Data Analysis Using R and RStudio

Module 0 : Introduction to R and RStudio

0.1 What is R and Why Use It for Statistics

- Overview of R as a statistical computing language
- R vs Excel vs SPSS vs Python: strengths and use cases
- Open-source ecosystem and CRAN repository
- Installing R and RStudio on Windows, Mac, and Linux

0.2 RStudio IDE Orientation

- Console, Script Editor, Environment, and Files panes
- Running code interactively vs script-based execution
- Setting and managing the working directory
- Installing and loading packages: install.packages(), library()

0.3 R Syntax and Core Data Structures

- Variables and assignment operators (<-, =)
- Data types: numeric, integer, character, logical, complex
- Vectors: creation, indexing, arithmetic, vector recycling
- Matrices: creation, row/column operations
- Lists: heterogeneous data containers
- Data frames: the primary tabular data structure in R
- Factors: categorical data with levels and labels

0.4 Introduction to R Markdown

- What is R Markdown and why use it
- Creating a basic .Rmd document
- Code chunks, markdown text, and inline R
- Knitting to HTML, PDF, and Word outputs

Learning Outcome : Participants can navigate RStudio, write and run R scripts, understand core data structures, and set up a reproducible analysis environment.

Module 1 : Data Import, Cleaning, and Transformation

Category : Data Wrangling

1.1 Importing Data into R

- Reading CSV files: read.csv(), read_csv() from readr
- Reading Excel files: readxl package, read_excel()
- Reading text and tab-delimited files: read.table()
- Importing data from the clipboard and other sources
- Checking data on import: head(), tail(), str(), dim(), nrow(), ncol()

1.2 Understanding and Fixing Data Types

- Detecting data types: class(), typeof(), is.numeric(), is.character()
- Type conversion: as.numeric(), as.character(), as.factor(), as.Date()
- Working with dates: lubridate package, ymd(), dmy(), date arithmetic
- Factor levels: levels(), relevel(), droplevels()

1.3 Handling Missing Values

- Identifying NA: is.na(), colSums(is.na())
- Removing missing values: na.omit(), complete.cases()
- Imputation strategies: mean imputation, median imputation
- Visualizing missingness: naniar package, vis_miss()

1.4 Data Transformation with dplyr

- Introduction to the tidyverse philosophy
- select(): choosing columns by name or condition
- filter(): subsetting rows based on logical conditions
- mutate(): creating and transforming new columns
- arrange(): sorting rows ascending and descending
- summarise(): computing summary statistics per group
- group_by(): grouping data for aggregation
- The pipe operator %>%: chaining operations cleanly
- rename(), relocate(), distinct(), count()

1.5 Reshaping Data with tidyr

- Wide vs long format: when and why to convert
- pivot_longer(): converting wide data to long format
- pivot_wider(): converting long data to wide format
- separate() and unite(): splitting and combining columns
- Handling nested data with unnest()

1.6 Joining and Merging Datasets

- left_join(), right_join(), inner_join(), full_join()
- semi_join() and anti_join() for filtering
- bind_rows() and bind_cols() for stacking data

Learning Outcome : Participants can import data from multiple sources, clean and correct data types, handle missing values, and reshape and merge datasets using the tidyverse.

Module 2 : Descriptive Statistics

Category: Statistical Foundations

2.1 Measures of Central Tendency

- Mean: arithmetic, weighted; mean(), weighted.mean()
- Median: resistant to outliers; median()
- Mode: identifying the most frequent value in R
- Trimmed mean and its applications

2.2 Measures of Spread and Variability

- Range: max() - min()
- Variance: var() - population vs sample formula
- Standard deviation: sd()
- Interquartile Range (IQR): IQR(), quantile()
- Coefficient of Variation (CV) and its uses

2.3 Measures of Shape

- Skewness: positive, negative, symmetric; skewness() from e1071
- Kurtosis: leptokurtic, platykurtic, mesokurtic
- Interpreting departures from normality

2.4 Frequency Tables and Cross-Tabulation

- table(), prop.table() for frequency and relative frequency
- Cross-tabulation with two categorical variables
- addmargins() and margin percentages

2.5 Summary Statistics with dplyr and base R

- summary() function for quick overview
- Grouped summaries: group_by() + summarise()
- Custom descriptive tables using gt or kableExtra
- The skimr package for rich summary output

Learning Outcome : Participants can compute, interpret, and report descriptive statistics including central tendency, variability, skewness, kurtosis, and grouped summaries.

Module 3 : Data Visualization - Foundations

Category : Visualization

3.1 Grammar of Graphics with ggplot2

- The layered ggplot2 philosophy: data, aesthetics, geoms, scales, themes
- ggplot() + aes() + geom_*() structure
- Mapping vs setting aesthetics: color, size, shape, fill

3.2 Core Plot Types

- Histogram: geom_histogram(), binwidth, boundary choices
- Density plot: geom_density(), comparing distributions
- Bar chart: geom_bar() vs geom_col(), stacked and dodged
- Box plot: geom_boxplot(), outliers, notched box plots
- Scatter plot: geom_point(), overplotting solutions (alpha, jitter)

3.3 Customization and Themes

- Axis labels, titles, subtitles, captions: labs()
- Built-in themes: theme_bw(), theme_classic(), theme_minimal()
- Custom theme elements: theme() function
- Color scales: scale_color_manual(), scale_fill_brewer(), viridis palettes
- Saving plots: ggsave() with resolution control

3.4 Faceting for Multi-Panel Plots

- facet_wrap(): wrapping by one variable
- facet_grid(): crossing two variables
- Controlling scales, spacing, and labels within facets

Learning Outcome : Participants can build, customize, and export professional-quality visualizations using ggplot2 with full control over aesthetics and layout.

Module 4 : Data Visualization - Advanced

Category : Visualization

4.1 Distribution and Composition Plots

- Violin plot: geom_violin() + geom_boxplot() overlay
- Ridgeline plot: ggridges package for comparing many distributions
- Pie and donut charts: when to use and when to avoid
- Waffle and treemap charts for part-to-whole relationships

4.2 Relationship and Pattern Plots

- Bubble plot: size as a third variable with geom_point()
- Lollipop chart as a clean alternative to bar charts
- Slope chart for showing change between two timepoints
- Dumbbell plot for comparing two groups

4.3 Heatmaps

- Heatmaps with geom_tile() in ggplot2
- pheatmap package: hierarchical clustering and dendrograms
- ComplexHeatmap: advanced annotation and splitting
- Choosing color palettes for diverging vs sequential data

4.4 Correlation Plots

- corrplot package: circle, square, and number methods
- ggcorrplot for ggplot2-style correlation matrices
- Pair plots: GGally::ggpairs() for multivariate exploration

4.5 Interactive Plots with Plotly

- Converting ggplot2 to Plotly: ggplotly()
- Building Plotly charts directly: plot_ly()
- Hover tooltips, zoom, pan, and selection tools
- Exporting interactive charts as HTML widgets

Learning Outcome : Participants can produce advanced visualizations for complex datasets including heatmaps, violin plots, correlation matrices, and interactive charts.

Module 5 : Probability and Statistical Distributions

Category : Statistical Theory and Application

5.1 Foundations of Probability

- Sample space, events, probability rules
- Conditional probability and independence

- Bayes' theorem and its applications

5.2 Discrete Distributions

- Binomial distribution: dbinom(), pbinom(), qbinom(), rbinom()
- Poisson distribution: dpois(), ppois(), qpois(), rpois()
- Practical examples: modeling counts and rare events

5.3 Continuous Distributions

- Normal distribution: dnorm(), pnorm(), qnorm(), rnorm()
- Visualizing the standard normal curve in ggplot2
- t-distribution, chi-square distribution, F-distribution
- Uniform and exponential distributions

5.4 Assessing Normality

- Shapiro-Wilk test: shapiro.test() - interpretation and limitations
- Kolmogorov-Smirnov test: ks.test()
- Q-Q plot: qqnorm() and qqline(); ggplot2 version with stat_qq()
- When normality matters and when it does not

5.5 Central Limit Theorem

- Concept and significance of CLT in statistics
- Simulation-based demonstration in R
- Sampling distributions and standard error

Learning Outcome : Participants understand and can work with key probability distributions in R, assess normality, and explain the Central Limit Theorem with simulation.

Module 6 : Hypothesis Testing

Category : Inferential Statistics

6.1 Principles of Hypothesis Testing

- Null and alternative hypotheses: formulation
- Type I error (alpha) and Type II error (beta)
- p-value: definition, common misconceptions, and correct interpretation
- Statistical power and effect size
- One-tailed vs two-tailed tests

6.2 One-Sample Tests

- One-sample t-test: t.test(mu = ...)
- Wilcoxon signed-rank test: wilcox.test() as non-parametric alternative
- One-sample z-test: when to use

6.3 Two-Sample Tests

- Independent two-sample t-test: t.test(var.equal = TRUE / FALSE)
- Welch's t-test: default in R, handles unequal variances
- Paired t-test: t.test(paired = TRUE) for before-after designs
- Mann-Whitney U test: wilcox.test() for non-parametric two-group comparison
- Checking assumptions: F-test for equal variances, var.test()

6.4 Chi-Square Tests

- Chi-square goodness-of-fit test: chisq.test()
- Chi-square test of independence for contingency tables
- Expected frequency requirements and Fisher's exact test alternative
- Visualizing contingency tables with mosaic plots

6.5 Confidence Intervals

- Constructing confidence intervals for means and proportions
- Interpreting 95% and 99% confidence intervals correctly
- Confidence intervals vs p-values in reporting

6.6 Practical Reporting of Results

- APA-style reporting of t-tests and chi-square results
- Effect sizes: Cohen's d, Cramer's V, odds ratio
- rstatix package for tidy hypothesis test outputs

Learning Outcome : Participants can select, perform, and correctly interpret parametric and non-parametric hypothesis tests, and report results with appropriate effect sizes.

Module 7: ANOVA and Post-Hoc Analysis

Category : Inferential Statistics

7.1 One-Way ANOVA

- When to use ANOVA instead of multiple t-tests
- ANOVA logic: partitioning total variance into between and within groups
- Running one-way ANOVA: aov(), summary.aov()
- F-statistic and p-value interpretation

7.2 Checking ANOVA Assumptions

- Normality of residuals: Shapiro-Wilk on residuals
- Homogeneity of variances: Levene's test with car::leveneTest()
- Bartlett's test: bartlett.test()
- What to do when assumptions are violated

7.3 Post-Hoc Multiple Comparison Tests

- Tukey HSD: TukeyHSD() - when all pairwise comparisons are needed
- Bonferroni correction: p.adjust(method = 'bonferroni')
- DMRT (Duncan's Multiple Range Test): agricolae package
- Dunnett's test: comparing all groups to a control
- Compact letter display (CLD) for visualizing group differences

7.4 Two-Way ANOVA

- Main effects and interaction effects
- Interaction plots: visualizing factor interactions
- Balanced vs unbalanced designs and Type I, II, III sums of squares
- Running two-way ANOVA: aov() with interaction term

7.5 Repeated Measures ANOVA

- Within-subjects designs and sphericity assumption
- Mauchly's test: using ez::ezANOVA()
- Greenhouse-Geisser and Huynh-Feldt corrections
- Practical example: pre-post-followup study design

7.6 Non-Parametric Alternatives

- Kruskal-Wallis test: kruskal.test() as ANOVA alternative
- Dunn's test with Bonferroni adjustment: dunn.test package
- Friedman test for repeated measures non-parametric data

Learning Outcome : Participants can run one-way, two-way, and repeated measures ANOVA, verify assumptions, apply the correct post-hoc test, and interpret group differences with letter notation.

Module 8 : Correlation and Regression Analysis

Category : Modeling

8.1 Correlation Analysis

- Pearson correlation: cor(), cor.test() - linear relationships
- Spearman rank correlation: method = 'spearman' for non-normal data
- Kendall's tau: method = 'kendall' for small samples with ties
- Correlation matrix: cor() for multiple variables
- Testing significance of correlations

8.2 Simple Linear Regression

- The regression equation: Y = b0 + b1X + epsilon
- Fitting a model: lm(Y ~ X, data = ...)
- Interpreting coefficients: intercept and slope
- R-squared: proportion of variance explained
- F-test for overall model significance
- Residuals: checking model fit

8.3 Multiple Linear Regression

- Adding predictors: lm(Y ~ X1 + X2 + X3, data = ...)
- Adjusted R-squared: penalty for adding predictors
- Standardized coefficients for variable importance
- Multicollinearity: VIF (Variance Inflation Factor) using car::vif()
- Variable selection: stepwise, AIC-based with step()

8.4 Regression Diagnostics

- Residuals vs Fitted plot: linearity check
- Q-Q plot of residuals: normality check
- Scale-Location plot: homoscedasticity check
- Cook's distance: identifying influential observations
- plot(model) for base R diagnostic panels

8.5 Logistic Regression

- When to use logistic regression: binary outcome variable
- The logit link function and odds ratios
- Fitting the model: glm(Y ~ X, family = binomial)
- Interpreting coefficients as log-odds and odds ratios
- Model evaluation: confusion matrix, ROC curve, AUC

Learning Outcome : Participants can perform correlation analysis, build and diagnose linear regression models, interpret coefficients, handle multicollinearity, and fit logistic regression models for binary outcomes.

Module 9 : Multivariate Analysis

Category : Advanced Statistics

9.1 Principal Component Analysis (PCA)

- When and why to use PCA: dimensionality reduction
- Running PCA: prcomp(), scale. = TRUE importance
- Scree plot: identifying the number of components to retain
- Proportion of variance explained by each component
- Biplot: visualizing samples and variable loadings together
- Loading scores: which variables contribute most to each PC
- PCA with ggplot2 using factoextra package

9.2 Cluster Analysis

- Hierarchical clustering: hclust(), distance matrices, linkage methods
- Dendrogram: visualizing cluster structure
- Cutting the dendrogram: cutree()
- K-means clustering: kmeans(), choosing optimal k with elbow method
- Silhouette analysis for cluster validation
- Combining PCA and clustering for exploratory data analysis

9.3 Discriminant Analysis (Intro)

- Linear Discriminant Analysis (LDA) vs PCA: supervised vs unsupervised
- Running LDA: MASS::lda()
- Classification accuracy and cross-validation

Learning Outcome : Participants can apply PCA for dimensionality reduction, interpret biplots and loading scores, perform hierarchical and k-means clustering, and validate cluster solutions.

Learning Path Overview

Phase 1 : Foundation (Modules 0 - 2)

Build the R environment, learn core programming syntax, master data import, cleaning, and transformation, and develop a solid understanding of descriptive statistics.

Phase 2 : Visualization (Modules 3 - 4)

Develop the ability to create and customize a full range of statistical graphics, from basic histograms and box plots to advanced heatmaps and interactive Plotly charts.

Phase 3 : Statistical Inference (Modules 5 - 7)

Understand probability distributions, assess normality, perform hypothesis tests, and conduct ANOVA with appropriate post-hoc comparisons.

Phase 4 : Modeling and Multivariate Analysis (Modules 8 - 9)

Build correlation and regression models, interpret results, and apply multivariate methods including PCA and clustering.

Key R Packages Covered

Data Wrangling

- tidyverse (dplyr, tidyr, readr, purrr, stringr)
- lubridate
- janitor
- naniar
- readxl

Visualization

- ggplot2
- ggridges
- GGally
- corrplot
- ggcorrplot
- pheatmap
- plotly
- patchwork

Statistics

- rstatix
- car
- ez
- e1071
- agricolae
- dunn.test
- MASS
- DescTools

Multivariate

- factoextra
- cluster
- FactoMineR