Introduction to R — Quick Reference (Sessions 1, 2 & 3)

A compact study reference for demo purposes

1. The Pipe Operator %>%

The pipe takes the result of one function and passes it to the next. Read it as 'and then'. Keyboard shortcut: Ctrl + Shift + M.

```
c(1,0) %>% sample(size=100) %>% table() %>% prop.table()
```

2. Key dplyr Verbs

Verb	What it does
select(-col)	Drop columns with minus sign
filter(price > 100)	Keep rows matching a condition
mutate(new = expr)	Add or change a column
group_by(col)	Split data into groups
summarise(x = mean(col))	Collapse to summary statistics
arrange(desc(col))	Sort rows descending

3. across() and where()

across() applies a function to many columns at once. where() finds columns by type. Use inside mutate() or summarise().

```
mutate(across(where(is.logical), ~ as.numeric(.x), .names = '{.col}_numeric'))
```

4. Apply Family & purrr

Function	Use for	Returns
apply(X, MARGIN, FUN)	Matrices. MARGIN=1 rows, 2 cols	Vector/matrix
lapply(X, FUN)	Lists or dataframe columns	Always a list
sapply(X, FUN)	Lists or dataframe columns	Simplified vector
map(.x, .f)	tidyverse style looping	List
walk(.x, .f)	Side effects (e.g. saving files)	Nothing

5. case_when() — Multi-condition replacement

Like ifelse() but handles unlimited conditions. R checks each condition top to bottom and uses the first match.

```
mutate(price_cat = case_when(price < 50 ~ 'cheap', price < 150 ~ 'medium', .default =
'expensive'))
```

6. glue() and paste0() — Building text dynamically

paste0() sticks text together with no separator. glue() lets you embed variables in curly braces {} for cleaner syntax.

```
glue('./DATA/{filename}') # becomes: ./DATA/data.xlsx
```

```
paste0('data', Sys.Date(), '.xlsx') # becomes: data2026-02-22.xlsx
```

7. Logical Operators

Used inside filter(), ifelse(), case_when(): == (equal), != (not equal), >= (greater or equal), <= (less or equal), & (AND — both true), | (OR — at least one true).

This document is intentionally compact for demo purposes — ~350 words total.