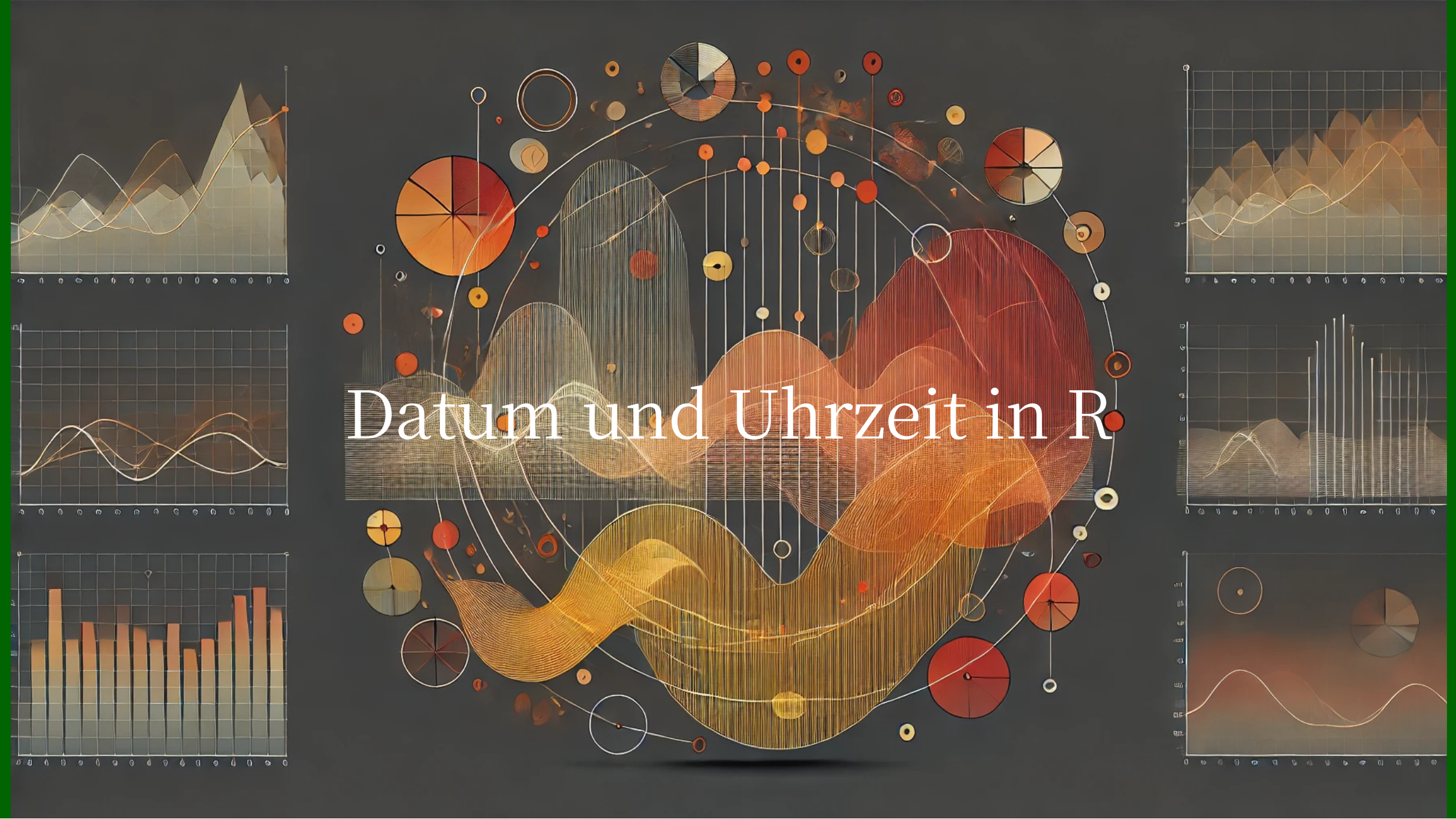
Datum und Uhrzeit in R

Datum und Uhrzeit in R

Damit R mit Datum und Uhrzeit richtig umgehen kann werden Strings wie `15. Januar 2009` intern in eine Zahl umgewandelt.

- Hierbei wird ein sogenannter Referenztag angelegt (*epoch*), von welchem aus alle Tage berechnet werden
- In R ist der Referenztag der `1. Januar 1970`
- Intern wird dadurch der `15. Januar 2009` als 14259 abgespeichert
- Wie bei Faktorvariablen erscheint ein Datum als String-Variable
  - Jedoch: Bei der Ausgabe des Datentyps erkennen wir, dass es sich um ein Datum handelt

```
library(lubridate)
x <- dmy("15. Januar 2009")
class(x) # welcher Objekttyp ist hier enthalten
```

```
[1] "Date"
```

```
as.numeric(x) # Von R intern verwendet
```

```
[1] 14259
```

Datum und Uhrzeit in R

- `ggplot2` beachtet den Datentyp und kann Datumsformate richtig wiedergeben
- Mit dem Paket `lubridate` können Sie in R effizient mit dem Datumsformat arbeiten
- Sie können mit lubridate den Tag, Monat und Jahr einzeln aus einem Datum extrahieren

```
dates <- c("2010-03-05", "2012-03-09", "2006-01-04",
       "2002-12-12")
dates2 <- ymd(dates) # Umwandeln in Datum
data.frame(Datum = dates2,
           Monat = month(dates2),
           Tag = day(dates2),
           Jahr = year(dates2))
```

```
       Datum Monat Tag Jahr
1 2010-03-05     3   5 2010
2 2012-03-09     3   9 2012
3 2006-01-04     1   4 2006
4 2002-12-12    12  12 2002
```

Datum und Uhrzeit in R

Eine weitere wichtige Funktion um Strings ins Datumsformat zu konvertieren sind die *Parser*

- Hier ein etwas ausführlicheres Beispiel zu den Möglichkeiten eines *Parsers* in `lubridate`:

```
x <- c(20030101, "Geschrieben am 2007 2 9", "2005/01/02",
       20090103, "201004 ... 07", "2007-1-14",
       "2003-1, 5")
ymd(x)
```

```
[1] "2003-01-01" "2007-02-09" "2005-01-02" "2009-01-03" "2010-04-07"
[6] "2007-01-14" "2003-01-05"
```

Datum und Uhrzeit in R

Problematisch: Wenn das Datum *nicht* im Format "Jahr, Monat, Tag" vorhanden ist, sondern bspw. in "Monat, Tag, Jahr"

Hierfür bietet `lubridate` alle möglichen Kombinationen von `ymd` als Parser an:

- Bspw. `mdy` oder `dmy`

Datum und Uhrzeit in R

> Angenommen Sie haben heute mittag um 14:00 Uhr Ortszeit einen Flug nach New York. Die Flugzeit beträgt 9h. Zu welcher Uhrzeit landen Sie in New York (Ortszeit) und wie viel Uhr ist es dann in Frankfurt?

Datum und Uhrzeit in R

> Angenommen Sie haben heute mittag um 14:00 Uhr Ortszeit einen Flug nach New York. Die Flugzeit beträgt 9h. Zu welcher Uhrzeit landen Sie in New York (Ortszeit) und wie viel Uhr ist es dann in Frankfurt?

Sie können mit `lubridate` neben dem Tag auch die Uhrzeit auslesen indem Sie `hms` an den Datumsparser anhängen und zusätzlich alle Zeitzonen einpflegen mit den `OlsonNames()`:

Datum und Uhrzeit in R

> Angenommen Sie haben heute mittag um 14:00 Uhr Ortszeit einen Flug nach New York. Die Flugzeit beträgt 9h. Zu welcher Uhrzeit landen Sie in New York (Ortszeit) und wie viel Uhr ist es dann in Frankfurt?

Sie können mit `lubridate` neben dem Tag auch die Uhrzeit auslesen indem Sie `hms` an den Datumsparser anhängen und zusätzlich alle Zeitzonen einpflegen mit den `OlsonNames()`:

```
start <- dmy_hms("6. Mai 2019 14:00:00", tz = "Europe/Berlin")
```

Datum und Uhrzeit in R

> Angenommen Sie haben heute mittag um 14:00 Uhr Ortszeit einen Flug nach New York. Die Flugzeit beträgt 9h. Zu welcher Uhrzeit landen Sie in New York (Ortszeit) und wie viel Uhr ist es dann in Frankfurt?

Sie können mit `lubridate` neben dem Tag auch die Uhrzeit auslesen indem Sie `hms` an den Datumsparser anhängen und zusätzlich alle Zeitzonen einpflegen mit den `OlsonNames()`:

```
start <- dmy_hms("6. Mai 2019 14:00:00", tz = "Europe/Berlin")
```

```
start <- dmy_hms("6. Mai 2019 14:00:00", tz = "Europe/Berlin")
ankunft <- start + hours(9)

ankunft_ortszeit <- with_tz(ankunft, tz = "America/New_York")

ankunft_ortszeit
```

```
[1] "2019-05-06 17:00:00 EDT"
```