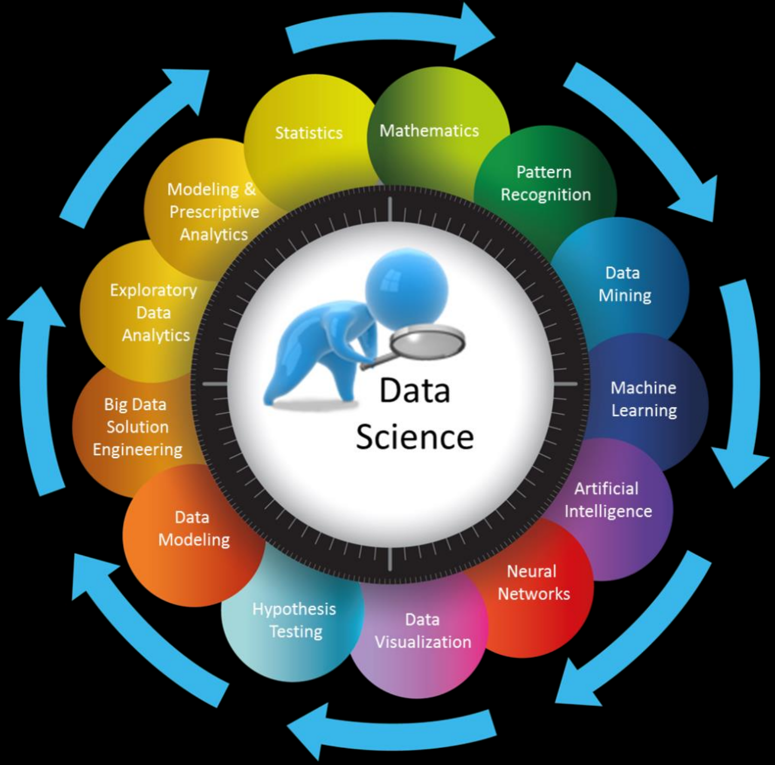


Introduction to Data Science

Unit 1

Course Contents

Introduction to Data Science [10 Hrs.]

Introduction to data science, Applications of data science; Limitations of data science Commonly used tools in data science, their strengths and common use-cases: R/RStudio, Python/Pandas/Jupyter Notebooks, Excel/Tableau/PowerBI;

Data Science life-cycle/Common methodologies for data science: CRISP-DM, OSEMN Framework, TDSP lifecycle;

Review of statistics and probability: Probability distributions, compound events and independence. Statistics: Centrality measures, variability measures, interpreting variance. Correlation analysis: Correlation coefficients, autocorrelation

Before we begin…. Let’s understand the data first

- Data is generated in various sources

Mobile Apps

Computer Applications

Point of Sale

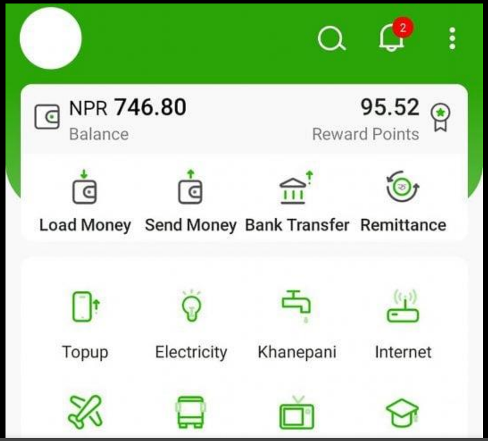
Stock Market

Weather

Security Surveillance

Before we begin…. Let’s understand the data first (contd.)

- Also,



NPR 746.80
Balance

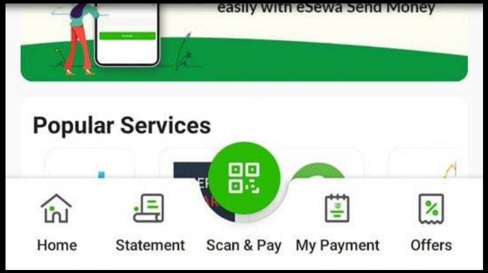
95.52
Reward Points

& many more

Astronomy

Medical Diagnosis

Scientific Study

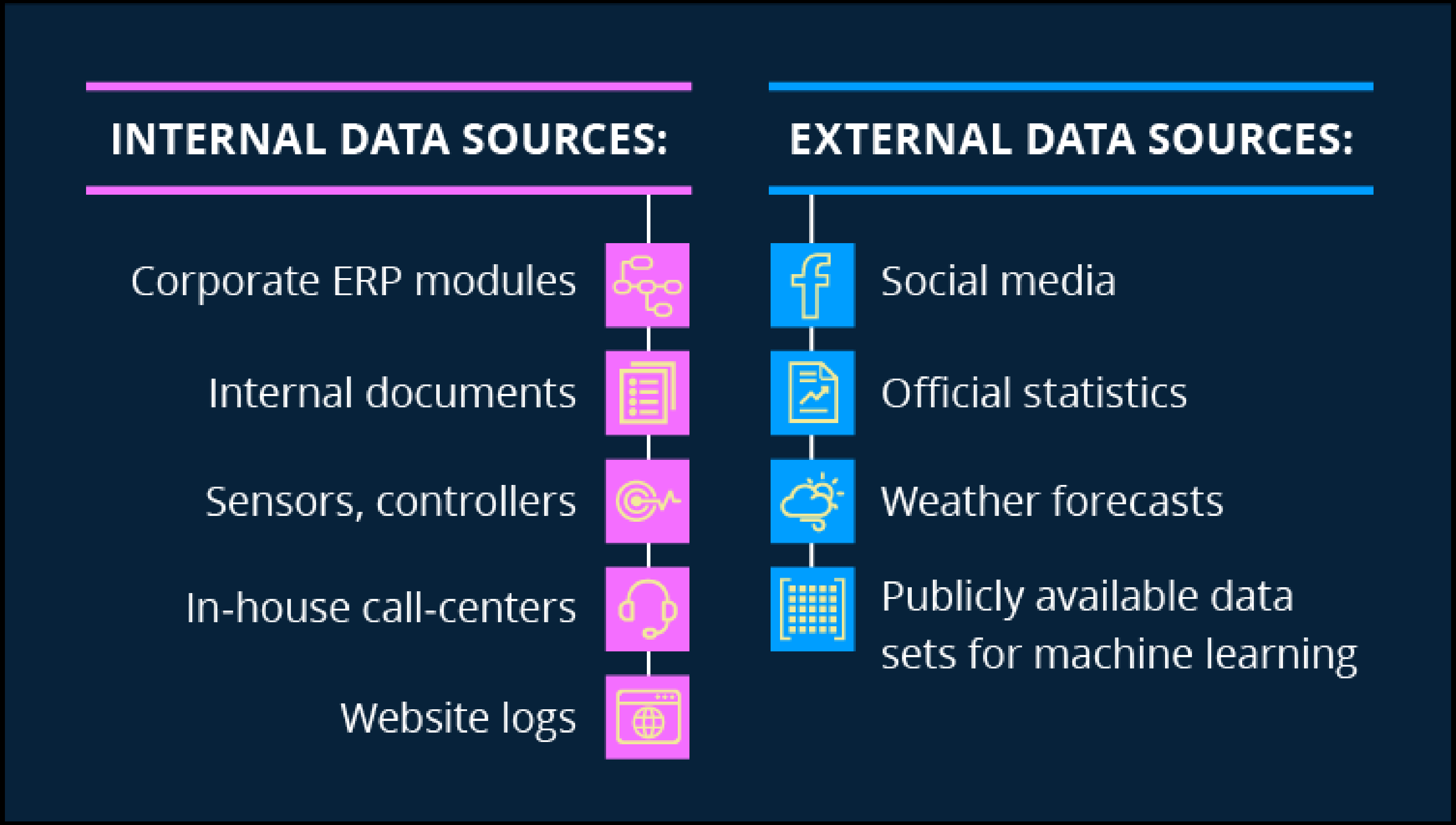


easily with eSewa Send Money

Popular Services

Financial Transactions

And, we categorize all these data sources into:



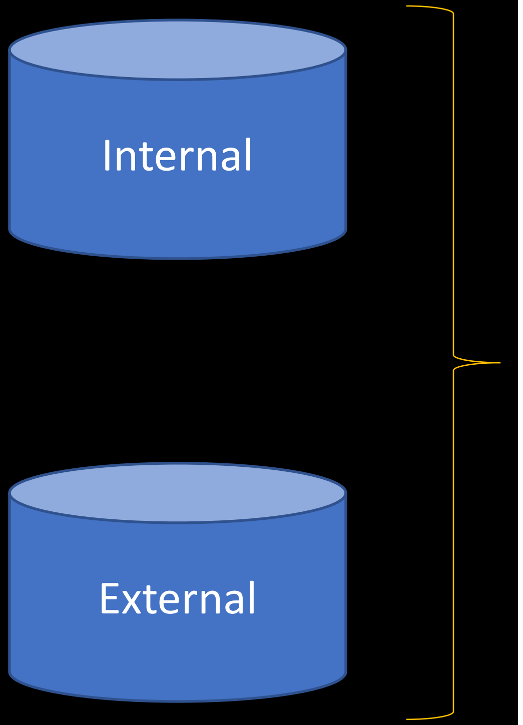
INTERNAL DATA SOURCES:

- Corporate ERP modules
- Internal documents
- Sensors, controllers
- In-house call-centers
- Website logs

EXTERNAL DATA SOURCES:

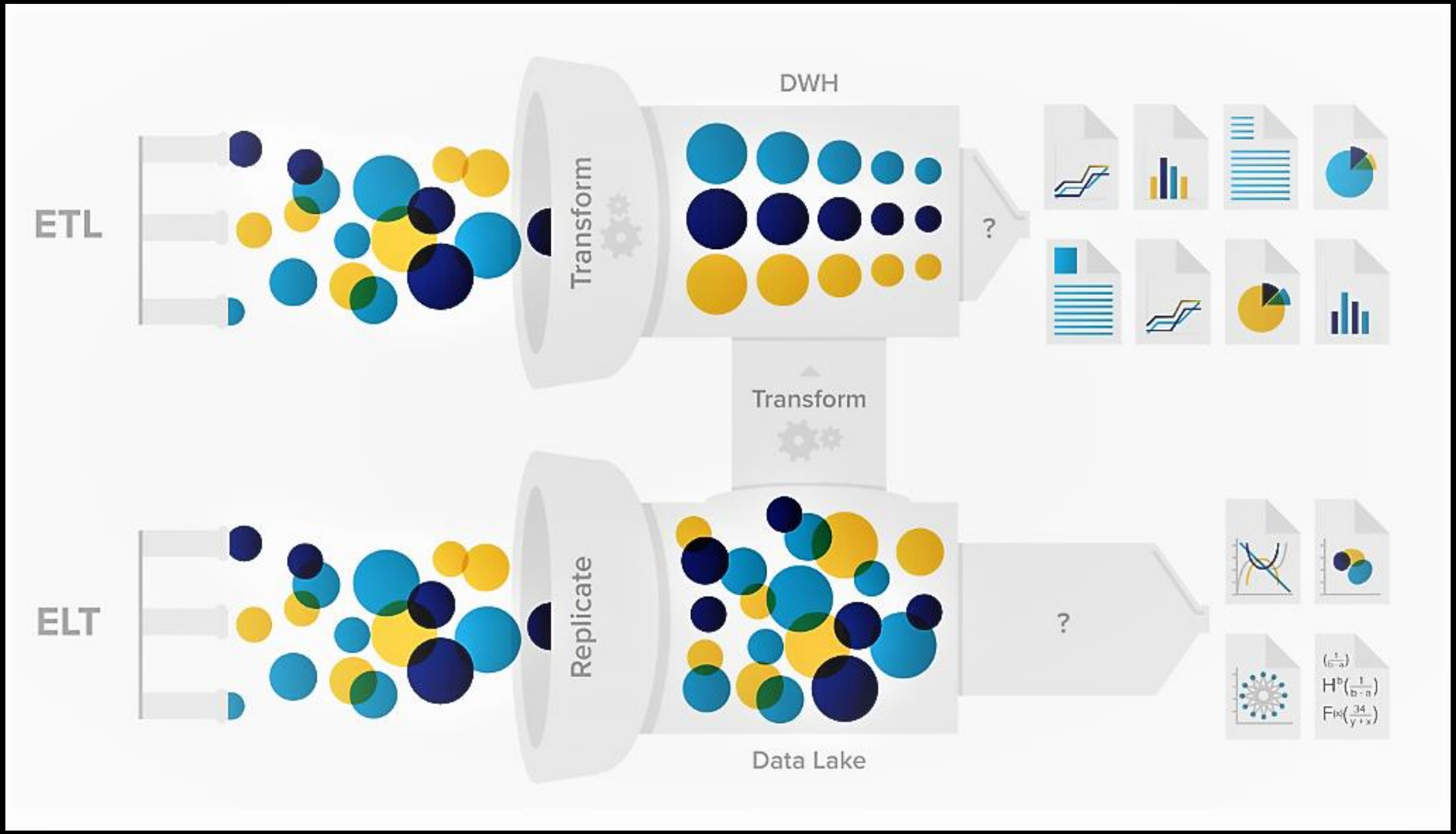
- Social media
- Official statistics
- Weather forecasts
- Publicly available data sets for machine learning

Data migrates from various sources to Data Warehouse (Data Lake)

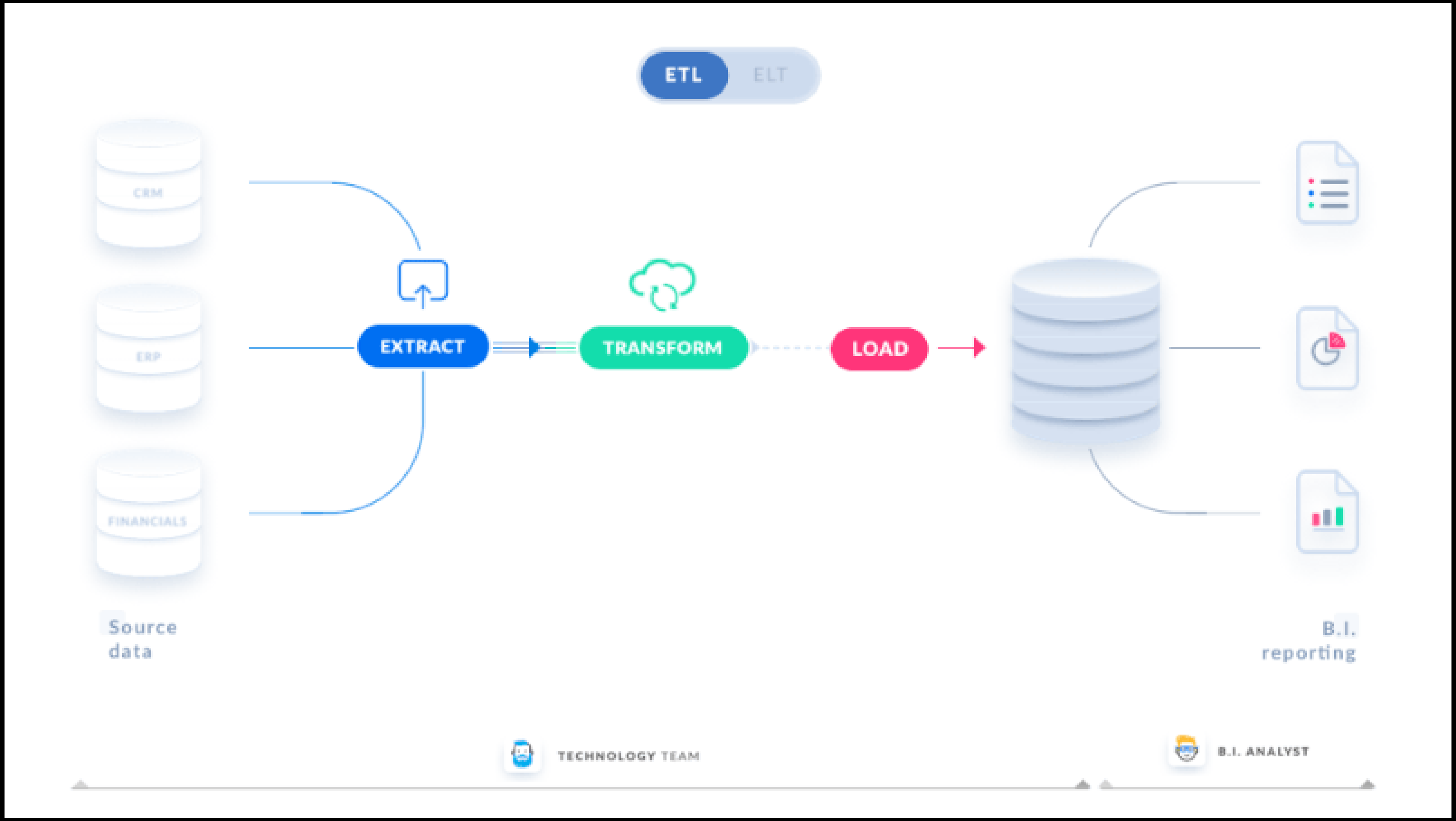


Internal

External



ETL vs ELT tootls



Data Warehouse

- A data warehouse is a huge collection of business data used to help an organization make decisions.
- The large amount of data in data warehouses comes from different sources such as internal applications such as marketing, sales, and finance; customer-facing apps; and external partner systems, among others.
- On a technical level, a data warehouse periodically pulls data from those apps and systems; then, the data goes through formatting and import processes to match the data already in the warehouse.

Data Warehouse (contd.)

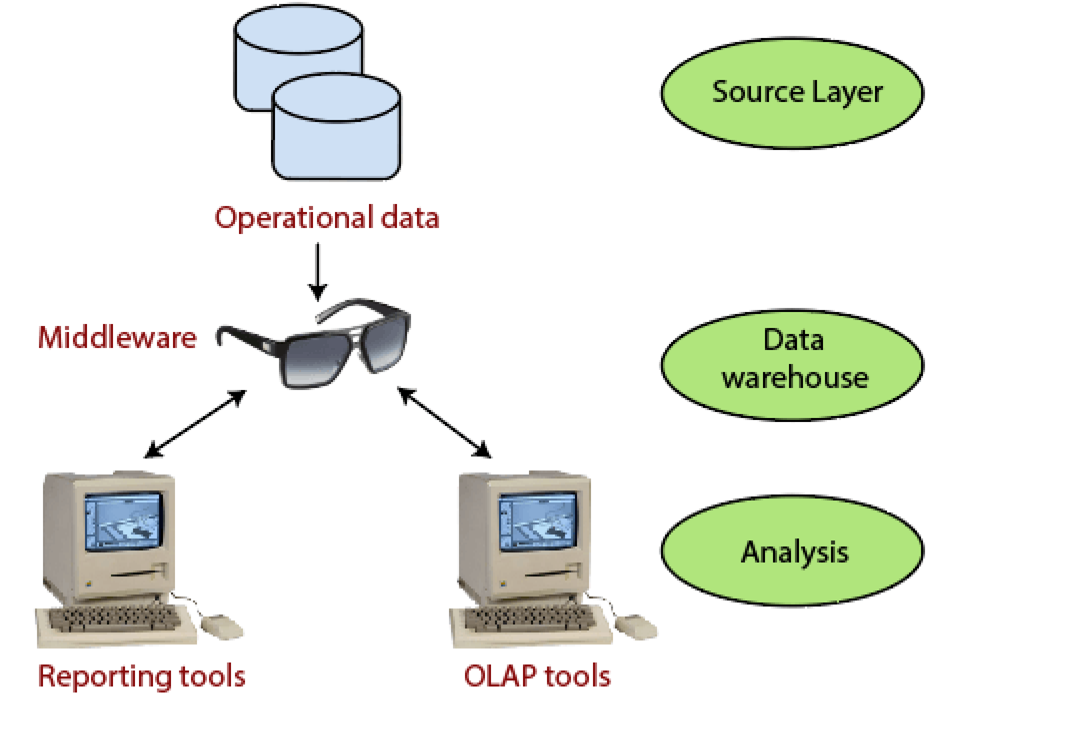
- The data warehouse stores this processed data so it’s ready for decision makers to access.
- How frequently data pulls occur, or how data is formatted, etc., will vary depending on the needs of the organization.

Characteristics of Datawarehouse

1. **Subject Oriented**: focused on specific subject area
2. **Integrated**: Integrates data from multiple sources
3. **Time Variant**: Stores historical data
4. Non Volatile: Permanent Storage

Data Warehouse Architecture

Single-Tier architecture is not periodically used in practice. Its purpose is to minimize the amount of data stored to reach this goal; it removes data redundancies.



Single-Tier Data Warehouse Architecture

Data Warehouse Architecture (contd.)

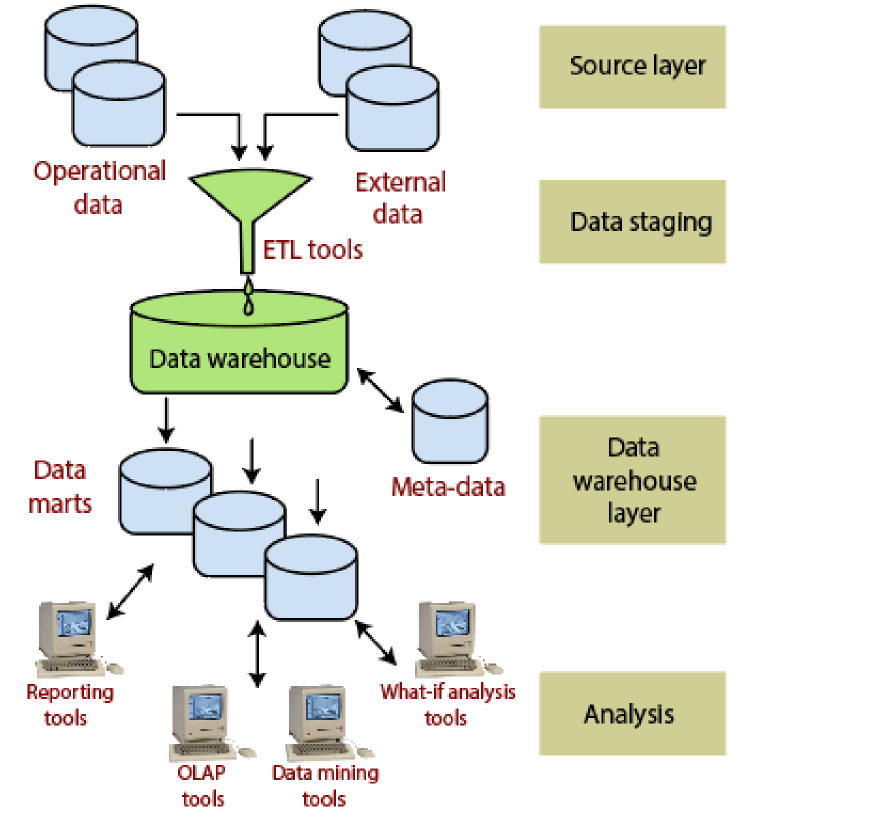
Although it is typically called two-layer architecture to highlight a separation between physically available sources and data warehouses, in fact, consists of four subsequent data flow stages:

a) Source layer

b) Data Staging

c) Data Warehouse layer

d) Analysis

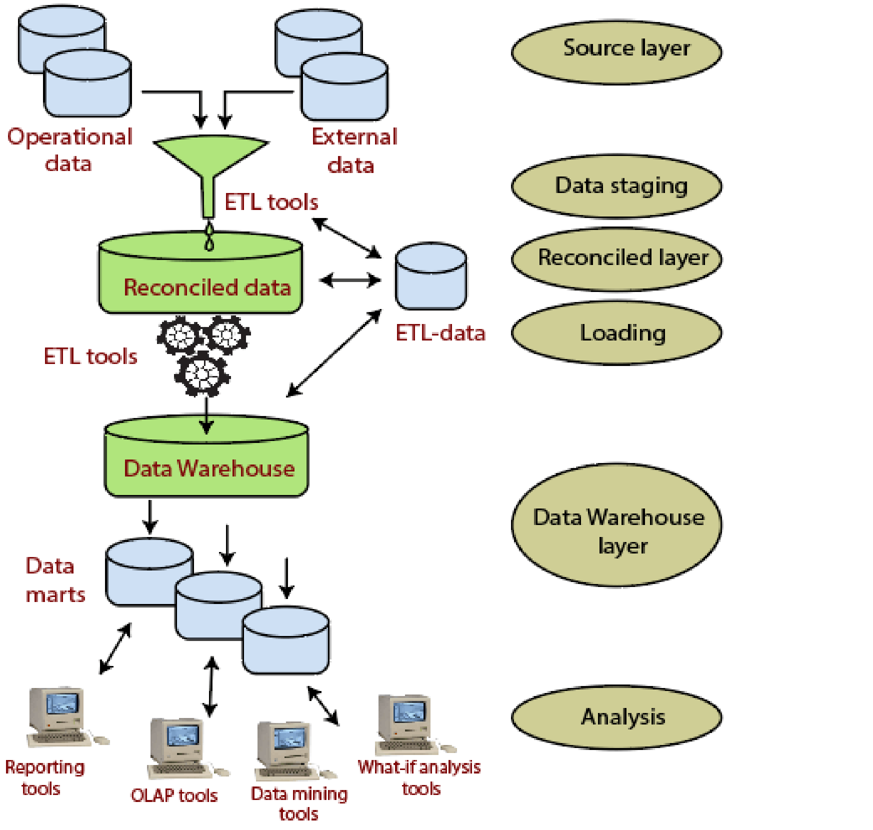


Two-Tier Data Warehouse Architecture

Data Warehouse Architecture (contd.)

The three-tier architecture consists of the source layer (containing multiple source system), the reconciled layer and the data warehouse layer (containing both data warehouses and data marts). The reconciled layer sits between the source data and data warehouse.

The main advantage of the reconciled layer is that it creates a standard reference data model for a whole enterprise. At the same time, it separates the problems of source data extraction and integration from those of data warehouse population. In some cases, the reconciled layer is also directly used to accomplish better some operational tasks, such as producing daily reports that cannot be satisfactorily prepared using the corporate applications or generating data flows to feed external processes periodically to benefit from cleaning and integration.



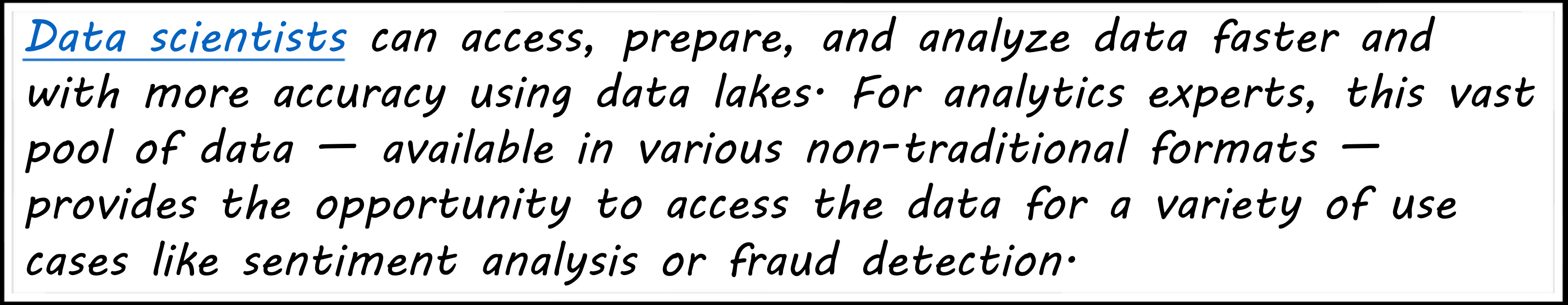
Three-Tier Architecture for a data warehouse system

Data Lake

- A data lake is a central storage repository that holds big data from many resources in a raw, granular format.
- It can store structured, semi-structured or unstructured data, which means data can be kept in a more flexible format for future use.
- When storing data, a data lake associates it with identifiers and metadata tags for faster retrieval.
- The term “data lake” refers to the ad hoc nature of data in a data lake, as opposed to the clean and processed data stored in traditional data warehouse system.

Data Lake (contd.)

- A data lake works on a principle called schema-on-read.
- This means that there is no predefined schema into which data needs to be fitted before storage.
- Only when the data is read during processing is it parsed and adapted into a schema as needed.
- This feature saves a lot of time that's usually spent on defining a schema. This also enables data to be stored as is, in any format.



Data scientists can access, prepare, and analyze data faster and with more accuracy using data lakes. For analytics experts, this vast pool of data — available in various non-traditional formats — provides the opportunity to access the data for a variety of use cases like sentiment analysis or fraud detection.

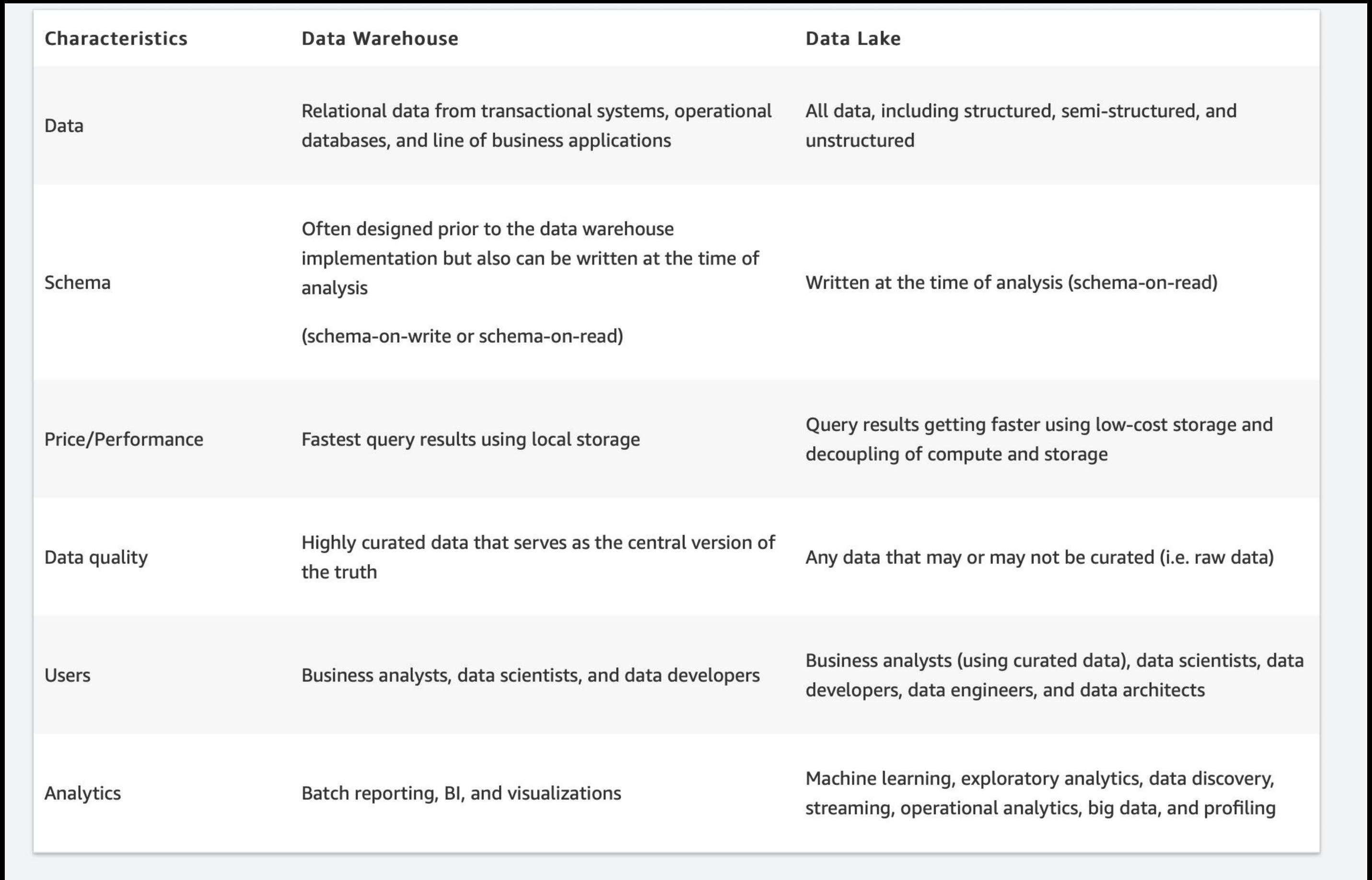
Data lake vs data warehouse - Similarities

- A data lake and a data warehouse are similar in their basic purpose and objective, which make them easily confused:
  - Both are storage repositories that consolidate the various data stores in an organization.
  - The objective of both is to create a one-stop data store that will feed into various applications.

Data lake vs data warehouse - Differences

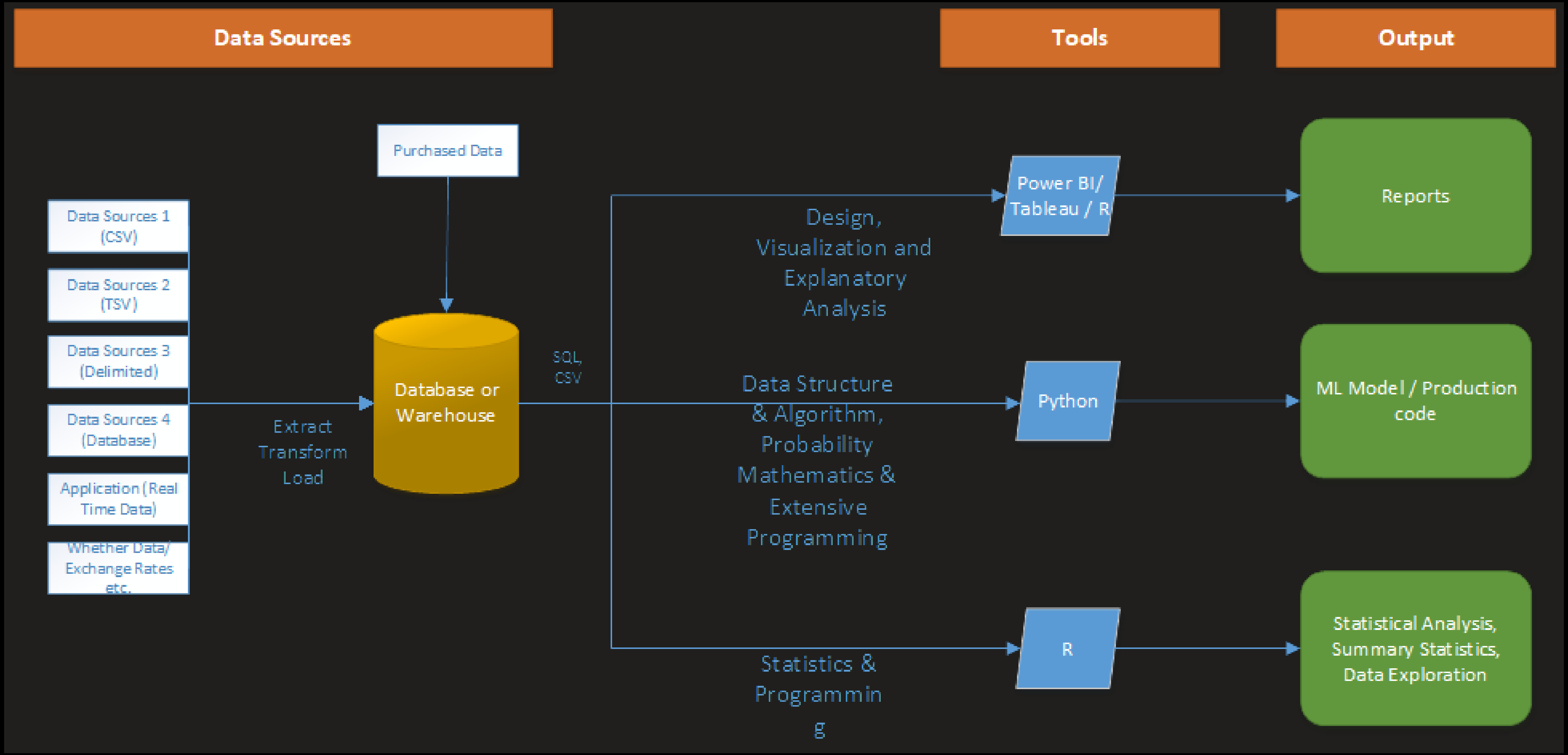
- **Schema-on-read vs schema-on-write**
  - The schema of a data warehouse is defined and structured before storage (schema is applied while writing data). A data lake, in contrast, has no predefined schema, which allows it to store data in its native format.
  - In a data warehouse most of the data preparation usually happens before processing. In a data lake, it happens later, when the data is actually being used.
- **Complex vs simple user accessibility**
  - As data is not organized in a simplified form before storage, a data lake often needs an expert with a thorough understanding of the various kinds of data and their relationships, to read through it. A data warehouse, in contrast, is easily accessible to both tech and non-tech users due its well-defined and documented schema. Even a new member on the team can begin to use a warehouse quickly.
- **Flexibility vs rigidity**
  - With a data warehouse, not only does it take time to define the schema at first, it also takes considerable resources to modify it when requirements change in the future. However, data lakes can adapt to changes easily. Also, as the need for storage capacity increases, it is easier to scale the servers on a data lake cluster.

Data lake vs data warehouse - Differences



Characteristics	Data Warehouse	Data Lake
Data	Relational data from transactional systems, operational databases, and line of business applications	All data, including structured, semi-structured, and unstructured
Schema	Often designed prior to the data warehouse implementation but also can be written at the time of analysis (schema-on-write or schema-on-read)	Written at the time of analysis (schema-on-read)
Price/Performance	Fastest query results using local storage	Query results getting faster using low-cost storage and decoupling of compute and storage
Data quality	Highly curated data that serves as the central version of the truth	Any data that may or may not be curated (i.e. raw data)
Users	Business analysts, data scientists, and data developers	Business analysts (using curated data), data scientists, data developers, data engineers, and data architects
Analytics	Batch reporting, BI, and visualizations	Machine learning, exploratory analytics, data discovery, streaming, operational analytics, big data, and profiling

Thus,


Data Sources
Tools
Output
Purchased Data
Data Sources 1 (CSV)
Data Sources 2 (TSV)
Data Sources 3 (Delimited)
Data Sources 4 (Database)
Application (Real Time Data)
Whether Data/ Exchange Rates etc.
Extract Transform Load
Database or Warehouse
SQL, CSV
Design, Visualization and Explanatory Analysis
Data Structure & Algorithm, Probability Mathematics & Extensive Programming
Statistics & Programmin g
Power BI/ Tableau / R
Python
R
Reports
ML Model / Production code
Statistical Analysis, Summary Statistics, Data Exploration

Introduction to Data Science

- Over the past few years, there’s been a lot of hype in the internet about “data science” and “Big Data.”
- But, what actually “Data Science” is? And what does “Big Data” means?
- How “Data Science” and “Big Data” are related?
- Is data science the science of Big Data?
- Is data science only the stuff going on in companies like Google and Facebook and tech companies?
- Why do many people refer to Big Data as crossing disciplines (astronomy, finance, tech, etc.) and to data science as only taking place in tech? Just how big is big? Or is it just a relative term?

Introduction to Data Science (contd.)

Is it new Age Thing?

- Statisticians already feel that they are studying and working on the "Science of Data." That's their bread and butter.
- Many of the algorithms (*e.g. linear regression, logistic regression, Bayesian Statistics, and even neural network*) we used today were discovered long back in the past.
- However, many of the methods and techniques we're using—and the challenges we're facing now—are part of the evolution of everything that's come before.

Introduction to Data Science (contd.)

Why now?

- We have massive amounts of data about many aspects of our lives, and, simultaneously, an abundance of inexpensive computing power.
- *Shopping, communicating, reading news, listening to music, searching for information, expressing our opinions*—all this is being tracked online, as most people know.
- What people might not know is that the **"datafication"** of our offline behavior has started as well, mirroring the online data collection revolution.
- Put the two together, and there's a lot to learn about our behavior and, by extension, who we are as a species.

Introduction to Data Science (contd.)

Why now?

- It’s not just Internet data, though—it’s finance, the medical industry, pharmaceuticals, bioinformatics, social welfare, government, education, retail, and the list goes on.
- But it’s not only the massiveness that makes all this new data interesting (or poses challenges).
- It’s that the data itself, often in real time, becomes the building blocks of data products.

Introduction to Data Science (contd.)

Why now?

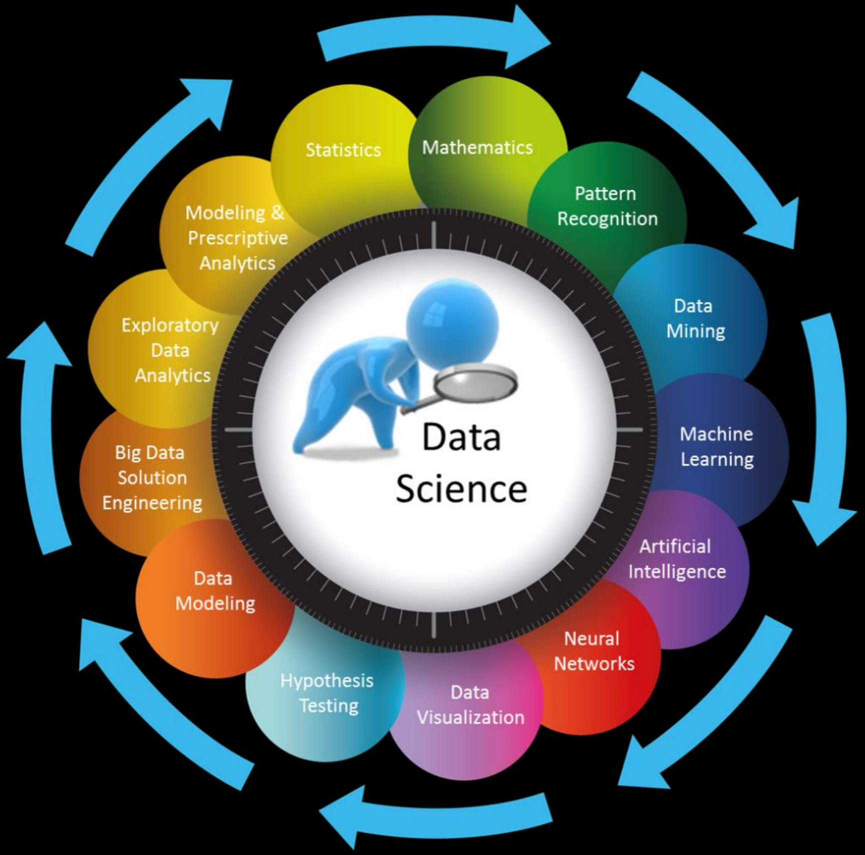
- We’re witnessing the beginning of a massive, culturally saturated feedback loop where our behavior changes the product and the product changes our behavior.
- Technology makes this possible: infrastructure for large-scale data processing, increased memory, and bandwidth, as well as a cultural acceptance of technology in the fabric of our lives.

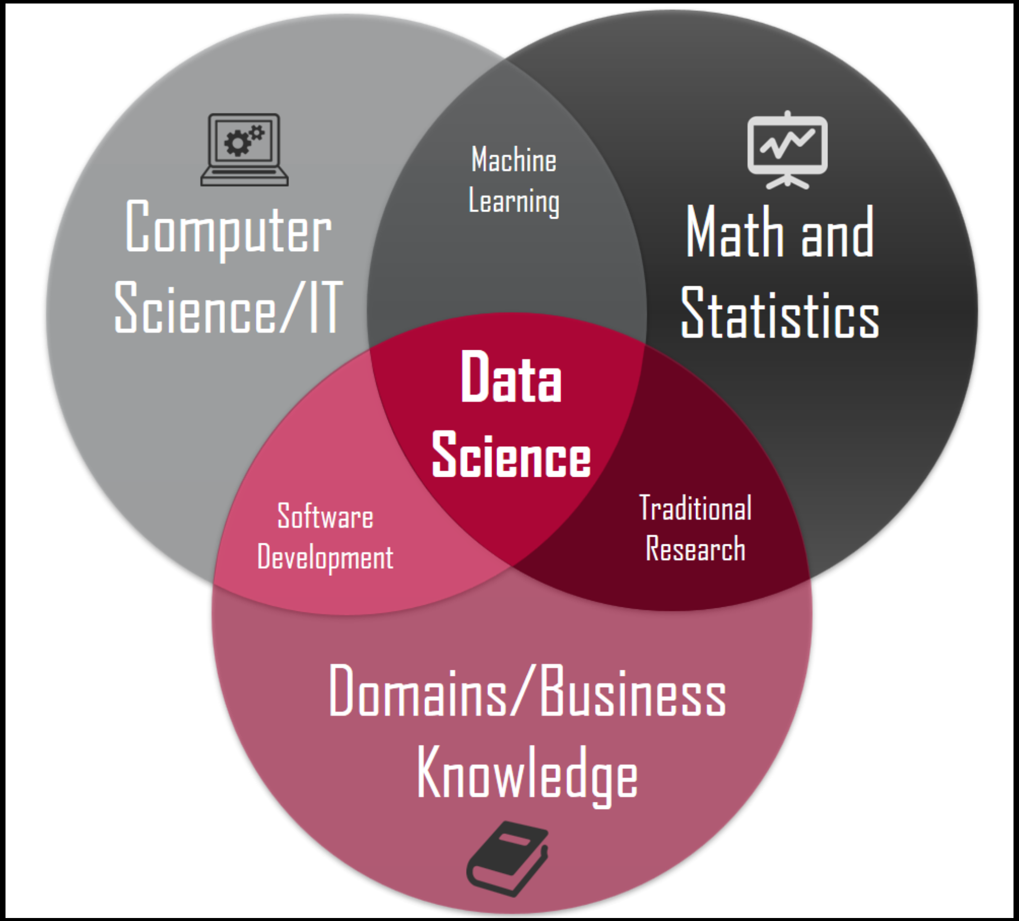
Introduction to Data Science (contd.)

So, what is data science?

- Is it new, or is it just statistics or analytics rebranded? Is it real, or is it pure hype? And if it’s new and if it’s real, what does that mean?
- This is an ongoing discussion, and many people have their own definition, their own explanations.

Introduction to Data Science (contd.)





Drew Conway's Venn diagram of data science

Introduction to Data Science (contd.)

- Metamarket CEO Mike Driscoll defines data science as:
  - Data science, as it’s practiced, is a blend of Red-Bull-fueled hacking and espresso-inspired statistics.
  - But data science is not merely hacking—because when hackers finish debugging their Bash one-liners and Pig scripts, few of them care about non-Euclidean distance metrics.
  - And data science is not merely statistics, because when statisticians finish theorizing the perfect model, few could read a tab-delimited file into R if their job depended on it.
  - Data science is the civil engineering of data. Its acolytes possess a practical knowledge of tools and materials, coupled with a theoretical understanding of what’s possible.

Introduction to Data Science (contd.)

- Who is data scientist?

A data scientist is someone who knows how to extract meaning from and interpret data, which requires both tools and methods from statistics and machine learning, as well as being human. She spends a lot of time in the process of collecting, cleaning, and munging data, because data is never clean. This process requires persistence, statistics, and software engineering skills—skills that are also necessary for understanding biases in the data, and for debugging logging output from code.

Introduction to Data Science (contd.)

- Who is data scientist?

A chief data scientist should be setting the data strategy of the company, which involves a variety of things: setting everything up from the engineering and infrastructure for collecting data and logging, to privacy concerns, to deciding what data will be user-facing, how data is going to be used to make decisions, and how it’s going to be built back into the product. She should manage a team of engineers, scientists, and analysts and should communicate with leadership across the company, including the CEO, CTO, and product leadership. She’ll also be concerned with patenting innovative solutions and set- ting research goals.

Introduction to Data Science (contd.)

- Who is data scientist?

Once she gets the data into shape, a crucial part is exploratory data analysis, which combines visualization and data sense. She'll find patterns, build models, and algorithms—some with the intention of understanding product usage and the overall health of the product, and others to serve as prototypes that ultimately get baked back into the product. She may design experiments, and she is a critical part of data driven decision making. She'll communicate with team members, engineers, and leadership in clear language and with data visualizations so that even if her colleagues are not immersed in the data themselves, they will understand the implications.

Applications of Data Science

1.Finance
- Risk Management
- Financial Analysis
- Stock price Prediction

2.Retailing
- Inventory Replenishment
- Customer Segmentation
- Promotion campaigning
- Sales Campaigning
- Demand and Supply projection

3.Banking
- Resource Utilization
- Fraud Detection
- Loan Management
- Predictive Analysis
- Customer Segmentation

4. Manufacturing
- Resource Utilization
- Monitoring of energy costs and optimize production hour.
- Identify potential problems through analysis of continuous stream of data

Applications of Data Science (contd.)

5. Health
    - Medical Image Analysis
    - Genetics and Genomics
    - Drug Discovery
    - Predictive Modeling for Diagnosis
    - Health bots or virtual assistants

- Other Applications
    - Fraud and Risk Detection
    - Personalized Advertising
    - Content Recommendation
    - Advanced Image Recognition
    - Airline Route Planning
    - Gaming etc.

Limitations of Data Science

- It's a Blurry Team
- Mastering Data Science is near to impossible.
- Large amount of Domain knowledge is required
- Arbitrary data may yield unexpected result.
- Data alone cannot guide decisions because they are a means of orienting you toward
- Difficulty in devising a conclusion if dataset is smaller.
- Garbage in Garbage Out

Commonly used tools

- Python
- R
- Tableau
- Power BI
- Excel / Google Sheets
- Jupyter
- Weka
- Apache Spark
- Apache Hadoop
- MATLAB
- Julia
- Scala
- Azure/Google Cloud/AWS
- MLOps
- AutoML

Commonly used library

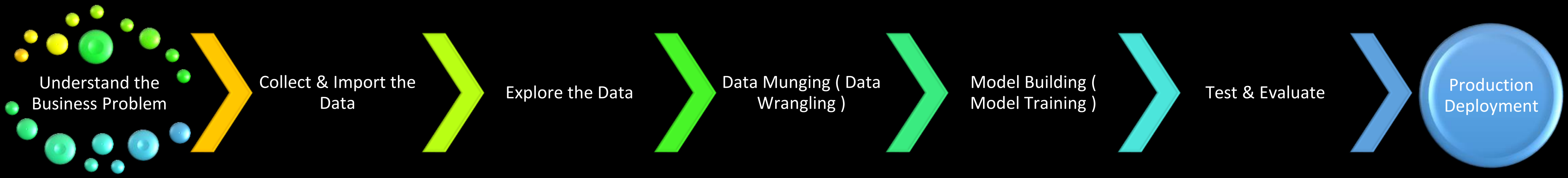
Python

- NumPy
- SciPy
- Pandas
- Matplotlib
- SciKit-Learn
- Tensorflow
- Pytorch
- BeautifulSoup
- NLTK
- OpenCV
- ScraPY
- XGBoost
- Seaborn

R

- Dplyr
- Tidyr
- Readr
- Stringr
- Ggplot2
- Lubridate
- Jsonlite
- BioConductor
- Shinny
- KnittR
- Caret
- Rmarkdown

Basic Steps for Data Science



Understand the Business Problem

Collect & Import the Data

Explore the Data

Data Munging (Data Wrangling)

Model Building (Model Training)

Test & Evaluate

Production Deployment

Let’s make hands dirty

Task 01 – Build Classification Model to classify Survival in Titanic

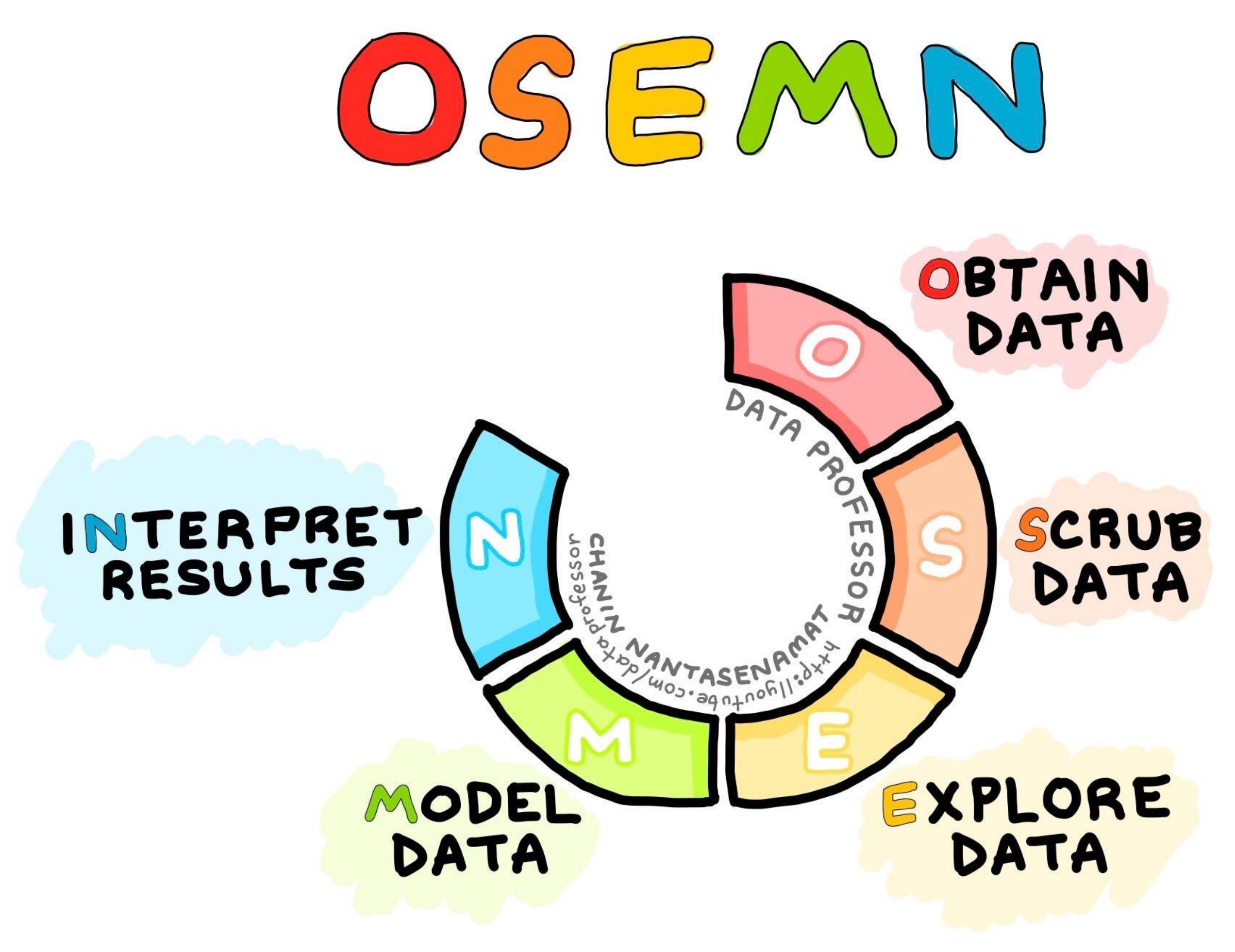
URL - shorturl.at/grxBJ

Data Science Life Cycle

OSEMN Framework

OSEMN Framework

- **O**btain
- **S**crub
- **E**xplore
- **M**odel
- i**N**terpret



OSEMN Framework (Contd.)

- Benefits:
  - Simple – It distills the complex process of a data science project into five clear steps. This is especially noteworthy given that this general process and the modern concept of data science were still new when Mason and Wiggins created OSEMN in 2010.
  - Catchy – OSEMN is Awesome!
  - Makes sense – The steps presented have a logical flow representative of the general data science life cycle.
  - Provides a shared understanding – OSEMN creates a taxonomy to help define how a data science project progresses.

Source: https://www.datascience-pm.com/osemn/

OSEMN Framework (Contd.)

- Shortcomings:
  - **Misses business understanding** – The framework starts with Obtain which ignores the key base questions that should come first, namely: “Should I invest time on this project?” and “What outcome am I trying to drive?”
  - **Doesn’t consider deployment** – OSEMN implicitly assumes that you are delivering a one-time output. In reality, you often need to deploy a model in a production system so that it continues to provide value over time.
  - **Ignores teamwork** – Data science is increasingly a team sport. Yet, OSEMN ignores the broader team aspect of modern projects.
  - **It’s linear** – OSEMN proceeds in a waterfall-like manner with each phase following the other. In reality, you often switch back and forth between phases as needed. Moreover, you will want frequent decision points where you re-assess and adjust your plan based on recent learnings.

Source: https://www.datascience-pm.com/osemn/

Team Data Science Process (TDSP)

Source: https://learn.microsoft.com/en-us/azure/architecture/data-science-process/overview

TDSP

- The Team Data Science Process (TDSP) is an agile, iterative data science methodology to deliver predictive analytics solutions and intelligent applications efficiently.
- TDSP helps improve team collaboration and learning by suggesting how team roles work best together.
- TDSP includes best practices and structures from Microsoft and other industry leaders to help toward successful implementation of data science initiatives.
- The goal is to help companies fully realize the benefits of their analytics program.

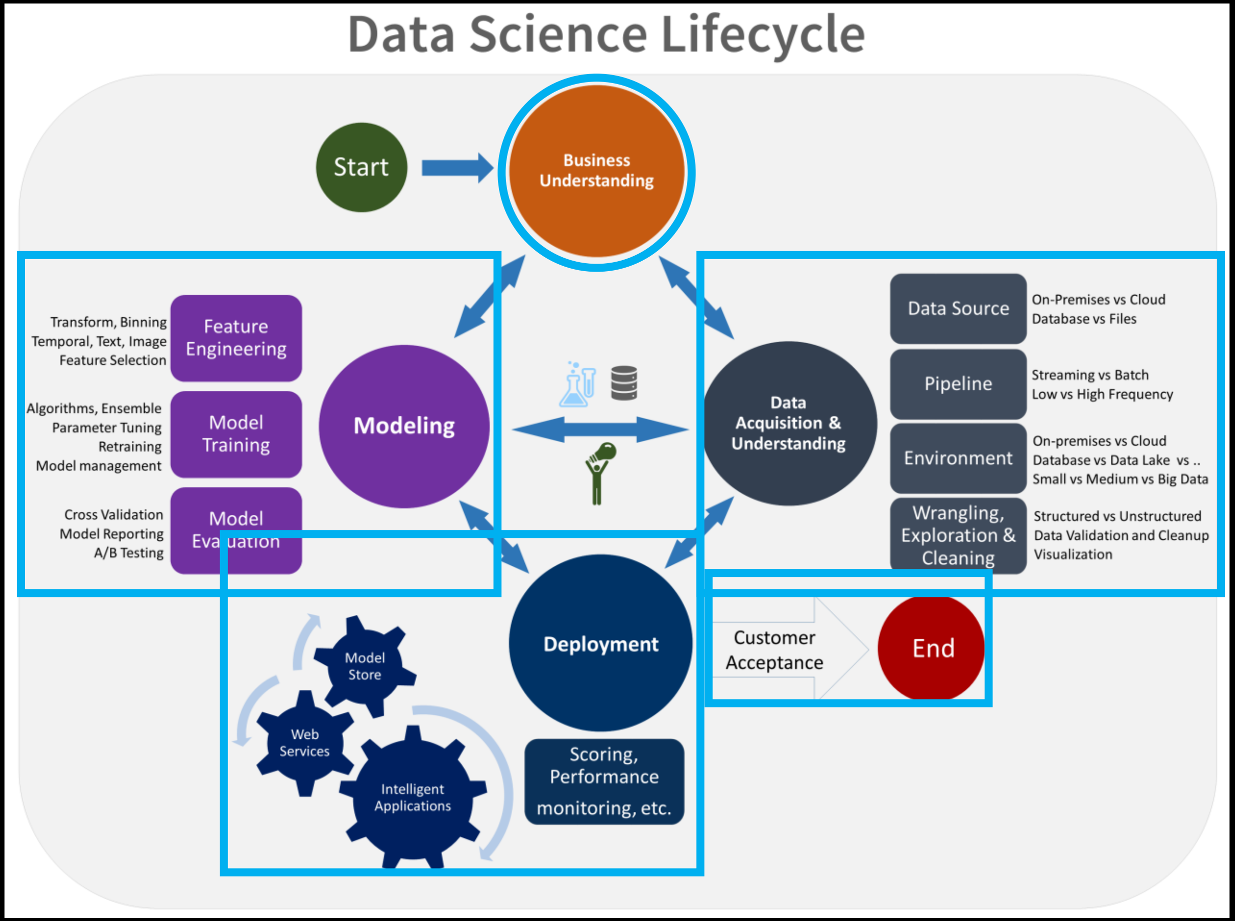
TDSP (contd.)

- Key Components of TDSP
  - A data science lifecycle
  - A standardized project structure
  - Infrastructure and resources
  - Tools and utilities

TDSP (contd.)

- Data Science Lifecycle
  - The Team Data Science Process (TDSP) provides a lifecycle to structure the development of your data science projects.
  - The lifecycle outlines the full steps that successful projects follow.
  - The lifecycle outlines the major stages that projects typically execute, often iteratively:
    - Business Understanding
    - Data Acquisition and Understanding
    - Modeling
    - Deployment

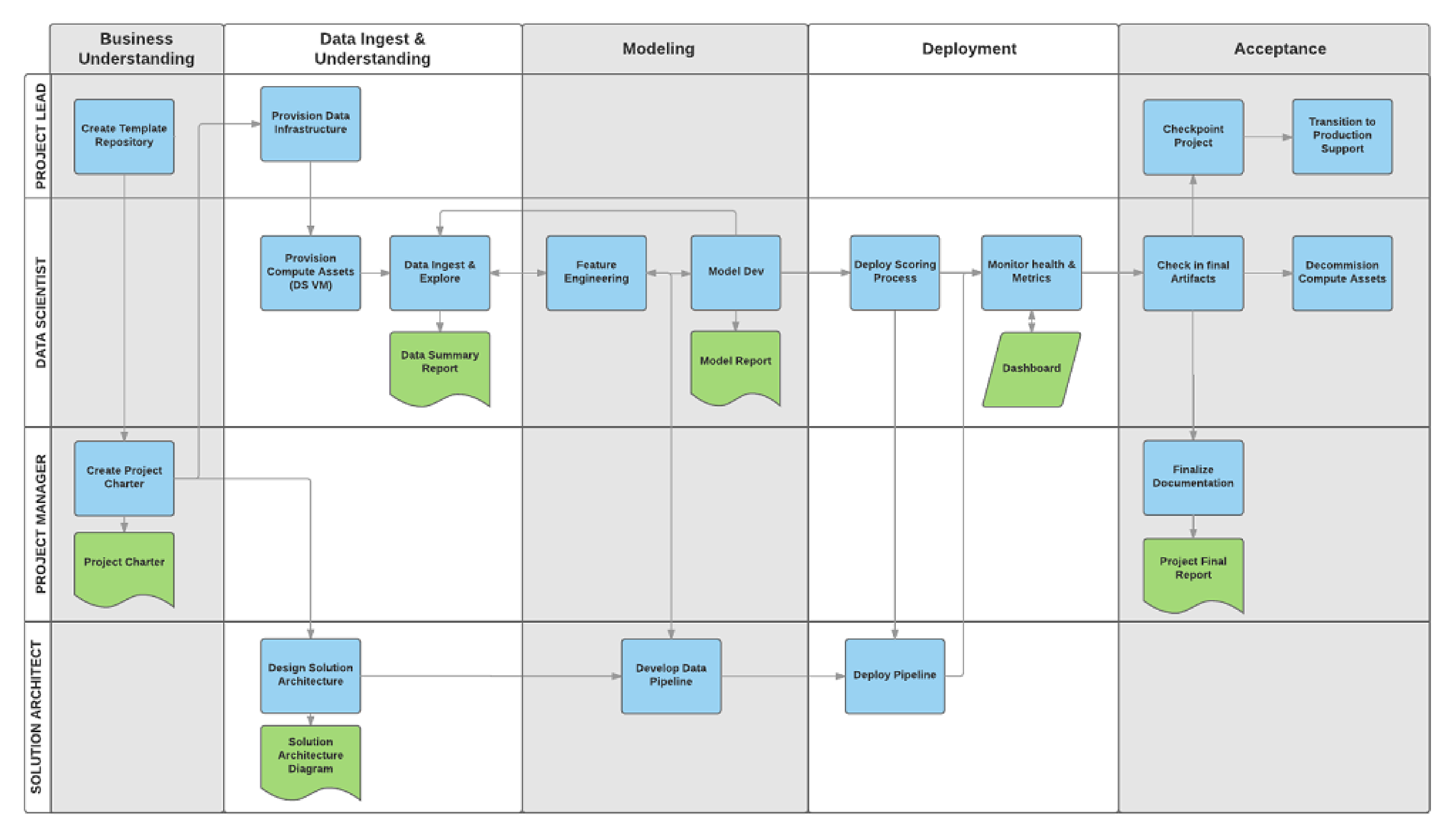
TDSP (contd.)


Data Science Lifecycle
Start
Business Understanding
Transform, Binning
Temporal, Text, Image
Feature Selection
Feature Engineering
Algorithms, Ensemble
Parameter Tuning
Retraining
Model management
Model Training
Cross Validation
Model Reporting
A/B Testing
Model Evaluation
Modeling
Data Acquisition & Understanding
Data Source
On-Premises vs Cloud
Database vs Files
Pipeline
Streaming vs Batch
Low vs High Frequency
Environment
On-premises vs Cloud
Database vs Data Lake vs ..
Small vs Medium vs Big Data
Wrangling, Exploration & Cleaning
Structured vs Unstructured
Data Validation and Cleanup
Visualization
Model Store
Web Services
Intelligent Applications
Deployment
Scoring, Performance monitoring, etc.
Customer Acceptance
End

TDSP (contd.)

- The goals, tasks, and documentation artifacts for each stage of the lifecycle in TDSP are described in the Team Data Science Process lifecycle topic.
- These tasks and artifacts are associated with project roles:
  - Solution architect
  - Project manager
  - Data engineer
  - Data scientist
  - Application developer
  - Project lead

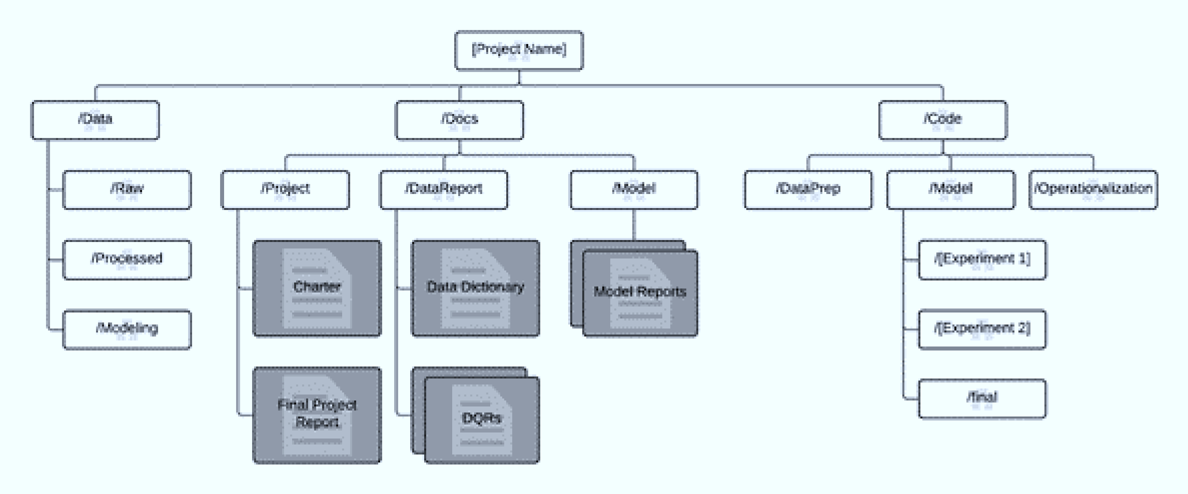
TDSP (contd.)



	Business Understanding	Data Ingest & Understanding	Modeling	Deployment	Acceptance
PROJECT LEAD	Create Template Repository	Provision Data Infrastructure			Checkpoint Project Transition to Production Support
DATA SCIENTIST		Provision Compute Assets (DS VM) Data Ingest & Explore Data Summary Report	Feature Engineering Model Dev Model Report	Deploy Scoring Process Monitor health & Metrics Dashboard	Check in final Artifacts Decommission Compute Assets
PROJECT MANAGER	Create Project Charter Project Charter				Finalize Documentation Project Final Report
SOLUTION ARCHITECT		Design Solution Architecture Solution Architecture Diagram	Develop Data Pipeline	Deploy Pipeline	

TDSP (contd.)

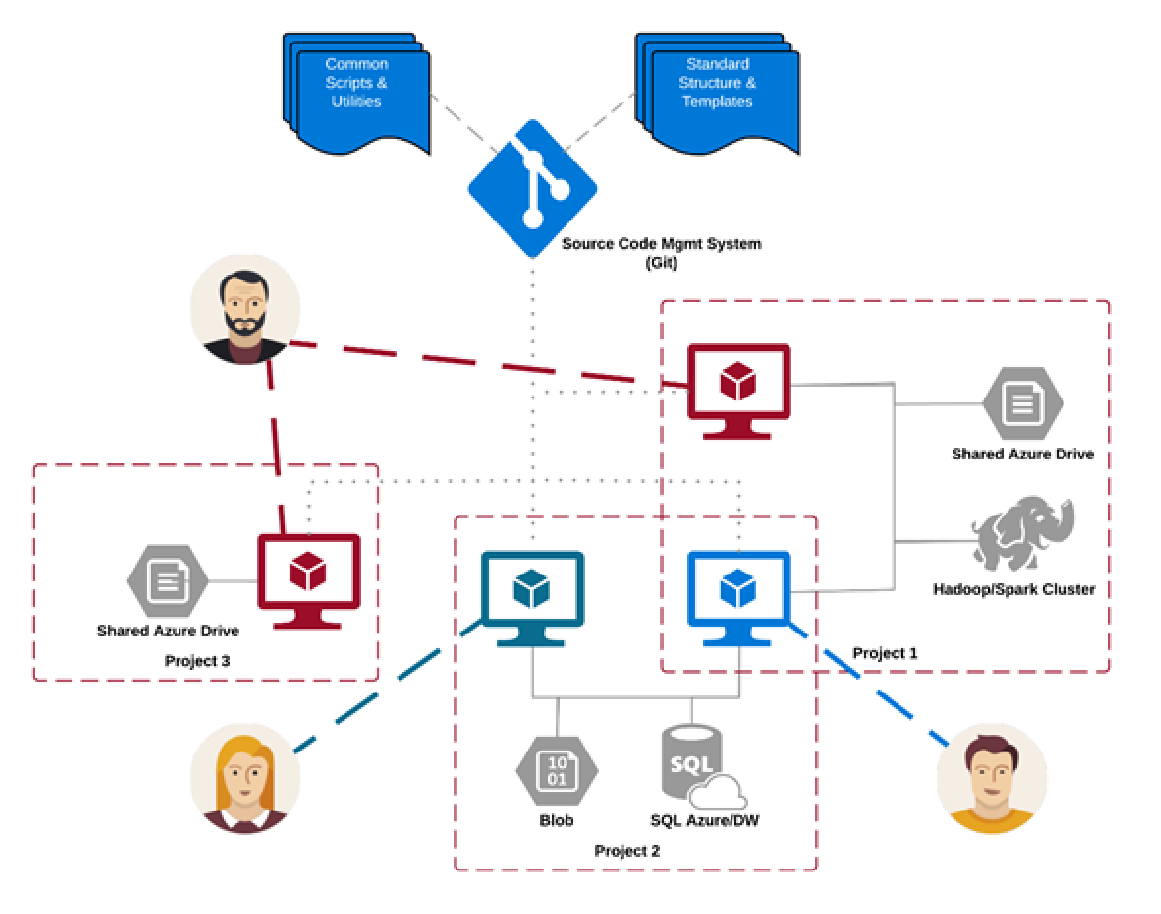
- **Standardized project structure**
  - Having all projects share a directory structure and use templates for project documents makes it easy for the team members to find information about their projects.
  - All code and documents are stored in a version control system (VCS) like Git, TFS, or Subversion to enable team collaboration.
  - Tracking tasks and features in an agile project tracking system like Jira, Rally, and Azure DevOps allows closer tracking of the code for individual features. Such tracking also enables teams to obtain better cost estimates.
  - TDSP recommends creating a separate repository for each project on the VCS for versioning, information security, and collaboration. The standardized structure for all projects helps build institutional knowledge across the organization.



TDSP (contd.)

- Infrastructure and resources for data science projects
  - TDSP provides recommendations for managing shared analytics and storage infrastructure such as:
    - cloud file systems for storing datasets
    - databases
    - big data (SQL or Spark) clusters
    - machine learning service
  - The analytics and storage infrastructure, where raw and processed datasets are stored, may be in the cloud or on-premises.
  - This infrastructure enables reproducible analysis.
  - It also avoids duplication, which may lead to inconsistencies and unnecessary infrastructure costs. Tools are provided to provision the shared resources, track them, and allow each team member to connect to those resources securely.
  - It is also a good practice to have project members create a consistent compute environment. Different team members can then replicate and validate experiments.

TDSP (contd.)



Here is an example of a team working on multiple projects and sharing various cloud analytics infrastructure components.

TDSP (contd.)

- Tools and utilities for project execution
  - Introducing processes in most organizations is challenging.
  - Tools provided to implement the data science process and lifecycle help lower the barriers to and increase the consistency of their adoption.
  - TDSP provides an initial set of tools and scripts to jump-start adoption of TDSP within a team.
  - It also helps automate some of the common tasks in the data science lifecycle such as data exploration and baseline modeling.
  - There is a well-defined structure provided for individuals to contribute shared tools and utilities into their team's shared code repository.
  - These resources can then be leveraged by other projects within the team or the organization.
  - Microsoft provides extensive tooling inside Azure Machine Learning supporting both open-source (Python, R, ONNX, and common deep-learning frameworks) and also Microsoft's own tooling (AutoML).

TDSP (contd.)

Benefits

- Elaborated Documentation
- Agile
- Familiar
- Data Science Native
- Flexible
- Detailed
- Free Templates

Short comings

- Fixed Sprints: TDSP leverages fixed-length planning sprints which many data scientists struggle with.
- Some Inconsistencies: Not all of Microsoft’s documentation is consistent.

TDSP (contd.)

Check ODSC.com for conference updates and OpenDataScience.com for the latest Articles

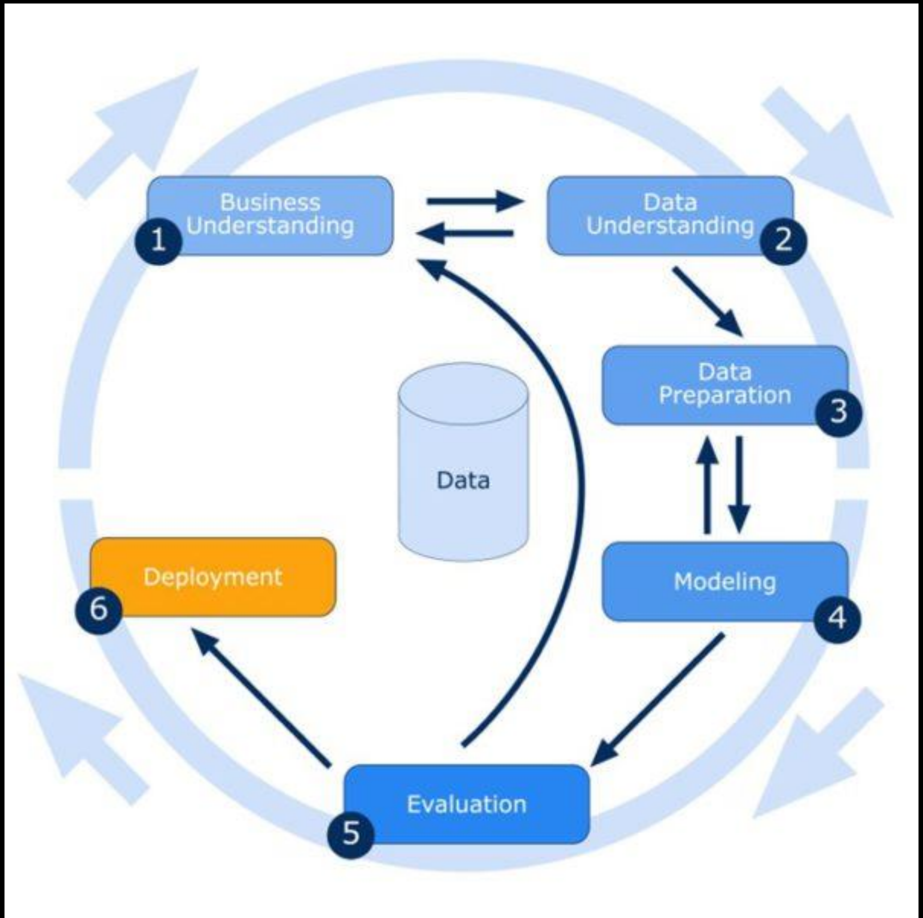
The Team Data Science Process Workshop

[illegible]

Microsoft Cloud and AI

CRISP-DM

- The CRoss Industry Standard Process for Data Mining (CRISP-DM) is a process model that serves as the base for a data science process. It has six sequential phases:
  - Business understanding
  - Data understanding
  - Data preparation
  - Modeling
  - Evaluation
  - Deployment



https://www.datascience-pm.com/crisp-dm-2/
https://web.archive.org/web/20220401041957/https://www.the-modeling-agency.com/crisp-dm.pdf

CRISP-DM (contd.)

I. Business Understanding

- Any good project starts with a deep understanding of the customer’s needs. Data mining projects are no exception and CRISP-DM recognizes this.
- The Business Understanding phase focuses on understanding the objectives and requirements of the project. Aside from the third task, the three other tasks in this phase are foundational project management activities that are universal to most projects:
  1. Determine business objectives: You should first “thoroughly understand, from a business perspective, what the customer really wants to accomplish.” (CRISP-DM Guide) and then define business success criteria.
  2. Assess situation: Determine resources availability, project requirements, assess risks and contingencies, and conduct a cost-benefit analysis.
  3. Determine data mining goals: In addition to defining the business objectives, you should also define what success looks like from a technical data mining perspective.
  4. Produce project plan: Select technologies and tools and define detailed plans for each project phase.

CRISP-DM (contd.)

II. Data Understanding

- Next is the Data Understanding phase. Adding to the foundation of Business Understanding, it drives the focus to identify, collect, and analyze the data sets that can help you accomplish the project goals. This phase also has four tasks:
  1. Collect initial data: Acquire the necessary data and (if necessary) load it into your analysis tool.
  2. Describe data: Examine the data and document its surface properties like data format, number of records, or field identities.
  3. Explore data: Dig deeper into the data. Query it, visualize it, and identify relationships among the data.
  4. Verify data quality: How clean/dirty is the data? Document any quality issues.

CRISP-DM (contd.)

III. Data Preparation

- A common rule of thumb is that 80% of the project is data preparation.
- This phase, which is often referred to as “data munging”, prepares the final data set(s) for modeling. It has five tasks:
  1. Select data: Determine which data sets will be used and document reasons for inclusion/exclusion.
  2. Clean data: Often this is the lengthiest task. Without it, you’ll likely fall victim to garbage-in, garbage-out. A common practice during this task is to correct, impute, or remove erroneous values.
  3. Construct data: Derive new attributes that will be helpful. For example, derive someone’s body mass index from height and weight fields.
  4. Integrate data: Create new data sets by combining data from multiple sources.
  5. Format data: Re-format data as necessary. For example, you might convert string values that store numbers to numeric values so that you can perform mathematical operations.

CRISP-DM (contd.)

IV. Modeling

- What is widely regarded as data science’s most exciting work is also often the shortest phase of the project.
- Here you’ll likely build and assess various models based on several different modeling techniques. This phase has four tasks:
  1. **Select modeling techniques**: Determine which algorithms to try (e.g. regression, neural net).
  2. **Generate test design**: Pending your modeling approach, you might need to split the data into training, test, and validation sets.
  3. **Build model**: As glamorous as this might sound, this might just be executing a few lines of code like “reg = LinearRegression().fit(X, y)”.
  4. **Assess model**: Generally, multiple models are competing against each other, and the data scientist needs to interpret the model results based on domain knowledge, the pre-defined success criteria, and the test design.

CRISP-DM (contd.)

V. Evaluation

- Whereas the Assess Model task of the Modeling phase focuses on technical model assessment, the Evaluation phase looks more broadly at which model best meets the business and what to do next. This phase has three tasks:
  1. **Evaluate results**: Do the models meet the business success criteria? Which one(s) should we approve for the business?
  2. **Review process**: Review the work accomplished. Was anything overlooked? Were all steps properly executed? Summarize findings and correct anything if needed.
  3. **Determine next steps**: Based on the previous three tasks, determine whether to proceed to deployment, iterate further, or initiate new projects.

CRISP-DM (contd.)

VI. Deployment

"Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise."

– CRISP-DM Guide

- A model is not particularly useful unless the customer can access its results. The complexity of this phase varies widely. This final phase has four tasks:
  - **Plan deployment**: Develop and document a plan for deploying the model.
  - **Plan monitoring and maintenance**: Develop a thorough monitoring and maintenance plan to avoid issues during the operational phase (or post-project phase) of a model.
  - **Produce final report**: The project team documents a summary of the project which might include a final presentation of data mining results.
  - **Review project**: Conduct a project retrospective about what went well, what could have been better, and how to improve in the future.

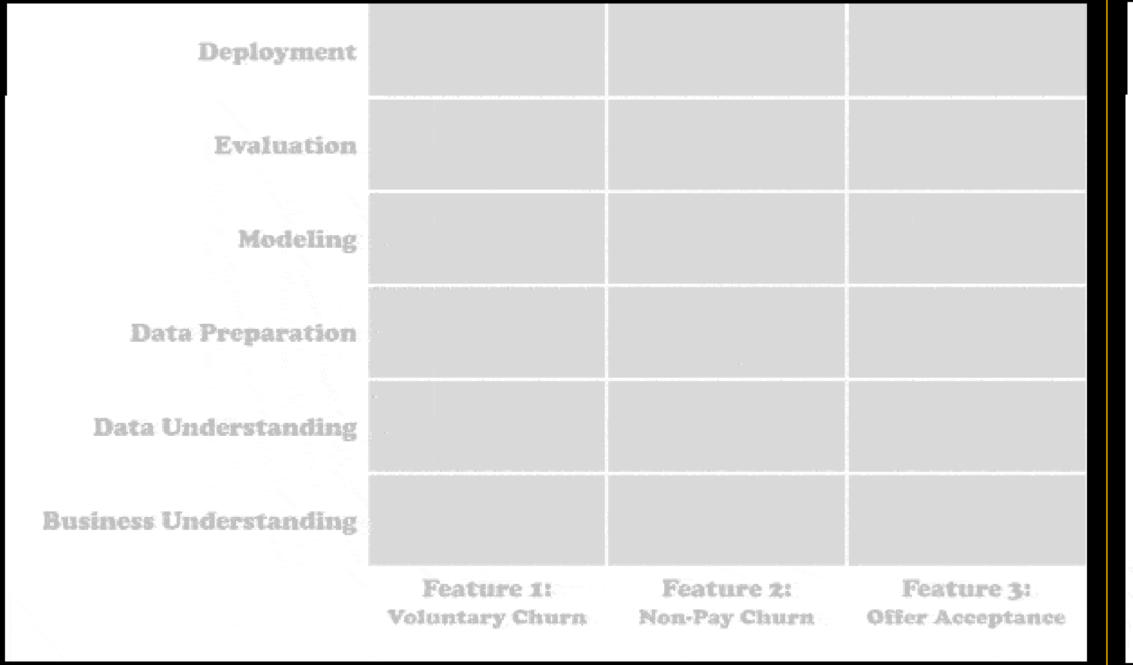
Is CRISP-DM Agile or Waterfall?

- Some argue that it is flexible and agile and while others see CRISP-DM as rigid. What really matters is how you implement it.
- **Waterfall**
  - On one hand, many view CRISP-DM as a rigid waterfall process – in part because of its reporting requirements are excessive for most projects. Moreover, the guide states in the business understanding phase that “the project plan contains detailed plans for each phase” – a hallmark aspect of traditional waterfall approaches that require detailed, upfront planning.
  - Indeed, if you follow CRISP-DM precisely (defining detailed plans for each phase at the project start and include every report) and choose not to iterate frequently, then you’re operating more of a waterfall process.
- **Agile**
  - On the other hand, CRISP-DM indirectly advocates agile principles and practices by stating: “The sequence of the phases is not rigid. Moving back and forth between different phases is always required. The outcome of each phase determines which phase, or particular task of a phase, has to be performed next.”
  - Thus if you follow CRISP-DM in a more flexible way, iterate quickly, and layer in other agile processes, you’ll wind up with an agile approach.

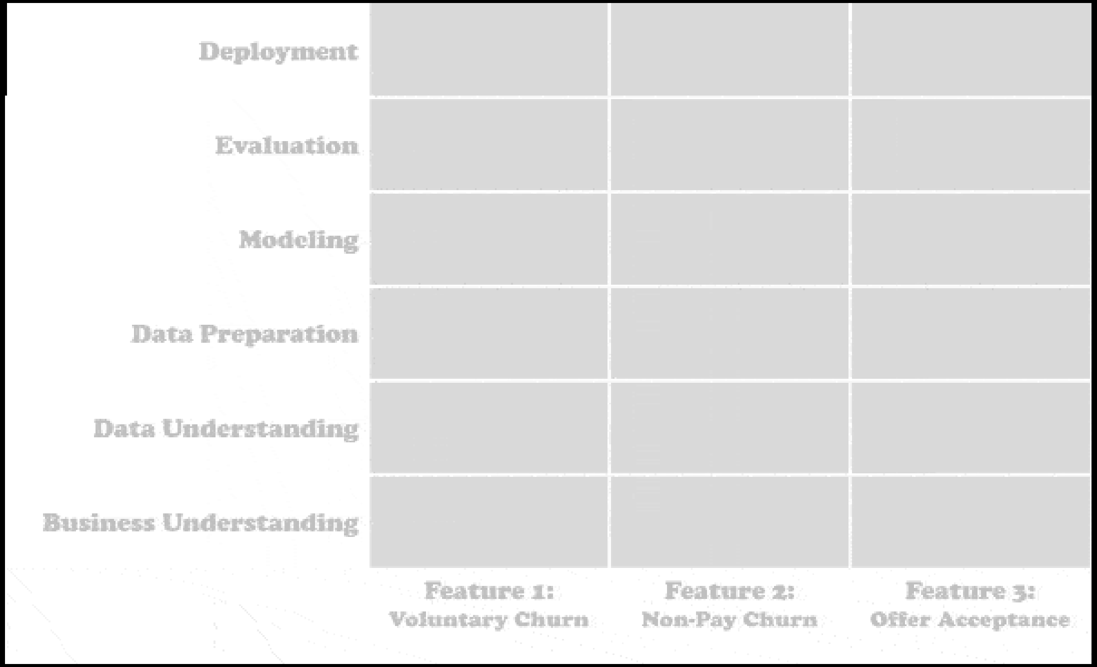
Example: *To illustrate how CRISP-DM could be implemented in either an Agile or waterfall manner, imagine a churn project with three deliverables: a voluntary churn model, a non-pay disconnect churn model, and a propensity to accept a retention-focused offer.*

CRISP-DM (contd.)

Waterfall: Horizontal Slicing



Agile: Vertical Slicing



CRISP-DM (contd.)

Which is better?

- When possible, take an agile approach and slice vertically so that:
  - Stakeholders get value sooner
  - Stakeholders can provide meaningful feedback
  - The data scientists can assess model performance earlier
  - The project team can adjust the plan based on stakeholder feedback

CRISP-DM (contd.)

Benefits

- Generalizable
- Common Sense
- Adoptable
- Right Start
- Strong Finish
- Flexible

Shortcomings

- Rigid
- Documentation Heavy
- Not Modern
- Not a Project Management Approach

Review of statistics and probability

Probability

- Probability is a measure of the likelihood of an event to occur. Many events cannot be predicted with total certainty.
- We can predict only the chance of an event to occur i.e., how likely they are going to happen, using it.
- Probability can range from 0 to 1, where 0 means the event to be an impossible one and 1 indicates a certain event.
- Probability for Class 10 is an important topic for the students which explains all the basic concepts of this topic.
- The probability of all the events in a sample space adds up to 1.

Probability

- For example, when we toss a coin, either we get Head OR Tail, only two possible outcomes are possible (H, T).
- But when two coins are tossed then there will be four possible outcomes, i.e {(H, H), (H, T), (T, H), (T, T)}.

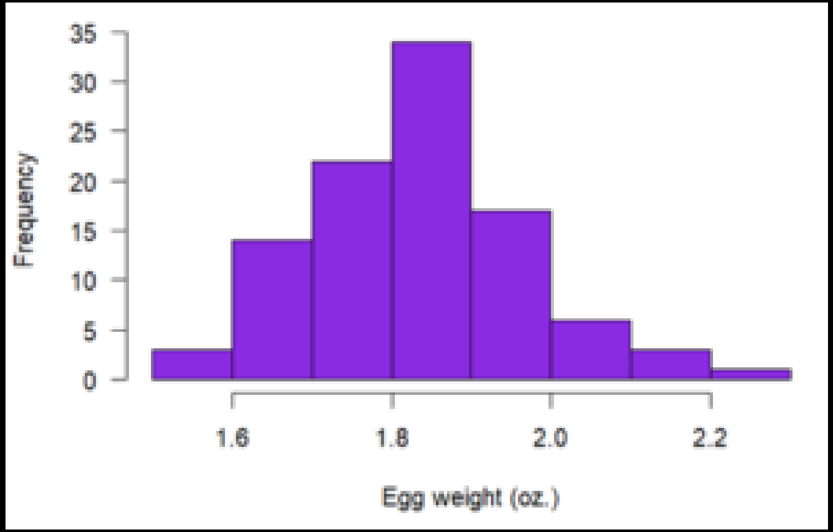
$$Probability\ of\ event\ to\ happen\ P(E) = \frac{Number\ of\ favourable\ outcomes}{Total\ Number\ of\ Outcomes}$$

Probability Distribution

- A probability distribution is an idealized frequency distribution.
- A frequency distribution describes a specific sample or dataset. It’s the number of times each possible value of a variable occurs in the dataset.
- The number of times a value occurs in a sample is determined by its probability of occurrence.
- The higher the probability of a value, the higher its frequency in a sample.
- More specifically, the probability of a value is its relative frequency in an infinitely large sample.
- Infinitely large samples are impossible in real life, so probability distributions are theoretical. They’re idealized versions of frequency distributions that aim to describe the population the sample was drawn from.
- Probability distributions are used to describe the populations of real-life variables, like coin tosses or the weight of chicken eggs.

Probability Distribution (contd.)

- Imagine that an egg farmer wants to know the probability of an egg from her farm being a certain size.
- The farmer weighs 100 random eggs and describes their frequency distribution using a histogram:



Probability Distribution (contd.)

- Thus, a probability distribution is a mathematical function that describes the probability of different possible values of a variable.
- It is a mathematical description of a random phenomenon in terms of its sample space and the probabilities of events (subsets of the sample space).
- Probability distributions are often depicted using graphs or probability tables.
- There are many types of probability distributions, including the normal distribution, binomial distribution, Poisson distribution, and exponential distribution

Compound events

- A compound event is an event that includes two or more simple events.
- Simple events are events that can have only one outcome, while compound events can have multiple different outcomes.
- Compound events can be made up of a number of independent events (*events in which the outcome of one event has no effect on the probability of the other*) or dependent events (*events in which the outcome of one event affects the probability of another*).

Compound Probability

- A compound probability is the probability of a compound event.
- Generally, it is the ratio of favorable outcomes to the total number of outcomes within the sample space of the compound event and can be calculated using one of two rules: the addition rule *and* the multiplication rule.

Compound Probability

- Addition Rule:

  For mutually exclusive

$$P(A\ or\ B) = P(A) + P(B)$$

  For non-mutually exclusive events

$$P(A\ or\ B) = P(A) + P(B) - P(A\ and\ B)$$

- Product Rule:

  For independent events

$$P(A\ and\ B) = P(A).P(B)$$

  For dependent events

$$P(A\ and\ B) = P(A).P(B|A)\ or\ P(B).P(A|B)$$

Centrality Measures

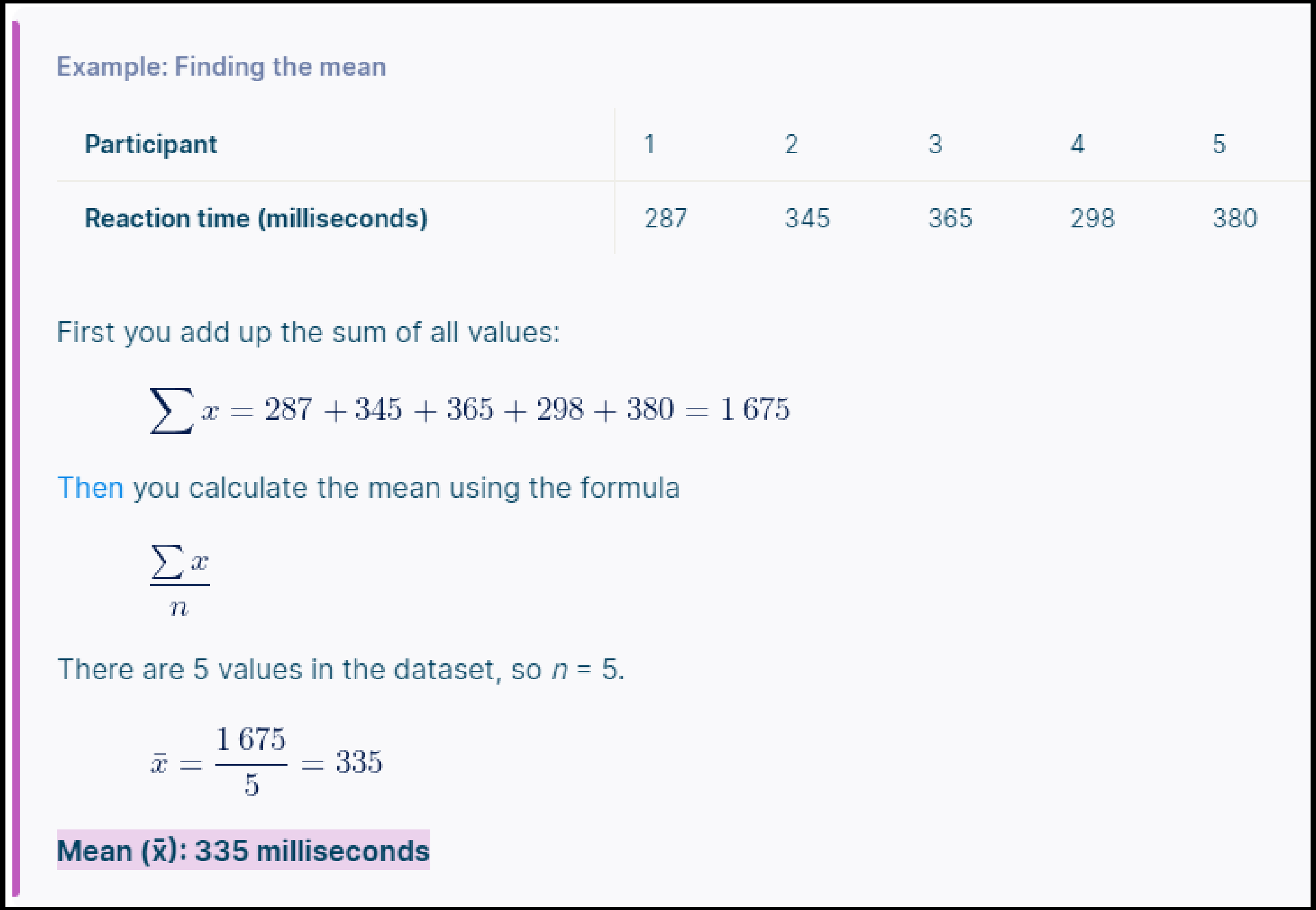
- Measures of central tendency help you find the middle, or the average, of a dataset.
- The 3 most common measures of central tendency are the mode, median, and mean.
    - Mode: the most frequent value.
    - Median: the middle number in an ordered dataset.
    - Mean: the sum of all values divided by the total number of values.

Central Tendency | Understanding the Mean, Median & Mode (scribbr.com)

Centrality Measures (contd.)

- Mean
  - The arithmetic mean of a dataset (which is different from the geometric mean) is the sum of all values divided by the total number of values.
  - It’s the most commonly used measure of central tendency because all values are used in the calculation.
  - Outliers can significantly increase or decrease the mean when they are included in the calculation.
  - Since all values are used to calculate the mean, it can be affected by extreme outliers.
  - An outlier is a value that differs significantly from the others in a dataset.

Centrality Measures (contd.)



Example: Finding the mean

Participant	1	2	3	4	5
Reaction time (milliseconds)	287	345	365	298	380

First you add up the sum of all values:

$$\sum x = 287 + 345 + 365 + 298 + 380 = 1\,675$$

Then you calculate the mean using the formula

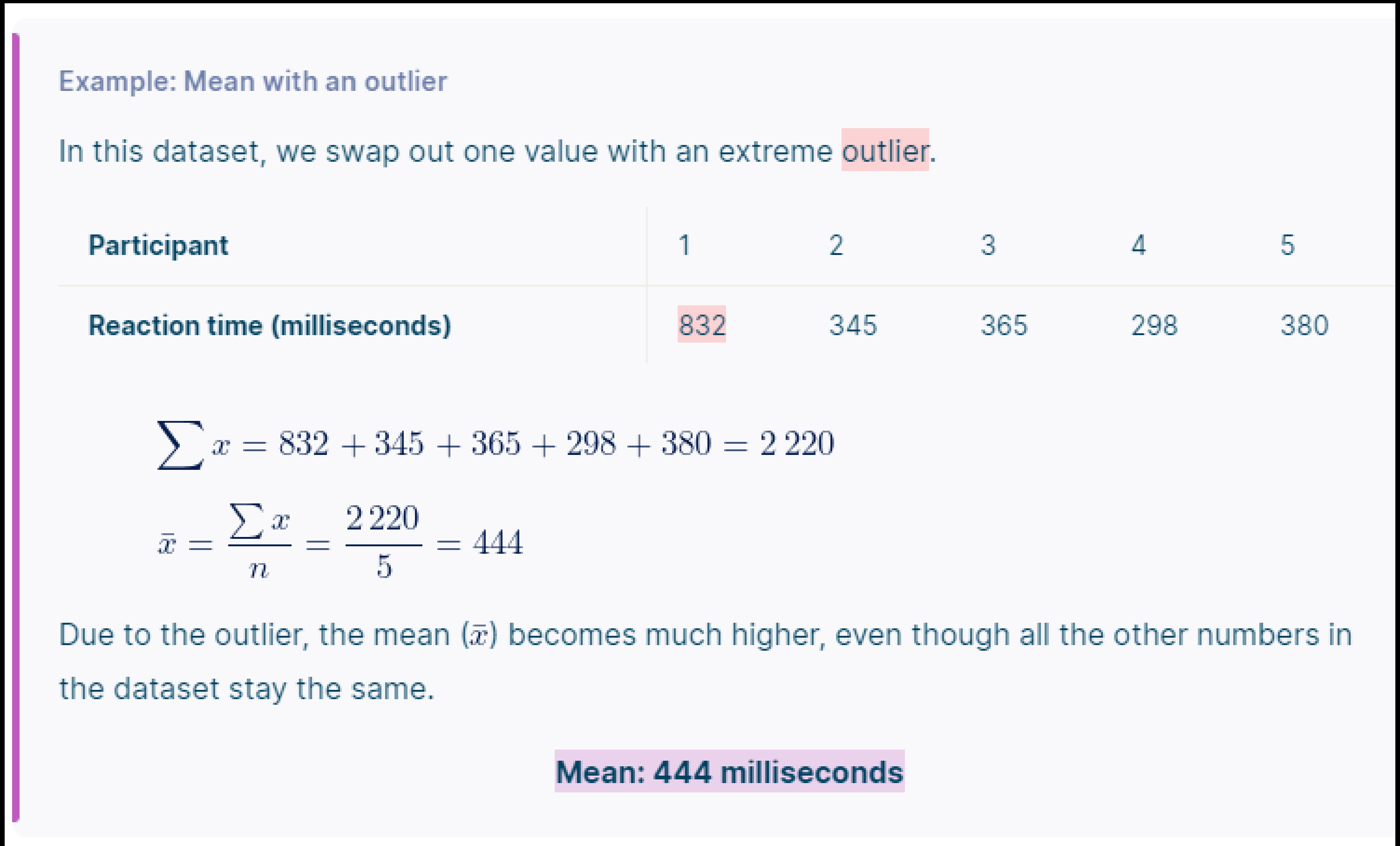
$$\frac{\sum x}{n}$$

There are 5 values in the dataset, so n = 5.

$$\bar{x} = \frac{1\,675}{5} = 335$$

Mean (x̄): 335 milliseconds

Centrality Measures (contd.)



Example: Mean with an outlier

In this dataset, we swap out one value with an extreme outlier.

Participant	1	2	3	4	5
Reaction time (milliseconds)	832	345	365	298	380

$$\sum x = 832 + 345 + 365 + 298 + 380 = 2\,220$$

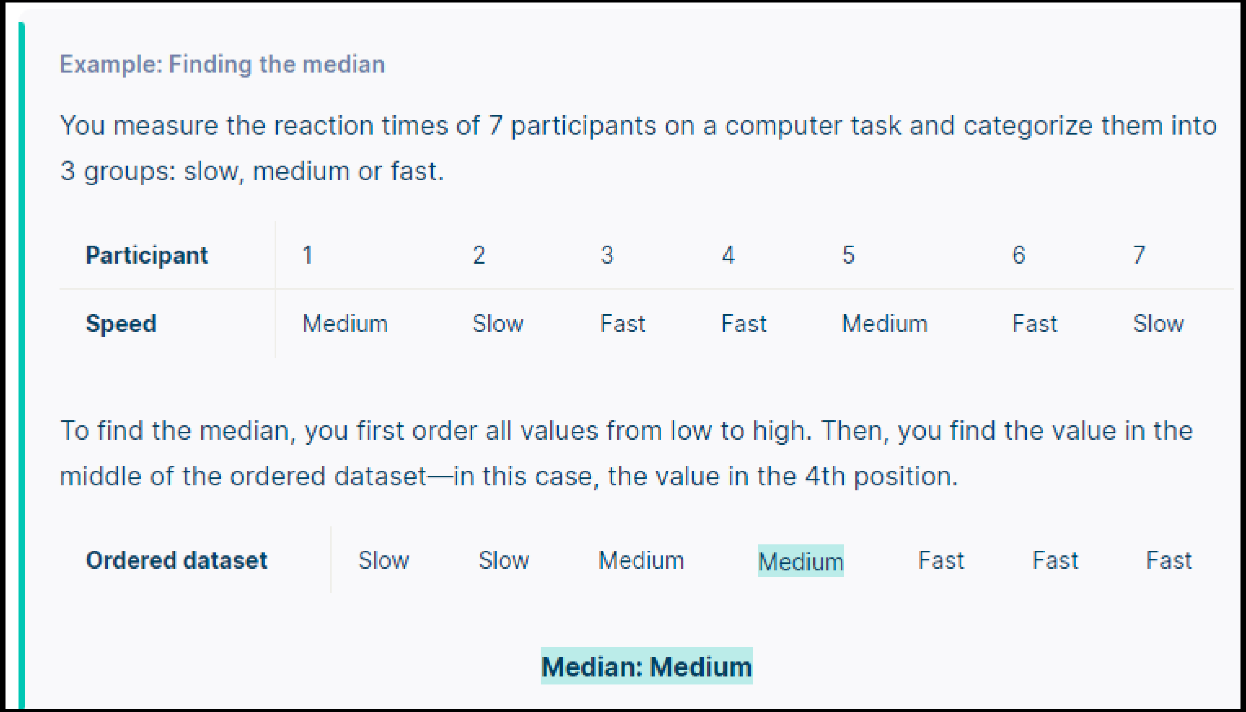
$$\bar{x} = \frac{\sum x}{n} = \frac{2\,220}{5} = 444$$

Due to the outlier, the mean ($\bar{x}$) becomes much higher, even though all the other numbers in the dataset stay the same.

Mean: 444 milliseconds

Centrality Measures (contd.)

- Median
  - The median of a dataset is the value that's exactly in the middle when it is ordered from low to high.



Example: Finding the median

You measure the reaction times of 7 participants on a computer task and categorize them into 3 groups: slow, medium or fast.

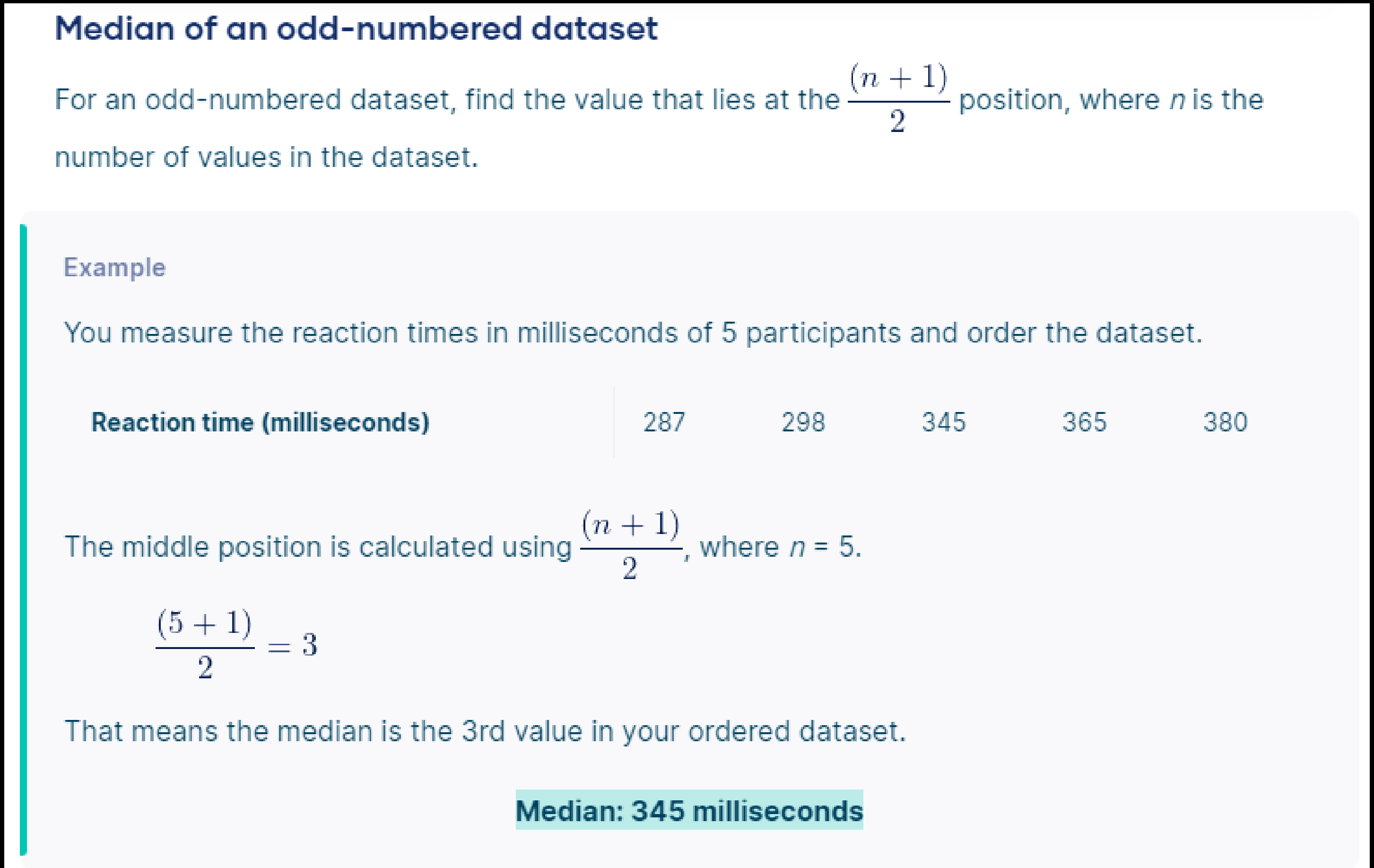
Participant	1	2	3	4	5	6	7
Speed	Medium	Slow	Fast	Fast	Medium	Fast	Slow

To find the median, you first order all values from low to high. Then, you find the value in the middle of the ordered dataset—in this case, the value in the 4th position.

Ordered dataset	Slow	Slow	Medium	Medium	Fast	Fast	Fast

Median: Medium

Centrality Measures (contd.)



Median of an odd-numbered dataset

For an odd-numbered dataset, find the value that lies at the $\frac{(n+1)}{2}$ position, where n is the number of values in the dataset.

Example

You measure the reaction times in milliseconds of 5 participants and order the dataset.

Reaction time (milliseconds)	287	298	345	365	380

The middle position is calculated using $\frac{(n+1)}{2}$, where n = 5.

$$\frac{(5+1)}{2} = 3$$

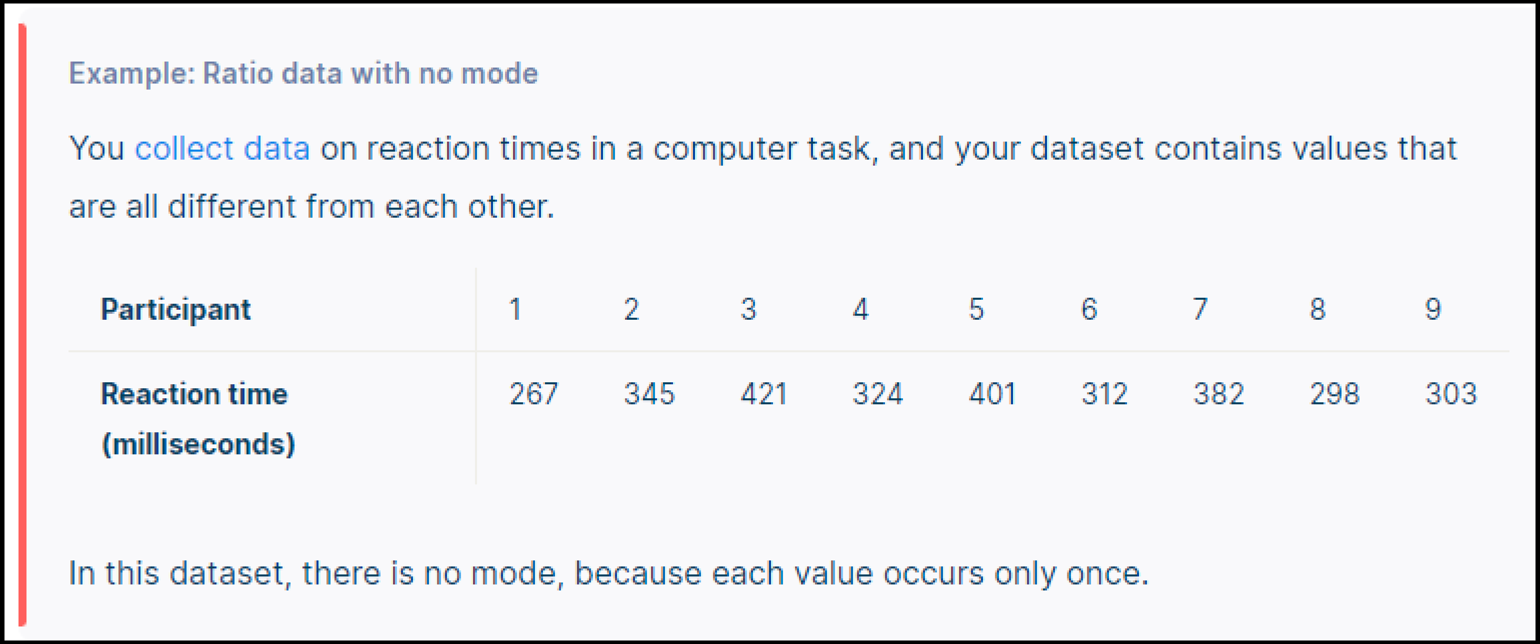
That means the median is the 3rd value in your ordered dataset.

Median: 345 milliseconds

Centrality Measures (contd.)

- Mode
  - The mode is the most frequently occurring value in the dataset. It’s possible to have no mode, one mode, or more than one mode.
  - To find the mode, sort your dataset numerically or categorically and select the response that occurs most frequently.
  - The mode is most applicable to data from a nominal level of measurement. Nominal data is classified into mutually exclusive categories, so the mode tells you the most popular category.
  - For continuous variables or ratio levels of measurement, the mode may not be a helpful measure of central tendency.

Centrality Measures (contd.)



Example: Ratio data with no mode

You collect data on reaction times in a computer task, and your dataset contains values that are all different from each other.

Participant	1	2	3	4	5	6	7	8	9
Reaction time (milliseconds)	267	345	421	324	401	312	382	298	303

In this dataset, there is no mode, because each value occurs only once.

Variability Measures

- Variability describes how far apart data points lie from each other and from the center of a distribution.
- Along with measures of central tendency, measures of variability give you descriptive statistics that summarize your data.
- Variability is also referred to as spread, scatter or dispersion. It is most commonly measured with the following:
    - Range: the difference between the highest and lowest values
    - Interquartile range: the range of the middle half of a distribution
    - Standard deviation: average distance from the mean
    - Variance: average of squared distances from the mean

Variability Measures (contd.)

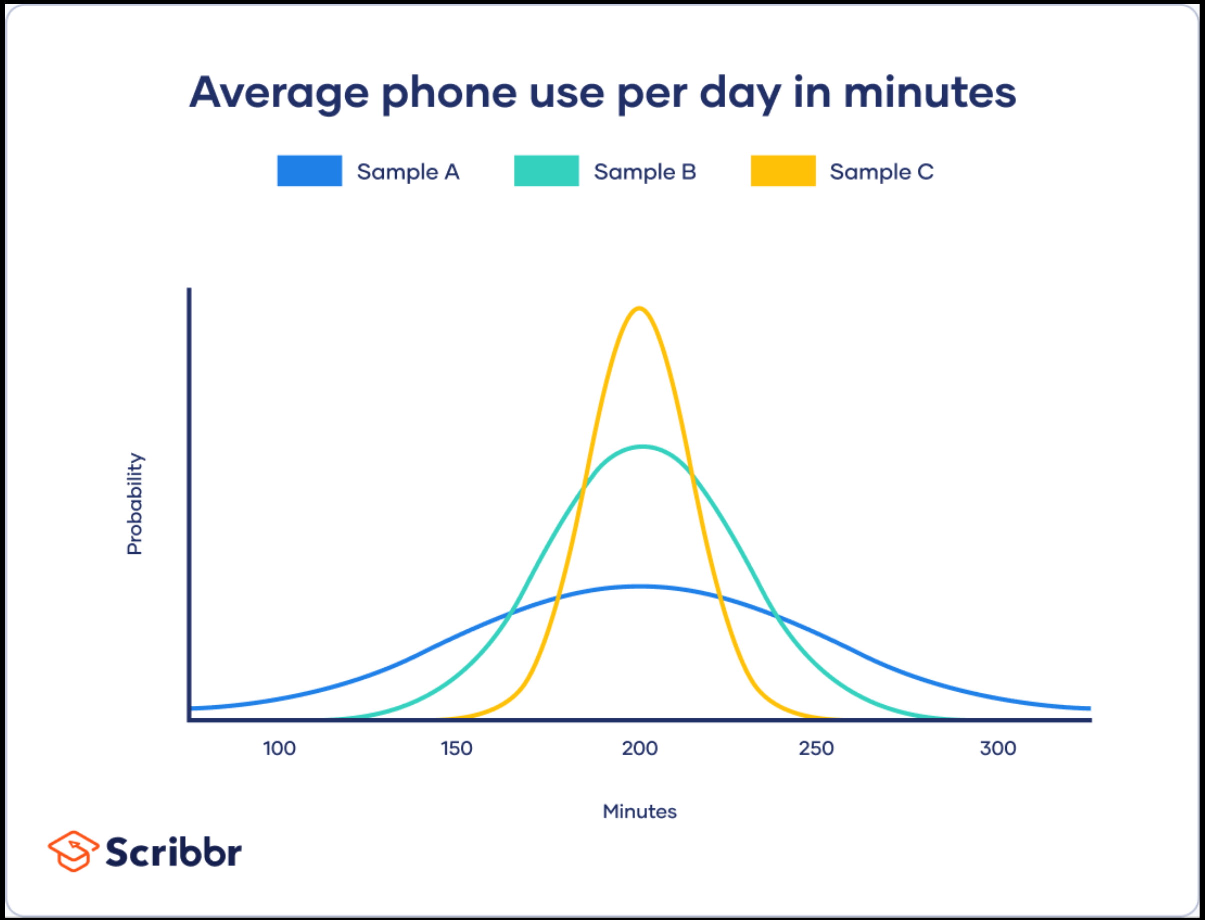
- Why does it matter?

While the central tendency, or average, tells you where most of your points lie, variability summarizes how far apart they are. This is important because the amount of variability determines how well you can generalize results from the sample to your population.

Low variability is ideal because it means that you can better predict information about the population based on sample data. High variability means that the values are less consistent, so it’s harder to make predictions.

Data sets can have the same central tendency but different levels of variability or vice versa. If you know only the central tendency or the variability, you can’t say anything about the other aspect. Both of them together give you a complete picture of your data.

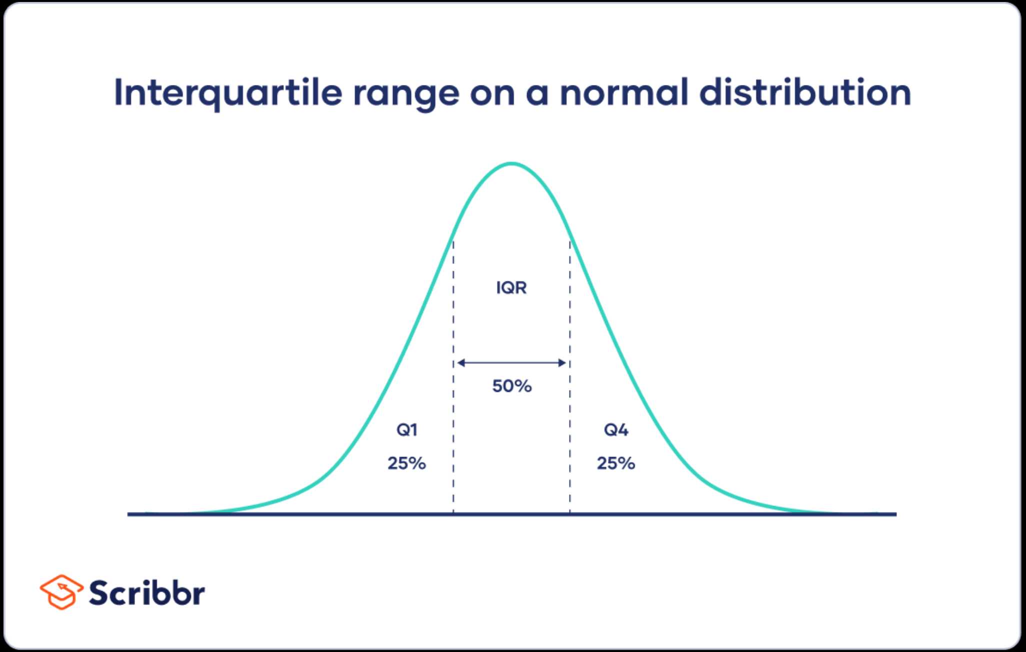
Variability Measures (contd.)



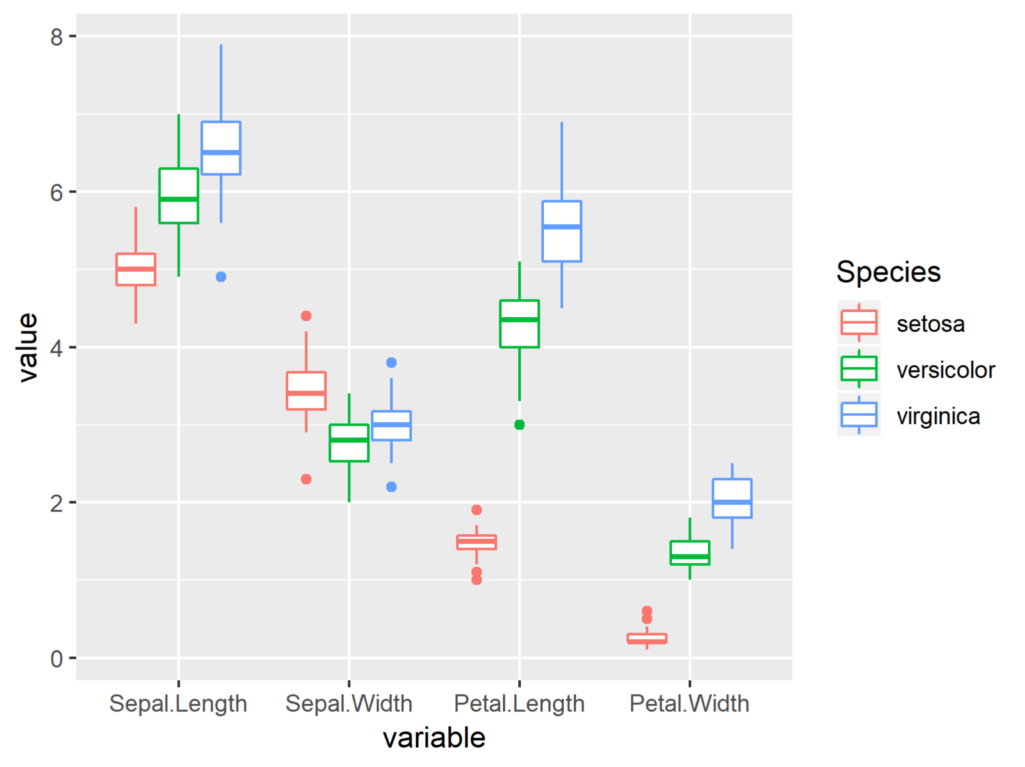
Although the data follows a normal distribution, each sample has different spreads. Sample A has the largest variability while Sample C has the smallest variability.

Variability Measures (contd.)

- Inter Quartile Range
  - The interquartile range gives you the spread of the middle of your distribution.
  - For any distribution that’s ordered from low to high, the interquartile range contains half of the values.
  - While the first quartile (Q1) contains the first 25% of values, the fourth quartile (Q4) contains the last 25% of values.
  - The interquartile range is the third quartile (Q3) minus the first quartile (Q1). This gives us the range of the middle half of a data set.



Variability Measures (contd.)



Five-number summary
Every distribution can be organized using a five-number summary:

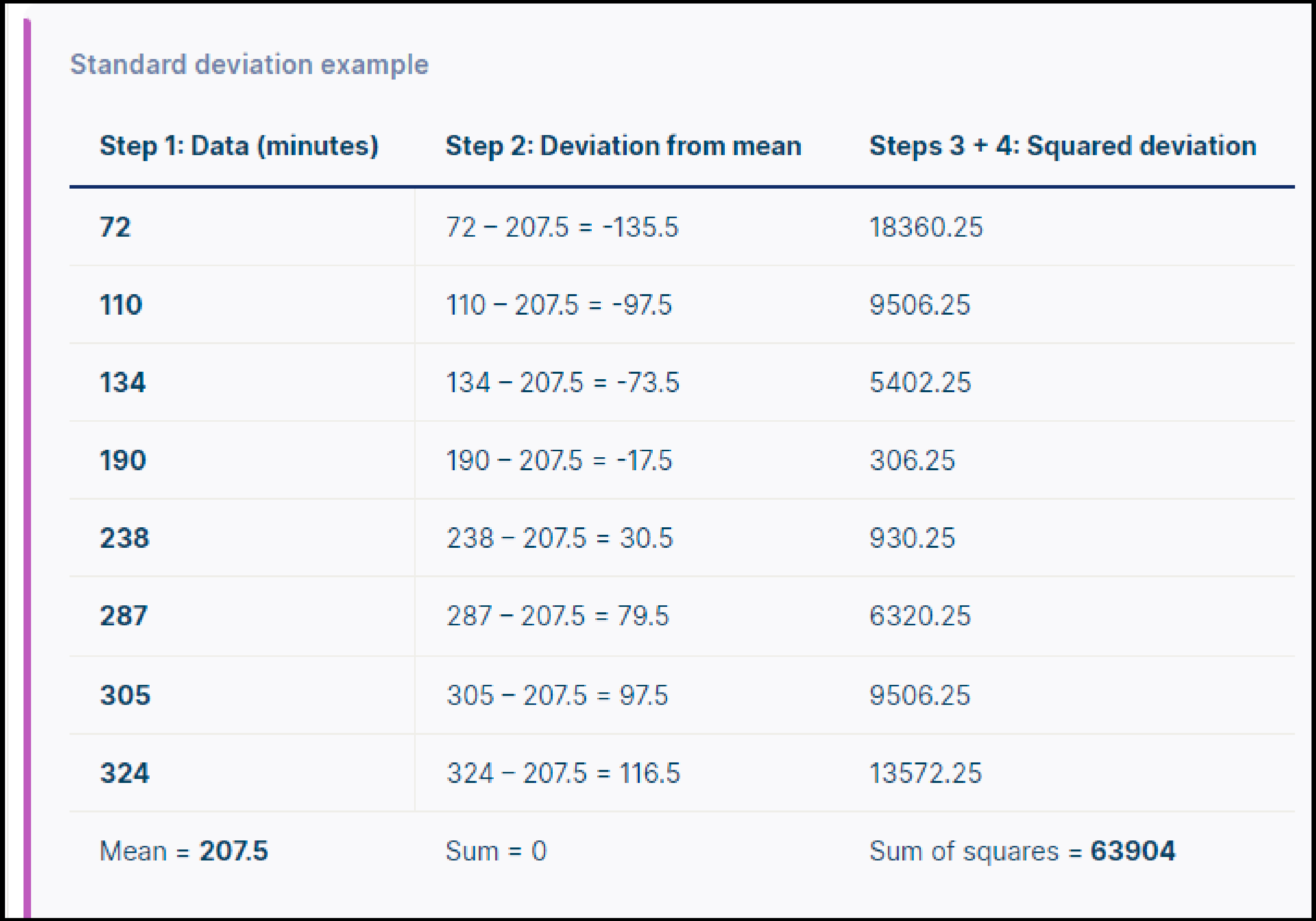
- Lowest value
- Q1: 25th percentile
- Q2: the median
- Q3: 75th percentile
- Highest value (Q4)

These five-number summaries can be easily visualized using box and whisker plots.

Variability Measures (contd.)

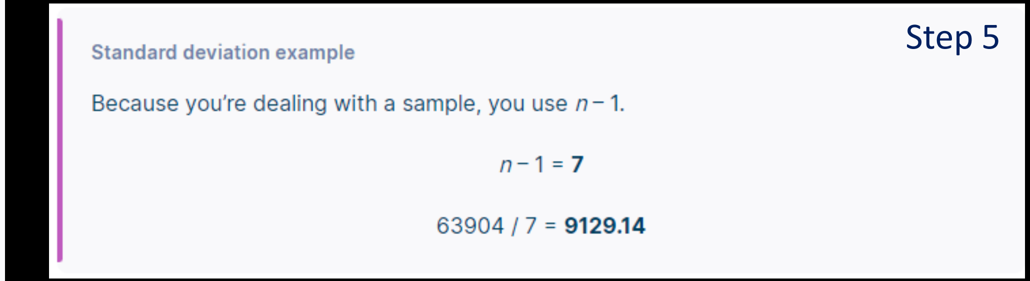
- Standard deviation
  - The standard deviation is the average amount of variability in your dataset.
  - It tells you, on average, how far each score lies from the mean. The larger the standard deviation, the more variable the data set is.
  - There are six steps for finding the standard deviation by hand:
    - List each score and find their mean.
    - Subtract the mean from each score to get the deviation from the mean.
    - Square each of these deviations.
    - Add up all of the squared deviations.
    - Divide the sum of the squared deviations by n – 1 (for a sample) or N (for a population).
    - Find the square root of the number you found.

Variability Measures (contd.)



Standard deviation example

Step 1: Data (minutes)	Step 2: Deviation from mean	Steps 3 + 4: Squared deviation
72	72 – 207.5 = -135.5	18360.25
110	110 – 207.5 = -97.5	9506.25
134	134 – 207.5 = -73.5	5402.25
190	190 – 207.5 = -17.5	306.25
238	238 – 207.5 = 30.5	930.25
287	287 – 207.5 = 79.5	6320.25
305	305 – 207.5 = 97.5	9506.25
324	324 – 207.5 = 116.5	13572.25
Mean = **207.5**	Sum = 0	Sum of squares = **63904**



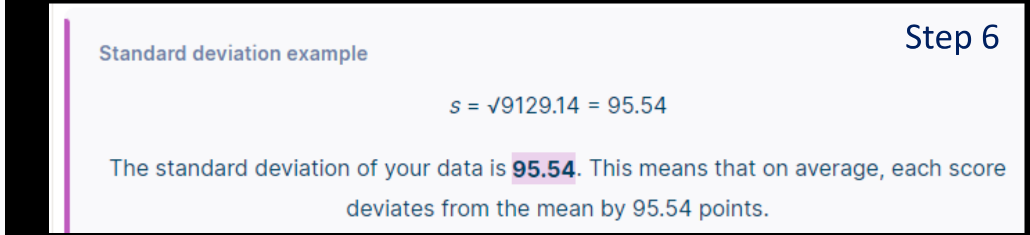
Step 5

Standard deviation example

Because you're dealing with a sample, you use $n - 1$.

$n - 1 =$ **7**

63904 / 7 = **9129.14**



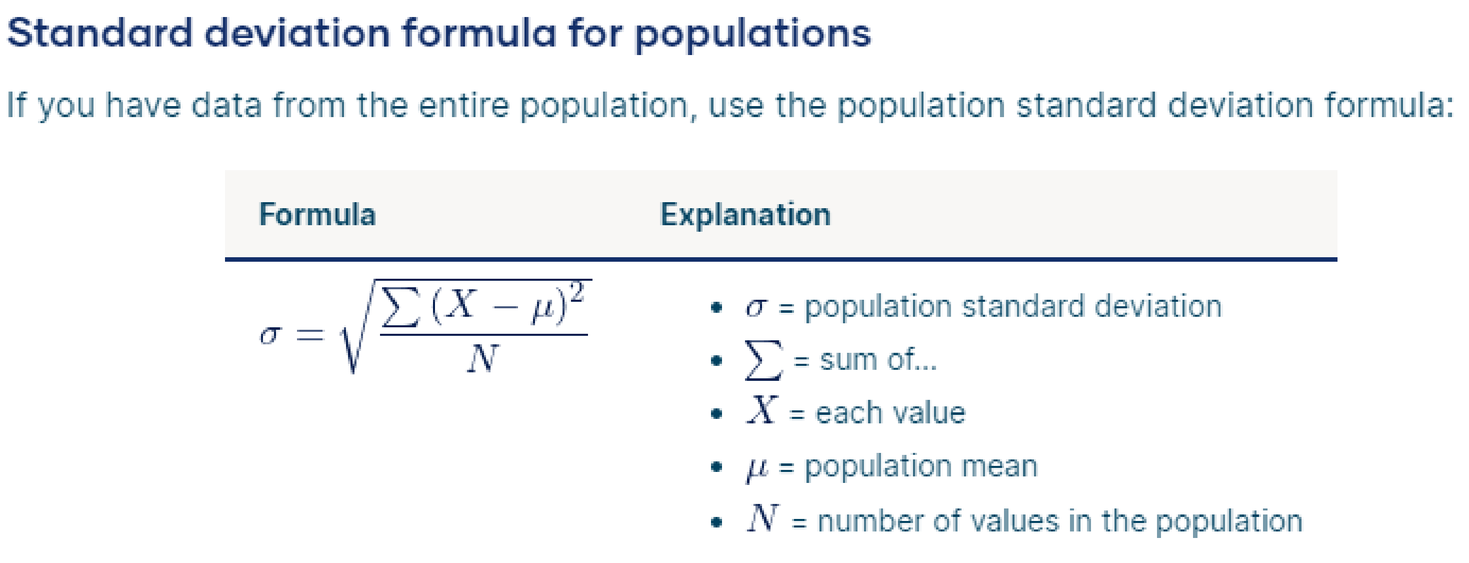
Step 6

Standard deviation example

$s = \sqrt{9129.14} = 95.54$

The standard deviation of your data is **95.54**. This means that on average, each score deviates from the mean by 95.54 points.

Variability Measures (contd.)



Standard deviation formula for populations

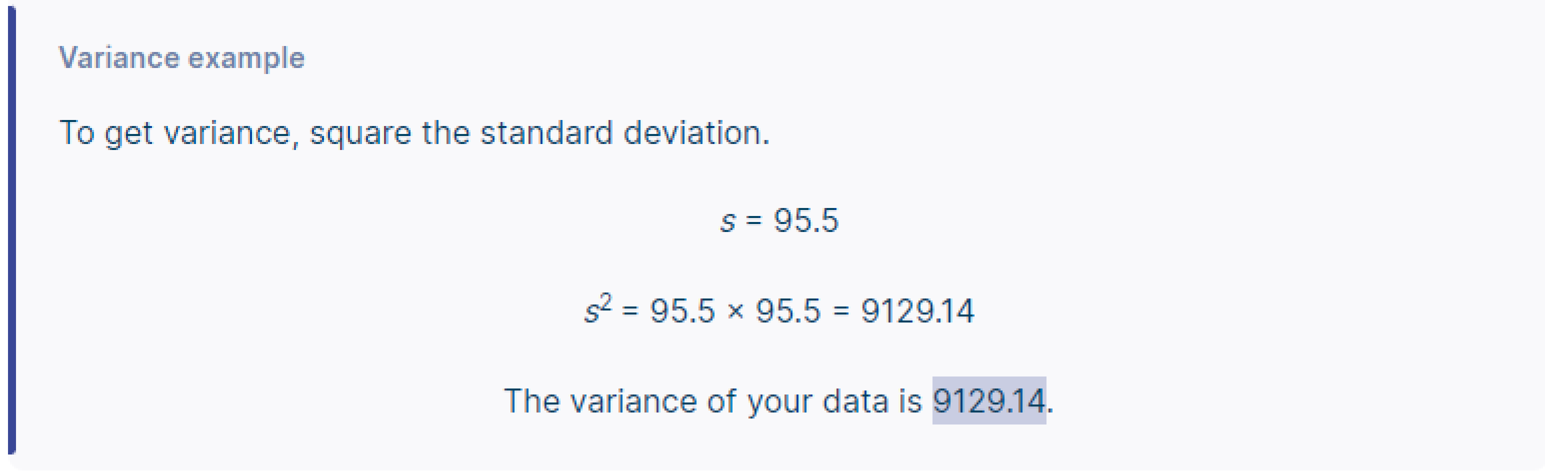
If you have data from the entire population, use the population standard deviation formula:

Formula	Explanation
$\sigma = \sqrt{\frac{\sum (X - \mu)^2}{N}}$	• σ = population standard deviation • $\sum$ = sum of... • X = each value • μ = population mean • N = number of values in the population

Variability Measures (contd.)

- Variance
  - The variance is the average of squared deviations from the mean. A deviation from the mean is how far a score lies from the mean.
  - Variance is the square of the standard deviation. This means that the units of variance are much larger than those of a typical value of a data set.
  - While it’s harder to interpret the variance number intuitively, it’s important to calculate variance for comparing different data sets in statistical tests like ANOVAs.
  - Variance reflects the degree of spread in the data set. The more spread the data, the larger the variance is in relation to the mean.

Variability Measures (contd.)



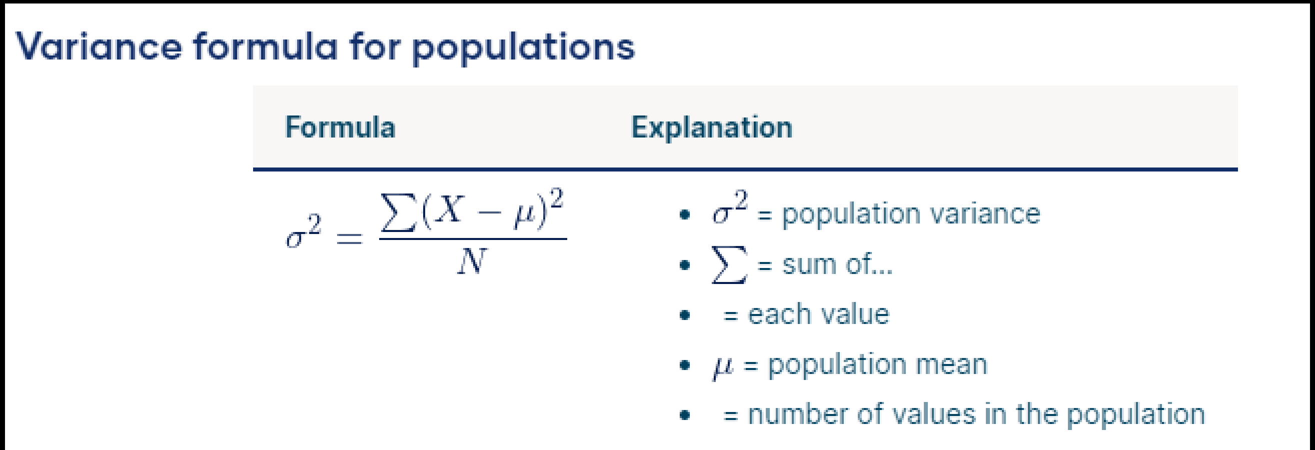
Variance example

To get variance, square the standard deviation.

$$s = 95.5$$

$$s^2 = 95.5 \times 95.5 = 9129.14$$

The variance of your data is 9129.14.



Variance formula for populations

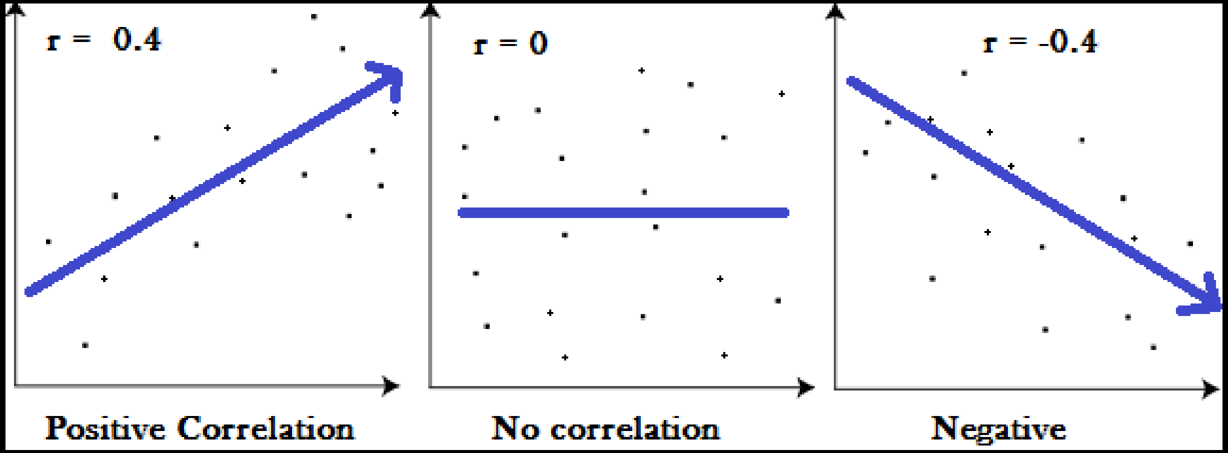
Formula	Explanation
$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$	• σ^2 = population variance • $\sum$ = sum of... • = each value • μ = population mean • = number of values in the population

Correlation Analysis

- Correlation is used to test relationships between quantitative variables or categorical variables.
- In other words, it’s a measure of how things are related. The study of how variables are correlated is called correlation analysis.
- Correlations are useful because if you can find out what relationship variables have, you can make predictions about future behavior.
- Knowing what the future holds is very important in the social sciences like government and healthcare.

Correlation Analysis

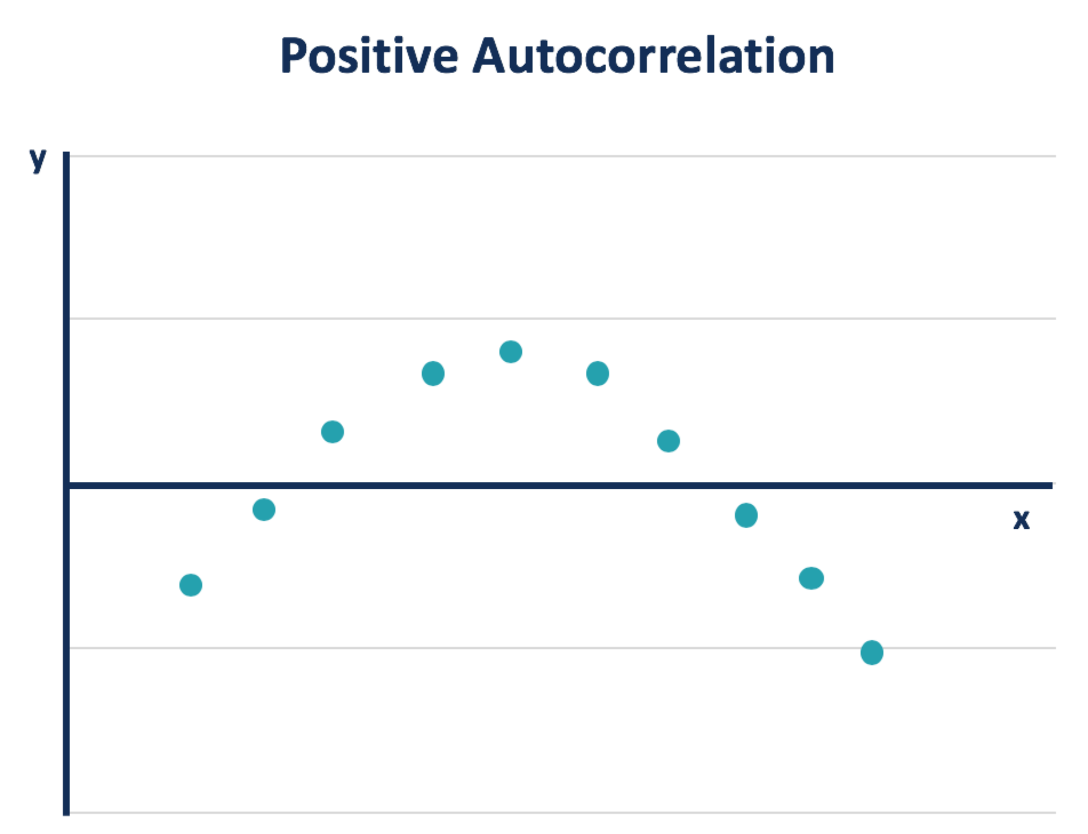
- A correlation coefficient is a way to put a value to the relationship. Correlation coefficients have a value of between -1 and 1.
- A “0” means there is no relationship between the variables at all, while -1 or 1 means that there is a perfect negative or positive correlation (*negative or positive correlation here refers to the type of graph the relationship will produce*).

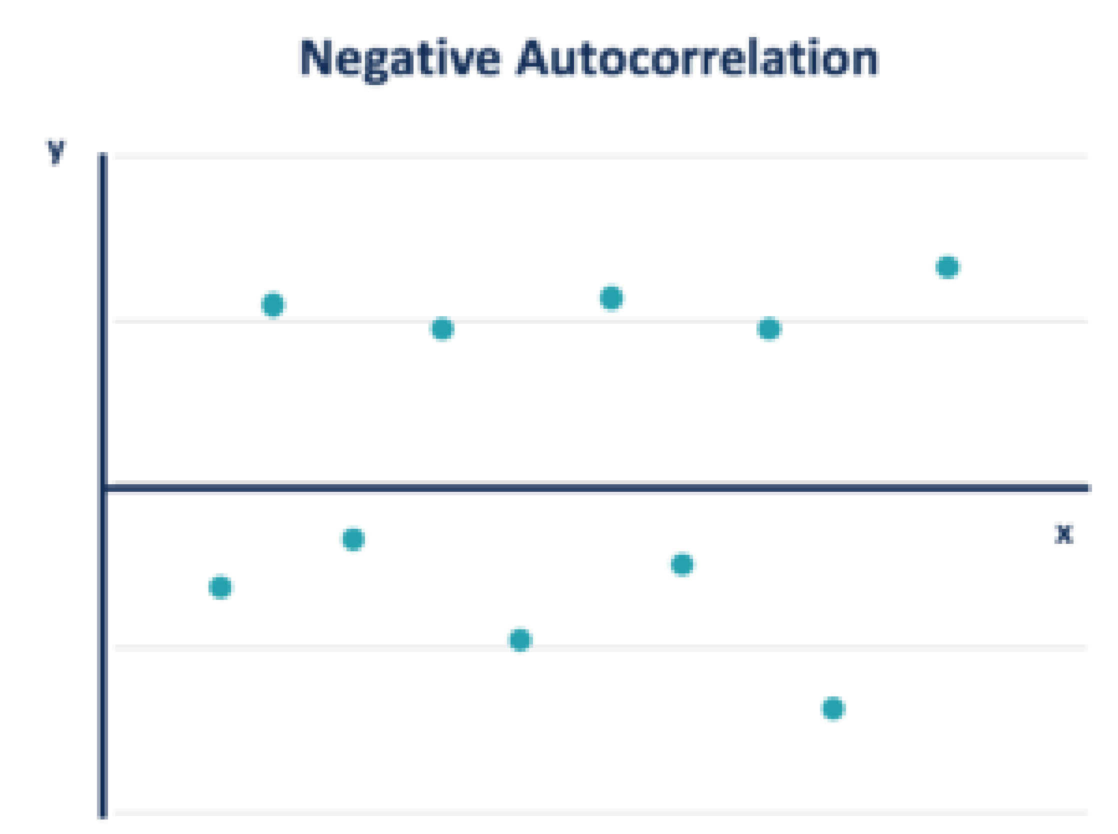


Autocorrelation

- Autocorrelation is a mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals.
- It's conceptually similar to the correlation between two different time series, but autocorrelation uses the same time series twice: once in its original form and once lagged one or more time periods.
- For example,
    - if it's rainy today, the data suggests that it's more likely to rain tomorrow than if it's clear today.
    - When it comes to investing, a stock might have a strong positive autocorrelation of returns, suggesting that if it's "up" today, it's more likely to be up tomorrow, too.
- Naturally, autocorrelation can be a useful tool for data scientist to utilize; particularly for technical analysts.

Autocorrelation (contd.)





Autocorrelation (contd.)

- Autocorrelation can be applied to different numbers of time gaps, which is known as lag.
- A lag 1 autocorrelation measures the correlation between the observations that are a one-time gap apart.
- For example, to learn the correlation between the temperatures of one day and the corresponding day in the next month, a lag 30 autocorrelation should be used (assuming 30 days in that month).

Autocorrelation (contd.)

- The Durbin-Watson statistic is commonly used to test for autocorrelation.
- It can be applied to a data set by statistical software. The outcome of the Durbin-Watson test ranges from 0 to 4.
- An outcome closely around 2 means a very low level of autocorrelation.
- An outcome closer to 0 suggests a stronger positive autocorrelation, and an outcome closer to 4 suggests a stronger negative autocorrelation.

Autocorrelation (contd.)

- It is necessary to test for autocorrelation when analyzing a set of historical data.
- For example, in the equity market, the stock prices on one day can be highly correlated to the prices on another day.
- However, it provides little information for statistical data analysis and does not tell the actual performance of the stock.
- Therefore, it is necessary to test for the autocorrelation of the historical prices to identify to what extent the price change is merely a pattern or caused by other factors.
- In finance, an ordinary way to eliminate the impact of autocorrelation is to use percentage changes in asset prices instead of historical prices themselves.

End of the Chapter

Next: Data Munging (Chapter 2)