

Project_indicators

Pablo Medrano

29/3/2018

Load libraries

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.4.4
```

```
library(sqldf)
```

```
## Warning: package 'sqldf' was built under R version 3.4.4
```

```
## Loading required package: gsubfn
```

```
## Warning: package 'gsubfn' was built under R version 3.4.4
```

```
## Loading required package: proto
```

```
## Warning: package 'proto' was built under R version 3.4.4
```

```
## Loading required package: RSQLite
```

```
## Warning: package 'RSQLite' was built under R version 3.4.4
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.4.4
```

```
## -----
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)
```

```
## -----
```

```
##  
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##   arrange, count, desc, failwith, id, mutate, rename, summarise,  
##   summarize
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(stats)  
library(imputeTS)
```

```
## Warning: package 'imputeTS' was built under R version 3.4.4
```

```
library(MTS)
```

```
## Warning: package 'MTS' was built under R version 3.4.4
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.4.4
```

```
##  
## Attaching package: 'tseries'
```

```
## The following object is masked from 'package:imputeTS':  
##  
##   na.remove
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.4.4
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 3.4.4
```

```
##  
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:forecast':  
##  
## accuracy
```

Load dataset

```
world_indicators <- read_csv("indicators.csv")
```

```
## Parsed with column specification:  
## cols(  
##   CountryName = col_character(),  
##   CountryCode = col_character(),  
##   IndicatorName = col_character(),  
##   IndicatorCode = col_character(),  
##   Year = col_integer(),  
##   Value = col_double()  
## )
```

Filter countries we want to study: China, France, India, Spain and USA. Create a dataset with all the indicators of all the countries we want to study: `our_indicators` Then create a dataset for each country containing all the indicators for that country: `COUNTRY_indicators`.

```
our_indicators <- filter(world_indicators, CountryCode %in% c("CHN", "USA", "IND", "ESP", "FRA"))  
CHN_indicators <- our_indicators[our_indicators$CountryCode == 'CHN', ]  
USA_indicators <- our_indicators[our_indicators$CountryCode == 'USA', ]  
IND_indicators <- our_indicators[our_indicators$CountryCode == 'IND', ]  
ESP_indicators <- our_indicators[our_indicators$CountryCode == 'ESP', ]  
FRA_indicators <- our_indicators[our_indicators$CountryCode == 'FRA', ]
```

Get all the different indicator names to choose the ones we think that are relevant for our question

```
indicatornames <- subset(sqldf('select distinct IndicatorName from world_indicators'))
```

Select the indicators we want to use as predictors for every country. First we create a data frame for the country. Then we obtain the values for each indicator from the country's indicators dataset and save them in a vector. After that a vector with length all possible years(1960-2015) is created and filled with NA values. The NA values are replace with the known values we got from the data set. If after that there are any NA values still in the vector, those are replace using kalman replacing. Finally the vector is added to the country's dataset.

SPAIN

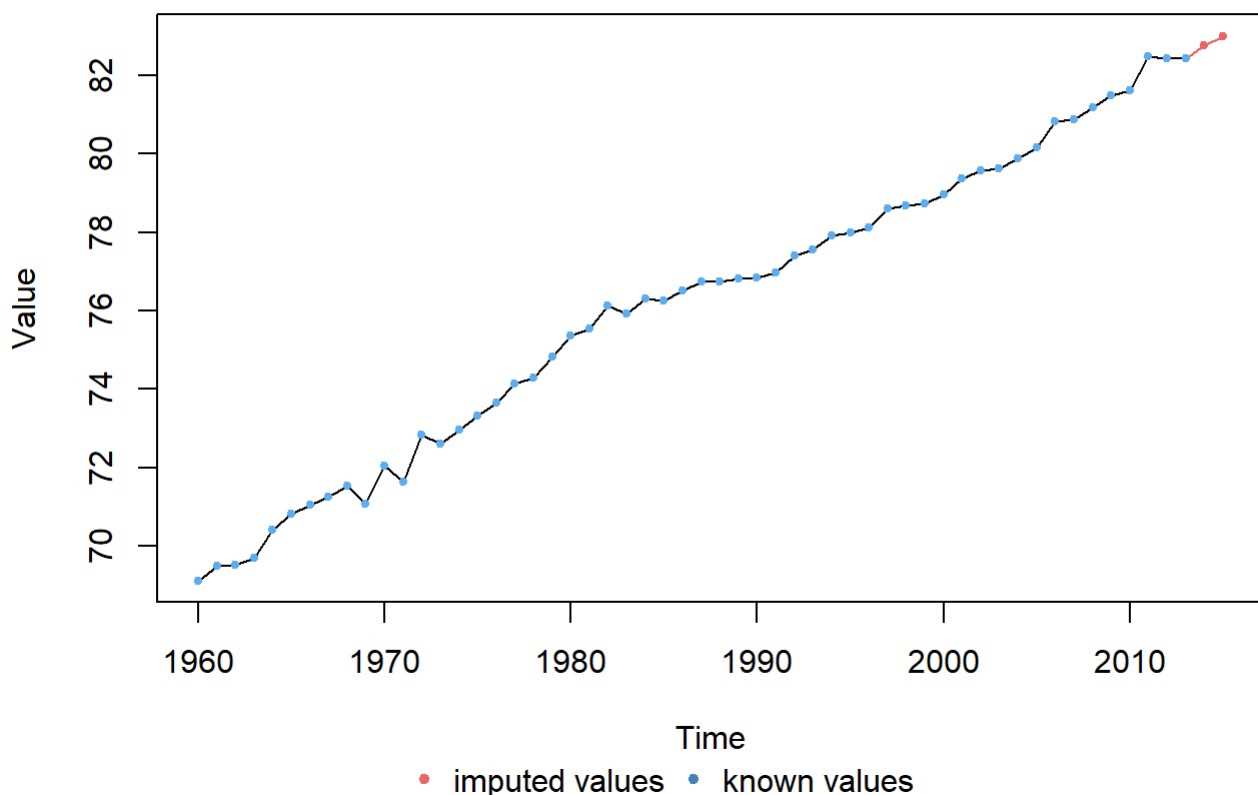
```

year <- sqldf('select distinct Year from world_indicators')
health_ESP <- data.frame(year)

#Life expectancy years
life_expectancy <- sqldf('select Year, Value from ESP_indicators where IndicatorName
== "Life expectancy at birth, total (years)"')
life_expectancy_na <- rep(NA, 56)
i <- 1
for (year in life_expectancy$Year){
  life_expectancy_na[(year-1960+1)] <- life_expectancy$Value[i]
  i = i + 1
}
life_expectancy_ts <- ts(data = life_expectancy_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(life_expectancy_ts)){
  life_expectancy_kalman <- na.kalman(life_expectancy_ts)
} else{
  life_expectancy_kalman <- life_expectancy_ts
}
health_ESP$'Life expectancy at birth, total (years)' <- life_expectancy_kalman
plotNA.imputations(life_expectancy_ts, life_expectancy_kalman, main = "Life expectanc
y years ESP")

```

Life expectancy years ESP

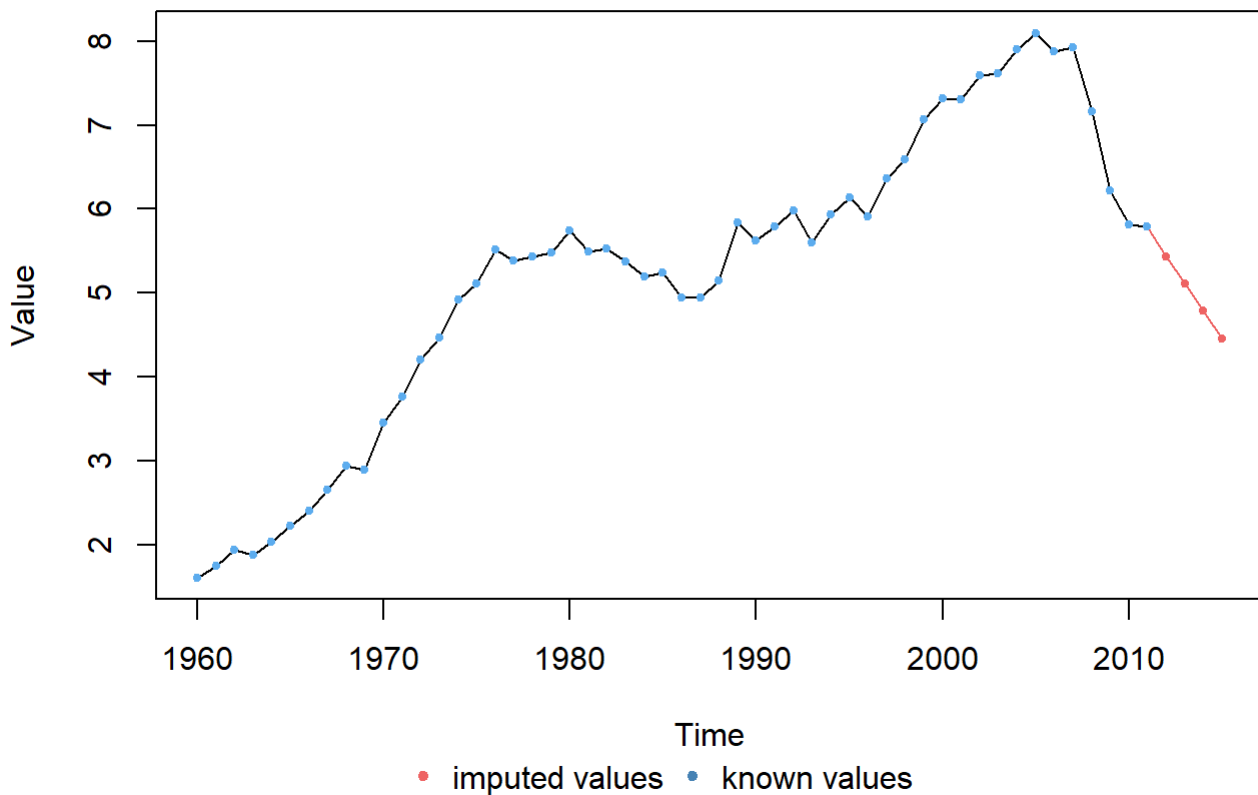


```

#CO2 emissions metrics tons per capita
CO2_emissions <- sqldf('select Year, Value from ESP_indicators where IndicatorName ==
  "CO2 emissions (metric tons per capita)"')
CO2_emissions_na <- rep(NA, 56)
i <- 1
for (year in CO2_emissions$Year){
  CO2_emissions_na[(year-1960+1)] <- CO2_emissions$Value[i]
  i = i + 1
}
CO2_emissions_ts <- ts(data = CO2_emissions_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(CO2_emissions_ts)){
  CO2_emissions_kalman <- na.kalman(CO2_emissions_ts)
} else{
  CO2_emissions_kalman <- CO2_emissions_ts
}
health_ESP$'CO2 emissions (metric tons per capita)' <- CO2_emissions_kalman
plotNA.imputations(CO2_emissions_ts, CO2_emissions_kalman, main = "CO2 emissions ESP"
)

```

CO2 emissions ESP

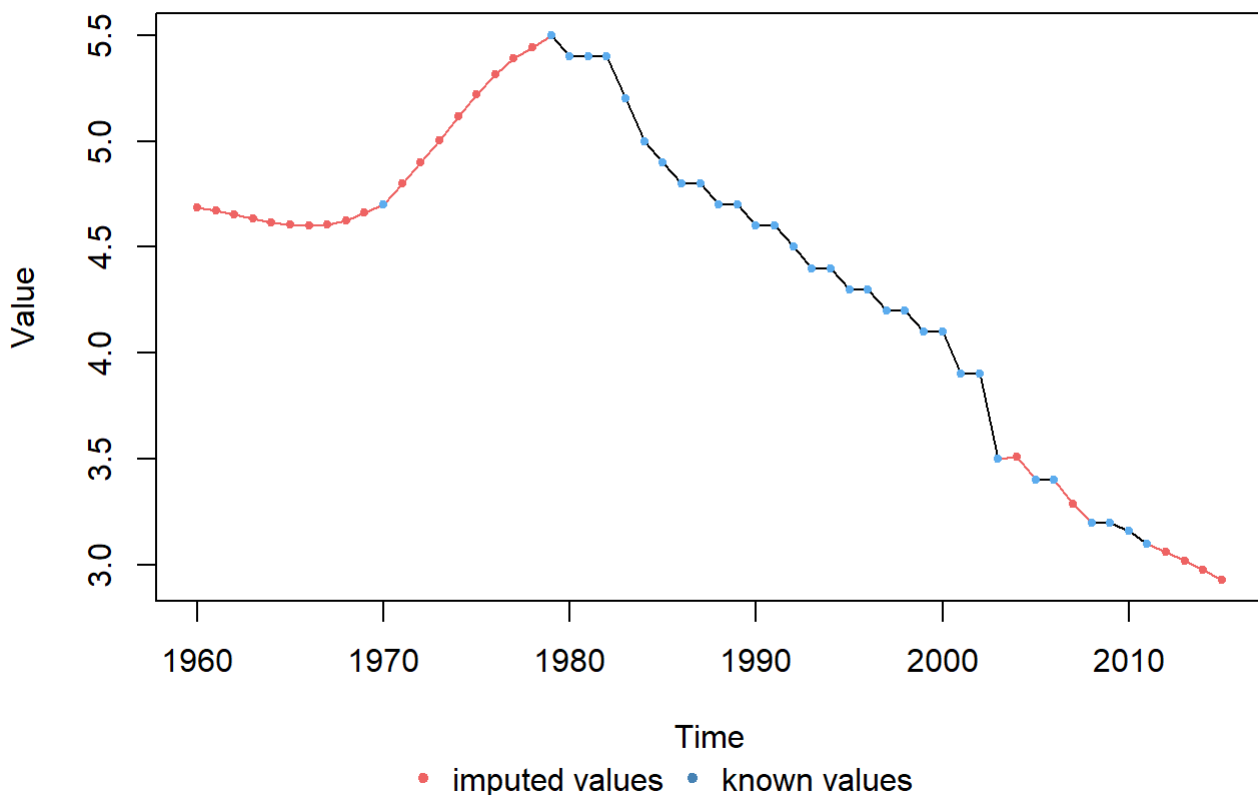


```

#Hospital beds per 1000 people
hospital_beds <- sqldf('select Year, Value from ESP_indicators where IndicatorName ==
  "Hospital beds (per 1,000 people)"')
hospital_beds_na <- rep(NA, 56)
i <- 1
for (year in hospital_beds$Year){
  hospital_beds_na[(year-1960+1)] <- hospital_beds$Value[i]
  i = i + 1
}
hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(hospital_beds_ts)){
  hospital_beds_kalman <- na.kalman(hospital_beds_ts)
} else{
  hospital_beds_kalman <- hospital_beds_ts
}
health_ESP$'Hospital beds (per 1,000 people)' <- hospital_beds_kalman
plotNA.imputations(hospital_beds_ts, hospital_beds_kalman, main = "Hospital beds ESP"
)

```

Hospital beds ESP

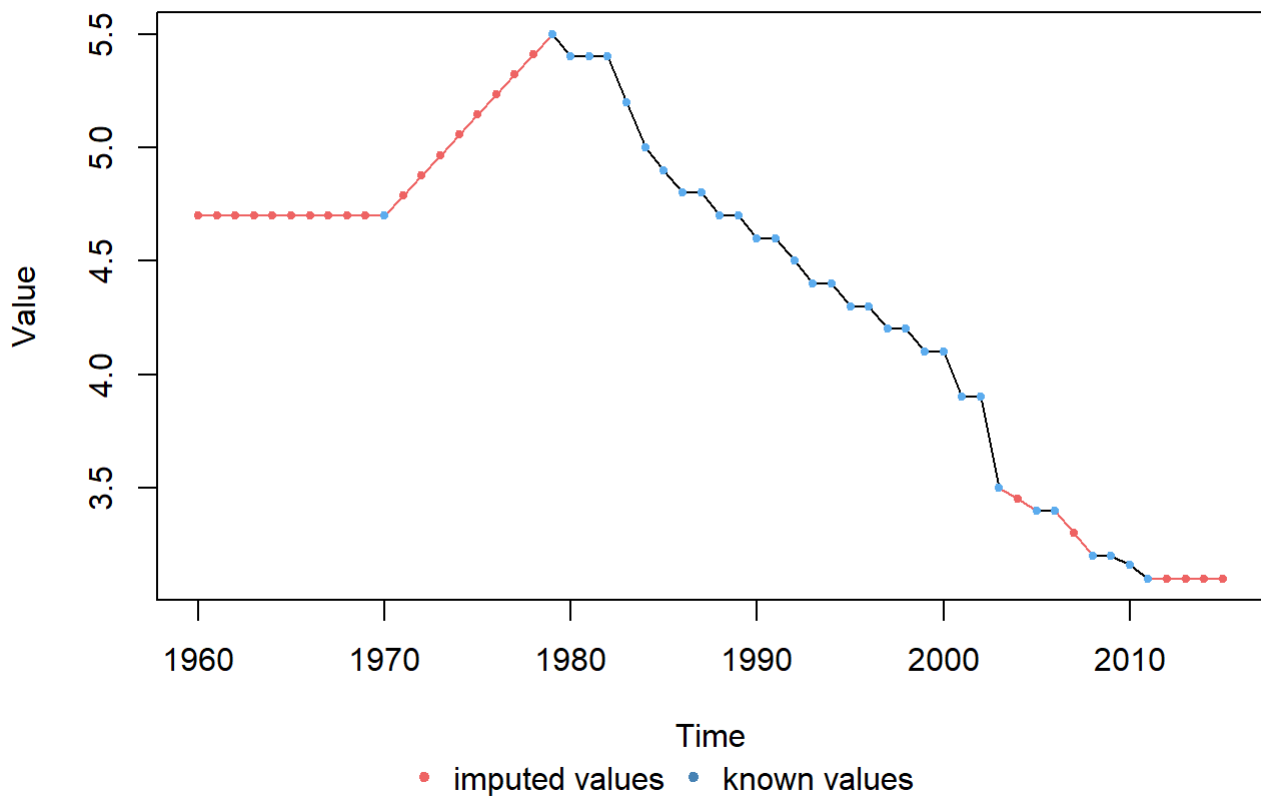


```

#para el report y presentacion enseñar que hay varias formas de completar los datos
hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency =
  1)
hospital_beds_linear <- na.interpolation(hospital_beds_ts)
hospital_beds_spline <- na.interpolation(hospital_beds_ts, option = "spline")
hospital_beds_kalman <- na.kalman(hospital_beds_ts)
plotNA.imputations(hospital_beds_ts, hospital_beds_linear, main = "linear")

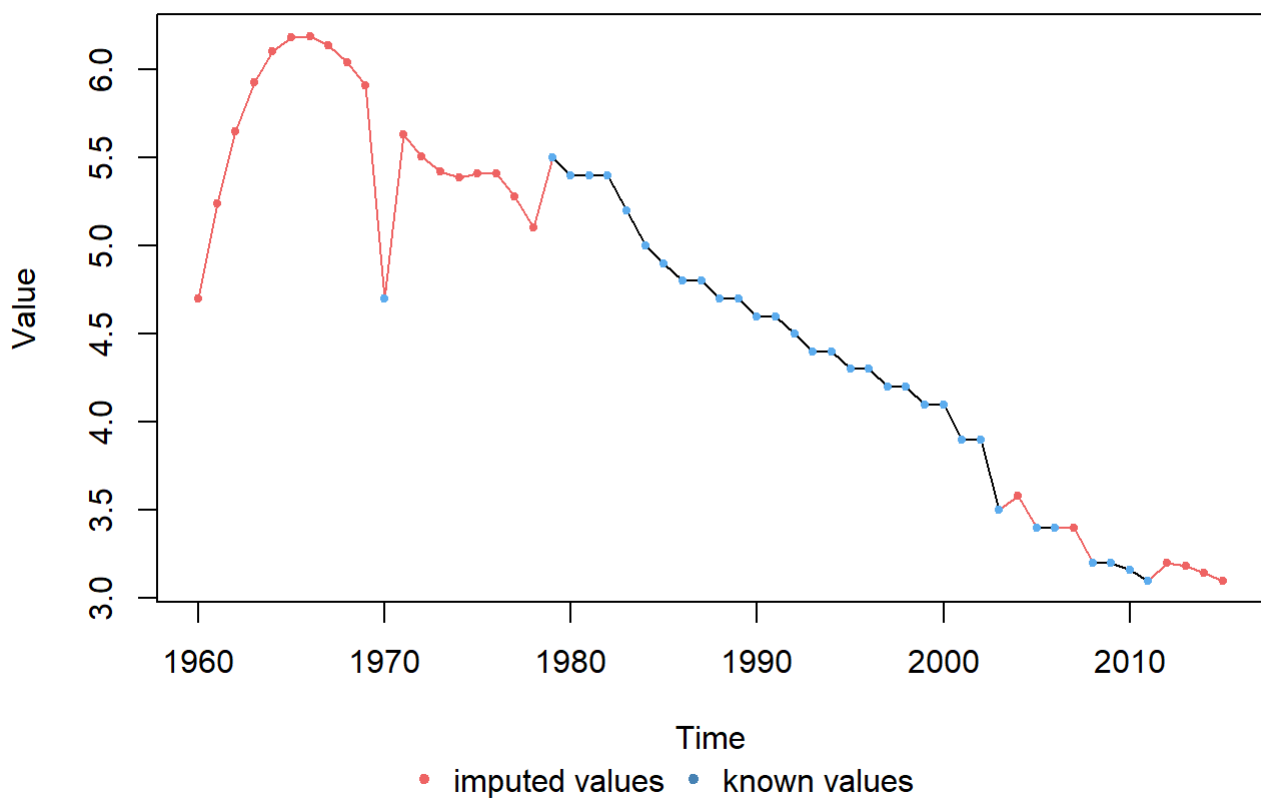
```

linear

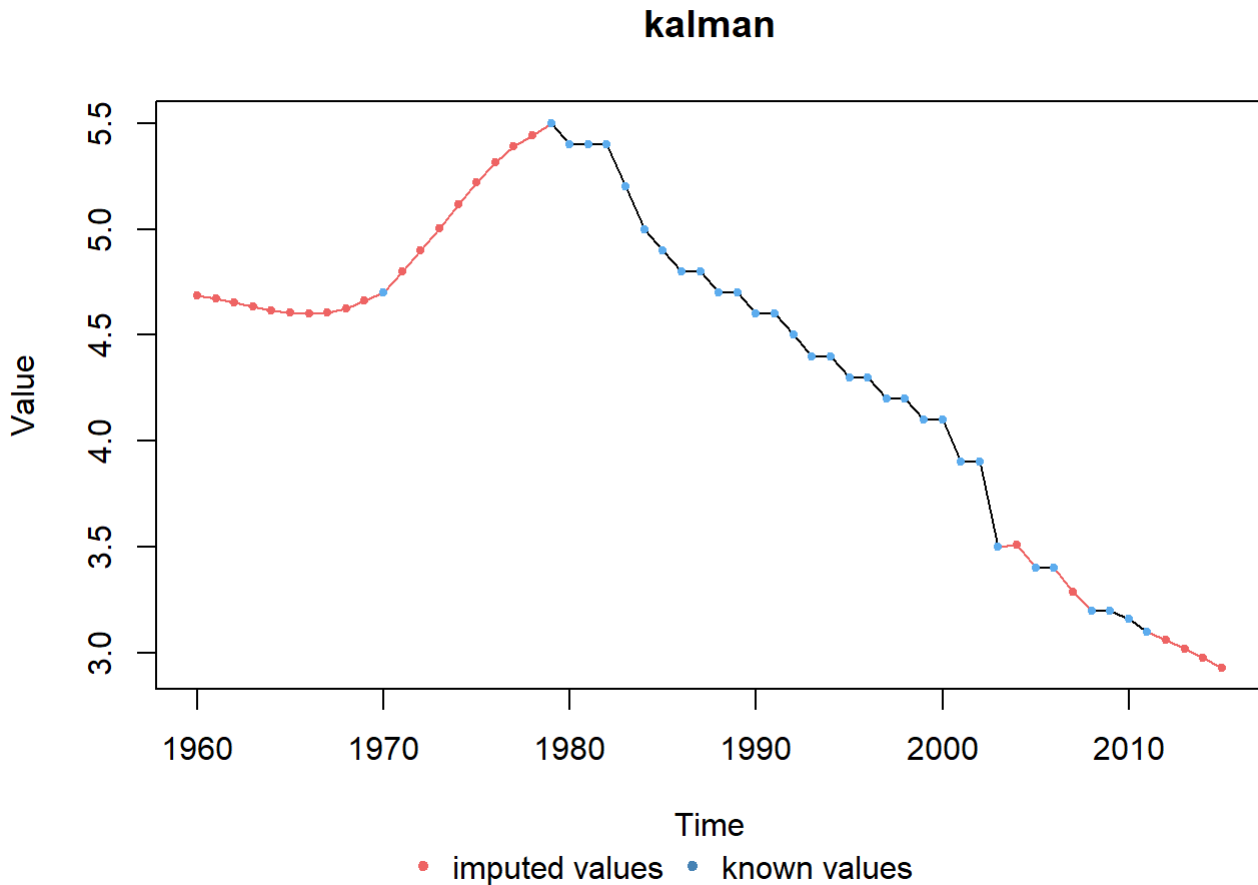


```
plotNA.imputations(hospital_beds_ts, hospital_beds_spline, main = "spline")
```

spline

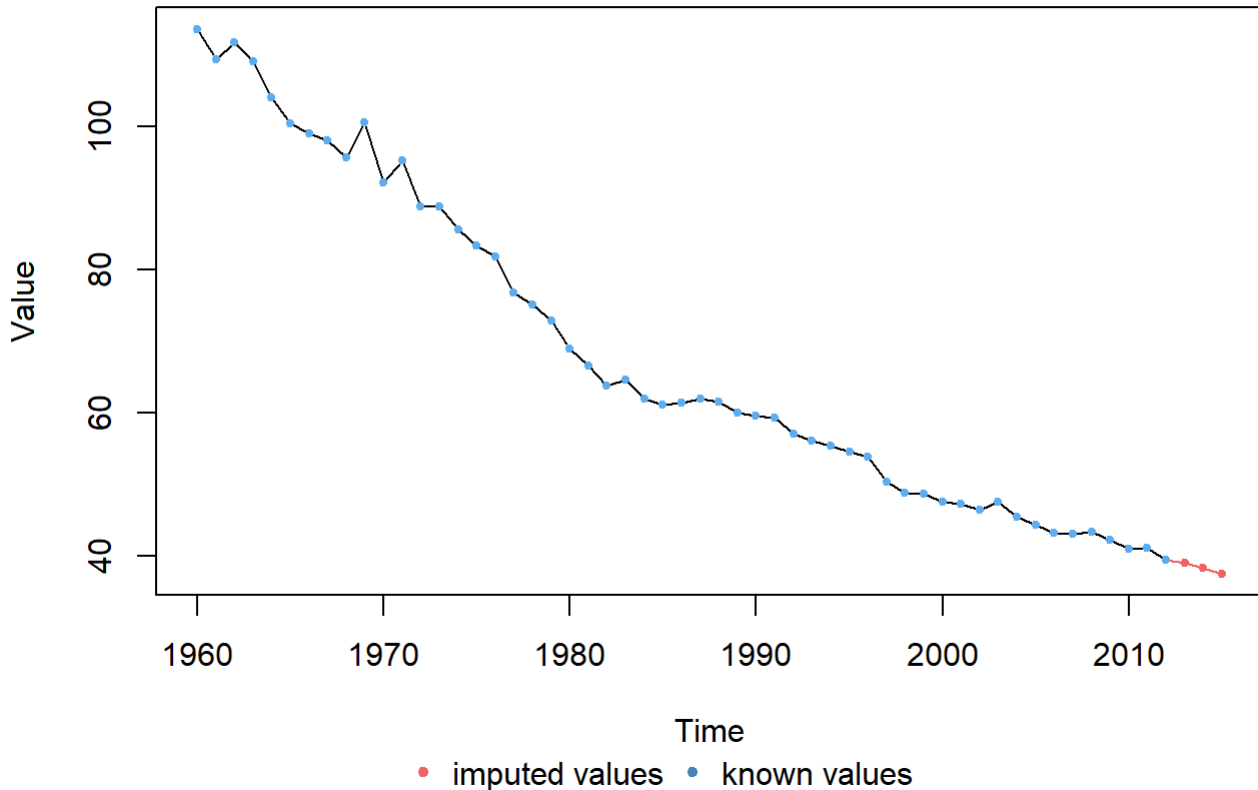


```
plotNA.imputations(hospital_beds_ts, hospital_beds_kalman, main = "kalman")
```



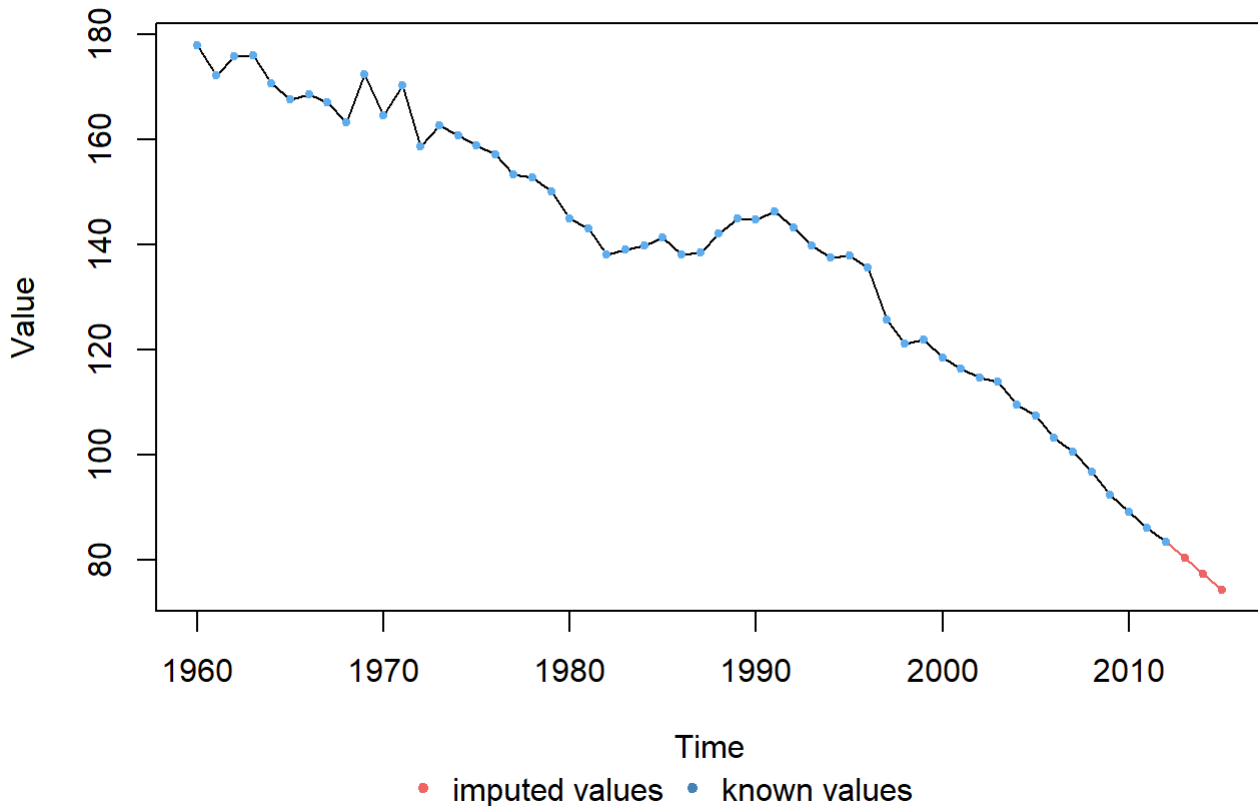
```
#Female mortality per 1000 females
female_mortality <- sqldf('select Year, Value from ESP_indicators where IndicatorName
== "Mortality rate, adult, female (per 1,000 female adults)"')
female_mortality_na <- rep(NA, 56)
i <- 1
for (year in female_mortality$Year){
  female_mortality_na[(year-1960+1)] <- female_mortality$Value[i]
  i = i + 1
}
female_mortality_ts <- ts(data = female_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(female_mortality_ts)){
  female_mortality_kalman <- na.kalman(female_mortality_ts)
} else{
  female_mortality_kalman <- female_mortality_ts
}
health_ESP$'Mortality rate, adult, female (per 1,000 female adults)' <- female_mortal
ity_kalman
plotNA.imputations(female_mortality_ts, female_mortality_kalman, main = "Female morta
lity ESP")
```


Female mortality ESP



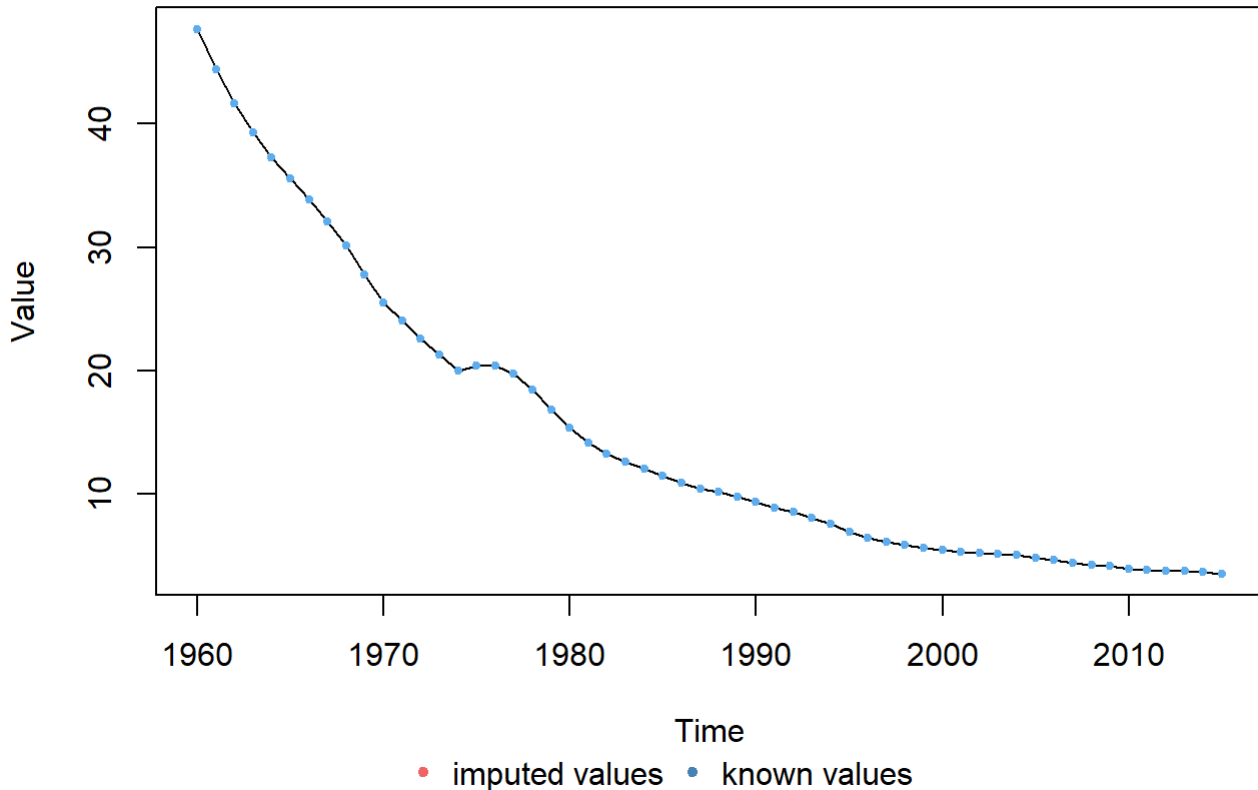
```
#Male mortality per 1000 males
male_mortality <- sqldf('select Year, Value from ESP_indicators where IndicatorName =
= "Mortality rate, adult, male (per 1,000 male adults)"')
male_mortality_na <- rep(NA, 56)
i <- 1
for (year in male_mortality$Year){
  male_mortality_na[(year-1960+1)] <- male_mortality$Value[i]
  i = i + 1
}
male_mortality_ts <- ts(data = male_mortality_na, start = 1960, end = 2015, frequency
= 1)
if(anyNA(male_mortality_ts)){
  male_mortality_kalman <- na.kalman(male_mortality_ts)
} else{
  male_mortality_kalman <- male_mortality_ts
}
health_ESP$'Mortality rate, adult, male (per 1,000 male adults)' <- male_mortality_kalman
plotNA.imputations(male_mortality_ts, male_mortality_kalman, main = "Male mortality ESP")
```

Male mortality ESP



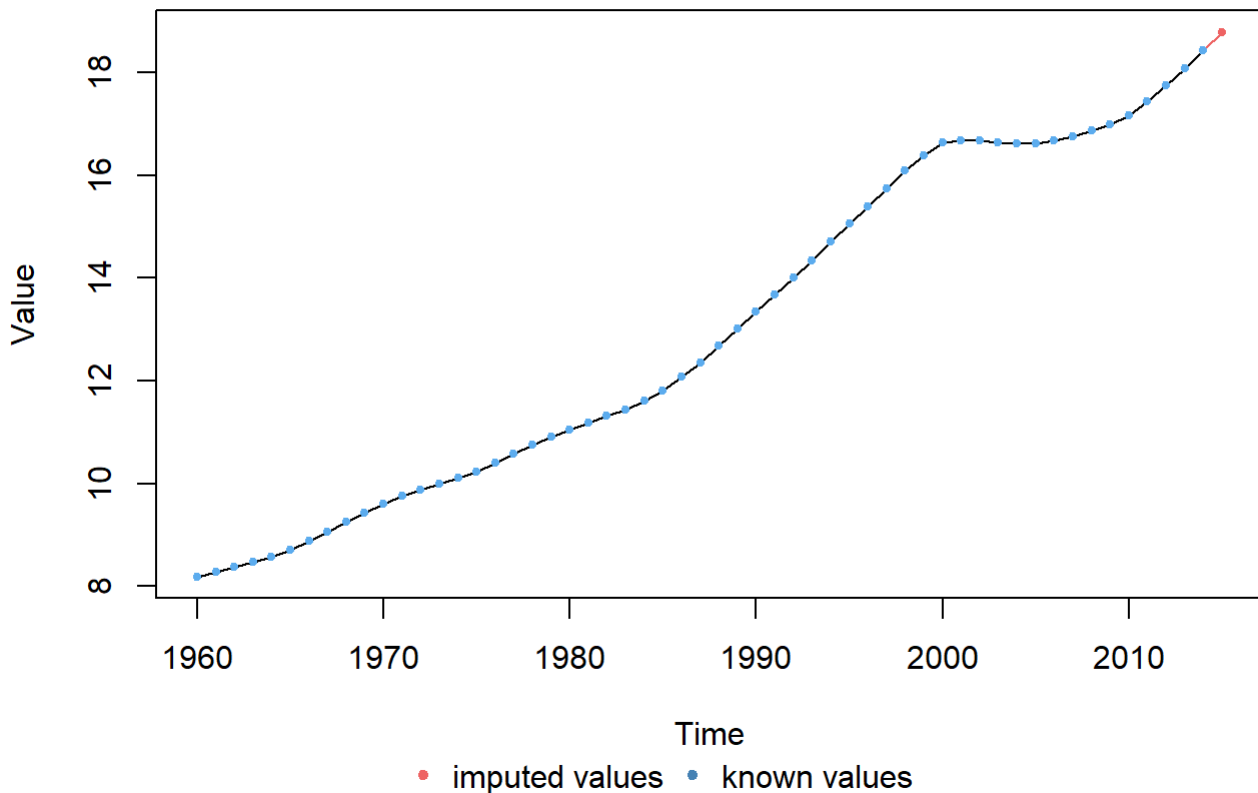
```
#Infant mortality per 1000 live births
infant_mortality <- sqldf('select Year, Value from ESP_indicators where IndicatorName
== "Mortality rate, infant (per 1,000 live births)"')
infant_mortality_na <- rep(NA, 56)
i <- 1
for (year in infant_mortality$Year){
  infant_mortality_na[(year-1960+1)] <- infant_mortality$Value[i]
  i = i + 1
}
infant_mortality_ts <- ts(data = infant_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(infant_mortality_ts)){
  infant_mortality_kalman <- na.kalman(infant_mortality_ts)
} else{
  infant_mortality_kalman <- infant_mortality_ts
}
health_ESP$'Mortality rate, infant (per 1,000 live births)' <- infant_mortality_kalma
n
plotNA.imputations(infant_mortality_ts, infant_mortality_kalman, main = "Infant morta
lity ESP")
```

Infant mortality ESP



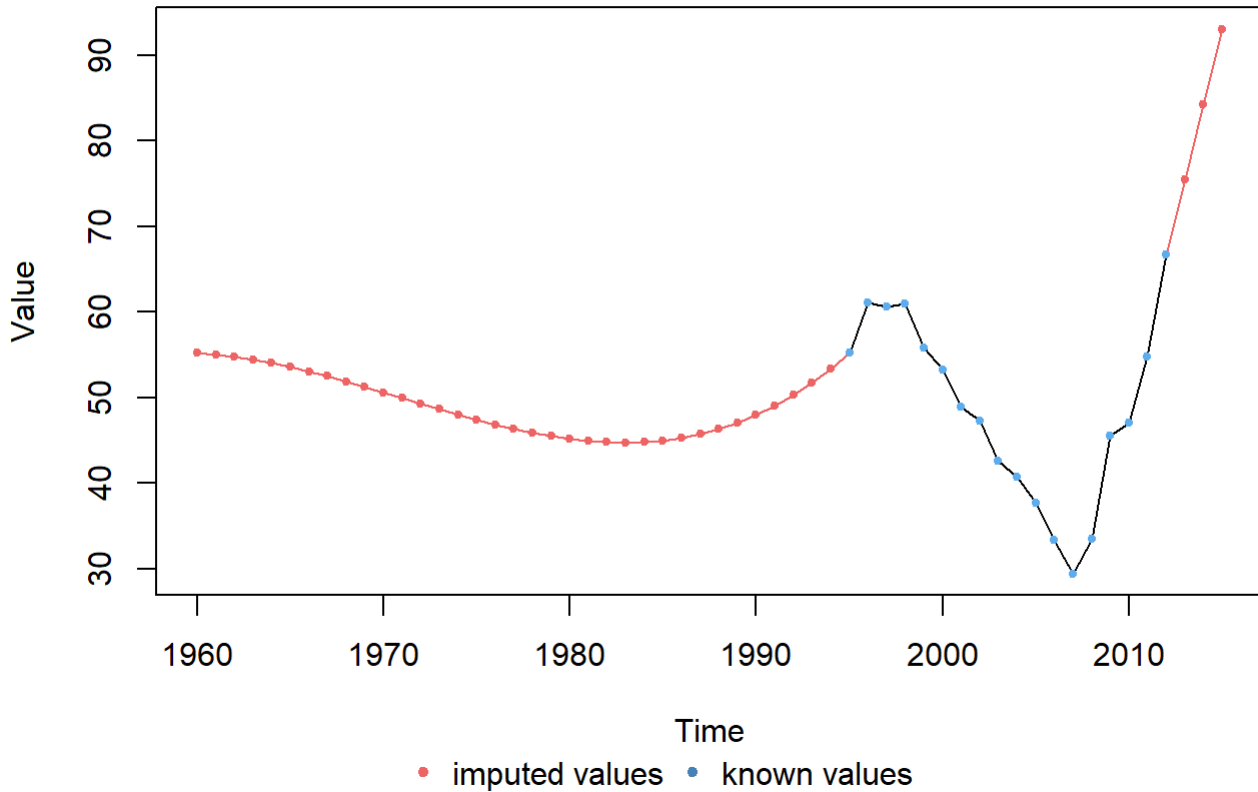
```
#Population over 65
population_over65 <- sqldf('select Year, Value from ESP_indicators where IndicatorName == "Population ages 65 and above (% of total)"')
population_over65_na <- rep(NA, 56)
i <- 1
for (year in population_over65$Year){
  population_over65_na[(year-1960+1)] <- population_over65$Value[i]
  i = i + 1
}
population_over65_ts <- ts(data = population_over65_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(population_over65_ts)){
  population_over65_kalman <- na.kalman(population_over65_ts)
} else{
  population_over65_kalman <- population_over65_ts
}
health_ESP$'Population ages 65 and above (% of total)' <- population_over65_kalman
plotNA.imputations(population_over65_ts, population_over65_kalman, main = "Population over 65 ESP")
```

Population over 65 ESP



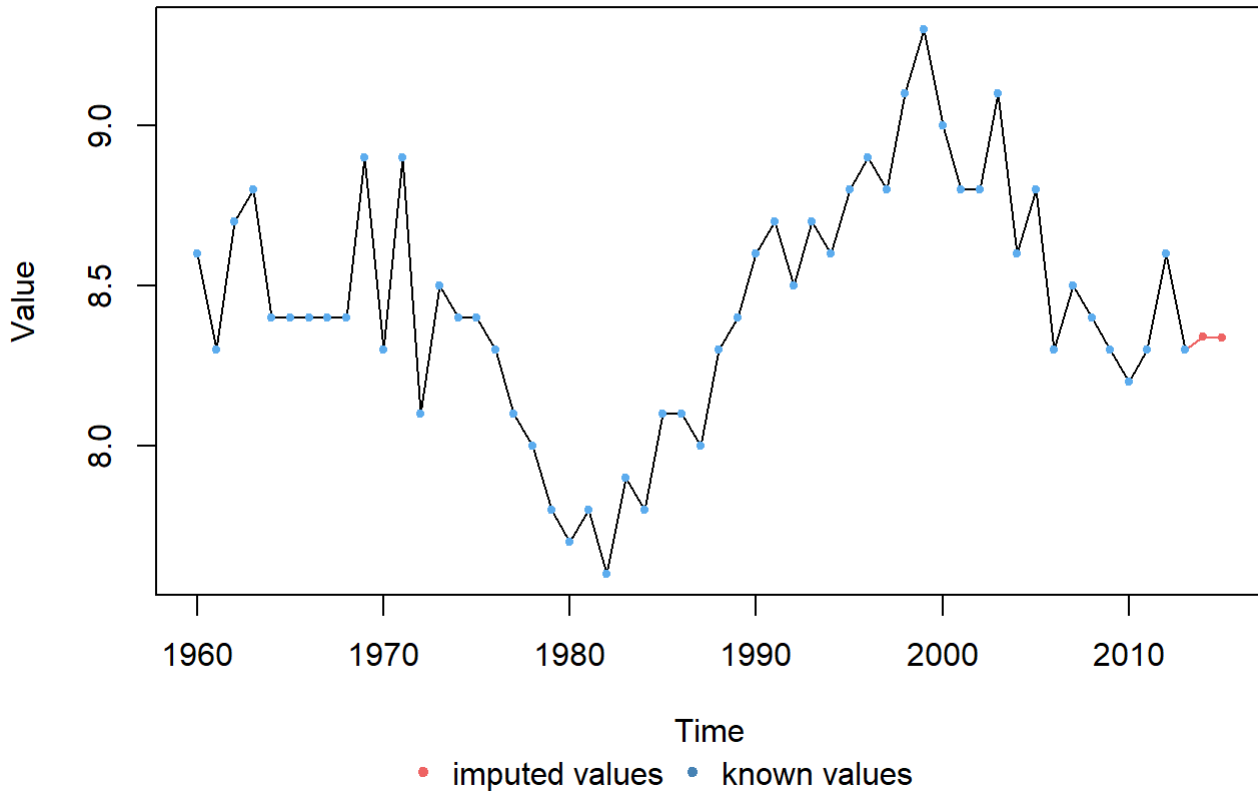
```
#Government debt, total % of GDP
government_debt <- sqldf('select Year, Value from ESP_indicators where IndicatorName
== "Central government debt, total (% of GDP)"')
government_debt_na <- rep(NA, 56)
i <- 1
for (year in government_debt$Year){
  government_debt_na[(year-1960+1)] <- government_debt$Value[i]
  i = i + 1
}
government_debt_ts <- ts(data = government_debt_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(government_debt_ts)){
  government_debt_kalman <- na.kalman(government_debt_ts)
} else{
  government_debt_kalman <- government_debt_ts
}
health_ESP$'Central government debt, total (% of GDP)' <- government_debt_kalman
plotNA.imputations(government_debt_ts, government_debt_kalman, main = "Government deb
t % of GDP ESP")
```

Government debt % of GDP ESP



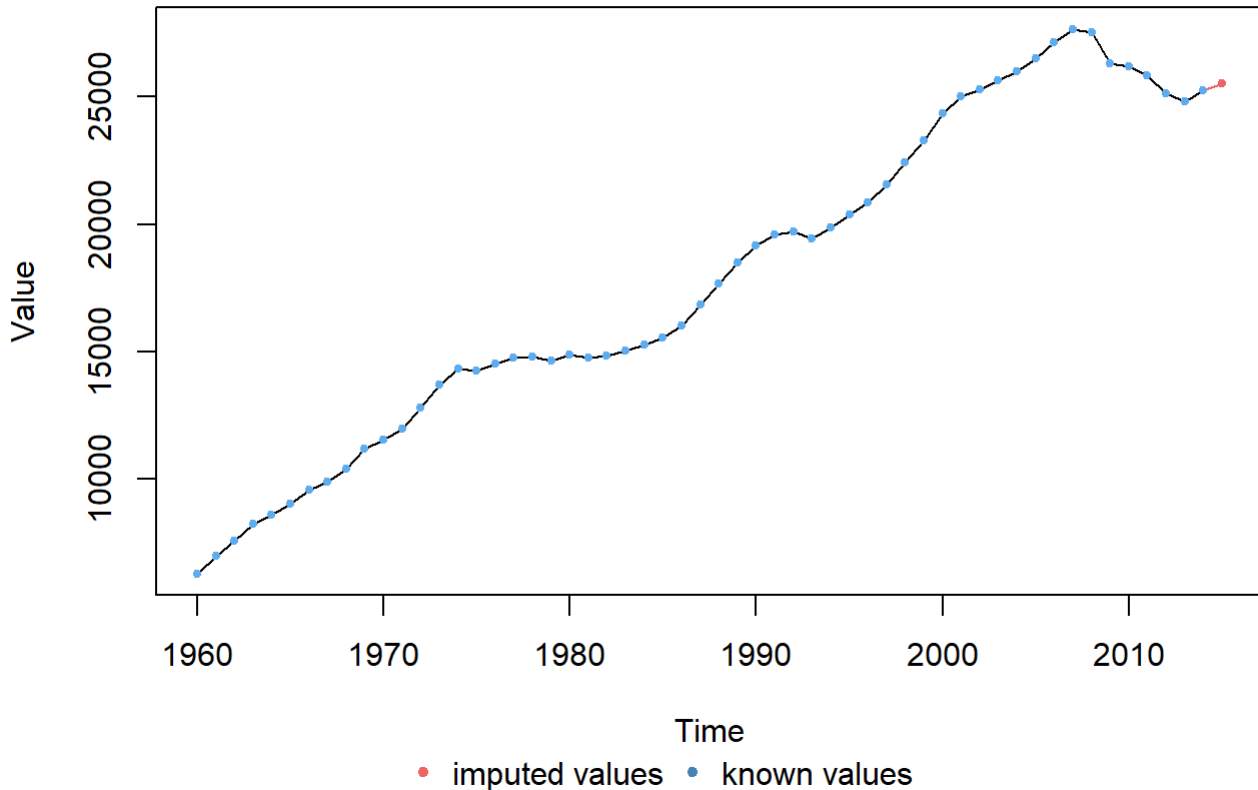
```
#Death rate per 1000 people
death_rate <- sqldf('select Year, Value from ESP_indicators where IndicatorName == "D
eath rate, crude (per 1,000 people)"')
death_rate_na <- rep(NA, 56)
i <- 1
for (year in death_rate$Year){
  death_rate_na[(year-1960+1)] <- death_rate$Value[i]
  i = i + 1
}
death_rate_ts <- ts(data = death_rate_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(death_rate_ts)){
  death_rate_kalman <- na.kalman(death_rate_ts)
} else{
  death_rate_kalman <- death_rate_ts
}
health_ESP$'Death rate, crude (per 1,000 people)' <- death_rate_kalman
plotNA.imputations(death_rate_ts, death_rate_kalman, main = "Death rate ESP")
```

Death rate ESP



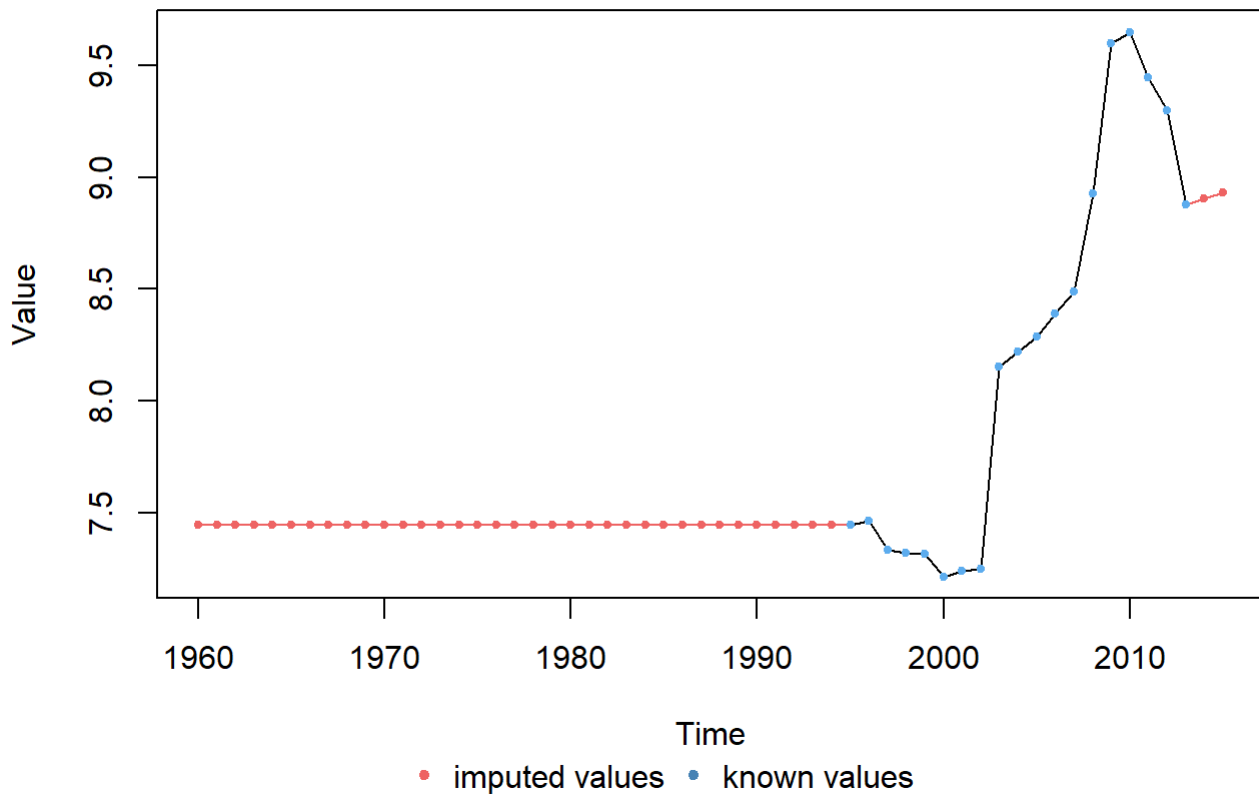
```
#GDP per capita (constant 2005 US$)
GDP <- sqldf('select Year, Value from ESP_indicators where IndicatorName == "GDP per
capita (constant 2005 US$)"')
GDP_na <- rep(NA, 56)
i <- 1
for (year in GDP$Year){
  GDP_na[(year-1960+1)] <- GDP$Value[i]
  i = i + 1
}
GDP_ts <- ts(data = GDP_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(GDP_ts)){
  GDP_kalman <- na.kalman(GDP_ts)
} else{
  GDP_kalman <- GDP_ts
}
health_ESP$'GDP per capita (constant 2005 US$)' <- GDP_kalman
plotNA.imputations(GDP_ts, GDP_kalman, main = "GDP per capita (constant 2005 US$)")
```

GDP per capita (constant 2005 US\$)



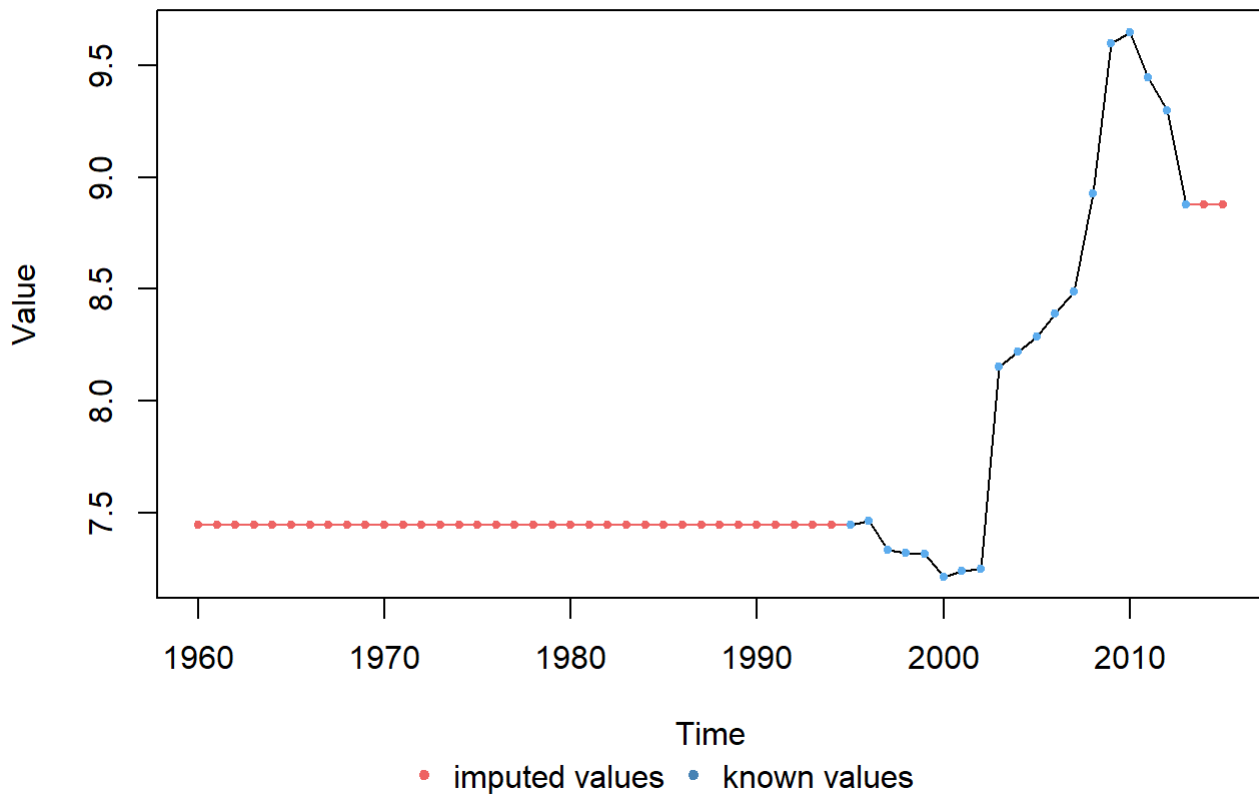
```
#Health expenditure, total % of GDP
health_expenditure <- sqldf('select Year, Value from ESP_indicators where IndicatorName == "Health expenditure, total (% of GDP)"')
health_expenditure_na <- rep(NA, 56)
i <- 1
for (year in health_expenditure$Year){
  health_expenditure_na[(year-1960+1)] <- health_expenditure$Value[i]
  i = i + 1
}
health_expenditure_ts <- ts(data = health_expenditure_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(health_expenditure_ts)){
  health_expenditure_kalman <- na.kalman(health_expenditure_ts)
} else{
  health_expenditure_kalman <- health_expenditure_ts
}
health_ESP$'Health expenditure, total (% of GDP)' <- health_expenditure_kalman
plotNA.imputations(health_expenditure_ts, health_expenditure_kalman, main = "Health expenditure % of GDP ESP")
```

Health expenditure % of GDP ESP



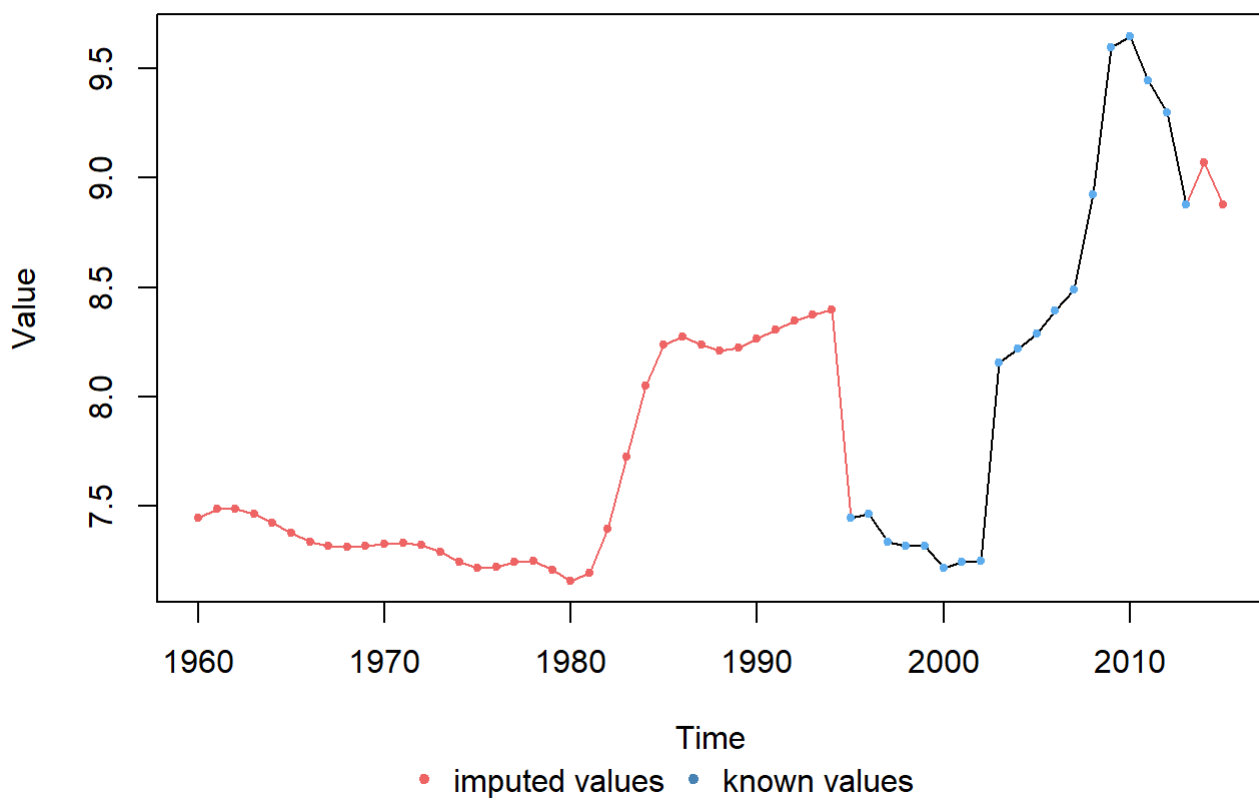
```
health_expenditure_ts <- ts(data = health_expenditure_na, start = 1960, end = 2015, frequency = 1)
health_expenditure_linear <- na.interpolation(health_expenditure_ts)
health_expenditure_spline <- na.interpolation(health_expenditure_ts, option = "spline")
health_expenditure_kalman <- na.kalman(health_expenditure_ts)
plotNA.imputations(health_expenditure_ts, health_expenditure_linear, main = "linear")
```


linear

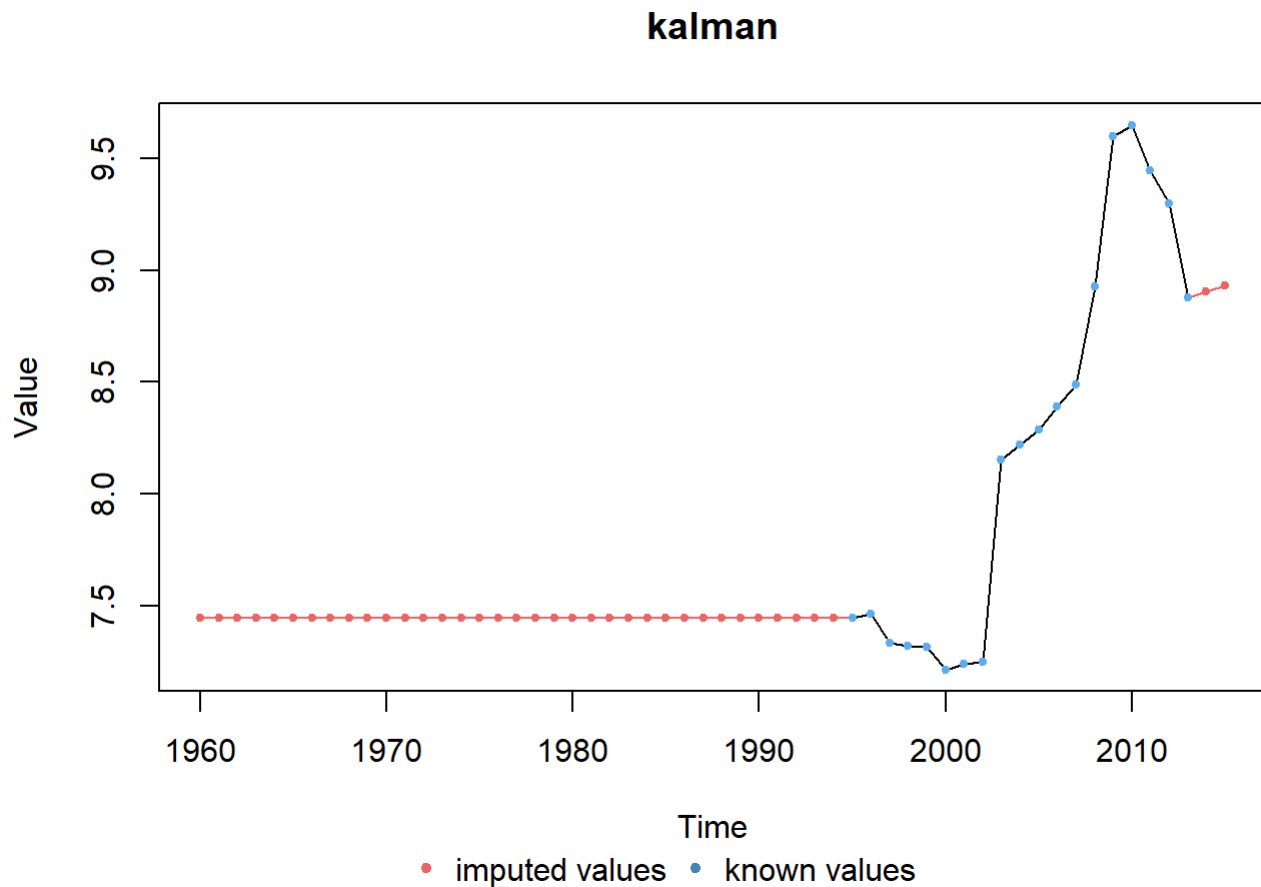


```
plotNA.imputations(health_expenditure_ts, health_expenditure_spline, main = "spline")
```

spline



```
plotNA.imputations(health_expenditure_ts, health_expenditure_kalman, main = "kalman")
```



There are some indicators selected to use as predictors for life expectancy that have too many NA values to impute to be useful since we are imputing even more values that we had and there are not any values in a long period to have some reference values for the imputations. Therefore, Governemnt debt(% of GDP), Health expenditure(% of GDP) and Hospital beds are discarded to use as predictors. We erase these predictor from the dataset we are going to use.

```
health_ESP <- health_ESP[, -c(4, 9, 12)]
```

CHINA

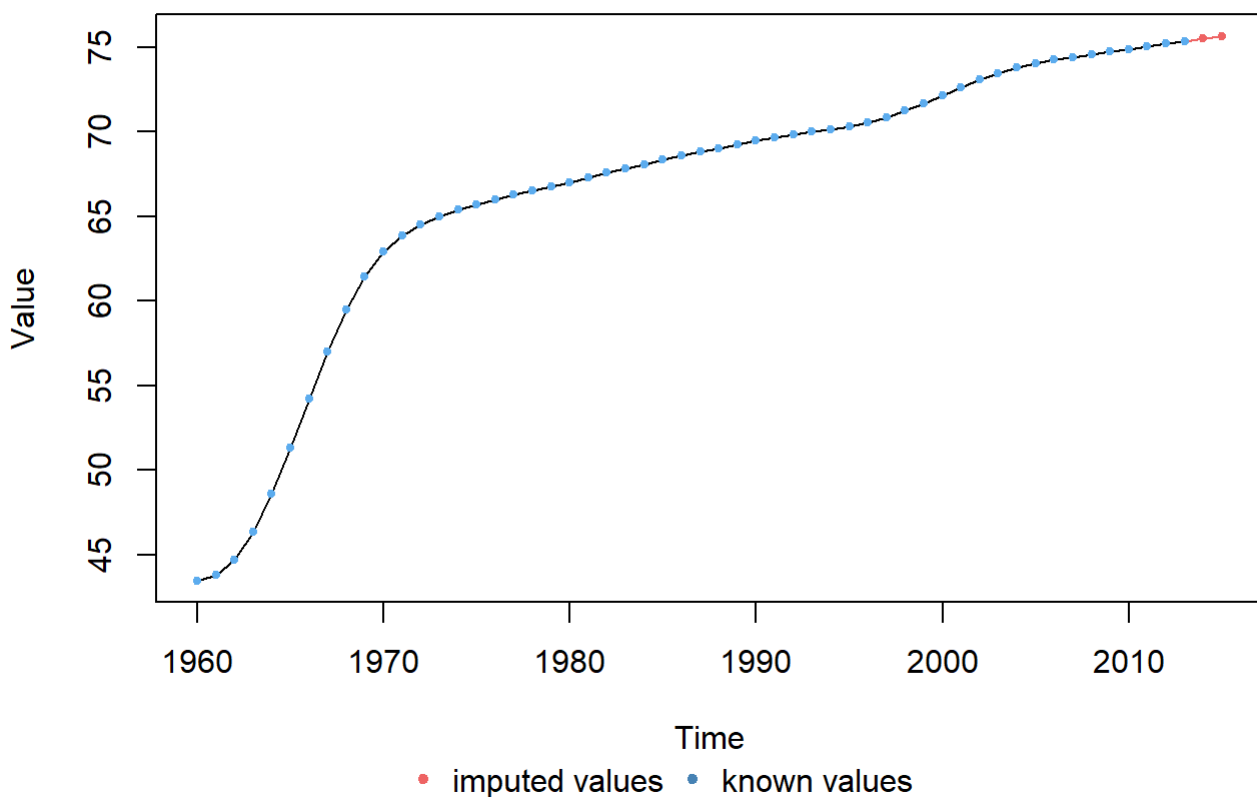
```

year <- sqldf('select distinct Year from world_indicators')
health_CHN <- data.frame(year)

#Life expectancy years
life_expectancy <- sqldf('select Year, Value from CHN_indicators where IndicatorName
== "Life expectancy at birth, total (years)"')
life_expectancy_na <- rep(NA, 56)
i <- 1
for (year in life_expectancy$Year){
  life_expectancy_na[(year-1960+1)] <- life_expectancy$Value[i]
  i = i + 1
}
life_expectancy_ts <- ts(data = life_expectancy_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(life_expectancy_ts)){
  life_expectancy_kalman <- na.kalman(life_expectancy_ts)
} else{
  life_expectancy_kalman <- life_expectancy_ts
}
health_CHN$'Life expectancy at birth, total (years)' <- life_expectancy_kalman
plotNA.imputations(life_expectancy_ts, life_expectancy_kalman, main = "Life expectanc
y years CHN")

```

Life expectancy years CHN

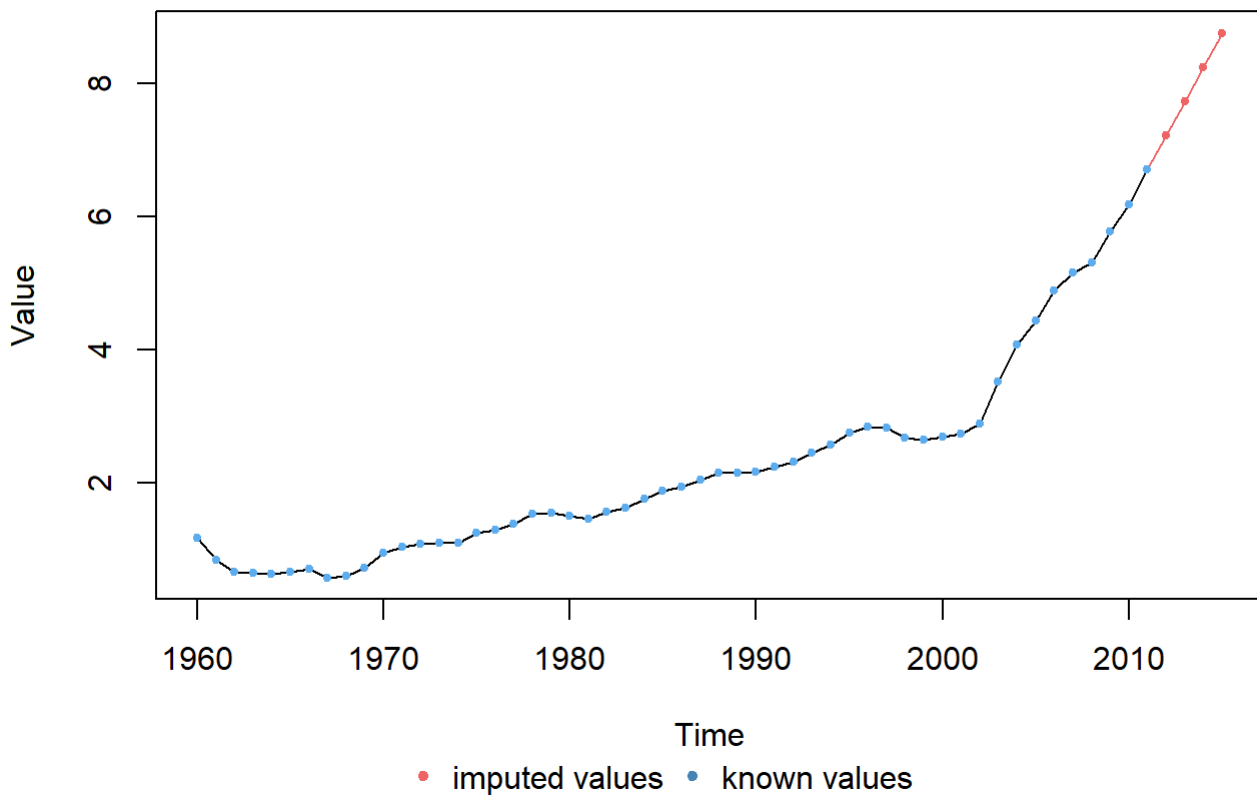


```

#CO2 emissions metrics tons per capita
CO2_emissions <- sqldf('select Year, Value from CHN_indicators where IndicatorName ==
  "CO2 emissions (metric tons per capita)"')
CO2_emissions_na <- rep(NA, 56)
i <- 1
for (year in CO2_emissions$Year){
  CO2_emissions_na[(year-1960+1)] <- CO2_emissions$Value[i]
  i = i + 1
}
CO2_emissions_ts <- ts(data = CO2_emissions_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(CO2_emissions_ts)){
  CO2_emissions_kalman <- na.kalman(CO2_emissions_ts)
} else{
  CO2_emissions_kalman <- CO2_emissions_ts
}
health_CHN$'CO2 emissions (metric tons per capita)' <- CO2_emissions_kalman
plotNA.imputations(CO2_emissions_ts, CO2_emissions_kalman, main = "CO2 emissions CHN"
)

```

CO2 emissions CHN

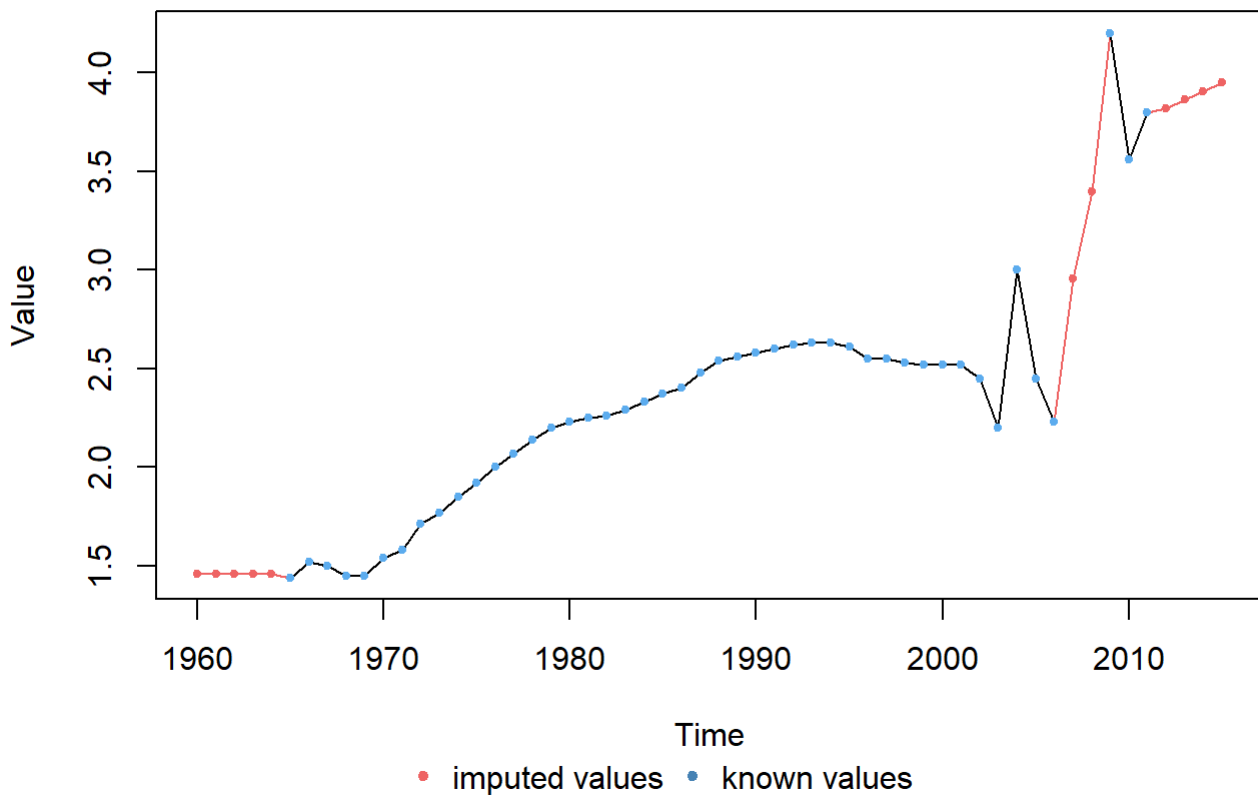


```

#Hospital beds per 1000 people
hospital_beds <- sqldf('select Year, Value from CHN_indicators where IndicatorName ==
  "Hospital beds (per 1,000 people)"')
hospital_beds_na <- rep(NA, 56)
i <- 1
for (year in hospital_beds$Year){
  hospital_beds_na[(year-1960+1)] <- hospital_beds$Value[i]
  i = i + 1
}
hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(hospital_beds_ts)){
  hospital_beds_kalman <- na.kalman(hospital_beds_ts)
} else{
  hospital_beds_kalman <- hospital_beds_ts
}
health_CHN$'Hospital beds (per 1,000 people)' <- hospital_beds_kalman
plotNA.imputations(hospital_beds_ts, hospital_beds_kalman, main = "Hospital beds CHN"
)

```

Hospital beds CHN



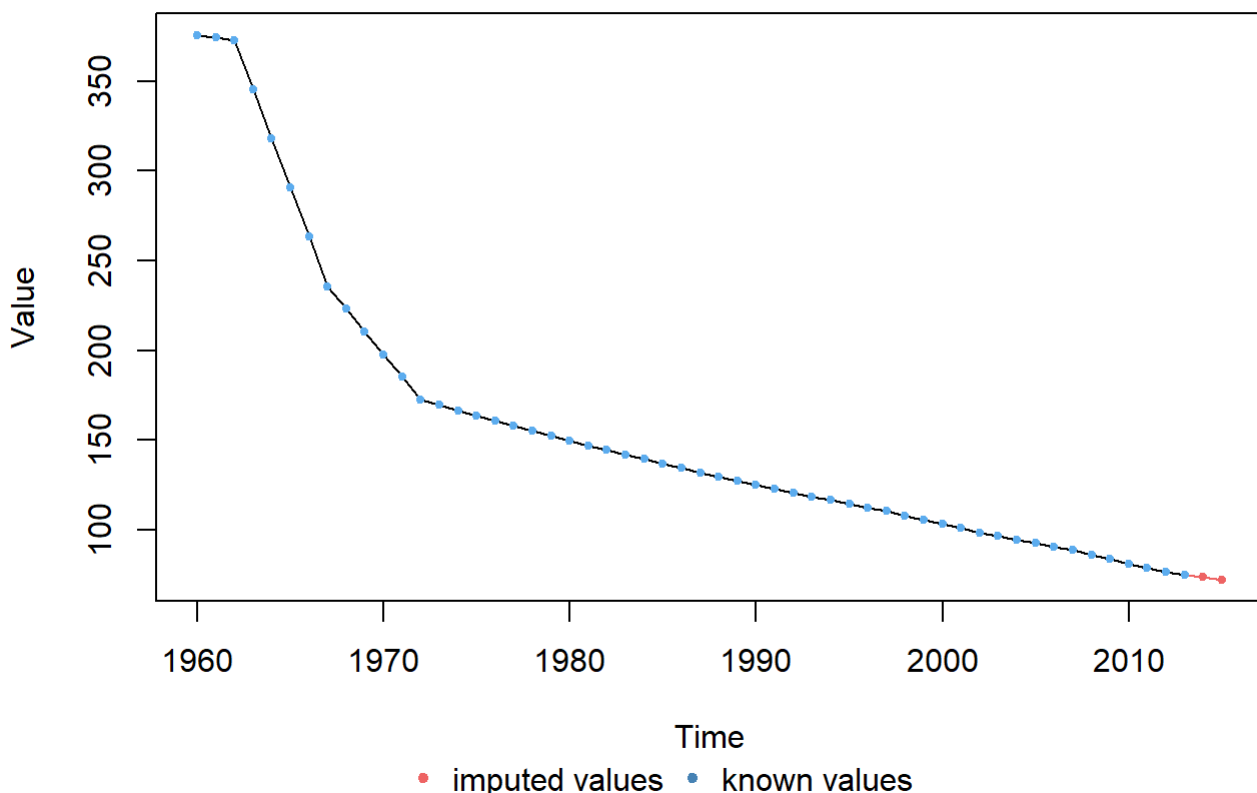
```

# hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency
= 1)
# hospital_beds_linear <- na.interpolation(hospital_beds_ts)
# hospital_beds_spline <- na.interpolation(hospital_beds_ts, option = "spline")
# hospital_beds_kalman <- na.kalman(hospital_beds_ts)
# plotNA.imputations(hospital_beds_ts, hospital_beds_linear)
# plotNA.imputations(hospital_beds_ts, hospital_beds_spline)
# plotNA.imputations(hospital_beds_ts, hospital_beds_kalman)

#Female mortality per 1000 females
female_mortality <- sqldf('select Year, Value from CHN_indicators where IndicatorName
== "Mortality rate, adult, female (per 1,000 female adults)"')
female_mortality_na <- rep(NA, 56)
i <- 1
for (year in female_mortality$Year){
  female_mortality_na[(year-1960+1)] <- female_mortality$Value[i]
  i = i + 1
}
female_mortality_ts <- ts(data = female_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(female_mortality_ts)){
  female_mortality_kalman <- na.kalman(female_mortality_ts)
} else{
  female_mortality_kalman <- female_mortality_ts
}
health_CHN$'Mortality rate, adult, female (per 1,000 female adults)' <- female_mortal
ity_kalman
plotNA.imputations(female_mortality_ts, female_mortality_kalman, main = "Female morta
lity CHN")

```

Female mortality CHN

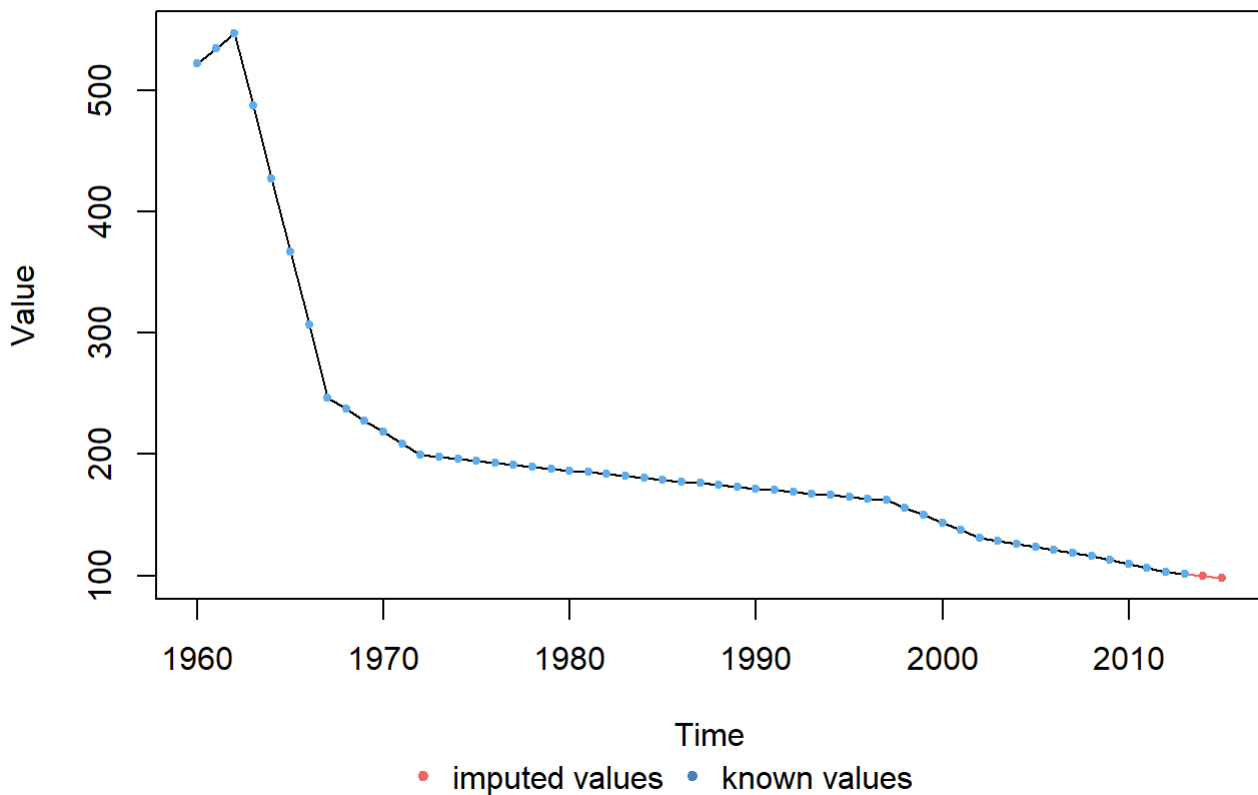


```

#Male mortality per 1000 males
male_mortality <- sqldf('select Year, Value from CHN_indicators where IndicatorName =
= "Mortality rate, adult, male (per 1,000 male adults)"')
male_mortality_na <- rep(NA, 56)
i <- 1
for (year in male_mortality$Year){
  male_mortality_na[(year-1960+1)] <- male_mortality$Value[i]
  i = i + 1
}
male_mortality_ts <- ts(data = male_mortality_na, start = 1960, end = 2015, frequency
= 1)
if(anyNA(male_mortality_ts)){
  male_mortality_kalman <- na.kalman(male_mortality_ts)
} else{
  male_mortality_kalman <- male_mortality_ts
}
health_CHN$'Mortality rate, adult, male (per 1,000 male adults)' <- male_mortality_kalman
plotNA.imputations(male_mortality_ts, male_mortality_kalman, main = "Male mortality C
HN")

```

Male mortality CHN

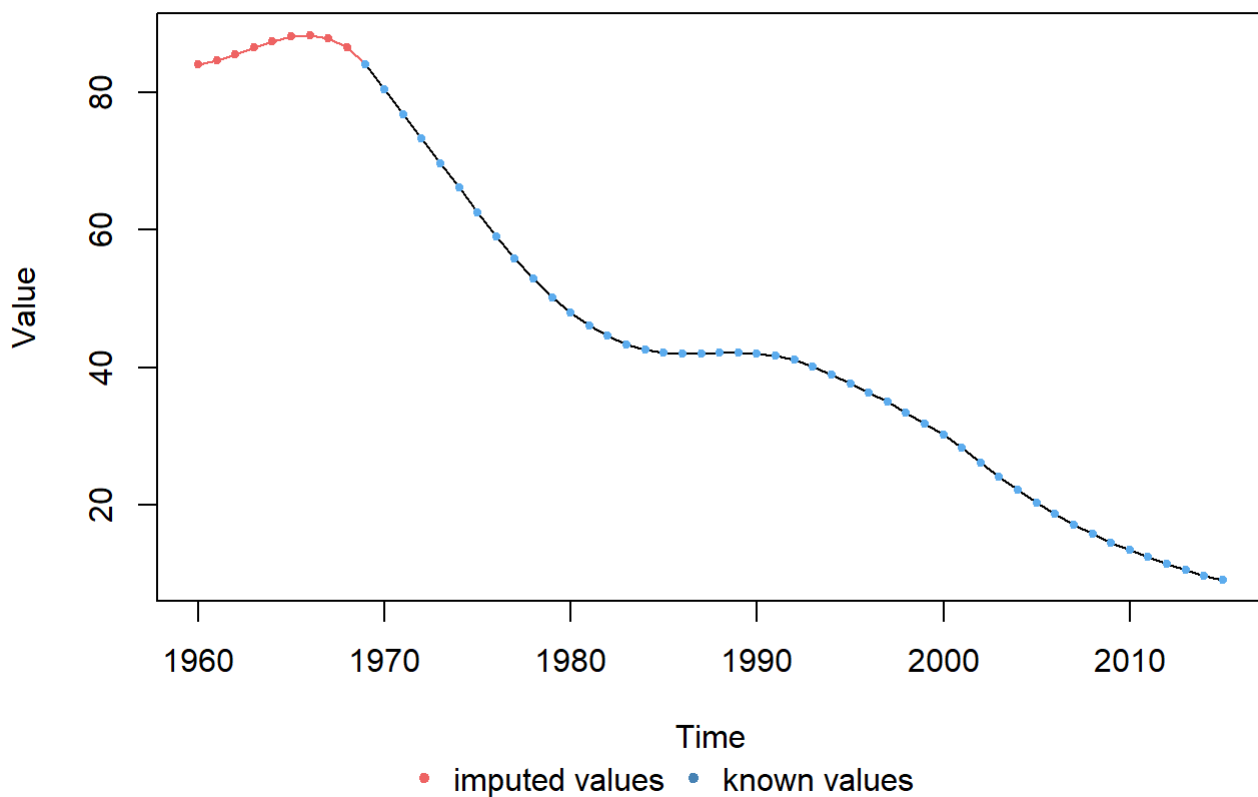


```

#Infant mortality per 1000 live births
infant_mortality <- sqldf('select Year, Value from CHN_indicators where IndicatorName
== "Mortality rate, infant (per 1,000 live births)"')
infant_mortality_na <- rep(NA, 56)
i <- 1
for (year in infant_mortality$Year){
  infant_mortality_na[(year-1960+1)] <- infant_mortality$Value[i]
  i = i + 1
}
infant_mortality_ts <- ts(data = infant_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(infant_mortality_ts)){
  infant_mortality_kalman <- na.kalman(infant_mortality_ts)
} else{
  infant_mortality_kalman <- infant_mortality_ts
}
health_CHN$'Mortality rate, infant (per 1,000 live births)' <- infant_mortality_kalma
n
plotNA.imputations(infant_mortality_ts, infant_mortality_kalman, main = "Infant morta
lity CHN")

```

Infant mortality CHN

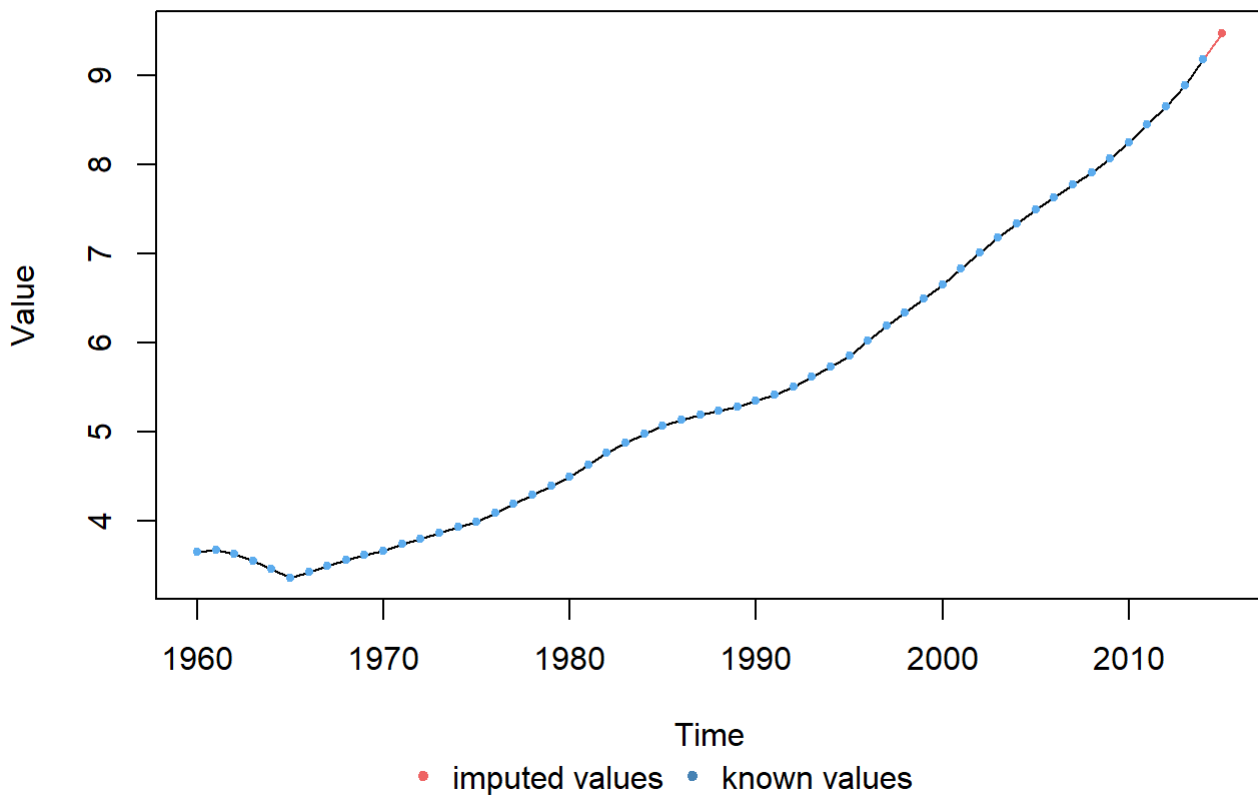



```

#Population over 65
population_over65 <- sqldf('select Year, Value from CHN_indicators where IndicatorName == "Population ages 65 and above (% of total)"')
population_over65_na <- rep(NA, 56)
i <- 1
for (year in population_over65$Year){
  population_over65_na[(year-1960+1)] <- population_over65$Value[i]
  i = i + 1
}
population_over65_ts <- ts(data = population_over65_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(population_over65_ts)){
  population_over65_kalman <- na.kalman(population_over65_ts)
} else{
  population_over65_kalman <- population_over65_ts
}
health_CHN$'Population ages 65 and above (% of total)' <- population_over65_kalman
plotNA.imputations(population_over65_ts, population_over65_kalman, main = "Population over 65 CHN")

```

Population over 65 CHN

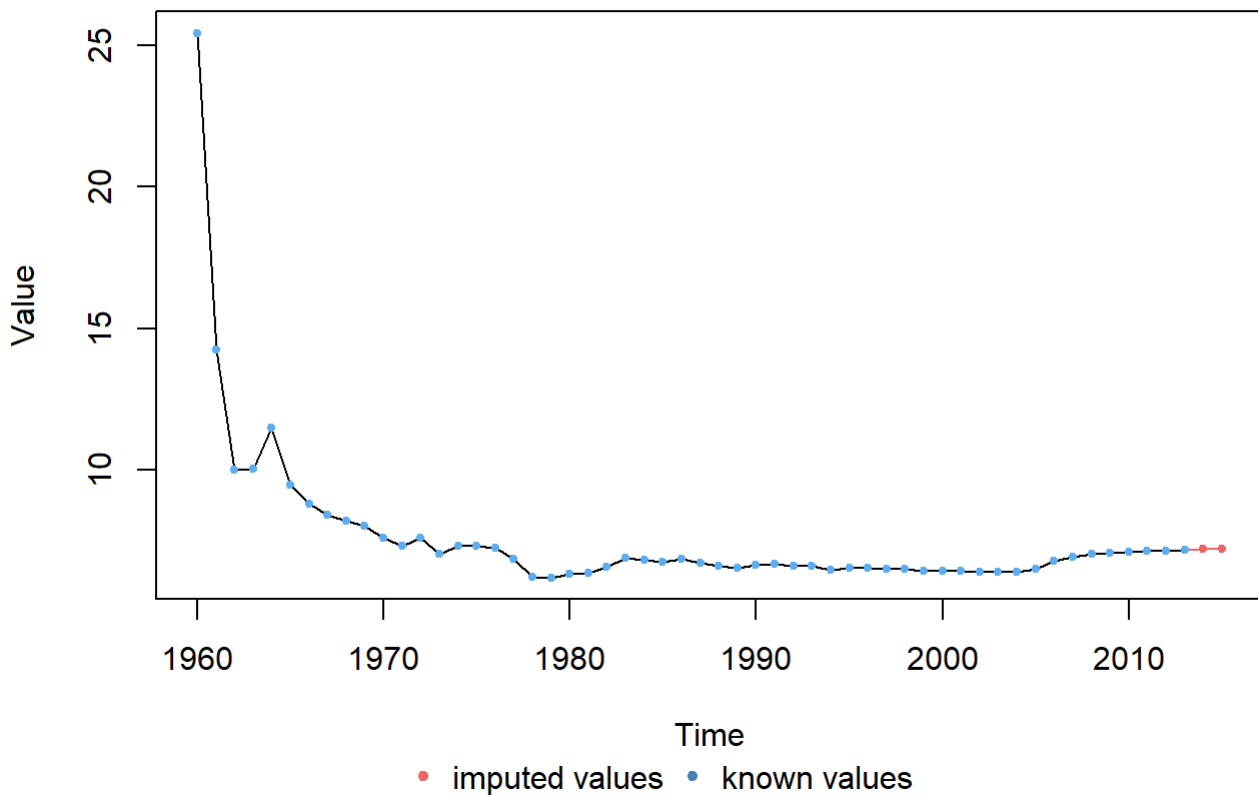


```

#Death rate per 1000 people
death_rate <- sqldf('select Year, Value from CHN_indicators where IndicatorName == "D
eath rate, crude (per 1,000 people)"')
death_rate_na <- rep(NA, 56)
i <- 1
for (year in death_rate$Year){
  death_rate_na[(year-1960+1)] <- death_rate$Value[i]
  i = i + 1
}
death_rate_ts <- ts(data = death_rate_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(death_rate_ts)){
  death_rate_kalman <- na.kalman(death_rate_ts)
} else{
  death_rate_kalman <- death_rate_ts
}
health_CHN$'Death rate, crude (per 1,000 people)' <- death_rate_kalman
plotNA.imputations(death_rate_ts, death_rate_kalman, main = "Death rate CHN")

```

Death rate CHN



```

#GDP per capita (constant 2005 US$)
GDP <- sqldf('select Year, Value from CHN_indicators where IndicatorName == "GDP per
  capita (constant 2005 US$)"')
GDP_na <- rep(NA, 56)
i <- 1
for (year in GDP$Year){
  GDP_na[(year-1960+1)] <- GDP$Value[i]
  i = i + 1
}
GDP_ts <- ts(data = GDP_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(GDP_ts)){
  GDP_kalman <- na.kalman(GDP_ts)
} else{
  GDP_kalman <- GDP_ts
}

```

```

## Warning in StructTS(data, ...): possible convergence problem: 'optim' gave
## code = 52 and message 'ERROR: ABNORMAL_TERMINATION_IN_LNSRCH'

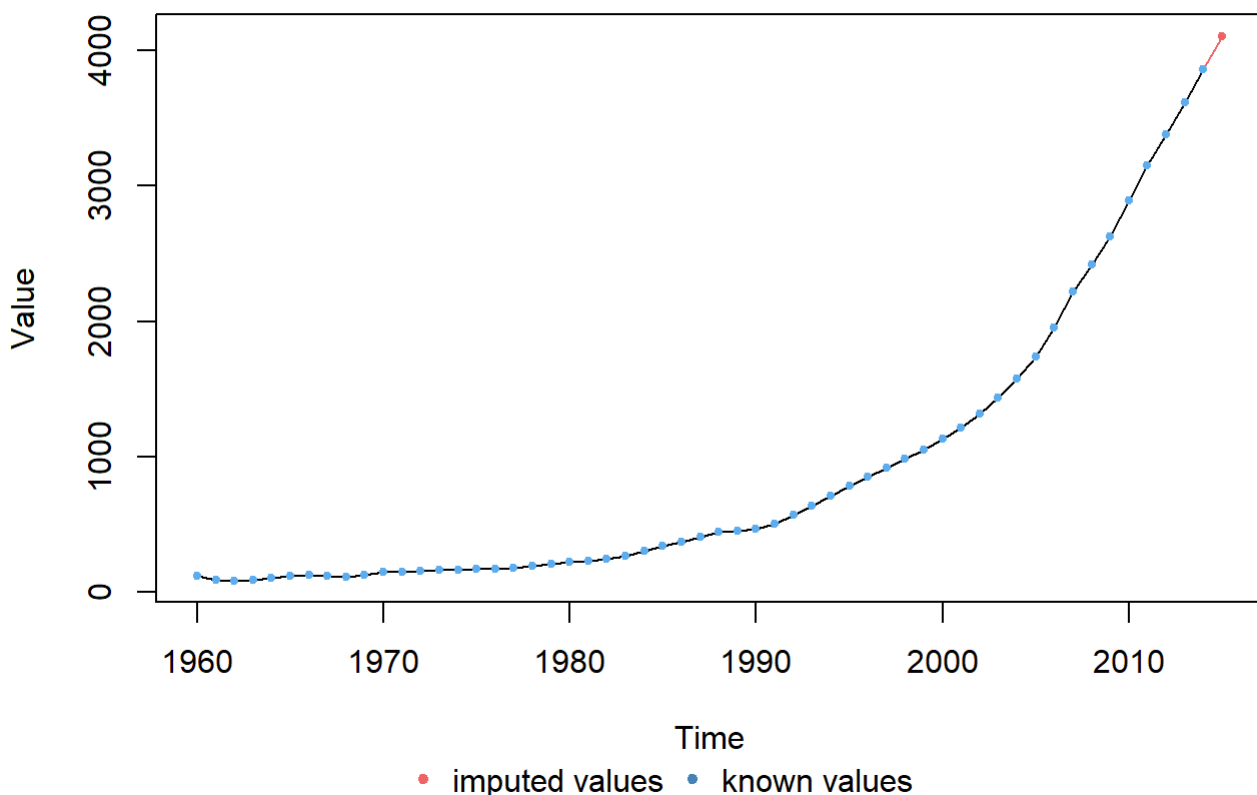
```

```

health_CHN$'GDP per capita (constant 2005 US$)' <- GDP_kalman
plotNA.imputations(GDP_ts, GDP_kalman, main = "GDP per capita (constant 2005 US$)")

```

GDP per capita (constant 2005 US\$)

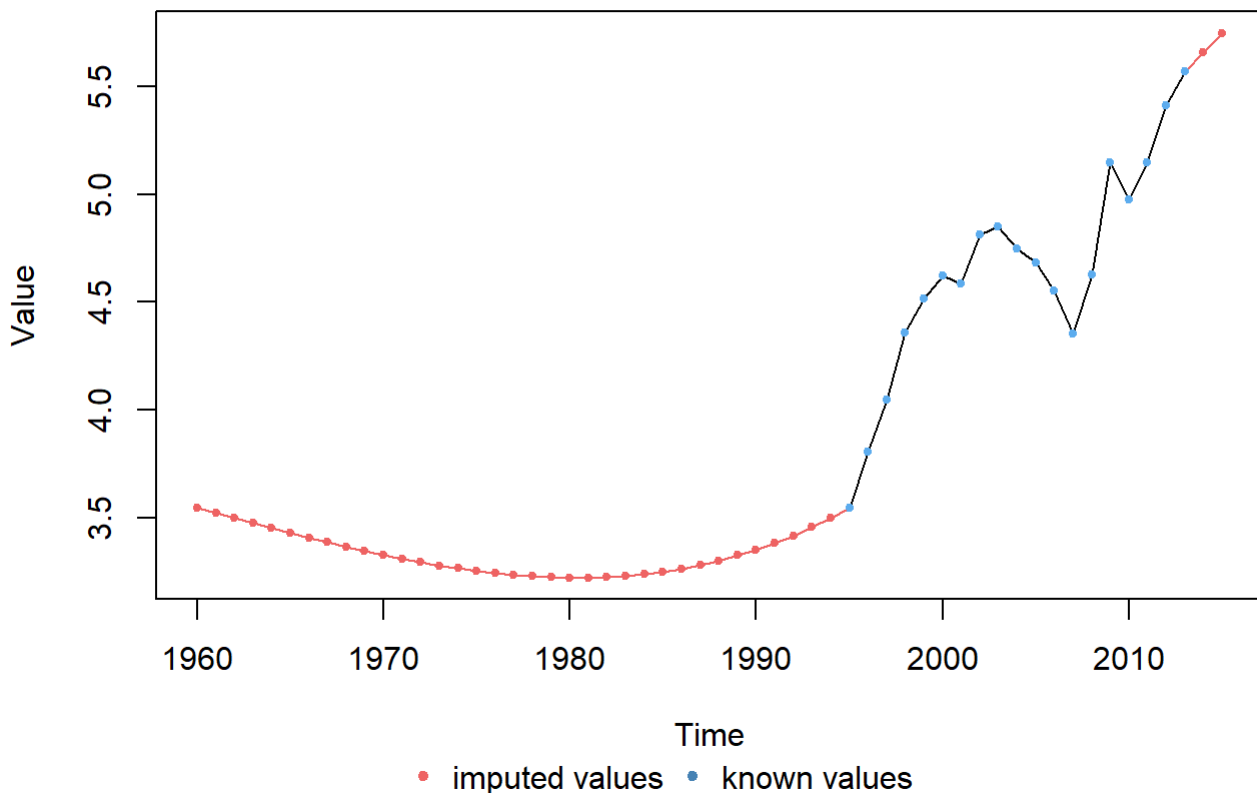


```

#health expenditure, total % of GDP
health_expenditure <- sqldf('select Year, Value from CHN_indicators where IndicatorName == "Health expenditure, total (% of GDP)"')
health_expenditure_na <- rep(NA, 56)
i <- 1
for (year in health_expenditure$Year){
  health_expenditure_na[(year-1960+1)] <- health_expenditure$Value[i]
  i = i + 1
}
health_expenditure_ts <- ts(data = health_expenditure_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(health_expenditure_ts)){
  health_expenditure_kalman <- na.kalman(health_expenditure_ts)
} else{
  health_expenditure_kalman <- health_expenditure_ts
}
health_CHN$'Health expenditure, total (% of GDP)' <- health_expenditure_kalman
plotNA.imputations(health_expenditure_ts, health_expenditure_kalman, main = "Health expenditure % of GDP CHN")

```

Health expenditure % of GDP CHN



For China, there is only one predictor that has too many NA values to impute that is Health expenditure(% of GDP) that we are not going to be able to use and also there is not information about the Government debt(% of GDP) in the case of China so we cannot use that one either. We delete these predictors from the dataset we are going to use.

```
health_CHN <- health_CHN[, -11]
```

INDIA

```

year <- sqldf('select distinct Year from world_indicators')
health_IND <- data.frame(year)

#Life expecntacy years
life_expectancy <- sqldf('select Year, Value from IND_indicators where IndicatorName
== "Life expectancy at birth, total (years)"')
life_expectancy_na <- rep(NA, 56)
i <- 1
for (year in life_expectancy$Year){
  life_expectancy_na[(year-1960+1)] <- life_expectancy$Value[i]
  i = i + 1
}
life_expectancy_ts <- ts(data = life_expectancy_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(life_expectancy_ts)){
  life_expectancy_kalman <- na.kalman(life_expectancy_ts)
} else{
  life_expectancy_kalman <- life_expectancy_ts
}

```

```

## Warning in StructTS(data, ...): possible convergence problem: 'optim' gave
## code = 52 and message 'ERROR: ABNORMAL_TERMINATION_IN_LNSRCH'

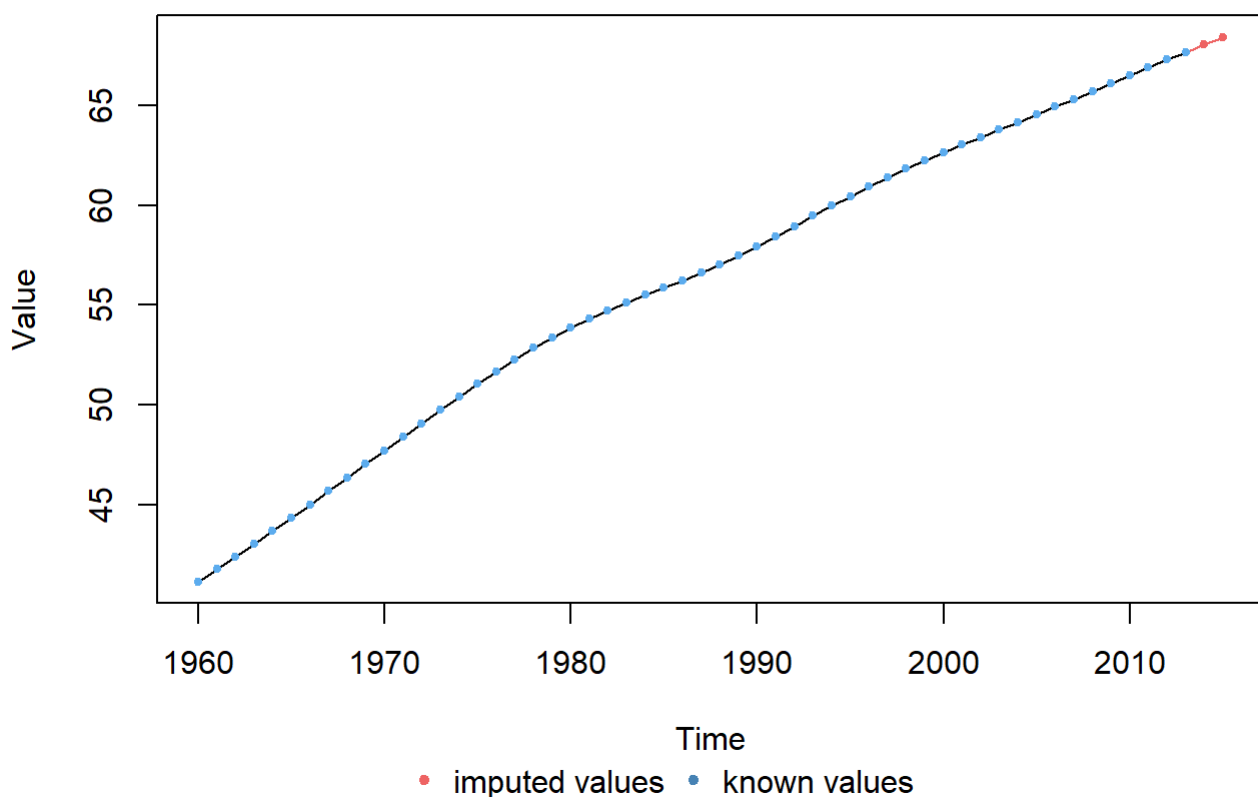
```

```

health_IND$'Life expectancy at birth, total (years)' <- life_expectancy_kalman
plotNA.imputations(life_expectancy_ts, life_expectancy_kalman, main = "Life expectanc
y years IND")

```

Life expectancy years IND

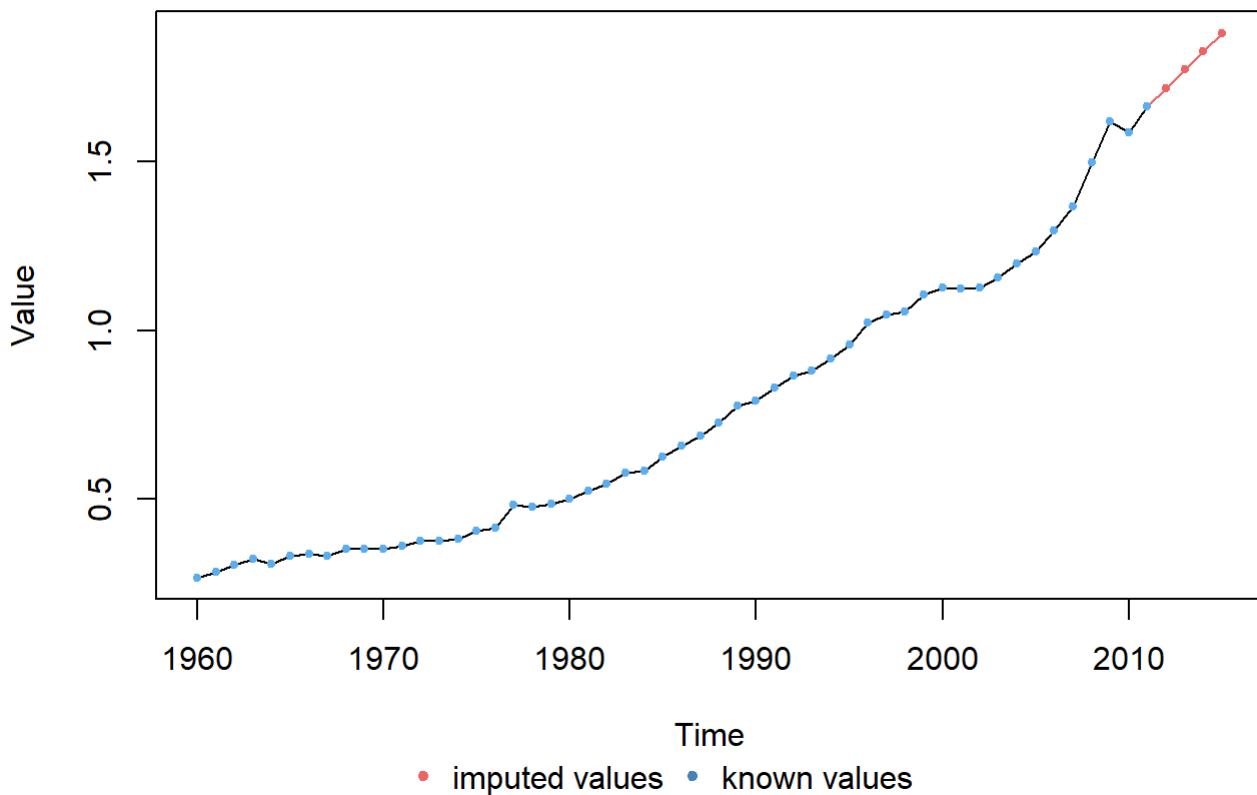


```

#CO2 emissions metrics tons per capita
CO2_emissions <- sqldf('select Year, Value from IND_indicators where IndicatorName ==
  "CO2 emissions (metric tons per capita)"')
CO2_emissions_na <- rep(NA, 56)
i <- 1
for (year in CO2_emissions$Year){
  CO2_emissions_na[(year-1960+1)] <- CO2_emissions$Value[i]
  i = i + 1
}
CO2_emissions_ts <- ts(data = CO2_emissions_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(CO2_emissions_ts)){
  CO2_emissions_kalman <- na.kalman(CO2_emissions_ts)
} else{
  CO2_emissions_kalman <- CO2_emissions_ts
}
health_IND$'CO2 emissions (metric tons per capita)' <- CO2_emissions_kalman
plotNA.imputations(CO2_emissions_ts, CO2_emissions_kalman, main = "CO2 emissions IND"
)

```

CO2 emissions IND

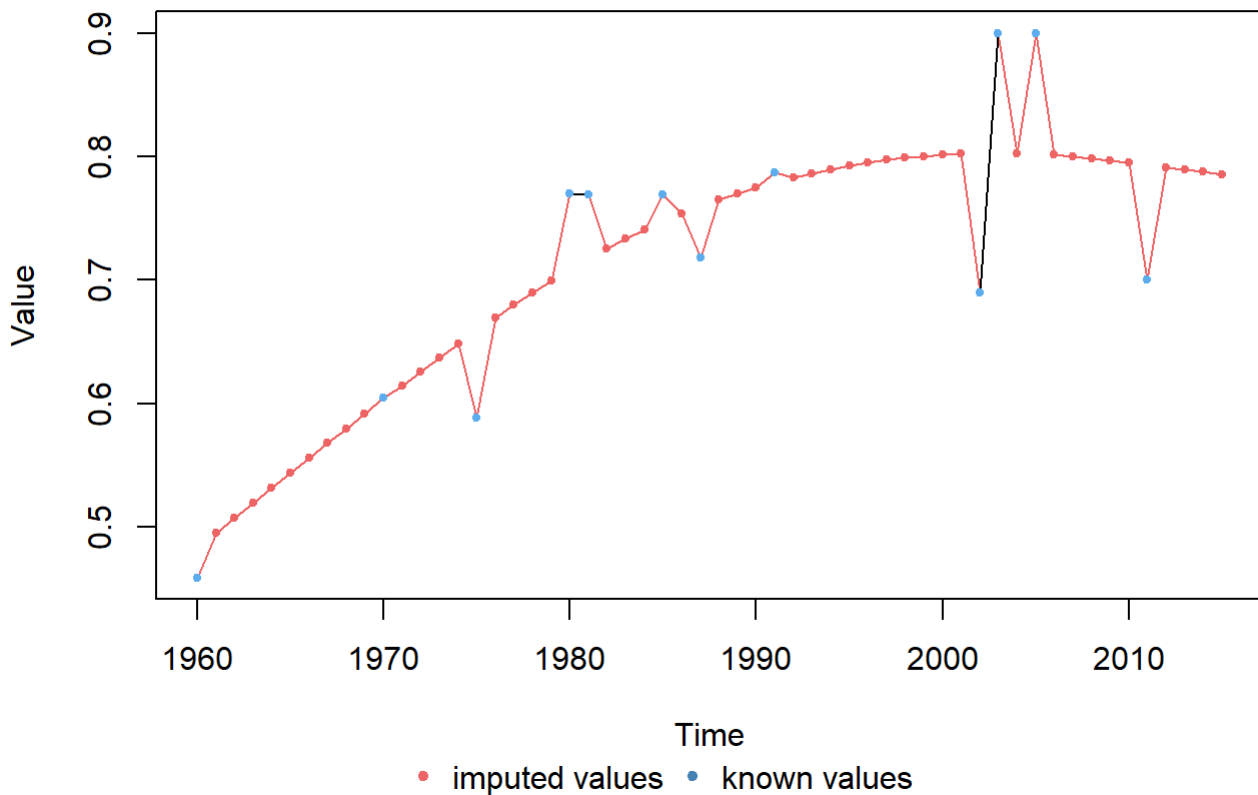


```

#Hospital beds per 1000 people
hospital_beds <- sqldf('select Year, Value from IND_indicators where IndicatorName ==
  "Hospital beds (per 1,000 people)"')
hospital_beds_na <- rep(NA, 56)
i <- 1
for (year in hospital_beds$Year){
  hospital_beds_na[(year-1960+1)] <- hospital_beds$Value[i]
  i = i + 1
}
hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(hospital_beds_ts)){
  hospital_beds_kalman <- na.kalman(hospital_beds_ts)
} else{
  hospital_beds_kalman <- hospital_beds_ts
}
health_IND$'Hospital beds (per 1,000 people)' <- hospital_beds_kalman
plotNA.imputations(hospital_beds_ts, hospital_beds_kalman, main = "Hospital beds IND"
)

```

Hospital beds IND



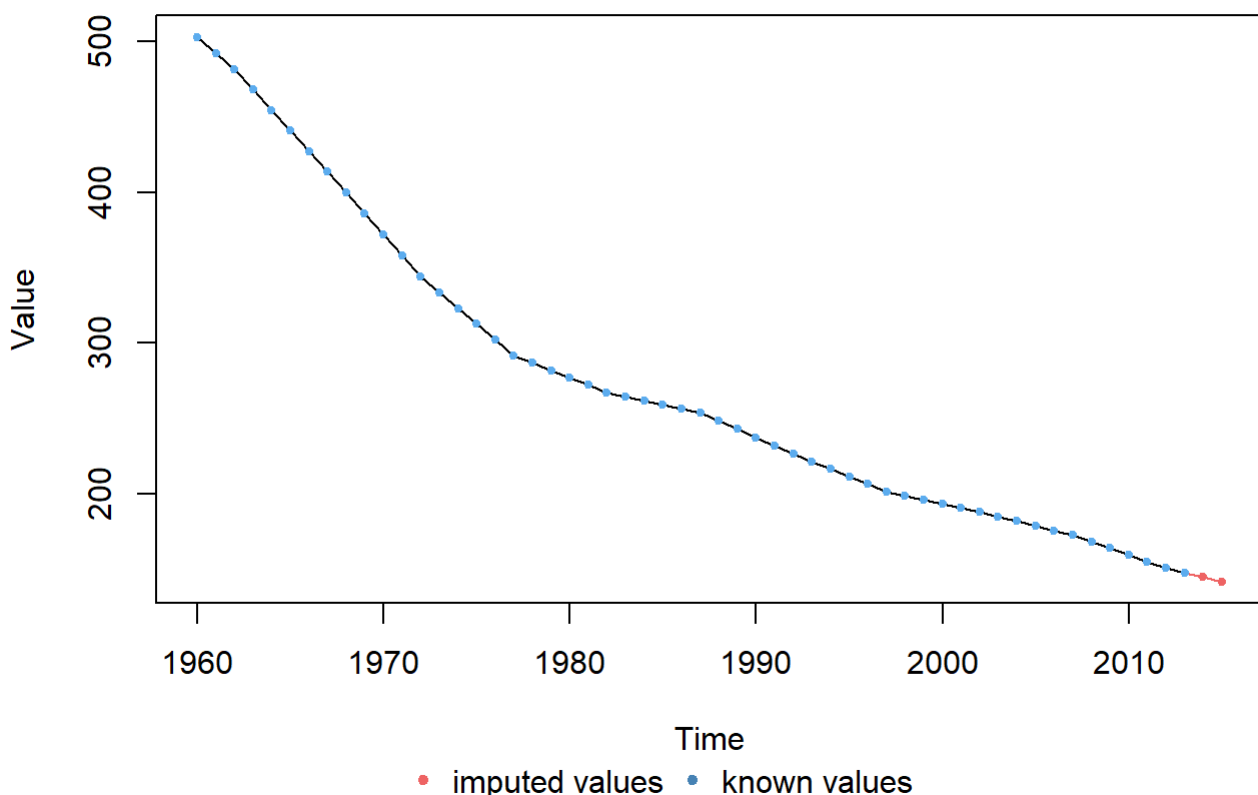
```

# hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency
= 1)
# hospital_beds_linear <- na.interpolation(hospital_beds_ts)
# hospital_beds_spline <- na.interpolation(hospital_beds_ts, option = "spline")
# hospital_beds_kalman <- na.kalman(hospital_beds_ts)
# plotNA.imputations(hospital_beds_ts, hospital_beds_linear)
# plotNA.imputations(hospital_beds_ts, hospital_beds_spline)
# plotNA.imputations(hospital_beds_ts, hospital_beds_kalman)

#Female mortality per 1000 females
female_mortality <- sqldf('select Year, Value from IND_indicators where IndicatorName
== "Mortality rate, adult, female (per 1,000 female adults)"')
female_mortality_na <- rep(NA, 56)
i <- 1
for (year in female_mortality$Year){
  female_mortality_na[(year-1960+1)] <- female_mortality$Value[i]
  i = i + 1
}
female_mortality_ts <- ts(data = female_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(female_mortality_ts)){
  female_mortality_kalman <- na.kalman(female_mortality_ts)
} else{
  female_mortality_kalman <- female_mortality_ts
}
health_IND$'Mortality rate, adult, female (per 1,000 female adults)' <- female_mortal
ity_kalman
plotNA.imputations(female_mortality_ts, female_mortality_kalman, main = "Female morta
lity IND")

```

Female mortality IND

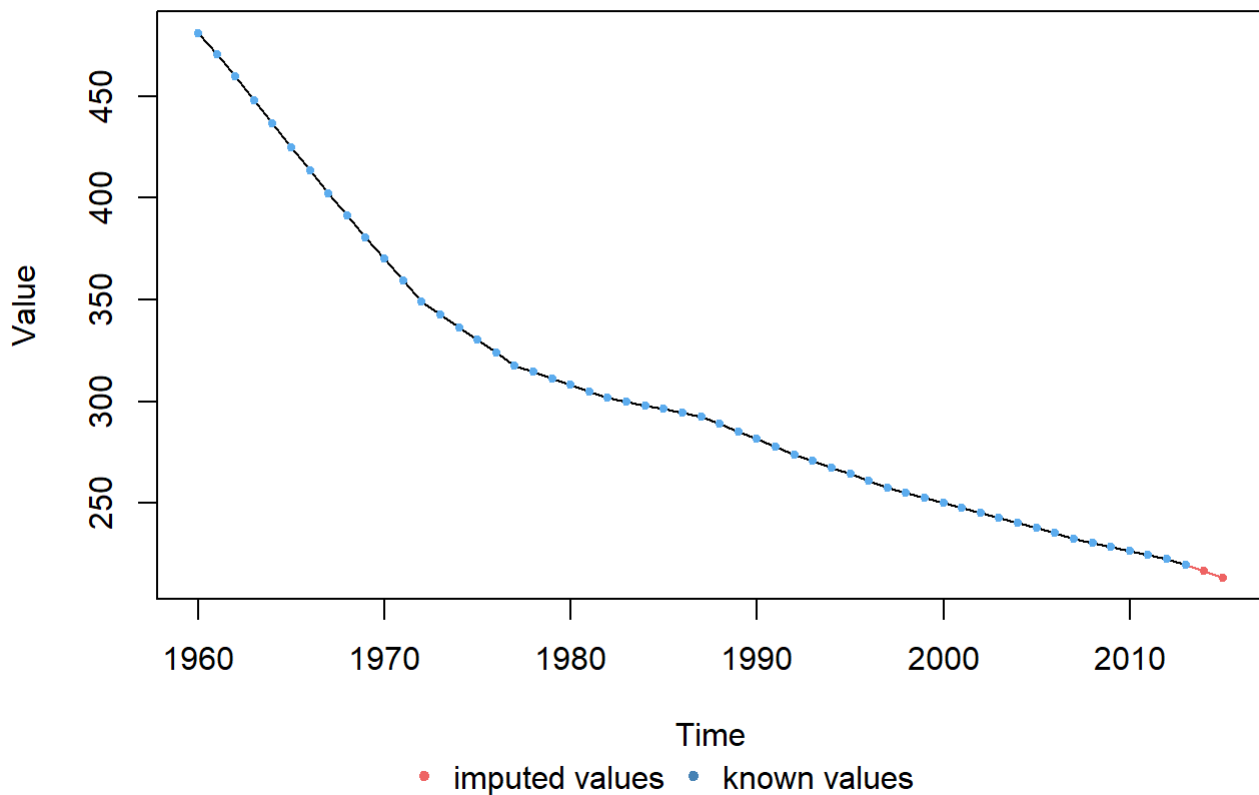



```

#Male mortality per 1000 males
male_mortality <- sqldf('select Year, Value from IND_indicators where IndicatorName =
= "Mortality rate, adult, male (per 1,000 male adults)"')
male_mortality_na <- rep(NA, 56)
i <- 1
for (year in male_mortality$Year){
  male_mortality_na[(year-1960+1)] <- male_mortality$Value[i]
  i = i + 1
}
male_mortality_ts <- ts(data = male_mortality_na, start = 1960, end = 2015, frequency
= 1)
if(anyNA(male_mortality_ts)){
  male_mortality_kalman <- na.kalman(male_mortality_ts)
} else{
  male_mortality_kalman <- male_mortality_ts
}
health_IND$'Mortality rate, adult, male (per 1,000 male adults)' <- male_mortality_kalman
plotNA.imputations(male_mortality_ts, male_mortality_kalman, main = "Male mortality I
ND")

```

Male mortality IND

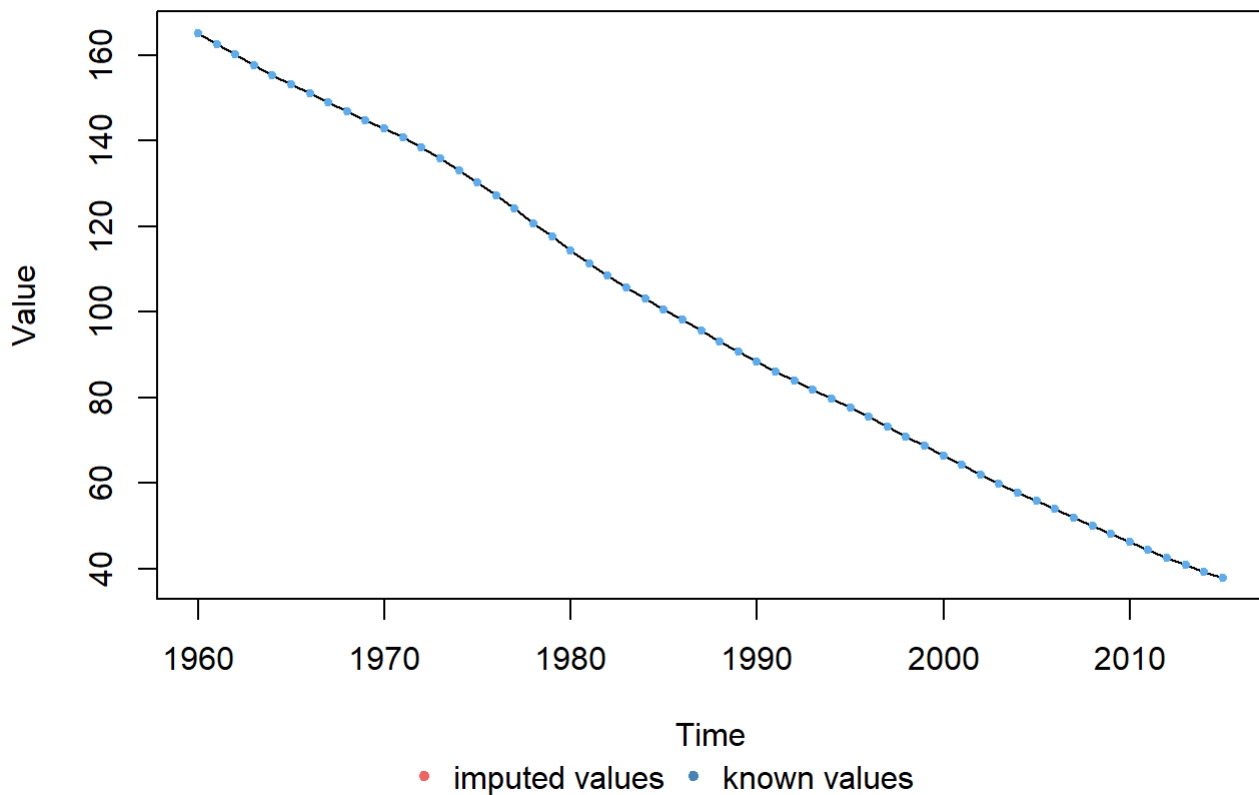


```

#Infant mortality per 1000 live births
infant_mortality <- sqldf('select Year, Value from IND_indicators where IndicatorName
== "Mortality rate, infant (per 1,000 live births)"')
infant_mortality_na <- rep(NA, 56)
i <- 1
for (year in infant_mortality$Year){
  infant_mortality_na[(year-1960+1)] <- infant_mortality$Value[i]
  i = i + 1
}
infant_mortality_ts <- ts(data = infant_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(infant_mortality_ts)){
  infant_mortality_kalman <- na.kalman(infant_mortality_ts)
} else{
  infant_mortality_kalman <- infant_mortality_ts
}
health_IND$'Mortality rate, infant (per 1,000 live births)' <- infant_mortality_kalma
n
plotNA.imputations(infant_mortality_ts, infant_mortality_kalman, main = "Infant morta
lity IND")

```

Infant mortality IND

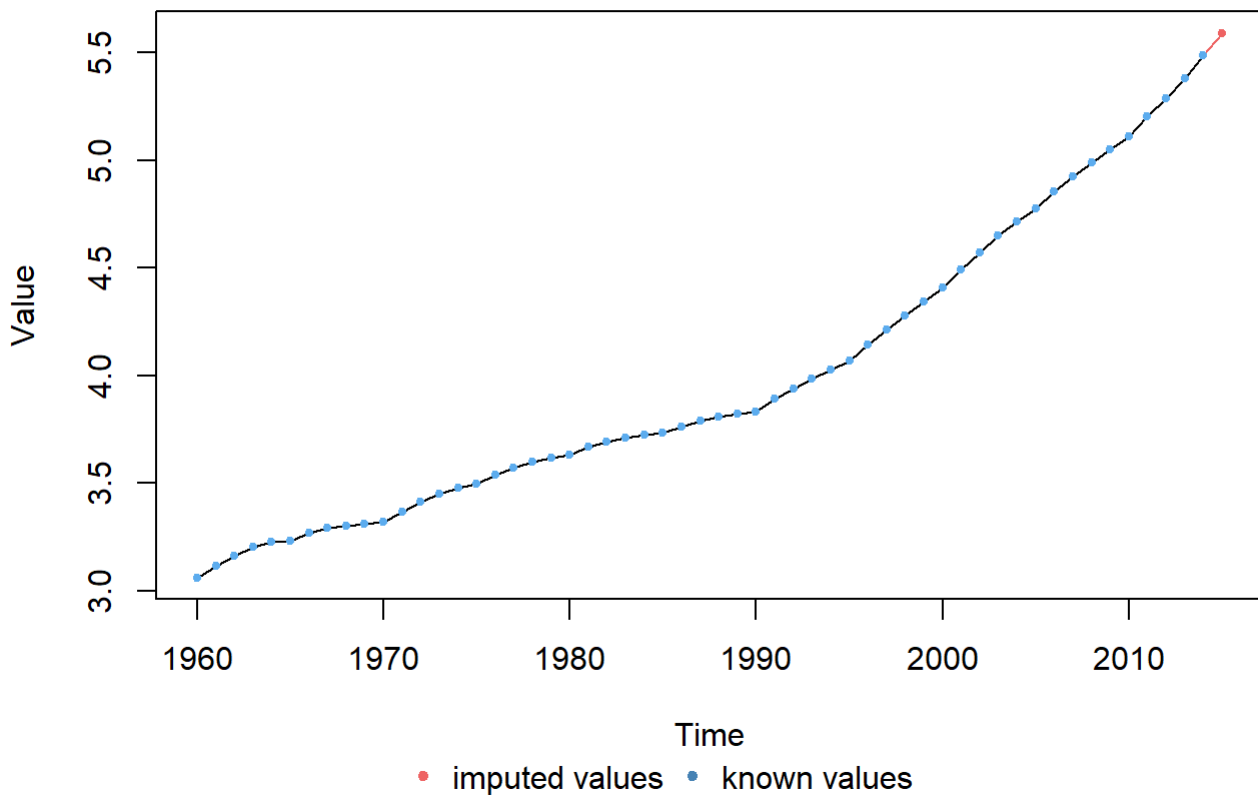


```

#Population over 65
population_over65 <- sqldf('select Year, Value from IND_indicators where IndicatorName == "Population ages 65 and above (% of total)"')
population_over65_na <- rep(NA, 56)
i <- 1
for (year in population_over65$Year){
  population_over65_na[(year-1960+1)] <- population_over65$Value[i]
  i = i + 1
}
population_over65_ts <- ts(data = population_over65_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(population_over65_ts)){
  population_over65_kalman <- na.kalman(population_over65_ts)
} else{
  population_over65_kalman <- population_over65_ts
}
health_IND$'Population ages 65 and above (% of total)' <- population_over65_kalman
plotNA.imputations(population_over65_ts, population_over65_kalman, main = "Population over 65 IND")

```

Population over 65 IND

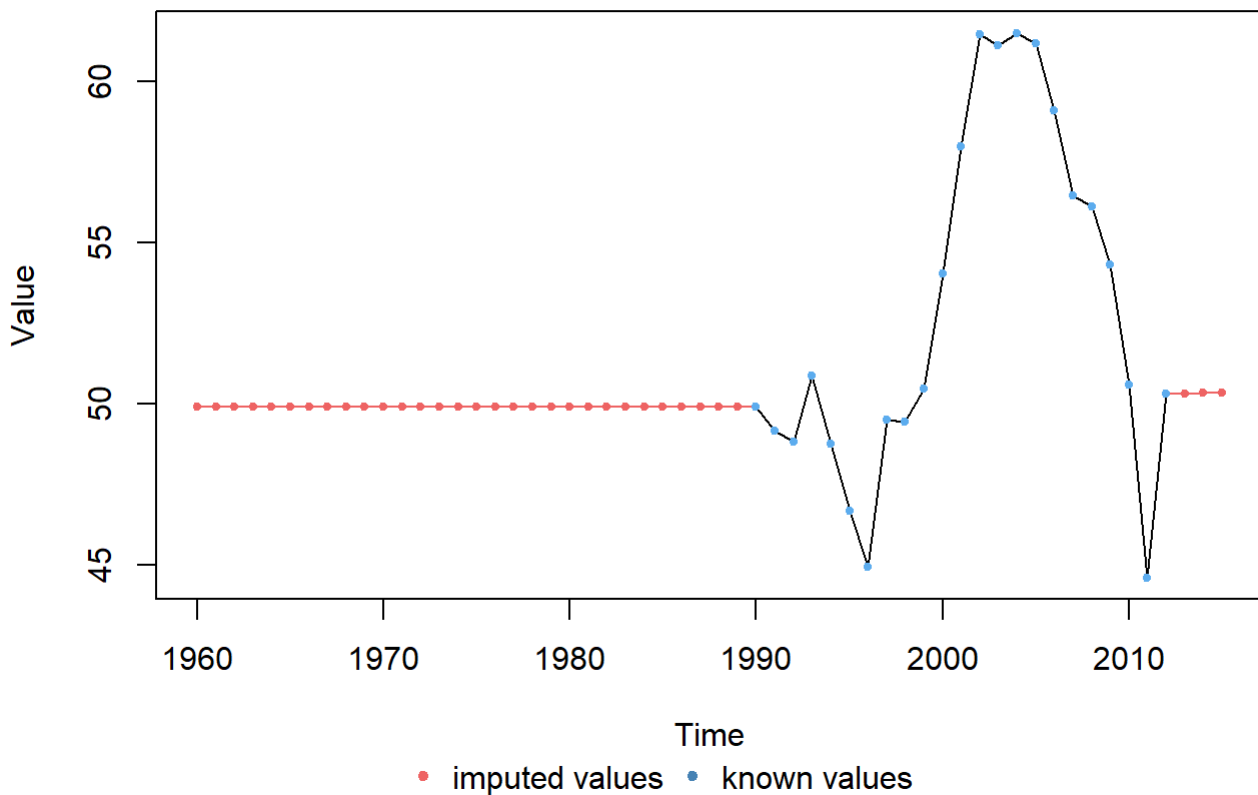


```

#Government debt, total % of GDP
government_debt <- sqldf('select Year, Value from IND_indicators where IndicatorName
== "Central government debt, total (% of GDP)"')
government_debt_na <- rep(NA, 56)
i <- 1
for (year in government_debt$Year){
  government_debt_na[(year-1960+1)] <- government_debt$Value[i]
  i = i + 1
}
government_debt_ts <- ts(data = government_debt_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(government_debt_ts)){
  government_debt_kalman <- na.kalman(government_debt_ts)
} else{
  government_debt_kalman <- government_debt_ts
}
health_IND$'Central government debt, total (% of GDP)' <- government_debt_kalman
plotNA.imputations(government_debt_ts, government_debt_kalman, main = "Government deb
t % of GDP IND")

```

Government debt % of GDP IND

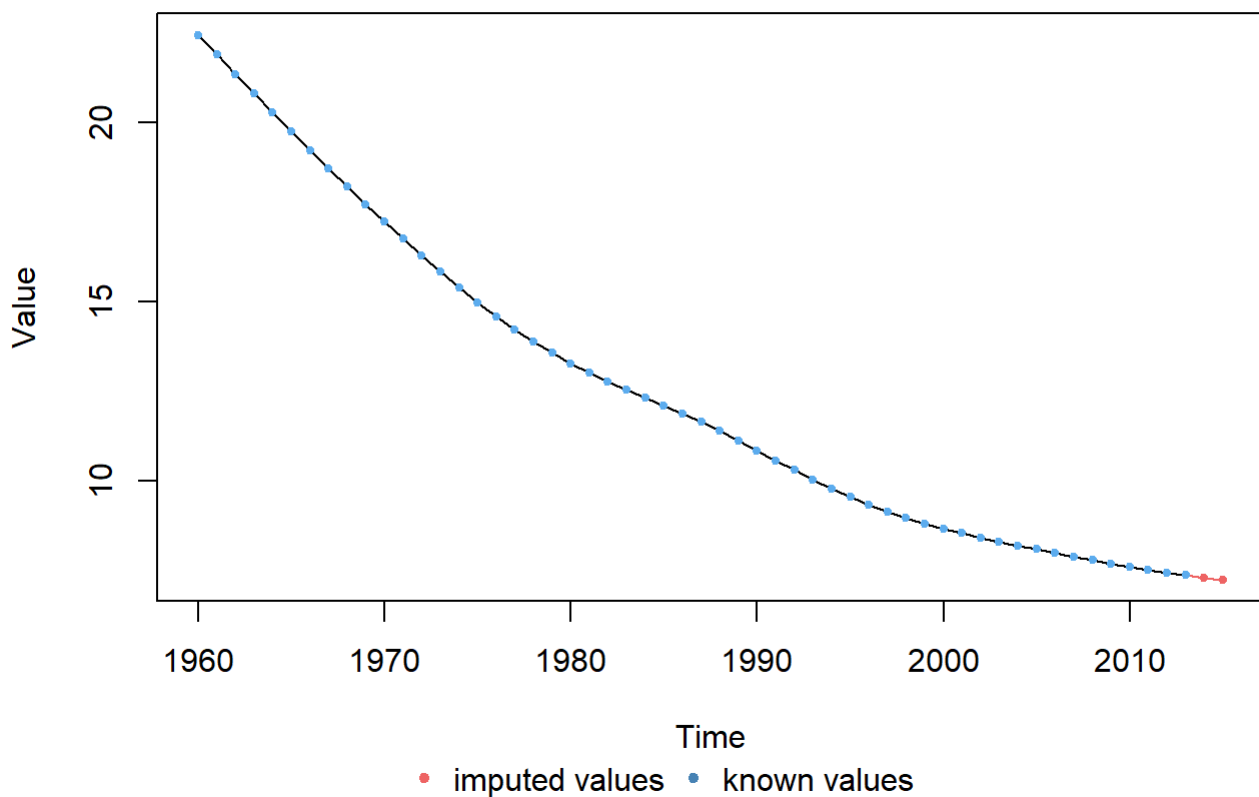


```

#Death rate per 1000 people
death_rate <- sqldf('select Year, Value from IND_indicators where IndicatorName == "D
eath rate, crude (per 1,000 people)"')
death_rate_na <- rep(NA, 56)
i <- 1
for (year in death_rate$Year){
  death_rate_na[(year-1960+1)] <- death_rate$Value[i]
  i = i + 1
}
death_rate_ts <- ts(data = death_rate_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(death_rate_ts)){
  death_rate_kalman <- na.kalman(death_rate_ts)
} else{
  death_rate_kalman <- death_rate_ts
}
health_IND$'Death rate, crude (per 1,000 people)' <- death_rate_kalman
plotNA.imputations(death_rate_ts, death_rate_kalman, main = "Death rate IND")

```

Death rate IND

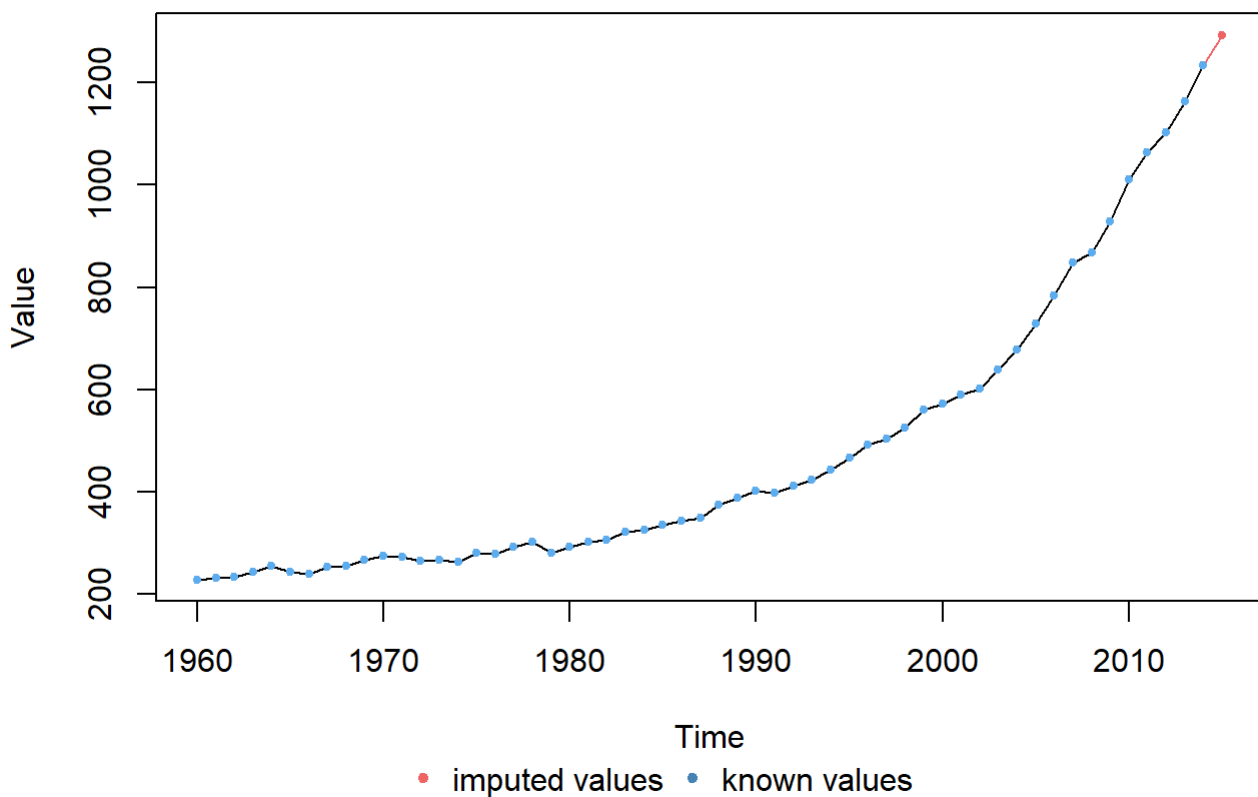


```

#GDP per capita (constant 2005 US$)
GDP <- sqldf('select Year, Value from IND_indicators where IndicatorName == "GDP per
  capita (constant 2005 US$)"')
GDP_na <- rep(NA, 56)
i <- 1
for (year in GDP$Year){
  GDP_na[(year-1960+1)] <- GDP$Value[i]
  i = i + 1
}
GDP_ts <- ts(data = GDP_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(GDP_ts)){
  GDP_kalman <- na.kalman(GDP_ts)
} else{
  GDP_kalman <- GDP_ts
}
health_IND$'GDP per capita (constant 2005 US$)' <- GDP_kalman
plotNA.imputations(GDP_ts, GDP_kalman, main = "GDP per capita (constant 2005 US$)")

```

GDP per capita (constant 2005 US\$)

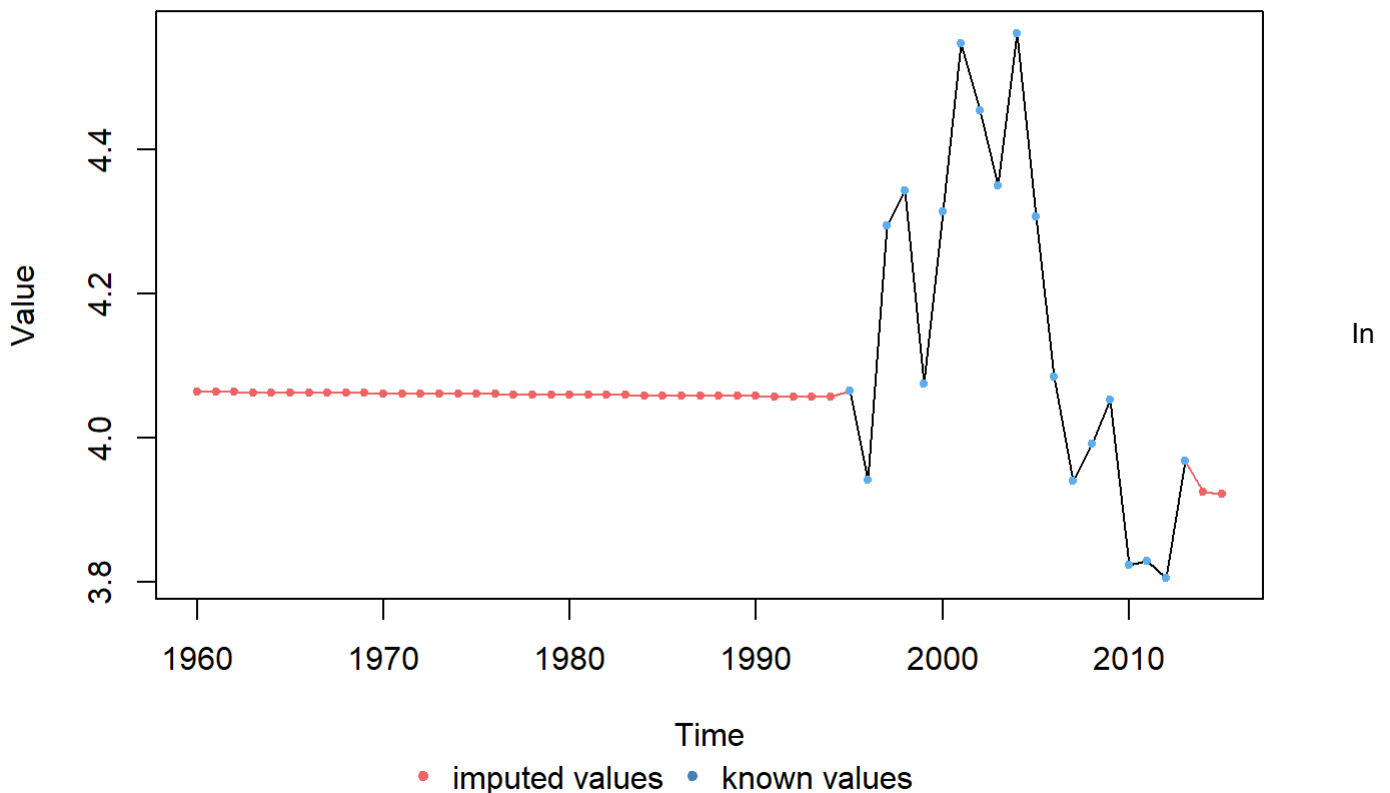


```

#health expenditure, total % of GDP
health_expenditure <- sqldf('select Year, Value from IND_indicators where IndicatorName == "Health expenditure, total (% of GDP)"')
health_expenditure_na <- rep(NA, 56)
i <- 1
for (year in health_expenditure$Year){
  health_expenditure_na[(year-1960+1)] <- health_expenditure$Value[i]
  i = i + 1
}
health_expenditure_ts <- ts(data = health_expenditure_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(health_expenditure_ts)){
  health_expenditure_kalman <- na.kalman(health_expenditure_ts)
} else{
  health_expenditure_kalman <- health_expenditure_ts
}
health_IND$'Health expenditure, total (% of GDP)' <- health_expenditure_kalman
plotNA.imputations(health_expenditure_ts, health_expenditure_kalman, main = "Health expenditure % of GDP IND")

```

Health expenditure % of GDP IND



the case of India there is not enough values for the same indicators that for Spain: Hospital beds, Government debt(% of GDP) and Health expenditure(% of GDP)

```
health_IND <- health_IND[, -c(4, 9, 12)]
```

FRANCE

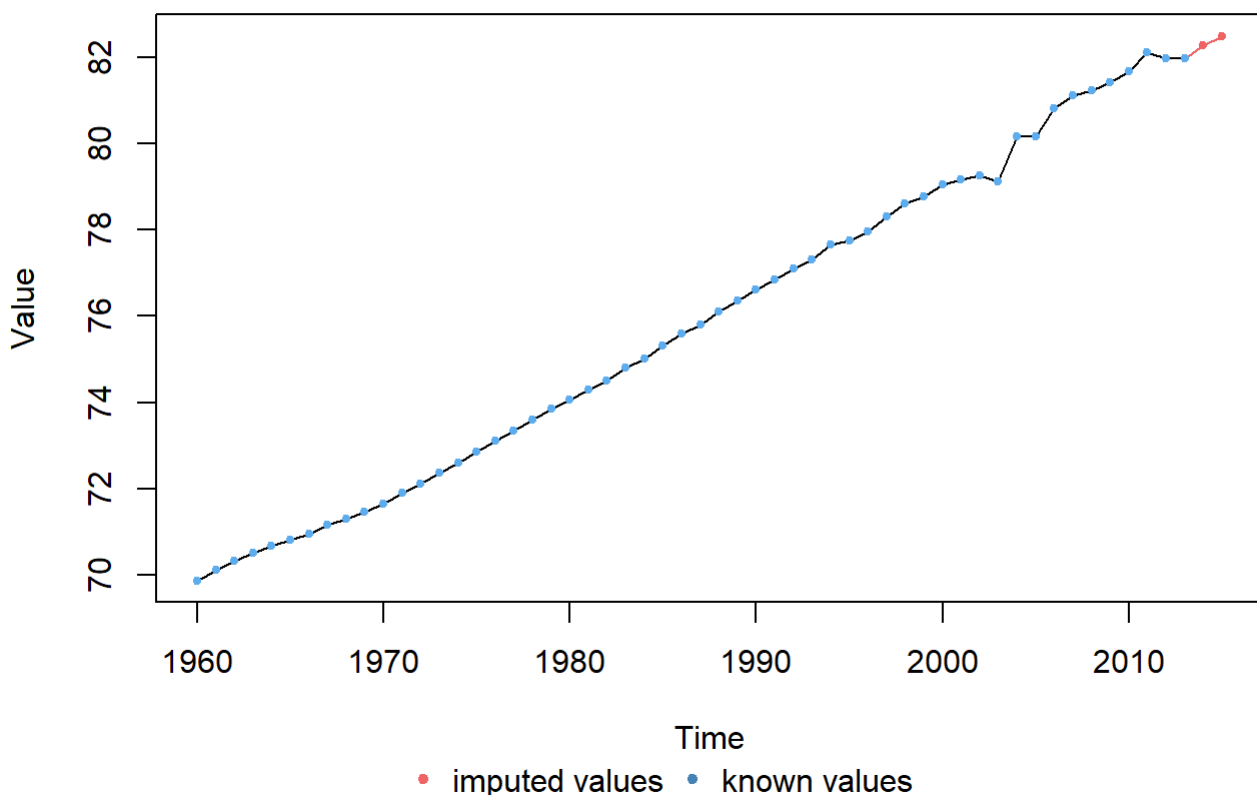
```

year <- sqldf('select distinct Year from world_indicators')
health_FRA <- data.frame(year)

#Life expectnacy years
life_expectancy <- sqldf('select Year, Value from FRA_indicators where IndicatorName
== "Life expectancy at birth, total (years)"')
life_expectancy_na <- rep(NA, 56)
i <- 1
for (year in life_expectancy$Year){
  life_expectancy_na[(year-1960+1)] <- life_expectancy$Value[i]
  i = i + 1
}
life_expectancy_ts <- ts(data = life_expectancy_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(life_expectancy_ts)){
  life_expectancy_kalman <- na.kalman(life_expectancy_ts)
} else{
  life_expectancy_kalman <- life_expectancy_ts
}
health_FRA$'Life expectancy at birth, total (years)' <- life_expectancy_kalman
plotNA.imputations(life_expectancy_ts, life_expectancy_kalman, main = "Life expectanc
y years FRA")

```

Life expectancy years FRA

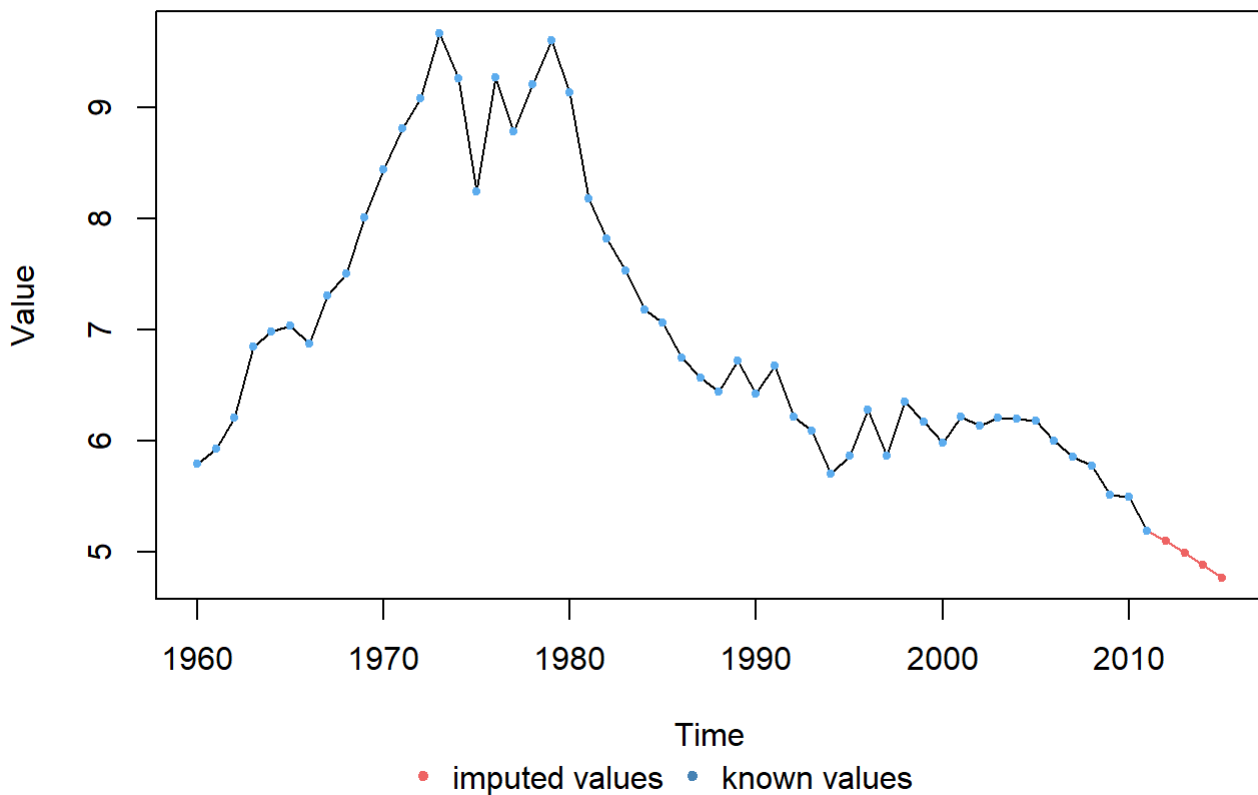



```

#CO2 emissions metrics tons per capita
CO2_emissions <- sqldf('select Year, Value from FRA_indicators where IndicatorName ==
  "CO2 emissions (metric tons per capita)"')
CO2_emissions_na <- rep(NA, 56)
i <- 1
for (year in CO2_emissions$Year){
  CO2_emissions_na[(year-1960+1)] <- CO2_emissions$Value[i]
  i = i + 1
}
CO2_emissions_ts <- ts(data = CO2_emissions_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(CO2_emissions_ts)){
  CO2_emissions_kalman <- na.kalman(CO2_emissions_ts)
} else{
  CO2_emissions_kalman <- CO2_emissions_ts
}
health_FRA$'CO2 emissions (metric tons per capita)' <- CO2_emissions_kalman
plotNA.imputations(CO2_emissions_ts, CO2_emissions_kalman, main = "CO2 emissions FRA"
)

```

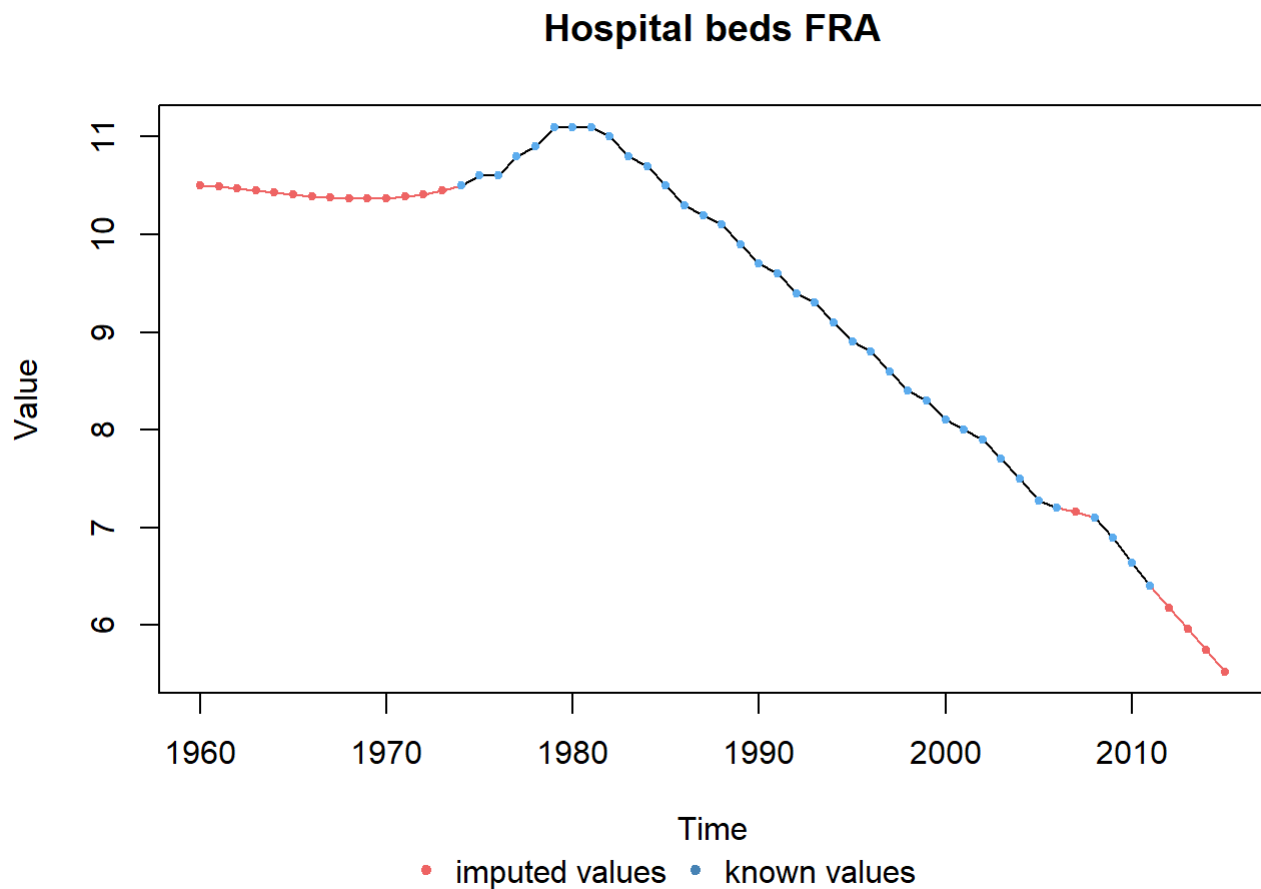
CO2 emissions FRA



```

#Hospital beds per 1000 people
hospital_beds <- sqldf('select Year, Value from FRA_indicators where IndicatorName ==
  "Hospital beds (per 1,000 people)"')
hospital_beds_na <- rep(NA, 56)
i <- 1
for (year in hospital_beds$Year){
  hospital_beds_na[(year-1960+1)] <- hospital_beds$Value[i]
  i = i + 1
}
hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(hospital_beds_ts)){
  hospital_beds_kalman <- na.kalman(hospital_beds_ts)
} else{
  hospital_beds_kalman <- hospital_beds_ts
}
health_FRA$'Hospital beds (per 1,000 people)' <- hospital_beds_kalman
plotNA.imputations(hospital_beds_ts, hospital_beds_kalman, main = "Hospital beds FRA"
)

```



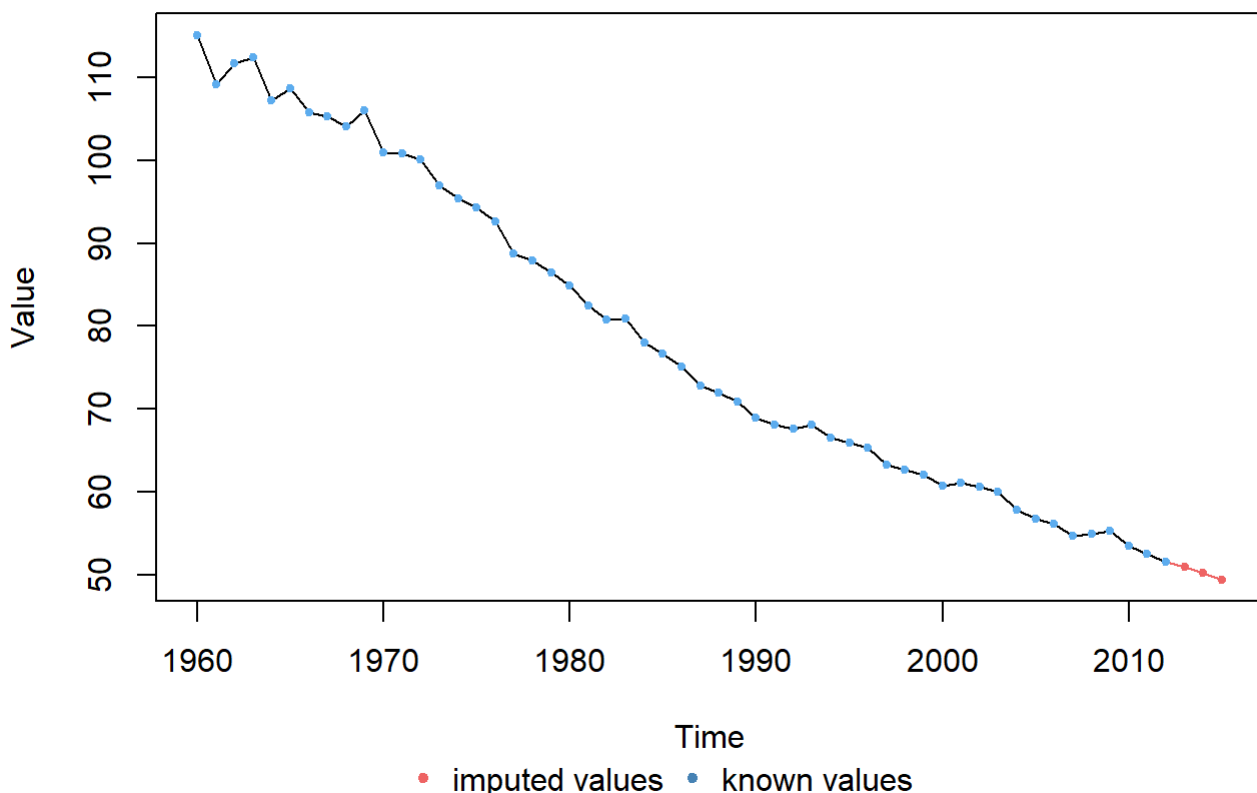
```

# hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency
= 1)
# hospital_beds_linear <- na.interpolation(hospital_beds_ts)
# hospital_beds_spline <- na.interpolation(hospital_beds_ts, option = "spline")
# hospital_beds_kalman <- na.kalman(hospital_beds_ts)
# plotNA.imputations(hospital_beds_ts, hospital_beds_linear)
# plotNA.imputations(hospital_beds_ts, hospital_beds_spline)
# plotNA.imputations(hospital_beds_ts, hospital_beds_kalman)

#Female mortality per 1000 females
female_mortality <- sqldf('select Year, Value from FRA_indicators where IndicatorName
== "Mortality rate, adult, female (per 1,000 female adults)"')
female_mortality_na <- rep(NA, 56)
i <- 1
for (year in female_mortality$Year){
  female_mortality_na[(year-1960+1)] <- female_mortality$Value[i]
  i = i + 1
}
female_mortality_ts <- ts(data = female_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(female_mortality_ts)){
  female_mortality_kalman <- na.kalman(female_mortality_ts)
} else{
  female_mortality_kalman <- female_mortality_ts
}
health_FRA$'Mortality rate, adult, female (per 1,000 female adults)' <- female_mortal
ity_kalman
plotNA.imputations(female_mortality_ts, female_mortality_kalman, main = "Female morta
lity FRA")

```

Female mortality FRA

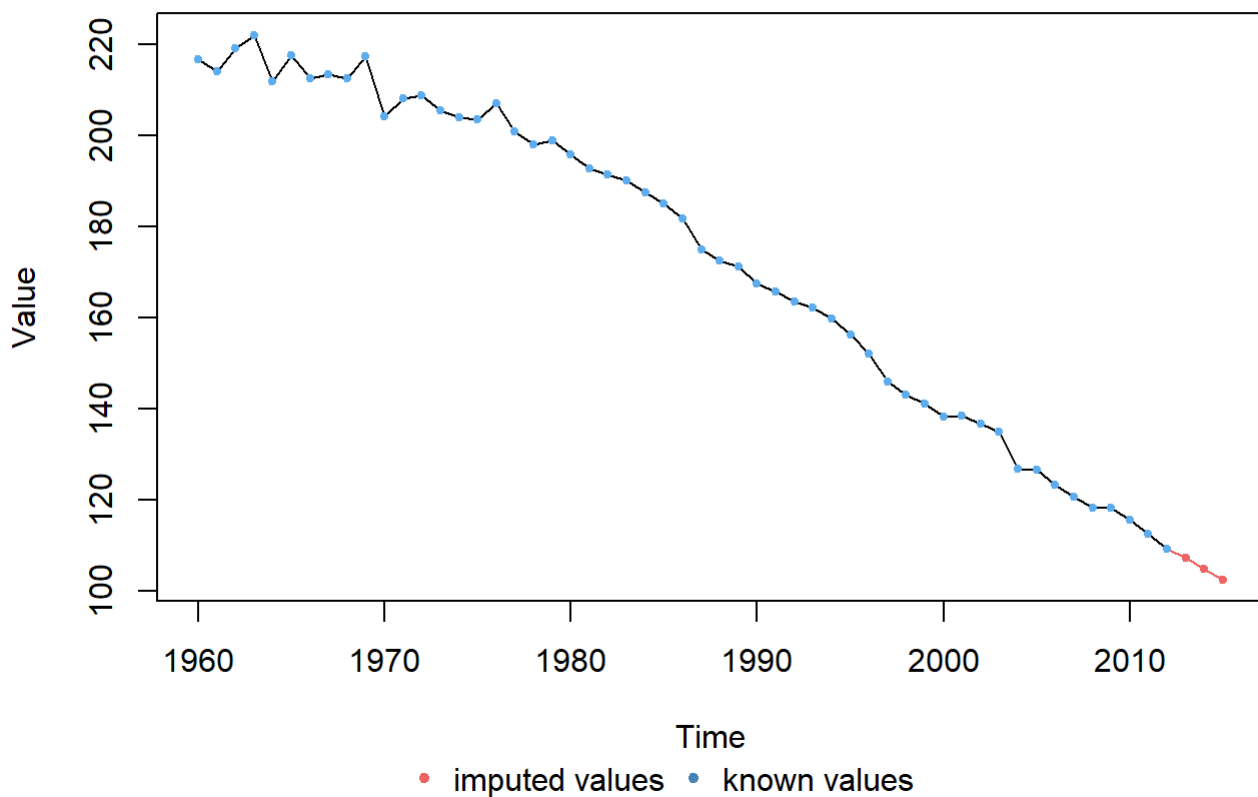


```

#Male mortality per 1000 males
male_mortality <- sqldf('select Year, Value from FRA_indicators where IndicatorName =
= "Mortality rate, adult, male (per 1,000 male adults)"')
male_mortality_na <- rep(NA, 56)
i <- 1
for (year in male_mortality$Year){
  male_mortality_na[(year-1960+1)] <- male_mortality$Value[i]
  i = i + 1
}
male_mortality_ts <- ts(data = male_mortality_na, start = 1960, end = 2015, frequency
= 1)
if(anyNA(male_mortality_ts)){
  male_mortality_kalman <- na.kalman(male_mortality_ts)
} else{
  male_mortality_kalman <- male_mortality_ts
}
health_FRA$'Mortality rate, adult, male (per 1,000 male adults)' <- male_mortality_kalman
plotNA.imputations(male_mortality_ts, male_mortality_kalman, main = "Male mortality F
RA")

```

Male mortality FRA

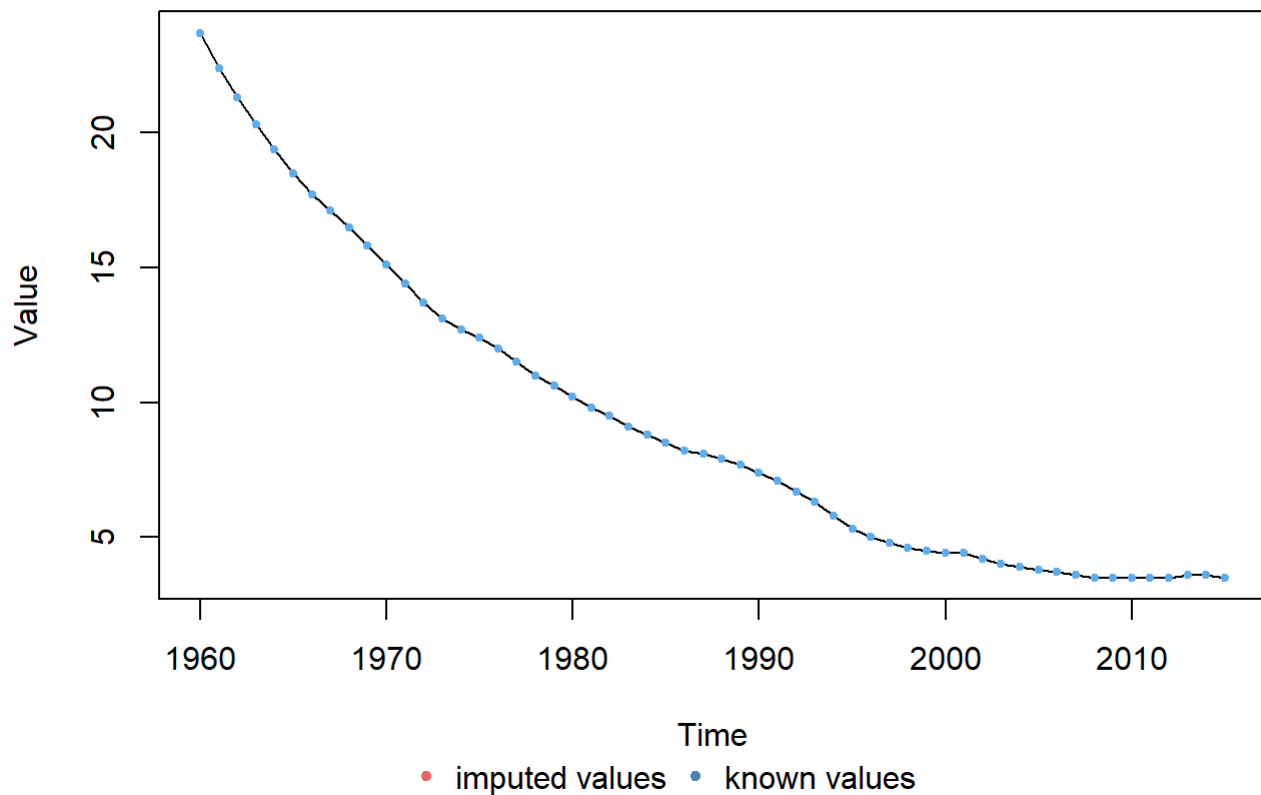


```

#Infant mortality per 1000 live births
infant_mortality <- sqldf('select Year, Value from FRA_indicators where IndicatorName
== "Mortality rate, infant (per 1,000 live births)"')
infant_mortality_na <- rep(NA, 56)
i <- 1
for (year in infant_mortality$Year){
  infant_mortality_na[(year-1960+1)] <- infant_mortality$Value[i]
  i = i + 1
}
infant_mortality_ts <- ts(data = infant_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(infant_mortality_ts)){
  infant_mortality_kalman <- na.kalman(infant_mortality_ts)
} else{
  infant_mortality_kalman <- infant_mortality_ts
}
health_FRA$'Mortality rate, infant (per 1,000 live births)' <- infant_mortality_kalma
n
plotNA.imputations(infant_mortality_ts, infant_mortality_kalman, main = "Infant morta
lity FRA")

```

Infant mortality FRA

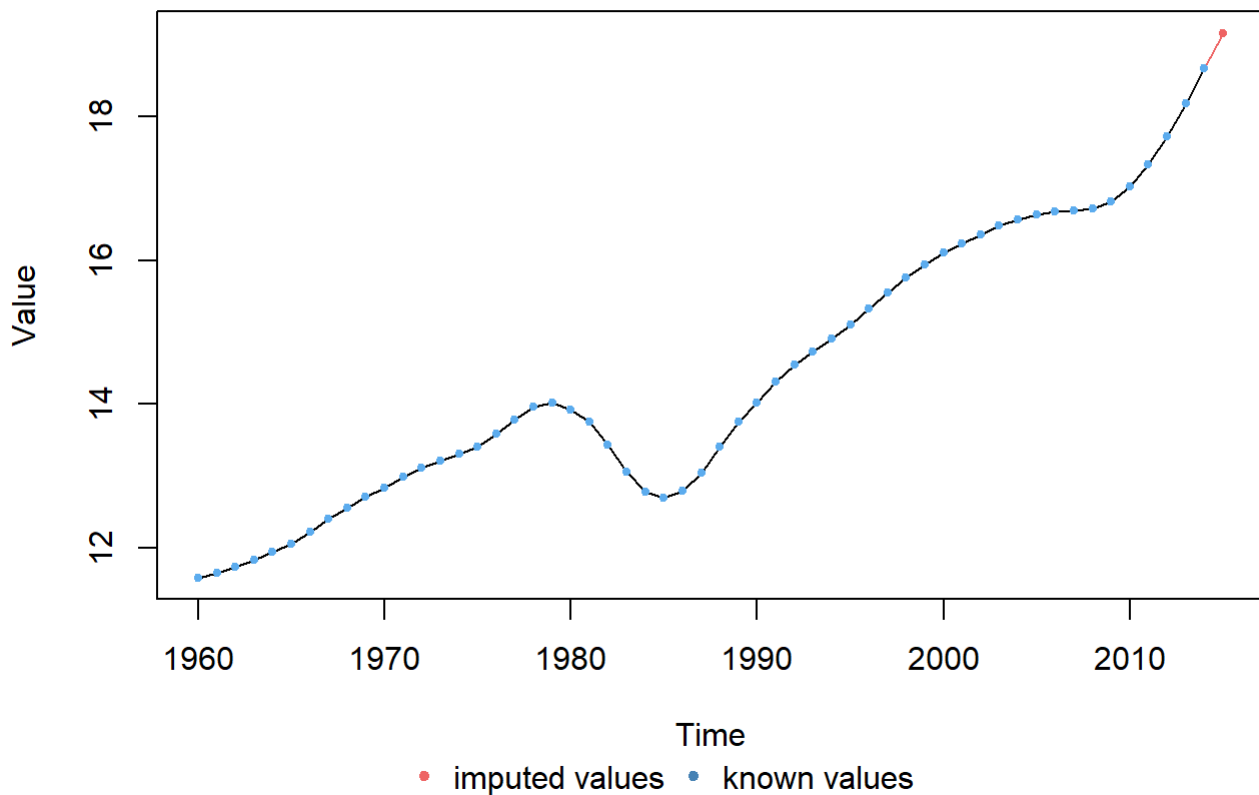


```

#Population over 65
population_over65 <- sqldf('select Year, Value from FRA_indicators where IndicatorName == "Population ages 65 and above (% of total)"')
population_over65_na <- rep(NA, 56)
i <- 1
for (year in population_over65$Year){
  population_over65_na[(year-1960+1)] <- population_over65$Value[i]
  i = i + 1
}
population_over65_ts <- ts(data = population_over65_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(population_over65_ts)){
  population_over65_kalman <- na.kalman(population_over65_ts)
} else{
  population_over65_kalman <- population_over65_ts
}
health_FRA$'Population ages 65 and above (% of total)' <- population_over65_kalman
plotNA.imputations(population_over65_ts, population_over65_kalman, main = "Population over 65 FRA")

```

Population over 65 FRA

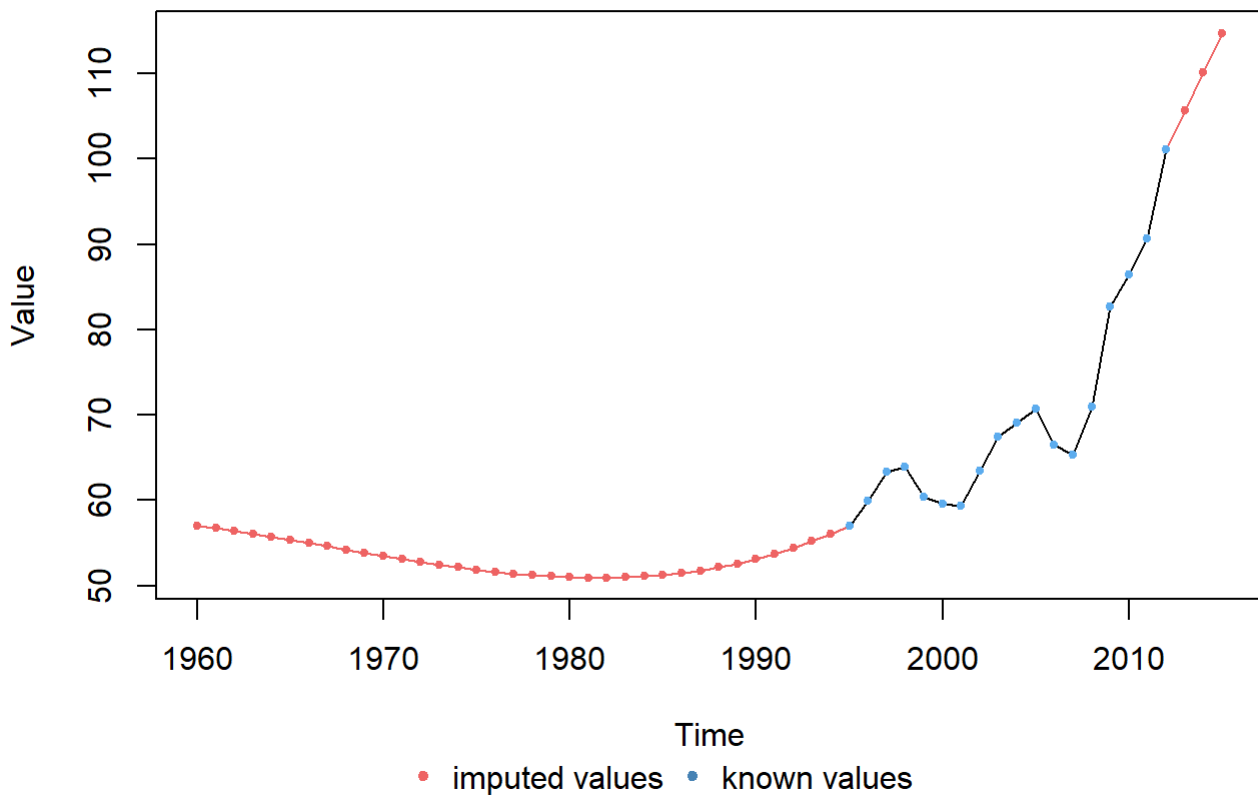


```

#Government debt, total % of GDP
government_debt <- sqldf('select Year, Value from FRA_indicators where IndicatorName
  == "Central government debt, total (% of GDP)"')
government_debt_na <- rep(NA, 56)
i <- 1
for (year in government_debt$Year){
  government_debt_na[(year-1960+1)] <- government_debt$Value[i]
  i = i + 1
}
government_debt_ts <- ts(data = government_debt_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(government_debt_ts)){
  government_debt_kalman <- na.kalman(government_debt_ts)
} else{
  government_debt_kalman <- government_debt_ts
}
health_FRA$'Central government debt, total (% of GDP)' <- government_debt_kalman
plotNA.imputations(government_debt_ts, government_debt_kalman, main = "Government deb
t % of GDP FRA")

```

Government debt % of GDP FRA

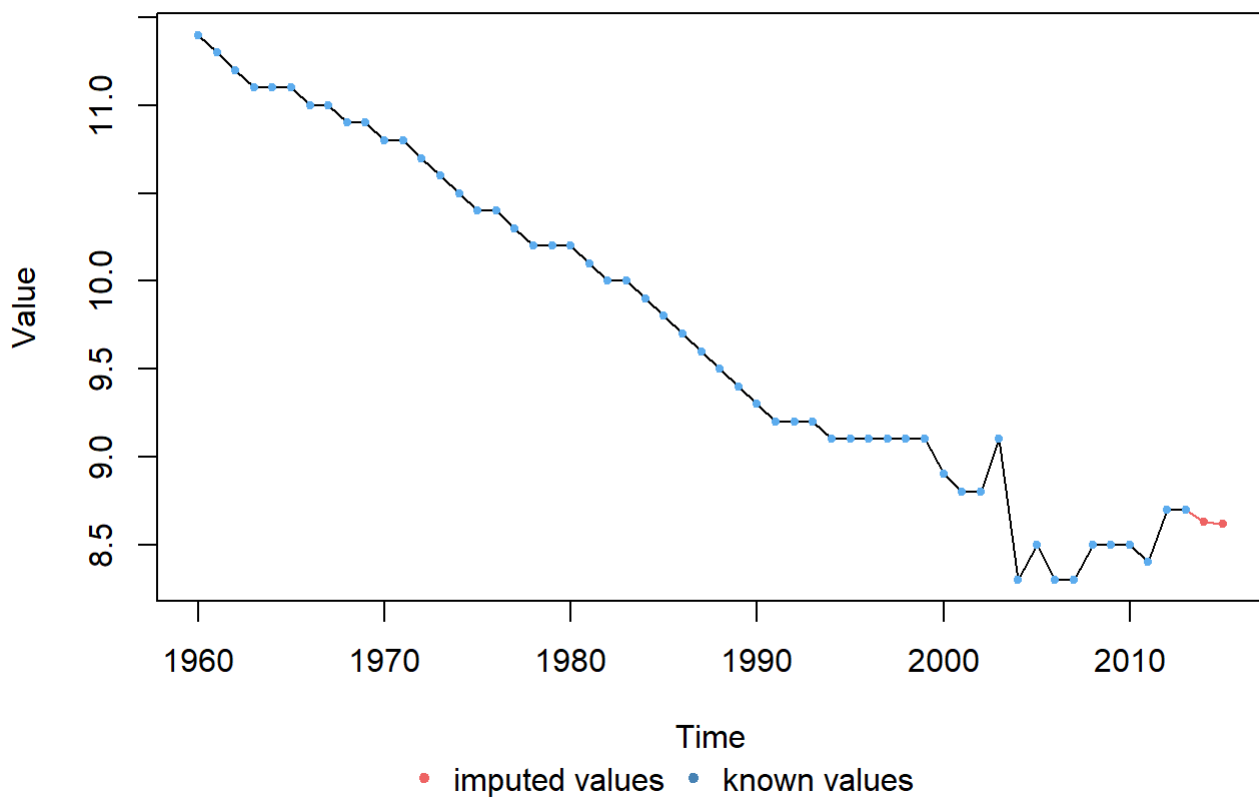


```

#Death rate per 1000 people
death_rate <- sqldf('select Year, Value from FRA_indicators where IndicatorName == "D
eath rate, crude (per 1,000 people)"')
death_rate_na <- rep(NA, 56)
i <- 1
for (year in death_rate$Year){
  death_rate_na[(year-1960+1)] <- death_rate$Value[i]
  i = i + 1
}
death_rate_ts <- ts(data = death_rate_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(death_rate_ts)){
  death_rate_kalman <- na.kalman(death_rate_ts)
} else{
  death_rate_kalman <- death_rate_ts
}
health_FRA$'Death rate, crude (per 1,000 people)' <- death_rate_kalman
plotNA.imputations(death_rate_ts, death_rate_kalman, main = "Death rate FRA")

```

Death rate FRA

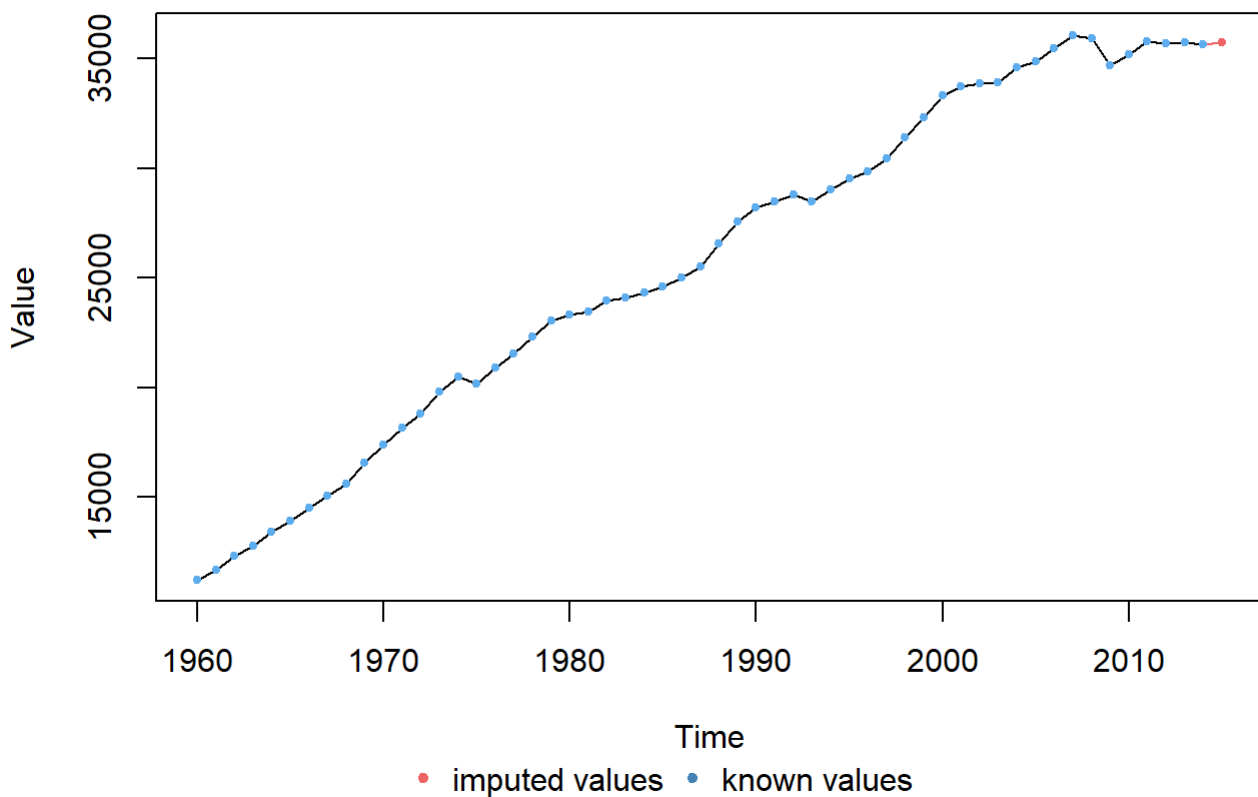



```

#GDP per capita (constant 2005 US$)
GDP <- sqldf('select Year, Value from FRA_indicators where IndicatorName == "GDP per
  capita (constant 2005 US$)"')
GDP_na <- rep(NA, 56)
i <- 1
for (year in GDP$Year){
  GDP_na[(year-1960+1)] <- GDP$Value[i]
  i = i + 1
}
GDP_ts <- ts(data = GDP_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(GDP_ts)){
  GDP_kalman <- na.kalman(GDP_ts)
} else{
  GDP_kalman <- GDP_ts
}
health_FRA$'GDP per capita (constant 2005 US$)' <- GDP_kalman
plotNA.imputations(GDP_ts, GDP_kalman, main = "GDP per capita (constant 2005 US$)")

```

GDP per capita (constant 2005 US\$))

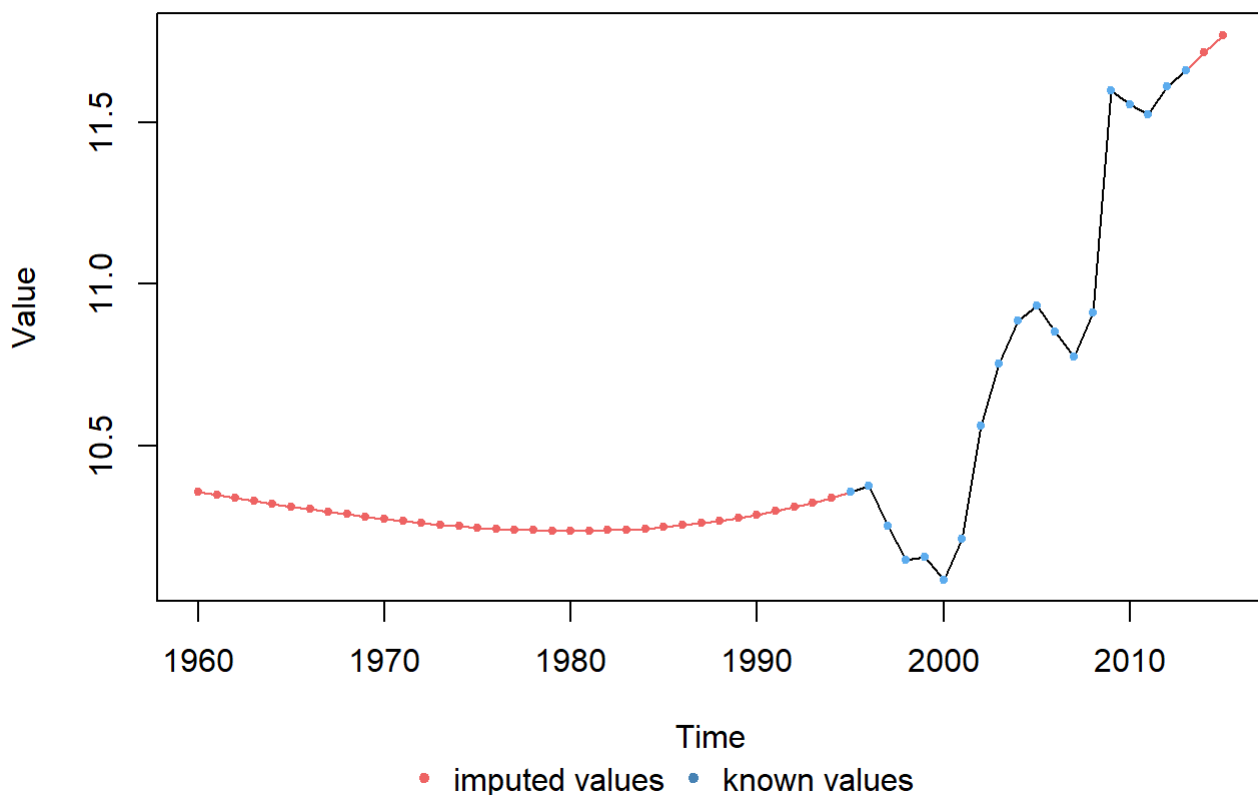


```

#health expenditure, total % of GDP
health_expenditure <- sqldf('select Year, Value from FRA_indicators where IndicatorName == "Health expenditure, total (% of GDP)"')
health_expenditure_na <- rep(NA, 56)
i <- 1
for (year in health_expenditure$Year){
  health_expenditure_na[(year-1960+1)] <- health_expenditure$Value[i]
  i = i + 1
}
health_expenditure_ts <- ts(data = health_expenditure_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(health_expenditure_ts)){
  health_expenditure_kalman <- na.kalman(health_expenditure_ts)
} else{
  health_expenditure_kalman <- health_expenditure_ts
}
health_FRA$'Health expenditure, total (% of GDP)' <- health_expenditure_kalman
plotNA.imputations(health_expenditure_ts, health_expenditure_kalman, main = "Health expenditure % of GDP FRA")

```

Health expenditure % of GDP FRA



For France, we don't have enough values for Government debt(% of GDP) and Health expenditure(% of GDP) again but we have some more values for Hospital beds so we may use this predictor in this case.

```
health_FRA <- health_FRA[, -c(9, 12)]
```

USA

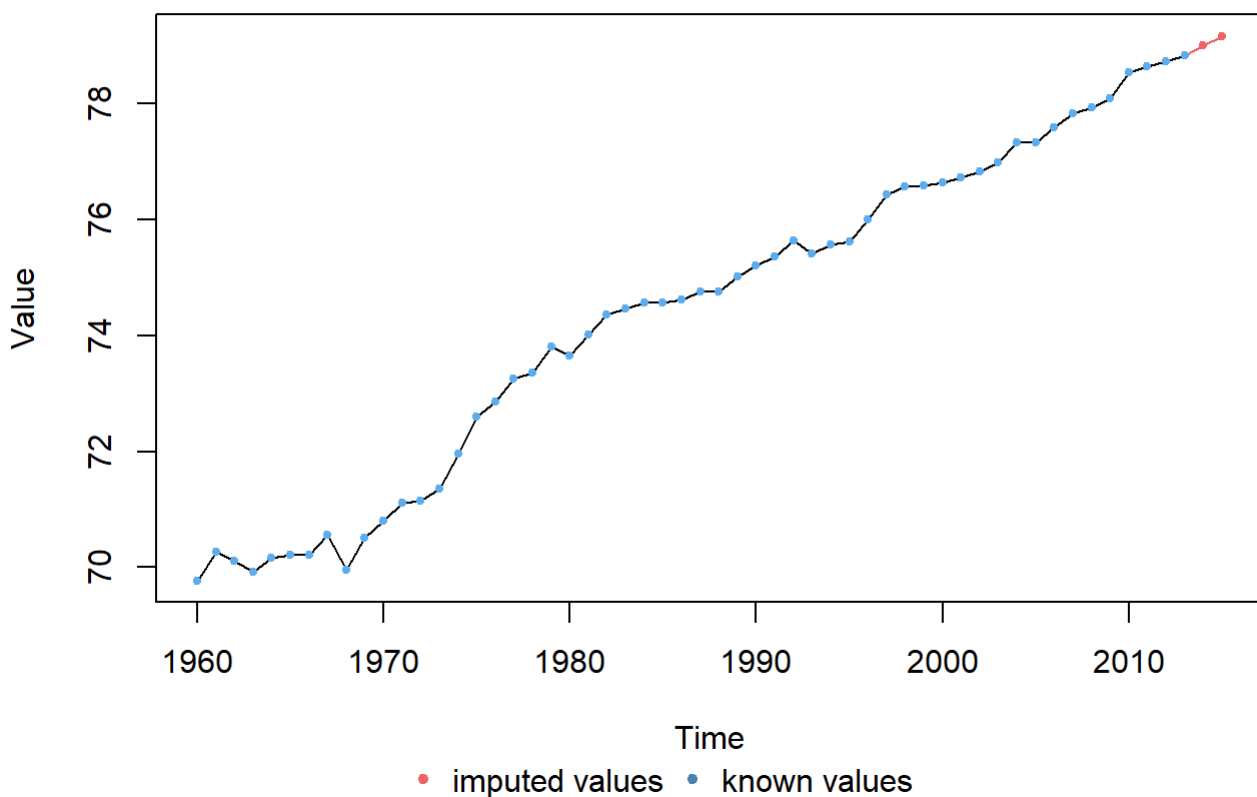
```

year <- sqldf('select distinct Year from world_indicators')
health_USA <- data.frame(year)

#Life expectancy years
life_expectancy <- sqldf('select Year, Value from USA_indicators where IndicatorName
== "Life expectancy at birth, total (years)"')
life_expectancy_na <- rep(NA, 56)
i <- 1
for (year in life_expectancy$Year){
  life_expectancy_na[(year-1960+1)] <- life_expectancy$Value[i]
  i = i + 1
}
life_expectancy_ts <- ts(data = life_expectancy_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(life_expectancy_ts)){
  life_expectancy_kalman <- na.kalman(life_expectancy_ts)
} else{
  life_expectancy_kalman <- life_expectancy_ts
}
health_USA$'Life expectancy at birth, total (years)' <- life_expectancy_kalman
plotNA.imputations(life_expectancy_ts, life_expectancy_kalman, main = "Life expectanc
y years USA")

```

Life expectancy years USA

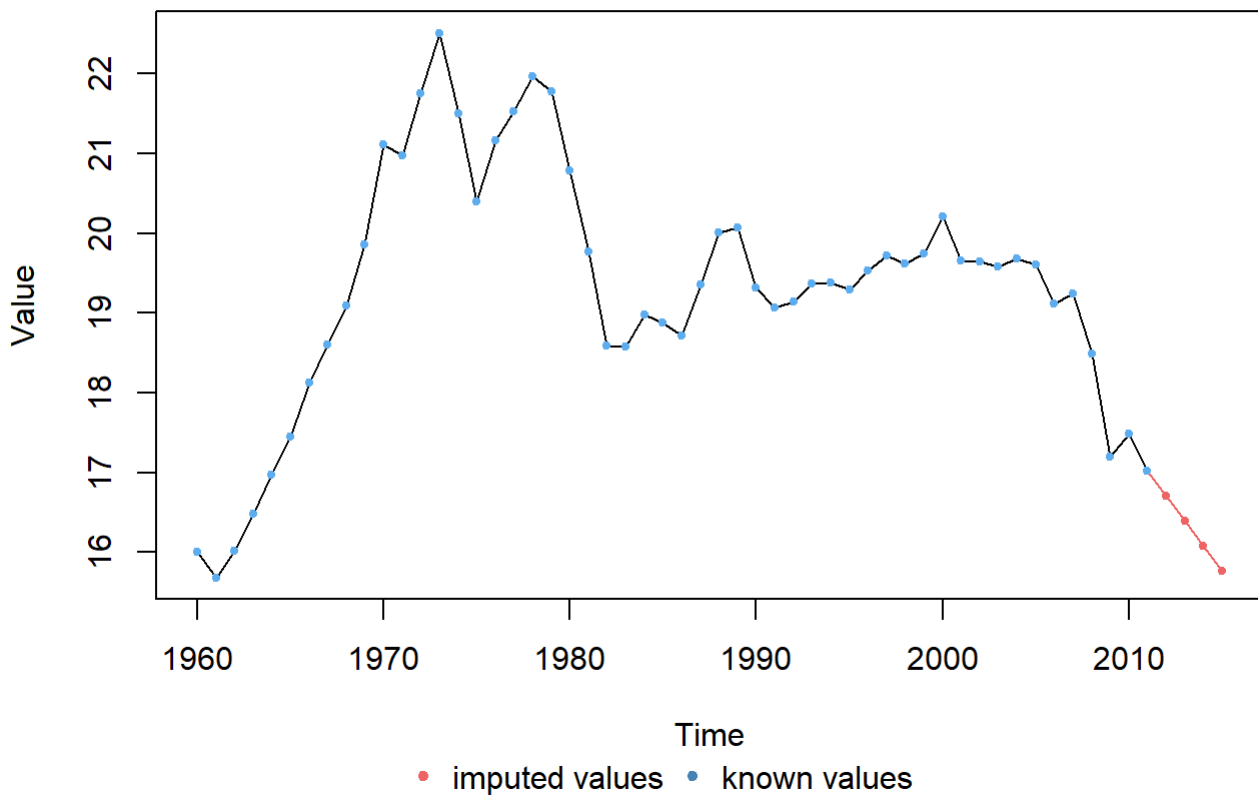


```

#CO2 emissions metrics tons per capita
CO2_emissions <- sqldf('select Year, Value from USA_indicators where IndicatorName ==
  "CO2 emissions (metric tons per capita)"')
CO2_emissions_na <- rep(NA, 56)
i <- 1
for (year in CO2_emissions$Year){
  CO2_emissions_na[(year-1960+1)] <- CO2_emissions$Value[i]
  i = i + 1
}
CO2_emissions_ts <- ts(data = CO2_emissions_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(CO2_emissions_ts)){
  CO2_emissions_kalman <- na.kalman(CO2_emissions_ts)
} else{
  CO2_emissions_kalman <- CO2_emissions_ts
}
health_USA$'CO2 emissions (metric tons per capita)' <- CO2_emissions_kalman
plotNA.imputations(CO2_emissions_ts, CO2_emissions_kalman, main = "CO2 emissions USA"
)

```

CO2 emissions USA

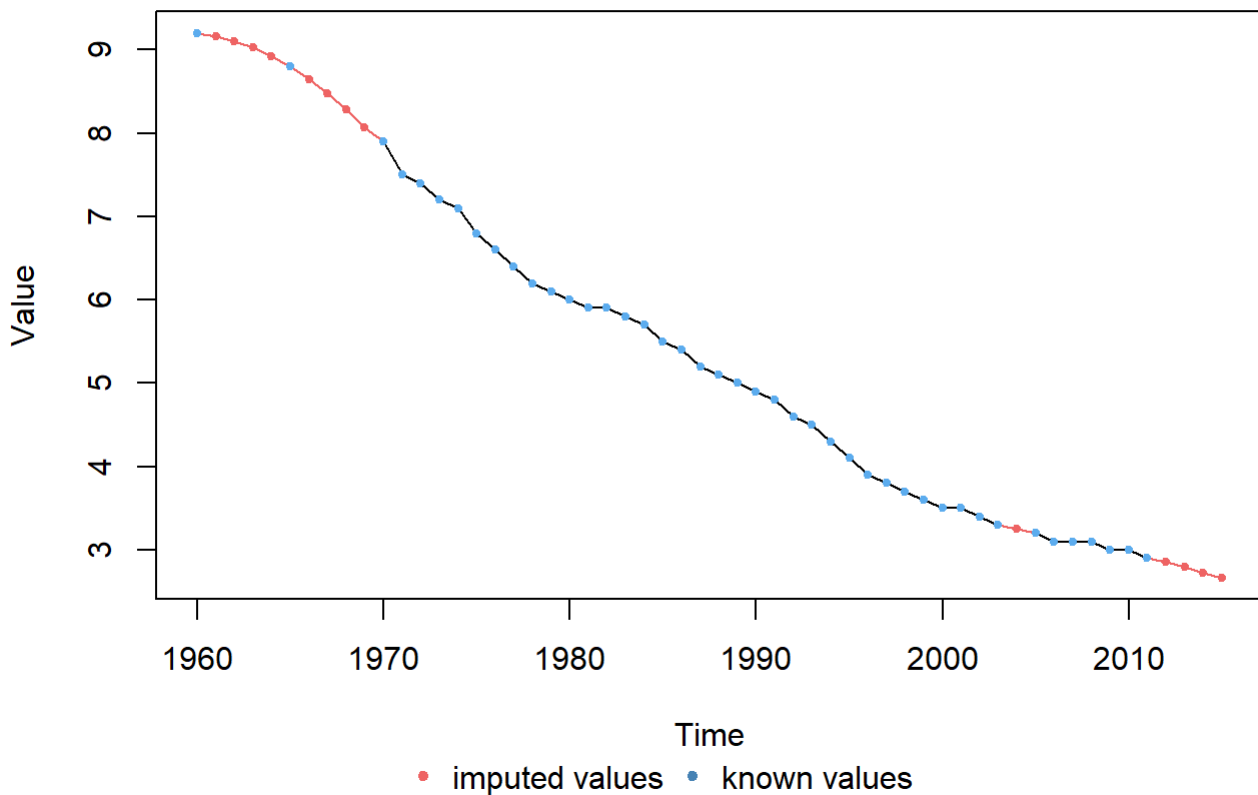


```

#Hospital beds per 1000 people
hospital_beds <- sqldf('select Year, Value from USA_indicators where IndicatorName ==
  "Hospital beds (per 1,000 people)"')
hospital_beds_na <- rep(NA, 56)
i <- 1
for (year in hospital_beds$Year){
  hospital_beds_na[(year-1960+1)] <- hospital_beds$Value[i]
  i = i + 1
}
hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency =
  1)
if(anyNA(hospital_beds_ts)){
  hospital_beds_kalman <- na.kalman(hospital_beds_ts)
} else{
  hospital_beds_kalman <- hospital_beds_ts
}
health_USA$'Hospital beds (per 1,000 people)' <- hospital_beds_kalman
plotNA.imputations(hospital_beds_ts, hospital_beds_kalman, main = "Hospital beds USA"
)

```

Hospital beds USA



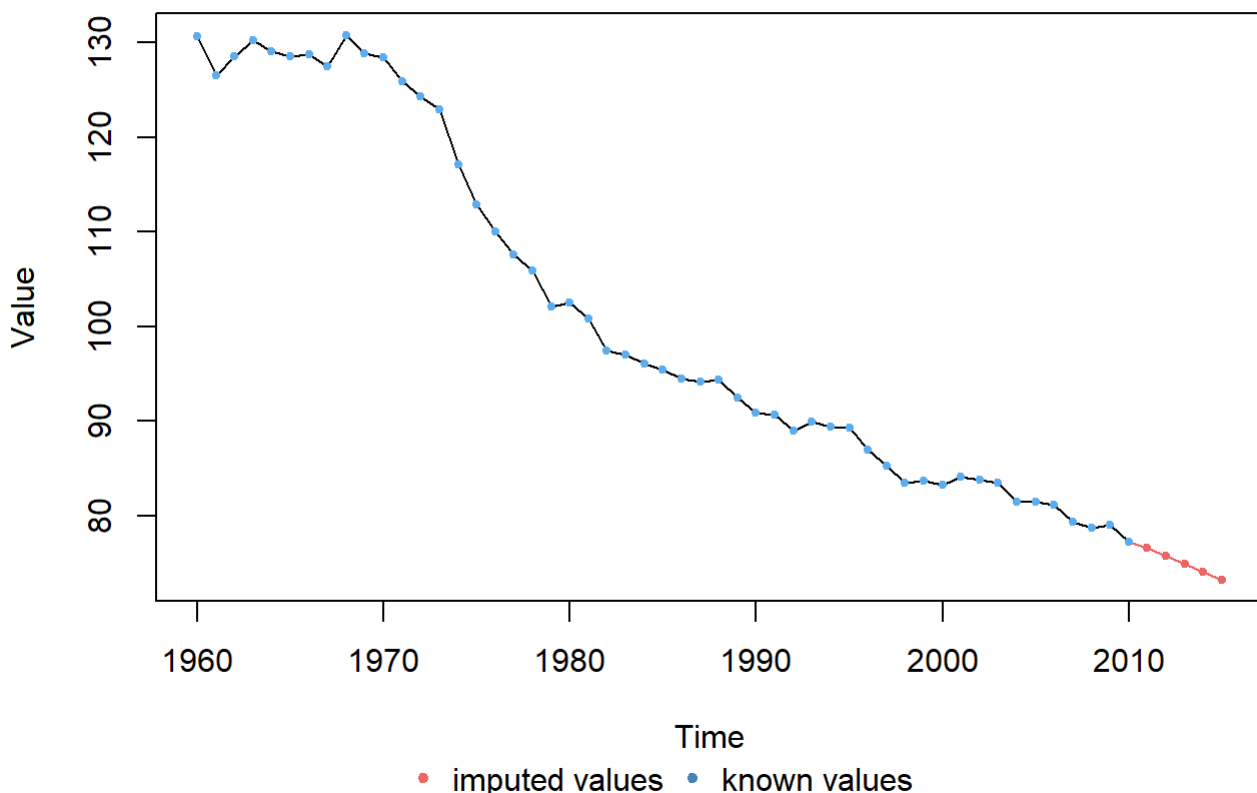
```

# hospital_beds_ts <- ts(data = hospital_beds_na, start = 1960, end = 2015, frequency
= 1)
# hospital_beds_linear <- na.interpolation(hospital_beds_ts)
# hospital_beds_spline <- na.interpolation(hospital_beds_ts, option = "spline")
# hospital_beds_kalman <- na.kalman(hospital_beds_ts)
# plotNA.imputations(hospital_beds_ts, hospital_beds_linear)
# plotNA.imputations(hospital_beds_ts, hospital_beds_spline)
# plotNA.imputations(hospital_beds_ts, hospital_beds_kalman)

#Female mortality per 1000 females
female_mortality <- sqldf('select Year, Value from USA_indicators where IndicatorName
== "Mortality rate, adult, female (per 1,000 female adults)"')
female_mortality_na <- rep(NA, 56)
i <- 1
for (year in female_mortality$Year){
  female_mortality_na[(year-1960+1)] <- female_mortality$Value[i]
  i = i + 1
}
female_mortality_ts <- ts(data = female_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(female_mortality_ts)){
  female_mortality_kalman <- na.kalman(female_mortality_ts)
} else{
  female_mortality_kalman <- female_mortality_ts
}
health_USA$'Mortality rate, adult, female (per 1,000 female adults)' <- female_mortal
ity_kalman
plotNA.imputations(female_mortality_ts, female_mortality_kalman, main = "Female morta
lity USA")

```

Female mortality USA

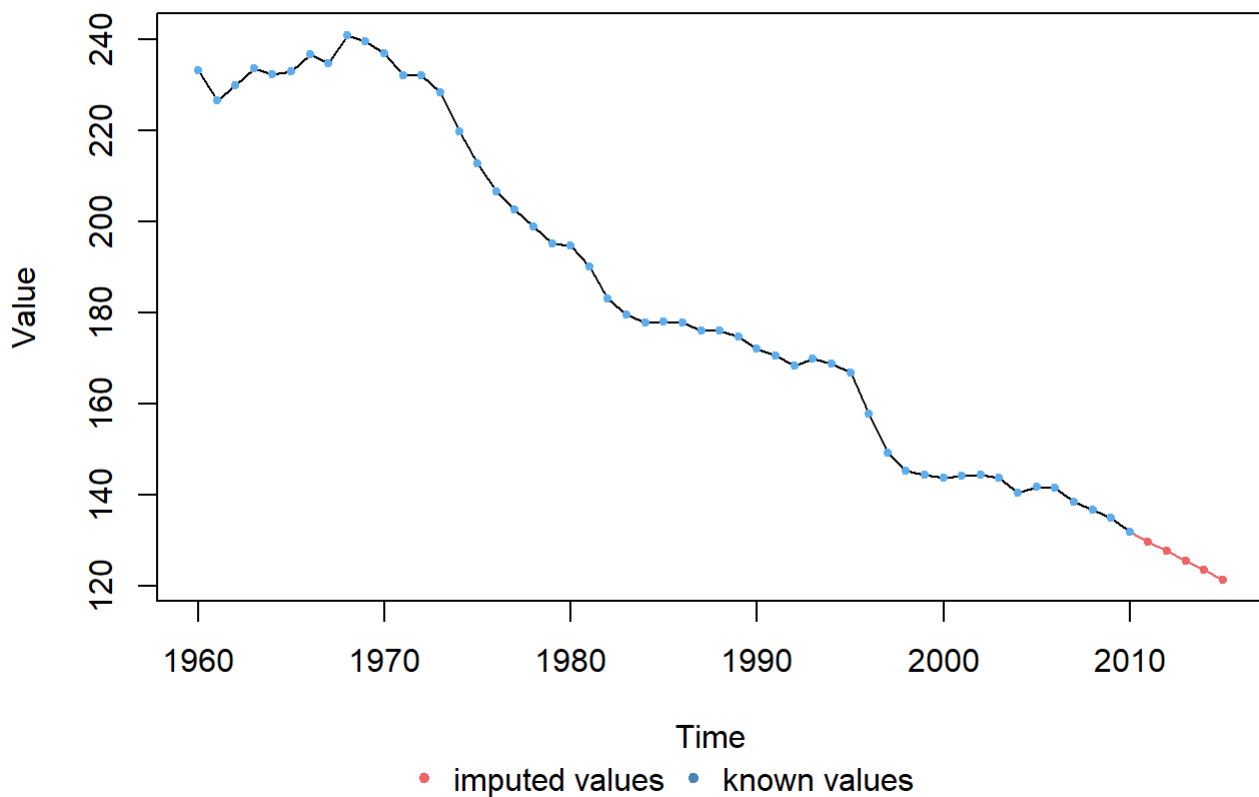


```

#Male mortality per 1000 males
male_mortality <- sqldf('select Year, Value from USA_indicators where IndicatorName =
= "Mortality rate, adult, male (per 1,000 male adults)"')
male_mortality_na <- rep(NA, 56)
i <- 1
for (year in male_mortality$Year){
  male_mortality_na[(year-1960+1)] <- male_mortality$Value[i]
  i = i + 1
}
male_mortality_ts <- ts(data = male_mortality_na, start = 1960, end = 2015, frequency
= 1)
if(anyNA(male_mortality_ts)){
  male_mortality_kalman <- na.kalman(male_mortality_ts)
} else{
  male_mortality_kalman <- male_mortality_ts
}
health_USA$'Mortality rate, adult, male (per 1,000 male adults)' <- male_mortality_kalman
plotNA.imputations(male_mortality_ts, male_mortality_kalman, main = "Male mortality U
SA")

```

Male mortality USA

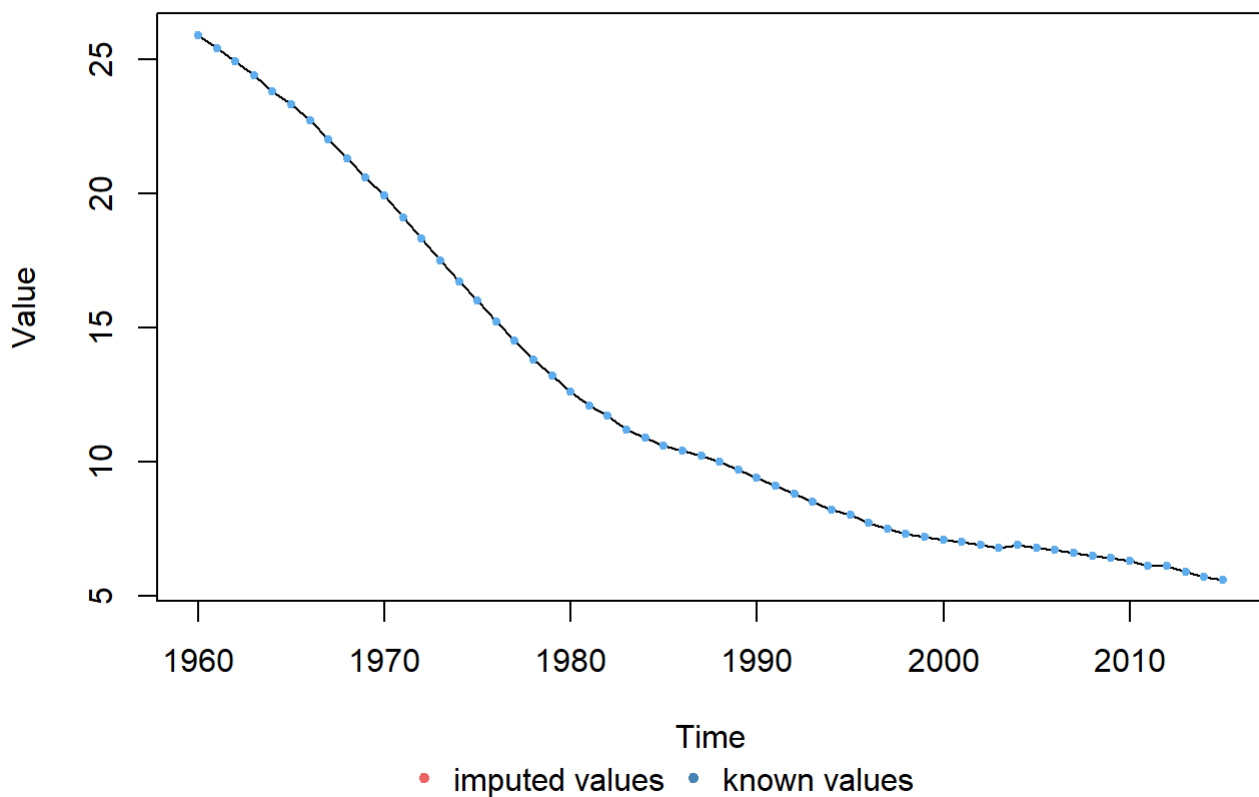


```

#Infant mortality per 1000 live births
infant_mortality <- sqldf('select Year, Value from USA_indicators where IndicatorName
== "Mortality rate, infant (per 1,000 live births)"')
infant_mortality_na <- rep(NA, 56)
i <- 1
for (year in infant_mortality$Year){
  infant_mortality_na[(year-1960+1)] <- infant_mortality$Value[i]
  i = i + 1
}
infant_mortality_ts <- ts(data = infant_mortality_na, start = 1960, end = 2015, frequ
ency = 1)
if(anyNA(infant_mortality_ts)){
  infant_mortality_kalman <- na.kalman(infant_mortality_ts)
} else{
  infant_mortality_kalman <- infant_mortality_ts
}
health_USA$'Mortality rate, infant (per 1,000 live births)' <- infant_mortality_kalma
n
plotNA.imputations(infant_mortality_ts, infant_mortality_kalman, main = "Infant morta
lity USA")

```

Infant mortality USA

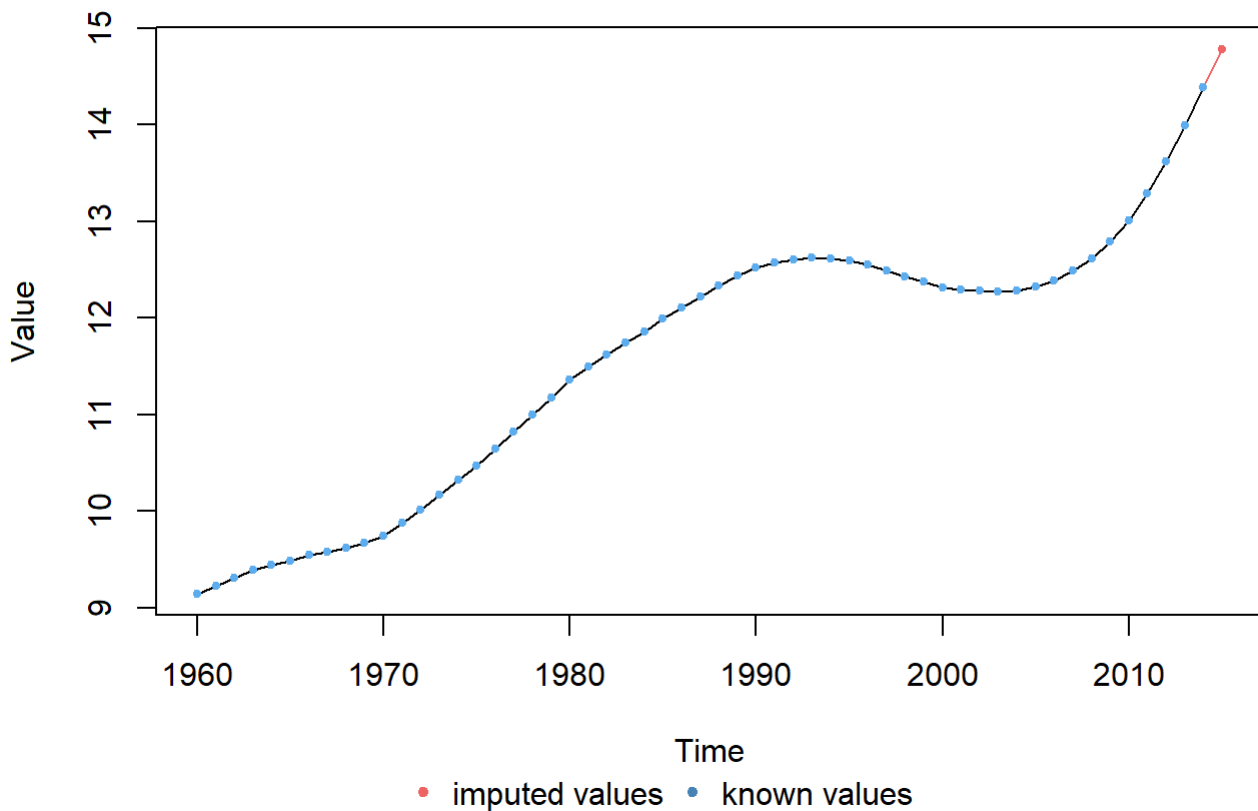



```

#Population over 65
population_over65 <- sqldf('select Year, Value from USA_indicators where IndicatorName == "Population ages 65 and above (% of total)"')
population_over65_na <- rep(NA, 56)
i <- 1
for (year in population_over65$Year){
  population_over65_na[(year-1960+1)] <- population_over65$Value[i]
  i = i + 1
}
population_over65_ts <- ts(data = population_over65_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(population_over65_ts)){
  population_over65_kalman <- na.kalman(population_over65_ts)
} else{
  population_over65_kalman <- population_over65_ts
}
health_USA$'Population ages 65 and above (% of total)' <- population_over65_kalman
plotNA.imputations(population_over65_ts, population_over65_kalman, main = "Population over 65 USA")

```

Population over 65 USA

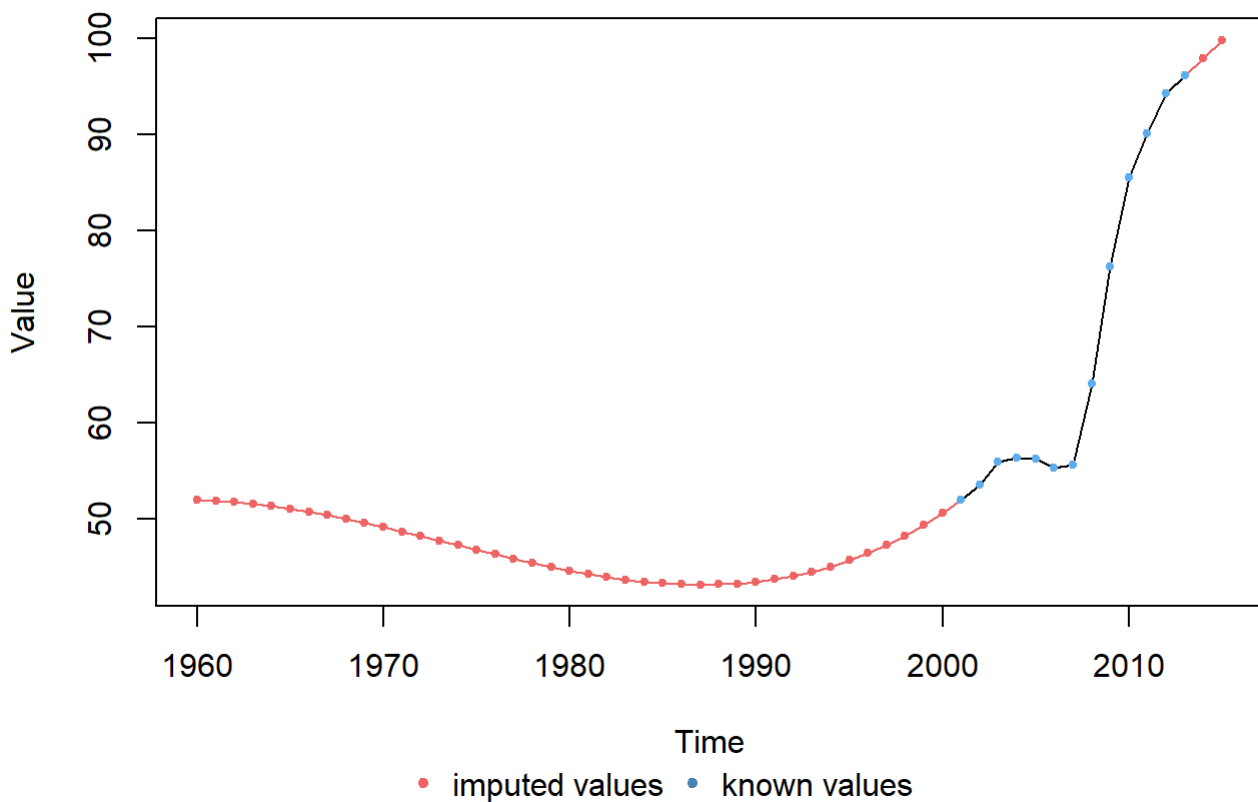


```

#Government debt, total % of GDP
government_debt <- sqldf('select Year, Value from USA_indicators where IndicatorName
  == "Central government debt, total (% of GDP)"')
government_debt_na <- rep(NA, 56)
i <- 1
for (year in government_debt$Year){
  government_debt_na[(year-1960+1)] <- government_debt$Value[i]
  i = i + 1
}
government_debt_ts <- ts(data = government_debt_na, start = 1960, end = 2015, frequen
cy = 1)
if(anyNA(government_debt_ts)){
  government_debt_kalman <- na.kalman(government_debt_ts)
} else{
  government_debt_kalman <- government_debt_ts
}
health_USA$'Central government debt, total (% of GDP)' <- government_debt_kalman
plotNA.imputations(government_debt_ts, government_debt_kalman, main = "Government deb
t % of GDP USA")

```

Government debt % of GDP USA

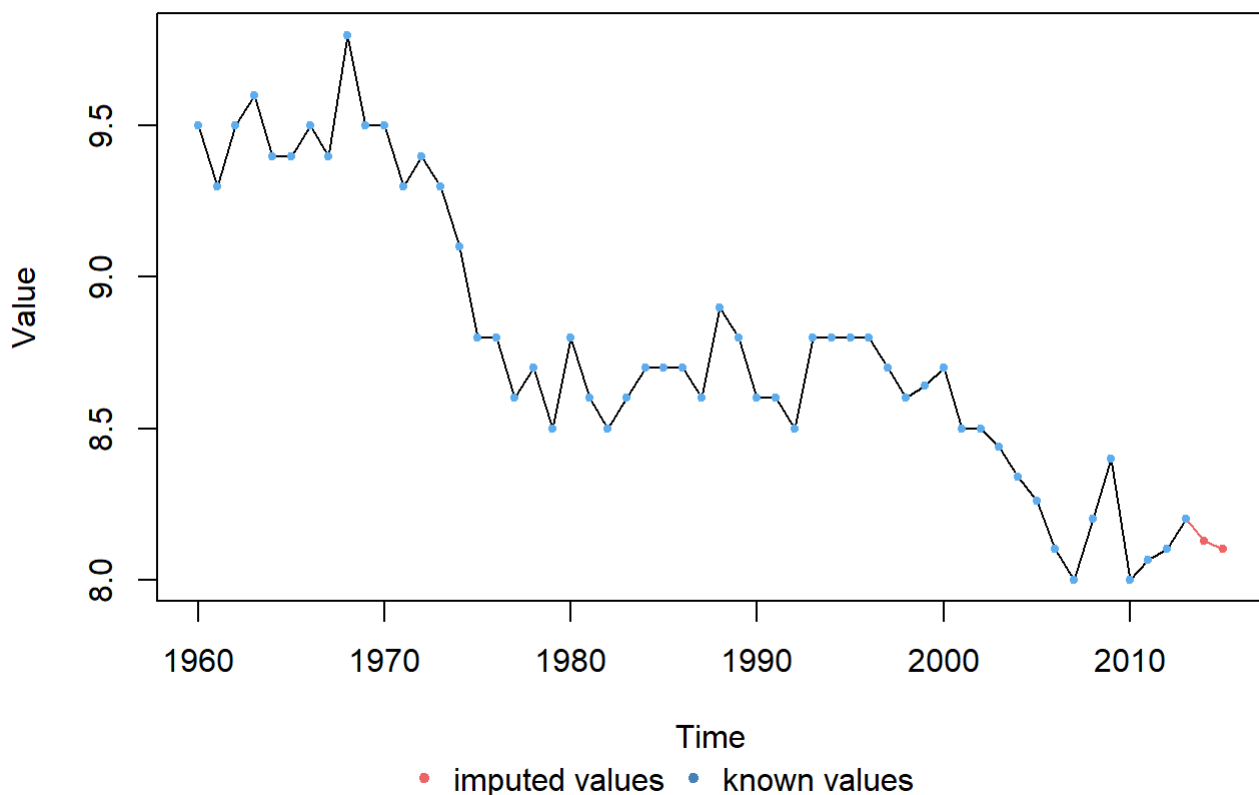


```

#Death rate per 1000 people
death_rate <- sqldf('select Year, Value from USA_indicators where IndicatorName == "D
eath rate, crude (per 1,000 people)"')
death_rate_na <- rep(NA, 56)
i <- 1
for (year in death_rate$Year){
  death_rate_na[(year-1960+1)] <- death_rate$Value[i]
  i = i + 1
}
death_rate_ts <- ts(data = death_rate_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(death_rate_ts)){
  death_rate_kalman <- na.kalman(death_rate_ts)
} else{
  death_rate_kalman <- death_rate_ts
}
health_USA$'Death rate, crude (per 1,000 people)' <- death_rate_kalman
plotNA.imputations(death_rate_ts, death_rate_kalman, main = "Death rate USA")

```

Death rate USA

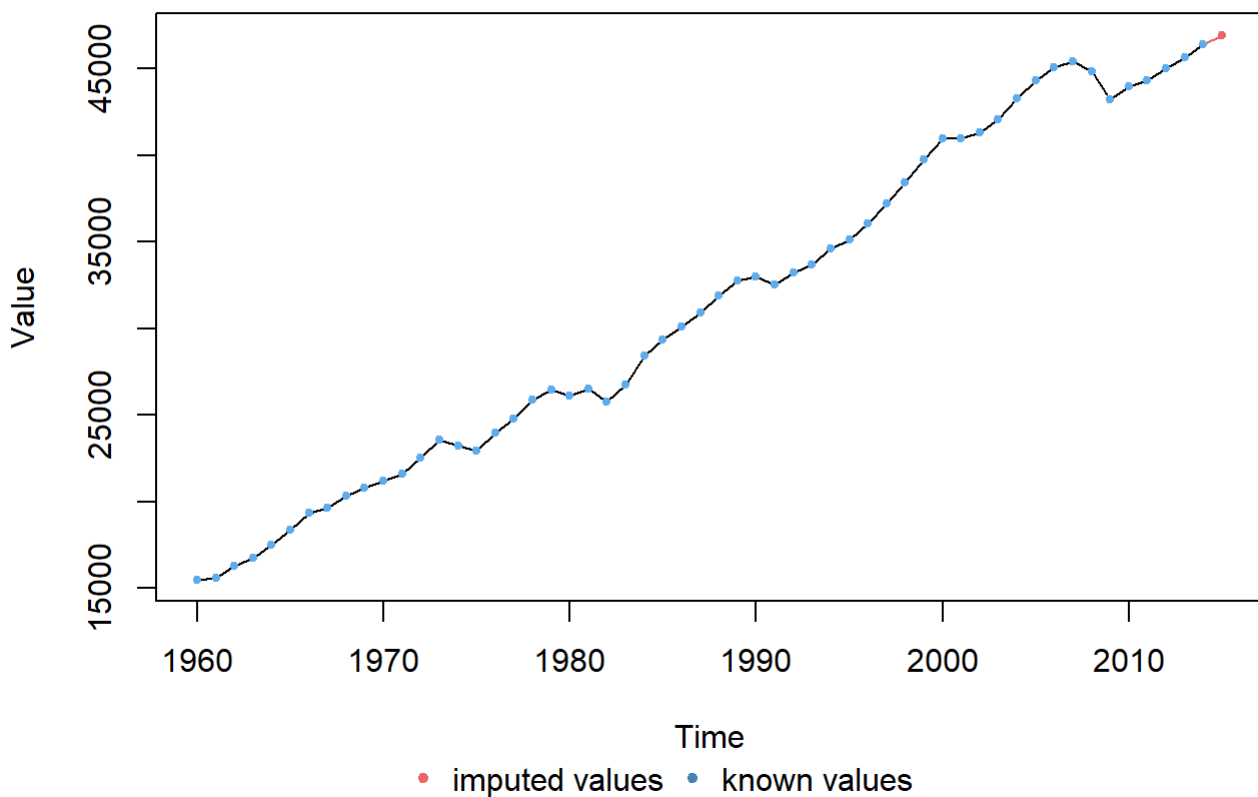


```

#GDP per capita (constant 2005 US$)
GDP <- sqldf('select Year, Value from USA_indicators where IndicatorName == "GDP per
  capita (constant 2005 US$)"')
GDP_na <- rep(NA, 56)
i <- 1
for (year in GDP$Year){
  GDP_na[(year-1960+1)] <- GDP$Value[i]
  i = i + 1
}
GDP_ts <- ts(data = GDP_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(GDP_ts)){
  GDP_kalman <- na.kalman(GDP_ts)
} else{
  GDP_kalman <- GDP_ts
}
health_USA$'GDP per capita (constant 2005 US$)' <- GDP_kalman
plotNA.imputations(GDP_ts, GDP_kalman, main = "GDP per capita (constant 2005 US$)")

```

GDP per capita (constant 2005 US\$)

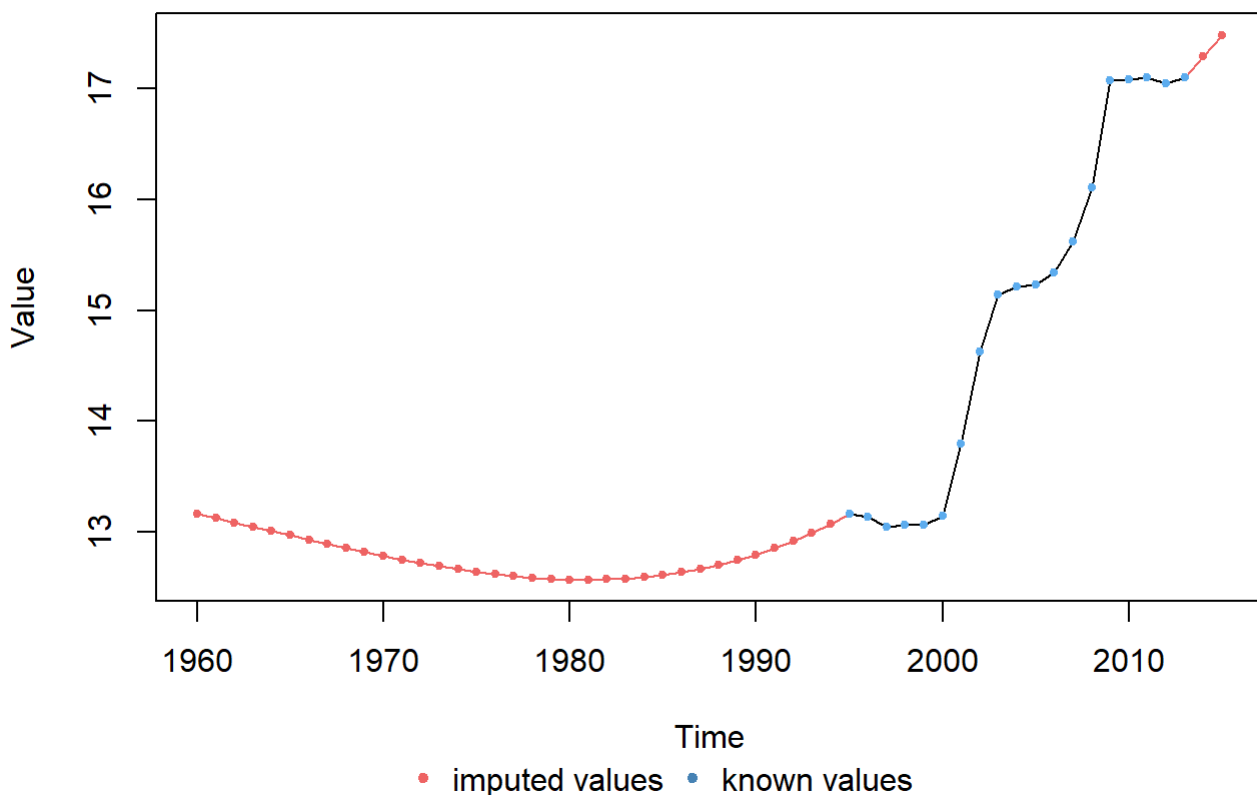


```

#health expenditure, total % of GDP
health_expenditure <- sqldf('select Year, Value from USA_indicators where IndicatorName == "Health expenditure, total (% of GDP)"')
health_expenditure_na <- rep(NA, 56)
i <- 1
for (year in health_expenditure$Year){
  health_expenditure_na[(year-1960+1)] <- health_expenditure$Value[i]
  i = i + 1
}
health_expenditure_ts <- ts(data = health_expenditure_na, start = 1960, end = 2015, frequency = 1)
if(anyNA(health_expenditure_ts)){
  health_expenditure_kalman <- na.kalman(health_expenditure_ts)
} else{
  health_expenditure_kalman <- health_expenditure_ts
}
health_USA$'Health expenditure, total (% of GDP)' <- health_expenditure_kalman
plotNA.imputations(health_expenditure_ts, health_expenditure_kalman, main = "Health expenditure % of GDP USA")

```

Health expenditure % of GDP USA



For USA, same as for France, we don't have enough values for Government debt(% of GDP) and Health expenditure(% of GDP) again but we have some more values for Hospital beds so we may use this predictor in this case.

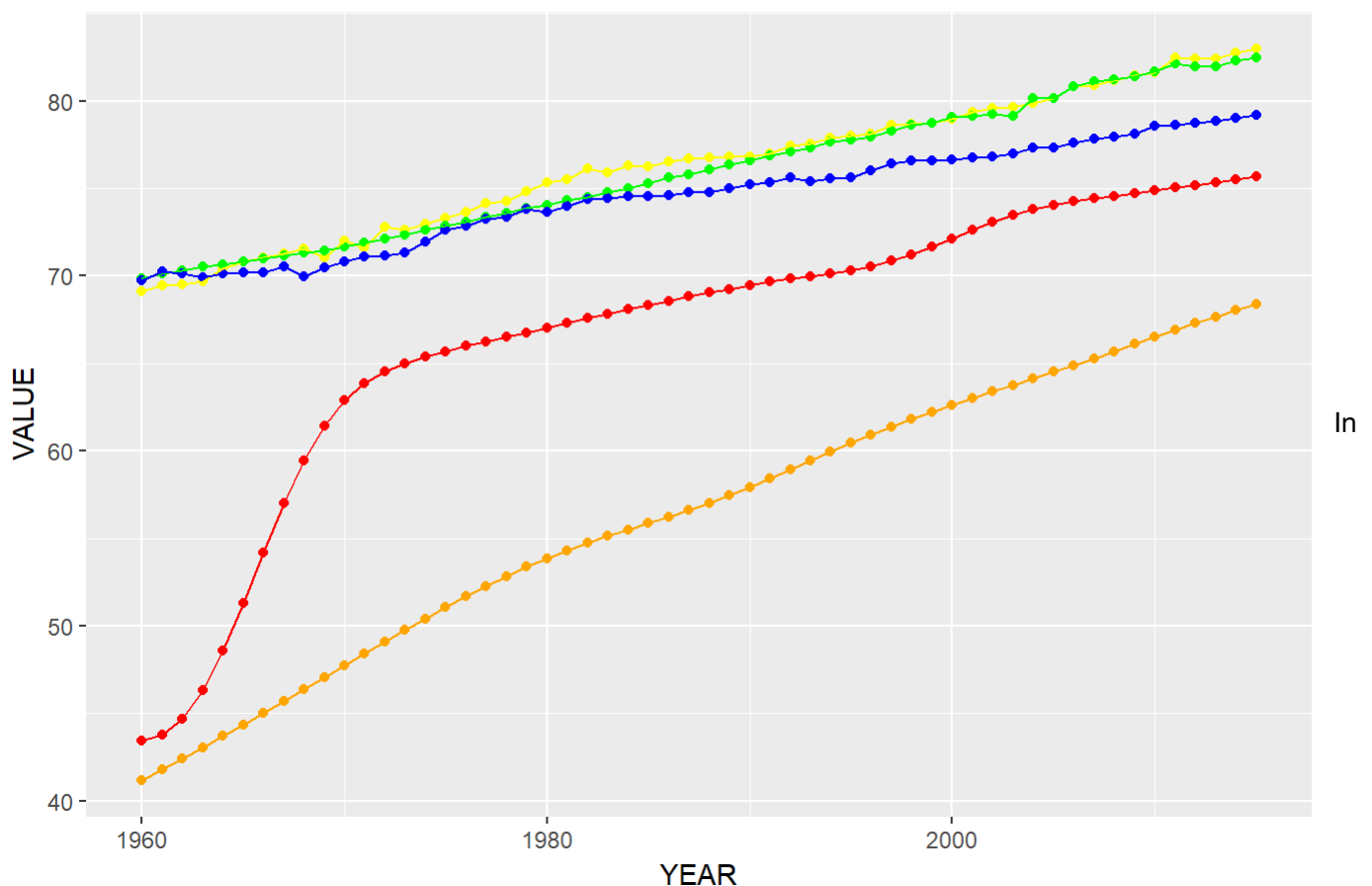
```
health_USA <- health_USA[, -c(9, 12)]
```

Life expectancy

```
ggplot() +
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Life expectancy at birth, total (years)`), color = "yellow") + geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Life expectancy at birth, total (years)`), color = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Life expectancy at birth, total (years)`), color = "orange") + geom_point(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Life expectancy at birth, total (years)`), color = "green") + geom_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Life expectancy at birth, total (years)`), color = "blue") + geom_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Life expectancy at birth, total (years)`), color = "yellow") + geom_line(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Life expectancy at birth, total (years)`), color = "red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Life expectancy at birth, total (years)`), color = "orange") + geom_line(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Life expectancy at birth, total (years)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Life expectancy at birth, total (years)`), color = "blue") + ggtitle("Life expectancy at birth (years)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

Life expectancy at birth (years)



this plot we can see the trend that life expectancy has had from 1960 to 2015 for all the countries. The trend for Spain, USA and France are quite similar but life expectancy in Spain and France is over 80 year while in USA it is not. Then in China, life expectancy grew exponentially in the 1960s, from less than 45 years to more than 60, to follow a linear growing trend after that with a more significant slope what has made life expectancy in China to be over 75 years and get closer to the values for the previous countries. Finally, India's life

expectancy was lower than China's in the 1960s but it didn't had an exponential growth, it has follow a linear growth over all the years and, although its sloper is the biggest, India's life expectancy is the lowest of all the countries we are considering and it is below 70 years.

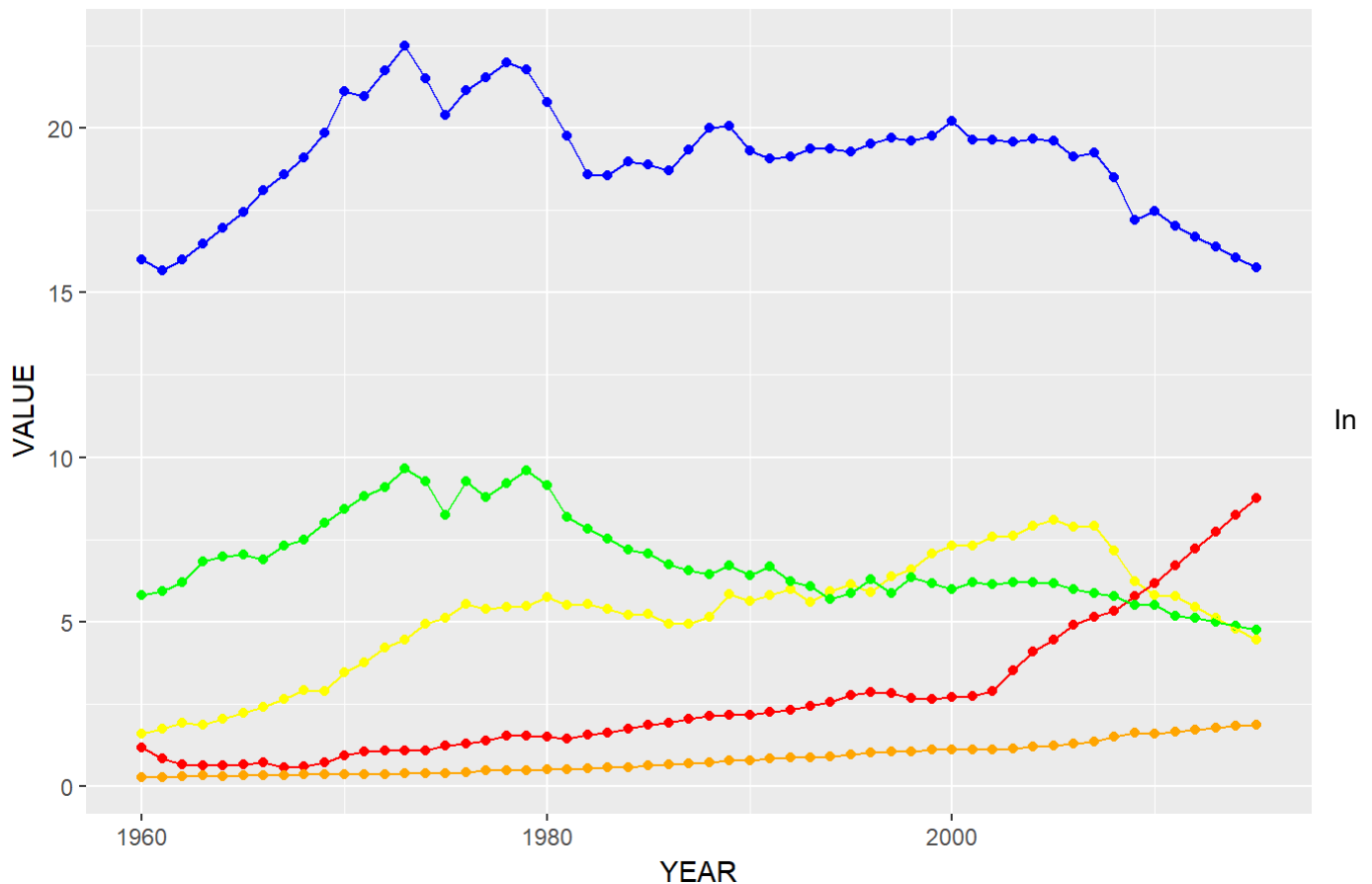
In the plots for the next indicators, we want to see if these trends in life expectancy are somehow related to the trends of these others indicators.

CO2 emissions

```
ggplot() +  
geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`CO2 emissions  
(metric tons per capita)`), color = "yellow") + geom_point(data = health_CHN, aes(x =  
health_CHN$Year, y = health_CHN$`CO2 emissions (metric tons per capita)`), color =  
"red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`CO2 e  
missions (metric tons per capita)`), color = "orange") + geom_point(data = health_FR  
A, aes(x = health_FRA$Year, y = health_FRA$`CO2 emissions (metric tons per capita)`),  
color = "green") + geom_point(data = health_USA, aes(x = health_USA$Year, y = health  
_USA$`CO2 emissions (metric tons per capita)`), color = "blue") + geom_line(data = he  
alth_ESP, aes(x = health_ESP$Year, y = health_ESP$`CO2 emissions (metric tons per cap  
ita)`), color = "yellow") + geom_line(data = health_CHN, aes(x = health_CHN$Year, y =  
health_CHN$`CO2 emissions (metric tons per capita)`), color = "red") + geom_line(dat  
a = health_IND, aes(x = health_IND$Year, y = health_IND$`CO2 emissions (metric tons p  
er capita)`), color = "orange") + geom_line(data = health_FRA, aes(x = health_FRA$Yea  
r, y = health_FRA$`CO2 emissions (metric tons per capita)`), color = "green") + geom_  
line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`CO2 emissions (metri  
c tons per capita)`), color = "blue") + ggtitle("CO2 emissions (metrics tons per capi  
ta)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_d  
iscrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to co  
ntinuous.
```

CO2 emissions (metrics tons per capita)



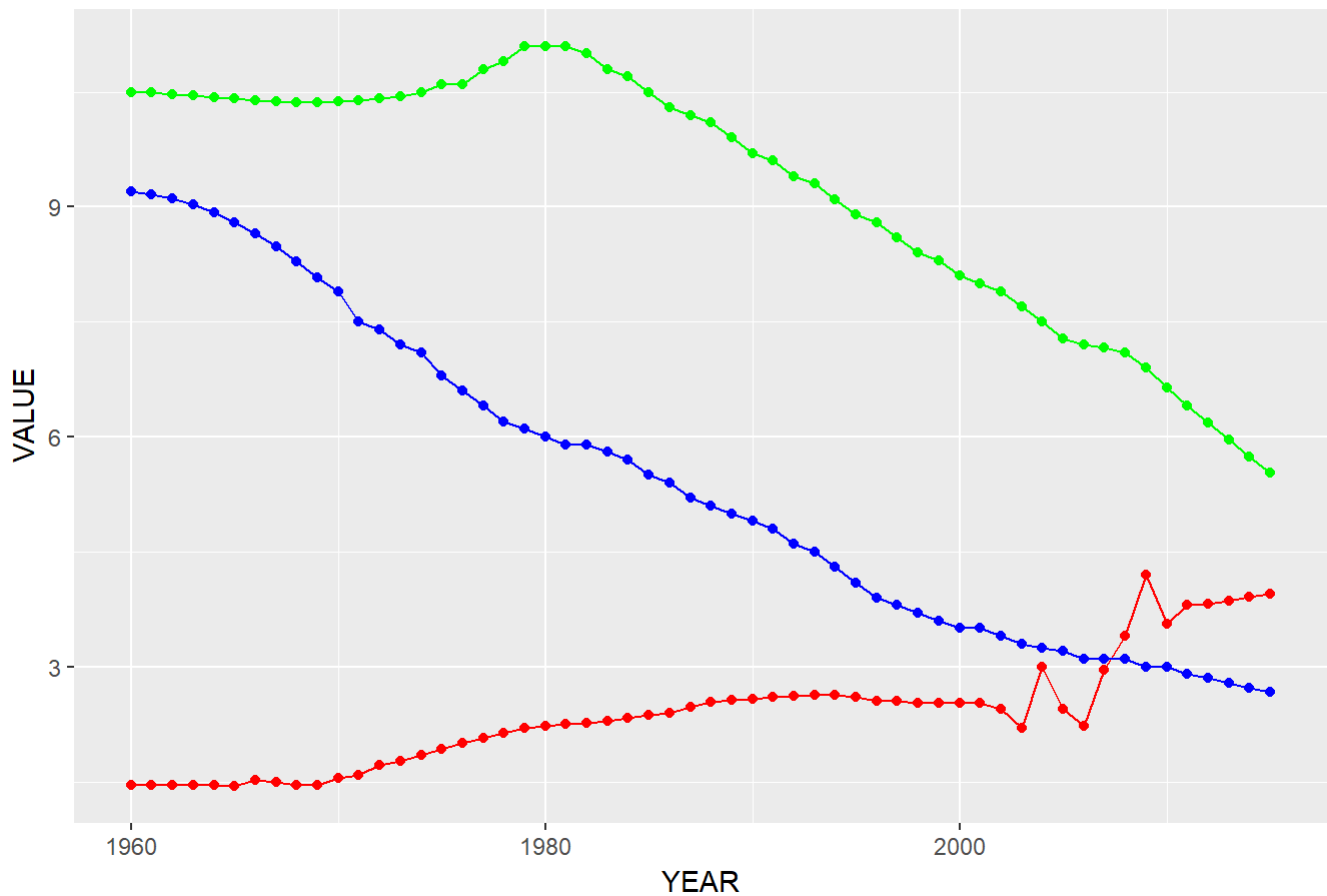
this plot the CO2 emission (metrics tons per capita) are shown. USA is by far the country with the most emissions. France was historically in second place but Spain caught up in the 1990s but they are both quite equal since 2010 and following a decreasing tendency. Both China and India had really low values in 1960 and had been following a slowly increasing tendency with China over India. Then, after 2000 China had a significant increment on its tendency having more emissions than France and Spain after 2010 and keeping this tendency. There doesn't seem to be a connection between these values of CO2 emissions and life expectancy since the two countries with the highest life expectancy, France and Spain, aren't the countries with more emissions or the ones with less emissions.

Hospital beds

```
ggplot() +
  geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Hospital beds
(per 1,000 people)`), color = "red") + geom_point(data = health_FRA, aes(x = health_
FRA$Year, y = health_FRA$`Hospital beds (per 1,000 people)`), color = "green") + geom_
_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Hospital beds (per
1,000 people)`), color = "blue") + geom_line(data = health_CHN, aes(x = health_CHN
$Year, y = health_CHN$`Hospital beds (per 1,000 people)`), color = "red") + geom_lin
e(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Hospital beds (per 1,00
0 people)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year,
y = health_USA$`Hospital beds (per 1,000 people)`), color = "blue") + ggtitle("Hospi
tal beds (per 1,000 people)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position
= "top") + scale_fill_discrete(name = "Country", labels = c("CHN", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to co
ntinuous.
```


Hospital beds (per 1,000 people)

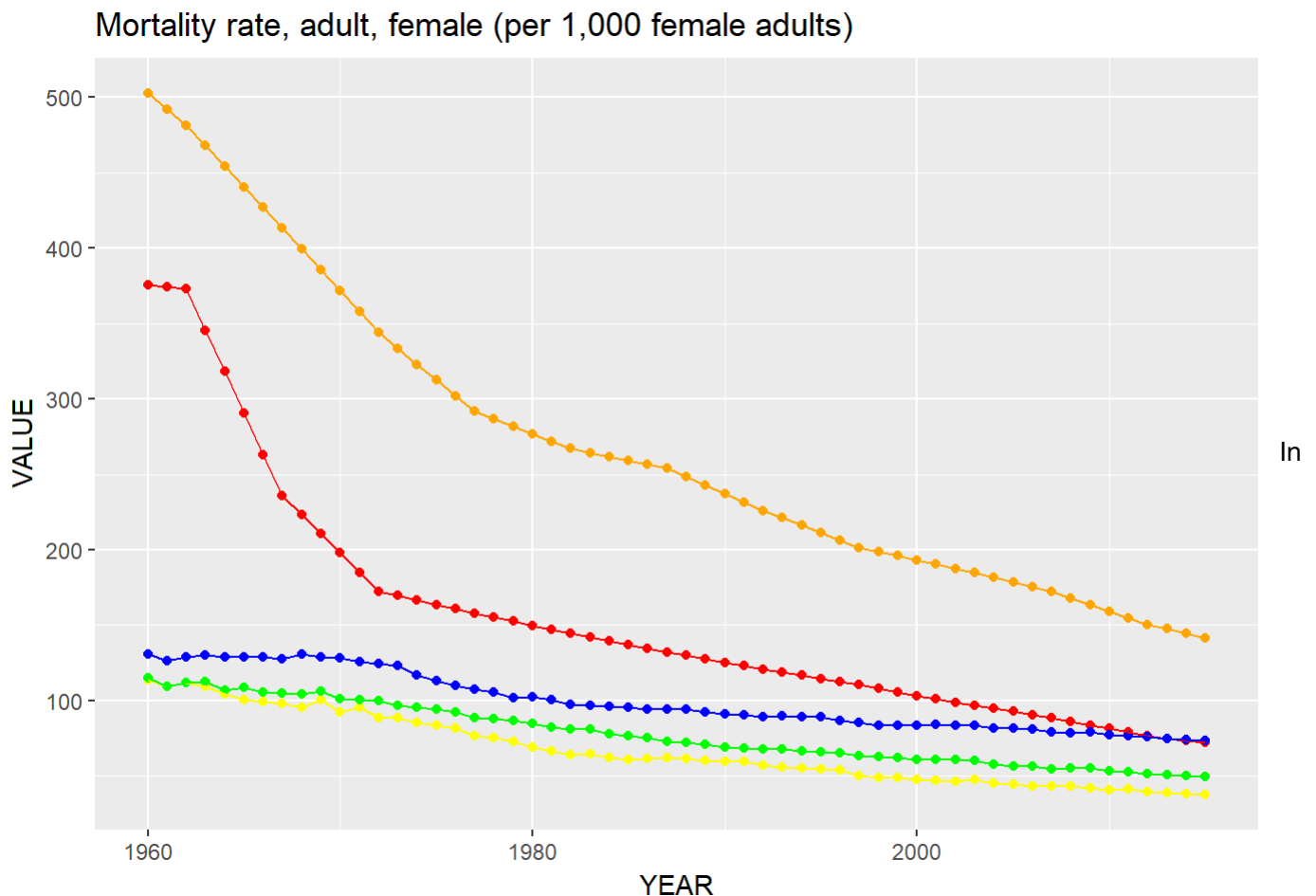


The number of hospital beds (per 1000 people) is shown in this plot. Only USA, France and China are represented because there weren't enough values for Spain and India. Both France and USA follow a decreasing tendency while, as we saw before, life expectancy was increasing in these countries. However, China has an increasing tendency of hospital beds and his life expectancy was increasing as well. France has the lowest hospital beds per 1000 people but it has the highest life expectancy so we may think that having more hospital beds doesn't imply a higher life expectancy.

Mortality rate, female

```
ggplot() +
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Mortality rate, adult, female (per 1,000 female adults)`), color = "yellow") + geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Mortality rate, adult, female (per 1,000 female adults)`), color = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Mortality rate, adult, female (per 1,000 female adults)`), color = "orange") + geom_point(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Mortality rate, adult, female (per 1,000 female adults)`), color = "green") + geom_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Mortality rate, adult, female (per 1,000 female adults)`), color = "blue") + geom_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Mortality rate, adult, female (per 1,000 female adults)`), color = "yellow") + geom_line(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Mortality rate, adult, female (per 1,000 female adults)`), color = "red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Mortality rate, adult, female (per 1,000 female adults)`), color = "orange") + geom_line(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Mortality rate, adult, female (per 1,000 female adults)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Mortality rate, adult, female (per 1,000 female adults)`), color = "blue") + ggtitle("Mortality rate, adult, female (per 1,000 female adults)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



this plot, the female mortality rate (per 1000 females) is shown. Spain, France and USA follow a similar decreasing trend, China had a huge decrease in the 1960s and then followed a similar trend to the previous

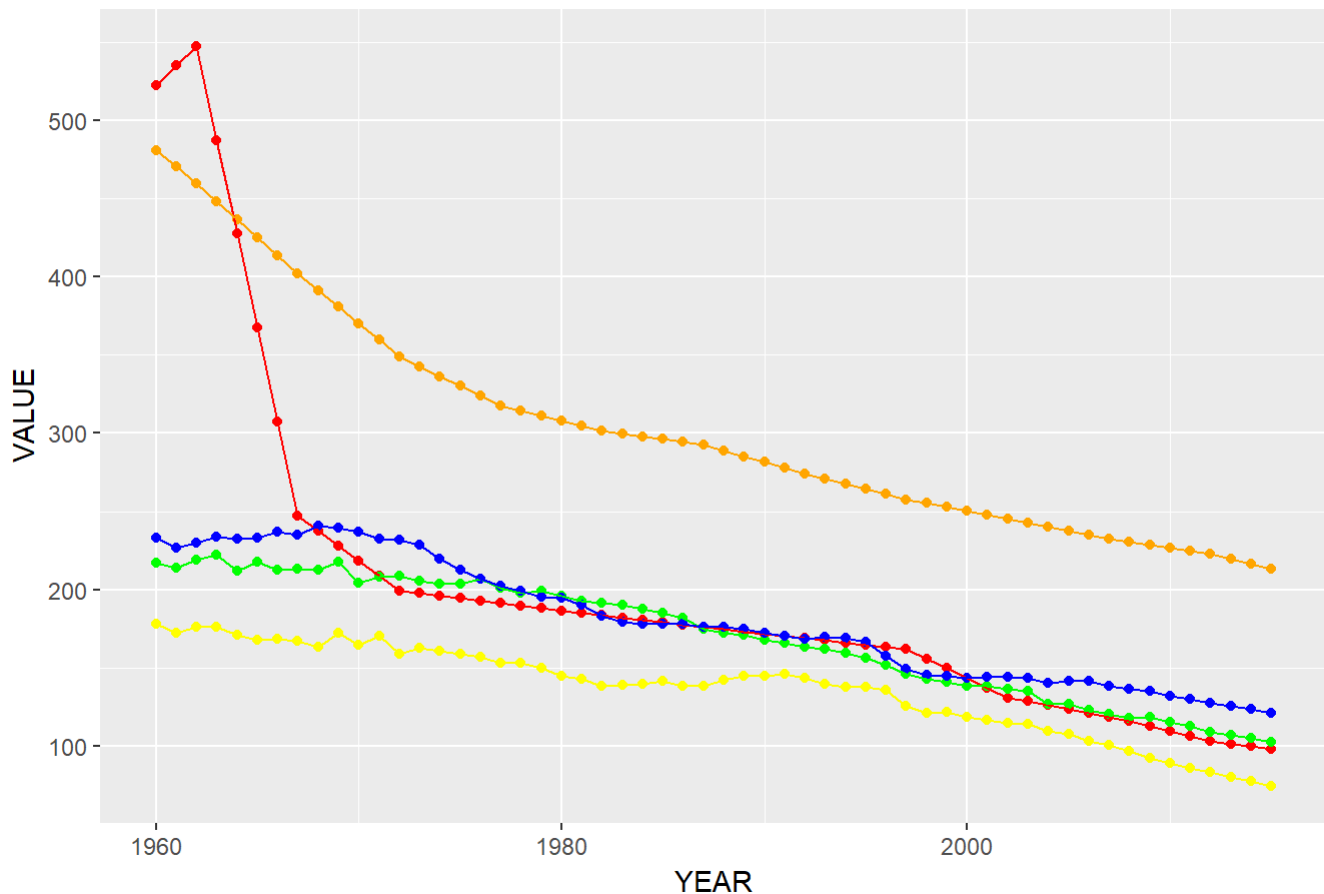
countries and it is at the same values now. India has had an important decrease as well but there is still a gap between this country and the others. In this case we can relate the mortality rate to the life expectancy since it seems that the countries with a higher life expectancy have a lower female mortality rate.

Mortality rate, male

```
ggplot() +  
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Mortality rate, adult, male (per 1,000 male adults)`), color = "yellow") + geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Mortality rate, adult, male (per 1,000 male adults)`), color = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Mortality rate, adult, male (per 1,000 male adults)`), color = "orange") + geom_point(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Mortality rate, adult, male (per 1,000 male adults)`), color = "green") + geom_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Mortality rate, adult, male (per 1,000 male adults)`), color = "blue") + geom_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Mortality rate, adult, male (per 1,000 male adults)`), color = "yellow") + geom_line(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Mortality rate, adult, male (per 1,000 male adults)`), color = "red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Mortality rate, adult, male (per 1,000 male adults)`), color = "orange") + geom_line(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Mortality rate, adult, male (per 1,000 male adults)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Mortality rate, adult, male (per 1,000 male adults)`), color = "blue") +  
  ggtitle("Mortality rate, adult, male (per 1,000 male adults)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

Mortality rate, adult, male (per 1,000 male adults)



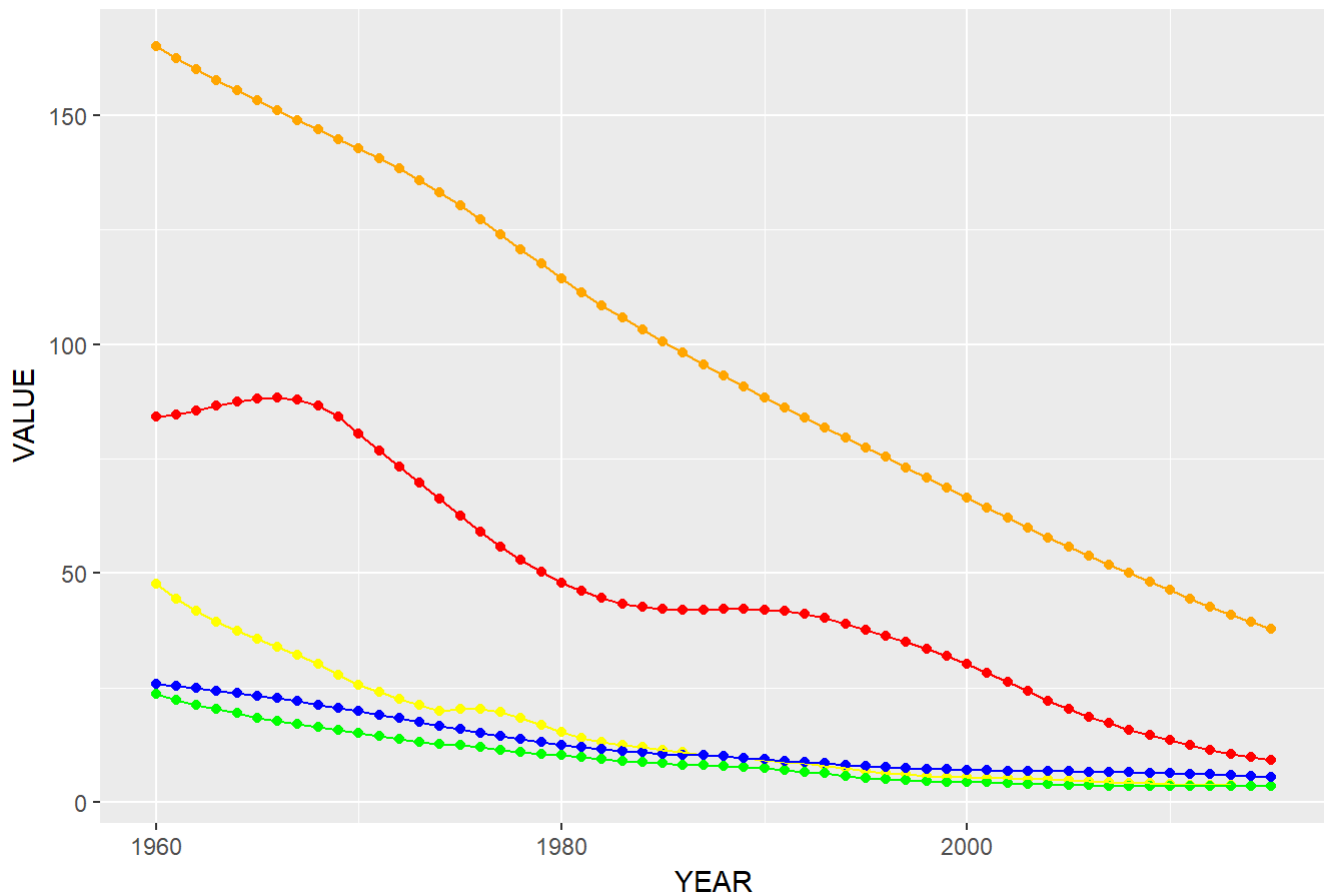
This plot is similar to the previous one but it represents the male mortality rate now. The trends are quite similar to the previous plot and we can see that the values of male mortality were higher in the past in all countries but they are quite the same nowadays.

Mortality rate, infant

```
ggplot() +
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Mortality rate, infant (per 1,000 live births)`), color = "yellow") + geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Mortality rate, infant (per 1,000 live births)`), color = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Mortality rate, infant (per 1,000 live births)`), color = "orange") + geom_point(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Mortality rate, infant (per 1,000 live births)`), color = "green") + geom_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Mortality rate, infant (per 1,000 live births)`), color = "blue") + geom_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Mortality rate, infant (per 1,000 live births)`), color = "yellow") + geom_line(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Mortality rate, infant (per 1,000 live births)`), color = "red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Mortality rate, infant (per 1,000 live births)`), color = "orange") + geom_line(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Mortality rate, infant (per 1,000 live births)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Mortality rate, infant (per 1,000 live births)`), color = "blue") + ggtitle("Mortality rate, infant (per 1,000 live births)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

Mortality rate, infant (per 1,000 live births)



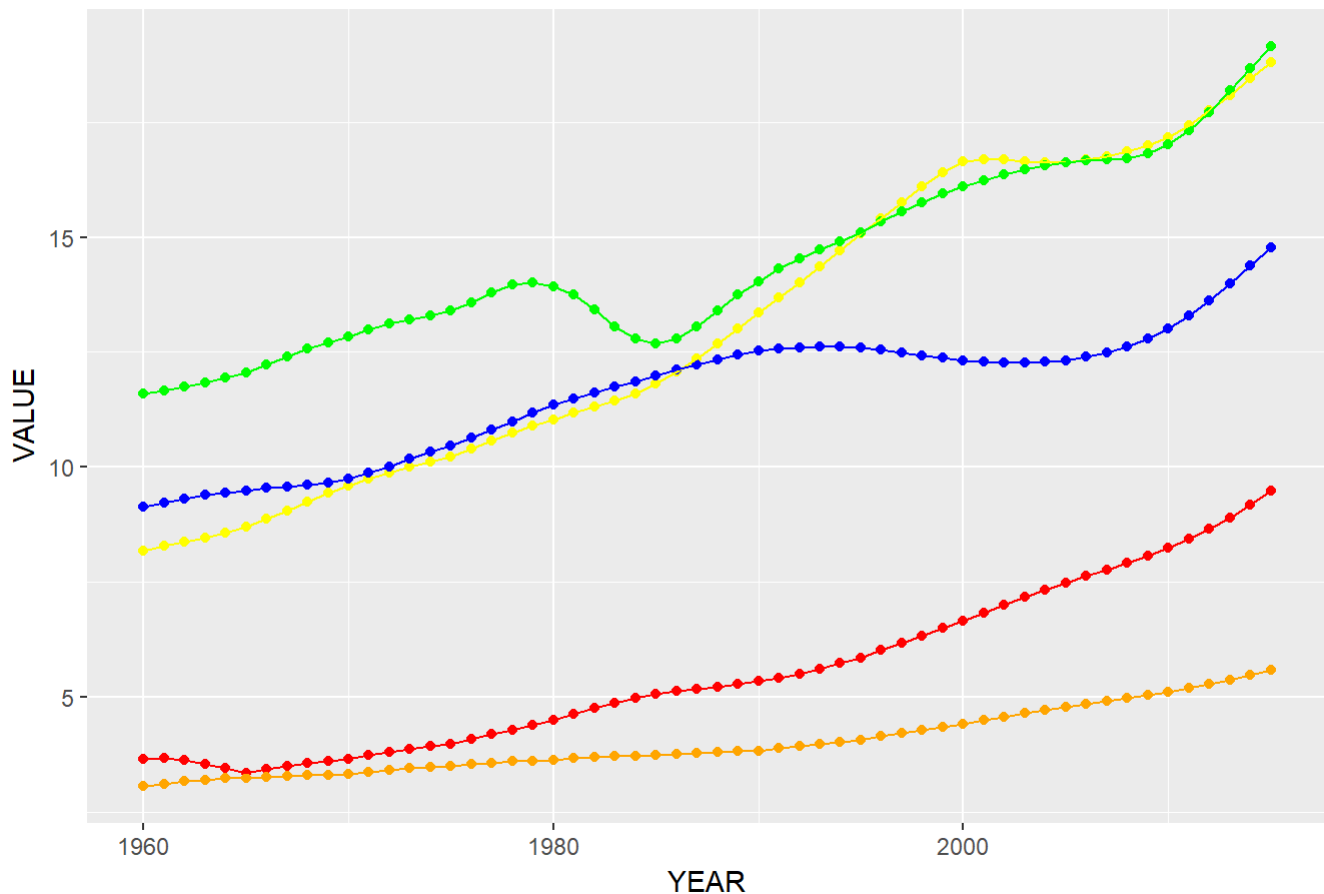
The mortality rate in infants (per 1000 births) is shown in this plot. All countries follow a decreasing trend with all but India having really low values nowadays although India is following a more decreasing tendency.

Population ages 65 and over

```
ggplot() +
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Population age
s 65 and above (% of total)`), color = "yellow") + geom_point(data = health_CHN, aes
(x = health_CHN$Year, y = health_CHN$`Population ages 65 and above (% of total)`), co
lor = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND
$`Population ages 65 and above (% of total)`), color = "orange") + geom_point(data =
health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Population ages 65 and above (%
of total)`), color = "green") + geom_point(data = health_USA, aes(x = health_USA$Yea
r, y = health_USA$`Population ages 65 and above (% of total)`), color = "blue") + geo
m_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Population ages 65
and above (% of total)`), color = "yellow") + geom_line(data = health_CHN, aes(x = h
ealth_CHN$Year, y = health_CHN$`Population ages 65 and above (% of total)`), color =
"red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Popula
tion ages 65 and above (% of total)`), color = "orange") + geom_line(data = health_FR
A, aes(x = health_FRA$Year, y = health_FRA$`Population ages 65 and above (% of total)
`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = heal
th_USA$`Population ages 65 and above (% of total)`), color = "blue") + ggtitle("Popul
ation ages 65 and above (% of total)") + xlab('YEAR') + ylab('VALUE') + theme(legend.
position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN",
"IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to co
ntinuous.
```

Population ages 65 and above (% of total)



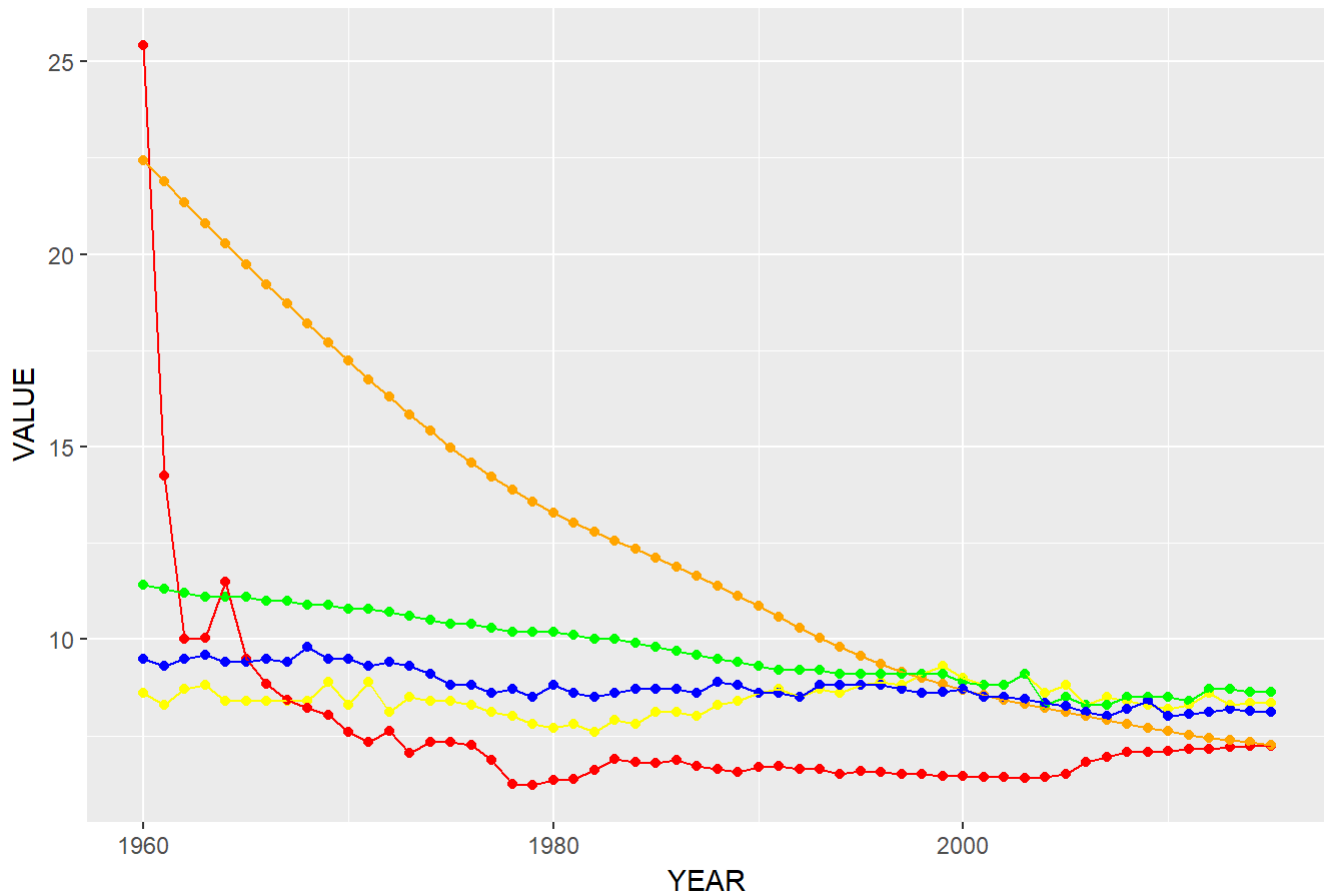
The population with ages 65 and above(% of total) is represented in this plot. Spain and France are the countries with the highest values what makes sense since they have the highest life expectancy but USA is a bit far from them even if it has a similar life expectancy. Then China and India are far away from the others and the follow an increasing tendency as well but with a smaller slope. This makes sense because even if their life expectancy is nowadays closer their population is still younger.

Death rate

```
ggplot() +
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Death rate, crude (per 1,000 people)`), color = "yellow") + geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Death rate, crude (per 1,000 people)`), color = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Death rate, crude (per 1,000 people)`), color = "orange") + geom_point(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Death rate, crude (per 1,000 people)`), color = "green") + geom_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Death rate, crude (per 1,000 people)`), color = "blue") + geom_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`Death rate, crude (per 1,000 people)`), color = "yellow") + geom_line(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`Death rate, crude (per 1,000 people)`), color = "red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`Death rate, crude (per 1,000 people)`), color = "orange") + geom_line(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`Death rate, crude (per 1,000 people)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`Death rate, crude (per 1,000 people)`), color = "blue") + ggtitle("Death rate, crude (per 1,000 people)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

Death rate, crude (per 1,000 people)



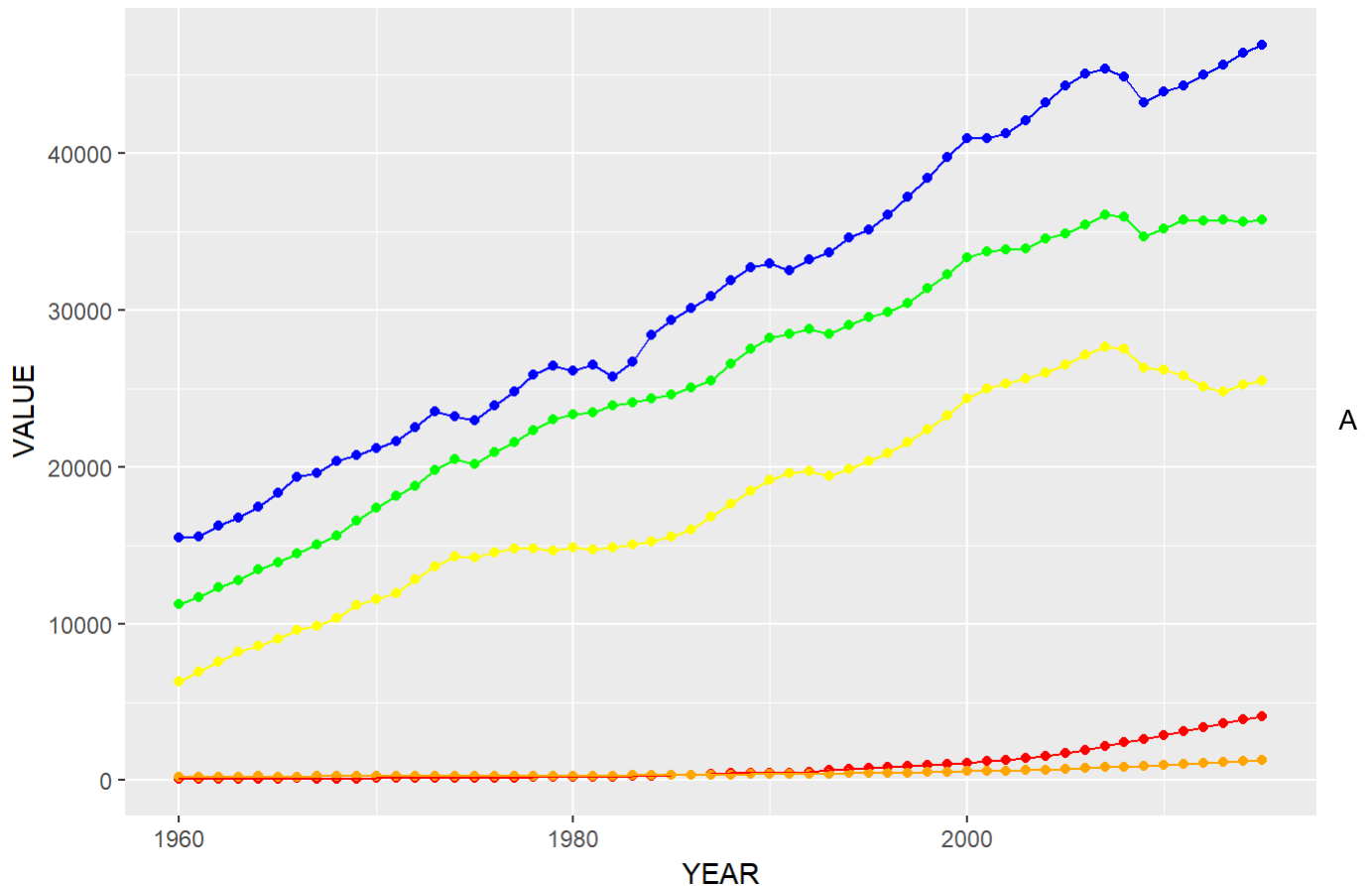
The death rate (per 1000 people) shown in this plot has been very similar for all countries except India since mid 1960s and India joined the same trend in the mid 1990s.

GDP at market prices(constant 2005 US\$)

```
ggplot() +
  geom_point(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`GDP per capita
    (constant 2005 US$)`), color = "yellow") + geom_point(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`GDP per capita
    (constant 2005 US$)`), color = "red") + geom_point(data = health_IND, aes(x = health_IND$Year, y = health_IND$`GDP per capita
    (constant 2005 US$)`), color = "orange") + geom_point(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`GDP per capita
    (constant 2005 US$)`), color = "green")
  + geom_point(data = health_USA, aes(x = health_USA$Year, y = health_USA$`GDP per capita
    (constant 2005 US$)`), color = "blue") + geom_line(data = health_ESP, aes(x = health_ESP$Year, y = health_ESP$`GDP per capita
    (constant 2005 US$)`), color = "yellow")
  + geom_line(data = health_CHN, aes(x = health_CHN$Year, y = health_CHN$`GDP per capita
    (constant 2005 US$)`), color = "red") + geom_line(data = health_IND, aes(x = health_IND$Year, y = health_IND$`GDP per capita
    (constant 2005 US$)`), color = "orange") +
  geom_line(data = health_FRA, aes(x = health_FRA$Year, y = health_FRA$`GDP per capita
    (constant 2005 US$)`), color = "green") + geom_line(data = health_USA, aes(x = health_USA$Year, y = health_USA$`GDP per capita
    (constant 2005 US$)`), color = "blue") +
  ggtitle("GDP per capita (constant 2005 US$)") + xlab('YEAR') + ylab('VALUE') + theme(legend.position = "top") + scale_fill_discrete(name = "Country", labels = c("ESP", "CHN", "IND", "FRA", "USA"))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

GDP per capita (constant 2005 US\$)



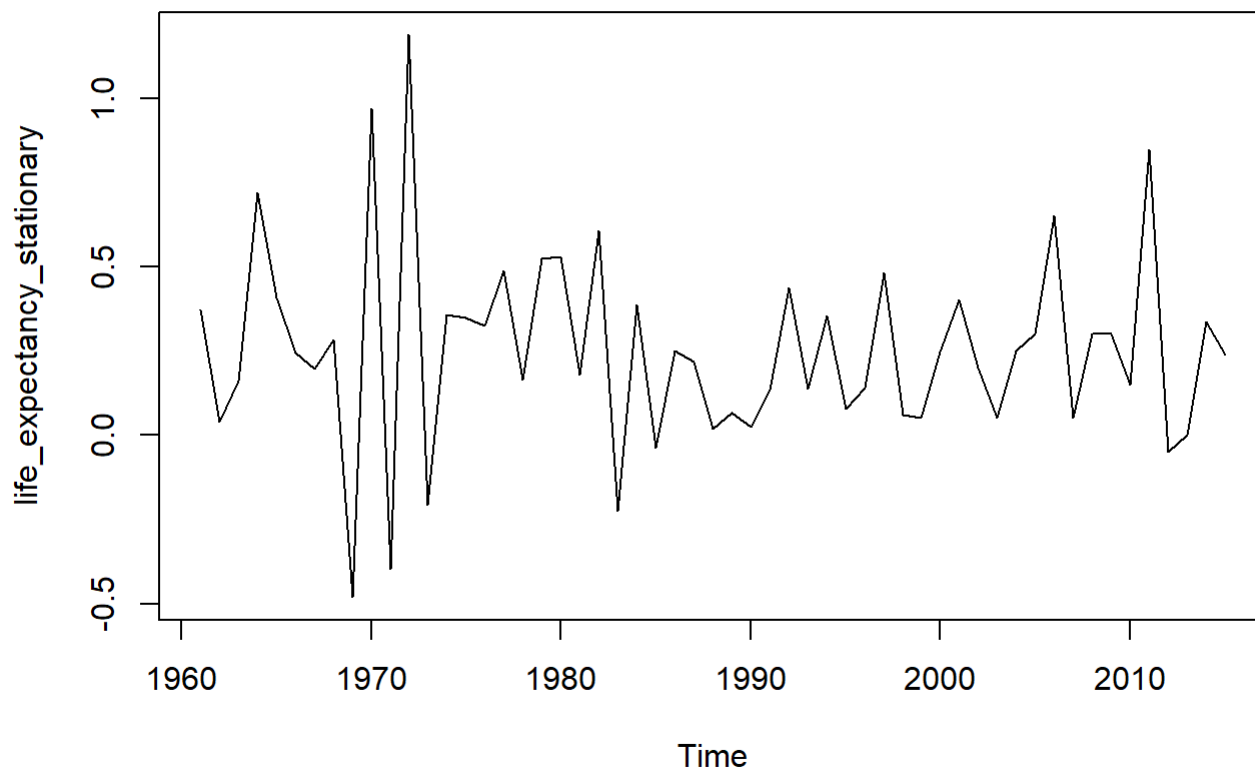
priori, we could think that the more GDP per capita a country has the greater life expectancy it may have since people in the country could afford a better life. However, it does not seem the case since USA is the country with the greatest GDP per capita and France is the second but Spain has a greater life expectancy than the previous two and France greater than USA as well. Also the huge difference of GDP per capita between these three countries and China and India does not correspond with the difference in the life expectancy that is considerably lower at least with China.

ARIMA FIT SPAIN

```
#check if the response time series is stationary
adf.test(health_ESP$`Life expectancy at birth, total (years)`)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: health_ESP$`Life expectancy at birth, total (years)`
## Dickey-Fuller = -2.3709, Lag order = 3, p-value = 0.4251
## alternative hypothesis: stationary
```

```
#non-stationary
life_expectancy_stationary <- diff(health_ESP$`Life expectancy at birth, total (years)`, differences = 1)
plot(life_expectancy_stationary)
```

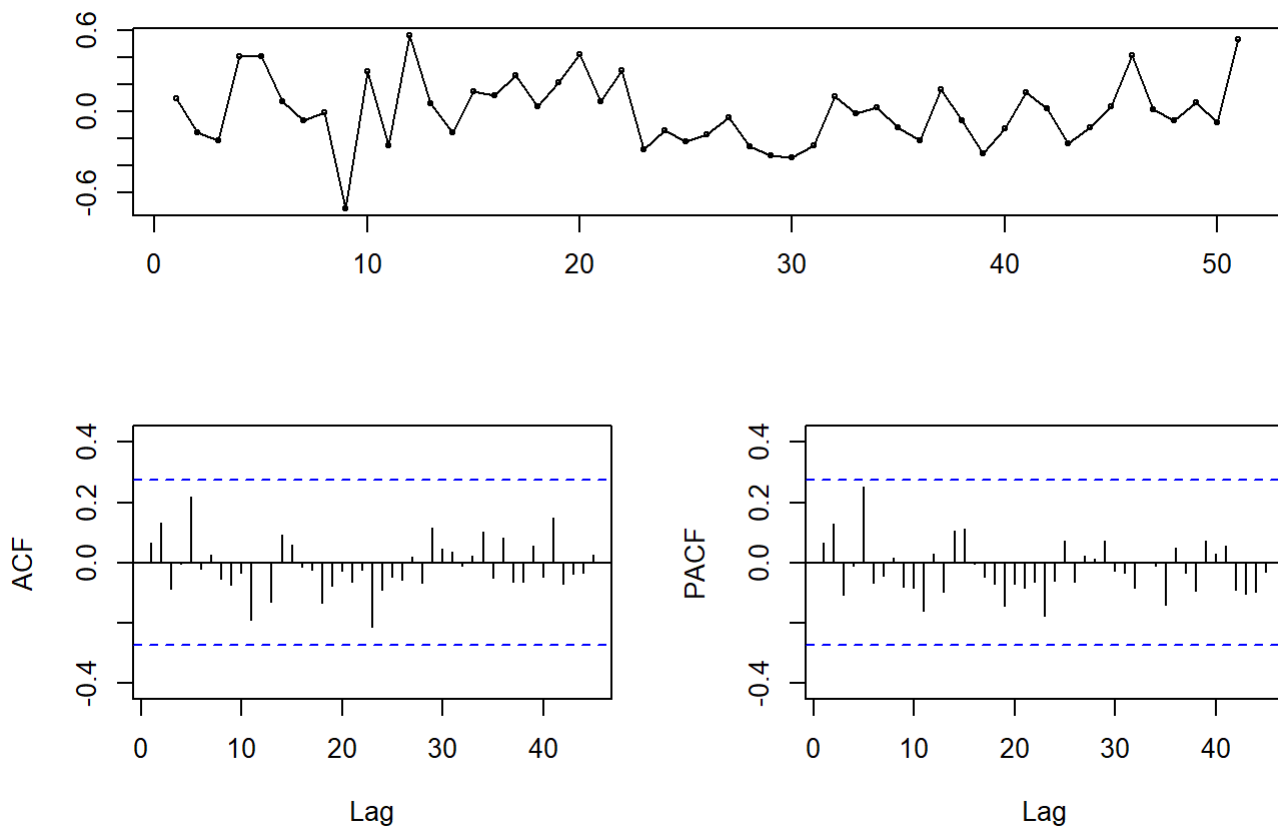
```
life.train <- life_expectancy_stationary[1:51]
life.test <- life_expectancy_stationary[52:55]

#fit ARIMA model
arima.fit <- auto.arima(life.train, stationary = TRUE, seasonal = FALSE)
summary(arima.fit)
```

```
## Series: life.train
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1      mean
##       -0.5613  0.2571
## s.e.    0.1195  0.0227
##
## sigma^2 estimated as 0.0654:  log likelihood=-1.99
## AIC=9.98   AICc=10.49   BIC=15.78
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.000870168 0.2506661 0.1968254 -78.45473 153.9383 0.5484762
##              ACF1
## Training set 0.06453686
```

```
#Plot to see if there are any patterns in ACF or PACF to choose better p, d, q
tsdisplay(residuals(arima.fit), lag.max=45, main='(1,0,0) Model Residuals')
```

(1,0,0) Model Residuals



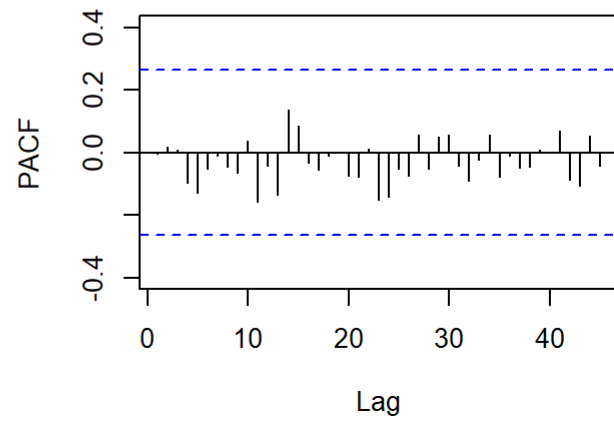
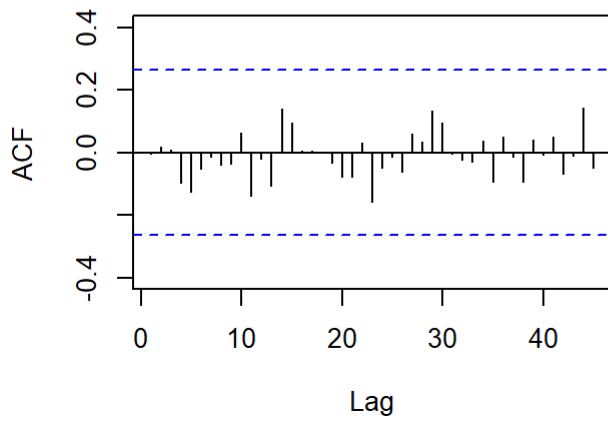
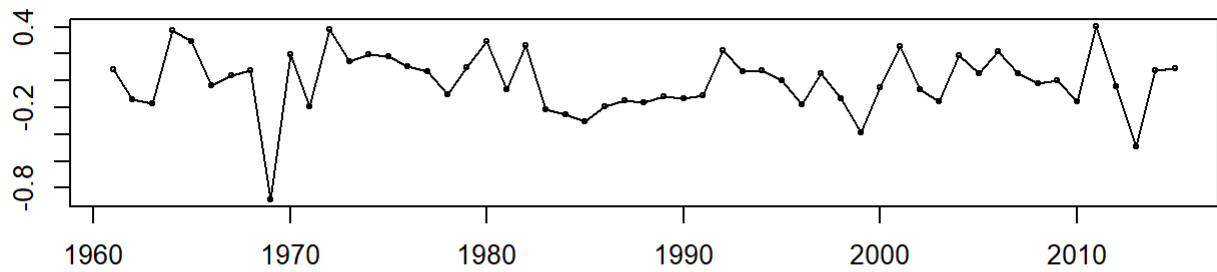
#Look for patterns: maybe a pattern around lag 5?? Fit a new model but I don't know how to compare both

```
arima.fit2 <- arima(life_expectancy_stationary, order = c(1, 0, 5))
summary(arima.fit2)
```

```
##
## Call:
## arima(x = life_expectancy_stationary, order = c(1, 0, 5))
##
## Coefficients:
##          ar1          ma1          ma2          ma3          ma4          ma5  intercept
##       -0.2846   -0.1937    0.2439   -0.2938    0.2593    0.4644         0.2598
## s.e.    0.4524    0.4433    0.2408    0.2158    0.2766    0.2447         0.0344
##
## sigma^2 estimated as 0.05098:  log likelihood = 2.24,  aic = 11.51
##
## Training set error measures:
##              ME          RMSE          MAE   MPE  MAPE          MASE
## Training set -0.006207387 0.225781 0.1717211 -Inf  Inf 0.4798902
##              ACF1
## Training set -0.005168959
```

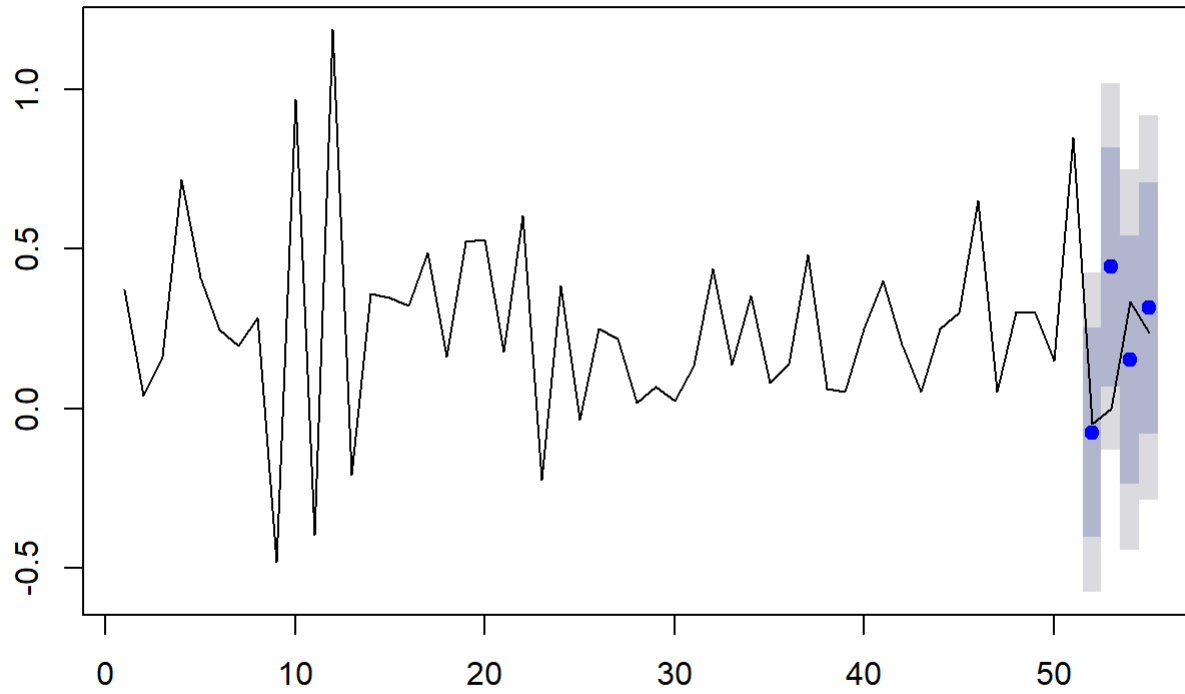
```
tsdisplay(residuals(arima.fit2), lag.max=45, main='(1,0,5) Model Residuals')
```

(1,0,5) Model Residuals



```
fcast <- forecast(arima.fit, h = 4)
plot(fcast, main = "life_expectancy_ESP")
lines(ts(life_expectancy_stationary))
```

life_expectancy_ESP

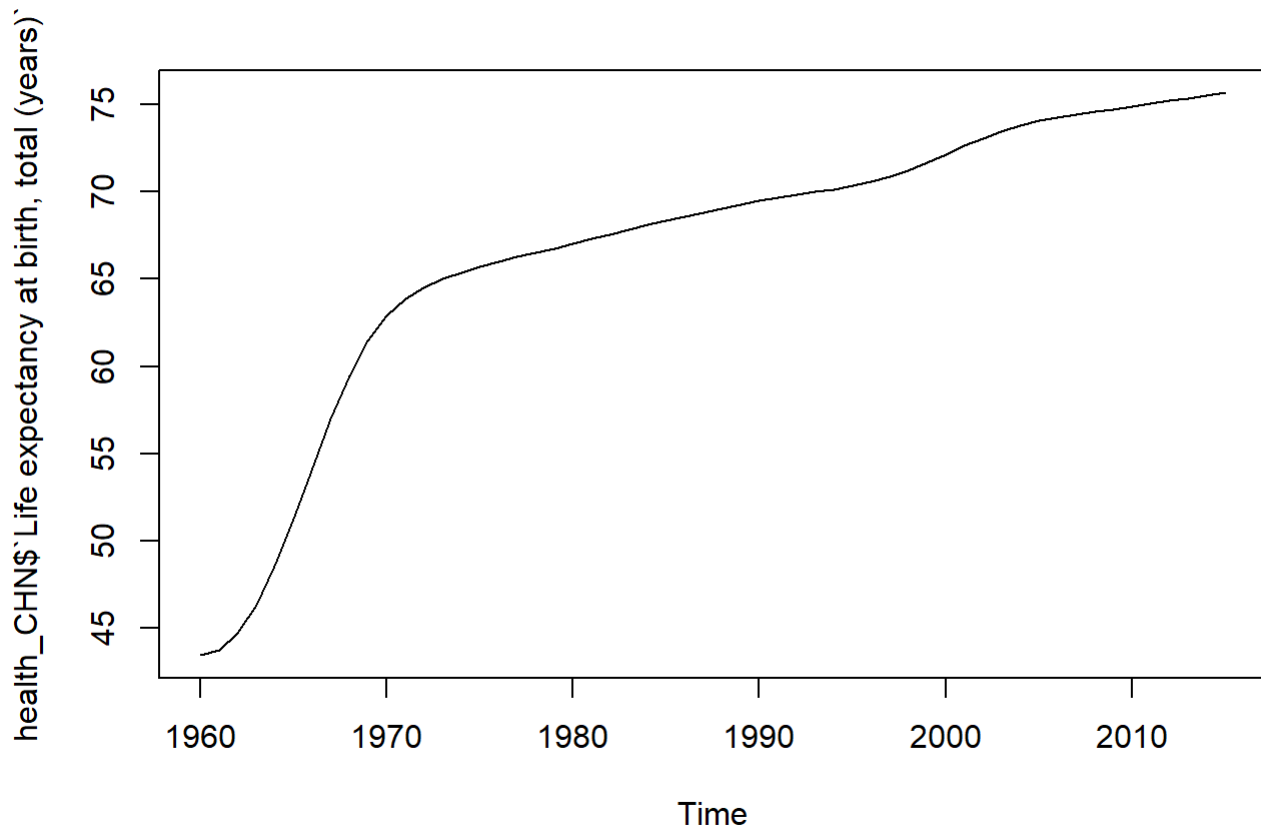


```
mse(as.numeric(life.test), as.numeric(fcast$mean))
```

```
## [1] 0.05928289
```

ARIMA FIT CHINA

```
plot(health_CHN$`Life expectancy at birth, total (years)`)
```



```
#check if the response time series is stationary  
adf.test(health_CHN$`Life expectancy at birth, total (years)`)
```

```
## Warning in adf.test(health_CHN$`Life expectancy at birth, total (years)`):  
## p-value smaller than printed p-value
```

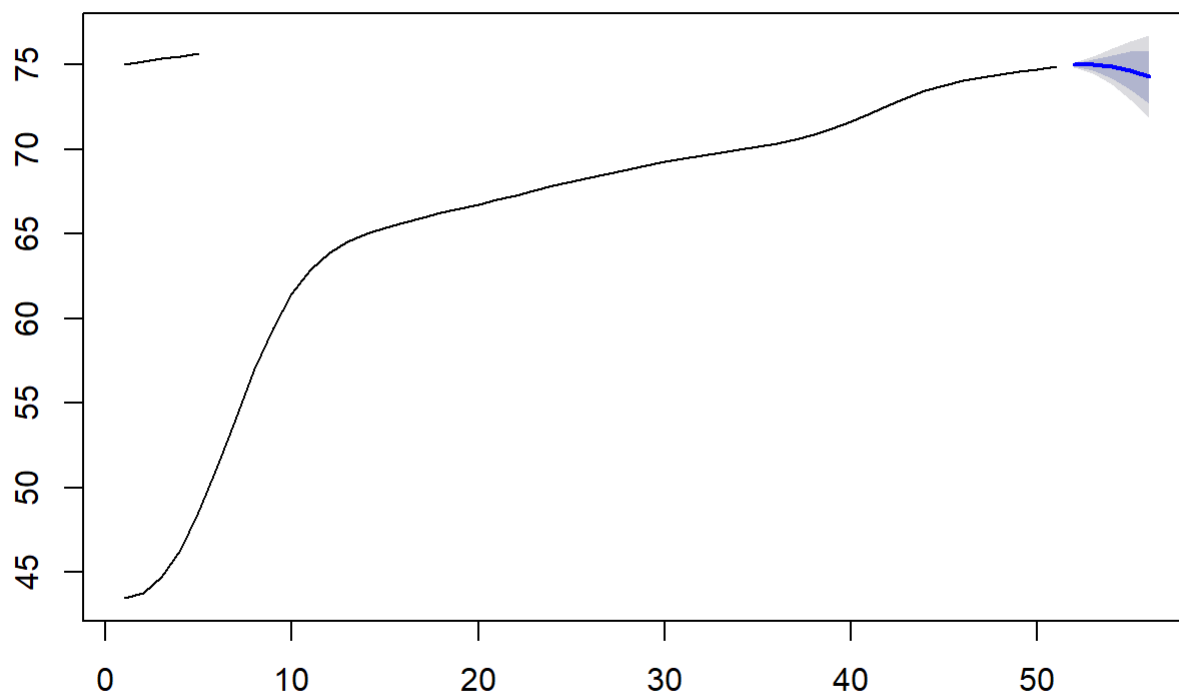
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: health_CHN$`Life expectancy at birth, total (years)`  
## Dickey-Fuller = -6.2272, Lag order = 3, p-value = 0.01  
## alternative hypothesis: stationary
```

```
life.train <- health_CHN$`Life expectancy at birth, total (years)`[1:51]  
life.test <- health_CHN$`Life expectancy at birth, total (years)`[52:56]  
  
#fit ARIMA model  
arima.fit <- auto.arima(life.train, stationary = TRUE, seasonal = FALSE)  
summary(arima.fit)
```

```
## Series: life.train
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1          ar2          ma1          ma2          mean
##          1.9381    -0.9516    1.4373    0.9270    65.4654
## s.e.    0.0207     0.0218    0.2217    0.0883     2.2785
##
## sigma^2 estimated as 0.005234:  log likelihood=54.99
## AIC=-97.98   AICc=-96.07   BIC=-86.39
##
## Training set error measures:
##              ME          RMSE          MAE          MPE          MAPE
## Training set 0.006347124 0.06870565 0.04558647 0.008706917 0.0790803
##              MASE          ACF1
## Training set 0.07254551 0.5950081
```

```
fcast <- forecast(arima.fit, h = 5)
plot(fcast, main = "life_expectancy_CHN")
lines(ts(life.test))
```

life_expectancy_CHN



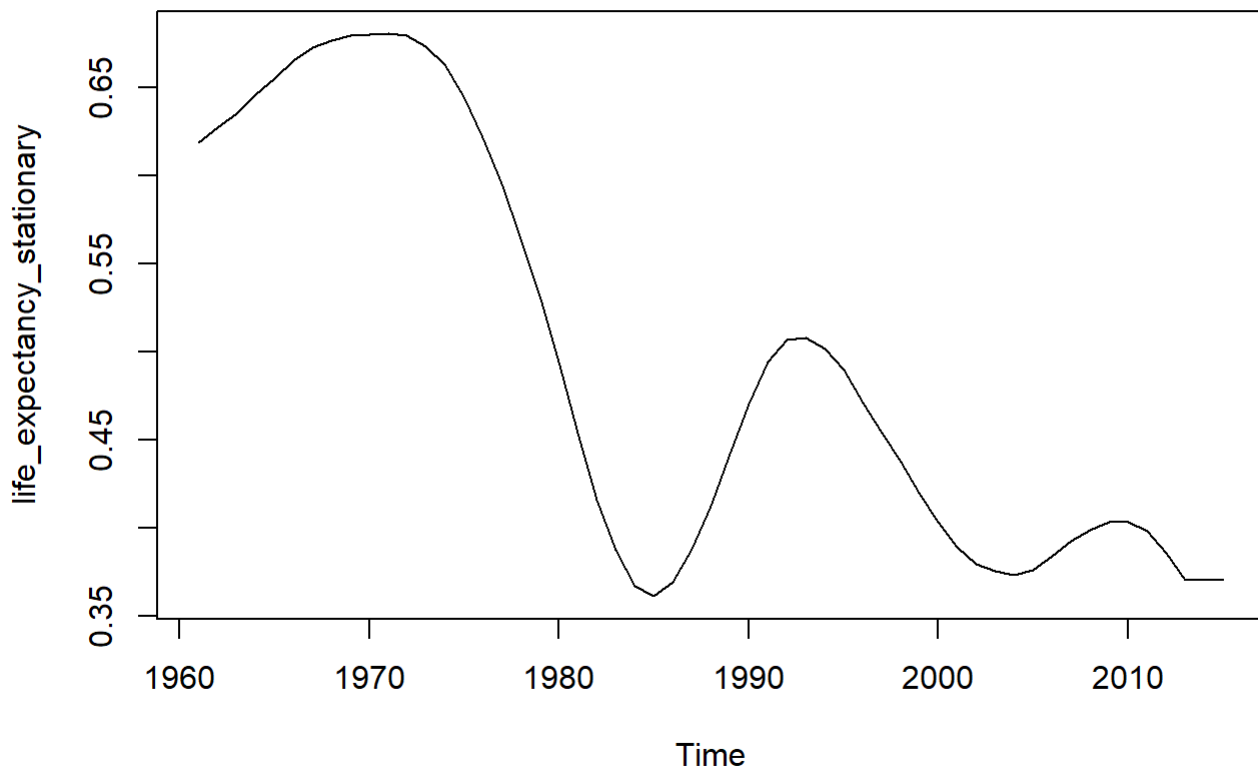
```
mse(as.numeric(life.test), as.numeric(fcast$mean))
```

```
## [1] 0.5610963
```

```
#check if the response time series is stationary  
adf.test(health_IND$`Life expectancy at birth, total (years)`)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: health_IND$`Life expectancy at birth, total (years)`  
## Dickey-Fuller = -3.0919, Lag order = 3, p-value = 0.1343  
## alternative hypothesis: stationary
```

```
#non-stationary  
life_expectancy_stationary <- diff(health_IND$`Life expectancy at birth, total (years)`,  
  differences = 1)  
plot(life_expectancy_stationary)
```



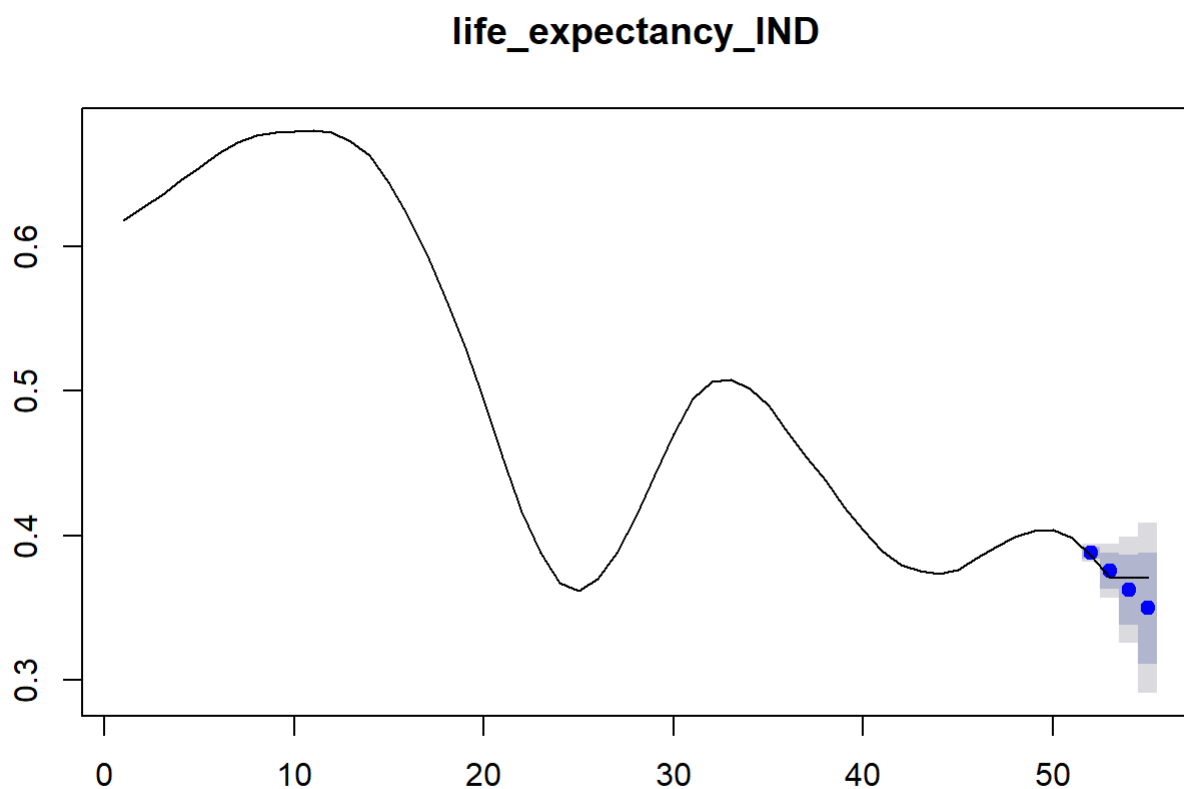
```
life.train <- life_expectancy_stationary[1:51]  
life.test <- life_expectancy_stationary[52:55]  
  
#fit ARIMA model  
arima.fit <- auto.arima(life.train, stationary = TRUE, seasonal = FALSE)  
summary(arima.fit)
```

```
## Series: life.train  
## ARIMA(2,0,3) with zero mean  
##  
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): Se han producido NaNs
```

```
##          ar1          ar2          ma1          ma2          ma3
##          1.8517 -0.8547  0.9340  0.6771  0.3752
## s.e.       NaN          NaN  0.0738  0.0607  0.0819
##
## sigma^2 estimated as 1.056e-05:  log likelihood=221.83
## AIC=-431.67  AICc=-429.76  BIC=-420.08
##
## Training set error measures:
##              ME          RMSE          MAE          MPE          MAPE
## Training set 0.0002826178 0.003086098 0.002444337 0.09412383 0.5279257
##              MASE          ACF1
## Training set 0.1751508 0.1332051
```

```
fcast <- forecast(arima.fit, h = 4)
plot(fcast, main = "life_expectancy_IND")
lines(ts(life_expectancy_stationary))
```



```
mse(as.numeric(life.test), as.numeric(fcast$mean))
```

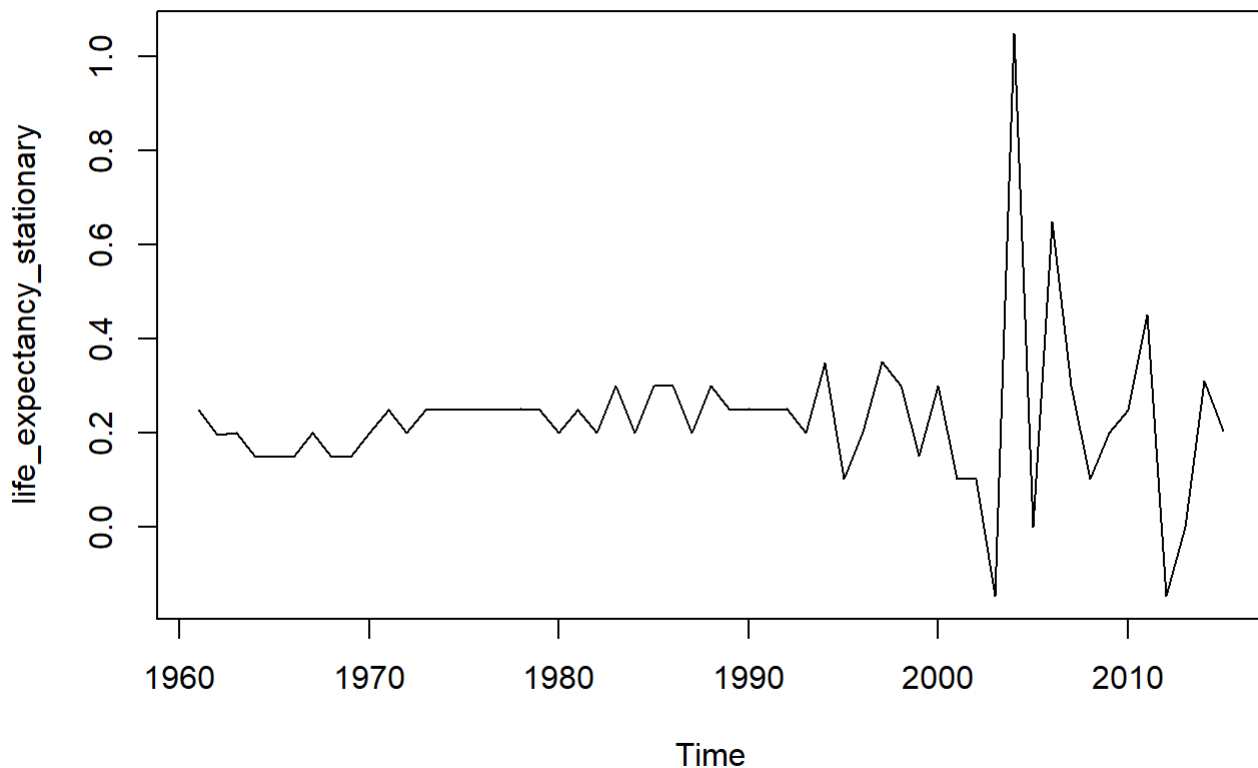
```
## [1] 0.0001348046
```

ARIMA FIT FRANCE


```
#check if the response time series is stationary  
adf.test(health_FRA$`Life expectancy at birth, total (years)`)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: health_FRA$`Life expectancy at birth, total (years)`  
## Dickey-Fuller = -3.3279, Lag order = 3, p-value = 0.07628  
## alternative hypothesis: stationary
```

```
#non-stationary  
life_expectancy_stationary <- diff(health_FRA$`Life expectancy at birth, total (years)`,  
  differences = 1)  
plot(life_expectancy_stationary)
```

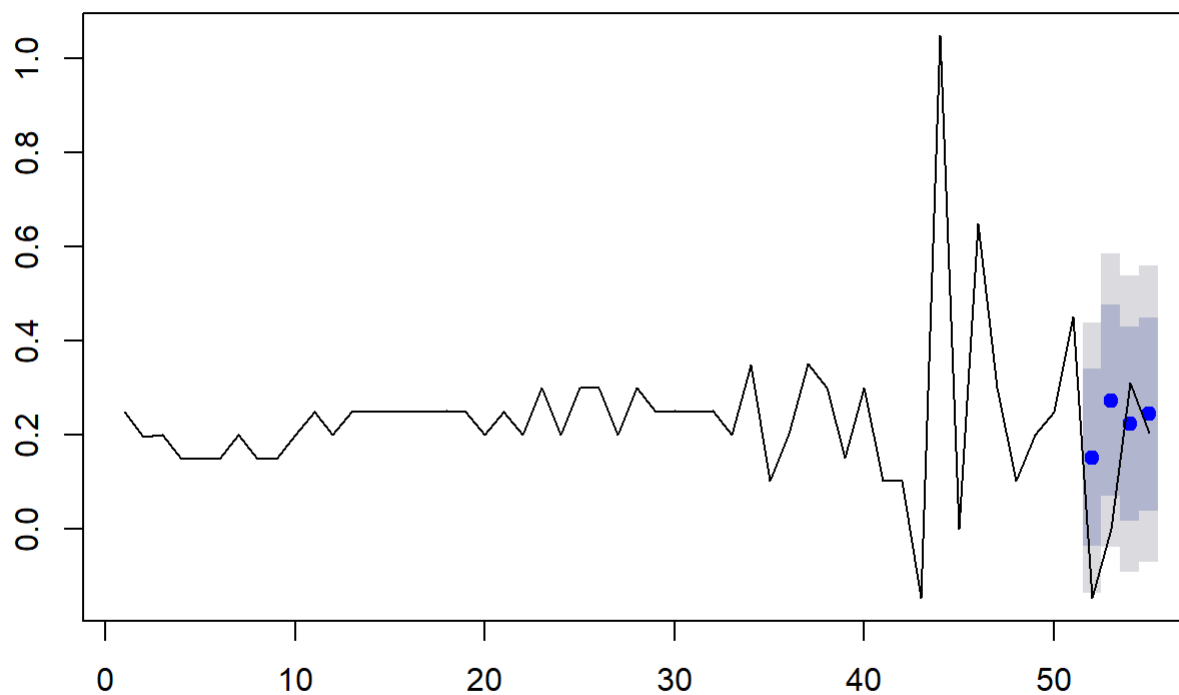


```
life.train <- life_expectancy_stationary[1:51]  
life.test <- life_expectancy_stationary[52:55]  
  
#fit ARIMA model  
arima.fit <- auto.arima(life.train, stationary = TRUE, seasonal = FALSE)  
summary(arima.fit)
```

```
## Series: life.train
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1      mean
##        -0.4079  0.2389
## s.e.    0.1284  0.0144
##
## sigma^2 estimated as 0.02156:  log likelihood=26.41
## AIC=-46.82   AICc=-46.31   BIC=-41.02
##
## Training set error measures:
##              ME      RMSE      MAE MPE MAPE      MASE
## Training set 6.241306e-05 0.1439108 0.0872422 Inf  Inf 0.6823599
##              ACF1
## Training set 0.02848907
```

```
fcast <- forecast(arima.fit, h = 4)
plot(fcast, main = "life_expectancy_FRA")
lines(ts(life_expectancy_stationary))
```

life_expectancy_FRA



```
mse(as.numeric(life.test), as.numeric(fcast$mean))
```

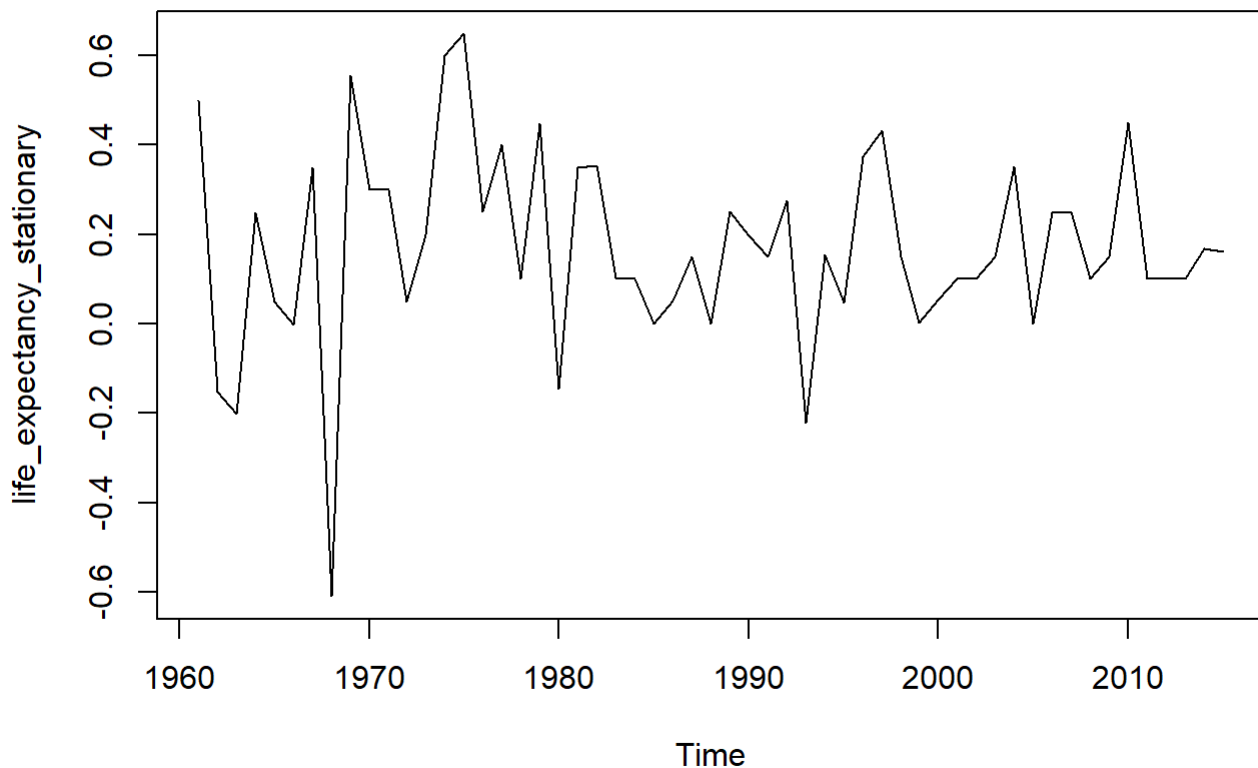
```
## [1] 0.04333
```

ARIMA FIT USA

```
#check if the response time series is stationary  
adf.test(health_USA$`Life expectancy at birth, total (years)`)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: health_USA$`Life expectancy at birth, total (years)`  
## Dickey-Fuller = -1.8719, Lag order = 3, p-value = 0.6263  
## alternative hypothesis: stationary
```

```
#non-stationary  
life_expectancy_stationary <- diff(health_USA$`Life expectancy at birth, total (years)`,  
differences = 1)  
plot(life_expectancy_stationary)
```



```
life.train <- life_expectancy_stationary[1:51]  
life.test <- life_expectancy_stationary[52:55]  
  
#fit ARIMA model  
arima.fit <- auto.arima(life.train, stationary = TRUE, seasonal = FALSE)  
summary(arima.fit)
```

```
## Series: life.train
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##          mean
##         0.1739
## s.e.  0.0314
##
## sigma^2 estimated as 0.05116:  log likelihood=3.95
## AIC=-3.89   AICc=-3.64   BIC=-0.03
##
## Training set error measures:
##              ME      RMSE      MAE  MPE  MAPE      MASE
## Training set 3.49508e-14 0.2239588 0.1701659 -Inf  Inf 0.7081609
##              ACF1
## Training set -0.1277979
```

```
fcast <- forecast(arima.fit, h = 4)
plot(fcast, main = "life_expectancy_USA")
lines(ts(life_expectancy_stationary))
```



```
mse(as.numeric(life.test), as.numeric(fcast$mean))
```

```
## [1] 0.002787415
```

MULTIVARIATE VAR FIT

```
train <- health_ESP[1:50, c(2, 3, 7, 9)]
var.fit <- VAR(train, p = 2)
```

```
## Constant term:
## Estimates: 14.60863 10.25736 -2.201894 -4463.713
## Std.Error: 4.034343 5.139568 0.548745 5012.693
## AR coefficient matrix
## AR( 1 )-matrix
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.277 -0.16383 0.251 -1.21e-04
## [2,] -0.171 1.02642 0.624 1.75e-04
## [3,] 0.016 -0.00118 1.865 -3.21e-05
## [4,] -21.433 364.43790 -100.346 1.47e+00
## standard error
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.1186 0.1458 0.3806 1.35e-04
## [2,] 0.1512 0.1857 0.4848 1.72e-04
## [3,] 0.0161 0.0198 0.0518 1.84e-05
## [4,] 147.4218 181.1244 472.8648 1.68e-01
## AR( 2 )-matrix
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.50600 0.3577 -0.193 1.53e-04
## [2,] 0.00585 0.1704 -0.427 -2.54e-04
## [3,] 0.02083 -0.0514 -0.905 4.61e-05
## [4,] 75.41020 -252.8790 390.260 -6.84e-01
## standard error
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.118 0.1768 0.3922 1.39e-04
## [2,] 0.150 0.2253 0.4996 1.77e-04
## [3,] 0.016 0.0241 0.0533 1.88e-05
## [4,] 146.188 219.7316 487.3028 1.72e-01
##
## Residuals cov-mtx:
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.0401767583 0.003857329 0.0003895891 1.507325
## [2,] 0.0038573287 0.065205236 0.0013415804 35.396019
## [3,] 0.0003895891 0.001341580 0.0007433107 1.663037
## [4,] 1.5073248800 35.396018955 1.6630372924 62025.666524
##
## det(SSE) = 0.07723076
## AIC = -1.280957
## BIC = -0.05726274
## HQ = -0.8149675
```

MULTIVARIATE ARIMA FIT SPAIN

```
#fit multivariate ARIMA model
predictors <- health_ESP[1:51, c(3, 7, 9)]
arima.fit <- auto.arima(life.train, stationary = TRUE, seasonal = FALSE, xreg = predictors)
summary(arima.fit)
```

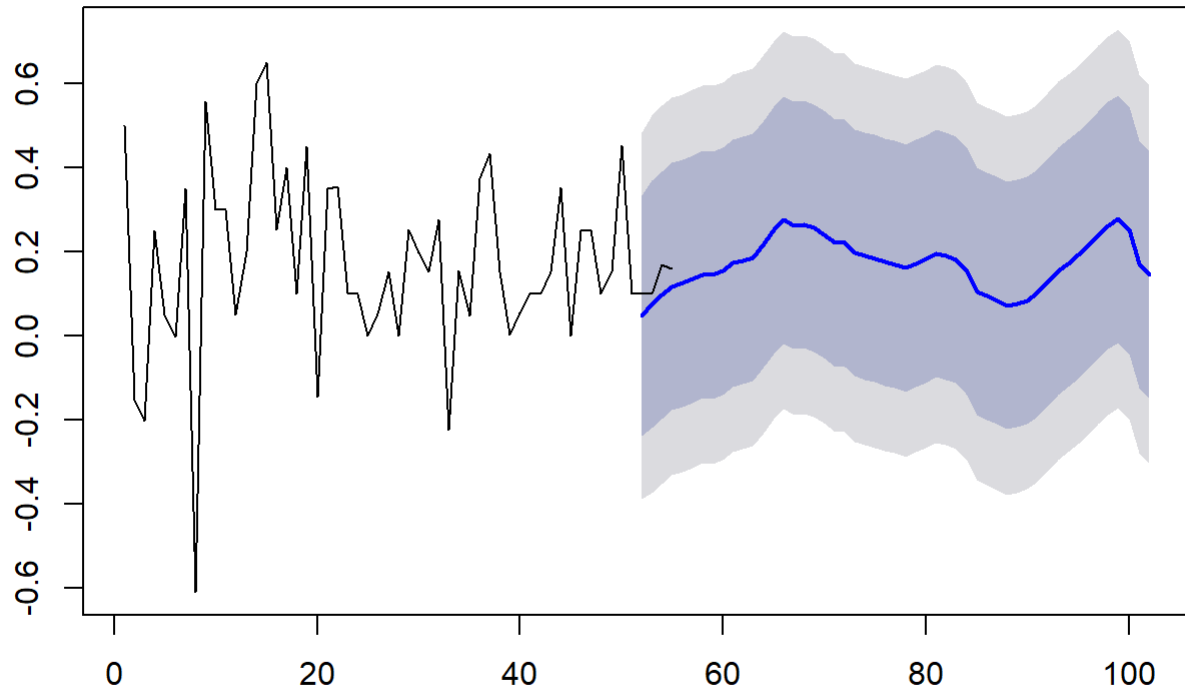
```
## Series: life.train
## Regression with ARIMA(0,0,1) errors
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): Se han producido NaNs
```

```
##          mal  intercept  CO2 emissions (metric tons per capita)
##          -0.2484    0.5444                                0.0122
## s.e.      0.0955    0.1982                                0.0182
##          Population ages 65 and above (% of total)
##                                -0.0981
## s.e.                                0.0435
##          GDP per capita (constant 2005 US$)
##                                0
## s.e.                                NaN
##
## sigma^2 estimated as 0.04951:  log likelihood=6.88
## AIC=-1.76  AICc=0.15  BIC=9.83
##
## Training set error measures:
##                                ME      RMSE      MAE  MPE MAPE      MASE
## Training set 0.002010702 0.2113096 0.1585905 -Inf  Inf 0.6599887
##                                ACF1
## Training set 0.01627778
```

```
fcast <- forecast(arima.fit, xreg = predictors, h = 10)
plot(fcast)
lines(ts(life_expectancy_stationary))
```

Forecasts from Regression with ARIMA(0,0,1) errors



```
mse(as.numeric(life.test), as.numeric(fcast$mean[1:4]))
```

```
## [1] 0.002566861
```