DYB Summary Heatmaps

Shabrina Mardevi

```
knitr::opts_chunk$set(echo = TRUE)

library(dplyr)
library(janitor)
library(tidyverse)
library(ggplot2)
library(readxl)
library(openxlsx)
library(data.table)
library(pointblank)
library(haven)
library(naniar)
library(pointblank)
library(heatmaply)
```

Summary Heatmaps: Data Availability (All Categories) by Number of Years from 2009 to 2018 (Blue Heatmaps)

Prep Data

```
# Create function to :1.) Join two 5-year period data, and 2.) Add the n years of availability

join_transform <- function(table_1, table_2){
  table_all <- full_join(table_1, table_2, by = c("countries_areas", "data_available"))
  table_clean <-
    table_all %>%
    mutate_at(vars(n.x, n.y), ~replace(., is.na(.), 0)) %>%
    mutate (n = n.x + n.y) %>%
    select(-c(n.x, n.y))

  return(table_clean)
}
```

DYB Table 4 - Vital statistics summary and life expectancy at birth: 2014 - 2018

Live Births, Infant Deaths, and Deaths (General) by Countries

```
# Create Function to Transform Table 4: Number of Years Available for
#Live Births, Infant Deaths, and Deaths (General) by Countries
```

```
t4_transform <- function(table, years, table_id){

# Prep original
table_clean <-
  table[- 1, ]   %>%
  select (c("...1", contains("number"))) %>%
  rename_all(~c("countries_areas",
                "live_births",
                "deaths",
                "infant_deaths")) %>%
  mutate(year=countries_areas) %>%
  replace_with_na(replace = list(countries_areas = years)) %>%
  separate(countries_areas, c("countries_areas"), " - ") %>%
  fill(countries_areas) %>%
  replace_with_na_all(condition = ~.x == "...") %>%
  replace_with_na_all(condition = ~.x == "-")

# Create count dataframe
table_count <-
  table_clean %>%
  gather(key="data_available", value="value", -c(countries_areas, year)) %>%
  mutate (countries_areas = gsub('[0-9]+', '', countries_areas)) %>%
  filter(value != is.na(.)) %>%
  group_by(countries_areas, data_available) %>%
  summarise(n=n())

return(table_count)
}
```

```
# Load Table 4s
table4_DYB_all_2 <- read_excel("table04.xlsx", skip = 4) #2014-2018
table4_DYB_all_1 <- read_excel("Table04 (1).xlsx", skip = 4) #2009-2013
```

```
# Transform Table 4s
table4_1 <- t4_transform(table4_DYB_all_1,
                         c("2009", "2010", "2011", "2012", "2013"))
table4_2 <- t4_transform(table4_DYB_all_2,
                         c("2014", "2015", "2016", "2017", "2018"))
```

```
# Merge Table 4s :2009-2018
table4_clean <- join_transform(table4_1, table4_2)
```

DYB Table 22 - Marriages and crude marriage rates, by urban/rural residence: 2014 - 2018

DYB Table 24 - Divorces and crude divorce rates by urban/rural residence: 2014 - 2018

```
# Create Function to Transform Table 22-25: Number of Years Available for
# Marriage and Divorces by Countries
ur_transform <- function(data_all, data_type) {

data <-
```

```
  data_all [-1,]   %>%
  select (c("...1", contains("20"))) %>%
  select (1:6) %>%
  rename_at(.vars = 1, ~c("countries_areas")) %>%
  separate(countries_areas, c("countries_areas"), " - ") %>%
  filter (!(countries_areas %in% c("Urban", "Rural"))) %>%
  replace_with_na(replace = list(countries_areas = "Total")) %>%
  fill(countries_areas) %>%
  replace_with_na_all(condition = ~.x == "...") %>%
  mutate (countries_areas = gsub('[0-9]+', '', countries_areas))

data_final <-
  data %>%
  gather(key="year", value=value, -c(countries_areas)) %>%
  filter(value != is.na(.)) %>%
  mutate(data_available = data_type) %>%
  group_by(countries_areas, data_available) %>%
  summarise(n=n())

return(data_final)

}
```

```
#Load Tables 22-25
table22_DYB <- read_excel("table22.xlsx", skip = 4) #2014-2018
table23_DYB <- read_excel("table23_2013.xlsx", skip = 4) #2009-2013
table24_DYB <- read_excel("table24.xlsx", skip = 4) #2014-2018
table25_DYB <- read_excel("Table25.xlsx", skip = 4) #2009-2013
```

```
#Transform Tables 22-25
table_m2 <- ur_transform(table22_DYB, "marriages")
table_m1 <- ur_transform(table23_DYB, "marriages")
table_d2 <- ur_transform(table24_DYB, "divorces")
table_d1 <- ur_transform(table25_DYB, "divorces")
```

```
table_mall <- join_transform(table_m1, table_m2)
table_dall <- join_transform(table_d1, table_d2)
```

Foetal Deaths by Countries

```
# Load Table 12s
table12_DYB_all_2 <- read_excel("./Gen Death/table12_2.xlsx", skip = 4) #2014-2018
table12_DYB_all_1 <- read_excel("./Gen Death/table12_1.xlsx", skip = 4) #2009-2013
```

```
# Transform Table 4s
table12_1 <- ur_transform(table12_DYB_all_1, "foetal_deaths")
table12_2 <- ur_transform(table12_DYB_all_2, "foetal_deaths")
```

```
# Merge Table 4s :2009-2018
table12_clean <- join_transform(table12_1, table12_2)
```

Merge lookup with DYB data

```
# Load UNPA Countries + Regions Lookup
lookup_table_all <-
  as_tibble(read_excel("UNFPA_countries.xlsx"))

all_tables_dyb <- rbind(table_mall, table_dall, table4_clean, table12_clean)
```

Create lookup consisting all Indicators (5 per country) and Scores

```
data_available <- c("live_births", "infant_deaths", "marriages", "divorces", "deaths", "foetal_deaths")

lookup_score<- function(avail, data) {

# Compute completeness Score by Category
complete_score <-
  data %>%
    group_by(countries_areas) %>%
    summarise(complete = n())

# Compute completeness Score by Overall Sum by Country
count_score <-
  aggregate(data$n, by = list(data$countries_areas), sum)
count_score <-
  rename(count_score, "countries_areas" = Group.1)

# Join lookup and indicators
available_count_lookup <- left_join(merge(avail, lookup_table_all),
                                    complete_score, by="countries_areas")
available_count_lookup <-
  rename(available_count_lookup, "data_available" = x)

available_count_lookup <- left_join(available_count_lookup, count_score, by = "countries_areas")

}
```

```
data_available_lookup <- lookup_score(data_available, all_tables_dyb)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#Create a UNFPA Life Course Approach Complete Data
UNFPA_lifeapp <-
  left_join(data_available_lookup, all_tables_dyb,
            by = c("countries_areas", "data_available"), ignore_case = T)
```

```r
UNFPA_lifeapp <-
  UNFPA_lifeapp %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  # Create score :0.75 on completeness across categories + 0.25 completeness in years
  mutate(score = ((complete/5)*0.85) # scored by completeness across categories
         + ((x/50)*0.15)) %>% # scored by completeness across years
  mutate (countries_areas = reorder(countries_areas, score))

head(data_available_lookup, 5)
```

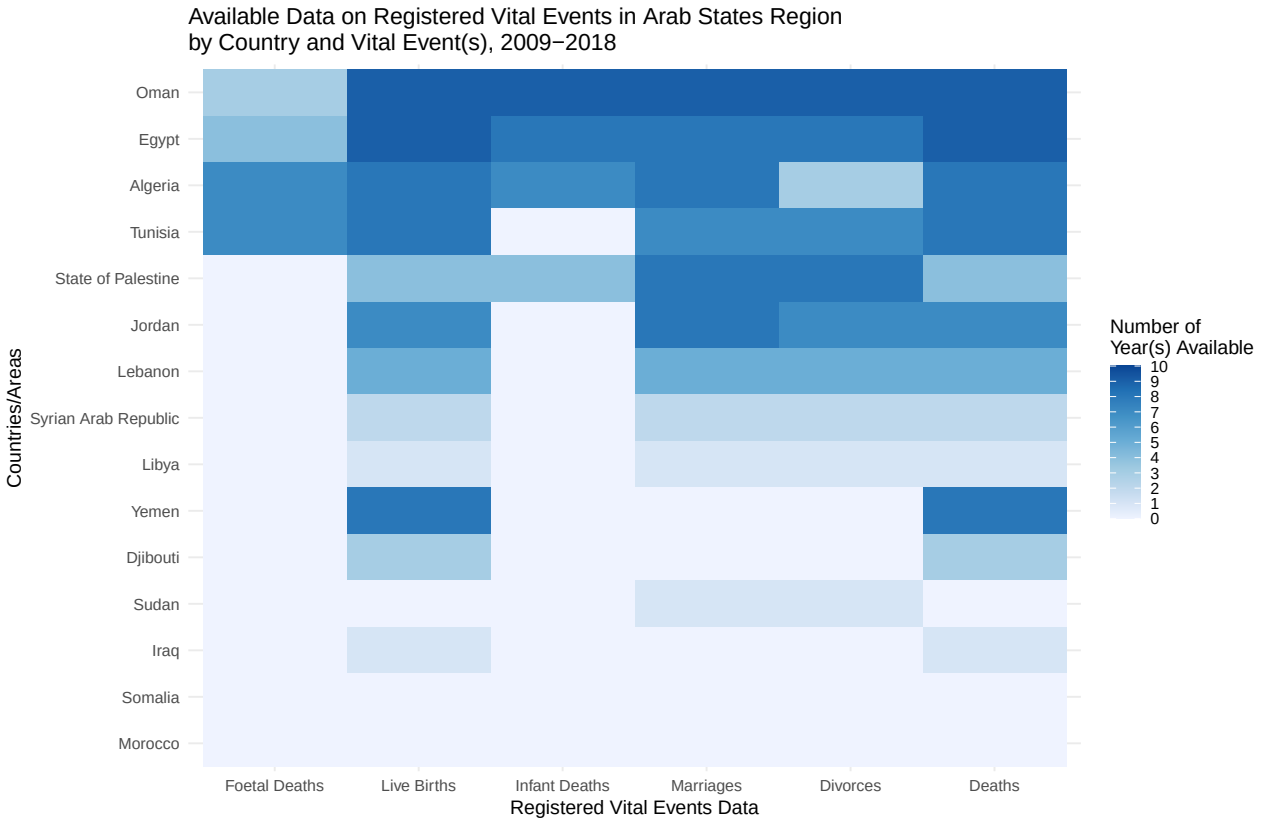
```
##   data_available countries_areas UNFPA_Regions complete x
## 1    live_births          Angola           ESA        3 3
## 2  infant_deaths          Angola           ESA        3 3
## 3      marriages          Angola           ESA        3 3
## 4       divorces          Angola           ESA        3 3
## 5         deaths          Angola           ESA        3 3
```

Generate Summary Completeness Heatmaps by Region

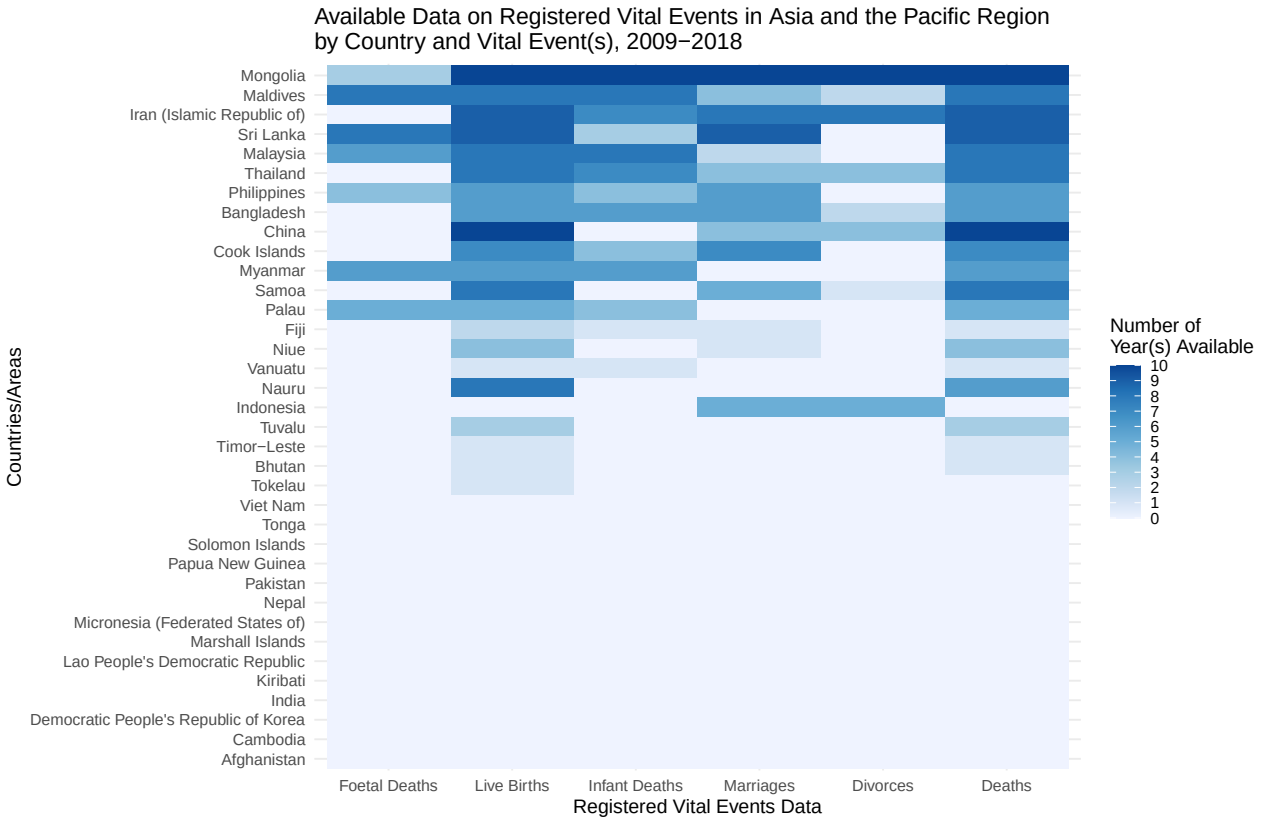
```r
# Set parameters for levels and labels
levels_pref <- c("foetal_deaths", "live_births", "infant_deaths", "marriages", "divorces", "deaths")
labels_pref <- c("Foetal Deaths", "Live Births", "Infant Deaths", "Marriages", "Divorces", "Deaths")
```

```r
# Create function to produce Heatmaps
num_years_plot <- function(region)
  ggplot(subset(UNFPA_lifeapp, UNFPA_Regions %in% region),
         aes(x = factor(data_available,
                        levels = levels_pref), y = countries_areas, fill = n)) +
  geom_tile() +
  scale_fill_distiller(name = "Number of\nYear(s) Available", palette = "Blues",
                       direction = +1, breaks = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
                       limits=c(0,10)) +
  theme_minimal() +
  labs(title = paste("Available Data on Registered Vital Events in",
                     ifelse(region == "AP", "Asia and the Pacific",
                            ifelse(region == "AS", "Arab States",
                                   ifelse(region == "EECA", "Eastern Europe and Central Asia",
                                          ifelse(region == "ESA", "Eastern and Southern Africa",
                                                 ifelse(region == "LAC", "Latin America and Caribbean
                                                        ifelse(region == "WCA", "Western and Central
                     "\nby Country and Vital Event(s), 2009-2018",
                     sep = " "),
       x = "Registered Vital Events Data", y = "Countries/Areas") +
  scale_x_discrete(labels = labels_pref)
```

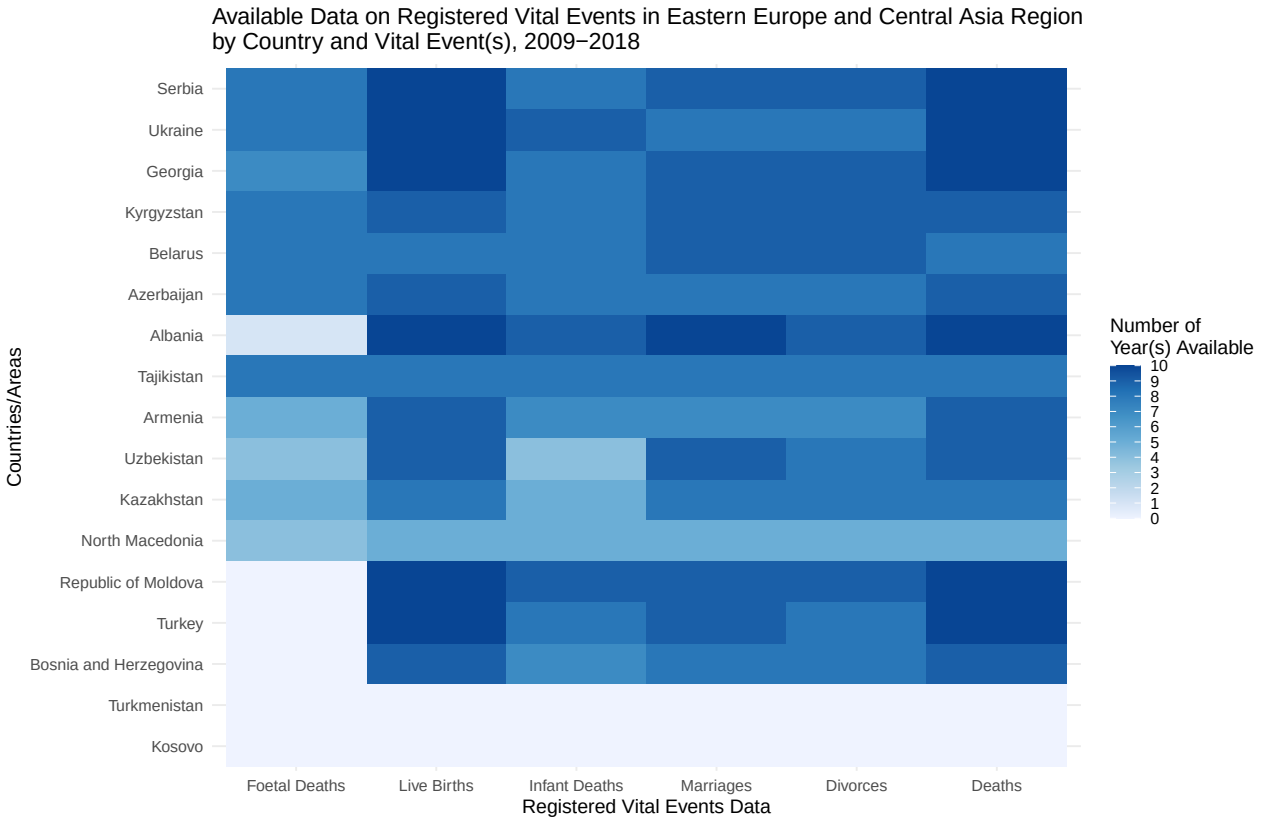
```r
# Run heatmaps
num_years_plot("AS")
```



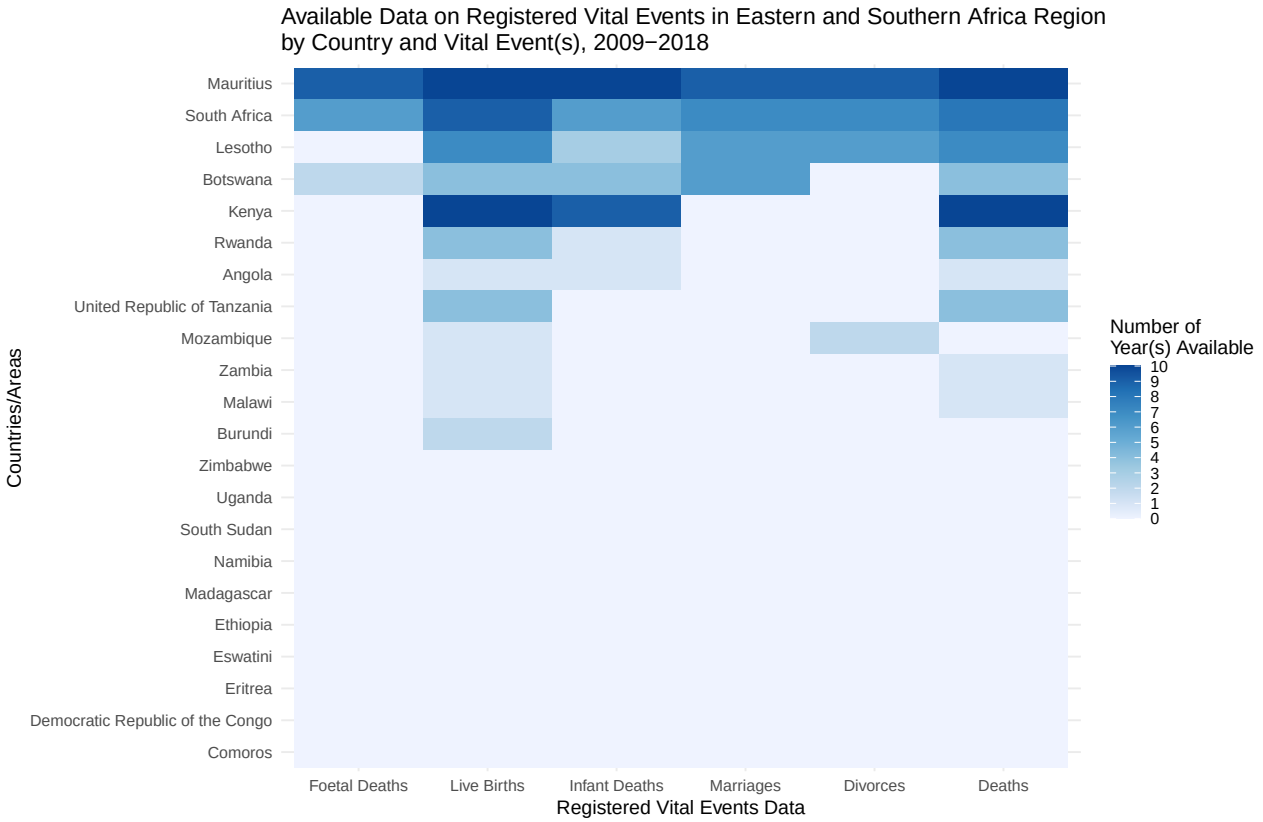

```
num_years_plot("AP")
```



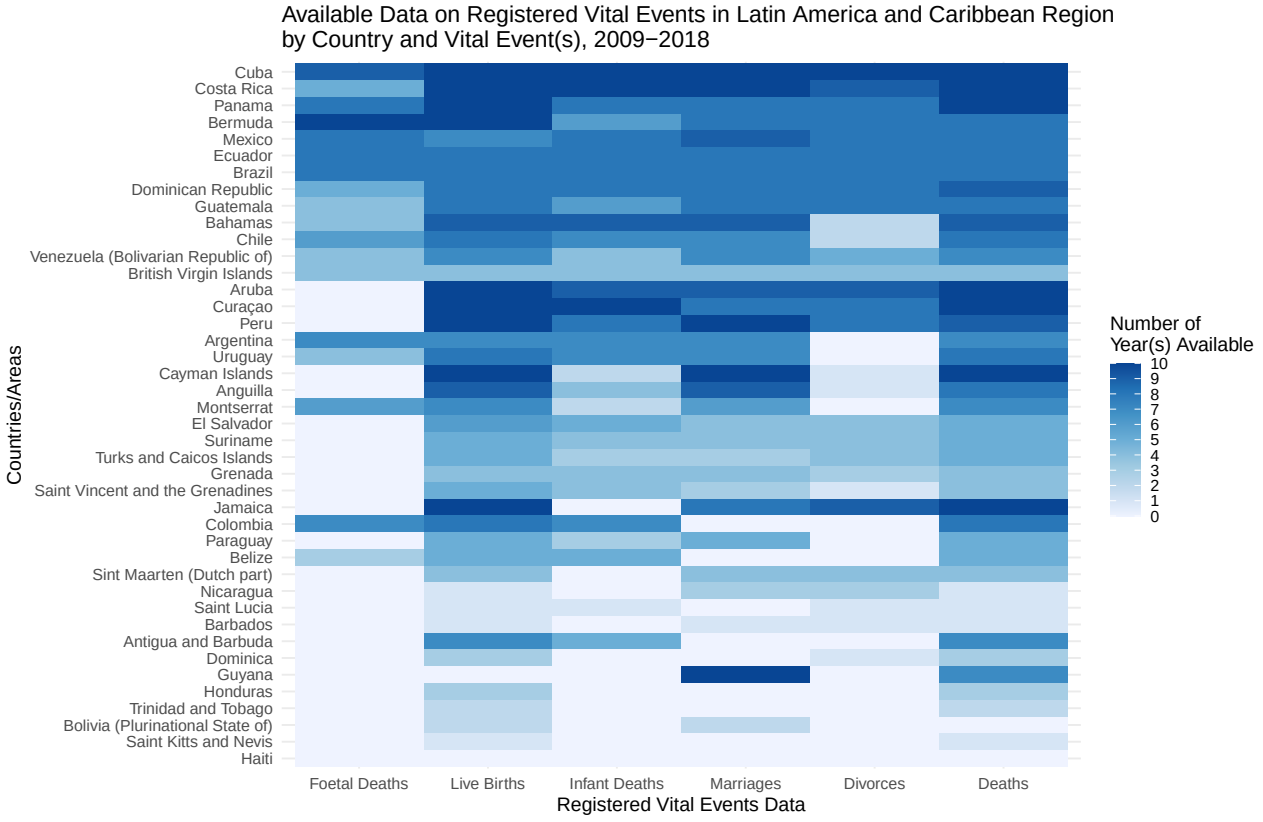
```
num_years_plot("EECA")
```



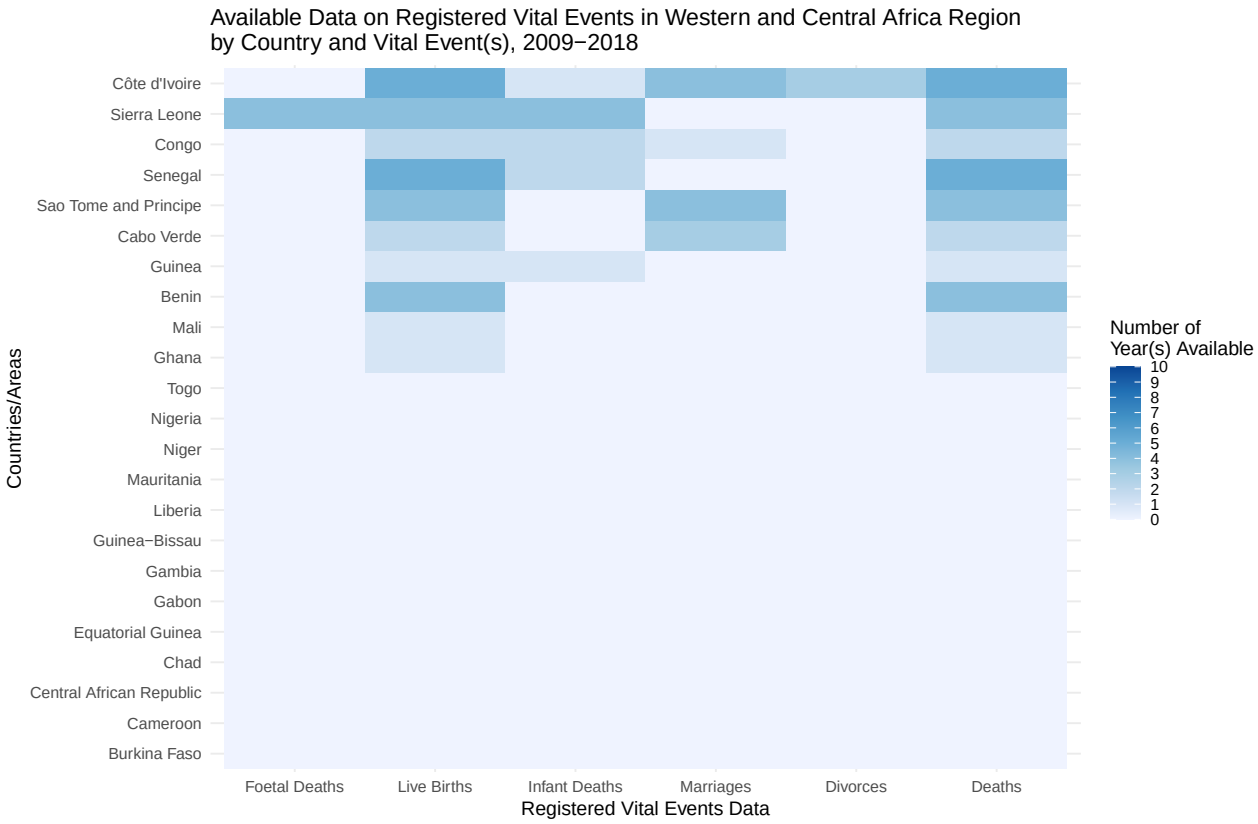

```
num_years_plot("ESA")
```




```
num_years_plot("LAC")
```



```
num_years_plot("WCA")
```



Summary Heatmaps: Data Availability Year from 2009 to 2018 (Purple Heatmaps)

This visualization will be remodelled with the following prompts: - Reordered by countries (most complete to least, top-bottom) - Facets removed

Prepare Data

```
# Create table 4 transformation function that includes years and recodes values to binary
# (1 = "Available", 0 = "Unavailable)
t4_year_transform <- function(t4_table, years)
  t4_table[- 1, ]   %>%
  select (c("...1", contains("number"))) %>%
  rename_all(~c("countries_areas", "live_births", "deaths", "infant_deaths")) %>%
  mutate(year=countries_areas) %>%
  replace_with_na(replace = list(countries_areas = years)) %>%
  separate(countries_areas, c("countries_areas"), " - ") %>%
  fill(countries_areas) %>%
  replace_with_na_all(condition = ~.x == "...") %>%
  replace_with_na_all(condition = ~.x == "-")   %>%
  filter(!(live_births %in% NA & deaths %in% NA & infant_deaths %in% NA)) %>%
```

```
  mutate_at(vars(live_births, deaths, infant_deaths), ~replace(., is.na(.), 0)) %>%
  mutate(live_births = ifelse(live_births > 0, 1, 0)) %>%
  mutate(deaths = ifelse(deaths > 0, 1, 0)) %>%
  mutate(infant_deaths = ifelse(infant_deaths > 0, 1, 0)) %>%
  gather(-c(countries_areas, year), key = "data_available", value = "availability") %>%
  mutate (countries_areas = gsub('[0-9]+', '', countries_areas))
```

```
# Transform table 4s
table4_1_year <-
  t4_year_transform(table4_DYB_all_1,
                    c("2009", "2010", "2011", "2012", "2013"))
table4_2_year <-
  t4_year_transform(table4_DYB_all_2,
                    c("2014", "2015", "2016", "2017", "2018"))
```

```
# Create table 22-25 transformation function that includes years and recodes values to binary
# (1 = "Available", 0 = "Unavailable)
ur_year_transform <- function(data_ur, names, data_type){
  data_ur[- 1, ]   %>%
  select (c("...1", contains("20"))) %>%
  select (1:6) %>%
  rename_all(~names) %>%
  separate(countries_areas, c("countries_areas"), " - ") %>%
  filter (!(countries_areas %in% c("Urban", "Rural"))) %>%
  replace_with_na(replace = list(countries_areas = "Total")) %>%
  fill(countries_areas) %>%
  replace_with_na_all(condition = ~.x == "...") %>%
  replace_with_na_all(condition = ~.x == "-") %>%
  mutate (countries_areas = gsub('[0-9]+', '', countries_areas)) %>%
  gather(-c(countries_areas), key = "year", value = "availability") %>%
  mutate_at(vars(availability), ~replace(., is.na(.), 0)) %>%
  mutate(availability = ifelse(availability > 0, 1, 0)) %>%
  mutate(data_available = data_type) %>%
  select(countries_areas, year, data_available,availability)
  }
```

```
# Transform 22-25, Marriages and Divorces
table_m2_year <-
  ur_year_transform(table22_DYB,
                    c("countries_areas", "2014", "2015", "2016", "2017", "2018"), "marriages")
table_m1_year <-
  ur_year_transform(table23_DYB,
                    c("countries_areas", "2009", "2010", "2011", "2012", "2013"), "marriages")
table_d2_year <-
  ur_year_transform(table24_DYB,
                    c("countries_areas", "2014", "2015", "2016", "2017", "2018"), "divorces")
table_d1_year <-
  ur_year_transform(table25_DYB,
                    c("countries_areas", "2009", "2010", "2011", "2012", "2013"), "divorces")
```

```
# Transform 12, Foetal Deaths
table_f2_year <-
  ur_year_transform(table12_DYB_all_2,
```

```
                    c("countries_areas", "2014", "2015", "2016", "2017", "2018"), "foetal_deaths")
table_f1_year <-
  ur_year_transform(table12_DYB_all_1,
                    c("countries_areas", "2009", "2010", "2011", "2012", "2013"), "foetal_deaths")
```

```
# Bind all Data
all_tables_year <- rbind(table4_1_year, table4_2_year, table_d1_year,
                         table_d2_year, table_m1_year, table_m2_year,
                         table_f2_year, table_f1_year)
```

Create lookup consisting all Indicators (5 per country) and Year

```
# Create Lookup  :Country, Region, Available Data, Year
data_year_lookup <-
  data_available_lookup %>% # Reuse previous lookup that already includes Available Data
  select(data_available:UNFPA_Regions) %>%
  merge(., c("2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017", "2018")) %>%
  # Merge available numbers of years possible
  rename("year" = y)
```

Generate Data

```
# Merge lookup
table_year_unfpa <-
  data_year_lookup %>%
  left_join(., all_tables_year, by=c("countries_areas", "data_available", "year")) %>%
  mutate_at(vars(availability), ~replace(., is.na(.), 0)) %>%
  mutate(UNFPA_Regions = reorder(UNFPA_Regions, desc(availability)))
```

Generate Summary Yearly Availability Heatmaps by Data Category

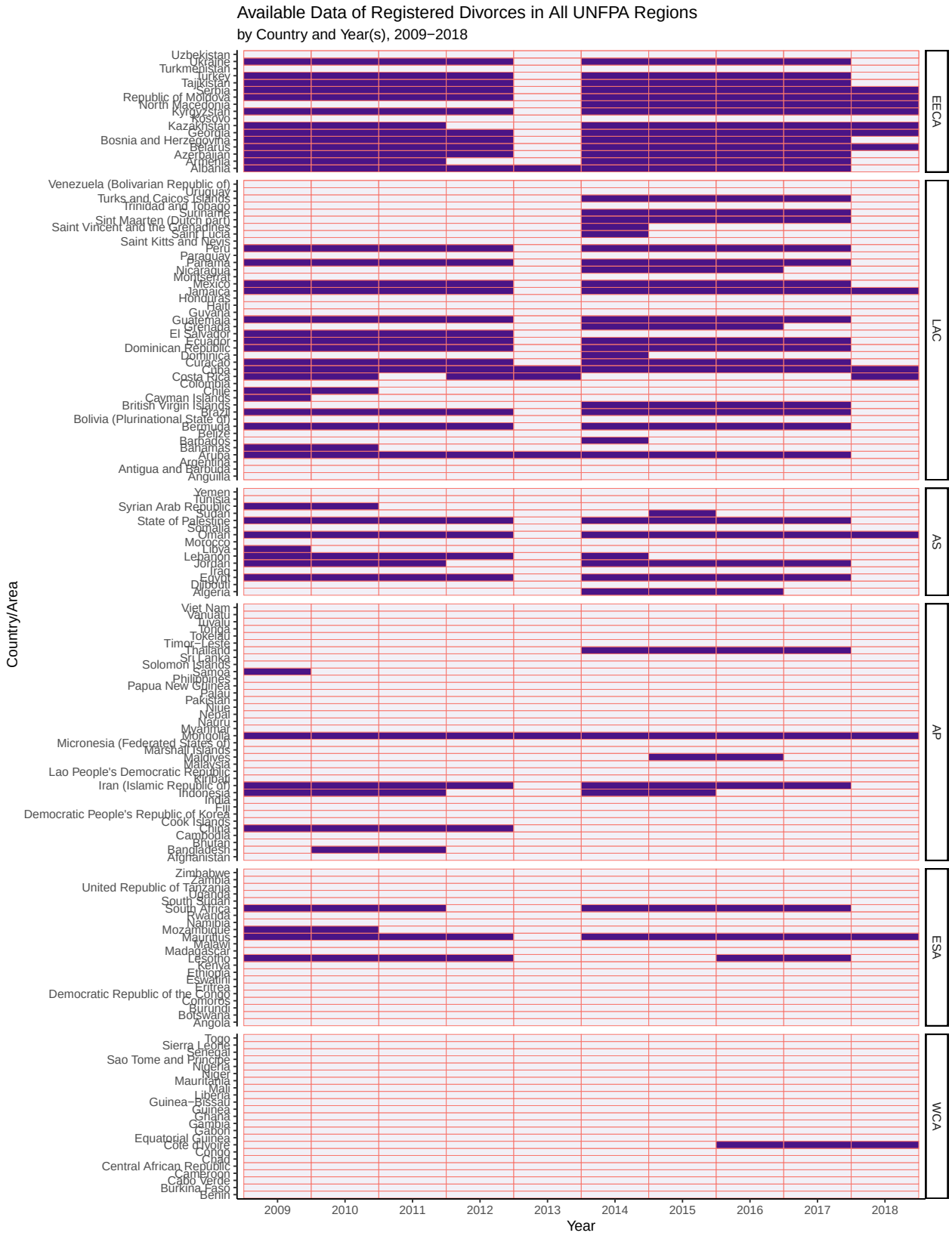
```
all_plot_sum <- function(data_type, data_type_name) {
ggplot(subset(table_year_unfpa, data_available %in% data_type), aes(x = year, y = countries_areas, fill=
  geom_tile() +
  facet_grid(UNFPA_Regions~., space = "free_y", scale = "free_y") +
  scale_fill_distiller(palette = "Purples", direction = +1, breaks = c(0, 1), limits=c(0,1)) +
  theme(axis.text.x = element_text(size =0.5)) +
  theme_classic()+
  theme(legend.position = "none")+
  labs(title = paste("Available Data of Registered",
                     data_type_name, "in All UNFPA Regions"),
       subtitle= "by Country and Year(s), 2009-2018",
       x = "Year", y = "Country/Area")
}
```

```
all_plot_sum("live_births", "Live Births")
```

```
all_plot_sum("infant_deaths", "Infant Deaths")
```

```
all_plot_sum("marriages", "Marriages")
```

```
all_plot_sum("divorces", "Divorces")
```



```
all_plot_sum("deaths", "Deaths")
```

```
all_plot_sum("foetal_deaths", "Foetal Deaths")
```

Detailed Heatmaps: Numbers of Years Available by Types of Registration from 2009 to 2018 (Green Heatmaps)

```
# Create Function to Generate Detailed (Green) Heatmaps

det_heatmap <- function(data, region, title, labels, levels) {

ggplot(subset(data, UNFPA_Regions %in% region), aes(x = factor(data_available, levels = levels), y = cou
  geom_tile() +
  scale_fill_distiller(name = "Number of \n Year(s) Available", palette = "Greens",
                       direction = +1, breaks = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
                       limits=c(0,10)) +
  theme_minimal() +
  labs(x = paste(title, "Data", sep = " "), y = "Countries/Areas") +
  labs(title = paste("Available Data on Registered",
                     title,
                     "in the",
                     ifelse(region == "AP", "Asia and the Pacific",
                              ifelse(region == "AS", "Arab States",
                                     ifelse(region == "EECA", "Eastern Europe and Central Asia",
                                            ifelse(region == "ESA", "Eastern and Southern Africa",
                                                   ifelse(region == "LAC", "Latin America and Caribbean
                                                          ifelse(region == "WCA", "Western and Central
                     "Region",
                      "\nby Country and Disaggregation Variable(s), 2009-2018",
                     sep = " "),
       x = paste(title, "Data"), y = "Countries/Areas") +
  theme(axis.text.x=element_text(angle=45,hjust=1)) +
  scale_x_discrete(labels = labels)
}
```

Live Births

```
# Table Live Births

all_bt <- list.files(path = "./Birth Regs",
                       pattern = "table",
                       full.names = TRUE)

all_bts_excel <-
  lapply(all_bt, function (i) {
  x = read_excel(i, skip = 4)
})
```

```
# Create Freq Table Function for non-Urban/Rural DYB Tables
ap_transform <- function(data, data_type) {
  data[- 1, ] %>%
  select (1:2) %>%
  rename_all(~c("countries_areas", "total")) %>%
  mutate(countries_areas = gsub('[0-9]+', '', countries_areas)) %>%
```

```
  filter (!(countries_areas %in% c(" - ", " +", " (C)", " (+U)", " (|)", " (U)",
                                   " (+C)", "* (C)", "* (+C)", "* (+U)",
                                   "Unknown - Inconnu", "* (U)"))) %>%
  replace_with_na(replace = list(countries_areas = "Total")) %>%
  fill(countries_areas) %>%
  filter(!(total == is.na(.))) %>%
  separate(countries_areas, c("countries_areas"), " - ") %>%
  group_by(countries_areas) %>%
  summarise(n=n()) %>%
  mutate(data_available = data_type)
}
```

```
# Transform all live births data
# Will abbreviate this to lapply(data[1:2], ur_transform) and lapply(data[3:6], ap_transform)
bt_ur_clean1 <- ur_transform(all_bts_excel[[1]], "births_urbanrural")
bt_ur_clean2 <- ur_transform(all_bts_excel[[2]], "births_urbanrural")
bt_am_clean1 <- ap_transform(all_bts_excel[[3]], "births_age_mother")
bt_am_clean2 <- ap_transform(all_bts_excel[[4]], "births_age_mother")
bt_af_clean1 <- ap_transform(all_bts_excel[[5]], "births_age_father")
bt_af_clean2 <- ap_transform(all_bts_excel[[6]], "births_age_father")
bt_ur_all <- join_transform(bt_ur_clean1, bt_ur_clean2)
bt_am_all <- join_transform(bt_am_clean1, bt_am_clean2)
bt_af_all <- join_transform(bt_af_clean1, bt_af_clean2)
```

```
# Create Freq Table Function for Supplementary Tables
st_clean <- function(data) {
    data %>%
    clean_names() %>%
    group_by(country_or_area, year) %>%
    summarise(n=n()) %>%
    select(country_or_area, year) %>%
    rename("countries_areas" = country_or_area) %>%
    group_by(countries_areas) %>%
    summarise(n=n())
  }
```

```
# Load Supplementary Tables
all_lb <- list.files(path = "./Birth Regs", # insert path to folder
                        pattern = "LB", # do not change this
                        full.names = TRUE)

all_lbs_csv <-
  lapply(all_lb, function (i) {
  x = read_csv(i)
})
```

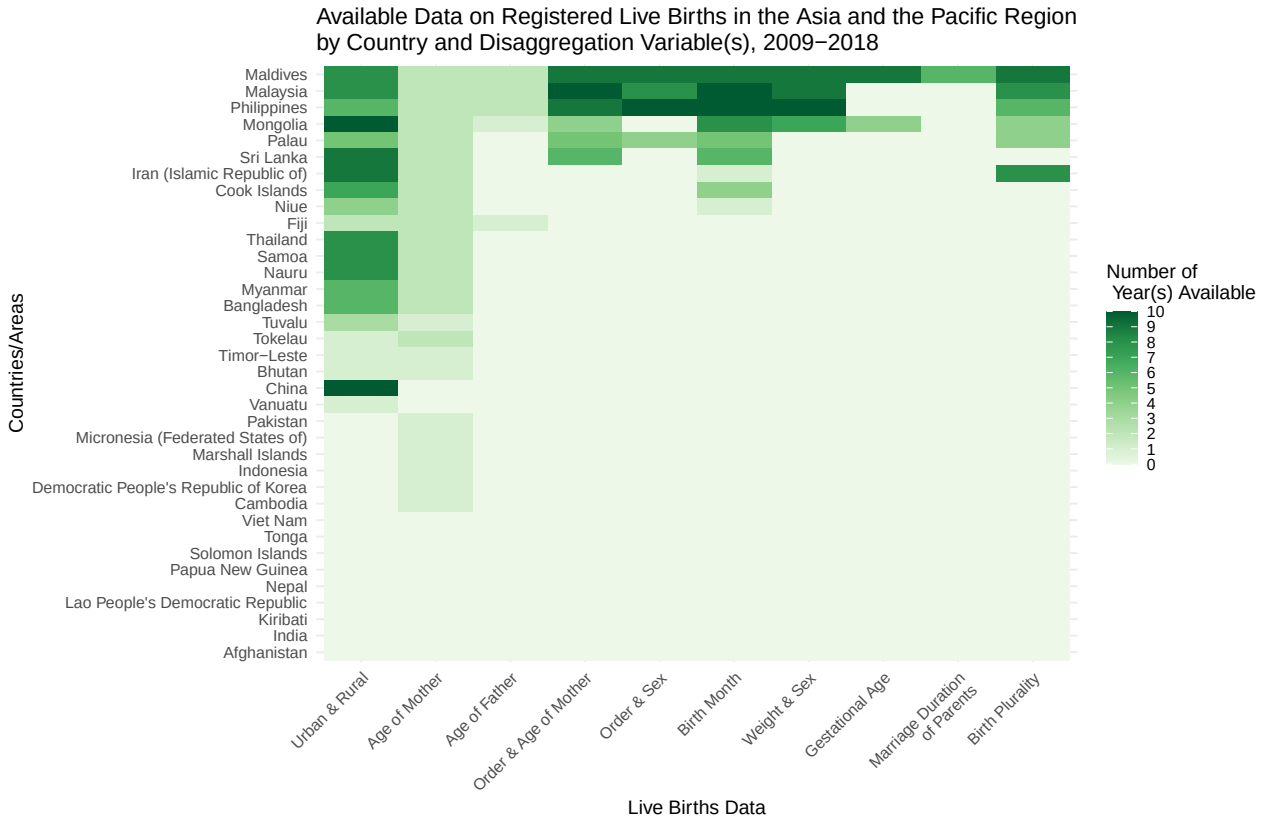
```
# Create Frequency Tables for All STs

all_births_long <-
  lapply(all_lbs_csv, function (i) {
  x = st_clean(i)
})
```

```
all_births_long <-
  Map(cbind, all_births_long, data_available =
        list("birth_order", "birth_bw_sex","birth_gestation",
             "birth_month", "birth_plural", "birth_ord_sex",
             "birth_mar_len"))

all_births_long <- rbindlist(all_births_long)
all_births_long <- select(all_births_long, c(countries_areas, data_available, n))
all_births_tables <- bind_rows(all_births_long, bt_ur_all, bt_af_all, bt_am_all)
```

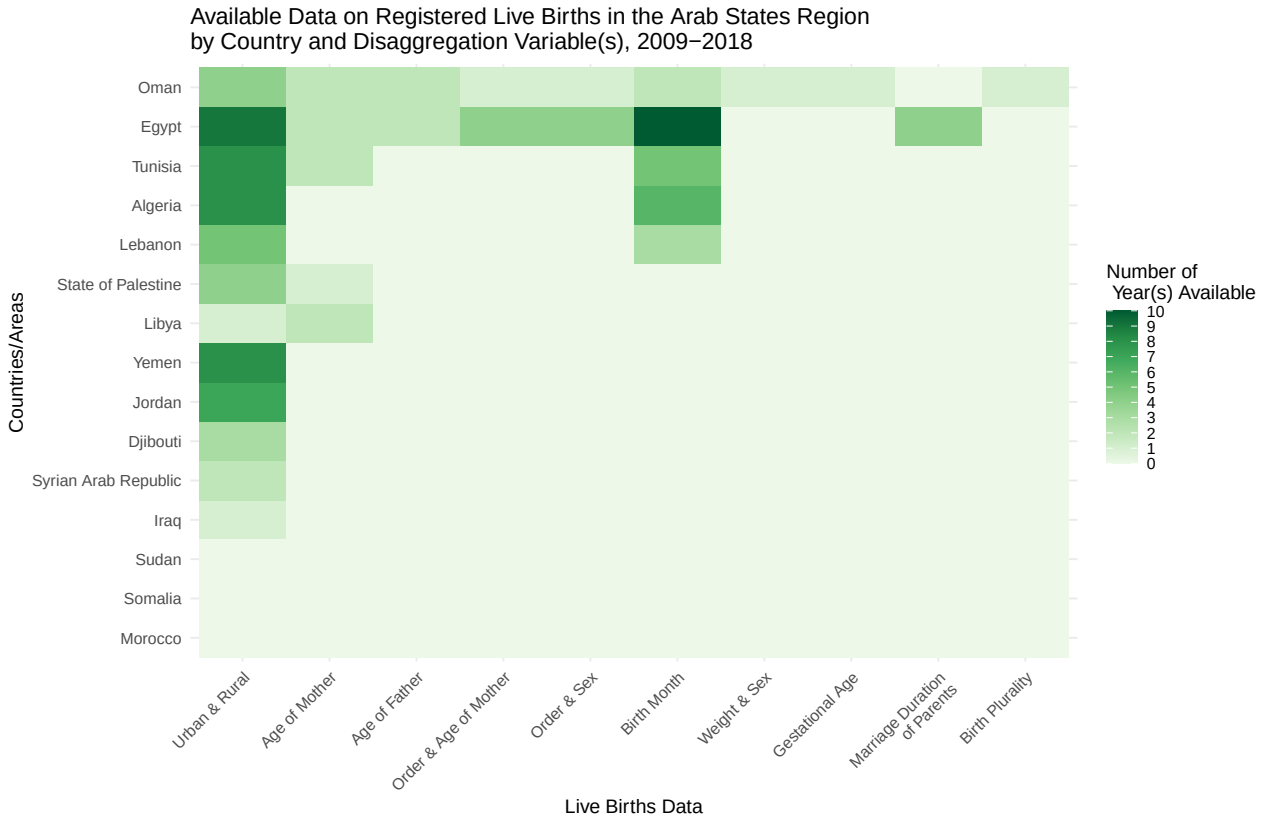
```
# Create Live Births Lookup
types_birth_long <- c("birth_order", "birth_bw_sex","birth_gestation",  "birth_month",
                      "birth_plural", "birth_ord_sex", "birth_mar_len", "births_urbanrural",
                      "births_age_mother", "births_age_father")
lookup_birth_unfpa <- lookup_score(types_birth_long, all_births_tables)
```

```
# Create UNFPA-specific disaggregated Live Births data
all_births_unfpa <- left_join(lookup_birth_unfpa, all_births_tables, by = c("countries_areas", "data_ava

births_unfpa_clean <-
  all_births_unfpa %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate(score = ((complete/5)*0.85)+((x/50)*0.15)) %>%
  mutate (countries_areas = reorder(countries_areas, score))
```

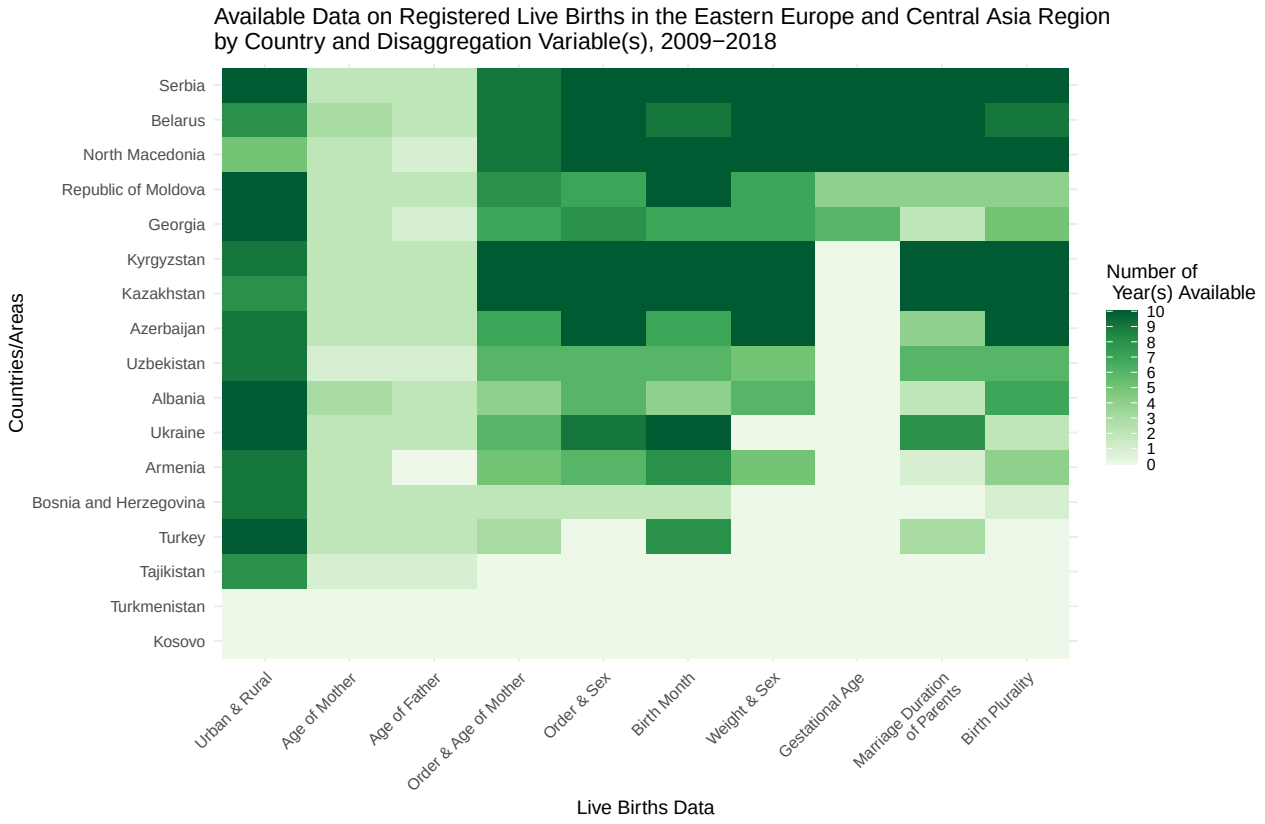
```
# Generate Live Birth Heatmaps

birth_labels <- c("Urban & Rural", "Age of Mother", "Age of Father", "Order & Age of Mother",  "Order &
birth_prefs <- c("births_urbanrural", "births_age_mother", "births_age_father", "birth_order", "birth_o

det_heatmap(births_unfpa_clean, "AP", "Live Births", birth_labels, birth_prefs)
```



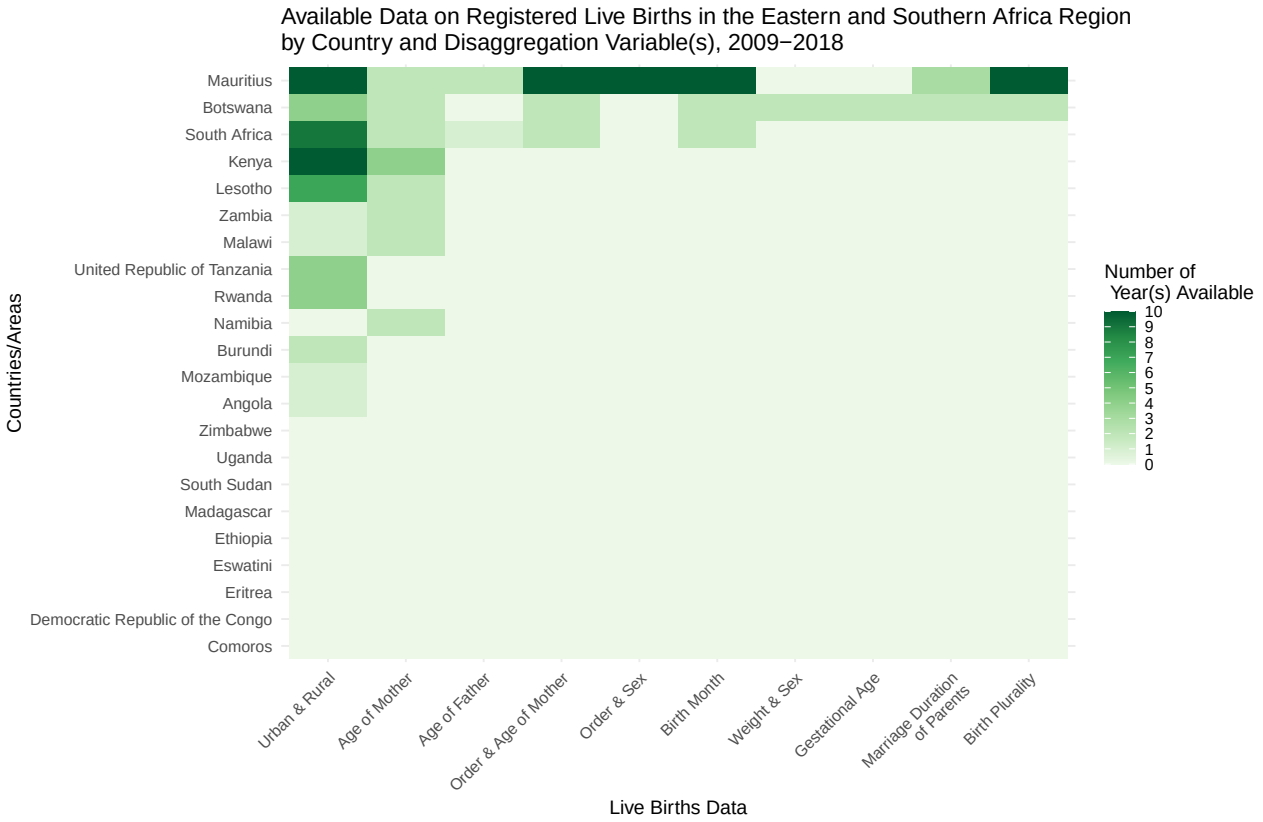

```
det_heatmap(births_unfpa_clean, "AS", "Live Births", birth_labels, birth_prefs)
```



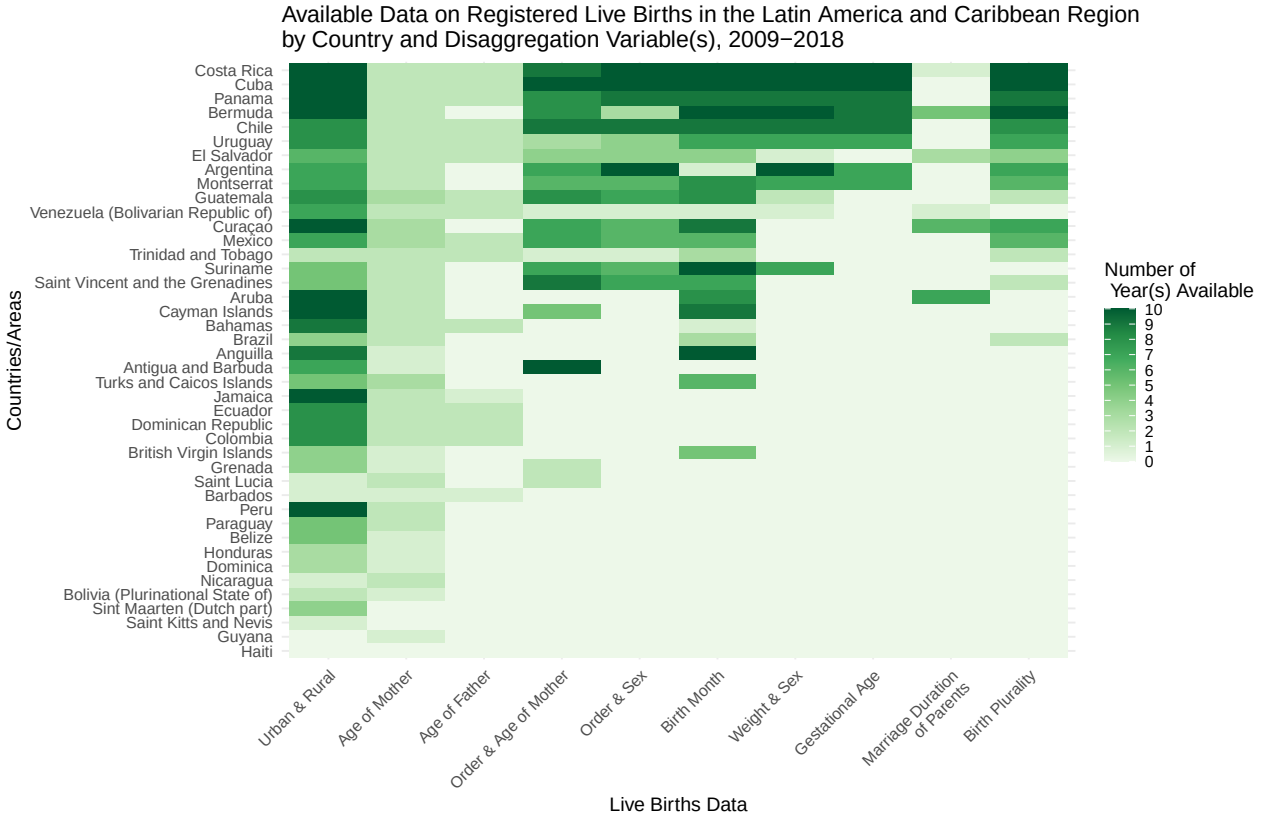
```
det_heatmap(births_unfpa_clean, "EECA", "Live Births", birth_labels, birth_prefs)
```



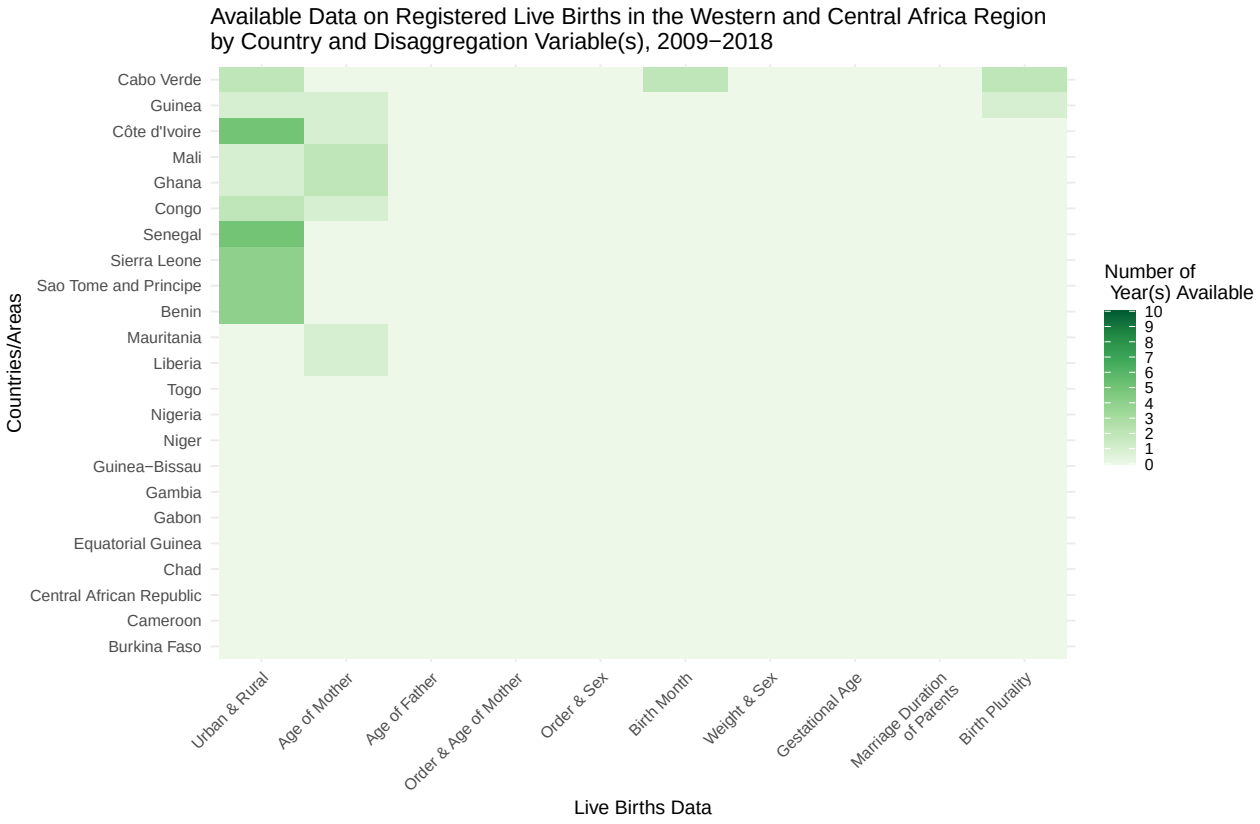

```
det_heatmap(births_unfpa_clean, "ESA", "Live Births", birth_labels, birth_prefs)
```




```
det_heatmap(births_unfpa_clean, "LAC", "Live Births", birth_labels, birth_prefs)
```



```
det_heatmap(births_unfpa_clean, "WCA", "Live Births",birth_labels, birth_prefs)
```



Marriages and Divorces

```
# Load Marriages and Divorces Tables

all_mdt <- list.files(path = "./MarriageDiv",
                      pattern = "table",
                      full.names = TRUE)

all_mdt_excel <-
  lapply(all_mdt, function (i) {
  x = read_excel(i, skip = 4)
})

md_am_clean1 <- ap_transform(all_mdt_excel[[1]], "marriage_age")
md_am_clean2 <- ap_transform(all_mdt_excel[[2]], "marriage_age")
md_am_all <- join_transform(md_am_clean1, md_am_clean2)
```

```
# Load Supplementary Tables :Marriages and Divorces

all_mds <- list.files(path = "./MarriageDiv",
                      pattern = "MD",
                      full.names = TRUE)
```

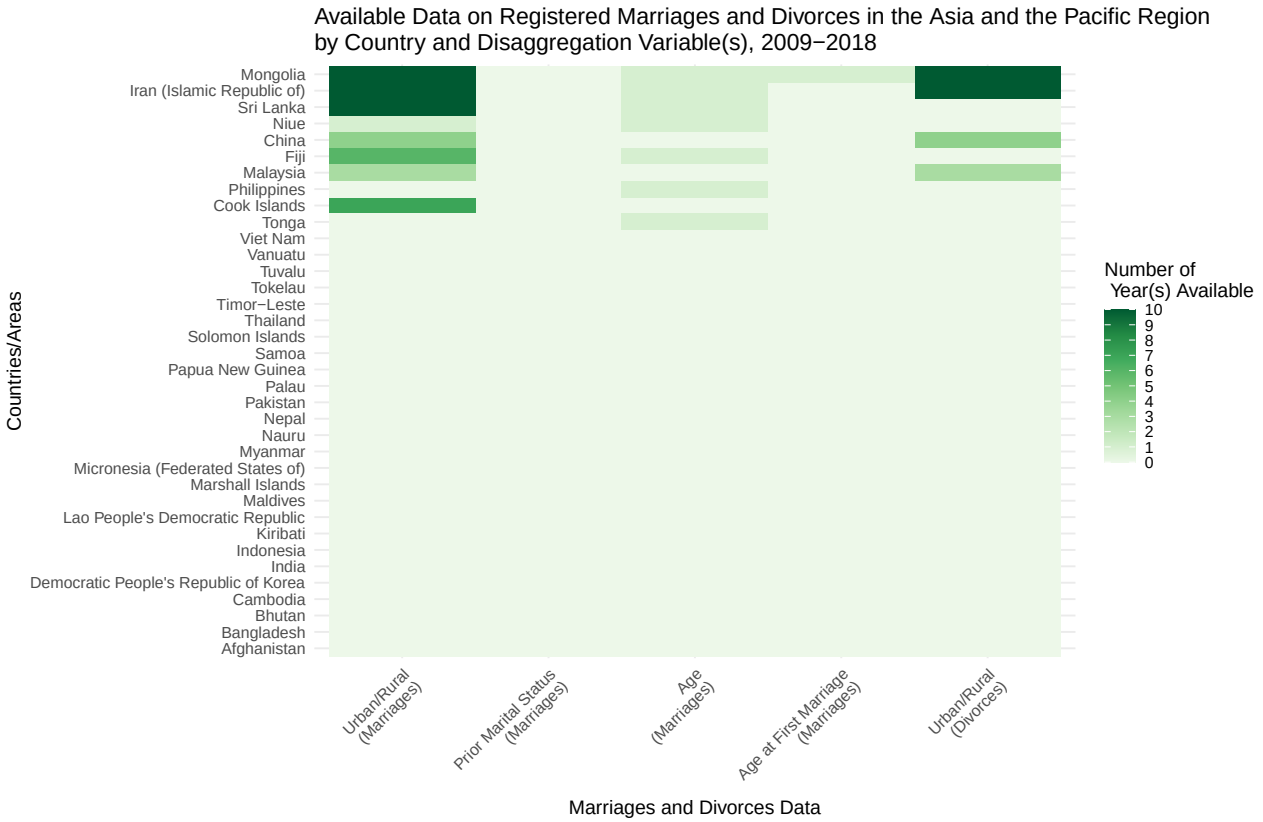
```r
all_mds_csv <-
  lapply(all_mds, function (i) {
  x = read_csv(i)
})
```

```r
# Transform all Marriages and Divorces STs into Frequencies

all_mardivs_long <-
  lapply(all_mds_csv, function (i) {
  x = st_clean(i)
})

all_mardivs_long <-
  Map(cbind, all_mardivs_long, data_available =
        list("marriage_cross", "divorce_ur", "marriage_1st_age", "marriage_ur"))

all_mardivs_long <- rbindlist(all_mardivs_long)
all_mardivs_long <- select(all_mardivs_long, c(countries_areas, data_available, n))
all_mardivs_tables <- bind_rows(all_mardivs_long, md_am_clean1, md_am_clean2)
```

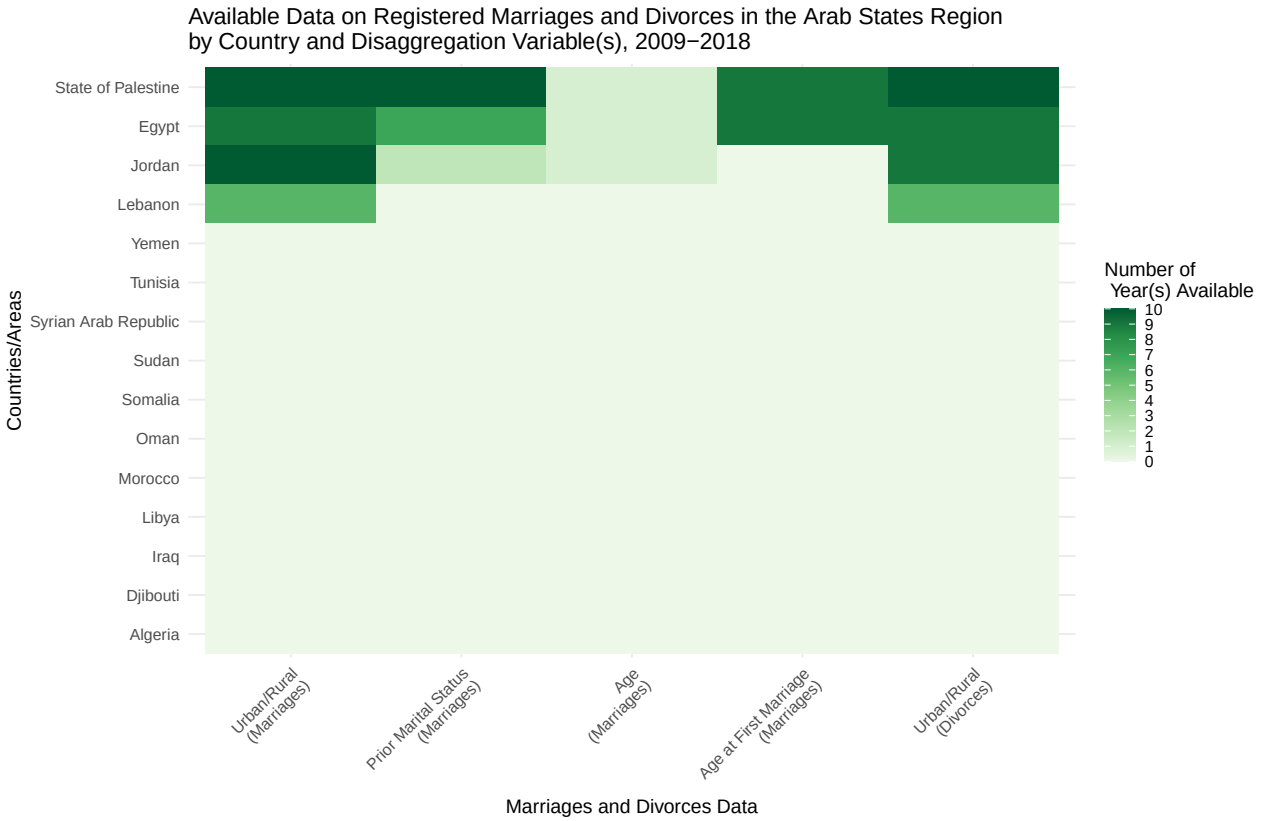
```r
# Create Marriage and Divorces Lookup
lookup_mardiv_unfpa <- lookup_score(c("marriage_cross", "divorce_ur", "marriage_1st_age", "marriage_ur"
```

```r
# Generate all UNFPA-specific marriage and divorce availability data
all_mardiv_unfpa <- left_join(lookup_mardiv_unfpa, all_mardivs_tables, by = c("countries_areas", "data_a

mardiv_unfpa_clean <-
  all_mardiv_unfpa %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate(score = ((complete/5)*0.85)+((x/50)*0.15)) %>%
  mutate (countries_areas = reorder(countries_areas, score))
```

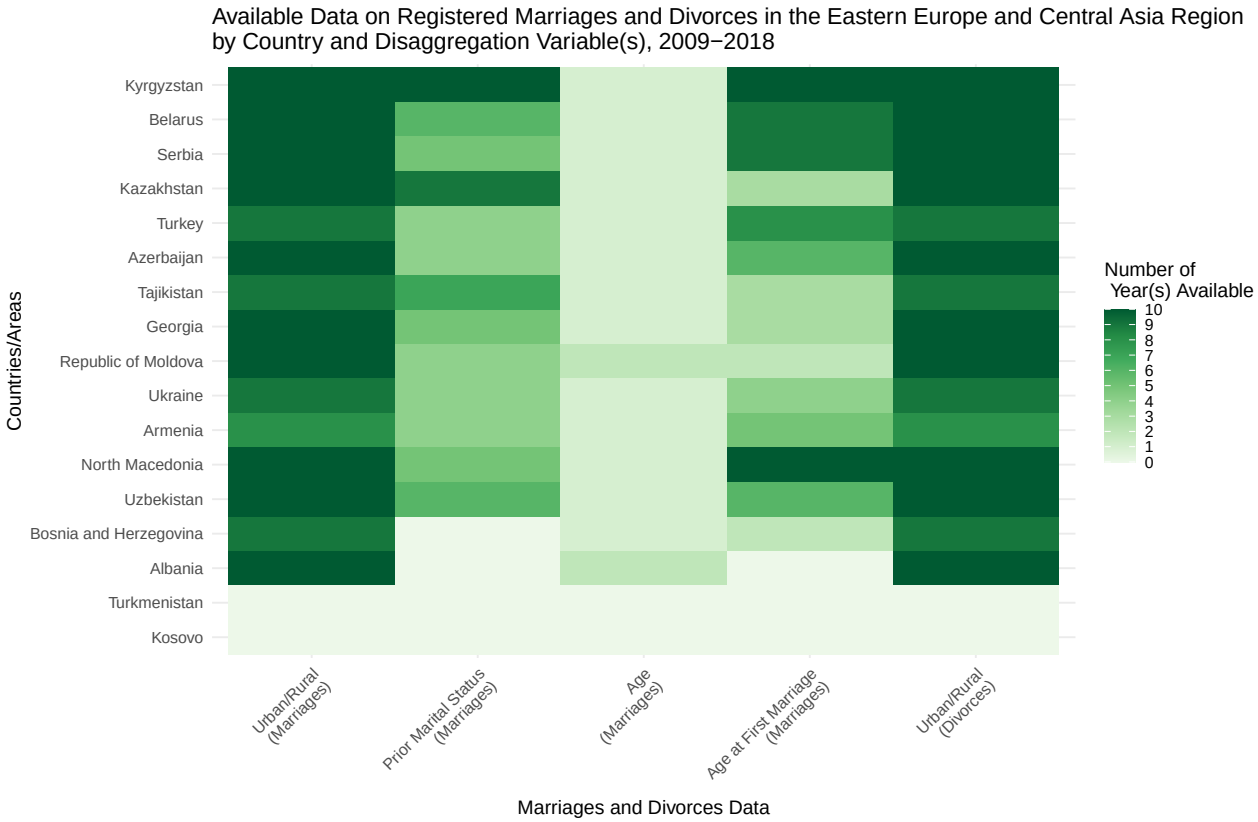
```r
# Generate Marriage/Divorces Heatmaps

mardiv_levels <- c("marriage_ur", "marriage_cross", "marriage_age", "marriage_1st_age", "divorce_ur")
mardiv_labels <- c("Urban/Rural\n(Marriages)", "Prior Marital Status\n(Marriages)", "Age\n(Marriages)",

det_heatmap(mardiv_unfpa_clean, "AP", "Marriages and Divorces", mardiv_labels, mardiv_levels)
```



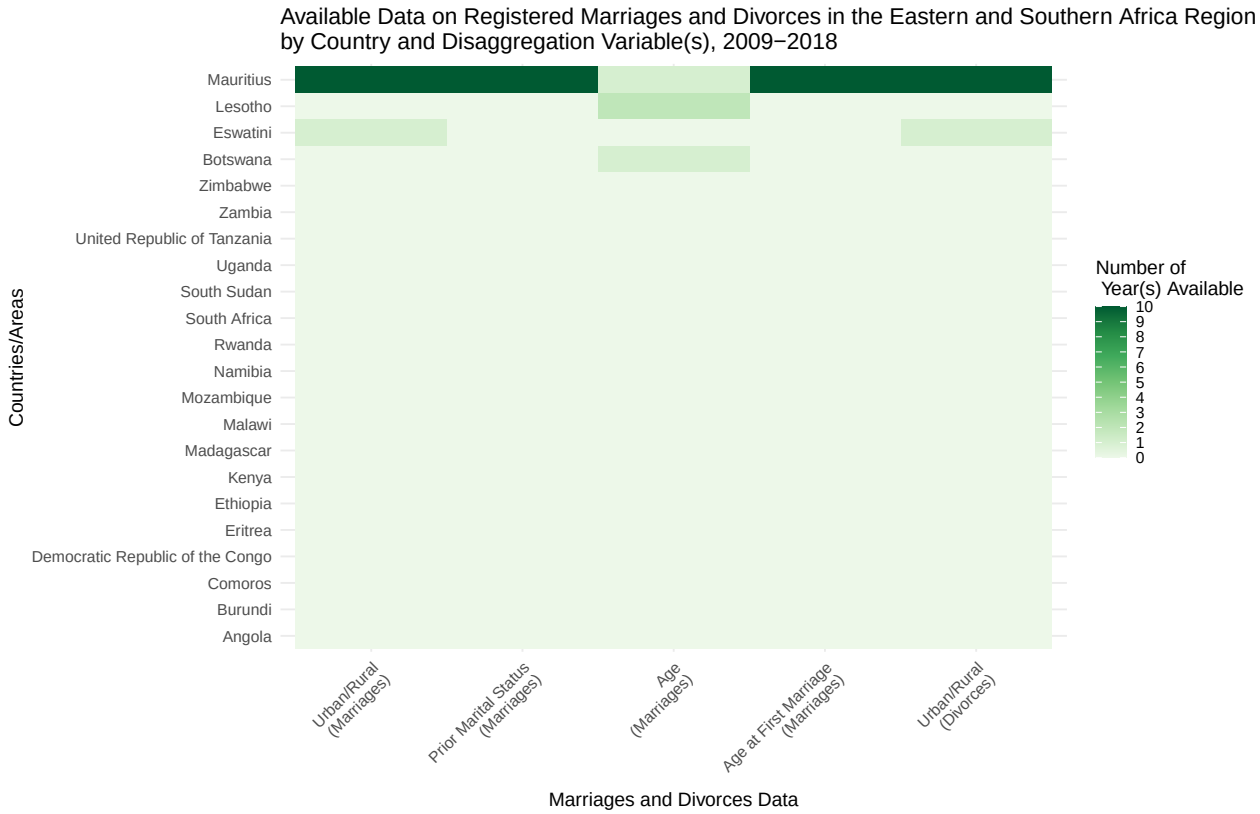

```
det_heatmap(mardiv_unfpa_clean, "AS", "Marriages and Divorces", mardiv_labels, mardiv_levels)
```



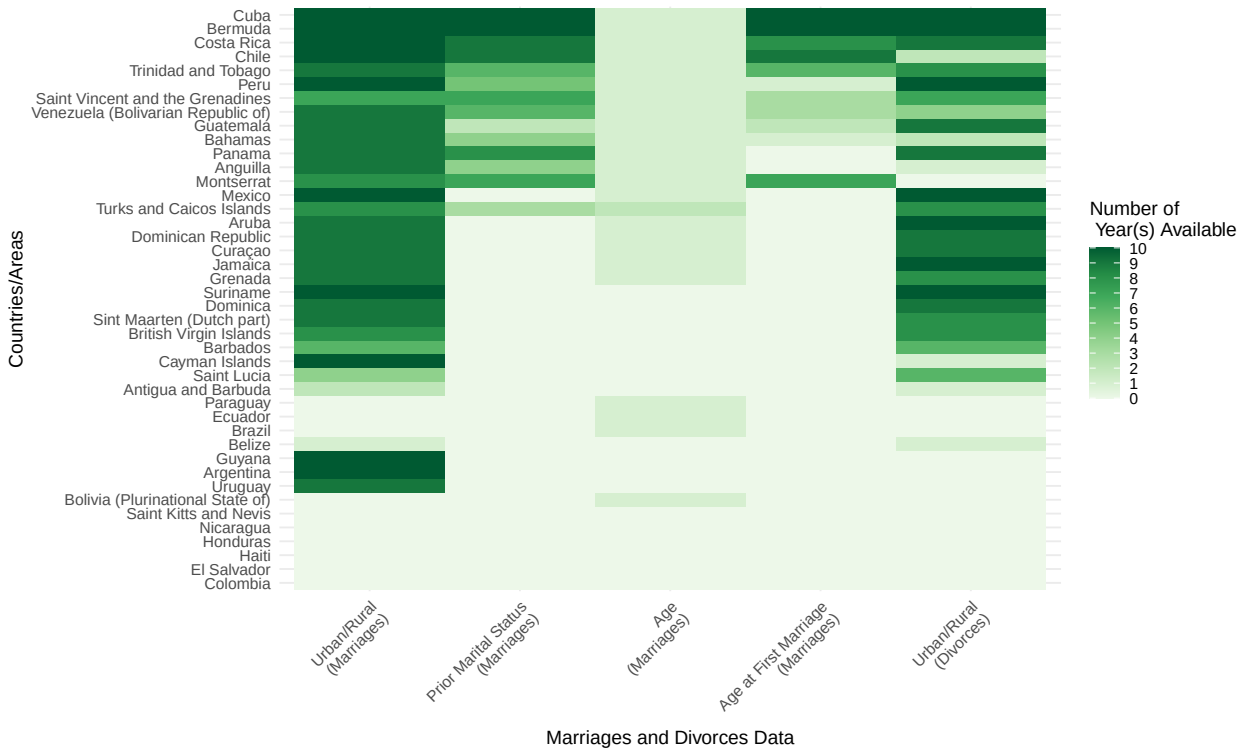
```
det_heatmap(mardiv_unfpa_clean, "EECA", "Marriages and Divorces", mardiv_labels, mardiv_levels)
```



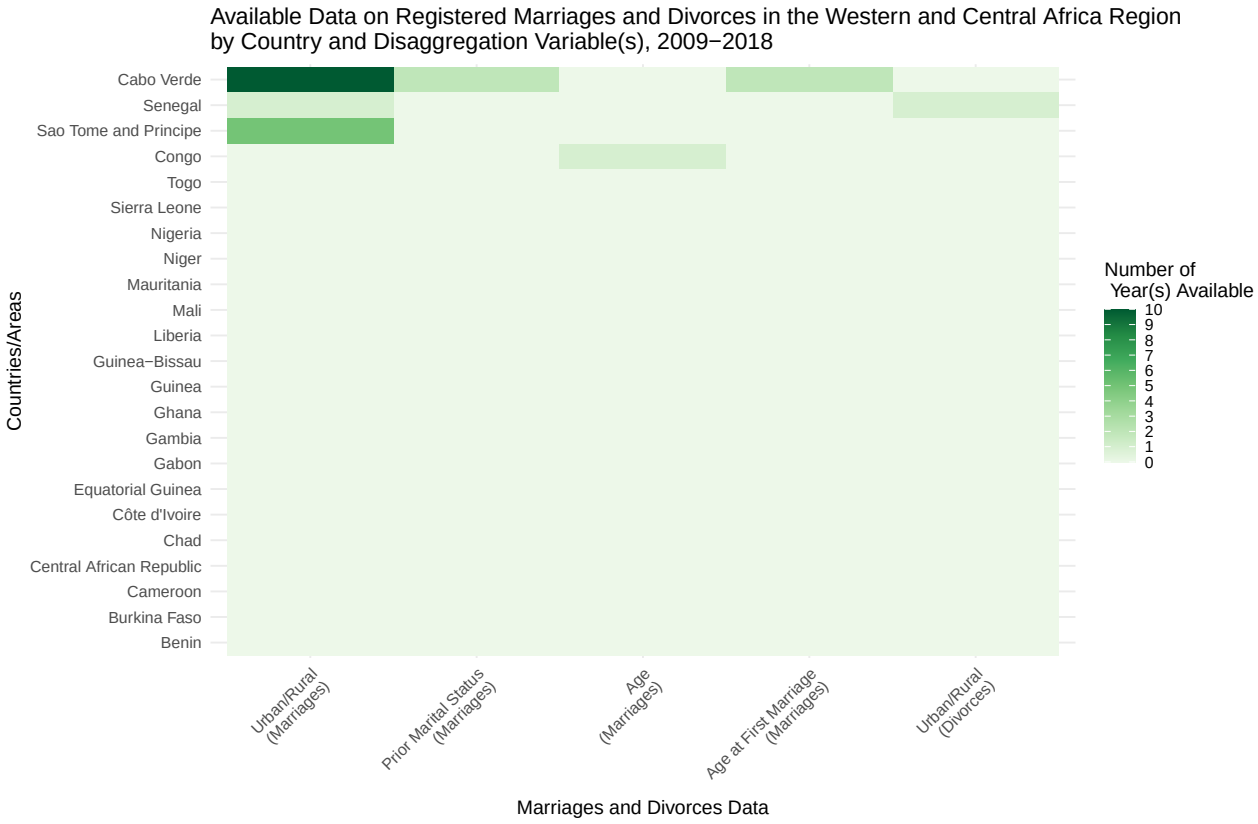

```
det_heatmap(mardiv_unfpa_clean, "ESA", "Marriages and Divorces", mardiv_labels, mardiv_levels)
```




```
det_heatmap(mardiv_unfpa_clean, "LAC", "Marriages and Divorces",mardiv_labels, mardiv_levels)
```



```
det_heatmap(mardiv_unfpa_clean, "WCA", "Marriages and Divorces", mardiv_labels, mardiv_levels)
```




```
# Supplementary Tables :General and Foetal Deaths

all_gds <- list.files(path = "./Gen Death",
                      pattern = "GD",
                      full.names = TRUE)

all_gds_csv <-
  lapply(all_gds, function (i) {
  x = read_csv(i)
})

all_fds <- list.files(path = "./Gen Death",
                      pattern = "FD",
                      full.names = TRUE)

all_fds_csv <-
  lapply(all_fds, function (i) {
  x = read_csv(i)
})

id_agesex<-read_excel("./Gen Death/IFagesex.xlsx")
id_urbanrural<-read_csv("./Gen Death/IFurbanrural.csv")
```

```r
all_gds_long <-
  lapply(all_gds_csv, function (i) {
  x = st_clean(i)
})

all_gds_long <-
  Map(cbind, all_gds_long, data_available =
        list("gm_cause", "gm_age_sex_ur", "gm_month", "gm_sex_ur"))

all_fds_long <-
  lapply(all_fds_csv, function (i) {
  x = st_clean(i)
})

all_fds_long <-
  Map(cbind, all_fds_long, data_available =
        list("ab_urbanrural", "fd_agewoman", "fd_gest_age", "fd_sex_ur"))

id_agesex_long<- ap_transform(id_agesex, "inf_death_age_sex")

id_urbanrural_long <-
  id_urbanrural %>%
  st_clean() %>%
  cbind(data_available = "inf_death_urbanrural")


all_gfd_tables <- bind_rows(rbindlist(all_gds_long), rbindlist(all_fds_long), all_fds_long, id_agesex_l
```

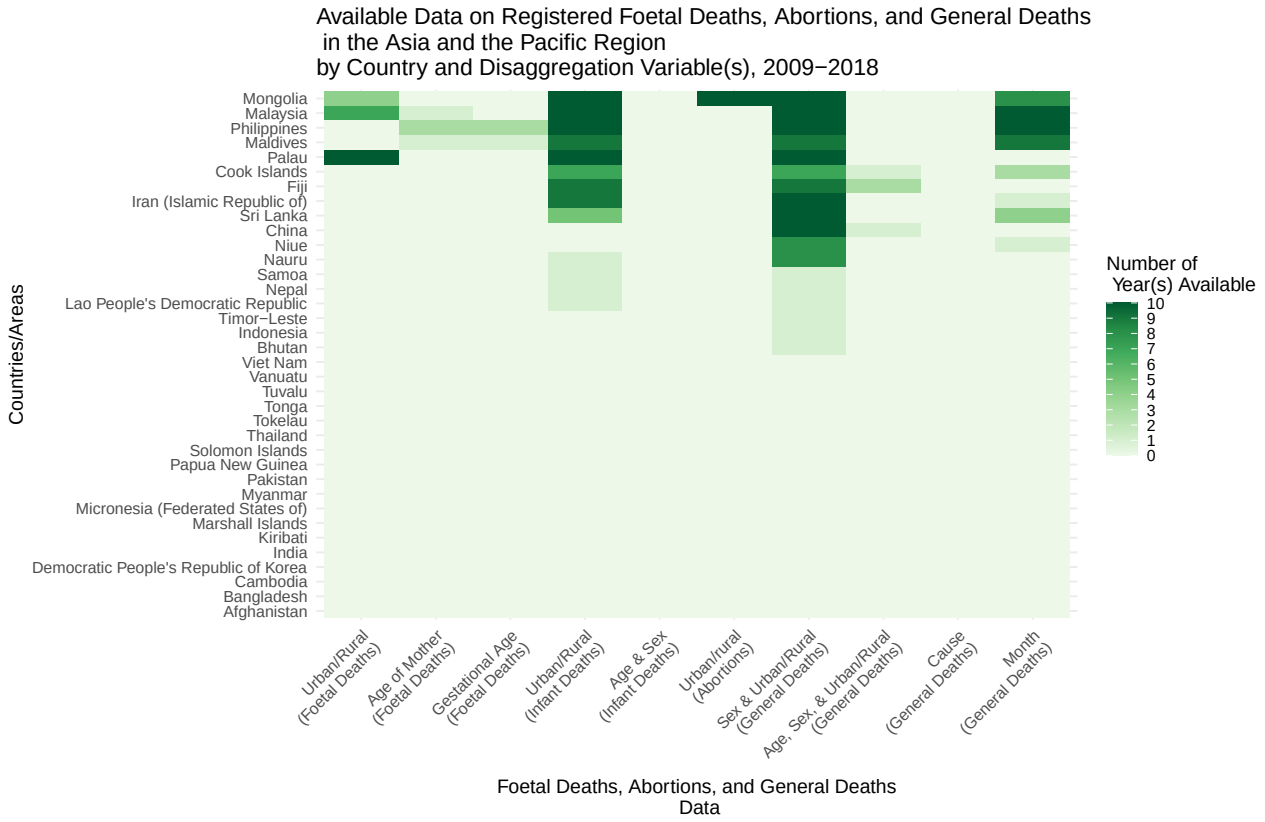
```r
# I just identified a problem in the wrangling of Infant Death tabs by age and sex. I will come back to
id_agesex_long<-
  id_agesex[- 1, ] %>%
  select (1:2) %>%
  rename_all(~c("countries_areas", "total")) %>%
  mutate(countries_areas = gsub('[0-9]+', '', countries_areas)) %>%
  separate(countries_areas, c("countries_areas"), " - ") %>%
  filter (!(countries_areas %in% c(" - ", "+", " (C)", "(+U)", "(|)", " (U)",
                                   " (+C)", "*(C)", "*(+C)", "*(+U)",
                                   "Unknown - Inconnu", "* (U)", "days", "months", "Less than  day")))
```

```r
lookup_gfd_unfpa <- lookup_score(c("inf_death_age_sex", "inf_death_urbanrural", "gm_cause", "gm_age_sex
```

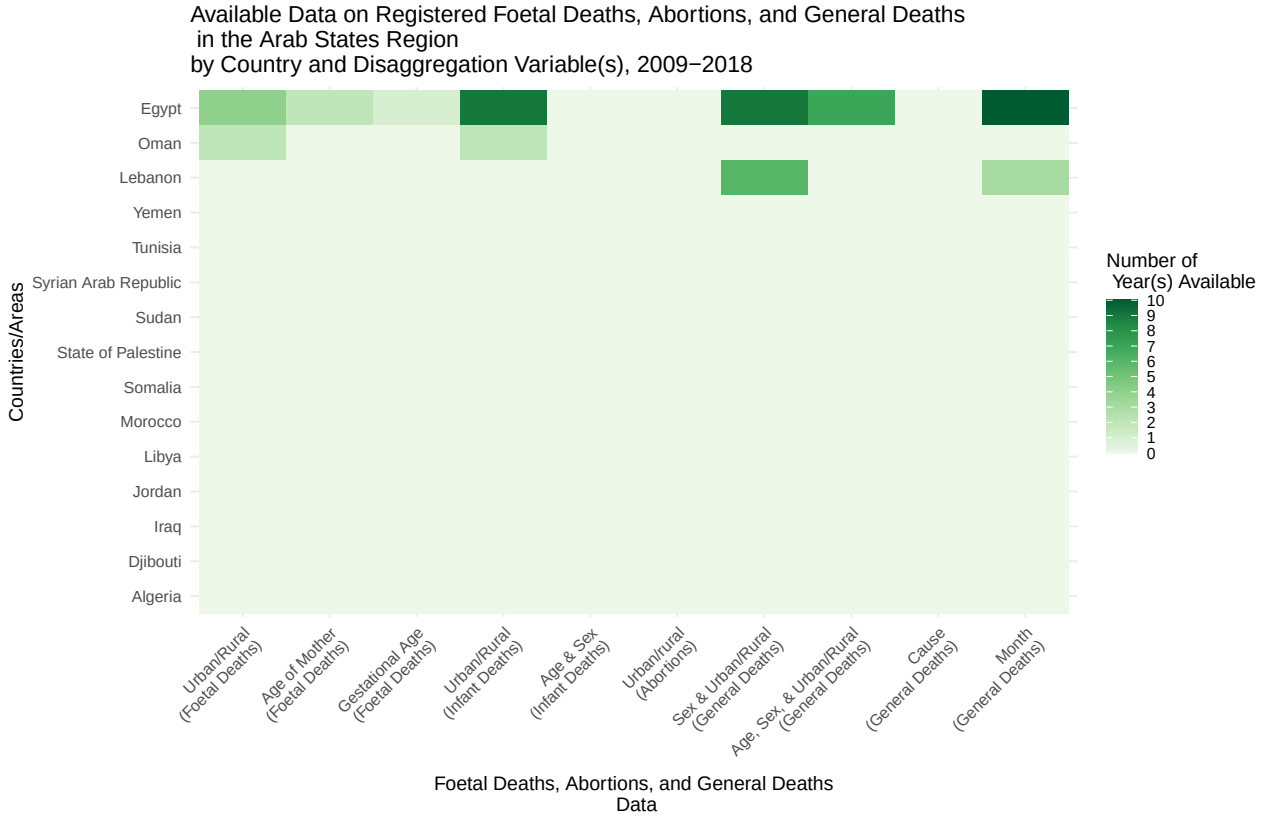
```r
all_gfd_unfpa <- left_join(lookup_gfd_unfpa, all_gfd_tables, by = c("countries_areas", "data_available"
```

```r
gfd_unfpa_clean <-
  all_gfd_unfpa %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate(score = ((complete/5)*0.85)+((x/50)*0.15)) %>%
  mutate (countries_areas = reorder(countries_areas, score))
```

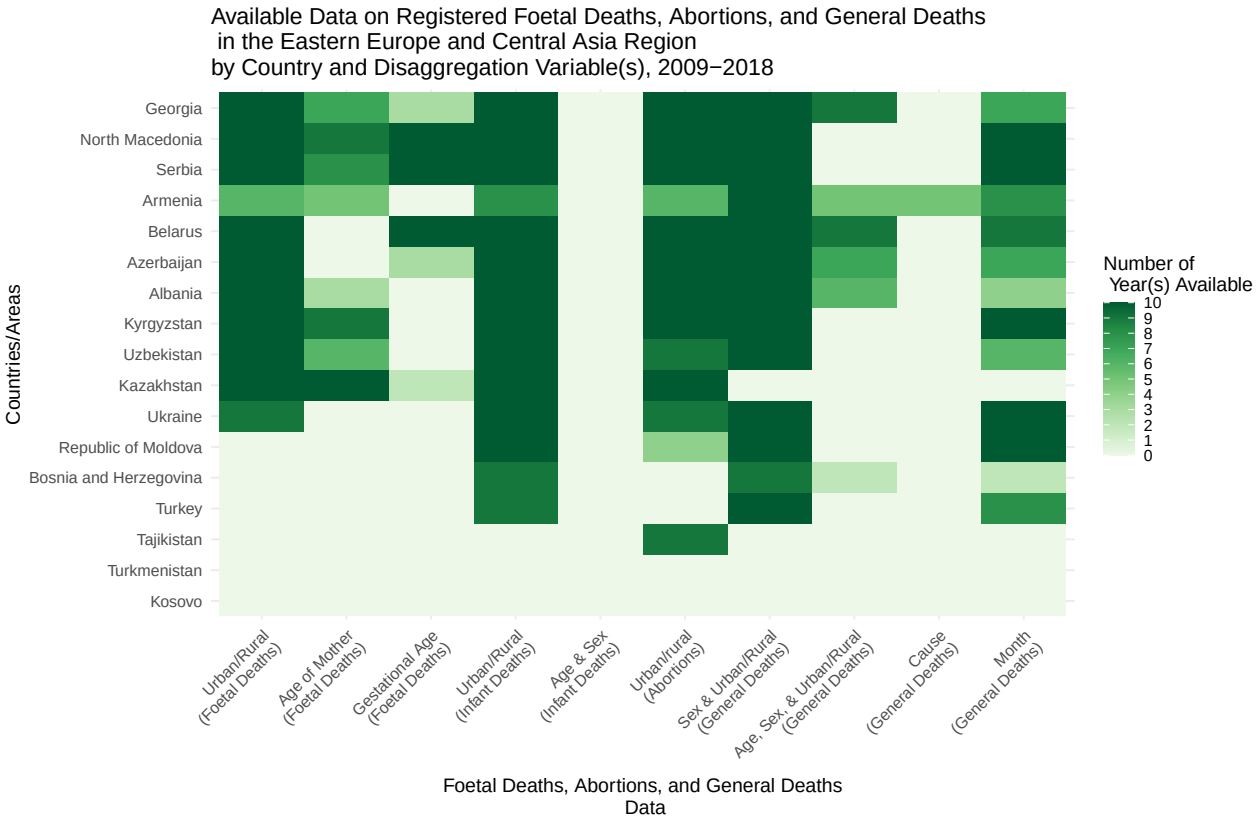
```r
GF_levels <- c("fd_sex_ur", "fd_agewoman", "fd_gest_age", "inf_death_urbanrural", "inf_death_age_sex","a
GF_labels <- c("Urban/Rural\n(Foetal Deaths)", "Age of Mother\n(Foetal Deaths)", "Gestational Age\n(Foe

det_heatmap(gfd_unfpa_clean, "AP", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels, GF_level
```



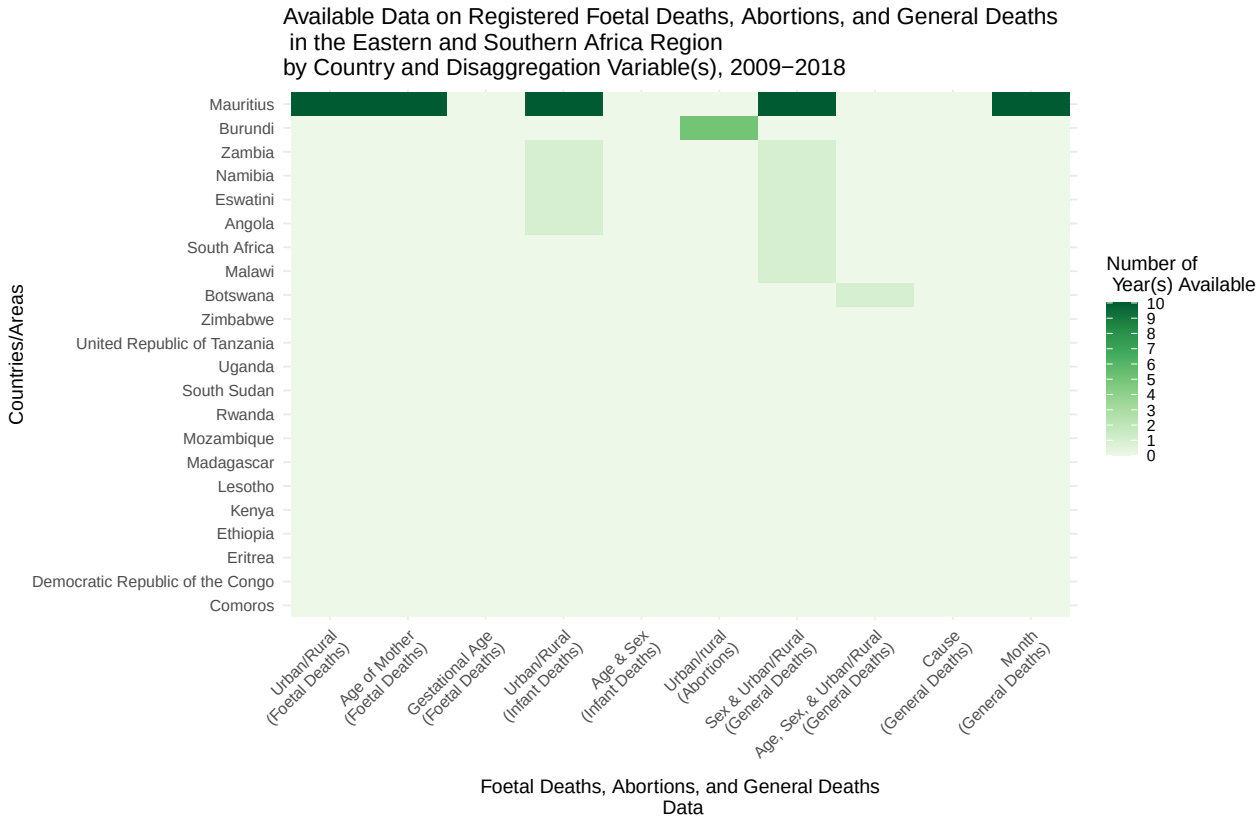

```
det_heatmap(gfd_unfpa_clean, "AS", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels, GF_level
```



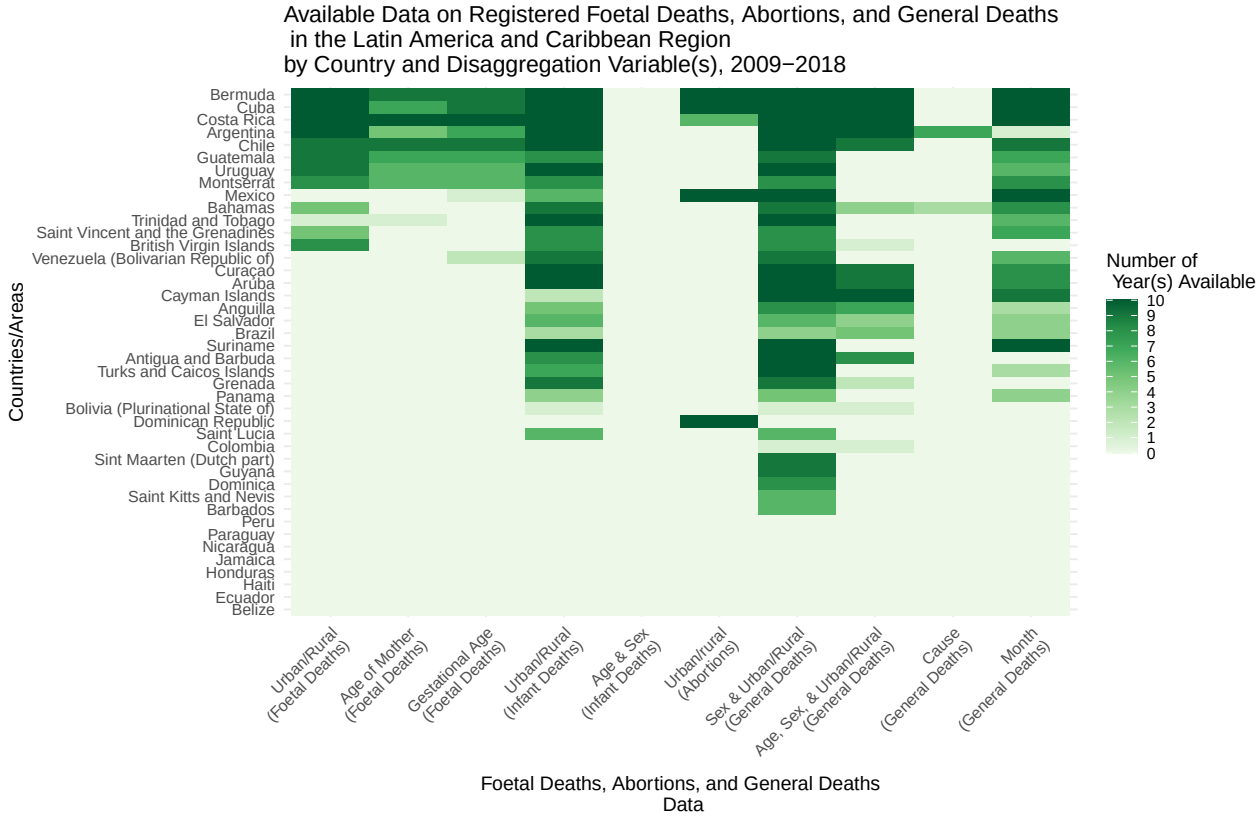
```
det_heatmap(gfd_unfpa_clean, "EECA", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels, GF_lev
```



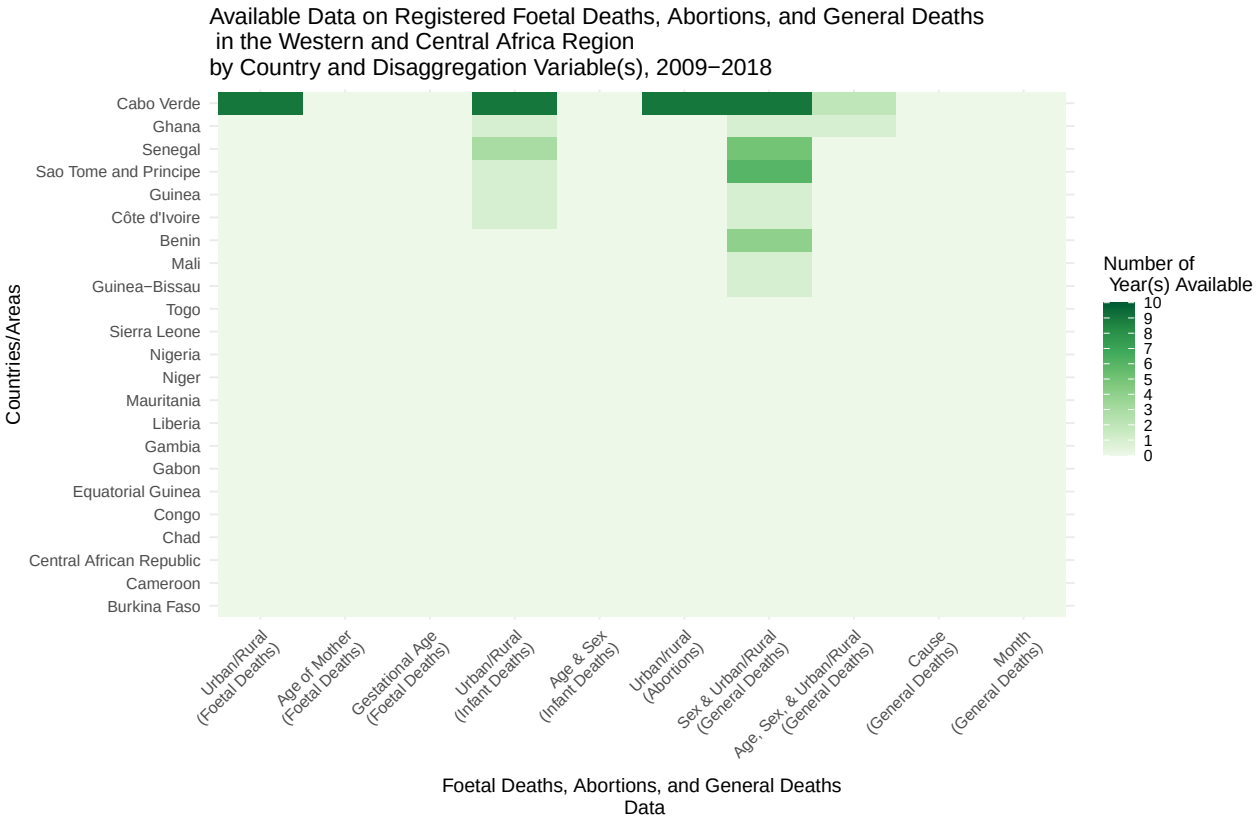

```
det_heatmap(gfd_unfpa_clean, "ESA", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels, GF_lev
```




```
det_heatmap(gfd_unfpa_clean, "LAC", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels, GF_leve
```



```
det_heatmap(gfd_unfpa_clean, "WCA", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels, GF_leve
```



Create Steps to Include Complete Data of Tables with "latest available year"

Live Births -Table 10 - Live births by age of mother and sex of child, general and age-specific fertility rates: latest available year, 2009-2018 -Table 11 - Live births and live birth rates by age of father: latest available year, 2009-2018

Marriages -Table 23 - Marriages by age of groom and by age of bride: latest available year, 2009-2018

Divorces -Table 25 - Divorces and percentage distribution by duration of marriage, latest available year: 2009-2018

Foetal Deaths and Abortions -Table 14 - Legally induced abortions by age and number of previous live births of women: latest available year, 2009-2018

Infant Deaths -Table 16 - Infant deaths and infant mortality rates by age and sex, latest available year: 2009-2018

General Deaths -Table 19 - Deaths by age and sex, age-specific death rates by sex: latest available year, 2009-2018

```
# Test steps with Table 19 - General Death, Age and Sex
all_gds_age_sex <- list.files(path = "./Gen Death", # Load all General Deaths by Age and Sex, Latest Av
                          pattern = "table19",
                          full.names = TRUE)
```

```r
all_gds_age_sex_excel <- # Convert into Excel
  lapply(all_gds_age_sex, function (i) {
  x = read_excel(i)
})
```

```r
# Create Freq Table Function for non-Urban/Rural DYB Tables, modified
la_transform_mod <- function(data_list, data_type) {

  data_list_tfr<-
  lapply(data_list, function (i) {
  x = ap_transform(i, data_type)
})

  data_1<-rbindlist( data_list_tfr)

   data_1<-
    data_1 %>%
    group_by(countries_areas) %>%
    summarise(n=n()) %>%
    mutate("data_available" = data_type)

  return(data_1)
}
```

```r
#Create dataset of previous years
all_gds_age_sex_long <- la_transform_mod(all_gds_age_sex_excel, "gd_age_sex")
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 105 rows [5, 9, 13,
## 15, 19, 25, 32, 40, 42, 46, 52, 54, 58, 64, 66, 70, 76, 78, 80, 82, ...].

## `summarise()` ungrouping output (override with `.groups` argument)

## Warning: Expected 1 pieces. Additional pieces discarded in 106 rows [5, 11, 13,
## 15, 19, 25, 36, 38, 42, 50, 52, 56, 62, 64, 68, 76, 78, 80, 82, 84, ...].

## `summarise()` ungrouping output (override with `.groups` argument)

## Warning: Expected 1 pieces. Additional pieces discarded in 107 rows [6, 16, 18,
## 20, 22, 26, 32, 36, 44, 46, 50, 58, 60, 64, 68, 71, 79, 81, 83, 85, ...].

## `summarise()` ungrouping output (override with `.groups` argument)

## Warning: Expected 1 pieces. Additional pieces discarded in 109 rows [3, 8, 18,
## 20, 22, 26, 28, 30, 32, 36, 40, 48, 50, 54, 61, 63, 67, 71, 74, 82, ...].

## `summarise()` ungrouping output (override with `.groups` argument)

## Warning: Expected 1 pieces. Additional pieces discarded in 107 rows [3, 8, 12,
## 20, 22, 24, 28, 30, 32, 34, 40, 44, 53, 57, 61, 68, 72, 76, 79, 87, ...].

## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 104 rows [3, 8, 12,
## 20, 22, 26, 28, 30, 36, 38, 49, 53, 60, 64, 66, 70, 71, 79, 81, 83, ...].

## `summarise()` ungrouping output (override with `.groups` argument)

## Warning: Expected 1 pieces. Additional pieces discarded in 108 rows [3, 9, 13,
## 21, 25, 29, 31, 33, 39, 41, 46, 54, 55, 57, 64, 68, 70, 76, 79, 87, ...].

## `summarise()` ungrouping output (override with `.groups` argument)
## `summarise()` ungrouping output (override with `.groups` argument)
```

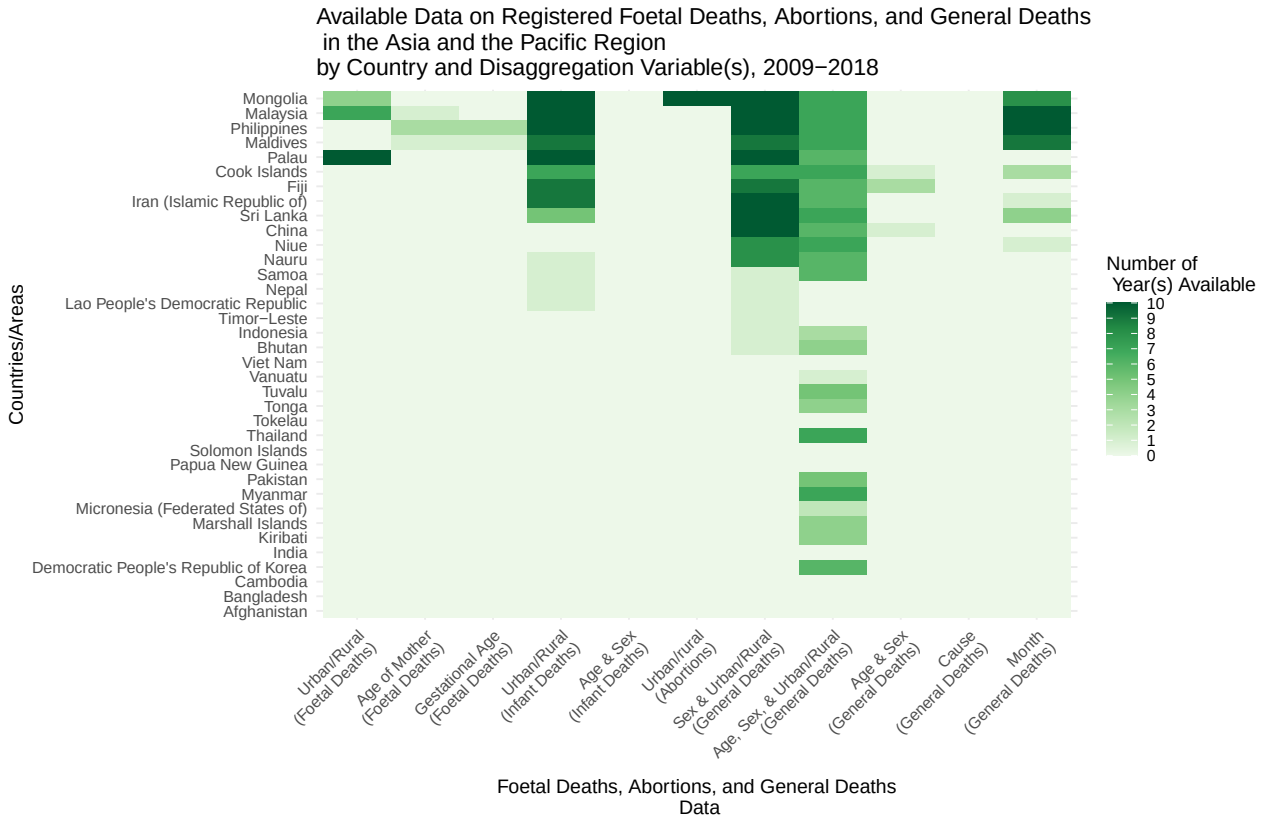
Test Visualization

```
lookup_gfd_unfpa_mod <- lookup_score(c("inf_death_age_sex", "inf_death_urbanrural", "gm_cause", "gm_age
```

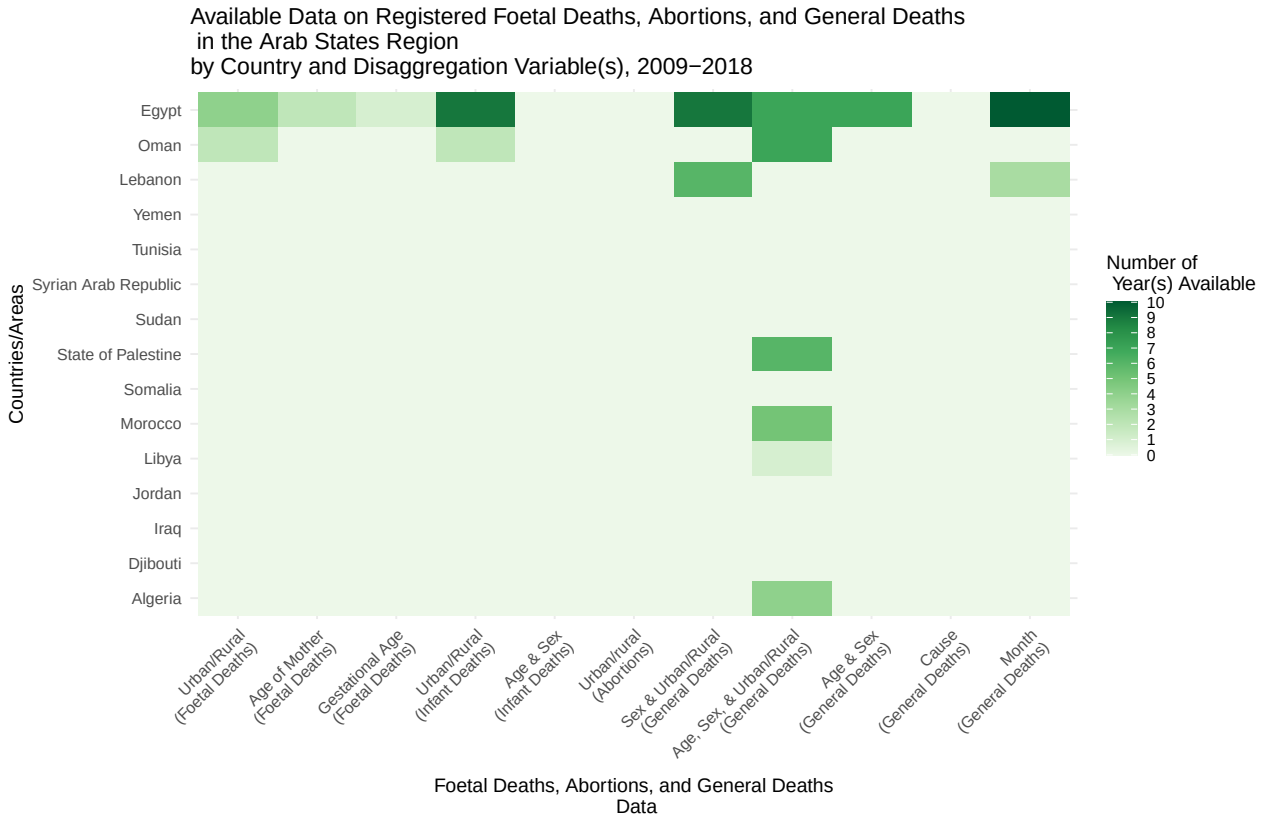
```
all_gfd_tables_mod <- bind_rows(rbindlist(all_gds_long), rbindlist(all_fds_long), all_fds_long, id_ages
all_gfd_unfpa_mod <- left_join(lookup_gfd_unfpa_mod, all_gfd_tables_mod, by = c("countries_areas", "dat
```

```
gfd_unfpa_clean_mod <-
  all_gfd_unfpa_mod %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  mutate(score = ((complete/5)*0.85)+((x/50)*0.15)) %>%
  mutate (countries_areas = reorder(countries_areas, score))
```

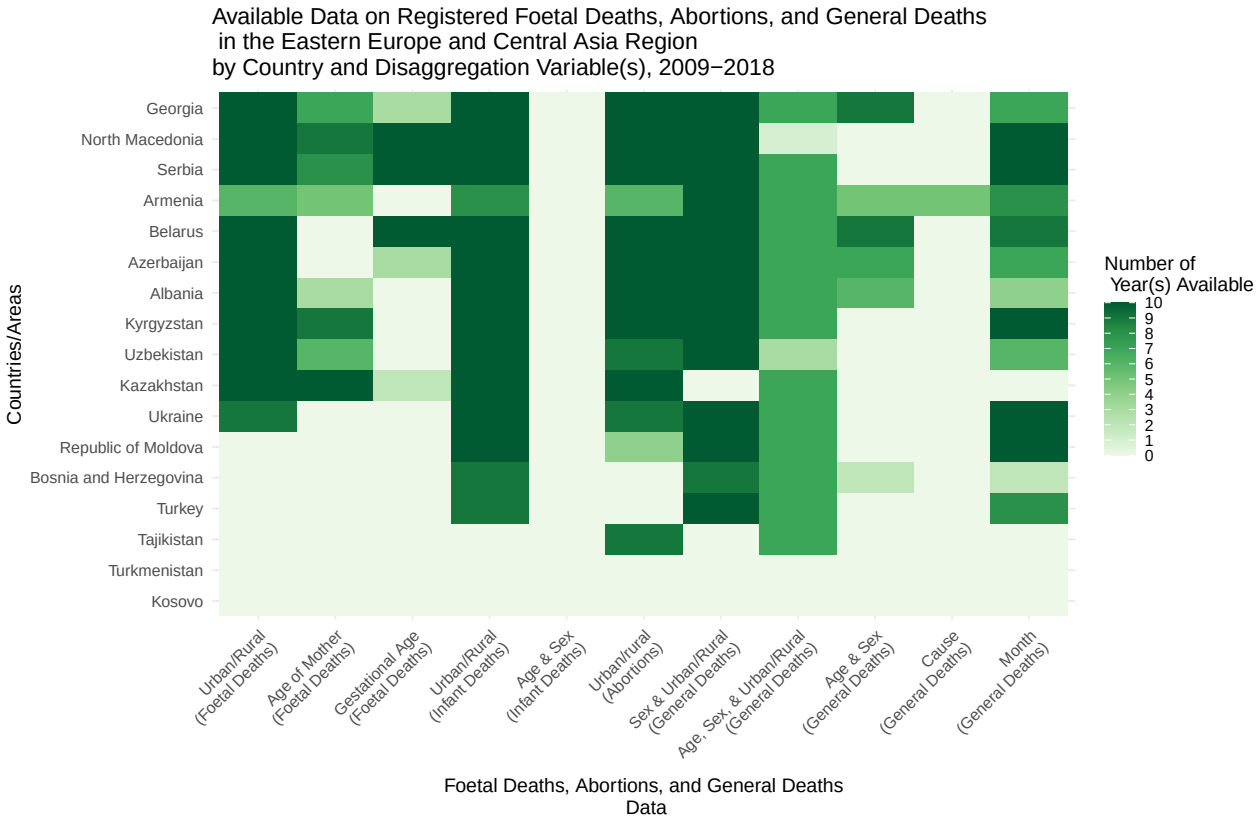
```
GF_levels_mod <- c("fd_sex_ur", "fd_agewoman", "fd_gest_age", "inf_death_urbanrural", "inf_death_age_se
GF_labels_mod <- c("Urban/Rural\n(Foetal Deaths)", "Age of Mother\n(Foetal Deaths)", "Gestational Age\n

det_heatmap(gfd_unfpa_clean_mod, "AP", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels_mod,
```



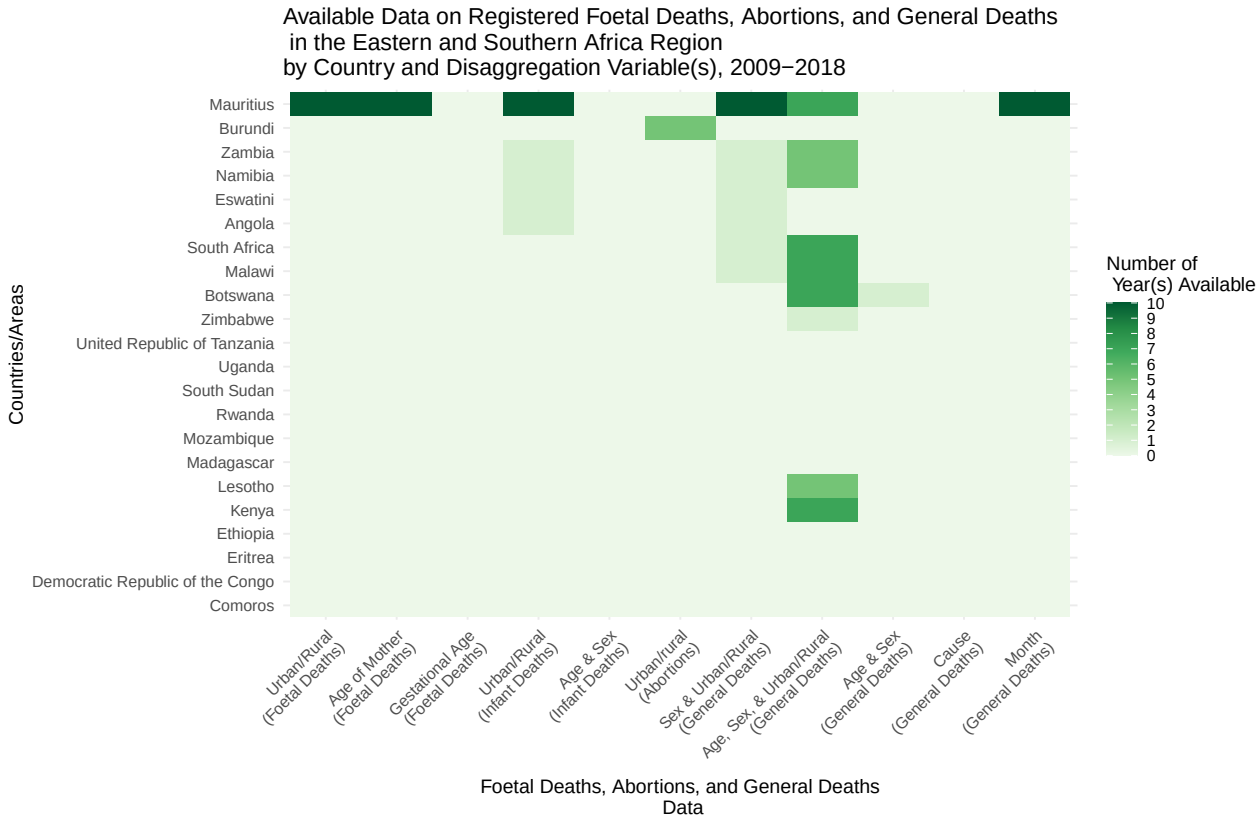

```
det_heatmap(gfd_unfpa_clean_mod, "AS", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels_mod,
```



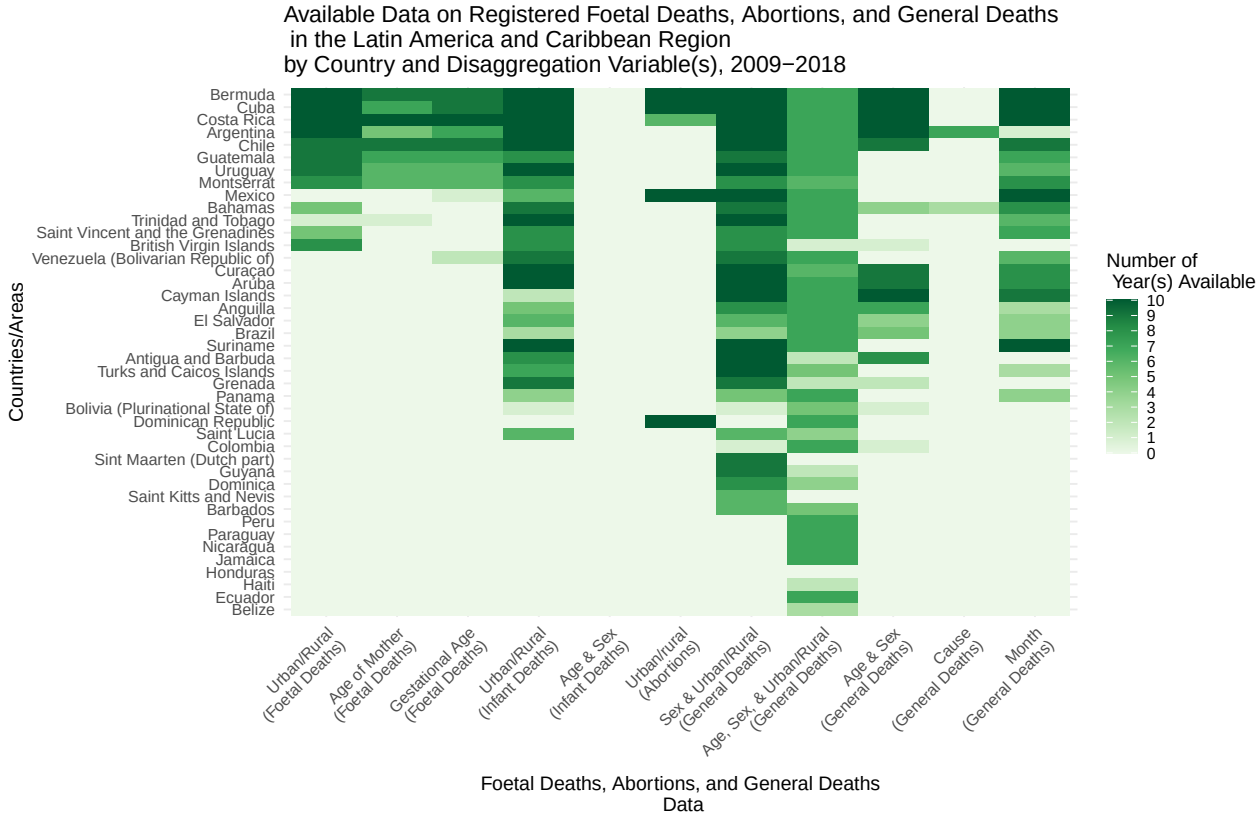
```
det_heatmap(gfd_unfpa_clean_mod, "EECA", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels_mod
```




```
det_heatmap(gfd_unfpa_clean_mod, "ESA", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels_mod
```




```
det_heatmap(gfd_unfpa_clean_mod, "LAC", "Foetal Deaths, Abortions, and General Deaths\n", GF_labels_mod
```



```
det_heatmap(gfd_unfpa_clean_mod, "WCA", "Foetal Deaths, Abortions, and General Deaths\n",GF_labels_mod,
```