R: A Practical Introduction

Outline

Outline

What is R?

- A language and environment for data analysis, statistics, and graphics.
- Combines interactive exploration with reproducible scripting.
- Runs on Linux, macOS, and Windows.

Why use R?

- Rich statistical toolbox and graphics.
- First-class vectorisation and concise syntax.
- Vast ecosystem of packages for specialised tasks.

Session Basics

- Prompt > evaluates expressions; results print unless assigned.
- Workspace holds objects; history records commands.
- Use scripts (`.R` files) for reproducibility.

Outline

Numeric Vectors

```
x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
2 * x + 1
log(x)
```

Sequences and Replication

```
1:10
seq(-5, 5, by = 0.5)
rep(1:3, each = 2)
```

Logical Values and Missing Data

```
x > 10
is.na(x)
x[is.na(x)] <- 0
```

Characters and Names

```
names <- c("Alice", "Bob", "Carol")
labs <- paste("X", 1:5, sep = "")
```

Indexing and Subsetting

```
x[1]        # first element
x[2:4]      # slice
x[-1]       # drop
x[x > 10]   # logical mask
```

Hands-on #1 (10 min)

- Create a length-12 vector of random normals.
- Compute mean, median, sd, and a z-scored version.
- Replace any values $|z| > 2$ by `NA`.

Outline

Matrices and Arrays

```
A <- matrix(1:6, nrow = 2, ncol = 3)
A[1, 2]; A[, 2]; A[1, ]
```

Factors

```
sex <- factor(c("M", "F", "F", "M"))
levels(sex)
```

Lists

```
person <- list(name="Alice", age=30, scores=c(10,9,8))
person$name
person[["scores"]]
```

Data Frames

```
df <- data.frame(
  height=c(160,170,175),
  weight=c(55,65,72),
  sex=factor(c("F","M","M"))
)
str(df); summary(df)
```

Hands-on #2 (10 min)

- Build a small data frame with 4 numeric columns and 1 factor.
- Add a derived column (e.g., BMI).
- Summarise each column and inspect structure.

Outline

Reading and Writing

```
dat <- read.csv("mydata.csv")
write.csv(dat, "out.csv", row.names = FALSE)
```

Workspaces and Scripts

```
save.image("session.RData")
load("session.RData")
source("analysis.R")
```

Hands-on #3 (5–10 min)

- Load a built-in dataset (e.g., `iris`).
- Create a filtered subset and export to CSV.

Outline

Control Structures

```
if (x > 0) msg <- "positive" else msg <- "non-positive"
for (i in 1:3) print(i^2)
```

Functions

```
summ <- function(x, na.rm = FALSE) {
  if (!is.numeric(x)) stop("x must be numeric")
  if (na.rm) x <- x[!is.na(x)]
  list(mean=mean(x), median=median(x), sd=sd(x))
}
```

... and Argument Passing

```
my_plot <- function(x, y, ...) {
  plot(x, y, ...)
}
```

Hands-on #4 (10 min)

- Write a function that standardises a numeric vector and clips to $[-3, 3]$.
- Add input checks and an `na.rm` option.

Outline

Formula Interface

```
response ~ x1 + x2
y ~ x1 * x2       # main effects + interaction
```

Linear Models

```
fit <- lm(Sepal.Length ~ Sepal.Width + Species, data = iris)
summary(fit)
coef(fit); resid(fit); fitted(fit)
```

Model Diagnostics and Comparison

```
par(mfrow=c(2,2)); plot(fit)
fit1 <- lm(y ~ x1, data=df)
fit2 <- lm(y ~ x1 + x2, data=df)
anova(fit1, fit2)
```

Generalised Linear Models

```
glm_fit <- glm(y ~ x1 + x2, family = binomial, data = df)
summary(glm_fit)
```

Hands-on #5 (10 min)

- Fit a linear model on a tabular dataset.
- Inspect coefficients, residuals, and diagnostic plots.
- Compare nested models with `anova()`.

Outline

High-Level Graphics

```
plot(iris$Sepal.Length, iris$Sepal.Width)
hist(iris$Sepal.Length)
boxplot(Sepal.Length ~ Species, data=iris)
```

Low-Level Additions and Devices

```
plot(iris$Sepal.Length, iris$Sepal.Width)
abline(lm(Sepal.Width ~ Sepal.Length, data=iris))
pdf("plot.pdf"); plot(1:10); dev.off()
```

Hands-on #6 (5–10 min)

- Produce a scatter plot with a fitted line and labels.
- Export to PDF and PNG.

Outline

Packages

```
install.packages("ggplot2")
library(ggplot2)
```

Project-Oriented Workflow

- One project per analysis; keep data, code, and outputs organised.
- Use version control to track changes.
- Prefer scripts over manual steps for repeatability.

Outline

Putting It Together

- Core language: vectors, indexing, data structures.
- Data I/O and management.
- Functions, modelling, and graphics.
- Packages and a reproducible workflow.

Questions

Questions and discussion