Python: Data Analysis and Data Visualization

David Pinezich

Learning Targets

- Import data and pre process
  - formats, parse data, use suited data structures
- Aggregate
  - Perform basic analysis
  - descriptive statistics
  - text analysis
- Visualization types
  - Tables, x-y Plot, Normal distribution, Pie chart, Spider diagram, Word cluster, Histogram, 3D Plot

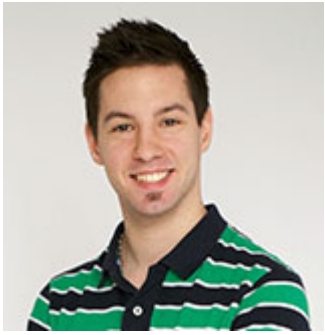
Course Information

- I expect huge skill differences
  - Please tell me if we are moving too fast / too slow
- Per course part we have one exercise to face (5 in total)
  - Working independently with my support when needed
  - Use the internet, but be critical with copy&paste
- Be patient if you progress slowly!

- At the end of each part, I will have discussed our solution with you in detail. There is not one solution, therefore I am always interested in your approach.

About me

- Education
  - Informatiker Applikationsentwicklung EFZ (BMS / Passerelle)
  - Bachelor of Informatics at UZH
  - Currently – Master of Informatics at UZH



- Work Experience
  - Paul Scherrer Institut (PSI)
  - Architonic
  - ti&m
  - Helsana (Lead Web engineering)

- Programming Experience

- david.pinezich@gmail.com / david.pinezich@uzh.ch

What about you?

- Name?
- Field of Study?
- Do you have any programming experience?
  - Languages / Projects?
- Expectations for this course?
- Special requests?

Schedule Today

- Intro and setup of the environment ~ 30min
- Python Crash course & craps.py ~ 30min
- Break ~ 15min
- IMDB Exercise
  - Part 1 ~ 15min
  - Discussion part 1 ~ 15min
  - Remaining parts (independent work) ~ 60min
  - Discussion remaining parts ~ 15min

Starter

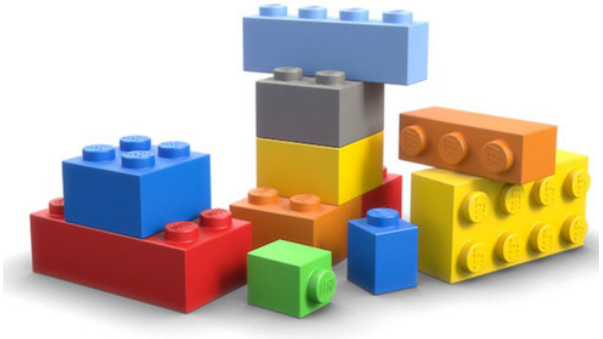
- All starter files are found here:
  - https://github.com/dpinezich/apyd_19/archive/master.zip

Functions and Libraries

- Built-in Python-libraries
  - Math
  - Time
- Self made functions and libraries
  - You can create your own building blocks

3rd party libraries

- Huge potential
- https://medium.com/activewizards-machine-learning-company/top-20-python-libraries-for-data-science-in-2018-2ae7d1db8049



Text files

Common procedure

- 3 Steps:
  - Open file
  - Do something with the file
  - Close file

```
file = open('my_file.txt', 'modus')
# do some stuff
file.close()
```

Text files

Different modes

- The mode defines how the content of the file should be treated
- Modes
  - 'r': read only
  - 'w': write only
  - 'r+': read and write
  - 'a': append

```
file = open('my_file.txt', 'mode')
```

Text files

Write

- The write() function is used to write something into a file
- '\n' is used to insert a line break

```
file = open('my_file.txt', 'a')
file.write('this is a new line')
file.write('this is another new line')
file.close()
```

Text files

Read

- A for loop can be used to read a file line by line
- line.strip() removes the trailing '\n'

```
file = open('my_file.txt', 'r')
for line in file:
    line = line.strip()
    print line
file.close()
```

CSV

Comma Separated Value

- Well known format for structured data
- Values separated by commas
- A new line is indicating a new record
- Movie example:

```
"41662","Pulp Fiction",1994,168,8000000,8.8,132745,4.5,4.5,4.5,4.5,4.5,4.5,4.5,14.5,24.5,44.5,"R",0,0,0,1,0,0,0
"41663","Pulse",1988,95,NA,4.7,246,4.5,4.5,4.5,14.5,24.5,14.5,14.5,4.5,4.5,4.5,"",0,0,0,0,0,0,0
"41664","Pulse: A Stomp Odyssey",2002,40,3000000,8.5,62,4.5,4.5,4.5,0,4.5,4.5,14.5,14.5,24.5,34.5,"",0,0,0,0,1,0,1
"41665","Pulso, O",1997,21,NA,9.3,27,0,14.5,0,0,0,4.5,0,4.5,34.5,44.5,"",0,0,0,0,0,0,1
"41666","Pump Up the Volume",1990,105,NA,6.8,6054,4.5,4.5,4.5,4.5,4.5,14.5,24.5,14.5,14.5,14.5,"",0,0,1,1,0,0,0
```

CSV

Read

- A for loop can be used to read a csv file line by line
- line.strip() removes the '\n' at the end of the line

```
import csv
csvfile = open(eggs.csv', 'rb') # the b is specific
spamereader = csv.reader(csvfile, delimiter='', quotechar='|'
for row in spamreader:
    print ,‘join(row)
```

Other questions

- Any other questions on your side about the python basics?

Exercise 1

- Read the movies.csv file
- Save the following features into a textfile
  - Number of movies in the data set
  - All the movies, starting With “Zero” their count
  - The average movie rating
  - The average vote count for a movie
  - The top 10 best rated movies having more than 5000 votes
  - The top 10 best rated movies which had a lower budget < 1 Mio USD
  - The top 10 best rated, with < Mio USD budget and > 5000 votes
  - Fiddle around with the data, do you find interesting things?

Exercise 1

Clues

- How to treat with the header line of the movies.csv file?
- Try to convert numbers when you read them from the file (using float() and int())
  - How can you test if a string contains a number
  - https://docs.python.org/3/library/stdtypes.html
  - If values are missing («N/A»), assign None as a value

Exercise 1

Clues

- If you do operations on ‘length’, ‘budget’, ‘rating’ or ‘votes’, make sure your values are correct (not None)
- Use a counter variable, in the for loop which counts the number of valid values

Exercise 1

Clues

- How to sort a list
  - https://wiki.python.org/moin/HowTo/Sorting
  - Have a look at the example

```
student_tuples = [
        ('john', 'A', 15),
        ('jane', 'B', 12),
        ('dave', 'C', 10)
]

sorted(student_tuples, key=lambda student: student[2]) # sort by age
[('dave', 'C', 10), ('jane', 'B', 12), ('john', 'A', 15)]
```