```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v dplyr   1.0.9
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.0     v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(RColorBrewer)
library(ggthemes)
library(lubridate)
```

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

```
library(ggrepel)
library(reshape)
```

```
##
## Attaching package: 'reshape'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
##
## The following object is masked from 'package:dplyr':
##
##     rename
##
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(stringr)
library(ggcorrplot)
library(viridis)
```

```
## Loading required package: viridisLite
##
## Attaching package: 'viridis'
##
## The following object is masked from 'package:maps':
##
##     unemp
##
## The following object is masked from 'package:scales':
##
##     viridis_pal
```

```
df <- read_csv("countries of the world.csv")
```

```
## Rows: 227 Columns: 20
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): Country, Region, PopDensity, Coastline, Net migration, Phones, Ara...
## dbl  (3): Population, Area, GDP
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(df)
```

```
dim(df)
```

```
## [1] 227  20
```

```
summary(df)
```

```
##    Country             Region             Population             Area
##  Length:227         Length:227         Min.   :7.026e+03   Min.   :       2
##  Class :character   Class :character   1st Qu.:4.376e+05   1st Qu.:    4648
##  Mode  :character   Mode  :character   Median :4.787e+06   Median :   86600
##                                        Mean   :2.874e+07   Mean   :  598227
##                                        3rd Qu.:1.750e+07   3rd Qu.:  441811
##                                        Max.   :1.314e+09   Max.   :17075200
##
##   PopDensity         Coastline         Net migration      Infant mortality
##  Length:227         Length:227         Length:227         Min.   :   19.0
##  Class :character   Class :character   Class :character   1st Qu.:  631.2
##  Mode  :character   Mode  :character   Mode  :character   Median : 1731.0
##                                                           Mean   : 3164.7
##                                                           3rd Qu.: 4929.8
##                                                           Max.   :19119.0
##                                                           NA's   :3
##       GDP            Literacy         Phones             Arable
##  Min.   :  500   Min.   : 176.0   Length:227         Length:227
##  1st Qu.: 1900   1st Qu.: 706.0   Class :character   Class :character
##  Median : 5550   Median : 925.0   Mode  :character   Mode  :character
##  Mean   : 9690   Mean   : 828.4
##  3rd Qu.:15700   3rd Qu.: 980.0
##  Max.   :55100   Max.   :1000.0
##  NA's   :1       NA's   :18
##     Crops              Other         Climate         Birthrate
##  Length:227         Min.   :  50   Min.   : 1.000   Min.   :  10
##  Class :character   1st Qu.:5608   1st Qu.: 2.000   1st Qu.:1077
##  Mode  :character   Median :8015   Median : 2.000   Median :1800
##                     Mean   :6813   Mean   : 2.995   Mean   :2043
##                     3rd Qu.:9299   3rd Qu.: 3.000   3rd Qu.:2934
##                     Max.   :9998   Max.   :25.000   Max.   :5073
##                     NA's   :2      NA's   :22       NA's   :3
##    Deathrate      Agriculture          Industry           Service
##  Min.   :  22.0   Length:227         Length:227         Length:227
##  1st Qu.: 517.5   Class :character   Class :character   Class :character
##  Median : 713.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 819.0
##  3rd Qu.:1025.5
##  Max.   :2974.0
##  NA's   :4
```

This dataset contains 227 rows and 20 cloumns. 110

```
sum(is.na(df))
```

```
## [1] 110
```

There are 110 null values present in the data set.

```r
sapply(df, function(x) sum(is.na(x)))
```

```
##          Country           Region       Population             Area
##                0                0                0                0
##       PopDensity        Coastline    Net migration Infant mortality
##                0                0                3                3
##              GDP         Literacy           Phones           Arable
##                1               18                4                2
##            Crops            Other          Climate        Birthrate
##                2                2               22                3
##        Deathrate      Agriculture         Industry          Service
##                4               15               16               15
```

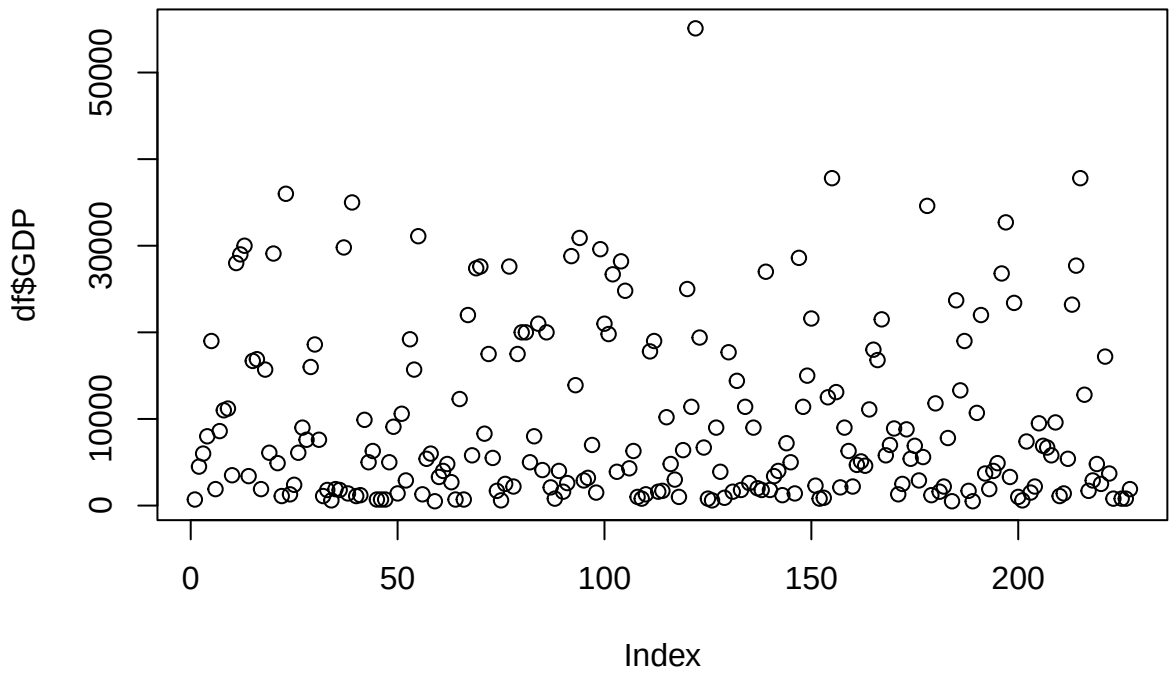
```r
colnames(df)
```

```
##  [1] "Country"          "Region"           "Population"       "Area"
##  [5] "PopDensity"       "Coastline"        "Net migration"    "Infant mortality"
##  [9] "GDP"              "Literacy"         "Phones"           "Arable"
## [13] "Crops"            "Other"            "Climate"          "Birthrate"
## [17] "Deathrate"        "Agriculture"      "Industry"         "Service"
```

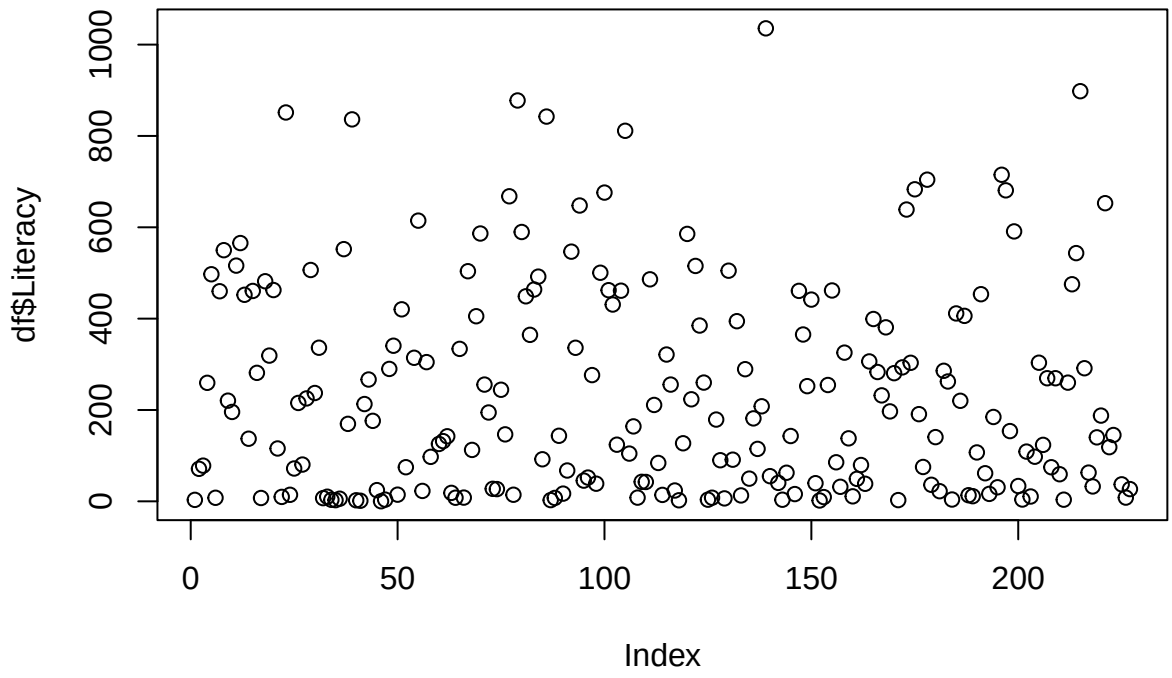
```r
df <- df[,c("Country","Population","Area","PopDensity","GDP","Literacy","Phones","Birthrate","Deathrate
names(df) <- c("Region","Population","Area","PopDensity","GDP","Phones","Literacy","Birthrate","Deathra
```

```r
df$Region <- gsub(" ", "", df$Region)
df$PopDensity <- as.numeric(gsub(",", ".", df$PopDensity))
df$Phones <- as.numeric(gsub(",", ".", df$Phones))
df$Literacy <- as.numeric(gsub(",", ".", df$Literacy))
df$Birthrate    <- as.numeric(gsub(",", ".", df$Birthrate))
df$Deathrate    <- as.numeric(gsub(",", ".", df$Deathrate))
df$Agriculture <- as.numeric(gsub(",", ".", df$Agriculture))
df$Industry <- as.numeric(gsub(",", ".", df$Industry))
df$Service <- as.numeric(gsub(",", ".", df$Service))
```

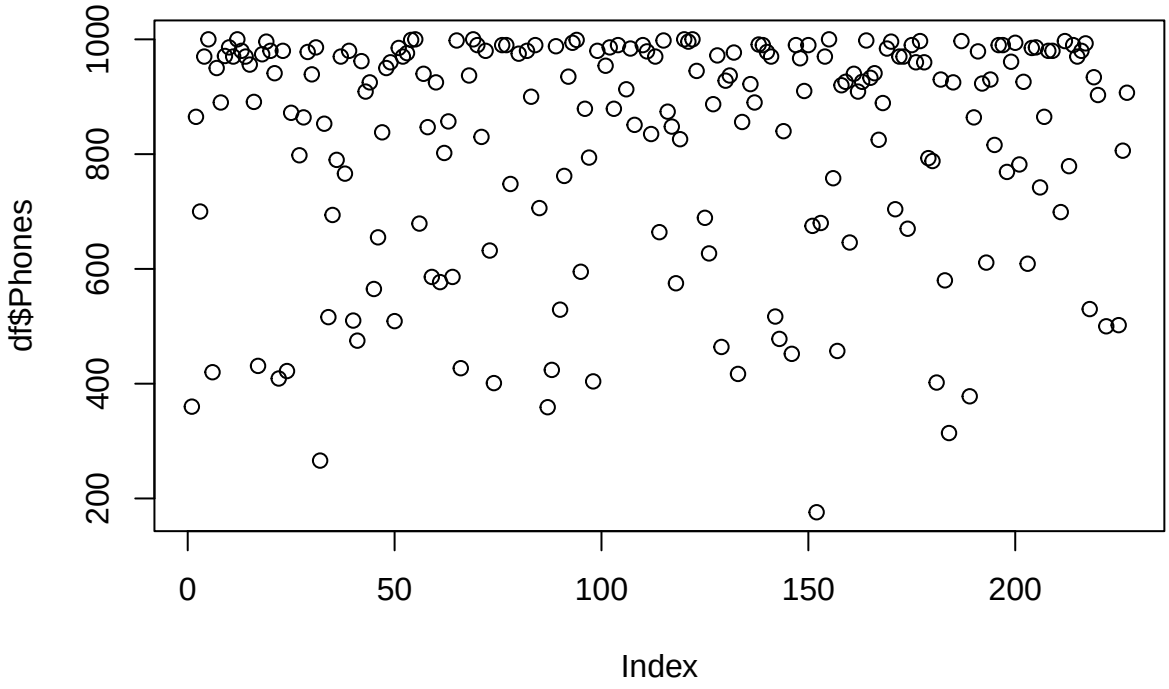
```r
library(dplyr)
plot(df$GDP)
```



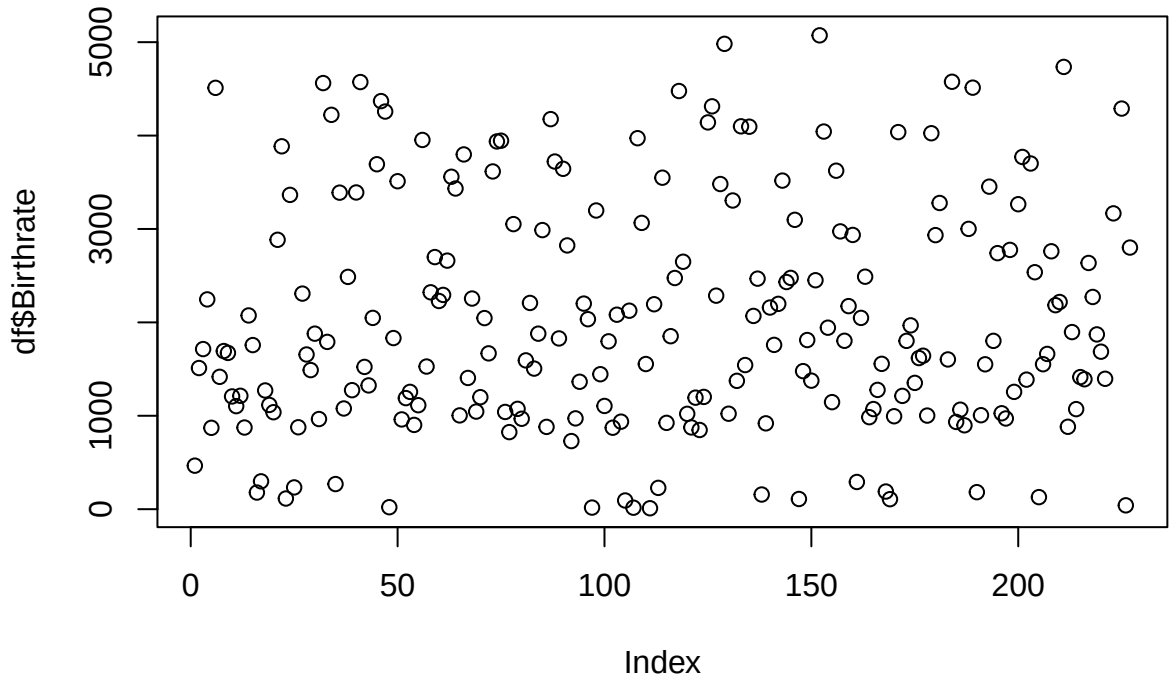

```
plot(df$Literacy)
```



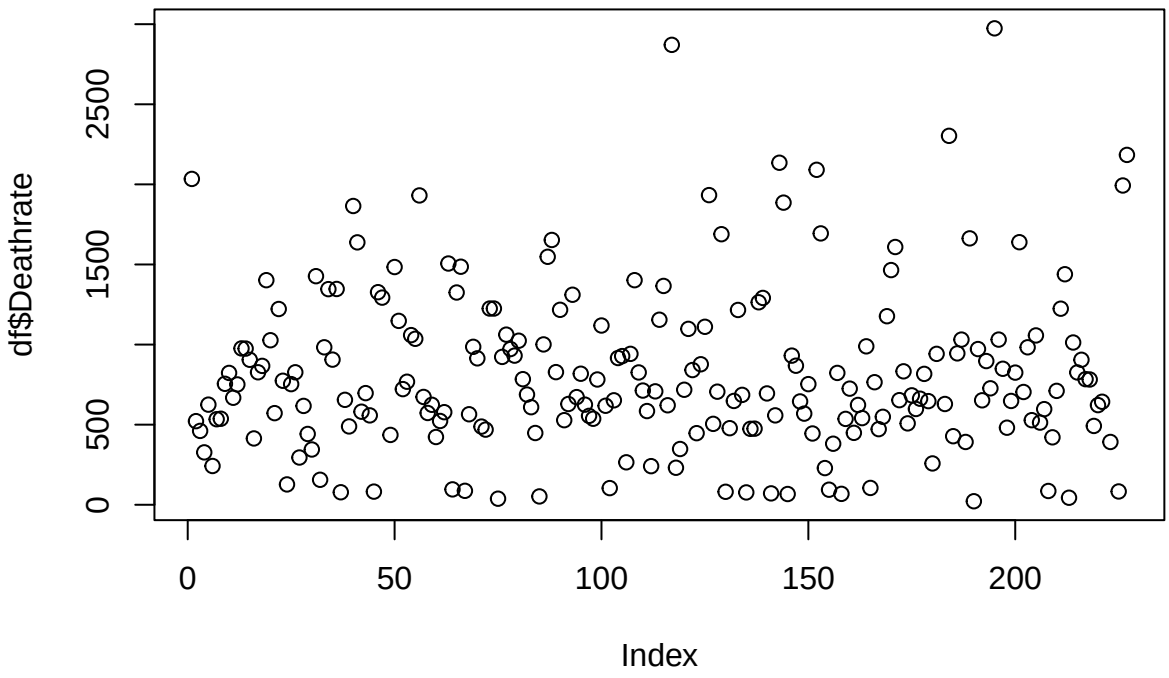

```
plot(df$Phones)
```



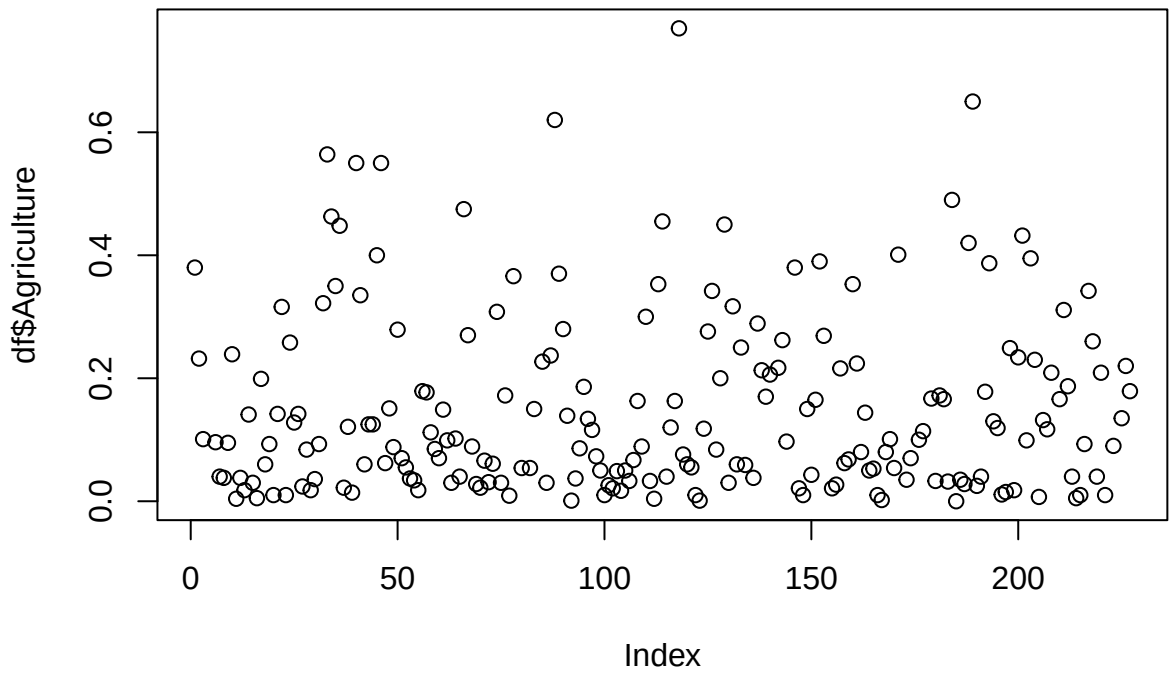

```
plot(df$Birthrate)
```



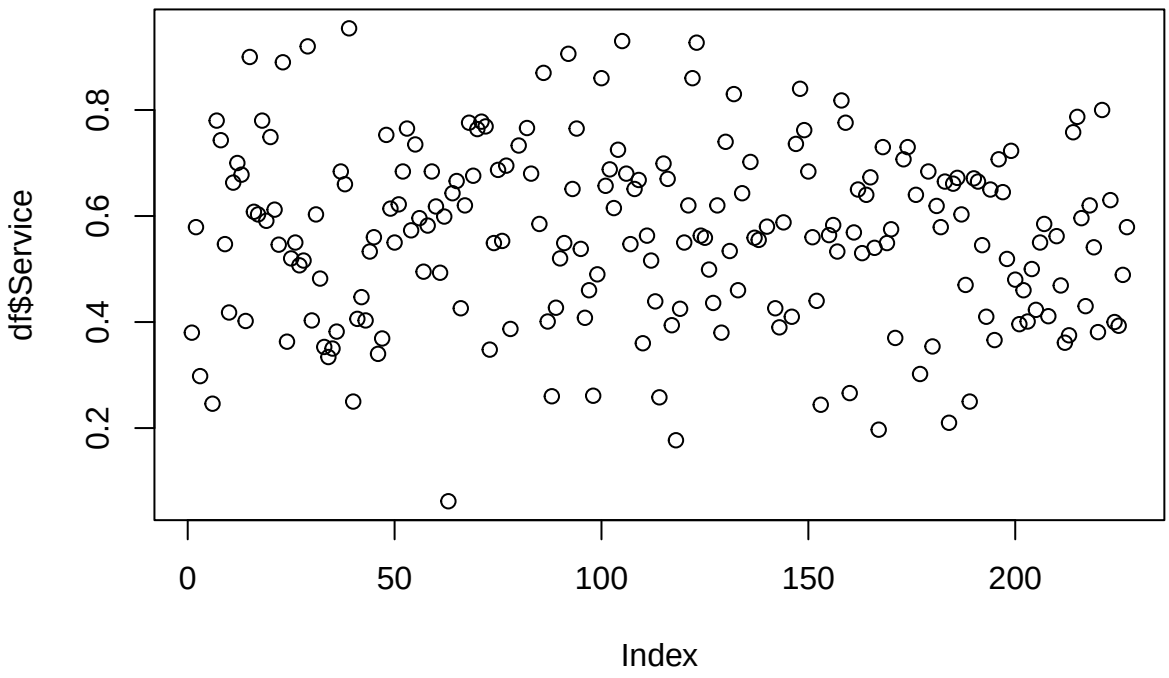

```
plot(df$Deathrate)
```



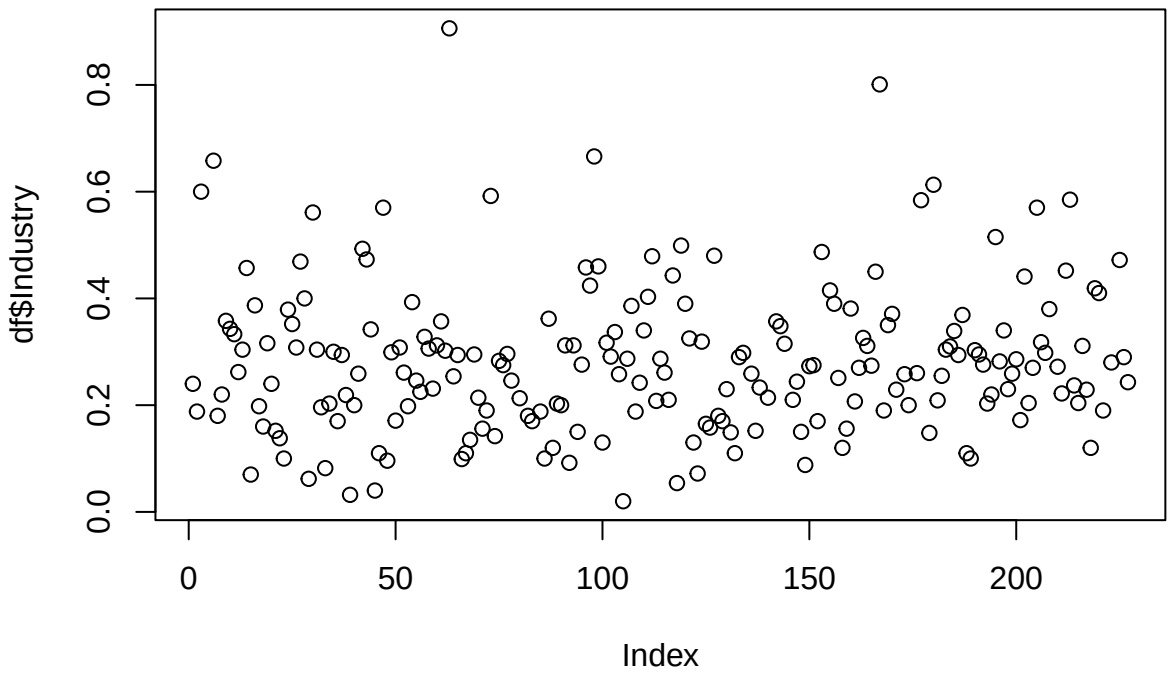

```
plot(df$Agriculture)
```




```
plot(df$Service)
```




```
plot(df$Industry)
```



Replacing the null/missing values with their respective mean.

```
df$GDP[is.na(df$GDP)] <- mean(df$GDP, na.rm = TRUE)
df$Literacy[is.na(df$Literacy)] <- mean(df$Literacy, na.rm = TRUE)
df$Phones[is.na(df$Phones)] <- mean(df$Phones, na.rm = TRUE)
df$Birthrate[is.na(df$Birthrate)] <- mean(df$Birthrate, na.rm = TRUE)
df$Deathrate[is.na(df$Deathrate)] <- mean(df$Deathrate, na.rm = TRUE)
df$Agriculture[is.na(df$Agriculture)] <- mean(df$Agriculture, na.rm = TRUE)
df$Service[is.na(df$Service)] <- mean(df$Service, na.rm = TRUE)
df$Industry[is.na(df$Industry)] <- mean(df$Industry, na.rm = TRUE)
sum(is.na(df))
```

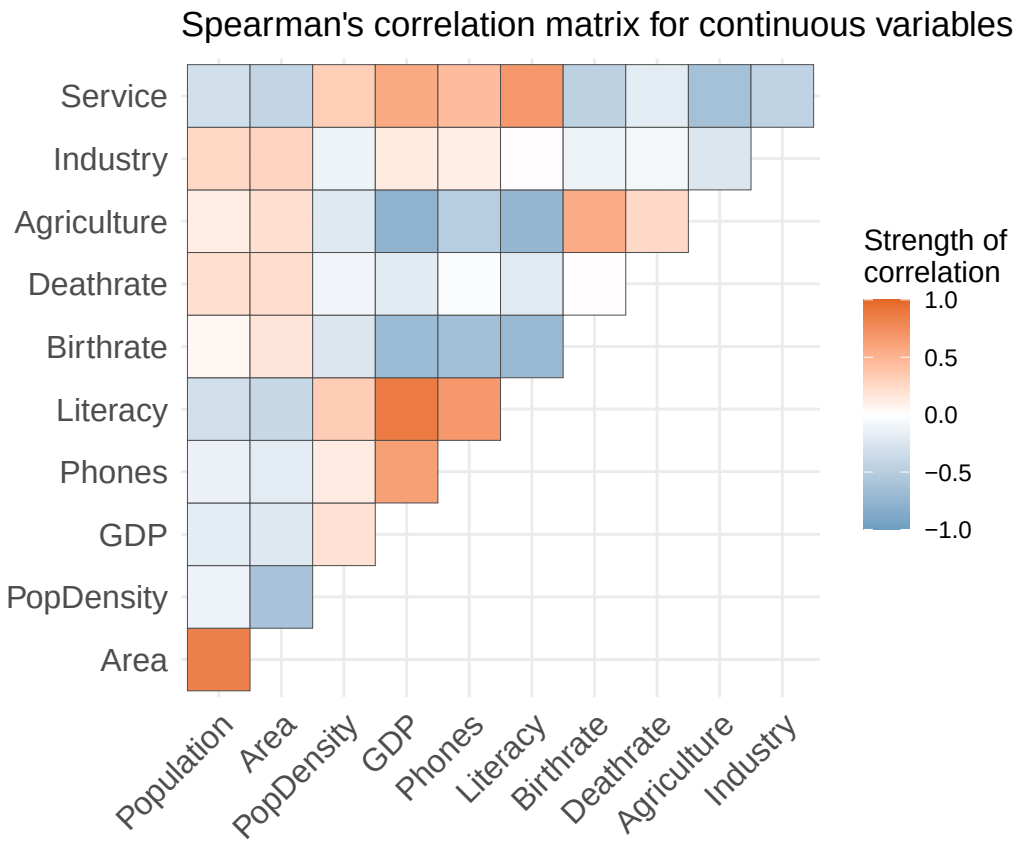
```
## [1] 0
```

```
view(df)
```

Successfully replaced all NA's and missing values present in the dataset with their respective mean.

```
options(warn = -1)
options(scipen = 10000)
options(repr.plot.width = 13.8, repr.plot.height = 9.2)
library(ggcorrplot)
annotate <- ggplot2::annotate
core <- cor(df[,c(2:ncol(df))], method = "spearman", use = "complete.obs")
options(repr.plot.width = 13, repr.plot.height = 11.18)
ggcorrplot(core, outline.col = "gray30", type = "upper",
```

```
          legend.title = "Strength of \ncorrelation", colors = c("#6D9EC1", "white", "#E46726"))+
  labs(y = "", x = "", title = "Spearman's correlation matrix for continuous variables")
```



Positive correlations are displayed in orange and negative correlations in blue color. Color intensity and the size of the circle are proportional to the correlation coefficients. (Area,Population),(Literacy,GDP),(Service,Literacy) are strongly correlated. (Birthrate,Phones),(Birthrate,GDP),(Agriculture,GDP),(Agriculture,Phones) are negatively correlated.