read.me()

ideas or information

mind map

a visual representation

of hierarchical and interconnected

to read this

you should be able to

zoom in

to read the detailed information

and zoom out to

understand the context

and main concepts

Ethics and Privacy

Addressing ethical considerations and data privacy issues

Storytelling

Effective communication of findings and insights to non-technical stakeholders

Domain Knowledge

Understanding the specific industry or field to apply data science effectively

Big Data

Tools and frameworks for processing and analyzing large datasets

Machine Learning

Algorithms and techniques for building predictive models and making data-driven decisions

Visualization

Using charts, graphs, and visual representations to convey insights from data

Processing and cleaning

Techniques to handle missing data, outliers, and prepare data for analysis

Data Collection

The process of gathering raw data from various sources, including databases, sensors, or web scraping

DATA SCIENCE

Marcos Lima

Hands-on studies compilation

Statistics

Statistics plays a central role in the realm of data science, serving as the backbone for extracting meaningful insights from data. Data scientists extensively employ statistical techniques to explore, understand, and interpret complex datasets. Descriptive statistics allow us to summarize and visualize data, providing a snapshot of its main characteristics. Inferential statistics, on the other hand, empowers data scientists to make informed decisions, identify patterns, and draw conclusions based on sample data, extending these findings to the broader population. Moreover, hypothesis testing and probability theory underpin the scientific rigor of data science, allowing us to make statistically sound statements and predictions. Statistics is the compass that guides data scientists through the labyrinth of data, aiding in the discovery of hidden patterns and valuable knowledge for a wide array of applications, from predictive modeling to business intelligence.

Ethics and
Privacy

Random Sampling
Stratification
Split

DATA COLLECTION

Archive
Data Lake
Storage

Values
Structured
Unstructured

from

Sources
IOT Devices
Web Scraping
API Integration

with

Quality
Cleansing
Validation

The process of data collection is the main building of data science. This chapter is the beginning of an exploration of diverse data sources dive into the intricacies of structured data, demonstrating how it is meticulously collected from databases and well-organized repositories. Unstructured data, represented by text, images, and videos, poses unique challenges. Semi-structured data, which exhibits a degree of organization but may not adhere to a strict schema.

Moving beyond data sources, there is a vast landscape of data acquisition techniques. The world of web scraping uses HTTP protocol and navigation system over the scripts to harvest data from websites. The integration of Application Programming Interfaces (APIs) can be used as a means to access a wealth of data from various online services and platforms.

Data quality is a crucial facet. All process bellow this point will reflect any failure or miscalculation. Data must have consistency for your results thrive. Techniques for data validation ensure that the data collected remains trustworthy.

In our data-driven age, data privacy and ethics play pivotal roles. Every chapter will bring ideas and techniques in concern to individual privacy.

Furthermore, the journey goes through the world of data sampling, a important branch of statistics and its history. Methods of random sampling, which involves selecting a representative subset of data, stratified sampling, a technique that divides data into subgroups and samples from each were discovered and applied into different scope of data.

Collect data and quality

The most used tools for data manipulation include programming languages such as Python, R and Matlab. These languages are considered 'Swiss Army knives' and share a common thread of versatility and functionality. The code examples will be primarily provided in Python or R. It's important to note that the intention here is to provide a practical perspective, concrete illustrations of the theoretical concepts, navigating to scenarios and showcase how these versatile languages can be applied to solve data-related problems.

Data frame

A data frame is a fundamental data structure used in statistics, data analysis and programming. It can be thought of as a two-dimensional table where data is organized in rows and columns. Each column typically represents a variable or a field, while each row corresponds to an observation or a data point. Data frames are versatile and can store a variety of data types, including numeric, character and factor data. They offer an efficient way to manipulate, analyze, and visualize data, making them a crucial component in data science and statistical analysis. Data frames allow for seamless integration of data from various sources, making them a popular choice for organizing and exploring datasets in data-centric tasks.

Installing R

To detailed instructions on how to install R visit the project website at https://www.r-project.org/

```
> # Example of a data frame
> df <- data.frame(x=c(1,2,3),
                   y=c(7,8,9),
                   z=c("A","B","C"))

> print(df)
  x y z
1 1 7 A
2 2 8 B
3 3 9 C
```

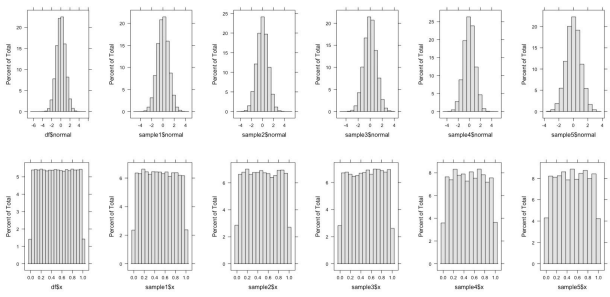
Data sources and their quality play pivotal roles. It's essential to consider the sources from which data is collected ensuring that data is collected from reliable, relevant, and diverse sources enhances the richness and representativeness of the data frame. Moreover, addressing data quality, which includes handling missing data, outliers, and inconsistencies, is crucial for maintaining the integrity of the data frame. Quality assurance processes, such as data cleansing and validation, are fundamental to producing accurate and reliable data frames. This subject will be described further with more details. In this page you can find simple examples how to handle a data frame.

```
# library rvest contains several functions
# to scrape data from internet
scrape_WC02_df <- function(){
  library(rvest)
  # Load World Cup data from wikipedia website
  url <- "https://en.wikipedia.org/wiki/2002_FIFA_World_Cup"
  html_code <- read_html(url)
  # You can find xpath using browser developer tools
  # identifying the table to scrape
  nodes <- html_nodes(html_code,
    xpath='//*[@id="mw-content-text"]/div[1]/table[26]')
  table <- html_table(nodes)
  return(data.frame(table))
}
> df <- scrape_WC02_df()
> print(paste("df has",dim(df)[2],"columns,",dim(df)[1],"rows"))
[1] "df has 12 columns, 32 rows"
> # Selecting columns
> print(df[1:5,c("Team","Pos")])
         Team Pos
1      Brazil   1
2     Germany   2
3      Turkey   3
4 South Korea   4
5       Spain   5
> # Search queries
> print(df[df$Team=="Brazil",c("Team","Pos","Pld","W","D","L")])
    Team Pos Pld W D L
1 Brazil   1   7 7 0 0
> print(df[df$L==0,c("Team","Pos","Pld","W","D","L","GF","GA")])
                  Team Pos Pld W D L GF GA
1               Brazil   1   7 7 0 0 18  4
5                Spain   5   5 3 2 0 10  5
12 Republic of Ireland  12   4 1 3 0  6  3
> # Make some calculations
> print(paste("Sum of goals in the tournament:",sum(df$GF)))
[1] "Sum of goals in the tournament: 161"
> print(paste("Average of goals:",sum(df$GF)/sum(df$Pld)))
[1] "Average of goals: 1.2578125"
> # Adding new column to the data frame
> df$goals_ratio <- df$GF/df$Pld
> # Top 10 scoring ratio teams
> print(head(
+   df[order(df$goals_ratio,decreasing=TRUE),
+      c("Team","Pos","GF","GA","goals_ratio")],10))
                  Team Pos GF GA goals_ratio
1               Brazil   1 18  4    2.571429
2              Germany   2 14  3    2.000000
5                Spain   5 10  5    2.000000
21            Portugal  21  6  4    2.000000
17        South Africa  17  5  5    1.666667
19          Costa Rica  19  5  6    1.666667
12 Republic of Ireland  12  6  3    1.500000
14             Belgium  14  6  7    1.500000
16            Paraguay  16  6  7    1.500000
3               Turkey   3 10  6    1.428571
```

Sampling

Statistical sampling involves the process of selecting a subset of data from a larger population to make inferences about that population. This field has a rich history dating back to the early days of statistical analysis, with pioneers like Francis Galton, who used sampling techniques to study heredity, and Ronald A. Fisher, who made substantial contributions to the theory of sampling. In modern data science, sampling is integral to efficiently handling large datasets, conducting surveys, and performing hypothesis testing. Understanding various sampling methods, such as random and stratified sampling, allows data scientists to extract valuable insights and draw conclusions from data with precision and reliability. Sampling is a powerful tool to data science to its analytical core.

```
> # Generate a data frame with random values
> # rnorm creates normal distribution and runif uniform
> df = data.frame(normal=rnorm(78**3),x=runif(78**3))
> summary(df)
     normal                x
 Min.   :-4.545741   Min.   :0.0000036
 1st Qu.:-0.673120   1st Qu.:0.2516106
 Median :-0.000155   Median :0.5012793
 Mean   : 0.001163   Mean   :0.5007995
 3rd Qu.: 0.674084   3rd Qu.:0.7507580
 Max.   : 4.615366   Max.   :0.9999989
> require(lattice)
> histogram(df$normal,col="gray")
> # Create samples with different sizes 10-1% of original
> sample1 <- df[sample(nrow(df),round(nrow(df)*.10)),]
> sample2 <- df[sample(nrow(df),round(nrow(df)*.08)),]
> sample3 <- df[sample(nrow(df),round(nrow(df)*.05)),]
> sample4 <- df[sample(nrow(df),round(nrow(df)*.02)),]
> sample5 <- df[sample(nrow(df),round(nrow(df)*.01)),]
> # Plot will have same distributions as population
```



```
> # Now, if you use the same seed...
> set.seed(9)
> df = data.frame(normal=rnorm(78**3),x=runif(78**3))
> # You will find the same "random numbers" as bellow
> summary(df)
     normal                x
 Min.   :-4.618437   Min.   :0.0000004
 1st Qu.:-0.676649   1st Qu.:0.2496231
 Median :-0.000465   Median :0.4991344
 Mean   :-0.001224   Mean   :0.4994693
 3rd Qu.: 0.674798   3rd Qu.:0.7493647
 Max.   : 4.832043   Max.   :0.9999978
```

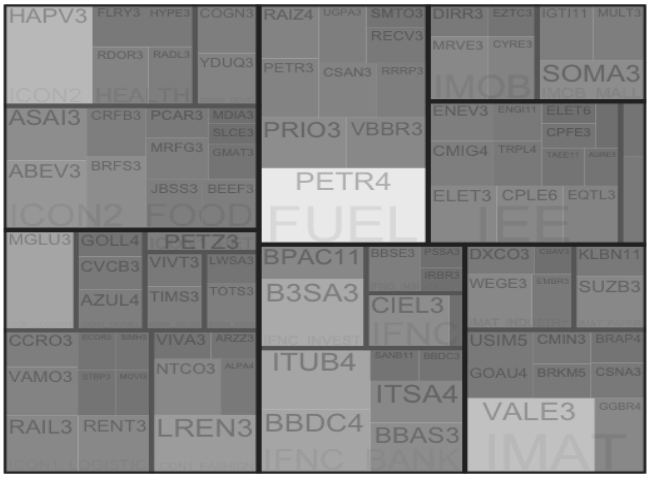
Randomization

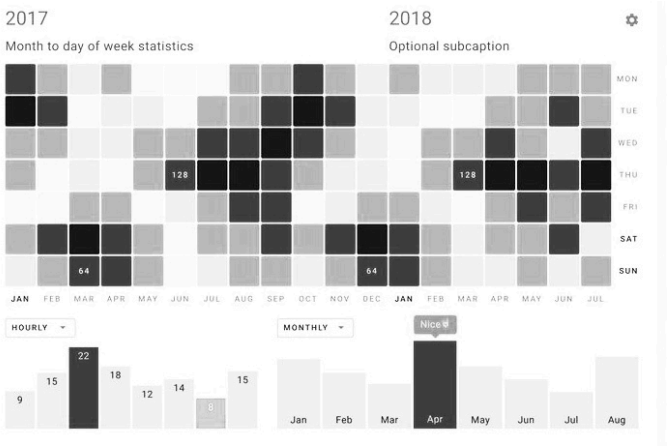
This is a process widely used in statistics and data science to ensure the selection of a representative subset of data from a larger dataset. It avoid bias and increase the generalizability of results. Randomness reduces the likelihood of unintentional patterns or biases enabling robust statistical analyses and dependable insights. Although random subsets of data have the same distribution shape as the population.

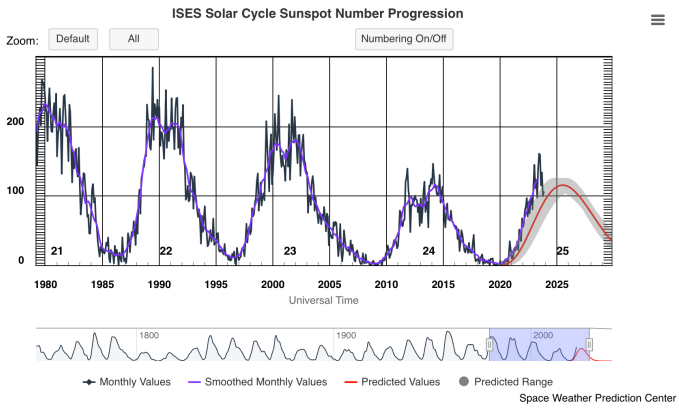
Randomization in computer is achieved using pseudorandom number generators (PRNGs). These algorithms generate sequences of numbers that appear to be random but are actually determined by an initial value called a seed. The seed is typically set based on some unpredictable factor, such as system time or user input. While PRNGs can't produce truly random numbers like radioactive decay or quantum processes, they are sufficient for most practical purposes. The key is to use well-designed PRNGs and frequently update the seed to minimize predictability.

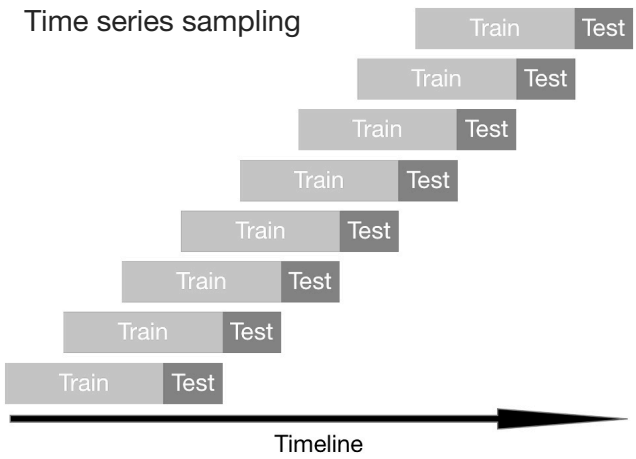
Seed

In random selection algorithms, the seed plays a crucial role, as it not only ensures randomness but also enables replicability. By setting the same seed, you can choose the same sample repeatedly and achieve same results. This capability has several valuable applications. First, it's a useful technique for troubleshooting as it allows you to retrace your steps and investigate issues in your analysis. Second, it promotes collaboration and idea sharing among colleagues. When you share your code and analysis with colleagues providing them with the same seed ensures they can reproduce your results precisely.









Stratification

Stratified sampling involves dividing a diverse dataset into homogeneous subgroups or strata based on specific characteristics or attributes. Each stratum represents a subset of the population, and then random sampling is applied within each stratum. The key idea is to ensure that each subgroup is well-represented in the sample, which can lead to more accurate and reliable inferences. Stratified sampling is particularly useful when dealing with a population that exhibits significant variability across certain characteristics, as it allows you to obtain a balanced and representative sample.

Time series sampling

It is a specialized form of sampling commonly used in analytics when dealing with time-ordered data. Time series data consists of observations or measurements collected and recorded in chronological order. It aims to select specific data points or segments within this chronological sequence for analysis. This type of sampling is instrumental in tasks such as forecasting future trends, identifying seasonality patterns, and understanding the temporal behavior of data. Some common techniques include simple random sampling from time intervals, systematic sampling at fixed intervals, or sampling specific time points of interest. Time series sampling allows data scientists to uncover patterns and insights within temporal data, making it a valuable component in various applications, from financial analysis to climate modeling.

Moreover, we will dive into the evaluation process of predictions. It's essential to have two distinct samples collected from different timeframes to ensure that our assessments out-of-sample are not influenced by any seasonal biases. This helps in providing a more comprehensive and unbiased evaluation of predictive models and their performance across various temporal scenarios.

Storage

A virtual drive can be used for storing data and scripts in the same context. Many cloud storage providers offer virtual drives that are accessible via the internet. Examples like Google Drive, Dropbox, OneDrive, and Box. You can store data files, scripts, and other assets in these virtual drives. They provide the advantage of easy access and collaboration from various devices and locations.

In an enterprise or on-premises data lake setup, you can use network-mapped drives to access centralized storage resources. These drives can be used to store and share data and scripts across a local network.

Using shared drives can simplify data and script management, but it's essential to ensure proper security, access control, and versioning practices to maintain data integrity and manage scripts effectively.

Many data lakes use object storage solutions like Amazon S3, Azure Blob Storage, or Google Cloud Storage. Object storage provides a scalable and cost-effective way to store both data and scripts. You can organize your data into containers or buckets and include script files alongside the data.

Data lakes built on Hadoop-based platforms often use Hadoop Distributed File System (HDFS) or Hadoop-compatible distributed file systems. These systems allow you to store data and scripts within the same environment.

For managing data-related scripts version control, systems like Git are commonly used. Git allows you to track changes to your scripts, collaborate with team members, and maintain a history of script versions. While data itself is not stored in Git, you can include references to data files or data storage locations within your scripts.

The choice of storage depends on the specific architecture and tools used in your data lake ecosystem. It's common to use a combination of options to ensure that data and scripts are well-organized, versioned, and accessible for data processing and analysis.

Data Lake

The concept of a Data Lake, as it is understood today, evolved over time as organizations sought more efficient and flexible ways to store and manage vast amounts of data. It can be traced back to the early 2000s when the volume of data generated by businesses and individuals began to grow exponentially.

The term "Data Lake" gained prominence with the advent of big data technologies which enabled organizations to store and process data at a scale that was previously unattainable. Hadoop, for example, allowed the storage of large datasets across distributed clusters, functioning as a sort of data reservoir. Over time, this concept expanded to include other storage solutions such as cloud-based object storage services provided by companies like Amazon, Microsoft, and Google.

Data Lakes offer a way to store structured and unstructured data in a single repository, providing data scientists, analysts, and engineers the ability to access and analyze data without the need for extensive preprocessing. They have become a central component of modern data architectures, enabling organizations to harness the power of big data and analytics.

Archive

In the dynamic field of data science, maintaining a structured and accessible data archive is indispensable. Data archiving involves the systematic storage and preservation of valuable datasets, ensuring their availability for future analysis.

Data is an asset, and like any asset, it requires safeguarding. Different datasets demand distinct archiving strategies: local, cloud-based or offline. Adequate documentation is a must-piece of data archiving creating comprehensive metadata, including descriptions, data dictionaries, version histories, simple readme.md file.

Time Machine Archive is a concept. Much like its fictional namesake, it allows you to revisit and restore data as it existed at specific points in time. It can be built in a special file format known as .rda (R Data).

Storing archive in these files involves creating snapshots of your datasets and functions at different time points and saving them as .rda files. These files are essentially binary R objects that contain the state of your data at specific moments in time. To store a Time Machine Archive, you can follow a version control approach, such as Git, which helps you manage changes to your datasets over time. By creating and committing .rda files with clear labels or descriptions for each snapshot, you maintain a historical record of your data's evolution. When needed, you can retrieve, explore, or restore any version of your dataset from these files, ensuring data reproducibility and facilitating error correction in data science projects.

An essential aspect of data archiving is understanding the data's lifecycle. The lifecycle of data doesn't end with archiving; it also includes responsible data purging and deletion practices, especially in compliance with data privacy regulations.

In an era of ever-expanding data volumes, the knowledge of data archiving, Time Machine Archives, and their implementation through .rda files is useful for data scientists to maintain data integrity, support research reproducibility, and ensure data's long-term accessibility.

price_data.log
Web Server Logs

QDB
Databases

Analytics
Social Media

API
Third Party Data

Files

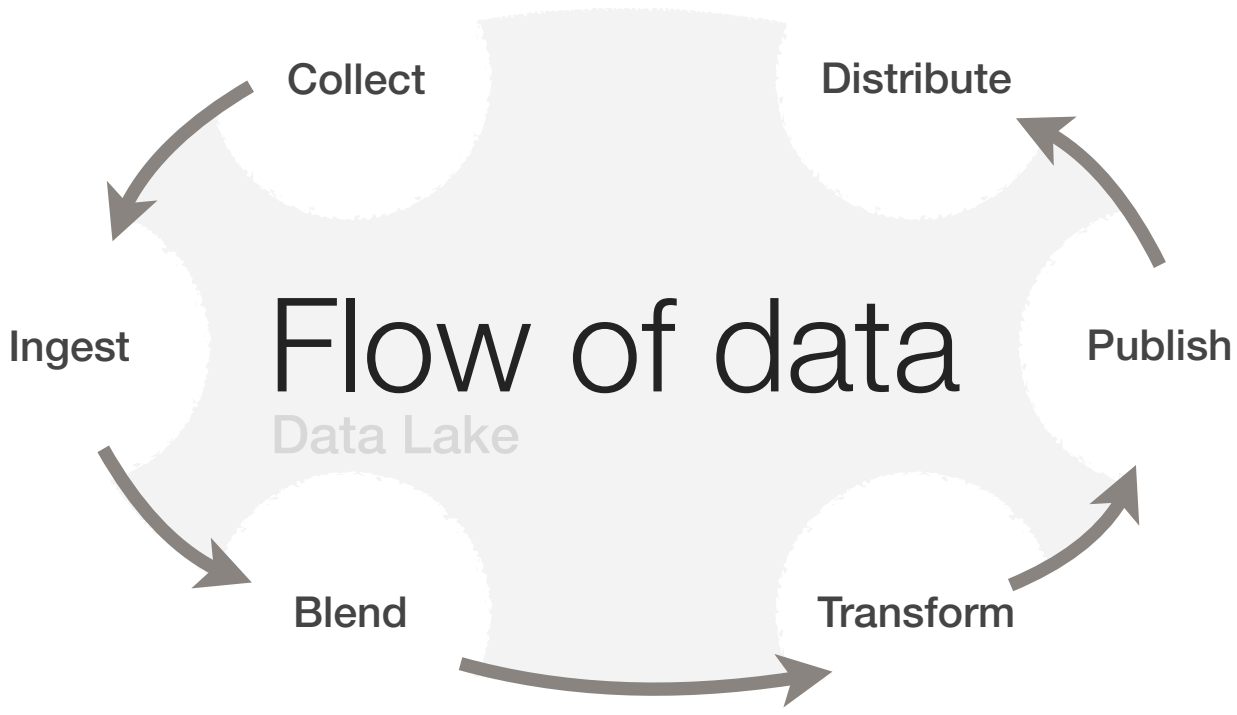
Dashboards

Databases

Big Data Warehouse

Business Intelligence Tools

Storage



PROCESSING

AND

CLEANING

Data preprocessing encompasses a range of techniques aimed at improving data quality. In the collect phase, raw data may contain missing values, outliers, or inconsistencies that need to be addressed. Handling missing data might involve imputation methods like filling in missing values with means or medians. Outliers, data points that deviate significantly from the norm, can be identified and treated appropriately. Inconsistent data, such as different formats or units, may need standardization. The goal of data preprocessing is to create a clean and reliable dataset as a foundation for analysis.

Once the data is cleaned and reliable, data transformation comes into play. This phase is about shaping the data to make it suitable for analysis. Categorical variables, for instance, can be transformed into numerical format through encoding techniques like one-hot encoding. Numeric data might be normalized or standardized to bring all features to a similar scale. Mathematical transformations, like taking logarithms, can be applied to create more meaningful relationships within the data. The goal is to ensure that the data meets the assumptions and requirements of the analytical methods to be used.

Data integration deals with combining data from various sources to create a unified dataset. During the blend and transform phases, you might have collected data from multiple databases, files, or streams. Data integration addresses challenges such as merging datasets with different schemas, resolving inconsistencies, and ensuring overall data consistency. This stage aims to create a comprehensive dataset that can provide a holistic view of the subject matter. It's crucial to ensure that the data from different sources align in terms of formats, units, and other relevant attributes.

The flow of data through these processes is like a journey where data is collected, refined, transformed, and integrated to create a valuable resource for analysis. Each phase is an integral part of this journey, ensuring that the data is well-prepared to derive meaningful insights and support data-driven decision-making. In data science, the quality and effectiveness of the data processing phases significantly impact the quality of the results.

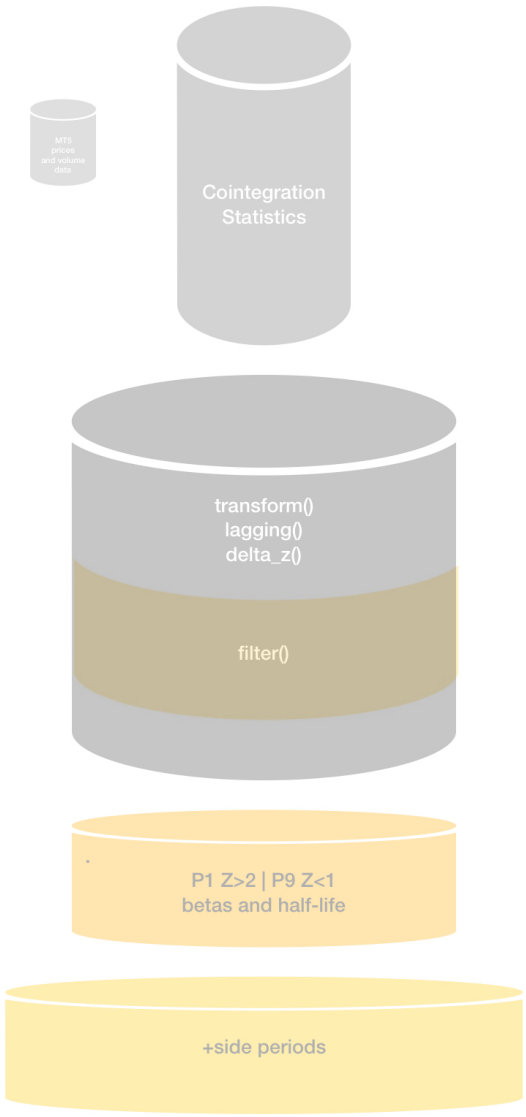
Data Reduction

Involves techniques to decrease the dimensionality of data, which is particularly useful when dealing with datasets containing many features or variables. High dimensionality can lead to computational challenges and may not necessarily contribute to the quality of analysis. Feature selection methods identify a subset of the most relevant features, reducing computational complexity and often improving model interpretability. Feature extraction techniques transform the data into a lower-dimensional space, preserving essential information. Principal Component Analysis (PCA) is a common method for feature extraction. In the context of machine learning, reducing the dimensionality of data can enhance the efficiency of modeling processes. It reduces the risk of overfitting, simplifies model interpretation, and accelerates training and prediction.

Data Imputation

Vital for handling missing values in a dataset. Missing data is a common issue that can hinder analysis and modeling. Data imputation methods replace or fill in missing values, making the dataset more complete. Simple approaches include mean imputation, where missing values are replaced by the mean of the available data for that variable, or regression imputation, where missing values are estimated based on regression models. Advanced machine learning-based imputation techniques leverage relationships within the data to predict and fill in missing values. Data imputation is critical in preserving data integrity and ensuring that the analysis includes all available information.

Efficient data processing is crucial in modern data science, where large datasets and complex models are common. These techniques are part of the toolkit for addressing challenges that can arise in real-world data scenarios.



Efficiency in machine learning pertains to the ability to achieve high-quality results using the least amount of computational resources and time. In the context of data reduction, by selecting or extracting the most informative features, machine learning models can focus on the aspects of data that truly influence the target variable. This leads to more efficient model training and predictions. Data imputation also enhances efficiency by preventing the loss of valuable data due to missing values, ensuring that models are built on as much relevant information as possible. In both cases, efficiency is a significant consideration in data processing as it impacts the model's performance, computational resources required, and ultimately, the ability to derive meaningful insights from data.

Data Cleaning Tools

Data cleaning tools play a pivotal role in the data processing pipeline. These tools are designed to streamline and automate the process of identifying and rectifying data inconsistencies, errors, and missing values. OpenRefine, for instance, provides a user-friendly environment for exploring data, making transformations, and detecting anomalies. Trifacta offers a more advanced data cleaning platform that employs machine learning to assist in the data wrangling process. Moreover, both Python and R offer specialized libraries for data cleaning, such as pandas in Python and tidyr in R, empowering data professionals to implement custom data cleaning routines and maintain data quality.

Data Quality Assessment

Evaluating data quality is fundamental in data processing. This branch focuses on the criteria and methods used to assess the quality of datasets. It involves metrics like accuracy, which measures how closely data aligns with the true values it represents, consistency, which gauges the uniformity of data across sources, and completeness, which assesses the extent of missing or incomplete data. Data quality assessment ensures that the data used for analysis and modeling is trustworthy and reliable.

Text Data Cleaning

Cleaning and preprocessing text data is vital, especially in natural language processing tasks. This section concentrates on techniques for transforming unstructured text into a structured format suitable for analysis. Tasks like text normalization, which converts text to lowercase and removes special characters, tokenization, which breaks text into individual words or tokens, and stemming, which reduces words to their root form, are key processes in text data cleaning.

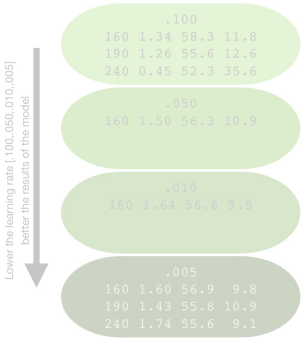
Time Series Data Cleaning

Time series data, characterized by sequential data points over time, often presents unique challenges. This part of the chapter focuses on cleaning and handling time series data, including addressing irregular timestamps, managing missing values, and detecting anomalies. Effective time series data cleaning is crucial for maintaining data accuracy, especially in applications like financial analysis, weather forecasting, and IoT data processing.

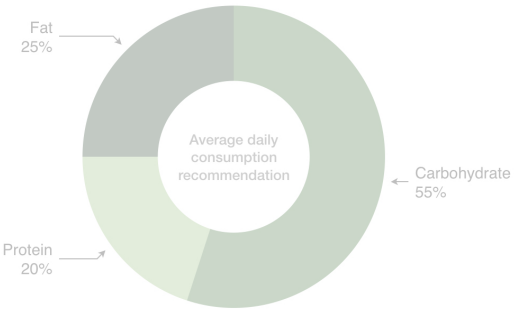
Best Practices and Tips

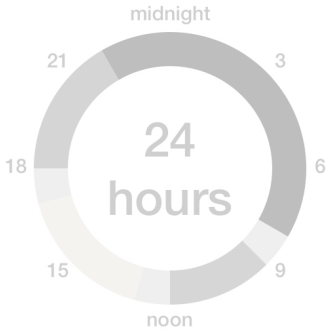
Finally, this section provides an essential guide to best practices in data processing and cleaning. It emphasizes the significance of maintaining data integrity and quality throughout the data journey. These best practices include documenting data cleaning procedures, employing version control, and creating robust data pipelines to ensure that data remains reliable and valuable for analysis.

Data processing and cleaning are foundational steps in any data science project, as the quality and accuracy of your data directly impact the validity of your analyses and the reliability of your insights. These branches equip data professionals with tools and practices to effectively address data inconsistencies and ensure that the data remains an asset rather than a hindrance in the analysis process.



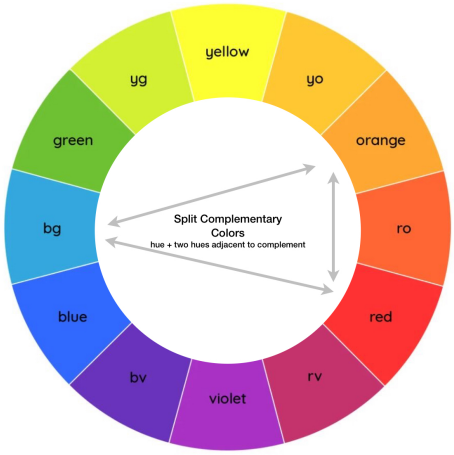
VISUALIZATION





Data visualization serve as a bridge between complex datasets and meaningful insights. It is such prominence in the field of data science by transforming raw data into visual representations enhancing the interpretability of information, making it accessible to both technical and non-technical audiences. Through compelling visual narratives, data scientists can convey patterns, trends, and outliers, facilitating better-informed decision-making.

Understanding the principles that underpin effective data visualization is crucial: simplicity, clarity, and accuracy. By adhering to these principles, visualizations become powerful tools for communication. Select appropriate chart types for different types of data and explores best practices for color usage, ensuring that visualizations are not only aesthetically pleasing but also convey information accurately.



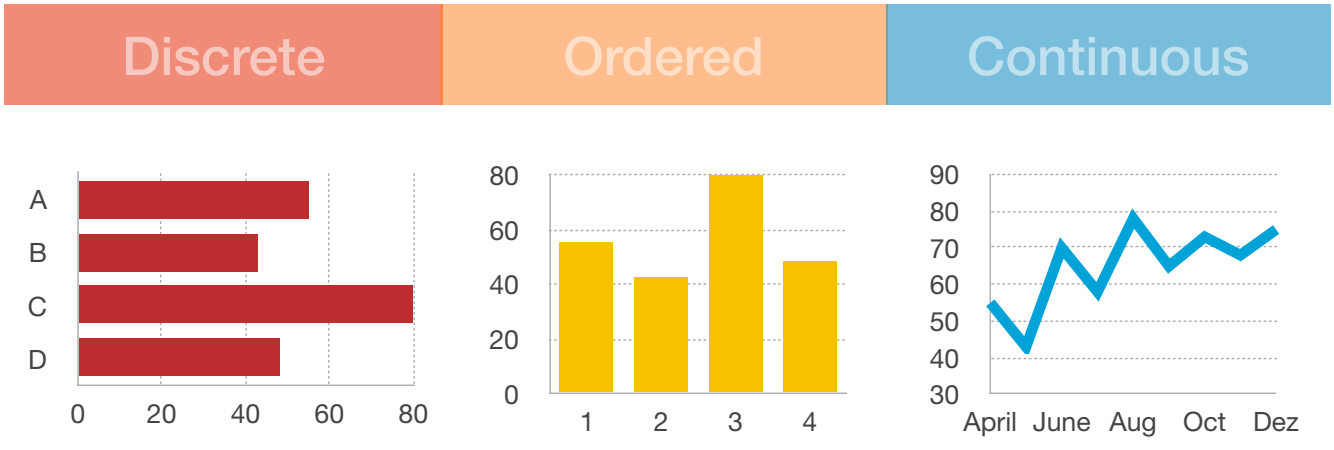
Types of Data Visualizations

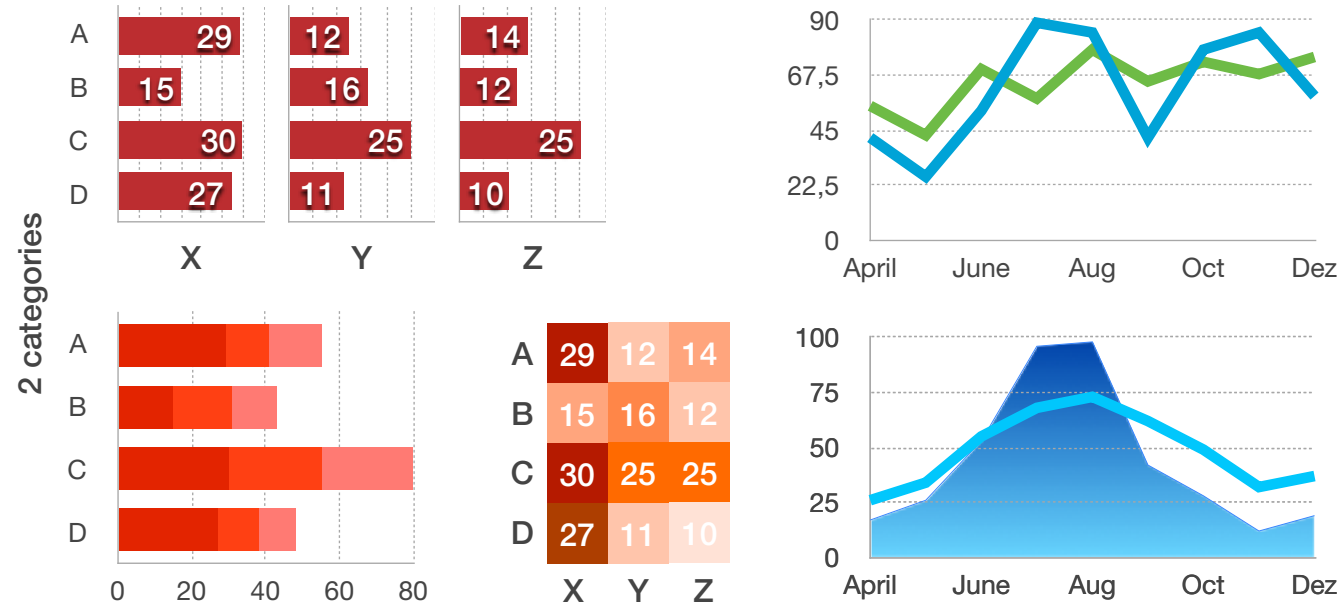
There are various chart types, ranging from foundational ones like bar charts and line charts to more advanced visualizations such as heatmaps and treemaps. Each visualization type is discussed in terms of its strengths, weaknesses, and ideal use cases. Readers gain insights into the nuanced decisions involved in choosing the right visualization method based on the nature of the data they are working with.

Tools

Readers are familiarized with platforms like Tableau and Power BI, as well as programming libraries such as Matplotlib (Python) and ggplot2 (R). The chapter includes practical demonstrations, illustrating how to create basic visualizations using these tools. It emphasizes the flexibility and customization offered by programming languages, enabling data scientists to tailor visualizations to their specific needs.

This comprehensive exploration equips readers with a robust foundation in data visualization, emphasizing both theoretical principles and practical applications. Through a combination of insights into the importance of visualization, guiding principles, diverse visualization types, and hands-on exposure to tools, empowers individuals to harness the full potential of data visualization in their data science endeavors.





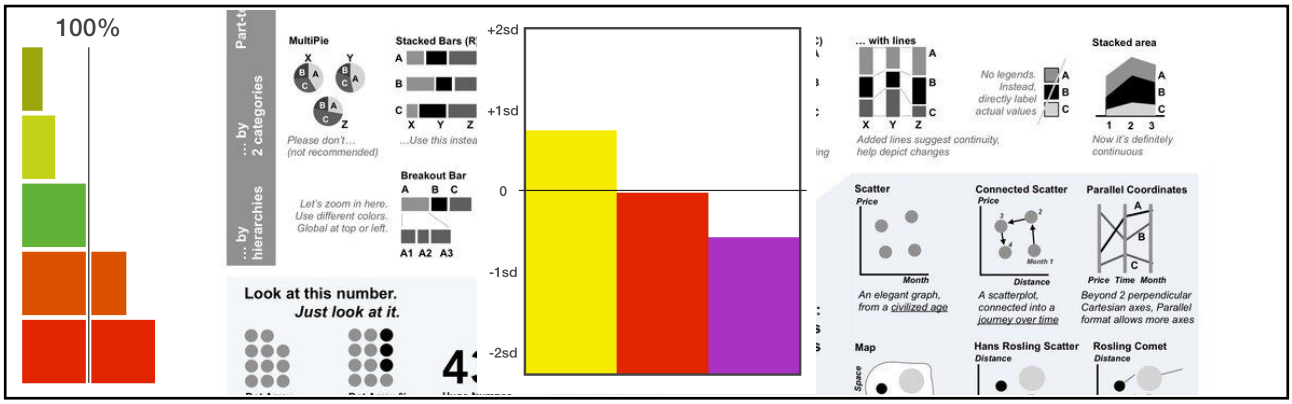
Yyyyyyyyyyyyyyyyyyyyy

There are various chart types, ranging from foundational ones like bar charts and line charts to more advanced visualizations such as heatmaps and treemaps. Each visualization type is discussed in terms of its strengths, weaknesses, and ideal use cases. Readers gain insights into the nuanced decisions involved in choosing the right visualization method based on the nature of the data they are working with.

Xxxxxx. Xxxxxxxxxxxx

Readers are familiarized with platforms like Tableau and Power BI, as well as programming libraries such as Matplotlib (Python) and ggplot2 (R). The chapter includes practical demonstrations, illustrating how to create basic visualizations using these tools. It emphasizes the flexibility and customization offered by programming languages, enabling data scientists to tailor visualizations to their specific needs.

This comprehensive exploration equips readers with a robust foundation in data visualization, emphasizing both theoretical principles and practical applications. Through a combination of insights into the importance of visualization, guiding principles, diverse visualization types, and hands-on exposure to tools, empowers individuals to harness the full potential of data visualization in their data science endeavors.



Interactive Data Visualizations

Understanding the advantages of interactivity in conveying complex datasets, readers are introduced to tools and libraries supporting interactive features. Practical examples of interactive dashboards illustrate their applications, showcasing how users can engage with and explore data in real time.

Scatter Plot, relationship between two continuous variables

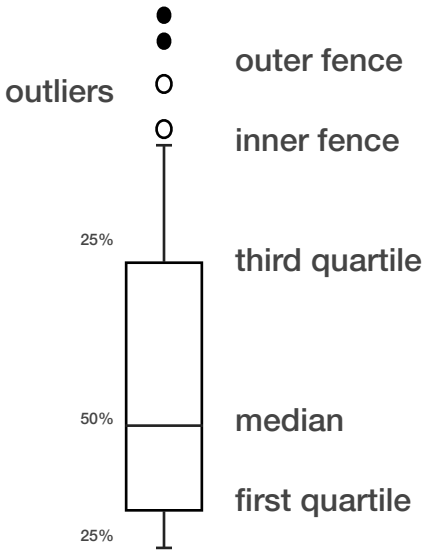
Storytelling with Data

Connecting the art of storytelling to data visualization, this section emphasizes the narrative aspect of visualizations. Techniques for creating a compelling story through data are discussed, allowing readers to move beyond standalone visualizations to coherent and impactful data-driven narratives. Case studies provide real-world examples of successful storytelling with data in various domains.

Best Practices

Guidelines for creating effective visualizations are central and covering aspects like labeling, titling, and annotation, readers gain insights into practices that enhance the clarity and interpretability of visualizations. The section also highlights common pitfalls to avoid and stresses the importance of incorporating feedback and iterative improvements in the visualization process. Emphasizing the principles of fairness, transparency, and accuracy, readers explore how ethical practices should guide the creation and interpretation of visual representations. Discussions include potential biases in visualizations and strategies to mitigate them, ensuring responsible and unbiased use of data.

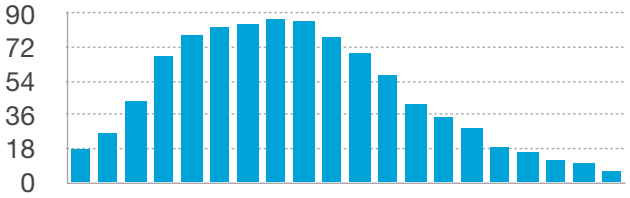
Exploratory Data Analysis



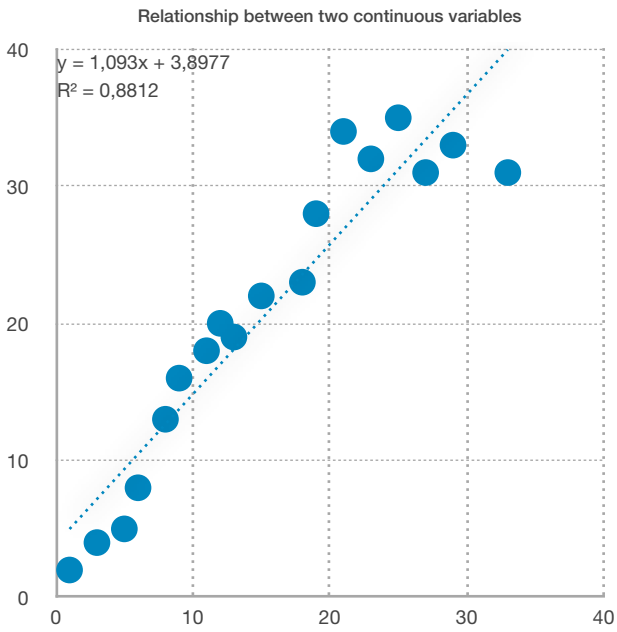
"The Future of Data Analysis" is a notable paper written by John W. Tukey, a renowned statistician. The paper was published in 1962 in the journal "The Annals of Mathematical Statistics." In this influential work, Tukey discusses the challenges and opportunities in the field of data analysis, emphasizing the need for exploratory data analysis (EDA) and the importance of graphical methods in understanding complex datasets.

Tukey's ideas in this paper laid the foundation for modern data analysis practices, and his advocacy for EDA has significantly influenced how analysts and data scientists approach the initial stages of working with data. The paper is considered a classic in the field of statistics and data science, and it continues to be referenced in discussions about the evolving landscape of data analysis.

EDA involves a suite of techniques that play a crucial role in understanding the inherent patterns and characteristics within a dataset. Summary Statistics serve as the initial lens through which data is scrutinized, providing key insights into central tendency (mean, median, mode) and variability (range, standard deviation). These measures offer a concise overview, enabling quick assessments of the data's basic properties.



Data Visualization emerges as a powerful ally in EDA, transforming raw numbers into visual narratives. Techniques like histograms, box plots, scatter plots, and heatmaps not only unveil the distributional aspects but also reveal intricate relationships and potential outliers. Visualization becomes a bridge between raw data and meaningful insights, enhancing the interpretability of complex datasets.



Univariate Analysis directs attention to individual variables, dissecting them in isolation. This technique allows for a deep dive into the distributional nuances of each variable, uncovering trends, patterns, and potential anomalies.

Consider a dataset of students' exam scores in a mathematics class. Univariate analysis of the variable "exam scores" involves examining its distribution. A histogram can be created to illustrate the frequency distribution of scores, providing insights into the central tendency and variability. Bivariate analysis could involve exploring the relationship between "exam scores" and another variable, such as "study hours." A scatter plot can be used to visualize how study hours correlate with exam scores. This analysis helps ascertain if there's a positive, negative, or neutral relationship between the two variables. Extending the analysis further, suppose we introduce a third variable, "sleep hours." Multivariate analysis can then explore how "exam scores" relate to both "study hours" and "sleep hours" simultaneously. Techniques like multiple regression analysis could be applied to understand the combined influence of study and sleep hours on exam performance.

Outlier detection is another indispensable aspect of EDA. Outliers, or data points significantly deviating from the norm, can distort analyses and impact the reliability of models. Detecting and appropriately handling outliers is essential to ensure that insights derived from the data accurately reflect underlying trends and patterns. Various statistical methods and visualization techniques are employed in this process.

Handling missing data is a common challenge in any data analysis endeavor. The presence of missing values can impede the completeness of analyses, leading to biased or incomplete conclusions. EDA involves a meticulous assessment of missing data, and analysts employ strategies such as imputation or exclusion to address these gaps effectively. Ensuring a robust approach to handling missing data enhances the overall reliability of the analysis.

Pattern recognition is a powerful tool in EDA that involves the identification of trends, clusters, or distinctive patterns within the dataset. This process goes beyond basic summary statistics and visualizations, aiming to unearth more subtle and complex structures within the data. Pattern recognition techniques contribute to a deeper comprehension of the underlying dynamics, providing valuable insights for subsequent stages of analysis and decision-making.

For instance, in customer segmentation for an e-commerce platform, pattern recognition can unveil distinct clusters of users with similar purchasing behavior. This knowledge can inform targeted marketing strategies tailored to each segment, optimizing customer engagement and satisfaction.

Another example lies in fraud detection for financial transactions. Through pattern recognition algorithms, anomalies in transaction patterns can be identified, signaling potential fraudulent activities. Unusual withdrawal locations, atypical spending patterns, or irregular transaction frequencies can be indicative of fraudulent behavior, prompting timely intervention.

Moreover, in medical diagnostics, pattern recognition plays a pivotal role. For instance, in analyzing medical imaging data, algorithms can recognize patterns associated with specific diseases or conditions. This aids clinicians in early and accurate diagnosis, enhancing patient outcomes.

Clustering
Pattern search
Dimension Reduction

Classification
Regression

Supervised

Unsupervised

Classical

Data mining is a subset of machine learning that focuses on discovering patterns and knowledge from large datasets. It involves techniques such as clustering, classification, regression, and association rule mining. Replacing "Data Mining" under "Machine Learning" can emphasize its role as part of the broader field of machine learning within the context of data science.

Data Mining

Evaluation and Validation

MACHINE LEARNING

In traditional statistical approaches, the analysis often begins with the assumption of a specific data model that reflects the underlying structure of the phenomenon being studied. Parameters for this model are then estimated from the available data. For instance, in linear regression, a model assumes a linear relationship between the response variable and predictors, and the analysis aims to estimate coefficients that define this relationship. This approach works well when the true data-generating process aligns with the assumed model.

Reinforcement Learning

Ensemble Methods

In contrast, machine learning takes a different route. Instead of starting with a predefined data model, machine learning algorithms are designed to learn the relationship between the response variable and its predictors directly from the data. This is particularly advantageous when the true data-generating process is complex or not well understood. Machine learning algorithms, such as decision trees, support vector machines, or neural networks, assume that the underlying patterns are intricate and might involve non-linear relationships or interactions between variables.

Deep Learning

Perceptrons
Autoencoders
Generative Adversarial Networks
Recurrent Neural Networks
Convolutional NN

For example, in a machine learning context, consider a classification task where the goal is to predict whether an email is spam or not based on various features. Instead of explicitly assuming a specific probability distribution for the data, machine learning algorithms like a random forest or a support vector machine learn patterns in the data that distinguish between spam and non-spam emails. They adapt to the inherent complexity of the data, capturing intricate relationships that might not be easily expressed by a predefined statistical model.

In essence, while traditional statistics starts with a model and estimates parameters, machine learning embraces the complexity of real-world data by leveraging algorithms that can adapt and learn patterns directly from the data without relying on a predetermined model. This flexibility makes machine learning particularly powerful in scenarios where the underlying data structure is complex and not explicitly known.

Supervised learning is a paradigm where the algorithm is trained on a labeled dataset, which means the input data is paired with corresponding output labels. The goal is for the model to learn the mapping between inputs and outputs so that it can make predictions or classifications on new, unseen data. The algorithm is provided with a training dataset that includes both features (inputs) and labels (outputs), and the model adjusts its parameters through optimization to minimize the difference between its predictions and the actual labels. Common algorithms in supervised learning include linear regression and classification algorithms like support vector machines, decision trees, and neural networks. Applications of supervised learning include image classification, spam detection, and predicting house prices.

Unsupervised learning involves tasks where the algorithm is given input data without explicit instructions on what to do with it. The algorithm explores the structure or patterns within the data without the guidance of labeled outputs. The primary goal is often to discover inherent relationships, groupings, or representations within the data. Clustering and dimensionality reduction are common tasks in unsupervised learning. Clustering algorithms, such as k-means clustering and hierarchical clustering, group similar data points together. Dimensionality reduction techniques, like principal component analysis (PCA), aim to reduce the number of features while preserving essential information. Unsupervised learning is used in applications such as customer segmentation, anomaly detection, and topic modeling.

Supervised	Unsupervised
algorithm is trained on a labeled dataset	algorithm works with unlabeled data
focused on making predictions or classifications based on input-output mappings	concerned with exploring and understanding the inherent structure or relationships within the data
regression and classification methods	clustering and dimensionality reduction techniques
Number of classes is known	Number of classes is NOT known
widely used in scenarios where predictions or classifications are required	applied when the goal is to explore and uncover patterns in data
uses offline analysis	real-time analysis
more complex computation	less complex computation
Linear Regression Neural Networks Decision Trees Naive Bayes Regression Trees Gradient Boosting	K-means Clustering Hierarchical Clustering Principal Component Analysis Self-Organizing Maps (neural network) Independent Component Analysis t-Distributed Stochastic Neighbor Embedding

The roots of supervised learning can be traced to the mid-20th century. The development of linear regression, a fundamental technique in supervised learning, dates back to the work of Francis Galton in the late 19th century. However, the formalization of the supervised learning paradigm gained momentum in the 20th century. In the 1950s and 1960s, pioneers like Arthur Samuel and Frank Rosenblatt worked on early forms of machine learning and neural networks. The perceptron, a simple neural network model, was introduced by Rosenblatt in 1957.

The 1980s and 1990s saw significant advancements in supervised learning with the development of decision tree models. Leo Breiman and Jerome Friedman, along with other researchers at the University of California, Berkeley, and Stanford University, played a crucial role in the evolution of tree-based models. The introduction of ensemble methods like bagging and boosting further enhanced the predictive capabilities of models. Notably, Breiman's work on random forests in 2001 contributed to the popularity of ensemble methods.

The history of unsupervised learning is intertwined with the exploration of patterns and structures within data. Clustering methods, a key component of unsupervised learning, have roots in early statistical methods. However, the formalization of clustering algorithms gained momentum in the mid-20th century.

In the 1950s, the k-means clustering algorithm was introduced by Stuart Lloyd for signal processing applications. Throughout the following decades, researchers continued to develop and refine clustering algorithms. Hierarchical clustering methods, which organize data in a tree-like structure.

Dimensionality reduction saw notable advancements with the introduction of principal component analysis (PCA) by Karl Pearson in the early 20th century. However, its widespread use in machine learning gained traction later, particularly with the advent of digital computing.

The late 20th century and early 21st century witnessed the integration of supervised and unsupervised learning techniques. This integration, along with advancements in computational power and the emergence of large datasets, led to the rise of modern statistical machine learning. The development of neural networks, particularly deep learning models, marked a significant milestone in achieving complex representations from data, contributing to breakthroughs in image and speech recognition.

In summary, the historical development of classical statistical machine learning involved the evolution of both supervised and unsupervised learning paradigms, driven by the contributions of key researchers and advancements in computational capabilities. The integration of these paradigms, along with the rise of deep learning, has shaped the landscape of contemporary machine learning.

Data Mining

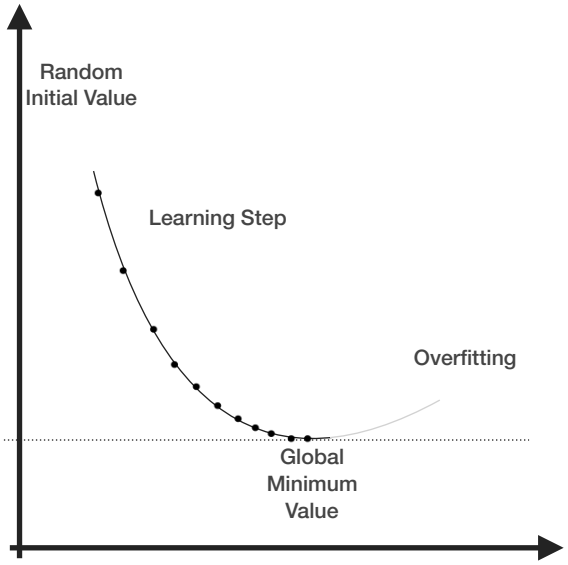
The Law of Large Numbers

The Law of Large Numbers

Statistical approaches analysis to model fitting start by assuming an appropriate data model, and parameters for this model are then estimated from the data. By contrast, machine learning avoids starting with a data model and rather uses an algorithm to learn the relationship between the response and its predictors. ML assumes that the data-generating process is complex and unknown, and tries to learn the response by observing inputs and responses and finding dominant patterns.

Ensemble methods

Ensemble methods in machine learning blend predictions from multiple models to create a more robust and powerful predictive model. By leveraging diverse models, ensemble methods mitigate the weaknesses of individual models, resulting in enhanced overall performance. They are categorized into bagging and boosting methods. While effective in improving generalization and mitigating overfitting, ensemble methods come with computational costs. The choice of the appropriate ensemble method depends on data characteristics and specific problem requirements, considering factors like model interpretability and available computational resources.



Random Forest is a prominent bagging algorithm that builds multiple decision trees during training. Each tree is constructed using a random subset of the training data (bootstrap samples), and features are randomly sampled at each split. The final prediction is an average or voting of the predictions made by individual trees, reducing overfitting and increasing robustness.

Gradient Boosting is a boosting algorithm that builds trees sequentially, with each tree correcting the errors of the previous ones. It focuses on instances that previous trees misclassified, assigning higher weights to them. The final prediction is a weighted sum of the predictions made by each tree. Gradient Boosting is known for its high predictive accuracy but requires careful tuning to prevent overfitting.

AdaBoost is an early boosting algorithm that assigns weights to each instance in the dataset. It starts with a weak learner and assigns higher weights to misclassified instances, making them more influential in subsequent iterations. The final prediction is a weighted sum of weak learners.

Stacking involves training multiple diverse models and combining their predictions using another model called a meta-learner. The base models can be of different types or trained on different subsets of data. The meta-learner takes the predictions of the base models as input and outputs the final prediction. Stacking aims to leverage the complementary strengths of different models.

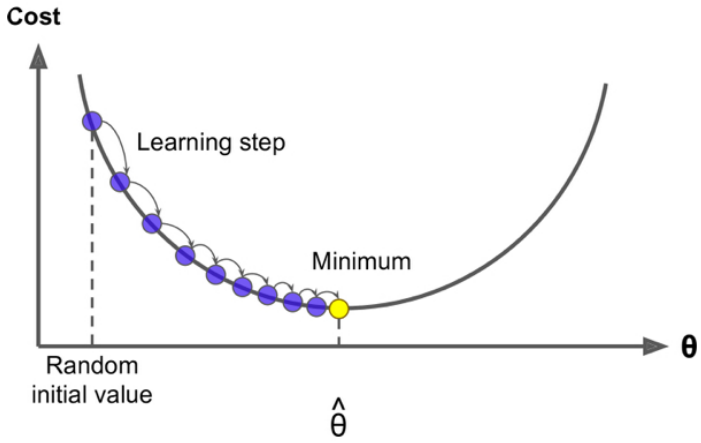
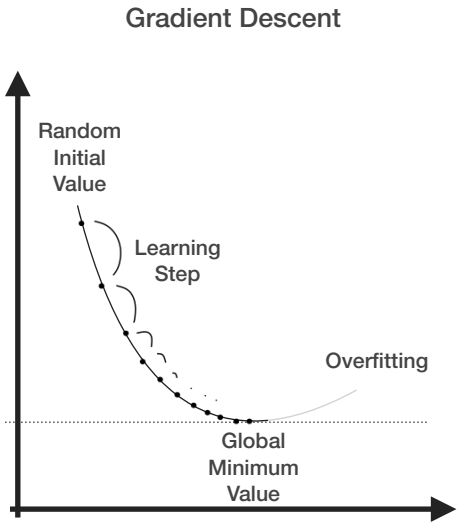


Fig 3. Gradient descent (Geron, 2017).



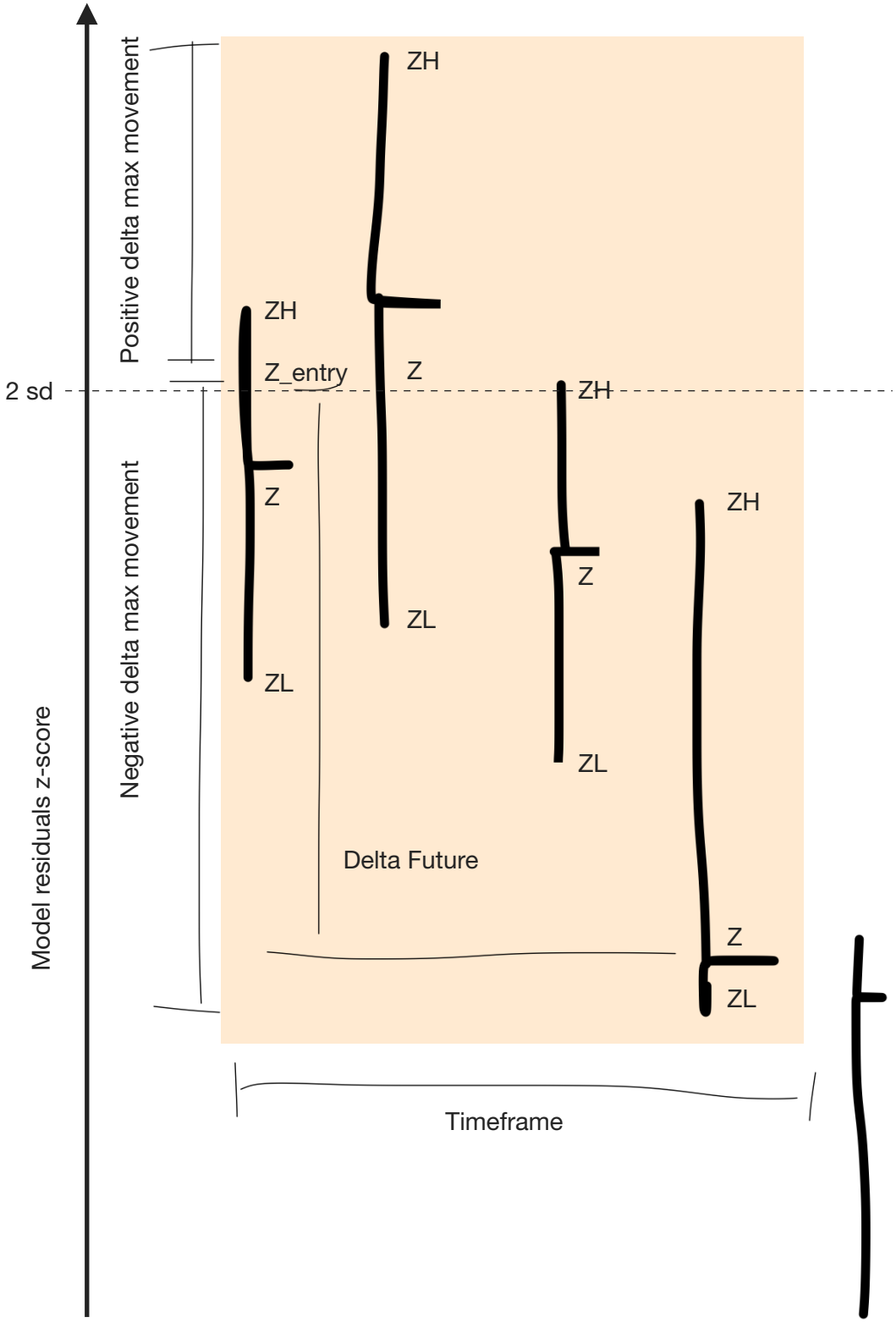
Confusion Matrix

Optimization and fine-tuning

In the journey to enhance the predictive accuracy of a learning model, several key strategies come into play. Firstly, feature engineering and variable selection offer avenues to mold the input features for better capturing underlying patterns. Concurrently, addressing missing data and outliers through effective data cleaning ensures a more robust dataset.

Hyperparameter tuning, ensemble methods, and regularization collectively contribute to the fine-tuning and regularization of the gradient boost model. Techniques such as hyperparameter tuning through methods like grid search or random search empower model optimization, while ensemble methods, combining predictions from diverse models, harness their individual strengths. Regularization techniques further aid in preventing overfitting, promoting a more balanced and effective predictive model.

Domain knowledge plays the final word in shaping the predictive model. Drawing from a deep understanding of the subject matter, domain knowledge facilitates the creation of meaningful features that encapsulate specific nuances in the data. Simultaneously, it guides the removal of irrelevant variables, streamlining the model to focus on factors that contribute significantly to the prediction task. This iterative process, informed by domain expertise, aligns the model with the intricacies of the real-world scenario, enhancing its interpretability and efficacy.


Positive delta max movement
2 sd
Negative delta max movement
Model residuals z-score
ZH
ZH
Z_entry
Z
Z
ZL
ZL
ZH
Z
ZL
ZH
Z
ZL
Delta Future
Timeframe

If you're working with gradient boost machines and aiming to enhance prediction accuracy while optimizing the choice of variables, several techniques can be considered:

1. **Feature Engineering**:

- **Create New Features**: Derive new features from existing ones that might capture more meaningful patterns.

- **Variable Transformation**: Apply transformations like logarithmic or power transformations to make relationships more linear.

2. **Variable Selection**:

- **Recursive Feature Elimination (RFE)**: Iteratively remove the least important variables and retrain the model until the desired number of features is reached.

- **LASSO Regression**: Use L1 regularization to shrink some coefficients to zero, effectively performing variable selection.

- **Random Forest Feature Importance**: If applicable, use the feature importance scores from a random forest model to identify influential variables.

3. **Data Cleaning**:

- **Handle Missing Data**: Implement strategies to address missing values, either by imputation or removing rows/columns with missing data.

- **Outlier Detection and Treatment**: Identify and address outliers that might be impacting the model's performance.

4. **Model Tuning**:

- **Hyperparameter Tuning**: Fine-tune the hyperparameters of your gradient boost model through techniques like grid search or random search.

- **Ensemble Methods**: Explore combining predictions from multiple models, potentially different algorithms, to leverage their strengths.

5. **Cross-Validation**:

- Use techniques like k-fold cross-validation to get a better estimate of the model's performance on unseen data and detect overfitting.

6. **Domain Knowledge**:

- Leverage your understanding of the domain to engineer features that might capture specific nuances in the data.

- Remove irrelevant variables that don't contribute meaningfully to the prediction task.

7. **Regularization**:

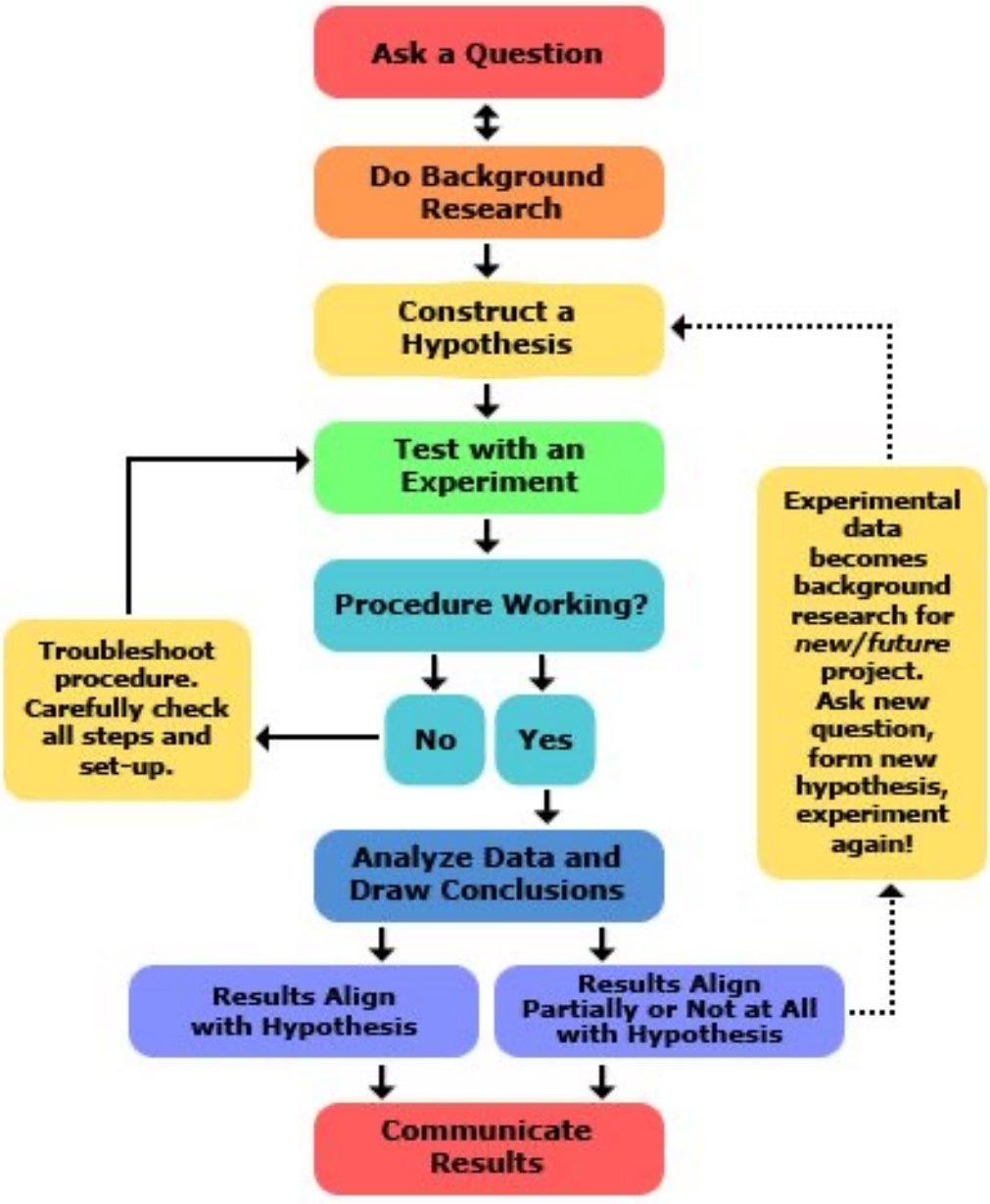
- Apply regularization techniques within the gradient boost model to penalize overly complex models, which might help generalize better to new data.

8. **Data Scaling**:

- Ensure that variables are on similar scales, especially if the gradient boost algorithm is sensitive to scale differences.

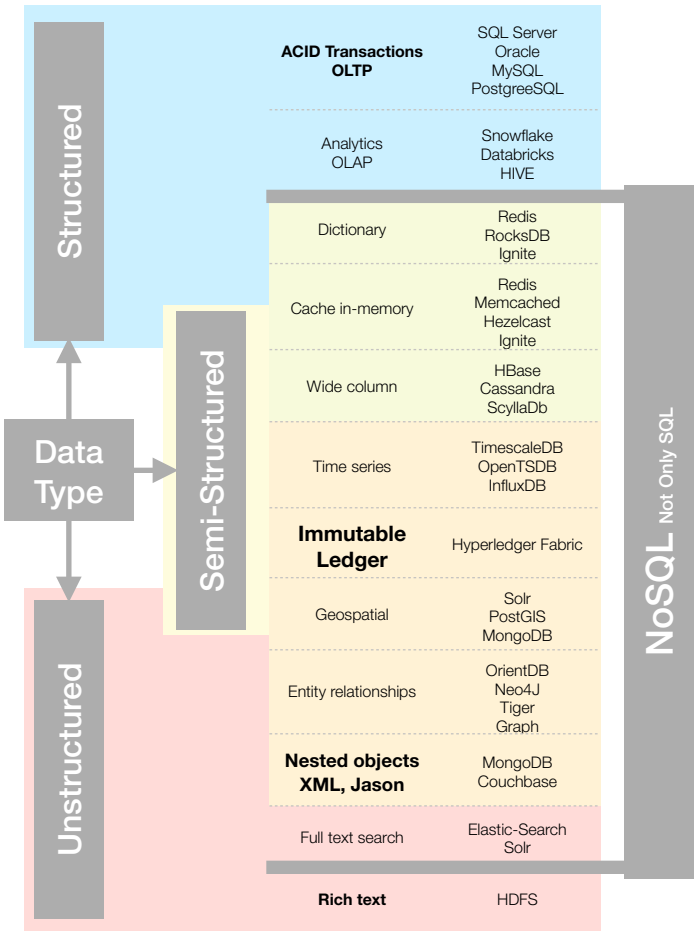
Remember, it's often an iterative process involving experimentation and evaluation to find the combination of techniques that works best for your specific dataset and prediction task. Additionally, considering the interpretability of the model and the business context is crucial.

Scientific method



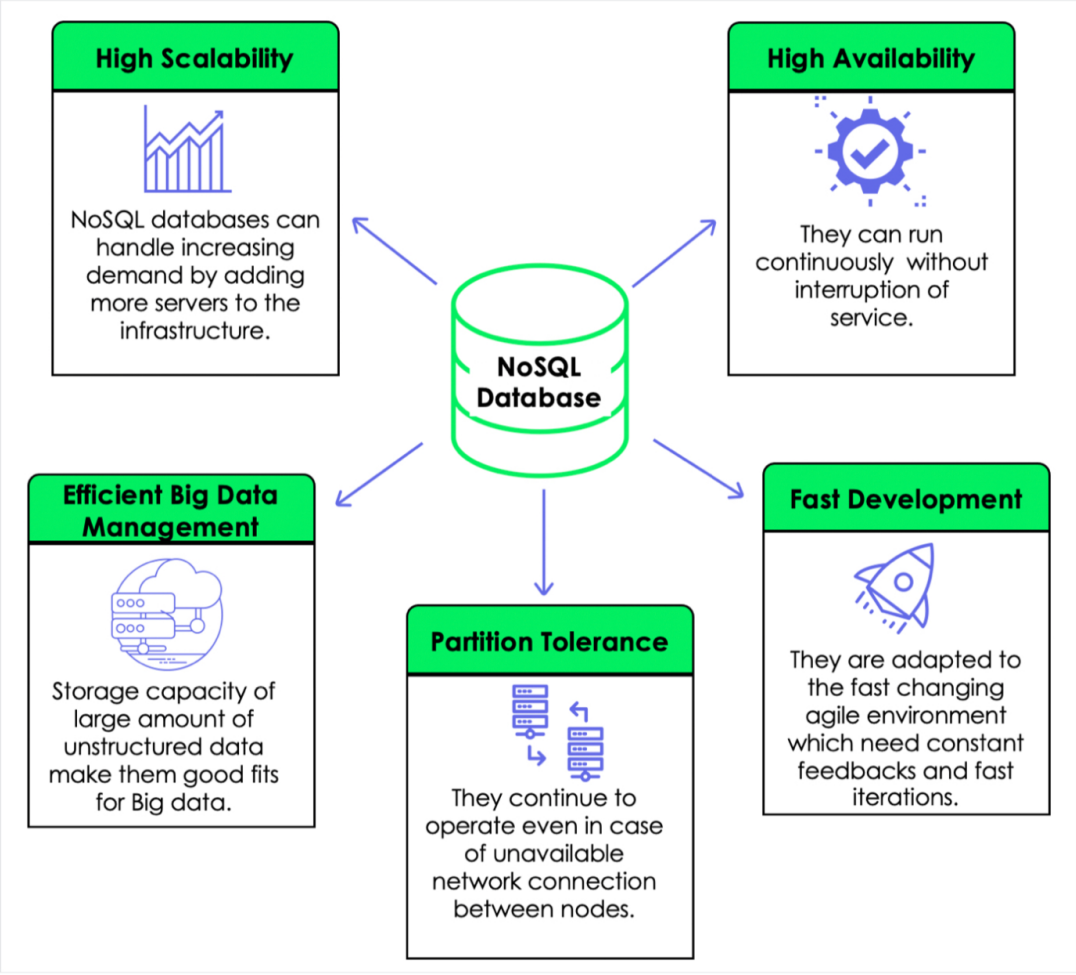
BIG DATA

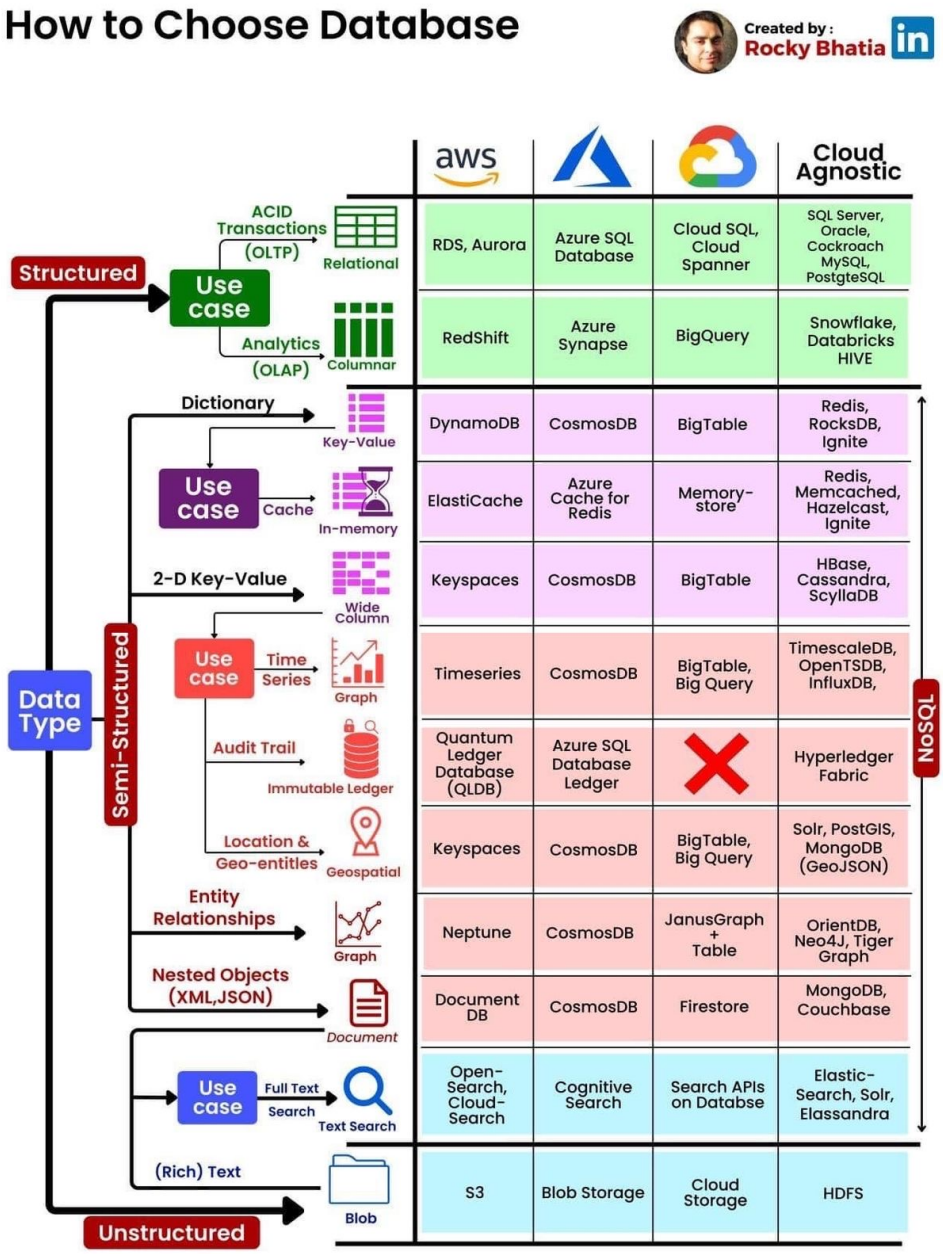
Environment (load RDA)

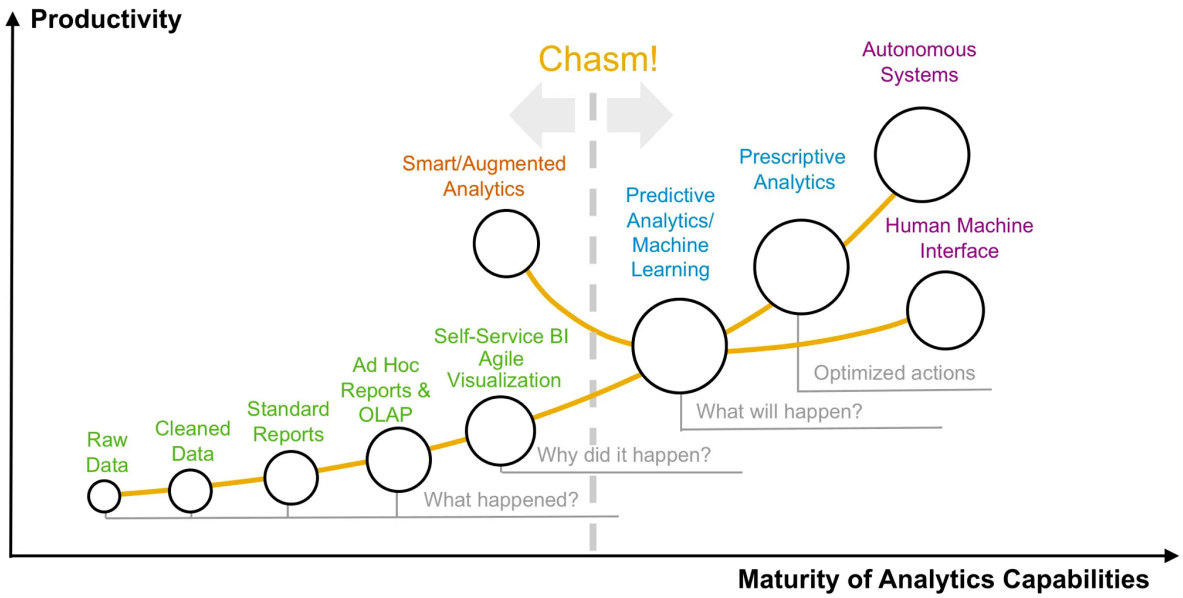

Structured
Data Type
Semi-Structured
Unstructured
NoSQL Not Only SQL
ACID Transactions OLTP
SQL Server Oracle MySQL PostgreeSQL
Analytics OLAP
Snowflake Databricks HIVE
Dictionary
Redis RocksDB Ignite
Cache in-memory
Redis Memcached Hezelcast Ignite
Wide column
HBase Cassandra ScyllaDb
Time series
TimescaleDB OpenTSDB InfluxDB
Immutable Ledger
Hyperledger Fabric
Geospatial
Solr PostGIS MongoDB
Entity relationships
OrientDB Neo4J Tiger Graph
Nested objects XML, Jason
MongoDB Couchbase
Full text search
Elastic-Search Solr
Rich text
HDFS

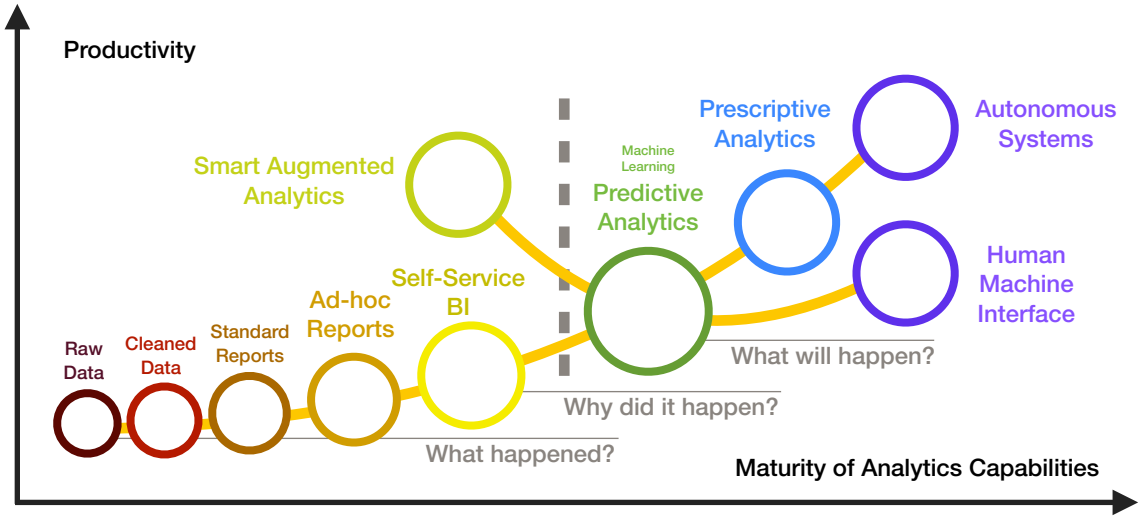
What are NoSQL databases?

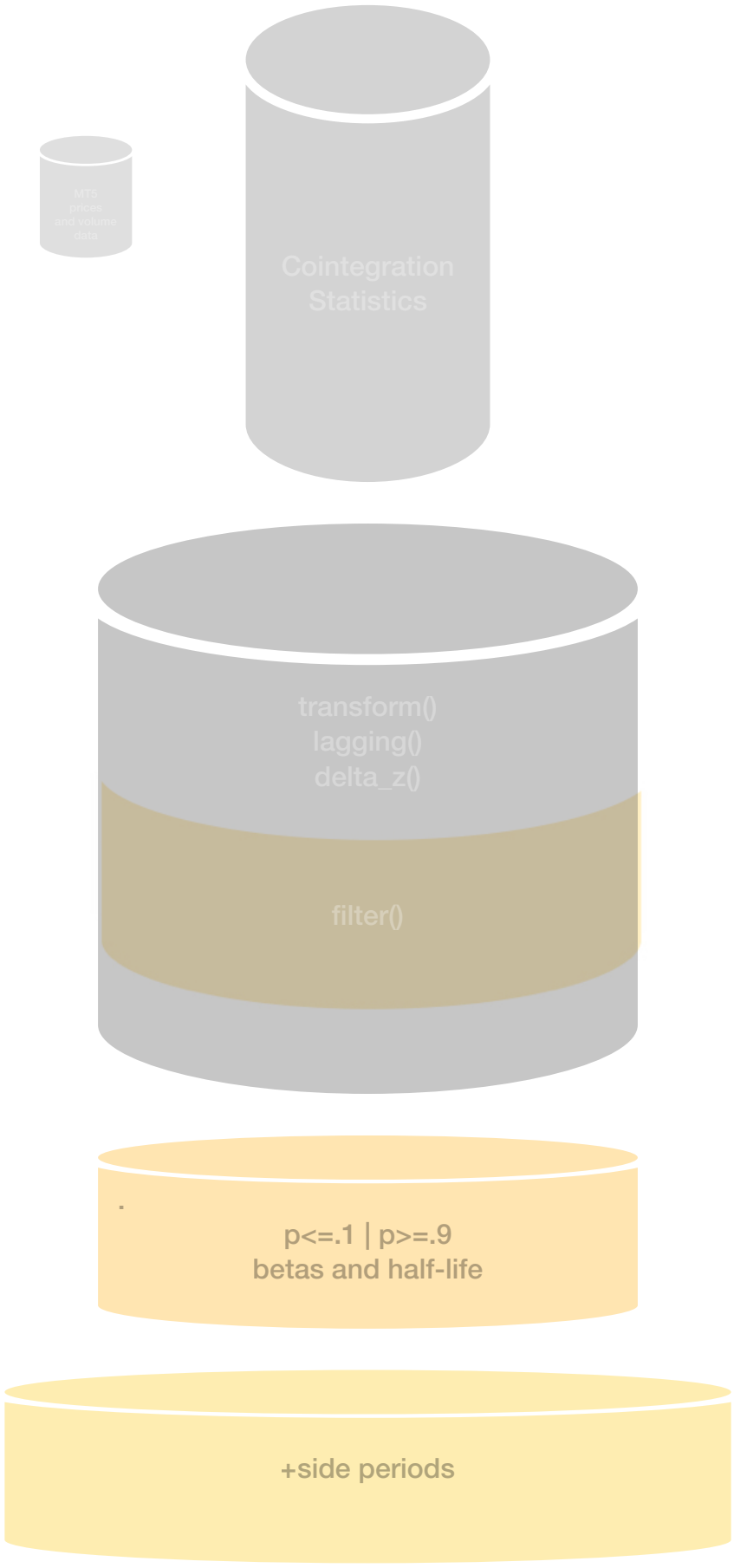
NoSQL stands for ***Not Only SQL***, meaning that NoSQL databases have the specificity of not being relational because they can store data in an unstructured format. The following graphic highlights the main five key features of NoSQL databases.




How to Choose Database
Created by : Rocky Bhatia
Data Type
Structured
Use case
ACID Transactions (OLTP)
Relational
Analytics (OLAP)
Columnar
Semi-Structured
Dictionary
Key-Value
Use case
Cache
In-memory
2-D Key-Value
Wide Column
Use case
Time Series
Graph
Audit Trail
Immutable Ledger
Location & Geo-entitles
Geospatial
Entity Relationships
Graph
Nested Objects (XML,JSON)
Document
Use case
Full Text Search
Text Search
Unstructured
(Rich) Text
Blob
aws
Cloud Agnostic
RDS, Aurora
Azure SQL Database
Cloud SQL, Cloud Spanner
SQL Server, Oracle, Cockroach MySQL, PostgteSQL
RedShift
Azure Synapse
BigQuery
Snowflake, Databricks HIVE
DynamoDB
CosmosDB
BigTable
Redis, RocksDB, Ignite
ElastiCache
Azure Cache for Redis
Memory-store
Redis, Memcached, Hazelcast, Ignite
Keyspaces
CosmosDB
BigTable
HBase, Cassandra, ScyllaDB
Timeseries
CosmosDB
BigTable, Big Query
TimescaleDB, OpenTSDB, InfluxDB,
Quantum Ledger Database (QLDB)
Azure SQL Database Ledger
Hyperledger Fabric
Keyspaces
CosmosDB
BigTable, Big Query
Solr, PostGIS, MongoDB (GeoJSON)
Neptune
CosmosDB
JanusGraph + Table
OrientDB, Neo4J, Tiger Graph
Document DB
CosmosDB
Firestore
MongoDB, Couchbase
Open-Search, Cloud-Search
Cognitive Search
Search APIs on Databse
Elastic-Search, Solr, Elassandra
S3
Blob Storage
Cloud Storage
HDFS
NoSQL


Productivity
Chasm!
Autonomous Systems
Smart/Augmented Analytics
Prescriptive Analytics
Predictive Analytics/ Machine Learning
Human Machine Interface
Self-Service BI Agile Visualization
Ad Hoc Reports & OLAP
Standard Reports
Cleaned Data
Raw Data
Optimized actions
What will happen?
Why did it happen?
What happened?
Maturity of Analytics Capabilities


Productivity
Prescriptive Analytics
Autonomous Systems
Machine Learning
Smart Augmented Analytics
Predictive Analytics
Human Machine Interface
Self-Service BI
Ad-hoc Reports
Standard Reports
Cleaned Data
Raw Data
What will happen?
Why did it happen?
What happened?
Maturity of Analytics Capabilities


MT5
prices
and volume
data
Cointegration
Statistics
transform()
lagging()
delta_z()
filter()
.
p<=.1 | p>=.9
betas and half-life
+side periods

Streamlined Ingestion Process

Value Added
Self-Service
Data-driven

Timeliness
Always Ready
Easy to Find

Scale
Robust
Support Growth

Flexibility
Easy Modified
Automated
Streamlined

Quality
Explicit Visibility
Trustworthy

A data lake is an integral component of modern data management, serving as a foundational element for organizations striving to become data-driven. It encompasses several crucial characteristics, each of which contributes significantly to its value and utility.

At its core, a data lake is designed for scalability, capable of accommodating vast volumes of data. This scalability is essential in today's data-rich environment, where information accumulates at an unprecedented rate. A robust infrastructure ensures that the data lake can handle diverse data types, ranging from structured to unstructured, without compromising its performance. This combination of scalability and robustness ensures that the data lake can support an organization's growth without data-related constraints.

One of the standout features of a data lake is its self-service capability. Users across the organization can access and retrieve data independently, reducing the burden on IT teams and accelerating decision-making processes. This self-service aspect adds substantial value by promoting agility and enabling users to extract insights swiftly. Additionally, it empowers individuals within the organization to explore and analyze data, fostering a culture of data-driven decision-making.

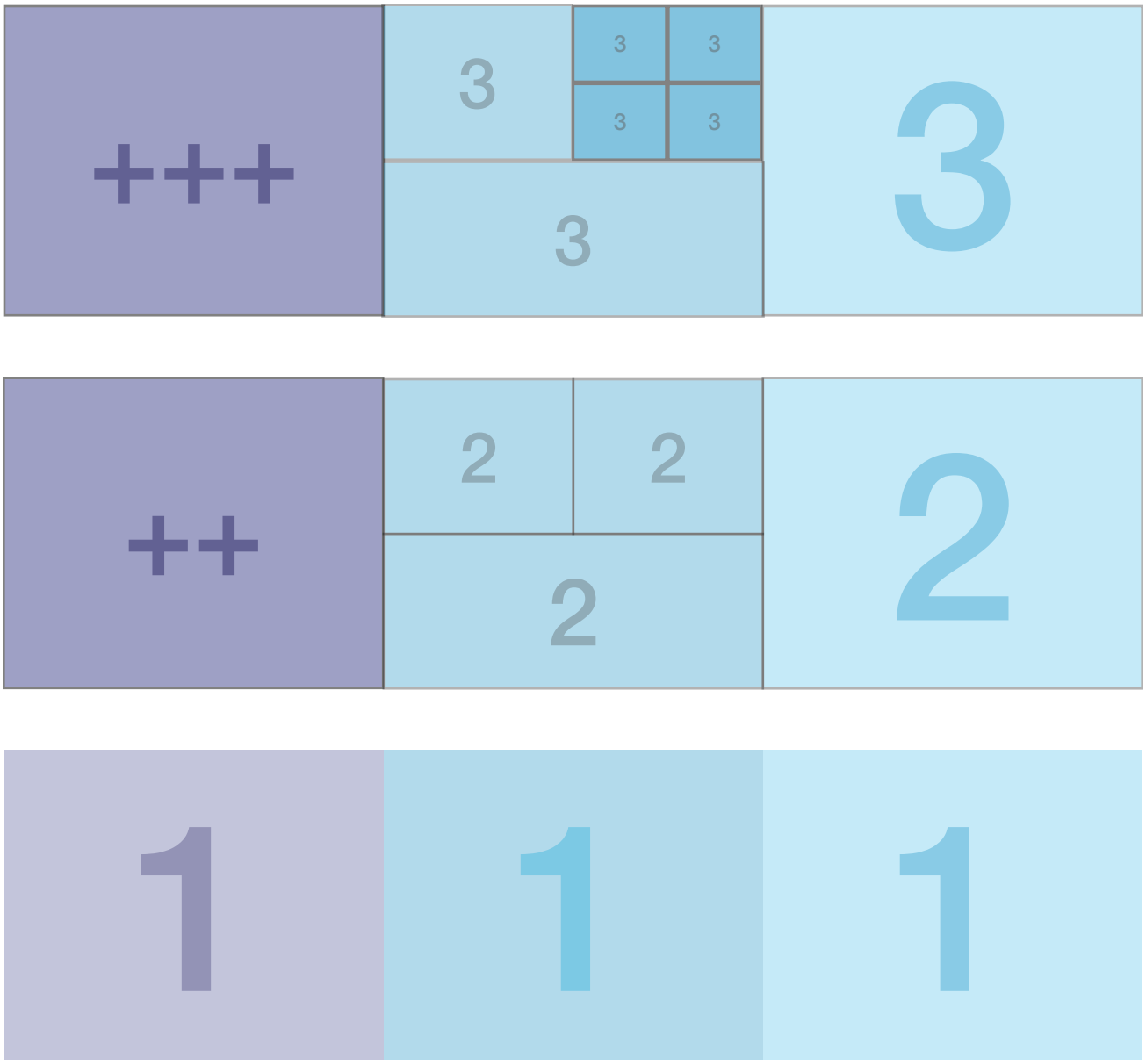
Data lakes are engineered to be always ready. This means that data is readily available for analysis, ensuring that time-sensitive insights can be derived promptly. The timeliness of data retrieval and analysis can be a critical factor in various business scenarios. Furthermore, data lakes are designed with accessibility in mind, making it easy for users to find the specific data they require. This accessibility streamlines workflows and minimizes the time spent searching for relevant information.

Flexibility is another key attribute of data lakes. They can seamlessly adapt to evolving business needs and changing data requirements. This adaptability is particularly crucial in dynamic environments where data formats and sources may vary over time. Whether it's incorporating new data sources or modifying existing data structures, a data lake's flexibility ensures that it remains aligned with the organization's goals.

Data quality is a fundamental aspect of any data management strategy, and data lakes are no exception. They offer explicit visibility into data quality, enabling organizations to establish and maintain trustworthy data sources. This visibility builds confidence among users that the data they access is accurate and reliable, reinforcing the credibility of data-driven insights.

Efficiency is a hallmark of data lakes. They are designed to automate and streamline data ingestion processes, reducing manual intervention and the risk of errors. This automation not only enhances data processing speed but also ensures that data is ingested consistently, preserving its integrity.

Data Lake serves as a comprehensive solution of self-service capability that empowers users, timeliness and accessibility to promote quick decision-making. Quality and trustworthiness ensure the reliability of insights. Together, these characteristics make the data lake an invaluable asset in the quest for data-driven excellence.



STORYTELLING

Bridging Numbers and Narratives

Data scientists should be adept at telling a compelling story with data. This involves creating narratives that make complex data and analysis accessible and understandable to non-technical stakeholders.

Certainly, within the context of data science, "Communication Skills" refer to the ability of data scientists to effectively convey their findings, insights, and results to both technical and non-technical audiences. This skill set is essential because data-driven insights need to be translated into actionable information that can inform decision-making within a business.

Effective communication skills are vital in ensuring that data science insights lead to informed decisions and positive outcomes within an organization. It bridges the gap between data analysis and real-world impact.

The art of data storytelling emerges as a vital skill, transcending the boundaries of data analysis and venturing into the territory of human understanding and decision-making. Data, on its own, can be abstract and impenetrable, often speaking a language of numbers and figures that remains inaccessible to many. However, the moment data is woven into a story, it becomes a force of enlightenment, and transformation.

Data storytelling is not merely a process of presenting facts and figures; it's the craft of crafting a compelling narrative. It's about taking the reader or audience on a journey, guiding them through the data's twists and turns to a clear and resonant destination. The storyteller's palette includes data visualizations, insightful analyses, and a deep understanding of the context. The stories they tell can expose trends, highlight outliers, and unveil profound insights that might otherwise remain hidden.

The craft is built from the merger of left-brain analytics and right-brain creativity. It's a fusion of science and art, precision and passion. A great data story is structured to engage and educate, to elicit emotions and provoke action. It navigates the fine balance between too much detail and too little context, between overwhelming complexity and oversimplified generalization.

Masters of this art use storytelling techniques to frame a problem and deliver a resolution that leaves a lasting impact. They employ visuals to simplify complexity, metaphors to make the unfamiliar familiar, and anecdotes to humanize the data. Through the art of data storytelling, data becomes more than just information; it becomes a force for change, an instrument of persuasion, and a vessel for empowerment.

To excel in data storytelling, one must be part analyst, part communicator, part artist, and part detective. It's about interpreting the language of data and translating it into the language of people. It's about understanding that the real power of data lies not in the numbers themselves but in the stories they can tell. In a world awash with data, it's the storytellers who can harness its potential, making the obscure clear, the complex understandable, and the mundane remarkable. The art of data storytelling is where the beauty of data science truly shines, illuminating the path to informed decisions, inspired actions, and meaningful change.

The ability to craft compelling data narratives is an art that transcends the mere presentation of numbers, charts, and graphs. It is the process of weaving together a meaningful and impactful story from data, one that resonates with your audience and empowers them to make informed decisions. Much like a skilled author crafting a novel or a screenwriter developing a screenplay, a data scientist crafting data narratives has the essential task of engaging the audience and conveying insights effectively.

At the core of crafting data narratives lies the data itself. Data scientists gather, clean, and analyze vast amounts of information to extract valuable insights. But these insights only become meaningful when placed in the right context. Understanding the context, whether it's the industry, business problem, or societal issue, is the foundation upon which data narratives are built. Without context, data is just noise; with context, it transforms into a powerful narrative.

Much like an artist, the data storyteller employs creativity to shape the narrative. It involves choosing the right visualizations, structuring the story, and creating a flow that guides the audience. This creative process demands not only a deep understanding of data but also an appreciation for the art of storytelling. The data narrative should not only convey information but also evoke emotions, spark curiosity, and encourage action.

Effective data narratives are always tailored to the audience. Whether presenting to business executives, policymakers, or the general public, the storyteller adapts the narrative to match the audience's level of expertise and interests. An audience-centric approach ensures that the story is relatable and resonates with the people it aims to reach.

The data narratives serves as a bridge between data and action. It elevates raw information into a story that has the power to inform, educate, and inspire. If you're a data scientist, analyst, or a business professional, mastering the craft of data storytelling can be a transformative skill that enables you to navigate the data-rich landscape of the modern world and create meaningful change. Whether you're presenting your findings or simply sharing insights, your ability to communicate clearly is paramount.

In the world of data science, the phrase "a picture is worth a thousand words" takes on profound significance. The power of visualization lies in its ability to distill complex datasets into comprehensible and memorable images.

Visualization simplifies complexity. It transforms raw data into clear and accessible forms. Humans are inherently visual creatures. We remember images far more effectively than raw numbers or text. Narratives are more compelling when bolstered by visual evidence. The combination of data-driven stories and visual proof enhances the effectiveness of communication. Visualizations are a universal language that transcends barriers like language or technical expertise.

The power of visualization is an art and science. It's a fusion of creativity and data-driven precision. By harnessing this power, data scientists can transform numbers into narratives, aiding better understanding, decision-making, and change in an increasingly data-centric world.

Nice simple visualizations

Examples

read.me()

The Power of Visualization Here, we explore the extraordinary impact of visualization in the world of data.

1. **Clarity and Comprehension**: Visualization simplifies complexity. It transforms raw data into clear and accessible forms. Whether it's a bar chart, scatterplot, or heat map, visual representations help us quickly grasp trends, patterns, and relationships within data. Complex datasets become intelligible, enabling data scientists, analysts, and stakeholders to make informed decisions.

2. **Memory and Impact**: Humans are inherently visual creatures. We remember images far more effectively than raw numbers or text. A well-crafted visualization can imprint data insights in our memory. This makes it a powerful tool for conveying messages, sharing insights, and triggering action.

3. **Discovery and Exploration**: Visualizations aren't just tools for presenting conclusions; they're instruments for exploration and discovery. By manipulating visualizations interactively, data professionals can unearth hidden trends and outliers, enabling deeper insights.

4. **Effective Communication**: Data storytelling, another integral part of data science, relies heavily on visualization. Narratives are more compelling when bolstered by visual evidence. The combination of data-driven stories and visual proof enhances the effectiveness of communication.

5. **Universal Language**: Visualizations are a universal language that transcends barriers like language or technical expertise. Whether you're communicating with a fellow data scientist or a non-technical stakeholder, a well-designed graph can convey insights in a format anyone can understand.

6. **Decision-Making**: Visualizations empower better decision-making. They transform abstract figures into actionable information. Whether it's optimizing business operations, tracking disease outbreaks, or planning resource allocation, visualizations guide effective choices.

7. **Contextualization**: Visualization places data in context. By illustrating data in relation to geographic locations, time periods, or other variables, it provides a comprehensive view. The ability to contextualize data is especially valuable for scenario planning and trend analysis.

8. **Engagement and Storytelling**: Compelling stories often include compelling visuals. Engaging visualizations captivate audiences, ensuring that data-driven messages resonate. They are essential tools for crafting narratives that compel action.

9. **Rapid Insights**: Time is a valuable resource. Visualizations offer a swift route to insights. Data scientists can promptly discern trends or anomalies, which is particularly crucial in scenarios like fraud detection or real-time monitoring.

10. **Motivation and Advocacy**: Visualizations can be motivational tools. They highlight challenges or successes, driving teams and organizations to action. In addition, they serve as persuasive advocacy instruments for causes that rely on data to bolster their message.

The power of visualization is an art and science. It's a fusion of creativity and data-driven precision. By harnessing this power, data scientists can transform numbers into narratives, aiding better understanding, decision-making, and change in an increasingly data-centric world.

1. **Data Storytelling**: Data scientists should be adept at telling a compelling story with data. This involves creating narratives that make complex data and analysis accessible and understandable to non-technical stakeholders.

2. **Data Visualization**: Proficiency in data visualization tools and techniques is crucial. Data scientists should be able to create clear and informative charts, graphs, and dashboards that visually represent data trends and patterns.

3. **Clear Reporting**: Effective written communication is essential. Data scientists must produce clear and concise reports that explain their methodology, findings, and recommendations in a language that stakeholders can comprehend.

4. **Oral Communication**: Data scientists often need to present their findings in meetings or presentations. Strong oral communication skills enable them to articulate their results, answer questions, and engage with diverse audiences.

5. **Stakeholder Engagement**: Communication skills extend to engaging with stakeholders to understand their specific needs and requirements. Data scientists should be able to ask the right questions and actively listen to stakeholder concerns.

6. **Translation of Jargon**: Data professionals often use technical jargon. Effective communication involves the ability to translate this jargon into plain language, ensuring that non-technical team members and decision-makers can grasp the significance of the data.

7. **Interdisciplinary Collaboration**: Data scientists often work in interdisciplinary teams. Effective communication is crucial for collaborating with professionals from different backgrounds, such as business analysts, engineers, and domain experts.

8. **Visual and Written Literacy**: Data scientists should possess visual and written literacy, which includes the ability to critically evaluate data visualizations and written reports for clarity, accuracy, and potential bias.

9. **Ethical Communication**: Communicating about data-driven insights should also consider ethical implications. Data scientists must address privacy and security concerns when sharing results and be transparent about data sources and limitations.

Chapter 7: Data Storytelling - Bridging Numbers and Narratives

Section 1: The Art of Data Storytelling

In the data-driven landscape, the ability to tell compelling stories with data is a skill that sets apart effective data scientists. This section introduces the concept of data storytelling, explaining why it's crucial in data science. It explores the impact of well-structured narratives in making data insights actionable.

Section 2: Crafting Data Narratives

This section dives into the process of crafting data narratives. It covers techniques for structuring stories, choosing the right visuals, and keeping your audience engaged. Learn to translate complex data into relatable tales that resonate with stakeholders.

Section 3: Effective Communication

Effective communication is the key to delivering data stories that drive decision-making. This part of the chapter provides guidance on clear and persuasive data presentation. It includes tips for tailoring your message to various audiences and using data to answer critical questions.

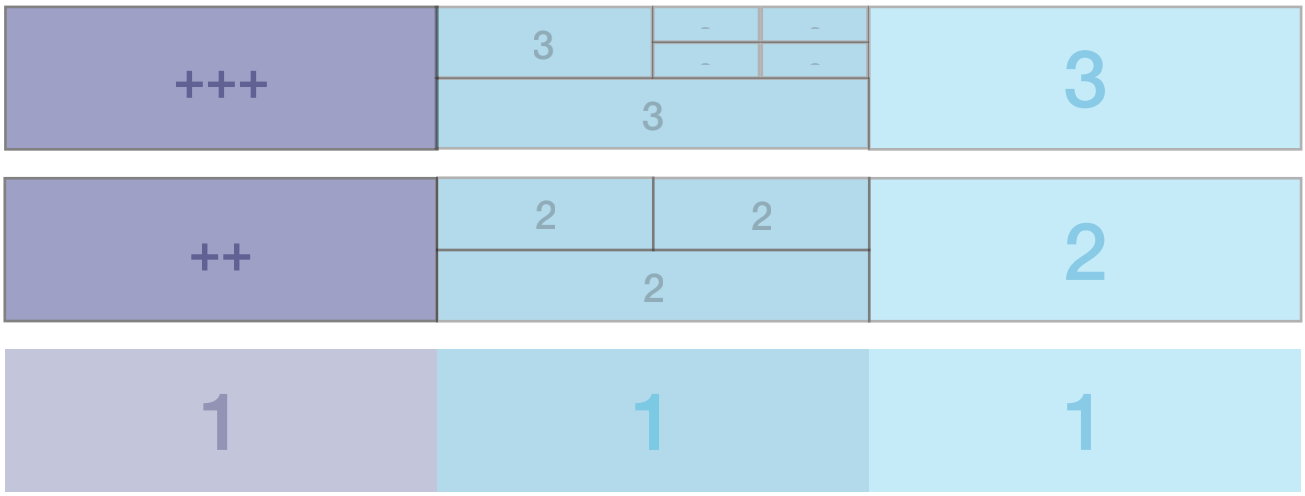
Section 4: The Power of Visualization

Visuals are an essential component of data storytelling. Here, you'll explore the art of data visualization, from selecting the right charts to enhancing their impact. Learn how to create compelling visuals that enrich your narratives.

Section 5: Data Storytelling in Practice

The final section brings it all together with real-world examples of data storytelling. You'll see how data narratives are applied in various industries and scenarios, turning information into decisions and data into action.

By the end of this chapter, you'll be equipped with the knowledge and skills to become a master data storyteller, transforming raw data into narratives that drive change and empower informed choices.



Opening phases

Dashboard Structure

...

`left_plot()`

`morning()`
`lunch()`
`afternoon()`
`night()`
`weekend()`

`right_plot()`

Example of

Telemetry

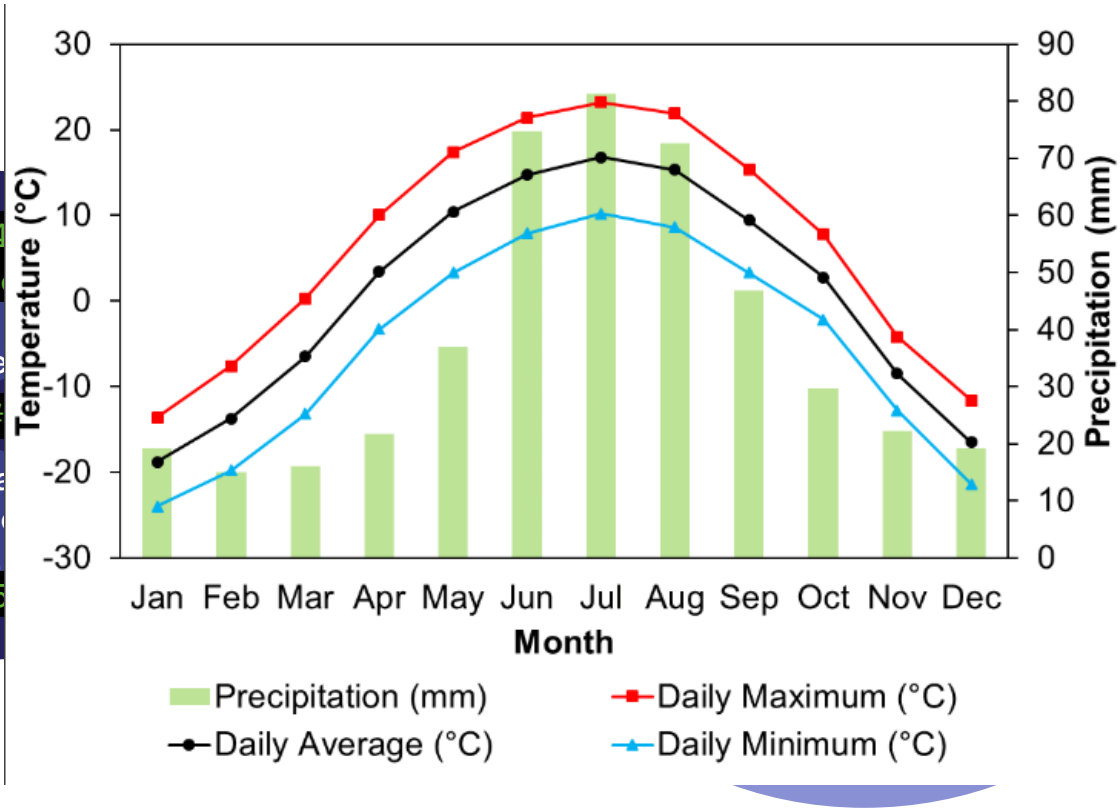
`assets_list()`
`market_stats_grid()`
`cointegration_grid()`

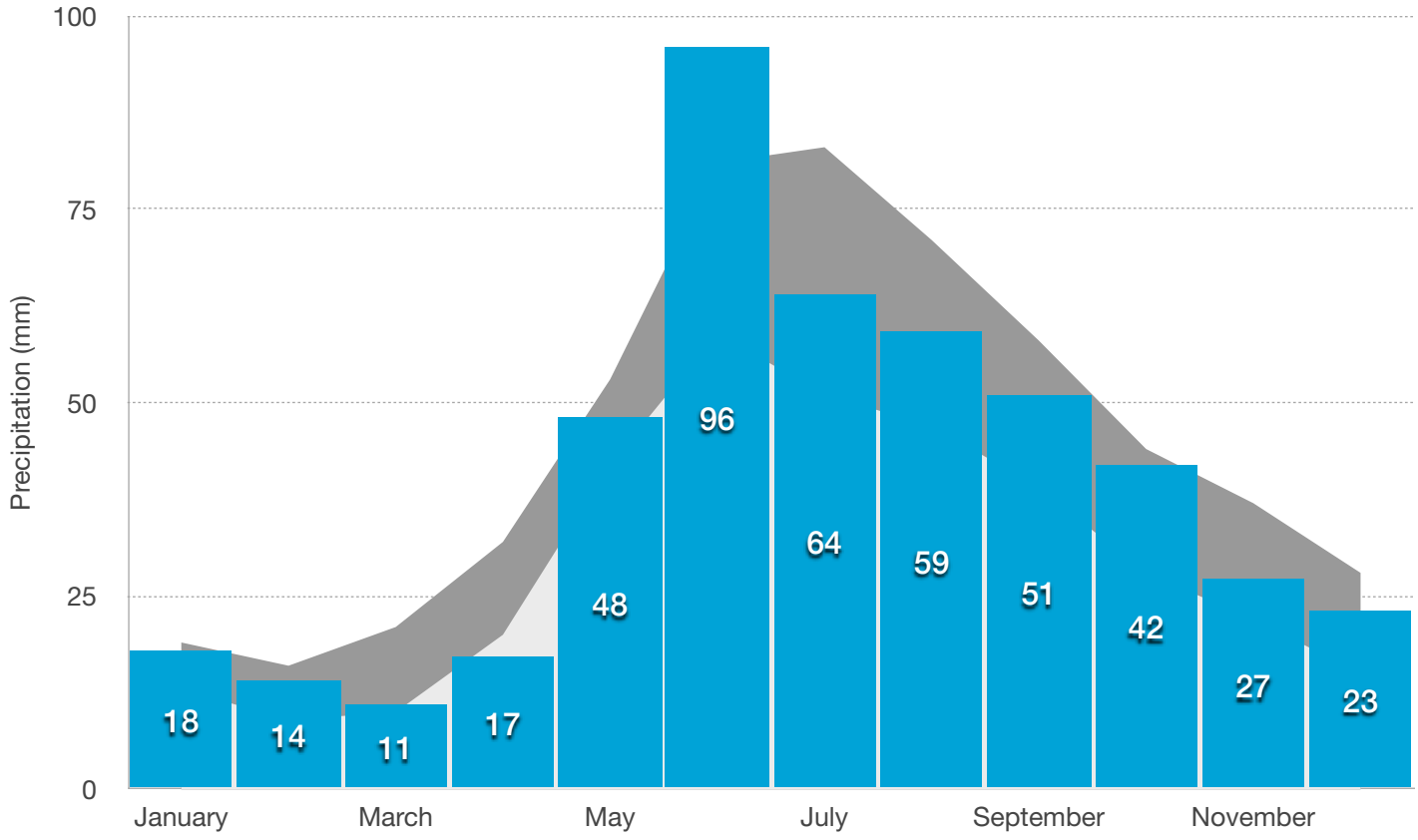
`center_plot()`

`general_timeline_plot()`

`news_grid()`
`calendar_grid()`
`earnings_list()`

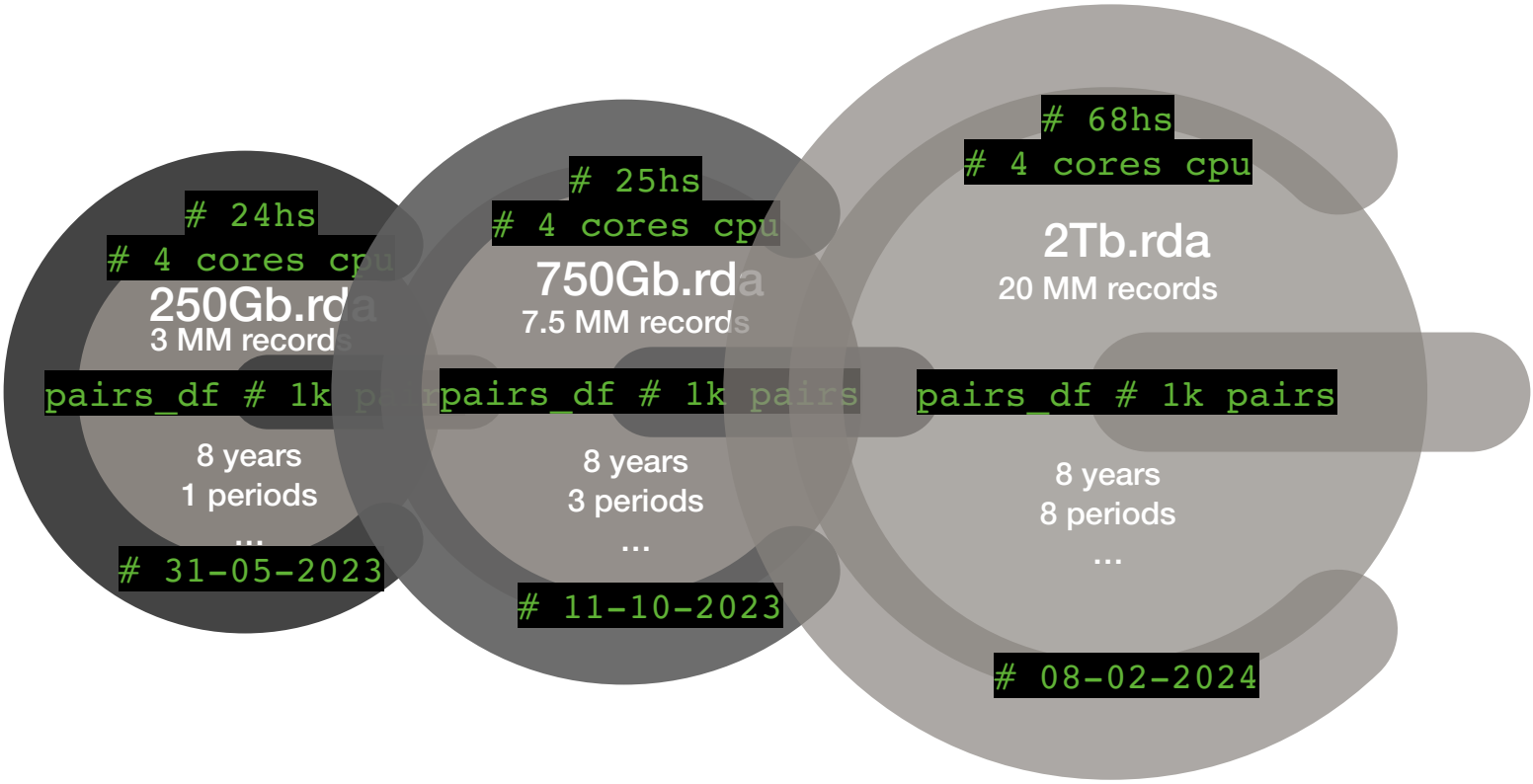
```
# 24
# 4 cor
250G
4 MM re
pairs_df #
8 yea
1 peri
...
# 31-05
```

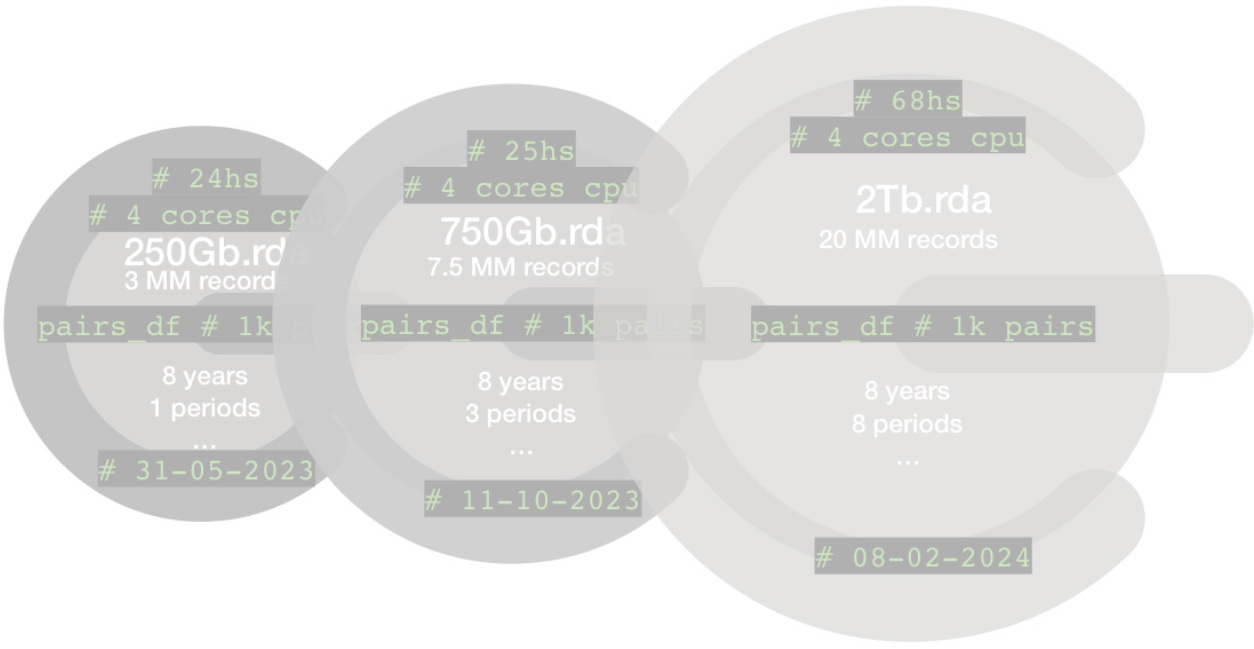





.0345
.0956
.0234
.0104
.0256

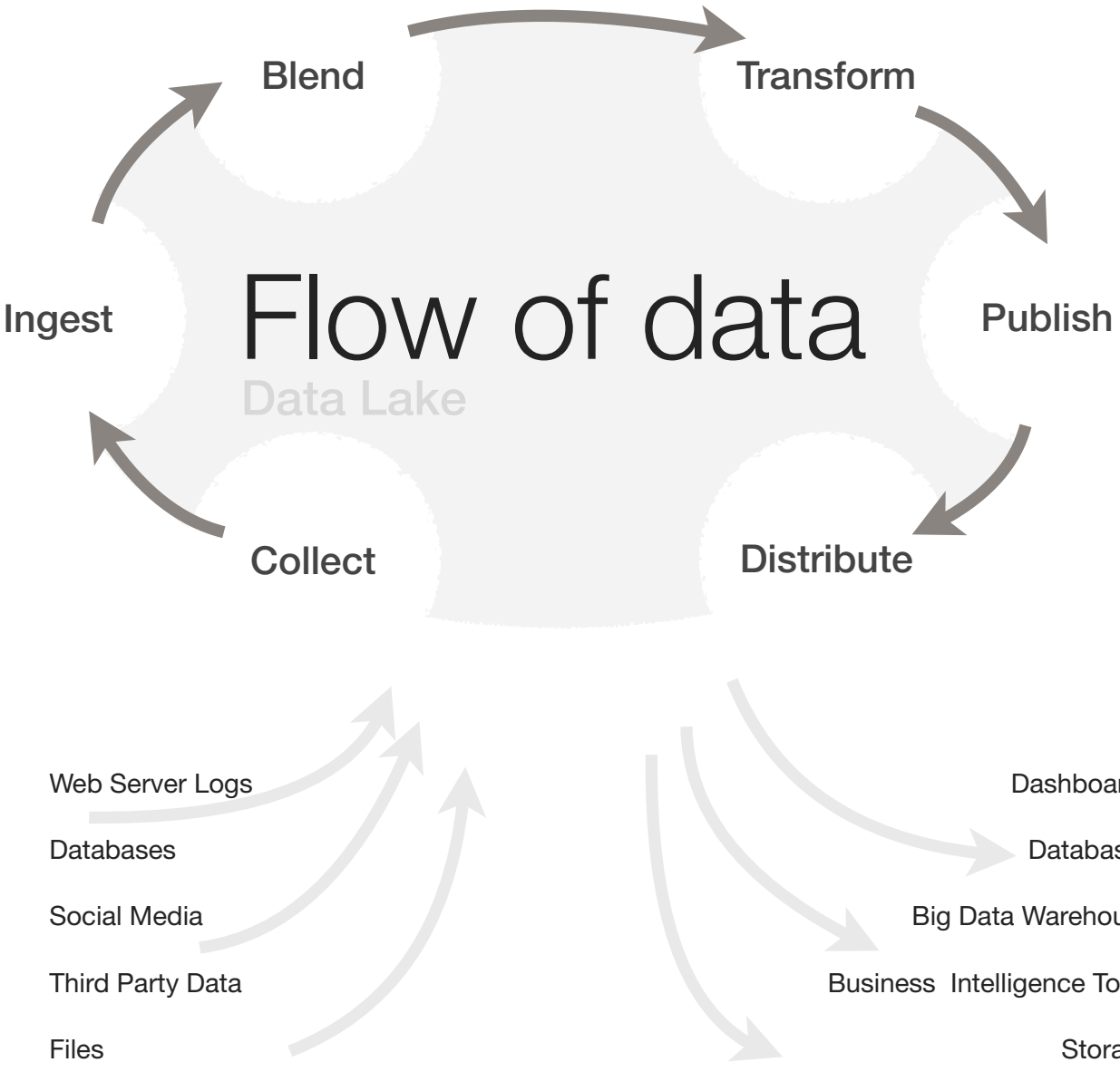
Archive

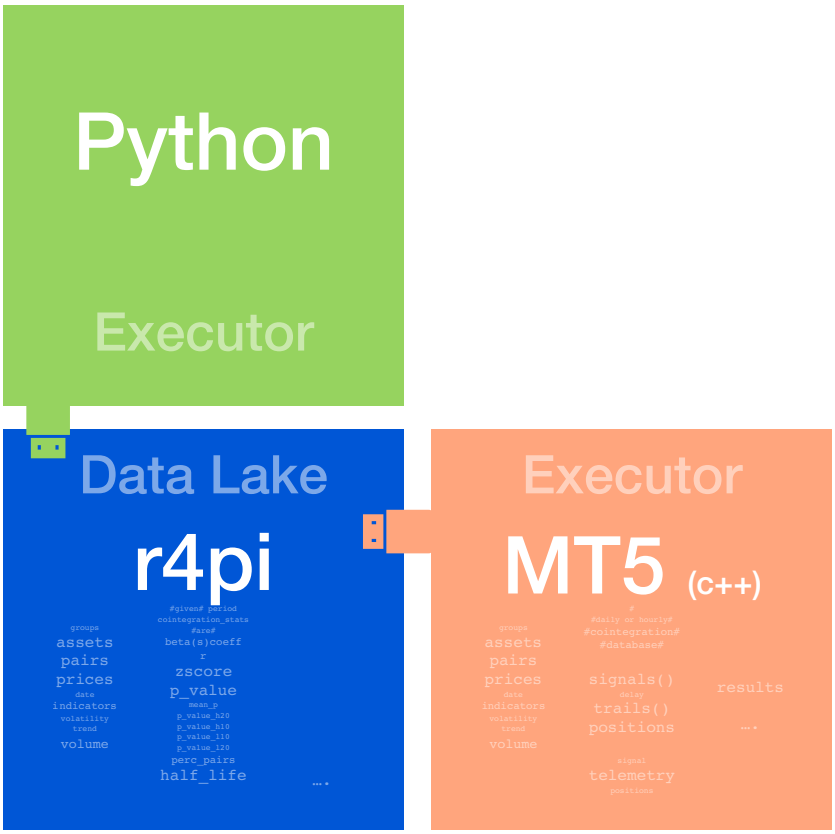

24hs
4 cores cpu
250Gb.rda
3 MM record
pairs_df # 1k pairs
8 years
1 periods
...
31-05-2023
25hs
4 cores cpu
750Gb.rda
7.5 MM records
pairs_df # 1k pairs
8 years
3 periods
...
11-10-2023
68hs
4 cores cpu
2Tb.rda
20 MM records
pairs_df # 1k pairs
8 years
8 periods
...
08-02-2024

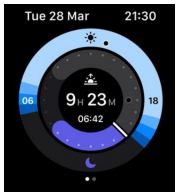


Source, Gathering and Quality Code Samples Load from API? Web scraping

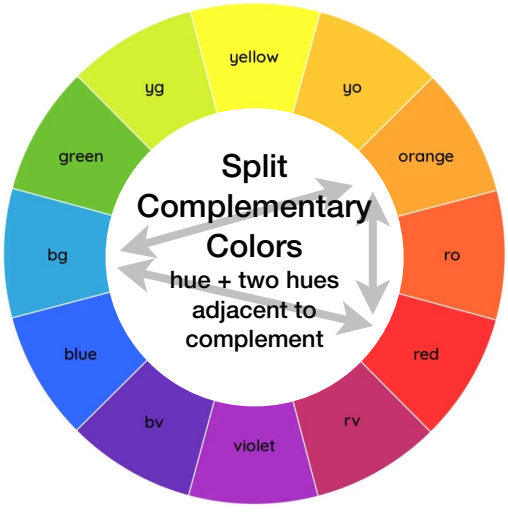
PROCESSING AND CLEANING

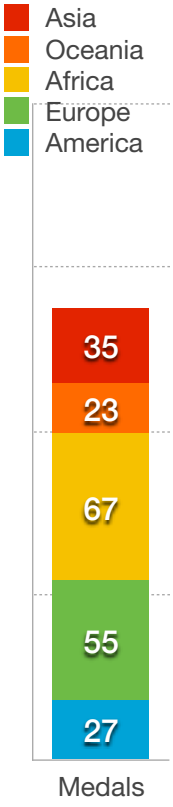


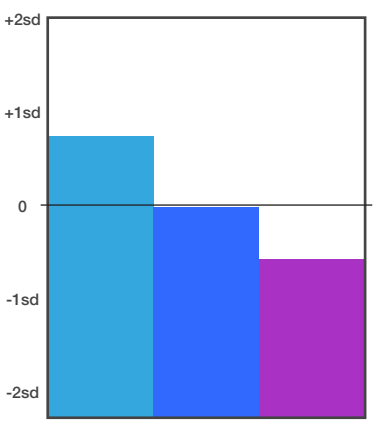

Python
Executor
Data Lake
r4pi
groups
assets
pairs
prices
date
indicators
volatility
trend
volume
#given# period
cointegration_stats
#axs#
beta(s)coeff
r
zscore
p_value
mean_p
p_value_h20
p_value_h10
p_value_l10
p_value_l20
perc_pairs
half_life
Executor
MT5 (c++)
#daily or hourly#
#cointegration#
#database#
signals()
delay
trails()
positions
signal
telemetry
results

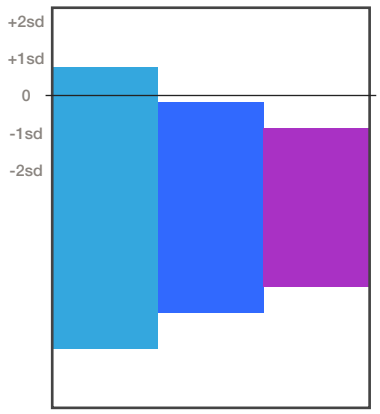

Tue 28 Mar
21:30
06
18
9H 23M
06:42

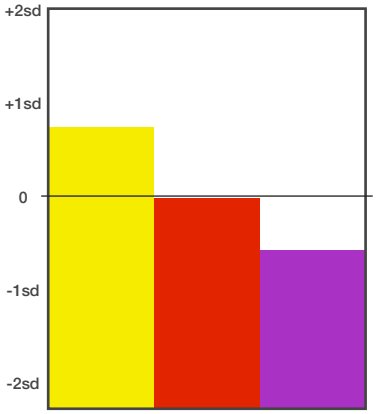

100%

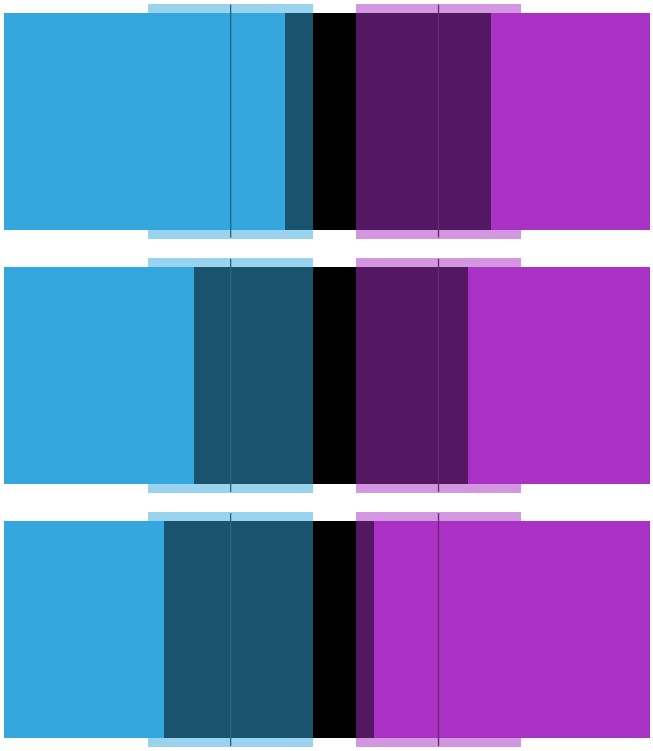

yellow
yg
yo
green
orange
bg
ro
blue
red
bv
violet
rv
Split
Complementary
Colors
hue + two hues
adjacent to
complement


Asia
Oceania
Africa
Europe
America
35
23
67
55
27
Medals

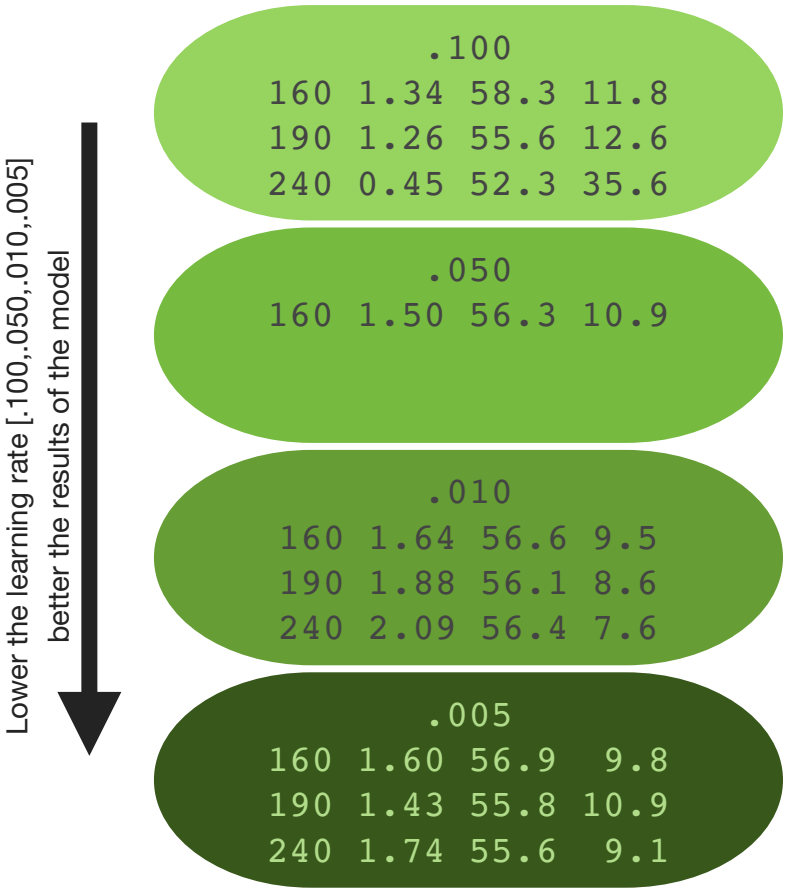

+2sd
+1sd
0
-1sd
-2sd


+2sd
+1sd
0
-1sd
-2sd


+2sd
+1sd
0
-1sd
-2sd

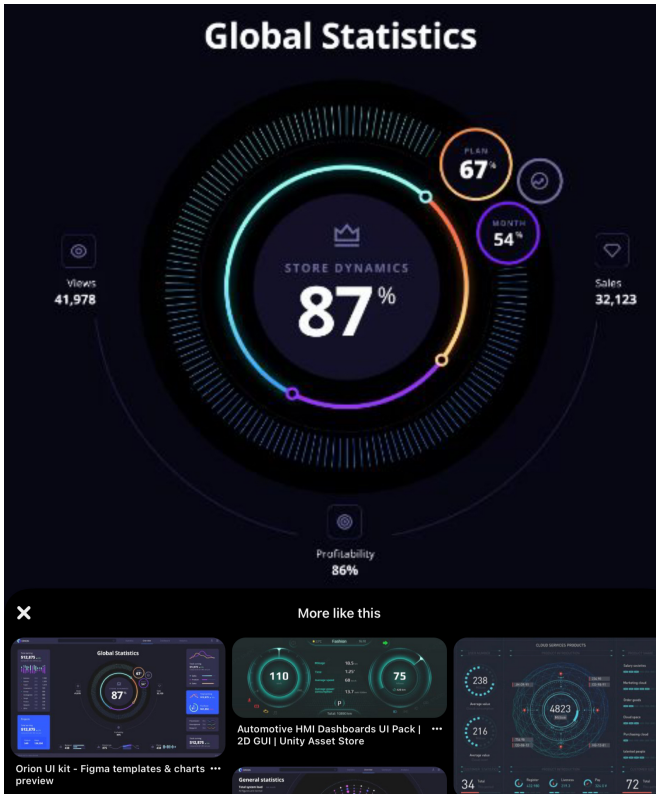


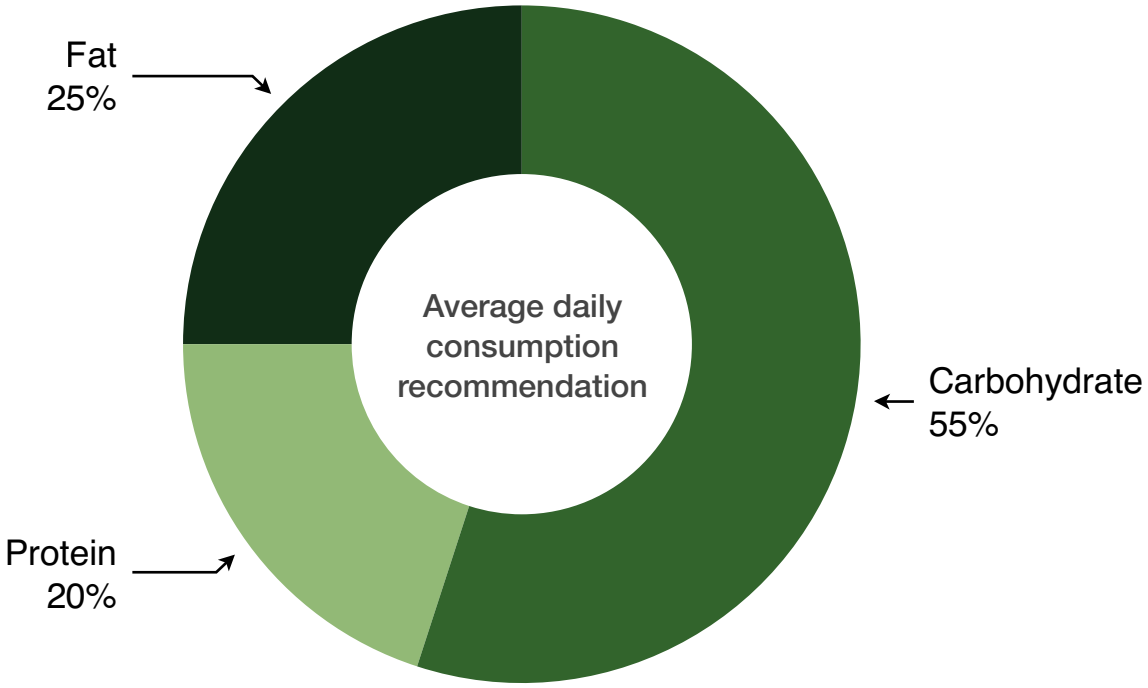

6M
2Y
3..5Y
Long-term investments


Lower the learning rate [.100,.050,.010,.005]
better the results of the model
.100
160 1.34 58.3 11.8
190 1.26 55.6 12.6
240 0.45 52.3 35.6
.050
160 1.50 56.3 10.9
.010
160 1.64 56.6 9.5
190 1.88 56.1 8.6
240 2.09 56.4 7.6
.005
160 1.60 56.9 9.8
190 1.43 55.8 10.9
240 1.74 55.6 9.1

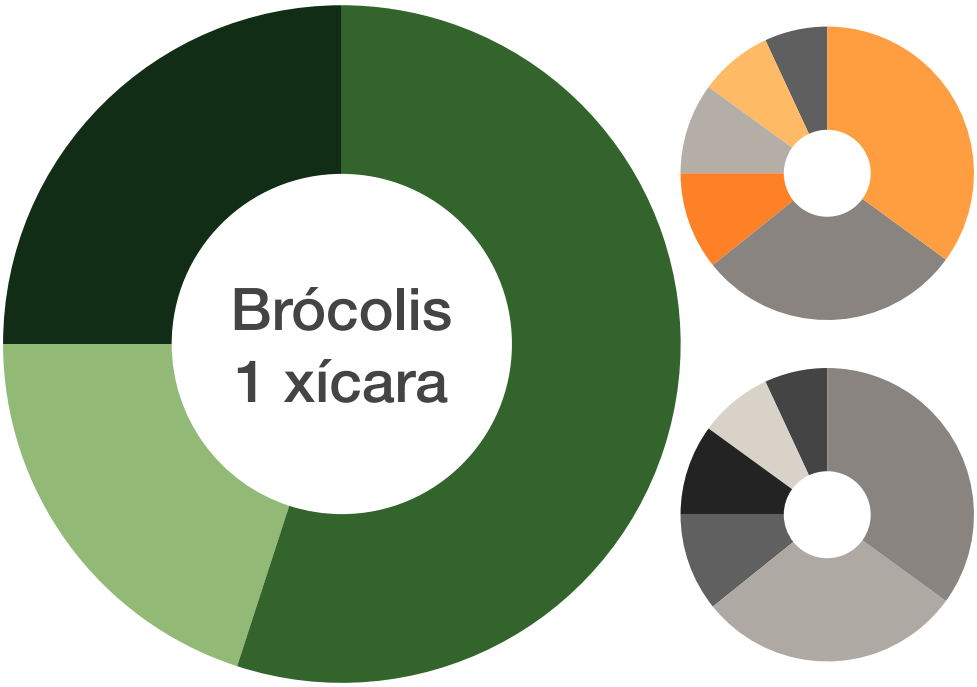

midnight
3
6
9
noon
15
18
21
24
hours

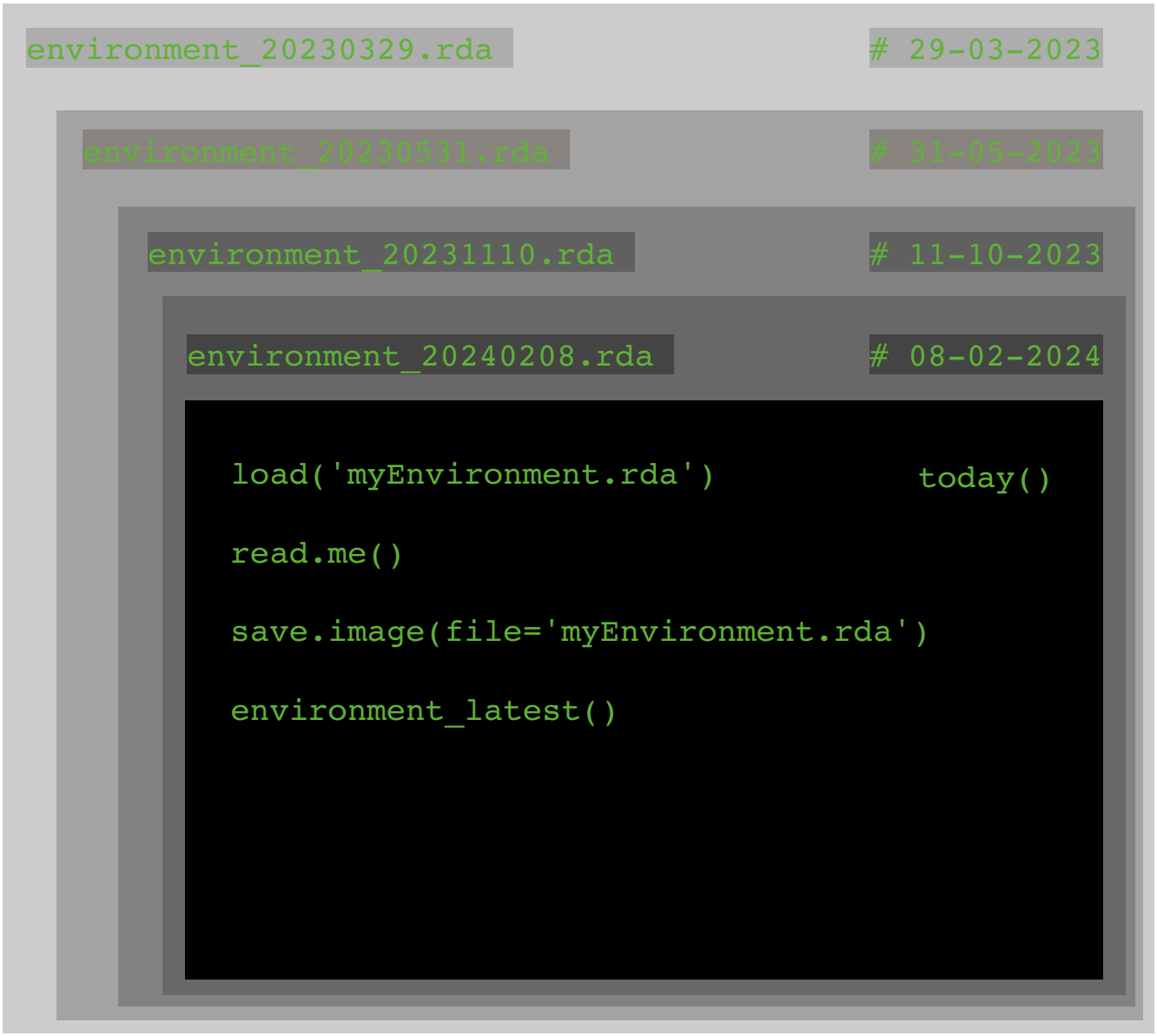
Badge


Global Statistics
PLAN
67%
MONTH
54%
STORE DYNAMICS
87%
Views
41,978
Sales
32,123
Profitability
86%
More like this
Automotive HMI Dashboards UI Pack | 2D GUI | Unity Asset Store
Orion UI kit - Figma templates & charts preview

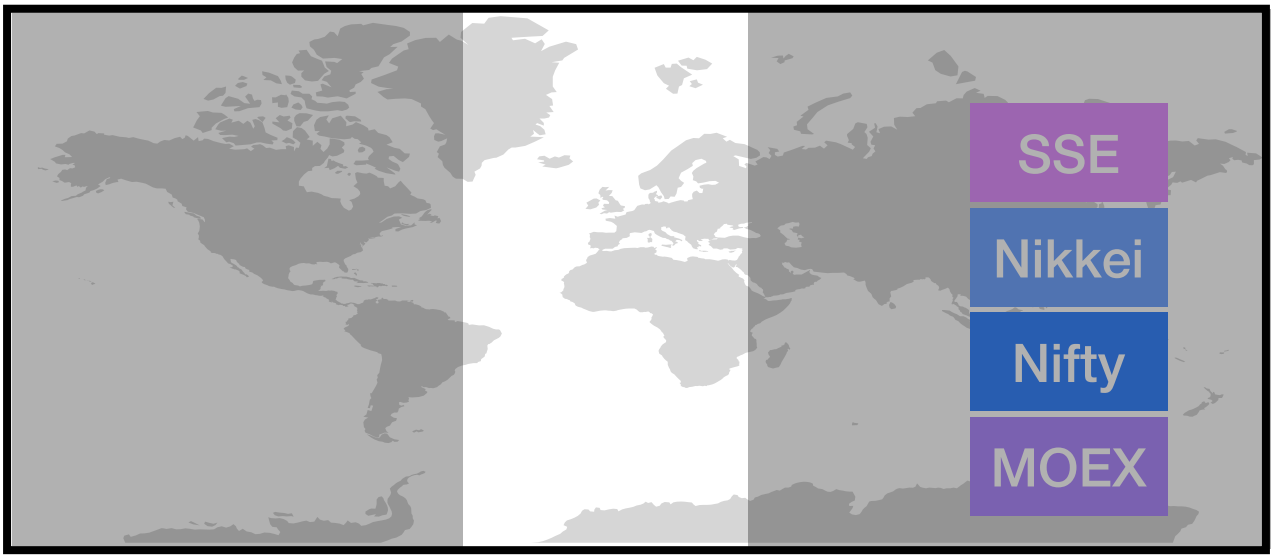

Fat
25%
Average daily consumption recommendation
Carbohydrate
55%
Protein
20%

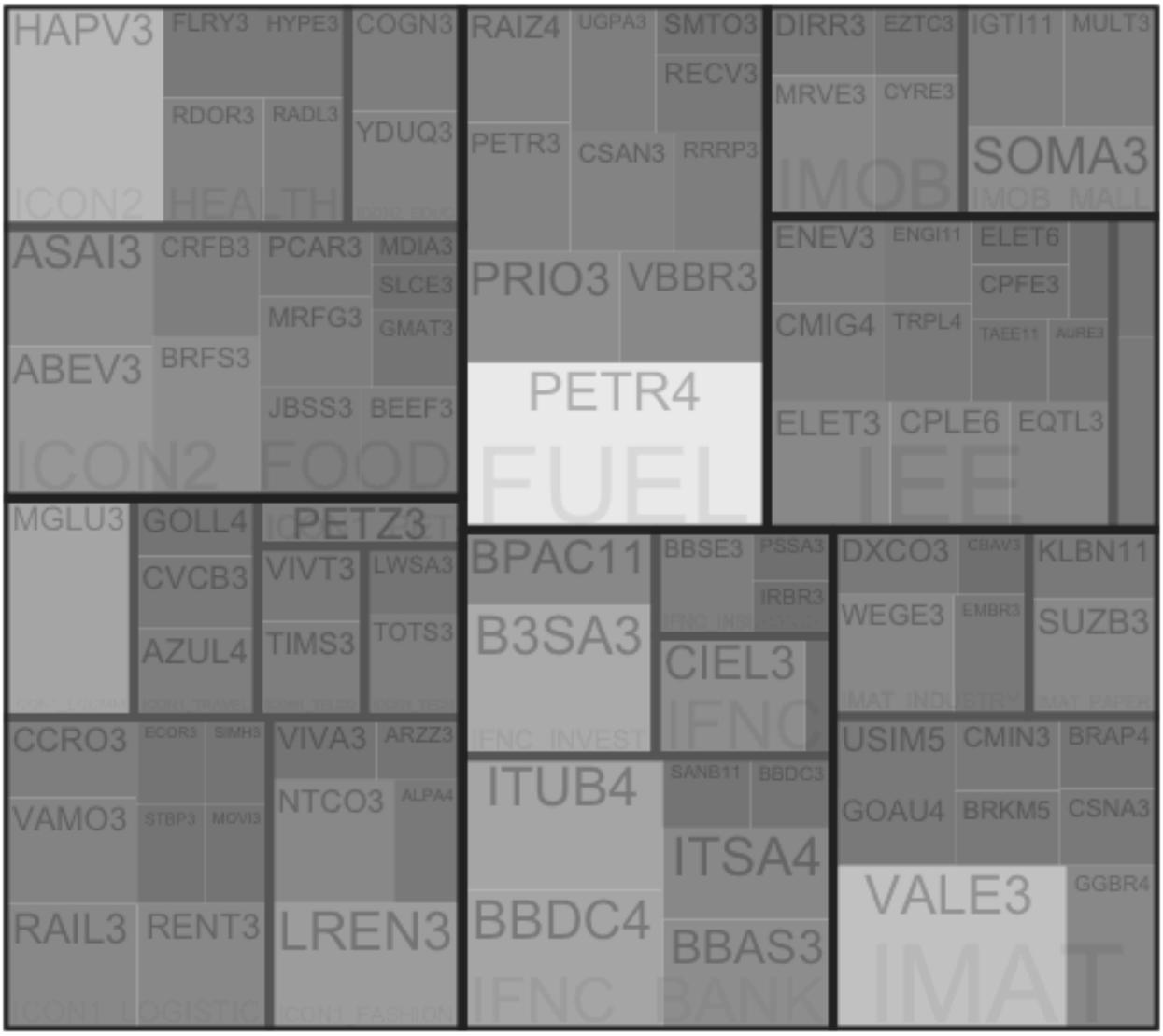
Badge



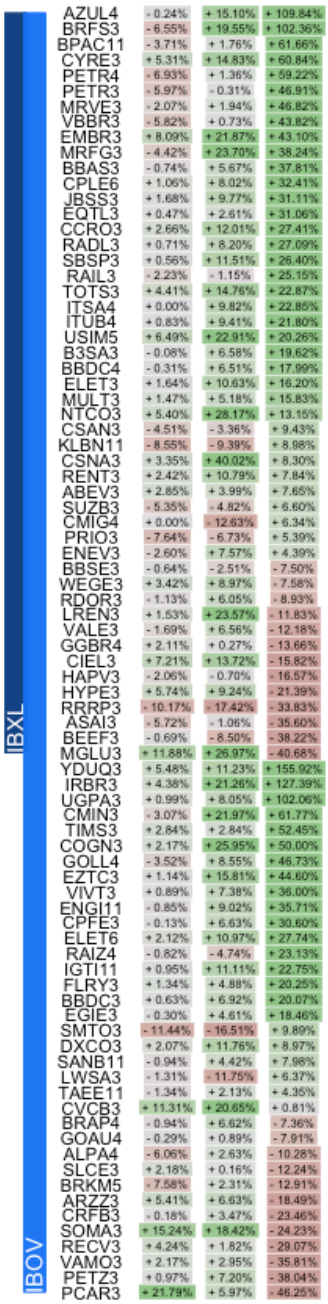


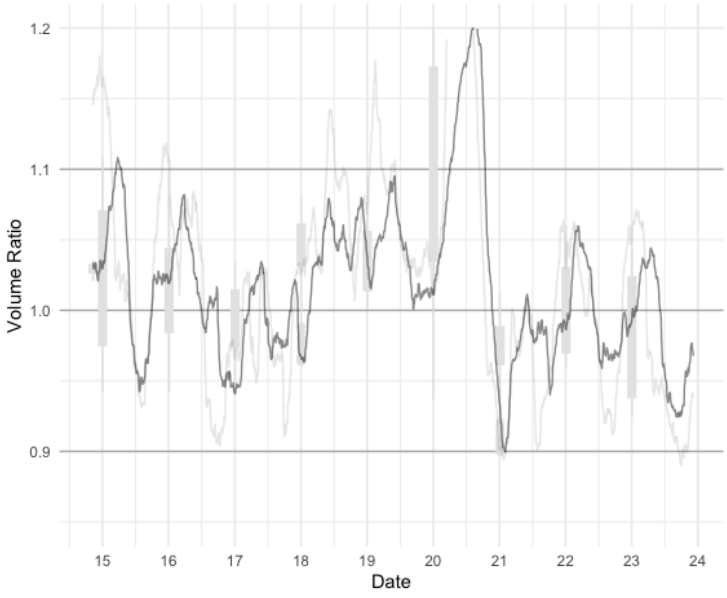
Time Machine

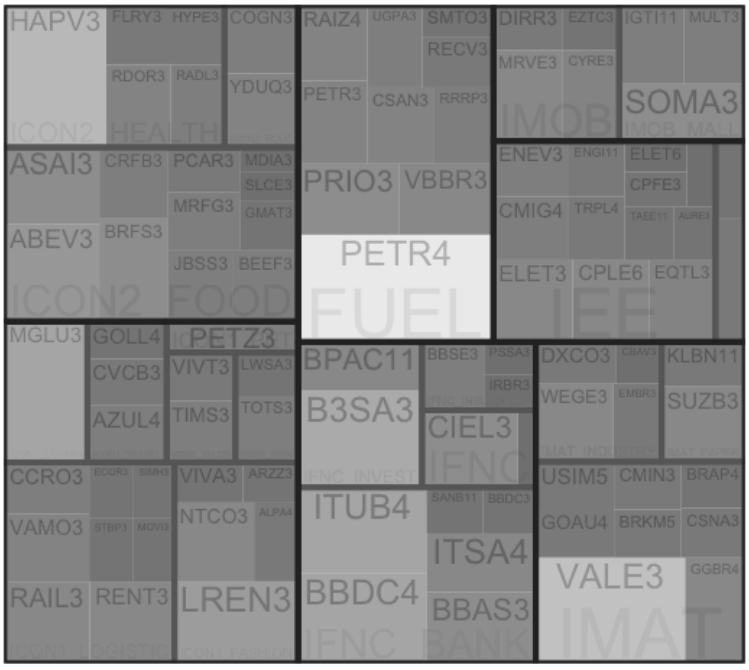

SSE
Nikkei
Nifty
MOEX

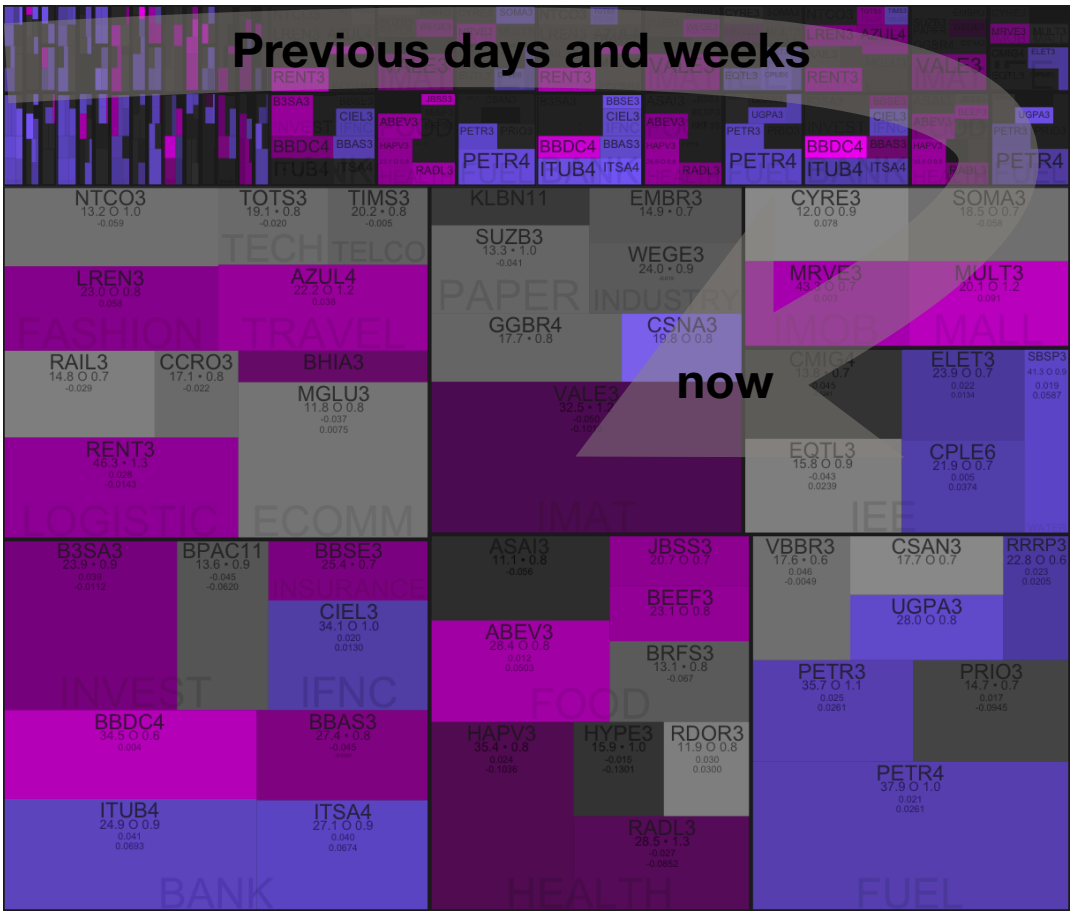

HAPV3
FLRY3
HYPE3
COGN3
RDOR3
RADL3
YDUQ3
ICON2
HEALTH
RAIZ4
UGPA3
SMTO3
RECV3
PETR3
CSAN3
RRRP3
PRIO3
VBBR3
PETR4
FUEL
DIRR3
EZTC3
IGTI11
MULT3
MRVE3
CYRE3
IMOB
SOMA3
ASAI3
CRFB3
PCAR3
MDIA3
SLCE3
MRFG3
GMAT3
ABEV3
BRFS3
JBSS3
BEEF3
ICON2
FOOD
ENEV3
ENGI11
ELET6
CPFE3
CMIG4
TRPL4
TAEE11
AURE3
ELET3
CPLE6
EQTL3
IEE
MGLU3
GOLL4
PETZ3
CVCB3
VIVT3
LWSA3
AZUL4
TIMS3
TOTS3
BPAC11
BBSE3
PSSA3
IRBR3
B3SA3
CIEL3
IFNC
DXCO3
CBAV3
KLBN11
WEGE3
EMBR3
SUZB3
CCRO3
ECOR3
SIMH3
VAMO3
STBP3
MOVI3
RAIL3
RENT3
VIVA3
ARZZ3
NTCO3
ALPA4
LREN3
ITUB4
SANB11
BBDC3
ITSA4
BBDC4
BBAS3
IFNC
BANK
USIM5
CMIN3
BRAP4
GOAU4
BRKM5
CSNA3
VALE3
GGBR4
IMAT

IBOV

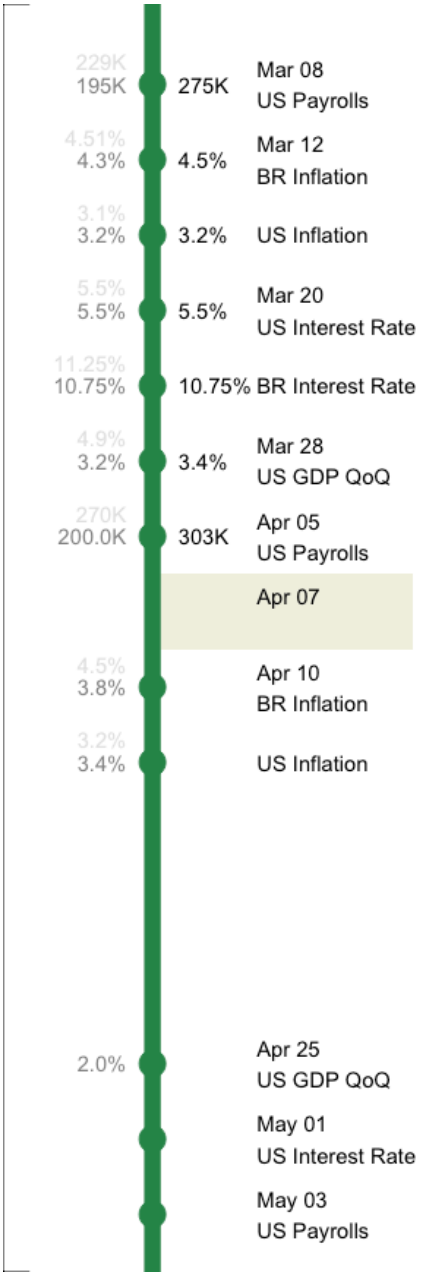

IBXL
IBOV
AZUL4 - 0.24% + 15.10% + 109.84%
BRFS3 - 6.55% + 19.55% + 102.36%
BPAC11 - 3.71% + 1.76% + 61.66%
CYRE3 + 5.31% + 14.83% + 60.84%
PETR4 - 6.93% + 1.36% + 59.22%
PETR3 - 5.97% - 0.31% + 46.91%
MRVE3 - 2.07% + 1.94% + 46.82%
VBBR3 - 5.82% + 0.73% + 43.82%
EMBR3 + 8.09% + 21.87% + 43.10%
MRFG3 - 4.42% + 23.70% + 38.24%
BBAS3 - 0.74% + 5.67% + 37.81%
CPLE6 + 1.06% + 8.02% + 32.41%
JBSS3 + 1.68% + 9.77% + 31.11%
EQTL3 + 0.47% + 2.61% + 31.06%
CCRO3 + 2.66% + 12.01% + 27.41%
RADL3 + 0.71% + 8.20% + 27.09%
SBSP3 + 0.56% + 11.51% + 26.40%
RAIL3 - 2.23% - 1.15% + 25.15%
TOTS3 + 4.41% + 14.76% + 22.87%
ITSA4 + 0.00% + 9.82% + 22.85%
ITUB4 + 0.83% + 9.41% + 21.80%
USIM5 + 6.49% + 22.91% + 20.26%
B3SA3 - 0.08% + 6.58% + 19.62%
BBDC4 - 0.31% + 6.51% + 17.99%
ELET3 + 1.64% + 10.63% + 16.20%
MULT3 + 1.47% + 5.18% + 15.83%
NTCO3 + 5.40% + 28.17% + 13.15%
CSAN3 - 4.51% - 3.36% + 9.43%
KLBN11 - 8.55% - 9.39% + 8.98%
CSNA3 + 3.35% + 40.02% + 8.30%
RENT3 + 2.42% + 10.79% + 7.84%
ABEV3 + 2.85% + 3.99% + 7.65%
SUZB3 - 5.35% - 4.82% + 6.60%
CMIG4 + 0.00% - 12.63% + 6.34%
PRIO3 - 7.64% - 6.73% + 5.39%
ENEV3 - 2.60% + 7.57% + 4.39%
BBSE3 - 0.64% - 2.51% - 7.50%
WEGE3 + 3.42% + 8.97% - 7.58%
RDOR3 - 1.13% + 6.05% - 8.93%
LREN3 + 1.53% + 23.57% - 11.83%
VALE3 - 1.69% + 6.56% - 12.18%
GGBR4 + 2.11% + 0.27% - 13.66%
CIEL3 + 7.21% + 13.72% - 15.82%
HAPV3 - 2.06% - 0.70% - 16.57%
HYPE3 + 5.74% + 9.24% - 21.39%
RRRP3 - 10.17% - 17.42% - 33.83%
ASAI3 - 5.72% - 1.06% - 35.60%
BEEF3 - 0.69% - 8.50% - 38.22%
MGLU3 + 11.88% + 26.97% - 40.68%
YDUQ3 + 5.48% + 11.23% + 155.92%
IRBR3 + 4.38% + 21.26% + 127.39%
UGPA3 + 0.99% + 8.05% + 102.06%
CMIN3 - 3.07% + 21.97% + 61.77%
TIMS3 + 2.84% + 2.84% + 52.45%
COGN3 + 2.17% + 25.95% + 50.00%
GOLL4 - 3.52% + 8.55% + 46.73%
EZTC3 + 1.14% + 15.81% + 44.60%
VIVT3 + 0.89% + 7.38% + 36.00%
ENGI11 - 0.85% + 9.02% + 35.71%
CPFE3 - 0.13% + 6.63% + 30.60%
ELET6 + 2.12% + 10.97% + 27.74%
RAIZ4 - 0.82% - 4.74% + 23.13%
IGTI11 + 0.95% + 11.11% + 22.75%
FLRY3 + 1.34% + 4.88% + 20.25%
BBDC3 + 0.63% + 6.92% + 20.07%
EGIE3 - 0.30% + 4.61% + 18.46%
SMTO3 - 11.44% - 16.51% + 9.89%
DXCO3 + 2.07% + 11.76% + 8.97%
SANB11 - 0.94% + 4.42% + 7.98%
LWSA3 - 1.31% - 11.75% + 6.37%
TAEE11 - 1.34% + 2.13% + 4.35%
CVCB3 + 11.31% + 20.65% + 0.81%
BRAP4 - 0.94% + 6.62% - 7.36%
GOAU4 - 0.29% + 0.89% - 7.91%
ALPA4 - 6.06% + 2.63% - 10.28%
SLCE3 + 2.18% + 0.16% - 12.24%
BRKM5 - 7.58% + 2.31% - 12.91%
ARZZ3 + 5.41% + 6.63% - 18.49%
CRFB3 - 0.18% + 3.47% - 23.46%
SOMA3 + 15.24% + 18.42% - 24.23%
RECV3 + 4.24% + 1.82% - 29.07%
VAMO3 + 2.17% + 2.95% - 35.81%
PETZ3 + 0.97% + 7.20% - 38.04%
PCAR3 + 21.79% + 5.97% - 46.25%


Volume Ratio
1.2
1.1
1.0
0.9
15
16
17
18
19
20
21
22
23
24
Date

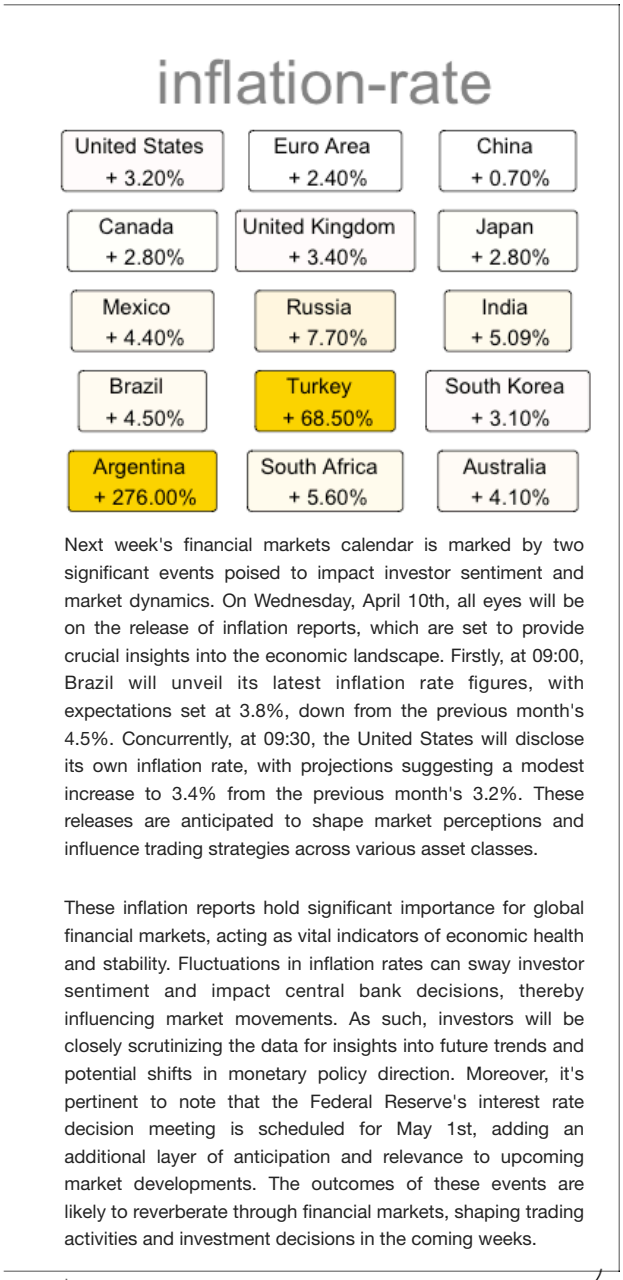

HAPV3
FLRY3
HYPE3
COGN3
RDOR3
RADL3
YDUQ3
RAIZ4
UGPA3
SMTO3
RECV3
PETR3
CSAN3
RRRP3
DIRR3
EZTC3
IGTI11
MULT3
MRVE3
CYRE3
SOMA3
ASAI3
CRFB3
PCAR3
MDIA3
SLCE3
MRFG3
GMAT3
ABEV3
BRFS3
JBSS3
BEEF3
PRIO3
VBBR3
PETR4
ENEV3
ENGI11
ELET6
CPFE3
CMIG4
TRPL4
ELET3
CPLE6
EQTL3
MGLU3
GOLL4
CVCB3
AZUL4
PETZ3
VIVT3
LWSA3
TIMS3
TOTS3
BPAC11
BBSE3
IRBR3
B3SA3
CIEL3
DXCO3
KLBN11
WEGE3
SUZB3
CCRO3
VAMO3
RAIL3
RENT3
VIVA3
ARZZ3
NTCO3
LREN3
ITUB4
SANB11
BBDC3
ITSA4
BBDC4
BBAS3
USIM5
CMIN3
BRAP4
GOAU4
BRKM5
CSNA3
VALE3
GGBR4


Previous days and weeks
now
NTCO3
TOTS3
TIMS3
KLBN11
EMBR3
CYRE3
SOMA3
SUZB3
WEGE3
LREN3
AZUL4
MRVE3
MULT3
GGBR4
CSNA3
RAIL3
CCRO3
BHIA3
ELET3
SBSP3
MGLU3
VALE3
RENT3
EQTL3
CPLE6
B3SA3
BPAC11
BBSE3
ASAI3
JBSS3
VBBR3
CSAN3
RRRP3
CIEL3
BEEF3
UGPA3
ABEV3
BRFS3
PETR3
PRIO3
BBDC4
BBAS3
HAPV3
HYPE3
RDOR3
PETR4
ITUB4
ITSA4
RADL3

This is also ggplot2 package using calendar data



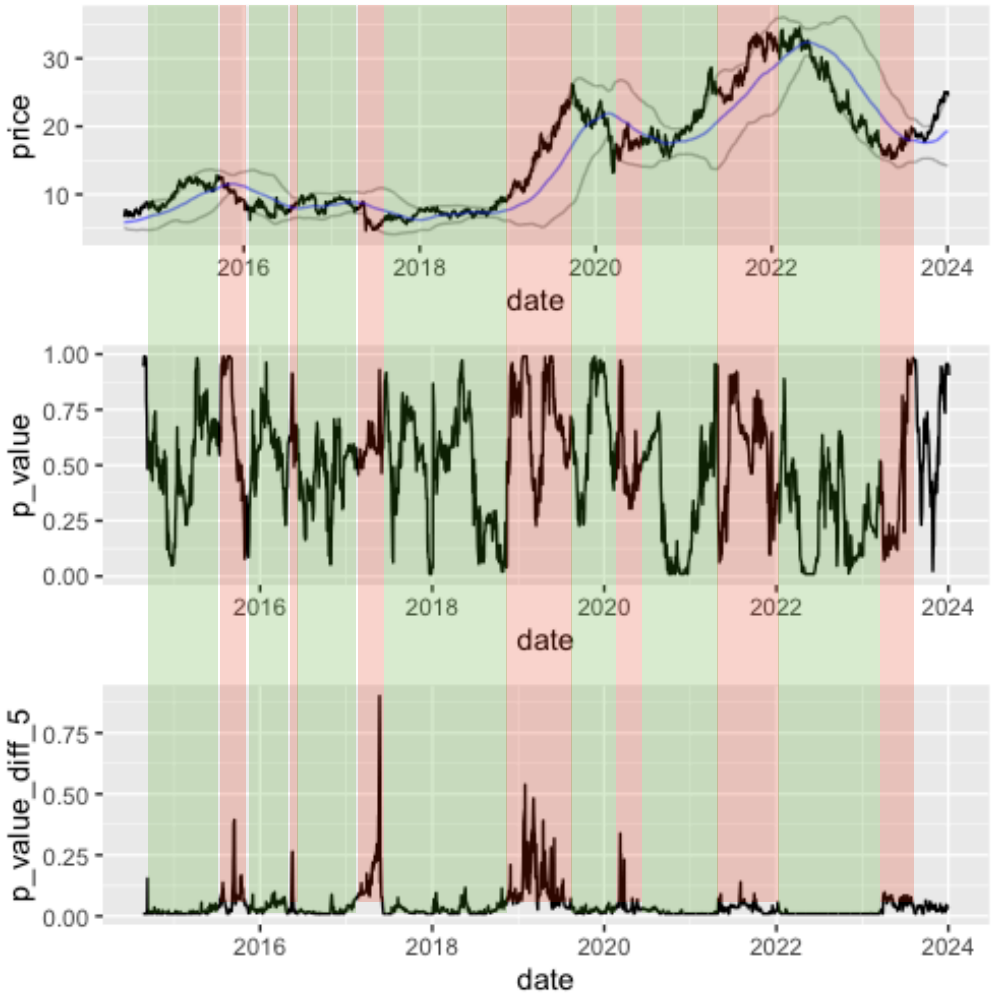
This was plotted using ggplot2 package using data from trading economics



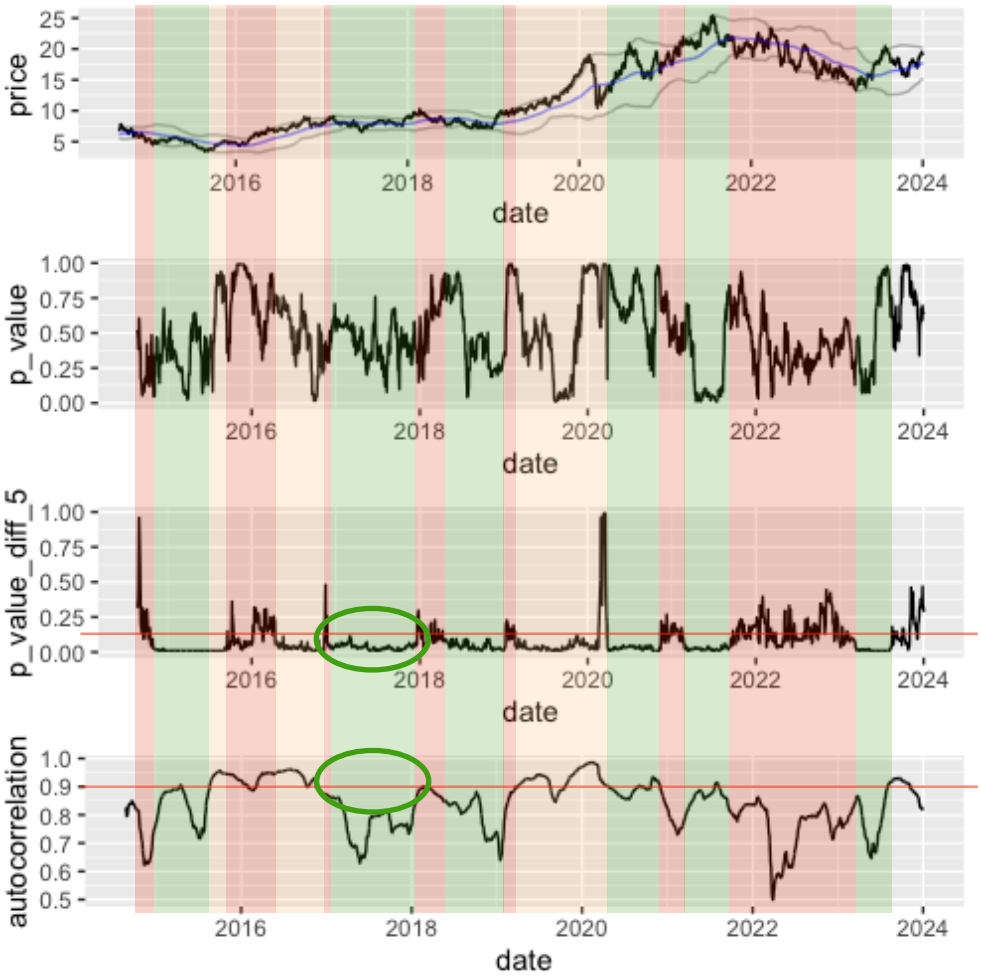
Next week's financial markets calendar is marked by two significant events poised to impact investor sentiment and market dynamics. On Wednesday, April 10th, all eyes will be on the release of inflation reports, which are set to provide crucial insights into the economic landscape. Firstly, at 09:00, Brazil will unveil its latest inflation rate figures, with expectations set at 3.8%, down from the previous month's 4.5%. Concurrently, at 09:30, the United States will disclose its own inflation rate, with projections suggesting a modest increase to 3.4% from the previous month's 3.2%. These releases are anticipated to shape market perceptions and influence trading strategies across various asset classes.

These inflation reports hold significant importance for global financial markets, acting as vital indicators of economic health and stability. Fluctuations in inflation rates can sway investor sentiment and impact central bank decisions, thereby influencing market movements. As such, investors will be closely scrutinizing the data for insights into future trends and potential shifts in monetary policy direction. Moreover, it's pertinent to note that the Federal Reserve's interest rate decision meeting is scheduled for May 1st, adding an additional layer of anticipation and relevance to upcoming market developments. The outcomes of these events are likely to reverberate through financial markets, shaping trading activities and investment decisions in the coming weeks.

This was produced by GPT using this prompt https://chat.openai.com/share/d3a7b575-e4a4-4ef9-9d31-e000448a239f


price
30
20
10
2016
2018
2020
2022
2024
date
p_value
1.00
0.75
0.50
0.25
0.00
2016
2018
2020
2022
2024
date
p_value_diff_5
0.75
0.50
0.25
0.00
2016
2018
2020
2022
2024
date


Fees BTC
Fees USD
4 BTC
3 BTC
2 BTC
1 BTC
0 BTC
$150k
$120k
$90k
$60k
$30k
$0.0
Mar
May
Jul
Sep
Nov
2024
BTC
Bitcoin / U.S. Dollar · 1W · BITFINEX
O52075.0 H52880.0 L50584.0 C52216.0 +146.0 (+0.28%)
USD
52216.0
5d 1h
TradingView
Mon 25 Mar


price
25
20
15
10
5
2016
2018
2020
2022
2024
date
p_value
1.00
0.75
0.50
0.25
0.00
2016
2018
2020
2022
2024
date
p_value_diff_5
1.00
0.75
0.50
0.25
0.00
2016
2018
2020
2022
2024
date
autocorrelation
1.0
0.9
0.8
0.7
0.6
0.5
2016
2018
2020
2022
2024
date

Goal of the Project:

The primary goal of this project is to leverage financial markets analysis to empower an expert advisor trading system. The overarching objective is to enhance decision-making processes by providing comprehensive information and knowledge. Through the fusion of various data sources, including stock prices, trading volumes, calendar events, earnings, and the latest news, the project aims to create a robust foundation for an automated trading system. The theoretical underpinning of cointegration econometric theory forms the basis for the statistical analysis, with the ultimate aim of optimizing trading strategies.

Implementation Details:

The project's implementation is a multi-faceted process that begins with the collection of diverse data sets. Using R language, the statistical analysis is conducted to identify cointegration relationships, informing the trading system's strategies. The automated trading system, coded in C++, is seamlessly integrated with an exchange, allowing real-time execution of trades. The Shiny package is employed to create interactive dashboards that consolidate the abundance of information generated. These dashboards serve as a user-friendly interface, enabling stakeholders to monitor and analyze the system's performance. Additionally, the system has the capability to generate comprehensive reports in PDF format, facilitating in-depth analysis and documentation.

Summary of Success:

After several months of continuous operation, the project has achieved notable success in automating the trading process and delivering meaningful insights. The expert advisor trading system, informed by cointegration econometric theory, has demonstrated its effectiveness in adapting to dynamic market conditions. The seamless integration with the exchange, coupled with the real-time data analysis, has led to informed and timely decision-making. The Shiny dashboards have proven instrumental in providing a holistic view of the system's performance, while the exportable PDF reports offer a tangible means of documentation and communication. Overall, the project's success is evident in its ability to consistently deliver automated, data-driven trading strategies and comprehensive analytical tools to stakeholders.

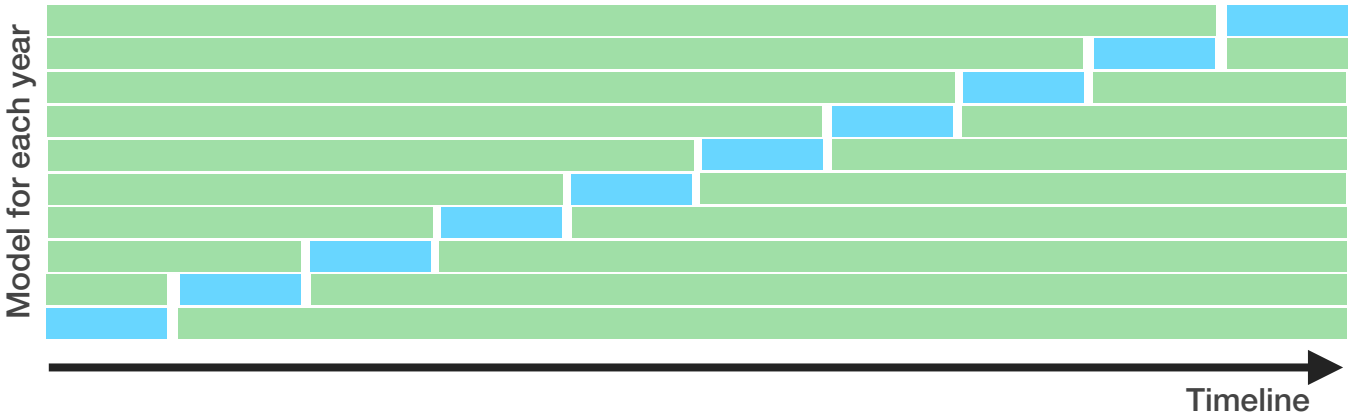
Same sector combinations

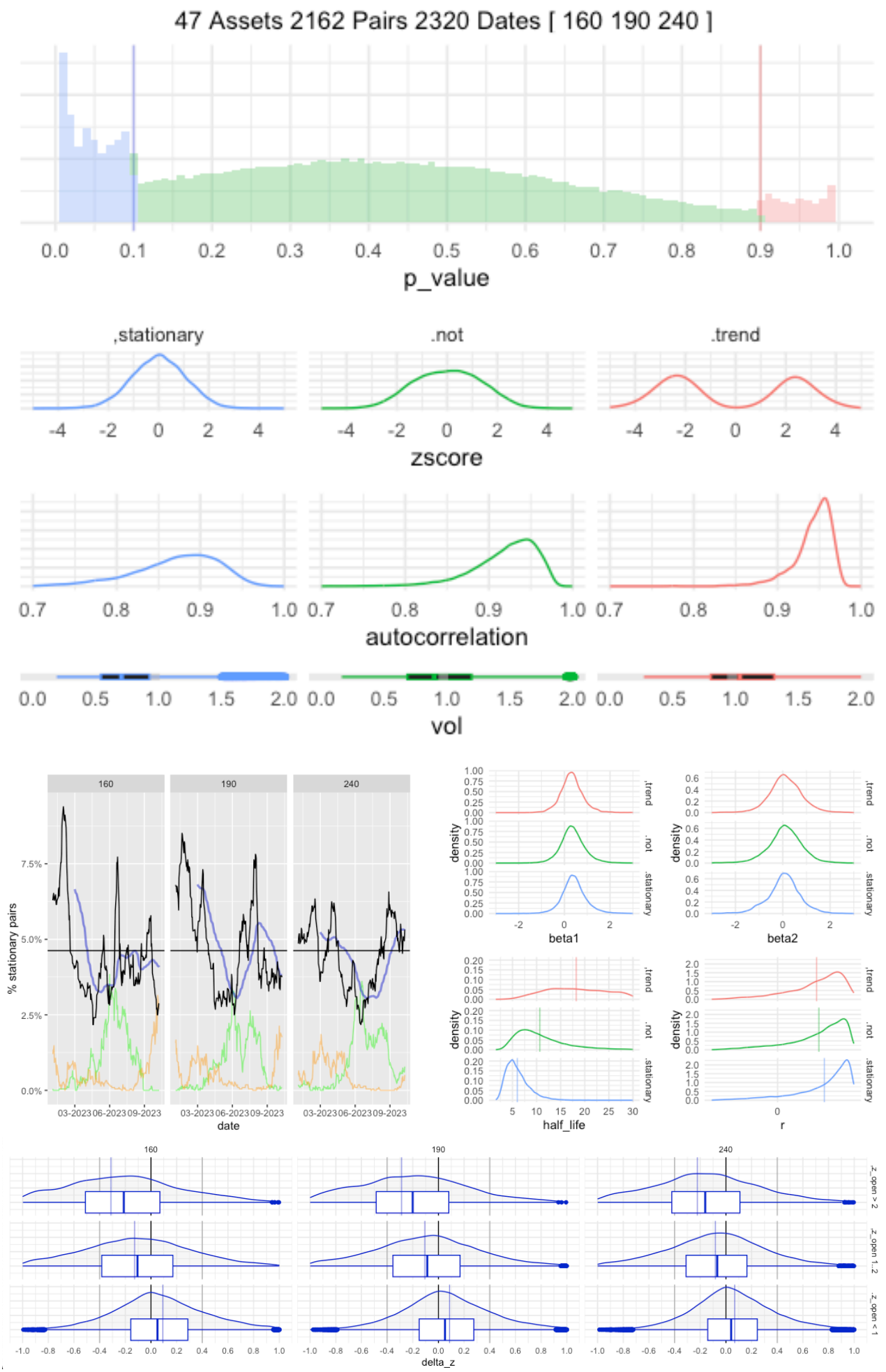
Different sectors

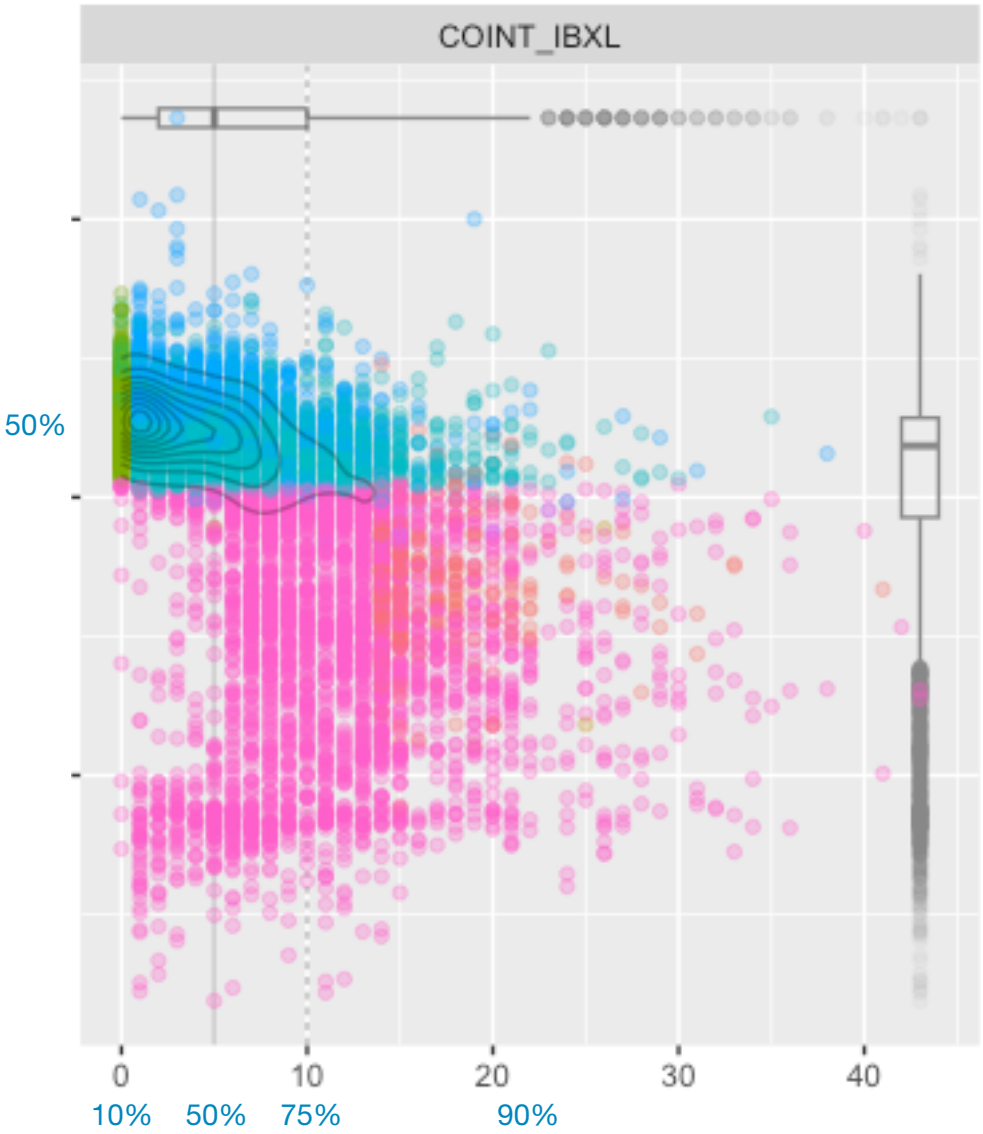
Whole backtest data over time

IBXL x IBXL

Others pairs combinations




47 Assets 2162 Pairs 2320 Dates [160 190 240]
p_value
,stationary
.not
.trend
zscore
autocorrelation
vol
160
190
240
% stationary pairs
date
density
beta1
beta2
half_life
r
delta_z
.z_open > 2
.z_open 1..2
.z_open < 1


COINT_IBXL
50%
0
10
20
30
40
10%
50%
75%
90%

GDP

Type to enter text

TODAY'S NEWS

Main Heading

To get started, just tap or click this placeholder text and begin typing. You can view and edit this newsletter on your Mac, iPad, iPhone, or on iCloud.com.

Use paragraph styles to give your newsletter a consistent look. For example, this paragraph uses Body style. You can change it in the Text tab of the Format controls.

This newsletter template uses linked text boxes, so the text you type can flow from one text box to the next. You can identify which text boxes are linked by the shared color of the circle at the top of each box when it's selected. Numbers in the circles indicate the order of the text boxes. You

Drag your own photos onto any image placeholders in this template, then crop or resize them if you wish.

can add additional text boxes, reorder how text flows, remove a box, and more.

To add more photos, image galleries, audio clips, videos, charts, or any of more than 700 customizable shapes, tap or click one of the insert buttons in the toolbar or drag and drop the objects onto the page.

You can use Pages for both word processing and page layout. This newsletter template is set up for page layout, so you can manually rearrange pages and freely position text boxes, images, and other objects on the page.

You can layer objects, resize them, and place them anywhere on the page. To change how an object moves with text, select the object and then tap or click the Arrange tab in the Format controls.

“This is an example of a pull quote (a key phrase from your newsletter). Tap or click this text to add your own.”

-SOURCE

In word processing documents, your text flows from one page to the next as you type, with new pages created automatically when you reach the end of a page. To create a word processing document, choose a word processing template in the template chooser. You can also change this document to word processing on your Mac, iPad, or iPhone by turning on Document Body in the Document controls.