LESSON 11

Similarity and Deep Learning (with coding)

Exploratory Data Analysis

Feature engineering




Exam Simulation: Lesson #13

- **Since we will reach the mileston of ½ of the course, we will do a simulation of the exam to check your preparation and your method for studying**

Lesson outline

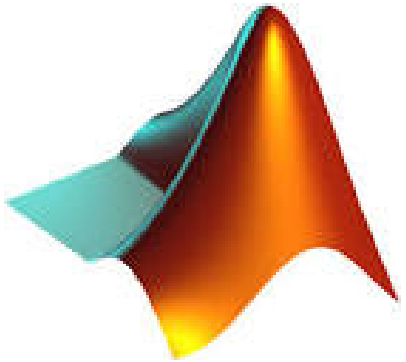
- Similarity as a tool for deep learning
- Exploratory Data Analysis
- Feature engineering
- Data preprocessing:
  a statistical point of view
- **Main points**

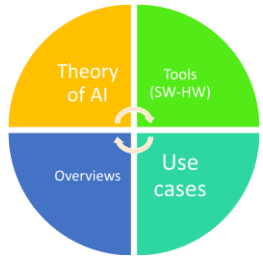



Toolboxes

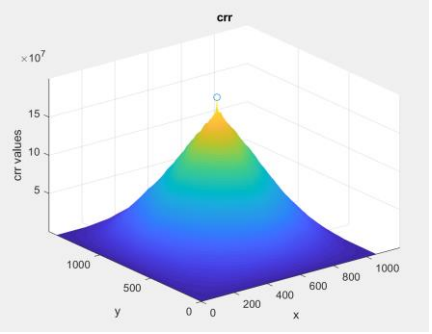
Matlab

Homework solutions…

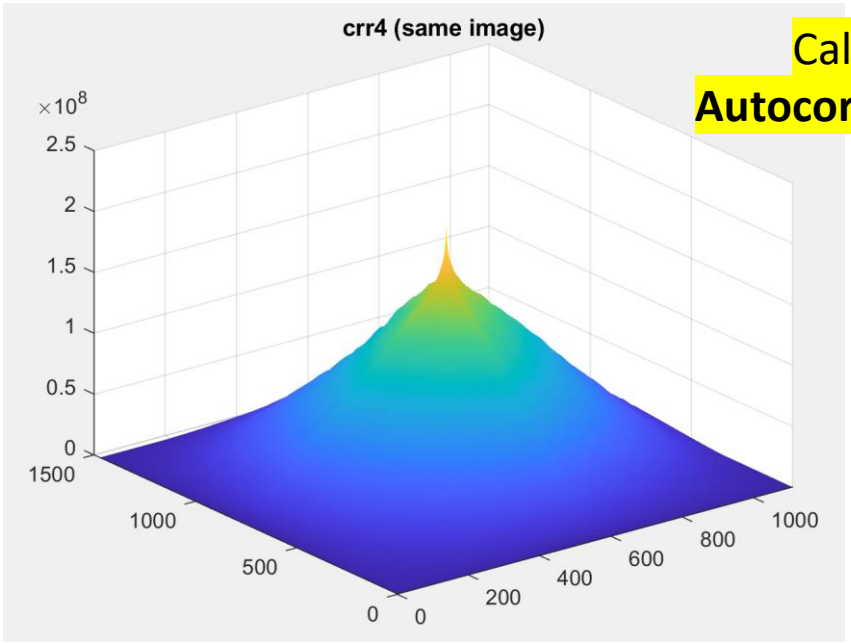




Homework #1



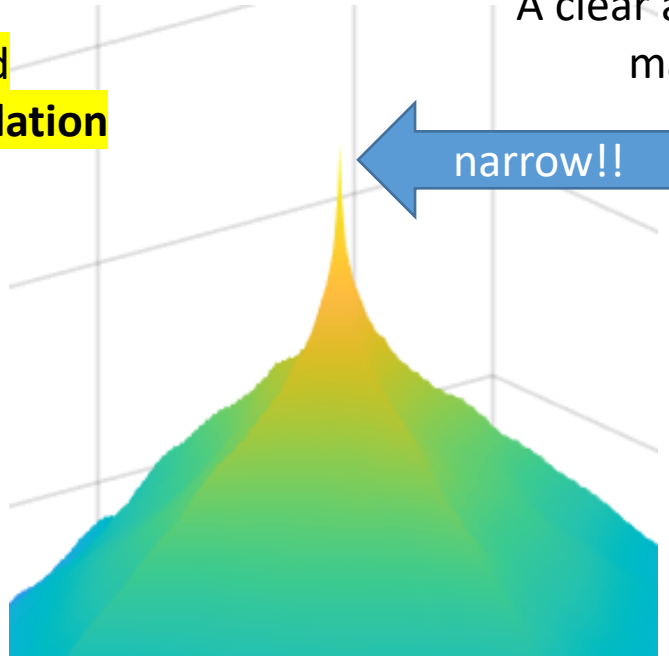
- How is changing the CRR if you put in input the same template image as subimage
  - What about the maximum?
  - What about the shape of the CRR?



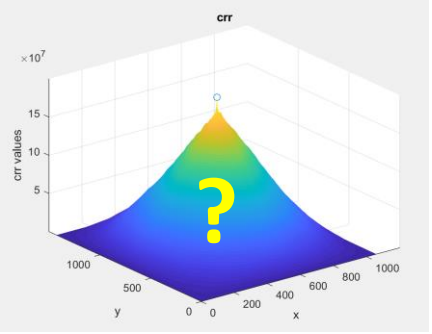
Called
Autocorrelation

A clear and separated maximum!

narrow!!



Homework #2

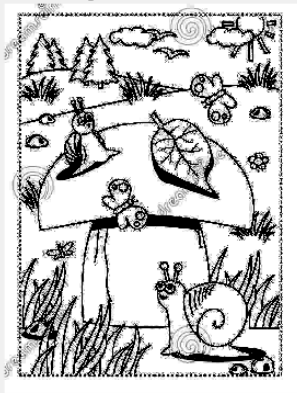


- How is changing the CRR if you put two completely different images in input?
  - What about the maximum?
  - What about the shape of the CRR?

```
%% comparison
% the moon image is builtin in matlab
img_moon_gray = imread('moon.tif');
img_moon_gray_norm = img_moon_gray - mean(mean(img_moon_gray)) ;
img_template_gray_norm = img_template_gray - mean(mean(img_template_gray));

% xcorr2(A,B) computes the crosscorrelation of matrices A and B.
crr2 = xcorr2(double(img_template_gray_norm) , double(img_moon_gray_norm) );
```

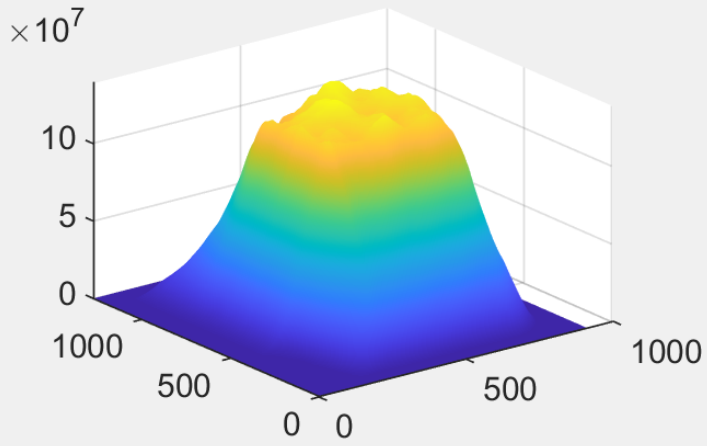
```
h3 = figure;
subplot(2,3,1);
imshow(img_template_gray_norm, []);
title('template norm.')

subplot(2,3,2);
imshow(img_moon_gray_norm, []);
title('moon norm.')

subplot(2,3,3);
surf(crr2);
shading interp;   % just to remove black line around the tiles
title('crr2')
```

template norm.

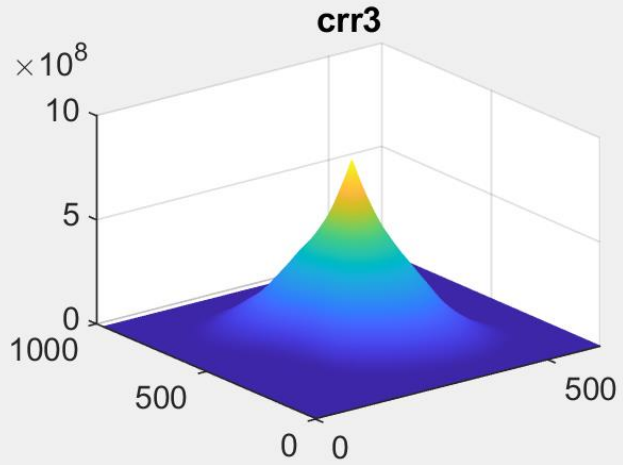


moon norm.

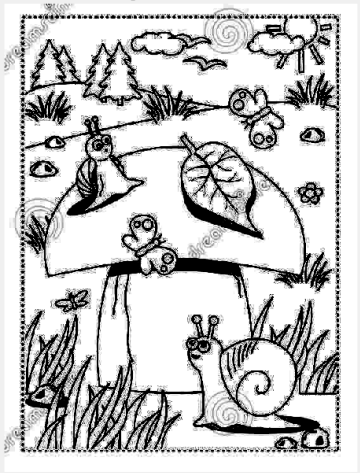
crr2



```
% xcorr2(A,B) computes the crosscorrelation of matrices A and B.
img_crop = img_moon_gray_norm( [(1+50):(end-50)], [(1+50):(end-50)]) ;

crr3 = xcorr2( double( img_moon_gray_norm) , double(img_crop)  ) ;


subplot(2,3,4);
imshow(img_moon_gray_norm, []);
title('moon norm.')

subplot(2,3,5);
imshow(img_crop, []);
title('moon crop norm.')

subplot(2,3,6);
surf(crr3);
shading interp;   % just to remove black line around the tiles
title('crr3')
```

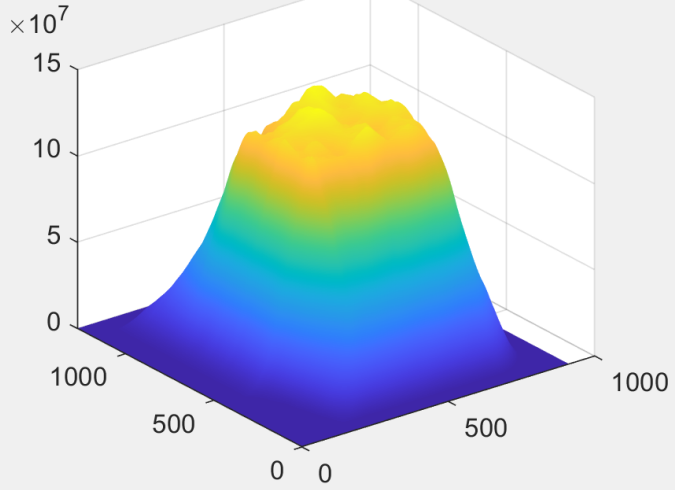


template norm.



moon norm.

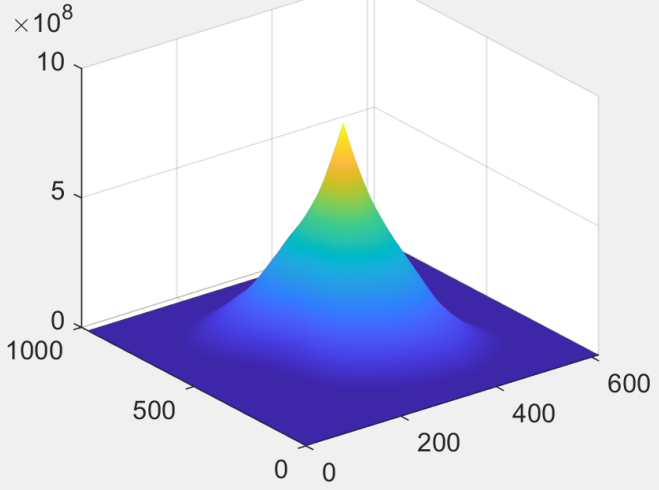
crr2



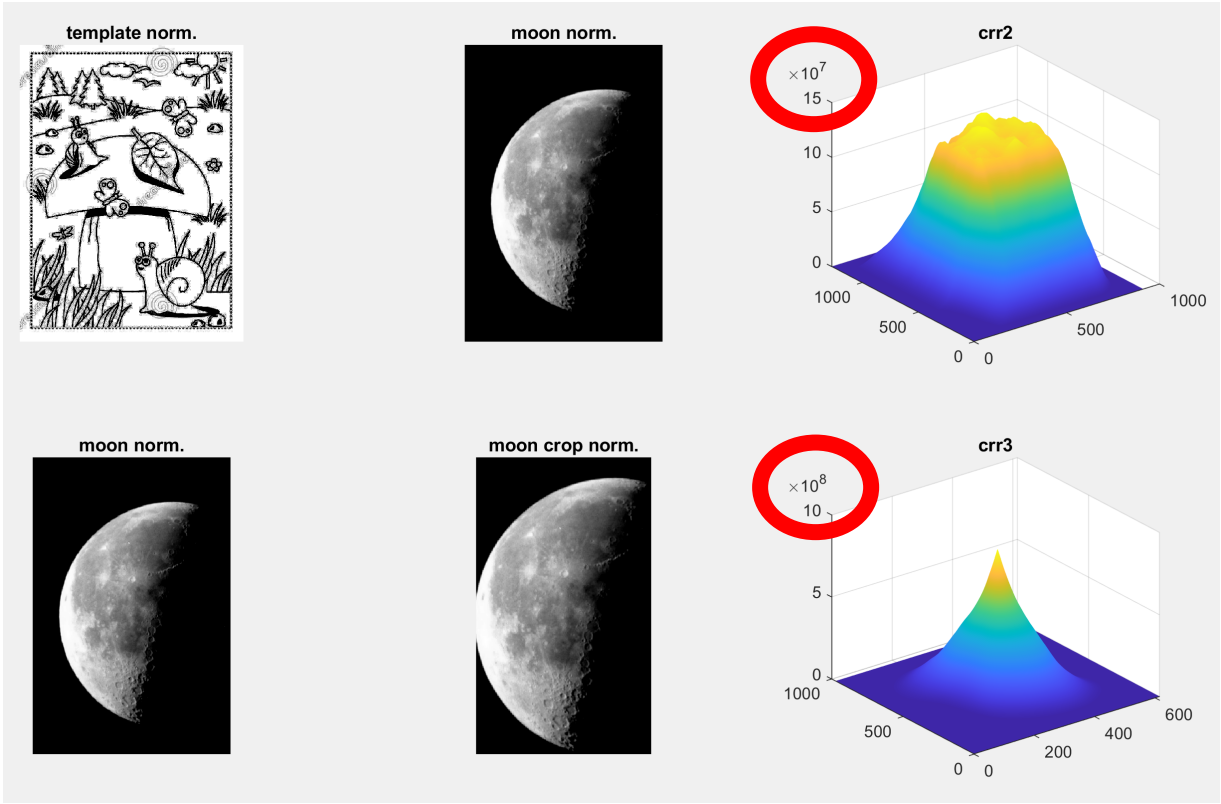
moon norm.

moon crop norm.

crr3



Max comparison → normalization with respect to number of pixels is required



$$G = h \otimes F \qquad G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i+u, j+v]$$

Matlab

Max comparison → normalization with respect to number of pixels is required

Normalized cross-correlation = (NCC) $t * f = \frac{1}{n} \sum_{x,y} \frac{1}{\sigma_f \sigma_t} (f(x,y) - \mu_f)(t(x,y) - \mu_t).$

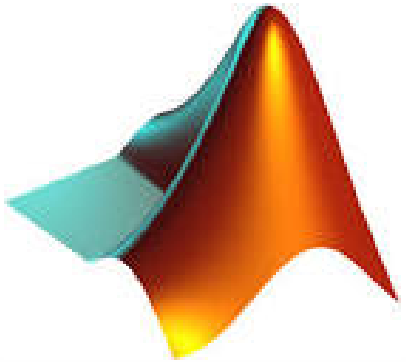
1/n

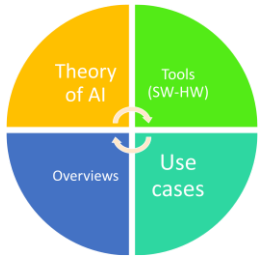



Toolboxes/Theory

Similarity via conv.

The basis to Convolutional NNs.





Similarity ←→ Cross-correlation (convolution)

- It is important to understand very well the concept of similarity analysis via cross-correlation (convolution)
  → used in for Convolutional Neural Networks
  (CNNs, one of the most famous deep learning models)

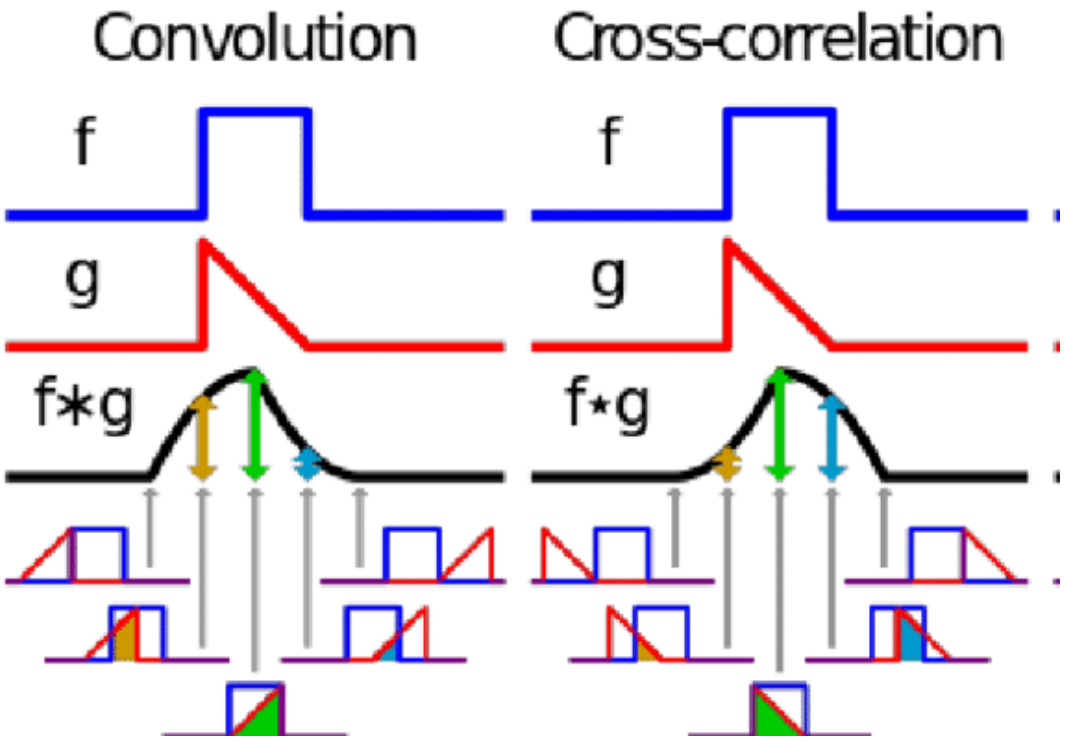
Cross-Correlation:

$$G = h \otimes F \qquad G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i+u, j+v]$$

Just the minus!

Convolution:

$$G = h * F \qquad G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i-u, j-v]$$



Cross-Correlation:

$$\text{CRR} = h \otimes F \qquad \text{CRR}[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i+u, j+v]$$

Just the minus!

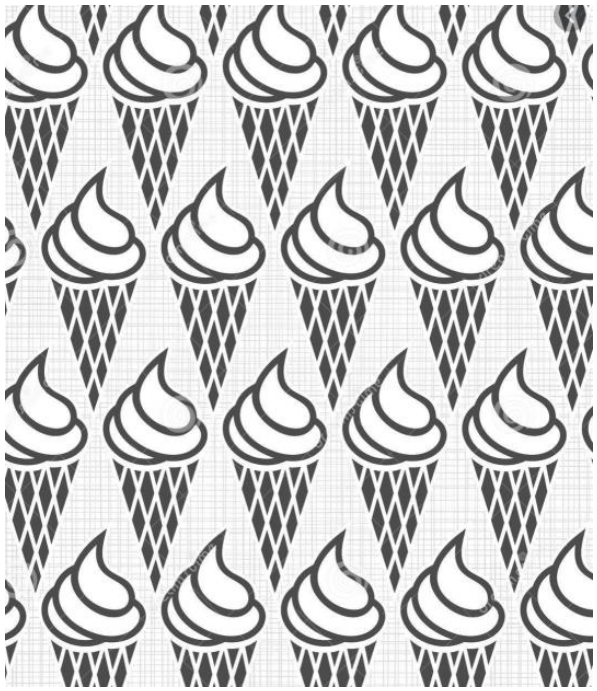
Convolution:

$$\text{CONV} = h * F \qquad \text{CONV}[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i-u, j-v]$$

Similarity…

- Cross correlation and Internal similarity

(approaching the basis of CNNs)

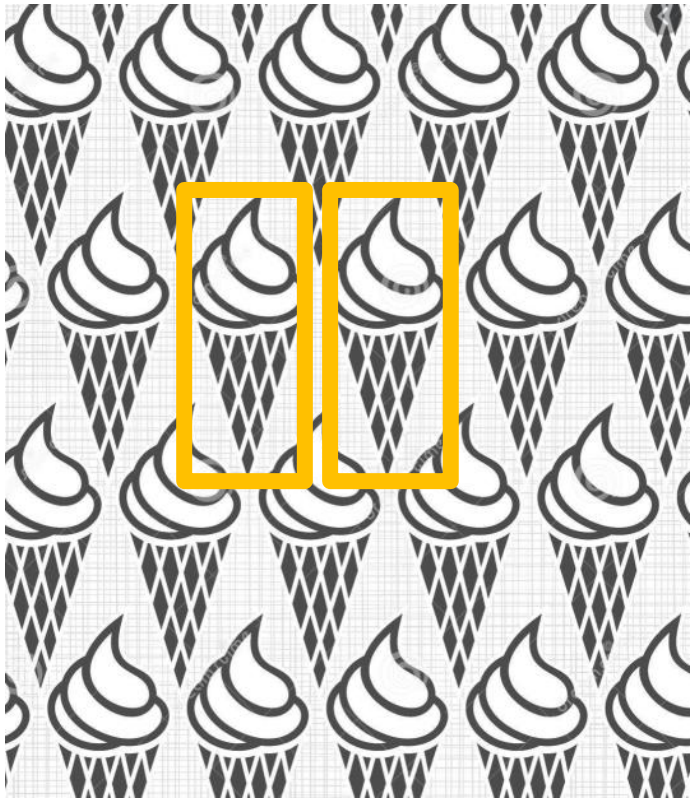


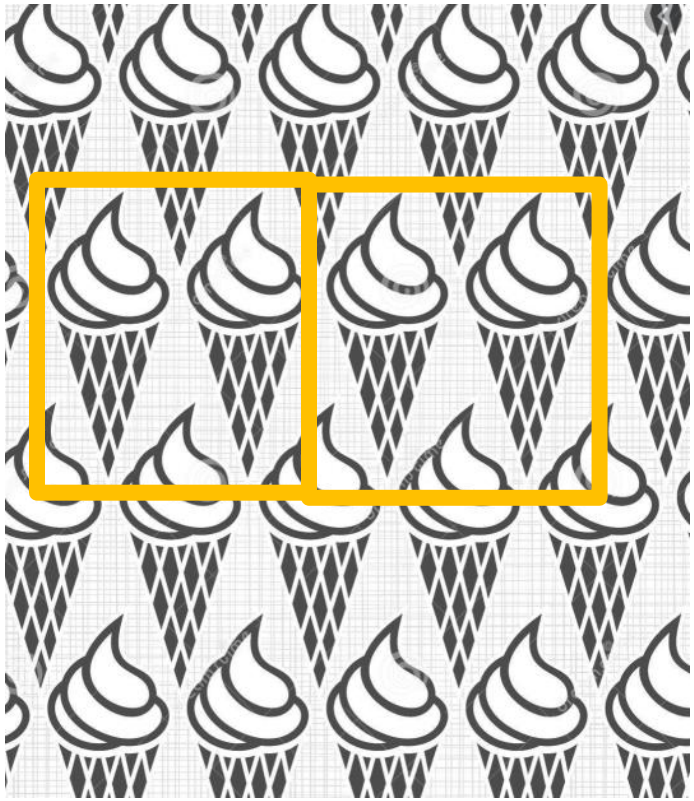
```
%% similarity
img_cream = double(rgb2gray( imread( 'ice_creams.jpg')));
img_cream_norm =  img_cream - mean(mean(img_cream));


h5 = figure
crr5 = xcorr2( double( img_cream_norm) , double(img_cream_norm)  ) ;
surf(crr5);
shading interp;   % just to remove black line around the
title('crr5 (same image)')
```

AUTOCORRELATION

Similarity…

- Internal similarity

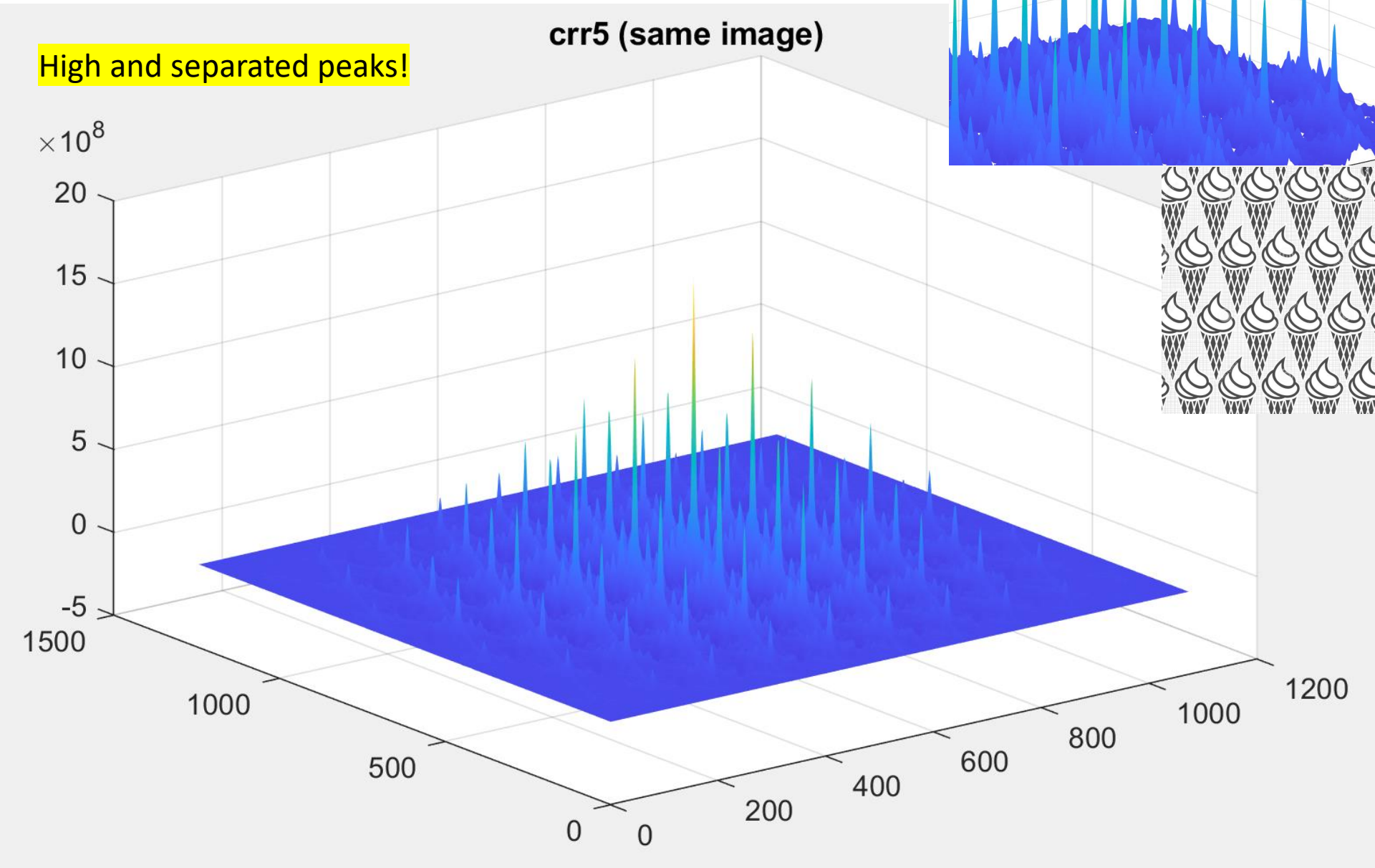


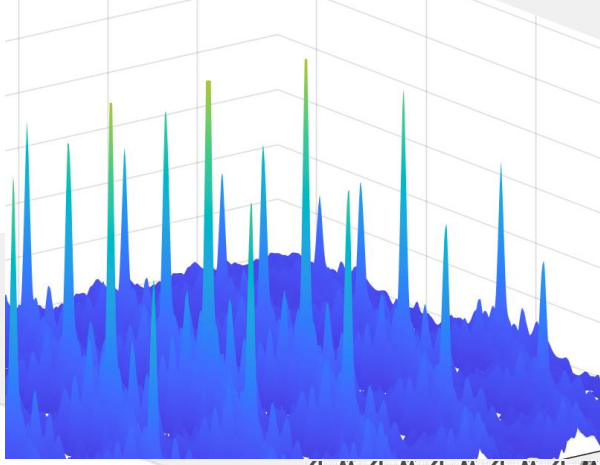


…

Similarity…

High and separated peaks!

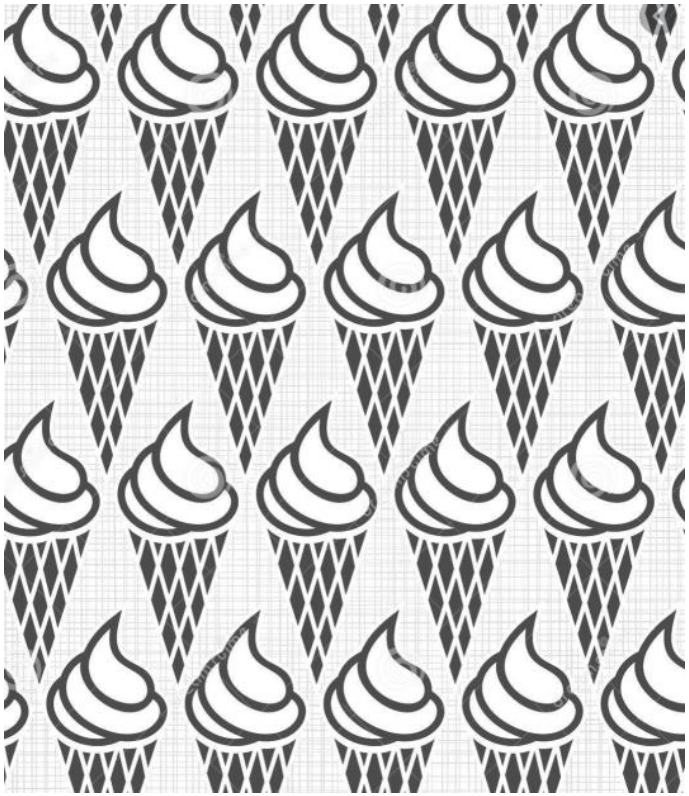




Similarity…

General similarity between patterns…
Not just in images…. in data!

Images are different! (but similar)



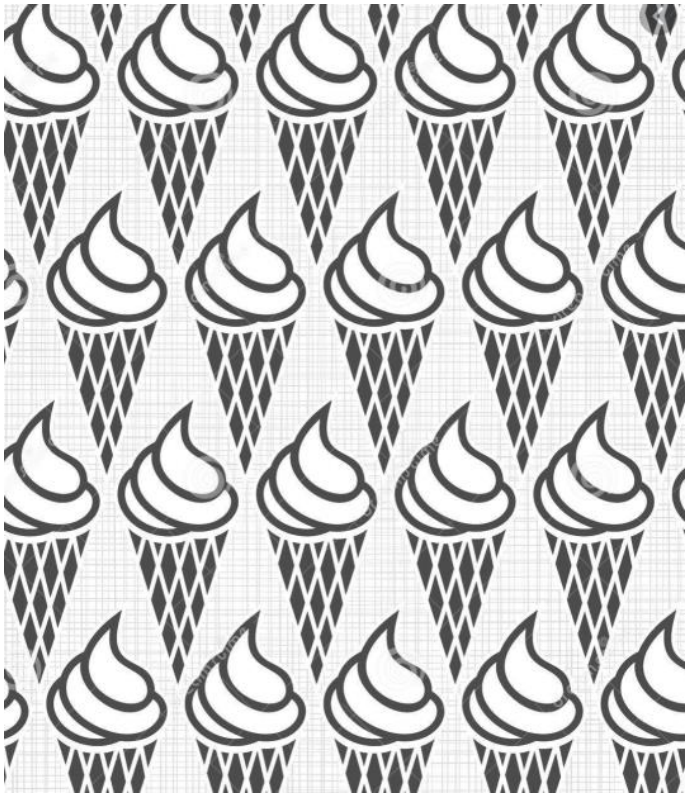
Find it!!

«KERNEL» of our search

Similarity…

General similarity between patterns…
Not just in images…. in data!

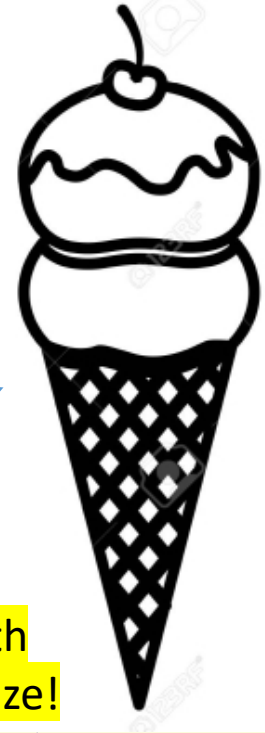
Images are different! (but similar)



Find it!!

Let's ZOOM «KERNEL» of our search just for visualization

Prepare a similar pattern
(think to “patterns” not just an image…)

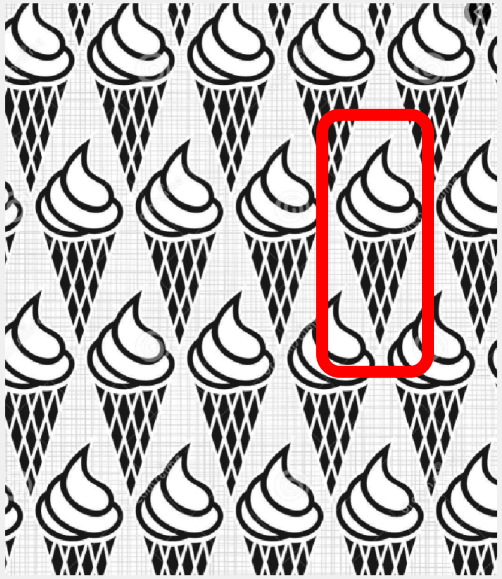


also with
similar size!

```
%% similarity in CNN...
img_single = double(rgb2gray( imread( 'icecream2.jpg')));
img_single_norm =  img_single - mean(mean(img_single));


h6 = figure;
subplot(1,3,1); imshow( img_cream_norm, []); title('ice creams');
subplot(1,3,2); imshow( img_single_norm, []); title('similar single ice cream (kernel)');
crr6 = xcorr2( double( img_cream_norm) , double(img_single_norm)  ) ;
subplot(1,3,3);
surf(crr6);
shading interp;   % just to remove black line around the
title('Crr')
```

We found the similar pattern!

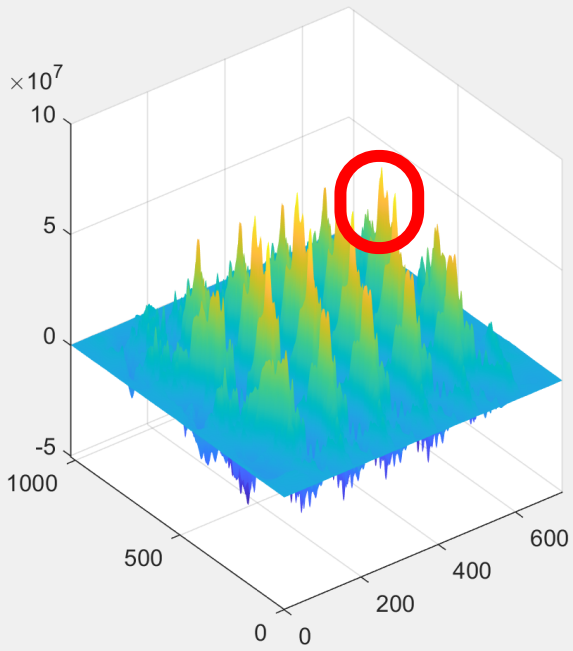
ice creams



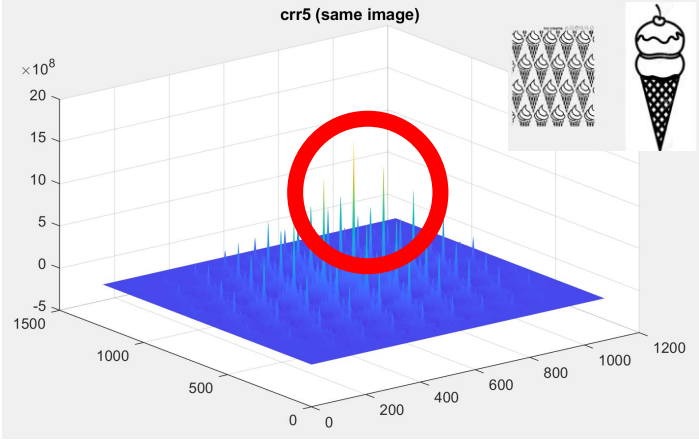
similar single ice cream (kernel)

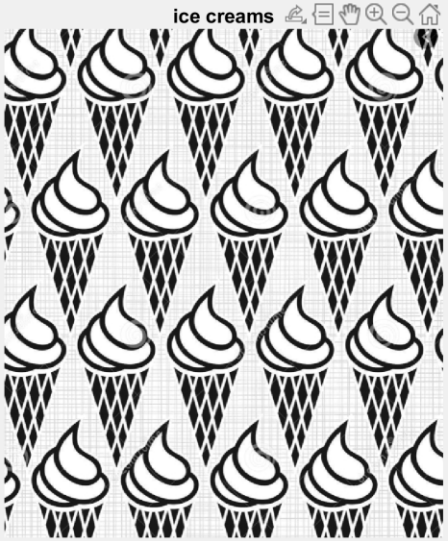
Note: this image is zoommed for visualizazion

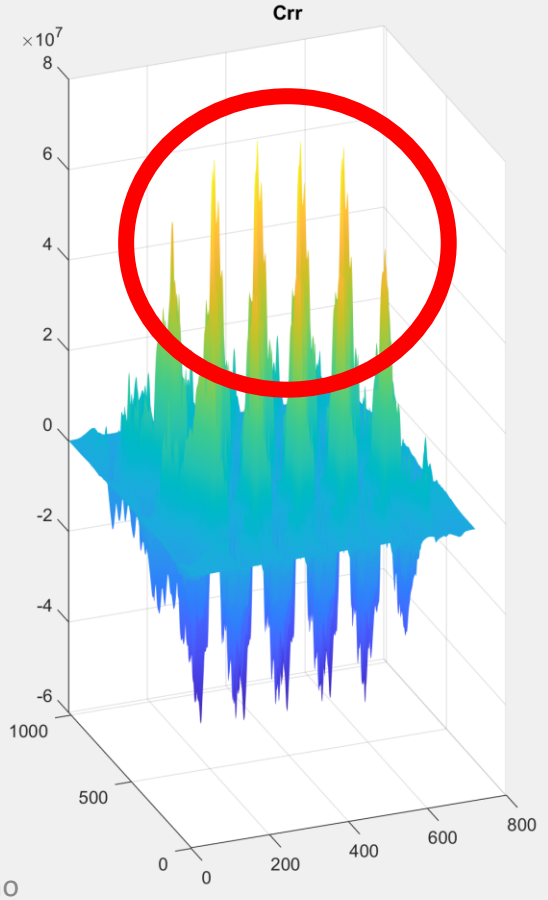
Crr



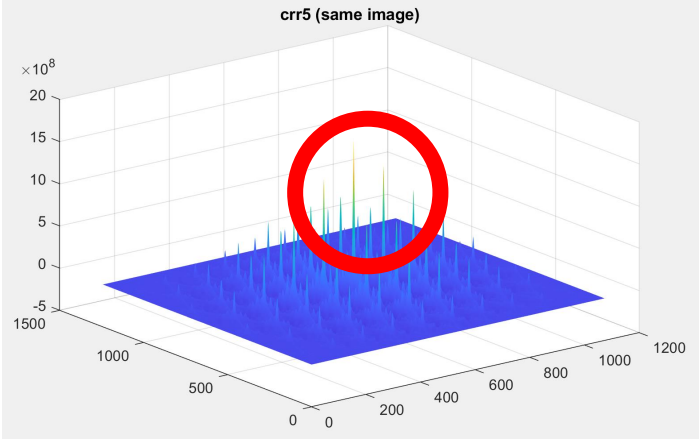
Peaks in the CCR



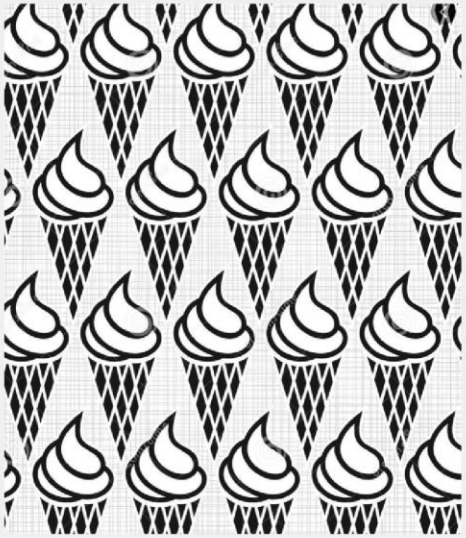




Rebuttal with a panda!
→Noise in the CCR

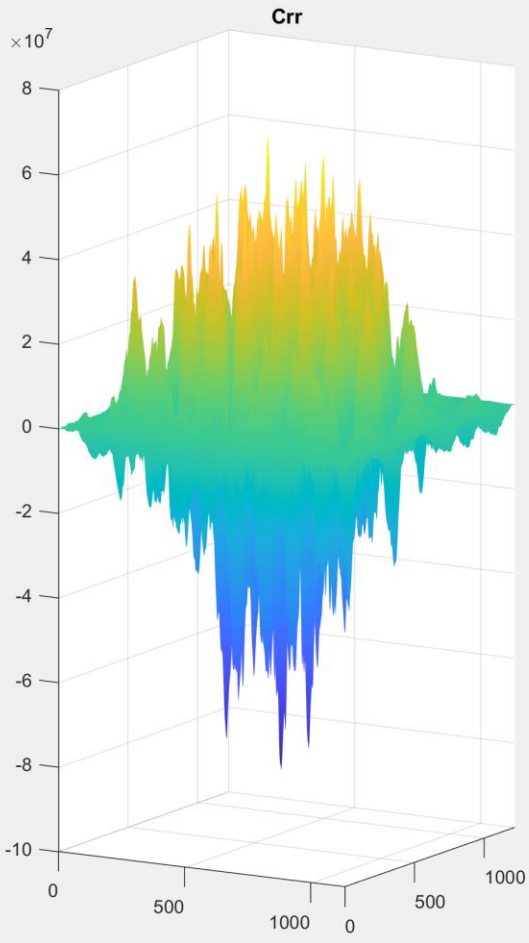


ice creams

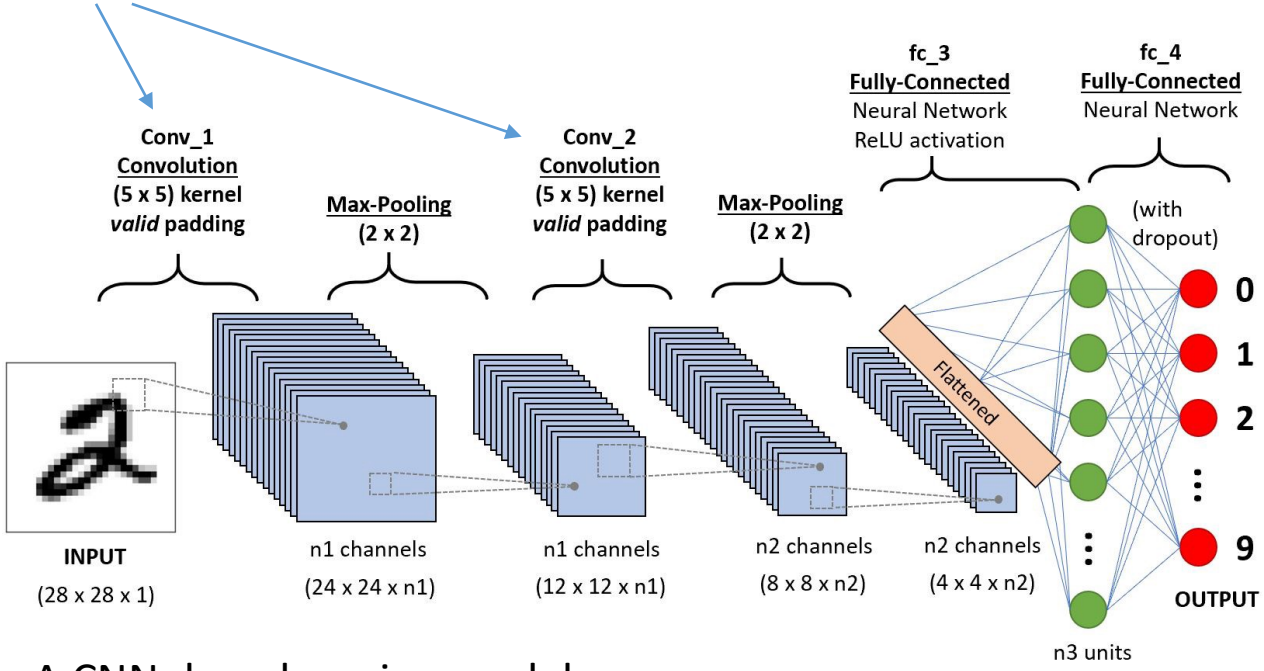


a panda (kernel)

Nevertheless, we have (noisy) peaks, since the panda is working as blob detector



Convolution in deep learning



A CNN deep learning model

In case….

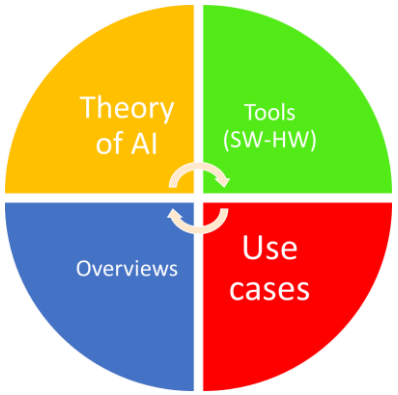
Note:
in the coding sessions
we did some simplifications
of the theory, just to get the main concepts without loosing ourselves in technicalities which are outside the scope of this course.

```
%% similarity
close all
img_cream = double(rgb2gray( imread( 'ice_creams.jpg')));
img_cream_norm =  img_cream - mean(mean(img_cream));


h5 = figure;
crr5 = xcorr2( double( img_cream_norm) , double(img_cream_norm)  ) ;
surf(crr5);
shading interp;   % just to remove black line around the
title('crr5 (same image)')


% similarity in CNN...
img_single = double(rgb2gray( imread( 'icecream2.jpg')));
img_single_norm =  img_single - mean(mean(img_single));


h6 = figure;
subplot(1,3,1); imshow( img_cream_norm, []); title('ice creams');
subplot(1,3,2); imshow( img_single_norm, []); title('similar single ice cream (kernel)');
crr6 = xcorr2( double( img_cream_norm) , double(img_single_norm)  ) ;
subplot(1,3,3);
surf(crr6);
shading interp;   % just to remove black line around the
title('Crr')

%
img_single = double(rgb2gray( imread( 'panda.jpg')));
img_single_norm =  img_single - mean(mean(img_single));


h6 = figure;
subplot(1,3,1); imshow( img_cream_norm, []); title('ice creams');
subplot(1,3,2); imshow( img_single_norm, []); title('a panda (kernel)');
crr6 = xcorr2( double( img_cream_norm) , double(img_single_norm)  ) ;
subplot(1,3,3);
surf(crr6);
shading interp;   % just to remove black line around the
title('Crr')
```

THEORY

Exploratory Data Analysis

Understanding what is inside your data

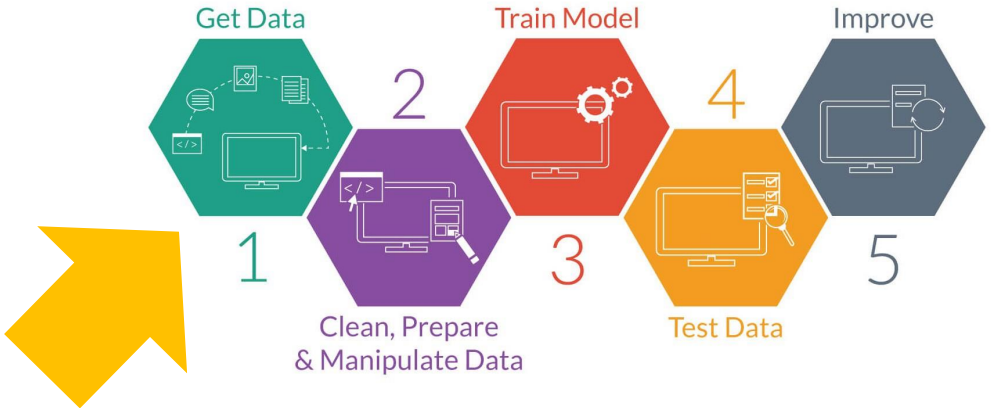


Step 1) in practice

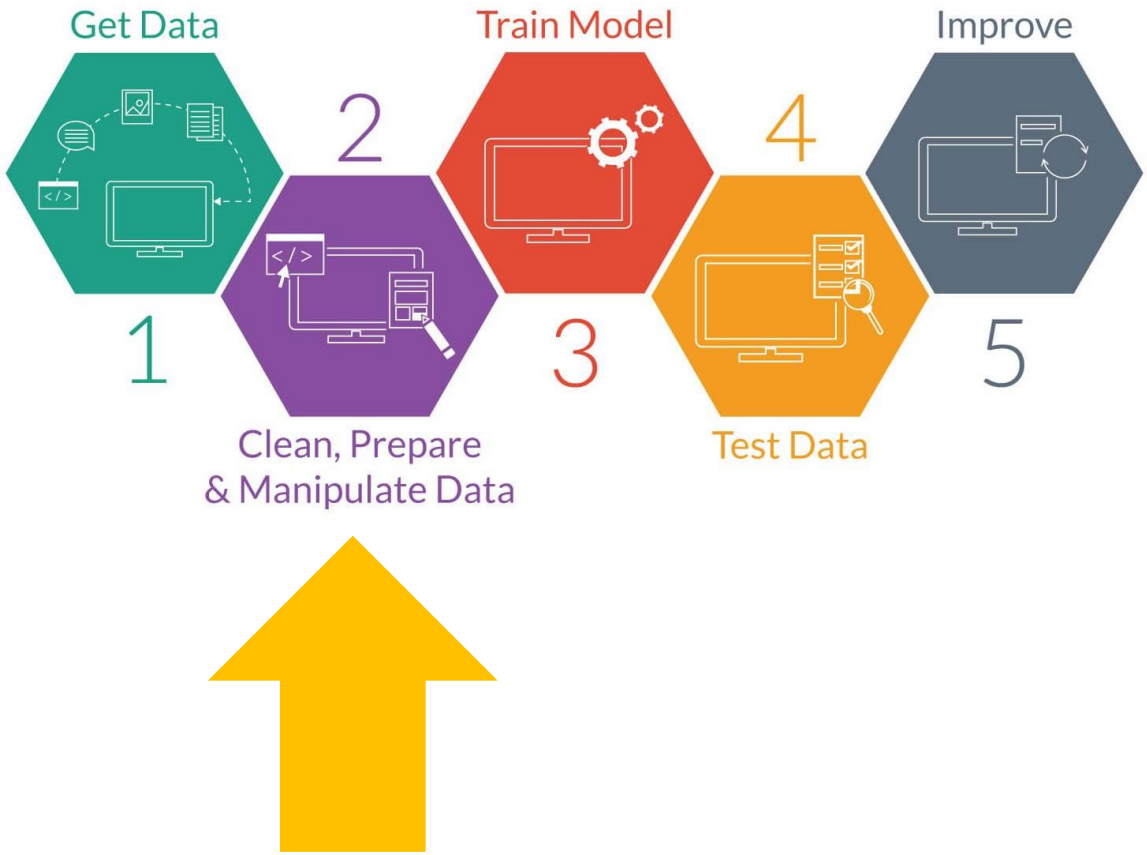
- Classical DB query
- File processing
- Custom formats
- Stream of data from IoT and AIoT
- Online platform (AWS, Azure, Google, …)

Outside the focus of this course

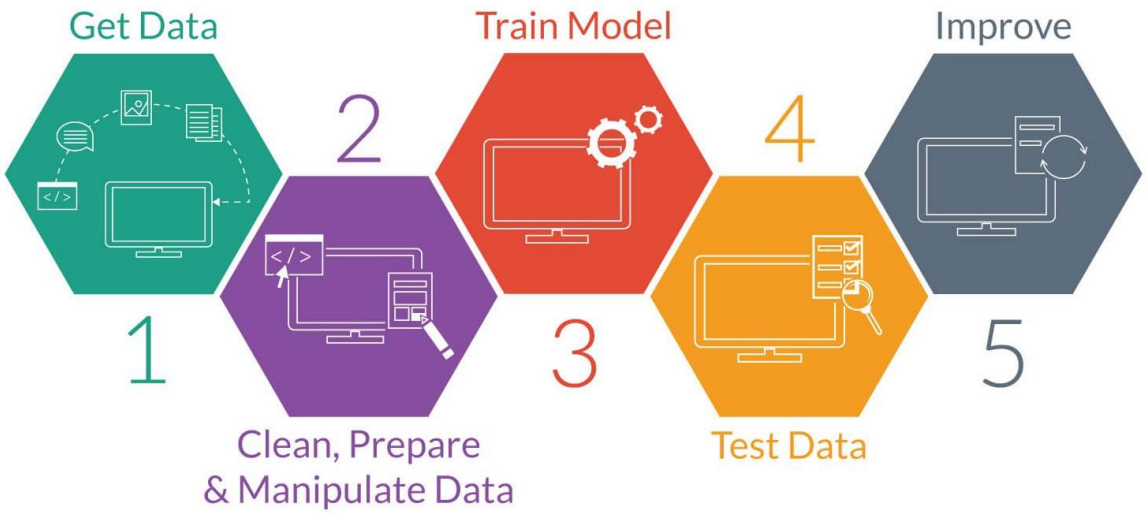
We have so many powerful sources capable to generate data



Step 2 of the ML workflow

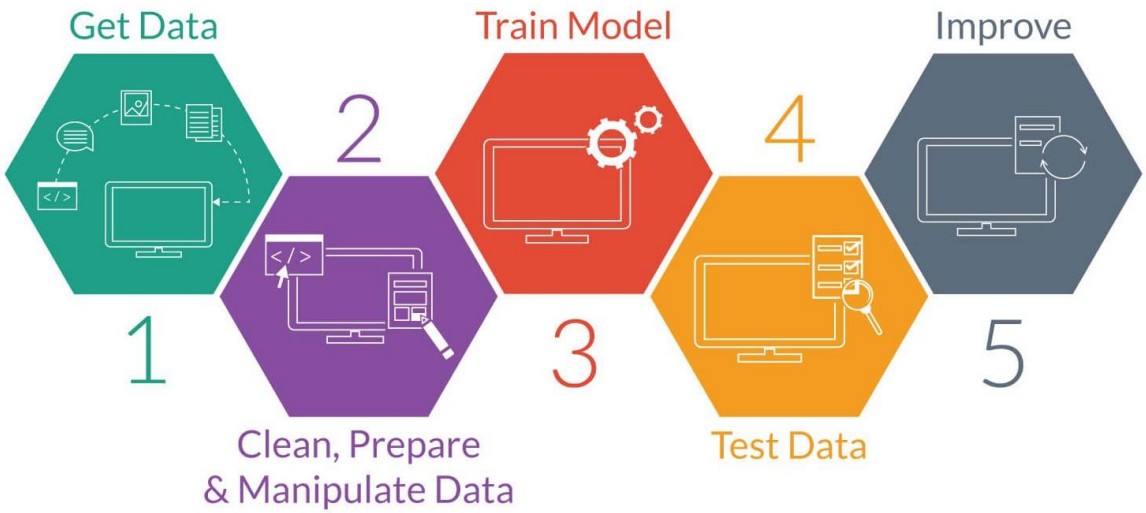


Step 2 of the ML workflow: Substeps



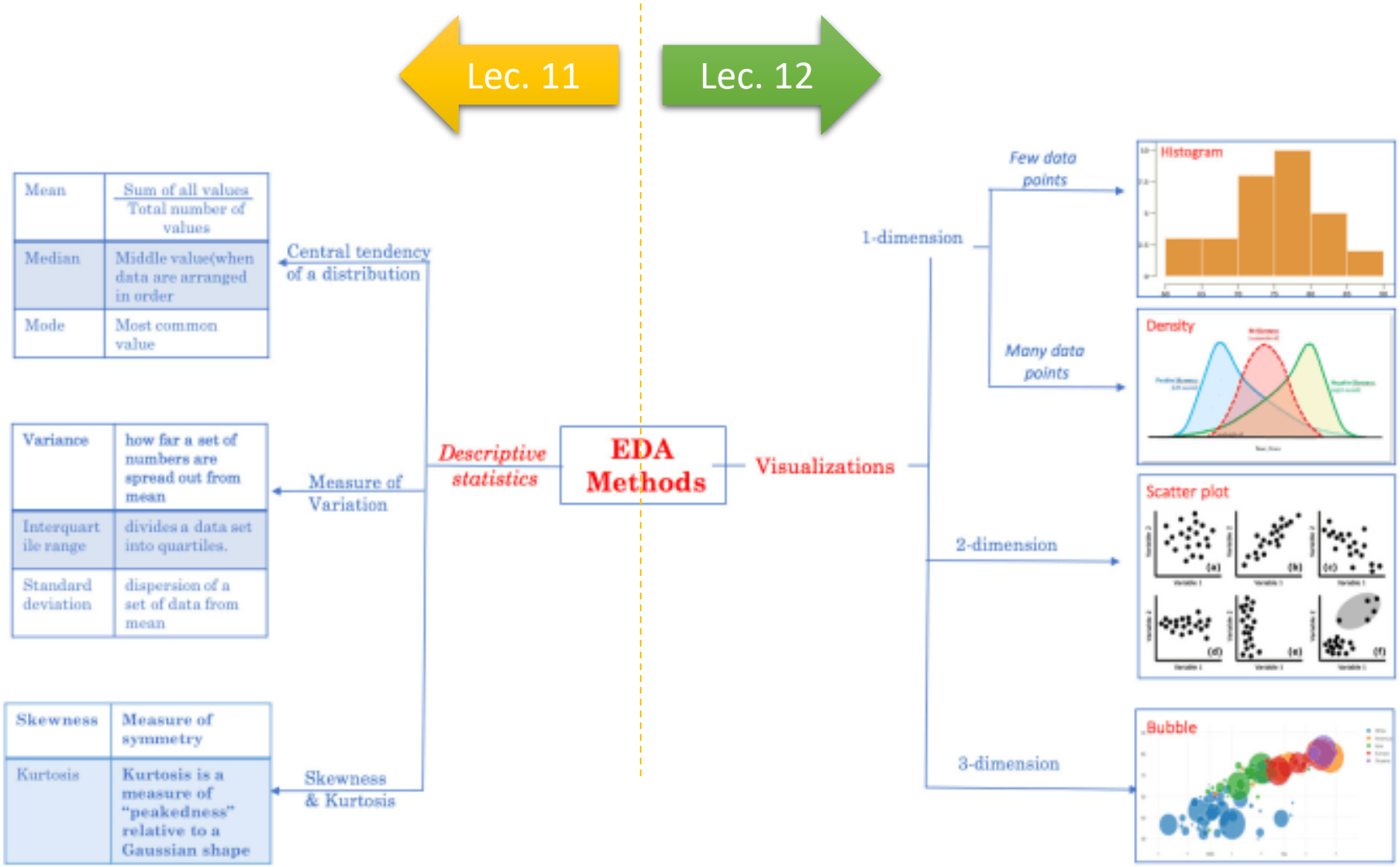
2a) Data Munging/Wrangling
2b) Exploratory Data Analysis
2c) Feature engineering

Step 2 of the ML workflow: Substeps



2a) Data Munging/Wrangling
2b) Exploratory Data Analysis
2c) Feature engineering

Exploratory Data Analysis
for ML



Different tools

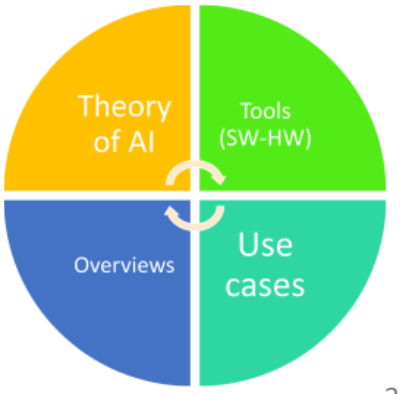
Exploratory Data Analysis in Matlab




Theory / Toolboxes

MathWorks®

Let’s start with the first tools about Exploratory Data Analysis (EDA) and Feature Engineering (FE)



Matlab's basic functions for Descriptive Statistics

Very simple and effective commands

```
data = [1 2 3 4 50];

        % The arithmetic mean of the data:
mean(data) → → ans: 12

        %The median of the data:
median(data) → → ans: 3

        % The standard deviation:
std(data) → → ans: 21.27

        % The smallest value in the data:
min(data) → → ans: 1

        % The largest value in the data:
max(data ) → →  ans: 50
```

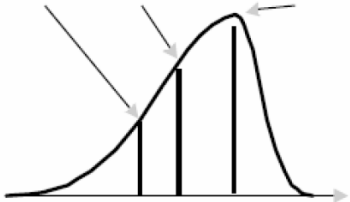
Descriptive Statistics (Exploratory Data Analysis) → Values used also in feature engineering

Shape of the distributions

1. Describes How Data Are Distributed
2. Measures of Shape
   - Skew = Symmetry

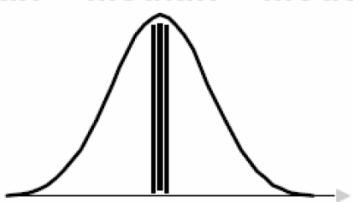
Left-Skewed

Mean Median Mode



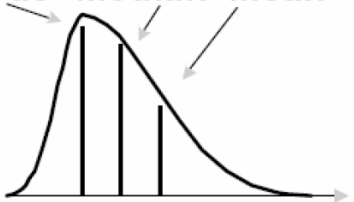
Symmetric

Mean = Median = Mode



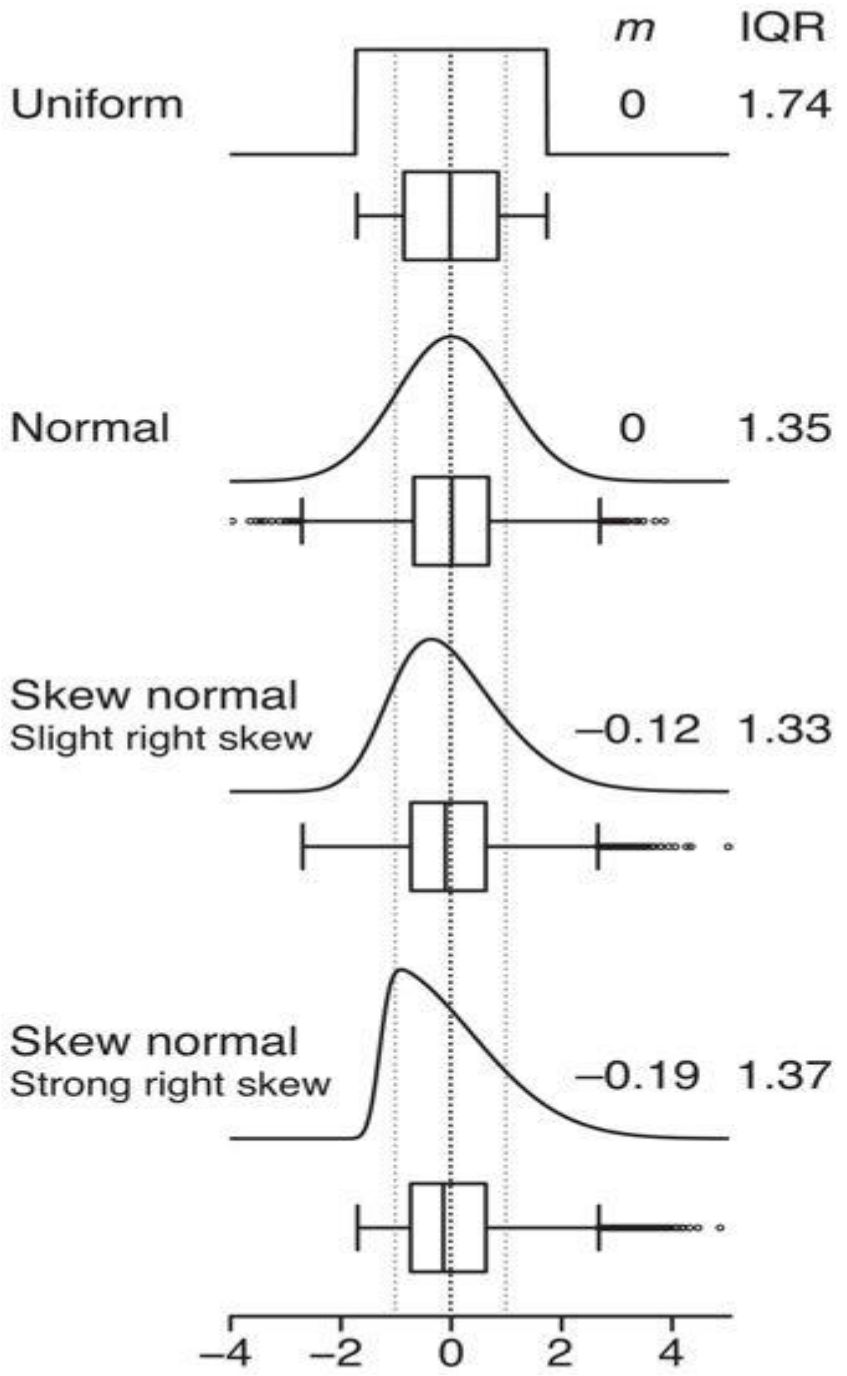
Right-Skewed

Mode Median Mean



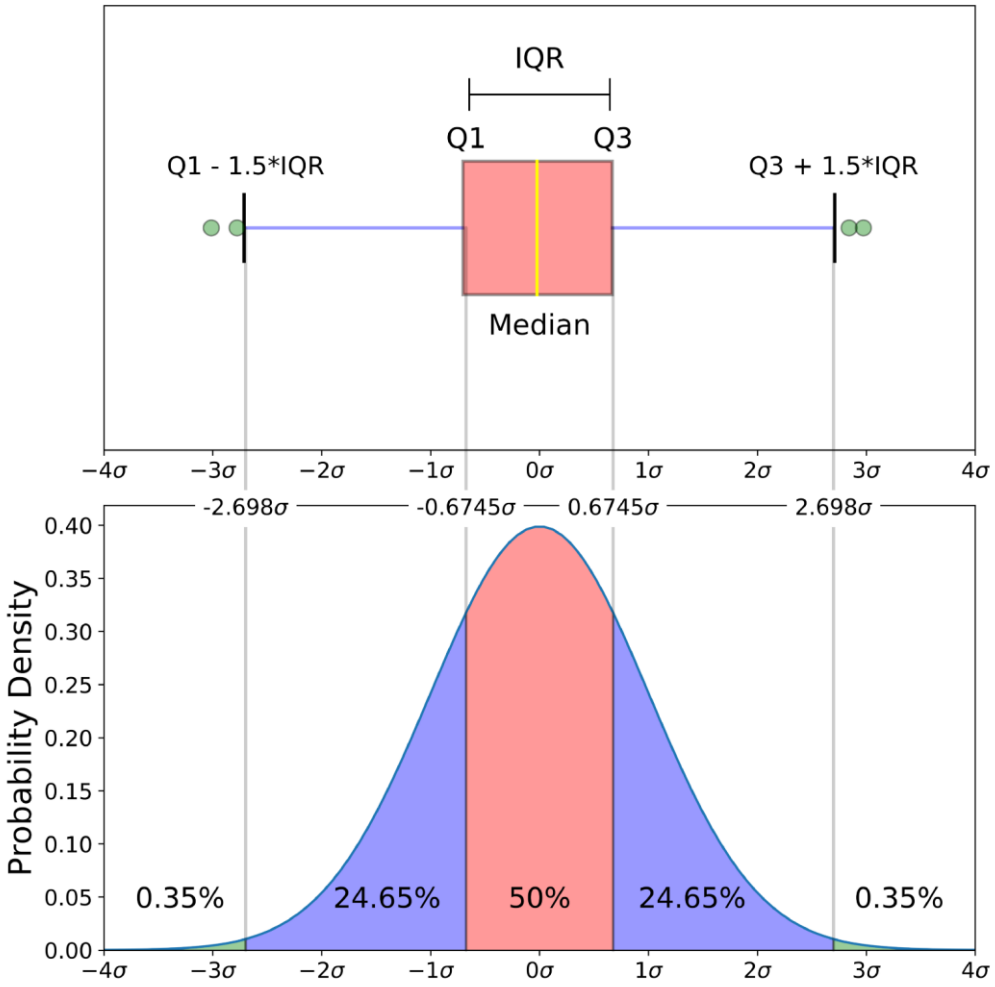
Example: in a regression error, the position and size of the tails are very important

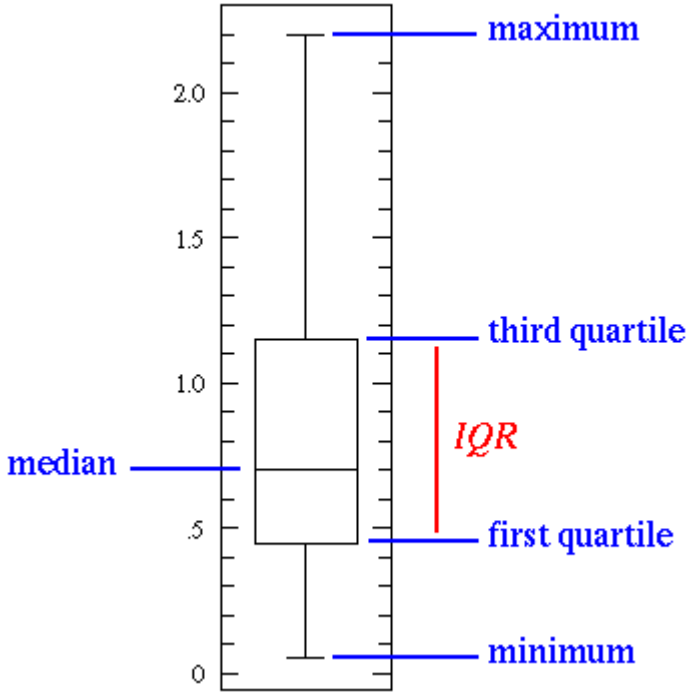
Distributions are more interesting…



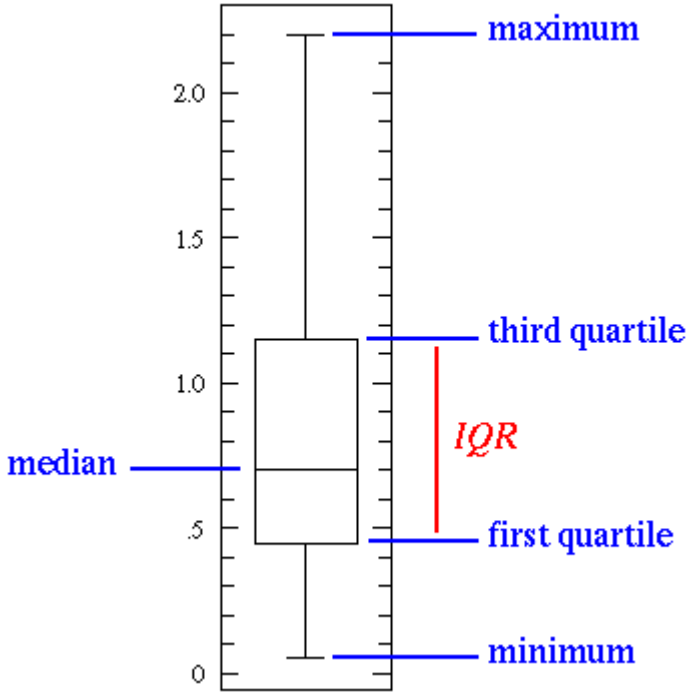
IQR = InterQuartile Range

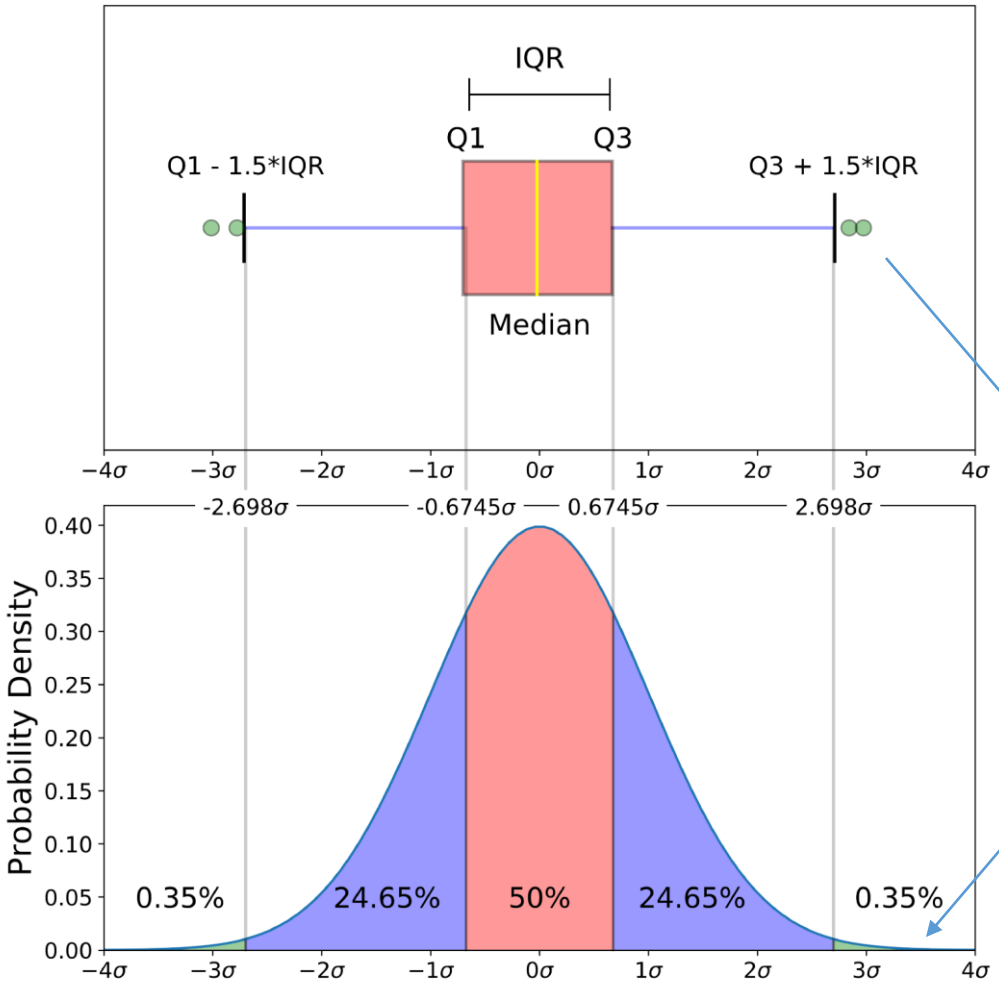
Box-plot


IQR
Q1
Q3
Q1 - 1.5*IQR
Q3 + 1.5*IQR
Median
−4σ
−3σ
−2σ
−1σ
0σ
1σ
2σ
3σ
4σ
-2.698σ
-0.6745σ
0.6745σ
2.698σ
Probability Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.35%
24.65%
50%
24.65%
0.35%


maximum
2.0
1.5
third quartile
1.0
IQR
median
.5
first quartile
0
minimum

Choose box-plots with outliers!



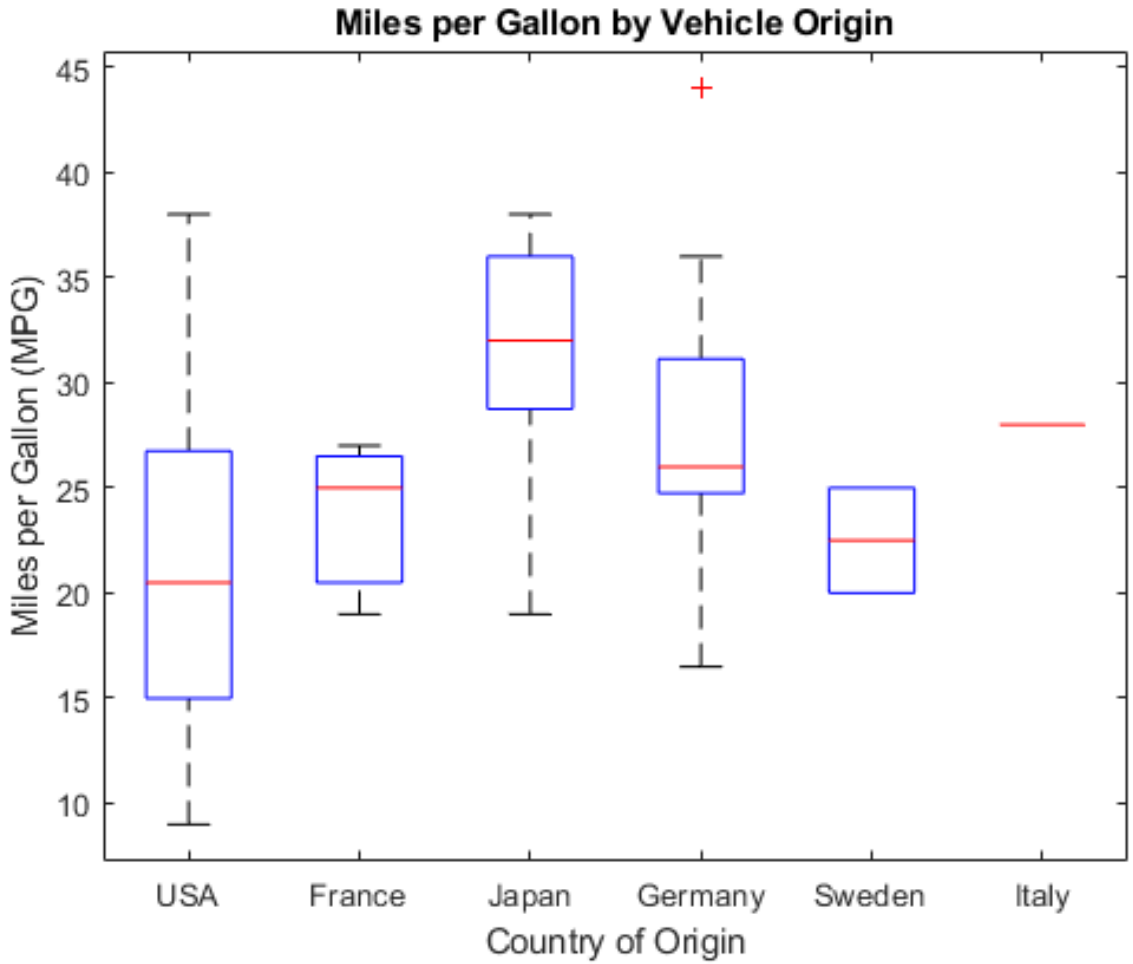


Outliers!!
Can you see them here?

Outliers can effect very much the normalization!!

The matlab boxplot

```
boxplot(MPG,Origin)

title('Miles per Gallon by Vehicle Origin')
xlabel('Country of Origin')
ylabel('Miles per Gallon (MPG)')
```

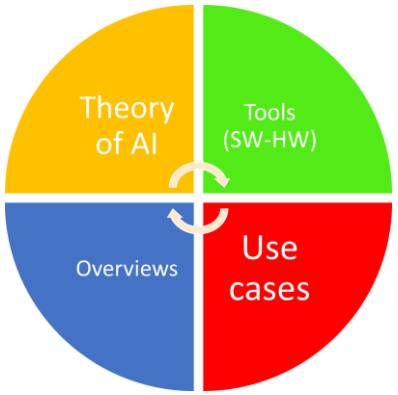




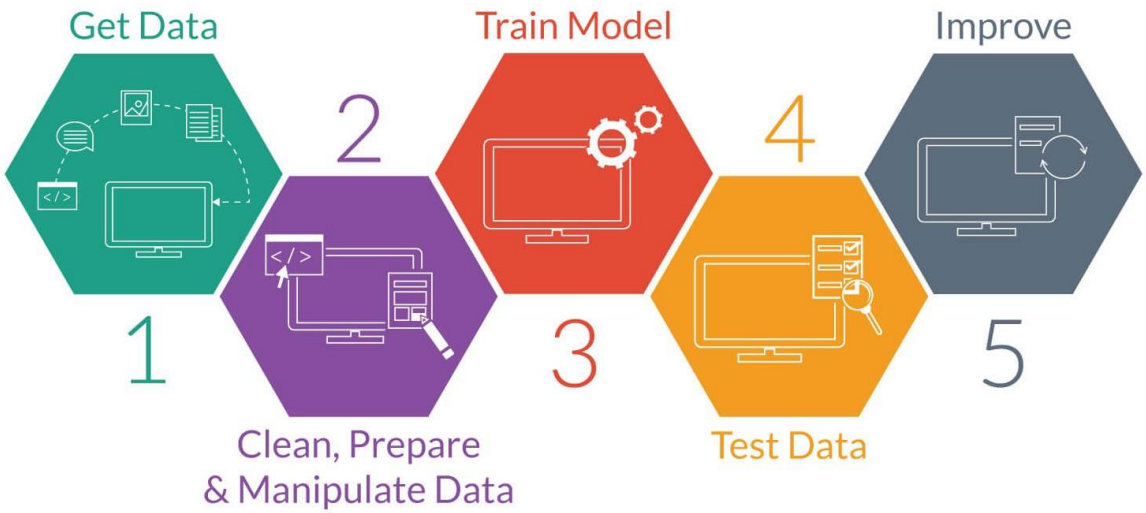

THEORY

Feature engineering

Design your features!



Step 2 of the ML workflow: Substeps

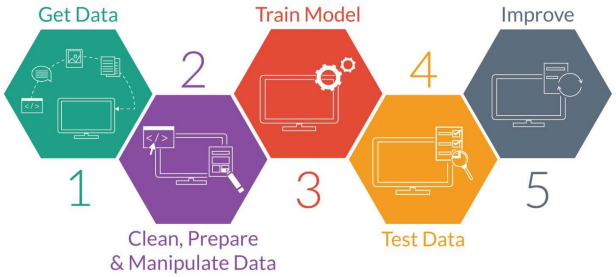


2a) Data Munging/Wrangling

2b) Exploratory Data Analysis

2c) Feature engineering

Feature engineering

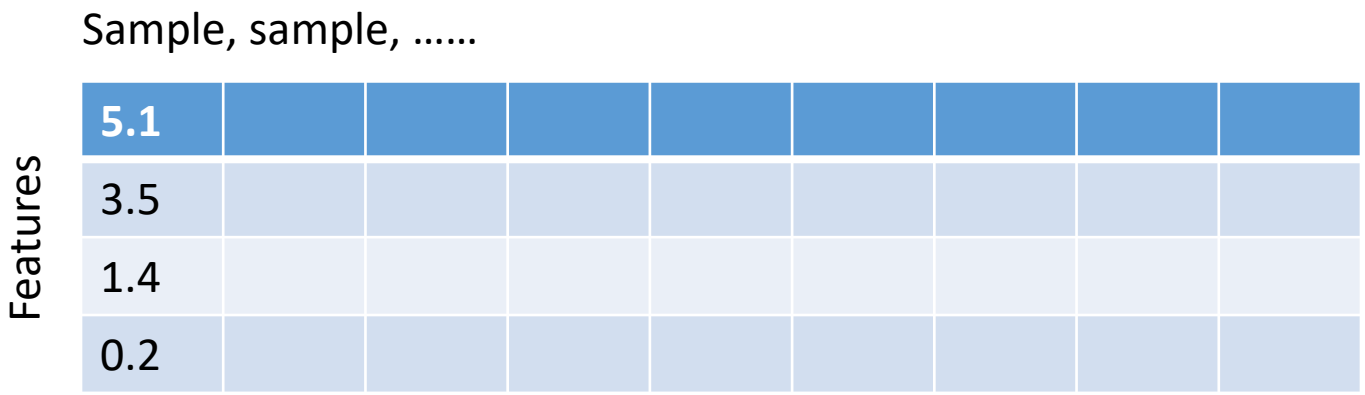


- Brainstorming or testing features
- Deciding what features to create
- Creating features
- Checking how the features work with your model
- Improving your features if needed
- Go back to brainstorming/creating more features until the work is done.

AI Model → Y = FUNC(X)

- For every dataset in machine learning or toolbox, is all about to create X and Y

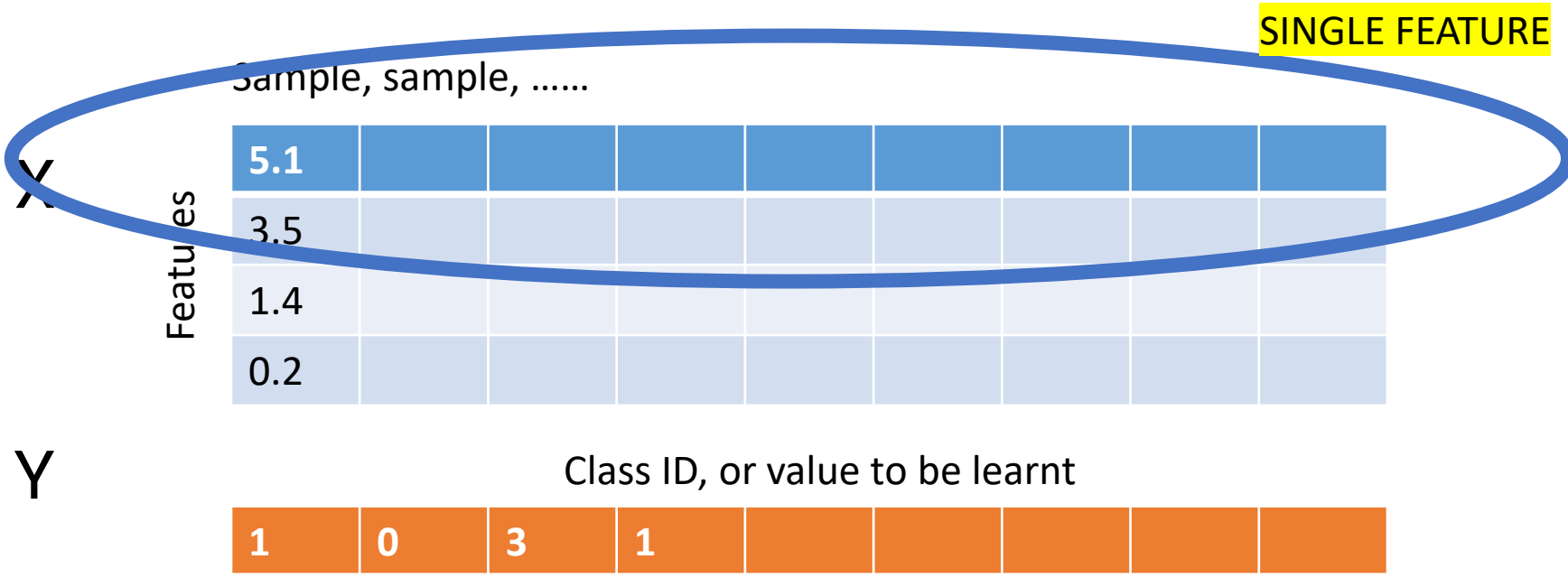
- X



- Y

Y = FUNC(X)
FEATURE ENGINEERING (A)

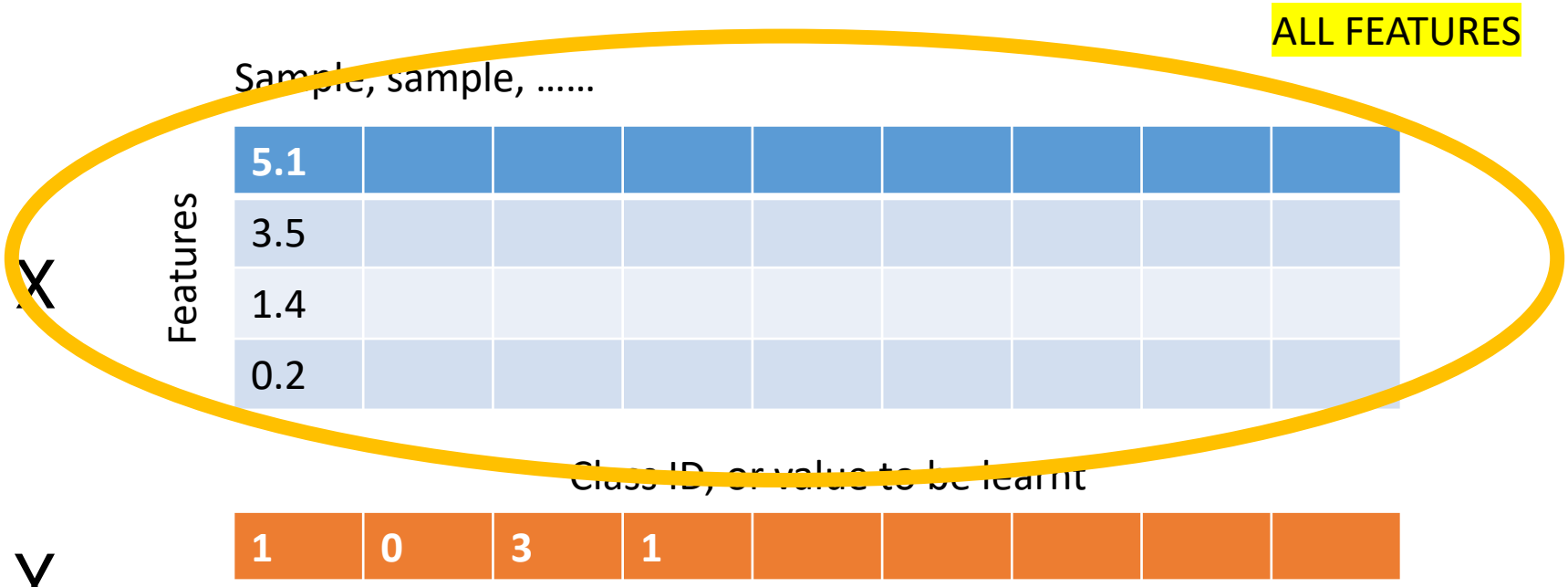
- Optimizing one single feature at time!



- X
- Y

Y = FUNC(X)
FEATURE ENGINEERING (B)

- Optimizing all features at time!

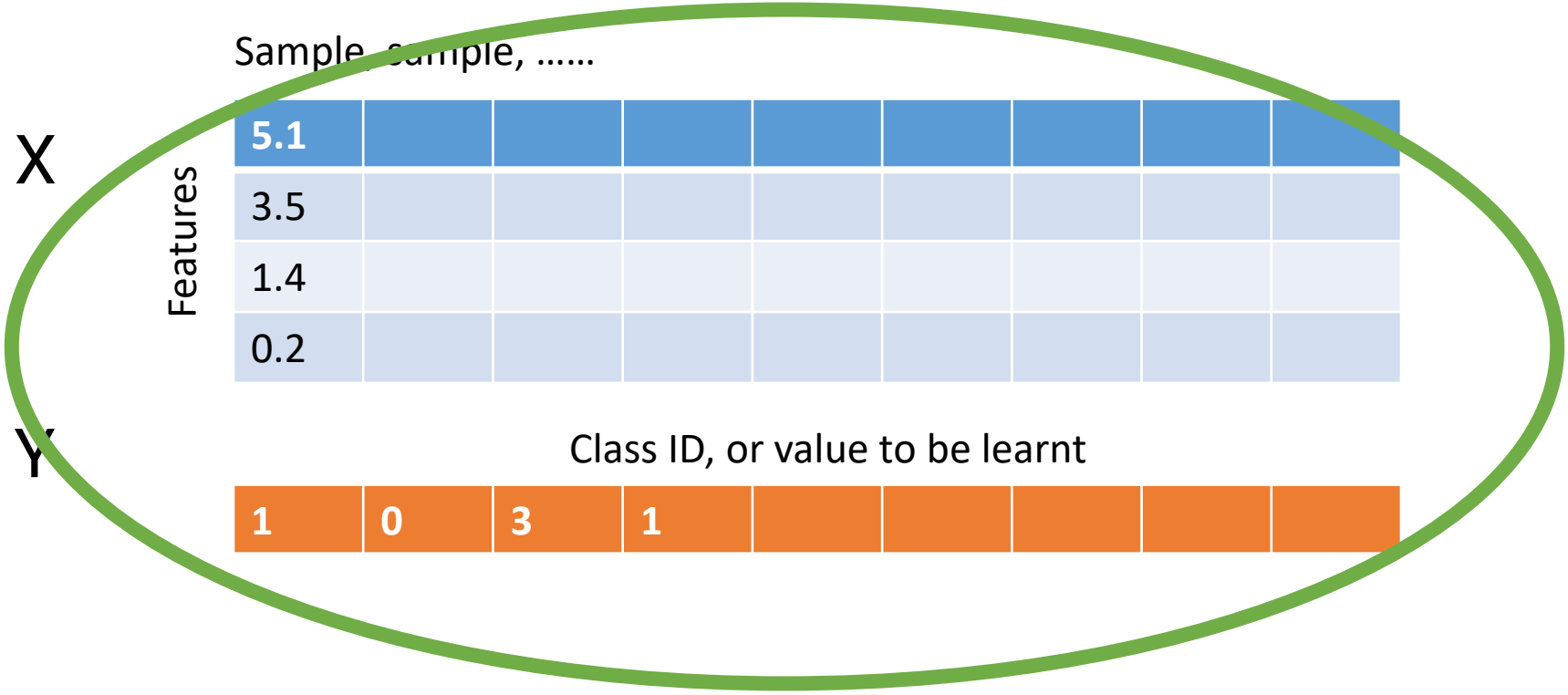


- X
- Y

Y = FUNC(X)
FEATURE ENGINEERING (C)

- Optimizing all the dataset at time!

- X
- Y



Data preprocessing: a **statistical point of view**

For intelligent systems

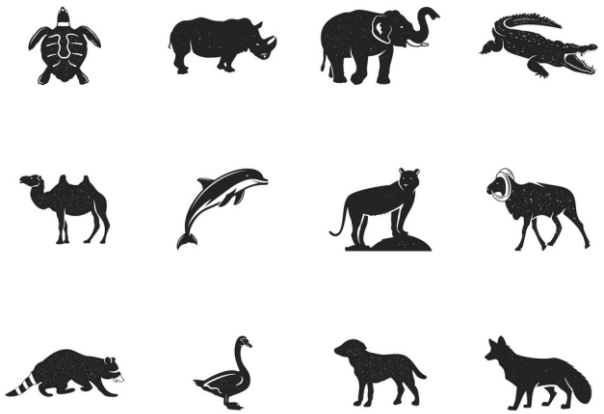
Note:
part of Exploratory Data Analysis (EDA)
and part of Feature Engineering (FE)




Feature engineering:

Normalization

For intelligent systems datasets



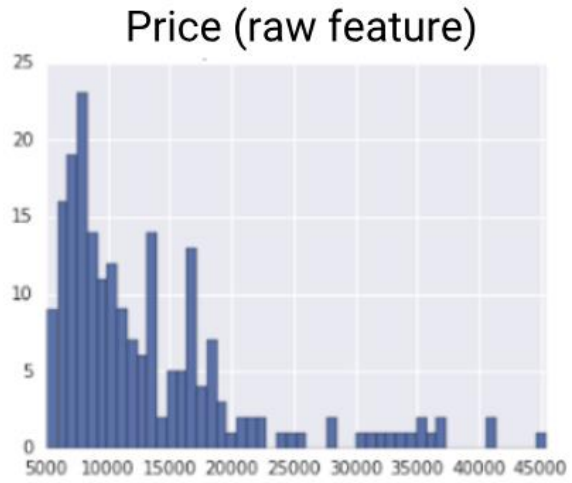
(shapes not sizes…)

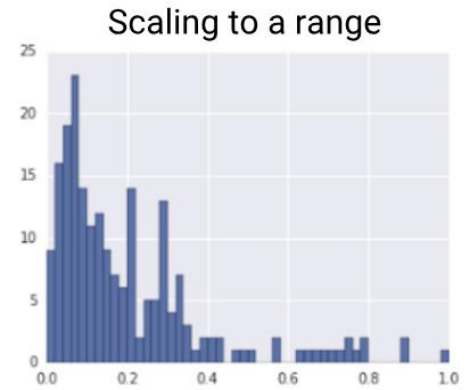
Normalization

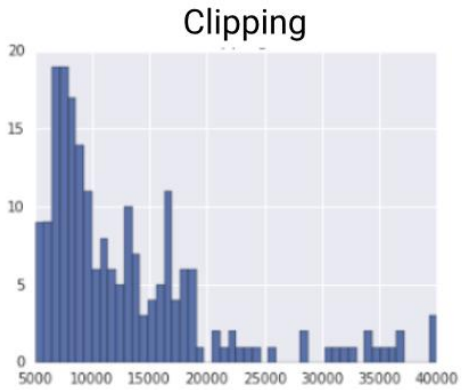
- The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model.

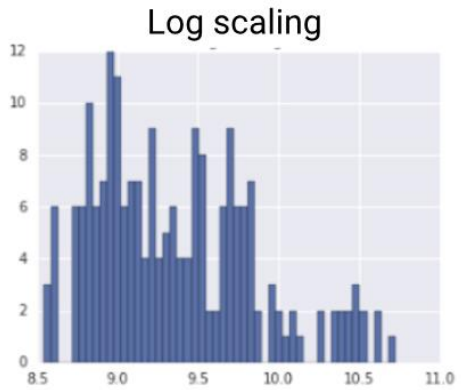
- 4 common normalization techniques:
  - scaling to a range $x' = (x - x_{min})/(x_{max} - x_{min})$
  - clipping
  - log scaling $x' = log(x)$
  - z-score $x' = (x - \mu)/\sigma$

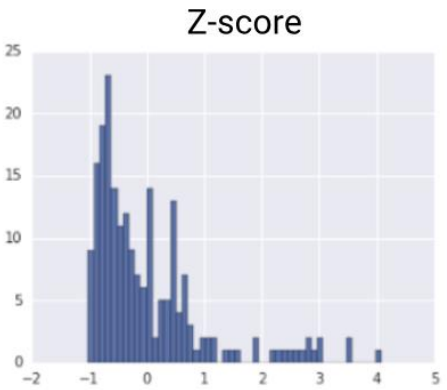
Examples of normalization











Feature Clipping

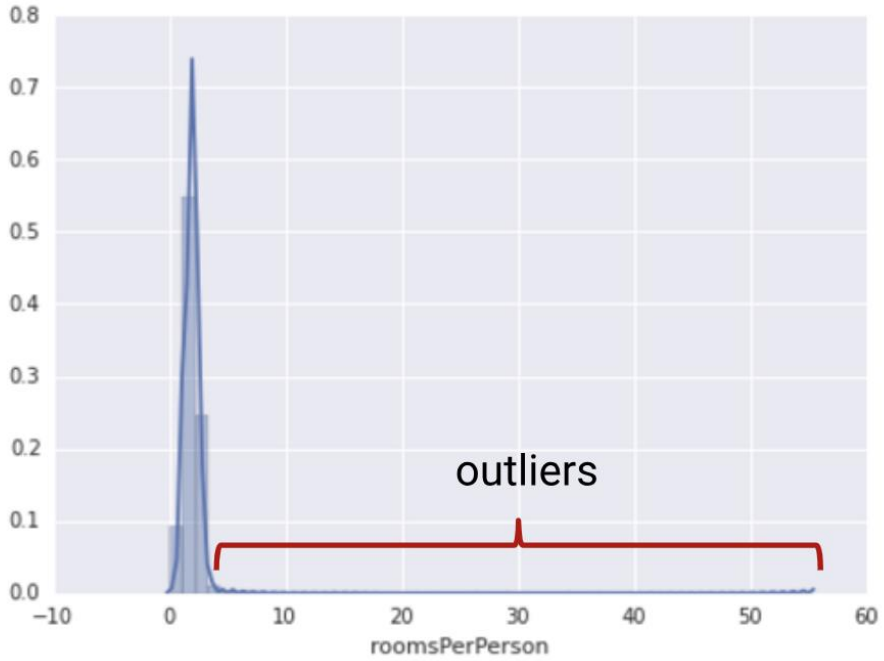
Hint!

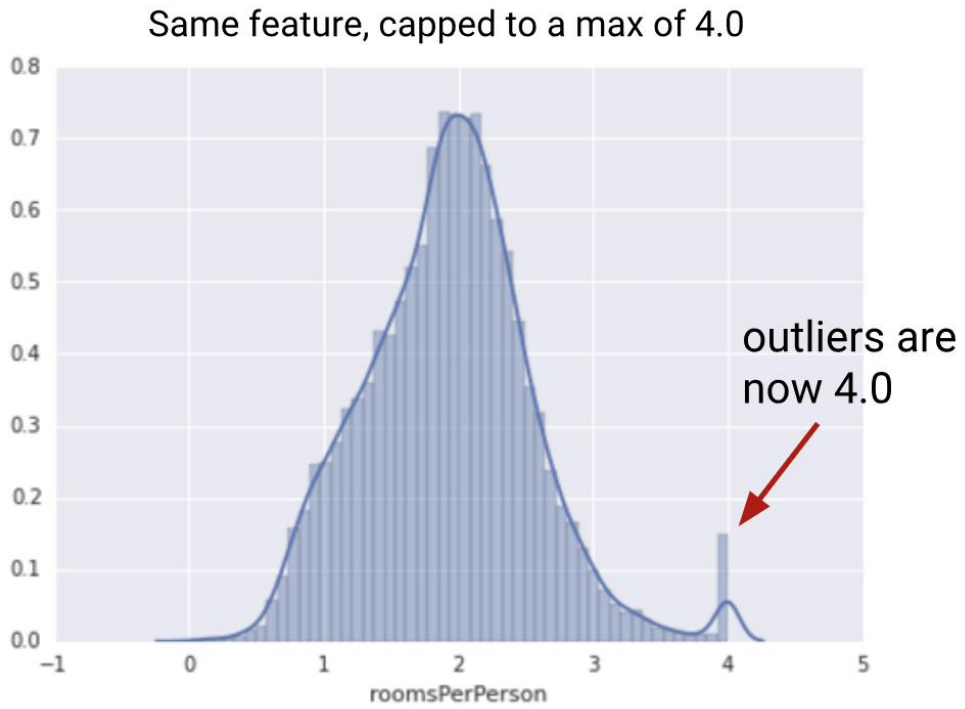
Clip by z-score to **+-Nσ**

(for example, limit to +-3σ).

σ is the standard deviation

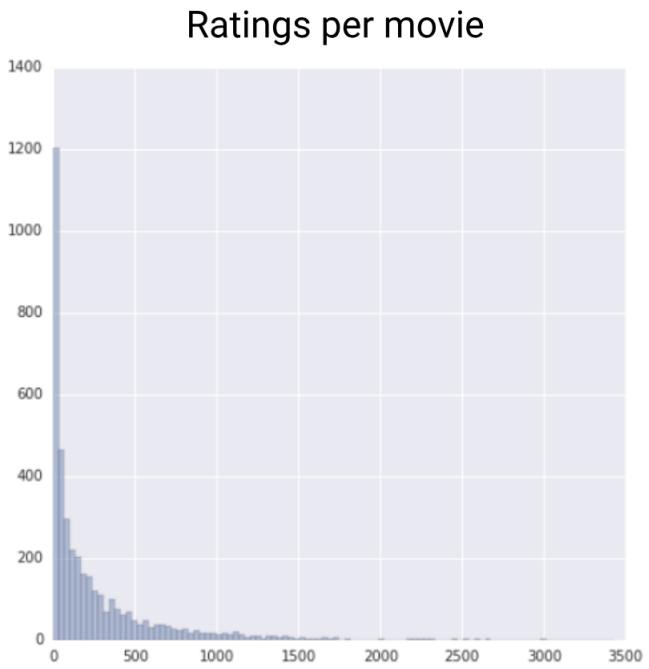
- If your data set contains extreme outliers, capping all feature values above (or below) a certain value to fixed value.

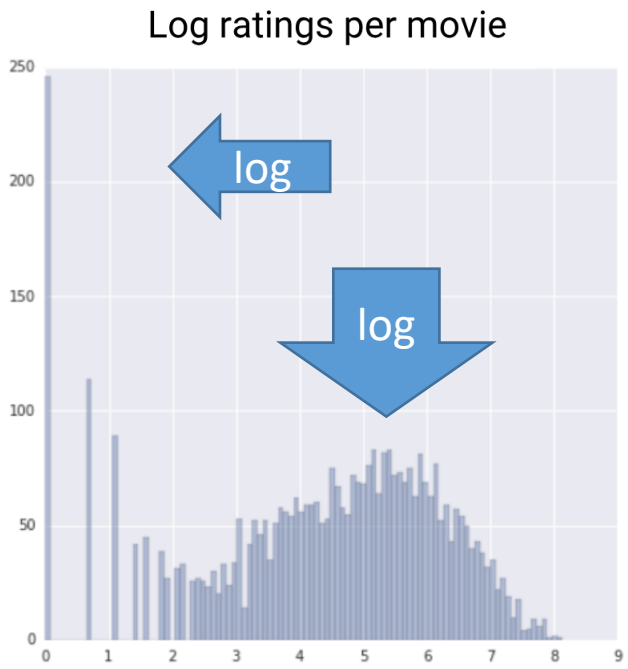




Log Scaling

- When a handful of your values have many points, while most other values have few points.
  - This data distribution is known as the *power law* distribution.



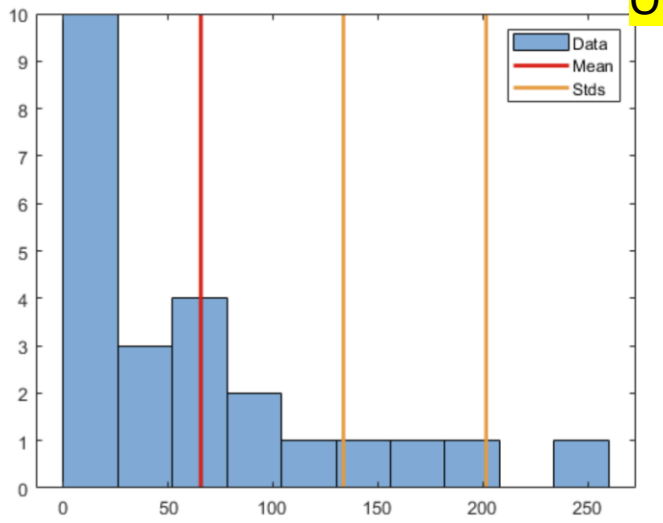


Homework (not in the exam)
Preprocessing Data Example

https://it.mathworks.com/help/matlab/learn_matlab/data-analysis.html#PreprocessingDataExample-4

BASIC STATS

```
h = histogram(c3,10); % Histogram
N = max(h.Values); % Maximum bin count
mu3 = mean(c3); % Data mean
sigma3 = std(c3); % Data standard deviation

hold on
plot([mu3 mu3],[0 N],'r','LineWidth',2) % Mean
X = repmat(mu3+(1:2)*sigma3,2,1);
Y = repmat([0;N],1,2);
plot(X,Y,'Color',[255 153 51]./255,'LineWidth',2) % Standard deviations
legend('Data','Mean','Stds')
hold off
```



DEALING WITH OUTLIERS DATA

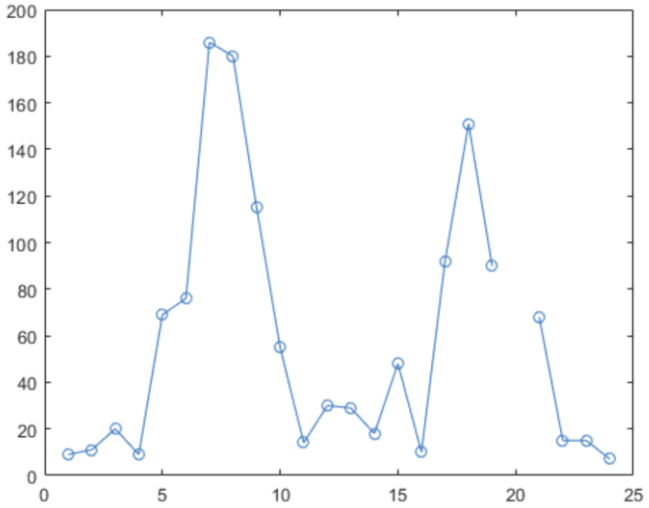
CLIPPING!

```
outliers = (c3 - mu3) > 2*sigma3;
c3m = c3; % Copy c3 to c3m
c3m(outliers) = NaN; % Add NaN values
```

Smoothing and Filtering

A time-series plot of the data at the third intersection (with the outlier removed in Outliers) resu

```
plot(c3m,'o-')
hold on
```



Main points

- Similarity as a tool for deep learning
- Exploratory Data Analysis
- Feature engineering
  - Data preprocessing: a statistical point of view