

▼ Лабораторная работа 3. Массивы, списки, факторы, таблицы данных.

Задание 1.

Создайте вектор `state` с 5 элементами: "prof","prof","docent","docent","assist".

Выделите различные категории (факторы) в векторе `state`.

Отобразьте отдельно эти категории (уровни).

Задайте вектор доходов 60,59,40,42,23.

Найдите средние доходы для каждой категории (с помощью функции `tapply`).

Определите еще одну категорию "st.prep" в дополнение к уже исследованным.

Вывести таблицу статистики для всех категорий.

```
# Создание вектора state
state <- c("prof", "prof", "docent", "docent", "assist")

# Преобразование вектора state в фактор
state_factor <- factor(state)

# Уровни (различные категории) фактора state
levels(state_factor)

# Задание вектора доходов
income <- c(60, 59, 40, 42, 23)

# Средние доходы для каждой категории с помощью функции tapply
mean_income <- tapply(income, state_factor, mean)

# Определение новой категории "st.prep"
state <- c(state, "st.prep")

# Обновление фактора state_factor
state_factor <- factor(state)

# Создание таблицы статистики для всех категорий
summary(state_factor)
```

```
assist' · 'docent' · 'prof'
assist:      1 docent:      2 prof:      2 st.prep:      1
```

Задание 2.

Создайте список `x` с тремя компонентами разных типов:

$(1, 2, \dots, 10)$; $(\exp(-3), \exp(-2), \exp(-1), \dots, \exp(3))$; $(\text{TRUE}, \text{FALSE}, \text{FALSE}, \text{TRUE})$.

Задайте имена компонентам списка (например: `a`, `beta`, `logic`).

Выведите на экран содержимое третьей компоненты списка двумя способами - по имени и по индексу. В чем различие между записями: `x[1]` и `x[[1]]`?

Найдите среднее арифметическое значений каждой компоненты списка `x`.

Добавьте к списку `x` четвертую компоненту - последовательность первых 10 букв латинского алфавита. Проименуйте четвертую компоненту как `letters`.

Создайте новый список `y` на основании старого `x` двумя способами - используя индексы и с помощью операции конкатенации.

Преобразуйте список `y` в вектор `z`. К какому классу данных (структур данных) относится `z`? Объекты какого типа содержит `z`?

Объекты какого типа содержал бы `z`, если бы `y` был образован из компонент 1 и 3 списка `x`?

Преобразуйте список `y` в таблицу данных.

```
library(openxlsx)
install.packages("openxlsx", dependencies = TRUE)

# Создание списка x с разными типами данных
x <- list(
  a = 1:10,
  beta = exp(-3:3),
  logic = c(TRUE, FALSE, FALSE, TRUE)
```

```
)

# Добавление имени к компонентам списка
names(x) <- c("a", "beta", "logic")

# Выведение содержимого третьей компоненты списка по имени и по индексу
print(x$logic)
print(x[[3]])

# Различие между x[1] и x[[1]]:
# x[1] возвращает подсписок, тогда как x[[1]] возвращает элемент списка

# Находим среднее арифметическое значений каждой компоненты списка x
sapply(x, mean)

# Добавляем четвертую компоненту к списку x
x[["letters"]] <- letters

# Создаем новый список y на основе списка x
y <- list(x[[1]], x[[3]])
y <- c(x[[1]], x[[3]])

# Преобразуем список y в вектор z
z <- unlist(y)

# Класс данных (структуры данных) z
class(z)

# Объекты, содержащиеся в z
typeof(z)

# Если бы y был образован из компонент 1 и 3 списка x, то объекты, содержащиеся в z, были бы разного типа (числа и логические значения)

# Преобразуем список y в таблицу данных
data <- data.frame(y, stringsAsFactors = FALSE)
```

```
Error in library(openxlsx): there is no package called 'openxlsx'
Traceback:
```

```
1. library(openxlsx)
```

ИСКАТЬ НА STACK OVERFLOW

Задание 3.

Создайте таблицу данных из двух столбцов – год рождения и год поступления для студентов вашей группы. Задайте имена столбцов («год рождения», «год поступления») и строк (фамилии). Фамилии в таблице должны быть расположены в лексикографическом порядке.

Добавьте два новых столбца к таблице данных – «задолженность» и «курс», на котором имеется задолженность. Если задолженности нет, то номер курса = NA. Заполните таблицу данных, руководствуясь достоверными сведениями от ваших одногруппников.

Какова размерность полученной таблицы?

Осуществите поиск по таблице данных (с помощью функции `grep()`) и определите номер(-а) самого молодого(-ых) и самого старого(-ых) студентов в группе.

Выведите на экран фамилию(-ии) самого(-ых) старого(-ых) студента(-ов) в группе.

Выведите на экран фамилию(-ии) самого(-ых) молодого(-ых) студента(-ов) в группе.

Запишите таблицу данных в файлы `table1` и `table2` с помощью различных функций записи (`write.table` и `write.csv`) и прочтите полученные файлы с помощью соответствующих функций чтения.

Запишите таблицу данных в файл Excel (`book.xls`), а затем и прочтите его. (Для этого нужно установить один из специализированных пакетов для работы с Excel файлами - `xlsx` или `openxlsx`).

Добавьте к книге Excel `book.xls` еще один лист, в котором будут храниться только фамилии студентов группы (в алфавитном порядке).

```
data <- data.frame(
  Фамилия = c("Доберштейн", "Серёгин", "Бузов", "Ястребов"),
  "Год рождения" = c(1995, 1996, 1933, 1923),
```

```

"Год поступления" = c(2014, 2015, 1948, 1943)
)

# Сортировка таблицы по фамилиям
data <- data[order(data$Фамилия), ]

# Добавление столбцов "Задолженность" и "Курс"
data$Задолженность <- c(NA, "Да", NA, "Да")
data$Курс <- ifelse(is.na(data$Задолженность), NA, as.integer(substr(data$'год поступления', 3, 4)))

# Размерность таблицы
размерность <- dim(data)
print(размерность)

# Поиск самого молодого и самого старого студентов
min_year_index <- which.min(data$"Год рождения")
min_year_row <- data[min_year_index, ]
min_year_surname <- min_year_row$Фамилия
cat("Самый молодой студент", min_year_surname, "\n")

max_year_index <- which.max(data$"Год рождения")
max_year_row <- data[max_year_index, ]
max_year_surname <- max_year_row$Фамилия
cat("Самый старый студент", min_year_surname, "\n")

# Вывод фамилии самого старого студента
print(max_year_row$Фамилия)

# Вывод фамилии самого молодого студента
print(min_year_row$Фамилия)

# Запись таблицы данных в файлы
write.table(data, "table1.txt", sep="\t", row.names=FALSE)
write.csv(data, "table2.csv", row.names=FALSE)

# Чтение файлов
table1 <- read.table("table1.txt", header=TRUE)
table2 <- read.csv("table2.csv", header=TRUE)

# Загрузка пакета
library(openxlsx)

# Запись таблицы данных в файл Excel
write.xlsx(data, "book.xlsx", sheetName="Sheet1")

# Чтение файла Excel
book <- read.xlsx("book.xlsx", sheet = 1)

# Добавление нового листа в файл Excel
new_sheet <- data.frame(Фамилия = data$Фамилия)
write.xlsx(list(book, new_sheet), "book.xlsx", sheetName=c("Sheet1", "Sheet2"))
data <- read.xlsx("/content/data8.xlsx", sheet = 1)

```

```

[1] 4 5
Самый молодой студент
Самый старый студент
character(0)
character(0)
Error in library(openxlsx): there is no package called 'openxlsx'
Traceback:

```

```
1. library(openxlsx)
```

ИСКАТЬ НА STACK OVERFLOW

Задание 4.

Имеется файл с данными (data8.csv), где записаны результаты измерений погодных условий и даты измерений в столбцах с именами: Ozone, Solar. R, Wind, Temp, Month, Day.

- Создайте таблицу данных `w_data`, прочитав данные из этого файла. Сколько всего наблюдений (по дням) содержится в таблице данных?
- Извлеките первые 2 строки из таблицы данных и выведите их на экран.
- Извлеките последние 2 строки данных выведите их на экран.
- Какое значение у параметра `Ozone` в 47 строке?
- Сколько пропущенных значений (NA) в столбце `Ozone` в таблице данных?

- Найдите среднее значение параметра `Ozone`, исключив из рассмотрения пропущенные значения.
- Извлеките из таблицы данных `w_data` подмножество строк, таких что значения `Ozone` больше 31, а значения `Temp` больше 90. Каково среднее значение параметра `Solar.R` в этой новой выборке?
- Каково среднее значение `Temp`, когда `Month` равен 6?
- Каково максимальное значение `Ozone` в мае месяце?

```
# Чтение файла с данными
w_data <- read.xlsx("data8.xlsx")

# Количество наблюдений в таблице данных
количество_наблюдений <- nrow(w_data)
print(количество_наблюдений)

# Извлечение первых 2 строк и вывод на экран
первые_2_строки <- head(w_data, 2)
print(первые_2_строки)

# Извлечение последних 2 строк и вывод на экран
последние_2_строки <- tail(w_data, 2)
print(последние_2_строки)

# Значение параметра Ozone в 47 строке
значение_Ozone_в_47_строке <- w_data[47, "Ozone"]
print(значение_Ozone_в_47_строке)

# Значение параметра Ozone в 47 строке
значение_Ozone_в_47_строке <- w_data[47, "Ozone"]
print(значение_Ozone_в_47_строке)

# Количество пропущенных значений (NA) в столбце Ozone
количество_пропущенных_значений <- sum(is.na(w_data$Ozone))
print(количество_пропущенных_значений)

# Среднее значение параметра Ozone, исключая пропущенные значения
среднее_значение_Ozone <- mean(w_data$Ozone, na.rm = TRUE)
print(среднее_значение_Ozone)

# Извлечение подмножества строк с условиями Ozone > 31 и Temp > 90
новая_выборка <- subset(w_data, Ozone > 31 & Temp > 90)
# Среднее значение параметра Solar.R в новой выборке
среднее_значение_Solar_R <- mean(новая_выборка$Solar.R)
print(среднее_значение_Solar_R)

# Среднее значение параметра Temp, когда Month равен 6
среднее_значение_Temp_при_Month_равно_6 <- mean(w_data$Temp[w_data$Month == 6])
print(среднее_значение_Temp_при_Month_равно_6)

# Максимальное значение параметра Ozone в мае
максимальное_значение_Ozone_в_мае <- max(w_data$Ozone[w_data$Month == 5])
print(максимальное_значение_Ozone_в_мае)

Error in read.xlsx("data8.xlsx"): could not find function "read.xlsx"
Traceback:
```

ИСКАТЬ НА STACK OVERFLOW