ASPIRE
KNOWLEDGE & SKILLS

Unit 3

Statistical Tools and Usage

AI - Business Intelligence Analyst

Table of Content

- ✔*Discover the basics of using statistical software packages and IDEs such as RStudio, Jupyter Notebooks*
- ✔*Apply basic functions and libraries present in statistical software packages and IDEs*
- ✔*Use statistical packages, frameworks and libraries such as NumPy and Pandas for developing applications*

Individuals at this job are responsible for performing different aspects of Business Analysis. S/he will be responsible for importing and preprocessing data and perform exploratory analysis to derive actionable insights. A BI analyst needs to have strong analytical skills and problem solving ability. S/he needs to have good communication skills to work with stakeholders across multiple teams such as marketing, sales, product development, etc.

Introduction to Statistical Software Packages - "Empowering Analysis with Tools"

Subtitle: Navigating the Landscape of Statistical Software

Importance of Statistical Software:

- Foundational Tool: Statistical software is integral to analyzing and interpreting data in the field of Business Intelligence and AI.
- Efficiency: Streamlines complex statistical calculations, visualization, and data manipulation tasks.

Popular Statistical Software Packages:

- R: An open-source statistical programming language with a vast ecosystem of packages for data analysis, visualization, and machine learning.
- Python: Widely used for statistical analysis, machine learning, and data visualization, with popular libraries like NumPy, Pandas, and SciPy.
- SPSS (Statistical Package for the Social Sciences): A software suite used for statistical analysis, data management, and reporting.
- SAS (Statistical Analysis System): Widely used in various industries for advanced analytics, business intelligence, and data management.

Benefits of Statistical Software:

- Automation: Reduces manual effort by automating repetitive tasks in data analysis.
- Reproducibility: Enables the replication of analyses, ensuring transparency and reproducibility.
- Visualization: Facilitates the creation of meaningful visualizations for effective communication of insights.


Business Intelligence
Improves the decision making Process
Makes it easy to Access and Share information
Enables real time analysis with Quick Navigation
Helps identify waste in the system
Reduces the risk of bottlenecks
Help you know your business

RStudio - "Harnessing the Power of R for Statistical Analysis"

Subtitle: A Deep Dive into RStudio for Data Analysis

Introduction to RStudio:

- Purpose: RStudio is an integrated development environment (IDE) for R, providing a user-friendly interface for statistical analysis, data visualization, and coding.
- Features:
  - ✔ Script Editor: Allows writing and executing R scripts.
  - ✔ Console: Direct interaction with R through a command-line interface.
  - ✔ Environment and History: Displays workspace variables and command history.
  - ✔ Plots and Files: Integrates plotting and file management capabilities.

Key Functionalities:

- Package Management: Simplifies the installation and management of R packages for specialized tasks.
- Code Execution: Executes R code in chunks, making it easy to iterate and test.
- Integrated Help: Provides easy access to documentation and help resources.
- Data Viewer: Facilitates exploration of datasets through a graphical interface.

Workflow in RStudio:

- Data Import: Import datasets from various file formats using built-in functions.
- Data Cleaning and Transformation: Perform data cleaning and transformation using R functions and packages.
- Statistical Analysis: Conduct statistical analyses, hypothesis testing, and regression modeling.
- Visualization: Create visualizations using popular packages like ggplot2.


SiSense
ECommerce › Ecommerce Overview
ADD FILTER
NEW DASHBOARD
HOME
MY DASHBOARDS
Ecommerce Overview
Ecommerce Summary
Marketing Analysis
Marketing Daily
FEMALE REVENUE
333K
MALE REVENUE
2M
GROWTH OF REVENUE
-0.1%
GROWTH OF VISITORS
13.0%
OF BRANDS
1,569
AVERAGE COST
$162
Category Analysis
Revenue By Age
14%
10%
12%
13%
40%
Years to Date
Country
SISENSE

Jupyter Notebooks - "Interactive Data Science in Action"

Subtitle: A Comprehensive Overview of Jupyter Notebooks

Introduction to Jupyter Notebooks:

- Purpose: Jupyter Notebooks are open-source, web-based interactive computational notebooks, supporting multiple programming languages such as Python, R, and Julia.
- Features:
  - Live Code Execution: Code cells allow immediate execution and visualization of results.
  - Markdown Cells: Support for adding formatted text, images, and hyperlinks.
  - Integration: Connects code, visualizations, and narrative in a single document.

Key Functionalities:

- Rich Output: Supports the display of images, plots, interactive widgets, and more.
- Collaboration: Facilitates sharing and collaboration through exportable HTML, PDF, or GitHub integration.
- Interactive Widgets: Enhances interactivity with sliders, buttons, and dropdowns.
- Integration with Data Science Libraries: Seamless integration with libraries like Pandas, NumPy, and SciPy.

Workflow in Jupyter Notebooks:

- Data Exploration: Use Pandas for loading and exploring datasets interactively.
- Visualization: Leverage Matplotlib or Seaborn for creating visualizations within the notebook.
- Machine Learning: Implement machine learning models using scikit-learn or TensorFlow.
- Documentation and Reporting: Combine code with formatted text to create comprehensive reports.

Core Functions in Statistical Software Packages - "Essentials for Data Analysis"

Subtitle: Mastering Fundamental Functions for Statistical Exploration

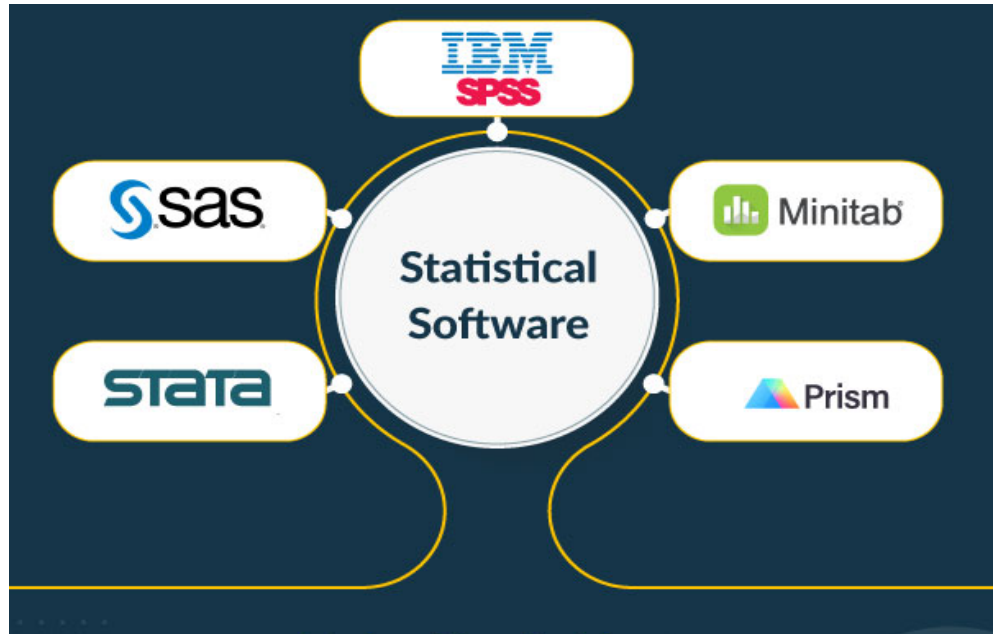
1. Data Import and Inspection:

- Functionality: Importing datasets is a crucial first step.
- R Example:

```
# Read CSV file into a data frame
data <- read.csv("dataset.csv")

# Display the structure of the data
str(data)
```

- Python Example:

```
# Import necessary library
import pandas as pd

# Read CSV file into a DataFrame
data = pd.read_csv("dataset.csv")

# Display basic information about the DataFrame
data.info()
```



2. Descriptive Statistics:

- Functionality: Summarizing and exploring data.
- R Example:

```
# Display summary statistics
summary(data)
```

- Python Example:

```
# Display summary statistics
summary(data)
```

3. Data Visualization:

- Functionality: Creating visual representations of data.
- R Example:

```
# Create a scatterplot
plot(data$X, data$Y, main="Scatterplot", xlab="X-axis", ylab="Y-axis")
```

- Python Example:

```
# Import necessary libraries
import matplotlib.pyplot as plt

# Create a scatterplot
plt.scatter(data['X'], data['Y'])
plt.title('Scatterplot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

Essential Libraries in RStudio - "Unlocking Advanced Capabilities"

Subtitle: Harnessing the Power of R Packages for Data Analysis

1. dplyr for Data Manipulation:

- Functionality: Efficient data manipulation and transformation.
- R Example:

```
# Filter rows based on a condition
filtered_data <- data %>%
  filter(X > 10)
```

2. ggplot2 for Data Visualization:

- Functionality: A versatile package for creating complex and customizable visualizations.
- R Example:

```
# Create a scatterplot using ggplot2
ggplot(data, aes(x=X, y=Y)) +
  geom_point() +
  labs(title="Scatterplot", x="X-axis", y="Y-axis")
```

3. caret for Machine Learning:

- Functionality: Streamlining the process of building and evaluating machine learning models.
- R Example:

```
# Train a random forest model
library(caret)
model <- train(Y ~ ., data=data, method="rf")
```

Key Libraries in Jupyter Notebooks - "Enhancing Analytical Capabilities"

Subtitle: Leveraging Python Libraries for Data Science

1. NumPy for Numerical Computing:

- Functionality: Efficient handling of arrays and mathematical operations.
- python Example:

```
# Import NumPy
import numpy as np

# Create a NumPy array
np_array = np.array([1, 2, 3, 4, 5])

# Perform a mathematical operation
result = np.mean(np_array)
```

2. Pandas for Data Manipulation:

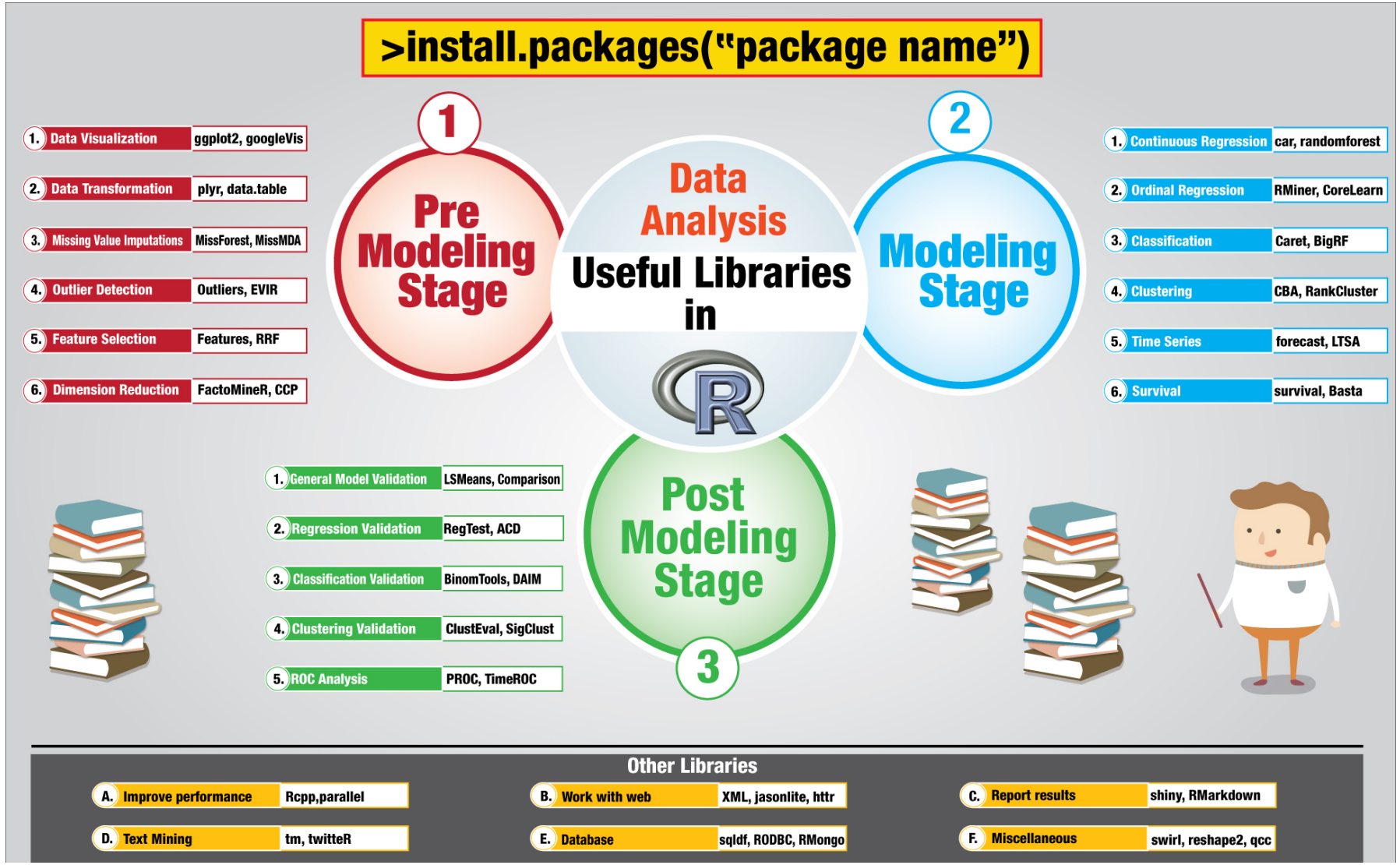
- Functionality: Provides data structures like DataFrames for easy manipulation.
- python Example:

```
# Import Pandas
import pandas as pd

# Create a DataFrame
df = pd.DataFrame({'X': [1, 2, 3], 'Y': [4, 5, 6]})

# Filter rows based on a condition
filtered_df = df[df['X'] > 1]
```

3. Matplotlib and Seaborn for Data Visualization:

- Functionality: Versatile libraries for creating static and interactive visualizations.
- python Example:

```
# Import Matplotlib and Seaborn
import matplotlib.pyplot as plt
import seaborn as sns

# Create a scatterplot using Seaborn
sns.scatterplot(x='X', y='Y', data=df)
plt.title('Scatterplot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```


>install.packages("package name")
Data Analysis
Useful Libraries in
R
1
Pre Modeling Stage
1. Data Visualization ggplot2, googleVis
2. Data Transformation plyr, data.table
3. Missing Value Imputations MissForest, MissMDA
4. Outlier Detection Outliers, EVIR
5. Feature Selection Features, RRF
6. Dimension Reduction FactoMineR, CCP
2
Modeling Stage
1. Continuous Regression car, randomforest
2. Ordinal Regression RMiner, CoreLearn
3. Classification Caret, BigRF
4. Clustering CBA, RankCluster
5. Time Series forecast, LTSA
6. Survival survival, Basta
3
Post Modeling Stage
1. General Model Validation LSMeans, Comparison
2. Regression Validation RegTest, ACD
3. Classification Validation BinomTools, DAIM
4. Clustering Validation ClustEval, SigClust
5. ROC Analysis PROC, TimeROC
Other Libraries
A. Improve performance Rcpp,parallel
B. Work with web XML, jasonlite, httr
C. Report results shiny, RMarkdown
D. Text Mining tm, twitteR
E. Database sqldf, RODBC, RMongo
F. Miscellaneous swirl, reshape2, qcc

Overview of Statistical Packages and Libraries - "Empowering AI Applications"

Subtitle: Unleashing the Power of Statistical Frameworks for Development

Introduction to Statistical Packages:

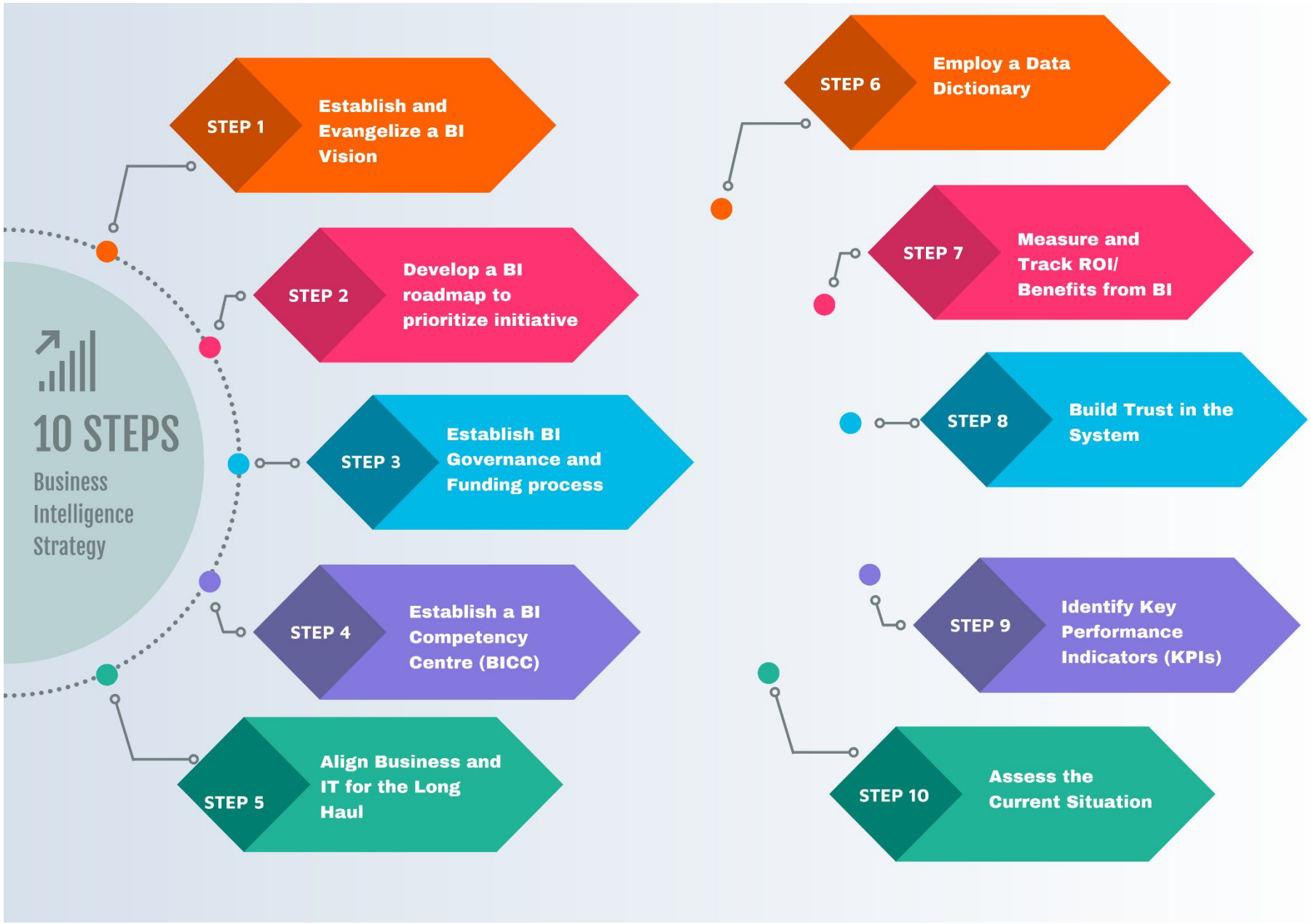
- Integral Components: Statistical packages play a crucial role in developing applications with advanced analytical capabilities.
- Foundation for AI: Essential for implementing statistical models, machine learning algorithms, and data manipulation.

Key Statistical Packages and Libraries:

- NumPy (Numerical Python): A powerful library for numerical computing, providing support for large, multi-dimensional arrays and matrices, along with mathematical functions.
- Pandas: A data manipulation library built on top of NumPy, offering data structures like DataFrame for efficient data analysis and manipulation.

Versatility of Use:

- NumPy: Facilitates numerical operations, element-wise operations, and advanced mathematical functions. Used in tasks such as linear algebra, Fourier analysis, and random number generation.
- Pandas: Streamlines data manipulation tasks, enabling data cleaning, filtering, grouping, and merging. Essential for data preparation in AI applications.


10 STEPS
Business Intelligence Strategy
STEP 1
Establish and Evangelize a BI Vision
STEP 2
Develop a BI roadmap to prioritize initiative
STEP 3
Establish BI Governance and Funding process
STEP 4
Establish a BI Competency Centre (BICC)
STEP 5
Align Business and IT for the Long Haul
STEP 6
Employ a Data Dictionary
STEP 7
Measure and Track ROI/ Benefits from BI
STEP 8
Build Trust in the System
STEP 9
Identify Key Performance Indicators (KPIs)
STEP 10
Assess the Current Situation

NumPy - "Building the Foundation for Numerical Computing"

Subtitle: Leveraging NumPy for Efficient Numerical Operations

1. Array Creation and Manipulation:

- NumPy Arrays: Efficient representation of numerical data, supporting operations on entire arrays without the need for explicit loops.
- python Example:

```
# Import NumPy
import numpy as np

# Create a NumPy array
my_array = np.array([1, 2, 3, 4, 5])

# Perform element-wise operations
squared_array = my_array ** 2
```

2. Linear Algebra Operations:

- Matrix Multiplication: NumPy provides functions for matrix operations, facilitating linear algebra tasks.
- python Example:

```
# Perform matrix multiplication
matrix_A = np.array([[1, 2], [3, 4]])
matrix_B = np.array([[5, 6], [7, 8]])
result_matrix = np.dot(matrix_A, matrix_B)
```

3. Random Number Generation:

- NumPy Random Module: Generates random numbers for simulations, experiments, and statistical analysis.
- python Example:

```
# Generate random numbers from a normal distribution
random_numbers = np.random.normal(size=1000)
```

Pandas - "Efficient Data Manipulation for AI Applications"

Subtitle: Streamlining Data Analysis and Manipulation with Pandas

1. Introduction to Pandas DataFrames:

- DataFrame Structure: A two-dimensional, tabular data structure with labeled axes (rows and columns).
- python Example:

```
# Import Pandas
import pandas as pd

# Create a DataFrame from a dictionary
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 22],
        'Salary': [50000, 60000, 45000]}

df = pd.DataFrame(data)
```

2. Data Manipulation with Pandas:

- Filtering and Selection: Easily filter rows and select specific columns based on conditions.
- python Example:

```
# Filter rows based on age
young_employees = df[df['Age'] < 30]

# Select specific columns
selected_data = df[['Name', 'Salary']]
```

3. Grouping and Aggregation:

- GroupBy Operations: Group data based on certain criteria and perform aggregation functions.
- python Example:

```
# Group by age and calculate average salary
avg_salary_by_age = df.groupby('Age')['Salary'].mean()
```

Video

Statistical Packages

https://www.youtube.com/watch?v=hHywVkLwLzg

Selection of Best Statistical Software: Based On Your Data and Need

https://www.youtube.com/watch?v=Gbte97Z5ot8

Using Jupyter with RStudio Server Pro

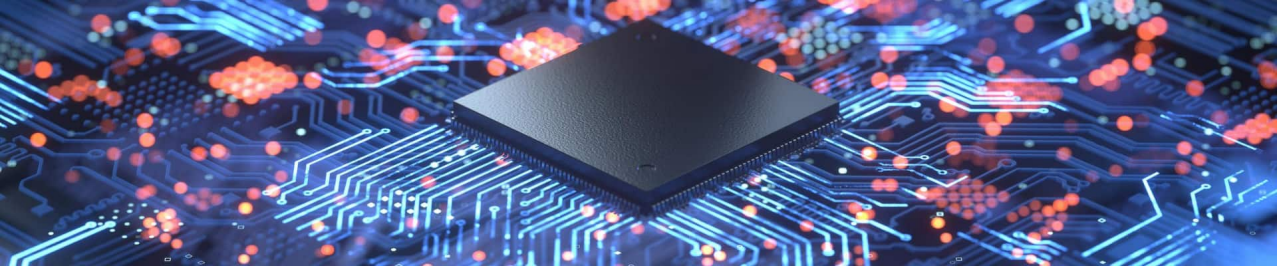
https://www.youtube.com/watch?v=30oPT43Y4-I

IDES - Jupyter, Spyder, PyCharm

https://www.youtube.com/watch?v=WeAlV2_EhBI

NumPy and Pandas Tutorial

https://www.youtube.com/watch?v=FniLzpaSFGk



Thank You

Aspire Knowledge & Skills India Pvt Ltd.

1204, J.M Road, Kamala Arcade ,
Office No. 301-305, Opp. Bal Gandharva Rang Mandir, Deccan, Pune – 411 004

Phone No: 020-25530291

Website: www.aspireks.com