Spatial Analytics: Assignment 2

W2 - Projections and maps

EXERCISE 1

DESCRIPTION

Goals Modify the provided code to improve the resulting map

We highlighted all parts of the R script in which you are supposed to add your own code with:

Loading relevant libraries We will use the sf, raster, and tmap packages. Additionally, we will use the spData and spDataLarge packages that provide new datasets. These packages have been preloaded to the worker2 workspace.

```
library(sf)
```

```
## Linking to GEOS 3.6.2, GDAL 2.2.3, PROJ 4.9.3
```

```
library(raster)
```

```
## Loading required package: sp
```

```
library(tmap)
library(spData)
library(spDataLarge)
```

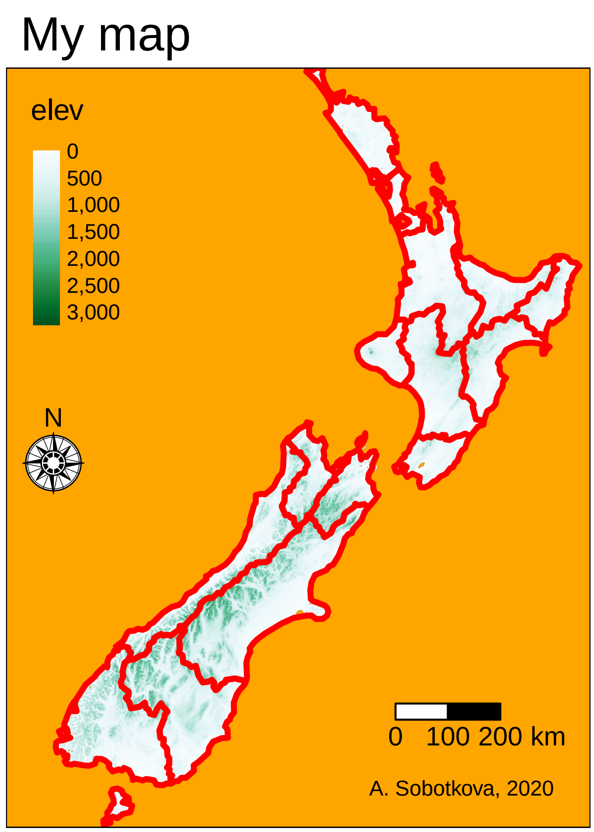
Data sets We will use two data sets: `nz_elev` and `nz`. They are contained by the libraries.
The first one is an elevation raster object for the New Zealand area, and the second one is an sf object with polygons representing the 16 regions of New Zealand.

Existing code We wrote the code to create a new map of New Zealand. Your role is to improve this map based on the suggestions below.

```
tm_shape(nz_elev)  +
  tm_raster(title = "elev",
            style = "cont",
            palette = "BuGn") +
  tm_shape(nz) +
  tm_borders(col = "red",
             lwd = 3) +
```

```
tm_scale_bar(breaks = c(0, 100, 200),
             text.size = 1) +
tm_compass(position = c("LEFT", "center"),
           type = "rose",
           size = 2) +
tm_credits(text = "A. Sobotkova, 2020") +
tm_layout(main.title = "My map",
          bg.color = "orange",
          inner.margins = c(0, 0, 0, 0))
```

```
## stars object downsampled to 877 by 1140 cells. See tm_shape manual (argument raster.downsample)
```



Exercise I

1. Change the map title from "My map" to "New Zealand".
2. Update the map credits with your own name and today's date.
3. Change the color palette to "-RdYlGn". (You can also try other palettes from http://colorbrewer2.org/)
4. Put the north arrow in the top right corner of the map.
5. Improve the legend title by adding the used units (m asl).
6. Increase the number of breaks in the scale bar.
7. Change the borders' color of the New Zealand's regions to black. Decrease the line width.
8. Change the background color to any color of your choice.

EXERCISE II

9. Read two new datasets, `srtm` and `zion`, using the code below. To create a new map representing these datasets.

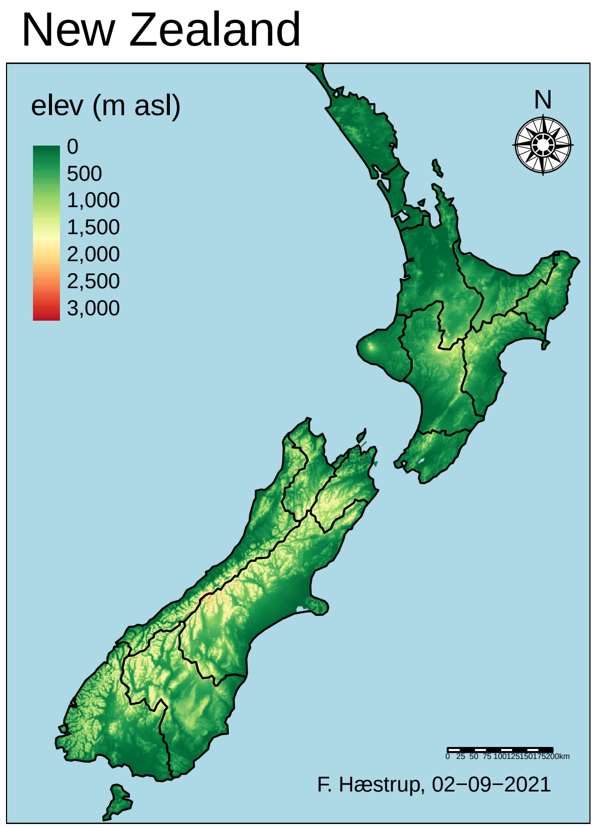
```
srtm = raster(system.file("raster/srtm.tif", package = "spDataLarge"))
zion = read_sf(system.file("vector/zion.gpkg", package = "spDataLarge"))
```

MY SOLUTION

Exercise I Using the provided code, I updated the map of New Zealand:

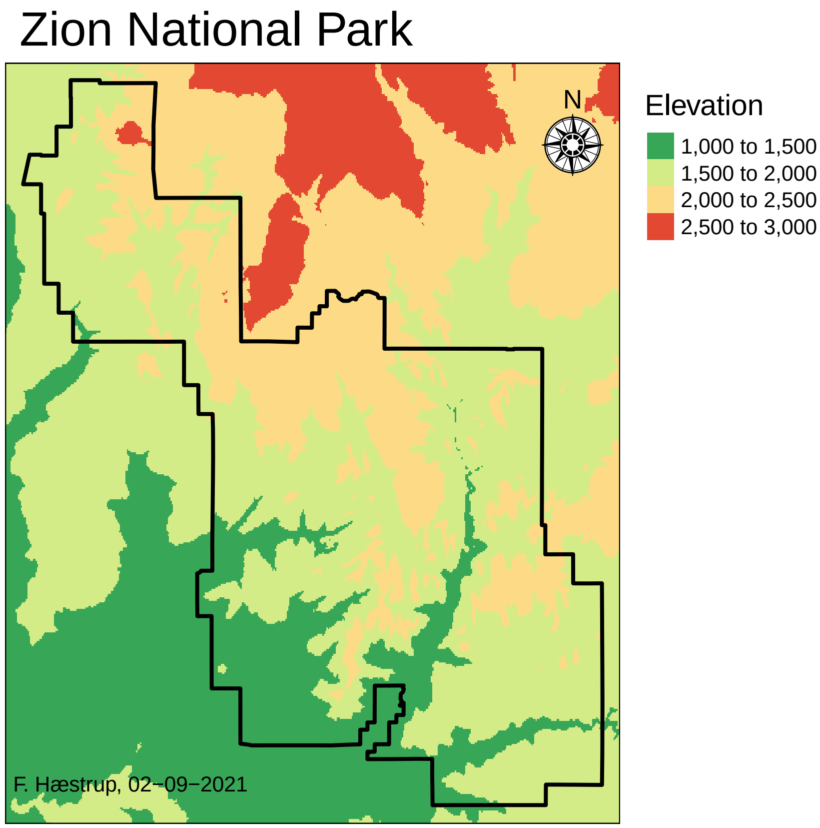
```
tm_shape(nz_elev)  +
  tm_raster(title = "elev (m asl)",  #added units to legend title
            style = "cont",
            palette = "-RdYlGn") + #changed color palette
  tm_shape(nz) +
  tm_borders(col = "black",  #changed line color of borders between regions
             lwd = 1) +  #changed line width of borders between regions
  tm_scale_bar(breaks = c(0, 25, 50, 75, 100, 125, 150, 175, 200),  #added more breaks in scale bar
               text.size = 1) +
  tm_compass(position = c("RIGHT", "top"),  #move North-arrow to top-right corner of map
             type = "rose",
             size = 2) +
  tm_credits(text = "F. Hæstrup, 02-09-2021") +   #added date and my name
  tm_layout(main.title = "New Zealand",
            bg.color = "lightblue",   #changed background color
            inner.margins = c(0, 0, 0, 0))
```

```
## stars object downsampled to 877 by 1140 cells. See tm_shape manual (argument raster.downsample)
```



Exercise II Here, I have created a map representing the srtm and zion data sets:

```
tm_shape(srtm)  +
  tm_raster(title = "Elevation",
            palette = "-RdYlGn") + #color palette
  tm_layout(legend.outside = TRUE) +
  tm_shape(zion) +
  tm_borders(col = "black",  #line color
             lwd = 2) + #line width
  tm_compass(position = c("RIGHT", "top"),  #position of North-arrow
             type = "rose",
             size = 2) +
  tm_credits(text = "F. Hæstrup, 02-09-2021", #added date and my name
             position = c("LEFT", "bottom")) +
  tm_layout(main.title = "Zion National Park", #title
            inner.margins = c(0, 0, 0, 0))
```



EXERCISE 2

DESCRIPTION

Goals

- Understand the provided datasets
- Learn how to reproject spatial data
- Limit your data into an area of interest
- Create a new map

We highlighted all parts of the R script in which you are supposed to add your own code with:

Data sets We will use two data sets: `srtm` and `zion`. The first one is an elevation raster object for the Zion National Park area, and the second one is an sf object with polygons representing borders of the Zion National Park.

```
srtm <- raster(system.file("raster/srtm.tif", package = "spDataLarge"))
zion <- read_sf(system.file("vector/zion.gpkg", package = "spDataLarge"))
```

Exercise I

1. Display the `zion` object and view its structure. What can you say about the content of this file? What type of data does it store? What is the coordinate system used? How many attributes does it contain? What is its geometry?
2. Display the `srtm` object and view its structure. What can you say about the content of this file? What type of data does it store? What is the coordinate system used? How many attributes does it contain? How many dimensions does it have? What is the data resolution?

Exercise II

1. Reproject the `srtm` dataset into the coordinate reference system used in the `zion` object. Create a new object `srtm2` Vizualize the results using the `plot()` function.
2. Reproject the `zion` dataset into the coordinate reference system used in the `srtm` object. Create a new object `zion2` Vizualize the results using the `plot()` function.

MY SOLUTION

Exercise I First, I inspect the zion object to view its structure:

```r
# Inspecting content of the file
head(zion)
```

```
## Simple feature collection with 1 feature and 11 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: 302903.1 ymin: 4112244 xmax: 334735.5 ymax: 4153087
## proj4string:    +proj=utm +zone=12 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
## # A tibble: 1 x 12
##   UNIT_CODE GIS_Notes UNIT_NAME DATE_EDIT  STATE REGION GNIS_ID UNIT_TYPE
##   <chr>     <chr>     <chr>     <date>     <chr> <chr>  <chr>   <chr>
## 1 ZION      Lands - ~ Zion Nat~ 2017-06-22 UT    IM     1455157 National~
## # ... with 4 more variables: CREATED_BY <chr>, METADATA <chr>, PARKNAME <chr>,
## #   geom <POLYGON [m]>
```

```r
# Investigating coordinate system
st_crs(zion)
```

```
## Coordinate Reference System:
##   No user input
##   wkt:
## PROJCS["UTM Zone 12, Northern Hemisphere",
##     GEOGCS["GRS 1980(IUGG, 1980)",
##         DATUM["unknown",
##             SPHEROID["GRS80",6378137,298.257222101],
##             TOWGS84[0,0,0,0,0,0,0]],
##         PRIMEM["Greenwich",0],
##         UNIT["degree",0.0174532925199433]],
##     PROJECTION["Transverse_Mercator"],
##     PARAMETER["latitude_of_origin",0],
##     PARAMETER["central_meridian",-111],
##     PARAMETER["scale_factor",0.9996],
##     PARAMETER["false_easting",500000],
##     PARAMETER["false_northing",0],
##     UNIT["Meter",1]]
```

```r
# Investigating attributes
class(zion)
```

```
## [1] "sf"         "tbl_df"     "tbl"        "data.frame"
```

```r
# Inspecting dimension
dim(zion)
```

```
## [1]  1 12
```

From the code above, we see that the zion object is a vector with x- and y coordinates. The object consists of one row and 12 columns and contains information on region, unit_type, unit_type, etc. From the column, 'geom' we see that it has the geometry of a polygon. Looking at the attributes of the object, we see that it contains 4; a simple feature, a tibble dataframe, a tibble, and a dataframe. The coordinate system used is UTM Zone 12, Northern Hemisphere.

Now we look at the srtm object:

```r
# Inspecting content of file
head(srtm)
```

```
##       1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 1  1728 1718 1715 1710 1703 1701 1700 1704 1701 1692 1682 1676 1684 1693 1688
## 2  1737 1727 1717 1712 1705 1695 1690 1695 1696 1690 1681 1673 1673 1680 1688
## 3  1739 1734 1727 1720 1715 1707 1695 1685 1681 1679 1674 1671 1667 1670 1682
## 4  1729 1723 1718 1718 1721 1714 1698 1686 1678 1673 1670 1665 1662 1664 1667
## 5  1730 1724 1711 1705 1712 1715 1708 1696 1690 1682 1676 1666 1659 1657 1658
## 6  1726 1727 1711 1700 1695 1703 1708 1700 1690 1681 1675 1669 1662 1652 1651
## 7  1711 1719 1705 1691 1683 1693 1697 1687 1682 1673 1669 1667 1663 1657 1652
## 8  1699 1714 1704 1688 1678 1678 1685 1678 1671 1667 1662 1662 1661 1658 1652
## 9  1690 1706 1698 1685 1677 1669 1672 1671 1663 1657 1654 1657 1657 1652 1647
## 10 1683 1697 1694 1681 1667 1660 1658 1659 1654 1649 1647 1652 1652 1647 1640
##      16   17   18   19   20
## 1  1683 1685 1682 1690 1698
## 2  1680 1674 1681 1689 1679
## 3  1673 1669 1685 1689 1667
## 4  1662 1670 1691 1688 1666
## 5  1655 1667 1681 1674 1655
## 6  1650 1656 1667 1659 1642
## 7  1648 1647 1650 1646 1635
## 8  1647 1643 1641 1639 1634
## 9  1642 1640 1637 1634 1634
## 10 1638 1637 1634 1630 1632
```

```r
# Investigating class of the object
class(srtm)
```

```
## [1] "RasterLayer"
## attr(,"package")
## [1] "raster"
```

```r
# Investigating number of layers
nlayers(srtm)
```

```
## [1] 1
```

```r
# Investigating dimensions
dim(srtm)
```

```
## [1] 457 465   1
```

```r
# Invesgating coordinate system
crs(srtm)
```

```
## CRS arguments:
##  +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
```

```r
# Investigating data resolution
res(srtm)
```

```
## [1] 0.0008333333 0.0008333333
```

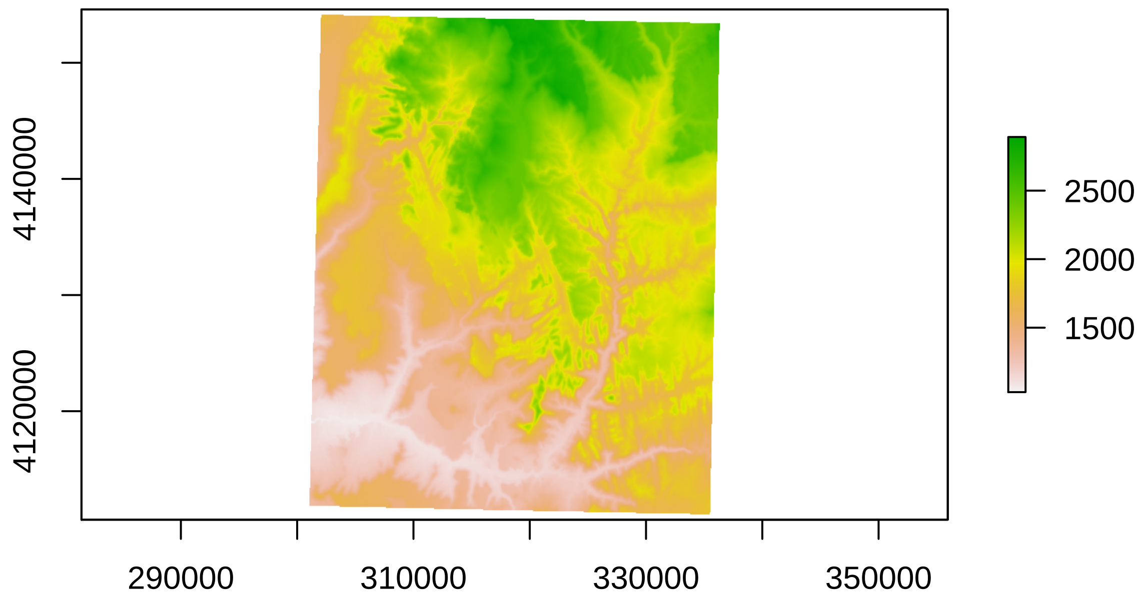
From inspecting the srtm object, we see that it is a raster object with one layer and with 457 rows and 465 columns. The data resolution is x: 0.0008333333, y: 0.0008333333. The coordinate system is " +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0 ".

Exercise II In this exercise, I first reproject the srtm data set into the coordinate reference system of the zion object and visualize the results:

```r
# Get CRS
crs_1 <- crs(zion, asText = TRUE)

# Project srtm into CRS of zion
srtm2 <- projectRaster(srtm, crs = crs_1)

# Plotting results
plot(srtm2)
```

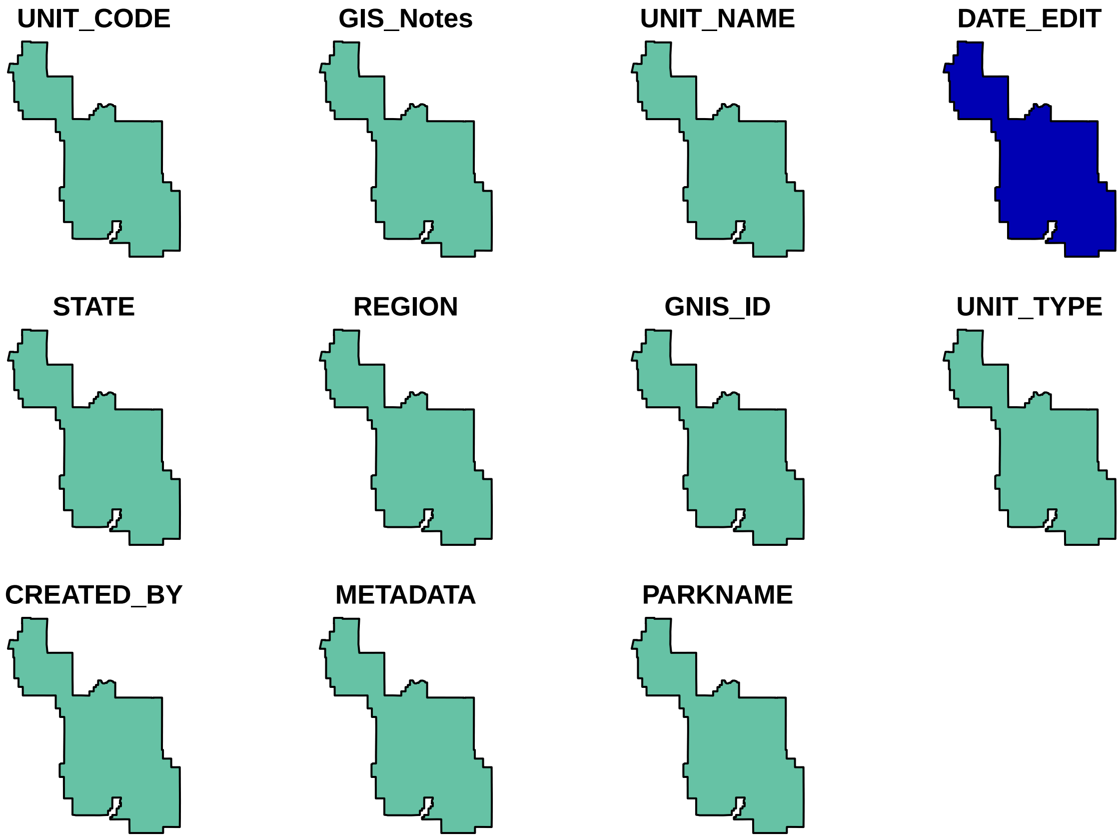


Now, I reproject the zion dataset into the coordinate reference system of the srtm object and visualize the results:

```
# Get CRS
crs_2 <- crs(srtm, asText = TRUE)

# Reproject zion into CRS of srtm
zion2 <- st_transform(zion, crs = crs_2)

# PLotting results
plot(zion2, max.plot =11)
```


UNIT_CODE
GIS_Notes
UNIT_NAME
DATE_EDIT
STATE
REGION
GNIS_ID
UNIT_TYPE
CREATED_BY
METADATA
PARKNAME