Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра вычислительных технологий

**ОТЧЕТ**

о выполнении лабораторной работы № 4

по дисциплине «Обработка больших данных»

Тема: Извлечение данных с WEB-страниц. Пакет rvest.

Выполнил: ст. гр. 36/2

Агаджанян А. С.

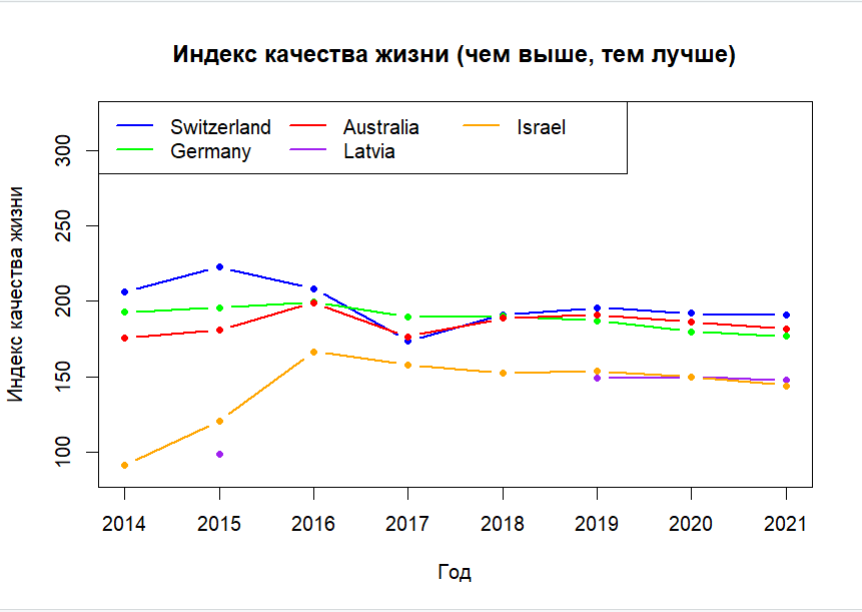
Проверил: преп. кафедры ВТ

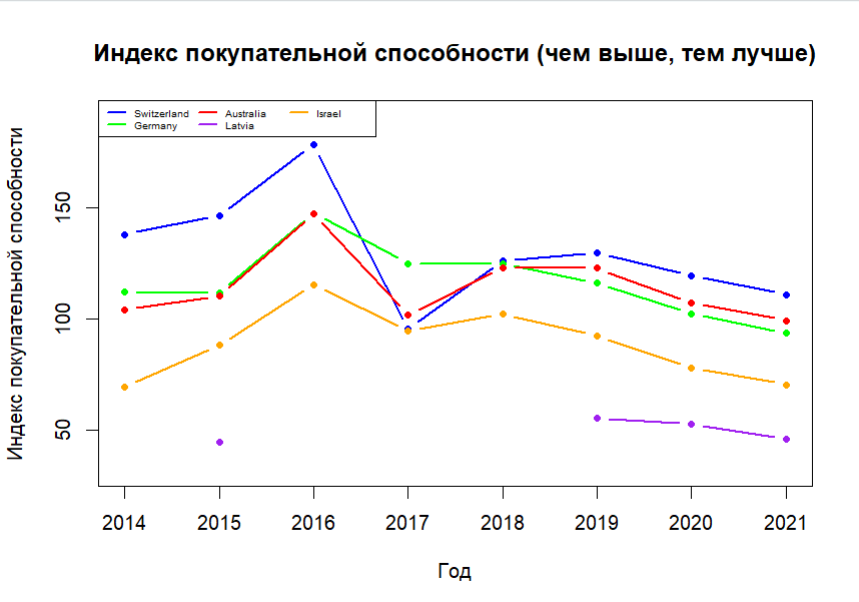
Яхонтов А. А.

Краснодар

2025

Вариант 17: Швейцария, Германия, Австралия, Латвия, Израиль

Индекс качества жизни вычисляется на основании нескольких факторов, графики которых приведены ниже. Чем выше данный показатель, тем лучше. Исходя из полученных данных можно сделать вывод, что с индекс качества жизни во всех представленных странах примерно одинаковый, только у Израиля и Латвии он чуть ниже, чем у остальных. Также заметим, что линия у Израиля прерывается в период с 2015 года по 2019. Данных за этот период времени не найдены.

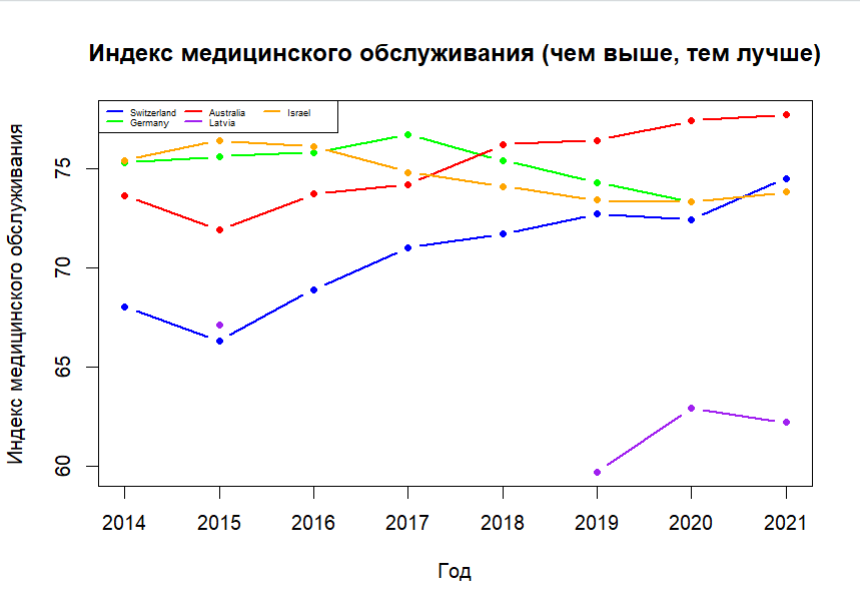


По индексу покупательной способности все годы, кроме 2017 лидирует Швейцария. В целом показатели Швейцарии, Германии, Австралии и Израиля держатся близко друг к другу. Латвия по данному критерию сильно ниже.

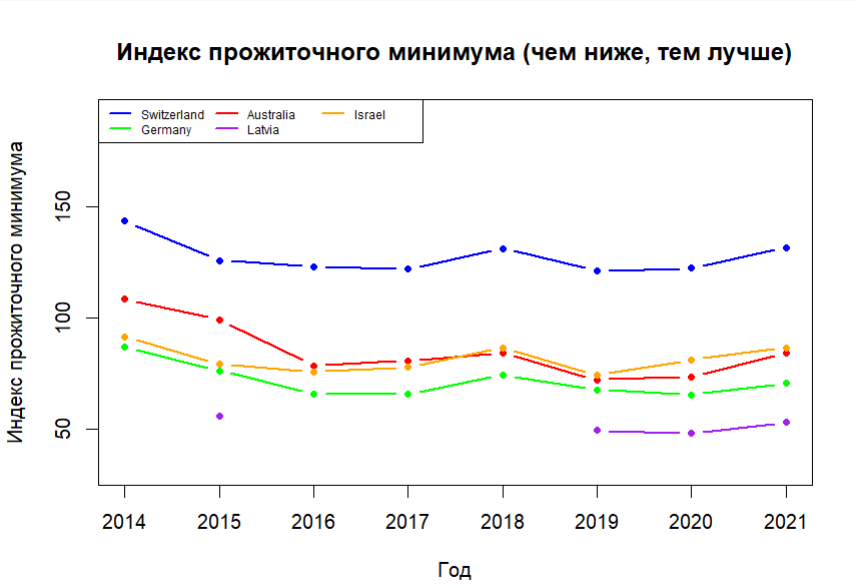
Изображение выглядит как текст, снимок экрана, линия, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

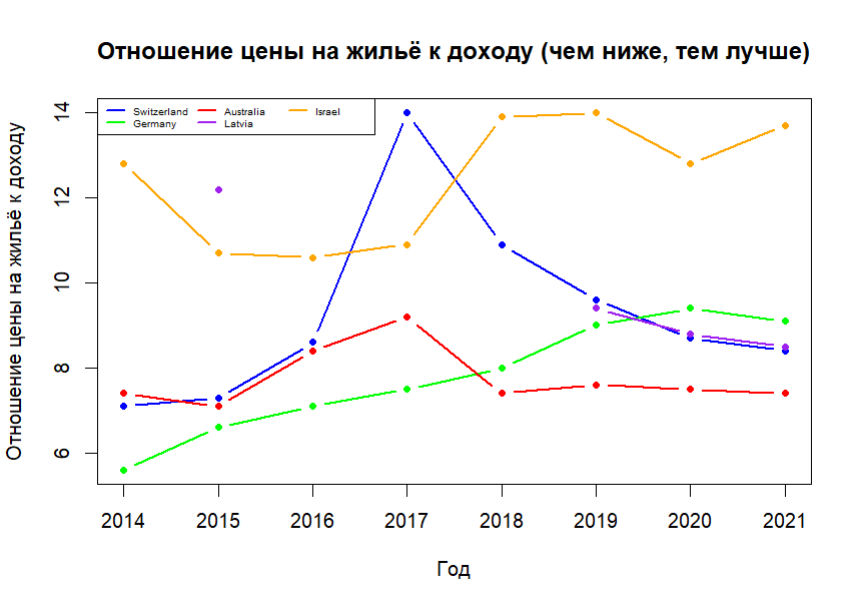
По индексу безопасности все страны держатся близко друг к другу, однако показатель Швейцарии немного выше, чем остальные. Также видно, что в странах нет никаких резких изменений по данному индексу.



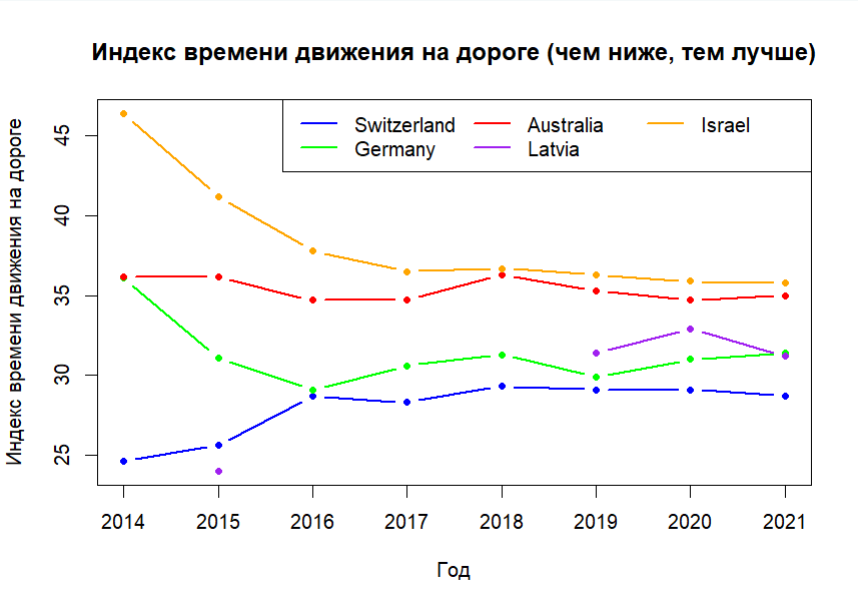
В 2014-2016 гг. индекс медицинского обслуживание лучше всего был в Израиле, в 2017 году – в Германии, в 2018-2021 гг. – в Австралии. Показатель Латвии сильно ниже, чем в остальных странах.



Чем ниже индекс прожиточного минимума, тем лучше. Исходя из графика, можно сделать вывод, что самый высокий индекс наблюдается в Швейцарии на протяжении всех лет. Наилучшим показателем обладает Латвия. А показатели Германии, Австралии и Израиля держатся близко друг к другу.



На данном графике видно, что в Германии отношение цены на жильё к доходу наименьшее в период с 2014 года по 2017 – это означает, что жителям Германии проще купить жильё, чем жителям других рассматриваемых стран. В Австралии с 2018 по 2021 год данный показатель значительно уменьшился. В Швейцарии индекс сначала увеличивался, достигнув своего пика в 2017 году.



Здесь также чем ниже, тем лучше. В Израиле люди тратят наибольшее количество времени на дорогу среди всех пяти стран, причём отрыв незначительный. Быстрее всего добираться на транспорте в Швейцарии.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Видно, что наиболее загрязненная страна – Израиль, причем отрыв значительный. Остальные страны по индексу находятся примерно в одном диапазоне. Наименее загрязненная страна – Швейцария.

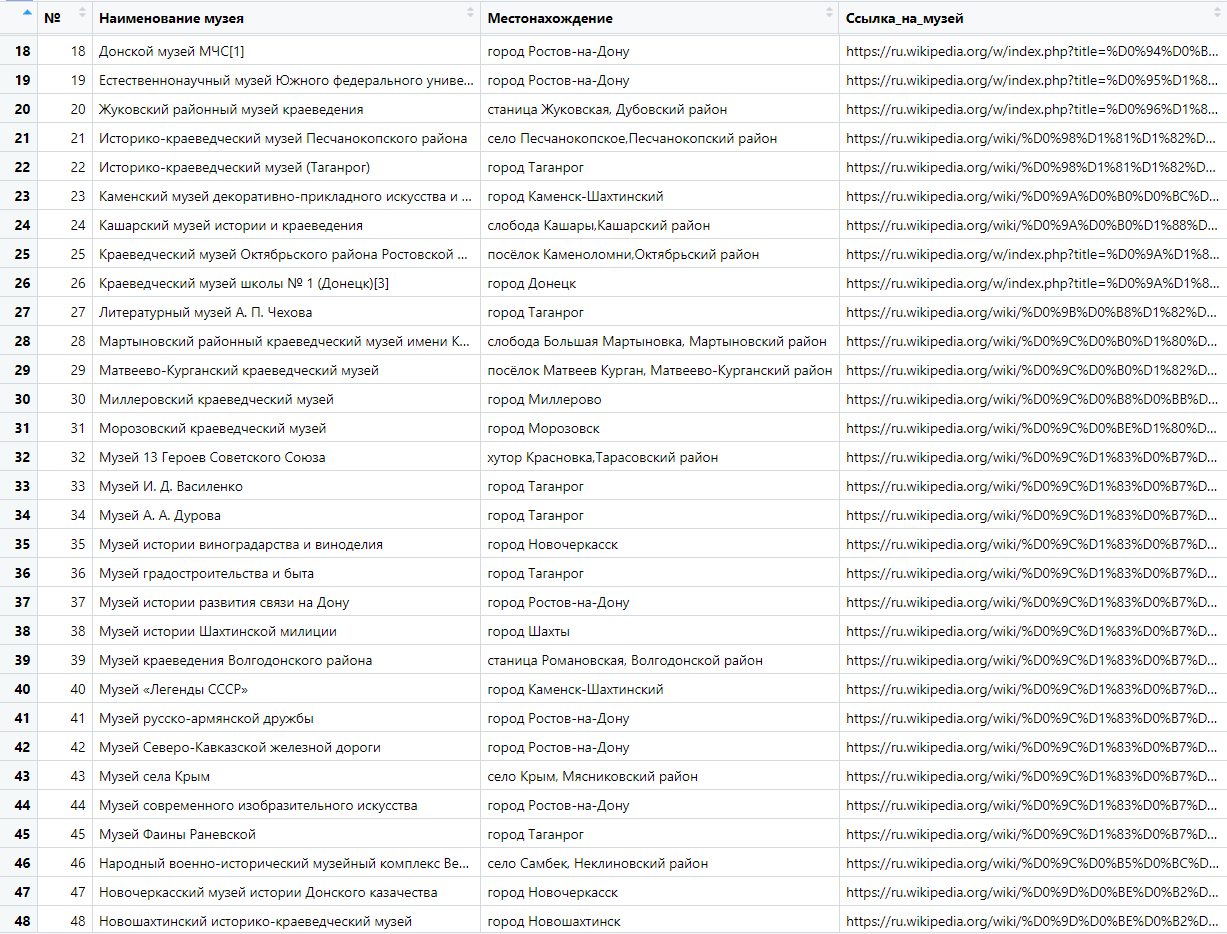
Изображение выглядит как текст, диаграмма, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Данные по климату до 2016 года отсутствуют. Наилучший показатель у Израиля, наихудший у Латвии. Наибольший рост показателя датируется 2018 годом, после этого показатель практически не менялся.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.



Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Листинг (lab4.R):

install.packages("rvest")

options(timeout = 300)

install.packages("stringi")

library(stringi)

library(rvest)

years <- 2014:2021

countries <- c("Switzerland", "Germany", "Australia", "Latvia", "Israel")

colors <- c("blue", "green", "red", "purple", "orange")

getData <- function(year) {

cat(sprintf("Получение данных за %d год...", year))

url <- sprintf("https://www.numbeo.com/quality-of-life/rankings\_by\_country.jsp?title=%d", year)

page <- read\_html(url)

table\_node <- html\_nodes(page, "table#t2")[[1]]

table <- as.data.frame(html\_table(table\_node, na.strings="-"))

table <- table[-1] # убираем RANK

cat(" Готово\n")

return(table)

}

getCountryRow <- function(allData, year, country) {

table <- allData[as.character(year)][[1]]

table <- subset(table, table$Country == country)

table <- table[-1]

return(table)

}

allData <- lapply(years, getData) #возвращает список той же длины, что и years, в котором к каждому элементу years была применена функция

names(allData) <- years

makeCountriesComparisonDF <- function(allData, countries, dataColumnIndex) {

years <- names(allData)

countriesDataList <- lapply(countries, function(country) {

listDataByYears <- lapply(years, function(year) {

country\_row <- getCountryRow(allData, year, country)

if (is.null(country\_row) || nrow(country\_row) == 0) {

index\_value <- NA

} else {

index\_value <- country\_row[1, dataColumnIndex]

}

df <- data.frame(value = index\_value)

rownames(df) <- year

colnames(df) <- country

return(df)

})

do.call("rbind", listDataByYears)

})

do.call("cbind", countriesDataList)

}

QOLIData <- makeCountriesComparisonDF(allData, countries, 1)

# Вычисляем min и max с обработкой NA

min\_qoli <- min(as.matrix(QOLIData), na.rm = TRUE)

max\_qoli <- max(as.matrix(QOLIData), na.rm = TRUE)

# Проверяем, являются ли min и max конечными, и устанавливаем значения по умолчанию, если нет

if (!is.finite(min\_qoli)) {

min\_qoli <- 0 # Значение по умолчанию

}

if (!is.finite(max\_qoli)) {

max\_qoli <- 100 # Значение по умолчанию

}

matplot(

years,

QOLIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

ylim=c(min\_qoli - 5, max\_qoli + 100),

main='Индекс качества жизни (чем выше, тем лучше)',

xlab='Год',

ylab='Индекс качества жизни'

)

legend('topleft', countries, ncol=3, lty=1, lwd=2, col=colors)

# Второй график (Индекс покупательной способности)

PPIData <- makeCountriesComparisonDF(allData, countries, 2)

# Вычисляем диапазон для PPIData

ppi\_range <- range(as.matrix(PPIData), na.rm = TRUE)

# Проверяем, являются ли значения конечными, и устанавливаем значения по умолчанию, если нет

if (!all(is.finite(ppi\_range))) {

min\_ppi <- 0 # Значение по умолчанию

max\_ppi <- 200 # Значение по умолчанию (увеличено)

ppi\_range <- c(min\_ppi, max\_ppi) # Устанавливаем диапазон вручную

}

# Расширяем диапазон на 10% (можно изменить этот процент)

range\_extension <- 0.1 \* diff(ppi\_range)

min\_ppi <- ppi\_range[1] - range\_extension

max\_ppi <- ppi\_range[2] + range\_extension

matplot(

years,

PPIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

ylim=c(min\_ppi, max\_ppi), # Используем вычисленные min\_ppi и max\_ppi

main='Индекс покупательной способности (чем выше, тем лучше)',

xlab='Год',

ylab='Индекс покупательной способности'

)

legend('topleft', countries, ncol=3, cex=0.5, lty=1, lwd=2, col=colors)

SIData <- makeCountriesComparisonDF(allData, countries, 3)

matplot(

years,

SIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

ylim=c(min\_ppi, max\_ppi), # Используем вычисленные min\_ppi и max\_ppi

main='Индекс безопасности (чем выше, тем лучше)',

xlab='Год',

ylab='Индекс безопасности'

)

legend('topleft', countries, ncol=3, lty=1, lwd=2, col=colors)

HCIData <- makeCountriesComparisonDF(allData, countries, 4)

matplot(

years,

HCIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

main='Индекс медицинского обслуживания (чем выше, тем лучше)',

xlab='Год',

ylab='Индекс медицинского обслуживания'

)

legend('topleft', countries, ncol=3, lty=1, cex = 0.45, lwd=2, col=colors)

COLIData <- makeCountriesComparisonDF(allData, countries, 5)

matplot(

years,

COLIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

ylim=c(min\_ppi, max\_ppi),

main='Индекс прожиточного минимума (чем ниже, тем лучше)',

xlab='Год',

ylab='Индекс прожиточного минимума'

)

legend('topleft', countries, ncol=3, lty=1, cex = 0.6, lwd=2, col=colors)

PPTIRData <- makeCountriesComparisonDF(allData, countries, 6)

matplot(

years,

PPTIRData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

main='Отношение цены на жильё к доходу (чем ниже, тем лучше)',

xlab='Год',

ylab='Отношение цены на жильё к доходу'

)

legend('topleft', countries, ncol=3, lty=1, cex = 0.5, lwd=2, col=colors)

TCTIData <- makeCountriesComparisonDF(allData, countries, 7)

matplot(

years,

TCTIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

main='Индекс времени движения на дороге (чем ниже, тем лучше)',

xlab='Год',

ylab='Индекс времени движения на дороге'

)

legend('topright', countries, ncol=3, lty=1, lwd=2, col=colors)

PIData <- makeCountriesComparisonDF(allData, countries, 8)

matplot(

years,

PIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

main='Индекс загрязнения (чем ниже, тем лучше)',

xlab='Год',

ylab='Индекс загрязнения'

)

legend('topright', countries, ncol=3, cex=0.5, lty=1, lwd=2, col=colors)

CIData <- makeCountriesComparisonDF(allData, countries, 9)

matplot(

years,

CIData,

type="b",

pch=16,

lty=1,

lwd=1.8,

cex=0.8,

col=colors,

main='Климатический индекс (чем выше, тем лучше)',

xlab='Год',

ylab='Климатический индекс'

)

legend('topleft', countries, ncol=3, cex=0.5, lty=1, lwd=2, col=colors)

Листинг (lab4\_museum.R):

library(rvest)

# URL страницы Википедии

url <- "https://ru.wikipedia.org/wiki/Список\_музеев\_Ростовской\_области"

# Чтение HTML-кода страницы

page <- read\_html(url)

# Определение CSS-селектора для таблицы (ищем таблицы с классом 'wikitable')

table\_nodes <- html\_nodes(page, "table.wikitable")

museum\_table <- table\_nodes[[1]]

table\_rows <- html\_nodes(museum\_table, "tr")

if (length(table\_rows) > 1) {

table\_rows <- table\_rows[-1] # Убираем первую строку

}

second\_column\_cells <- html\_nodes(table\_rows, "td:nth-child(2)")

links <- html\_nodes(second\_column\_cells, "a:not(a[href^=\"#cite\_note\"])")

museum\_links <- html\_attr(links, "href")

base\_url <- "https://ru.wikipedia.org"

full\_museum\_links <- paste0(base\_url, museum\_links)

full\_museum\_links <- unique(full\_museum\_links)

full\_museum\_links <- full\_museum\_links[full\_museum\_links != ""]

museum\_links\_df <- data.frame(

Ссылка\_на\_музей = full\_museum\_links,

stringsAsFactors = FALSE

)

museum <- html\_table(table\_nodes[[1]], fill = TRUE)

museum <- museum[, -ncol(museum)]

museum <- museum[, -ncol(museum)]

merged\_df <- cbind(museum, museum\_links\_df)

Вывод: в ходе выполнения лабораторной работы были изучены основы библиотеки rvest для веб-скрейпинга. Библиотека позволяет напрямую работать с данными, представленными на сайтах. В ней имеется удобная возможность использования CSS-селекторов, функция для парсинга HTML таблиц и др. Также были построены графики на основе данных из Интернета, что позволило легко проанализировать разные аспекты жизни нескольких стран.