

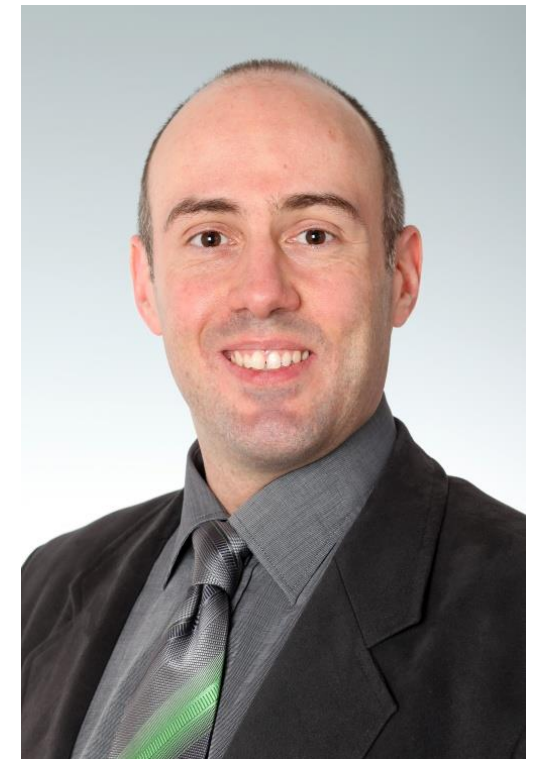
Datenvisualisierung mit Chart.js

JavaScript Track 3/4

Thomas Hirt

Über den Sprecher

- Studium der Biochemie und des Software Engineering
- betreut seit 25+ Jahren Kleinunternehmen und Privatkunden in Informatikbelangen
- Geschäftsführer und Inhaber der Tek:Guides GmbH
- Windows Umgebungen als Hauptbetätigungsfeld
- Einsatz von FileMaker als Entwicklungsplattform seit 2009



Inhalt

- Was ist Chart.js?
- Überblick über die Fähigkeiten von Chart.js
- Einbindung von Chart.js in FileMaker
- Grundsetup zur Erzeugung von Kurven und Graphen
- Beispiele
 - Balkendiagramm
 - Kurvendiagramm
 - Tortendiagramm
 - Polar-Area-Diagramm
 - Radar-Diagramm
 - Blasendiagramm
 - Streudiagramm

Was ist Chart.js?

Was ist Chart.js?

- Chart.js ist eine beliebte JavaScript-Bibliothek zur Visualisierung von Daten
- 2013 erstmals veröffentlicht
- wird immer noch aktiv weiterentwickelt
- ist relativ einfach einzusetzen und auch relativ vielfältig, kann aber nicht mit grossen Projekten, wie z.B. D3.js, mithalten
- performant
- wird unter der sehr liberalen MIT-Lizenz veröffentlicht

Überblick über die Fähigkeiten

- alle Details unter <https://www.chartjs.org/docs/latest/>
- Chart.js enthält eine schöne Auswahl an gängigen Diagramm-Arten
- Chart.js ist durch Plugins erweiterbar und es gibt zahlreiche Plugins
- interaktive Möglichkeiten durch JavaScript-Programmierung
 - dynamisches entfernen/nachladen von Datensätzen
 - dynamischer Anpassung der Skalierung
 - dynamische Anpassung des Aussehens
 - u.v.m

Überblick über die Fähigkeiten

- enthaltene Diagramm-Arten
 - Balkendiagramm
 - Kurvendiagramm
 - Tortendiagramm und Donut-Diagramm
 - Blasendiagramm
 - Streudiagramm
 - Polar-Area-Diagramm
 - Radar-Diagramm
- gemischte Diagrammarten möglich
- verschiedene Achsen und Achsenarten möglich

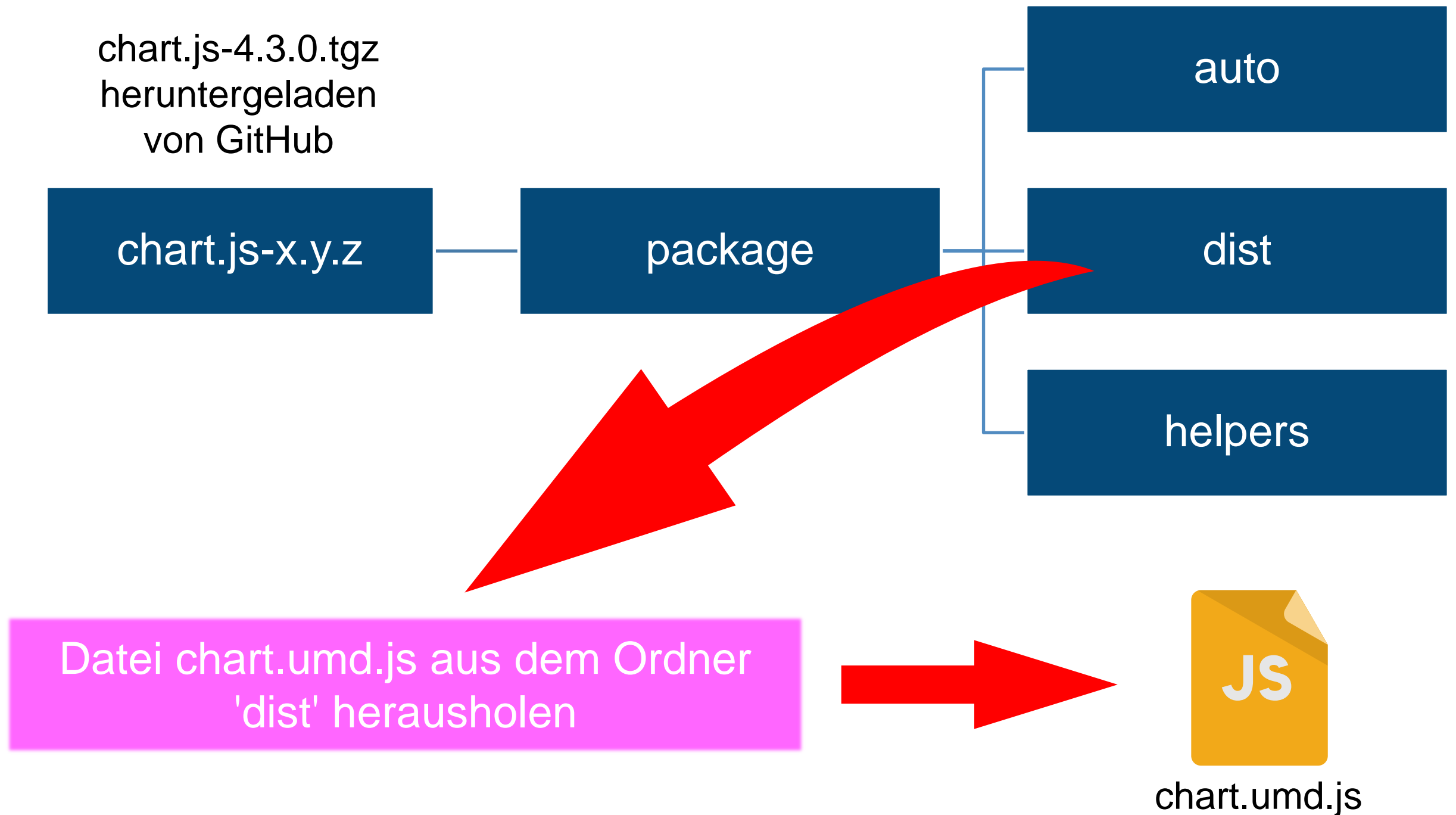
Einbindung von Chart.js in FileMaker

Einbindung von Chart.js in FileMaker

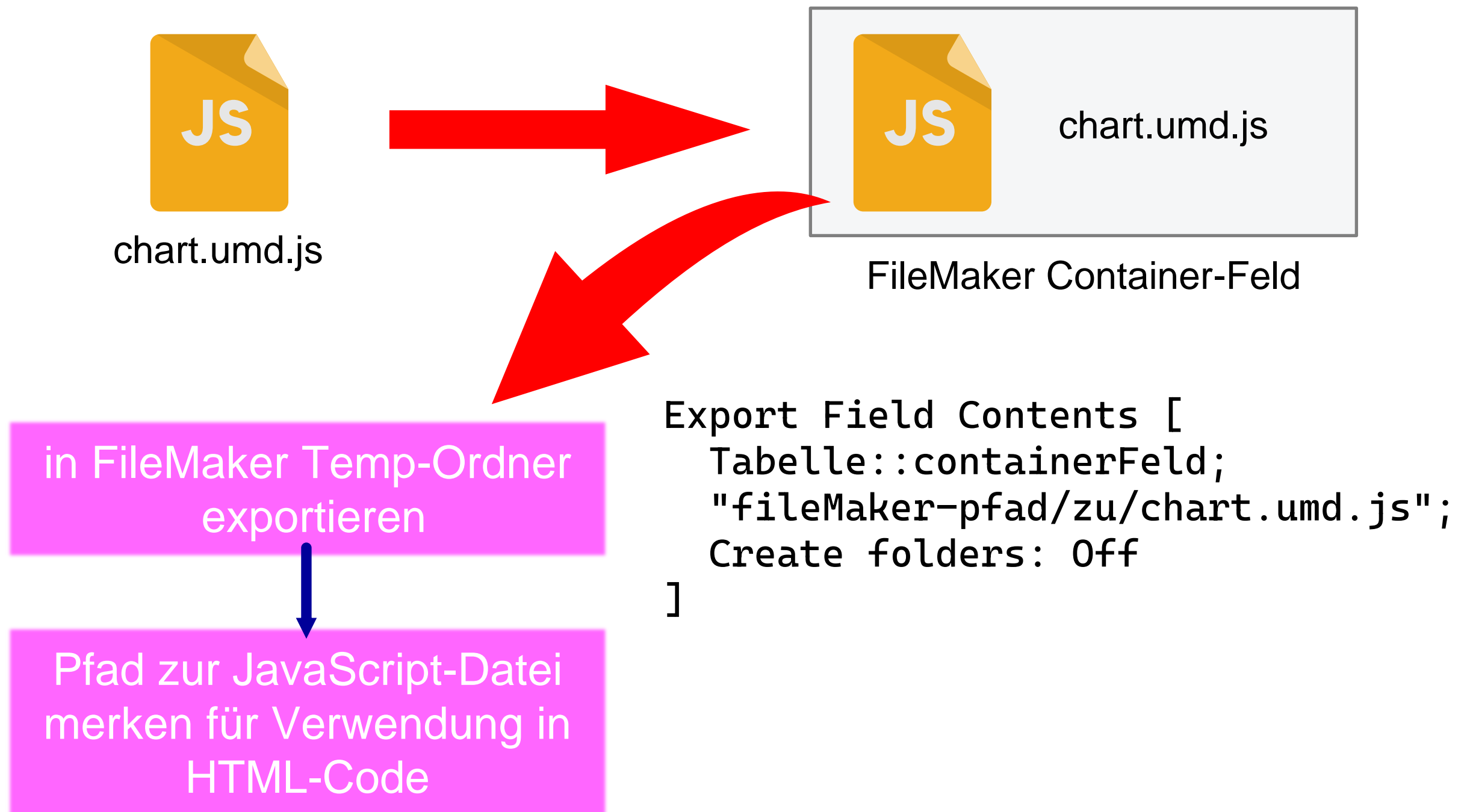
- Chart.js hat sich über die Jahre von einer einfach einbindbaren JavaScript-Library wegbewegt hin zu einer auf node.js ausgerichteten Installationsweise
- kann aber immer noch auf mehrere Arten eingebunden werden
 - npm
 - Paket-Manager für node.js
 - online Software Bibliothek ("registry")
 - dynamische Einbindung
 - CDN
 - 2 Content Delivery Networks
 - dynamische Einbindung
 - geeignet für FileMaker Web Viewer
 - abhängig von Internetverbindung
 - abhängig von Online-Verfügbarkeit
 - GitHub
 - Quellcode für node.js-Umgebungen
 - herunterladbares, fixfertiges TGZ-Archiv
 - geeignet für FileMaker Web Viewer
 - unabhängig von Internetverbindung
 - lokale Einbindung muss selbst "geskriptet" werden

Einbindung von Chart.js in FileMaker

Einbindung von Chart.js in FileMaker



Einbindung von Chart.js in FileMaker



Grundsetup zur Erzeugung von Kurven und Graphen

Webseite für Graphen mit Chart.js

```
<!DOCTYPE html>
<html>
<head>
  <title>Chart.js Demo</title>
</head>
<body>
  <div><canvas id="meinGraph"></canvas></div>
  <script src="file:///pfad/zu/temp/chart.js"></script>

  <script>
    const c = document.getElementById('meinGraph');
    new Chart(c,{

      });
  </script>
</body>
</html>
```

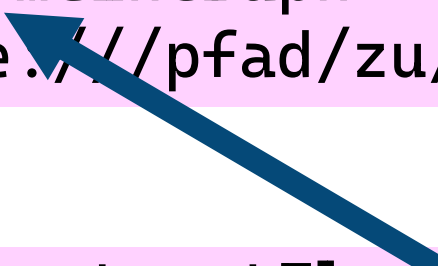
Webseite für Graphen mit Chart.js

```
<!DOCTYPE html>
<html>
<head>
  <title>Chart.js Demo</title>
</head>
<body>
  <div><canvas id="meinGraph"></canvas></div>
  <script src="file:///pfad/zu/temp/chart.js"></script>

  <script>
    const c = document.getElementById('meinGraph');
    new Chart(c, {

      Definition des Graphen als JavaScript-Objekt

    });
  </script>
</body>
</html>
```



grundsätzliche Vorgehensweise

Webseite für Graphen mit Chart.js aus vorgefertigten Versatzstücken in FileMaker zusammenbauen



Konfiguration des Graphen gemäss Dokumentation von Chart.js vorbereiten und in JavaScript-Objekt einfügen



zu visualisierende Daten aus FileMaker Datensätzen in JavaScript-Objekt zusammenfassen



fertiges JavaScript-Objekt in Webseite einbauen und Webseite in den Temp-Pfad exportieren



zusammengebaute Webseite im WebViewer anzeigen

zu visualisierende Daten aus FileMaker-Datensätzen zusammenfassen

- für kleinere Datenmengen
 - Suche ausführen
 - mittels Skript in Schleife durch alle Datensätze gehen
- universell
 - ExecuteSQL
 - ganze Listen über Beziehungen abgreifen

einfaches Chart-Beispiel

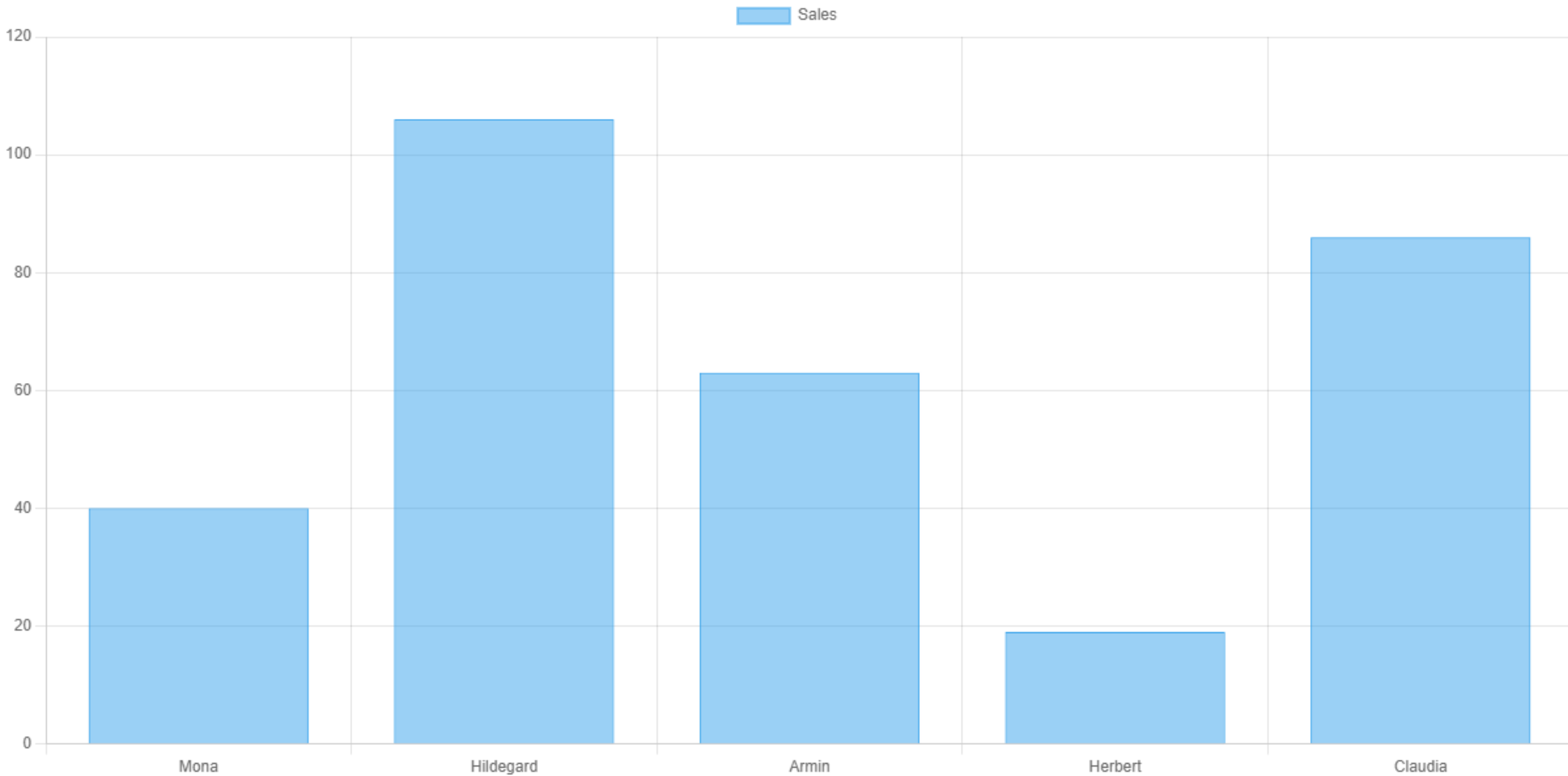
```
new Chart(c, {  
  type: 'bar',  
  data: {  
    labels: ['Q1', 'Q2', 'Q3', 'Q4'],  
    datasets: [  
      {  
        label: 'Olga',  
        borderWidth: 1,  
        data: [1200, 980, 890, 2310]  
      },  
      {  
        label: 'Stefanie',  
        borderWidth: 1,  
        data: [900, 810, 1230, 1980]  
      }  
    ]  
  },  
  options: {  
    scales: {  
      y: { beginAtZero: true }  
    }  
  }  
})
```

Chart.js Beispiele

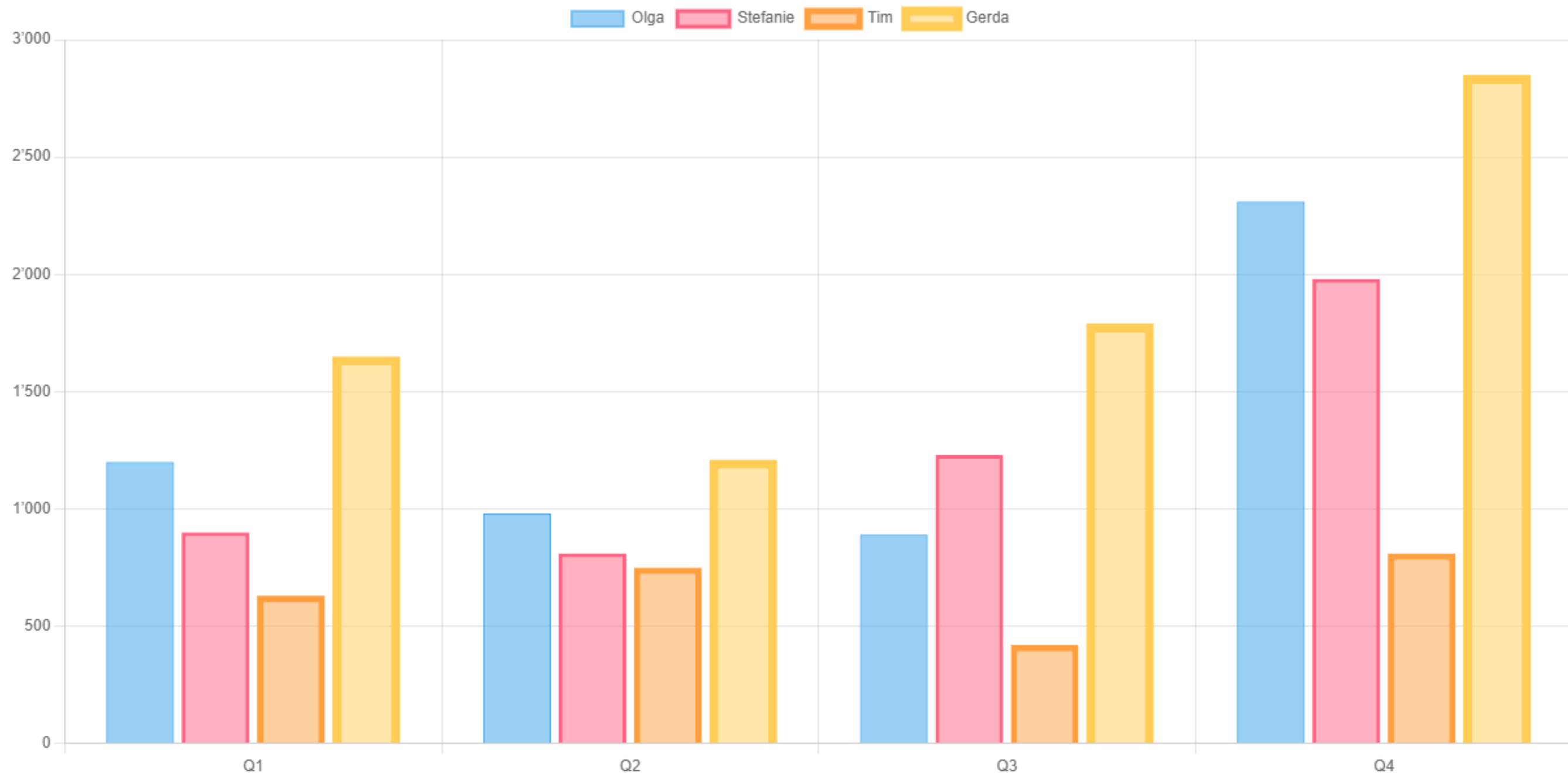
Live Demo

- Die folgenden Folien zeigen die Beispielgraphen, welche auch in der Live Demo gezeigt wurden.
- Die Details zu den Beispielen können Sie der Demo-Datei entnehmen.

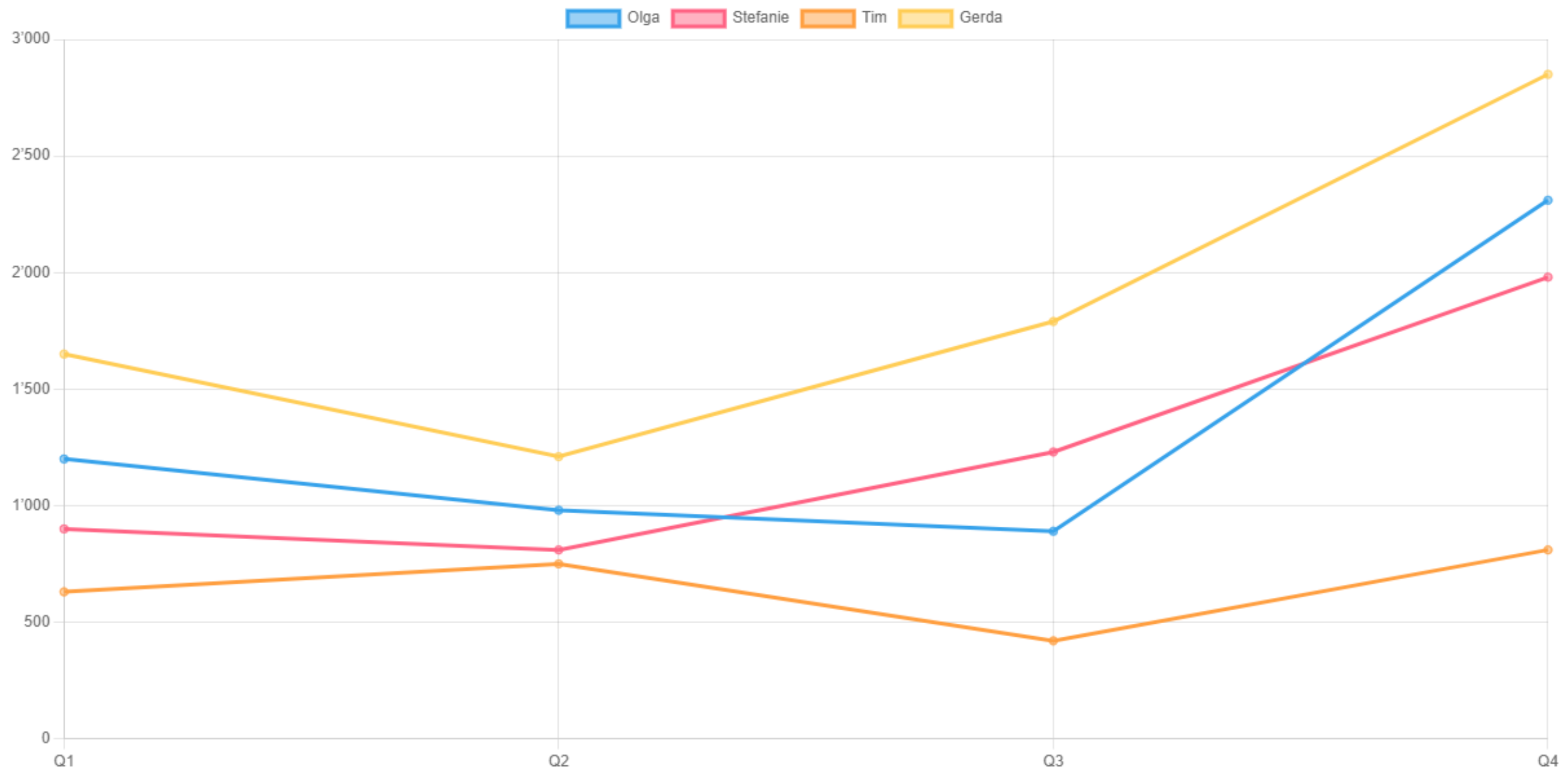
Beispiel 1: Balkendiagramm



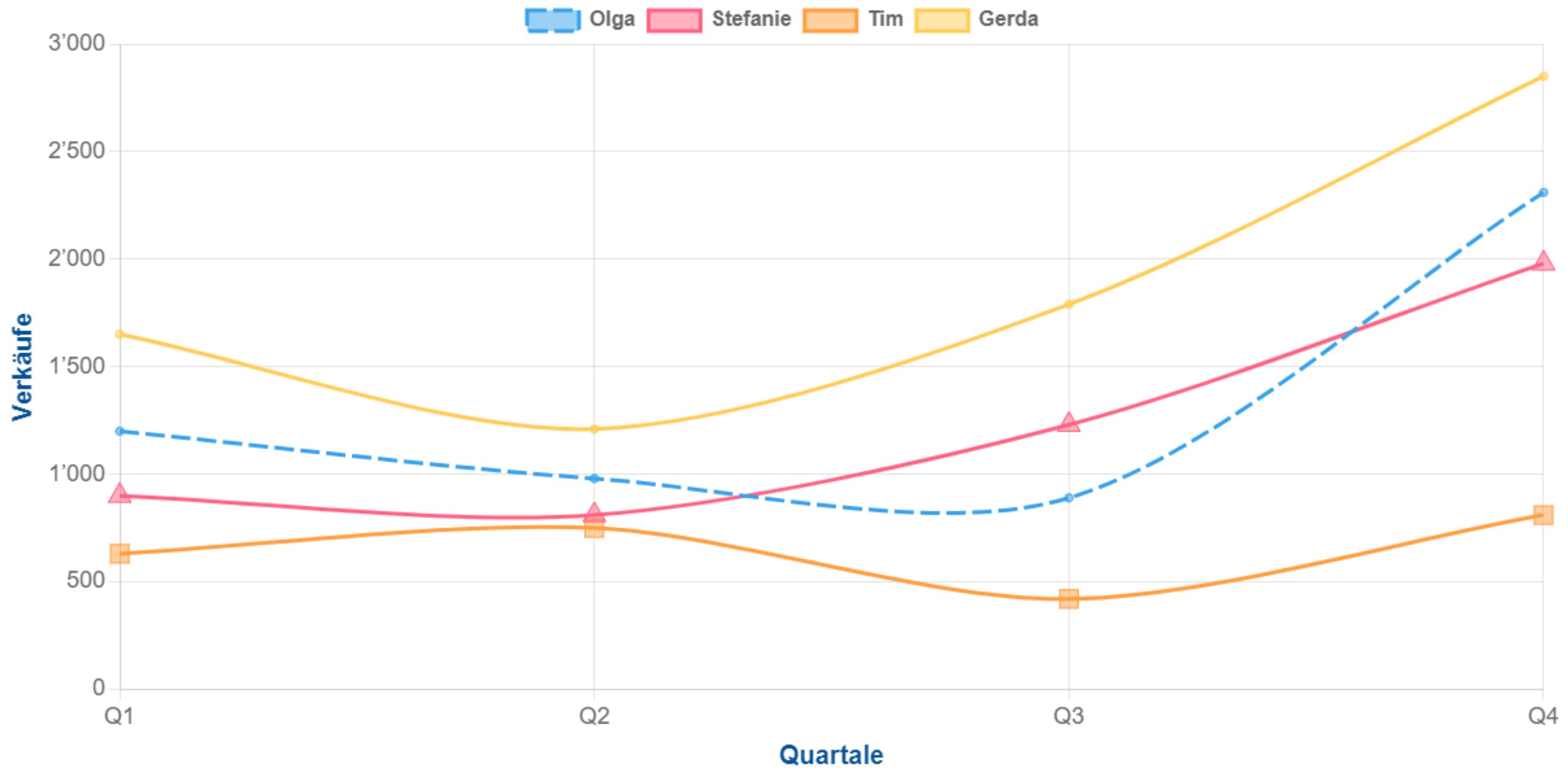
Beispiel 2: Balkendiagramm



Beispiel 3: Kurvendiagramm

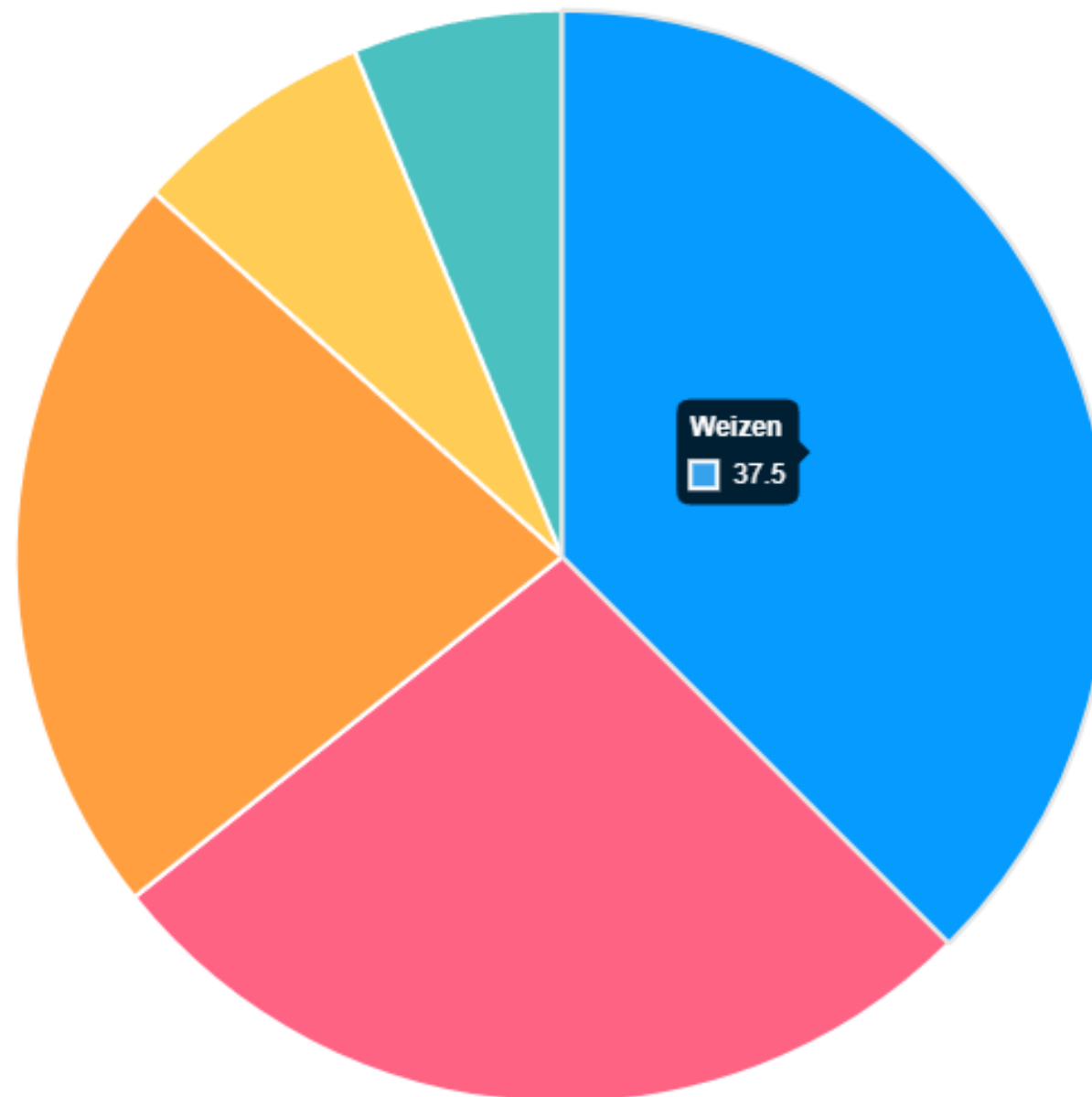


Beispiel 4: Kurvendiagramm

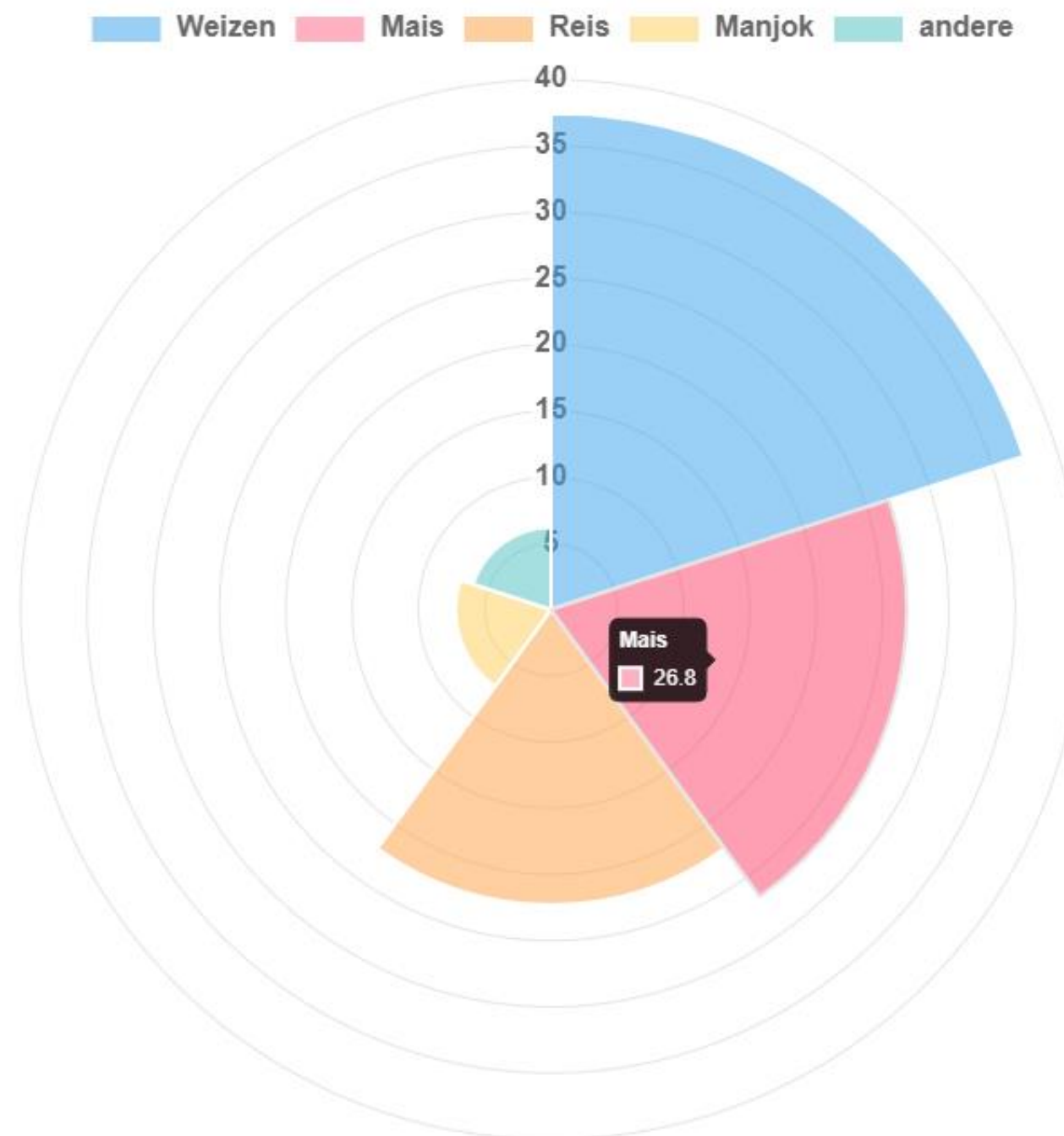


Beispiel 5: Tortendiagramm

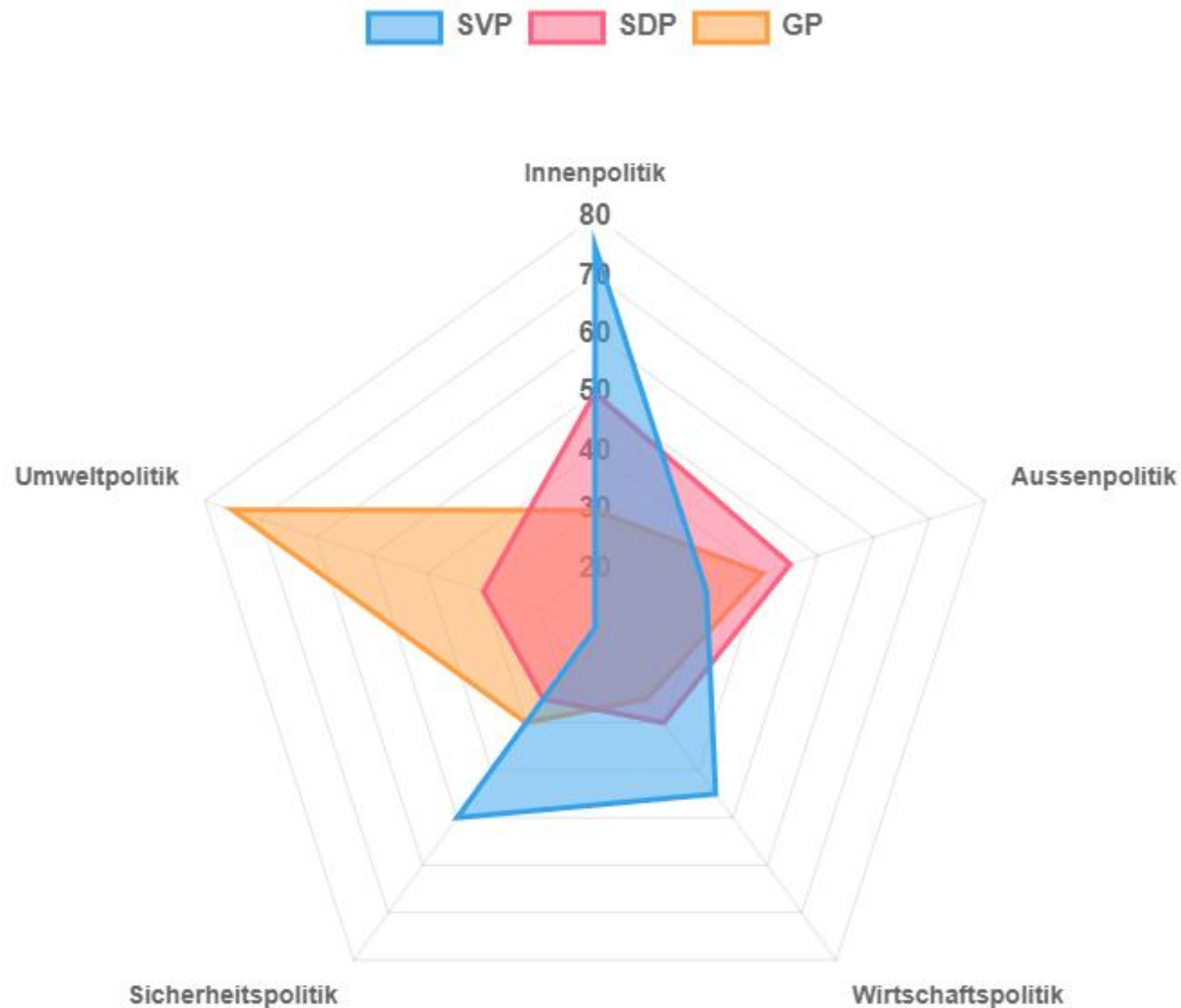
■ Weizen ■ Mais ■ Reis ■ Manjok ■ andere



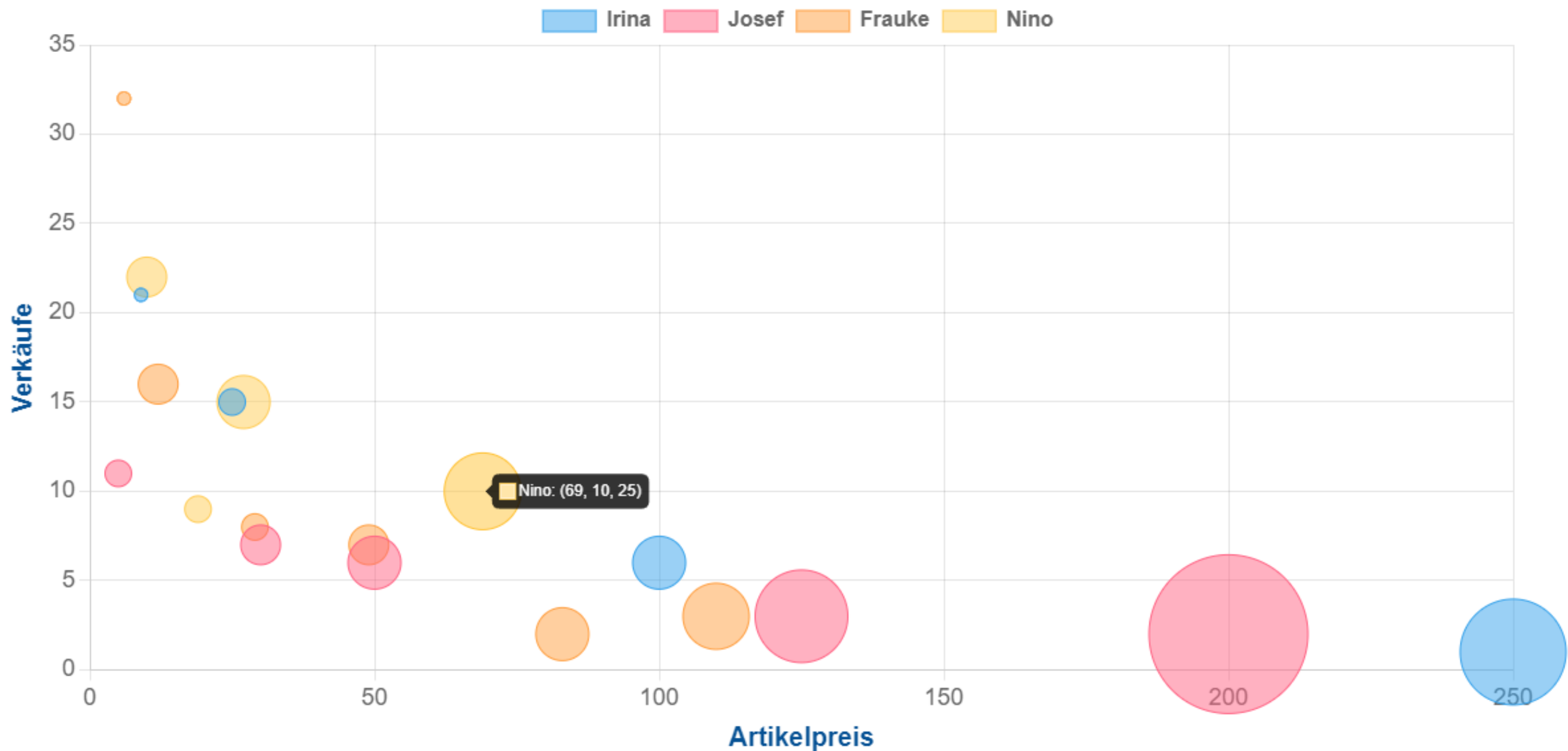
Beispiel 6: Polar-Area-Diagramm



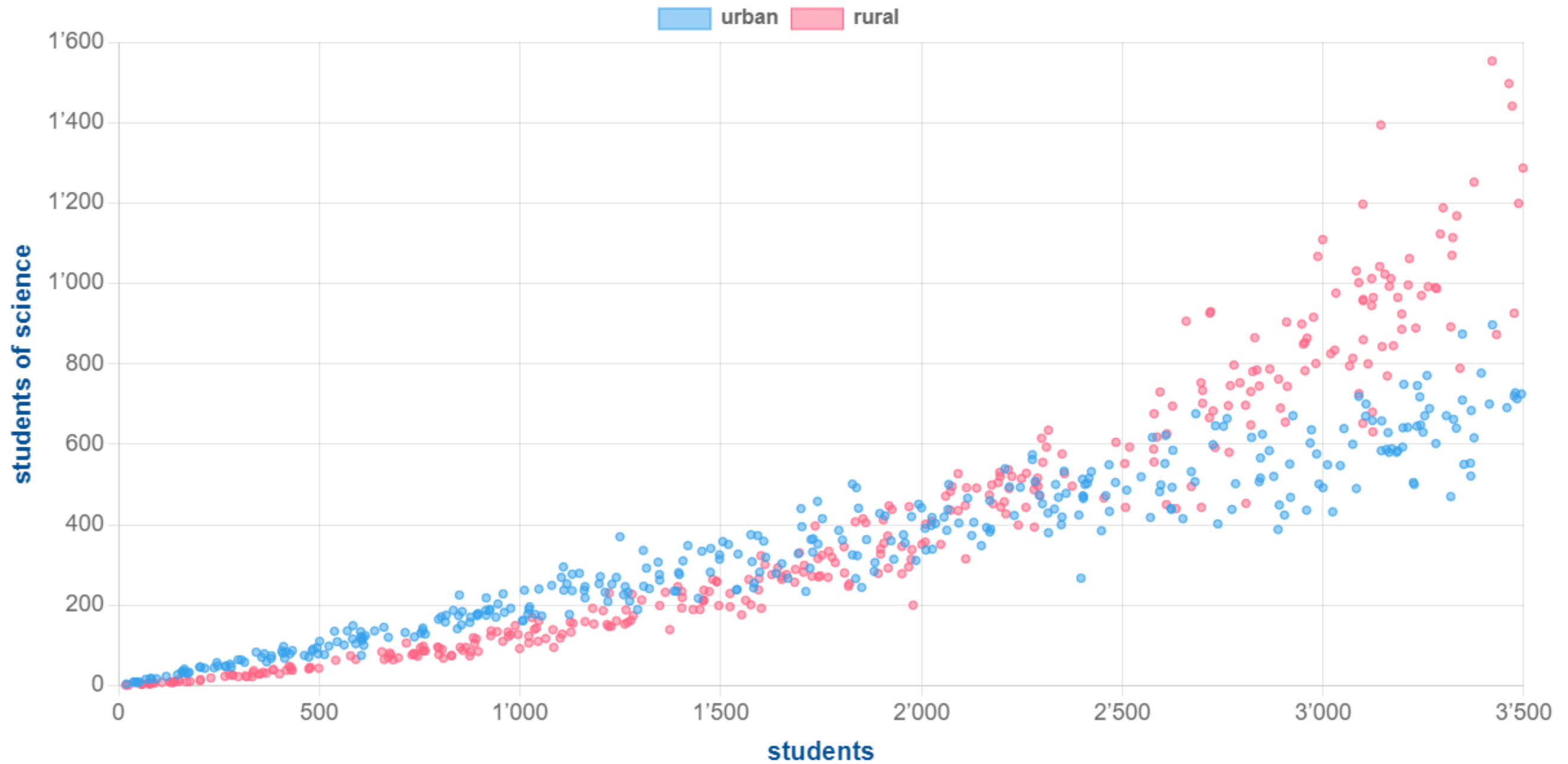
Beispiel 7: Radar-Diagramm



Beispiel 8: Blasendiagramm



Beispiel 9: Scatter-Plot



Q & R

Vielen Dank für Ihr Interesse!

Vielen Dank unseren Sponsoren

