Test Project IS 517

Abhinav Choudhry

May 2, 2021

So it begins…

```
library(ISLR)
econtrain<-read.csv("C:\\Users\\TCI\\Documents\\training_economic_full.csv")
econtest<-read.csv("C:\\Users\\TCI\\Documents\\test_economic_full.csv")
trainfull<-read.csv("C:\\Users\\TCI\\Documents\\training_small_cleaner_with_tobacco_BMI_TechFuel.csv")
testfull<-read.csv("C:\\Users\\TCI\\Documents\\test_small_cleaner_with_tobacco_BMI_TechFuel.csv")

summary(testfull)
```

```
##                 Location        NTDs             Poisoning
##  Afghanistan        :  1   Min.   :        0   Min.   :0.041
##  Albania            :  1   1st Qu.:       56   1st Qu.:0.300
##  Algeria            :  1   Median :    59781   Median :0.570
##  Andorra            :  1   Mean   :  8630647   Mean   :1.161
##  Angola             :  1   3rd Qu.:  3634160   3rd Qu.:1.880
##  Antigua and Barbuda:  1   Max.   :631000000   Max.   :5.190
##  (Other)            :188                       NA's   :11
##   Tuberculosis   Under5Mortality  DrinkingWater      Alcohol
##  Min.   :  0.0   Min.   :  1.90   Min.   : 22.83   Min.   : 0.003
##  1st Qu.: 12.0   1st Qu.:  7.33   1st Qu.: 68.26   1st Qu.: 2.288
##  Median : 47.0   Median : 16.42   Median : 91.17   Median : 5.720
##  Mean   :117.3   Mean   : 29.96   Mean   : 81.59   Mean   : 6.026
##  3rd Qu.:173.8   3rd Qu.: 45.17   3rd Qu.: 99.24   3rd Qu.: 9.322
##  Max.   :805.0   Max.   :128.40   Max.   :100.00   Max.   :20.500
##                  NA's   :22       NA's   :2        NA's   :6
##  BasicSanitization   CancerEtc          HALE       InfantMortality
##  Min.   :  7.09    Min.   : 7.80   Min.   :44.24   Min.   : 1.69
##  1st Qu.: 54.51    1st Qu.:14.85   1st Qu.:57.99   1st Qu.: 6.30
##  Median : 88.42    Median :18.80   Median :64.54   Median :14.54
##  Mean   : 74.81    Mean   :18.88   Mean   :63.31   Mean   :22.43
##  3rd Qu.: 98.09    3rd Qu.:22.90   3rd Qu.:67.72   3rd Qu.:33.09
##  Max.   :100.00    Max.   :30.60   Max.   :74.09   Max.   :88.57
##  NA's   :3         NA's   :11      NA's   :11      NA's   :22
##  LifeExpectancy  MaternalMortality NeonatalMortality    Tobacco
##  Min.   :50.75   Min.   :   2.0    Min.   : 0.88     Min.   : 3.90
##  1st Qu.:66.55   1st Qu.:  12.5    1st Qu.: 4.10     1st Qu.:14.60
##  Median :73.74   Median :  55.0    Median :10.31     Median :23.20
##  Mean   :72.54   Mean   : 164.8    Mean   :13.43     Mean   :22.92
##  3rd Qu.:77.73   3rd Qu.: 203.5    3rd Qu.:21.56     3rd Qu.:28.70
##  Max.   :84.26   Max.   :1140.0    Max.   :44.22     Max.   :53.90
##  NA's   :11      NA's   :11        NA's   :1         NA's   :45
##    BMI30Plus     CleanTechFuel
##  Min.   : 2.10   Min.   : 5.00
##  1st Qu.: 9.60   1st Qu.:27.25
##  Median :20.60   Median :84.50
##  Mean   :20.01   Mean   :63.67
##  3rd Qu.:25.70   3rd Qu.:95.00
##  Max.   :61.00   Max.   :95.00
##  NA’s   :5       NA’s   :4
```

```
#Creating Tobacco excluding sets for some models
trainmain<-trainfull
trainmain$Tobacco<-NULL
testmain<-testfull
testmain$Tobacco<-NULL

summary(trainfull)
```

```
##                 Location        NTDs             Poisoning
##  Afghanistan        :  1   Min.   :        0   Min.   :0.046
##  Albania            :  1   1st Qu.:       40   1st Qu.:0.310
##  Algeria            :  1   Median :    41453   Median :0.580
##  Andorra            :  1   Mean   :  9049353   Mean   :1.196
##  Angola             :  1   3rd Qu.:  3838014   3rd Qu.:1.935
##  Antigua and Barbuda:  1   Max.   :668000000   Max.   :5.300
##  (Other)            :188                       NA’s   :11
##     Tobacco       Tuberculosis    Under5Mortality  DrinkingWater
##  Min.   : 4.00   Min.   :  0.00   Min.   :  1.99   Min.   : 22.34
##  1st Qu.:15.00   1st Qu.: 11.25   1st Qu.:  7.43   1st Qu.: 66.61
##  Median :23.50   Median : 49.50   Median : 16.89   Median : 90.52
##  Mean   :23.31   Mean   :116.49   Mean   : 31.01   Mean   : 81.10
##  3rd Qu.:29.00   3rd Qu.:162.25   3rd Qu.: 46.69   3rd Qu.: 98.89
##  Max.   :54.70   Max.   :988.00   Max.   :138.30   Max.   :100.00
##  NA’s   :45                       NA’s   :22       NA’s   :1
##     Alcohol       BasicSanitization   CancerEtc         HALE
##  Min.   : 0.003   Min.   :  6.86    Min.   : 8.3   Min.   :41.64
##  1st Qu.: 2.178   1st Qu.: 53.39    1st Qu.:15.0   1st Qu.:57.09
##  Median : 5.865   Median : 88.29    Median :19.1   Median :63.85
##  Mean   : 6.102   Mean   : 74.52    Mean   :19.1   Mean   :62.50
##  3rd Qu.: 9.580   3rd Qu.: 97.93    3rd Qu.:23.0   3rd Qu.:67.56
##  Max.   :18.350   Max.   :100.00    Max.   :30.8   Max.   :73.62
##  NA’s   :6        NA’s   :1         NA’s   :11     NA’s   :11
##  InfantMortality  LifeExpectancy  MaternalMortality NeonatalMortality
##  Min.   : 1.760   Min.   :47.67   Min.   :   2.0    Min.   : 0.92
##  1st Qu.: 6.388   1st Qu.:64.83   1st Qu.:  13.5    1st Qu.: 4.20
##  Median :14.525   Median :72.70   Median :  58.0    Median :10.58
##  Mean   :23.073   Mean   :71.50   Mean   : 169.5    Mean   :13.73
##  3rd Qu.:34.227   3rd Qu.:77.11   3rd Qu.: 224.5    3rd Qu.:22.10
##  Max.   :94.170   Max.   :83.62   Max.   :1180.0    Max.   :45.22
##  NA’s   :22       NA’s   :11      NA’s   :11        NA’s   :1
##    BMI30Plus     CleanTechFuel
##  Min.   : 2.00   Min.   : 5.00
##  1st Qu.: 9.20   1st Qu.:26.25
##  Median :20.20   Median :84.00
##  Mean   :19.55   Mean   :63.36
##  3rd Qu.:25.20   3rd Qu.:95.00
##  Max.   :60.70   Max.   :95.00
##  NA’s   :5       NA’s   :4
```

```
#fix(testfull)
#Creating data sets devoid of NAs

trainfullNA<-na.omit(trainfull)
row.names(trainfullNA)=1:nrow(trainfullNA)
dim(trainfull)
```

```
## [1] 194  17
```

```
dim(trainfullNA)
```

```
## [1] 124  17
```

```
summary(trainfullNA)
```

```
##       Location        NTDs             Poisoning         Tobacco
##  Albania   :  1   Min.   :        0   Min.   :0.0830   Min.   : 4.00
##  Algeria   :  1   1st Qu.:       24   1st Qu.:0.2575   1st Qu.:14.90
##  Argentina:  1   Median :    33274   Median :0.4900   Median :23.00
##  Armenia   :  1   Mean   : 11943426   Mean   :1.0827   Mean   :22.77
##  Australia:  1   3rd Qu.:  4151157   3rd Qu.:1.6425   3rd Qu.:28.93
##  Austria   :  1   Max.   :668000000   Max.   :5.3000   Max.   :54.20
##  (Other)   :118
##   Tuberculosis    Under5Mortality   DrinkingWater       Alcohol
##  Min.   :  0.00   Min.   :  2.230   Min.   : 27.80   Min.   : 0.003
##  1st Qu.: 10.75   1st Qu.:  5.082   1st Qu.: 70.80   1st Qu.: 2.750
##  Median : 51.00   Median : 15.455   Median : 90.22   Median : 6.800
##  Mean   :120.81   Mean   : 28.197   Mean   : 83.22   Mean   : 6.618
##  3rd Qu.:157.00   3rd Qu.: 41.998   3rd Qu.: 99.62   3rd Qu.:10.127
##  Max.   :988.00   Max.   :138.300   Max.   :100.00   Max.   :18.350
##
##  BasicSanitization   CancerEtc          HALE       InfantMortality
##  Min.   :  6.86    Min.   : 8.30    Min.   :41.64    Min.   : 1.760
##  1st Qu.: 55.39    1st Qu.:13.03    1st Qu.:58.73    1st Qu.: 4.258
##  Median : 90.89    Median :18.15    Median :64.50    Median :13.085
##  Mean   : 76.10    Mean   :18.40    Mean   :63.38    Mean   :20.940
##  3rd Qu.: 98.77    3rd Qu.:22.85    3rd Qu.:68.89    3rd Qu.:32.417
##  Max.   :100.00    Max.   :30.80    Max.   :73.62    Max.   :94.170
##
##  LifeExpectancy  MaternalMortality NeonatalMortality   BMI30Plus
##  Min.   :47.67   Min.   :   2.00   Min.   : 0.920    Min.   : 3.40
##  1st Qu.:67.17   1st Qu.:   9.75   1st Qu.: 3.105    1st Qu.: 8.50
##  Median :73.98   Median :  38.50   Median : 8.745    Median :19.95
##  Mean   :72.51   Mean   : 147.23   Mean   :12.314    Mean   :18.05
##  3rd Qu.:78.81   3rd Qu.: 181.50   3rd Qu.:20.593    3rd Qu.:24.10
##  Max.   :83.62   Max.   :1180.00   Max.   :45.220    Max.   :46.70
##
##  CleanTechFuel
##  Min.   : 5.00
##  1st Qu.:26.00
##  Median :88.00
##  Mean   :64.45
##  3rd Qu.:95.00
##  Max.   :95.00
##
```

```r
trainmainNA<-na.omit(trainmain)
row.names(trainmainNA)=1:nrow(trainmainNA)

testfullNA<-na.omit(testfull)
row.names(testfullNA)=1:nrow(testfullNA)
testmainNA<-na.omit(testmain)
row.names(testmainNA)=1:nrow(testmainNA)
econtrainNA<-na.omit(econtrain)
row.names(econtrainNA)=1:nrow(econtrainNA)
dim(testfull)
```

```
## [1] 194  17
```

```r
dim(testfullNA)
```

```
## [1] 123  17
```

```r
dim(econtrain)
```

```
## [1] 195   6
```

```r
dim(econtrainNA)
```

```
## [1] 174   6
```

```r
summary(econtrainNA)
```

```
##                  Location    GDP_currentUSD      HealthExpGDPperc
##  Afghanistan        :  1   Min.   :1.711e+08   Min.   : 1.819
##  Albania            :  1   1st Qu.:1.114e+10   1st Qu.: 4.593
##  Algeria            :  1   Median :3.912e+10   Median : 6.219
##  Angola             :  1   Mean   :4.224e+11   Mean   : 6.526
##  Antigua and Barbuda:  1   3rd Qu.:1.965e+11   3rd Qu.: 8.010
##  Argentina          :  1   Max.   :1.820e+13   Max.   :20.413
##  (Other)            :168
##       HALE       LifeExpectancy  NeonatalMortality
##  Min.   :41.64   Min.   :47.67   Min.   : 0.92
##  1st Qu.:57.59   1st Qu.:65.81   1st Qu.: 4.20
##  Median :63.89   Median :73.03   Median :10.19
##  Mean   :62.72   Mean   :71.75   Mean   :13.67
##  3rd Qu.:67.70   3rd Qu.:77.38   3rd Qu.:22.29
##  Max.   :73.62   Max.   :83.62   Max.   :45.22
##
```

```
econtestNA<-na.omit(econtest)
row.names(econtestNA)=1:nrow(econtestNA)
dim(econtest)
```

```
## [1] 194   6
```

```
dim(econtestNA)
```

```
## [1] 173   6
```

##Linear regression on Economic Data

###For HALE

```
library(MLmetrics) #for MAPE-Mean Absolute Percentage Regression Loss
```

```
## Warning: package 'MLmetrics' was built under R version 3.6.3
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
# set.seed(26)
# Both GDP and percentage of GDP spent on health:all economic indicators
linregecon<-lm(HALE~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA)
summary(linregecon)
```

```
##
## Call:
## lm(formula = HALE ~ GDP_currentUSD + HealthExpGDPperc, data = econtrainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.875  -3.915   1.846   4.944  11.732
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.919e+01  1.299e+00  45.582  < 2e-16 ***
## GDP_currentUSD   4.851e-13  3.016e-13   1.608  0.10965
## HealthExpGDPperc 5.092e-01  1.895e-01   2.687  0.00793 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.408 on 171 degrees of freedom
## Multiple R-squared:  0.07417,    Adjusted R-squared:  0.06334
## F-statistic: 6.849 on 2 and 171 DF,  p-value: 0.001376
```

```
print("Training error of Economic indicators on Healthy Adjusted Life Expectancy are:")
```

```
## [1] "Training error of Economic indicators on Healthy Adjusted Life Expectancy are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtrainNA)
MAPE(econtrainNA$HALE, predictlinregecon)
```

```
## [1] 0.08168859
```

```
print("Test error of Economic indicators on Healthy Adjusted Life Expectancy are:")
```

```
## [1] "Test error of Economic indicators on Healthy Adjusted Life Expectancy are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtestNA)
MAPE(econtestNA$HALE, predictlinregecon)
```

```
## [1] 0.07841312
```

```
# Just GDP
linregecon<-lm(HALE~GDP_currentUSD,data=econtrainNA)
summary(linregecon)
```

```
##
## Call:
## lm(formula = HALE ~ GDP_currentUSD, data = econtrainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.770  -5.039   1.303   5.111  10.566
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.241e+01  5.097e-01 122.452   <2e-16 ***
## GDP_currentUSD 7.316e-13  2.925e-13   2.501   0.0133 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6.522 on 172 degrees of freedom
## Multiple R-squared:  0.03509,    Adjusted R-squared:  0.02948
## F-statistic: 6.255 on 1 and 172 DF,  p-value: 0.01332
```

```
print("Training error of just GDP on Healthy Adjusted Life Expectancy are:")
```

```
## [1] "Training error of just GDP on Healthy Adjusted Life Expectancy are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtrainNA)
MAPE(econtrainNA$HALE, predictlinregecon)
```

```
## [1] 0.08622197
```

```r
print("Test error of just GDP on Healthy Adjusted Life Expectancy are:")
```

```
## [1] "Test error of just GDP on Healthy Adjusted Life Expectancy are:"
```

```r
predictlinregecon = predict(linregecon, newdata = econtestNA)
MAPE(econtestNA$HALE, predictlinregecon)
```

```
## [1] 0.08270131
```

```r
# Just Health expenditure as GDP percentage
linregecon<-lm(HALE~HealthExpGDPperc,data=econtrainNA)
summary(linregecon)
```

```
##
## Call:
## lm(formula = HALE ~ HealthExpGDPperc, data = econtrainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.261  -3.860   1.774   4.923  11.894
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       58.7899     1.2803  45.917  < 2e-16 ***
## HealthExpGDPperc   0.6019     0.1814   3.318  0.00111 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.437 on 172 degrees of freedom
## Multiple R-squared:  0.06017,    Adjusted R-squared:  0.0547
## F-statistic: 11.01 on 1 and 172 DF,  p-value: 0.001105
```

```r
print("Training error of just Health Expenditure on Healthy Adjusted Life Expectancy are:")
```

```
## [1] "Training error of just Health Expenditure on Healthy Adjusted Life Expectancy are:"
```

```r
predictlinregecon = predict(linregecon, newdata = econtrainNA)
MAPE(econtrainNA$HALE, predictlinregecon)
```

```
## [1] 0.0815868
```

```r
print("Test error of just Health Expenditure on Healthy Adjusted Life Expectancy are:")
```

```
## [1] "Test error of just Health Expenditure on Healthy Adjusted Life Expectancy are:"
```

```r
predictlinregecon = predict(linregecon, newdata = econtestNA)
MAPE(econtestNA$HALE, predictlinregecon)
```

```
## [1] 0.07826581
```

```
# #
```

Health expenditure as percentage of GDP is better than GDP as an economic indicator and GDP is not significant when both GDP and health spending are regressed together. However, the R-square is only 7.4%

For Life Expectancy at Birth

```
# Both GDP and percentage of GDP spent on health:all economic indicators
linregecon<-lm(LifeExpectancy~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA)
summary(linregecon)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ GDP_currentUSD + HealthExpGDPperc,
##     data = econtrainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.027  -4.173   2.037   5.548  12.570
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.748e+01  1.478e+00  45.661  < 2e-16 ***
## GDP_currentUSD   6.250e-13  3.433e-13   1.820  0.07043 .
## HealthExpGDPperc 6.142e-01  2.157e-01   2.847  0.00495 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.292 on 171 degrees of freedom
## Multiple R-squared:  0.08585,    Adjusted R-squared:  0.07516
## F-statistic: 8.029 on 2 and 171 DF,  p-value: 0.0004646
```

```
print("Training error of Economic indicators on Life Expectancy at birth are:")
```

```
## [1] "Training error of Economic indicators on Life Expectancy at birth are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtrainNA)
MAPE(econtrainNA$LifeExpectancy, predictlinregecon)
```

```
## [1] 0.08111891
```

```
print("Test error of Economic indicators on Life Expectancy at birth are:")
```

```
## [1] "Test error of Economic indicators on Life Expectancy at birth are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtestNA)
MAPE(econtestNA$LifeExpectancy, predictlinregecon)
```

```
## [1] 0.07819311
```

```
# Just GDP
linregecon<-lm(LifeExpectancy~GDP_currentUSD,data=econtrainNA)
summary(linregecon)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ GDP_currentUSD, data = econtrainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.694  -5.979   1.219   5.628  11.164
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.136e+01  5.815e-01 122.726  < 2e-16 ***
## GDP_currentUSD 9.222e-13  3.337e-13   2.764  0.00634 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.441 on 172 degrees of freedom
## Multiple R-squared:  0.04251,    Adjusted R-squared:  0.03695
## F-statistic: 7.637 on 1 and 172 DF,  p-value: 0.006341
```

```
print("Training error of just GDP on Life Expectancy at birth are:")
```

```
## [1] "Training error of just GDP on Life Expectancy at birth are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtrainNA)
MAPE(econtrainNA$LifeExpectancy, predictlinregecon)
```

```
## [1] 0.08612337
```

```
print("Test error of just GDP on Life Expectancy at birth are:")
```

```
## [1] "Test error of just GDP on Life Expectancy at birth are:"
```

```
predictlinregecon = predict(linregecon, newdata = econtestNA)
MAPE(econtestNA$LifeExpectancy, predictlinregecon)
```

```
## [1] 0.08313759
```

```
# Just Health expenditure as GDP percentage
linregecon<-lm(LifeExpectancy~HealthExpGDPperc,data=econtrainNA)
summary(linregecon)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ HealthExpGDPperc, data = econtrainNA)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.524  -4.209   1.780   5.672  12.778
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       66.9646     1.4602  45.860  < 2e-16 ***
## HealthExpGDPperc   0.7336     0.2069   3.546 0.000504 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.341 on 172 degrees of freedom
## Multiple R-squared:  0.06813,    Adjusted R-squared:  0.06271
## F-statistic: 12.58 on 1 and 172 DF,  p-value: 0.0005037
```

```r
print("Training error of just Health Expenditure on Life Expectancy at birth are:")
```

```
## [1] "Training error of just Health Expenditure on Life Expectancy at birth are:"
```

```r
predictlinregecon = predict(linregecon, newdata = econtrainNA)
MAPE(econtrainNA$LifeExpectancy, predictlinregecon)
```

```
## [1] 0.08101375
```

```r
print("Test error of just Health Expenditure on Life Expectancy at birth are:")
```

```
## [1] "Test error of just Health Expenditure on Life Expectancy at birth are:"
```

```r
predictlinregecon = predict(linregecon, newdata = econtestNA)
MAPE(econtestNA$LifeExpectancy, predictlinregecon)
```

```
## [1] 0.07815278
```

Error rates are similar as for life expectancy but Health Expenditure as a percentage of GDP exerts the most influence and is much more influential than GDP ## Linear regression of Health Parameters

Linear Regression of Health Infrastructure

on HALE

```r
linreghealth<-lm(HALE~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ DrinkingWater + BasicSanitization + CleanTechFuel,
##     data = trainmainNA)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -14.9069  -1.9489   0.2398   2.5806   8.9457
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        45.36262    1.22049  37.167  < 2e-16 ***
## DrinkingWater       0.11295    0.02726   4.143 5.69e-05 ***
## BasicSanitization   0.07548    0.02506   3.012  0.00305 **
## CleanTechFuel       0.04119    0.01554   2.650  0.00891 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.588 on 151 degrees of freedom
## Multiple R-squared:  0.733,  Adjusted R-squared:  0.7277
## F-statistic: 138.2 on 3 and 151 DF,  p-value: < 2.2e-16
```

```
print("Training error of Health Infrastructure on HALE are :")
```

```
## [1] "Training error of Health Infrastructure on HALE are :"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.04414357
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.04369447
```

```
linreghealth<-lm(HALE~DrinkingWater,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ DrinkingWater, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.7486  -2.5897   0.4372   2.9383   8.7010
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.24697    1.20533   35.05   <2e-16 ***
## DrinkingWater  0.25325    0.01445   17.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.978 on 153 degrees of freedom
## Multiple R-squared:  0.6674, Adjusted R-squared:  0.6653
## F-statistic: 307.1 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
print("Training error of proportion of population using basic drinking water services on HALE are :")
```

```
## [1] "Training error of proportion of population using basic drinking water services on HALE are :"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.05095229
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.04953339
```

```
linreghealth<-lm(HALE~BasicSanitization,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ BasicSanitization, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1087  -1.9122   0.3371   2.8744   8.8934
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       48.07977    0.85894   55.98   <2e-16 ***
## BasicSanitization  0.19634    0.01081   18.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.883 on 153 degrees of freedom
## Multiple R-squared:  0.6831, Adjusted R-squared:  0.681
## F-statistic: 329.8 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
print("Training error of proportion of population having availability of basic sanitization facilities
```

```
## [1] "Training error of proportion of population having availability of basic sanitization facilities
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.04774236
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.04819555
```

```
linreghealth<-lm(HALE~CleanTechFuel,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ CleanTechFuel, data = trainmainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.088  -2.610   0.302   3.166   9.510
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   53.477457   0.682437   78.36   <2e-16 ***
## CleanTechFuel  0.145856   0.009398   15.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.299 on 153 degrees of freedom
## Multiple R-squared:  0.6115, Adjusted R-squared:  0.609
## F-statistic: 240.9 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
print("Training error of proportion of population with primary reliance on clean fuels and technologies
```

```
## [1] "Training error of proportion of population with primary reliance on clean fuels and technologie
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.05380911
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.05208417
```

Health Infra On Life Expectancy

```r
linreghealth<-lm(LifeExpectancy~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ DrinkingWater + BasicSanitization +
##     CleanTechFuel, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2224  -2.4217   0.3565   3.1569  10.0620
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       52.07179    1.40744  36.998  < 2e-16 ***
## DrinkingWater      0.13481    0.03144   4.288  3.2e-05 ***
## BasicSanitization  0.07025    0.02890   2.431  0.01625 *
## CleanTechFuel      0.05601    0.01792   3.125  0.00213 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.137 on 151 degrees of freedom
## Multiple R-squared:  0.7281, Adjusted R-squared:  0.7227
## F-statistic: 134.8 on 3 and 151 DF,  p-value: < 2.2e-16
```

```r
print("Training error of Health Infrastructure on Life Expectancy at Birth are :")
```

```
## [1] "Training error of Health Infrastructure on Life Expectancy at Birth are :"
```

```r
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.04471182
```

```r
print("Test error:")
```

```
## [1] "Test error:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.04453619
```

```
linreghealth<-lm(LifeExpectancy~DrinkingWater,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ DrinkingWater, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.9975  -2.8824   0.2543   3.8445  10.0069
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    48.41184    1.38529   34.95   <2e-16 ***
## DrinkingWater   0.28860    0.01661   17.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.571 on 153 degrees of freedom
## Multiple R-squared:  0.6637, Adjusted R-squared:  0.6615
## F-statistic: 301.9 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
print("Training error of proportion of population using basic drinking water services on Life Expectancy
```

```
## [1] "Training error of proportion of population using basic drinking water services on Life Expectanc
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.05170885
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.04985947
```

```
linreghealth<-lm(LifeExpectancy~BasicSanitization,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ BasicSanitization, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1898  -2.2911   0.4744   3.3766  10.1086
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       55.18851    1.00375   54.98   <2e-16 ***
## BasicSanitization  0.22200    0.01263   17.57   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.538 on 153 degrees of freedom
## Multiple R-squared:  0.6686, Adjusted R-squared:  0.6665
## F-statistic: 308.7 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
print("Training error of proportion of population having availability of basic sanitization facilities
```

```
## [1] "Training error of proportion of population having availability of basic sanitization facilities
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.04968925
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.05012416
```

```
linreghealth<-lm(LifeExpectancy~CleanTechFuel,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ CleanTechFuel, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4890  -3.0296   0.2459   3.8747  10.7865
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   61.12736    0.77360   79.02   <2e-16 ***
## CleanTechFuel  0.16754    0.01065   15.73   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.873 on 153 degrees of freedom
## Multiple R-squared:  0.6178, Adjusted R-squared:  0.6153
## F-statistic: 247.3 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
print("Training error of proportion of population with primary reliance on clean fuels and technologies
```

```
## [1] "Training error of proportion of population with primary reliance on clean fuels and technologie
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.05352415
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.05257985
```

The error rates do not differ significantly between HALE and life expectancy.

###Linear Regression of Health Behaviour #### On HALE

```
# Both GDP and percentage of GDP spent on health:all health behaviours
linreghealth<-lm(HALE~Alcohol+Tobacco+BMI30Plus,data=trainfullNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ Alcohol + Tobacco + BMI30Plus, data = trainfullNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.7639  -2.9309   0.7176   3.8407  15.4125
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.10690    1.60591  33.692  < 2e-16 ***
## Alcohol      0.51893    0.12737   4.074 8.32e-05 ***
## Tobacco      0.07440    0.05380   1.383    0.169
## BMI30Plus    0.22982    0.05681   4.046 9.27e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.83 on 120 degrees of freedom
## Multiple R-squared:  0.2779, Adjusted R-squared:  0.2598
## F-statistic: 15.39 on 3 and 120 DF,  p-value: 1.557e-08
```

```
print("Training error of Health Behaviours on HALE are:")
```

```
## [1] "Training error of Health Behaviours on HALE are:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = trainfullNA)
MAPE(trainfullNA$HALE, predictlinreghealth)
```

```
## [1] 0.06938704
```

```r
print("Test error:")
```

```
## [1] "Test error:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = testfullNA)
MAPE(testfullNA$HALE, predictlinreghealth)
```

```
## [1] 0.06544362
```

```r
# Just Alcohol
linreghealth<-lm(HALE~Alcohol,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ Alcohol, data = trainmainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.063  -4.149   1.299   4.652  13.344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.4565     0.8943  65.367  < 2e-16 ***
## Alcohol       0.6763     0.1202   5.628 8.49e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6.278 on 153 degrees of freedom
## Multiple R-squared:  0.1715, Adjusted R-squared:  0.1661
## F-statistic: 31.67 on 1 and 153 DF,  p-value: 8.485e-08
```

```r
print("Training error of Alcohol on HALE are:")
```

```
## [1] "Training error of Alcohol on HALE are:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.0806903
```

```r
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.07736505
```

```
# Just Tobacco
linreghealth<-lm(HALE~Tobacco,data=trainfullNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ Tobacco, data = trainfullNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.578  -5.098   1.246   5.637  10.818
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 59.74767    1.46321  40.833  < 2e-16 ***
## Tobacco      0.15964    0.05874   2.718  0.00753 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.607 on 122 degrees of freedom
## Multiple R-squared:  0.05708,    Adjusted R-squared:  0.04935
## F-statistic: 7.386 on 1 and 122 DF,  p-value: 0.007531
```

```
print("Training error of Tobacco on HALE are:")
```

```
## [1] "Training error of Tobacco on HALE are:"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainfullNA)
MAPE(trainfullNA$HALE, predictlinreghealth)
```

```
## [1] 0.08562256
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testfullNA)
MAPE(testfullNA$HALE, predictlinreghealth)
```

```
## [1] 0.08249917
```

```
# Just High BMI (Overweight and Obesity)
linreghealth<-lm(HALE~BMI30Plus,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ BMI30Plus, data = trainmainNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.5092  -4.3889   0.6392   4.4338  14.9502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 57.49079    1.09830  52.345  < 2e-16 ***
## BMI30Plus    0.28755    0.05462   5.265 4.67e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6.346 on 153 degrees of freedom
## Multiple R-squared:  0.1534, Adjusted R-squared:  0.1479
## F-statistic: 27.72 on 1 and 153 DF,  p-value: 4.673e-07
```

```
print("Training error of obesity and overweight on HALE are:")
```

```
## [1] "Training error of obesity and overweight on HALE are:"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.07994592
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$HALE, predictlinreghealth)
```

```
## [1] 0.075778
```

Health Behaviour On Life Expectancy at birth

```
# All health behaviours
linreghealth<-lm(LifeExpectancy~Alcohol+Tobacco+BMI30Plus,data=trainfullNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ Alcohol + Tobacco + BMI30Plus,
##     data = trainfullNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.6135  -3.6952   0.9912   4.7798  16.6824
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 62.16414    1.84312  33.728  < 2e-16 ***
## Alcohol      0.59363    0.14618   4.061 8.75e-05 ***
## Tobacco      0.06729    0.06175   1.090    0.278
## BMI30Plus    0.27074    0.06520   4.153 6.18e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6.691 on 120 degrees of freedom
## Multiple R-squared:  0.2741, Adjusted R-squared:  0.256
## F-statistic: 15.11 on 3 and 120 DF,  p-value: 2.106e-08
```

```
print("Training error of Health Behaviours on Life expectancy at birth are:")
```

```
## [1] "Training error of Health Behaviours on Life expectancy at birth are:"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainfullNA)
MAPE(trainfullNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.07092473
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.06733174
```

```
# Just Alcohol
linreghealth<-lm(LifeExpectancy~Alcohol,data=trainmainNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ Alcohol, data = trainmainNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.918  -4.873   1.185   5.363  14.318
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.9034     1.0235  65.370  < 2e-16 ***
## Alcohol       0.7677     0.1375   5.582 1.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.185 on 153 degrees of freedom
## Multiple R-squared:  0.1692, Adjusted R-squared:  0.1637
## F-statistic: 31.15 on 1 and 153 DF,  p-value: 1.058e-07
```

```
print("Training error of Alcohol on LifeExpectancy at birth are:")
```

```
## [1] "Training error of Alcohol on LifeExpectancy at birth are:"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.08116622
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.07802916
```

```
# Just Tobacco
linreghealth<-lm(LifeExpectancy~Tobacco,data=trainfullNA)
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ Tobacco, data = trainfullNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.709  -5.562   1.119   6.231  11.340
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  68.7321     1.6839  40.817   <2e-16 ***
## Tobacco       0.1659     0.0676   2.455   0.0155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.604 on 122 degrees of freedom
## Multiple R-squared:  0.04707,    Adjusted R-squared:  0.03926
## F-statistic: 6.026 on 1 and 122 DF,  p-value: 0.01551
```

```
print("Training error of tobacco on LifeExpectancy at birth are:")
```

```
## [1] "Training error of tobacco on LifeExpectancy at birth are:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = trainfullNA)
MAPE(trainfullNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.08636125
```

```r
print("Test error:")
```

```
## [1] "Test error:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.08306901
```

```r
# Just High BMI (Overweight and Obesity)
linreghealth<-lm(LifeExpectancy~BMI30Plus,data=trainmainNA)
summary(linreghealth)
```

```
## 
## Call:
## lm(formula = LifeExpectancy ~ BMI30Plus, data = trainmainNA)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -23.4202  -4.9617   0.5734   5.0229  16.5133 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  65.7569     1.2548  52.405  < 2e-16 ***
## BMI30Plus     0.3292     0.0624   5.276 4.44e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 7.251 on 153 degrees of freedom
## Multiple R-squared:  0.1539, Adjusted R-squared:  0.1484 
## F-statistic: 27.84 on 1 and 153 DF,  p-value: 4.44e-07
```

```r
print("Training error of obesity and overweight on LifeExpectancy at birth are:")
```

```
## [1] "Training error of obesity and overweight on LifeExpectancy at birth are:"
```

```r
predictlinreghealth = predict(linreghealth, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.0798465
```

```r
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.07574762
```

Health Infrastructure and Behaviour Together

```
linreghealth<-lm(HALE~Alcohol+Tobacco+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel,data=trai
summary(linreghealth)
```

```
##
## Call:
## lm(formula = HALE ~ Alcohol + Tobacco + BMI30Plus + DrinkingWater +
##     BasicSanitization + CleanTechFuel, data = trainfullNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2681  -1.8508   0.5201   2.5665   8.5422
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       46.07672    1.67309  27.540  < 2e-16 ***
## Alcohol            0.08107    0.08337   0.972  0.33290
## Tobacco           -0.01938    0.03507  -0.553  0.58160
## BMI30Plus         -0.13229    0.04299  -3.077  0.00260 **
## DrinkingWater      0.11367    0.03665   3.102  0.00241 **
## BasicSanitization  0.09606    0.03176   3.024  0.00306 **
## CleanTechFuel      0.04387    0.01965   2.232  0.02750 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.479 on 117 degrees of freedom
## Multiple R-squared:  0.7493, Adjusted R-squared:  0.7364
## F-statistic: 58.28 on 6 and 117 DF,  p-value: < 2.2e-16
```

```
print("Training error of Health Behaviours+Health Infrastructure on HALE are:")
```

```
## [1] "Training error of Health Behaviours+Health Infrastructure on HALE are:"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainfullNA)
MAPE(trainfullNA$HALE, predictlinreghealth)
```

```
## [1] 0.04090307
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testfullNA)
MAPE(testfullNA$HALE, predictlinreghealth)
```

```
## [1] 0.04111791
```

```
linreghealth<-lm(LifeExpectancy~Alcohol+Tobacco+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel
summary(linreghealth)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ Alcohol + Tobacco + BMI30Plus +
##     DrinkingWater + BasicSanitization + CleanTechFuel, data = trainfullNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.314  -2.491   0.621   2.744   9.650
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        52.65427    1.93225  27.250  < 2e-16 ***
## Alcohol             0.09391    0.09629   0.975 0.331440
## Tobacco            -0.03844    0.04050  -0.949 0.344551
## BMI30Plus          -0.14231    0.04965  -2.866 0.004928 **
## DrinkingWater       0.14398    0.04233   3.402 0.000917 ***
## BasicSanitization   0.09348    0.03668   2.548 0.012114 *
## CleanTechFuel       0.05557    0.02270   2.448 0.015843 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.018 on 117 degrees of freedom
## Multiple R-squared:  0.7448, Adjusted R-squared:  0.7317
## F-statistic: 56.92 on 6 and 117 DF,  p-value: < 2.2e-16
```

```
print("Training error of Health Behaviours+Health Infrastructure on Life expectancy at birth are:")
```

```
## [1] "Training error of Health Behaviours+Health Infrastructure on Life expectancy at birth are:"
```

```
predictlinreghealth = predict(linreghealth, newdata = trainfullNA)
MAPE(trainfullNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.0420137
```

```
print("Test error:")
```

```
## [1] "Test error:"
```

```
predictlinreghealth = predict(linreghealth, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, predictlinreghealth)
```

```
## [1] 0.04226727
```

The regression shows health infrastructure parameters to be more significant with drinking water being the most salient and basic sanitization and clean techmnology and fuel use also being significant at the 95% confidence level. Obesity is significant at the 99% confidence level. The training and test errors are impressive at only 4.20% and 4.23% respectively.

LASSO

For HALE

```
trainlasso<-trainfullNA
trainlasso$Location<-trainlasso$LifeExpectancy<-NULL #Removing dependent variables and non-numeric valu
set.seed(32)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

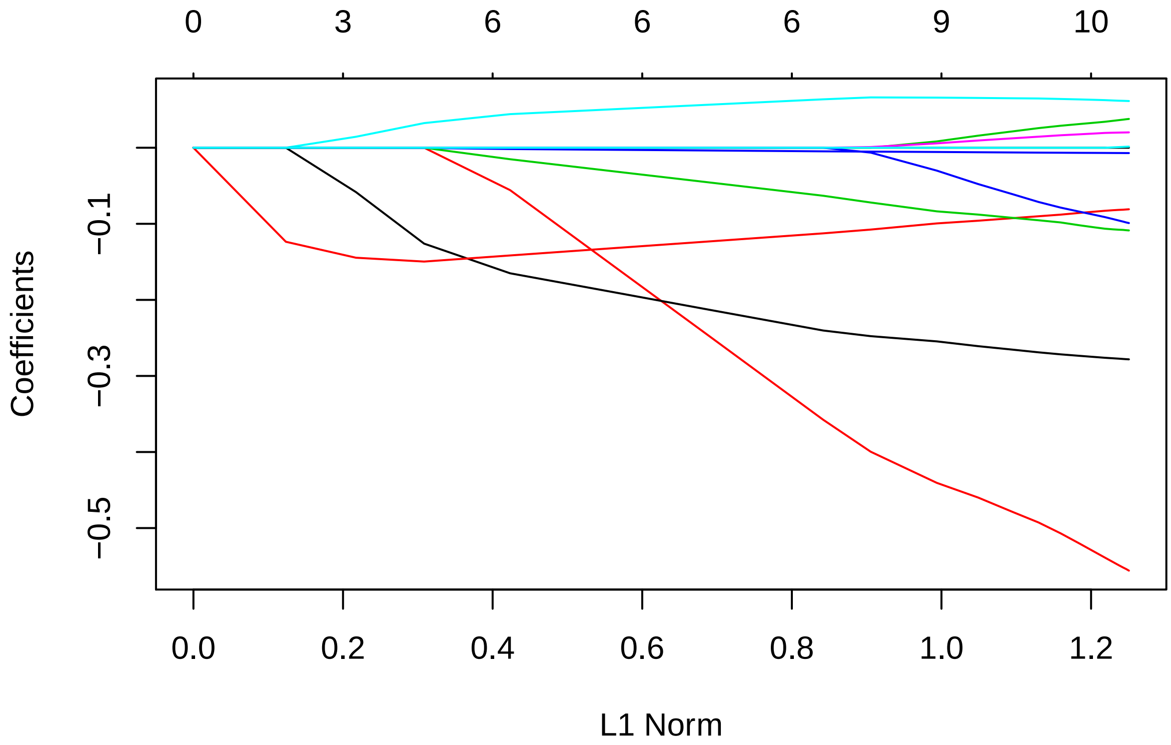
```
## Loaded glmnet 2.0-16
```

```
x=model.matrix(HALE~.,trainlasso)
y=trainlasso$HALE
grid=10^seq(10,-2,length=100)
lasso.mod=glmnet(x,y,alpha=1,lambda=grid)
dim(coef(lasso.mod))
```

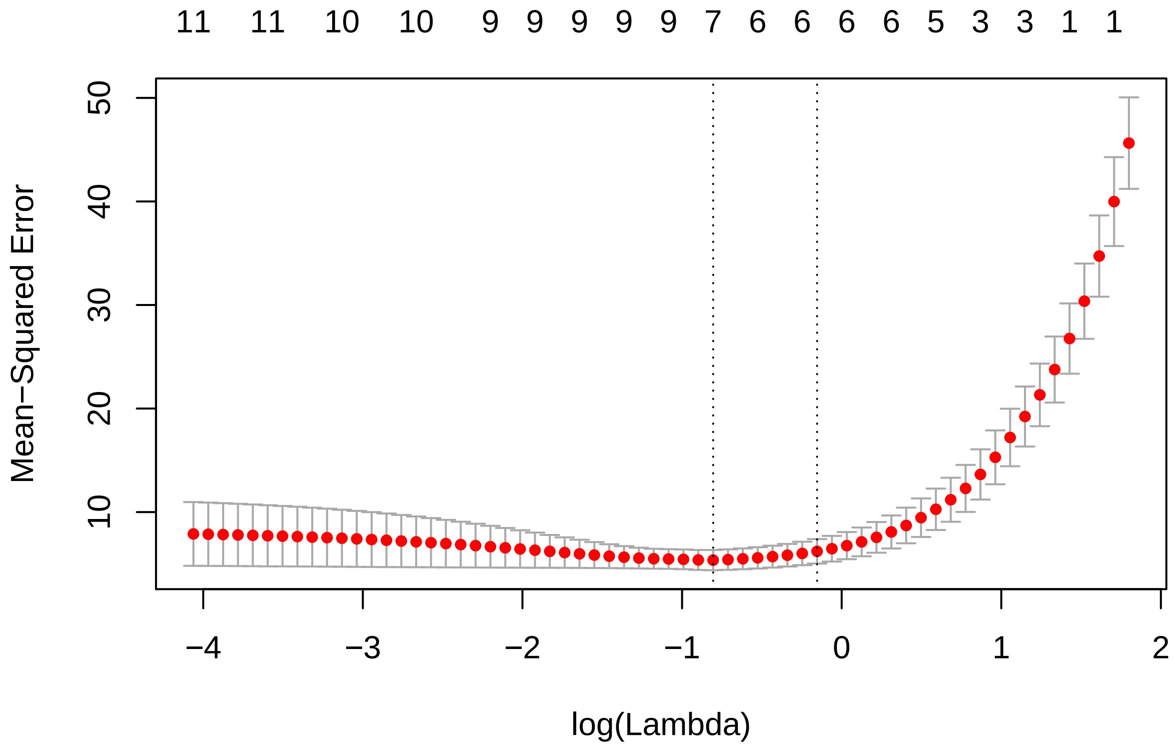
```
## [1]  16 100
```

```
plot(lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique ’x’ values
```



```
cv.out=cv.glmnet(x,y,alpha=1)
plot(cv.out)
```



```
bestlam=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=4,newx=x)
mean((lasso.pred-y)^2)
```

```
## [1] 24.97151
```

```
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef=predict(lasso.mod,type="coefficients",s=bestlam)[1:16,]
lasso.coef
```

```
##       (Intercept)       (Intercept)              NTDs         Poisoning
##     66.6251362355      0.0000000000      0.0000000000     -0.3751109625
##           Tobacco      Tuberculosis   Under5Mortality     DrinkingWater
##      0.0000000000     -0.0048618267      0.0000000000      0.0647109924
##           Alcohol BasicSanitization         CancerEtc   InfantMortality
##      0.0000000000      0.0003736524     -0.2433524097     -0.1104758268
## MaternalMortality NeonatalMortality         BMI30Plus     CleanTechFuel
##      0.0000000000     -0.0668166536     -0.0026865670      0.0000000000
```

The provision of Drinking water and Sanitization facilities are having a positive effect on HALE among Health Infrastructures. On the negative side, the population dying due to poisoning, which could be linked to prevalent toxicity and thus indirectly to infrastructure is relevant. Poisoning could also be linked to health behaviours. Among health behaviours, only high BMI has an impact, a negative impact on HALE. Tobacco and Alcohol are suppressed to zero values. Tuberculosis, Cancer and other non-communicable diseases, infant mortality, and neonatal mortality have an inverse relaionship with HALE.

LASSO results do not show any difference in factors affecting HALE and Life expectancy at birth.

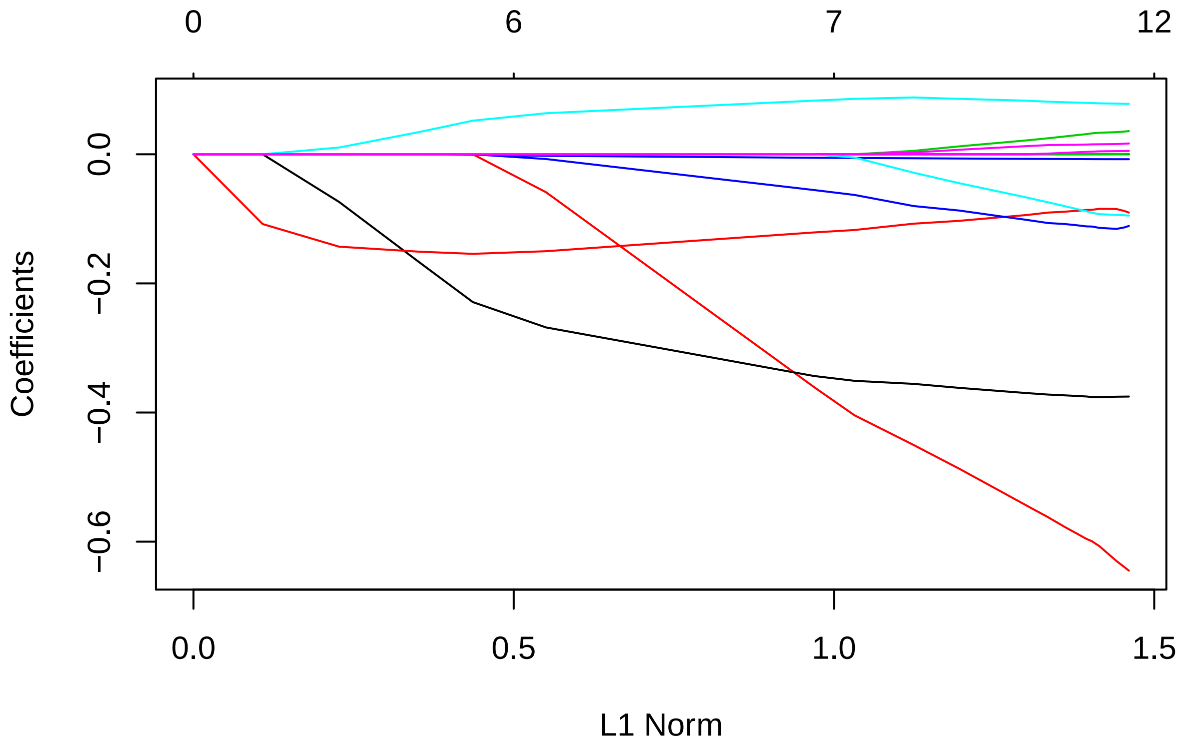
Lasso for Life Expectancy

```
set.seed(22)
trainlasso<-trainfullNA
trainlasso$Location<-trainlasso$HALE<-NULL #Removing dependent variables and non-numeric values
x=model.matrix(LifeExpectancy~.,trainlasso)
y=trainlasso$LifeExpectancy
grid=10^seq(10,-2,length=100)
lasso.mod=glmnet(x,y,alpha=1,lambda=grid)
dim(coef(lasso.mod))
```

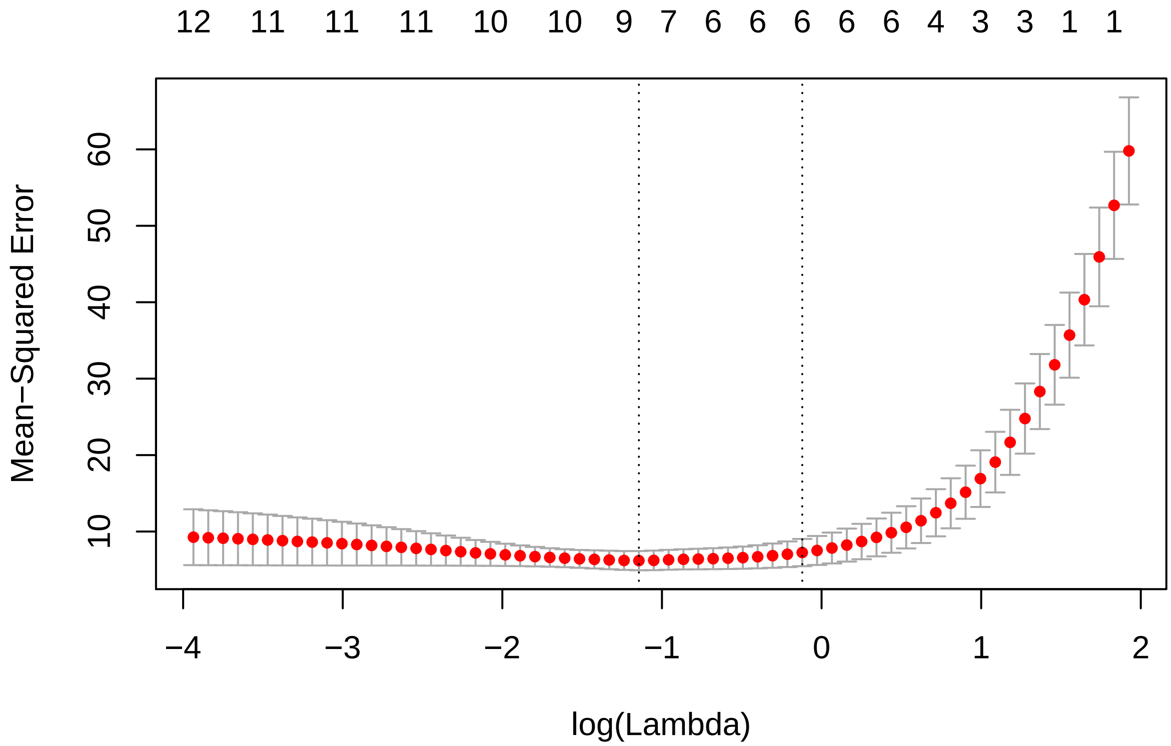
```
## [1]  16 100
```

```
plot(lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```



```
cv.out=cv.glmnet(x,y,alpha=1)
plot(cv.out)
```



```
bestlam=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=4,newx=x)
mean((lasso.pred-y)^2)
```

```
## [1] 27.55985
```

```
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef=predict(lasso.mod,type="coefficients",s=bestlam)[1:16,]
lasso.coef
```

```
##       (Intercept)       (Intercept)              NTDs         Poisoning
##      76.349910744       0.000000000       0.000000000      -0.433436916
##           Tobacco      Tuberculosis   Under5Mortality     DrinkingWater
##       0.003366896      -0.006047105       0.000000000       0.087203320
##           Alcohol BasicSanitization         CancerEtc   InfantMortality
##       0.000000000       0.001766641      -0.353872210      -0.111083690
## MaternalMortality NeonatalMortality          BMI30Plus      CleanTechFuel
##       0.000000000      -0.073882266      -0.019939322       0.000000000
```

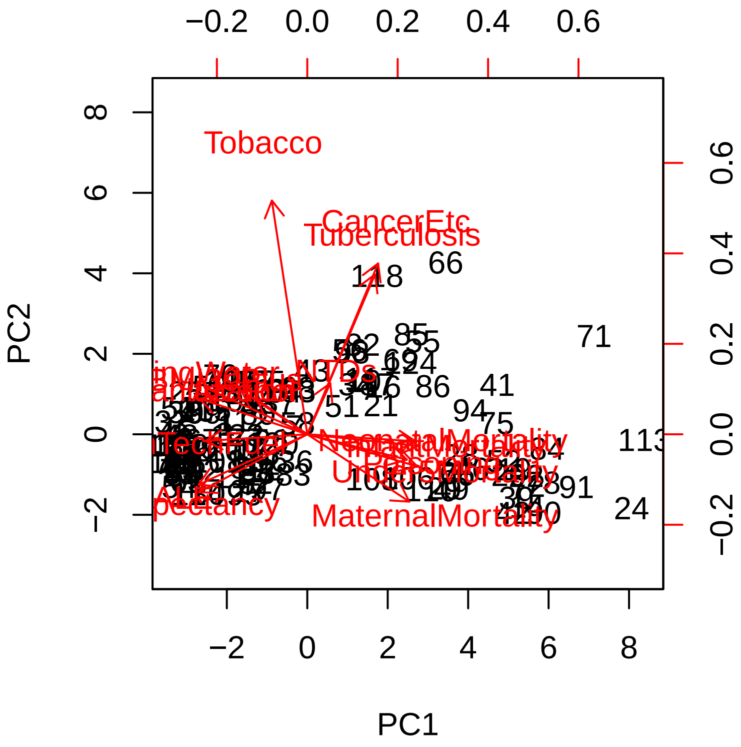
The provision of Drinking water and Sanitization facilities are having an effect among Health Infrastructures. On the negative side, the population dying due to poisoning, which could be linked to prevalent toxicity and thus indirectly to infrastructure is relevant. Poisoning could also be linked to health behaviours. Among health behaviours, high BMI has a negative impact on life expectancy. Tobacco is relevant as per Lasso but it surprisingly is positively related, likely an anomalous result. Alcohol has no effect. Tuberculosis, Cancer and other non-communicable diseases, infant mortality, and neonatal mortality have an inverse relaionship with life expectancy.

Principal Components Analysis

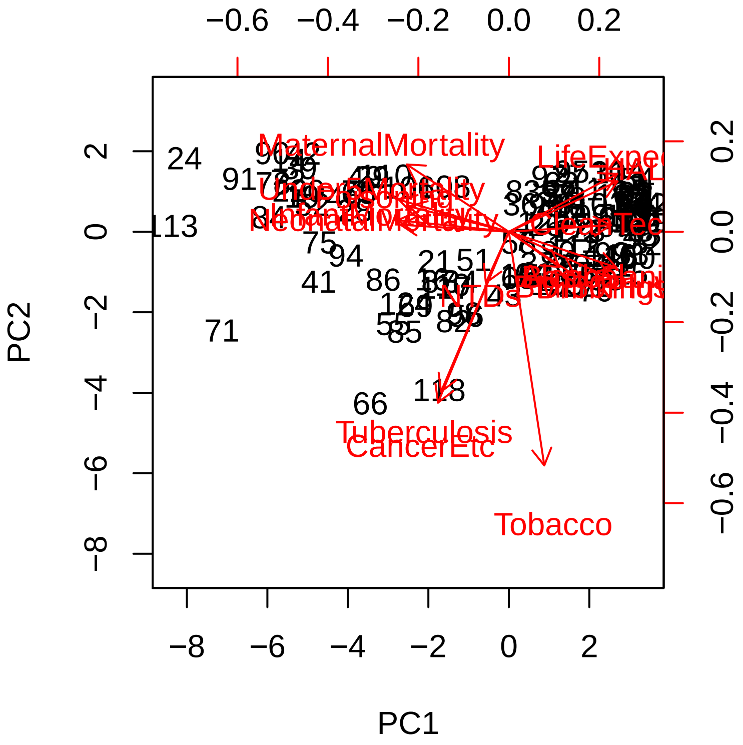
```
set.seed(8)
trainlasso<-trainfullNA
trainlasso$Location<- NULL  # as PCA needs only numeric values
pr.out=prcomp(trainlasso,scale=TRUE)
pr.out$rotation
```

```
##                           PC1         PC2           PC3          PC4
## NTDs               0.06366656  0.14092609 -0.7459854550 -0.467605530
## Poisoning          0.27537555 -0.07010893 -0.0267227506  0.127370546
## Tobacco           -0.09794320  0.64580080 -0.1419964841  0.147736277
## Tuberculosis       0.18746636  0.44195972  0.0579640312  0.041065757
## Under5Mortality    0.30372133 -0.08862076  0.0002854606  0.044684355
## DrinkingWater     -0.28429026  0.12971044 -0.0402535364 -0.141799218
## Alcohol           -0.14030566  0.09734977 -0.2387113157  0.788020356
## BasicSanitization -0.29703406  0.09833366  0.0643498614 -0.060763858
## CancerEtc          0.19589566  0.47224718  0.1606292293 -0.067911461
## HALE              -0.30048948 -0.13932125 -0.1614894998  0.020740160
## InfantMortality    0.30687223 -0.03200100  0.0144439889 -0.002861043
## LifeExpectancy    -0.29948486 -0.16018760 -0.1722789385  0.005820188
## MaternalMortality  0.28208642 -0.18602696 -0.0260360533  0.142335239
## NeonatalMortality  0.29944085 -0.01954530 -0.0216539668 -0.114686011
## BMI30Plus         -0.18239572  0.12070975  0.5161644181 -0.232002564
## CleanTechFuel     -0.28170784 -0.01848659  0.0847501817 -0.050157742
##                           PC5         PC6         PC7         PC8
## NTDs              -0.38902786  0.06090773 -0.07149181 -0.15566995
## Poisoning         -0.21142642 -0.15174129  0.13047689 -0.35868744
## Tobacco            0.31821898 -0.46547395 -0.38170894 -0.07679975
## Tuberculosis       0.16703205  0.78334332 -0.14767808 -0.04049885
## Under5Mortality   -0.09159429 -0.10097616 -0.26260279  0.25979665
## DrinkingWater     -0.06078954  0.01277605 -0.04286920  0.51196858
## Alcohol           -0.45292539  0.09964353 -0.03141903  0.03415308
## BasicSanitization -0.12924766  0.03051308  0.07025966  0.28621676
## CancerEtc         -0.22905420 -0.24734876  0.61524487  0.20343026
## HALE               0.14986439 -0.07515836 -0.02872698  0.03397257
## InfantMortality   -0.06579931 -0.05519052 -0.24583344  0.29208304
## LifeExpectancy     0.13214223 -0.05284730 -0.07592757  0.03309127
## MaternalMortality -0.10476013 -0.11452359 -0.24347819  0.21707948
## NeonatalMortality -0.01028213 -0.02520392 -0.16867738  0.33684028
## BMI30Plus         -0.49545157 -0.08915690 -0.44736741 -0.30056244
## CleanTechFuel     -0.29596109  0.16548087 -0.01451726  0.21682032
##                           PC9        PC10         PC11        PC12
## NTDs              -0.06221512  0.07835620 -0.039254776  0.04991329
## Poisoning          0.66622297 -0.47923085  0.024297315 -0.08813402
## Tobacco            0.16463738  0.09259687  0.148968369  0.04285618
## Tuberculosis       0.16921053 -0.03056263 -0.205276668 -0.15914723
## Under5Mortality    0.05419434 -0.02571039  0.005089566 -0.11338702
## DrinkingWater      0.07258959 -0.50651557 -0.217210191  0.20841032
## Alcohol           -0.25600465 -0.08145285  0.027762591 -0.03193919
## BasicSanitization  0.26064272 -0.09909101 -0.019337450  0.32746764
## CancerEtc         -0.06755791  0.15599372 -0.230836245 -0.30021687
## HALE               0.08866087 -0.02236167 -0.267308283 -0.50758259
## InfantMortality   -0.01498233 -0.06572235  0.091940801 -0.14959403
## LifeExpectancy     0.08728271 -0.02890682 -0.256873988 -0.47839411
## MaternalMortality  0.25999513  0.43585542 -0.544946598  0.21948624
## NeonatalMortality -0.18070020 -0.23884616  0.337697292 -0.30604212
## BMI30Plus         -0.16525395 -0.07206055 -0.187002488 -0.13192871
## CleanTechFuel      0.45500859  0.44589004  0.489933786 -0.19944722
##                          PC13         PC14         PC15         PC16
## NTDs              -0.041943267 -0.022472838  0.001811532 -0.015175436
## Poisoning          0.040123200  0.068532580 -0.026486792 -0.005183951
## Tobacco            0.026465679  0.036700523  0.005861529  0.006373944
## Tuberculosis      -0.038683903  0.008043777  0.029963695  0.003836156
## Under5Mortality   -0.142146615 -0.567571266  0.603766548  0.139661375
## DrinkingWater      0.503589952 -0.078362785 -0.010382455 -0.022911467
## Alcohol           -0.009981872  0.039473659 -0.005122882 -0.002542345
## BasicSanitization -0.765848499  0.128347079  0.030519006  0.022015701
## CancerEtc         -0.004020396 -0.026217024  0.003065696  0.040123280
## HALE              -0.128523507 -0.011846753  0.126427696 -0.680883222
## InfantMortality   -0.176547955 -0.319495879 -0.750373486 -0.140655202
## LifeExpectancy    -0.082559833  0.037654217 -0.152468307  0.702168647
## MaternalMortality  0.120431366  0.340735848 -0.018048838 -0.027381519
## NeonatalMortality -0.054017147  0.650543210  0.173413524  0.008895891
## BMI30Plus         -0.015295795  0.051720785  0.007666401 -0.012999458
## CleanTechFuel      0.248649683 -0.027785864 -0.002528952 -0.011475888
```

```r
biplot(pr.out,scale=0)
```



```r
pr.out$rotation=-pr.out$rotation
pr.out$x=-pr.out$x
biplot(pr.out,scale=0)
```



```r
pr.var=pr.out$sdev^2
pr.var
```

```
##  [1] 9.827949590 1.442739917 1.098201263 0.906083109 0.708175516
##  [6] 0.526281539 0.436016738 0.353803908 0.230386139 0.140075706
## [11] 0.137254455 0.085899823 0.069521745 0.031409574 0.004314610
## [16] 0.001886369
```

```r
print("Proportion of variance explained")
```

```
## [1] "Proportion of variance explained"
```

```r
pve=pr.var/sum(pr.var)
pve
```

```
##  [1] 0.6142468494 0.0901712448 0.0686375789 0.0566301943 0.0442609697
##  [6] 0.0328925962 0.0272510461 0.0221127442 0.0143991337 0.0087547316
## [11] 0.0085784034 0.0053687390 0.0043451091 0.0019630984 0.0002696631
## [16] 0.0001178981
```

PCA1 explains most of the variance but the results are not useful as they are spread among too many parameters. Lasso has provided us better results previously.

Random Forests

Economic Factors for HALE

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed (74)
fit.regforest <- randomForest(HALE~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA, mtry = 1, importanc
predictregforest1 <- predict(fit.regforest, newdata = econtrainNA, n.trees =5000)
MAPE(econtrainNA$HALE, predictregforest1)
```

```
## [1] 0.03558749
```

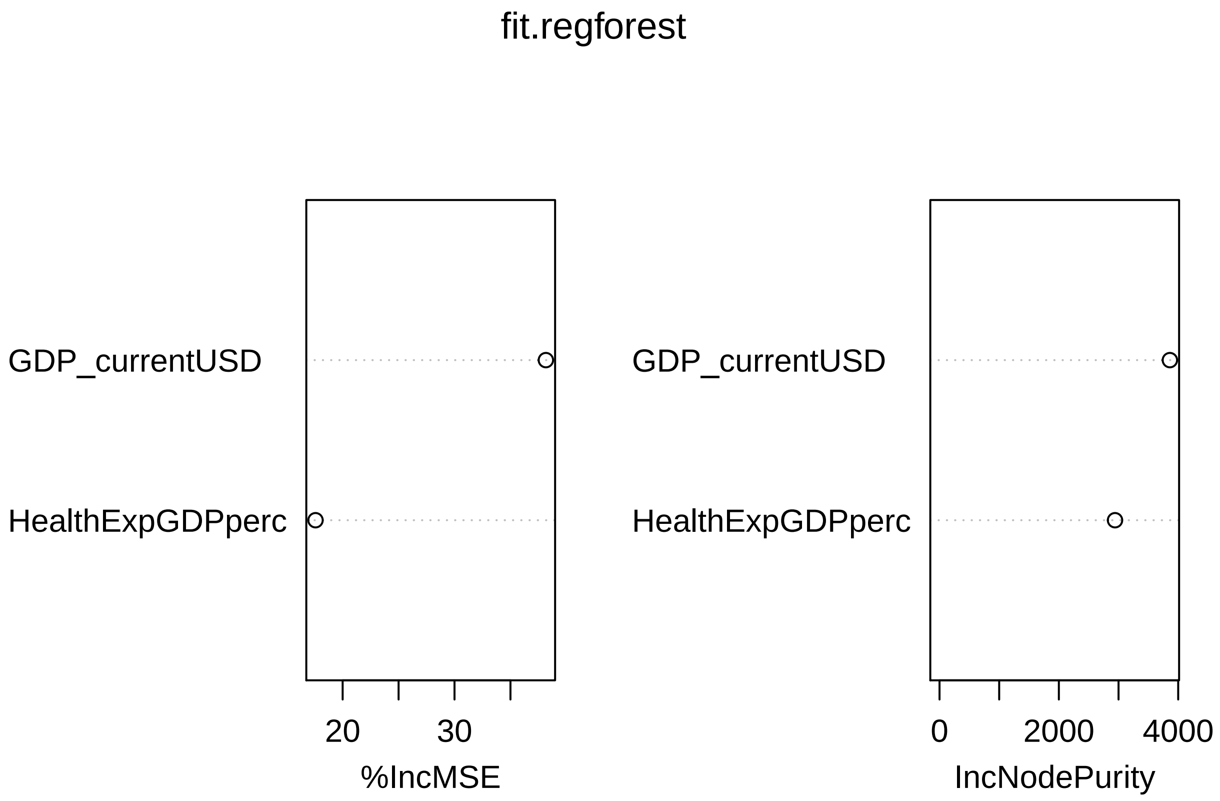
```
predictregforest2 = predict(fit.regforest, newdata = econtestNA)
MAPE(econtestNA$HALE, predictregforest2)
```

```
## [1] 0.05108338
```

```
importance(fit.regforest)
```

```
##                   %IncMSE IncNodePurity
## GDP_currentUSD   38.16649      3860.607
## HealthExpGDPperc 17.57501      2941.229
```

```
varImpPlot(fit.regforest)
```



Training error of 3.56% and test error of 5.11% on HALE with economic factors. GDP shown to have a slightly greater impact.

Economic factors for Life Expectancy

```
fit.regforest <- randomForest(LifeExpectancy~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA, mtry = 1
predictregforest1 <- predict(fit.regforest, newdata = econtrainNA, n.trees =5000)
MAPE(econtrainNA$LifeExpectancy, predictregforest1)
```

```
## [1] 0.0350192
```

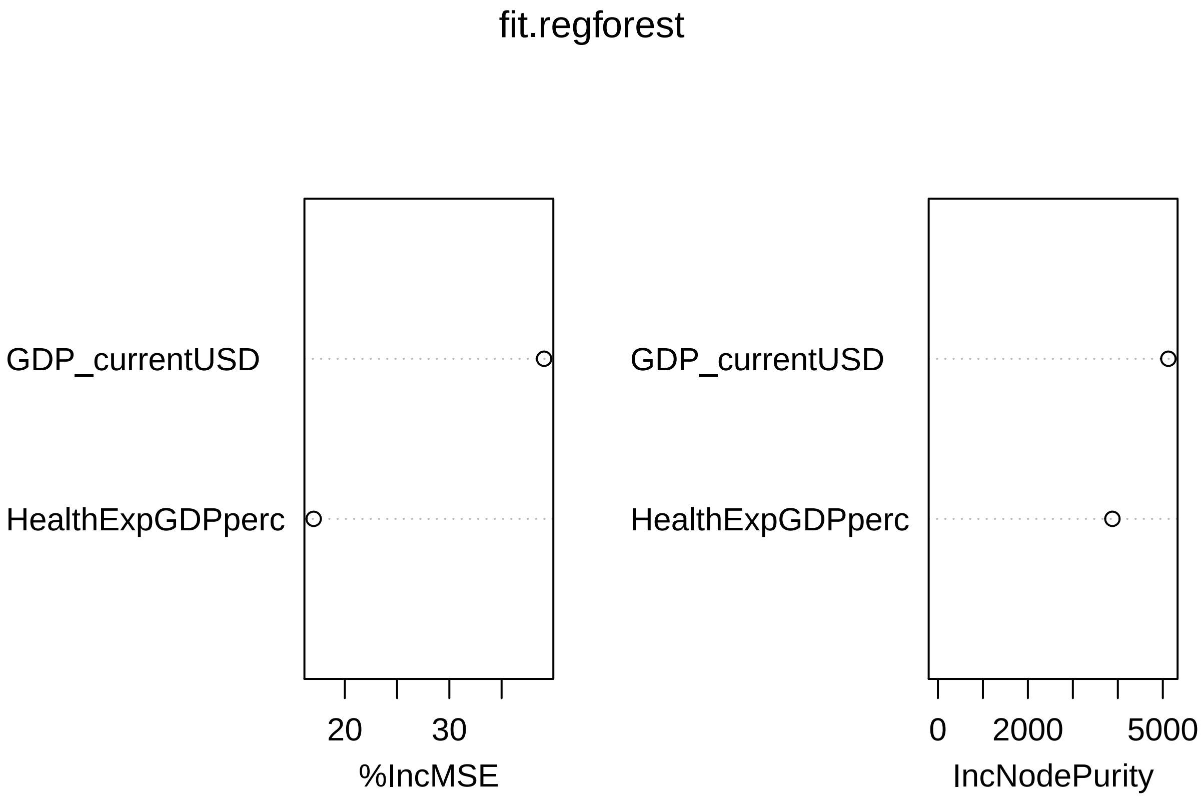
```
predictregforest2 = predict(fit.regforest, newdata = econtestNA)
MAPE(econtestNA$LifeExpectancy, predictregforest2)
```

```
## [1] 0.05081277
```

```
importance(fit.regforest)
```

```
##                    %IncMSE IncNodePurity
## GDP_currentUSD    39.06696      5123.672
## HealthExpGDPperc 17.01842      3879.431
```

```
varImpPlot(fit.regforest)
```



Training error of 3.50% and test error of 5.08% on Life expectancy with economic factors. GDP shown to have a slightly greater impact. No differences in results for HALE and Life expectancy

Health Infrastructure For HALE

```
library(randomForest)
set.seed (74)
fit.regforest <- randomForest(HALE~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA, mtry
predictregforest1 <- predict(fit.regforest, newdata = trainmainNA, n.trees =5000)
MAPE(trainmainNA$HALE, predictregforest1)
```

```
## [1] 0.02038385
```

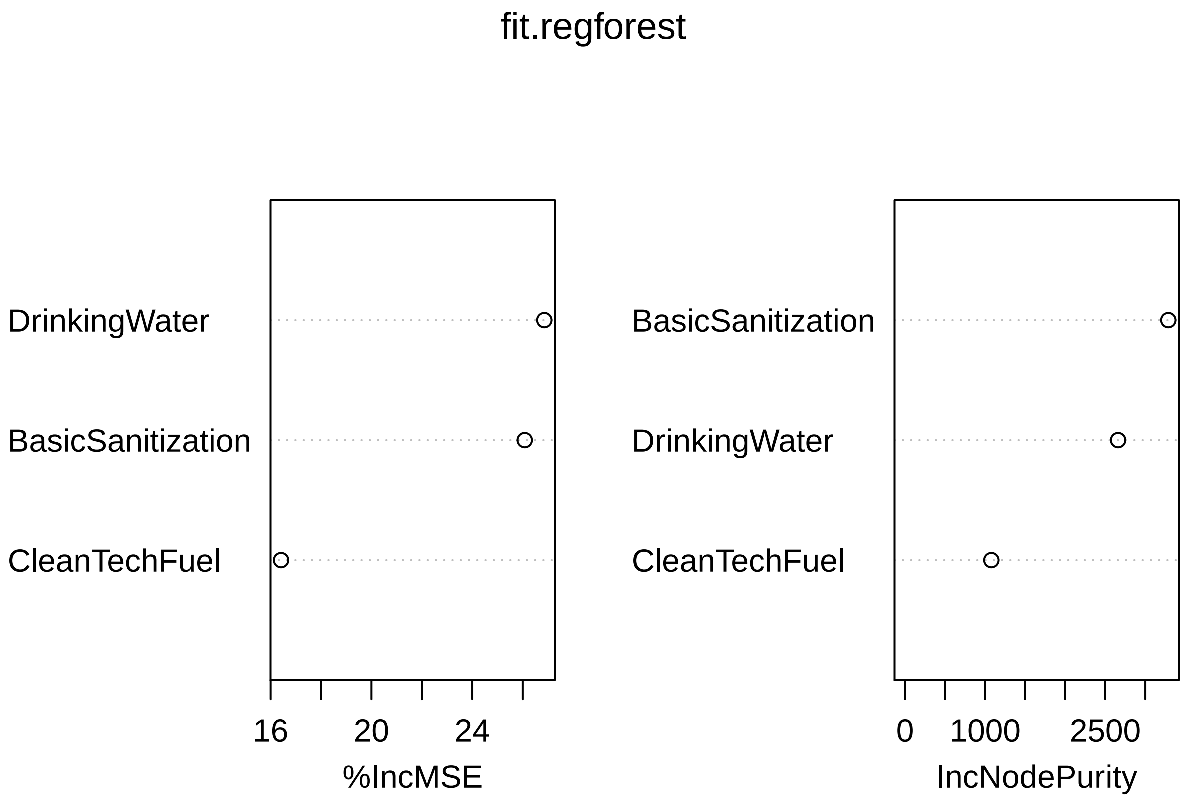
```
predictregforest2 = predict(fit.regforest, newdata = testmainNA)
MAPE(testmainNA$HALE, predictregforest2)
```

```
## [1] 0.02865956
```

```
importance(fit.regforest)
```

```
##                      %IncMSE IncNodePurity
## DrinkingWater      26.85805       2660.164
## BasicSanitization 26.07945       3287.725
## CleanTechFuel      16.41460       1077.731
```

```
varImpPlot(fit.regforest)
```



Training error is extremely low at 2.01%

Health Infrastructure For Life Expectancy

```
set.seed (73)
fit.regforest <- randomForest(LifeExpectancy~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainma
predictregforest1 <- predict(fit.regforest, newdata = trainmainNA, n.trees =5000)
MAPE(trainmainNA$LifeExpectancy, predictregforest1)
```

```
## [1] 0.02066733
```

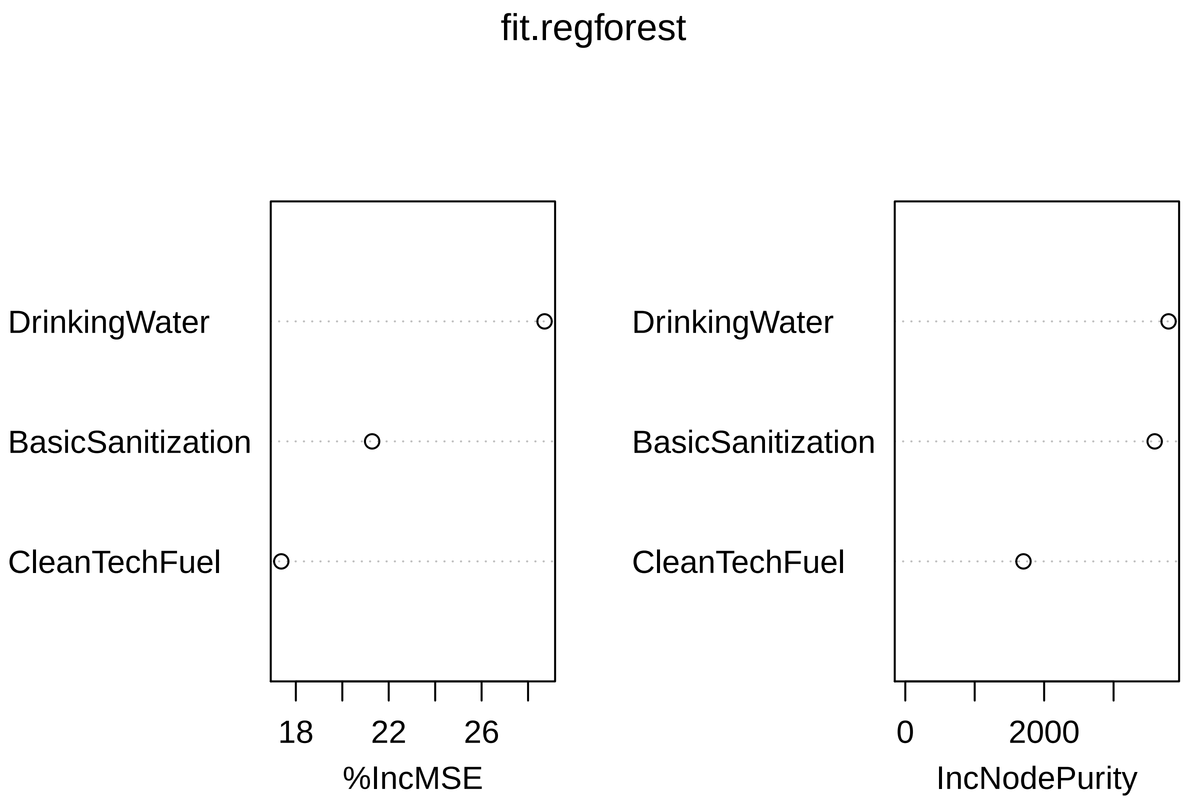
```
predictregforest2 = predict(fit.regforest, newdata = testmainNA)
MAPE(testmainNA$HALE, predictregforest2)
```

```
## [1] 0.1155203
```

```
importance(fit.regforest)
```

```
##                      %IncMSE IncNodePurity
## DrinkingWater      28.71153      3792.170
## BasicSanitization 21.28546      3592.556
## CleanTechFuel      17.37216      1702.331
```

```
varImpPlot(fit.regforest)
```



Training error is extremely low at 1.99%

Health Behaviours For HALE

```
set.seed (74)
fit.regforest <- randomForest(HALE~Alcohol+Tobacco+BMI30Plus,data=trainfullNA, mtry = 2, importance=TRU
predictregforest1 <- predict(fit.regforest, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$HALE, predictregforest1)
```

```
## [1] 0.03021143
```

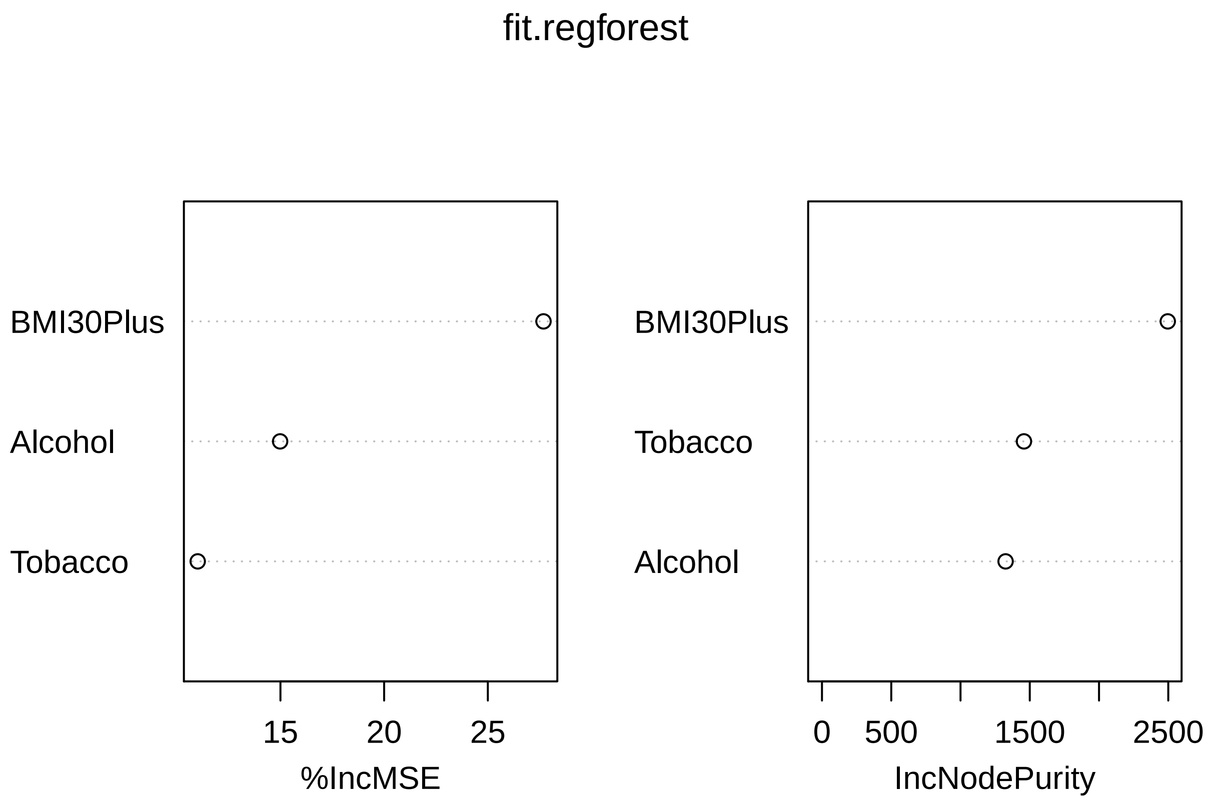
```
predictregforest2 = predict(fit.regforest, newdata = testfullNA)
MAPE(testfullNA$HALE, predictregforest2)
```

```
## [1] 0.03991063
```

```
importance(fit.regforest)
```

```
##            %IncMSE IncNodePurity
## Alcohol   14.98438      1325.534
## Tobacco   11.01194      1458.068
## BMI30Plus 27.68343      2496.240
```

```
varImpPlot(fit.regforest)
```



Training error is 2.92%, a very low figure

Health Behaviours For Life Expectancy

```
set.seed (74)
fit.regforest <- randomForest(LifeExpectancy~Alcohol+Tobacco+BMI30Plus,data=trainfullNA, mtry = 2, impor
predictregforest1 <- predict(fit.regforest, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$LifeExpectancy, predictregforest1)
```

```
## [1] 0.03100852
```

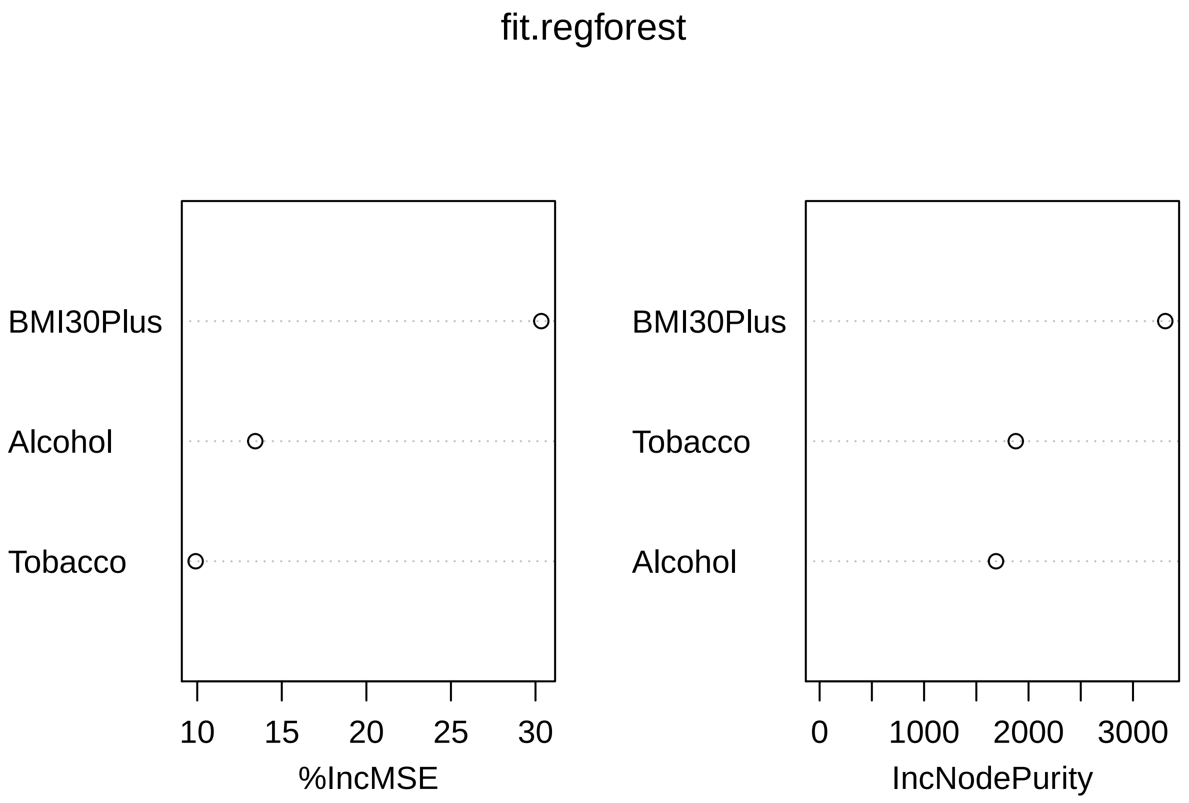
```
predictregforest2 = predict(fit.regforest, newdata = testfullNA)
MAPE(testfullNA$HALE, predictregforest2)
```

```
## [1] 0.1175942
```

```
importance(fit.regforest)
```

```
##            %IncMSE IncNodePurity
## Alcohol   13.43047      1688.331
## Tobacco    9.90616      1877.754
## BMI30Plus 30.34120      3308.981
```

```
varImpPlot(fit.regforest)
```



Training error is very low at 2.99%

Bagging

Bagging for Health Infrastructure

```
# Actual beginning of bagging code

fit.bag <- randomForest(HALE~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA, mtry = 3, 
predictbag1 <- predict(fit.bag, newdata = trainmainNA, n.trees =5000)
MAPE(trainmainNA$HALE, predictbag1)
```

```
## [1] 0.01995552
```

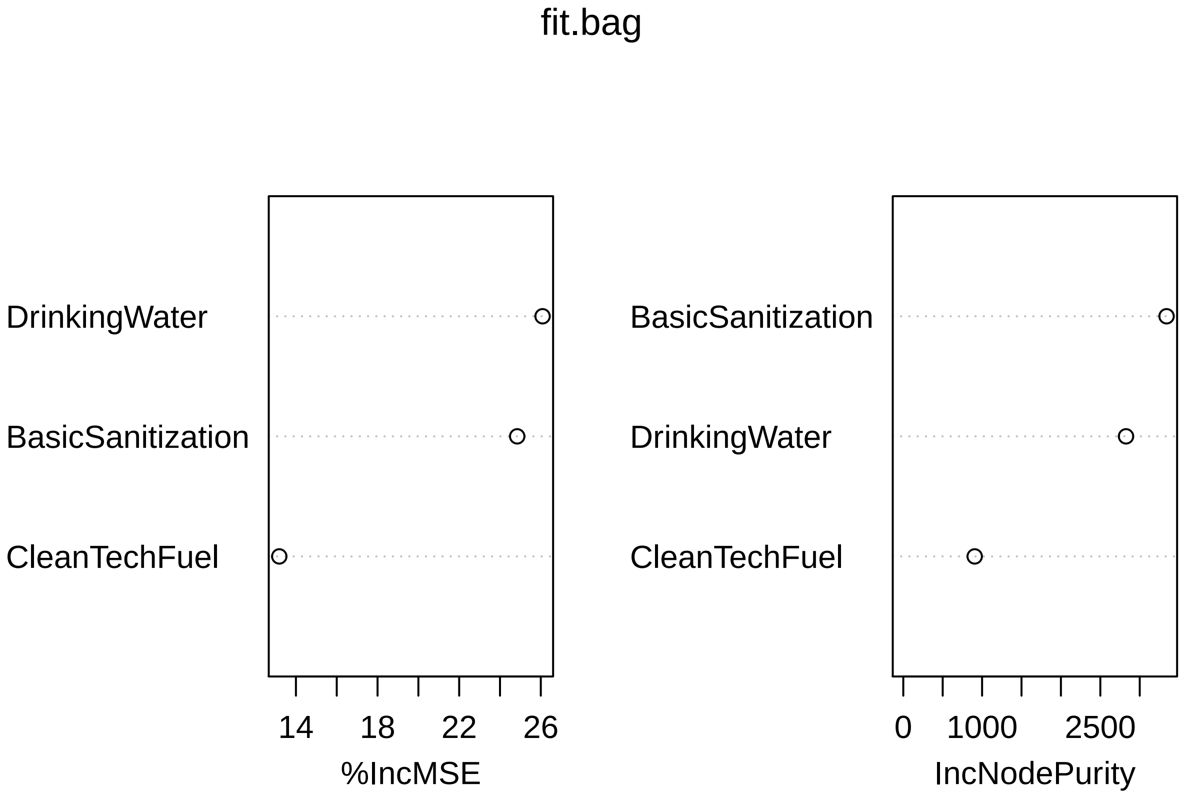
```
predictbag2 = predict(fit.bag, newdata = testmainNA)
MAPE(testmainNA$HALE, predictbag2)
```

```
## [1] 0.02870271
```

```
importance(fit.bag)
```

```
##                     %IncMSE IncNodePurity
## DrinkingWater      26.08583      2826.147
## BasicSanitization 24.84361      3340.507
## CleanTechFuel      13.18410       906.194
```

```
varImpPlot(fit.bag)
```



```
fit.bag <- randomForest(LifeExpectancy~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,
predictbag1 <- predict(fit.bag, newdata = trainmainNA, n.trees =5000)
MAPE(trainmainNA$LifeExpectancy, predictbag1)
```

```
## [1] 0.0200255
```

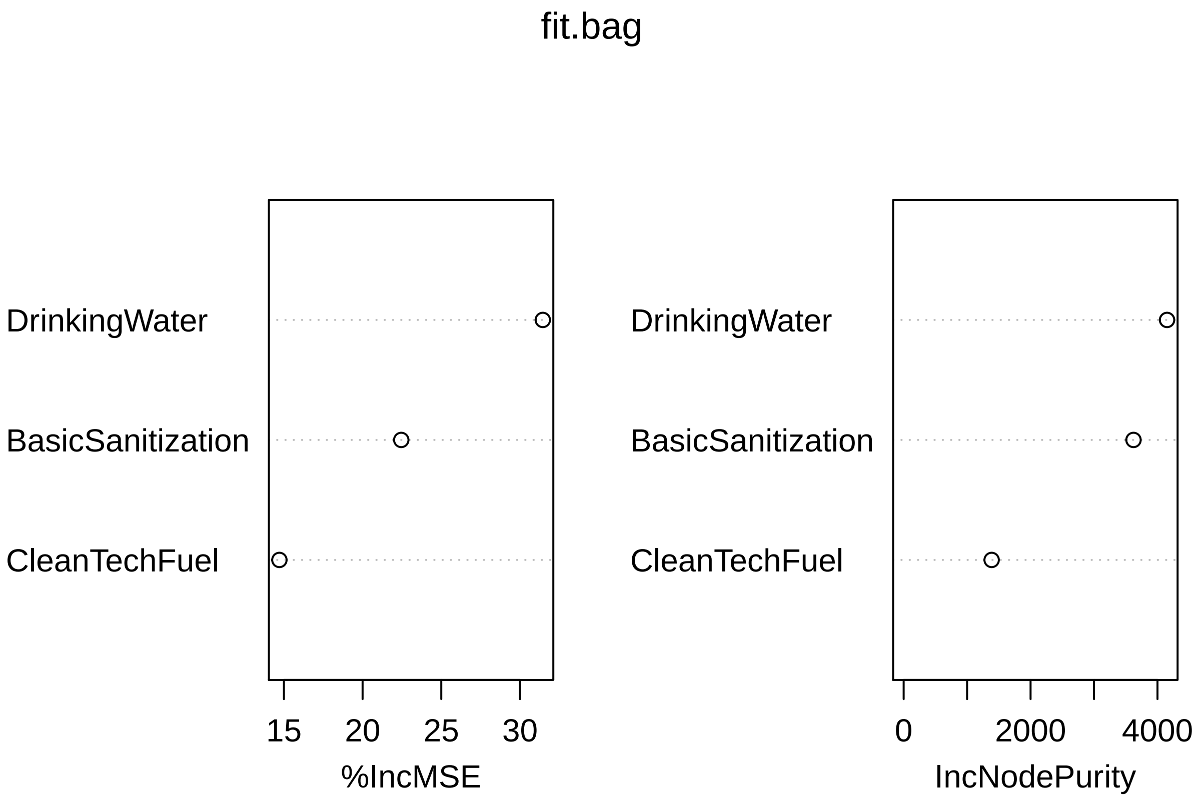
```
predictbag2 = predict(fit.bag, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, predictbag2)
```

```
## [1] 0.02940598
```

```
importance(fit.bag)
```

```
##                     %IncMSE IncNodePurity
## DrinkingWater      31.44957      4151.160
## BasicSanitization 22.45634      3622.392
## CleanTechFuel      14.71367      1386.842
```

```
varImpPlot(fit.bag)
```



The training errors are 1.99% and 1.97% respectively for HALE and Life expectancy There is some difference observed in factors with basic sanitization being more important for HALE than drinking water. For life expectancy at birth, drinking water has the larger effect.

Bagging for Health Behaviours

```
fit.bag <- randomForest(HALE~Alcohol+BMI30Plus+Tobacco,data=trainfullNA, mtry = 3, importance=TRUE) #mt
predictbag1 <- predict(fit.bag, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$HALE, predictbag1)
```

```
## [1] 0.02886825
```

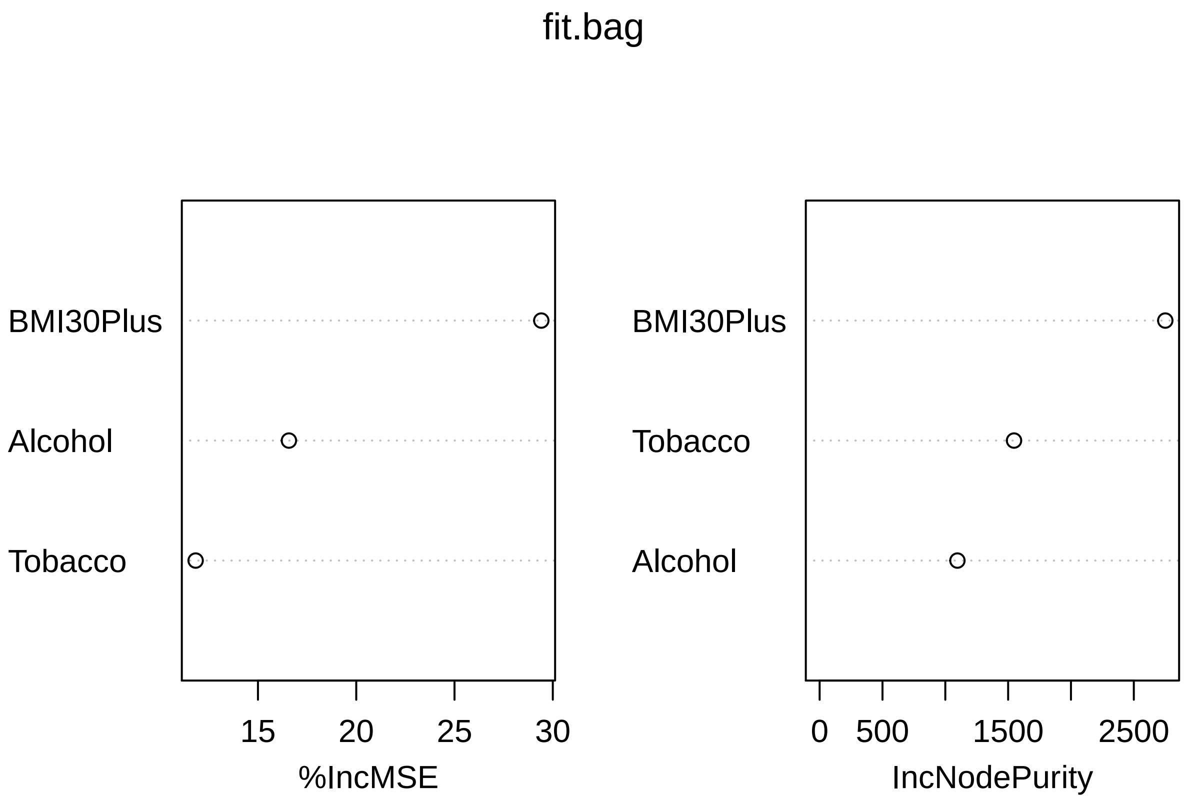
```
predictbag2 = predict(fit.regforest, newdata = testfullNA)
MAPE(testfullNA$HALE, predictbag2)
```

```
## [1] 0.1175942
```

```
importance(fit.bag)
```

```
##             %IncMSE IncNodePurity
## Alcohol    16.57526      1096.359
## BMI30Plus  29.41396      2750.381
## Tobacco    11.82966      1546.792
```

```
varImpPlot(fit.bag)
```



```
fit.bag <- randomForest(LifeExpectancy~Alcohol+BMI30Plus+Tobacco,data=trainfullNA, mtry = 3, importance
predictbag1 <- predict(fit.bag, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$LifeExpectancy, predictbag1)
```

```
## [1] 0.02980362
```

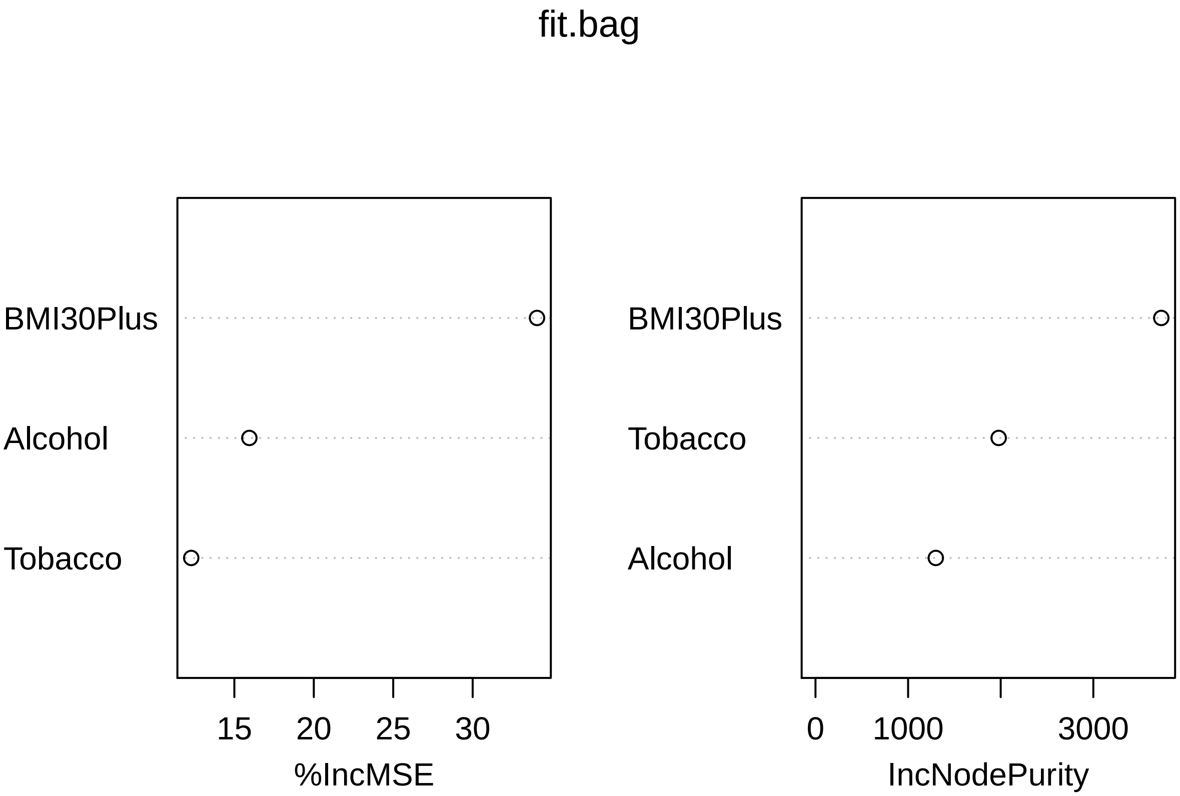
```
predictbag2 = predict(fit.regforest, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, predictbag2)
```

```
## [1] 0.04101268
```

```
importance(fit.bag)
```

```
##              %IncMSE IncNodePurity
## Alcohol    15.94309      1299.138
## BMI30Plus  34.04967      3733.566
## Tobacco    12.28554      1978.028
```

```
varImpPlot(fit.bag)
```



Bagging indicates high BMI to be the most important health behaviour for healthy life expectancy as well as life expectancy at birth but life expectancy at birth is affected more than HALE by substance abuse:alcohol and tobacco.

##Bagging combined model of Health behaviour and Health infrastructure

```
fit.bag <- randomForest(HALE~Alcohol+BMI30Plus+Tobacco+DrinkingWater+BasicSanitization+CleanTechFuel,da
predictbag1 <- predict(fit.bag, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$HALE, predictbag1)
```

```
## [1] 0.01754144
```

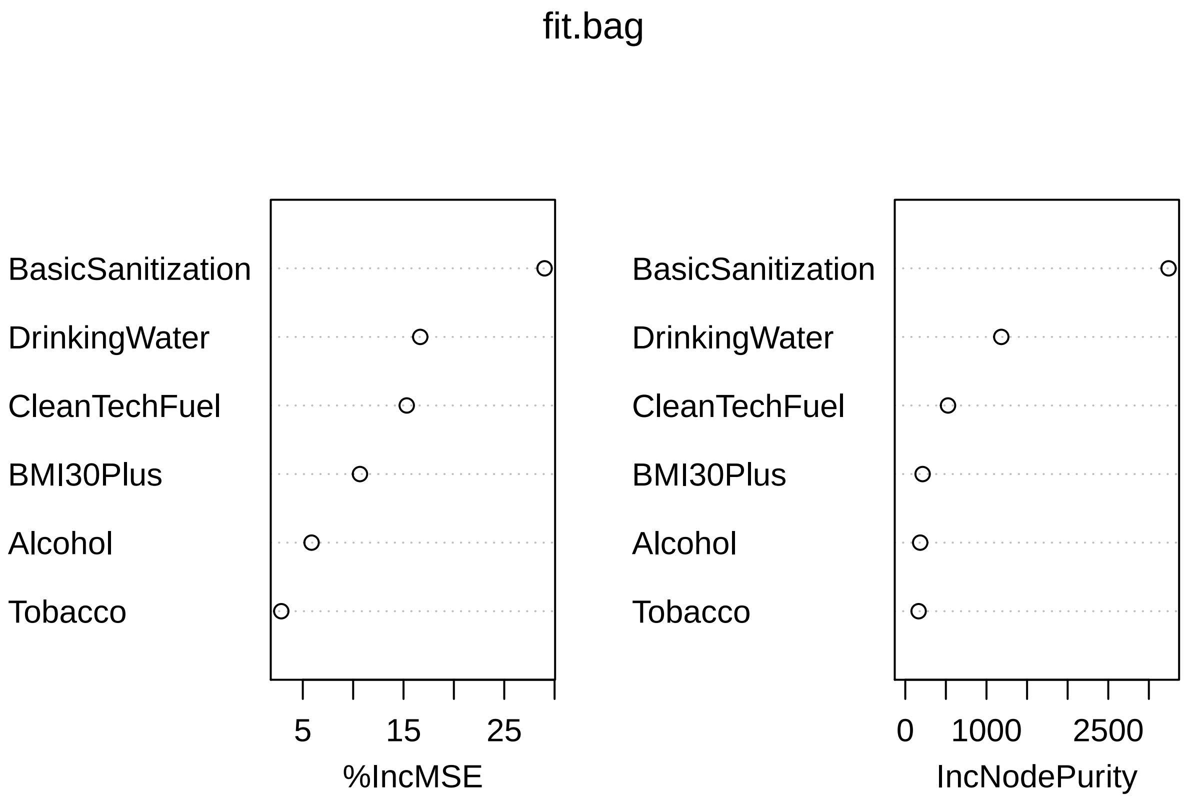
```
predictbag2 = predict(fit.regforest, newdata = testfullNA)
MAPE(testfullNA$HALE, predictbag2)
```

```
## [1] 0.1175942
```

```
importance(fit.bag)
```

```
##                      %IncMSE IncNodePurity
## Alcohol             5.871661      183.7396
## BMI30Plus          10.675212      213.5271
## Tobacco             2.863730      164.3374
## DrinkingWater      16.660288     1182.4606
## BasicSanitization 29.003723     3242.6396
## CleanTechFuel      15.319812      526.3753
```

```
varImpPlot(fit.bag)
```



```
fit.bag <- randomForest(LifeExpectancy~Alcohol+BMI30Plus+Tobacco+DrinkingWater+BasicSanitization+CleanTe
predictbag1 <- predict(fit.bag, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$LifeExpectancy, predictbag1)
```

```
## [1] 0.01807141
```

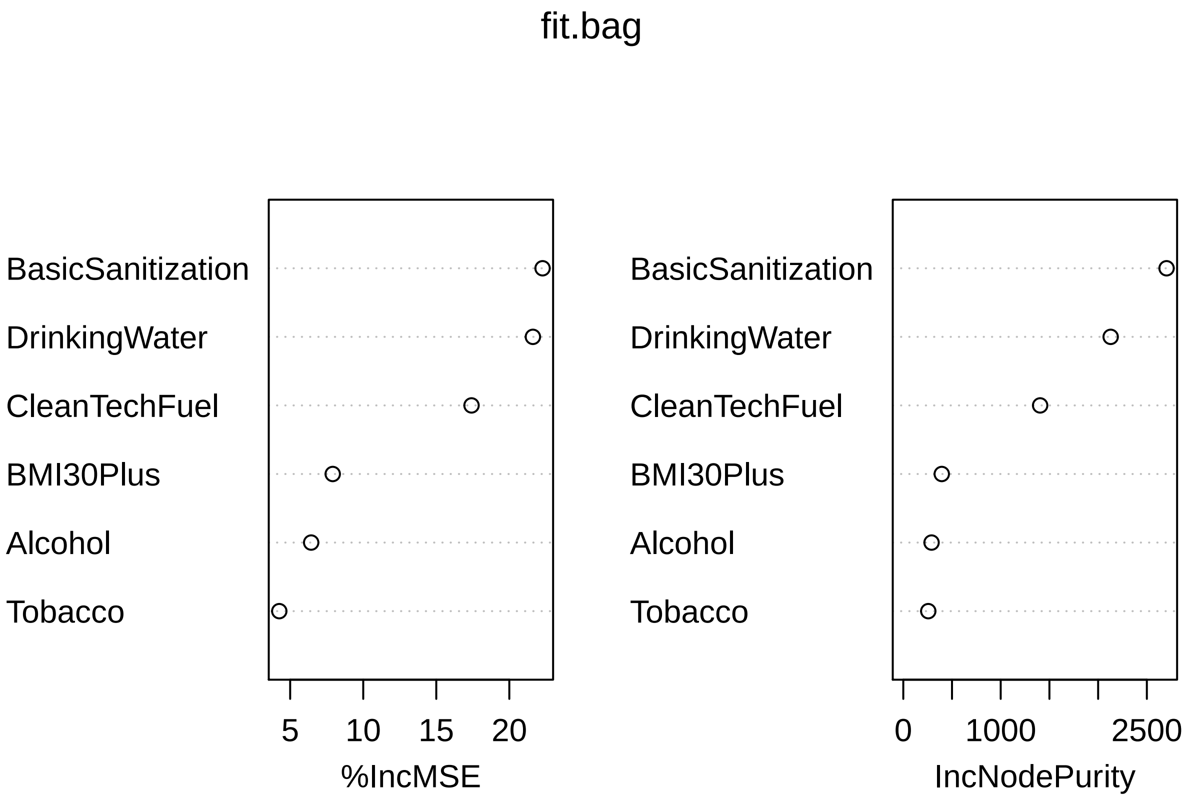
```
predictbag2 = predict(fit.regforest, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, predictbag2)
```

```
## [1] 0.04101268
```

```
importance(fit.bag)
```

```
##                       %IncMSE IncNodePurity
## Alcohol              6.437052      290.2582
## BMI30Plus            7.913214      395.1650
## Tobacco              4.254067      256.6657
## DrinkingWater       21.614951     2128.6124
## BasicSanitization 22.276813     2702.4023
## CleanTechFuel      17.413605     1404.6558
```

```
varImpPlot(fit.bag)
```



When bagging both health infrastructure and behaviours, health infrastructure appears to have the much larger effect with all three parameters being ahead of health behaviour parameters. Basic sanitization has largest impact in infra and obesity has highest impact among behaviours.

##Boosting

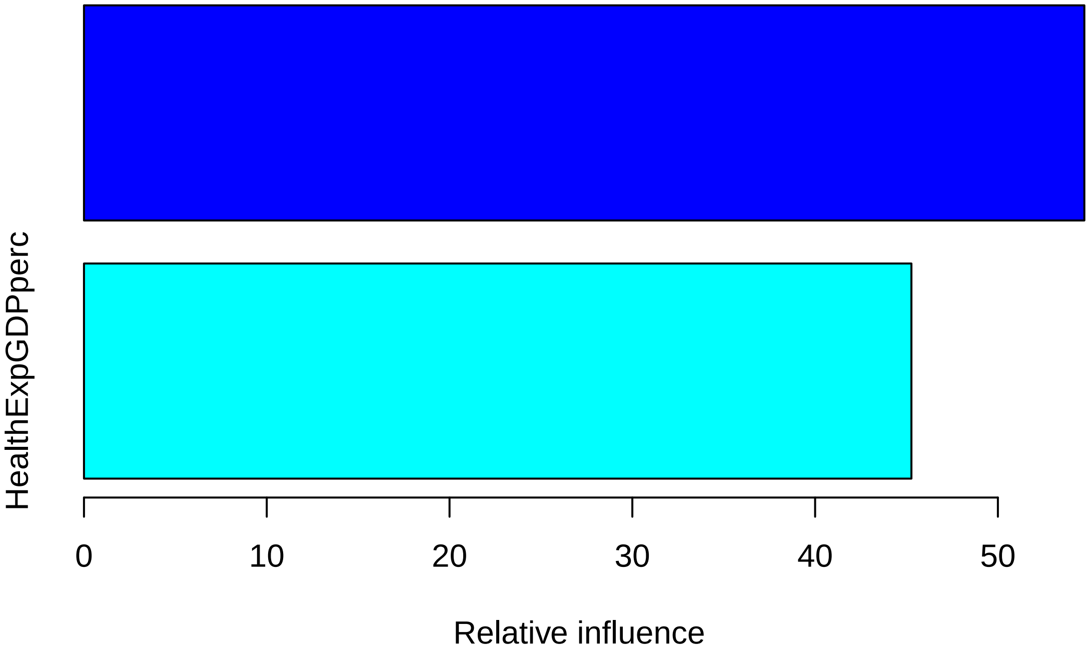
With Economic Data

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.6.3
```

```
## Loaded gbm 2.1.8
```

```
set.seed(8)
fit.boost=gbm(HALE~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA,distribution="gaussian",n.trees=500
summary(fit.boost)
```



```
##                               var  rel.inf
## GDP_currentUSD     GDP_currentUSD 54.73072
## HealthExpGDPperc HealthExpGDPperc 45.26928
```

```
print("Training error is:")
```

```
## [1] "Training error is:"
```

```
predictboost1 <- predict(fit.boost, newdata = econtrainNA, n.trees =5000)
MAPE(econtrainNA$HALE, predictboost1)
```

```
## [1] 0.001357309
```

```
print("Test error is:")
```

```
## [1] "Test error is:"
```

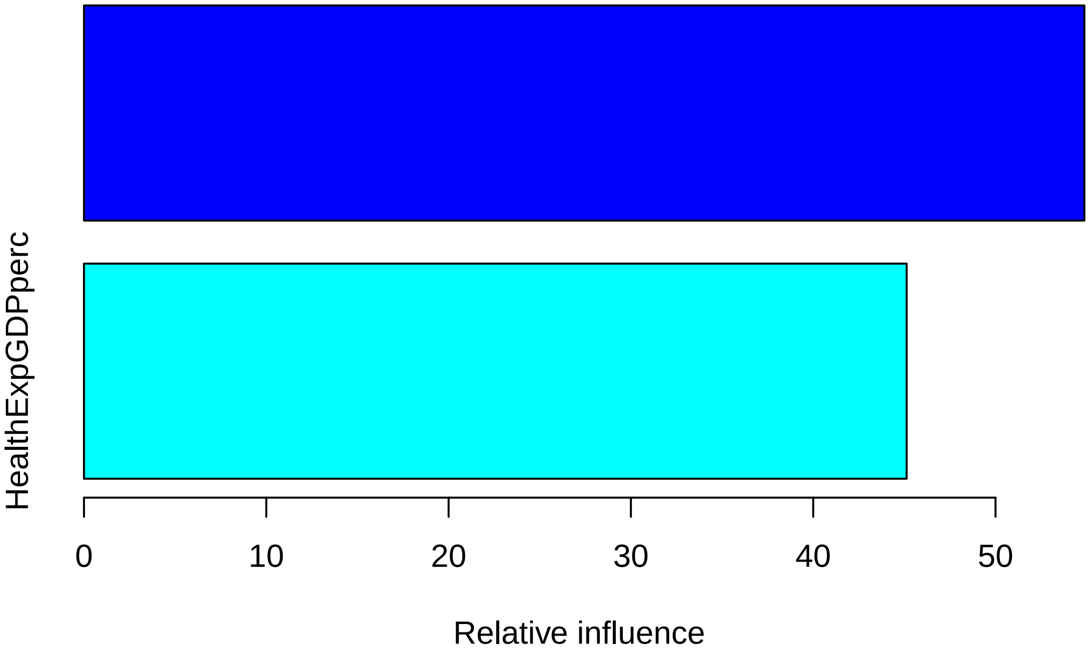
```
predictboost2 = predict(fit.boost, newdata = econtestNA)
```

```
## Using 5000 trees...
```

```
MAPE(econtestNA$HALE, predictboost2)
```

```
## [1] 0.06677522
```

```
fit.boost=gbm(LifeExpectancy~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA,distribution="gaussian",n
summary(fit.boost)
```



```
##                               var  rel.inf
## GDP_currentUSD     GDP_currentUSD 54.87295
## HealthExpGDPperc HealthExpGDPperc 45.12705
```

```
print("Training error is:")
```

```
## [1] "Training error is:"
```

```
predictboost1 <- predict(fit.boost, newdata = econtrainNA, n.trees =5000)
MAPE(econtrainNA$LifeExpectancy, predictboost1)
```

```
## [1] 0.0009883179
```

```
print("Test error is:")
```

```
## [1] "Test error is:"
```

```
predictboost2 = predict(fit.boost, newdata = econtestNA)
```

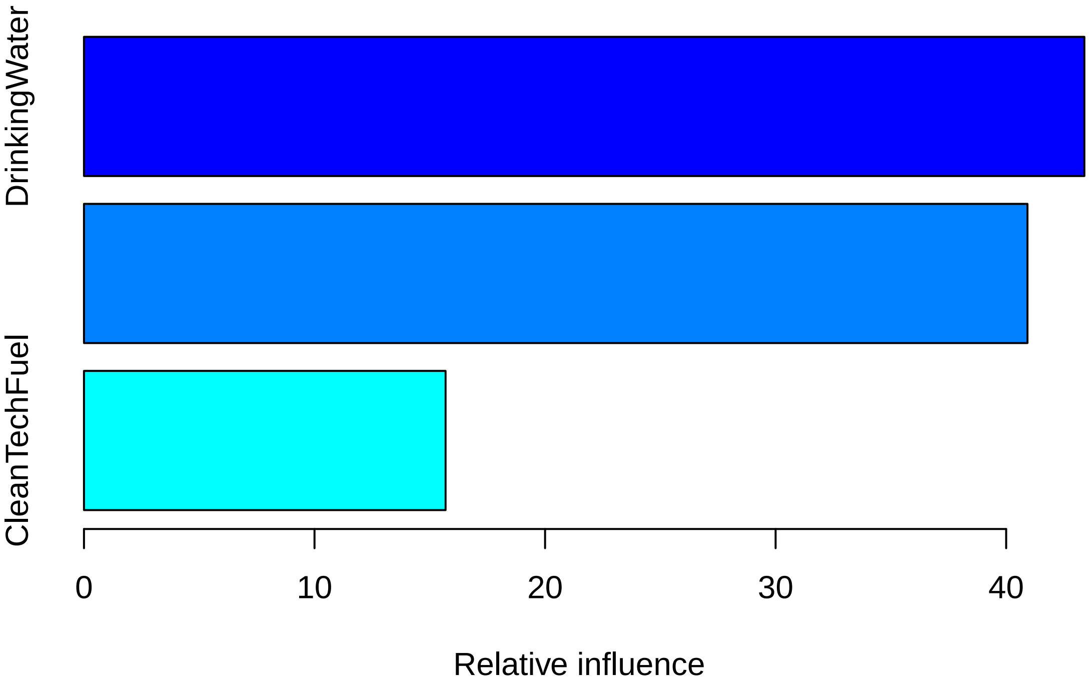
```
## Using 5000 trees...
```

```
MAPE(econtestNA$LifeExpectancy, predictboost2)
```

```
## [1] 0.06622501
```

With Health Infrastructure on both HALE and Life expectancy

```
fit.boost=gbm(HALE~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,distribution="gaussian
summary(fit.boost)
```



```
##                                   var  rel.inf
## DrinkingWater         DrinkingWater 43.39236
## BasicSanitization BasicSanitization 40.92341
## CleanTechFuel         CleanTechFuel 15.68423
```

```
predictboost1 <- predict(fit.boost, newdata = trainmainNA, n.trees =5000)
MAPE(trainmainNA$HALE, predictboost1)
```

```
## [1] 0.002837221
```

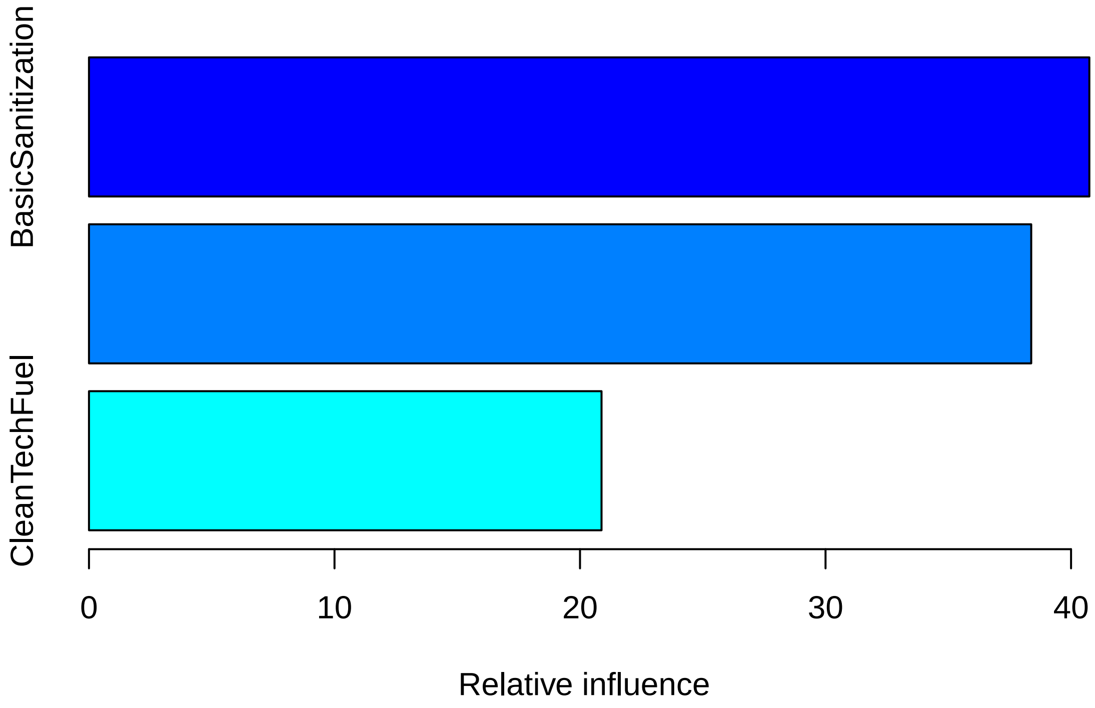
```
predictboost2 = predict(fit.boost, newdata = testmainNA)
```

```
## Using 5000 trees...
```

```
MAPE(testmainNA$HALE, predictboost2)
```

```
## [1] 0.02779887
```

```
fit.boost=gbm(LifeExpectancy~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,distributio
summary(fit.boost)
```



```
##                                   var  rel.inf
## BasicSanitization BasicSanitization 40.74573
## DrinkingWater         DrinkingWater 38.37772
## CleanTechFuel         CleanTechFuel 20.87655
```

```
print("Training error is:")
```

```
## [1] "Training error is:"
```

```
predictboost1 <- predict(fit.boost, newdata = trainmainNA, n.trees =5000)
MAPE(trainmainNA$LifeExpectancy, predictboost1)
```

```
## [1] 0.002486393
```

```
print("Test error is:")
```

```
## [1] "Test error is:"
```

```
predictboost2 = predict(fit.boost, newdata = testmainNA)
```

```
## Using 5000 trees...
```

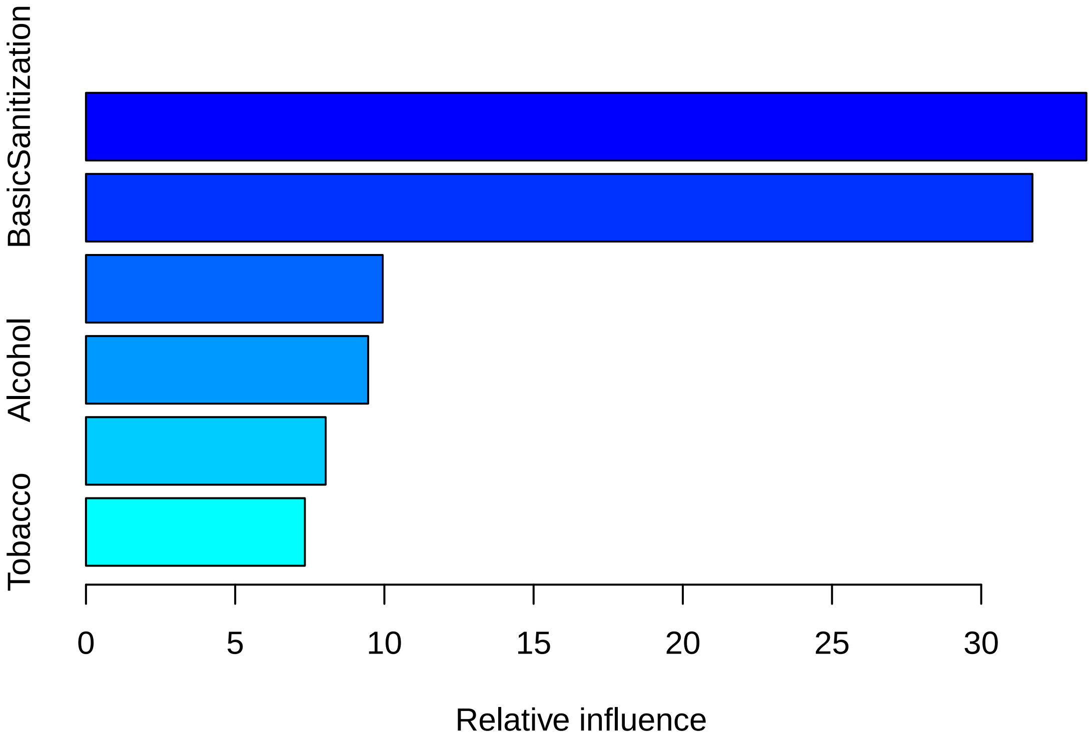
```
MAPE(testmainNA$LifeExpectancy, predictboost2)
```

```
## [1] 0.02880168
```

Basic sanitization is most important followed by drinking water.

With Health Behaviours and Health infrastructure both

```
fit.boost=gbm(HALE~Tobacco+Alcohol+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel,data=trainful
summary(fit.boost)
```



```
##                                 var   rel.inf
## BasicSanitization BasicSanitization 33.523046
## DrinkingWater         DrinkingWater 31.715283
## BMI30Plus                 BMI30Plus  9.941723
## Alcohol                     Alcohol  9.455268
## CleanTechFuel         CleanTechFuel  8.030379
## Tobacco                     Tobacco  7.334301
```

```
print("Training error is:")
```

```
## [1] "Training error is:"
```

```
predictboost1 <- predict(fit.boost, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$HALE, predictboost1)
```

```
## [1] 1.32114e-05
```

```
print("Test error is:")
```

```
## [1] "Test error is:"
```

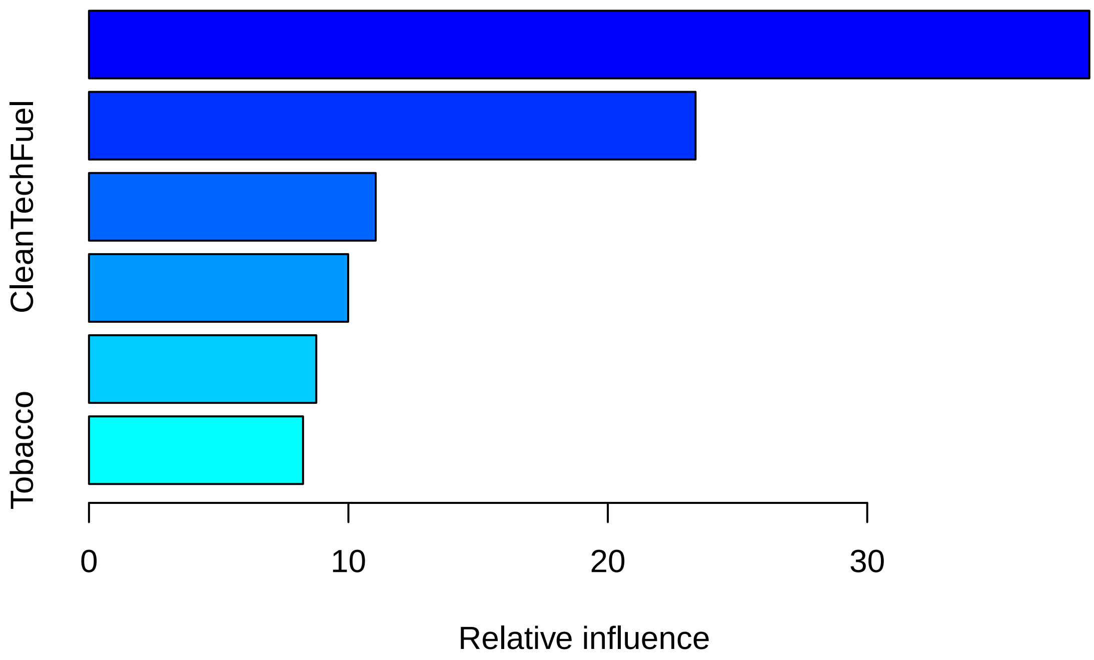
```
predictboost2 = predict(fit.boost, newdata = testfullNA)
```

```
## Using 5000 trees...
```

```
MAPE(testfullNA$HALE, predictboost2)
```

```
## [1] 0.02170539
```

```
fit.boost=gbm(LifeExpectancy~Tobacco+Alcohol+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel,da
summary(fit.boost)
```



```
##                                   var   rel.inf
## BasicSanitization BasicSanitization 38.562254
## DrinkingWater         DrinkingWater 23.381575
## CleanTechFuel         CleanTechFuel 11.051577
## BMI30Plus                 BMI30Plus  9.990556
## Alcohol                     Alcohol  8.762230
## Tobacco                     Tobacco  8.251809
```

```
print("Training error is:")
```

```
## [1] "Training error is:"
```

```
predictboost1 <- predict(fit.boost, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$LifeExpectancy, predictboost1)
```

```
## [1] 7.202812e-06
```

```
print("Test error is:")
```

```
## [1] "Test error is:"
```

```
predictboost2 = predict(fit.boost, newdata = testfullNA)
```

```
## Using 5000 trees...
```

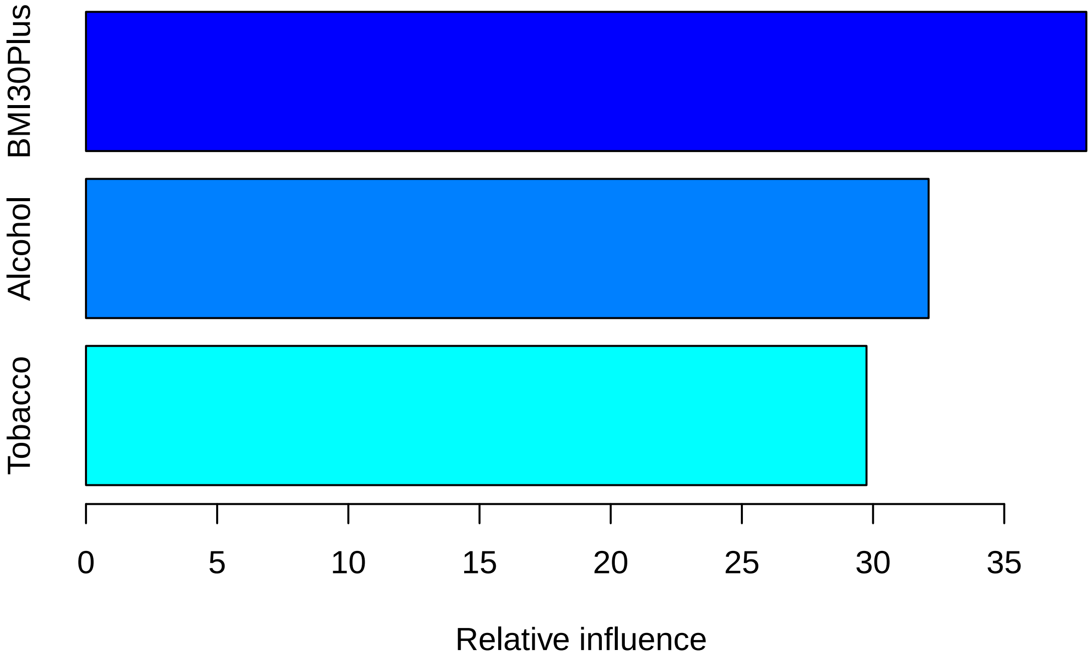
```
MAPE(testfullNA$LifeExpectancy, predictboost2)
```

```
## [1] 0.02249099
```

The effect of health infrastructure is the greatest with basic sanitization and drinking water having the greatest effect. There is some difference observed in the ordering of factors affecting HALE and Life Expectancy but differences are not major. Test errors are only about 2% and training erors are nearly zero.

Health Behaviours

```
fit.boost=gbm(HALE~Tobacco+Alcohol+BMI30Plus,data=trainfullNA,distribution="gaussian",n.trees=5000,inte
summary(fit.boost)
```



```
##                  var rel.inf
## BMI30Plus BMI30Plus 38.1304
## Alcohol     Alcohol 32.1181
## Tobacco     Tobacco 29.7515
```

```
predictboost1 <- predict(fit.boost, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$HALE, predictboost1)
```

```
## [1] 0.0001829155
```

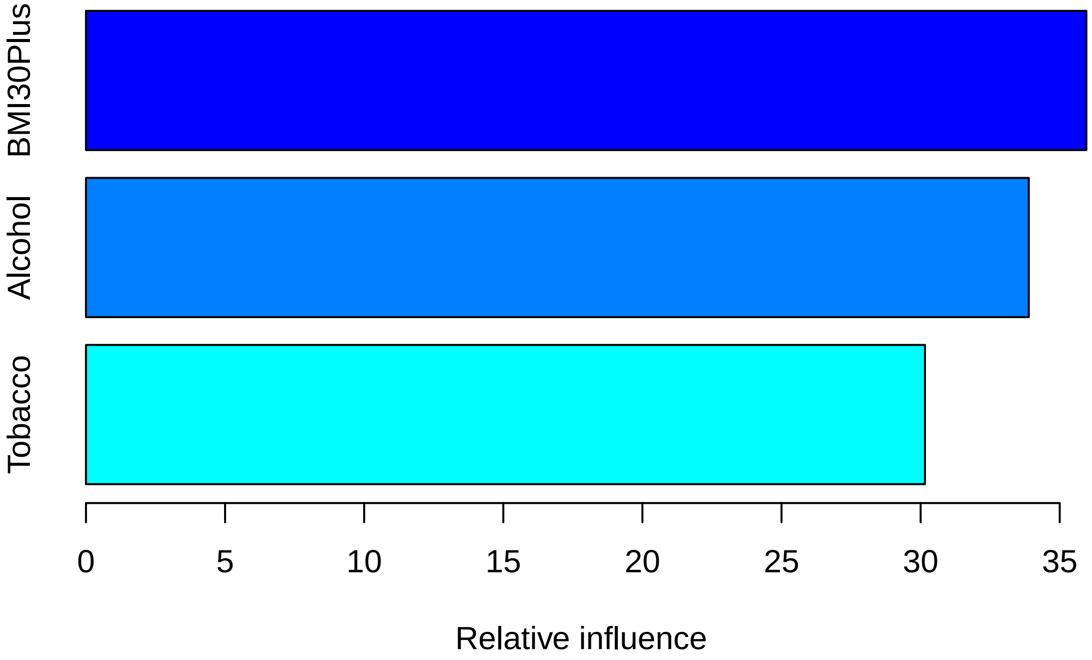
```
predictboost2 = predict(fit.boost, newdata = testfullNA)
```

```
## Using 5000 trees...
```

```
MAPE(testfullNA$HALE, predictboost2)
```

```
## [1] 0.04793637
```

```
fit.boost=gbm(LifeExpectancy~Tobacco+Alcohol+BMI30Plus,data=trainfullNA,distribution="gaussian",n.trees=
summary(fit.boost)
```



```
##                  var  rel.inf
## BMI30Plus BMI30Plus 35.95656
## Alcohol     Alcohol 33.88827
## Tobacco     Tobacco 30.15517
```

```
print("Training error is:")
```

```
## [1] "Training error is:"
```

```
predictboost1 <- predict(fit.boost, newdata = trainfullNA, n.trees =5000)
MAPE(trainfullNA$LifeExpectancy, predictboost1)
```

```
## [1] 0.000186372
```

```
print("Test error is:")
```

```
## [1] "Test error is:"
```

```
predictboost2 = predict(fit.boost, newdata = testfullNA)
```

```
## Using 5000 trees...
```

```
MAPE(testfullNA$LifeExpectancy, predictboost2)
```

```
## [1] 0.0499974
```

Boosting gives excellent performance with very low error rates.Obesity and alcohol are most important behaviours.

##Support Vector machines

####For Infrastructure for both HALE and Life Expectancy

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.6.3
```

```
set.seed(11)
# Fit support vector machine with linear kernel for HALE
svm.model=svm(HALE~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,kernel="linear",cost=0
summary(svm.model)
```

```
##
## Call:
## svm(formula = HALE ~ DrinkingWater + BasicSanitization + CleanTechFuel,
##     data = trainmainNA, kernel = "linear", cost = 0.38)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.38
##       gamma:  0.3333333
##     epsilon:  0.1
##
##
## Number of Support Vectors:  134
```

```
#We re ran at 0.38 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = trainmainNA)
MAPE(trainmainNA$HALE, prediction)
```

```
## [1] 0.04379204
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = testmainNA)
MAPE(testmainNA$HALE, prediction)
```

```
## [1] 0.04192901
```

```
# Using Tune to optimize cost

tune.out=tune(svm,HALE~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,kernel="linear",ra
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.38
##
## - best performance: 13.49622
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 13.97681   7.183852
## 2   0.38 13.49622   6.369271
## 3   0.75 13.51988   6.346154
## 4   1.12 13.57577   6.301740
## 5   1.49 13.60779   6.298465
## 6   1.86 13.62060   6.291752
## 7   2.23 13.62213   6.290890
## 8   2.60 13.62303   6.291633
## 9   2.97 13.61942   6.290774
## 10  3.34 13.62169   6.290899
## 11  3.71 13.62198   6.292537
## 12  4.08 13.62222   6.291407
## 13  4.45 13.62033   6.291126
## 14  4.82 13.62008   6.289560
## 15  5.19 13.62072   6.290556
## 16  5.56 13.62039   6.290496
## 17  5.93 13.62180   6.291586
## 18  6.30 13.61953   6.290627
## 19  6.67 13.62096   6.290459
## 20  7.04 13.62087   6.289530
## 21  7.41 13.62235   6.292413
## 22  7.78 13.62174   6.291758
## 23  8.15 13.62036   6.290982
## 24  8.52 13.62176   6.293259
## 25  8.89 13.62033   6.291038
## 26  9.26 13.62100   6.291912
## 27  9.63 13.62123   6.291050
## 28 10.00 13.61943   6.290525
```

```
#The lowest error rate is at cost of 0.38

# Fit support vector machine with linear kernel for Life Expectancy
svm.model=svm(LifeExpectancy~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,kernel="lin
summary(svm.model)
```

```
##
## Call:
## svm(formula = LifeExpectancy ~ DrinkingWater + BasicSanitization +
##     CleanTechFuel, data = trainmainNA, kernel = "linear", cost = 0.75)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.75
##       gamma:  0.3333333
##     epsilon:  0.1
##
##
## Number of Support Vectors:  130
```

```
#We re ran at 0.75 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = trainmainNA)
MAPE(trainmainNA$LifeExpectancy, prediction)
```

```
## [1] 0.0443656
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = testmainNA)
MAPE(testmainNA$LifeExpectancy, prediction)
```

```
## [1] 0.04333485
```

```
# Using Tune to optimize cost

tune.out=tune(svm,LifeExpectancy~DrinkingWater+BasicSanitization+CleanTechFuel,data=trainmainNA,kernel=
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  2.97
##
## - best performance: 17.56622
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 18.64264   8.075354
## 2   0.38 17.57425   7.364368
## 3   0.75 17.57211   7.354233
## 4   1.12 17.57379   7.356125
## 5   1.49 17.57509   7.355997
## 6   1.86 17.57490   7.355958
## 7   2.23 17.57447   7.356581
## 8   2.60 17.57298   7.353900
## 9   2.97 17.56622   7.351501
## 10  3.34 17.57451   7.346318
## 11  3.71 17.57191   7.345389
## 12  4.08 17.57441   7.344088
## 13  4.45 17.57383   7.344476
## 14  4.82 17.57298   7.345342
## 15  5.19 17.57305   7.343903
## 16  5.56 17.57558   7.344909
## 17  5.93 17.57672   7.343359
## 18  6.30 17.57426   7.344768
## 19  6.67 17.57293   7.345337
## 20  7.04 17.57512   7.347081
## 21  7.41 17.57405   7.343908
## 22  7.78 17.57837   7.340417
## 23  8.15 17.58160   7.340922
## 24  8.52 17.58061   7.334912
## 25  8.89 17.58659   7.330782
## 26  9.26 17.58893   7.328893
## 27  9.63 17.58954   7.329059
## 28 10.00 17.58922   7.335010
```

```
#The lowest error rate is at cost of 0.75
```

The training and test error rates fall in similar ranges and are slightly lower for HALE than Life Expectancy. At less than 4.5%, the prediction performance is good.

Using SVM for Health Behaviours

```
set.seed(11)
# Fit support vector machine with linear kernel for HALE
svm.model=svm(HALE~Tobacco+Alcohol+BMI30Plus,data=trainfullNA,kernel="linear",cost=0.01)
summary(svm.model)
```

```
##
## Call:
## svm(formula = HALE ~ Tobacco + Alcohol + BMI30Plus, data = trainfullNA,
##     kernel = "linear", cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.01
##       gamma:  0.3333333
##     epsilon:  0.1
##
##
## Number of Support Vectors:  115
```

```
#We re ran at 0.01 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = trainfullNA)
MAPE(trainfullNA$HALE, prediction)
```

```
## [1] 0.06971924
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = testfullNA)
MAPE(testfullNA$HALE, prediction)
```

```
## [1] 0.06384669
```

```
# Using Tune to optimize cost

tune.out=tune(svm,HALE~Tobacco+Alcohol+BMI30Plus,data=trainfullNA,kernel="linear",ranges=list(cost=seq(
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.01
##
## - best performance: 36.05617
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 36.05617   21.24464
## 2   0.38 37.06896   27.96191
## 3   0.75 37.08766   27.98062
## 4   1.12 37.10541   27.99329
## 5   1.49 37.15755   28.15068
## 6   1.86 37.19062   28.18733
## 7   2.23 37.19454   28.18784
## 8   2.60 37.18557   28.18767
## 9   2.97 37.19427   28.18762
## 10  3.34 37.22842   28.18542
## 11  3.71 37.24510   28.21511
## 12  4.08 37.24164   28.21591
## 13  4.45 37.24440   28.21827
## 14  4.82 37.23629   28.22007
## 15  5.19 37.24011   28.22257
## 16  5.56 37.24744   28.21317
## 17  5.93 37.22745   28.18063
## 18  6.30 37.25340   28.16514
## 19  6.67 37.25665   28.16427
## 20  7.04 37.26913   28.20014
## 21  7.41 37.25314   28.16026
## 22  7.78 37.26932   28.20010
## 23  8.15 37.25242   28.16033
## 24  8.52 37.25165   28.15866
## 25  8.89 37.26980   28.19872
## 26  9.26 37.24863   28.16889
## 27  9.63 37.27485   28.19810
## 28 10.00 37.25787   28.16396
```

```
#The lowest error rate is at cost of 0.01

# Fit support vector machine with linear kernel for Life Expectancy
svm.model=svm(LifeExpectancy~Tobacco+Alcohol+BMI30Plus,data=trainfullNA,kernel="linear",cost=0.01)
summary(svm.model)
```

```
##
## Call:
## svm(formula = LifeExpectancy ~ Tobacco + Alcohol + BMI30Plus,
##     data = trainfullNA, kernel = "linear", cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.01
##       gamma:  0.3333333
##     epsilon:  0.1
##
##
## Number of Support Vectors:  116
```

```
#We re ran at 0.01 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = trainfullNA)
MAPE(trainfullNA$LifeExpectancy, prediction)
```

```
## [1] 0.07151636
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, prediction)
```

```
## [1] 0.06648637
```

```
# Using Tune to optimize cost

tune.out=tune(svm,LifeExpectancy~Tobacco+Alcohol+BMI30Plus,data=trainfullNA,kernel="linear",ranges=list
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.01
##
```

```
## - best performance: 46.29686
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 46.29686   17.80818
## 2   0.38 47.31245   21.50518
## 3   0.75 47.40083   21.58319
## 4   1.12 47.40199   21.56659
## 5   1.49 47.39449   21.56732
## 6   1.86 47.34105   21.59472
## 7   2.23 47.34038   21.59969
## 8   2.60 47.34076   21.59845
## 9   2.97 47.33754   21.59990
## 10  3.34 47.33948   21.60084
## 11  3.71 47.34241   21.60778
## 12  4.08 47.33798   21.59949
## 13  4.45 47.33994   21.59633
## 14  4.82 47.33709   21.60663
## 15  5.19 47.34793   21.59458
## 16  5.56 47.34309   21.59538
## 17  5.93 47.33707   21.59811
## 18  6.30 47.34517   21.59785
## 19  6.67 47.34432   21.60427
## 20  7.04 47.33994   21.60235
## 21  7.41 47.34236   21.60087
## 22  7.78 47.34235   21.59485
## 23  8.15 47.34720   21.59954
## 24  8.52 47.34023   21.60102
## 25  8.89 47.34248   21.60280
## 26  9.26 47.33757   21.59412
## 27  9.63 47.34116   21.60282
## 28 10.00 47.34585   21.59746
```

```
#The lowest error rate is at cost of 0.01
```

Training error rate for Health Behaviours was worse than for health infrastructure. HALE was predicted far more accurately at 6.97% and 6.38% error rates for training and test sets respectively. Life expectancy was predicted slightly less accurately at 7.15% and 6.65% error rates.

SVM For both health behaviours and health infrastructure

```
set.seed(11)
# Fit support vector machine with linear kernel for HALE
svm.model=svm(HALE~Tobacco+Alcohol+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel,data=trainful
summary(svm.model)
```

```
##
## Call:
## svm(formula = HALE ~ Tobacco + Alcohol + BMI30Plus + DrinkingWater +
##     BasicSanitization + CleanTechFuel, data = trainfullNA, kernel = "linear",
##     cost = 0.75)
##
```

```
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.75
##       gamma:  0.1666667
##     epsilon:  0.1
##
##
## Number of Support Vectors:  109
```

```
#We re ran at 0.75 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = trainfullNA)
MAPE(trainfullNA$HALE, prediction)
```

```
## [1] 0.04006474
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = testfullNA)
MAPE(testfullNA$HALE, prediction)
```

```
## [1] 0.03890183
```

```
# Using Tune to optimize cost

tune.out=tune(svm,HALE~Tobacco+Alcohol+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel,data=tra
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.75
##
## - best performance: 12.62755
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 14.41799   9.393911
## 2   0.38 12.73362   6.611510
## 3   0.75 12.62755   6.644336
## 4   1.12 12.63336   6.640022
## 5   1.49 12.65455   6.634878
## 6   1.86 12.69740   6.625419
## 7   2.23 12.70567   6.630138
## 8   2.60 12.73932   6.628379
## 9   2.97 12.75807   6.658646
## 10  3.34 12.73803   6.628652
## 11  3.71 12.74506   6.650157
## 12  4.08 12.73997   6.632636
## 13  4.45 12.74411   6.650231
## 14  4.82 12.74762   6.648800
## 15  5.19 12.73777   6.630198
## 16  5.56 12.73562   6.627607
## 17  5.93 12.73715   6.628497
## 18  6.30 12.74911   6.627169
## 19  6.67 12.74773   6.641609
## 20  7.04 12.73840   6.627519
## 21  7.41 12.74849   6.651134
## 22  7.78 12.75863   6.649273
## 23  8.15 12.74827   6.651106
## 24  8.52 12.74566   6.642174
## 25  8.89 12.73754   6.643913
## 26  9.26 12.74406   6.634675
## 27  9.63 12.74179   6.651634
## 28 10.00 12.73958   6.629885
```

```
#The lowest error rate is at cost of 0.75

# Fit support vector machine with linear kernel for Life Expectancy
svm.model=svm(LifeExpectancy~Tobacco+Alcohol+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel,da
summary(svm.model)
```

```
##
## Call:
## svm(formula = LifeExpectancy ~ Tobacco + Alcohol + BMI30Plus +
##     DrinkingWater + BasicSanitization + CleanTechFuel, data = trainfullNA,
##     kernel = "linear", cost = 1.86)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  1.86
##       gamma:  0.1666667
##     epsilon:  0.1
##
##
## Number of Support Vectors:  107
```

```
#We re ran at 1.86instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = trainfullNA)
MAPE(trainfullNA$LifeExpectancy, prediction)
```

```
## [1] 0.04108495
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = testfullNA)
MAPE(testfullNA$LifeExpectancy, prediction)
```

```
## [1] 0.04005858
```

```
# Using Tune to optimize cost

tune.out=tune(svm,LifeExpectancy~Tobacco+Alcohol+BMI30Plus+DrinkingWater+BasicSanitization+CleanTechFuel
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  1.86
##
## - best performance: 16.71837
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 18.80437   9.359197
## 2   0.38 16.83007   9.951975
## 3   0.75 16.79952   9.908013
## 4   1.12 16.77619   9.878161
## 5   1.49 16.71858   9.820444
## 6   1.86 16.71837   9.814768
## 7   2.23 16.72719   9.809814
## 8   2.60 16.72158   9.817571
## 9   2.97 16.72291   9.812676
## 10  3.34 16.73253   9.808527
## 11  3.71 16.73458   9.800761
## 12  4.08 16.73787   9.793946
## 13  4.45 16.74194   9.785486
## 14  4.82 16.75140   9.780914
## 15  5.19 16.74956   9.779239
## 16  5.56 16.75896   9.778899
## 17  5.93 16.76120   9.771288
## 18  6.30 16.76569   9.764130
## 19  6.67 16.77368   9.758483
## 20  7.04 16.77846   9.752381
## 21  7.41 16.78454   9.744183
## 22  7.78 16.78532   9.742900
## 23  8.15 16.78823   9.740295
## 24  8.52 16.78774   9.745158
## 25  8.89 16.78683   9.743211
## 26  9.26 16.78796   9.742821
## 27  9.63 16.78709   9.742086
## 28 10.00 16.78698   9.741750
```

```
#The lowest error rate is at cost of 1.86
```

SVM for Economic Factors

```
# Fit support vector machine with linear kernel for HALE
svm.model=svm(HALE~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA,kernel="linear",cost=0.01)
summary(svm.model)
```

```
##
## Call:
## svm(formula = HALE ~ GDP_currentUSD + HealthExpGDPperc, data = econtrainNA,
##     kernel = "linear", cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.01
##       gamma:  0.5
##     epsilon:  0.1
##
##
## Number of Support Vectors:  161
```

```
#We re ran at 0.01 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```
prediction = predict(svm.model, newdata = econtrainNA)
MAPE(econtrainNA$HALE, prediction)
```

```
## [1] 0.07584037
```

```
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```
prediction = predict(svm.model, newdata = econtestNA)
MAPE(econtestNA$HALE, prediction)
```

```
## [1] 0.07033002
```

```
# Using Tune to optimize cost

tune.out=tune(svm,HALE~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA,kernel="linear",ranges=list(cos
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.01
##
## - best performance: 45.38828
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 45.38828   21.48089
## 2   0.38 49.02963   30.37343
## 3   0.75 49.33372   30.28951
## 4   1.12 49.48148   30.38611
## 5   1.49 49.47601   30.37978
## 6   1.86 49.47669   30.39341
## 7   2.23 49.47859   30.40289
## 8   2.60 49.43504   30.39360
## 9   2.97 49.43539   30.40202
## 10  3.34 49.42993   30.38165
## 11  3.71 49.47192   30.48755
## 12  4.08 49.46510   30.47059
## 13  4.45 49.46887   30.47900
## 14  4.82 49.48123   30.49996
## 15  5.19 49.47510   30.48334
## 16  5.56 49.46478   30.46297
## 17  5.93 49.40948   30.50976
## 18  6.30 49.40348   30.49916
## 19  6.67 49.40760   30.49867
## 20  7.04 49.40965   30.51317
## 21  7.41 49.39903   30.48432
## 22  7.78 49.40859   30.50710
## 23  8.15 49.41112   30.49271
## 24  8.52 49.41178   30.51403
## 25  8.89 49.39812   30.48577
## 26  9.26 49.40483   30.50661
## 27  9.63 49.41527   30.50882
## 28 10.00 49.41990   30.52115
```

```r
#The lowest error rate is at cost of 0.01

# Fit support vector machine with linear kernel for Life Expectancy
svm.model=svm(LifeExpectancy~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA,kernel="linear",cost=0.01
summary(svm.model)
```

```
##
## Call:
## svm(formula = LifeExpectancy ~ GDP_currentUSD + HealthExpGDPperc,
##     data = econtrainNA, kernel = "linear", cost = 0.01)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  0.01
##       gamma:  0.5
##     epsilon:  0.1
##
##
## Number of Support Vectors:  159
```

```r
#We re ran at 0.38 instead of initial 0.1 based on tune.out results

print("Training error rate is")
```

```
## [1] "Training error rate is"
```

```r
prediction = predict(svm.model, newdata = econtrainNA)
MAPE(econtrainNA$LifeExpectancy, prediction)
```

```
## [1] 0.07665
```

```r
print("Test error rate is")
```

```
## [1] "Test error rate is"
```

```r
prediction = predict(svm.model, newdata = econtestNA)
MAPE(econtestNA$LifeExpectancy, prediction)
```

```
## [1] 0.07168694
```

```r
# Using Tune to optimize cost

tune.out=tune(svm,LifeExpectancy~GDP_currentUSD+HealthExpGDPperc,data=econtrainNA,kernel="linear",range
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
```

```
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##  0.01
##
## - best performance: 57.64903
##
## - Detailed performance results:
##     cost    error dispersion
## 1   0.01 57.64903   34.23757
## 2   0.38 63.11707   45.61558
## 3   0.75 63.35743   45.99207
## 4   1.12 63.35671   45.99717
## 5   1.49 63.35469   45.98707
## 6   1.86 63.31280   46.00475
## 7   2.23 63.30542   46.00242
## 8   2.60 63.31348   46.00116
## 9   2.97 63.31353   45.99930
## 10  3.34 63.31225   45.99766
## 11  3.71 63.32218   46.01810
## 12  4.08 63.31257   45.99774
## 13  4.45 63.31420   46.00016
## 14  4.82 63.31008   45.99315
## 15  5.19 63.30782   45.99531
## 16  5.56 63.31613   46.00506
## 17  5.93 63.30913   46.00192
## 18  6.30 63.31367   45.99919
## 19  6.67 63.31603   45.99621
## 20  7.04 63.31374   45.98941
## 21  7.41 63.32346   46.00694
## 22  7.78 63.31834   46.00021
## 23  8.15 63.31608   45.99059
## 24  8.52 63.32104   45.99859
## 25  8.89 63.32045   45.99875
## 26  9.26 63.32138   46.00086
## 27  9.63 63.32116   46.00003
## 28 10.00 63.32585   46.00145
```

```
#The lowest error rate is at cost of 0.01
```

Though the economic data works better with SVM than linear regressuion, the error rates are still higher than for health infrastructure and behaviour.

Clustering

###K-Means Clustering

```
vec=c("cyan","red","black","purple","green","black","gold","orange","pink","purple")
trainfullNA$Location<-NULL
km.out=kmeans(trainfullNA,10,nstart=20)
table(km.out$cluster)
```

```
##
##  1  2  3  4  5  6  7  8  9 10
##  6  2 73  9  3  7  8 11  4  1
```

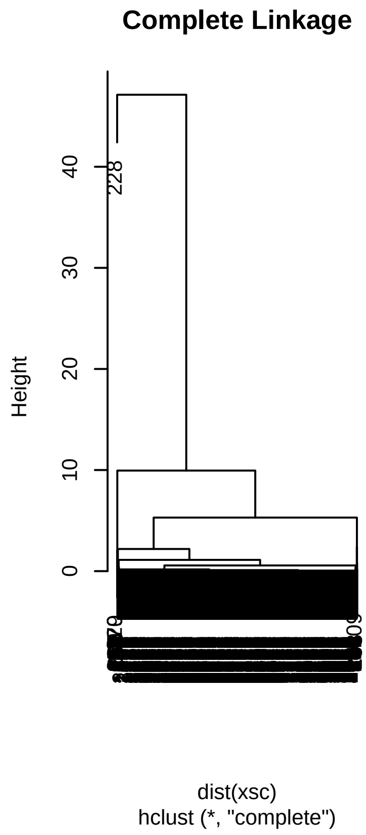
```
plot(trainfullNA,col=vec,main="K-Means clustering results with K=10",pch=12,cex=3)
```

Hierarchial

```
alka=as.numeric(unlist(trainmainNA))
xsc=scale(alka)
hc.complete=hclust(dist(xsc),method="complete")
par(mfrow=c(1,3))
plot(hc.complete,main="Complete Linkage",cex=1)
summary(hc.complete)
```

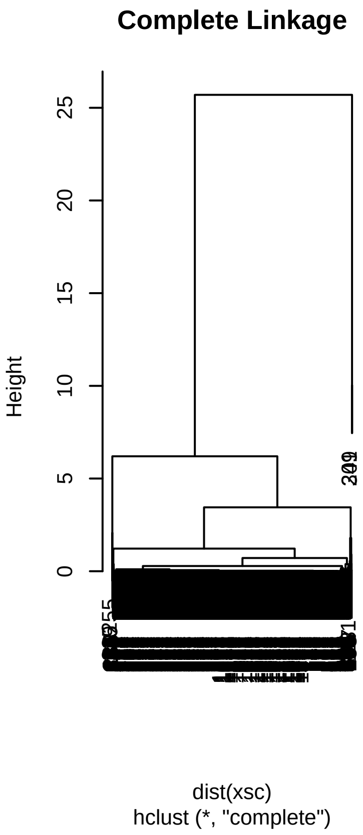
```
##             Length Class  Mode
## merge       4958   -none- numeric
## height      2479   -none- numeric
## order       2480   -none- numeric
## labels         0   -none- NULL
## method         1   -none- character
## call           3   -none- call
## dist.method    1   -none- character
```

```
alka=as.numeric(unlist(econtrainNA))
xsc=scale(alka)
hc.complete=hclust(dist(xsc),method="complete")
par(mfrow=c(1,3))
```



```
plot(hc.complete,main="Complete Linkage",cex=1)
summary(hc.complete)
```

```
##             Length Class  Mode
## merge       2086   -none- numeric
## height      1043   -none- numeric
## order       1044   -none- numeric
## labels         0   -none- NULL
## method         1   -none- character
## call           3   -none- call
## dist.method    1   -none- character
```


Complete Linkage
Height
0
5
10
15
20
25
255
201
dist(xsc)
hclust (*, "complete")