

1. Работа с файлами и директориями

- 1.1. **dir.create** — Directory Create — создать директорию. Создает новую директорию (папку).
Пример: dir.create("новая_папка")
- 1.2. **list.files** — List Files — список файлов. Возвращает список файлов в указанной директории.
Пример: list.files("моя_папка", pattern = "\\.*txt\$")
- 1.3. **file.exists** — File Exists — файл существует. Проверяет существование файла или папки.
Пример: file.exists("отчет.txt")
- 1.4. **file.remove** — File Remove — удалить файл. Удаляет файл.
Пример: file.remove("старый_файл.csv")
- 1.5. **setwd** — Set Working Directory — установить рабочую директорию. Устанавливает текущую рабочую директорию.
Пример: setwd("/путь/к/моей/папке")
- 1.6. **getwd** — Get Working Directory — получить рабочую директорию. Возвращает путь текущей рабочей директории.
Пример: getwd()
- 1.7. **source** - Source - выполнить скрипт. Выполняет R-скрипт из файла.
Пример: source("my_script.R")

2. Чтение и запись данных

- 2.1. **read.csv** — Read CSV — читать CSV. Читает данные из CSV файла.
Пример: my_data <- read.csv("my_file.csv")
- 2.2. **write.csv** — Write CSV — записать CSV. Записывает данные в CSV файл.
Пример: write.csv(my_data, "output.csv", row.names=FALSE)
- 2.3. **read.table** - Read Table - Читать таблицу. Читает данные из текстового файла.
Пример: my_data <- read.table("my_file.txt", header = TRUE)
- 2.4. **write.table** - Write Table - Записать таблицу. Записывает данные в текстовый файл.
Пример: write.table(my_data, "output.txt", row.names=FALSE, sep="\t")

- 2.5. **readRDS** - Read RDS - Читать RDS. Читает R-объект из RDS файла.

Пример: my_object <- readRDS("my_object.rds")

- 2.6. **saveRDS** - Save RDS - Сохранить RDS. Записывает R-объект в RDS файл.

Пример: saveRDS(my_object, "my_object.rds")

- 2.7. **read_xlsx** — Read XLSX — читать XLSX. Читает данные из файла Excel (xlsx).

Пример: my_data <- read_xlsx("my_file.xlsx")

- 2.8. **write_xlsx** - Write XLSX - записать XLSX. Записывает данные в файл Excel (xlsx). *Пример:* write_xlsx(my_data, "output.xlsx")

3. Работа со строками

- 3.1. **paste** — Paste — склеить. Объединяет строки в одну строку.
Пример: paste("Температура:", 25, "градусов")
- 3.2. **paste0** — Paste0 — склеить без разделителя. Объединяет строки в одну строку без разделителя.
Пример: paste0("file", 1, ".txt")
- 3.3. **strsplit** — String Split — разделить строку. Разделяет строку на части по разделителю.
Пример: strsplit("Иван;Петр;Сидор", ";")
- 3.4. **gsub** — Global Substitute — глобальная замена. Заменяет текст в строке по шаблону.
Пример: gsub("яблоко", "груша", "яблоко в саду")

4. Работа с векторами и списками

- 4.1. **c** — Combine — объединить. Объединяет элементы в вектор.
Пример: c(10, 20, 30, 40)
- 4.2. **rep** — Repeat — повторить. Повторяет элементы вектора указанное количество раз.
Пример: rep("привет", 3)
- 4.3. **seq** — Sequence — последовательность. Создает последовательность чисел.
Пример: seq(1, 10, by = 2)
- 4.4. **length** — Length — длина. Возвращает длину вектора или списка.
Пример: length(c(1, 2, 3, 4, 5))
- 4.5. **unique** — Unique — уникальные. Выводит уникальные значения в векторе.
Пример: unique(c(1, 2, 2, 3, 1))

- 4.6. **sample** — Sample — выборка, взять образец. Берет случайную выборку элементов. *Пример:* sample(1:10, 5)

5. Работа с таблицами (data.frame) и tibble

- 5.1. **data.frame** - Data Frame - таблица данных. Создает таблицу данных.
Пример: my_df <- data.frame(col1 = c(1,2,3), col2 = c("A", "B", "C"))
- 5.2. **tibble** — Tibble — таблица. Создает современную таблицу данных. *Пример:* my_tibble <- tibble(col1 = c(1,2,3), col2 = c("A", "B", "C"))
- 5.3. **cbind** — Column Bind — столбцы соединить. Объединяет векторы или таблицы по столбцам.
Пример: cbind(c(1,2,3), c(4,5,6))
- 5.4. **rbind** — Row Bind — строки соединить. Объединяет векторы или таблицы по строкам.
Пример: rbind(c(1,2), c(3,4))
- 5.5. **merge** — Merge — объединить таблицы. Объединяет таблицы по общим столбцам или строкам.
Пример: merge(таблица1, таблица2, by = "номер_клиента")
- 5.6. **left_join** - Left Join - левое объединение. Объединяет таблицы, сохраняя все строки из левой таблицы. *Пример:* left_join(table1, table2, by = "id")
- 5.7. **right_join** - Right Join - правое объединение. Объединяет таблицы, сохраняя все строки из правой таблицы. *Пример:* right_join(table1, table2, by = "id")
- 5.8. **full_join** - Full Join - полное объединение. Объединяет таблицы, сохраняя все строки из обеих таблиц. *Пример:* full_join(table1, table2, by = "id")
- 5.9. **inner_join** - Inner Join - внутреннее объединение. Объединяет таблицы, сохраняя только общие строки.
Пример: inner_join(table1, table2, by = "id")
- 5.10. **names** — Names — имена. Устанавливает или получает имена элементов объекта.
Пример: names(my_vector) <- c("a", "b", "c")
- 5.11. **rename** — Rename — переименовать. Переименовывает столбцы в таблице.
Пример: rename(my_df, new_col = old_col)

5.12. **head** – Head – первые строки. Возвращает первые несколько строк таблицы или вектора.

Пример: head(my_dataframe)

5.13. **tail** – Tail – последние строки. Возвращает последние несколько строк таблицы или вектора.

Пример: tail(my_dataframe)

6. Преобразование данных

6.1. **factor** – Factor – фактор.

Преобразует вектор в факторный.

Пример: my_factor <- factor(c("red", "green", "blue", "red"))

6.2. **as.numeric** – As numeric – как число. Преобразует данные в числовой формат.

Пример: as.numeric(c("1", "2", "3"))

6.3. **as.character** – As character – как текст. Преобразует данные в текстовый формат.

Пример: as.character(c(1, 2, 3))

6.4. **as.Date** – As Date – как дата.

Преобразует данные в формат даты.

Пример: as.Date("2023-10-26")

6.5. **mutate** – Mutate – изменить.

Создаёт или изменяет столбцы в таблице. *Пример:* mutate(my_df, new_col = col1 * 2)

7. Агрегация и применение функций

7.1. **lapply** – List Apply – применить к списку. Применяет функцию ко всем элементам списка.

Пример: lapply(list(1, 4, 9), sqrt)

7.2. **sapply** – Simplify Apply – упростить применение.

Применяет функцию к элементам и возвращает упрощённый результат. *Пример:* sapply(list(2, 4, 6), function(x) x + 10)

7.3. **tapply** – Table Apply – применить к таблицам.

Применяет функцию к подгруппам данных.

Пример: tapply(1:10, rep(c("A", "B"), 5), sum)

7.4. **apply** – Apply – применить.

Применяет функцию к строкам или столбцам матрицы.

Пример: apply(matrix(1:6, nrow=2), 2, prod)

7.5. **aggregate** – Aggregate – агрегировать, собирать.

Агрегирует данные по группам.

Пример: aggregate(my_data\$value, by=list(my_data\$group), FUN=sum)

8. Сортировка и порядок

8.1. **order** – Order – порядок.

Возвращает индексы, которые отсортируют вектор.

Пример: order(c(5, 2, 8, 1))

8.2. **sort** – Sort – сортировать.

Сортирует вектор.

Пример: sort(c(5, 2, 8, 1))

9. Работа с NA (пропущенными значениями)

9.1. **is.na** – Is NA – является ли NA?

Проверяет, является ли значение NA.

Пример: is.na(c(1, NA, 3, NA))

9.2. **na.omit** – NA Omit – опустить NA. Удаляет строки с NA.

Пример: na.omit(my_dataframe)

10. Фильтрация данных

10.1. **filter** – Filter – фильтровать.

Фильтрует строки в таблице по условию.

Пример: filter(my_df, col1 > 10)

11. Условные выражения, циклы и печать

11.1. **ifelse** – If Else – если иначе.

Возвращает значение в зависимости от условия.

Пример: ifelse(my_vector > 10, "больше 10", "не больше 10")

11.2. **if** – If – если

Условное выполнение кода.

Пример: if (x > 0) { print("x положительное") }

11.3. **else** – Else – иначе.

Дополняет if оператор.

Пример: if (x > 0) { print("x положительное") } else { print("x не положительное") }

11.4. **for** – For – цикл для.

Цикл для перебора элементов.

Пример: for (i in 1:5) { print(i) }

11.5. **while** – While – цикл пока

Цикл пока условие истинно.

Пример: i <- 1; while (i <= 5) { print(i); i <- i + 1 }

11.6. **print** – Print – напечатать

Выводит значение на консоль.

Пример: print("Привет, мир!")

12. Работа с функциями

12.1. **function** – Function – функция.

Определяет пользовательскую функцию.

Пример: my_func <- function(x) { return(x * 2) }

13. Информация о данных

13.1. **class** – Class – класс, тип.

Определяет тип переменной.

Пример: class(10)

13.2. **str** – Structure – структура.

Показывает структуру объекта.

Пример: str(my_dataframe)

13.3. **summary** – Summary – сводка.

Предоставляет сводную статистику для данных.

Пример: summary(my_vector)

13.4. **which** – Which – который.

Возвращает индексы,

удовлетворяющие условию.

Пример: which(my_vector > 5)

14. Математические функции

14.1. **abs** – Absolute – абсолютное

значение. Возвращает абсолютное значение числа.

Пример: abs(-5)

15. Системные функции

15.1. **Sys.time()** – System Time – системное время.

Возвращает текущую дату и время.

Пример: format(Sys.time(), "%Y-%m-%d %H:%M:%S")

16. Установка и загрузка пакетов

16.1. **install.packages** – Install Packages – установить пакеты.

Устанавливает пакеты из CRAN.

Пример: install.packages("dplyr")

16.2. **library** – Library – подключить библиотеку

Подключает установленный пакет.

Пример: library(dplyr)

17. Другое / работу с переменными

17.1. **assign** – Assign – присвоить.

Присваивает значение переменной.

Прим.: assign("x", 10)

17.2. **get** – Get – получить, взять.

Возвращает значение переменной по имени.

Пример: get("x")

18. Графика

18.1. **ggplot** – Grammar of Graphics Plot – грамматика графики.

Создаёт графики на основе эстетик и данных.

Пример: ggplot(данные, aes(x=год, y=продажи, color = регион)) + geom_line()