First approach to service availability

Contents

This notebook is a first exploration of how SNIS and RBF data collected in DRC through dedicated DHIS2 portals can be used to measure availability of vaccination services. For each of these portals, all available metadata has been extracted using the package DHISExtractr. For the SNIS portal, the following elements have also been extracted:

- The list of indicators mentioning the initials "BCG"
- The list of facilities supposed to be reporting on at least one of these indicators.
- All available data for these indicators in these facilities for the years 2016-2017. Looking at the data, it turns out there has been a structural change in the level of data availability in 2017, and we thus dropped the (very) few available 2016 data.

For legibility of this notebook, all *housekeeping* code has been muted.

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/grlurton/data/dhis/snis/map_polygons.shp", layer: "map_polygons"
## with 533 features
## It has 1 fields
```

Data Completeness

We first want to look at the completeness of the data in the SNIS DHIS reporting. To do this by zone, we consider each facility in a given zone should have reported 12 times in the year 2017.

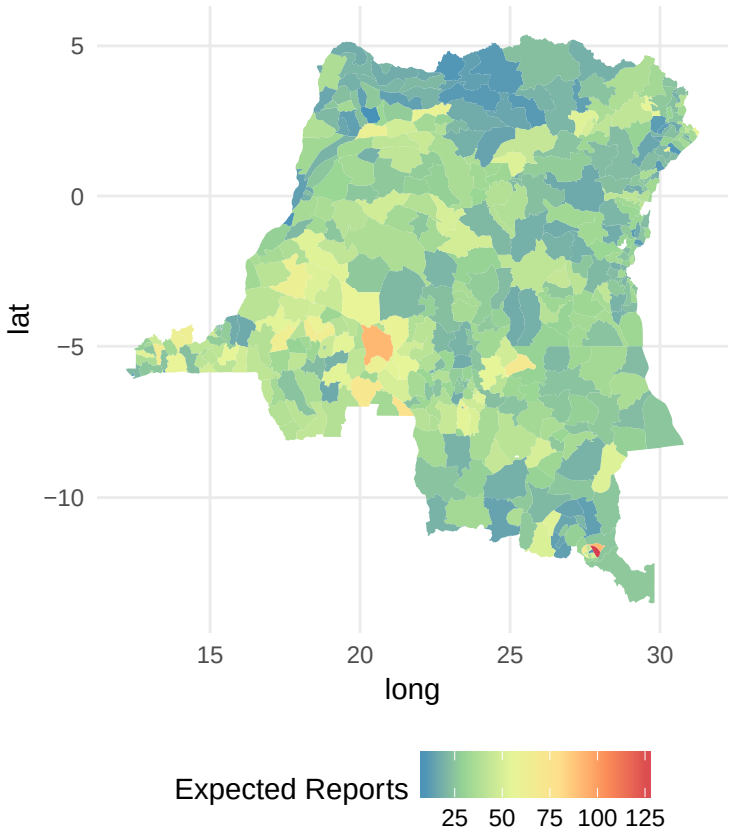
```
fosa_by_zone <- group_by(org_units_bcg, zone)
n_expected <- summarise(fosa_by_zone, n_expected = length(unique(as.character(units))))

map_expectation <- right_join(coordinates , n_expected, by = 'zone')

summary(n_expected$n_expected)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   22.00   29.00   31.18   37.00  129.00
```

```
ggplot(map_expectation)+
  geom_polygon(aes(x  = long, y = lat ,
                   group=zone,
                   fill = n_expected)) +
  theme_minimal() +
  coord_map()+
  scale_fill_distiller(name = "Expected Reports",
                       palette = "Spectral" , direction = -1)+
  theme(legend.position = "bottom")
```



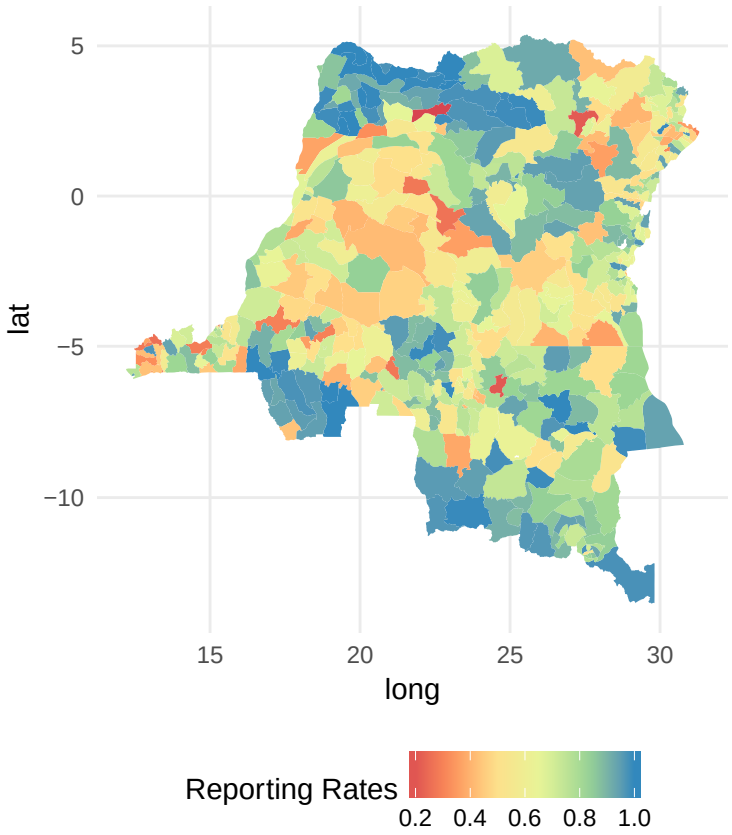
We can now compare this data to the number of reports actually obtained by zone on each of the target indicators.

```
data_by_zone_period <- group_by(data, zone , period)
n_reporting_period <- summarise(data_by_zone_period, n_reporting_monthly = length(unique(as.character(o

n_reporting_zone <- group_by(n_reporting_period, zone)
n_reporting_zone <- summarise(n_reporting_zone, n_reporting = sum(n_reporting_monthly))

n_reporting <- inner_join(n_reporting_zone , n_expected)
n_reporting$reporting_rate <- n_reporting$n_reporting / (12 * n_reporting$n_expected)

map_reporting_rates <- right_join(coordinates , n_reporting, by = 'zone')
ggplot(map_reporting_rates)+
  geom_polygon(aes(x  = long, y = lat ,
                   group=zone,
                   fill = reporting_rate)) +
  theme_minimal() +
  coord_map()+
  scale_fill_distiller(name = "Reporting Rates",
                       palette = "Spectral" , direction = 1)+
  theme(legend.position = "bottom")
```



BCG stockouts description

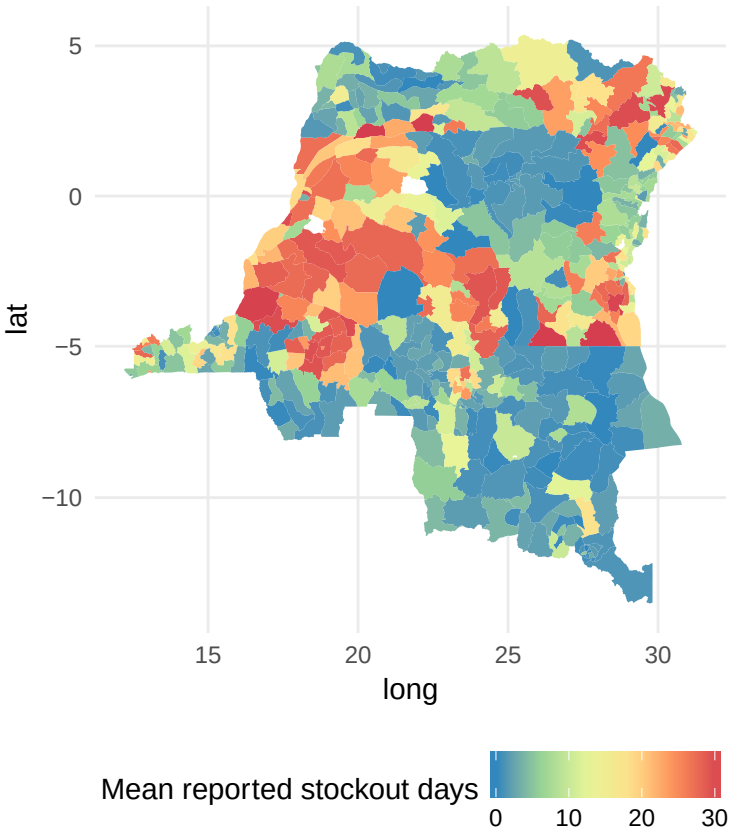
Based on reported stockouts

First, let's look at the reported stockouts.

```
data_rupture <- data[(data$name == 'B 6.1 BCG diluants - jours de rupture de stock' |
                        data$name == 'F 4.1 BCG diluants - Jours de rupture de stock') &
                            data$value < 32 ,]

stockout_by_zone <- group_by(data_rupture , zone)
mean_reported_stockout <- summarize(stockout_by_zone, mean_rupture = mean(value))

map_reported_stockout <- right_join(coordinates , mean_reported_stockout, by = 'zone')
ggplot(map_reported_stockout)+
  geom_polygon(aes(x  = long, y = lat ,
                   group=zone,
                   fill = mean_rupture)) +
  theme_minimal() +
  coord_map()+
  scale_fill_distiller(name = "Mean reported stockout days",
                       palette = "Spectral" , direction = -1)+
  theme(legend.position = "bottom")
```



PDSS data

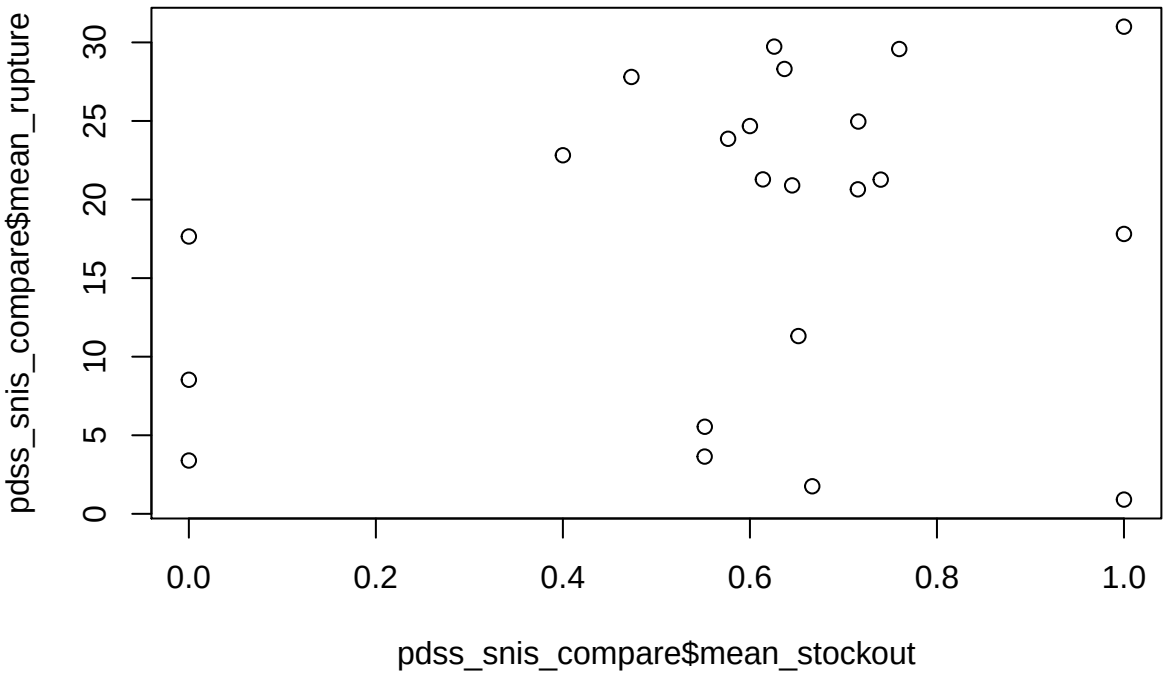
```
pdss_bcg <- data_pdss[as.character(data_pdss$name) %in% c('13.5.1 BCG, Penta, la poliomyélite, la fièvr

#'13.5.1 BCG',

pdss_bcg$stockout <- pdss_bcg$value == 0
pdss_bcg <- merge(pdss_bcg , matched_zones, by.x = 'zone' , by.y = 'id_pdss')
pdss_bcg$id_snis <- as.character(pdss_bcg$id_snis)


pdss_bcg_zone <- group_by(pdss_bcg , id_snis)
pdss_bcg_stockout <- summarise(pdss_bcg_zone, mean_stockout = mean(stockout))

pdss_snis_compare <- merge(pdss_bcg_stockout, mean_reported_stockout, by.x = 'id_snis', by.y = 'zone')
plot(pdss_snis_compare$mean_stockout , pdss_snis_compare$mean_rupture)
```


pdss_snis_compare$mean_rupture
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0
pdss_snis_compare$mean_stockout