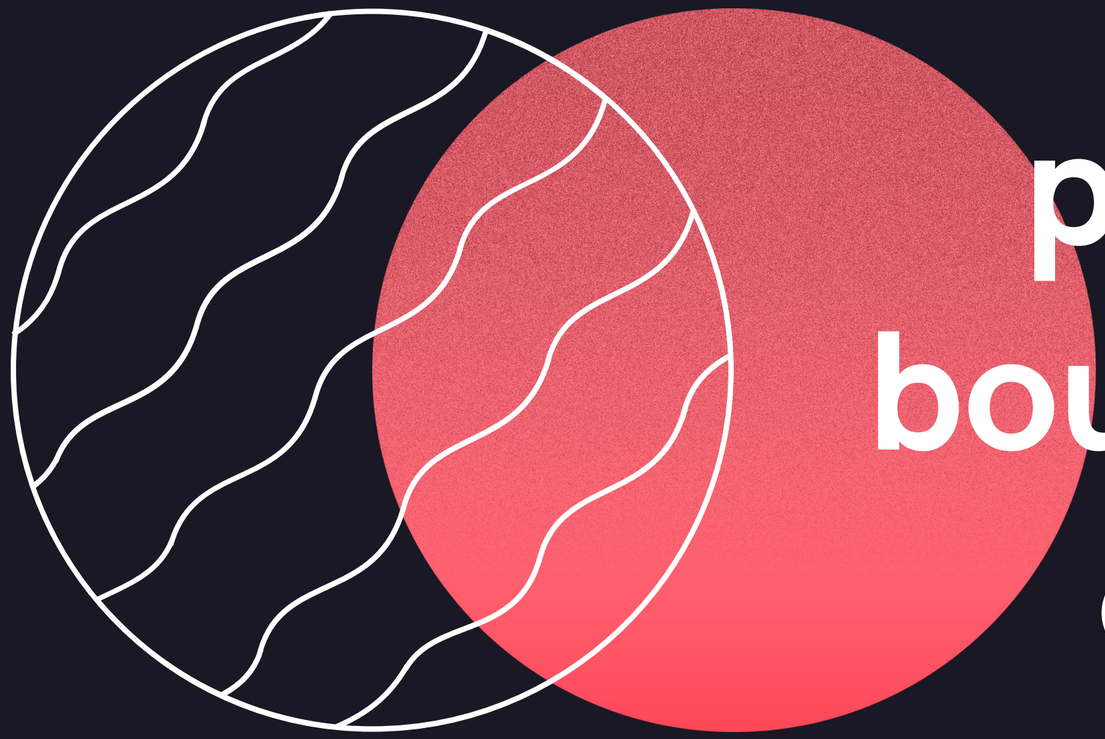
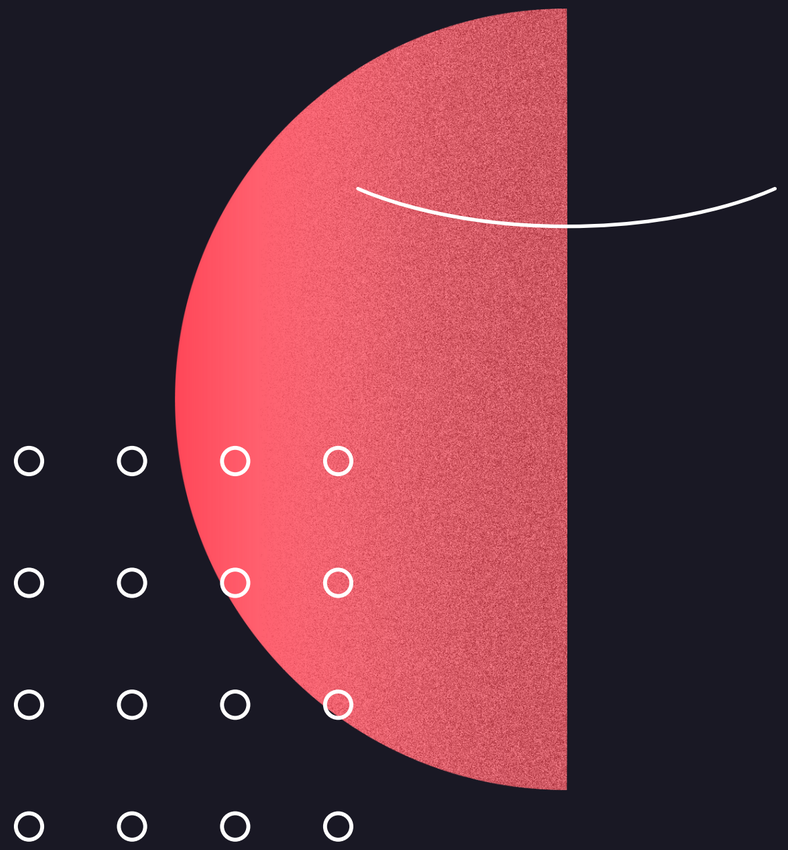


Crossing pedagogical boundaries with open data

Santeri Koivula, Peitsa Veteli
and Veera Juntunen
Helsinki Institute of Physics HIP

The agenda



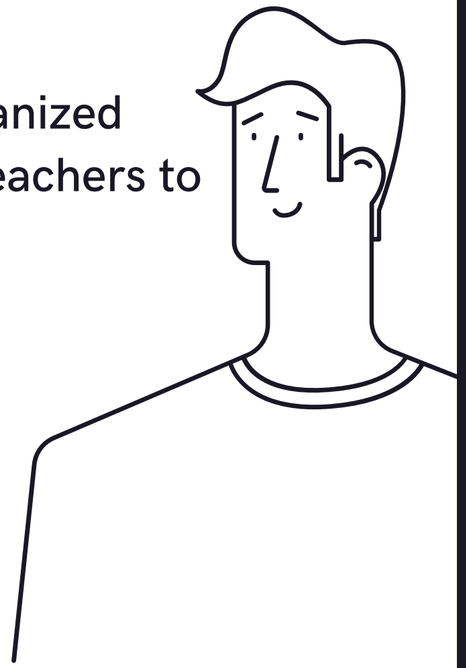
- What is this about?
- Exercise: What is the Jupyter Notebook?
- Going through an advanced example
- Discussion exercises
- Website
- Feedback

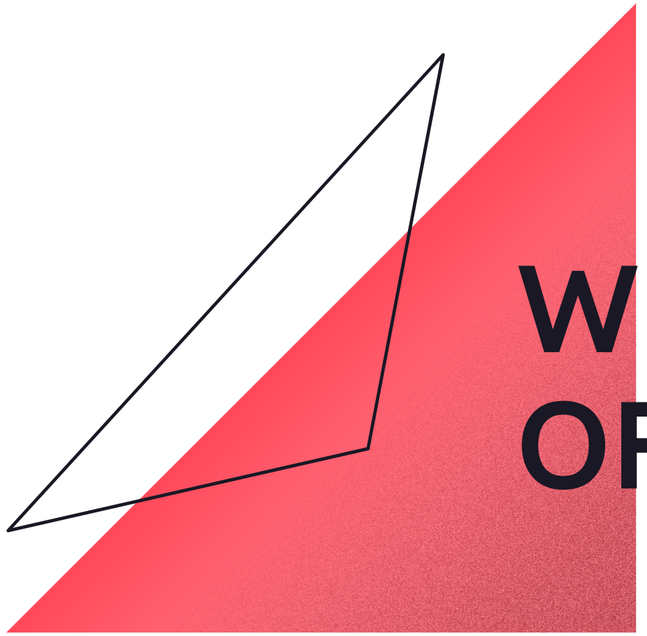
WHAT IS THIS ABOUT?

Open data is everywhere — CERN, NASA, The World Bank, etc.

Our mission is to help teachers to create research-based exercises for their courses using open data. Multidisciplinarity; curiosity; and skills for data analysis are at the heart of what we do.

We have created study materials, organized workshops for teachers, and helped teachers to use open data in their classes.





WHAT IS OPEN DATA?

Data:
Factual information collected to be used for reasoning, discussion, or decision-making

Open data:
Data that can be freely used, modified, and shared by anyone for any purpose (according to Open Definition)

HOW WOULD THIS BE USEFUL FOR YOU?

- You are planning to teach at some point
- You are planning to go into research
  - Communicating about your results
- Critical thinking
  - Assessing the credibility of data
  - Analyzing information in different contexts

WHY?

WHY?

Future citizenship - understanding, analyzing information and assessing its credibility

The amount of data is increasing

Communication skills

Getting acquainted with the tools of science

Understanding large phenomena and different contexts

Multidisciplinary learning

Ability to understand and question

THE PERSPECTIVE OF A STUDENT

In its simplest form, the student is provided with a link of an exercise that uses open data. An exercise can be saved either as a notebook or a PDF file.

The exercises can also work as a tool for a student to reflect on their learning, in which case other programs or platforms are not even needed.

Easy!

VISUALIZING DATA

INFORMATIVE COMMUNICATION

UNDERSTANDING DATA

ASSESSING CREDIBILITY

PROGRAMMING

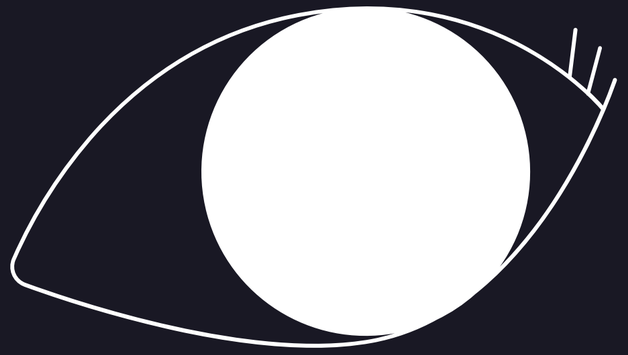
SEARCHING FOR DATA

FINDING REPEATING PATTERNS

What kind of materials?

Do you want to make materials by yourself?

How much coding?



What is the group size?

How much can students influence on the exercises?

Returning the exercises?

HOW MUCH WORK FOR TEACHERS?

BEGINNER – INTERMEDIATE – EXPERT

WHAT OUR EXERCISES LOOK LIKE

Plotting the invariant mass histogram

In this excercise, we learn how to plot the histogram of invariant masses with Python. Let us use the data collected by the CMS detector in 2011 [1]. Events with specific criteria [2] have been selected in the CSV file *Ymumu_Run2011A.csv*, which we are using.

Explore the different code cells below and run the code. Note that normally the code would not be commented as much as this. Here, the reason for these comments is to explain in detail what the code is doing.

[1] CMS collaboration (2016). DoubleMu primary dataset in AOD format from RunA of 2011 (/DoubleMu/Run2011A-12Oct2013-v1/AOD). CERN Open Data Portal. DOI: 10.7483/OPENDATA.CMS.RZ34.QR6N.
[2] Thomas McCauley (2016). Ymumu. Jupyter Notebook file. https://github.com/tpmccauley/cmsopendata-jupyter/blob/hst-0.1/Ymumu.ipynb.

1) Start

```
# Import the needed modules. Pandas is for the data-analysis
# and matplotlib.pyplot for making plots. Modules are named as pd and plt.
import pandas as pd
import matplotlib.pyplot as plt

# Jupyter Notebook uses "magic functions". With this function it is possible to plot
# the histogram straight to notebook.
%matplotlib inline
```

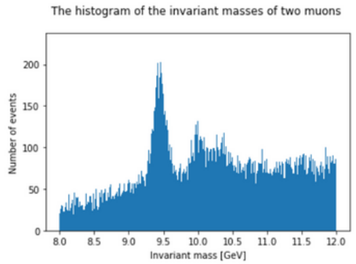
2) Getting the data

```
# Create a new DataFrame structure from the file "Ymumu_Run2011A.csv"
dataset = pd.read_csv('../../Data/Ymumu_Run2011A.csv')

# Create a Series structure (basically a list) and name it "invariant_mass".
# Save the column "M" from the "dataset" to the variable "invariant_mass".
invariant_mass = dataset['M']
```

3) Plotting the histogram

Now we can create and plot the histogram of the values of the invariant masses. The histogram shows for how many events the invariant mass of the muon pair is in a certain value range. Note that we will use total 500 bins in the histogram, so you will not spot the separate bins because there are so many of them.

```
# Plot the histogram with the function hist() of the matplotlib.pyplot module:
# (http://matplotlib.org/api/pyplot_api.html?highlight=matplotlib.pyplot.hist#matplotlib.pyplot
# 'Bins' determines the number of bins used.
plt.hist(invariant_mass, bins=500)

# Name the axises and give a title.
plt.xlabel('Invariant mass [GeV]')
plt.ylabel('Number of events')
plt.title('The histogram of the invariant masses of two muons \n') # \n creates a new line for

# Show the plot.
plt.show()
```



4) Analysis

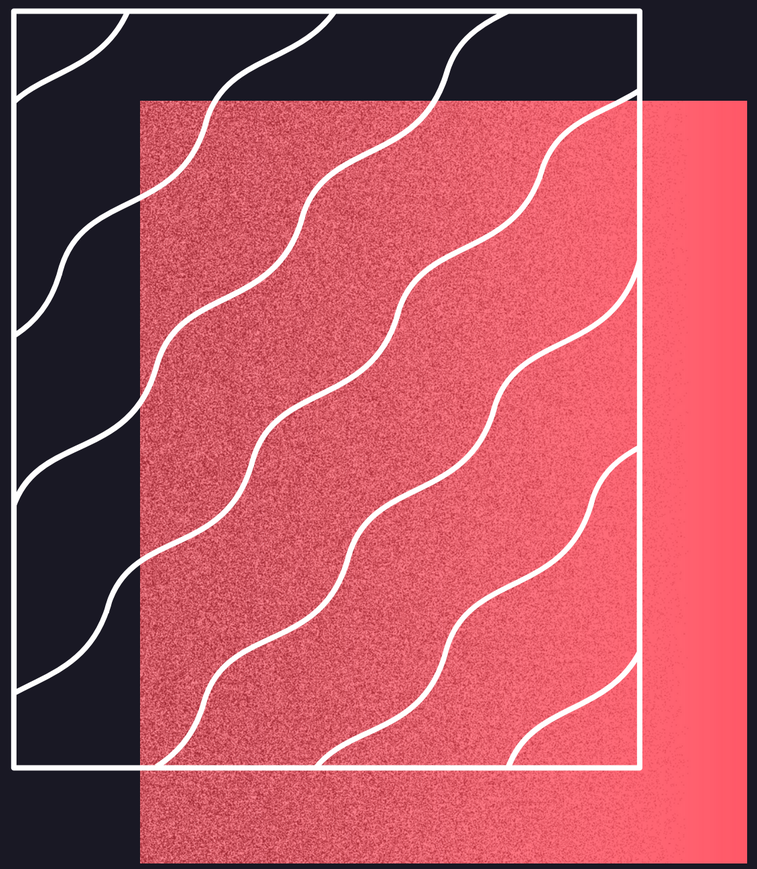
- What does the histogram tell us?
- What happens around the mass 9.5 GeV?

Previous Calculating the invariant mass

Next Advanced

By HIP Education and Open Data Team

Development of this material is made possible by a grant from Finnish National Agency of Education



Versatility - text, code, images, videos, animations

Students can return one document only

... What about Excel?

WHY LEARN A NEW PLATFORM?

Everything from instructions to exercises are in the same place

Teacher can easily run the commands again while going over the results

DIFFERENT SUBJECTS

Open data can be used in many different subjects.

In Finnish we have materials on physics, biology, text analysis, geography, and mathematics.

Similar exercises could be used in other fields as well, such as history, economics, and psychology.

Pumput - saastumislähteet ja terveysmaantiede

Tässä osiossa käytetään autenttisia tietoja tapahtumista elo-syyskuussa 1854.

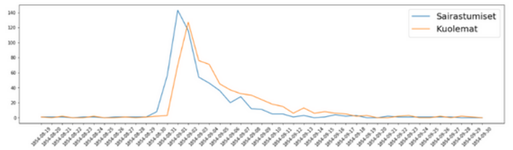
John Snow kuuli asiasta 4.9. ja käytti seuraavat kolme päivää juosten ympäriinsä keräämässä aineistoa, piirtäen karttoja ja vakuuttaen paikallishallintoa tarpeellisista vastatoimista. Jokainen tuhlattu hetki tarkoitti lisää tartuntoja ja kuolleita.

```
# AJA NÄMÄ PAKETIT ENSIN, JOLLET AJANUT EDELLISTÄ OSIOTA

import pandas as pd
import matplotlib.pyplot as plt
import random as rnd
import numpy as np
```

```
# Historiallinen data, joka on kerätty Robin Wilsonin julkaisemista paketeista
# täältä http://blog.rtwilson.com/john-snows-cholera-data-in-more-formats/

ajat = pd.read_csv("../data/johnsnow_dataset_dates_all.csv")
kuolinluvut = pd.read_csv("../data/johnsnow_dataset_deaths.csv")
pumput = pd.read_csv("../data/johnsnow_dataset_pumps_names.csv")
```

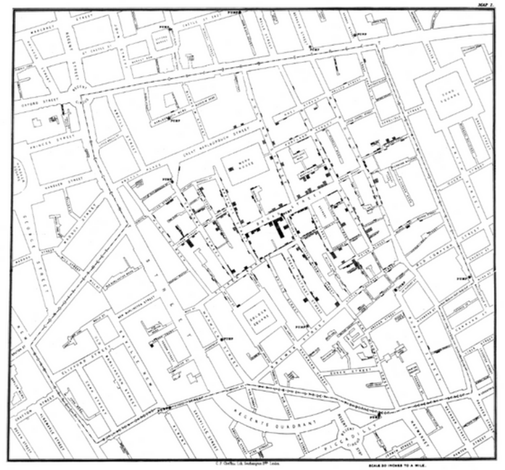
```
# Tästä nähdään tapahtumien aikakehitys.

plt.figure(figsize=(20,5))
plt.plot(ajat["date"], ajat["attacks"], label = 'Sairastumiset')
plt.plot(ajat['date'], ajat['deaths'], label = 'Kuolemat')
plt.xticks(rotation='45')
plt.legend(fontsize = 20)
plt.title('Päivittäiset uudet tapaukset \n', fontsize = 20)

plt.show()
```

Päivittäiset uudet tapaukset



Yllä olevasta kuvaajasta nähdään naapuruston sairastapausten räjähtävän käsiin kuun taitteessa. Myöhemmissä arvioissa Snow on uskonut taudin olleen jo luonnostaan laskussa toimiensa aikaan (esimerkiksi ihmisten karattua paikalta), mutta jotain ratkaisevaa tapahtuu 8.9., mikä katkaisee isomman leviämisen lähes samantien.

Snow kiersi paikanpäällä aikansa, mutta yhden ihmisen tiedonkeruuoperaatio kuolevien ihmisten, paetessa hylättyjen talojen ja yleisen kaaoksen keskellä olisi tullut liian hitaaksi. Sen sijaan tohtorimme kääntyi tehokkaampaan suuntaan ja marssi paikallisen tilastokeskuksen, Office of Register Generalin, puheille ja vaati käyttöönsä kaikkien kuolemantapausten ajat ja osoitteet. Alla on alueen kartta, mihin hän merkkasi kuolleet mustina vaakaviivoina kuin vierekkäiset hauta-arkut pihoille.

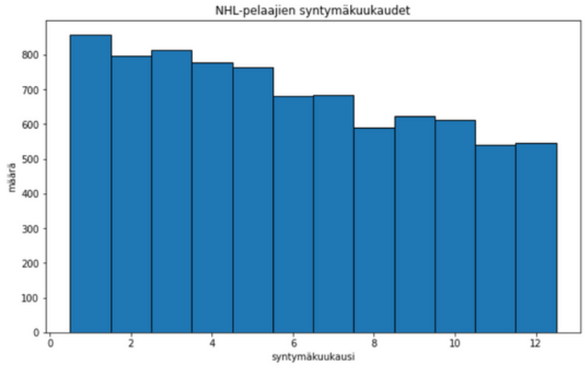


Tehtävä 2:

Katsomalla yllä olevaa karttaa, mitä voit sanoa kuolintapausten asettumisesta kartalle?

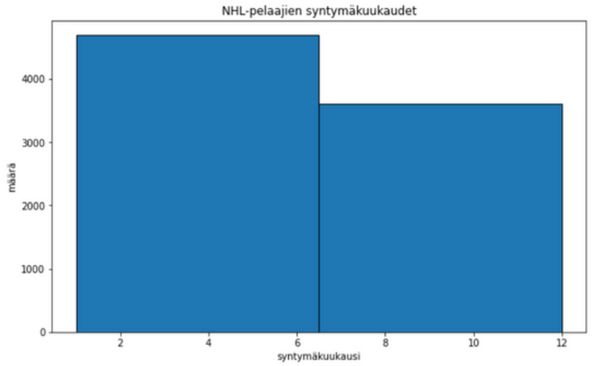
Hienoa, saimme valmiiksi ensimmäisen histogrammin! Voimme halutessamme tehdä siitä kauniimman nimeämällä akselit ja lisäämällä muita selkeyttäviä ominaisuuksia.

```
fig, ax = plt.subplots()  #Talletetaan kuvaaja ja sen akselit muuttujiin
fig.set_figheight(6)
fig.set_figwidth(10)      #Asetetaan kuvaajan mitat sopivan kokoisiksi
bins = np.arange(1,14)    #Käytetään luokkien määrään numpy-kirjaston arange-funktiota, jotta l
plt.hist(kuukaudet, bins=bins,edgecolor='k', align='left')  #'edgecolor'-muuttujan avulla eri k
plt.title('NHL-pelaajien syntymäkuukaudet')  #Nimetään kuvaaja
plt.xlabel('syntymäkuukausi')   #Nimetään x-akseli
plt.ylabel('määrä')     #Nimetään y-akseli
plt.show()
```



Nyt näyttää jo paremmalta! Muuttamalla binien määrää voimme tehdä uudenlaisia havaintoja. Voi esimerkiksi olla hyödyllistä tarkastella, kuinka suuri osuus pelaajista on syntynyt alku- ja loppuvuonna. Tämä saadaan aikaan vaihtamalla luokkien määräksi kaksi.

```
fig = plt.figure(figsize=(10,6))
plt.hist(kuukaudet, bins=2,edgecolor='k')
plt.title('NHL-pelaajien syntymäkuukaudet')  #Nimetään kuvaaja
plt.xlabel('syntymäkuukausi')   #Nimetään x-akseli
plt.ylabel('määrä')     #Nimetään y-akseli
plt.show()
```



Todetaan, että NHL-pelaajien keskuudessa alkuvuonna syntyneitä on huomattavasti enemmän kuin loppuvuotena syntyneitä. Osaatko sanoa, mistä tämä voisi johtua?

Jos tavoitteenasi oli oppia aivan perusteet histogrammeista, yllä olevan lukeminen on tarpeeksi. Jatka lukemista, jos haluat syventyä vielä lisää histogrammeihin ja niiden ominaisuuksiin.

Remember that we also wanted to display the words in a shape of an ice cream. For this, we need an image of an ice cream to be used as a mask. The area we want our words to be in should be black in the image and the backgbround should be white. Therefore, we need a black icecream on a white background. This kind of image can be found in the file called "icecream.png". To open the image in python, we first need to use a module called Image from PIL-package and then transform it into an RGB-array using `array()` from numpy-package.

```
import numpy as np
from PIL import Image

# Open the image file
image = Image.open('../images/icecream.png')

# Transform the image into an RGB array
icecream_mask = np.array(image)
```

Now we can make the word cloud by a similar manner as before, but adding some additional parameters to the WordCloud-object. Check the cell before for details!

```
# Get the cleaned review column from data-table as single string.
data_string = data['review_cleaned'].to_string()

# Make a WordCloud-object with specified mask, background color, contour width and contour color
wordcloud = WordCloud(mask = icecream_mask, background_color='white', contour_width=3, contour_color='violet'

# Generate the words to cloud from our data string
wordcloud.generate(data_string)

# Plot the figure, optionally ignore the axis.
plt.figure (figsize = (8,8), facecolor = None)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

That's it! We've created a cool, ice cream -shaped wordcloud from ice cream stall reviews! Now, try to make a word cloud yourself from some other set of words! You can even try to create your own mask for the figure.

Happy coding!

Note

If you want to define the weight of each word as a number instead of it's frequency in a text, you can generate the cloud using `generate_from_frequencies()`-method. You just need to pass a dictionary as a parameter. For example you could write

```
words = {"Dog":10, "Cat": 5, "Cow":5, "Platypus":3}
wordcloud.generate(words)
```

This would result in a cloud, where the word "Dog" would be the largest and "Platypus" the smallest.

You can save your image by writing

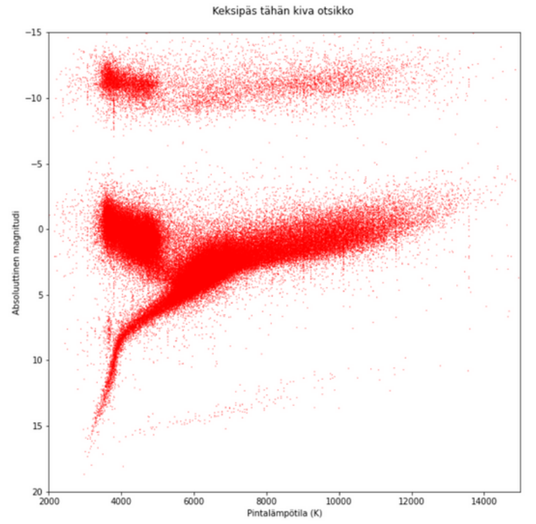
```
plt.savefig("filename.png")
```

Kerro, kerro, kuvaaja

```
# Tässä otetaan "taivas"-tietorakenteesta kaksi saraketta ja luodaan niiden arvopareista ku

plt.figure(figsize=(15, 4))
plt.scatter(taivas.ra, taivas.dec, s=0.01) # Mitä tietoja tässä käytettiin?
plt.xlim(24, 0)

plt.title("Kaikkien katalogissa olevien tähtien paikat Maasta katsottuna \n")
plt.xlabel('Tarvitsen nimen') # Korjaapa tämä otsikko sopivaksi!
plt.ylabel('Akseloin, siis olen') # Korjaapa tämä otsikko sopivaksi!

plt.show()

# Näetkö muotoja kuvassa? Mistä moiset voisivat johtua?
```

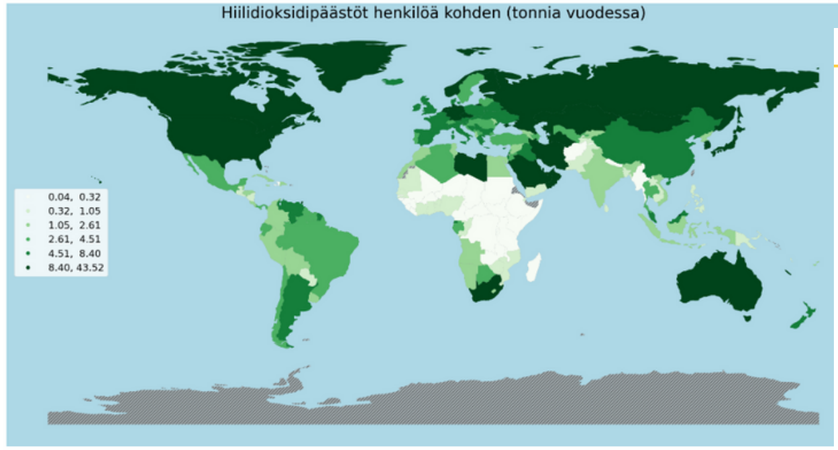
Tähtiä luokitellaan usein lämpötilansa ja kirkkautensa mukaan, siten että kirkkaammilla tähdillä on pienempi absoluuttinen magnitudi. Auringolla se on noin 4,8 (vai oliko? Mitä sanoo tarkastelemamme data?).

```
# Piirretään taas x,y-parien mukainen kuvaaja, jonka pisteitä
# käsitellään hieman niin, että niistä erottaakin jotakin.

plt.figure(figsize = (10,10))
plt.scatter(taivas.temp, taivas.absmag, s = 1, edgecolors = 'none', color = "red")
plt.xlim(2000,15000) # Mitä käy jos säädät näitä numeroita?
plt.ylim(20,-15)

plt.title("Keksipäs tähän kiva otsikko \n") # Muokkaa tämä sopivammaksi.
plt.ylabel("Absoluuttinen magnitudi")
plt.xlabel("Pintalämpötila (K)")

plt.show()
```



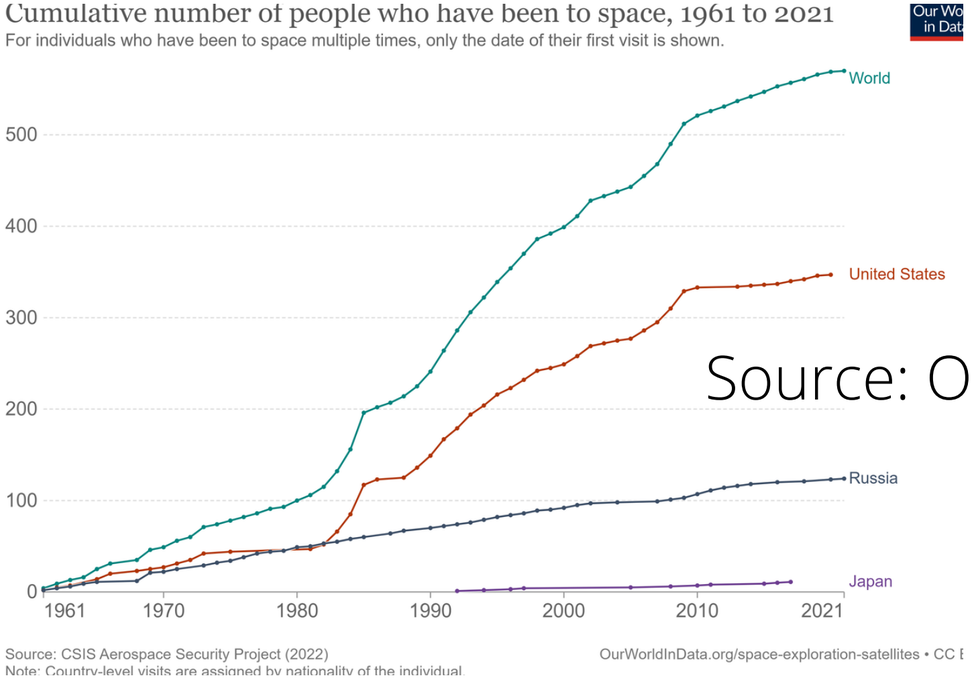
Painottuvatko tulokset jonnekin? Erottuvatko jotkin alueet muusta massasta? Löydätkö kuvasta kirkkaan, muttei erityisen lämpimän alueen (jättiläisiä)? Erityisen kuuman ja kirkkaan, mutta pienistä tähdistä koostuvan valkoisten kääpiöiden alueen?

```python
# Valmistellaan kuva, tehdään kuvasta suuri ja asetetaan taustaväriksi sininen
fig, ax = plt.subplots(1, figsize=(50,20), facecolor='lightblue')

# Piirretään taustaksi kaikki maat ja täytetään ne viivoilla.
# Tällöin ne maat, joista ei ole dataa, erottuvat helpommin.
world.plot(ax=ax, color='darkgrey', alpha=0.8, hatch= "///")

# Piirretään data taustan päälle
data.plot(
    column='2014 [YR2014]', # Määritetään piirrettävä sarake
    ax=ax,                  # Piirretään kuva samalle akselille kuin tausta
    cmap='Greens',          # Käytetään vihreää värikarttaa
    legend=True,            # Lisätään selite
    legend_kwds={           #
        'fontsize': 25,     # Asetetaan selitteen fonttikoko
        'loc':'center left' # Asetetaan selitteen sijainti
    },                      #
    scheme='quantiles',     # Jaetaan datan värit samankokoisiin osiin
    k=6                     # Valitaan osien lukumääräksi 6
)

# Lisätään kuvalle otsikko
plt.title('Hiilidioksidipäästöt henkilöä kohden (tonnia vuodessa)',fontsize = 40)

# Poistetaan akselit kuvan reunoilta
ax.axis('off')

# näytetään kuva
plt.show()
```

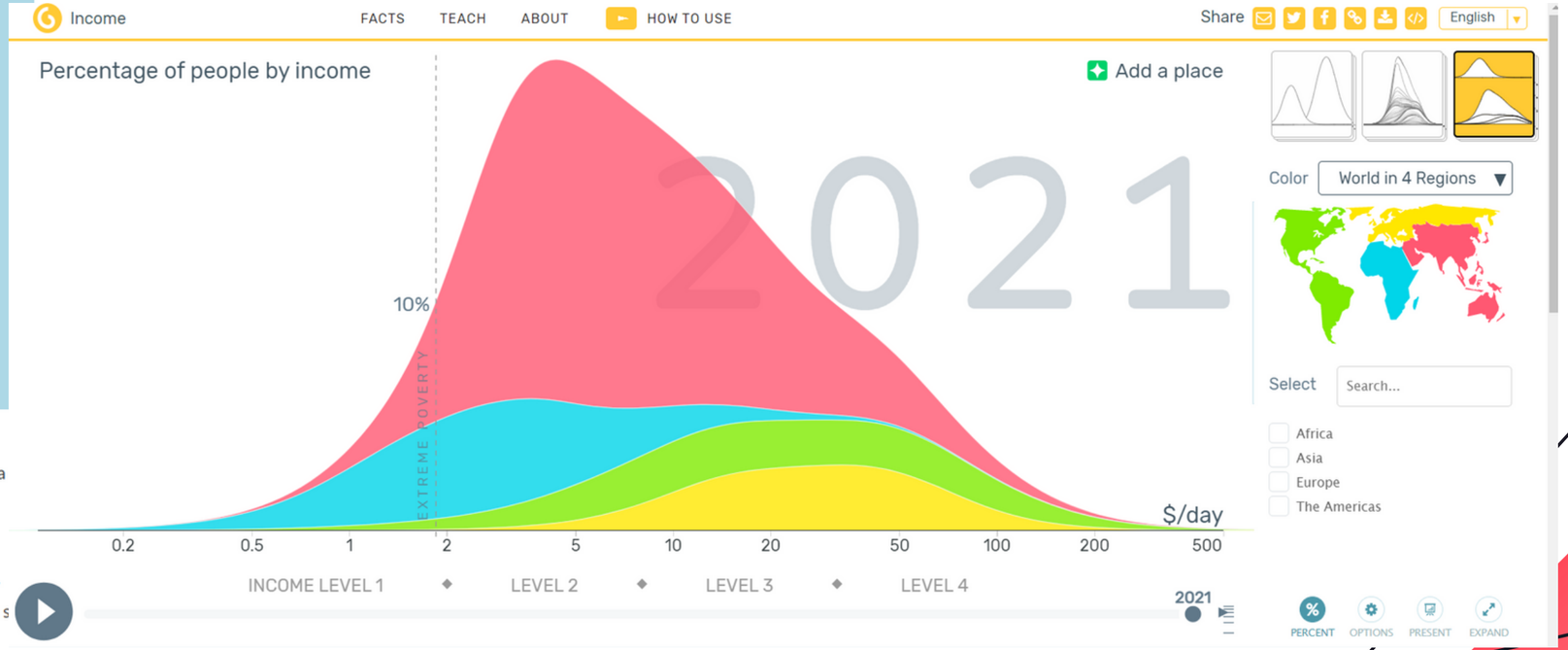


Nyt kartasta erottaa jo huomattavasti paremmin eri alueet ja niitä vastaavat hiilidioksidipäästöt. Voit kokeilla piirtää datan myös jakamalla datan useampaan tai vähempään kuin kuuteen osaan. Miltä data tällöin näyttää? Muokkaa kuvaa haluamallasi tavalla.

Etsi itse jokin toinen valtiokohtainen data ja piirrä se samaan tapaan. Valtiokohtaista dataa löytää googlaamalla tai esimerkiksi osoitteesta https://databank.worldbank.org/source/world-development-indicators#. Maailmanpankin sivuilta pystyy myös tarkastelemaan dataa kartalla, mutta kokeile piirtää s itse!



Source: Our World In Data



Source: Gapminder

PROJECTS

2021

HEALTH EDUCATION
Autumn 2021

PROGRAMMING
Autumn 2021

BIOLOGY
Autumn 2021

Spring 2022

SCIENCE COURSE
Spring 2022

Autumn 2022

SCIENCE COURSE
Autumn 2022

CLIMATE THEME DAY
Autumn 2022

Teacher training Workshops Classes Theme days

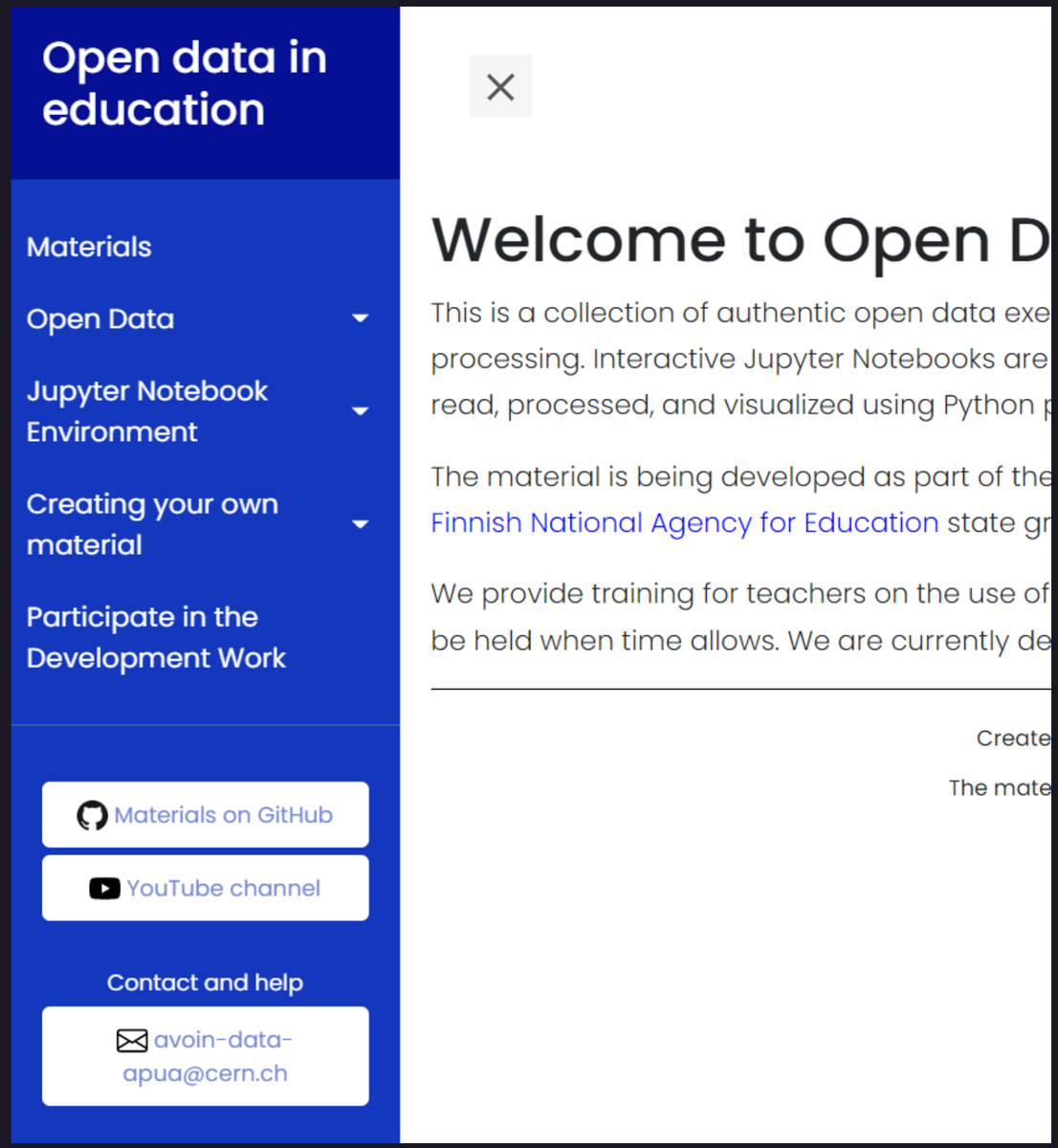
individual work – groupwork – laboratory work – essays – analyses – research work

WEBSITE

and materials

https://opendata-education.github.io/en/

You can find everything you need to get started from our website, such as materials that are ready to use or to modify, and links to websites that publish open data. Currently we have materials in English on particle physics and text analysis.

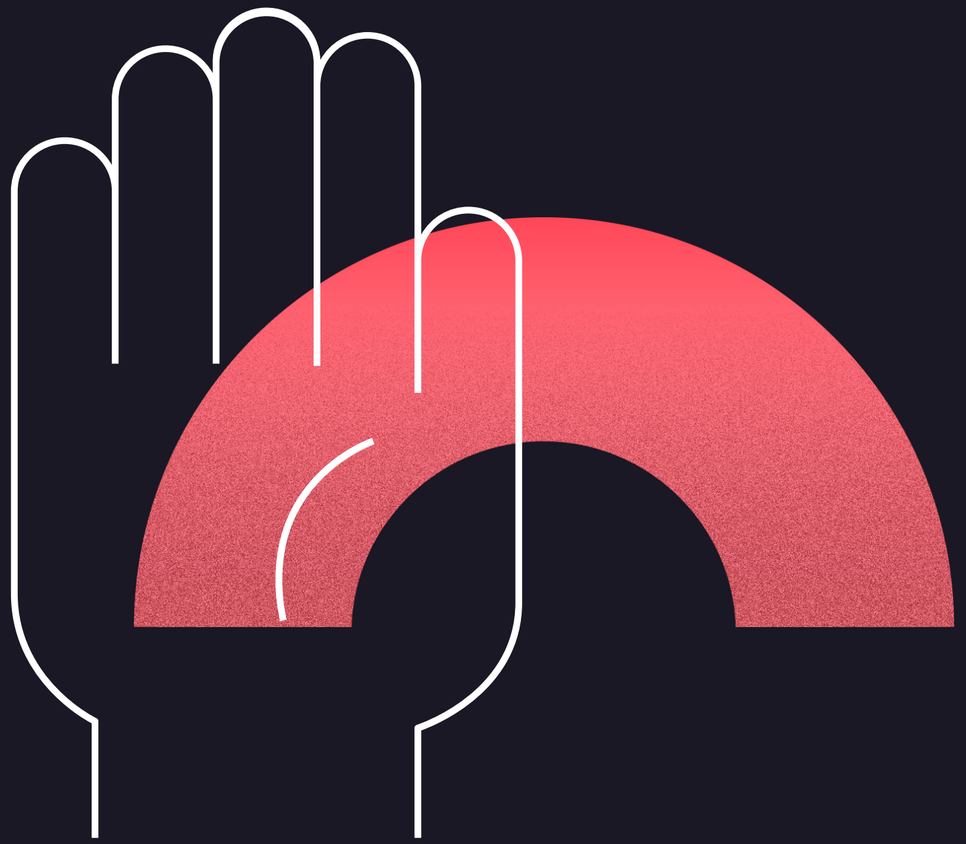


CONTACT US

Peitsa Veteli
peitsa.veteli@helsinki.fi

Veera Juntunen
veera.juntunen@helsinki.fi

Santeri Koivula
santeri.jan.viliam.koivula@cern.ch



NEXT

1. Jupyter Notebook -exercise
2. Going through an advanced example
3. Our website

DATA RESOURCES

- CERN Open Data Portal
- Our World In Data
- Figshare
- Zenodo
- World Bank
- WHO: Global Health Observatory