МИНОБРНАУКИ РОССИИ

РГУ НЕФТИ И ГАЗА (НИУ) ИМЕНИ И.М. ГУБКИНА

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ИНФОРМАТИКИ

ДИСЦИПЛИНА «ОСНОВЫ АНАЛИЗА БОЛЬШИХ ДАННЫХ И МАШИННОЕ ОБУЧЕНИЕ»

**О Т Ч Е Т**

**по Лабораторной работе 3**

**«Визуализация данных в среде R»**

Выполнила: студентка группы АА-19-05

Данилова М.А.

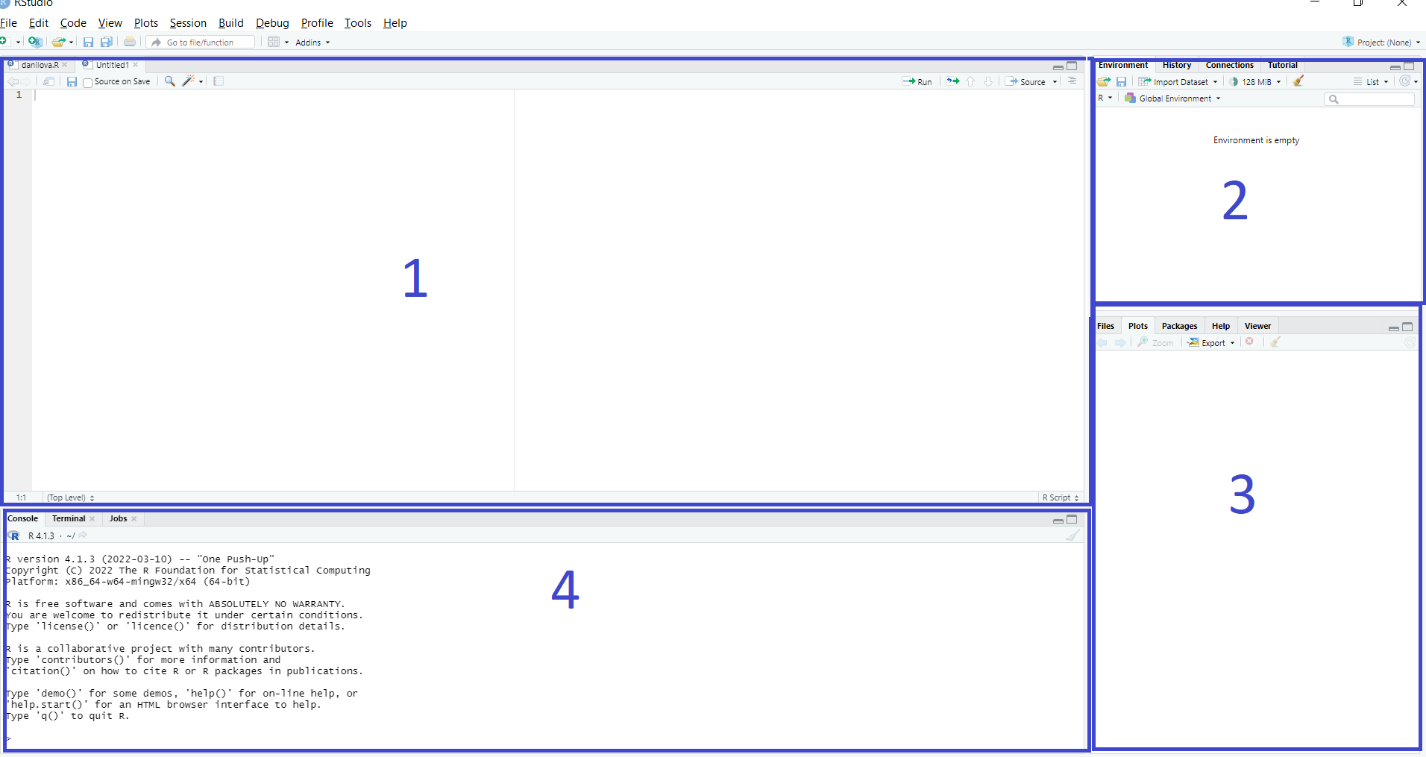
Проверила: доцент Вишневская Е. А.

Москва 2022

**1-2. Изучить среду разработки и организацию работы с данными в R.**

Для визуализации в среде R потребуется RStudio, а также инструмент RTools.

Рабочее поле RStudio выглядит следующим образом:



Окно редактора разделено на 4 области:

1- код - обычный текстовый редактор, в котором редактируют файлы с программами.

2 - среда/история. На вкладке история есть список всех выполненных команд R. На вкладке среда расположен список объектов, находящихся в памяти.

3 - файлы/графики/пакеты/помощь/просмотр. На вкладке помощь можно просматривать файлы справки. На вкладке графики будут появляться графики. На вкладке файлы расположен файловый менеджер, который позволяет перемещаться по директориям проекта и компьютера.

4 - консоль. На вкладке можно в живом режиме выполнять команды R, если вы не хотите их сохранять в виде программы.

В R всего шесть различных типов данных: числовые - numeric позволяет хранить целые и дробные числа, integer предназначен для хранения целых чисел, complex используется для хранения комплексных чисел, текстовые для хранения последовательностей символов произвольной длины, логические для записи значений типа истина/ложь, raw используется для хранения “сырых” последовательностей байтов. Данные можно вводить с клавиатуры, импортировать из текстовых файлов из Microsoft Excel и Access, а также из различных систем управления базами данных.

**2. Ознакомиться с возможностями графического пакета ggplot2.**

ggplot2 – это расширение языка R, предназначенное для визуализации данных; система декларативного создания графики, основанная на грамматике графики.

В ggplot2 график является результатом взаимодействия ряда элементов:

* Массив данных (**data**), используемых для построения.
* Схема соответствия переменных из массива визуальным средствам (**aes**thetic) – это набор правил, как различные переменные превращаются в визуальные особенности геометрии. У каждой геометрии свой набор эстетик, но многие из них совпадают у разных geom, например, x, y, colour, fill, size. Без некоторых эстетик geom не будет работать. Например, геометрия в виде точек не будет работать без двух координат этих точек (x и y). Другие эстетики необязательны и имеют значения по умолчанию.
* Геометрический объект (**geom**) – в какой геометрический объект мы собираемся превращать данные. Например, в точки, прямоугольники или линии.
* Статистическое преобразование (**stat**) - название используемой статистической трансформации. В ggplot2 у каждого геома есть статистика по умолчанию, а у каждой статистики - свой геом по умолчанию. Например, для барплота (geom\_barplot()) используется статистика "count", которая считает частоты, и именно частоты затем трансформируются в высоту барплотов.
* Координатная система (**coord**) - работа с пространством, на котором располагается график. В большинстве случаев используется декартова система координат, но можно использовать и другие, например, полярную систему координат или картографическую проекцию.
* Шкалы (**scales**). Шкалы задают то, как именно значения превращаются в эстетики. Например, если мы задали, что разные значения в колонке будут влиять на цвет точки, то какая именно палитра будет использоваться; в какие конкретно цвета будут превращаться числовые, логические или строковые значения в колонке и тд. В ggplot2 есть правила по умолчанию для всех эстестик.
* Панели (**facet**). Фасетки повзоляют разбить график на множество похожих, задав переменную, по которой график будет разделен.
* Художественное оформление (**theme**). Зрительное оформление "подложки" графика, не относящийся к содержанию: размер шрифта, цвет фона, размер и цвет линий на фоне и т.д

Основной синтаксис ggplot2:

ggplot(data, mapping=aes()) +

geometric object

arguments:

data: Dataset used to plot the graph

mapping: Control the x and y-axis

geometric object: The type of plot you want to show. The most common object are:

- Point: `geom\_point()`

- Bar: `geom\_bar()`

- Line: `geom\_line()`

- Histogram: `geom\_histogram()`

Знак + означает, что вы хотите, чтобы R продолжал читать код. Это делает код более читабельным, разбивая его. Все явно не указанные пользователем элементы графика берутся из значений по умолчанию.

Графики ggplot2 многослойные, то есть строятся поэтапно: сначала указывается датафрейм, с которым мы работаем, и интересующие нас показатели (первый слой), затем указывается тип графика (второй слой), затем настройки для подписей, легенды и прочее (остальные слои). Все слои добавляются через +.

Пакет позволяет строить линейные графики, гистограммы, «ящики с усами», скрипичные диаграммы, диаграммы рассеивания и тд.

**3. Изучить базовые средства визуализации данных в R.**

**plot(x, y, …)** универсальная функция для построения графиков.

x, y – координаты точек;

type - позволяет изменять внешний вид точек на графике ("p" - точки по умолчанию; "l" - линии (lines); "b" - изображаются и точки, и линии (both); "o" - точки изображаются поверх линий (over lines); "h" - гистограмма (histogram);

"s" - ступенчатая кривая (steps); "n" - данные не отображаются (no points));

main - общее название сюжета;

sub - подзаголовок для сюжета;

xlab - заголовок для оси x;

ylab - заголовок для оси y;

asp - соотношение сторон и тд.

**barplot** - столбчатые диаграммы; отражают распределение категориальной переменной в виде вертикальных или горизонтальных столбиков. Самый простой способ применить функцию barplot:

barplot(height)

где height – это вектор или матрица.

**Круговые диаграммы** создаются при помощи функции

pie(x, labels)

где x – это лишенный отрицательных значений числовой вектор, определяющий площадь каждого сегмента диаграммы, а labels – текстовый вектор, содержащий подписи для сегментов.

**Гистограммы** графически отображают распределение значений непрерывных переменных, разделяя диапазон значений на заданное число отрезков по оси x и отображая частоту значений внутри каждого отрезка на оси y. Гистограмму можно создать при помощи команды

hist(x)

где x – это числовой вектор. Опция freq=FALSE позволяет построить гистограмму на основе плотности вероятности, а не частот значений. Параметр breaks определяет число отрезков. По умолчанию все отрезки имеют одинаковую длину.

**Диаграммы размахов (boxplot)** иллюстрируют распределение значений непрерывной переменной, отображая пять параметров: минимум, нижний квартиль (25-й процентиль), медиану (50-й процентиль), верхний квартиль (75-й процентиль) и максимум. На этой диаграмме также могут быть отображены вероятные выбросы (значения, выходящие за диапазон в ±1.5 межквартильного размаха, разности верхней и нижней квартилей).

**4. Для подготовленного набора данных создать различные визуализации с**

**использованием базовых диаграмм.**

Для визуализации используем данные, представленные на сайте: <http://www.statsci.org/data/general/fev.html>

Данные содержат определения ОФВ у 654 детей в возрасте от 6 до 22 лет, которые были задействованы в исследовании респираторной болезни в 1980 году в Восточном Бостоне, штат Массачусетс. Объём форсированного выдоха (ОФВ) – это объём воздуха, который исследуемый может выдохнуть за первую секунду максимального форсированного выдоха.

Структура данных:

ID - Идентификационный номер

Age - возраст

FEV - ОФВ, литров

Height - рост, дюймы

Sex - пол: мужской или женский

Smoker – является ли исследуемый курильщиком. Non = некурящий, Current = курильщик.

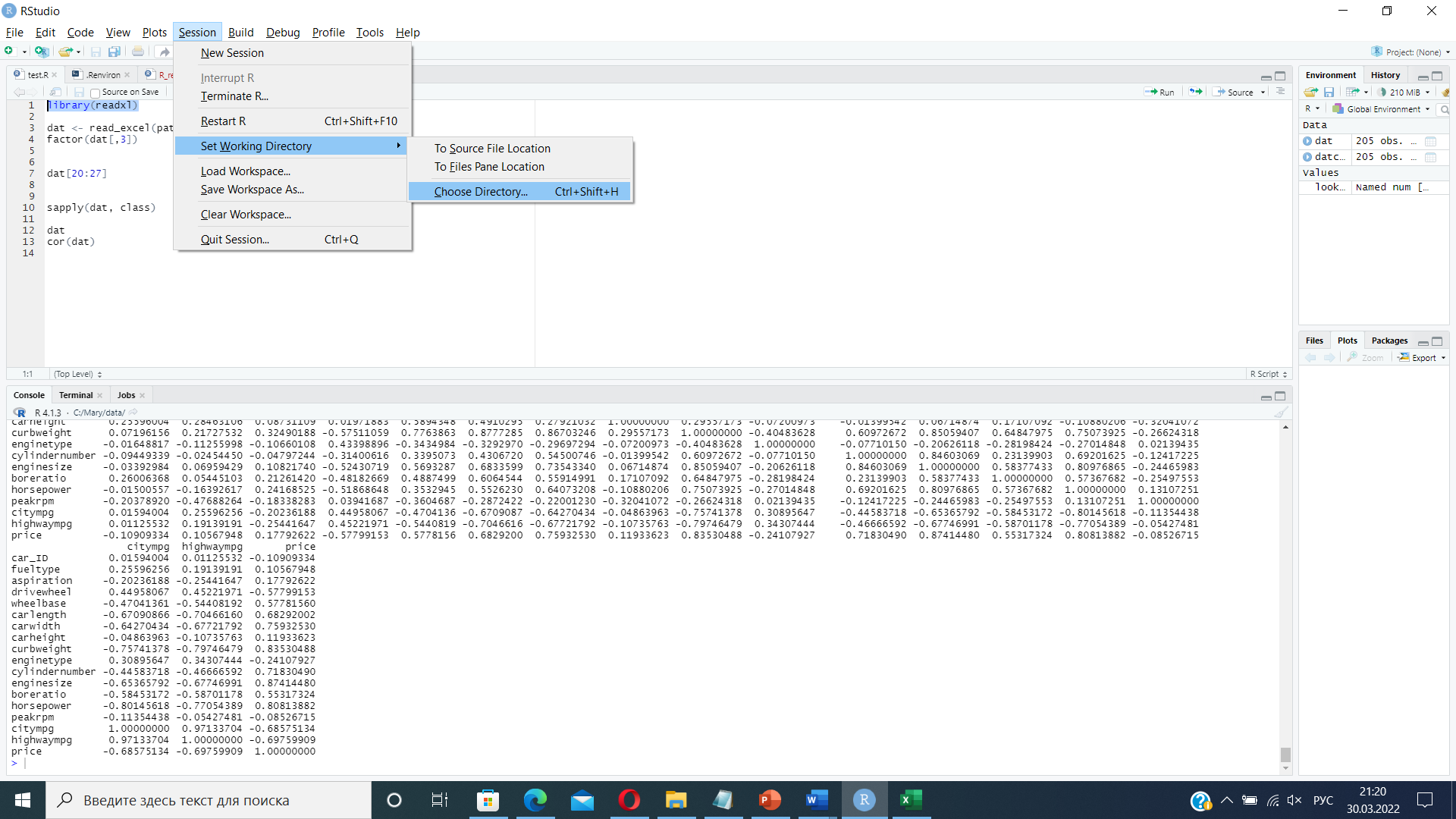
Требуется рассмотреть зависимость ОФВ от остальных параметров, в частности определить влияние курения на ОФВ и проанализировать состав курильщиков (представителей какого пола и возраста среди них больше всего).

Для чтения данных из Excel загрузим и установим необходимый пакет

[install.packages](https://rdrr.io/r/utils/install.packages.html)("readxl")

library(readxl)

Прочитаем данные из Excel, предварительно указав рабочую директорию:

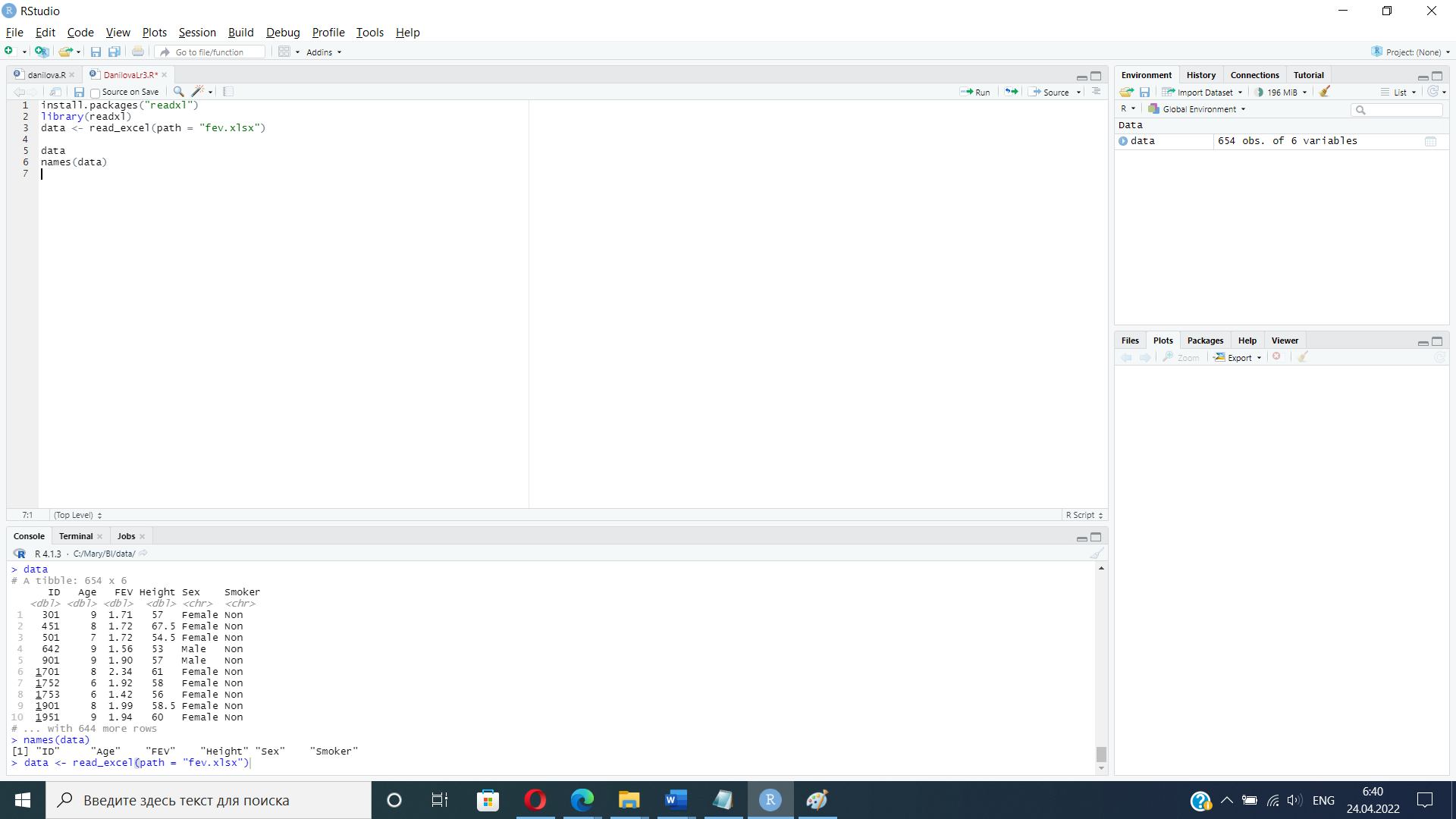


data <- read\_excel(path = "fev.xlsx")

где path-путь относительно рабочей директории.

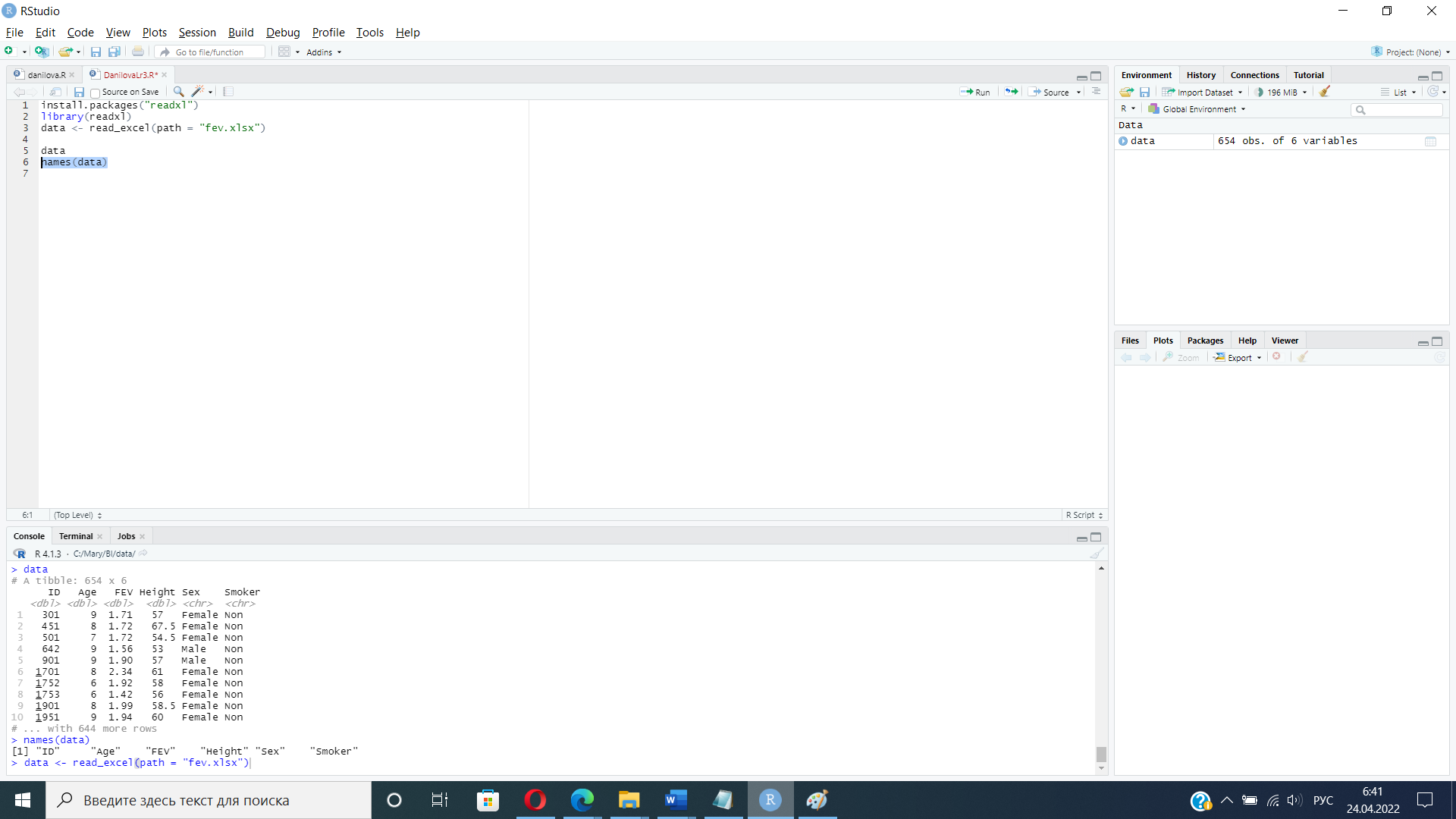
Посмотрим на полученные данные:

data



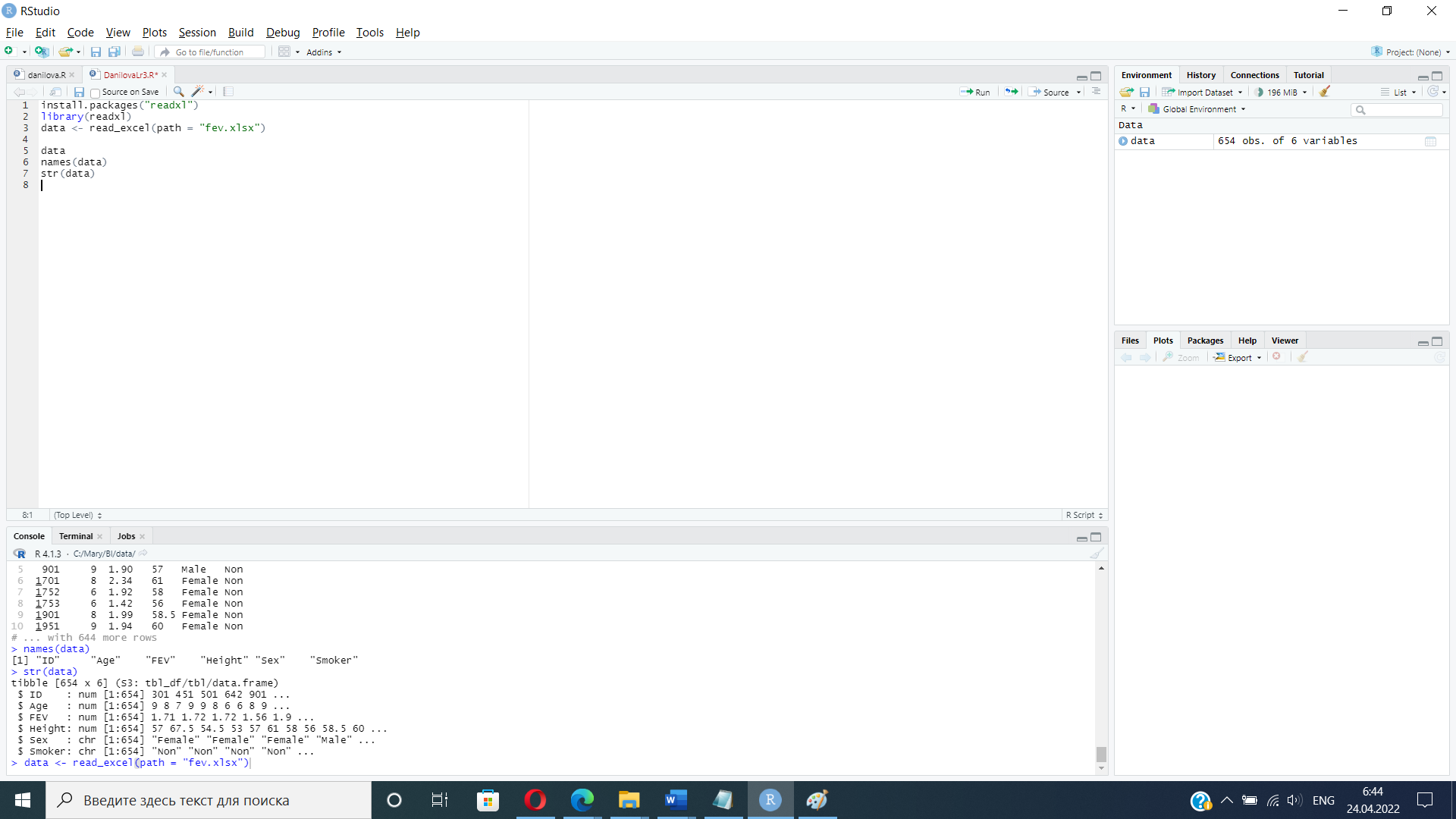
Отобразим имена полей

names(data)



Отобразим тип переменных с помощью функции str. Str() может отображать внутреннюю структуру больших вложенных списков. Метод str() принимает объект R в качестве аргумента и возвращает внутренние сведения об этом объекте.

str(data)



Поля «пол» и «курильщик» являются текстовыми.

Необходимо заменить формат в переменных Sex и Smoker

data$Sex <- factor(data$Sex)

data$Smoker <- factor(data$Smoker)

функция factor() превращает числовые или текстовые данные в дискретные факторы

Проверим, есть ли «пустые» ячейки в данных (NA). Для этого можно создать собственную функцию, передать ей в качестве аргумента данные. nrow считает количество строк в списке, ncol – количество столбцов. is.na возвращает 1 в случае, если аргумент является NA. Print выводит полученное значение

find\_na <- function(data){

sum =0

for(i in 1:nrow(data)){

for (j in 1:ncol(data))

{

sum<-sum+is.na(data [i,j])

}

}

print(sum)

}

find\_na(data)

Результат: 0, те пропущенных значений нет.

Загрузим и установим необходимый пакет для визуализации

install.packages("ggplot2")

library(ggplot2)

Для начала определим, одинаковое ли количество исследуемых подростков по полу представлено в рассматриваемых данных и кого из них больше – курящих или нет. Поскольку эти переменные – дискретные факторы, можно отобразить их на круговой диаграмме.

plot0 <- ggplot(data = data) +

geom\_bar(aes(x = "", fill = Sex)) +

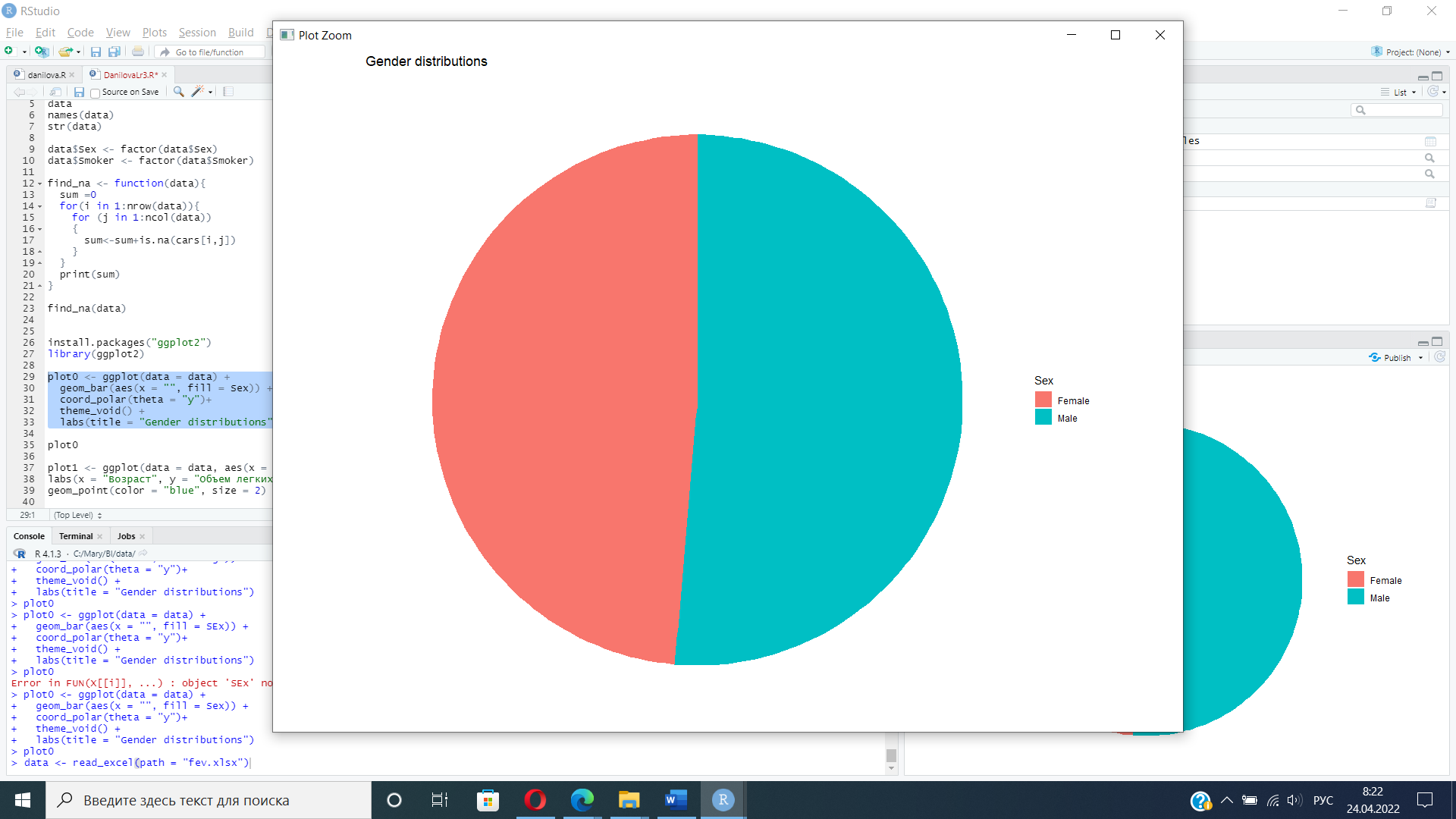
coord\_polar(theta = "y")+

theme\_void() +

labs(title = "Gender distributions")

plot0 отобразим график

Здесь в функцию ggplot передается data в качестве данных для визуализации, далее добавляется слой geom\_bar() для отрисовки барплота. В качестве эстетик поставим x = «» (чтобы собрать все столбцы в один) и fill = Gender. Поскольку это эстетики, они обозначаются внутри функции параметра mapping = aes() или просто внутри функции aes(). По умолчанию, geom\_bar() имеет статистику “count,” что нас устраивает. Далее меняем систему координат на полярную (поскольку пайчарт – барплот в полярной системе координат), убираем все лишние элементы подложки с помощью минималистичной темы theme\_void() и добавляем название графика в labs. Поскольку функция ggplot() не просто отрисовывает график, а создает объект класса ggplot, который можно сохранить и модифицировать в дальнейшем, то можно присвоить полученное выражение переменной plot0.



Видно, что число представителей мужского и женского пола примерно одинаково.

Аналогичным образом построим круговую диаграмму для курильщиков, заполнив fill=Smoker

plot00 <- ggplot(data = data) +

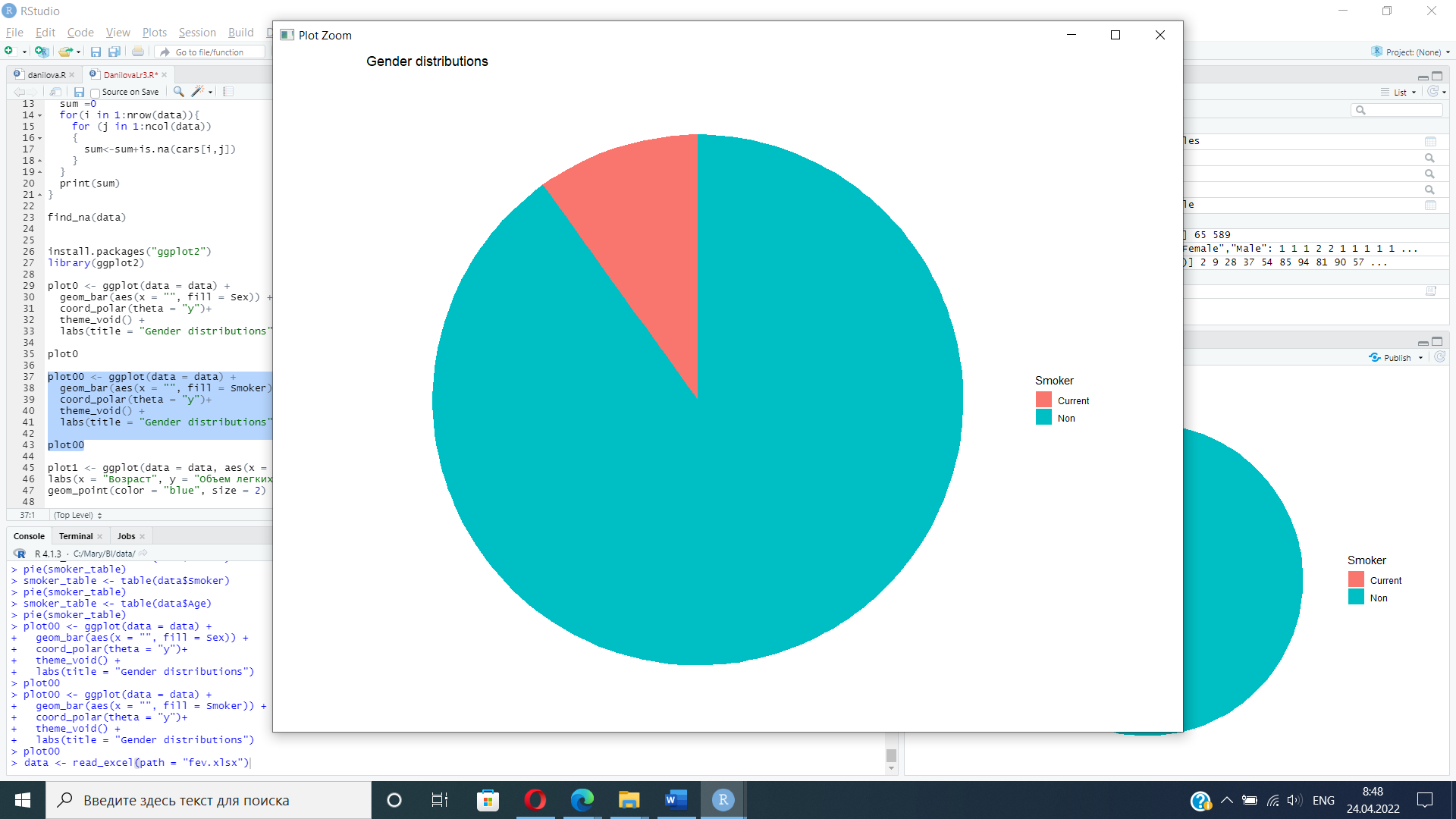
geom\_bar(aes(x = "", fill = Smoker)) +

coord\_polar(theta = "y")+

theme\_void() +

labs(title = "Gender distributions")

plot00



Из диаграммы видно, что некурящих детей в возрасте до 22 лет меньше, чем куряющих.

Исследуем зависимость между возрастом и ОФВ подростков в предположении, что она присутствует (чем старше ребенок, тем больше у него объем легких). Для этого построим точечную диаграмму, где по оси OX отложен Age, а по оси OY отложен FEV.

plot1 <- ggplot(data = data) +

geom\_point(mapping=aes(x = Age, y = FEV), color = "blue", size = 2)+

theme\_dark() +

labs(x = "Возраст", y = "ОФВ",title = "Зависимость между возрастом и ОФВ")

plot1

В функцию ggplot() передается обязательный аргумент data – набор данных.

Функция geom\_point() накладывает следующий слой, который состоит из точек. Эта функция имеет обязательный аргумент mapping, в котором указаны оси (переменные) для графика, здесь же задается цвет и размер точек. С помощью функции labs добавлено название графика (title), название осей (x и y).



Из графика видно, что догадка о связи возраста и ОФВ подтвердилась – чем старше ребенок, тем выше максимальный объём форсированного выдоха; при этом в пределах одного возраста ОФВ отличается максимум на 4 литра, сам показатель варьируется от 0 (не включая) до 6 л.

Аналогичная тенденция и у графика зависимости роста и ОФВ (строится аналогично, тема заменена на theme\_light() и применена другая форма точек)

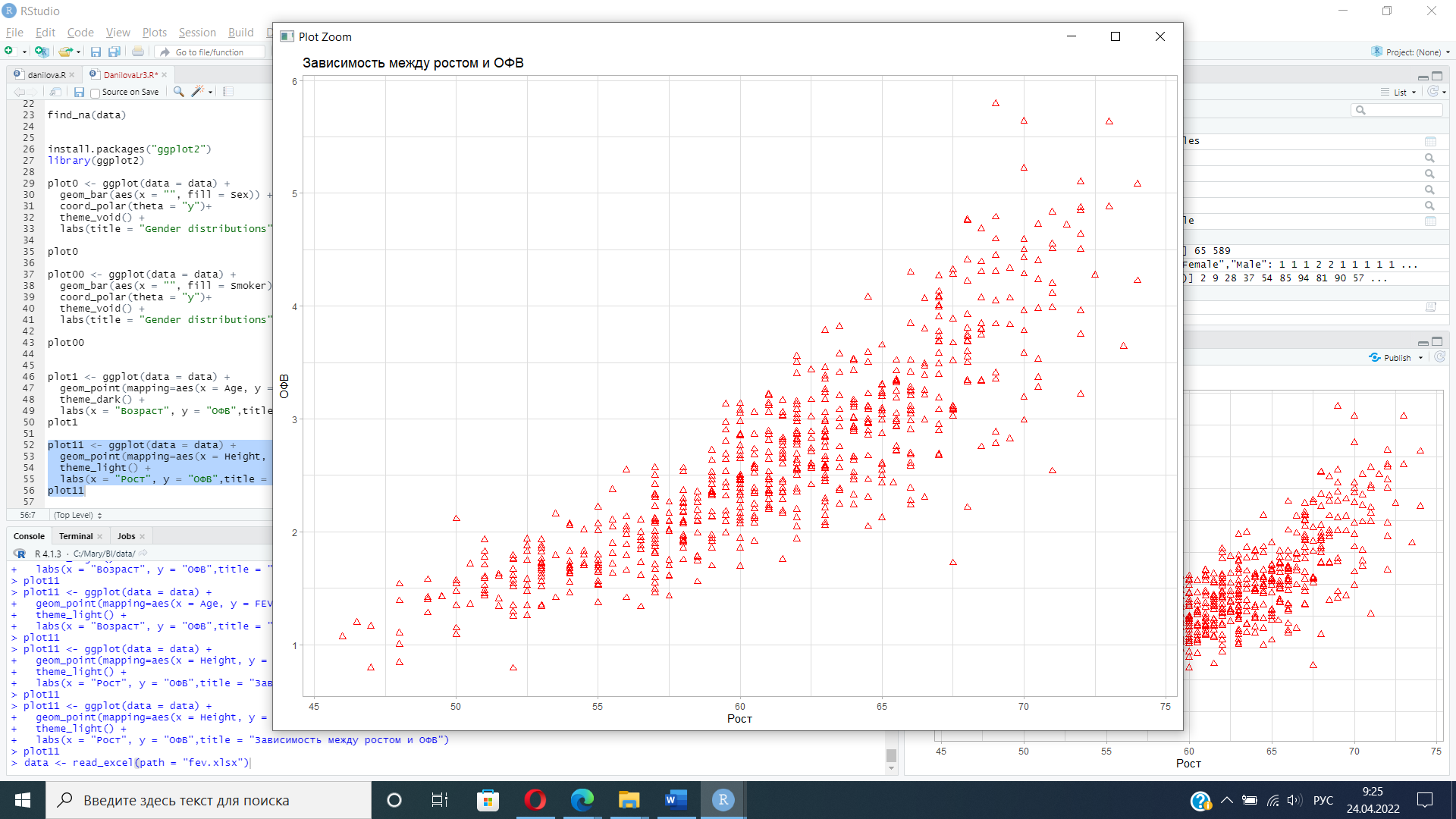
plot11 <- ggplot(data = data) +

geom\_point(mapping=aes(x = Height, y = FEV), color = "red", size = 2, shape=2)+

theme\_light() +

labs(x = "Рост", y = "ОФВ",title = "Зависимость между ростом и ОФВ")

plot11



Также можно отметить, что рост исследуемых меняется в диапазоне 45-75 дюймов.

Однако такое отображение недостаточно информативное – можно добавить визуализацию пола и принадлежности исследуемого к группе курильщиков.

plot12 <-ggplot(data = data) + geom\_point(mapping=aes(x = Age, y = FEV, color = Smoker, shape = Sex ),size=2)+

labs(x = "Возраст", y = "ОФВ", title = "Зависимость между возрастом и ОФВ в зависимости от пола и поля Smoker")+

theme\_light() +

facet\_grid(Sex ~ Smoker)

plot12

На этот раз в функцию geom\_point() передается аргумент mapping, в котором указаны ось x = возраст, ось у=ОФВ для графика, цвет задается в зависимости от принадлежности к группе курильщиков, форма в зависимости от пола (по умолчанию треугольники для женщин и круги для мужчин), а также размер точек. С помощью функции labs добавлено название графика (title) и осей (x и y). theme\_light() задает светлую тему фона. В данном случае мы хотим нарисовать отдельные графики для каждой категории факторной переменной. Делим один сложный график на несколько простых с помощью двух факторных переменных с помощью функции facet\_grid. Аргументом является формула: rows ~ columns. Необязательно приводить переменные к факторам, ggplot сам их принудительно приведет. Если используем только одну категориальную переменную, то вторую обозначаем просто точкой (.). В данном случае в качестве rows выступает пол, в качестве columns – является ли курильщиком или нет.



Из графика видно, что девочки начинают курить в возрасте 10 лет, мальчики немного раньше. В большей степени ОФВ вследствие курения меняется у мальчиков, начиная с 13 лет (курение приводит к уменьшению ОФВ), на ОФВ девочек курение влияет несильно. Также видно, что ОФВ мальчиков-курильщиков не достигает значения максимального значения в 6 л и находится в пределах 2-5 литров.

Такое отображение достаточно информативное, но не очень удобное для анализа, поэтому построим зависимость среднего ОФВ от пола и возраста.

plot13 <- ggplot(data=data) +

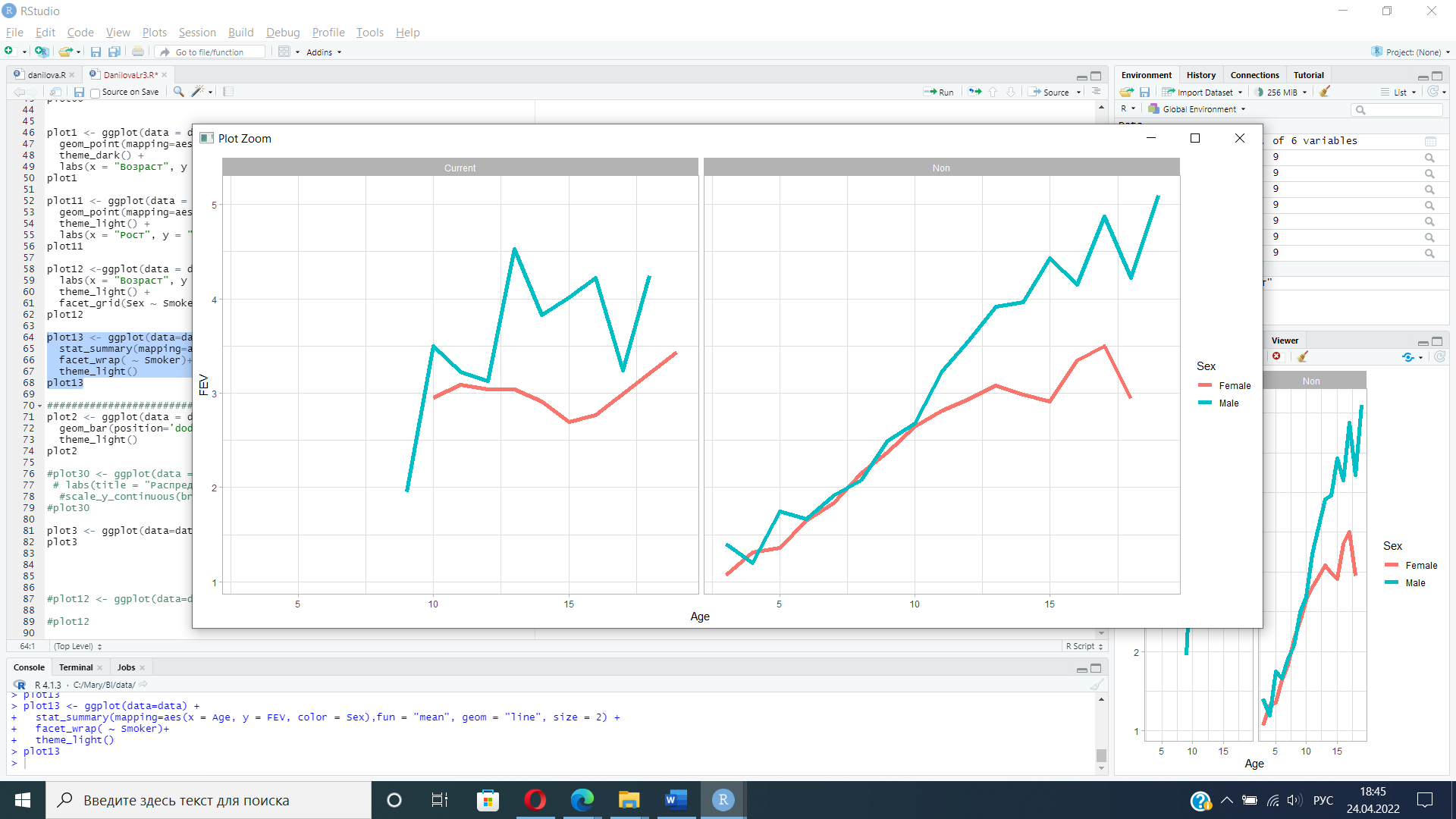
stat\_summary(mapping=aes(x = Age, y = FEV, color = Sex),fun = "mean", geom = "line", size = 2) +

facet\_wrap( ~ Smoker)+

theme\_light()

plot13

Здесь в функцию ggplot передается массив данных, функция stat\_summary() настроена для вычисления сводки (в данном случае среднего) для переменной, указанной в ее аргументе y. В данном случае mapping задает оси х и у (возраст и ОФВ), цвет в зависимости от пола, в аргументе fun указывается, что необходимо построить среднее значение, geom задает тип визуализации – линия. Далее добавлены слои фасетки facet\_wrap по типу курильщика и светлая тема theme\_light.



Видно, что среднее ОФВ зависит от возраста линейно с небольшими скачками для некурящих подростков. По этой же визуализации, как и по предыдущей, видно, что курение негативно сказывается на среднее ОФВ мальчиков старше 13 лет.

Теперь ответим на вопросы – курильщиков какого пола и в каком возрасте больше всего.

Для ответа на 1 вопрос построим диаграмму (geom\_bar). Диаграмма строится для одной дискретной переменной, те принимающей конечное число значений (факторной).

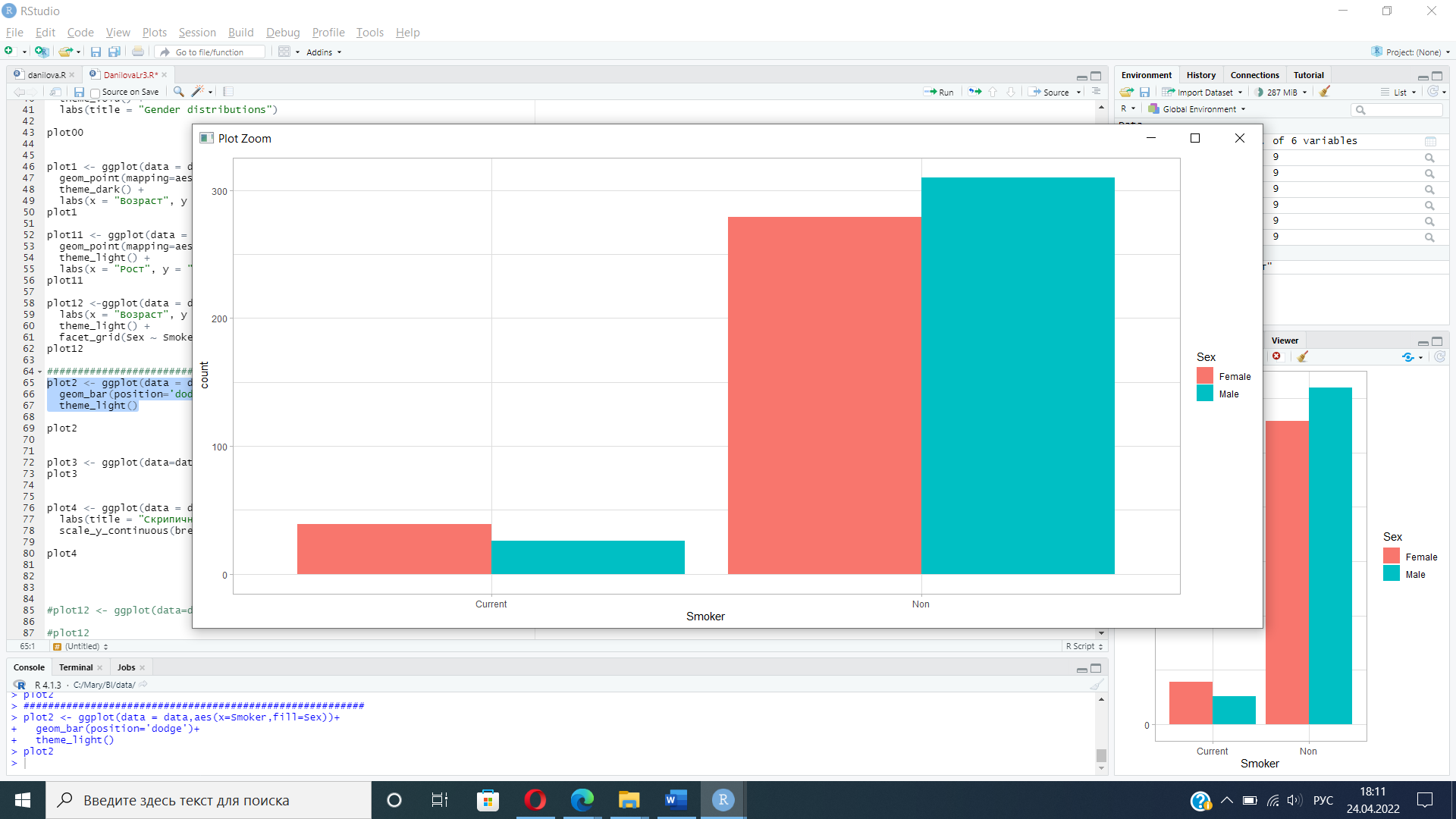
plot2 <- ggplot(data = data,aes(x=Smoker,fill=Sex))+

geom\_bar(position='dodge')+

theme\_light()

plot2

В функцию ggplot() помимо набора данных передается aes, в котором указана ось x по полю Smoker. Для закрашивания фигуры используется атрибут fil, в который передается пол исследуемого. По каждой переменной поля сделан отдельный столбец с помощью аргумента position.

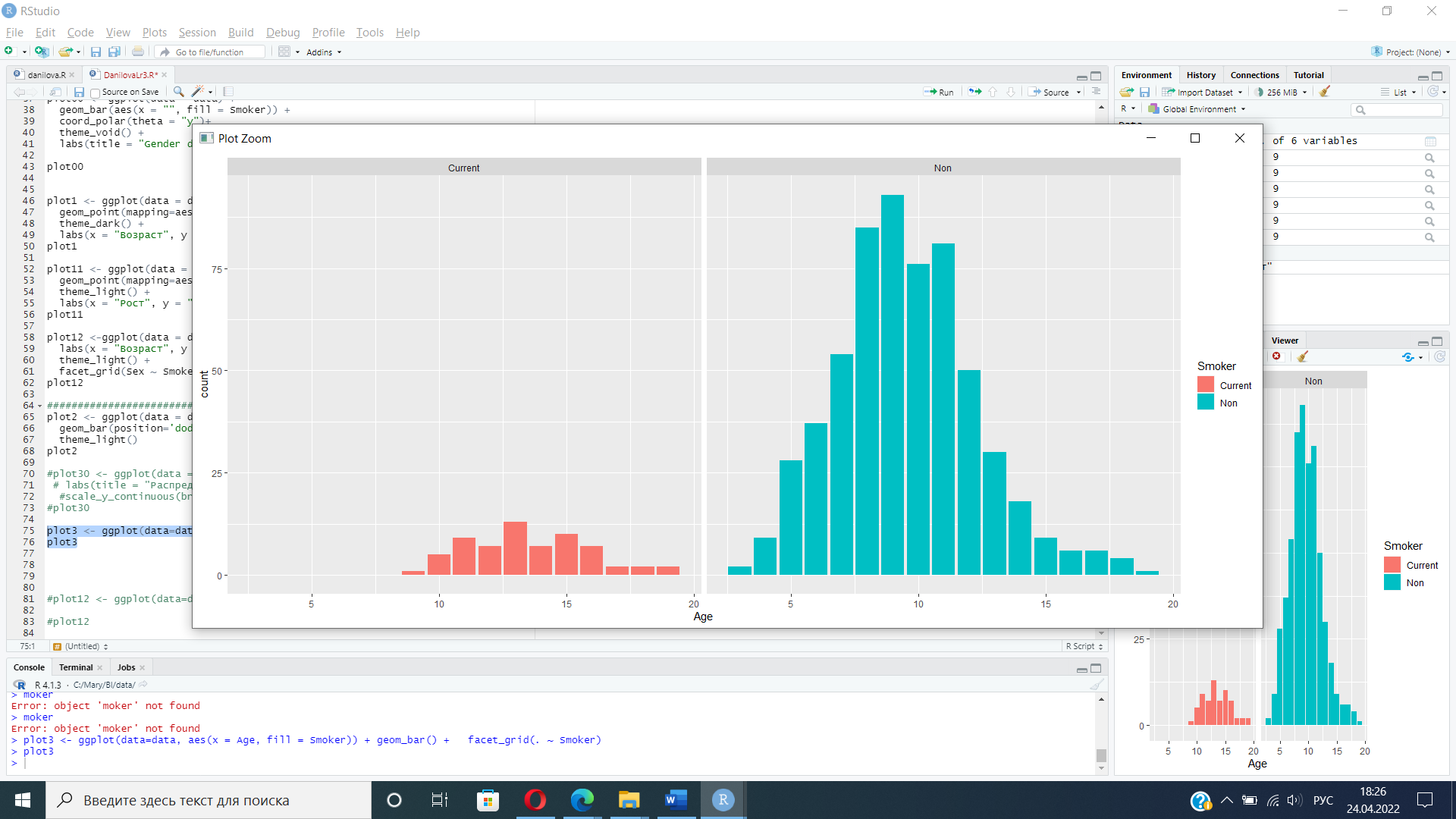


Из диаграммы видно, что среди курильщиков преобладают девочки. .

plot31 <- ggplot(data=data, aes(x = Age, fill = Smoker)) + geom\_bar() + facet\_grid(. ~ Smoker)

plot31

Диаграмма строится аналогично описаниям выше, по оси х откладывается возраст, цвет – в зависимости от принадлежности к группе курильщиков, также добавлена фасетка по группе курильщиков.

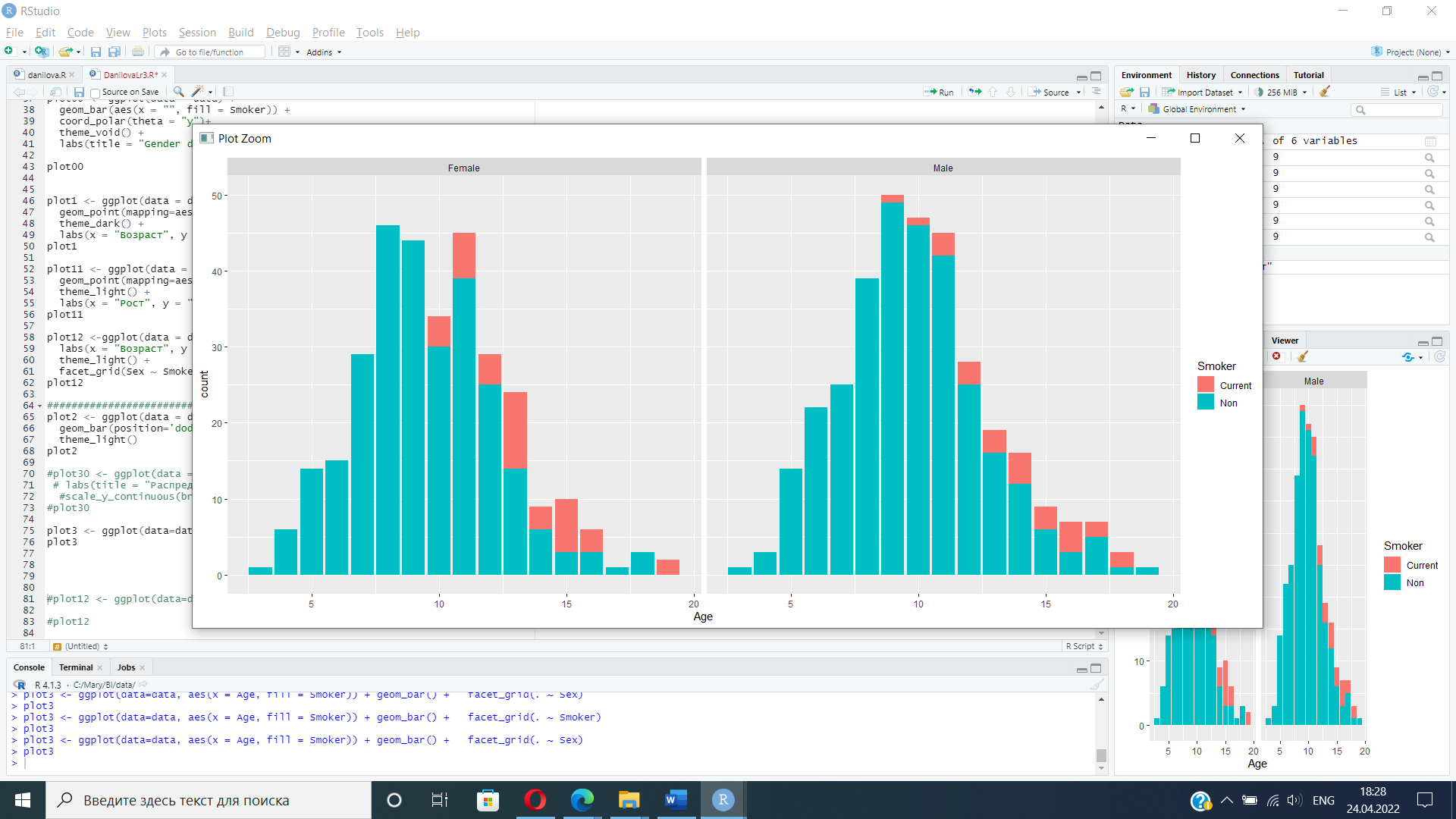


Вывод: больше всего курильщиков в возрасте 13 лет.

Рассмотрим эту же диаграмму в зависимости от пола-добавим фасетку по полу.

Plot32 <- ggplot(data=data, aes(x = Age, fill = Smoker)) + geom\_bar() + facet\_grid(. ~ Sex)

plot32



Из этой диаграммы следует, что на количество курильщиков в 13 лет в основном влияют девочки – их больше.

Общие выводы:

* число представителей мужского и женского пола примерно одинаково;
* некурящих детей в возрасте до 22 лет меньше, чем курящих;
* чем старше некурящий ребенок, тем, как правило, выше максимальный ОФВ, сам показатель варьируется от 0 (не включая) до 6 л;
* ОФВ курильщиков не превышает объема 6 л;
* аналогичная тенденция и у графика зависимости роста и ОФВ, рост исследуемых меняется в диапазоне 45-75 дюймов
* девочки начинают курить в возрасте 10 лет, мальчики немного раньше;
* в большей степени ОФВ вследствие курения меняется у мальчиков, начиная с 13 лет (курение приводит к уменьшению ОФВ);
* среди курильщиков преобладают девочки;
* больше всего курильщиков в возрасте 13 лет.

**Визуализация сетевых данных**

Для визуализации сетевых данных воспользуемся данными, представленными на сайте: <https://www.kaggle.com/datasets/rahulgoel1106/network-analysis-data-from-various-sources>

Это пример набора данных, который содержит справочную информацию из различных источников информации, таких как телевидение, газеты и онлайн-статьи.

Необходимо рассмотреть структуру представленных данных.

InputFileEdges.csv содержит информацию о ребрах между узлами. Поля в этом наборе данных следующие:

from: идентификатор начального узла ребра

to: идентификатор конечного узла ребра

weight: количество раз, когда они ссылались друг на друга

type: тип связи (гиперссылка или упоминание) между этими узлами

InputFileNodes.csv содержит информацию об узлах. Поля в этом наборе данных следующие:

id: уникальный идентификатор узла

media: название СМИ узла (например, NY Times, Washington Post и т. д.)

media.type: тип средства массовой информации (1 представляет собой газету, 2 представляет собой телевидение и 3 представляет собой Интернет).

type.label: тип СМИ (газета, телевидение и Интернет)

Audience.size: размер аудитории для каждого медиа, тыс. человек

Для чтения csv-файла воспользуемся функцией read.csv, в которую 1 параметром передадим название файла, параметр header=TRUE отвечает за то, что есть ли в файле наименования столбцов

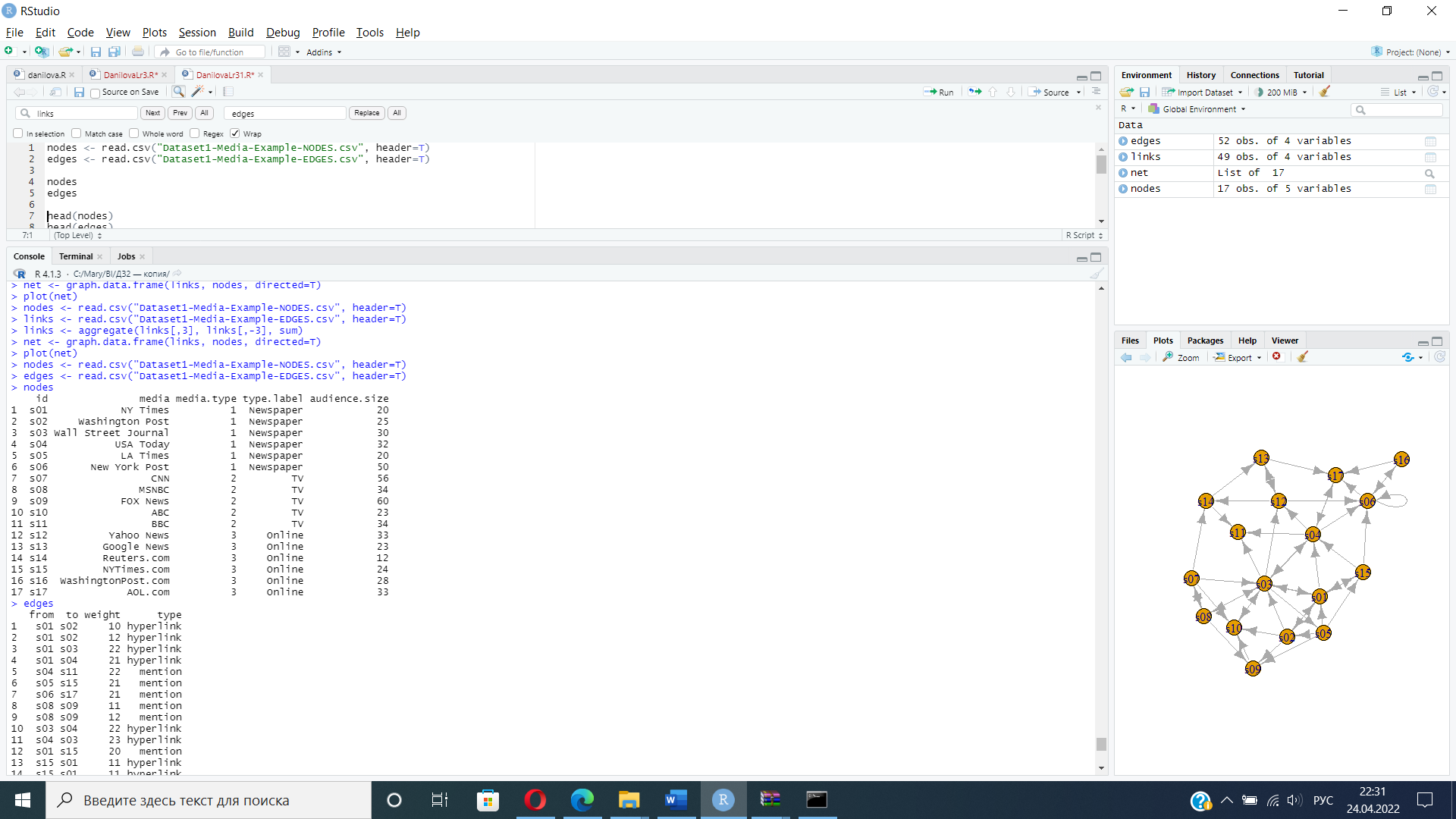
nodes <- read.csv("Dataset1-Media-Example-NODES.csv", header=T)

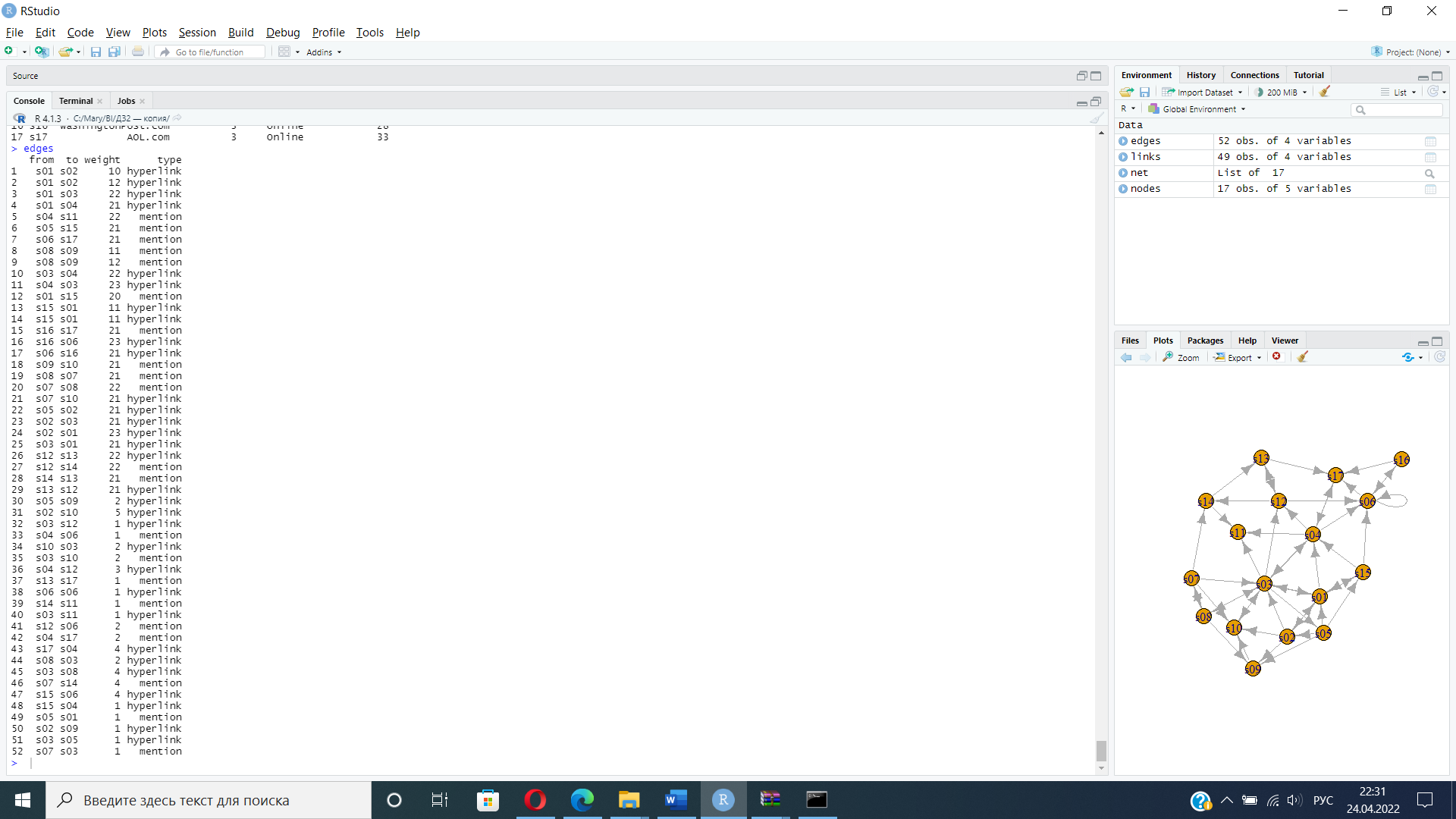
edges <- read.csv("Dataset1-Media-Example-EDGES.csv", header=T)

Посмотрим на считанные данные

nodes

edges





Видно, что в полученных данных нет NA.

Выведем количество строк в списке edges всего и количество уникальных строк

nrow(edges) возвращает количество строк в списке edges

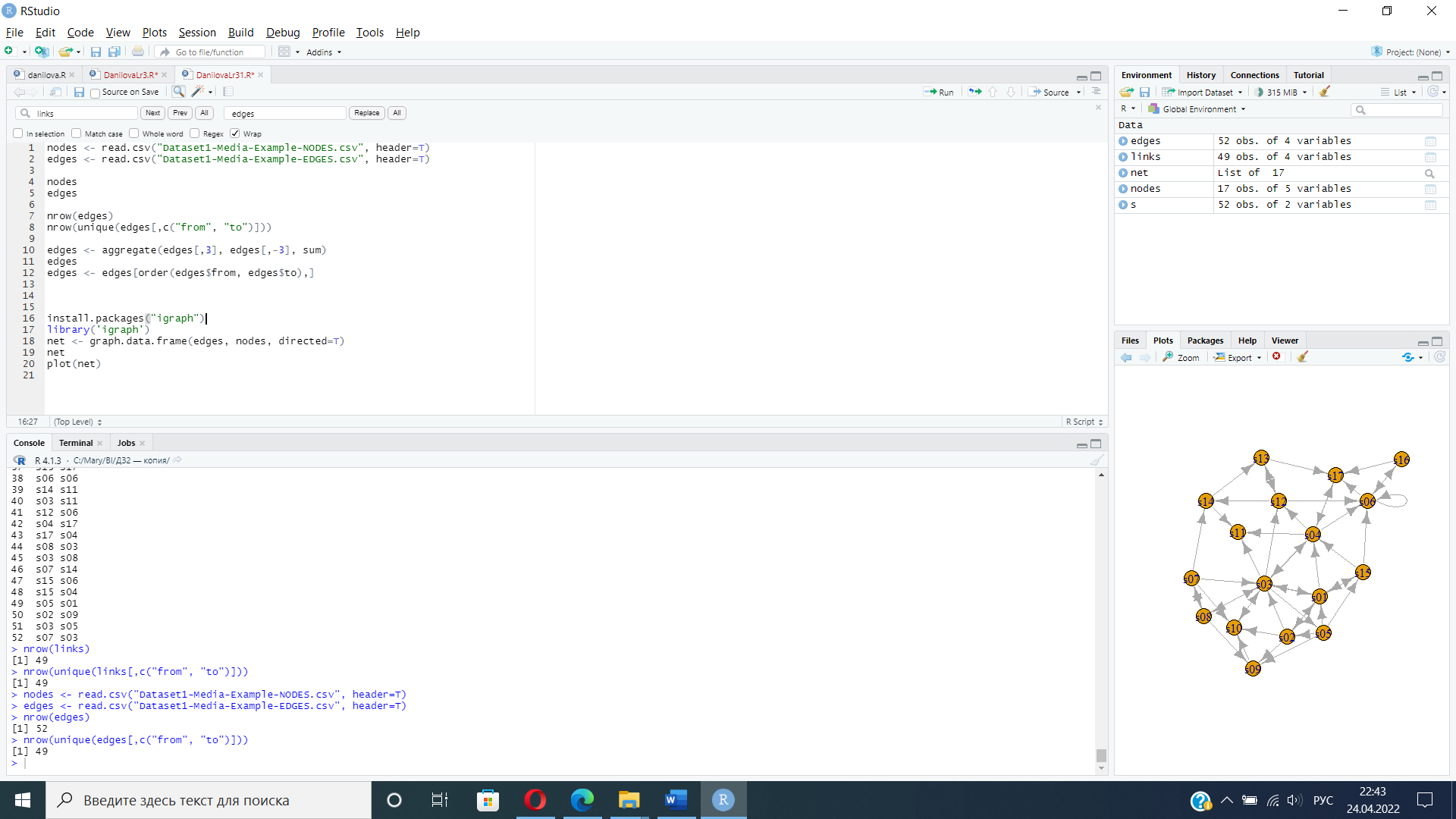
nrow(unique(edges[,c("from", "to")]))

edges[,c("from", "to")]) формирует список, состоящий только из столбцов from, to

unique возвращает вектор, фрейм данных или массив типа, но с удаленными дублирующимися элементами/строками

далее считаем количество строк в этом массиве с помощью той же самой функции nrow

Результат:



Те ребер больше, чем уникальных комбинаций (есть несколько отсылок между одними и теми же узлами), поэтому необходимо просуммировать веса между одинаковыми узлами.

edges <- aggregate(edges[,3], edges[,-3], sum)

colnames(edges)[4] <- "weight"

1 параметр - столбец, который должен быть агрегирован в фрейме данных (3 - веса), 2 параметр - столбец, который будет сгруппирован с помощью функции из параметра 3 (в нашем случае все оставшиеся столбцы). 3 параметр представляет сумму/среднее значение/мин/ макс (в нашем случае сумма).

С помощью функции colnames переименуем 4 столбец в weight

Для визуализации загружаем и устанавливаем необходимый пакет

install.packages("igraph")

library('igraph')

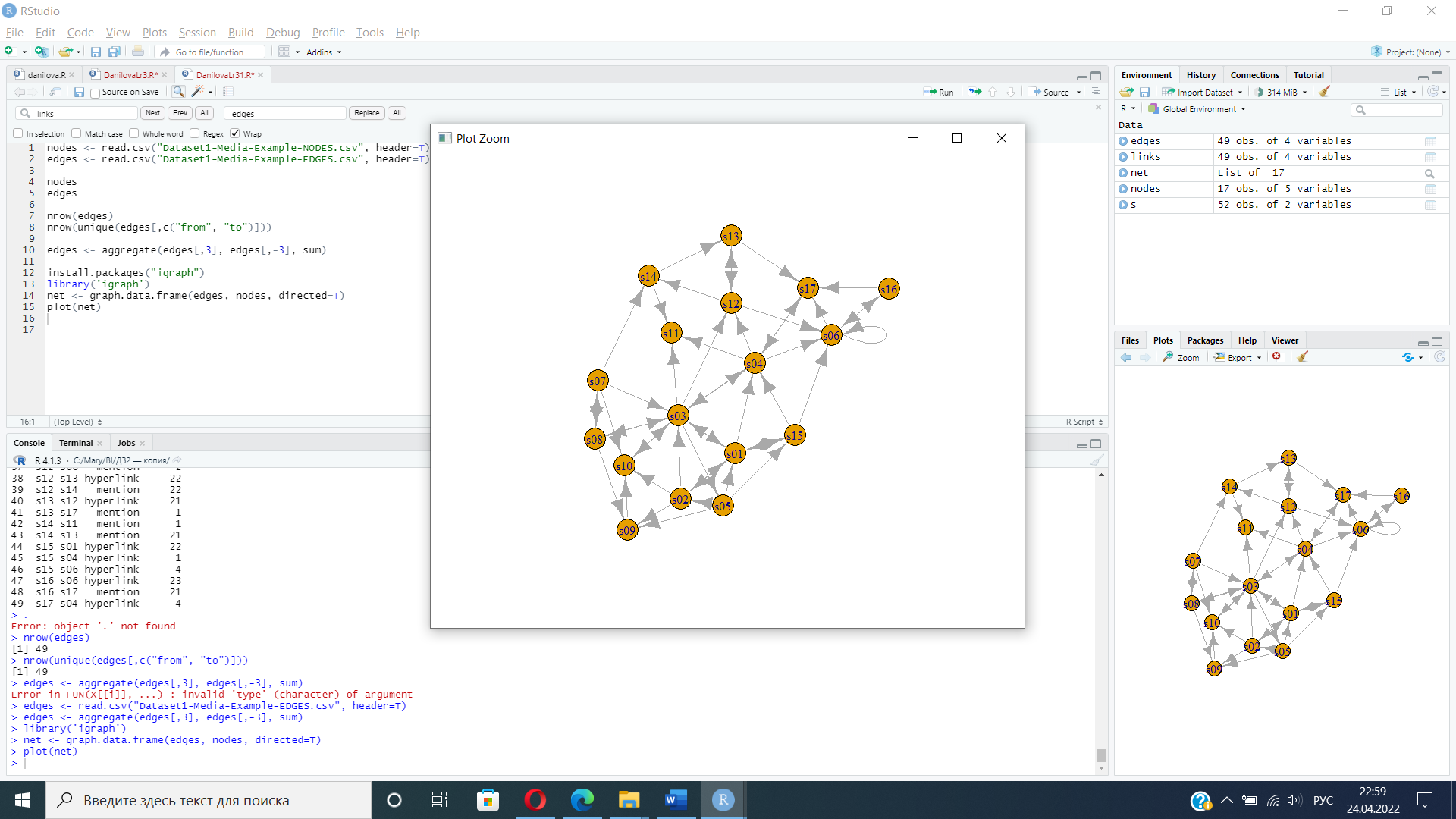
Далее строим график

net <- graph.data.frame(edges, nodes, directed=T)

plot(net)

Для этого используем функцию graph.data.frame, которая принимает на вход два блока данных. Первый описывает ребра сети (в нашем случае edges). В первых двух колонках содержатся идентификаторы начальной и конечной вершины для каждого ребра. В следующих колонках находятся параметры ребра (вес, тип, метка, другое). Второй блок данных nodes начинается с колонки идентификаторов вершин. Все следующие колонки интерпретируются как параметры вершины.

Directed=T указывает на то, что граф будет ориентированным.



Удалим из графа циклы и применим к нему форматирование.

net <- simplify(net, remove.loops = T)

plot(net, vertex.color='green', vertex.size=15, edge.color='blue', edge.arrow.size=.5)

simplify(graph, remove.multiple = TRUE, remove.loops = TRUE)

Аргументы:

graph-График для работы.

remove.multiple - следует ли удалять несколько ребер.

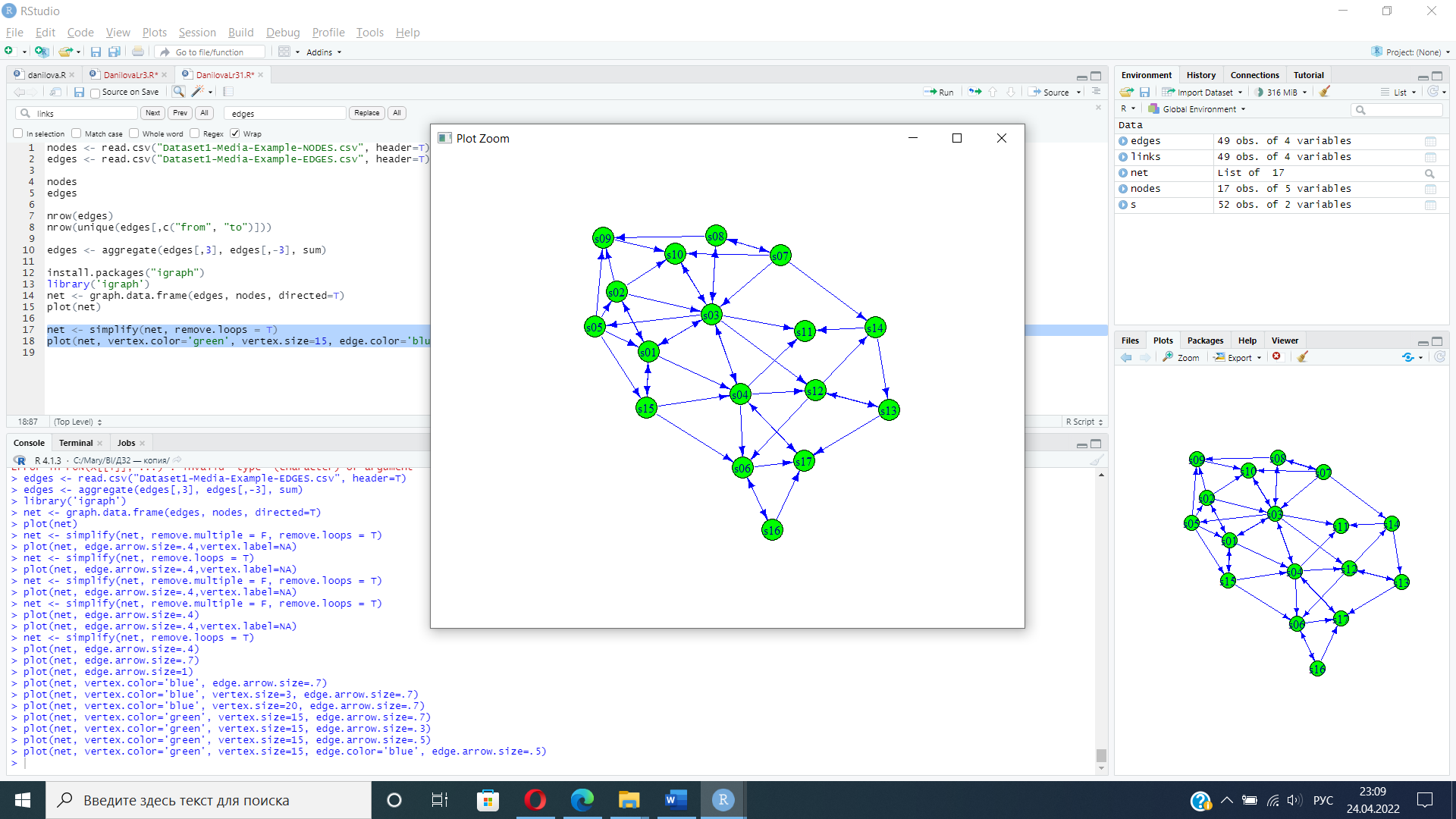
remove.loops - следует ли удалять циклы, в нашем случае true=да.

vertex.color - цвет вершины (зеленый)

vertex.size - размер вершины (по умолчанию 15)

edge.color - цвет ребра (синий)

edge.arrow.size - размер стрелки, по умолчанию 1



Изменим толщину ребра в зависимости от веса (количества ссылок на СМИ), размер вершины – в зависимости от размера аудитории и цвет вершины в зависимости от типа издания.

Можно легко получить доступ к вершинам, ребрам и их атрибутам:

E(net) Ребра объекта "net"

V(net) Вершины объекта "net"

E(net)$type Свойство ребра "type"

V(net)$audience.size Свойство вершины " audience.size "

colour <- c("red", "yellow", "green")

зададим список цветов

V(net)$color <- colour[V(net)$media.type]

Обратимся к атрибуту color вершины, отвечающему за ее цвет. Цвет задается в зависимости от свойства вершины media.type из списка цветов, созданного ранее

V(net)$size <- V(net)$audience.size\*0.5

Обратимся к атрибуту size вершины, отвечающему за ее размер. Присвоим ему значение, равное свойству вершины «размер аудитории», умноженному на коэффициент

E(net)$width <- E(net)$weight/5

Обратимся к атрибуту width ребра, отвечающему за его ширину. Присвоим ему значение, равное свойству ребра «вес», поделенному на коэффициент

plot(net, vertex.label.cex=0.8, edge.arrow.size=0.3, vertex.label.font=2)

vertex.label.cex - размер шрифта (множитель, зависит от устройства)

edge.arrow.size - размер стрелки, по умолчанию 1

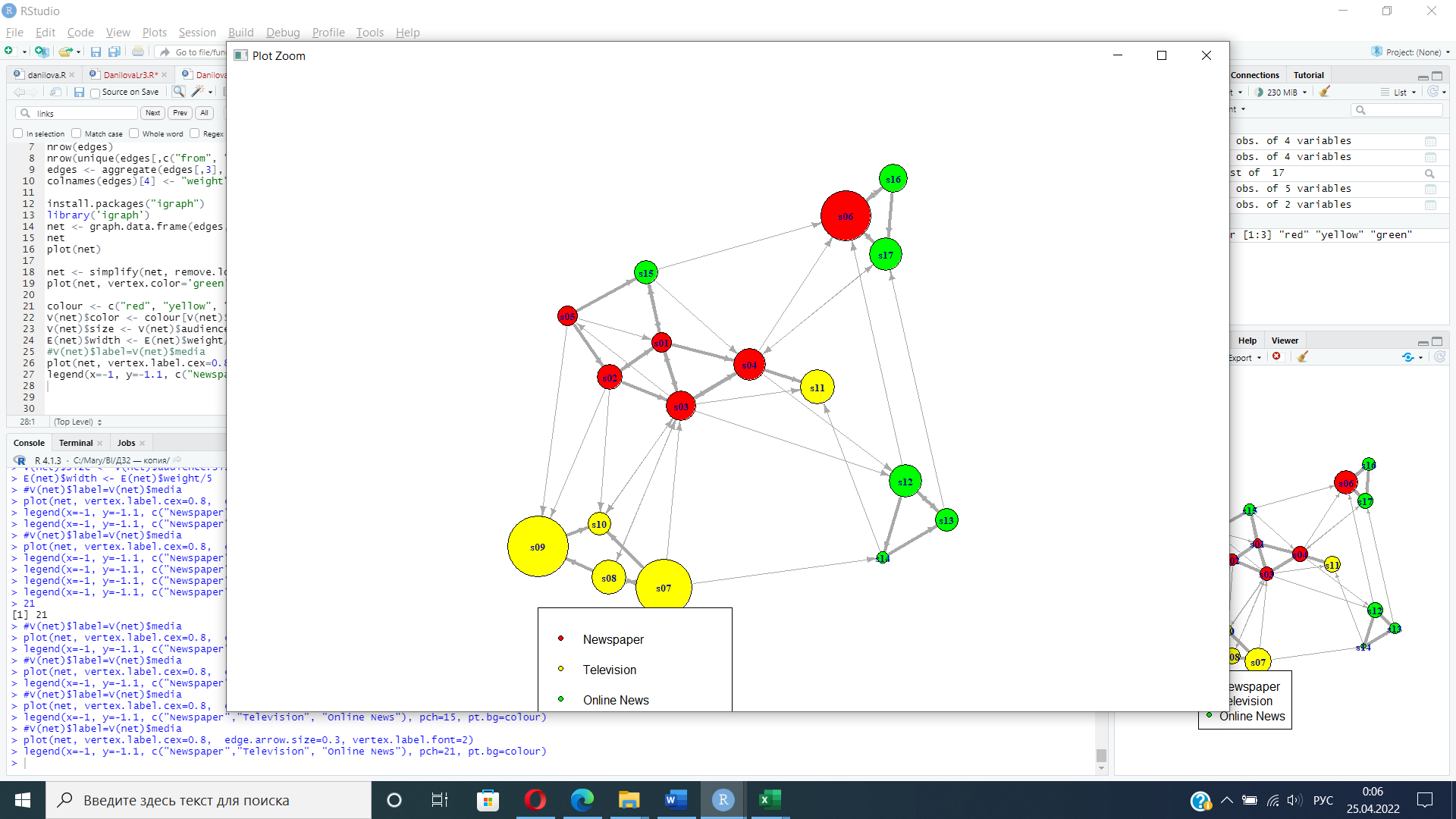
vertex.label.font - шрифт: 1 — обычный, 2 — жирный, 3 — курсив, 4 — жирный курсив, 5 — символьный

legend(x=-1, y=-1, c("Newspaper","Television", "Online News"), pch=21, pt.bg=colour)

x, y - координаты x и y, используемые для позиционирования легенды.

c("Newspaper","Television", "Online News") – названия, отображаемые в легенде

pch = 21, точечные символы, в данном случае заполненный круг

pt.bg - цвет фона для параметра , соответствующий его аргументу  


Из графика видно, что СМИ s09 имеет самую большую аудиторию (самый большой круг), меньше всего представлено телевизионных СМИ (5 желтых кругов против 6 красных и 6 зеленых). Цитируемость того или иного СМИ другими можно определить по толщине ребра.

**Визуализация иерархических данных: карты деревьев**

Для визуализации иерархических данных воспользуемся данными, представленными на сайте: <https://github.com/kwartler/Harvard_DataMining_Business_Student>

Здесь представлены данные почти 10079 аукционов ebay, организованных по категориям товаров, подкатегориям и бренду.

Структура данных:

High Bid средняя цена закрытия, Seller Feedback – рейтинг продавца, Category – категория товара, Sub-Category – подкатегория товара, Brand – бренд.

Задача – определить категорию товаров, которые имели наибольшую среднюю цену закрытия.

Считаем данные и определим, есть ли в них NA значения по аналогии с предыдущими заданиями.

data <- read.csv("TreeMap.csv", header=T)

data

find\_na <- function(data){

sum =0

for(i in 1:nrow(data)){

for (j in 1:ncol(data))

{

sum<-sum+is.na(data[i,j])

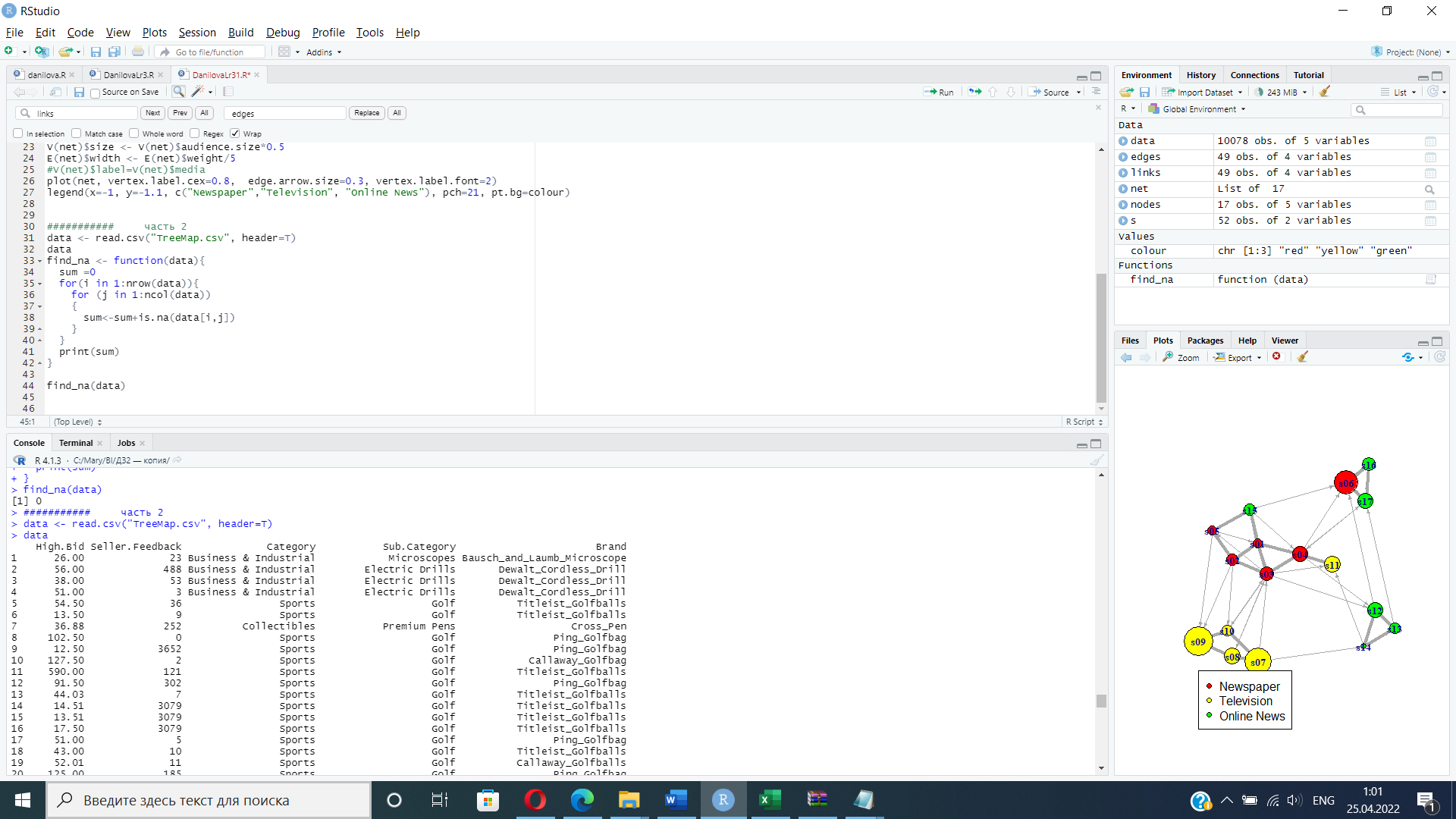
}

}

print(sum)

}

find\_na(data)



Результат: 0. Пропущенных значений нет.

Установим необходимые пакеты

install.packages("treemap")

library(treemap)

Строим карту деревьев

treemap(data, index = c("Category","Sub.Category" , "Brand"),

vSize = "High.Bid",

align.labels = list(c("center", "top"), c("center", "center") , c("center", "bottom")),

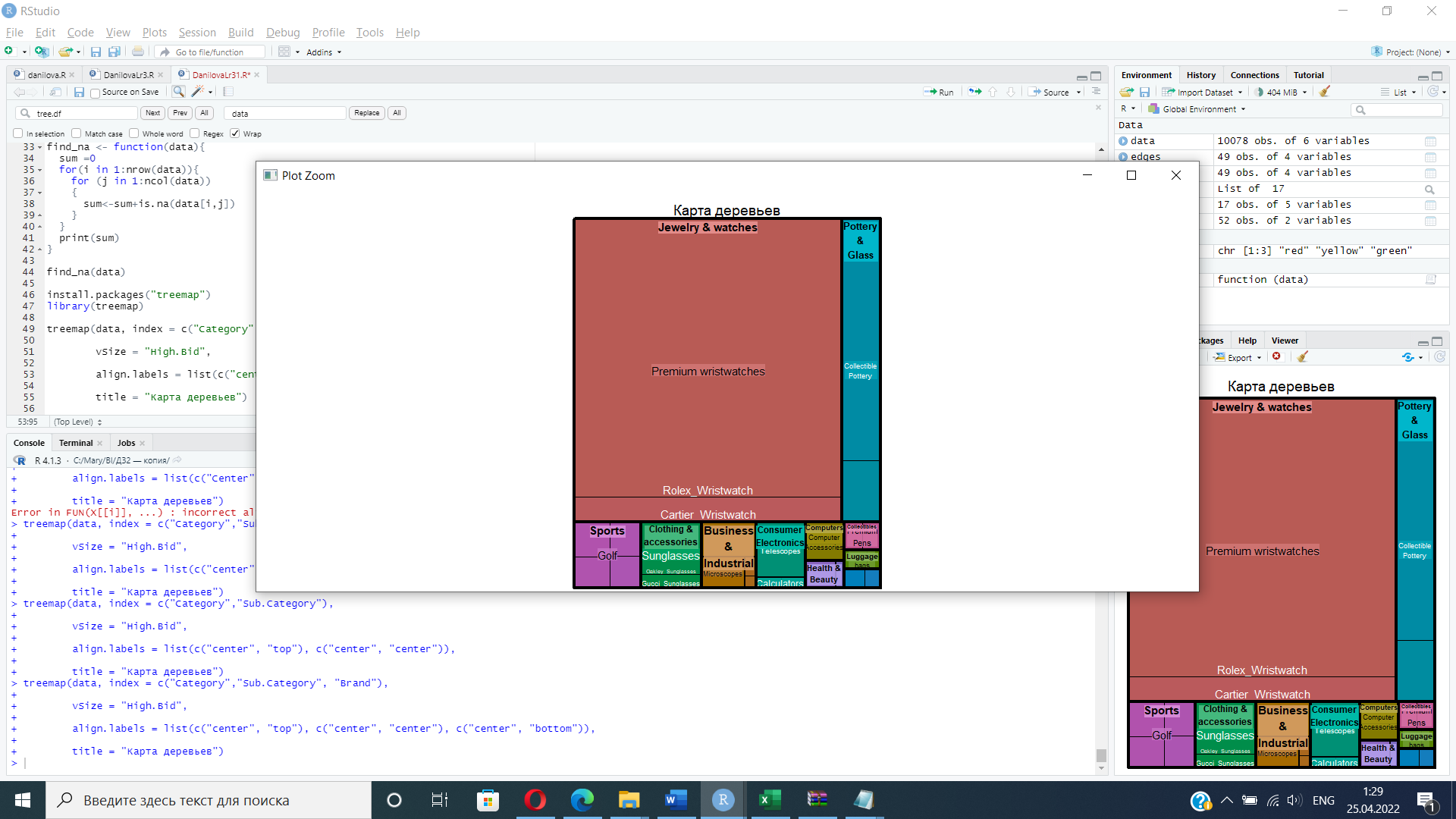
title = "Карта деревьев")

data – набор данных для анализа

index - вектор имен столбцов в data, которые указывают индексы агрегации. Если приведены несколько имен столбцов, первое наименование - самый высокий уровень агрегирования, второе наименование – следующий уровень после самого высокого уровня агрегации и тд. Обязательный параметр

align.labels – параметр, который определяет выравнивание меток. Вектор символов двух значений, указывающих горизонтальное выравнивание («лево», «центр» или «право») и вертикальное выравнивание («верх», «центр» или «низ»). По 1 вектору для каждого уровня агрегации.

Title – название карты деревьев.



Вывод: наибольшую среднюю цену закрытия имели товары категории «Ювелирные изделия и часы» (имеют наибольшую площадь).

**Визуализация географических данных: карта-график.**

Для визуализации географических данных воспользуемся данными, представленными на сайте: <https://github.com/kwartler/Harvard_DataMining_Business_Student>

Здесь представлены ВВП различных стран

Структура данных:

Country Name – название страны

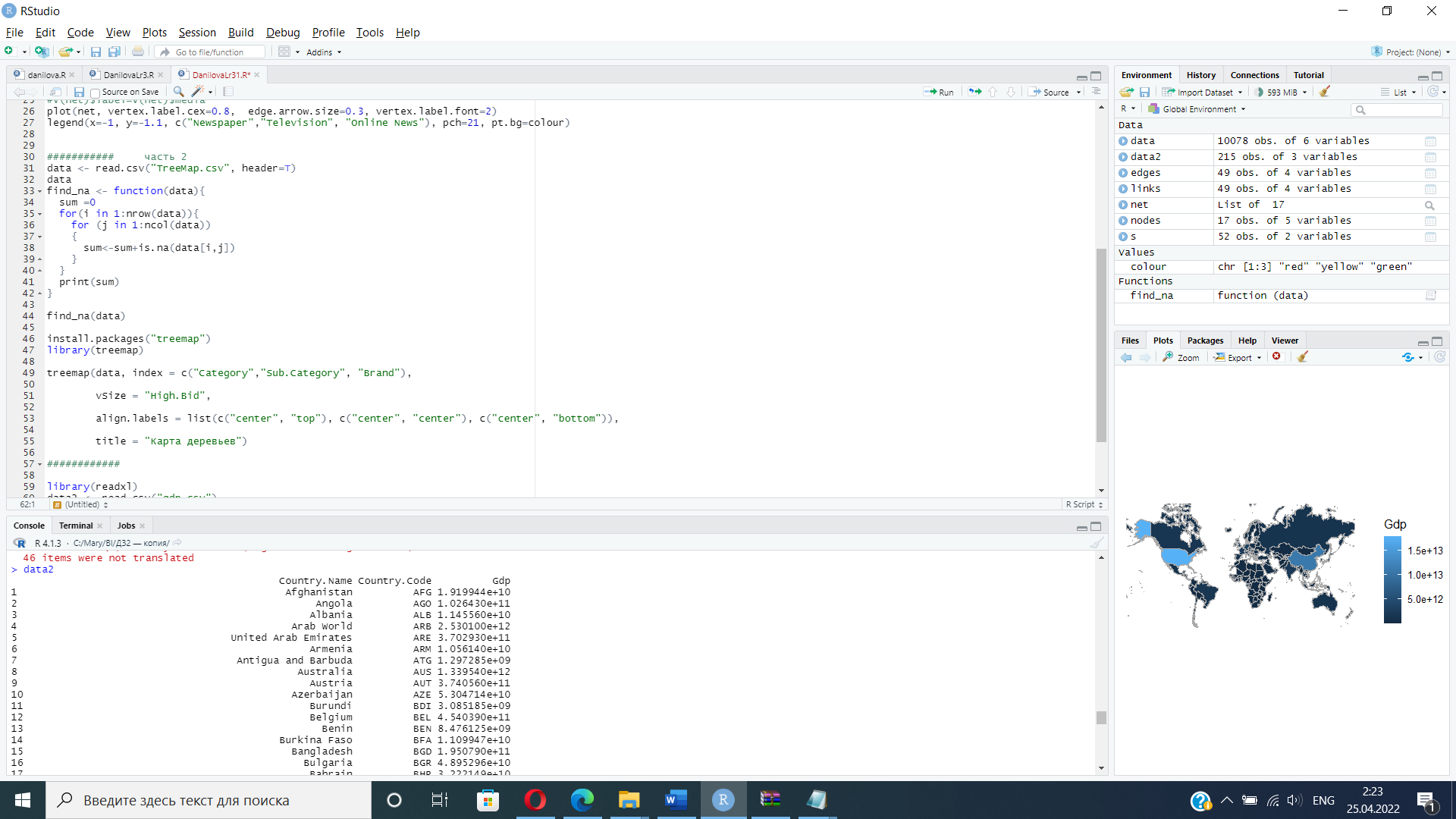
Country Code – код страны

Gdp – ВВП (в долларах)

Считаем данные по аналогии с предыдущими заданиями

data2 <- read.csv("gdp.csv")

data2



Проверим наличие пропущенных значений

find\_na(data2)

Результат: 0

Установим необходимый пакет для визуализации

install.packages("mosaic")

library(mosaic)

Нарисуем карту с ВВП, используя функцию   
mWorldMap(data, key, fill)

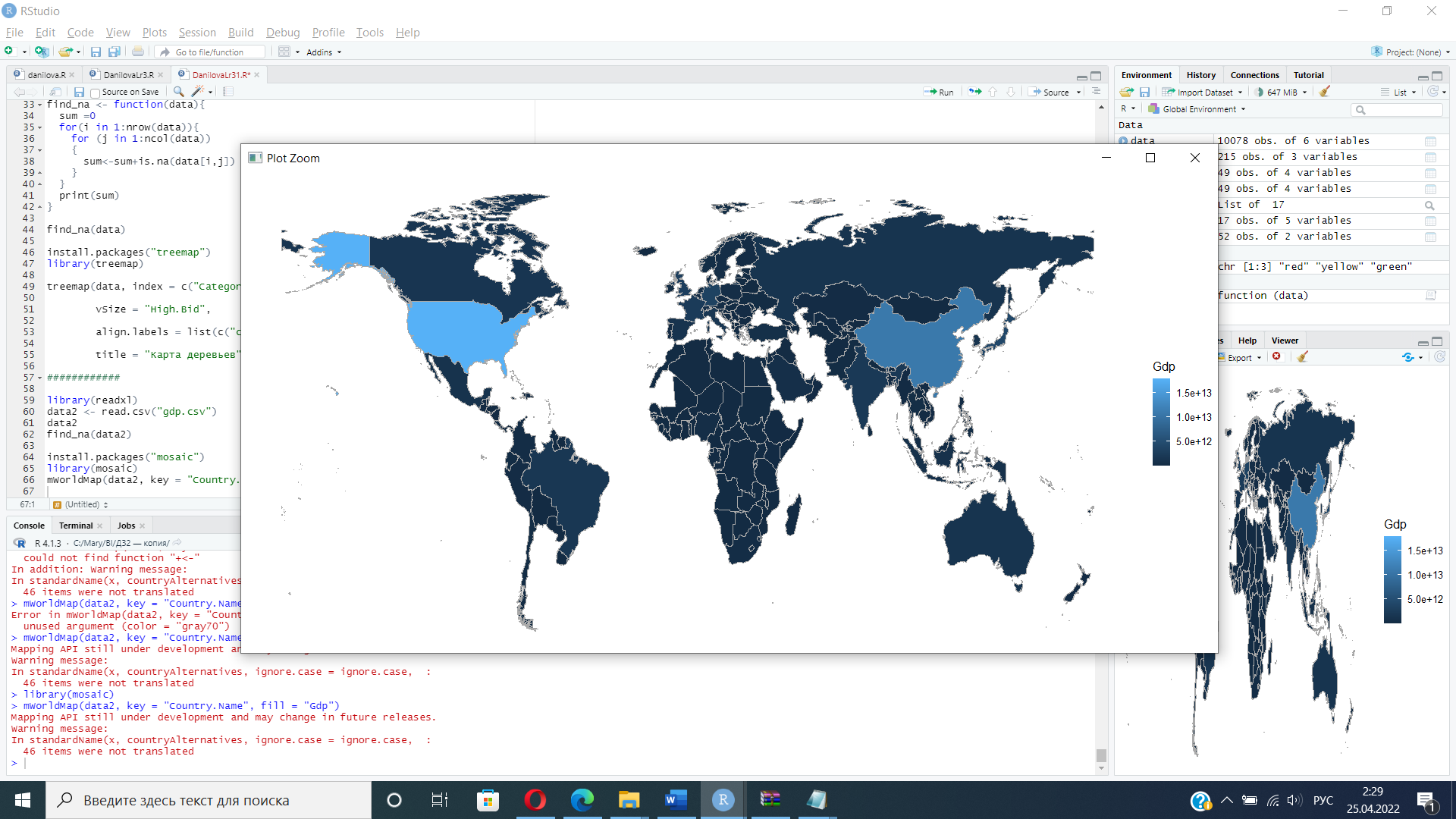
data – исходные данные

key - имя столбца, содержащего уникальные названия каждой страны

fill - Переменная, используемая для указания цвета заливки стран на карте

mWorldMap(data2, key = "Country.Name", fill = "Gdp")

то есть цветовая схема соответствует ВВП страны



Темные цвета означают более низкие значения ВВП, и наоборот. Большее ВВП имеют страны, расположенные в Северной Америке и на юге Евразии.