**## dplyr函数：选择变量/列**

data\_selected <- select(data, ID, country, mean\_score)

**## dplyr函数：过滤行**

data\_filtered <- filter(data, mean\_score > 50)

**## dplyr函数：排序数据**

data\_arranged <- arrange(data, desc(mean\_score))

**## dplyr函数：转换或重编码变量**

data\_mutated <- mutate(data, score\_ratio = mean\_score / max\_score)

**## dplyr函数：汇总数据**

data\_summarized <- summarize(data, mean\_of\_mean\_score = mean(mean\_score, na.rm = TRUE))

**## dplyr函数：分组操作**

data\_grouped <- group\_by(data, country)

data\_summarized\_by\_group <- summarize(data\_grouped, mean\_score = mean(mean\_score, na.rm = TRUE))

**## dplyr函数：重命名变量**

data\_renamed <- rename(data, average\_score = mean\_score)

**## dplyr函数：取样数据**

data\_sampled <- sample\_n(data, 10) # 随机取10行

**## dplyr函数：左连接数据框**

data\_joined <- left\_join(data1, data2, by = "ID")

**## dplyr函数：内连接数据框**

data\_inner\_joined <- inner\_join(data1, data2, by = "ID")

**## dplyr函数：右连接数据框**

data\_right\_joined <- right\_join(data1, data2, by = "ID")

**## dplyr函数：全连接数据框**

data\_full\_joined <- full\_join(data1, data2, by = "ID")

**## dplyr函数：半连接数据框**

data\_semi\_joined <- semi\_join(data1, data2, by = "ID")

**## dplyr函数：反连接数据框**

data\_anti\_joined <- anti\_join(data1, data2, by = "ID")

**## dplyr函数：重编码变量**

data\_recoded <- mutate(data, category = recode(category, "A" = "Group 1", "B" = "Group 2", "C" = "Group 3"))

**## tidyr函数：长宽转换 - 将数据从宽格式转换为长格式**

data\_long <- pivot\_longer(data, cols = c(column1, column2), names\_to = "variable", values\_to = "value")

**## tidyr函数：长宽转换 - 将数据从长格式转换为宽格式**

data\_wide <- pivot\_wider(data\_long, names\_from = "variable", values\_from = "value")

**## tidyr函数：填充缺失值**

data\_filled <- fill(data, column\_name, .direction = "down")

**## tidyr函数：分离一个列为多个列**

data\_separated <- separate(data, col = "full\_name", into = c("first\_name", "last\_name"), sep = " ")

**## tidyr函数：合并多个列为一个列**

data\_united <- unite(data, new\_column, c(column1, column2), sep = "\_")

**## tidyr函数：补全数据框中的所有可能组合**

data\_completed <- complete(data, col1, col2)

**## tidyr函数：删除完全缺失的行**

data\_dropped <- drop\_na(data)

**## tidyr函数：替换缺失值**

data\_replaced <- replace\_na(data, list(column\_name = 0))

**## tidyr函数：从列名提取信息并生成新列**

data\_extracted <- extract(data, col = "info", into = c("part1", "part2"), regex = "(\\w+)-(\\d+)")

**## tidyr函数：扩展数据框**

data\_expanded <- expand(data, col1, nesting(col2, col3))

**## base R函数：合并数据框**

data\_merged <- merge(data1, data2, by = "ID")

**## base R函数：计算均值**

mean\_value <- mean(data$numeric\_variable, na.rm = TRUE)

**## base R函数：创建数据框**

data <- data.frame(ID = 1:5, score = c(10, 20, 15, 25, 30))

**## base R函数：查看数据框结构**

str(data)

**## base R函数：查看前几行**

head(data, n = 5)

**## base R函数：查看后几行**

tail(data, n = 5)

**## base R函数：子集数据框**

subset\_data <- data[data$score > 15, ]

**## base R函数：计算列的均值，去除缺失值**

mean\_score <- mean(data$score, na.rm = TRUE)

**## base R函数：排序数据框**

data\_sorted <- data[order(data$score, decreasing = TRUE), ]

**## base R函数：添加新列**

data$total <- data$score \* 2

**## base R函数：合并向量**

combined\_vector <- c(1, 2, 3, 4, 5)

**## base R函数：重复元素**

repeated <- rep(1:3, times = 2)

**## base R函数：生成序列**

sequence <- seq(from = 1, to = 10, by = 2)

**## base R函数：获取数据框列名**

column\_names <- names(data)

**## base R函数：计算唯一值的数量**

num\_unique <- length(unique(data$ID))

**## base R函数：获取唯一值**

unique\_values <- unique(data$score)

**## base R函数：计算向量长度**

vector\_length <- length(data$score)

**## base R函数：检测缺失值**

missing\_values <- is.na(data$score)

**## base R函数：替换某些值**

data$score[data$score == 20] <- 99

**## base R函数：合并多个数据框列**

data$combined <- paste(data$ID, data$score, sep = "-")

**## base R函数：随机打乱顺序**

shuffled\_data <- data[sample(nrow(data)), ]

**## base R函数：重复值检测**

duplicated\_values <- duplicated(data$ID)

**## base R函数：按条件替换值**

data$category <- ifelse(data$score > 15, "High", "Low")