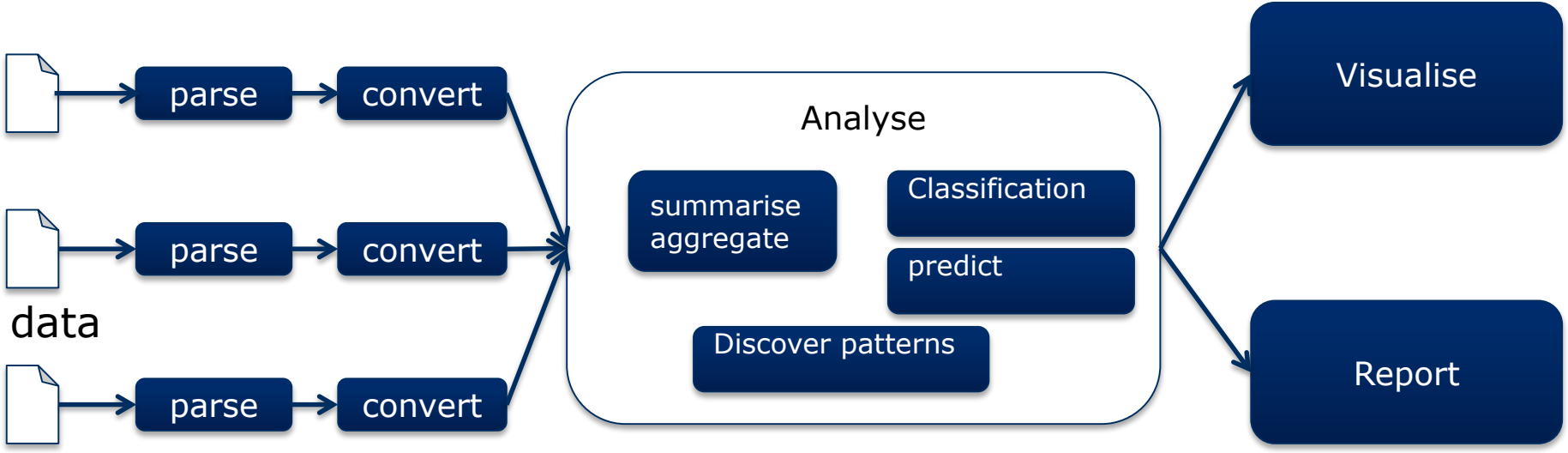
What can you do with the data

- Data processing
  - Provide/filter data
- Visualisations
  - charts, graphs, maps, …

From data to “meaning”

- Parsing, converting, analysing, representing



WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Parsing File Formats

Tabular, Tree, Graph, Text

How to parse the data?

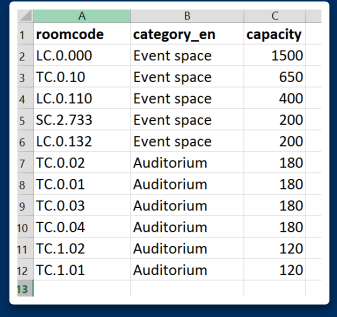

Lorem ipsum dolor si

Integer aliquam sem vitae ipsum vehicula e fringilla

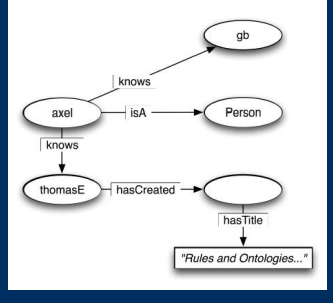
Donec et nisi lorem, sed rhoncus odio. Pellentesque est nulla, commodo id accumsan ac, volutpat quis ligula. Nulla libero felis, venenatis id varius in, vehicula eu lectus. Suspendisse porttitor odio in massa luctus viverra interdum eros hendrerit. porttitor volutpat quis .

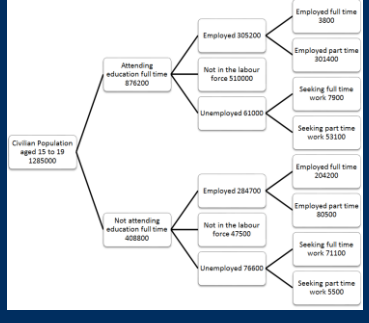
Mauris molestie consequat vulputate. Donec dignissim tempus suscipit. Sed tempor malesuada molestie. Etiam ullamcorper, orci vitae blandit malesuada, lacus orci consequat dolor, id adipiscing magna quam eu felis. Nunc sit amet turpis nisl. Cras nulla turpis, imperdiet non hendrerit vitae, ullamcorper varius ligula. Ut lacinia, risus sit amet sodales cursus, nc sit amet turpis nisl. Cras nulla turpis, imperdiet non hendrerit vitae, ullamcorper varius ligula. Ut lacinia, risus sit amet sodales cursus, sapien felis gravida nulla, ullamcorper dignissim turpis lacus sed nunc. Donec nisi sem, tincidunt eget aliquet sollicitudin, suscipit eu nulla. Suspendisse vitae risus lacus, eget euismod lectus tincidunt eget aliquet sollicitudin.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum id odio lorem, in bibendum erat. Integer tristique tincidunt aliquet. Suspendisse eget magna vitae.



	A	B	C
1	**roomcode**	**category_en**	**capacity**
2	LC.0.000	Event space	1500
3	TC.0.10	Event space	650
4	LC.0.110	Event space	400
5	SC.2.733	Event space	200
6	LC.0.132	Event space	200
7	TC.0.02	Auditorium	180
8	TC.0.01	Auditorium	180
9	TC.0.03	Auditorium	180
10	TC.0.04	Auditorium	180
11	TC.1.02	Auditorium	120
12	TC.1.01	Auditorium	120
13			





EFMD EQUIS ACCREDITED

Different Formats

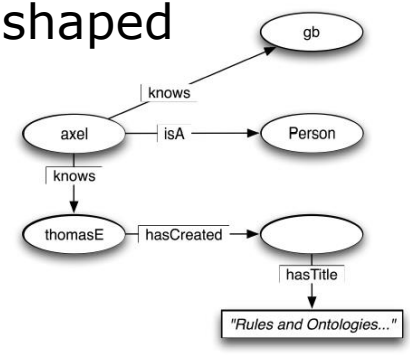
```
A N W E S E N D E .

VORSITZENDE:    Bürgermeister Mag. Siegfried NAGL
                Gemeinderätin Gerda GESEK

Weiters 48 Mitglieder des Gemeinderates, und zwar:

BERGMANN Ingeborg
BRAUNERSREUTHER Christine
DREISIEBNER Karl
EBER Manfred
FABISCH Andreas Mag.
FRÖLICH Klaus Mag.
```

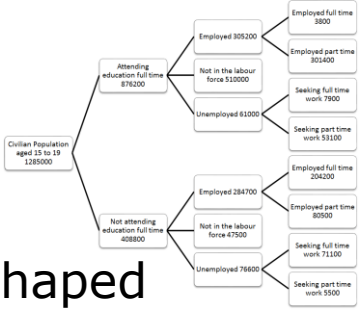
plain text

tabular

	A	B	C
1	**roomcode**	**category_en**	**capacity**
2	LC.0.000	Event space	1500
3	TC.0.10	Event space	650
4	LC.0.110	Event space	400
5	SC.2.733	Event space	200
6	LC.0.132	Event space	200
7	TC.0.02	Auditorium	180
8	TC.0.01	Auditorium	180
9	TC.0.03	Auditorium	180
10	TC.0.04	Auditorium	180
11	TC.1.02	Auditorium	120
12	TC.1.01	Auditorium	120
13			

graph-shaped





tree-shaped

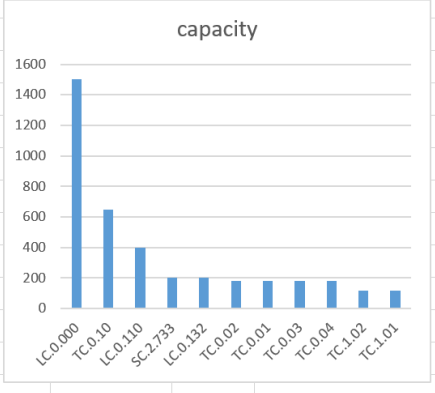
Tabular

- CSV
  - Comma-separated values
  - Plain text
  - Variations on the format

```
course_id,semester,name
4000,15S,Marketing
4001,15S,Marketing
4002,15S,"Personal, Führung, Organisation"
4004,15S,Grundlagen der Volkswirtschaftslehre
4006,15S,Mathematik
4007,15S,Statistik
```

- Excel
  - .xls, .xlsx, .xlsm, … (binary)
  - Widely used but proprietary
  - Built-in visualisation

	A	B	C
1	**roomcode**	**category_en**	**capacity**
2	LC.0.000	Event space	1500
3	TC.0.10	Event space	650
4	LC.0.110	Event space	400
5	SC.2.733	Event space	200
6	LC.0.132	Event space	200
7	TC.0.02	Auditorium	180
8	TC.0.01	Auditorium	180
9	TC.0.03	Auditorium	180
10	TC.0.04	Auditorium	180
11	TC.1.02	Auditorium	120
12	TC.1.01	Auditorium	120
13			



CSV

- Java: Commons CSV
- Python:
  - **CSV**: standard library
  - messytables: delimiter detection, datatype detection, …
- JavaScript:
  - Papa Parse: JSON conversion, delimiter detection

```
File csvData = new File("/path/to/csv");
CSVParser parser = CSVParser.parse(csvData, CSVFormat.RFC4180);
for (CSVRecord csvRecord : parser) {
    ...
}
```
Common CSV

```
var results = Papa.parse(csvString);
console.log(results.meta.delimiter);
```
Papa Parse

- CSVKit:
  - Command line tool, Python lib
  - grep, clean, processing

Tree-based: Hierachies

- XML
  - Markup language
  - Tag based (not predefined like HTML)
  - Human readable

```
<Gemeindedaten>
    <Gemeinde>
        <bgmname>Mag. Siegfried NAGL</bgmname>
        <email>stadtverwaltung@graz.at</email>
        <gemfax>+43 (316) 872-2369</gemfax>
        <gemname>Graz</gemname>
        <gemtel>+43 (316) 872-0</gemtel>
        <ort>Graz</ort>
        <plz>8010</plz>
        <strasse>Hauptplatz 1</strasse>
        <vbgmname>Mag.a Dr.in Martina SCHRÖCK</vbgmname>
        <webadr>http://www.graz.at</webadr>
    </Gemeinde>
</Gemeindedaten>
```

- JSON
  - **J**ava**S**cript **O**bject **N**otation
  - Identical to JavaScript objects
  - Name/value pairs
  - Easy-to-use alternative to XML

```
{
    id: "UNIVERSITAETOGD.762955",
    geometry: {
        type: "Point",
        coordinates: [
            16.408860446502068,
            48.21384341856702
        ]
    },
    properties: {
        NAME: "Wirtschaftsuniversität Wien",
        BEZEICHNUNG: "LC - Hauptgebäude"
    }
}
```

XML

- Java:
  - DOM parser
  - SAX parser: event based, no file size limits
- Python:
  - ElementTree: simple-to-use

```
import xml.etree.ElementTree as ET
tree = ET.parse('myfile.xml')
root = tree.getroot()

for child in root:
     print child.tag, child.attrib
```

ElementTree

- Command line:
  - xmllint: validation (against a DTD)

```
user@users-desktop:~$ xmllint --valid --noout gem.xml
gem.xml:1: validity error : Validation failed: no DTD found !
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><Gemeindedaten><Gemeinde>
                                  ^
user@users-desktop:~$
```

xmllint

KML

- Keyhole Markup Language
- XML-based schema for geographic data

- Default XML parser

- Java:
  - JAK
- Python:
  - pyKML: validation, KML construction

```java
final Kml kml = Kml.unmarshal(new File("..."));
final Placemark placemark = (Placemark) kml.getFeature();
Point point = (Point) placemark.getGeometry();
List<Coordinate> coordinates = point.getCoordinates();
for (Coordinate coordinate : coordinates) {
                System.out.println(coordinate.getLatitude());
                System.out.println(coordinate.getLongitude());
                System.out.println(coordinate.getAltitude());
}
```

JAK

JSON

- Java:
  - org.json: simple-to-use
  - Gson: serialize java classes, support for java generics

```java
String str = "{ \"name\": \"Alice\", \"age\": 20 }";
JSONObject obj = new JSONObject(str);
String n = obj.getString("name");
int a = obj.getInt("age");
System.out.println(n + " " + a);  // prints "Alice 20"
```
org.json

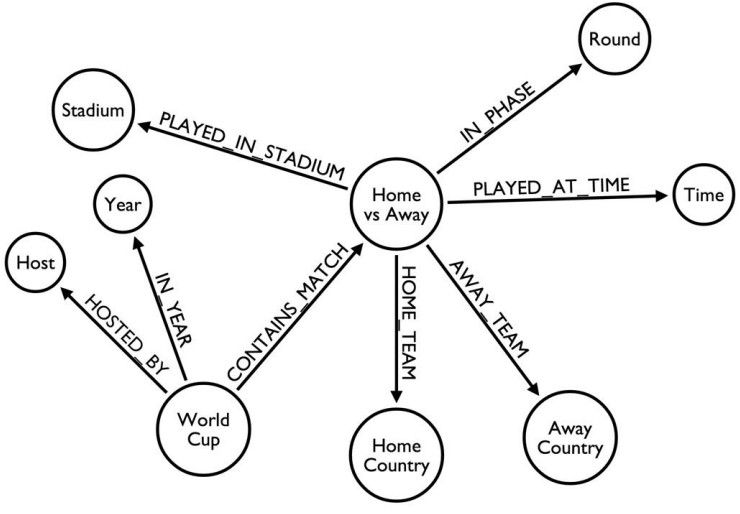
- Python:
  - json: standard library

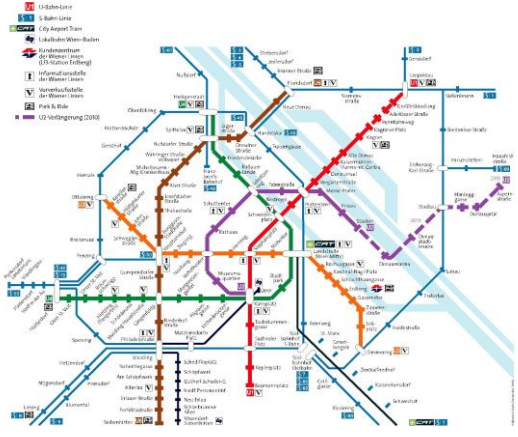
```python
import json
with open("...") as json_file:
    # json.load for file object arguments
    json_data = json.load(json_file)

# json.loads for string arguments
json_data = json.loads("{ \"name\": \"Alice\", \"age\": 20 }")
print json_data["name"] # prints Alice
```
Python

- Command line:
  - jq

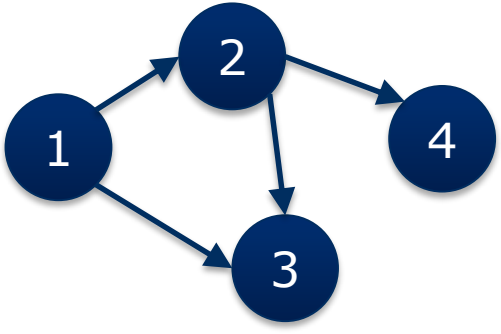
```
user@users-desktop:~$ curl http://data.wu.ac.at/api/rest/dataset/all_course_events_2015s |
| jq '.title'
"All Course-Events during SS15 at WU Vienna"
user@users-desktop:~$
```
jq

Graph-shaped data formats





Graph representation



- Edge-list
  1 2
  1 3
  2 3
  2 4

- Adjacency-list
  1: 2,3
  2: 3,4

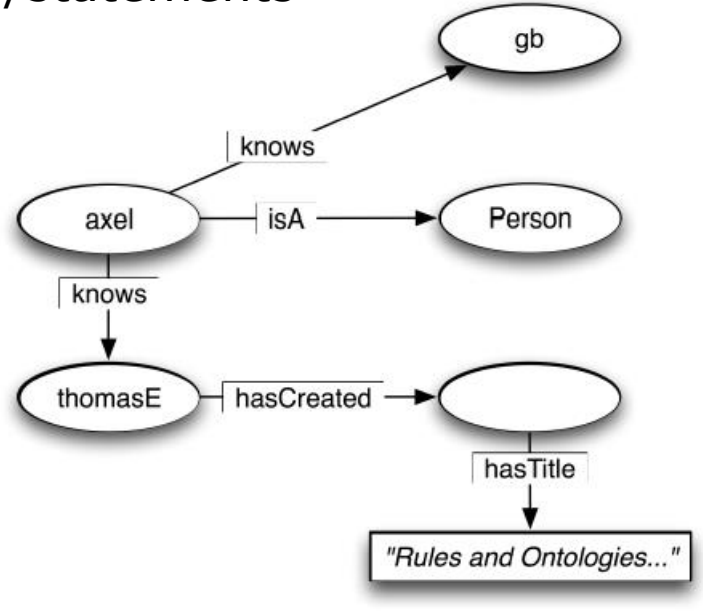
- Matrix

	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	0
4	0	1	0	0

Graph

- RDF (**R**esource **D**escription **F**ramework)
  - Describing resources per triples/statements
  - **S**ubject **P**redicate **O**bject

axel isA Person .
axel hasName “Axel Polleres”.
axel knows gb .
axel knows thomas.
thomas hasCreated an Article
titled “Rules and Ontologies …”.



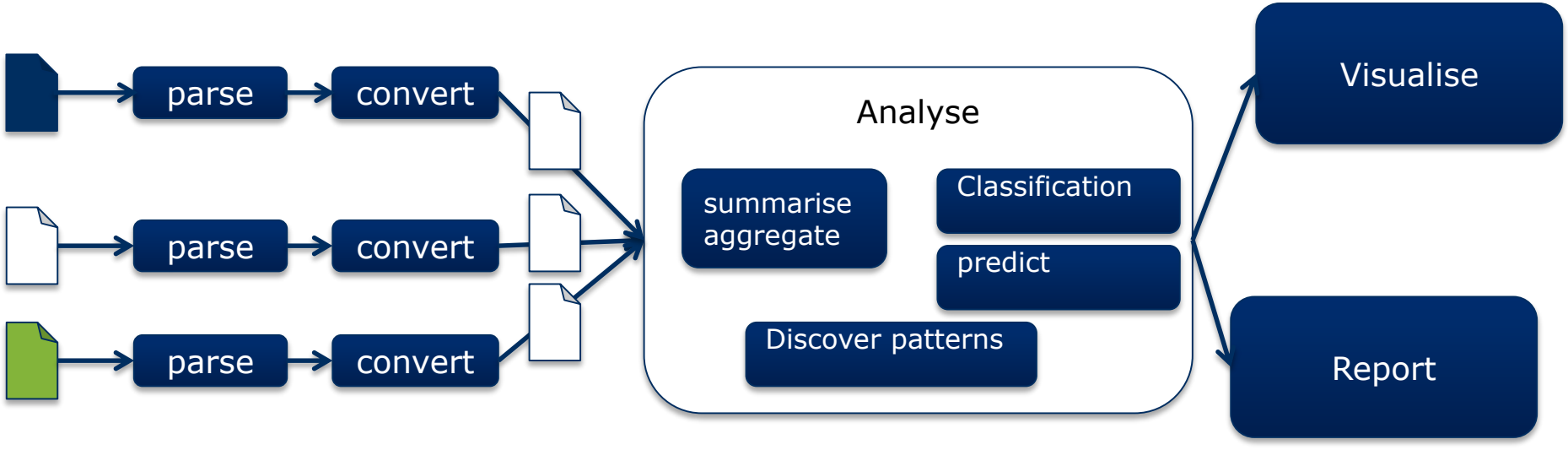
RDF

- Different syntaxes for RDF:
  - **RDF/XML**
  - Turtle
  - RDFa
  - JSON-LD
- Accessing RDF:
  - Query language: SPARQL
  - Parser

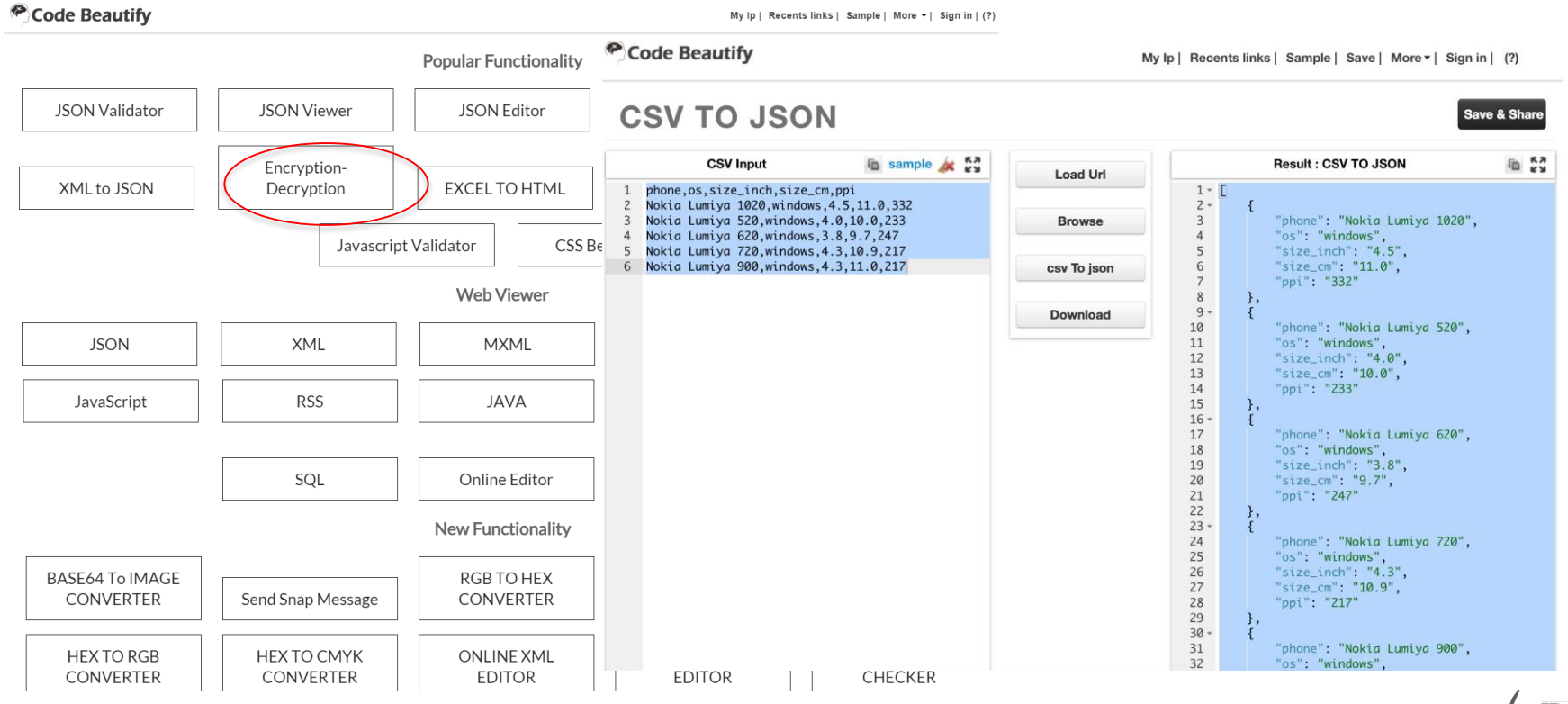
```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/" >
  <foaf:Person rdf:ID="me">
    <foaf:name>Axel Polleres</foaf:name>
    <foaf:title>Dr</foaf:title>
    <foaf:firstName>Axel</foaf:firstName>
    <foaf:lastName>Polleres</foaf:lastName>

    <foaf:knows>
      <foaf:Person>
        <foaf:name>Thomas Eiter</foaf:name>
      </foaf:Person>
    </foaf:knows>

 </foaf:Person>

</rdf:RDF>
```

RDF/XML

Converting


parse
convert
parse
convert
parse
convert
Analyse
summarise
aggregate
Classification
predict
Discover patterns
Visualise
Report

Code Beautify

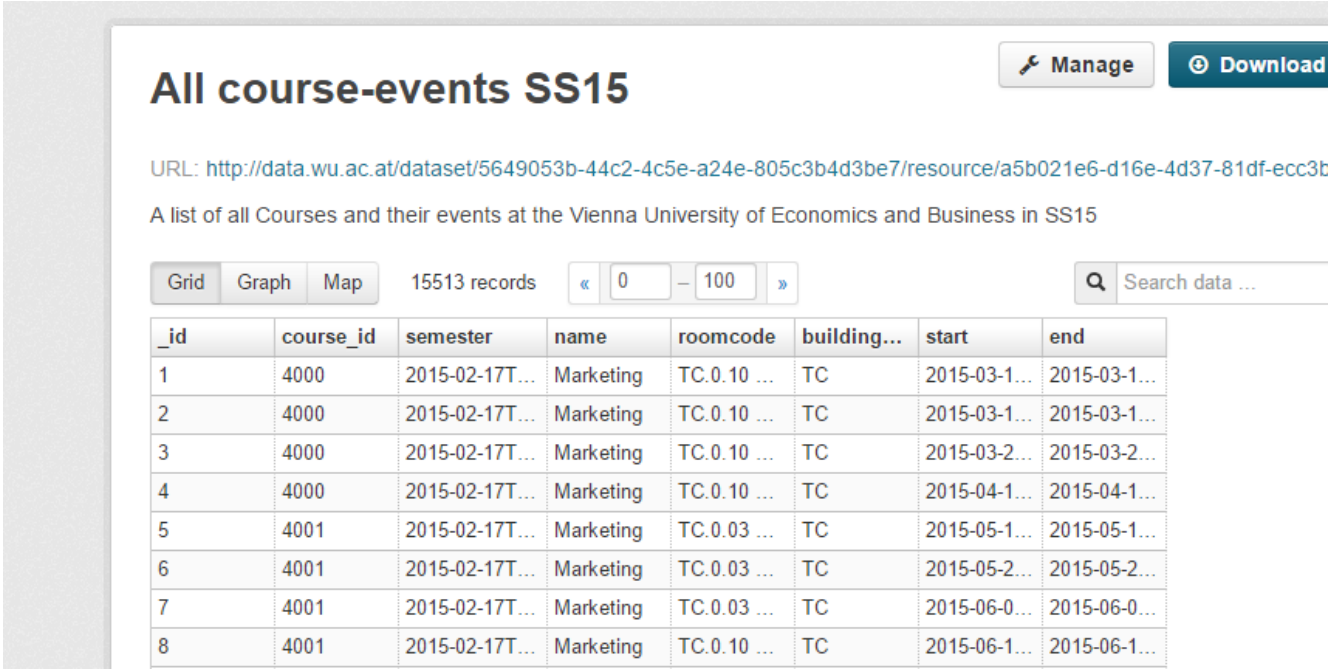


Data processing: Some Buzzwords

- Validation
- Converting
- Sorting
- Summarization
- Aggregation
- Analysis
- Reporting
- Classification
- …

E.g.: When to go to Mensa?

- Locate the data:



All course-events SS15

Manage | Download

URL: http://data.wu.ac.at/dataset/5649053b-44c2-4c5e-a24e-805c3b4d3be7/resource/a5b021e6-d16e-4d37-81df-ecc3b

A list of all Courses and their events at the Vienna University of Economics and Business in SS15

Grid | Graph | Map | 15513 records | « 0 – 100 » | Search data ...

_id	course_id	semester	name	roomcode	building...	start	end
1	4000	2015-02-17T...	Marketing	TC.0.10 ...	TC	2015-03-1...	2015-03-1...
2	4000	2015-02-17T...	Marketing	TC.0.10 ...	TC	2015-03-1...	2015-03-1...
3	4000	2015-02-17T...	Marketing	TC.0.10 ...	TC	2015-03-2...	2015-03-2...
4	4000	2015-02-17T...	Marketing	TC.0.10 ...	TC	2015-04-1...	2015-04-1...
5	4001	2015-02-17T...	Marketing	TC.0.03 ...	TC	2015-05-1...	2015-05-1...
6	4001	2015-02-17T...	Marketing	TC.0.03 ...	TC	2015-05-2...	2015-05-2...
7	4001	2015-02-17T...	Marketing	TC.0.03 ...	TC	2015-06-0...	2015-06-0...
8	4001	2015-02-17T...	Marketing	TC.0.10 ...	TC	2015-06-1...	2015-06-1...

E.g.: When to go to Mensa?

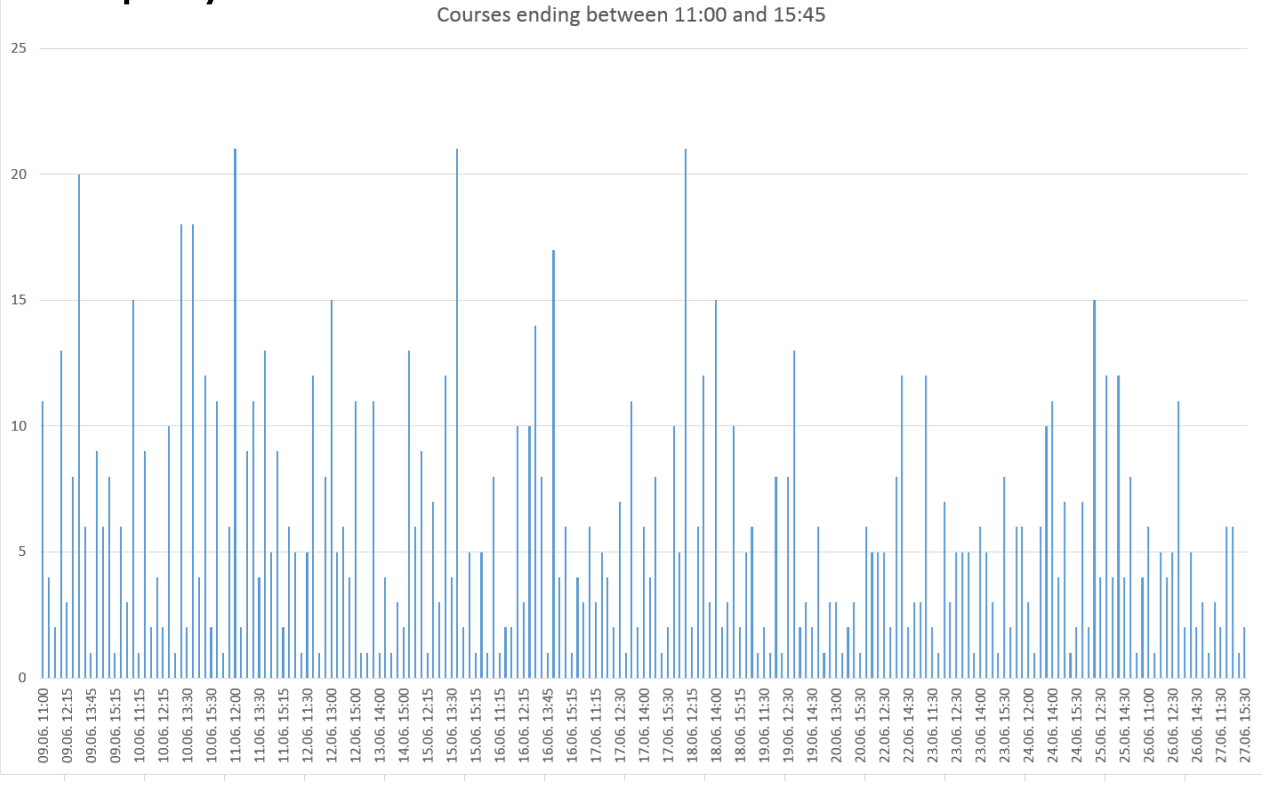
- Process and filter:

```
import csv
from dateutil.parser import parse
from collections import Counter
import datetime

f = open('allcoursesandevents15s.csv', 'r')
csvf = csv.reader(f)

ends = []
for row in csvf:
    ends.append(row[6])
ends = ends[1:]
dates = [parse(e) for e in ends]

counts = Counter(dates)
now = datetime.datetime.now() + datetime.timedelta(days=5)
next_week = now + datetime.timedelta(days=20)

week = []
for date in counts:
    if now < date < next_week:
        if 11 <= date.hour <= 15:
            week.append((date, counts[date]))
sorted_week = sorted(week)

outfile = open('this_week_course_end.csv', 'w')
for date, count in sorted_week:
    outfile.write(date.strftime("%d.%m. %H:%M") + ',' + str(count) + '\n')
```

E.g.: When to go to Mensa?

- Report and display:

09.06. 11:30	4
09.06. 11:45	2
09.06. 12:00	13
09.06. 12:15	3
09.06. 12:30	8
09.06. 13:00	20
09.06. 13:30	6
09.06. 13:45	1
09.06. 14:00	9
09.06. 14:30	6
09.06. 15:00	8
09.06. 15:15	1
09.06. 15:30	6
09.06. 15:45	3
10.06. 11:00	15
10.06. 11:15	1
10.06. 11:30	9
10.06. 11:45	2
10.06. 12:00	4
10.06. 12:15	2
10.06. 12:30	10
10.06. 12:45	1
10.06. 13:00	18
10.06. 13:30	2
10.06. 14:00	18
10.06. 14:30	4
10.06. 15:00	12
10.06. 15:30	2
11.06. 11:00	11
11.06. 11:15	1
11.06. 11:30	6
11.06. 12:00	21



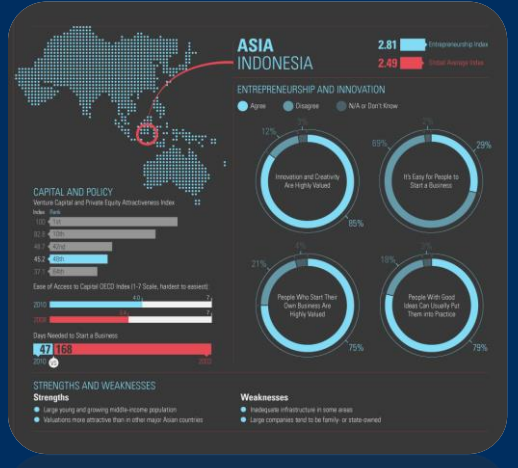
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Visualisation

tell a story with your data

appealing presentation

pictures tell more than thousand words

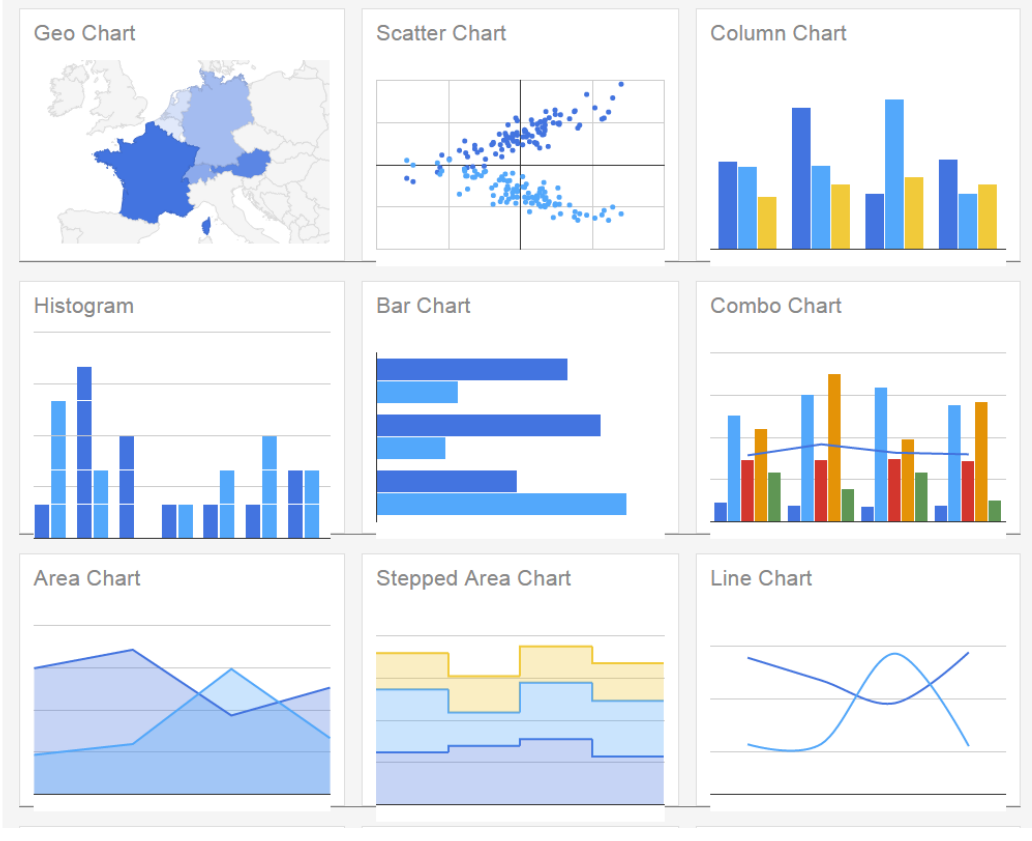


EFMD EQUIS ACCREDITED

Types of visualisation

- Charts
- Graphs
- Maps
- Infographics
- Interactive/complex Visualisations

Google Charts API



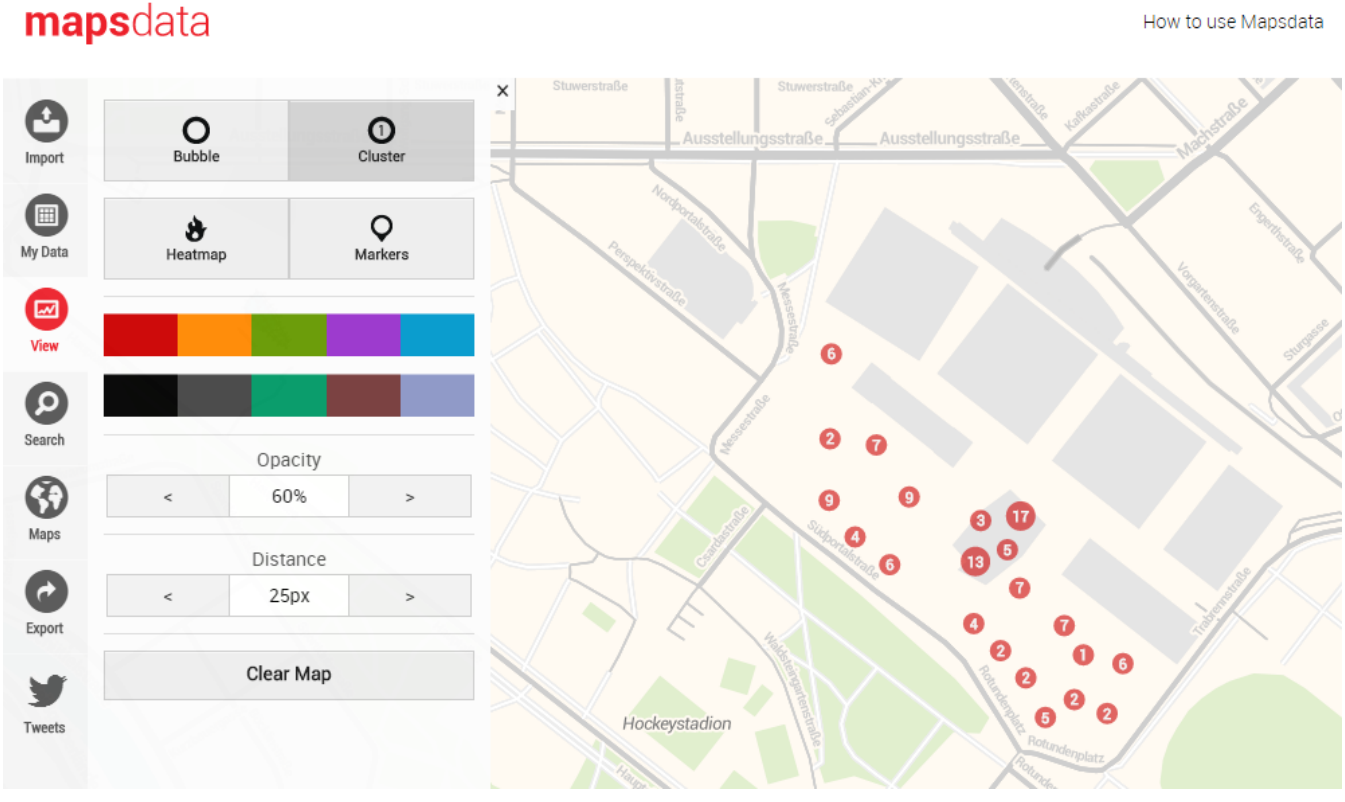
Maps

- mapsdata
  - Simple online tool
- `JavaScript:`
  - OpenLayers
  - Leaflet
- `Python:`
  - basemap: matplotlib plugin

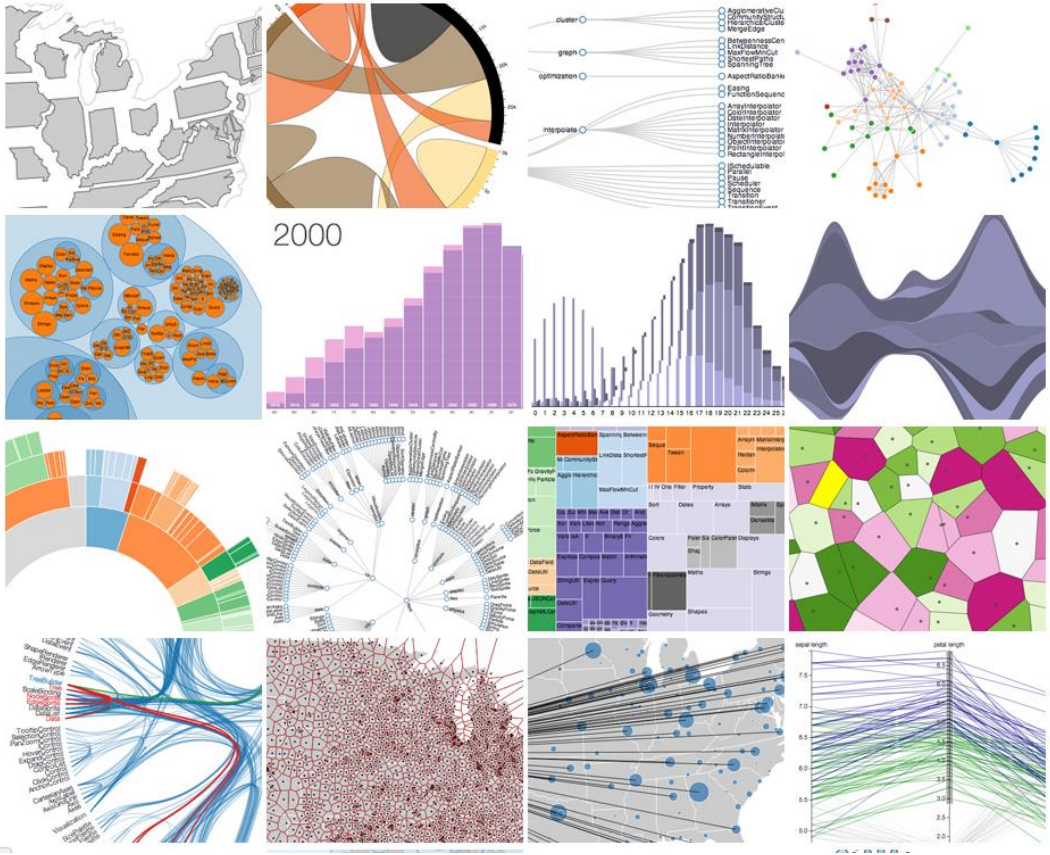
Google Maps API

- Google Maps API
  - API Clients for JS, Java, Python, .NET, …

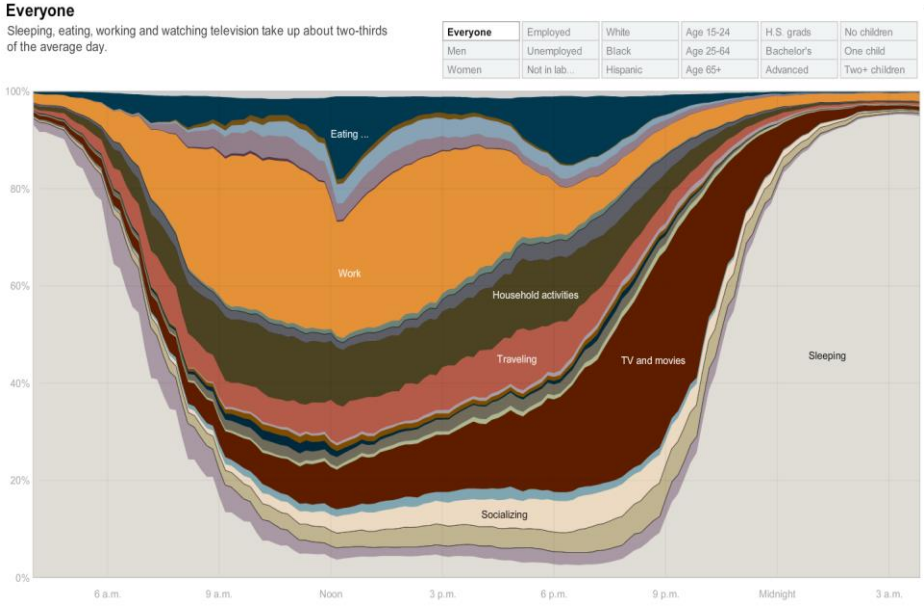
Lecture rooms on WU campus

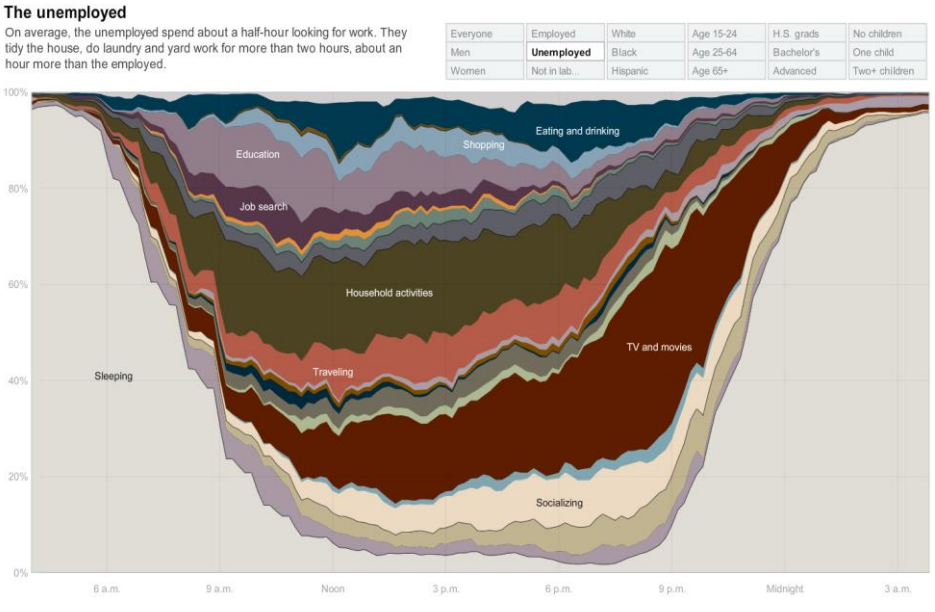


Graphs



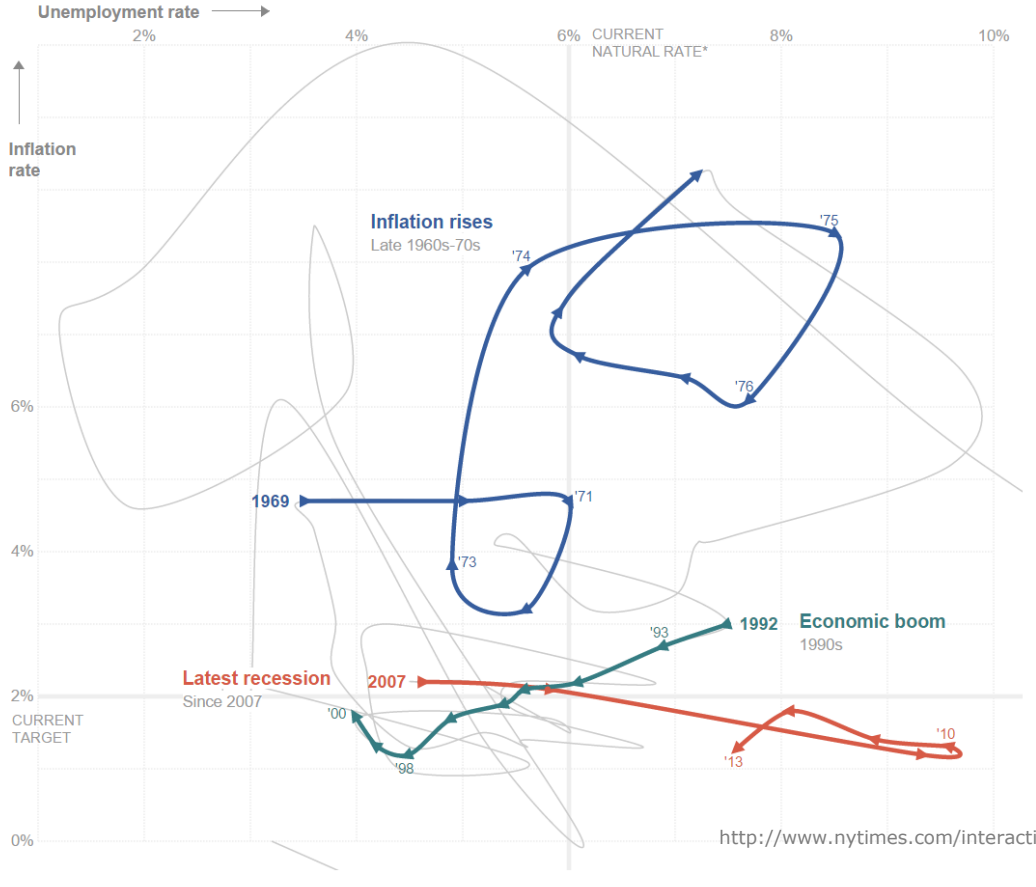
Visualising people's behaviour





http://www.nytimes.com/interactive/2009/07/31/business/20080801-metrics-graphic.html

D3 (Data-Driven Documents)



http://www.nytimes.com/interactive/2013/10/09/us/yellen-fed-chart.html

Libraries and IDEs

- `JavaScript:`
  - Flot: library for jQuery
  - Raphaël
  - D3 (complex visualisations)
- `Python:`
  - matplotlib
  - Seaborn
- Processing: JS and Python bindings
- R: statistical package

Further Links

- http://selection.datavisualization.ch/
- http://www.dmoz.org/Science/Math/Combinatorics/Software/Graph_Drawing/
- Java Open Source Chart libs:
  - http://www.fromdev.com/2012/09/Free-Open-Source-Java-Charting-Library.html
- http://socialcompare.com/en/comparison/javascript-graphs-and-charts-libraries