# DATASCIENCE, LEARNING AND APPLICATIONS

## DALAS - EDA

26 janvier 2024

Laure Soulier - Nicolas Baskiotis

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

MLIA Machine Learning & Deep Learning for Information Access

SCIENCES SORBONNE UNIVERSITÉ

# EDA : DEFINITION

# Definitions

## Exploratory Data Analysis (EDA) (Wikipedia)

"Approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing."

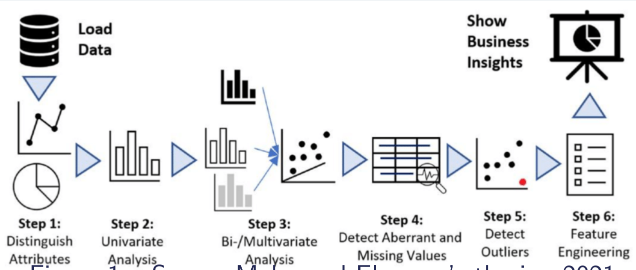



Figure 1 – Source Mahmoud Elansary's thesis - 2021

# EDA vs IDA

## Exploratory Data Analysis vs. Initial Data Anaysis (Wikipedia)

"EDA is different from **initial data analysis (IDA)**, which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA."

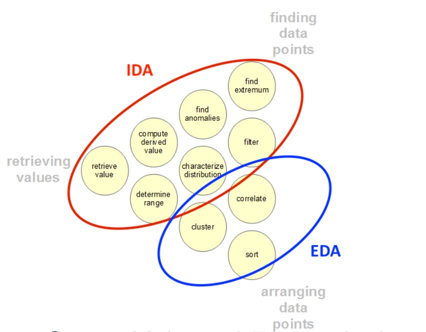



Figure 2 – Source Mahmoud Elansary's thesis - 2021

# Why EDA ? Dirty data : recurrent errors

SCIENCES SORBONNE UNIVERSITÉ

- Data might lack coherency or might be poorly collected/done
- Human process : gathering/building/annotating data without objective, without semantics, without consistency

| name<br>string<br>Text | address<br>string<br>Natural lang. | city<br>string<br>Text | state<br>string<br>US State | zip<br>string<br>Integer | phone<br>string<br>Text | email<br>string<br>E-mail address | work<br>string<br>Natural lang. | work address<br>string<br>Natural lang. |
|---|---|---|---|---|---|---|---|---|
| Cuba Pollich | 312 Ozie Mall | Kingberg | Montana | 23815-0517 | (418)319-1428 | bennie84@hotmail.com | Heaney, Bins and Kautzer | 89544 Thiel Gateway |
| Connie Wisozk | 5598 Marcelle Neck | Marvinstad | Georgia | 82345-8569 | 283-223-4387x37697 | cecilia82@hotmail.com | Carroll, Jakubowski and Stark | 9810 Elenor Lock Apt. 011 |
| Raven Ebert | 5049 Beier Island Apt. 959 | Kilbackshire | West Virginia | 27237-2029 | 1-734-031-7070x39183 | zachariah17@yahoo.com | Witting-Blick | 356 Huels Union |
| Charls Ankunding | 35061 Johnston Ridge | Mannborough | Georgia | 78570-5617 | (741)269-1148 | frami.nya@gmail.com | Bechtelar, Bins and Powlowski | 57015 Walton Island |
| Rahn Dibbert | 1288 Stoltenberg Isle Apt. 857 | Mathildafurt | Wisconsin | 74391-7096 | 970.080.7981x8661 | abbigail.boyer@gmail.com | Cummerata, Reilly and Spencer | 32740 Elyse Turnpike |
| Reed Braun | 70973 Ennis Mountains Apt. 701 | New Jeffie | South Dakota | 99497 | | heller.sid@gmail.com | Hamill-Greenfelder | 051 Connelly Centers Apt. 548 |
| Orra Blick | 1749 Treutel Ways Apt. 400 | Maritachester | West Virginia | 21307 | 1-961-909-2105 | von.petra@hotmail.com | Murphy-Berge | 38763 Iver Junction Suite 576 |
| Lonzo Kulas | 9596 Annika Street | East Jaradview | New Jersey | 50497-2648 | (796)016-5472 | sberge@yahoo.com | Lebsack, Boyer and Trantow | 01332 Cristine Mountains |
| Hester Lehner | 200 Henery Motorway | Bergebury | Kansas | 28601-4961 | 127-873-1321x25586 | leannon.jazlyn@gmail.com | Schroeder Ltd | 354 Pagac Causeway Suite 349 |
| Docia Bogan | 2879 Loy Shoals Apt. 957 | Port Breonnamouth | Maryland | 09798 | 1-775-388-5392 | alonzo96@hotmail.com | Quigley, Schinner and Baumbach | 78230 Lakin Run |
| Bridger Cormier | 289 Katlyn Spurs | Port Karlee | Florida | 98190-1095 | (635)032-8847x7451 | sinda79@yahoo.com | Mohr and Sons | 29420 Ebba Harbor |
| Sylvester Frami | 85015 Lubowitz Lake Suite 158 | South Rainahaven | Connecticut | 31969-8294 | (386)355-4939x668 | fklocko@hotmail.com | Runolfsson LLC | 9554 Kihn Path Suite 818 |
| Celina Jaskolski | 0423 Batz Causeway Suite 226 | Hahnton | Idaho | 56178-0335 | 056.639.0919 | yundt.nikolas@hotmail.com | Rice Group | 930 Bergnaum Parkway |
| Daisey Willms | 2606 Dach Canyon | South Mertieburgh | Oregon | 44021 | 239.986.4731 | esau.greenfelder@yahoo.com | Goyette, Trantow and Gutmann | 615 Cass Street |
| Armstead Schulist | 80460 Kreiger Stravenue | East Sanjuanamouth | Kansas | 00103-0491 | +52(8)6713748523 | ddare@hotmail.com | Schumm-Quigley | 8323 Wiegand Cliff Apt. 050 |
| Taryn Davis | 66335 Powell Drives | Bergnaumbury | Connecticut | 91977 | (175)355-2083x33804 | elle.schuster@hotmail.com | Monahan, Gutkowski and Grady | 9715 McCullough Landing Suite |
| Yessenia McLaughlin | 274 Lassie Village Apt. 684 | Giovannyberg | South Carolina | | 272.624.5239 | wilburn38@hotmail.com | Yost Ltd | 81531 Marylyn Inlet Suite 000 |
| Fran Casper | | East Levern | Delaware | 90912 | (963)749-9395 | dickens.bettyjane@yahoo.com | McDermott-Reynolds | 0450 Baumbach Squares |
| London Hammes | 318 Connelly Fords Apt. 361 | New Franz | Connecticut | 11491-4604 | 07612860813 | mfeeney@gmail.com | Lehner LLC | 95708 Howe Junctions Apt. 521 |

Figure 3 – source https://gallery.dataiku.com/projects/DKU_CLEANING_CONTACTS/datasets/dss_dirty_data_example/explore/

# Why EDA ? Dirty data : recurrent errors

## Importance of data quality

Impact the representativeness of the model

- Everything under the same label (Name → first name, last name, address → street, apartment, ... )
- No consistency in the same label (zip code is different for the same state)
- No format (phone number, date, ...)
- Spelling mistakes
- Missing values
- Duplication

Example of dataset cleaning : https://gallery.dataiku.com/projects/DKU_CLEANING_CONTACTS/recipes/compute_dss_dde_fully_clean/

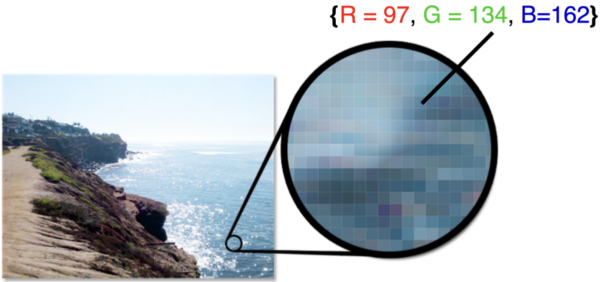
# Types of data

# Facing all data types : Numerical data

- Different types : continuous (interval, ratio, ...), discrete
- Different temporal dimension : single values, sequential values
- Not the same operations, not the same analysis
- Often stored in Data Frame

# Facing all data types : Visual data (images)

- Matrix of pixels → array of pixels
- 1 pixel : 3 dimensions (x,y,(R,G,B))




```
img= imageio.imread("coco.png")
# transformation in 2D, we loose proximity between
    pixels
img_2D = img.reshape(-1,3)
```

# Facing all data types : Audio data

- Wave : continuous signal
- Need to be transformed into a series of discrete values.

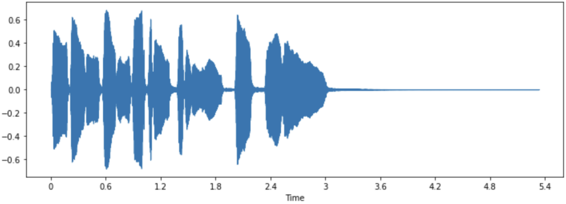



Figure 4 – https://huggingface.co/learn/audio-course/chapter1/audio_data

```python
import matplotlib.pyplot as plt
import librosa.display

plt.figure().set_figwidth(12)
librosa.display.waveshow(array, sr=sampling_rate)
```

# Facing all data types : Audio data

- Sampling rate : the number of samples taken in one second, measured in hertz (Hz)
- The **amplitude** of a sound : the sound pressure level at any given timestamp, measured in decibels (dB)

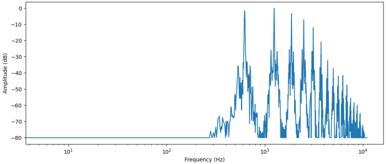



Figure 5 – Frequency spectrum
https://huggingface.co/learn/audio-course/chapter1/audio_data

```
# get the amplitude spectrum in decibels
amplitude_db = librosa.amplitude_to_db(amplitude)
# get the frequency bins
frequency = librosa.fft_frequencies(sr=sampling_rate,
    n_fft=len(dft_input))
```

# Facing all data types : Audio data

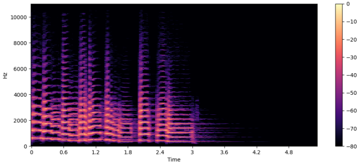



Figure 6 – Spectrogram
https://huggingface.co/learn/audio-course/chapter1/audio_data

```
D = librosa.stft(array)
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
plt.figure().set_figwidth(12)
librosa.display.specshow(S_db, x_axis="time", y_axis="
    hz")
plt.colorbar()
```

# Facing all data types : Textual data

- Different purposes : categorical data or free text
- Free text can by processed to become a vector - or several (of terms, of topics, of sentiment, ....)

| Nom court | Nom complet | Produit | Arrondissement | Localisation |
|---|---|---|---|---|
| BOBILLOT | MARCHÉ BOBILLOT | Alimentaire | 13 | rue Bobillot, cc |
| PORTE DE VANVES | MARCHÉ AUX PUCES PORTE D... | Puces | 14 | Avenue Marc S |
| PORTE MOLITOR | MARCHÉ PORTE MOLITOR | Alimentaire | 16 | sur le trottoir b |
| CONVENTION | MARCHÉ CONVENTION | Alimentaire | 15 | sur les trottoirs |
| ALESIA | MARCHÉ ALESIA | Alimentaire | 13 | rue de la Glaciè |
| LECOURBE | MARCHÉ LECOURBE | Alimentaire | 15 | rue Lecourbe, |
| COURS DE VINCENNES | MARCHÉ COURS DE VINCENNES | Alimentaire | 12 | terre-plein du |
| MAUBERT | MARCHÉ MAUBERT | Alimentaire | 5 | place Maubert |
| BELLEVILLE | MARCHÉ BELLEVILLE | Alimentaire | 11 | terre-pleins du |
| TELEGRAPHE | MARCHÉ TELEGRAPHE | Alimentaire | 20 | sur les trottoirs |
| CARRE MARIGNY | MARCHÉ AUX TIMBRES CARRE ... | Timbres | 8 | Angle des aver |
| BEAUVAU - BROC | MARCHÉ BEAUVAU (Brocante) | Brocante | 12 | Place d'Aligre |

# Facing all data types : Structured data (XML, JSON, ...)

- (Semi-)structured dataset
- Dataset loading easy with Pandas :
  - JSON

```
data_json=pd.read_json("url")
```

  - XML

```
data_xml=request.get("url").content
obj=XML2DataFrame(data_xml)
xml_dataframe=obj.process_data()
```

  - ...

# Dataset structure

# Describing data structure

- Synthesizing information from a dataset using metrics, tables or graphs
- Describing the dataset structure
  - The size, the type of variables

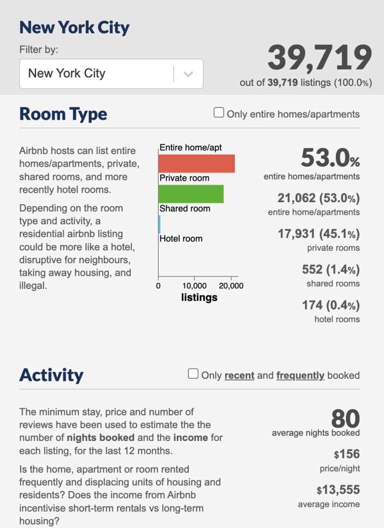
```
data.shape #dimension/size
data.info() #size of dataframe, type of data
    by column, used memory
data.columns #names of columns
data[column_name].dtype #type of column_name
```

  - Do not hesitate to display some lines in the table

```
  data.head()
```

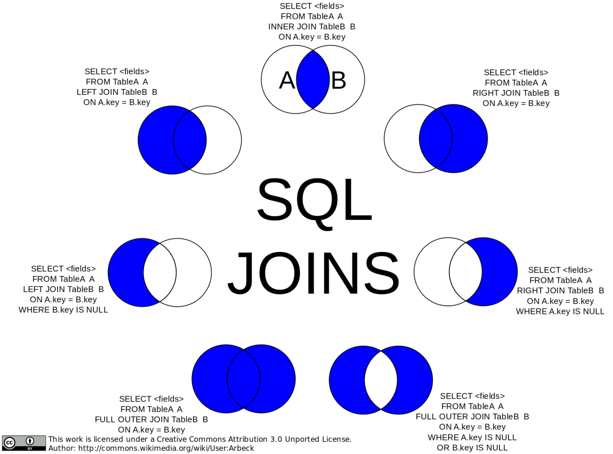
# Transforming data

- By default, Pandas uses three main types :
  - integers **int in 32 or 64 bits,**
  - decimal numbers **float in 32 or 64 bits,**
  - Object **objects**, which include most of the other types
- Transforming the type of data
  - Identifying the reason ("price : $48" is an Object and not a float64)
  - Process the data (remove $)
  - Transforming into the right format : `pd.to_numeric()`



# Fusing/concatenating datasets

- Join : Aligning two datasets according to a join key, a method (left, right, inner, outer)
- Concatenation : without join key.



# Duplicate data

- Detecting same lines in a dataset and removing duplicate data

```
dataset.duplicated().sum() #number of duplicated
    data
dataset.duplicated(['NAME']).sum() #focus on the
    column NAME
dataset.drop_duplicates(['NAME'],keep="first") #
    remove
```

| NAME | TITLE | Number |
|---|---|---|
| Doherty | Officer | 365 |
| Robert | Fire fighter | 457 |
| Robert | Fire Fighter | 127 |
| ... | | |

# Descriptive data analysis and transformation

## Descriptive analysis

SCIENCES SORBONNE UNIVERSITÉ

- Summary of key characteristics of the data distribution
- Different analyses according to the considered variables :
  - Univariate analysis : on a single variable
  - Bivariate analysis : on two variables
  - Multivariate analysis : on many variables
- Different analyses according to the objectives :
  - Central Tendency measures : general center in which the data are distributed
  - Variability measures : "data spread" or how far away the data are from the center.
  - Relative Standing measures : relative position within the dataset.

# Descriptive statistics on quantitative data

- Mean, Variance, standard deviation, median, percentiles, correlation matrix
  - Mean vs. median : depend on the distribution (outlier/bias, ...)

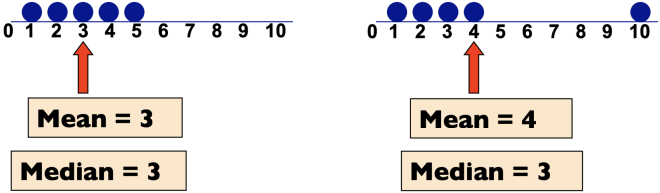



Figure 7 – (c) Jeffrey Heer - University of Washington

- Based on probability distribution : distribution asymetrie (skewness)

```
from scipy.stats import skew
skew (dataset["price"])
```

# Descriptive analysis on qualitative data

- Modalities, frequency, mode, ...

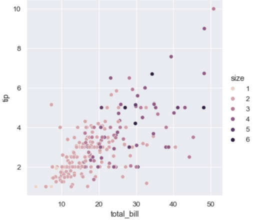
## Categorial type

This type allows to format data as categories/classes instead of considering just textual data. It allows to **improve the reliability of data** (e.g., modalities are set up and a new data should follow this setup, avoids spelling mistakes in textual data), and **lower the memory consumption**.

```
pd.Categorial(["cat1", "cat2", ...])
```

# Visualization

- Library seaborn https://seaborn.pydata.org/
  - Visualizing relationships : Scatter plot (data order important when lines)



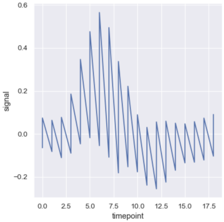
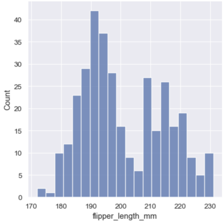




Figure 8 – https://seaborn.pydata.org/tutorial/relational.html

```
#relplot function for graphs with Scatter by default
sns.relplot(data=tips, x="total_bill", y="tip", hue="
    size", palette="ch:r=-.5,l=.75")
#with lines
sns.relplot(data=fmri, kind="line", x="timepoint", y="
    signal",estimator=None)
```

# Visualization

- Visualizing distribution
  - Histograms



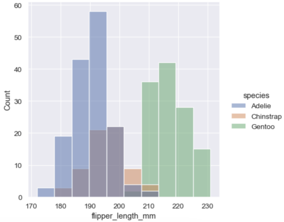




Figure 9 – https://seaborn.pydata.org/tutorial/distributions.html

```
sns.displot(penguins, x="flipper_length_mm", bins=20)
sns.displot(penguins, x="flipper_length_mm", hue="
    species")
```

# Visualization

- Visualizing distribution
  - Kernel density : kernel smooting of probability distribution

$$\hat{f_h}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) \quad (1)$$

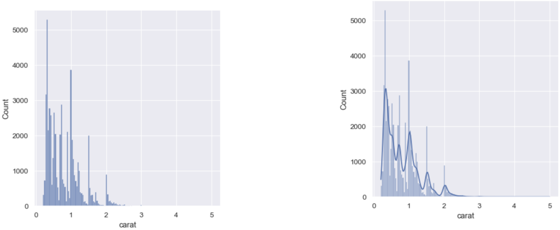



Figure 10 – https://seaborn.pydata.org/tutorial/distributions.html

```
sns.displot(diamonds, x="carat")
sns.displot(diamonds, x="carat", kde=True)
```

# Visualization

SCIENCES SORBONNE UNIVERSITÉ

- Visualizing distribution
  - Cumulative distribution

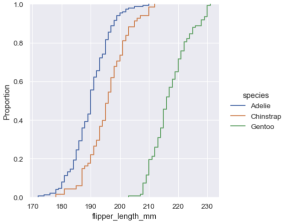



Figure 11 – https://seaborn.pydata.org/tutorial/distributions.html

```
sns.displot(penguins, x="flipper_length_mm", hue="
    species", kind="ecdf")
```

# Visualization

- Visualizing distribution
  - Bivariate distributions

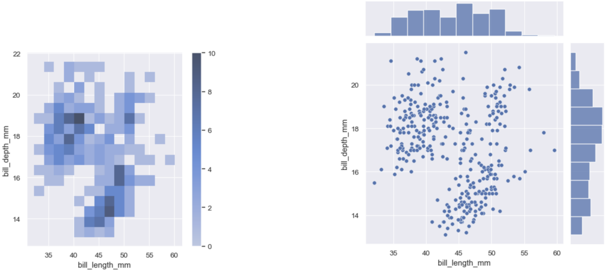



Figure 12 – https://seaborn.pydata.org/tutorial/distributions.html

```
sns.displot(penguins, x="bill_length_mm", y="bill_depth_mm", hue="species")
sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm")
```

# Visualization

- Visualizing distribution
  - Plotting many distribution

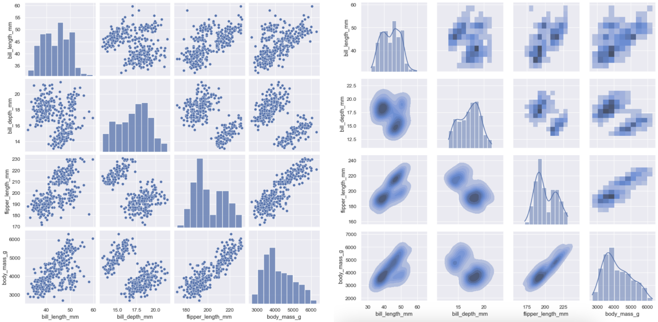



Figure 13 – https://seaborn.pydata.org/tutorial/distributions.html

```
sns.pairplot(penguins)
g = sns.PairGrid(penguins) #more flexible
g.map_upper(sns.histplot)
g.map_lower(sns.kdeplot, fill=True)
g.map_diag(sns.histplot, kde=True)
```

# Visualization

SCIENCES SORBONNE UNIVERSITÉ

- Visualizing categorical data
  - Scatter plot

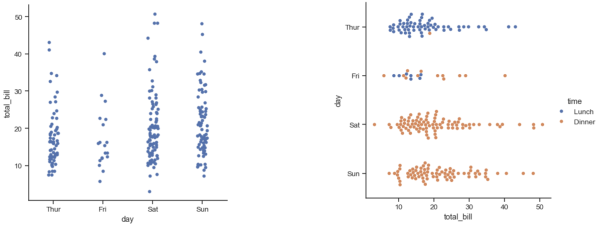



Figure 14 – https://seaborn.pydata.org/tutorial/categorical.html

```
sns.catplot(data=tips, x="day", y="total_bill")
sns.catplot(data=tips, x="total_bill", y="day", hue="
    time", kind="swarm")
```

# Visualization

- Visualizing categorical data
  - Boxplot

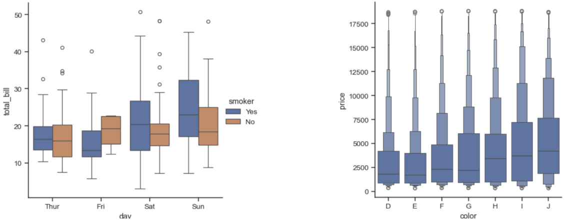



Figure 15 – https://seaborn.pydata.org/tutorial/categorical.html

```
sns.catplot(data=tips, x="day", y="total_bill", hue="smoker", kind="box")
sns.catplot(data=diamonds.sort_values("color"), x="color", y="price", kind="boxen")
```

# Visualization

SCIENCES SORBONNE UNIVERSITÉ

- Visualizing categorical data
  - Violin plot

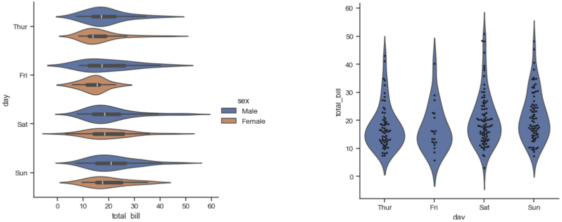



Figure 16 – https://seaborn.pydata.org/tutorial/categorical.html

```
sns.catplot(data=tips, x="total_bill", y="day", hue="
    sex", kind="violin")
#with data distribution
g = sns.catplot(data=tips, x="day", y="total_bill",
    kind="violin", inner=None)
sns.swarmplot(data=tips, x="day", y="total_bill",
    color="k", size=3, ax=g.ax)
```

# Visualization

- Visualizing categorical data
  - Violin plot

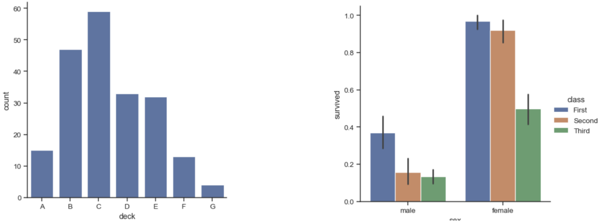
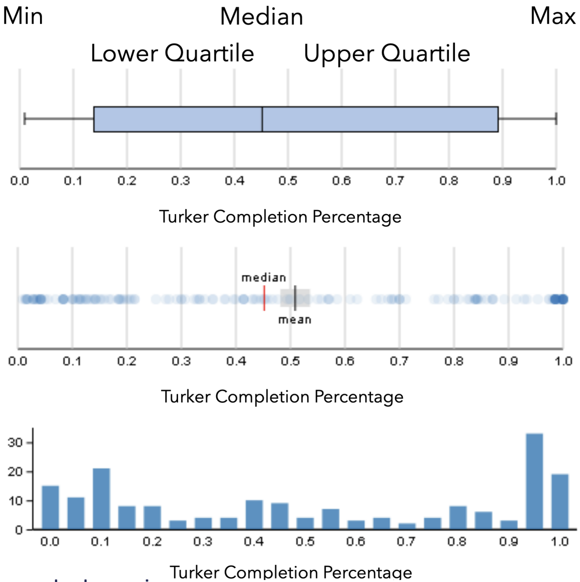



Figure 17 – https://seaborn.pydata.org/tutorial/categorical.html

```python
sns.catplot(data=titanic, x="deck", kind="count")
sns.catplot(data=titanic, x="sex", y="survived", hue="class", kind="bar")
```

# Choose the right plot and the right scale...



# Bivariate analysis

- Correlation (more in 2 weeks)
- Pivot tables : Visualizing/analyzing the intersection of several qualitative variables

```
pd.crosstab(dataset['col1'],dataset['col2'])
```

# Discretization

SCIENCES SORBONNE UNIVERSITÉ

- Transforming quantitative variable into a qualitative one.
- Example : age into classes

```
pd.cut(dataset["age"],bins=3, labels=range(5)) # constant interval
pd.cut(dataset["age"],bins=[dataset["age"].min(), 40, dataset["age"].max()], include_lowest=True) # interval defined by a user
pd.qcut(dataset["age"],q-3) # intervals with uniform frequency
```

# Missing data

- Identifying why ? (capture, transformation, other ?)
- Deleting observations with missing data
  - Reduce the size of the dataset

```
dataset.dropna()
```

- Completing with mean, mode, median for quantitative variables
  - Useful if the missing values occur completely randomly
  - Or in case of rare frequency

```
dataset.fillna(dataset[col].mean())

#autre option avec scikit-learn
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(strategy="mean")
new_dataset=imputer.fit_transform(dataset.
    select_dtypes(np.number))
```

- Add new modality for qualitative variables with `.fillna()`
- More advances methods (multiple data imputation, KNN)

# Missing data : multiple imputation

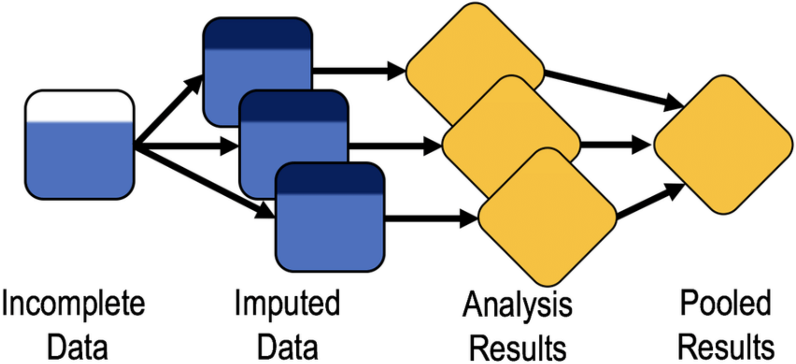
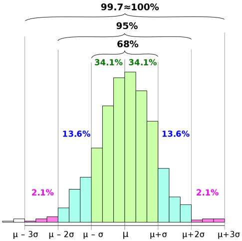



Figure 19 – Nissen et al. 2019

# Outlier detection

- Data that is markedly different from others
- Causes (important to understand why) :
  - Data errors : wrong measurement, wrong annotation, error reporting, ... (1.73 cm for human height, income in billions euros vs. euros)
  - Normal variance in the data : Outside of the 99.7% of the data pointing within three stdev. Those data are legitimate but skew some of the descriptive statistics (e.g., mean).



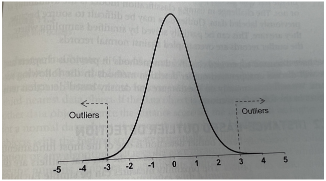

  - Data from other distribution classes : originate from incorrect assumptions (surge in retail after Thanksgiving vs. daily retail)

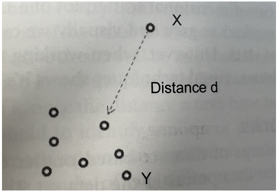
# Outlier detection

- Examples
  - Click fraud in online advertising for free internet services
    - Fraudulent traffic does not follow logical actions
    - It contains repetitive actions
    - Signals : Very high click depth, time between each click, high number of clicks in a session, IP different from the target market, ...
  - Credit card fraud
    - Difficult task : irregular purchase is our regular life
    - The recurrence of irregular purchases is a signal

# Anomaly detection techniques

- Statistical methods
  - Normal distribution with parameters estimated on the dataset (mean, stdev). Outliers are detected according on where they fall in the standard normal distribution
- Data mining methods
  - Distance-based : Average distance of the nearest neighbor, outliers will have a higher value than other points



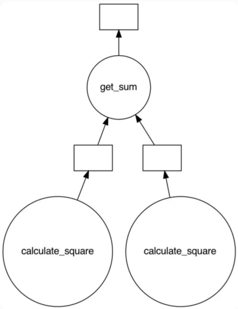

  - Clustering : detection with minimum threshold to belong to clusters
  - Classification techniques : with dedicated label

# Acceleration and parallelization with Numba and Dask

# Parallelization with Dask

- Dask https://domino.ai/blog/dask-step-by-step-tutorial

```
%%time
## Wrapping the function calls using dask.delayed
x = delayed(calculate_square)(10)
y = delayed(calculate_square)(20)
z = delayed(get_sum)(x, y)
## visualize the task graph
z.visualize()
```



## Acceleration with Numba

- Python : interpreted language, not optimized
- Parallelization can be adapted to accelerate the code
- If nor sufficient different alternatives :
  - Changing the language (C, C++, Cython : python with a compiler)
  - Use Numba (https://numba.pydata.org/) : does not require to change the python code

```python
# Python without Numba : 943 ns + 20.8 ns per loop
def hypot_python(x, y) :
  return math.sqrt(x**2 + y**2)

# Numba with decorator @jit : 193 ns + 5.56 ns
def hypot_numba_jit(x, y) :
  return math.sqrt(x**2 + y**2)

# Numba autojit function to transform the Python
    function : 194 ns + 3.56 ns
hypot_numba_autojit = autojit(hypot_python)
```

# Before data science...

# Transforming numerical data

- Standard normalization

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler(with_mean=True,with_std=True)
scaler.fit_transform(dataset)
```

- Change of scale

```
from sklearn.preprocessing import MinMaxScaler
minmaxScaler=MinMaxScaler((0,100))
minmaxScaler.fit_transform(dataset)
```

- Box-Cox transformation : allow to transform data so that it follows Normal law

```
from scipy import stats
stats.boxcox(dataset["earnings"])
```

# Transforming textual data

1-hot encoding image/explication

```
pd.get_dummies(dataset["description"]

#with scikit-learn
from sklearn.preprocessing import OneHotEncoder
encode=OneHotEncoder(sparse=False)
encode.fit_transform(....)
```

## Sampling

- Random sampling without replacement

```
dataset.sample(n=1000)
```

- Stratified sampling

```
dataset.groupby('type').apply(lambda x: x.sample(
    frac=.1))
```

# Roadmap for data exploration

- Organize the dataset : structure the dataset with standard rows and columns
- Find the central point for each attribute (mean, mode, ...)
- Understand the spread of attributes (std, range, ...)
- Visualize the distribution of each attribute
- Detect outliers
- Understand the relationships between attributes