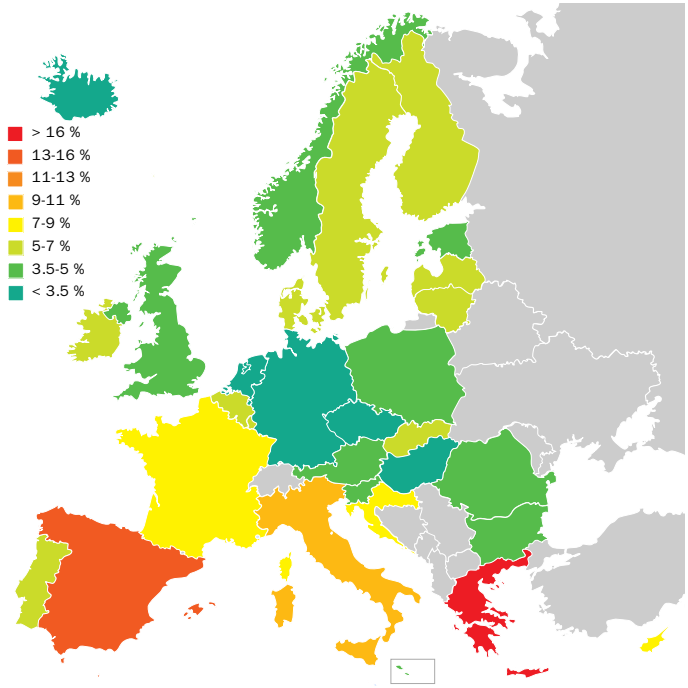
Case-Study zur Arbeitslosigkeit in Deutschland

Ziel der Case-Study

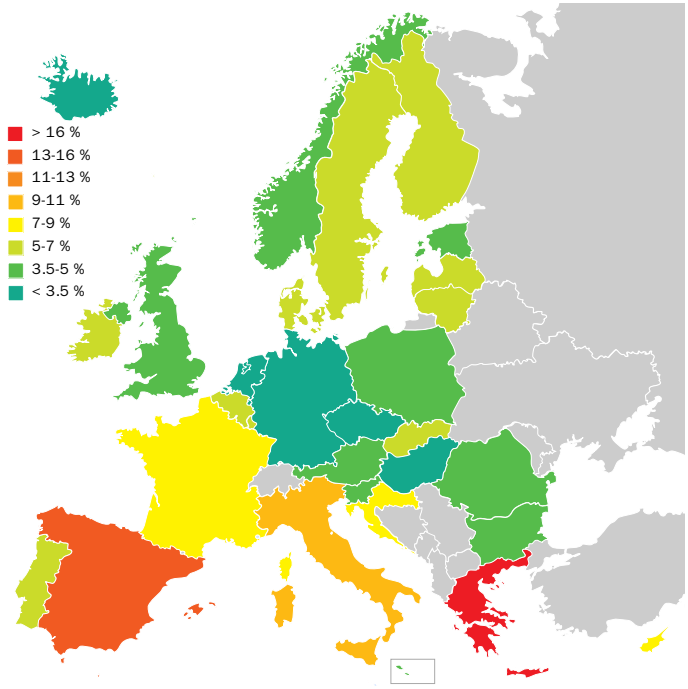
Deutschland hat europaweit eine der niedrigsten Arbeitslosenquoten:



Quelle: Von Heycci - Daten von Eurostat, CC BY-SA 2.5

Ziel der Case-Study

Deutschland hat europaweit eine der niedrigsten Arbeitslosenquoten:



Quelle: Von Heycci - Daten von Eurostat, CC BY-SA 2.5

> Doch gilt dies für alle Regionen in Deutschland? Warum ist die Arbeitslosenquote in manchen Regionen höher als in anderen?

Dem wollen wir in dieser Case-Study auf den Grund gehen.

Ziele der Case Study

Diese Case-Study besteht aus **mehreren Teilen** und wird Sie durch die komplette Vorlesung als **konkretes Anschauungsobjekt** begleiten.

Diese Case-Study dient als:

- konkretes und umfangreiches Beispiel für ein Projekt
- ökonomische und geographische Kenntnisse über Deutschland erhalten
- Beispiel wie statistische und programmiertechnische Kenntnisse in der empirischen Arbeit eingesetzt werden können

Datensätze herunterladen

Ersten Teil der Case Study

- Daten einlesen
- Daten bearbeiten und in eine geeignete Form bringen (`tidy`)

Anwenden auf

- Daten zur Arbeitslosenstatistik
- Daten zur Verschuldung einzelner Landkreise bzw. Gemeinden
- Daten zum BIP

Wichtig für die Datenbeschaffung

- Zuverlässige und qualitativ hochwertige Datenquellen ausfinding machen
- Automatisierten Download programmieren
- Einlesen, verarbeiten und zusammenführen verschiedener Datensätze in R

Wichtig für die Datenbeschaffung

- Zuverlässige und qualitativ hochwertige Datenquellen ausfinding machen
- Automatisierten Download programmieren
- Einlesen, verarbeiten und zusammenführen verschiedener Datensätze in R

Verbindung zum 2. RTutor Problem Set:

- **Im Problem Set:** Kennzahlen zu verschiedenen Ländern der europäischen Union
- **Hier:** Kennzahlen innerhalb Deutschlands

Wichtig für die Datenbeschaffung

- Zuverlässige und qualitativ hochwertige Datenquellen ausfinding machen
- Automatisierten Download programmieren
- Einlesen, verarbeiten und zusammenführen verschiedener Datensätze in R

Verbindung zum 2. RTutor Problem Set:

- **Im Problem Set:** Kennzahlen zu verschiedenen Ländern der europäischen Union
- **Hier:** Kennzahlen innerhalb Deutschlands

Sowohl in der Case-Study als auch in den RTutor Problem Sets treffen Sie auf konkrete Probleme, die Sie mit ihren Kenntnissen aus der Vorlesung lösen sollen.

Daten beschaffen

Woher beziehen wir unsere Informationen?

Daten beschaffen

Woher beziehen wir unsere Informationen?

- Die Informationen über die Verschuldung der **Gemeinden** finden wir auf den Seiten des Statistischen Bundesamts im Report: Integrierte Schulden der Gemeinden und Gemeindeverbände.
- Die Informationen zur Arbeitslosigkeit auf **Verwaltungsgemeinschaftsebene** finden wir auf den Seiten der Bundesagentur für Arbeit.
- Die Informationen zum BIP auf **Landkreisebene** finden wir auf den Seiten der Statistischen Ämter des Bundes und der Länder.

Daten beschaffen

Woher beziehen wir unsere Informationen?

- Die Informationen über die Verschuldung der **Gemeinden** finden wir auf den Seiten des Statistischen Bundesamts im Report: Integrierte Schulden der Gemeinden und Gemeindeverbände.
- Die Informationen zur Arbeitslosigkeit auf **Verwaltungsgemeinschaftsebene** finden wir auf den Seiten der Bundesagentur für Arbeit.
- Die Informationen zum BIP auf **Landkreisebene** finden wir auf den Seiten der Statistischen Ämter des Bundes und der Länder.

Zuverlässige und qualitativ hochwertige Datenquellen ausfinding machen ✓

Nötige Pakete laden

```
library(readxl)
library(skimr)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.5
## ✔ tibble  3.1.7      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Daten herunterladen

- Daten können von URLs mit Befehlen aus den Paketen `readxl` und `readr` direkt eingelesen werden
  - Für Text und Excel-Dateien
- Allerdings, wenn URL nicht mehr verfügbar, was dann?
  - Daten immer mit `download.file()` herunterladen und in einem Unterordner `data` abspeichern!

Daten herunterladen

- Daten können von URLs mit Befehlen aus den Paketen `readxl` und `readr` direkt eingelesen werden
  - Für Text und Excel-Dateien
- Allerdings, wenn URL nicht mehr verfügbar, was dann?
  - Daten immer mit `download.file()` herunterladen und in einem Unterordner `data` abspeichern!

Automatisierten Download programmieren (wird in der ausformulierten Case-Study gemacht) (✓)

Daten herunterladen

- Daten können von URLs mit Befehlen aus den Paketen `readxl` und `readr` direkt eingelesen werden
  - Für Text und Excel-Dateien
- Allerdings, wenn URL nicht mehr verfügbar, was dann?
  - Daten immer mit `download.file()` herunterladen und in einem Unterordner `data` abspeichern!

Automatisierten Download programmieren (wird in der ausformulierten Case-Study gemacht) (✓)

Wir haben die Daten bereits im Github Repository `case-study-germany` heruntergeladen und abgespeichert. Klonen Sie dieses Repository von Github auf ihren PC!

Klonen Sie unsere Github Seite

- Gehen Sie auf die Github Seite des Projektkurses
- Klicken Sie auf des grünen "Code" Button
- Kopieren Sie sich die angezeigte HTTPS
- Gehen Sie in Github Desktop und fügen dort die kopierte HTTPS in "Clone a repository" -> "URL"

Hier eine Step-by-Step Anleitung

Wenn Sie zu Beginn der Woche in Github Desktop auf "Pull" klicken werden alle Vorlesungsinhalte automatisch aktualisiert, d.h. alle Vorlesungsfolien, die Case-Study, Tutorials etc.!

05:00

Daten einlesen

Unterschiedliche Tabellenblätter, welches ist für uns interessant?

```
# Öffnen des ZIP-Archivs
alo_name <- as.character(unzip("../case-study/data/Arbeitslose_2017.xlsx.zip", list = TRUE)$Name)
unzip("../case-study/data/Arbeitslose_2017.xlsx.zip", alo_name)
excel_sheets(alo_name)
```

```
##  [1] "Deckblatt"           "Impressum"           "Inhalt"
##  [4] "Hinweis"             "5"                   "6"
##  [7] "7"                   "8"                   "9"
## [10] "Statistik-Infoseite"
```

Daten einlesen

Unterschiedliche Tabellenblätter, welches ist für uns interessant?

```
# Öffnen des ZIP-Archivs
alo_name <- as.character(unzip("../case-study/data/Arbeitslose_2017.xlsx.zip", list = TRUE)$Name)
unzip("../case-study/data/Arbeitslose_2017.xlsx.zip", alo_name)
excel_sheets(alo_name)
```

```
##  [1] "Deckblatt"           "Impressum"           "Inhalt"
##  [4] "Hinweis"             "5"                   "6"
##  [7] "7"                   "8"                   "9"
## [10] "Statistik-Infoseite"
```

Vermutung: Durch Tabellenblatt "Inhalt" könnten wir schlauer werden

```r
alo_inhalt <- read_xlsx(alo_name, sheet = "Inhalt")
head(alo_inhalt, 15)
```

```
## # A tibble: 15 × 4
##    ...1                                ...2                ...3  Arbeitslose nach …[1]
##    <chr>                               <chr>               <lgl> <chr>
##  1 <NA>                                <NA>                NA    <NA>
##  2 Inhaltsverzeichnis                  <NA>                NA    <NA>
##  3 <NA>                                <NA>                NA    <NA>
##  4 Berichtsmonat: Jahreszahlen 2017    <NA>                NA    <NA>
##  5 <NA>                                <NA>                NA    <NA>
##  6 Arbeitslose nach Gemeinden          <NA>                NA    <NA>
##  7 <NA>                                Impressum           NA    2
##  8 <NA>                                Inhaltsverzeichnis  NA    3
##  9 <NA>                                Hinweise            NA    4
## 10 Zugang                              <NA>                NA    <NA>
## 11 <NA>                                Insgesamt           NA    5
## 12 Bestand                             <NA>                NA    <NA>
## 13 <NA>                                Insgesamt           NA    6
## 14 <NA>                                Männer              NA    7
## 15 <NA>                                Frauen              NA    8
## # … with abbreviated variable name
## #   [1]`Arbeitslose nach Gemeinden     Jahreszahlen 2017`
```

```r
alo_inhalt <- read_xlsx(alo_name, sheet = "Inhalt")
head(alo_inhalt, 15)
```

```
## # A tibble: 15 x 4
##    ...1                  ...2          ...3  `Arbeitslose nach Gemeinden     Ja…
##    <chr>                 <chr>         <lgl> <chr>
##  1 <NA>                  <NA>          NA    <NA>
##  2 Inhaltsverzeichnis    <NA>          NA    <NA>
##  3 <NA>                  <NA>          NA    <NA>
##  4 Berichtsmonat: Jahre… <NA>          NA    <NA>
##  5 <NA>                  <NA>          NA    <NA>
##  6 Arbeitslose nach Gem… <NA>          NA    <NA>
##  7 <NA>                  Impressum     NA    2
##  8 <NA>                  Inhaltsverze… NA    3
##  9 <NA>                  Hinweise      NA    4
## 10 Zugang                <NA>          NA    <NA>
## 11 <NA>                  Insgesamt     NA    5
## 12 Bestand               <NA>          NA    <NA>
## 13 <NA>                  Insgesamt     NA    6
## 14 <NA>                  Männer        NA    7
## 15 <NA>                  Frauen        NA    8
```

```
alo_inhalt <- read_xlsx(alo_name, sheet = "Inhalt")
head(alo_inhalt, 15)
```

```
## # A tibble: 15 x 4
##    ...1                  ...2          ...3  `Arbeitslose nach Gemeinden     Ja…
##    <chr>                 <chr>         <lgl> <chr>
##  1 <NA>                  <NA>          NA    <NA>
##  2 Inhaltsverzeichnis    <NA>          NA    <NA>
##  3 <NA>                  <NA>          NA    <NA>
##  4 Berichtsmonat: Jahre… <NA>          NA    <NA>
##  5 <NA>                  <NA>          NA    <NA>
##  6 Arbeitslose nach Gem… <NA>          NA    <NA>
##  7 <NA>                  Impressum     NA    2
##  8 <NA>                  Inhaltsverze… NA    3
##  9 <NA>                  Hinweise      NA    4
## 10 Zugang                <NA>          NA    <NA>
## 11 <NA>                  Insgesamt     NA    5
## 12 Bestand               <NA>          NA    <NA>
## 13 <NA>                  Insgesamt     NA    6
## 14 <NA>                  Männer        NA    7
## 15 <NA>                  Frauen        NA    8
```

Alternative: Schauen Sie sich die Excel-Datei in Excel oder LibreOffice an und entscheiden Sie dann, welches Tabellenblatt Sie einlesen möchten.

Spezifizieren welche Spalten eingelesen werden sollen

Welche Information benötigen wir aus der Tabelle

- Die Anzahl aller Arbeitslosen pro Gemeinde (d.h. SGB II und III gemeinsam)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis (z.B. nur SGB II)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis und ein bestimmtes Alter (z.B. SGB II alle unter 25 Jahre)

Spezifizieren welche Spalten eingelesen werden sollen

Welche Information benötigen wir aus der Tabelle

- Die Anzahl aller Arbeitslosen pro Gemeinde (d.h. SGB II und III gemeinsam)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis (z.B. nur SGB II)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis und ein bestimmtes Alter (z.B. SGB II alle unter 25 Jahre)

Spezifizieren welche Spalten eingelesen werden sollen

Welche Information benötigen wir aus der Tabelle

- Die Anzahl aller Arbeitslosen pro Gemeinde (d.h. SGB II und III gemeinsam)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis (z.B. nur SGB II)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis und ein bestimmtes Alter (z.B. SGB II alle unter 25 Jahre)

> Was ist hier eine Beobachtung?

Spezifizieren welche Spalten eingelesen werden sollen

Welche Information benötigen wir aus der Tabelle

- Die Anzahl aller Arbeitslosen pro Gemeinde (d.h. SGB II und III gemeinsam)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis (z.B. nur SGB II)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis und ein bestimmtes Alter (z.B. SGB II alle unter 25 Jahre)

> Was ist hier eine Beobachtung?

Weiterhin benötigen wir noch den `Schlüssel` und den Gemeindenamen.

Wie können wir die von uns benötigte Information möglichst einfach extrahieren?

Spezifizieren welche Spalten eingelesen werden sollen

Welche Information benötigen wir aus der Tabelle

- Die Anzahl aller Arbeitslosen pro Gemeinde (d.h. SGB II und III gemeinsam)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis (z.B. nur SGB II)
- Die Anzahl der Arbeitslosen pro Gemeinde für einen bestimmten Rechtskreis und ein bestimmtes Alter (z.B. SGB II alle unter 25 Jahre)

> Was ist hier eine Beobachtung?

Weiterhin benötigen wir noch den `Schlüssel` und den Gemeindenamen.

Wie können wir die von uns benötigte Information möglichst einfach extrahieren?

- Der einfachste Weg: Die ersten acht Zeilen abzuschneiden und die Daten erst ab dort einzulesen.
- Anschließend behalten wir nur die ersten 3 Spalten

```
alo_skip <- read_xlsx(alo_name, sheet = "6", skip =
```

```
alo_skip <- read_xlsx(alo_name, sheet = "6", skip =

alo_skip
```

```
## # A tibble: 11,346 × 32
##    Schlüssel Gemei…¹    `1`   `2`    `3`    `4`    `5`    `6`   `7`   `8`    `
##    <chr>     <chr>    <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl>  <db
##  1 <NA>      Deutsc… 2.53e6 47840 230694 839545 529002 655249 14022 71133 1360
##  2 01        Schles… 9.24e4  2072   9703  30138  18788  18688   437  2297   34
##  3 01001000  Flensb… 4.51e3   108    566   1120    654    983    18   141    1
##  4 01002000  Kiel, … 1.23e4   220   1100   3306   1842   3103    54   294    5
##  5 01003000  Lübeck… 9.69e3   180    836   2902   1667   2050    41   204    3
##  6 01004000  Neumün… 3.84e3   105    440   1134    685    724    10    45    1
##  7 01051     Dithma… 4.63e3   147    634   1402    826    791    27   141    1
##  8 01051001  Albers… 1.5 e2     5     23     43     23     24     0     4
##  9 01051002  Arkebek 5   e0     0      1      1      1      2    NA    NA
## 10 01051003  Averlak 1.5 e1     0      3      4      4      1    NA     0
## # … with 11,336 more rows, 21 more variables: `10` <dbl>, `11` <dbl>,
## #   `12` <dbl>, `13` <dbl>, `14` <dbl>, `15` <dbl>, `16` <dbl>, `17` <dbl>,
## #   `18` <dbl>, `19` <dbl>, `20` <dbl>, `21` <dbl>, `22` <dbl>, `23` <dbl>,
## #   `24` <dbl>, `25` <dbl>, `26` <dbl>, `27` <dbl>, `28` <dbl>, `29` <dbl>,
## #   `30` <dbl>, and abbreviated variable name ¹Gemeinde
```

```r
alo_skip <- read_xlsx(alo_name, sheet = "6", skip =

alo_skip %>%
  select(c(`Schlüssel`, Gemeinde, `1`))
```

```
## # A tibble: 11,346 × 3
##    Schlüssel Gemeinde                    `1`
##    <chr>     <chr>                     <dbl>
##  1 <NA>      Deutschland             2532837
##  2 01        Schleswig-Holstein        92434
##  3 01001000  Flensburg, Stadt           4512
##  4 01002000  Kiel, Landeshauptstadt    12345
##  5 01003000  Lübeck, Hansestadt         9692
##  6 01004000  Neumünster, Stadt          3836
##  7 01051     Dithmarschen               4628
##  8 01051001  Albersdorf                  150
##  9 01051002  Arkebek                       5
## 10 01051003  Averlak                      15
## # … with 11,336 more rows
```

```r
alo_skip <- read_xlsx(alo_name, sheet = "6", skip =

alo_skip %>%
  select(c(`Schlüssel`, Gemeinde, `1`)) %>%
  rename(Regionalschluessel = `Schlüssel`,
         alo = `1`)
```

```
## # A tibble: 11,346 × 3
##    Regionalschluessel Gemeinde                  alo
##    <chr>              <chr>                   <dbl>
##  1 <NA>               Deutschland           2532837
##  2 01                 Schleswig-Holstein      92434
##  3 01001000           Flensburg, Stadt         4512
##  4 01002000           Kiel, Landeshauptstadt  12345
##  5 01003000           Lübeck, Hansestadt       9692
##  6 01004000           Neumünster, Stadt        3836
##  7 01051              Dithmarschen             4628
##  8 01051001           Albersdorf                150
##  9 01051002           Arkebek                     5
## 10 01051003           Averlak                    15
## # … with 11,336 more rows
```

```
alo_skip <- read_xlsx(alo_name, sheet = "6", skip =

alo_skip %>%
  select(c(`Schlüssel`, Gemeinde, `1`)) %>%
  rename(Regionalschluessel = `Schlüssel`,
         alo = `1`) %>%
  filter(!is.na(alo) & Gemeinde!= "Deutschland")
```

```
## # A tibble: 11,343 × 3
##    Regionalschluessel Gemeinde                 alo
##    <chr>              <chr>                  <dbl>
##  1 01                 Schleswig-Holstein     92434
##  2 01001000           Flensburg, Stadt        4512
##  3 01002000           Kiel, Landeshauptstadt 12345
##  4 01003000           Lübeck, Hansestadt      9692
##  5 01004000           Neumünster, Stadt       3836
##  6 01051              Dithmarschen            4628
##  7 01051001           Albersdorf               150
##  8 01051002           Arkebek                    5
##  9 01051003           Averlak                   15
## 10 01051004           Bargenstedt               17
## # … with 11,333 more rows
```

```
#Abspeichern als Datensatz data_alo

data_alo <- alo_skip %>%
  select(c(`Schlüssel`, Gemeinde, `1`)) %>%
  rename(Regionalschluessel = `Schlüssel`,
         alo = `1`) %>%
  filter(!is.na(alo) & Gemeinde!= "Deutschland")
```

Konsistenzcheck

- Machen die Angaben Sinn und sind die Daten in sich konsistent?
- Externe Datenquelle suchen und intern auf konsistenz prüfen.
- Informationen aggregieren und mit anderen Quellen vergleichen

Konsistenzcheck

- Machen die Angaben Sinn und sind die Daten in sich konsistent?
- Externe Datenquelle suchen und intern auf konsistenz prüfen.
- Informationen aggregieren und mit anderen Quellen vergleichen
- Zunächst: Anzahl an Arbeitslosen für jedes **Bundesland** in 2017.
  - zweistelligen `Regionalschluessel`
  - "Buchstaben" für jeden `Regionalschluessel` zählen (`nchar()` (number of characters))
- **Alternative Datenquelle:** Die Anzahl der Arbeitslosen für das Jahr 2017 unterteilt nach Ländern der Arbeitsagentur
  - Wichtig: Tabellenblatt 8

```
data_alo
```

```
## # A tibble: 11,343 × 3
##    Regionalschluessel Gemeinde                 alo
##    <chr>              <chr>                  <dbl>
##  1 01                 Schleswig-Holstein     92434
##  2 01001000           Flensburg, Stadt        4512
##  3 01002000           Kiel, Landeshauptstadt 12345
##  4 01003000           Lübeck, Hansestadt      9692
##  5 01004000           Neumünster, Stadt       3836
##  6 01051              Dithmarschen            4628
##  7 01051001           Albersdorf               150
##  8 01051002           Arkebek                    5
##  9 01051003           Averlak                   15
## 10 01051004           Bargenstedt               17
## # … with 11,333 more rows
```

```r
data_alo %>%
  filter(nchar(Regionalschluessel) == 2)
```

```
## # A tibble: 16 × 3
##    Regionalschluessel Gemeinde                 alo
##    <chr>              <chr>                  <dbl>
##  1 01                 Schleswig-Holstein      92434
##  2 02                 Hamburg                 69248
##  3 03                 Niedersachsen          244260
##  4 04                 Bremen                  35687
##  5 05                 Nordrhein-Westfalen    701219
##  6 06                 Hessen                 166286
##  7 07                 Rheinland-Pfalz        106299
##  8 08                 Baden-Württemberg      212837
##  9 09                 Bayern                 231353
## 10 10                 Saarland                34672
## 11 11                 Berlin                 168991
## 12 12                 Brandenburg             92648
## 13 13                 Mecklenburg-Vorpommern  70982
## 14 14                 Sachsen                140348
## 15 15                 Sachsen-Anhalt          96960
## 16 16                 Thüringen               68614
```

```r
data_alo %>%
  filter(nchar(Regionalschluessel) == 2) %>%
  rename(bundesland = Regionalschluessel)
```

```
## # A tibble: 16 × 3
##    bundesland Gemeinde                  alo
##    <chr>      <chr>                   <dbl>
##  1 01         Schleswig-Holstein      92434
##  2 02         Hamburg                 69248
##  3 03         Niedersachsen          244260
##  4 04         Bremen                  35687
##  5 05         Nordrhein-Westfalen    701219
##  6 06         Hessen                 166286
##  7 07         Rheinland-Pfalz        106299
##  8 08         Baden-Württemberg      212837
##  9 09         Bayern                 231353
## 10 10         Saarland                34672
## 11 11         Berlin                 168991
## 12 12         Brandenburg             92648
## 13 13         Mecklenburg-Vorpommern  70982
## 14 14         Sachsen                140348
## 15 15         Sachsen-Anhalt          96960
## 16 16         Thüringen               68614
```

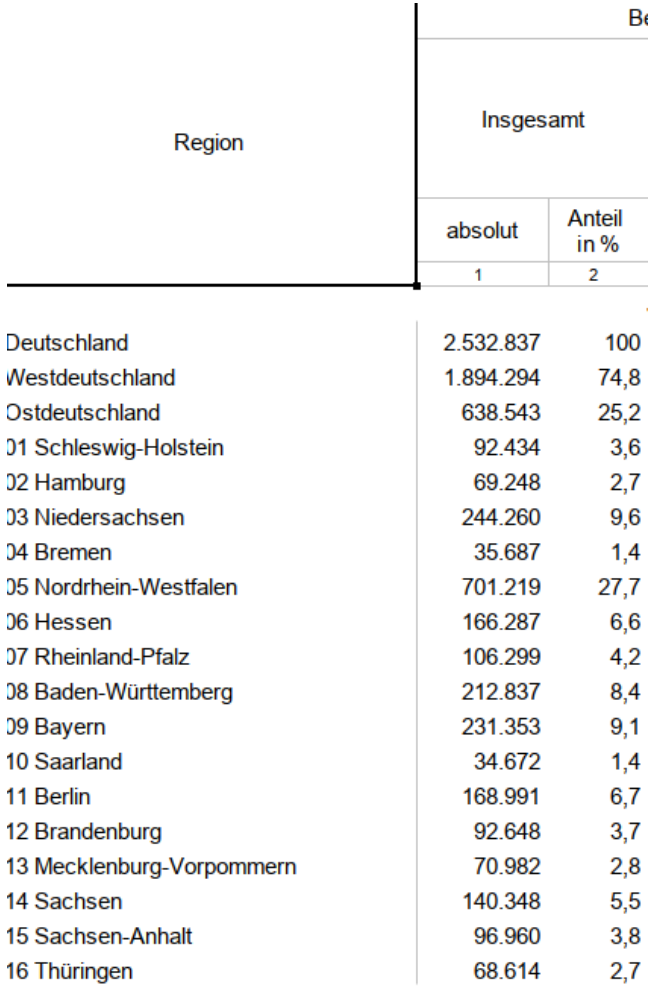
```r
# Abspeichern als check_alo_bundesland

check_alo_bundesland <- data_alo %>%
  filter(nchar(Regionalschluessel) == 2) %>%
  rename(bundesland = Regionalschluessel)
```

```
check_alo_bundesland
```

```
## # A tibble: 16 × 3
##    bundesland Gemeinde                 alo
##    <chr>      <chr>                  <dbl>
##  1 01         Schleswig-Holstein      92434
##  2 02         Hamburg                 69248
##  3 03         Niedersachsen          244260
##  4 04         Bremen                  35687
##  5 05         Nordrhein-Westfalen    701219
##  6 06         Hessen                 166286
##  7 07         Rheinland-Pfalz        106299
##  8 08         Baden-Württemberg      212837
##  9 09         Bayern                 231353
## 10 10         Saarland                34672
## 11 11         Berlin                 168991
## 12 12         Brandenburg             92648
## 13 13         Mecklenburg-Vorpommern  70982
## 14 14         Sachsen                140348
## 15 15         Sachsen-Anhalt          96960
## 16 16         Thüringen               68614
```

```
include_graphics("./figs/Alo_Laender.png")
```

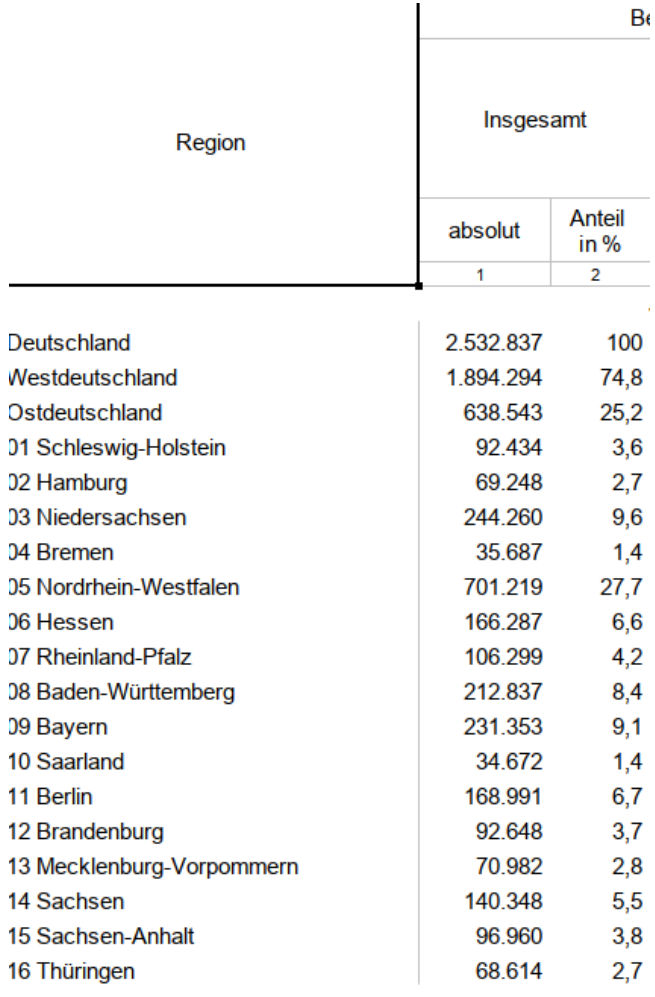


Region	Insgesamt	
	absolut	Anteil in %
	1	2
Deutschland	2.532.837	100
Westdeutschland	1.894.294	74,8
Ostdeutschland	638.543	25,2
01 Schleswig-Holstein	92.434	3,6
02 Hamburg	69.248	2,7
03 Niedersachsen	244.260	9,6
04 Bremen	35.687	1,4
05 Nordrhein-Westfalen	701.219	27,7
06 Hessen	166.287	6,6
07 Rheinland-Pfalz	106.299	4,2
08 Baden-Württemberg	212.837	8,4
09 Bayern	231.353	9,1
10 Saarland	34.672	1,4
11 Berlin	168.991	6,7
12 Brandenburg	92.648	3,7
13 Mecklenburg-Vorpommern	70.982	2,8
14 Sachsen	140.348	5,5
15 Sachsen-Anhalt	96.960	3,8
16 Thüringen	68.614	2,7

```
check_alo_bundesland
```

```
## # A tibble: 16 × 3
##    bundesland Gemeinde                 alo
##    <chr>      <chr>                  <dbl>
##  1 01         Schleswig-Holstein     92434
##  2 02         Hamburg                69248
##  3 03         Niedersachsen         244260
##  4 04         Bremen                 35687
##  5 05         Nordrhein-Westfalen   701219
##  6 06         Hessen                166286
##  7 07         Rheinland-Pfalz       106299
##  8 08         Baden-Württemberg     212837
##  9 09         Bayern                231353
## 10 10         Saarland               34672
## 11 11         Berlin                168991
## 12 12         Brandenburg            92648
## 13 13         Mecklenburg-Vorpommern 70982
## 14 14         Sachsen               140348
## 15 15         Sachsen-Anhalt         96960
## 16 16         Thüringen              68614
```

```
include_graphics("./figs/Alo_Laender.png")
```



Region	Insgesamt	
	absolut	Anteil in %
	1	2
Deutschland	2.532.837	100
Westdeutschland	1.894.294	74,8
Ostdeutschland	638.543	25,2
01 Schleswig-Holstein	92.434	3,6
02 Hamburg	69.248	2,7
03 Niedersachsen	244.260	9,6
04 Bremen	35.687	1,4
05 Nordrhein-Westfalen	701.219	27,7
06 Hessen	166.287	6,6
07 Rheinland-Pfalz	106.299	4,2
08 Baden-Württemberg	212.837	8,4
09 Bayern	231.353	9,1
10 Saarland	34.672	1,4
11 Berlin	168.991	6,7
12 Brandenburg	92.648	3,7
13 Mecklenburg-Vorpommern	70.982	2,8
14 Sachsen	140.348	5,5
15 Sachsen-Anhalt	96.960	3,8
16 Thüringen	68.614	2,7

Beide Datenreihen sind identisch (lediglich Hessen weicht um eine Person ab)

INTERNE KONSISTENZ ÜBERPRÜFEN

Berechne: Anzahl an Arbeitslosen für jedes Bundesland als Summe der Arbeitslosen einer Gemeinde.

```
# Nur Gemeindedaten nutzen, dann auf Bundeslandebende die Summe aus den Gemeindedaten berechnen
alo_meta <- data_alo %>%
  filter(nchar(Regionalschluessel) == 8) %>%
  mutate(landkreis = str_extract(Regionalschluessel, "^.{5}"),
         bundesland = str_extract(Regionalschluessel, "^.{2}"))

alo_bundesland <- alo_meta %>%
  group_by(bundesland) %>%
  summarise(total_alo = sum(alo))

alo_landkreis <- alo_meta %>%
  group_by(landkreis) %>%
  summarise(total_alo = sum(alo)) %>%
  rename(Regionalschluessel = landkreis)
```

```
data_alo
```

```
## # A tibble: 11,343 × 3
##    Regionalschluessel Gemeinde                 alo
##    <chr>              <chr>                  <dbl>
##  1 01                 Schleswig-Holstein     92434
##  2 01001000           Flensburg, Stadt        4512
##  3 01002000           Kiel, Landeshauptstadt 12345
##  4 01003000           Lübeck, Hansestadt      9692
##  5 01004000           Neumünster, Stadt       3836
##  6 01051              Dithmarschen            4628
##  7 01051001           Albersdorf               150
##  8 01051002           Arkebek                    5
##  9 01051003           Averlak                   15
## 10 01051004           Bargenstedt               17
## # … with 11,333 more rows
```

```
data_alo %>%
  filter(nchar(Regionalschluessel) == 8)
```

```
## # A tibble: 11,010 × 3
##    Regionalschluessel Gemeinde                 alo
##    <chr>              <chr>                  <dbl>
##  1 01001000           Flensburg, Stadt        4512
##  2 01002000           Kiel, Landeshauptstadt 12345
##  3 01003000           Lübeck, Hansestadt      9692
##  4 01004000           Neumünster, Stadt       3836
##  5 01051001           Albersdorf               150
##  6 01051002           Arkebek                    5
##  7 01051003           Averlak                   15
##  8 01051004           Bargenstedt               17
##  9 01051005           Barkenholm                 3
## 10 01051006           Barlt                     21
## # … with 11,000 more rows
```

```r
data_alo %>%
  filter(nchar(Regionalschluessel) == 8) %>%
  mutate(landkreis = str_extract(Regionalschluessel,
```

```
## # A tibble: 11,010 × 4
##    Regionalschluessel Gemeinde                 alo landkreis
##    <chr>              <chr>                  <dbl> <chr>
##  1 01001000           Flensburg, Stadt        4512 01001
##  2 01002000           Kiel, Landeshauptstadt 12345 01002
##  3 01003000           Lübeck, Hansestadt      9692 01003
##  4 01004000           Neumünster, Stadt       3836 01004
##  5 01051001           Albersdorf               150 01051
##  6 01051002           Arkebek                    5 01051
##  7 01051003           Averlak                   15 01051
##  8 01051004           Bargenstedt               17 01051
##  9 01051005           Barkenholm                 3 01051
## 10 01051006           Barlt                     21 01051
## # … with 11,000 more rows
```

```r
data_alo %>%
  filter(nchar(Regionalschluessel) == 8) %>%
  mutate(landkreis = str_extract(Regionalschluessel,
  mutate(bundesland = str_extract(Regionalschluessel
```

```
## # A tibble: 11,010 × 5
##    Regionalschluessel Gemeinde                 alo landkreis bundesland
##    <chr>              <chr>                  <dbl> <chr>     <chr>
##  1 01001000           Flensburg, Stadt        4512 01001     01
##  2 01002000           Kiel, Landeshauptstadt 12345 01002     01
##  3 01003000           Lübeck, Hansestadt      9692 01003     01
##  4 01004000           Neumünster, Stadt       3836 01004     01
##  5 01051001           Albersdorf               150 01051     01
##  6 01051002           Arkebek                    5 01051     01
##  7 01051003           Averlak                   15 01051     01
##  8 01051004           Bargenstedt               17 01051     01
##  9 01051005           Barkenholm                 3 01051     01
## 10 01051006           Barlt                     21 01051     01
## # … with 11,000 more rows
```

```
data_alo %>%
  filter(nchar(Regionalschluessel) == 8) %>%
  mutate(landkreis = str_extract(Regionalschluessel,
  mutate(bundesland = str_extract(Regionalschluessel
alo_meta
```

```
alo_meta
```

```
## # A tibble: 11,010 × 5
##    Regionalschluessel Gemeinde                 alo landkreis bundesland
##    <chr>              <chr>                  <dbl> <chr>     <chr>
##  1 01001000           Flensburg, Stadt        4512 01001     01
##  2 01002000           Kiel, Landeshauptstadt 12345 01002     01
##  3 01003000           Lübeck, Hansestadt      9692 01003     01
##  4 01004000           Neumünster, Stadt       3836 01004     01
##  5 01051001           Albersdorf               150 01051     01
##  6 01051002           Arkebek                    5 01051     01
##  7 01051003           Averlak                   15 01051     01
##  8 01051004           Bargenstedt               17 01051     01
##  9 01051005           Barkenholm                 3 01051     01
## 10 01051006           Barlt                     21 01051     01
## # … with 11,000 more rows
```

```
alo_meta %>%
  group_by(bundesland)
```

```
## # A tibble: 11,010 × 5
## # Groups:   bundesland [16]
##    Regionalschluessel Gemeinde                 alo landkreis bundesland
##    <chr>              <chr>                  <dbl> <chr>     <chr>
##  1 01001000           Flensburg, Stadt        4512 01001     01
##  2 01002000           Kiel, Landeshauptstadt 12345 01002     01
##  3 01003000           Lübeck, Hansestadt      9692 01003     01
##  4 01004000           Neumünster, Stadt       3836 01004     01
##  5 01051001           Albersdorf               150 01051     01
##  6 01051002           Arkebek                    5 01051     01
##  7 01051003           Averlak                   15 01051     01
##  8 01051004           Bargenstedt               17 01051     01
##  9 01051005           Barkenholm                 3 01051     01
## 10 01051006           Barlt                     21 01051     01
## # … with 11,000 more rows
```

```r
alo_meta %>%
  group_by(bundesland) %>%
  summarise(total_alo = sum(alo))
```

```
## # A tibble: 16 × 2
##    bundesland total_alo
##    <chr>          <dbl>
##  1 01             92449
##  2 02             69248
##  3 03            244277
##  4 04             35687
##  5 05            701212
##  6 06            166296
##  7 07            106287
##  8 08            212835
##  9 09            231355
## 10 10             34675
## 11 11            168991
## 12 12             92644
## 13 13             70989
## 14 14            140348
## 15 15             96960
## 16 16             68609
```

```
alo_meta %>%
  group_by(bundesland) %>%
  summarise(total_alo = sum(alo)) ->
alo_bundesland
```

```
alo_meta
```

```
## # A tibble: 11,010 × 5
##    Regionalschluessel Gemeinde                 alo landkreis bundesland
##    <chr>              <chr>                  <dbl> <chr>     <chr>
##  1 01001000           Flensburg, Stadt        4512 01001     01
##  2 01002000           Kiel, Landeshauptstadt 12345 01002     01
##  3 01003000           Lübeck, Hansestadt      9692 01003     01
##  4 01004000           Neumünster, Stadt       3836 01004     01
##  5 01051001           Albersdorf               150 01051     01
##  6 01051002           Arkebek                    5 01051     01
##  7 01051003           Averlak                   15 01051     01
##  8 01051004           Bargenstedt               17 01051     01
##  9 01051005           Barkenholm                 3 01051     01
## 10 01051006           Barlt                     21 01051     01
## # … with 11,000 more rows
```

```
alo_meta %>%
  group_by(landkreis)
```

```
## # A tibble: 11,010 × 5
## # Groups:   landkreis [401]
##    Regionalschluessel Gemeinde                 alo landkreis bundesland
##    <chr>              <chr>                  <dbl> <chr>     <chr>
##  1 01001000           Flensburg, Stadt        4512 01001     01
##  2 01002000           Kiel, Landeshauptstadt 12345 01002     01
##  3 01003000           Lübeck, Hansestadt      9692 01003     01
##  4 01004000           Neumünster, Stadt       3836 01004     01
##  5 01051001           Albersdorf               150 01051     01
##  6 01051002           Arkebek                    5 01051     01
##  7 01051003           Averlak                   15 01051     01
##  8 01051004           Bargenstedt               17 01051     01
##  9 01051005           Barkenholm                 3 01051     01
## 10 01051006           Barlt                     21 01051     01
## # … with 11,000 more rows
```

```r
alo_meta %>%
  group_by(landkreis) %>%
  summarise(total_alo = sum(alo))
```

```
## # A tibble: 401 × 2
##    landkreis total_alo
##    <chr>         <dbl>
##  1 01001          4512
##  2 01002         12345
##  3 01003          9692
##  4 01004          3836
##  5 01051          4632
##  6 01053          5592
##  7 01054          5657
##  8 01055          5748
##  9 01056          8599
## 10 01057          3264
## # … with 391 more rows
```

```
alo_meta %>%
  group_by(landkreis) %>%
  summarise(total_alo = sum(alo)) %>%
  rename(Regionalschluessel = landkreis)
```

```
## # A tibble: 401 × 2
##    Regionalschluessel total_alo
##    <chr>                   <dbl>
##  1 01001                    4512
##  2 01002                   12345
##  3 01003                    9692
##  4 01004                    3836
##  5 01051                    4632
##  6 01053                    5592
##  7 01054                    5657
##  8 01055                    5748
##  9 01056                    8599
## 10 01057                    3264
## # … with 391 more rows
```

```
alo_meta %>%
  group_by(landkreis) %>%
  summarise(total_alo = sum(alo)) %>%
  rename(Regionalschluessel = landkreis) ->
alo_landkreis
```

INTERNE KONSISTENZ ÜBERPRÜFEN

Wir wollen nun die zwei Tabellen miteinander verbinden (besserer Überblick)

- Datensatz `check_alo_bundeland`: Auf Bundesland aggregierte Zahlen der Arbeitslosigkeit aus den Gemeinden
- Datessatz `alo_bundesland`: Die schon von der Arbeitsagentur aggregierte Zahlen in unserem Datensatz

```
left_join(check_alo_bundesland, alo_bundesland, by =
```

```
## # A tibble: 16 × 4
##    bundesland Gemeinde                 alo total_alo
##    <chr>      <chr>                  <dbl>     <dbl>
##  1 01         Schleswig-Holstein     92434     92449
##  2 02         Hamburg                69248     69248
##  3 03         Niedersachsen         244260    244277
##  4 04         Bremen                 35687     35687
##  5 05         Nordrhein-Westfalen   701219    701212
##  6 06         Hessen                166286    166296
##  7 07         Rheinland-Pfalz       106299    106287
##  8 08         Baden-Württemberg     212837    212835
##  9 09         Bayern                231353    231355
## 10 10         Saarland               34672     34675
## 11 11         Berlin                168991    168991
## 12 12         Brandenburg            92648     92644
## 13 13         Mecklenburg-Vorpommern 70982     70989
## 14 14         Sachsen               140348    140348
## 15 15         Sachsen-Anhalt         96960     96960
## 16 16         Thüringen              68614     68609
```

```
left_join(check_alo_bundesland, alo_bundesland, by =
  check_consitency
```

```
left_join(check_alo_bundesland, alo_bundesland, by =
  check_consitency

check_consitency
```

```
## # A tibble: 16 × 4
##    bundesland Gemeinde                 alo total_alo
##    <chr>      <chr>                  <dbl>     <dbl>
##  1 01         Schleswig-Holstein     92434     92449
##  2 02         Hamburg                69248     69248
##  3 03         Niedersachsen         244260    244277
##  4 04         Bremen                 35687     35687
##  5 05         Nordrhein-Westfalen   701219    701212
##  6 06         Hessen                166286    166296
##  7 07         Rheinland-Pfalz       106299    106287
##  8 08         Baden-Württemberg     212837    212835
##  9 09         Bayern                231353    231355
## 10 10         Saarland               34672     34675
## 11 11         Berlin                168991    168991
## 12 12         Brandenburg            92648     92644
## 13 13         Mecklenburg-Vorpommern 70982     70989
## 14 14         Sachsen               140348    140348
## 15 15         Sachsen-Anhalt         96960     96960
## 16 16         Thüringen              68614     68609
```

```
left_join(check_alo_bundesland, alo_bundesland, by =
  check_consitency

check_consitency %>%
  mutate(diff = alo - total_alo)
```

```
## # A tibble: 16 × 5
##    bundesland Gemeinde                 alo total_alo  diff
##    <chr>      <chr>                  <dbl>     <dbl> <dbl>
##  1 01         Schleswig-Holstein     92434     92449   -15
##  2 02         Hamburg                69248     69248     0
##  3 03         Niedersachsen         244260    244277   -17
##  4 04         Bremen                 35687     35687     0
##  5 05         Nordrhein-Westfalen   701219    701212     7
##  6 06         Hessen                166286    166296   -10
##  7 07         Rheinland-Pfalz       106299    106287    12
##  8 08         Baden-Württemberg     212837    212835     2
##  9 09         Bayern                231353    231355    -2
## 10 10         Saarland               34672     34675    -3
## 11 11         Berlin                168991    168991     0
## 12 12         Brandenburg            92648     92644     4
## 13 13         Mecklenburg-Vorpommern 70982     70989    -7
## 14 14         Sachsen               140348    140348     0
## 15 15         Sachsen-Anhalt         96960     96960     0
## 16 16         Thüringen              68614     68609     5
```

```r
left_join(check_alo_bundesland, alo_bundesland, by =
  check_consitency

check_consitency %>%
  mutate(diff = alo - total_alo)
```

```
## # A tibble: 16 × 5
##    bundesland Gemeinde                 alo total_alo  diff
##    <chr>      <chr>                  <dbl>     <dbl> <dbl>
##  1 01         Schleswig-Holstein     92434     92449   -15
##  2 02         Hamburg                69248     69248     0
##  3 03         Niedersachsen         244260    244277   -17
##  4 04         Bremen                 35687     35687     0
##  5 05         Nordrhein-Westfalen   701219    701212     7
##  6 06         Hessen                166286    166296   -10
##  7 07         Rheinland-Pfalz       106299    106287    12
##  8 08         Baden-Württemberg     212837    212835     2
##  9 09         Bayern                231353    231355    -2
## 10 10         Saarland               34672     34675    -3
## 11 11         Berlin                168991    168991     0
## 12 12         Brandenburg            92648     92644     4
## 13 13         Mecklenburg-Vorpommern 70982     70989    -7
## 14 14         Sachsen               140348    140348     0
## 15 15         Sachsen-Anhalt         96960     96960     0
## 16 16         Thüringen              68614     68609     5
```

Es bestehen kleinere Unstimmigkeiten, jedoch sind diese marginal.

Pro-Kopf Verschuldung

Pro-Kopf Verschuldung auf Gemeindeebene

- ✚ Auf Gemeindeebene aus dem Jahr 2017
- ✚ Querschnittsdaten
- ✚ Vom Statistischen Bundesamt direkt als Excel-Tabelle heruntergeladen (✓)

> Welche Tabellenblätter sollten wir nutzen?

```
excel_sheets("../case-study/data/Schulden_2017.xlsx")
```

```
##  [1] "Titel"              "Impressum"          "Inhalt"
##  [4] "Abkürzungen"        "Erläuterungen"      "SH"
##  [7] "NI"                 "NW"                 "HE"
## [10] "RP"                 "BW"                 "BY"
## [13] "SL"                 "BB"                 "MV"
## [16] "SN"                 "ST"                 "TH"
## [19] "Statistische Ämter"
```

Mehrere Tabellenblätter einlesen

- Nicht alle Informationen in **einem Tabellenblatt** enthalten
  - Viele separate Tabellenblätter
  - Hier kommt die `for`-Schleife zum Einsatz

Mehrere Tabellenblätter einlesen

- Nicht alle Informationen in **einem Tabellenblatt** enthalten
  - Viele separate Tabellenblätter
  - Hier kommt die `for`-Schleife zum Einsatz

> Zuerst schauen wir jedoch welche Informationen wir benötigen anhand eines Beispiels:

Mehrere Tabellenblätter einlesen

```
sh <- read_xlsx("../case-study/data/Schulden_2017.xlsx", sheet = "SH")
head(sh,20)
```

```
## # A tibble: 20 × 16
##    Zurück zu…¹ ...2  ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11 ...12
##    <chr>       <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
##  1 <NA>        <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  2 <NA>        <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  3 "Tabelle 1… <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  4 "nach Höhe… <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
##  5 "Regional-… Geme… Verw… "Ein… Schu… "Sch… Schu… <NA>  <NA>  <NA>  <NA>  <NA>
##  6 <NA>        <NA>  <NA>  <NA>  <NA>  <NA>  zusa… Schu… ante… <NA>  <NA>  <NA>
##  7 <NA>        <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  zusa… davo… <NA>  <NA>
##  8 <NA>        <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  100%  50% … unte…
##  9 <NA>        <NA>  <NA>  <NA>  EUR   <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## 10 <NA>        <NA>  <NA>  <NA>  1     "2"   3     4     5     6     7     8
## 11 "010010000… Flen… krei… "877… 5082… "579… 2979… 1126… 1853… 1822… 2906… 2072…
## 12 "010020000… Kiel… krei… "247… 9488… "383… 5632… 5626… 5701… 4437… 0     1264…
## 13 "010030000… Lübe… krei… "216… 1206… "556… 6014… 5938… 7606… 7027… 0     5788…
## 14 "010040000… Neum… krei… "787… 4260… "540… 1233… 1152… 8176… 8103… 0     72975
## 15 "01051"     Krei… Krei… "{13… 6517… "487… 4309… 4302… 72975 0     0     72975
```

Mehrere Tabellenblätter einlesen

Wir benötigen:

- "Regionalschlüssel"
- "Gemeindename"
- "Einwohner"
- "Schuldes des öffentlichen Bereichs insgesamt"
- "Schulden je Einwohner"

Variablenbezeichnungen beginnen in Zeile 5, d.h. wir ignorieren die ersten 4 Zeilen beim Einlesen.

Was ist hier eine Beobachtung?

Mehrere Tabellenblätter einlesen

Der Übersicht halber wollen wir noch eine Spalte hinzufügen, welche den Namen des Tabellenblattes enthält, welches wir gerade eingelesen haben.

```
# Einlesen des Tabellenblattes "SH" ohne die ersten 5 Zeilen und nur die Spalten 1-6
schulden_individuell <- read_xlsx("../case-study/data/Schulden_2017.xlsx", sheet = "SH", skip = 5)[1:6]
# Umbenennen der ersten 6 Spalten
colnames(schulden_individuell) <- c("Regionalschluessel", "Gemeinde",
                                    "Verwaltungsform", "Einwohner", "Schulden_gesamt", "Schulden_pro_kopf")

# Zusätzliche Spalte hinzufügen mit dem Namen des Tabellenblattes
schulden_individuell$Bundesland <- "SH"
```

```r
# Einlesen des Tabellenblattes "SH" ohne die ersten
read_xlsx("../case-study/data/Schulden_2017.xlsx", s
```

```
## # A tibble: 1,312 × 6
##    `Regional-\r\nschlüssel` Gemeinde/Gemeindev…¹ Verwa…² Einwo…³ Schul…⁴ Schul
##    <chr>                    <chr>                <chr>   <chr>   <chr>   <chr>
##  1 <NA>                     <NA>                 <NA>    <NA>    <NA>    <NA>
##  2 <NA>                     <NA>                 <NA>    <NA>    <NA>    <NA>
##  3 <NA>                     <NA>                 <NA>    <NA>    <NA>    <NA>
##  4 <NA>                     <NA>                 <NA>    <NA>    EUR     <NA>
##  5 <NA>                     <NA>                 <NA>    <NA>    1       2
##  6 010010000000             Flensburg, Stadt     kreisf… 87770   508281… 5791.
##  7 010020000000             Kiel, Landeshauptst… kreisf… 247135  948848… 3839.
##  8 010030000000             Lübeck, Hansestadt   kreisf… 216739  120662… 5567.
##  9 010040000000             Neumünster, Stadt    kreisf… 78759   426019… 5409.
## 10 01051                    Kreisverwaltung Dit… Kreisv… {133 6… 651790… 487.5
## # … with 1,302 more rows, and abbreviated variable names
## #   ¹`Gemeinde/Gemeindeverband`, ²Verwaltungsform,
## #   ³`Einwohner/in\r\nam\r\n30.06.2017`,
## #   ⁴`Schulden des öffentlichen Bereichs insgesamt`,
## #   ⁵`Schulden je \r\nEinwohner/in`
```

```
# Einlesen des Tabellenblattes "SH" ohne die ersten
read_xlsx("../case-study/data/Schulden_2017.xlsx", s
  schulden_individuell
```

```
# Einlesen des Tabellenblattes "SH" ohne die ersten
read_xlsx("../case-study/data/Schulden_2017.xlsx", s
  schulden_individuell

# Umbenennen der ersten 6 Spalten
colnames(schulden_individuell) <- c("Regionalschlues
                                    "Verwaltungsform
```

```
# Einlesen des Tabellenblattes "SH" ohne die ersten
read_xlsx("../case-study/data/Schulden_2017.xlsx", s
  schulden_individuell

# Umbenennen der ersten 6 Spalten
colnames(schulden_individuell) <- c("Regionalschlues
                                    "Verwaltungsform

# Zusätzliche Spalte hinzufügen mit dem Namen des Ta
schulden_individuell$Bundesland <- "SH"
```

```r
# Einlesen des Tabellenblattes "SH" ohne die ersten
read_xlsx("../case-study/data/Schulden_2017.xlsx", s
  schulden_individuell

# Umbenennen der ersten 6 Spalten
colnames(schulden_individuell) <- c("Regionalschlues
                                    "Verwaltungsform

# Zusätzliche Spalte hinzufügen mit dem Namen des Ta
schulden_individuell$Bundesland <- "SH"

schulden_individuell
```

```
## # A tibble: 1,312 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>   <chr>   <chr>   <chr>
##  1 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  2 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  3 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  4 <NA>               <NA>               <NA>    <NA>    EUR     <NA>    SH
##  5 <NA>               <NA>               <NA>    <NA>    1       2       SH
##  6 010010000000       Flensburg, Stadt   kreisf… 87770   508281… 5791.06 SH
##  7 010020000000       Kiel, Landeshaupt… kreisf… 247135  948848… 3839.39 SH
##  8 010030000000       Lübeck, Hansestadt kreisf… 216739  120662… 5567.16 SH
##  9 010040000000       Neumünster, Stadt  kreisf… 78759   426019… 5409.15 SH
## 10 01051              Kreisverwaltung D… Kreisv… {133 6… 651790… 487.56  SH
## # … with 1,302 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

Mehrere Tabellenblätter einlesen

Nun können wir genauso bei allen anderen Tabellenblättern vorgehen:

Mehrere Tabellenblätter einlesen

Nun können wir genauso bei allen anderen Tabellenblättern vorgehen:

```
# Daten mit for-Schleife einlesen (Struktur gleich wie im vorherigen Chunk)
sheet_names <- excel_sheets("../case-study/data/Schulden_2017.xlsx")
# Einlesen der Tabellenblätter 7-18 (alle Bundesländer)
sheet_read <- sheet_names[7:18]
for (i in 1:length(sheet_read)){
  tmp <- read_xlsx("../case-study/data/Schulden_2017.xlsx", sheet = sheet_read[i], skip = 5)[1:6]
   tmp$Bundesland <- sheet_read[i]
  colnames(tmp) <- c("Regionalschluessel", "Gemeinde", "Verwaltungsform",
                     "Einwohner", "Schulden_gesamt", "Schulden_pro_kopf", "Bundesland")
  # Daten aller weiteren Tabellenblätter unter den aktuellen Datensatz anheften
  schulden_individuell <- bind_rows(schulden_individuell, tmp)
}
```

Mehrere Tabellenblätter einlesen

Nun können wir genauso bei allen anderen Tabellenblättern vorgehen:

```
# Daten mit for-Schleife einlesen (Struktur gleich wie im vorherigen Chunk)
sheet_names <- excel_sheets("../case-study/data/Schulden_2017.xlsx")
# Einlesen der Tabellenblätter 7-18 (alle Bundesländer)
sheet_read <- sheet_names[7:18]
for (i in 1:length(sheet_read)){
  tmp <- read_xlsx("../case-study/data/Schulden_2017.xlsx", sheet = sheet_read[i], skip = 5)[1:6]
   tmp$Bundesland <- sheet_read[i]
  colnames(tmp) <- c("Regionalschluessel", "Gemeinde", "Verwaltungsform",
                     "Einwohner", "Schulden_gesamt", "Schulden_pro_kopf", "Bundesland")
  # Daten aller weiteren Tabellenblätter unter den aktuellen Datensatz anheften
  schulden_individuell <- bind_rows(schulden_individuell, tmp)
}
```

Mehrere Tabellenblätter einlesen

Nun können wir genauso bei allen anderen Tabellenblättern vorgehen:

```
# Daten mit for-Schleife einlesen (Struktur gleich wie im vorherigen Chunk)
sheet_names <- excel_sheets("../case-study/data/Schulden_2017.xlsx")
# Einlesen der Tabellenblätter 7-18 (alle Bundesländer)
sheet_read <- sheet_names[7:18]
for (i in 1:length(sheet_read)){
  tmp <- read_xlsx("../case-study/data/Schulden_2017.xlsx", sheet = sheet_read[i], skip = 5)[1:6]
   tmp$Bundesland <- sheet_read[i]
  colnames(tmp) <- c("Regionalschluessel", "Gemeinde", "Verwaltungsform",
                     "Einwohner", "Schulden_gesamt", "Schulden_pro_kopf", "Bundesland")
  # Daten aller weiteren Tabellenblätter unter den aktuellen Datensatz anheften
  schulden_individuell <- bind_rows(schulden_individuell, tmp)
}
```

Eine zusätzliche Spalte generieren, welche die Information pro Bundesland enthält

```
# Daten mit for-Schleife einlesen (Struktur gleich w
```

```
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
```

```
##  [1] "Titel"               "Impressum"           "Inhalt"
##  [4] "Abkürzungen"         "Erläuterungen"       "SH"
##  [7] "NI"                  "NW"                  "HE"
## [10] "RP"                  "BW"                  "BY"
## [13] "SL"                  "BB"                  "MV"
## [16] "SN"                  "ST"                  "TH"
## [19] "Statistische Ämter"
```

```
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names
```

```
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names

# Einlesen der Tabellenblätter 7-18 (alle Bundesländ
```

```r
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names

# Einlesen der Tabellenblätter 7-18 (alle Bundesländ
sheet_names[7:18]
```

```
##  [1] "NI" "NW" "HE" "RP" "BW" "BY" "SL" "BB" "MV" "SN" "ST" "TH"
```

```
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names

# Einlesen der Tabellenblätter 7-18 (alle Bundesländ
sheet_names[7:18] ->
  sheet_read
```

```
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names

# Einlesen der Tabellenblätter 7-18 (alle Bundesländ
sheet_names[7:18] ->
  sheet_read

length(sheet_read)
```

```
## [1] 12
```

```
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names

# Einlesen der Tabellenblätter 7-18 (alle Bundesländ
sheet_names[7:18] ->
  sheet_read

length(sheet_read)
```

```
for (i in 1:length(sheet_read)){
  tmp <- read_xlsx("../case-study/data/Schulden_2017
  tmp$Bundesland <- sheet_read[i]
  colnames(tmp) <- c("Regionalschluessel", "Gemeinde
                     "Einwohner", "Schulden_gesamt",
# Daten aller weiteren Tabellenblätter unter den akt
  schulden_individuell <- bind_rows(schulden_individ
}
```

```
## [1] 12
```

```r
# Daten mit for-Schleife einlesen (Struktur gleich w
excel_sheets("../case-study/data/Schulden_2017.xlsx"
  sheet_names

# Einlesen der Tabellenblätter 7-18 (alle Bundesländ
sheet_names[7:18] ->
  sheet_read

length(sheet_read)

for (i in 1:length(sheet_read)){
  tmp <- read_xlsx("../case-study/data/Schulden_2017
  tmp$Bundesland <- sheet_read[i]
  colnames(tmp) <- c("Regionalschluessel", "Gemeinde
                     "Einwohner", "Schulden_gesamt",
# Daten aller weiteren Tabellenblätter unter den akt
  schulden_individuell <- bind_rows(schulden_individ
}

schulden_individuell
```

```
## [1] 12
```

```
## # A tibble: 25,796 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>   <chr>   <chr>   <chr>
##  1 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  2 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  3 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  4 <NA>               <NA>               <NA>    <NA>    EUR     <NA>    SH
##  5 <NA>               <NA>               <NA>    <NA>    1       2       SH
##  6 010010000000       Flensburg, Stadt   kreisf… 87770   508281… 5791.06 SH
##  7 010020000000       Kiel, Landeshaupt… kreisf… 247135  948848… 3839.39 SH
##  8 010030000000       Lübeck, Hansestadt kreisf… 216739  120662… 5567.16 SH
##  9 010040000000       Neumünster, Stadt  kreisf… 78759   426019… 5409.15 SH
## 10 01051              Kreisverwaltung D… Kreisv… {133 6… 651790… 487.56  SH
## # … with 25,786 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

Variablen umformen

```
head(schulden_individuell,15)
```

```
## # A tibble: 15 × 7
##    Regionalschluessel Gemeinde           Verwa…[1] Einwo…[2] Schul…[3] Schul…[4] Bunde…[5]
##    <chr>              <chr>              <chr>     <chr>     <chr>     <chr>     <chr>
##  1 <NA>               <NA>               <NA>      <NA>      <NA>      <NA>      SH
##  2 <NA>               <NA>               <NA>      <NA>      <NA>      <NA>      SH
##  3 <NA>               <NA>               <NA>      <NA>      <NA>      <NA>      SH
##  4 <NA>               <NA>               <NA>      <NA>      EUR       <NA>      SH
##  5 <NA>               <NA>               <NA>      <NA>      1         2         SH
##  6 010010000000       Flensburg, Stadt   kreisf…   87770     508281…   5791.06   SH
##  7 010020000000       Kiel, Landeshaupt… kreisf…   247135    948848…   3839.39   SH
##  8 010030000000       Lübeck, Hansestadt kreisf…   216739    120662…   5567.16   SH
##  9 010040000000       Neumünster, Stadt  kreisf…   78759     426019…   5409.15   SH
## 10 01051              Kreisverwaltung D… Kreisv…   {133 6…   651790…   487.56    SH
## 11 010510011011       Brunsbüttel, Stadt amtsfr…   12781     349348…   2733.34   SH
## 12 010510044044       Heide, Stadt       amtsfr…   21508     370562…   1722.91   SH
## 13 010515163          Amtsverwaltung Bu… Amtsve…   {15 65…   1047175   66.900…   SH
## 14 010515163003       Averlak            amtsan…   576       1197761   2079.4…   SH
## 15 010515163010       Brickeln           amtsan…   219       496264    2266.0…   SH
## # … with abbreviated variable names [1]Verwaltungsform, [2]Einwohner,
```

Variablen umformen

Wir sehen, es gibt immer noch einige Probleme:

- Die Werte unserer Variablen stehen nicht direkt unter dem Variablennamen
  - Dies können wir am einfachsten bereinigen indem wir alle `NA`s im Regionalschlüssel entfernen

Variablen umformen

Wir sehen, es gibt immer noch einige Probleme:

- Die Werte unserer Variablen stehen nicht direkt unter dem Variablennamen
  - Dies können wir am einfachsten bereinigen indem wir alle `NA`s im Regionalschlüssel entfernen
- Die Variablen "Einwohner", "Schulden_gesamt" und "Schulden_pro_Kopf" sind alle als `character` hinterlegt (`<chr>` unter dem Variablennamen in der vorherigen Tabelle)
  - Beispiel warum Klasse `character` (Zeile 28): Es sind geschweifte Klammern enthalten

```
schulden_individuell[28,]
```

```
## # A tibble: 1 × 7
##   Regionalschluessel   Gemeinde          Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵
##   <chr>                <chr>             <chr>   <chr>   <chr>   <chr>   <chr>
## 1 010515163_Summe(Amt) Burg-Sankt Micha… Amtsge… {15 65… {22 84… {1 460} SH
## # … with abbreviated variable names ¹Verwaltungsform, ²Einwohner,
## #   ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

Variablen umformen

Wir sehen, es gibt immer noch einige Probleme:

- Die Werte unserer Variablen stehen nicht direkt unter dem Variablennamen
  - Dies können wir am einfachsten bereinigen indem wir alle `NA`s im Regionalschlüssel entfernen
- Die Variablen "Einwohner", "Schulden_gesamt" und "Schulden_pro_Kopf" sind alle als `character` hinterlegt (`<chr>` unter dem Variablennamen in der vorherigen Tabelle)
  - Beispiel warum Klasse `character` (Zeile 28): Es sind geschweifte Klammern enthalten

```
schulden_individuell[28,]
```

```
## # A tibble: 1 × 7
##   Regionalschluessel   Gemeinde          Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵
##   <chr>                <chr>             <chr>   <chr>   <chr>   <chr>   <chr>
## 1 010515163_Summe(Amt) Burg-Sankt Micha… Amtsge… {15 65… {22 84… {1 460} SH
## # … with abbreviated variable names ¹Verwaltungsform, ²Einwohner,
## #   ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

- Definition einer Variablen `landkreis`: Ersten 5 Zeichen im Regionalschlüssel

```
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell
```

```
## # A tibble: 13,554 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>   <chr>   <chr>   <chr>
##  1 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  2 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  3 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  4 <NA>               <NA>               <NA>    <NA>    EUR     <NA>    SH
##  5 <NA>               <NA>               <NA>    <NA>    1       2       SH
##  6 010010000000       Flensburg, Stadt   kreisf… 87770   508281… 5791.06 SH
##  7 010020000000       Kiel, Landeshaupt… kreisf… 247135  948848… 3839.39 SH
##  8 010030000000       Lübeck, Hansestadt kreisf… 216739  120662… 5567.16 SH
##  9 010040000000       Neumünster, Stadt  kreisf… 78759   426019… 5409.15 SH
## 10 01051              Kreisverwaltung D… Kreisv… {133 6… 651790… 487.56  SH
## # … with 13,544 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel))
```

```
## # A tibble: 13,450 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>   <chr>   <chr>   <chr>
##  1 010010000000       Flensburg, Stadt   kreisf… 87770   508281… 5791.06 SH
##  2 010020000000       Kiel, Landeshaupt… kreisf… 247135  948848… 3839.39 SH
##  3 010030000000       Lübeck, Hansestadt kreisf… 216739  120662… 5567.16 SH
##  4 010040000000       Neumünster, Stadt  kreisf… 78759   426019… 5409.15 SH
##  5 01051              Kreisverwaltung D… Kreisv… {133 6… 651790… 487.56  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr… 12781   349348… 2733.34 SH
##  7 010510044044       Heide, Stadt       amtsfr… 21508   370562… 1722.91 SH
##  8 010515163          Amtsverwaltung Bu… Amtsve… {15 65… 1047175 66.900… SH
##  9 010515163003       Averlak            amtsan… 576     1197761 2079.4… SH
## 10 010515163010       Brickeln           amtsan… 219     496264  2266.0… SH
## # … with 13,440 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
```

```
## # A tibble: 13,450 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>     <dbl> <chr>   <chr>
##  1 010010000000       Flensburg, Stadt   kreisf… 87770    5.08e8 5791.06 SH
##  2 010020000000       Kiel, Landeshaupt… kreisf… 247135   9.49e8 3839.39 SH
##  3 010030000000       Lübeck, Hansestadt kreisf… 216739   1.21e9 5567.16 SH
##  4 010040000000       Neumünster, Stadt  kreisf… 78759    4.26e8 5409.15 SH
##  5 01051              Kreisverwaltung D… Kreisv… {133 6…  6.52e7 487.56  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr… 12781    3.49e7 2733.34 SH
##  7 010510044044       Heide, Stadt       amtsfr… 21508    3.71e7 1722.91 SH
##  8 010515163          Amtsverwaltung Bu… Amtsve… {15 65…  1.05e6 66.900… SH
##  9 010515163003       Averlak            amtsan… 576      1.20e6 2079.4… SH
## 10 010515163010       Brickeln           amtsan… 219      4.96e5 2266.0… SH
## # … with 13,440 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner))
```

```
## # A tibble: 13,450 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>     <dbl>   <dbl> <chr>   <chr>
##  1 010010000000       Flensburg, Stadt   kreisf…   87770  5.08e8 5791.06 SH
##  2 010020000000       Kiel, Landeshaupt… kreisf…  247135  9.49e8 3839.39 SH
##  3 010030000000       Lübeck, Hansestadt kreisf…  216739  1.21e9 5567.16 SH
##  4 010040000000       Neumünster, Stadt  kreisf…   78759  4.26e8 5409.15 SH
##  5 01051              Kreisverwaltung D… Kreisv…      NA  6.52e7 487.56  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr…   12781  3.49e7 2733.34 SH
##  7 010510044044       Heide, Stadt       amtsfr…   21508  3.71e7 1722.91 SH
##  8 010515163          Amtsverwaltung Bu… Amtsve…      NA  1.05e6 66.900… SH
##  9 010515163003       Averlak            amtsan…     576  1.20e6 2079.4… SH
## 10 010515163010       Brickeln           amtsan…     219  4.96e5 2266.0… SH
## # … with 13,440 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner)) %>%
  mutate(Schulden_pro_kopf = as.numeric(Schulden_pro
```

```
## # A tibble: 13,450 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>     <dbl>   <dbl>   <dbl> <chr>
##  1 010010000000       Flensburg, Stadt   kreisf…   87770  5.08e8  5791.  SH
##  2 010020000000       Kiel, Landeshaupt… kreisf…  247135  9.49e8  3839.  SH
##  3 010030000000       Lübeck, Hansestadt kreisf…  216739  1.21e9  5567.  SH
##  4 010040000000       Neumünster, Stadt  kreisf…   78759  4.26e8  5409.  SH
##  5 01051              Kreisverwaltung D… Kreisv…      NA  6.52e7   488.  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr…   12781  3.49e7  2733.  SH
##  7 010510044044       Heide, Stadt       amtsfr…   21508  3.71e7  1723.  SH
##  8 010515163          Amtsverwaltung Bu… Amtsve…      NA  1.05e6    66.9 SH
##  9 010515163003       Averlak            amtsan…     576  1.20e6  2079.  SH
## 10 010515163010       Brickeln           amtsan…     219  4.96e5  2266.  SH
## # … with 13,440 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner)) %>%
  mutate(Schulden_pro_kopf = as.numeric(Schulden_pro
  mutate(landkreis = str_extract(Regionalschluessel,
```

```
## # A tibble: 13,450 × 8
##    Regionalschluessel Gemeinde   Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵ landk
##    <chr>              <chr>      <chr>     <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 010010000000       Flensburg… kreisf…   87770  5.08e8  5791. SH      01001
##  2 010020000000       Kiel, Lan… kreisf…  247135  9.49e8  3839. SH      01002
##  3 010030000000       Lübeck, H… kreisf…  216739  1.21e9  5567. SH      01003
##  4 010040000000       Neumünste… kreisf…   78759  4.26e8  5409. SH      01004
##  5 01051              Kreisverw… Kreisv…      NA  6.52e7   488. SH      01051
##  6 010510011011       Brunsbütt… amtsfr…   12781  3.49e7  2733. SH      01051
##  7 010510044044       Heide, St… amtsfr…   21508  3.71e7  1723. SH      01051
##  8 010515163          Amtsverwa… Amtsve…      NA  1.05e6    66.9 SH      01051
##  9 010515163003       Averlak    amtsan…     576  1.20e6  2079. SH      01051
## 10 010515163010       Brickeln   amtsan…     219  4.96e5  2266. SH      01051
## # … with 13,440 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland, ⁶landkreis
```

```
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner)) %>%
  mutate(Schulden_pro_kopf = as.numeric(Schulden_pro
  mutate(landkreis = str_extract(Regionalschluessel,
#bei der Gemeinde Selent wurde in der Excel Tabelle
  mutate(landkreis = ifelse(landkreis == "10575", "0
```

```
## # A tibble: 13,450 × 8
##    Regionalschluessel Gemeinde   Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵ landk
##    <chr>              <chr>      <chr>     <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 010010000000       Flensburg… kreisf…   87770  5.08e8  5791.  SH      01001
##  2 010020000000       Kiel, Lan… kreisf…  247135  9.49e8  3839.  SH      01002
##  3 010030000000       Lübeck, H… kreisf…  216739  1.21e9  5567.  SH      01003
##  4 010040000000       Neumünste… kreisf…   78759  4.26e8  5409.  SH      01004
##  5 01051              Kreisverw… Kreisv…      NA  6.52e7   488.  SH      01051
##  6 010510011011       Brunsbütt… amtsfr…   12781  3.49e7  2733.  SH      01051
##  7 010510044044       Heide, St… amtsfr…   21508  3.71e7  1723.  SH      01051
##  8 010515163          Amtsverwa… Amtsve…      NA  1.05e6    66.9 SH      01051
##  9 010515163003       Averlak    amtsan…     576  1.20e6  2079.  SH      01051
## 10 010515163010       Brickeln   amtsan…     219  4.96e5  2266.  SH      01051
## # … with 13,440 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland, ⁶landkreis
```

```
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner)) %>%
  mutate(Schulden_pro_kopf = as.numeric(Schulden_pro
  mutate(landkreis = str_extract(Regionalschluessel,
#bei der Gemeinde Selent wurde in der Excel Tabelle
  mutate(landkreis = ifelse(landkreis == "10575", "0
#manche Landkreise haben keine Infos zu den Einwohne
  filter( !is.na( Einwohner ))
```

```
## # A tibble: 11,050 × 8
##    Regionalschluessel Gemeinde   Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵ landk
##    <chr>              <chr>      <chr>     <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 010010000000       Flensburg… kreisf…   87770  5.08e8   5791. SH      01001
##  2 010020000000       Kiel, Lan… kreisf…  247135  9.49e8   3839. SH      01002
##  3 010030000000       Lübeck, H… kreisf…  216739  1.21e9   5567. SH      01003
##  4 010040000000       Neumünste… kreisf…   78759  4.26e8   5409. SH      01004
##  5 010510011011       Brunsbütt… amtsfr…   12781  3.49e7   2733. SH      01051
##  6 010510044044       Heide, St… amtsfr…   21508  3.71e7   1723. SH      01051
##  7 010515163003       Averlak    amtsan…     576  1.20e6   2079. SH      01051
##  8 010515163010       Brickeln   amtsan…     219  4.96e5   2266. SH      01051
##  9 010515163012       Buchholz   amtsan…    1008  1.23e6   1220. SH      01051
## 10 010515163016       Burg (Dit… amtsan…    4114  7.92e6   1926. SH      01051
## # … with 11,040 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland, ⁶landkreis
```

```
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner)) %>%
  mutate(Schulden_pro_kopf = as.numeric(Schulden_pro
  mutate(landkreis = str_extract(Regionalschluessel,
#bei der Gemeinde Selent wurde in der Excel Tabelle
  mutate(landkreis = ifelse(landkreis == "10575", "0
#manche Landkreise haben keine Infos zu den Einwohne
  filter( !is.na( Einwohner )) ->
schulden_bereinigt
```

```
# Die Daten wurden noch nicht schön eingelesen, in d
# waren die Variablennamen über mehrere Reihen gezog
schulden_individuell %>%
  filter(!is.na(Regionalschluessel)) %>%
  mutate(Schulden_gesamt = as.numeric(Schulden_gesam
  mutate(Einwohner = as.numeric(Einwohner)) %>%
  mutate(Schulden_pro_kopf = as.numeric(Schulden_pro
  mutate(landkreis = str_extract(Regionalschluessel,
#bei der Gemeinde Selent wurde in der Excel Tabelle
  mutate(landkreis = ifelse(landkreis == "10575", "0
#manche Landkreise haben keine Infos zu den Einwohne
  filter( !is.na( Einwohner )) ->
schulden_bereinigt
```

Konsistenzcheck zum Schulden-Datensatz

Interne Validität Schulden pro Kopf

- `Schulden_pro_Kopf_new` von Hand berechnen
- **Beachte:**
  - Geschweiften Klammern entfernen bei `Schulden_gesamt` (mit `str_remove_all`), als auch die Leerzeichen innerhalb der Zahlen (z.B. 15 653), was wir mit `gsub("[[:space:]]")` erreichen.
  - Tun wir das nicht, so würden wir wieder `NA`s im Datensatz erhalten
  - Durch die ifelse Bedingung wird der Befehl `str_remove_all` nur angewendet, wenn tatsächlich geschweifte Klammern vorhanden sind

```
# Erstellen der Vergleichstabelle
schulden_consistency <- schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschluessel) ) %>%
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(Schulden_gesamt))==TRUE,
                                  as.numeric(gsub("[[:space:]]", "", str_remove_all(Schulden_gesamt, "[{}]"))),
                                  as.numeric(Schulden_gesamt)),
         Schulden_pro_kopf = ifelse(is.na(as.numeric(Schulden_pro_kopf))==TRUE,
                                    as.numeric(gsub("[[:space:]]", "", str_remove_all(Schulden_pro_kopf, "[{}]"))),
                                    as.numeric(Schulden_pro_kopf)),
         Einwohner_num = ifelse(is.na(as.numeric(Einwohner))==TRUE,
                                as.numeric(gsub("[[:space:]]", "", str_remove_all(Einwohner, "[{}]"))),
                                as.numeric(Einwohner)),
         Schulden_pro_kopf_new = round(Schulden_gesamt / Einwohner_num,2)) %>%
  mutate(landkreis = str_extract(Regionalschluessel, "^.{5}"),
         differenz = Schulden_pro_kopf - Schulden_pro_kopf_new)
```

```
# Erstellen der Vergleichstabelle
schulden_individuell
```

```
## # A tibble: 13,554 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>   <chr>   <chr>   <chr>
##  1 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  2 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  3 <NA>               <NA>               <NA>    <NA>    <NA>    <NA>    SH
##  4 <NA>               <NA>               <NA>    <NA>    EUR     <NA>    SH
##  5 <NA>               <NA>               <NA>    <NA>    1       2       SH
##  6 010010000000       Flensburg, Stadt   kreisf… 87770   508281… 5791.06 SH
##  7 010020000000       Kiel, Landeshaupt… kreisf… 247135  948848… 3839.39 SH
##  8 010030000000       Lübeck, Hansestadt kreisf… 216739  120662… 5567.16 SH
##  9 010040000000       Neumünster, Stadt  kreisf… 78759   426019… 5409.15 SH
## 10 01051              Kreisverwaltung D… Kreisv… {133 6… 651790… 487.56  SH
## # … with 13,544 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
```

```
## # A tibble: 13,421 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>   <chr>   <chr>   <chr>
##  1 010010000000       Flensburg, Stadt   kreisf… 87770   508281… 5791.06 SH
##  2 010020000000       Kiel, Landeshaupt… kreisf… 247135  948848… 3839.39 SH
##  3 010030000000       Lübeck, Hansestadt kreisf… 216739  120662… 5567.16 SH
##  4 010040000000       Neumünster, Stadt  kreisf… 78759   426019… 5409.15 SH
##  5 01051              Kreisverwaltung D… Kreisv… {133 6… 651790… 487.56  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr… 12781   349348… 2733.34 SH
##  7 010510044044       Heide, Stadt       amtsfr… 21508   370562… 1722.91 SH
##  8 010515163          Amtsverwaltung Bu… Amtsve… {15 65… 1047175 66.900… SH
##  9 010515163003       Averlak            amtsan… 576     1197761 2079.4… SH
## 10 010515163010       Brickeln           amtsan… 219     496264  2266.0… SH
## # … with 13,411 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(S
                                  as.numeric(gsub("[
                                  as.numeric(Schulde
```

```
## # A tibble: 13,421 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>     <dbl> <chr>   <chr>
##  1 010010000000       Flensburg, Stadt   kreisf… 87770    5.08e8 5791.06 SH
##  2 010020000000       Kiel, Landeshaupt… kreisf… 247135   9.49e8 3839.39 SH
##  3 010030000000       Lübeck, Hansestadt kreisf… 216739   1.21e9 5567.16 SH
##  4 010040000000       Neumünster, Stadt  kreisf… 78759    4.26e8 5409.15 SH
##  5 01051              Kreisverwaltung D… Kreisv… {133 6…  6.52e7 487.56  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr… 12781    3.49e7 2733.34 SH
##  7 010510044044       Heide, Stadt       amtsfr… 21508    3.71e7 1722.91 SH
##  8 010515163          Amtsverwaltung Bu… Amtsve… {15 65…  1.05e6 66.900… SH
##  9 010515163003       Averlak            amtsan… 576      1.20e6 2079.4… SH
## 10 010515163010       Brickeln           amtsan… 219      4.96e5 2266.0… SH
## # … with 13,411 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(S
                                  as.numeric(gsub("[
                                  as.numeric(Schulde
  mutate(Schulden_pro_kopf = ifelse(is.na(as.numeric
                                    as.numeric(gsub(
                                    as.numeric(Schul
```

```
## # A tibble: 13,421 × 7
##    Regionalschluessel Gemeinde           Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde
##    <chr>              <chr>              <chr>   <chr>     <dbl>   <dbl> <chr>
##  1 010010000000       Flensburg, Stadt   kreisf… 87770    5.08e8  5791.  SH
##  2 010020000000       Kiel, Landeshaupt… kreisf… 247135   9.49e8  3839.  SH
##  3 010030000000       Lübeck, Hansestadt kreisf… 216739   1.21e9  5567.  SH
##  4 010040000000       Neumünster, Stadt  kreisf… 78759    4.26e8  5409.  SH
##  5 01051              Kreisverwaltung D… Kreisv… {133 6…  6.52e7   488.  SH
##  6 010510011011       Brunsbüttel, Stadt amtsfr… 12781    3.49e7  2733.  SH
##  7 010510044044       Heide, Stadt       amtsfr… 21508    3.71e7  1723.  SH
##  8 010515163          Amtsverwaltung Bu… Amtsve… {15 65…  1.05e6    66.9 SH
##  9 010515163003       Averlak            amtsan… 576      1.20e6  2079.  SH
## 10 010515163010       Brickeln           amtsan… 219      4.96e5  2266.  SH
## # … with 13,411 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland
```

```r
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(S
                                  as.numeric(gsub("[
                                  as.numeric(Schulde
  mutate(Schulden_pro_kopf = ifelse(is.na(as.numeric
                                    as.numeric(gsub(
                                    as.numeric(Schul
  mutate(Einwohner_num = ifelse(is.na(as.numeric(Ein
                                as.numeric(gsub("[[:
                                as.numeric(Einwohner
```

```
## # A tibble: 13,421 × 8
##    Regionalschluessel Gemeinde   Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵ Einwo
##    <chr>              <chr>      <chr>   <chr>     <dbl>   <dbl> <chr>     <db
##  1 010010000000       Flensburg… kreisf… 87770    5.08e8  5791.  SH         87
##  2 010020000000       Kiel, Lan… kreisf… 247135   9.49e8  3839.  SH       2471
##  3 010030000000       Lübeck, H… kreisf… 216739   1.21e9  5567.  SH       216
##  4 010040000000       Neumünste… kreisf… 78759    4.26e8  5409.  SH         78
##  5 01051              Kreisverw… Kreisv… {133 6…  6.52e7   488.  SH       133
##  6 010510011011       Brunsbütt… amtsfr… 12781    3.49e7  2733.  SH         12
##  7 010510044044       Heide, St… amtsfr… 21508    3.71e7  1723.  SH         21
##  8 010515163          Amtsverwa… Amtsve… {15 65…  1.05e6    66.9 SH         15
##  9 010515163003       Averlak    amtsan… 576      1.20e6  2079.  SH          5
## 10 010515163010       Brickeln   amtsan… 219      4.96e5  2266.  SH          2
## # … with 13,411 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland,
## #   ⁶Einwohner_num
```

```r
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(S
                                  as.numeric(gsub("[
                                  as.numeric(Schulde
  mutate(Schulden_pro_kopf = ifelse(is.na(as.numeric
                                    as.numeric(gsub(
                                    as.numeric(Schul
  mutate(Einwohner_num = ifelse(is.na(as.numeric(Ein
                                as.numeric(gsub("[[:
                                as.numeric(Einwohner
  mutate(Schulden_pro_kopf_new = round(Schulden_gesa
```

```
## # A tibble: 13,421 × 9
##    Regionalsch…¹ Gemei…² Verwa…³ Einwo…⁴ Schul…⁵ Schul…⁶ Bunde…⁷ Einwo…⁸ Schul
##    <chr>         <chr>   <chr>   <chr>     <dbl>   <dbl> <chr>     <dbl>   <db
##  1 010010000000  Flensb… kreisf… 87770    5.08e8  5791.  SH        87770  5791
##  2 010020000000  Kiel, … kreisf… 247135   9.49e8  3839.  SH       247135  3839
##  3 010030000000  Lübeck… kreisf… 216739   1.21e9  5567.  SH       216739  5567
##  4 010040000000  Neumün… kreisf… 78759    4.26e8  5409.  SH        78759  5409
##  5 01051         Kreisv… Kreisv… {133 6…  6.52e7   488.  SH       133684   488
##  6 010510011011  Brunsb… amtsfr… 12781    3.49e7  2733.  SH        12781  2733
##  7 010510044044  Heide,… amtsfr… 21508    3.71e7  1723.  SH        21508  1723
##  8 010515163     Amtsve… Amtsve… {15 65…  1.05e6    66.9 SH        15653    66
##  9 010515163003  Averlak amtsan… 576      1.20e6  2079.  SH          576  2079
## 10 010515163010  Bricke… amtsan… 219      4.96e5  2266.  SH          219  2266
## # … with 13,411 more rows, and abbreviated variable names ¹Regionalschluessel,
## #   ²Gemeinde, ³Verwaltungsform, ⁴Einwohner, ⁵Schulden_gesamt,
## #   ⁶Schulden_pro_kopf, ⁷Bundesland, ⁸Einwohner_num, ⁹Schulden_pro_kopf_new
```

```r
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(S
                                  as.numeric(gsub("[
                                  as.numeric(Schulde
  mutate(Schulden_pro_kopf = ifelse(is.na(as.numeric
                                    as.numeric(gsub(
                                    as.numeric(Schul
  mutate(Einwohner_num = ifelse(is.na(as.numeric(Ein
                                as.numeric(gsub("[[:
                                as.numeric(Einwohner
  mutate(Schulden_pro_kopf_new = round(Schulden_gesa
  mutate(landkreis = str_extract(Regionalschluessel,
         differenz = Schulden_pro_kopf - Schulden_pr
```

```
## # A tibble: 13,421 × 11
##    Regionalsch…[1] Gemei…[2] Verwa…[3] Einwo…[4] Schul…[5] Schul…[6] Bunde…[7] Einwo…[8] Schul
##    <chr>           <chr>     <chr>     <chr>       <dbl>     <dbl> <chr>       <dbl>   <db
##  1 010010000000    Flensb…   kreisf…   87770     5.08e8    5791.   SH         87770   5791
##  2 010020000000    Kiel, …   kreisf…   247135    9.49e8    3839.   SH        247135   3839
##  3 010030000000    Lübeck…   kreisf…   216739    1.21e9    5567.   SH        216739   5567
##  4 010040000000    Neumün…   kreisf…   78759     4.26e8    5409.   SH         78759   5409
##  5 01051           Kreisv…   Kreisv…   {133 6…   6.52e7     488.   SH        133684    488
##  6 010510011011    Brunsb…   amtsfr…   12781     3.49e7    2733.   SH         12781   2733
##  7 010510044044    Heide,…   amtsfr…   21508     3.71e7    1723.   SH         21508   1723
##  8 010515163       Amtsve…   Amtsve…   {15 65…   1.05e6      66.9  SH         15653     66
##  9 010515163003    Averlak   amtsan…   576       1.20e6    2079.   SH           576   2079
## 10 010515163010    Bricke…   amtsan…   219       4.96e5    2266.   SH           219   2266
## # … with 13,411 more rows, 2 more variables: landkreis <chr>, differenz <dbl>,
## #   and abbreviated variable names [1]Regionalschluessel, [2]Gemeinde,
## #   [3]Verwaltungsform, [4]Einwohner, [5]Schulden_gesamt, [6]Schulden_pro_kopf,
## #   [7]Bundesland, [8]Einwohner_num, [9]Schulden_pro_kopf_new
```

```
# Erstellen der Vergleichstabelle
schulden_individuell %>%
  filter( !is.na(Einwohner) & !is.na(Regionalschlues
  mutate(Schulden_gesamt = ifelse(is.na(as.numeric(S
                                  as.numeric(gsub("[
                                  as.numeric(Schulde
  mutate(Schulden_pro_kopf = ifelse(is.na(as.numeric
                                    as.numeric(gsub(
                                    as.numeric(Schul
  mutate(Einwohner_num = ifelse(is.na(as.numeric(Ein
                                as.numeric(gsub("[[:
                                as.numeric(Einwohner
  mutate(Schulden_pro_kopf_new = round(Schulden_gesa
  mutate(landkreis = str_extract(Regionalschluessel,
         differenz = Schulden_pro_kopf - Schulden_pr
schulden_consistency
```

Interne Validität Schulden pro Kopf

```
range(schulden_consistency$differenz, na.rm=TRUE)
```

```
## [1] -0.49  0.50
```

Interne Validität Schulden pro Kopf

```
range(schulden_consistency$differenz, na.rm=TRUE)
```

```
## [1] -0.49  0.50
```

Die Differenzen liegen zwischen +/- 50 Cent

Interne Validität Schulden pro Kopf

Es gibt 12 nicht verfügbaren Werte:

```
filter(schulden_consistency, is.na(differenz))
```

```
## # A tibble: 12 × 11
##    Regionalsch…[1] Gemei…[2] Verwa…[3] Einwo…[4] Schul…[5] Schul…[6] Bunde…[7] Einwo…[8] Schul…[9]
##    <chr>           <chr>     <chr>     <chr>        <dbl>     <dbl> <chr>        <dbl>     <dbl>
##  1 033519501_Su… Lohhei… Samtge… {768}      NA            NA NI             768        NA
##  2 033589501_Su… Osterh… Samtge… {2 884}    NA            NA NI            2884        NA
##  3 052           Landsc… Landsc… X           7.37e8       NA NW              NA        NA
##  4 056           Landsc… Landsc… X           6.72e8       NA NW              NA        NA
##  5 058           Region… Kommun… X           1.32e8       NA NW              NA        NA
##  6 067           Verwal… Landes… X           3.00e7       NA HE              NA        NA
##  7 074           Bezirk… Bezirk… X           1.40e8       NA RP              NA        NA
##  8 081a          Landes… Landes… X           5.32e6       NA BW              NA        NA
##  9 081b          Kommun… Landes… X           0            NA BW              NA        NA
## 10 091785127_Su… Allers… Verwal… {7 190}    NA            NA BY            7190        NA
## 11 095725512_Su… Aurach… Verwal… {4 368}    NA            NA BY            4368        NA
## 12 144           Kommun… Landes… X           1.00e7       NA SN              NA        NA
## # … with 2 more variables: landkreis <chr>, differenz <dbl>, and abbreviated
## #   variable names [1]Regionalschluessel, [2]Gemeinde, [3]Verwaltungsform,
```

Bruttoinlandsprodukt

Informationen bzgl. des Bruttoinlandsprodukts

Nach dem Download bei den Statistischen Ämtern des Bundes und der Länder und einer ersten Betrachtung interessieren uns folgende Tabellenblätter:

- Betrachten der Daten
  - Tabellenblatt "1.1" ist für unsere Analyse ausschlaggebend (für das BIP)
  - Tabellenblatt "3.1" ist für die Anzahl an Erwerbstätigen ausschlaggebend
  - Tabellenblatt "5." ist für die Anzahl an Einwohnern ausschlaggebend
- Die ersten vier Zeilen benötigen wir nicht
- Die letzte Zeile enthält eine kurze Beschreibung die wir nicht benötigen
  - **Lösung:** Behalte alle Zeilen, welche bei der `Lfd. Nr.` numerisch sind
- Die folgenden Variablen benötigen wir nicht für unsere Analyse und können entfernt werden: `Lfd. Nr.`, `EU-Code`, `NUTS 1`, `NUTS 2`, `NUTS 3`, `Land`, `Gebietseinheit`

Informationen bzgl. des Bruttoinlandsprodukts

```
# Blatt 1.1 einlesen und die ersten 4 Zeilen skippen
bip <- read_xlsx("../case-study/data/BIP_2021.xlsx", sheet="1.1", skip = 4)
erwerb <- read_xlsx("../case-study/data/BIP_2021.xlsx", sheet="3.1", skip = 4)
einwohner <- read_xlsx("../case-study/data/BIP_2021.xlsx", sheet = "5.", skip = 4)
```

```
# Zeile löschen in der die `Lfd. Nr.` nicht nummeris
# Zusätzliche Spalten löschen
bip
```

```
## # A tibble: 448 × 36
##    Lfd. Nr…¹ EU-Co…² Regio…³ Land  NUTS …⁴ NUTS …⁵ NUTS …⁶ Gebie…⁷ `1992` `199
##    <chr>     <chr>   <chr>   <chr> <chr>   <chr>   <chr>   <chr>   <chr>  <chr>
##  1 <NA>      <NA>    <NA>    <NA>  <NA>    <NA>    <NA>    <NA>    <NA>   <NA>
##  2 1         DE1     08      BW    1       <NA>    <NA>    Baden-… 25586… 2626
##  3 2         DE11    081     BW    <NA>    2       <NA>    Stuttg… 11097… 1116
##  4 3         DE111   08111   BW    <NA>    <NA>    3       Stuttg… 32946… 3173
##  5 4         DE112   08115   BW    <NA>    <NA>    3       Böblin… 12090… 1183
##  6 5         DE113   08116   BW    <NA>    <NA>    3       Esslin… 12275… 1248
##  7 6         DE114   08117   BW    <NA>    <NA>    3       Göppin… 5062.… 5180
##  8 7         DE115   08118   BW    <NA>    <NA>    3       Ludwig… 11714… 1216
##  9 8         DE116   08119   BW    <NA>    <NA>    3       Rems-M… 8500.… 8723
## 10 9         DE117   08121   BW    <NA>    <NA>    3       Heilbr… 4219.… 4387
## # … with 438 more rows, 26 more variables: `1995` <chr>, `1996` <chr>,
## #   `1997` <chr>, `1998` <chr>, `1999` <chr>, `2000` <dbl>, `2001` <dbl>,
## #   `2002` <dbl>, `2003` <dbl>, `2004` <dbl>, `2005` <dbl>, `2006` <dbl>,
## #   `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>, `2011` <dbl>,
## #   `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>,
## #   `2017` <dbl>, `2018` <dbl>, `2019` <dbl>, `2020` <dbl>, and abbreviated
## #   variable names ¹`Lfd. Nr.`, ²`EU-Code`, ³`Regional-schlüssel`, ⁴`NUTS 1`,
```

```r
# Zeile löschen in der die `Lfd. Nr.` nicht nummeris
# Zusätzliche Spalten löschen
bip %>%
  filter(is.na(as.numeric(`Lfd. Nr.`))==FALSE)
```

```
## # A tibble: 445 × 36
##    Lfd. Nr…¹ EU-Co…² Regio…³ Land  NUTS …⁴ NUTS …⁵ NUTS …⁶ Gebie…⁷ `1992` `199
##    <chr>     <chr>   <chr>   <chr> <chr>   <chr>   <chr>   <chr>   <chr>  <chr
##  1 1         DE1     08      BW    1       <NA>    <NA>    Baden-… 25586… 2626
##  2 2         DE11    081     BW    <NA>    2       <NA>    Stuttg… 11097… 1116
##  3 3         DE111   08111   BW    <NA>    <NA>    3       Stuttg… 32946… 3173
##  4 4         DE112   08115   BW    <NA>    <NA>    3       Böblin… 12090… 1183
##  5 5         DE113   08116   BW    <NA>    <NA>    3       Esslin… 12275… 1248
##  6 6         DE114   08117   BW    <NA>    <NA>    3       Göppin… 5062.… 5180
##  7 7         DE115   08118   BW    <NA>    <NA>    3       Ludwig… 11714… 1216
##  8 8         DE116   08119   BW    <NA>    <NA>    3       Rems-M… 8500.… 8723
##  9 9         DE117   08121   BW    <NA>    <NA>    3       Heilbr… 4219.… 4387
## 10 10        DE118   08125   BW    <NA>    <NA>    3       Heilbr… 6073.… 6126
## # … with 435 more rows, 26 more variables: `1995` <chr>, `1996` <chr>,
## #   `1997` <chr>, `1998` <chr>, `1999` <chr>, `2000` <dbl>, `2001` <dbl>,
## #   `2002` <dbl>, `2003` <dbl>, `2004` <dbl>, `2005` <dbl>, `2006` <dbl>,
## #   `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>, `2011` <dbl>,
## #   `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>,
## #   `2017` <dbl>, `2018` <dbl>, `2019` <dbl>, `2020` <dbl>, and abbreviated
## #   variable names ¹`Lfd. Nr.`, ²`EU-Code`, ³`Regional-schlüssel`, ⁴`NUTS 1`,
```

```r
# Zeile löschen in der die `Lfd. Nr.` nicht nummeris
# Zusätzliche Spalten löschen
bip %>%
  filter(is.na(as.numeric(`Lfd. Nr.`))==FALSE) %>%
  select(-c(`Lfd. Nr.`, `EU-Code`, `NUTS 1`, `NUTS 2
```

```
## # A tibble: 445 × 29
##    Regio…¹ `1992` `1994` `1995` `1996` `1997` `1998` `1999` `2000` `2001` `200
##    <chr>   <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>   <dbl>  <dbl>  <db
##  1 08      25586… 26264… 27174… 27677… 28219… 29109… 30072… 3.09e5 3.23e5 3.26
##  2 081     11097… 11160… 11528… 11678… 12086… 12384… 12779… 1.30e5 1.38e5 1.38
##  3 08111   32946… 31736… 32281… 32802… 34339… 33553… 35048… 3.53e4 3.84e4 3.97
##  4 08115   12090… 11833… 11937… 12097… 13919… 13679… 14424… 1.39e4 1.53e4 1.47
##  5 08116   12275… 12482… 12748… 13169… 13284… 13952… 14192… 1.44e4 1.55e4 1.48
##  6 08117   5062.… 5180.… 5447.… 5643.… 5667.… 5838.… 5920.… 6.00e3 6.05e3 6.10
##  7 08118   11714… 12163… 12756… 12895… 13143… 13516… 13866… 1.47e4 1.56e4 1.57
##  8 08119   8500.… 8723.… 9320.… 8780.… 8928.… 9175.… 9707.… 1.04e4 1.04e4 1.05
##  9 08121   4219.… 4387.… 4522.… 4510.… 4581.… 5645.… 5282.… 5.27e3 5.45e3 5.28
## 10 08125   6073.… 6126.… 6577.… 6811.… 7019.… 7645.… 7928.… 8.45e3 8.82e3 8.75
## # … with 435 more rows, 18 more variables: `2003` <dbl>, `2004` <dbl>,
## #   `2005` <dbl>, `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>,
## #   `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
## #   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,
## #   `2020` <dbl>, and abbreviated variable name ¹`Regional-schlüssel`
```

```r
# Zeile löschen in der die `Lfd. Nr.` nicht nummeris
# Zusätzliche Spalten löschen
bip %>%
  filter(is.na(as.numeric(`Lfd. Nr.`))==FALSE) %>%
  select(-c(`Lfd. Nr.`, `EU-Code`, `NUTS 1`, `NUTS 2
  rename(Regionalschluessel = `Regional-schlüssel`)
```

```
## # A tibble: 445 × 29
##    Regio…¹ `1992` `1994` `1995` `1996` `1997` `1998` `1999` `2000` `2001` `200
##    <chr>   <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>   <dbl>  <dbl>  <db
##  1 08      25586… 26264… 27174… 27677… 28219… 29109… 30072… 3.09e5 3.23e5 3.26
##  2 081     11097… 11160… 11528… 11678… 12086… 12384… 12779… 1.30e5 1.38e5 1.38
##  3 08111   32946… 31736… 32281… 32802… 34339… 33553… 35048… 3.53e4 3.84e4 3.97
##  4 08115   12090… 11833… 11937… 12097… 13919… 13679… 14424… 1.39e4 1.53e4 1.47
##  5 08116   12275… 12482… 12748… 13169… 13284… 13952… 14192… 1.44e4 1.55e4 1.48
##  6 08117   5062.… 5180.… 5447.… 5643.… 5667.… 5838.… 5920.… 6.00e3 6.05e3 6.10
##  7 08118   11714… 12163… 12756… 12895… 13143… 13516… 13866… 1.47e4 1.56e4 1.57
##  8 08119   8500.… 8723.… 9320.… 8780.… 8928.… 9175.… 9707.… 1.04e4 1.04e4 1.05
##  9 08121   4219.… 4387.… 4522.… 4510.… 4581.… 5645.… 5282.… 5.27e3 5.45e3 5.28
## 10 08125   6073.… 6126.… 6577.… 6811.… 7019.… 7645.… 7928.… 8.45e3 8.82e3 8.75
## # … with 435 more rows, 18 more variables: `2003` <dbl>, `2004` <dbl>,
## #   `2005` <dbl>, `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>,
## #   `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
## #   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,
## #   `2020` <dbl>, and abbreviated variable name ¹Regionalschluessel
```

```
# Zeile löschen in der die `Lfd. Nr.` nicht nummeris
# Zusätzliche Spalten löschen
bip %>%
  filter(is.na(as.numeric(`Lfd. Nr.`))==FALSE) %>%
  select(-c(`Lfd. Nr.`, `EU-Code`, `NUTS 1`, `NUTS 2
  rename(Regionalschluessel = `Regional-schlüssel`)
bip_wide
```

Informationen bzgl. des Bruttoinlandsprodukts

> Was ist hier eine Beobachtung?

Informationen bzgl. des Bruttoinlandsprodukts

> Was ist hier eine Beobachtung?

Entsprechend können wir bei den Erwerbstätigen und den Einwohnern vorgehen:

```
# Zeile löschen in der die `Lfd. Nr.` nicht nummerisch ist
# Zusätzliche Spalten löschen
erwerb_wide <- erwerb %>%
  filter(is.na(as.numeric(`Lfd. Nr.`))==FALSE) %>%
  select(-c(`Lfd. Nr.`, `EU-Code`, `NUTS 1`, `NUTS 2`, `NUTS 3`, Land, Gebietseinheit)) %>%
  rename(Regionalschluessel = `Regional-schlüssel`)

einwohner_wide <- einwohner %>%
  filter(is.na(as.numeric(`Lfd. Nr.`))==FALSE) %>%
  select(-c(`Lfd. Nr.`, `EU-Code`, `NUTS 1`, `NUTS 2`, `NUTS 3`, Land, Gebietseinheit)) %>%
  rename(Regionalschluessel = `Regional-schlüssel`)
```

Informationen bzgl. des Bruttoinlandsprodukts

Datensatz,

- ist ein Panel: Mehrere Jahre für mehrere Landkreise in Deutschland vorhanden
- ist im `wide` Format -> d.h. die Daten sind nicht `tidy`

```
head(bip_wide, 3)
```

```
## # A tibble: 3 × 29
##   Region…¹ `1992` `1994` `1995` `1996` `1997` `1998` `1999` `2000` `2001` `2002`
##   <chr>    <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>    <dbl>  <dbl>  <dbl>
## 1 08       25586… 26264… 27174… 27677… 28219… 29109… 30072… 3.09e5 3.23e5 3.26e5
## 2 081      11097… 11160… 11528… 11678… 12086… 12384… 12779… 1.30e5 1.38e5 1.38e5
## 3 08111    32946… 31736… 32281… 32802… 34339… 33553… 35048… 3.53e4 3.84e4 3.97e4
## # … with 18 more variables: `2003` <dbl>, `2004` <dbl>, `2005` <dbl>,
## #   `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>,
## #   `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>,
## #   `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>, `2020` <dbl>, and
## #   abbreviated variable name ¹Regionalschluessel
```

Informationen bzgl. des Bruttoinlandsprodukts

Datensatz,

- ist ein Panel: Mehrere Jahre für mehrere Landkreise in Deutschland vorhanden
- ist im `wide` Format -> d.h. die Daten sind nicht `tidy`

```
head(bip_wide, 3)
```

```
## # A tibble: 3 × 29
##   Region…¹ `1992` `1994` `1995` `1996` `1997` `1998` `1999` `2000` `2001` `2002`
##   <chr>    <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>    <dbl>  <dbl>  <dbl>
## 1 08       25586… 26264… 27174… 27677… 28219… 29109… 30072… 3.09e5 3.23e5 3.26e5
## 2 081      11097… 11160… 11528… 11678… 12086… 12384… 12779… 1.30e5 1.38e5 1.38e5
## 3 08111    32946… 31736… 32281… 32802… 34339… 33553… 35048… 3.53e4 3.84e4 3.97e4
## # … with 18 more variables: `2003` <dbl>, `2004` <dbl>, `2005` <dbl>,
## #   `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>,
## #   `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>,
## #   `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>, `2020` <dbl>, and
## #   abbreviated variable name ¹Regionalschluessel
```

> Was sind die Bedigungen für einen `tidy` Datensatz?

Daten in das `long`-Format überführen

Datensatz ins `long`-Format überführen mit `pivot_longer`:

```
bip_long <- pivot_longer(bip_wide, cols = c("1992":"2019") , names_to = "Jahr", values_to = "BIP")
```

```
Fehler: Can't combine `1992` <character> and `2000` <double>.
```

Daten in das `long`-Format überführen

BIP sollte normalerweise nummerisch sein:

- Klasse `double` sollte korrekt sein
- umformatieren der Spalten `1992` - `1999`
- mit `across()` kann der `mutate()`-Befehl über mehrere Spalten angewendet werden

```
#BIP von 1992 - 1999 umformen (als numerische Variab
```

```
#BIP von 1992 - 1999 umformen (als numerische Variab
bip_wide
```

```
## # A tibble: 445 × 29
##    Regio…[1] `1992` `1994` `1995` `1996` `1997` `1998` `1999` `2000` `2001` `200
##    <chr>   <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <dbl>  <dbl>  <db
##  1 08      25586… 26264… 27174… 27677… 28219… 29109… 30072… 3.09e5 3.23e5 3.26
##  2 081     11097… 11160… 11528… 11678… 12086… 12384… 12779… 1.30e5 1.38e5 1.38
##  3 08111   32946… 31736… 32281… 32802… 34339… 33553… 35048… 3.53e4 3.84e4 3.97
##  4 08115   12090… 11833… 11937… 12097… 13919… 13679… 14424… 1.39e4 1.53e4 1.47
##  5 08116   12275… 12482… 12748… 13169… 13284… 13952… 14192… 1.44e4 1.55e4 1.48
##  6 08117   5062.… 5180.… 5447.… 5643.… 5667.… 5838.… 5920.… 6.00e3 6.05e3 6.10
##  7 08118   11714… 12163… 12756… 12895… 13143… 13516… 13866… 1.47e4 1.56e4 1.57
##  8 08119   8500.… 8723.… 9320.… 8780.… 8928.… 9175.… 9707.… 1.04e4 1.04e4 1.05
##  9 08121   4219.… 4387.… 4522.… 4510.… 4581.… 5645.… 5282.… 5.27e3 5.45e3 5.28
## 10 08125   6073.… 6126.… 6577.… 6811.… 7019.… 7645.… 7928.… 8.45e3 8.82e3 8.75
## # … with 435 more rows, 18 more variables: `2003` <dbl>, `2004` <dbl>,
## #   `2005` <dbl>, `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>,
## #   `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
## #   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,
## #   `2020` <dbl>, and abbreviated variable name [1]Regionalschluessel
```

```
#BIP von 1992 - 1999 umformen (als numerische Variab
bip_wide %>%
  select(`1992`:`1999`)
```

```
## # A tibble: 445 × 7
##    `1992`             `1994`             `1995`      `1996` `1997` `1998` `199
##    <chr>              <chr>              <chr>       <chr>  <chr>  <chr>  <chr
##  1 255866.41899999999 262645.41600000003 271746.699… 27677… 28219… 29109… 3007
##  2 110977.071         111602.66499999999 115280.807  11678… 12086… 12384… 1277
##  3 32946.883999999998 31736.567999999999 32281.0040… 32802… 34339… 33553… 3504
##  4 12090.93           11833.816000000001 11937.788   12097… 13919… 13679… 1442
##  5 12275.605          12482.948          12748.703   13169… 13284… 13952… 1419
##  6 5062.0370000000003 5180.0739999999996 5447.49399… 5643.… 5667.… 5838.… 5920
##  7 11714.16           12163.822          12756.3989… 12895… 13143… 13516… 1386
##  8 8500.4050000000007 8723.0990000000002 9320.15600… 8780.… 8928.… 9175.… 9707
##  9 4219.259           4387.4809999999998 4522.82399… 4510.… 4581.… 5645.… 5282
## 10 6073.5249999999996 6126.3310000000001 6577.05599… 6811.… 7019.… 7645.… 7928
## # … with 435 more rows
```

```r
#BIP von 1992 - 1999 umformen (als numerische Variab
bip_wide %>%
  select(`1992`:`1999`) %>%
  mutate(across(is.character, as.double))
```

```
## # A tibble: 445 × 7
##     `1992`  `1994`  `1995`  `1996`  `1997`  `1998`  `1999`
##      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 255866. 262645. 271747. 276777. 282190. 291100. 300727.
##  2 110977. 111603. 115281. 116787. 120867. 123842. 127799.
##  3  32947.  31737.  32281.  32803.  34340.  33553.  35048.
##  4  12091.  11834.  11938.  12097.  13919.  13679.  14424.
##  5  12276.  12483.  12749.  13169.  13285.  13952.  14192.
##  6   5062.   5180.   5447.   5643.   5668.   5839.   5920.
##  7  11714.  12164.  12756.  12895.  13144.  13516.  13867.
##  8   8500.   8723.   9320.   8781.   8928.   9176.   9708.
##  9   4219.   4387.   4523.   4511.   4581.   5646.   5282.
## 10   6074.   6126.   6577.   6812.   7020.   7646.   7929.
## # … with 435 more rows
```

```
#BIP von 1992 - 1999 umformen (als numerische Variab
bip_wide %>%
  select(`1992`:`1999`) %>%
  mutate(across(is.character, as.double)) ->
bip_double
```

Entsprechend dann bei den Einwohnern und Erwerbstätigen:

> Es wird eine Warnmeldung ausgegeben das `NA`s bei der Umwandlung erzeugt wurden. Warum?

```
# Erwerbstätige von 1992 - 1999 umformen (als numerische Variable)
erwerb_double <- erwerb_wide %>%
  select(`1992`:`1999`) %>%
  mutate(across(is.character, as.double))
```

```
## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt

## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt

## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt

## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt

## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt

## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt

## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt
```

```
# Einwohner von 1992 - 1999 umformen (als numerische Variable)
einwohner_double <- einwohner_wide %>%
  select(`1992`:`1999`) %>%
  mutate(across(is.character, as.double))
```

Daten in das `long`-Format überführen

Wir überprüfen, welche Spalten die Warnung hervorgerufen haben und wo `NA`s erzeugt wurden

```
bip_wide_test <- bip_wide %>%
  bind_cols(bip_double)

head(filter(bip_wide_test, is.na(`1994...31`)))
```

```
## # A tibble: 6 × 36
##   Region…¹ 1992.…² 1994.…³ 1995.…⁴ 1996.…⁵ 1997.…⁶ 1998.…⁷ 1999.…⁸ `2000` `2001`
##   <chr>    <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>    <dbl>  <dbl>
## 1 13003    .       .       .       .       .       .       .        4767.  4550.
## 2 13004    .       .       .       .       .       .       .        2611.  2695.
## 3 13071    .       .       .       .       .       .       .        5269.  5373.
## 4 13072    .       .       .       .       .       .       .        3570.  3617.
## 5 13073    .       .       .       .       .       .       .        3608.  3631.
## 6 13074    .       .       .       .       .       .       .        2393.  2453.
## # … with 26 more variables: `2002` <dbl>, `2003` <dbl>, `2004` <dbl>,
## #   `2005` <dbl>, `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>,
## #   `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
## #   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,
## #   `2020` <dbl>, `1992...30` <dbl>, `1994...31` <dbl>, `1995...32` <dbl>,
```

Eine Umwandlung zu `NA` geschieht bei den Werten bei denen – eingetragen wurde. D.h. für uns ist es ok hier ein `NA` einzutragen. Somit können wir die Umwandlung in die Klasse `double` durchführen:

```
bip_wide <- bip_wide %>%
  select(-(`1992`:`1999`)) %>%
  bind_cols(bip_double)

erwerb_wide <- erwerb_wide %>%
  select(-(`1992`:`1999`)) %>%
  bind_cols(erwerb_double)

einwohner_wide <- einwohner_wide %>%
  select(-(`1992`:`1999`)) %>%
  bind_cols(einwohner_double)
```

Daten in das `long`-Format überführen

Nun können wir den Datensatz ins `long`-Format transferieren und nach dem Jahr sortieren.

- Einwohner und Erwerbstätigen in 1000 Personen angegeben, daher Erwerbstätigen und Einwohner mit 1000 multiplizieren.
- BIP ist in 1 Mio. Euro angegeben, daher die Multiplikation mit 1 Mio.

```r
# BIP ins long-Format
pivot_longer(bip_wide, cols = c("2000":"1999") , nam
```

```
## # A tibble: 12,460 × 3
##    Regionalschluessel Jahr      bip
##    <chr>              <chr>   <dbl>
##  1 08                 2000  308823.
##  2 08                 2001  323078.
##  3 08                 2002  325510.
##  4 08                 2003  329164.
##  5 08                 2004  333276.
##  6 08                 2005  335789.
##  7 08                 2006  357283.
##  8 08                 2007  377021.
##  9 08                 2008  381903.
## 10 08                 2009  353463.
## # … with 12,450 more rows
```

```r
# BIP ins long-Format
pivot_longer(bip_wide, cols = c("2000":"1999") , nam
  mutate( Jahr = as.numeric(Jahr),
          bip = bip * 1000000)
```

```
## # A tibble: 12,460 × 3
##    Regionalschluessel  Jahr          bip
##    <chr>              <dbl>        <dbl>
##  1 08                  2000 308822815000
##  2 08                  2001 323077717000
##  3 08                  2002 325510403000
##  4 08                  2003 329164078000
##  5 08                  2004 333275845000
##  6 08                  2005 335788716000
##  7 08                  2006 357283378000
##  8 08                  2007 377021382000
##  9 08                  2008 381902739000
## 10 08                  2009 353462984000
## # … with 12,450 more rows
```

```r
# BIP ins long-Format
pivot_longer(bip_wide, cols = c("2000":"1999") , nam
  mutate( Jahr = as.numeric(Jahr),
          bip = bip * 1000000) %>%
  arrange( Jahr )
```

```
## # A tibble: 12,460 × 3
##    Regionalschluessel  Jahr          bip
##    <chr>              <dbl>        <dbl>
##  1 08                  1992 255866419000
##  2 081                 1992 110977071000
##  3 08111               1992  32946884000
##  4 08115               1992  12090930000
##  5 08116               1992  12275605000
##  6 08117               1992   5062037000
##  7 08118               1992  11714160000
##  8 08119               1992   8500405000
##  9 08121               1992   4219259000
## 10 08125               1992   6073525000
## # … with 12,450 more rows
```

```
# BIP ins long-Format
pivot_longer(bip_wide, cols = c("2000":"1999") , nam
  mutate( Jahr = as.numeric(Jahr),
          bip = bip * 1000000) %>%
  arrange( Jahr ) ->
bip_long
```

Für die Erwerbstätigen und Einwohner entsprechend:

```
# Anzahl der Erwerbstätigen ins long-Format
erwerb_long <- pivot_longer(erwerb_wide, cols = c("2000":"1999") , names_to = "Jahr", values_to = "erw") %>%
  mutate( Jahr = as.numeric(Jahr),
          erw = erw * 1000) %>%
  arrange( Jahr )

# Anzahl der Einwohner ins long-Format
einwohner_long <- pivot_longer(einwohner_wide, cols = c("2000":"1999") , names_to = "Jahr", values_to = "einw
  mutate( Jahr = as.numeric(Jahr),
          einwohner = einwohner * 1000) %>%
  arrange( Jahr )
```

Konsistenzchecks

Hier sollten Sie selbst aktiv werden und die Daten auf Konsistenz prüfen:

> Als Konsistenzcheck könnten Sie hier die Anzahl der Einwohner aus den verschiedenen Datensätzen vergleichen.

Kartenmaterial hinzufügen

Wir benötigen hier eine Karte von Deutschland mit den einzelnen Verwaltungsgrenzen als SHAPE-File und können diese mittels des `sf`-Pakets einlesen.

Das OpenData Portal des Bundesamts für Kartographie und Geodäsie stellt die nötigen Informationen kostenlos zur Verfügung.

Die Dokumentation der Daten sollten wir uns immer zuerst anschauen, bevor wir die Datenquelle herunterladen.

Dies gilt nicht nur für die Geodaten, sondern allgemein für alle Datenreihen.

> Bitte versuchen Sie selbst die Daten herunterzuladen und anhand des Regionalschlüssels (ARS) mit dem BIP, den Arbeitslosen und den Schulden zusammenzuführen.

Datensätze zusammenführen

> Nun möchten wir die unterschiedlichen Datensätze noch zu einem zusammenfügen!

Zuerst müssen wir folgende Schritte unternehmen:

- ✚ Informationen zur Verschuldung auf Landkreisebene aggregieren
- ✚ Daten zum BIP auf das Jahr 2017 einschränken.
- ✚ Datensätze anhand des Regionalschlüssels miteinander verbinden.

Weiterhin können wir die geografischen Daten separat abspeichern und bei Bedarf anhand des Regionalschlüssels zu unserem Datensatz hinzumergen.

```
# Schulden auf Landkreisebene
schulden_bereinigt
```

```
## # A tibble: 11,050 × 8
##    Regionalschluessel Gemeinde   Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵ landk
##    <chr>              <chr>      <chr>     <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 010010000000       Flensburg… kreisf…   87770  5.08e8   5791. SH      01001
##  2 010020000000       Kiel, Lan… kreisf…  247135  9.49e8   3839. SH      01002
##  3 010030000000       Lübeck, H… kreisf…  216739  1.21e9   5567. SH      01003
##  4 010040000000       Neumünste… kreisf…   78759  4.26e8   5409. SH      01004
##  5 010510011011       Brunsbütt… amtsfr…   12781  3.49e7   2733. SH      01051
##  6 010510044044       Heide, St… amtsfr…   21508  3.71e7   1723. SH      01051
##  7 010515163003       Averlak    amtsan…     576  1.20e6   2079. SH      01051
##  8 010515163010       Brickeln   amtsan…     219  4.96e5   2266. SH      01051
##  9 010515163012       Buchholz   amtsan…    1008  1.23e6   1220. SH      01051
## 10 010515163016       Burg (Dit… amtsan…    4114  7.92e6   1926. SH      01051
## # … with 11,040 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland, ⁶landkreis
```

```
# Schulden auf Landkreisebene
schulden_bereinigt %>%
  group_by(landkreis)
```

```
## # A tibble: 11,050 × 8
## # Groups:   landkreis [397]
##    Regionalschluessel Gemeinde   Verwa…¹ Einwo…² Schul…³ Schul…⁴ Bunde…⁵ landk
##    <chr>              <chr>      <chr>     <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 010010000000       Flensburg… kreisf…   87770  5.08e8   5791. SH      01001
##  2 010020000000       Kiel, Lan… kreisf…  247135  9.49e8   3839. SH      01002
##  3 010030000000       Lübeck, H… kreisf…  216739  1.21e9   5567. SH      01003
##  4 010040000000       Neumünste… kreisf…   78759  4.26e8   5409. SH      01004
##  5 010510011011       Brunsbütt… amtsfr…   12781  3.49e7   2733. SH      01051
##  6 010510044044       Heide, St… amtsfr…   21508  3.71e7   1723. SH      01051
##  7 010515163003       Averlak    amtsan…     576  1.20e6   2079. SH      01051
##  8 010515163010       Brickeln   amtsan…     219  4.96e5   2266. SH      01051
##  9 010515163012       Buchholz   amtsan…    1008  1.23e6   1220. SH      01051
## 10 010515163016       Burg (Dit… amtsan…    4114  7.92e6   1926. SH      01051
## # … with 11,040 more rows, and abbreviated variable names ¹Verwaltungsform,
## #   ²Einwohner, ³Schulden_gesamt, ⁴Schulden_pro_kopf, ⁵Bundesland, ⁶landkreis
```

```r
# Schulden auf Landkreisebene
schulden_bereinigt %>%
  group_by(landkreis) %>%
  summarise( Schulden_pro_kopf_lk = sum(Schulden_ges
             Einwohner = sum(Einwohner),
             Schulden_gesamt = sum(Schulden_gesamt))
```

```
## # A tibble: 397 × 4
##    landkreis Schulden_pro_kopf_lk Einwohner Schulden_gesamt
##    <chr>                    <dbl>     <dbl>           <dbl>
##  1 01001                    5791.     87770       508281539
##  2 01002                    3839.    247135       948848421
##  3 01003                    5567.    216739      1206620094
##  4 01004                    5409.     78759       426019276
##  5 01051                    1670.    133684       223191181
##  6 01053                    1293.    195677       252944185
##  7 01054                    2624.    165642       434624906
##  8 01055                    1890.    200931       379698731
##  9 01056                    2225.    311713       693581474
## 10 01057                    1532.    128763       197203962
## # … with 387 more rows
```

```r
# Schulden auf Landkreisebene
schulden_bereinigt %>%
  group_by(landkreis) %>%
  summarise( Schulden_pro_kopf_lk = sum(Schulden_ges
             Einwohner = sum(Einwohner),
             Schulden_gesamt = sum(Schulden_gesamt))
  rename(Regionalschluessel = landkreis)
```

```
## # A tibble: 397 × 4
##    Regionalschluessel Schulden_pro_kopf_lk Einwohner Schulden_gesamt
##    <chr>                             <dbl>     <dbl>           <dbl>
##  1 01001                             5791.     87770       508281539
##  2 01002                             3839.    247135       948848421
##  3 01003                             5567.    216739      1206620094
##  4 01004                             5409.     78759       426019276
##  5 01051                             1670.    133684       223191181
##  6 01053                             1293.    195677       252944185
##  7 01054                             2624.    165642       434624906
##  8 01055                             1890.    200931       379698731
##  9 01056                             2225.    311713       693581474
## 10 01057                             1532.    128763       197203962
## # … with 387 more rows
```

```
# Schulden auf Landkreisebene
schulden_bereinigt %>%
  group_by(landkreis) %>%
  summarise( Schulden_pro_kopf_lk = sum(Schulden_ges
             Einwohner = sum(Einwohner),
             Schulden_gesamt = sum(Schulden_gesamt))
  rename(Regionalschluessel = landkreis) ->
schulden_kombi
```

```r
# Anzahl an Erwerbstätigen für das Jahr 2017
erwerb_long
```

```
## # A tibble: 12,460 × 3
##    Regionalschluessel  Jahr     erw
##    <chr>              <dbl>   <dbl>
##  1 08                  1992 5230587
##  2 081                 1992 2046858
##  3 08111               1992  486895
##  4 08115               1992  188312
##  5 08116               1992  237498
##  6 08117               1992  118140
##  7 08118               1992  223059
##  8 08119               1992  176939
##  9 08121               1992   94510
## 10 08125               1992  115906
## # … with 12,450 more rows
```

```r
# Anzahl an Erwerbstätigen für das Jahr 2017
erwerb_long %>%
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
```

```
## # A tibble: 399 × 3
##    Regionalschluessel  Jahr    erw
##    <chr>              <dbl>  <dbl>
##  1 08111               2017 529355
##  2 08115               2017 230023
##  3 08116               2017 286644
##  4 08117               2017 121597
##  5 08118               2017 263814
##  6 08119               2017 204535
##  7 08121               2017  97270
##  8 08125               2017 176734
##  9 08126               2017  70782
## 10 08127               2017 110285
## # … with 389 more rows
```

```r
# Anzahl an Erwerbstätigen für das Jahr 2017
erwerb_long %>%
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
  select(-Jahr)
```

```
## # A tibble: 399 × 2
##    Regionalschluessel    erw
##    <chr>               <dbl>
##  1 08111              529355
##  2 08115              230023
##  3 08116              286644
##  4 08117              121597
##  5 08118              263814
##  6 08119              204535
##  7 08121               97270
##  8 08125              176734
##  9 08126               70782
## 10 08127              110285
## # … with 389 more rows
```

```
# Anzahl an Erwerbstätigen für das Jahr 2017
erwerb_long %>%
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
  select(-Jahr) ->
erwerb_kombi
```

```
# Anzahl an Einwohner für das Jahr 2017
einwohner_long
```

```
## # A tibble: 12,460 × 3
##    Regionalschluessel  Jahr einwohner
##    <chr>              <dbl>     <dbl>
##  1 08                  1992  10050431
##  2 081                 1992   3771006
##  3 08111               1992    593628
##  4 08115               1992    343190
##  5 08116               1992    487370
##  6 08117               1992    248688
##  7 08118               1992    475248
##  8 08119               1992    389670
##  9 08121               1992    118566
## 10 08125               1992    283163
## # … with 12,450 more rows
```

```r
# Anzahl an Einwohner für das Jahr 2017
einwohner_long %>%
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
```

```
## # A tibble: 399 × 3
##    Regionalschluessel  Jahr einwohner
##    <chr>              <dbl>     <dbl>
##  1 08111               2017    630388
##  2 08115               2017    387718
##  3 08116               2017    530620
##  4 08117               2017    255482
##  5 08118               2017    540266
##  6 08119               2017    423788
##  7 08121               2017    124442
##  8 08125               2017    339172
##  9 08126               2017    111041
## 10 08127               2017    193581
## # … with 389 more rows
```

```r
# Anzahl an Einwohner für das Jahr 2017
einwohner_long %>%
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
  select(-Jahr)
```

```
## # A tibble: 399 × 2
##    Regionalschluessel einwohner
##    <chr>                  <dbl>
##  1 08111                 630388
##  2 08115                 387718
##  3 08116                 530620
##  4 08117                 255482
##  5 08118                 540266
##  6 08119                 423788
##  7 08121                 124442
##  8 08125                 339172
##  9 08126                 111041
## 10 08127                 193581
## # … with 389 more rows
```

```
# Anzahl an Einwohner für das Jahr 2017
einwohner_long %>%
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
  select(-Jahr) ->
einwohner_kombi
```

```
# Anzahl der Einwohner mit dem BIP verbinden um das
left_join(bip_long, einwohner_long, by=c("Regionalsc
```

```
## # A tibble: 12,460 × 4
##    Regionalschluessel  Jahr          bip einwohner
##    <chr>              <dbl>        <dbl>     <dbl>
##  1 08                  1992 255866419000  10050431
##  2 081                 1992 110977071000   3771006
##  3 08111               1992  32946884000    593628
##  4 08115               1992  12090930000    343190
##  5 08116               1992  12275605000    487370
##  6 08117               1992   5062037000    248688
##  7 08118               1992  11714160000    475248
##  8 08119               1992   8500405000    389670
##  9 08121               1992   4219259000    118566
## 10 08125               1992   6073525000    283163
## # … with 12,450 more rows
```

```
# Anzahl der Einwohner mit dem BIP verbinden um das
left_join(bip_long, einwohner_long, by=c("Regionalsc
  mutate(bip_pro_kopf = bip / einwohner)
```

```
## # A tibble: 12,460 × 5
##    Regionalschluessel  Jahr          bip einwohner bip_pro_kopf
##    <chr>              <dbl>        <dbl>     <dbl>        <dbl>
##  1 08                  1992 255866419000  10050431       25458.
##  2 081                 1992 110977071000   3771006       29429.
##  3 08111               1992  32946884000    593628       55501.
##  4 08115               1992  12090930000    343190       35231.
##  5 08116               1992  12275605000    487370       25187.
##  6 08117               1992   5062037000    248688       20355.
##  7 08118               1992  11714160000    475248       24649.
##  8 08119               1992   8500405000    389670       21814.
##  9 08121               1992   4219259000    118566       35586.
## 10 08125               1992   6073525000    283163       21449.
## # … with 12,450 more rows
```

```r
# Anzahl der Einwohner mit dem BIP verbinden um das
left_join(bip_long, einwohner_long, by=c("Regionalsc
  mutate(bip_pro_kopf = bip / einwohner) %>%
# BIP auf Landkreisebene im Jahr 2017
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
```

```
## # A tibble: 399 × 5
##    Regionalschluessel  Jahr         bip einwohner bip_pro_kopf
##    <chr>              <dbl>       <dbl>     <dbl>        <dbl>
##  1 08111               2017 54898140000    630388       87086.
##  2 08115               2017 25653016000    387718       66164.
##  3 08116               2017 22563031000    530620       42522.
##  4 08117               2017  8544665000    255482       33445.
##  5 08118               2017 24948748000    540266       46179.
##  6 08119               2017 14644873000    423788       34557.
##  7 08121               2017  6666905000    124442       53574.
##  8 08125               2017 18355253000    339172       54118.
##  9 08126               2017  5379526000    111041       48446.
## 10 08127               2017  7879086000    193581       40702.
## # … with 389 more rows
```

```r
# Anzahl der Einwohner mit dem BIP verbinden um das
left_join(bip_long, einwohner_long, by=c("Regionalsc
  mutate(bip_pro_kopf = bip / einwohner) %>%
# BIP auf Landkreisebene im Jahr 2017
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
  select(-c(Jahr, einwohner))
```

```
## # A tibble: 399 × 3
##    Regionalschluessel         bip bip_pro_kopf
##    <chr>                    <dbl>        <dbl>
##  1 08111               54898140000       87086.
##  2 08115               25653016000       66164.
##  3 08116               22563031000       42522.
##  4 08117                8544665000       33445.
##  5 08118               24948748000       46179.
##  6 08119               14644873000       34557.
##  7 08121                6666905000       53574.
##  8 08125               18355253000       54118.
##  9 08126                5379526000       48446.
## 10 08127                7879086000       40702.
## # … with 389 more rows
```

```
# Anzahl der Einwohner mit dem BIP verbinden um das
left_join(bip_long, einwohner_long, by=c("Regionalsc
  mutate(bip_pro_kopf = bip / einwohner) %>%
# BIP auf Landkreisebene im Jahr 2017
  filter(nchar(Regionalschluessel) == 5 & Jahr == 20
  select(-c(Jahr, einwohner)) ->
bip_kombi
```

```
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis
```

```
## # A tibble: 401 × 2
##    Regionalschluessel total_alo
##    <chr>                  <dbl>
##  1 01001                   4512
##  2 01002                  12345
##  3 01003                   9692
##  4 01004                   3836
##  5 01051                   4632
##  6 01053                   5592
##  7 01054                   5657
##  8 01055                   5748
##  9 01056                   8599
## 10 01057                   3264
## # … with 391 more rows
```

```r
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
```

```
## # A tibble: 401 × 3
##    Regionalschluessel total_alo bundesland
##    <chr>                  <dbl> <chr>
##  1 01001                   4512 01
##  2 01002                  12345 01
##  3 01003                   9692 01
##  4 01004                   3836 01
##  5 01051                   4632 01
##  6 01053                   5592 01
##  7 01054                   5657 01
##  8 01055                   5748 01
##  9 01056                   8599 01
## 10 01057                   3264 01
## # … with 391 more rows
```

```r
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
```

```
## # A tibble: 401 × 6
##    Regionalschluessel total_alo bundesland Schulden_pro_kopf_lk Einwoh…¹ Schul
##    <chr>                  <dbl> <chr>                      <dbl>    <dbl>   <db
##  1 01001                   4512 01                         5791.    87770  5.08
##  2 01002                  12345 01                         3839.   247135  9.49
##  3 01003                   9692 01                         5567.   216739  1.21
##  4 01004                   3836 01                         5409.    78759  4.26
##  5 01051                   4632 01                         1670.   133684  2.23
##  6 01053                   5592 01                         1293.   195677  2.53
##  7 01054                   5657 01                         2624.   165642  4.35
##  8 01055                   5748 01                         1890.   200931  3.80
##  9 01056                   8599 01                         2225.   311713  6.94
## 10 01057                   3264 01                         1532.   128763  1.97
## # … with 391 more rows, and abbreviated variable names ¹Einwohner,
## #   ²Schulden_gesamt
```

```r
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
  left_join(., bip_kombi, by = "Regionalschluessel")
```

```
## # A tibble: 401 × 8
##    Regionalschluessel total_alo bundes…¹ Schul…² Einwo…³ Schul…⁴     bip bip_p
##    <chr>                  <dbl> <chr>      <dbl>   <dbl>   <dbl>   <dbl>   <db
##  1 01001                   4512 01         5791.   87770  5.08e8 3.67e 9  4166
##  2 01002                  12345 01         3839.  247135  9.49e8 1.14e10  4596
##  3 01003                   9692 01         5567.  216739  1.21e9 9.16e 9  4230
##  4 01004                   3836 01         5409.   78759  4.26e8 3.34e 9  4196
##  5 01051                   4632 01         1670.  133684  2.23e8 4.47e 9  3346
##  6 01053                   5592 01         1293.  195677  2.53e8 4.50e 9  2298
##  7 01054                   5657 01         2624.  165642  4.35e8 5.74e 9  3471
##  8 01055                   5748 01         1890.  200931  3.80e8 5.27e 9  2626
##  9 01056                   8599 01         2225.  311713  6.94e8 9.07e 9  2909
## 10 01057                   3264 01         1532.  128763  1.97e8 2.55e 9  1976
## # … with 391 more rows, and abbreviated variable names ¹bundesland,
## #   ²Schulden_pro_kopf_lk, ³Einwohner, ⁴Schulden_gesamt, ⁵bip_pro_kopf
```

```r
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
  left_join(., bip_kombi, by = "Regionalschluessel")
# Zahl der Erwerbstätigen zumergen
  left_join(., erwerb_kombi, by = "Regionalschluesse
```

```
## # A tibble: 401 × 9
##    Regionalschl…¹ total…² bunde…³ Schul…⁴ Einwo…⁵ Schul…⁶     bip bip_p…⁷    e
##    <chr>            <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <db
##  1 01001             4512 01        5791.   87770  5.08e8 3.67e 9  41669.  598
##  2 01002            12345 01        3839.  247135  9.49e8 1.14e10  45967. 1714
##  3 01003             9692 01        5567.  216739  1.21e9 9.16e 9  42308. 1280
##  4 01004             3836 01        5409.   78759  4.26e8 3.34e 9  41963.  523
##  5 01051             4632 01        1670.  133684  2.23e8 4.47e 9  33469.  605
##  6 01053             5592 01        1293.  195677  2.53e8 4.50e 9  22988.  696
##  7 01054             5657 01        2624.  165642  4.35e8 5.74e 9  34717.  902
##  8 01055             5748 01        1890.  200931  3.80e8 5.27e 9  26264.  905
##  9 01056             8599 01        2225.  311713  6.94e8 9.07e 9  29093. 1294
## 10 01057             3264 01        1532.  128763  1.97e8 2.55e 9  19765.  439
## # … with 391 more rows, and abbreviated variable names ¹Regionalschluessel,
## #   ²total_alo, ³bundesland, ⁴Schulden_pro_kopf_lk, ⁵Einwohner,
## #   ⁶Schulden_gesamt, ⁷bip_pro_kopf
```

```
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
  left_join(., bip_kombi, by = "Regionalschluessel")
# Zahl der Erwerbstätigen zumergen
  left_join(., erwerb_kombi, by = "Regionalschluesse
gesamtdaten
```

```
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
  left_join(., bip_kombi, by = "Regionalschluessel")
# Zahl der Erwerbstätigen zumergen
  left_join(., erwerb_kombi, by = "Regionalschluesse
gesamtdaten

#saveRDS(gesamtdaten, "data/gesamtdaten.rds")  #save
```

```
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
  left_join(., bip_kombi, by = "Regionalschluessel")
# Zahl der Erwerbstätigen zumergen
  left_join(., erwerb_kombi, by = "Regionalschluesse
gesamtdaten

#saveRDS(gesamtdaten, "data/gesamtdaten.rds")  #save
#saveRDS(schulden_bereinigt, "data/schulden_bereinig
```

```
# Datensätze zusammenführen

# Basisdatensatz -> Arbeitslosenzahlen pro Landkreis
# Namen der Bundesländer zumergen
alo_landkreis %>%
  mutate(bundesland = str_extract(Regionalschluessel
  left_join(., schulden_kombi, by = "Regionalschlues
  left_join(., bip_kombi, by = "Regionalschluessel")
# Zahl der Erwerbstätigen zumergen
  left_join(., erwerb_kombi, by = "Regionalschluesse
gesamtdaten

#saveRDS(gesamtdaten, "data/gesamtdaten.rds")  #save
#saveRDS(schulden_bereinigt, "data/schulden_bereinig
#saveRDS(bip_zeitreihe, "data/bip_zeitreihe.rds")  #
```

Übungsaufgaben

Laden Sie sich das durchschnittliche Arbeitnehmerentgelt pro Arbeitnehmer und Landkreis auf der Seite der Statistischen Ämter des Bundes und der Länder herunter und lesen Sie diesen in R ein.

- Finden Sie in dem heruntergeladenen Datensatz heraus, was der Unterschied zwischen *Arbeitnehmerentgelt* und *Bruttolöhne- und Gehälter* ist.
- Lesen Sie die für Sie relevante Tabelle *Bruttolöhne- und Gehälter* in R ein.
- Bereinigen Sie die Tabelle, d.h. der Datensatz sollte danach `tidy` sein.
- Berechnen Sie die Bruttolöhne pro Bundesland mit den Bruttolöhnen der einzelnen Landkreise als Konsistenzcheck.