script002.R

2018-04-20

This script analysis the 1950 to 2014 demographics data of Bangladesh. (Complete document)

Getting started

Load and Prepare data

Set working directory:

```
setwd("~/Desktop/github/Bangladesh_demographics")
```

Load data:

```
df_population <- read.csv("data/Bangladesh_population_history.csv")
df_indicator <- read.csv("data/Bangladesh_indicator_history.csv")
```

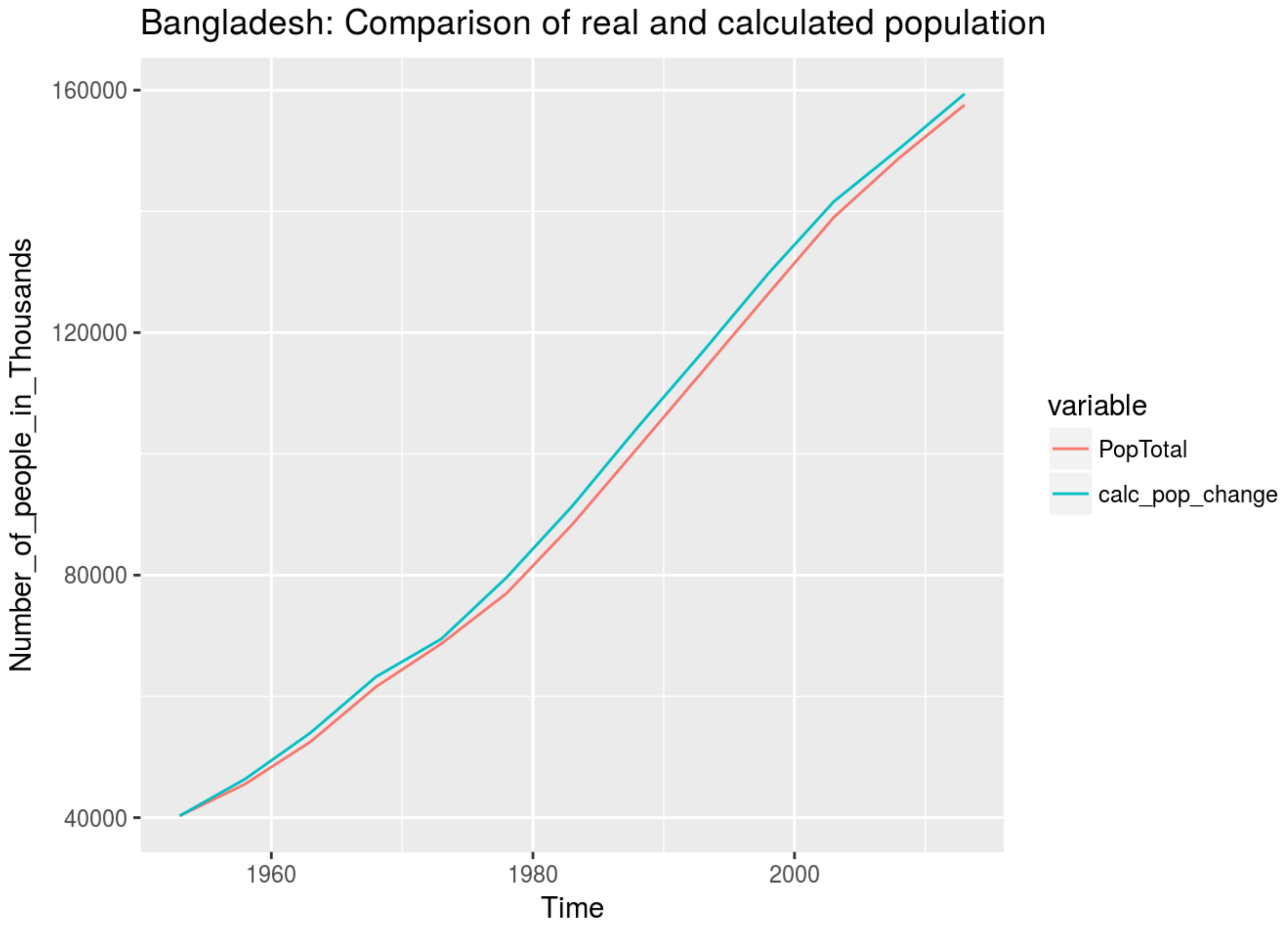
Prepare and merge data.frames:

```
df_indicator2 <- df_indicator # ajust MidPeriod entries so that they can be used a common column for merger
colnames(df_indicator2)[5] <- "TimePeriod" # change col.name to "TimePeriod"" to avoid confounding it with df_total's "time" values
df_indicator2[,"MidPeriod"] <- df_indicator2[,"MidPeriod"] + 0.5
# There is another potential error source in this data.frame. The entries for Births, Deaths, DeathsMale, DeathsFemale, NetMigrations are calculated for the 5 year bins. We have to divide these values by 5 if we want the average annual values:
df_indicator2[,c("Births", "Deaths", "DeathsMale", "DeathsFemale", "NetMigrations")] <- df_indicator2[,c("Births", "Deaths", "DeathsMale", "DeathsFemale", "NetMigrations")]/5
df_total <- merge(df_population, df_indicator2, by=c("LocID", "Location", "VarID", "Variant", "MidPeriod"))
```

Calculate population based on births, deaths and migration - compare to real data

A short calculation to understand the data structure: I calculate the theoretical population for each year based on the start value in 1953 and by adding and substracting Birth, Deaths and NetMigratrions throughout the years:

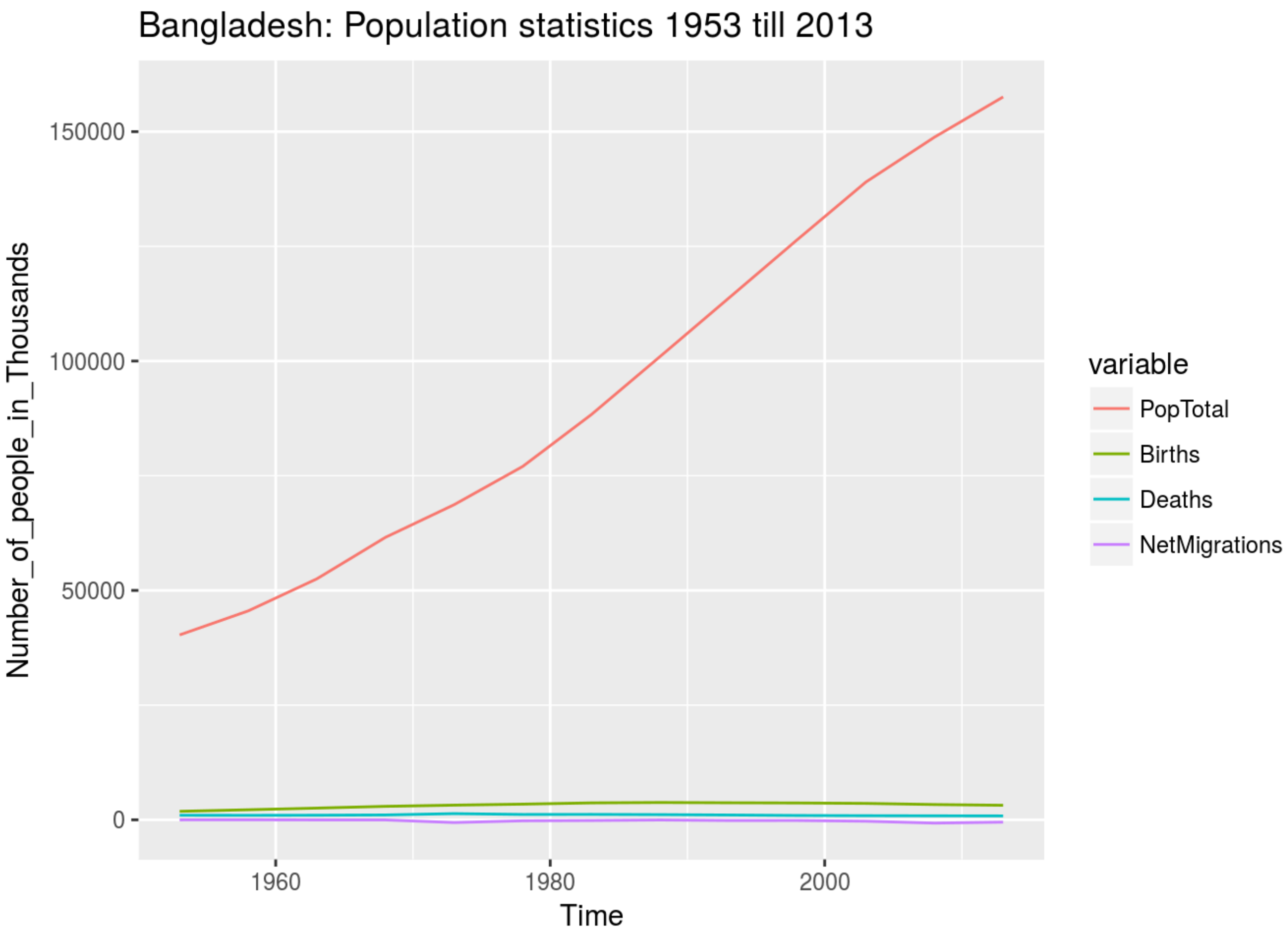
```
# calculating
pop_1953 <- df_total[1,"PopTotal"]
pop_change <- vector(mode="numeric") # create empty vector
pop_change[1] <- pop_1953 # 1st entry = population of 1953
for (i in 2:nrow(df_total)){
pop_change[i] <- pop_change[i-1] + 5*(df_total[i, "Births"] - df_total[i, "Deaths"] + df_total[i, "NetMigrations"])
}
df_total[,"calc_pop_change"] <- pop_change # add to data.frame

# plotting the data
df_total_long <- melt(df_total[,c("Time", "PopTotal", "Births", "Deaths", "NetMigrations", "calc_pop_change")], id="Time")
colnames(df_total_long)[3] <- "Number_of_people_in_Thousands"
subset <- which(df_total_long[,"variable"] %in% c("PopTotal", "calc_pop_change")) # for comparison of real and calculated population
ggplot(data=df_total_long[subset,], aes(x=Time, y=Number_of_people_in_Thousands, colour=variable)) +
  geom_line() +
  ggtitle("Bangladesh: Comparison of real and calculated population")
```



Comparison of real population and calculated population (based on births, deaths, and migration). The graph shows that births, deaths, and migration are sufficient for predicting population growth.

Plotting the population data in one graph:

```
subset <- which(df_total_long[,"variable"] != "calc_pop_change")
ggplot(data=df_total_long[subset,], aes(x=Time, y=Number_of_people_in_Thousands, colo
ur=variable)) +
  geom_line() +
  ggtitle("Bangladesh: Population statistics 1953 till 2013")
```



Bangladesh: Overview showing total Population, Births, Deaths, Migration in one graph. Data range from 1953 till 2013.

Excursion: Assessing the impact of the Liberation War

Calculating the impact of the war on vital statistics: Determine row for period 1970 to 1975:

```
which(df_total[,"TimePeriod"] == "1970-1975")
```

```
## [1] 5
```

```
war_deaths_thousands <- 5*(df_total[5,"Deaths"] - df_total[6, "Deaths"])
war_deaths <- 1000 * war_deaths_thousands
war_deaths
```

```
## [1] 913908
```

```
war_migrants_thousands <- 5*(df_total[5,"NetMigrations"] - df_total[6, "NetMigration
s"])
war_migrants <- 1000 * war_migrants_thousands
war_migrants
```

```
## [1] -1901122
```

```
rm(war_deaths, war_deaths_thousands, war_migrants, war_migrants_thousands)
```

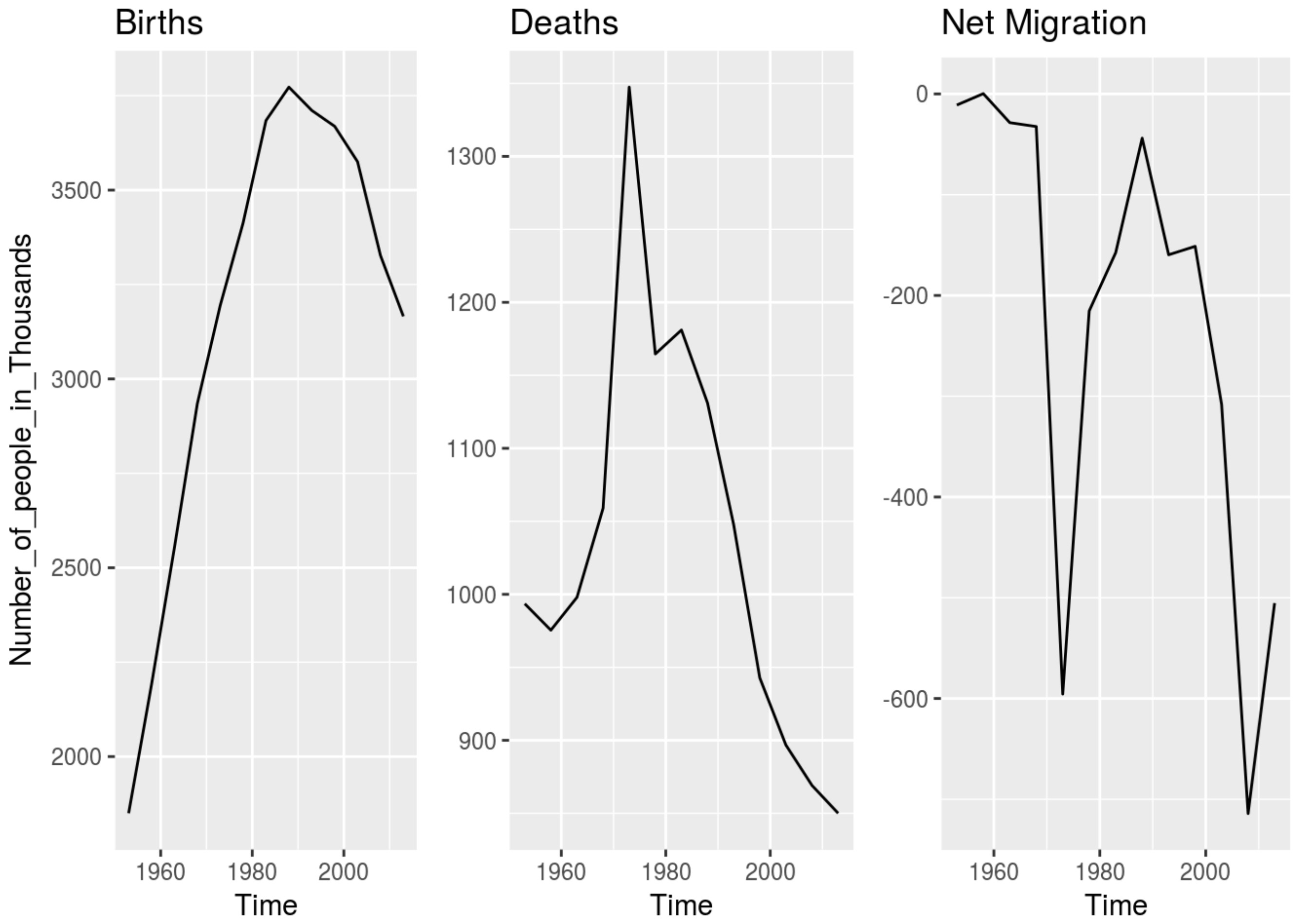
You can see the impact of the Bangladesh Liberation War in 1971. In the interval from 1970 to 1975, the population suffered 913908 additionals death compared to the years 1975 to 1980 (where the overall population was higher). In addition, the war lead to an increase in outward migration of 1.9 Mio people when compared to the 1975-1980 period.

Finding a function to characterize births, deaths and migration

I will search for individal functions for the data series for “Birth”, “Deaths”, and “NetMigrations”. Based on these functions, I will estimate Bangladesh’s future population.

At first, I plot all three data sets individually:

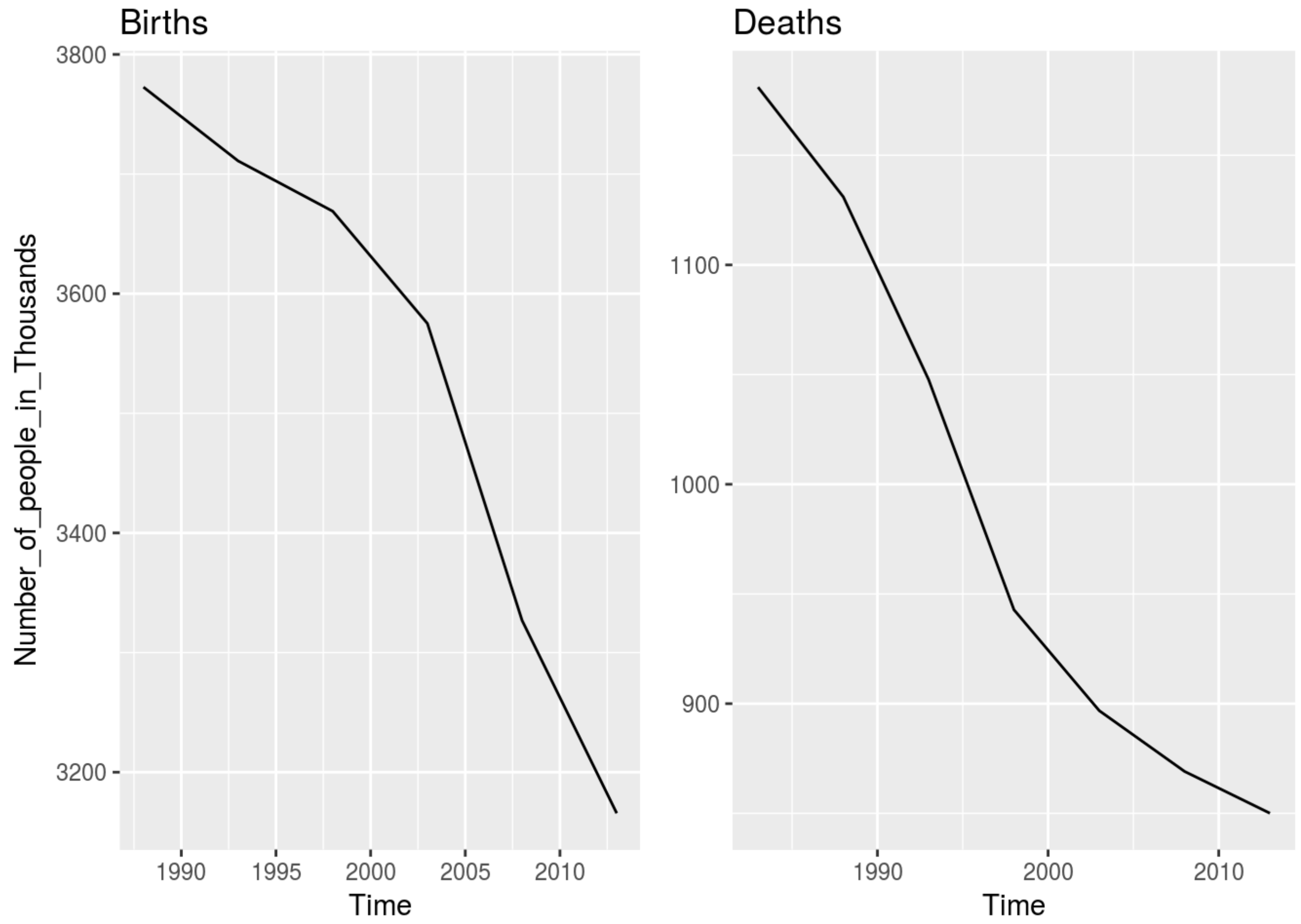
```
subset1 <- which(df_total_long[,"variable"] == "Births")
subset2 <- which(df_total_long[,"variable"] == "Deaths")
subset3 <- which(df_total_long[,"variable"] == "NetMigrations")

plot1 <- ggplot(data=df_total_long[subset1,], aes(x=Time, y=Number_of_people_in_Thous
ands))
plot1 <- plot1 + geom_line() + ggtitle("Births")

plot2 <- ggplot(data=df_total_long[subset2,], aes(x=Time, y=Number_of_people_in_Thous
ands))
plot2 <- plot2 + geom_line() + ggtitle("Deaths") + theme(axis.title.y = element_blank
())

plot3 <- ggplot(data=df_total_long[subset3,], aes(x=Time, y=Number_of_people_in_Thous
ands))
plot3 <- plot3 + geom_line() + ggtitle("Net Migration") + theme(axis.title.y = elemen
t_blank())

grid.arrange(plot1, plot2, plot3, ncol=3)
```



PLots for births, deaths, and migrations. The birth rate peaked in the 1980s and is on a moderate decline since that time. Both migration and deaths show the effect of the Liberation War. *Births* and *Deaths* underwent a radical change between 1980 and 1990. The transitions took place during the precidency of Hussain Muhammad Ershad (1983 to 1990).

I will therefore focus the fitting curves on the period starting with the peak in the 1980s till the last available data (2013). At first, I will plot again the relevant sections of the data:

```
x1 <- max(df_total[df_total[,"Time"]>1980,"Births"])
x2 <- which(df_total[,"Births"] == x1)
y1 <- max(df_total[df_total[,"Time"]>1980,"Deaths"])
y2 <- which(df_total[,"Deaths"] == y1)
z <- nrow(df_total)

plot1 <- ggplot(data=df_total_long[subset1[x2:z],], aes(x=Time, y=Number_of_people_in
_Thousands))
plot1 <- plot1 + geom_line() + ggtitle("Births")

plot2 <- ggplot(data=df_total_long[subset2[y2:z],], aes(x=Time, y=Number_of_people_in
_Thousands))
plot2 <- plot2 + geom_line() + ggtitle("Deaths") + theme(axis.title.y = element_blank
())

grid.arrange(plot1, plot2, ncol=2)
```



Births

There are only six data points for the time period of interest (1988-2013) for *Births*. Non-linear regression analysis with nls2 did not lead to any sensible results with these few data points. So I will make a number of assumptions in order to fit a function to the births-per-year development:

1, Bangladesh has roughly half the population of the USA and double the population of Iran

```
Country        Population   Births/year   Total fertility rate (TFR)
---------------------------------------------------------------------------
Bangladesh     160          3.2           2.2
United States  320          3.8           1.8
Iran            80          0.9           1.5

* Population and Births/year in millions
```

Source:
https://en.wikipedia.org/wiki/Demography_of_the_United_States (https://en.wikipedia.org/wiki/Demography_of_the_United_States)(accessed 2018-04-19)
https://en.wikipedia.org/wiki/Demographics_of_Iran (https://en.wikipedia.org/wiki/Demographics_of_Iran) (accessed 2018-04-19)

I therefore make the assumption that the following statement becomes true in the long run: All three countries will have roughly 1 Mio babies per 100 Mio inhabitants. The population of Bangladesh is still sharply increasing. I therefore assume that the current decrease in births will level out at around 2 Mio births/year.

2, The drop in Births during *Demographic transition* follows a negative logistic function.
See https://commons.wikimedia.org/wiki/File:Demographic-TransitionOWID.png (https://commons.wikimedia.org/wiki/File:Demographic-TransitionOWID.png) for an example curve.

The general formula for a logistic function is:

```
f(x) = L / (1 + e^(-k(x-x0)) )
```

And for the negative logistic function:

```
f(x) = L / (1 + e^(k(x-x0)) )
f(x) = L / (1 + (e^k)^(x-x0))
```

We need to add an additional constant as the function won’t level out at y=0:

```
f(x) = L / (1 + (e^k)^(x-x0)) + c
```

The variables are:

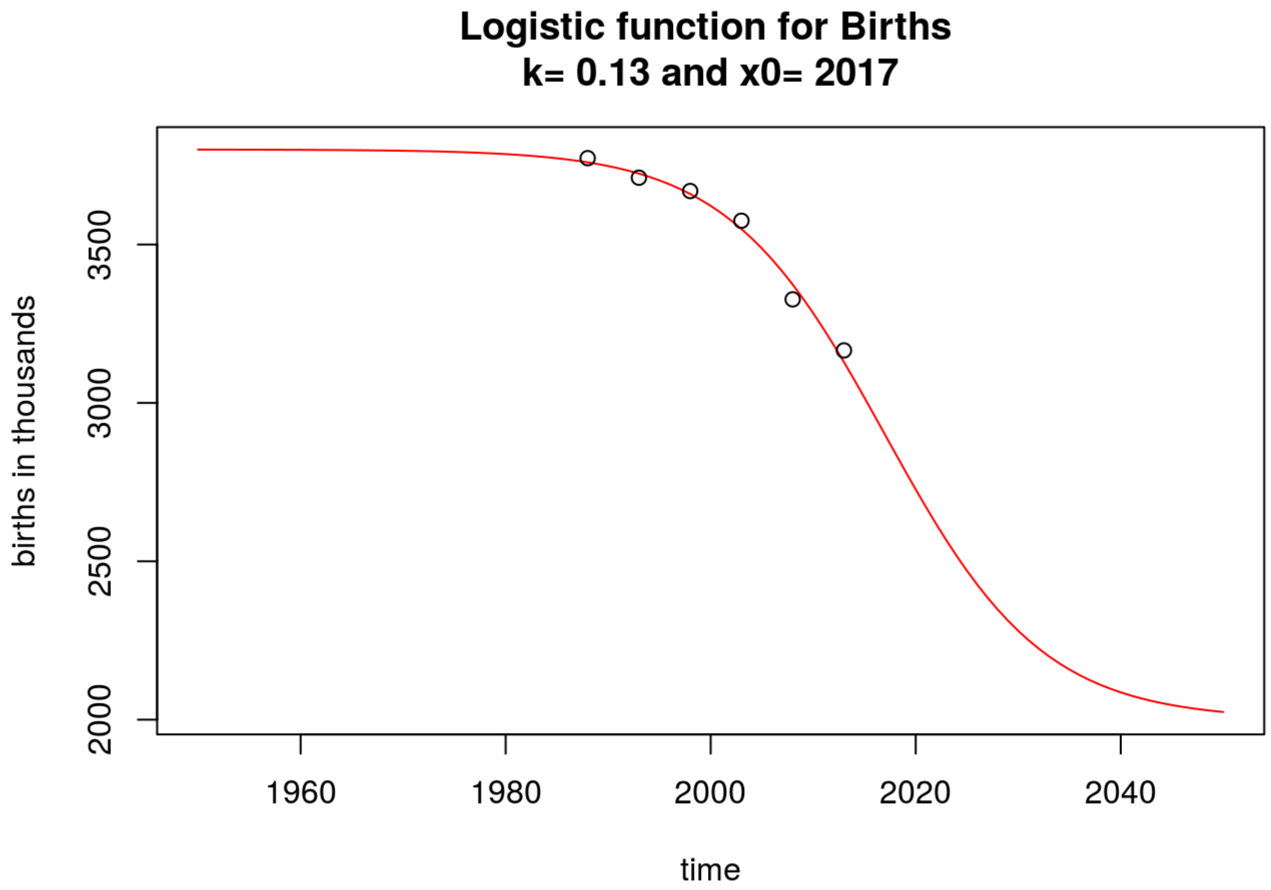
```
L   = max(no. of birth) - min(no. of birth)
c   = min(no. of birth)
x0  = The year where the function has its inflection point
e   = Eurler's number (2.718282)
k   = unknown factor influencing the steepness of the curve
```

3, So I need a negative logistic function that goes from around 3.8 Mio births/year in 1990 to 2 Mio births/year in the long run:

```
f(x) = (3.8^(10^6) - 2.0^(10^6))/(1+(e^k)^(x-x0)) + 2.0^(10^6)
```

Estimates for `k` and `x0` will be repeatedly guessed and plotted:

```
years <- 1950:2050
e <- exp(1) # Euler's number
k_Births = 0.13
x0_Births = 2017
birthsperyear <- ((3.8*10^3 - 2.0*10^3)/(1+(e^k_Births)^(years-x0_Births))) + (2.0*10
^3)
plot(x=years, y=birthsperyear, main = paste("Logistic function for Births", "\nk=",k_
Births , "and x0=",x0_Births), xlab="time", ylab="births in thousands", type="l",col=
"red")
points(x=df_total[8:13,"Time"], y=df_total[8:13,"Births"])
```



Logistic function projecting the development of births/year. k is the steepness estimator and x0 the midpoint (or inflection point) estimator.

Calculate estimated births numbers based on function and check fitting with Pearson correlation:

```
births_estimated <- ((3.8*10^3 - 2.0*10^3)/(1+(e^k_Births)^(df_total[x2:z,"Time"]-x0_
Births))) + (2.0*10^3)
births_reality <- df_total[x2:z,"Births"]
cor.test(as.numeric(births_estimated), births_reality, method="pearson")
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  as.numeric(births_estimated) and births_reality
## t = 15.938, df = 4, p-value = 9.059e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9276205 0.9991878
## sample estimates:
##       cor 
## 0.9922188
```

Conclusion: Function is a good fit for births/year.

Deaths

Similar to Births/year, Deaths/year is also following a negative logistic function according to the *Demographic transition* model. We will again take compare the approx. situation in 2013 in the USA, Iran and Bangladesh (numbers in thousands):

```
Country        Population   Deaths/year
---------------------------------------
Bangladesh     160          0.905
United States  320          2.777
Iran            80          0.440

* Population and Deaths/year in millions
```

Source: https://en.wikipedia.org/wiki/Demography_of_the_United_States (https://en.wikipedia.org/wiki/Demography_of_the_United_States)(accessed 2018-04-19) https://en.wikipedia.org/wiki/Demographics_of_Iran (https://en.wikipedia.org/wiki/Demographics_of_Iran) (accessed 2018-04-19)

Deaths/year will follow the same type of function as Births/year:

```
f(x) = L / (1 + (e^k)^(x-x0)) + c
```

The variables are:

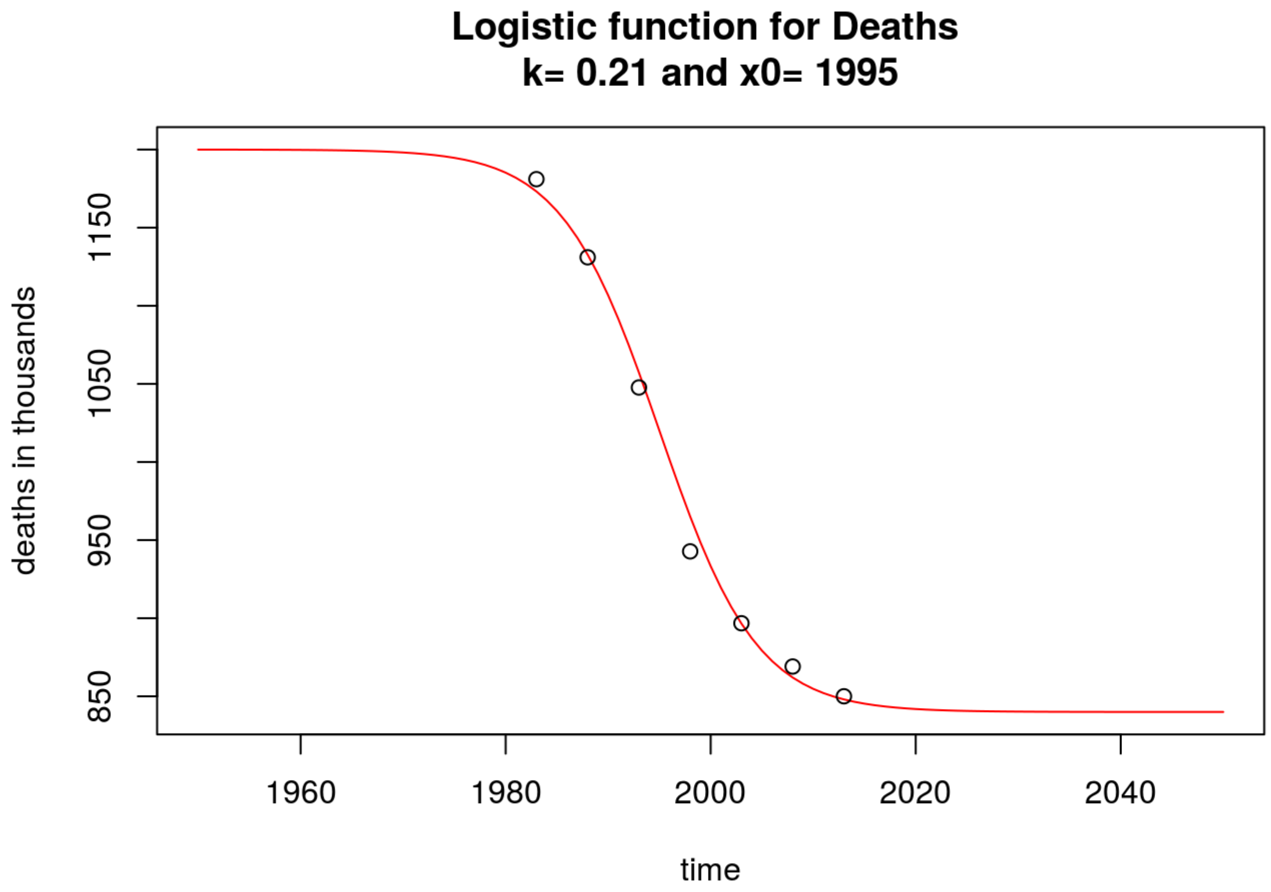
```
L   = max(no. of death) - min(no. of death)
c   = min(no. of death)
x0  = The year where the function has its inflection point
e   = Eurler's number (2.718282)
k   = unknown factor influencing the steepness of the curve
```

We need a negative logistic function that goes from around 1.2 Mio deaths/year in 1983 to 0.84 deaths/year in the long run:

```
f(x) = (1.20*10^6 - 0.84*10^6)/(1+(e^k)^(x-x0)) + 0.84*10^6
```

Estimates for `k` and `x0` will be repeatedly guessed and plotted:

```
e <- exp(1) # Euler's number
years <- 1950:2050
k_Deaths = 0.21
x0_Deaths = 1995
deathsperyear <- ((1.2*10^3 - 0.84*10^3)/(1+(e^k_Deaths)^(years-x0_Deaths))) + (0.84*
10^3)
plot(x=years, y=deathsperyear, main = paste("Logistic function for Deaths", "\nk=",k_
Deaths , "and x0=",x0_Deaths), xlab="time", ylab="deaths in thousands", type="l", col
="red")
points(x=df_total[7:13,"Time"], y=df_total[7:13,"Deaths"])
```



Logistic function projecting the development of deaths/year. k is the steepness estimator and x0 the midpoint (or inflection point) estimator.

Calculate estimated deaths numbers based on function and check fitting with Pearson correlation:

```
deaths_estimated <- ((1.2*10^3 - 0.84*10^3)/(1+(e^k_Deaths)^(df_total[x2:z,"Time"]-x0
_Deaths))) + (0.84*10^3)
deaths_reality <- df_total[x2:z,"Deaths"]
cor.test(x=deaths_estimated, y=deaths_reality, method="pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  deaths_estimated and deaths_reality
## t = 22.796, df = 4, p-value = 2.194e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9638096 0.9996013
## sample estimates:
##       cor
## 0.9961734
```

Conclusion: Function is a good fit for deaths/year.

Migration

The migration rate depends heavily on the political situation in Bangladesh and the surrounding countries. We already saw above the impact of the Liberation war on migration. The Rohinga crisis in neighoring Myanmar could lead to an ongoing influx of refugees to Bangladesh. It is very difficult to predict the future development.

For lack of a better strategy, I will fit a linear function that shall represent the longterm migration pattern.

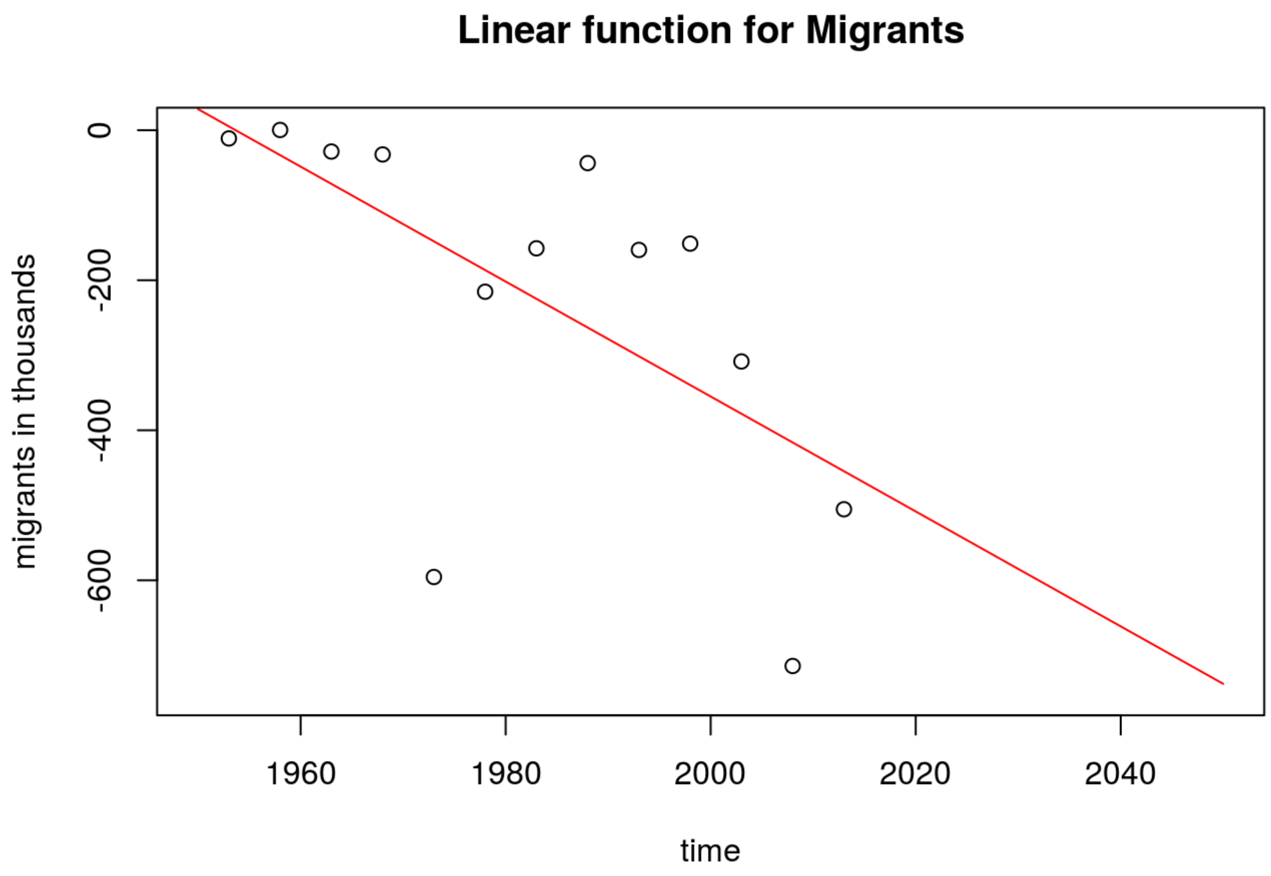
```
f(x) = a*x + b
```

Conceiving a linear function using linear regression (lm):

```
years <- 1950:2050
time <- df_total[,"Time"]
netmigration_reality <- df_total[,"NetMigrations"]
fit <- lm(netmigration_reality ~ time)
summary(fit)
```

```
##
## Call:
## lm(formula = netmigration_reality ~ time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -447.36  -28.84   43.13   77.54  219.23
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14964.961   5707.684   2.622   0.0237 *
## time           -7.660      2.878  -2.661   0.0221 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 194.1 on 11 degrees of freedom
## Multiple R-squared:  0.3917, Adjusted R-squared:  0.3364
## F-statistic: 7.083 on 1 and 11 DF,  p-value: 0.02213
```

```
a_migrants <- fit$coefficients["time"]
b_migrants <- fit$coefficients["(Intercept)"]
migrantsperyear <- a_migrants * years + b_migrants
plot(x=years, y=migrantsperyear, main = paste("Linear function for Migrants"), ylim=c
(-750, 0), xlab="time", ylab="migrants in thousands", type="l", col="red")
points(x=time, y=netmigration_reality)
```



Linear function projecting the development of migration. The modell shows a substantial increase in outward migration (negative migration ratio) over time.

Modelling the future population growth

The modell will be based on the relationship which was determined earlier on in this study:

```
population = births - deaths + migrants
```

Setting up the population modell:

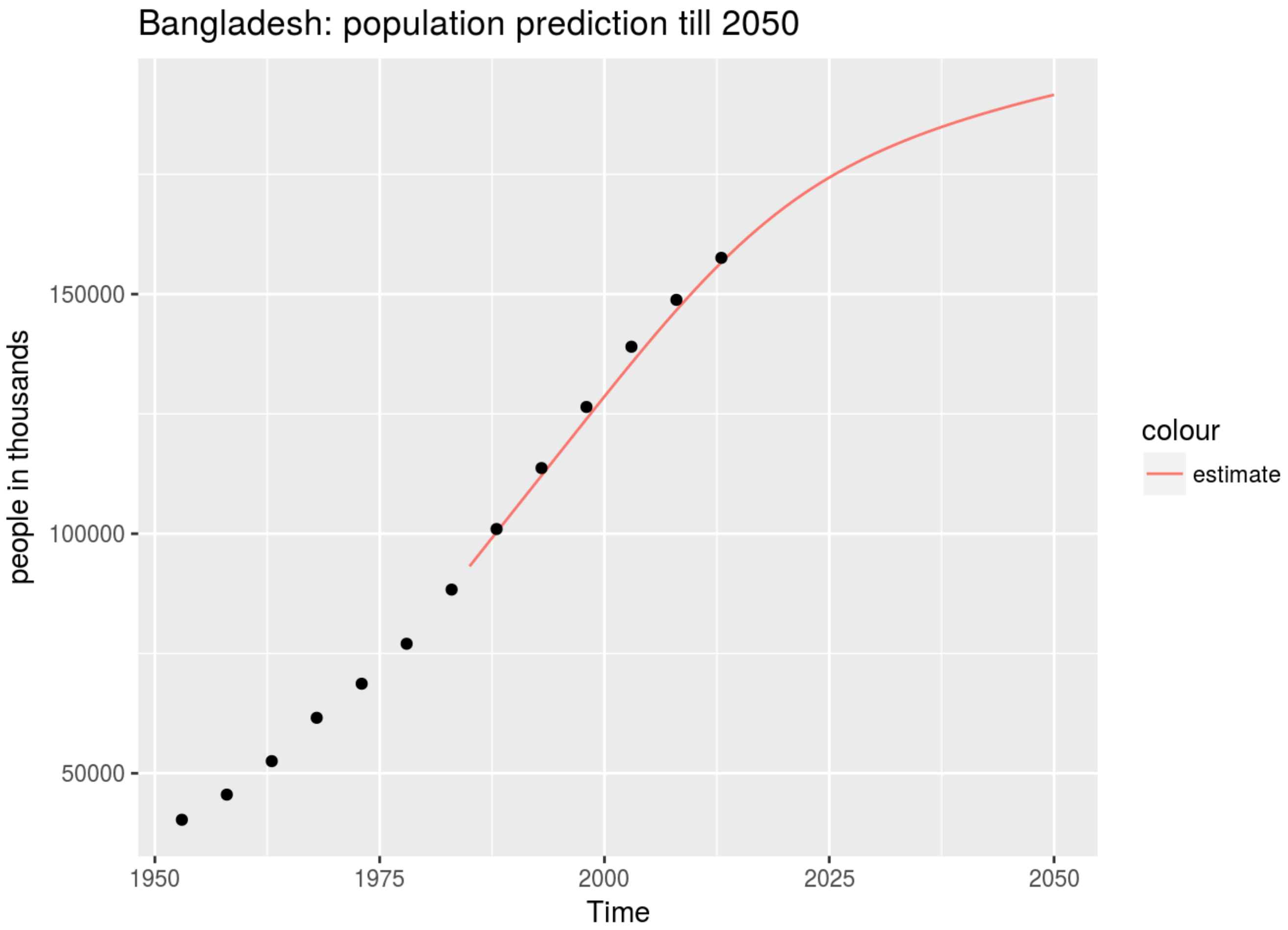
```
begin <- 1985
end <- 2100
pop_modell <- data.frame(begin:end)
pop_modell[1,2] <- df_population[begin-1950+1, "PopTotal"]
colnames(pop_modell) <- c("Time", "PopTotal")

for (i in begin:(end-1)) {  #start = 1983 till end = 2049 (!)
  j <- i + 1 - begin # start j = 1983 + 1 - 1983 = 1
  BIRTH <- (3.8*10^3 - 2.0*10^3)/(1+(e^k_Births)^(i-x0_Births)) + (2.0*10^3) #start = 1983
  DEATH <- (1.2*10^3 - 0.84*10^3)/(1+(e^k_Deaths)^(i-x0_Deaths)) + (0.84*10^3) #start = 1983
  MIGRA <- a_migrants * i + b_migrants #start = 1983
  pop_modell[j+1,2] <- pop_modell[j,2] + BIRTH - DEATH + MIGRA
  }
# Take a peak at some core dates:
key_dates <- pop_modell[which(pop_modell[,"Time"]==2025),]
key_dates <- rbind(key_dates, pop_modell[which(pop_modell[,"Time"]==2050),])
key_dates <- rbind(key_dates, pop_modell[which(pop_modell[,"Time"]==2075),])
key_dates <- rbind(key_dates, pop_modell[which(pop_modell[,"Time"]==2100),])
key_dates
```

```
##     Time PopTotal
## 41  2025 174335.9
## 66  2050 191599.9
## 91  2075 200044.3
## 116 2100 203515.7
```

```
# Check statistical congruency for available period 1985-2013:
pop_comparison <- merge(df_population, pop_modell, by="Time")
cor.test(x=pop_comparison[,"PopTotal.x"], y=pop_comparison[,"PopTotal.y"], method="pearson")
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  pop_comparison[, "PopTotal.x"] and pop_comparison[, "PopTotal.y"]
## t = 115.5, df = 28, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9977733 0.9995071
## sample estimates:
##       cor 
## 0.9989521
```

```
# plot modell
# plot(pop_modell, main = paste("Population dev estimation","\nStarting=",begin), xla
b="time", ylab="people in thousands", type="l", col="red", xlim=c(1950,end), ylim=c
(0,200000))
# points(x=df_total[,"Time"], y=df_total[,"PopTotal"])
cutoff <- which(pop_modell[,"Time"]==2050)
ggplot(pop_modell[1:cutoff,], aes(x=Time, y=PopTotal)) +
  geom_line(aes(colour="estimate")) +
  ylab("people in thousands") +
  ggtitle("Bangladesh: population prediction till 2050") +
# theme(legend.position = "none") +
  geom_point(data=df_total, aes(x=Time, y=PopTotal))
```



Estimation of future population growth. The projection fits well the observed variables. The modell starts only in the year 1985, hence the late onset of the line.

Conclusion

A population growth modell based on births/year, deaths/year and net migration is a reliable fit to the observed population data from the past.
The modell projects Bangladesh's population to reach 192 Mio in 2050 and 203 Mio in the far future of 2100.
According to the prediction, the population growth will eventually level out slightly above 200 Mio people.

You can compare the data with the projections from the Population Division of the Department of Economic and Social Affairs of the United Nations:
https://esa.un.org/unpd/wpp/Graphs/Probabilistic/POP/TOT/
(https://esa.un.org/unpd/wpp/Graphs/Probabilistic/POP/TOT/)
The UN median prediction has a similar value in the year 2050. According to the UN modell, the population

might experience a slight decline in the second half of the 21 century.

Among the countries with more than 10 Mio inhabitants, Bangladesh will remain the most densly populated nation for the foreseeable future. The great number of people in this relatively small country will continue to be a tremendous social, political, economic and environmental challenge. But Bangladesh’s society is in the middle of the democraphic transition process. Unlike many African states, Bangladesh has managed to bring its free-running population growth under control. This might prove to be a valuable asset in the fight against poverty. In the ideal case, Bandladesh might become a role model for regional neighbours with similar problems such as India, Pakistan, and Afghanistan.