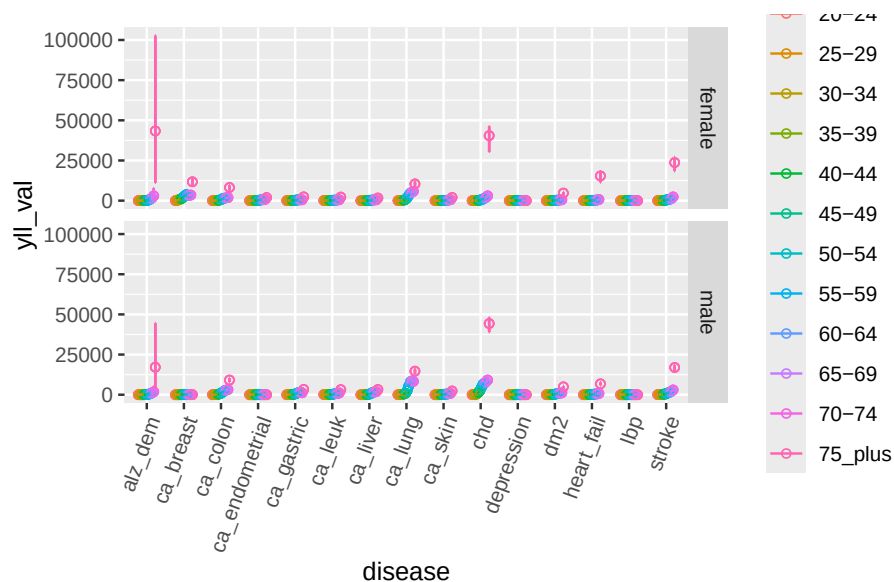


Figure 6: YLL 2022 after integration in d.global

Data Consistency Check: A comparison was made between the YLL values in the original d.yll dataset and the values now in d.global to ensure they matched correctly.

This step ensures that disease burden estimates are fully integrated into the global analysis dataset, enabling comprehensive burden assessments by disease, sex, age group, and region.

```
d.yld %>%
  filter(sex == "male",
         age_group == "30-34",
         disease == "ca_colon")

d.global %>%
  filter(sex == "male",
         age_group == "30-34",
         disease == "ca_colon") %>%
  select(yld_val)
```

Population Data: Preparation (2021-2023)

The Swiss population dataset for the years 2021-2023 was cleaned and formatted to align with the structure of the analytical datasets:

Header Adjustment: The first row of the dataset was used to set the column names, and then removed from the data.

Sex Variable Standardization: Values in the Geschlecht (sex) column were translated from German to English: "Frau" → "female", "Mann" → "male".

Language Region Assignment: Each canton was assigned to a language region (german, french, italian) based on the majority spoken language. This mapping enables stratified demographic analysis by region.

Age Group Classification: Individuals were grouped into predefined 5-year age intervals based on their reported age (Alter). Special categories such as "Alter - Total" and "100 Jahre und mehr" were handled accordingly.

This processing step ensures that the population data is ready to be merged with health-related datasets.

```
# Header Adjustment
colnames(d.population2021_2023) <- d.population2021_2023[1, ]
d.population2021_2023 <- d.population2021_2023[-1, ]

# Sex Variable Standardization
d.population2021_2023 <- d.population2021_2023 %>%
  mutate(Geschlecht = gsub("Frau", "female", Geschlecht)) %>%
  mutate(Geschlecht = gsub("Mann", "male", Geschlecht)) %>%

# Language Region Assignment
# Assignment is based on the majority language in the respective canton
mutate(
  language = case_when(
    Kanton %in% c("Schwyz", "Basel-Landschaft", "Aargau",
                  "Bern / Berne", "Solothurn", "Basel-Stadt",
                  "Zürich", "Luzern", "St.Gallen", "Zug", "Thurgau",
                  "Nidwalden", "Graubünden / Grigioni / Grischun",
                  "Obwalden", "Appenzell Ausserrhoden", "Uri",
                  "Schaffhausen", "Appenzell Innerrhoden", "Glarus")
    ~ "german",

    Kanton %in% c("Valais / Wallis", "Fribourg / Freiburg",
                  "Neuchâtel", "Vaud", "Genève", "Jura")
    ~ "french",

    Kanton == "Ticino" ~ "italian",

    TRUE ~ "others"))
```

```

# Age Group Classification
d.population2021_2023 <- d.population2021_2023 %>%
  mutate(age_num = suppressWarnings(parse_number(Alter)),
         age_group = case_when(
           Alter == "Alter - Total" ~ "all_ages",
           Alter == "100 Jahre und mehr" ~ "75_plus",
           !is.na(age_num) & age_num >= 75 ~ "75_plus",
           !is.na(age_num) & age_num >= 70 ~ "70-74",
           !is.na(age_num) & age_num >= 65 ~ "65-69",
           !is.na(age_num) & age_num >= 60 ~ "60-64",
           !is.na(age_num) & age_num >= 55 ~ "55-59",
           !is.na(age_num) & age_num >= 50 ~ "50-54",
           !is.na(age_num) & age_num >= 45 ~ "45-49",
           !is.na(age_num) & age_num >= 40 ~ "40-44",
           !is.na(age_num) & age_num >= 35 ~ "35-39",
           !is.na(age_num) & age_num >= 30 ~ "30-34",
           !is.na(age_num) & age_num >= 25 ~ "25-29",
           !is.na(age_num) & age_num >= 20 ~ "20-24",
           TRUE ~ NA))

# Renaming Columns
d.population2021_2023 <- d.population2021_2023 %>%
  rename(sex = Geschlecht)

```

National Population: The total population of Switzerland was filtered from the dataset (`Kanton == "Schweiz"`, `age_group == "all_ages"`), and the sum of the population in 2022 was computed.

```

# National Population
d.population_ch <- d.population2021_2023 %>%
  filter(
    Alter == "Alter - Total",
    age_group == "all_ages",
    Kanton == "Schweiz") %>%
  summarise(
    pop_2022_total = sum(`2022`, na.rm = TRUE)
  )

d.population_ch

```

```

  pop_2022_total
1      8815385

```

Disaggregated Population: Population counts were grouped and summed by age group and sex. Subgroups representing total Switzer-

land (i.e., “others” in language or “all_ages” in age group) were excluded from the analysis. For each subgroup, the sum was calculated. Additionally, the relative population share in 2022 (pop_rel) was computed by dividing each subgroup’s population by the total 2022 population for later analysis.

```
#pop_factor (whole CH) - age - sex
# Disaggregated Population Factors
d.population_disagg <- d.population2021_2023 %>%
  filter(Kanton == "Schweiz",
         age_group != "all_ages") %>%
  group_by(age_group, sex) %>%
  summarize(
    across(`2022`, sum, na.rm = TRUE), .groups = "drop")

d.population_disagg <- d.population_disagg %>%
  rename(pop_age_sex = `2022`)
```

```
# pop_region - sex - age - region
d.population_region <- d.population2021_2023 %>%
  #Summarizing number by sex, age group and region
  group_by(age_group, sex, language) %>%
  summarize(
    across(`2022`, sum, na.rm = TRUE), .groups = "drop")

d.population_region <- d.population_region %>%
  rename(
    pop_region = `2022`)
```

Population data was inserted in the global dataframe.

```
# Merging Population Factors (pop factor sex/age)
d.global <- d.global %>%
  left_join(d.population_disagg,
           by = c("age_group", "sex"))

# Merging Population Factors (pop factor sex/age/langage)
d.global <- d.global %>%
  left_join(d.population_region, by = c("age_group", "sex",
                                       "language")) %>%
  mutate(pop_rel = pop_region/pop_age_sex)
```

Validation of population data

```

d.global %>%
  filter(
    age_group == "75_plus",
    disease == "ca_colon",
    sex == "female") %>%
  select(disease:age_group, pop_age_sex, pop_region)

d.population2021_2023 %>%
  filter(
    age_group == "75_plus",
    sex == "female",
    language == "german") %>%
  summarise(sum = sum(`2022`))

d.global %>%
  filter( age_group == "75_plus",
    disease == "ca_colon",
    sex == "female") %>%
  select(contains("pop")) %>%
  summarise(pop_rel_sum = sum(pop_rel))

d.global %>%
  filter(disease == "ca_lung") %>%
  select(disease:age_group, contains("pop")) %>%
  summarise(sum = sum(pop_region))

d.population2021_2023 %>%
  filter(!grepl("Total", Alter),
    !grepl("Schweiz", Kanton)) %>%
  summarise(total_ch = sum(`2022`, na.rm = TRUE))

```

Validation

```

d.global %>%
  filter(disease== "alz_dem",
    age_group == "40-44",
    sex == "male",
    language == "italian") %>%
  summarise(sum = sum(yll_val))

d.ihme_gbd_data %>%
  filter(measure_id == "4",
    sex_name == "male",
    cause_name == "alz_dem",
    age_name == "40-44 years",

```

```
metric_name == "Number") %>%
select(val)
```

Mortality Data Preparation

Mortality data from the Swiss Federal Statistical Office (FSO) for 2022 was imported, cleaned, and structured for further analysis:

Import and Initialization: The raw Excel file was imported and converted into a data frame (d.deaths).

Standardization of Variables: Sex labels were harmonized ("F" → "female", "M" → "male"). Language regions were assigned to each canton based on the majority spoken language, using consistent regional groupings (German, French, Italian).

```
# Standardization of Variables
d.deaths <- d.deaths %>%
  mutate(sex = gsub("F", "female", sex)) %>%
  mutate(sex = gsub("M", "male", sex)) %>%

# Assignment of regions based on the majority language in each canton
mutate(
  language = case_when(
    kanton %in% c("Schwyz", "Basel-Landschaft", "Aargau", "Bern",
                  "Solothurn", "Basel-Stadt", "Zürich", "Luzern",
                  "St.Gallen", "Zug", "Thurgau", "Nidwalden",
                  "Graubünden", "Obwalden", "Appenzell A.Rh.",
                  "Uri", "Schaffhausen", "Appenzell I.Rh.", "Glarus")
    ~ "german",

    kanton %in% c("Valais", "Fribourg", "Neuchâtel", "Vaud",
                  "Genève", "Jura") ~ "french",

    kanton == "Ticino" ~ "italian", TRUE ~ "others"))
```

Age Group Classification: Individuals were grouped into predefined age bands. Those under 20 were assigned to a placeholder group ("<20").

```
# Age Group Classification
d.deaths <- d.deaths %>%
  mutate(age_group = case_when(
    age >= 20 & age <= 24 ~ "20-24",
    age >= 25 & age <= 29 ~ "25-29",
```

```

age >= 30 & age <= 34 ~ "30-34",
age >= 35 & age <= 39 ~ "35-39",
age >= 40 & age <= 44 ~ "40-44",
age >= 45 & age <= 49 ~ "45-49",
age >= 50 & age <= 54 ~ "50-54",
age >= 55 & age <= 59 ~ "55-59",
age >= 60 & age <= 64 ~ "60-64",
age >= 65 & age <= 69 ~ "65-69",
age >= 70 & age <= 74 ~ "70-74",
age >= 75 ~ "75_plus",
TRUE ~ "<20"))

```

Disease Mapping via ICD Codes: Cause of death codes (ICD-10) were used to assign each record to a specific disease category relevant to the study (e.g., “C34” → “ca_lung” for lung cancer). A `case_when()` statement mapped known ICD code patterns to internal disease identifiers; unmatched codes were labeled “others”.

This preprocessing ensures that death records are consistently categorised by region, disease, sex and age group, enabling reliable integration with other datasets such as burden of disease estimates and population statistics.

```

# Disease Mapping via ICD Codes
d.deaths <- d.deaths %>%
  mutate(disease = case_when(
    str_detect(ICD, "F00|F01|F020|F028|F029|F03|G30|G310|G311|
      G318|G319") ~ "alz_dem",
    str_detect(ICD, "C50[0-9]|D05[0-9]|D24[0-9]|D486|D493")
      ~ "ca_breast",
    str_detect(ICD, "C18|C19|C20|C21[0-9]|D01[0-3]|D12[0-9]|
      D37[3-5]") ~ "ca_colon",
    str_detect(ICD, "I2[0-5][0-9]?") ~ "chd",
    str_detect(ICD, "F3[2-3][0-9]?|F341") ~ "depression",
    str_detect(ICD, "E10[0-1]|E10[3-9]|E11[0-1]|E11[3-9]|
      P702") ~ "dm2",
    str_detect(ICD, "C54[0-9]|D07[0-2]|D26[1-9]")
      ~ "ca_endometrial",
    str_detect(ICD, "C16[0-9]|D002|D131|D371") ~ "ca_gastric",
    str_detect(ICD, "I11[0-2]|I119") ~ "heart_fail",
    str_detect(ICD, "C22[0-8]|D134") ~ "ca_liver",
    str_detect(ICD, "G544|M47[0-8]|M48[0-5]|M49[8]|M51[0-4]|
      M51[8]|M53[3-8]|M54[1]|M54[3-5]|M99[0-8]")
      ~ "lbp",
    str_detect(ICD, "C3[3-4][0-9]|D02[1-3]|D14[2-3]|D381")
      ~ "ca_lung",
  )

```

```

str_detect(ICD, "C91(0|2|3|6)?|C92[0-6]|C93(0|1|3|8)?|
  C94[0-5]|C9[4-5][7-9]?") ~ "ca_leuk",
str_detect(ICD, "C88|C89|C90([0-9])?") ~ "ca_skin",
str_detect(ICD, "G4[5-6][0-8]?|I6[0-3]|I6[5-6]|I67[0-3]|
  I67[5-6]|I68[1-2]|I69[0-3]") ~ "stroke",
TRUE ~ "others"))

```

Filtering and Aggregation of Deaths: The d.deaths dataset was filtered to include only relevant diseases (i.e., excluding “others”) and individuals aged 20 or older (age_group != “<20”). Death counts were then aggregated by region, disease, sex and age group.

Merging with Global Dataset: The aggregated death counts were joined into the d.global dataset using a left join on the same four dimensions. Missing combinations (i.e., no deaths recorded) were filled with zeros to ensure data completeness. In addition, the total number of deaths in 2022 (deaths_2022_total) was computed, and the relative share of each subgroup (deaths_rel) was calculated as the proportion of deaths in that subgroup relative to the overall death count for later analysis.

```

# Filtering and Aggregation of Deaths
d.deaths <- d.deaths %>%
  # include only relevant diseases
  filter(disease != "others",
    age_group != "<20") %>%
  # aggregation of deaths by disease, language region, sex,
  # and age group
  group_by(disease, language, sex, age_group) %>%
  summarise(deaths_region = n()) %>%
  arrange(language, sex, age_group, disease)

# Merging with Global Dataset
d.global <- d.global %>%
  left_join(d.deaths, by = c("disease", "language", "sex",
    "age_group"))

# complete missing combinations when death count is 0
d.global <- d.global %>%
  mutate(deaths_region = ifelse(is.na(deaths_region), 0,
    deaths_region))

# calc n.deaths
d.global <- d.global %>%
  mutate(deaths_2022_total = sum(d.global$deaths_region))

d.deaths.ch <- d.global %>%

```

```

group_by(disease, age_group, sex) %>%
  summarise(deaths_age_sex = sum(deaths_region))

d.global <- d.global %>%
  left_join(d.deaths.ch, by = c("disease", "age_group", "sex"))

d.global <- d.global %>%
  mutate(deaths_rel = deaths_region / deaths_age_sex,
         deaths_rel = ifelse(is.na(deaths_rel), 0, deaths_rel))

```

Validation check of integration

```

d.global %>%
  filter(disease == "ca_breast",
         age_group == "45-49",
         sex == "female") %>%
  select(deaths_region, deaths_age_sex)

d.global %>%
  filter( disease == "chd",
         sex == "male",
         age_group == "50-54") %>%
  select(contains("deaths"))

```

Validation of Death Counts: A second, independent summary (d.deaths.test) was created by reclassifying causes of death from raw ICD codes and grouping the results by disease and sex. This was compared against the corresponding totals from the d.global dataset (d.deaths.test.gl). The combined table (cbind(...)) allows for side-by-side comparison to verify that the integration and summarization of death counts were performed correctly.

This step ensures that actual mortality data is consistently integrated into the disease burden framework and that no relevant subgroup is omitted in the analysis.

```

d.deaths.test <- deaths_canton %>%
  mutate(disease = case_when(
    str_detect(ICD, "F00|F01|F020|F028|F029|F03|G30|G310|G311|
                  G318|G319") ~ "alz_dem",
    str_detect(ICD, "C50[0-9]|D05[0-9]|D24[0-9]|D486|D493")
    ~ "ca_breast",
    str_detect(ICD, "C18|C19|C20|C21[0-9]|D01[0-3]|D12[0-9]|
                  D37[3-5]") ~ "ca_colon",
    str_detect(ICD, "I2[0-5][0-9]?") ~ "chd",

```

```

str_detect(ICD, "F3[2-3][0-9]?|F341") ~ "depression",
str_detect(ICD, "E10[0-1]|E10[3-9]|E11[0-1]|E11[3-9]|P702")
~ "dm2",
str_detect(ICD, "C54[0-9]|D07[0-2]|D26[1-9]")
~ "ca_endometrial",
str_detect(ICD, "C16[0-9]|D002|D131|D371") ~ "ca_gastric",
str_detect(ICD, "I11[0-2]|I119") ~ "heart_fail",
str_detect(ICD, "C22[0-8]|D134") ~ "ca_liver",
str_detect(ICD, "G544|M47[0-8]|M48[0-5]|M49[8]|M51[0-4]|
M51[8]|M53[3-8]|M54[1]|M54[3-5]|M99[0-8]")
~ "lbp",
str_detect(ICD, "C3[3-4][0-9]|D02[1-3]|D14[2-3]|D381")
~ "ca_lung",
str_detect(ICD, "C91(0|2|3|6)?|C92[0-6]|C93(0|1|3|8)?|
C94[0-5]|C9[4-5][7-9]?") ~ "ca_leuk",
str_detect(ICD, "C88|C89|C90([0-9])?") ~ "ca_skin",
str_detect(ICD, "G4[5-6][0-8]?|I6[0-3]|I6[5-6]|I67[0-3]|
I67[5-6]|I68[1-2]|I69[0-3]") ~ "stroke",
TRUE ~ "others"))) %>%
filter(age>19, disease != "others") %>%
group_by(disease, sex) %>%
summarise(sum = n())

d.deaths.test.gl <- d.global %>%
group_by(disease, sex) %>%
filter(deaths_region >0) %>%
summarise(sum = sum(deaths_region))

cbind(d.deaths.test, d.deaths.test.gl)

```

Weighted Distribution of burden of disease / Proxy by region, sex and age

This section estimates the regional distribution of disease burden (DALYs) using a proxy weighting approach that combines population size and death counts.

Weight Construction: A composite weight (weights) was calculated per disease, sex, and age group:

- 80% based on the relative population size (pop_rel)
- 20% based on the relative number of deaths (deaths_rel)

This weighting scheme reflects the assumption that DALYs are primarily driven by population size but also influenced by mortality patterns.

Weighting and Normalization Across Regions: To allocate disease burden more realistically across Switzerland's regions, a weighted factor (w.norm) was constructed for each disease–sex–age group combination. The weight is based on a combination of population share (pop_rel) and mortality share (deaths_rel), with default weights of 80% and 20%. This approach allows for proportional regional allocation even in the absence of direct regional burden data.

```
# w <- (0.8 * s) + (0.2 * death_prop)
# 90% of the distribution of DALYs is due to population size,
# 10% from deaths.

# Adjustment w.norm
# This version applies normalization across the entire dataset.
# Normalize w.norm within each language region separately
d.weights <- d.global %>%
  mutate(weights = 0.8 * pop_rel + 0.2 * deaths_rel) %>%
  group_by(age_group, sex, disease) %>%
  mutate(w_sum = sum(weights)) %>%
  select(disease, age_group, language, sex, weights, w_sum)

d.global <- d.global %>%
  left_join(d.weights, by = c("age_group", "sex", "language",
                             "disease"))

d.global <- d.global %>%
  mutate(w.norm = weights/w_sum)

d.global %>%
  filter(
    sex == "male",
    age_group == "75_plus") %>%
  select(age_group, sex, disease, contains("pop"), weights,
         deaths_rel, w.norm) %>%
  summarise(w_sum = sum(w.norm))
```

```
      w_sum
1      15
```

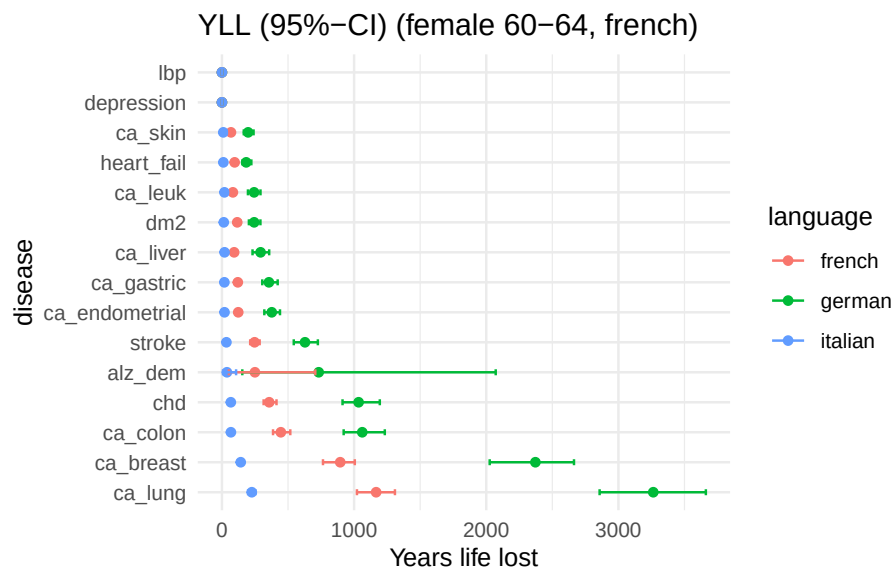
Weighted DALY Calculation: Normalized weights (w.norm) were applied to scale YLD and YLL values and their confidence intervals (_lower, _upper). This produced regionally distributed estimates and their respective lower and upper bounds.

```
# Weighted DALY Calculation
d.global <- d.global %>%
  mutate(
    #Apply on DALYs and CI
    yll_region_val = (w.norm * yll_val),
    yll_region_lower = (w.norm * yll_lower),
    yll_region_upper = (w.norm * yll_upper),

    yld_region_val = (w.norm * yld_val),
    yld_region_lower = (w.norm * yld_lower),
    yld_region_upper = (w.norm * yld_upper))
```

Plausibility and Visualization: A diagnostic plot was created to visualize weighted YLL by disease for French-speaking females aged 60-64. This allows visual inspection of whether the distribution across diseases aligns with expectations. Absolute values are displayed.

```
d.global %>%
  filter(sex == "female", age_group == "60-64") %>%
  ggplot(aes(x = reorder(disease, -yll_region_val),
              y = yll_region_val, color = language)) +
  geom_point() +
  geom_errorbar(aes(ymin = yll_region_lower,
                    ymax = yll_region_upper), width = 0.2) +
  scale_y_continuous() +
  labs(
    title = "YLL (95%-CI) (female 60-64, french)",
    x = "disease", y = "Years life lost"
  ) +
  theme_minimal() +
  coord_flip()
```



Validation of Normalized Weights: As a quality check, we verify that the normalized weights (w.norm) sum to 1 within each disease-sex-age group combination.

These checks ensure that the weighting procedure is correctly applied and that all groups are complete and usable in downstream analyses.

```
d.global %>%
  group_by(disease, sex, age_group) %>%
  summarise(w_sum = sum(w.norm))

# Check for NAs in weights:
sum(is.na(d.global$w.norm))
```

Validation of Weighted YLD Estimates: For validation, we used the subgroup “Diabetes mellitus type 2”, women aged 50-54. Normalized weights (w.norm), based on relative population and mortality shares, were applied to national YLD estimates (yld_2022_ch_val) to compute weighted values (yld_2022_region_val). Summing these across regions gives the total weighted YLD, compared against the unweighted average.

```
d.global %>%
  filter(disease == "dm2", sex == "female", age_group == "50-54") %>%
  select(pop_rel, deaths_rel, weights, w.norm, yld_val,
         yld_region_val) %>%
  summarise(sum_test = sum(yld_region_val))
```

Attributable Burden Estimates (YLD & YLL)

The attributable burden was calculated by multiplying the PAF with weighted YLD and YLL values. Specifically, `paf * yld_region_val` and `paf * yll_region_val` yielded `attr_yld_point_est` and `attr_yll_point_est`.

```
# Attributable Burden Estimates (YLD & YLL)
d.global <- d.global %>%
  mutate(attr_yld_point_est = paf * yld_region_val,
         attr_yll_point_est = paf * yll_region_val) %>%
  arrange(disease, language, age_group, sex)
```

Parametric Bootstrap for Uncertainty Estimation

Implementing Parametric bootstrapp: To calculate uncertainty in prevalence and relative risk estimates into the PIA-attributable burden calculations, a parametric bootstrap approach was implemented.

A function (`estBetaParams`) was defined to estimate the shape parameters (alpha, beta) of a Beta distribution from given mean and standard deviation values. This allows modeling uncertainty in prevalence values.

The bootstrap replicates were specified. For each replicate, three output matrices were initialized to store simulated results for:

- `d.paf.boot`: Population Attributable Fractions (PAFs)
- `d.attr.yll.boot`: Attributable Years of Life Lost (YLL)
- `d.attr.yld.boot`: Attributable Years Lived with Disability (YLD)

Each matrix has `nrow = R` and `ncol = n`, where `n` is the number of disease-region-age-sex combinations in the global dataset. The column names are dynamically created from these subgroup identifiers to track each combination across simulations.

A point was chosen as the separator to simplify subsequent analyses, since underscores (`_`) are already used in disease names and hyphens (`-`) in age labels. This setup forms the foundation for subsequent simulation steps, where random draws will be made from fitted distributions to generate a distribution of plausible PIA-attributable burden values.

```
# create function to est params for beta distr.
estBetaParams <- function(mu, sd) {
```

```

var <- sd^2
alpha <- ((1 - mu) / var - 1 / mu) * mu ^ 2
beta <- alpha * (1 / mu - 1)
return(params = list(alpha = alpha, beta = beta))}

# define number of iterations
R <- 5000

# create empty matrices
d.paf.boot <- matrix(NA, nrow = R, ncol = nrow(d.global))
d.attr.yld.boot <- matrix(NA, nrow = R, ncol = nrow(d.global))
d.attr.yll.boot <- matrix(NA, nrow = R, ncol = nrow(d.global))

```

Parametric Bootstrapping

To quantify uncertainty in PIA-attributable burden estimates, a parametric bootstrap approach was applied across all disease-age-sex-region combinations in the d.global dataset. In each iteration, values for relative risk (log-normal), prevalence (beta), and YLD/YLL (log-normal) were randomly drawn based on their estimated uncertainty.

The simulated PAF was applied to the sampled YLD and YLL to compute PIA-attributable values. Results were stored in dedicated matrices for PAF, YLD, and YLL, forming the basis for deriving confidence intervals.

```

# define set.seed to ensure reproducibility
set.seed(123)

for (i in 1:nrow(d.global)) {

  for (j in 1:R) {

# sample rr from log-normal distr.
log.rr <- log(d.global$riskratio[i])
se.log.rr <- (log(d.global$rr_upper[i]) -
              log(d.global$riskratio[i]))/1.96
rr.temp <- exp(rnorm(1, log.rr, se.log.rr))

# sample prev from beta distr.
prev <- d.global$prev[i]
se.prev <- (d.global$prev_upper[i] -
            d.global$prev[i])/1.96
alpha.beta <- estBetaParams(prev, se.prev)
prev.temp <- rbeta(1, alpha.beta$alpha,

```

```

alpha.beta$beta)

# calculate PAF with Levins formula
paf.temp <- PAF2(prev.temp, rr.temp)

# sample YLD from log-normal distr.
yld <- log(d.global$yld_region_val[i])
se.yld <- (log(d.global$yld_region_upper[i]) -
           log(d.global$yld_region_val[i]))/1.96
yld.temp <- exp(rnorm(1, yld, se.yld))

# calculate PIA attr. YLD
yld.attr.temp <- paf.temp * yld.temp
yld.attr.temp <- ifelse(is.na(yld.attr.temp),
                       0, yld.attr.temp)

# sample YLL from log-normal distr.
yll <- log(d.global$yll_region_val[i])
se.yll <- (log(d.global$yll_region_upper[i]) -
           log(d.global$yll_region_val[i]))/1.96
yll.temp <- exp(rnorm(1, yll, se.yll))

# calculate PIA attr. YLD
yll.attr.temp <- paf.temp * yll.temp
yll.attr.temp <- ifelse(is.na(yll.attr.temp),
                       0, yll.attr.temp)

# save calculated values in data.frame
d.paf.boot[j, i] <- paf.temp
d.attr.yld.boot[j, i] <- yld.attr.temp
d.attr.yll.boot[j, i] <- yll.attr.temp
}}

# store results in dataframes
#PAF
d.paf.boot <- as.data.frame(d.paf.boot)
names(d.paf.boot) <- paste(d.global$disease,
                           d.global$language,d.global$age,
                           d.global$sex, sep = ".")

#YLL
d.attr.yll.boot <- as.data.frame(d.attr.yll.boot)
names(d.attr.yll.boot) <- paste(d.global$disease,
                                d.global$language,d.global$age,
                                d.global$sex, sep = ".")

```

```
#YLD
d.attr.yld.boot <- as.data.frame(d.attr.yld.boot)
names(d.attr.yld.boot) <- paste(d.global$disease,
                                d.global$language,d.global$age,
                                d.global$sex, sep = ".")
```

Bootstrap Confidence Intervals for PAFs

To quantify uncertainty in PAF estimates, 95% confidence intervals were derived from the bootstrap results. For each disease-age-sex-language subgroup, the mean, 2.5th percentile (lower bound), and 97.5th percentile (upper bound) of the simulated PAFs were calculated.

These summary statistics were combined into a new dataset and merged back into d.global.

```
# get CI for each region-disease-sex-age subgroup
#PAF
d.paf.ci <- t(sapply(d.paf.boot, function(col) {
  c(mean = mean(col),
    lower = quantile(col, probs = 0.025),
    upper = quantile(col, probs = 0.975))}))

d.paf.ci <- as.data.frame(d.paf.ci)

# add to d.global
d.global <- d.global %>%
  mutate(paf_boot_mean = d.paf.ci$mean,
         paf_boot_lower = d.paf.ci$`lower.2.5%`,
         paf_boot_upper = d.paf.ci$`upper.97.5%`,
         group = rownames(d.paf.ci))
```

A random sample is drawn from d.global, selecting key demographic variables and PAF estimates for a quick exploratory check. The previous point estimates should be close to the values of the bootstrap results.

```
# check
d.global %>%
  select(disease, language, sex, age_group, group,
         paf, paf_boot_mean,
         paf_boot_lower,
         paf_boot_upper) %>%
  slice_sample(n = 20)
```

```

#YLD
# get CI for each disease-sex-age-region group
d.yld.ci <- t(sapply(d.attr.yld.boot, function(col) {
  c(mean = mean(col),
    lower = quantile(col, probs = 0.025),
    upper = quantile(col, probs = 0.975))}))

d.yld.ci <- as.data.frame(d.yld.ci)

# add to d.global
d.global <- d.global %>%
  mutate(yld_boot_mean = d.yld.ci$mean,
    yld_boot_lower = d.yld.ci$`lower.2.5%`,
    yld_boot_upper = d.yld.ci$`upper.97.5%`,
    group = rownames(d.yld.ci))

#YLL
# get CI for each disease-sex-age-region group
d.yll.ci <- t(sapply(d.attr.yll.boot, function(col) {
  c(mean = mean(col),
    lower = quantile(col, probs = 0.025),
    upper = quantile(col, probs = 0.975))}))

d.yll.ci <- as.data.frame(d.yll.ci)

# add to d.global
d.global <- d.global %>%
  mutate(yll_boot_mean = d.yll.ci$mean,
    yll_boot_lower = d.yll.ci$`lower.2.5%`,
    yll_boot_upper = d.yll.ci$`upper.97.5%`,
    group = rownames(d.yll.ci))

# check
d.global %>%
  select(disease, language, sex, age_group, group,
    attr_yld_point_est, yld_boot_mean,
    attr_yll_point_est, yll_boot_mean
  ) %>%
  slice_sample(n = 20)

```

Consistency Check of Input and Derived YLD Variables

To verify the consistency of PIA-attributable YLD calculations, a subset of d.global was filtered for women aged 55–59 with Alzheimer’s and dementia. The output compares the sum of regionally weighted YLD

values (yld_2022_region_val) across language regions to the national total (yld_val).

The sum of the regionally weighted YLD values (yld_regions) across the German-, French-, and Italian-speaking regions should approximately equal the national YLD value (yld_val).

$yld_w(ger, fr, it) \approx yld_val$

$$\sum_{region \in \{ger, fr, it\}} yld_w^{region} \approx yld_{2022}^{val}$$

```
d.global %>%
  filter(sex == "female",
         age_group == "55-59",
         disease == "alz_dem") %>%
  select(sex, language, yld_val, yld_region_val, w.norm,
         weights) %>%
  summarise(sum = sum(yld_region_val))
```

Total YLD + YLL Values for Switzerland Aggregation

Total Estimates for Switzerland

To estimate the total burden of disease attributable to PIA in Switzerland bootstrap-simulated YLD + YLL values across all combinations of region, disease, sex and age group were aggregated.

```
# Sum for YLD of all all disease-region-age-sex groups
attr.yld.total.ch <- rowSums(d.attr.yld.boot)

# Sum for YLL of all disease-region-age-sex groups
attr.yll.total.ch <- rowSums(d.attr.yll.boot, na.rm = TRUE)

# Sum all vectors for YLL+YLD of all disease-region-age-sex groups
attr.total.ch <- attr.yld.total.ch + attr.yll.total.ch
```

Totals CH

```
# mean and 95%-CI YLD
mean.yld.total.ch <- mean(attr.yld.total.ch, na.rm = TRUE)
lower.yld.total.ch <- quantile(attr.yld.total.ch, probs = 0.025,
```

```

                                na.rm = TRUE)
upper.yld.total.ch <- quantile(attr.yld.total.ch, probs = 0.975,
                                na.rm = TRUE)

# mean and 95%-CI YLL
mean.yll.total.ch <- mean(attr.yll.total.ch, na.rm = TRUE)
lower.yll.total.ch <- quantile(attr.yll.total.ch, probs = 0.025,
                                na.rm = TRUE)
upper.yll.total.ch <- quantile(attr.yll.total.ch, probs = 0.975,
                                na.rm = TRUE)

```

PIA-Attributable Rates for Switzerland

To express the health burden in relative terms, YLD and YLL rates per 100 000 population were calculated using the total Swiss population in 2022.

Note: Population data included only individuals > 20 years.

```

# sum population > 20
pop_ch_total <- sum(d.population_disagg$pop_age_sex)

```

Rates YLD CH

```

# Rates per Bootstrap for YLD
yld.rate.total.ch_boot <- (attr.yld.total.ch /
                            pop_ch_total) * 100000

# mean and 95%-CI (YLD-Rate):
mean.yld.rate.total.ch <- mean(yld.rate.total.ch_boot,
                                na.rm = TRUE)
lower.yld.rate.total.ch <- quantile(yld.rate.total.ch_boot,
                                    probs = 0.025, na.rm = TRUE)
upper.yld.rate.total.ch <- quantile(yld.rate.total.ch_boot,
                                    probs = 0.975, na.rm = TRUE)

```

Rates YLL CH

```

yll.rate.total.ch_boot <- (attr.yll.total.ch /
                           pop_ch_total) * 100000

# mean and 95%-CI (YLL)
mean.yll.rate.total.ch <- mean(yll.rate.total.ch_boot,
                               na.rm = TRUE)
lower.yll.rate.total.ch <- quantile(yll.rate.total.ch_boot,
                                    probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.ch <- quantile(yll.rate.total.ch_boot,
                                    probs = 0.975, na.rm = TRUE)

```

Rate CH Total Burden (YLD + YLL)

```

total.rate.total.ch.boot <- ((attr.yld.total.ch + attr.yll.total.ch) /
                             pop_ch_total) * 100000

# mean and 95%-CI (Total-Rate)
mean.total.rate.total.ch <- mean(total.rate.total.ch.boot,
                                 na.rm = TRUE)
lower.total.rate.total.ch <- quantile(total.rate.total.ch.boot,
                                    probs = 0.025, na.rm = TRUE)
upper.total.rate.total.ch <- quantile(total.rate.total.ch.boot,
                                    probs = 0.975, na.rm = TRUE)

```

A summary data frame was created to compile the national PIA-attributable rates for YLD, YLL, and total burden (YLD + YLL) per 100 000.

```

#Summary of national burden of disease
d.rate.total.ch <- data.frame(
  measure = c("YLD_rate", "YLL_rate", "Total_rate"),
  mean     = c(mean.yld.rate.total.ch,
               mean.yll.rate.total.ch, mean.total.rate.total.ch),
  lower    = c(lower.yld.rate.total.ch, lower.yll.rate.total.ch,
               lower.total.rate.total.ch),
  upper    = c(upper.yld.rate.total.ch, upper.yll.rate.total.ch,
               upper.total.rate.total.ch)
)

# Absolute values
d.abs.total.ch <- data.frame(
  measure = c("YLD_abs", "YLL_abs", "Total_abs"),

```

```

mean    = c(mean.yld.total.ch, mean.yll.total.ch,
             mean.yld.total.ch + mean.yll.total.ch),
lower    = c(lower.yld.total.ch, lower.yll.total.ch,
             lower.yld.total.ch + lower.yll.total.ch),
upper    = c(upper.yld.total.ch, upper.yll.total.ch,
             upper.yld.total.ch + upper.yll.total.ch)
)

# merge and format CI
d.total.ch <- bind_rows(d.rate.total.ch, d.abs.total.ch) %>%
  mutate(
    ci = paste0(sprintf("%.2f", lower), ", ",
                 sprintf("%.2f", upper)),
    mean_ci = paste0(sprintf("%.2f", mean), " [",
                     sprintf("%.2f", lower), ", ",
                     sprintf("%.2f", upper), "]")
  )

```

Aggregation by sex

Sex-specific estimates of YLD + YLL attributable to PIA were calculated based on bootstrap iterations. Population totals for males and females > 20 years in 2022 were first extracted. The rates for PIA-attributable YLD + YLL per iteration was then computed separately for each sex by summing columns containing sex-specific data. Mean values and 95% confidence intervals were derived from these distributions.

```

# Population by sex (2022) over 20 years
pop.male <- sum(d.population_disagg$pop_age_sex
               [d.population_disagg$sex == "male"])
pop.female <- sum(d.population_disagg$pop_age_sex
                  [d.population_disagg$sex == "female"])

```

YLD

```

# Total YLD per bootstrap iteration for each sex
#\.\. tells R: "this is a real dot, not a wildcard"

# male
attr.yld.total.male <- rowSums(
  d.attr.yld.boot[, str_detect(colnames(d.attr.yld.boot),

```

```

                                "\\male")],
  na.rm = TRUE)

# female
attr.yld.total.female <- rowSums(d.attr.yld.boot
                                [, which(grepl("female",
                                                colnames(d.attr.yld.boot)))],
                                na.rm = TRUE)

```

PIA-attributable YLD rates were calculated separately for males and females, expressed per 100 000 population.

```

# YLD rates per 100 000 by sex (bootstrap)
yld.rate.total.male.boot <-
  (attr.yld.total.male / pop.male) * 100000
yld.rate.total.female.boot <-
  (attr.yld.total.female / pop.female) * 100000

# Rates summary statistics (mean and 95% CI)
# female
mean.yld.rate.total.female <- mean(yld.rate.total.female.boot,
                                   na.rm = TRUE)
lower.yld.rate.total.female <- quantile(yld.rate.total.female.boot,
                                       probs = 0.025, na.rm = TRUE)
upper.yld.rate.total.female <- quantile(yld.rate.total.female.boot,
                                       probs = 0.975, na.rm = TRUE)

#male
mean.yld.rate.total.male <- mean(yld.rate.total.male.boot,
                                 na.rm = TRUE)
lower.yld.rate.total.male <- quantile(yld.rate.total.male.boot,
                                       probs = 0.025, na.rm = TRUE)
upper.yld.rate.total.male <- quantile(yld.rate.total.male.boot,
                                       probs = 0.975, na.rm = TRUE)

d.rate.yld.sex <- data.frame(
  sex = c("male", "female"),
  mean = c(mean.yld.rate.total.male,
            mean.yld.rate.total.female),
  lower = c(lower.yld.rate.total.male,
             lower.yld.rate.total.female),
  upper = c(upper.yld.rate.total.male,
            upper.yld.rate.total.female)) %>%
  mutate(
    ci = paste0(sprintf("%.2f", lower), ", ",

```

Table 1: PIA attributable YLD Rate per 100 000 per sex (2022)

Sex	Rate per 100 000 (95%-CI)
male	166.31 [155.06, 178.43]
female	286.05 [264.57, 308.89]

```

        sprintf("%.2f", upper)),
mean_ci = paste0(sprintf("%.2f", mean), " [",
                  sprintf("%.2f", lower), ", ",
                  sprintf("%.2f", upper), "]")

```

```

d.rate.yld.sex %>%
  select(sex, mean_ci) %>%
  kbl(
    digits = 2,
    col.names = c("Sex", "Rate per 100 000 (95%-CI)")
  ) %>%
  kable_styling(full_width = FALSE, position = "center")

```

YLL

Attributable YLL Rates by Sex (per 100 000 population) were estimated separately for males and females using bootstrap-derived totals. The total YLL per iteration was calculated by summing sex-specific columns, followed by computation of the mean and 95% confidence intervals. These totals were then scaled by the respective sex-specific populations to obtain YLL rates per 100 000.

```

# Total YLL per bootstrap iteration for each sex
# male
attr.yll.total.male <- rowSums(
  d.attr.yll.boot[, str_detect(colnames(d.attr.yll.boot),
                              "\\..male$"), na.rm = TRUE)

# female
attr.yll.total.female <-
  rowSums(d.attr.yll.boot[, which(grepl("female",
                                       colnames(d.attr.yll.boot)))], na.rm = TRUE)

# Rates
# YLL rates per 100 000 by sex
yll.rate.total.male.boot <- (attr.yll.total.male /

```

```

      pop.male) * 100000
yll.rate.total.female.boot <- (attr.yll.total.female /
      pop.female) * 100000

# Summary statistics
mean.yll.rate.total.female <- mean(yll.rate.total.female.boot,
      na.rm = TRUE)
lower.yll.rate.total.female <- quantile(yll.rate.total.female.boot,
      probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.female <- quantile(yll.rate.total.female.boot,
      probs = 0.975, na.rm = TRUE)

mean.yll.rate.total.male <- mean(yll.rate.total.male.boot,
      na.rm = TRUE)
lower.yll.rate.total.male <- quantile(yll.rate.total.male.boot,
      probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.male <- quantile(yll.rate.total.male.boot,
      probs = 0.975, na.rm = TRUE)

d.rate.yll.sex <- data.frame(
  sex = c("male", "female"),
  mean = c(mean.yll.rate.total.male,
    mean.yll.rate.total.female),
  lower = c(lower.yll.rate.total.male,
    lower.yll.rate.total.female),
  upper = c(upper.yll.rate.total.male,
    upper.yll.rate.total.female)) %>%
  mutate(
    ci = paste0(sprintf("%.2f", lower), ", ",
      sprintf("%.2f", upper)),
    mean_ci = paste0(sprintf("%.2f", mean), " [",
      sprintf("%.2f", lower), ", ",
      sprintf("%.2f", upper), "]"))

d.rate.yll.sex %>%
  select(sex, mean_ci) %>%
  kbl(
    digits = 2,
    col.names = c("Sex", "Mean (95%-CI)") %>%
    kable_styling(full_width = FALSE, position = "center")

```

Table 2: PIA attributable YLL Rate per 100 000 per sex (2022)

Sex	Mean (95%-CI)
male	371.67 [337.51, 414.86]
female	475.21 [404.82, 577.64]

Table 3: PIA attributable burden of disease per sex (2022)

Measure	Sex	Mean (95%-CI)
YLL	male	371.67 [337.51, 414.86]
YLL	female	475.21 [404.82, 577.64]
YLD	male	166.31 [155.06, 178.43]
YLD	female	286.05 [264.57, 308.89]

YLD and YLL Rate Data by Sex

To summarize the total burden of disease attributable to PIA by sex, YLD and YLL rate estimates were merged into a dataset (d.yld.yll.sex).

```
d.yld.yll.sex <- bind_rows(
  d.rate.yll.sex %>% mutate(measure = "yll"),
  d.rate.yld.sex %>% mutate(measure = "yld"))
```

```
d.yld.yll.sex %>%
  mutate(
    measure = case_when(
      measure == "yll" ~ "YLL",
      measure == "yld" ~ "YLD",
      TRUE ~ measure)) %>%
  select(measure, sex, mean_ci) %>%
  kbl(
    digits = 2,
    col.names = c("Measure", "Sex", "Mean (95%-CI)")) %>%
  kable_styling(full_width = FALSE, position = "center")
```

```
d.yld.yll.sex %>%
  mutate(measure = case_when(
    measure == "yld" ~ "YLD",
    measure == "yll" ~ "YLL",
    TRUE ~ measure)) %>%

  ggplot(aes(x = measure,
             y = mean,
```

```

    color = sex)) +
  geom_point(position = position_dodge(width = 0.5), size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper),
    position = position_dodge(width = 0.5),
    width = 0.2) +
  scale_y_continuous(name = "Rate per 100 000") +
  labs(
    x = "Measure",
    color = "") +
  theme_minimal()

```

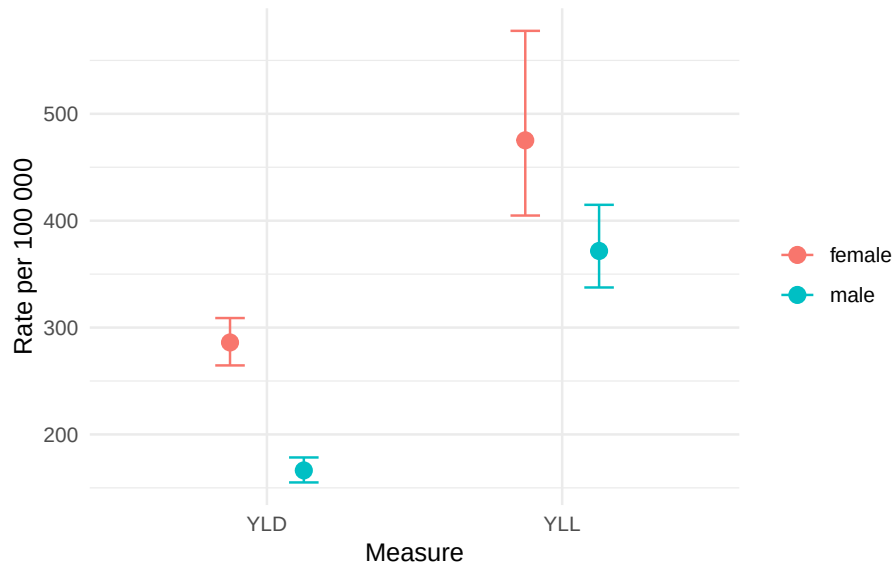


Figure 7: PIA-attributable YLD and YLL-Rate per 100 000 (2022)

Aggregation per region

Regional population totals for 2022 were calculated for the French-, German-, and Italian-speaking areas by summing the disaggregated population data for each language group. The calculation includes only individuals aged 20 years and older, aligning with the age criteria used in the YLD + YLL estimations.

```

#Population per Region
pop.french <- sum(d.population_region$pop_region
  [d.population_region$language == "french"])

```

```

pop.german <- sum(d.population_region$pop_region
                 [d.population_region$language == "german"])
pop.italian <- sum(d.population_region$pop_region
                  [d.population_region$language == "italian"])

# combined total for the population 20+ for check
pop.french+pop.german+pop.italian

```

```
[1] 14930983
```

YLD total by region

Total PIA-attributable YLD values were calculated for each region — French, German, and Italian — using bootstrap iterations.

```

# Total YLD per bootstrap iteration for each language region
attr.yld.total.french <-
  rowSums(d.attr.yld.boot [, grep("french",
                                   colnames(d.attr.yld.boot))],
          na.rm = TRUE)

attr.yld.total.german <-
  rowSums(d.attr.yld.boot [, grep("german",
                                   colnames(d.attr.yld.boot))],
          na.rm = TRUE)

attr.yld.total.italian <-
  rowSums(d.attr.yld.boot [, grep("italian",
                                   colnames(d.attr.yld.boot))],
          na.rm = TRUE)

```

```

#Totals by region YLD

# mean and 95%-CI YLD, French-speaking region
mean.yld.total.french <- mean(attr.yld.total.french,
                              na.rm = TRUE)
lower.yld.total.french <- quantile(attr.yld.total.french,
                                   probs = 0.025, na.rm = TRUE)
upper.yld.total.french <- quantile(attr.yld.total.french,
                                   probs = 0.975, na.rm = TRUE)

# mean and 95%-CI YLD, German-speaking region
mean.yld.total.german <- mean(attr.yld.total.german,

```

```

                                na.rm = TRUE)
lower.yld.total.german <- quantile(attr.yld.total.german,
                                probs = 0.025, na.rm = TRUE)
upper.yld.total.german <- quantile(attr.yld.total.german,
                                probs = 0.975, na.rm = TRUE)

# mean and 95%-CI YLD, Italian-speaking region
mean.yld.total.italian <- mean(attr.yld.total.italian,
                                na.rm = TRUE)
lower.yld.total.italian <- quantile(attr.yld.total.italian,
                                probs = 0.025, na.rm = TRUE)
upper.yld.total.italian <- quantile(attr.yld.total.italian,
                                probs = 0.975, na.rm = TRUE)

```

Attributable YLD Rates by region (per 100 000 population) attributable to PIA were calculated for each language region by dividing region-specific YLD totals from each bootstrap iteration by the respective population size, and scaling to 100 000 population.

```

# Rates
yld.rate.total.french.boot <- (attr.yld.total.french /
                                pop.french) * 100000
yld.rate.total.german.boot <- (attr.yld.total.german /
                                pop.german) * 100000
yld.rate.total.italian.boot <- (attr.yld.total.italian /
                                pop.italian) * 100000

mean.yld.rate.total.french <- mean(yld.rate.total.french.boot,
                                na.rm = TRUE)
lower.yld.rate.total.french <- quantile(yld.rate.total.french.boot,
                                probs = 0.025, na.rm = TRUE)
upper.yld.rate.total.french <- quantile(yld.rate.total.french.boot,
                                probs = 0.975, na.rm = TRUE)

mean.yld.rate.total.german <- mean(yld.rate.total.german.boot,
                                na.rm = TRUE)
lower.yld.rate.total.german <- quantile(yld.rate.total.german.boot,
                                probs = 0.025, na.rm = TRUE)
upper.yld.rate.total.german <- quantile(yld.rate.total.german.boot,
                                probs = 0.975, na.rm = TRUE)

mean.yld.rate.total.italian <- mean(yld.rate.total.italian.boot,
                                na.rm = TRUE)
lower.yld.rate.total.italian <- quantile(yld.rate.total.italian.boot,
                                probs = 0.025, na.rm = TRUE)

```

```
upper.yld.rate.total.italian <- quantile(yld.rate.total.italian.boot,
                                         probs = 0.975, na.rm = TRUE)
```

The calculated YLD rates for each region were compiled into a summary data frame (d.rate.yld.region). This table includes the mean rate and corresponding 95% confidence interval for the French-, German-, and Italian-speaking populations. Confidence intervals were formatted for clarity, supporting regional comparisons of PIA-attributable morbidity.

```
# Create summary data frame of YLD by region (rates + absolute numbers)
d.yld.region <- data.frame(
  region      = c("french", "german", "italian"),
  mean_rate   = c(mean.yld.rate.total.french,
                  mean.yld.rate.total.german,
                  mean.yld.rate.total.italian),
  lower_rate  = c(lower.yld.rate.total.french,
                  lower.yld.rate.total.german,
                  lower.yld.rate.total.italian),
  upper_rate  = c(upper.yld.rate.total.french,
                  upper.yld.rate.total.german,
                  upper.yld.rate.total.italian),
  mean_abs    = c(mean.yld.total.french,
                  mean.yld.total.german,
                  mean.yld.total.italian),
  lower_abs   = c(lower.yld.total.french,
                  lower.yld.total.german,
                  lower.yld.total.italian),
  upper_abs   = c(upper.yld.total.french,
                  upper.yld.total.german,
                  upper.yld.total.italian)) %>%
  mutate(
    ci_rate = paste0(sprintf("%.2f", mean_rate),
                      ", 95% CI [", sprintf("%.2f", lower_rate),
                      ", ", sprintf("%.2f", upper_rate), "]",
    ci_abs  = paste0(format(round(mean_abs, 0), big.mark = ""),
                      ", 95% CI [", format(round(lower_abs, 0),
                                           big.mark = ""),
                      ", ", format(round(upper_abs, 0),
                                   big.mark = ""), "])"))
```

YLL totals by region

YLL due to PIA was estimated separately for the French-, German-, and Italian-speaking regions. Bootstrap-based totals were calculated by

summing the region-specific columns for each iteration.

```
# Total YLL per bootstrap iteration for each region
attr.yll.total.french <-
  rowSums(d.attr.yll.boot[, grep("french",
                                colnames(d.attr.yll.boot))],
          na.rm = TRUE)

attr.yll.total.german <-
  rowSums(d.attr.yll.boot[, grep("german",
                                colnames(d.attr.yll.boot))],
          na.rm = TRUE)

attr.yll.total.italian <-
  rowSums(d.attr.yll.boot[, grep("italian",
                                colnames(d.attr.yll.boot))],
          na.rm = TRUE)

#Totals by region YLL
# Total YLL per bootstrap iteration for each language region
attr.yll.total.french <-
  rowSums(d.attr.yll.boot[, grep("french",
                                colnames(d.attr.yll.boot))],
          na.rm = TRUE)

attr.yll.total.german <-
  rowSums(d.attr.yll.boot[, grep("german",
                                colnames(d.attr.yll.boot))],
          na.rm = TRUE)

attr.yll.total.italian <-
  rowSums(d.attr.yll.boot[, grep("italian",
                                colnames(d.attr.yll.boot))],
          na.rm = TRUE)

# mean and 95%-CI YLL, French-speaking region
mean.yll.total.french <- mean(attr.yll.total.french, na.rm = TRUE)
lower.yll.total.french <- quantile(attr.yll.total.french,
                                  probs = 0.025, na.rm = TRUE)
upper.yll.total.french <- quantile(attr.yll.total.french,
                                  probs = 0.975, na.rm = TRUE)

# mean and 95%-CI YLL, German-speaking region
mean.yll.total.german <- mean(attr.yll.total.german, na.rm = TRUE)
lower.yll.total.german <- quantile(attr.yll.total.german,
                                  probs = 0.025, na.rm = TRUE)
upper.yll.total.german <- quantile(attr.yll.total.german,
```

```

                                probs = 0.975, na.rm = TRUE)

# mean and 95%-CI YLL, Italian-speaking region
mean.yll.total.italian <- mean(attr.yll.total.italian, na.rm = TRUE)
lower.yll.total.italian <- quantile(attr.yll.total.italian,
                                probs = 0.025, na.rm = TRUE)
upper.yll.total.italian <- quantile(attr.yll.total.italian,
                                probs = 0.975, na.rm = TRUE)

```

Attributable YLL Rates by Region (per 100 000 population) attributable to PIA were calculated for each region by dividing region-specific YLL totals by the respective population and scaling the result to 100 000 population.

```

# Calculate YLL rates per 100 000 population by region
yll.rate.total.french.boot <- (attr.yll.total.french /
                                pop.french) * 100000
yll.rate.total.german.boot <- (attr.yll.total.german /
                                pop.german) * 100000
yll.rate.total.italian.boot <- (attr.yll.total.italian /
                                pop.italian) * 100000

# mean and 95% confidence intervals
# French
mean.yll.rate.total.french <- mean(yll.rate.total.french.boot,
                                na.rm = TRUE)
lower.yll.rate.total.french <- quantile(yll.rate.total.french.boot,
                                probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.french <- quantile(yll.rate.total.french.boot,
                                probs = 0.975, na.rm = TRUE)

# German
mean.yll.rate.total.german <- mean(yll.rate.total.german.boot,
                                na.rm = TRUE)
lower.yll.rate.total.german <- quantile(yll.rate.total.german.boot,
                                probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.german <- quantile(yll.rate.total.german.boot,
                                probs = 0.975, na.rm = TRUE)

# Italian
mean.yll.rate.total.italian <- mean(yll.rate.total.italian.boot,
                                na.rm = TRUE)
lower.yll.rate.total.italian <- quantile(yll.rate.total.italian.boot,
                                probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.italian <- quantile(yll.rate.total.italian.boot,

```

```
probs = 0.975, na.rm = TRUE)
```

The YLL rates attributable to PIA were compiled into a data frame (d.rate.yll.region) for the French-, German-, and Italian-speaking regions. This table includes the mean YLL rate and the corresponding 95% confidence interval for each region.

```
# Create summary data frame of YLL by region (rates + absolute numbers)
d.yll.region <- data.frame(
  region      = c("french", "german", "italian"),
  mean_rate   = c(mean.yll.rate.total.french,
                  mean.yll.rate.total.german,
                  mean.yll.rate.total.italian),
  lower_rate  = c(lower.yll.rate.total.french,
                  lower.yll.rate.total.german,
                  lower.yll.rate.total.italian),
  upper_rate  = c(upper.yll.rate.total.french,
                  upper.yll.rate.total.german,
                  upper.yll.rate.total.italian),
  mean_abs    = c(mean.yll.total.french,
                  mean.yll.total.german,
                  mean.yll.total.italian),
  lower_abs   = c(lower.yll.total.french,
                  lower.yll.total.german,
                  lower.yll.total.italian),
  upper_abs   = c(upper.yll.total.french,
                  upper.yll.total.german,
                  upper.yll.total.italian)) %>%
mutate(
  ci_rate = paste0(sprintf("%.2f", mean_rate),
                    ", 95% CI [", sprintf("%.2f", lower_rate),
                    ", ", sprintf("%.2f", upper_rate), "]"),
  ci_abs  = paste0(format(round(mean_abs, 0), big.mark = ""),
                    ", 95% CI [", format(round(lower_abs, 0),
                                          big.mark = ""),
                    ", ", format(round(upper_abs, 0),
                                  big.mark = ""), "]"))
```

Aggregation by region and sex

Sex-specific population counts for each region were calculated using disaggregated 2022 data. Totals were derived by summing individuals identified by both language region and sex.

```

# Population by region and sex (2022)
pop.french.female <- sum(d.population_region$pop_region
                        [d.population_region$language == "french" &
                          d.population_region$sex == "female"])

pop.french.male <- sum(d.population_region$pop_region
                      [d.population_region$language == "french" &
                        d.population_region$sex == "male"])

pop.german.female <- sum(d.population_region$pop_region
                        [d.population_region$language == "german" &
                          d.population_region$sex == "female"])

pop.german.male <- sum(d.population_region$pop_region
                      [d.population_region$language == "german" &
                        d.population_region$sex == "male"])

pop.italian.female <- sum(d.population_region$pop_region
                          [d.population_region$language == "italian" &
                            d.population_region$sex == "female"])

pop.italian.male <- sum(d.population_region$pop_region
                       [d.population_region$language == "italian" &
                         d.population_region$sex == "male"])

# Consistency check for population
pop.italian.female+pop.italian.male
pop.italian

pop.french.female+pop.french.male
pop.french

pop.german.female + pop.german.male
pop.german

pop.german+pop.french+pop.italian

```

YLD total per region and sex

PIA-attributable YLD Totals by Region and Sex YLD values attributable to PIA were calculated separately for each combination of language region and sex using bootstrap iterations. To ensure accuracy in the male estimates, care was taken to explicitly exclude female-related columns when filtering for male data. Row-wise sums were computed for each

group, generating sex- and region-specific YLD totals for further analysis.

```
# YLD totals per bootstrap iteration by language region and sex
# Note: Exclude female when filtering for male columns to avoid
#misclassification when using grepl
# na.rm TRUE: ignoring missing data

# French - Female
attr.yld.total.french.female <- rowSums(
  d.attr.yld.boot[, grepl("french", colnames(d.attr.yld.boot)) &
                    grepl("female", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

# French - Male
attr.yld.total.french.male <- rowSums(
  d.attr.yld.boot[, str_detect(colnames(d.attr.yld.boot),
                              "^.*\\.french\\.\\.\\.male$")],
  na.rm = TRUE)

# German - Female
attr.yld.total.german.female <- rowSums(
  d.attr.yld.boot[, grepl("german", colnames(d.attr.yld.boot)) &
                    grepl("female", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

# German - Male
attr.yld.total.german.male <- rowSums(
  d.attr.yld.boot[, str_detect(colnames(d.attr.yld.boot),
                              "^.*\\.german\\.\\.\\.male$")],
  na.rm = TRUE)

# Italian - Female
attr.yld.total.italian.female <- rowSums(
  d.attr.yld.boot[, grepl("italian", colnames(d.attr.yld.boot)) &
                    grepl("female", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

# Italian - Male
attr.yld.total.italian.male <- rowSums(
  d.attr.yld.boot[, str_detect(colnames(d.attr.yld.boot),
                              "^.*\\.italian\\.\\.\\.male$")],
  na.rm = TRUE)
```

Attributable YLD Rates by Region and Sex were calculated for each subgroup by dividing bootstrap-based YLD totals by the corresponding sex-

specific population and scaling to 100 000 population. For each region-sex combination, the mean rate and 95% confidence interval were computed.

```
# Calculate YLD rates per 100 000 by region and sex
yld.rate.total.french.female.boot <- (attr.yld.total.french.female /
                                       pop.french.female) * 100000
yld.rate.total.french.male.boot <- (attr.yld.total.french.male /
                                    pop.french.male) * 100000
yld.rate.total.german.female.boot <- (attr.yld.total.german.female /
                                       pop.german.female) * 100000
yld.rate.total.german.male.boot <- (attr.yld.total.german.male /
                                    pop.german.male) * 100000
yld.rate.total.italian.female.boot <- (attr.yld.total.italian.female /
                                       pop.italian.female) * 100000
yld.rate.total.italian.male.boot <- (attr.yld.total.italian.male /
                                    pop.italian.male) * 100000

# Mean and 95% confidence intervals
# French
mean.yld.rate.total.french.female <-
  mean(yld.rate.total.french.female.boot,
       na.rm = TRUE)

lower.yld.rate.total.french.female <-
  quantile(yld.rate.total.french.female.boot,
           probs = 0.025, na.rm = TRUE)

upper.yld.rate.total.french.female <-
  quantile(yld.rate.total.french.female.boot,
           probs = 0.975, na.rm = TRUE)

mean.yld.rate.total.french.male <-
  mean(yld.rate.total.french.male.boot,
       na.rm = TRUE)

lower.yld.rate.total.french.male <-
  quantile(yld.rate.total.french.male.boot,
           probs = 0.025, na.rm = TRUE)

upper.yld.rate.total.french.male <-
  quantile(yld.rate.total.french.male.boot,
           probs = 0.975, na.rm = TRUE)

# German
mean.yld.rate.total.german.female <-
```

```

mean(yld.rate.total.german.female.boot, na.rm = TRUE)

lower.yld.rate.total.german.female <-
  quantile(yld.rate.total.german.female.boot, probs = 0.025,
    na.rm = TRUE)

upper.yld.rate.total.german.female <-
  quantile(yld.rate.total.german.female.boot, probs = 0.975,
    na.rm = TRUE)

mean.yld.rate.total.german.male <-
  mean(yld.rate.total.german.male.boot, na.rm = TRUE)

lower.yld.rate.total.german.male <-
  quantile(yld.rate.total.german.male.boot,
    probs = 0.025, na.rm = TRUE)

upper.yld.rate.total.german.male <-
  quantile(yld.rate.total.german.male.boot,
    probs = 0.975, na.rm = TRUE)

# Italian
mean.yld.rate.total.italian.female <-
  mean(yld.rate.total.italian.female.boot, na.rm = TRUE)

lower.yld.rate.total.italian.female <-
  quantile(yld.rate.total.italian.female.boot,
    probs = 0.025, na.rm = TRUE)

upper.yld.rate.total.italian.female <-
  quantile(yld.rate.total.italian.female.boot,
    probs = 0.975, na.rm = TRUE)

mean.yld.rate.total.italian.male <-
  mean(yld.rate.total.italian.male.boot, na.rm = TRUE)

lower.yld.rate.total.italian.male <-
  quantile(yld.rate.total.italian.male.boot,
    probs = 0.025, na.rm = TRUE)

upper.yld.rate.total.italian.male <-
  quantile(yld.rate.total.italian.male.boot,
    probs = 0.975, na.rm = TRUE)

```

The sex-disaggregated YLD rates for each language region were com-

binned into a data frame (d.rate.yld.region.sex).

```
# Create data frame with YLD rates by region and sex
d.rate.yld.region.sex <- data.frame(
  region = rep(c("french", "german", "italian"), each = 2),
  sex     = rep(c("female", "male"), times = 3),
  mean    = c(
    mean.yld.rate.total.french.female,
    mean.yld.rate.total.french.male,
    mean.yld.rate.total.german.female,
    mean.yld.rate.total.german.male,
    mean.yld.rate.total.italian.female,
    mean.yld.rate.total.italian.male),
  lower   = c(
    lower.yld.rate.total.french.female,
    lower.yld.rate.total.french.male,
    lower.yld.rate.total.german.female,
    lower.yld.rate.total.german.male,
    lower.yld.rate.total.italian.female,
    lower.yld.rate.total.italian.male),
  upper   = c(
    upper.yld.rate.total.french.female,
    upper.yld.rate.total.french.male,
    upper.yld.rate.total.german.female,
    upper.yld.rate.total.german.male,
    upper.yld.rate.total.italian.female,
    upper.yld.rate.total.italian.male)) %>%
mutate(ci = paste0(sprintf("%.2f", lower), " ",
                    sprintf("%.2f", upper)),
       mean_ci = paste0(sprintf("%.2f", mean), " [",
                        sprintf("%.2f", lower), " ",
                        sprintf("%.2f", upper), "]""))
```

```
d.rate.yld.region.sex %>%
mutate(
  region = case_when(
    region == "french" ~ "French",
    region == "german" ~ "German",
    region == "italian" ~ "Italian",
    TRUE ~ region)) %>%
ggplot(aes(x = region, y = mean, color = sex)) +
geom_point(position = position_dodge(width = 0.5), size = 3) +
geom_errorbar(
  aes(ymin = lower, ymax = upper),
  position = position_dodge(width = 0.5),
```

```
width = 0.2) +
labs(
  x = "Region",
  y = "Rate per 100 000",
  color = "Sex") +
theme_minimal()
```

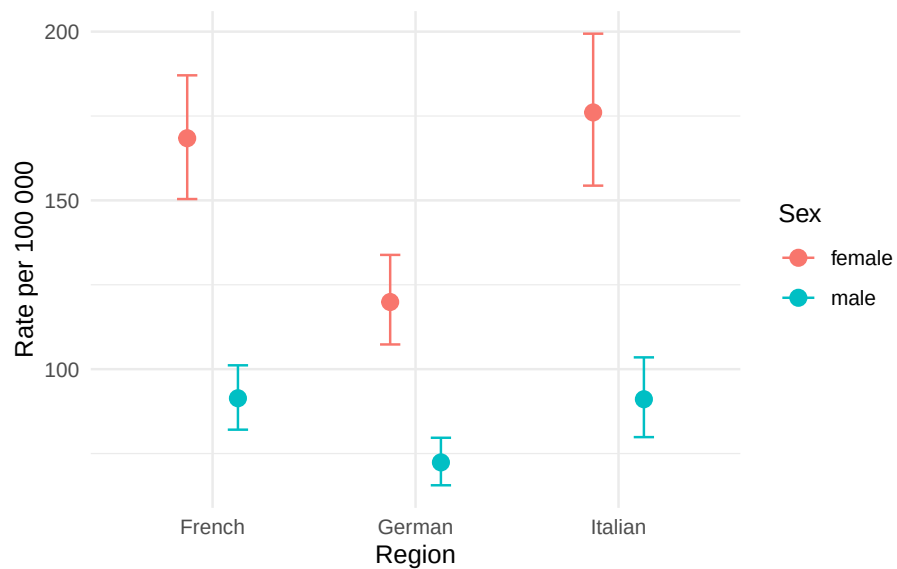


Figure 8: PIA attributable YLD rates per 100 000 by region and sex (2022)

YLL total per region and sex

YLL values due to PIA were calculated separately for each combination of region and sex using bootstrap iterations. Columns were carefully filtered to avoid misclassification—especially ensuring that female values were not inadvertently included in male estimates.

```
# YLL totals per bootstrap iteration by language region and sex

# French - Female
attr.yll.total.french.female <- rowSums(
  d.attr.yll.boot[, grepl("french", colnames(d.attr.yll.boot)) &
    grepl("female", colnames(d.attr.yll.boot))],
  na.rm = TRUE)
```

```

# French - Male
attr.yll.total.french.male <-
  rowSums(d.attr.yll.boot[, str_detect(colnames(d.attr.yll.boot),
                                         "^.*\\.french\\.\\.\\.male$")],
          na.rm = TRUE)

# German - Female
attr.yll.total.german.female <-
  rowSums(
    d.attr.yll.boot[, grepl("german", colnames(d.attr.yll.boot)) &
                     grepl("female", colnames(d.attr.yll.boot))],
    na.rm = TRUE)

# German - Male
attr.yll.total.german.male <-
  rowSums(d.attr.yll.boot[, str_detect(colnames(d.attr.yll.boot),
                                         "^.*\\.german\\.\\.\\.male$")],
          na.rm = TRUE)

# Italian - Female
attr.yll.total.italian.female <-
  rowSums(d.attr.yll.boot[, grepl("italian", colnames(d.attr.yll.boot)) &
                           grepl("female", colnames(d.attr.yll.boot))],
          na.rm = TRUE)

# Italian - Male
attr.yll.total.italian.male <-
  rowSums(d.attr.yll.boot[, str_detect(colnames(d.attr.yll.boot),
                                         "^.*\\.italian\\.\\.\\.male$")],
          na.rm = TRUE)

```

Attributable YLL Rates by Region and Sex were computed by dividing the respective bootstrap totals by population counts and standardizing per 100 000 individuals. This approach enables direct comparisons across demographic subgroups.

```

# Calculate YLL rates per 100 000 by region and sex
yll.rate.total.french.female.boot <-
  (attr.yll.total.french.female / pop.french.female) * 100000
yll.rate.total.french.male.boot <-
  (attr.yll.total.french.male / pop.french.male) * 100000
yll.rate.total.german.female.boot <-
  (attr.yll.total.german.female / pop.german.female) * 100000
yll.rate.total.german.male.boot <-
  (attr.yll.total.german.male / pop.german.male) * 100000

```

```

yll.rate.total.italian.female.boot <-
  (attr.yll.total.italian.female / pop.italian.female) * 100000
yll.rate.total.italian.male.boot <-
  (attr.yll.total.italian.male / pop.italian.male) * 100000

# Compute mean and 95% confidence intervals
# French
mean.yll.rate.total.french.female <-
  mean(yll.rate.total.french.female.boot, na.rm = TRUE)
lower.yll.rate.total.french.female <-
  quantile(yll.rate.total.french.female.boot, probs = 0.025,
    na.rm = TRUE)
upper.yll.rate.total.french.female <-
  quantile(yll.rate.total.french.female.boot, probs = 0.975,
    na.rm = TRUE)

mean.yll.rate.total.french.male <-
  mean(yll.rate.total.french.male.boot, na.rm = TRUE)
lower.yll.rate.total.french.male <-
  quantile(yll.rate.total.french.male.boot, probs = 0.025,
    na.rm = TRUE)
upper.yll.rate.total.french.male <-
  quantile(yll.rate.total.french.male.boot, probs = 0.975,
    na.rm = TRUE)

# German
mean.yll.rate.total.german.female <-
  mean(yll.rate.total.german.female.boot, na.rm = TRUE)
lower.yll.rate.total.german.female <-
  quantile(yll.rate.total.german.female.boot,
    probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.german.female <-
  quantile(yll.rate.total.german.female.boot,
    probs = 0.975, na.rm = TRUE)

mean.yll.rate.total.german.male <-
  mean(yll.rate.total.german.male.boot, na.rm = TRUE)
lower.yll.rate.total.german.male <-
  quantile(yll.rate.total.german.male.boot,
    probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.german.male <-
  quantile(yll.rate.total.german.male.boot,
    probs = 0.975, na.rm = TRUE)

# Italian

```

```

mean.yll.rate.total.italian.female <-
  mean(yll.rate.total.italian.female.boot,
        na.rm = TRUE)

lower.yll.rate.total.italian.female <-
  quantile(yll.rate.total.italian.female.boot,
            probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.italian.female <-
  quantile(yll.rate.total.italian.female.boot,
            probs = 0.975, na.rm = TRUE)

mean.yll.rate.total.italian.male <-
  mean(yll.rate.total.italian.male.boot, na.rm = TRUE)
lower.yll.rate.total.italian.male <-
  quantile(yll.rate.total.italian.male.boot,
            probs = 0.025, na.rm = TRUE)
upper.yll.rate.total.italian.male <-
  quantile(yll.rate.total.italian.male.boot,
            probs = 0.975, na.rm = TRUE)

# create df
d.rate.yll.region.sex <- data.frame(
  region = rep(c("french", "german", "italian"), each = 2),
  sex    = rep(c("female", "male"), times = 3),
  mean   = c(
    mean.yll.rate.total.french.female,
    mean.yll.rate.total.french.male,
    mean.yll.rate.total.german.female,
    mean.yll.rate.total.german.male,
    mean.yll.rate.total.italian.female,
    mean.yll.rate.total.italian.male),
  lower  = c(
    lower.yll.rate.total.french.female,
    lower.yll.rate.total.french.male,
    lower.yll.rate.total.german.female,
    lower.yll.rate.total.german.male,
    lower.yll.rate.total.italian.female,
    lower.yll.rate.total.italian.male),
  upper  = c(
    upper.yll.rate.total.french.female,
    upper.yll.rate.total.french.male,
    upper.yll.rate.total.german.female,
    upper.yll.rate.total.german.male,
    upper.yll.rate.total.italian.female,
    upper.yll.rate.total.italian.male)) %>%

```

Table 4: PIA attributable YLL rates per 100 000 by region and sex (2022)

Region	Sex	Mean YLL-Rate (95%-CI)
French	female	252.07 [203.06, 330.90]
French	male	189.51 [165.39, 220.32]
German	female	207.38 [166.65, 273.56]
German	male	165.79 [145.26, 192.92]
Italian	female	339.73 [269.63, 453.80]
Italian	male	232.28 [194.29, 276.36]

```
mutate(ci = paste0(sprintf("%.2f", lower), ", ",
                     sprintf("%.2f", upper)),
       mean_ci = paste0(sprintf("%.2f", mean), " [",
                         sprintf("%.2f", lower), ", ",
                         sprintf("%.2f", upper), "]))
```

```
d.rate.yll.region.sex %>%
  mutate(
    region = case_when(
      region == "french" ~ "French",
      region == "german" ~ "German",
      region == "italian" ~ "Italian",
      TRUE ~ region
    ) %>%
  select(region, sex, mean_ci) %>%
  kbl(
    digits = 2,
    col.names = c("Region", "Sex", "Mean YLL-Rate (95%-CI)")
  ) %>%
  kable_styling(full_width = FALSE, position = "center")
```

```
d.rate.yll.region.sex %>%
  mutate(
    region = case_when(
      region == "french" ~ "French",
      region == "german" ~ "German",
      region == "italian" ~ "Italian",
      TRUE ~ region
    ) %>%

  ggplot(aes(x = region, y = mean, color = sex)) +
  geom_point(position = position_dodge(width = 0.5), size = 3) +
  geom_errorbar(
```

```

aes(ymin = lower, ymax = upper),
position = position_dodge(width = 0.5),
width = 0.2) +
labs(
  x = "Region",
  y = "Rate per 100 000",
  color = "Sex") +
theme_minimal()

```

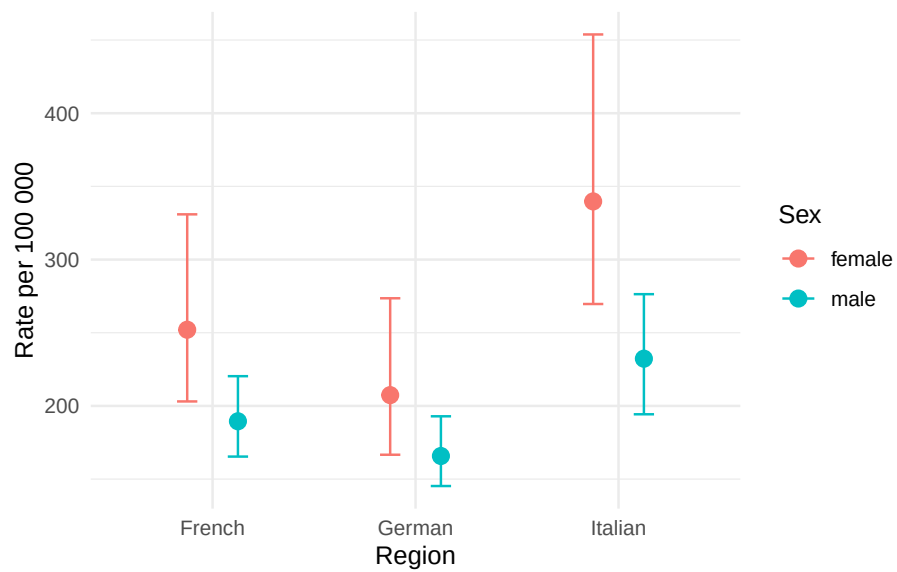


Figure 9: PIA attributable YLL rates per 100 000 by region and sex (2022)

Aggregation by disease

YLL

```

# List of diseases
diseases <- d.riskratios$disease

# population
pop_ch_total

```

[1] 7060172

```

# results
result_list <- vector("list", length(diseases))
names(result_list) <- diseases

for (d in diseases) {

  # select columns for disease
  sel <- vapply(colnames(d.attr.yll.boot), function(name) {
    parts <- strsplit(name, "\\.")[[1]]
    length(parts) == 4 && parts[1] == d
  }, logical(1))

  if (!any(sel)) {
    result_list[[d]] <- data.frame(
      disease = d, mean = NA_real_, lower = NA_real_,
      upper = NA_real_,
      stringsAsFactors = FALSE
    )
    next
  }

  vec <- rowSums(d.attr.yll.boot[, sel, drop = FALSE],
    na.rm = TRUE)
  vec.rate.yll <- vec / pop_ch_total * 100000

  result_list[[d]] <- data.frame(
    disease = d,
    mean = mean(vec.rate.yll, na.rm = TRUE),
    lower = quantile(vec.rate.yll, probs = 0.025, na.rm = TRUE),
    upper = quantile(vec.rate.yll, probs = 0.975, na.rm = TRUE),
    stringsAsFactors = FALSE
  )
}

# result_list contains for each disease: disease, mean, lower, upper
d.rate.yll.disease.total <- bind_rows(result_list) %>%
  select(disease, mean, lower, upper) %>%
  arrange(desc(mean))

rownames(d.rate.yll.disease.total) <- NULL

```

YLD

```
# list of disease
diseases <- d.riskratios$disease

# results
result_list_yld <- vector("list", length(diseases))
names(result_list_yld) <- diseases

for (d in diseases) {

  # select columns
  sel <- vapply(colnames(d.attr.yld.boot), function(name) {
    parts <- strsplit(name, "\\.")[[1]]
    length(parts) == 4 && parts[1] == d
  }, logical(1))

  if (!any(sel)) {
    result_list_yld[[d]] <- data.frame(
      disease = d,
      mean = NA_real_,
      lower = NA_real_,
      upper = NA_real_,
      stringsAsFactors = FALSE
    )
    next
  }

  # Bootstrap-values
  vec <- rowSums(d.attr.yld.boot[, sel, drop = FALSE],
    na.rm = TRUE)

  # Rate per 100'000
  vec.rate.yld <- vec / pop_ch_total * 100000

  # save
  result_list_yld[[d]] <- data.frame(
    disease = d,
    mean = mean(vec.rate.yld, na.rm = TRUE),
    lower = quantile(vec.rate.yld, probs = 0.025,
      na.rm = TRUE),
    upper = quantile(vec.rate.yld, probs = 0.975,
      na.rm = TRUE),
    stringsAsFactors = FALSE
  )
}
```

```

}

# merge in one DataFrame
d.rate.yld.disease <- bind_rows(result_list_yld) %>%
  mutate(
    ci      = sprintf("%.2f, %.2f", lower, upper),
    mean_ci = sprintf("%.2f [%.2f, %.2f]", mean, lower, upper))

rownames(d.rate.yld.disease) <- NULL

# result_list_yld contains for each Disease: disease, mean, lower, upper
d.rate.yld.disease.total <- bind_rows(result_list_yld) %>%
  select(disease, mean, lower, upper) %>%
  arrange(desc(mean))

rownames(d.rate.yld.disease.total) <- NULL

```

Aggregation by Region, Disease, and Sex

YLL

To compute detailed PIA-attributable YLL rates stratified by region, disease, and sex, a loop iterated through all combinations of these variables. For each group, relevant columns in the bootstrap dataset were filtered and the summed YLL values were scaled to rates using the appropriate region- and sex-specific population. Summary statistics—mean and 95% confidence intervals—were calculated for each subgroup and stored as individual data frames in a result list for further processing.

```

# List of diseases, sexes, and regions
diseases <- d.riskratios$disease
sexes    <- unique(d.global$sex)
regions  <- unique(d.global$language)

# Initialize empty list for results
result_list <- list()

# Loop over regions, diseases, and sexes
for (reg in regions) {
  for (d in diseases) {
    for (s in sexes) {

      # Identify matching columns (exact match on region, disease,

```

```

#and sex)
sel <- sapply(colnames(d.attr.yll.boot), function(name) {
  parts <- strsplit(name, "\\\\.")[[1]]
  length(parts) == 4 &&
  parts[1] == d &&
  parts[2] == reg &&
  parts[4] == s
})

# Sum bootstrap values for the selected columns
# use drop = FALSE to prevent loss of structure and to ensure
#the dataframe structure
vec <- rowSums(d.attr.yll.boot[, sel, drop = FALSE],
              na.rm = TRUE)

# Retrieve population for region and sex
pop_s <- get(paste0("pop.", reg, ".", s))

# Calculate YLL rate per 100 000
vec.rate.yll <- vec / pop_s * 100000

# Compute mean and 95% confidence interval
mean_val <- mean(vec.rate.yll, na.rm = TRUE)
lower_val <- quantile(vec.rate.yll, probs = 0.025,
                     na.rm = TRUE)
upper_val <- quantile(vec.rate.yll, probs = 0.975,
                     na.rm = TRUE)

# Store result in list
key <- paste(reg, d, s, sep = "_")
result_list[[key]] <- data.frame(
  region = reg,
  disease = d,
  sex = s,
  mean = mean_val,
  lower = lower_val,
  upper = upper_val,
  stringsAsFactors = FALSE)
})

# Combine all subgroup results into one data frame
d.rate.yll.disease <- bind_rows(result_list) %>%
  mutate(
    ci = paste0(sprintf("%.2f", lower), ", ",
                 sprintf("%.2f", upper)))

```

```
# Add formatted column: "mean (lower - upper)"
d.rate.yll.disease <- d.rate.yll.disease %>%
  mutate(
    mean_ci = paste0(sprintf("%.2f", mean), " [",
                      sprintf("%.2f", lower), ", ",
                      sprintf("%.2f", upper), "]")

# Remove row names
rownames(d.rate.yll.disease) <- NULL
```

Check the results: region, disease, sex

```
# CHD YLL - French - Female
attr.yll.chd.french.female <- rowSums(
  d.attr.yll.boot[, grepl("french", colnames(d.attr.yll.boot)) &
                    grepl("female", colnames(d.attr.yll.boot)) &
                    grepl("chd", colnames(d.attr.yll.boot))],
  na.rm = TRUE)

# CHD YLL rate per 100 000
yll.rate.chd.french.female.boot <- (attr.yll.chd.french.female /
                                     pop.french.female) * 100000

# Rates Mean and 95% CI for CHD YLL
mean.attr.yll.chd.french.female <-
  mean(yll.rate.chd.french.female.boot)

lower.attr.yll.chd.french.female <-
  quantile(yll.rate.chd.french.female.boot, probs = 0.025)
upper.attr.yll.chd.french.female <-
  quantile(yll.rate.chd.french.female.boot, probs = 0.975)

#check if results of d.rate.yll.disease (generated by loop) is
#matching the individual calculation
mean.attr.yll.chd.french.female
d.rate.yll.disease %>%
  filter(sex == "female",
         disease == "chd",
         region == "french")

# ca_lung YLL - Italian - Male
attr.yll.ca_lung.italian.male <-
  rowSums(d.attr.yll.boot[,
                          grepl("italian", colnames(d.attr.yll.boot)) &
                          grepl("[.]male", colnames(d.attr.yll.boot)) &
```

Table 5: PIA attributable YLL-Rate per 100 000 stratified by region, disease and sex (2022)

sex	Alzheimer's disease and dementia	Breast cancer	Colon cancer	Coronary heart disease	Depression	Chlamydia, nonchlamydia type 2	Endometrial cancer	Cervix cancer	Heart failure	Liver cancer	Low back pain	Lung cancer	Myeloid leukemias	Myeloma	Stroke
French	65.68 [26.28, 140.44]	6.18 [2.57, 10.80]	6.02 [2.56, 9.47]	52.64 [45.45, 61.45]	0.00 [0.00, 0.00]	5.52 [3.89, 7.24]	2.23 [1.33, 3.14]	6.59 [3.97, 9.25]	16.81 [5.89, 28.14]	5.11 [3.38, 6.87]	0.00 [0.00, 0.00]	26.13 [22.60, 30.18]	6.82 [3.85, 10.00]	5.25 [2.61, 8.11]	37.05 [26.65, 48.26]
German	19.69 [7.47, 41.86]	6.00 [3.02, 8.94]	5.79 [2.93, 8.46]	16.46 [15.55, 64.45]	0.00 [0.00, 0.00]	5.23 [3.87, 6.76]	0.00 [0.00, 0.00]	8.41 [4.46, 11.94]	5.97 [2.87, 9.46]	6.46 [4.03, 9.45]	0.00 [0.00, 0.00]	35.19 [25.52, 35.34]	8.39 [5.99, 12.24]	5.43 [3.04, 8.45]	22.77 [16.09, 29.65]
Italian	15.44 [12.40, 189.11]	6.00 [4.12, 8.20]	4.79 [3.02, 7.18]	16.43 [16.58, 70.50]	0.00 [0.00, 0.00]	4.46 [3.24, 5.86]	1.92 [1.15, 2.45]	4.85 [2.89, 7.00]	14.22 [5.26, 26.04]	5.48 [3.26, 9.73]	0.00 [0.00, 0.00]	30.26 [17.38, 33.30]	5.38 [3.06, 8.09]	4.28 [2.15, 6.45]	60.11 [31.78, 109.68]
Male	16.11 [7.20, 40.59]	6.02 [3.05, 8.05]	4.73 [2.91, 7.24]	65.11 [51.78, 60.07]	0.00 [0.00, 0.00]	4.60 [3.52, 5.75]	0.00 [0.00, 0.00]	7.36 [4.83, 10.20]	5.60 [2.68, 9.50]	5.07 [3.08, 7.24]	0.00 [0.00, 0.00]	24.07 [20.69, 27.60]	6.83 [3.94, 10.18]	4.57 [2.60, 6.70]	19.71 [14.69, 24.91]
Female	90.71 [36.05, 189.67]	6.02 [3.28, 13.57]	7.75 [5.15, 11.00]	65.22 [59.66, 115.18]	0.00 [0.00, 0.00]	6.82 [4.76, 9.20]	2.87 [1.60, 4.14]	8.82 [5.22, 12.87]	23.00 [9.36, 36.04]	4.80 [2.89, 7.47]	0.00 [0.00, 0.00]	34.14 [28.30, 40.70]	8.72 [6.46, 13.00]	6.68 [3.16, 19.60]	51.67 [35.67, 69.51]
Male	23.38 [8.35, 51.92]	6.02 [3.05, 8.05]	6.00 [3.10, 11.00]	60.70 [50.51, 120.27]	0.00 [0.00, 0.00]	5.86 [3.86, 8.12]	0.00 [0.00, 0.00]	10.24 [6.52, 11.58]	6.96 [3.50, 11.30]	6.80 [4.68, 12.71]	0.00 [0.00, 0.00]	37.44 [27.61, 48.30]	6.10 [3.05, 16.45]	6.00 [3.20, 9.40]	26.71 [18.00, 40.44]

```

                                grepl("ca_lung", colnames(d.attr.yll.boot))],
                                na.rm = TRUE)

# ca_lung YLL rate per 100 000 - Italian - Male
yll.rate.ca_lung.italian.male.boot <-
  (attr.yll.ca_lung.italian.male / pop.italian.male) * 100000

# Mean and 95% CI of ca_lung YLL rate - Italian - Male
mean.attr.yll.ca_lung.italian.male <-
  mean(yll.rate.ca_lung.italian.male.boot)
lower.attr.yll.ca_lung.italian.male <-
  quantile(yll.rate.ca_lung.italian.male.boot,
            probs = 0.025)
upper.attr.yll.ca_lung.italian.male <-
  quantile(yll.rate.ca_lung.italian.male.boot,
            probs = 0.975)

# Filter to compare against static dataset
mean.attr.yll.ca_lung.italian.male
d.rate.yll.disease %>%
  filter(sex == "male",
         disease == "ca_lung",
         region == "italian")

```

For better display of the results results were filtered by region and transformed in a wide format. Three separate dataframes were created: - d.yll.french.wide - d.yll.german.wide - d.yll.italian.wide

The three dataframes were merged in one table.

```

#Plot: all region stacked (Shape = Region, Color = Sex)
ggplot(d.rate.yll.disease, aes(
  x = reorder(disease, -mean),
  y = mean,
  color = sex)) +
  geom_point(position = position_dodge(width = 0.7), size = 3) +
  geom_errorbar(
    aes(ymin = lower, ymax = upper),
    position = position_dodge(width = 0.7),

```

```

width = 0.2) +
scale_x_discrete(labels = mapping_disease) +
facet_wrap(~ region, ncol = 1) +
labs(
  x      = "Disease",
  y      = "Rate per 100 000",
  color  = "Sex") +
theme_minimal() +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1),
  strip.text  = element_text(face = "bold"))

```

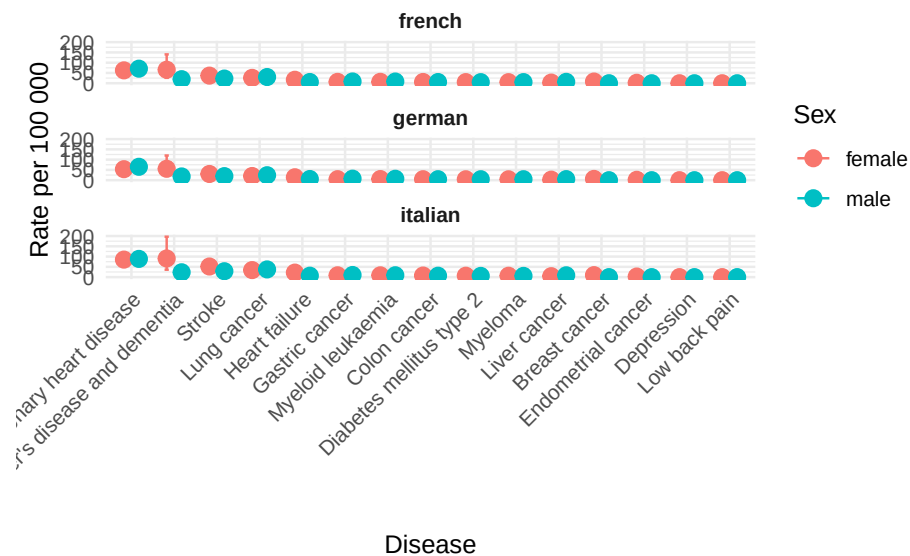


Figure 10: PIA attributable YLL (2022) – stratified by Region

YLD

YLD rates attributable to PIA were calculated for each region, disease, and sex by summing relevant bootstrap values, standardizing by region- and sex-specific populations, and computing mean rates with 95% confidence intervals. Results were stored in a list for further analysis.

The YLD results were merged into a single data frame.

Check values with static calculation

```

# CHD YLD - French - Female
attr.yld.chd.french.female <- rowSums(
  d.attr.yld.boot[, grepl("french", colnames(d.attr.yld.boot)) &
                    grepl("female", colnames(d.attr.yld.boot)) &
                    grepl("chd", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

# CHD YLD rate per 100 000
yld.rate.chd.french.female.boot <-
  (attr.yld.chd.french.female / pop.french.female) * 100000

# Mean and CI
mean.attr.yld.chd.french.female <-
  mean(yld.rate.chd.french.female.boot)
lower.attr.yld.chd.french.female <-
  quantile(yld.rate.chd.french.female.boot, probs = 0.025)
upper.attr.yld.chd.french.female <-
  quantile(yld.rate.chd.french.female.boot, probs = 0.975)

# Check with loop results
mean.attr.yld.chd.french.female
d.rate.yld.disease %>%
  filter(sex == "female",
         disease == "chd",
         region == "french")

```

A helper function was used to filter, reshape, and relabel the YLD data by region. This produced three region-specific wide-format tables: `d.yld.french.wide`, `d.yld.german.wide`, and `d.yld.italian.wide`, each displaying diseases as columns and sex as rows.

The three region-specific YLD tables were combined into a single dataset for presentation. The output was structured with visual grouping (`pack_rows`) to separate the French-, German-, and Italian-speaking regions for improved readability.

```

kbl(
  d.yld.combined,
  digits = 2,
  booktabs = TRUE
) %>%
  kable_styling(
    full_width = FALSE,
    position = "center",
    latex_options = "scale_down"
  )

```

Table 6: PIA attributable YLD-Rate per 100 000 stratified by region, disease and sex (2022)

sex	Alzheimer's disease and dementia	Breast cancer	Colon cancer	Coronary heart disease	Depression	Diabetes mellitus type 2	Endometrial cancer	Gastric cancer	Heart failure	Liver cancer	Low back pain	Lung cancer	Musculoskeletal	Myeloma	Stroke
French															
Female	22.78 [14.80, 32.70]	1.28 [0.87, 1.70]	0.54 [0.22, 0.91]	4.28 [1.17, 9.71]	79.89 [66.95, 93.88]	11.26 [8.84, 14.24]	0.38 [0.24, 0.53]	0.17 [0.10, 0.27]	0.74 [0.32, 1.20]	0.08 [0.02, 0.10]	37.46 [28.12, 46.80]	0.43 [0.34, 0.54]	0.58 [0.29, 0.91]	0.33 [0.17, 0.68]	8.03 [6.20, 10.28]
Male	7.54 [4.86, 11.02]	0.51 [0.30, 0.81]	0.53 [0.25, 0.84]	0.80 [0.38, 0.91]	53.74 [27.76, 48.58]	11.82 [9.58, 14.45]	0.05 [0.00, 0.00]	0.23 [0.14, 0.34]	0.35 [0.19, 0.50]	0.12 [0.06, 0.19]	22.59 [17.45, 28.15]	0.44 [0.35, 0.50]	0.82 [0.58, 0.97]	0.40 [0.21, 0.68]	6.19 [4.77, 7.95]
German															
Female	18.97 [12.25, 27.83]	0.98 [0.66, 1.30]	0.40 [0.15, 0.71]	5.80 [1.76, 8.87]	51.56 [44.63, 60.47]	0.80 [0.61, 11.21]	0.31 [0.20, 0.40]	0.11 [0.07, 0.20]	0.83 [0.38, 1.30]	0.05 [0.02, 0.08]	27.97 [20.79, 34.80]	0.33 [0.26, 0.40]	0.49 [0.24, 0.77]	0.29 [0.14, 0.51]	6.46 [4.93, 8.26]
Male	6.93 [4.55, 10.94]	0.09 [0.00, 0.81]	0.43 [0.20, 0.70]	0.08 [0.41, 7.83]	25.64 [21.45, 30.42]	0.89 [0.86, 11.82]	0.05 [0.00, 0.00]	0.20 [0.12, 0.29]	0.31 [0.17, 0.51]	0.09 [0.05, 0.14]	16.56 [12.91, 20.66]	0.36 [0.28, 0.44]	0.51 [0.29, 0.80]	0.34 [0.18, 0.58]	5.24 [4.09, 6.65]
Italian															
Female	31.11 [19.29, 46.11]	1.52 [0.98, 2.38]	0.70 [0.27, 1.21]	5.86 [1.16, 7.87]	68.09 [53.87, 79.87]	13.29 [9.99, 17.42]	0.48 [0.30, 0.68]	0.23 [0.13, 0.37]	1.00 [0.46, 1.71]	0.09 [0.03, 0.17]	42.70 [31.28, 55.91]	0.56 [0.45, 0.70]	0.73 [0.56, 1.20]	0.45 [0.20, 0.90]	11.24 [8.45, 14.74]
Male	9.23 [5.35, 14.62]	0.01 [0.00, 0.01]	0.82 [0.38, 1.34]	4.57 [0.20, 11.48]	27.97 [21.27, 35.50]	13.71 [10.39, 17.60]	0.00 [0.00, 0.00]	0.28 [0.16, 0.46]	0.41 [0.21, 0.68]	0.16 [0.08, 0.27]	20.83 [15.02, 27.43]	0.56 [0.39, 0.76]	0.68 [0.35, 1.10]	0.44 [0.21, 0.76]	7.82 [5.32, 10.53]

```

) %>%
pack_rows("French", 1, 2) %>%
pack_rows("German", 3, 4) %>%
pack_rows("Italian", 5, 6)

```

```

# Plot: all regions stacked (Shape = Region, Color = Sex)
ggplot(d.rate.yld.disease, aes(
  x = reorder(disease, -mean),
  y = mean,
  color = sex)) +
  geom_point(position = position_dodge(width = 0.7), size = 3) +
  geom_errorbar(
    aes(ymin = lower, ymax = upper),
    position = position_dodge(width = 0.7),
    width = 0.2) +
  scale_x_discrete(labels = mapping_disease) +
  facet_wrap(~ region, ncol = 1) +
  labs(
    x = "Disease",
    y = "Rate per 100 000",
    color = "Sex") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text = element_text(face = "bold"))

```

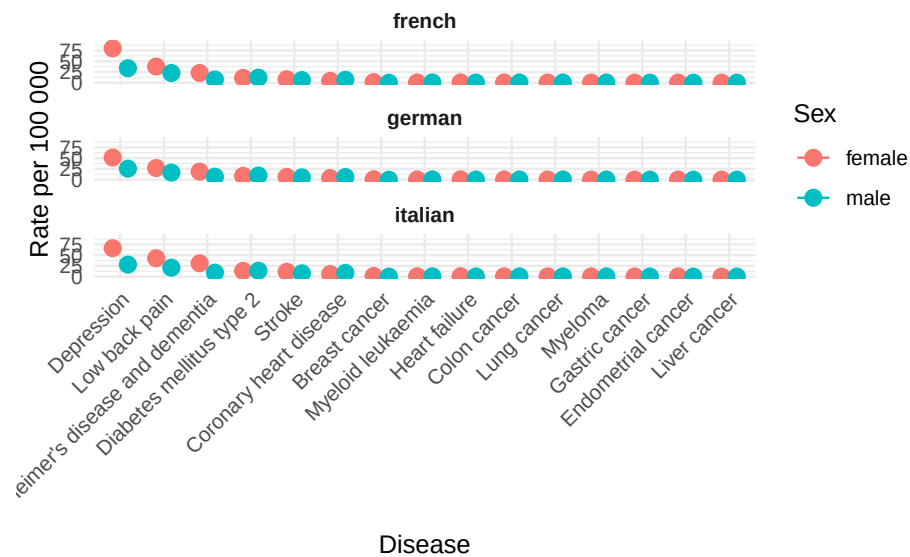


Figure 11: PIA attributable YLD (2022) – stratified by Region

Aggregation per region, sex for all cancers

YLD

```
# French
attr.yld.ca.french.female <- rowSums(d.attr.yld.boot
[, grepl("^ca_", colnames(d.attr.yld.boot)) &
  grepl("french", colnames(d.attr.yld.boot)) &
  grepl("female", colnames(d.attr.yld.boot))],
na.rm = TRUE)

attr.yld.ca.french.male <- rowSums(d.attr.yld.boot
[, grepl("^ca_", colnames(d.attr.yld.boot)) &
  grepl("french", colnames(d.attr.yld.boot)) &
  grepl("male", colnames(d.attr.yld.boot))],
na.rm = TRUE)

# German
attr.yld.ca.german.female <- rowSums(d.attr.yld.boot
[, grepl("^ca_", colnames(d.attr.yld.boot)) &
  grepl("german", colnames(d.attr.yld.boot)) &
  grepl("female", colnames(d.attr.yld.boot))],
```

```

na.rm = TRUE)

attr.yld.ca.german.male <- rowSums(d.attr.yld.boot
  [, grepl("^ca_", colnames(d.attr.yld.boot)) &
    grepl("german", colnames(d.attr.yld.boot)) &
    grepl("male", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

# Italian
attr.yld.ca.italian.female <- rowSums(d.attr.yld.boot
  [, grepl("^ca_", colnames(d.attr.yld.boot)) &
    grepl("italian", colnames(d.attr.yld.boot)) &
    grepl("female", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

attr.yld.ca.italian.male <- rowSums(d.attr.yld.boot
  [, grepl("^ca_", colnames(d.attr.yld.boot)) &
    grepl("italian", colnames(d.attr.yld.boot)) &
    grepl("male", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

# Rates per 100 000
yld.ca.french.female.boot <-
  (attr.yld.ca.french.female / pop.french.female) * 100000
yld.ca.french.male.boot <-
  (attr.yld.ca.french.male / pop.french.male) * 100000
yld.ca.german.female.boot <-
  (attr.yld.ca.german.female / pop.german.female) * 100000
yld.ca.german.male.boot <-
  (attr.yld.ca.german.male / pop.german.male) * 100000
yld.ca.italian.female.boot <-
  (attr.yld.ca.italian.female / pop.italian.female) * 100000
yld.ca.italian.male.boot <-
  (attr.yld.ca.italian.male / pop.italian.male) * 100000

get_stats <- function(x) {
  data.frame(
    mean_rate = mean(x, na.rm = TRUE),
    lower_rate = quantile(x, probs = 0.025, na.rm = TRUE),
    upper_rate = quantile(x, probs = 0.975, na.rm = TRUE))
}

d.rate.yld.ca.region.sex <- dplyr::bind_rows(
  get_stats(yld.ca.french.female.boot),
  get_stats(yld.ca.french.male.boot),

```

```

get_stats(yld.ca.german.female.boot),
get_stats(yld.ca.german.male.boot),
get_stats(yld.ca.italian.female.boot),
get_stats(yld.ca.italian.male.boot)) %>%

mutate(
  region = rep(c("French", "French", "German", "German",
                 "Italian", "Italian"), each = 1),
  sex = rep(c("Female", "Male"), times = 3),

  ci_rate = paste0(sprintf("%.2f", mean_rate), ", 95% CI [",
                        sprintf("%.2f", lower_rate), ", ",
                        sprintf("%.2f", upper_rate), "])") %>%

select(region, sex, mean_rate, lower_rate, upper_rate, ci_rate)

d.rate.yld.ca.region.sex <- d.rate.yld.ca.region.sex %>%
  rownames_to_column(var = "region_row") %>%
  select(-region_row)

```

Aggregation per region, sex and age

###YLD per region, sex and age

To generate PIA-attributable YLD estimates across detailed strata, a loop was constructed to iterate over sex, age group, and region. For each combination, relevant columns in the bootstrap dataset were selected and row-wise sums were computed to obtain total YLD vectors. Each result was stored as a separate object in the global environment, with age group labels sanitized for use in variable names (hyphens replaced with underscores).

```

# Dimensions
sexes <- unique(d.global$sex)
ages <- unique(d.global$age)
regions <- unique(d.global$language)

# Initialize result list
result_list_yld <- list()

# Loop to generate 72 vectors (sex × age × region)
for (s in sexes) {
  for (a in ages) {
    for (r in regions) {

```

```

# Select matching columns by region, age, sex using strsplit
sel <- sapply(names(d.attr.yld.boot), function(name) {
  parts <- strsplit(name, "\\\\.")[1]
  length(parts) == 4 &&
  parts[2] == r &&
  parts[3] == a &&
  parts[4] == s
})

# Compute row-wise sum (YLD per bootstrap iteration)
vec <- rowSums(d.attr.yld.boot[, sel, drop = FALSE],
              na.rm = TRUE)

# results (mean, CI)
res <- data.frame(
  sex    = s,
  age    = a,
  region = r,
  mean   = mean(vec, na.rm = TRUE),
  lower  = quantile(vec, probs = 0.025, na.rm = TRUE),
  upper  = quantile(vec, probs = 0.975, na.rm = TRUE),
  stringsAsFactors = FALSE
)

result_list_yld[[paste(s, a, r, sep = "_")] <- res
}
}
}

d.ci.yld <- bind_rows(result_list_yld)

```

Rates YLD To calculate YLD rates per 100 000 population, population data were first prepared by harmonizing variable names and recoding sex categories. The YLD bootstrap summary data (d.ci.yld) were then joined with the corresponding population strata (pop_2022) by sex, age group, and region. YLD rates were computed by dividing the mean and confidence interval estimates by the relevant population counts and scaling to 100 000. A formatted column displaying the mean and 95% CI was created for use in tables and visualizations.

Check values with static calculation

```

# select vector
attr.yld.40_44.french.female <- rowSums(
  d.attr.yld.boot[, grepl("french", colnames(d.attr.yld.boot)) &

```

```

      grepl("female", colnames(d.attr.yld.boot)) &
      grepl("40-44", colnames(d.attr.yld.boot))],
  na.rm = TRUE)

#population
pop_40_44_french_female <- d.population_region %>%
  filter(sex == "female", language == "french",
         age_group == "40-44") %>%
  pull(pop_region)

# YLL rate per 100 000
yld.rate.40_44.french.female.boot <-
  attr.yld.40_44.french.female / pop_40_44_french_female * 100000

# Rates Mean and 95% CI for CHD YLL
mean.attr.yld.40_44.french.female <-
  mean(yld.rate.40_44.french.female.boot)
lower.attr.yld.40_44.french.female <-
  quantile(yld.rate.40_44.french.female.boot, probs = 0.025)
upper.attr.yld.40_44.french.female <-
  quantile(yld.rate.40_44.french.female.boot, probs = 0.975)

#check if results of d.rate.yll.disease (generated by loop)
#is matching the individual calculation
mean.attr.yld.40_44.french.female
d.ci.yld.rate %>%
  filter(sex == "female",
         region == "french",
         age == "40-44")

```

To present YLD rates stratified by age group, sex, and region, the dataset `d.ci.yld.rate` was filtered separately for females and males. For each sex, the data were reshaped into a wide format, with age groups as columns and regions as rows. The resulting tables (`d.rate.yld.wide.female` and `d.rate.yld.wide.male`) contain formatted mean YLD rates (with 95% confidence intervals) per 100 000 population, facilitating comparison across demographic strata.

YLL per age group, region and sex

To generate YLL estimates stratified by sex, age group, and region, a triple loop was implemented. For each combination, relevant bootstrap columns were selected using pattern matching on column names. The values were then summed row-wise to obtain total YLLs per bootstrap

sample. Each resulting vector was assigned to a uniquely named object reflecting the respective subgroup.

```
# Initialize result list
result_list_yll <- list()

# Dimensions
sexes <- unique(d.global$sex)
ages <- unique(d.global$age)
regions <- unique(d.global$language)

# Loop to generate 72 vectors (sex × age × region) for YLL
for (s in sexes) {
  for (a in ages) {
    for (r in regions) {

      # Identify matching columns using structured split
      sel <- sapply(names(d.attr.yll.boot), function(n) {
        parts <- strsplit(n, "\\\\.")[1]
        length(parts) == 4 &&
        parts[2] == r &&
        parts[3] == a &&
        parts[4] == s
      })

      # Compute row-wise sum (YLL per bootstrap)
      vec <- rowSums(d.attr.yll.boot[, sel, drop = FALSE],
                    na.rm = TRUE)

      # mean & ci
      res <- data.frame(
        sex = s,
        age = a,
        region = r,
        mean = mean(vec, na.rm = TRUE),
        lower = quantile(vec, probs = 0.025, na.rm = TRUE),
        upper = quantile(vec, probs = 0.975, na.rm = TRUE),
        stringsAsFactors = FALSE)

      result_list_yll[[paste(s, a, r, sep = "_")] <- res
    }
  }
}

# create DataFrame
d.ci.yll <- bind_rows(result_list_yll)
```

```

# population: prepare and rename
pop_region <- d.population_region %>%
  rename(
    age = age_group,
    region = language) %>%
  select(sex, age, region, pop_region)

# rates (per 100 000)
d.ci.yll.rate <- d.ci.yll %>%
  left_join(pop_region, by = c("sex", "age", "region")) %>%
  mutate(
    mean = (mean / pop_region) * 1e5,
    lower = (lower / pop_region) * 1e5,
    upper = (upper / pop_region) * 1e5,
    mean_ci = paste0(sprintf("%.2f", mean), " [",
                      sprintf("%.2f", lower), " ",
                      sprintf("%.2f", upper), "]"))

# Select all relevant bootstrap columns (ages 40-44, French region, female)
attr.yll.40_44.french.female <- rowSums(
  d.attr.yll.boot[, grepl("french", colnames(d.attr.yll.boot)) &
                    grepl("female", colnames(d.attr.yll.boot)) &
                    grepl("40-44", colnames(d.attr.yll.boot))],
  na.rm = TRUE)

# Retrieve the corresponding population (specific to region,
#sex, and age group)
pop_40_44_french_female <- d.population_region %>%
  filter(sex == "female", language == "french",
         age_group == "40-44") %>%
  pull(pop_region)

# Calculate the YLL rate per 100 000 population
yll.rate.40_44.french.female.boot <- attr.yll.40_44.french.female / pop_40_44_french_female

# Calculate summary statistics
mean.attr.yll.40_44.french.female <-
  mean(yll.rate.40_44.french.female.boot)
lower.attr.yll.40_44.french.female <-
  quantile(yll.rate.40_44.french.female.boot, probs = 0.025)
upper.attr.yll.40_44.french.female <-
  quantile(yll.rate.40_44.french.female.boot, probs = 0.975)

```

```
# Output the manually calculated values
mean.attr.yll.40_44.french.female
lower.attr.yll.40_44.french.female
upper.attr.yll.40_44.french.female

# check with df
d.ci.yll.rate %>%
  filter(sex == "female",
         region == "french",
         age == "40-44")
```

To facilitate age-specific comparisons of PIA-attributable YLL rates by sex and region, the dataset was filtered separately for females and males. For each subgroup, the data were transformed into wide format with age groups as columns and regions as rows, displaying mean YLL rates with 95% confidence intervals.

```
# YLL - females
d.rate.yll.wide.female <- d.ci.yll.rate %>%
  filter(sex == "female") %>%
  select(region, age, mean_ci) %>%
  pivot_wider(
    names_from = age,
    values_from = mean_ci)

# YLL - males
d.rate.yll.wide.male <- d.ci.yll.rate %>%
  filter(sex == "male") %>%
  select(region, age, mean_ci) %>%
  pivot_wider(
    names_from = age,
    values_from = mean_ci)
```

```
#export data
write.xlsx(d.riskratios, file = "../tables/d.riskratios.xlsx")
write.xlsx(d.prev_pia, file = "../tables/d.prev_pia.xlsx")
write.xlsx(d.global, file = "../tables/d.global.xlsx")
write.xlsx(d.deaths, file = "../tables/d.deaths.xlsx")

write.xlsx(d.total.ch, file = "../tables/d.total.ch.xlsx")

write.xlsx(d.rate.yld.sex, file = "../tables/d.rate.yld.sex.xlsx")
write.xlsx(d.rate.yll.sex, file = "../tables/d.rate.yll.sex.xlsx")

write.xlsx(d.yll.region, file = "../tables/d.yll.region.xlsx")
```

```

write.xlsx(d.yld.region, file = "../tables/d.yld.region.xlsx")

write.xlsx(d.rate.yld.disease.total, file =
  "../tables/d.rate.yld.disease.total.xlsx")
write.xlsx(d.rate.yll.disease.total, file =
  "../tables/d.rate.yll.disease.total.xlsx")

write.xlsx(d.rate.yll.region.sex, file =
  "../tables/d.rate.yll.region.sex.xlsx")
write.xlsx(d.rate.yld.region.sex, file =
  "../tables/d.rate.yld.region.sex.xlsx")

write.xlsx(d.rate.yld.disease, file =
  "../tables/d.rate.yld.disease.xlsx")
write.xlsx(d.rate.yll.disease, file =
  "../tables/d.rate.yll.disease.xlsx")

```

Session Info

```
utils::sessionInfo()
```

R version 4.5.3 (2026-03-11)

Platform: aarch64-apple-darwin20

Running under: macOS Tahoe 26.6.2

Matrix products: default

BLAS: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib

LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib;

locale:

[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Zurich

tzcode source: internal

attached base packages:

[1] grid stats graphics grDevices utils datasets methods
[8] base

other attached packages:

```

[1] survey_4.4-8      survival_3.8-6    Matrix_1.7-4     devtools_2.5.2
[5] usethis_3.2.1     gridExtra_2.3.1  patchwork_1.3.2  gt_1.3.0
[9] tibble_3.3.1      tinytex_0.60     rtticles_0.27    scales_1.4.0
[13] openxlsx_4.2.8.1  tidyr_1.3.2      stringr_1.6.0    ggplot2_4.0.3
[17] kableExtra_1.4.0  readr_2.2.0      readxl_1.5.0     dplyr_1.2.1

```

loaded via a namespace (and not attached):

```

[1] gtable_0.3.6      xfun_0.53         lattice_0.22-9    tzdb_0.5.0
[5] vctrs_0.7.2       tools_4.5.3       generics_0.1.4    parallel_4.5.3
[9] pkgconfig_2.0.3   RColorBrewer_1.1-3 S7_0.2.0          lifecycle_1.0.5
[13] compiler_4.5.3    farver_2.1.2      textshaping_1.0.4 mitools_2.4
[17] htmltools_0.5.8.1 yaml_2.3.12       crayon_1.5.3      pillar_1.11.1
[21] ellipsis_0.3.3    cachem_1.1.0      sessioninfo_1.2.3 tidyselect_1.2.1
[25] zip_2.3.3         digest_0.6.37     stringi_1.8.7     purrr_1.2.2
[29] labeling_0.4.3    splines_4.5.3     fastmap_1.2.0     cli_3.6.6
[33] magrittr_2.0.4    pkgbuild_1.4.8    withr_3.0.2       bit64_4.6.0-
1
[37] rmarkdown_2.31    bit_4.6.0         otel_0.2.0        cellranger_1.1.0
[41] hms_1.1.4         memoise_2.0.1     evaluate_1.0.5    knitr_1.51
[45] viridisLite_0.4.2 rlang_1.2.0       Rcpp_1.1.0        DBI_1.2.3
[49] glue_1.8.0        xml2_1.6.0        pkgload_1.5.3     vroom_1.7.1
[53] svglite_2.2.2     rstudioapi_0.17.1 jsonlite_2.0.0    R6_2.6.1
[57] systemfonts_1.3.1 fs_2.1.0

```