Capstone_01

Shyju Kozhukkunnon

3/26/2022

R Markdown

```r
#install.packages("ggpubr")

library(ggpubr)
```

```
## Loading required package: ggplot2
```

```r
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
```

Load the combined data from the data warehouse for statistical analysis.

```
##    country           countrycode             year       death_per_1000
##  Length:15799       Length:15799       Min.   :1960   Min.   : 1.127
##  Class :character   Class :character   1st Qu.:1975   1st Qu.: 6.941
##  Mode  :character   Mode  :character   Median :1990   Median : 9.121
##                                        Mean   :1990   Mean   :10.435
##                                        3rd Qu.:2005   3rd Qu.:12.437
##                                        Max.   :2020   Max.   :54.444
##
##    medageval         kvcval         leabval         hcfval
##  Min.   :14.30   Min.   :10.28   Min.   :18.91   Min.   : 129.5
##  1st Qu.:18.00   1st Qu.:68.57   1st Qu.:56.27   1st Qu.: 985.2
##  Median :21.20   Median :84.20   Median :67.34   Median :1762.5
##  Mean   :24.27   Mean   :77.86   Mean   :64.22   Mean   :2066.8
##  3rd Qu.:29.92   3rd Qu.:93.20   3rd Qu.:72.87   3rd Qu.:2854.8
##  Max.   :48.20   Max.   :99.91   Max.   :85.08   Max.   :8714.9
##  NA's   :4026    NA's   :4392    NA's   :4172    NA's   :14281
##   poverty_perc    gdp_pc_growth       rur_pop_growth        gdp_pc_ppp
##  Min.   : 0.00   Min.   :-64.9924   Min.   :-235.7924   Min.   :   436.7
##  1st Qu.:19.40   1st Qu.: -0.3038   1st Qu.:  -0.2909   1st Qu.:  3240.2
##  Median :29.00   Median :  2.0930   Median :   0.8375   Median :  8451.4
##  Mean   :32.69   Mean   :  1.7593   Mean   :  -0.2678   Mean   : 15386.1
##  3rd Qu.:45.80   3rd Qu.:  4.3917   3rd Qu.:   1.9234   3rd Qu.: 19837.1
```

```
##  Max.   :73.20   Max.   :140.3670   Max.   :  29.6283   Max.   :161971.5
##  NA's   :8540    NA's   :183        NA's   :671         NA's   :1220
##    countrygii       rur_perc      pri_exp_ppp          countrygdi
##  Min.   :0.012   Min.   : 0.00   Min.   :     2.96   Min.   :0.539
##  1st Qu.:0.252   1st Qu.:32.29   1st Qu.:    99.22   1st Qu.:0.862
##  Median :0.462   Median :52.45   Median :   415.29   Median :0.939
##  Mean   :0.431   Mean   :51.31   Mean   :  4418.21   Mean   :0.910
##  3rd Qu.:0.598   3rd Qu.:70.64   3rd Qu.:  1985.43   3rd Qu.:0.973
##  Max.   :0.828   Max.   :97.92   Max.   :293636.20   Max.   :1.096
##  NA's   :12751   NA's   :671     NA's   :7690        NA's   :12583
##   sec_exp_ppp
##  Min.   :     0.0
##  1st Qu.:    62.0
##  Median :   373.4
##  Mean   :  5336.8
##  3rd Qu.:  2170.8
##  Max.   :325625.6
##  NA's   :7690
```

Next check the correlation coefficients

```
##       cor
## 0.1635351
```

```
## [1] 1.026688e-19
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$countrygii
## t = 9.1488, df = 3046, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1287792 0.1978897
## sample estimates:
##       cor
## 0.1635351
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$countrygdi
## t = -14.67, df = 3214, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2826278 -0.2178338
## sample estimates:
##        cor
## -0.2505113
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$rur_pop_growth
## t = 14.255, df = 15126, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.09938127 0.13082975
## sample estimates:
##       cor
## 0.1151344

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$rur_perc
## t = 82.043, df = 15126, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5438123 0.5658700
## sample estimates:
##       cor
## 0.5549387

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$pri_exp_ppp
## t = -5.892, df = 8107, p-value = 3.969e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08694070 -0.04359447
## sample estimates:
##         cor
## -0.06529839

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$sec_exp_ppp
## t = -5.2349, df = 8107, p-value = 1.692e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07970729 -0.03632212
## sample estimates:
##         cor
## -0.05804211

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$kvcval
## t = -76.917, df = 11405, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5963871 -0.5722177
## sample estimates:
##       cor
## -0.584432

##
##  Pearson's product-moment correlation
##
```

```
## data:  df_death_rate$death_per_1000 and df_death_rate$hcfval
## t = -6.4899, df = 1516, p-value = 1.16e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2129651 -0.1150538
## sample estimates:
##        cor
## -0.1644144

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$gdp_pc_growth
## t = -5.8826, df = 15614, p-value = 4.12e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06266383 -0.03136428
## sample estimates:
##        cor
## -0.0470256

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$poverty_perc
## t = 36.42, df = 7257, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3734788 0.4123820
## sample estimates:
##       cor
## 0.3931063

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$death_per_1000 and df_death_rate$gdp_pc_ppp
## t = -47.115, df = 14577, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3775425 -0.3493667
## sample estimates:
##        cor
## -0.3635378

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$countrygii, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$countrygdi, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$rur_pop_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$rur_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
```

```
## df_death_rate$pri_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$sec_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$kvcval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$hcfval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$gdp_pc_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$poverty_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$death_per_1000,
## df_death_rate$gdp_pc_ppp, : Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$countrygii
## S = 4201703786, p-value = 1.257e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1097093

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$countrygdi
## S = 5765302978, p-value = 0.02337
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.03998077

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$rur_pop_growth
## S = 4.2156e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.2694261

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$rur_perc
## S = 2.876e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5015864
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$pri_exp_ppp
## S = 1.0332e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1625851
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$sec_exp_ppp
## S = 1.0388e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1689081
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$kvcval
## S = 3.6394e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4711805
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$hcfval
## S = 665321699, p-value = 3.292e-08
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1412159
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$gdp_pc_growth
## S = 6.5985e+11, p-value = 7.165e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.03965361
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$poverty_perc
## S = 4.3898e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
```

```
## sample estimates:
##       rho
## 0.3113961

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$death_per_1000 and df_death_rate$gdp_pc_ppp
## S = 7.4995e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4521199

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$countrygii
## z = 5.2021, p-value = 1.971e-07
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.06290502

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$countrygdi
## z = -0.87226, p-value = 0.3831
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.0102692

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$rur_pop_growth
## z = 32.118, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.1741698

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$rur_perc
## z = 65.086, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3529483

##
##  Kendall's rank correlation tau
##
```

```
## data:  df_death_rate$death_per_1000 and df_death_rate$pri_exp_ppp
## z = -14.983, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.1111149
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$sec_exp_ppp
## z = -15.598, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.1156765
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$kvcval
## z = -52.761, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.3297428
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$hcfval
## z = -6.1311, p-value = 8.728e-10
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.1058009
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$gdp_pc_growth
## z = -4.9307, p-value = 8.192e-07
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##         tau
## -0.02632266
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$poverty_perc
## z = 25.981, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2040985
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$death_per_1000 and df_death_rate$gdp_pc_ppp
## z = -55.72, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.3079678

##       cor
## -0.894784

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$countrygii
## t = -110.6, df = 3046, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9016441 -0.8874737
## sample estimates:
##       cor
## -0.894784

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$countrygdi
## t = 51.464, df = 3214, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6527404 0.6906584
## sample estimates:
##       cor
## 0.6721398

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$rur_pop_growth
## t = -19.109, df = 11710, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1914039 -0.1562770
## sample estimates:
##        cor
## -0.1738958

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$rur_perc
## t = -88.903, df = 11710, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.6454888 -0.6238604
## sample estimates:
##        cor
## -0.6347989

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$pri_exp_ppp
## t = 16.478, df = 7852, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1613581 0.2041132
## sample estimates:
##       cor
## 0.1828221

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$sec_exp_ppp
## t = 21.061, df = 7852, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2101931 0.2520620
## sample estimates:
##       cor
## 0.2312346

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$kvcval
## t = 58.9, df = 11222, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4716815 0.4999474
## sample estimates:
##       cor
## 0.4859415

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$hcfval
## t = 28.072, df = 1516, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5507235 0.6169892
## sample estimates:
##       cor
## 0.5848312

##
##  Pearson's product-moment correlation
##
```

```
## data:  df_death_rate$medageval and df_death_rate$gdp_pc_growth
## t = 3.3299, df = 11710, p-value = 0.0008714
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01265353 0.04884113
## sample estimates:
##        cor
## 0.03075741

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$poverty_perc
## t = -68.433, df = 7196, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6416727 -0.6136775
## sample estimates:
##        cor
## -0.6278782

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$medageval and df_death_rate$gdp_pc_ppp
## t = 78.912, df = 11100, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5874379 0.6112740
## sample estimates:
##       cor
## 0.5994889

## Warning in cor.test.default(df_death_rate$medageval, df_death_rate$countrygii, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval, df_death_rate$countrygdi, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval,
## df_death_rate$rur_pop_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval, df_death_rate$rur_perc, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval,
## df_death_rate$pri_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval,
## df_death_rate$sec_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval, df_death_rate$kvcval, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval, df_death_rate$hcfval, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval,
## df_death_rate$gdp_pc_growth, : Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(df_death_rate$medageval,
## df_death_rate$poverty_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$medageval, df_death_rate$gdp_pc_ppp, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$countrygii
## S = 8927083891, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.8915422

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$countrygdi
## S = 1429392853, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7421573

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$rur_pop_growth
## S = 4.4143e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6486116

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$rur_perc
## S = 4.4362e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6568086

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$pri_exp_ppp
## S = 4.3819e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4573207

##
```

```
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$sec_exp_ppp
## S = 3.4788e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5691645
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$kvcval
## S = 1.0289e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5634122
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$hcfval
## S = 194659103, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.6661043
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$gdp_pc_growth
## S = 2.527e+11, p-value = 1.133e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.05623004
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$poverty_perc
## S = 1.0356e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6661056
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$medageval and df_death_rate$gdp_pc_ppp
## S = 6.0974e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
```

```
##       rho
## 0.7326433

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$countrygii
## z = -58.099, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.7022332

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$countrygdi
## z = 45.532, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5358554

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$rur_pop_growth
## z = -74.621, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4598999

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$rur_perc
## z = -73.839, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4550834

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$pri_exp_ppp
## z = 40.555, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3055502

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$sec_exp_ppp
```

```
## z = 51.481, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3878623

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$kvcval
## z = 60.871, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3834976

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$hcfval
## z = 27.704, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.4753256

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$gdp_pc_growth
## z = 5.8072, p-value = 6.353e-09
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.03579978

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$poverty_perc
## z = -58.736, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4632441

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$medageval and df_death_rate$gdp_pc_ppp
## z = 83.183, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5269545
```

LEAB & no lag:

```
##        cor
## -0.7559529

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$countrygii
## t = -63.733, df = 3046, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7707698 -0.7403188
## sample estimates:
##        cor
## -0.7559529

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$countrygdi
## t = 43.347, df = 3214, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5851186 0.6287618
## sample estimates:
##       cor
## 0.6073983

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$rur_pop_growth
## t = -16.998, df = 11496, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1743592 -0.1386978
## sample estimates:
##        cor
## -0.1565795

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$rur_perc
## t = -112.64, df = 11496, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7329009 -0.7155202
## sample estimates:
##        cor
## -0.7243256

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$pri_exp_ppp
## t = 13.508, df = 7693, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1303168 0.1739694
## sample estimates:
##       cor
## 0.1522174

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$sec_exp_ppp
## t = 16.037, df = 7693, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1581488 0.2013915
## sample estimates:
##      cor
## 0.179857

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$kvcval
## t = 99.262, df = 10954, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6781577 0.6978772
## sample estimates:
##       cor
## 0.6881445

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$hcfval
## t = 37.995, df = 1516, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6717011 0.7233071
## sample estimates:
##       cor
## 0.6984108

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$gdp_pc_growth
## t = 6.9513, df = 11491, p-value = 3.816e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04648292 0.08289520
## sample estimates:
##       cor
## 0.0647106

##
```

```
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$poverty_perc
## t = -71.637, df = 6990, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6639710 -0.6369313
## sample estimates:
##        cor
## -0.6506573

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate$leabval and df_death_rate$gdp_pc_ppp
## t = 71.559, df = 10825, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5537599 0.5793379
## sample estimates:
##       cor
## 0.5666854

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$countrygii, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$countrygdi, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval,
## df_death_rate$rur_pop_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$rur_perc, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$pri_exp_ppp, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$sec_exp_ppp, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$kvcval, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$hcfval, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval,
## df_death_rate$gdp_pc_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$poverty_perc, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate$leabval, df_death_rate$gdp_pc_ppp, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$countrygii
```

```
## S = 8587481346, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.8195844

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$countrygdi
## S = 2453555180, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5574126

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$rur_pop_growth
## S = 3.9283e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.5505644

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$rur_perc
## S = 4.3918e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.733523

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$pri_exp_ppp
## S = 4.2637e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4385469

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$sec_exp_ppp
## S = 3.6546e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5187541

##
```

```
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$kvcval
## S = 7.0766e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.677135
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$hcfval
## S = 107109805, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8162762
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$gdp_pc_growth
## S = 2.3507e+11, p-value = 2.674e-14
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.07093331
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$poverty_perc
## S = 9.579e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6813778
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate$leabval and df_death_rate$gdp_pc_ppp
## S = 4.4971e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.787403
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$countrygii
## z = -51.391, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
```

```
##        tau
## -0.6210178

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$countrygdi
## z = 32.709, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3848543

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$rur_pop_growth
## z = -59.931, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.3726677

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$rur_perc
## z = -86.914, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.5404651

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$pri_exp_ppp
## z = 39.472, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## 0.300367

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$sec_exp_ppp
## z = 47.369, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3604497

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$kvcval
```

```
## z = 76.305, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## 0.486431
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$hcfval
## z = 36.374, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.6239264
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$gdp_pc_growth
## z = 6.8654, p-value = 6.631e-12
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.04271142
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$poverty_perc
## z = -60.094, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4807687
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate$leabval and df_death_rate$gdp_pc_ppp
## z = 92.271, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5917306
```

Now moving the Major lag set. #Death rate and major lag

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$countrygii
## t = 9.1488, df = 3046, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1287792 0.1978897
## sample estimates:
```

```
##       cor
## 0.1635351
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$countrygdi
## t = -14.67, df = 3214, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2826278 -0.2178338
## sample estimates:
##        cor
## -0.2505113
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$rur_pop_growth
## t = 8.8764, df = 7686, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.07855561 0.12281004
## sample estimates:
##       cor
## 0.1007327
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$rur_perc
## t = 33.479, df = 7686, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3370832 0.3761035
## sample estimates:
##       cor
## 0.3567489
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$pri_exp_ppp
## t = -1.2722, df = 4132, p-value = 0.2034
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05024289  0.01070322
## sample estimates:
##         cor
## -0.01978822
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$sec_exp_ppp
## t = -0.37772, df = 4132, p-value = 0.7057
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03635452  0.02461334
## sample estimates:
##          cor
## -0.005876051

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$kvcval
## t = -24.402, df = 5795, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3284122 -0.2817203
## sample estimates:
##        cor
## -0.3052497

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$hcfval
## t = -4.1079, df = 1021, p-value = 4.313e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.18733820 -0.06674006
## sample estimates:
##        cor
## -0.1275104

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$gdp_pc_growth
## t = -4.8441, df = 7934, p-value = 1.296e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07621431 -0.03234030
## sample estimates:
##         cor
## -0.05430351

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$poverty_perc
## t = 12.151, df = 3687, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1650014 0.2270617
## sample estimates:
##       cor
## 0.1962281

##
```

```
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$gdp_pc_ppp
## t = -26.736, df = 7407, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3172986 -0.2757631
## sample estimates:
##        cor
## -0.2966712
```

Spearman method

```
## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$countrygii, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$countrygdi, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$rur_pop_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$rur_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$pri_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$sec_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$kvcval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$hcfval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$gdp_pc_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$poverty_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$death_per_1000,
## df_death_rate_major_lag$gdp_pc_ppp, : Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$countrygii
## S = 4201703786, p-value = 1.257e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1097093

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$countrygdi
## S = 5765302978, p-value = 0.02337
```

```
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.03998077

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$rur_pop_growth
## S = 7.2864e+10, p-value = 0.0008891
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.03789635

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$rur_perc
## S = 5.2648e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3048257

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$pri_exp_ppp
## S = 1.2395e+10, p-value = 0.0007068
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.05265635

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$sec_exp_ppp
## S = 1.2097e+10, p-value = 0.079
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.02732207

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$kvcval
## S = 3.9015e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.2016242

##
##  Spearman's rank correlation rho
```

```
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$hcfval
## S = 195892191, p-value = 0.001729
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.09784717
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$gdp_pc_growth
## S = 9.0675e+10, p-value = 2.795e-15
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.0885169
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$poverty_perc
## S = 7661142244, p-value = 2.867e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.08437247
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$gdp_pc_ppp
## S = 8.5338e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.2589664
```

Kendall

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$countrygii
## z = 5.2021, p-value = 1.971e-07
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.06290502
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$countrygdi
## z = -0.87226, p-value = 0.3831
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
```

```
##        tau
## -0.0102692

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$rur_pop_growth
## z = 2.6142, p-value = 0.008943
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.01988815

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$rur_perc
## z = 27.665, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2104694

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$pri_exp_ppp
## z = -3.2537, p-value = 0.001139
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##         tau
## -0.03385712

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$sec_exp_ppp
## z = -1.9525, p-value = 0.05087
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##         tau
## -0.02031748

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$kvcval
## z = -15.665, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.1374176

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$hcfval
```

```
## z = -3.9345, p-value = 8.339e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##         tau
## -0.08275308
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$gdp_pc_growth
## z = -8.0186, p-value = 1.069e-15
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##         tau
## -0.06006234
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$poverty_perc
## z = 4.9869, p-value = 6.135e-07
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.05504317
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$death_per_1000 and df_death_rate_major_lag$gdp_pc_ppp
## z = -22.958, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.1780998
```

Median age and major lag

```
##       cor
## -0.894784
```

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

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$countrygii
## t = -110.6, df = 3046, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9016441 -0.8874737
## sample estimates:
##       cor
## -0.894784
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$countrygdi
## t = 51.464, df = 3214, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6527404 0.6906584
## sample estimates:
##      cor
## 0.6721398

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$rur_pop_growth
## t = -13.693, df = 5950, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1993079 -0.1500477
## sample estimates:
##        cor
## -0.1747872

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$rur_perc
## t = -67.771, df = 5950, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6741310 -0.6454469
## sample estimates:
##        cor
## -0.6600294

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$pri_exp_ppp
## t = 8.9714, df = 4002, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1099126 0.1706443
## sample estimates:
##      cor
## 0.1404106

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$sec_exp_ppp
## t = 11.399, df = 4002, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1471608 0.2071664
## sample estimates:
```

```
##       cor
## 0.1773284

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$kvcval
## t = 52.912, df = 5702, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5561863 0.5910059
## sample estimates:
##       cor
## 0.5738554

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$hcfval
## t = 17.572, df = 1021, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4333870 0.5275882
## sample estimates:
##       cor
## 0.4818787

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$gdp_pc_growth
## t = 6.1848, df = 5950, p-value = 6.633e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05462855 0.10511576
## sample estimates:
##        cor
## 0.07992341

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$poverty_perc
## t = -51.795, df = 3656, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6688658 -0.6314651
## sample estimates:
##        cor
## -0.6505597

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$gdp_pc_ppp
## t = 51.853, df = 5640, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5502383 0.5855872
## sample estimates:
##       cor
## 0.5681748
```

```
## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$countrygii, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$countrygdi, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$rur_pop_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$rur_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$pri_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$sec_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$kvcval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$hcfval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$gdp_pc_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$poverty_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$medageval,
## df_death_rate_major_lag$gdp_pc_ppp, : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$countrygii
## S = 8927083891, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.8915422
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$countrygdi
## S = 1429392853, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7421573
```

```
##
```

```
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$rur_pop_growth
## S = 5.7697e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6417821

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$rur_perc
## S = 5.9299e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6873698

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$pri_exp_ppp
## S = 5766136370, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4610432

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$sec_exp_ppp
## S = 4737669213, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5571733

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$kvcval
## S = 1.1588e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.6253452

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$hcfval
## S = 84229397, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
```

```
##       rho
## 0.5279495

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$gdp_pc_growth
## S = 3.0461e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1332108

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$poverty_perc
## S = 1.3703e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6797012

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$gdp_pc_ppp
## S = 6.774e+09, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7736918

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$countrygii
## z = -58.099, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.7022332

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$countrygdi
## z = 45.532, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5358554

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$rur_pop_growth
```

```
## z = -53.158, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4595955

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$rur_perc
## z = -56.033, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4844633

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$pri_exp_ppp
## z = 28.976, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3063068

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$sec_exp_ppp
## z = 35.564, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3759548

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$kvcval
## z = 50.271, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.4445109

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$hcfval
## z = 17.676, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3695007

##
```

```
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$gdp_pc_growth
## z = 10.035, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.08680281
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$poverty_perc
## z = -42.945, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4758389
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$medageval and df_death_rate_major_lag$gdp_pc_ppp
## z = 64.317, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5719103
```

Ninth one. LEAB and minor lag.

```
##        cor
## -0.7559529
```

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

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$countrygii
## t = -63.733, df = 3046, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7707698 -0.7403188
## sample estimates:
##        cor
## -0.7559529
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$countrygdi
## t = 43.347, df = 3214, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5851186 0.6287618
```

```
## sample estimates:
##       cor
## 0.6073983

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$rur_pop_growth
## t = -12.335, df = 5827, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1844395 -0.1344003
## sample estimates:
##        cor
## -0.1595223

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$rur_perc
## t = -69.447, df = 5827, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6867554 -0.6586541
## sample estimates:
##        cor
## -0.6729475

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$pri_exp_ppp
## t = 7.13, df = 3868, p-value = 1.192e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08268617 0.14488497
## sample estimates:
##       cor
## 0.1138972

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$sec_exp_ppp
## t = 8.5732, df = 3868, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1055021 0.1673436
## sample estimates:
##       cor
## 0.1365559

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$kvcval
```

```
## t = 58.859, df = 5515, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6046636 0.6370870
## sample estimates:
##      cor
## 0.621141

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$hcfval
## t = 27.4, df = 988, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6201885 0.6911134
## sample estimates:
##       cor
## 0.6571028

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$gdp_pc_growth
## t = 6.4028, df = 5825, p-value = 1.644e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05804688 0.10904193
## sample estimates:
##        cor
## 0.08359914

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$poverty_perc
## t = -53.917, df = 3530, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6897132 -0.6535224
## sample estimates:
##        cor
## -0.6720188

##
##  Pearson's product-moment correlation
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$gdp_pc_ppp
## t = 49.92, df = 5471, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5409399 0.5773539
## sample estimates:
##       cor
## 0.5594167
```

```
## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$countrygii, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$countrygdi, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$rur_pop_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$rur_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$pri_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$sec_exp_ppp, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$kvcval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$hcfval, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$gdp_pc_growth, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$poverty_perc, : Cannot compute exact p-value with ties

## Warning in cor.test.default(df_death_rate_major_lag$leabval,
## df_death_rate_major_lag$gdp_pc_ppp, : Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$countrygii
## S = 8587481346, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.8195844

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$countrygdi
## S = 2453555180, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5574126

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$rur_pop_growth
## S = 4.9973e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
```

```
##       rho
## -0.513939

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$rur_perc
## S = 5.6011e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6968479

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$pri_exp_ppp
## S = 5674269439, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4126076

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$sec_exp_ppp
## S = 5063735178, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4758092

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$kvcval
## S = 1.0568e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.622383

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$hcfval
## S = 39752634, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7541829

##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$gdp_pc_growth
```

```
## S = 2.7852e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1553629
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$poverty_perc
## S = 1.2382e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6860844
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$gdp_pc_ppp
## S = 5627245893, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7940457
```

Kendall method

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$countrygii
## z = -51.391, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.6210178
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$countrygdi
## z = 32.709, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.3848543
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$rur_pop_growth
## z = -40.661, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.3551639
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$rur_perc
## z = -57.866, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.5054573
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$pri_exp_ppp
## z = 25.661, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2758835
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$sec_exp_ppp
## z = 30.125, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## 0.323883
```

```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$kvcval
## z = 47.948, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.4310439
```

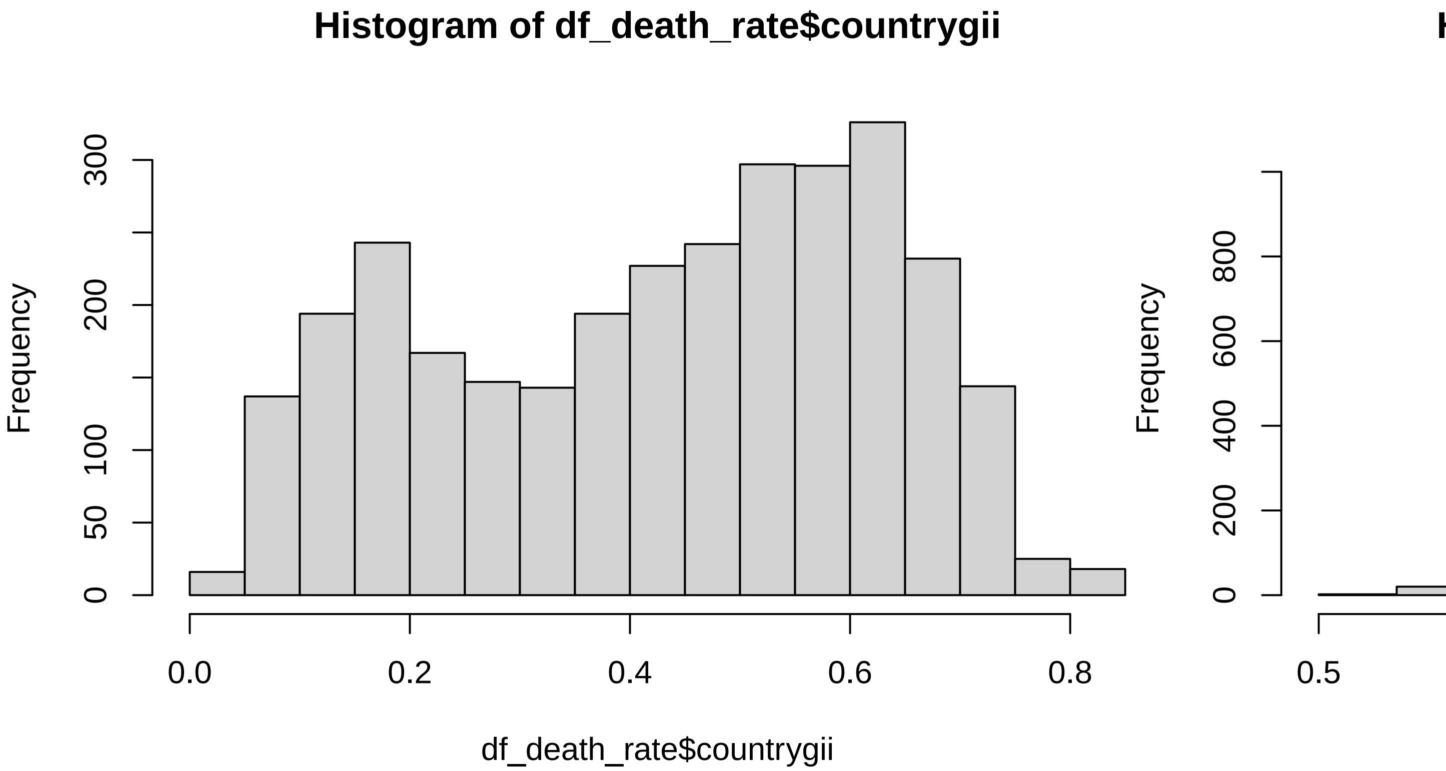
```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$hcfval
## z = 26.785, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5691292
```

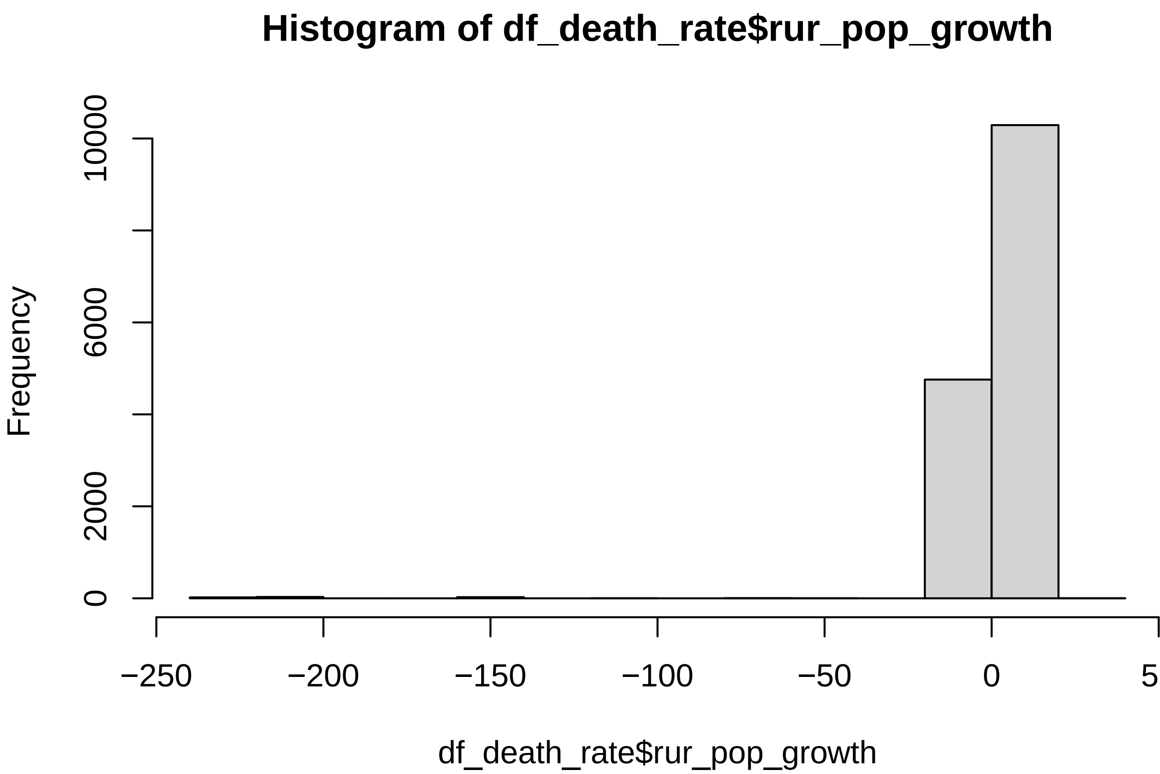
```
##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$gdp_pc_growth
## z = 11.472, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
```

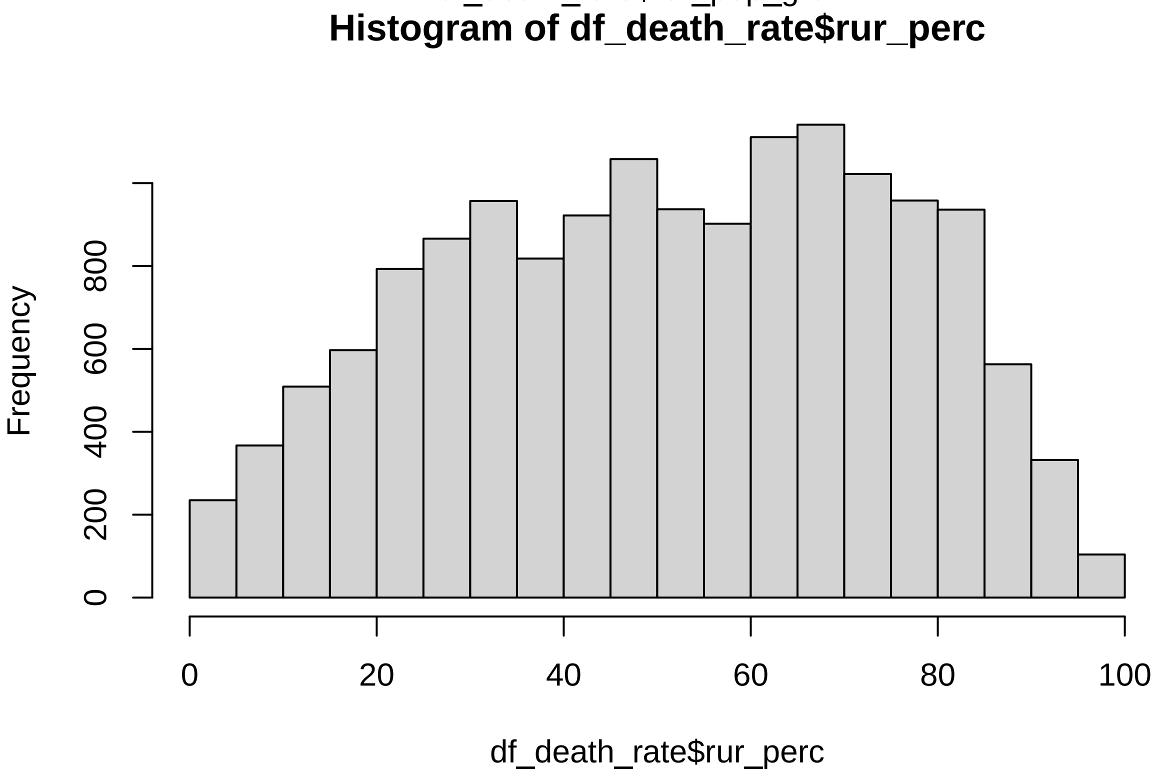
```
## sample estimates:
##       tau
## 0.1002674

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$poverty_perc
## z = -43.717, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## -0.492865

##
##  Kendall's rank correlation tau
##
## data:  df_death_rate_major_lag$leabval and df_death_rate_major_lag$gdp_pc_ppp
## z = 65.673, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5928341
```

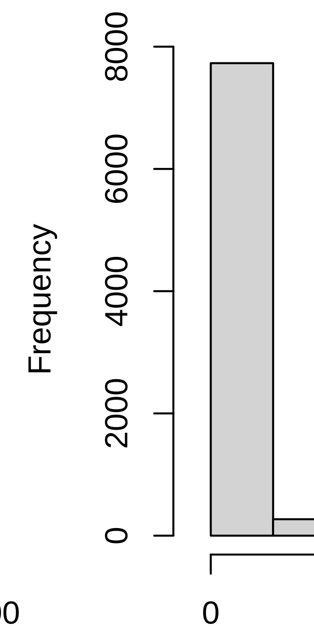
Subsequent steps are doing the regression to see the impact of each of the dependent variables when applied together. We will first look for death rate followed by life expectancy at birth and median age. End result will be a prediction model.

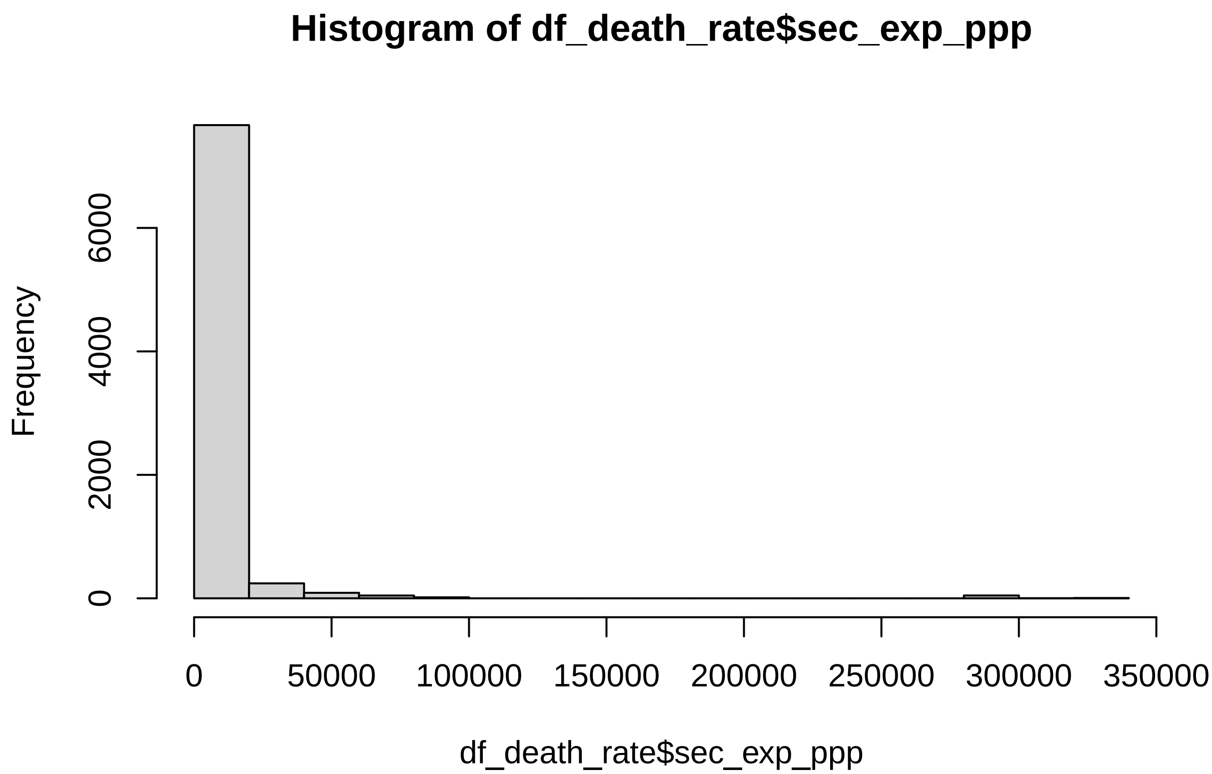
First checking the histogram to see if any variables need log transformation.

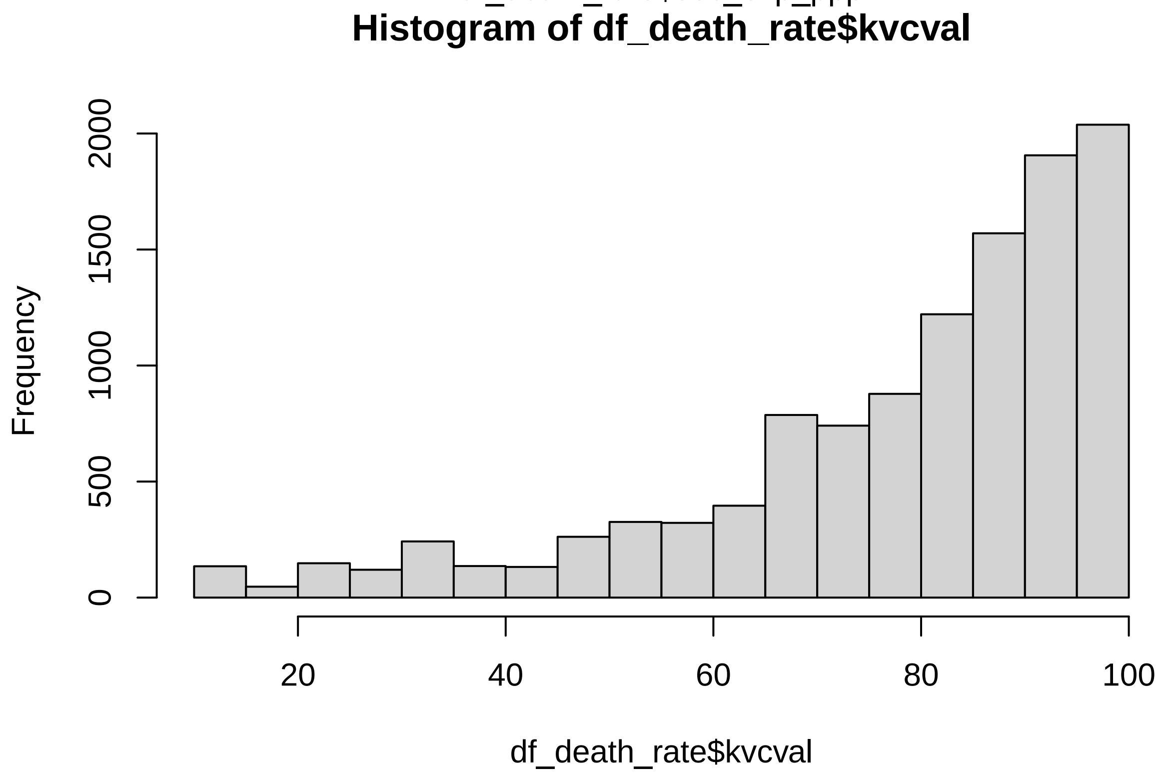


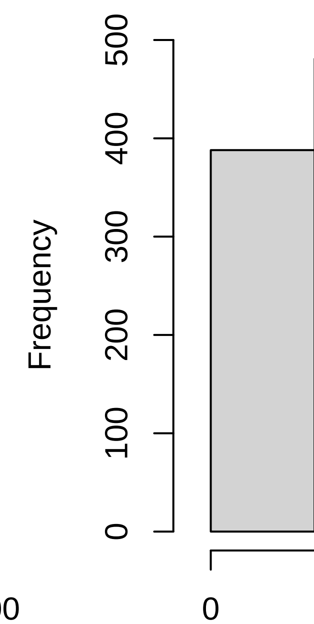

Histogram of df_death_rate$rur_pop_growth
Frequency
0
2000
6000
10000
−250
−200
−150
−100
−50
0
50
df_death_rate$rur_pop_growth

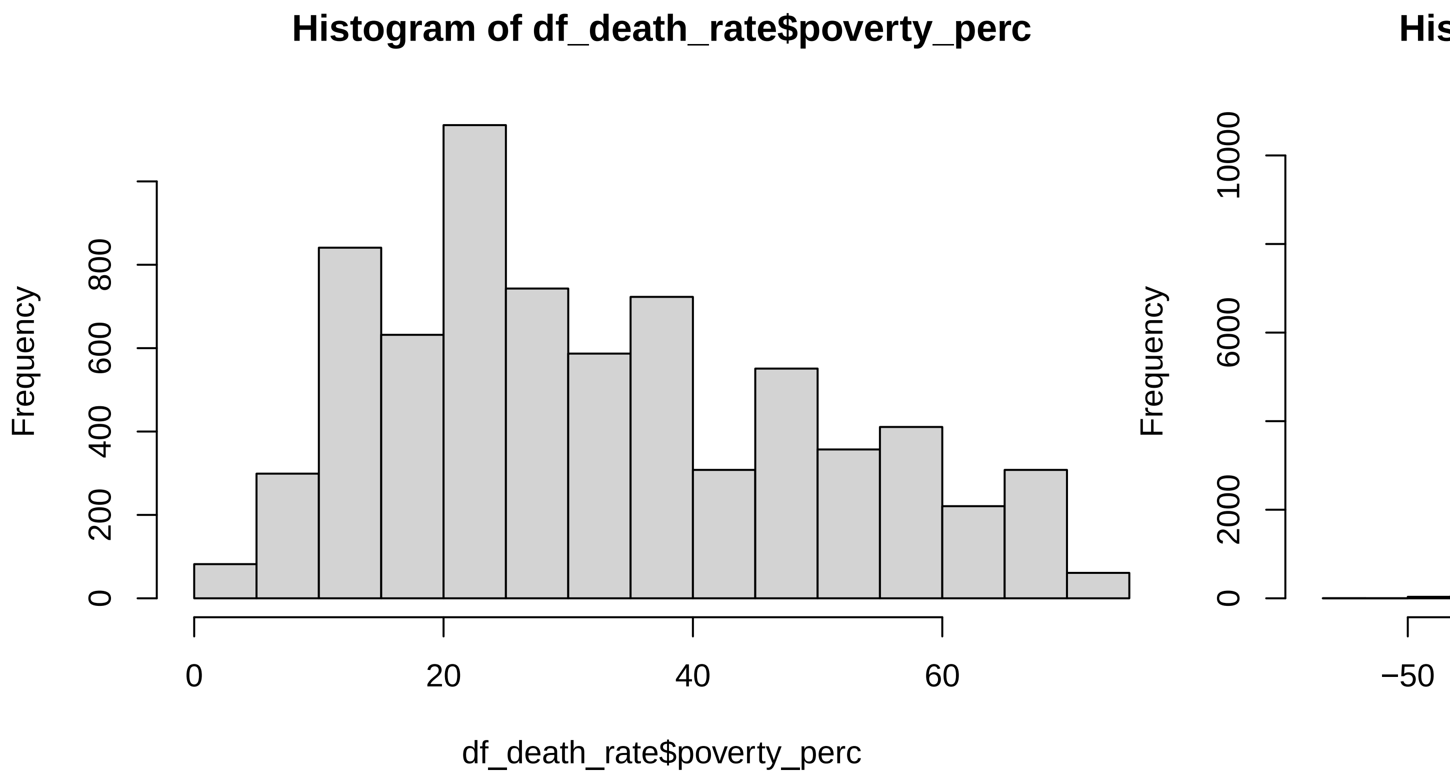

Histogram of df_death_rate$rur_perc
Frequency
0
200
400
600
800
0
20
40
60
80
100
df_death_rate$rur_perc

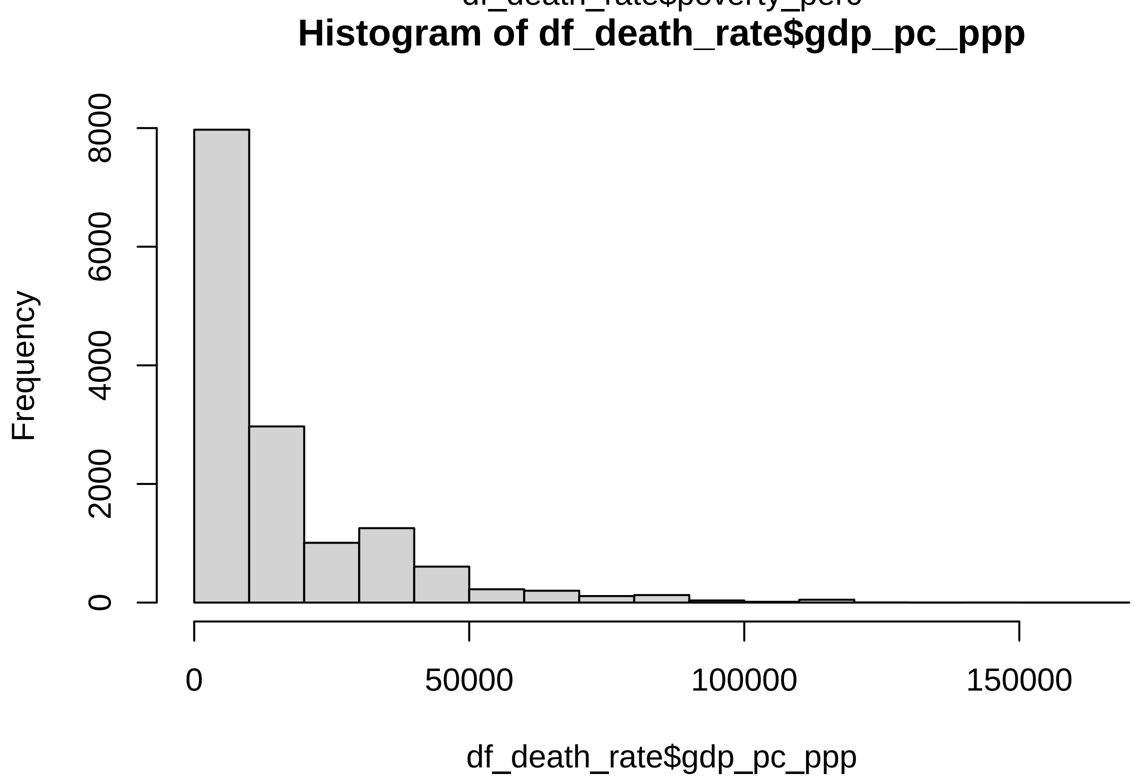

Frequency
0
2000
4000
6000
8000
0


Histogram of df_death_rate$sec_exp_ppp
Frequency
0
2000
4000
6000
0
50000
100000
150000
200000
250000
300000
350000
df_death_rate$sec_exp_ppp


Histogram of df_death_rate$kvcval
Frequency
0
500
1000
1500
2000
20
40
60
80
100
df_death_rate$kvcval


Frequency
0
100
200
300
400
500
0





Convert the three right skewed parameters to log scale.

```
df_death_rate$pri_exp_ppp = log(df_death_rate$pri_exp_ppp)
df_death_rate$sec_exp_ppp = log(df_death_rate$sec_exp_ppp)
```

```
df_death_rate$gdp_pc_ppp = log(df_death_rate$gdp_pc_ppp)

df_death_rate_minor_lag$pri_exp_ppp = log(df_death_rate_minor_lag$pri_exp_ppp)
df_death_rate_minor_lag$sec_exp_ppp = log(df_death_rate_minor_lag$sec_exp_ppp)
df_death_rate_minor_lag$gdp_pc_ppp = log(df_death_rate_minor_lag$gdp_pc_ppp)


df_death_rate_major_lag$pri_exp_ppp = log(df_death_rate_major_lag$pri_exp_ppp)
df_death_rate_major_lag$sec_exp_ppp = log(df_death_rate_major_lag$sec_exp_ppp)
df_death_rate_major_lag$gdp_pc_ppp = log(df_death_rate_major_lag$gdp_pc_ppp)
```

Next step check the multiple linear regression model and check for collinearity of factors.

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
fitlm <- lm(death_per_1000 ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       2.156078       7.479743       2.958056       1.168472       1.171214
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      12.021314      13.277321       1.345733       8.030916       3.597925
##    sec_exp_ppp           year
##       8.185932       1.602734
```

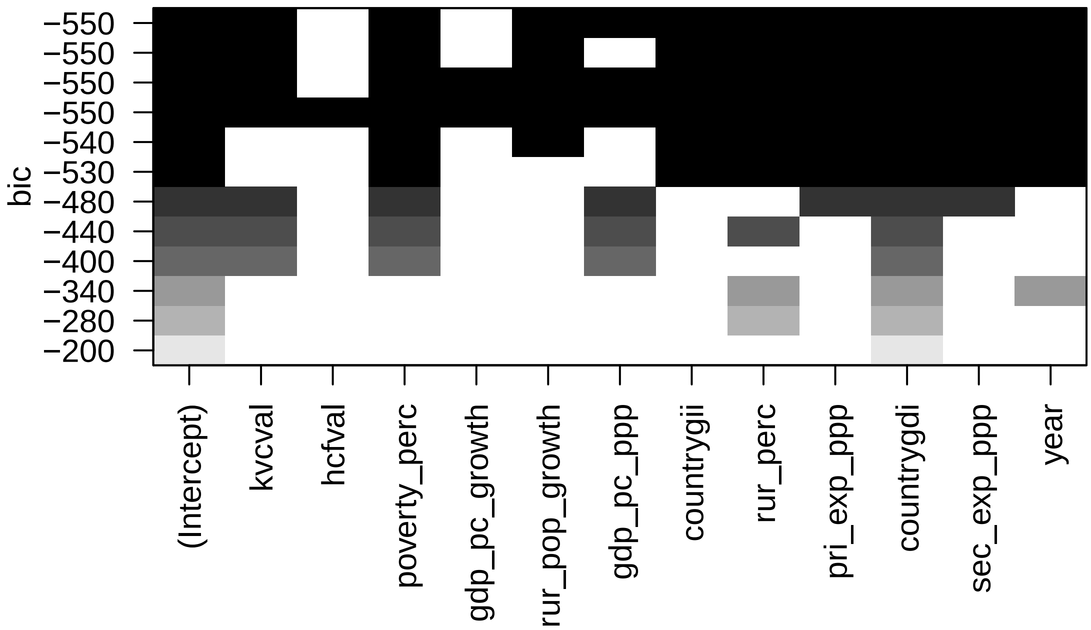
We can see that gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```
library(leaps)
fit.full = regsubsets(death_per_1000 ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + g

fit.full
```

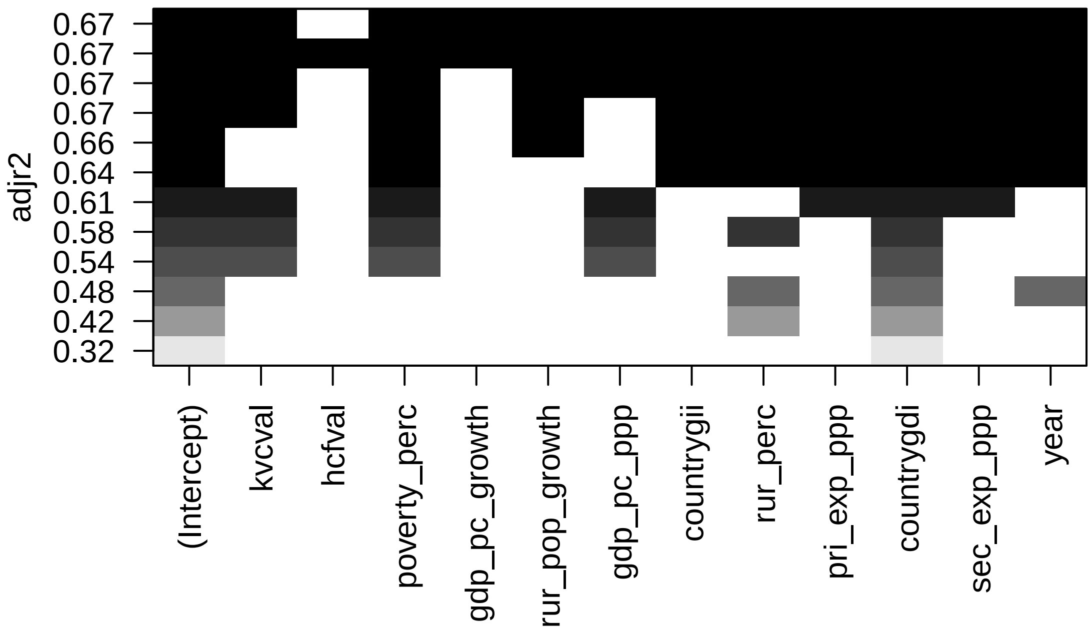
```
## Subset selection object
## Call: regsubsets.formula(death_per_1000 ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
## gdp_pc_ppp         FALSE      FALSE
## countrygii         FALSE      FALSE
## rur_perc           FALSE      FALSE
## pri_exp_ppp        FALSE      FALSE
## countrygdi         FALSE      FALSE
```

```
## sec_exp_ppp          FALSE      FALSE
## year                 FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 11
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 10
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         kvcval   poverty_perc  gdp_pc_growth rur_pop_growth
##    96.25562086    -0.04667221    -0.14075616    -0.03407750    -0.27335038
##      gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##    -0.81040958     6.74736285     0.04862583    -0.77353981    43.29337105
##     sec_exp_ppp           year
##     1.08205541    -0.05945198
```

```r
coef(fit.full,10)
```

```
##    (Intercept)         kvcval   poverty_perc rur_pop_growth     gdp_pc_ppp
##    95.09497929    -0.04636941    -0.13991026    -0.27232421    -0.85018703
##     countrygii       rur_perc    pri_exp_ppp     countrygdi    sec_exp_ppp
##     6.30240993     0.04831054    -0.74465577    42.57306045     1.06842188
##           year
##    -0.05838482
```

We will create the regrssion models with 9 variables.

kvcval , poverty_perc, rur_pop_growth, gdp_pc_ppp, countrygii,rur_perc, pri_exp_ppp ,countrygdi,sec_exp_ppp

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(death_per_1000 ~ kvcval +  poverty_perc + rur_pop_growth + gdp_pc_ppp +countrygii + rur_perc
summary(fitlm)
```

```
##
## Call:
## lm(formula = death_per_1000 ~ kvcval + poverty_perc + rur_pop_growth +
##     gdp_pc_ppp + countrygii + rur_perc + pri_exp_ppp + countrygdi +
##     sec_exp_ppp + year, data = df_death_rate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3416 -2.1642 -0.4454  1.6767 22.6192
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    132.012156  21.599085   6.112 1.18e-09 ***
## kvcval          -0.098103   0.007410 -13.239  < 2e-16 ***
## poverty_perc     0.056169   0.006768   8.299  < 2e-16 ***
## rur_pop_growth  -0.551491   0.059242  -9.309  < 2e-16 ***
## gdp_pc_ppp      -1.464077   0.166085  -8.815  < 2e-16 ***
## countrygii     -14.130930   0.784689 -18.008  < 2e-16 ***
## rur_perc         0.040487   0.006233   6.495 1.04e-10 ***
## pri_exp_ppp     -1.047509   0.096434 -10.862  < 2e-16 ***
## countrygdi      -7.734811   1.231474  -6.281 4.13e-10 ***
## sec_exp_ppp      0.620597   0.094029   6.600 5.26e-11 ***
## year            -0.044092   0.010770  -4.094 4.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.092 on 1981 degrees of freedom
##   (13807 observations deleted due to missingness)
## Multiple R-squared:  0.3581, Adjusted R-squared:  0.3549
## F-statistic: 110.5 on 10 and 1981 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```
df_death_rate <- na.omit(df_death_rate)
fitModel <- (death_per_1000 ~ kvcval +  poverty_perc + rur_pop_growth + gdp_pc_ppp +countrygii + rur_pe
n = nrow(df_death_rate)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate[test,])
}

sum((allpredictedCV-df_death_rate$death_per_1000)^2)/n
```

```
## [1] 1.740599
```

```
**********************************
```

Repeat 2: Median age and no lag.

Next step check the multiple linear regression model and check for collinearity of factors.

```
library(car)
fitlm <- lm(medageval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +coun
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       2.156078       7.479743       2.958056       1.168472       1.171214
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      12.021314      13.277321       1.345733       8.030916       3.597925
##    sec_exp_ppp           year
##       8.185932       1.602734
```

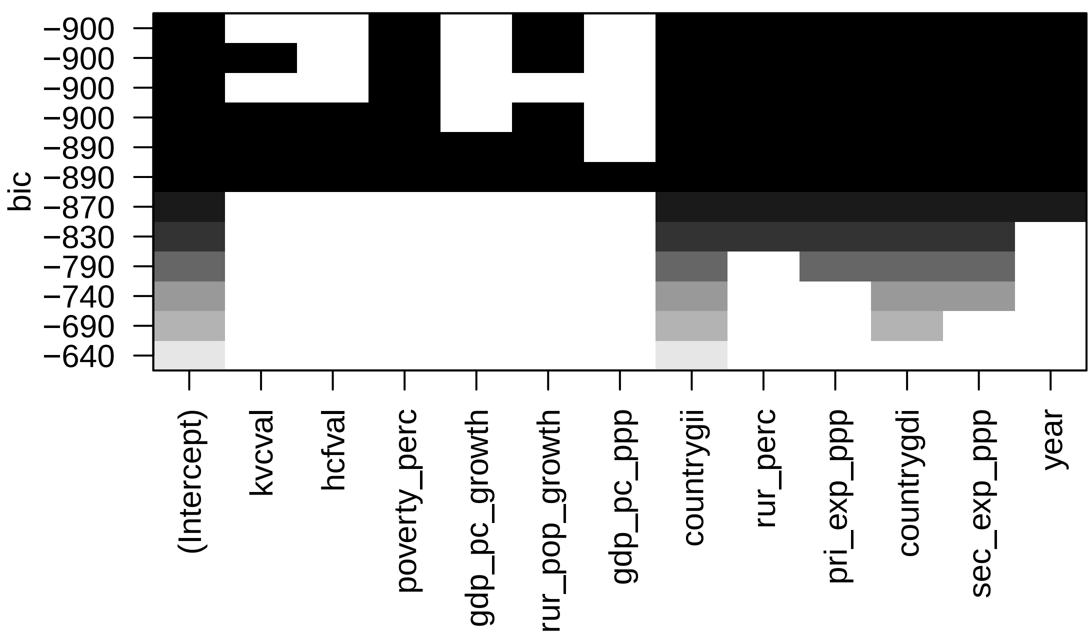
We can see that gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```
library(leaps)
fit.full = regsubsets(medageval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_

fit.full
```

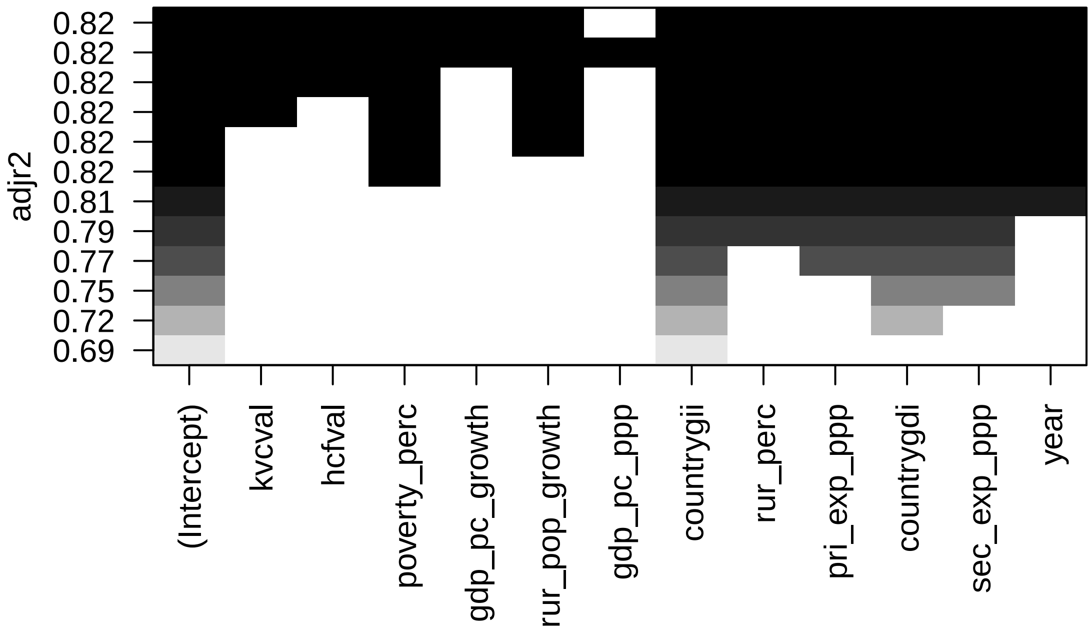
```
## Subset selection object
## Call: regsubsets.formula(medageval ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate, nvmax = 12)
## 12 Variables  (and intercept)
```

```
##                  Forced in Forced out
## kvcval               FALSE      FALSE
## hcfval               FALSE      FALSE
## poverty_perc         FALSE      FALSE
## gdp_pc_growth        FALSE      FALSE
## rur_pop_growth       FALSE      FALSE
## gdp_pc_ppp           FALSE      FALSE
## countrygii           FALSE      FALSE
## rur_perc             FALSE      FALSE
## pri_exp_ppp          FALSE      FALSE
## countrygdi           FALSE      FALSE
## sec_exp_ppp          FALSE      FALSE
## year                 FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 11
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 8
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         kvcval         hcfval   poverty_perc  gdp_pc_growth
##  -2.069754e+02  -2.720911e-02   2.356876e-04  -1.256569e-01  -3.771836e-02
## rur_pop_growth     countrygii       rur_perc    pri_exp_ppp     countrygdi
##  -2.561139e-01  -1.397427e+01   7.093583e-02  -1.267001e+00   3.937405e+01
##    sec_exp_ppp           year
##   1.729259e+00   1.030735e-01
```

```r
coef(fit.full,9)
```

```
##    (Intercept)         kvcval   poverty_perc rur_pop_growth     countrygii
##  -223.23524587    -0.03364138    -0.12176864    -0.25729790   -16.71957170
##       rur_perc    pri_exp_ppp     countrygdi    sec_exp_ppp           year
##     0.06683968    -1.23365186    35.85932659     1.69434571     0.11371211
```

We will create the regrssion models with 9 variables.

kvcval , poverty_perc, rur_pop_growth, gdp_pc_ppp, countrygii,rur_perc, pri_exp_ppp ,countrygdi,sec_exp_ppp

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(medageval ~ kvcval  + poverty_perc + rur_pop_growth  +countrygii + rur_perc +pri_exp_ppp +c
summary(fitlm)
```

```
##
## Call:
## lm(formula = medageval ~ kvcval + poverty_perc + rur_pop_growth +
##     countrygii + rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp +
```

```
##     year, data = df_death_rate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7638 -1.3351  0.2979  1.4441  4.1879
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -223.23525   28.92010  -7.719 5.68e-14 ***
## kvcval           -0.03364    0.01482  -2.270  0.02363 *
## poverty_perc     -0.12177    0.02083  -5.844 8.78e-09 ***
## rur_pop_growth   -0.25730    0.08215  -3.132  0.00183 **
## countrygii      -16.71957    1.51021 -11.071  < 2e-16 ***
## rur_perc          0.06684    0.00853   7.835 2.48e-14 ***
## pri_exp_ppp      -1.23365    0.18191  -6.782 3.12e-11 ***
## countrygdi       35.85933    3.42908  10.457  < 2e-16 ***
## sec_exp_ppp       1.69435    0.17358   9.761  < 2e-16 ***
## year              0.11371    0.01480   7.684 7.29e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.084 on 542 degrees of freedom
## Multiple R-squared:  0.8255, Adjusted R-squared:  0.8226
## F-statistic: 284.9 on 9 and 542 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```
df_death_rate <- na.omit(df_death_rate)
fitModel <- (medageval ~ kvcval +  poverty_perc + rur_pop_growth  +countrygii + rur_perc +pri_exp_ppp +
n = nrow(df_death_rate)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate[test,])
}

sum((allpredictedCV-df_death_rate$medageval)^2)/n
```

```
## [1] 4.431936
```

****

Repeat 3 - LEAB and no lag

Next step check the multiple linear regression model and check for collinearity of factors.

```
library(car)
fitlm <- lm(leabval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +country
vif(fitlm)
```

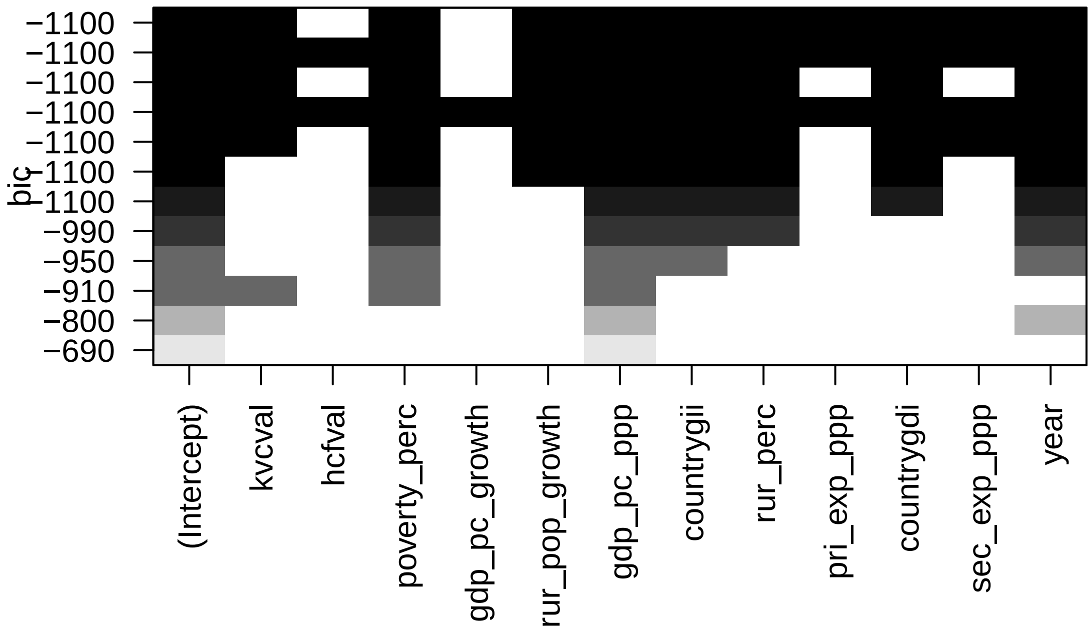
```
##          kvcval          hcfval    poverty_perc   gdp_pc_growth rur_pop_growth
##        2.156078        7.479743        2.958056        1.168472        1.171214
##      gdp_pc_ppp      countrygii        rur_perc     pri_exp_ppp     countrygdi
##       12.021314       13.277321        1.345733        8.030916        3.597925
##     sec_exp_ppp            year
##        8.185932        1.602734
```

We can see that gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

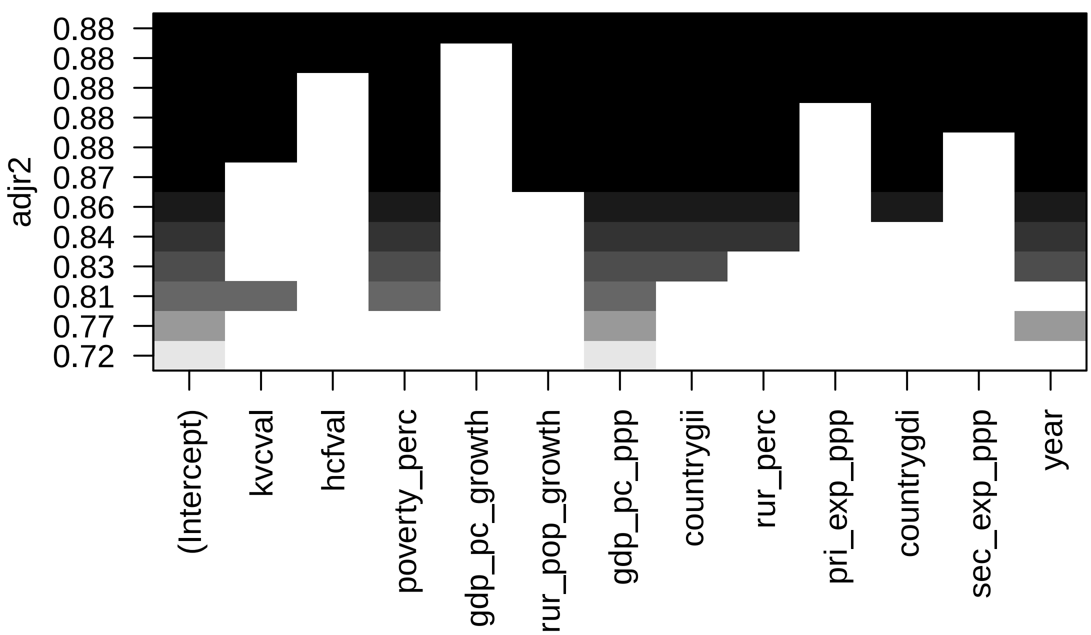
```r
library(leaps)
fit.full = regsubsets(leabval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_pp

fit.full
```

```
## Subset selection object
## Call: regsubsets.formula(leabval ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
## gdp_pc_ppp         FALSE      FALSE
## countrygii         FALSE      FALSE
## rur_perc           FALSE      FALSE
## pri_exp_ppp        FALSE      FALSE
## countrygdi         FALSE      FALSE
## sec_exp_ppp        FALSE      FALSE
## year               FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 12
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 10
```

```r
coef(fit.full,12)
```

```
##    (Intercept)         kvcval         hcfval   poverty_perc  gdp_pc_growth
##  -1.574848e+02   4.152833e-02  -2.312566e-04   1.717839e-01  -1.939673e-02
## rur_pop_growth     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp
```

```
##   3.184414e-01   4.235329e+00  -1.929822e+01  -5.471554e-02   4.036295e-01
##     countrygdi    sec_exp_ppp           year
##  -2.290621e+01  -4.683759e-01   1.061627e-01
```

```r
coef(fit.full,10)
```

```
##    (Intercept)         kvcval   poverty_perc rur_pop_growth     gdp_pc_ppp
##  -145.69635594     0.04276016     0.16968855     0.32831731     3.70916403
##     countrygii       rur_perc    pri_exp_ppp     countrygdi    sec_exp_ppp
##   -19.10239001    -0.05449845     0.42160042   -22.68570338    -0.46825793
##           year
##     0.10247267
```

We will create the regrssion models with 9 variables.

kvcval , poverty_perc, rur_pop_growth, gdp_pc_ppp, countrygii,rur_perc, pri_exp_ppp ,countrygdi,sec_exp_ppp

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(leabval ~ kvcval +  poverty_perc + rur_pop_growth + gdp_pc_ppp +countrygii + rur_perc +pri_
summary(fitlm)
```

```
##
## Call:
## lm(formula = leabval ~ kvcval + poverty_perc + rur_pop_growth +
##     gdp_pc_ppp + countrygii + rur_perc + pri_exp_ppp + countrygdi +
##     sec_exp_ppp + year, data = df_death_rate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8611 -0.8122 -0.1287  0.9519  4.2620
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.457e+02  1.772e+01  -8.220 1.52e-15 ***
## kvcval          4.276e-02  9.828e-03   4.351 1.62e-05 ***
## poverty_perc    1.697e-01  1.272e-02  13.337  < 2e-16 ***
## rur_pop_growth  3.283e-01  5.039e-02   6.515 1.66e-10 ***
## gdp_pc_ppp      3.709e+00  3.161e-01  11.733  < 2e-16 ***
## countrygii     -1.910e+01  1.521e+00 -12.559  < 2e-16 ***
## rur_perc       -5.450e-02  5.675e-03  -9.604  < 2e-16 ***
## pri_exp_ppp     4.216e-01  1.111e-01   3.795 0.000164 ***
## countrygdi     -2.269e+01  2.518e+00  -9.011  < 2e-16 ***
## sec_exp_ppp    -4.683e-01  1.063e-01  -4.406 1.27e-05 ***
## year            1.025e-01  9.662e-03  10.606  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.273 on 541 degrees of freedom
## Multiple R-squared:  0.8835, Adjusted R-squared:  0.8814
## F-statistic: 410.3 on 10 and 541 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```r
df_death_rate <- na.omit(df_death_rate)
fitModel <- (leabval ~ kvcval +  poverty_perc + rur_pop_growth + gdp_pc_ppp +countrygii + rur_perc +pri_
n = nrow(df_death_rate)
```

```r
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate[test,])
}

sum((allpredictedCV-df_death_rate$leabval)^2)/n
```

```
## [1] 1.670946
```

****

Try 4:

Next step check the multiple linear regression model and check for collinearity of factors.

```r
library(car)
fitlm <- lm(death_per_1000 ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       2.860844       8.220773       3.692408       1.434765       1.191501
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      13.272494      14.793223       1.338813       4.073877       3.800399
##    sec_exp_ppp           year
##       4.649380       2.021295
```

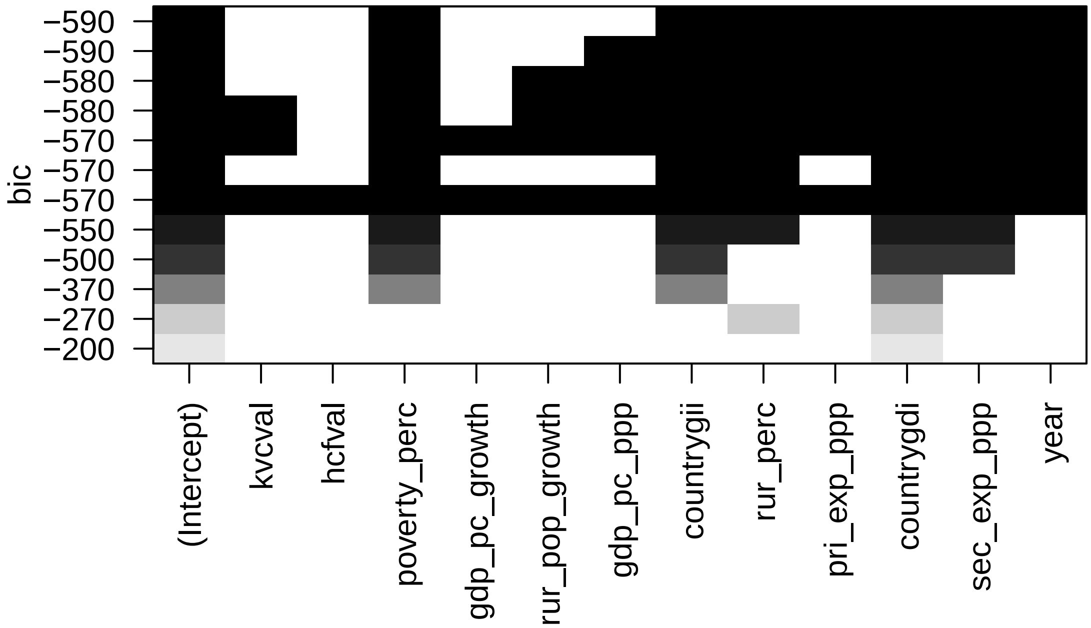
We can see that gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```r
library(leaps)
fit.full = regsubsets(death_per_1000 ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + g

fit.full
```

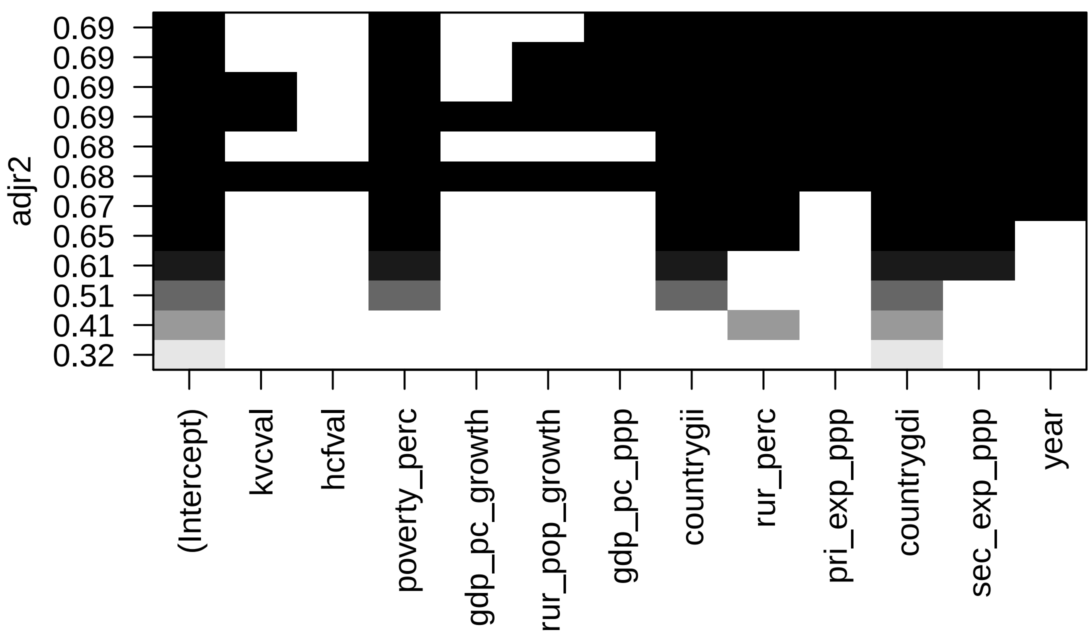
```
## Subset selection object
## Call: regsubsets.formula(death_per_1000 ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate_minor_lag, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
## gdp_pc_ppp         FALSE      FALSE
## countrygii         FALSE      FALSE
## rur_perc           FALSE      FALSE
```

```
## pri_exp_ppp         FALSE      FALSE
## countrygdi          FALSE      FALSE
## sec_exp_ppp         FALSE      FALSE
## year                FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 8
```

```
which.min(fit.full.summary$bic)
```

```
## [1] 7
```

```
coef(fit.full,11)
```

```
##    (Intercept)         kvcval   poverty_perc  gdp_pc_growth rur_pop_growth
##   74.118818844   -0.008431347   -0.187234877    0.009455478   -0.053971209
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##   -0.476350640   11.077581894    0.036693266   -0.368161047   45.452208426
##    sec_exp_ppp           year
##    0.834726737   -0.053231867
```

```
coef(fit.full,7)
```

```
##  (Intercept) poverty_perc   countrygii     rur_perc  pri_exp_ppp   countrygdi
##  60.85904116  -0.19021082  13.31442292   0.03891958  -0.40062615  47.98052625
##  sec_exp_ppp         year
##   0.86594438  -0.05082482
```

We will create the regrssion models with 7 variables.

kvcval poverty_perc countrygii rur_perc countrygdi sec_exp_ppp year

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```
fitlm <- lm(death_per_1000 ~ kvcval +  poverty_perc  +countrygii + rur_perc +countrygdi + sec_exp_ppp +
summary(fitlm)
```

```
##
## Call:
## lm(formula = death_per_1000 ~ kvcval + poverty_perc + countrygii +
##     rur_perc + countrygdi + sec_exp_ppp + year, data = df_death_rate_minor_lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2072 -2.3476 -0.5847  1.9273 28.6224
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  176.489252  23.644244   7.464 1.25e-13 ***
## kvcval        -0.069240   0.007455  -9.287  < 2e-16 ***
## poverty_perc   0.049547   0.006616   7.489 1.03e-13 ***
## countrygii   -10.745604   0.809433 -13.275  < 2e-16 ***
## rur_perc       0.057057   0.004906  11.630  < 2e-16 ***
## countrygdi    -6.138317   1.341727  -4.575 5.06e-06 ***
## sec_exp_ppp   -0.205151   0.045609  -4.498 7.25e-06 ***
## year          -0.077025   0.011857  -6.496 1.04e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.42 on 1984 degrees of freedom
##   (8627 observations deleted due to missingness)
## Multiple R-squared:  0.2132, Adjusted R-squared:  0.2104
## F-statistic:  76.8 on 7 and 1984 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```
df_death_rate_minor_lag <- na.omit(df_death_rate_minor_lag)
fitModel <- (death_per_1000 ~  +  poverty_perc  +countrygii + rur_perc +countrygdi + sec_exp_ppp + year)
n = nrow(df_death_rate_minor_lag)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate_minor_lag,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate_minor_lag[test,])
}

sum((allpredictedCV-df_death_rate_minor_lag$death_per_1000)^2)/n
```

```
## [1] 1.755814
```

****

Set 5 -

```
library(car)
fitlm <- lm(medageval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +coun
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       2.860844       8.220773       3.692408       1.434765       1.191501
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      13.272494      14.793223       1.338813       4.073877       3.800399
##    sec_exp_ppp           year
##       4.649380       2.021295
```

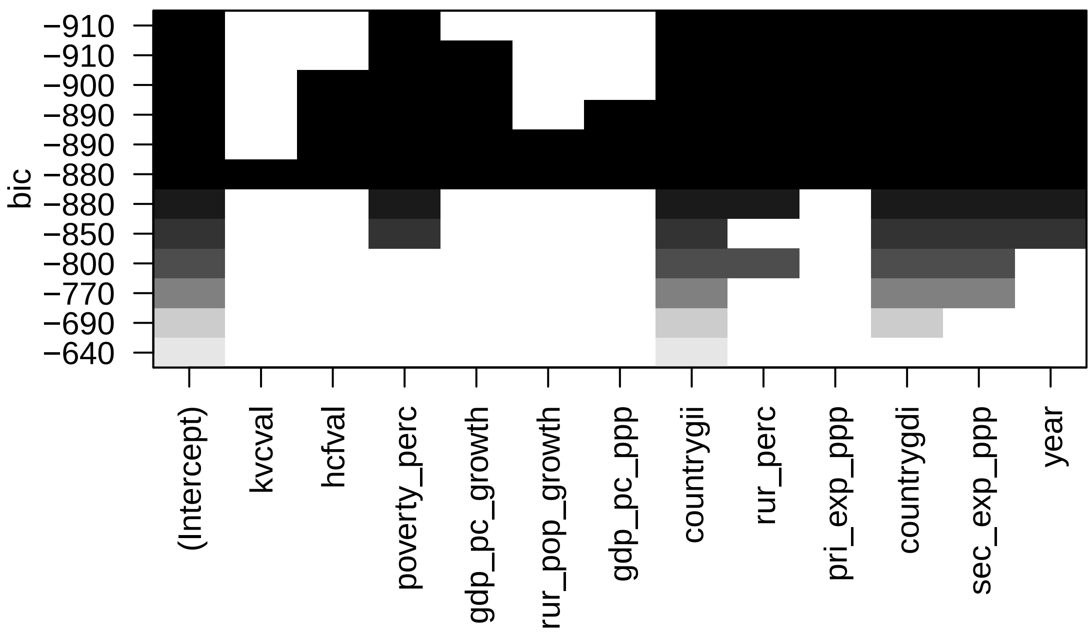
We can see that gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```
library(leaps)
fit.full = regsubsets(medageval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_

fit.full
```

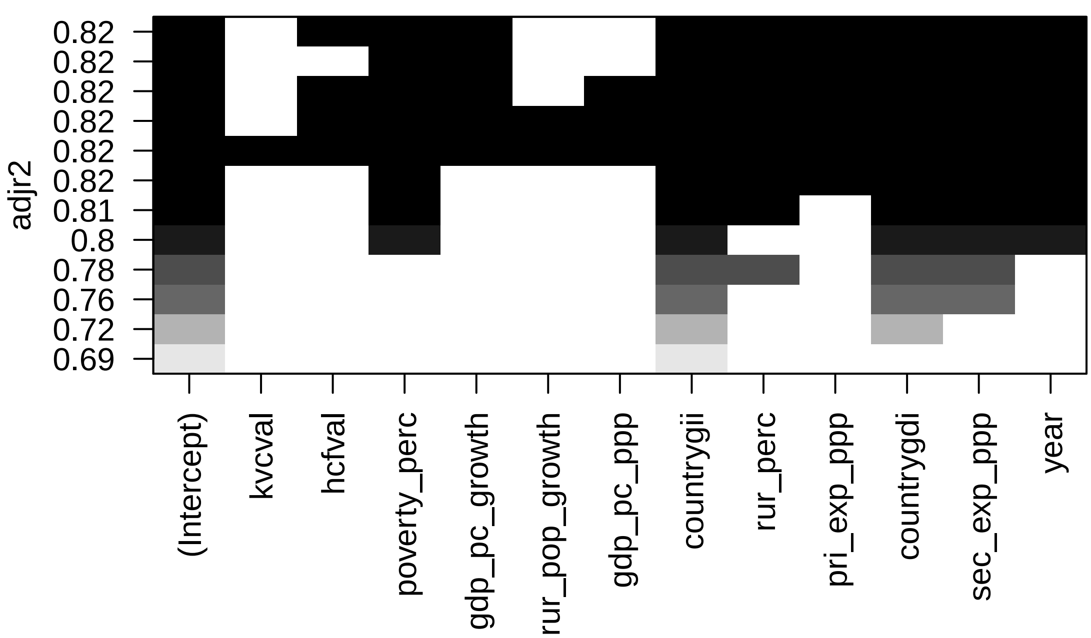
```
## Subset selection object
## Call: regsubsets.formula(medageval ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate_minor_lag, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
## gdp_pc_ppp         FALSE      FALSE
```

```
## countrygii          FALSE      FALSE
## rur_perc            FALSE      FALSE
## pri_exp_ppp         FALSE      FALSE
## countrygdi          FALSE      FALSE
## sec_exp_ppp         FALSE      FALSE
## year                FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 9
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 7
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##  -2.137537e+02   2.698320e-04  -1.612413e-01   4.939902e-02  -2.752636e-02
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##  -2.219746e-01  -1.277719e+01   5.400524e-02  -6.174538e-01   4.042810e+01
##    sec_exp_ppp           year
##   1.245828e+00   1.058392e-01
```

```r
coef(fit.full,7)
```

```
##   (Intercept)  poverty_perc    countrygii      rur_perc   pri_exp_ppp
## -246.14076593   -0.15434532  -14.26656286    0.05098417   -0.72361917
##    countrygdi   sec_exp_ppp          year
##   38.42981304    1.28630198    0.12253838
```

We will create the regrssion models with 7 variables.

kvcval poverty_perc countrygii rur_perc countrygdi sec_exp_ppp year

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(medageval ~ kvcval +  poverty_perc  +countrygii + rur_perc +countrygdi + sec_exp_ppp + year
summary(fitlm)
```

```
##
## Call:
## lm(formula = medageval ~ kvcval + poverty_perc + countrygii +
##     rur_perc + countrygdi + sec_exp_ppp + year, data = df_death_rate_minor_lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0505 -1.1594  0.2128  1.5767  4.4979
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.529e+02  3.035e+01  -8.331 6.56e-16 ***
## kvcval       -2.970e-02  1.440e-02  -2.062   0.0396 *
## poverty_perc -1.639e-01  2.147e-02  -7.636 1.02e-13 ***
## countrygii   -1.493e+01  1.669e+00  -8.949  < 2e-16 ***
## rur_perc      5.335e-02  8.517e-03   6.264 7.61e-10 ***
## countrygdi    4.393e+01  3.330e+00  13.194  < 2e-16 ***
## sec_exp_ppp   7.295e-01  8.468e-02   8.615  < 2e-16 ***
## year          1.241e-01  1.535e-02   8.086 4.02e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.146 on 544 degrees of freedom
## Multiple R-squared:  0.8142, Adjusted R-squared:  0.8118
## F-statistic: 340.6 on 7 and 544 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```
df_death_rate_minor_lag <- na.omit(df_death_rate_minor_lag)
fitModel <- (medageval ~  kvcval+  poverty_perc  +countrygii + rur_perc +countrygdi + sec_exp_ppp + yea
n = nrow(df_death_rate_minor_lag)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate_minor_lag,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate_minor_lag[test,])
}

sum((allpredictedCV-df_death_rate_minor_lag$medageval)^2)/n
```

```
## [1] 4.686317
```

****

Set 6

Next step check the multiple linear regression model and check for collinearity of factors.

```
library(car)
fitlm <- lm(leabval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +countr
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       2.860844       8.220773       3.692408       1.434765       1.191501
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      13.272494      14.793223       1.338813       4.073877       3.800399
##    sec_exp_ppp           year
##       4.649380       2.021295
```

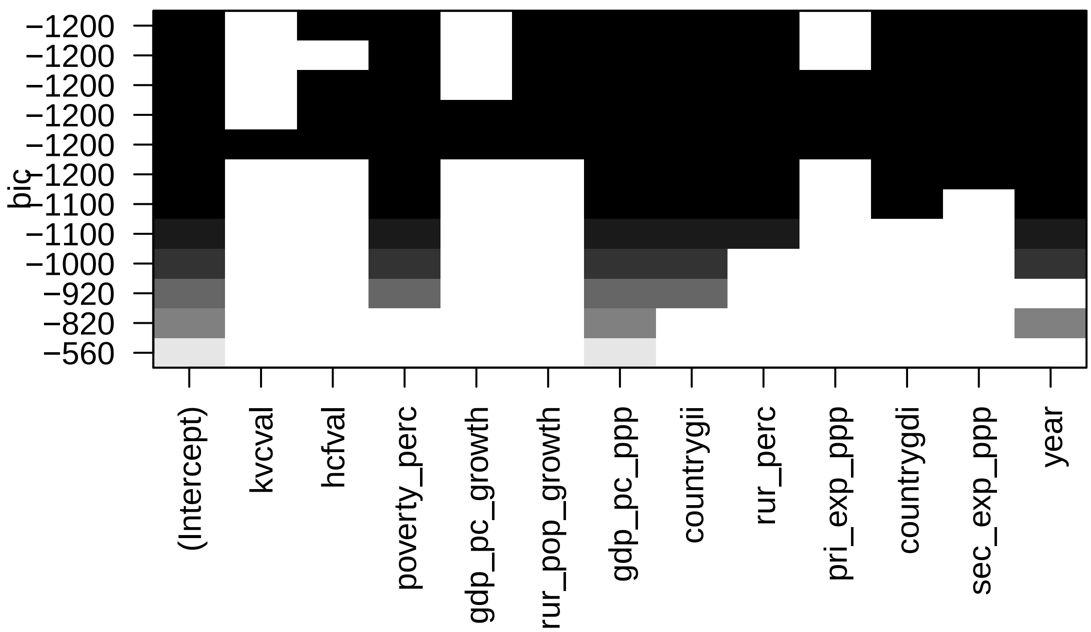
We can see that hcfval, gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```
library(leaps)
fit.full = regsubsets(leabval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_p

fit.full
```

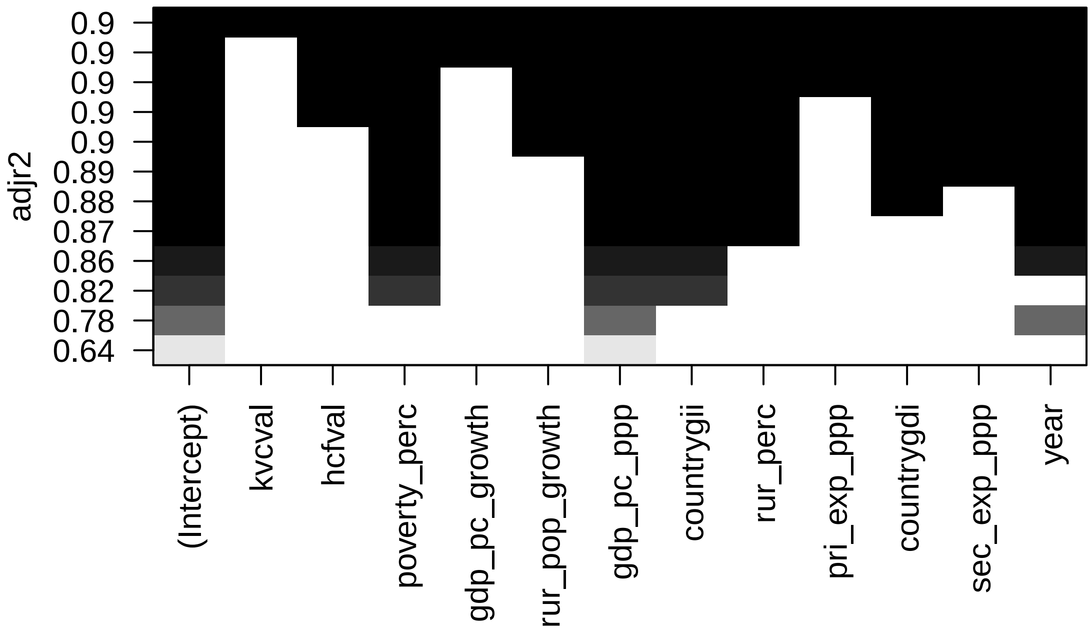
```
## Subset selection object
## Call: regsubsets.formula(leabval ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate_minor_lag, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
```

```
## gdp_pc_growth       FALSE      FALSE
## rur_pop_growth      FALSE      FALSE
## gdp_pc_ppp          FALSE      FALSE
## countrygii          FALSE      FALSE
## rur_perc            FALSE      FALSE
## pri_exp_ppp         FALSE      FALSE
## countrygdi          FALSE      FALSE
## sec_exp_ppp         FALSE      FALSE
## year                FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 12
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 9
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##  -2.044885e+02  -2.924129e-04   2.171461e-01   2.660608e-02   2.560169e-01
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##   3.454076e+00  -2.533682e+01  -5.040799e-02   1.870555e-01  -2.421964e+01
##    sec_exp_ppp           year
##  -4.233106e-01   1.371678e-01
```

```r
coef(fit.full,9)
```

```
##    (Intercept)         hcfval   poverty_perc rur_pop_growth     gdp_pc_ppp
##  -2.177546e+02  -3.496870e-04   2.248325e-01   2.455437e-01   3.646636e+00
##     countrygii       rur_perc     countrygdi    sec_exp_ppp           year
##  -2.525804e+01  -5.011869e-02  -2.443821e+01  -3.148479e-01   1.432043e-01
```

We will create the regrssion models with 8 variables.

kvcval poverty_perc countrygii rur_perc countrygdi sec_exp_ppp year

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(leabval ~ kvcval + gdp_pc_ppp +  poverty_perc + rur_pop_growth +countrygii + rur_perc +coun
summary(fitlm)
```

```
##
## Call:
## lm(formula = leabval ~ kvcval + gdp_pc_ppp + poverty_perc + rur_pop_growth +
##     countrygii + rur_perc + countrygdi + gdp_pc_ppp + year, data = df_death_rate_minor_lag)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7297 -0.8461 -0.0270  0.8539  2.7723
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.793e+02  1.841e+01  -9.741  < 2e-16 ***
## kvcval          3.279e-02  8.158e-03   4.019 6.66e-05 ***
## gdp_pc_ppp      3.240e+00  2.766e-01  11.713  < 2e-16 ***
## poverty_perc    2.005e-01  1.200e-02  16.708  < 2e-16 ***
## rur_pop_growth  2.727e-01  5.281e-02   5.164 3.39e-07 ***
## countrygii     -2.068e+01  1.440e+00 -14.368  < 2e-16 ***
## rur_perc       -5.095e-02  5.428e-03  -9.386  < 2e-16 ***
## countrygdi     -1.961e+01  2.337e+00  -8.390 4.23e-16 ***
## year            1.205e-01  8.760e-03  13.755  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.231 on 543 degrees of freedom
## Multiple R-squared:  0.8905, Adjusted R-squared:  0.8889
## F-statistic: 552.1 on 8 and 543 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```
df_death_rate_minor_lag <- na.omit(df_death_rate_minor_lag)
fitModel <- (leabval ~  kvcval + gdp_pc_ppp +  poverty_perc + rur_pop_growth +countrygii + rur_perc +co
n = nrow(df_death_rate_minor_lag)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate_minor_lag,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate_minor_lag[test,])
}

sum((allpredictedCV-df_death_rate_minor_lag$leabval)^2)/n
```

```
## [1] 1.536414
```

****

Set 7 - death_per_1000 and df_death_rate_major_lag

Next step check the multiple linear regression model and check for collinearity of factors.

```
library(car)
fitlm <- lm(death_per_1000 ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       5.261101       8.490140       3.789061       1.817697       1.298113
```

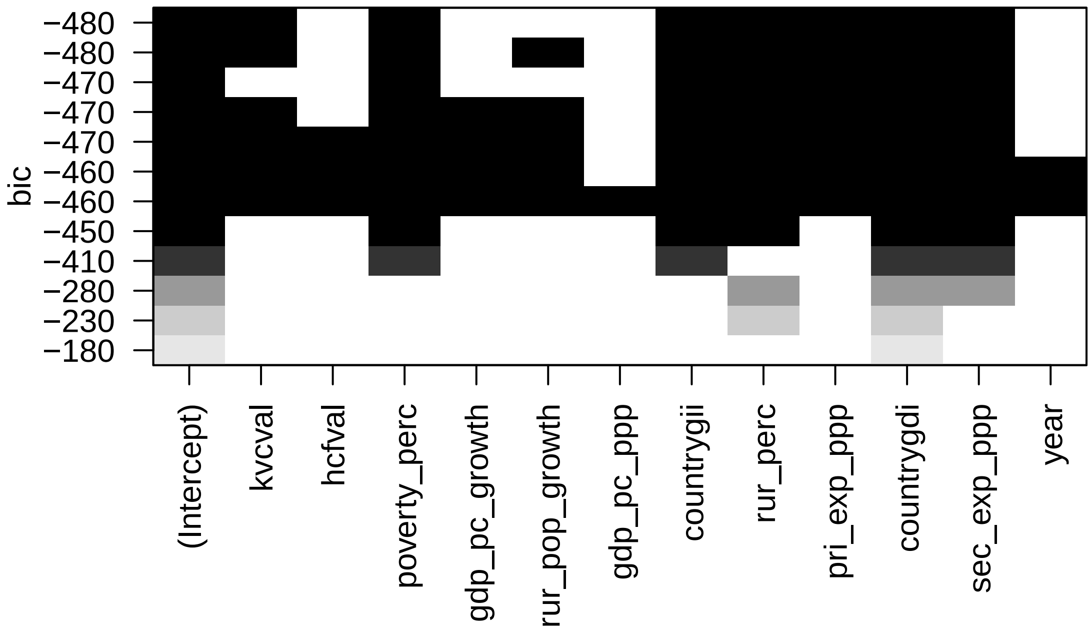
```
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      15.135567      16.593923       1.422889       3.208745       3.813500
##    sec_exp_ppp           year
##       4.677100       2.883742
```

We can see that gdp_pc_ppp, hcfval & countrygii have VIF higher than 10. We will also perform a subset reduction

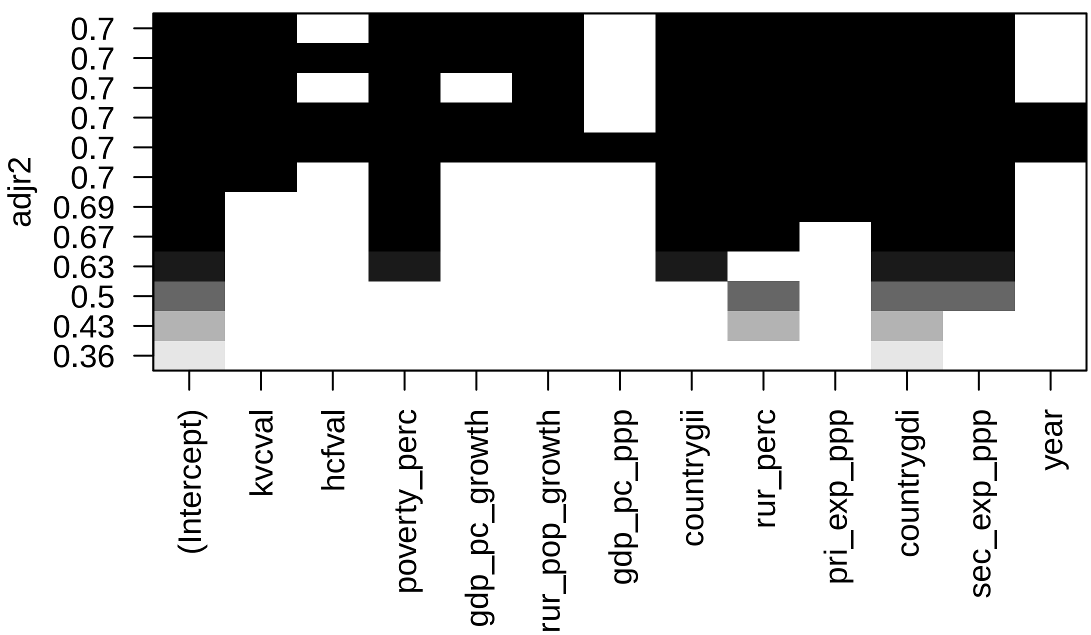
```r
library(leaps)
fit.full = regsubsets(death_per_1000 ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + g

fit.full
```

```
## Subset selection object
## Call: regsubsets.formula(death_per_1000 ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate_major_lag, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
## gdp_pc_ppp         FALSE      FALSE
## countrygii         FALSE      FALSE
## rur_perc           FALSE      FALSE
## pri_exp_ppp        FALSE      FALSE
## countrygdi         FALSE      FALSE
## sec_exp_ppp        FALSE      FALSE
## year               FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 9
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 7
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         kvcval         hcfval   poverty_perc  gdp_pc_growth
##  -2.872586e+01   3.949368e-02  -9.120865e-05  -1.858260e-01   2.495194e-02
## rur_pop_growth     countrygii       rur_perc    pri_exp_ppp     countrygdi
```

```
##   1.399685e-01   1.373052e+01   3.757818e-02  -4.510529e-01   4.568753e+01
##    sec_exp_ppp           year
##   9.936133e-01  -6.911202e-03
```

```r
coef(fit.full,6)
```

```
##  (Intercept) poverty_perc   countrygii     rur_perc  pri_exp_ppp   countrygdi
## -42.99830693  -0.18173830  13.23789859   0.03967527  -0.38841387  50.20975303
##  sec_exp_ppp
##   0.83328209
```

We will create the regrssion models with 9 variables.

kvcval , poverty_perc, rur_pop_growth, gdp_pc_ppp, countrygii,rur_perc, pri_exp_ppp ,countrygdi,sec_exp_ppp

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(death_per_1000 ~ poverty_perc +countrygii + rur_perc + countrygdi + sec_exp_ppp + year,  da
summary(fitlm)
```

```
##
## Call:
## lm(formula = death_per_1000 ~ poverty_perc + countrygii + rur_perc +
##     countrygdi + sec_exp_ppp + year, data = df_death_rate_major_lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8179 -2.3330 -0.2917  1.8964 22.1961
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  275.375793  31.124069   8.848  < 2e-16 ***
## poverty_perc   0.060116   0.006885   8.731  < 2e-16 ***
## countrygii    -7.297922   0.851993  -8.566  < 2e-16 ***
## rur_perc       0.044365   0.005244   8.460  < 2e-16 ***
## countrygdi    -6.653672   1.442372  -4.613 4.29e-06 ***
## sec_exp_ppp   -0.003833   0.047339  -0.081    0.935
## year          -0.130268   0.015506  -8.401  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.235 on 1570 degrees of freedom
##   (6452 observations deleted due to missingness)
## Multiple R-squared:  0.1542, Adjusted R-squared:  0.151
## F-statistic: 47.72 on 6 and 1570 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```r
df_death_rate_major_lag <- na.omit(df_death_rate_major_lag)
fitModel <- (death_per_1000 ~ poverty_perc +countrygii + rur_perc + countrygdi + sec_exp_ppp + year)
n = nrow(df_death_rate)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
```

```r
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate_major_lag,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate_major_lag[test,])
}

sum((allpredictedCV-df_death_rate_major_lag$death_per_1000)^2)/n
```

```
## Warning in allpredictedCV - df_death_rate_major_lag$death_per_1000: longer
## object length is not a multiple of shorter object length

## [1] NA
```

****

Set 8

Next step check the multiple linear regression model and check for collinearity of factors.

```r
library(car)
fitlm <- lm(medageval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +coun
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       5.261101       8.490140       3.789061       1.817697       1.298113
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      15.135567      16.593923       1.422889       3.208745       3.813500
##    sec_exp_ppp           year
##       4.677100       2.883742
```

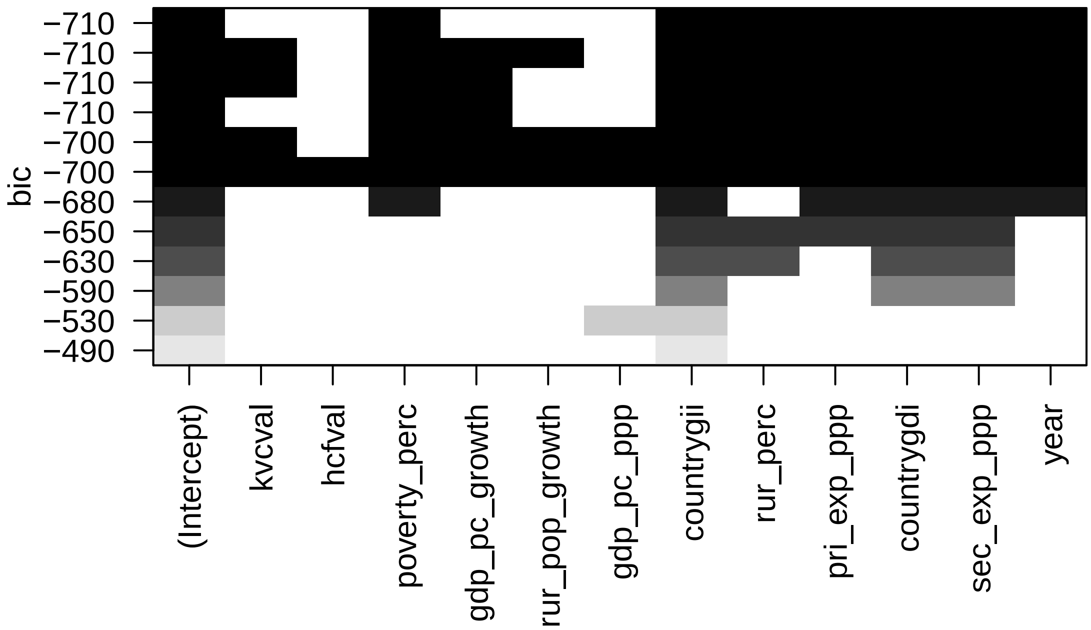
We can see that hcfval, gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```r
library(leaps)
fit.full = regsubsets(medageval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_

fit.full
```

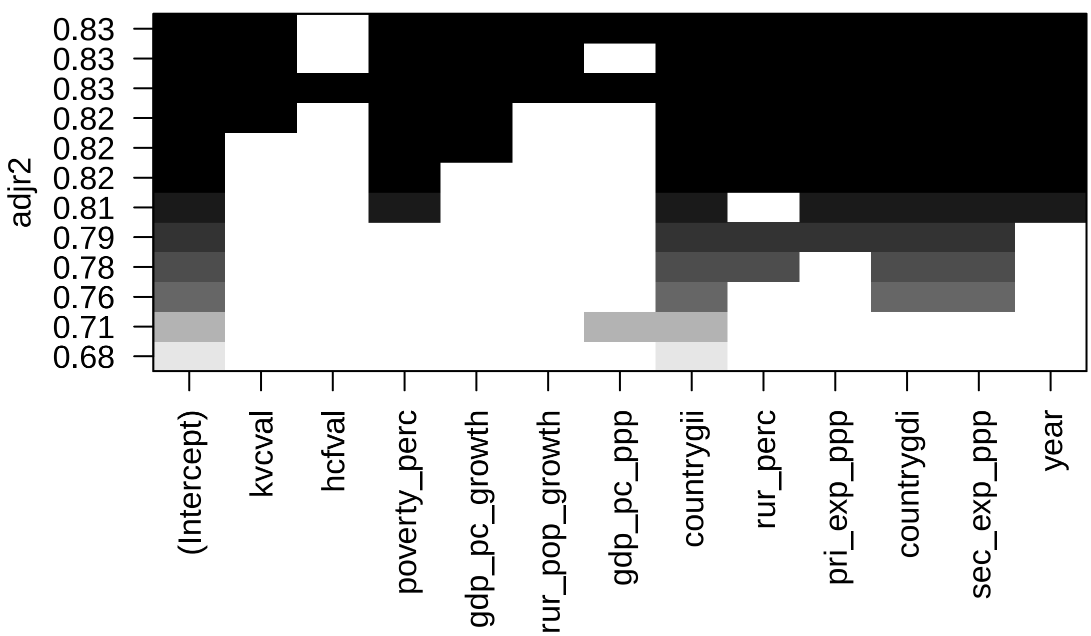
```
## Subset selection object
## Call: regsubsets.formula(medageval ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate_major_lag, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
## gdp_pc_ppp         FALSE      FALSE
## countrygii         FALSE      FALSE
## rur_perc           FALSE      FALSE
## pri_exp_ppp        FALSE      FALSE
## countrygdi         FALSE      FALSE
```

```
## sec_exp_ppp          FALSE      FALSE
## year                 FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```




```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 11
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 7
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         kvcval   poverty_perc  gdp_pc_growth rur_pop_growth 
##  -399.27908830     0.07004586    -0.17688627     0.09220187     0.27566078 
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi 
##     0.67630935    -8.45556219     0.05682699    -0.76767191    39.29269706 
##    sec_exp_ppp           year 
##     1.52103036     0.19107364
```

```r
coef(fit.full,6)
```

```
##   (Intercept) poverty_perc   countrygii  pri_exp_ppp   countrygdi  sec_exp_ppp 
## -304.0196530   -0.1549053  -13.5375536   -0.7783506   40.7222501    1.3183122 
##         year 
##    0.1511568
```

We will create the regrssion models with 9 variables.

kvcval , poverty_perc, rur_pop_growth, gdp_pc_ppp, countrygii,rur_perc, pri_exp_ppp ,countrygdi,sec_exp_ppp

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(medageval ~ poverty_perc  +  countrygii   +   rur_perc +   countrygdi +  sec_exp_ppp  +    y
summary(fitlm)
```

```
## 
## Call:
## lm(formula = medageval ~ poverty_perc + countrygii + rur_perc + 
##     countrygdi + sec_exp_ppp + year, data = df_death_rate_major_lag)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.8562 -1.0814  0.0924  1.5051  4.4071 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -2.997e+02  4.118e+01  -7.278 1.63e-12 ***
## poverty_perc -1.438e-01  2.312e-02  -6.218 1.19e-09 ***
## countrygii   -1.529e+01  1.748e+00  -8.745  < 2e-16 ***
## rur_perc      5.578e-02  9.376e-03   5.950 5.58e-09 ***
## countrygdi    4.415e+01  3.644e+00  12.115  < 2e-16 ***
## sec_exp_ppp   8.419e-01  8.011e-02  10.510  < 2e-16 ***
## year          1.455e-01  2.046e-02   7.114 4.73e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.113 on 430 degrees of freedom
## Multiple R-squared:  0.8072, Adjusted R-squared:  0.8045 
## F-statistic:   300 on 6 and 430 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```r
df_death_rate_major_lag <- na.omit(df_death_rate_major_lag)
fitModel <- (medageval ~ poverty_perc  +  countrygii   +   rur_perc +   countrygdi +  sec_exp_ppp  +
n = nrow(df_death_rate_major_lag)
k = 5
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate_major_lag,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate_major_lag[test,])
}

sum((allpredictedCV-df_death_rate_major_lag$medageval)^2)/n
```

```
## [1] 4.564385
```

****

Set 9

Next step check the multiple linear regression model and check for collinearity of factors.

```r
library(car)
fitlm <- lm(leabval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_ppp +countr
vif(fitlm)
```

```
##         kvcval         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##       5.261101       8.490140       3.789061       1.817697       1.298113
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##      15.135567      16.593923       1.422889       3.208745       3.813500
##    sec_exp_ppp           year
##       4.677100       2.883742
```

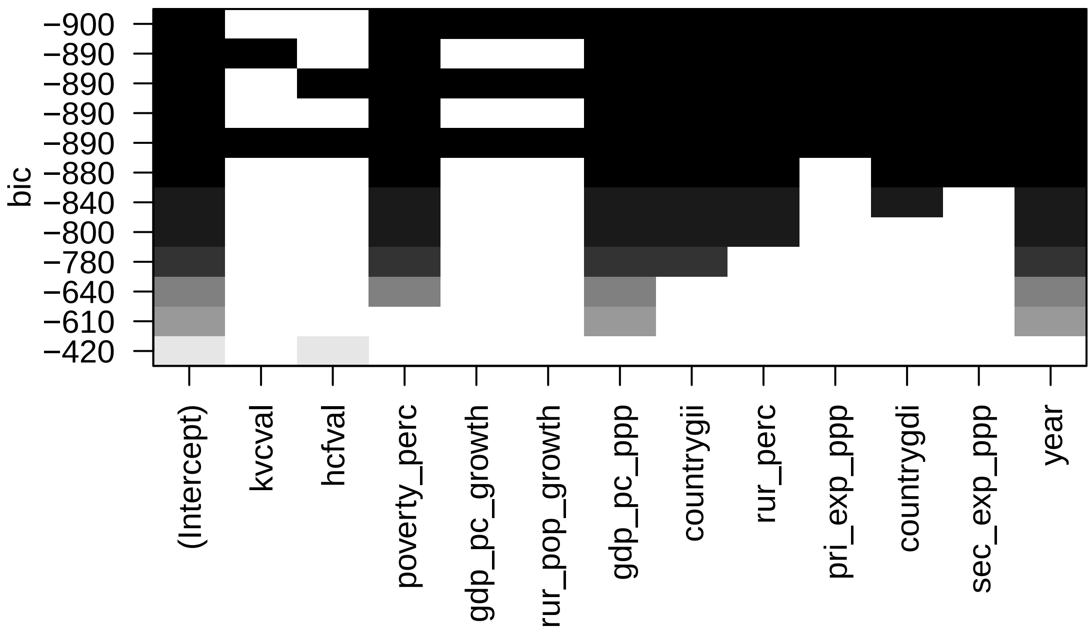
We can see that gdp__pc__ppp & countrygii have VIF higher than 10. We will also perform a subset reduction

```r
library(leaps)
fit.full = regsubsets(leabval ~ kvcval +hcfval+ poverty_perc + gdp_pc_growth+ rur_pop_growth + gdp_pc_p

fit.full
```

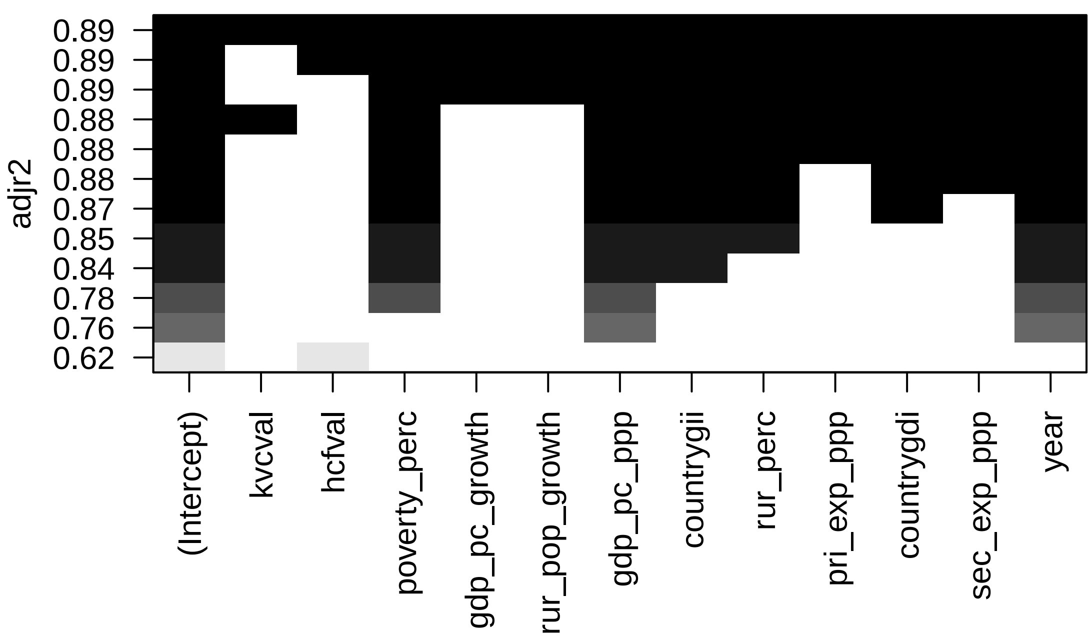
```
## Subset selection object
## Call: regsubsets.formula(leabval ~ kvcval + hcfval + poverty_perc +
##     gdp_pc_growth + rur_pop_growth + gdp_pc_ppp + countrygii +
##     rur_perc + pri_exp_ppp + countrygdi + sec_exp_ppp + year,
##     data = df_death_rate_major_lag, nvmax = 12)
## 12 Variables  (and intercept)
##                Forced in Forced out
## kvcval             FALSE      FALSE
## hcfval             FALSE      FALSE
## poverty_perc       FALSE      FALSE
## gdp_pc_growth      FALSE      FALSE
## rur_pop_growth     FALSE      FALSE
```

```
## gdp_pc_ppp          FALSE      FALSE
## countrygii          FALSE      FALSE
## rur_perc            FALSE      FALSE
## pri_exp_ppp         FALSE      FALSE
## countrygdi          FALSE      FALSE
## sec_exp_ppp         FALSE      FALSE
## year                FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
```

```r
plot(fit.full)
```




```r
plot(fit.full, scale = "adjr2")
```



```r
fit.full.summary = summary(fit.full)
which.max(fit.full.summary$adjr2)
```

```
## [1] 12
```

```r
which.min(fit.full.summary$bic)
```

```
## [1] 10
```

```r
coef(fit.full,11)
```

```
##    (Intercept)         hcfval   poverty_perc  gdp_pc_growth rur_pop_growth
##  -2.536468e+02  -2.532549e-04   1.997872e-01   4.966260e-02   2.183300e-01
##     gdp_pc_ppp     countrygii       rur_perc    pri_exp_ppp     countrygdi
##   2.890911e+00  -2.562643e+01  -4.881597e-02   3.037669e-01  -2.662832e+01
##    sec_exp_ppp           year
##  -4.627028e-01   1.656101e-01
```

```r
coef(fit.full,10)
```

```
##    (Intercept)   poverty_perc  gdp_pc_growth rur_pop_growth     gdp_pc_ppp
##  -222.46549835     0.19937932     0.04714109     0.21101366     2.46688322
##     countrygii       rur_perc    pri_exp_ppp     countrygdi    sec_exp_ppp
##   -25.43582625    -0.04825226     0.33541548   -26.39694189    -0.47246632
##           year
##     0.15175322
```

We will create the regrssion models with 9 variables.

kvcval , poverty_perc, rur_pop_growth, gdp_pc_ppp, countrygii,rur_perc, pri_exp_ppp ,countrygdi,sec_exp_ppp

Through the variable reduction gdp_pc_ppp remained instead of gdp_pc_growth.

```r
fitlm <- lm(leabval ~ poverty_perc + gdp_pc_growth + rur_pop_growth + gdp_pc_ppp +countrygii + rur_perc
summary(fitlm)
```

```
##
## Call:
## lm(formula = leabval ~ poverty_perc + gdp_pc_growth + rur_pop_growth +
##     gdp_pc_ppp + countrygii + rur_perc + pri_exp_ppp + countrygdi +
##     sec_exp_ppp + year, data = df_death_rate_major_lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5739 -0.8160  0.0339  0.8496  2.7327
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.225e+02  2.617e+01  -8.499 3.24e-16 ***
## poverty_perc    1.994e-01  1.341e-02  14.868  < 2e-16 ***
## gdp_pc_growth   4.714e-02  1.604e-02   2.938 0.003481 **
## rur_pop_growth  2.110e-01  6.226e-02   3.389 0.000766 ***
## gdp_pc_ppp      2.467e+00  2.825e-01   8.733  < 2e-16 ***
## countrygii     -2.544e+01  1.470e+00 -17.301  < 2e-16 ***
## rur_perc       -4.825e-02  5.774e-03  -8.356 9.24e-16 ***
## pri_exp_ppp     3.354e-01  7.326e-02   4.578 6.16e-06 ***
## countrygdi     -2.640e+01  2.793e+00  -9.450  < 2e-16 ***
```

```
## sec_exp_ppp    -4.725e-01  5.789e-02  -8.161 3.78e-15 ***
## year            1.518e-01  1.194e-02  12.707  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.147 on 426 degrees of freedom
## Multiple R-squared:  0.8894, Adjusted R-squared:  0.8868
## F-statistic: 342.5 on 10 and 426 DF,  p-value: < 2.2e-16
```

Doing 10 fold cross validation

```
df_death_rate_major_lag <- na.omit(df_death_rate_major_lag)
fitModel <- (leabval ~ poverty_perc + gdp_pc_growth + rur_pop_growth + gdp_pc_ppp +countrygii + rur_perc
n = nrow(df_death_rate_major_lag)
k = 10
groups = c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))  #produces list of group labels
cvgroups = sample(groups,n)
set.seed(20)

  #prediction via cross-validation
allpredictedCV = rep(0,n)
for (i in 1:k)  {
  test = (cvgroups == i)
  lmfitCV = lm(formula = fitModel,data=df_death_rate_major_lag,subset=!test)
  allpredictedCV[test] = predict.lm(lmfitCV,df_death_rate_major_lag[test,])
}

sum((allpredictedCV-df_death_rate_major_lag$leabval)^2)/n
```

```
## [1] 1.366406
```