

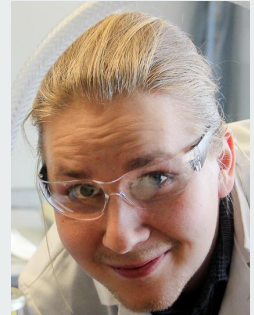


Databearbetning

Steget innan datavetenskap

Lektion 6 - Visualisering recap, Förhör och Inlämningsuppg 3

Dennis Biström
bistromd@arcada.fi



2 veckor kvar! - 3 lektioner, 3 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. 1 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 - Hårdaste 10 dagarna i kursen



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, Uppg 3 ut
Lektion 6 - Visualisering forts. Matplotlib med textfiler, Ljud och Bilder som data	Uppg 2 in 21.10 kl 16.59
Lektion 7 - Inlämningsuppg 3 fortsättning, Övning med Bilder och Signaler	Uppg 3 in 28.10 kl 16.59
Lektion 8 - Inlämningsuppg 4. Kodande & Feedback, Glögg på cornern? 1.11 elr 8.11?	Uppg 4 in 4.11 kl 16.59

Python in one slide?



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. *Se upp!* `2**(1/2)` = ?

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 betende.
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>

Python Listor & Moduler för att bredda python

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")

Märk att y = x inte kopierar värden. För att initiera en ny lista y med värden från x, gör y = list(x) eller y = x[:]

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

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

Moduler i form av bibliotek

Numpy för att jobba med arrays (alltså listor men inte python listor :S) 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

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

Även möjligt att köra `t.ex from numpy import array` # försiktigt!

Numpy & np.array - 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!
```

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 praktiska metoder



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
```

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='Pastell')`

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:

Inlämningsuppg 2 - Steam Sales

Senast: hitta grafikortens element på newegg.com

```
In [13]: from bs4 import BeautifulSoup # Import BeautifulSoup
from urllib.request import urlopen # Import urlopen
soup = BeautifulSoup(urlopen('http://store.steampowered.com').read()) #make soup
containers = soup.findAll("div", {"class": "discount_final_price"}) #findALL containers
print("Sale Price: " + containers[0].text)
```

Sale Price: 4,99€

Lab 2: **List comprehension:** for loop och list creation oneliner @ 13:30

```
[t["class"] for t in soup.find_all("table") if t.get("class")]
```

Lab 2: **Lambda uttryck:** returnerar värde av uttrycket inom sig @ 21:00

```
rem_nl = lambda s: s.replace("\n", " ")
```

BeautifulSoup - Bra snabbstartresurser



1. [Intro to BeautifulSoup](#) - 10 min quickstart ifall ni missa newegg
2. [Intro till BeautifulSoup \(Text\)](#) - Python for beginners
3. [Docs för BeautifulSoup](#)
4. [Newegg Grafikkortsexempel](#)
5. **Labb 2** - Numpy, Matplotlib och Pandas
Harvards kurs CS109 Data Science
<http://cs109.github.io/2015/pages/videos.html>
6. **Inlämningsuppg 2** - Steam Sales
<https://store.steampowered.com/search/?specials=1&os=win>

Matplotlib - Snabbstartresurser i visualisering



1. [Chapter 1](#) - DataCamp - *Doit*
2. [Picking the right graph for your data](#) - Olika grafer för olika data - *Sen*
3. [Basic Plotting](#) - MatPlotLib & Pandas
4. [Inline Plotting](#) - Charles Kelly *mwah super bass*
Avancerade grafer, flera linjer etc
5. [Chapter 2](#) - Line Bar and Pie Plots (även intro to seaborn) [Lilian Pearson!](#)
På djupet standardgrafer

Rostigt med lådagram? - Recap från d3 kursen



[Online Akademin - MatteCentrum](#) - Vad är ett histogram

[Online Akademin - MatteCentrum](#) - Vad är lådagram

[Matteboken - Kvartiler och Lådagram](#) - Statistik repetition

[Matteguiden - Spridningsmått, lådagram](#) - Statistik repetition

Vi lyssnar i klassen

[Tidsserier - Emma Saunders](#) - Kontinuerlig eller diskret data? Line Bar Area Dot elr Candlestick?

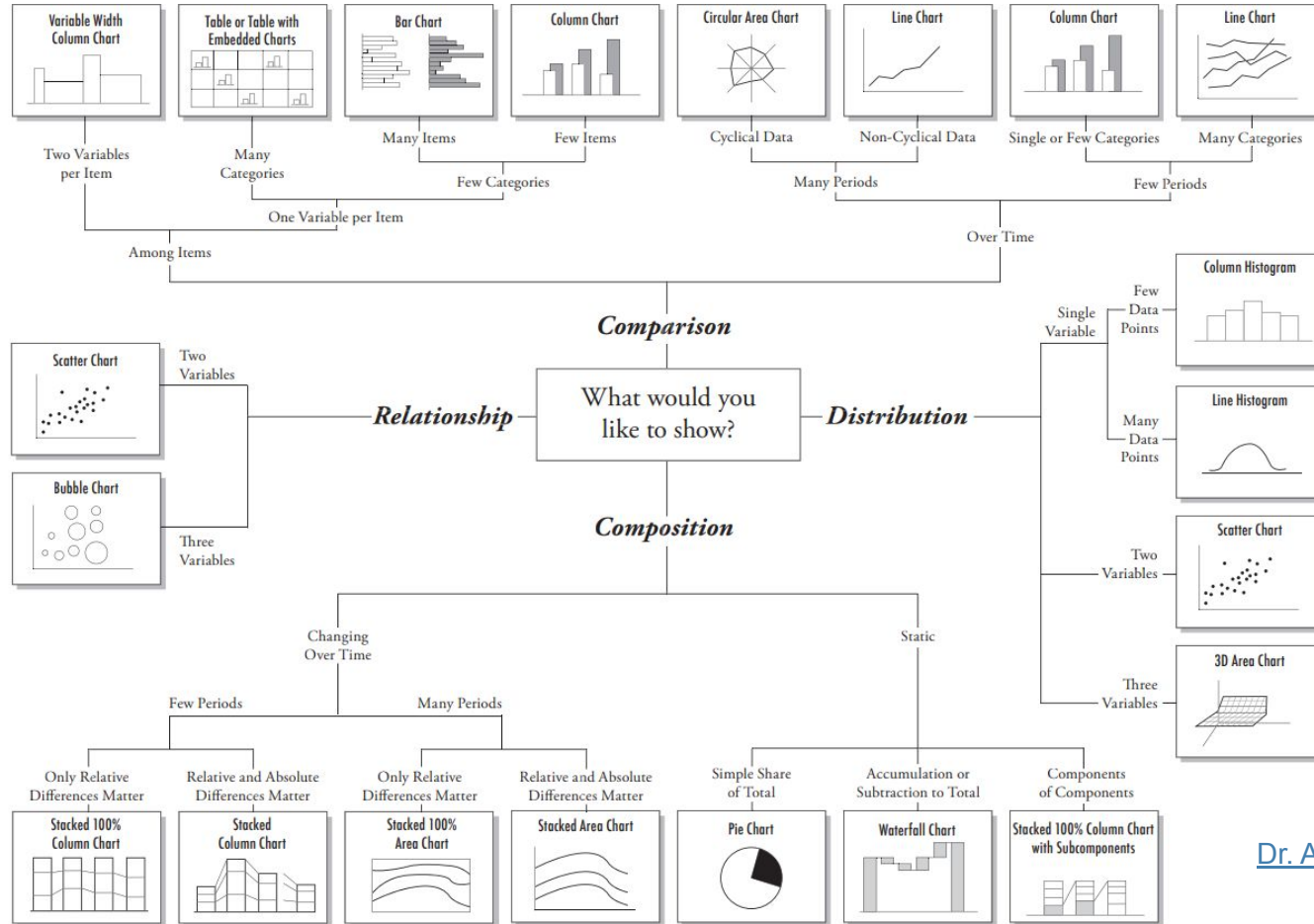
[Distribution - Emma Saunders - Lynda](#) - Scatter för regression (samband), Histo, Box, Overlays

[Hierarisk Data - Emma Saunders - Lynda](#) - Sunburst, Träd eller Trädkarta (Treemap)

[Geografisk och Text - Emma Saunders - Lynda](#) - Heatmap, Spotmap, Tubemap (Ta bort geografin)

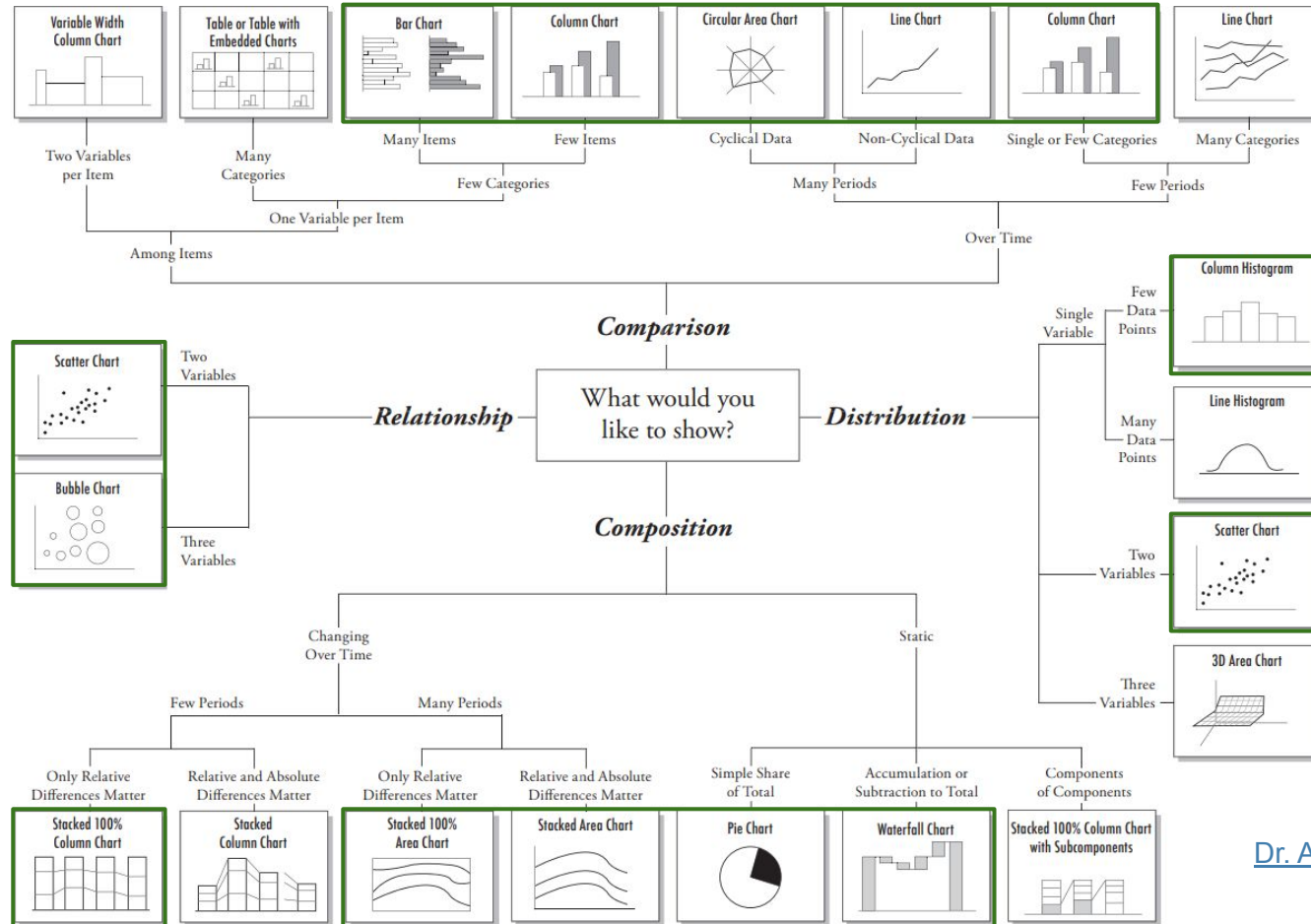
WordCloud, Occurance Matrix, Chord

The Chart Chooser



Dr. Andrew Abela

Även: The Slide Chooser



Dr. Andrew Abela

Visualisering

[Andrew Gelman - Why tables are better than charts](#) (1:a april - "my discussion above is serious.")

"Graphs are a way of implying results that are often not statistically significant"

En välstrukturerad tabell är ärlig.

Vi är vana med tabulär data och har inga problem att uppfatta förhållanden. Använd för presentation

Alla grafer uppmuntrar eller insinuerar till slutsatser. Använd för att övertyga

Gross National Happiness i Bhutan

40.8% of people in Bhutan have achieved happiness.

The GNH Index requires an array of conditions to be met.

Those who are happy enjoy it in 56.6% of the domains.

Happiness (GNH) is reached when people reach sufficiency in roughly half of the domains.

Källa: [World Happiness Report \(2012\)](#)

[Årets rapport](#)



Visualisering

Andrew Gelman - Why tables are better than charts (1:a april - "my discussion above is serious.")

"Graphs are a way of implying results that are often not statistically significant"

En välstrukturerad tabell är ärlig.

Vi är vana med tabulär data och har inga problem att uppfatta förhållanden. Använd för presentation

Alla grafer uppmuntrar eller insinuerar till slutsatser. Använd för att övertyga

Figure 11: GNH index by gender

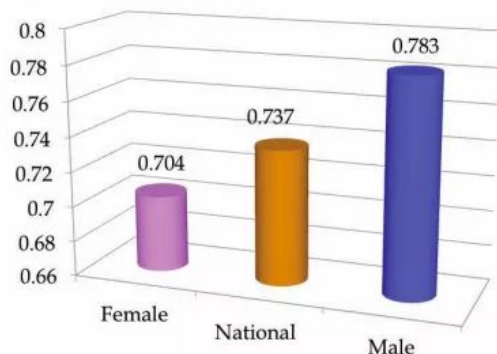


Figure 11

When we decompose the GNH Index by gender we see that men are happier than women.

49% of men are happy, while only one-third of women are happy.

Gender	Happiness
Men	49 %
Women	33 %

Tabellen

Remove
to improve
the **data tables** edition

Visualisering

Andrew Gelman - Why tables are better than charts (Skriven första april)

“Graphs are a way of implying results that are often not statistically significant”

En välstrukturerad tabell är ärlig.

Vi är vana med tabulär data och har inga problem att uppfatta förhållanden. Använd för presentation

Alla grafer uppmuntrar eller insinuerar till slutsatser. Använd för att övertyga

Figure 4: In which domains do happy people enjoy sufficiency?

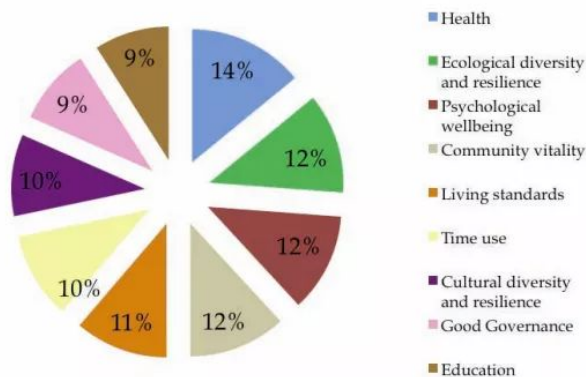


Figure 4

Shows in which domains happy people enjoy sufficiency.

We can see that all nine dimensions contribute to GNH

Happy people live relatively balanced lives.

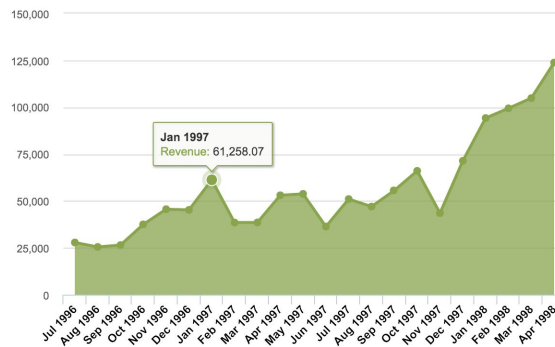
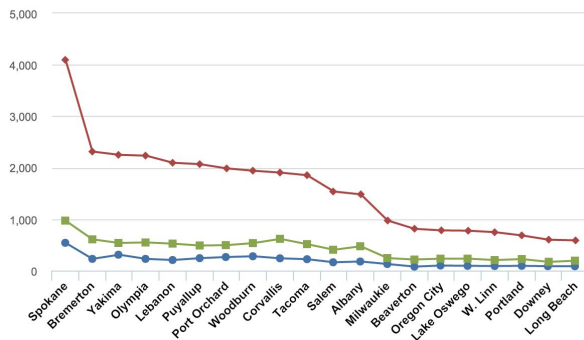
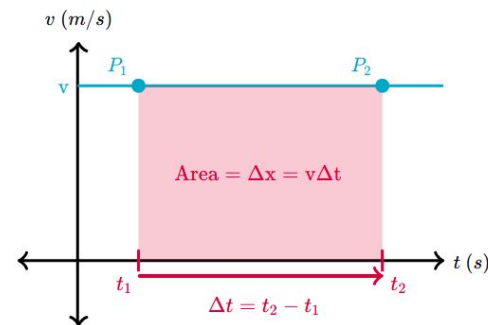
Pajdiagram

Remove
to improve
the **pie chart** edition

Vilken graf - Kontinuerlig data

1. Är det en tidsserie du vill visa?
 - a. Är det numerisk data?
 - i. **Kontinuerlig data?** Ex temperatur

Linje eller Area - Fördelen med area är att integralen kan ge mervärde



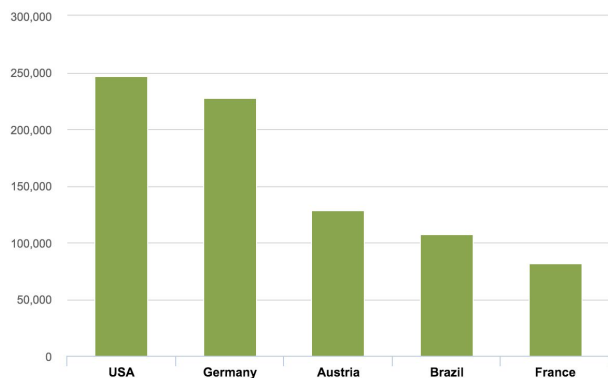
Vilken graf - Diskret data

1. Är det en tidsserie du vill visa?

a. Är det numerisk data?

i. **Diskret data?** Försiktigt med sampel och medeltal

Barchart eller Candlestick - Fördelen med candlestick är att man kan illustrera variation



Och förutspå framtid?

Barchart

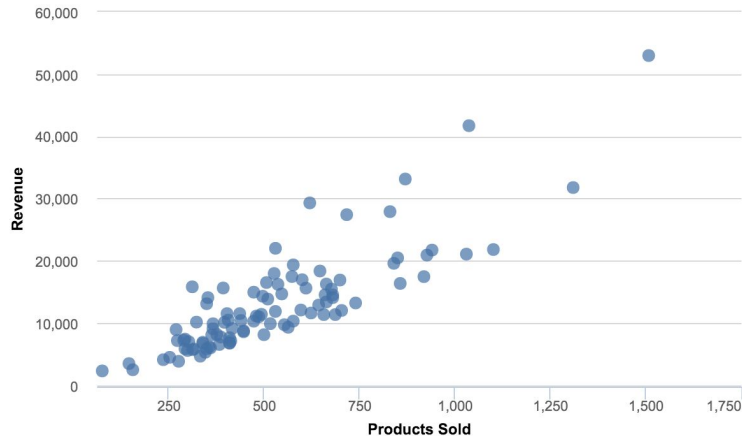
Remove
to improve
(the **data-ink** ratio)

Vilken graf - Korrelation

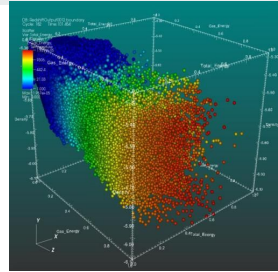
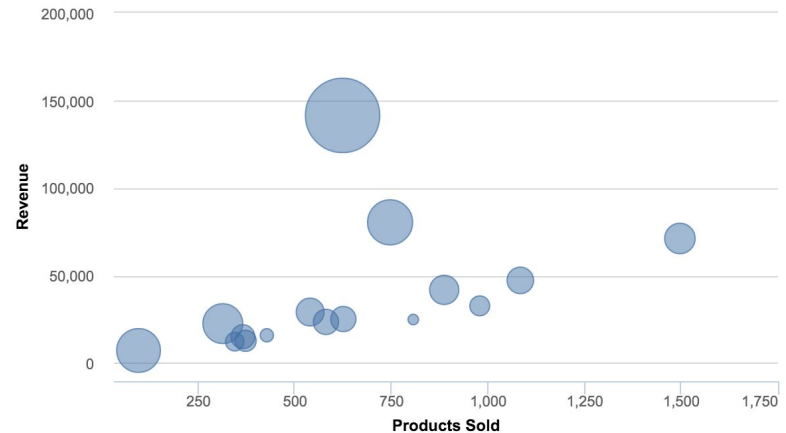
2. För korrelation och distribution (men även det snabbaste sättet att få en insikt i din data)

Scatter aka punktdiagram, även spridningsdiagram eller sambandsdiagram

Illustrerar även outliers eller gruppering



Få en insikt i samband mellan upp till fyra variabler



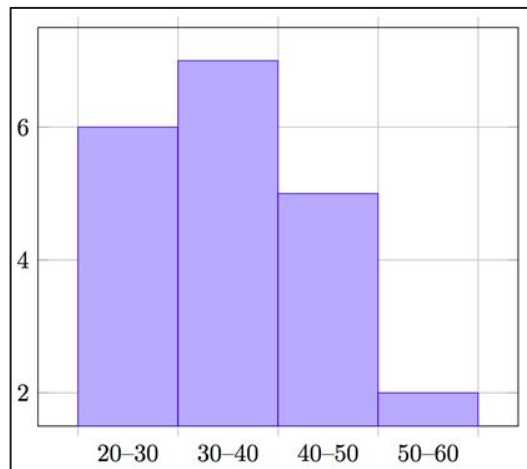
Histogram - Distribution

(även för fler än 7 staplar)

Histogram - Används för att åskådliggöra en distribution med många värden

Exempel: Åldersfördelning, Längdfördelning

Ingen point att säga "Av 30 arbetare finns det 1 som är 20år, 2 som är 21år, 1 som är 22 år..."



Klasser - På x-axeln

Skapa de här med en for loop och en if sats

Frekvens - På y-axeln

Skapa en counter tabell som räknar hur många gånger vi faller inom klasserna.

Exempel: `df.describe()` men visuellt!

[Matplotlib.pyplot.hist](#)

Läxa: [Läs den här sidan](#)

Låddiagram - Are you a part of the 50th %ile

Låddiagram - Baserat på tre mått

1. Variation/Spridning: Maxvärde - minvärde
2. Kvartiler/Fjärdedelar: Beskriver spridningen kring medianen

“Kvartilavstånd = 50% av värden →

1. Percentiler:

Till P33 hör värden 1,3,4,8,15

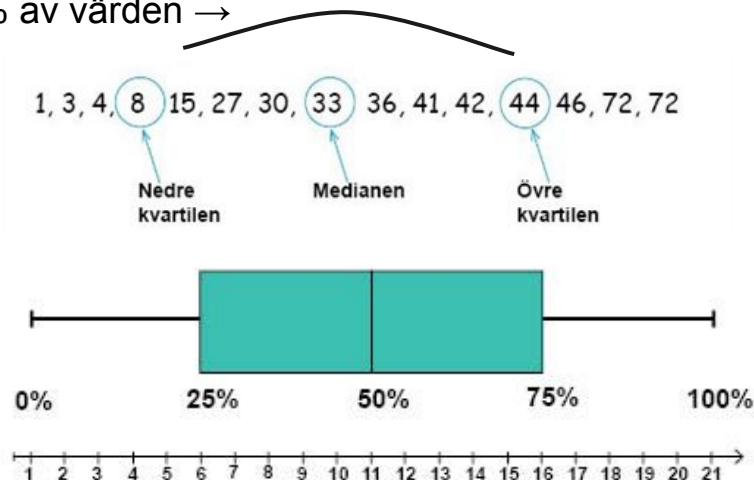
Q1 - P25 (alltså neråt!)

Q2 & Q3 - Medianen P50 (upp och ner!)

Q4 - P75 (alltså uppåt!)

Läxa: Läs, ladda ner demofilen, lek med den

[Pyplot.subplots, axis.boxplot](#)



Okej enough talk - Visualiseringsövning!

Demo: Numpy och Matplotlib, läsa från txt fil, lite prepping för inlämningsuppg 3

Sen:

03PythonPandasScrape.ipynb - Uppdaterad och förbättrad by me ;)

04NewEggScrape-Läxa2.ipynb - Nu med video

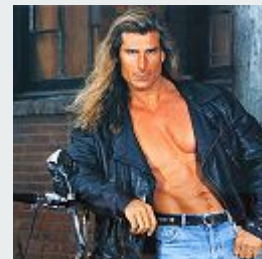
Inlämningsuppg 2 - Steam sales

Inlämningsuppg 3 - Moores Law

Kämpa hårt idag kanske ni får lite

Inlamn3 Tips.ipynb :)

Kodtillfälle/Glöggtilfälle?



Nästa gång: Ljud, Ljus & Bilder som data


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.

Läxor: Förhör på innehållet coming soon

- 
- Step 1: [Intro to python for data science](#)
[Chapter 4 - Numpy \(DataCamp\)](#)
- Step 2: [Intermediate python for data science](#)
[Chapter 1 - Matplotlib \(DataCamp\)](#)
- Step 3: [Pandas Essential Training -> Kapitel 6 \(Fernandes\)](#) - Ytlig?
[Pandas for data science -> Kapitel 5 \(Kelly\)](#) - Långsam?
- Step 4: [Data Exploration, Distribution analysis,](#)
[Categorical variable analysis, Data Munging](#)

Hur långt har ni kommit? Ointressanta resurser är även mitt problem..

Jag har en övning till - Vi börjar tillsammans



Som conveniently är en del av inlämningsuppg 3 :)

Inlämningsuppg 3 - Del 1

Moore's lag säger att antalet transistorer i en mikroprocessor fördubblas ungefär varje 2 år.

[Moore's Law over 120 Years](#)

Stämmer det?


Ta in data från https://en.wikipedia.org/wiki/Transistor_count om år och transistorantal

(OBS! scraping, tvättning, sorting av data behövs som vanligt), och rita en graf av transistorantalens utveckling.

Använd logaritmisk skala på y-axeln, och sätt in en linje som visar vad ökningen borde vara enligt Moore's lag.

Märk ut några valda punkter med processorns namn.

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?](#)

Interaktiva, bra helheter

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