

# Tostan MERL Graph Reproduction

This is an R Notebook to help reproduce graphs frequently made by MERL, using example data from Guinea.

R notebooks are a combination of code (in the grey boxes, called chunks) and surrounding text explanations. You can run the code for each chunk individually, making the graphs easily reproducible and reusable with new data.

First, let's read in the data we want.

```
getwd()
```

```
## [1] "C:/Users/Dina/Documents/Watson/Tostan"
```

```
d <- read.csv("Forms.csv", na.strings = '---', check.names=FALSE)
```

```
EF_data <- read.csv("17-02-21-GB_EF_CG3_Questionnaire individuel-v2.csv", na.strings = '---', check.names=FALSE)
```

Unfortunately, the variable names from the server are not very compatible with R. This code fixes the variable names.

```
source("clean_names.R")
```

```
names(d) <- clean_names(d)
```

```
names(EF_data) <- clean_names(EF_data)
```

We can visually check and see that the variable names look right by looking at the first few variables visually.

```
head(names(d),10)
```

```
## [1] "number" "formid"
## [3] "departement" "commune"
## [5] "village" "type_de_communaut"
## [7] "type_evaluation" "coordonnes_gps_latitudedegrees"
## [9] "coordonnes_gps_longitudedegrees" "coordonnes_gps_altitudemeters"
```

Hopefully, those variable names now look good. Next, we can make some graphs!

Now trying to get the data into the proper format for the graph below

```
d$country <- "Gambia"
```

```
head(d$country,10)
```

```
## [1] "Gambia" "Gambia" "Gambia" "Gambia" "Gambia" "Gambia" "Gambia" "Gambia"
```

```
## [8] "Gambia" "Gambia" "Gambia"
```

```
registered_girls <- sum(d$GOUV4_nbre_fille_enregistre, na.rm=TRUE)
```

```
registered_boys <- sum(d$GOUV5_nbre_garcon_enregistre, na.rm=TRUE)
```

```
total_kids <- sum(d$nombre_denfants, na.rm=TRUE)
```

```
EB_perc <- (registered_boys + registered_girls) / total_kids
```

```
EB_perc
```

```
## [1] 0.5494845
```

```
table(d$interviewer[d$nombre_denfants>0])
```

```
##
```

```
## membre_simple participant
```

```
## 184 1
```

```
k1 <- sum(EF_data$child_under_one_declared[EF_data$interviewer=="participant"], na.rm=TRUE)
```

```
k2 <- sum(EF_data$child_between_1_and_5_declared[EF_data$interviewer=="participant"], na.rm=TRUE)
```

```
total_kids <- sum(EF_data$nombre_denfants[EF_data$interviewer=="participant"], na.rm=TRUE)
EF_perc <- (k1+k2) / total_kids
table(EF_data$interviewer[EF_data$nombre_denfants>0])
```

```
##
##      adopte membre_simple participant
##      118      369      90
country <- c('GB', 'GB', 'GB', 'SEN', 'SEN', 'SEN')
interviewer <- c('simple', 'adoptee', 'participant', 'simple', 'adoptee', 'participant')
period <- c('EB', 'EF', 'EF', 'EB', 'EF', 'EF')
percentage_declared <- c(.5, .5, .4, .3, .6, .2)
x <- data.frame(country, interviewer, period, percentage_declared)
ftable(xtabs(percentage_declared ~ country+period+interviewer, data = x))
```

```
##               interviewer adoptee participant simple
## country period
## GB      EB              0.0              0.0      0.5
##      EF              0.5              0.4      0.0
## SEN      EB              0.0              0.0      0.3
##      EF              0.6              0.2      0.0
```

```
#barplot(as.matrix(x), main="Title", xlab="Phase", ylab="Percentage", names.arg=c("EB", "EF", "EF"),
#beside = TRUE)
#legend("topright", c("First", "Second", "third"), cex=1.3, bty="n")
```

Okay to do things with lots of factors it looks like lattice is a good choice. Explanations I liked are <https://www.statmethods.net/advgraphs/trellis.html> for basic plot organization <https://magesblog.com/post/2012-12-04-changing-colours-and-legends-in-lattice/> for styling

```
library(RColorBrewer)
myColours <- brewer.pal(4, "Blues")

my.settings <- list(
  superpose.polygon=list(col=myColours[2:4], border="transparent"),
  strip.background=list(col=myColours[1]),
  strip.border=list(col="black")
)

library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.3.3
```

```
barchart(percentage_declared*100~period|country,
  groups=interviewer,data=x,
  ylim=c(0,100),
  ylab = "Percentage Declared",
  main = "Title Here",
  auto.key=list(space="bottom", columns=3, title="District", cex.title=1),
  par.settings = my.settings
)
```

# Title Here

