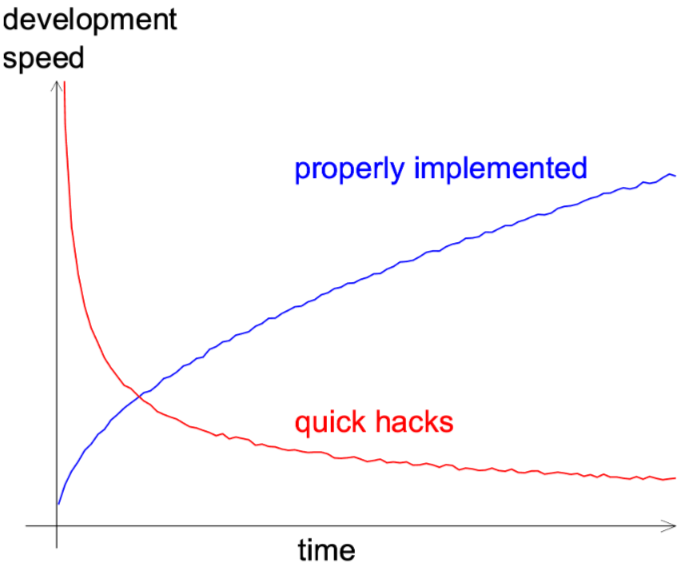
Fundaments of Machine learning for and with engineering applications

Enrico Riccardi[1]

Department of Mathematics and Physics, University of Stavanger (UiS).[1]

Sep 8, 2025

Flexible



Developing approaches

Different code editors are available to interpret python language.

- jupyter notebooks are mostly dedicated to learning (Markdown)
- ipython is for interactive coding (similar to R, Matlab, etc)
- python packages (.py) developing suites (debug possibilities and git integration)

Developing approaches

Different code editors are available to interpret python language.

- jupyter notebooks are mostly dedicated to learning (Markdown)
- ipython is for interactive coding (similar to R, Matlab, etc)
- python packages (.py) developing suites (debug possibilities and git integration)

Developing approaches

Different code editors are available to interpret python language.

- jupyter notebooks are mostly dedicated to learning (Markdown)
- ipython is for interactive coding (similar to R, Matlab, etc)
- python packages (.py) developing suites (debug possibilities and git integration)

Developing approaches

Different code editors are available to interpret python language.

- jupyter notebooks are mostly dedicated to learning (Markdown)
- ipython is for interactive coding (similar to R, Matlab, etc)
- python packages (.py) developing suites (debug possibilities and git integration)

Introducing code standards

When developing code, there are **guidelines** and best practices aimed at improving the quality, readability, and maintainability of a code.

There are different levels of coding quality, mostly depending on the code intended usage (and developer skills).

- Private codes can be whatever (Cpt. Obvious)
- Public packages shall use a 'Golden code standards' such to be used and eventually supported by communities.

Introducing code standards

When developing code, there are **guidelines** and best practices aimed at improving the quality, readability, and maintainability of a code.

There are different levels of coding quality, mostly depending on the code intended usage (and developer skills).

- Private codes can be whatever (Cpt. Obvious)
- Public packages shall use a 'Golden code standards' such to be used and eventually supported by communities.

Community level standards

Principle of 'clean coding':

1. Readability and Clarity: A good code shall be possible to read as when reading a book
2. Structure and object oriented: A code shall be composed by objects, each of them connected in the less redundant way possible.
3. Consistency and Style: Variable naming, function naming and classes naming has to be consistent.
4. Documentation: Each file, each function and each class shall contain the relative description of its aim and its usage
5. Maintainability: Code dependencies have to be stated and consistently defined and updated, such that a suitable environment can be developed at any point in time.

Community level standards

Principle of 'clean coding':

1. Readability and Clarity: A good code shall be possible to read as when reading a book
2. Structure and object oriented: A code shall be composed by objects, each of them connected in the less redundant way possible.
3. Consistency and Style: Variable naming, function naming and classes naming has to be consistent.
4. Documentation: Each file, each function and each class shall contain the relative description of its aim and its usage
5. Maintainability: Code dependencies have to be stated and consistently defined and updated, such that a suitable environment can be developed at any point in time.

Community level standards

Principle of 'clean coding':

1. Readability and Clarity: A good code shall be possible to read as when reading a book
2. Structure and object oriented: A code shall be composed by objects, each of them connected in the less redundant way possible.
3. Consistency and Style: Variable naming, function naming and classes naming has to be consistent.
4. Documentation: Each file, each function and each class shall contain the relative description of its aim and its usage
5. Maintainability: Code dependencies have to be stated and consistently defined and updated, such that a suitable environment can be developed at any point in time.

Community level standards

Principle of 'clean coding':

1. Readability and Clarity: A good code shall be possible to read as when reading a book
2. Structure and object oriented: A code shall be composed by objects, each of them connected in the less redundant way possible.
3. Consistency and Style: Variable naming, function naming and classes naming has to be consistent.
4. Documentation: Each file, each function and each class shall contain the relative description of its aim and its usage
5. Maintainability: Code dependencies have to be stated and consistently defined and updated, such that a suitable environment can be developed at any point in time.

Community level standards

Principle of 'clean coding':

1. Readability and Clarity: A good code shall be possible to read as when reading a book
2. Structure and object oriented: A code shall be composed by objects, each of them connected in the less redundant way possible.
3. Consistency and Style: Variable naming, function naming and classes naming has to be consistent.
4. Documentation: Each file, each function and each class shall contain the relative description of its aim and its usage
5. Maintainability: Code dependencies have to be stated and consistently defined and updated, such that a suitable environment can be developed at any point in time.

Community level standards

Principle of 'clean coding':

1. Readability and Clarity: A good code shall be possible to read as when reading a book
2. Structure and object oriented: A code shall be composed by objects, each of them connected in the less redundant way possible.
3. Consistency and Style: Variable naming, function naming and classes naming has to be consistent.
4. Documentation: Each file, each function and each class shall contain the relative description of its aim and its usage
5. Maintainability: Code dependencies have to be stated and consistently defined and updated, such that a suitable environment can be developed at any point in time.

Community level standard

1. Testing: Unit testing shall cover the majority of the code
2. Error Handling: Each error shall be captured and properly identified.
3. Examples and benchmarks: Users shall be able to execute minimal examples of the code for computational checks.
4. Performance Optimization: Libraries shall be able to use the available computational power in the machine (e.g. GPU-CUDA)

Community level standard

1. Testing: Unit testing shall cover the majority of the code
2. Error Handling: Each error shall be captured and properly identified.
3. Examples and benchmarks: Users shall be able to execute minimal examples of the code for computational checks.
4. Performance Optimization: Libraries shall be able to use the available computational power in the machine (e.g. GPU-CUDA)

Community level standard

1. Testing: Unit testing shall cover the majority of the code
2. Error Handling: Each error shall be captured and properly identified.
3. Examples and benchmarks: Users shall be able to execute minimal examples of the code for computational checks.
4. Performance Optimization: Libraries shall be able to use the available computational power in the machine (e.g. GPU-CUDA)

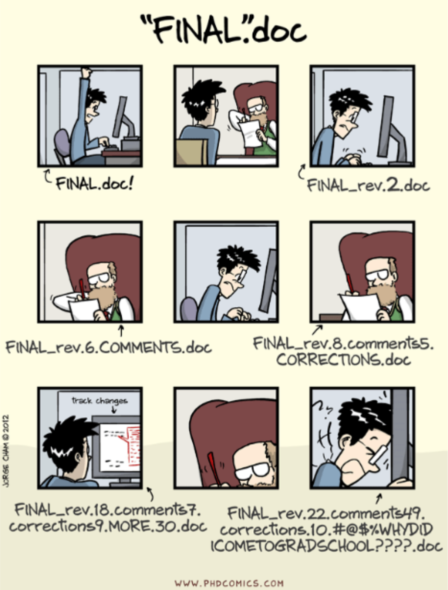
Community level standard

1. Testing: Unit testing shall cover the majority of the code
2. Error Handling: Each error shall be captured and properly identified.
3. Examples and benchmarks: Users shall be able to execute minimal examples of the code for computational checks.
4. Performance Optimization: Libraries shall be able to use the available computational power in the machine (e.g. GPU-CUDA)

Community level standard

1. Testing: Unit testing shall cover the majority of the code
2. Error Handling: Each error shall be captured and properly identified.
3. Examples and benchmarks: Users shall be able to execute minimal examples of the code for computational checks.
4. Performance Optimization: Libraries shall be able to use the available computational power in the machine (e.g. GPU-CUDA)

Version control



Git

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different computers). [Wiki]

Let's try to be more accessible.

Git is a computer program/tool to save and download files on a hosting server (e.g. GitHub and GitLab).

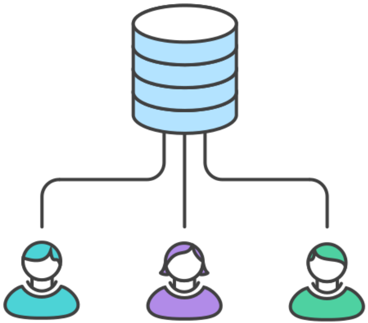
Git

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different computers). [Wiki]

Let's try to be more accessible.

Git is a computer program/tool to save and download files on a hosting server (e.g. GitHub and GitLab).

Centralized workflow



A distributed version control system

GIT

- Git facilitates users to track the various versions of files. It is not a necessary tool, but it can be very very helpful. Generally, the time spent to learn its syntax is well paid off

(do you remember to save some file like *manuscript_draft_v4.02_final_definitive_forreal_lastcomments_editedbyER_submittedVe*
Exactly! Imagine to do that for a repository of files...)

- It permits to save and share the intermediate stages of a work in progress (which software is complete and always up to date?) in an accessible, consistent and structured way, allowing an effective version tracking. It allows retrieval of previous working versions, limiting the risk to overwrite useful files.

A distributed version control system

GIT

- Git facilitates users to track the various versions of files. It is not a necessary tool, but it can be very very helpful. Generally, the time spent to learn its syntax is well paid off

(do you remember to save some file like
manuscript_draft_v4.02_final_definitive_forreal_lastcomments_editedbyER_submittedVe
Exactly! Imagine to do that for a repository of files...)

- It permits to save and share the intermediate stages of a work in progress (which software is complete and always up to date?) in an accessible, consistent and structured way, allowing an effective version tracking. It allows retrieval of previous working versions, limiting the risk to overwrite useful files.

A distributed version control system

GIT

- Git facilitates users to track the various versions of files. It is not a necessary tool, but it can be very very helpful. Generally, the time spent to learn its syntax is well paid off

(do you remember to save some file like
manuscript_draft_v4.02_final_definitive_forreal_lastcomments_editedbyER_submittedVe
Exactly! Imagine to do that for a repository of files...)

- It permits to save and share the intermediate stages of a work in progress (which software is complete and always up to date?) in an accessible, consistent and structured way, allowing an effective version tracking. It allows retrieval of previous working versions, limiting the risk to overwrite useful files.

What is git actually for

The tool is particularly useful for programmers working in teams or in projects whose outcomes can be used by others.

- Git helps to co-develop a code, test its functions and the compatibility of the various code sections.
- A long list of further possibilities became possible by git.
- Different software integration on development platforms, based on git, will help you to develop and co-develop your code.
- The platform GitLab and GitHub have a large set of functionalities to further support code documentation and public releases.
- Files can be disclosed to the public, becoming a great integration of your CV, showing what you are able to do in an open and accessible way.

What is git actually for

The tool is particularly useful for programmers working in teams or in projects whose outcomes can be used by others.

- Git helps to co-develop a code, test its functions and the compatibility of the various code sections.
- A long list of further possibilities became possible by git.
- Different software integration on development platforms, based on git, will help you to develop and co-develop your code.
- The platform GitLab and GitHub have a large set of functionalities to further support code documentation and public releases.
- Files can be disclosed to the public, becoming a great integration of your CV, showing what you are able to do in an open and accessible way.

What is git actually for

The tool is particularly useful for programmers working in teams or in projects whose outcomes can be used by others.

- Git helps to co-develop a code, test its functions and the compatibility of the various code sections.
- A long list of further possibilities became possible by git.
- Different software integration on development platforms, based on git, will help you to develop and co-develop your code.
- The platform GitLab and GitHub have a large set of functionalities to further support code documentation and public releases.
- Files can be disclosed to the public, becoming a great integration of your CV, showing what you are able to do in an open and accessible way.

What is git actually for

The tool is particularly useful for programmers working in teams or in projects whose outcomes can be used by others.

- Git helps to co-develop a code, test its functions and the compatibility of the various code sections.
- A long list of further possibilities became possible by git.
- Different software integration on development platforms, based on git, will help you to develop and co-develop your code.
- The platform GitLab and GitHub have a large set of functionalities to further support code documentation and public releases.
- Files can be disclosed to the public, becoming a great integration of your CV, showing what you are able to do in an open and accessible way.

What is git actually for

The tool is particularly useful for programmers working in teams or in projects whose outcomes can be used by others.

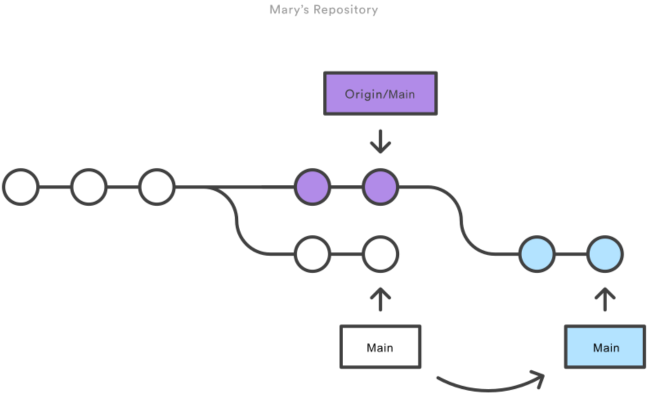
- Git helps to co-develop a code, test its functions and the compatibility of the various code sections.
- A long list of further possibilities became possible by git.
- Different software integration on development platforms, based on git, will help you to develop and co-develop your code.
- The platform GitLab and GitHub have a large set of functionalities to further support code documentation and public releases.
- Files can be disclosed to the public, becoming a great integration of your CV, showing what you are able to do in an open and accessible way.

What is git actually for

The tool is particularly useful for programmers working in teams or in projects whose outcomes can be used by others.

- Git helps to co-develop a code, test its functions and the compatibility of the various code sections.
- A long list of further possibilities became possible by git.
- Different software integration on development platforms, based on git, will help you to develop and co-develop your code.
- The platform GitLab and GitHub have a large set of functionalities to further support code documentation and public releases.
- Files can be disclosed to the public, becoming a great integration of your CV, showing what you are able to do in an open and accessible way.

How does it work -in short-



Why should I care?

As the open libraries are exploding in numbers, you might need some criteria to assert the reliability of a project.

Unit test driven development!

That is taking full advantage of python object oriented structure.

Community

Good project are not only used by communities, but also **supported**

Git allows the development of projects without a clear lead. Community engagement is generally a desirable target to help develop to directly integrate feedbacks by users (and fix bugs).

Why should I care?

As the open libraries are exploding in numbers, you might need some criteria to assert the reliability of a project.

Unit test driven development!

That is taking full advantage of python object oriented structure.

Community

Good project are not only used by communities, but also **supported**

Git allows the development of projects without a clear lead. Community engagement is generally a desirable target to help develop to directly integrate feedbacks by users (and fix bugs).

Statistics (recaps)

Definition

Statistics is the science of acquiring and utilizing data

- It comprises tools for data collection, summarization, and interpretation.
- The aim is identifying the underlying structure, trends, and relationships inherent in the data.
- Is it all statistics then? Yes.
- **Numbers to data, data to information**

Statistics (recaps)

Definition

Statistics is the science of acquiring and utilizing data

- It comprises tools for data collection, summarization, and interpretation.
- The aim is identifying the underlying structure, trends, and relationships inherent in the data.
- Is it all statistics then? Yes.
- **Numbers to data, data to information**

Statistics (recaps)

Definition

Statistics is the science of acquiring and utilizing data

- It comprises tools for data collection, summarization, and interpretation.
- The aim is identifying the underlying structure, trends, and relationships inherent in the data.
- Is it all statistics then? Yes.
- **Numbers to data, data to information**

Statistics (recaps)

Definition

Statistics is the science of acquiring and utilizing data

- It comprises tools for data collection, summarization, and interpretation.
- The aim is identifying the underlying structure, trends, and relationships inherent in the data.
- Is it all statistics then? Yes.
- **Numbers to data, data to information**

Statistics (recaps)

Definition

Statistics is the science of acquiring and utilizing data

- It comprises tools for data collection, summarization, and interpretation.
- The aim is identifying the underlying structure, trends, and relationships inherent in the data.
- Is it all statistics then? Yes.
- **Numbers to data, data to information**

Data properties

Before we talk about machine learning, we need to refresh some terminology.

Population

The universe of all possible outcomes and events.

Sample

A finite subset extracted from the population.

Exhaustivity

The samples covered the population spectra.

Representativity

The population is properly described by the samples.

Data properties

Before we talk about machine learning, we need to refresh some terminology.

Population

The universe of all possible outcomes and events.

Sample

A finite subset extracted from the population.

Exhaustivity

The samples covered the population spectra.

Representativity

The population is properly described by the samples.

Big data

We speak of big data when dataset are very large: i.e. many instances and features Models have thus a large set of parameters (and often no one has a clue anymore of what is going on).

- Volume of data
- Variety different types of data sources with different length and scale.
- Frequency of data generation

Big data

We speak of big data when dataset are very large: i.e. many instances and features Models have thus a large set of parameters (and often no one has a clue anymore of what is going on).

- Volume of data
- Variety different types of data sources with different length and scale.
- Frequency of data generation

Big data

We speak of big data when dataset are very large: i.e. many instances and features Models have thus a large set of parameters (and often no one has a clue anymore of what is going on).

- Volume of data
- Variety different types of data sources with different length and scale.
- Frequency of data generation

Big data

We speak of big data when dataset are very large: i.e. many instances and features Models have thus a large set of parameters (and often no one has a clue anymore of what is going on).

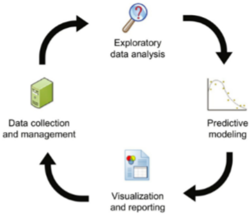
- Volume of data
- Variety different types of data sources with different length and scale.
- Frequency of data generation

Sampling

Samples shall have no bias (to be randomly selected). If not, the bias has to be corrected for.

Cycle of data

1. Data is collected
2. Checked upon
3. Some modelling
4. Analysis and visualization

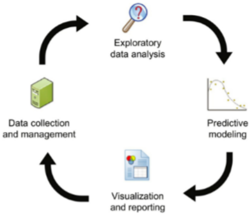


Sampling

Samples shall have no bias (to be randomly selected). If not, the bias has to be corrected for.

Cycle of data

1. Data is collected
2. Checked upon
3. Some modelling
4. Analysis and visualization



Data quality

1. Data has to be acquired and integrated
2. Data are passed to a quality analysis and control
3. Data cleaning, consistency check. Most of time goes here

Preliminary Modeling

Main tasks:

1. Hunt for redundancy
2. Reduce dimensionality
3. AnOmAlIes removal

- Descriptive modeling (unsupervised learning)
- Predictive modeling (supervised learning)
- The model can be used to guide data acquisition (risky!)

Visualization and reporting

- The data has to be condensed into a visualization to provide input for decisions.
- Depending on the goal, very very different visualizations are possible.
- Use a model to indicate what is undersampled or oversampled.

Summarizing and visualizing data as a starting point for more analysis later on.

- Computing summary statistics (e.g. means and variance)
- Determining conditional probabilities of cause+effect relationships
- Calculating correlation and rank correlation coefficient between two variables
- Visualizing univariate, bivariate and multivariate data
- ...

Exploratory data analysis

Summarizing and visualizing data as a starting point for more analysis later on.

- ...
- Estimating probability coverage levels for different distributions
- Analyzing behaviour of normal distributions
- Calculating confidence interval and sampling distribution for the mean
- Testing for significance of difference in means
- Comparing two different distributions for statistical equivalence
- Developing a nonparametric regression model from given data
- Reducing data dimensionality
- Grouping data

Random Variables

- A random variable is a real valued function that assigns a value to each outcome in the sample space
- A random variable (RV) can be either discrete or continuous
  - Discrete RV
  - Continuous RV

- The probability mass function (PMF),P, of a discrete RV, X, denotes the probability that the RV is equal to a specified value, a.

p(a) = p(X = a)

- The cumulative distribution function (CDF), F, denotes the sum

$F(a) = P(X \leq a) = \sum_0^a f(x)dx$

Random Variables

- A random variable is a real valued function that assigns a value to each outcome in the sample space
- A random variable (RV) can be either discrete or continuous
    - Discrete RV
    - Continuous RV

- The probability mass function (PMF),P, of a discrete RV, X, denotes the probability that the RV is equal to a specified value, a.

$$p(a) = p(X = a)$$

- The cumulative distribution function (CDF), F, denotes the sum

$$F(a) = P(X \leq a) = \sum_0^a f(x)dx$$

Random Variables

- A random variable is a real valued function that assigns a value to each outcome in the sample space
- A random variable (RV) can be either discrete or continuous
  - Discrete RV
  - Continuous RV

- The probability mass function (PMF),P, of a discrete RV, X, denotes the probability that the RV is equal to a specified value, a.

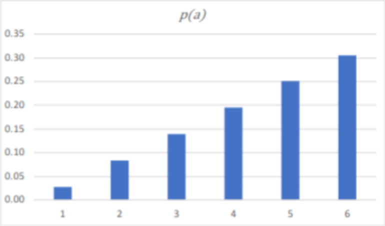
p(a) = p(X = a)

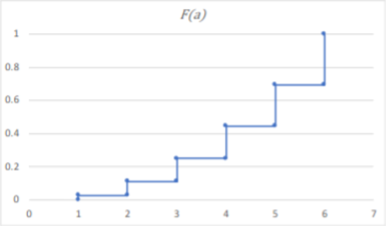
- The cumulative distribution function (CDF), F, denotes the sum

$F(a) = P(X \leq a) = \sum_0^a f(x)dx$

Random Variables

a	1	2	3	4	5	6
$p(a)$	1/36	3/36	5/36	7/36	9/36	11/36
$F(a)$	1/36	4/36	9/36	16/36	25/36	1





Sampling

- What are the effective sampling strategies? (Wind turbine example)
- Solar Panels to determine the efficiency of the source (Usage patterns, energy production forecast)
- Drilling (penetration rate)
- Corrosion extension
- Concrete Rigidity
- Experimental design!

Wind turbine example

Turbine	Height	X	Y	Wind Speed	Air Density	Temperature	Power Output	Rotor Diameter	Hub Height	Air Pressure	Turbulence Intensity
WT-1	80	752.1	3945	7.5	1.225	15	1500	82	80	1013	0.1
WT-1	80	752.2	3945	8	1.223	15	1600	82	80	1012	0.12
WT-1	80	752.3	3945	7.8	1.224	16	1550	82	80	1013	0.11
WT-2	90	753.5	3946	6.5	1.226	14	1400	85	90	1012	0.15
WT-2	90	753.6	3946	7	1.225	14	1500	85	90	1011	0.13
WT-2	90	753.7	3946	7.2	1.227	14	1520	85	90	1012	0.14

Sampling approaches

Experimental design

Grid, parallel, series.

Sampling without replacement

SPR (single point representation).

Sampling with replacement

The number of the members of the population does not change.

Sampling approaches

Experimental design

Grid, parallel, series.

Sampling without replacement

SPR (single point representation).

Sampling with replacement

The number of the members of the population does not change.

Sampling approaches

Experimental design

Grid, parallel, series.

Sampling without replacement

SPR (single point representation).

Sampling with replacement

The number of the members of the population does not change.

Univariate statistics

- Easy to displaying data:
  - histogram
  - frequency plots
  - cumulative
- Measures of Location
  - Mean, median, mode
  - Quartiles, Percentiles, Quantiles
- Measure of Dispersion (Spread)
  - Standard deviation (sd)
  - Variance (Var) or coefficient of variation
- Measures of shape
  - Skewness, modality

Univariate statistics

- Easy to displaying data:
    - histogram
    - frequency plots
    - cumulative
- Measures of Location
    - Mean, median, mode
    - Quartiles, Percentiles, Quantiles
- Measure of Dispersion (Spread)
    - Standard deviation (sd)
    - Variance (Var) or coefficient of variation
- Measures of shape
    - Skewness, modality

Univariate statistics

- Easy to displaying data:
  - histogram
  - frequency plots
  - cumulative
- Measures of Location
  - Mean, median, mode
  - Quartiles, Percentiles, Quantiles
- Measure of Dispersion (Spread)
  - Standard deviation (sd)
  - Variance (Var) or coefficient of variation
- Measures of shape
  - Skewness, modality

Univariate statistics

- Easy to displaying data:
  - histogram
  - frequency plots
  - cumulative
- Measures of Location
  - Mean, median, mode
  - Quartiles, Percentiles, Quantiles
- Measure of Dispersion (Spread)
  - Standard deviation (sd)
  - Variance (Var) or coefficient of variation
- Measures of shape
  - Skewness, modality

Task (due 15.9.2025)

- Task 1: make a histogram from a 2d random distribution
- Task 2: make a 2d heat map from a 2d random distribution
- Task 3: make a histogram for the source data you selected
- Task 4: convert the histogram into a discrete PMF
- Task 5: calculate the comulative for each feature

Frequency plots and Histograms

Given a set of data

1. Look for min and max values
2. Divide the range of values into a number of sensible class intervals (bins)
3. Count
4. Make a frequency table (or percentage)
5. Plot (see jupyter notebook)

Does this histogram represent uncertainty?

No. It shows variability, but it can be used to quantify uncertainty.

Frequency plots and Histograms

Given a set of data

1. Look for min and max values
2. Divide the range of values into a number of sensible class intervals (bins)
3. Count
4. Make a frequency table (or percentage)
5. Plot (see jupyter notebook)

Does this histogram represent uncertainty?

No. It shows variability, but it can be used to quantify uncertainty.

Class widths

- Class widths (bin sizes) are usually CONSTANT
  - the height of each bar is proportional to the number of values in it
- If class width are VARIABLE
  - the AREA of each bar is proportional to the number of values in it
- For small samples, the shape of the histogram can be very sensitive to the number and definition of the class intervals

Class widths

- Class widths (bin sizes) are usually CONSTANT
  - the height of each bar is proportional to the number of values in it
- If class width are VARIABLE
  - the AREA of each bar is proportional to the number of values in it
- For small samples, the shape of the histogram can be very sensitive to the number and definition of the class intervals

Cumulative Histogram

- Cumulative frequency
- Each data point can be plotted individually
- It helps to read quantiles and compare distributions

Measure of Location: Central Tendency, MEAN

$$m_x = < x > = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Each point weighted equally by $\frac{1}{n}$ (assumption)

- Every element is the data set contributes to the values of the mean
- An average provides a common measure for comparing one set of data to another
- The mean is influenced by the extreme values in the data set
- The mean may not be an actual element if the dataset
- The sum of all deviation from the mean is zero, and the sum of squared deviation is minimized when those deviations are measured from the mean

Measure of Location: Central Tendency, MEAN

$$m_x = < x > = \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Each point weighted equally by $\frac{1}{n}$ (assumption)

- Every element is the data set contributes to the values of the mean
- An average provides a common measure for comparing one set of data to another
- The mean is influenced by the extreme values in the data set
- The mean may not be an actual element if the dataset
- The sum of all deviation from the mean is zero, and the sum of squared deviation is minimized when those deviations are measured from the mean

Measure of Location: Central Tendency, MEAN

$$m_x = < x > = \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Each point weighted equally by $\frac{1}{n}$ (assumption)

- Every element is the data set contributes to the values of the mean
- An average provides a common measure for comparing one set of data to another
- The mean is influenced by the extreme values in the data set
- The mean may not be an actual element if the dataset
- The sum of all deviation from the mean is zero, and the sum of squared deviation is minimized when those deviations are measured from the mean

Means

- Arithmetic
  - Mean of raw data

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

- Geometric
  - n^{th} root of product

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

- Geometric

  * Mean of logarithms $exp\left(\frac{1}{n}\sum_{i=1}^{n} ln(x_i)\right)$

- Harmonic
  - Mean of inverses

$$\left(\frac{1}{n}\sum_{i=1}^{n} \frac{1}{x_i}\right)^{-1}$$

Means

- Arithmetic
  - Mean of raw data

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

- Geometric
  - n^{th} root of product

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

- Geometric

  * Mean of logarithms $exp\left(\frac{1}{n}\sum_{i=1}^{n} ln(x_i)\right)$

- Harmonic
  - Mean of inverses

$$\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i}\right)^{-1}$$

Means

- Arithmetic
  - Mean of raw data

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

- Geometric
  - n^{th} root of product

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

- Geometric

* Mean of logarithms $exp\left(\frac{1}{n}\sum_{i=1}^{n} ln(x_i)\right)$

- Harmonic
  - Mean of inverses

$$\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i}\right)^{-1}$$

Median

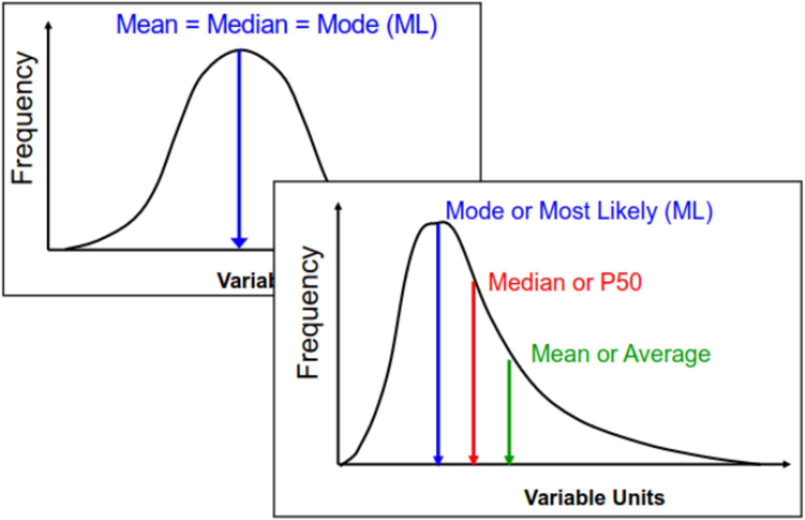
```
if n is odd:
    median = x[(n+1)/2]
else:
    median = x[n/2] + x[(n/2)+1]
```

- On a cumulative density plot, the value of the x-axis that corresponds to 50 % of the y-axis
- Not influenced by extreme values
- May not be contained in the dataset (if n is even)
- For a perfectly symmetrical dataset, means = median

Mode

- The mode is the most frequently occurring data element or the most likely or most probable value (for a pmf)
- A data set may have more than one mode and it thus called multimodal
- A mode is always a data element in the set
- For a perfectly symmetrical dataset, means = median = mode

Distribution Descriptors



Quantiles

Quartiles

The data split into quarters.

Deciles

The data are split into tenths. The fifth decile is also the median.

Percentiles

The data are split into hundredths. P10, P25, P50, P75 and P90 are the most commonly used.

Quantiles

A generalization of splitting data into any fraction

Quantiles

Quartiles

The data split into quarters.

Deciles

The data are split into tenths. The fifth decile is also the median.

Percentiles

The data are split into hundredths. P10, P25, P50, P75 and P90 are the most commonly used.

Quantiles

A generalization of splitting data into any fraction

Quantiles

Quartiles

The data split into quarters.

Deciles

The data are split into tenths. The fifth decile is also the median.

Percentiles

The data are split into hundredths. P10, P25, P50, P75 and P90 are the most commonly used.

Quantiles

A generalization of splitting data into any fraction

Quantiles

Quartiles

The data split into quarters.

Deciles

The data are split into tenths. The fifth decile is also the median.

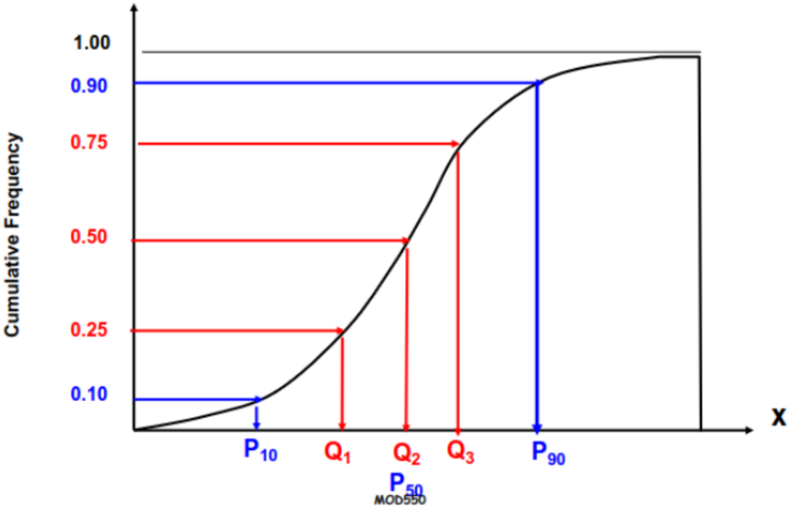
Percentiles

The data are split into hundredths. P10, P25, P50, P75 and P90 are the most commonly used.

Quantiles

A generalization of splitting data into any fraction

Distribution Descriptors



Dispersion (Spread)

Range

R = maximum - minimum

Inter-quantile Range

IQR = Q3 - Q1

Mean Deviation from the Mean

$$\text{MD} = \sum_{i=1}^{n}(x_i - \bar{x})/n$$

Mean Absolute Deviation

$$\text{MAD} = \sum_{i=1}^{n}|x_i - \bar{x}|/n$$

Variance

The variance is the average of squared differences between the sample data points and their mean

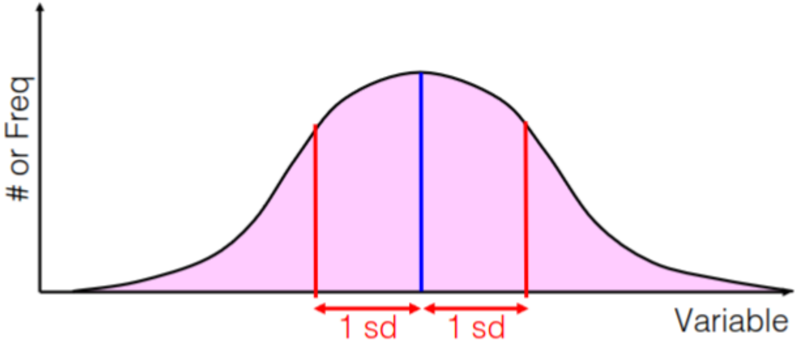
Variance

$$s_x^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

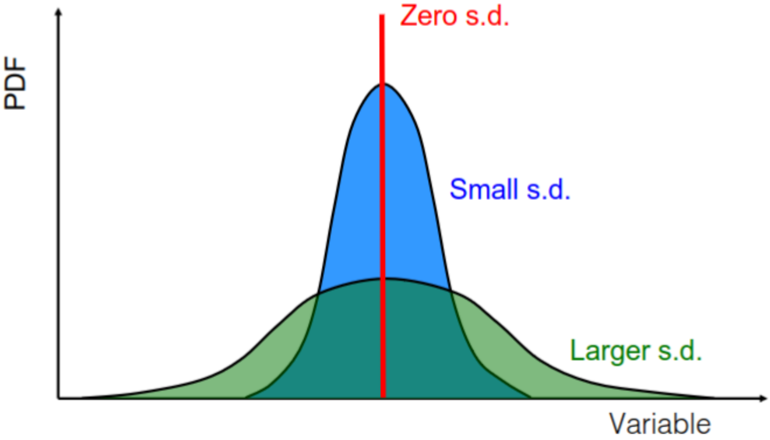
Standard Deviation (SD)

$$s_x = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Standard Deviation



Standard Deviation



Measures of dispersion

Standard Deviation (SD)

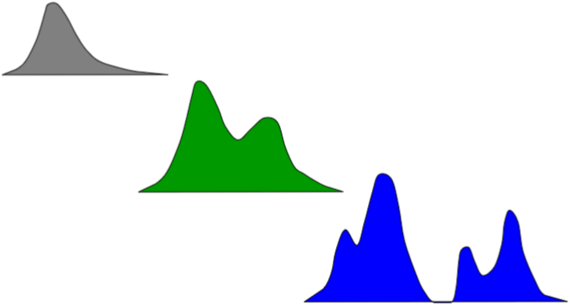
$$SE_x = \frac{\boldsymbol{s_x}}{\sqrt{\boldsymbol{n}}}$$

Coefficient of Variability

$$CV = \frac{\boldsymbol{s_x}}{\bar{x}}$$

Modality

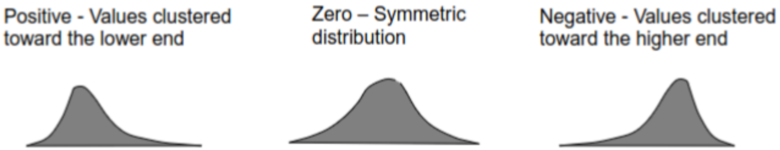
- Unimodal
- Bimodal
- Polymodal



Skewness

It measures the symmetry in a distribution

$$Sk = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{s^3}$$



A bit out of fashion with ML

Distribution Models

Distribution

means of expressing uncertainty or variability

Models

- Uniform: useful when only upper and lower bounds are known
- Triangular: useful when estimates of min, max, mode [P10, P50, P90] are available
- Normal: symmetric model of random errors or unbiased uncertainties with mean of standard deviation specified
  - very common for observed data
  - additive processes tend to be normal as a result of the Central Limit Theorem
- log normal comes from multiplicative uncertainties with mean and standard deviation specified

Uniform Distribution

- The uniform distribution is useful as a rough model for representing low states of knowledge when only the upper and lower bounds are known.
- All possible values within the specified maximum and minimum values are equally likely (b=max, a=min):
- It can express maximum uncertainty

PDF: $f(x) =$

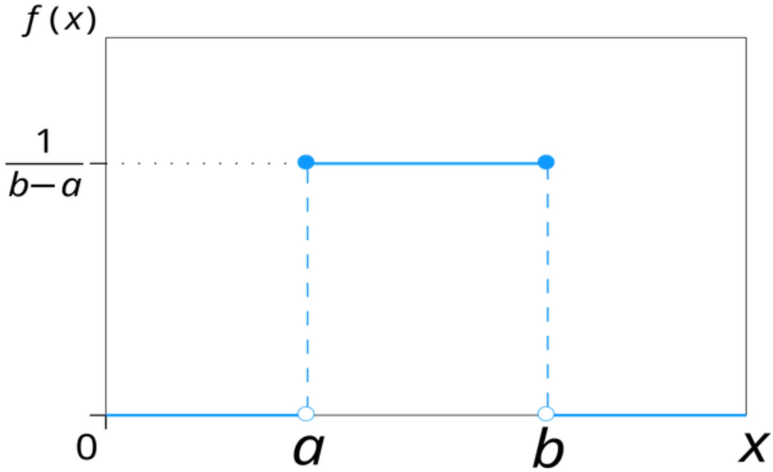
$\frac{1}{b-a}, a \leq x \leq b$

CDF: $F(x) =$

$\frac{x-a}{b-a}$

Notation: $X \sim U(a, b)$

Uniform Distribution



Triangular distribution

- The triangular distribution can be used for modeling situations, where non extremal (central) values are more likely than the upper and lower bounds.
- Take min, mode and max as inputs. Typically on the basis of subjective judgement:

PDF: $f(x) =$

$\frac{2(x-a)}{(b-a)(c-a)}$; if $a \leq x \leq c$

$\frac{2(b-x)}{(b-a)(c-a)}$; if $c \leq x \leq b$

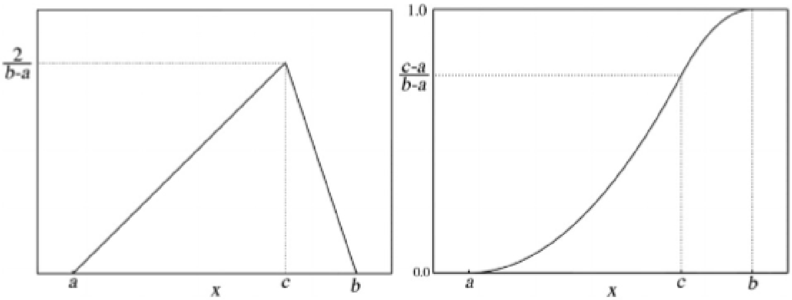
CDF: $F(x) =$

$\frac{(x-a)^2}{(b-a)(c-a)}$; if $a \leq x \leq c$

$1 - \frac{(b-x)^2}{(b-a)(c-a)}$; if $c \leq x \leq b$

Triangular Distribution

Notation: $X \sim T(a, b, c)$



It can be symmetric or asymmetric

Normal Distribution

- The normal distribution ('bell curve' or Gaussian) for modeling unbiased uncertainties and random errors of the additive kind of symmetrical distributions of many material processes and phenomena.
- A commonly cited rational for assuming normal distribution is the central limit theorem, which states that the sum of independent observations asymptotically approaches a normal distribution regardless of the shape of the underlying distributions

PDF:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\};\ -\infty \leq x \leq \infty$$

CDF: $F(x) =$

has no closed form solution but is often presented using the complementary error function solution

Normal Distribution

Notation: $X \sim G(\mu, \sigma)$

It is a Symmetric distribution around the mean

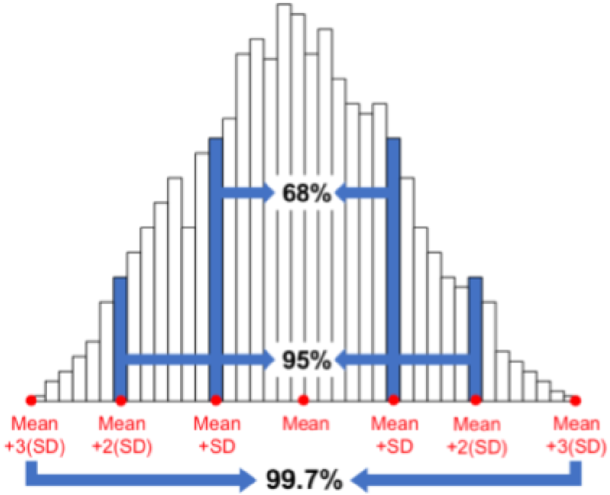
μ is the mean, σ is the standard deviation

$\mu \pm \sigma : 68.3\% probability$

$\mu \pm 2\sigma : 95.4\% probability$

$\mu \pm 3\sigma : 99.7\% probability$

Normal Distribution



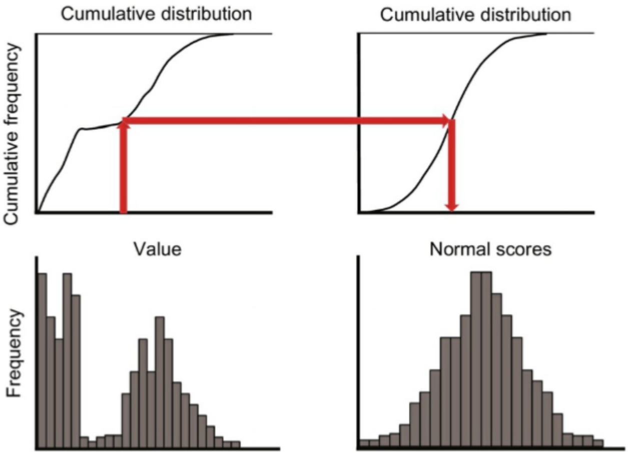
Data transformations

- Often, it is useful to transform a sample distribution into the space of an equivalent normal distribution, where many statistical operations can be easily performed and visualized
- The approach involves a rank-preserving one-to-one transformation.
- Transforming the data so that their distribution matches a prescribed (target) distribution.
- Sometimes we must transform the data...

Normal Score Transformation

1. From data to cumulative distribution.
2. From cumulative distribution and map back.

O Quantile-to-quantile normal score transformation



Match Quantiles

Log - Normal distribution

For a log-normal distribution, we define the standard normal variate as

$\alpha = \textit{means of } ln(x)$ $\beta = \textit{SD of } ln(x)$

Notation: $ln(X) \sim G(\mu, \sigma)$