API-231 / GIS-PubPol
Meeting 02 (Lab Exercise + Problem Set 2)

Yuri M. Zhukov
Visiting Associate Professor of Public Policy
Harvard Kennedy School

February 1, 2024

Goal of lab exercise: Make multi-layer maps

1. Lab exercise
   a) Map of violence and *terrain* in Afghanistan
   b) Map of violence and *ethnicity* in Afghanistan
2. Problem set
   a) Map of violence and ethnicity in Yemen
      - same approach as the second map above ↑
      - name the file `yem_map_2.png`
      - upload map to Canvas (by next Wednesday)

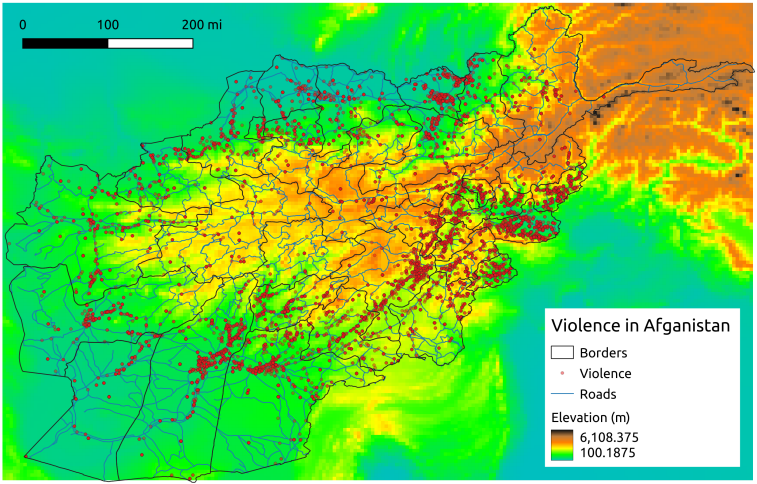


Figure 1: Violence and terrain in Afghanistan (we'll make this)

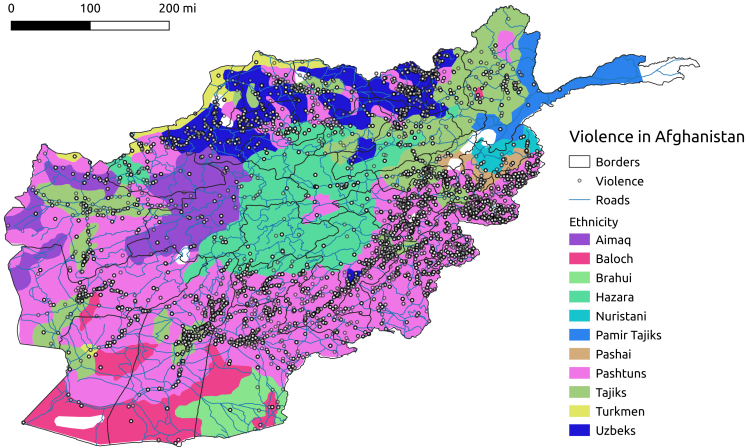


Figure 2: Violence and ethnicity in Afghanistan (this too)

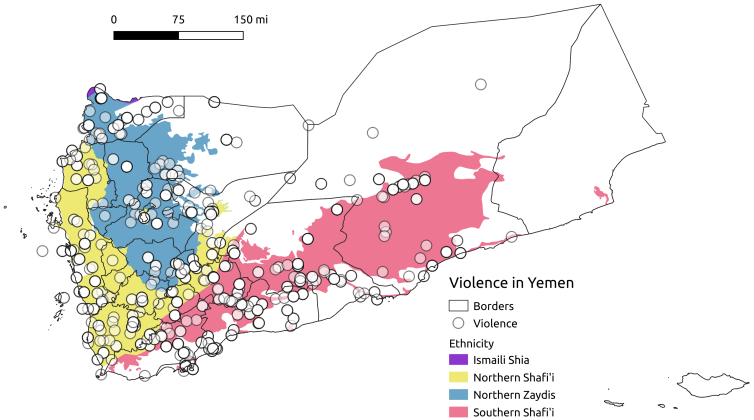


Figure 3: Violence and ethnicity in Yemen (you'll make this)

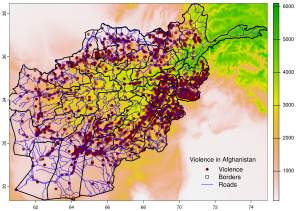


Figure 4: Map 1 in R

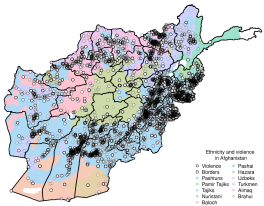


Figure 5: Map 2 in R

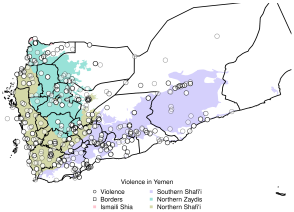


Figure 6: PS02 in R

You can make these plots in QGIS or in R. Instructions for both are below.

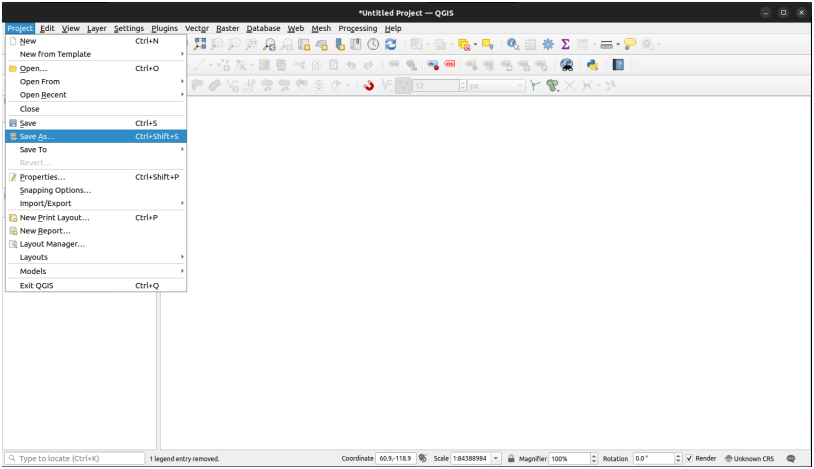
We will put multiple types of data on same map:

Category	Type	Format	Data source
Elevation (meters)	Raster	`.tif`	ETOPO
Province borders	Vector (polygon)	`.shp`	GADM
Roads	Vector (polyline)	`.shp`	DCW
Violence (1989-2016)	Vector (point)	`.csv`	xSub/UCDP
Ethnic settlement patterns	Vector (polygon)	`.shp`	GeoEPR

These are all in the `PS02.zip` file posted on Canvas.

Let's open QGIS...

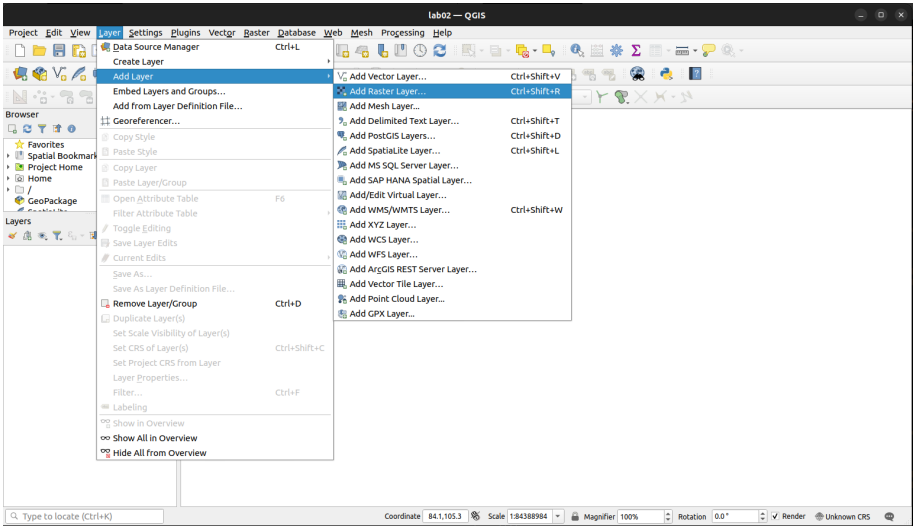
QGIS



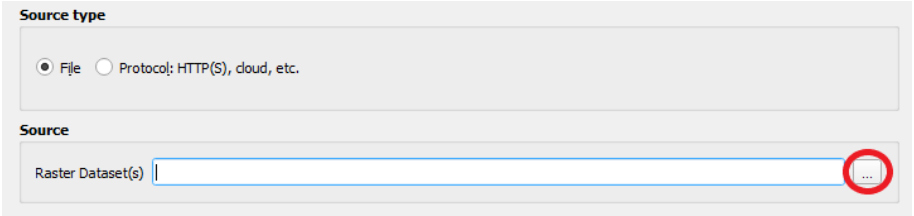
Save your progress!

Go to `Project` → `Save As...`

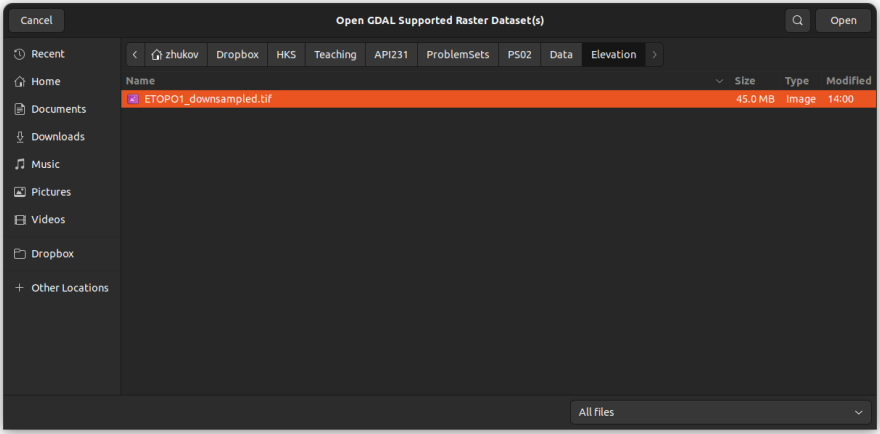
Map 1



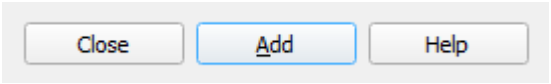
Load elevation data: `Layer` → `Add Layer` → `Add Raster Layer...`



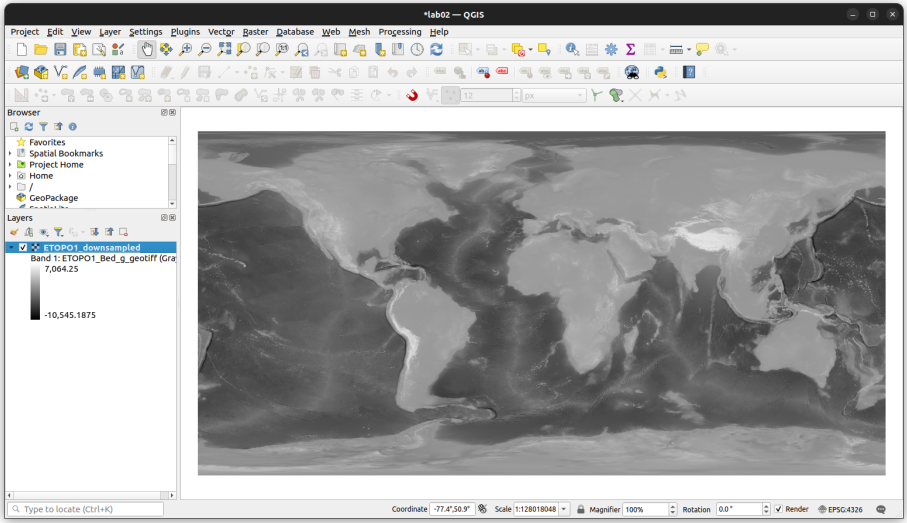
Press the ... button to select the file



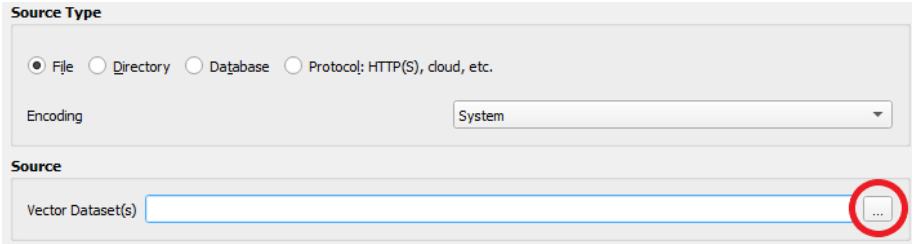
Navigate to `ETOPO1_downsampled.tif` file in `Data/Elevation/` folder, click `Open`



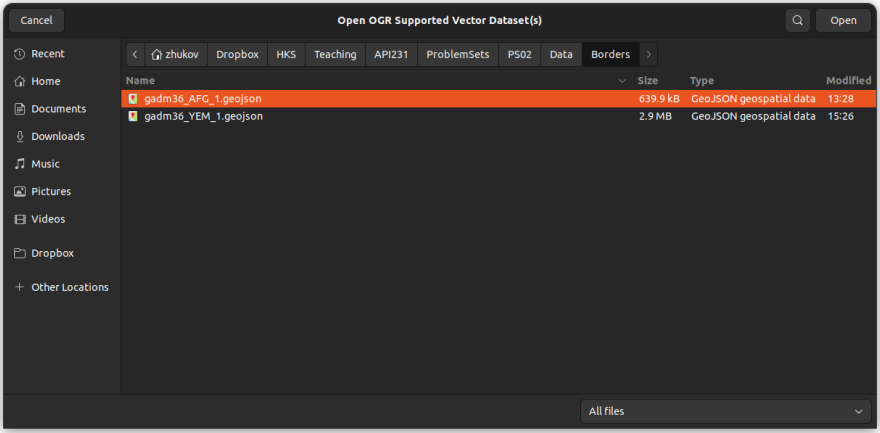
Click `Add` on next screen and go back to main window.



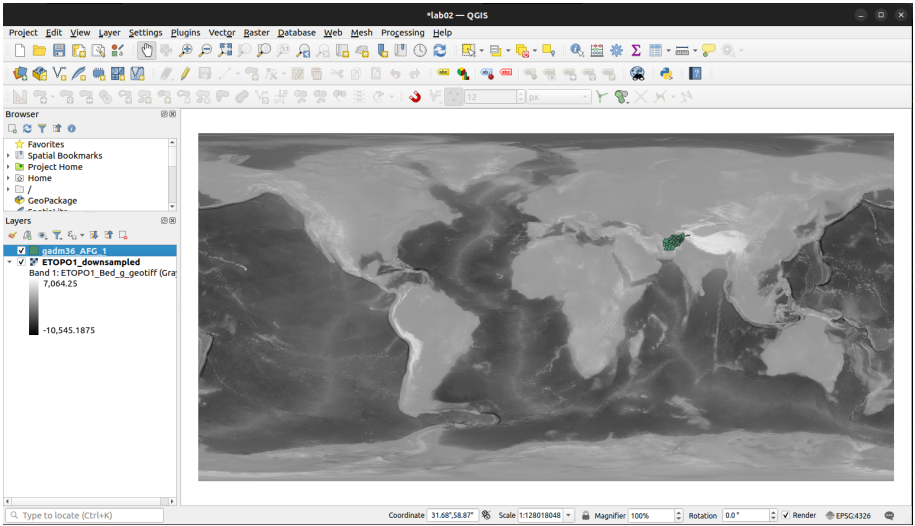
You should now see a global elevation raster



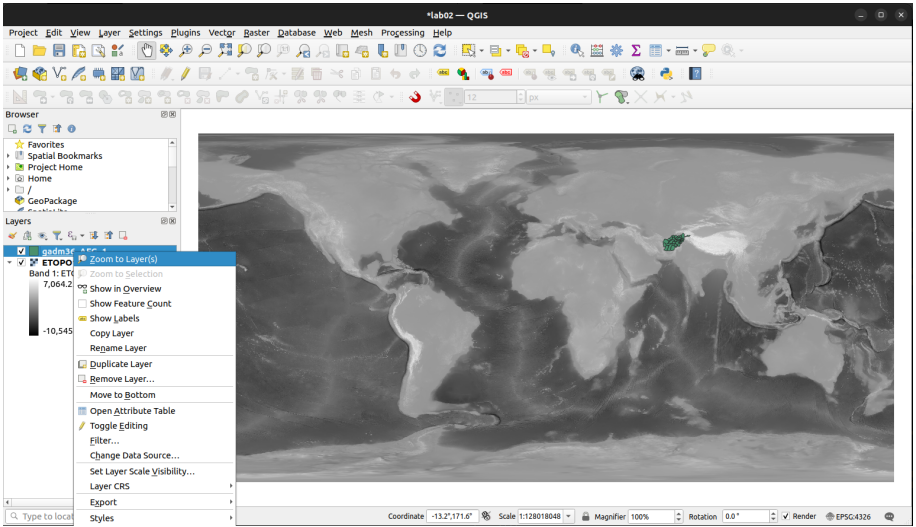
Load Afghanistan borders: `Layer` → `Add Layer` → `Add Vector Layer...`
Click `...` in dialog box to select the file.



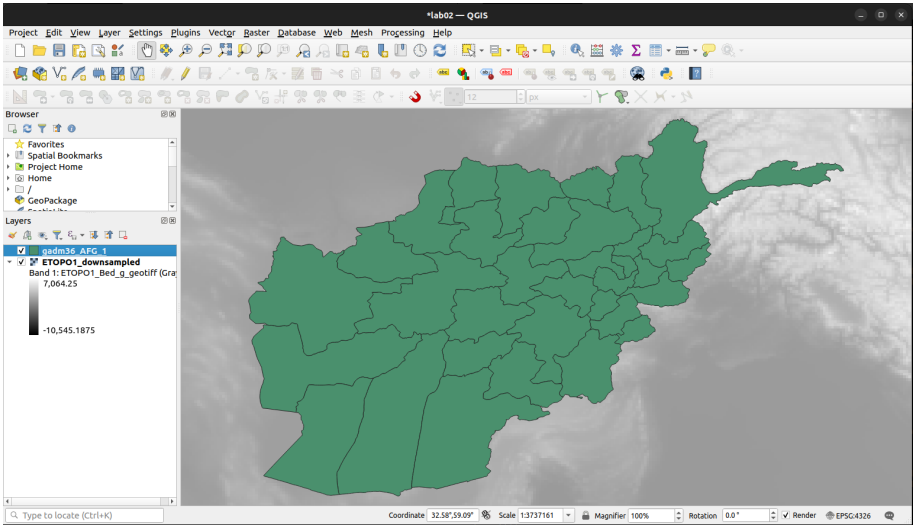
Navigate to `gadm36_AFG_1.geojson` in `Data/Borders/` folder.
Click `Open`, then `Add`



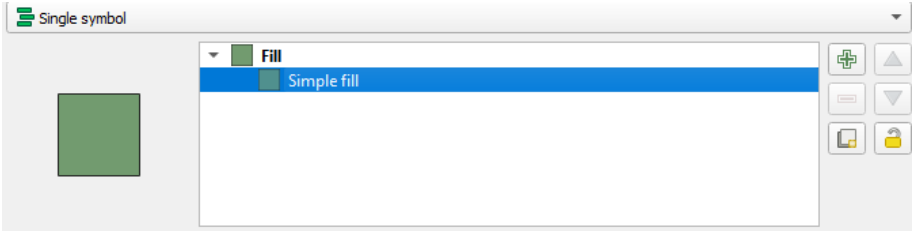
Afghanistan's borders should appear on the map. But they are hard to see



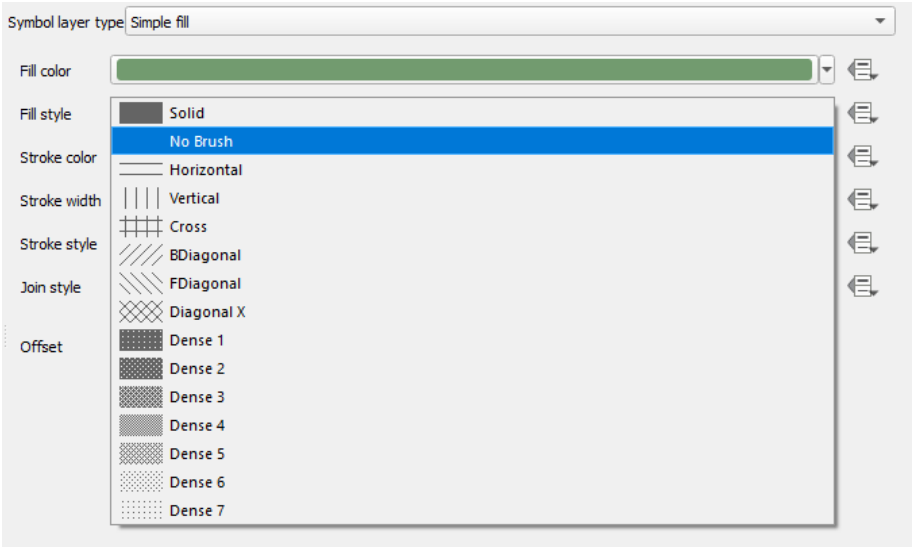
Right click on `gadm36_AFG_1` in the layer menu. Click `Zoom to Layer`



Now we can see Afghanistan, but the elevation layer is hidden behind it

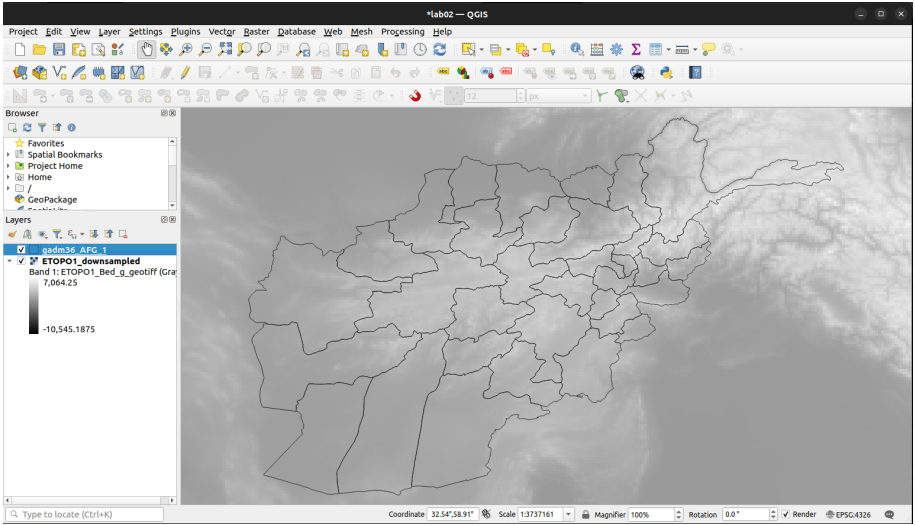


Make polygons transparent: Right-click on `gadm36_AFG_1` in layer menu, click on `Properties`. Highlight `Simple fill` in the symbology menu.

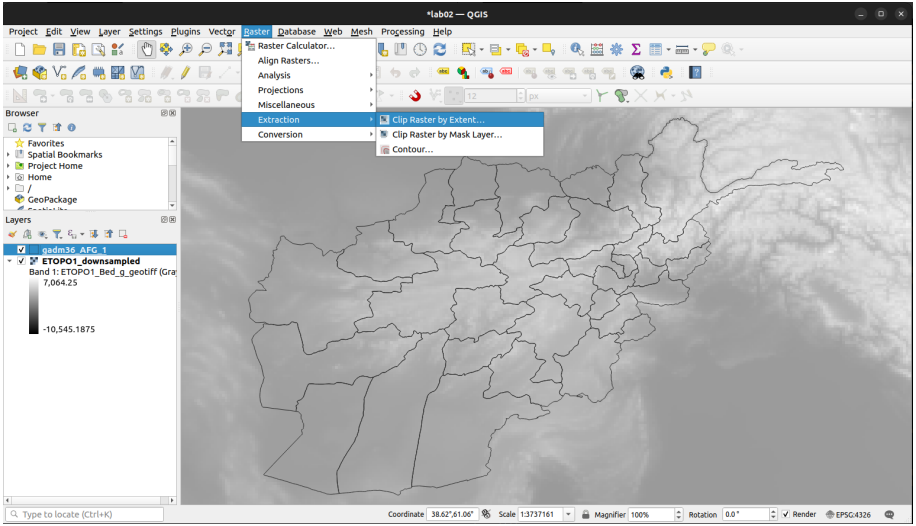


Select `Fill style: No Brush`

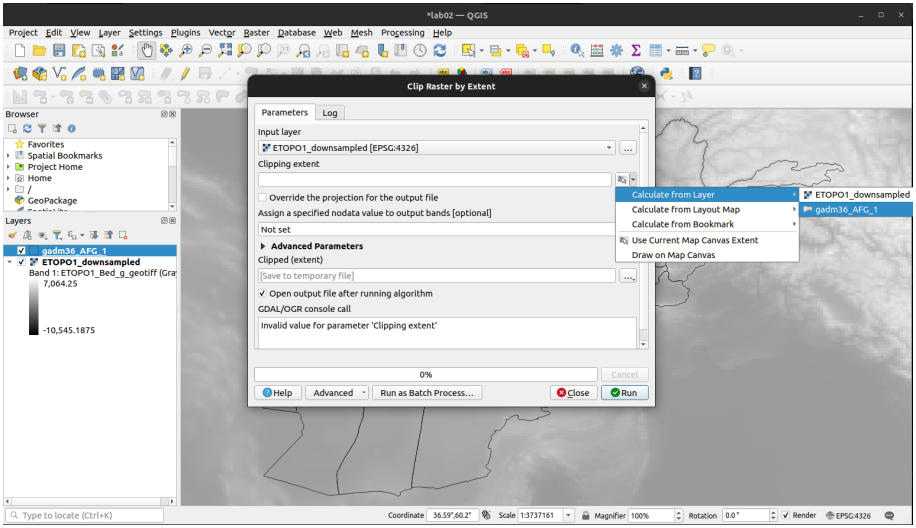
Click OK



The polygons should now be transparent, with provincial borders still visible.

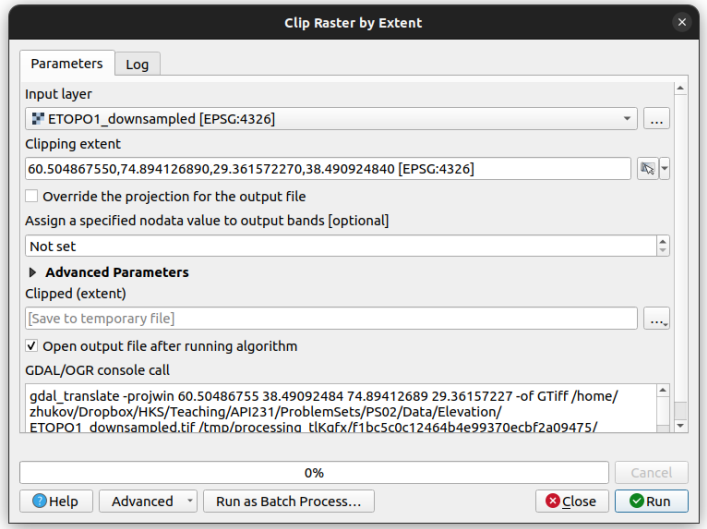


Let's drop the rest of the raster, keeping only the overlap with Afghanistan.
Clip raster: `Raster` → `Extraction` → `Clip Raster by Extent...`

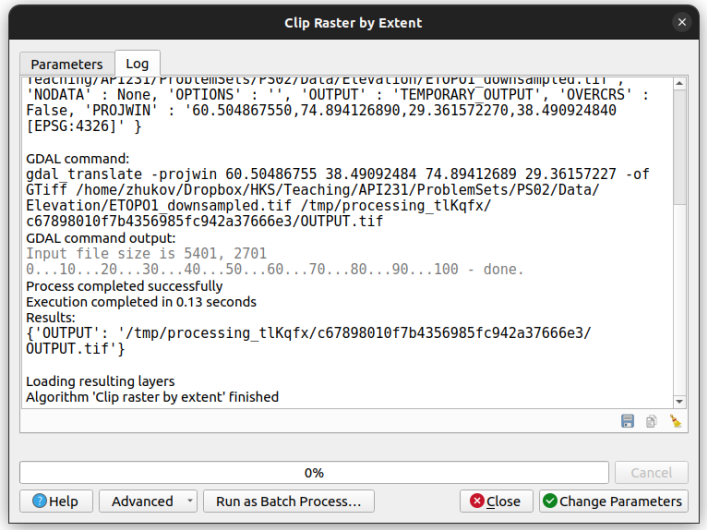


Set the `Input layer` as `ETOPO1_downsampled`.

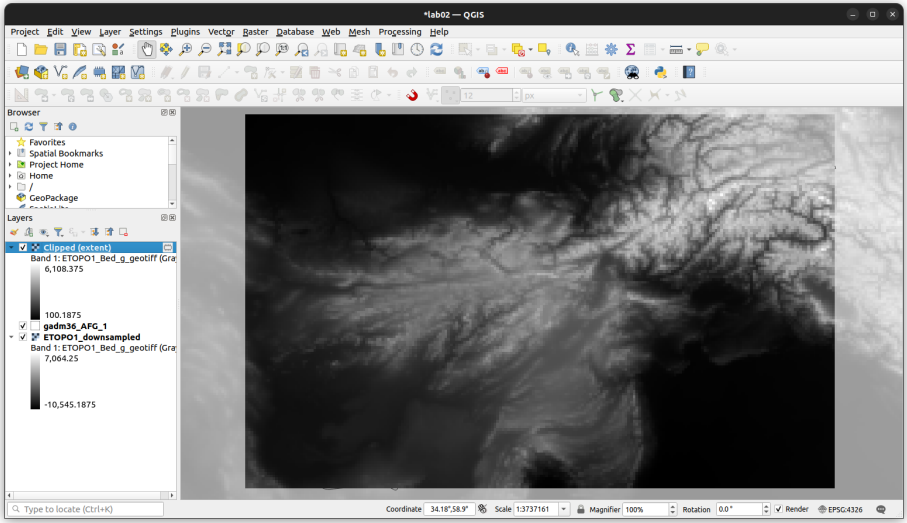
Click `Clipping extent` → `Calculate from Layer` → `gadm36_AFG_1`


Clip Raster by Extent
Parameters
Log
Input layer
ETOPO1_downsampled [EPSG:4326]
Clipping extent
60.504867550,74.894126890,29.361572270,38.490924840 [EPSG:4326]
Override the projection for the output file
Assign a specified nodata value to output bands [optional]
Not set
Advanced Parameters
Clipped (extent)
[Save to temporary file]
Open output file after running algorithm
GDAL/OGR console call
gdal_translate -projwin 60.50486755 38.49092484 74.89412689 29.36157227 -of GTiff /home/zhukov/Dropbox/HKS/Teaching/API231/ProblemSets/PS02/Data/Elevation/ETOPO1_downsampled.tif /tmp/processing_tlKqfx/f1bc5c0c12464b4e99370ecbf2a09475/
0%
Cancel
Help
Advanced
Run as Batch Process...
Close
Run

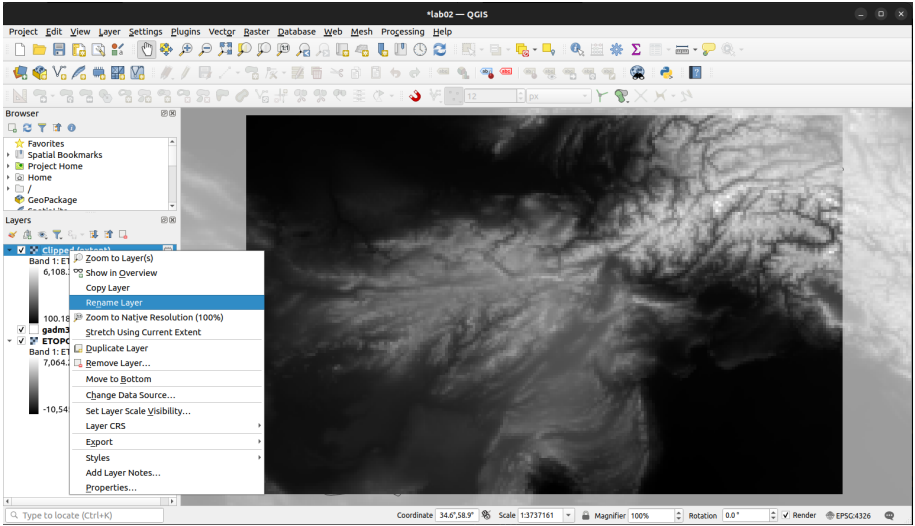
Click Run



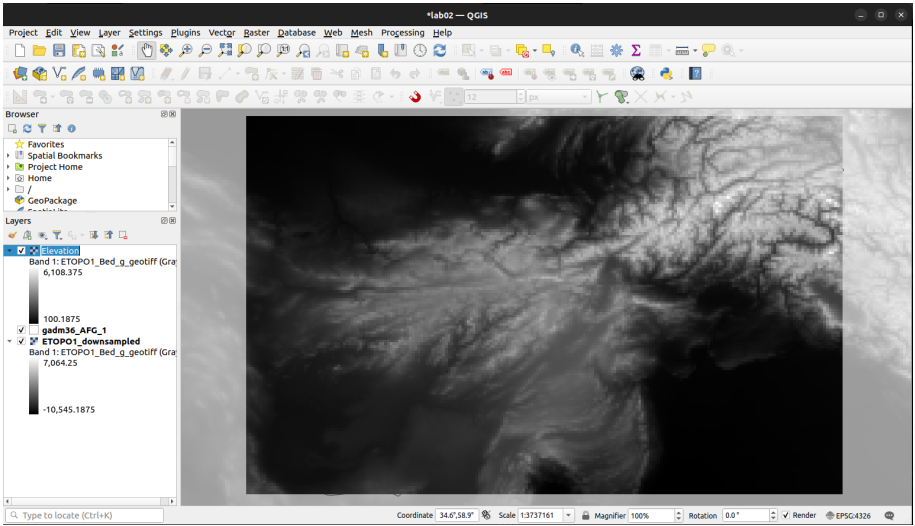
Check log to make sure there were no errors.



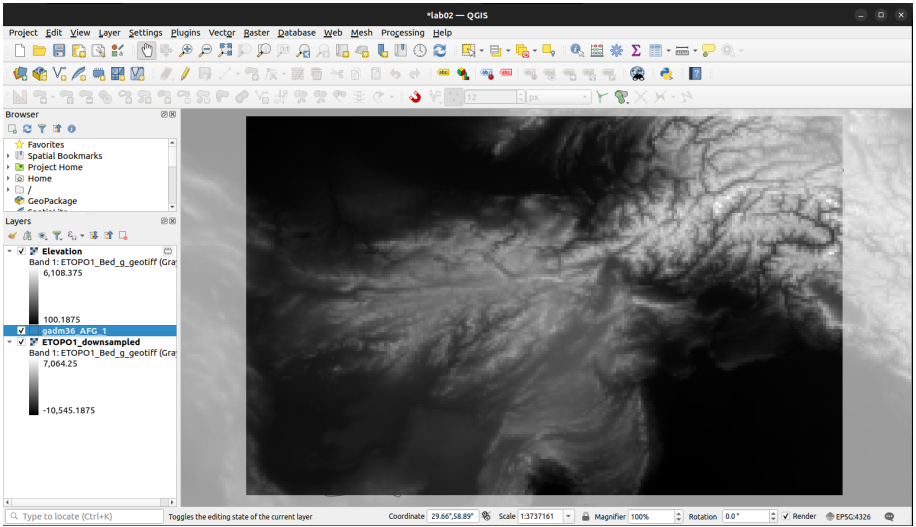
The output is a temporary raster dataset, clipped to extent of Afghanistan.



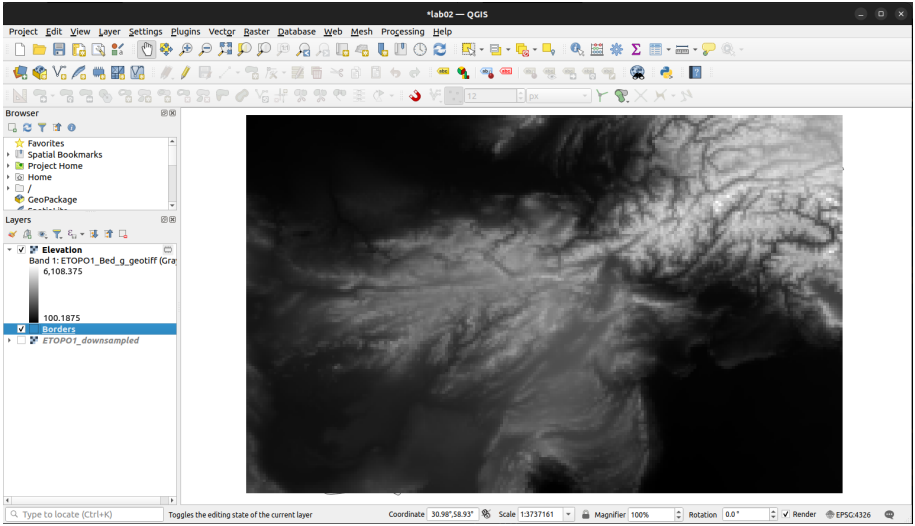
Rename the new layer to “Elevation”. Right click `Clipped` → `Rename Layer`



Change the name to `Elevation`

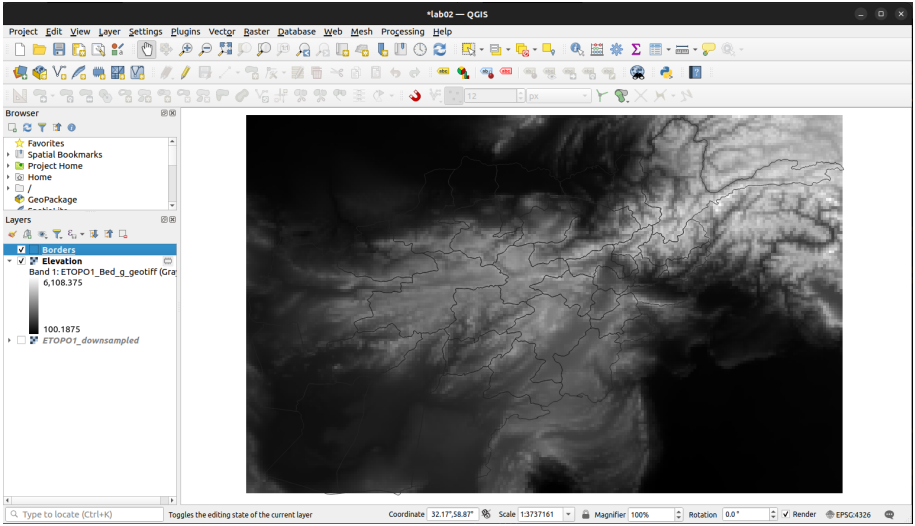


Let's do the same for `gadm36_AFG_1`

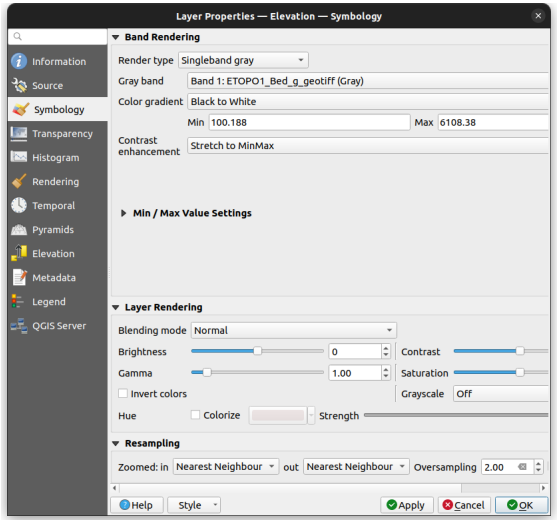


Change its name to `Borders`.

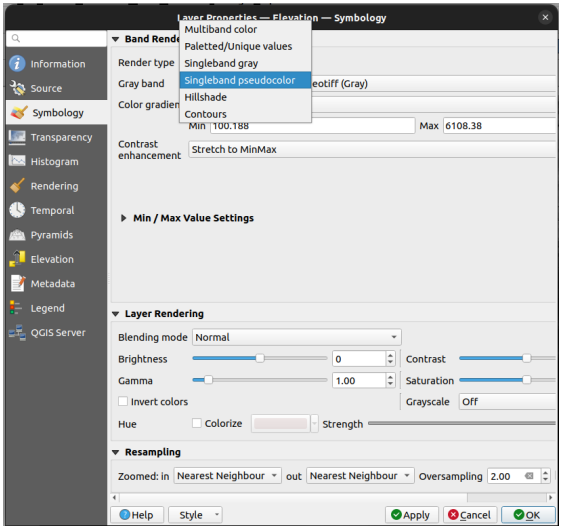
Also, un-check the box next to `ETOPO1_downsampled` to hide the big raster



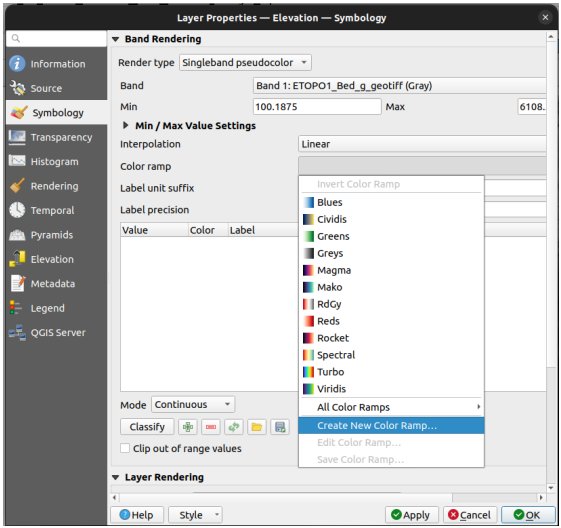
Change plot order. Drag `Borders` above `Elevation` in Layer menu.
Borders should now be (faintly) visible above elevation layer.



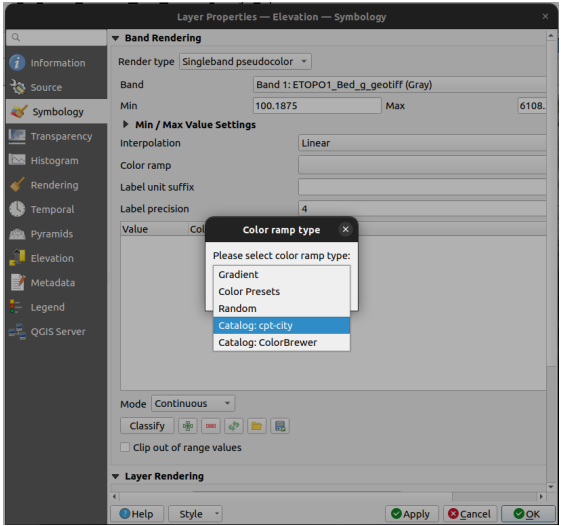
Change raster color. Right click `Elevation` → `Properties`.
Click on the menu next to `Render type`



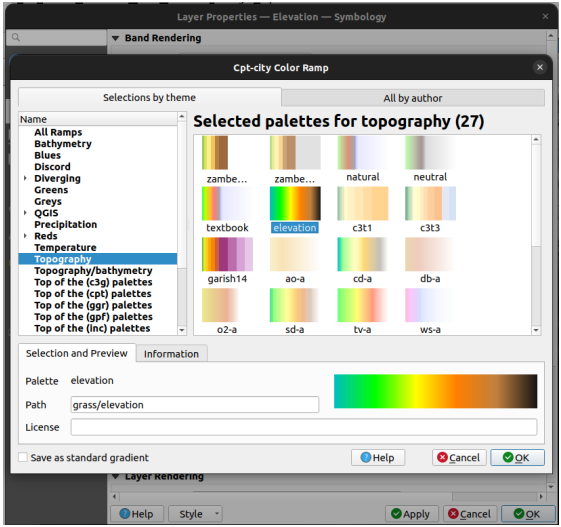
Select `Render type`: `Singleband pseudocolor`



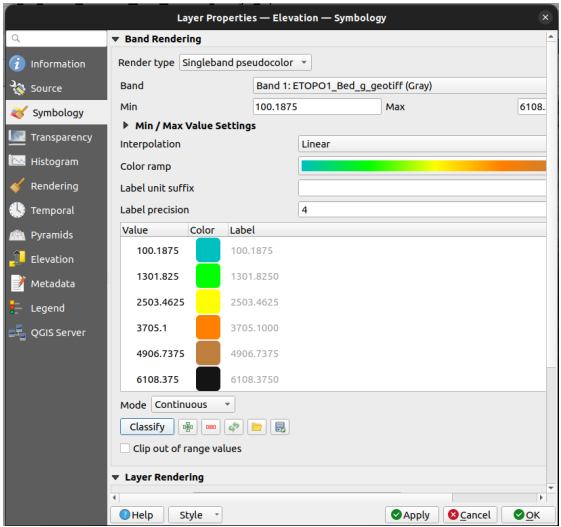
Click menu button next to `Color ramp` → `Create New Color Ramp...`



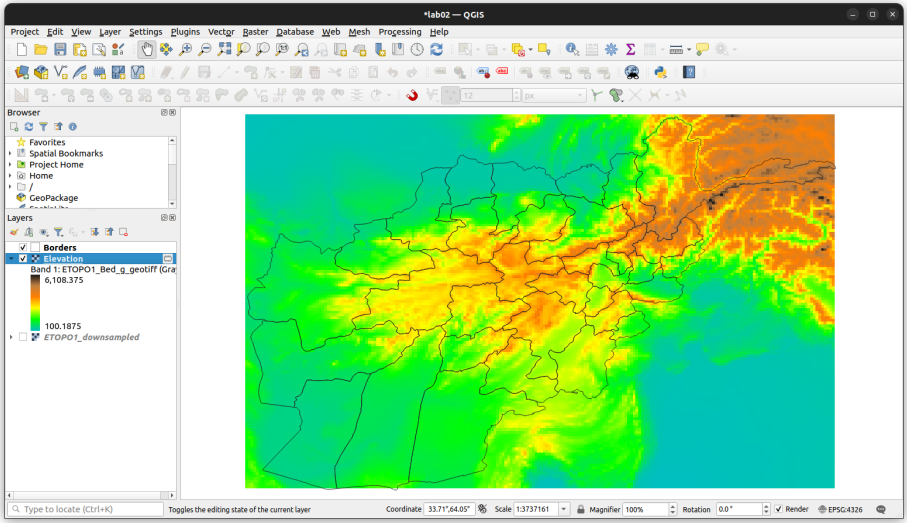
Select `Catalog: cpt-city` as color ramp type



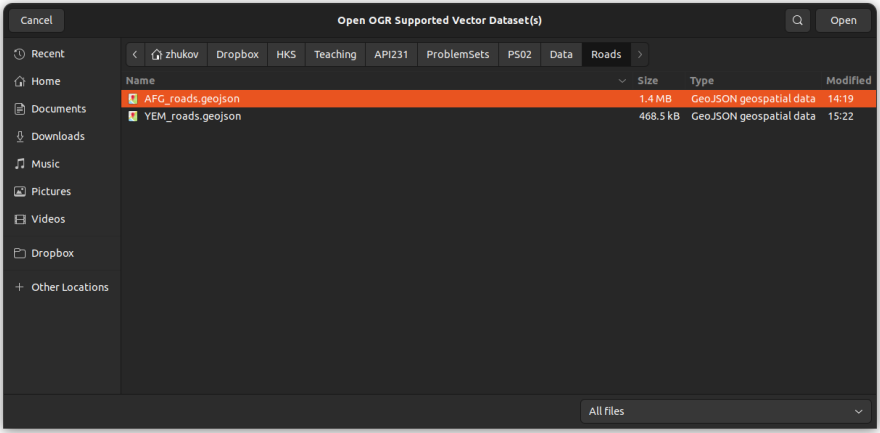
This will open a window with a lot of color themes.
Select `Topography` → `elevation`. Click `OK`



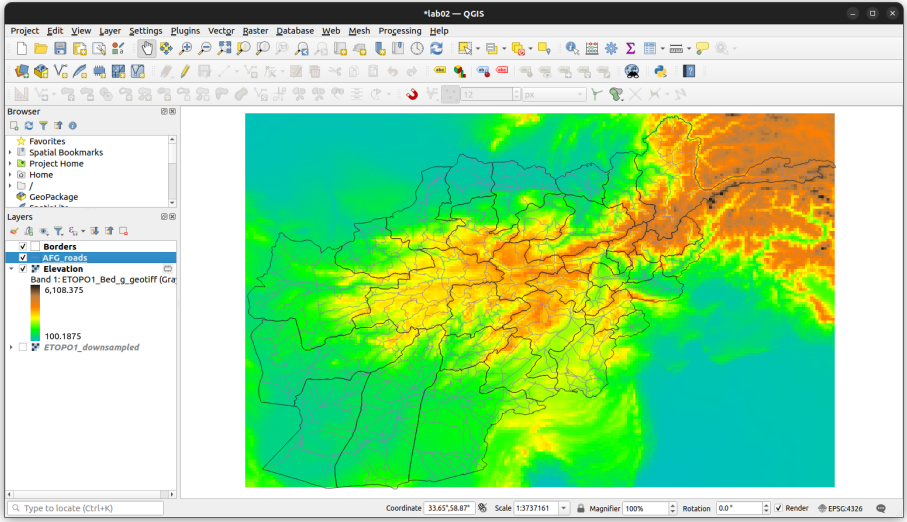
Click `Classify` button, then `OK`



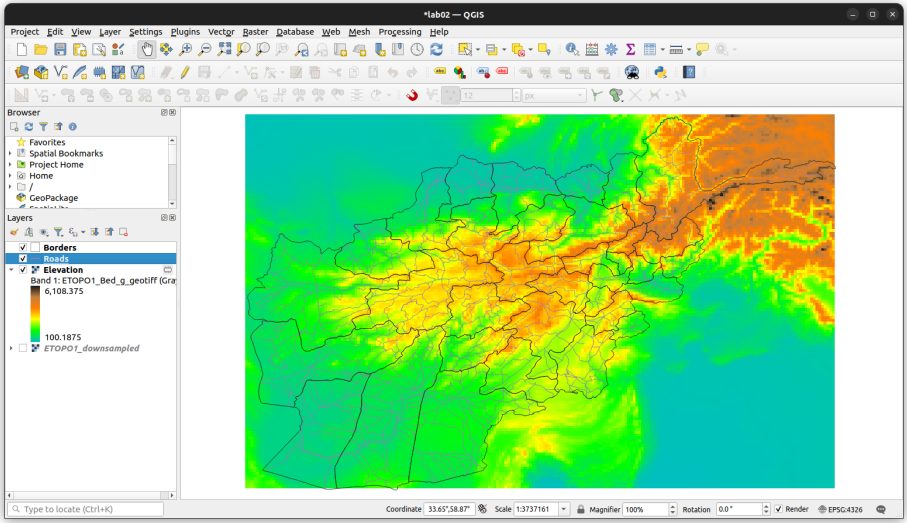
The elevation layer should now appear in blue-green-brown



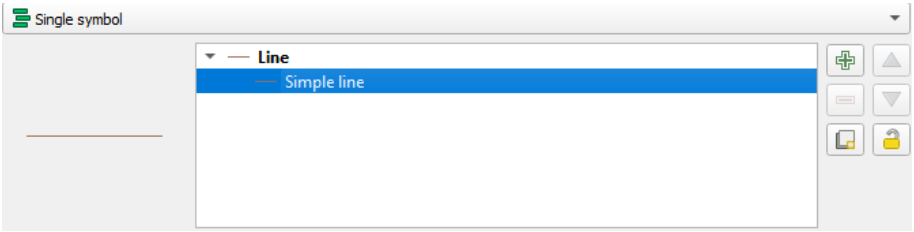
Add roads layer. Same `Add Layer` procedure as for borders. Open `AFG_roads.geojson` from `Data/Roads/` folder



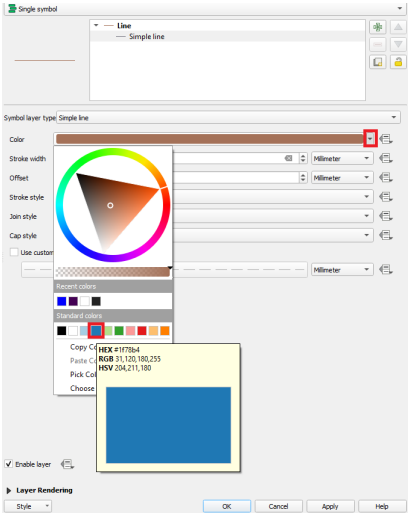
You should see the `AFG_roads` layer on the map



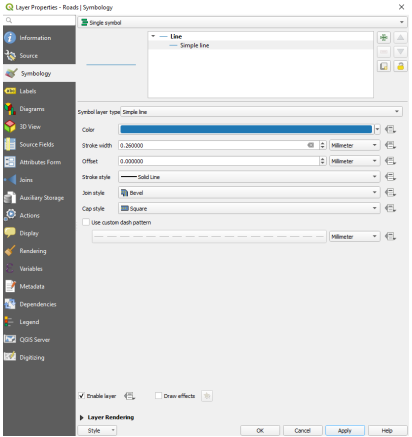
Change layer's name from `AFG_roads` to `Roads`



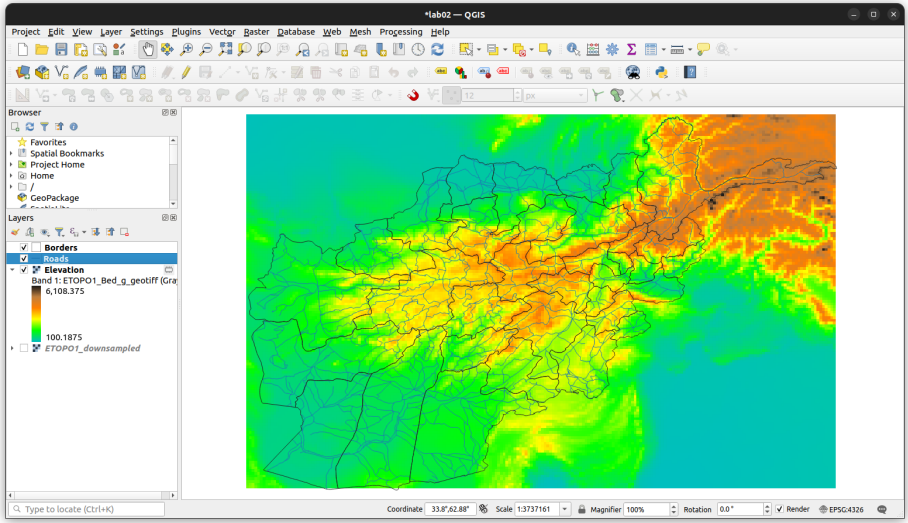
Change color to blue. `Right click` on `Roads` → `Properties`.
Highlight `Simple line` in Symbology menu



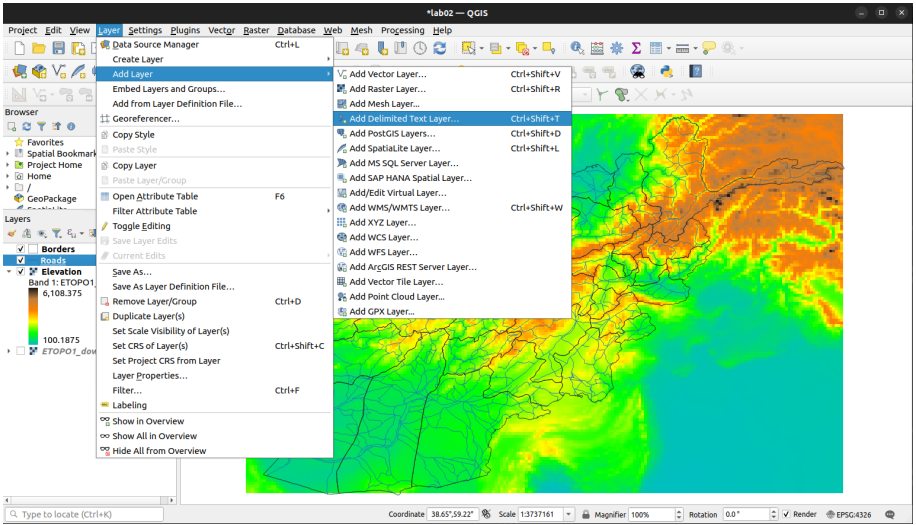
Click menu button next to `Color`. Select blue from `Standard colors` list



Click `Apply` or `OK`

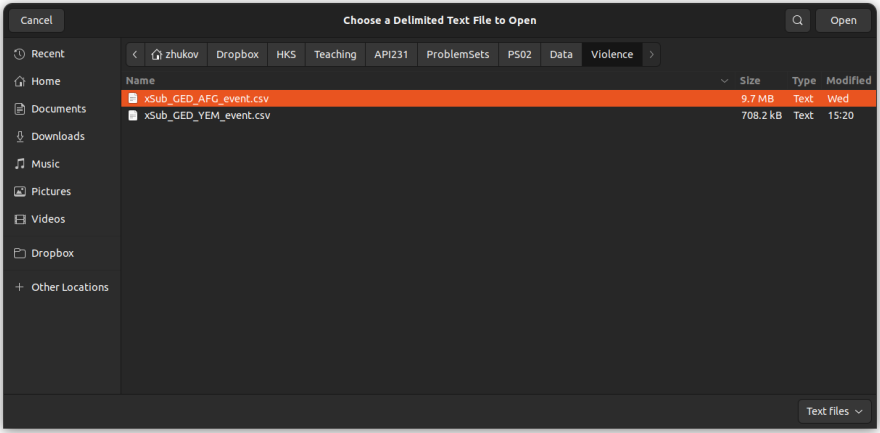


The roads should appear in blue

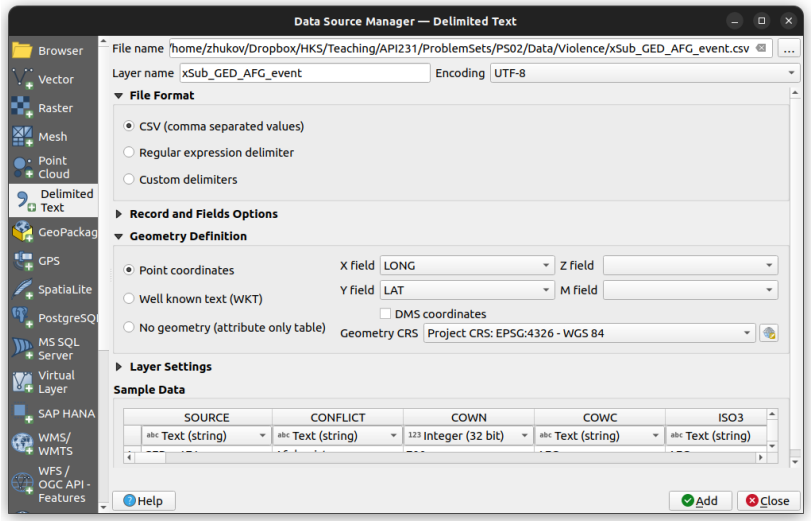


Add violence data. `Layer` → `Add Layer` → `Add Delimited Text Layer...`

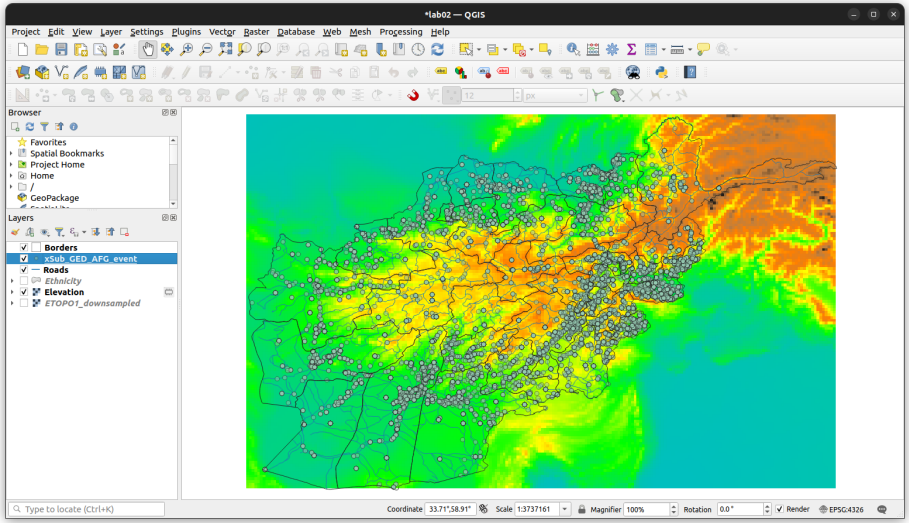
Click `...` button next to `File name` dialog



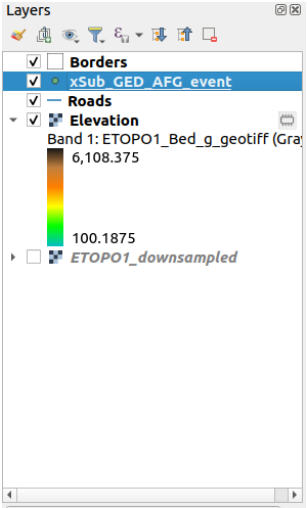
Navigate to `xSub_GED_AFG_event.csv` in `Data/Violence/` menu.
Click `Open`

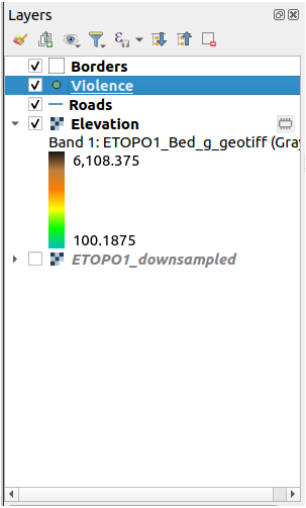


Under `Geometry Definition`, set `X field` = `LONG` and `Y field` = `LAT`. Click `Add`

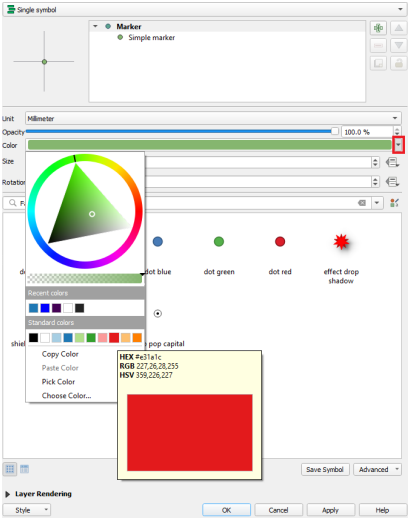


You should now see about 30,000 violent events on the map

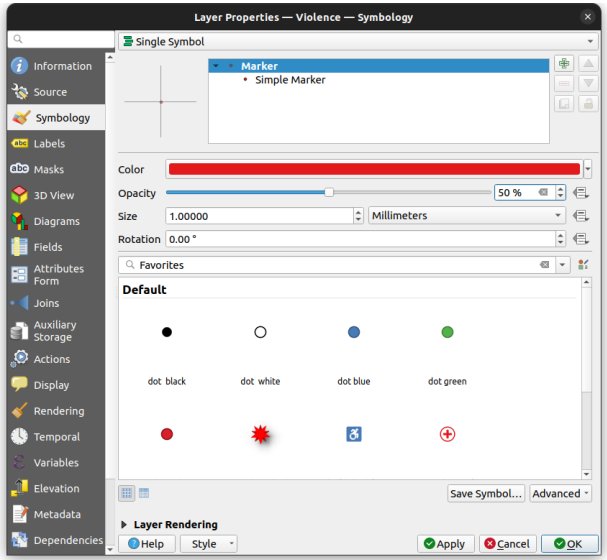




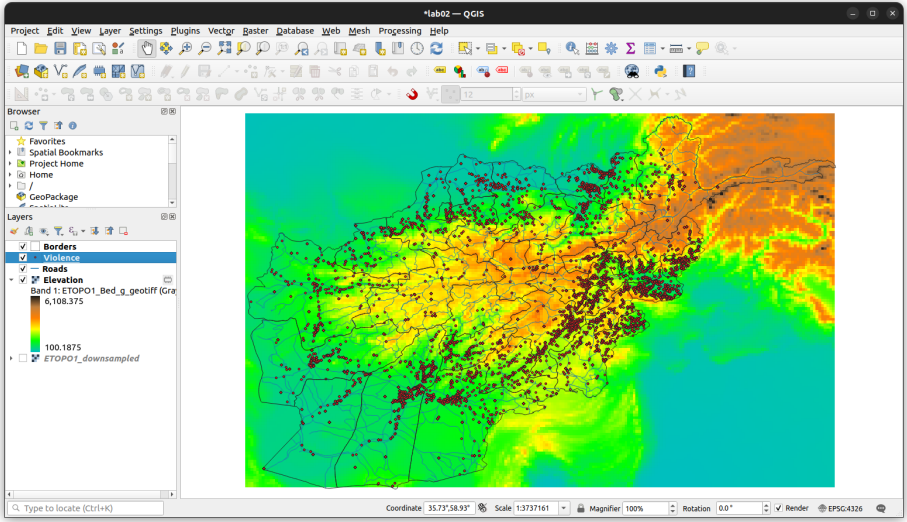
Change layer's name to `Violence`



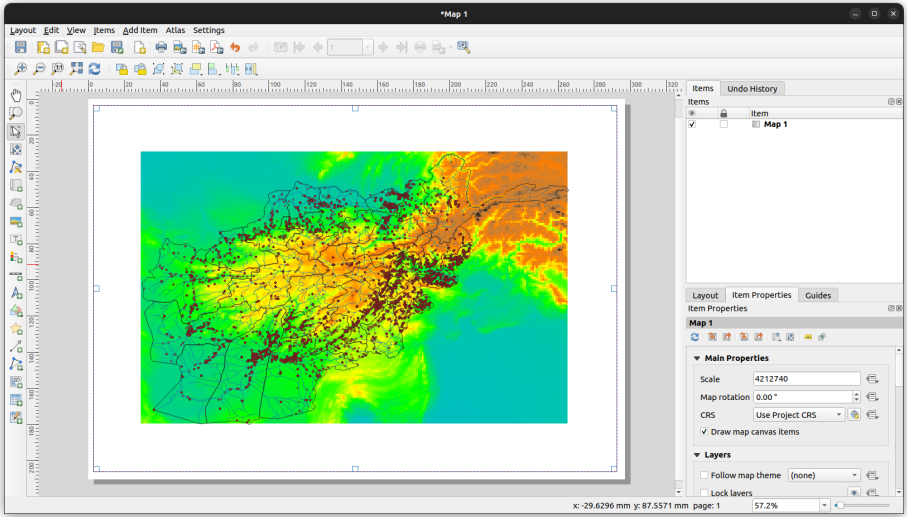
Change layer's color. `Right click` → `Properties`. Click menu button next to `Color`. Select red color in `Standard colors`



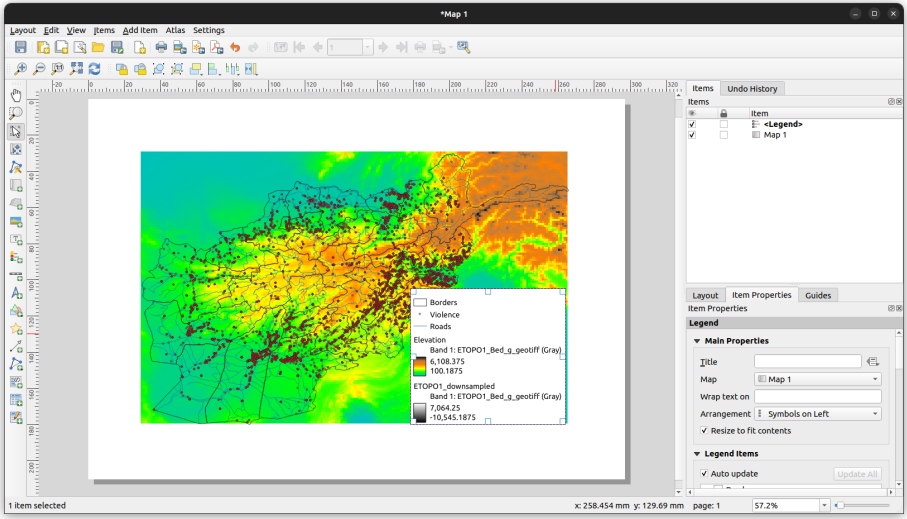
Make points smaller and translucent. Set `Opacity` = 50% and `Size` = `1.0`



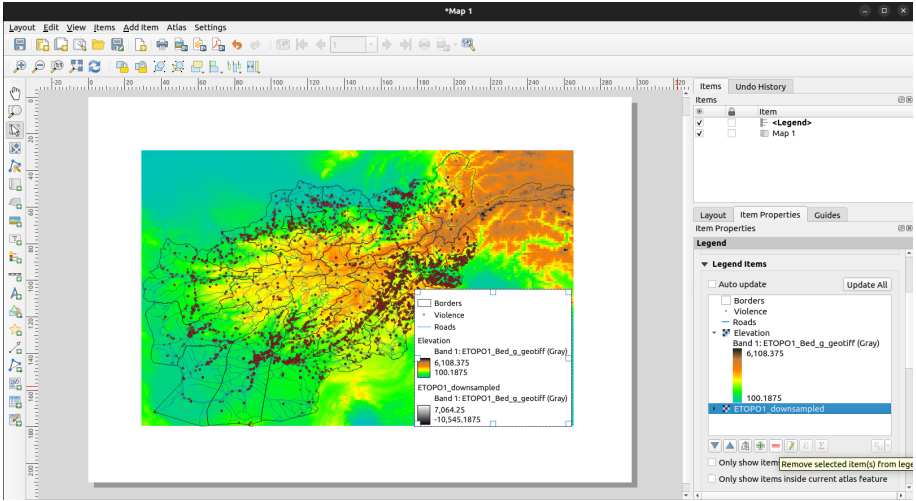
The points should now be smaller and redder.



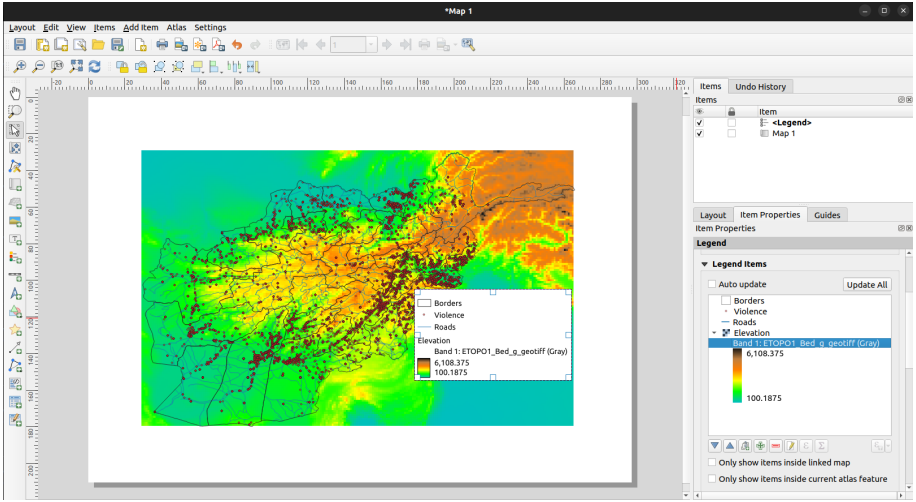
Export the map to image. `Project` → `New Print Layout`.
Place map on layout, with `Add Item` → `Add Map`



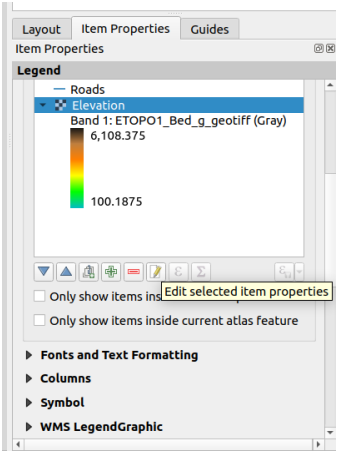
Add legend with `Add Item` → `Add Legend`.
The legend includes an item we need to remove (`ETOPO1_downsampled`)

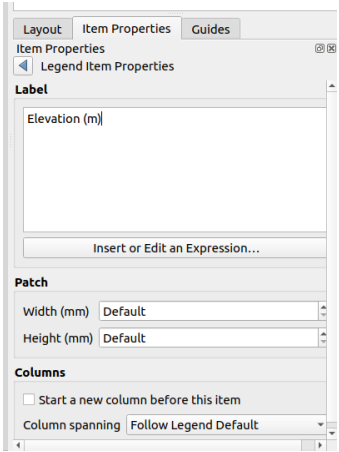


To remove this item, highlight the legend, go to `Item Properties`. Un-check the `Auto update` box under `Legend Items`. Highlight `ETOPO1_downsampled` and click on the red "−" button to remove it

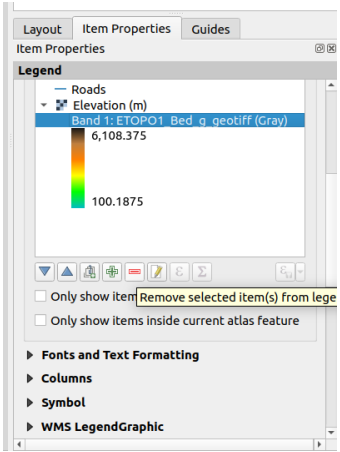


The `ETOPO1_downsampled` item should now disappear from the legend.

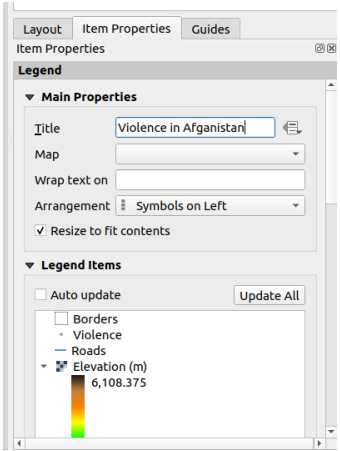




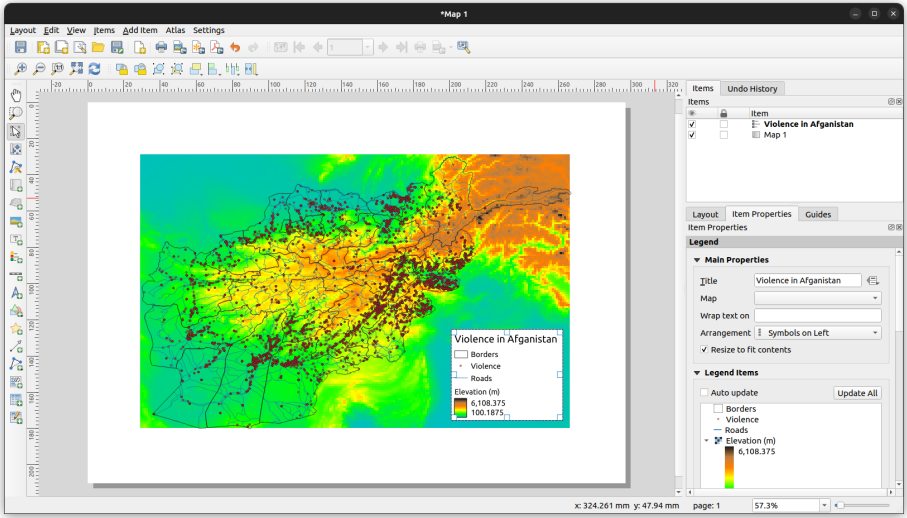
Edit layers' names in legend. Highlight `Elevation` and click on the `Edit` button (pencil and paper icon). Change text to "Elevation (m)"



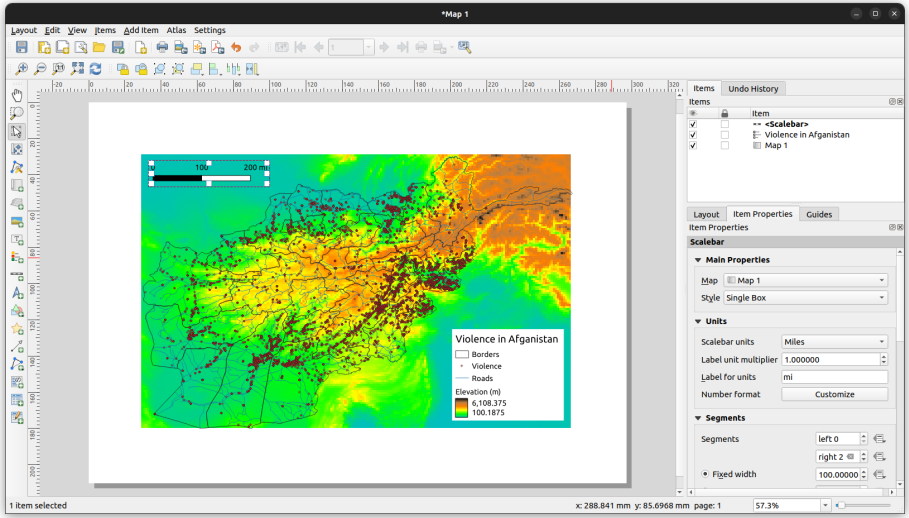
Let's also remove the band information from the legend. Highlight `Band 1: ETOPO1_Bed...`, click the "–" button to remove it.



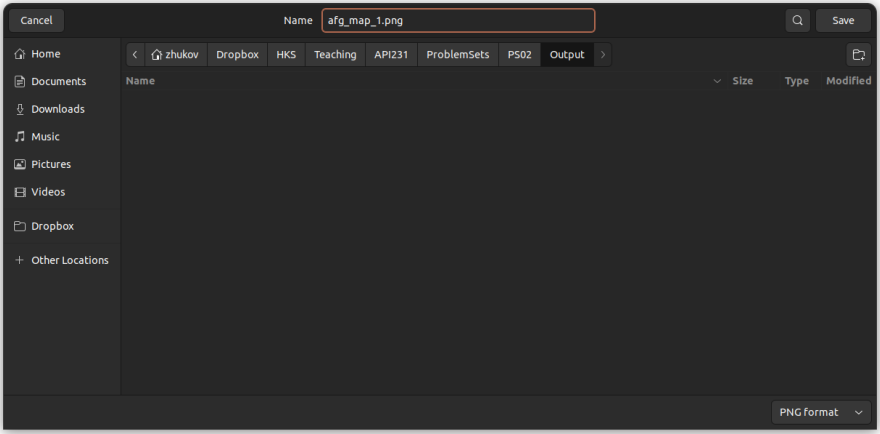
Change legend `Title` to “Violence in Afghanistan” in `Item Properties`



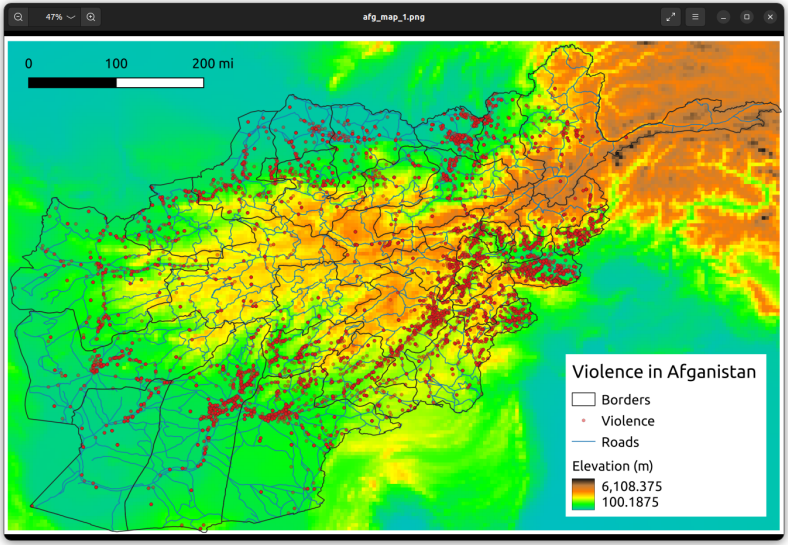
The legend should now look something like this



Add scale bar. Change `Scalebar units` to `Miles`

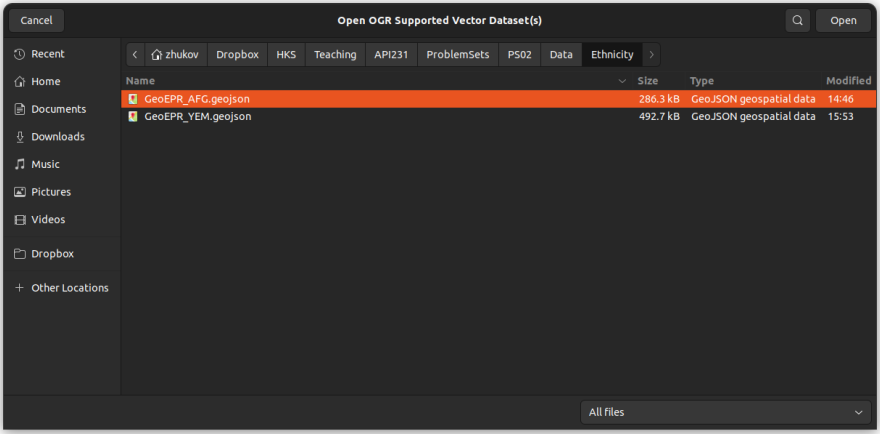


Export to file. `Layout` menu → `Export as Image...`. Name it `afg_map_1.png`

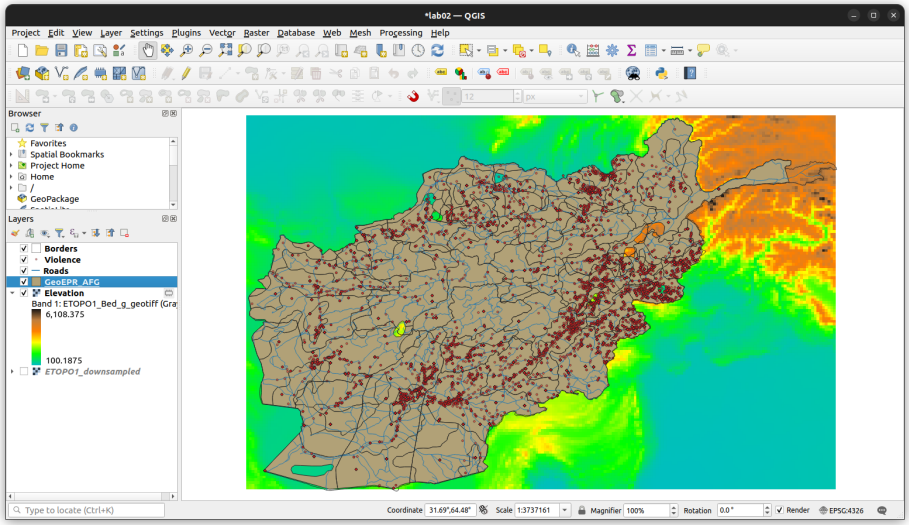


The output file should look like this.

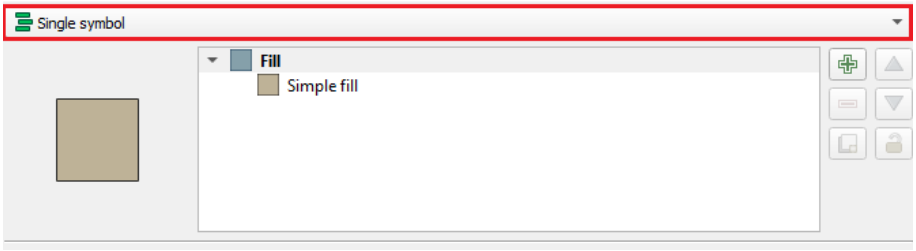
Map 2



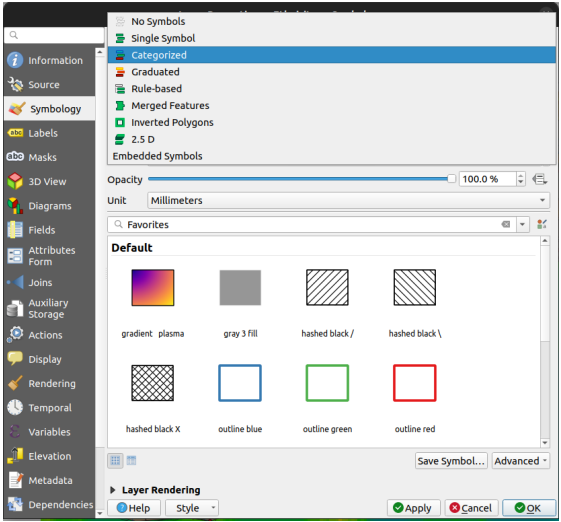
Load ethnicity data. Add (as vector layer) `GeoEPR_AFG.geojson` from `Data/Ethnicity/` folder



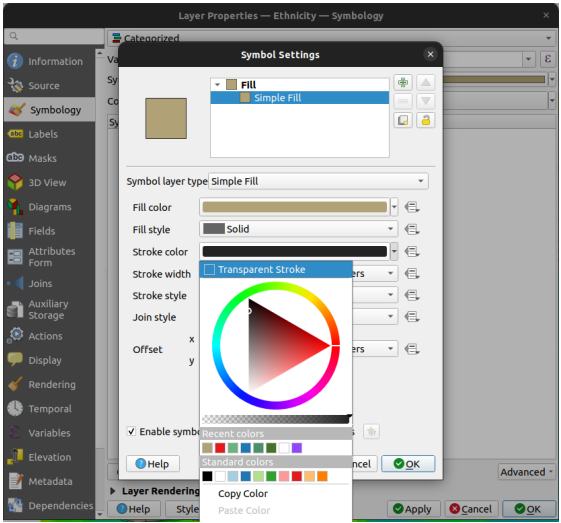
Load ethnicity data. Add (as vector layer) `GeoEPR_AFG.geojson` from `Data/Ethnicity/` folder. This will add a new polygon layer to the map. Change its name from `GeoEPR_AFG` to `Ethnicity`



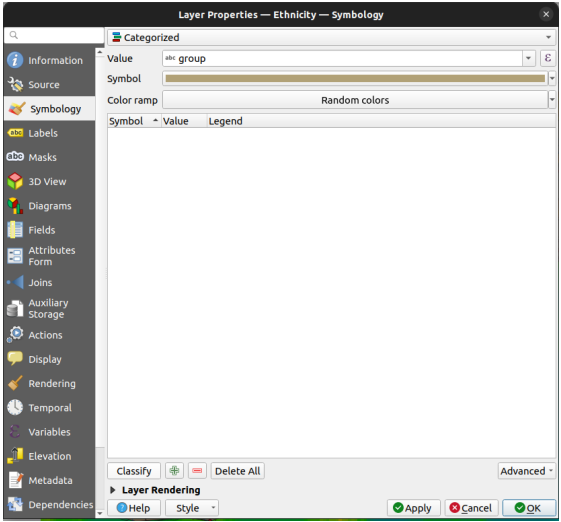
Open the layer's `Properties`. Click on `Single symbol` in Symbology menu



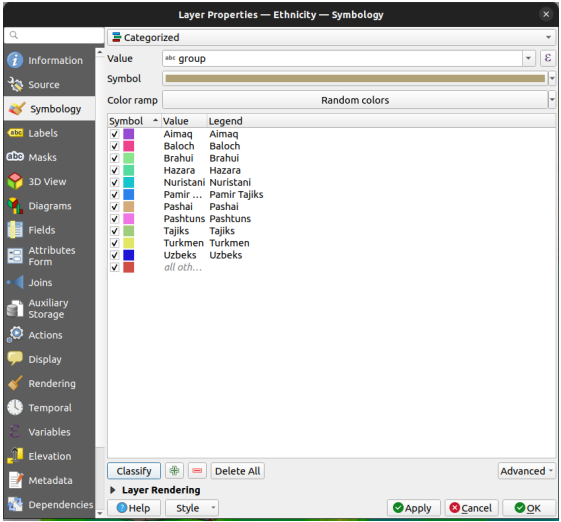
Change symbol type from `Single symbol` to `Categorized`



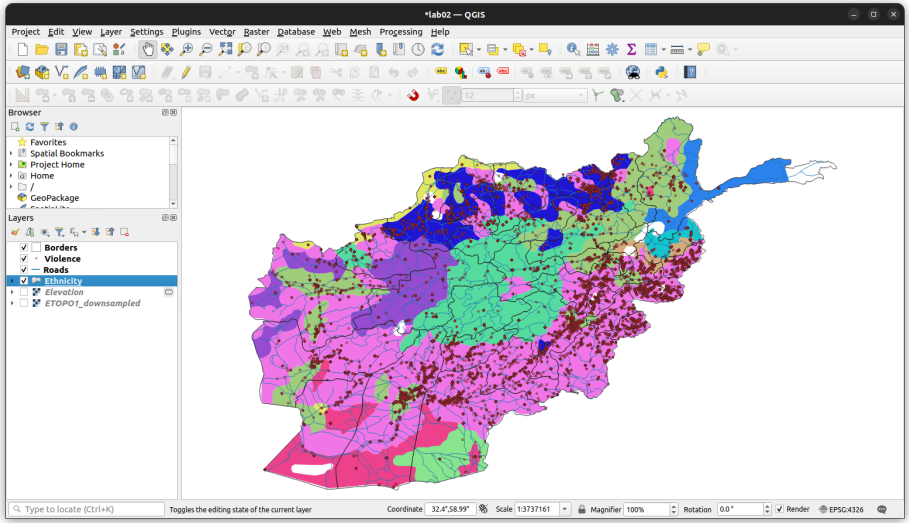
Make polygon boundaries transparent. Click on menu button next to `Stroke color` → check `Transparent Stroke` box



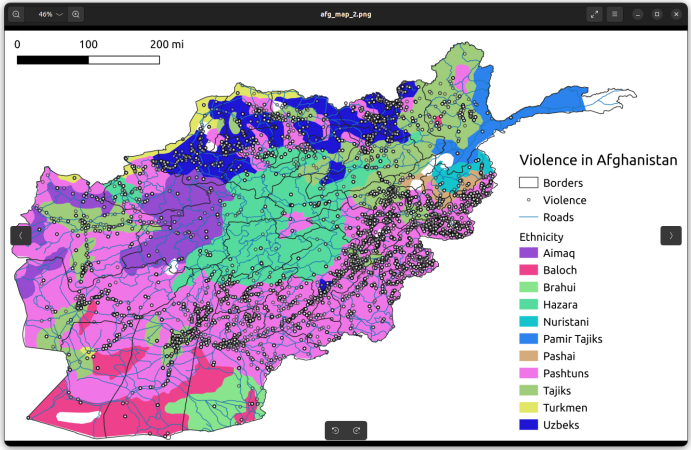
Select `Value` = `group`



Click on `Classify` button, then `Apply` or `OK`



Each ethnic group's polygon should appear as a different color



Create a new map layout and export it as `afg_map_2.png`. You can do it

Problem Set 2

Your assignment (if using QGIS):

- create the same kind of "violence and ethnicity" map for Yemen
- use these files:
  - `GeoEPR_YEM.geojson`
  - `xSub_GED_YEM_event.csv`
  - `gadm36_YEM_1.geojson`
- follow the same steps as for "Map 2" above
- name it `yem_map_2.png`
- upload map to Canvas (by next Wednesday)

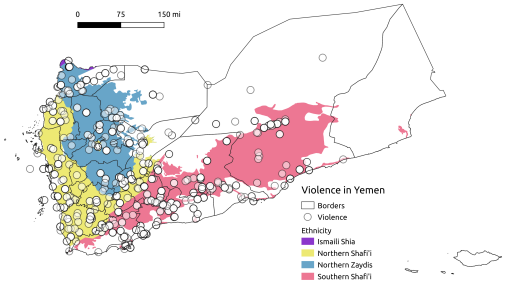


Figure 7: Can you make this map?

R

Loading R packages

To implement these steps in R, we will be using two packages:

- `sf` ("simple features", to handle spatial vector data)
- `terra` (to handle raster data, like Elevation)

```
library(sf)
library(terra) # NEW!
```

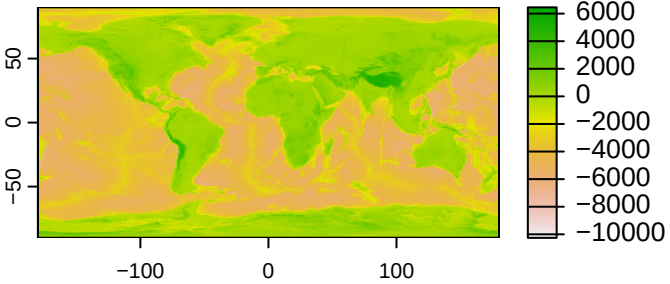
NOTE: The code to produce Map 1 and Map 2 in R is in `ps02_demo.R` on RStudio Cloud, and in `PS02.zip` (posted on Canvas).

Map 1

Loading spatial data

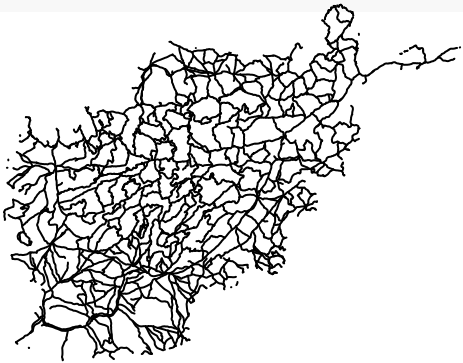
Let's load the *elevation raster* into R, using the `rast()` command from `terra`:

```
elevation_global = terra::rast("Data/Elevation/ETOPO1_downsampled.tif")
plot(elevation_global)
```



Let's load the *provincial boundaries* and *roads* into R, using `sf::read_sf()`:

```
borders = sf::read_sf("Data/Borders/gadm36_AFG_1.geojson")
roads = sf::read_sf("Data/Roads/AFG_roads.geojson")
plot(borders["geometry"])
plot(roads["geometry"])
```

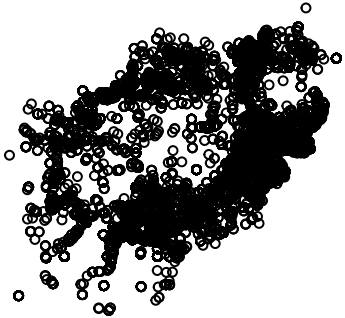


Let's load the *violence data* into R, using `read.csv()`:

```
violence = read.csv("Data/Violence/xSub_GED_AFG_event.csv")
```

To convert this table into a spatial object, we will use the `sf::st_as_sf()` command, specifying columns corresponding to geographic coordinates (`LONG`, `LAT`), and copying the CRS from the `borders` object:

```
violence = sf::st_as_sf(violence, coords=c("LONG","LAT"))
sf::st_crs(violence) = sf::st_crs(borders)
plot(violence["geometry"])
```



Clipping the raster to Afghanistan's extent

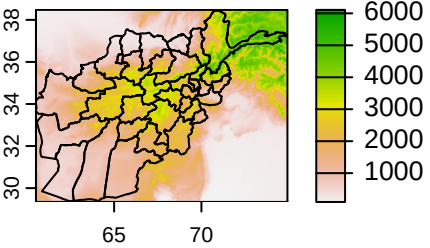
R's equivalent of QGIS's `Clip Raster by Extent...` command is the following:

1. Create a rectangular polygon corresponding to the extent of the `borders`
   - `sf::st_bbox()` finds the spatial extent of a layer
   - `sf::st_as_sfc()` converts the extent object into a polygon

```
extent_afg = sf::st_as_sfc(sf::st_bbox(borders))
```

2. Crop the raster by this new polygon, using `terra::crop()`

```
elevation = terra::crop(elevation_global, extent_afg)
plot(elevation); plot(borders["geometry"], add=TRUE)
```



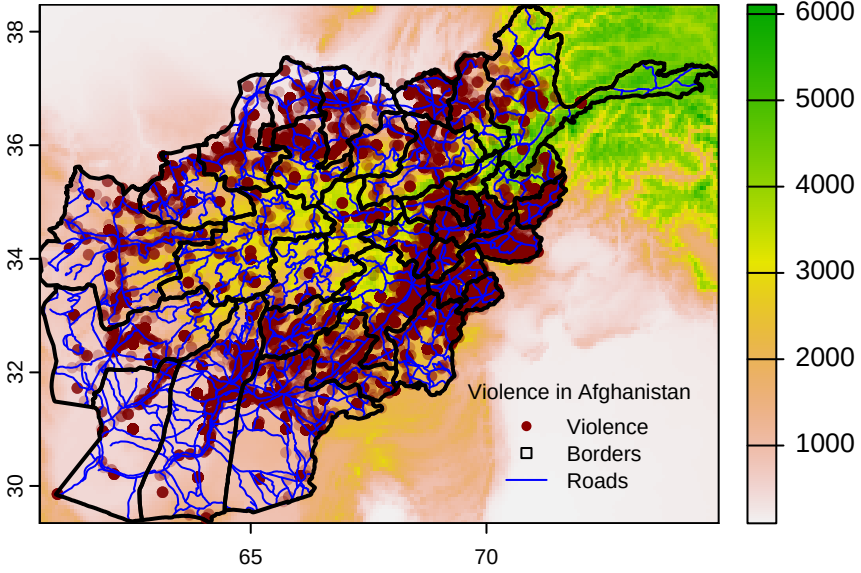
Plotting Map 1

To create the map, we will execute the following commands in sequence:

- set margins, `par()`
- draw elevation layer, `plot()` (note different syntax for rasters vs. vectors)
- add violence (transluscent dark red points)
- add borders, roads (adjusting thickness, color)
- add `legend()` (specify x and y coordinates for top-left corner of legend)

```
par(mar=c(2,2,.5,.5))
plot(elevation)
plot(violence["geometry"],
plot(borders["geometry"],add=T,lwd=2)
plot(roads["geometry"],add=T,col="blue")
  col=rgb(red = .5,green = 0,blue = 0,alpha = .25),pch=16, add=T)
legend(x=70.5,y=31.5,legend=c("Violence","Borders","Roads"),
  pch=c(16,0,NA),lwd=c(NA,NA,1),col=c("darkred","black","blue"),
  title="Violence in Afghanistan",bty="n")
```

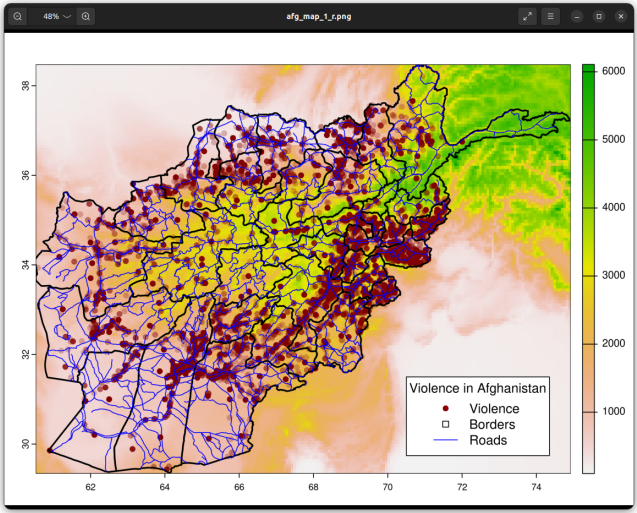
Here's what the plot should look like:



Exporting Map 1 to image file

To save the map, we will use the `png()` and `dev.off()` commands, as in PS01:

```
png("Output/afg_map_1_r.png",width = 8,height = 6,
  units = "in",res = 300)
par(mar=c(2,2,.5,.5))
plot(elevation)
plot(violence["geometry"],
plot(borders["geometry"],add=T,lwd=2)
plot(roads["geometry"],add=T,col="blue")
  col=rgb(red = .5,green = 0,blue = 0,alpha = .25),pch=16, add=T)
legend(x=70.5,y=31.5,legend=c("Violence","Borders","Roads"),
  pch=c(16,0,NA),lwd=c(NA,NA,1),col=c("darkred","black","blue"),
  title="Violence in Afghanistan",bty="n")
dev.off()
```



The output file should look like this.

Map 2

Loading ethnicity data

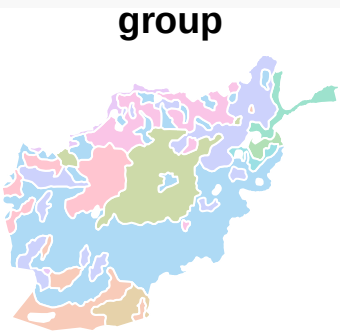
Let's load the *ethnic settlements data* into R, using `sf::read_sf()`:

```
ethnicity = sf::read_sf("Data/Ethnicity/GeoEPR_AFG.geojson")
plot(ethnicity["geometry"])
```

Plotting Map 2

Let's begin by creating a vector of colors, corresponding to the group column:

```
groups = as.factor(ethnicity$group)
n = length(unique(groups))
cols = hcl.colors(n,palette="Pastel 1")[groups]
plot(ethnicity["group"],col=cols,border="white")
```



To see what other color palettes are available in `hcl.colors()`, see `hcl.pals()`:

```
hcl.pals()
```

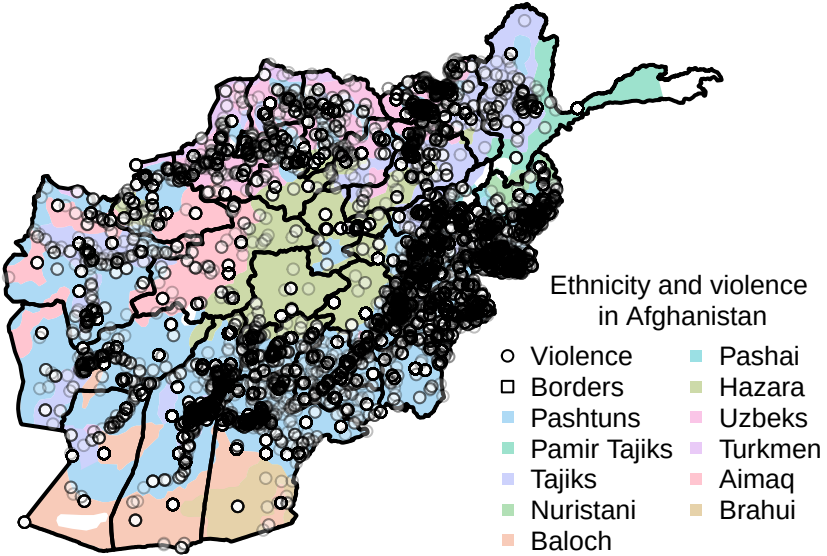
```
##   [1] "Pastel 1"      "Dark 2"        "Dark 3"        "Set 2"
##   [5] "Set 3"         "Warm"          "Cold"          "Harmonic"
##   [9] "Dynamic"       "Grays"         "Light Grays"   "Blues 2"
##  [13] "Blues 3"       "Purples 2"     "Purples 3"     "Reds 2"
##  [17] "Reds 3"        "Greens 2"      "Greens 3"      "Oslo"
##  [21] "Purple-Blue"   "Red-Purple"    "Red-Blue"      "Purple-Orange"
##  [25] "Purple-Yellow" "Blue-Yellow"   "Green-Yellow"  "Red-Yellow"
##  [29] "Heat"          "Heat 2"        "Terrain"       "Terrain 2"
##  [33] "Viridis"       "Plasma"        "Inferno"       "Rocket"
##  [37] "Mako"          "Dark Mint"     "Mint"          "BluGrn"
##  [41] "Teal"          "TealGrn"       "Emrld"         "BluYl"
##  [45] "ag_GrnYl"      "Peach"         "PinkYl"        "Burg"
##  [49] "BurgYl"        "RedOr"         "OrYel"         "Purp"
##  [53] "PurpOr"        "Sunset"        "Magenta"       "SunsetDark"
##  [57] "ag_Sunset"     "BrwnYl"        "YlOrRd"        "YlOrBr"
##  [61] "OrRd"          "Oranges"       "YlGn"          "YlGnBu"
##  [65] "Reds"          "RdPu"          "PuRd"          "Purples"
##  [69] "PuBuGn"        "PuBu"          "Greens"        "BuGn"
##  [73] "GnBu"          "BuPu"          "Blues"         "Lajolla"
##  [77] "Turku"         "Hawaii"        "Batlow"        "Blue-Red"
##  [81] "Blue-Red 2"    "Blue-Red 3"    "Red-Green"     "Purple-Green"
##  [85] "Purple-Brown"  "Green-Brown"   "Blue-Yellow 2" "Blue-Yellow 3"
##  [89] "Green-Orange"  "Cyan-Magenta"  "Tropic"        "Broc"
##  [93] "Cork"          "Vik"           "Berlin"        "Lisbon"
##  [97] "Tofino"        "ArmyRose"      "Earth"         "Fall"
## [101] "Geyser"        "TealRose"      "Temps"         "PuOr"
## [105] "RdBu"          "RdGy"          "PiYG"          "PRGn"
## [109] "BrBG"          "RdYlBu"        "RdYlGn"        "Spectral"
## [113] "Zissou 1"      "Cividis"       "Roma"
```

To create the map, we will use the same commands as above, with some differences:

- set margins to zero
- include option `reset=FALSE` when plotting the `ethnicity` layer
- add violence data twice (white points + black outlines)
- add `legend()` (including the vector of group colors)

```
par(mar=c(0,0,0,0))
plot(ethnicity["group"],col=cols,border=NA,reset=FALSE,main="")
plot(borders["geometry"],add=T,lwd=2)
# Add violence (white points + black outlines)
plot(violence["geometry"],col=rgb(1,1,1,alpha = .25),pch=16,add=T)
plot(violence["geometry"],col=rgb(0,0,0,alpha = .25),pch=1,add=T)
# Add legend
legend("bottomright",bty="n",ncol=2,
  legend=c("Violence","Borders",as.character(groups)),
  col=c("black","black",cols),pch=c(1,0,rep(15,length(cols))),
  cex=.8,title="Ethnicity and violence\n in Afghanistan"
  )
```

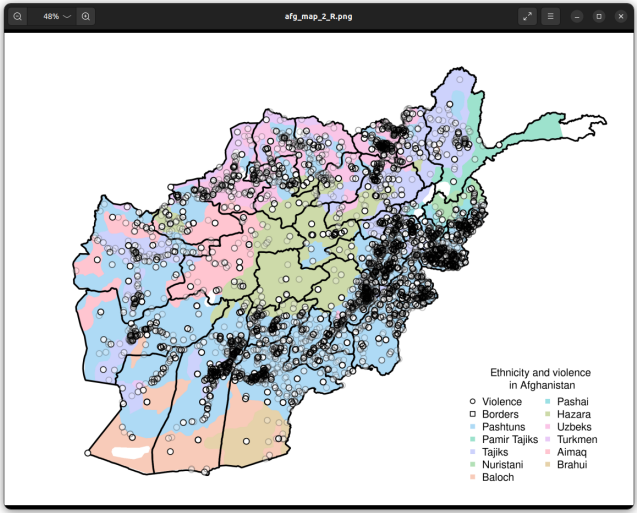
Here's what the plot should look like:



Exporting Map 2 to image file

Using the `png()` and `dev.off()` commands as bookends:

```
png("Output/afg_map_2_r.png",width = 8,height = 6,
  units = "in",res = 300)
par(mar=c(0,0,0,0))
plot(ethnicity["group"],col=cols,border=NA,reset=FALSE,main="")
plot(borders["geometry"],add=T,lwd=2)
# Add violence (white points + black outlines)
plot(violence["geometry"],col=rgb(1,1,1,alpha = .25),pch=16,add=T)
plot(violence["geometry"],col=rgb(0,0,0,alpha = .25),pch=1,add=T)
# Add legend
legend("bottomright",bty="n",ncol=2,
  legend=c("Violence","Borders",as.character(groups)),
  col=c("black","black",cols),pch=c(1,0,rep(15,length(cols))),
  cex=.8,title="Ethnicity and violence\n in Afghanistan")
dev.off()
```



The output file should look like this.

Problem Set 2

Your assignment (if using R):

- create the same kind of "violence and ethnicity" map for Yemen
- use these files:
  - `GeoEPR_YEM.geojson`
  - `xSub_GED_YEM_event.csv`
  - `gadm36_YEM_1.geojson`
- follow the same steps as for "Map 2" here
- name it `yem_map_2_r.png`
- upload map to Canvas (by next Wednesday)

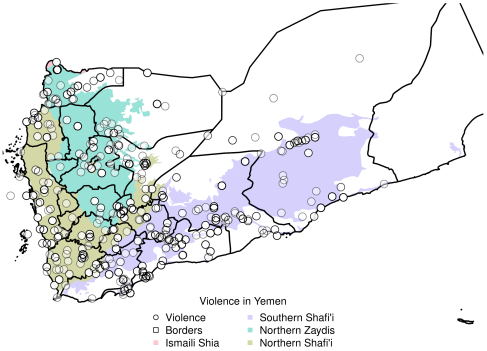


Figure 8: Can you make this map?