Spatial Analytics

Villiam Molte Kristiane Warncke

04-06-2025

setup chunk

importing packages

```
library(tidyverse)
library(ggrepel)
library(geodata)
library(sf)
library(geosphere)
library(cartogram)
library(readxl)
library(magick)
library(viridis)
library(ggtext)
```

loading in data

```
#listing female files files
file_paths <- list.files("new_data", pattern = "^f_\\d+-\\d+\\.csv$", full.names = TRUE)

#reading and cleaning each file
data_list <- lapply(file_paths, function(path) {
  df <- read_csv2(path, locale = locale(encoding = "latin1"))

  # Get x and y from filename
  x_y <- str_match(basename(path), "f_(\\d+)-(\\d+)\\.csv")[,2:3]
  colnames(df)[5:14] <- as.character(seq(as.integer(x_y[1]), as.integer(x_y[2])))

  # Rename and clean 'orig' and 'dest' columns
  df <- df %>%
    mutate(
    orig = sub("^Fra ", "", orig),
    dest = sub("^Til ", "", dest)
    )

  return(df)
})
```

```r
#creating female df
combined_df_female <- data_list[[1]]

#binding only the numeric columns (last 10) from the rest
for (i in 2:length(data_list)) {
  numeric_cols <- data_list[[i]] %>% select((ncol(.) - 9):ncol(.))
  combined_df_female<- bind_cols(combined_df_female, numeric_cols)
}

### doing the same for the male data

# listing all male files
file_paths <- list.files("new_data", pattern = "^m_\\d+-\\d+\\.csv$", full.names = TRUE)

#reading and cleaning each file
data_list <- lapply(file_paths, function(path) {
  df <- read_csv2(path, locale = locale(encoding = "latin1"))

  #extracting age
  x_y <- str_match(basename(path), "m_(\\d+)-(\\d+)\\.csv")[,2:3]
  colnames(df)[5:14] <- as.character(seq(as.integer(x_y[1]), as.integer(x_y[2])))

  #renaming and cleaning 'orig' and 'dest' columns
  df <- df %>%
    mutate(
      orig = sub("^Fra ", "", orig),
      dest = sub("^Til ", "", dest)
    )

  return(df)
})

#saving male
combined_df_male <- data_list[[1]]

# Bind only the numeric columns (last 10) from the rest
for (i in 2:length(data_list)) {
  numeric_cols <- data_list[[i]] %>% select((ncol(.) - 9):ncol(.))
  combined_df_male  <- bind_cols(combined_df_male, numeric_cols)
}

#creating gender columns
combined_df_male <- combined_df_male %>% mutate(sex = 1)
combined_df_female <- combined_df_female %>% mutate(sex = 0)
names(combined_df_female) <- names(combined_df_male)  # ensure columns match

combined_df <- rbind(combined_df_male, combined_df_female)

#combining the two
df <- combined_df %>% select(-'Mænd' & -'2024')


#### note: this process was necessary as statistikbanken has a limit to how much data can be downloaded
```

cleaning data

```
# Normalize city names: convert "Århus" to "Aarhus"
df$orig <- recode(df$orig, "Århus" = "Aarhus")
df$dest <- recode(df$dest, "Århus" = "Aarhus")


# adding total column
df <- df %>%
  mutate(total = rowSums(select(., matches("^\\d+$"))))
```

loading in municipalities and transforming to sf

```
munic <- gadm(country = "DNK", level = 2, path = ".")

# Convert to sf
munic_sf <- st_as_sf(munic)

# Get centroids and coordinates
centroids <- st_centroid(munic_sf)
```

extract coordinates and attach muni name from NAME_2; calculating centroids and extract coordinates safely

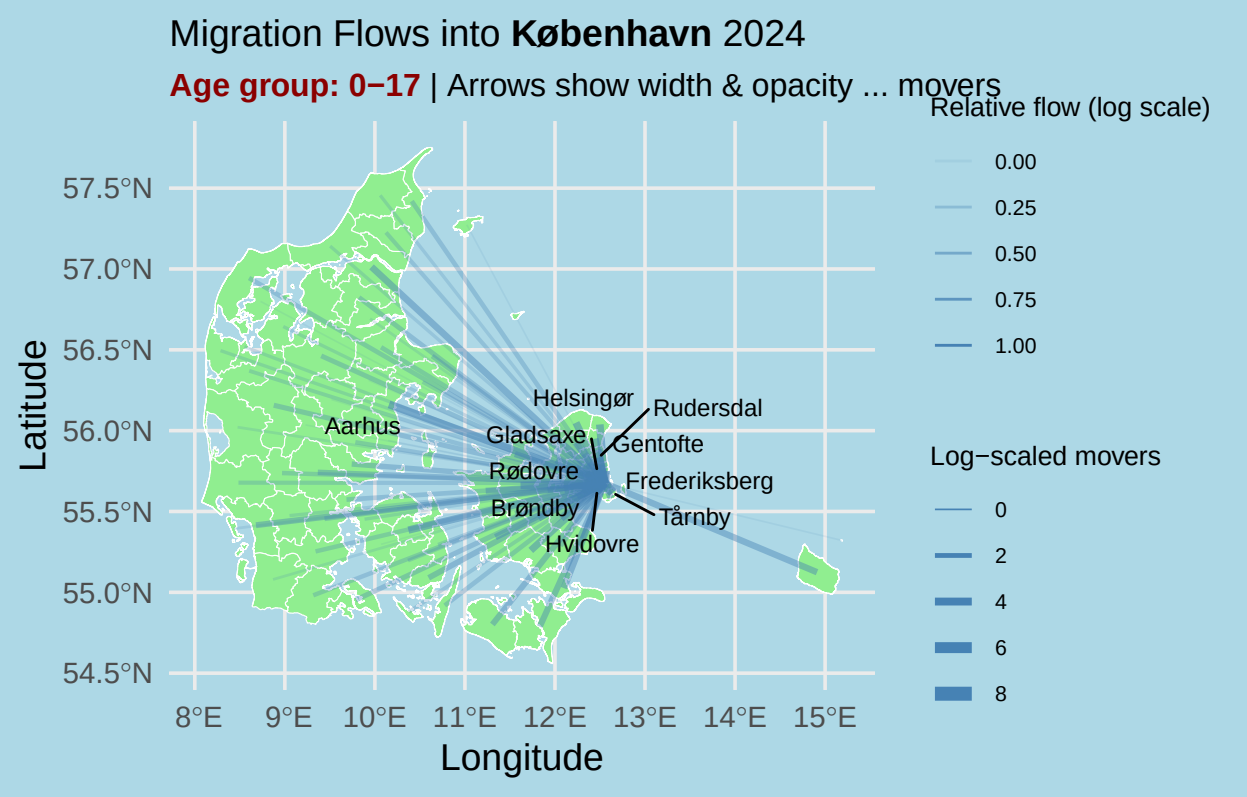
```
centroids_coords <- munic_sf %>%
  st_centroid() %>%
  mutate(
    lon = st_coordinates(.)[, 1],
    lat = st_coordinates(.)[, 2],
    muni = NAME_2  # explicitly create a 'muni' column
  ) %>%
  st_drop_geometry()  # drop geometry so it's a plain tibble

centroids_coords$muni <- recode(centroids_coords$muni, "Århus" = "Aarhus")
```

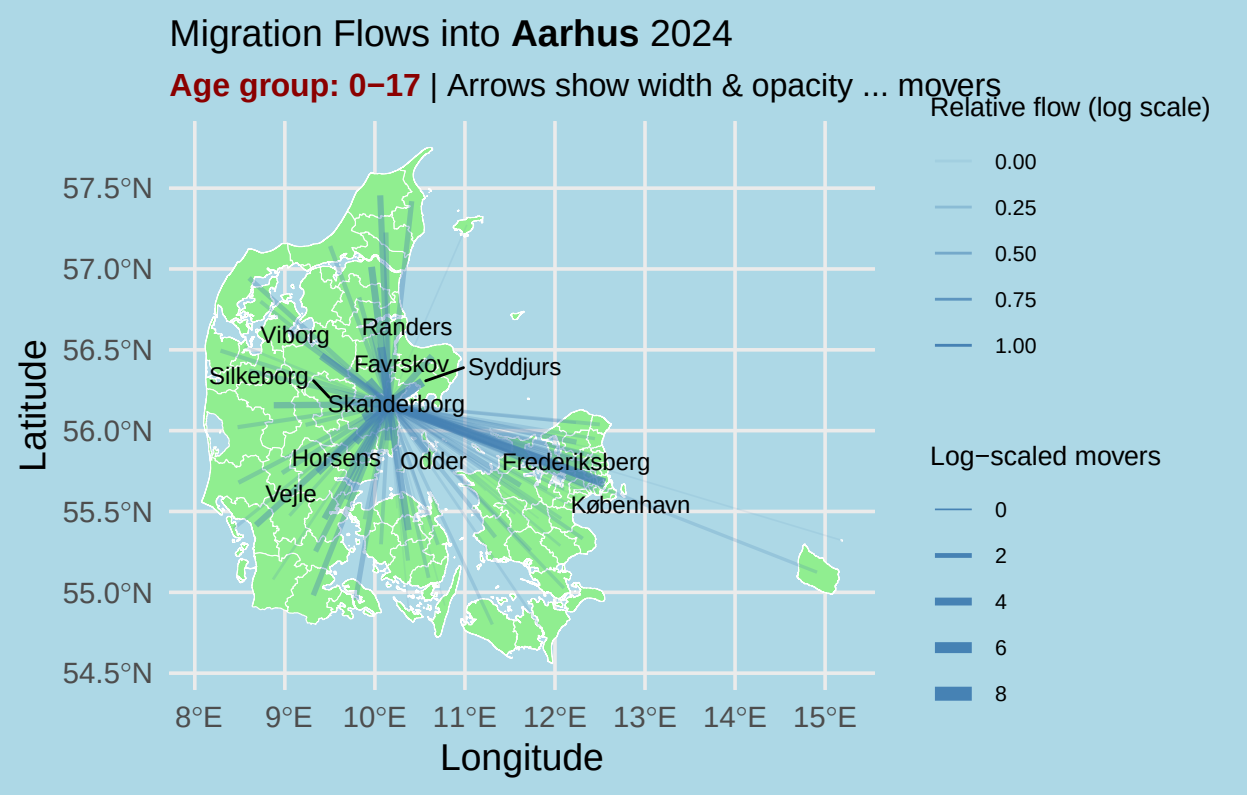
plotting flow line maps

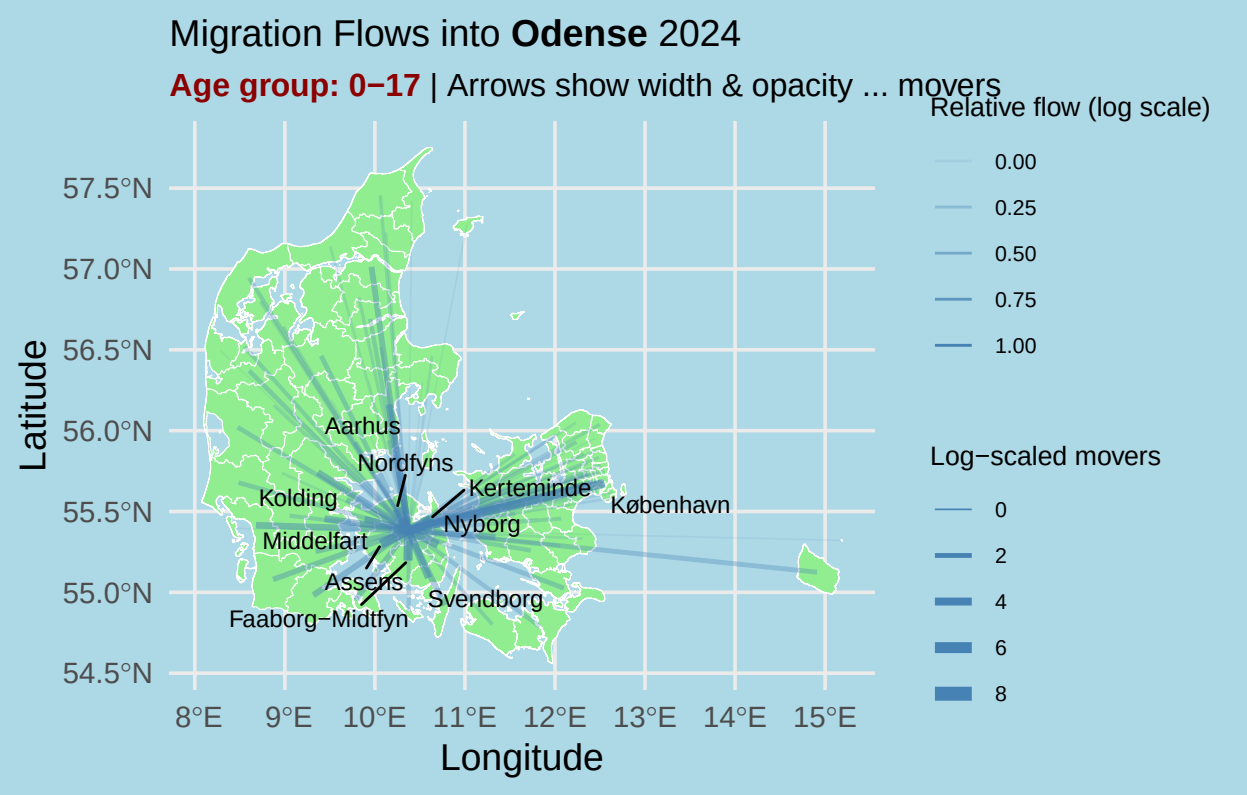
```
#defining age groups
age_groups <- list(
  "0-17"   = 0:17,
  "18-29"  = 18:29,
  "30-44"  = 30:44,
  "45-65"  = 45:65,
  "66-99"  = 66:99
```

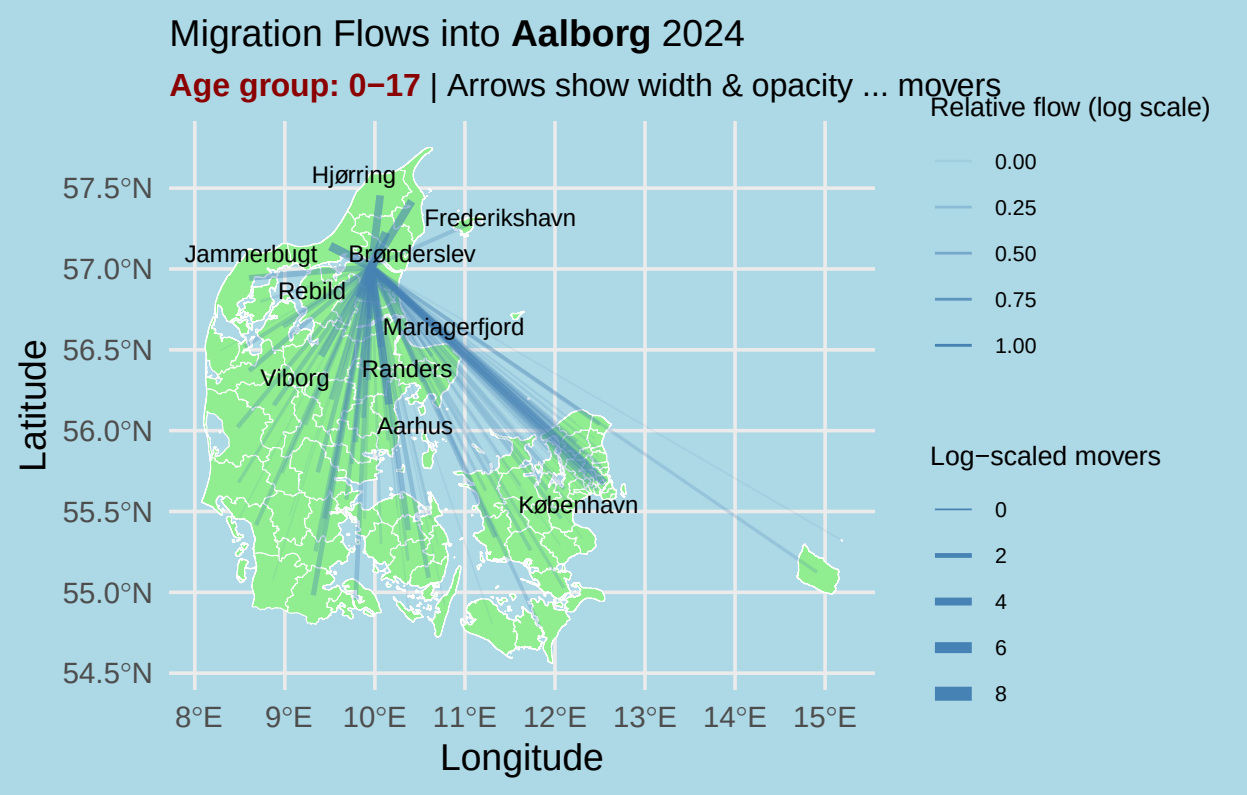
```
)

#defining top municipalities
top5_cities <- c("København", "Aarhus", "Odense", "Aalborg")

# calculating global max movers for consistent alpha scaling across all figures
all_age_cols <- paste0(0:99)
df$total_movers <- rowSums(df[all_age_cols], na.rm = TRUE)
global_max_movers_log <- log1p(max(df$total_movers, na.rm = TRUE))

# looping through age group
for (age_label in names(age_groups)) {
  age_range <- age_groups[[age_label]]
  age_cols <- paste0(age_range)

  # aggregating by age group
  df_age <- df %>%
    mutate(total = rowSums(select(., all_of(age_cols)), na.rm = TRUE))

  # for each municipality
  for (city in top5_cities) {

    flow_city <- df_age %>%
      filter(dest == city, orig != dest) %>%
      group_by(orig, dest) %>%
      summarise(movers = sum(total), .groups = "drop") %>%
      left_join(centroids_coords, by = c("orig" = "muni")) %>%
      rename(lon_o = lon, lat_o = lat) %>%
      left_join(centroids_coords, by = c("dest" = "muni")) %>%
      rename(lon_d = lon, lat_d = lat) %>%
      filter(!is.na(lon_o), !is.na(lon_d)) %>%
      mutate(
        movers_log = log1p(movers),
        alpha_scaled = movers_log / global_max_movers_log
      )

    # select top origin municipalities for labelling
    top_origins <- flow_city %>%
      slice_max(order_by = movers, n = 10)

    # plotting
    p <- ggplot() +
      geom_sf(data = munic_sf, fill = "lightgreen", color = "white", size = 0.2) +
      geom_segment(
        data = flow_city,
        aes(x = lon_o, y = lat_o, xend = lon_d, yend = lat_d,
            linewidth = movers_log, alpha = alpha_scaled),
        color = "steelblue"
      ) +
      geom_text_repel(
        data = top_origins,
        aes(x = lon_o, y = lat_o, label = orig),
        size = 3,
```

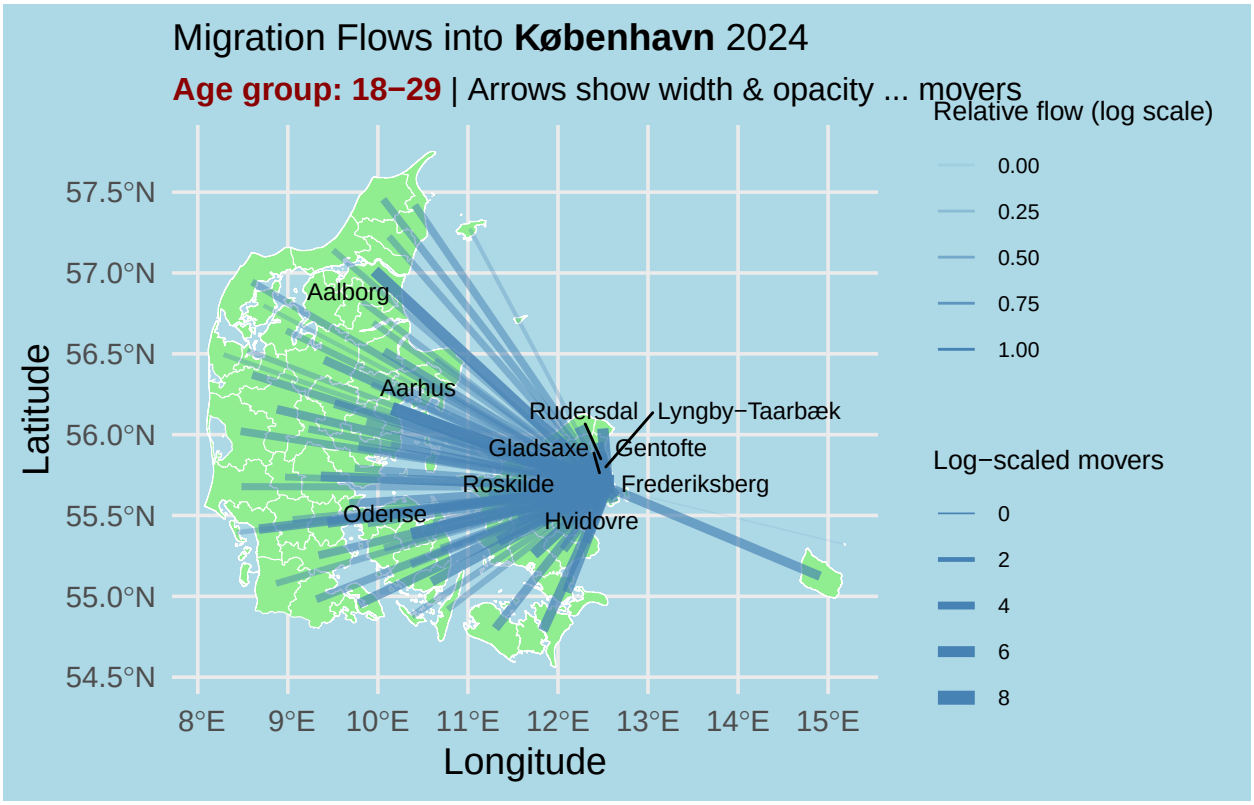
```
        color = "black",
        max.overlaps = Inf
      ) +
      scale_linewidth_continuous(
        range = c(0.3, 2.5),  # Adjust thickness as needed
        limits = c(0, global_max_movers_log),
        name = "Log-scaled movers"
      ) +
      scale_alpha_continuous(
        name = "Relative flow (log scale)",
        limits = c(0, 1)
      ) +
      guides(
        alpha = guide_legend(order = 1),
        linewidth = guide_legend(order = 2)
      ) +
      theme_minimal(base_size = 14) +
      theme(
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        panel.background = element_rect(fill = "lightblue", color = NA),
        plot.background = element_rect(fill = "lightblue", color = NA),
        plot.subtitle = element_markdown(size = 12),
        plot.title = element_markdown(size = 14)
      ) +
      labs(
        title = paste0("Migration Flows into <b>", city, "</b> 2024"),
        subtitle = paste0(
          "<span style='color:darkred;'><b>Age group: ", age_label,
          "</b></span> | Arrows show width & opacity  movers"
        ),
        x = 'Longitude',
        y = 'Latitude'
      )

    #saving the plot
    filename <- paste0("out/migration_flows_", city, "_age_", gsub("-", "_", age_label), ".png")
    ggsave(filename = filename, plot = p, width = 8, height = 6, dpi = 300)

    print(p)
  }
}
```

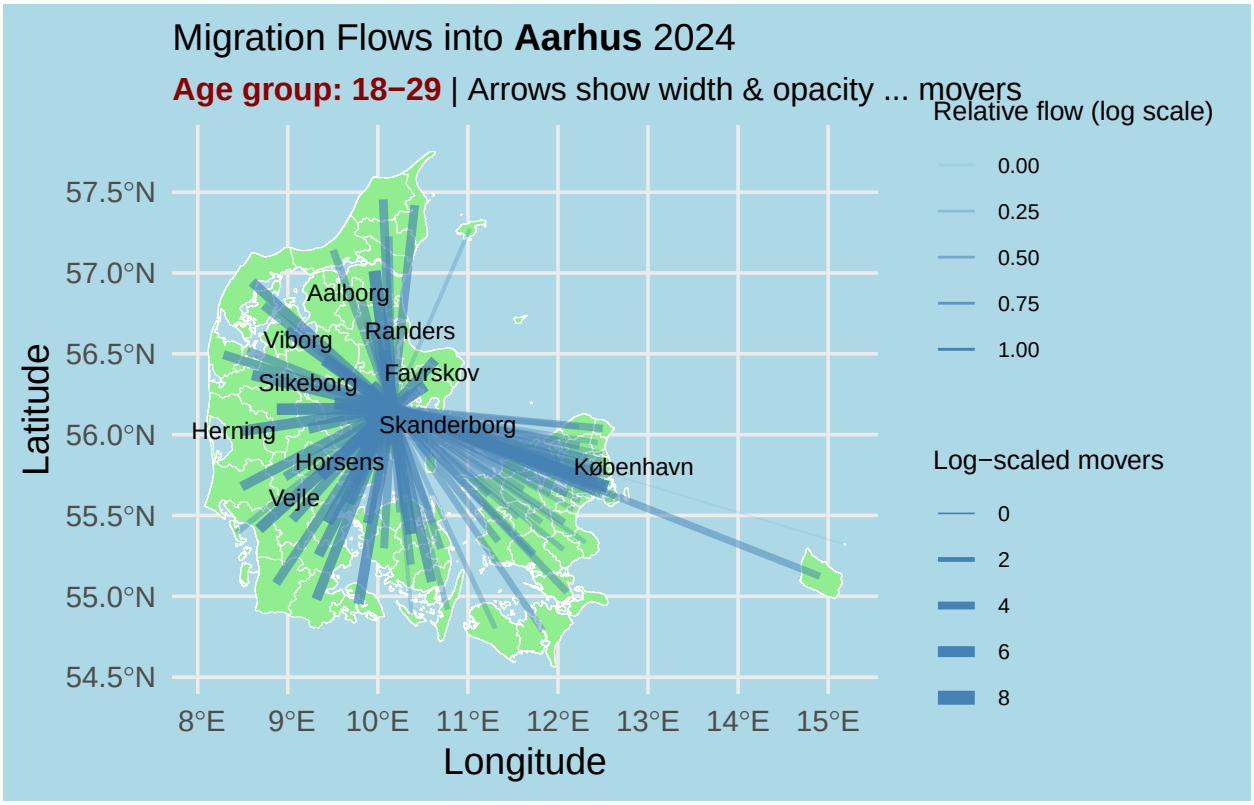

Migration Flows into **København** 2024
Age group: 0–17 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aarhus
Helsingør
Rudersdal
Gladsaxe
Gentofte
Rødovre
Frederiksberg
Brøndby
Tårnby
Hvidovre

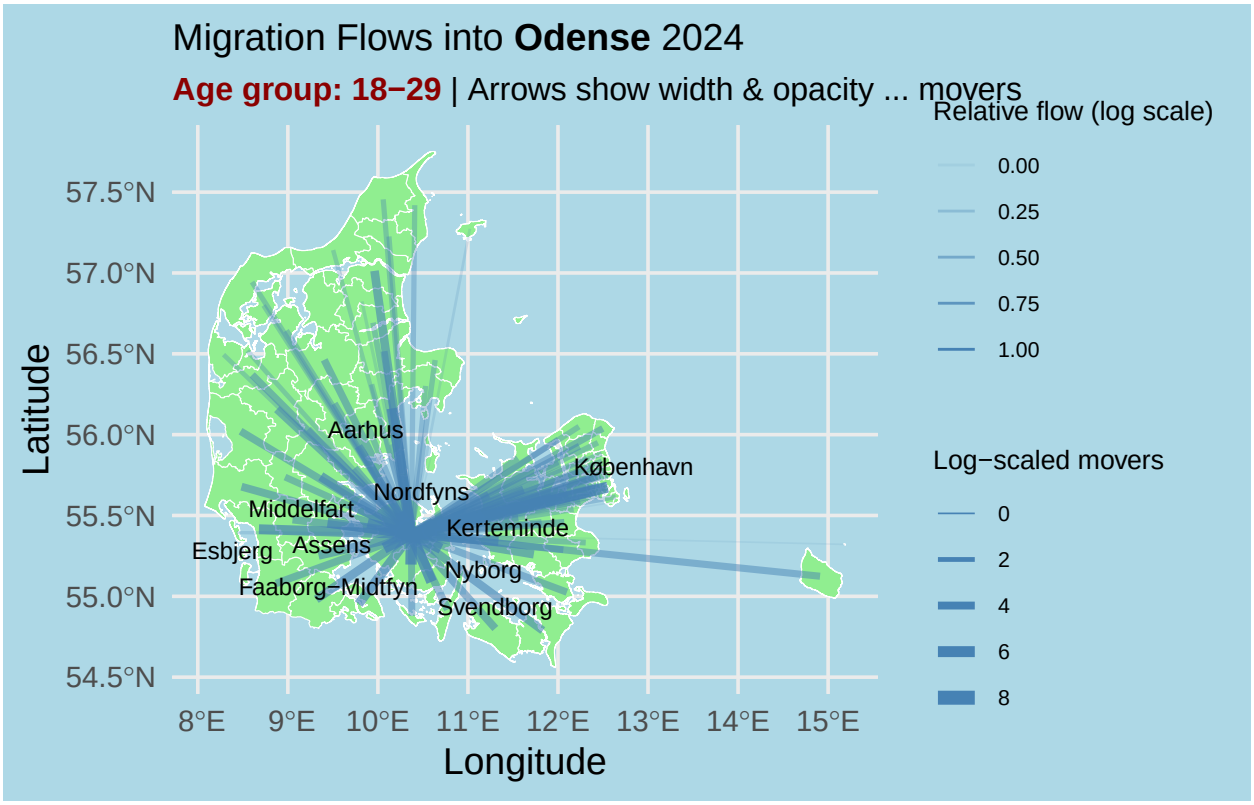

Migration Flows into **Aarhus** 2024
Age group: 0–17 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Viborg
Randers
Favrskov
Syddjurs
Silkeborg
Skanderborg
Horsens
Odder
Frederiksberg
Vejle
København

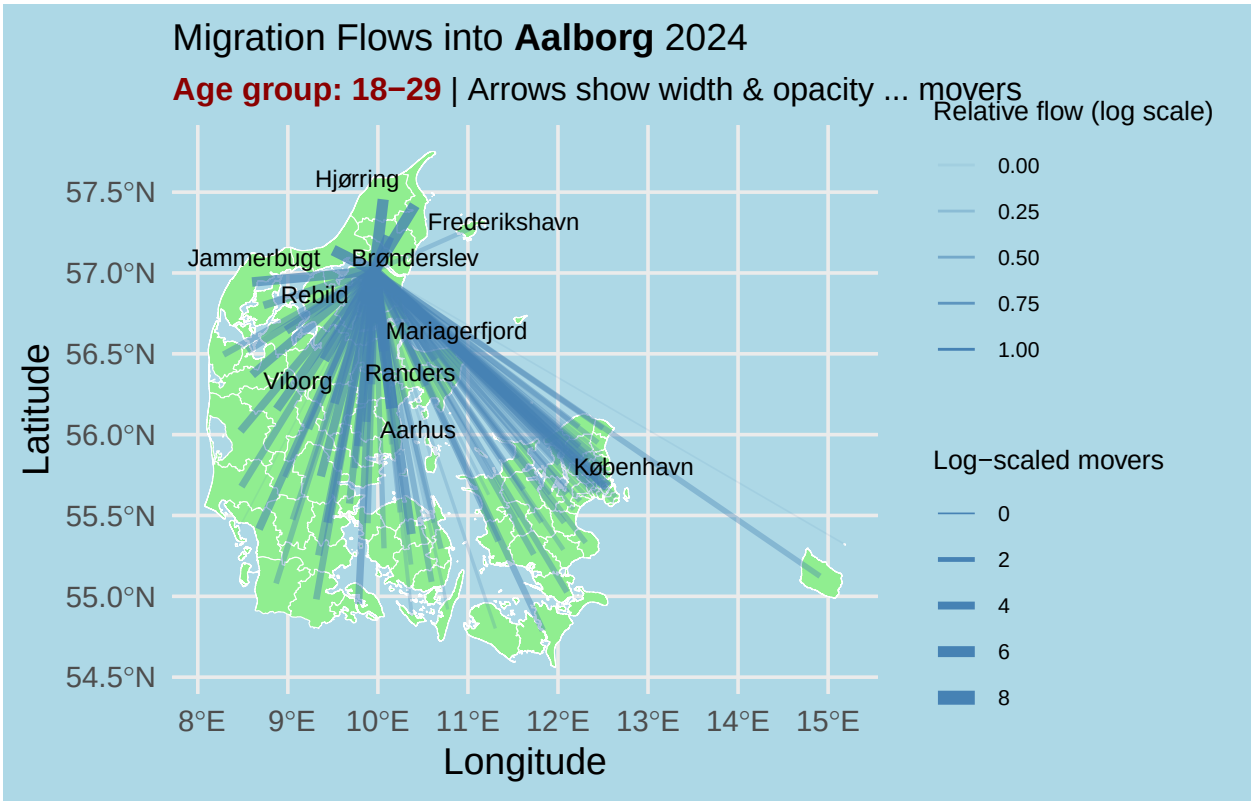

Migration Flows into **Odense** 2024
Age group: 0–17 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aarhus
Nordfyns
Kerteminde
Kolding
Nyborg
København
Middelfart
Assens
Svendborg
Faaborg−Midtfyn

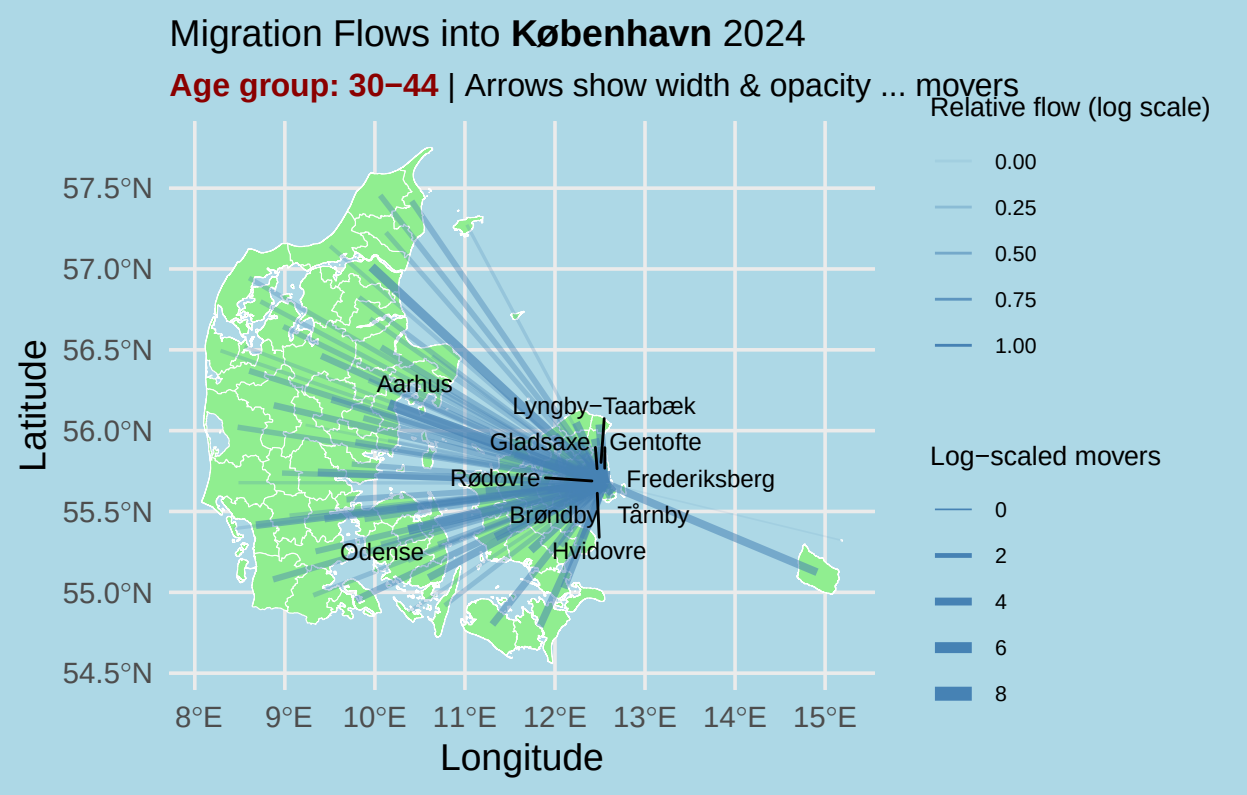

Migration Flows into **Aalborg** 2024
Age group: 0–17 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Hjørring
Frederikshavn
Jammerbugt
Brønderslev
Rebild
Mariagerfjord
Viborg
Randers
Aarhus
København
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude

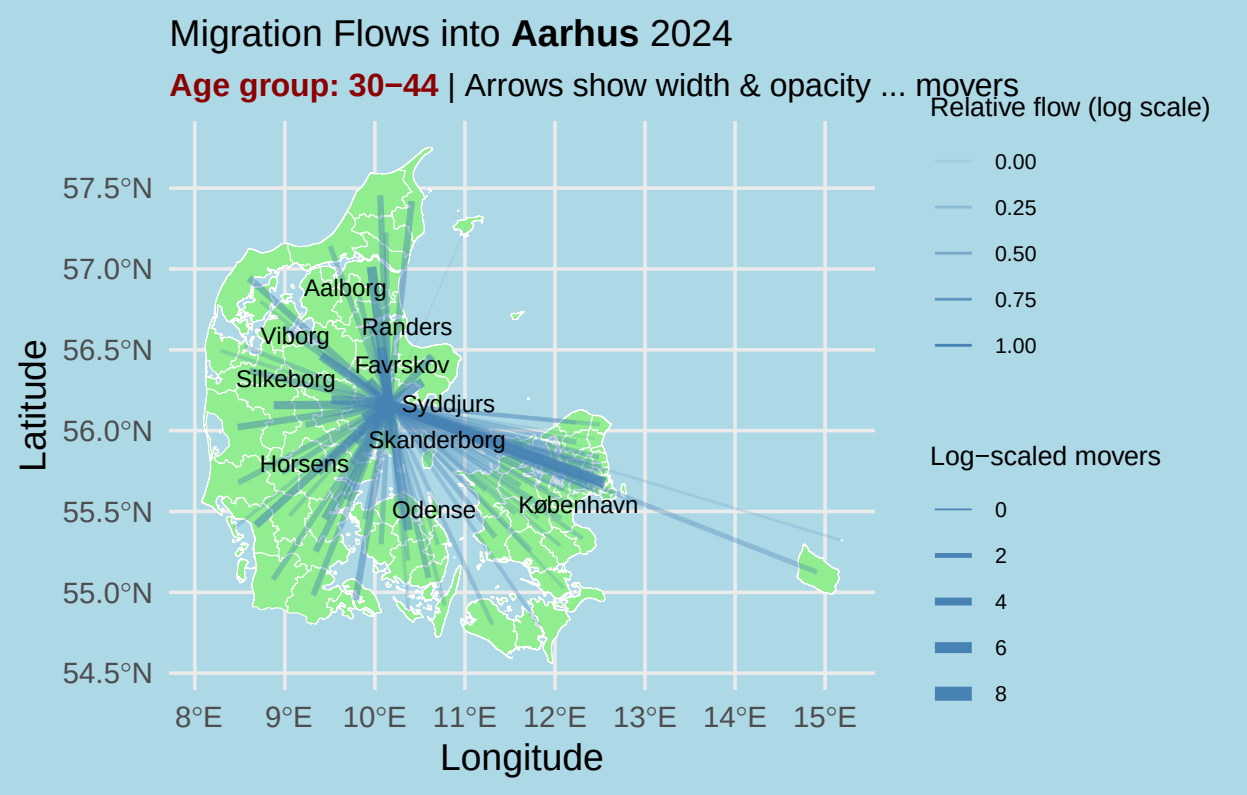

Migration Flows into **København** 2024
Age group: 18–29 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aalborg
Aarhus
Rudersdal
Lyngby−Taarbæk
Gladsaxe
Gentofte
Roskilde
Frederiksberg
Odense
Hvidovre

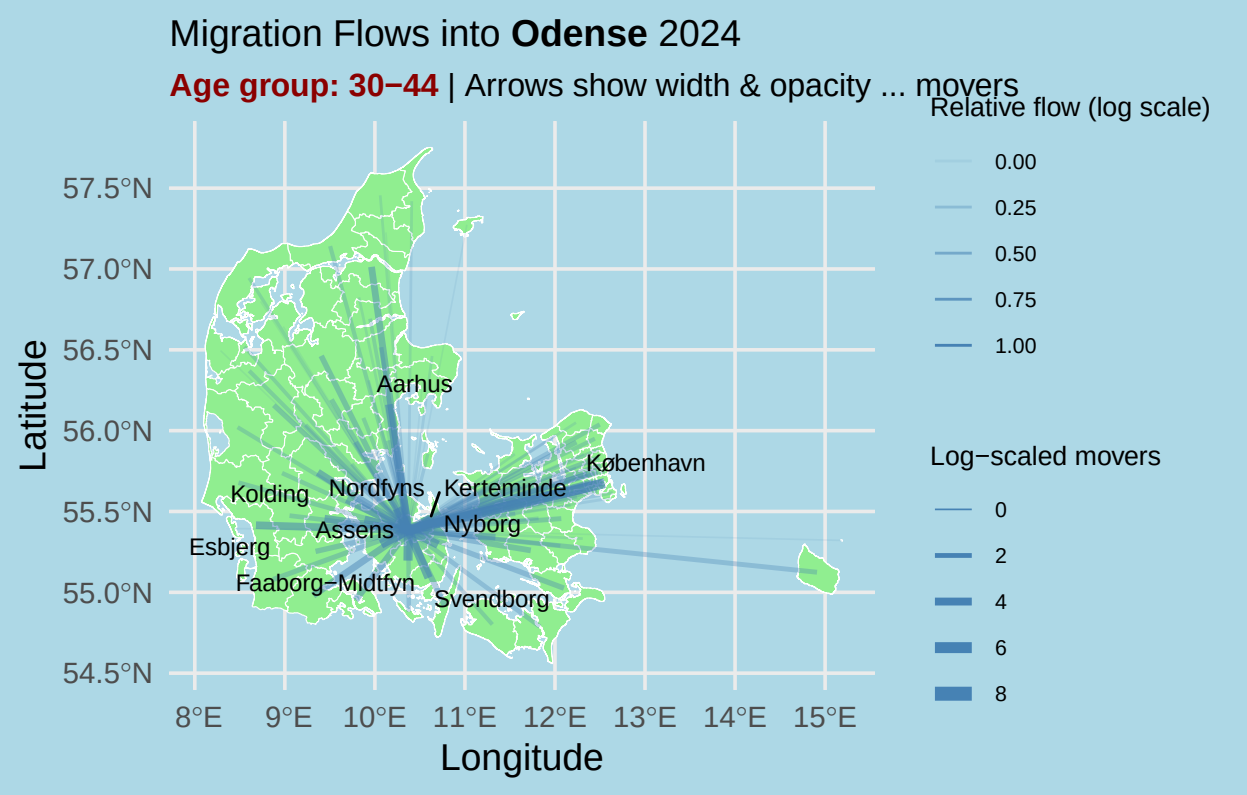

Migration Flows into **Aarhus** 2024
Age group: 18–29 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aalborg
Randers
Viborg
Favrskov
Silkeborg
Skanderborg
Herning
Horsens
København
Vejle

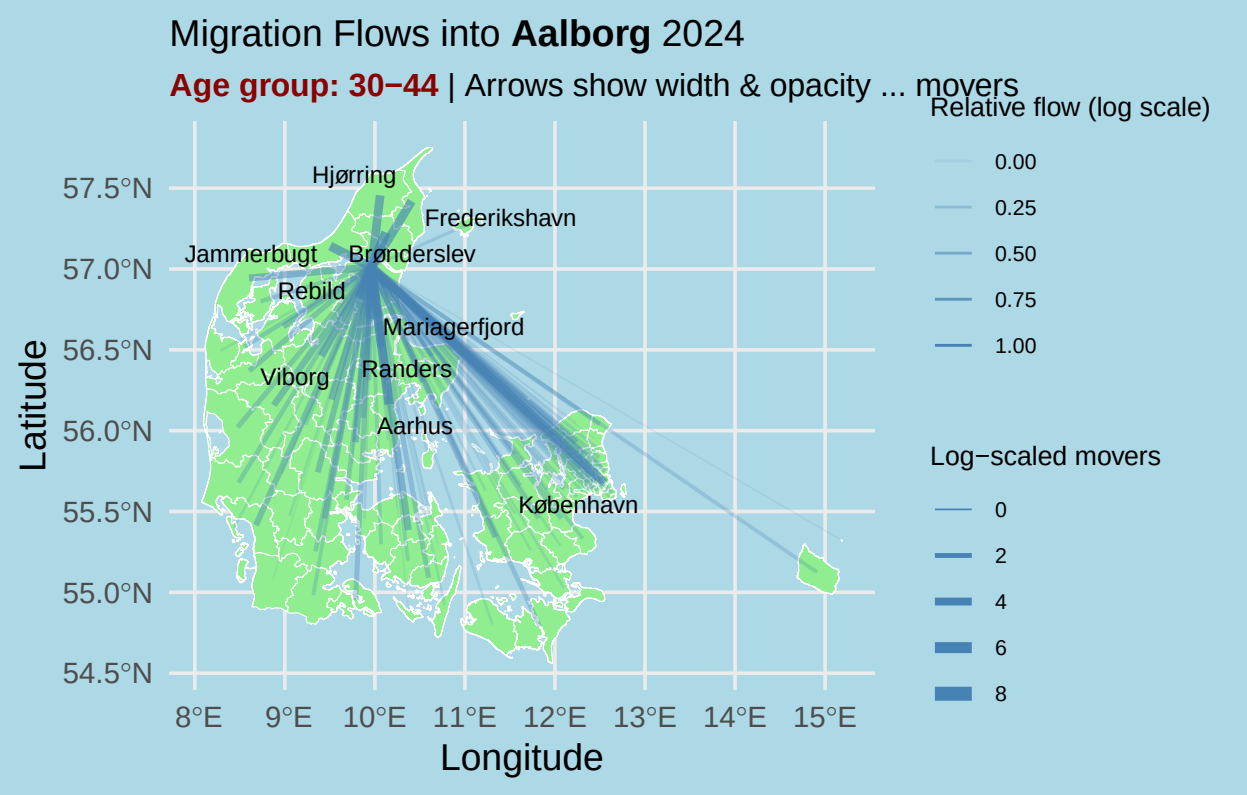

Migration Flows into **Odense** 2024
Age group: 18–29 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aarhus
København
Nordfyns
Middelfart
Kerteminde
Esbjerg
Assens
Nyborg
Faaborg−Midtfyn
Svendborg

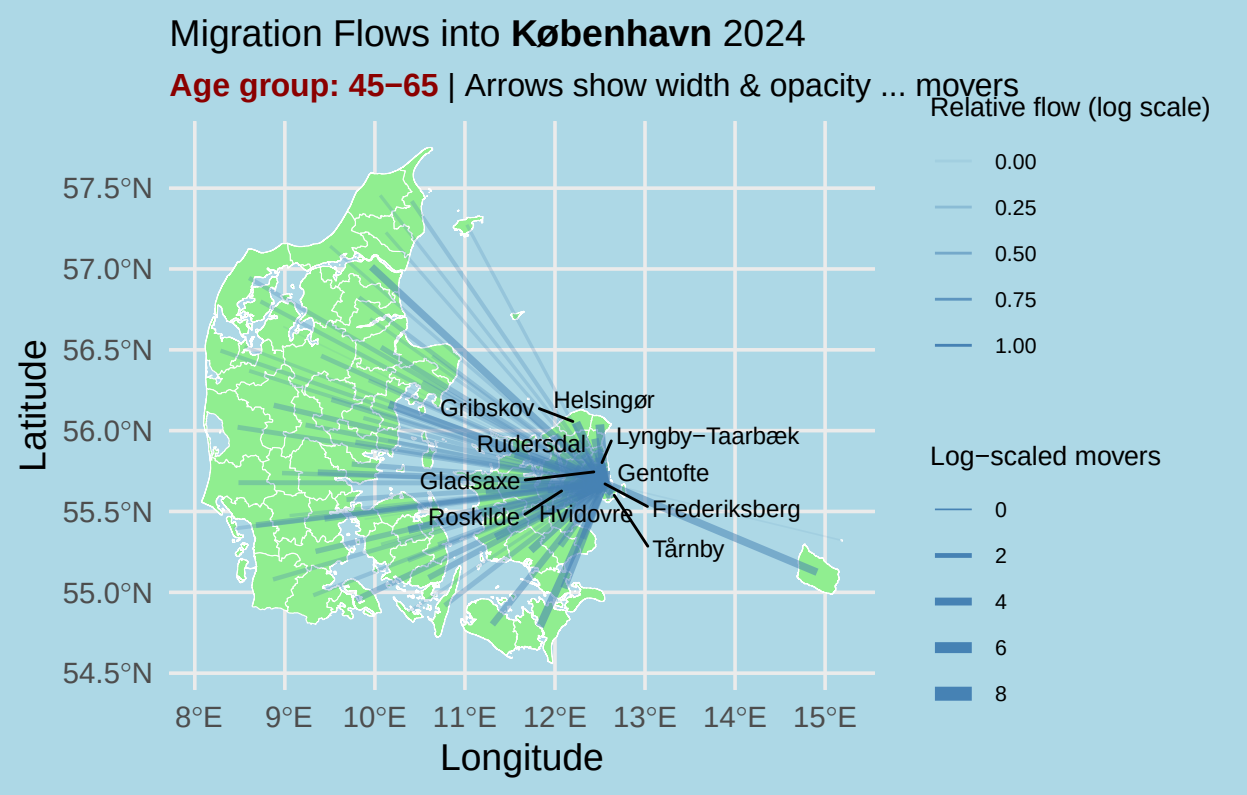

Migration Flows into **Aalborg** 2024
Age group: 18–29 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Hjørring
Frederikshavn
Jammerbugt
Brønderslev
Rebild
Mariagerfjord
Viborg
Randers
Aarhus
København
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude

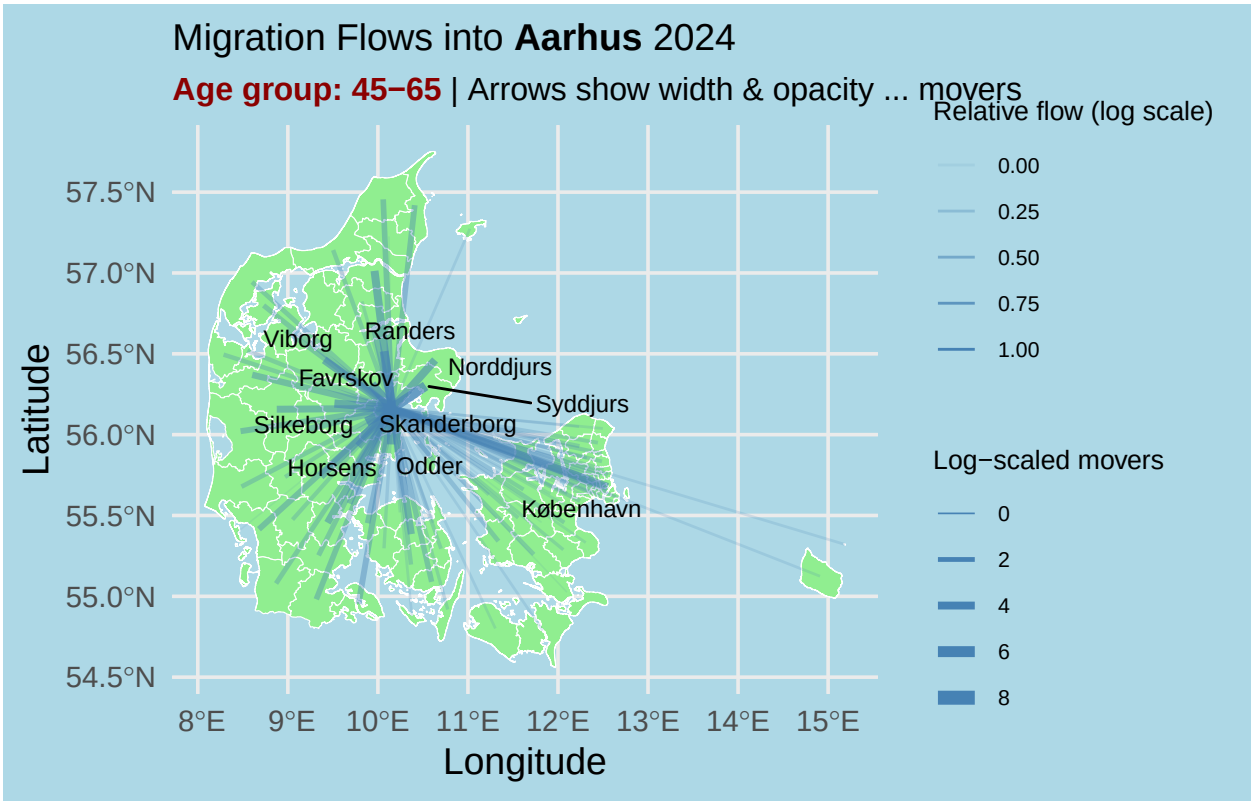

Migration Flows into **København** 2024
Age group: 30–44 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aarhus
Lyngby−Taarbæk
Gladsaxe
Gentofte
Rødovre
Frederiksberg
Brøndby
Tårnby
Odense
Hvidovre

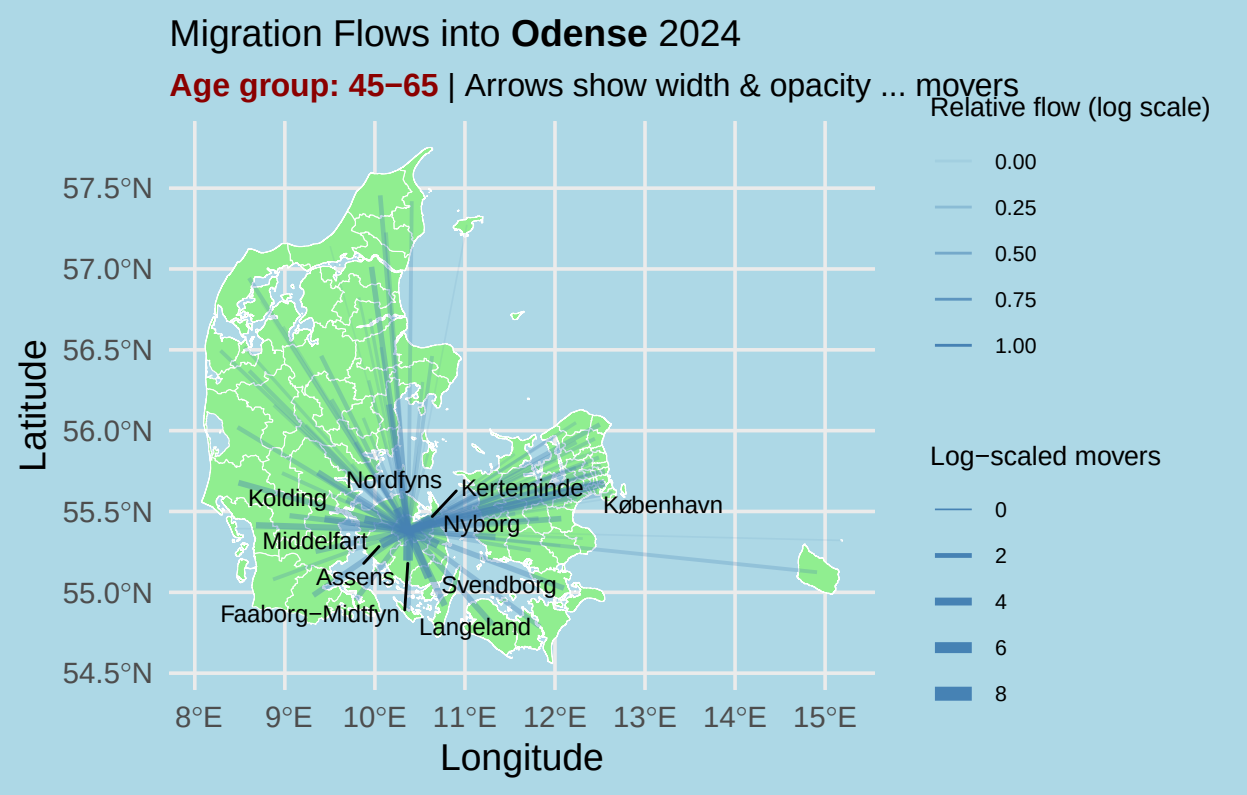

Migration Flows into **Aarhus** 2024
Age group: 30−44 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aalborg
Randers
Viborg
Favrskov
Silkeborg
Syddjurs
Skanderborg
Horsens
Odense
København

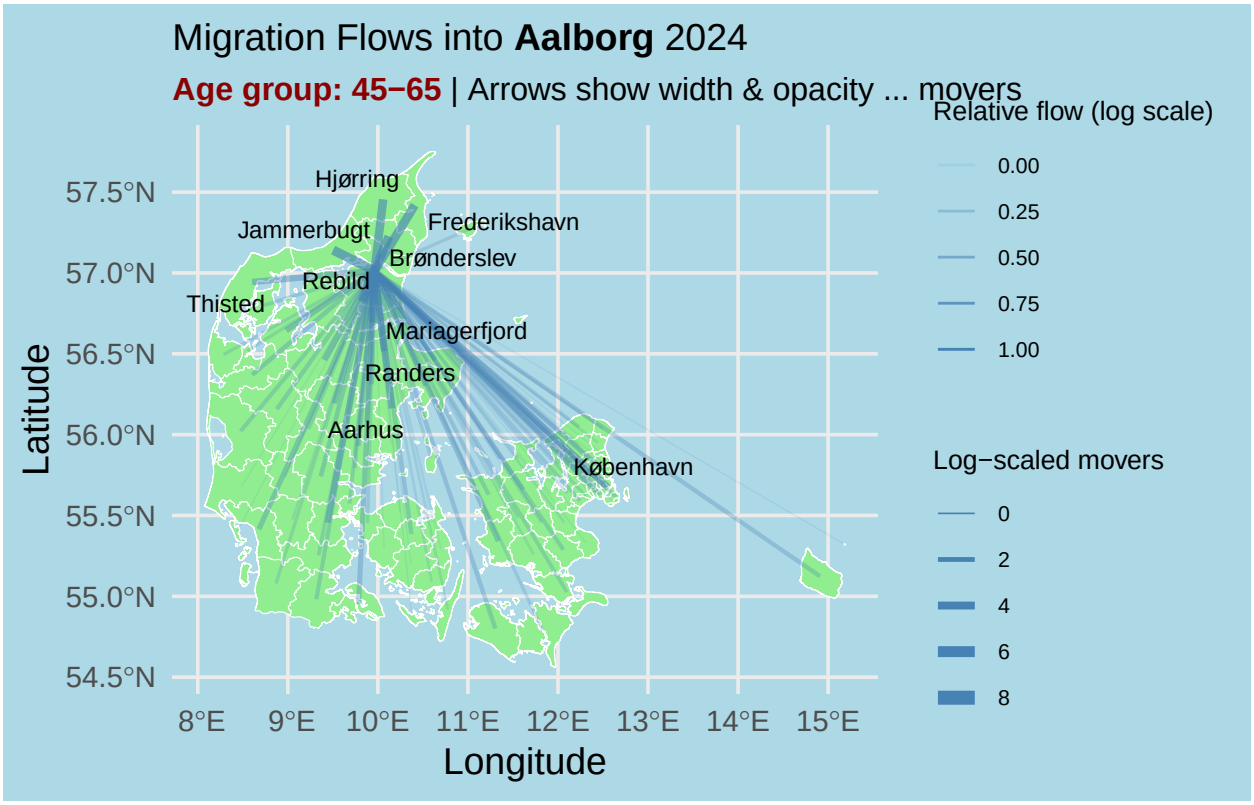

Migration Flows into **Odense** 2024
Age group: 30–44 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aarhus
København
Kolding
Nordfyns
Kerteminde
Nyborg
Assens
Esbjerg
Faaborg−Midtfyn
Svendborg

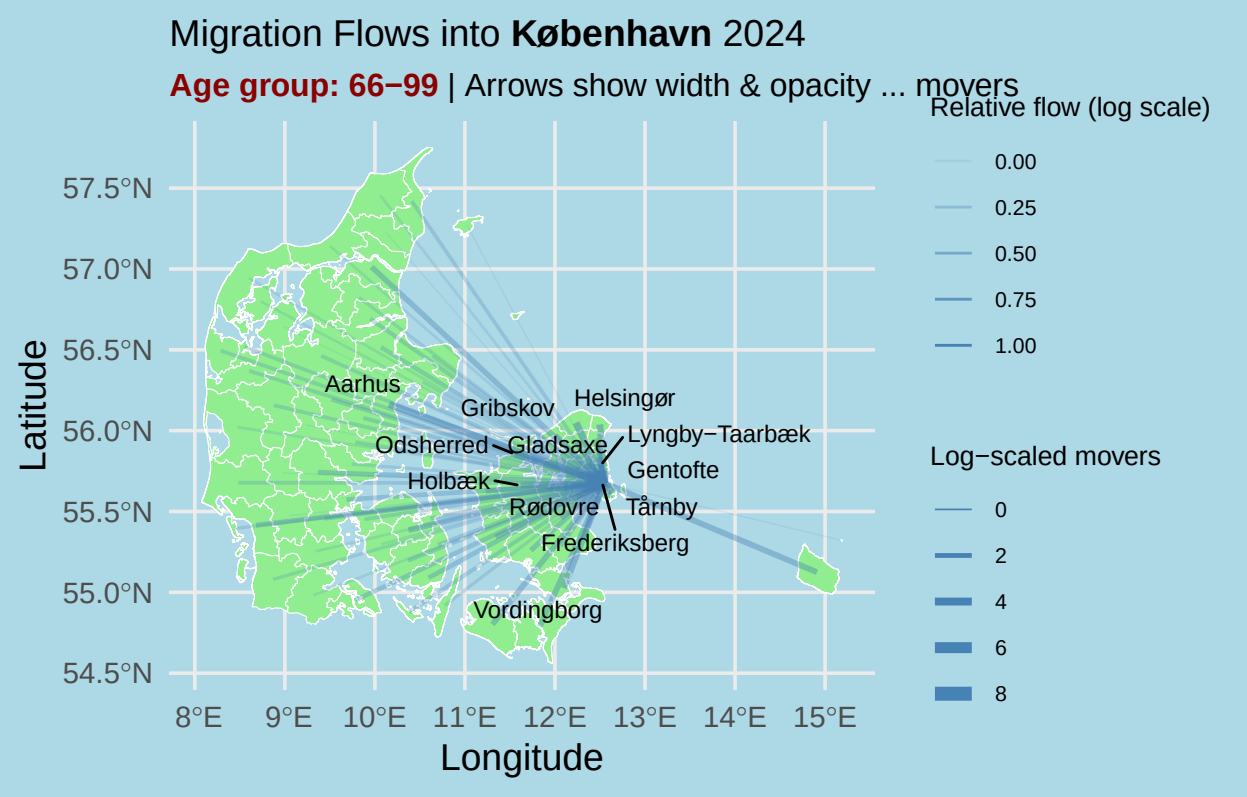

Migration Flows into **Aalborg** 2024
Age group: 30–44 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Hjørring
Frederikshavn
Jammerbugt
Brønderslev
Rebild
Mariagerfjord
Viborg
Randers
Aarhus
København
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Latitude
Longitude

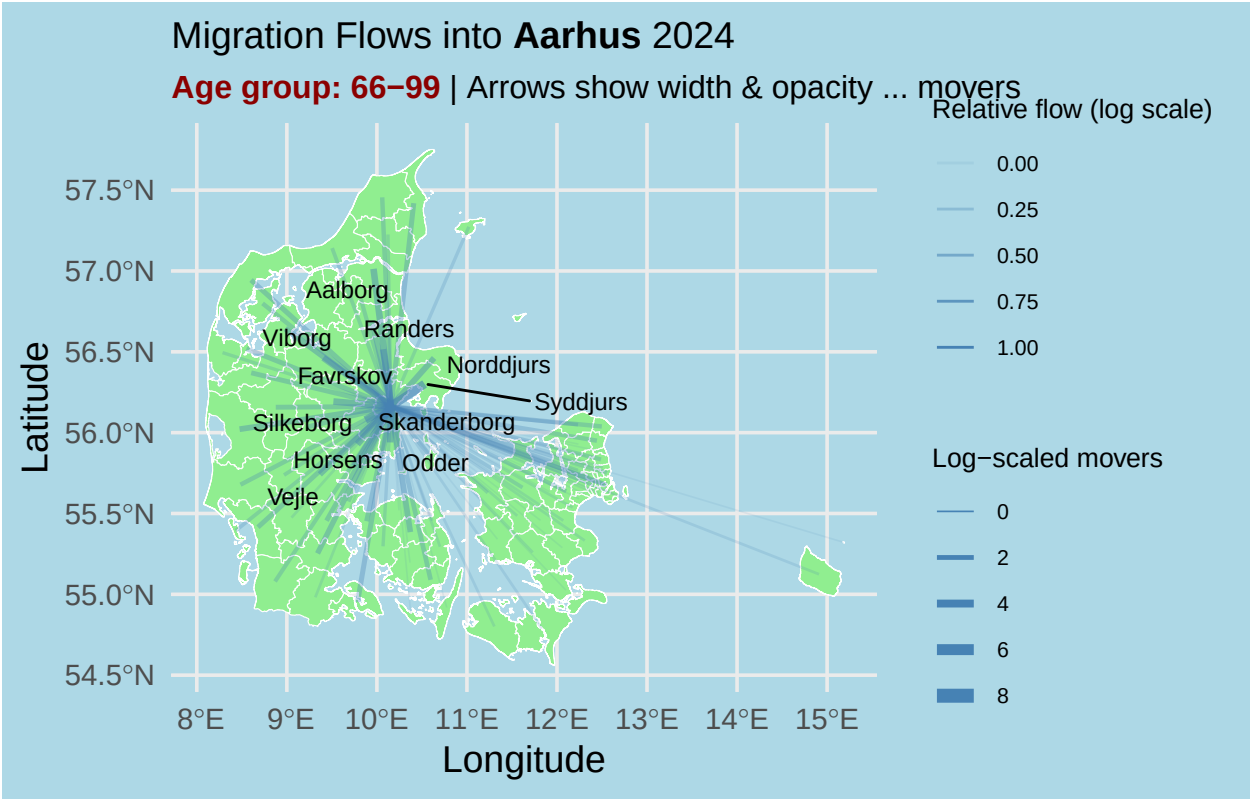

Migration Flows into **København** 2024
Age group: 45–65 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Gribskov
Helsingør
Rudersdal
Lyngby−Taarbæk
Gladsaxe
Gentofte
Roskilde
Hvidovre
Frederiksberg
Tårnby

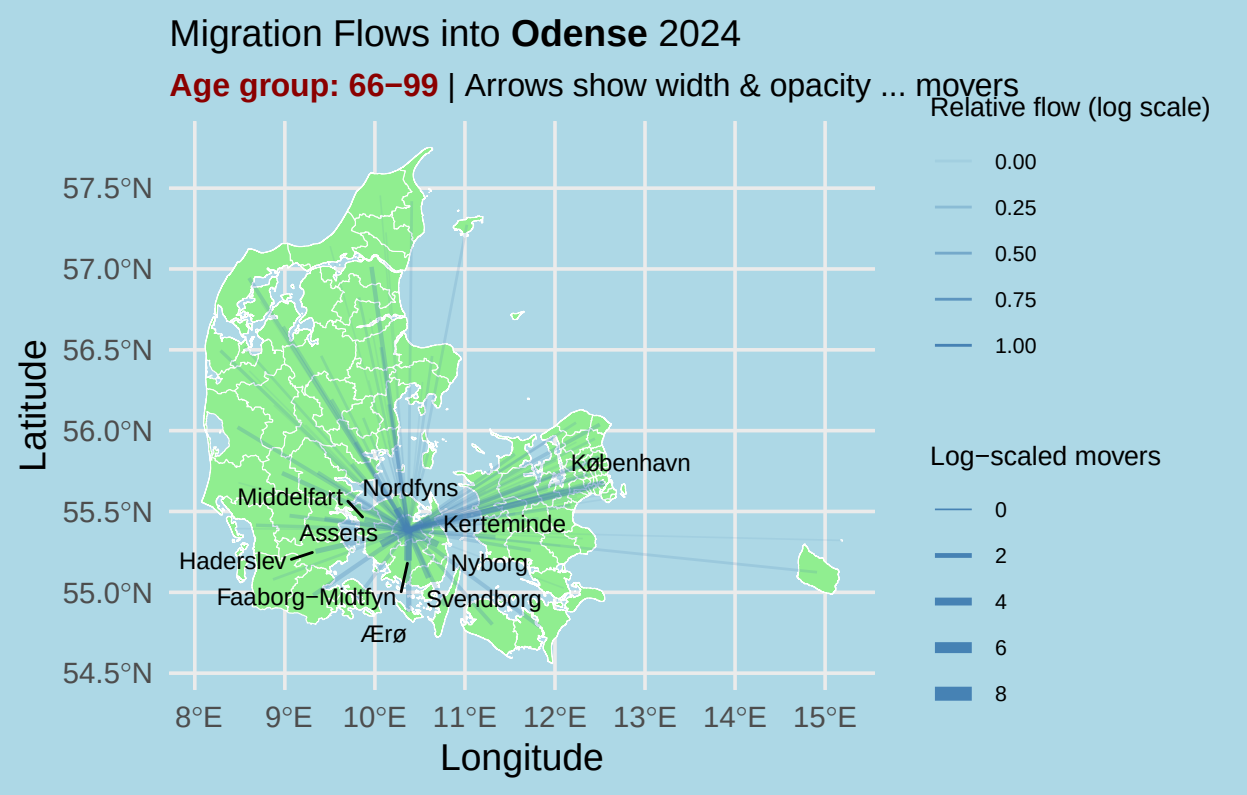

Migration Flows into **Aarhus** 2024
Age group: 45–65 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Viborg
Randers
Favrskov
Norddjurs
Syddjurs
Silkeborg
Skanderborg
Horsens
Odder
København

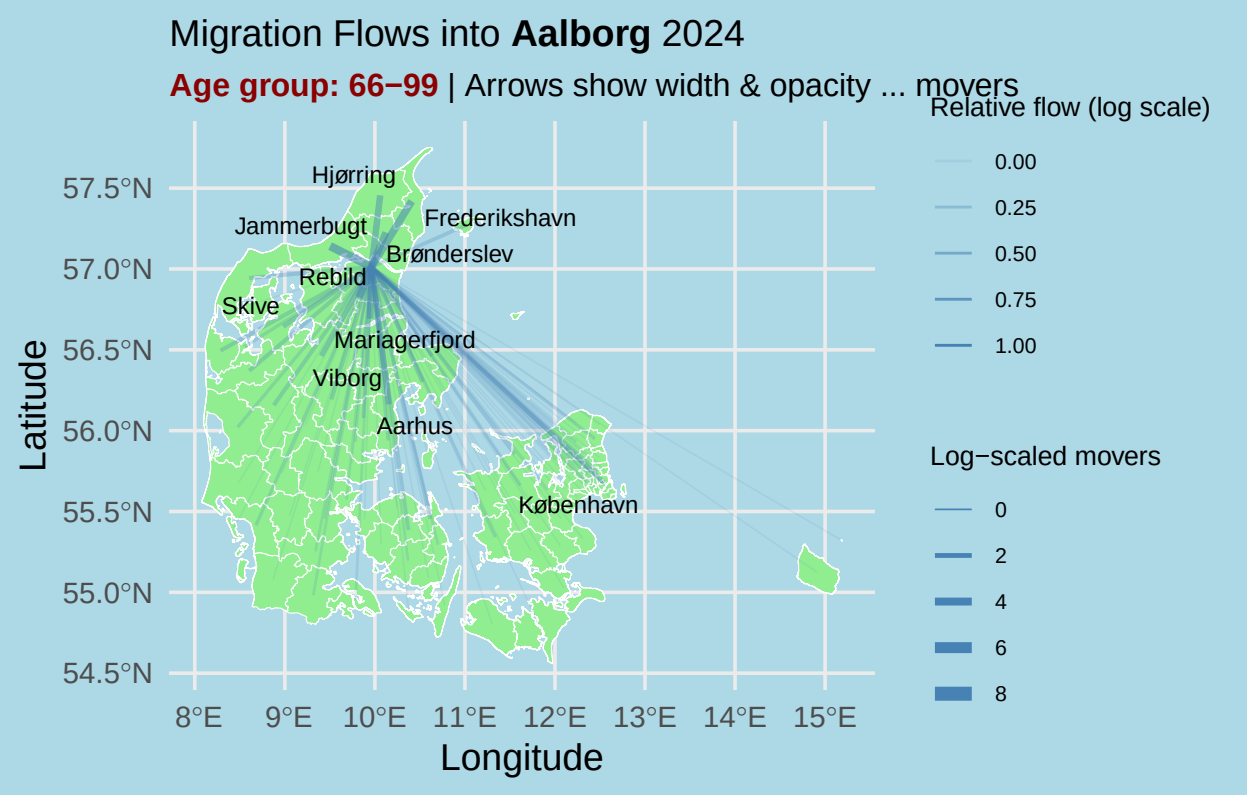

Migration Flows into **Odense** 2024
Age group: 45–65 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Nordfyns
Kerteminde
Kolding
Nyborg
København
Middelfart
Assens
Svendborg
Faaborg−Midtfyn
Langeland


Migration Flows into **Aalborg** 2024
Age group: 45–65 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Hjørring
Frederikshavn
Jammerbugt
Brønderslev
Rebild
Thisted
Mariagerfjord
Randers
Aarhus
København
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude


Migration Flows into **København** 2024
Age group: 66–99 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aarhus
Gribskov
Helsingør
Odsherred
Gladsaxe
Lyngby−Taarbæk
Gentofte
Holbæk
Rødovre
Tårnby
Frederiksberg
Vordingborg


Migration Flows into **Aarhus** 2024
Age group: 66–99 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
Latitude
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
Aalborg
Randers
Viborg
Favrskov
Norddjurs
Syddjurs
Silkeborg
Skanderborg
Horsens
Odder
Vejle


Migration Flows into **Odense** 2024
Age group: 66–99 | Arrows show width & opacity ... movers
Relative flow (log scale)
0.00
0.25
0.50
0.75
1.00
Log−scaled movers
0
2
4
6
8
Latitude
57.5°N
57.0°N
56.5°N
56.0°N
55.5°N
55.0°N
54.5°N
8°E
9°E
10°E
11°E
12°E
13°E
14°E
15°E
Longitude
København
Middelfart
Nordfyns
Kerteminde
Assens
Haderslev
Nyborg
Faaborg−Midtfyn
Svendborg
Ærø




```
## note: a radius circle was initially created, but we elected not to include it in our final product -
```

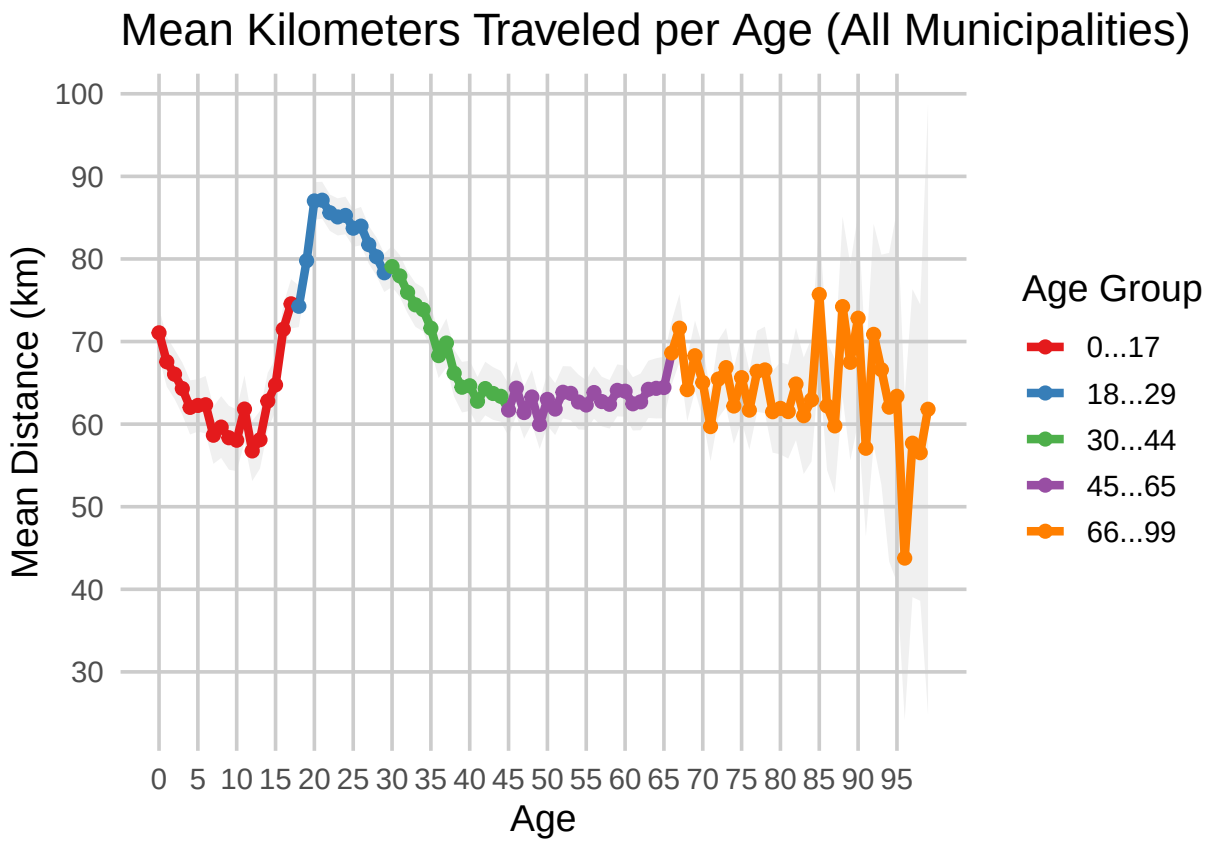
creating gif

```
#defining parameters
age_labels <- names(age_groups)
output_dir <- "out"
gif_dir <- "gifs"
dir.create(gif_dir, showWarnings = FALSE)

for (city in top5_cities) {
  # Generate file paths for this city's images
  image_files <- paste0(output_dir, "/migration_flows_", city, "_age_", gsub("-", "_", age_labels), ".pn

  # Read and combine images
  images <- image_read(image_files)
  animation <- image_animate(image_join(images), fps = 0.5 )  # Adjust fps as desired

  # Save gif
  gif_filename <- file.path(gif_dir, paste0("migration_flows_", city, ".gif"))
  image_write(animation, gif_filename)
}
```

calculating and visualising mean kilometers traveled per relocation by age and age groups

```
# Prepare a dataframe for storing results
age_distances_all <- data.frame()

#loop through each age (0 to 99)
for (age in 0:99) {

  #calculate total movers for this specific age
  df_age <- df %>%
    filter(!is.na(.[[as.character(age)]])) %>%
    mutate(total = .[[as.character(age)]]) %>%
    filter(total > 0)  # Remove rows with zero movers

  #calculate distances for each movement
  distances <- df_age %>%
    left_join(centroids_coords, by = c("orig" = "muni")) %>%
    rename(lon_o = lon, lat_o = lat) %>%
    left_join(centroids_coords, by = c("dest" = "muni")) %>%
    rename(lon_d = lon, lat_d = lat) %>%
    filter(!is.na(lon_o), !is.na(lon_d)) %>%
    mutate(distance_km = distHaversine(cbind(lon_o, lat_o), cbind(lon_d, lat_d)) / 1000) %>%
    summarise(
      mean_distance = mean(distance_km),
      sd_distance = sd(distance_km),
      n = n()
    ) %>%
    mutate(
      age = age,
      se = sd_distance / sqrt(n),  # Standard error
      ci_lower = mean_distance - 1.96 * se,
      ci_upper = mean_distance + 1.96 * se
    )

  # Append to the main dataframe
  age_distances_all <- bind_rows(age_distances_all, distances)
}

age_distances_all <- age_distances_all %>%
  mutate(age_group = case_when(
    age <= 17 ~ "0-17",
    age <= 29 ~ "18-29",
    age <= 44 ~ "30-44",
    age <= 65 ~ "45-65",
    TRUE ~ "66-99"
  ))


#create segment dataframe for colored line segments
segment_data <- age_distances_all %>%
  arrange(age) %>%
```

```r
  mutate(
    xend = lead(age),
    yend = lead(mean_distance),
    group_end = lead(age_group)
  ) %>%
  filter(!is.na(xend))


# plot
ggplot() +
  # confidence interval ribbon in grey
  geom_ribbon(data = age_distances_all, aes(x = age, ymin = ci_lower, ymax = ci_upper),
              fill = "grey80", alpha = 0.3) +

  # base grey line to maintain continuity
  geom_line(data = age_distances_all, aes(x = age, y = mean_distance), color = "grey80", size = 1.2) +

  # colored segments by age group
  geom_segment(data = segment_data, aes(x = age, y = mean_distance,
                                        xend = xend, yend = yend, color = age_group), size = 1.5) +

  # points colored by age group
  geom_point(data = age_distances_all, aes(x = age, y = mean_distance, color = age_group), size = 2) +

  labs(
    title = "Mean Kilometers Traveled per Age (All Municipalities)",
    x = "Age",
    y = "Mean Distance (km)",
    color = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank()
  ) +
  scale_x_continuous(breaks = seq(0, 99, 5)) +
  scale_y_continuous(breaks = seq(20, 100, 10)) +
  scale_color_manual(values = c(
    "0-17" = "#E41A1C",
    "18-29" = "#377EB8",
    "30-44" = "#4DAF4A",
    "45-65" = "#984EA3",
    "66-99" = "#FF7F00"
  ))
```

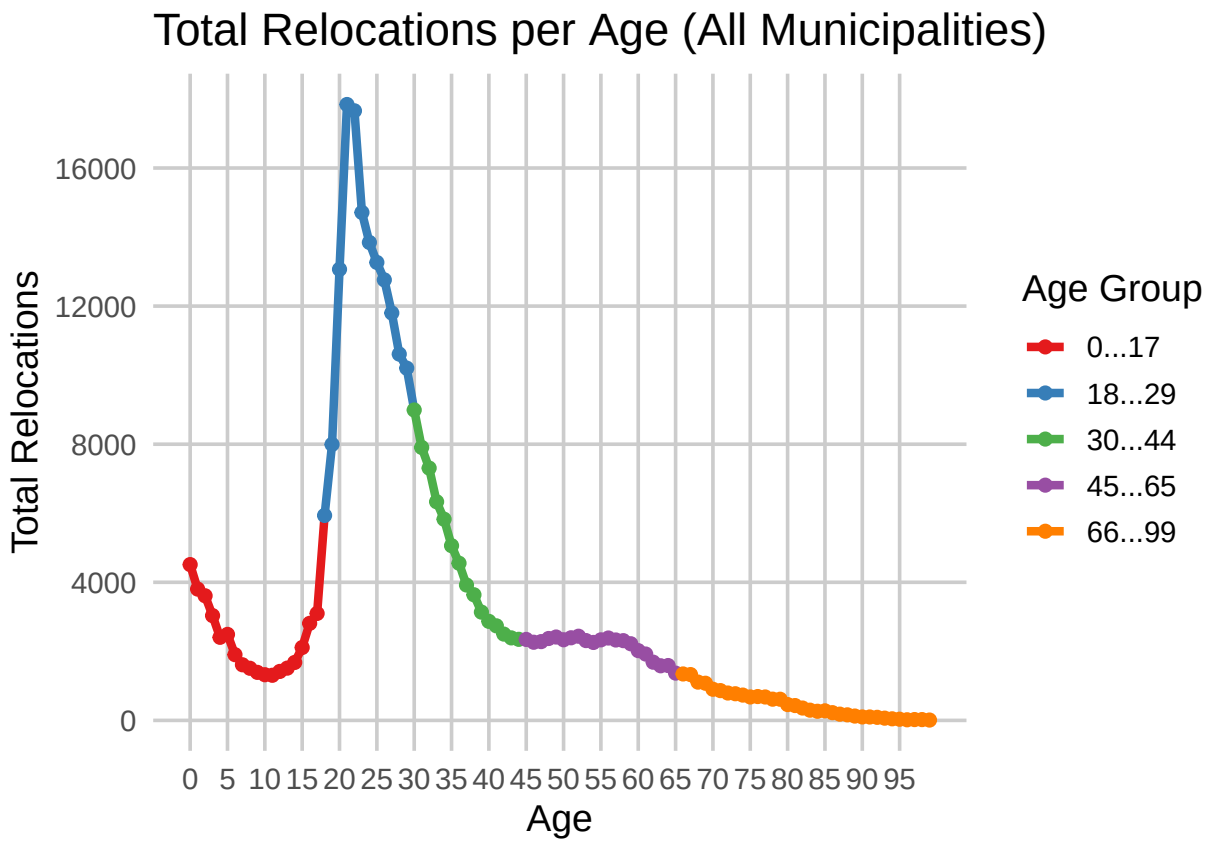



```
# Save the plot
ggsave("out/mean_distance_by_age_colored.png", width = 10, height = 6, dpi = 300)
```

calculating and visualizing total relocations per age and age group

```
#preparing a dataframe for storing results
age_relocations <- data.frame()

#looping through each age
for (age in 0:99) {
  total_relocations <- df %>%
    filter(!is.na(.[[as.character(age)]])) %>%
    summarise(total_relocations = sum(.[[as.character(age)]], na.rm = TRUE)) %>%
    mutate(age = age)

  age_relocations <- bind_rows(age_relocations, total_relocations)
}

# create age group column
age_relocations <- age_relocations %>%
  mutate(age_group = case_when(
    age <= 17 ~ "0-17",
    age <= 29 ~ "18-29",
    age <= 44 ~ "30-44",
```

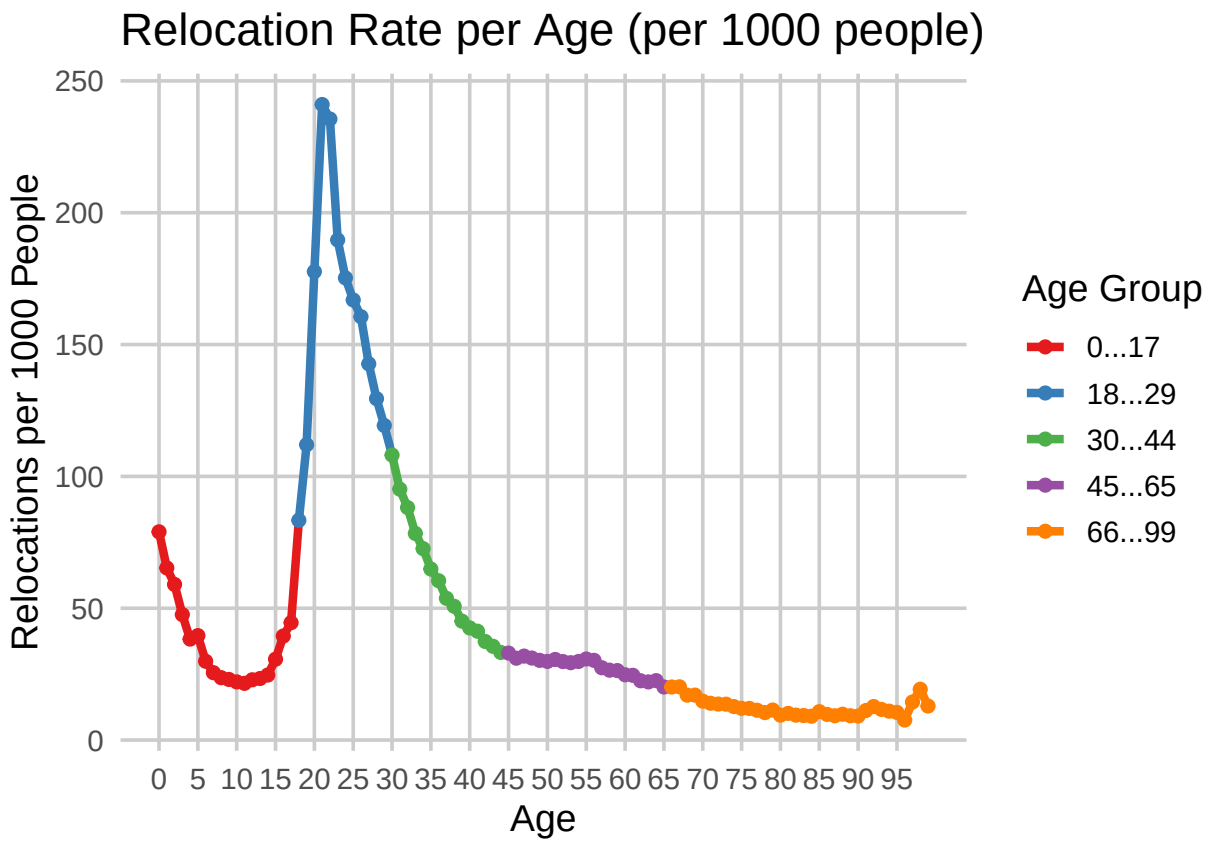
```
    age <= 65 ~ "45-65",
    TRUE ~ "66-99"
  ))

# create a segment dataframe for colored line segments
segment_data <- age_relocations %>%
  arrange(age) %>%
  mutate(
    xend = lead(age),
    yend = lead(total_relocations),
    group_end = lead(age_group)
  ) %>%
  filter(!is.na(xend))  # Remove last row which has NA in lead

# plot continuous line in grey and overlay colored segments
ggplot() +
  geom_line(data = age_relocations, aes(x = age, y = total_relocations), color = "grey80", size = 1) +
  geom_segment(data = segment_data, aes(x = age, y = total_relocations,
                                        xend = xend, yend = yend, color = age_group), size = 1.5) +
  geom_point(data = age_relocations, aes(x = age, y = total_relocations, color = age_group), size = 2) +
  scale_x_continuous(breaks = seq(0, 99, 5)) +
  scale_y_continuous(breaks = seq(0, 20000, 4000)) +
  labs(
    title = "Total Relocations per Age (All Municipalities)",
    x = "Age",
    y = "Total Relocations",
    color = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank()
  ) +
  scale_color_manual(values = c(
    "0-17" = "#E41A1C",   # red
    "18-29" = "#377EB8",  # blue
    "30-44" = "#4DAF4A",  # green
    "45-65" = "#984EA3",  # purple
    "66-99" = "#FF7F00"   # orange
  ))
```




```
ggsave("out/total_relocations_by_age.png", width = 10, height = 6, dpi = 300)
```

```
population_df <- read_csv('new_data/age_cleaned.csv')

# prepare a dataframe for storing total relocations per age
age_relocations <- data.frame()

# loop through each age (0 to 99)
for (age in 0:99) {
  total_relocations <- df %>%
    filter(!is.na(.[[as.character(age)]])) %>%
    summarise(total_relocations = sum(.[[as.character(age)]], na.rm = TRUE)) %>%
    mutate(age = age)

  age_relocations <- bind_rows(age_relocations, total_relocations)
}

#merge with population data and compute relocation rate per 1000 people
age_relocations <- age_relocations %>%
  left_join(population_df, by = "age") %>%
  mutate(
    relocation_rate = (total_relocations / population) * 1000,
    age_group = case_when(
      age <= 17 ~ "0-17",
      age <= 29 ~ "18-29",
```

```
      age <= 44 ~ "30-44",
      age <= 65 ~ "45-65",
      TRUE ~ "66-99"
    )
  )

# create segment data for colored line segments
segment_data <- age_relocations %>%
  arrange(age) %>%
  mutate(
    xend = lead(age),
    yend = lead(relocation_rate),
    group_end = lead(age_group)
  ) %>%
  filter(!is.na(xend))

# plot: relocation rate per 1000 people
ggplot() +
  geom_line(data = age_relocations, aes(x = age, y = relocation_rate), color = "grey80", size = 1) +
  geom_segment(data = segment_data, aes(x = age, y = relocation_rate,
                                        xend = xend, yend = yend, color = age_group), size = 1.5) +
  geom_point(data = age_relocations, aes(x = age, y = relocation_rate, color = age_group), size = 2) +
  scale_x_continuous(breaks = seq(0, 99, 5)) +
  labs(
    title = "Relocation Rate per Age (per 1000 people)",
    x = "Age",
    y = "Relocations per 1000 People",
    color = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank()
  ) +
  scale_color_manual(values = c(
    "0-17" = "#E41A1C",   # red
    "18-29" = "#377EB8",  # blue
    "30-44" = "#4DAF4A",  # green
    "45-65" = "#984EA3",  # purple
    "66-99" = "#FF7F00"   # orange
  ))
```




```
ggsave("out/relocation_rate_by_age.png", width = 10, height = 6, dpi = 300)


### note: this plot may look very similar to the previous one but this one takes into the skewness of a
```

creating cartograms

```
munic <- gadm(country = "DNK", level = 2, path = ".")

# convert to sf
munic_sf <- st_as_sf(munic)

# rename Århus to Aarhus in the NAME_2 column
munic_sf$NAME_2 <- gsub("^Århus$", "Aarhus", munic_sf$NAME_2)
munic_sf$NAME_2 <- gsub("^Høje Taastrup$", "Høje-Taastrup", munic_sf$NAME_2)
munic_sf$NAME_2 <- gsub("^Vesthimmerland$", "Vesthimmerlands", munic_sf$NAME_2)

# transform to a Projected Coordinate System (ETRS89 / UTM zone 32N)
munic_sf <- st_transform(munic_sf, crs = 25832)

#defining age groups
age_groups <- list(
  "0-17"   = 0:17,
  "18-29"  = 18:29,
```

```
  "30-44"  = 30:44,
  "45-65"  = 45:65,
  "66-99"  = 66:99
)

# looping through age groups
for (group_name in names(age_groups)) {

  # getting age range
  age_range <- age_groups[[group_name]]

  # calculate total relocations per municipality for the current age group
  relocations_per_muni <- df %>%
    group_by(dest) %>%
    summarise(total_relocations = sum(across(as.character(age_range)), na.rm = TRUE)) %>%
    ungroup()

  # join with Spatial Data
  muni_group_sf <- munic_sf %>%
    left_join(relocations_per_muni, by = c("NAME_2" = "dest")) %>%
    filter(!is.na(total_relocations))

  # create the cartogram
  muni_cartogram <- cartogram_cont(muni_group_sf, weight = "total_relocations", itermax = 10)

  # plot the cartogram
  p <- ggplot(muni_cartogram) +
    geom_sf(aes(fill = total_relocations), color = "black", size = 0.2) +
    scale_fill_viridis_c(option = "plasma", trans = "log", limits = c(1, 30000)) +
    labs(
      title = paste("Cartogram of Relocations (Age", group_name, ") per Municipality in Denmark"),
      fill = "Total Relocations"
    ) +
    theme_minimal(base_size = 14) +
    theme(
      legend.position = "right",
      plot.title = element_text(size = 16, face = "bold")
    )

  # Save the Plot
  ggsave(filename = paste0("out/relocations_wee__cartogram_", group_name, ".png"), plot = p, width = 10
}
```

```
#creating gif
pacman::p_load('magick')

# Define your age labels (in the same order as your cartogram PNGs)
age_labels <- c("0-17", "18-29", "30-44", "45-65", "66-99")
output_dir <- "out"
gif_dir <- "gifs"
dir.create(gif_dir, showWarnings = FALSE)

# Generate file paths for the images
```

```r
image_files <- paste0(output_dir, "/relocations_wee__cartogram_", age_labels, ".png")

# Read and animate the images
if (all(file.exists(image_files))) {
  images <- image_read(image_files)
  animation <- image_animate(image_join(images), fps = 0.5)  # 1 frame every 4 seconds

  # Save the GIF
  gif_filename <- file.path(gif_dir, "relocation_cartogram.gif")
  image_write(animation, gif_filename)
  message("GIF saved to: ", gif_filename)
} else {
  warning("One or more image files are missing.")
}
```

creating popularity cartograms

```r
library(readxl)

# loading population data
population_file_path <- "new_data/FOLK1AM.xlsx"

# extracting relevant data
munic_pop_df <- read_excel(population_file_path, skip = 2) %>%
  select(municipality = `...1`,
         population_jan_2025 = `2024M12`) %>%
  mutate(municipality = trimws(municipality)) %>%
  filter(!is.na(municipality), municipality != "Hele landet") %>%
  filter(!municipality %in% c("Region Hovedstaden", "Region Sjælland",
                              "Region Syddanmark", "Region Midtjylland",
                              "Region Nordjylland"))

# view the cleaned population data
head(munic_pop_df)
```

```
## # A tibble: 6 x 2
##   municipality  population_jan_2025
##   <chr>                       <dbl>
## 1 København                  668906
## 2 Frederiksberg              106360
## 3 Dragør                      14468
## 4 Tårnby                      44009
## 5 Albertslund                 28133
## 6 Ballerup                    52744
```

```r
#loading municipalities
munic <- gadm(country = "DNK", level = 2, path = ".")

munic_sf <- st_as_sf(munic)
munic_sf$NAME_2 <- gsub("^Århus$", "Aarhus", munic_sf$NAME_2)
munic_sf$NAME_2 <- gsub("^Høje Taastrup$", "Høje-Taastrup", munic_sf$NAME_2)
```

```r
munic_sf$NAME_2 <- gsub("^Vesthimmerland$", "Vesthimmerlands", munic_sf$NAME_2)
munic_sf <- st_transform(munic_sf, crs = 25832)

#    Load Population Data
population_file_path <- "new_data/FOLK1AM.xlsx"

munic_pop_df <- read_excel(population_file_path, skip = 2) %>%
  select(municipality = `...1`,
         population_jan_2025 = `2024M12`) %>%
  mutate(municipality = trimws(municipality)) %>%
  filter(!is.na(municipality), municipality != "Hele landet") %>%
  filter(!municipality %in% c("Region Hovedstaden", "Region Sjælland",
                              "Region Syddanmark", "Region Midtjylland",
                              "Region Nordjylland"))

#clean df$dest to match municipality names
df$dest <- gsub("^Århus$", "Aarhus", df$dest)
df$dest <- gsub("^Høje Taastrup$", "Høje-Taastrup", df$dest)
df$dest <- gsub("^Vesthimmerland$", "Vesthimmerlands", df$dest)
df$dest <- trimws(df$dest)

#defining age groups
age_groups <- list(
  "0-17"   = 0:17,
  "18-29"  = 18:29,
  "30-44"  = 30:44,
  "45-65"  = 45:65,
  "66-99"  = 66:99
)

# calculating popularity indeces
popularity_all <- map_dfr(names(age_groups), function(group_name) {
  age_range <- age_groups[[group_name]]

  relocations_per_muni <- df %>%
    group_by(dest) %>%
    summarise(total_relocations = sum(across(as.character(age_range)), na.rm = TRUE)) %>%
    ungroup()

  relocations_per_muni %>%
    left_join(munic_pop_df, by = c("dest" = "municipality")) %>%
    mutate(
      popularity_index = (total_relocations / population_jan_2025) * 1000,
      age_group = group_name
    ) %>%
    filter(!is.na(popularity_index))
})

#plotting cartograms
for (group_name in names(age_groups)) {

  group_data <- popularity_all %>%
    filter(age_group == group_name)
```

```r
  muni_group_sf <- munic_sf %>%
    left_join(group_data, by = c("NAME_2" = "dest")) %>%
    filter(!is.na(popularity_index))

  muni_cartogram <- cartogram_cont(muni_group_sf, weight = "popularity_index", itermax = 10)

  p <- ggplot(muni_cartogram) +
    geom_sf(aes(fill = popularity_index), color = "black", size = 0.2) +
    scale_fill_viridis_c(option = "plasma", trans = "log", limits = c(1, 65)) + ##the values were found
    labs(
      title = paste("Popularity Index (Age",group_name,"): Relocations per 1000 Residents"),
      fill = "Relocations per 1000 Residents"
    ) +
    theme_minimal(base_size = 14) +
    theme(
      legend.position = "right",
      plot.title = element_text(size = 16, face = "bold")
    )

  ggsave(filename = paste0("out/popularity_cartogram_", group_name, ".png"),
         plot = p, width = 10, height = 8, dpi = 300)
}

#creating gif of cartograms
pacman::p_load('magick')

age_labels <- names(age_groups)
output_dir <- "out"
gif_dir <- "gifs"
dir.create(gif_dir, showWarnings = FALSE)

image_files <- paste0(output_dir, "/popularity_cartogram_", age_labels, ".png")

if (all(file.exists(image_files))) {
  images <- image_read(image_files)
  animation <- image_animate(image_join(images), fps = 0.5)

  gif_filename <- file.path(gif_dir, "popularity_cartogram.gif")
  image_write(animation, gif_filename)
  message("GIF saved to: ", gif_filename)
} else {
  warning(" One or more image files are missing.")
}
```