rstatsZH - Data Science mit R

Mit mehreren Dataframes arbeiten / Tabellen darstellen

Lars Schöbitz

2020-05-02

Rückblick - Woche 6

- Erweiterte Vektoren
  - Faktoren
  - Datums- und Zeitwerte
  - Tibbles
- Daten importieren
  - `read_csv()`
  - `read_excel()`
  - etc.
- SQL in R Markdown
- Tidy Data Konzept
- Daten drehen (pivoting) mit `{tidyr}`
  - `pivot_longer()`
  - `pivot_wider()`

Hausaufgabe 6

Hausaufgabe 6

Hausaufgabe 6 - Lösungen

- **GitHub Organisation:** rstatsZH
  - https://github.com/rstatsZH/
- **Repo:** ha-06-treibhausgase
  - https://github.com/rstatsZH/ha-06-treibhausgase
- **R Markdown Datei:** ha-06-solutions.Rmd
  - https://github.com/rstatsZH/ha-06-treibhausgase/blob/main/ha-06-solutions.Rmd

Ziele für diese Woche

Am Ende dieser Woche könnt ihr:

- Mehrere Dataframes zusammenfügen
- Tabellen mit dem `{gt}` Package erstellen
- Einen mit `ggplot()` erstellten Plot interaktiv darstellen
- Einen R Markdown Bericht als Website publizieren
- Mit den im Kurs erlernten Fähigkeiten selbstständig weiter arbeiten

Mit mehreren Dataframes arbeiten

Wir...

haben mehrere Dataframes

wollen diese zusammenbringen

Data: Women in science

Informationen zu 10 Frauen in der Wissenschaft welche die Welt verändert haben

name
Ada Lovelace
Marie Curie
Janaki Ammal
Chien-Shiung Wu
Katherine Johnson
Rosalind Franklin
Vera Rubin
Gladys West
Flossie Wong-Staal
Jennifer Doudna

Quelle: Discover Magazine

Inputs - Drei Dataframes

professions dates works

```
# A tibble: 10 x 2
   name              profession
   <chr>             <chr>
 1 Ada Lovelace      Mathematician
 2 Marie Curie       Physicist and Chemist
 3 Janaki Ammal      Botanist
 4 Chien-Shiung Wu   Physicist
 5 Katherine Johnson Mathematician
 6 Rosalind Franklin Chemist
 7 Vera Rubin        Astronomer
 8 Gladys West       Mathematician
 9 Flossie Wong-Staal Virologist and Molecular Biologist
10 Jennifer Doudna   Biochemist
```

Gewünschter Output

```
# A tibble: 10 x 5
   name      profession    birth_year death_year known_for
   <chr>     <chr>              <dbl>      <dbl> <chr>
 1 Ada Lov… Mathematician         NA         NA first computer a…
 2 Marie C… Physicist an…         NA         NA theory of radioa…
 3 Janaki … Botanist            1897       1984 hybrid species, …
 4 Chien-S… Physicist           1912       1997 confim and refin…
 5 Katheri… Mathematician       1918       2020 calculations of …
 6 Rosalin… Chemist             1920       1958 <NA>
 7 Vera Ru… Astronomer          1928       2016 existence of dar…
 8 Gladys … Mathematician       1930         NA mathematical mod…
 9 Flossie… Virologist a…       1947         NA first scientist …
10 Jennife… Biochemist          1964         NA one of the prima…
```

Inputs als Erinnerung

```
names(professions)
```

```
[1] "name"       "profession"
```

```
names(dates)
```

```
[1] "name"       "birth_year" "death_y
```

```
names(works)
```

```
[1] "name"      "known_for"
```

```
nrow(professions)
```

```
[1] 10
```

```
nrow(dates)
```

```
[1] 8
```

```
nrow(works)
```

```
[1] 9
```

Dataframes zusammenfügen

Dataframes zusammenfügen

```
abcd_join(x, y)
```

- `left_join()`: alle Reihen aus x
- `right_join()`: alle Reihen aus y
- `full_join()`: alle Reihen aus x und y
- ...

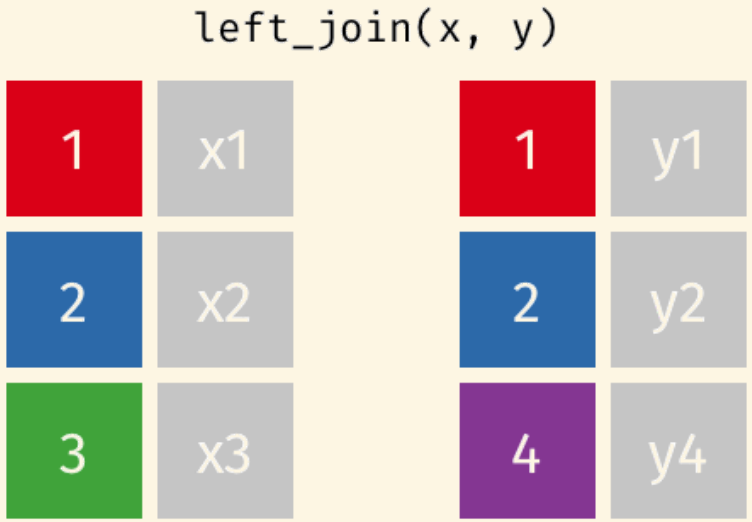
Beispiel

Für die nächsten Folien

```
# A tibble: 3 x 2
     id var_x
  <dbl> <chr>
1     1 x1
2     2 x2
3     3 x3
```

```
# A tibble: 3 x 2
     id var_y
  <dbl> <chr>
1     1 y1
2     2 y2
3     4 y4
```

left_join()




```
left_join(tib_x, tib_y)
```

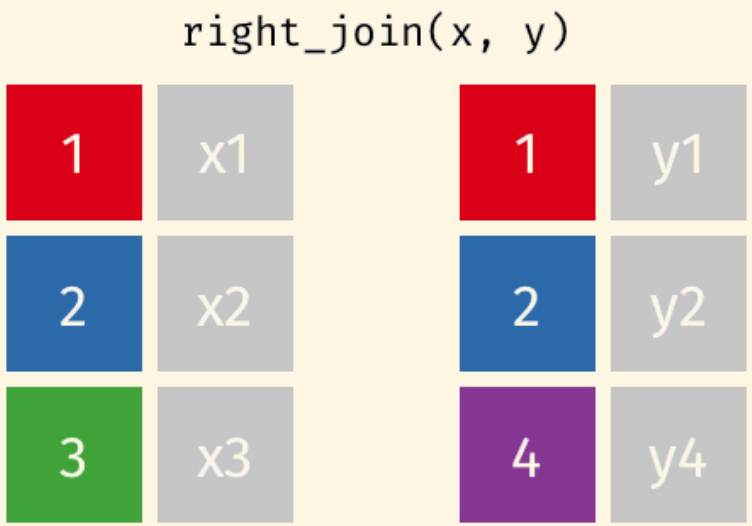
```
# A tibble: 3 x 3
     id var_x var_y
  <dbl> <chr> <chr>
1     1 x1    y1
2     2 x2    y2
3     3 x3    <NA>
```

left_join()

```
professions %>%
  left_join(dates)
```

```
# A tibble: 10 x 4
   name            profession                  birth_year death_year
   <chr>           <chr>                            <dbl>      <dbl>
 1 Ada Lovelace    Mathematician                       NA         NA
 2 Marie Curie     Physicist and Chemist               NA         NA
 3 Janaki Ammal    Botanist                          1897       1984
 4 Chien-Shiung …  Physicist                         1912       1997
 5 Katherine Joh…  Mathematician                     1918       2020
 6 Rosalind Fran…  Chemist                           1920       1958
 7 Vera Rubin      Astronomer                        1928       2016
 8 Gladys West     Mathematician                     1930         NA
 9 Flossie Wong-…  Virologist and Molecular…         1947         NA
10 Jennifer Doud…  Biochemist                        1964         NA
```

right_join()




```
right_join(tib_x, tib_y)
```

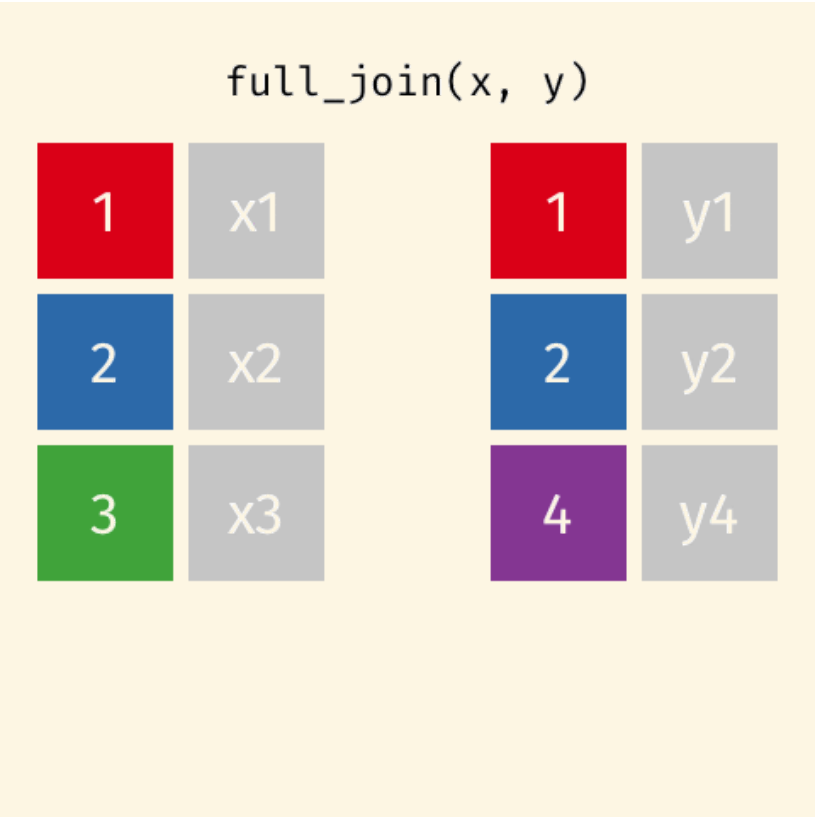
```
# A tibble: 3 x 3
     id var_x var_y
  <dbl> <chr> <chr>
1     1 x1    y1
2     2 x2    y2
3     4 <NA>  y4
```

right_join()

```
professions %>%
  right_join(dates)
```

```
# A tibble: 8 x 4
  name            profession                    birth_year death_year
  <chr>           <chr>                              <dbl>      <dbl>
1 Janaki Ammal    Botanist                            1897       1984
2 Chien-Shiung …  Physicist                           1912       1997
3 Katherine Joh…  Mathematician                       1918       2020
4 Rosalind Fran…  Chemist                             1920       1958
5 Vera Rubin      Astronomer                          1928       2016
6 Gladys West     Mathematician                       1930         NA
7 Flossie Wong-…  Virologist and Molecular …          1947         NA
8 Jennifer Doud…  Biochemist                          1964         NA
```

full_join()




```
full_join(tib_x, tib_y)
```

```
# A tibble: 4 x 3
     id var_x var_y
  <dbl> <chr> <chr>
1     1 x1    y1
2     2 x2    y2
3     3 x3    <NA>
4     4 <NA>  y4
```

full_join()

```
dates %>%
  full_join(works)
```

```
# A tibble: 10 x 4
   name        birth_year death_year known_for
   <chr>            <dbl>      <dbl> <chr>
 1 Janaki Am…        1897       1984 hybrid species, biodiversity…
 2 Chien-Shi…        1912       1997 confim and refine theory of …
 3 Katherine…        1918       2020 calculations of orbital mech…
 4 Rosalind …        1920       1958 <NA>
 5 Vera Rubin        1928       2016 existence of dark matter
 6 Gladys We…        1930         NA mathematical modeling of the…
 7 Flossie W…        1947         NA first scientist to clone HIV…
 8 Jennifer …        1964         NA one of the primary developer…
 9 Ada Lovel…          NA         NA first computer algorithm
10 Marie Cur…          NA         NA theory of radioactivity,  di…
```

Alles in einer Code Sequenz

```
professions %>%
  left_join(dates) %>%
  left_join(works)
```

```
# A tibble: 10 x 5
   name     profession    birth_year death_year known_for
   <chr>    <chr>              <dbl>      <dbl> <chr>
 1 Ada Lov… Mathematician         NA         NA first computer a…
 2 Marie C… Physicist an…         NA         NA theory of radioa…
 3 Janaki … Botanist            1897       1984 hybrid species, …
 4 Chien-S… Physicist           1912       1997 confim and refin…
 5 Katheri… Mathematician       1918       2020 calculations of …
 6 Rosalin… Chemist             1920       1958 <NA>
 7 Vera Ru… Astronomer          1928       2016 existence of dar…
 8 Gladys … Mathematician       1930         NA mathematical mod…
 9 Flossie… Virologist a…       1947         NA first scientist …
10 Jennife… Biochemist          1964         NA one of the prima…
```

Praktikum 10 - Daten zusammenfügen

2er Teams

1. **E-Mail**: Öffne deine Email und klicke auf den Link zu deinem persönlichen GitHub repo
2. **GitHub**: Klicke auf den grünen Button "Code" und kopiere den Link für das Repo in deine Zwischenablage
3. **RStudio Cloud**: Öffne deinen Arbeitsbereich für den Kurs in der RStudio Cloud
4. **RStudio Cloud / Projects**: Klicke auf "New Project from GitHub Repository"

Inputs - Drei Dataframes

einkaeufe | preise | kundenprofile

```
# A tibble: 9 x 4
  kunden_id produkt_name einkauf einheit
  <chr>     <chr>           <dbl> <chr>
1 k1        Chips             2   anzahl
2 k1        Milch             3   anzahl
3 k1        Avocado           1   anzahl
4 k2        Pfirsich          2.5 kg
5 k2        Birne             0.5 kg
6 k2        Apfel             2   kg
7 k2        Tomate            1.5 kg
8 k2        Pfirsich          1   kg
9 k2        Milch             4   anzahl
```

Gewünschter Output

vorname	nachname	summe	email
Edwin	Dumont	9.2	edwin.dumont@example.com
Leonora	Garcia	24.6	leonora.garcia@example.com

Schritt 1 - Daten zusammenfügen

```
einkaeufe_preise <- einkaeufe %>%
  left_join(preise)

einkaeufe_preise
```

```
# A tibble: 9 x 5
  kunden_id produkt_name einkauf einheit preis
  <chr>     <chr>           <dbl> <chr>   <dbl>
1 k1        Chips             2   anzahl    3.8
2 k1        Milch             3   anzahl    2.2
3 k1        Avocado           1   anzahl    3.2
4 k2        Pfirsich          2.5 kg        6.5
5 k2        Birne             0.5 kg        2.6
6 k2        Apfel             2   kg        4.1
7 k2        Tomate            1.5 kg        2.7
8 k2        Pfirsich          1   kg        6.5
9 k2        Milch             4   anzahl    2.2
```

Schritt 2 - Daten zusammefassen

```
einkaeufe_preise_sum <- einkaeufe_preise %>%
  group_by(kunden_id) %>%
  summarise(
    summe = sum(preis)
  )

einkaeufe_preise_sum
```

```
# A tibble: 2 x 2
  kunden_id summe
  <chr>     <dbl>
1 k1          9.2
2 k2         24.6
```

Schritt 3 - Daten zusammenfügen + eingrenzen

```
kunden_tab <- einkaeufe_preise_sum %>%
  left_join(kundenprofile) %>%
  select(ends_with("name"), summe, email)

kunden_tab
```

```
# A tibble: 2 x 4
  vorname nachname summe email
  <chr>   <chr>    <dbl> <chr>
1 Edwin   Dumont      9.2 edwin.dumont@example.com
2 Leonora Garcia     24.6 leonora.garcia@example.com
```

Schritt 4 - Daten als Tabelle darstellen

```
kunden_tab %>%
  gt()
```

vorname	nachname	summe	email
Edwin	Dumont	9.2	edwin.dumont@example.com
Leonora	Garcia	24.6	leonora.garcia@example.com

Als eine Code Sequenz

```
einkaeufe %>%
  left_join(preise) %>%
  group_by(kunden_id) %>%
  summarise(
    summe = sum(preis)
  ) %>%
  left_join(kundenprofile) %>%
  select(ends_with("name"), summe, email) %>%
  gt()
```

vorname	nachname	summe	email
Edwin	Dumont	9.2	edwin.dumont@example.com
Leonora	Garcia	24.6	leonora.garcia@example.com

Tabellen darstellen

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

	A	B	C	H	M	R	S	T	U	V	W	X	Y	Z
1	**Kosten des Gesundheitswesens nach Leistungen 1)**													**T 14.5.1.3**
2	In Millionen Franken													
3														
4			1995	2000	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018
5		**Total**	**36 056**	**43 072**	**52 388**	**62 565**	**64 243**	**66 512**	**69 118**	**71 429**	**74 385**	**77 455**	**79 643**	**80 242**
6	L	Stationäre Kurativbehandlung 2)	9 742	10 786	12 584	13 373	13 583	14 176	14 791	14 947	15 386	15 758	15 718	15 548
7	L1	Stationäre somatische Akutbehandlung 2)				11 696	11 878	12 397	12 946	13 118	13 469	13 832	13 786	13 622
8	L2	Stationäre Psychiatriebehandlung 2)				1 674	1 699	1 771	1 836	1 819	1 905	1 912	1 917	1 908
9	L3	Stationäre Geburtshausbehandlung 2)				3	5	8	9	10	12	14	16	17
10	M	Ambulante Kurativbehandlung	8 336	10 243	12 699	15 808	16 109	16 924	17 688	18 681	19 541	20 436	21 108	20 753
11	M2	Ambulante somatische Akutbehandlung im Spital				4 226	4 315	4 717	4 969	5 427	5 677	6 136	6 307	6 409
12	M3	Ärztliche Behandlung, ambulant, Einzelleistungen 3)				4 509	4 317	4 273	4 343	4 405	4 638	4 711	4 690	3 871
13	M4	Ärztliche Behandlung, ambulant, Managed Care 3)				1 582	1 984	2 277	2 578	2 839	3 195	3 400	3 659	3 797
14	M6	Zahnbehandlung				4 022	4 089	4 171	4 251	4 347	4 279	4 256	4 473	4 684
15	M7	Ambulante Psychiatrie- und Psychologiebehandlung, kurativ 4)				864	854	923	987	1 121	1 169	1 263	1 301	1 391

Quelle: Bundesamt für Statistik - Kosten des Gesundheitswesens nach Leistungen

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

	A	B	C	H	M	R	S	T	U	V	W	X	Y	Z
1	**Kosten des Gesundheitswesens nach Leistungen 1)**													**T 14.5.1.3**
2	In Millionen Franken													
3														
4			1995	2000	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018
5		**Total**	**36 056**	**43 072**	**52 388**	**62 565**	**64 243**	**66 512**	**69 118**	**71 429**	**74 385**	**77 455**	**79 643**	**80 242**
6	L	Stationäre Kurativbehandlung 2)	9 742	10 786	12 584	13 373	13 583	14 176	14 791	14 947	15 386	15 758	15 718	15 548
7	L1	Stationäre somatische Akutbehandlung 2)				11 696	11 878	12 397	12 946	13 118	13 469	13 832	13 786	13 622
8	L2	Stationäre Psychiatriebehandlung 2)				1 674	1 699	1 771	1 836	1 819	1 905	1 912	1 917	1 908
9	L3	Stationäre Geburtshausbehandlung 2)				3	5	8	9	10	12	14	16	17
10	M	Ambulante Kurativbehandlung	8 336	10 243	12 699	15 808	16 109	16 924	17 688	18 681	19 541	20 436	21 108	20 753
11	M2	Ambulante somatische Akutbehandlung im Spital				4 226	4 315	4 717	4 969	5 427	5 677	6 136	6 307	6 409
12	M3	Ärztliche Behandlung, ambulant, Einzelleistungen 3)				4 509	4 317	4 273	4 343	4 405	4 638	4 711	4 690	3 871
13	M4	Ärztliche Behandlung, ambulant, Managed Care 3)				1 582	1 984	2 277	2 578	2 839	3 195	3 400	3 659	3 797
14	M6	Zahnbehandlung				4 022	4 089	4 171	4 251	4 347	4 279	4 256	4 473	4 684
15	M7	Ambulante Psychiatrie- und Psychologiebehandlung, kurativ 4)				864	854	923	987	1 121	1 169	1 263	1 301	1 391

Variable Jahr als Reihe

Quelle: Bundesamt für Statistik - Kosten des Gesundheitswesens nach Leistungen

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

	A	B	C	H	M	R	S	T	U	V	W	X	Y	Z
1	**Kosten des Gesundheitswesens nach Leistungen 1)**													**T 14.5.1.3**
2	In Millionen Franken													
3														
4						1995	2000	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018
5		**Total**				**36 056**	**43 072**	**52 388**	**62 565**	**64 243**	**66 512**	**69 118**	**71 429**	**74 385**	**77 455**	**79 643**	**80 242**
6	L	Stationäre Kurativbehandlung 2)				9 742	10 786	12 584	13 373	13 583	14 176	14 791	14 947	15 386	15 758	15 718	15 548
7	L1	Stationäre somatische Akutbehandlung 2)															
8	L2	Stationäre Psychiatriebehandlung 2)							1 674	1 699	1 771	1 836	1 819	1 905	1 912	1 917	1 908
9	L3	Stationäre Geburtshausbehandlung 2)															
10	M	Ambulante Kurativbehandlung				8 336	10 243	12 699	15 808	16 109	16 924	17 688	18 681	19 541	20 436	21 108	20 753
11	M2	Ambulante somatische Akutbehandlung im Spital															
12	M3	Ärztliche Behandlung, ambulant, Einzelleistungen 3)							4 509	4 317	4 273	4 343	4 405	4 638	4 711	4 690	3 871
13	M4	Ärztliche Behandlung, ambulant, Managed Care 3)							1 582	1 984	2 277	2 578	2 839	3 195	3 400	3 659	3 797
14	M6	Zahnbehandlung							4 022	4 089	4 171	4 251	4 347	4 279	4 256	4 473	4 684
15	M7	Ambulante Psychiatrie- und Psychologiebehandlung, kurativ 4)							864	854	923	987	1 121	1 169	1 263	1 301	1 391

Variable Jahr als Reihe

Reihen als Zusammenfassung (Summe)

Quelle: Bundesamt für Statistik - Kosten des Gesundheitswesens nach Leistungen

Relevanter Unterschied - Ziel der Daten Publikation

Daten in Tabellen darstellen

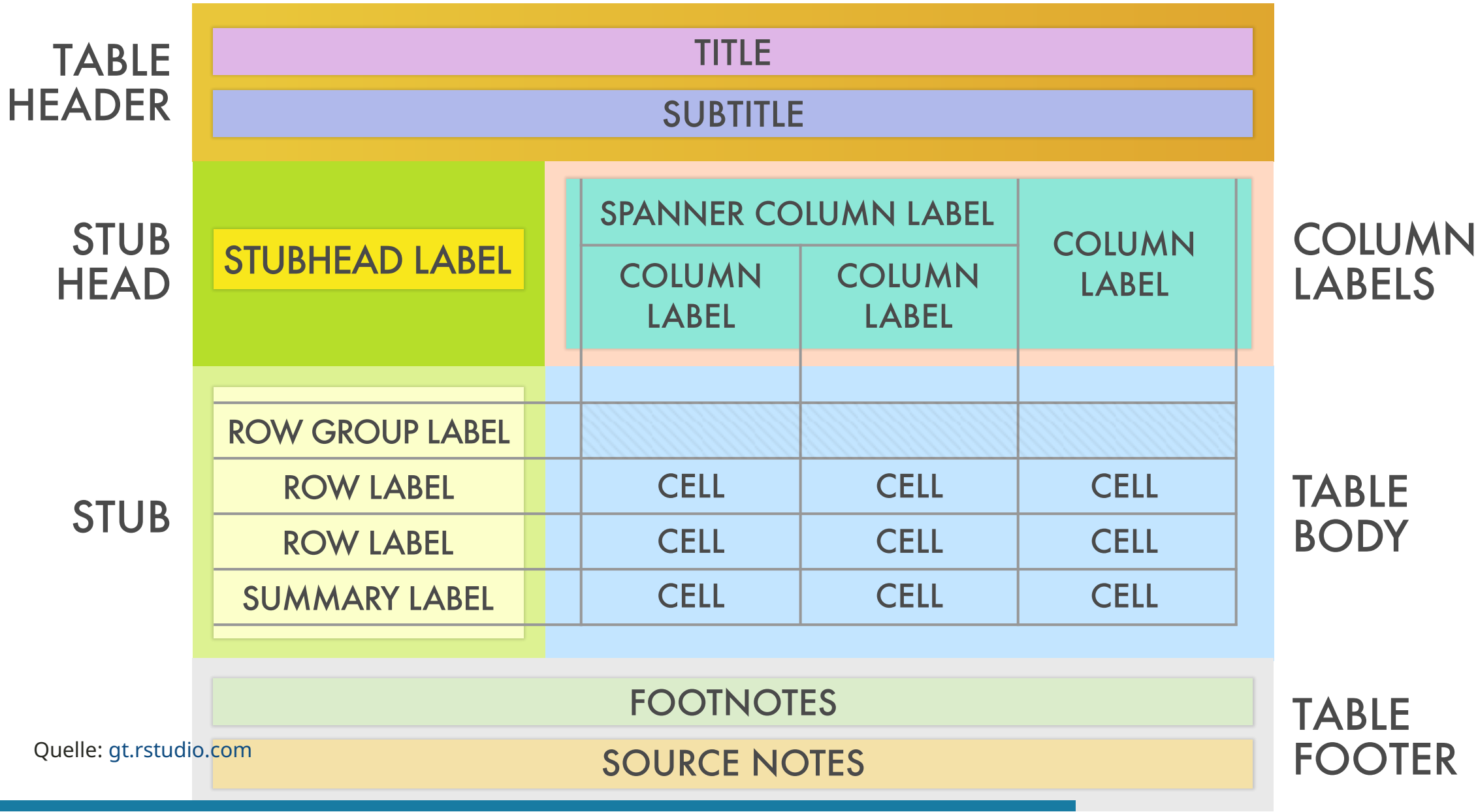
- Layout
  - Gut leserlich
  - Kompakt
  - Erkenntnis bringend
- Metadaten

Daten für weitere Nutzung bereitstellen

- Layout (Tidy data)
  - Eigenschaft 1: Jede Spalte ist eine Variable
  - Eigenschaft 2: Jede Reihe ist eine Beobachtung
  - Eigenschaft 3: Jede Zelle enthält eine Messung
- Keine Metadaten
- Keine Farben, Formatierungen, etc.
- Folgt Standards (Datum: ISO 8601)
- etc.

Tabellen darstellen mit dem {gt} Package

Parts of a *gt* Table



Quelle: gt.rstudio.com

A Typical *gt* Workflow

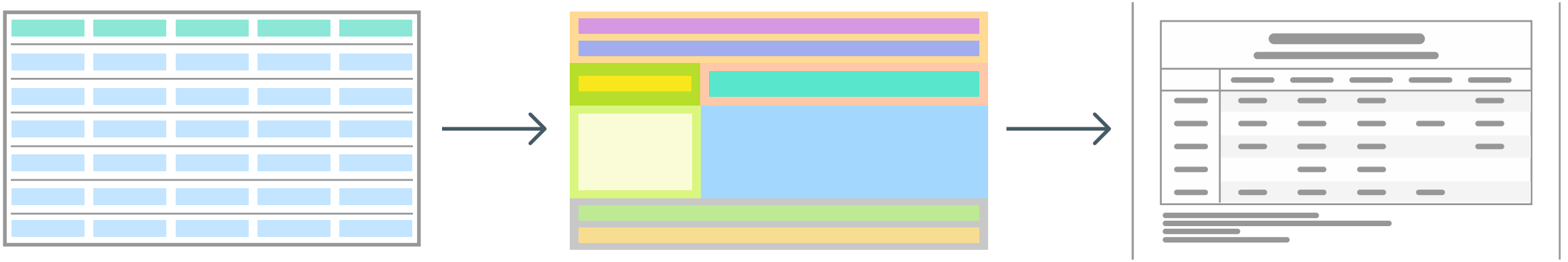


TABLE DATA

tibble or data frame

GT OBJECT

modify with gt API functions

GT TABLE

output as HTML

Quelle: gt.rstudio.com

Demonstration 3 - Tabellen darstellen

1. Schaut mir beim Programmieren zu
2. Macht euch Notizen und stellt Fragen

{gt} - Lerne das Package besser kennen

Quelle: gt.rstudio.com

Tabellen darstellen - Weitere Packages

- `{kableExtra}`: https://haozhu233.github.io/kableExtra/
- `{formattable}`: https://renkun-ken.github.io/formattable/
- `{DT}`: https://rstudio.github.io/DT/
- `{flextable}`: https://davidgohel.github.io/flextable/
- etc.

The End

The End - Noch nicht ganz

Was habt ihr gelernt?

- Anwendung von Tidyverse Packages zum
  - Importieren,
  - Aufräumen (Tidying),
  - Transformieren,
  - Visualisieren, und
  - Kommunizieren von Daten.
- Kollaboration und Versionsverwaltung mit Git/GitHub
- Datenprojekte reproduzierbar publizieren mit GitHub
- Das Konzept von Tidy Data

Wie geht's weiter?

Raus aus der RStudio Cloud

1. Installationen: https://github.com/rstatsZH/kochbuch/tree/main/01-Installation
2. Einmalig: Tidyverse Packages installieren
3. Danach: Tidyverse Packages laden
4. Packages ausserhalb des Tidyverse installieren und laden (e.g. `janitor`)

```
# Einmalig in Konsole ausführen
install.packages("tidyverse")

# In jedem Skript
library(tidyverse)
```

Weiterführende Ressourcen - Üben + Vertiefen

https://rstatszh.github.io/website/posts/2021-04-30-woche07/

Projektarbeit - Unterstützung bis Anfang Juli

Hausaufgabe 6

1. GitHub Repository erstellen und RStudio Projekt aufgleisen (Hausaufgabe 6)
2. Daten für das Projekt identifizieren

Wie es weiter geht: Bericht mit R Markdown schreiben

1. Daten importieren
2. Daten (visuell) erkunden
3. Daten ggf. transformieren und dann erneut (visuell) erkunden
4. Fragen an den Datensatz formulieren
5. Versuchen zu Antworten zu kommen und dokumentieren
6. Immer wieder, git add, commit, push

Feedback

Ziele erreicht?

Bitte ausfüllen: kutt.it/rstatszh-eval

Photo by: Virgil Cayasa

Wie es für mich weiter geht

1. **Beratung:** Projektbezogener Support, Code Review, Coaching
2. **rstatsZH Kursleitung:** Info über den Kurs verbreiten
3. **Kurse zu vertiefenden Themen:** Entwicklung von 4-Stunden Workshops

Contact: Lars@Lse.de

🌻 Danke

Für die Aufmerksamkeit!

Für die R packages {xaringan} und {xaringanthemer} mit welchen die Folien geschrieben wurden.

Eine PDF Version der Folien kann hier heruntergeladen werden: https://github.com/rstatsZH/website/raw/master/slides/e1_d07-data-join/e1_d07-data-join.pdf

---

Für Data Science in a Box und Remaster the Tidyverse, von welchen ich Materialien für diesen Kurs nutze und welche genau wie diese Folien mit Creative Commons Attribution Share Alike 4.0 International lizensiert sind.