

Getting started with R

- R is a statistical programming language available for free at <https://www.r-project.org/>
- RStudio is a user-friendly interface for R (also free) available at <https://posit.co/download/rstudio-desktop/>
- You can use R in the console (type commands directly) or write a script file (a collection of commands to run together)
- Comments in R begin with # and are ignored when the code runs
- Most R commands require parentheses: `function(arguments)`

General information

- R is case-sensitive: `Mean` and `mean` are different
- Use the assignment operator `<-` to assign values to variables (e.g., `x <- 5`)
- You can also use `=` but `<-` is preferred in R
- A vector is a collection of values, created with `c()` (e.g., `x <- c(1, 2, 3, 4, 5)`)
- A data frame is a table where each column is a variable and each row is an observation
- Use `head(data)` to see the first few rows of a data frame
- Use `str(data)` to see the structure of a data frame
- Use `summary(data)` to get basic summary statistics

Basic arithmetic and functions

Command	Output	Description
<code>> 5 + 3</code>	8	Basic addition
<code>> 10 - 4</code>	6	Basic subtraction
<code>> 7 * 2</code>	14	Multiplication
<code>> 20 / 4</code>	5	Division

> 2^5	32	Exponents
> sqrt(16)	4	Square root
> abs(-7)	7	Absolute value
> exp(1)	2.718282	Euler's number (e)
> log(10)	2.302585	Natural logarithm
> log10(100)	2	Base-10 logarithm
> pi	3.141593	The value of π
> round(3.7)	4	Round to nearest integer

Creating and working with vectors

Command	Output	Description
> x <- c(2, 5, 8, 11, 14)		Creates a vector with 5 values
> x	2 5 8 11 14	Displays the vector
> x[2]	5	Accesses the 2nd element
> x[c(1, 3, 5)]	2 8 14	Accesses elements 1, 3, and 5
> length(x)	5	Number of elements in the vector
> sum(x)	40	Sum of all elements
> mean(x)	8	Mean (average) of the values
> median(x)	8	Median value
> var(x)	30	Variance
> sd(x)	5.477226	Standard deviation
> min(x)	2	Minimum value
> max(x)	14	Maximum value
> seq(1, 10, by=2)	1 3 5 7 9	Sequence from 1 to 10 by 2s
> rep(3, 5)	3 3 3 3 3	Repeats 3 five times

Descriptive statistics and data summaries

Command	Output	Description
> table(x)		Frequency table of values
> quantile(x, 0.25)	(value)	First quartile (25th percentile)
> IQR(x)	(value)	Interquartile range
> summary(x)		6-number summary (Min, Q1, Median, Mean, Q3, Max)

Data visualization

Command	Description
> hist(x)	Histogram of the data
> hist(x, breaks=10)	Histogram with 10 bins
> boxplot(x)	Boxplot of the data
> plot(x, y)	Scatterplot of x vs. y
> lines(x, y)	Add a line to an existing plot
> abline(a, b)	Add a line with intercept a and slope b
> barplot(x)	Bar plot of the data
> pairs(data)	Scatterplot matrix for all variables in a data frame

Probability distributions

Command	Description
> dnorm(x, mean=0, sd=1)	Probability density at x for $N(\mu = 0, \sigma = 1)$
> pnorm(x, mean=0, sd=1)	Cumulative probability $P(X \leq x)$ for normal distribution
> qnorm(p, mean=0, sd=1)	Quantile: value where $P(X \leq \text{value}) = p$
> rnorm(n, mean=0, sd=1)	Generate n random values from $N(\mu = 0, \sigma = 1)$
> dbinom(x, size=n, prob=p)	Probability of exactly x successes in n trials
> pbinom(x, size=n, prob=p)	Cumulative probability $P(X \leq x)$ for binomial
> qbinom(p, size=n, prob=prob)	Quantile for binomial distribution
> rbinom(k, size=n, prob=p)	Generate k random values from binomial(n, p)
> dt(x, df)	Probability density for t -distribution with df degrees of freedom
> pt(x, df)	Cumulative probability for t -distribution
> qt(p, df)	Quantile for t -distribution

Hypothesis testing

Command	Description
> t.test(x, mu=0)	One-sample t -test of mean = 0
> t.test(x, y)	Two-sample t -test (independent samples)
> t.test(x, y, paired=TRUE)	Paired t -test
> prop.test(x, n, p=0.5)	Test for a single proportion
> aov(y ~ x, data=data)	One-way ANOVA
> summary(aov.result)	Display ANOVA results

Correlation and regression

Command	Description
---------	-------------

<code>> cor(x, y)</code>	Pearson correlation coefficient
<code>> cor.test(x, y)</code>	Test whether correlation is significant
<code>> lm(y ~ x, data=data)</code>	Fit a linear regression model
<code>> summary(lm.result)</code>	Display regression results
<code>> predict(lm.result, newdata)</code>	Make predictions with regression model
<code>> plot(lm.result)</code>	Diagnostic plots for regression

Confidence intervals

Command	Description
<code>> t.test(x, conf.level=0.95)\$conf.int</code>	95% confidence interval for mean
<code>> t.test(x, y, conf.level=0.95)\$conf.int</code>	95% CI for difference of two means
<code>> prop.test(x, n, conf.level=0.95)\$conf.int</code>	95% CI for a proportion

Useful data operations

Command	Description
<code>> data <- read.csv("file.csv")</code>	Load data from a CSV file
<code>> data\$variable</code>	Access a column in a data frame
<code>> subset(data, x > 5)</code>	Filter rows where $x > 5$
<code>> data\$newvar <- ...</code>	Create a new column
<code>> na.omit(data)</code>	Remove rows with missing values (NA)
<code>> is.na(data)</code>	Check for missing values
<code>> sort(x)</code>	Sort values in ascending order
<code>> sort(x, decreasing=TRUE)</code>	Sort in descending order
<code>> unique(x)</code>	Get unique values
<code>> as.numeric(x)</code>	Convert to numeric type
<code>> as.factor(x)</code>	Convert to categorical (factor) type