0_3_data_frames

March 17, 2023

1 Datenstrukturen

1.1 Data Frames

Data Frames stellen eine bestimmte Form einer zweidimensionalen Datenstrukturen dar.

Sie ähneln am ehesten einem SAS Data Set.

Die Senkrechten sind die Variablen, die Waagerechten entsprechen den Beobachtungen.

Sie bestehen aus einer bis vielen Beobachtungen verschiedener Datentypen, die alle ins sich von der Struktur gleich sind.

1.1.1 Erzeugung von Data Frames aus Vektoren

[1]:
```
# Data Frame aus drei Vektoren
v1 <- c(1, 2, 3, 4)
v2 <- c("rot", "gelb", "rot", "grün")
v3 <- c(T, F, F, T)
df <- data.frame("Id" = v1, "Farbe" = v2, "Abgeschlossen" = v3)
df
```

A data.frame: 4 × 3

Id <dbl>	Farbe <chr>	Abgeschlossen <lgl>
1	rot	TRUE
2	gelb	FALSE
3	rot	FALSE
4	grün	TRUE

[2]:
```
# Data Frame aus vier Vektoren
# v0 wird 4x recycelt
v0 <- "ABC"
v1 <- c(1, 2, 3, 4)
v2 <- c("rot", "gelb", "rot", "grün")
v3 <- c(T, F, F, T)
df <- data.frame("Study" = v0, "PatId" = v1, "Farbe" = v2, "Abgeschlossen" = v3)
df
```

A data.frame: 4 × 4

Study <chr>	PatId <dbl>	Farbe <chr>	Abgeschlossen <lgl>
ABC	1	rot	TRUE
ABC	2	gelb	FALSE
ABC	3	rot	FALSE
ABC	4	grün	TRUE

[3]:
```
# Data Frame aus vier Vektoren
# v0 wird 2x recycelt
v0 <- c("ABC", "BCD")
v1 <- c(1, 2, 3, 4)
v2 <- c("rot", "gelb", "rot", "grün")
v3 <- c(T, F, F, T)
df <- data.frame("Study" = v0, "PatId" = v1, "Farbe" = v2, "Abgeschlossen" = v3)
df
```

A data.frame: 4 × 4

Study <chr>	PatId <dbl>	Farbe <chr>	Abgeschlossen <lgl>
ABC	1	rot	TRUE
BCD	2	gelb	FALSE
ABC	3	rot	FALSE
BCD	4	grün	TRUE

[4]:
```
# Data Frame aus vier Vektoren
# v0 ist von der Länge kein Vielfaches der anderen Vektoren, daher gibt es eine␣
 ↪Fehlermeldung beim Zusammenführen.
v0 <- c("ABC", "BCD", "CDE")
v1 <- c(1, 2, 3, 4)
v2 <- c("rot", "gelb", "rot", "grün")
v3 <- c(T, F, F, T)
#df <- data.frame("Study" = v0, "PatId" = v1, "Farbe" = v2, "Abgeschlossen" =␣
 ↪v3)
#df
# Die vorigen beiden Zeilen sind nicht ausgeführt worden.
# Die Fehlermeldung in der nächsten Zelle wurde händisch kopiert.
```

Error in data.frame(Study = v0, PatId = v1, Farbe = v2, Abgeschlossen = v3): Argumente implizieren unterschiedliche Anzahl Zeilen: 3, 4 Traceback:

1. data.frame(Study = v0, PatId = v1, Farbe = v2, Abgeschlossen = v3) 2. stop(gettextf("arguments imply differing number of rows: . paste(unique(nrows), collapse = ", ")), domain = NA)

1.1.2 Erzeugung von Data Frames aus Rohdaten

Diese Möglichkeit entspricht in R am ehesten den Datalines für einfache Datensätze zur Demonstration.

Die Datenzeilen können aus einer CSV-Datei herauskopiert werden und in das R-Programm eingefügt werden.

[5]:
```
# Datenzeilen als CSV-String laden
CSV_text_string = "STUDYID, USUBJID, AGE, GENDER, RFICDTC
'XYZ', 'S001', 33, 'M', '2023-01-02'
'XYZ', 'S002', 43, 'F', '2023-01-03'
'XYZ', 'S003', 24, 'F', '2023-01-03'
'XYZ', 'S004', 19, 'M', '2023-01-04'
"

df = read.csv(text = CSV_text_string, header = TRUE, quote = "\"'")
df
```

A data.frame: 4 × 5

STUDYID <chr>	USUBJID <chr>	AGE <int>	GENDER <chr>	RFICDTC <chr>
XYZ	S001	33	M	2023-01-02
XYZ	S002	43	F	2023-01-03
XYZ	S003	24	F	2023-01-03
XYZ	S004	19	M	2023-01-04

Zwischenbemerkung:

Es gibt auch Datumswerte. Diese kann man aus Strings erzeugen und in Variablen abspeichern.

Wenn man einfacher mit Datumswerten arbeiten will, sollte man sich das Paket **lubridate** anschauen. Das erleichtert vieles im Zusammenhang mit der Verarbeitung von Datumswerten.

[6]:
```
# Datumsvariable generieren
# R: Referenzdatum 01-Jan-1970
# SAS: Referenzdatum 01-Jan-1960
d <- as.Date('2023-01-02')
d
```

2023-01-02

[7]:
```
typeof(d)
```

'double'

[8]:
```
str(d)
```

```
 Date[1:1], format: "2023-01-02"
```

[9]:
```
as.integer(d)
```

19359

[10]:
```
# Datumsdifferenzen ermitteln, das Ergebnis ist ein Objekt Datumsdifferenz.
d1 <- as.Date('2023-01-02')
d2 <- as.Date('28-02-2023', format = "%d-%m-%Y")
dd <- d2 - d1
dd
```

```
Time difference of 57 days
```

[11]:
```
# Dieses Objekt wird vor der Weiterverarbeitung in einen Integerwert für␣
↪Datumsdifferenzen umgewandelt.
as.integer(dd)
```

57

1.1.3 Erzeugung von Data Frames aus CSV-Dateien

Die Syntax des Befehls entspricht derjenigen, die bereits oben benutzt worden ist.

Der Parameter **text** wird durch den Parameter **file** ersetzt.

Weitere Informationen zu den Parametern findet man in der Hilfe.

Dazu gibt man auf der Kommandozeile *?read.csv()* ein.

Die hier benutzte Datei kann unter **Safety Data** https://github.com/SafetyGraphics/safetyData (Letzter Zugriff 10.03.2023) heruntergeladen werden. Sie ist in den Kursunterlagen auf GitHub bereitgestellt.

Sie ist ebenfalls Teil des R-Pakets **safetyData**.

[12]:
```
# CSV-Datei einlesen.
df = read.csv(file = "data/ae.csv", sep = ",", header = TRUE, quote = "\"'")
```

[13]:
```
# Einfache Zusammenfassung ausgeben.
summary(df)
```

```
   STUDYID              DOMAIN             USUBJID              AESEQ
 Length:1191        Length:1191        Length:1191        Min.   : 1.00
 Class :character   Class :character   Class :character   1st Qu.: 2.00
 Mode  :character   Mode  :character   Mode  :character   Median : 4.00
                                                          Mean   : 4.53
                                                          3rd Qu.: 6.00
                                                          Max.   :23.00

    AESPID             AETERM              AELLT             AELLTCD
 Length:1191        Length:1191        Length:1191        Mode:logical
 Class :character   Class :character   Class :character   NA's:1191
 Mode  :character   Mode  :character   Mode  :character


   AEDECOD            AEPTCD             AEHLT             AEHLTCD
 Length:1191        Mode:logical       Length:1191        Mode:logical
 Class :character   NA's:1191          Class :character   NA's:1191
 Mode  :character                      Mode  :character
```

```
   AEHLGT          AEHLGTCD         AEBODSYS         AEBDSYCD
Length:1191        Mode:logical   Length:1191        Mode:logical
Class :character   NA's:1191      Class :character   NA's:1191
Mode  :character                  Mode  :character

   AESOC           AESOCCD          AESEV              AESER
Length:1191        Mode:logical   Length:1191        Length:1191
Class :character   NA's:1191      Class :character   Class :character
Mode  :character                  Mode  :character   Mode  :character

 AEACN            AEREL             AEOUT              AESCAN
Mode:logical   Length:1191        Length:1191        Length:1191
NA's:1191      Class :character   Class :character   Class :character
               Mode  :character   Mode  :character   Mode  :character

  AESCONG          AESDISAB          AESDTH           AESHOSP
Length:1191        Length:1191      Length:1191        Length:1191
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character

  AESLIFE            AESOD             AEDTC            AESTDTC
Length:1191        Length:1191      Length:1191        Length:1191
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character

  AEENDTC             AESTDY           AEENDY
Length:1191        Min.   :-277.00   Min.   : -2.00
Class :character   1st Qu.:  15.00   1st Qu.: 27.00
Mode  :character   Median :  32.00   Median : 53.00
                   Mean   :  45.83   Mean   : 67.14
                   3rd Qu.:  63.00   3rd Qu.:101.25
                   Max.   : 366.00   Max.   :211.00
```

```
                   NA's   :26        NA's   :473
```

[14]:
```
# Die ersten sechs Zeilen darstellen.
head(df)
```

A data.frame: 6 × 35

	STUDYID <chr>	DOMAIN <chr>	USUBJID <chr>	AESEQ <int>	AESPID <chr>	AETERM <chr>
1	CDISCPILOT01	AE	01-701-1015	1	E07	APPLICATION SI
2	CDISCPILOT01	AE	01-701-1015	2	E08	APPLICATION SI
3	CDISCPILOT01	AE	01-701-1015	3	E06	DIARRHOEA
4	CDISCPILOT01	AE	01-701-1023	3	E10	ATRIOVENTRICU
5	CDISCPILOT01	AE	01-701-1023	1	E08	ERYTHEMA
6	CDISCPILOT01	AE	01-701-1023	2	E09	ERYTHEMA

[15]:
```
# Die ersten 6 Zeilen der Häufigkeitstabelle darstellen.
head(table(df$AEBODSYS))
```

```
                                CARDIAC DISORDERS
                                               91
CONGENITAL, FAMILIAL AND GENETIC DISORDERS
                                                3
                     EAR AND LABYRINTH DISORDERS
                                                6
                                  EYE DISORDERS
                                               12
                       GASTROINTESTINAL DISORDERS
                                               87
GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS
                                              292
```

[16]:
```
# Zeige die ersten 10 Einträge
head(table(df$AEDECOD), 10)
```

```
           ABDOMINAL DISCOMFORT                   ABDOMINAL PAIN
                              1                                6
           ACROCHORDON EXCISION                ACTINIC KERATOSIS
                              1                                1
                      AGITATION                     ALCOHOL USE
                              5                                1
ALLERGIC GRANULOMATOUS ANGIITIS                        ALOPECIA
                              1                                1
                       AMNESIA                         ANXIETY
                              2                                6
```

[17]:
```
# Zeige die letzten 10 Einträge
tail(table(df$AEDECOD), 10)
```

```
       VENTRICULAR SEPTAL DEFECT                          VERTIGO
                               3                                2
                 VIRAL INFECTION                     VISION BLURRED
                               1                                3
                        VOMITING                  WEIGHT DECREASED
                              16                                2
WHITE BLOOD CELL COUNT INCREASED   WOLFF-PARKINSON-WHITE SYNDROME
                               2                                2
                           WOUND                WOUND HAEMORRHAGE
                               2                                1
```

Mit diesem Datensatz werden wir jetzt weiterarbeiten und uns die Möglichkeiten von **tidyverse** anschauen.

[ ]: