Additional Layers

Introduction

Until now, we have learnt general concepts about geospatial visualization in independent lessons. The **modular** approach of `{ggplot2}` allows to successively add all of them in different **layers**.

However, these enriched thematic maps also require to contain **text** and **labels** referring to specific places or regions, and important map elements like **scale bars** and a **north arrow**, as will be illustrated in this part.

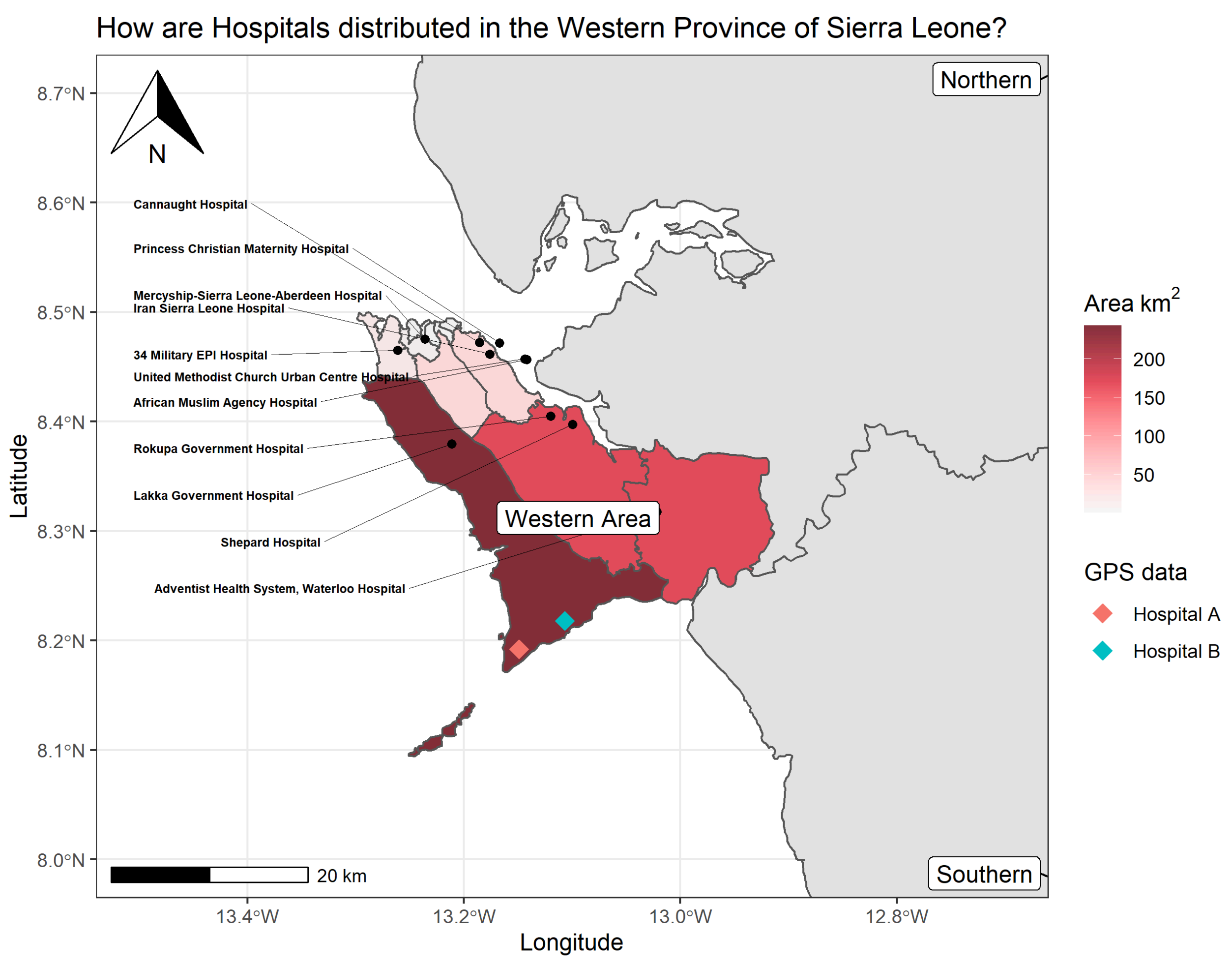

How are Hospitals distributed in the Western Province of Sierra Leone?
N
Northern
Southern
Western Area
Cannaught Hospital
Princess Christian Maternity Hospital
Mercyship-Sierra Leone-Aberdeen Hospital
Iran Sierra Leone Hospital
34 Military EPI Hospital
United Methodist Church Urban Centre Hospital
African Muslim Agency Hospital
Rokupa Government Hospital
Lakka Government Hospital
Shepard Hospital
Adventist Health System, Waterloo Hospital
20 km
8.7°N
8.6°N
8.5°N
8.4°N
8.3°N
8.2°N
8.1°N
8.0°N
Latitude
13.4°W
13.2°W
13.0°W
12.8°W
Longitude
Area km²
200
150
100
50
GPS data
Hospital A
Hospital B

Figure 1. Ggplot map with multiple layers.

In this lesson, we will learn to use functions from the `{ggspatial}` and `{ggsflabel}` packages to add all of these additional elements to our maps!

Learning objectives

1. Access **hospital** data coordinates using the `{afrihealthsites}` package.
2. Add text and **labels** to ggplot maps using the `{ggsflabel}` package.
3. Add arrow and scale **annotations** in ggplot maps using the `{ggspatial}` package.

Prerequisites

This lesson requires the following packages:

```
if(!require('pacman')) install.packages('pacman')

pacman::p_load(tidyverse,
               colorspace,
               ggspatial,
               janitor,
               ggplot2,
               readxl,
               spData,
               units,
               here,
               sf)

pacman::p_load_gh("yutannihilation/ggsflabel",
                  "afrimapr/afrihealthsites",
                  "afrimapr/afrilearndata",
                  "wmgeolab/rgeoboundaries")
```

Build an informative map

We will create a map to figure out how Tier 3 Hospitals are distributed in the Western Province of Sierra Leone. For this, we are going to retrieve real Hospital information from a public repository.

Access hospital sites with `{afrihealthsites}`

Instead of looking up coordinates manually, the package `{afrihealthsites}` provides a function `afrihealthsites()`, which allows to retrieve geographic coordinates of African health facilities from different sources.

You can run the following code to retrieve geographic coordinates of all the health facilities in Sierra Leone available in the who database:

```
sle_healthsites_all <-
  afrihealthsites(country = "Sierra Leone",
                  datasource = 'who',
                  plot = FALSE) %>%
  janitor::clean_names()

sle_healthsites_all
```

```
## Simple feature collection with 1120 features and 10 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -13.271 ymin: 6.9679 xmax: -10.3065 ymax: 9.9745
## Geodetic CRS:  WGS 84
## # A tibble: 1,120 × 11
##    country      admin1  facility_name        facility_type
##  * <chr>        <chr>   <chr>                <chr>
##  1 Sierra Leone Eastern Ahmadiyya Mission H… Mission Hospit…
##  2 Sierra Leone Eastern Baama Community Hea… Community Heal…
##  3 Sierra Leone Eastern Baiama Community He… Community Heal…
##  4 Sierra Leone Eastern Baiima Community He… Community Heal…
##  5 Sierra Leone Eastern Baiwala Community H… Community Heal…
##  6 Sierra Leone Eastern Bambara Kaima Commu… Community Heal…
##  7 Sierra Leone Eastern Bambara Maternal an… Maternal & Chi…
##  8 Sierra Leone Eastern Bambawulo Community… Community Heal…
##  9 Sierra Leone Eastern Bandajuma Community… Community Heal…
## 10 Sierra Leone Eastern Bandajuma Kpolihun … Community Heal…
## # ℹ 1,110 more rows
## # ℹ 7 more variables: ownership <chr>, ll_source <chr>, …
```

RSTUDIO CLOUD

Access to all the health facilities of Zimbabwe from the healthsites data source using the `afrihealthsites()` function.

SIDE NOTE

According to the *three-tier* health delivery classification, the highest tier (Tier 3) includes county hospitals in rural regions and city hospitals in urban regions, which are responsible for most inpatient services as well as teaching and research missions (Wang, 2021).}

We can now keep the Tier 3 health facilities inside the Western province:

```r
sle_healthsites_set <-

  sle_healthsites_all %>%

  filter(admin1 == "Western Area") %>% # 👈👈👈👈👈👈👈👈👈👈👈👈👈👈👈
  filter(tier == 3)
```

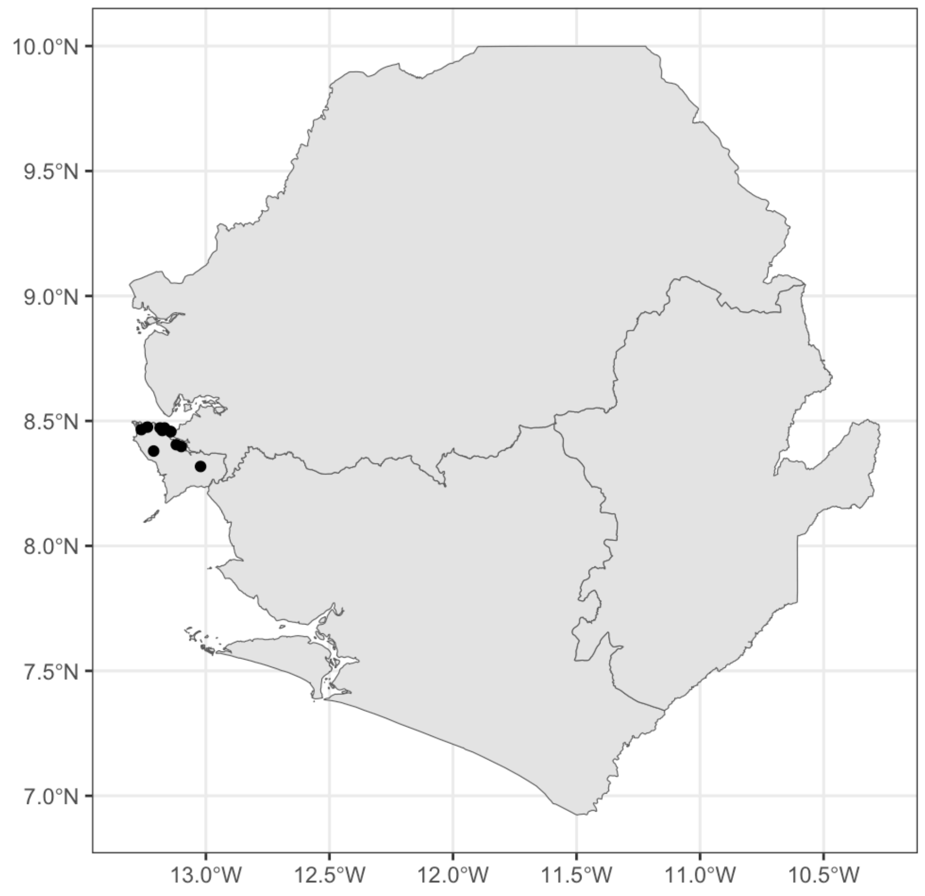
These sites belong to Sierra Leone. Let's get their province boundaries with `{rgeoboundaries}`:

```r
sierra_leone <- geoboundaries(country = "Sierra Leone", adm_lvl = 1)
```

```
## [1] "WARNING: geoBoundaries now provides two only types of boundaries:
simplified and unsimplified.All other types are deprecated. If you selected
SSCGS or SSCU it will be changed to simlified, HPSCU will be changed to
usimplifed "
```

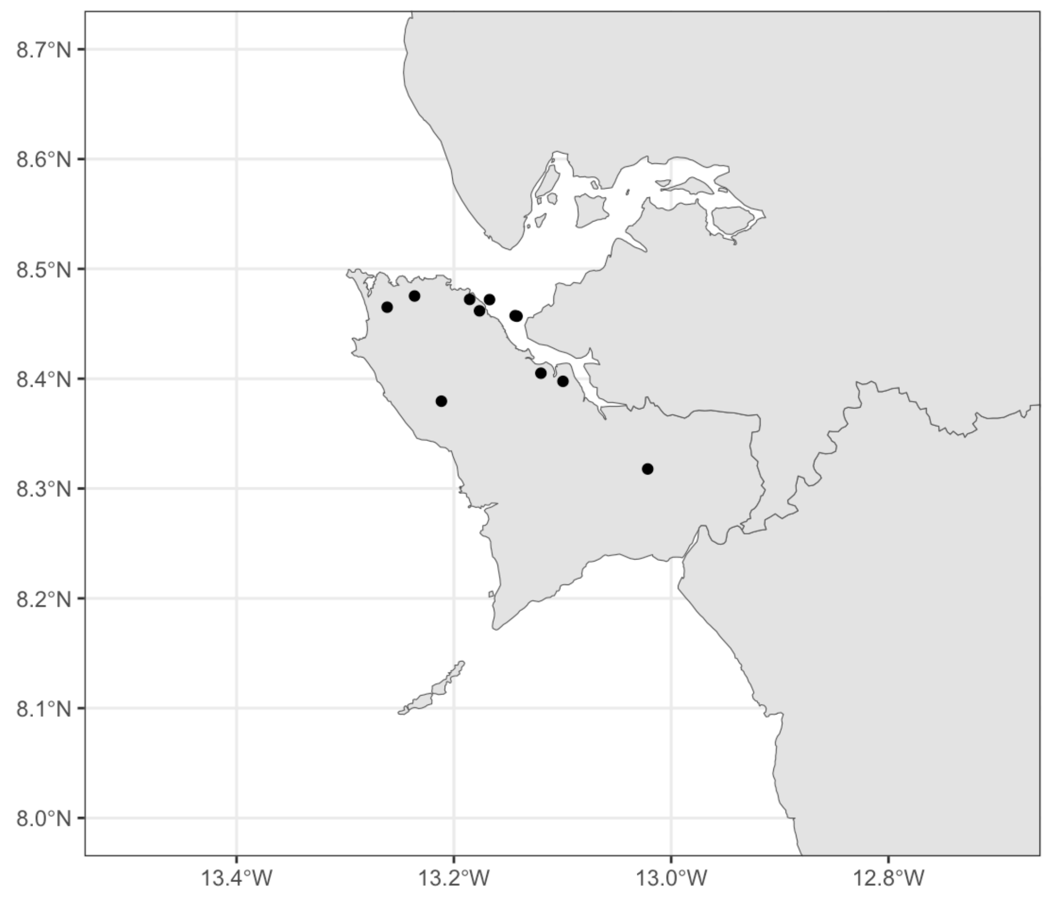
Now we can make a map with `sierra_leone` and `sle_healthsites_set` objects:

```r
ggplot() +

  # country map
  geom_sf(data = sierra_leone) +

  # hospital points
  geom_sf(data = sle_healthsites_set)
```



We can use `coord_sf()` after all `geom_sf()` calls, to **zoom in** to our area of interest inside Sierra Leone:

```
ggplot() +

  # country map
  geom_sf(data = sierra_leone) +

  # hospital points
  geom_sf(data = sle_healthsites_set) +

  # map extent
  coord_sf(xlim = c(-13.5,-12.7), ylim = c(8.0,8.7)) # 👈👈👈👈👈👈👈
```



RECAP

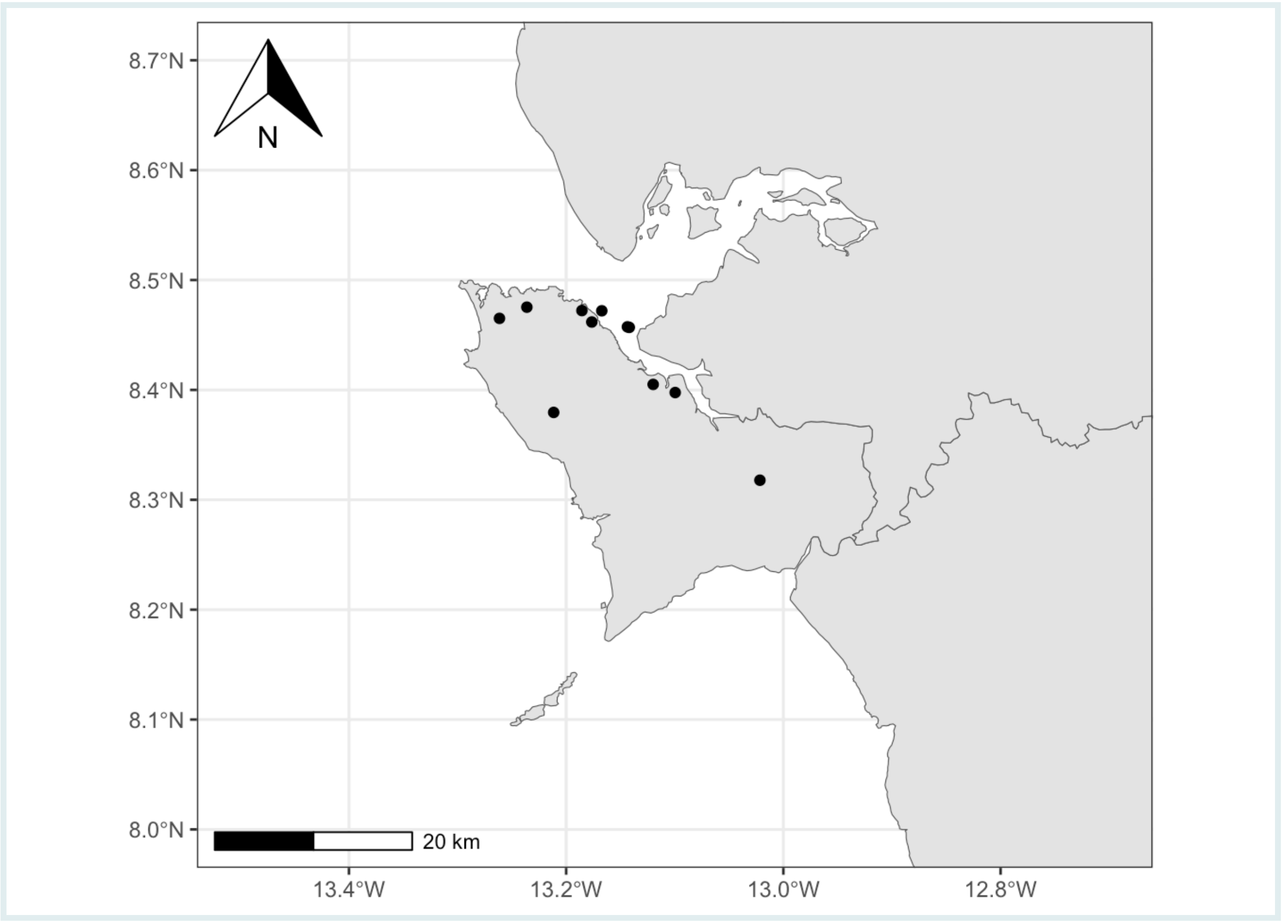
`sf` objects can handle on the fly the **coordinate system** (both *projection* and *extent*), which can be very useful if more than one spatial objects are not in the same projection.

Add a Scale bar and North arrow with `{ggspatial}`

Here we introduce the `{ggspatial}` package, which provides easy-to-use functions to create a scale bar and north arrow on a `ggplot` map:

- `annotation_north_arrow()` allows to add the north symbol and
- `annotation_scale()` a scale bar.

```r
ggplot() +

  # country map
  geom_sf(data = sierra_leone) +

  # hospital points
  geom_sf(data = sle_healthsites_set) +

  # map annotations
  annotation_north_arrow(location="tl") + # 👈👈👈👈👈👈👈👈👈👈👈👈
  annotation_scale(location="bl") +

  # map extent
  coord_sf(xlim = c(-13.5,-12.7), ylim = c(8.0,8.7))
```



The **location** of the scale bar and north arrow are by default in the bottom left (`"bl"`) side of the map. They can be specified using the `location` argument with `"tr"` for top right, `"bl"` for bottom left, etc.

PRACTICE
(in RMD)

The `zimbabwe_adm1` object contains the boundaries of all the provinces in Zimbabwe.

```
zimbabwe_adm1 <- geoboundaries(country = "Zimbabwe", adm_lvl =
1)
```

```
## [1] "WARNING: geoBoundaries now provides two only types of
boundaries: simplified and unsimplified.All other types are
deprecated. If you selected SSCGS or SSCU it will be changed
to simlified, HPSCU will be changed to usimplifed "
```

PRACTICE

(in RMD)

To this `ggplot` map with the `zimbabwe_adm1` object:

Add a Scale bar located in the `bottom right` of the map, and a North arrow in the `top left`.

```
q4 <-
  zimbabwe_adm1 %>%
  ggplot() +
  geom_sf() +
  _________ +
  _________
q4
```

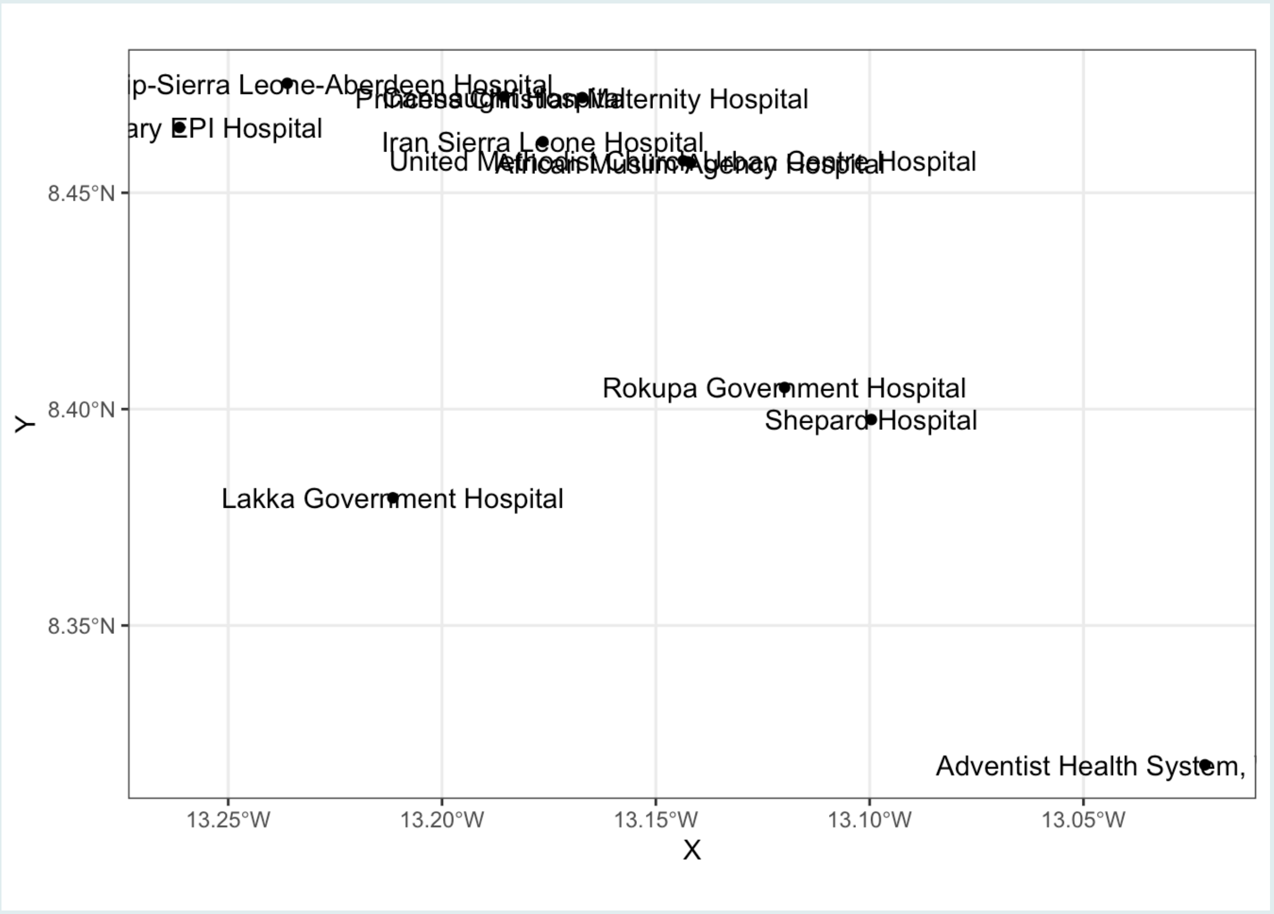
SIDE NOTE

- The North arrow style in `annotation_north_arrow()` can also be adjusted using the `style` argument.
- Note that scale distance is set to `"km"` by default in `annotation_scale()`; you can set it in “m”, “cm”, “mi”, “ft”, or “in”.

Add hospital names with `geom_sf_text_repel()`

We add hospital locations and names as text on the map with `geom_sf_text()`:

```
sle_healthsites_set %>%
  ggplot() +
  # geometry
  geom_sf() +
  # label
  geom_sf_text(mapping = aes(label= facility_name)) # 👈👈👈👈👈👈👈
```

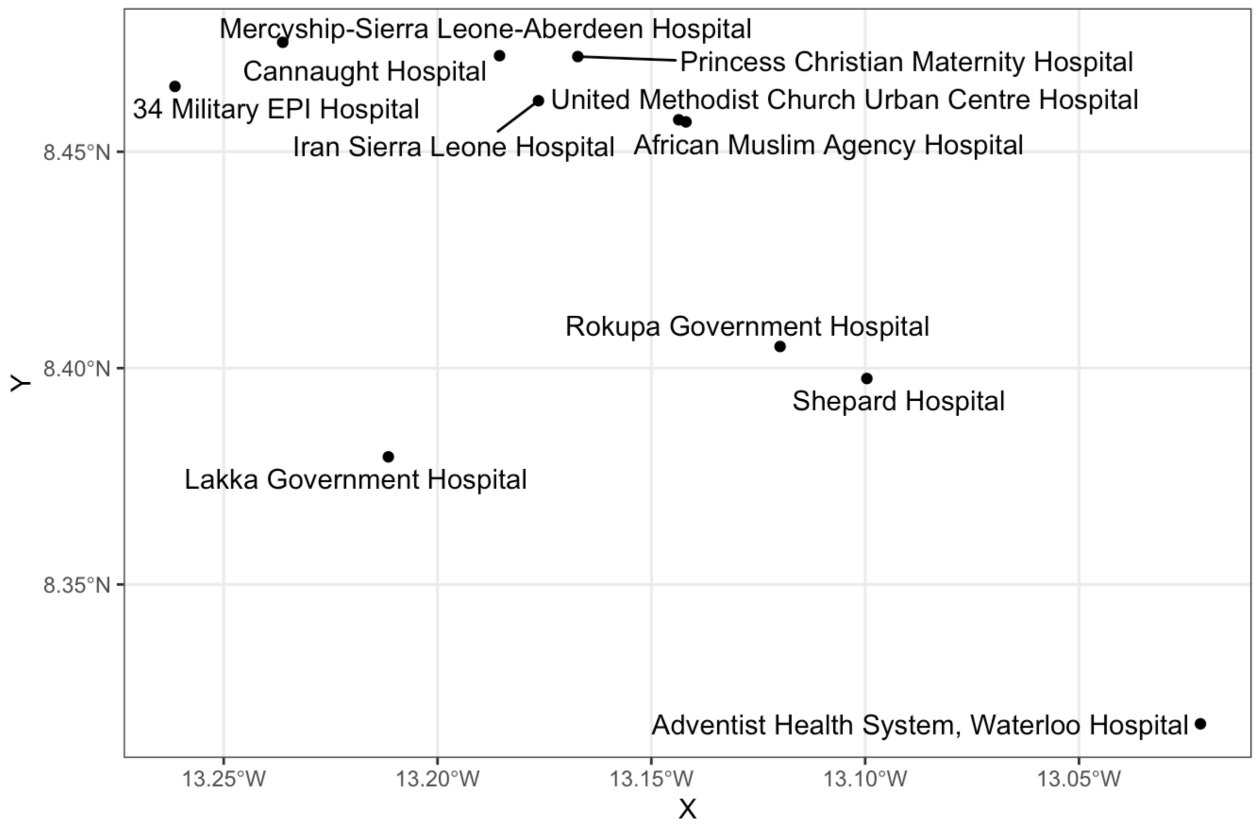


ip-Sierra Leone-Aberdeen Hospital
Princess Christian Maternity Hospital
ary EPI Hospital
Iran Sierra Leone Hospital
United Methodist Church Urban Centre Hospital
African Muslim Agency Hospital
Rokupa Government Hospital
Shepard Hospital
Lakka Government Hospital
Adventist Health System,
8.45°N
8.40°N
8.35°N
Y
13.25°W
13.20°W
13.15°W
13.10°W
13.05°W
X

This is not really satisfactory, as the names overlap on the points, and they are not easy to read on the grey background. The `{ggsflabel}` package offers a very flexible approach (inspired in `{ggrepel}`) to deal with label placement in `ggplot` maps with `sf` objects (with `geom_sf_text_repel` and `geom_sf_label_repel`), including automated movement of labels in case of overlap.

We use it here to “nudge” the labels away, and connect them to the city locations:

```
sle_healthsites_set %>%
  ggplot() +
  # geometry
  geom_sf() +
  # label
  geom_sf_text_repel(mapping = aes(label= facility_name)) # 👈👈👈👈
```


Mercyship-Sierra Leone-Aberdeen Hospital
Cannaught Hospital
Princess Christian Maternity Hospital
34 Military EPI Hospital
United Methodist Church Urban Centre Hospital
Iran Sierra Leone Hospital
African Muslim Agency Hospital
Rokupa Government Hospital
Shepard Hospital
Lakka Government Hospital
Adventist Health System, Waterloo Hospital
8.45°N
8.40°N
8.35°N
Y
13.25°W
13.20°W
13.15°W
13.10°W
13.05°W
X

PRACTICE

(in RMD)

This is `zimbabwe_hospitals` contains all the `hospital` facilities of `Zimbabwe`:

```
zimbabwe_hospitals <-
  afrihealthsites(country = "Zimbabwe",
                  datasource = 'healthsites',
                  plot = FALSE) %>%
  filter(amenity == "hospital")

zimbabwe_hospitals
```

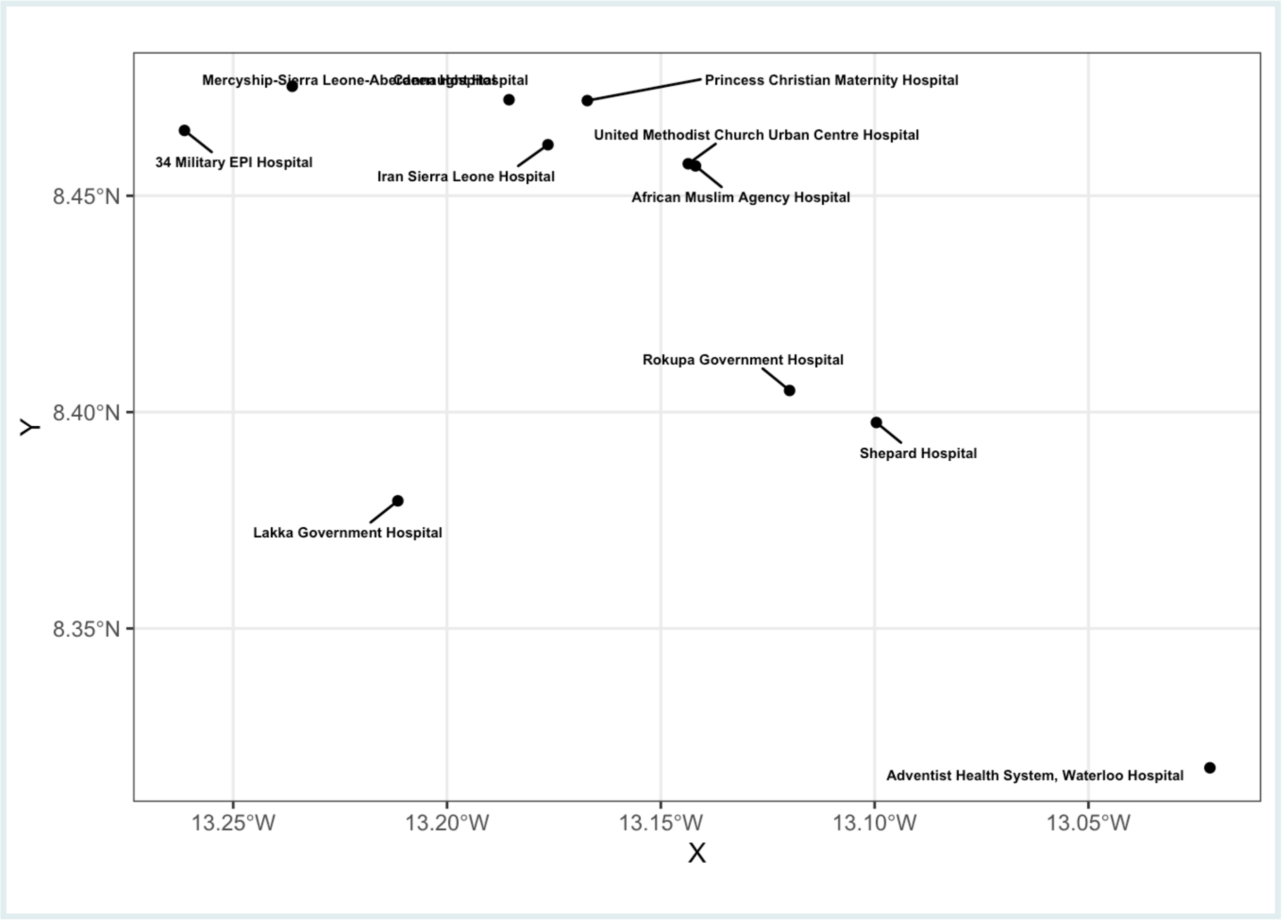
Add the name of these facilities as text without overlaps using the `geom_sf_text_repel()` function.

```
q5 <-
  zimbabwe_hospitals %>%
  ggplot() +
  geom_sf() +
  _______()
q5
```

Additionally, we can manually set {ggrepel} arguments to improve its output:

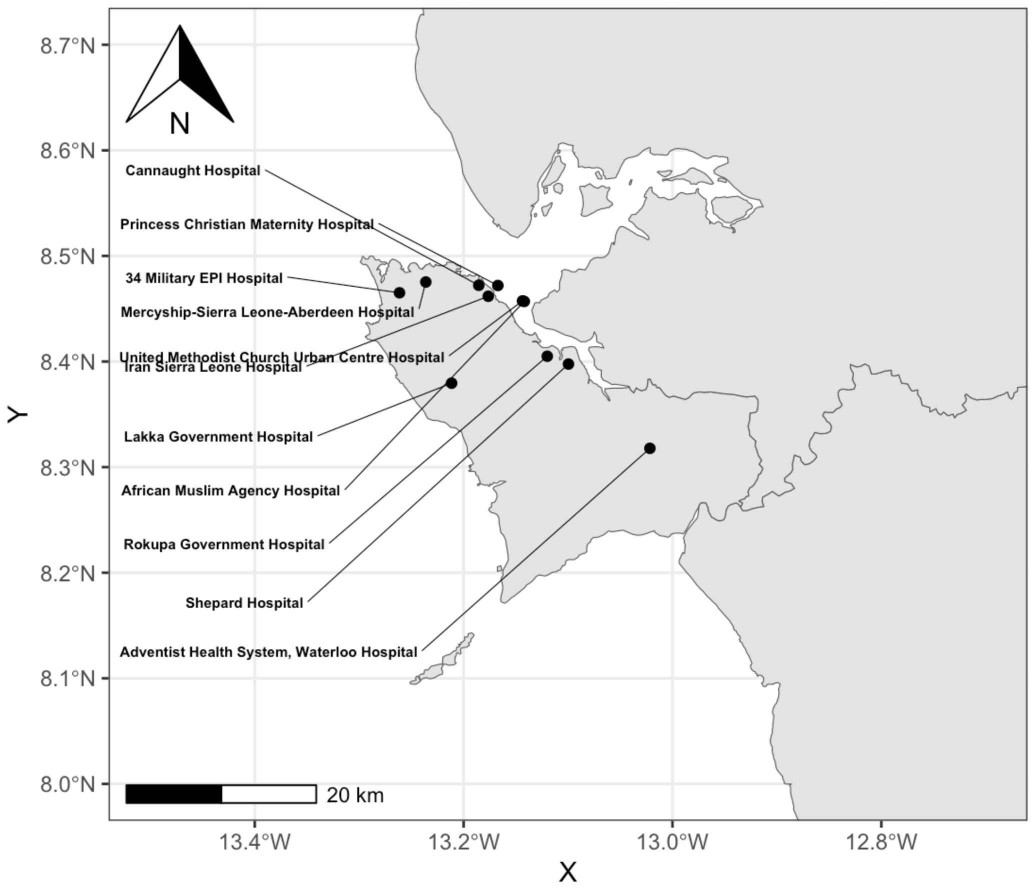
- The size (argument `size`);
- The type of font of the text (`fontface`);
- The force of repulsion between overlapping text labels (`force`);
- The additional padding around the each text label (`box.padding`).

```r
sle_healthsites_set %>%
  ggplot() +
  # geometry
  geom_sf() +
  # label
  geom_sf_text_repel(mapping = aes(label= facility_name),
                     size = 2,         # 👈👈👈👈👈👈👈👈👈👈👈👈👈👈
                     fontface = "bold",
                     force = 40,
                     box.padding = 0.6)
```


Mercyship-Sierra Leone-Aberdeen
Connaught Hospital
Princess Christian Maternity Hospital
34 Military EPI Hospital
Iran Sierra Leone Hospital
United Methodist Church Urban Centre Hospital
African Muslim Agency Hospital
Rokupa Government Hospital
Shepard Hospital
Lakka Government Hospital
Adventist Health System, Waterloo Hospital
8.45°N
8.40°N
8.35°N
Y
13.25°W
13.20°W
13.15°W
13.10°W
13.05°W
X

Adding this layer to the current map, we obtain:

```r
ggplot() +

  # country map
  geom_sf(data = sierra_leone) +

  # hospital points
  geom_sf(data = sle_healthsites_set) +
  # hospital names with repelled text
  geom_sf_text_repel(data = sle_healthsites_set,          # 👈👈👈👈👈
                     mapping = aes(label = facility_name),
                     size          = 2,
                     fontface      = "bold",
                     force         = 0.5,
                     box.padding   = 0.6,
                     nudge_x       = -0.25,
                     direction     = "y",
                     hjust         = 1,
                     segment.size = 0.2) +

  # map annotation
  annotation_north_arrow(location="tl") +
  annotation_scale(location="bl") +

  # map extent
  coord_sf(xlim = c(-13.5,-12.7),
           ylim = c(8.0,8.7))
```



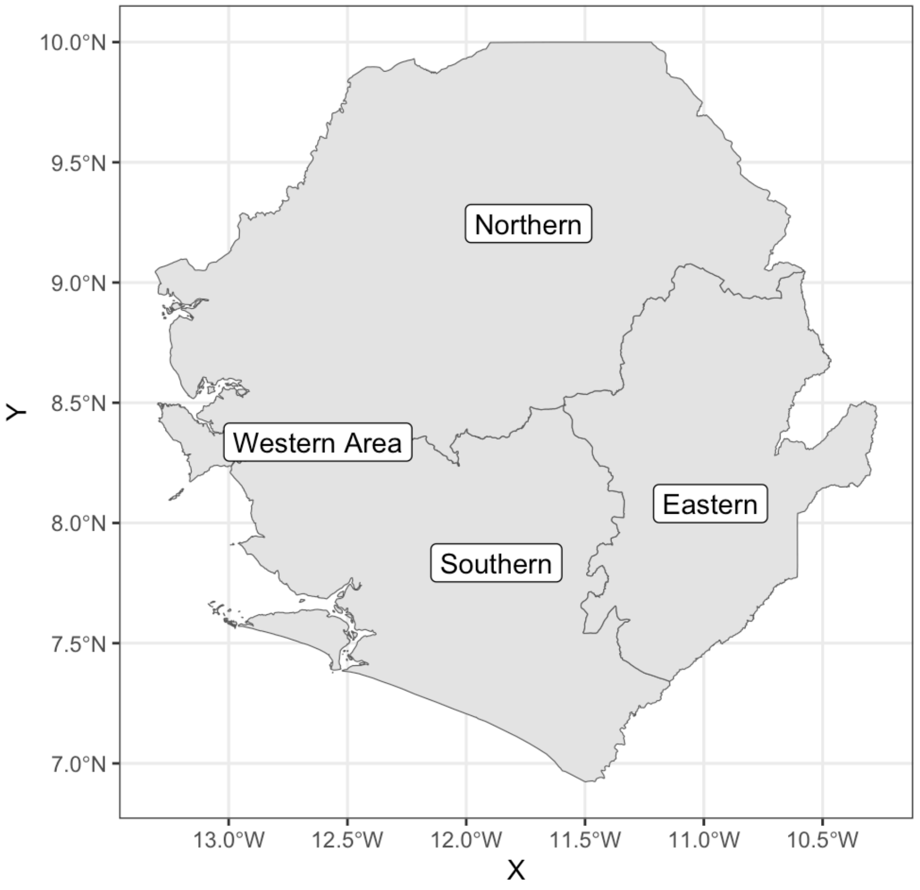
N
Cannaught Hospital
Princess Christian Maternity Hospital
34 Military EPI Hospital
Mercyship-Sierra Leone-Aberdeen Hospital
United Methodist Church Urban Centre Hospital
Iran Sierra Leone Hospital
Lakka Government Hospital
African Muslim Agency Hospital
Rokupa Government Hospital
Shepard Hospital
Adventist Health System, Waterloo Hospital
20 km
8.7°N
8.6°N
8.5°N
8.4°N
8.3°N
8.2°N
8.1°N
8.0°N
Y
13.4°W
13.2°W
13.0°W
12.8°W
X

Add province labels with `geom_sf_label_repel()`

It would be informative to add finer administrative information on top of the previous map, starting with administrative borders (`sf` object: polygon data) and their names. The package `ggsflabel` provides functions to add labels to `sf` objects (points and polygons) in maps.

Province names are part of this data, as the `shapeName` variable.

```r
sierra_leone %>%
  ggplot() +

  # country map
  geom_sf() +

  # province labels
  geom_sf_label_repel(aes(label=shapeName)) # 👈👈👈👈👈👈👈👈👈👈👈
```



PRACTICE (in RMD)

The `zimbabwe_adm1` object contains the boundaries of all the provinces in Zimbabwe.

```
zimbabwe_adm1 <- geoboundaries(country = "Zimbabwe", adm_lvl =
1)
```

```
## [1] "WARNING: geoBoundaries now provides two only types of
boundaries: simplified and unsimplified.All other types are
deprecated. If you selected SSCGS or SSCU it will be changed
to simlified, HPSCU will be changed to usimplifed "
```

PRACTICE (in RMD)

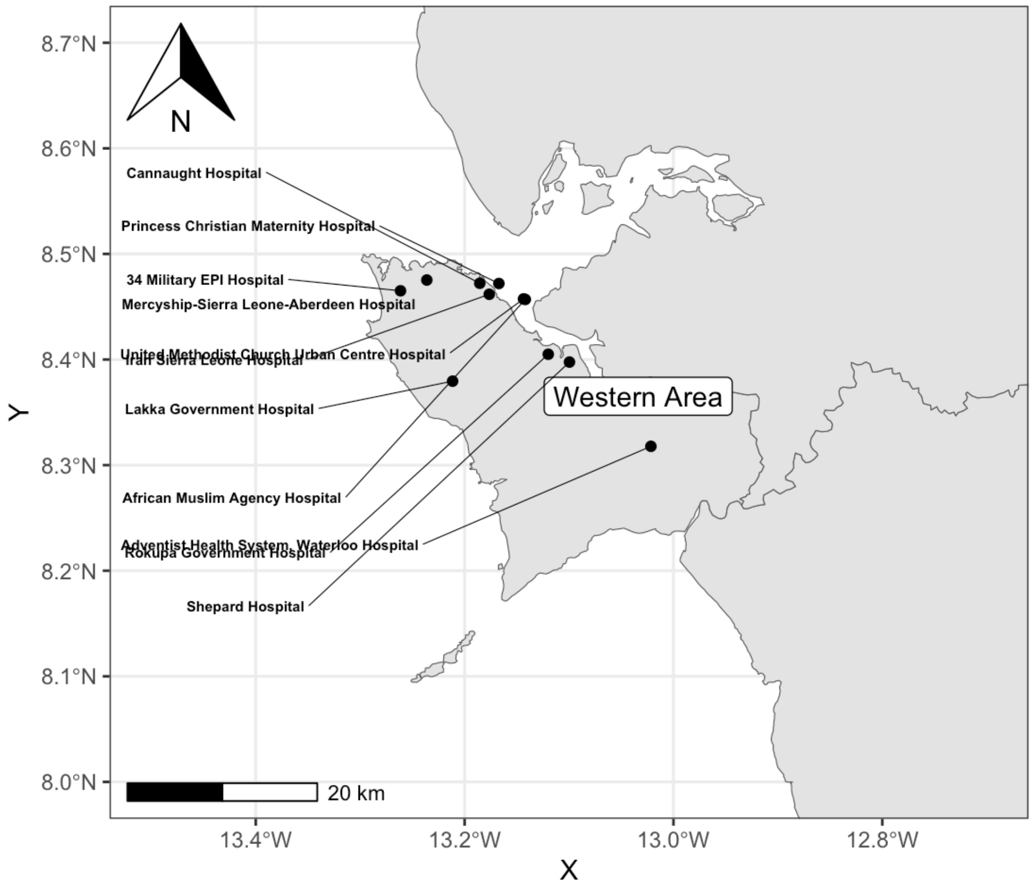
Create a `ggplot` map of Zimbabwe with labels for the name of each of its provinces.

```
q2 <-
  zimbabwe_adm1 %>%
  ggplot() +

  # country map
  geom_sf() +

  # province labels
  geom_sf_....._.....(.....)
q2
```

To continue building the complex map introduced at the beginning of the lesson, province data is directly plotted as an additional `sf` layer using `geom_sf()`. In addition, province names will be added using `geom_sf_label_repel()`, as well as the label (from `shapeName`), and a relatively big font size.

Since the Eastern province is not showed in the zoomed map, we dropped it using `filter()` from the `{dplyr}` package:

```r
ggplot() +

  # country map
  geom_sf(data = sierra_leone) +

  # province labels
  geom_sf_label_repel(data = sierra_leone %>%         # 👈👈👈👈👈👈
                      filter(shapeName!="Eastern"),
                      mapping = aes(label=shapeName)) +

  # hospital points
  geom_sf(data = sle_healthsites_set) +
  # hospital names with repelled text
  geom_sf_text_repel(data = sle_healthsites_set,
                     mapping = aes(label = facility_name),
                     size         = 2,
                     fontface     = "bold",
                     force        = 0.5,
                     box.padding  = 0.6,
                     nudge_x      = -0.25,
                     direction    = "y",
                     hjust        = 1,
                     segment.size = 0.2) +

  # map annotation
  annotation_north_arrow(location="tl") +
  annotation_scale(location="bl") +

  # map extent
  coord_sf(xlim = c(-13.5,-12.7), ylim = c(8.0,8.7))
```



8.7°N
8.6°N
8.5°N
8.4°N
8.3°N
8.2°N
8.1°N
8.0°N
Y
N
Cannaught Hospital
Princess Christian Maternity Hospital
34 Military EPI Hospital
Mercyship-Sierra Leone-Aberdeen Hospital
United Methodist Church Urban Centre Hospital
Iran Sierra Leone Hospital
Lakka Government Hospital
Western Area
African Muslim Agency Hospital
Adventist Health System, Waterloo Hospital
Rokupa Government Hospital
Shepard Hospital
20 km
13.4°W
13.2°W
13.0°W
12.8°W
X

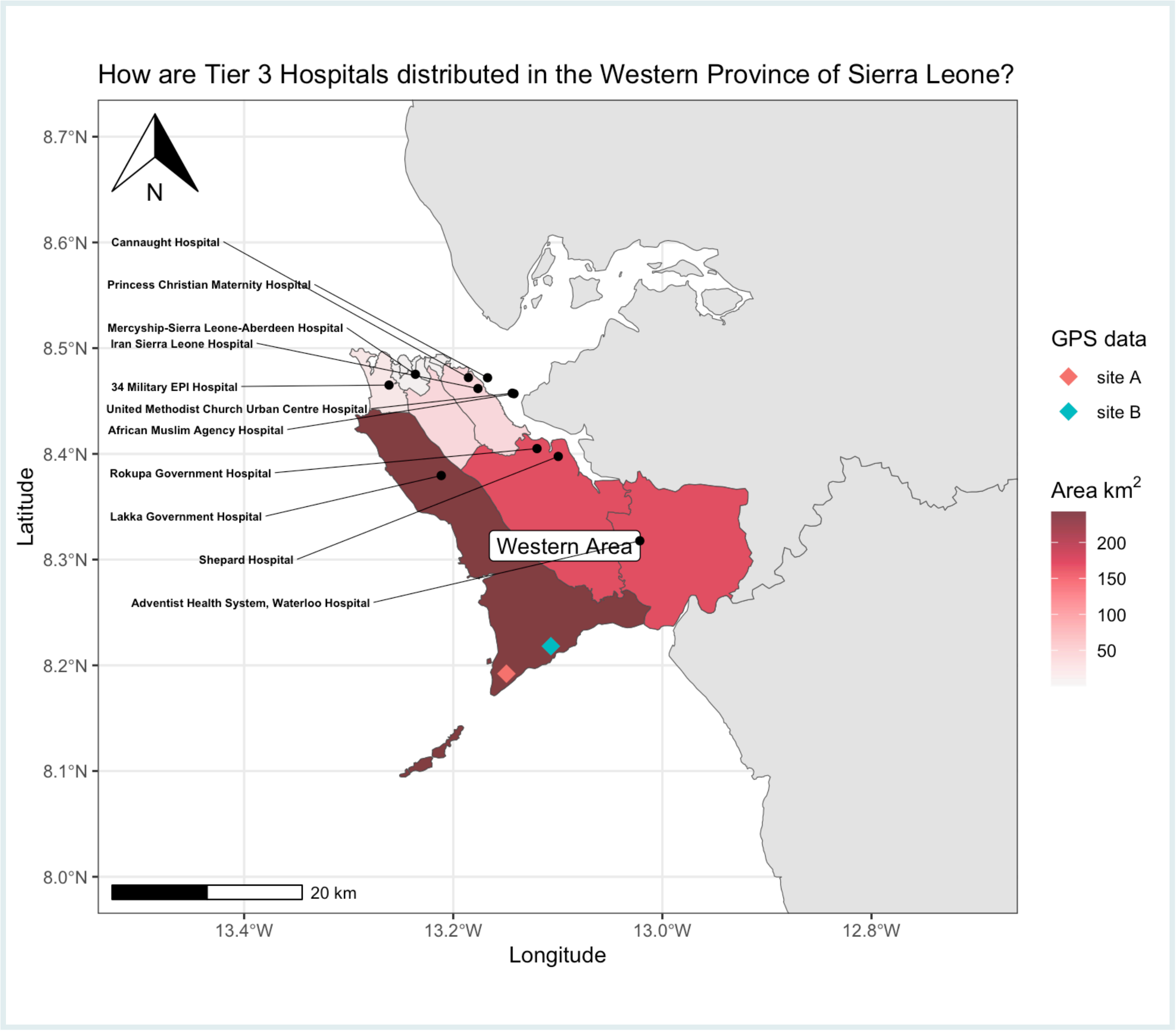
Final map

For the final map, we put everything together, having a country boundary map based on the Sierra Leona map, province labels, main hospital names and locations, custom GPS collected field sites, as well as a theme adjusted with axis labels, and a north arrow and scale bar.

We also added District boundaries from a local shapefile data to fill each polygon with respect to district area, and field sited recorded by a GPS device:

```r
# districts
sierra_leone_shp <-
  sf::read_sf(here("data/boundaries/sle_adm3.shp")) %>%
  filter(admin1Name=="Western")

# field site points
sites_sf <- readr::read_csv(
  here("data/gps_healthsites.csv")) %>%
  st_as_sf(coords = c("gps_longitude","gps_latitude"), crs = 4326)
```

```r
ggplot() +

  # country map
  geom_sf(data = sierra_leone) +

  # districts filled by area
  geom_sf(data = sierra_leone_shp, # 👈👈👈👈👈👈👈👈👈👈👈👈👈👈👈
          mapping = aes(fill = area_km2)) +
  # color scale for fill
  scale_fill_continuous_sequential(palette="Reds 3",
                                   alpha = 0.8) +
  # field site points
  geom_sf(data = sites_sf,              # 👈👈👈👈👈👈👈👈👈👈👈👈👈
          mapping = aes(color = gps_name),
          shape = 18, size = 4) +

  # province names with repelled labels
  geom_sf_label_repel(data = sierra_leone %>%
                        filter(shapeName!="Eastern"),
                      mapping = aes(label=shapeName)) +

  # hospital points
  geom_sf(data = sle_healthsites_set) +
  # hospital names with repelled text
  geom_sf_text_repel(data = sle_healthsites_set,
                     mapping = aes(label = facility_name),
                     size         = 2,
                     fontface     = "bold",
                     force        = 0.5,
                     box.padding  = 0.6,
                     nudge_x      = -0.25,
                     direction    = "y",
                     hjust        = 1,
                     segment.size = 0.2) +

  # map annotation
  annotation_north_arrow(location="tl") +
  annotation_scale(location="bl") +

  # map extent
  coord_sf(xlim = c(-13.5,-12.7),
           ylim = c(8.0,8.7)) +

  # ggplot labels
  labs(x = "Longitude",
       y = "Latitude",
       fill = expression(Area~km^2),
       color = "GPS data",
       title = "How are Tier 3 Hospitals distributed in the Western Province
of Sierra Leone?")
```



How are Tier 3 Hospitals distributed in the Western Province of Sierra Leone?
N
Cannaught Hospital
Princess Christian Maternity Hospital
Mercyship-Sierra Leone-Aberdeen Hospital
Iran Sierra Leone Hospital
34 Military EPI Hospital
United Methodist Church Urban Centre Hospital
African Muslim Agency Hospital
Rokupa Government Hospital
Lakka Government Hospital
Shepard Hospital
Adventist Health System, Waterloo Hospital
Western Area
20 km
8.7°N
8.6°N
8.5°N
8.4°N
8.3°N
8.2°N
8.1°N
8.0°N
Latitude
13.4°W
13.2°W
13.0°W
12.8°W
Longitude
GPS data
site A
site B
Area km2
200
150
100
50

Wrap up

This example fully demonstrates that adding **layers** on `ggplot2` is relatively straightforward, as long as the data is properly stored as an `sf` object. Adding additional layers like **external** data, color **palettes**, point or polygon **labels** and **map annotations** would simply follow the same logic, with additional calls *after* `geom_sf()` and at the *right place* in the `ggplot2` sequence.

Contributors

The following team members contributed to this lesson:

ANDREE VALLE CAMPOS

R Developer and Instructor, the GRAPH Network
Motivated by reproducible science and education

LAURE VANCAUWENBERGHE

Data analyst, the GRAPH Network
A firm believer in science for good, striving to ally programming, health and education

References

Some material in this lesson was adapted from the following sources:

- *Moreno, M., Basille, M. Drawing beautiful maps programmatically with R, sf and ggplot2 — Part 1: Basics*. (2018). Retrieved 01 June 2022, from https://r-spatial.org/r/2018/10/25/ggplot2-sf.html

- *Moreno, M., Basille, M. Drawing beautiful maps programmatically with R, sf and ggplot2 — Part 2: Layers*. (2018). Retrieved 01 June 2022, from https://r-spatial.org/r/2018/10/25/ggplot2-sf-2.html

- *Wilke, Claus O. Fundamentals of Data Visualization. Chapter 4: Color scales*. (2020). Retrieved 01 June 2022, from https://clauswilke.com/dataviz/color-basics.html#color-to-represent-data-values