

# Workshop Data Science

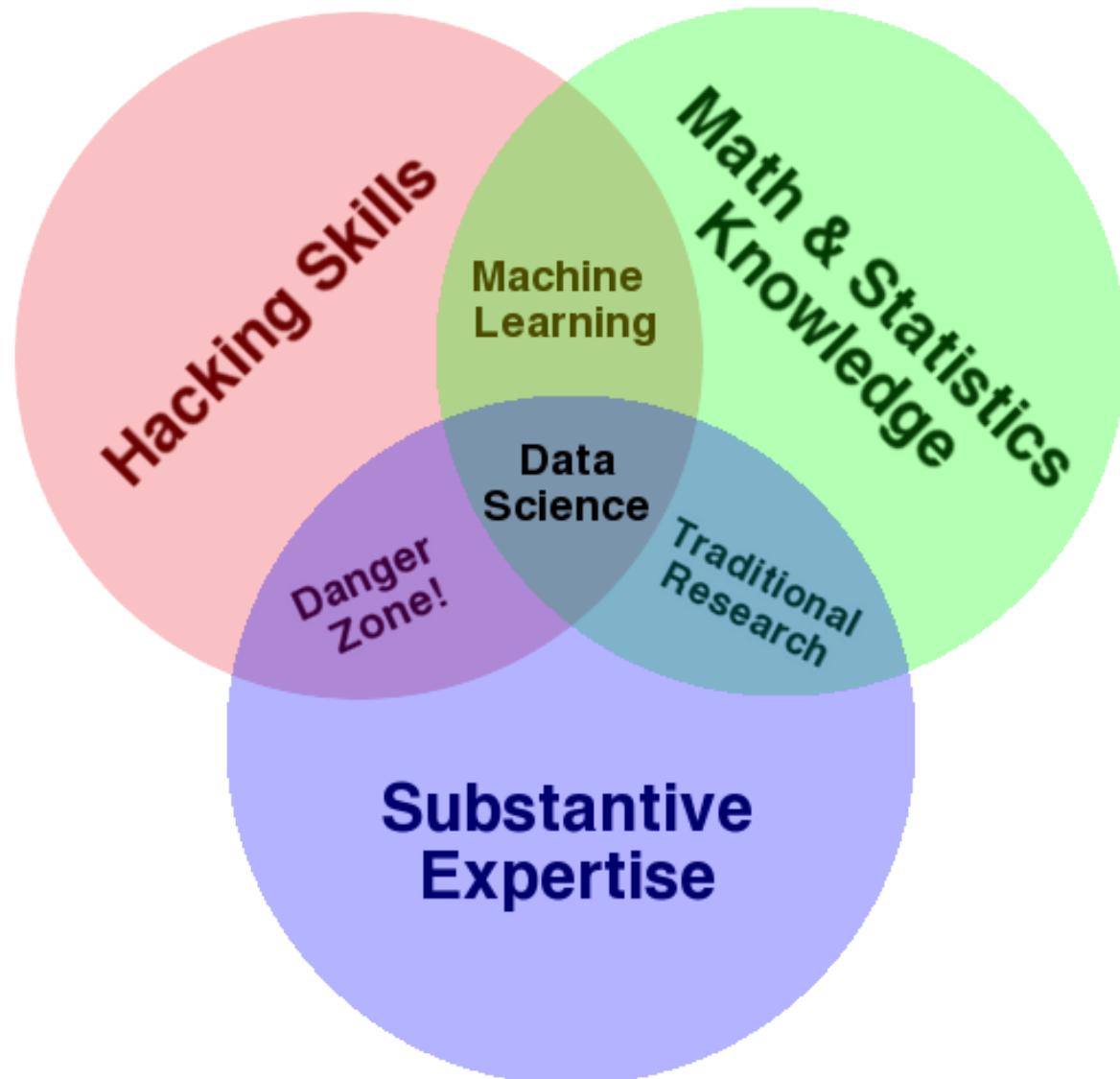
1 februari 2017

Rory Sie

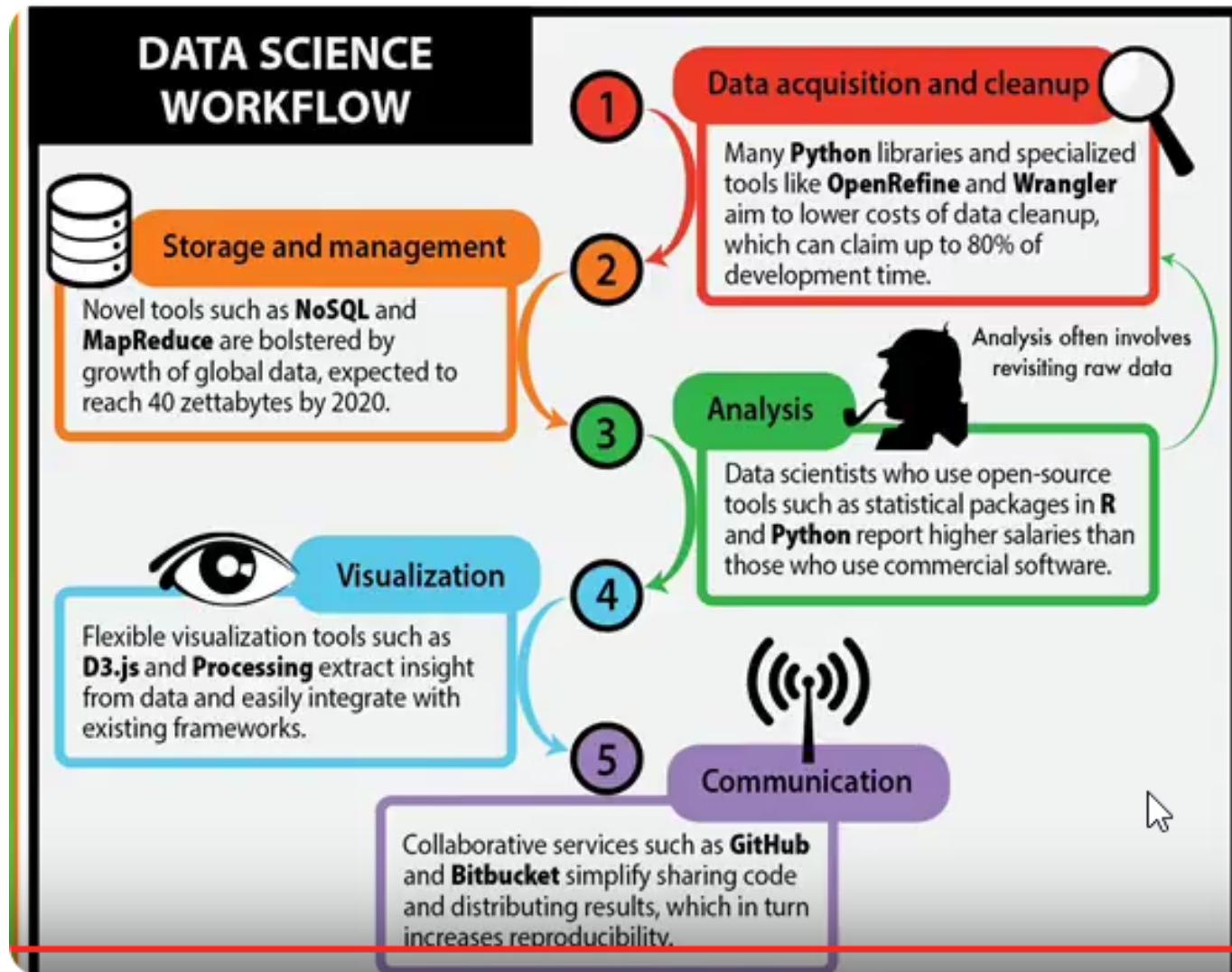
# Agenda

- Wat is Data Science?
- Numpy
- Pandas en inlezen bestanden
- Scikit-learn
- Visualisatie in Bokeh

# Wat is Data Science?



# Wat doet een data scientist?



# Wat heb je dan minimaal nodig?

- Python/Pycharm/Anaconda/Jupyter Notebook
- Data acquisition & cleanup: dedupe (deduplication), fuzzywuzzy (string matching), arrow (date/time), datacleaner (NAs)
  - Evt. Anonymization: scrubadub
- Storage & Management: bijvoorbeeld SQL & sqlalchemy of Django
- Analysis: Numpy, Pandas, Scikit-learn (Anaconda/Jupyter Notebook)
- Visualization: Matplotlib, Bokeh
- Communication: domeinkennis

# Jupyter Notebook Voorbeeld

The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with various icons for file operations like opening, saving, and running cells. Below the toolbar is a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A sub-menu for 'Cell' is open, showing options like Run Cell, Run Cell In Place, and Insert Cell Below.

The main area has a section titled 'Data Mungling'. It contains a text block: 'De ingelezen data moeten nu omgevormd worden naar een bruikbaar formaat: per jaartal het aantal perioden'. Below this is a code cell labeled 'In [4]:' containing Python code for data manipulation:

```
# alleen de kolommen 1 t/m 5 overnemen:  
df = df[[1,2,3,4,5]]  
# Rijen en kolommen verwisselen:  
df = df.T  
# Kolomnamen wijzigen:  
df.columns=['Jaar','Aantal']  
# Jaar: omvormen naar jaartal  
df['Jaar'] = df['Jaar'].str[0:4]  
# Jaar: omvormen naar numerieke waarde  
df['Jaar'] = df['Jaar'].astype(int)  
# Aantal: omvormen naar numerieke waarde  
df['Aantal'] = df['Aantal'].astype(int)  
# toon omgevormde waarde  
df
```

Below the code cell is an output cell labeled 'Out [4:]' showing a table:

	Jaar	Aantal
1	2010	19188
2	2011	20449
3	2012	21498
4	2013	23509
5	2014	25128

- Cellen met
  - Tekst (markdown)
  - Code (Python)

## Shortcuts

- <Esc>: Command mode
- <Enter> edit mode

## Edit mode:

- Shift-Enter: run cell
- Ctrl-Enter: run cell in-place
- Alt-Enter: run cell, insert below

## Command mode:

- Esc X: delete cell
- Esc B: Add cell Below

click Help, Keyboard Shortcuts

# Markdown

- Tekstopmaak web o.a. ook bij github

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

## syntax

```
Plain text  
End a line with two spaces to start a new paragraph.  
*italics* and _italics_  
**bold** and __bold__  
superscript^2^  
~~strikethrough~~  
[link] (www.rstudio.com)
```

```
# Header 1  
  
## Header 2  
  
### Header 3  
  
#### Header 4  
  
##### Header 5  
  
##### Header 6  
  
endash: --  
emdash: ---  
ellipsis: ...  
inline equation: $A = \pi * r^2$  
image:   
  
horizontal rule (or slide break):  
  
***
```

```
> block quote  
  
* unordered list  
* item 2  
  + sub-item 1  
  + sub-item 2  
  
1. ordered list  
2. item 2  
  + sub-item 1  
  + sub-item 2
```

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

## becomes

```
Plain text  
End a line with two spaces to start a new paragraph.  
italics and italics  
bold and bold  
superscript2  
strikethrough  
link
```

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: –  
emdash: —  
ellipsis: ...  
inline equation:  $A = \pi * r^2$   
image: 

horizontal rule (or slide break):

block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
  - sub-item 1
  - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

# Meer info

- <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook#gs.TD4BcSc>
- <https://www.jetbrains.com/help/pycharm/2016.3/using-ipython-jupyter-notebook-with-pycharm.html>
- <https://youtu.be/e9cSF3eVQv0>
- TL;DR: Je kunt in Jupyter Notebooks Python stukjes **code runnen**, maar ook **voorzien van uitleg** en **visualisaties**

# Opdracht in de les (10 min.)

- Start een Jupyter notebook in Pycharm  
(kan ook op: <https://try.jupyter.org/> )
- Maak zelf:

The screenshot shows a Jupyter Notebook interface with the following elements:

- Title:** Mijn eerste Notebook
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help; icons for New, Open, Save, Run, Kernel, CellToolbar, Cloud, GitHub, YouTube.
- Section:** H1 titled "Mijn cijferoverzicht".
- Text:** "Doel is om het gemiddelde van mijn cijfers in het 2e jaar"
- Code:** In [15]: 

```
# mijn cijfers in een lijst
cijfers = [7.3, 6.4, 5.6, 6.7, 7.1, 5.9]
```
- Output:** Out[15]: [7.3, 6.4, 5.6, 6.7, 7.1, 5.9]
- Text:** "Bepalen van het gemiddelde"
- Code:** In [16]: 

```
# het gemiddelde is de optelling van de cijfers gedeeld door het aantal
som = sum(cijfers)
aantal = len(cijfers)
gemiddelde = som / aantal
print('Mijn gemiddelde cijfer is: {:.1f}'.format(gemiddelde))
```
- Output:** Mijn gemiddelde cijfer is: 6.5

# Numpy

- Python module voor *Scientific Computing*
  - Arrays van data en wiskundige berekeningen
- Array?
  - Een ‘lijst’ met homogene waarden
    - Kan ééndimensionaal zijn 1, 2, 4, 6
    - Maar ook multidimensionaal, 2-D in dit geval: 1, 2, 4, 6  
4, 6, 9, 3  
0, 3, 7, 2  
8, 5, 3, 2

# NumPy - Introductie

- Te bestuderen:
  - <http://programmersdiary.com/python/numpy-tutorial-for-beginners-array-basics/>
  - <http://programmersdiary.com/python/numpy-tutorial-for-beginners-array-operations/>
- Detail informatie:
  - <https://docs.scipy.org/doc/numpy-dev>

# En nu maggie oefene (15 min.)

1. Maak een Jupyter notebook aan
2. Maak een 1-D array aan met de cijfers 0 t/m 3
3. Laat de dimensies zien
4. Hervorm de array tot een 2-D array (2, 2)
5. Wat zijn nu de dimensies?
6. Laat het datatype zien

# NumPy - Introductie

- Te bestuderen:
  - <http://programmersdiary.com/python/numpy-tutorial-for-beginners-array-basics/>
  - <http://programmersdiary.com/python/numpy-tutorial-for-beginners-array-operations/>
- Detail informatie:
  - <https://docs.scipy.org/doc/numpy-dev>

# En nu maggie oefene (20 min.)

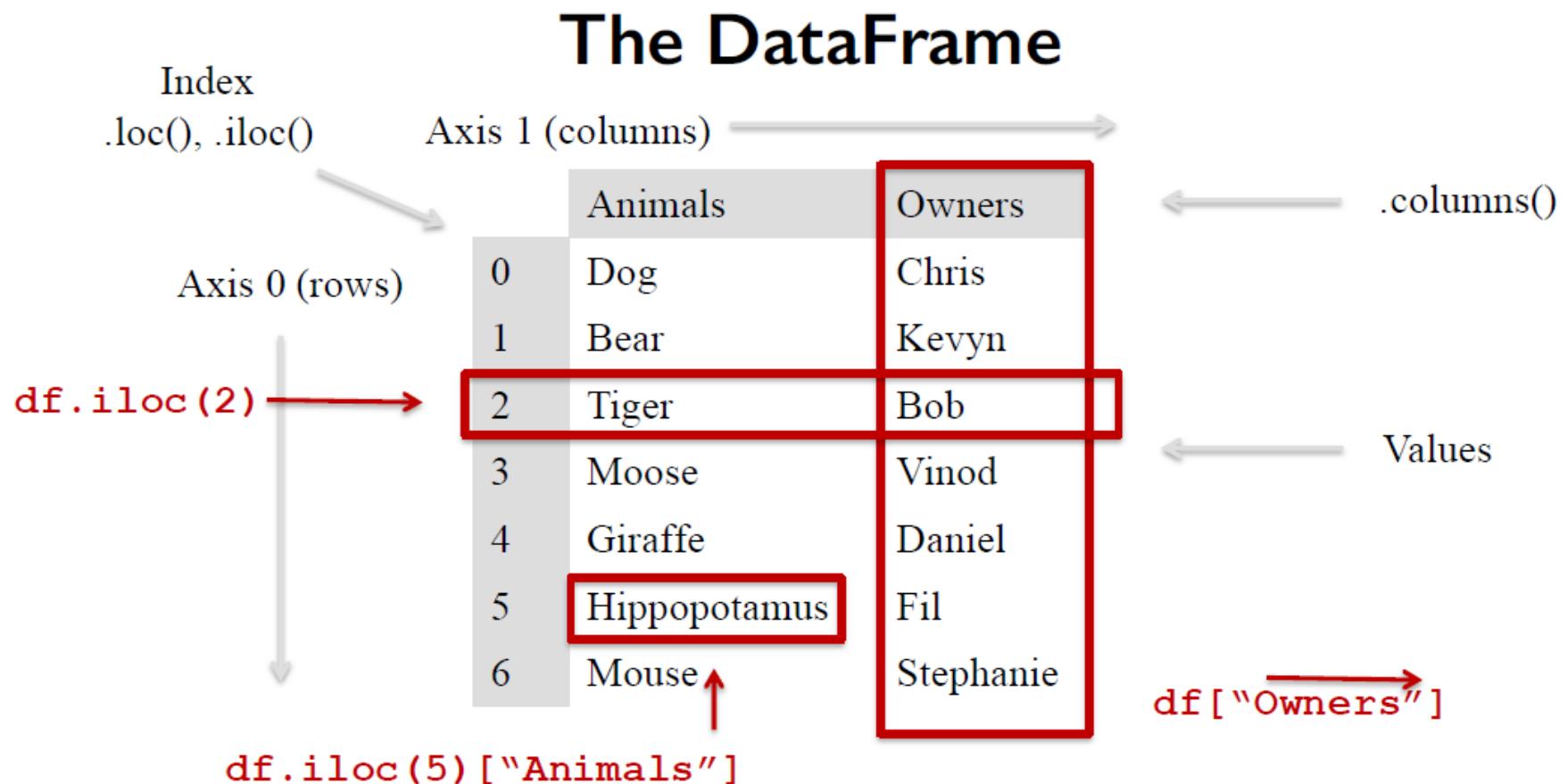
1. Ga verder met het vorige notebook
2. Sla een kopie van de array op in **y**
3. Maak een transpose van **x**
4. Maak een transpose van **y**
5. Vermenigvuldig alle waarden van **y** met 2
6. Concateneer de twee arrays in een nieuwe array
7. Geef het gemiddelde van **z**
8. Geef de standaarddeviatie van **z**

# Pandas – Introductie

- Doel: multi-typed dataverzamelingen beheren
- **Series**: 1 dimensionaal (indexed array)
- **Dataframe**: 2 dimensionaal, indexed dataset
  - Multicolumn: each column is a Series
- **Panel**: 3 dimensionaal (niet veel gebruikt)

Zie: <http://pandas.pydata.org/pandas-docs/stable/dsintro.html>

# Dataframe weergave



# Learning Pandas

## Dataschool.io – 1

- Introductie
  - <http://www.dataschool.io/easier-data-analysis-with-pandas/>
- Playlist:
  - <https://www.youtube.com/playlist?list=PL5-da3qGB5ICCsgW1MxIz0Hq8LL5U3u9y>
- Notebook:
  - <http://nbviewer.jupyter.org/github/justmarkham/pandas-videos/blob/master/pandas.ipynb>

# Learning Pandas

## Dataschool.io – 2

### Leerpunten: 1 t/m 11

```
import pandas as pd  
pd.read_table(), pd.read_csv()  
.head(), .tail(), .describe(), .shape,
```

### Selecting columns:

- a column as a series: ufo['City'] or ufo.City
- a column as a dataframe: ufo[['City']]
- Multiple columns ufo[['City', 'State']]

### Adding / changing a column:

- All rows same value ufo['Country'] = 'USA'
- Combined value ufo['Location'] = ufo['City'] + ufo.State

Drop Rows/Columns: drop(): rows: axis=0, columns: axis = 1

- .sort\_values()

### Selecting rows (filtering):

- Met een lijst met booleans
- Met een statement: movies[movies.duration >= 200]
- Met .loc, met .iloc (zie 19), met isin()
- Multiple filters: &, |

# Inladen van een bestand

```
import pandas as pd  
data = pd.read_csv(<filepath>)  
data = pd.read_excel(<filepath>) (installeer package xlrd)
```

<https://github.com/openrory/WorkshopDataScience/blob/master/hu-pandas-voorbeeld.ipynb>

# Ennu maggie oefene (15 min.)

- Importeer de iris dataset:
  - import pandas as pd
  - from sklearn import datasets
  - data = datasets.load\_iris()
  - iris = pd.DataFrame(data.data,  
columns=data.feature\_names)
- Bekijk eens hoe de dataset eruitziet
- Probeer een aantal rows te selecteren
- Probeer alle rows te vinden met sepal length > 5
- Verwijder alle rows met petal length > 1.4

# Oplossing

<https://github.com/openrory/WorkshopDataScience/blob/master/hu-pandas-iris.ipynb>

# Data cleanup

- Zie code  
<https://github.com/openrory/WorkshopDataScience/blob/master/hu-pandas-titanic.ipynb>
- NAs verwijderen
- sklearn.preprocessing.Imputer

# Data-analyse: scikit-learn

- K-Means clustering

# Wat is machine learning (ML)

1. What is machine learning, and how does it work? ([video](#), [notebook](#), [blog post](#))

- What is machine learning?
- What are the two main categories of machine learning?
- What are some examples of machine learning?
- How does machine learning "work"?

- "Machine learning (ML) is the semi-automated extraction of knowledge from data"
  - Machine : Een computer vormt gegevens om naar kennis
  - Knowledge : bruikbare kennis
  - Automated : de computer voert uitgebreide berekeningen uit
  - Semi-automated : de gebruiker neemt de beslissingen

# Machine Learning soorten

- Supervised: (predictions)
  - Gebaseerd op datasets waarvan de uitkomst al bekend is. Deze datasets worden gebruikt om te leren (“fit”)
  - Toepassen op nieuwe datasets (“predict”)
- Unsupervised: (structuring, datamining)
  - Gebaseerd op datasets zonder specifieke uitkomst.
  - Proberen groepen/indeling te vinden die nuttig zijn

Labeled

Unlabeled

Pattern recognition

<https://www.e-sites.nl/blog/476-machine-learning-een-korte-toelichting-op-de-techniek-en-toepassing.html>

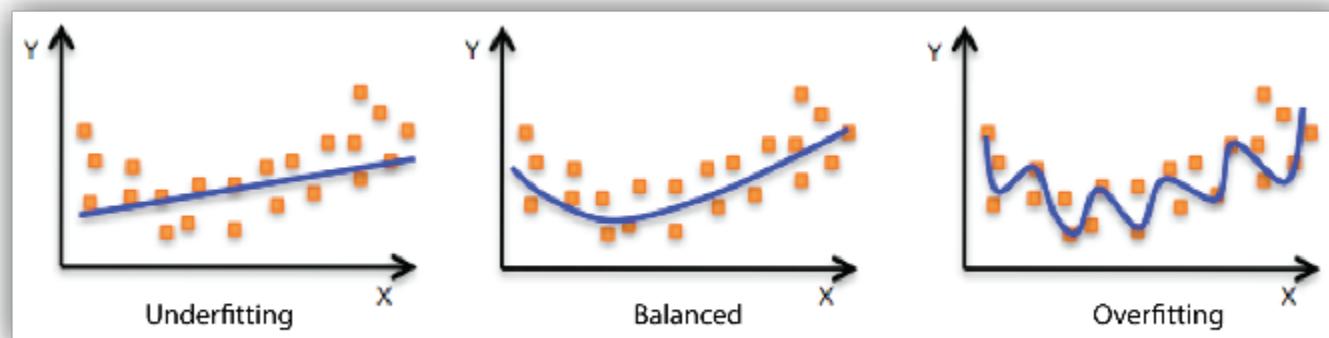
[https://en.wikipedia.org/wiki/Unsupervised\\_learning](https://en.wikipedia.org/wiki/Unsupervised_learning)

[https://en.wikipedia.org/wiki/Supervised\\_learning](https://en.wikipedia.org/wiki/Supervised_learning)

# Train, test, overfit

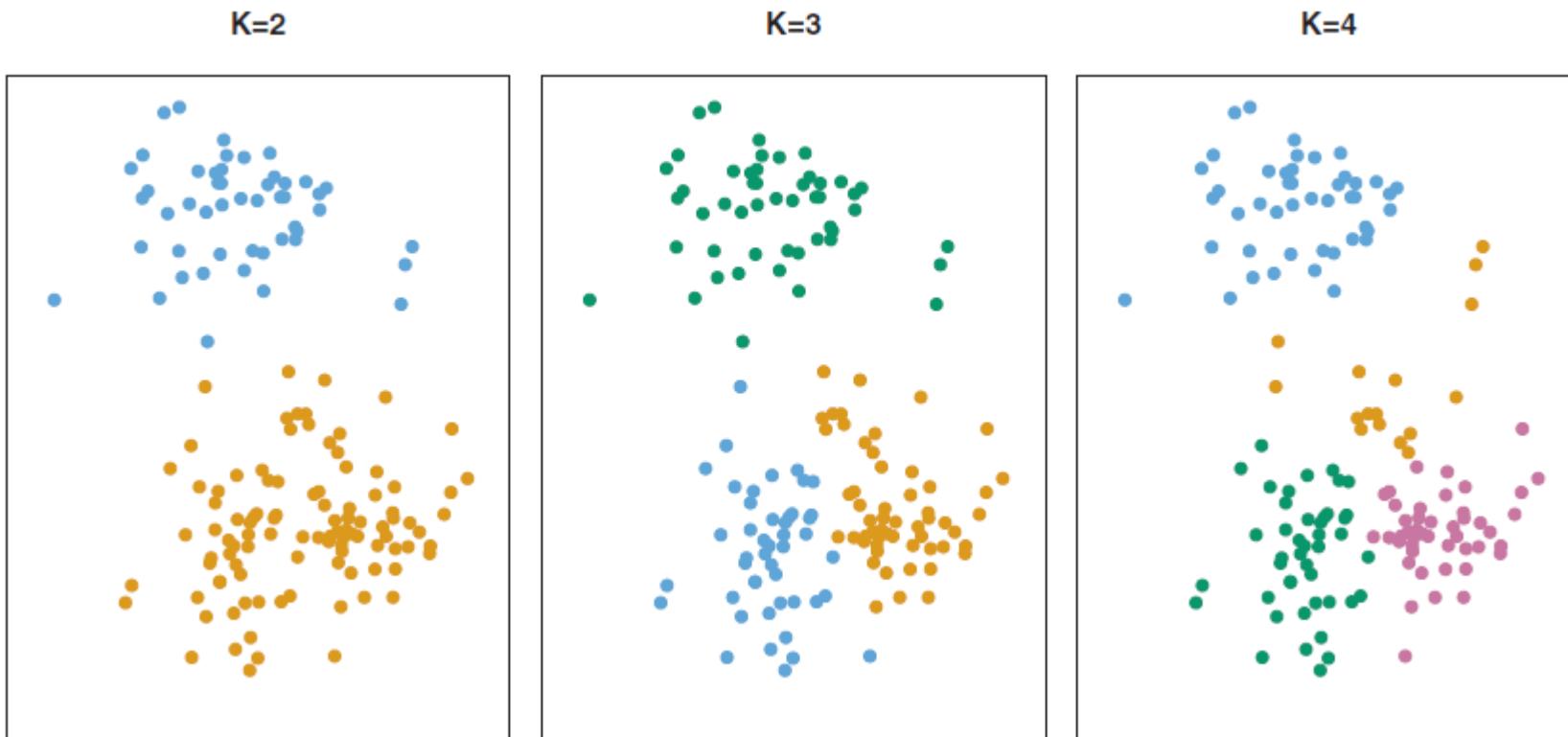
- Supervised learning:
  - Splits de gegevensverzameling met bekende uitkomsten ("labeled data") in een train-deel (60-80%) en test-deel (20-40%)
  - Het train-deel wordt gebruikt voor 'learning'.
  - Het test-deel wordt gebruikt om na 'learning' te controleren hoe goed het model is.
  - Dit voorkomt overfitting: situatie dat het model in feite heel erg goed is voor de data waaruit het geleerd heeft en minder voor alle andere situaties

Zie: <https://en.wikipedia.org/wiki/Overfitting>



# K-means clustering

- Voorbeeld (bij 2 features)



# Voorbeeld

- <https://github.com/openrory/WorkshopDataScience/blob/master/hu-pandas-titanic.ipynb>

# Ennu maggie oefene (20 min)

- De iris dataset bestaat uit drie soorten: *Iris setosa*, *Iris virginica* en *Iris versicolor*
- Gebruik k-means clustering ( $k = 3$ ) om de drie clusters te vinden
- in `data.target` staat de juiste classificatie: vergelijk deze met je eigen classificatie

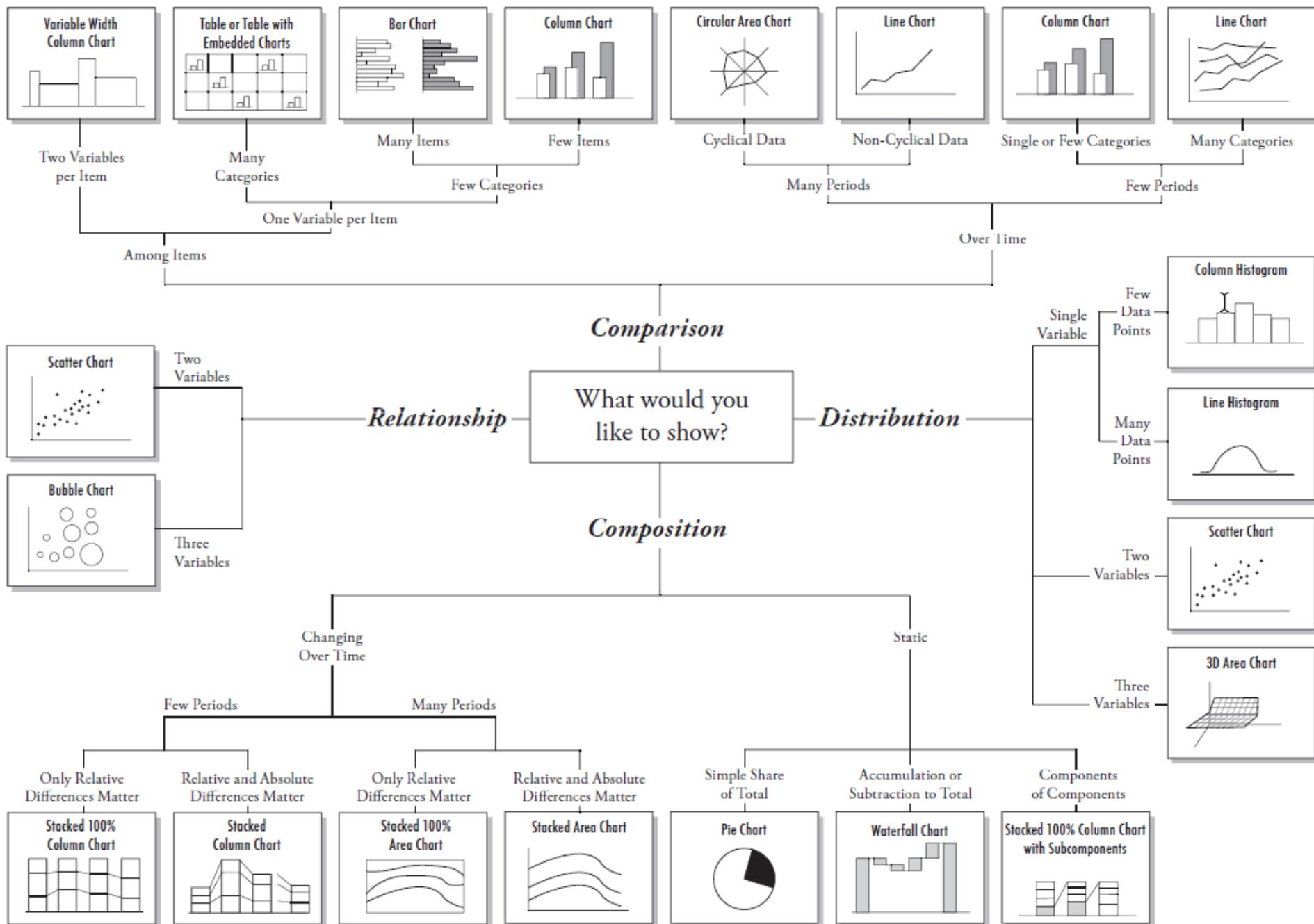
# Oplossing

- <https://github.com/openrory/WorkshopDataScience/blob/master/hu-ml1.ipynb>

# Visualisatie

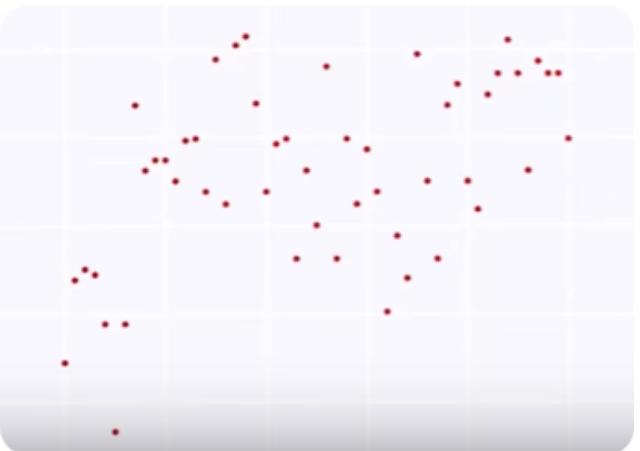


# Chart Suggestions—A Thought-Starter



# Scatter -> line -> Loose curve

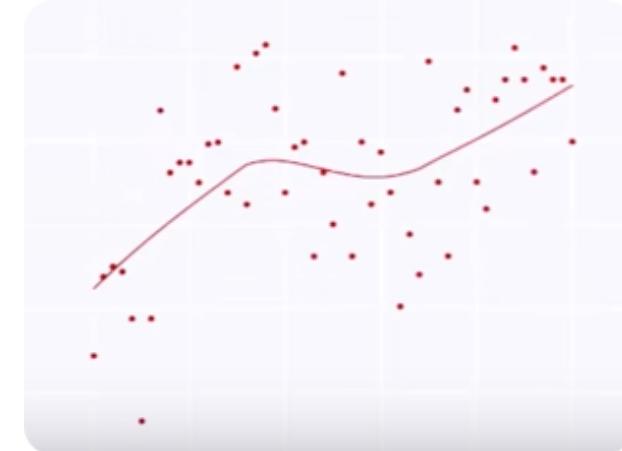
- Wat wil je zien



Algemene verdeling



Kort termijn trends

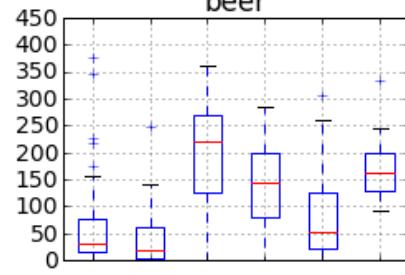
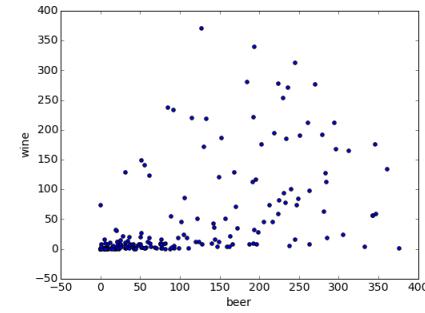
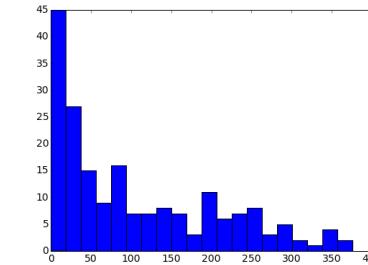


Lange termijn trend

Udacity videos: <https://www.youtube.com/watch?list=PLAwxTw4SYaPk41og7PER4HBpGciPw6n3x>  
(video 158, 159, 160)

# Charts voor data exploration

- Histogram: hoe zijn de meetwaarden verdeeld
- Scatterdiagram: hoe is de samenhang tussen de gegevens
- Boxplot: hoe is de verdeling rond het ‘midden’

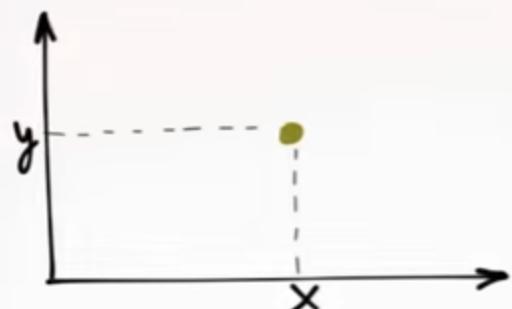


Bron:

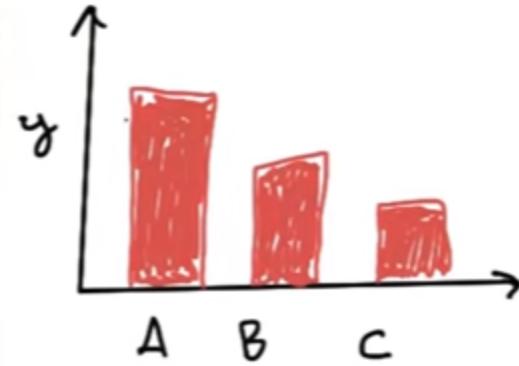
[https://github.com/justmarkham/DAT8/blob/master/notebooks/05\\_pandas\\_visualization.ipynb](https://github.com/justmarkham/DAT8/blob/master/notebooks/05_pandas_visualization.ipynb)

# Visual cues / Visual encoding - 1

## Visual Encoding



Position



Length



Angle

# Visual cues / Visual encoding - 2

Visual Encoding

--\ \ \ \

Direction

○□△◊♣ •○○○○

Shape

Area / Volume

# Bokeh

Zie: <http://bokeh.pydata.org/en/latest/>

- Python interactive visualization library that targets modern web browsers for presentation



# Bokeh User\_guide - 1

- [http://bokeh.pydata.org/en/latest/docs/user\\_guide.html](http://bokeh.pydata.org/en/latest/docs/user_guide.html)
- [http://bokeh.pydata.org/en/latest/docs/user\\_guide/quickstart.html](http://bokeh.pydata.org/en/latest/docs/user_guide/quickstart.html)

Leerpunten

Concepts: plot, Glyphs, Guides and Annotations, Ranges, Resources

Show Inline in notebook

```
from bokeh.plotting import figure, output_file, show, output_notebook
figure(), show(), output_file(), output_notebook(), gridplot()
x-as label, y-as lable, legend
```

```
.circle(), .line(), .triangle(), .square(). xaxis, .yaxis, .xgrid, .ygrid
```

Tools

# Bokeh User\_guide - 2

- [http://bokeh.pydata.org/en/latest/docs/user\\_guide/charts.html](http://bokeh.pydata.org/en/latest/docs/user_guide/charts.html) (Making high level graphs)

Leerpunten:

Accepted inputs (Dataframe, list, tuple, dict,

Chart(), Bar(), BoxPlot(), Histogram(), Scatter()

Aggregation possibilities: sum, mean, count, median, min, max

Grouping, Stacking

- [http://bokeh.pydata.org/en/latest/docs/user\\_guide/notebook.html](http://bokeh.pydata.org/en/latest/docs/user_guide/notebook.html) (inline plots)

Leerpunten:

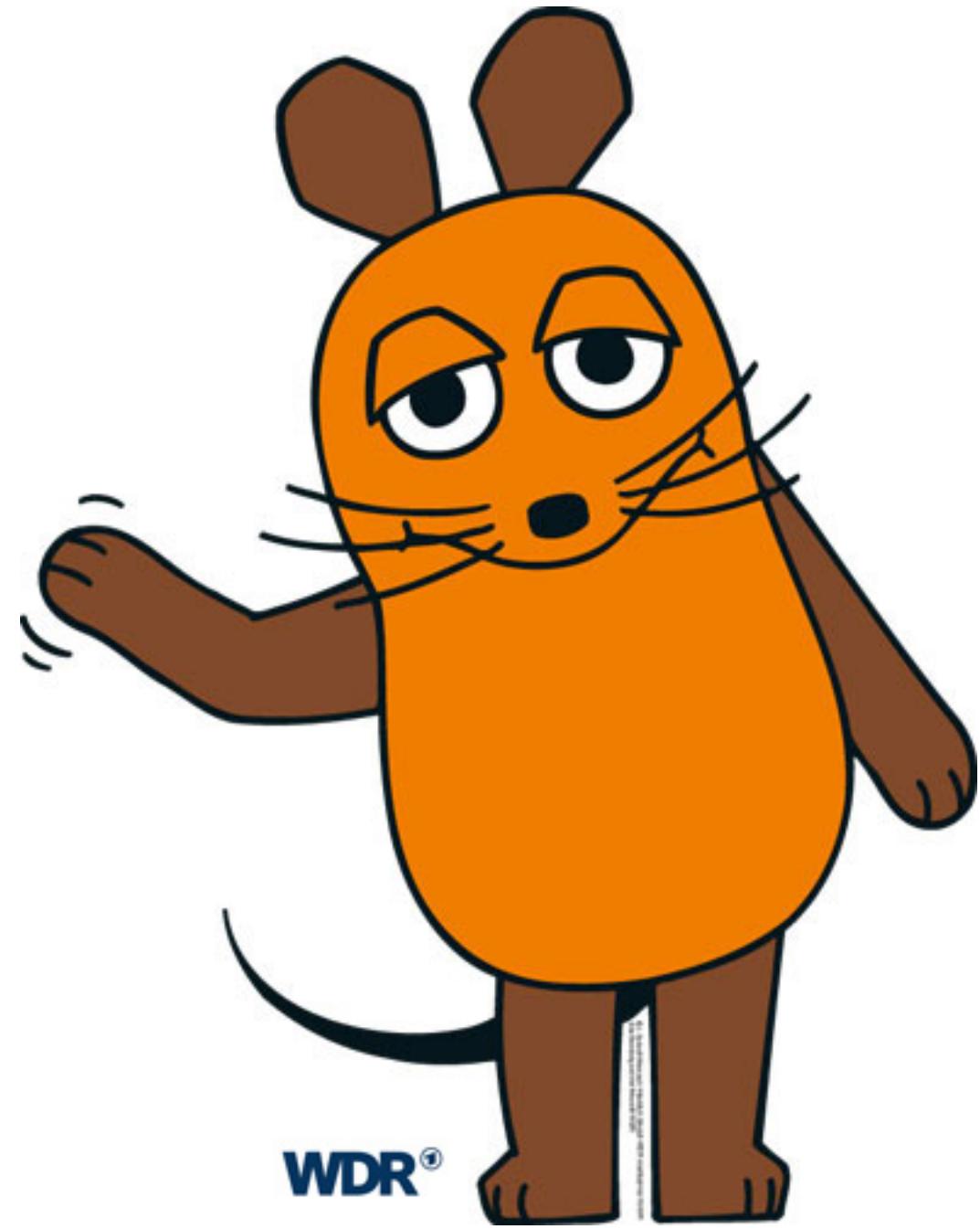
output\_notebook()

# Bokeh tutorial notebook – 1 (leerpunten zie User\_guide)

- Introductie :  
<http://nbviewer.jupyter.org/github/bokeh/bokeh-notebooks/blob/master/tutorial/00%20-%20intro.ipynb>
- Plotting:  
<http://nbviewer.jupyter.org/github/bokeh/bokeh-notebooks/blob/master/tutorial/01%20-%20plotting.ipynb>
- Highlevel graphs:  
<http://nbviewer.jupyter.org/github/bokeh/bokeh-notebooks/blob/master/tutorial/10%20-%20charts.ipynb>

# Opdracht in de les (30 min.)

- Maak in Bokeh een grafiek waarmee je het ‘cijferverloop’ van een 2<sup>e</sup> jaars SIE student toont.
- File:  
<https://github.com/openrory/WorkshopDataScience/blob/master/files/cijfers.csv>
- Let op: je hebt de functie pd.to\_datetime(x, dayfirst=True) nodig



**WDR®**