Computergestützte Methoden

Datenverarbeitung

Meik Teßmer

Was ist Datenverarbeitung?

- Was ist darunter zu verstehen?
- Wozu braucht man das?

Begriffsdefinition

Wikipedia sagt dazu:

> *„Datenverarbeitung (DV) bezeichnet den organisierten Umgang mit Datenmengen mit dem Ziel, Informationen über diese Datenmengen zu gewinnen oder diese Datenmengen zu verändern.“*

Fallstudie: Fahrrad-Verleih

Welche interessanten Daten gibt es? Wie und wofür könnten wir sie verarbeiten?

Verwaltung von Daten

interessante Daten:

- Vermietungsdatum
- Station
- Menge
- Wetterdaten

Verarbeitung der Daten

interessante Verarbeitungen:

- Auslastung der Stationen
- Abhängigkeit der Auslastung vom Wetter
- Kapazitätsplanungen (je Station, saisonal)

Organisieren eines Fahrrad-Verleihs

Kontext Datenverarbeitung

1. Welche Lösungsmöglichkeiten gibt es?
2. Wie kann ein Computer dabei helfen?

Organisieren eines Fahrrad-Verleihs

Lösungsalternativen

- Papier:
  - Akten
  - Kalender
  - Listen
- digitales Pendant: Textdateien
- Tabellenkalkulation
- Datenbanken
- programmierte Lösung: spezielle Vermietungssoftware

...

Lösungsalternative *Programmierte Lösung*

- Daten liegen als Datei vor
- Beispiel: `bikeshare_data_2022.csv`

```
"date","station","count","wind_speed"
2022-01-01,"10th & E St NW",1,4.7
2022-01-01,"10th & Florida Ave NW",24,4.7
2022-01-01,"10th & G St NW",11,4.7
...
```

Lösungsalternative *Programmierte Lösung*

```
1  import csv
2
3  with open("bikeshare_data_2022.csv") as daten:
4    reader = csv.reader(daten, delimiter=",")
5
6    verleih = []
7    for row in reader:
8      # Daten, als Zeichenkette kombiniert, sammeln
9      verleih.append(row[0] + ", " + row[1] + ", " + row[2])
10
11 nach_datum_sortiert = sorted(verleih)
12
13 with open("ergebnisse.txt", "w") as ergebnisse:
14   for eintrag  in nach_datum_sortiert:
15     ergebnisse.write(eintrag + "\n")
```

Kurzbewertung dieser Lösungsalternative

Programmierte Lösung vs. bspw. Tabellenkalkulation

Programmierte Lösung

- Daten liegen als Datei vor oder in einer Datenbank
- Unterstützung einzelner Tätigkeiten werden direkt unterstützt
- nur bedingt erweiter-/veränderbar
- erfordert Programmierkenntnisse

Tabellenkalkulation

- Basis ist eine tabellenartige Datenhaltung und -verarbeitung
- sehr flexibel
- keine Vorgaben $\rightarrow$ Unterstützung muss „gebaut“ werden
- geringe Einstiegshürden

Lösungsalternative *Programmierte Lösung*

für eine “vollständige Verarbeitung” fehlt noch:

- Möglichkeit der Bearbeitung (Eingabe, Veränderung) der Daten

  $\rightarrow$ normalerweise Oberfläche (lokale Anwendung, Web-Browser)

aber das grundsätzliche Prinzip der Datenverarbeitung ist erkennbar: EVA

Lösungsalternative *Tabellenkalkulation*

- Was ist das?
- Eigenschaften?

Lösungsalternative *Tabellenkalkulation*

Interessante Aspekte:

- Interaktion
- Verarbeiten von Daten verschiedenen Typs (Zahlen, Zeit/Datum, Text)
- Tabellenform
- Visualisierung von Daten (Diagram)
- Manipulation

Aspekte

- Datenhandhabung:
  - Adressierung von (einzelnen) Daten $\rightarrow$ besonders wichtig bei mehrdimensionalen Daten
  - Verarbeitung „gemischter“ Daten (Datentypen)
  - Angabe von Bereichen
- Berechnung:
  - einfache Auswahl von Funktionen
  - Mischen mehrerer unterschiedlicher Funktionen
- Visualisierung

Datenhandhabung

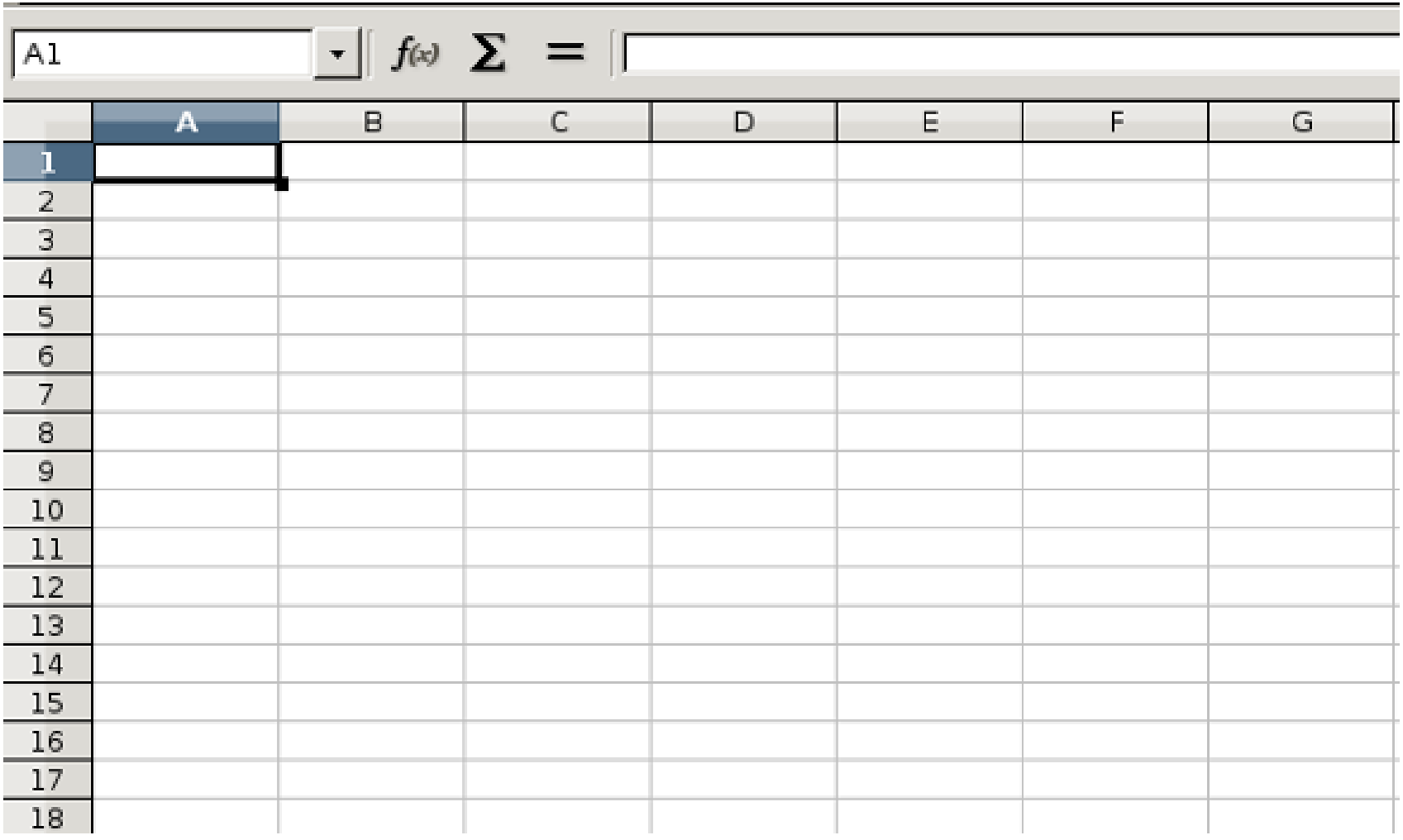
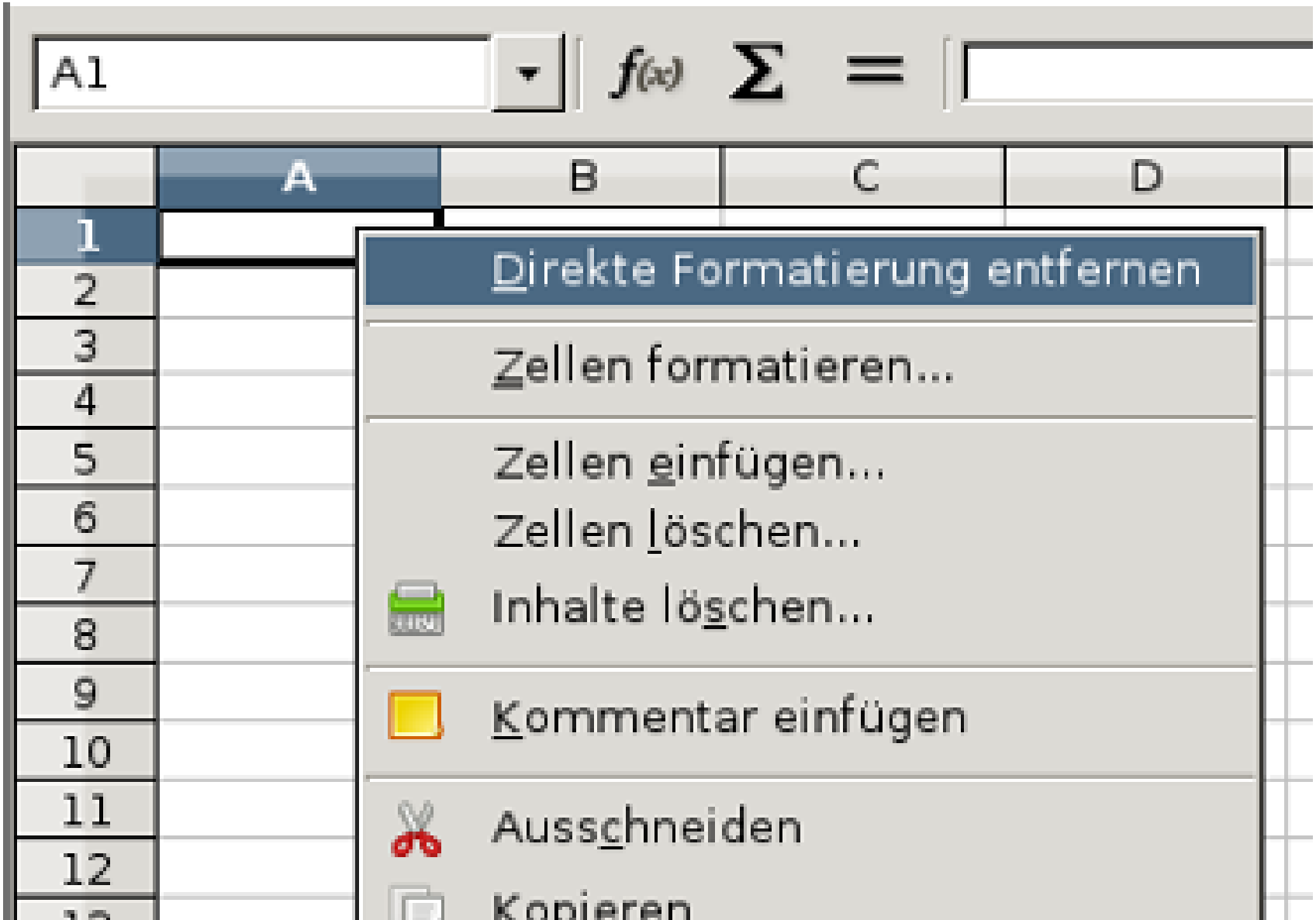


Figure 1: Tabelle

Zellen

→ Inhalt vs. Darstellung



Zellen

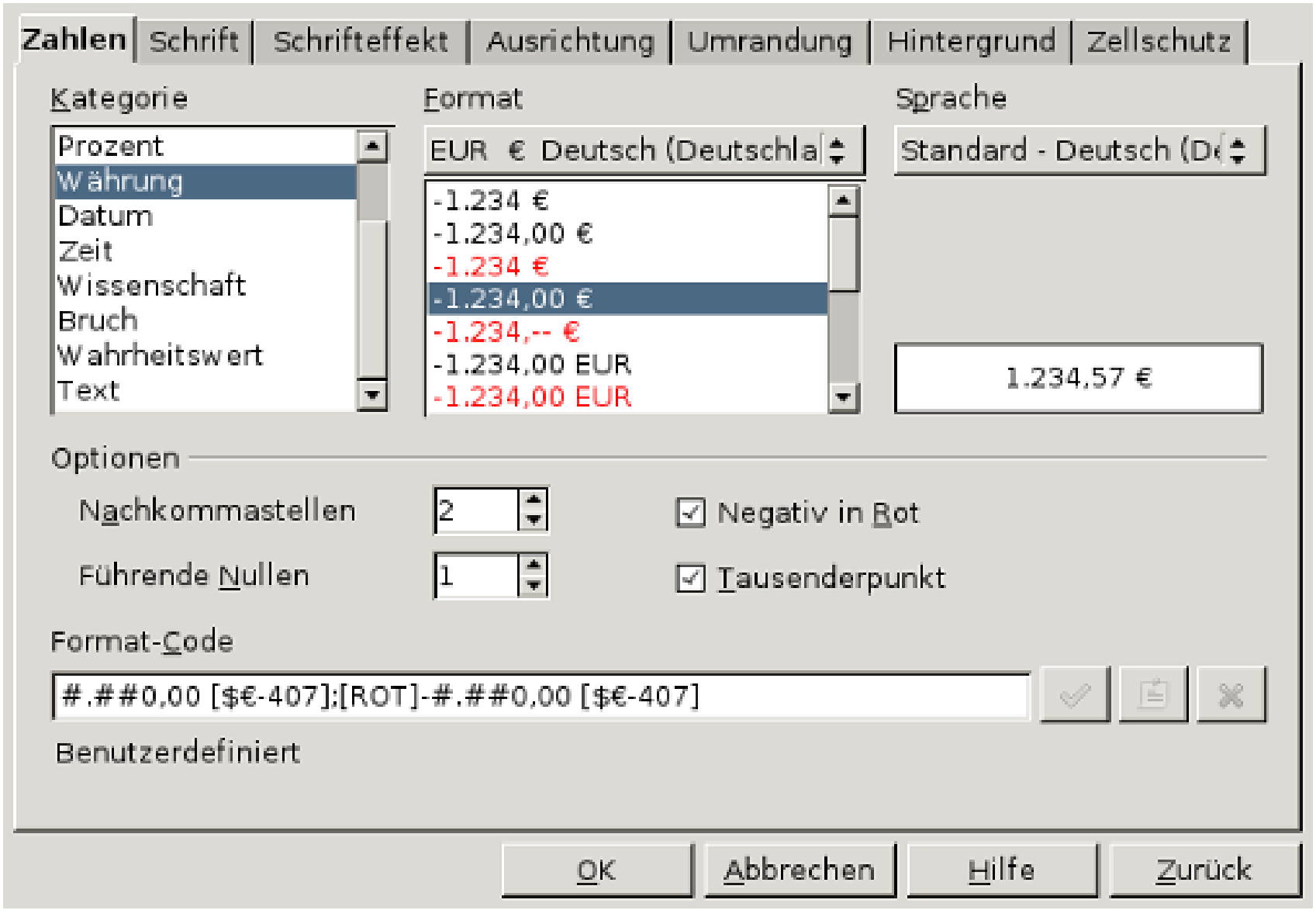


Figure 3: Formatierung einer Zelle

Zeile

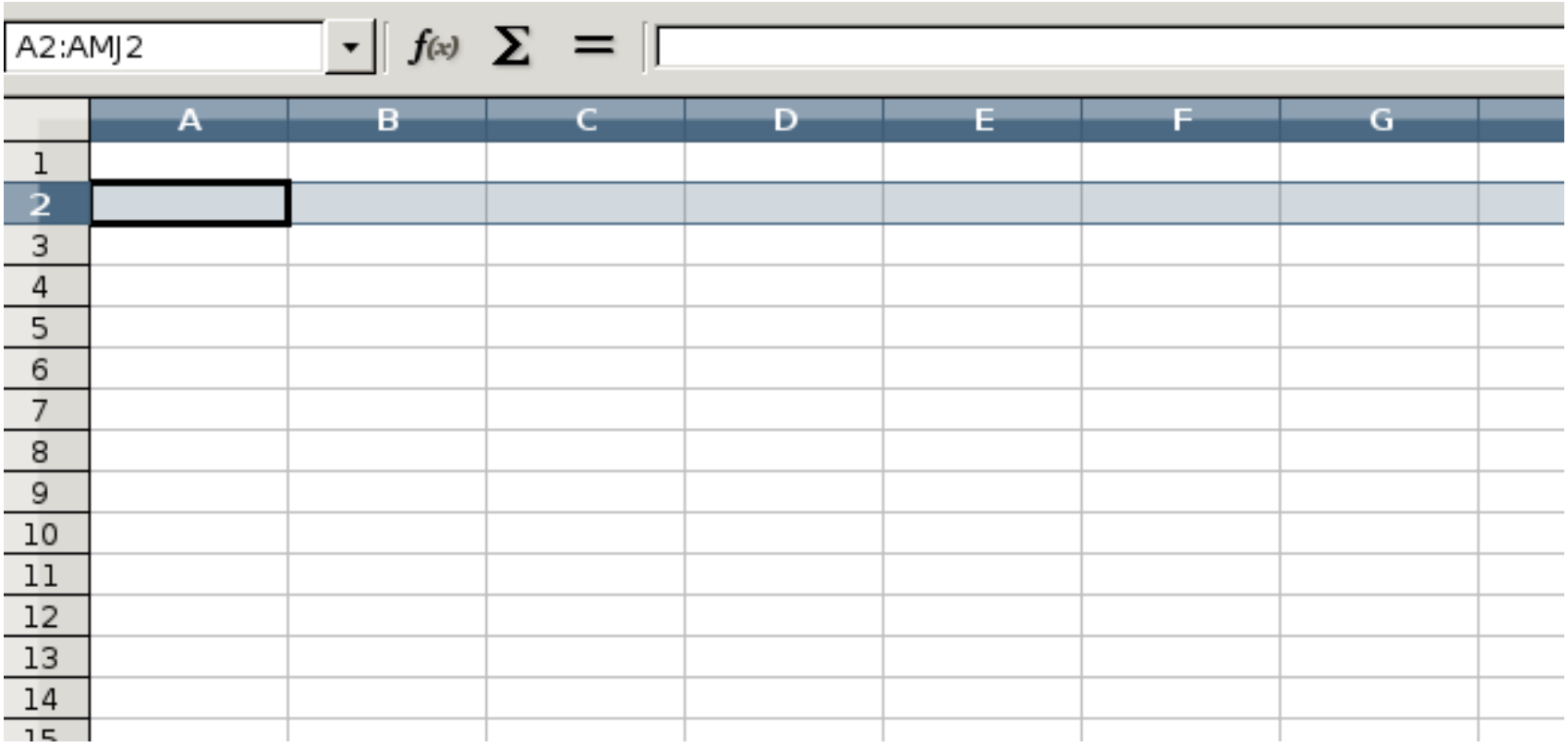
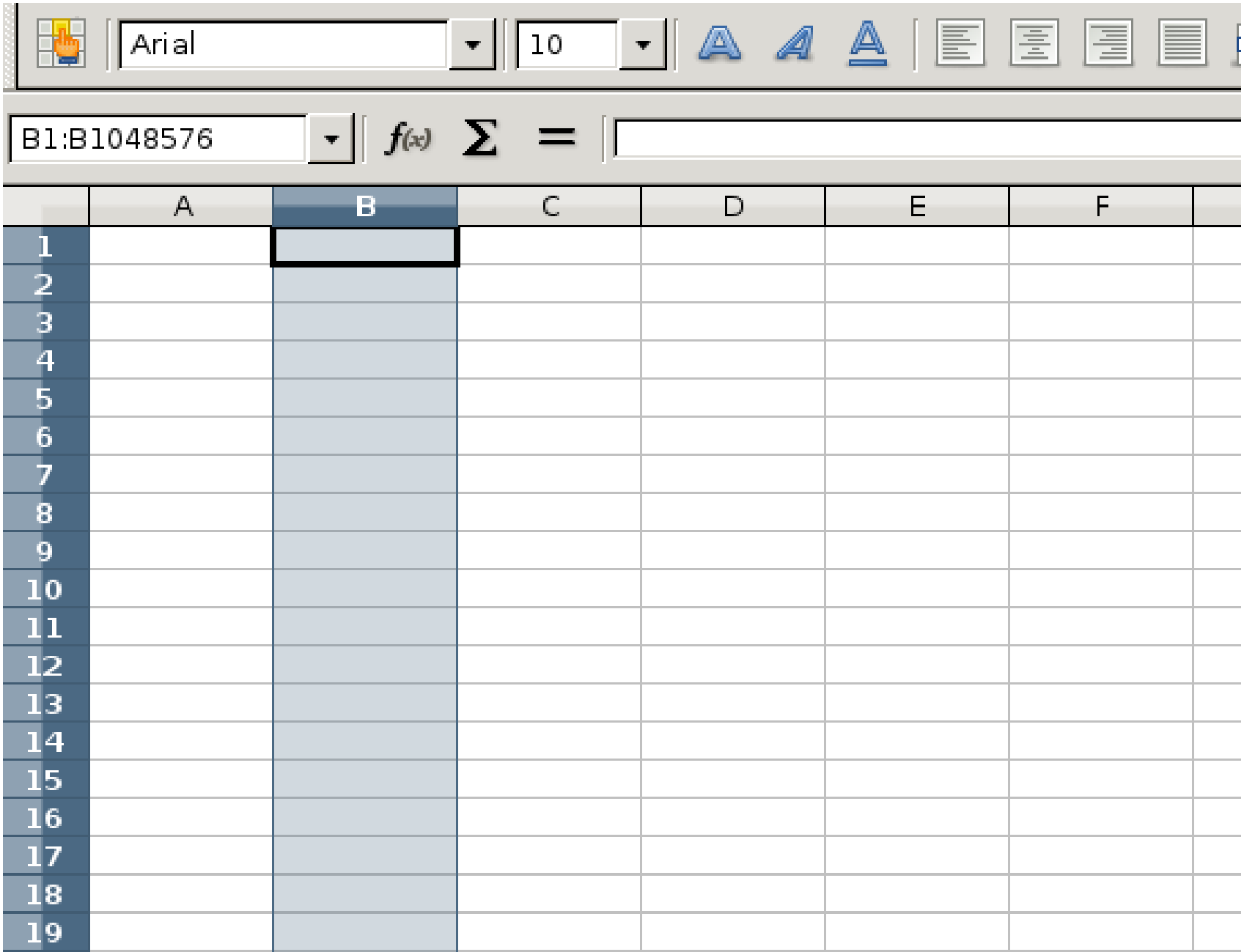


Figure 4: Zeile

Spalte


Arial
10
B1:B1048576
A
B
C
D
E
F
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

Seiten

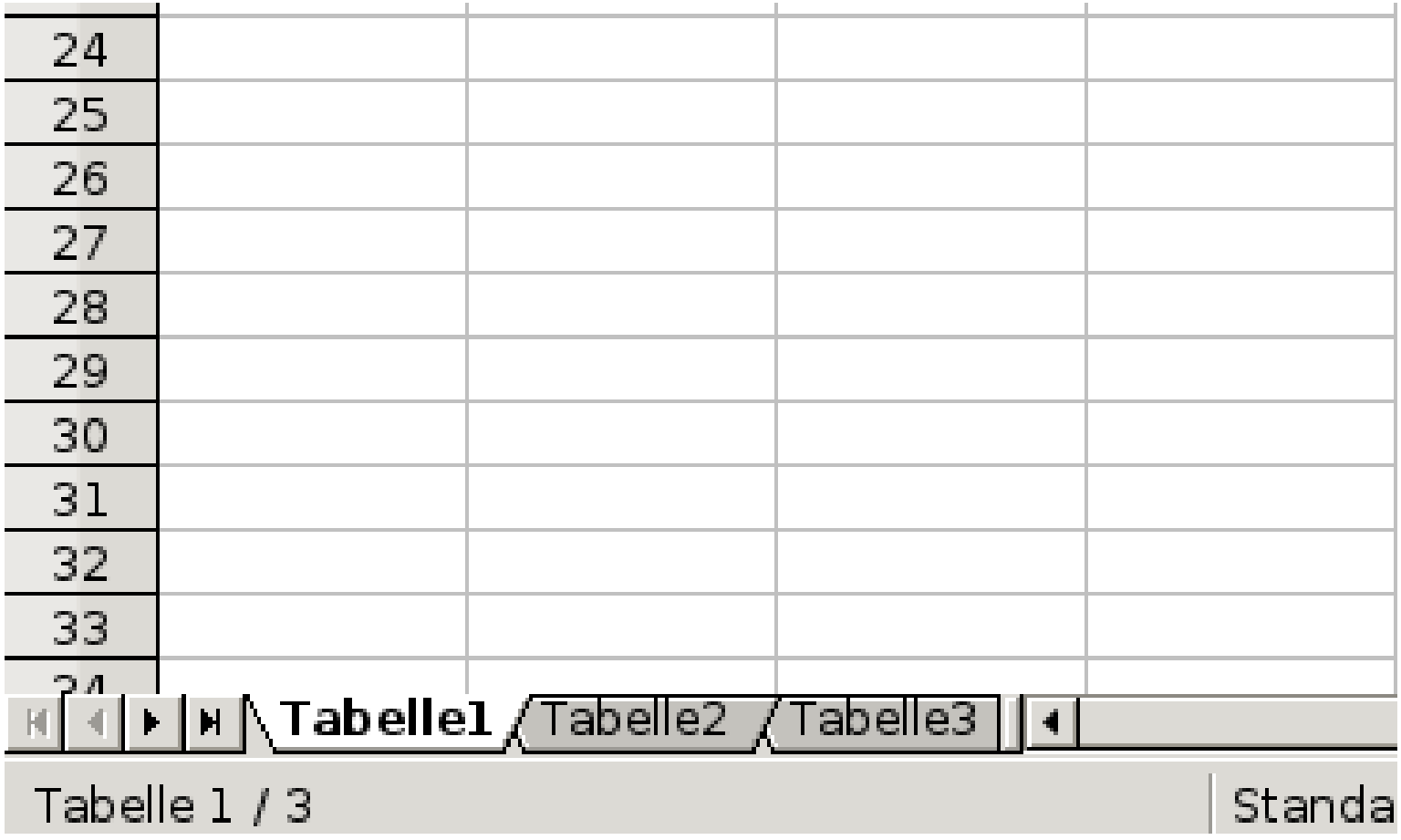


Figure 6: Seiten

$\rightarrow$ erweiterte Adresse: Tabelle2.A1

Verbesserung der Datenhandhabung

- direkte Manipulation der Daten
- Adressieren einzelner oder mehrerer Zellen

  $\rightarrow$ man *sieht* die Daten/-bereiche
- Festlegen des Zelltyps und der Darstellungsform
- mehrere Datensätze können zusammen bearbeitet werden (Seiten)

Berechnungen

- Produkte bieten verschiedene Funktionen an

  $\rightarrow$ es muss nicht selbst programmiert werden
- Auswahl der gewünschten Daten
- optional: Ziel der Operation angegeben
- gewünschte Operation angeben

es muss nichts programmiert werden $\rightarrow$ Datenverarbeitung auch für Nicht-Informatiker möglich

Berechungsbeispiel: Addition

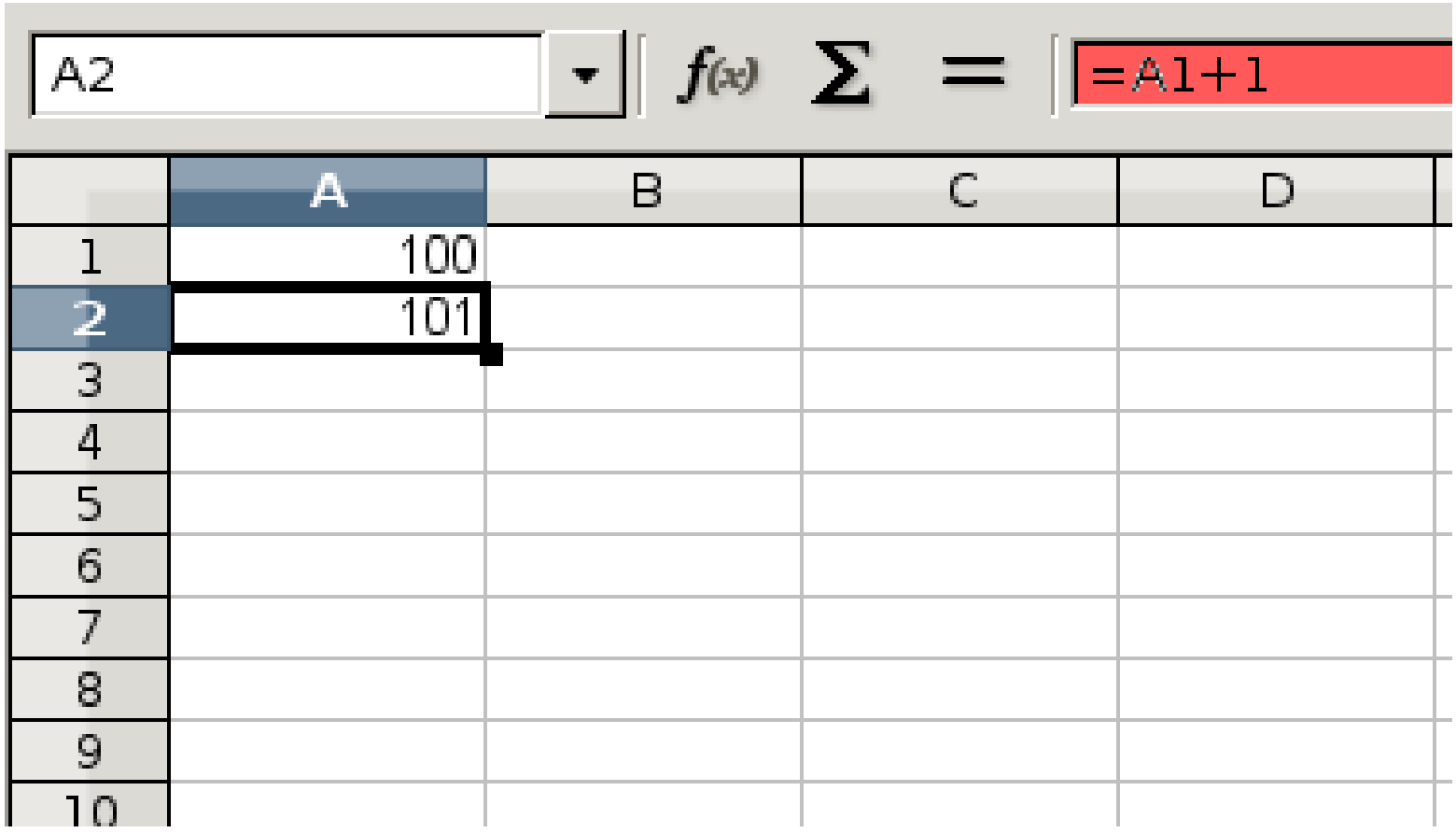


Figure 7: Addition

→ Zellen können Daten *oder* Formeln beinhalten

Operationen

- =A1+10
- =A1*16%
- =A1 * A2
- =RUNDEN(A1;1)
- =EFFEKTIV(5%;12)
- =B8-SUMME(B10:B14)
- =SUMME(B8;SUMME(B10:B14))

Weitere Verbesserung: Verknüpfung Funktion-Daten

Daten auswählen, Funktion auswählen → Ergebnis wird in einer freien Zelle abgelegt

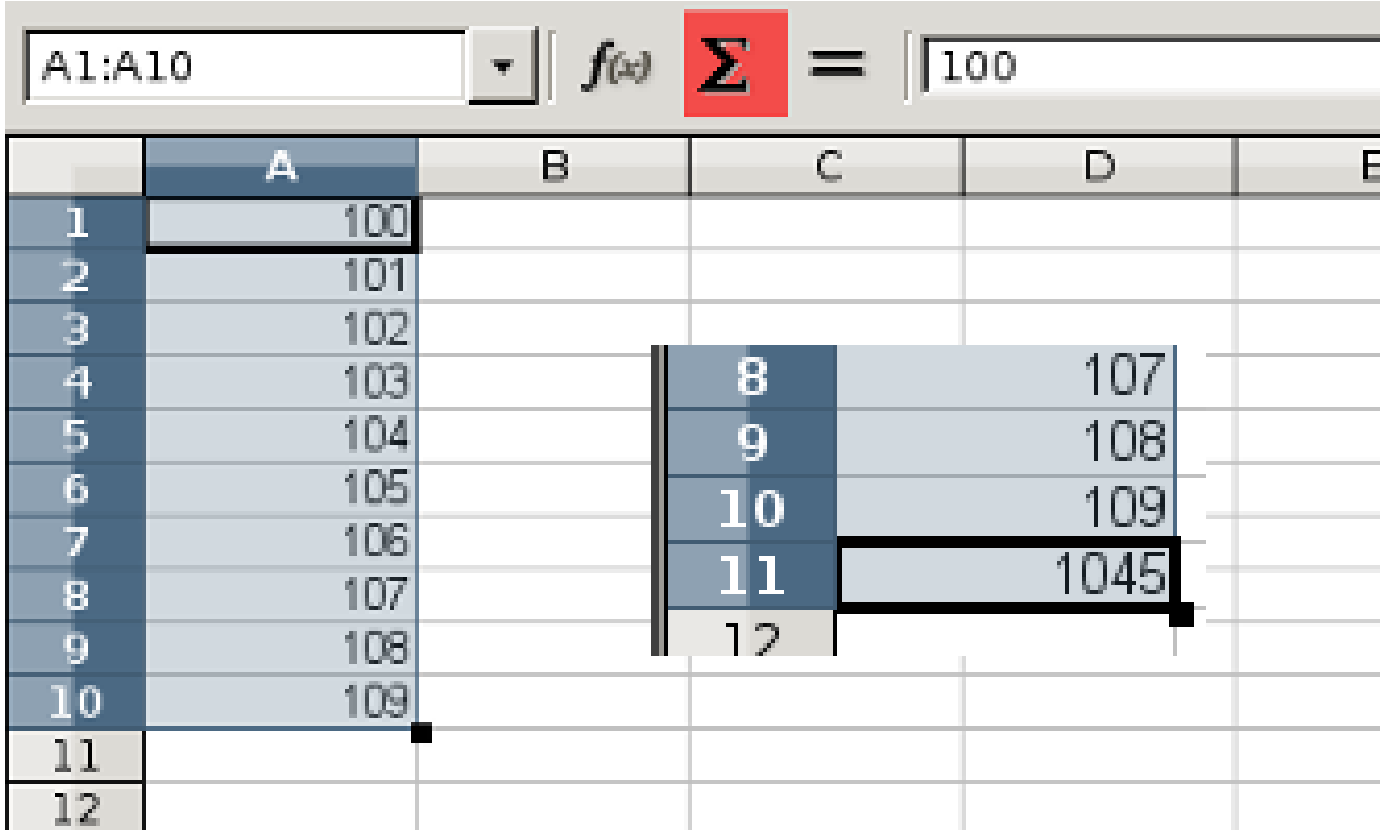


Figure 8: Summenfunktion

Adressen und Referenzen

- relative Adressierung: A1

  $\rightarrow$ werden beim Einfügen/Löschen von Zellen automatisch angepasst

- absolute Adressierung: A1

  feste Angabe von Spalte und/oder Zeile

Vorteil der relativen Adressierung: automatische Korrektur beim Kopieren von Formeln

Beispiel für relative Adressierung

B2:B10 | $f(x)$ | Σ | $=$ | =SIN(A10)

	A	B	C	D
1	100	-0,506365641		
2	101	0,452025787		
3	102	0,994826791		
4	103	0,622988631		
5	104	-0,321622403		
6	105	-0,970535284		
7	106	-0,7271425		
8	107	0,184781745		
9	108	0,926818505		
10	109	0,816742607		
11				

Figure 9: Relative Adressierung

Ausgewählte Funktionen

- Vergleichsoperatoren wie $<$, $>$ usw.
- logische Ausdrücke:

  =WAHR(), =FALSCH(), =NICHT(), =UND(), =ODER()
- Bedingungen: =WENN()

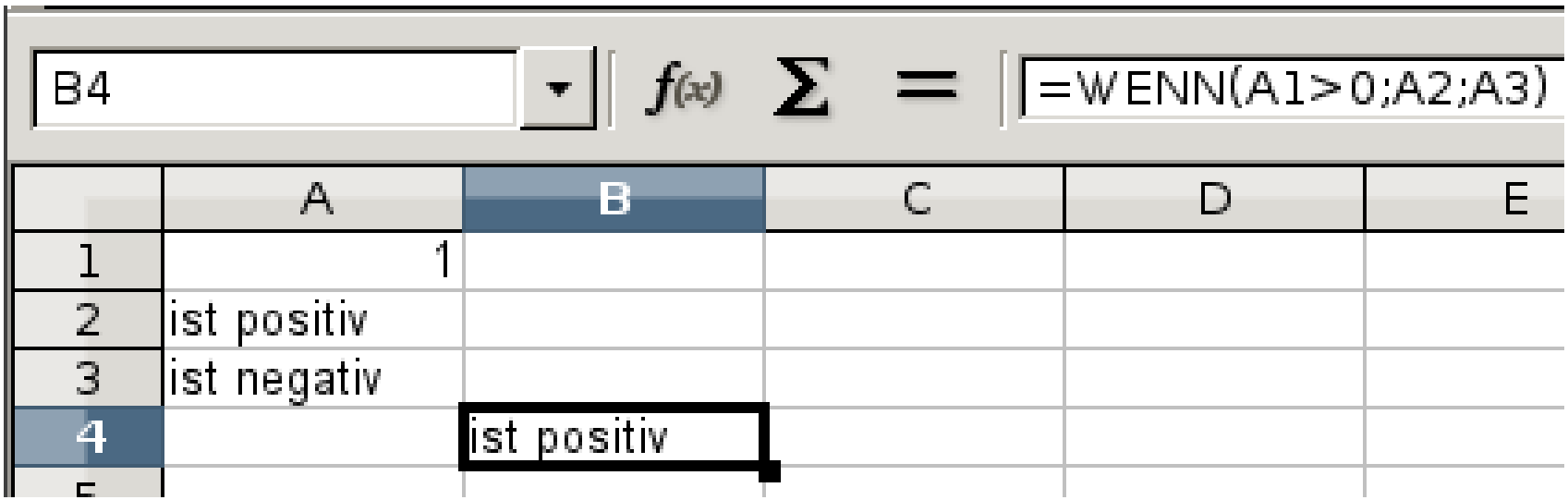


Figure 10: Bedingungen

- Schleifen?

Verbesserung im Vergleich zu Daten in Dateien

- einfache Auswahl der Funktionen und der zugehörigen Zellen
- viele (auch komplizierte) vorgefertigte Funktionen
- (gestaffelte) Verarbeitung mehrerer Funktionen
- visuelle Entwicklung der Berechnungsvorschrift

$\rightarrow$ Flexibilität der Tabellenkalkulation erlaubt beliebige Form der Organisation der zu verwaltenden Daten

Zusätzliche Funktionalität: Visualisierung von Daten

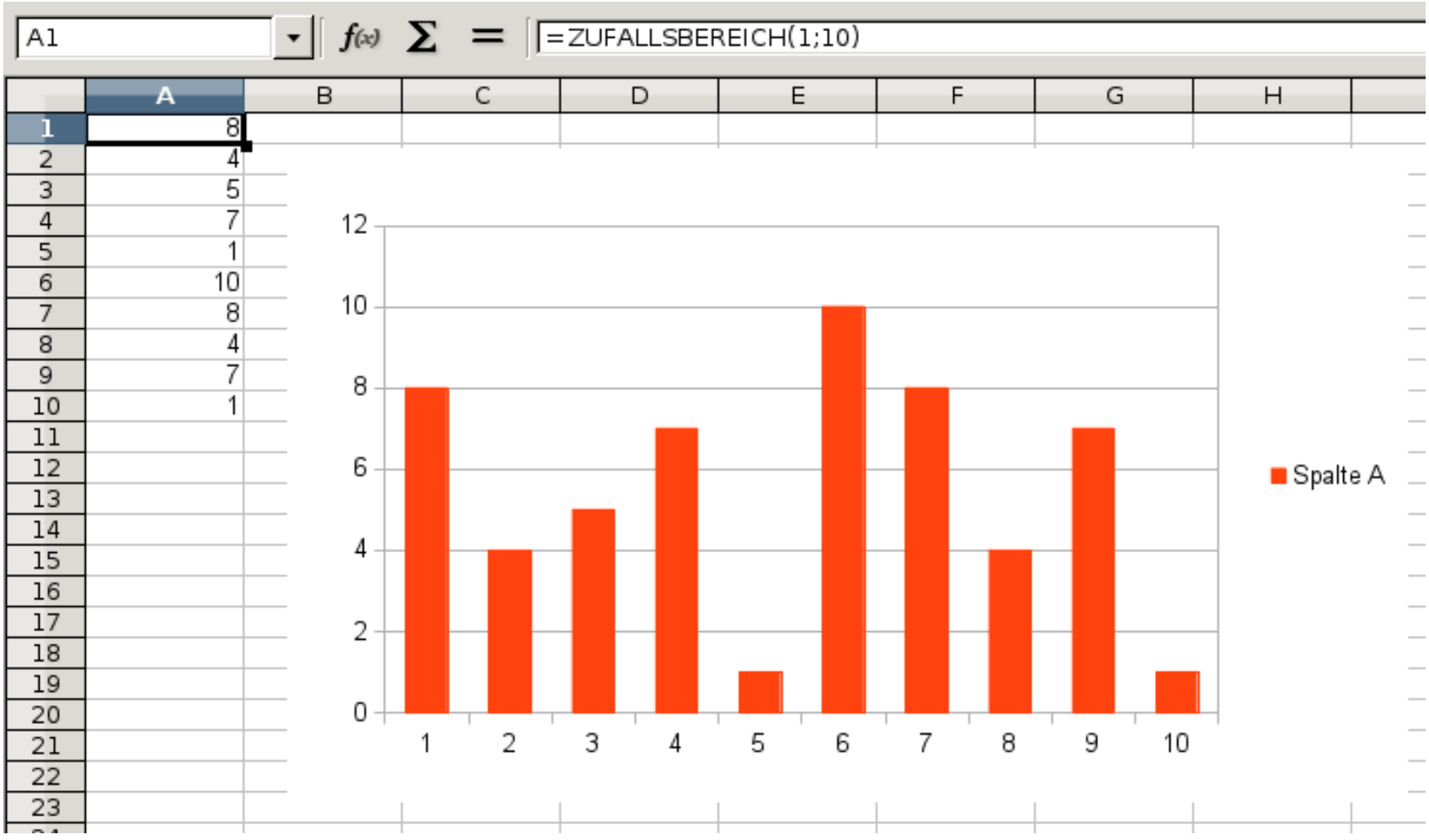


Figure 11: Säulendiagramm

Lösungsalternative *Tabellenkalkulation*

Geschichte

- 1979: VisiCalc (Dan Bricklin) für den Apple II (später u.a. auch für Apple III, Atari 800 und 1981 für den IBM PC)
- 1983: aus VisiCalc wurde Lotus 1-2-3 (Mitch Kapor) für den PC
- 1985: Multiplan von Microsoft für CP/M und MS-DOS und Excel für Mac
- 1987: Excel für Windows 1987
- 1987: Quattro Pro von Borland
- offene/freie Software: Gnumeric, Open/LibreOffice Calc, KSpread

Lösungsalternative *Tabellenkalkulation*

Funktionsweise:

- Wie arbeitet man mit einer Tabellenkalkulation?
- Was passiert „unter der Haube“?

Lösungsalternative *Tabellenkalkulation*

Funktionsweise

- auf Daten eine bestimmte „Rechenvorschrift“ anwenden

  $\rightarrow$ *Algorithmus* zur Datenverarbeitung

- Beispiele:
  - Sortieren (auf-/absteigend)
  - (Zwischen-)Summen bilden

das alles auch mehrstufig (Summen von Summen, wechselnde Sortierkriterien usw.)

Vergleich beider Lösungsalternativen

pro Tabellenkalkulation

- bietet übersichtliche Organsiationsmöglichkeiten für Daten
- Berechnungen etc. sind einfach umzusetzen
- flexibel bei Änderungen

contra:

- muss erst „vorbereitet“ werden (Tabellen entwerfen usw.)

pro Programmierte Lösung:

- sofort einsatzbereit
- zusätzliche Funktionen (Bankeinzug, Mailinglisten, Kalender, Materiallisten usw.)
- funktioniert effizient mit großen Datenmengen

contra

- nicht flexibel bei strukturellen Änderungen
  z.B. wenn zusätzliche Daten sollen verwaltet werden sollen

Grenzen

Wann sollte eine Tabellenkalkulation besser nicht eingesetzt werden?

Grenzen

- sehr große Datensätze
- lange laufende Berechnungen
- Nutzung mehrerer Prozessoren/Maschinen
- komplizierte Berechnungen, zu denen die Tabellenkalkulation keine passenden Funktionen bietet
- Geschwindigkeit!
- Fehler in der Tabellenkalkulation

Zusammenfassung

Was haben wir angesehen?

- grundlegende Funktion einer Tabellenkalkulation
- Vergleich mit einer programmierten Lösung
- Vor-/Nachteile bei der Datenhandhabung und Berechnung
- Grenzen

Programmierte Lösung

Beispiel-Daten `bikeshare_daten_2022.csv`

- Format: CSV
- je Zeile ein Datensatz
- Datenbestandteile durch Trennzeichen abgegrenzt (Komma)

Programmierte Lösung: Demonstration

Beispiellösung: `datenverarbeitung.py`

- Eingabe: `daten = extrahiere_daten()`
- Verarbeitung: `ergebnis = durchschnittliche_ausleihe(daten)`
- Ausgabe: `ausgabe(ergebnis)`

Programmierte Lösung

- allgemein sinnvolles Vorgehen:

  Rohdaten (CSV) $\rightarrow$ Einlesen $\rightarrow$ Umwandlung $\rightarrow$ Verarbeitung $\rightarrow$ Ausgabe

- Mögliche Probleme oder Schwierigkeiten?

Tabellenkalkulation: Umsetzung

- Anforderungen: s. programmierte Lösung
- komfortabel: Datei `bikeshare_data_2022.csv` in Tabellenblatt einlesen (Trennzeichen angeben)

Tabellenkalkulation: Pivot-Tabelle erstellen

- Menüpunkt auswählen
- Zeilenfelder: `station`
- Datenfelder: `count`
- Anpassen der Funktion auf *Mittelwert*

Rückblick

Was haben wir uns angesehen?

- grundlegende Funktion einer Tabellenkalkulation
- Vergleich mit einer programmierten Lösung
- Vorteile bei der Datenhandhabung und Berechnung
- Grenzen

Literatur

- Bensberg, Grob, and Reepmeyer (2008)
- Depner (2012)
- Hay-Tun (2013)

Bensberg, Frank, Heinz Lothar Grob, and Jan-Armin Reepmeyer. 2008. *Excel Für Wirtschaftswissenschaftler*. München: Vahlen. https://katalogplus.ub.uni-bielefeld.de/cgi-bin/new/_titel.cgi?katkey=2409207/&query=excel/&vr=1/&nurOnl=1/&pagesize=50/&sprache=GER/&bestand=lok/&sess=2eec50f8b53930a2be47ae3aea1ca078.

Depner, Eduard. 2012. *Excel Für Fortgeschrittene Am Beispiel Der Darlehenskalkulation Und Investitionsrechnung*. Wiesbaden: Springer Fachmedien Wiesbaden. http://dx.doi.org/10.1007/978-3-8348-1978-9.

Hay-Tun. 2013. "Tabellenkalkulation - Die Formeln." 2013. http://de.ccm.net/contents/123-tabellenkalkulation-die-formeln.