

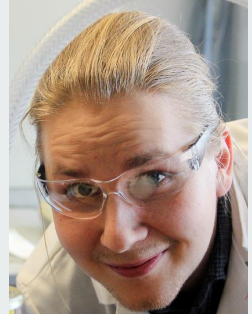


Databearbetning


Steget innan datavetenskap

Lektion 4 - Pandas och web scraping

Dennis Biström
bistromd@arcada.fi



Halva kursen har gått! - 4 lektioner, 4 uppg kvar



Upplägg

Föreläsningar med exempel - Var på plats, följ med!

Videoföreläsningar (60–120 min) att se på hemma

Veckouppgifter med deadline **varje** vecka.

Inget kodtilfälle! Använd F369 och fråga kaveri?

Kursverktyg

Python

Pandas (Python Data Analysis Library)

Jupyter Notebook

Installering: Anaconda (Linux / Mac / Windows)

<https://www.continuum.io/anaconda-overview>

Bedömning

Vitsordet bestäms på basis av era lösningar på kursuppgifterna. Maxpoäng 110p

Varje uppg är värd 20p. 3 förhör 10p, 3 läxor 10p

Bonus upp till 10p för smarta lösningar elr tilläggsfunktioner

5p avdrag per förseningsvecka

Närvaro

Jag använder mig av en närvarolista.

De som inte har deltagit på nån av de två första föreläsningarna blir borttagna från ASTA

<70% närvaro => begränsad klagomålsrätt

Upplägg - Checka lektion 7 o.O



Lektion 1 - Kursinfo, verktyg & resurser, Intro till Databearbetning. My first python app

Läxa 1 ut

Lektion 2 - Python Moduler och Klasser, My second and third app. Läxa 1 hjälp?

Förhör 1 ut, ~~Läxa 2 ut~~

Lektion 3 - Python Datastrukturer, Numpy & Matplotlib, Uppg 1 start

Förhör 1 in

Lektion 4 - Pandas, Uppg 1 forts

Läxa 2 ut, Uppg 1 ut

Lektion 5 - Visualisering, Webscraping & BeautifulSoup, Pandas, Uppg 2 start

Uppg 1 in, Uppg 2 ut, Förhör 2 ut

Lektion 6 - Förhör 2 (Numpy, Matplotlib & Pandas), Ljud och Bilder som data

Förhör 2 in, Uppg 3 ut

Lektion 7 - Inlämningsuppg 3 fortsättning, Övning med Bilder och Signaler

Uppg 2 in, Uppg 3 in, Uppg 4 ut

Lektion 8 - Inlämningsuppg 4. Kodande & Feedback, Julglögg?

Uppg 4 in

Hur många gjorde läxan? - Lite numpy



Python o Numpy: [Numpy - Charles K](#) Videorna 2.1 till 3.1

Recap av idag: [Numpy - Michail V](#) Videorna 4.2 till 5.4

[Bra Numpy resurser](#)

[Tutorialspoint Matplotlib](#)

Jag måst hålla koll på hur ni hänger med

Dåliga resurser är också mitt problem

Python in one slide? - Kom ihåg shift+tab

Python quirks - Indentation styr koden, försiktigt med mellanslag! Kolontecken efter if och else, *and or not*

Python - GPP, bygga webbsidor, analysera data, koda verktyg # Kommentar, även """ Multiline comment """

Variabler - behöver endast ett namn, tolken känner igen typen "Sträng" + str(int) + "."

Strings - "Text" eller 'strängar' **Escape chars** med \ för att skriva t.ex citattecken bland strängar.

Numror - Decimaltecken . | j för komplexa tal | int -> float -> complex. Tolk konv till bredare innan aritmetik.

Aritmetik - % modulo returnerar resten, // returnerar kvoten, ** fungerar som exponent.

Booleans - = för tilldelning (assignment), == för utvärdering. Efter str(True) går variabeln inte att använda i logik!

If elif else - raw_input("Mata in en sträng"). (Error handling) med try: except Error: + if else för input validation

Interaktiv hjälp:

dir() - Se vilka moduler, objekt, klasser och metoder ni har.

help(someObject.someMethod()) - Få tilläggsinformation om objekt, metod eller funktion

someObject.someMethod? - Visa docstring

jupyter-notebook - tryck tab 1-4 gånger för att utöka information om det ni håller på att skriva just nu

Moduler & Klasser - Encapsulation & Message Passing

Moduler - En modul innehåller python **Objekt**. Exempel på en moduler `__builtins__` eller `math`
Objekt i moduler kan innehålla **Klasser**. Många fungerar även som funktioner ex: `datetime.time(6,30)`
Objekt kan innehålla funktioner, som ofta tar emot **parametrar** ex: `math.cos(90)`

Metod = funktion, men vi kallar ofta funktioner inuti objekt för metoder ex: `myTimeVariable.isoformat()`
Metoder används för att kapsla in beteende. När den är inkapslad kan vi enkelt återanvända samma beteende.
Metoder kommunicerar med varandra genom parametrar och returvärden. Det här kallas Message passing

Klasser innehåller instruktioner om hur man skapar objekt (även funktioner och data). Data i objekt sparas i **fält**

Exempel:

<code>gamla_bettan</code> är en instans av klassen <code>bil</code>	<code>#klassen bil innehåller instruktioner över hur man skapar bilar</code>
<code>gamla_bettan.color</code>	<code>#Klassen bil innehåller även data som färg och märke</code>
<code>gamla_bettan.accel(10)</code>	<code>#Klassen bil innehåller även metoder, som tar emot parametrar</code>
<code>~/bistromd \$</code> 82 km/h	<code>#Returvärde för metoden accel() kunde vara hastigheten</code>

Datastrukturer i python



Sets([]) - Inspirerad av mattan, vi kommer int röra de här ordning

[Socratica Sets](#)

Inga dubletter

Ingen

[Pythonspot Set tutorial](#)

Tuples() - Immutable list

[Socratica Tuples](#)

Har ordning

[inLearning](#)

Mindre och snabbare än lists

[Pythonspot Tuple Tutorial](#)

Listor[] - Mutable list

[Socratica Lists](#)

Har ordning

[inLearning](#)

Kan ha dubletter

[Pythonspot List Tutorial](#)

Dictionary{} - Key:value pair

[Socratica Dictionaries](#)

Bekant från JS objekt?

[inLearning](#)

Ingen ordning

[Pythonspot Dictionaries](#)

Listor - Klurigheter och tilläggsmoduler



Listor - Från andra språk kanske bekant som arrays, i python kallas det här en lista: [1, 2.3, "hej"]

list[1][3:] - returnerar all värden efter det fjärde värde i den andra sublistan av list

Lägg till/modifiera eller ta bort värden:

list + ["new", 2.3] del(list[0]) list.append("hej")

Listor har metoder, liksom strängar. **Allting är objekt** men ha koll på ifall du gör string.index eller list.index

Vissa metoder ändrar på instansen list.reverse, andra skapar nya objekt med ändringarna gjorda list[5:6]

Moduler i form av bibliotek

Numpy för att jobba med **arrays** bl.a. Aritmetik över listor

Matplotlib för att **visualisera** data - Line, Bar, Pie, Histogram etc.

Pandas för att introducera **Data Frames** och därmed bredda listfunktionaliteten i Python

Numpy - flerdimensionella tabeller (arrays)

Installera numpy med pip - pip3 install numpy

import numpy - för att få access till **numpy.array(list)** ofta `import numpy as np` för att minska syntax

Python list to Numpy array

```
np.array([1,2,3,4,5])
```

Numpy Arrays kan sparas

```
np.save('data.npy',a)
```

Matematik över arrays

```
sin(np.array)
```

[inLearning Numpy Math](#) - Michele Vallisneri

[inLearning Numpy Slicing and Boolean Masks](#) - Charles Kelly

Numpy 2D array

```
np.array([[1,2,3,4] , [6,7,8,9]])
```

och läsas in till/från .npy filer

```
np.load('data.npy')
```

Aritmetik över arrays

```
y1 = sinx * cosx          y2 = cosx**2 - sinx**2
```

Matplotlib - Visualisering made easy

Matplotlib - `import matplotlib.pyplot as plt` # Använder pyplot paketet från matplotlib

Line och pie `plt.plot(x,y)` # `kind='bar'` `kind='barh'` `kind='pie'`

Färger - colormaps - sekventiell, divergent, kvalitativ `plt.plot(colormap='Pastell1')`

Scatter `plt.scatter(x,y)`

Skalor `plt.scale('log')`

Histogram - `plt.hist(data, bins=10)` Visualisera distribution av data # standard Python optional variable!

Anpassa graf - `plt.xlabel("X-axel")` `.title` `.yticks` # använd aritmetik för att förbättra det visuella meddelandet

Läs om flera alternativ för [pyplot.plot](#) och [pyplot.scatter](#)

[%matplotlib inline](#) - För att få matplotlib o funka inline i jupyter:

Numpy - betydligt färre for loops

Matematiska operationer över listor

```
numpy_bmi_array = list_of_weights / list_of_heights ** 2    #Bara en data type i array!
```

Märk också skillnad mellan `pylist + pylist` #konkatenering `np_array + np_array` #aritmetik

Array of booleans:

```
numpy_bmi_array > 20 returnerar en list av booleans:      [False,False]
```

```
numpy_bmi_array[numpy_bmi_array > 20] returnerar:      [ 24.20, 21,24 ] # Praktiskt!
```

2D numpy arrays: En förbättrad version av list of lists `array[0,10] * array[2,:]`

`array[row][column]` eller `array[row,col]` t.ex `array[2,3:5]` # Fjärde och femte kolumnen på tredje raden.

Numpy simple data analytics `np.mean()`, `np.median()`, `np.std()`, `np.corrcoef()`, `np.column_stack()`

Om du delar upp datan i två np.arrays, nycklar och värden, kan du hänvisa till index med endast nyckelvärden

```
positions = ['GK', 'M', 'A', 'D', ...]      heights = [191, 184, 185, 180, ...]
```

```
gk_heights = heights[positions=='GK']      # Superhändigt!
```

Idag - Pandas och web scraping



Python, Numpy och Matplotlib recap från senast

Exercises.ipynb ifall inte redan gjort - Öppna t.ex pythonspot.com

Om du redan gjort, läs Lists-Functions på [Tutorialspoint Python 3 tut](#) (obs Datetime)

~~Gör övningar på [Datacamp](#)~~

~~Ch1 Python basics~~

~~Ch2 Python lists~~

~~Ch3 Functions and packages~~

~~Ch4 Numpy~~

Data Analysis and Visualization på [DataQuest](#)

Course 1 - Numpy

Course 2 - Matplotlib

Pandas

Pandas - Series



High-performance data manipulation and analysis tool because of its series, dataframes and [panel](#) additions. Before Pandas, Python was majorly used for data munging (cleaning) and preparation but not data analysis. With Pandas, we can load, prepare, manipulate, model, and analyze data, regardless of its origin.

Series 1D en datatype, fixed storlek

10	23	56	17	52	61
----	----	----	----	----	----

Numpy array to Pandas series

```
pd.Series(np.array([1,2,3,4,5]))
```

Accessing data from Series

```
S[:3] #elem or s['a'] #label
```

Python dict to Pandas series

```
pd.Series({'a': 0., 'b': 1., 'c': 2.})
```

Accessing multiple elements from series

```
s[['a','c','d']] #using label
```

[inLearning Pandas Series](#) - Michele Vallisneri

[inLearning Pandas Essential Training](#) - Series med Jonathan Fernandes

Pandas - DataFrame hmm bekant från R?

2 dimensionella datastrukturer för flera datatyper, ändrande storlek, labels

Python 1D List till DataFrame

```
pd.DataFrame([1,2,3,4,5])
```

2D List till DataFrame

```
pd.DataFrame( [['Alex',10],['Bob',12],['Clarke',13]] , columns=['Name','Age'] )
```

Python Dict till DataFrame

```
pd.DataFrame({'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]})
```

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

[inLearning Pandas DataFrames](#) - Michele Vallisneri

[inLearning Pandas Essential Training](#) - DataFrames med Jonathan Fernandes

Movie time! - DF, VC, Bool, indexing, loc, [groupby](#)



Python, Numpy och Matplotlib recap från senast

Exercises.ipynb ifall inte redan gjort - Öppna t.ex pythonspot.com

Om du redan gjort, läs Lists-Functions på [Tutorialspoint Python 3 tut](#) (obs Datetime)

Data Analysis and Visualization på [DataQuest](#)

Course 1 - Numpy

Course 2 - Matplotlib

Pandas

[inLearning Pandas Essential Training](#) - **DataFrames med Fernandes**

Tips: Kellys inLearning sen innan ni börjar på uppg 1 (sätt captions & läs transcripten om ni int har hörlurar)

Pandas Intro Demo

Good reads Uppgift

Pandas - Notes från Fernandes & Kelly Ch2



Dataframe - 2D array like from numpy

Series - 1d array of indexed data (col 1)

DataFrame['Series'] - Access series in dataframe. #Different functions for DF and Series, do type(obj)

DataFrame[['series1', 'series2']] - Access several series from dataframe # Märk att svaret är en ny DF

Data Input - Stöd för read_csv, read_excel, read_json, read_sql_table

DataFrame.shape - tuple for confirming dataframe dimensions

DataFrame.head() and **DataFrame.tail()** - visar första eller sista raderna från DF, mycket praktiskt

DataFrame.info() översikt för DF, märk datatyper!

DataFrame.describe() ger dig counts och mean min max quartiles

DataFrame.T står för transpose och gör kolumner till rader

DataFrame.loc[:,['A','B']] - Index är tillåtet, pandas förstår också sig på datum, **loc()** för att välja enligt label

[inLearning Pandas Selection](#) 5 min framåt

Pandas - Kelly Ch2 bredare & djupare än Fernandes



DataFrame - 2D array like from numpy

Series - 1d array of indexed data (column) # en 1D DF "med en col" ser ut som en rad, don't be fooled

DataFrame['Series'] - Access series in dataframe. # Different functions for DF and Series, do type(obj)

DataFrame[['series1', 'series2']] - Access several series from dataframe # Märk att svaret är en ny DF

Data Input - Stöd för read_csv, read_excel, read_json, read_sql_table

DataFrame.shape - tuple for confirming dataframe dimensions

DataFrame.head() and **DataFrame.tail()** - visar första eller sista raderna från DF, mycket praktiskt

DataFrame.info() översikt för DF, märk datatyper!

DataFrame.describe() ger dig counts och mean min max quartiles

DataFrame.T står för transpose och gör kolumner till rader

DataFrame.loc[:,['A','B']] - Index är tillåtet, pandas förstår också sig på datum, **loc()** för att välja rad enligt label

[inLearning Pandas Selection](#) 5 min framåt

Data analysis - Lite råd för uppgiften



df.year.value_counts(dropna=false) - hur många värden i fallande ordning #hur många filmer per år sort desc

df.sort_values(by=['rating','title'], inplace=false) - sortera enligt rating, sedan filmtitel, skriv inte över

df[(df.oscar >= 1) & (df.rating >= 3.4)] - Boolean indexing #Visa endast top rated oscarfilmer

- Flera krav inom parenteser, and operand & (visst minns ni?)

df.str.genre.contains("Horror") - startswith(), isnumeric() # Visa horror filmer

Querying Data Frames - Ett par övningar ([Manipulating DF w Pandas](#) [Pandas Foundations](#))

Ex:

```
filtered = df[ (df.genre == "Horror") & (df.oscar == 1) ] # Visa horrorfilmer som fått oscars
```

```
filtered.sort_values('oscar', ascending=false) # sort by oscars desc.
```

```
filtered[ ['rating', 'title'] ] #Visa bara 2 col
```

Idag - Pandas och web scraping



Pandas

Fernandes videon Kolla Kellys videon innan ni börjar på uppg 1

Pandas Intro Demo

Good reads Uppgift

Labb 2 från Harvards kurs CS109 Data Science som

Videon "Lab 2: Scraping, Pandas, Python, and viz" går igenom den här filen.

Länken till videon: <http://cs109.github.io/2015/pages/videos.html>

BeautifulSoup - Intro till Data Scraping



Vilka produkter finns på newEgg?

Exempel: Vi gör tillsammans ^.^

my_url = "<https://www.newegg.com/Video-Cards-Video-Devices/Category/ID-38?Tpk=graphics%20card>"

För kommande Scrapinguppgift vore det bra att:

Läsa: [Intro till BeautifulSoup](#)

Sedan: Läs [dokumentationen](#)

Resurser för Uppg 1! inLearning videor är lättast?



Pandas Essentials - [Kapitel 2 till 7 \(Fernandes\)](#)

- Lättsam o räcker nog för att klara Uppg 1

Pandas for data science [Kapitel 2, 4 och 5 \(Kelly\)](#)

- Här från en mer datavetenskaplig approach (dat voice)

Python data analysis - [Kapitel 6 & 7 \(Vallisneri\)](#)

- Baby Names excersizen e bra

Resurser för Uppg 1! Interaktivt på DataCamp?




[Data ingestion & inspection](#) - Pandas foundations på DataCamp

- Inspecting & Datatypes
- Labeling and Broadcasting
- Importing Exporting
- Plotting

[Extracting and transforming data](#) - Manipulating DataFrames på DC

- Indexing & Slicing
- Subselecting & Filtering
- Transforming & Mapping

Lynda och resurser - Kolla även itslearning!

- 
- Cheat sheet: [Anaconda Cheat Sheet - Getting Started - PDF](#)
[Pandas Cheat Sheet - PDF](#)
- Manual/Docs [Conda package manger - Docs](#)
[Pandas - QuickStart & Cookbook](#)
- Tutorials (text) [Anaconda Getting Started - User Guide](#)
[Python - Intro till avancerat - Övningar och förklaringar](#)
[Pandas tutorial - PythonSpot](#)
[Intro to data science Numpy, MatPlot & Panda](#)
[\(Pandas - How do pivot tables work - ExcelCampus\)](#)
- Tutorials (video) [Socratica python tutorial - Youtube](#)
[Derek Banas - "Learn Python in one video"](#)

Lynda och resurser2 - Kolla även itslearning!



Interaktiva:

[Intro to **python** for data science - Gratiskurs - DataCamp](#)

[Intro to python for data science - Ch4 - **Numpy** \(DataCamp\)](#)

[Intermediate python for data science - Ch1 - **MatPlotLib** \(DataCamp\)](#)

Lynda:

[6h nybörjarkurs **Python** för datavetenskap med Lillian Pierson - Lynda](#)

[2h intermediate - **Numpy** Data Science Essentials - Charles Kelly](#)

[Intermediate - Ch3: **Numpy**, Ch4: **Pandas**, Ch 9: **matplotlib** - Miki Tebaka](#)

[2h intermediate kurs i **Pandas** med Jonathan Fernandes - Lynda](#)

[2h intermediate **Pandas** för Datavetenskap med Charles Kelly - Lynda](#)

[Big Data Analysis in python using **Numpy** and **Pandas** - Michele Vallisneri](#)

Lynda och resurser3 - Kolla även itslearning!



Interaktiva roligheter

[The Python Challenge](#)

[Roliga övningar i logisk ordning - Practice Python](#)

[How to think like a Computer Scientist](#)

[CodeSignal - Interaktiva utmaningar, badges, points etc.](#)

[Reddit daily programmer challenges](#)

Vill du vinna 1 miljon \$

[7h gratis tutorial på Kaggle - Känner ni till kaggle?](#)

Python 2 vs Python 3 trubbel, kolla [här](#)

Jag har redan fallit av kälken! - Brush up ur skills



1. Kolla [Socratica tutorialen](#) videorna 1-17
2. [Chapter 1-3 - Python for data science \(DataCamp\)](#)
3. [Chapter 1 - Writing python functions \(DataCamp\)](#)
4. [Chapter 4 - Numpy \(DataCamp\)](#)
5. [Chapter 1 - Matplotlib \(DataCamp\)](#)
6. [Chapter 1-4 - Pandas for data science \(Lynda: Kelly\)](#)

Sök hjälp bland resurserna om du kör fast.