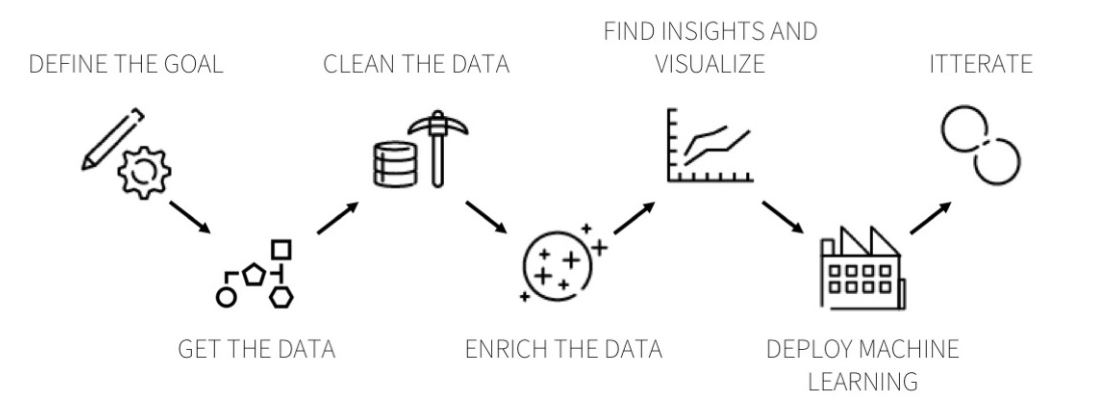
# Exploring Data with Statistics

# The Data Science Process



ds\_process\_2.png

# The Data Science Process

As we’ve seen before, the DS process consists of the following steps

| Step | Task |
| --- | --- |
| 1. Question | Clearly define the business problem |
| 2. Get Data | Obtain data from internal/external sources, APIs |
| 3. Wrangle | Clean messy data. Engineer features. Summarize, aggregate data. |
| 4. Explore | Visualize distributions. Investigate relationships. Build intuition for subsequent steps. |
| 5. Model | Build and Tune models. Select the best from competing statistical models. |
| 6. Interpret | Assess model performance on out-of-sample data. Understand results. Draw Insights. |
| 7. Deploy | Productionalize your analysis. Build a data product. |

# Agenda

In this module, we will focus on **#4 - Explore Data**.

Data, as we’ve seen, consists of numerical, categorical and boolean variables organized as columns in a DataFrame. This knowledge is critical.

The type of your data directly determines the tests that are available for the data analysis.

In the following sections we will learn about and apply topics in **Statistics** that will help us explore these variables and the relationships they share. Specifically, we will take up the following topics -

* univariate distributions
* reporting effect sizes and testing statistical hypothesis
* regression analysis

## Why study Statistics?

The world around us, its natural processes and the behavior of the living things that inhabit it are all **data generating machines** to a statistician. Every time you post something on Social Media or shop online or buy a ticket to a concert or take the train to work or click on an advertisement you produce data points. This data contains within itself moving snapshots of real-world processes and an understanding of these could potentially help you understand the past and, with rigorous application, predict the future.

## What is Statistics?

In a nutshell, statistics is a set of tools and procedures that help us in

* summarizing or **explaining complex phenomena** using simple and reasonable probability **models** (model *fitting*)
* Infer **underlying parameters** that may have generated the data
* generating **knowledge** about a population by analyzing **noisy** data from a sample (*inference*)
* **Make predictions** about unobserved data, or expected future observations.

## Types of questions that statistics can help us answer

* **Descriptive** to summarize the information contained inside a variable
* **Exploratory** to identify patterns, trends, or relationships between variables
* **Inferential** to test a hypotheses
* **Predictive** to forecast an unknown value or label
* **Causal** to investigate whether changing one factor will change another factor
* **Deterministic** to establish *how* the change in one factor results in change in another factor”

## Practical (and business) Questions answered with Statistics

Most of these deal with the measurement of differences between groups on a metric of interest

- Does one medicine work better than another?   
- Do cells with one version of a gene synthesize more of an enzyme than cells with another version?   
- Does one kind of signal processing algorithm detect pulsars better than another?   
- Is one catalyst more effective at speeding a chemical reaction than another?   
- Do girls perform better than boys in the SATs?  
- Does lowering the consumption of salt in your diet affect your blood pressure?  
- Do first babies arrive late?

To answer these questions, we would collect data and put them in a form that is easy to summarize, visualize, and discuss. Loosely speaking, the collection and aggregation of data result in a **distribution.**

Distributions are most often in the form of a **histogram** or a **frequency table**. That way, we can “see” the data immediately and begin our scientific inquiry.

After data is imported into Python, we begin **exploring** it, - first by looking at each variable in isolation (**univariate** analysis) and - then by exploring relationships between variables (**bivariate** or **multivariate** analysis.)

## Univariate Analysis

* visualising the probability distribution with a **histogram** using Series.plot.hist()
* determining the **summary statistics** such as mean, median, stddev using Series.describe()
* finding the underlying model parameters of the data

### Common Statistical Distributions

When you look at the histogram of a variable in your data, chances are that it will resemble one of these distributions. Next, we’ll look at the Python tools that will help in understanding these distributions, summarising them, and fitting them to the data provided.

# Statistical Distributions using scipy.stats

All functions concerning probability distributions are contained in scipy.stats. Overall, there’s an exhaustive list of

* **81 continuous distribution families**
  + examples include Beta, Cauchy, Chi-squared, Gamma, Weibull, Exponential, Pareto, Logistic, Gaussian
* **12 discrete distribution families**
  + such as Bernoulli, Binomial, Poisson

For each of these distributions available in scipy.stats, we have many crucial methods, such as:

* Generate random variates .rvs()
* Probability Density Function as .pdf()
* Cumulative Distribution Function as .cdf() and Survival Function as .sf()
* Summary metrics (Mean, Variance, Skewness, Kurtosis) as .stats()

There’s a simple pattern to access these: scipy.stats.<distribution family>.<function>

### Probability Density Function

The PDF, or **density** of a **continuous** random variable, is a function that describes the probability for a random variable to take on a given value . This function helps us answer questions such as: *What is the chance that a man is between 160 and 165 cm tall?*

* The integral over the PDF between and gives the probability of that lies in the range

### Cumulative Distribution Function

The CDF of a **continuous** random variable, is the function that describes the probability for a random variable to take on value **less than** . This function helps us answer questions suchs as: *What is the chance that a man is less than 165 cm tall?*

* Thus, the probability of that lies in the range will be given by

Both the PDF and CDF take the generic forms , where is the random variable, along with certain which characterise the distribution. As an example, let’s look at the PDF, CDF of the Normal Distribution.

###### PDF and CDF of the Normal Distribution

Here, and are the parameters of the Normal Distribution that represent the mean and standard deviationrespectively.

### PDFs and CDFs of Known Distributions

### PDFs and CDFs of Known Distributions

## Distribution Parameters

Each distribution is characterized by two or three parameters. These control the position, nature and shape of the distributions’s function on the X-Y plane. To inspect these, we use the .numargs and .shapes attributes of a distribution.

<distribution>.numargs  
<distribution>.shapes

It is important to understand the concepts of **location**, **scale** and **shape** for these continuous distributions.

* For a Normal (Gaussian) distribution
  + the loc parameter defines the mean of the distribution and determines its position on the X-axis
  + the scale parameter defines the standard deviation and hence controls the *spread* of the bell-curve
* For a Pareto distribution,
  + the shape parameter b determines what the pdf will look like

## Location and Scale

Here, we generate data from 3 normal distributions, each with a difference loc and scale value. Note how the loc controls position along the X-axis and scale controls the spread of the bell-curve.

Series(norm.rvs(loc=0, scale=1, size=1000)).plot.hist()  
Series(norm.rvs(loc=-5, scale=0.3, size=1000)).plot.hist()  
Series(norm.rvs(loc=6, scale=0.8, size=1000)).plot.hist()

The green histogram corresponds to loc= -5 and scale=0.3 which is why it’s located to the far left and has a narrow bell-shape.

## Shape

Here, we see how the parameter a of the gamma distribution controls the shape of its probability density function

As we can see, higher values of a produce a pdf that begins to take the shape of a bell-curve.

## Summary Statistics

Given a variable, we seek to characterize/summarize its distribution in one or two values that would give us a fair idea of what the distribution would look like, and allow us to compare distributions.

Some of the most common summary statistics are;

| ### Mean |
| --- |
| ### Median |
| - 50th Percentile |
| - A Median always exists |
| - Median is not unique |
| - Can be computed in linear time |
| - Not influenced by outliers (robust) |

### Standard Deviation

* Measure the ‘typical’ displacement (or ) from the mean value
* Standard deviation is popular because it has extremely nice mathematical properties.
* Standard deviation is a good deviation for normal distributed data
* Large effect on Outliers

The image above shows two distributions with different values of mean and standard deviation. Note that higher values of standard deviation correspond to a more ‘spread-out’ curve.

## A quick note on *Skewness* and *Kurtosis*

## stats.describe()

The describe function takes as input a numeric variable, and produces all summary statistics

The syntax is stats.describe(data) and it returns the following results:

* nobs: Number of observations.
* minmax: tuple of Minimum and maximum value of data array (or range)
* mean: Arithmetic mean of data
* variance: Unbiased variance of the data with denominator
* skewness: based on moment calculations with denominator
* kurtosis: normalized so that it is zero for the normal distribution

## Fitting a Distribution to Sample Data using the .fit() method

Let’s say you plot the histogram of a variable, and it vaguely looks like one of the known statistical distributions. You’re interested in finding out the and for this variable to be able to calculated probabilities and help people make decisions. Now, we know that these functions have unknown parameters. In order to be able to use these functions to find probabilities, we must first *estimate the parameters that fit the data*.

As an example, consider a variable that has the following values

my\_data = array([ 2.36, 7.82, 1.75, ..., 1.62, 1.86, 3.13])

You plot the histogram and it looks something like this

* You realize that this resembles a slightly skewed normal distribution.
* You recall that the PDF and CDF of such a distribution are characterized by two parameters - and

Now, to find the PDF and CDF, there are 2 ways:

## 1. Maximum Likelihood Estimation (*parametric*)

* The .fit() method of each distribution type return MLEs (or Maximum Likelihood Estimates) for shape, location, and scale parameters as learned from the data.
* We plug these values into the PDF and CDF and proceed to find the probabilities required.

from scipy.stats import norm  
mu1, sig1 = norm.fit(my\_data)  
  
# Find P(X=x)  
norm.pdf(x, mu1, sig1)  
  
# Find P(X <= x)  
norm.cdf(x, mu1, sig1)

## 2. Kernel Density Estimation (*non-parametric*)

Kernel density estimates are closely related to histograms, but can be endowed with properties such as smoothness or continuity by using a suitable kernel. The gaussian\_kde() estimator applied to univariate, unimodal data returns the PDF directly.

kde1 = stats.gaussian\_kde(my\_data)  
  
# Find P(X = x)  
kde1.pdf(x)  
# --- or use kde1(x)  
  
# Plot the PDF  
x\_1 = np.linspace(my\_data.min(), my\_data.max(), 100)  
Series(kde1(x\_1), index=x\_1).plot()

# Quiz

* The PDF of a distribution can be used to find the probability that the random variable takes on a value greater than or equal to . True or False?
* What would be a good distribution to model the following
  + Number of people in a checkout queue at the mall.
  + Prices of apartments on rent on AirBnb in San Francisco.

## Whether an employee will leave the company in the next six months.

# Practice - 10 mins

mtcars data

# Install altair using  
!pip install "altair[all]"  
# or   
!conda install -c conda-forge altair-all  
  
# Then  
from vega\_datasets import data  
cars = data.cars()

* Import that data using pandas and explore it.
* Draw histograms for the mileage of automatic and manual cars.
* Which distribution would you use to model these? Why?
* Find that probability that the mileage of a) automatic b) manual cars could be higher than 25 using parametric methods.
* Find a) and b) above again, but this time only use non-parametric methods
* Compare the results. Is there a difference? Why?