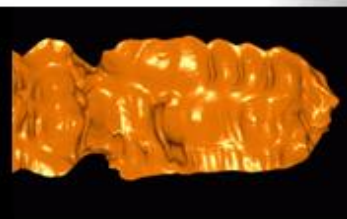
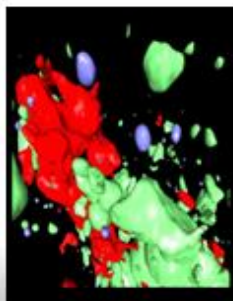
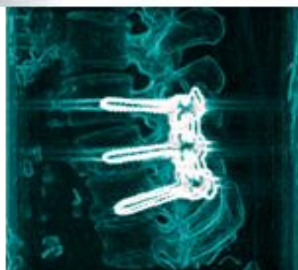
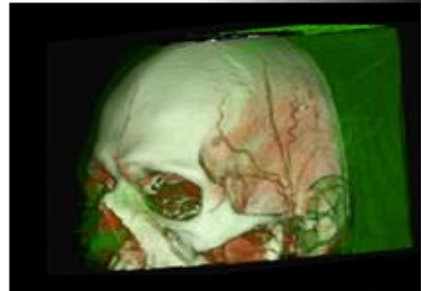
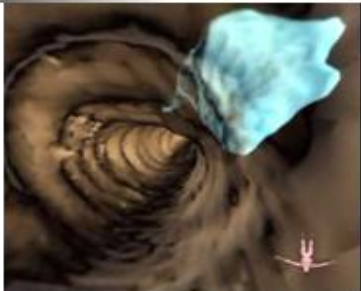
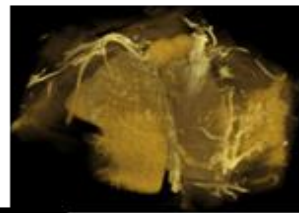
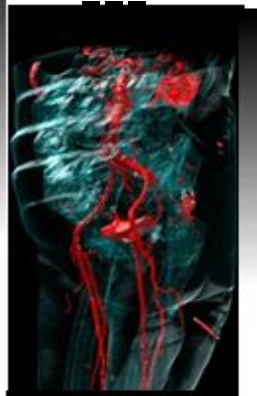
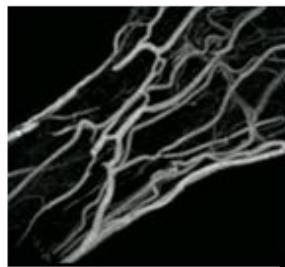


6170 UE Übung: Visualisierung 1

WiSe 22/23

Übung 2 - InfoVis

Aufgabenstellung



Aufgabe 2: Scatterplot und Radar Chart in d3

General information

In dieser Übung werden die klassischen InfoVis-Techniken Scatterplot und Radar Chart für multivariate Daten implementiert und über interaktive Selektionen verbunden. Wir verwenden die weit verbreitete **d3**-Bibliothek, um skalierbare Vektorgrafiken (**SVG**) auf einer Webseite zu erzeugen.

Für diese Übung werden Ihnen die folgenden Dateien zur Verfügung gestellt:

- (1) index.html
- (2) style.css
- (3) vis1.js
- (4) 3 csv Datasets

In 1-3 befindet sich der notwendige HTML- und JavaScript-Code, um leere Achsen für den Scatterplot und das Radar Chart mithilfe der JavaScript-Bibliothek d3 zu erstellen. Für das Menü wird **jQuery UI** verwendet.

Bitte beachten Sie die verwendeten Versionen:

Library	Version
D3.js	7.x
jQuery	3.x
jQuery UI	1.13.x

Bezüglich d3 ist zu beachten, dass es viele Online-Tutorials gibt, die auf einer älteren Version von d3 basieren und teilweise nicht mehr kompatibel sind! Es wird empfohlen, die offizielle API-Referenz von d3 Version 7 zu verwenden: <https://github.com/d3/d3/blob/main/API.md>.

Die Entwicklungsumgebung, die Sie verwenden, ist Ihnen überlassen, wir empfehlen z.B. Visual Studio IDE 2022 oder JetBrains WebStorm.

Informationen zur Einreichung

Für diese Aufgabe sind **insgesamt 26 Punkte** zu erreichen (die Mindestanforderung ist mit einem * markiert):

- Daten laden (3) *
- Achsen adaptieren (2) *
- Scatterplot Punkte rendern (8) *
- Scatterplot Selektion (4 + 1 Bonuspunkt)
- Radar Chart (5)
- Animierte Übergänge (1)

- Eigenen Datensatz visualisieren (2)

Mindestanforderung: Die Scatterplot-Darstellung mit korrekten Achsen, korrekt skalierten und positionierten Punkten.

Ihre Abgabe sollte die folgenden Dateien enthalten:

- index.html (Insert your names here in the place marked TODO)
- style.css
- vis1.js
- data.csv (your own dataset)
- screenshot.png

Aufgaben

Daten laden (3 Punkte)

Die Daten können über einen FileUpload-Button geladen werden. Die Funktionalität zum Parsen der Datei ist bereits implementiert.

Select your data here: cars.csv

Für diese Übung können Daten in folgendem Format geladen werden:

Name/ID	Numeric attribute	Numeric attribute	...
Label	1	2	...
...

In csv-Dateien stehen die Titel der Spalten in der ersten Zeile. Wir stellen 3 weit verbreitete Datensätze zur Verfügung:

- **cars.csv:** Automarken mit Attributen wie Baujahr, Beschleunigung, Verbrauch....
- **flowers.csv:** ("Iris dataset"): 3 Arten von Iris-Pflanzen mit Längen der Blütenblätter und Kelchblätter
- **nutritients.csv:** Lebensmittel mit Eigenschaften wie Energie, Eiweiß, Zucker (Achtung: viele Daten!)

Die Daten müssen so aufbereitet werden, dass sie für die Visualisierung verwendet werden können. Ein guter Weg, dies zu tun, ist die Verwendung der d3-dsv-Funktionalität. Das Ergebnis sollte ein Array von JavaScript-Objekten sein, wie hier für den Datensatz "Autos":

```

▼ Array(406) [ { }, { }, { }, { }, { }, { }, { }, { }, { }, { }, ... ]
  ▼ [0_99]
    0: Object { name: "AMC Ambassador Brougham", "economy (mpg)": 13, cylinders: 8, ... }
      "0-60 mph (s)": 11
      cylinders: 8
      "displacement (cc)": 360
      "economy (mpg)": 13
      name: "AMC Ambassador Brougham"
      "power (hp)": 175
      "weight (lb)": 3821
      year: 73
      <prototype>: Object { ... }
    1: Object { name: "AMC Ambassador DPL", "economy (mpg)": 15, cylinders: 8, ... }
    2: Object { name: "AMC Ambassador SST", "economy (mpg)": 17, cylinders: 8, ... }
    3: Object { name: "AMC Concord DL 6", "economy (mpg)": 20.2, cylinders: 6, ... }
    4: Object { name: "AMC Concord DL", "economy (mpg)": 18.1, cylinders: 6, ... }
    5: Object { name: "AMC Concord DL", "economy (mpg)": 23, cylinders: 4, ... }

```

Die Namen der numerischen Attribute, die visualisiert werden sollen, sollten in dem vorhandenen Array "dimensions" gespeichert werden:

```

▼ Array(7) [ "economy (mpg)", "cylinders", "displacement (cc)", "power (hp)", "weight (lb)", "0-60 mph (s)", "year" ]
  0: "economy (mpg)"
  1: "cylinders"
  2: "displacement (cc)"
  3: "power (hp)"
  4: "weight (lb)"
  5: "0-60 mph (s)"
  6: "year"
  length: 7
  <prototype>: Array []

```

Achsen anpassen (2 Punkte)

Beschriften Sie alle Achsen des Scatterplots und des Radar Charts mit den richtigen Attributen. Die Attribute, die im Scatterplot angezeigt werden, werden über das Menü ausgewählt. Im Radar Chart sollten immer alle numerischen Attribute angezeigt werden. Zeichnen Sie im Radar Chart auch die **Gitterlinien** (hellgraue Linien in einem Kreis um die Achsen).

Skalieren Sie alle Achsen entsprechend den anzuzeigenden Attributen.

Abbildung 1 zeigt als Beispiel das Ergebnis für den Datensatz "Autos" (mit den im Scatterplot ausgewählten Achsen für Wirtschaft und Leistung):

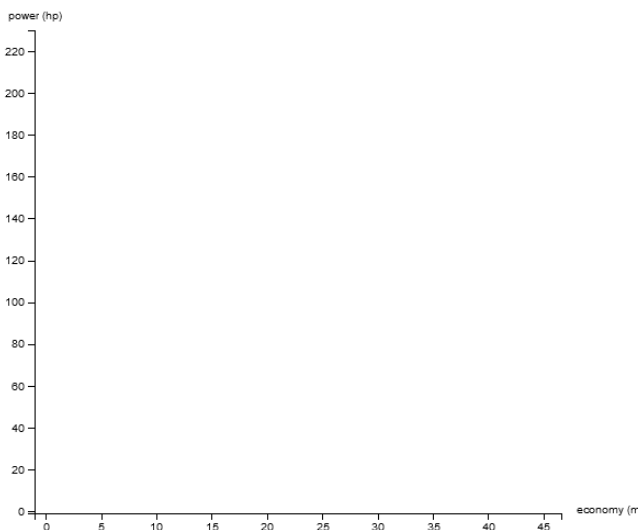
Scatterplot

Select dimensions to be displayed here:

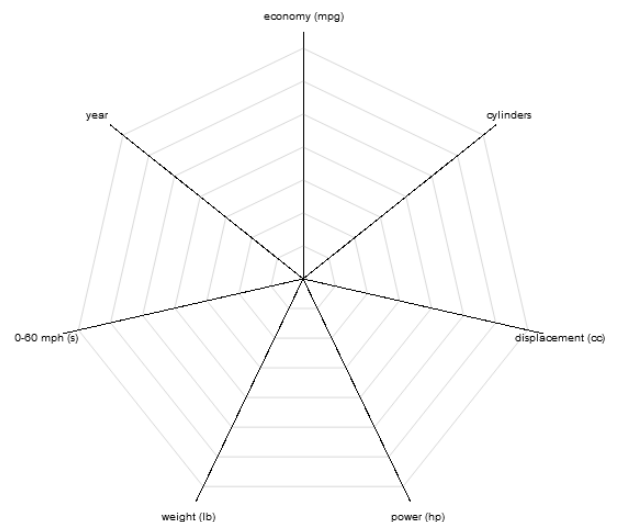
x axis:

y axis:

size:



Radar Chart



Legend:

Abbildung 1

Rendering von Scatterplot Punkten (8 Punkte)

Rendern Sie für jedes Element (z. B. die Linie in der Eingabedatei) die Position und Größe der Punkte entsprechend den drei im Menü ausgewählten Attributen. Stellen Sie sicher, dass die visuellen Kanäle korrekt angepasst werden, wenn Sie die Achsen erneut auswählen.

Hinweis: Die Farbe, Transparenz usw. ist Ihnen überlassen. Experimentieren Sie ruhig mit Transparenzen, Schatten oder anderen Effekten.

Abbildung 2 zeigt das Ergebnis für den Datensatz "Autos" (mit den im Scatterplot ausgewählten Achsen für Wirtschaft und Gewicht):

Scatterplot

Select dimensions to be displayed here:

x axis:

y axis:

size:

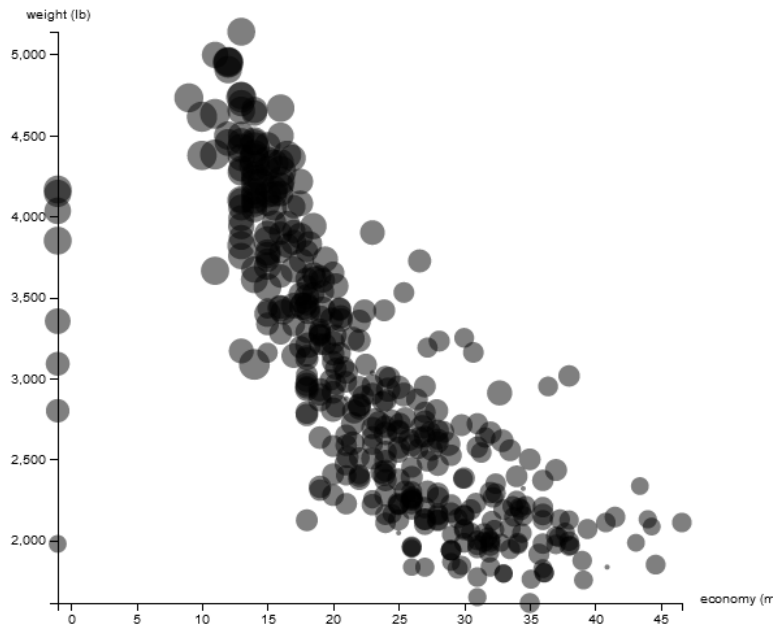


Abbildung 2

Scatterplot Selektion (4 Punkte)

Wenn Sie auf einen Punkt im Scatterplot klicken, werden die Punkte ausgewählt und markiert. Jeder ausgewählte Punkt erhält eine eigene Farbe und wird in der Legende unterhalb des Radar Charts angezeigt.

Die Legende sollte den Namen oder die ID (d. h. das erste, nicht numerische Attribut im Datensatz) des ausgewählten Punktes zusammen mit der zugewiesenen Farbe anzeigen. Die Art der Anzeige ist Ihnen überlassen.

Wichtig ist, dass die Legende gut lesbar ist, dass keine Farben doppelt vorkommen und dass Sie nicht mehr als eine vorgegebene Anzahl von Elementen auswählen können (5 bis 10). Bei der Auswahl der Farben ist es wichtig, dass Sie ein Farbschema wählen, damit die einzelnen Kategorien leicht unterschieden werden können.

In Abbildung 3 sehen Sie das Ergebnis für den Datensatz "Autos" mit 5 ausgewählten Elementen:

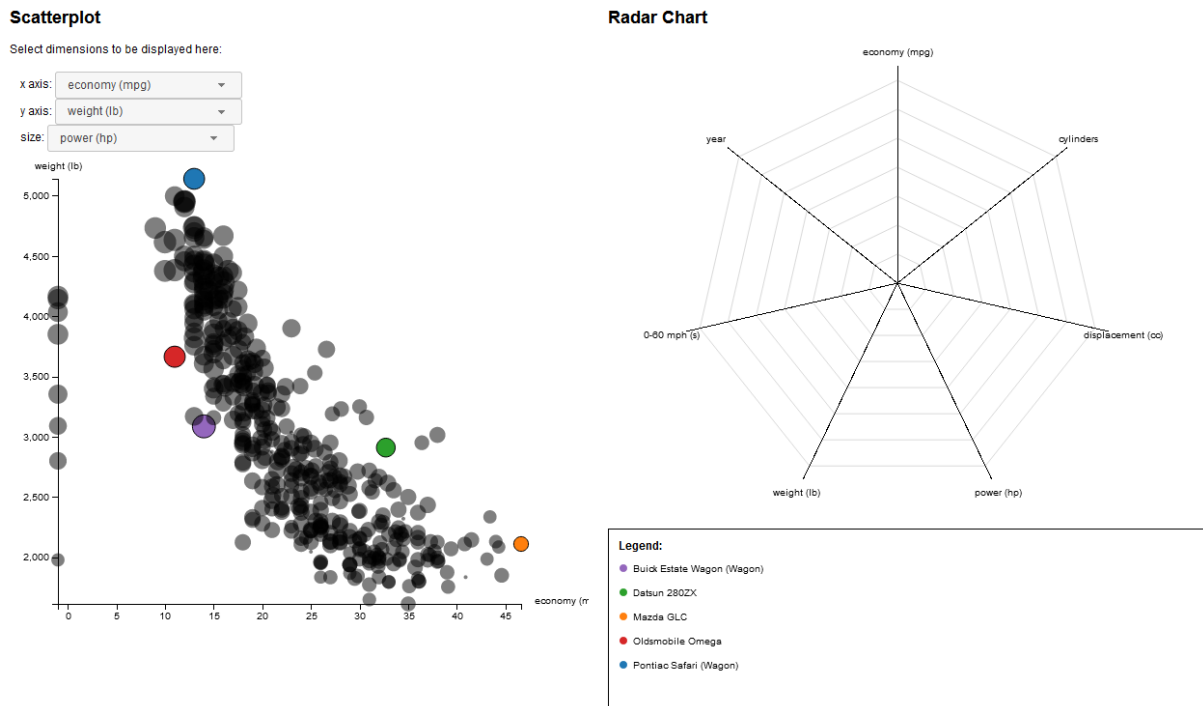


Abbildung 3

Bonus Deselektion (1 Punkt)

Fügen Sie die Möglichkeit, Elemente auch über die Legende abzuwählen hinzu (d.h. das Element wird im Radar Chart nicht mehr angezeigt und die farbliche Hervorhebung im Scatterplot wird entfernt). Bitte beachten Sie auch die bereits vorhandenen CSS-Stile!

Legend:	
● Datsun 280ZX	<input type="checkbox"/>
● Mazda GLC	<input type="checkbox"/>
● Oldsmobile Omega	<input type="checkbox"/>

Radar Chart (5 Punkte)

Der Scatterplot sollte mit dem Radar Chart verlinkt sein, d.h. die ausgewählten Punkte im Scatterplot sollten **als Linien im Radar Chart angezeigt** werden.

Stellen Sie sicher, dass die Farben konsistent zugewiesen werden (d. h. dieselbe Farbe für das ausgewählte Element im Scatterplot und die entsprechende Linie im Radar Chart) und passen Sie die Linien korrekt an oder löschen Sie sie, wenn der Benutzer eine neue Auswahl im Scatterplot trifft.

Abbildung 4 zeigt das Ergebnis für den Datensatz "Autos" mit 5 ausgewählten Elementen:

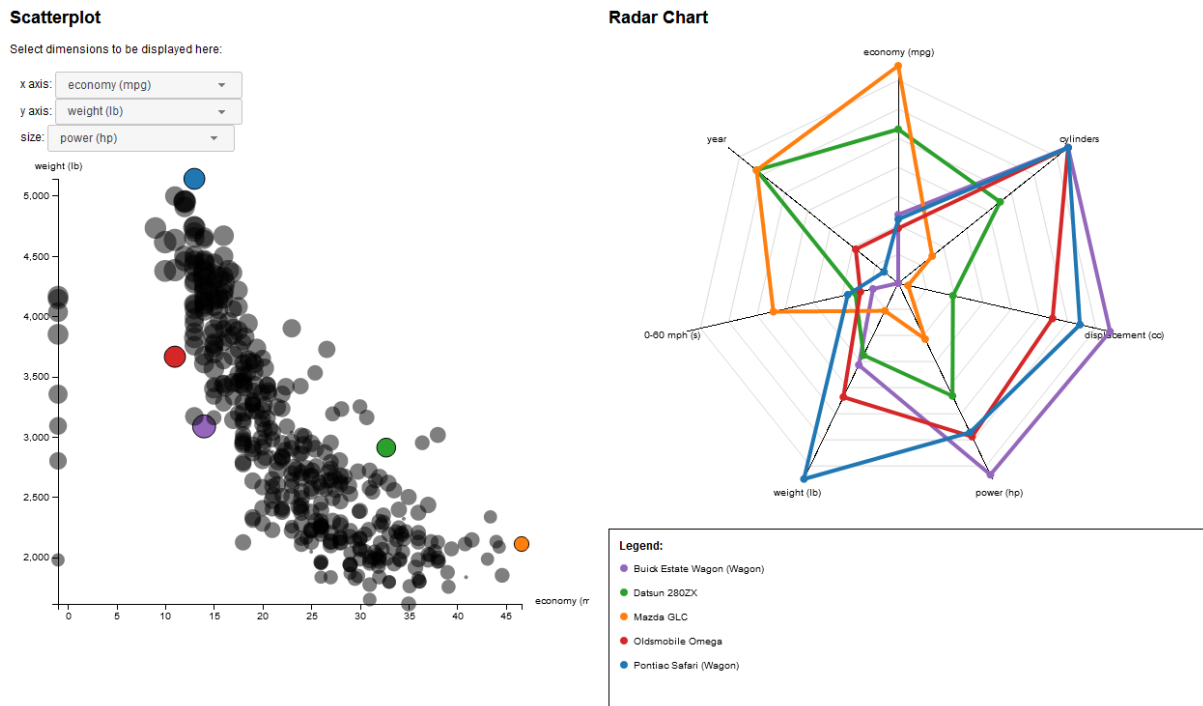


Abbildung 4

Animierte Übergänge im Scatterplot (1 Punkt)

D3 bietet Möglichkeiten, **Übergänge zu animieren**, wenn sich die Daten ändern. Wenn der Benutzer die Zuordnung der Attribute für den Scatterplot über das Menü ändert, sollte die Änderung der Punkte (d. h. die Position, die Farbe, die Größe) animiert werden.

Laden, visualisieren und erklären von eigenem Datensatz (2 Punkte)

Laden Sie einen Datensatz Ihrer Wahl, der nicht von uns bereitgestellt wurde (siehe Aufgabe "Daten laden") und visualisieren Sie die Daten.

Legen Sie der Abgabe Ihren Datensatz und einen Screenshot der Visualisierungen bei. Während des Abgabegesprächs sollten Sie in der Lage sein, die Daten anhand der Visualisierungen zu erklären.

Mögliche Datensätze finden Sie zum Beispiel hier:

- <https://archive.ics.uci.edu/ml/datasets.php> (UCI Machine Learning Repository)
- <https://www.kaggle.com/datasets> Kaggle Datasets)

Wichtiger Hinweis:

Bitte vergewissern Sie sich, dass in Ihrem Datensatz keine fehlenden Werte vorhanden sind,

da diese in dem gegebenen Framework nicht behandelt werden. Wenn Ihre Daten fehlende Werte enthalten, müssen Sie den Code anpassen, um diese Fälle zu erfassen, oder die fehlenden Werte in einem Vorverarbeitungsschritt hinzuzufügen. Informationen über die Imputation von Daten finden Sie hier: <https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779>

Referenzen:

- D3js.org mit vielen Beispielen (Achtung: viele verwenden eine ältere API-Version!)
<https://d3js.org/>
- D3 API Referenz:
<https://github.com/d3/d3/blob/master/API.md>
- Weitere d3-Beispiele von Mike Bostock, Autor von d3:
<https://bl.ocks.org/mbostock>
- D3 tutorial:
<http://square.github.io/intro-to-d3/>
- JQuery UI user interface:
<https://jqueryui.com/>
- Parsing von Daten mit d3:
<https://github.com/d3/d3-dsv>