3 Krebsfälle in Deutschland

3 Krebsfälle in Deutschland

Alle benötigten Libraries aktivieren

```
library(ggplot2)
library(dplyr)
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(tibble)
```

3.1 Einlesen der Daten und Data Preprocessing

```
krebs_df = read.csv2(
  "Krebsdaten.csv",
  header = FALSE,
  skip = 1, # skips the first row
  stringsAsFactors = FALSE
  # FALSE because we will convert all data types manually later
)
```

```
head(krebs_df, 5)
```

```
##     V1                              V2       V3           V4      V5 V6
## 1 2004 Krebs gesamt (C00-C97 ohne C44) weiblich       0 - 44  69.417 NA
## 2 2004 Krebs gesamt (C00-C97 ohne C44) weiblich      45 - 54  92.221 NA
## 3 2004 Krebs gesamt (C00-C97 ohne C44) weiblich      55 - 64 149.642 NA
## 4 2004 Krebs gesamt (C00-C97 ohne C44) weiblich      65 - 74 174.556 NA
## 5 2004 Krebs gesamt (C00-C97 ohne C44) weiblich 75 und älter 188.884 NA
```

Letzte leere Spalte entfernen

```
krebs_df = krebs_df[, 1:5]
```

Spalten Namen benennen

```
colnames(krebs_df) = c(
  "Jahr",
  "Diagnose",
  "Geschlecht",
  "Altersgruppe",
  "Prävalenz"
)
```

```
head(krebs_df, 5)
```

```
##   Jahr                         Diagnose Geschlecht Altersgruppe Prävalenz
## 1 2004 Krebs gesamt (C00-C97 ohne C44)   weiblich       0 - 44    69.417
## 2 2004 Krebs gesamt (C00-C97 ohne C44)   weiblich      45 - 54    92.221
## 3 2004 Krebs gesamt (C00-C97 ohne C44)   weiblich      55 - 64   149.642
## 4 2004 Krebs gesamt (C00-C97 ohne C44)   weiblich      65 - 74   174.556
## 5 2004 Krebs gesamt (C00-C97 ohne C44)   weiblich 75 und älter   188.884
```

```
dim(krebs_df)
```

```
## [1] 6300    5
```

Data Frame in ein Tibble umwandeln

```
krebs_tib = as_tibble(krebs_df)

head(krebs_tib, 5)
```

```
## # A tibble: 5 x 5
##    Jahr Diagnose                         Geschlecht Altersgruppe Prävalenz
##   <int> <chr>                            <chr>      <chr>        <chr>
## 1  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   0 - 44       69.417
## 2  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   45 - 54      92.221
## 3  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   55 - 64      149.642
## 4  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   65 - 74      174.556
## 5  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   75 und älter 188.884
```

Datentypen aller Spalten überprüfen

```
sapply(krebs_tib, class)
```

```
##         Jahr     Diagnose   Geschlecht Altersgruppe    Prävalenz
##    "integer"  "character"  "character"  "character"  "character"
```

Spalten vor der Data Type Conversion genauer analysieren

For each column, we should check all unique values so that we can see if there are any Null values, etc... For the `Prävalenz` column, we need to do some additional analysis steps.

1. Werte der Spalte `Jahr` checken

```
sort(unique(krebs_tib$Jahr))
```

```
##  [1] 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
## [16] 2019 2020 2021 2022 2023 2024
```

2. Werte der Spalte `Diagnose` checken

```r
unique(krebs_tib$Diagnose)
```

```
##  [1] "Krebs gesamt (C00-C97 ohne C44)"
##  [2] "Mundhöhle und Rachen (C00-C14)"
##  [3] "Speiseröhre (C15)"
##  [4] "Magen (C16)"
##  [5] "Dünndarm (C17)"
##  [6] "Darm (C18-C20)"
##  [7] "Anus u. Analkanal (C21)"
##  [8] "Leber (C22)"
##  [9] "Gallenblase und Gallenwege (C23-C24)"
## [10] "Bauchspeicheldrüse (C25)"
## [11] "Kehlkopf (C32)"
## [12] "Lunge (C33-C34)"
## [13] "Malignes Melanom der Haut (C43)"
## [14] "Mesotheliom (C45)"
## [15] "Weichteilgewebe ohne Mesotheliom (C46-C49)"
## [16] "Brustdrüse (C50)"
## [17] "Vulva (C51)"
## [18] "Gebärmutterhals (C53)"
## [19] "Gebärmutterkörper (C54-C55)"
## [20] "Eierstöcke (C56)"
## [21] "Prostata (C61)"
## [22] "Hoden (C62)"
## [23] "Niere (C64)"
## [24] "Harnblase (C67)"
## [25] "Zentrales Nervensystem (C70-C72)"
## [26] "Schilddrüse (C73)"
## [27] "Morbus Hodgkin (C81)"
## [28] "Non-Hodgkin-Lymphome (C82-C88)"
## [29] "Multiples Myelom (C90)"
## [30] "Leukämien (C91-C95)"
```

3. Werte der Spalte `Geschlecht` checken

```r
unique(krebs_tib$Geschlecht)
```

```
## [1] "weiblich" "männlich"
```

4. Werte der Spalte `Altersgruppe` checken

```r
unique(krebs_tib$Altersgruppe)
```

```
## [1] "0 - 44"       "45 - 54"      "55 - 64"      "65 - 74"      "75 und älter"
```

5. Werte der Spalte `Prävalenz` checken

We should manually check if there are any non-numeric values. We can simply convert the entire column to numeric and anything that is not able to be converted will automatically be converted to `NA`. Then we can just count those `NA`s to see how many they are. *(It's important to know how many they are so that later we can either fill them or delete the entire row, etc...)*

```r
# Original values
krebs_prävalenz = krebs_tib$Prävalenz

# Converting to numeric
converted = as.numeric(gsub("\\.", "", krebs_prävalenz))
```

```
## Warning: NAs durch Umwandlung erzeugt
```

```
# Find all values that could not be converted
invalid_values = krebs_prävalenz[is.na(converted)]

# Show all the non-numeric values that were not converted
unique(invalid_values)
```

```
## [1] "x"
```

```
length(invalid_values)
```

```
## [1] 630
```

Deleting those rows will lead to a loss of information, so we will keep them.

```
colSums(is.na(krebs_tib))
```

Fehlende Werte pro Spalte prüfen (vor der Data Type Conversion)

```
##         Jahr     Diagnose   Geschlecht Altersgruppe    Prävalenz
##            0            0            0            0            0
```

Convert Data Types

```
krebs_tib = krebs_tib |>
  mutate(
    Geschlecht = as.factor(Geschlecht),
    Altersgruppe = as.factor(Altersgruppe),
    Prävalenz = as.numeric(gsub("\\.", "", Prävalenz))
    # Removes the '.' from '600.000' => '600000'
  )
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Prävalenz = as.numeric(gsub("\\.", "", Prävalenz))`.
## Caused by warning:
## ! NAs durch Umwandlung erzeugt
```

```
sapply(krebs_tib, class)
```

Datentypen nach der Conversion checken

```
##         Jahr     Diagnose   Geschlecht Altersgruppe    Prävalenz
##    "integer"  "character"     "factor"     "factor"    "numeric"
```

Noch einmal auf fehlende Werte checken *We see that these 630 missing values in* ***Prävalenz*** *are the values we analysed above.*

```
colSums(is.na(krebs_tib))
```

```
##         Jahr     Diagnose   Geschlecht Altersgruppe    Prävalenz
##            0            0            0            0          630
```

Warum enthält der Datensatz x-Werte?

Bei der Datenaufbereitung wurden **630 Einträge (Cells) mit dem Wert `x`** in der Spalte `Prävalenz` gefunden.

Laut der Datenbank kennzeichnet `x` eine **nicht sinnvolle Merkmalskombination**, also zum Beispiel bei geschlechtsspezifischen Krebsdiagnosen.

Das sind alle Kombinationen:

Diagnose	Geschlecht	Anzahl `x`
Eierstöcke (C56)	männlich	105
Gebärmutterhals (C53)	männlich	105
Gebärmutterkörper (C54–C55)	männlich	105
Vulva (C51)	männlich	105
Hoden (C62)	weiblich	105
Prostata (C61)	weiblich	105

Beispiel:

Prostatakrebs ist in dieser Statistik eine männliche Krebsdiagnose. Deshalb wird für die Kombination `Prostata (C61)` und `weiblich` keine numerische Prävalenz angegeben, sondern `x`.

Die Werte sind also **absichtlich nicht angegeben**. Es handelt sich nicht um zufällig fehlende Messwerte. Also macht es auch keinen Sinn die Werte zu **imputieren**.

Lösung: Geben Sie die ersten 10 Zeilen und 7 Spalten als Tabelle aus

```
head(krebs_tib, 10)
```

```
## # A tibble: 10 x 5
##     Jahr Diagnose                          Geschlecht Altersgruppe Prävalenz
##    <int> <chr>                             <fct>      <fct>            <dbl>
##  1  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   0 - 44           69417
##  2  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   45 - 54          92221
##  3  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   55 - 64         149642
##  4  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   65 - 74         174556
##  5  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   75 und älter    188884
##  6  2004 Krebs gesamt (C00-C97 ohne C44) männlich   0 - 44           45244
##  7  2004 Krebs gesamt (C00-C97 ohne C44) männlich   45 - 54          54175
##  8  2004 Krebs gesamt (C00-C97 ohne C44) männlich   55 - 64         156830
##  9  2004 Krebs gesamt (C00-C97 ohne C44) männlich   65 - 74         244275
## 10  2004 Krebs gesamt (C00-C97 ohne C44) männlich   75 und älter    167108
```

Note

Im originalen Datensatz haben wir nur 5 Spalten und nicht 7.

3.2 Übersicht aller Krebsarten

Daten für die Visualization vorbereiten

```
präv_krebs = krebs_tib |>
  filter(Jahr == 2024 & Altersgruppe == "55 - 64" &
           Diagnose != "Krebs gesamt (C00-C97 ohne C44)") |>
```

```
  drop_na(Prävalenz)

dim(präv_krebs)
```

```
## [1] 52  5
```

```
head(präv_krebs, 5)
```

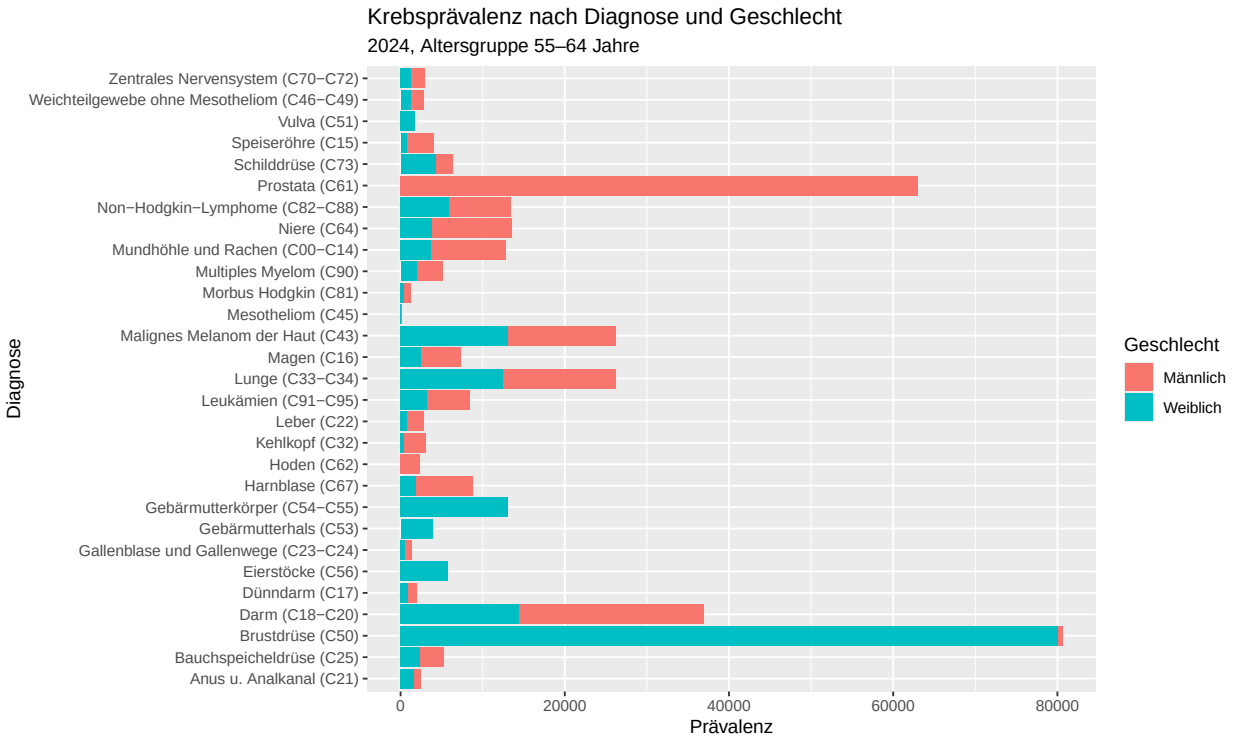
```
## # A tibble: 5 x 5
##    Jahr Diagnose                          Geschlecht Altersgruppe Prävalenz
##   <int> <chr>                             <fct>      <fct>            <dbl>
## 1  2024 Mundhöhle und Rachen (C00-C14) weiblich   55 - 64           3777
## 2  2024 Mundhöhle und Rachen (C00-C14) männlich   55 - 64           9055
## 3  2024 Speiseröhre (C15)              weiblich   55 - 64            727
## 4  2024 Speiseröhre (C15)              männlich   55 - 64           3273
## 5  2024 Magen (C16)                    weiblich   55 - 64           2498
```

Note

We filter out "Krebs gesamt" because it isn't an actual Diagnosis.

Lösung: Balkendiagramm erstellen

```
ggplot(präv_krebs) +
  geom_bar(aes(y = Diagnose, x = Prävalenz, fill = Geschlecht), stat = "identity") +
  labs(
    title = "Krebsprävalenz nach Diagnose und Geschlecht",
    subtitle = "2024, Altersgruppe 55-64 Jahre"
  ) +
  scale_fill_discrete(
    labels = c("Männlich", "Weiblich")
  )
```



Begründung unserer Motivation für den Plot Design

Für die Position der Bars, haben wir den default value `stacked` benutzt, da es die unterschiedlichen Geschlechte schon klar darstellt. Eine andere Option wäre `dodge`, aber da der Plot schon ziemlich groß ist, würde es ein bisschen chaotisch sein. Zusatätzlich, haben wir die Legend Labels überschrieben und titel/ subtitel hinzugefügt.

3.3 Prävalenz nach Altergruppen und Geschlecht

Dataset kurz für einen Überblick zeigen

```
head(krebs_tib, 100)
```

```
## # A tibble: 100 x 5
##     Jahr Diagnose                          Geschlecht Altersgruppe Prävalenz
##    <int> <chr>                             <fct>      <fct>            <dbl>
##  1  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   0 - 44           69417
##  2  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   45 - 54          92221
##  3  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   55 - 64         149642
##  4  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   65 - 74         174556
##  5  2004 Krebs gesamt (C00-C97 ohne C44) weiblich   75 und älter    188884
##  6  2004 Krebs gesamt (C00-C97 ohne C44) männlich   0 - 44           45244
##  7  2004 Krebs gesamt (C00-C97 ohne C44) männlich   45 - 54          54175
##  8  2004 Krebs gesamt (C00-C97 ohne C44) männlich   55 - 64         156830
##  9  2004 Krebs gesamt (C00-C97 ohne C44) männlich   65 - 74         244275
## 10  2004 Krebs gesamt (C00-C97 ohne C44) männlich   75 und älter    167108
## # i 90 more rows
```

Daten für die Visualization vorbereiten

```
präv_krebs_2 = krebs_tib |>
  filter(Diagnose == "Krebs gesamt (C00-C97 ohne C44)") |>
  # We only keep the 'Krebs gesamt' Diagnosen
  mutate(
    Altersstufe = ifelse( # Add a new column Altersstufe consisting of
      Altersgruppe == "0 - 44" | Altersgruppe == "45 - 54",
      # values either `bis 54` or `ab 55` for each
      "bis 54", # existing Altersgruppe
      "ab 55"
    )
  ) |>
  group_by(Jahr, Geschlecht, Altersstufe) |>
  summarize("Prävalenz Altersstufe" = sum(Prävalenz))
```

```
## `summarise()` has regrouped the output.
## i Summaries were computed grouped by Jahr, Geschlecht, and Altersstufe.
## i Output is grouped by Jahr and Geschlecht.
## i Use `summarise(.groups = "drop_last")` to silence this message.
## i Use `summarise(.by = c(Jahr, Geschlecht, Altersstufe))` for per-operation
##   grouping (`?dplyr::dplyr_by`) instead.
```

```
  # Add a new column Prävalenz Altersstufe
  # containing the summed Prävalenz per Altersstufe

dim(präv_krebs_2)
```

```
## [1] 84  4
```

```
head(präv_krebs_2, 100)
```

```
## # A tibble: 84 x 4
## # Groups:   Jahr, Geschlecht [42]
##     Jahr Geschlecht Altersstufe `Prävalenz Altersstufe`
##    <int> <fct>      <chr>                         <dbl>
##  1  2004 männlich   ab 55                        568213
##  2  2004 männlich   bis 54                        99419
##  3  2004 weiblich   ab 55                        513082
##  4  2004 weiblich   bis 54                       161638
##  5  2005 männlich   ab 55                        595420
##  6  2005 männlich   bis 54                       101820
##  7  2005 weiblich   ab 55                        522429
##  8  2005 weiblich   bis 54                       163736
##  9  2006 männlich   ab 55                        627969
## 10  2006 männlich   bis 54                       105856
## # i 74 more rows
```

Here we don't have to drop `NA`*s because we are only using the* `Diagnose` *"Krebs gesamt" which contains all the summed* `Prävalenzen` *of each* `Diagnose`*, so we wouldn't get any 0 values/ Null values.*

Lösung: 2 Liniendiagramme erstellen

```
ggplot(präv_krebs_2, aes(Jahr, `Prävalenz Altersstufe`, color = Altersstufe)) +
  geom_line(linewidth = 1.5) +
  facet_wrap(~ Geschlecht) +
```

```r
  labs(
    title = "Krebsprävalenz nach Altersstufe und Geschlecht",
    subtitle = "2004 bis 2024, Alterstufen: bis 54 & ab 55"
  ) +
  scale_y_continuous(
    labels = scales::label_number(),
    limits = c(0, NA)
  )
```

