

17.11.20

Einführung in Data Science und maschinelles Lernen mit RStudio

Grafische Darstellung von Daten



- **Quiz**
- **Wiederholung Datenstrukturen**
- **Besprechung Aufgaben**
- **Einlesen von Daten**
- **Erstellen eines Balkendiagramms**
- **Struktur der Funktionen in ggplot**
- **Erstellen eines Balkendiagramms mit Schätzfehlern**

Quiz

DATENSTRUKTUREN

Es gibt drei Grundtypen für ein Datum

- **Boolean** (TRUE / FALSE)
- **Numeric** (1.1392)
- **String** ("Text")

und zusätzlich abgeleitete, speziellere Typen (integer, date, ...)

- **Integer** (**Untertyp von Numeric**; 12)
- **Date** (**Untertyp von Numeric**; "2019-04-11")
- **Factor** (**Untertyp von String**; "female"/"male")

SPEZIELLE LISTEN

Datentabellen

**bestehen aus Vektoren gleicher Länge
(aber potentiell unterschiedlichen Typs)**

→ Datentabellen haben keinen eindeutigen Typ

- Data Frame (base Package)**
- Tibble (tidyverse Package)**
- Data Table (data.table Package)**

SELEKTION VON SPALTEN

Am Beispiel des Data Frames zu mtcars

Selektion genau einer Spalte als *Vektor*

- `mtcars$mpg` **oder** `mtcars[["mpg"]]` (**besser nicht:** `mtcars[, "mpg"]`)
- `mtcars[[1]]` (**besser nicht:** `mtcars[, 1]`)

Selektion einer oder mehrerer Spalten als *Data Frame*

- `mtcars["mpg"]` **oder** `mtcars[1]`
- `mtcars[c("mpg", "cyl")]` **oder** `mtcars[c(1, 2)]`

SELEKTION VON ZEILEN

Am Beispiel des Data Frames zu mtcars

Selektion einer oder mehrerer Zeilen als *Data Frame*

- `mtcars[1,]`
- `mtcars[c(1,2,3),]` oder `mtcars[c(1:3, 5:20),]`

Löschen einer oder mehrerer Zeilen

- `mtcars[-1,]`
- `mtcars[-c(1,3),]`

→ Selektion als Vektor nicht möglich / Keine Selektion über den Namen

SELEKTION ANHAND VON VEKTOREN DES TYPS BOOLEAN

[1] Konstruktion des Vektors

- `mtcars$hp < 100`
- `mtcars$gear == 5`

[2] Selektion der Fälle (Zeilen) mit dem Wert TRUE

- `mtcars[mtcars$hp<100,]`
- `mtcars[mtcars$gear==5,]`

AUFGABEN

- **Speichere den Datensatz airquality in der Variable airQuality.**
- **Berechne die Gesamtdurchschnittstemperatur.**
- **Berechne die Durchschnittstemperatur für den Monat Juli.**
- **Vergleiche, ob die Monate Juli und Mai sich signifikant in ihrer Durchschnittstemperatur unterscheiden.**

ZUWEISUNG VS. VERGLEICH

Zuweisung von Datenobjekten:

`a <- x`

Zuweisung von Funktionsargumenten:

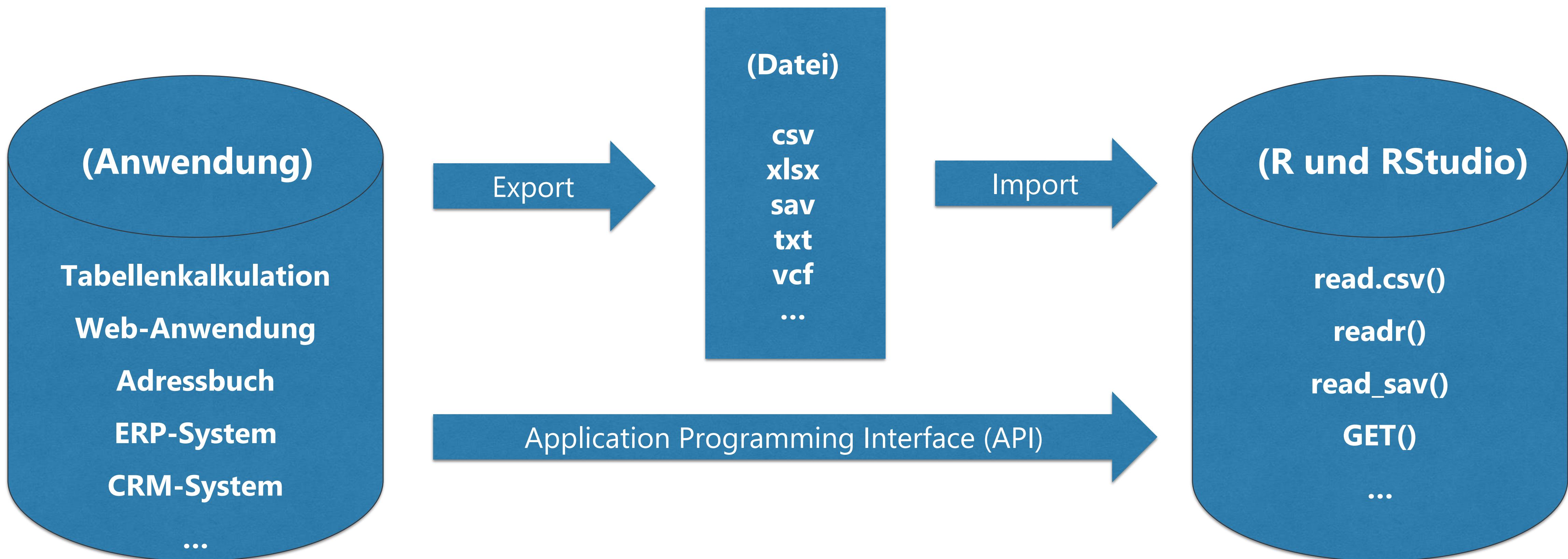
`mean(x, na.rm = TRUE)`

Vergleich von Datenobjekten oder Werten:

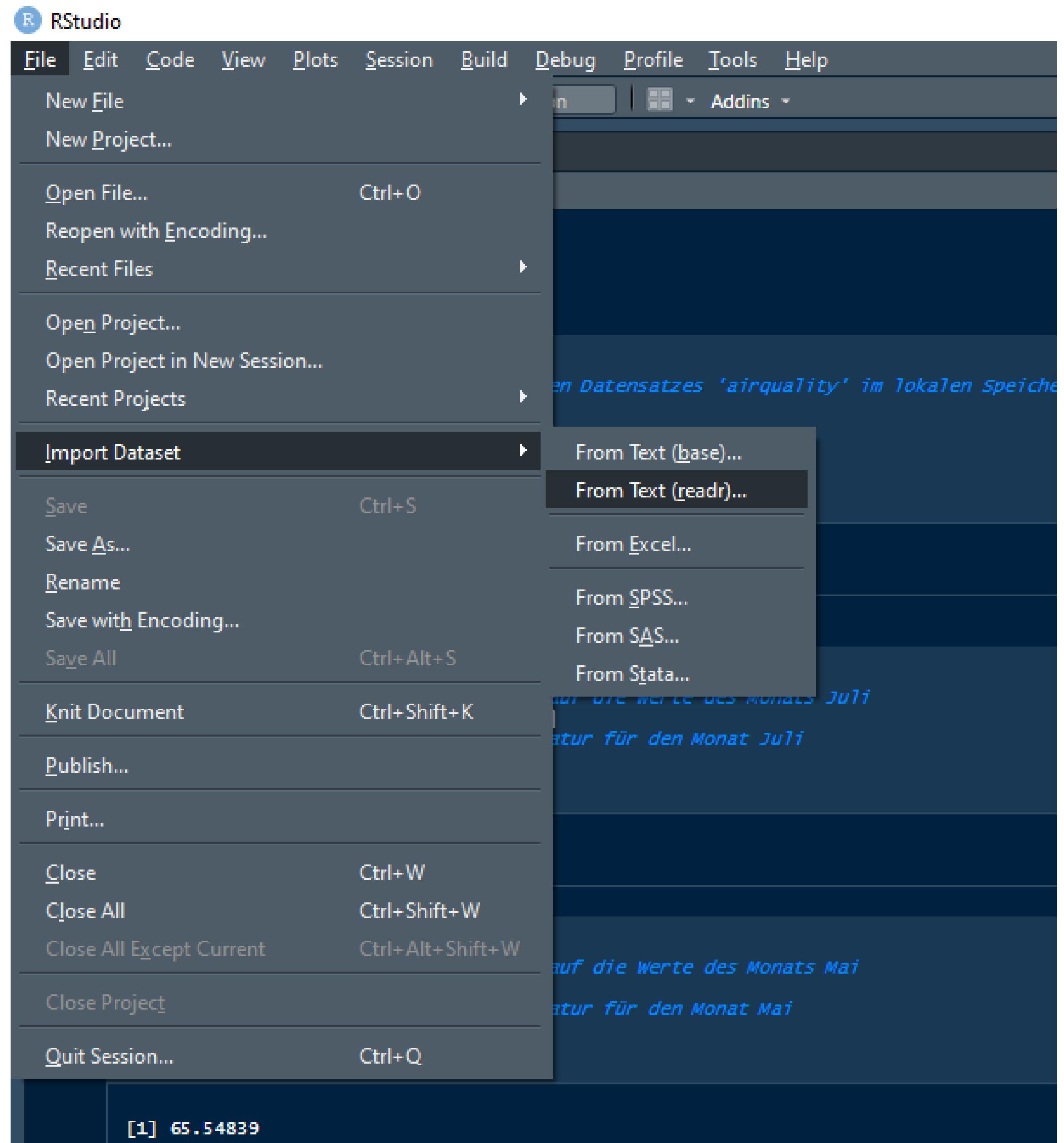
`a == x`

Das Gleichheitszeichen am besten nicht zur Zuweisung von Datenobjekten verwenden.

IMPORT VON DATEN



IMPORT MIT HILFE VON RSTUDIO (1)



IMPORT MIT HILFE VON RSTUDIO (2)

Import Text Data

File/URL:

Browse...

Data Preview:

Import Options:

Name: <input type="text" value="dataset"/>	<input checked="" type="checkbox"/> First Row as Names	Delimiter: <input type="button" value="Comma"/>	Escape: <input type="button" value="None"/>
Skip: <input type="text" value="0"/>	<input checked="" type="checkbox"/> Trim Spaces	Quotes: <input type="button" value="Default"/>	Comment: <input type="button" value="Default"/>
	<input checked="" type="checkbox"/> Open Data Viewer	Locale: <input type="button" value="Configure..."/>	NA: <input type="button" value="Default"/>

Code Preview:

```
library(readr)
dataset <- read_csv(NULL)
View(dataset)
```

① Reading rectangular data using readr

ZUSAMMENFASSUNG IMPORT

- Nutzen der Vorlage des Programmcodes aus dem RStudio Import-Aufruf

oder

- Suche im Internet:
„How to import [*Endung der Datendatei*] file into R?“

AUFGABE

- **Lege mit Hilfe von RStudio ein Projektverzeichnis auf Deinem Rechner an**
- **Lade die Dateien „kiwo.csv“, „umsatzdaten_gekuerzt.csv“ und „wetter.csv“ herunter und speichere sie in Deinem Projektverzeichnis. Die Dateien befinden sich unter:
<https://github.com/opencampus-sh/wise20-datascience>**
- **Importiere eine der Dateien in RStudio**

DIAGRAMMTYPEN

<https://www.r-graph-gallery.com/>



CHART TYPES

QUICK

TOOLS

ALL

D3.JS

PYTHON

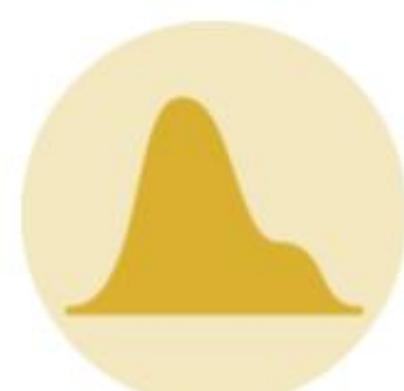
DATA TO VIZ

ABOUT

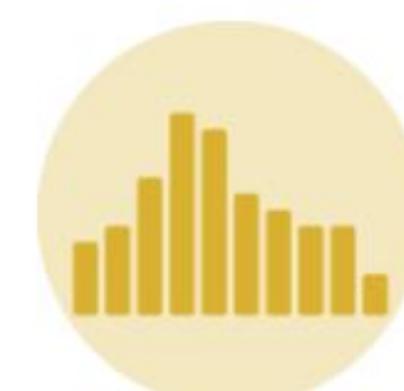
Distribution



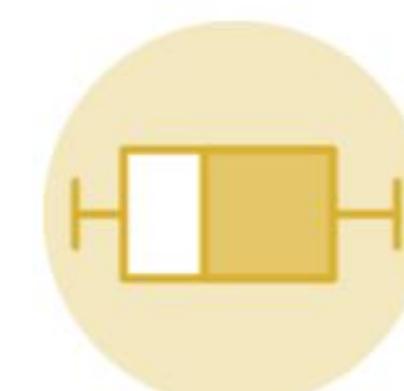
Violin



Density



Histogram



Boxplot



Ridgeline

Correlation



Scatter



Heatmap



Correlogram



Bubble



Connected scatter



Density 2d

Ranking



Barplot



Spider / Radar



Wordcloud



Parallel



Lollipop



Circular Barplot

Part of a whole

SKALENTYPEN

- **Nominalskaliert (kategorial)**
(Geschlecht, Religionszugehörigkeit)
- **Ordinalskaliert**
(Letzte Englischnote, Testantwort auf einer Skala gut-mittel-schlecht)
- **Intervallskaliert**
(Temperatur in Celsius, Intelligenzquotient)
- **Verhältnisskaliert**
(Geschwindigkeit, Einkommen)

GÄNGIGE DIAGRAMMTYPEN

- **Histogramm**
Darstellung der Verteilung einer numerischen (mind. ordinalen) Variable
- **Scatterplot**
Darstellung der Beziehung von zwei numerischen (mind. ordinalen) Variablen
- **Balkendiagramm (Barplot)**
Darstellung zwischen einer numerischen (mind. ordinalen Variable) und einer kategoriellem Variable

GGPLOT BASICS

Eine ggplot Abbildung ist ein R-Objekt, das über eine beliebige Anzahl von „Layer“ definiert wird.

Jedes Objekt wird mit ggplot() erzeugt.

Die wichtigsten Layer sind:

- Aesthetics – aes()**
Zuordnung von Daten zur Abbildung (x-Werte, y-Werte, Label, Farbwerte dargestellter Punkte, ...)
- Gemoetries – geoms()**
Definition der Darstellungsform (Histogramm, Scatterplot, ...)

Jeder Layer wird durch ein „+“ hinzugefügt.

GGPLOT LAYER

- **Facets – Layout von mehreren, nebeneinander dargestellten Abbildungen in einer Grafik**
- **Statistics – Durchführung/Darstellung einfacher statistischer Funktionen**
- **Coordinates – Definition/Layout des Raums, in dem die Daten dargestellt werden.**
- **Themes – Selektion von Templates mit unterschiedlichen (datenunabhängigen) Voreinstellungen**
- **Data – Definition eines grundlegenden Datensatzes**

BEISPIEL SCATTERPLOT

```
ggplot(mpg)+  
  geom_point(aes(x = hwy, y = cty))
```

Grundlegende Datentabelle wird für alle nachfolgenden Layer definiert.

```
ggplot(mpg)+  
  aes(x = hwy, y = cty)+  
  geom_point()
```

Aesthetics werden für alle nachfolgenden Layer definiert.

```
ggplot()+  
  aes(x = mpg$hwy, y = mpg$cty)+  
  geom_point()
```

Grundlegende Datentabelle ist nicht definiert, muss also hier angegeben werden.

WEITERE BEISPIELE VON DIAGRAMMEN

Scatterplot

```
ggplot(mpg)+  
  geom_point(aes(x = hwy, y = cty, color = displ))
```

Histogramm

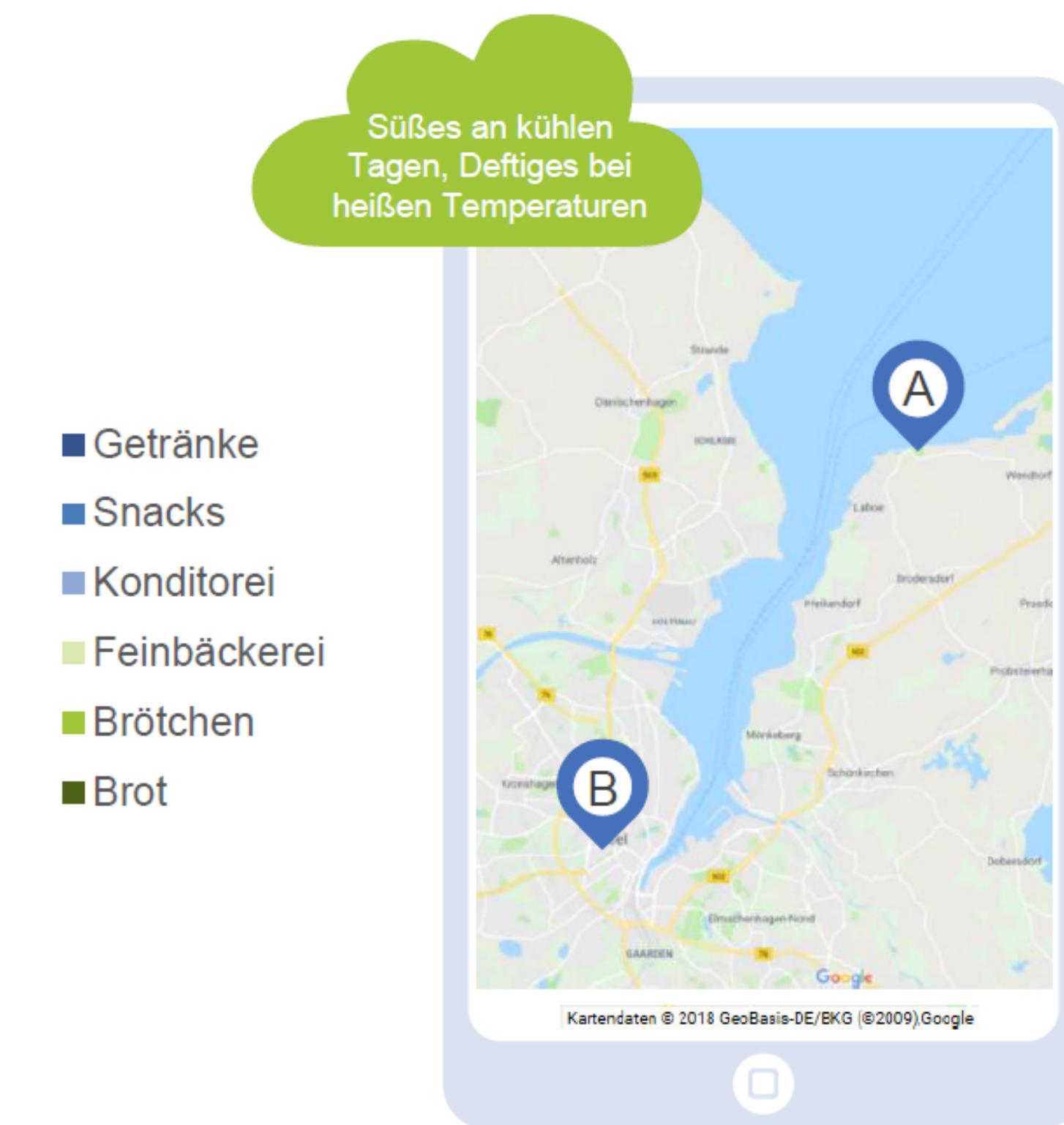
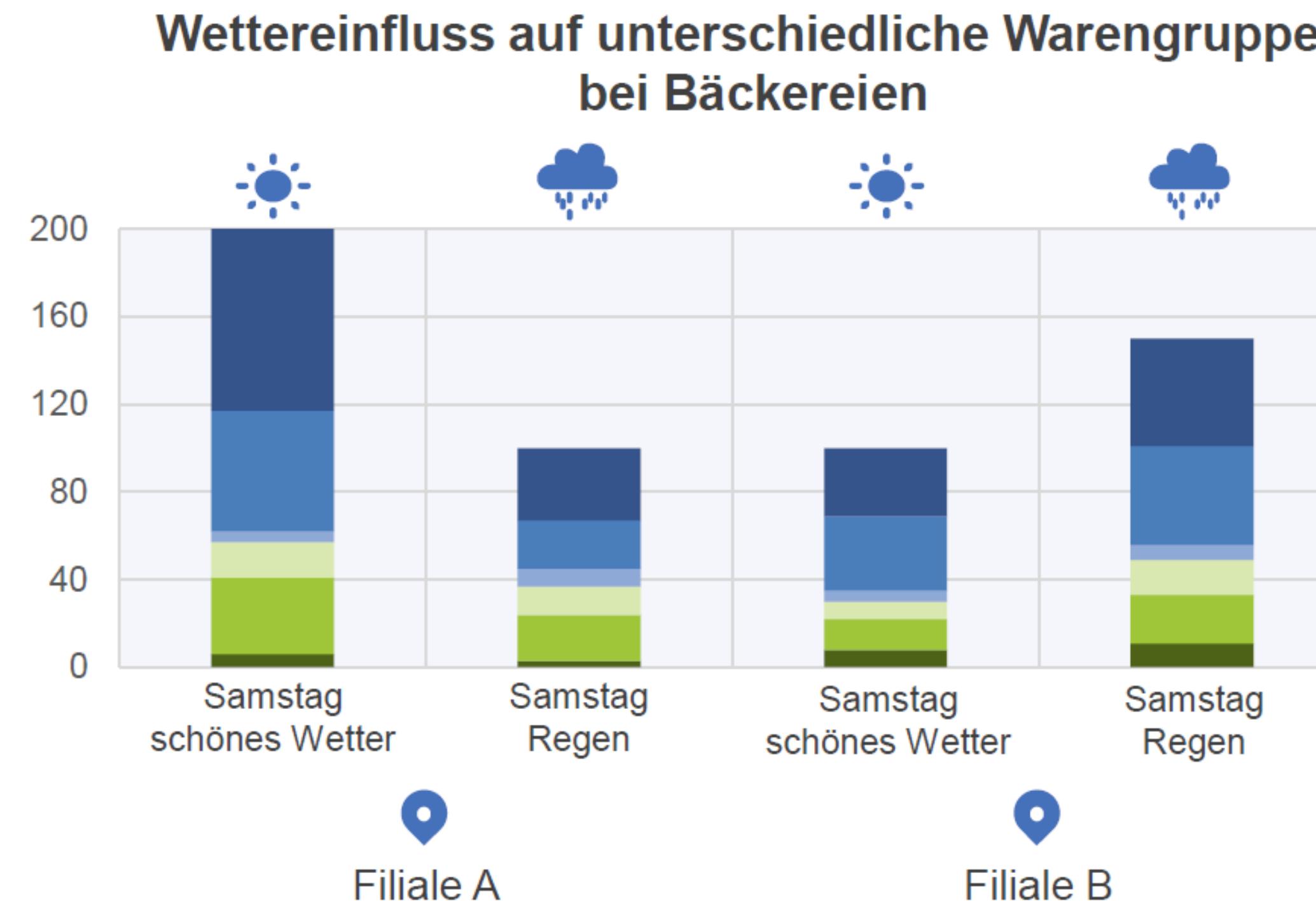
```
ggplot(mpg)+  
  geom_histogram(aes(x = cty))
```

Balkendiagramm

```
ggplot(mtcars)+  
  geom_bar(aes(x = as.factor(cyl), y = mpg), stat = "identity")
```

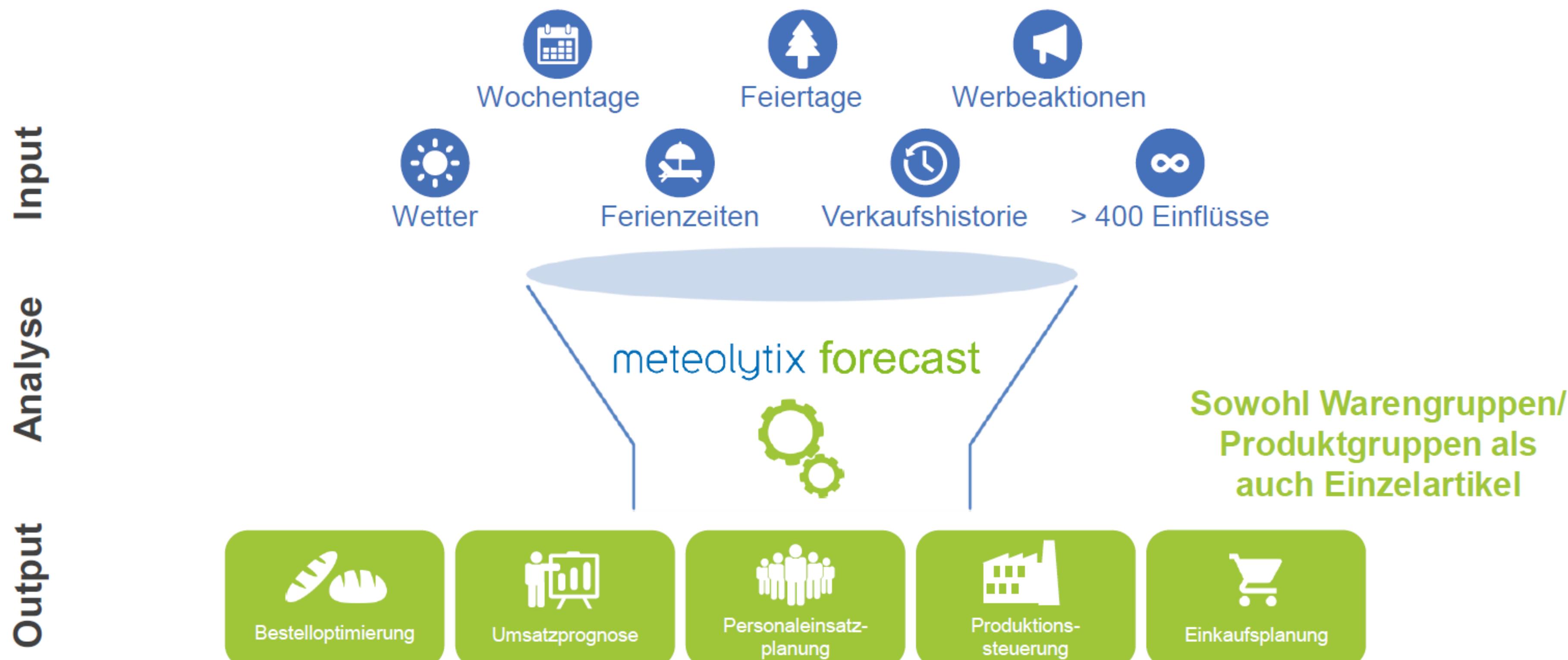
Die Stärke des Wettereffekts variiert von Ort zu Ort und wird jeweils filialindividuell berücksichtigt.

WAS WIR MACHEN



meteolytix forecast analysiert die Datenzusammenhänge von mehr als 400 Einflussfaktoren und liefert Absatzprognosen für viele Einsatzfelder.

WAS WIR MACHEN



UMSATZDATEN MIT WARENGRUPPEN

- 1 Brot**
- 2 Brötchen**
- 3 Croissant**
- 4 Konditorei**
- 5 Kuchen**
- 6 Saisonbrot**

WETTERDATEN

- mittlerer Bewölkungsgrad am Tag (0: min bis 8: max)
- mittlere Temperatur in Celsius
- mittlere Windgeschwindigkeit in m/s
- Wettercode – eine Liste mit Beschreibungen gibt es z.B. hier:
http://www.seewetter-kiel.de/seewetter/daten_symbole.htm

AUFGABE

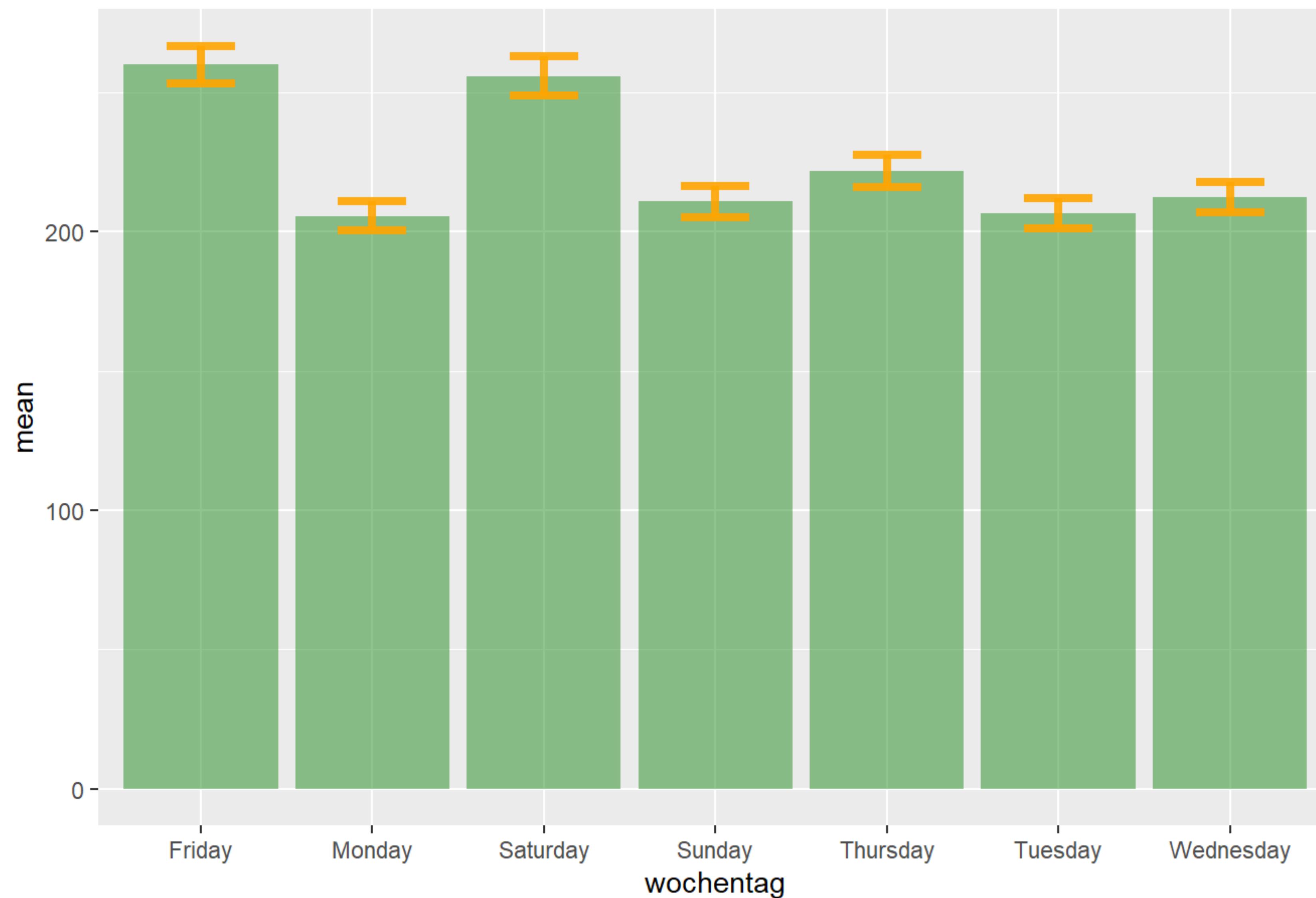
**Erstellt jeweils einmal eines der folgenden
Diagrammtypen und nutzt dazu den Datensatz
„wetter.csv“:**

- **Scatterplot**
- **Histogramm**
- **Balkendiagramm**

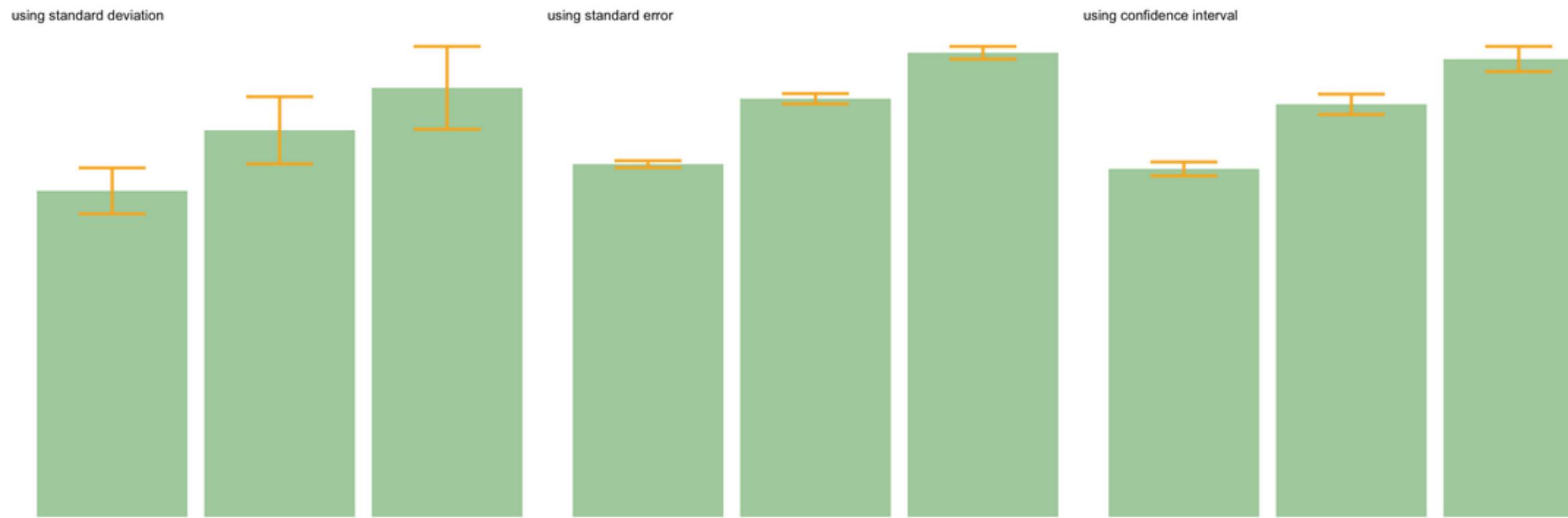
GGPLOT HILFE-SEITEN

- **Übersicht über existierende Layer und den Funktionen, die existieren:**
<https://ggplot2.tidyverse.org/reference/>
- **Gute bildliche Darstellung der Elemente einer Abbildung:**
<http://sape.inf.usi.ch/quick-reference/ggplot2>
- **Übersicht mit verschiedenen Beispielen:**
www.sthda.com/english/articles/32-r-graphics-essentials/125-ggplot-cheat-sheet-for-great-customization/
- **Cheat-Sheet von Rstudio:**
<https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>

using confidence interval



Standard deviation, Standard error or Confidence Interval?



Three different types of values are commonly used for error bars, sometimes without even specifying which one is used. It is important to understand how they are calculated, since they give very different results (see above). Let's compute them on a simple vector:

```
vec=c(1,3,5,9,38,7,2,4,9,19,19)
```

→ Standard Deviation (SD). [wiki](#)

It represents the amount of dispersion of the variable. Calculated as the root square of the variance:

```
sd <- sd(vec)
sd <- sqrt(var(vec))
```

→ Standard Error (SE). [wiki](#)

It is the standard deviation of the vector sampling distribution. Calculated as the SD divided by the square root of the sample size. By construction, SE is

<https://www.r-graph-gallery.com/4-barplot-with-error-bar.html>

AUFGABEN

- **Erstelle ein Balkendiagramm, dass über alle Warengruppen hinweg die durchschnittlichen Umsätze je Wochentag zeigt.**
- **Füge in einem zweiten Schritt zusätzlich Konfidenzintervalle der Umsätze je Wochentag hinzu („barplot with error bars“).**
- ***Freiwillige Zusatzaufgabe:***
Stelle die Umsätze je Wochentag getrennt nach Warengruppe dar (ein eigenes Balkendiagramm je Warengruppe)

STARTHILFE

Einbinden benötigter Bibliotheken

```
library(readr)  
library(lubridate)  
library(ggplot2)  
library(dplyr)
```

Einlesen der Umsatzdaten

```
umsatzdaten <- read_csv("https://raw.githubusercontent.com/opencampus-  
sh/wise20-datascience/main/umsatzdaten_gekuerzt.csv")
```

Erstellung der Variable mit dem Wochentag

```
umsatzdaten$wochentag <- weekdays(umsatzdaten$Datum)
```