252: Life Expectancy Analytics

Your submissions:

- Report_252.pdf
- R Codes_252.txt
- R Outputs_252.pdf

Notes

- The deadline is in the noon, not midnight
- No extension to the deadline
- Follow the given template
- Each team can only submit one copy by a single member, just list all of your members in the report

First Name	Last Name	Email (hawk.iit.edu)	Student ID
Minguk	Kim	mkim105@hawk.iit.edu	A20437179
Boyun	Jang	bjang7@hawk.iit.edu	A20437298

Table of Contents

1. Introduction

Life expectancy is one of the factors that measure human development index (HDI) in each country besides human education level and living standard. It is used to describe the quality of life in a particular area. The difference in life expectancy is also cited to demonstrate the need for medical care and the improvement of social support.

In order to prove the necessity of medical and social support for each country, we are going to analyze the factors of life expectancy such as the national mortality rate, economic factors, social factors and other health-related factors.

We will present what kind of improvements are needed to improve the life expectancy of the population by predicting the vulnerable diseases and environments that affect the mortality rate in a specific area in order to improve the mortality rate.

For the result, our main goal is find the factors which are needed to improve the life expectancy.

2. Data

Data sets related to life expectancy and health factors in 193 countries were collected from the same WHO data store website. Economic data was collected on the UN website. Only representative elements were selected from all categories of health-related factors. The data set consists of 22 columns and 2938 rows, which means 20 prediction variables.

Variable	Description	Data type
Country	Country	Nomial
Year	Year	Discrete
Status	Developed or Developing status	Binary
Life expectancy	Life Expectancy in age	Continous
Adult Mortality	Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)	Discrete
Infant deaths	Number of Infant Deaths per 1000 population	Discrete
Alcohol	Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)	Continuous
percentage expenditure	Expenditure on health as a percentage of Gross Domestic Product per capita(%)	Continuous
Hepatitis B	Hepatitis B (HepB) immunization coverage among 1-year-olds (%)	Discrete
Measles	Measles - number of reported cases per 1000 population	Discrete
BMI	Average Body Mass Index of entire population	Continuous
under-five deaths	Number of under-five deaths per 1000 population	Discrete
Polio	Polio (Pol3) immunization coverage among 1-year-olds (%)	Discrete
Total expenditure	General government expenditure on health as a percentage of total government expenditure (%)	Continuous
Diphtheria	Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)	Discrete
HIV/AIDS	Deaths per 1 000 live births HIV/AIDS (0-4 years)	Continuous
GDP	Gross Domestic Product per capita (in USD)	Continuous
Population	Population of the country	Continuous
thinness 1-19 years	Prevalence of thinness among children and adolescents for Age 10 to 19 (%)	Continuous
thinness 5-9 years	Prevalence of thinness among children for Age 5 to 9(%)	Continuous
Income composition of resources	Human Development Index in terms of income composition of resources (index ranging from 0 to 1)	Continuous
Schooling	Number of years of Schooling(years)	Continuous

Source: https://www.kaggle.com/kumarajarshi/life-expectancy-who

3. Problems to be solved

The main goal of this project is to predict if the various data elements we collect will affect the average life expectancy.

The following objectives are based on these key objectives.

1. Predict what vulnerable diseases and environments will affect the mortality rate in a given area in order to improve anticipated life expectancy.

2. Analyze the data and discover patterns that can increase life expectancy. For example, we can analyze if increasing medical expenditure or increasing the compulsory schooling period can affect positively the average life expectancy.

4. Solutions

The main goal of this project is to predict factors that affect life expectancy and life expectancy. We used multiple linear regressions as a predictive model.

When we had built a model, we found a polynomial and interaction term during the process. Therefore, for the accurate model, we built 3 models by using pure (row) variables, polynomial model and polynomial + interaction term. Finally, we compared among these three models for finding best model.

- Dependent and Independent variables
  - **Dependent** :
    Life Expectancy
  - **Independent** :
    Country
    Year
    Status
    Adult Mortality
    Infant deaths
    Alcohol
    percentage expenditure
    Hepatitis B
    Measles
    BMI
    under-five deaths
    Polio
    Total expenditure
    Diphtheria
    HIV/AIDS
    GDP
    Population
    thinness 1-19 years
    thinness 5-9 years
    Income composition of resources
    Schooling

5. Experiments and Results

Data size is relatively small (2939), so we use N-Fold evaluation.

To confirm the qualification of the created model, F-test and residual analysis will be performed to check the accuracy. We will analyze the data as the most appropriate model for problem solving through confidence interval and multiple criteria such as p-value test, adjust R-squared test and RMSE verification.

5.1. Methods and Process

Before start building linear models, we performed preprocessing the data first. Understand the data set first, found the missing values and fill these values as the average of each country. Since we will analyze country as continent, separate the counties as 5 continents (Asia, Africa, America, Oceania, and Europe). We converted nominal variables to discrete variables. Country and status columns are nominal; convert these columns to dummy variables.

```
for (i in (1:214))
  for (j in (1:14))
  {
    newdata[newdata$Country==l[i],][[l2[j]]] = ifelse(is.na(newdata[newdata$Country==l[i],][[l2[j]]]),
                                                      ave(newdata[newdata$Country==l[i],][[l2[j]]], FUN = function(x) mean(x, na.rm=T)),
                                                      newdata[newdata$Country==l[i],][[l2[j]]])

    table(is.na(newdata[[l2[j]]]))
  }

for (i in (1:214))
  for (j in (1:14))
  {
    newdata[newdata$Country==l[i],][[l2[j]]] = ifelse(is.na(newdata[newdata$Country==l[i],][[l2[j]]]),
                                                      ave(newdata[[l2[j]]], FUN = function(x) mean(x, na.rm=T)),
                                                      newdata[newdata$Country==l[i],][[l2[j]]])

  }
table(is.na(newdata))
```

```
> table(is.na(newdata))

FALSE
66424
```

```
library(countrycode)
b <- data.frame(country = newdata$Country)
newdata$Country <- countrycode(sourcevar = b[,"country"],
                               origin = "country.name",
                               destination = "continent")

# convert nomial(Country) variables to dummy variables.
library(dummies)
newdata = dummy.data.frame(newdata,names=c("Country"))

# Binary
library(plyr)
newdata$Status<- revalue(newdata$Status, c("Developing"="0"))
newdata$Status<- revalue(newdata$Status, c("Developed"="1"))
newdata$Status
```

First of all, we examined linear relationship between X and Y variables via checking the correlation by transforming Y. Also, confirmed the transformed X variables which have a small correlation with Y.

```
y = lfey
y2 = y*y
y3 = y2*y
invy = 1/y
sqrty = sqrt(y)
logy = log(y)
cor(columns2)
```

```
               y          y2          y3        invy      sqrty        logy
y      1.0000000  0.9964692  0.9865385 -0.9840388  0.9990572  0.9961528
y2     0.9964692  1.0000000  0.9967559 -0.9659635  0.9919044  0.9853495
y3     0.9865385  0.9967559  1.0000000 -0.9428048  0.9786094  0.9687096
invy  -0.9840388 -0.9659635 -0.9428048  1.0000000 -0.9907889 -0.9958038
sqrty  0.9990572  0.9919044  0.9786094 -0.9907889  1.0000000  0.9990156
logy   0.9961528  0.9853495  0.9687096 -0.9958038  0.9990156  1.0000000
yr     0.1713633  0.1687933  0.1665783 -0.1769372  0.1727473  0.1741633
atmy  -0.6900096 -0.6820656 -0.6696242  0.6894734 -0.6920587 -0.6927061
itdth -0.1437800 -0.1464632 -0.1475007  0.1331904 -0.1417910 -0.1393624
achl   0.4082513  0.4303905  0.4483107 -0.3527560  0.3956853  0.3821962
perex  0.4071675  0.4356213  0.4614520 -0.3448334  0.3921359  0.3766769
hb     0.2851625  0.2769798  0.2659277 -0.2905172  0.2879802  0.2898522
msls  -0.1557258 -0.1529208 -0.1492574  0.1576385 -0.1567230 -0.1574004
bmi    0.5653220  0.5654775  0.5602744 -0.5463651  0.5630135  0.5591196
ufdth -0.1713977 -0.1721773 -0.1711809  0.1638123 -0.1702806 -0.1686525
plio   0.4632779  0.4586167  0.4499948 -0.4579351  0.4638943  0.4632561
totalex 0.2178426 0.2357236  0.2508319 -0.1748096  0.2079293  0.1974189
dpria  0.4769532  0.4707225  0.4606456 -0.4748065  0.4783775  0.4785492
hiv   -0.5590283 -0.5261011 -0.4924427  0.6170289 -0.5748154 -0.5898938
gdp    0.4390118  0.4645927  0.4869996 -0.3803028  0.4251706  0.4107197
th119 -0.4775712 -0.4895396 -0.4966647  0.4384313 -0.4696942 -0.4605388
th59  -0.4720338 -0.4853335 -0.4937789  0.4304966 -0.4635094 -0.4537319
iccr   0.8525520  0.8602397  0.8607671 -0.8135706  0.8458296  0.8371124
sch    0.7514156  0.7634491  0.7688813 -0.7061978  0.7427896  0.7323815
```

```
# transforming x with small correlation variables with y
columns3 = cbind(y,yr,itdth,hb,msls,ufdth,totalex)
cor(columns3)

#comparison yr tranasformation
t1 = yr
t2 = yr*yr
t3 = log(yr)
t4 = 1/yr
t5 = sqrt(yr)
tcol = cbind(y,t1,t2,t3,t4,t5)
cor(tcol)
```

Transformed 'Hepatitis B', 'under-five deaths' >0.3
So, doing transformation for these variables.

hb-> hb^2
ufdth -> log(ufdth)
ignore yr, msls, totalex, itdth

Second, we built models by feature selection and tested the VIF to model. If the VIF test value is larger than 4, remove the value for building models. We performed feature selection methods through backward/forward/both stepwise function and best subset function.

```
#build basic model(Just only one x variables)
basicfull = glm(lfey~cAf + cAm+ cAs+cOc+
                sts+
                achl+perex+bmi+gdp+th119+th59+iccr+sch+hiv+
                atmy+logufdth+hb2+plio+dpria)
```

```
vif(basicfull)
     cAf      cAm      cAs      cOc      sts     achl    perex      bmi      gdp    th119     th59     iccr      sch      hiv
5.036362 2.384928 3.921990 1.679143 2.417647 2.586669 6.058090 1.953509 6.202818 8.747050 8.918560 11.108005 7.700737 1.583615
    atmy logufdth      hb2     plio    dpria
1.826753 1.235896 1.782178 2.016860 2.313377
```

```
#iccr,th59,perex,cAf,cEu are removed.
basicfull = glm(lfey~ cAm+ cAs+cOc+
                    sts+
                    achl+bmi+gdp+th119+sch+hiv+
                    atmy+logufdth+hb2+plio+dpria)
```

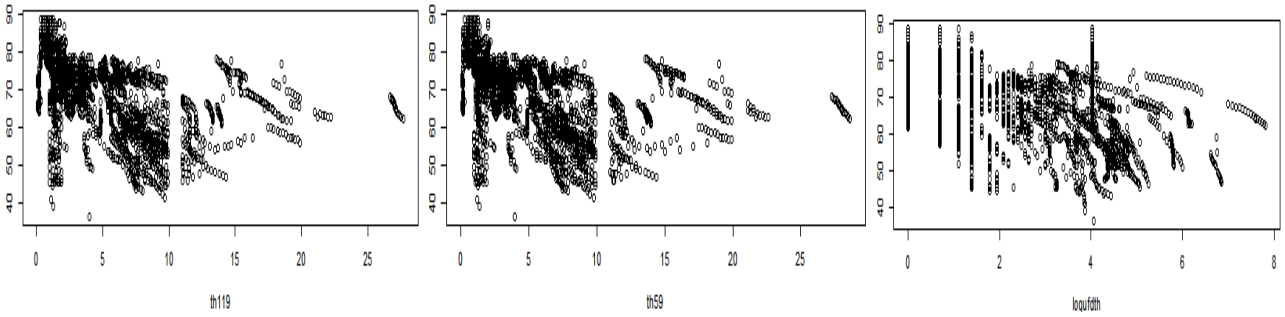
```
> vif(basicfull)
     cAm      cAs      cOc      sts     achl      bmi      gdp    th119      sch      hiv     atmy logufdth      hb2     plio    dpria
1.487572 1.633895 1.271044 2.225988 2.261671 1.771586 1.414127 1.916015 2.492591 1.461849 1.762092 1.224861 1.739596 2.005159 2.299622
```

```
#build full model CASE1
full_1 = glm(lfey~ cAm+ cAs+cOc+ #iccr,th59,perex,cAf,cEu are removed.
                 sts+
                 achl+bmi+gdp+th119+sch+hiv+
                 atmy+logufdth+hb2+plio+dpria)

#build base model CASE1
base_1 = glm(lfey~achl,data=newdata)

# model 1  Both direction by stepwise()
m1_1 = step(base_1, scope=list(upper=full_1, lower=~1), direction="both", trace=T)
summary(m1_1)  # aic 15990

# m2 Forward Selection by step()
m1_2 = step(base_1, scope=list(upper=full_1, lower=~1), direction="forward", trace=T)
summary(m1_2) # AIC: 16000

# m3 Backward Elimination by step() based on AIC
m1_3 = step(full_1, direction = "backward", trace=TRUE)
summary(m1_3) # AIC: 15990
```

```
# m1_4 by best subset

library(leaps)
regsubsets.out <-
  regsubsets(lfey~cAm+cAs+cOc+
               sts+
               achl+bmi+gdp+th119+sch+hiv+
               atmy+logufdth+hb2+plio+dpria,
             data = newdata,
             nbest = 1,       # 1 best model for each number of predictors
             nvmax = NULL,    # NULL for no limit on number of variables
             force.in = NULL, force.out = NULL,
             method = "exhaustive")
summary.out <- summary(regsubsets.out)
as.data.frame(summary.out$outmat)
res.legend <-
  subsets(regsubsets.out, statistic="adjr2", legend = FALSE, min.size = 5, main = "Adjusted R^2")
res.legend
which.max(summary.out$adjr2)  #13
summary.out$which[13,]
m1_4 = glm(formula = lfey~cAm+cAs+cOc+
           sts+
           achl+bmi+gdp+th119+sch+hiv+
           atmy+logufdth+hb2+plio+dpria,data=newdata)
```

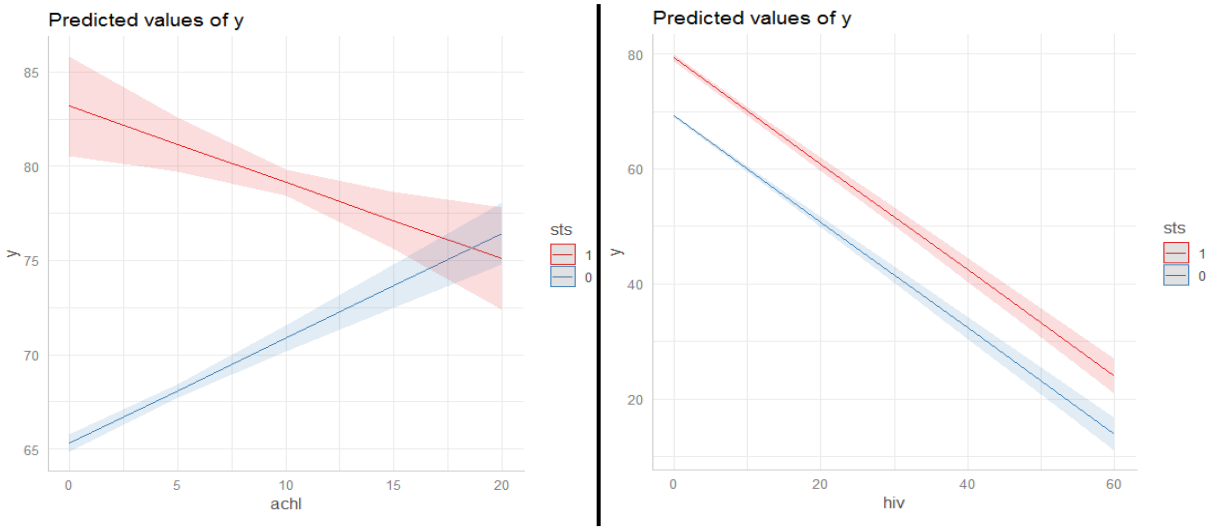
```
#Conclusion:
#m1_1, m1_3 has less AIC value.
# glm(formula = lfey ~ cAm + cAs + cOc + sts + bmi + gdp + th119 + sch + hiv + atmy + logufdth + plio + dpria) is best model
```

Third, we checked the polynomial and interaction term through each variable of the graph.



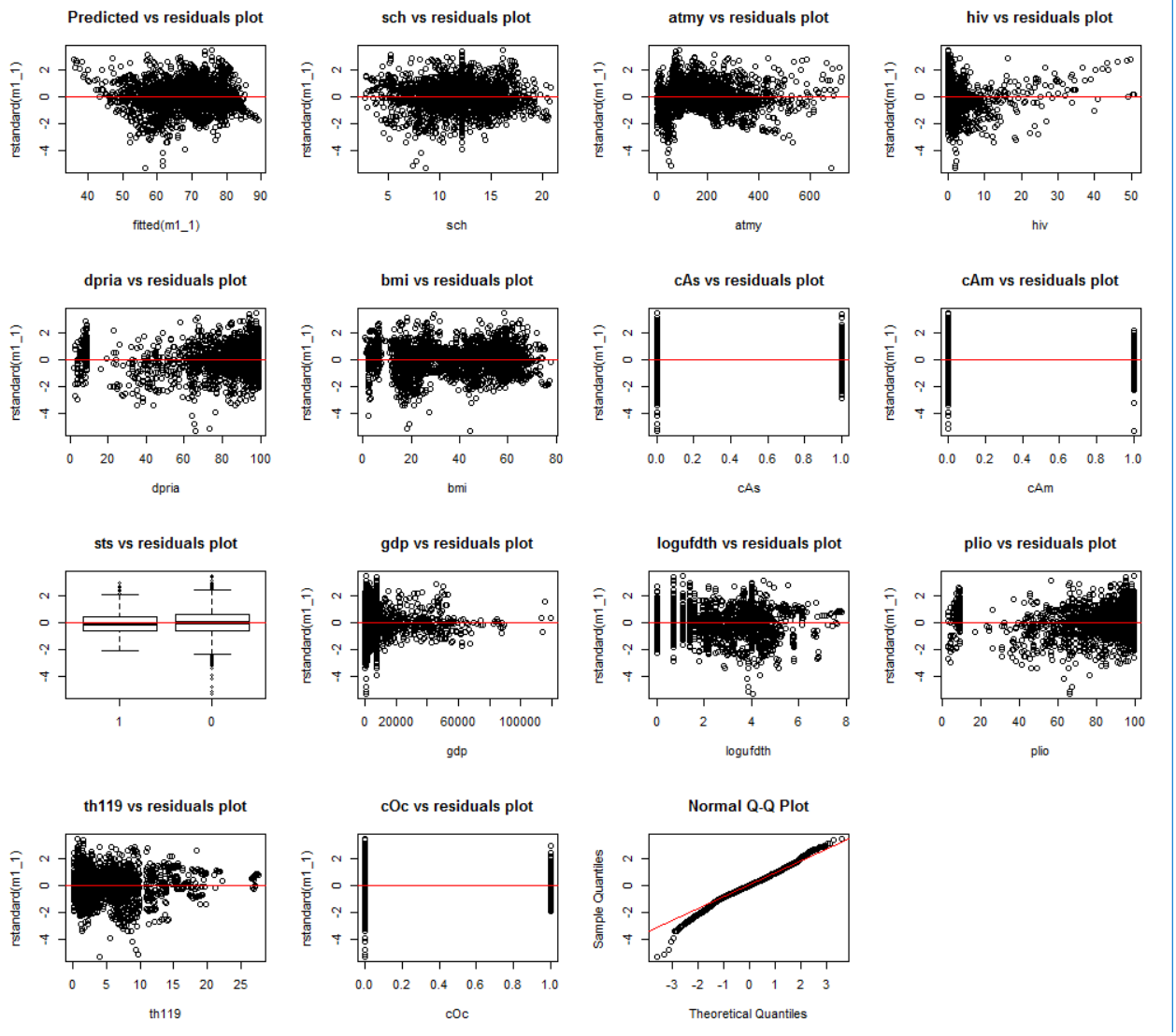
It is doubtful that these variables are polynomial.

```
library(sjPlot)
library(sjmisc)
library(ggplot2)
library(snakecase)

theme_set(theme_sjplot())

# fit model with interaction

tf1 <- glm(y ~ achl * sts, data = newdata)
plot_model(tf1, type = "int", terms = c("achl", "sts"))
```



If the slopes are different, it means there is interaction term between two x variables.

For the fourth step, since data size is small we'll use N-fold evaluation, so we don't need to check F-test. We verified model is qualified or not. Constant variance, linearity relationship and distribution of residuals have been checked for validation.



Finally, after performing residual analysis, we evaluated model by N-fold cross validations.

Also for the accurate model, found the influential points and removed it.

```
# Step 5: Model Evaluations by N-folds Cross validations
library(boot)
m1_1 = glm(lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
             sts + gdp + log(newdata$under.five.deaths) + plio + th119 + cOc, data = newdata) #에러나서 재설정

m2_1 = glm(lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
             sts + gdp + log(newdata$under.five.deaths) + plio + cOc, data = newdata)

m3_1 = glm(lfey ~ achl + sch + atmy + hiv + dpria + bmi +
             cAs + cAm + gdp + log(newdata$under.five.deaths) + plio + th119 + cOc + hiv:cAm +
             cAm:gdp + cAs:gdp + gdp:cOc + th119:cOc, data = newdata)

mse1 = cv.glm(newdata, m1_1, K=10)$delta
mse2 = cv.glm(newdata, m2_1, K=10)$delta
mse3 = cv.glm(newdata, m3_1, K=10)$delta
```

```
> errs = cbind(mse1, mse2, mse3)
> errs
        mse1     mse2     mse3
[1,] 167.597 166.6777 168.6911
[2,] 167.597 166.6777 168.6911
```

```
#conclusion: errors are similar, so accept m1_1 as best model
```

```
> cooksd <- cooks.distance(m1_1)
> influential <- as.numeric(names(cooksd)[(cooksd > (4/2888))])
> influential
  [1]   49   63   64  113  114  115  116  117  118  120  122  127  133  196  310  317  326  327  334  351  352  410  432  433  435  437  438
 [28]  439  440  445  455  456  483  497  514  517  518  519  520  531  532  533  534  535  536  540  624  625  629  633  666  715  720  721
 [55]  744  788  840  842  848  884  888  889  900  901  903  919  920  922  923  924  961  962  982  983 1012 1025 1043 1087 1089 1090 1100
 [82] 1112 1156 1222 1223 1233 1364 1365 1366 1367 1368 1369 1370 1386 1455 1462 1463 1470 1479 1482 1510 1519 1541 1550 1557 1560 1608 1639
[109] 1698 1701 1718 1747 1751 1856 1880 1881 1882 1883 1884 1885 1893 1937 1961 2033 2034 2066 2067 2068 2069 2070 2075 2084 2140 2176 2183
[136] 2185 2186 2189 2190 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2288 2289 2299 2322 2343 2361 2362 2363 2364 2365 2366
[163] 2380 2381 2382 2383 2384 2410 2473 2475 2479 2480 2481 2490 2516 2517 2630 2703 2708 2741 2742 2743 2760 2768 2769 2822 2825 2826 2827
[190] 2832 2833 2890 2895 2896 2907 2908 2910 2911 2912 2913 2914
```

```
newdata2 <- newdata[-influential, ]
```

```
m1_1_new = glm(lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
                 sts + gdp + log(newdata2$under.five.deaths) + plio + th119 + cOc, data = newdata2)
mse1_new = cv.glm(newdata2, m1_1_new, K=10)$delta
errs2 = cbind(mse1, mse1_new)
errs2
```

```
> errs2 = cbind(mse1, mse1_new)
> errs2
        mse1 mse1_new
[1,] 167.0504 157.8734
[2,] 167.0504 157.8734
>
```

5.2. Evaluations and Results

CASE 1 (Linear Regression)

```
#build full model CASE1
full_1 = glm(lfey~ cAm+ cAs+cOc+ #iccr,th59,perex,cAf,cEu are removed.
                   sts+
                   achl+bmi+gdp+th119+sch+hiv+
                   atmy+logufdth+hb2+plio+dpria)

#build base model CASE1
base_1 = glm(lfey~achl,data=newdata)

# model 1  Both direction by stepwise()
m1_1 = step(base_1, scope=list(upper=full_1, lower=~1), direction="both", trace=T)
summary(m1_1)  # aic 15990

# m2 Forward Selection by step()
m1_2 = step(base_1, scope=list(upper=full_1, lower=~1), direction="forward", trace=T)
summary(m1_2) # AIC: 16000

# m3 Backward Elimination by step() based on AIC
m1_3 = step(full_1, direction = "backward", trace=TRUE)
summary(m1_3) # AIC: 15990

# m1_4 by best subset

#install.packages("leaps")
library(leaps)
regsubsets.out <-
  regsubsets(lfey~cAm+cAs+cOc+
               sts+
               achl+bmi+gdp+th119+sch+hiv+
               atmy+logufdth+hb2+plio+dpria,
             data = newdata,
             nbest = 1,       # 1 best model for each number of predictors
             nvmax = NULL,    # NULL for no limit on number of variables
             force.in = NULL, force.out = NULL,
             method = "exhaustive")
summary.out <- summary(regsubsets.out)
as.data.frame(summary.out$outmat)
res.legend <-
  subsets(regsubsets.out, statistic="adjr2", legend = FALSE, min.size = 5, main = "Adjusted R^2")
res.legend
which.max(summary.out$adjr2)  #13
summary.out$which[13,]
m1_4 = glm(formula = lfey~cAm+cAs+cOc+
           sts+
           achl+bmi+gdp+th119+sch+hiv+
           atmy+logufdth+hb2+plio+dpria,data=newdata)
summary(m1_4) # AIC - 16000


#Conclusion:
#m1_1, m1_3 are the best models
# glm(formula = lfey ~ cAm + cAs + cOc + sts + bmi + gdp + th119 + sch + hiv + atmy + logufdth + plio + dpria)
```

CASE2 (Polynomial Regression)

```
full_2 = glm(lfey~cAm+cAs+cOc+
               sts+
               achl+bmi+gdp+th1193+
               sch+hiv+atmy+logufdth+hb2+plio+dpria,
             data=newdata)
vif(full_2) #Multicollinearity 해결

#build base model CASE1
base_2 = glm(lfey~achl,data=newdata)

# model 1  Both direction by stepwise()
m2_1 = step(base_2, scope=list(upper=full_2, lower=~1), direction="both", trace=T)
summary(m2_1)  # aic 16010

# m2 Forward Selection by step()
m2_2 = step(base_2, scope=list(upper=full_2, lower=~1), direction="forward", trace=T)
summary(m2_2) # AIC: 16010

# m3 Backward Elimination by step() based on AIC
m2_3 = step(full_2, direction = "backward", trace=TRUE)
summary(m2_3) # AIC: 16010

# m2_4 by best subset
#install.packages("leaps")
library(leaps)
regsubsets.out <-
  regsubsets(lfey~cAm+cAs+cOc+
               sts+
               achl+bmi+gdp+th1193+
               sch+hiv+atmy+logufdth+hb2+plio+dpria,
             data = newdata,
             nbest = 1,       # 1 best model for each number of predictors
             nvmax = NULL,    # NULL for no limit on number of variables
             force.in = NULL, force.out = NULL,
             method = "exhaustive")
summary.out <- summary(regsubsets.out)
as.data.frame(summary.out$outmat)
res.legend <-
  subsets(regsubsets.out, statistic="adjr2", legend = FALSE, min.size = 5, main = "Adjusted R^2")
res.legend
which.max(summary.out$adjr2)  #12
summary.out$which[12,]
m2_4 = glm(formula = lfey~cAm+cAs+cOc+
             sts+
             bmi+gdp+sch+hiv+
             atmy+logufdth+plio+dpria,data=newdata)
summary(m2_4) # AIC - 16010

# Conclusion: There is no model which in better than case_1 when compared models by using AIC
# Anyway, the best model is m2_1 in here.
```

CASE3 (Polynomial Regression + Interaction Term)

```
full_3 = glm(lfey~
               achl+bmi+gdp+th119+sch+hiv+
               atmy+logufdth+hb2+plio+dpria+
               (gdp*cAm)+(hiv*cAm)+
               (achl*cAs)+(gdp*cAs)+(hiv*cAs)+
               (gdp*cOc)+(th119*cOc)+(hiv*cOc))
vif(full_3)
max(vif(full_3)) # all the multicollinearity variables are removed.


#build base model CASE1
base_3 = glm(lfey~achl,data=newdata)

# model 1  Both direction by stepwise()
m3_1 = step(base_3, scope=list(upper=full_3, lower=~1), direction="both", trace=T)
summary(m3_1)  # aic 16124

# m2 Forward Selection by step()
m3_2 = step(base_3, scope=list(upper=full_3, lower=~1), direction="forward", trace=T)
summary(m3_2) # AIC: 16124

# m3 Backward Elimination by step() based on AIC
m3_3 = step(full_3, direction = "backward", trace=TRUE)
summary(m3_3) # AIC: 16124
```

```
# m3_4 by best subset
#install.packages("leaps")
library(leaps)
regsubsets.out <-
  regsubsets(lfey~
               achl+bmi+gdp+th119+sch+hiv+
               atmy+logufdth+hb2+plio+dpria+
               (gdp*cAm)+(hiv*cAm)+
               (achl*cAs)+(gdp*cAs)+(hiv*cAs)+
               (gdp*cOc)+(th119*cOc)+(hiv*cOc),
             data = newdata,
             nbest = 1,       # 1 best model for each number of predictors
             nvmax = NULL,    # NULL for no limit on number of variables
             force.in = NULL, force.out = NULL,
             method = "exhaustive")
summary.out <- summary(regsubsets.out)
as.data.frame(summary.out$outmat)
res.legend <-
  subsets(regsubsets.out, statistic="adjr2", legend = FALSE, min.size = 5, main = "Adjusted R^2")
res.legend
which.max(summary.out$adjr2)  #12
summary.out$which[19,]
m3_4 = glm(lfey~
             achl+bmi+gdp+th119+sch+hiv+
             atmy+logufdth+plio+dpria+
             (gdp*cAm)+(hiv*cAm)+
             (achl*cAs)+(gdp*cAs)+
             (gdp*cOc)+(th119*cOc),data=newdata)
summary(m3_4) # AIC - 16124
```

```
# Conclusion: There is no model better than CASE_1
# Anyway, best model is m3_1 in here.
```

```
# Step 5: Model Evaluations by N-folds Cross validations
library(boot)
m1_1 = glm(lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
             sts + gdp + log(newdata$under.five.deaths) + plio + th119 + cOc, data = newdata)

m2_1 = glm(lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
             sts + gdp + log(newdata$under.five.deaths) + plio + cOc, data = newdata)

m3_1 = glm(lfey ~ achl + sch + atmy + hiv + dpria + bmi +
             cAs + cAm + gdp + log(newdata$under.five.deaths) + plio + th119 + cOc + hiv:cAm +
             cAm:gdp + cAs:gdp + gdp:cOc + th119:cOc, data = newdata)

mse1 = cv.glm(newdata, m1_1, K=10)$delta
mse2 = cv.glm(newdata, m2_1, K=10)$delta
mse3 = cv.glm(newdata, m3_1, K=10)$delta

errs = cbind(mse1, mse2, mse3)
errs
```

```
> errs = cbind(mse1, mse2, mse3)
> errs
          mse1     mse2     mse3
[1,] 167.0504 167.2727 167.1351
[2,] 167.0504 167.2727 167.1351

#conclusion: So the best model is m1_1
```

So we decided m1_1 as a best model in comparison with error of models.

5.3. Findings

From the above steps, we conclude that the optimal model is m1_1_new.

In conclusion, there are no polynomial variables or interaction term in the optimal model.

We were also able to improve the model by removing Influential Points.

```
> cooksd <- cooks.distance(m1_1)
> influential <- as.numeric(names(cooksd)[(cooksd > (4/2888))])
> influential
  [1]   49   63   64  113  114  115  116  117  118  120  122  127  133  196  310  317  326  327  334  351  352  410  432  433  435  437  438
 [28]  439  440  445  455  456  483  497  514  517  518  519  520  531  532  533  534  535  536  540  624  625  629  633  666  715  720  721
 [55]  744  788  840  842  848  884  888  889  900  901  903  919  920  922  923  924  961  962  982  983 1012 1025 1043 1087 1089 1090 1100
 [82] 1112 1156 1222 1223 1233 1364 1365 1366 1367 1368 1369 1370 1386 1455 1462 1463 1470 1479 1482 1510 1519 1541 1550 1557 1560 1608 1639
[109] 1698 1701 1718 1747 1751 1856 1880 1881 1882 1883 1884 1885 1893 1937 1961 2033 2034 2066 2067 2068 2069 2070 2075 2084 2140 2176 2183
[136] 2185 2186 2189 2190 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2288 2289 2299 2322 2343 2361 2362 2363 2364 2365 2366
[163] 2380 2381 2382 2383 2384 2410 2473 2475 2479 2480 2481 2490 2516 2517 2630 2703 2708 2741 2742 2743 2760 2768 2769 2822 2825 2826 2827
[190] 2832 2833 2890 2895 2896 2907 2908 2910 2911 2912 2913 2914
```

```
newdata2 <- newdata[-influential, ]

m1_1_new = glm(lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
             sts + gdp + log(newdata2$under.five.deaths) + plio + th119 + cOc, data = newdata2)
mse1_new = cv.glm(newdata2, m1_1_new, K=10)$delta
errs2 = cbind(mse1, mse1_new)
errs2
```

```
> errs2 = cbind(mse1, mse1_new)
> errs2
          mse1 mse1_new
[1,] 167.0504 157.8734
[2,] 167.0504 157.8734
```

Our conclusions will be addressed in the following steps.

6. Conclusions and Future Work

6.1. Conclusions

Life Expectancy = 5.746e+01+

(1.003e+00*__Schooling__)+(-1.844e-02*__Adult.Mortality__) + (-4.094e-01*__HIV.AIDS__) +

(3.460e-02*__Diphtheria__) + (3.565e-02*__BMI__)+ (-3.633e+00*(__Status__=0)) + (5.132e-05*__GDP__) +

(-2.801e-01*log(__Under.Five.Deaths__)) + (2.157e-02*__Polio__) + (-9.838e-02*__thinness..1.19.years__)

```
> m1_1_new

Call:  glm(formula = lfey ~ sch + atmy + hiv + dpria + bmi + cAs + cAm +
    sts + gdp + log(newdata2$under.five.deaths) + plio + th119 +
    coc, data = newdata2)
```

With respect to world data, the most significant factors in life expectancy are Schooling and Status. As the number of years of education increases, life expectancy also increases by about one year. It was also found that developing countries are about 3.6 years shorter than developed countries.

Therefore, in order to increase the life expectancy, it can be inferred that the development of education and the formation of infrastructure to become a developed country are very important.

6.2. Limitations

Our limitation is that we can see which x variables can affect life expectancy or not, but it is difficult to interpret it in detail. For example, we predicted that high BMI would lower life expectancy due to adult disease or other factors. However, the higher the BMI, the higher the life expectancy.
Therefore, it is difficult to interpret the variables without background knowledge.

Secondly, because the data are based on world data, the variables may be different for each region and country. Therefore, I think that if we classified the data by region and proceeded, we would have got a more reliable expression.

6.3. Potential Improvements or Future Work

WORLD	ASIA	EUROPE	AFRICA	AMERICA	OCEANIA
	achl	achl		achl	achl
atmy	atmy	atmy	atmy	atmy	
hiv	hiv	hiv	hiv	hiv	hiv
dpra	gdp		dpra		
bmi			bmi		
sts	sts	sts			
gdp		gdp		gdp	
ufdth			ufdth	ufdth	
plio				plio	
th119		th119		th119	
	sch	sch	sch	sch	
	hb				

The table above is a table summarizing the variables affecting the y variable after dividing the world and the world into five categories and proceeding with linear regression.

As mentioned in limitations, if we predict linear regression based on world, environment, disease importance and mortality rate are different in each region. Therefore, if you want to obtain information to increase your life expectancy from a world-based linear regression, this will be unreliable.

Therefore, if you make a prediction by making a linear regression for each region or country, you will get much more reliable information.