

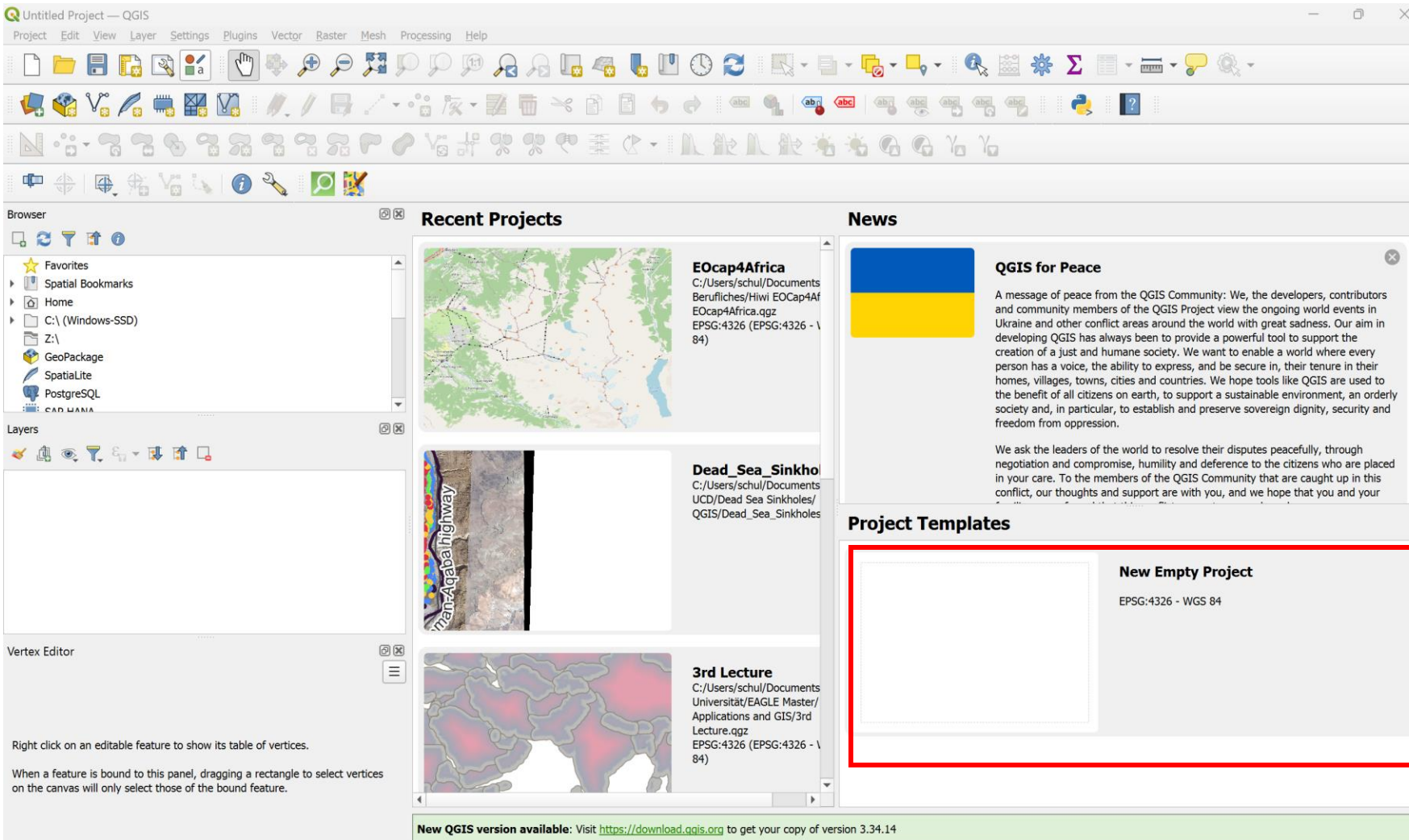
5.1 How to handle Vector Data

After this lesson you can ...

- Load in vector data in QGIS and R
- Visualise vector data in QGIS and R

Vector Data in QGIS

Starting a QGIS Project



- Start your QGIS application
- Open a new Project

QGIS *Untitled Project — QGIS

Project Edit View Layer Settings Plugins Vector Raster Mesh Processing Help

Menu Bar

Tool Bars

Browser

Icons: Add, Refresh, Filter, Up, Info

- ★ Favorites
- ▶ Spatial Bookmarks
- ▶ Home
- ▶ C:\(W
- Z:\
- GeoPa
- Spatialite
- PostgreSQL
- SAR HANA

Layers

Icons: Add, Refresh, Filter, Up, Down, Copy, Paste

Layer panel

The data itself is here

Vertex Editor

Right click on an editable feature to show its table of vertices.

When a feature is bound to this panel, dragging a rectangle to select vertices on the canvas will only select those of the bound feature.

Map view
You will see your data
visualised here

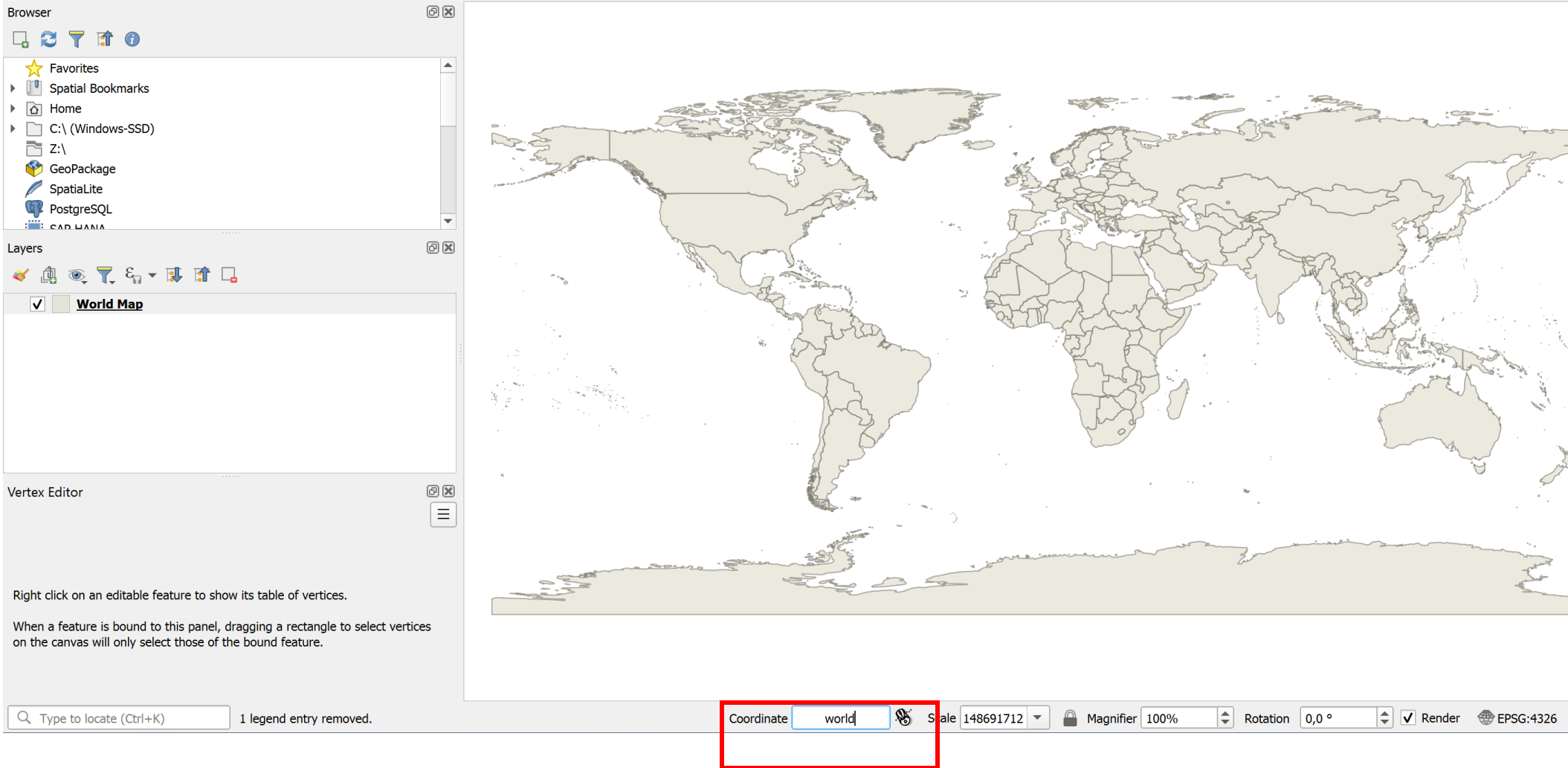
Status Bar

Search: Type to locate (Ctrl+K) | 1 legend entry

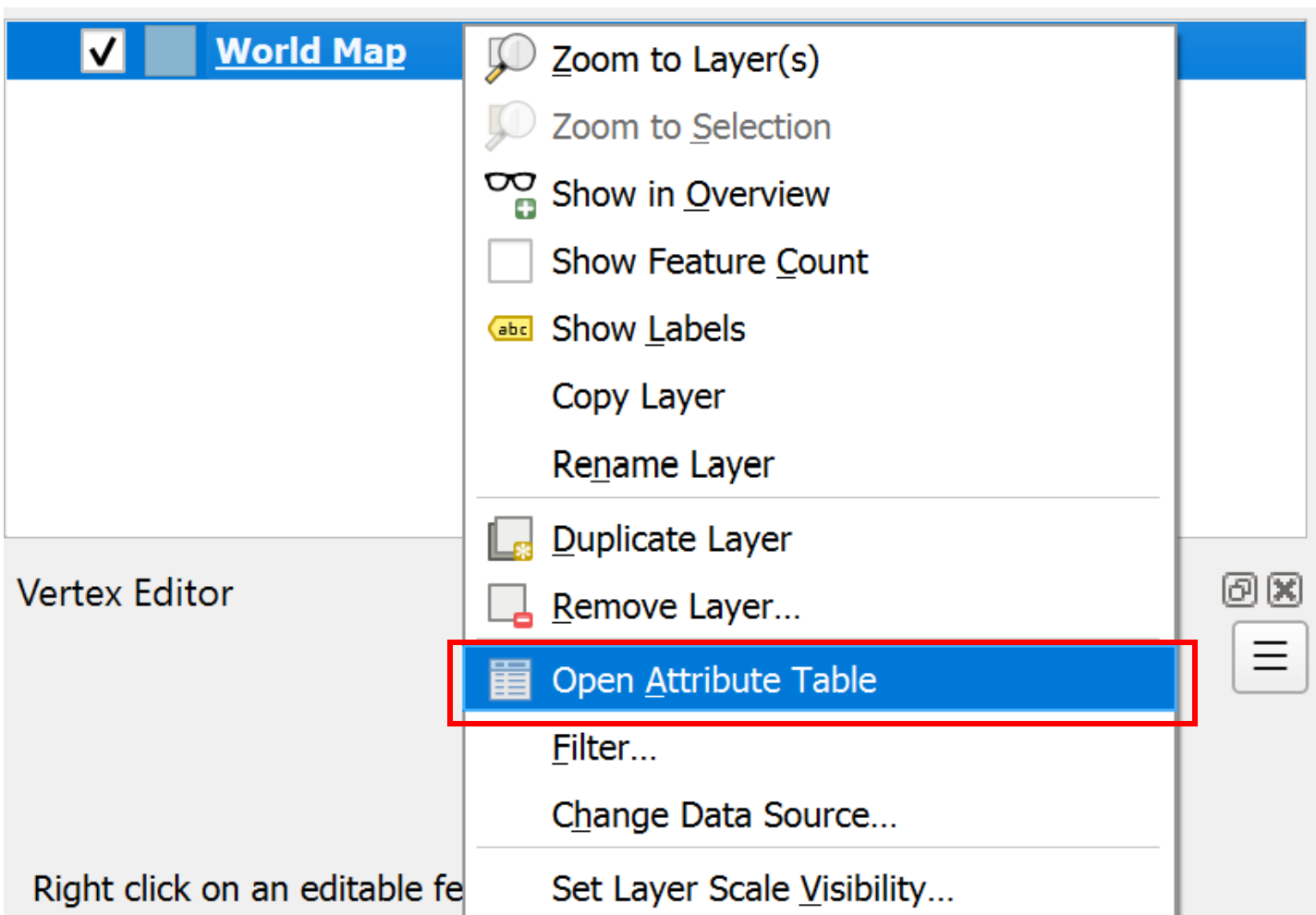
Coordinate: 5° -172,5° | Scale: 143341696 | Magnifier: 100% | Rotation: 0,0 ° | Render: ☒ | EPSG:4326

Access Vector Data

Type World in the Corrdinate field in the status bar to access a vector wolrd map



Attribute Table



Attribute Table

Explore the attribute table!

What data does it contain? Are there different types of data
in the attribute table?

Load in your own Vector Data

1. Download the shapefile and unzip it
2. Move the folder of the shapefile to your desired location
3. Load in the Data

Method 1)

Just pull the .shp file from
your explorer into QGIS

Tip!

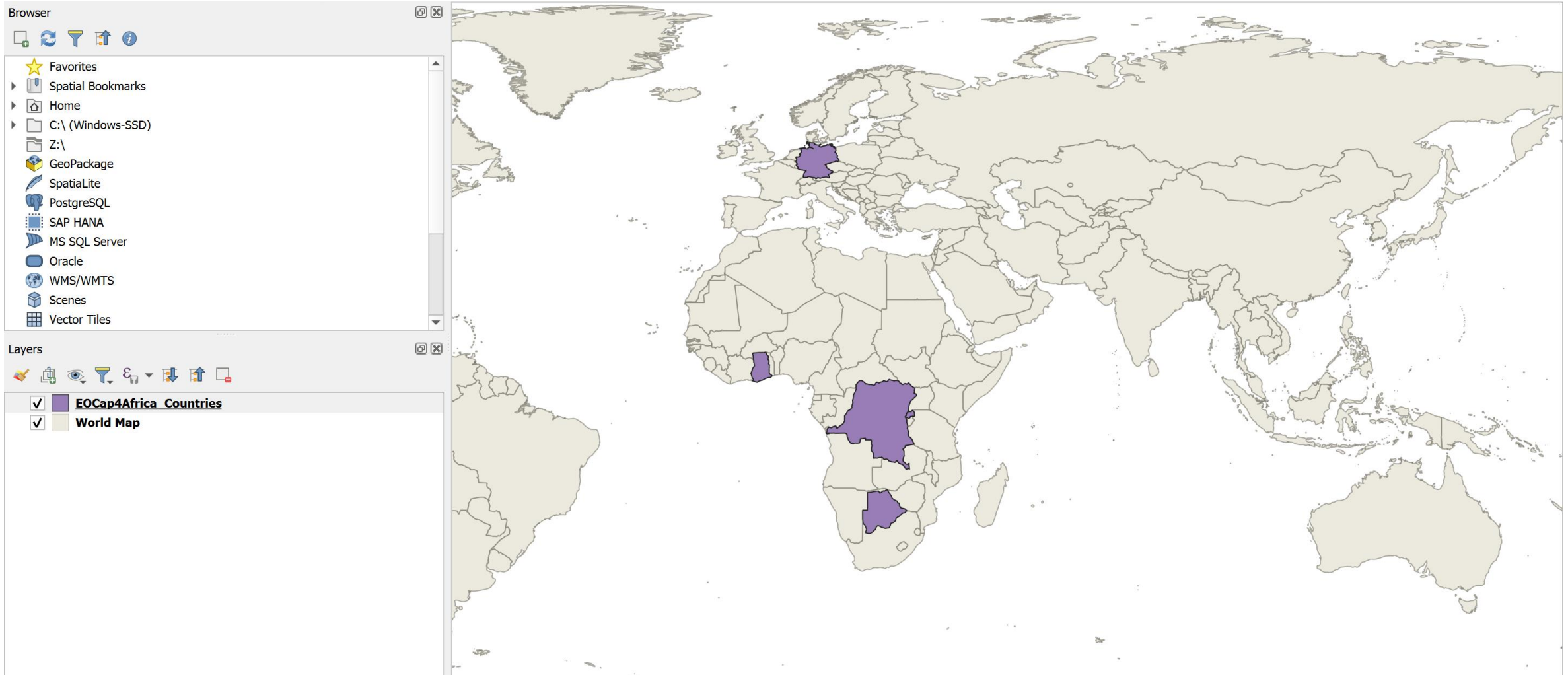
After you have started a project do
not move or rename your files!

Method 2)

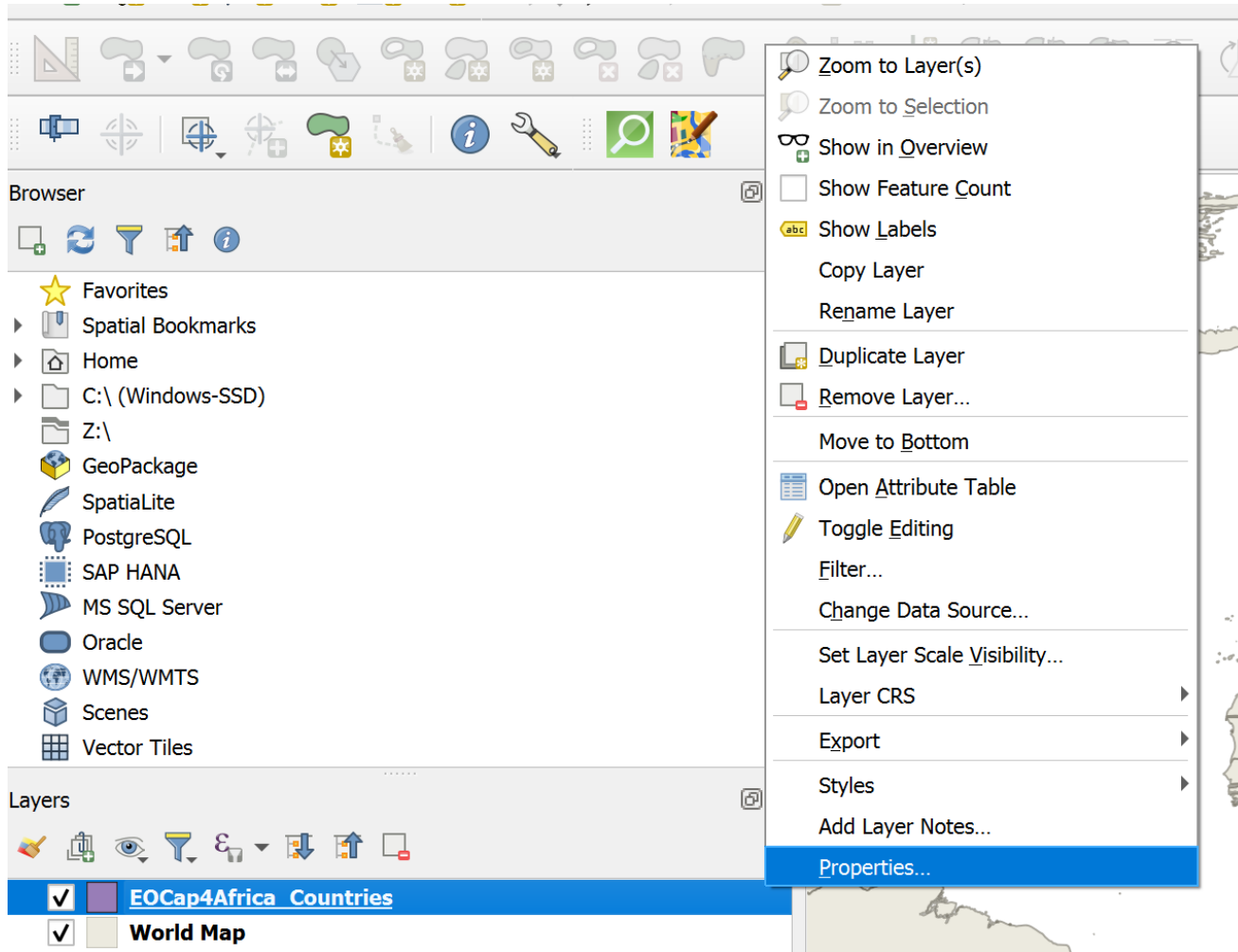
- 1) Go to Layer in the Menu Bar
- 2) Add Layer
- 3) Add Vector Layer
- 4) Go to the shapefile and
doubleclick it

Load in your own Vector Data

The result:



Vector Properties

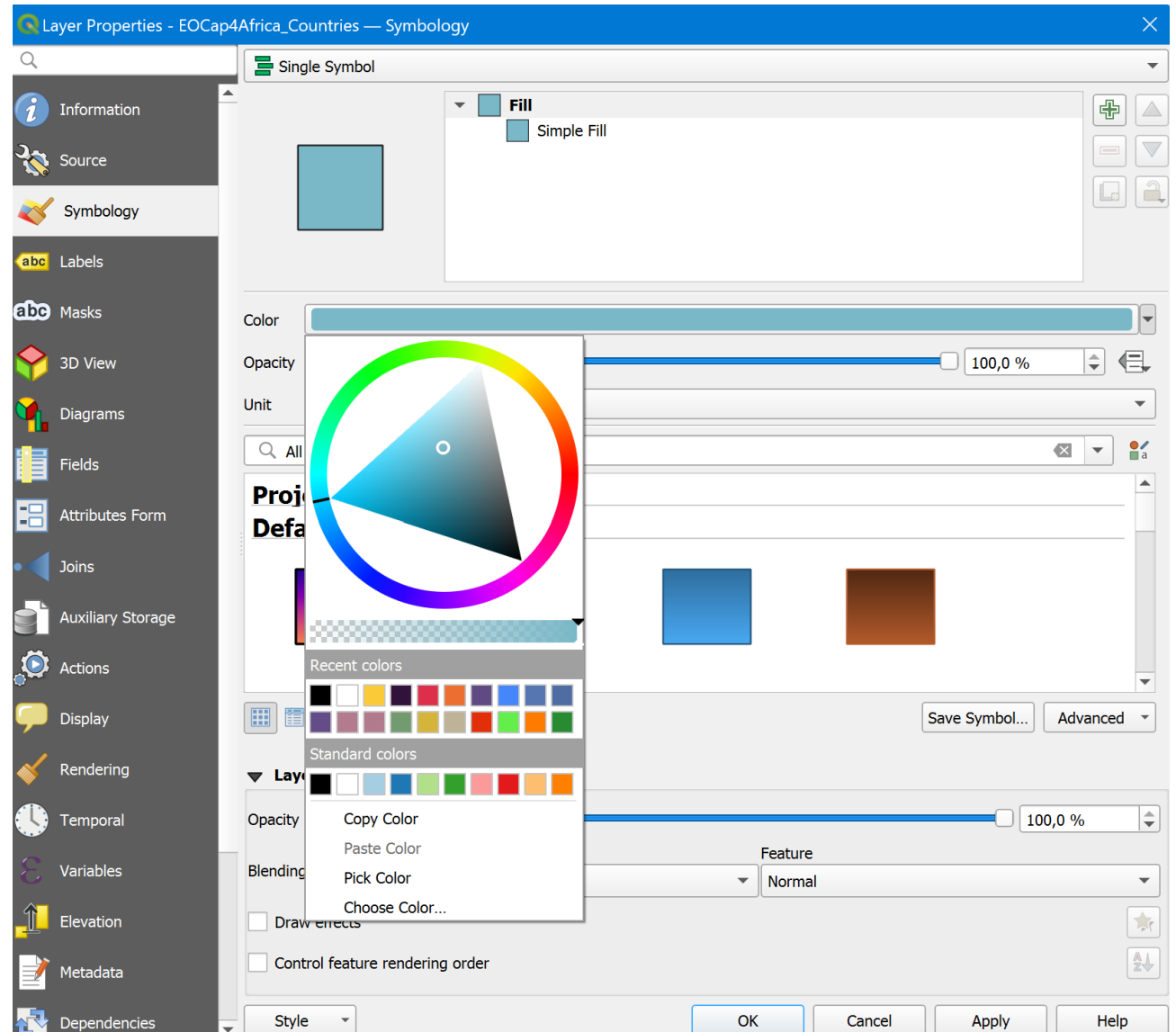


- Rightclick on your new vector layer
- Under properties you can find all sort of information and options to work with your data

Vector Symbolology

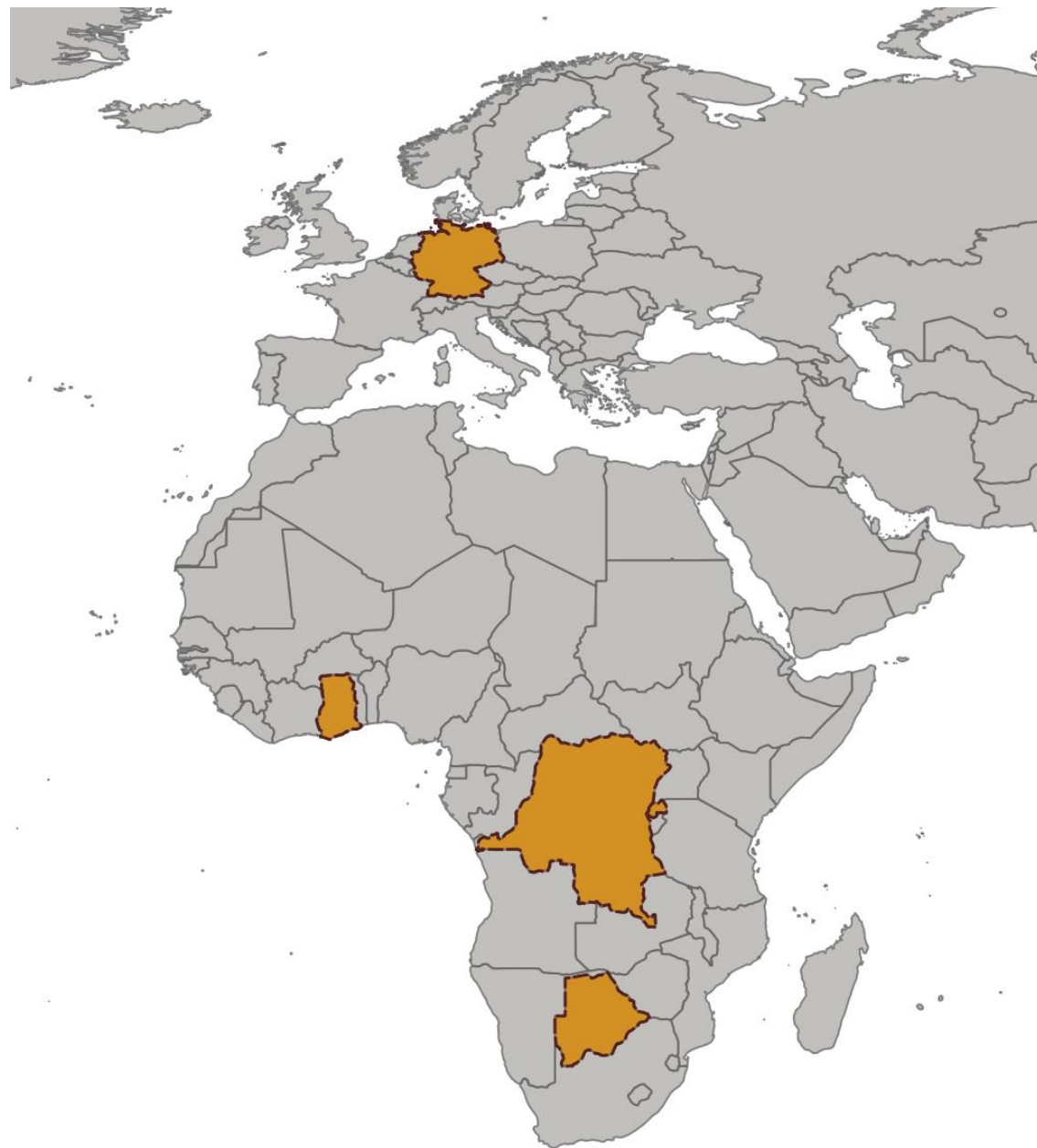
- Under the symbology tab you can stylize your layer in whatever way you see fit
- By pressing Ok (closes the window) or Apply you change the style!

Play around with the symbology options and change the color, transparency and border of the two vector data sets



Vector Symbolology example

In this example I adjusted the color scheme, the transparency and the border style of the vector layers



Vector Data in R

Start Rstudio

1. Double click on the downloaded and unzipped R-file
"5.1_How_to_handle_Vector_Data.r"
2. The file that will open in RStudio

Your Code

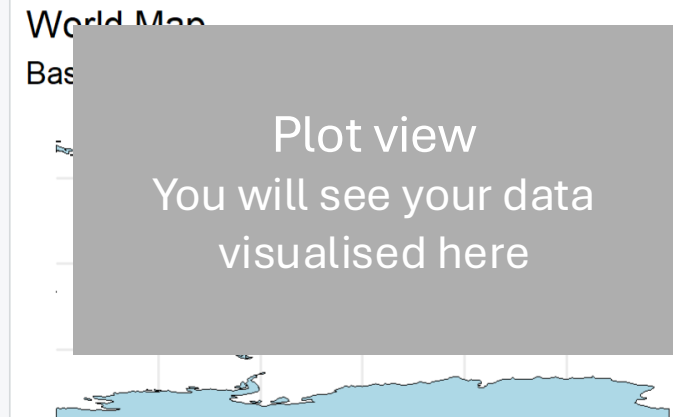
Enviroment tab
Shows all variables you
defined

Plot view

You will see your data
visualised here

Console

Your code is executed and error messages are displayed



Using packages

Packages come with new functions you can use for all kinds of analysis and visualisations

To execute your code, move your cursor to the line of code and press Ctrl + ENTER

```
# 1. Install and load necessary packages  
# First, we need to install some packages that help us load and display visual data  
install.packages(c("sf", "ggplot2"))  
  
# Load the libraries  
library(sf)           # For handling spatial data  
library(ggplot2)      # For visualization, from maps to plots
```

The "c" allows you to create a list of packages to install

Get the Vector Data

- 1) We are using freely available data from natural earth and defining it to the variable "url"
- 2) Next we download the file and unzip it

```
# Download example vector data  
# Example: Using a sample dataset of natural Earth admin boundaries  
url <- "https://naturalearth.s3.amazonaws.com/110m\_cultural/ne\_110m\_admin\_0\_countries.zip"  
download.file(url, "ne_countries.zip")  
unzip("ne_countries.zip", exdir = "ne_countries")  
  
# Read the Shapefile into R using `st_read` from `sf`  
shapefile_path <- "ne_countries/ne_110m_admin_0_countries.shp"  
countries <- st_read(shapefile_path)
```

- 3) We assign the unzipped shapefile to the variable "shapefile_path"
- 4) And read it out using "st_read" in the variable "countries"

Assess the metadata

- Head() gives you the first few rows of all attributes in the attribute table
- Summary() displays all attributes and their data type as well as a range of information about the geometry

```
# 3. Check the Metadata  
# Check the structure of the loaded data and access the attribute data  
head(countries) # Inspect the first few rows of attribute data  
summary(countries) # Summary of attributes and geometry types
```

Visualise Vector Data

- We can use ggplot2 to visualise maps
- We define our variable "countries" as the data
- Using "geom_sf" we can describe the aesthetic of our map
- "labs" lets you set the labels of the map
- "theme_minimal()" sets the overall design of the map

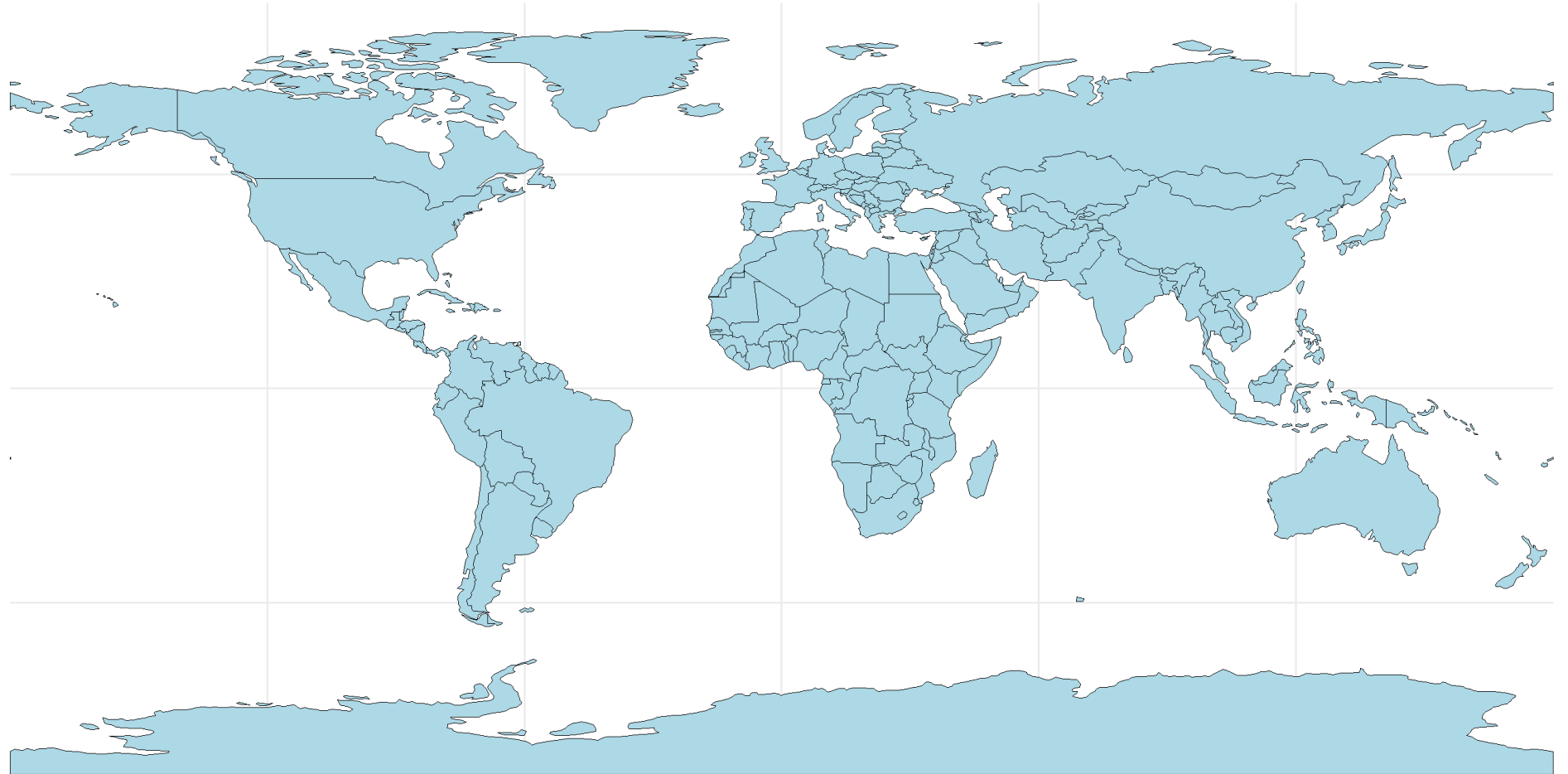
```
# 4. Visualize the data: Create a basic map  
# Use ggplot2 to create a simple map of the countries  
ggplot(data = countries) +  
  geom_sf(fill = "lightblue", color = "black", size = 0.2) +  
  labs(title = "World Map", subtitle = "Basic visualization of vector data") +  
  theme_minimal()
```

Visualise Vector Data

The result:

World Map

Basic visualization of vector data



Visualise Vector Data

- Using "geom_sf(data=" we can ask R to show the attribute "brazil" in a different color on the map

```
# Select and plot specific attributes
# Example: Highlight a single country (e.g., Brazil)
brazil <- countries[countries$NAME == "Brazil", ]
ggplot(data = countries) +
  geom_sf(fill = "gray90", color = "black", size = 0.2) +
  geom_sf(data = brazil, fill = "green", color = "darkgreen", size = 0.5) +
  labs(title = "Highlighting Brazil", subtitle = "Example of subsetting spatial data") +
  theme_minimal()
```

Visualise Vector Data

The result:

Highlighting Brazil
Example of subsetting spatial data



Your turn

- Create another map highlighting any country of your choice
- Adjust the symbology and change the colorscheme of the map