A

MOOC based Seminar Report

On

Data Analysis with R Programming

Name of website registered: https://www.coursera.org

Submitted in partial fulfillment of the requirement Seminar for the Fifth Semester

BCA

By

Name of the Student: Deepankar Sharma

University Roll No: 2092014(14)

Under the Guidance of: Ms. Richa Pandey

Name of Faculty

Designation

Deptt. of CS&A

DEPARTMENT OF SCHOOL OF COMPUTING

GRAPHIC ERA HILL UNIVERSITY HALDWANI CAMPUS

BAREILLY ROAD, BERIPARAO, HALDWANI

DISTRICT- NAINITAL-263136

2022 - 2023

Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

HALDWANI CAMPUS

THIS IS TO CERTIFY THAT MR. **DEEPANKAR SHARMA** HAS SATISFACTORILY PRESENTED MOOC BASED SEMINAR. THE COURSE OF THE MOOCS REGISTRATION **DATA ANALYSIS WITH R PROGRAMMING** IN PARTIAL FULLFILLMENT OF THE SEMINAR PRESENTATION REQUIREMENT IN **SIXTH** SEMESTER OF **BCA** DEGREE COURSE PRESCRIBED BY GRAPHIC ERA HILL UNIVERSITY, HALDWANI CAMPUS DURING THE YEAR **2022- 2023**.

Campus MOOC-Coordinator

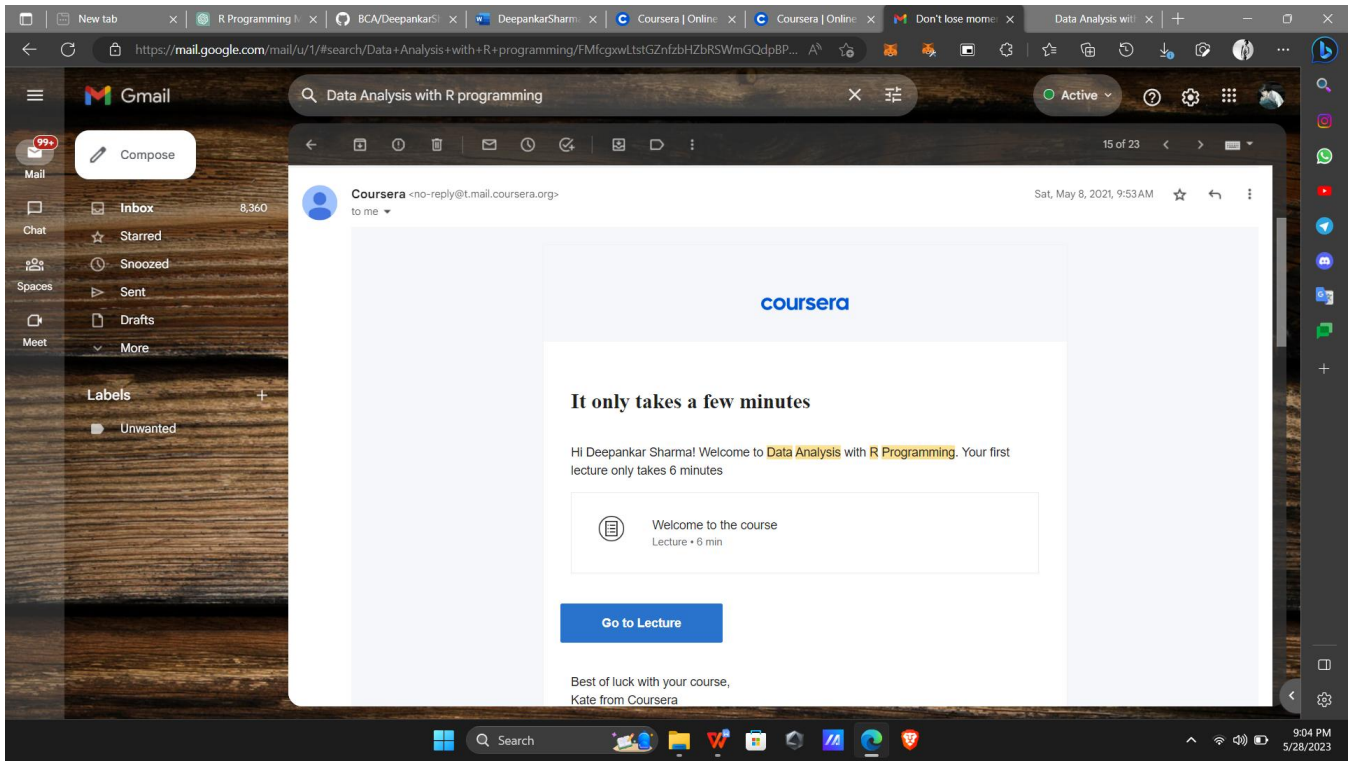
Name

Signature

Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

HALDWANI CAMPUS

Copy of confirmation Email of registration Received

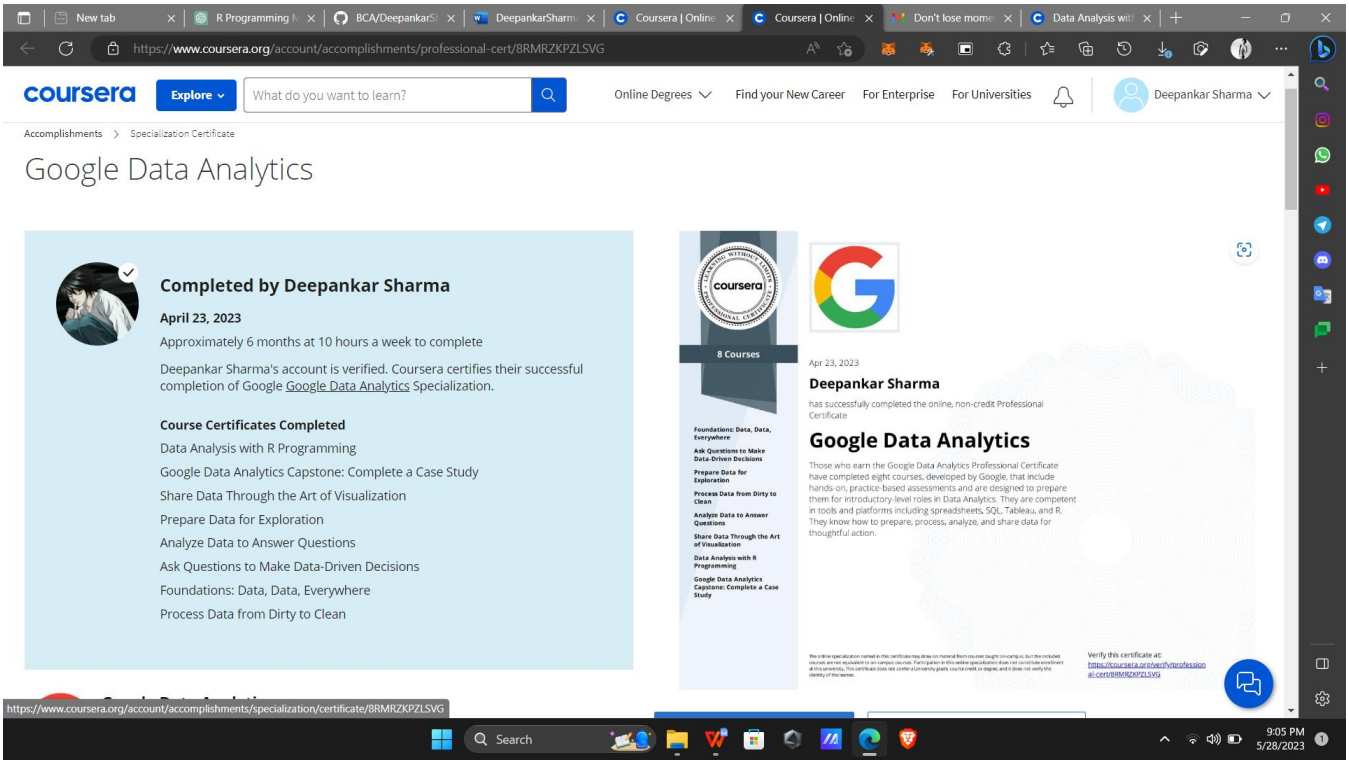


Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

HALDWANI CAMPUS

Copy of Course Completion Certificate Received



Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

HALDWANI CAMPUS

Modules Attended

Course Structure and Content

The course was structured into several modules, covering various topics essential to data analytics with R programming. The modules included:

1. Introduction to R Programming: This module provided an overview of R programming, its advantages for data analytics, and step-by-step instructions for installing and setting up R programming.

2. Data Wrangling with R: This module focused on the essential skills required for data manipulation and cleaning. Learners were introduced to various data types in R programming and learned how to perform data wrangling tasks using functions and packages.

3. Data Visualization with R: In this module, learners explored the importance of data visualization and discovered different types of plots available in R programming. They learned how to create visually appealing and informative data visualizations using R libraries and packages.

4. Introduction to Statistics with R: This module introduced learners to statistical concepts and their relevance in data analytics. Learners gained an understanding of descriptive and inferential statistics and learned how to perform statistical analysis using R programming.

5. Machine Learning with R: The module delved into the exciting field of machine learning. Learners explored different machine learning algorithms and techniques available in R programming. They learned how to apply these techniques to build predictive models and make data-driven decisions.

6. Real-World Data Analytics Projects: This module provided an opportunity for learners to apply their skills and knowledge to real-world data analytics projects. They gained hands-on experience by working on projects such as customer churn prediction, sentiment analysis, and fraud detection.

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my teacher As. Prof. Richa Pandey.

For their exemplary guidance, monitoring and constant encouragement throughout the course of this report. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark.

Lastly, I thank almighty, my parents and friends for their constant encouragement without which this project would not be possible.

Name of Student

Deepankar Sharma

Email ID

DEEPANKARSHARMA.20041299@gehu.ac.in

LEARNING OUTCOME:

Course Overview
The "Data Analytics with R Programming" MOOCs course provided a comprehensive introduction to using R programming for data analytics. The course aimed to equip learners with the necessary skills and knowledge to manipulate, visualize, analyze, and make predictions from data using the R programming language. The course was designed for individuals with a basic understanding of programming concepts and an interest in data analytics.

Upon completing the "Data Analytics with R Programming" MOOCs course, learners achieved the following learning outcomes:

1. Proficiency in R Programming: Learners gained a solid understanding of R programming language, its syntax, and its capabilities for data analytics.

2. Data Wrangling Skills: Learners acquired the skills to manipulate, clean, and transform raw data into a suitable format for analysis using various functions and packages in R programming.

3. Data Visualization Abilities: Learners developed the ability to create meaningful and visually appealing data visualizations using R programming, enabling them to effectively communicate insights from data.

4. Statistical Analysis Competence: Learners acquired knowledge of statistical concepts and techniques, enabling them to perform descriptive and inferential statistics using R programming.

5. Machine Learning Proficiency: Learners gained exposure to machine learning algorithms and techniques, and developed the ability to apply them in R programming for predictive modeling and decision-making tasks.

6. Practical Experience: Through real-world data analytics projects, learners gained hands-on experience in solving data-related problems using R programming, enhancing their practical skills and building a portfolio of projects.

Conclusion
The "Data Analytics with R Programming" MOOCs course provided a comprehensive foundation in using R programming for data analytics. Learners were equipped with the essential skills to manipulate data, create visualizations, perform statistical analysis, and apply machine learning techniques. The practical experience gained through real-world projects added value to the course, allowing learners to apply their knowledge in practical scenarios. The course successfully fulfilled its objectives of providing learners with a strong foundation in data analytics with R programming and empowering them to embark on data-driven projects.

Overall, the "Data Analytics with R Programming" MOOCs course was a valuable and enriching learning experience, fostering the development of essential skills in data analytics using R programming. It served as an excellent starting point for individuals interested in pursuing a career or further studies in the field of data analytics.

Introduction to R Programming

- What is R programming?
R programming is an open-source programming language widely used for statistical computing, data analysis, and graphical visualization. It provides a comprehensive suite of tools and libraries specifically designed for handling and manipulating data. R is known for its flexibility, extensibility, and powerful statistical capabilities.

- Why use R programming for data analytics?
R programming offers numerous advantages for data analytics tasks:
1. Rich ecosystem: R has a vast collection of packages and libraries dedicated to data manipulation, statistical analysis, and visualization, making it a versatile tool for data analytics.
2. Statistical capabilities: R provides a wide range of built-in statistical functions and packages, allowing users to perform complex analyses and generate meaningful insights.
3. Data visualization: R's graphics packages, such as ggplot2 and lattice, enable the creation of high-quality visualizations that aid in data exploration and communication of findings.
4. Reproducibility: R promotes reproducibility through its script-based nature, allowing users to document and share their analyses in a transparent manner.
5. Community support: R benefits from an active and vibrant community of users, which ensures continuous development, frequent updates, and abundant resources for learning and troubleshooting.

- How to install and set up R programming?
To start using R programming, follow these steps:
1. Visit the official R website (www.r-project.org) and download the latest version of R compatible with your operating system.
2. Run the installer and follow the prompts to install R on your computer.
3. Once installed, you can launch R either through the R GUI or RStudio, a popular integrated development environment (IDE) for R programming.
4. Familiarize yourself with the R environment, including the R console, where you can type and execute R commands, and the script editor, where you can write and save your R code.

With R programming installed and set up, you are now ready to dive into the world of data analytics and explore the various capabilities and features that R has to offer.

Data Wrangling with R

- What is data wrangling?
Data wrangling, also known as data munging, refers to the process of cleaning, transforming, and preparing raw data for further analysis. It involves tasks such as removing inconsistencies, handling missing values, reformatting data, and merging multiple datasets.

- Why is data wrangling important?
Data in its raw form often contains errors, inconsistencies, and missing values that can hinder analysis. Data wrangling is crucial because:
1. It ensures data quality: By cleaning and organizing the data, we can identify and correct errors or inconsistencies, ensuring that subsequent analysis is based on reliable and accurate data.
2. It enables data integration: Many analysis tasks require merging or combining multiple datasets. Data wrangling allows us to bring different sources of data together and create a unified dataset for analysis.

3. It facilitates data analysis: By transforming the data into a suitable format, data wrangling makes it easier to apply statistical techniques, create visualizations, and extract insights from the data.

- How to manipulate data using R programming?
R provides a wide range of functions, libraries, and packages for data manipulation. Here are some common data wrangling operations in R:

1. Subsetting: Selecting specific rows or columns from a dataset based on certain conditions using functions like `subset()`, `filter()`, or indexing.

2. Handling missing values: Identifying and dealing with missing data using functions like `is.na()`, `complete.cases()`, and applying techniques such as imputation or deletion.

3. Data transformation: Applying transformations to variables, such as scaling, log transformation, or creating new variables using functions like `mutate()` or `transform()`.

4. Aggregation and summarization: Summarizing data by calculating measures like mean, median, or count using functions like `aggregate()`, `summarize()`, or `group_by()`.

5. Joining and merging datasets: Combining datasets based on common keys or variables using functions like `merge()`, `join()`, or `bind_cols()`.

6. Reshaping data: Converting data from a wide format to a long format or vice versa using functions like `melt()` and `pivot_wider()`.

By mastering these techniques and using appropriate R packages such as dplyr, tidyr, or data.table, you can efficiently manipulate and transform your data for further analysis and visualization.

Data wrangling is a critical step in the data analytics process, ensuring that the data is clean, well-structured, and ready for exploration and analysis.

Data Visualization with R

- Why is data visualization important?
Data visualization is a powerful tool for understanding and communicating insights from data. It helps in:
1. Exploratory data analysis: Visualizations allow us to explore patterns, trends, and relationships within the data, enabling us to identify key insights and generate hypotheses.
2. Communication of findings: Visual representations of data make it easier to convey complex information and findings to others in a clear and understandable manner.
3. Decision-making: Visualizations provide a visual context that aids in decision-making by presenting data-driven evidence and supporting logical reasoning.

- Different types of plots in R programming:
R programming offers a wide range of plot types to visualize data effectively. Some commonly used plots include:
1. Scatter plots: Used to visualize the relationship between two continuous variables.
2. Line charts: Ideal for showing trends and changes over time or a continuous variable.
3. Bar charts: Used to compare categorical data or display counts or frequencies.
4. Histograms: Display the distribution of a continuous variable by dividing it into bins.

5. Box plots: Show the distribution of numerical data and identify outliers and quartiles.
6. Heatmaps: Represent data using color intensity to show patterns and correlations.

- How to create visualizations using R programming?
R programming provides a variety of libraries and packages for creating visualizations. Some popular ones include:
1. ggplot2: A powerful and flexible package for creating visually appealing and customizable graphics with a grammar of graphics approach.
2. plotly: Enables interactive and dynamic visualizations that can be easily shared and embedded in web applications.
3. lattice: Offers a comprehensive set of functions for creating conditioned plots, such as trellis plots and parallel coordinate plots.
4. ggvis: Provides interactive visualizations with built-in interactivity and linked views.

By utilizing the capabilities of these packages, you can create a wide range of static and interactive visualizations, customize their appearance, and incorporate additional elements like titles, labels, and annotations to enhance the clarity and aesthetics of your visual representations.

Introduction to Statistics with R

- What is statistics?
Statistics is a branch of mathematics that involves the collection, analysis, interpretation, presentation, and organization of data. It provides methods and techniques for summarizing and drawing conclusions from data, making inferences, and making informed decisions based on evidence.

- Why use statistics for data analytics?
Statistics plays a crucial role in data analytics for several reasons:
1. Descriptive analysis: Statistics allows us to summarize and describe the main features of a dataset, such as measures of central tendency, variability, and distributions.
2. Inferential analysis: By using statistical inference techniques, we can make predictions, estimate parameters, and test hypotheses about populations based on sample data.
3. Relationship identification: Statistics helps us identify and quantify relationships between variables, such as correlation and regression analysis, which can reveal insights and patterns in the data.
4. Confidence and significance: Statistical methods provide measures of confidence intervals and p-values, enabling us to assess the reliability and significance of our findings.

- How to perform statistical analysis using R programming?
R programming offers a comprehensive set of built-in functions and packages for statistical analysis. Some commonly used packages include:
1. stats: This package provides a wide range of statistical functions for descriptive analysis, hypothesis testing, linear regression, and more.
2. dplyr: Along with data manipulation capabilities, dplyr offers functions for group-wise operations, filtering, and summarizing data for statistical analysis.
3. tidyr: This package helps in transforming data into a tidy format, making it easier for statistical analysis.
4. ggplot2: With its powerful graphics capabilities, ggplot2 allows for the creation of visually appealing statistical plots and visualizations.

By leveraging these packages and functions, you can perform a variety of statistical analyses, including hypothesis testing, analysis of variance (ANOVA), chi-square tests, t-tests, and regression analysis, among others. Understanding these techniques and their application in R programming is essential for extracting meaningful insights from data.

Machine Learning with R

- What is machine learning?
Machine learning is a field of study that focuses on developing algorithms and models that can learn from data and make predictions or decisions without being explicitly programmed. It involves training models on existing data to identify patterns, make predictions, or classify new data based on the learned patterns.

- Why use machine learning for data analytics?
Machine learning offers several benefits for data analytics:

1. Predictive modeling: Machine learning algorithms can analyze historical data to make predictions about future outcomes or trends, enabling data-driven decision-making.
2. Pattern recognition: Machine learning can identify complex patterns and relationships within large datasets that may not be easily discernible using traditional statistical techniques.
3. Automation and efficiency: Machine learning algorithms can automate repetitive tasks and streamline data analysis processes, saving time and effort.
4. Scalability: Machine learning techniques can handle large and complex datasets, making them suitable for analyzing big data.

- How to perform machine learning using R programming?
R programming provides a wealth of libraries and packages for machine learning. Some widely used packages include:

1. caret: This package provides a unified interface to numerous machine learning algorithms, along with pre-processing techniques and model evaluation functions.
2. mlr: mlr offers a flexible framework for machine learning, providing access to various algorithms, automated hyperparameter tuning, and model evaluation capabilities.
3. randomForest: randomForest is a popular package for implementing the random forest algorithm, which is effective for classification and regression tasks.
4. xgboost: xgboost is a powerful package for gradient boosting, a popular machine learning technique known for its predictive accuracy

.

Using these packages, you can implement a wide range of machine learning algorithms, including decision trees, random forests, support vector machines (SVM), logistic regression, and neural networks. These algorithms can be used for tasks such as classification, regression, clustering, and anomaly detection.

By leveraging the capabilities of R programming and these machine learning packages, you can unlock the potential of data analytics and gain valuable insights from your data.