

rstatsZH - Data Science mit R

Vektoren Teil 2 / Daten Importieren / Daten Aufräumen

Lars Schöbitz

2021-11-29

Rückblick - Woche 5

- Git / GitHub
 - push / pull
 - fork
- Funktion zum Umgang mit einem Datum
 - `lubridate::as_date()`
- Visualisierungen
 - Verbundene Streudiagramm
 - `geom_point() + geom_path()`

Hausaufgabe 5 - Rückblick

Ziele für diese Woche

Am Ende dieser Woche könnt ihr:

- Daten aus verschiedenen Formaten in R importieren
- Erkennen wann es notwendig ist Datumswerte aus Rohdaten in R selbst zu codieren
- Daten aus einem weiten Format in ein langes Format bringen
- Erkennen ob Daten als Tidy data klassifiziert werden können

Demonstration 2 - Vektoren

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

Daten importieren

Rechteckige Daten



readr

- `read_csv()` - Dateien mit Kommatrennung der Spalten
- `read_csv2()` - Dateien mit Semicolon getrennten Spalten
- `read_tsv()` - Dateien mit Tab getrennten Spalten
- `read_delim()` - Dateien mit selbst definierter Trennung

readxl

- `read_excel()` - read xls or xlsx files

Daten lesen

```
treibhaus <- read_csv("data/ugz_treibhausgasbilanz.csv")
treibhaus
```

```
# A tibble: 27 × 9
  Jahr Strom Kerosin Diesel Benzin Holz_UW_BG_SK Fernwaerme
  <dbl> <dbl>   <dbl> <dbl>   <dbl>           <dbl>       <dbl>
1 1990 0.324   0.638  0.418   1.09      0.021      0.228
2 1991 0.292   0.614  0.415   1.08      0.02        0.227
3 1992 0.263   0.637  0.418   1.08      0.02        0.229
4 1993 0.243   0.651  0.419   1.09      0.02        0.232
5 1994 0.229   0.661  0.416   1.10      0.02        0.234
6 1995 0.242   0.689  0.415   1.10      0.02        0.237
# ... with 21 more rows, and 2 more variables: Erdgas <dbl>,
#   Heizoel_EL <dbl>
```

Daten schreiben

Eine Datei schreiben

```
fussball_weltmeister <- tibble(  
  jahr = as.integer(c(2018, 2014, 2010, 2006,  
                      2019, 2015, 2011, 2007)),  
  weltmeisterschaft = c(rep("Männer", 4), rep("Frauen", 4)),  
  titeltraeger = c("Frankreich", "Deutschland", "Spanien",  
                  "Italien", "USA", "USA",  
                  "Japan", "Deutschland"))  
  
write_csv(x = fussball_weltmeister, file = "data/fussball_weltmeister.csv")
```

Die Datei wieder einleisen

```
read_csv("data/fussball_weltmeister.csv")
```

```
# A tibble: 8 × 3
  jahr weltmeisterschaft titeltraeger
  <dbl> <chr>           <chr>
1 2018 Männer            Frankreich
2 2014 Männer            Deutschland
3 2010 Männer            Spanien
4 2006 Männer            Italien
5 2019 Frauen            USA
6 2015 Frauen            USA
# ... with 2 more rows
```

Variablen Namen

Variablen Namen

```
schlechte_namen <- read_csv("data/bsp_names(schlechte_namen)
```

```
[1] "Nachname Frau" "Nachname Mann"
```

In R sind Leerzeichen in Variablen nicht erlaubt

```
schlechte_namen %>%  
  filter(Nachname Frau == "Meier")
```

```
Error: <text>:2:20: unexpected symbol  
1: schlechte_namen %>%  
2:   filter(Nachname Frau  
^
```

Möglich mit Backticks, aber mühsam

```
schlechte_namen %>%  
  filter(`Nachname Frau` == "Meier")
```

```
# A tibble: 1 × 2  
  `Nachname Frau` `Nachname Mann`  
  <chr>          <chr>  
1 Meier          Müller
```

Möglichkeit 1 - Variablennamen in readr Funktion definieren

```
read_csv("data/bsp_namen.csv",
         col_names = c("nachname_frau", "nachname_mann"),
         skip = 1)
```

```
# A tibble: 3 × 2
  nachname_frau nachname_mann
  <chr>          <chr>
1 Müller        Meier
2 Meier         Müller
3 Schmid       Schmid
```

Möglichkeit 2 - Variablennamen mit janitor Package bereinigen

- Namen werden standardmäßig im sogenannten snake_case formatiert

```
library(janitor)

namen <- read_csv("data/bsp_namen.csv")

namen %>%
  clean_names()
```

```
# A tibble: 3 × 2
  nachname_frau nachname_mann
  <chr>          <chr>
1 Müller        Meier
2 Meier         Müller
3 Schmid       Schmid
```

Variable Typen

Welcher Variablen Typ ist die Spalte **id**?

- 1. character
- 2. double
- 3. integer
- 4. logical

	A	B	C
1	id	name	alter
2	1	Not applicable	46
3	2	Ellen	14
4	NA	Tilde	zwanzig
5	4	Lorraine	39
6	5	Juan	91
7	6	Anush	63
8	.	Sandile	9999
9	8	Martha	38
10	9	Mason	43
11	10	Ege	36

```
read_csv("data/data-na.csv")
```

```
# A tibble: 10 × 4
  id      name       alter bewertung
  <chr>   <chr>     <chr>  <chr>
1 1      Not applicable 46    trifft nicht zu
2 2      Ellen        14    trifft zu
3 <NA>   Tilde        zwanzig trifft eher zu
4 4      Lorraine     39    teils-teils
5 5      Juan         91    trifft eher nicht zu
6 6      Anush        63    trifft eher nicht zu
7 .      Sandile      9999   trifft zu
8 8      Martha       38    trifft zu
9 9      Mason        43    teils-teils
10 10    Ege          36    teils-teils
```

NAs beim einlesen definieren

```
read_csv("data/data-na.csv",
         na = c("NA", ".", "9999", "Not applicable"))
```

	A	B	C
1	id	name	alter
2	1	Not applicable	46
3	2	Ellen	14
4	NA	Tilde	zwanzig
5	4	Lorraine	39
6	5	Juan	91
7	6	Anush	63
8	.	Sandile	9999
9	8	Martha	38
10	9	Mason	43
11	10	Ege	36

```
# A tibble: 10 × 4
  id name      alter   bewertung
  <dbl> <chr>    <chr>   <chr>
1     1 <NA>     46      trifft nicht
2     2 Ellen     14      trifft zu
3    NA Tilde    zwanzig trifft eher
4     4 Lorraine 39      teils-teils
5     5 Juan      91      trifft eher
6     6 Anush     63      trifft eher
7    NA Sandile  <NA>    trifft zu
8     8 Martha    38      trifft zu
9     9 Mason     43      teils-teils
10    10 Ege      36      teils-teils
```

Welcher Variablen Typ ist die Spalte `alter`?

- 1. character
- 2. double
- 3. integer
- 4. logical

```
dat <- read_csv("data/data-na.csv",
                 na = c("NA", ".", "9999", "Not applicable"))
dat
```

```
# A tibble: 10 × 4
  id name    alter bewertung
  <dbl> <chr>   <chr>   <chr>
1 1 <NA>     46      trifft nicht zu
2 2 Ellen     14      trifft zu
3 NA Tilde    zwanzig trifft eher zu
4 4 Lorraine  39      teils-teils
5 5 Juan      91      trifft eher nicht zu
6 6 Anush     63      trifft eher nicht zu
7 NA Sandile  <NA>    trifft zu
8 8 Martha    38      trifft zu
9 9 Mason     43      teils-teils
10 10 Ege      36      teils-teils
```

Variable alter umwandeln - numerisch

```
dat <- read_csv("data/data-na.csv",
                 na = c("NA", ".", "9999", "Not applicable"))
```

```
dat <- dat %>%
  mutate(alter = case_when(
    alter == "zwanzig" ~ "20",           # Wenn "alter" gleich zwanzig dann "20"
    TRUE ~ alter)) %>%
  mutate(alter = as.numeric(alter))      # Unwandlung in den Typ numerisch
```

```
# A tibble: 10 × 4
  id name     alter bewertung
  <dbl> <chr>   <dbl> <chr>
1 1 <NA>       46 trifft nicht zu
2 2 Ellen       14 trifft zu
3 NA Tilde     20 trifft eher zu
4 4 Lorraine   39 teils-teils
5 5 Juan        91 trifft eher nicht zu
# ... with 5 more rows
```

Variable bewertung - Häufigkeitstabelle

```
dat %>%  
  count(bewertung)
```

```
# A tibble: 5 × 2  
  bewertung      n  
  <chr>        <int>  
1 teils-teils      3  
2 trifft eher nicht zu  2  
3 trifft eher zu      1  
4 trifft nicht zu     1  
5 trifft zu          3
```

	A	B	C	D
1	id	name	alter	bewertung
2	1	Not applicable	46	trifft nicht zu
3	2	Ellen	14	trifft zu
4	NA	Tilde	zwanzig	trifft eher zu
5	4	Lorraine	39	teils-teils
6	5	Juan	91	trifft eher nicht zu
7	6	Anush	63	trifft eher nicht zu
8	.	Sandile	9999	trifft zu
9	8	Martha	38	trifft zu
10	9	Mason	43	teils-teils
11	10	Ege	36	teils-teils

Variable bewertung - Visualisierung

```
ggplot(dat, aes(x = bewertung)) +  
  geom_bar() +  
  coord_flip()
```

Variable bewertung umwandeln - faktor

```
vek_bewertung_lvl <- c("trifft nicht zu", "trifft eher nicht zu",
                      "teils-teils", "trifft eher zu", "trifft zu")

dat <- dat %>%
  mutate(bewertung = fct_relevel(bewertung, vek_bewertung_lvl))
```

Variable bewertung - Häufigkeitstabelle

```
dat %>%  
  count(bewertung)
```

```
# A tibble: 5 × 2  
  bewertung      n  
  <fct>        <int>  
1 trifft nicht zu     1  
2 trifft eher nicht zu  2  
3 teils-teils       3  
4 trifft eher zu      1  
5 trifft zu          3
```

Variable bewertung - Visualisierung

```
ggplot(dat, aes(x = bewertung)) +  
  geom_bar() +  
  coord_flip()
```

Als eine Code Sequenz

```
vek_bewertung_lvl <- c("trifft nicht zu", "trifft eher nicht zu",
                      "teils-teils", "trifft eher zu", "trifft zu")

dat_clean <- read_csv("data/data-na.csv",
                      na = c("NA", ".", "9999", "Not applicable")) %>%
  mutate(alter = case_when(
    alter == "zwanzig" ~ "20",           # Wenn "alter" gleich zwanzig dann "20"
    TRUE ~ alter)) %>%                  # Sonst "alter"
  mutate(alter = as.numeric(alter)) %>%   # Unwandlung in den Typ numerisch
  mutate(bewertung = fct_relevel(bewertung, # Umwandlung in den Typ faktor
                                 vek_bewertung_lvl)) # Mit definierten Levels
```

Daten schreiben und wieder lesen

Was ist denn nun wieder mit der Variable `bewertung` passiert?

```
write_csv(dat_clean, file = "data/data-bewertung-clean.csv")
dat_clean_csv <- read_csv(file = "data/data-bewertung-clean.csv")
dat_clean_csv
```

```
# A tibble: 10 × 4
  id name      alter bewertung
  <dbl> <chr>    <dbl> <chr>
1 1 <NA>        46 trifft nicht zu
2 2 Ellen        14 trifft zu
3 NA Tilde       20 trifft eher zu
4 4 Lorraine    39 teils-teils
5 5 Juan         91 trifft eher nicht zu
6 6 Anush        63 trifft eher nicht zu
```

Funktionen: `read_rds()` und `write_rds()`

- Zwischenergebnisse als CSV zu speichern ist unzuverlässig, wenn bestimmte Variablen Typen beibehalten werden sollen
- `read_csv()` kann nicht wissen welche Level eine Faktor Variable hat
- Eine gute Alternative sind RDS-Dateien, ein R-internes Dateiformat

```
write_rds(dat_clean, file = "data/data-bewertung-clean.rds")
dat_clean_rds <- read_rds(file = "data/data-bewertung-clean.rds")
```

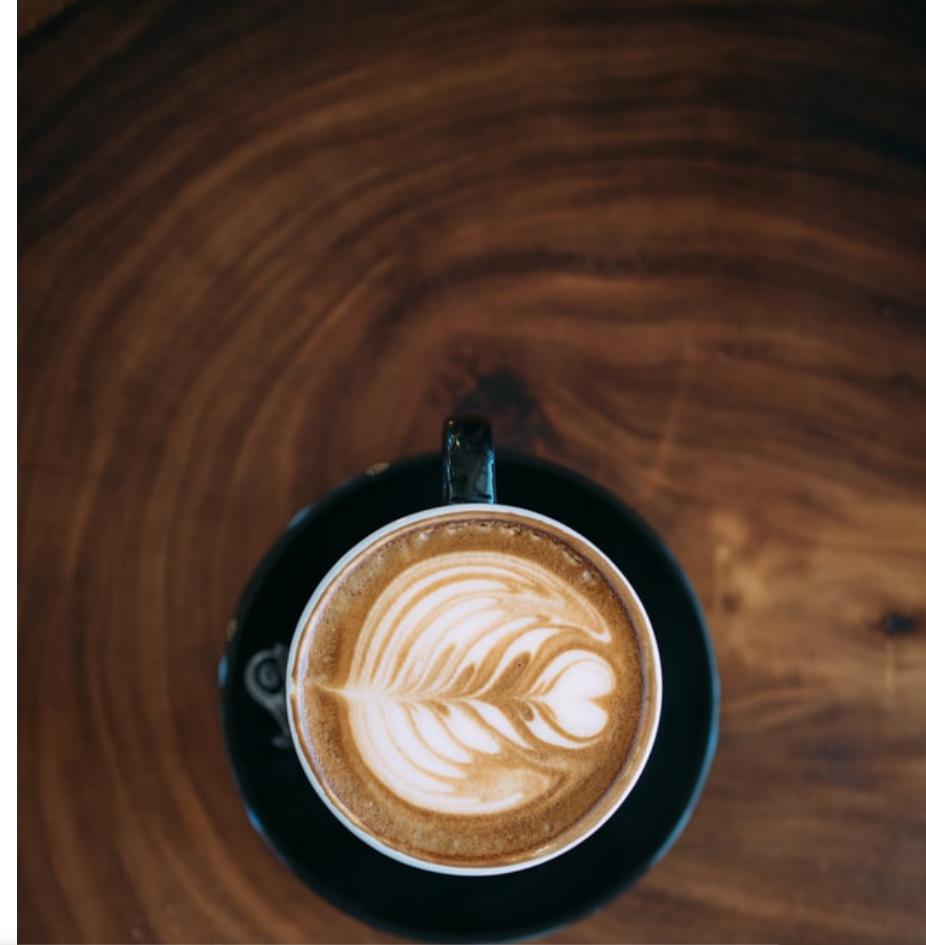
```
dat_clean_rds
```

```
# A tibble: 10 × 4
  id name      alter bewertung
  <dbl> <chr>    <dbl> <fct>
1 1 <NA>        46 trifft nicht zu
2 2 Ellen        14 trifft zu
3 NA Tilde       20 trifft eher zu
4 4 Lorraine    39 teils-teils
```

Pause

10 : 00

Photo by: [Blake Wisz](#)



Praktikum 9 - Daten importieren - Treibhausgasbilanz

2er Teams - Übung 1

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"

Tidy data

Tidy data

Eigenschaften von Tidy data:

- Eigenschaft 1: Jede Spalte ist eine Variable
- Eigenschaft 2: Jede Reihe ist eine Beobachtung
- Eigenschaft 3: Jede Zelle enthält eine Messung

Penguins

Erfüllen die Daten die Eingeschafften für Tidy data?

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

	A	B	C	D	E	F	G	H	I
1	cc-d-01.02.04.02	Ständige Wohnbevölkerung nach Altersklasse und Altersmasszahlen nach Kanton, am 31.12.2020							
2		Provisorische Jahresergebnisse							
3	Grossregionen Kantone	Total	0-19 Jahre	20-39 Jahre	40-64 Jahre	65-79 Jahre	80 Jahre und mehr	Jugendquotient ¹	Altersquotient ²
4	Schweiz	8 667 088	1 723 565	2 280 707	3 032 787	1 171 506	458 523	32.4	30.7
5	Genferseeregion	1 668 471	351 278	455 418	574 194	205 266	82 315	34.1	27.9
6	Waadt	814 075	177 447	225 069	276 942	96 173	38 444	35.3	26.8
7	Wallis	348 318	67 628	89 388	121 294	51 839	18 169	32.1	33.2
8	Genf	506 078	106 203	140 961	175 958	57 254	25 702	33.5	26.2
9	Espace Mittelland	1 894 879	373 753	481 384	659 098	275 838	104 806	32.8	33.4
10	Bern	1 042 516	197 760	260 221	362 072	160 402	62 061	31.8	35.7
11	Freiburg	325 419	71 505	88 526	112 785	39 646	12 957	35.5	26.1
12	Solothurn	277 396	52 800	69 973	98 981	40 445	15 197	31.3	32.9
13	Neuenburg	175 860	36 459	44 834	60 437	23 984	10 146	34.6	32.4
14	Jura	73 688	15 229	17 830	24 823	11 361	4 445	35.7	37.1
15	Nordwestschweiz	1 181 397	230 093	302 831	418 419	165 110	64 944	31.9	31.9

Quelle: Bundesamt für Statistik - Ständige Wohnbevölkerung nach Altersklasse und Altersmasszahlen nach Kanton, Provisorische Jahresergebnisse, 2020

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

A	B	C	D	E	F	G	H	I
cc-d-01.02.04.02	Ständige Wohnbevölkerung	Variable Altersgruppe als Reihe						
1		Provisorische Jahresergebnisse						
2	Grossregionen Kantone	Total	0-19 Jahre	20-39 Jahre	40-64 Jahre	65-79 Jahre	80 Jahre und mehr	Jugendquotient ¹
3	Schweiz	8 667 088	1 723 565	2 280 707	3 032 787	1 171 506	458 523	32.4
4	Genferseeregion	1 668 471	351 278	455 418	574 194	205 266	82 315	34.1
5	Waadt	814 075	177 447	225 069	276 942	96 173	38 444	35.3
6	Wallis	348 318	67 628	89 388	121 294	51 839	18 169	32.1
7	Genf	506 078	106 203	140 961	175 958	57 254	25 702	33.5
8	Espace Mittelland	1 894 879	373 753	481 384	659 098	275 838	104 806	32.8
9	Bern	1 042 516	197 760	260 221	362 072	160 402	62 061	31.8
10	Freiburg	325 419	71 505	88 526	112 785	39 646	12 957	35.5
11	Solothurn	277 396	52 800	69 973	98 981	40 445	15 197	31.3
12	Neuenburg	175 860	36 459	44 834	60 437	23 984	10 146	34.6
13	Jura	73 688	15 229	17 830	24 823	11 361	4 445	35.7
14	Nordwestschweiz	1 181 397	230 093	302 831	418 419	165 110	64 944	31.9
15								31.9

Quelle: Bundesamt für Statistik - Ständige Wohnbevölkerung nach Altersklasse und Altersmasszahlen nach Kanton, Provisorische Jahresergebnisse, 2020

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

A	B	C	D	E	F	G	H	I
1	cc-d-01.02.04.02	Ständige Wohnbevölkerung						
2		Provisorische Jahresergebnisse						
3	Grossregionen	Total	0-19 Jahre	20-39 Jahre	40-64 Jahre	65-79 Jahre	80 Jahre und mehr	Jugendquotient ¹
4	Kantone							Altersquotient ²
5	Schweiz	8 667 088	1 723 565	2 280 707	3 032 787	1 171 506	458 523	32.4
6	Genferseeregion	1 668 471	351 278	455 418	574 194	205 266	82 315	34.1
7	Waadt	814 075	177 447	225 069	276 942	96 173	38 444	35.3
8	Wallis	348 318	67 628	89 388	121 294	51 839	18 169	32.1
9	Genf	506 078	106 203	140 961	175 958	57 254	25 702	33.5
10	Espace Mittelland	1 894 879	373 753	481 384	659 098	275 838	104 806	32.8
11	Bern	1 042 516	197 760	260 221	362 072	160 402	62 061	31.8
12	Freiburg	325 419	71 505	88 526	112 785	39 646	12 957	35.5
13	Solothurn	277 396	52 800	69 973	98 981	40 445	15 197	31.3
14	Neuenburg	175 860	36 459	44 834	60 437	23 984	10 146	34.6
15	Jura	73 688	15 229	17 830	24 823	11 361	4 445	35.7
	Nordwestschweiz	1 181 397	230 093	302 831	418 419	165 110	64 944	31.9
								31.9
								31.9

Reihen als Zusammenfassung (Summe)

Quelle: Bundesamt für Statistik - Ständige Wohnbevölkerung nach Altersklasse und Altersmasszahlen nach Kanton, Provisorische Jahresergebnisse, 2020

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

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

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															
5	Total		1995	2000	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018	
6	L Stationäre Kurativbehandlung 2)		36 056	43 072	52 388	62 565	64 243	66 512	69 118	71 429	74 385	77 455	79 643	80 242	
7	L1 Stationäre somatische Akutbehandlung 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	
8	L2 Stationäre Psychiatriebehandlung 2)						11 096	11 870	12 597	12 946	13 116	13 409	13 052	13 766	13 022
9	L3 Stationäre Geburtshausbehandlung 2)						1 674	1 699	1 771	1 836	1 819	1 905	1 912	1 917	1 908
10	M Ambulante Kurativbehandlung						3	5	8	9	10	12	14	16	17
11	M2 Ambulante somatische Akutbehandlung im Spital														
12	M3 Ärztliche Behandlung, ambulant, Einzelleistungen 3)														
13	M4 Ärztliche Behandlung, ambulant, Managed Care 3)														
14	M6 Zahnbehandlung														
15	M7 Ambulante Psychiatrie- und Psychologiebehandlung, kurativ 4)														

Variable Jahr als Reihe

Reihen als Zusammenfassung (Summe)

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

1 Klimadaten: Eistage

4 Jahr Anzahl Eistage (maximale Temperatur < 0 °C) und langjähriger Mittelwert 1961-1990 (Ø) 1)

6	Basel-Binningen	Bern-Zollikofen 2)	Davos	Genf-Cointrin	Locarno-Monti	Lugano	Luzern	M
7	316 m ü. M.	553 m ü. M.	1594 m ü. M.	411 m ü. M.	367 m ü. M.	273 m ü. M.	454 m ü. M.	4
8	Ø 15,9 Tage	Ø 24,9 Tage	Ø 64,1 Tage	Ø 11,9 Tage	Ø 1,5 Tag	Ø 0,7 Tag	Ø 21,3 Tage	

11	1959	7	17	39	3	0	0	11
12	1960	18	23	42	11	2	2	17
13	1961	14	18	44	5	3	0	14
14	1962	22	29	86	15	2	2	25
15	1963	48	54	78	45	11	6	52
16	1964	21	38	53	18	0	0	30
17	1965	12	17	85	5	0	0	15
18	1966	15	17	78	11	1	0	17

Eistage

Frosttage

Sommertage

Hitzetage

Tropennächte

Niederschlagstage

+



160%

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

1 Klimadaten: Eistage

4 Jahr Anzahl Eistage (maximale Temperatur < 0 °C) und langjähriger Mittelwert 1961-1990 (Ø) 1)

6	Basel-Binningen	Bern-Zollikofen 2)	Davos	Genf-Cointrin	Locarno-Monti	Lugano	Luzern	M
7	316 m ü. M.	553 m ü. M.	1594 m ü. M.	411 m ü. M.	367 m ü. M.	273 m ü. M.	454 m ü. M.	4
8	Ø 15,9 Tage	Ø 24,9 Tage	Ø 64,1 Tage	Ø 11,9 Tage	Ø 1,5 Tag	Ø 0,7 Tag	Ø 21,3 Tage	

11	1959	7	17	39	3	0	0	11
12	1960	18	23	42	11	2	2	17
13	1961	14	18	44	5	3	0	14
14	1962	22	29	86	15	2	2	25
15	1963	48	54	78	45	11	6	52
16	1964	21	38	53	18	0	0	30
17	1965	12	17	85	5	0	0	15
18	1966	15	17	78	11	1	0	17

Eistage

Frosttage

Sommertage

Hitzetage

Tropennächte

Niederschlagstage

+



160%

Variable als Worksheets

Quelle: Bundesamt für Statistik - Klimadaten: Eistage, Frosttage, Sommertage, Hitzetage, Tropennächte und Niederschlagstage

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

1	Klimadaten: Eistage	Definitionen in Reihe						
4	Jahr	Anzahl Eistage (maximale Temperatur < 0 °C) und langjähriger Mittelwert 1961-1990 (Ø) 1)						
6		Basel-Binningen	Bern-Zollikofen 2)	Davos	Genf-Cointrin	Locarno-Monti	Lugano	Luzern
7		316 m ü. M.	553 m ü. M.	1594 m ü. M.	411 m ü. M.	367 m ü. M.	273 m ü. M.	454 m ü. M.
8		Ø 15,9 Tage	Ø 24,9 Tage	Ø 64,1 Tage	Ø 11,9 Tage	Ø 1,5 Tag	Ø 0,7 Tag	Ø 21,3 Tage
11	1959	7	17	39	3	0	0	11
12	1960	18	23	42	11	2	2	17
13	1961	14	18	44	5	3	0	14
14	1962	22	29	86	15	2	2	25
15	1963	48	54	78	45	11	6	52
16	1964	21	38	53	18	0	0	30
17	1965	12	17	85	5	0	0	15
18	1966	15	17	78	11	1	0	17

Quelle: Bundesamt für Statistik - Klimadaten: Eistage, Frosttage, Sommertage, Hitzetage, Tropennächte und Niederschlagstage

Variable als Worksheets

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

1	Klimadaten: Eistage	Definitionen in Reihe				Variablen als Reihen		
4	Jahr	Anzahl Eistage (maximale Temperatur < 0 °C) und langjähriger Mittelwert 1961-1990 (Ø) 1)						
6		Basel-Binningen	Bern-Zollikofen 2)	Davos	Genf-Cointrin	Locarno-Monti	Lugano	Luzern
7		316 m ü. M.	553 m ü. M.	1594 m ü. M.	411 m ü. M.	367 m ü. M.	273 m ü. M.	454 m ü. M.
8		Ø 15,9 Tage	Ø 24,9 Tage	Ø 64,1 Tage	Ø 11,9 Tage	Ø 1,5 Tag	Ø 0,7 Tag	Ø 21,3 Tage
11	1959	7	17	39	3	0	0	11
12	1960	18	23	42	11	2	2	17
13	1961	14	18	44	5	3	0	14
14	1962	22	29	86	15	2	2	25
15	1963	48	54	78	45	11	6	52
16	1964	21	38	53	18	0	0	30
17	1965	12	17	85	5	0	0	15
18	1966	15	17	78	11	1	0	17
◀ ▶ Eistage Frosttage Sommertage Hitzetage Tropennächte Niederschlagstage +								

Variable als Worksheets

Quelle: Bundesamt für Statistik - Klimadaten: Eistage, Frosttage, Sommertage, Hitzetage, Tropennächte und Niederschlagstage

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

1	Klimadaten: Eistage	Definitionen in Reihe				Variablen als Reihen		
4	Jahr	Anzahl Eistage (maximale Temperatur < 0 °C) und langjähriger Mittelwert 1961-1990 (Ø) 1)						
6	Basel-Binningen	Bern-Zollikofen 2)	Davos	Genf-Cointrin	Locarno-Monti	Lugano	Luzern	M
7	316 m ü. M.	553 m ü. M.	1594 m ü. M.	411 m ü. M.	367 m ü. M.	273 m ü. M.	454 m ü. M.	4
8	Ø 15,9 Tage	Ø 24,9 Tage	Ø 64,1 Tage	Ø 11,9 Tage	Ø 1,5 Tag	Ø 0,7 Tag	Ø 21,3 Tage	
11	1959	7	17	20	2	0	0	11
12	1960	18	29	29	2	0	0	17
13	1961	14	18	44	5	3	0	14
14	1962	22	29	86	15	2	2	25
15	1963	48	54	78	45	11	6	52
16	1964	21	38	53	18	0	0	30
17	1965	12	17	85	5	0	0	15
18	1966	15	17	78	11	1	0	17
Eistage Frosttage Sommertage Hitzetage Tropennächte Niederschlagstage +								

Quelle: Bundesamt für Statistik - Klimadaten: Eistage, Frosttage, Sommertage, Hitzetage, Tropennächte und Niederschlagstage

Variable als Worksheets

???

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

1 Klimadaten: Eistage

Definitionen in Reihe

Variablen als Reihen

4 Jahr Anzahl Eistage (maximale Temperatur < 0 °C) und langjähriger Mittelwert 1961-1990 (Ø) 1)

6 Basel-Binningen

7 316 m ü. M.

8 Ø 15,9 Tage

Bern-Zollikofen 2)

553 m ü. M.

Ø 24,9 Tage

Davos

1594 m ü. M.

Ø 64,1 Tage

Genf-Cointrin

411 m ü. M.

Ø 11,9 Tage

Locarno-Monti

367 m ü. M.

Ø 1,5 Tag

Lugano

273 m ü. M.

Ø 0,7 Tag

Luzern

454 m ü. M.

Ø 21,3 Tage

Reihe als Zusammenfassung (Durchschnitt)

11	1959	7	17	20	2	0	0	11
12	1960	18	17	20	2	0	0	17
13	1961	14	18	44	5	3	0	14
14	1962	22	29	86	15	2	2	25
15	1963	48	54	78	45	11	6	52
16	1964	21	38	53	18	0	0	30
17	1965	12	17	85	5	0	0	15
18	1966	15	17	78	11	1	0	17

Eistage

Frosttage

Sommertage

Hitzetage

Tropennächte

Niederschlagstage

+



Variable als Worksheets

Quelle: Bundesamt für Statistik - Klimadaten: Eistage, Frosttage, Sommertage, Hitzetage, Tropennächte und Niederschlagstage

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.

Praktikum 9 - Data tidying - Treibhausgasbilanz

2er Teams - Übung 2 + 3

- Öffnet nochmals das Praktikum 9

Data tidying

Treibhausgasemissionen

Welche Eigenschaften von Tidy data sind hier nicht erfüllt?

Jahr	Strom	Kerosin	Diesel	Benzin	Holz_UW_BG_SK	Fernwaerme	Erdgas	Heizoel_EL	
1990	0.324	0.638	0.418	1.087		0.021	0.228	1.015	2.445
1991	0.292	0.614	0.415	1.081		0.020	0.227	1.037	2.338
1992	0.263	0.637	0.418	1.083		0.020	0.229	1.071	2.258
1993	0.243	0.651	0.419	1.093		0.020	0.232	1.108	2.185
1994	0.229	0.661	0.416	1.099		0.020	0.234	1.143	2.109

- ✗ Eigenschaft 1: Jede Spalte ist eine Variable
- ✓ Eigenschaft 2: Jede Reihe ist eine Beobachtung
- ✓ Eigenschaft 3: Jede Zelle enthält eine Messung

Treibhausgasemissionen

Wie wären alle Eigenschaften erfüllt?

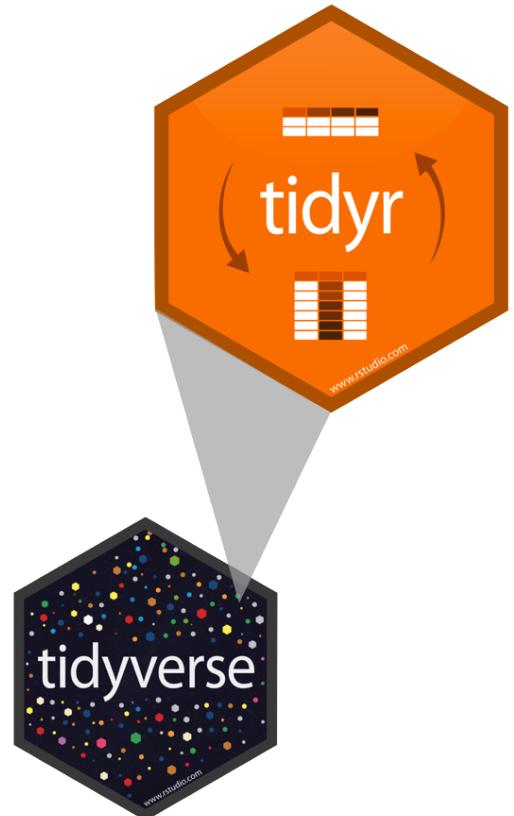
Vorher

```
# A tibble: 27 × 9
  Jahr Strom Kerosin Diesel Benzin Ho
  <dbl> <dbl>   <dbl> <dbl>   <dbl>
1 1990 0.324    0.638  0.418   1.09
2 1991 0.292    0.614  0.415   1.08
3 1992 0.263    0.637  0.418   1.08
4 1993 0.243    0.651  0.419   1.09
5 1994 0.229    0.661  0.416   1.10
6 1995 0.242    0.689  0.415   1.10
# ... with 21 more rows, and 2 more vari
#   Heizoel_EL <dbl>
```

Nachher

```
# A tibble: 216 × 3
  Jahr Energietraeger Emissionen
  <dbl> <chr>           <dbl>
1 1990 Strom            0.324
2 1990 Kerosin          0.638
3 1990 Diesel            0.418
4 1990 Benzin            1.09
5 1990 Holz_UW_BG_SK    0.021
6 1990 Fernwaerme        0.228
# ... with 210 more rows
```

R Package `tidyverse` - Grammatik zum Daten aufräumen



Das Ziel des `tidyverse` Package ist das Daten aufzuräumen mittels:

- drehen (pivoting) von Daten um das Datenformat zwischen lang und weit zu wechseln
- teilen und kombinieren von Spalten
- klarstellen wie mit `NAs` umgegangen werden soll

Pivoting

- `pivot_longer()` - Daten in ein langes Format bringen
- `pivot_wider()` - Daten in ein weites Format bringen

pivot_longer()

- **cols**: Spalten die in das lange Format konvertiert werden sollen
- **names_to**: Name der neuen Spalte in welcher die gedrehten Variablen auftauchen sollen
- **values_to**: Name der neuen Spalte in welcher die Werte der gedrehten Variablen auftauchen sollen

```
ghg_tidy <- ghg %>%
  pivot_longer(
    cols = Strom:Heizoel_EL,      # Variablen von Strom bis Heizoel_EL
    names_to = "Energietraeger",   # Variablen Namen -> Neue Spalte Energietraege
    values_to = "Emissionen"       # Variablen Werte -> Neue Spalte Emissionen
  ) %>%
  mutate(Jahr = as_factor(Jahr)) # Die Variable Jahr als Faktor definiert
```

Warum Tidy data? Warum pivoting?

Code Plot

```
ggplot(data = ghg_tidy,  
       mapping = aes(x = Jahr,  
                      y = Emissionen,  
                      fill = Energietraeger)) +  
  geom_col() +  
  
  # Plot Styling ab hier  
  scale_fill_brewer(type = "qual", palette = 1) +  
  scale_y_continuous(breaks = seq(0, 7, 1), expand = c(0, 0), limits = c(0, 7))  
  labs(title = "Treibhausgasbilanz 1990 bis 2016",  
       y = "Treibhausgasemissionen [t CO2eq/Person]",  
       x = NULL,  
       caption = "Daten: https://data.stadt-zuerich.ch/dataset/ugz\_treibhausgasb",  
       fill = "Energieträger") +  
  theme_minimal(base_size = 14) +  
  theme(panel.grid.major.x = element_blank(),
```

Hausaufgabe

Feedback

Ziele erreicht?

Bitte ausfüllen: kutt.it/rstatszh-eval





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_d06-data-import-tidy/e1_d06-data-import-tidy.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 lizenziert sind.