Determinants of Life Expectancy

Contents

Determinants of Life Expectancy

Abstract

We investigate the role of several socio-economic factors in determining life expectancy using simple and multiple linear regression model. We test our models for problems such as nonlinearity, heteroscedasticity, multicollinearity etc. In this regard we use standard test such as the Breusch-Pagan test, Correlation Matrix, VIF test etc. We infer that health expenditure per capita, fertility rate, alcohol consumption per capita and the development status of the country are key determinants of life expectancy.

1 Introduction

One of the most important socio-economic feats of the last century was the doubling of life expectancy. This has led to societal transformation and has an impact in several economic and social factors.[1] The level and variation in life expectancy has important implications in aggregate human behaviour, for it affects fertility behaviour, economic growth, human capital investment, pension planning, healthcare etc.[1] It is therefore vital for us to determine the factors that contribute to life expectancy.

In our model, we consider the following factors that determine life expectancy: economic indicators such as GDP per capita, health expenditure, demographic factors such as literacy rate, urban population, development classification on the basis of income level, basic factors determining the standard of living in a country such as sanitation, drinking water access, undernourishment and detrimental factors such as alcohol consumption and smoking prevalence.

- `lex` - Life expectancy at birth, total (years)
- `dp` - GDP per capita (current US$)
- `lit` - Literacy rate, adult total (% of people ages 15 and above)
- `hex` - Current health expenditure per capita (current US$)
- `urb` - Urban Population / Total population (%)
- `unt` - Prevalence of undernourishment (as a % of population)
- `phy` - Number of Physicians (per 1000 people)
- `san` - People using at least basic sanitation services (% of population)
- `dri` - People using at least basic drinking water services (% of population)
- `fer` - Fertility rate, total (births per woman)
- `smo` - Smoking prevalence, total (ages 15+)
- `alc` - Total alcohol consumption per capita (liters of pure alcohol, projected estimates, 15+ years of age)
- `dev` - Development (as per Income Classification)

[1] WHO Report on World Health Statistics Report 2019, - https://apps.who.int/iris/bitstream/handle/10665/311696/WHO-DAD-2019.1-eng.pdf

2 Variable Definitions

2.0.0.1 Life Expectancy

Life expectancy at birth indicates the number of years a newborn infant would live if prevailing patterns of mortality at the time of its birth were to stay the same throughout its life.
Source: United Nations Population Division. World Urbanization Prospects: 2018 Revision.

2.0.0.2 GDP per capita

GDP per capita is gross domestic product divided by midyear population. GDP is the sum of gross value added by all resident producers in the economy plus any product taxes and minus any subsidies not included in the value of the products. It is calculated without making deductions for depreciation of fabricated assets or for depletion and degradation of natural resources. Data are in current U.S. dollars.
Source: World Bank national accounts data, and OECD National Accounts data files.

2.0.0.3 Literacy Rate

Adult literacy rate is the percentage of people ages 15 and above who can both read and write with understanding a short simple statement about their everyday life.
Source: UNESCO Institute for Statistics

2.0.0.4 Health Expenditure

Current expenditures on health per capita in current US dollars. Estimates of current health expenditures include healthcare goods and services consumed during each year.
Source: World Health Organization Global Health Expenditure database

2.0.0.5 Urban Population

Urban population refers to people living in urban areas as defined by national statistical offices. The data are collected and smoothed by United Nations Population Division.
Source: United Nations Population Division. World Urbanization Prospects: 2018 Revision.

2.0.0.6 Prevalence of Undernourishment

Population below minimum level of dietary energy consumption (also referred to as prevalence of undernourishment) shows the percentage of the population whose food intake is insufficient to meet dietary energy requirements continuously. Data showing as 5 may signify a prevalence of undernourishment below 5%.
Source: Food and Agriculture Organization

2.0.0.7 Number of Physicians

Number of physicians for every 1000 people. Physicians include generalist and specialist medical practitioners.The WHO estimates that at least 2.5 medical staff (physicians, nurses and midwives) per 1,000 people are needed to provide adequate coverage with primary care interventions (WHO, World Health Report 2006)
Source: World Health Organization's Global Health Workforce Statistics, OECD, supplemented by country data.

2.0.0.8 Sanitation

The percentage of people using at least basic sanitation services, that is, improved sanitation facilities that are not shared with other households. This indicator encompasses both people using basic sanitation services as well as those using safely managed sanitation services. Improved sanitation facilities include flush/pour flush to piped sewer systems, septic tanks or pit latrines; ventilated improved pit latrines, compositing toilets or pit latrines with slabs.
Source: WHO/UNICEF Joint Monitoring Programme (JMP) for Water Supply, Sanitation and Hygiene

2.0.0.9 Drinking Water

The percentage of people using at least basic water services. This indicator encompasses both people using basic water services as well as those using safely managed water services. Basic drinking water services is defined as drinking water from an improved source, provided collection time is not more than 30 minutes for a round trip. Improved water sources include piped water, boreholes or tubewells, protected dug wells, protected springs, and packaged or delivered water.
Source: WHO/UNICEF Joint Monitoring Programme (JMP) for Water Supply, Sanitation and Hygiene

2.0.0.10 Fertility Rate

Total fertility rate represents the number of children that would be born to a woman if she were to live to the end of her childbearing years and bear children in accordance with age-specific fertility rates of the specified year.
Source: United Nations Population Division. World Population Prospects: 2019 Revision.

2.0.0.11 Smoking Prevalence

Prevalence of smoking is the percentage of men and women ages 15 and over who currently smoke any tobacco product on a daily or non-daily basis. It excludes smokeless tobacco use. The rates are age-standardized.
Source: World Health Organization, Global Health Observatory Data Repository

2.0.0.12 Alcohol Consumption

Total alcohol per capita consumption is defined as the total (sum of recorded and unrecorded alcohol) amount of alcohol consumed per person (15 years of age or older) over a calendar year, in litres of pure alcohol, adjusted for tourist consumption.
Source: World Health Organization, Global Health Observatory Data Repository

2.0.0.13 Development Status

The `dev` gives the development status of a country as per the income classification given by the World Bank. The status was allotted by WB as per the 2013 gross national income (GNI) per capita estimates.

As of 1 July 2014, low-income economies are defined as those with a GNI per capita, calculated using the World Bank Atlas method, of $1,045 or less in 2013; middle-income economies are those with a GNI per capita of more than $1,045 but less than $12,746; high-income economies are those with a GNI per capita of $12,746 or more. Lower-middle-income and upper-middle-income economies are separated at a GNI per capita of $4,125.

3 Data Preprocessing

Since the data at hand is largely to do with socio-economic and macrovariables, and since the regression analysis is cross-sectional, data cleaning is done with utmost care. We ensured that the sancity of the data be preserved for us to make valuable interpretations.

One method of dealing with missing vales is *imputation*. For those countries for which the data is available at a reasonably close point of time, imputations have been performed.

We relied on several reliable websites such as those of World Bank, United Nations, UNICEF, WHO etc., to find these missing values. [2]

[2] Data Cleaning record can be found at: https://drive.google.com/drive/u/0/folders/12IC4U59cONn3Tz-FAveWqez5LYmR0r6h

4 Handling Dataframes

4.1 Importing

The cleaned dataset is imported into R as a dataframe, for our statistical analysis. This involves loading the `.xlxs` or `.csv` file into working directory and importing using the `read_excel()` or the `read.csv()` command.

```
library("readxl")
df <- read_excel("lifeexpectancy.xlsx")
```

We shall hereon, refer to our dataframe as `df`. The `dim()` function returns the dimensions of the dataframe, while the `names()` function returns the variables involves. The `head(df, number = n)` returns the first n rows of `df`, while the `head(df, number = n)` returns the last n rows.

```
dim(df)
```

```
## [1] 195  15
```

```
names(df)
```

```
##  [1] "Entity" "Code"   "lex"    "gdp"    "lit"    "hex"    "urb"    "unt"
##  [9] "phy"    "san"    "dri"    "fer"    "smo"    "alc"    "dev"
```

```
head(df)
```

```
## # A tibble: 6 x 15
##   Entity Code    lex    gdp   lit    hex   urb   unt   phy   san   dri   fer
##   <chr>  <chr> <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Aruba  ABW    75.6 25534.  97.8 2500    43.0  26.9 0.391  97.5  97.9  1.83
## 2 Afgha~ AFG    62.9   614.  43.0   60.1  24.6  26.9 0.304  39.4  58.8  5.16
## 3 Angola AGO    60.9  5408.  66.0  132.   62.7  28.1 0.215  46.1  53.5  5.86
## 4 Alban~ ALB    78.0  4579.  97.2  313.   56.4   5.9 1.27   97.7  91.0  1.69
## 5 Unite~ ARE    76.9 43752.  92.8 1613.   85.4   3.7 2.03   98.6  96.1  1.60
## 6 Argen~ ARG    76.3 12335.  99.0 1087.   91.4   3.4 3.91   94.2  99.0  2.31
## # ... with 3 more variables: smo <dbl>, alc <dbl>, dev <chr>
```

```
tail(df)
```

```
## # A tibble: 6 x 15
##   Entity Code    lex   gdp   lit   hex   urb   unt    phy   san   dri   fer
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Vanua~ VUT    71.7 3088.  84.7 109.   24.9   6.7 0.186   40.2  89.3  3.94
## 2 Samoa  WSM    74.5 3938.  99.0 259.   19.1   2.8 0.34    98.0  96.4  4.09
## 3 Yemen  YEM    64.5 1674.  54.1  79.7  34.2  31.4 0.310   57.4  61.4  4.21
## 4 South~ ZAF    61.0 6433.  92.9 510.   64.3   5.2 0.754   72.9  91.4  2.51
## 5 Zambia ZMB    60.8 1763.  83.0  66.6  41.4  45    0.77    25.9  58.1  5.03
## 6 Zimba~ ZWE    59.4 1435.  88.7  89.1  32.5  46.9 0.0763  38.3  65.5  3.97
## # ... with 3 more variables: smo <dbl>, alc <dbl>, dev <chr>
```

4.2 Missing Values

Although, through data preprocessing, we have filled in several missing values, a significant number of missing values could still be present in the dataframe. Since our results hinge on this factor, we now chack for the number of missing values in `df`.

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

```
## [1] 35
```

```
colSums(is.na(df))
```

```
## Entity   Code    lex    gdp    lit    hex    urb    unt    phy    san    dri
##      0      0      0      0      2      6      1      8      8      2      1
##    fer    smo    alc    dev
##      0      4      3      0
```

```
missingVal <- df[!complete.cases(df), ]
head(missingVal, n = 10)
```

```
## # A tibble: 10 x 15
##    Entity Code    lex    gdp   lit    hex   urb   unt   phy   san   dri   fer
##    <chr>  <chr> <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Curac~ CUW    77.8 2.03e4  99   3355    89.4  NA    NA    99.0  99.5  2
##  2 Eritr~ ERI    64.2 8.11e2  76.6   24.5  36    NA    NA    11.7  51.1  4.27
##  3 Faero~ FRO    81.7 5.94e4  NA   3500    41.5  NA    NA    90.9 100    2.6
##  4 Guam   GUM    79.2 3.44e4  99.4   NA    94.4  NA    NA    90.4  99.7  2.41
##  5 Saint~ LCA    75.1 8.74e3  90    501.   18.5  NA    NA    88.4  98.2  1.48
##  6 Liech~ LIE    82.1 1.79e5  99     NA    14.3  NA    NA   100.0 100    1.59
##  7 Macao  MAC    83.5 9.38e4  96.5   NA   100    11.9  2.82  NA   100    1.20
##  8 Saint~ MAF    79.3 6.68e4  99     NA    NA    NA    NA    NA    NA    1.81
##  9 Frenc~ PYF    76.4 2.26e4  98     NA    61.6   4.1  2.13  96.9 100    2.01
## 10 Unite~ VIR    79.0 3.36e4  NA     NA    95.2  NA    NA    99.3  98.7  2.09
## # ... with 3 more variables: smo <dbl>, alc <dbl>, dev <chr>
```

The above countries are predominantly island states with a very small population. Macau is the largest of the lot with a population of about 6 lakhs.

Since we have a small number of NA entries, we alter our dataframe to omit these rows (i.e, omit the countries). On doing so one observes a reduction in the number of observations to 185.

```
df <- na.omit(df)
dim(df)
```

```
## [1] 185  15
```

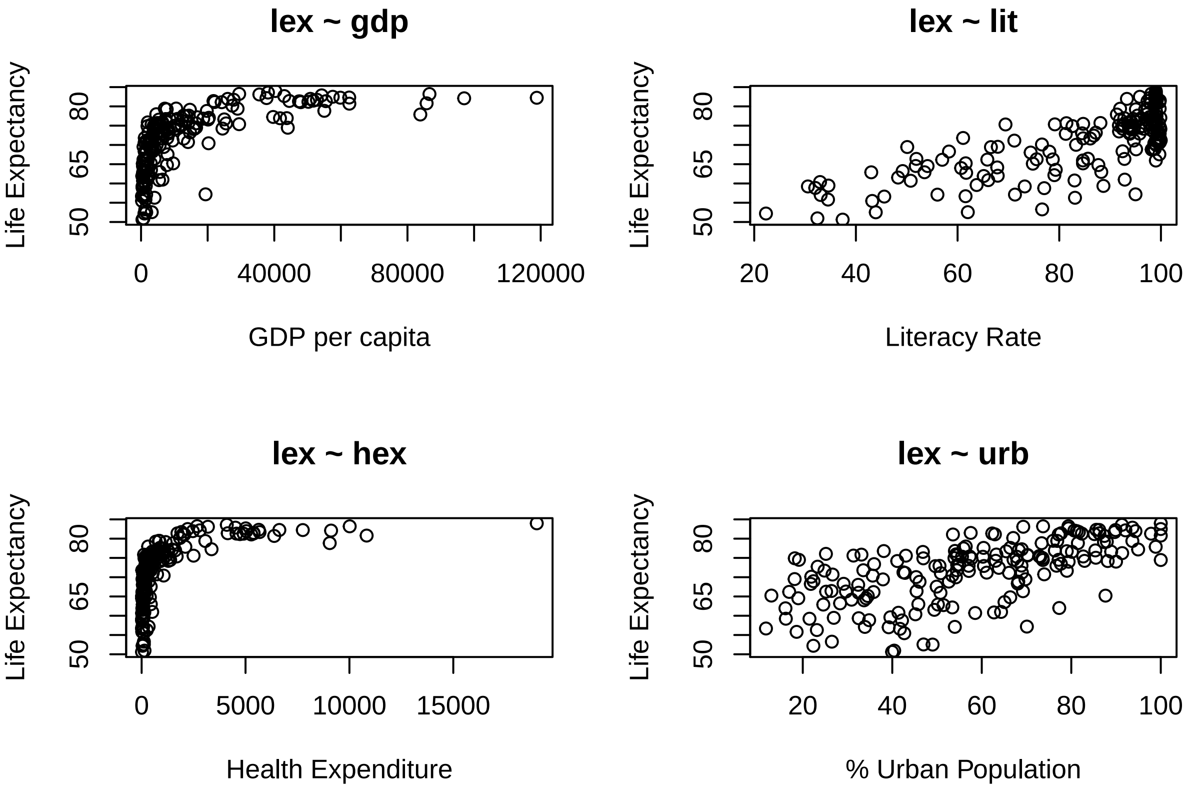
5 Visualizing the dataframe

The protocol to be followed before performing linear regression is to first visualize the data via scatter plot. Even mere eyeballing will show clear patterns in the data. These can be taken as cues for modelling the data, aptly. Note that, these plots depict only trends of non-linearity in the variables and not on parameters. Non-linearity in parameters is fatal for linear regression.

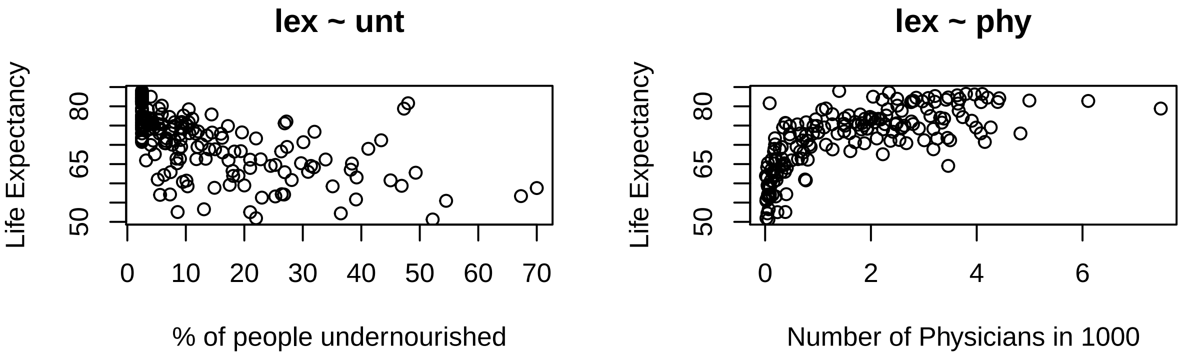
5.1 Scatter Plots

Scatter plotting the variables in the given dataframe with respect to life expectancy yields the following curves.

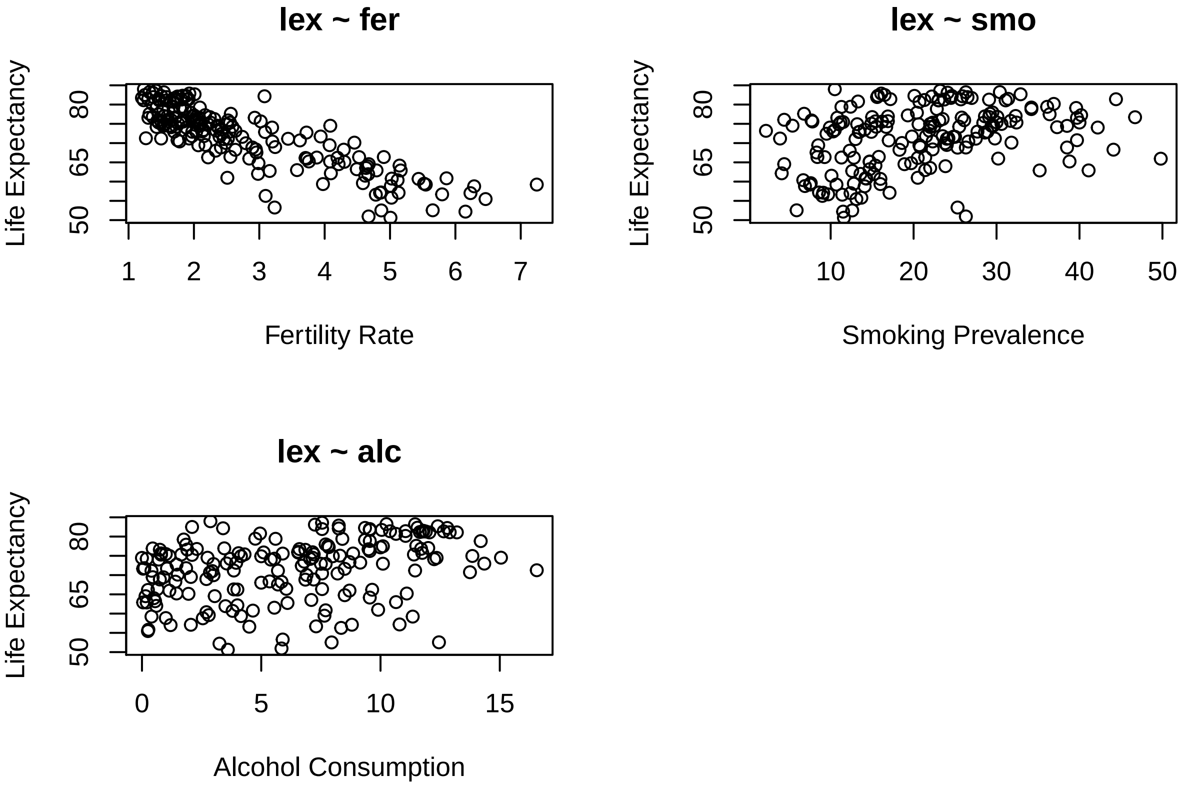
```
par(mfrow=c(2,2))
plot(df$gdp,df$lex, xlab ="GDP per capita", ylab = "Life Expectancy", main = "lex ~ gdp")
plot(df$lit,df$lex, xlab ="Literacy Rate", ylab = "Life Expectancy", main = "lex ~ lit")
plot(df$hex,df$lex, xlab ="Health Expenditure", ylab = "Life Expectancy", main = "lex ~ hex")
plot(df$urb,df$lex, xlab ="% Urban Population", ylab = "Life Expectancy", main = "lex ~ urb")
```



```
plot(df$unt,df$lex, xlab ="% of people undernourished",
     ylab = "Life Expectancy", main = "lex ~ unt")
plot(df$phy,df$lex, xlab ="Number of Physicians in 1000",
     ylab = "Life Expectancy", main = "lex ~ phy")
plot(df$san,df$lex, xlab ="% people using basic sanitation",
     ylab = "Life Expectancy", main = "lex ~ san")
plot(df$dri,df$lex, xlab ="% people with access to drinking water",
     ylab = "Life Expectancy", main = "lex ~ dri")
```



```r
plot(df$fer,df$lex, xlab ="Fertility Rate", ylab = "Life Expectancy", main = "lex ~ fer")
plot(df$smo,df$lex, xlab ="Smoking Prevalence", ylab = "Life Expectancy", main = "lex ~ smo")
plot(df$alc,df$lex, xlab ="Alcohol Consumption", ylab = "Life Expectancy", main = "lex ~ alc")
```



6 Simple Linear Regressions

We now wish to understand the contributions of individual explanatory variables on the dependent variable, by ignoring the existence of all other explanatory variables. This is carried out by simple linear regression.

The simple linear regression model is the following where y is the dependent variable `lex` and the x_i are the explanatory variables `gdp`, `dri` etc.

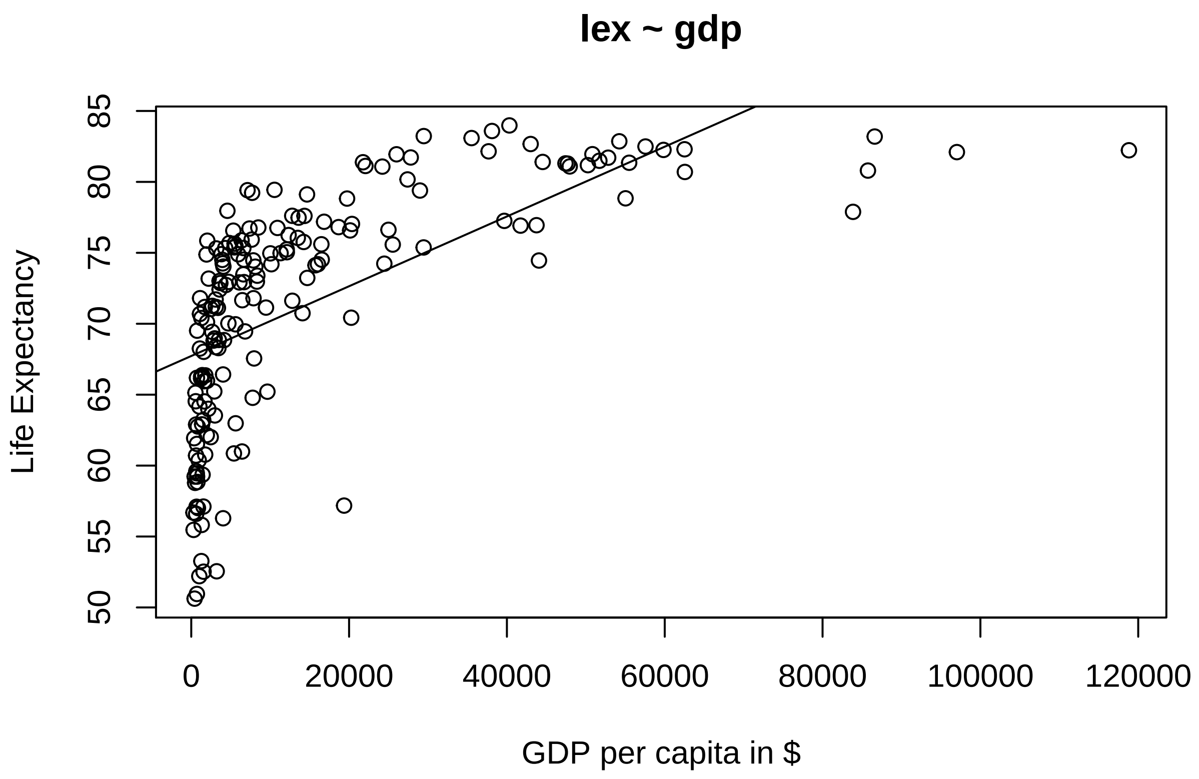
$$y = \beta_0 + \beta_1(x_i) \tag{1}$$

6.1 On GDP

```
lm.gdp <- lm(lex ~ gdp, df)
summary(lm.gdp)
```

```
##
## Call:
## lm(formula = lex ~ gdp, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.210  -3.817   1.576   4.768   9.934
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.773e+01  5.761e-01  117.56   <2e-16 ***
## gdp         2.459e-04  2.262e-05   10.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.357 on 183 degrees of freedom
## Multiple R-squared:  0.3925, Adjusted R-squared:  0.3892
## F-statistic: 118.2 on 1 and 183 DF,  p-value: < 2.2e-16
```

```
plot(df$gdp,df$lex, xlab ="GDP per capita in $",
     ylab = "Life Expectancy", main = "lex ~ gdp")
abline(lm.gdp)
```



6.2 The Preston Curve

The Preston curve is an empirical cross-sectional relationship between life expectancy and real per capita income or GDP, named adter Samuel H. Preston who first described it in 1975. Preston studied the relationship for the 1900s, 1930s and the 1960s and found it held for each of the three decades.[3]

A better fit for capturing the relationship between the above variables is a linear-log model. This model would be:

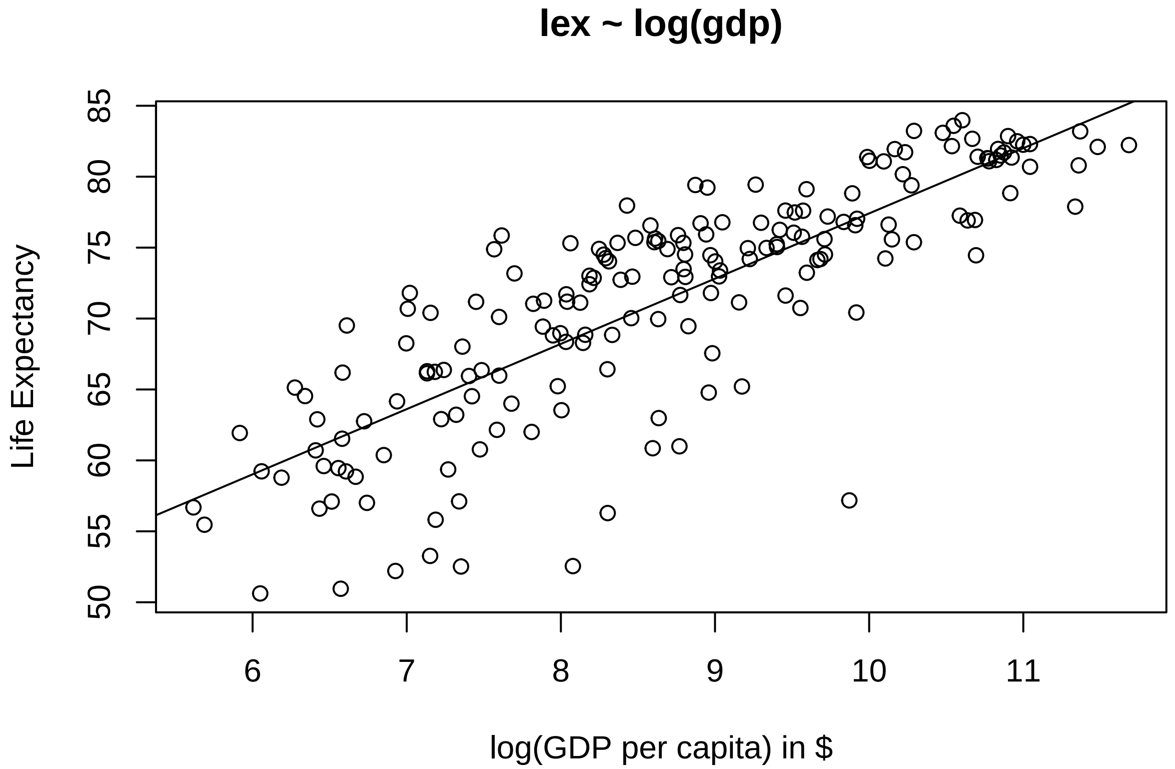
$$lex = \beta_0 + \beta_1[log(gdp)] \tag{2}$$

```
lngdp <- log(df$gdp)
lm.lngdp <- lm(lex ~ lngdp, df)
summary(lm.lngdp)
```

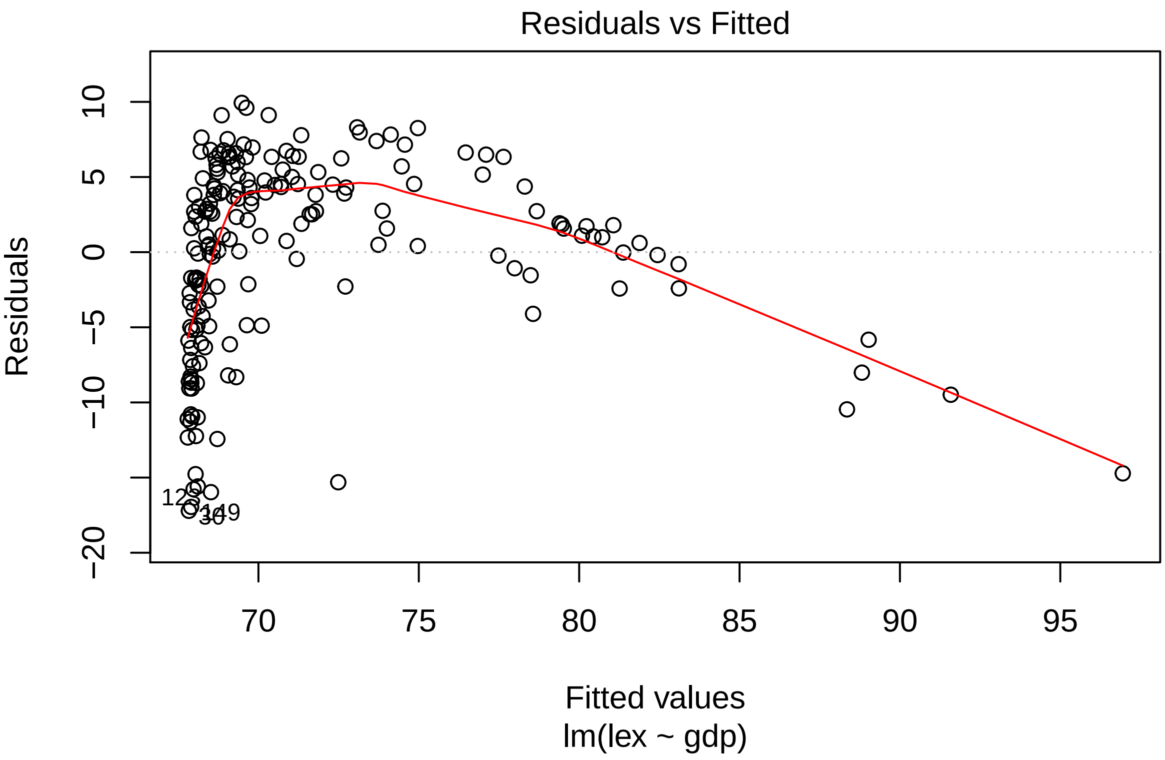
```
##
## Call:
## lm(formula = lex ~ lngdp, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.6411  -2.3235   0.4256   3.3504   9.4072
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  31.4210     2.0911   15.03   <2e-16 ***
## lngdp         4.5992     0.2374   19.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.669 on 183 degrees of freedom
## Multiple R-squared:  0.6723, Adjusted R-squared:  0.6705
## F-statistic: 375.5 on 1 and 183 DF,  p-value: < 2.2e-16
```

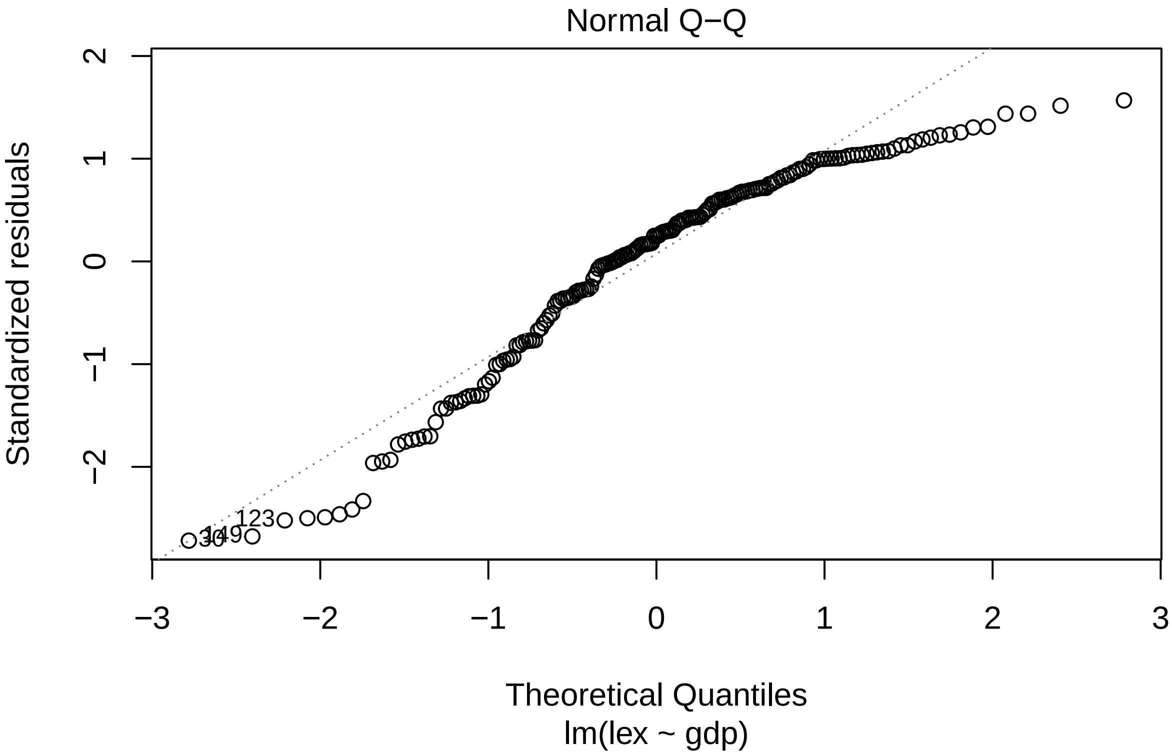
```
plot(lngdp,df$lex, xlab ="log(GDP per capita) in $",
     ylab = "Life Expectancy", main = "lex ~ log(gdp)")
abline(lm.lngdp)
```



```
plot(lm.gdp, which = 1)
```



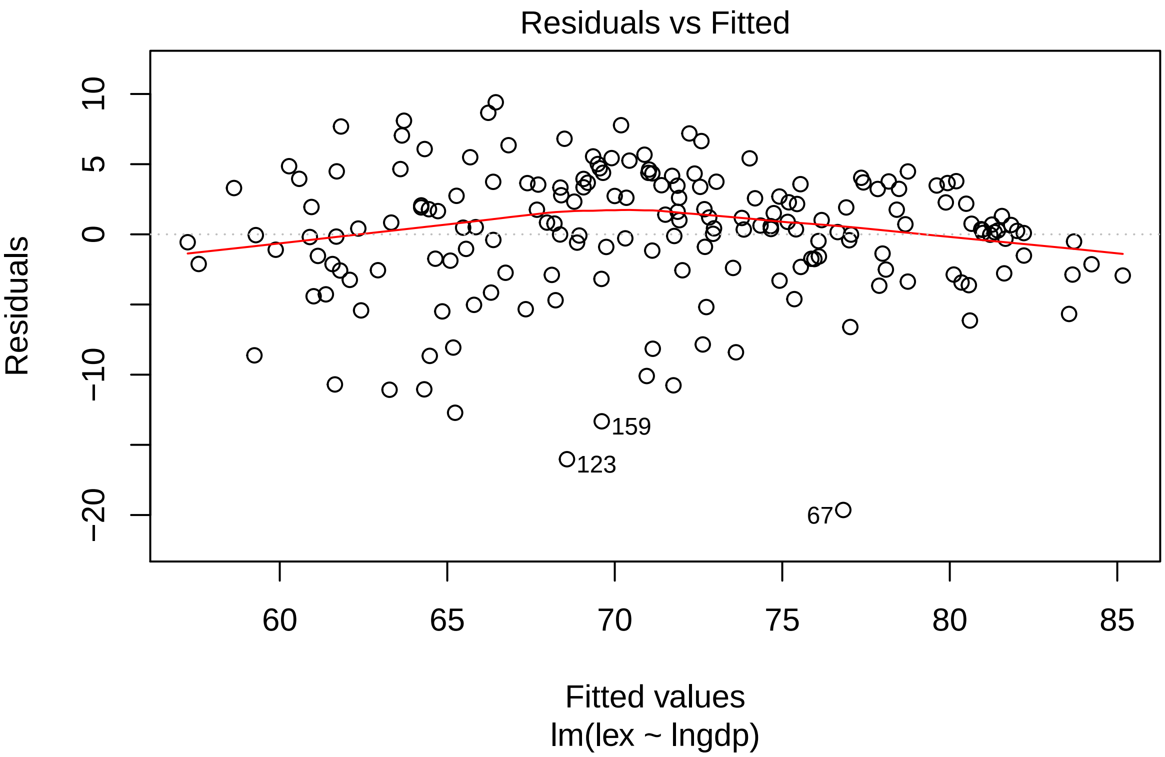
```
plot(lm.gdp, which = 1+1)
```



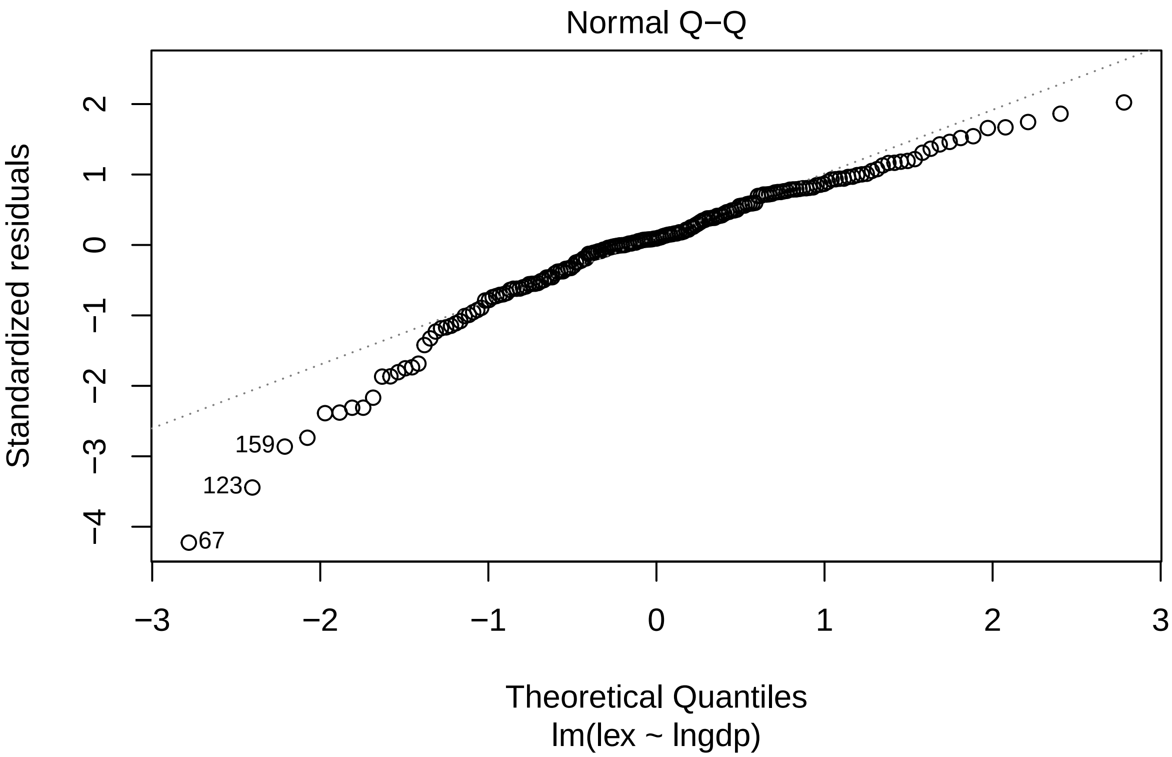
The first plot (residuals vs. fitted values) is a simple scatterplot between residuals and predicted values. This should look more or less random.

The second plot (normal Q-Q) is a normal probability plot. It will give a straight line if the errors are distributed normally.

```
plot(lm.lngdp, which = 1)
```



```
plot(lm.lngdp, which = 1+1)
```



6.3 Other simple regressions

```
lm.lit <- lm(lex ~ lit, df)
lm.hex <- lm(lex ~ hex, df)
lm.urb <- lm(lex ~ urb, df)
lm.unt <- lm(lex ~ unt, df)
lm.phy <- lm(lex ~ phy, df)
lm.san <- lm(lex ~ san, df)
lm.dri <- lm(lex ~ dri, df)
lm.fer <- lm(lex ~ fer, df)
lm.smo <- lm(lex ~ smo, df)
lm.alc <- lm(lex ~ alc, df)
```

The summary statistics of the above regressions are presented below:

```
summary(lm.lit)
```

```
##
## Call:
## lm(formula = lex ~ lit, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.4678  -3.2080   0.7355   4.1427   9.2617
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 44.07112    1.73065   25.46   <2e-16 ***
## lit          0.32186    0.01988   16.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.23 on 183 degrees of freedom
## Multiple R-squared:  0.5889, Adjusted R-squared:  0.5866
## F-statistic: 262.1 on 1 and 183 DF,  p-value: < 2.2e-16
```

```
summary(lm.hex)
```

```
##
## Call:
## lm(formula = lex ~ hex, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.608  -4.285   2.061   5.161   9.317
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.904e+01  5.671e-01 121.729  < 2e-16 ***
## hex         1.869e-03  2.107e-04   8.874 6.36e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.82 on 183 degrees of freedom
## Multiple R-squared:  0.3009, Adjusted R-squared:  0.297
## F-statistic: 78.75 on 1 and 183 DF,  p-value: 6.363e-16
```

```r
summary(lm.urb)
```

```
##
## Call:
## lm(formula = lex ~ urb, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.009  -2.847   1.061   4.070  12.367
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 58.44736    1.24807   46.83   <2e-16 ***
## urb          0.22461    0.02012   11.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.29 on 183 degrees of freedom
## Multiple R-squared:  0.4051, Adjusted R-squared:  0.4019
## F-statistic: 124.6 on 1 and 183 DF,  p-value: < 2.2e-16
```

```r
summary(lm.unt)
```

```
##
## Call:
## lm(formula = lex ~ unt, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4802  -3.3219   0.8138   4.2188  21.7646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 76.08428    0.66911  113.71   <2e-16 ***
## unt         -0.35524    0.03522  -10.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.538 on 183 degrees of freedom
## Multiple R-squared:  0.3573, Adjusted R-squared:  0.3538
## F-statistic: 101.8 on 1 and 183 DF,  p-value: < 2.2e-16
```

```r
summary(lm.phy)
```

```
##
## Call:
## lm(formula = lex ~ phy, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6584  -3.3924   0.4393   4.1954  15.5724
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.8885     0.6537   99.26   <2e-16 ***
## phy           3.9020     0.2959   13.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.84 on 183 degrees of freedom
## Multiple R-squared:  0.4872, Adjusted R-squared:  0.4844
## F-statistic: 173.9 on 1 and 183 DF,  p-value: < 2.2e-16
```

```
summary(lm.san)
```

```
##
## Call:
## lm(formula = lex ~ san, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3501  -2.5100   0.2222   3.1650   9.0814
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.88319    0.84242   63.96   <2e-16 ***
## san          0.23666    0.01059   22.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.223 on 183 degrees of freedom
## Multiple R-squared:  0.7319, Adjusted R-squared:  0.7305
## F-statistic: 499.7 on 1 and 183 DF,  p-value: < 2.2e-16
```

```
summary(lm.dri)
```

```
##
## Call:
## lm(formula = lex ~ dri, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4064  -2.8103  -0.0595   3.4810  12.4492
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.01249    1.70673   21.69   <2e-16 ***
## dri          0.39856    0.01941   20.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.488 on 183 degrees of freedom
## Multiple R-squared:  0.6972, Adjusted R-squared:  0.6956
## F-statistic: 421.4 on 1 and 183 DF,  p-value: < 2.2e-16
```

```
summary(lm.fer)
```

```
##
## Call:
## lm(formula = lex ~ fer, data = df)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0701  -2.5672   0.2982   2.9699  12.0443
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.4348     0.7211  118.47   <2e-16 ***
## fer          -4.9758     0.2294  -21.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.316 on 183 degrees of freedom
## Multiple R-squared:   0.72,  Adjusted R-squared:  0.7184
## F-statistic: 470.5 on 1 and 183 DF,  p-value: < 2.2e-16
```

```r
summary(lm.smo)
```

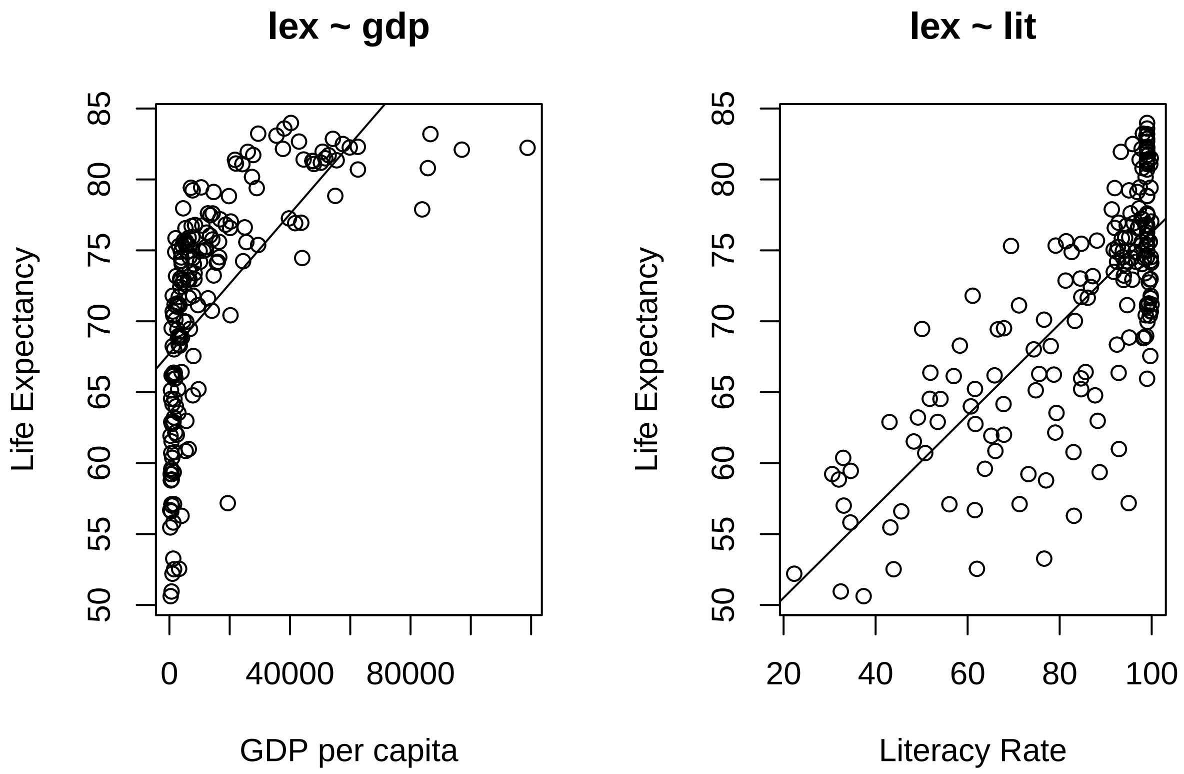
```
##
## Call:
## lm(formula = lex ~ smo, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.9646  -5.2379   0.6857   5.7675  15.5555
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  65.4407     1.3131  49.835  < 2e-16 ***
## smo           0.2842     0.0567   5.013 1.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.648 on 183 degrees of freedom
## Multiple R-squared:  0.1207, Adjusted R-squared:  0.1159
## F-statistic: 25.13 on 1 and 183 DF,  p-value: 1.261e-06
```

```r
summary(lm.alc)
```

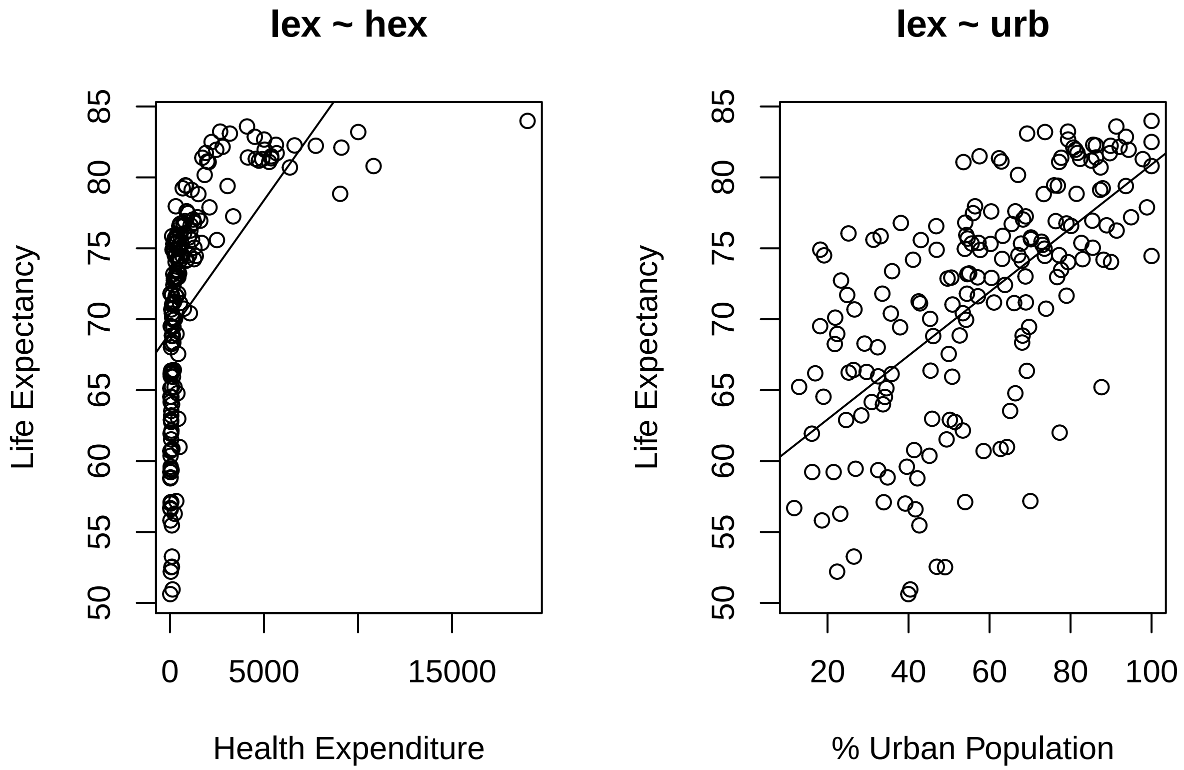
```
##
## Call:
## lm(formula = lex ~ alc, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.802  -4.603   1.705   5.727  14.688
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  67.4780     1.0273  65.684  < 2e-16 ***
## alc           0.6323     0.1384   4.569 8.97e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.727 on 183 degrees of freedom
## Multiple R-squared:  0.1024, Adjusted R-squared:  0.09751
## F-statistic: 20.88 on 1 and 183 DF,  p-value: 8.974e-06
```

The various regression lines are shown here.

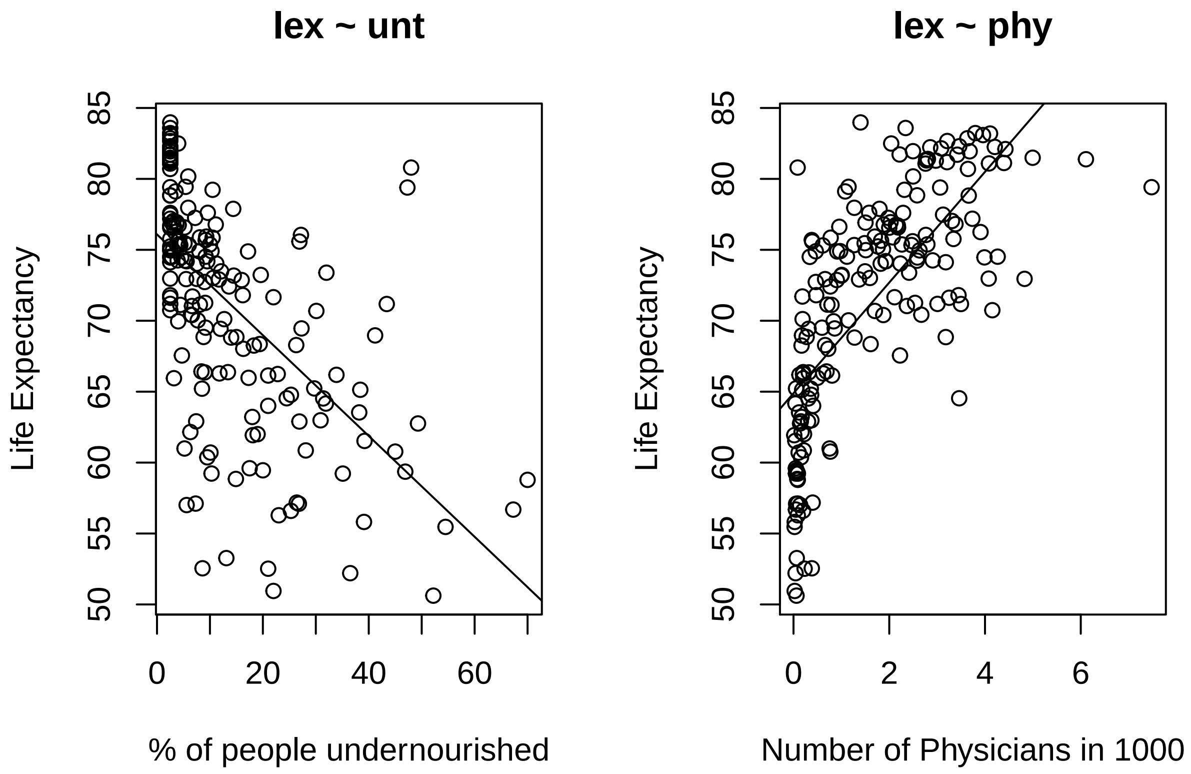
```
par(mfrow=c(1,2))
plot(df$gdp,df$lex, xlab ="GDP per capita", ylab = "Life Expectancy", main = "lex ~ gdp")
abline(lm.gdp)
plot(df$lit,df$lex, xlab ="Literacy Rate", ylab = "Life Expectancy", main = "lex ~ lit")
abline(lm.lit)
```



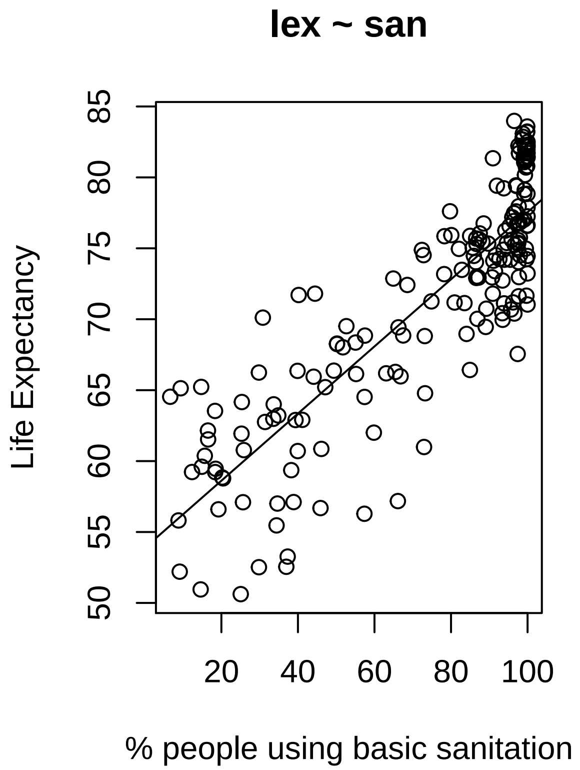
```
plot(df$hex,df$lex, xlab ="Health Expenditure", ylab = "Life Expectancy", main = "lex ~ hex")
abline(lm.hex)
plot(df$urb,df$lex, xlab ="% Urban Population", ylab = "Life Expectancy", main = "lex ~ urb")
abline(lm.urb)
```

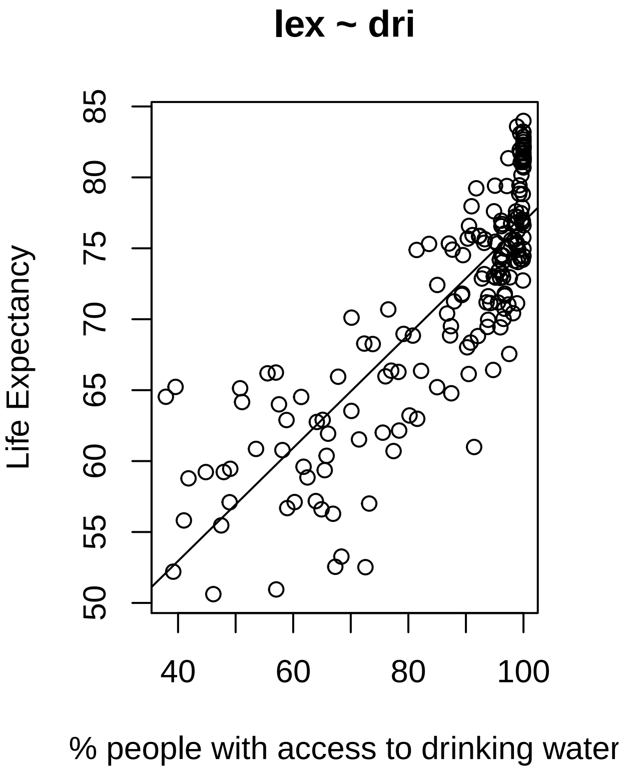


```
plot(df$unt,df$lex, xlab ="% of people undernourished",
     ylab = "Life Expectancy", main = "lex ~ unt")
abline(lm.unt)
plot(df$phy,df$lex, xlab ="Number of Physicians in 1000",
     ylab = "Life Expectancy", main = "lex ~ phy")
abline(lm.phy)
```

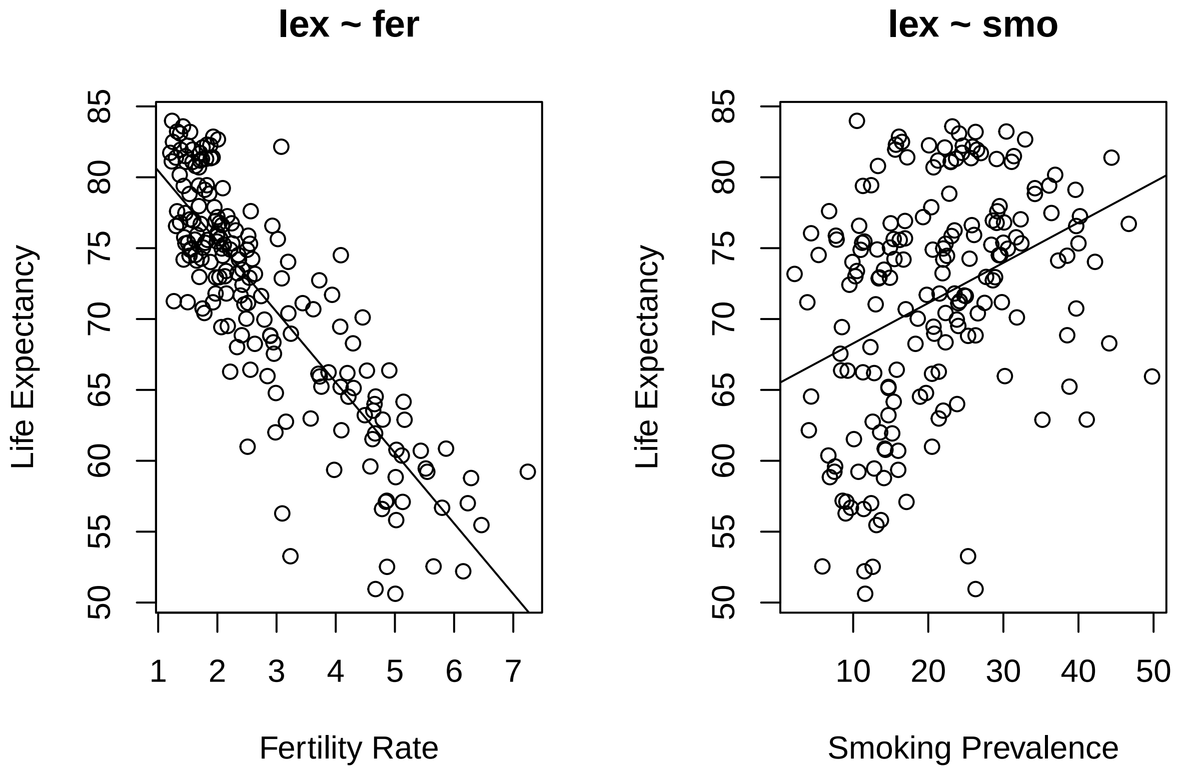


```
plot(df$san,df$lex, xlab ="% people using basic sanitation",
     ylab = "Life Expectancy", main = "lex ~ san")
abline(lm.san)
plot(df$dri,df$lex, xlab ="% people with access to drinking water",
     ylab = "Life Expectancy", main = "lex ~ dri")
abline(lm.dri)
```

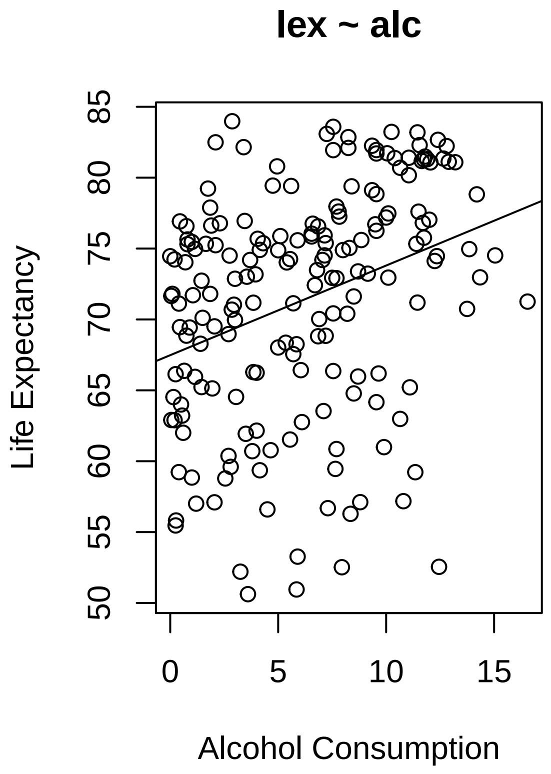




```
plot(df$fer,df$lex, xlab ="Fertility Rate",
     ylab = "Life Expectancy", main = "lex ~ fer")
abline(lm.fer)
plot(df$smo,df$lex, xlab ="Smoking Prevalence",
     ylab = "Life Expectancy", main = "lex ~ smo")
abline(lm.smo)
```



```r
plot(df$alc,df$lex, xlab ="Alcohol Consumption",
     ylab = "Life Expectancy", main = "lex ~ alc")
abline(lm.alc)
```



6.4 The Categorical Variable

Our categorical variable here is `dev`, which represents the development status as per income classification. The levels of development are four in number -

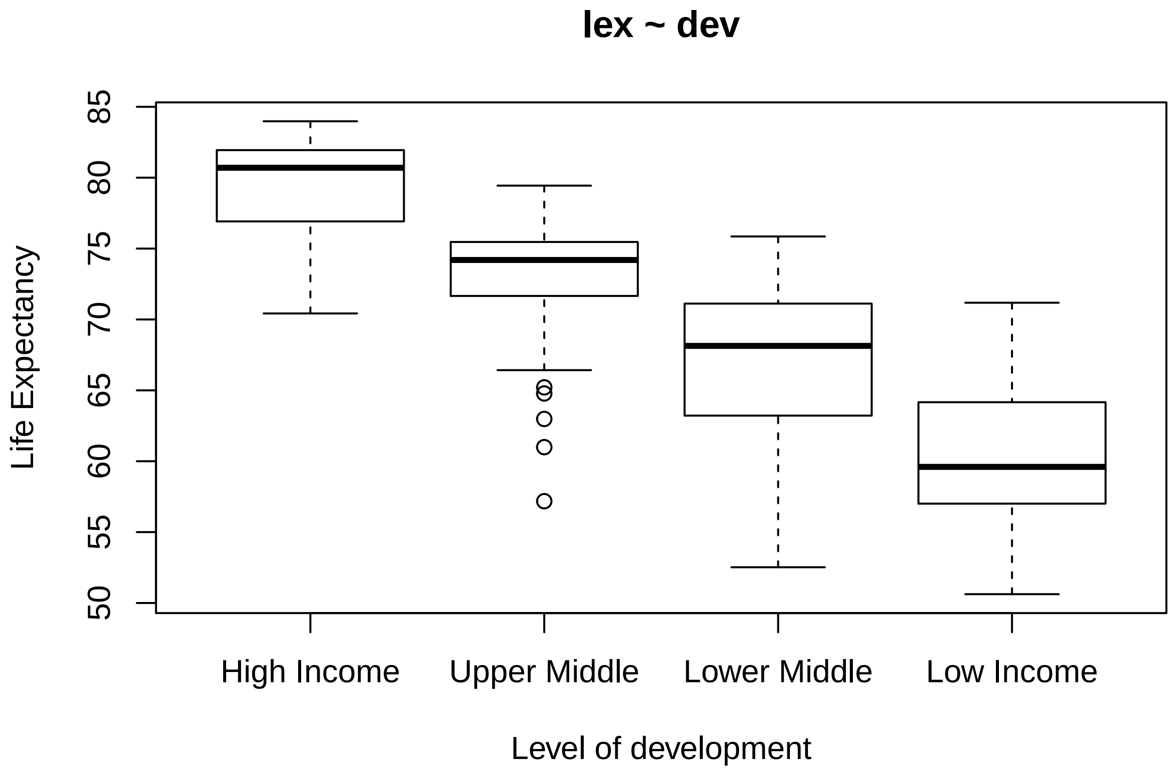
- High Income
- Upper Middle
- Lower Middle
- Low Income

Creating a factor variable for different income level groups

```
df$dev.f <- factor(df$dev, levels =c("High Income", "Upper Middle",
                                     "Lower Middle", "Low Income"),
                                      ordered = is.ordered(df$dev))
```

Boxplot for the categorical variable

```
plot(df$dev.f,df$lex, xlab ="Level of development",
     ylab = "Life Expectancy", main = "lex ~ dev")
```



Finding the mean of life expectancy (lex) for each level in dev

```
tapply(df$lex, df$dev.f, mean)
```

```
##  High Income Upper Middle Lower Middle   Low Income
##     79.28618     72.98897     66.64261     60.47252
```

6.5 Contrast Matrix

This is a matrix in which the rows sum to one, that we use to multiply our matrix of coefficients by in order to make those coefficients estimable. Its rows indicate the different linear combinations of contrasts that we are testing and its columns indicate which factors (coefficients) are being compared.

```
contrasts(df$dev.f) = contr.treatment(4)
```

```
summary(lm(lex ~ dev.f, df))
```

```
##
## Call:
## lm(formula = lex ~ dev.f, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.809  -2.474   1.054   2.814  10.707
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  79.2862     0.6219 127.488  < 2e-16 ***
## dev.f2       -6.2972     0.8960  -7.029 4.11e-11 ***
## dev.f3      -12.6436     0.9306 -13.586  < 2e-16 ***
## dev.f4      -18.8137     1.0710 -17.567  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.695 on 181 degrees of freedom
## Multiple R-squared:  0.6722, Adjusted R-squared:  0.6668
## F-statistic: 123.7 on 3 and 181 DF,  p-value: < 2.2e-16
```

7 Multiple Linear Regression - The Baseline Model

7.1 The Baseline Model

Our baseline model includes all the forementioned explanatory variables. It can be represented as shown

$$lex = \beta_0 + \beta_1(gdp) + \beta_2(lit) + \beta_3(hex) + \beta_4(urb)$$

$$+\beta_5(unt) + \beta_6(phy) + \beta_7(san) + \beta_8(dri)$$

$$+\beta_9(fer) + \beta_{10}(smo) + \beta_{11}(alc) + \beta_{12}(dev)$$

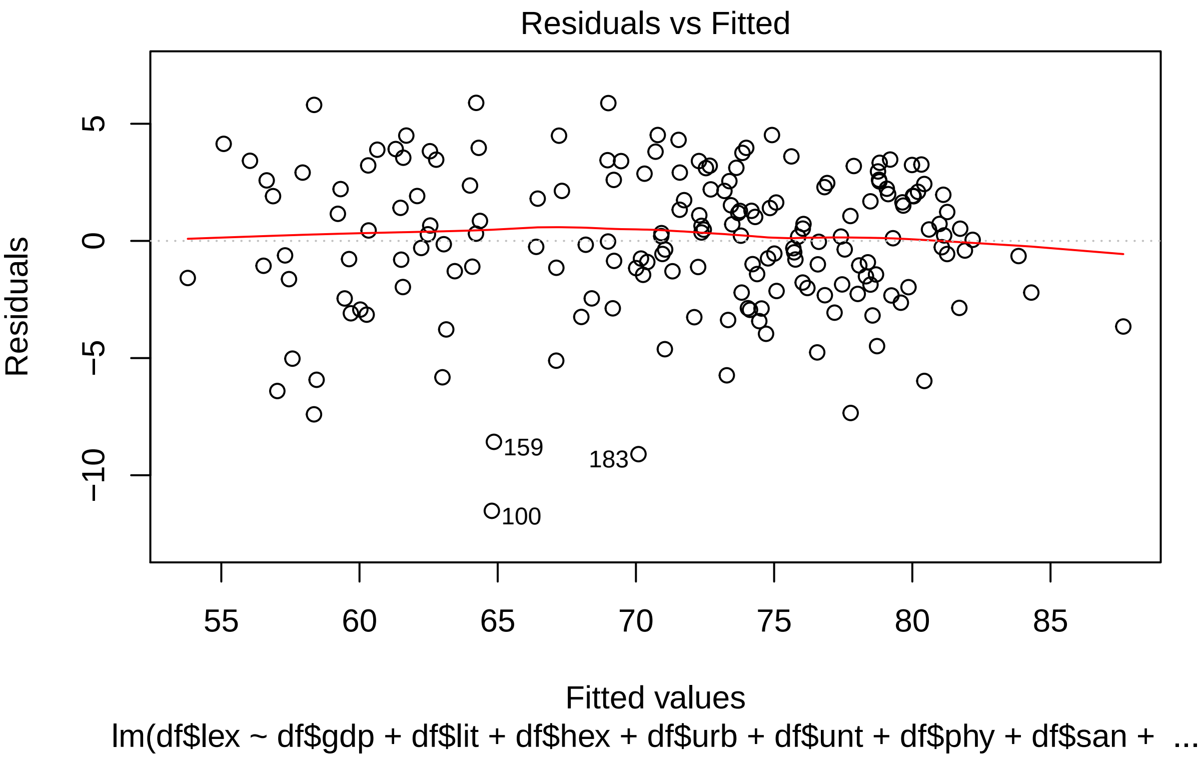
```
ml1 <- lm(df$lex ~ df$gdp+df$lit+df$hex+df$urb+df$unt+df$phy
          +df$san+df$dri+df$fer+df$smo+df$alc+df$dev.f)
summary(ml1)
```

```
##
## Call:
## lm(formula = df$lex ~ df$gdp + df$lit + df$hex + df$urb + df$unt +
##     df$phy + df$san + df$dri + df$fer + df$smo + df$alc + df$dev.f)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.519  -1.778   0.204   2.211   5.891
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.361e+01  4.534e+00  14.030  < 2e-16 ***
## df$gdp       2.081e-05  2.405e-05   0.865  0.38811
## df$lit       6.962e-02  2.527e-02   2.755  0.00651 **
## df$hex       4.398e-04  1.670e-04   2.632  0.00926 **
## df$urb      -2.265e-03  1.521e-02  -0.149  0.88182
## df$unt      -4.262e-02  2.451e-02  -1.739  0.08391 .
## df$phy       6.384e-01  2.601e-01   2.454  0.01513 *
## df$san       4.681e-02  2.193e-02   2.134  0.03428 *
## df$dri       5.499e-02  3.855e-02   1.426  0.15557
## df$fer      -1.646e+00  3.881e-01  -4.242 3.63e-05 ***
## df$smo       2.601e-02  2.663e-02   0.977  0.33013
## df$alc      -3.086e-01  7.165e-02  -4.307 2.79e-05 ***
## df$dev.f2   -2.239e+00  8.458e-01  -2.647  0.00888 **
## df$dev.f3   -2.674e+00  1.134e+00  -2.359  0.01948 *
## df$dev.f4   -1.584e+00  1.572e+00  -1.008  0.31507
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.125 on 170 degrees of freedom
## Multiple R-squared:  0.8636, Adjusted R-squared:  0.8524
## F-statistic:  76.9 on 14 and 170 DF,  p-value: < 2.2e-16
```

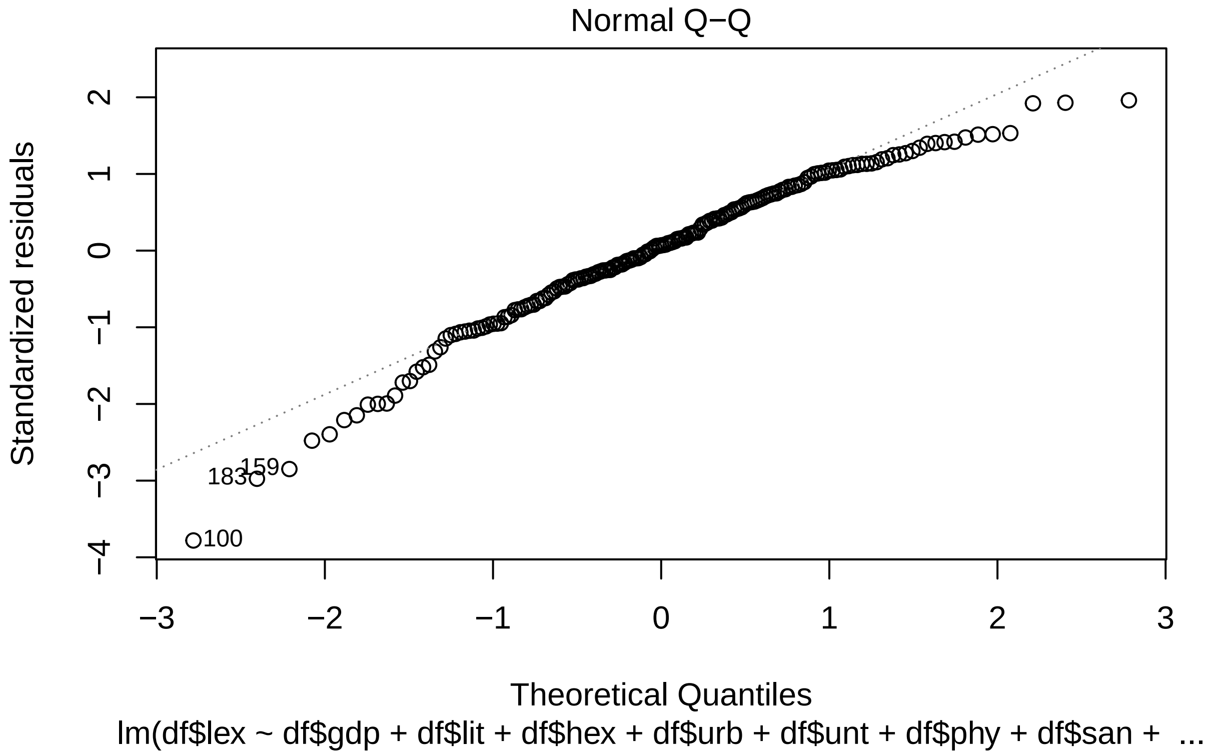
7.2 Observations

- The variable `gdp` which, under our simple linear model was a highly significant determinant of life expectancy `lex`, turns out to be insignificant here.
- The variable `urb` under our simple linear model was a highly significant determinant of life expectancy `lex`, turns out to be highly insignificant here with the p value greater than 0.8
- The variable `urb` is only significant under 10
- The variable `dri` under our simple linear model was a highly significant determinant of life expectancy `lex`, turns out to be insignificant here. (p value of 0.15)
- The variable `smo` under our simple linear model was a highly significant determinant of life expectancy `lex`, turns out to be insignificant here. (p value of 0.33)
- Amongst our categorical variable `dev.f` , the low income factor turned out to be insignificant.
- The adjusted R-squared for the model turns out to be 0.8524. The low residual standard error points out to the accuracy of the model.
- The residual plot for the multiple regression as shown below indicates no pattern amongst the residuals. The absence of a pattern indicates the model fits to the data.

```r
plot(ml1, which = 1)
```



```
plot(ml1, which = 1+1)
```



7.3 Log considerations

We also found that incoporating the log terms of explanatory variables such as `gdp`, `hex` and `phy` only makes these corresponding variables more insignificant and does nothing to improve R-squared value. We shall therefore not consider the logarithmic effects of variables.

```r
ml11 <- lm(df$lex ~ lngdp+df$lit+df$urb+df$unt+df$phy+df$hex
          +df$san+df$dri+df$fer+df$smo+df$alc+df$dev.f)
summary(ml11)
```

```
##
## Call:
## lm(formula = df$lex ~ lngdp + df$lit + df$urb + df$unt + df$phy +
##     df$hex + df$san + df$dri + df$fer + df$smo + df$alc + df$dev.f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3405  -1.7039   0.1411   2.2481   5.9909
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60.1151910  7.4661365   8.052 1.36e-13 ***
## lngdp        0.4015786  0.6313177   0.636 0.525569
## df$lit       0.0690026  0.0252993   2.727 0.007053 **
## df$urb      -0.0036889  0.0162273  -0.227 0.820441
## df$unt      -0.0413947  0.0245751  -1.684 0.093936 .
## df$phy       0.6389407  0.2604236   2.453 0.015158 *
## df$hex       0.0004907  0.0001445   3.397 0.000849 ***
## df$san       0.0455066  0.0222108   2.049 0.042014 *
## df$dri       0.0570005  0.0388592   1.467 0.144265
## df$fer      -1.6229231  0.3883146  -4.179 4.67e-05 ***
## df$smo       0.0267193  0.0266987   1.001 0.318361
## df$alc      -0.3116387  0.0718314  -4.338 2.45e-05 ***
## df$dev.f2   -2.1658698  0.9869434  -2.195 0.029554 *
## df$dev.f3   -2.3060295  1.5641298  -1.474 0.142245
## df$dev.f4   -0.8544718  2.3227570  -0.368 0.713428
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.128 on 170 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8521
## F-statistic: 76.72 on 14 and 170 DF,  p-value: < 2.2e-16
```

```r
lnhex <- log(df$hex)
lnphy <- log(df$phy)
ml12 <- lm(df$lex ~ lngdp+df$lit+df$urb+df$unt+lnphy+lnhex
          +df$san+df$dri+df$fer+df$smo+df$alc+df$dev.f)
summary(ml12)
```

```
##
## Call:
## lm(formula = df$lex ~ lngdp + df$lit + df$urb + df$unt + lnphy +
##     lnhex + df$san + df$dri + df$fer + df$smo + df$alc + df$dev.f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7259  -1.7671   0.1813   2.1799   7.1053
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 59.08574    7.21830   8.186 6.14e-14 ***
## lngdp       -0.26800    0.73142  -0.366 0.714520    
## df$lit       0.05724    0.02524   2.268 0.024588 *  
## df$urb      -0.01247    0.01627  -0.767 0.444411    
## df$unt      -0.03220    0.02454  -1.312 0.191324    
## lnphy        1.11023    0.33078   3.356 0.000974 ***
## lnhex        1.92295    0.55585   3.459 0.000684 ***
## df$san       0.02237    0.02266   0.987 0.325099    
## df$dri       0.05152    0.03851   1.338 0.182793    
## df$fer      -1.45559    0.38878  -3.744 0.000248 ***
## df$smo       0.02623    0.02606   1.007 0.315572    
## df$alc      -0.33579    0.07033  -4.775 3.87e-06 ***
## df$dev.f2   -1.86273    0.99208  -1.878 0.062150 .  
## df$dev.f3   -1.16710    1.54808  -0.754 0.451953    
## df$dev.f4    1.01686    2.24444   0.453 0.651087    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.099 on 170 degrees of freedom
## Multiple R-squared:  0.8659, Adjusted R-squared:  0.8548 
## F-statistic: 78.38 on 14 and 170 DF,  p-value: < 2.2e-16
```

8 Problems with the model

8.1 Heteroscedasticity

Heteroscedasticity is a result of non-constant variance of errors in the data. If there is absolutely no heteroscedastity, we should see a completely random, equal distribution of points throughout the range of X axis and a flat red line in the Residuals vs Fitted plot. This however is not observed and is suggestive of heteroscedasticity.

The Breush-Pagan test is used for the purpose of identifying heteroscedasticity. Our null hypothesis that the variance of the residuals is constant (homoscedasticity).

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
## 
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric
```

```
bptest(ml1)
```

```
## 
##  studentized Breusch-Pagan test
## 
## data:  ml1
## BP = 24.305, df = 14, p-value = 0.04208
```

8.2 Multicollinearity

Detection of Multicollinearity is done through two methods:

- Correlation Matrix
- Variance Inflation Factor (VIF)

8.3 Correlation Matrix

We first use the correlation matrix method to detect multicollinearity. X1 is the variable matrix on which we are testing for multicollinearity.

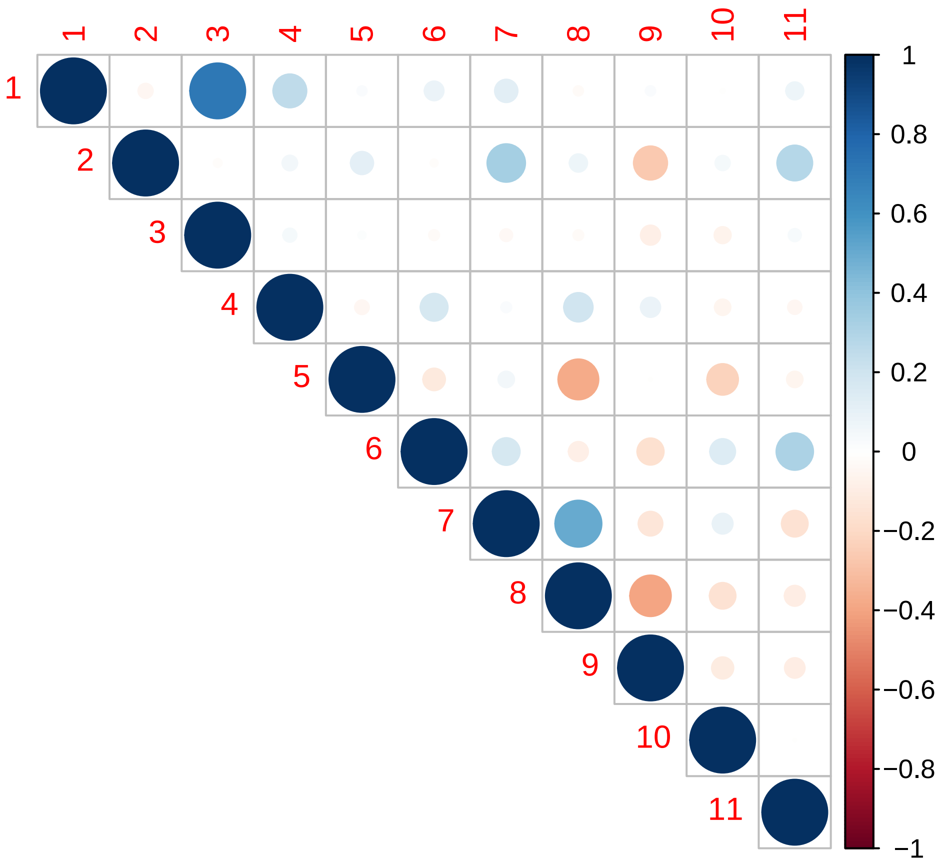
```
X1 <- df[,4:14]
```

Forming the correlation matrix amongst the variables,

```
library(corpcor)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corr1 <- cor2pcor(cov(X1))
corrplot(corr1, type="upper")
```



The correlation matrix shows a high correlation of 0.716 between `gdp` and `hex` and a substantial correlation between `gdp` and `urb` amongst correlation between other variables.

Overall checking for multicollinearity.

```
library(mctest)
omcdiag(X1,df$lex)
```

```
##
## Call:
```

```
## omcdiag(x = X1, y = df$lex)
##
##
## Overall Multicollinearity Diagnostics
##
##                         MC Results detection
## Determinant |X'X|:          0.0002         1
## Farrar Chi-Square:       1544.5986         1
## Red Indicator:              0.5178         1
## Sum of Lambda Inverse:     40.3917         0
## Theil's Method:            -1.6304         0
## Condition Number:          64.5737         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

As it can be seen, there is multicollinearity present in our dataset.

9 Variable Selection

```
library(MASS)
ml20 <- stepAIC(ml1,direction="both", trace=FALSE)
summary(ml20)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6857  -1.7775   0.2668   2.3192   6.3004
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 68.9908660  2.9883998  23.086  < 2e-16 ***
## df$lit       0.0690648  0.0252136   2.739 0.006801 **
## df$hex       0.0005208  0.0001259   4.136 5.48e-05 ***
## df$unt      -0.0558640  0.0229177  -2.438 0.015791 *
## df$phy       0.6709269  0.2511826   2.671 0.008278 **
## df$san       0.0622209  0.0194507   3.199 0.001639 **
## df$fer      -1.8448306  0.3572907  -5.163 6.58e-07 ***
## df$alc      -0.3194521  0.0708164  -4.511 1.18e-05 ***
## df$dev.f2   -2.5870380  0.7466704  -3.465 0.000668 ***
## df$dev.f3   -3.1693819  1.0101872  -3.137 0.002002 **
## df$dev.f4   -2.4761598  1.4169137  -1.748 0.082302 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.118 on 174 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.853
## F-statistic: 107.8 on 10 and 174 DF,  p-value: < 2.2e-16
```

10 Multiple Linear Regression - Refined Model

$$lex = \beta_0 + \beta_1(lit) + \beta_2(hex) + \beta_3(unt) + \beta_4(phy)$$
$$+\beta_5(san) + \beta_6(fer) + \beta_7(alc) + \beta_8(dev)$$

```
ml2 <- lm(df$lex ~ df$lit+df$hex+df$unt+df$phy
          +df$san+df$fer+df$alc+df$dev.f)
summary(ml2)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6857  -1.7775   0.2668   2.3192   6.3004
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 68.9908660  2.9883998  23.086  < 2e-16 ***
## df$lit       0.0690648  0.0252136   2.739 0.006801 **
## df$hex       0.0005208  0.0001259   4.136 5.48e-05 ***
## df$unt      -0.0558640  0.0229177  -2.438 0.015791 *
## df$phy       0.6709269  0.2511826   2.671 0.008278 **
## df$san       0.0622209  0.0194507   3.199 0.001639 **
## df$fer      -1.8448306  0.3572907  -5.163 6.58e-07 ***
## df$alc      -0.3194521  0.0708164  -4.511 1.18e-05 ***
## df$dev.f2   -2.5870380  0.7466704  -3.465 0.000668 ***
## df$dev.f3   -3.1693819  1.0101872  -3.137 0.002002 **
## df$dev.f4   -2.4761598  1.4169137  -1.748 0.082302 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.118 on 174 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.853
## F-statistic: 107.8 on 10 and 174 DF,  p-value: < 2.2e-16
```

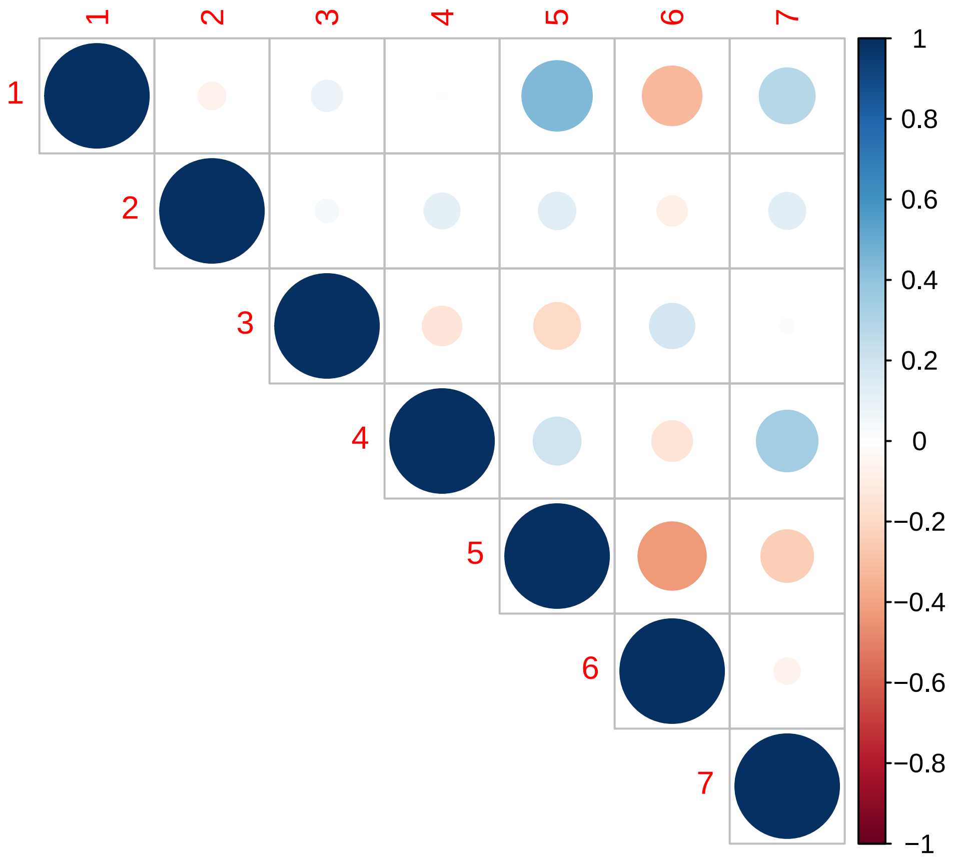
10.1 Interactive Terms

We used the correlation matrix to see if creating interactive variables between highly correlated variables make our model any better. There were five cases of somewhat high correlation (>0.3).

```
X2 <- df[c(5,6,8,9,10,12,14)]
```

Forming the correlation matrix amongst the variables,

```
library(corpcor)
library(corrplot)
corr2 <- cor2pcor(cov(X2))
corrplot(corr2, type="upper")
```



10.2 Interaction between lit and san

```
ml3 <- lm(df$lex ~ df$lit+df$hex+df$unt+df$phy +df$san+df$fer+df$alc+df$dev.f+(df$lit*df$san))
summary(ml3)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f + (df$lit * df$san))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8018  -1.5591   0.1567   2.1560   6.2102
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  65.4544380  3.5890547  18.237  < 2e-16 ***
## df$lit        0.1222333  0.0393079   3.110 0.002191 **
## df$hex        0.0005378  0.0001255   4.284 3.04e-05 ***
## df$unt       -0.0642610  0.0232782  -2.761 0.006393 **
## df$phy        0.7655581  0.2554425   2.997 0.003128 **
## df$san        0.1545206  0.0560095   2.759 0.006425 **
## df$fer       -1.8274525  0.3553089  -5.143 7.26e-07 ***
## df$alc       -0.3107443  0.0705708  -4.403 1.86e-05 ***
## df$dev.f2    -2.7916740  0.7513346  -3.716 0.000273 ***
## df$dev.f3    -3.7315665  1.0540006  -3.540 0.000513 ***
## df$dev.f4     -2.3086385  1.4117351  -1.635 0.103800
## df$lit:df$san -0.0011575  0.0006592  -1.756 0.080880 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.1 on 173 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8548
## F-statistic: 99.44 on 11 and 173 DF,  p-value: < 2.2e-16
```

The interactive variable between `lit` and `san` turns out to be significant only at the 10% level. Also, this model only minutely improves the R-squared value as compared to our model. Hence, due to the statistical insignificance of the interactive variable at 5%, we reject the model.

10.3 Interaction between `lit` and `fer`

```
ml4 <- lm(df$lex ~ df$lit+df$hex+df$unt+df$phy +df$san+df$fer+df$alc+df$dev.f+(df$lit*df$fer))
summary(ml4)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f + (df$lit * df$fer))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7245  -1.5288   0.3018   2.2816   6.2994
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   71.6846358  5.3808345  13.322  < 2e-16 ***
## df$lit         0.0342772  0.0630228   0.544 0.587220
## df$hex         0.0005281  0.0001267   4.168 4.85e-05 ***
## df$unt        -0.0582517  0.0232992  -2.500 0.013345 *
## df$phy         0.7078204  0.2589868   2.733 0.006927 **
## df$san         0.0642599  0.0197781   3.249 0.001391 **
## df$fer        -2.4199903  1.0195306  -2.374 0.018711 *
## df$alc        -0.3089454  0.0730582  -4.229 3.80e-05 ***
## df$dev.f2     -2.6406980  0.7533242  -3.505 0.000581 ***
## df$dev.f3     -3.3112314  1.0390661  -3.187 0.001708 **
## df$dev.f4     -2.4493341  1.4202125  -1.725 0.086381 .
## df$lit:df$fer  0.0076775  0.0127429   0.602 0.547633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.124 on 173 degrees of freedom
## Multiple R-squared:  0.8613, Adjusted R-squared:  0.8525
## F-statistic: 97.66 on 11 and 173 DF,  p-value: < 2.2e-16
```

Inference: The interactive variable between `lit` and `fer` turns out to be statistically insignificant and also makes `lit` as statistically insignifant . Also, this model has no change on the R-squared value as compared to our model. Hence, due to the statistical insignificance of the interactive variable at 5%, we reject this model.

10.4 Interaction between `lit` and `alc`

```r
ml5 <- lm(df$lex ~ df$lit+df$hex+df$unt+df$phy +df$san+df$fer+df$alc+df$dev.f+(df$lit*df$alc))
summary(ml5)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f + (df$lit * df$alc))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.447  -1.731   0.289   2.224   6.474
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)   70.3796596  3.0421263  23.135  < 2e-16 ***
## df$lit         0.0517370  0.0264409   1.957 0.051992 .
## df$hex         0.0005171  0.0001248   4.142 5.37e-05 ***
## df$unt        -0.0532164  0.0227582  -2.338 0.020513 *
## df$phy         0.5419258  0.2571535   2.107 0.036524 *
## df$san         0.0606825  0.0192981   3.144 0.001959 **
## df$fer        -1.7834090  0.3555250  -5.016 1.30e-06 ***
## df$alc        -1.0210580  0.3560357  -2.868 0.004647 **
## df$dev.f2     -2.4997619  0.7415035  -3.371 0.000923 ***
## df$dev.f3     -3.0985777  1.0020948  -3.092 0.002317 **
## df$dev.f4     -2.3110816  1.4070931  -1.642 0.102313
## df$lit:df$alc  0.0079671  0.0039636   2.010 0.045977 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.092 on 173 degrees of freedom
## Multiple R-squared:  0.8642, Adjusted R-squared:  0.8555
## F-statistic: 100.1 on 11 and 173 DF,  p-value: < 2.2e-16
```

Inference: The interactive variable between `lit` and `alc` turns out to be statistically significant although it also makes `lit` statistically signifant only at 10% level of significance. Also, this model has a slighly higher R square as compared to our model. Hence, this interactive variable is a valid determinant for life expectancy.

10.5 Interaction between `san` and `fer`

```r
ml6<- lm(df$lex ~ df$lit+df$hex+df$unt+df$phy +df$san+df$fer+df$alc+df$dev.f+(df$san*df$fer))
summary(ml6)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f + (df$san * df$fer))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4351  -1.9116   0.1788   2.1716   6.1713
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)   65.8632261  4.2388470  15.538  < 2e-16 ***
## df$lit         0.0736189  0.0255851   2.877  0.00451 **
## df$hex         0.0005069  0.0001266   4.005  9.2e-05 ***
## df$unt        -0.0539483  0.0229862  -2.347  0.02006 *
## df$phy         0.6137561  0.2570682   2.388  0.01804 *
## df$san         0.0996181  0.0408756   2.437  0.01582 *
## df$fer        -1.2238484  0.6957147  -1.759  0.08032 .
## df$alc        -0.3297869  0.0714936  -4.613  7.7e-06 ***
## df$dev.f2     -2.3669397  0.7759058  -3.051  0.00264 **
## df$dev.f3     -2.7496044  1.0875981  -2.528  0.01236 *
## df$dev.f4     -2.3549161  1.4213679  -1.657  0.09937 .
## df$san:df$fer -0.0103947  0.0099934  -1.040  0.29972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.118 on 173 degrees of freedom
## Multiple R-squared:  0.8619, Adjusted R-squared:  0.8531
## F-statistic: 98.13 on 11 and 173 DF,  p-value: < 2.2e-16
```

Inference: The interactive variable between `san` and `fer` turns out to be statistically insignificant. Also, this model has no change on the R square as compared to our model. Hence, due to the statistical insignificance of the interactive variable at 5%, we reject the model.

10.6 Interaction between `phy` and `alc`

```
ml7 <- lm(df$lex ~df$lit+df$hex+df$unt+df$phy
          +df$san+df$fer+df$alc+df$dev.f+(df$phy*df$alc))
summary(ml7)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f + (df$phy * df$alc))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.534  -1.850   0.202   2.449   6.188
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   68.8252447  2.9778351  23.113  < 2e-16 ***
## df$lit         0.0789013  0.0258808   3.049  0.00266 **
## df$hex         0.0005069  0.0001257   4.033 8.25e-05 ***
## df$unt        -0.0530730  0.0228916  -2.318  0.02159 *
## df$phy         0.0743293  0.4554043   0.163  0.87054
## df$san         0.0627883  0.0193731   3.241  0.00143 **
## df$fer        -1.8021856  0.3568418  -5.050 1.11e-06 ***
## df$alc        -0.4602385  0.1141851  -4.031 8.32e-05 ***
## df$dev.f2     -2.4832304  0.7465050  -3.326  0.00107 **
## df$dev.f3     -3.2624865  1.0077339  -3.237  0.00145 **
## df$dev.f4     -2.4797613  1.4110176  -1.757  0.08061 .
## df$phy:df$alc  0.0752317  0.0479889   1.568  0.11878
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.105 on 173 degrees of freedom
## Multiple R-squared:  0.8629, Adjusted R-squared:  0.8542
## F-statistic: 99.03 on 11 and 173 DF,  p-value: < 2.2e-16
```

Inference: The interactive variable between `phy` and `alc` turns out to be statistically insignificant. Also, this model has no change on the R square as compared to our model. Hence, due to the statistical insignificance of the interactive variable at 5%, we reject the model.

11 The Final Model and tests

We consider the model ml5 to be our correct model.

11.1 Heteroscedasticity

```
library(lmtest)
bptest(ml5)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ml5
## BP = 18.414, df = 11, p-value = 0.07246
```

Output: p-value = 0.07246 > 0.05, hence, we reject the null hypothesis and there is no problem of heteroscedasticity in the model.

11.2 Variable matrix to perform correlation

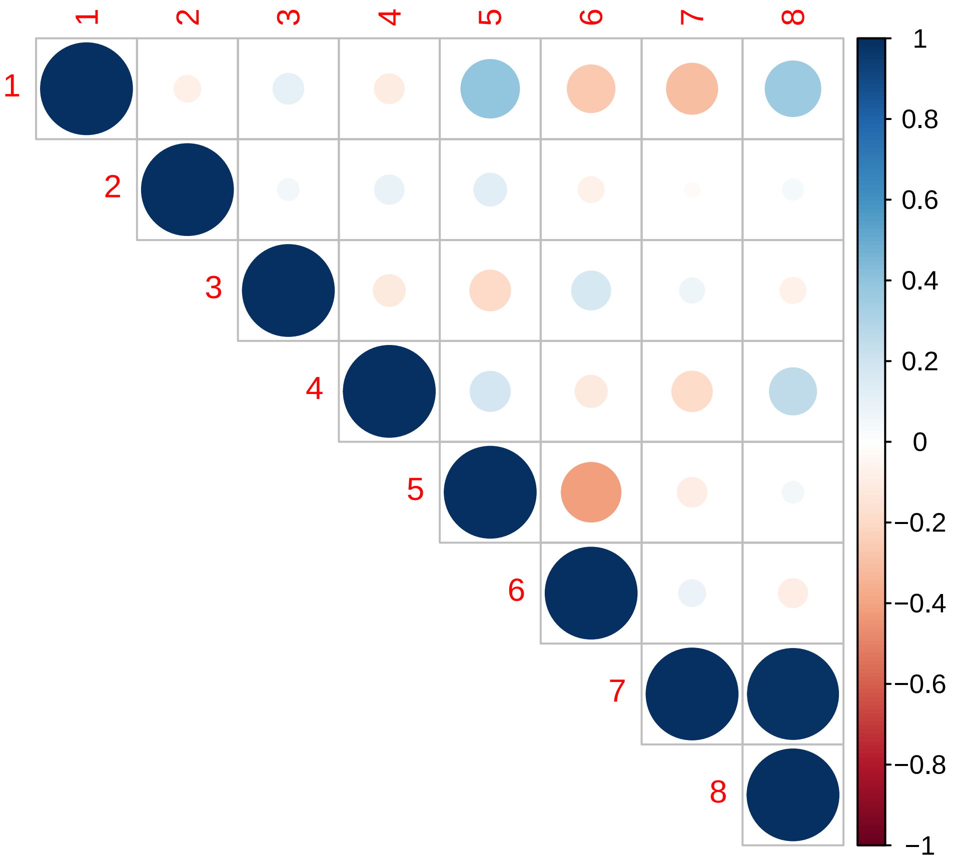
```
al<-cbind(X2,df$alc*df$lit)
head(al)
```

```
##        lit        hex  unt      phy      san   fer  alc df$alc * df$lit
## 1 97.80742 2500.00000 26.9 0.390625 97.54348 1.834 5.90      577.063778
## 2 43.01972   60.11276 26.9 0.303900 39.37493 5.163 0.20        8.603944
## 3 66.03011  131.75187 28.1 0.214900 46.11703 5.864 7.70      508.431847
## 4 97.24697  313.26290  5.9 1.270600 97.67685 1.688 7.70      748.801669
## 5 92.80000 1613.37514  3.7 2.026800 98.57367 1.595 3.45      320.160000
## 6 98.99389 1086.72728  3.4 3.906600 94.20105 2.312 9.55      945.391649
```

11.3 Correlation Matrix - Visualization

Visualising the correlation matrix (Alternative to Heat map) – Package: corrplot

```
library(corpcor)
library(corrplot)
corr3<-cor2pcor(cov(al))
corrplot(corr3, type="upper")
```



Inference: alc and alc*lit share a high positive correlation (0.98), which is only logical. High correlation is a clear sign of multicollinearity. It is a sufficient but not the necessary condition for the multicollinearity.

11.4 Multicollinearity Test

```
library(mctest)
omcdiag(al,df$lex)
```

```
##
## Call:
## omcdiag(x = al, y = df$lex)
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.0003         1
## Farrar Chi-Square:      1495.4373         1
## Red Indicator:             0.5519         1
## Sum of Lambda Inverse:   112.4235         1
## Theil's Method:           -0.4534         0
## Condition Number:         46.3381         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

VIF Test

```r
library(car)
```

```
## Loading required package: carData
```

```r
vif(ml5)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## df$lit         5.061553  1        2.249789
## df$hex         1.708695  1        1.307170
## df$unt         1.867991  1        1.366745
## df$phy         2.695296  1        1.641736
## df$san         6.198850  1        2.489749
## df$fer         4.681422  1        2.163659
## df$alc        41.352806  1        6.430615
## df$dev.f       7.807872  3        1.408495
## df$lit:df$alc 52.118517  1        7.219316
```

The VIFs for `alc` and `alclit` are extremely high. Clearly, this model although has a better R square, it suffers from a case of Multicollinearity due to the presence of both the variables `alc` and `alc*lit`.

11.5 Variable Selection

```r
library(MASS)
ml8 <- stepAIC(ml5,direction="both", trace=FALSE)
summary(ml8)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$alc + df$dev.f + (df$lit * df$alc))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.447  -1.731   0.289   2.224   6.474
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   70.3796596  3.0421263  23.135  < 2e-16 ***
## df$lit         0.0517370  0.0264409   1.957 0.051992 .
## df$hex         0.0005171  0.0001248   4.142 5.37e-05 ***
## df$unt        -0.0532164  0.0227582  -2.338 0.020513 *
## df$phy         0.5419258  0.2571535   2.107 0.036524 *
## df$san         0.0606825  0.0192981   3.144 0.001959 **
## df$fer        -1.7834090  0.3555250  -5.016 1.30e-06 ***
## df$alc        -1.0210580  0.3560357  -2.868 0.004647 **
## df$dev.f2     -2.4997619  0.7415035  -3.371 0.000923 ***
## df$dev.f3     -3.0985777  1.0020948  -3.092 0.002317 **
## df$dev.f4     -2.3110816  1.4070931  -1.642 0.102313
## df$lit:df$alc  0.0079671  0.0039636   2.010 0.045977 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.092 on 173 degrees of freedom
## Multiple R-squared:  0.8642, Adjusted R-squared:  0.8555
## F-statistic: 100.1 on 11 and 173 DF,  p-value: < 2.2e-16
```

Therefore we remove alcohol to avoid multicollinearity

```
ml9<- lm(df$lex ~ df$lit+df$hex+df$unt+df$phy+df$san
         +df$fer+df$dev.f+(df$lit*df$alc))
summary(ml9)
```

```
##
## Call:
## lm(formula = df$lex ~ df$lit + df$hex + df$unt + df$phy + df$san +
##     df$fer + df$dev.f + (df$lit * df$alc))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.447  -1.731   0.289   2.224   6.474
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   70.3796596  3.0421263  23.135  < 2e-16 ***
## df$lit         0.0517370  0.0264409   1.957 0.051992 .
## df$hex         0.0005171  0.0001248   4.142 5.37e-05 ***
## df$unt        -0.0532164  0.0227582  -2.338 0.020513 *
## df$phy         0.5419258  0.2571535   2.107 0.036524 *
## df$san         0.0606825  0.0192981   3.144 0.001959 **
## df$fer        -1.7834090  0.3555250  -5.016 1.30e-06 ***
## df$dev.f2     -2.4997619  0.7415035  -3.371 0.000923 ***
## df$dev.f3     -3.0985777  1.0020948  -3.092 0.002317 **
## df$dev.f4     -2.3110816  1.4070931  -1.642 0.102313
## df$alc        -1.0210580  0.3560357  -2.868 0.004647 **
## df$lit:df$alc  0.0079671  0.0039636   2.010 0.045977 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.092 on 173 degrees of freedom
## Multiple R-squared:  0.8642, Adjusted R-squared:  0.8555
## F-statistic: 100.1 on 11 and 173 DF,  p-value: < 2.2e-16
```

Thus, we drop the interactive variable `alc*lit` and return to our initial model `ml2`. The adjusted R square for the model drops from 0.855(`ml5`) to 0.853 (`ml2`). This can be done without much compromise to the regression fit, since the presence of collinearity implies that the information that this variable provides about the response is redundant in the presence of the other variables.

Our final model is given by

`lex` = 68.99 + 0.069`lit`+ 0.001`hex`- 0.056`unt`+ 0.671`phy`+ 0.062`san` -1.845`fer` -0.319`alc` - 2.58`uppermiddle`- 3.169 `lowermiddle` - 2.476 `lowincome`

12 Conclusions and Suggestions

Statistically, the most important determinants of the average life expectancy of a country are fertility rate, alcohol consumption per capita, health expenditure and the development status of the country. Other important determinants ordered according to their p-values are percentage of people using at least basic sanitation services, literacy rates, number of physicians per 1000 people and prevalance of undernourishment in the country. We therefore see that health and educations factors play the key role in determining life expectancy.

Achieve universal health coverage
One can avoid premature deaths can be averted by improving access to and use of preventive and curative health services, particularly in low-income countries. This requires a strengthened health workforce and increased provision of health facilities, equipment, medicines and vaccines. It will also require removing barriers to accessing services including economic and cultural barriers.

Achieve universal access to education
One can achieve inclusive growth and progressive societal transformation by imparting education and relevant skills to all sections of the population. This requires strengthening the workforce of schools and other academic institutions.

13 Acknowledgements

The authors wish to acknowledge Prof. Sowmya Dhanaraj for her valuable insights and guidance.

References

[1] Shaw, Horrace et al. *The Determinants of Life Expectancy: An Analysis of the OECD Health Data*, Southern Economic Journal, 2005, `https://www.jstor.org/stable/pdf/20062079`

[2] Mahfuz Kabir. *Determinants of life expectancy in developing countries*, The Journal of Developing areas, 2008, `https://www.jstor.org/stable/pdf/40376184.pdf`

[3] Preston, S *"The Changing Relation between Mortality and Level of Economic Development"*, Population Studies, 1975, 29 (2): 231–248. doi:10.2307/2173509

[4] T. Paul Schultz . *Handbook of development economics.* 2005, Elsevier. p. 3406. ISBN 978-0-444-53100-1.

[5] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R.* New York :Springer, 2013