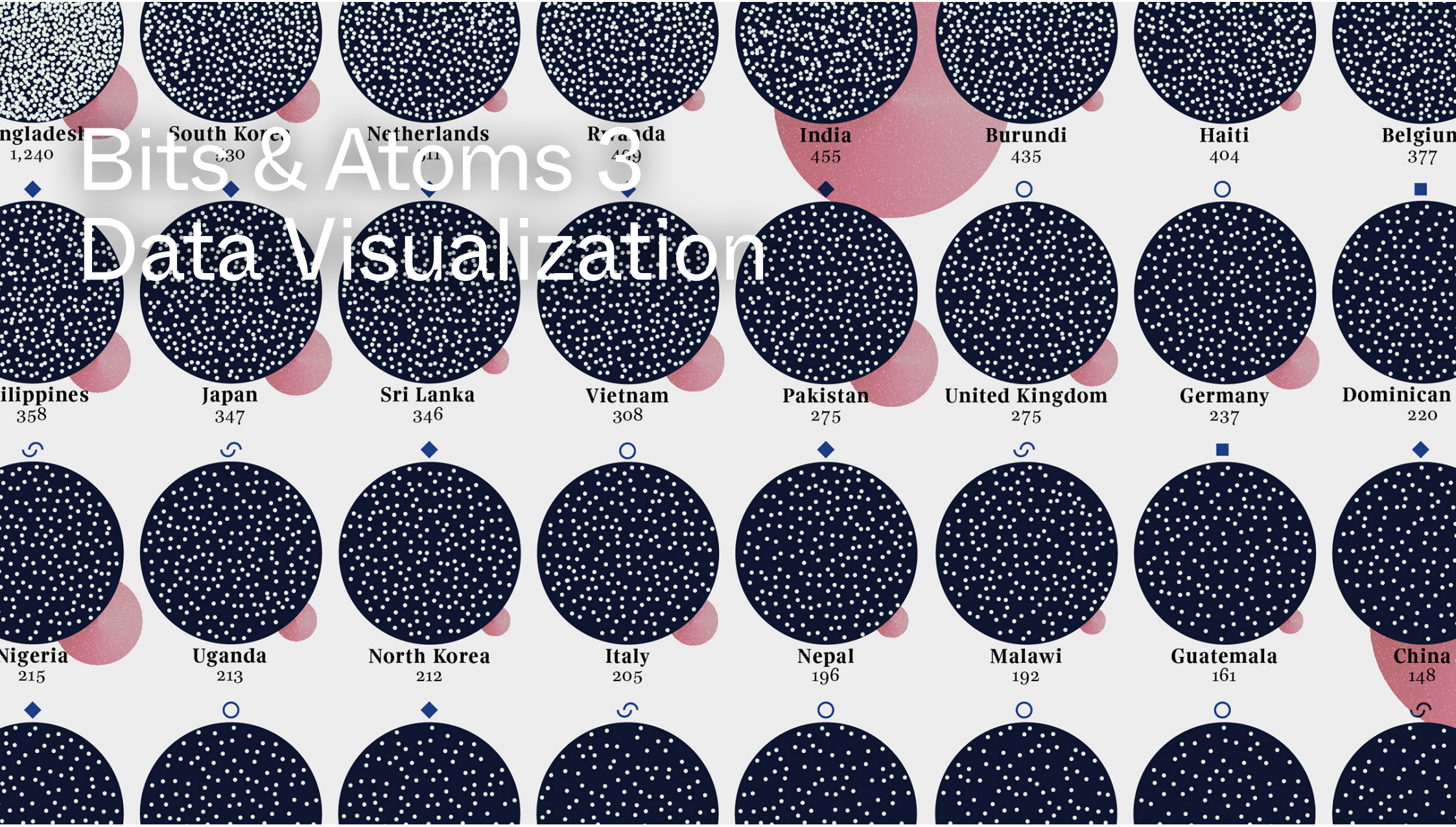


Bits & Atoms 3
Data Visualization

Course Overview

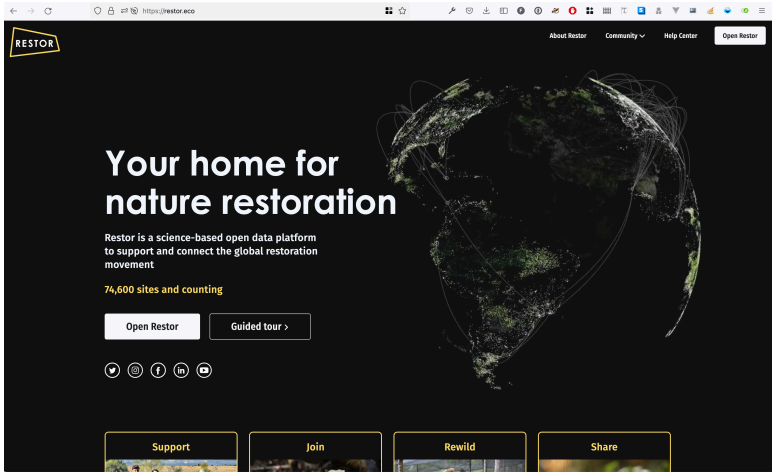
— Data visualization
— Regular attendence (80%)
— Grade: Exercises, Assignments, Participation

Crowther Lab—Restor

«Restor is a science-based open data platform to support and connect the global restoration movement»

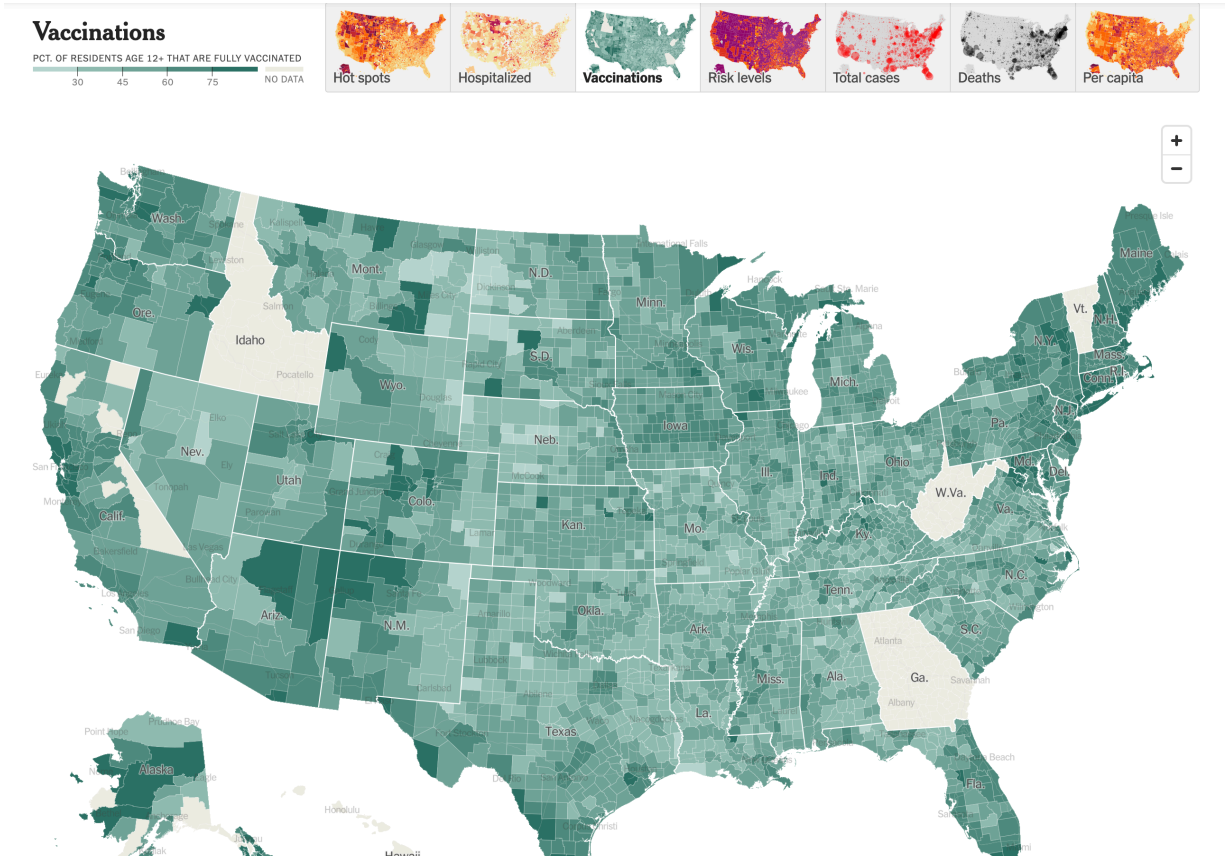
https://restor.eco

— Multitouch-Table

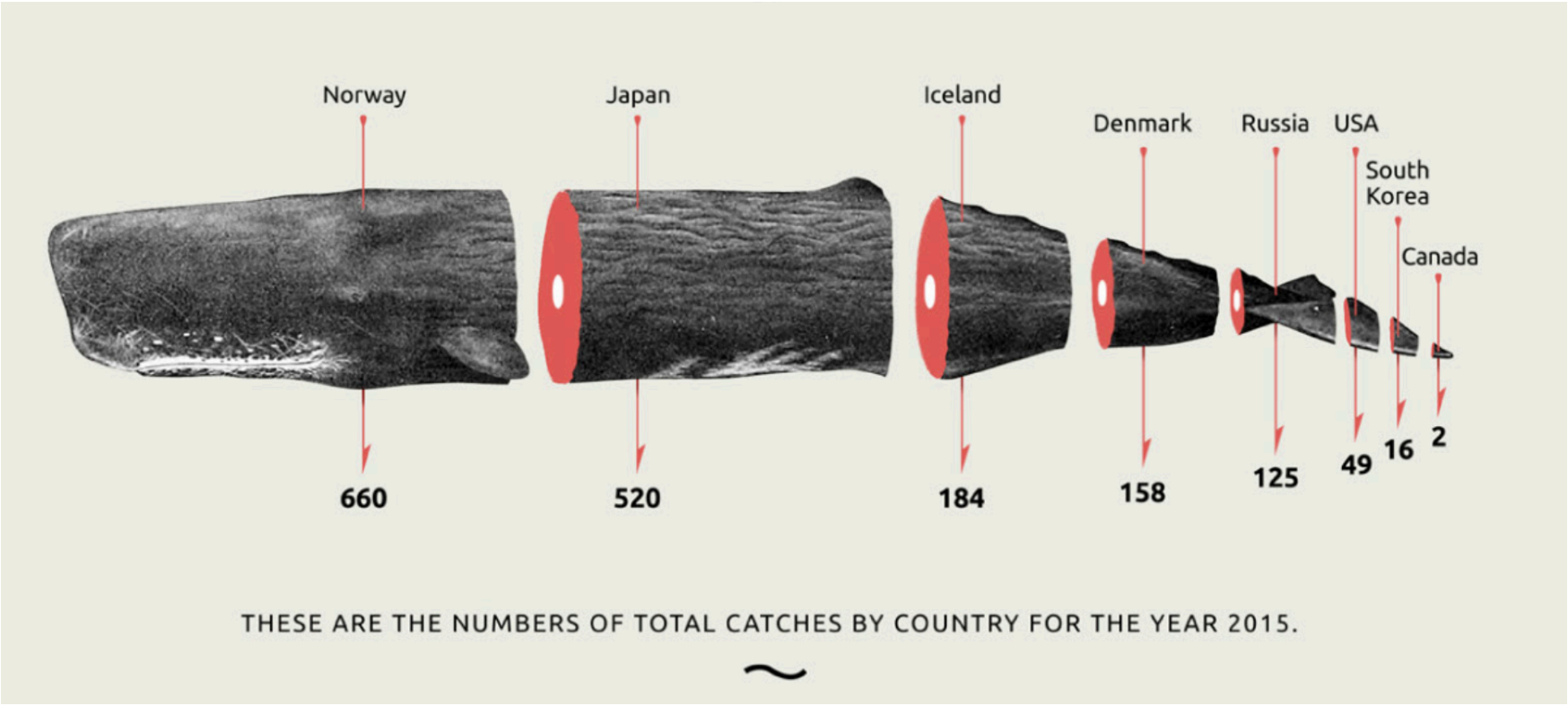


Why?

Storytelling!



https://www.nytimes.com/interactive/2021/us/covid-cases.html

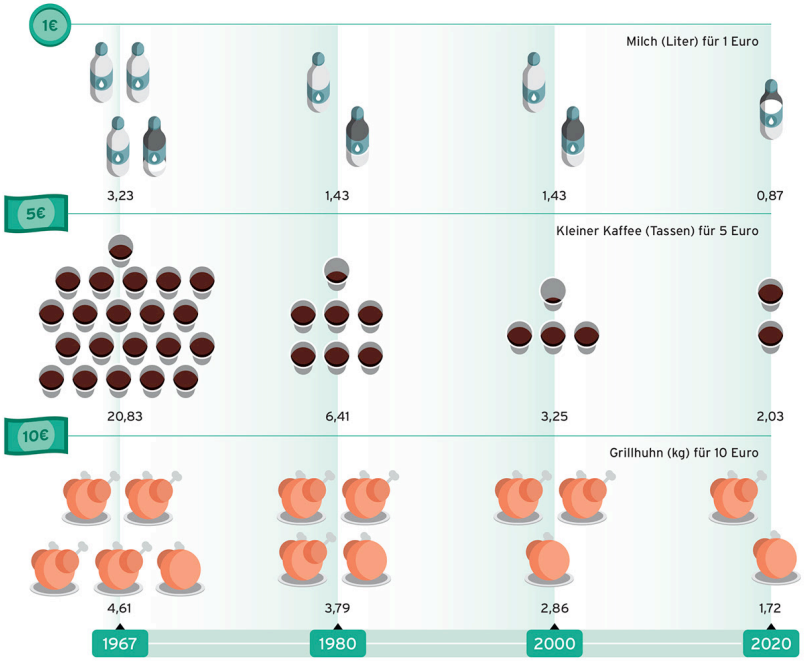


https://www.behance.net/gallery/87757735/Chit-Chart

Wie die Inflation zuschlägt

Was man fürs gleiche Geld über die Jahrzehnte kaufen konnte und kann.

Der Installateur kostet in Österreich rund 100 Euro pro Stunde. Blickt man zurück auf Schillingzeiten, wird die Geldentwertung in diesem Fall besonders krass: Nur sieben Euro kostete im Jahr 1967 ein Installateur pro Stunde. Solche Preise sind Momentaufnahmen, die von der Statistik Austria monatlich in Supermärkten, bei Handwerkern, Friseuren, Restaurants etc. eingeholt werden. Daher hinkt der langfristige Vergleich manchmal. Außerdem ist die Kaufkraft seither gestiegen. Weitere Aspekte: Ein Grillhuhn hat einen höheren Fleischanteil als früher, und Fahrräder aus den Sechzigern können mit neuen Modellen nicht mithalten. Am Kaffeehausbesuch hat sich hingegen über die Jahrzehnte wenig geändert. Nur bekommt man heute für fünf Euro nicht mehr als zwei Mokka – vor vierzig Jahren konnte man im Kaffeehaus ums gleiche Geld mehr als die dreifache Espressomenge bestellen.



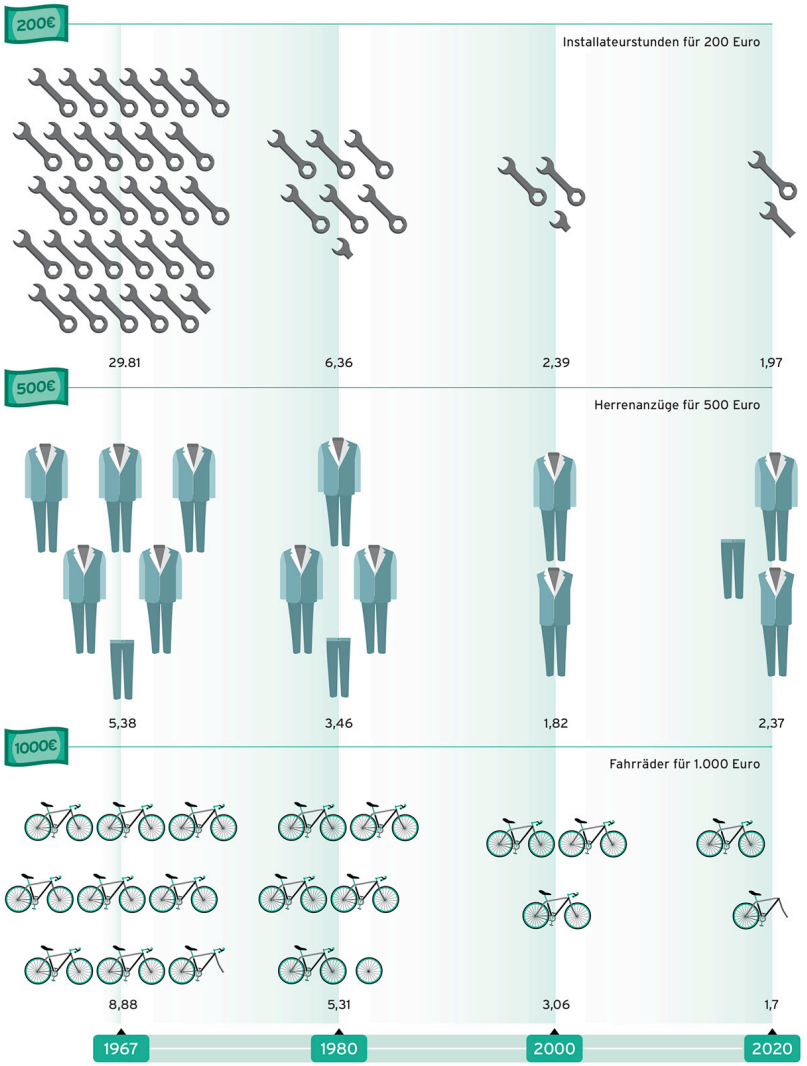


ILLUSTRATION: VALERIO PELLEGRINI

Quelle: Statistik Austria

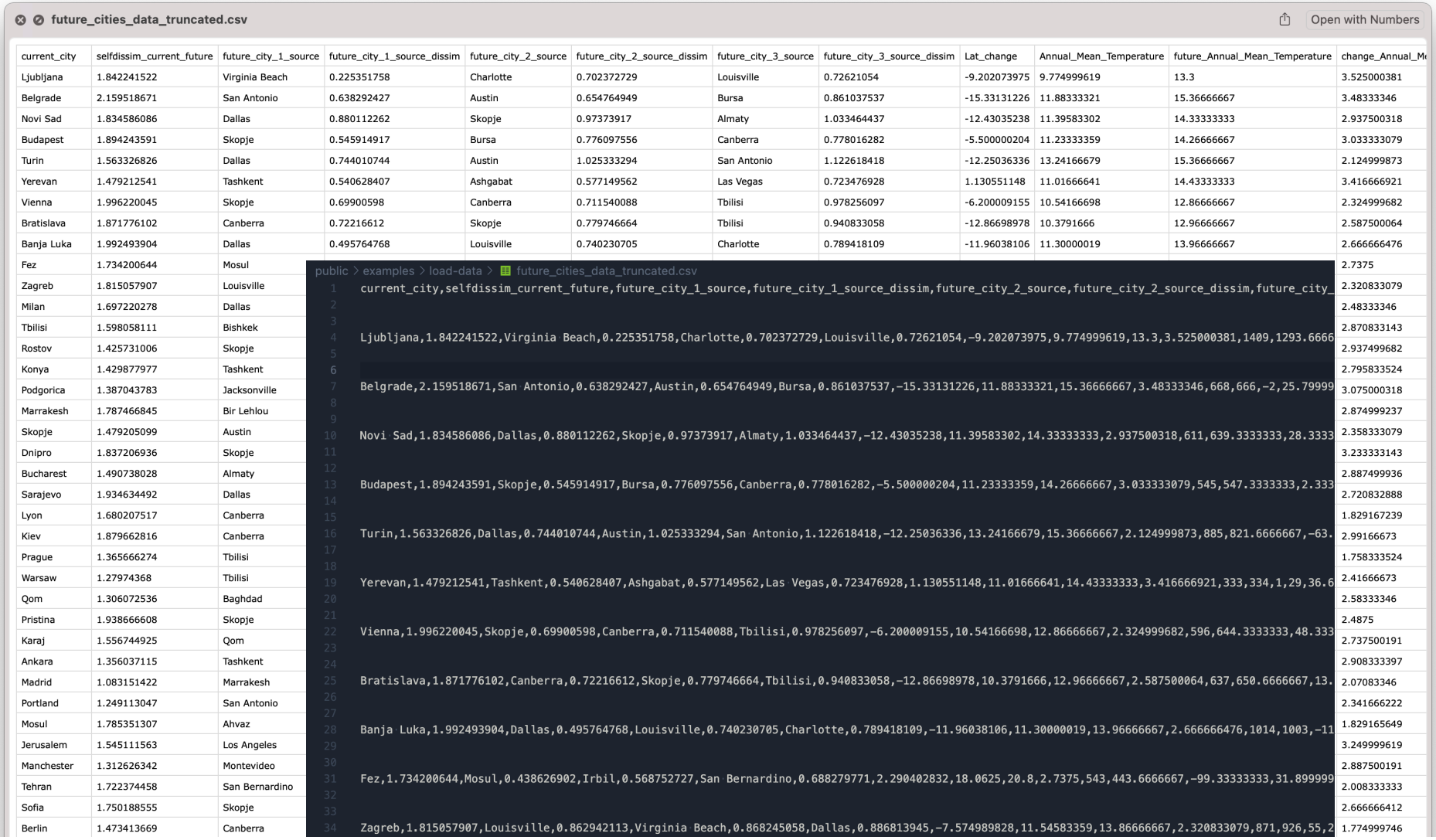
https://www.behance.net/gallery/129006847/DER-PRAGMATICUS-Wie-die-Inflation-zuschlagt

Data

Data Sources:
— https://www.kaggle.com/
— https://www.bfs.admin.ch/bfs/de/home.html
— https://opendata.swiss/de/dataset/kuhe-anz (e.g. Number of Cows)

Data Formats?

CSV


future_cities_data_truncated.csv
Open with Numbers
public > examples > load-data > future_cities_data_truncated.csv
current_city,selfdissim_current_future,future_city_1_source,future_city_1_source_dissim,future_city_2_source,future_city_2_source_dissim,future_city_
Ljubljana,1.842241522,Virginia Beach,0.225351758,Charlotte,0.702372729,Louisville,0.72621054,-9.202073975,9.774999619,13.3,3.525000381,1409,1293.6666
Belgrade,2.159518671,San Antonio,0.638292427,Austin,0.654764949,Bursa,0.861037537,-15.33131226,11.88333321,15.36666667,3.48333346,668,666,-2,25.79999
Novi Sad,1.834586086,Dallas,0.880112262,Skopje,0.97373917,Almaty,1.033464437,-12.43035238,11.39583302,14.33333333,2.937500318,611,639.3333333,28.3333
Budapest,1.894243591,Skopje,0.545914917,Bursa,0.776097556,Canberra,0.778016282,-5.500000204,11.23333359,14.26666667,3.033333079,545,547.3333333,2.333
Turin,1.563326826,Dallas,0.744010744,Austin,1.025333294,San Antonio,1.122618418,-12.25036336,13.24166679,15.36666667,2.124999873,885,821.6666667,-63.
Yerevan,1.479212541,Tashkent,0.540628407,Ashgabat,0.577149562,Las Vegas,0.723476928,1.130551148,11.01666641,14.43333333,3.416666921,333,334,1,29,36.6
Vienna,1.996220045,Skopje,0.69900598,Canberra,0.711540088,Tbilisi,0.978256097,-6.200009155,10.54166698,12.86666667,2.324999682,596,644.3333333,48.333
Bratislava,1.871776102,Canberra,0.72216612,Skopje,0.779746664,Tbilisi,0.940833058,-12.86698978,10.3791666,12.96666667,2.587500064,637,650.6666667,13.
Banja Luka,1.992493904,Dallas,0.495764768,Louisville,0.740230705,Charlotte,0.789418109,-11.96038106,11.30000019,13.96666667,2.666666476,1014,1003,-11
Fez,1.734200644,Mosul,0.438626902,Irbil,0.568752727,San Bernardino,0.688279771,2.290402832,18.0625,20.8,2.7375,543,443.6666667,-99.33333333,31.899999
Zagreb,1.815057907,Louisville,0.862942113,Virginia Beach,0.868245058,Dallas,0.886813945,-7.574989828,11.54583359,13.86666667,2.320833079,871,926,55,2
Milan,1.697220278,Dallas
Tbilisi,1.598058111,Bishkek
Rostov,1.425731006,Skopje
Konya,1.429877977,Tashkent
Podgorica,1.387043783,Jacksonville
Marrakesh,1.787466845,Bir Lehlou
Skopje,1.479205099,Austin
Dnipro,1.837206936,Skopje
Bucharest,1.490738028,Almaty
Sarajevo,1.934634492,Dallas
Lyon,1.680207517,Canberra
Kiev,1.879662816,Canberra
Prague,1.365666274,Tbilisi
Warsaw,1.27974368,Tbilisi
Qom,1.306072536,Baghdad
Pristina,1.938666608,Skopje
Karaj,1.556744925,Qom
Ankara,1.356037115,Tashkent
Madrid,1.083151422,Marrakesh
Portland,1.249113047,San Antonio
Mosul,1.785351307,Ahvaz
Jerusalem,1.545111563,Los Angeles
Manchester,1.312626342,Montevideo
Tehran,1.722374458,San Bernardino
Sofia,1.750188555,Skopje
Berlin,1.473413669,Canberra
2.7375
2.320833079
2.48333346
2.870833143
2.937499682
2.795833524
3.075000318
2.874999237
2.358333079
3.233333143
2.887499936
2.720832888
1.829167239
2.99166673
1.758333524
2.41666673
2.58333346
2.4875
2.737500191
2.908333397
2.07083346
2.341666222
1.829165649
3.249999619
2.887500191
2.008333333
2.666666412
1.774999746

JSON

```json
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

https://en.wikipedia.org/wiki/JSON

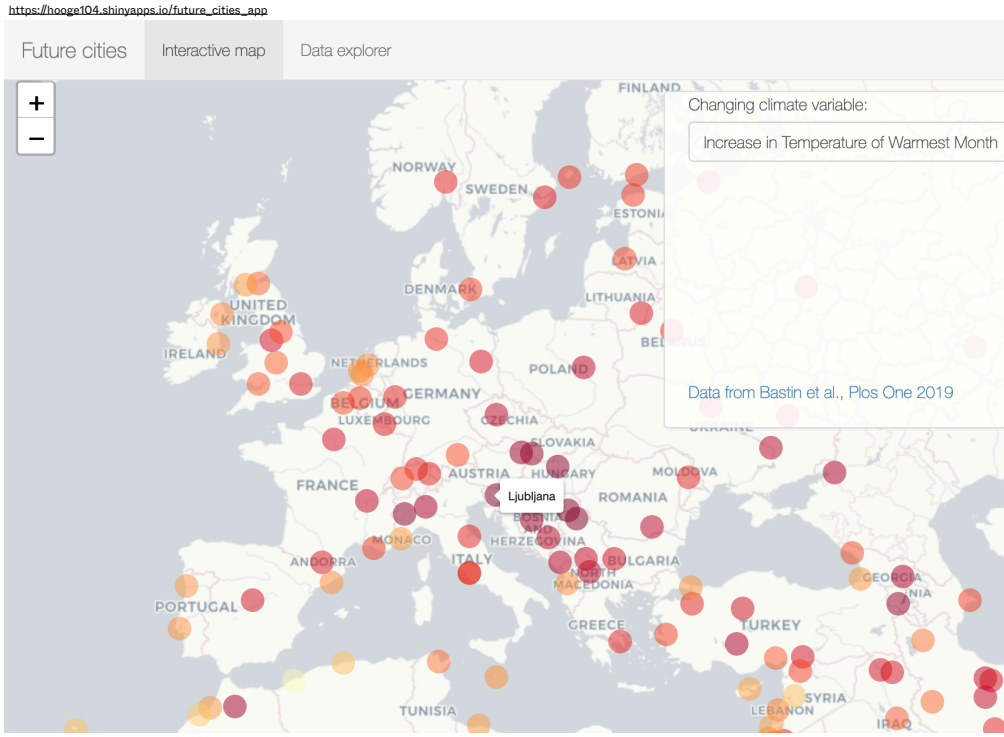
Technical Know-How

Task:

You have a dataset describing cities. You designed a way to display parts of it in an (interactive) visualization.

How would you start? What should we cover?

current_city	selfdissim_current_future	future_city_1_source	future_city_
Ljubljana	1.842241522	Virginia Beach	0.22535175
Belgrade	2.159518671	San Antonio	0.63829242
Novi Sad	1.834586086	Dallas	0.88011226
Budapest	1.894243591	Skopje	0.54591491
Turin	1.563326826	Dallas	0.74401074
Yerevan	1.479212541	Tashkent	0.54062840
Vienna	1.996220045	Skopje	0.69900598
Bratislava	1.871776102	Canberra	0.72216612
Banja Luka	1.992493904	Dallas	0.49576476
Fez	1.734200644	Mosul	0.43862690
Zagreb	1.815057907	Louisville	0.86294211
Milan	1.697220278	Dallas	0.60098778
Tbilisi	1.598058111	Bishkek	0.76871764
Rostov	1.425731006	Skopje	0.82619532
Konya	1.429877977	Tashkent	0.34310528



Technical Know-How

— How to load the data? (→ Input Paulina)
— How to display the data?
— How to make it interactive?
— How to use gestures to control interactivity?

— p5.js
  — Do you know: push(); pop(); / drawing contexts?
— (Vanilla JavaScript)
— d3.js
— Chart.js

Pause

Exercise (10–15min)

— Set up a new github repository for this module.
— Download p5.js example. Try to read and understand the code.
https://bits-atoms-3.jonasscheiwiller.ch/downloads/load-data.zip
— Use a local server (e. g. Visual Studio Code **Live Share** extension) to serve the index.html

Discussion

Exercise: Display data (15min)

— Use p5.js to display the loaded dataset (or parts of it).
— Try to write abstract code first, without using a code editor.

https://p5js.org/

Discussion

Exercise: Core concepts of D3

https://www.d3-graph-gallery.com/intro_d3js.html