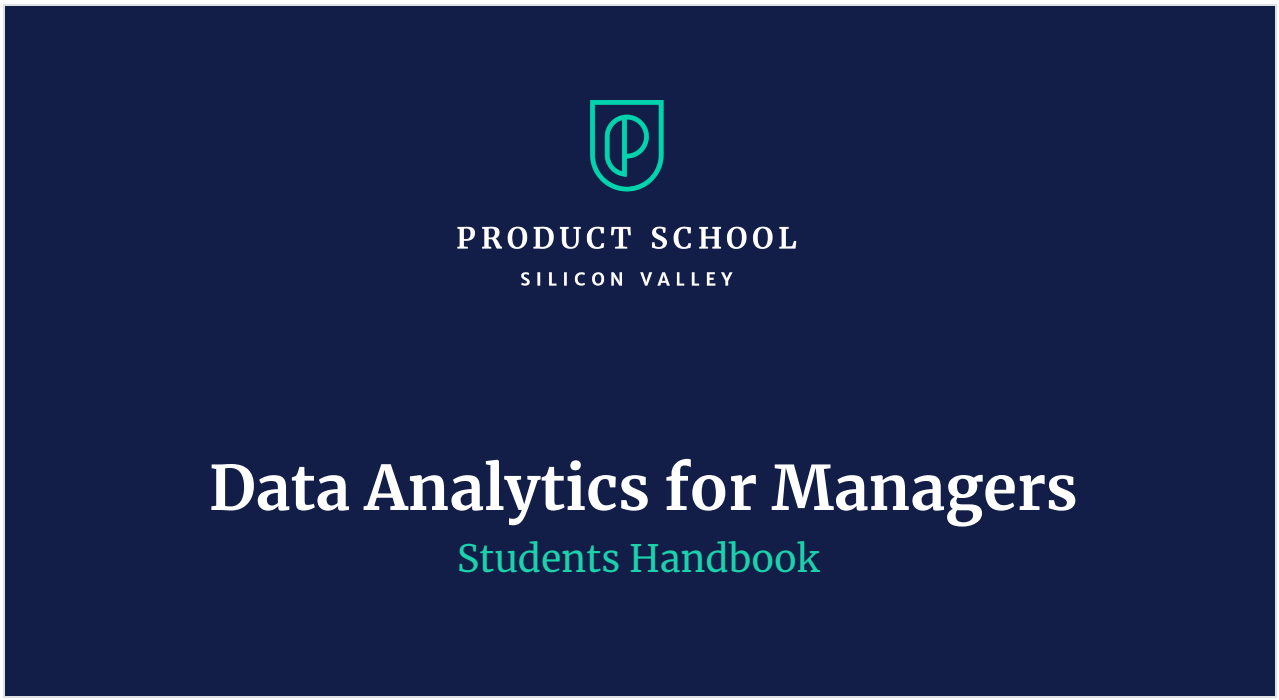

PRODUCT SCHOOL
SILICON VALLEY
Data Analytics for Managers
Students Handbook

Program Management Team

Program Managers:

Camila Thury
Shewit Doherty
Victor Silva
Richard Magpantay

Email: students@productschool.com
Slack ID: @camila, @shewit, @victor, @richard

We're here to help you with any questions about the course, student perks, workshops, and logistics throughout the program. You can reach out to us via email to students@productschool.com

In addition to that, please take a moment to read the student handbook where you can find valuable information to help you get ready for the class.

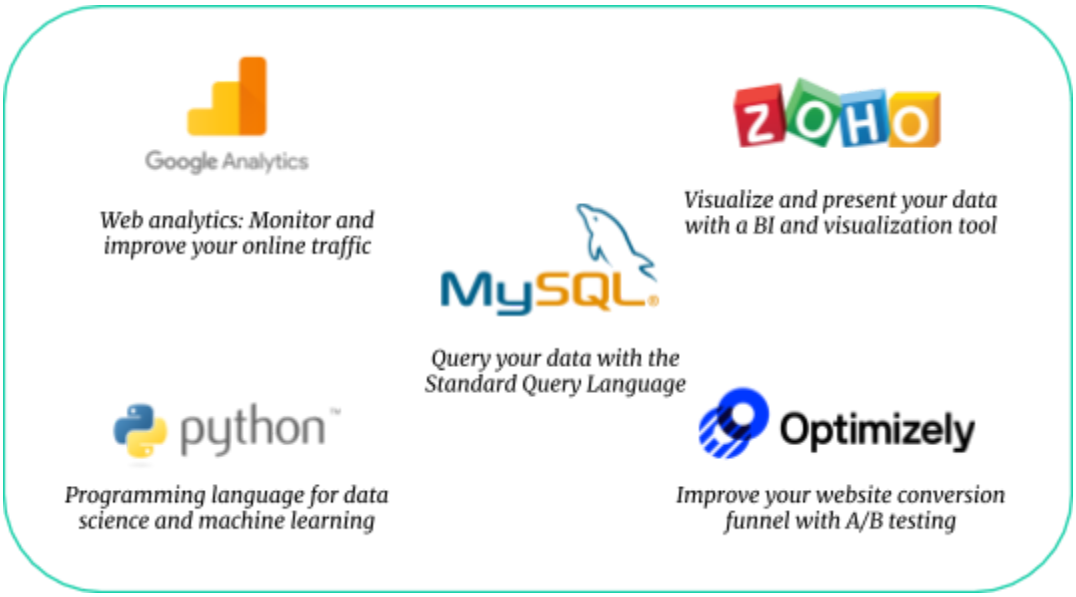
Data Analytics for Managers
Topic Syllabus

Topic 1A	Introduction to Data Analytics & Web Analytics with Google Analytics - Part 1
Topic 1B	Introduction to Data Analytics & Web Analytics with Google Analytics - Part 2
Topic 2A	Introduction to Databases and SQL - Part1
Topic 2B	Introduction to Databases and SQL - Part2
Topic 3A	Advanced SQL - Part1
Topic 3B	Advanced SQL - Part2
Topic 4A	Course Review
Topic 4B	Data visualization
Topic 5A	Statistical Thinking
Topic 5B	Estimates and Sample Sizes
Topic 6A	Introduction to Machine learning
Topic 6B	Machine learning Models: Linear Regression & Classification
Topic 7A	Machine Learning & Python programming - Part 1
Topic 7B	Machine Learning & Python programming - Part 2
Topic 8A	Big data, and machine learning review
Topic 8B	Final Project presentations

Notes and Exercises

Topic 1 A/B

Introduction to Data Analytics & Business Analytics with Google Analytics



Software, programming languages and tools that will be covered in this course.

Defining the buzzwords

When it comes to data science, as for any new and trendy topic the definition and use of words tend to vary rapidly, and depend a lot on the user. Nowadays everybody talks about data and AI, but with a different understanding of the underlying terms.

Notably, the term “analytics” is used more and more in different contexts. According to Wikipedia (https://en.wikipedia.org/wiki/Analytics) analytics refers to the "extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based management to drive decisions and actions."

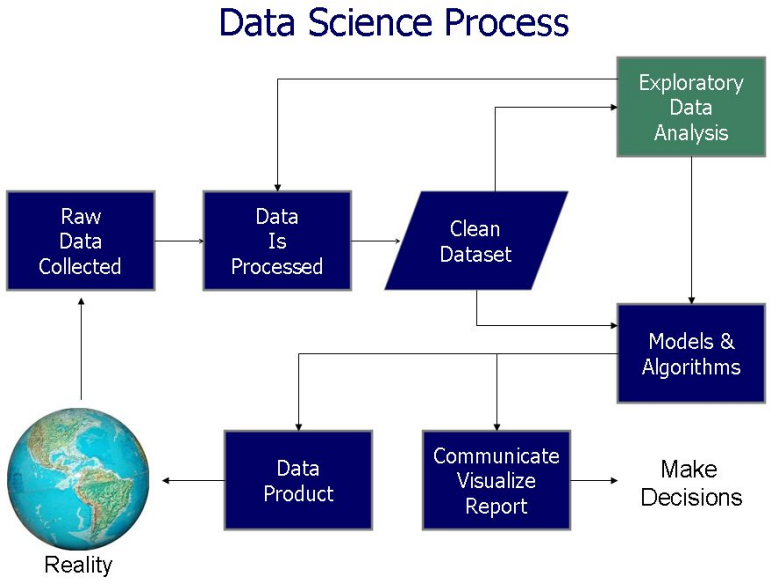
📄 **Business intelligence (BI)**: BI is a rather old term in the corporate world (from the 70s!). It refers to monitoring Key Performance Indicators (KPIs), analyzing trends, and leading the global reflection on data storage and organization.

📈 **Business/data analytics**: Analytics can be viewed as the evolved or improved BI. In includes BI and adds KPIs forecasting and trends prediction to it.

⛏️ **Data mining**: A term related to patterns discovery in datasets. Compared to BI it relies heavily on advanced statistical and mathematical methods, and tries to discover patterns that are not explicitly sought after in datasets.

🔎 **Data analysis**: A field similar to analytics, but does not necessarily sits in a business or corporate context.

🔪 **Data science**: Data science is an umbrella term for a lot of different methods, including data analysis, machine learning, data engineering, and all of the above terms. All the techniques mentioned so far more or less follow the data science process depicted in the figure below, with the end goal to influence business decisions, allow business monitoring, and possibly automate some processes.



The data science process illustrated: from raw data collection to modelling and decisions support.

An overview of the BI and analytics software and platforms

Every year, a global research and advisory firm named Gartner (https://www.gartner.com/en) publishes a global analysis of the main digital tools and practices in use in several domains related to IT, Finance, HR, Customer Service and Support, Legal and Compliance, Marketing, Sales, or the Supply Chain.

The following figure is their so-called Magic Quadrant for 2019, the purpose of which is to give you an overview of the main BI tools you may hear about.

Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms

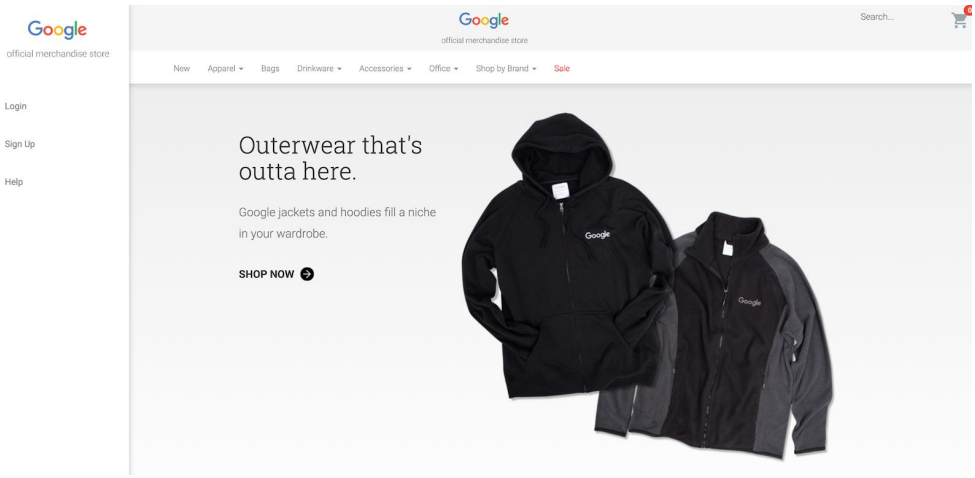
Source: Gartner (February 2019)

The 2019 Gartner quadrant for Analytics and BI platforms. In practice, the market is largely dominated by a few of those names (Microsoft PowerBI, Tableau Software, Salesforce, ...) but this landscape evolves fast.

Introducing our running example: the Google merchandise store

(https://www.googlemerchandisestore.com/)

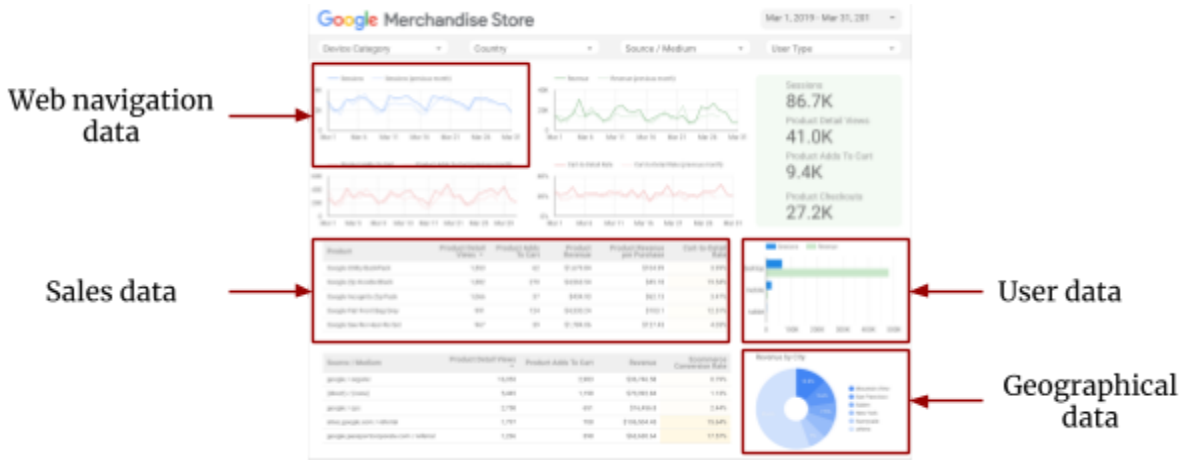
In the beginning of this class, we will see in detail the Google merchandise store data to illustrate the concepts to come. The Google merchandise store is a website on which Google sells goods with their logo, like hoodies, bags, water bottles... Feel free to explore the website for a few minutes in order to get an idea of what it is.



The google merchandise store home page.

The Google merchandise store is very convenient example because the business model is simple to understand, the data is freely available via Google Data Studio (https://datastudio.google.com/) (a data visualization tool) and Google Analytics (https://analytics.google.com/analytics/web/) (a traffic, online advertisement, and search engine optimization analysis tool for websites, more on this soon), and the large volume of data makes statistical analyses highly relevant.

The data provided by Google notably contains browsing, users, sales, and geographical data. It can be visualized here. (https://datastudio.google.com/u/0/reporting/0B2-rNcnRS4x5UG50LTBMT0E4aXM/page/nQN)



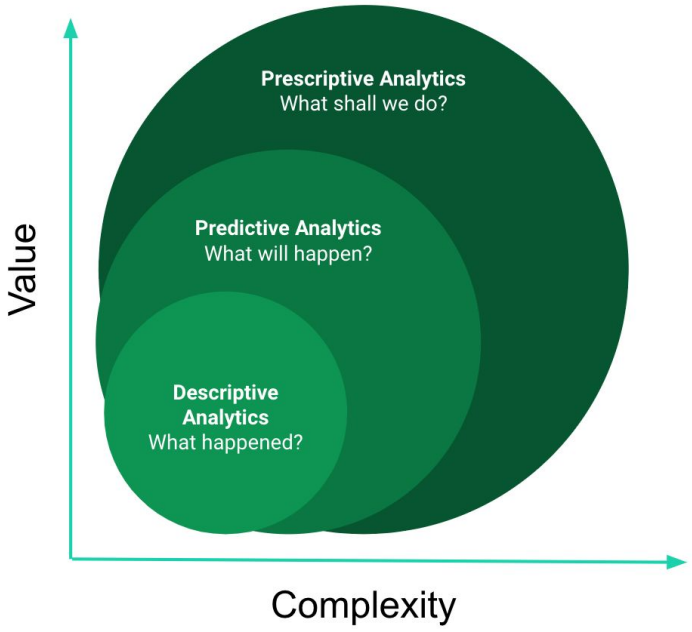
Part of the Google merchandise store data, visualized through Google Data Studio. Explore the data here. (https://datastudio.google.com/u/0/reporting/0B2-rNcnRS4x5UG5oLTBMToE4aXM/page/nQN)

Different analysis types: descriptive, predictive and prescriptive

In the analytics world, analysis are often classified depending on their degree of complexity and anticipation. While some of the corresponding classifications have many categories, we will stick to the main three.

- **Descriptive analytics**: The most basic type of analysis one may perform is a descriptive analysis. In the latter, you investigate and aggregate data from the past and describe what happened until now. For instance, in the Google merchandise store example you may compute and plot the number of hoodies that were sold year after year, and try to understand the resulting curve.
- **Predictive analytics**: Once you get a solid understanding of your past data, you have the option to enter the realm of predictive analytics. The goal of predictive analytics is more ambitious and aims at forecasting future trends, or the impact of hypothetical actions (e.g. targeting different groups of users with online advertisement, opening a shop in a new city, ...). It relies on statistical modelling and machine learning, and has an inherent degree of uncertainty that must be taken into account when taking decisions based on the corresponding predictions.

- **Prescriptive analytics**: Even more valuable and complex is prescriptive analytics, a field in which you build an action recommendation system using a combination of business rules and machine learning algorithms. The output of this kind of analytics is a recommendation like. It's rather new and not widely spread in companies, due to its complexity.



Please note that **no single type of analytic is better than the other**, they depend on the previous level. The starting point is always descriptive analytics, because you cannot leverage predictive or prescriptive analytics without a solid understanding of your past data.

Descriptive analytics	• Monitor KPIs, analyze past trends • Tools: visualization softwares • Example: "we've sold 10,000 hoodies last year"
Predictive analytics	• Predict future trends • Tools: machine learning, stat. modelling • Example: "because of campaign A, we'll sell 1,500 hoodies next month"
Prescriptive analytics	• Give business recommandations • Tools: ML and business acumen • Example: "we should spend $10k more on FB ads to increase sales by $15k"

Differences between the 3 main types of Analytics subfields, with examples of conclusions drawn from them.

The analytics and data science life cycle

When starting an analytics project, many questions have to be anticipated. Countless projects end up in a dead end because the data is not available, there are legal issues with data usage, or the scope and goals were poorly defined. Here is a useful list of the different steps involved in data science or analytics project, together with questions that have to be addressed at each of theses steps (from the Berkeley *Principles and Techniques of Data Science* (https://www.textbook.ds100.org/ch/01/lifecycle_students_1.html) course):

1. **Question/Problem Formulation**:
    a. What do we want to know or what problems are we trying to solve?
    b. What are our hypotheses?
    c. What are our metrics of success?

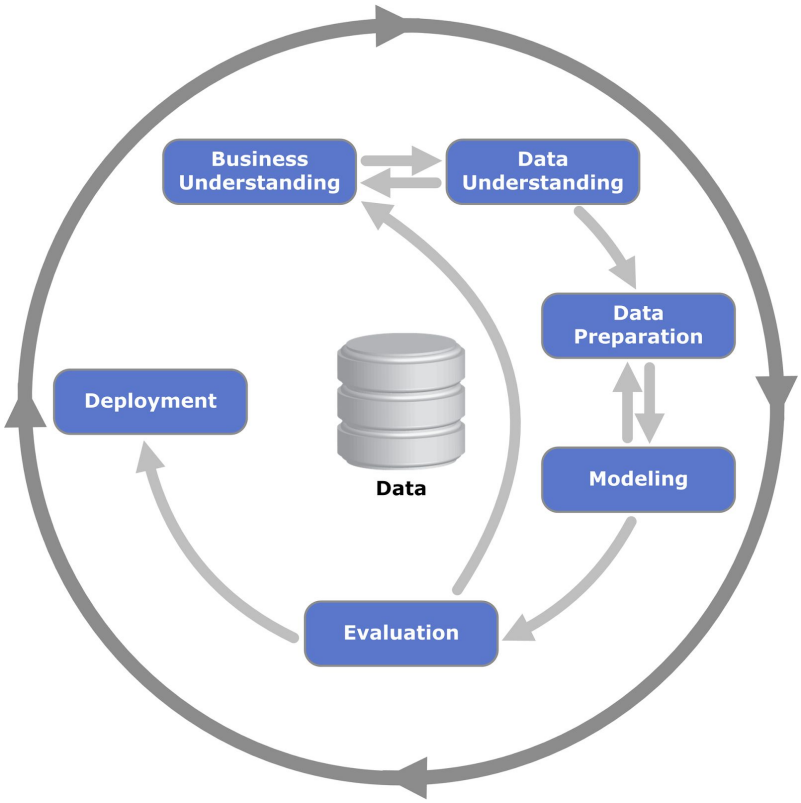
2. **Data Acquisition and Cleaning**:
    a. What data do we have and what data do we need?
    b. How will we collect more data?
    c. How do we organize the data for analysis?

3. **Exploratory Data Analysis**:
    a. Do we already have relevant data?
    b. What are the biases, anomalies, or other issues with the data?
    c. How do we transform the data to enable effective analysis?

4. **Prediction and Inference**:
    a. What does the data say about the world?
    b. Does it answer our questions or accurately solve the problem?
    c. How robust are our conclusions?

Please read this list carefully, as all of those questions are crucial for driving a successful analytics project, and we all too often forget to address them all as thoroughly as we should.

Another useful representation of the analytics and data science life cycle is provided by the "Cross-industry standard process for data mining" (CRISP-DM) diagram shown below. Although it refers to data mining for historical reasons, it applies equally well to any Analytics process.



The Cross-industry standard process for data mining (CRISP-DM) gives an overview of data science projects flow. Note the cycles and back arrows between the different phase. Leveraging data involves a lot of back and forth play between understanding, preparing, and modelling your data.

From analysis to recommendations: introducing SMART goals

The ultimate goal of analytics is to provide business recommendations in order to improve customer satisfaction, increase revenue, reduce costs, or optimize processes.

A common misconception about analytics is that you can use it to find insights automatically. You actually have to formulate precise questions in order to answer with data. The best starting point for doing so is a well-defined goal.

The relevance and precision of business, personal, or collective goals can be improved using the SMART methodology. It simply consists of a checklist of five criteria that must be ticked in order to have well-defined goal. The SMART checklist should not be overlooked despite its simplicity. It is a powerful tool and when you using it to refine a goal, more often than not people realise how fuzzy a goal was after they go through the SMART goal checklist.

The SMART goal checklist. Make sure you tick every of this five boxes in order to have well-defined goals. Note that the exact meaning of the acronym varies from one source to the other, and you may read about SMARTER (https://www.wanderlustworker.com/setting-s-m-a-r-t-e-r-goals-7-steps-to-achieving-any-goal/) goals. The latter just add the Evaluation and Readjust criteria to the SMART ones, the goal of which is to ensure continuous monitoring and refinement of the SMART goals.

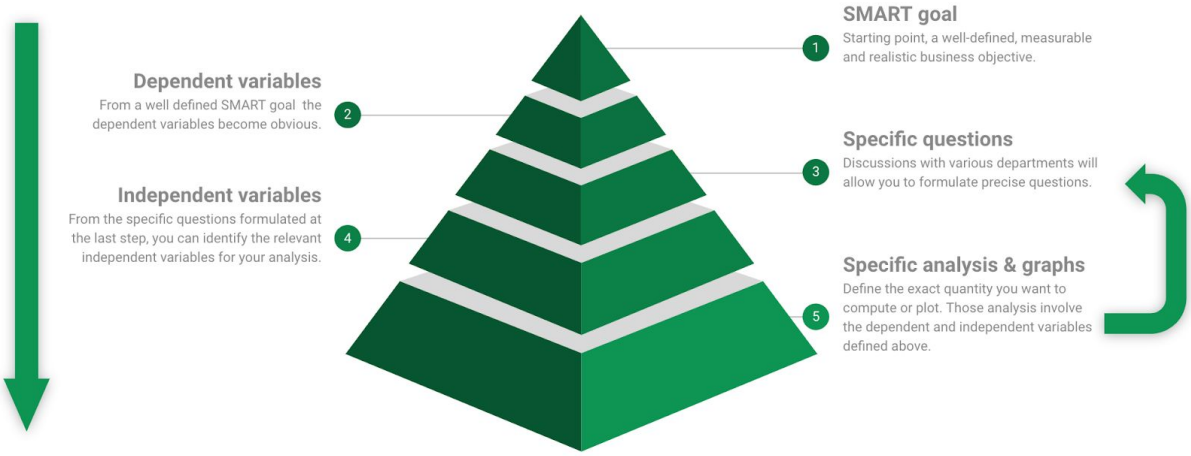
Dependent and independent variables

In any business analysis it is key to identify the variables you can directly control (e.g. price, ads, website color, ...), called **independent variables** (a.k.a. "input"), and the variables you want to monitor and influence (e.g. number of sales), or **dependent variables** (a.k.a "output", "target").

In most settings you typically want to plot the dependent variable versus the different independent variables in order to understand how the latter influences the former.

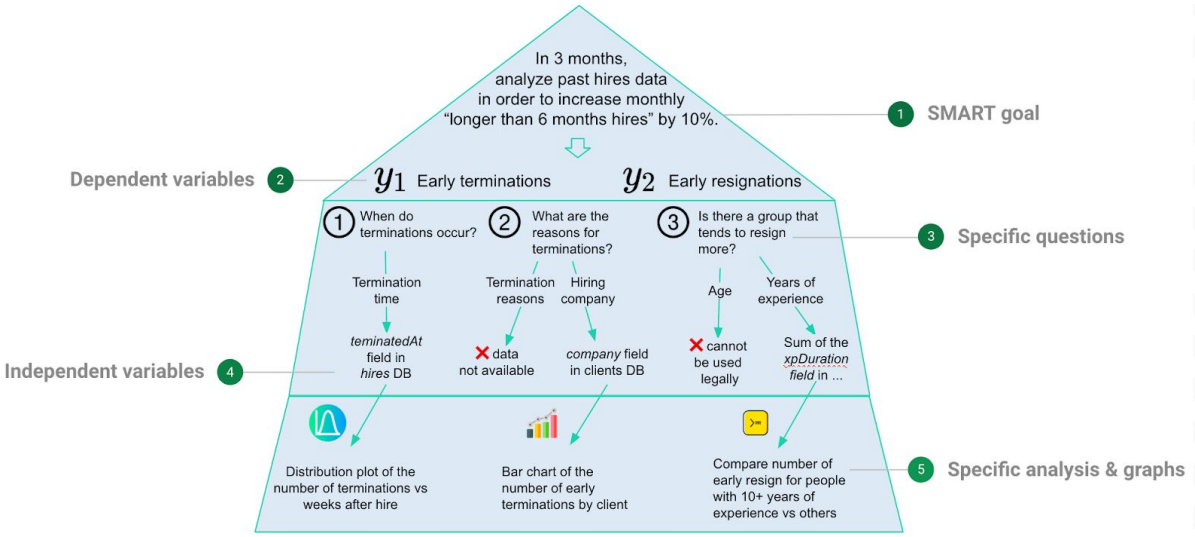
Structure your analysis with a Structured Pyramid Analysis Plan

Once you identified the SMART goal of your project, the next step is to unroll that goal into a series of concrete action steps toward a rigorous analysis. One of the possible approaches to structure your analysis is the Structured Pyramid Analysis Plan framework.



*The **Structured Pyramid Analysis Plan** consists in 5 steps. You start by formulating your SMART goal, from which you deduce one or several relevant dependent variables. Further thinking and discussions will allow you to formulate precise questions regarding the above defined dependent variables. For those specific questions you can identify independent variables, the influence of which you would like to understand. The last steps consists in defining precisely all the graphs and quantities you want to compute. The resulting analysis may lead to new specific questions.*

The following figure is a detailed example of a Structured Pyramid Analysis Plan.



Structured Pyramid Analysis Plan example for a recruitment business. From the SMART goal ("increase long hires"), two relevant dependent variables are identified (number of early terminations and early resignations, that we both want to reduce). Specific questions relative to those dependent variables can then be identified. The specific questions can be scrutinized in with relevant independent variables that have to be defined precisely here. Some variables' computation will be trivial, but some others have to be defined precisely in this plan because their computation is not obvious from the database structure. Please note that some questions cannot be addressed for technical or legal reasons. Lastly, we can define the precise graphs we want to plot or precise quantities we want to compute. Some analysis will bring answers or additional questions because there is an effect that sticks to the eye, and some others will be dead ends.

Bear in mind that the Structured Pyramid Analysis Plan is just one method to structure your analysis, and you may use a method of your own. But the highlight of this section is that it is important to detail your analysis plan in order to guarantee the consistency of your project, and the alignment of your goal and subsequent analysis.

Web Analytics with Google Analytics

Understand Web Analysis

We will start our practical Analytics journey with Web Analysis where you'll learn how to effectively use Google Analytics and understand your users.

What you will learn in this course:

- Learn how Google Analytics is built
- Understand the "Analytics cycle"
- Use Audience, Acquisition, Behavior and Conversions reports
- Build a plan to measure your website's performance
- Implement that plan using Google Analytics reports

ABCs of Web Analytics with Google Analytics

Google Analytics is a fantastic free tool to understand what is happening on your digital application. It is not only meant for your website, you can use Google Analytics with almost any tool that can connect with the internet. Anytime a user connects to your application, Google Analytics will start collecting data, processing it given the configuration you set up, store that data in a database and retrieve it through reports. We will dive into how to use Google Analytics effectively in your company so that you can start making smarter decisions.

Analysis Techniques

Google Analytics is based on two analysis techniques: **segmentation** and **context analysis**.

Segmentation consists in dividing your aggregate data into segments. For example, you might want to segment your data by marketing channels to understand which one has the best performance and which one you should look at. But you can also segment your data in a lot of different ways like by geographic location, gender, age, browser, etc.

Google Analytics will do most of the work for you when it comes to most popular segments such as Geographic location, Gender, Age and which browser your users are using. Your work as an analyst will be to choose which segments are the most pertinent for your business and create meaningful reports accordingly.

Context Analysis helps you measure a website's performance. There are two ways to do a context analysis: either **internally** or **externally**.

With the internal context you compare performances with historical data. For example, say your number of visits of last month is 5,000. This number means nothing if you don't compare it to the number of visits of the previous month. Depending on the result of this analysis you are able to say if your company's or website performances are improving or worsening.

The external context compares your performance with other businesses. You will of course need to benchmark how your competitors' perform and then compare their KPIs with yours.

Metrics and Dimensions

Google Analytics differentiates metrics from dimensions. Metrics are quantitative measurements such as number of users, number of sessions or number of clicks on a page. Metrics are numerical.
Conversely, dimensions refers to non-numerical data. It can be geographic locations, pages viewed by users, or the source a user came from before visiting your website. Usually dimensions and metrics go in pairs in reports. For example, you can see all the traffic sources of your website and then see all the metrics corresponding to that (Number of users from that particular source for example).

Account Structure

In Google Analytics, you will have an account that usually corresponds to one business. Then in this account, you have a property that is a subset of what your business does. It can be the e-commerce version of your site and your blog. It could also be your website and your mobile app. Then, under each property, you will have several views. A view will display your data in a certain way. For example, you may have a view that filters out all your internal employees or that display only data from a certain country etc.

Setup Filters

Filters will help you improve the quality of your reports. Within each view, you will be able to show a subset of your data with filters. For example, you might want to exclude your employees.

Please note that Google Analytics is case sensitive, so if you have 3 sources named "page.html", "Page.html" and "PAGE.html", Google Analytics will consider these as 3 different sources.

To set up a filter go to *Admin* > *View* > *Filters*, then click on "*Add a filter*". You will just need to enter a name, choose between predefined and custom filters, then define your filter. Predefined filters are very easy to setup and most of the time sufficient. However, you can use custom filters if you need to.

Web Analytics and the Analytics Cycle

When it comes to your application, you want to understand how your customers reach your website, what they do once they have reached it, and finally if they do end up buying/using what you are offering. This is what is called the "analytics cycle".

Acquisition

In this section, we want to know how your users get to your website. Google sorts your acquisitions by sources and mediums. You can then dig into the advanced settings to have a better distinction between each source and medium.

The **source** is the origin of your traffic, such as a search engine (for example, Google) or a domain (example.com). For example, imagine someone read a blog article that contains a link to your website, and the user followed this link. Let's say the URL of that website is www.supercoolblog.com/article1. What you'll see as a source in Google Analytics will be this exact URL. If the same user searched "Super Cool Blog" on google and clicked on your website in the search results, then the source will be Google.com.
The **medium** is the general category of the source, for example, organic search (organic), cost-per-click paid search (cpc), web referral (referral), or direct access (direct/none).
Organic corresponds to a user searching for your site on Google search engine and clicked on your website. For example, let's say you have a blog about Product Management. If a user types they keywords "Product Management" on Google (or other search engines) and click on your blog, then the medium will be organic
Referral is a user who found a link to your website on another website. For example, a blog wrote an article about your company. This traffic source will correspond to referral.
Direct / None is a user who directly typed the URL of your website. Be careful here though because Google will put any unknown traffic sources as Direct / None. So If you have a significant amount of your Direct traffic source, you might need to see with your IT team for a bug somewhere in Google Analytics.

Custom sources and mediums

The sources and mediums mentioned above are defined by Google by default. However, this might not fit to your business. As it gains in complexity, you might want to define custom sources and mediums. To do so, you will need to set up custom URLs in Google analytics that you will use wherever you need them.
For example, if you have an email campaign, instead of using the normal URL of your site (ex: *www.supercoolblog.com/article1*) you will use a URL that looks like this one: *www.superblog.com?utm_source=newsletter&utm_medium=banner&utm_campaign=junenewsletter*
Thanks to that URLs, Google Analytics will recognize that your source will be from a "newsletter". Inside that newsletter, you made a banner where inserted that link, hence your medium is "banner" and this corresponds to a campaign called "junenewsletter". It will appear clearly in your Google Analytics reports thanks to that link. The main advantage of this is that you have the ability to track perfectly where your customers are coming from.
Tips :

- The simplest way to create those URLs is to use the **Google URLs campaign builder**. However if you have many URLs to create at once, you can build them manually using spreadsheets. In any case, it's always good to keep track of your URLs somewhere so that you can refer back to them when needed.
- Make sure that your terms are clear and that everybody in your team understands them. For example, if you have several newsletters in your junenewsletter campaign, you might need to give a more descriptive name.
- Long links are always a little scary for users so use urls shorteners like Bitly or Google urls shortener
- If you are using Google Adwords for your paid campaigns, you don't need to build those URLs since Google Adwords will create them for you. The only thing you need to do is to link Google Adwords to Google Analytics.

Detect Keywords used with Search Console

The last thing you need to know regarding Acquisition is what Keywords people type on search engines to find your website. Google Analytics tracks this through Search Console. The only thing you have to do is to connect your Google Webmaster Tool to your property on Google Analytics and it will start tracking what keywords users are using to find your website.
Google webmaster tool will help you manage the relationship between your website and google search engine. If your website hasn't been setup with Google Webmaster Tool, You will need the help of your engineering team because it requires technical skills that are beyond the scope of this class.
Tips:
- You will most likely see a (not set) in Google Analytics regarding keywords. The official explanation from Google is that they filter out some keywords for privacy reasons. (not set) then corresponds to the sum of all the filtered out keywords.

Evaluate your Acquisition

There are different ways to evaluate the performance of your acquisitions channels. The main metrics that Google Analytics provide are the following:

Users VS New Users

Given the IP address of your users, Google Analytics is able to know if a particular user is new or not. This is very useful when these metrics are put in perspective. Depending on your business goals, you will want to optimize new users or returning users.
For example, if your website is a landing page that aims to showcase your product and get people to sign up to your newsletter, you want to maximize you new users. On the other hand, if you are a website like Reddit, you want to maximize retention since your community is already built.
Usually your websites are more complex and you need to balance the two metrics. For example, if you are video game hosted on the web, you want to build your community so you need new users but once new users tried your game, you want them to stay and keep playing. The way you handle those two metrics will really depend on the goals you set in the first place, either retention or acquisition, and how much focus you're giving to each of them.

Sessions

A session corresponds all the interactions a user makes on your website. It can be clicking on an image, on another page etc. There are few important things to distinguish with sessions. By default, Google sets a session to 30 minutes of interactions with your website.
However, given your business, you might want to set that differently. For example, a blog with long articles might have sessions that are longer than 30 minutes. Same thing if your website displays long videos (Youtube for example has hours long videos). On the other hand, a website that need fewer interactions such as Google search will have shorter sessions.
Session is a good indicator to measure people's interest in your website. Most likely you will see more sessions than your total number of users since one user can have more than one session. If you have a lot more sessions than users, it means that your users are returning a lot to your website. Depending on your business, you will want to maximize the number of sessions over the number of users. For example, if you are an e-commerce, you might want the same people to come back often to buy new things.
However, be careful with sessions. To be able to draw conclusions with sessions, you have to make sure that the time you set within your session correspond the time a user should spend on your site within one session. For example, an online calendar (such as Google Calendar) should have short sessions. No users will spend more than 15 min looking at their calendar. However, a video platform such as Youtube will have longer sessions.
Sessions can also easily become a vanity metric. There are plenty of robots on the internet that will go on your site and create sessions so that you could see them in your Google Analytics. Until you exclude them from your reports, Google Analytics will count those as sessions.

Click Through Rate
One final metric that is important to look is the Click Through Rate (CTR). Each time someone type keywords in a search engine, your website might appear in the results. If your website appears then Google Analytics count it as an impression. However, appearing in the search results doesn't necessarily mean that the user will click on your website. Hence the Click Through Rate metric.

$$CTR = number\ of\ clicks / number\ of\ impressions$$

In plain english, CTR represents the percentage of people who went to your website in the total number of people who saw your website in search results. This metric is really helpful to determine which keywords convert best and which don't. It is especially great when you want to do SEO (search engine optimization) or sponsor keywords to drive more traffic.
A low CTR could mean several things that can be related to the title or description of your website in Google. This should be solved with your engineering team to change this in your website. It could also mean that you sponsored the wrong keywords in your campaign and actually people are not looking for what you proposed in your website. Finally it can also mean that you are not ranked high enough in the search results. This implies to work on your SEO.

Behavior

Set a desired action

Now that you brought users to your website, the next thing you would like to know is what they do once they are in your website. This where the behavior section of Google Analytics comes handy.
First though, you need to know what are the desired actions that you want your users to do in your website so that you can then compare with the actual behavior on your reports and see if the goals are met.
A goal will be different for each company. For example, if you are Flickr you want to showcase a gallery of photos that people can download or if you are a restaurant you might want your users to go to the contact page so that they can give you a call to reserve a table.

Evaluate your user behavior

The best way to evaluate your user behavior is to look at key metrics that google analytics provide you. Those are the number of times a user viewed your page during a session, then you can check the bounce rate and final the percentage of exits on that page.

Total Pageviews VS Unique Page Views
This is metric is pretty straightforward. The number of page views corresponds to the number of times a user came on a page whereas unique page views are counted only once per user.

Those are pretty important metrics since they show how much a user interacted with your site. It's also a good quick way to check which of your pages are the most popular in your website.
Most likely, you will have more page views than unique page views since it is very probable that a user will come back to page for any reason. However, depending on your website and goals, you will want to minimize or not the difference between unique page views and page views.
Indeed, if you have an e-commerce site, once your users select to checkout, you want the process to be as smooth as possible. This means that you want them to be one time per page. Therefore, you want the number of unique page views and the number of page views to be as close to equal as possible.
Careful with the time range you chose, you might have a difference between page views and unique page views because your report has been set for the past week. Then if a customer executes a checkout several times during that time range, you will see more page views but still the same amount of unique page views.

Bounce Rate
The bounce rate is the percentage of people that didn't interact with your page. They basically just clicked on your page and then exited without clicking anywhere else. For example, someone clicked accidentally on your website
Generally, you want to have the lowest bounce rate possible because it would mean that your users are interacting more on your website. However, a high bounce rate means different things. One way to interpret a high bounce rate is that your target audience doesn't get what they were expecting coming to your site. Either you made a mistake when trying to find the right target audience or your target audience doesn't understand what is your product.
However, a high bounce rate is not necessarily a bad thing. A lot of websites will inherently have a higher bounce rate. For example, blogs don't really need their users to go anywhere else on the website once they read the article. Bloggers will then most likely see a high bounce rate on their Google Analytics report because a lot of their users are going to read the article and then exit the website without doing anything else.
Then what is important is to see the average time a user spent on your page to see if they were actually reading your article or not.

Exit
At some point your user will leave your website. Google Analytics keeps track of it with an Exit rate that is the number of exits divided by the number of page views.
Your exit rate should correspond to your goals and the way your website is built. For example, if you are an e-commerce, you want your exit rate to be high on the "thank you" page after checkout. However you want a very low exit rate on any landing pages of your website. This applies to sites that are not blogs or single page websites where the landing page will most of the time be the only page a user will see.

Conversions

The last step in the Analytics cycle is conversions. Once you acquired users and know how they behave, you want to see if your website actually converts. A conversion corresponds to the moment where a user fulfilled a goal you set. For example, it can be signing up to a newsletter, filling out a form, bought a product etc. To be able to track conversions, the first thing you need to do is to decide what your goals are.

Build a plan

This five steps plan has been built by Avinash Kaushik, Digital Evangelist at Google and is very powerful to set your goals that are relevant for any businesses regardless of their size.

Business Objectives
First step is to know what are you business objectives. You want to know why your business exists and what are your values. For example, Google's business objective is to organize the world's information, Stripe's business objective is to re-invent payments online or Trello's business objective will be to help people be more efficient and achieve their goals faster.
Business objectives will be very different from a business to another but they are very important to define as they will be the guidelines for the rest of your plan.

Business Strategies & Tactics
Once you define your business objectives, you can decide the strategies and tactics to achieve your business objectives. A strategy will help you achieve your business objective whereas the tactic will help you execute your strategy. You should have strategies for each objective as well as tactics for each strategy.

For example, with Stripe, one strategy could be to simplify the way to implement payments on a website. There are plenty of ways to simplify payments online, the first tactic could be to build very simple APIs that developers can embed easily in their code. Another way would be to have one product per type of payment (e-commerce, marketplace, subscription).
Another strategy would be to attract best talents to their company. One tactic could be to have a blog talking about how happy Stripe's employees are.

There are usually 5 common digital strategies:

Business	Strategy
E-commerce	Selling products or services
Lead Generation	Collecting potential leads
Content Publishers	Engagement & frequent visitation
Online information	Help customer find information
Branding	Driving awareness, engagement & loyalty

Even though your business might not fall into those categories, it might still help you with some part of your strategies. For example, if you decide to build a blog for your business, then you'll fall into the content publisher category and know that you will need to focus on user engagement and visits.

KPIs
You now need to define your Key Performance Indicators (KPIs) that will help you see measure your strategies and tactics performance. For example, for an e-commerce, a good KPI is the revenue generated on the website. If you are a blog, you might want to see the number of shares you get per articles.
Keep in mind that KPIs must be simple to understand and checked frequently. Be careful also with vanity metrics which won't really show you performance but will only make you feel good about your business. A typical vanity metric is the number of likes on a social media page. It's very easy to get and it doesn't mean that people who liked your page actually like your product.

Segment
Once you defined your KPIs, you want to see which segments of data are important of measure. For example, you might to see which marketing channels are performing best. Or you might want to segment your data by geographic location and see which countries engage the most with your content.
Segmenting data will help you make better business decisions. The way you segment your data will depend on your business. For example, if your business is local, you won't need to segment per geographic location but maybe by age etc.

Target
With all this, you should be able to see what is the audience that converts the most. Thanks to this you should be able to refine your target audience and therefore refine your website to optimize for that target audience.

Micro Conversion VS Macro Conversion

There are two types of conversions: micro and macro conversions. The first one corresponds to a step a user is making toward a macro conversion. For example, a macro conversion for an e-commerce will be a product sold. However, there is a lot of steps before this conversion. One micro conversion can be to click on a "add to cart" button.

Conversion attribution

The best way to measure conversions is to attribute value to it. This value will be monetary and will depend on the revenue that you expect to get with this conversion. For example, if someone sign-up for a newsletter, you might attribute 1$ per conversion because you know that on average a user who sign-up to your newsletter brings $1 revenue to your company.
This will then help you calculate your ROI and see how much investment you can allocate on your website (for social media campaign, engineering etc.)

Last click attribution
Now the way you attribute value to conversions can be different depending on your goals. The most popular way is to attribute all the value to the actual conversion. For example, when someone clicked on Sign Up or when someone actually paid for your product etc. This is the most popular because it is the simplest way to think about conversions.

First click attribution
However, you might want to do the other way around to see which micro-conversion converts the most and then you will put all your value in the first click. This is especially powerful when you know that once someone started a first micro-conversion, it is very likely that he will finish the process until the macro-conversion.

Linear click attribution
Finally, you can put the same value for each micro conversions. Usually, you would use it if you don't really have a funnel toward the micro-conversion or if you want to measure how each step of your funnel performs. For example, if your website goal is to inform user, you might want them to click everywhere on your website before convert them (to a newsletter or a contact page). Each page a user clicks will get her closer to the conversion but the order this user clicked on those pages doesn't matter.

Set up Goals

If you want to track your micro and macro conversions, you need to set up goals in Google Analytics. Goals are set up at the view level and they can be of 4 types:

- Destination: Going to a desired page
- Duration: Stay on a page for a certain amount of time

- Pages/ Screens per visit: View a minimum amount of pages
- Event: Watch a video

Each time you set up a goal, don't forget to test it out with Google Analytics to see if you did it right

1 A/B Exercises

---

1. Data landscape of an online store

What kind of data do you expect to have access to as the Google merchandise store owner? More precisely, what fields would you need to monitor the business?

2. Descriptive, predictive, and prescriptive analytics

Give examples of questions addressed by descriptive, predictive, and prescriptive analytics (3 examples per category).

3. SMART goals

1. List the missing SMART ticks in the following examples
    a. I want to save $10,000 a year for the next ten years.
    b. We have to reduce the churn as measured by the number of non-returning users after one month.
    c. We need to double the tires assembly line productivity on a year to year basis.

2. Turn the following goal into a SMART one: "We want to increase the hoodies sales"

4. Structured Pyramid Analysis Plan for the Google merchandise store example

Your turn! Define a SMART goal and design a consistent Structured Pyramid Analysis Plan from it, based on the Google merchandise store example.

Homework: Formulate a SMART goal and build the corresponding analysis plan

1. Choose a company and formulate a SMART goal to improve its business.
2. List the data sources and fields that this company should or may have access to.
3. Build a relevant Structured Pyramid Analysis Plan to reach this SMART goal.

First Google Analytics investigation

You've been hired by google to understand their data. Using the Google merchandise store demo account answer those problems:

1. What type of Audience converts best? (Age, Location, Gender etc.)
2. How could Google improve its acquisitions? (what type of channels works best / worst, should they refine their paid campaigns etc.)
3. According to the data, where should Google improve its site?

Homework: Google Analytics and the Google merchandise store

Use the demo account provided by Google, and answer the following questions:

- During the month of April, name the top 3 days for Page Views.
- Which were the top 3 age brackets by revenue? By Sessions?
- Establish the 3 channels which were most valuable by revenue per user during the month of April.
- Do you find that users who spent more time on the site were more or less likely to make a purchase?
- Based on the available data during the month of April, which countries would you suggest targeting outside of the United States? Why?

Further Reading

1. Halobi's blog, *Descriptive, predictive, and prescriptive analytics*: https://halobi.com/blog/descriptive-predictive-and-prescriptive-analytics-explained/
2. Khan Academy, *Dependent and independent variables*: https://www.khanacademy.org/math/pre-algebra/pre-algebra-equations-expressions/pre-algebra-dependent-independent/a/dependent-and-independent-variables-review
3. The Mind Tools Content Team (2016), *SMART goals*: https://www.mindtools.com/pages/article/smart-goals.htm
4. Ryan Wingate (2018, August 3) *Structured Plan for an Analysis Project (SPAP)*: https://ryanwingate.com/purpose/process/tableau-3/
5. Coursera, *Data Visualization and Communication with Tableau (5 weeks online course)*: https://www.coursera.org/learn/analytics-tableau
6. Berkeley, DS100, *Principles and Techniques of Data Science*: https://www.textbook.ds100.org/ch/01/lifecycle_students_1.html
7. Wikipedia, *The CRISP-DM process model of data mining:* https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining
8. *Google Analytics Academy*. A crystal clear course offered freely by Google. The best source to dig deeper into Google Analytics: https://analytics.google.com/analytics/academy/
9. Himanshu Sharma, *A short introduction to Google Analytics*: https://www.optimizesmart.com/understanding-users-in-google-analytics/
10. *Google URL Builder*. To generate URLs for custom campaigns: https://ga-dev-tools.appspot.com/campaign-url-builder/

Topic 2A

Introduction to Databases and SQL
Part 1

Databases are naturally central to data analytics, and have a rough understanding of why we use databases instead of files, what kind of databases exist, how they work, and how you access it is mandatory for data analytics practitioners.

Databases are so ubiquitous they are hard to define in a few words. Nevertheless, here are three definitions of a database that complete one another

- "*A set of information held in a computer*" Oxford English Dictionary
- "*One or more large structured sets of persistent data, usually associated with software to update and query the data*" Free On-Line Dictionary of Computing
- "*A collection of data arranged for ease and speed of search and retrieval*" Dictionary.com

To us, databases will be a set of data points of any kind stored on a computer with a software to interact with it.

Why use databases instead of file systems?

One may wonder why we need such a software to interact with database in the first place, as we could simply store data in files, and load the files in programs whenever we need it.

Actually, in the early days database applications were built directly on top of file systems, but here are the drawbacks of using file systems to store data:

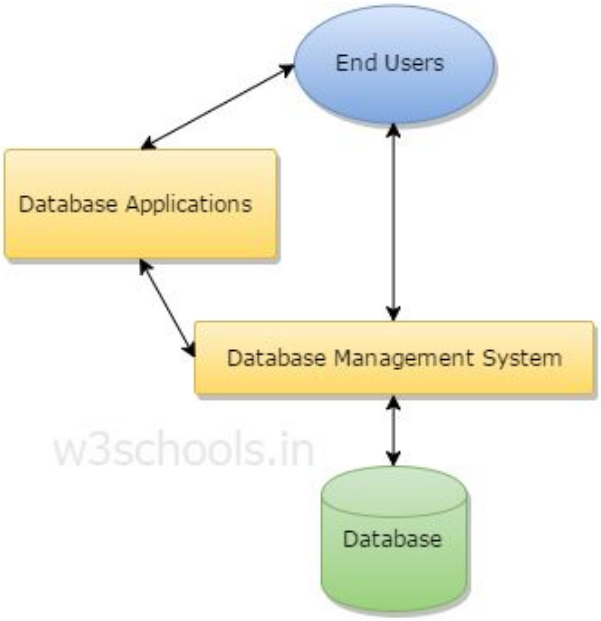
- Data redundancy and inconsistency
- Multiple file formats, duplication of information in different files
- Difficulty in accessing data
- Need to write a new program to carry out each new task

- Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
- Difficulty in accessing data
    - Need to write a new program to carry out each new task
- Data isolation — multiple files and formats
- Integrity problems
    - Integrity constraints (e.g. account balance > 0) become part of program code
    - Hard to add new constraints or change existing ones
- Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - E.g. transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
    - Concurrent accessed needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
        - E.g. two people reading a balance and updating it at the same time
- Security problems
    - Harder to control access over a set of files than for a single entrypoint

Database Management Systems (DBMS)

There is a variety of database management systems that handle the interaction between users, developers, administrators, and the data.

The different databases vary by their use case (operational vs. analytical) and data type (structured/unstructured, time-series, textual data, ...) and volume. What they all have in common is that they all sit between the data end the databases applications (like a website constantly retrieving the products list from the database) or directly an end-user (e.g. an employee of your company analysing the data).



Components of a Database Management System

The components of any DBMS are the following

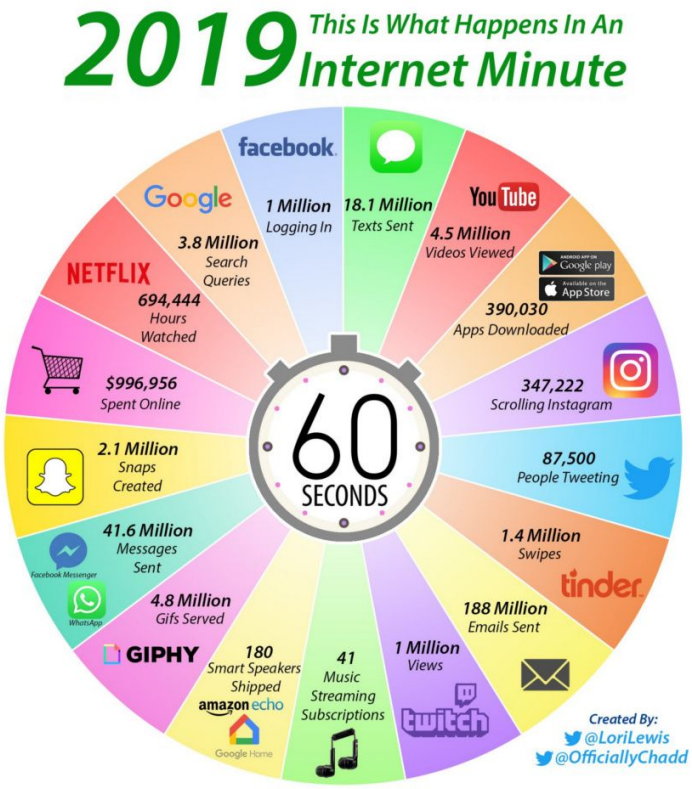
- **Users**: Users may be of any kind such as DB administrator, System developer or database users.
- **Database application**: Database application may be Departmental, Personal, organization's and / or Internal.
- **DBMS**: Software that allows users to create and manipulate database access,
- **Database**: Collection of logical data as a single unit.

To emphasize and elaborate on the role of the database in this workflow, its role is:

- To develop software applications In less time (uniform data access interface).
- Data independence (relative to the filesystem) and efficient use of data (reduced storage, optimized queries with indexes for instance).
- For uniform data administration (same operations for admins even if the data is spread on several machines).
- For data integrity (possibility to enforce constraints on data insert types) and security.
- For concurrent access to data (i.e. ensure the application won't crash when several users access the same data), and data recovery from crashes.

- To use user-friendly declarative query language (e.g. SQL that we will see later on).

Nowadays, in the era of Big Data databases face challenges regarding the large volume and great variety of data, as well as the pace at which it is generated.



The sheer amount of data created on the web in a minute. To ensure fluidity, consistency, and accessibility, databases have to be reliable, flexible in terms of scalability, and specialized for the data at hand.

Those huge numbers are responsible for the development of many different DMBS over the last few years, geared for new use cases (e.g. NoSQL DMBS), or for an unprecedented amount of data inserts and queries (e.g. the Cloudera Hadoop distribution for Big Data).

The Gartner Quadrant of DBMS (2015): A few of the main DBMS. The reasons why there are so many of them are the diversity of data types, use cases, and the speed at which new systems are required to handle unseen volumes of data.

The DBMS ecosystem

This paragraph presents the different DBMS categories and usage. The goal is to be able to understand what DBMS should be used in what use case and why.

Note that most companies have and need several different DBMS, and merging the data from those data sources and ensuring the data integrity is a never ending challenge.

Structured and unstructured data

The first distinction that drives the technological stack choice is the data type.

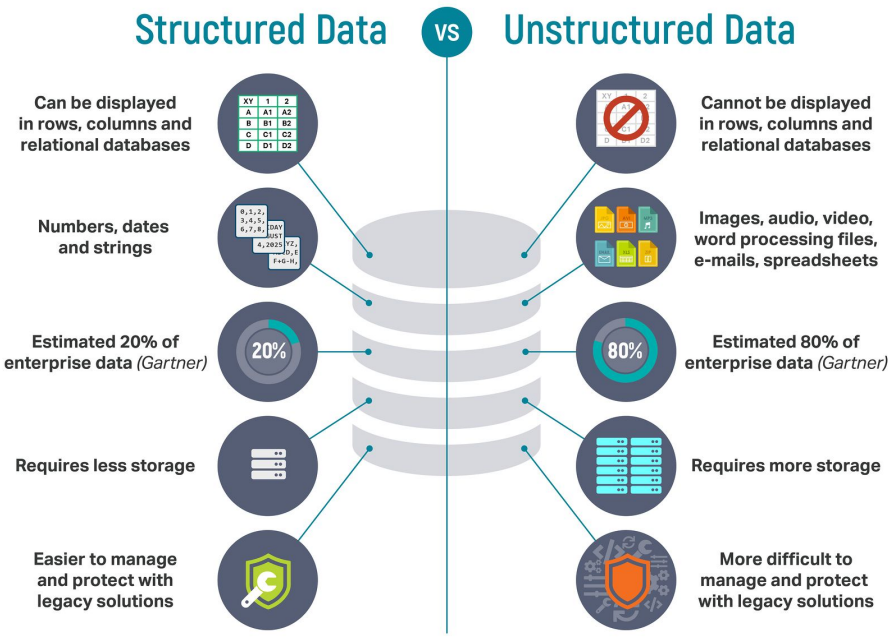
When dealing with data that is well represented by a set of interconnected tables, then the data is structured. For instance, if you run an e-commerce the data for your products and orders is structured.

Products can be stored in one table and orders in an another. The product identifier should be referenced for every order, such that you have a *foreign key* to link the two tables. (A foreign key in one table is a primary key in another table). The data can then be stored in relational database (more on this in a bit), or even in Excel files.

On the other hand, when storing videos or audio data, or the relationships within a social network, the table structure is not a good fit and you are dealing with unstructured data.

In between the two, you may have data that consists of several entries, but each of the entries do not necessarily contain the same fields such that you cannot store it in ables. An example is offered by social media like Twitter: a tweet may be composed of text, links, images, references, hashtags, ... but many of those fields are empty for most tweets so it is not wise to create a table with all the possible fields, because this table would be essentially empty. It would use storage unnecessarily and not be optimized for queries. This kind of data is better stored in XML or JSON files, or in NoSQL databases like MongoDB.

While structured is far more used and mastered overall, the unstructured data represents roughly 80% (see the next figure) of the data stored by companies. Note that the amount of unstructured data is larger for bigger companies, while smaller companies tend to store and use more unstructured data.

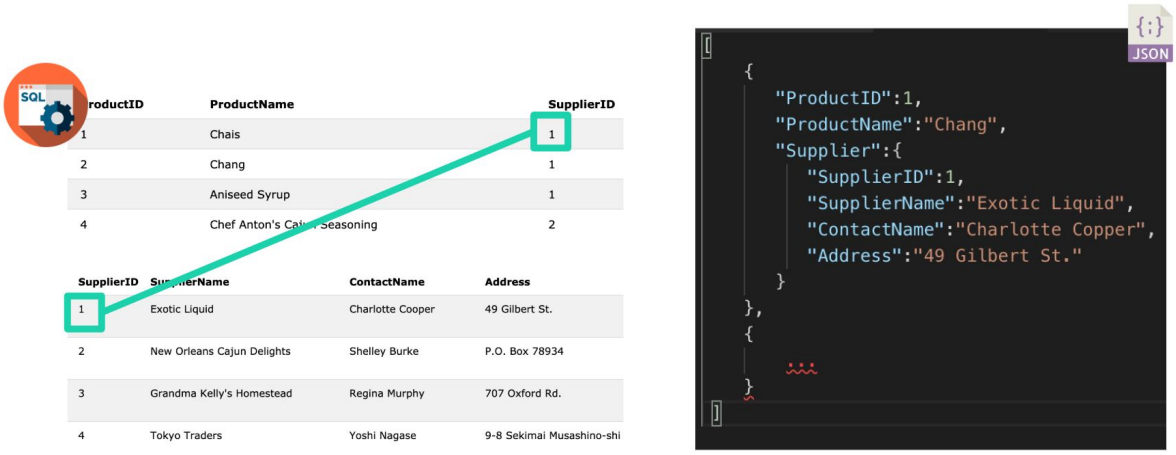


Main differences between structured and unstructured data. From the Lawtomated blog (5/2/2019) (https://lawtomated.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/)

SQL and NoSQL

Structured data can be stored, queried, and managed with the Structured Query Language (SQL) while NoSQL ("Not Only SQL") languages can handle documents with no schema and graph data, or structure-less, bare data. NoSQL databases handle semi-structured data and graph data, but other types of unstructured data like images and videos are typically not managed with DBMS.

The difference between the two is shown in the next figure.

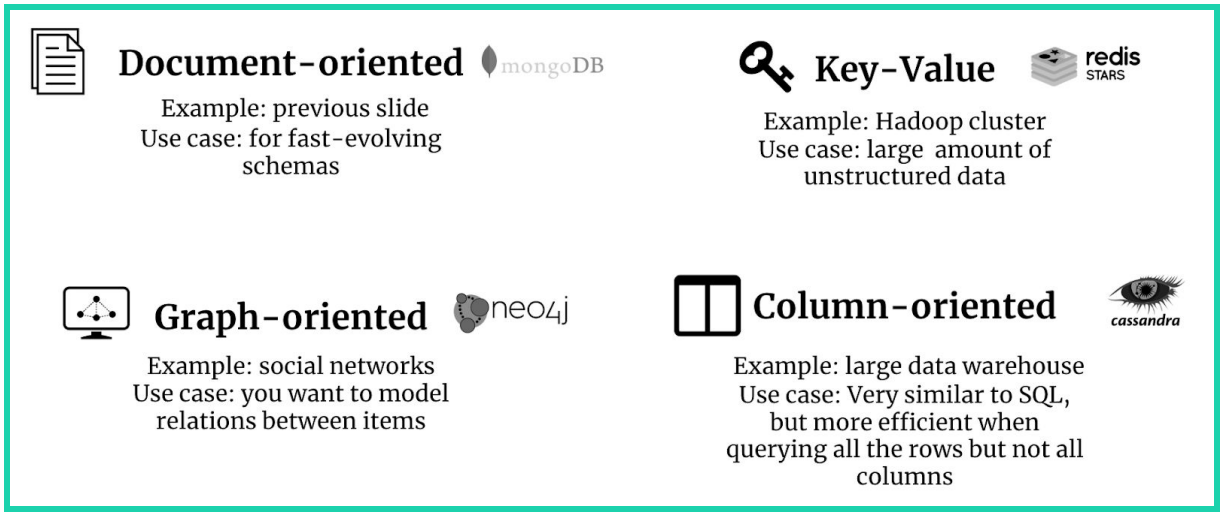


*Representation of SQL-like (i.e. structured) data (left) and document-like data (right), that is typically managed with a NoSQL database like MongoDB. On the left, the data is stored in tables (**products** and **suppliers**) with identifiers that allow to link tables. On the right, the same data is represented as a single list of products, with suppliers embedded as sub-documents. Conversely to SQL database, for the NoSQL database all the elements of the collection do not necessarily have the same fields. It gives more flexibility at the expense of structure.*

In the SQL example **normalization** avoids redundancy. It is easy to update suppliers' info for all products but queries are more complex (involve joins). Conversely, in the NoSQL example **denormalization** allows for simpler queries (all the elements are in the same collection) but update suppliers' information is harder because it involves modifying several products for a single supplier.

Different types of NoSQL databases

There are four main types of NoSQL databases, dealing with semi-structured and unstructured data.



Main NoSQL databases with corresponding tools logos. The previous example showed an example of a document-oriented NoSQL database.

As a final remark, note unless you are a database administrator (DBA) or a developer you don't need to learn how to use those different DMBS. Most likely, you will be able to access data with tools that provide SQL-like interfaces to NoSQL databases. While those interfaces are not suited for production operational usage, it allows to extract, query, and analysis data in a single standard way.

Operational and Analytical databases

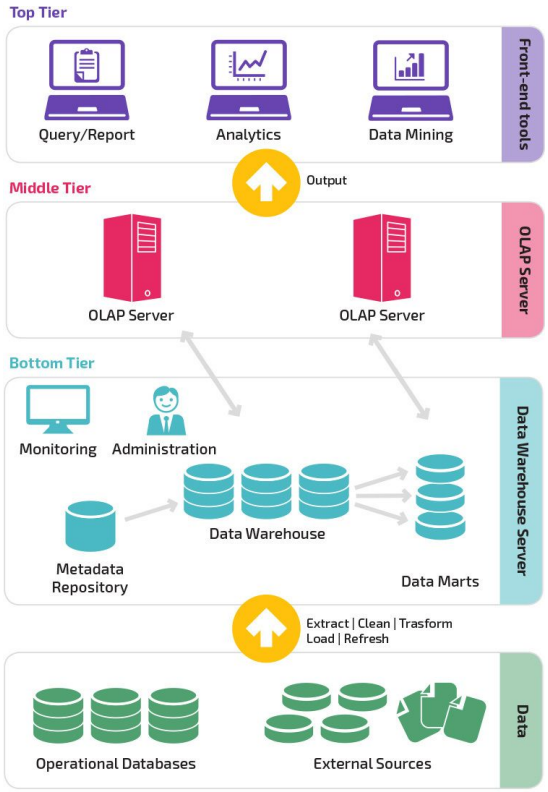
Database design is very different for operations and for analytics. This is why a typical pattern consists of building two different databases on top of the common, production data. One is operational and serves the purpose of the business on a daily basis (typically small queries that have to be fast), and the other one is meant to monitor and analyse your company (more aggregated queries, with lighter speed constraints).

Operational database

- a.k.a. OLTP, for OnLine Transaction Processing
- role = run your business
- optimized for inserts & small lookups
- Normalized: minimize redundancy

Analytical database

- a.k.a. OLAP, for OnLine Analytical Processing
- Role = manage your business
- Optimized for large, unfiltered lookups
- Pre-aggregated: e.g. per region, per item, ...



The three tiers of an analytical database. From the [holistics.io blog](https://www.holistics.io/blog/data-lake-vs-data-warehouse-vs-data-mart/) (5/2/2019). (https://www.holistics.io/blog/data-lake-vs-data-warehouse-vs-data-mart/) Business Analytics and reporting is built on top of an OLAP, or Analytical server that queries data warehouses and data marts (see the next part).

Data organization

Within small companies, data may consist of a single database. In medium-sized of big companies data belongs to different databases and data sources, and the data is organized in several layers of increasing aggregation, and decreasing size and complexity.

Data Lakes

A data lake is the place where you dump all forms of data generated in various parts of your business: structured data feeds, chat logs, emails, images (of invoices, receipts, checks etc.), and videos. The data collection routines does not filter any information out; data related to canceled, returned, and invalidated transactions will also be captured, for instance.

The fate of poorly governed Data Lakes: Data Swamps

Data lakes do not require much structure, and they accept all data. However, in poorly designed and neglected systems, they risk becoming data swamps. A Data Swamp is the term that describes the failure to document the stored data accurately, resulting in the inability to analyze and exploit the data efficiently; the original data may remain, but the data swamp cannot retrieve it without the metadata that gives it context.

Data Warehouses

A data warehouse usually only stores data that's already modeled/structured.

A Data Warehouse is multi-purpose and meant for all different use-cases. It doesn't take into account the nuances of requirements from a specific business unit or function.
As an example, let's take a Finance Department at a company.

They care about a few metrics, such as Profits, Costs, and Revenues to advise management on decisions, and not about others that Marketing & Sales would care about. Even if there are overlaps, the definitions could be different.

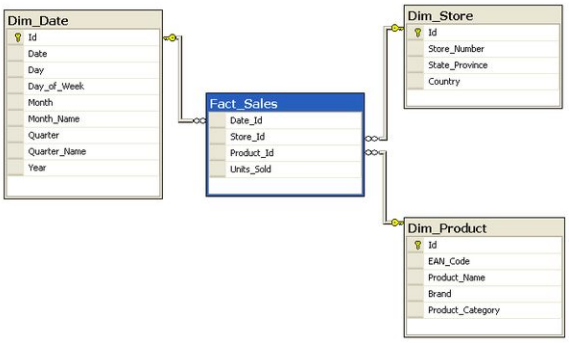
Data Mart

While a data-warehouse is a multi-purpose storage for different use cases, a data-mart is a subsection of the data-warehouse, designed and built specifically for a particular department/business function.

Implementations:

⭐ Star schema: organize your data mart data following a "star-shaped scheme" (picture on the right)

❄ Snowflake schema: advanced version of the star schema for versatile, cross-departments needs



*A data mart structured following the **star schema** database design pattern.*

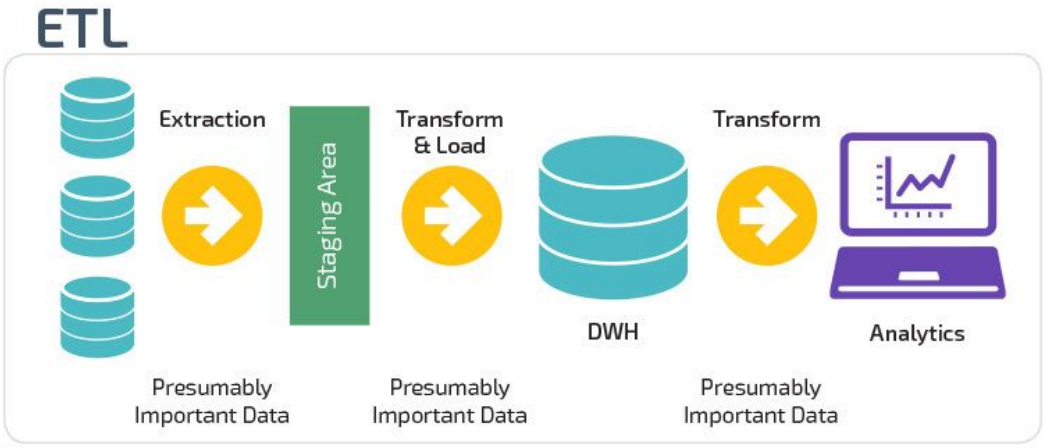
	Most Important Use Group & Use-Cases	Time-to-Market Questions & Solutions	Cost Implementation & Ownership	Users (# & Types)	Data Growth Volume & Variety
Data Lake	Predictive & Advanced Analytics	Weeks - Months	$$$$$		
Data Warehouse	Multi-Purpose Enabler of Operational & Performance Analytics	Hours - Days	$$$$$		
Data Mart	Line of Business Specific Reporting & Analytics	Minutes - Hours	$$$$$		

Differences between data layers. From the holistics.io blog (5/2/2019). (https://www.holistics.io/blog/data-lake-vs-data-warehouse-vs-data-mart)

The Data Lake is the bottom layer, it contains all the data of the company. The Data Warehouse is the entity with which most employees interact, and the data mart is a super fast, simple and aggregated database designed for a few usages only, or a single department. Note that small companies may only have an undifferentiated database.

The glue between data layers: Extract-Transform-Load (ETL) processes

The processes that refine, transform and loads the data in the next layer are referred to as "ETL" processes. There are softwares (like Talend) that handles those steps well, with graphical users interfaces to design new data pipelines, scheduling features, and pre-built data integrity verification features.



The ETL process from operational database or Data Lake to Data Warehouse (DWH) and Analytics. From the Panoply.io blog (5/2/2019)

Relational databases

DBMS that handle structured data are called relational, because it is bound to Codd's relational model, a mathematical theory proposed E.F. Codd et al., 1969, that eventually led to the development of the SQL language in the 90's.

The main component of relational databases are:

- **Table** (a.k.a. relation): main structure of the relational DB, contains rows and columns
- **Schema**: columns (or "attributes") names and types of the table
- **Row** (a.k.a. "tuple" or "record"): a single entry of the table
- **Key**: column with no duplicates that allows to unambiguously identify a row.

In the relational model, tables are related by common keys, in a one-to-one or one-to-many relationship.

OrderID	CustomerID	EmployeeID	OrderDate
10248	90	5	1996-07-04
10249	81	6	1996-07-05
10250	34	4	1996-07-08
10251	84	3	1996-07-08
10252	76	4	1996-07-09

Orders table

CustomerID	CustomerName	ContactName
1	Alfreds Futterkiste	Maria Anders
2	Ana Trujillo Emparedados y helados	Ana Trujillo
3	Antonio Moreno Taquería	Antonio Moreno
4	Around the Horn	Thomas Hardy
5	Berglunds snabbköp	Christina Berglund

Customers table

In the above figure is an example of a one-to-many relationship with a single **key**, where one customer from the *Customers* table on the right may correspond to several orders of the *Orders* table on the left.

The Standard Query Language is the language of choice to define, administer, and query relational databases. In the latter, every attribute must have atomic types:

- Characters: CHAR(20), VARCHAR(50), ...
- Numbers: INT, BIGINT, SMALLINT, FLOAT, ...
- Others: DATETIME, ...

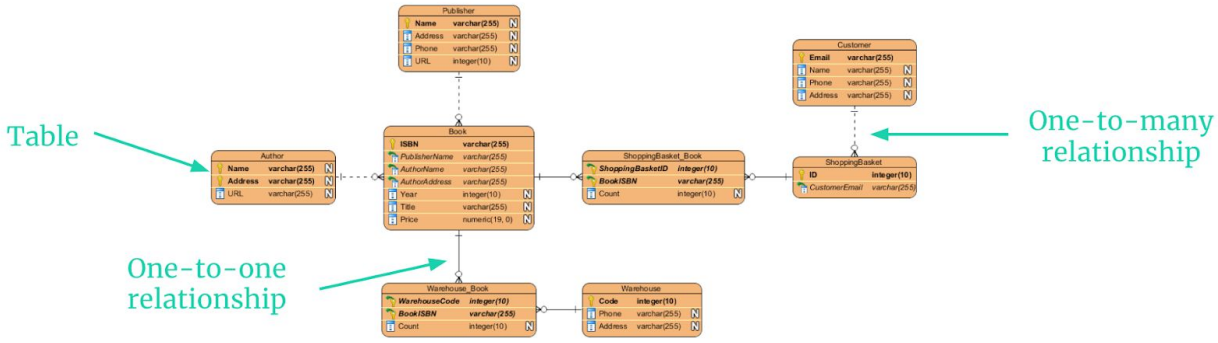
From a SQL development environment or terminal, tables are created with the following syntax:

```
CREATE TABLE Students (
  sid CHAR(20),
  gpa  FLOAT,
  birthday DATETIME,
  PRIMARY KEY (sid)
)
```

Example: Table creation query in SQL. Note the contrast with NoSQL databases: the schema must be defined beforehand.

Representing relational databases with Entity Relationship (ER) Diagrams

A database often contains dozens of tables, such that it is hard to visualize all tables at once. Fortunately, you can display the tables, their schema, and the relationships between tables with ER diagrams. Here is an example of such a diagram for a book publisher database.

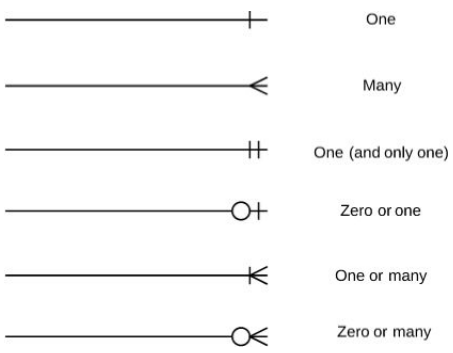


The ER diagram of a book publisher database. Picture from Visual Paradigm (5/9/2019)

(https://www.visual-paradigm.com/solution/free-erd-tool/)

ER diagrams also specify the type of relationship between tables, i.e. whether tables are in a one-to-one relationship (one row of the first table corresponds to on row in the second one), one-to-many (one row of the first table may correspond to several rows in the second one), ...

The different cases and corresponding symbols are depicted in the figure below.



ER diagrams symbols convention. Picture from LucidChart (5/9/2019)
(https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning)

2A Exercises

1. The social network database

You work for an online social network.

1. Cite three reasons why you should store your data in databases.
2. What kind of structured data are you expecting? (check all that apply)
    a. Postings content, comments, and metadata (posting date, location, ...)
    b. Users pictures and uploaded videos
    c. Users general information (name, age, gender, ...)
    d. Relationships (the "network graph")
    e. Browsing data (web logs): log in time, clicks, scrolling, ...
3. Same for unstructured data (with the same proposed answers).
4. What DBMS would you use for the different data types mentioned in question 2?

2. Course database design

Design a relational database for our class.

Think about what kind of data you would like to store about students, their general information, progress, grades, completed exercises. Data about the content and exercises should also be stored in this database.

Choose a relevant set of tables for this database, and describe all their fields with the corresponding types.

Homework: Draw an entity relationship diagram

Draw the entity relationship diagram of the W3school database.
(https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_columns)
You can see the tables on the r.h.s. and explore their content by clicking on them.

Further Reading

1. *W3schools tutorial on databases management systems:* https://www.w3schools.in/dbms
2. Short slide introduction to databases: http://geowww.geo.tcu.edu/faculty/50901/database.pdf
3. Jatin Raisinghani (2018), *Data Lake vs Data Warehouse vs Data Mart*: https://www.holistics.io/blog/data-lake-vs-data-warehouse-vs-data-mart/
4. Geoffrey Craig (2016), *What Is The Difference Between Data Lakes, Data Marts, Data Swamps, And Data Cubes?:* https://intersog.com/blog/what-is-the-difference-between-data-lakes-data-marts-data-swamps-and-data-cubes/
5. Wikipedia, *Snowflake schema:* https://en.wikipedia.org/wiki/Snowflake_schema
6. Wikipedia, *Star schema:* https://en.wikipedia.org/wiki/Star_schema
7. Panoply.io blog, on data warehouses and ETL, *Data warehouse guide:* https://panoply.io/data-warehouse-guide/data-warehouse-architecture-traditional-vs-cloud/
8. Janik Von Rotz *SQL Cheatsheet*: https://gist.github.com/janikvonrotz/6e27788f662fcdbba3fb
9. Stanford (2018), *CS 145: Data Management and Data Systems*: https://cs145-fa18.github.io/
10. Lucid Chart (2019), *Entity relationship diagrams:* https://www.lucidchart.com/pages/er-diagrams

Topic 2 B

Introduction to Databases and SQL Part 2

The Structured Query Language (SQL) is an unavoidable element any data practitioner's toolbox. Here are its main characteristics:

- SQL is by far the most widespread language to query and administrate relational databases (📣 pronounced "SQL" or "Sequel", experts cannot decide).
- It is a **declarative programming language** (i.e. like HTML, ≠ imperative programming languages like Java or Python). In short it means it contains no loops, conditions, or explicit instructions like those.
- We **usually use extensions** of SQL like MySQL, Oracle SQL, Transact-SQL or PostgreSQL. For basic functions they are very similar to one another.
- The DataBase Management System (DBMS) can be hosted on your laptop (for development) or on a server.
- Interaction with the DBMS usually via an Integrated Development Environment (IDE) or in a terminal or in another programming language via a "connector".

First SQL statements: creation, edition and deletion

The SQL statements can be divided into four categories:

- **DDL**: Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.
- **DML**: Data Manipulation Language which deals with data manipulation, and includes most common SQL statements such as SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in database.
- **DCL**: Data Control Language, which includes commands such as GRANT, and mostly concerns with rights, permissions and other controls of the database system.

- **TCL:** Transaction Control Language which deals with the transaction within the database, i.e. save points and commits.

Admins can create databases and print existing ones with CREATE DATABASE and SHOW DATABASE. The following figures gives examples of database and table creation, plus data insertion.

```
CREATE DATABASE course;
USE course;
SHOW tables;
```

```
CREATE TABLE students (
 sid CHAR(20) PRIMARY KEY,
 gpa  FLOAT,
 birthday DATETIME,
 PRIMARY KEY (sid)
)
```

```
INSERT INTO students
VALUES
   (1,3.4,'1995-03-12 12:09'),
   (2,3.7,'1997-04-19 15:27'),
   (3,2.6,'1996-12-04 02:37');
```

SQL queries to create a database (left), create a table (middle), and insert rows in the newly created table (right).

Most of the administration tasks can be done either

- with the Command Line Interface (CLI) in a terminal, an IDE, or a programming language interface (API)
- or with the IDEs predefined functions (click)

💡**By convention SQL keywords are written in upper case**💡

After creation, tables schemas and values can be updated, and databases, tables, columns, constraints, and indexes can be deleted.

```
UPDATE students SET gpa=0. WHERE id=1;
```

```
ALTER TABLE students  ADD COLUMN name char(30);
```

```
ALTER TABLE students  DROP COLUMN name;
DROP TABLE students;
DROP DATABASE course;
```

Table schemas and values updates, and column, table and database deletion.

In SQL the database is relational, the schema has to be defined in advance. The main types are shown below. SQL also allows to define constraints to forbid empty values in certain columns, or forbid duplicates.

Main data types

- CHAR(size): fixed-length string
- TEXT: long strings (up to ~65K characters)
- SMALLINT(size): integers from -32768 to 32767.
- INT(size): integer from ~-2B to ~+2B
- UNSIGNED INT(size): integer from 0 to ~4B
- FLOAT(size,d): decimal number
- DATETIME(): date and time

See an exhaustive list in this nice SQL cheatsheet (https://gist.github.com/janikvonrotz/6e27788f662fcdbba3fb)

Table schema constraints

Constraints can be defined at table creation like in the example above, or after using the ALTER keyword.

- NOT NULL: forbid "NULL" or "empty" cells in the corresponding columns.
- UNIQUE: the column cannot contain duplicates.
- PRIMARY KEY: a combination of NOT NULL and UNIQUE. Uniquely identifies each row in a table.

Note on indexes

Indexes are used to retrieve data from the database very fast. The users cannot see the indexes, they are just used to speed up searches/queries. We will not cover them in this class but the interested reader can learn more in this TutorialsPoint tutorial.(https://www.tutorialspoint.com/sql/sql-indexes.htm)

Database administration: the Data Control Language

One advantages of DBMS is the possibility to grant and revoke rights to users in a very easy and controlled way.

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE TO bobby;
```

```
REVOKE INSERT, DELETE, UPDATE TO bobby;
```

Grant and revoke rights to a user named bobby.

📝In practice, databases administration is handled by DataBase Administrators (DBA) and value insertion is automated

Basic SQL queries

We will use the W3Schools database (https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_where) as a running example. W3schools provides SQL tutorials and a playground in which you can play with a toy database without installing anything.

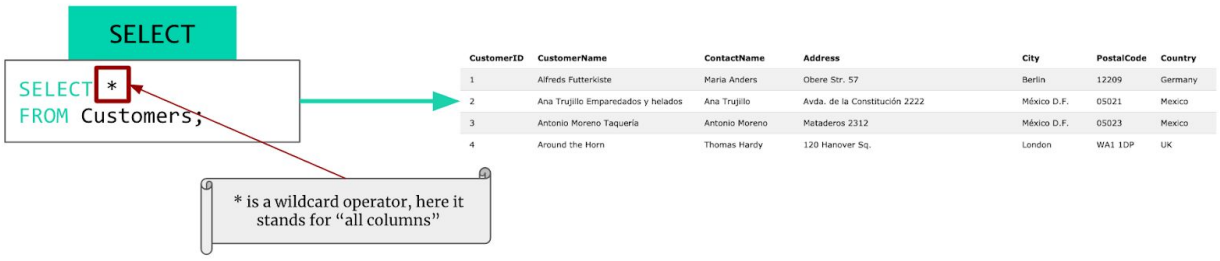
The database has a typical structure and contains 8 tables related by their keys (IDs): Customers, Categories, Employees, OrderDetails, Orders, Products, Shippers, Suppliers.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany

*The first 6 rows of the **Customers** table of our example database*

Selection (SELECT) and aliases (AS)

When you want to retrieve data from a database, either to print it, pass the result to another program, or create a new table with the output, you perform a query. All queries use the SELECT to choose the columns you want to output, with the FROM keyword to choose the table from where you want to retrieve those columns.

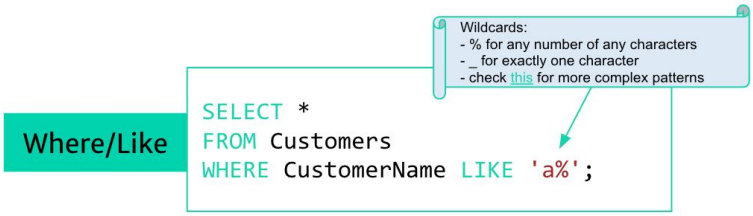


When querying data you can change the names of columns with the AS keyword.

Filtering (WHERE) and distinct values (DISTINCT)

We have seen that the output columns are specified after the SELECT keyword. The WHERE keyword allows to select specific rows. Any valid condition can be used to select columns. Here is an example with an equality test.

And another example, more advanced, involving pattern matching on strings.



Conditions in SQL

- Equality: Column = 3.2
- Difference: Column != 'Apple'
- Comparison: Column > 5
- Set membership: Column IN ('Apple', 'Pear', 'Orange')
- Fuzzy string matching: see "Like" example on the right
- Conditions can be combined with the AND, OR, and NOT keywords

Finally, if you want to retrieve unique values from a column you can use the DISTINCT keyword like so.

Distinct

```
SELECT DISTINCT Country
FROM Customers;
```

Sorting (ORDER BY)

Sorting is a basic operation in SQL that we often perform. Tables can be sorted according to one or several column, in increasing or decreasing order, and sorting on strings (alphabetically) and datetime objects is allowed.

Order by

```
SELECT *
FROM Products
ORDER BY Price DESC;
```

Sorting can be ascending (default, keyword ASC), or descending (keyword DESC)

When sorting by several columns, the sorting direction must be separately specified for each column.

Order by several columns

```
SELECT *
FROM Customers
ORDER BY Country ASC, CustomerName DESC;
```

In this example, the rows are sorted by countries alphabetically, and rows belonging to the same country are sorted by "anti-alphabetical" customer name order.

If you want to limit the output to the first N examples (N=10 in the example below) you can use the LIMIT keyword in MySQL, ROWNUM in Oracle SQL, and TOP with SQL Server. Beware that retrieving a top N only makes sense after sorting with ORDER BY.

Get a top N

```
SELECT ProductName
FROM Products
ORDER BY Price DESC
LIMIT 10;
```

⚠*Top 10 retrieval example. The syntax of this operation varies from one SQL distribution to the other: Oracle SQL uses ROWNUM, and SQL Server uses TOP*

2B Exercises

---

1. Select and table creation

We will use the W3Schools playground for these questions:

- URL: https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_where
- or Google "w3schools sql playground"

1. SELECT and AS
    a. Write a query that outputs all the rows and columns of the "Orders" table
    b. Output all the rows of the the "ProductName", "Unit" and "Price" columns of the "Products" table
    c. Same question but rename the "Unit" column as "Dimension" ,and the "Price" column as "Cost"
2. CREATE TABLE
    a. Create a new table named "students" that contains an id, and the names and ages of your classmates
    b. Print all the rows and columns of this new table

2. Select with filters

1. From the W3schools Customers table, retrieve all the columns corresponding to customers from Sweden.
2. Same, but only output the names and city of the Swedish customers.
3. Same, but filtering out customers with names starting with a "B"

3. Sorting in SQL

1. Order employees from the youngest to the oldest, and alphabetically (by LastName) for employees born on the same day. Print all the columns.
2. Find the 5 latest orders from the "Orders" table
3. Find the Products from the supplier with SupplierID 8, and sort them by increasing Price

Homework: W3Schools first exercises

For this part, the first 14 exercise of the W3Schools SQL exercises (https://www.w3schools.com/sql/exercise.asp) (from the beginning to "SQL Null" included) are great, with solutions. Try not to look at the solutions before trying to find the answers by yourself.

Homework: Multiple conditions query

From the W3School playground, select the ContactName (renamed "Contact") and City of all customers from the USA, with ContactName containing a "J".

Further Reading

1. *W3schools tutorial on databases management systems:* https://www.w3schools.in/dbms
2. *SQL tutorial on TutorialsPoint: https://www.tutorialspoint.com/sql/*
3. *(advanced) Introduction to Databases (Stanford online courses): https://lagunita.stanford.edu/courses/Engineering/db/2014_1/about*
4. *SQL cheatsheet (janikvonrotz's GitHub): https://gist.github.com/janikvonrotz/6e27788f662fcdbba3fb*

Topic 3A/B

Advanced SQL - Part 1 & 2

To get summaries of the data aggregate functions are needed. Aggregate functions return less rows than their input.

The COUNT() function returns the number of rows that matches a specified criteria, combined with the DISTINCT keyword it gives the number of distinct values in a column.

Count the rows of a table

```
SELECT COUNT(*)
FROM Customers;
```

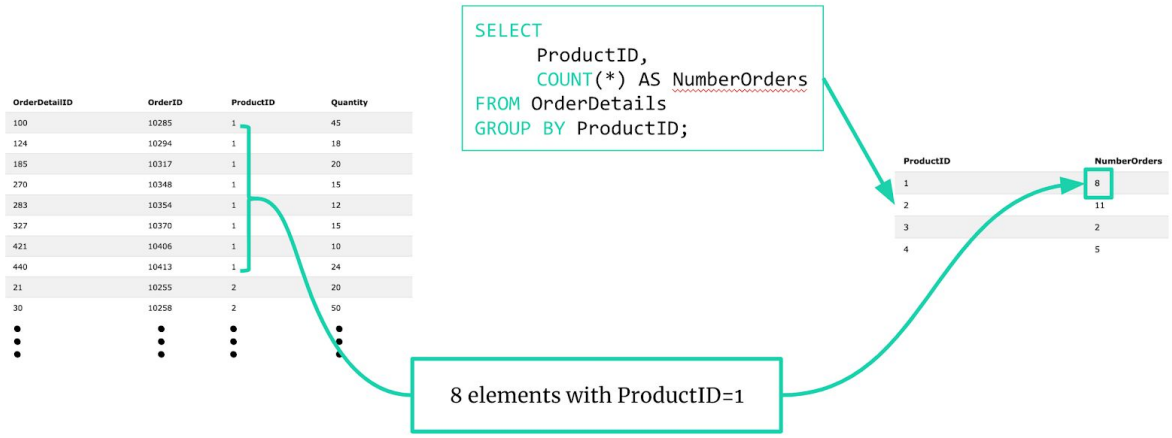
Count unique values in a column

```
SELECT COUNT(DISTINCT Country)
FROM Customers;
```

SQL supports 5 aggregate functions by default: COUNT, SUM, MIN, MAX, and AVG.

Compute group aggregates (GROUP BY)

The SQL GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

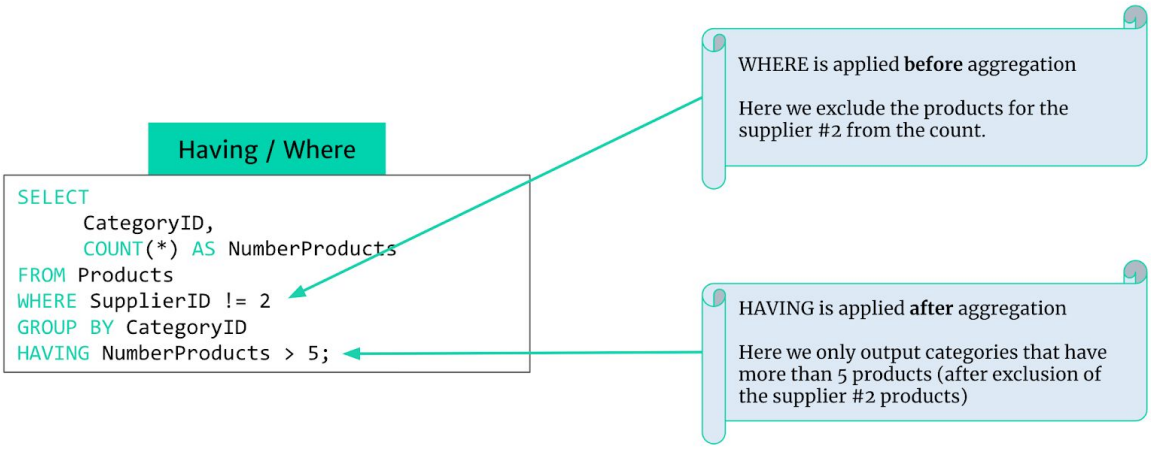


The GROUP BY operation illustrated. All the rows with the same ProductIDs are gathered and aggregated using the COUNT aggregate function. For example for the elements with ProductID=1 the count outputs 8.

GROUP BY with filters: WHERE and HAVING

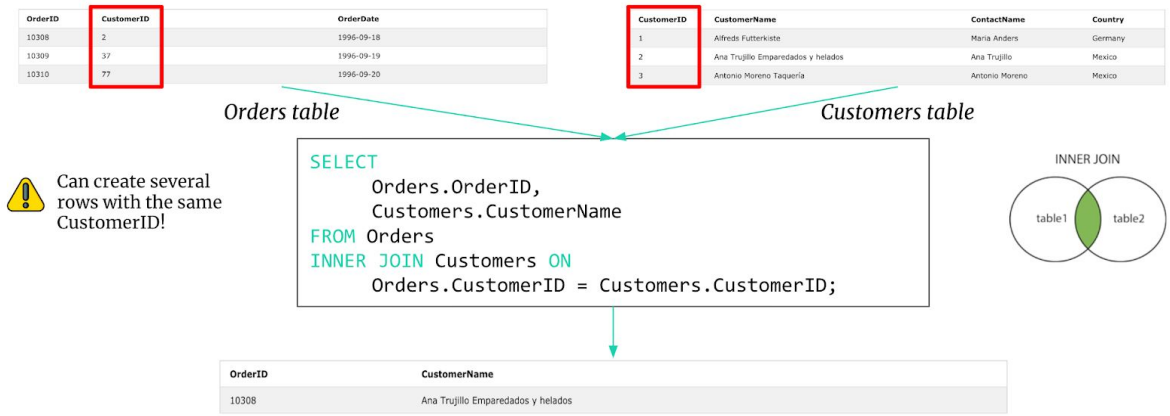
When using a GROUP BY statement and filtering data, one has to be careful as filtering the data before aggregation is often not equivalent to filtering after aggregation.

The WHERE statement is applied before aggregation, while the new HAVING statement is applied after aggregation.



Merge columns from several tables with JOIN

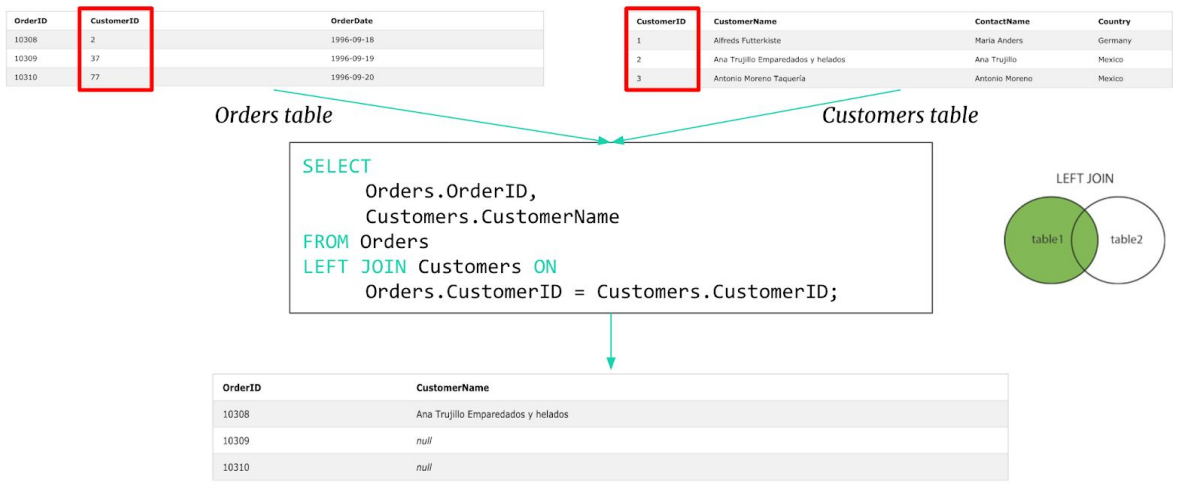
The JOIN operation is very common in SQL. Typically, in most analysis and queries fields from different tables are needed together. To merge data from different table a foreign key is needed, and is used to perform the JOIN operation.



(Inner) join between a Customers table and an orders one. The join operation allows to get the customer name associated to an order, while this field was not available in the Orders table originally.

Note that a join operation does not necessarily preserve the number of columns of the left nor the right table being joined.

While the INNER JOIN shown above only output rows for which there is an entry both in the left and the right table involved in the merge, users may want to keep all rows from one table (e.g. the left one), and add the other table's field only when the foreign key matches. This operation is called a LEFT JOIN when we keep all values from the left table (the RIGHT JOIN exists, but by convention we tend to rather use only LEFT JOIN and switch the tables positions in the query if necessary).



LEFT JOIN example. All the rows of the "left" table (i.e. the one appearing before the LEFT JOIN keywords, Orders here) are kept, even if there is no matching CustomerID in the "right" table. Missing fields from the right tables are replaced with NULL values.

It is very common to need to merge fields from more than two tables. In this situation multiple join can be performed. INNER, LEFT, RIGHT, and OUTER joins can be combined freely (an OUTER join is a JOIN in which the rows of both the left and the right are kept regardless of the foreign key matching, it is not very commonly used).

Rows concatenation: UNION

JOIN merges columns from different tables.

To concatenate rows from several tables with the same schema, use UNION (deduplicated) or UNION ALL (keeps duplicates).

Get the names of all your customers and suppliers

```
SELECT CustomerName AS name FROM Customers
UNION
SELECT SupplierName AS name FROM Suppliers;
```

UNION example. Her we concatenate the names of customers and suppliers in a single column. Here if a name appears in both the Customers and the Suppliers table it will only be returned once, while it would be returned as two rows with UNION ALL.

Not that UNION is much less common than JOIN, because users typically want to filter and/or aggregate rows, and mix columns from several tables. Those operations do not involve UNIONs.

What we do not cover about SQL

While you are now able to handle most of the queries you may need in the future, there are a few concepts that we do not cover because they are time-consuming, specific to the database administrator's role, or rarely useful:

- **Windowing functions**: Perform calculations on several rows at the same time
- **Indexes**: adding indexes to your table can make some queries infinitely faster (most of the time a few indexes already exist)
- **Procedures** (stored "functions"), **Views** (intermediate queries), **Recursive queries**, ...

You can find more about those topics in the very good Stanford Data Management and Data systems course.
(https://cs145-fa18.github.io/)

3 A/B Exercises

1. HackerRank challenges

HackerRank (https://www.hackerrank.com/) is a coding challenge website. It has a SQL category. Create an account and solve the following challenges:

1. Japanese cities names
   (https://www.hackerrank.com/challenges/japanese-cities-name/problem)
2. Name of employees
   (https://www.hackerrank.com/challenges/name-of-employees/problem)
3. More than 75 marks
   (https://www.hackerrank.com/challenges/more-than-75-marks/problem)
4. Type of Triangle
   (https://www.hackerrank.com/challenges/what-type-of-triangle/problem)
5. Aggregations
   (https://www.hackerrank.com/challenges/revising-aggregations-the-count-function/problem)
6. The blunder
   (https://www.hackerrank.com/challenges/the-blunder/problem)
7. African cities
   (https://www.hackerrank.com/challenges/african-cities/problem)
8. Average population
   (https://www.hackerrank.com/challenges/average-population-of-each-continent/problem)
9. Customers in each country
   (https://www.w3schools.com/sql/exercise.asp?filename=exercise_groupby2)
10. Placements
    (https://www.hackerrank.com/challenges/placements/problem)

Homework: More HackerRank challenges!

1. Finish the HackerRank exercises above
2. Do 5 additional exercises of your choice on HackerRank

Further Reading

1. *SQL Joins explained: http://www.sql-join.com/*
2. *Joins in steps (Brian Zindler, 2019): http://www.zindlerb.com/joins-in-steps/*
3. *Stanford Data Management and Data systems course: https://cs145-fa18.github.io/*
4. *Window functions: SQL PARTITION BY (on SQLTutorial): http://www.sqltutorial.org/sql-window-functions/sql-partition-by/*
5. *Coursera SQL for Data Science: https://www.coursera.org/learn/sql-for-data-science*

Week 4A

Course Review

Week 4B

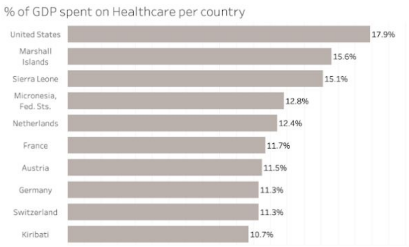
Data Visualization

Data visualization generalities

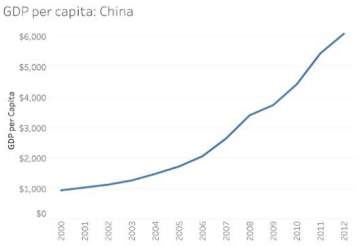
This paragraph is an introduction to statistical graphs. We'll show the main types of graphs that you may use to visualize and communicate a data analysis, and we'll discuss which graph is best in which situation.

Basic graphs

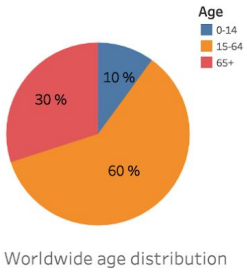
When plotting a quantity it is very easy to present it in a way that is biased and create a misleading graphic. We'll go through a series of common pitfalls to be aware of when creating a plot.



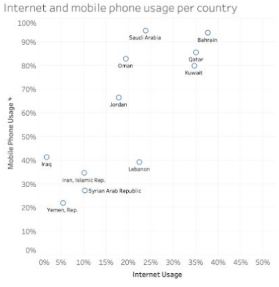
Bar chart



Line chart



Pie chart



Scatter plot

The most basic types of graphs. You should try to stick to these 4 types as much as possible.

In the figure above are the 4 main types of graphs that you will see in scientific works, business presentations, and in any quantitative analysis. You should use those 4 simple graphs most of time, and only consider fancier one in peculiar situations that we'll discuss later.

The first one in the top left corner is a bar chart displaying the % of the growth domestic product that is spent in healthcare for ten different countries. The **bar chart should be your default choice in most situations where you have to compare quantities between different categories** (here, the countries).

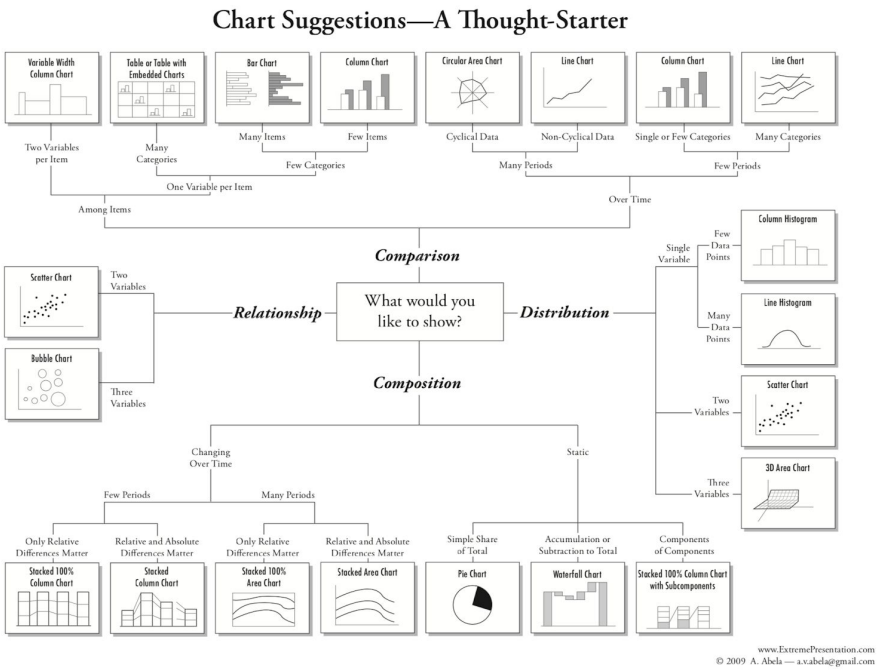
In the top right corner you see a line chart of the GDP per capita of China year per year. The **line chart is useful when you have to plot a trend over time**, or in any situation where the x-axis as a natural ordering.

In the bottom left corner you can see a pie chart of the (fake) worldwide age distribution. The **pie chart is often overused and can be misleading. You should only use it when you study a distribution over a few categories**. Because, if there are too many categories you can't see anything, and if you're not plotting a distribution (that should sum to 100 %) that means you can use a bar chart instead.

The last graph in the bottom right corner is a scatter plot of the % of citizens using internet vs the % of citizens using a mobile phone in different middle east countries. The **scatter plot is useful to investigate and highlight correlations** between variables. Here, for instance, we can see that the internet usage is (not so surprisingly) correlated with the mobile phone usage.

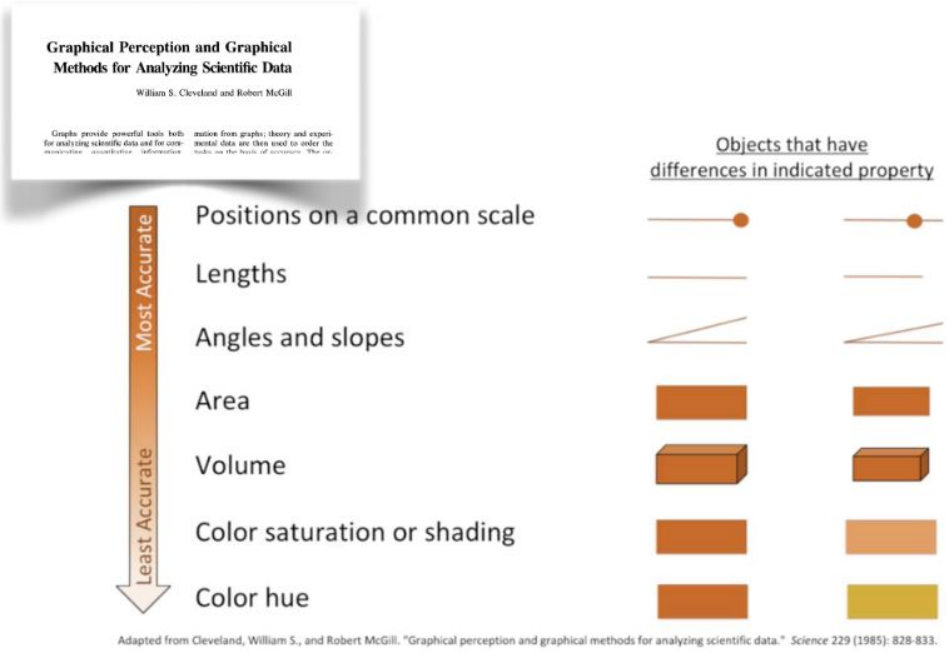
The comprehensive graphs landscape

Although you should use basic charts whenever it is possible, in some situations you may want to use more specific ones, especially when you are exploring data or presenting to an expert audience. The figure below shows gives a decision tree of the visualization one should use depending on the use case.



[Choosing a good chart](https://extremepresentation.typepad.com/blog/2006/09/choosing_a_good.html)
(https://extremepresentation.typepad.com/blog/2006/09/choosing_a_good.html)
(ExtremePresentation, A. Abela). A more comprehensive list of available plots, by use case.

When choosing a data representation, it is good to be aware of the findings shown in the next figure, unveiled by researchers in psychology in the 80's.

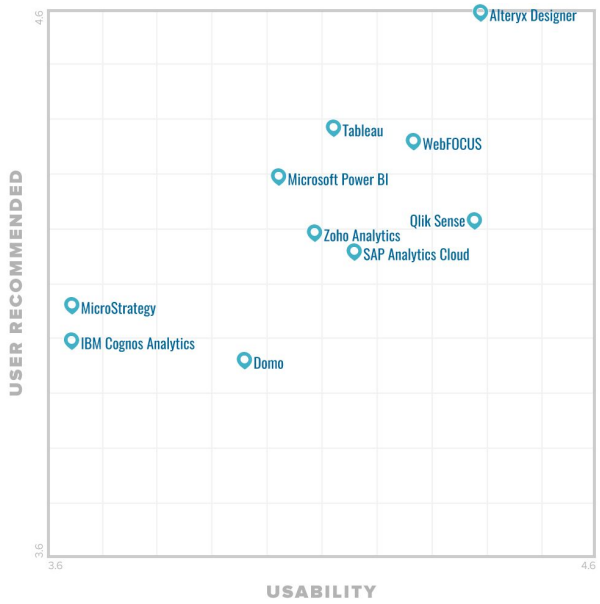


Accuracy of human perception by quantities representations. The top representation is the most accurately apprehended by humans, while the bottom ones are the most misleading. The bottom line is: stick to simple shapes when plotting. From Graphical Perception and Graphical Methods for Analyzing Scientific Data (Cleveland & McGill) (https://science.sciencemag.org/content/229/4716/828)

BI and Data Visualization platforms

There are many software solutions for BI and data visualization. While some of the corresponding tools serve different purposes, they are quite similar to one another, such that the knowledge gained on one of them is easily transferred to the others.

While Tableau Software and Microsoft Power BI still dominate the BI tools market, challengers are becoming more and more relevant.

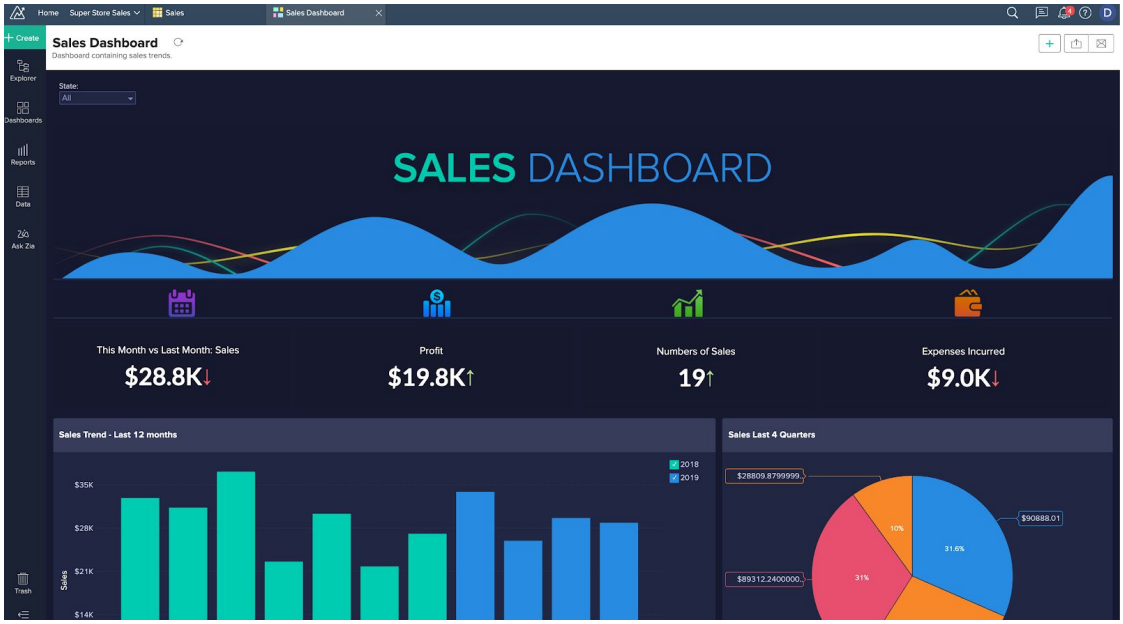


Top BI tools from Software Advice (5/5/2019) (https://www.softwareadvice.com/bi/#top-products)

Zoho Analytics

This course focuses on Zoho Analytics (https://www.zoho.com/analytics/) for the following reasons:

- Growing Business Intelligence (BI) Tool
- Used for Data Visualization
- Easy to use with Drag and Drop
- Free for up to two users on the same workspace
- Fully online



Zoho Analytics preview. Zoho runs fully in your browser.

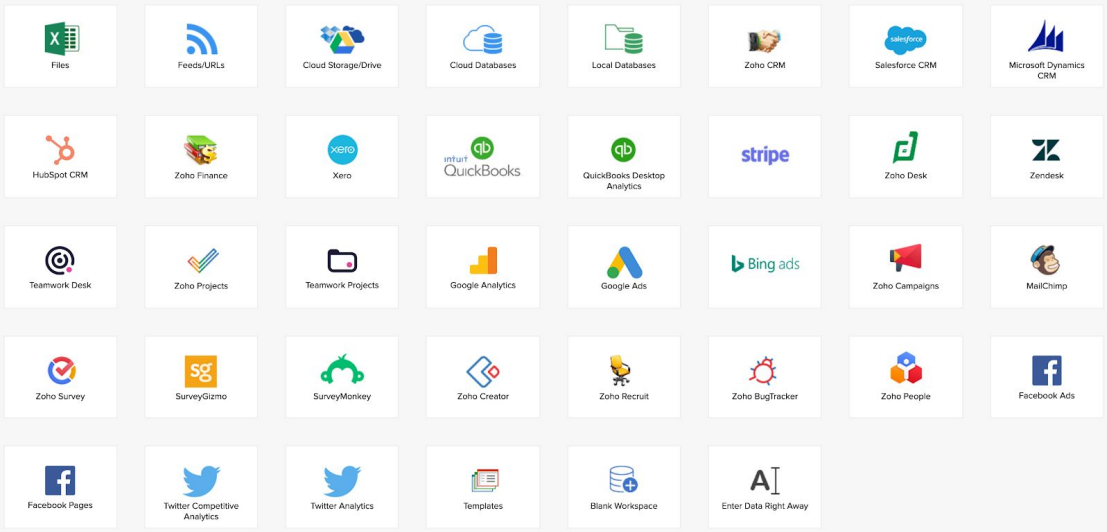
Zoho Analytics can integrate imported files:

- Imported files
- Excel
- CSV
- JSON
- ...

Or connect to a remote data source

- Connect to a remote data source
- Customer-relationship management (CRM) softwares (Salesforce, Zoho CRM, ...)
- Cloud databases (Amazon Redshift, Google Cloud, ...)
- Server databases (MySQL, SQL Server, ...)
- Google Analytics
- Cloud Storage (Dropbox, Google Drive, ...)
- ...

An exhaustive list of the allowed data sources is given by the next figure.



Available data sources in Zoho Analytics. Data can be imported from excel or csv files on your laptop, or from external data sources and databases.As you can see, you can even integrate data from social networks API, or build custom connections with "Blank Workspace". In short it is unlikely to handle a data source that cannot work nicely with Zoho Analytics. This is the case for most BI solutions.

4B Exercises

Homework: Home-price indices analysis

Use the "Case-Shiller home-price indices" example data available in Zoho Analytics.

1. Draw the Entity Relation diagram of the underlying relational database.
2. Create a dashboard with the evolution of prices in San Francisco.
3. Add a filter to select displayed years.

Further Reading

1. *A. Abela (2006), Choosing a good chart:* *https://extremepresentation.typepad.com/blog/2006/09/choosing_a_good.html*
2. *W. Cleveland & R. McGill (1985), From Graphical Perception and Graphical Methods for Analyzing Scientific Data:* *https://science.sciencemag.org/content/229/4716/828*
3. *Software advice, Top BI products:* *https://www.softwareadvice.com/bi/#top-products*
4. *Get Started with Zoho Analytics:* *https://www.zoho.com/analytics/help/getting-started/*
5. *Zoho Analytics Learning center:* *https://www.zoho.com/analytics/video-demo.html*
6. *Coursera: Data Visualization and Communication with Tableau:* *https://www.coursera.org/learn/analytics-tableau*
7. *SerialMentor, 2019, Data Visualization:* https://serialmentor.com/dataviz/

Week 5A

Statistical Thinking

What you will learn in this class

1. Define Statistics and different data types
2. How data is collected
3. How to create representative samples
4. Measure Center with Mean, Median, Mode
5. Measure Variation with Range, Standard Deviation, Z-Score

Statistics and Data Types

Definition of Statistics

As defined in the Merriam-Webster dictionary, "Statistics is a branch of mathematics dealing with the collection, analysis, interpretation, presentation, and organization of data."

Data Types

Quantitative Data

When we think about data, we generally think about numbers, like age or height. These are referred to as numerical or quantitative data.

Discrete Data

Quantitative data can take specific values. For instance, you can have 14 people in a given class or 15, but not 14.5. Other types of data can only take particular values, like shoe sizes, which can be 7, 7.5, or 8, but usually not any numbers in between. These are called discrete values

Continuous Data

On the other hand, Quantitative data can be continuous. This means the data can take any value within a reasonable range, and its precision is only limited by the fidelity of our measuring devices.

Qualitative Data

Not all data is quantitative. Some things, like race or gender, can be defined as categories, but generally cannot be assigned a numerical value. These are called qualitative or categorical values.

Ordinal Data

Qualitative Data can be ordinal, meaning that there is a clear scale from low to high or from bad to good. One such example is grades, where A is better than B, and B is better than C. Another example is customer feedback ratings like "fair," "good," and "excellent."

Nominal Data

Other types of labels are simply descriptive categories, like the country of birth or eye color, which defy any attempt to find a generally agreed upon order. These are called nominal.

Sampling and Collecting Data

What is sampling

It is also important to ask the question of whether our data should represent the entire population or can be just a subset of it. If we want to learn about everybody in the United States, we need to conduct a census. However, doing so is so expensive that even one of the richest countries in the world can only afford to do it just once every 10 years. It's also a very time consuming and complicated process. In business, censuses are generally impractical because usually we need our results obtained at a low cost and as fast as possible.

Parameters

When we talk about populations, we refer to any descriptor of it as a parameter. For instance, population parameters might be the proportion of men vs. women, or the average age, or the unemployment rate.

In many situations, we don't have either the time or the resources to collect data about every single person, which is why we most often deal with data samples. Samples are sub-collections of members selected from the population. They can be described using the same metrics, like proportions or averages or rates. However, when we talk about samples, we describe them using statistics, which is yet another use of the same word.

To reiterate, when referring to the entire population, we talk about population parameters. Conversely, when dealing with samples, we describe the data points as statistics.

Collect Data

Experimental VS Observational Studies

There are two fundamentally different ways of collecting data. We can simply observe the world without any attempt to alter it. Or, alternatively, we can conduct an experimental study, applying different treatments to various subjects and then recording the results. For instance, if we just write down the color of each car on our block, that's an observational study. If we take each car for a drive to measure its acceleration or maximum speed, we are clearly conducting an experiment.

The difference is not as obvious as it might sound. Simply put up, an online survey is clearly an observational study. But what if we first give our subjects a piece of information and then ask them a question? Have we applied a treatment upon them? In fact, extreme care must be taken to make sure that how we ask our questions doesn't impact the results. Even the order in which we present our multiple-choice answers matters a lot!

Cross-sectional Studies VS Longitudinal Studies

Studies can also be categorized based on the time frames they cover. Cross-sectional studies are performed by collecting data at a particular point in time. Retrospective studies use historical data to track how things have evolved in the past. Lastly, longitudinal studies follow a selected cohort of people over time into the future, sometimes even decades. It is important to realize that things always change and that just knowing something about the state of the world right now is not necessary fully informative about how it once was or how it will be in a few years.

Create representative samples

Why it is hard to get unbiased data

How can we make sure that our data is unbiased? The most straightforward answer is that our sample should be picked at random. If every person has the same chance of being selected, then we have what is technically referred to as a simple random sample. One way to achieve this is to select people in a systematic manner, by picking every 5^{th} or every 100^{th} person from a list. This can be easily accomplished, for instance, by selecting all the people whose Social Security Number ends in a 0 or a 5.

Conversely, what you do not want to do is what is commonly referred as convenience sampling, which is just asking a few random strangers off the street. People in your neighborhood often come from particular socio-economic groups, and are generally not representative of the city as a whole. They are certainly not representative or the entire country and even less of the diversity that is present in the world. The same biases are present among your friends on social media.

There are various ways that statisticians have come up with to achieve this, and we will cover some of the more widely used techniques today. What you need to keep in mind, however, is that if care is not taken to make sure that you sample properly, the data you collect will be completely unusable for proper statistical analysis. In that case, the only right thing to do is to start all over again. It's hard to overstate the importance of proper sampling!

Stratified sampling

One common technique is called stratified sampling. What you do is subdivide your population into clearly identifiable subgroups, technically referred to as strata, based on their inherent characteristics. The strata should be mutually exclusive but collectively cover the entire population. For instance, you might want to sample men vs. women. Or we could study dog owners vs. cat owners vs. those who don't own either. You can then randomly select a sample from each stratum, just like you would from the whole population, and compare and contrast your results for each subgroup.

Cluster Sampling

Another similar yet distinct approach is called cluster sampling. You first need to divide your population into clusters, and then survey every person in each cluster. For instance, imagine that we are running 10 sections of Data for Managers, and people were assigned to them randomly. In such a case, it makes sense to survey just a few of these sections, which should be still representative of our entire student population.

Test your Results

How do we know if our studies or experiments are not producing just random results? One of the ways to make sure is to repeat them a few times to see if the outcomes are replicated. This is exactly what the scientific community does to test findings by replicating the studies on different populations and under various circumstances. Just because something applies to a specific group, it doesn't mean that the results can be generalized to every possible situation.

Blinding

There is a commonly utilized technique called blinding, which is when the subjects are not aware of which treatment they are getting. In laboratory-administered drug tests, it's typical that half of the people get an actual treatment and the other half a placebo, yet the subjects are usually not told which one they are getting. An even better approach is when the experimenter is also not aware which one is which. This is called double-blinding and it ensures that our biases don't inadvertently influence our analysis.

Confounding

Another really important concept to keep in mind is confounding. Confounding occurs when it's not clear which particular factor actually influences the outcome. For instance, it's well known that smokers are more likely to have certain kinds of health issues. However, it is also well established that smokers that come from lower socioeconomic backgrounds are typically more likely to engage in other high-risk behaviors. It is actually almost impossible to untangle the effects of these factors from the direct impact of tobacco consumption.

Measures of Center and Variation

Measures of Center

The three basic measures that describe the "center" of any dataset are mean, median, and mode.

Mean

Mean is more commonly referred to as average. To calculate it, you need to add all of the values in a dataset and then divide this sum by the number of data points. This is the simplest and most commonly used calculation in statistics. It is also relatively stable across various samples you might draw from a population.

Median

Median is literally the number in the middle of a dataset. However, first you need to make sure to order the data from the lowest to the highest value so that you can simply pick the value that's right in the middle. For smaller datasets, you can do this by simultaneously crossing values on both sides, in pairs. If you have an even number of data points, you need to average the remaining two values near the center.

Mode

Mode is the number that appears with the most frequency. If no value repeats, then mode cannot be calculated. Conversely, if two or more sets of numbers tie for the same highest frequency, we call this a bimodal or multimodal distribution. Please note that many computer tools, like Excel, don't deal well with multiple modes and simply give you the first one they find.

When and how to use those measures

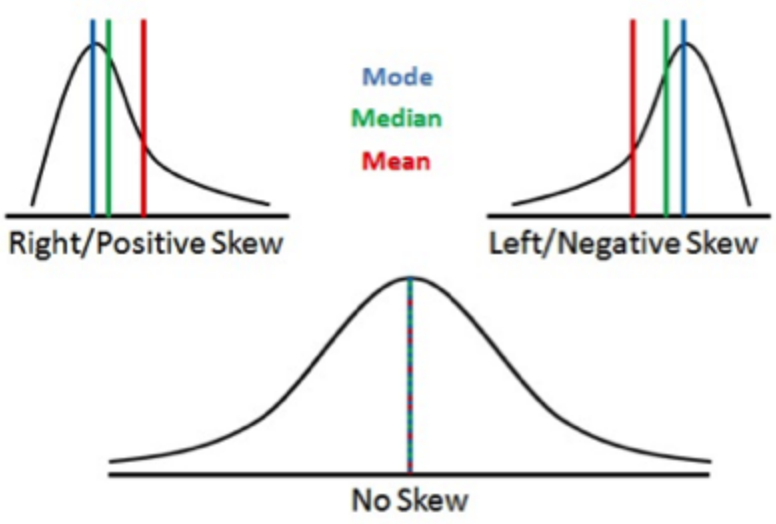
Means are simple to calculate, but they might not be best for certain datasets. For instance, if you have 9 people earning $50k and 1 person earning $1M, the mean is $145k, which is actually not representative of anybody in this dataset. Both the median and the mode here are $50k, which makes a lot more sense. In other words, outliers can disproportionately affect means. This is why government statistics usually cites median rather than mean incomes.

$$x = \sum_{i=0}^{N} w_i x_i / \sum_{i=0}^{N} w_i$$

Median can be found not only for quantitative but also for ordinal data. Mode is the only measure of center that works with nominal data sets. In other words, only some of these measures are possible to calculate depending on what kind of data you are dealing with.

A formula below can be used to figure out so-called weighted means. This is how final grades are calculated when, say, 80% of the grade is based on tests and the other 20% on class participation. What you do is multiply each value by the assigned weighting factor, add all of these up, and then divide this by the sum of all the weights.

If a distribution is symmetrical, meaning that it looks like the proverbial "bell curve," then mean, median, and mode are the same. In most situations, our data is skewed. We say that it's skewed to the right if it has a longer tail on the right, which puts mean and median to the right of the mode. If it's skewed to the left and has a longer tail on the left, the order of measures of center is reversed.



Having a normal distribution is important for certain statistical calculation. One example is a Student T-test that can be used in an A/B test to see if there is a statistical significance between two variance. Usually operations around means don't need a normal distribution but a best practice is to always check the assumptions behind each theorem.

Measures of Variation

Range

This brings us to the second part of this lecture, which covers measures of variation in the data. The most basic measure of variation is range, which is calculated as the difference between the largest and the smallest value in the dataset. For instance, in San Francisco the record low since 1849 was 27 degrees Fahrenheit and the record high was 103 degrees. This gives us a range of 76 degrees. It's a misleadingly large number, which is certainly not representative of the actual daily variations in temperature.

Standard Deviation

A more nuanced approach is to use all of the data available and to see how often and how much it deviates from the mean. That's the reason that we commonly utilize a measure called standard deviation, as shown below:

$$\sigma = \sqrt{\frac{\sum_{i=0}^{N} (x_i - \overline{x})^2}{N-1}}$$

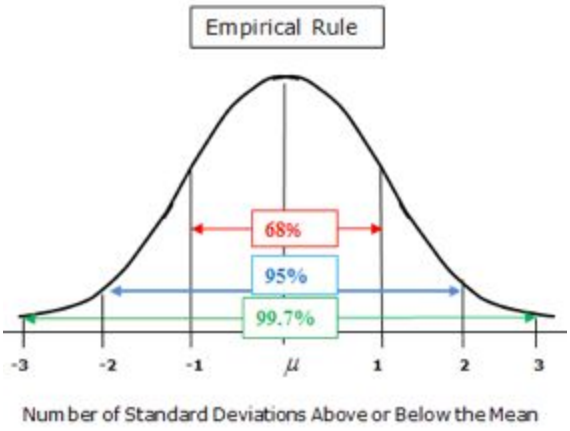
In order to calculate it, you first need to know the mean of your sample. You then subtract the mean from each of the values in the set and square each difference. All of

these results are summed up together and the sum is divided over the number of data points minus one. What you now have is called variance. The last step is to take a square root to bring this calculation back to the original units of the dataset. This operation also makes standard deviations a positive value.

While standard deviations are sensitive to outliers, they are not as affected by them as ranges. Having larger datasets further reduces the impact of a few extreme values. Still, it makes sense to check for any data that is too far away from "typical" values and to potentially exclude some points if they look suspicious. It helps to remove them before performing any further calculations.

Standard deviations can be calculated for any distribution. However, more often than not what we deal with are so-called normal or Gaussian distributions that look like a bell curve. It turns out that many phenomena in nature as well as in technology are normal distributions. This includes everything from human heights and IQ's to grades on math tests to sizes of manufacturing defects in industrial production.

It turns out that about 68% of values are usually located within one standard deviation above or below the mean and about 95% within two standard deviations. Extending our range to three standard deviations from the mean covers roughly 99.7% of all data, leaving out only 3 out of every 1,000 data points. However, the further away we get from the mean, the less predictive this rule will be in real life.



Z-Score

For each individual number in a dataset, you can also calculate its standardized value or Z-score, which measures how many standard deviations it is situated away from the mean. Z-score is simply the distance from the value to the mean divided by the standard deviation.

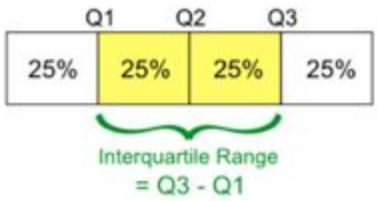
$$z_i = \frac{x_i - \bar{x}}{\sigma_x}$$

Z-scores are positive for values above the mean and negative for those below it. If a Z-score is 1.5, it means that a particular value is 1.5 standard deviations larger than the mean. For example, if the average test score for the class is 80 with a standard deviation of 10 and you personally got 95, your score is exactly 1.5 standard deviations above the mean. A grade of 65 has a Z-score of -1.5 in this situation.

As a matter of convention, we refer to any values within plus or minus two standard deviations as "ordinary" and those beyond as "unusual." Those outside of three standard deviations from the mean are "highly unusual." In manufacturing, once you get to six standard deviations or "six sigma" on either side, you are down to just 3.4 defects per million units outside of the range of acceptable values. These kinds of calculations are commonly used in quality control.

Box Plot

Another way of determining one's standing is to look at quartiles, quintiles, deciles, or percentiles, which further quantify where you stand within a sample or a population. These break an entire ordered dataset into 4, 5, 10, or 100 equally sized partitions respectively. It's typical for SAT or GRE test results to also include percentiles.



To create quartiles, you need to find 3 numbers that divide the sorted data into 4 equal parts, as shown below:

Q2, which is right in the middle of a dataset, is equal to the median. The difference between Q3 and Q1 is called the interquartile range, which covers the "middle 50%" of all of the values. The 5-number summary that is typically used to describe a dataset refers to the minimum and maximum values plus Q1, Q2, and Q3.

5 A Exercises

1. Data Types

For each of the following features, determine if they are:

- Quantitative / Qualitative?
- Ordinal / Nominal?
- Discrete / Continuous?

1. Age
2. Salary
3. Countries
4. Color
5. Gender
6. Client's satisfaction about a service

2. Sampling types

1. Google hired you because they are developing their new self-driving car. However, they would like to know if there is a real business opportunity. They would like to know if customers would be ready to let them drive by a robot. To help you, Google ran a simple and unbiased survey. Because Google would like to optimize the cost of this study, instead of surveying all the car buyers, they decided to survey the car buyers of 10 different cities picked randomly.
    a. What is this type of study (Observational / Experimental / Cross-sectional / Longitudinal)
    b. What type of Sampling did Google use for this survey (Stratified sampling / Convenience Sampling / Cluster Sampling / Simple Random Sampling)

2. Grammarly launched their product that will help people to never make grammar mistake. They would like to see if there product actually help their customers being better at writing English. The team wants to run that experience on non-native english speakers living in San Francisco. After further research, they discovered that non-native english speakers living in SF include 10% French, 10% Italian, 5% German, 20% Indian, 15% Chinese, 5% Japanese, 5% Taiwanese, 5% South Africans, 5% Moroccans, 15% Mexicans, 5% Brazilians. They gathered a group of people that represents exactly this population. The experiment will be during 6 months.
    a. What is this type of study (Observational / Experimental / Cross-sectional / Longitudinal)
    b. What type of Sampling did Grammarly use for this survey (Stratified sampling / Convenience Sampling / Cluster Sampling / Simple Random Sampling)

3. Chase would like to get more people to create a bank account. They thought about sending a blast email campaign to potential customers. However, they would like to test this new email campaign to a subset of users first. The bank is in a rush though, the email campaign is planning to be sent 2 months from now. Therefore the data team chose to simply select the people that are likely to respond to these email quickly.
    a. What is this type of study (Observational / Experimental / Cross-sectional / Longitudinal)
    b. What type of Sampling did Chase use for this survey (Stratified sampling / Convenience Sampling / Cluster Sampling / Simple Random Sampling)

Homework: Finish the notebook exercises

Everything is in the title 😉.

Homework: Descriptive analysis of the Titanic dataset

Perform the Exploratory Data Analysis (EDA) of the Titanic dataset. To do so, create a new notebook on Colab, load the Titanic dataset with the following code:

```
from seaborn import load_dataset

titanic = load_dataset("titanic")
```

And use the notebook to explore the dataset:

1. Present the most striking summary statistics with description in comments cells
2. Find the most interesting plot, and print and comment them

Further Reading

1. *MathIsFun, Standard Deviation:* https://www.mathsisfun.com/data/standard-deviation.html

2. *StatQuest, 2019, Quantiles and percentiles... Clearly explained:* *https://www.youtube.com/watch?v=IFKQLDmRKoY*

3. *TutorialsPoint: Pandas Descriptive Statistics:* *https://www.tutorialspoint.com/python_pandas/python_pandas_descriptive_statistics.htm*

4. *DataQuest: Descriptive Statistics in Python:* *https://www.dataquest.io/blog/basic-statistics-with-python-descriptive-statistics/*

Topic 5B

Estimates and Sample Sizes

What you will learn in this class

- What are confidence intervals
- How to build confidence intervals
- Understand the difference between Precision and Accuracy
- Create confidence intervals for Population Means and Population Proportions
- Know how to calculate the minimum sample size depending on your population parameter

Introduction to Confidence Intervals

Why you need a confidence interval

One of the most critical issues when conducting experiments is deciding when you have collected enough data so that you can make an informed decision. While there are always pressures to deliver results on accelerated schedules, it is vital to understand the potential repercussions of making an important business call with a low confidence in your data. The concept of statistical significance is one of the most common tools that statisticians utilize to address this issue.

Any time we get samples and calculate their statistics, they can serve as estimates for the actual parameters in the overall population. For instance, collecting data on age and gender from 100 randomly selected Product School alumni can provide us with a decent general idea about the demographics of all of our graduates. Nevertheless, such results are inherently not perfect because they don't represent everybody in our alumni population. In order to properly quantify the impact of this common uncertainty, we introduce the concept of confidence levels.

After today's lesson, you should be able to present your results in a format that will allow you and your peers to assess the precision of your measurements within a certain

range of values and with a specific level of confidence. In the hypothetical example above, you might end up with a pair of results such as:

With a confidence of 95%, we can say that:
1. Somewhere between 40% and 80% of our alumni are male.
2. The average age of our alumni is between 20 and 40 years.
(Disclaimer: These data are shown for demonstration purposes only and do not represent our actual alumni population in any way)

What is confidence interval

What "confidence" means is that in each case there is a 95% chance that the true value in the population is within the ranges above. On the other hand, there is a 5% chance that the actual population parameters are outside of these ranges. These kinds of estimates are usually the best information that we can get. You can then decide whether your confidence intervals are narrow enough to answer the questions you need answered with sufficient precision. Oftentimes, the ranges are too broad to yield actionable insights.

Why not try to get a 99% confidence or even 99.99%? In general, the higher confidence level you desire, the more data you need to collect. Setting a realistic level of confidence allows you to make decisions reasonably quickly and with "sufficient" confidence.

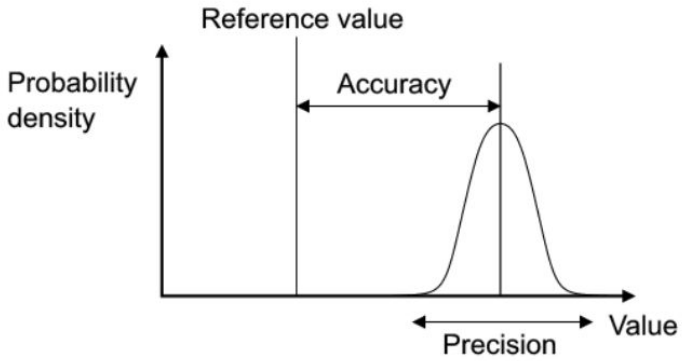
How to choose the right confidence interval

The level of confidence desired generally depends on the industry and on your specific function. You might only want to get an 80% confidence for a quick decision regarding a small marketing campaign. On the other hand, a much higher confidence is needed when calculating the chance that a guided missile will hit its target. Larger companies might also seek higher levels of confidence in their data and usually have more human and technological resources to obtain it.

Accuracy vs. Precision

Precision

One more distinction has to be made at this point. What confidence intervals do is estimate the precision of the result, as demonstrated below:



What this stylized graphics demonstrates is that even if we get a narrow confidence interval with a high degree of confidence, it doesn't guarantee that your results are actually correct. Precision is not the same thing as accuracy. In other words, Precision corresponds to how far are each data points from each other while accuracy corresponds to how far are all the data points from the "True Value".

Accuracy

Our results might not be accurate for a variety of reasons. This can happen for technical reasons such as if our data collection equipment is faulty or our databases store data improperly. More commonly, however, we might simply have biases in how we collect our data. If, for instance, younger men are for some reason more likely to reply to our survey on age and gender, we can get a systematic bias underestimating the average age and overestimating the proportion of males among our alumni.

Even with these caveats, such calculations are far from trivial. We will spend much of the rest of the lesson to demonstrate the statistical principles behind them.

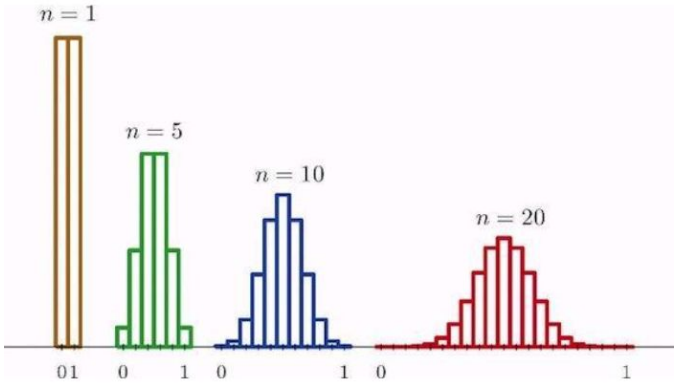
The Central Limit Theorem

Definition

One reason that these calculations are even possible is because of the so-called Center Limit Theorem. What it states is that if we collect a large number of sufficiently large samples from a population, the mean of all the samples will be approximately equal to the mean of the entire population. It also turns out that if you plot the samples' mean, they will fall into the classic pattern of a normal distribution.

Minimum Sample Size

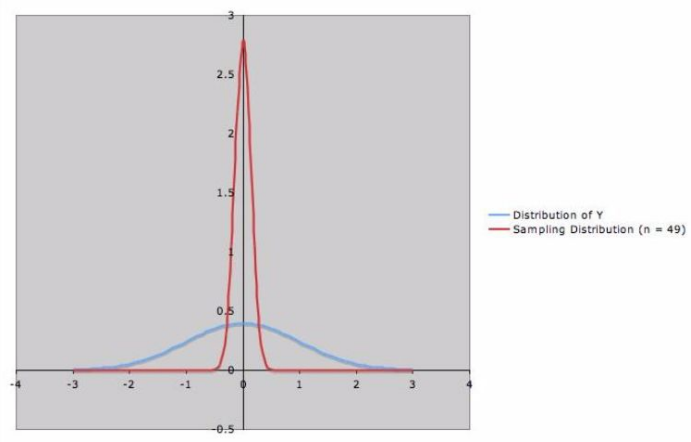
The graph below demonstrates what happens if you collect samples of different size and then plot all of their means on a single graph:



As the sample size increases from 1 to 20, the distribution looks more and more like a "bell curve." It turns out that each sample should contain at least 30 individual data points for the theorem to take full effect.

Standard Error

The major implication of this is that even if our data is not normally distributed, we can treat the distribution of sample means – also called sampling distribution – as if the data were normally distributed. If you plot both the original distribution as well as the sampling distribution together, you will get a graph similar to this:

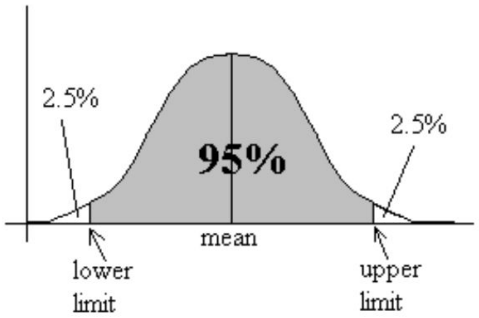


The sampling distribution is significantly narrower and therefore has a standard deviation that's much lower than that of the original population. In mathematical terms, we refer to the standard deviation of the sampling distribution as the standard error. It can be estimated by dividing the standard deviation of each sample by the square root of its size:

$$E = \frac{s}{\sqrt{n}}$$

In the above graph, where each sample contains 49 data points, the standard error is 1/7th of the standard deviation in each sample.

Any sample that we pick out of all of the possible samples will fall somewhere within the sampling distribution. Clearly, it is much more likely that it will be close to the mean of the population than that it will fall far away from it. If you're looking for a range of values that cover 95% of all possibilities and the distribution is symmetrical, then you will have a 2.5% chance of being left out on either side.



Confidence Intervals for Population Means

As you might remember from the previous lecture, there is roughly a 95% chance that a data point for a normal distribution will be within plus or minus 2 standard deviations from the mean. The exact number is actually 1.96 standard deviations. The same rule applies here.

Student T-Distribution

Thus, in order to estimate the range of mean values that fall within the most likely "middle" 95% of a sampling distribution, we need to take the mean plus or minus t standard errors, where t is a so-called "student t distribution."

$$\underline{X} \pm t\frac{s}{\sqrt{n}}$$

Tables or formulas are generally used to calculate t for various sample sizes with the required levels of confidence. For samples larger than 30 data points, only the level of confidence really matters. This is a tedious exercise, so for practical purposes you should consider using one of the online calculators, such as this one: http://statdistributions.com/t/

What you need to keep in mind is that, for archaic historical reasons, you have to input the number of degrees of freedom, which is commonly shortened as d.f. It is actually just the number of data points in your sample minus one:

Degrees of freedom = n -1

If you have 25 data points, you have 24 degrees of freedom.

p-value

The probability, denoted as p-value, is simply the acceptable chance that your sample mean is not within the confidence interval. For our examples above:

p-value = 100% - 95% confidence = 5% = 0.05

Z-values

For larger samples, we generally replace t-distribution with z-values, which only depend on your desired confidence level:

Confidence Level	Z Critical Value
80%	1.28
90%	1.645
95%	1.96
98%	2.33
99%	2.58
99.8%	3.09
99.9%	3.29

If you rearrange the formulas above to solve for n, the sample size, you can determine how many data points are necessary to obtain in order to achieve a desired confidence level with standard error not to exceed a particular value.

In the formula below, however, we also need to have sigma, which is the standard deviation for the entire population. It is usually not available, so we can substitute the standard deviation for the sample instead to get an estimate:

$$n = \left(\frac{Z_c \sigma}{E}\right)^2$$

Any number that you get from the above formula should be rounded up.

Confidence Intervals for Population Proportions

A similar approach can be taken to estimating the true proportion in the population using confidence intervals. In general, only z-values are used for proportions, as listed in the table above.

$$\hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

If you know roughly what proportion you're expecting to get, you can also estimate the number of data points that you need to get it within a certain margin of error and with a critical value z based on a required level of confidence:

$$n = p(1-p)(\frac{z}{E})^2$$

If no estimate is available for the expected proportion, it is prudent to use 0.5 to get a conservative estimate for the minimum sample size required.

Lastly, sometimes we also want to compare values between two different populations to determine how much they differ from each other. It is common in the business world to construct confidence intervals for each estimate and to see if they intersect. While this is not completely valid from the strictest statistical point of view, such a simple exercise often provides us with a general idea on how much two populations differ from each other.

1B Exercises

1. Confidence intervals for means

Using the Seaborn tips dataset, create a notebook on Colab and:
(https://github.com/mwaskom/seaborn-data/blob/master/tips.csv)

1. Compute the average tips obtained by waiters and the one obtained by waitresses.
2. Compute the corresponding 95 % confidence intervals.
3. Can we say if the average tip of waiters are different from the one obtained by waitresses with a 95 % probability?

Homework: Confidence intervals for proportions

In the same Seaborn tips dataset mentioned above, is the proportion of waitresses significantly different from the proportion of waiters?

Further Reading

1. *Confidence Intervals, 2019, Wikipedia:* https://en.wikipedia.org/wiki/Confidence_interval
2. *A/B Testing, Optimizely:* *https://www.optimizely.com/optimization-glossary/ab-testing/*

Week 6A

Introduction to Machine Learning

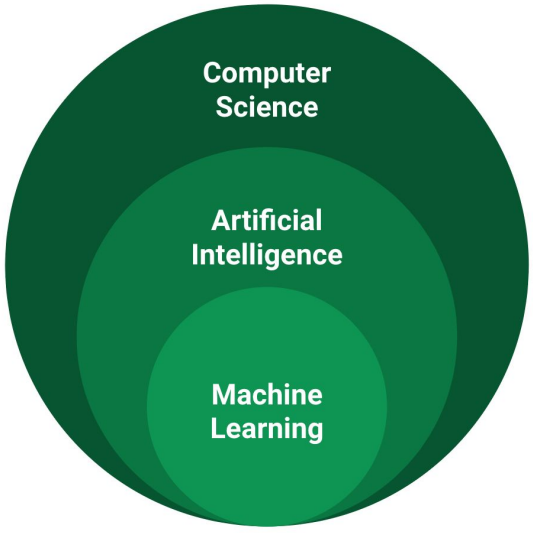
Machine Learning is all around these days, and is mentioned in a variety of contexts ranging from target marketing, to autonomous cars, but its ubiquity makes it difficult to grasp what this term covers. The goal of this first part is to give a definition of what Machine Learning covers, can and cannot do, and how it is used in a business context. This class is the first of a series of 3 lessons on Machine Learning.

Machine Learning definition and use cases

General presentation

Machine Learning is a subfield of Artificial Intelligence (AI). In a nutshell, AI usually uses Machine Learning, Symbolic Methods (old-fashioned), Mathematical Optimization, or a combination of those.

Also, Artificial intelligence is a subfield of computer science, as it consists in realising cognitively-elaborate tasks with machines, the machine systematically being a computer nowadays.



Machine Learning definition: "*A computer program is said to learn from experience E with respect to some class of tasks T and performance P, if its performance at tasks in T, as measured by P, improves with experience E*" Mitchell, 1997

This apparently sibyllin definition is broad enough to cover all of the applications of Machine Learning. To make it more concrete, here is a simple example: you want to predict the likelihood of a customer churning in the next month. To do so, you can train a machine learning model to **learn** what patterns in user behavior and data suggests a coming unsubscription. For the machine learning algorithm to learn those patterns you need a dataset of past user data with some of them churning of some others not churning. The **task** is the churn prediction, the **experience** is the fact that the learning algorithm sees more and more user examples during the learning process, and the **performance** is the churning prediction accuracy (how often is the algorithm right?). If the performance is improving as you give the algorithm more example, the computer is **learning**!

Here are a few examples of what the task, performance, and experience mentioned could be:

Examples:

- **Task**: classification, regression, transcription, translation, text structuring, anomaly detection, sampling, imputation of missing values, noise reduction,...
- **Performance**: task specific: accuracy (classification),...
- **Experience**: defined by the data set, supervised, semi-supervised or unsupervised learning, reinforcement,...

The explosion of Machine Learning (often simply presented as AI) is due to a combination of factors:

- Massive amounts of data collected through internet
- Scientific advances by the Machine Learning research community
- Better hardware (notably GPUs)
- Need for personalization
- Return On Investment proven (first by insurances and banks mostly)
- Promising applications

The applications of Machine Learning are numerous, but here is a non-exhaustive list that will be familiar to most of you.

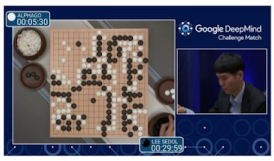
Speech to text

Image recognition

Automated translation

Recommender systems

Autonomous cars



And many others...

Note that for most companies the main use case for machine learning use structured data because the models are simpler to create, implement, and deploy in companies IT landscapes. Many businesses use machine learning to predict churn, have a targeted marketing process, or identify upsell opportunities automatically. Textual data is being leveraged more and more but is still cutting-edge as of 2019.

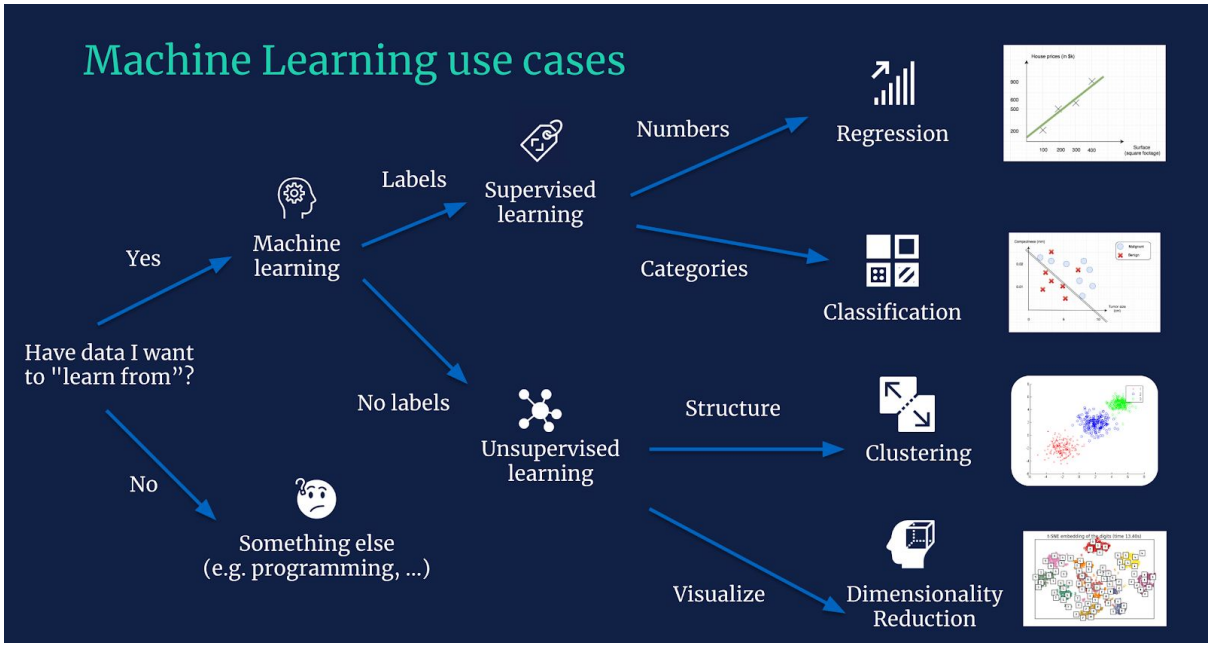
Learning tasks types

Machine Learning tasks can roughly be subdivided into 3 types, with further subdivision presented below:

- **Supervised learning**, the easiest and most common one, in which your goal is to predict a quantity or category given a set of predicting features. For instance, the churn prediction task mentioned above is a supervised learning task: you need to know the values of the *target variables* or *labels* (churn or no-churn) to be able to train a machine learning model. The task is supervised in the sense that you have to define a target variable and the model has to learn how to predict this specific variable.
- **Unsupervised learning** has to do with finding structures in dataset without trying to predict a given variable.
- **Semi-supervised learning** handles learning tasks in which you have labels for some examples but not for all of them, and you try to leverage all of those examples. For example if you try to predict the price of houses based on their surface, number of rooms, neighborhood, but you only know the price of some of them, you can still leverage the *unlabeled examples* (the houses for which you don't know the price) to find structures in the dataset and help in your learning task. This type of learning is rarely harnessed by companies.

- **Supervised Learning** can be further divided into **regression** tasks, in which your goal is to predict a quantity (e.g. the price of a house given its surface) and **classification** tasks, in which you have to predict a category (e.g. "malignant tumor" vs "benign tumor", given the tumor size).

- **Unsupervised Learning** also includes several learning tasks types, the main ones being **clustering**, the task of finding groups of similar examples in your dataset (e.g. customers with similar shopping behavior) and **dimensionality reduction,** which has to do with visualizing complex datasets in a human-readable way.



Learning tasks categories summary. This course mostly deals with supervised learning, i.e. predicting numbers or categories given features.

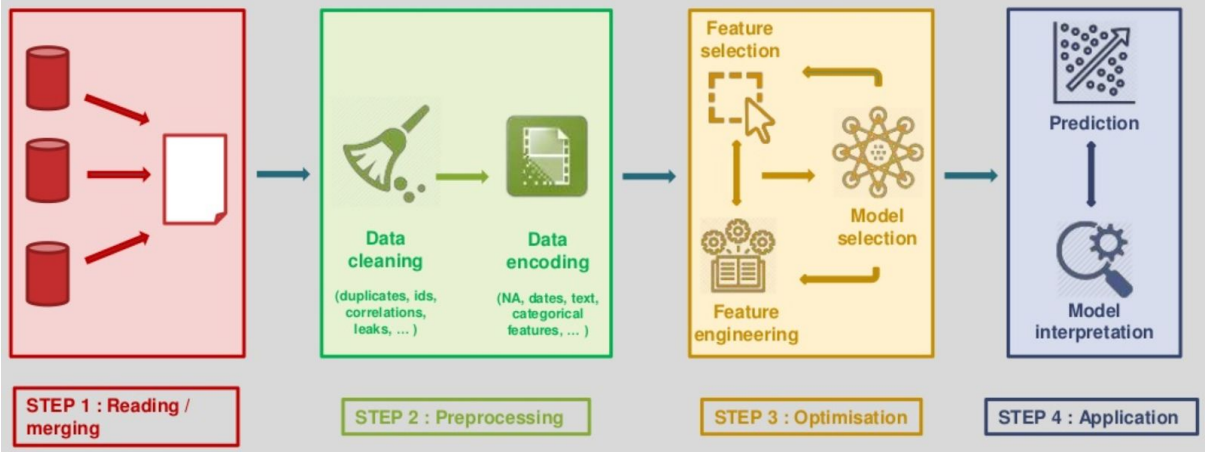
Machine Learning pipeline and preprocessing

Data preprocessing

Machine Learning models can only handle numerical values. But your input data is almost always has to be transformed to have the right format.

1. The **format of the data** itself may not be numerical, e.g. when dealing with textual data you have to find a way to transform it into numbers to be able to feed it to a machine learning model (e.g. you can replace words with a list of counts).
2. Your dataset will often contain **missing values**, for instance if your customers sign-up on your website and some form fields are optional, then some of your data points will have the corresponding field filled and the others will not. In this case you have to choose a missing values handling strategy. There is a variety of such techniques, but the main ones are the blunt removal or such examples from the training, substitutions by means or medians, or by the majority category for categorical data.
3. **Categorical data** has to be turned into numbers. For example, if you have several subscription tiers (basic, advanced, premium) for your customers, then you must transform these categories by numbers, and choose the corresponding strategy. In the latter example we can easily think of substituting the basic tier by 1, the advanced one by 2, and the premium one by 3, because there is a natural **ordering** between those categories, but in some cases there is no such natural ordering and other methods must be used (One-hot-encoding, labelling, custom grouping with domain expertise, ...)
4. Some machine learning algorithms further require that you **normalize** your data (i.e. subtract the column mean feature values and divide by the standard deviation, such that the output column varies around 0 on a typical scale of 1). The goal of normalization is to have all of the features varying on the same scale (e.g. if in a house price prediction example, the number of rooms typically varies between 1 and 5, while the surface typically varies between 200 sq. ft and 1000 sq. ft).
5. When preparing data for the training of machine learning algorithm, it is also common to perform **feature engineering**. The goal of feature engineering is create **derived features** that take the context into account. For instance, if you build a model to predict the price of houses worldwide given their surface, you may want to compute the surface of houses **relative to the local average**, because houses Hong-Kong are typically much smaller than in Montana, but you may want your model to have an information about how a specific house compares to the local standard in terms of surface. This is an example of feature engineering.

The data preprocessing is necessary and crucial. It has to be performed after reading the data, possibly for several data sources, and happens before training the machine learning model and optimizing it.

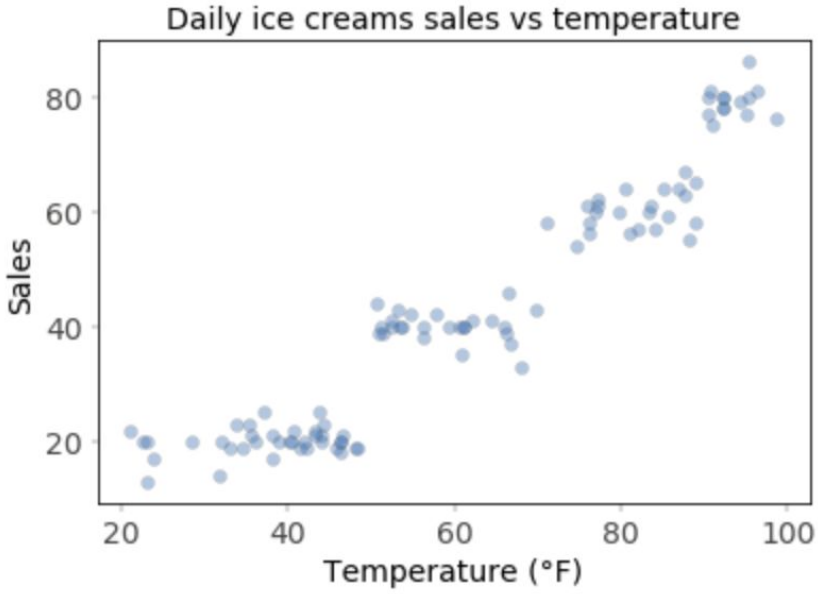


Picture from AutoML, Axel de Romblay (5/25/2019) (https://www.slideshare.net/AxeldeRomblay?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview)

Underfitting and overfitting

First Machine Learning model

In this part we will use the following synthetic example to illustrate the main concepts of Machine Learning: An ice cream truck owner records the temperature and the number of ice creams sold everyday. After 100 days, he obtains the following data:



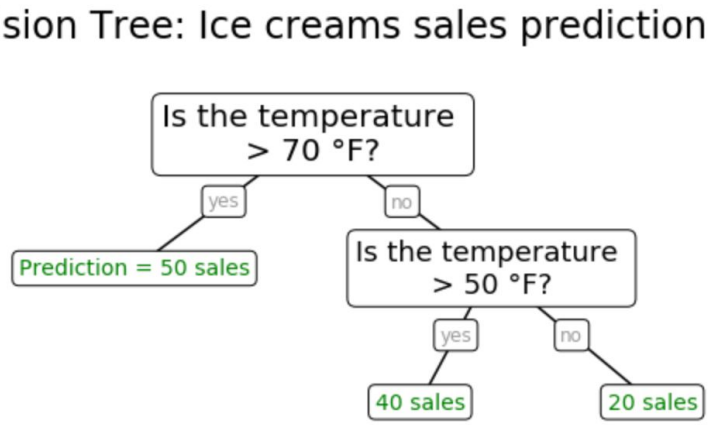
Note that a four-steps pattern is easily visible from the plot:

- Below 50 °F the owner sells around 20 ice creams during the day
- Between 50 °F and 70 °F he sells roughly 40 ice creams
- Between 70 °F & 90 °F → ~60 ice creams
- Above 90 °F → ~80 ice creams

📝 Also note that the target variable is a number here (number of ice creams sold), so we are dealing with a regression problem.

As an introduction to Machine Learning we will present the simplest of all: the Decision Tree sometimes called CART for Classification and Regression Tree. As the name suggests it can be used for regression and classification.

It consists of a series of binary splits. The variables and threshold of the splits can be hand-crafted, but in a machine learning setting they can be learnt on data, with the learning objective of minimizing the prediction error. The following pictures illustrates how such a tree may work for a house price prediction task.



First machine learning model: the regression decision tree. The features used for splits, the splits values, and the prediction values in the leaves are all learnt from the training dataset.

In our simple ice cream example, since we only have one predictor and a pattern is obvious from the plot, we could have hand-crafted the split values and associated predictions. But in order to explain the principle we will explain how to learn optimal split values automatically.

In order to train any Machine Learning model one needs three ingredients:

- The **training data**, that is an identified target variable (number of ice cream sold) and a set of predictors (only the temperature here) in our current, supervised learning setting.
- A Machine Learning **model**, here we have chosen a regression tree
- An **optimization algorithm** to fit the parameters of the ML model, i.e. the split values and leaves prediction values in our current example

Here is the optimization algorithm typically used for training decision trees:

Choose a maximum tree depth *max_depth*

While the depth of the tree is < *max_depth*:

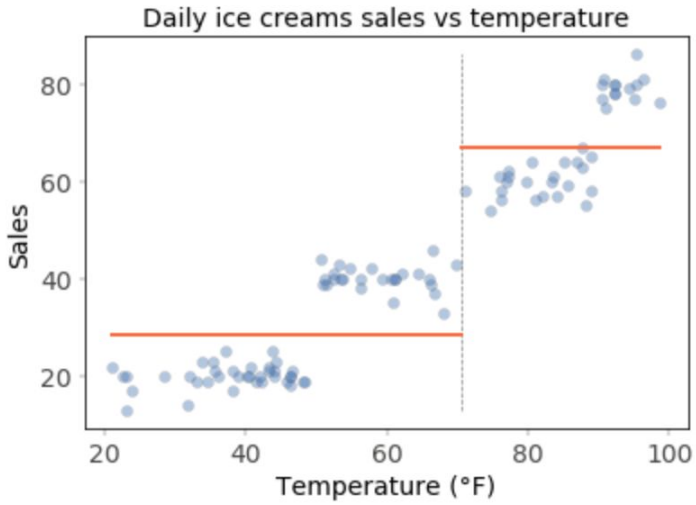
- For each possible split value:
  - Predict the average of target values for each leaf
  - Compute the Mean Squared Error (MSE) between the predictions and the true target values: $MSE = \sum_{i=0}^{N}(y_i - \hat{y}_i)^2/N$
- Return the split value with the lowest MSE

The regression decision tree optimization algorithm. At each step, we sought the split feature and value that minimizes the mean squared error of the model.

Underfitting: not enough parameters in the model to mimic data patterns

Before training our tree (i.e. adjusting its parameter), we have to define the model hyper-parameters, that is, the ones that are fixed in advance and won't be adjusted during the training phase. Regression trees can be fully defined by a single hyper-parameter: the maximum depth.

Here is a graphical representation of tree with maximum depth of 1 trained on the ice cream synthetic dataset. Note that a maximum depth of 1 means that we only allow for a single split.



Graphical representation of a single-split decision tree predictions. The data points used to compute the optimal split are the blue points, and the red line is the value predicted by the decision tree.

As you can see from the graph, the single-split tree is not very accurate: the blue points are far from the red lines on average.

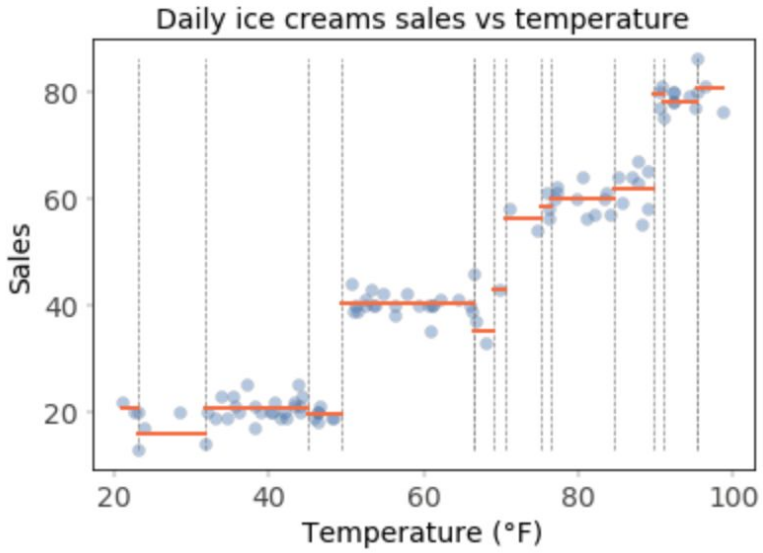
The predictive performance of the algorithm can be measured by the Mean Squared Error (MSE): the further the prediction from the data points on average, the larger the MSE. Here the MSE is large compared to the typical variation of the target (measured by the standard deviation of the number of sales.

This is a situation of **underfitting: the model does not have enough parameter/complexity to mimic the patterns of the data.**

Overfitting: over-complex model that captures noise rather than trends

Since our single-split tree is not complex enough to capture the patterns in the dataset, hence suffering from an **underfitting** problem, a natural solution consists in adding more parameters to avoid the problem as long as the computational resources are sufficient to run the corresponding training algorithm.

But we will see that another problem arises when the model is too complex. Here is a graphical representation of a decision tree with 16 leaves (maximum depth of 4).



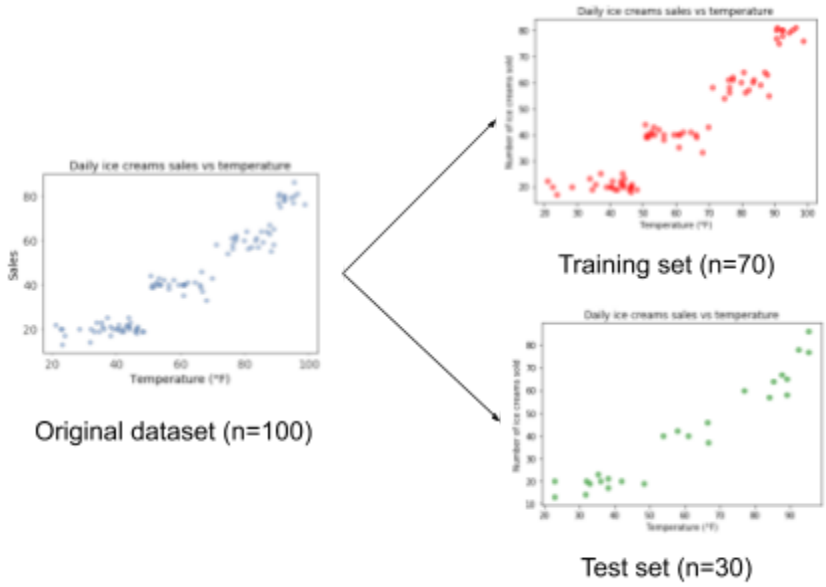
Graphical representation of a 16-leaves regression decision tree, fitted on the ice cream dataset. Notice how the numerous splits allow the model to reproduce even the most local variations in the dataset.

Our decision tree now reproduces the dataset very well, the red lines (model predictions) being very close to the blue points (training set).

The problem is the following: the model *fits the data* ***too well***. It wouldn't generalize well to new data points. Since the purpose of a machine learning model is to accurately predict the target for future, new examples, the model won't be accurate for future task. This problem is called **overfitting**.

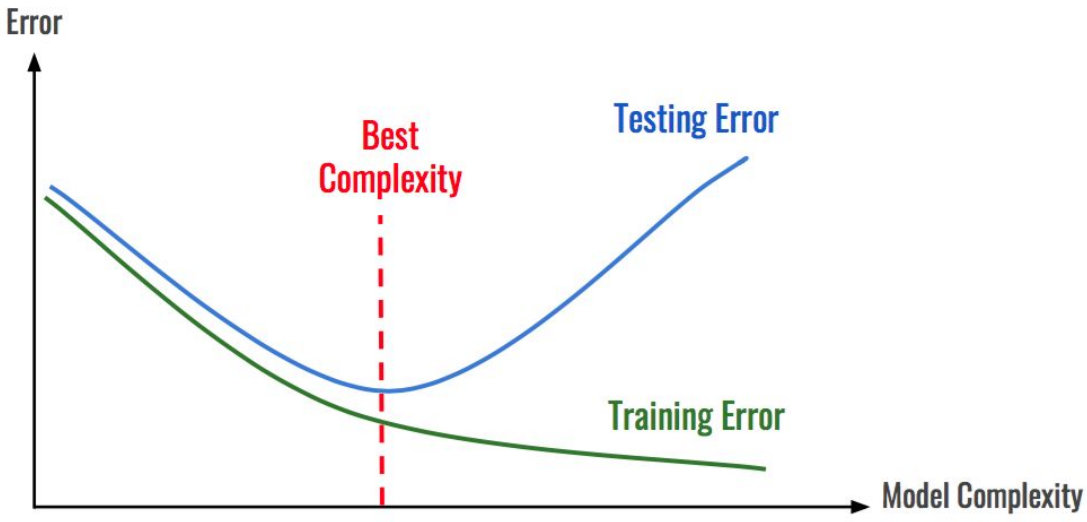
While underfitting is easy to spot (from the error on the training set) and to solve (by adding more parameters to the model), overfitting is more challenging in general.

In order to monitor overfitting, we use the cross-validation method depicted below.



The cross-validation method: the original data is split in a train and a test set. The train set is used to adjust the machine learning model parameters, while the test set is used to measure the ***generalization error****. A large generalization error compared to the training error betrays a situation of overfitting: the model will not generalize well to new instances.*

All machine learning models have a certain degree of complexity controlled by a set of hyper-parameters. For decision trees the maximum depth of the tree fully controls the model complexity.

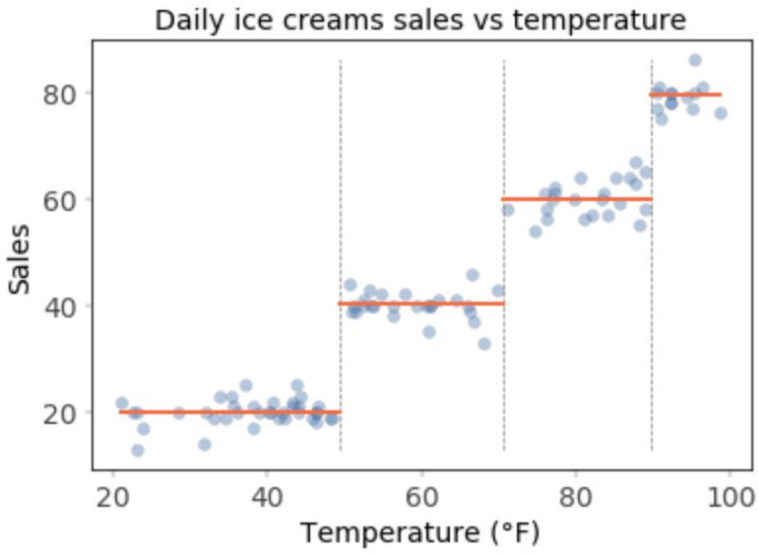


Picture from HackerNoon: [Memorizing is not learning!](https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42) (6/1/2019) (https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42)

Since in machine learning your final goal is always to minimize the generalization error, we usually plot the training and testing error while varying the model complexity (e.g. compute the training and testing error for a maximum depth of 1, then for a maximum depth of 2, ...). The best model is the one that minimizes the testing error.

⚠Note that most often data scientists split the dataset into three parts: a training set that is used to adjust model parameters, a **validation set** to find the optimal values of the hyper-parameters, and finally a test set that is used to compute the real generalization error or the model.

In our case of the ice cream dataset model modeled by a decision tree, the testing error is minimum for a maximum depth of 2, i.e. when the tree has 4 leaves (a.k.a. *Terminal nodes*). This finding is consistent with the pattern we observe visually: the data is a "four-step staircase".



For an intermediate complexity (maximum tree depth of 2, i.e. 4 leaves), the model captures the trends without the noise: we are not underfitting nor overfitting, and the generalization error is minimum.

6A Exercises

Homework: Finish the notebook exercises

Everything is in the title 😉.

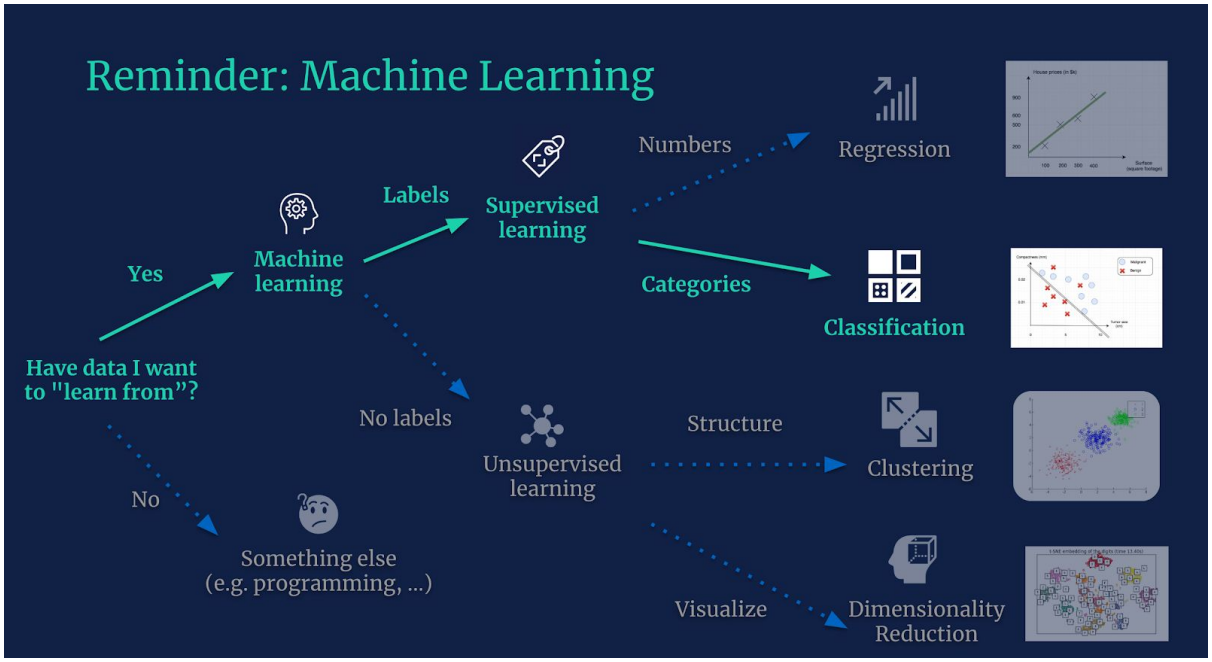
Further Reading

1. *Vas3k blog, 2019, Machine Learning for Everyone:* http://vas3k.com/blog/machine_learning/
2. *R. Priyadharshini, 2018, A Simple Overview of Machine Learning:* https://www.slideshare.net/priyadharshiniR8/simple-overview-of-machine-learning
3. *Axel de Romblay, AutoML:* https://www.slideshare.net/AxeldeRomblay?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview
4. *R2d3 blog, 2018, Visual Introduction to Machine Learning:* http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
5. *R2d3 blog, 2019, Underfitting and Overfitting:* http://www.r2d3.us/visual-intro-to-machine-learning-part-2/
6. *Wikipedia, Cross-validation:* https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Week 6B

Machine Learning: Models

This class presents machine learning based methods to predict a category given a set of features. This task, called classification, is different from the regression task seen in the previous class. Using our machine learning tasks taxonomy presented in the introduction on machine learning:

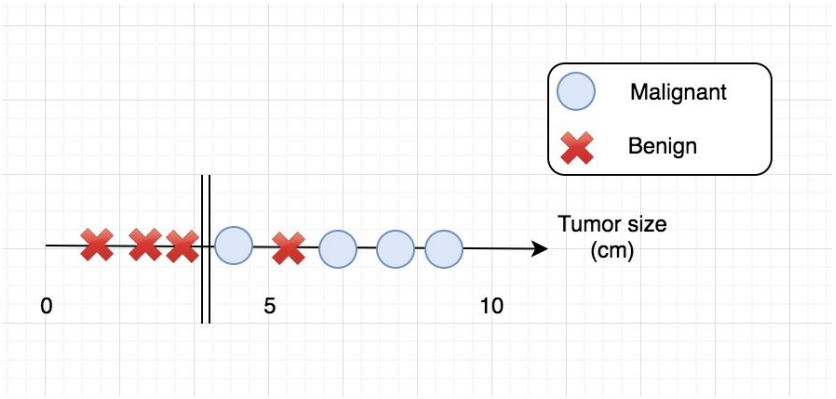


Classification is a supervised learning task, the goal of which is to predict an output category given a set of predictors.

As a running example, we will use a tumor classification dataset (a synthetic one for the first examples then a real one for the hands-on part). This dataset contains analysis of breast masses, together with a diagnosis column that is 0 for benign masses and 1 otherwise. Quoting the dataset documentation: *(https://scikit-learn.org/stable/datasets/index.html#breast-cancer-dataset)*
"Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image."

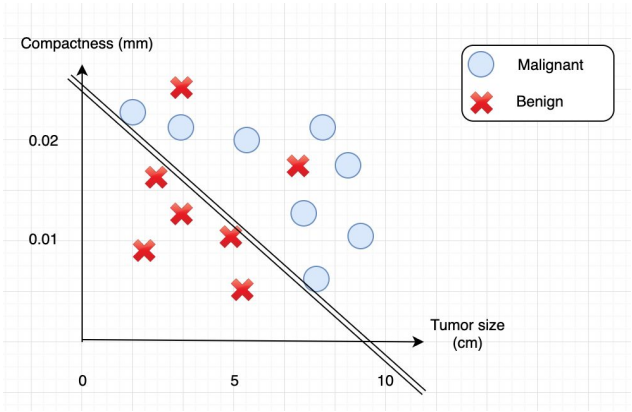
Hence our task is to predict a binary outcome (benign of malignant) given a set of numerical characteristics of the cells present in the mass that was analyzed.

The malignancy prediction problem is illustrated below:



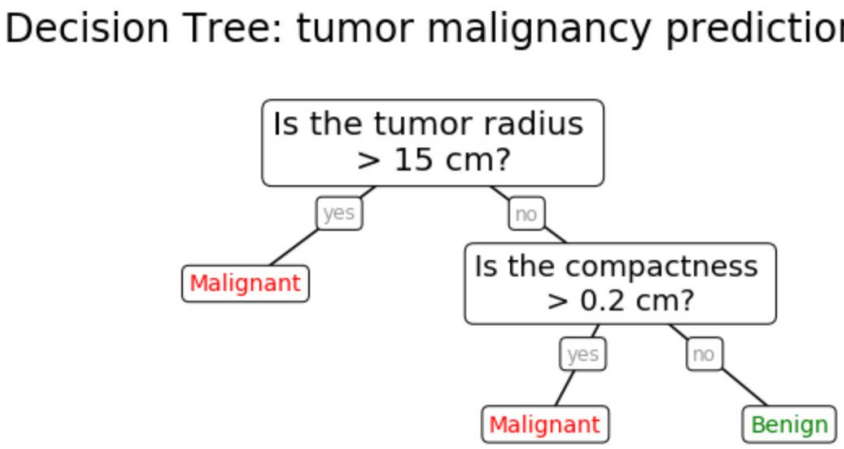
*Example of a tumor dataset. The task consists in predicting whether a tumor is benign or malignant given its size. This is a **binary classification** task because there are only two possible outcomes: benign or malignant. Our task is to find the best decision boundary to minimize the number of false positives and false negatives. Note that even in this simple example there is no perfect decision boundary as we will always have either one false positive or one false negative, at least.*

Similarly to the regression tasks presented in the previous classes, we can improve the accuracy of our model by feeding it with additional predictors, for instance we may have computed and recorded the compactness of the tumor (ratio volume/surface, describes how "spherical" is the tumor). In that case the dataset can be represented with the following graph:



Tumor dataset with two predictors: the tumor size and its compactness (i.e. how "spherical" it is). Our goal is also to find the right decision boundary between malignant and benign example.

As a first classification model we will use decision trees again. The decision trees can be used both for regression and classification. While the leaves (a.k.a. *terminal nodes*) were predicting a continuous value for regression trees, they predict a class (benign or malignant for classification trees. For this reason, classification and regression are usually gathered under the name CART, which stands for Classification And Regression Trees.



*Classification decision tree example on the breast cancer dataset. As for the regression decision tree, a series of binary decision leads to the **leaves** of the tree, where a class prediction is given.*

Note that the tree training algorithm is slightly different in the classification setting. In the regression case the tree is grown *greedily*, starting from the root, and each split is the one that minimizes the Mean Squared Error (MSE), among all possible features and split values. The leaves predictions are the average of the samples selected by the successive splits.

There is no MSE for classification, but its equivalent is the **misclassification rate**: in a given subset of the data (a leaf at a given stage of the tree growth), what is the proportion of the majority class? If the leaf is pure (i.e. contains only examples belonging to one class) then the misclassification rate is 0, if the leaf contains a balanced mix of two classes (impure case), the misclassification rate is 50%. Note that in practice, the misclassification rate is rarely used and data scientists favor the gini impurity coefficient, but the principle of measuring the *purity of the leaf* remains the same.

Choose a maximum tree depth *max_depth*

While the depth of the tree is < *max_depth*:

- For each possible split value:
    - Predict the **majority class** for each leaf
    - Compute the **misclassification rate** between the predictions $\hat{y}_i$ and the true target values y_i
- Return the split value with the lowest **misclassification rate**

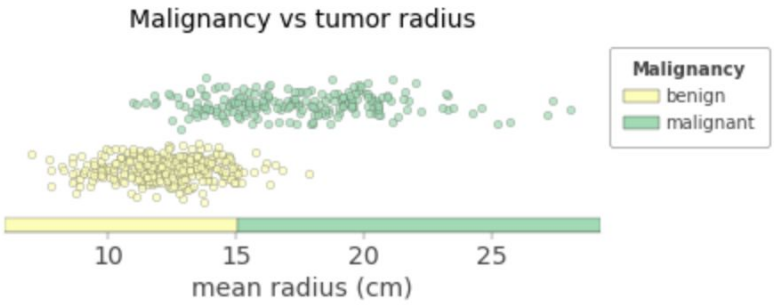
*Classification tree fitting algorithm. It is similar to the regression tree version but the Mean Squared Error minimization is replaced by the misclassification rate minimization (or the gini impurity that is roughly similar). Also, regression trees predict the mean of the target values contained in leaves, while the classification counterpart predicts the **majority class**.*

Another difference between regression and classification tree is that classification trees simply predict the majority class of leaves, while the regression tree predicts the average of the subset of samples belonging to the leaf.

After these introductory examples, from now we will use the real [Breast Cancer Wisconsin (Diagnosis) Dataset](https://scikit-learn.org/stable/datasets/index.html#breast-cancer-dataset).
(https://scikit-learn.org/stable/datasets/index.html#breast-cancer-dataset)
It contains 29 nuclei characteristics of 212 malignant masses and 357 benign ones.

mean radius	mean texture	mean perimeter	mean area	mean smoothness	...	malignancy
12.470	17.31	80.45	480.1	0.08928	...	1
13.870	16.21	88.52	593.7	0.08743	...	1
9.567	15.91	60.21	279.6	0.08464	...	1
16.300	15.70	104.70	819.8	0.09427	...	1
13.430	19.63	85.84	565.4	0.09048	...	0
14.710	21.59	95.55	656.9	0.11370	...	0
23.270	22.04	152.10	1686.0	0.08439	...	0
13.270	14.76	84.74	551.7	0.07355	...	1

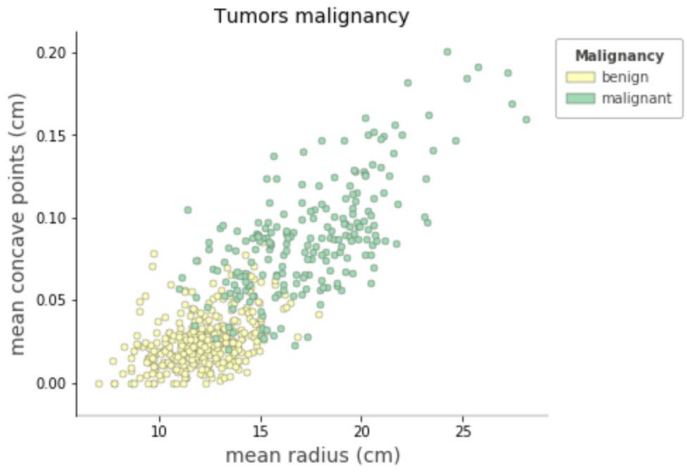
First columns and rows of the Breast Cancer Wisconsin (Diagnosis) Dataset. One the left are the 29 features of the cell nuclei contained in breast masses. The last column on the right is the target, it is 0 when the tumor is malignant and 1 otherwise, but we will reverse this convention from now.



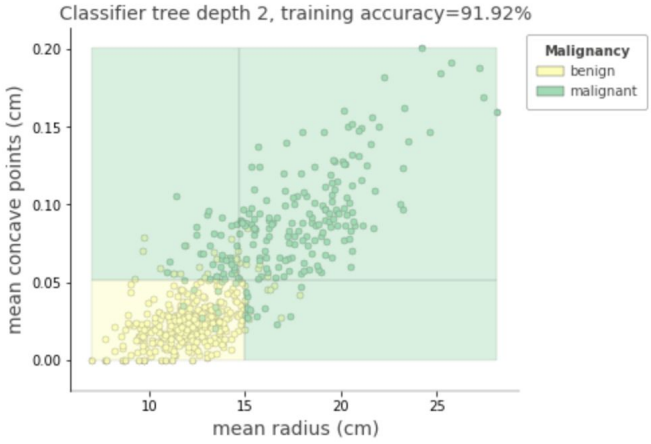
Visualization of the mean radius feature and the target (malignancy). A sensible decision boundary sits between 10 and 20 cm, but in order to precisely decide its value a machine learning model that minimizes the misclassification rate is best suited.

Note that in the value that minimizes the misclassification rate is found at 15 cm, it has an accuracy (i.e. 1 - misclassification rate) of 89 %. This single-split optimization is equivalent to classification tree with a maximum depth of one.

Using an additional predictor is likely to improve the model performance (i.e. its accuracy).

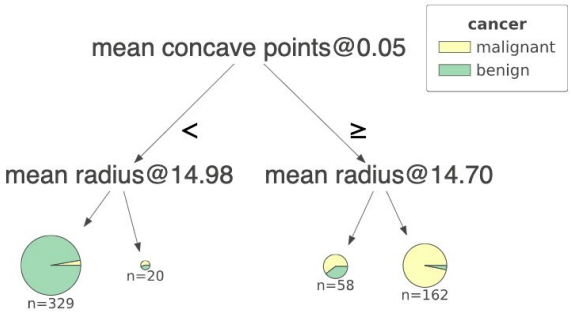


Tumors malignancy vs nuclei mean radius and mean number of concave points. The additional feature allows for more intricate splits, hence larger accuracies can be achieved, but beware of overfitting! (adding features tend to increase overfitting overall)



*Fitted classification tree with 4 leaves (maximum depth of 2). Using an additional features increased the training accuracy from 89% to 92%. Here only masses with small nuclei (radius <15 cm) **and** a mean number of concave points below 0.05 are classified as benign. Note that we haven't performed a train/test split and hence we cannot conclude on the generalization error improvement yet.*

A useful representation of the latter decision tree is given below:



Schema of a classification decision tree fitted on the breast cancer dataset. This tree maximum depth was set to 2, and the features are the mean number of concave points and the mean radius of the nuclei in the extracted breast mass. This tree performs a first split on the mean concave points features at 0.05, then a second one on the mean radius, the value of which depends on the branch. The number of samples selected by each of the leaves is shown below the pie charts. The majority class (dominant color of the pie charts) is predicted, i.e. benign is predicted for mean concave points > 0.05 ***and*** *mean radius < 14.98 , and malignant is predicted in all other cases.*

This representation is equivalent to the scatter plot shown above, and gives a better idea of why the split values and features were chosen by the tree fitting algorithm presented previously.

The Logistic Regression model

The logistic regression model is the linear regression equivalent of classification. Notice the misnomer here, the **logistic regression model is a classification model**. Of course there are good reasons for this name, but we will not need to delve into those details.

Before introducing the logistic regression model, we have to reformulate the problem with a numerical target, that is one when the tumor is malignant and zero otherwise.

Alternative representation of the tumor malignancy prediction problem with a single predictor. This representation is more suited to the explanation of the logistic regression model.

While the classification trees attempt to model the data with binary cuts, the logistic regression models attempts to model the **probability** that a given instance belongs to a given class (0 or 1). To model the probability is uses the so-called *sigmoid function*, the corresponding formula is the following:

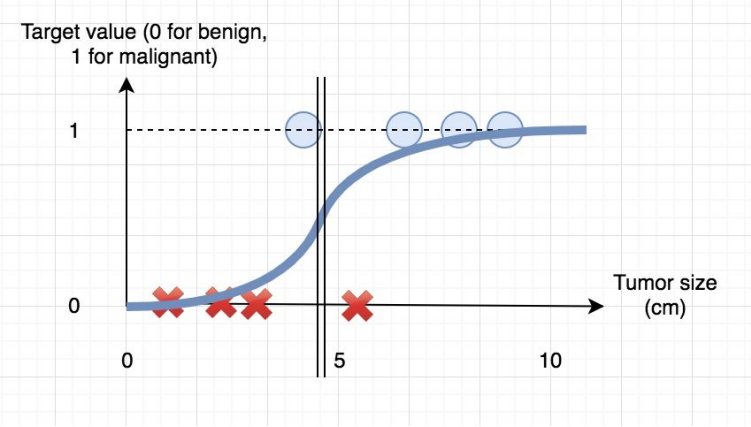
$$p\left(y|x,w,b\right) = \frac{1}{1+\exp[-(wx+b)]}$$

Here *y* is simply a symbol for the target (=1 for malignant tumors), *x* is the predictor (the tumor size in our current example), and just like for the linear regression *w* and *b* are the *parameters* or *coefficients* of the model.

Note that the sigmoid function used to model probabilities in the logistic regression model has the following properties:

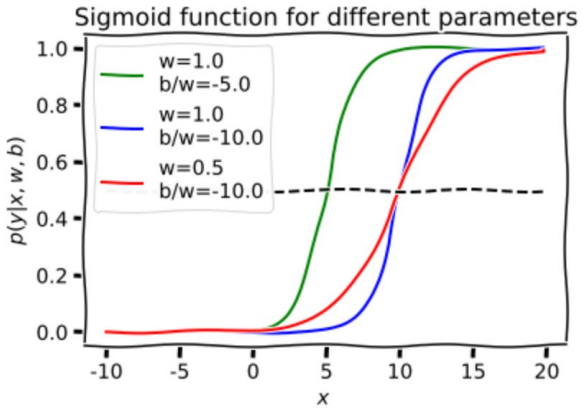
- It is between 0 and 1 (hence it can describe a probability)
- It has a natural decision boundary at $p = 0.5$, i.e. $x = -w/b$
- *b* controls the (horizontal) position of the separation line
- *w* controls the "slope" (degree of certainty of the prediction)

Here is a graphical representation of the logistic regression model, fitted to the example data presented before:

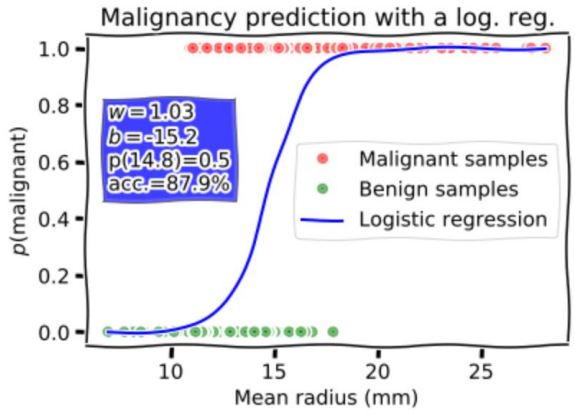


Graphical representation of the logistic regression model with the data points used to fit it. Notice the vertical boundaries of the model, 0 and 1, that makes it a good candidate for modeling a probability, and the horizontal decision boundary at a probability of 0.5.

To get a better intuition on the sigmoid function fitted by the logistic regression model, here are a couple of plots for given pairs of parameters *w* and *b*.



Sigmoid function used by the logistic regression model to predict malignancy probabilities. The decision boundary (*x*-value for which the probability is 0.5) is at *-b/w*, and the *w* parameter controls the slope of the function, i.e. its degree of certainty (compare the red and blue lines).



Example of a fitted logistic regression on the (real) breast cancer dataset, with associated coefficients and accuracy. For the logistic regression model the optimal values of the coefficients are found with the gradient descent method, a very general method that maximizes the likelihood of the data given the model.

The optimization method that provides the optimal coefficients for the logistic regression is a bit more intricate that the tree growing method, or the normal equations for the linear regression. The objective function is the likelihood function that has to be maximized in order to get a model that best mimics the data. In fancy mathematical terms, the *w* and *b* parameters are given by:

$$(\hat{w}, \hat{b}) \in \left\{ \arg\max_{(w,b)\in\mathbb{R}^2} p(\{y_i, x_i\} \mid w, b) \right\}$$

The method of choice to maximize the likelihood is the so-called **gradient descent algorithm**. It will not be presented here, but feel free to explore the references at the end of this chapter on this topic.

Metrics for classification tasks

The problem with the accuracy

There is a very natural performance metric for classification tasks: the accuracy. The accuracy is the number of correct predictions divided by the number of predictions.

The problem with the **accuracy** is that it is **only meaningful when the target classes are balanced**, i.e. when you have as many samples corresponding to class A as samples corresponding to class B, class C, ... In our tumor malignancy (binary) prediction tasks it would mean that you have 50 % of malignant tumors and 50 % of benign ones. It is not the case in general. You could throw away part of the majority class examples to have a balanced dataset, but you would lose valuable examples that would help your model improve.

To understand why the accuracy is not meaningful in the unbalanced case, consider the following situation: you are training a machine learning algorithm to spot fraud detection from credit cards transaction data. You may be happy with a 90 % accuracy, but imagine that in your dataset 99 % of transactions are regular. Hence, a dummy model that always predicts the "regular" class has a 99 % accuracy.

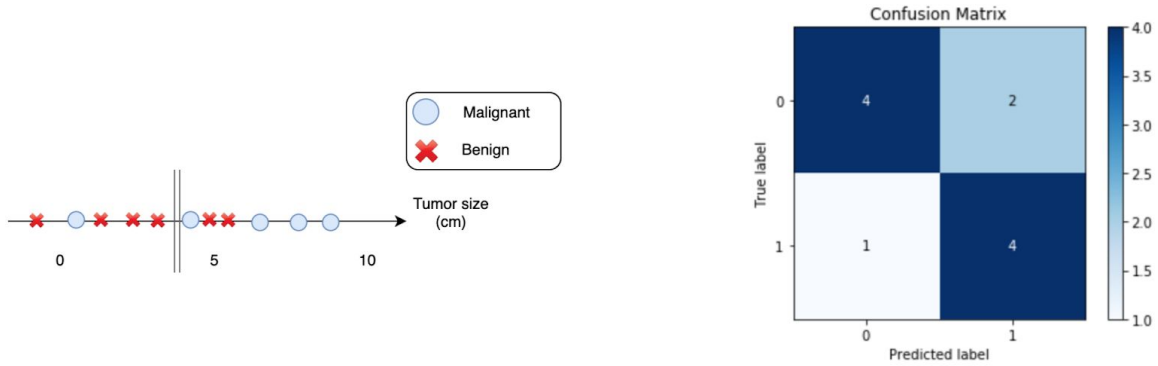
The solution to this problem is to **use different performance** metrics in the unbalanced case.

Accuracy also suffers from another problem: you have to fix the decision boundary of your algorithm in order to compute it. For example the logistic regression model outputs probabilities between 0. And 1. Although 0.5 seems like a natural decision boundary, you may want to lower this threshold for applications where False Negatives should be avoided (e.g. not mistaking a malignant tumor for a benign one), or increase it where False Positives are more problematic (e.g. for a spam detection algorithm).

There are metrics that evaluate the performance of a classification model without having to fix the decision boundary, hence describing the global performance of the model.

The Confusion Matrix

Rather than aggregating the model performance into a single number, a good understanding of your model mistakes (given a decision threshold) is provided by the confusion matrix. The latter simply show the True Positives, False Positives, False Negatives, and True Positives predictions of your model in a readable way.



(Left) Tumor toy dataset, with a prediction threshold, and (Right) corresponding Confusion Matrix.

Precision and Recall

A model capability to avoid False Positives and False Negatives are often described by the Precision and Recall.

The Precision is the number of True Positives divided by the number of positive predictions. The closer it is to 1, the less False Positives.

Conversely, the Recall is the number of True Positives divided by the number of Positive examples. The closer it is to 1, the less False Negatives.

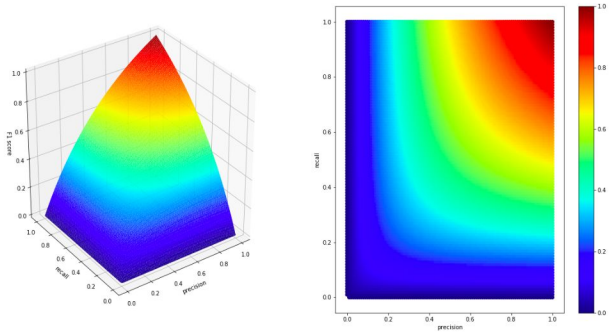
F1-score and ROC-AUC

There are two main metrics that take into account model imbalance and give a somewhat universal way to evaluate models.

The first one is the F1-score. It is the harmonic mean of Precision and Recall

$$F_1 = \left(\frac{\text{Precision}^{-1} + \text{Recall}^{-1}}{2} \right)^{-1}$$

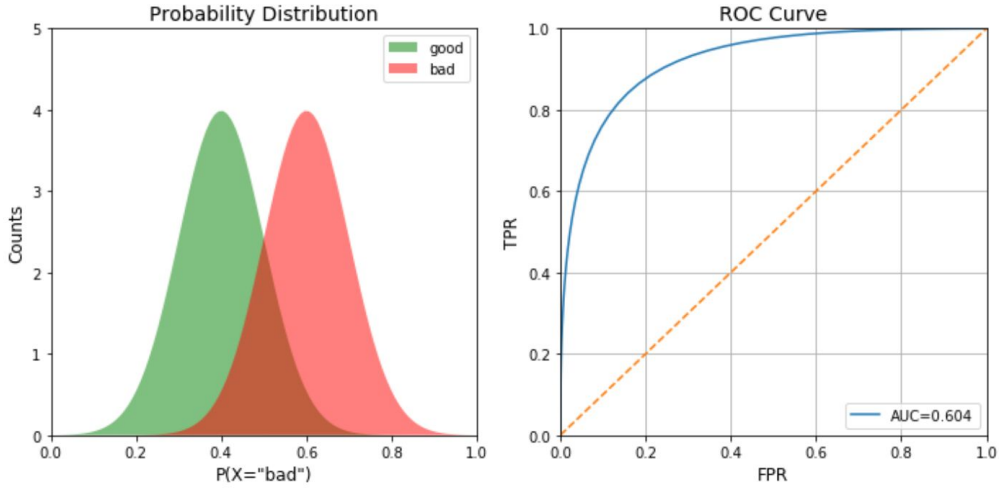
The harmonic mean is similar to the standard arithmetic mean, in the sense that it is larger when the elements being averaged are larger, but conversely to the arithmetic mean it increases if the precision and recall have close values (for a given arithmetic mean). Thus the F1-score favor large precision and recalls, but penalize having one of them being much larger than the other one. A large F1-score means that you manage to minimize both False Positives and False Negatives, without one compensating for the other.



F1-score vs Precision and Recall. Note that even if one of the Precision and Recall is large, the F1-score decreases quickly if the other one becomes smaller.

The F1-score is suitable for imbalanced datasets, but expects a model with a fixed decision threshold. The next metric we present does not suffer from this limitation.

Before presenting ROC-AUC metric itself, let's describe what the Receiver Operating Characteristic (ROC) is. Do not try to guess from the name, it is mostly historical.



Receiver Operating Characteristic curve. This parametric curve (on the right) plots the performance (False Positive Rate and True Positive Rate, a.k.a. Recall) of the model for a varying decision threshold. The top right point of the curve means that when the False Positive Rate is 1 (no False Positive) then the True Positive Rate (a.k.a. Recall) is also 1, because we never predict the positive class.

Conversely, when the False Positive Rate is 0 the model never predicts the positive class, and then the True Positive Rate is also 0. In between, the fact that the blue curve has a point at (FRP=0.4, TPR=0.9), for instance, means that when the threshold is such that the False Positive Rate is 0.4, then the True Positive Rate is 0.9. The closer the curve to the top left corner the better, because the top left corner corresponds to an ideal situation in which the True Positive Rate is 1, and the False Positive Rate is 0. The (fictive) corresponding model is represented on the left, with the red distribution being the positive class distribution (a.k.a "bad" here) and the green distribution the distribution of the negative examples. They are plotted as a function of the model predicted probabilities.

The bottom-line about the ROC-curve is that it represents the model performance for all possible thresholds, thus computing the Area Under this Curve (AUC, hence the name ROC-AUC) gives a single number that describes how the model performs on average for varying decision threshold. Note that the ROC curve can be generalized to the multi-class task, by treating it as an ensemble of binary classification tasks.

6B Exercises

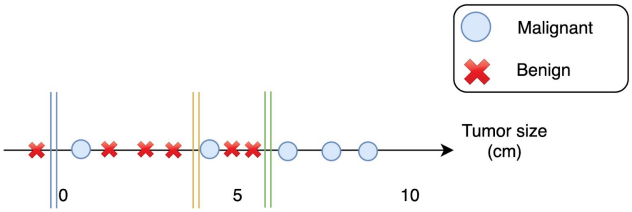
1. The False Positive / False Negative dilemma

Consider the following examples and decide if you are more willing to accept False Positives or False Negatives

1. Tumor malignancy prediction
2. A spam detection algorithm
3. A pregnancy test
4. A court judgment

2. Count your model mistakes

Compute the number of True Positives, True Negatives, False Positives and False Negatives in the following example, for the 3 decision boundaries (blue, yellow, green).



Homework: Finish the notebook exercises

Everything is in the title 😉.

Further Reading

1. *Prashant Gupta, 2018, Decision Trees in Machine Learning:* https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

2. *Caglar Subasi, 2019, Logistic Regression Classifier:* https://towardsdatascience.com/logistic-regression-classifier-8583e0c3cf9

3. *Sarang Narkhede, 2018, Understanding Logistic Regression:* https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102

4. *ML-Cheatsheet, 2017, Logistic Regression:* https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

5. *Niklas Donges, 2018, Gradient descent in a nutshell:* https://towardsdatascience.com/gradient-descent-in-a-nutshell-eaf8c18212f0

6. *Jocelyn D'Souza, 2018, Let's learn about AUC ROC Curve!:* https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152

Week 7A

Introduction to Python programming

What is Python?

Python is the main programming language of data science. Conversely to SQL that has been presented before, it is an imperative language (i.e. with explicit instructions like *if* statements and *for* loops). It is easy to get started, but possible to code rigorously. It used to be considered as "slow" but it is not the case anymore (there are now Python frameworks that rely on C++ compiled code for performance critical parts).

Python has the following characteristics:

🐣 **DOB: 1990** (creator: Guido van Rossum)

🐍 **Current Version: 3.7** (Python 2 is virtually not used anymore)

🕴 **Interpreted language** (≠ compiled language): code read & executed sequentially

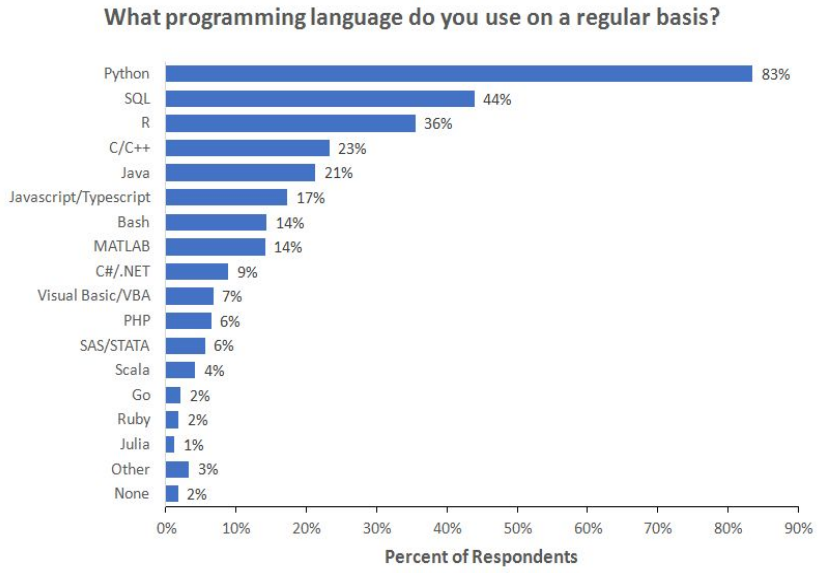
🏔 **High-level**: developer handles concepts rather than machine instructions

⏩ **Dynamically-typed**: variables types are not predefined and can be changed

📚 Supports several (intertwined) programming paradigms:

- **Procedural**: define functions that transform data
- **Object-oriented**: organize your code in objects carrying data and functions
- **Functional**: write functions that are meant to be transformed like data

As you can see on the following figure, with SQL Python completes the data scientist toolbox in terms of programming languages.



Data scientists' preferred programming languages, from CustomerThink (7/5/2019) (http://customerthink.com/programming-languages-most-used-and-recommended-by-data-scientists/)

Indeed, Python is the most widely used programming language among data scientist, and has the following prize list (as of 2019):

- Top programming language in terms of tutorial searches on Google (PYPL Popularity index)
- 3rd in terms of GitHub and StackOverflow popularity (RedMonk ranking)
- 4th one in terms of engineering usage (TIOBE Index)

When writing Python code, as in most programming language the developer actually uses libraries and frameworks more than the basics of the language (that she has to know nevertheless).

Scripts and notebooks

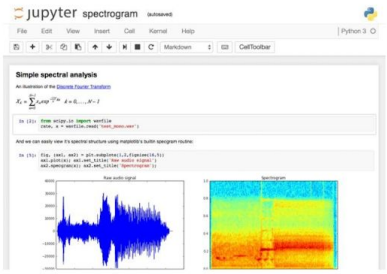
When coding in Python you can use regular scripts (i.e. text files that have to be interpreted and run), notebooks (interactive coding environments), or a combination of the two. The former has the advantage of being modular (naturally broken into complementary pieces) and more suited to a production environment. On the other hand, notebooks offer flexibility, and a convenient way to explore data and iterate fast in the exploration phases. Both are widely used by the data science community.

Scripts

- Set of text files
- Executed in a terminal, via an IDE, or constantly running on servers (APIs)
- Suitable for production code

Notebooks

- Interactive running "kernel"
- The kernel runs on the local machine or distant servers
- Nice display: can make great presentations, dashboards, or blog post
- Suitable for exploration

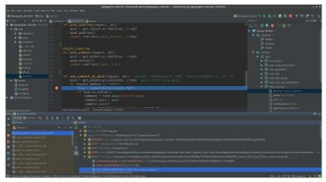


When writing and running Python scripts, it is better to use an Interactive Development Environment (IDE) than working in the terminal. IDEs offer advanced code inspection features (spot indentation error, autocomplete variables, show functions documentation, propose code "snippets", ...), a convenient overview of your project, and useful debug tools. The following figures shows the four main IDEs, but there are many more that have their advantages and drawbacks.

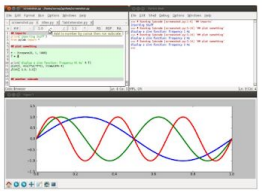
Main Python IDEs

Fully equipped IDES

PyCharm: the leader



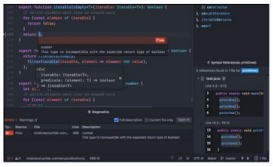
IDLE: a simple alternative



Highly customizable text editors

VSCode: new extensions every day

Atom: simple but highly customizable



Note that IDEs can also handle Jupyter notebooks, either by default or with extensions.

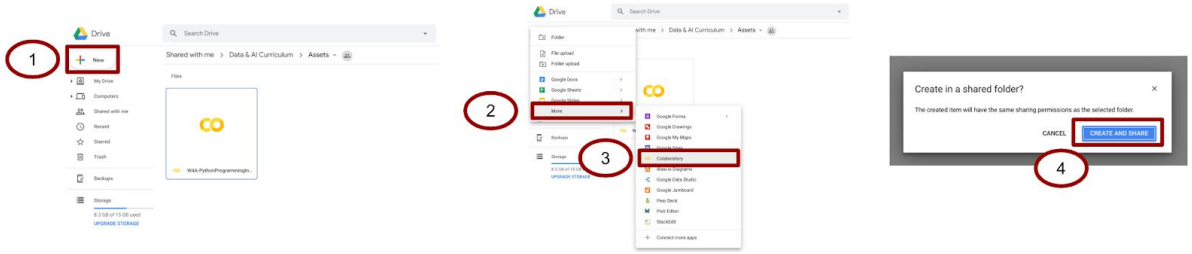
Python in practice with Colab

In this class we will run Jupyter notebooks through Colab (https://colab.research.google.com) , a nice cloud environment by Google that does not require any set up, and offers an environment that already contains all the libraries we will need.

Using a notebook for teaching Python has the advantage of offering nicely displayed "Markdown" cells for explanations, being interactive, and allowing to design exercises with hidden solutions.

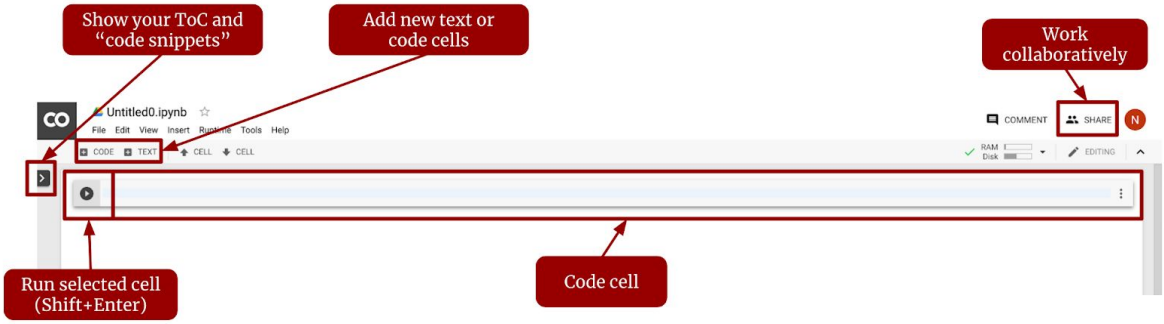
To use Colab you need a Google account (http://www.google.com) that you can create for free.

To create a new notebook in your Google Drive, first open your Google Drive in a browser (drive.google.com), then click on *New → More → Colaboratory → CREATE AND SHARE*.



How to create a Jupyter notebook on Google Colab from a Google Drive.

After hitting *CREATE AND SHARE* you should see the following screen.



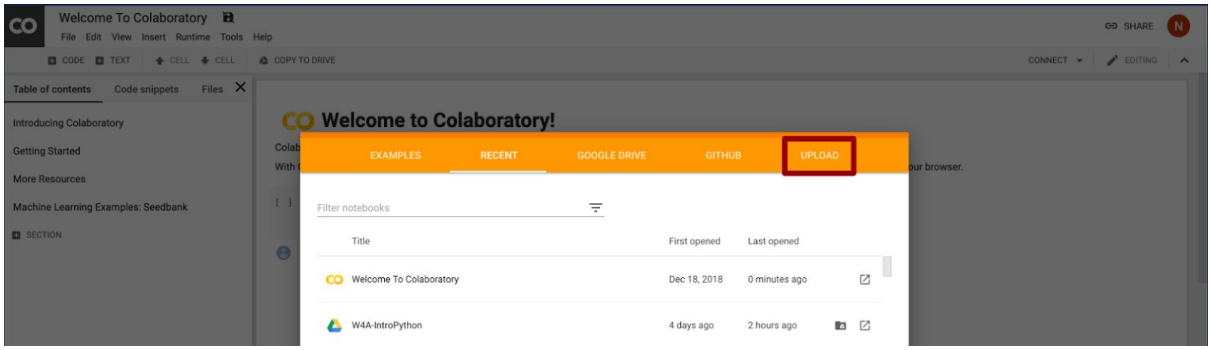
Jupyter notebooks in Colab.

You are ready to run your first Python instructions! To do so, select the first and only cell, write any valid Python instruction (e.g. "1+1") and hit *Shift+Enter* to run the cell.

After a few seconds (when you run the first instruction the "kernel" that will run your code is created, which is why it takes a bit of time) you should see the output below the cell.

To start your learning practice,

1. Open the first Python notebook tutorial here (https://colab.research.google.com/drive/1MWIccL2_CDtsjnstBX3QI9cdi_CvOl4-)
2. Save the notebook on your laptop ("*File*" → "*Download .ipynb*")
3. Open Colab: https://colab.research.google.com
4. Hit "*UPLOAD*", then choose the downloaded notebook and start learning!



The Colab home page, UPLOAD the first tutorial notebook and start learning. 👩‍🎓

7A Exercises

Homework: Finish the notebook exercises

Everything is in the title 😉.

Homework: Loops and strings

Do the following challenges on HackerRank:

1. Python Loops: (https://www.hackerrank.com/challenges/python-loops/problem)
2. Find the Runner-Up Score!: (https://www.hackerrank.com/challenges/find-second-maximum-number-in-a-list/problem)
3. Find a string: (https://www.hackerrank.com/challenges/find-a-string/problem)

Further Reading

1. *"Python for absolute beginners" class on Udemy (4h):* https://www.udemy.com/python-for-absolute-beginners-u/
2. *"Python Masterclass for beginners" (6h) on Udemy :* *https://www.udemy.com/python-masterclass-for-beginners/*
3. *The Python for data science notebook (gumption's GitHub):* *https://nbviewer.jupyter.org/github/gumption/Python_for_Data_Science/blob/master/Python_for_Data_Science_all.ipynb*
4. *A gallery of interesting Jupyter Notebooks (on GitHub)*: https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks
5. *Ehmattes' Introduction to Python Programming notebooks:* *https://nbviewer.jupyter.org/github/ehmatthes/intro_programming/blob/master/notebooks/index.ipynb*
6. *Exploratory computing with Python, with live exercises:* http://mbakker7.github.io/exploratory_computing_with_python/
7. *A nice, thorough introduction to Python in Colab:* https://colab.research.google.com/github/ondrolexa/r-python/blob/master/01-Introduction-to-Python.ipynb#scrollTo=XPHyWp9ZxpvO
8. *Get started with Colab:* https://colab.research.google.com/github/jckantor/CBE20255/blob/master/notebooks/Getting_Started_with_Jupyter_Notebooks_and_Python.ip

Week 7B

Python for Data Science

This class is dedicated to the main library for data science in Python: Pandas. Pandas can efficiently perform data integration, manipulation, and plots.

Before we dive in, let's explain what are libraries and frameworks in computer science.

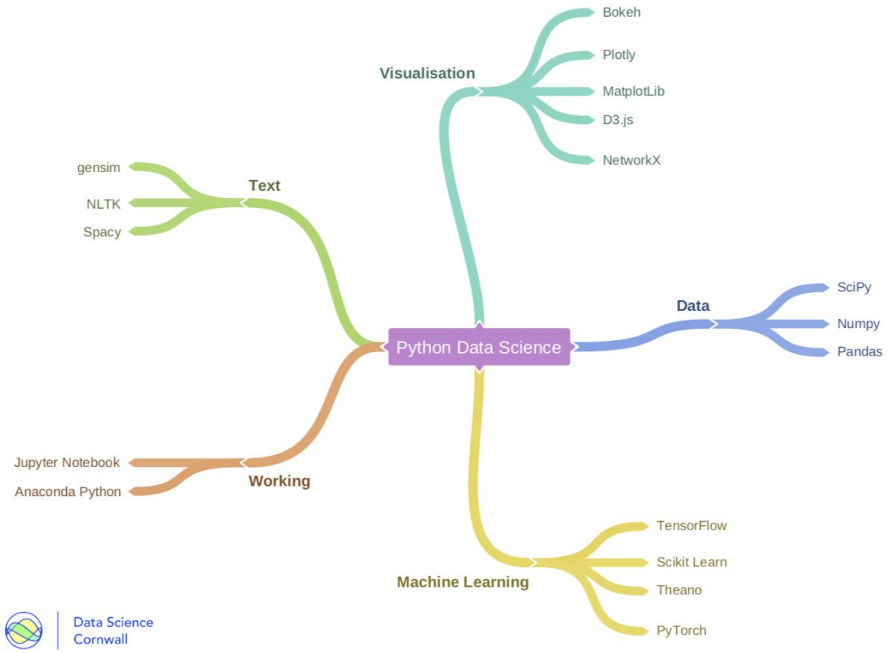
Libraries and frameworks

Libraries and frameworks are both bundles of functions and classes.

- They are imported with the ***import*** keyword in Python.
- When becoming a **developer, you spend more time using libraries** specific to your domain than dealing with the built-in features of the language.
- **Libraries contain utilities** that can be used here and there in your code without changing the global structure. **Frameworks are more greedy**: when using a framework you have to enter its world fully, a process called **inversion of control**.
- Python contains **built-in libraries**: *datetime*, *multiprocessing*, *html*, *json*, ... You can also use **third-party libraries** like *numpy* (linear algebra), *scikit-learn* (machine learning) or *pandas*. Those libraries are easily installed with the *pip* utility, but most of the ones you will need are already installed on Colab. Finally, you can build **your own library** and re-use it in other programs!

Main data science libraries and frameworks

The following figure shows an overview of the main data science libraries in 2019.



Main data science libraries, from [SoftwareCornwall](https://www.softwarecornwall.org/) (https://www.softwarecornwall.org/) (5/10/2019). Note that web development frameworks like Flask and Django may be added to this list, as they allow to create dashboards, or APIs for machine learning models.

These libraries and frameworks can be categorized as:

- **Jupyter Notebooks** that allows you to code and explore data very quickly and efficiently (more on this later),
- **Data Manipulation** libraries like Pandas that we will learn in this course,
- **Machine Learning** libraries like scikit-learn that we will learn in a future class or the TensorFlow for Deep Learning (a famous subset of machine learning),
- **Natural Language Processing** libraries to analyze textual data and build machine learning algorithms for it, and finally
- **Data Visualization** libraries to plot simple graphs (matplotlib), interactive visualizations (Plotly, Bokeh, and D3.js) or network graphs (NetworkX).
- **Web development** frameworks (Flask and Django mainly) that allow creation of dashboards, or APIs for machine learning models.

Among these numerous utilities, we can highlight the few leaders.

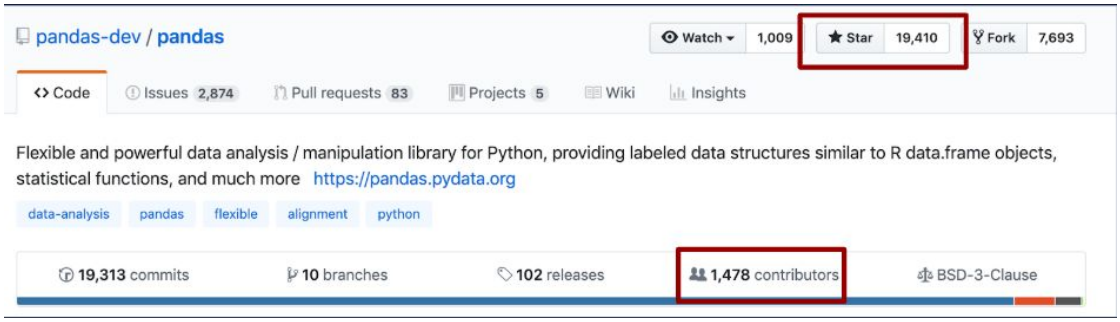
matplotlib	Matplotlib is a very customizable plotting library, old but still dominant
pandas	Pandas is the unavoidable data handling library, can plot data too
scikit learn	Scikit-learn is the main machine learning library
TensorFlow	Tensorflow is the leading deep learning framework, but PyTorch catches up
dj Flask	Django and Flask are the two principal web development frameworks, Useful to build dashboards, or machine learning APIs

In this course, we will learn Pandas and Scikit-learn. These two libraries are the daily bread of most data scientists.

Python libraries are countless. Apart from this course tips, and top ranking found in blog articles, it is sometimes difficult to decide if you should start using a library or another.

Since most libraries in Python are open source, meaning everybody can read their source code and access their GitHub page. GitHub is the main code hosting service for backup, version control, collaborative coding, and projects presentation. The main page of every project shows a number of stars given by other developers, as well as other information like the number of contributors. When a project has many contributors and stars you can safely anticipate a good maintenance and reliability of the corresponding library (i.e. less bugs, more explicit exceptions, more readable functions documentation, ...).

For instance, here is the GitHub home page of the Pandas project:



The large number of contributors and stars puts this library in a leading position for data manipulation.

Data manipulation and plotting with Pandas

Pandas is a library that is a great asset for any data scientist, analyst, or wrangler. A few of its features:

- **Main data manipulation library** in Python
- Central object: **DataFrames** (= tables)
- Can import and export (almost) everything
    - **Flat files**: csv, excel, json
    - **Databases**: SQL
- Can perform data transformation and merge, plots, descriptive statistics, preprocessing for machine learning

	name	city	grade
0	Anita	Sausalito	3.9
1	Bobby	San Francisco	4.1
2	Cindy	Oakland	4.8
3	Dennis	Berkeley	3.7

Example of a pandas DataFrame object. DataFrames are the central objects in pandas.

To start learning pandas:

5. Open the corresponding Python notebook tutorial here: https://colab.research.google.com/drive/1WoqLMnzcqedukxYz3Z9HbWwpR131KCNU
6. Save the notebook on your laptop ("*File*" → "*Download .ipynb*")
7. Open Colab: https://colab.research.google.com
8. Hit "*UPLOAD*", then choose the downloaded notebook and start learning!

7B Exercises

---

1. Pandas practice exercises

Try this excellent set of exercises from Guipsamora's GitHub repository (https://github.com/guipsamora/pandas_exercises) (corrections included).

Homework: Finish the notebook exercises

Everything is in the title 😉.

Homework: Attention dataset

Open a notebook on Google Colab and import the "Attention" DataFrame from the Seaborn library, with the following code:

```
from seaborn import load_dataset

df = load_dataset("attention")
```

1. Describe the dataset
2. Compute the average score for subjects who are their attention "focused"
3. What is the id of the subject with the highest score among subject with attention "divided" ?

Further Reading

1. *Big Data Made Simple (2019), Top 20 Python libraries for Data Science:* https://bigdata-madesimple.com/top-20-python-libraries-for-data-science/
2. *Pandas Cheatsheet (becominghuman.ai):* https://becominghuman.ai/cheat-sheets-for-ai-neural-networks-machine-learning-deep-learning-big-data-678c51b4b463
3. *Python for Data Analysis (William McKinney, O'Reilly):* http://shop.oreilly.com/product/0636920023784.do
4. *Kinsta Knowledge base (2018), What is GitHub?:* https://kinsta.com/knowledgebase/what-is-github/
5. *Pandas documentation:* https://pandas.pydata.org/pandas-docs/stable/
6. *Jay Alammar , 2018, Visualizing Pandas' Pivoting and Reshaping Functions:* http://jalammar.github.io/visualizing-pandas-pivoting-and-reshaping/

Week 8A

Machine Learning Review and Big Data

Machine Learning Review

As a Product Manager, it is unlikely that you will have to implement and train machine learning on your own, but it is important to remember how Machine Learning work, what are the different types of tasks, and how to evaluate the performance of a machine learning model.

This section gives a review of the topics presented previously in this course.

Speech to text

Image recognition

Automated translation

Recommender systems

Autonomous cars

And many others...

A few applications of machine learning. Those are the most famous ones, but Machine Learning is also ubiquitous in companies marketing departments, recommender systems, and planning systems.

An important concept we have covered in this course is data preprocessing. Remember that preprocessing is as important as the modeling itself, or even more. It impacts the performance of the model itself, as well as the analysis you may perform on your data.

The main data preprocessing tasks are the following:

- **Data ingestion**

- Handling **missing values**
- Transform **categorical data**
- **Normalize** your data,
- Perform **feature engineering** to create contextualized variables or ***derived features***

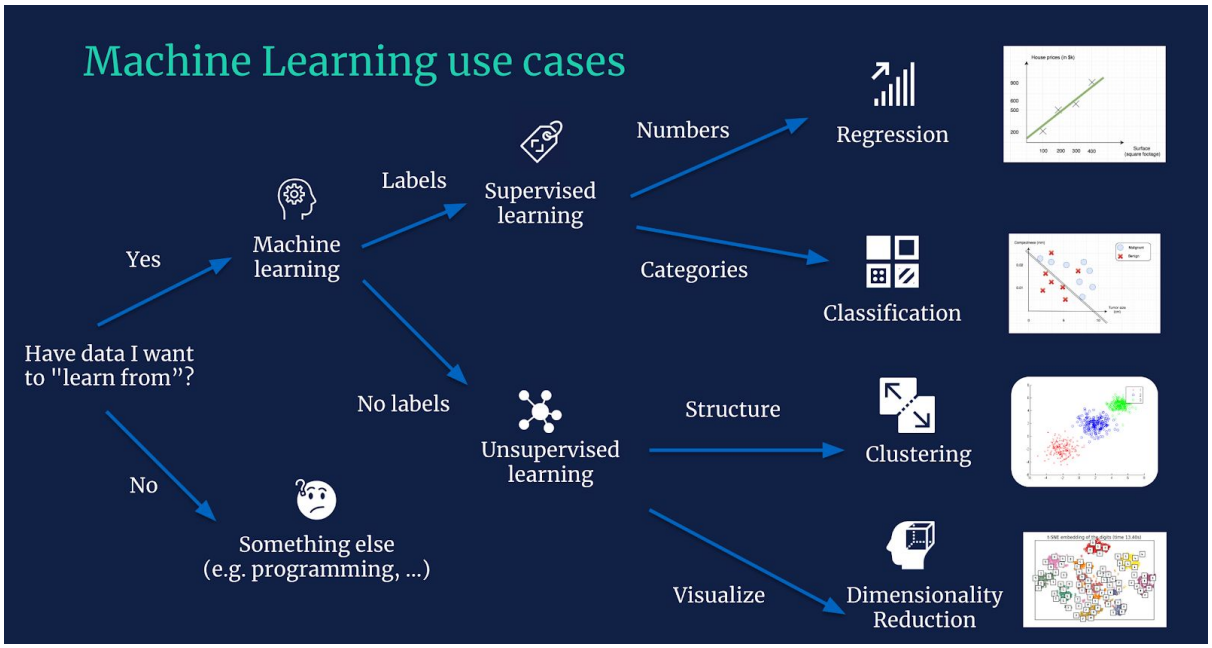
Something we have not presented yet is what Machine Learning can do, and what it cannot.

You should have a good sense of the possibilities offered by ML, but here is a non exhaustive list:

- **Forecast**: for example ML can predict the amount of a given product that will be ordered next month. It has important applications in logistics.
- **Memorize**: A machine learning model can find patterns in the data and memorize them, for instance remembering that users who visited the page A of a website before page B are more likely to click on a given ad.
- **Reproduce**: Machine Learning models allow to automate some tasks, such as the ones handled by chatbots those days. The underlying models learn to reproduce human answers from a large set of conversations examples.
- **Choose the best item for a given customer**: in the classification task that we have extensively covered, we can predict categories from a customer's behavior, for instance the item this client will buy next.

On the flip side, Machine Learning models have **limitations**: they are mostly **limited to reproducing datasets** given, and **require a lot of data** to be able to do so. Also, Machine Learning models have no cognitive ability to generalize to new tasks, meaning that they have to learn specific tasks.

The tasks tackled by Machine Learning models are numerous, but the main ones have been presented in a previous chapter. The most important and ubiquitous ones are supervised learning tasks: regression and classification.



Main Machine Learning tasks. This course focuses on supervised learning, i.e. guided predictions of quantities (regression) or categories (classification).

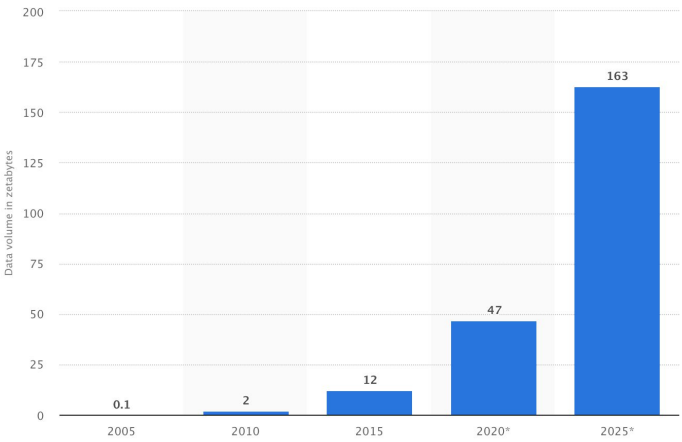
Machine Learning is a vast field in constant evolution. If you want to go further in ML here are the directions you may want to follow next:

- There are many **more algorithms than trees, linear regressions and logistic regressions for classification and regression** (tree-based: gradient boosting and random forests, neural networks, SVMs, ...). All of them have their specifics and can mostly be learnt separately. Nowadays deep learning models (artificial neural networks) are very popular for their versatility and accuracy at certain tasks. They require a good understanding of machine learning principles and demand a sizable commitment, but the principle you have learnt in this course remain for these fancy models, and you can get a good understanding of how they work if you are willing to dedicate some time to it.
- We have not covered **unsupervised learning** or **reinforcement learning**. Unsupervised learning is not much more complex than supervised learning in general, but are harder to leverage for companies. They are mostly used conjointly with human experts as guides or for analysis purpose. Reinforcement learning is a different topic that may explode in the coming years, that is closer to human learning that standard machine learning models, as it includes a concept of environment with which the model can interact. This is why reinforcement learning is widely used in robotics, or to create AI in video games.
- **Further learning resources are provided in the handbook** for all of the topics we have covered. Make sure to understand all those basics concept well, as they are underlying all modeling attempts by Machine Learning.

Big Data

Although big data is not only about the sheer amount of data, it is basically the ability to know which few numbers to look at.

First, the amount of data generated and stored in the world is growing exponentially. Also, the revenues generated by big data are expected to double in less than 6 years.

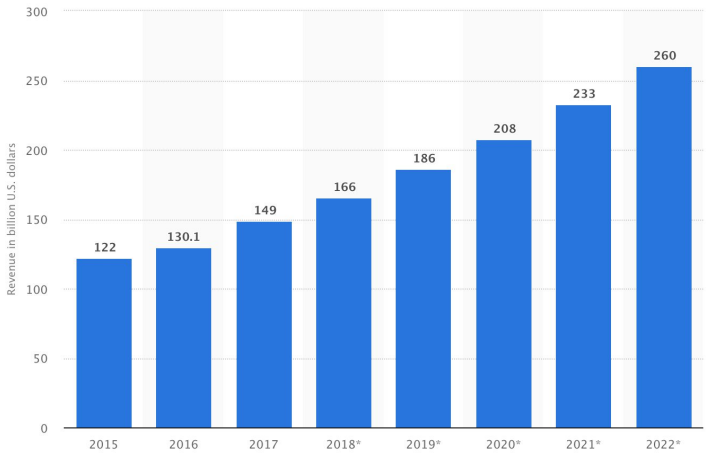


Volume of data generated and stored worldwide, in zetabytes (1,000,000,000 terabytes). The exponential growth is obvious.

The amount of data available has grown tremendously over the last few years, mostly due to three factors:

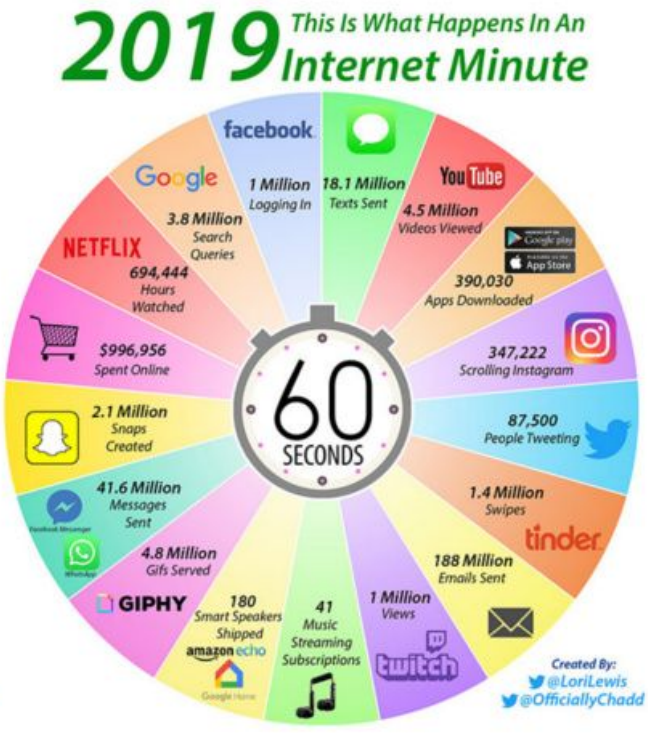
- Technological advances that made storage very cheap.
- New computer science development such as MapReduce that enable to store and retrieve large amounts of data efficiently.
- New possibilities to leverage data with machine learning.

While volume does not necessarily mean value, in the case of Big Data the two go hand-in-hand.



Revenues related to Big Data and Business Analytics in the world (in billions of dollars). By 2022 the revenues are expected to double compared to 2015.

In order to understand the causes of this surge in the amount of data available in the world, the following graph gives an idea of the amount of data that is created through platforms we all know.



What happens on the internet in a minute. This pie chart gives an idea of what kind of data is constantly generated by users. For instance, in a single minute almost 4 million queries are run on Google, 40 million messages are sent via whatsapp, and 4 million videos are watched on YouTube. All this data is stored by the corresponding companies, as data is the new gold due to the immense possibilities of content personalization and targeted advertising it offers. Picture from Jeff Desjardins, 2019, What Happens in an Internet Minute in 2019? (6/2/2019) (https://www.visualcapitalist.com/what-happens-in-an-internet-minute-in-2019/)

Although the growth in the amount of data in the world is impressive, volume is not the only characteristics of this new information science paradigm called "Big Data". A common definition is given by the "5 V's":

- We already mentioned the **Volume** that is obviously inherent to Big Data
- The **Velocity** at which data flows between servers, web browsers, connected devices, smartphones, ... and humans has also dramatically changed over the last few years
- The **Variety** of data being stored and proposed is new as well. While most data used by companies a decade ago was mostly tabular, now a single company as Facebook combines images, videos, texts, and social network information in order to understand its users.
- An overlooked aspect of a successful Big Data approach is the **Veracity** of data. It may sound trivial but a bigger data flow is harder to clean and monitor, thus leading to unreliable data points being stored on companies' servers. The veracity of data must be constantly assessed to ensure the usability of the corresponding information.
- Finally, storing everything is not necessarily a good idea, and one has to make sure the stored data has **Value** for the business.

📝 Some definitions of Big Data only include the first three V's, and some others add more concepts to this same definition, do not be surprised if you hear about the 3 V's or the 7 V's.

The history of Big Data

We will not go through a complete history of the development of Big Data tools in this class, and refer the interested reader to the literature proposed in the handbook, but we review quickly the development of the main Big Data tools here: Hadoop and Spark.

Hadoop was developed mainly by Doug Cutting when he was working for Yahoo, then for Google. The goal of Hadoop was to be able to store and manage large amounts of data while avoiding "**vertical scaling**". Vertical scaling consists in accommodating additional data by using more performant hardware, on the same machine. This approach is costly and is limited by the state of the art of computer architectures. The latter cannot keep up with the exponential growth of data.

Thus **Hadoop** and its inner algorithmic machinery (**MapReduce**) focuses on "**horizontal scaling**". Horizontal scaling consists in using large amounts of cheap computers ("commodity hardware") instead of one, limited super-computer. The challenges stemming from hosting a database on numerous computers ("Data nodes") are cunningly solved by the Map Reduce algorithm. Without MapReduce and its implementation in the Hadoop framework, it would be impossible to store, manage and retrieve large datasets.

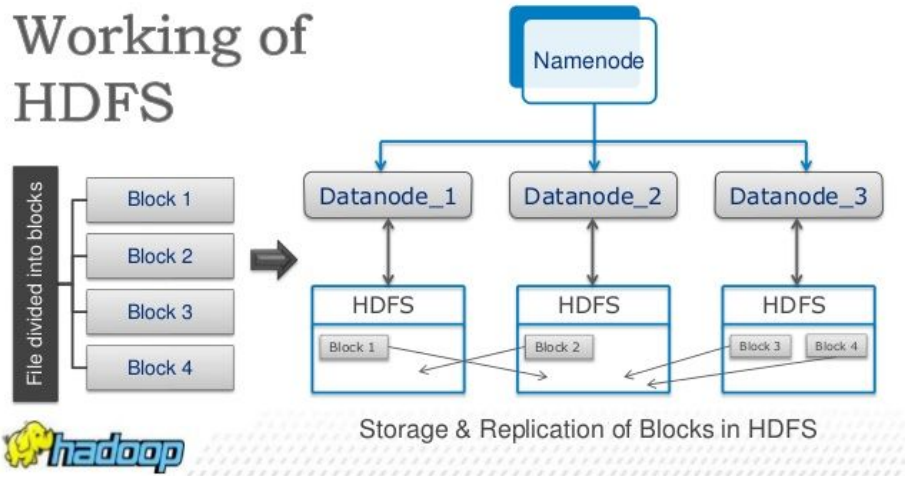
Hadoop was still facing performance issues due to the large amount of data transfers between nodes in the computers' network, and notably the time it takes to write and read hard drives. To solve this problem, a team in Berkeley developed **Spark**.

Spark is now the **main framework for Big Data clusters**. It relies on Hadoop and MapReduce, but adds crucial optimizations, and cannily keeps the data in the cluster shared memory to avoid the hard drive read and writes issues mentioned above. The development of Spark started in 2009, but it reached its full maturity in 2014, and is constantly being upgraded now.

Hadoop mechanisms: MapReduce and HDFS

Without diving in details, we give an overview of the inner workings of Hadoop: the MapReduce algorithm and the Hadoop Distributed File System (HDFS).

First the file storage in Hadoop is handled by HDFS. Since Hadoop assumes the use of commodity hardware, it replicates the data (typically 3 or 5 times) on several nodes of the computers' network. The data is said to be **distributed** across the network. Thus, if a node of the network fails (being unreachable because of network issues, or out-of-order because of its internal hardware failure), the data is not lost. HDFS is the software component that keeps track of the data location in the network, and re-creates the copies when a node fails.

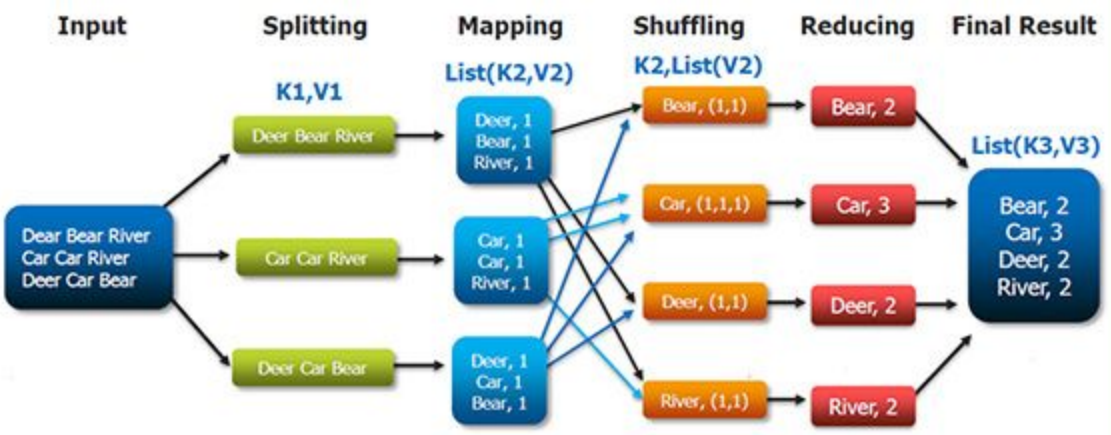


Picture from Shrey Mehrotra, 2014, Introduction to Hadoop and HDFS (6/2/2019) (https://www.slideshare.net/shreymehrotra/introduction-to-hadoop-and-hdfs-42994585)

Note that even if the network is composed of high-performance hardware, the large size of Big Data clusters makes it very likely that at least one node fails every single hour, hence the use of Hadoop remains highly relevant.

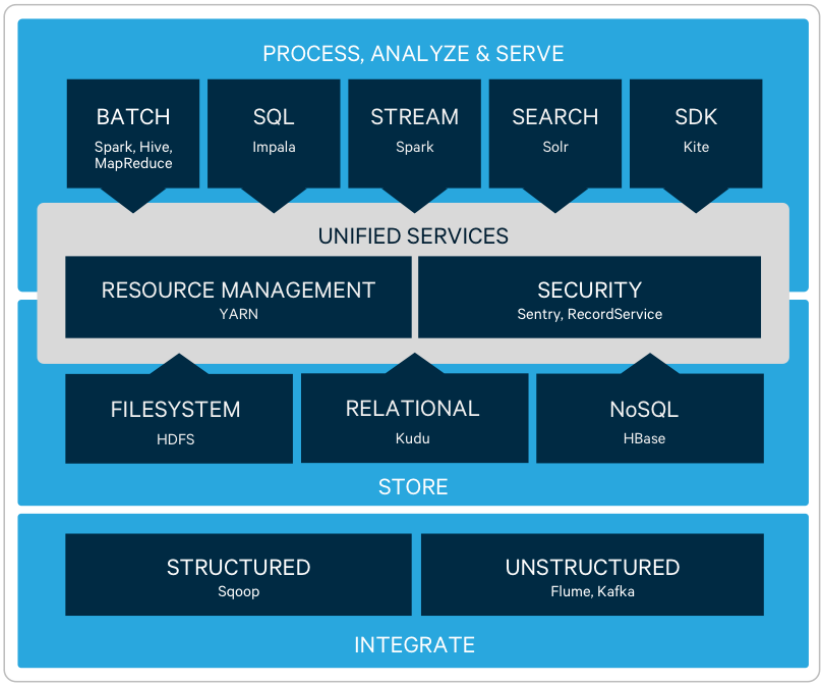
After the distributed file system that allows to guarantee the accessibility of large amounts of data, the second component of Hadoop is the MapReduce algorithm. The latter provide new methods to process the distributed data stored Hadoop clusters.

Here is an example of a simple task performed by **MapReduce** on a Hadoop cluster: a word count. Our goal here is to count the occurrences of each word appearing in the file stored on the Hadoop cluster, or in a subset of its data. While in a standard setting one would simply transfer the content of all the relevant files into a single computer's memory, on a Hadoop cluster a single node cannot contain all the data for most tasks. Thus, the data is split across several nodes. Then each node compute the occurrences of the words in the chunk of the relevant data that it stores. The nodes share the results of the individual counts in by assigning nodes to the counts of some words. Finally, during the reduce operation the nodes calculation results are merged into a single, final result.



The MapReduce algorithm in an example of a word count task. The goal is to compute the number of occurrences of words in a subset of the data, the data being distributed on several clusters. MapReduce is not affected by the fact if the data being processed cannot be stored on a single node, and it leverages several nodes for the computation itself. Picture from Big Data & Hadoop: MapReduce Framework | EduPristine (6/2/2019) (https://www.edupristine.com/blog/hadoop-mapreduce-framework)

This is an example of an operation performed with the MapReduce algorithm. It allows to handle large amounts of data stored in a distributed fashion, and to optimize the calculation using several nodes in parallel.



Typical Components of a Hadoop distribution. Those components offer interfaces with data streams, SQL-like queries, Scala, Java, R, or Python Programs, ...

Hadoop clusters can be hosted in the cloud, or on-premise. Hadoop distributions comes with a set of tools in addition to HDFS and MapReduce, that have been added over time since its creation in 2004. Those additional components allow to query the data in a user-friendly way (SQL-like) with Impala, Hive, Sqoop. Some components handle data streams (Kafka and Flume), and the most important component nowadays is Spark, that offers huge performance improvements over Hadoop for reasons mentioned previously, and comes with nice Application Programming Interfaces (API) that allows to program on a Hadoop cluster in Scala, Java, R, or Python.

Further Reading

1. *TutorialsPoint, Hadoop Tutorial: https://www.tutorialspoint.com/hadoop/index.htm*
2. *Jeff Desjardins, 2019, What Happens in an Internet Minute in 2019?: https://www.visualcapitalist.com/what-happens-in-an-internet-minute-in-2019/*
3. *DataBricks, Getting Started with Spark:* https://databricks.com/spark/getting-started-with-apache-spark
4. *Stanford HCI Group, A very brief introduction to MapReduce:* https://hci.stanford.edu/courses/cs448g/a2/files/map_reduce_tutorial.pdf

Week 8B

Final Projects Presentation

The final projects are a great opportunity for you to use several skills learned in this course, and apply them on a topic that you like.

Select a **topic of interest** and find a relevant dataset that you would like to explore. Alternatively, you can bring a **non-confidential dataset from your company** or your own.

The outcome of your presentation can either be a set of data-driven business recommendations, a data visualization with insights, or a proposed machine learning application supported by a feasibility analysis.

- Make sure that there are specific questions that you would like to have answered.
- Download the dataset and use at least one of the tools presented in this course (Google Analytics, SQL, Zoho Analytics, Python, or Optimizely).
- Perform Exploratory Data Analysis (EDA) of your data using descriptive statistics. Build confidence intervals for your results, whenever it is possible.
- Possibly apply correlation and/or regression analysis to find patterns in the dataset, or train and evaluate a machine learning model if this is the topic of your presentation.
- Visualize your results using appropriate technology tools in a way that makes them easy to present and to understand.

Prepare a presentation of your analysis, tools, methodology, and findings. Anticipate questions that your classmates or instructor might have.

Your presentation should include most of the following components:

- Rationale for your dataset selection, the particular questions you were seeking to answer, and whether you were successful in answering them.
- Justification for your tools and methods choices.
- Your overall findings, including their level of statistical significance, and their potential real world significance. The numerical results and effective visuals that back up your findings and make your presentation more engaging for the audience, and the data preprocessing pipeline description.
- Any difficulties you have encountered as well as recommendations for additional analysis that you would conduct given some extra time.

You may use the following datasets for your presentation:

- A non-confidential dataset of your company or your own
- A dataset on a topic you would like to investigate. If you have identified the question you want to address but not the relevant dataset, you can search on the following links:
    - Google Datasets Search (https://toolbox.google.com/datasetsearch)
    - Kaggle Datasets (https://www.kaggle.com/datasets)
- One of the following datasets:
    - The Mall Customer Segmentation Data (https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python)
    - The Black Friday Dataset https://www.kaggle.com/mehdidag/black-friday
    - Board Game Dataset (https://www.kaggle.com/gabrio/board-games-dataset#database.sqlite)
    - IBM HR Analytics Employee Attrition & Performance (https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset)

Remember that you should limit yourself to a maximum of ten slides, and rehearse as much as possible in order to fluidify the speech and notice any timing or narrative issues.

🎥 It is a good idea to record yourself and visualize the video. This is the best way to iterate on and improve your presentation skills!

Exercises solutions

Week 1 A/B

1. Data landscape of an online store

Available data and fields needed to monitor the business (in parenthesis):

- Web navigation logs (connections locations, pages visited, time spent on pages, ...)
- User info (username, location, profile creation date, age, gender)
- Products info (prices, categories, ...)
- Orders (date, customer, product, location, ...)
- Possibly external data provided by third parties (e.g. if sign up via LinkedIn: social network graphs and relations, job title, ...)
- ...

2. Descriptive, predictive, and prescriptive analytics

Descriptive Analytics:

- Are the hoodies sales increasing?
- What is the revenue per region?
- How has the conversion rate (# pages visits / # purchases) evolved over the last 6 months?

Predictive Analytics:

- How many users will click on a given ad next month ?
- Where will the fire brigade be needed the most next August
- What is the likely of a student to complete the class

Prescriptive Analytics:

- Should we display ad A or ad B on the home page?
- How much raw material should we order next month to meet our needs as precisely as possible?
- Where should I open my next shop?

3. SMART goals

1.
    a. Can check all the boxes, but the time limit is a bit far, and one would have to assess the feasibility (Attainable)
    b. Not Specific ("churn" is vague and there is no target number) nor Time-bound
    c. Not Measurable (need to be more precise than "productivity") nor Time-bound
2. "We need to increase by 5% the number of non-returned hoodies sales in the U.S. within 3 months." (5 % is probably more Attainable, non-returned hoodies sales may be more Relevant, and we have added a deadline).

4. Structured Pyramid Analysis Plan for the Google merchandise store example

This is exercise is very opened and most answers are good, just make sure your original goal is SMART, and that you identify independent variables correctly from the specific questions.

First Google Analytics investigation

Note that those results change over time, here are quoted computed of the last 30 days of May 2019.

1.
    - Age: the 25-34 range converts more: 0.05% Ecommerce Conversion Rate (*Audience → Demographics → Age)*
    - Gender: Female (0.07% vs 0.03%) (*Audience → Demographics → Gender)*
    - Location: Canada and U.S. (0.21 %) (*Audience → Geo → Location)*
2. Referrals do not convert much (*Audience → All Traffic → Channels)*. Can be better referenced on Baidu (*Audience → Campaigns → Cost Analysis*). No revenue with paid keywords (*Audience → Campaigns → Paid Keywords*), has to improve Google Ads bids.
3. Many possibilities here: explore the Behavior flow (*Behavior → Behavior Flow)* and spot the high drop-off rate transitions.

Week 2A

1. The social network database MCQ

1. Just check the list provided in this document.
2. c. and e. (but a. may be semi-structured or structured)
3. b. and d.
4. Any relational database for structured data, a document-oriented database like MongoDB for a., flat files for b., and a graph database like Neo4j for d.

2. Course database design

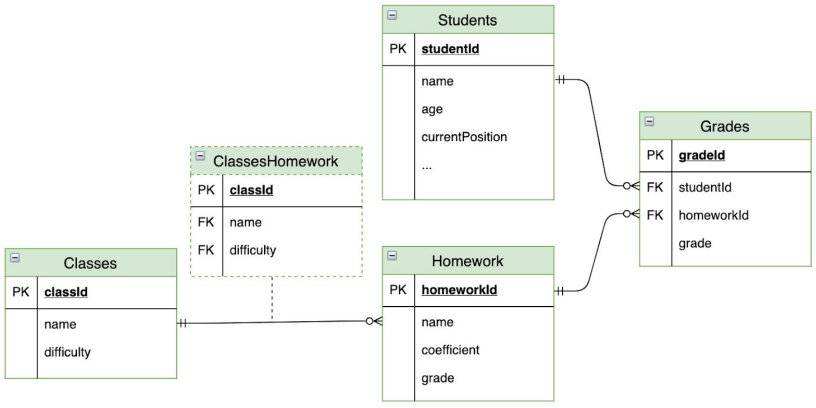
Example of data about students:

- Name,
- age,
- background,
- current position (if applicable),
- homework grades,
- ...

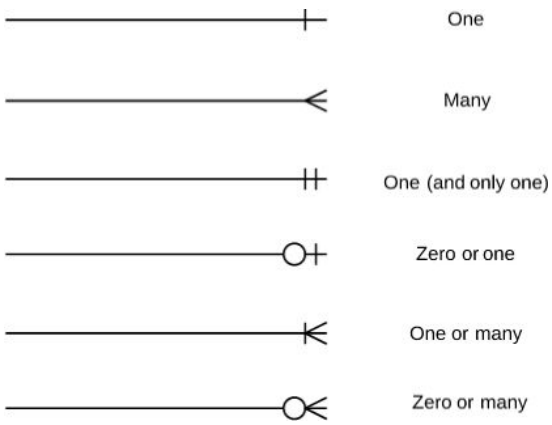
Content-related data:

- Name of the classes,
- Homework coefficients,
- Version of the class note,
- ...

The figure belows shows on possible modelling



One possible course database model. Each box represents a table, with the name at the top and fields below. PK means Primary Key and FK the acronym of Foreign Key (i.e. joins keys).



The ER diagram symbols meaning.

Week 2B

1. Select and table creation

1.
   a.
   ```
   SELECT * FROM Orders;
   ```
   b.
   ```
   SELECT ProductName, Unit, Price
   FROM Products;
   ```
   c.
   ```
   SELECT ProductName, Unit AS Dimension, Price AS Cost
   FROM Products;
   ```
2.
   a.
   ```
   CREATE TABLE students (
       id INT PRIMARY KEY,
       name VARCHAR(30),
       age INT
   );
   ```
   b.
   ```
   SELECT * FROM students;
   ```

2. Select with filters

1.
```sql
SELECT *
FROM Customers
WHERE Country = 'Sweden';
```

2.
```sql
SELECT CustomerName, City
FROM Customers
WHERE Country = 'Sweden';
```

3.
```sql
SELECT CustomerName, City
FROM Customers
WHERE Country = 'Sweden'
AND CustomerName LIKE 'B%';
```

3. Sorting in SQL

1.
```sql
SELECT *
FROM Employees
ORDER BY BirthDate DESC, LastName ASC;
```

2.
```sql
SELECT *
FROM Orders
ORDER BY OrderDate DESC
LIMIT 5;
```

3.
```sql
SELECT *
FROM Products
WHERE SupplierID = 8
ORDER BY Price ASC;
```

Week 3A

1. HackerRank challenges

All solutions are for the MySQL interpreter.

1. Japanese cities names

```
SELECT NAME
FROM CITY
WHERE COUNTRYCODE = 'JPN';
```

2. Name of employees

```
SELECT name
FROM Employee
ORDER BY name ASC;
```

3. More than 75 marks

```
SELECT Name
FROM STUDENTS
WHERE Marks > 75
ORDER BY REVERSE(SUBSTRING(REVERSE(Name), 1, 3)) ASC, ID ASC;
```

4. Type of Triangle

```
SELECT
    CASE
        WHEN (A + B <= C) OR (A + C <= B) OR (B + C <= A) THEN 'Not A
Triangle'
        WHEN A = B AND B = C THEN 'Equilateral'
        WHEN A = B OR B = C OR A = C THEN 'Isosceles'
        ELSE 'Scalene'
    END
FROM TRIANGLES;
```

5. Aggregations

```
SELECT COUNT(*)
```

```
FROM CITY
WHERE POPULATION > 100000;
```

6. The blunder

```
SELECT CEILING(AVG(Salary)-AVG(REPLACE(Salary,'0','')))
FROM EMPLOYEES;
```

7. African cities

```
SELECT CITY.NAME
FROM CITY
INNER JOIN COUNTRY
ON CITY.COUNTRYCODE = COUNTRY.CODE
WHERE COUNTRY.CONTINENT = 'Africa';
```

8. Average population

```
SELECT
    COUNTRY.Continent,
    FLOOR(AVG(CITY.Population))
FROM CITY
INNER JOIN COUNTRY
ON CITY.CountryCode = COUNTRY.Code
GROUP BY COUNTRY.Continent;
```

9. Customers in each Country

```
SELECT
Country
COUNT(CustomerID),
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

10. Placements

```
SELECT
Students.Name
FROM Students
JOIN Friends ON Students.ID=Friends.ID
```

```
JOIN Packages AS P1 ON Students.ID=P1.ID
JOIN Packages AS P2 ON Friends.Friend_ID=P2.ID
WHERE P2.Salary > P1.Salary
ORDER BY P2.Salary;
```

Week 4A

Solutions are all given in the "Introduction to Python" notebook.

Week 4B

Solutions are all given in the "Data science with Python" notebook, and the Pandas exercises solutions are contained [in the corresponding repository](https://github.com/guipsamora/pandas_exercises). (https://github.com/guipsamora/pandas_exercises)

Week 5A

1. Data Types

1. Quantitative Discrete
2. Quantitative Continuous (Even though we could say that you won't have a salary that has more that two decimals)
3. Qualitative Nominal
4. Qualitative Nominal
5. Qualitative Nominal
6. Qualitative Ordinal

2. Sampling Types

1. Observational & Cross Sectional because we don't provide any information or "treatments" before the survey and we don't run the experiment over time. The sampling method is Cluster Sampling because we picked randomly 10 cities, and we will study all the car buyers in those ten cities.

2. Experimental & Longitudinal because customers will test Grammarly and then we will see if they are better at writing in English or not & the experiment will be run for 6 months. The sampling method is Stratified sampling because we picked a sample that follows the same proportions than the population. The groups (countries) in the sample are mutually exclusive (assuming nobody has a double nationality).

3. Observational & Cross Sectional because we don't provide any treatments after the email is received and we don't run this experiment over time. Here we have a convenience sampling because of the lack of time.

Solutions are all given in the "Descriptive Statistics" notebook.

Week 5B

1. Confidence intervals for means

```
import numpy as np
from seaborn import load_dataset

tips = load_dataset("tips")

# Averages computation
waitresses_avg = tips[tips["sex"] == "Female"]["tip"].mean()
waiters_avg = tips[tips["sex"] == "Male"]["tip"].mean()

print("Average tip")
print(f"Waitresses: ${waitresses_avg:.2f}")
print(f"Waiters: ${waiters_avg:.2f}")

# Confidence intervals computation
waitresses_std = tips[tips["sex"] == "Female"]["tip"].std()
n_waitresses = tips[tips["sex"] == "Female"]["tip"].count()

waiters_std = tips[tips["sex"] == "Male"]["tip"].std()
n_waiters = tips[tips["sex"] == "Female"]["tip"].count()

confidence_waitresses = 1.96 * waitresses_std / np.sqrt(n_waitresses)  #
1.96 corresponds to the 95 % confidence interval
confidence_waiters = 1.96 * waiters_std / np.sqrt(n_waiters)

print("Confidence intervals")
print(f"Waitresses: {waitresses_avg:.2f}±{confidence_waitresses:.2f}")
print(f"Waiters: {waiters_avg:.2f}±{confidence_waiters:.2f}")

# The confidence intervals overlap: the tips difference is not
statistically significant
```

Week 6A

Solutions are all given in the "Introduction to Machine Learning" notebook.

Week 6B

1. The False Positive / False Negative dilemma

1. *Tumor malignancy prediction:* Avoid False Negatives (missed malignant tumors)
2. *A spam detection algorithm:* Avoid False Positives (valid emails mistakenly considered as spams)
3. *A pregnancy test*: avoid False Negatives (False Positives are pondered by performing an additional test, which potential mothers usually do)
4. *A court judgment*: it depends on your legal system! Avoiding False Negatives would mean that you are more likely to send an innocent to prison, and avoiding False Positives means that you may wrongfully release criminals.

2. Count your model mistakes

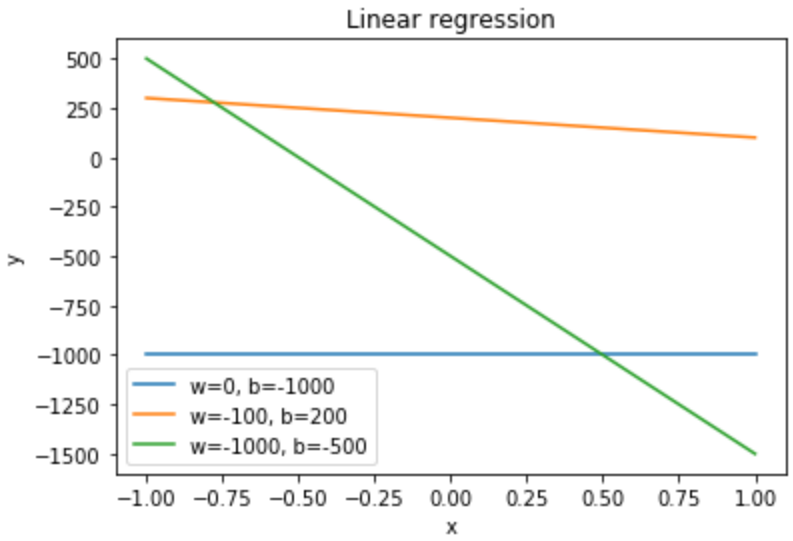
Boundary color	True Positives	False Positives	False Negatives	True Negatives
Blue	5	5	0	1
Yellow	4	2	1	4
Green	3	0	2	6

Week 7A/B

1. Linear Regression plots and intuition

```
import numpy as np
import matplotlib.pyplot as plt

x_fit = np.linspace(-1, 1, 100)

parameters = [(0, -1000), (-100, 200), (-1000, -500)]

for w, b in parameters:
    y_fit = w*x_fit + b
    plt.plot(x_fit, y_fit, label=f"w={w}, b={b}")

plt.legend()
plt.title("Linear regression")
plt.xlabel("x")
plt.ylabel("y");
```

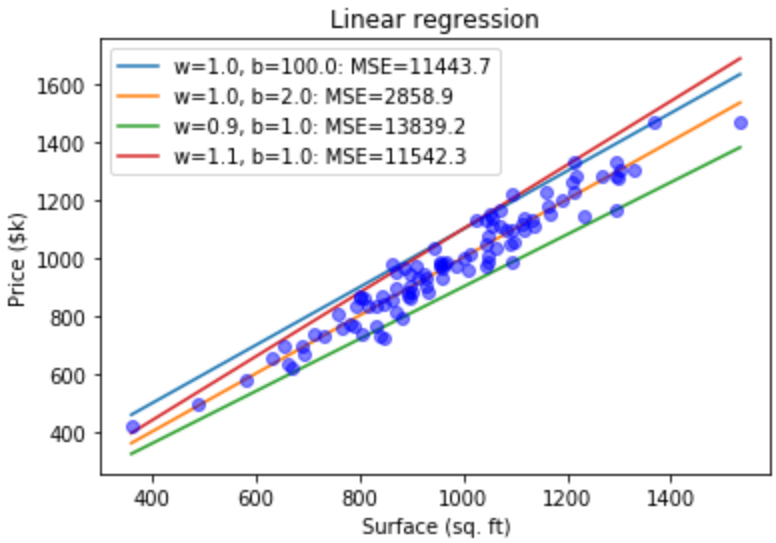
Output:



2. Mean Squared Error computation

```
import numpy as np
np.random.seed(2019)

surface = 1000 + (200*np.random.randn(100)).astype(int)
price = surface + 2 + (50*np.random.randn(100)).astype(int)

def compute_mse(true_values, predictions):
    n = len(true_values)
    return np.sum((true_values-predictions)**2)/n

x_fit = np.linspace(min(surface), max(surface), 100)

parameters = [(1., 100.), (1., 2.), (0.9, 1.), (1.1, 1.)]

for w, b in parameters:
    predicted_price = w*surface + b
    mse = compute_mse(price, predicted_price)
    y_fit = w*x_fit + b
    plt.plot(x_fit, y_fit, label=f"w={w}, b={b}: MSE={mse:.1f}")

# Plot the data points
plt.plot(surface, price, "bo", markersize=6, alpha=0.5)

plt.legend()
plt.title("Linear regression")
plt.xlabel("Surface (sq. ft)")
plt.ylabel("Price ($k)");
```

Output:



So the second set of parameters is the best one: it has the lowest Mean Squared Error.