MASTER-PRAKTIKUM
“DATA ANALYSIS WITH PYTHON”
SHOWING RELATIONSHIPS
ARC DIAGRAM

• **Type**: one-dimensional network diagram

• **Characteristics:**
  • Nodes are placed along a one-dimensional axis
  • Arcs show connections between the nodes \(\rightarrow\) networks
  • The thickness of each arc line can be used to represent the intensity of the relationship or frequency between the source and target node
ARC DIAGRAM

• **Usage:** finding the co-occurrence within data

• **Tools to generate visualisation:**
  • Protovis (code)
  • D3 (code)

• **Criticism:**
  • Don't show structure and connections between nodes as well as two-dimensional charts would do
  • Too many links can make the diagram confusing
Network representation of character co-occurrence in Victor Hugo’s classic novel *Les Misérables*

Source: [http://mbostock.github.io/protovis/ex/arc-full.html](http://mbostock.github.io/protovis/ex/arc-full.html)

Original data set: The Stanford GraphBase
RADIAL BAR CHART

- **Type**: Bar chart plotted on a polar coordinate system instead of a Cartesian system

- **Characteristics**:
  - One bar per category
  - Bar angles indicate the value amount
  - Values are radially arranged around the outermost bar
RADIAL BAR CHART

- **Usage:** visualization for comparing a small amount of data and placing them in a relationship

- **Tools to generate visualization:**
  - AnyChart

- **Criticism:** Risk of misinterpretation
  - Different lengths of bars representing the same value → different radii: (the outside one gets relatively longer to the last)
  - The Cartesian bar chart is a better choice for comparing values. Therefore, Radial Bar Charts are used primarily for aesthetic reasons.
Percentage of graduates finding employment: Comparisons of three institutes

Source: http://www.datarevelations.com/tag/radial-bar-chart
Tree diagrams are often used to show hierarchical structured relationships, such as:

- family relationships
- organizational relationships within companies
- relationships between species

![Tree Diagrams]

**TREE DIAGRAM**
Use `pip install pydot` to install the package

```python
# pydot is an interface to Graphviz
import pydot

# initialise the graph
graph = pydot.Dot(graph_type='graph')

# pydot.Edge() creates an Edge
# graph.add_edge() adds edges to the graph
graph.add_edge(pydot.Edge('root', 'node 1'))
graph.add_edge(pydot.Edge('root', 'node 2'))

# pydot.Node() creates a node
# graph.add_node() adds nodes to the graph
graph.add_node(pydot.Node('Node A'))
graph.add_node(pydot.Node('Node B', style='filled', fillcolor='green'))

# graph.write_pdf() created pdf file containing the tree
graph.write_pdf('tree_2.pdf')
```
Parallel coordinates plots are often used for plotting multivariate, numerical data. They are ideal for comparing objects with the same attributes, such as:

- car specs
- computer specs
PARALLEL COORDINATES PLOT

Characteristics:

- Each variable has its own axis
- Axis are placed in parallel
- Each axis can have its own scale and unit of measurement

Important:

- The order of the axis can have a greater impact on how the reader understands the data

Problem:

- If there is no correlations exist across the variables the diagram can become quite confusing
PARALLEL COORDINATES PLOT

Use `pip install plotly` to install the package

Docs: https://plot.ly/python/parallel-coordinates-plot/

```python
import plotly
import plotly.graph_objs as go

data = [
    # go.Parcoords initialises the graph
go.Parcoords(
        # line specifies the lines connecting the points on the axis
        line = dict(color = 'red'), # color sets line color to red
        # dimensions specifies the axis
        dimensions = list([
            dict(range = [0,1024], # range specifies the interval of the axis
                label = 'variable A', # label specifies the name of the axis
                values = [16,768]), # values specifies the points on this axis
            dict(range = [0,1],
                label = 'variable B',
                values = [0.2, 0.64])
        ])
    )
]
# plotly.offline.plot plots the graph offline
plotly.offline.plot(data, filename = 'parcoo Gard-dimensions')
```
FILLED CHORD DIAGRAM & NON-RIBBON CHORD DIAGRAM

Filled Chord Diagram

- Shows relationships between entities
- Compares similarities in dataset
- Connected with arcs or Bézier curves
- Values are assigned to each connection and shown by the size of the arc
- Color can be used to group data
- Danger: over-cluttering if too much data

Non-ribbon Chord Diagram

- Same as Filled Chord Diagram, but just lines
- Easier, if only relationship not „size“ is important
- Cleaner look, if more items are used

Ideal for paired numerical data
Points display values from two variables
Variables’ relationship or correlation:
- positive (both values increase), negative (one value decreases), null
- linear, exponential or U-shaped
Strength of correlation is shown by proximity of points, far away points are “outliers”

https://plot.ly/matplotlib/scatter/
import plotly.plotly as py
import plotly.graph_objs as go

# Create random data with numpy
import numpy as np
random_x = np.random.randn(1000)
random_y = np.random.randn(1000)

# Create a trace
trace = go.Scatter(x = random_x,
y = random_y, mode = 'markers')
data = [trace]

# Plot and embed in ipython notebook!
py.iplot(data, filename='basic-scatter')

random_x = np.linspace(0, 1, 100)
random_y0 = np.random.randn(100)+5
random_y1 = np.random.randn(100)
random_y2 = np.random.randn(100)-5

# Create traces
trace0 = go.Scatter(x = random_x, y = random_y0,
                   mode = 'markers', name = 'markers')
trace1 / trace2 / trace3 = go.Scatter(
   ... mode = 'lines+markers' ... 'lines')
data = [trace0, trace1, trace2]
py.iplot(data, filename='scatter-mode')

DESCRIPTION

• Also known as Network Graph, Network Map, Node-Link-Diagram

• Shows how things are interconnected through the use of nodes and link lines

• Helps to illuminate relationships between a group of entities
ANATOMY

- **Nodes**
  - → little dots, circles or icons

- **Links**
  - → simple lines between nodes

- **Additional variables** can be visualized
  - → node size, stroke width, color
TYPES

- **Undirected Network Diagrams** only display connections between entities.

- **Directed Network Diagrams** show if connections are one-way or two-way through small arrows.

![Undirected network diagram](image1.png)

![Directed network diagram](image2.png)

Undirected network  Directed network
LIMITATIONS

• Limited data capacity

• Become hard to read when there are too many nodes
Tools to generate visualization:

http://www.datavizcatalogue.com/methods/network_diagram.html

Network Diagram in Python:

VENN DIAGRAM

**Vegetables**
- Lettuce
- Potato
- Carrot
- Artichoke
- Cauliflower
- Leak
- Lentil
- Onion
- Asparagus
- Watercress
- Celery
- Broccoli
- Spinach
- Cabbage
- Chives
- Rocket

**Fruit**
- Blackberry
- Raspberry
- Pineapple
- Blueberry
- Cherry
- Apple
- Plumbs
- Apricot
- Banana
- Pear
- Strawberry
- Grapefruit
- Orange
- Grapes
- Lemon
- Kiwi
- Fig
- Lime
DESCRIPTION

- Also known as Set Diagram
- Visually displays all possible logical relations between a collection of sets
ANATOMY

- **Set**
  - → represented as a circle
  - → collection of objects or array of data that all have something in common

- **Intersection**
  - → overlapping of multiple circles
  - → data that has all the qualities of the overlapping sets
TYPES

- 2-set Venn Diagrams
- 3-set Venn Diagrams
- ...  
- 7-set Venn Diagrams
LIMITATIONS

- Limited number of sets
- Limited number of set objects
Tools to generate visualization:

http://www.datavizcatalogue.com/methods/venn_diagram.html

Venn Diagram in Python:

https://pypi.python.org/pypi/matplotlib-venn

https://pypi.python.org/pypi/matplotlib_venn_wordcloud
VENN DIAGRAM IN PYTHON

```python
v = venn2_wordcloud([set_A, set_R], ('Armut', 'Reichtum'))
```
BRAINSTORM

Related Subcategory  
Category A

Related Subcategory  
Category B

Related Subcategory  
Category C

Related Subcategory  
Category D
BRAINSTORM

Online-Tools:
http://coggle.it/
http://www.mindmup.com/
BRAINSTORM: EXAMPLE

Art of the Long 19th Century

Depiction of Wealth
- Symbols of Power
- Precious Clothing
- Rich Interior
- Bourgeoisie
- Rulers
- People

Depiction of Poverty
- Housing Conditions
- Barracks
- Evictions
- Trades/Professions
- Suffering in Work
- Hunger
- Health Conditions
- Illness
- Death
- Heroisation/Revolution
- People
- Sorrow/Misery
FLOW MAP

Map / Geographic Region

Thickness = Amount

Origin

Flow Line
FLOW MAP: EXAMPLE

Top Chinese Export Directions to the World
According to www.worldsrichestcountries.com

Percent of Chinese Overall Exports
- Less than 0.5%
- 0.5 - 1.5%
- 1.5 - 3%
- 3 - 5%
- 5 - 10%
- More than 10%
CONNECTION MAP: EXAMPLE

Charting Culture

http://www.nature.com/nature/videoarchive/charting-culture/index.html
FURTHER LINKS / USE IN PYTHON

Networks on Maps
http://www.sociology-hacks.org/?p=67

Graph on a Basemap
https://stackoverflow.com/questions/19915266/drawing-a-graph-with-networkx-on-a-basemap

Alternative: mplleaflet library
https://github.com/jwass/mplleaflet