

ggplot2

Options, packages and examples for PDF print

Fabian Koch

Contents

Data	3
Informationen	4
package Infos, repositories, links	4
scales	4
Vorlagen	4
color palettes	4
Themes	4
Snippets	4
Beschriftungen	4
Plots	5
Plot - Layout	13
Gemischtes 1 und 2 Spalten Layout	13
1 Spaltiges Layout	14

Data

Informationen

package Infos, repositories, links

scales

package zur Berechnung von Skalen, bspw. für Achseneinteilungen etc.

Vorlagen

color palettes

```
# Beispiel für manuelle Festlegung diskreter Variablen
PAL_Gliederung_Colour <- c(SR = "blue", UBZTP = "orange")
PAL_Gebiet_fill <- c("yellow3", "black", "grey", "maroon3")
PAL_pal9GnPu <- c("#762a83", "#9970ab", "#c2a5cf", "#e7d4e8", "#f7f7f7", "#d9f0d3", "#a6dba0", "#5aae61")
PAL_virpal <- viridisLite::viridis(6)
PAL_col6qual <- c("#66c2a5", "#fc8d62", "#8da0cb", "#e78ac3", "#a6d854", "#ffd92f")
PAL_div_lowcontrast <- c("#fc8d62", "#e78ac3", "#66c2a5", "#8da0cb", "#a6d854", "#ffd92f", "#e5c494")
```

Themes

Reference Manual

Hier werden themes für unterschiedliche Plot Typen und Größen festgelegt.

Vorschlag für eine Syntax:

theme_package (ggplot, plotly etc.)_Plottyp (bspw. map oder barchart)_Format(print oder HTML)_Seitenformat (DINA4 oder A5 etc.)_SpaltenLayout (1, 2 oder 3-spaltig)

Beispiel für ein Theme für ein ggplot2 plot, Typ Karte, für Printfassungen im A4 Format, 1-spaltig:

theme_ggplot2_map_print_A4_1c

ggf. später zu vereinfachen, wenn sich herausstellt, dass keine individuellen Anpassungen zwischen 1- und 2 spaltig erfolgen müssen

/newpage

Snippets

Beschriftungen

Copy/Paste Vorlage für labs()

Plots

Karte

```
mapData <- data_WorldGeom %>%
  select(
    name,
    continent,
    pop_est,
    income_grp,
    geometry) %>%
  filter(continent == "Asia") %>%
  mutate(
    # Vereinigung der 5 Kategorien zu 3
    income_grp = forcats::fct_collapse(income_grp,
      Hoch = c(
        "1. High income: OECD",
        "2. High income: nonOECD"),
      Mittel = c(
        "3. Upper middle income",
        "4. Lower middle income"),
      Niedrig = c(
        "5. Low income"))))

mapPlot <- ggplot(mapData) +
  # da das data.frame eine geometry Spalte besitzt, kommt geom_sf ohne x und y bzw. Rechts- und Hochw
  # data.frames mit Rechts- und Hochwerten können über sf::st_as_sf in dieses Format konvertiert werd
  # https://www.rdocumentation.org/packages/sf/versions/0.9-7/topics/st_as_sf
  # https://r-spatial.github.io/sf/reference/st_as_sf.html
  geom_sf(
    data = mapData,
    aes(fill = income_grp)) +
  # Externe Farbpalette, Beispiel viridis
  # https://www.rdocumentation.org/packages/viridis/versions/0.5.1/topics/scale_color_viridis
  viridis::scale_fill_viridis(
    # Diskrete Variable (Einkommensgruppen)
    discrete = TRUE,
    # Umkehr der Palette, damit dunkel = Niedrig
    direction = -1) +
  # ggrepel ist ein package, das Beschriftungen oder Labels so ausrichtet, dass es zu keinen Überlapp
  ggrepel::geom_label_repel(
    # man kann die ausgewählte Variable in ggplot vorab mit "subset" filtern
    data = subset(mapData, income_grp == "Niedrig"),
    # ohne stat = "sf_coordinates" kann ggrepel keine "geometry" Angaben verarbeiten
    stat = "sf_coordinates",
    aes(
      geometry = geometry,
      label = name)) +
  # siehe theme settings oben
  theme_ggplot2_map_print_A4_1C() +
  # Beschriftungen
  labs(
    title = "Titel",
    subtitle = "Untertitel",
```

```

caption = "Fußnote",
tag = "label",
fill = "Titel Legende") +
xlab("Beschriftung x") +
ylab("Beschriftung y")

```

Scatter Plot mit Facet Wrap

```

ScatterData <- data_WorldData %>%
  select(
    name,
    continent,
    inequality,
    well_being,
    gdp_cap_est,
    economy) %>%
  group_by(
    continent) %>%
  mutate(avg_gdp = mean(gdp_cap_est, na.rm = TRUE)) %>%
  ungroup() %>%
  drop_na() %>%
  mutate(
    # Vereinigung der Kategorien
    economy = forcats::fct_collapse(economy,
      entwickelt = c(
        "1. Developed region: G7",
        "2. Developed region: nonG7"),
      aufstrebend = c(
        "3. Emerging region: BRIC",
        "4. Emerging region: MIKT",
        "5. Emerging region: G20"),
      "nicht-entwickelt" = c(
        "6. Developing region",
        "7. Least developed region"))))

ScatterPlot <-
  ggplot(ScatterData) +
  geom_point(
    aes(
      inequality,
      well_being,
      # Anordnung der Kontinente nach absteigender, durchschnittlicher Bevölkerung
      colour = forcats::fct_reorder(continent, desc(avg_gdp))),
    alpha = 0.8) +
  facet_wrap(
    ~ economy,
    nrow = 2) +
  scale_colour_manual(
    values = PAL_div_lowcontrast,
    guide = guide_legend(
      title.position = "top",
      title="Kontinente",

```

```

        direction="horizontal",
        nrow = 3,
        ncol = 2)) +
geom_smooth(aes(x = inequality, y = well_being), method = "lm") +
theme_minimal() +
xlab("Wohlbefinden") +
ylab("Ungleichheit") +
theme(
  # Legenden Position, Alternativ: "top", "bottom", "right", "left"
  legend.position = c(0.72, 0.27),
  # Legenden Schrift fett
  legend.title = element_text(face="bold"),
  # Abstand der Achsentitel zum Achsentext
  axis.title.x = element_text(margin = margin(t = 15, r = 0, b = 0, l = 0)),
  axis.title.y = element_text(margin = margin(t = 0, r = 15, b = 0, l = 0)))

```

population Pyramid Varianten

data

```

popPyData <- data_popFM_long %>%
  # filtert Kontinente und Länder Gruppen heraus
  filter(!country_code >= 900) %>%
  dplyr::group_by(
    year,
    country,
    gender) %>%
  dplyr::mutate(
    percPop_country_gender = round(population/sum(population)*100,1)) %>%
  ungroup() %>%
  mutate(perc_MF = ifelse(gender == "F", percPop_country_gender*-1, percPop_country_gender)) %>%
  filter(
    year == "2015"
    & country == "Germany")

```

Population Pyramid with Bar Charts

```

data <- popPyData
scale_variable <- c("percPop_country_gender")

scale_max <-
  data %>%
  select(all_of(scale_variable)) %>%
  distinct() %>%
  max() %>%
  round(., 0)

theme_ggplot2_popPy_print_A4_1C <- function(...) {
  theme_minimal()
  theme(

```

```

text = element_text(
  # family = default_font_family,
  color = default_font_color),
# remove all axes
axis.line = element_blank(),
axis.text.x = element_text(
  size = 7, hjust = 0,
  color = default_font_color),
axis.text.y = element_text(
  margin = unit(c(t = 0, r = -0.7, b = 0, l = 0), "cm")),
axis.ticks = element_blank(),
axis.title.y=element_blank(),
# panel.grid.major = element_blank(),
# panel.grid.minor = element_blank(),
# background colors
plot.background = element_rect(fill = default_background_color,
                                color = NA),
panel.background = element_rect(fill = default_background_color,
                                color = NA),
legend.background = element_rect(fill = default_background_color,
                                color = NA),

# borders and margins
panel.border = element_blank(),
panel.spacing = unit(c(0, 0, 0, 0), "cm"),
# titles
legend.title = element_text(size = 7),
legend.text = element_text(size = 7, hjust = 0,
                             color = default_font_color),
plot.title = element_text(size = 20,
                           color = default_font_color,
                           face = "bold"),
plot.subtitle = element_text(size = 15,
                              color = default_font_color,
                              margin = margin(b = -0.1,
                                              t = -0.1,
                                              l = 2,
                                              unit = "cm"),
                              debug = F),

# captions
plot.caption = element_text(size = 10,
                             hjust = 0,
                             margin = margin(t = 0.2,
                                              b = 0,
                                              unit = "cm"),
                             color = "#939184"),

...
)
}

Male <- ggplot(
  data = popPyData[popPyData$gender=="M",],
  aes(

```



```

    x = percPop_country_gender,
    y = age)) +
geom_col(fill = "cornflowerblue") +
theme_ggplot2_popPy_print_A4_1C() +
scale_x_continuous(
  name = "",
  limits = c(0, scale_max),
  breaks= scales::breaks_pretty(5),
  expand = c(0, 1)) +
theme(
  # durch die umgekehrte Ausrichtung der weiblichen/männlichen Plots
  # müssen die margins separat gesetzt werden
  plot.margin = unit(c(t = 0, r = -0.5, b = 0, l = -0.5), "cm")) +
ylab("Anteil")

```

```

Female <- ggplot(
  data = popPyData[popPyData$gender=="F",],
  aes(
    x = percPop_country_gender,
    y = age)) +
geom_col(fill = "darkseagreen2") +
theme_ggplot2_popPy_print_A4_1C() +
scale_x_reverse(
  name = "",
  # da Achse für weibliche Personen umgekehrt wird (_reverse)
  # muss auch der min und max Wert umgekehrt angegeben werden
  limits = c(scale_max, 0),
  breaks= scales::breaks_pretty(5),
  expand = c(0, 1)) +
theme(
  # durch die umgekehrte Ausrichtung der weiblichen/männlichen Plots
  # müssen die margins separat gesetzt werden
  plot.margin = unit(c(t = 0, r = 0, b = 0, l = -0.5), "cm"),
  axis.text.y = element_blank()) +
ylab("Angaben in %")

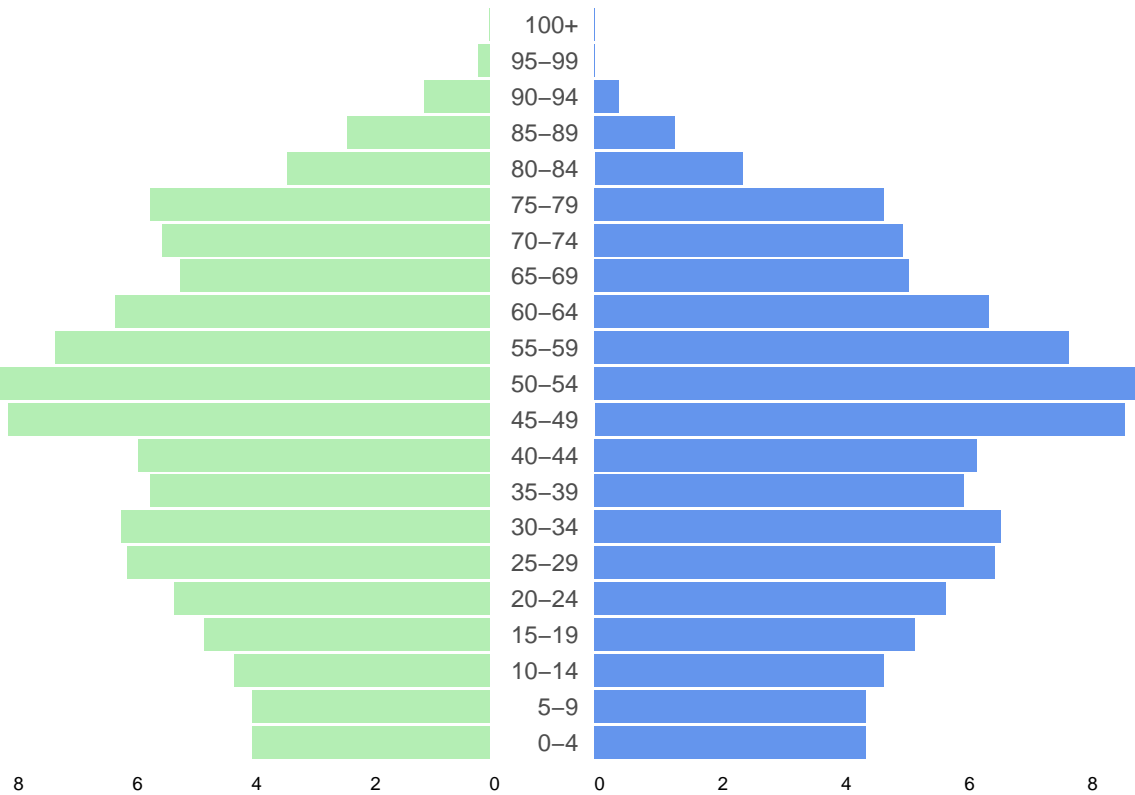
```

```

gridExtra::grid.arrange(
  Female,
  Male,
  widths=c(0.5,0.5),
  ncol=2,
  # https://www.rdocumentation.org/packages/grid/versions/3.6.2/topics/grid.text
  top = grid::textGrob(
    "Alterspyramide",
    hjust = 3,
    gp=grid::gpar(
      fontsize=10,
      font=2)))

```

Alterspyramide



Bar Charts

```
countryList <- c("Germany", "Romania", "Bulgaria", "Syrian Arab Republic")

BarData <- data_popFM_long %>%
  # filtert Kontinente und Länder Gruppen heraus
  filter(!country_code >= 900) %>%
  dplyr::group_by(
    year,
    country,
    gender) %>%
  dplyr::mutate(
    percPop_country_gender = round(population/sum(population)*100,1)) %>%
  filter(year == "2015") %>%
  filter(country %in% countryList)

plotBarGrouped <- BarData %>%
  ggplot(
    aes(
      x=age,
      y=percPop_country_gender,
      group=gender,
      fill=gender)) +
  geom_bar(
    position = "dodge", width=0.7,
```

```

    stat = "identity") +
  scale_fill_manual(labels = c("männlich", "weiblich"), values = c("cornflowerblue", "darkseagreen2")) +
  scale_x_discrete(guide = guide_axis(angle = 90)) +
  labs(
    title = "Anteile der Bevölkerung nach Geschlecht und Altersgruppe",
    subtitle = "in West-europäischen Ländern, 2015",
    caption = "Datenset wpp2015, Age- and sex-specific population estimates and projections ",
    tag = "Abb. 01",
    fill = "Geschlecht") +
  xlab("Altersgruppen") +
  ylab("Anteil an geschlechtsspezifischer Hauptwohnbevölkerung") +
  facet_wrap(~ country) +
  theme_minimal()

plotBarFill <- BarData %>%
  ggplot() +
  geom_bar(
    # filter des Datensatzes auf weibliche Personen
    # Alternative: BarData[BarData$gender=="F",]
    data = subset(BarData, gender == "F"),
    aes(
      x=age,
      y=percPop_country_gender),
    fill = NA,
    color = "Black",
    # position = "identity",
    stat="identity") +
  geom_bar(
    data = subset(BarData, gender == "M"),
    aes(
      x=age,
      y=percPop_country_gender),
    alpha = 0.3,
    # position = "identity",
    stat="identity") +
  scale_x_discrete(guide = guide_axis(angle = 90)) +
  # scale_colour_manual(labels = c("männlich", "weiblich"), values=gender, aesthetics = c("colour", "fill")) +
  scale_colour_manual(labels = c("männlich", "weiblich"), values=c("lightblue4", "red")) +
  labs(
    title = "Anteile der Bevölkerung nach Geschlecht und Altersgruppe",
    subtitle = "in West-europäischen Ländern, 2015",
    caption = "Datenset wpp2015, Age- and sex-specific population estimates and projections ",
    tag = "Abb. 01",
    fill = "Geschlecht") +
  xlab("Altersgruppen") +
  ylab("Anteil an geschlechtsspezifischer Hauptwohnbevölkerung") +
  facet_wrap(~ country, nrow = 2, ncol = 2) +
  theme_minimal()

plotBarOverlay <- BarData %>%

```

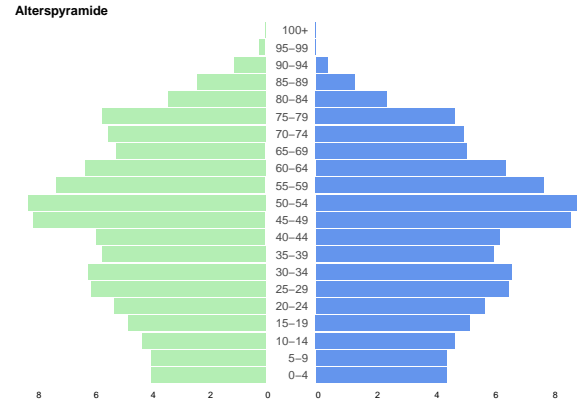
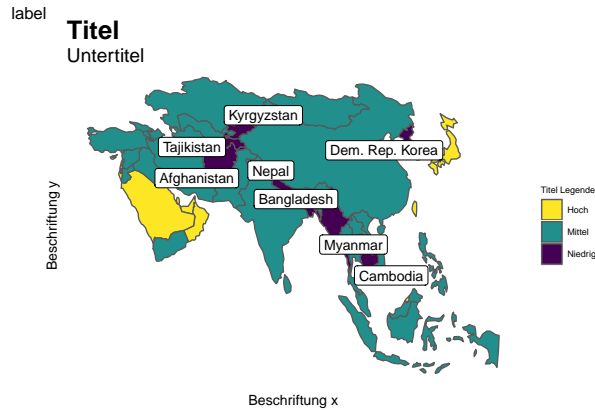
```

ggplot(
  aes(
    x=age,
    y=percPop_country_gender,
    fill = gender
    # color = gender,
    # alpha = gender
  )) +
  geom_bar(
    alpha = 0.6,
    position = "identity",
    stat="identity") +
  # scale_colour_manual(values=c("lightblue4", "red")) +
  scale_fill_manual(
    labels = c("männlich", "weiblich"),
    values=c("lightblue", "pink")) +
  scale_x_discrete(guide = guide_axis(angle = 90)) +
  # scale_alpha_manual(values=c(.3, .8)) +
  labs(
    title = "Anteile der Bevölkerung nach Geschlecht und Altersgruppe",
    subtitle = "in West-europäischen Ländern, 2015",
    caption = "Datenset wpp2015, Age- and sex-specific population estimates and projections ",
    tag = "Abb. 01",
    fill = "Geschlecht") +
  xlab("Altersgruppen") +
  ylab("Anteil an geschlechtsspezifischer Hauptwohnbevölkerung") +
  facet_wrap(~ country, nrow = 2, ncol = 2) +
  theme_minimal() +
  theme(legend.position = "top")

```

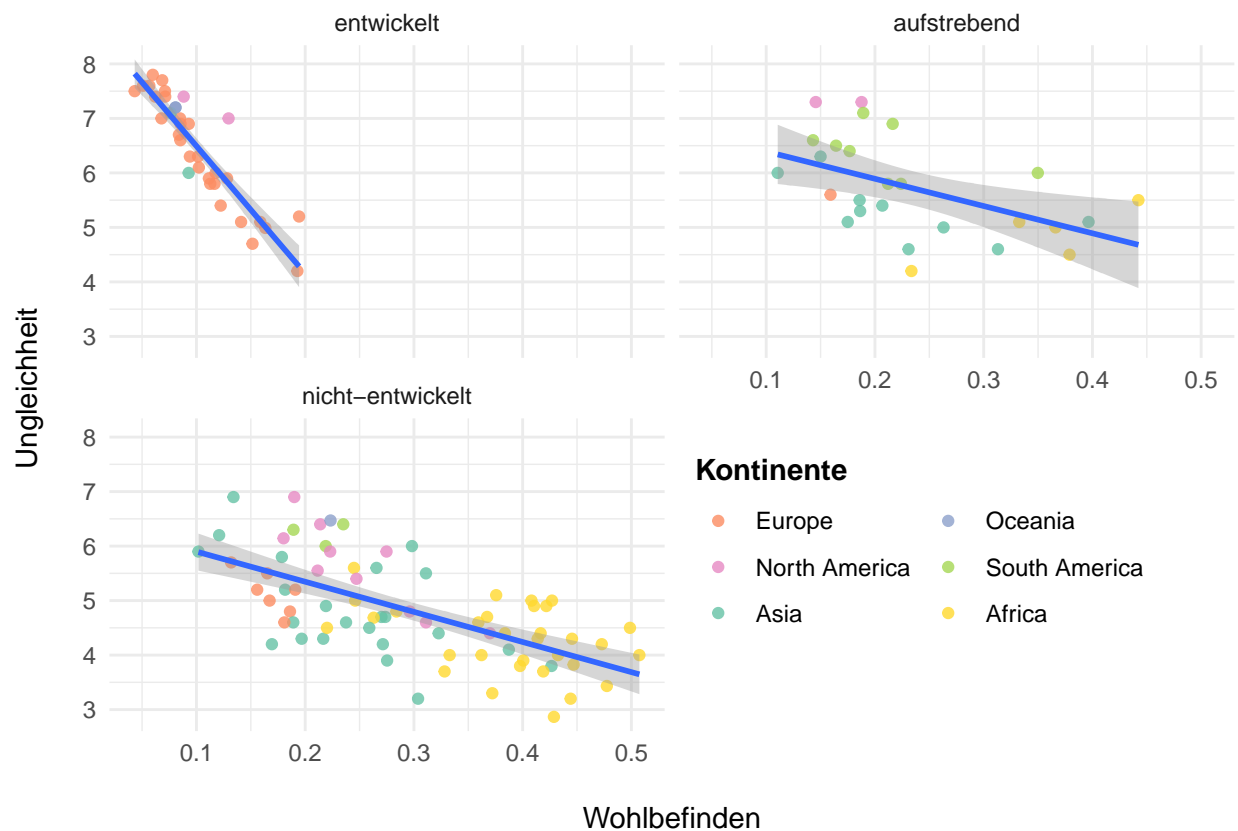
Plot - Layout

Gemischtes 1 und 2 Spalten Layout



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.



1 Spaltiges Layout

Alternativen zur Population Pyramid

plotBarGrouped

Abb. 01

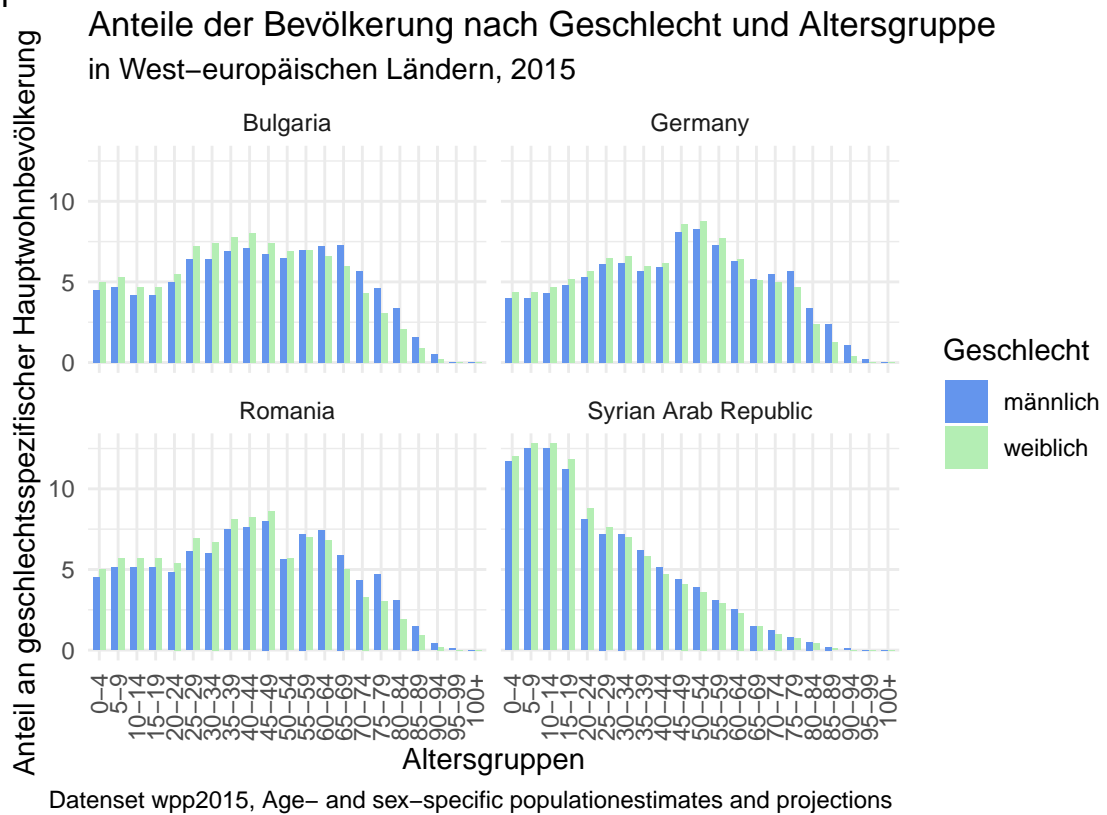


Abb. 01

Anteile der Bevölkerung nach Geschlecht und Altersgruppe
in West-europäischen Ländern, 2015

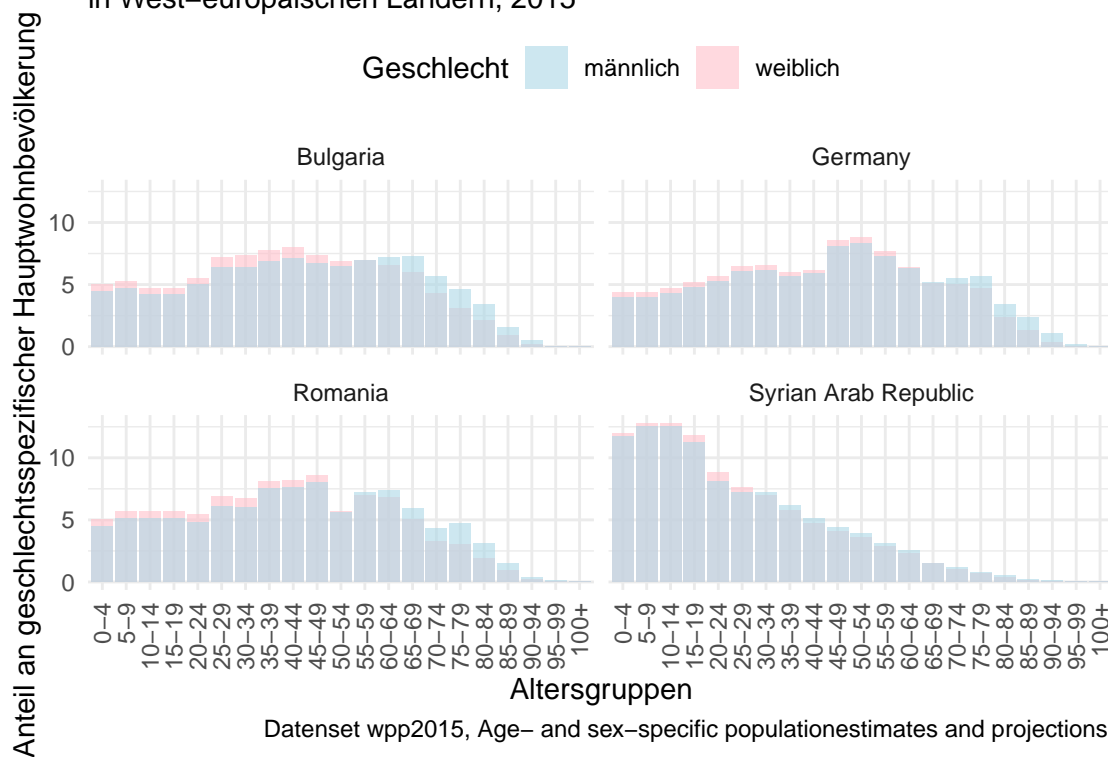
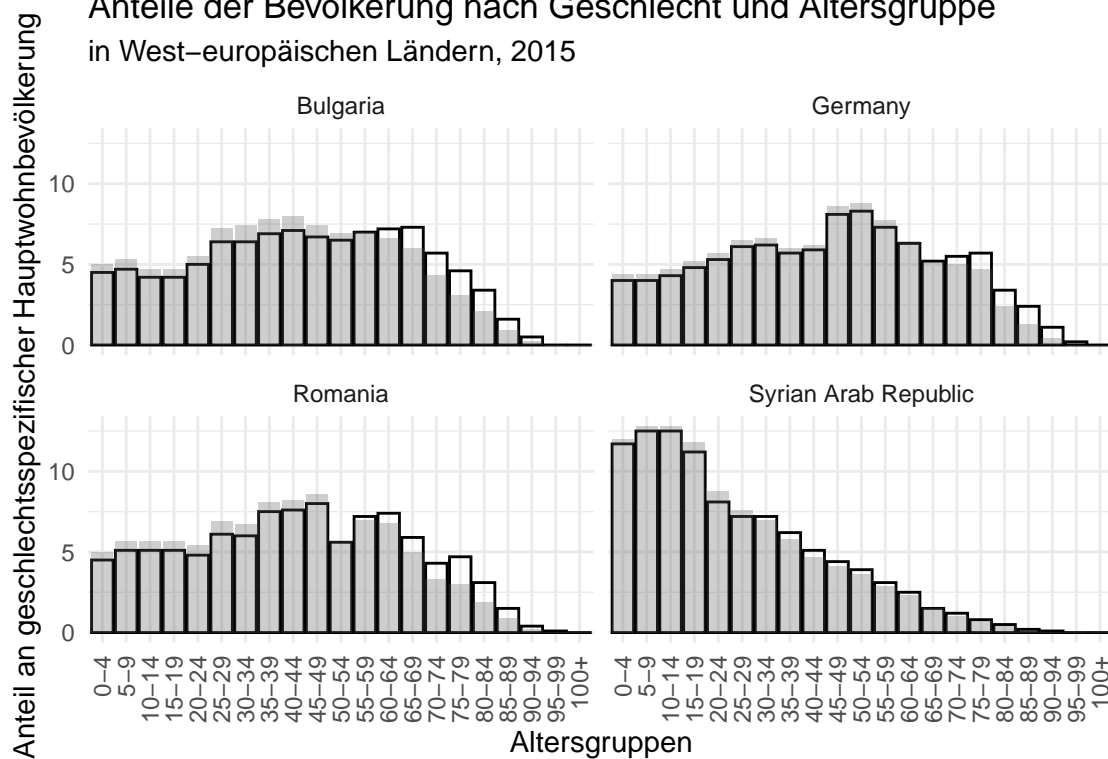


Abb. 01

Anteile der Bevölkerung nach Geschlecht und Altersgruppe
in West-europäischen Ländern, 2015



Datenset wpp2015, Age- and sex-specific population estimates and projections