



R Programming for Data Analysis Training: RStudio, tidyverse, ggplot2 and R Markdown



R Programming for Data Analysis Training: RStudio, tidyverse, ggplot2 and R Markdown

Introduction:

R Programming for Data Analysis training, using RStudio, tidyverse, ggplot2 and R Markdown, is a 5-day course for analysis, statistics and research teams that ends with a Reproducible R Analysis Report for a case dataset. Many units still copy figures between spreadsheets and point-and-click statistics packages, so tables cannot be rerun and reviewers cannot trace how a result was reached. Nominees already prepare tables, run basic statistics or write research findings, and every session is a hands-on RStudio lab. CoreConcept Training Center delivers this R programming course.

Course Objectives:

- Set up RStudio projects, install CRAN packages and write clean R scripts using vectors, factors, lists and data frames
- Import CSV and Excel files with readr and readxl, then reshape and summarise them with dplyr and tidyr verbs
- Build layered ggplot2 charts that match the analytical question and export them at publication quality
- Run descriptive statistics, t-tests, chi-square tests and ANOVA in R and read the console output correctly
- Fit linear and logistic regression models, check diagnostics and convert model output into tidy tables with broom
- Write reusable R functions and render R Markdown reports that colleagues can rerun on new data

Target Audience:

- Analysis teams responsible for recurring statistical tables, indicators and data extracts
 - Research units responsible for survey, study and evaluation datasets
 - Monitoring and evaluation teams responsible for indicator analysis and periodic results reporting
 - Statistics and planning teams responsible for official figures, projections and data quality checks
 - Quality and laboratory data teams responsible for measurement results and trend reviews
 - Business intelligence teams extending dashboards with statistical models
-

Course Outline:

Day 1: R Language Foundations and the RStudio Workspace

- RStudio Panes, Projects and Working Directory Set-Up
- CRAN Package Installation with `install.packages` and `library`
- R Atomic Vectors, Type Coercion and Missing NA Values
- Factors, Lists and Data Frames as Analysis Containers
- Base R Indexing with Brackets, Dollar Sign and Logical Tests

Day 2: Data Import and the tidyverse Wrangling Grammar

- `readr` and `readxl` Import of CSV and Excel Files
- Tidy Data Principles for Variables, Observations and Values
- `dplyr` `filter`, `select`, `mutate` and `arrange` Verbs
- `dplyr` `group_by` and `summarise` for Grouped Indicator Tables
- `tidyr` `pivot_longer` and `pivot_wider` with `dplyr` `left_join` Merges

Day 3: ggplot2 Visualisation and Statistical Testing in R

- `ggplot2` Aesthetics, Geoms and the Layered Grammar of Graphics
- `ggplot2` Facets, Scales, Themes and `ggsave` Export
- Descriptive Statistics with `summary`, `quantile` and `skimr` Profiles
- `t.test`, `chisq.test` and `aov` Hypothesis Testing Functions
- Effect Size, Confidence Interval and p-Value Reporting in R

Day 4: Regression Models, Dates, Text and Script Quality

- `lm` Linear Regression with Diagnostic Residual Plots
- `glm` Logistic Regression and Odds Ratio Interpretation
- `broom` `tidy` and `glance` for Model Output Tables
- `lubridate` Date Parsing and `stringr` Text Cleaning Patterns
- Custom R Functions, Error Messages and Code Debugging

Day 5: RStudio Lab and the Reproducible R Analysis Report

- R Markdown Report Skeleton with Code Chunks and `knitr`
 - Case Dataset Import, Cleaning and Wrangling Lab
 - Case Dataset `ggplot2` Charts and Statistical Test Lab
 - Case Dataset Regression Model and `broom` Results Table
 - Reproducible R Analysis Report Rendering and Peer Code Review
-

Skills You Will Gain:

- R Programming
- Data Import and Wrangling
- Statistical Graphics Design
- Hypothesis Test Selection
- Regression Modelling
- Date and Text Processing
- Function Writing
- Reproducible Reporting

Why Attend This Course:

- Deliver a Reproducible R Analysis Report for a case dataset to the head of research or analysis and the team that reviews published figures
- Choose between a t-test, chi-square test, ANOVA or regression model for a given research question and defend the choice from the R output
- Avoid untraceable spreadsheet edits, mismatched chart and table figures and manual copy-and-paste steps that slow recurring reports
- Share R scripts, custom functions, ggplot2 themes and R Markdown templates with analysts across the unit

Conclusion:

Back at work, the participant hands the head of research or analysis a Reproducible R Analysis Report in which data import, cleaning, charts, tests and regression output are generated from one script. Reviewers use it to check how each published figure was produced, and the unit uses the template to refresh recurring statistical tables when new data arrives. After its first reuse, the unit should review how long the rerun took, which data issues the script caught, and which functions and ggplot2 themes should become shared team standards.

Frequently Asked Questions FAQ:

What should participants know before an R programming for data analysis course?

Participants should already handle data tables in spreadsheets and understand averages, percentages and basic statistical ideas. No prior coding is required. Installing R and RStudio beforehand and bringing an anonymised dataset from their own work helps them apply the labs directly.

How does R programming for data analysis differ from Python analytics or point-and-click statistics courses?

This course works entirely in R and RStudio with tidyverse, ggplot2 and R Markdown, centred on statistical tests, regression output and reproducible reports. Python courses lean towards machine learning pipelines, while point-and-click statistics courses run tests through menus rather than rerunnable scripts.

Why do analysts use tidyverse packages in R programming for data analysis?

The tidyverse packages share one consistent grammar, so importing with readr, reshaping with dplyr and tidyr and charting with ggplot2 read as one pipeline. Code becomes easier to review, fix and reuse than nested base R commands.

What do participants take back from the R programming for data analysis course?

Participants take back a Reproducible R Analysis Report for a case dataset, with its R Markdown source, import and cleaning scripts, custom functions, ggplot2 charts, test results and broom regression tables, ready to adapt to datasets in their own unit.
