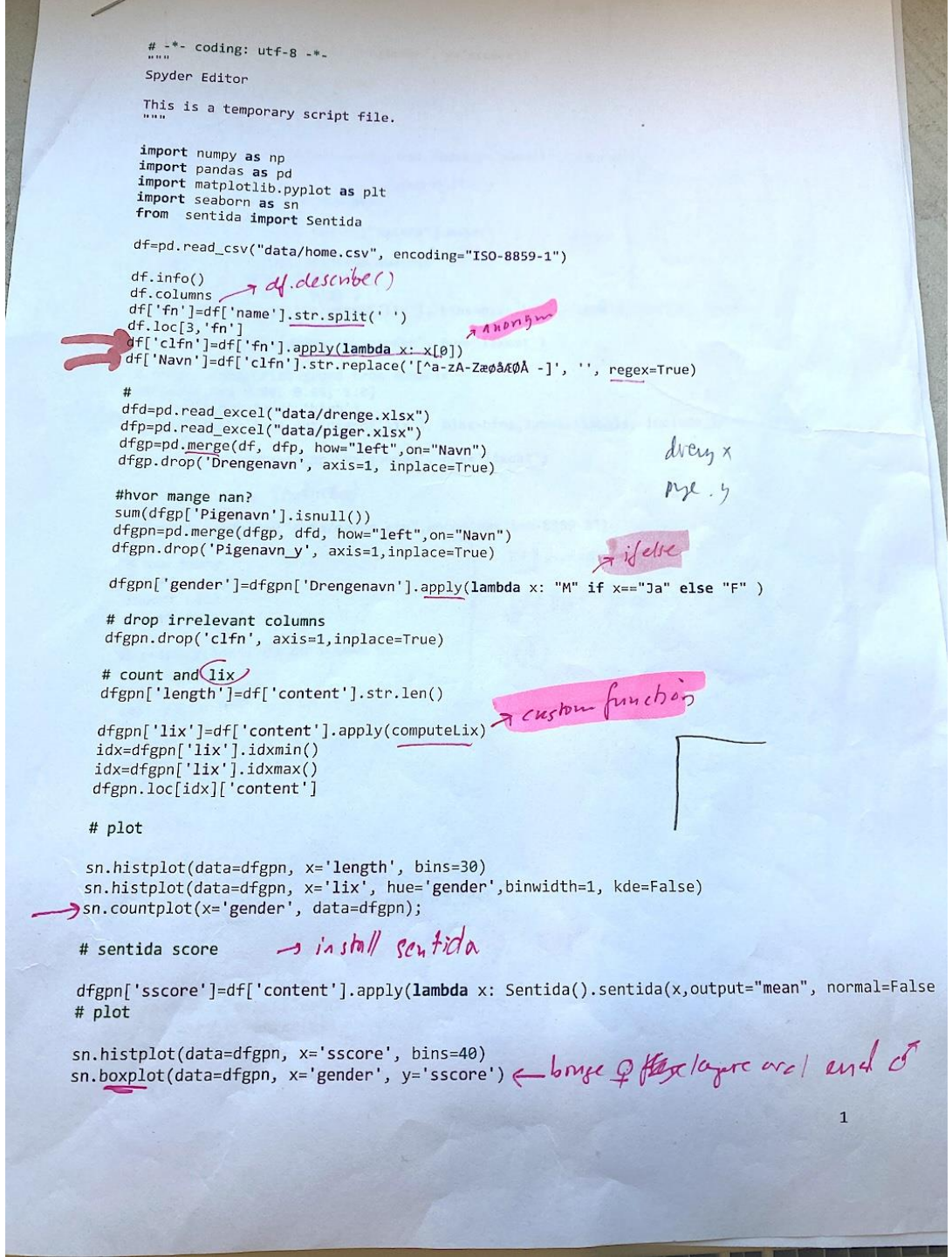
1. Recap fra i går
2. Python primer
    1. Værktøjer
    2. Quiz / Titanic
        1. Gennemgang
        2. Jeres egen vidensprep
3. Pentagon Papers



```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
from  sentida import Sentida

df=pd.read_csv("data/home.csv", encoding="ISO-8859-1")

df.info()
df.columns
df['fn']=df['name'].str.split(' ')
df.loc[3,'fn']
df['clfn']=df['fn'].apply(lambda x: x[0])
df['Navn']=df['clfn'].str.replace('[^a-zA-ZæøåÆØÅ -]', '', regex=True)

#
dfd=pd.read_excel("data/drenge.xlsx")
dfp=pd.read_excel("data/piger.xlsx")
dfgp=pd.merge(df, dfp, how="left",on="Navn")
dfgp.drop('Drengenavn', axis=1, inplace=True)

#hvor mange nan?
sum(dfgp['Pigenavn'].isnull())
dfgpn=pd.merge(dfgp, dfd, how="left",on="Navn")
dfgpn.drop('Pigenavn_y', axis=1,inplace=True)

dfgpn['gender']=dfgpn['Drengenavn'].apply(lambda x: "M" if x=="Ja" else "F" )

# drop irrelevant columns
dfgpn.drop('clfn', axis=1,inplace=True)

# count and lix
dfgpn['length']=df['content'].str.len()

dfgpn['lix']=df['content'].apply(computeLix)
idx=dfgpn['lix'].idxmin()
idx=dfgpn['lix'].idxmax()
dfgpn.loc[idx]['content']

# plot

sn.histplot(data=dfgpn, x='length', bins=30)
sn.histplot(data=dfgpn, x='lix', hue='gender',binwidth=1, kde=False)
sn.countplot(x='gender', data=dfgpn);

# sentida score

dfgpn['sscore']=df['content'].apply(lambda x: Sentida().sentida(x,output="mean", normal=False
# plot

sn.histplot(data=dfgpn, x='sscore', bins=40)
sn.boxplot(data=dfgpn, x='gender', y='sscore')
```

TIDYTEXT

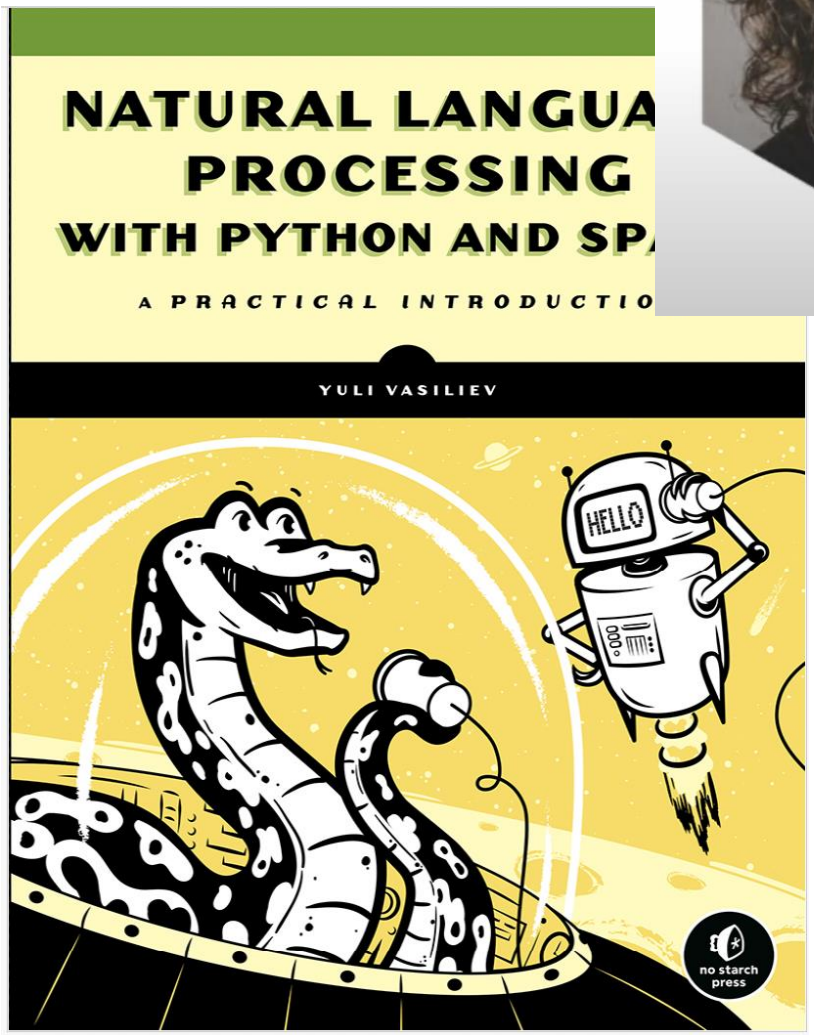
HELLO

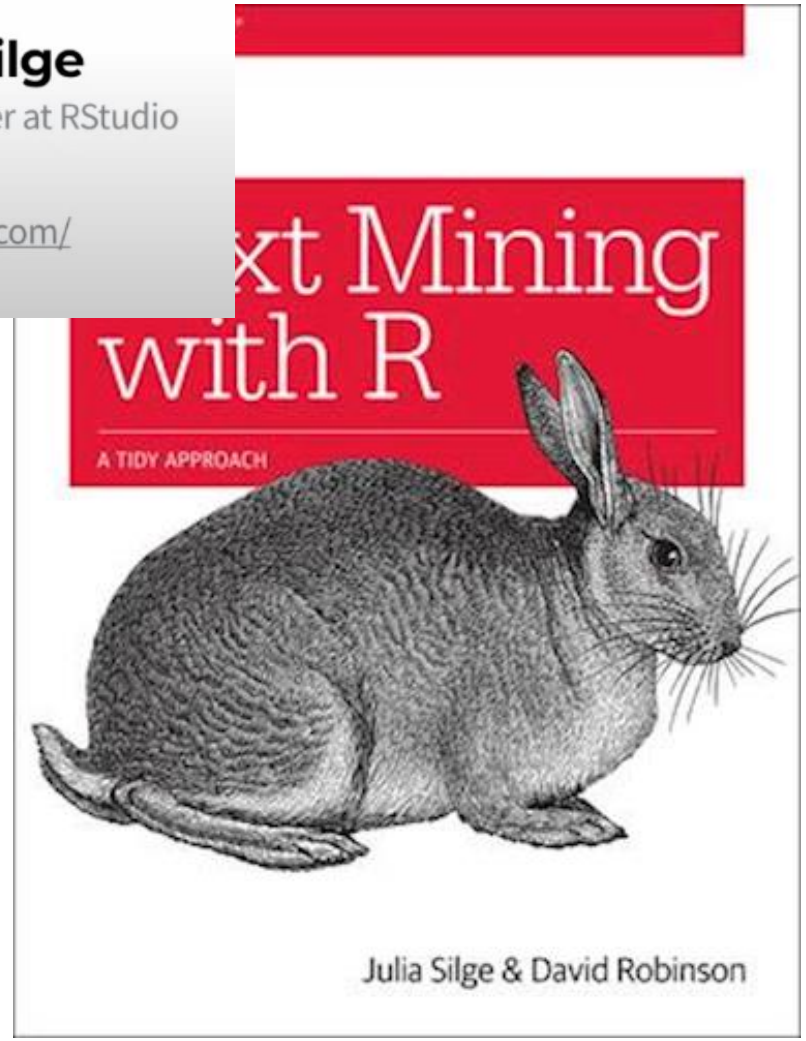
I'm Julia Silge

Software Engineer at RStudio

@juliasilge

https://juliasilge.com/

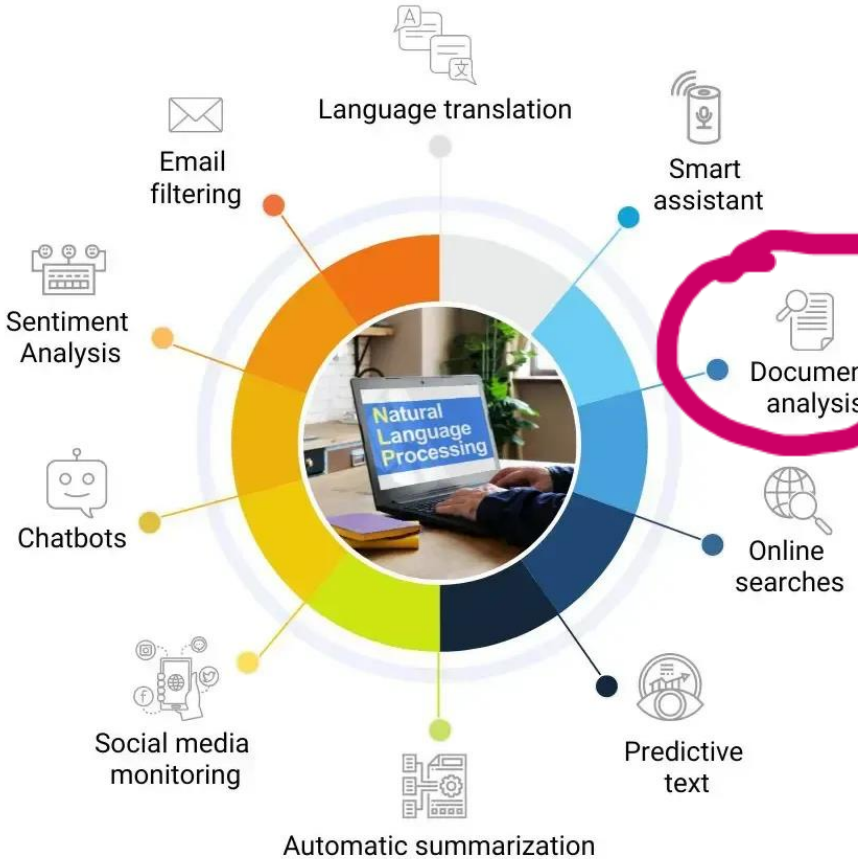


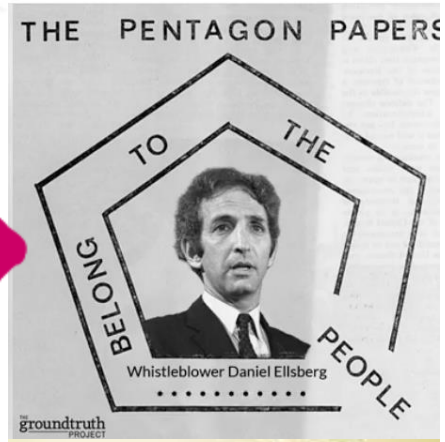


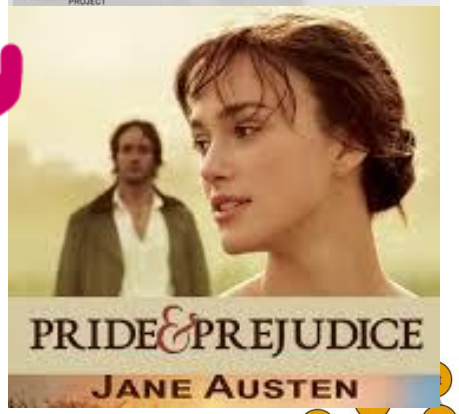
Planen

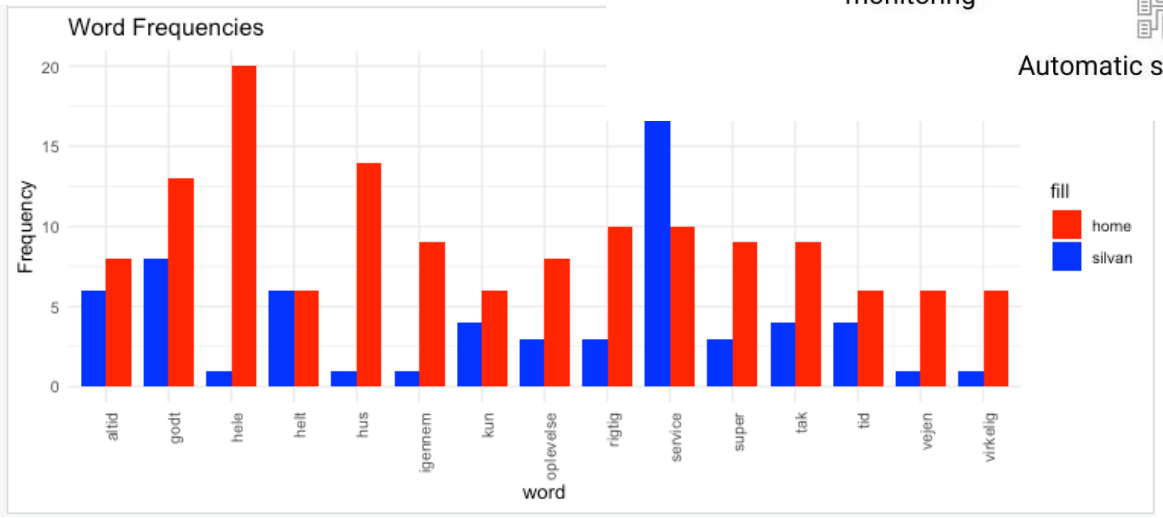
SILVAN

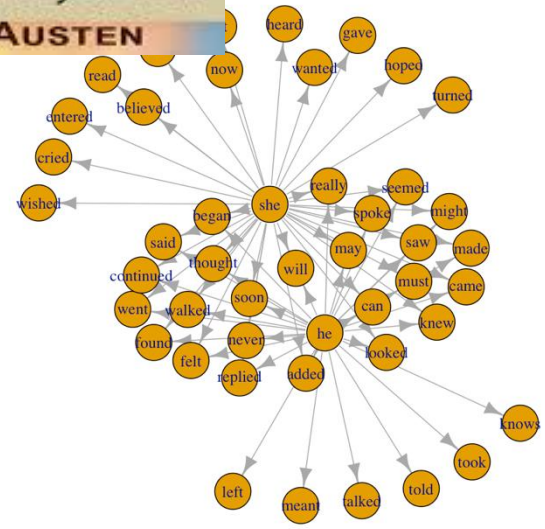
jem & fix
LAVPRIS BYGGEMARKED



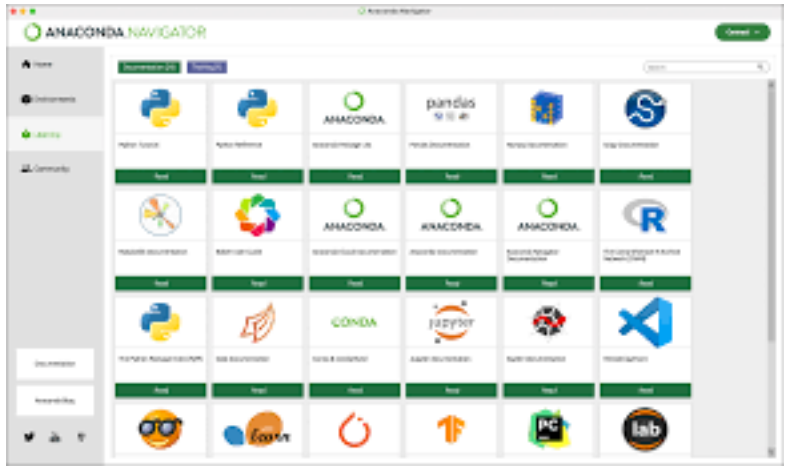






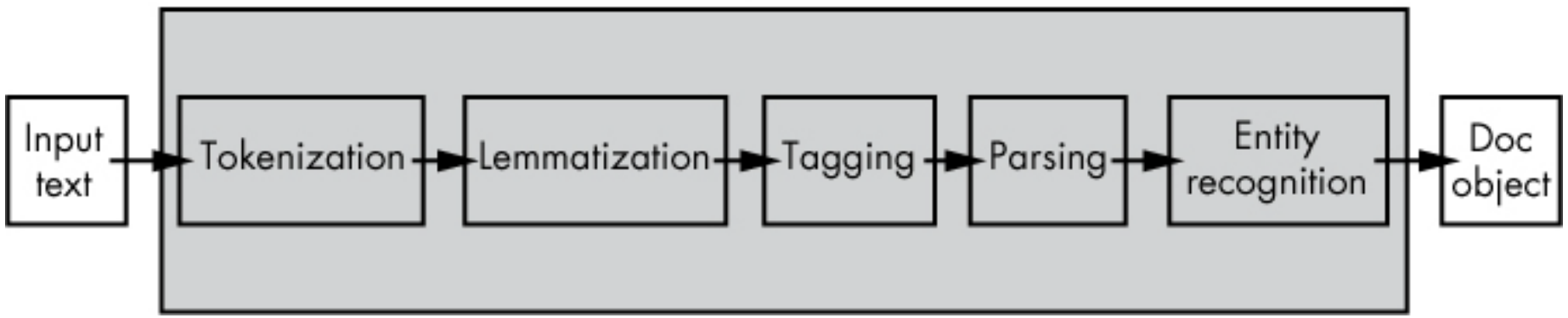


Værktøj


ANACONDA.NAVIGATOR
pandas
CONDA
jupyter


spaCy
NLP

Pipeline


Input text
Tokenization
Lemmatization
Tagging
Parsing
Entity recognition
Doc object

Advarsel

Hvad skete der med Biblioteket i Alexandria?

©Getty Images

Det Store Bibliotek i Alexandria var en af de mest ambitiøse intellektuelle bedrifter i menneskets historie – et sted hvor viden ingen grænser havde, og jagten på visdom blev betragtet som hellig. Oprettet i det 3. århundrede f.v.t. i den blomstrende by Alexandria i Egypten, var det langt mere end blot et bibliotek. Det var et centrum for læring, et kulturelt samlingspunkt og et fyrtårn for oplysning, der tiltrak datidens største tænkere, videnskabsfolk og filosoffer fra hele den kendte verden.

En vision om fremtiden

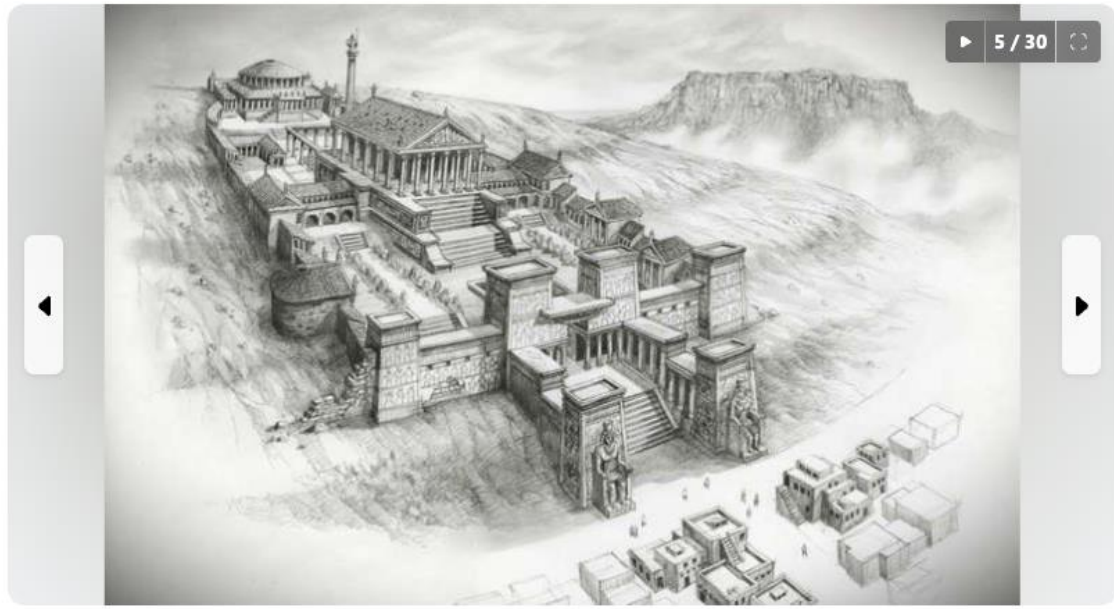
©Getty Images

Med udgangspunkt i Alexander den Stores drøm om en verden forenet gennem viden, forestillede Ptolemaios I og senere Ptolemaios II sig et storslået sted, hvor al menneskelig viden kunne samles, bevares og udvikles af tidens skarpeste hjerner.

Bøger ved kajen

©Getty Images

Under Ptolemaios III blev det lov, at alle skibe, der lagde til i Alexandrias havn, skulle aflevere eventuelle bøger. Skrivere lavede kopier, originalerne blev beholdt, og kopierne røg tilbage ombord. Sådan voksede samlingen konstant.



Musernes hellige hjem

©Public Domain

Biblioteket var en del af Mouseion – opkaldt efter muserne, de ni gudinder for kunst og viden. Med op mod en halv million skriftruller blev det et magnetisk centrum for lærde fra hele den antikke verden.


Mette Mortensen
PREPPERENS HÅNDBOG
NATURAL LANGUAGE PROCESSING WITH PYTHON AND SPACY
A PRACTICAL INTRODUCTION
YULI VASILIEV
O'REILLY
Text Mining with R
A TIDY APPROACH
Julia Silge & David Robinson
FIRST AID KIT

Planen

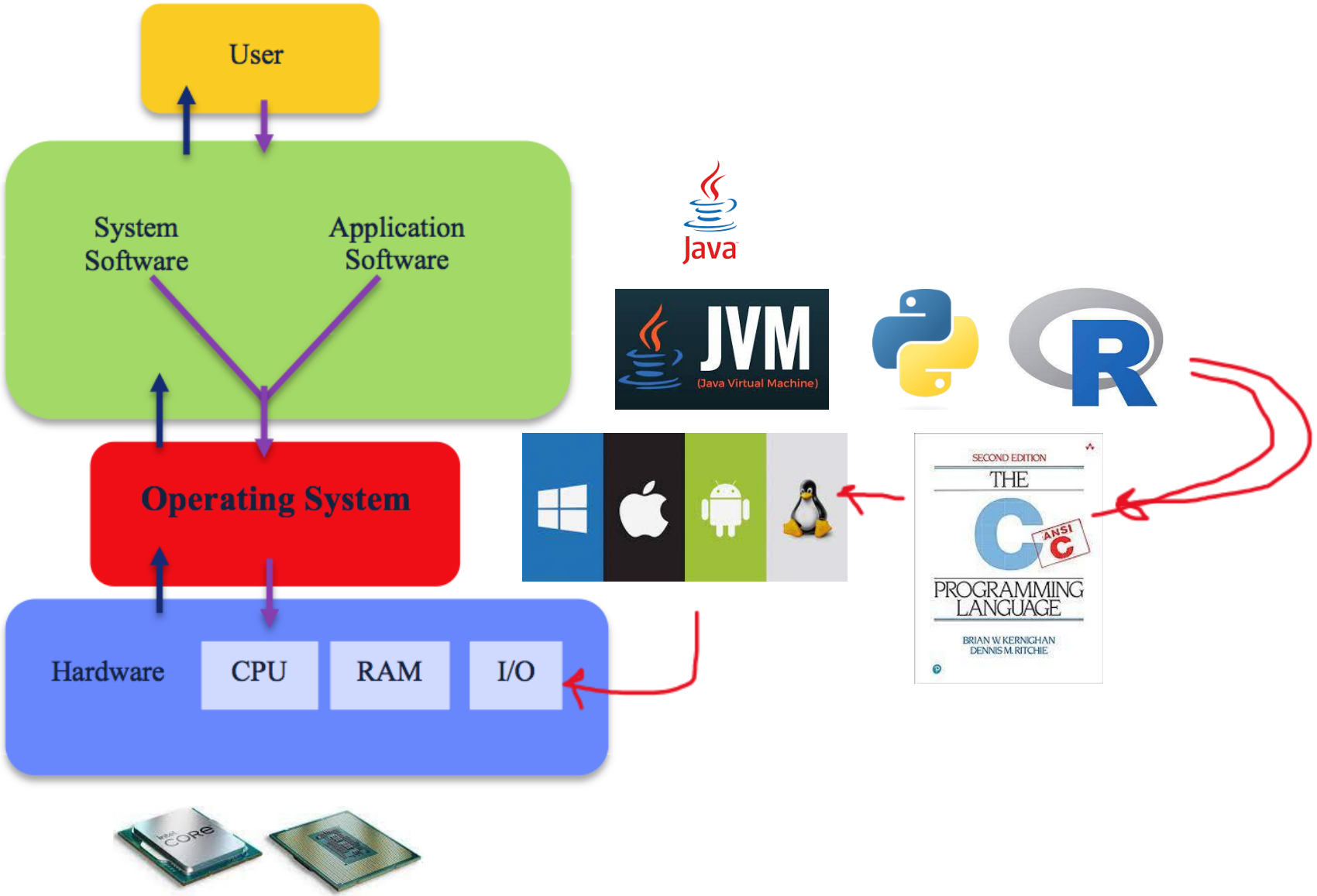
Tid

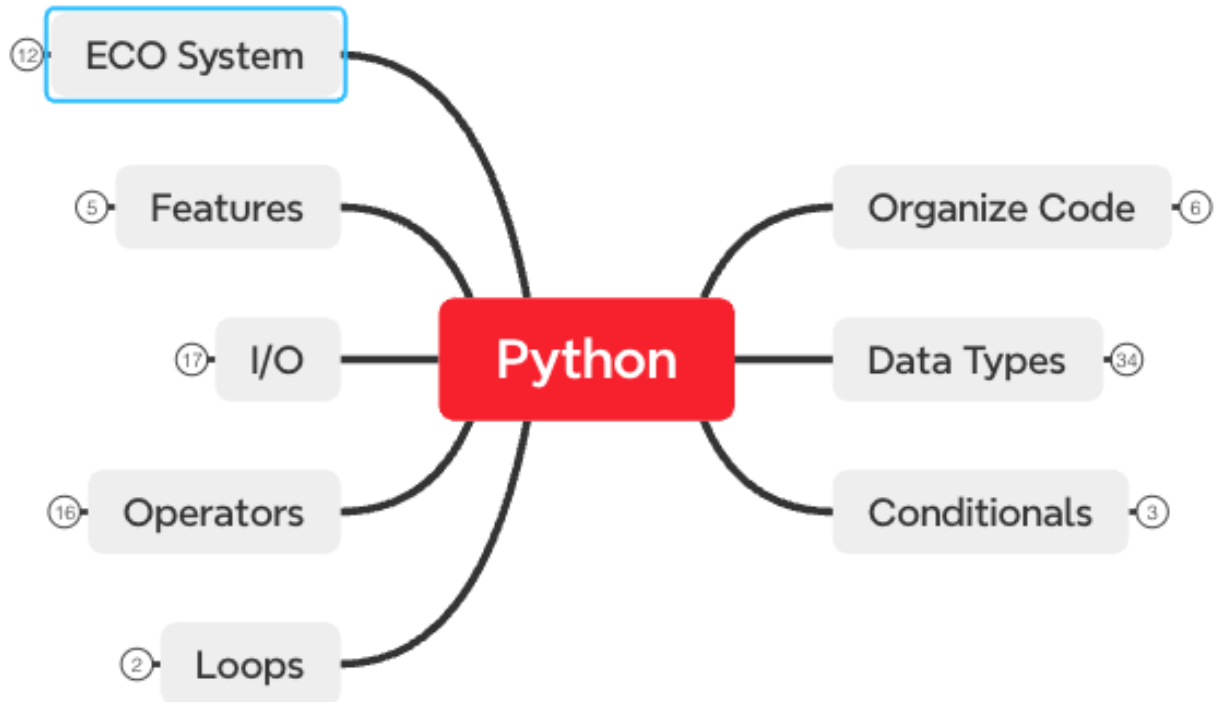
Uger	Dato	Fag	Tidspunkt	Indhold
Uge 18				
	30/4	NLP	Formiddag	Intro til valgfaget - 8.30 til 10.00
	1/5	NLP - Python	Formiddag	Intro til Python + regulære udtryk
	2/5	NLP - Python	Formiddag	"NLP" kapitel 2
Uge 19	5/5	NLP - Python	Eftermiddag	
	6/5	NLP - Python	Eftermiddag	
	8/5	NLP - R	Eftermiddag	
	9/5	NLP - R	Formiddag (OL)	
Uge 20	12/5		Formiddag	
	13/5		Formiddag	
	15/5		Formiddag	
	16/5		Formiddag	
Uge 21	19/5		Eftermiddag	Bigrams
	20/5		Eftermiddag	Bigrams
	22/5		Eftermiddag	
	23/5		Formiddag (OL)	
Uge 22	26/5	Deep Learning	Formiddag	
	27/5	Deep Learning	Formiddag	
	28/5			
	29/5	Fri		
	30/5			
Uge 23	2/6		Eftermiddag	
	3/6		Eftermiddag	
	4/6			
	5/6		Eftermiddag	
	6/6		Formiddag	
Uge 24	9/6	Pinse		
	10/6	Eksamen		
	11/6	Eksamen		

Planen

Indhold

Python


User
System Software
Application Software
Operating System
Hardware
CPU
RAM
I/O
Java
JVM
(Java Virtual Machine)
SECOND EDITION
THE
C
ANSI C
PROGRAMMING LANGUAGE
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE


Python
ECO System
12
Features
5
I/O
17
Operators
16
Loops
2
Organize Code
6
Data Types
34
Conditionals
3

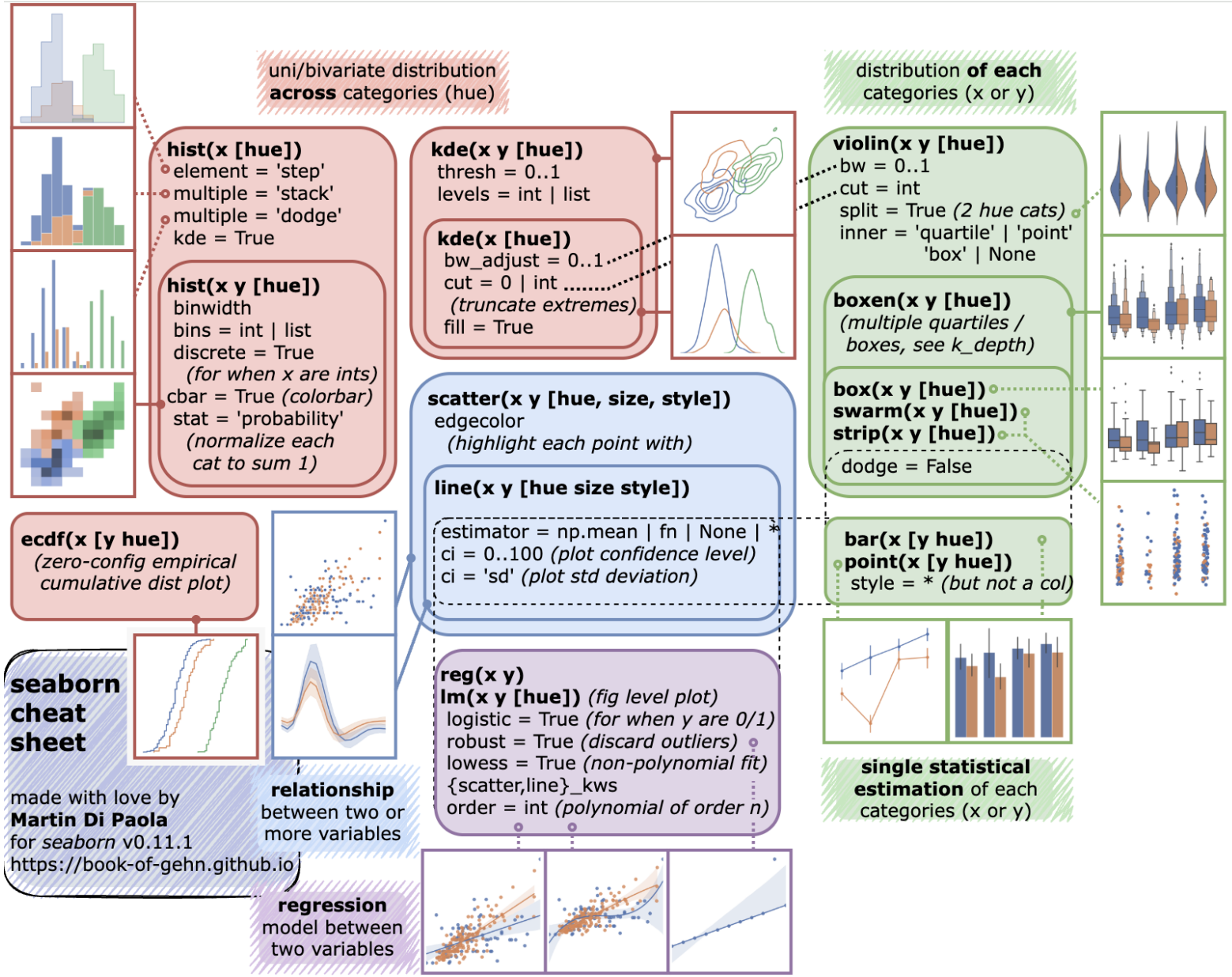
and	A logical operator
as	To create an alias
assert	For debugging
break	To break out of a loop
class	To define a class
continue	To go to the next iteration of a loop
def	To define a function
del	To delete an object
elif	A conditional statements, like else if
else	A conditional statements
except	Used with exceptions, what to do when an exception occurs
False	Boolean value
finally	Used with exceptions, will be executed no matter if there is an exception or not
for	To create a for loop
from	To import specific parts of a module
global	To declare a global variable
if	To make a conditional statement
import	To import a module
in	To check if a value is in a list, tuple
is	To test if two variables are equal
lambda	To create an anonymous function
None	Represents a null value
nonlocal	To declare a non-local variable
not	A logical operator
or	A logical operator
pass	A statement that will do nothing (null)
raise	To raise an exception
return	To exit a function and return a value
True	Boolean value
try	To make a try...except statement
while	To create a while loop
with	Used to simplify exception handling
yield	To end a function, returns a generator

Built-in Functions

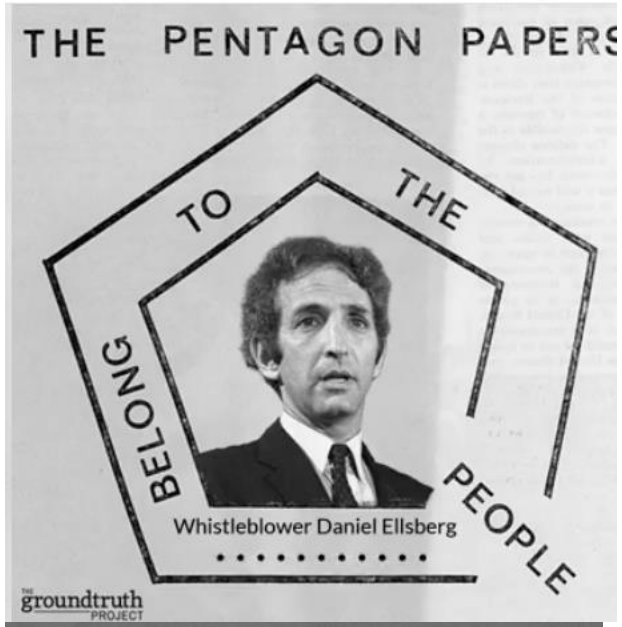
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

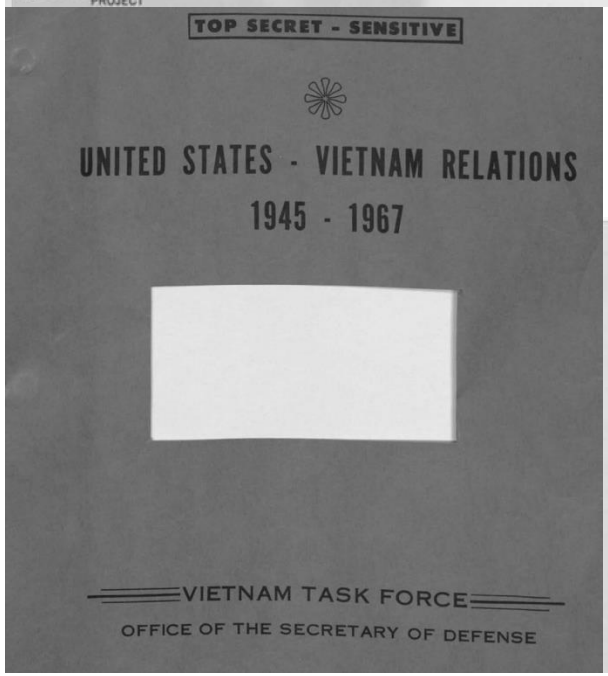
Function	Description	Syntax
compile	Returns a Regex pattern object	re.compile(pattern)
findall	Returns a list containing all the matches	re.findall(pattern, text)
match	Returns a match object if there is a match at the 0th position	re.match(pattern, text)
search	Returns a match object if there is a match anywhere in the string	re.search(pattern, text)
split	Returns a split where the string has been split at each match	re.split("seperator", text)
sub	Replaces one or many matches with a string	re.sub(old_value, new_value, text)

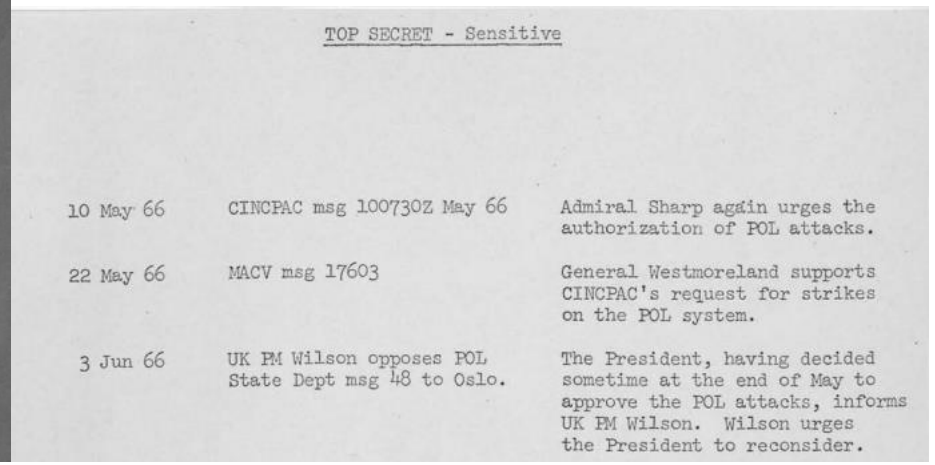
Method / Property	Description	Example
`m.group()`	Returns the matched string	`'o'`
`m.start()`	Start index of the match	`0`
`m.end()`	End index of the match	`1`
`m.span()`	Tuple of (start, end) indices	`(0, 1)`
`m.re`	The compiled regular expression object	`re.compile('^o')`
`m.string`	The original string	`'otto'`


uni/bivariate distribution
across categories (hue)
hist(x [hue])
element = 'step'
multiple = 'stack'
multiple = 'dodge'
kde = True
hist(x y [hue])
binwidth
bins = int | list
discrete = True
(for when x are ints)
cbar = True (colorbar)
stat = 'probability'
(normalize each
cat to sum 1)
kde(x y [hue])
thresh = 0..1
levels = int | list
kde(x [hue])
bw_adjust = 0..1
cut = 0 | int .....
(truncate extremes)
fill = True
distribution of each
categories (x or y)
violin(x y [hue])
bw = 0..1
cut = int
split = True (2 hue cats)
inner = 'quartile' | 'point'
'box' | None
boxen(x y [hue])
(multiple quartiles /
boxes, see k_depth)
box(x y [hue])
swarm(x y [hue])
strip(x y [hue])
dodge = False
scatter(x y [hue, size, style])
edgecolor
(highlight each point with)
line(x y [hue size style])
estimator = np.mean | fn | None | *
ci = 0..100 (plot confidence level)
ci = 'sd' (plot std deviation)
bar(x [y hue])
point(x [y hue])
style = * (but not a col)
ecdf(x [y hue])
(zero-config empirical
cumulative dist plot)
reg(x y)
lm(x y [hue]) (fig level plot)
logistic = True (for when y are 0/1)
robust = True (discard outliers)
lowess = True (non-polynomial fit)
{scatter,line}_kws
order = int (polynomial of order n)
seaborn
cheat
sheet
made with love by
Martin Di Paola
for seaborn v0.11.1
https://book-of-gehn.github.io
relationship
between two or
more variables
regression
model between
two variables
single statistical
estimation of each
categories (x or y)

Projektet







TOP SECRET - Sensitive

10 May 66	CINCPAC msg 100730Z May 66	Admiral Sharp again urges the authorization of POL attacks.
22 May 66	MACV msg 17603	General Westmoreland supports CINCPAC's request for strikes on the POL system.
3 Jun 66	UK PM Wilson opposes POL State Dept msg 48 to Oslo.	The President, having decided sometime at the end of May to approve the POL attacks, informs UK PM Wilson. Wilson urges the President to reconsider.

- https://www.archives.gov/research/pentagon-papers
- Download en pdf

TERMINALEN

- pdfseparate Pentagon-Papers-Part-IV-C-7-b.pdf page-%d.pdf
- ls *pdf | while read x; do convert -density 300 $x $x.png; done
- ls *png | while read x; do tesseract $x $x; done
- LØS SORTERINGSPROBLEMET vha Python
- ls *txt | while read x; do cat $x >> out.txt; done

SPYDER

- Indlæs out.txt i Spyder

LØS SORTERINGSPROBLEMET vha Python

```
page-1.pdf
page-10.pdf
page-100.pdf
page-101.pdf
page-102.pdf
page-103.pdf
```