Model Building & Analysis

Ellie Bi & Terrie Kim

Introduction:

The purpose of this project is to determine what affects life expectancy in countries the most. Given data about a country's top statistics, our goal is to find the best predictors by building a model that accurately predicts life expectancy. Through the process of model building – data preparation, data exploration, multicollinearity, model selection, model refinement, and assumption verification – we want to determine the collective impact of these factors and to determine which of these factors are the most important in understanding life expectancy. By using a data set that takes in a spectrum of critical variables, ranging from demographic indicators to environmental factors, our primary objective is to identify the key influences on life expectancy.

Data Preparation:

The data set we used was on country data, which included data about a country's land area, population, rural data, health data, internet data, birth rate, elderly population CO2, GDP, cell data, and life expectancy.

Since the original data set had two qualitative variables, country name and code, we removed those variables to prevent any discrepancies in our model selection process. We continued to fit a model using life expectancy as the outcome, and all other variables as predictors.

Data Exploration – Residual Analysis:

After we finished preparing the data, we then determined which variables needed to be transformed to better fit our model selection. We performed residual analysis by obtaining the relevant graphs: residual plots and Q-Q plot.

Based on the plots, we needed to transform life expectancy because all Q-Q plots indicated that the assumption of normality was violated. In almost all of the Residuals vs Fitted plots, the assumption of linearity as well as the assumption of equal and constant variances were violated. This meant a transformation of the predictors was necessary. The only predictor that did not violate the assumption of linearity and constant variance was the health variable.

To transform Y (life expectancy), we used the Box-Cox method to determine an appropriate method of transformation. Upon obtaining the Box-Cox plot, we observed that the lambda value

was much larger than 0, indicating that an appropriate transformation is to use the equation $\frac{Y^{\lambda}}{\lambda-1}$ to properly transform the outcome. Given the substantially large lambda value, it reinforced our decision to opt for the $\frac{Y^{\lambda}}{\lambda-1}$ equation as a means of appropriately transforming life expectancy.

Extending this approach to the predictors, we determined it was appropriate to apply the same transformation equation. This is because the lambda value was very large and was pulled from the model that included all predictors. In addition, by making the transformations uniform, it ensures a coherent and constituent transformation process, which aligns with the statistical considerations of the Box-Cox method.

Multicollinearity Assessment:

Once the transformations were completed, we built a new model based on the transformations, and then moved on to assess if multicollinearity was present. We used the car package in RStudio to extract the Variable Inflation Factors for each predictor.

Based on the VIFs, the variables that had the largest VIFs were internet, elderly population, and birth rate. The internet variable had the highest VIF at 3.0436. Land area, population, health, and CO2 had the smallest VIFs, indicating that the potential for collinearity was very small.

Although the VIFs were all less than 4, we still decided that it was important to look out for internet, elderly population, and birth rate when performing the model selection process as these three variables have the highest potential for collinearity.

Model Selection:

When selecting the predictors for our new model, we used forward stepwise regression, backward elimination, and the exhaustive method. We used the leaps package to perform the model selection process, and determined the appropriate predictors to choose for our new model.

The forward stepwise regression would start with the strongest predictor, based on the partial F, and add more predictors every time the models were built. The backward elimination is the opposite of forward stepwise regression, so it started with all predictors and dropped predictors until the partial F was significant. The exhaustive method considered all possible models for the predictors, and explored all combinations to evaluate the best models.

We used the adjusted R^2 criterion to determine which predictors were the best to use. The rationale behind this choice was that we wanted to maximize linearity. By selecting predictors based

on the largest adjusted R^2 values, we aimed to determine the combination that contributes most effectively to the model's power while accounting for the number of predictors. In all three models, the predictors that produced the largest R^2 at a value of 0.7851269 were the predictors of land area, rural data, health data, internet data, birth rate, and cell data. We then fit these predictors into our selected model.

Model Refinement:

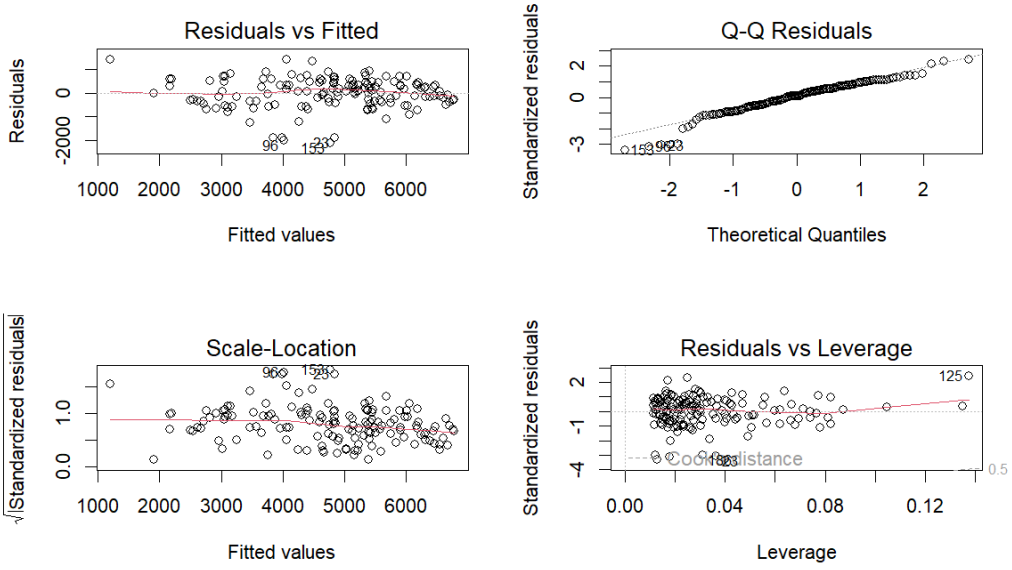
We decided to refine our model by using the Akaike Information Criterion. We used the step() function in RStudio to perform backward selection based on AIC. Models are built with the step function, and the models with predictors that produce low AICs are kept. This process continues until there are no more significant drops in AIC.

After performing the AIC selection, the land area and rural predictors were dropped. This meant that our final model has the predictors of health data, internet, birth rate, and cell data.

We then checked for multicollinearity one more time. This is because the internet and birth rate predictors had the highest VIFs previously. After checking again, all of the predictors had VIFs that were close to one, with the birth rate predictor having the highest VIF of 1.7474.

Since the multicollinearity assessment concluded good results, our final model uses health, internet, birth rate, and cell as predictors, with life expectancy as the response variable.

Assumption Verification:



We performed residual analysis one more time to determine if our final model violates any assumptions. Based on the Residuals vs Fitted graph, the new model definitely meets the assumptions of linearity. The assumption of equal variances is also met now. The Q-Q plot suggests a mild deviation from the assumption of normality as the ends of the plots slightly deviate, but overall, the assumption seems fine.

Our final predictors of health, internet, birth rate, and cell also all have p-values less than the standard alpha of 0.05, meaning the predictors are all statistically significant in proving correlation with life expectancy. The adjusted R^2 for the model is 0.7838, indicating strong linearity.

Final Model:

Predictors: Health, Internet, Birth Rate, Cell

Response: Life Expectancy

Conclusion:

Throughout the model building process, we used a variety of methods to extract the best predictors. From this project, we determined that transformations and multicollinearity assessments are important in reshaping and understanding the data. We developed a strong process for identifying good predictors, which is to select models and use AIC as a refinement criterion, as well as checking this new model with variable inflation factors again.

Overall, our key conclusion for this project is that health, internet, birth rate, and cell are the best predictors in assessing life expectancy. Ultimately, these four predictors are applicable variables that influence the life expectancy of every country.

Code Appendix

Ellie Bi & Terrie Kim

2023-11-28

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2
```

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

```
## Warning: package 'carData' was built under R version 4.3.2
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.2
```

Data Preparation

```
countries <- read.csv('~/rstudio/sta108/countries.csv')
n <- nrow(countries)
set.seed(8)
subset_id <- sample(n, 0.8*n)
countries <- countries[subset_id, ]
```

```
# removing country and code
data <- countries[, -(1:2)]
```

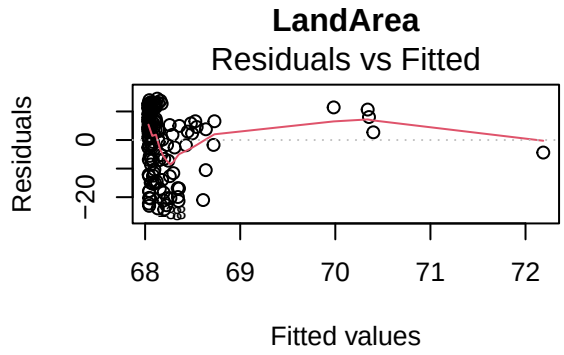
```
countries.model <- lm(LifeExpectancy ~
                        LandArea +
                        Population +
                        Rural +
                        Health +
                        Internet +
                        BirthRate +
                        ElderlyPop +
                        CO2 +
```

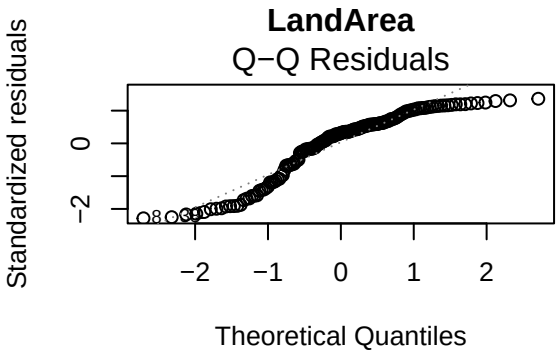
```
                        GDP +
                        Cell,
                      data = data)
summary(countries.model)
```

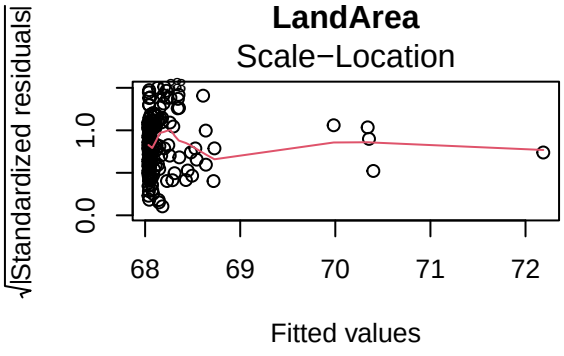
```
##
## Call:
## lm(formula = LifeExpectancy ~ LandArea + Population + Rural +
##     Health + Internet + BirthRate + ElderlyPop + CO2 + GDP +
##     Cell, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4181  -2.7193   0.6765   3.1449  10.9762
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.235e+01  3.600e+00  22.876  < 2e-16 ***
## LandArea    -2.481e-07  2.580e-07  -0.961   0.3381
## Population   2.081e-03  4.373e-03   0.476   0.6349
## Rural       -2.479e-02  2.721e-02  -0.911   0.3639
## Health       1.998e-01  1.152e-01   1.735   0.0851 .
## Internet     8.876e-02  3.526e-02   2.518   0.0130 *
## BirthRate   -7.169e-01  7.845e-02  -9.138 7.61e-16 ***
## ElderlyPop  -3.813e-01  1.600e-01  -2.384   0.0185 *
## CO2         -8.429e-02  8.443e-02  -0.998   0.3199
## GDP          3.099e-05  4.188e-05   0.740   0.4606
## Cell         1.860e-02  1.436e-02   1.295   0.1974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.084 on 137 degrees of freedom
## Multiple R-squared:  0.7873, Adjusted R-squared:  0.7718
## F-statistic:  50.7 on 10 and 137 DF,  p-value: < 2.2e-16
```

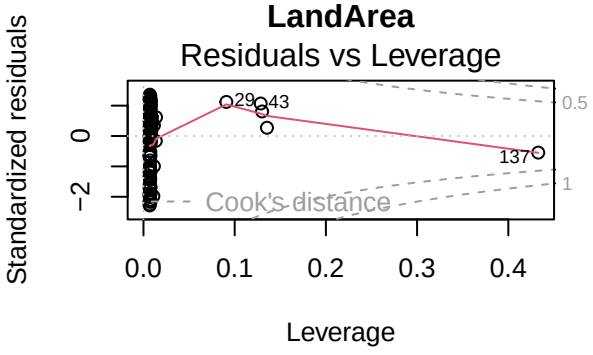
Data Exploration - Residual Analysis

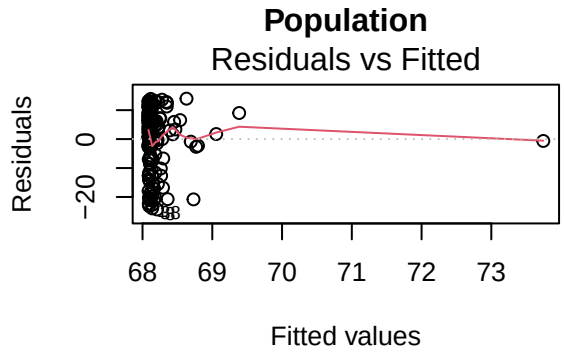
```
xdata <- data[c(1:7, 9:11)]
ydata <- data$LifeExpectancy
for (i in 1:10){
  model <- lm(ydata~xdata[[i]])
  par(mfrow=c(2,2))
  plot(model, main = names(xdata[i]))
}
```

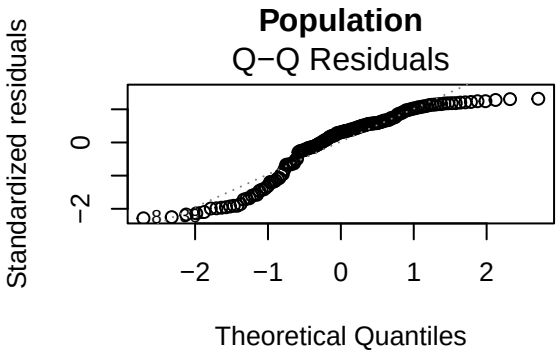

LandArea
Residuals vs Fitted
Residuals
−20
0
68
69
70
71
72
Fitted values

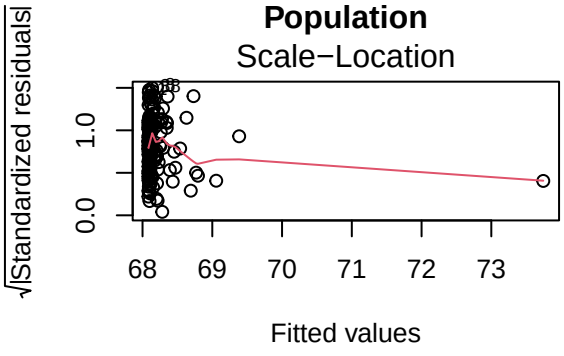

LandArea
Q−Q Residuals
Standardized residuals
−2
0
−2
−1
0
1
2
Theoretical Quantiles

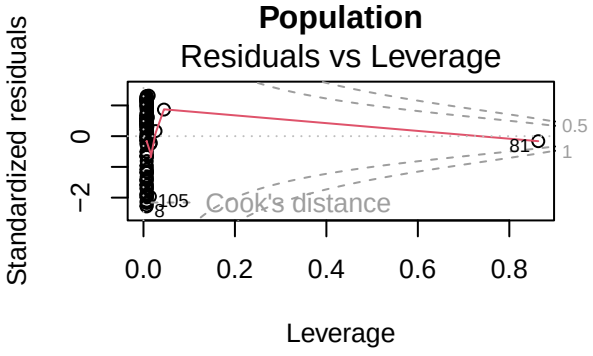

LandArea
Scale−Location
√|Standardized residuals|
0.0
1.0
68
69
70
71
72
Fitted values

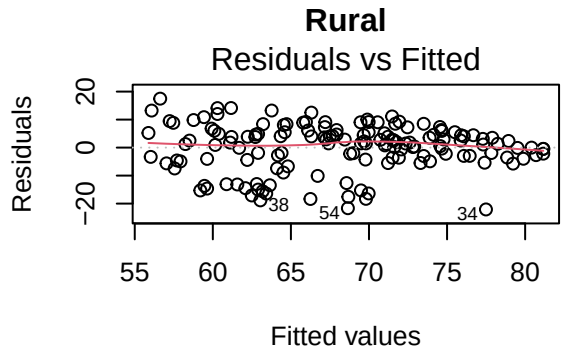

LandArea
Residuals vs Leverage
Standardized residuals
−2
0
29
43
137
0.5
1
Cook's distance
0.0
0.1
0.2
0.3
0.4
Leverage

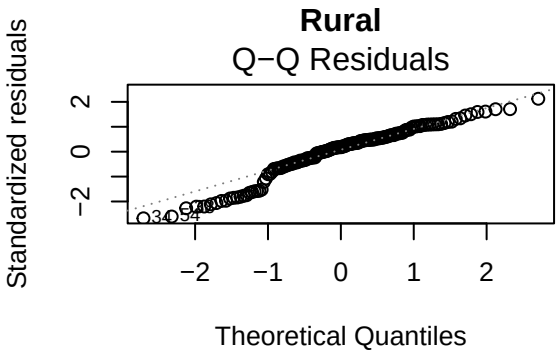

Population
Residuals vs Fitted
Residuals
Fitted values

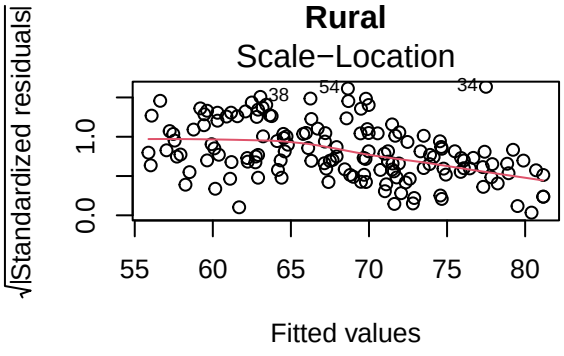

Population
Q−Q Residuals
Standardized residuals
Theoretical Quantiles

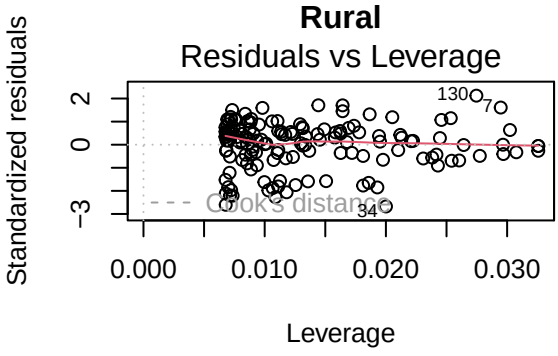

Population
Scale−Location
√|Standardized residuals|
Fitted values

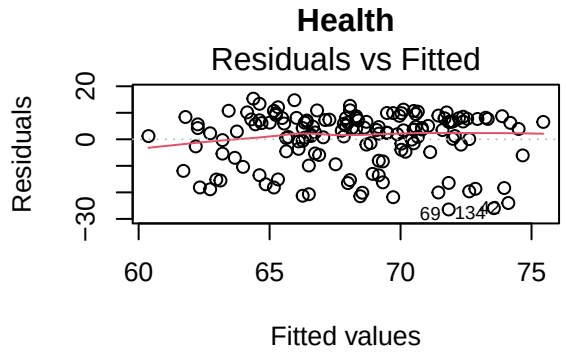

Population
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

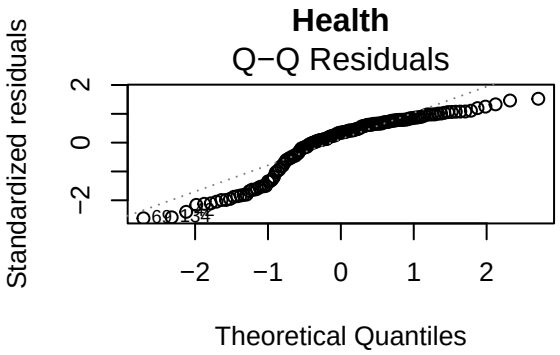

Rural
Residuals vs Fitted
Residuals
Fitted values
38
54
34

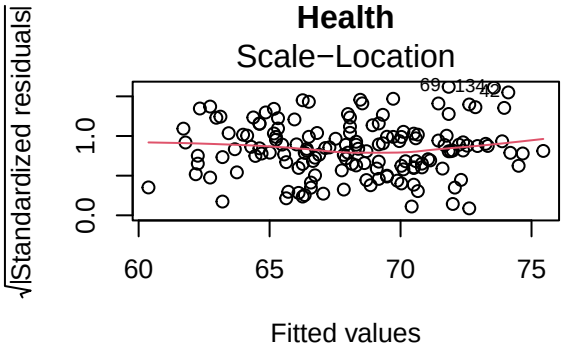

Rural
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
34
54

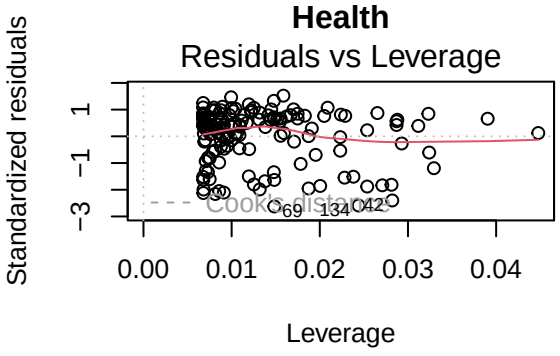

Rural
Scale−Location
√|Standardized residuals|
Fitted values
38
54
34

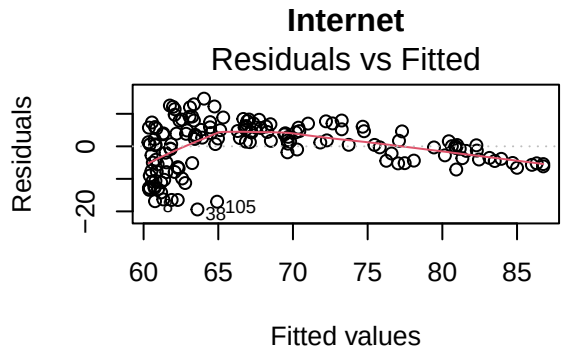

Rural
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
130
7
34

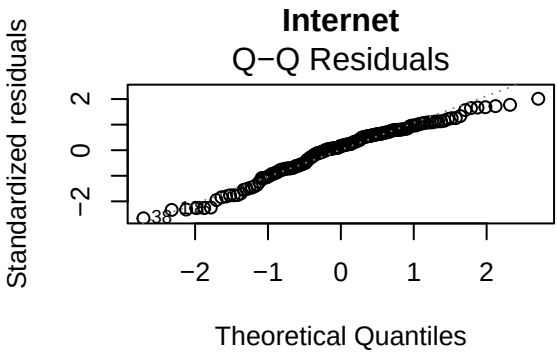

Health
Residuals vs Fitted
Residuals
Fitted values
69
134
42

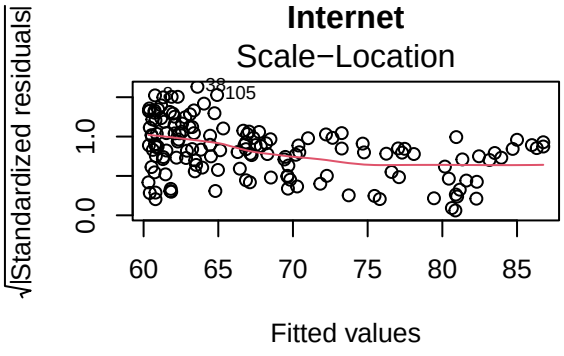

Health
Q−Q Residuals
Standardized residuals
Theoretical Quantiles

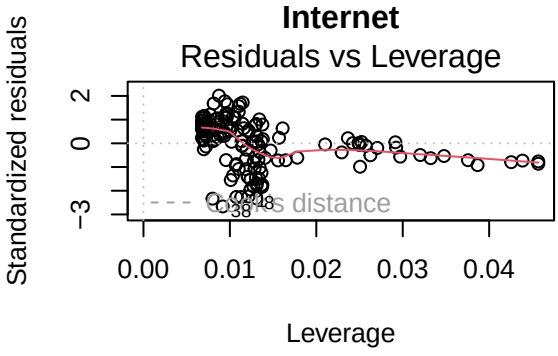

Health
Scale−Location
√|Standardized residuals|
Fitted values
69
134
42

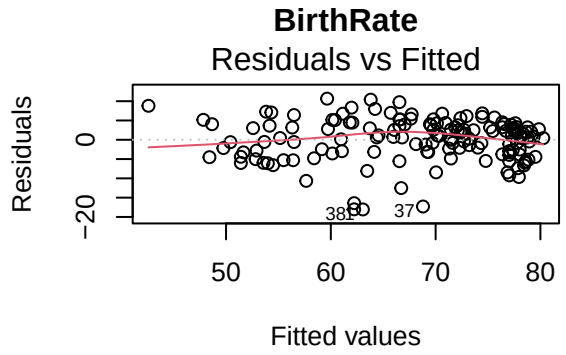

Health
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
69
134
42

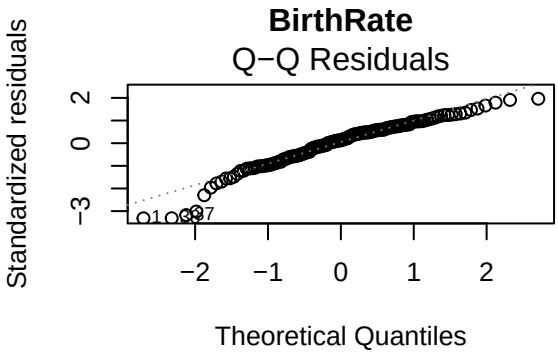

Internet
Residuals vs Fitted
Residuals
Fitted values
38
105

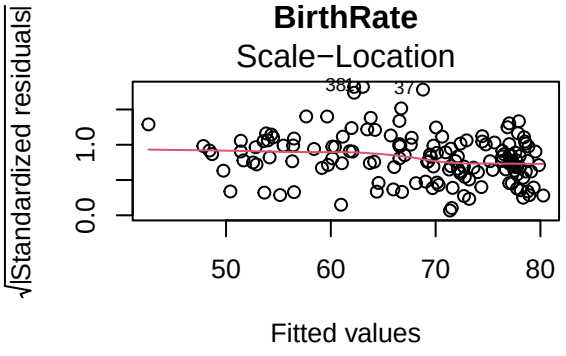

Internet
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
38

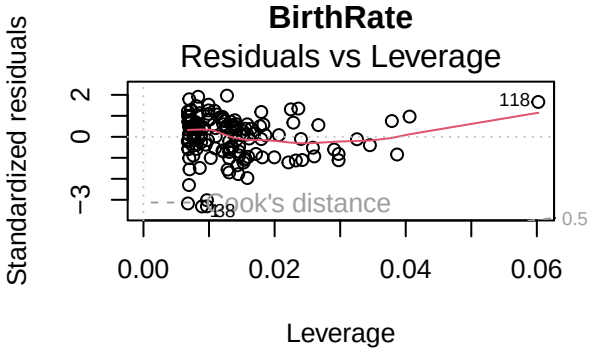

Internet
Scale−Location
√|Standardized residuals|
Fitted values
38
105

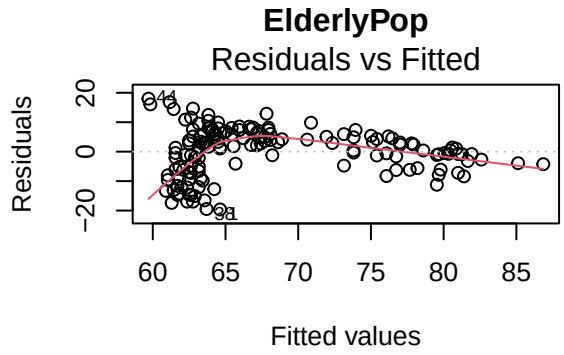

Internet
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
38

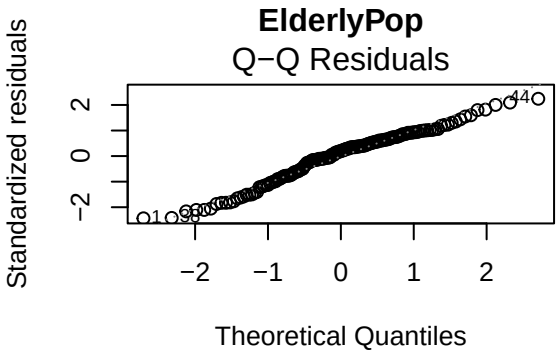

BirthRate
Residuals vs Fitted
Residuals
Fitted values
381
37

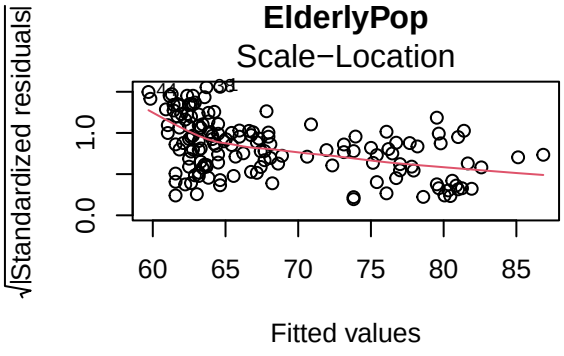

BirthRate
Q−Q Residuals
Standardized residuals
Theoretical Quantiles

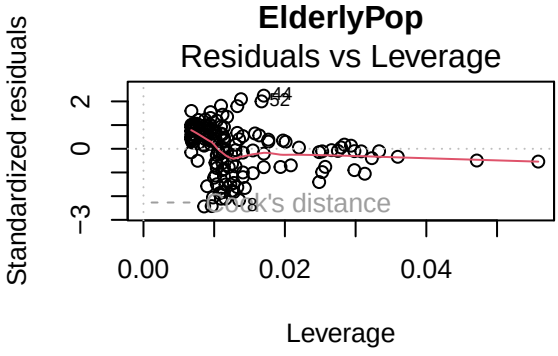

BirthRate
Scale−Location
√|Standardized residuals|
Fitted values
381
37

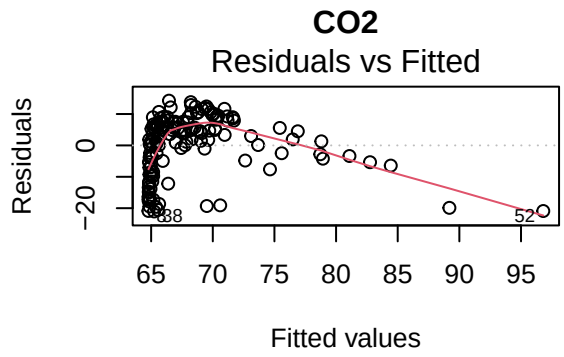

BirthRate
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
118

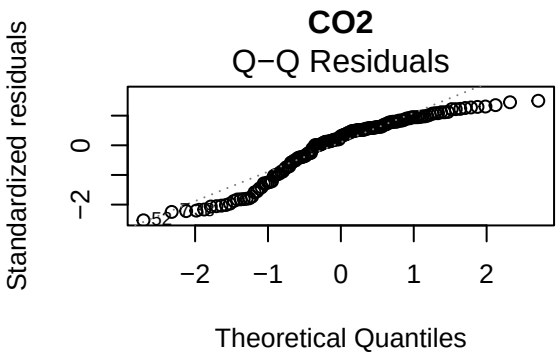

ElderlyPop
Residuals vs Fitted
Residuals
Fitted values

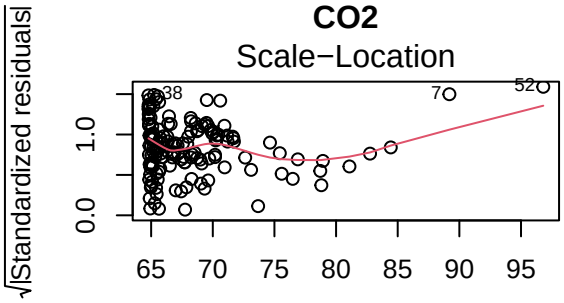

ElderlyPop
Q−Q Residuals
Standardized residuals
Theoretical Quantiles

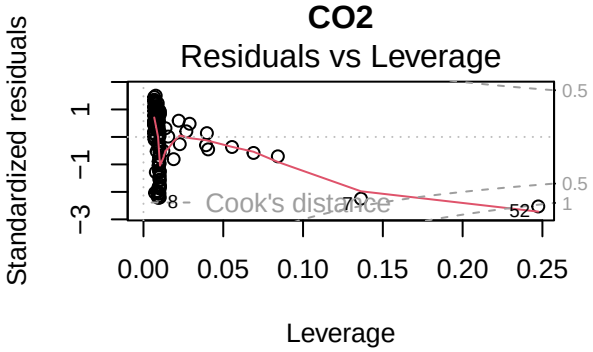

ElderlyPop
Scale−Location
√|Standardized residuals|
Fitted values

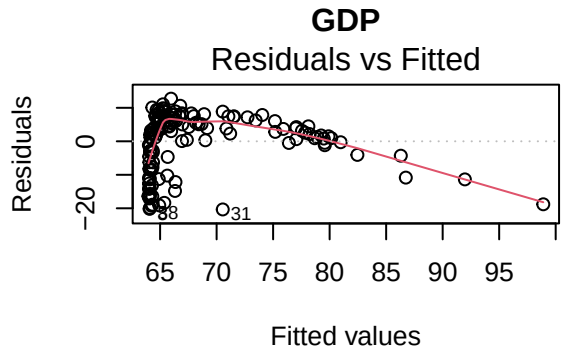

ElderlyPop
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance

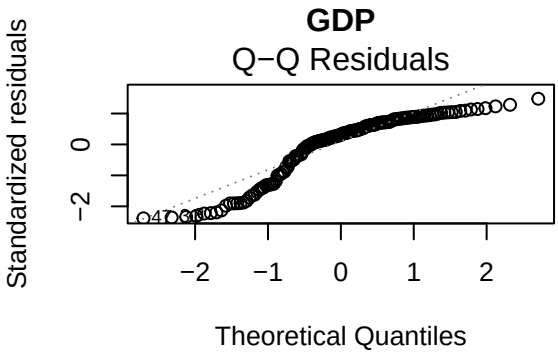

CO2
Residuals vs Fitted
Residuals
Fitted values
38
52

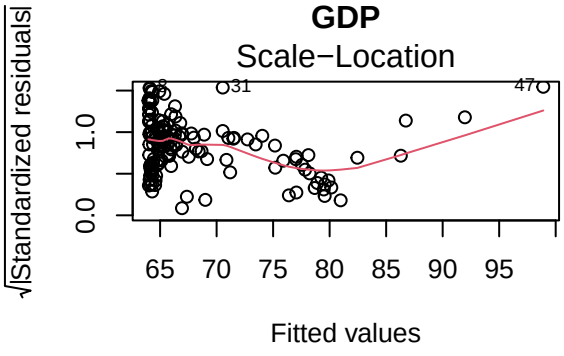

CO2
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
52

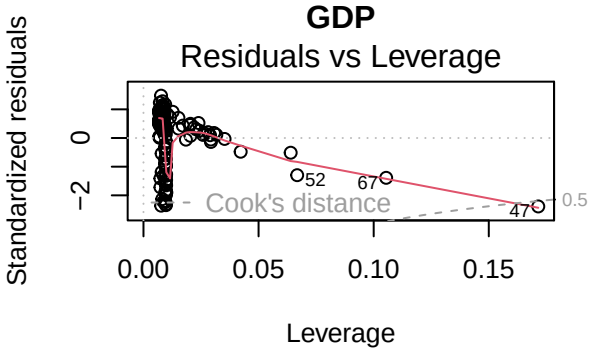

CO2
Scale−Location
√|Standardized residuals|
Fitted values
38
7
52

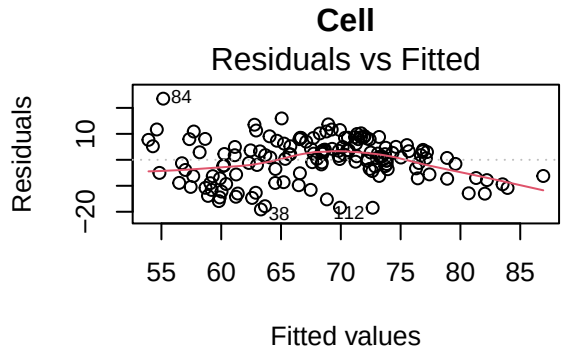

CO2
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
8
7
52

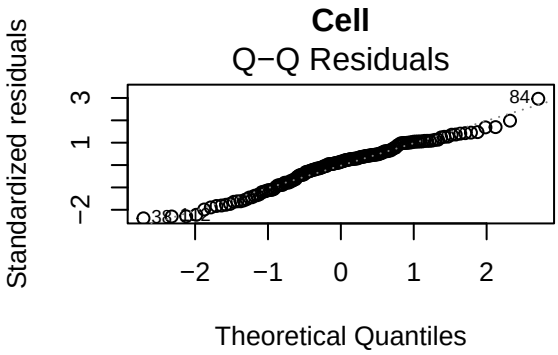

GDP
Residuals vs Fitted
Residuals
Fitted values
31
88

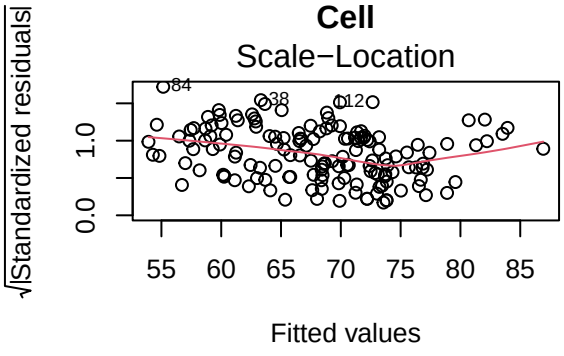

GDP
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
47

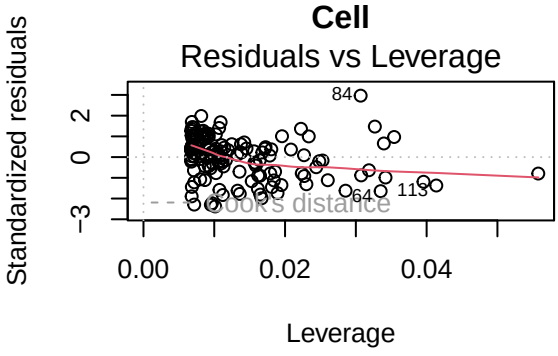

GDP
Scale−Location
√|Standardized residuals|
Fitted values
31
47


GDP
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.5
52
67
47



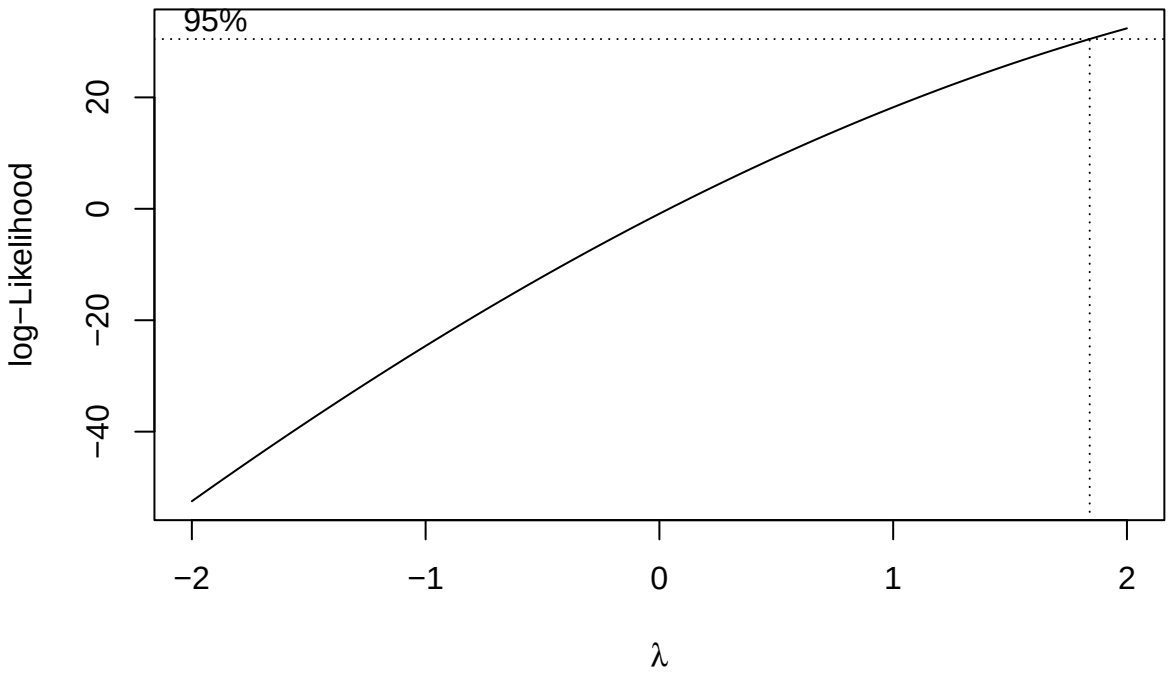






Transformations

```
countries.boxcox <- boxcox(countries.model, plotit = T)
```




```
lambda <- countries.boxcox$x[which.max(countries.boxcox$y)]
```

```
# perform y transformation
data$transf.LE <- (data$LifeExpectancy ^ lambda) / (lambda - 1)
```

```
# perform x transformation
xdata <- data[c(1:7, 9:11)]
new.ydata <- data$transf.LE

for (i in 1:length(xdata)){
  colname <- paste('transf.', names(xdata)[i], sep="")
  data[[colname]] <- (xdata[[i]] ^ lambda) / (lambda - 1)
}

# we don't need to transform health:
data$transf.Health <- data$Health

head(data)
```

```
##     LandArea Population Rural Health Internet BirthRate ElderlyPop
## 96     30360      2.049  74.5    8.2      3.6      28.9        4.7
## 52     20720      6.134  39.3   11.9     10.6      20.2        7.0
## 183   310070     86.211  72.2    9.3     24.2      17.2        6.3
## 79   1628550     71.956  31.5    8.7     32.0      18.9        4.9
## 12       760      0.776  11.5   10.3     51.9      18.0        2.3
```

```
## 119   823290      2.130  63.2   12.1      5.3      27.6        3.6
##     LifeExpectancy        CO2        GDP      Cell transf.LE transf.LandArea
## 96            45.0  1.1954683   982.1203  32.18322   2025.00    9.217296e+08
## 52            71.3  0.9972692  3425.5973 124.33949   5083.69    4.293184e+08
## 183           74.4  1.4964850  1224.1911 177.14086   5535.36    9.614340e+10
## 79            71.4  7.4479027  6274.0369  91.24873   5097.96    2.652175e+12
## 12            75.9 21.3603057 37624.7019 124.18422   5760.81    5.776000e+05
## 119           61.0  1.8031514  5330.1759  67.20691   3721.00    6.778064e+11
##     transf.Population transf.Rural transf.Health transf.Internet
## 96           4.198401      5550.25           8.2           12.96
## 52          37.625956      1544.49          11.9          112.36
## 183       7432.336521      5212.84           9.3          585.64
## 79        5177.665936       992.25           8.7         1024.00
## 12           0.602176       132.25          10.3         2693.61
## 119          4.536900      3994.24          12.1           28.09
##     transf.BirthRate transf.ElderlyPop  transf.CO2   transf.GDP transf.Cell
## 96            835.21             22.09   1.4291444     964560.4    1035.760
## 52            408.04             49.00   0.9945458   11734716.8   15460.309
## 183           295.84             39.69   2.2394674    1498643.8   31378.884
## 79            357.21             24.01  55.4712551   39363538.4    8326.331
## 12            324.00              5.29 456.2626617 1415618191.6   15421.721
## 119           761.76             12.96   3.2513550   28410774.8    4516.769
```

```
new.model <- lm(transf.LE ~
                  transf.LandArea +
                  transf.Population +
                  transf.Rural +
                  transf.Health +
                  transf.Internet +
                  transf.BirthRate +
                  transf.ElderlyPop +
                  transf.CO2 +
                  transf.GDP +
                  transf.Cell,
                      data = data)
summary(new.model)
```

```
##
## Call:
## lm(formula = transf.LE ~ transf.LandArea + transf.Population +
##     transf.Rural + transf.Health + transf.Internet + transf.BirthRate +
##     transf.ElderlyPop + transf.CO2 + transf.GDP + transf.Cell,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2157.46  -333.83    74.97   413.92  1444.93
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.052e+03  2.292e+02  22.042  < 2e-16 ***
## transf.LandArea    -3.276e-12  2.225e-12  -1.472  0.14323
## transf.Population   2.545e-04  3.904e-04   0.652  0.51545
## transf.Rural       -3.787e-02  3.413e-02  -1.110  0.26913
```

```
## transf.Health      3.172e+01  1.422e+01   2.231  0.02732 *
## transf.Internet    1.376e-01  4.275e-02   3.218  0.00161 **
## transf.BirthRate  -1.420e+00  1.367e-01 -10.385  < 2e-16 ***
## transf.ElderlyPop  2.181e-01  8.028e-01   0.272  0.78627
## transf.CO2         1.223e-01  2.365e-01   0.517  0.60601
## transf.GDP         1.550e-08  5.690e-08   0.272  0.78568
## transf.Cell        1.508e-02  8.376e-03   1.801  0.07394 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 641.6 on 137 degrees of freedom
## Multiple R-squared:  0.7953, Adjusted R-squared:  0.7803
## F-statistic: 53.22 on 10 and 137 DF,  p-value: < 2.2e-16
```

Assess Potential for Multicolinearity

```
all_vifs <- car::vif(new.model)
print(all_vifs)
```

```
##   transf.LandArea transf.Population      transf.Rural     transf.Health
##          1.133419          1.133936          1.854404          1.290100
##   transf.Internet  transf.BirthRate transf.ElderlyPop        transf.CO2
##          3.043646          2.322704          2.447736          1.338975
##        transf.GDP       transf.Cell
##          2.073165          1.693568
```

All VIFs are < 4, meaning all variables are independent of each other. LandArea, Population, and Health have the closest VIFs to 1, meaning they have the least correlation with other variables.

Model Selection

```
newdata <- data[c(12:22)]
```

```
forward.model <- regsubsets(transf.LE ~ ., data = newdata, nbest=1, nvmax=10, method="forward")
with(summary(forward.model), data.frame(rsq, adjr2, cp, rss, outmat))
```

```
##                rsq     adjr2        cp      rss transf.LandArea
## 1  ( 1 )  0.7071519 0.7051461 51.970160 80674427
## 2  ( 1 )  0.7757731 0.7726803  8.049763 61770519
## 3  ( 1 )  0.7832397 0.7787238  5.053243 59713618
## 4  ( 1 )  0.7897050 0.7838226  2.726715 57932529
## 5  ( 1 )  0.7921378 0.7848188  3.098687 57262324              *
## 6  ( 1 )  0.7938972 0.7851269  3.921314 56777639              *
## 7  ( 1 )  0.7944989 0.7842238  5.518717 56611903              *
## 8  ( 1 )  0.7950558 0.7832605  7.146000 56458468              *
## 9  ( 1 )  0.7951637 0.7818048  9.073818 56428753              *
## 10  ( 1 ) 0.7952740 0.7803305 11.000000 56398364              *
```

```
##           transf.Population transf.Rural transf.Health transf.Internet
## 1  ( 1 )                                                              
## 2  ( 1 )                                                           *  
## 3  ( 1 )                                              *            *  
## 4  ( 1 )                                              *            *  
## 5  ( 1 )                                              *            *  
## 6  ( 1 )                                 *            *            *  
## 7  ( 1 )                    *            *            *            *  
## 8  ( 1 )                    *            *            *            *  
## 9  ( 1 )                    *            *            *            *  
## 10  ( 1 )                   *            *            *            *  
##           transf.BirthRate transf.ElderlyPop transf.CO2 transf.GDP transf.Cell
## 1  ( 1 )                 *                                                    
## 2  ( 1 )                 *                                                    
## 3  ( 1 )                 *                                                    
## 4  ( 1 )                 *                                                  *
## 5  ( 1 )                 *                                                  *
## 6  ( 1 )                 *                                                  *
## 7  ( 1 )                 *                                                  *
## 8  ( 1 )                 *                          *                       *
## 9  ( 1 )                 *                          *          *            *
## 10  ( 1 )                *                *         *          *            *
```

adjr2 -> 6 - LandArea, Rural, Health, Internet, BirthRate, Cell

```
backward.model <- regsubsets(transf.LE ~ ., data = newdata, nbest=1, nvmax=10,
                             method="backward")
with(summary(backward.model), data.frame(rsq, adjr2, cp, rss, outmat))
```

```
##                rsq     adjr2        cp      rss transf.LandArea
## 1  ( 1 )  0.7071519 0.7051461 51.970160 80674427                
## 2  ( 1 )  0.7757731 0.7726803  8.049763 61770519                
## 3  ( 1 )  0.7832397 0.7787238  5.053243 59713618                
## 4  ( 1 )  0.7897050 0.7838226  2.726715 57932529                
## 5  ( 1 )  0.7921378 0.7848188  3.098687 57262324              *  
## 6  ( 1 )  0.7938972 0.7851269  3.921314 56777639              *  
## 7  ( 1 )  0.7944989 0.7842238  5.518717 56611903              *  
## 8  ( 1 )  0.7950558 0.7832605  7.146000 56458468              *  
## 9  ( 1 )  0.7951637 0.7818048  9.073818 56428753              *  
## 10  ( 1 ) 0.7952740 0.7803305 11.000000 56398364              *  
##           transf.Population transf.Rural transf.Health transf.Internet
## 1  ( 1 )                                                              
## 2  ( 1 )                                                           *  
## 3  ( 1 )                                              *            *  
## 4  ( 1 )                                              *            *  
## 5  ( 1 )                                              *            *  
## 6  ( 1 )                                 *            *            *  
## 7  ( 1 )                    *            *            *            *  
## 8  ( 1 )                    *            *            *            *  
## 9  ( 1 )                    *            *            *            *  
## 10  ( 1 )                   *            *            *            *  
##           transf.BirthRate transf.ElderlyPop transf.CO2 transf.GDP transf.Cell
## 1  ( 1 )                 *                                                    
```

```
## 2  ( 1 )                *                                                    
## 3  ( 1 )                *                                                    
## 4  ( 1 )                *                                                  * 
## 5  ( 1 )                *                                                  * 
## 6  ( 1 )                *                                                  * 
## 7  ( 1 )                *                                                  * 
## 8  ( 1 )                *                          *                       * 
## 9  ( 1 )                *                          *          *            * 
## 10  ( 1 )               *                *         *          *            * 
```

adjr2 -> 6 - LandArea, Rural, Health, Internet, BirthRate, Cell

```
exh.model <- regsubsets(transf.LE ~ ., data = newdata, nbest=1, nvmax=10,
                        method="exhaustive")
with(summary(exh.model), data.frame(rsq, adjr2, cp, rss, outmat))
```

```
##                rsq     adjr2        cp      rss transf.LandArea
## 1  ( 1 )  0.7071519 0.7051461 51.970160 80674427                
## 2  ( 1 )  0.7757731 0.7726803  8.049763 61770519                
## 3  ( 1 )  0.7832397 0.7787238  5.053243 59713618                
## 4  ( 1 )  0.7897050 0.7838226  2.726715 57932529                
## 5  ( 1 )  0.7921378 0.7848188  3.098687 57262324              * 
## 6  ( 1 )  0.7938972 0.7851269  3.921314 56777639              * 
## 7  ( 1 )  0.7944989 0.7842238  5.518717 56611903              * 
## 8  ( 1 )  0.7950558 0.7832605  7.146000 56458468              * 
## 9  ( 1 )  0.7951637 0.7818048  9.073818 56428753              * 
## 10  ( 1 ) 0.7952740 0.7803305 11.000000 56398364              * 
##           transf.Population transf.Rural transf.Health transf.Internet
## 1  ( 1 )                                                              
## 2  ( 1 )                                                           *  
## 3  ( 1 )                                              *            *  
## 4  ( 1 )                                              *            *  
## 5  ( 1 )                                              *            *  
## 6  ( 1 )                                 *            *            *  
## 7  ( 1 )                  *              *            *            *  
## 8  ( 1 )                  *              *            *            *  
## 9  ( 1 )                  *              *            *            *  
## 10  ( 1 )                 *              *            *            *  
##           transf.BirthRate transf.ElderlyPop transf.CO2 transf.GDP transf.Cell
## 1  ( 1 )                *                                                    
## 2  ( 1 )                *                                                    
## 3  ( 1 )                *                                                    
## 4  ( 1 )                *                                                  * 
## 5  ( 1 )                *                                                  * 
## 6  ( 1 )                *                                                  * 
## 7  ( 1 )                *                                                  * 
## 8  ( 1 )                *                          *                       * 
## 9  ( 1 )                *                          *          *            * 
## 10  ( 1 )               *                *         *          *            * 
```

adjr2 -> 6 - LandArea, Rural, Health, Internet, BirthRate, Cell

```r
# LandArea, Rural, Health, Internet, BirthRate, Cell
selected.model <- lm(transf.LE ~
                       transf.LandArea +
                       transf.Rural +
                       transf.Health +
                       transf.Internet +
                       transf.BirthRate +
                       transf.Cell, data = newdata)
summary(selected.model)
```

```
##
## Call:
## lm(formula = transf.LE ~ transf.LandArea + transf.Rural + transf.Health +
##     transf.Internet + transf.BirthRate + transf.Cell, data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2167.10  -336.63    90.13   407.15  1479.14
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.084e+03  2.182e+02  23.302  < 2e-16 ***
## transf.LandArea  -2.797e-12  2.092e-12  -1.337   0.1834
## transf.Rural     -3.693e-02  3.366e-02  -1.097   0.2745
## transf.Health     3.168e+01  1.356e+01   2.337   0.0209 *
## transf.Internet   1.493e-01  3.162e-02   4.723 5.55e-06 ***
## transf.BirthRate -1.446e+00  1.246e-01 -11.598  < 2e-16 ***
## transf.Cell       1.530e-02  8.098e-03   1.889   0.0609 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 634.6 on 141 degrees of freedom
## Multiple R-squared:  0.7939, Adjusted R-squared:  0.7851
## F-statistic: 90.52 on 6 and 141 DF,  p-value: < 2.2e-16
```

Model Refinement

AIC Refinement

```r
step(selected.model)
```

```
## Start:  AIC=1916.9
## transf.LE ~ transf.LandArea + transf.Rural + transf.Health +
##     transf.Internet + transf.BirthRate + transf.Cell
##
##                     Df Sum of Sq       RSS    AIC
## - transf.Rural       1    484685  57262324 1916.2
## - transf.LandArea    1    719794  57497433 1916.8
## <none>                            56777639 1916.9
## - transf.Cell        1   1437416  58215055 1918.6
```

```
## - transf.Health     1   2198649  58976287 1920.5
## - transf.Internet   1   8982424  65760063 1936.6
## - transf.BirthRate  1  54167543 110945182 2014.0
##
## Step:  AIC=1916.16
## transf.LE ~ transf.LandArea + transf.Health + transf.Internet +
##     transf.BirthRate + transf.Cell
##
##                    Df Sum of Sq       RSS    AIC
## - transf.LandArea   1    670205  57932529 1915.9
## <none>                           57262324 1916.2
## - transf.Cell       1   1919702  59182026 1919.0
## - transf.Health     1   2188146  59450470 1919.7
## - transf.Internet   1  11047887  68310211 1940.3
## - transf.BirthRate  1  64909614 122171938 2026.3
##
## Step:  AIC=1915.88
## transf.LE ~ transf.Health + transf.Internet + transf.BirthRate +
##     transf.Cell
##
##                    Df Sum of Sq       RSS    AIC
## <none>                           57932529 1915.9
## - transf.Cell       1   1781088  59713618 1918.4
## - transf.Health     1   2083972  60016501 1919.1
## - transf.Internet   1  11060910  68993439 1939.7
## - transf.BirthRate  1  64407227 122339756 2024.5


##
## Call:
## lm(formula = transf.LE ~ transf.Health + transf.Internet + transf.BirthRate +
##     transf.Cell, data = newdata)
##
## Coefficients:
##      (Intercept)     transf.Health   transf.Internet  transf.BirthRate
##       4985.10460          30.81427           0.15911          -1.48363
##      transf.Cell
##          0.01658
```

Using step() performs backward selection using AIC. Models are built, and the models with predictors that produce low AICs are kept. This process continues until there is no more significant drops in AIC.

```
AIC.model <- lm(transf.LE ~
              transf.Health +
              transf.Internet +
              transf.BirthRate +
              transf.Cell, data = newdata)
```

Checking VIFs & Multicollinearity

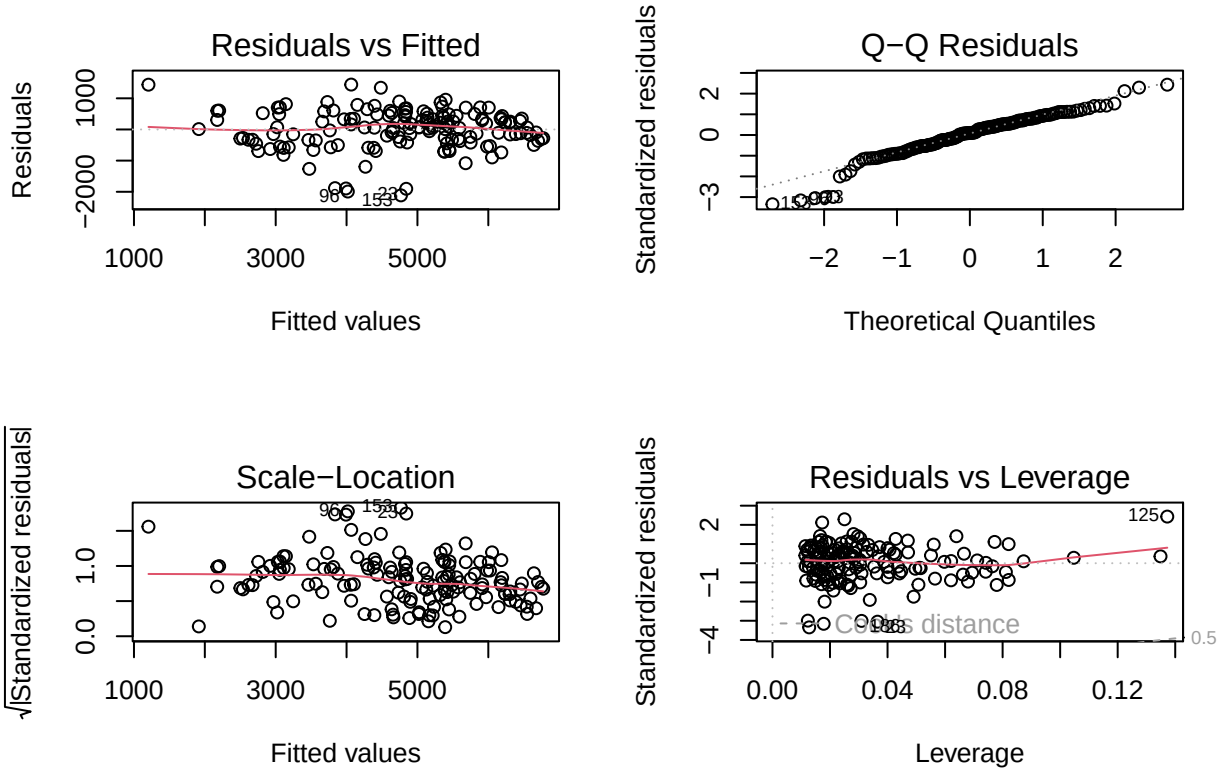
```
car::vif(AIC.model)
```

```
##    transf.Health  transf.Internet transf.BirthRate      transf.Cell
##         1.196603         1.568931         1.747469         1.533591
```

```
final <- AIC.model
```

Verification of Assumptions

```
par(mfrow = c(2,2))
plot(final)
```



Final

```
summary(final)
```

```
##
## Call:
## lm(formula = transf.LE ~ transf.Health + transf.Internet + transf.BirthRate +
##     transf.Cell, data = newdata)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2115.35  -340.86    62.12   432.58  1440.08
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.985e+03  1.999e+02  24.938  < 2e-16 ***
## transf.Health      3.081e+01  1.359e+01   2.268   0.0248 *
## transf.Internet    1.591e-01  3.045e-02   5.225 6.04e-07 ***
## transf.BirthRate -1.484e+00  1.177e-01 -12.609  < 2e-16 ***
## transf.Cell        1.658e-02  7.907e-03   2.097   0.0378 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 636.5 on 143 degrees of freedom
## Multiple R-squared:  0.7897, Adjusted R-squared:  0.7838
## F-statistic: 134.2 on 4 and 143 DF,  p-value: < 2.2e-16
```

```
avPlots(final)
```