Exploring the Impact of Factors on Life Expectancy
MTH 245 Statistical Methods I with R

Matthew Keller, Michael Wolfe, Moultrie Dangerfield

April 28, 2024

Contents

1 Abstract

According to the CDC, the average life expectancy in the United States is 76.4 years (CDC (2023)). This statistic can be influenced by a wide variety of factors such as daily habits, genetics, and the environment one lives in. In our study on the determinants of life expectancy, we employed linear regression analysis to establish a model that best fits the observed data and to understand the relationships between seven key variables and their effect on life expectancy. These are, minimum wage, birth rate, Co2 emissions, consumer price index, labor force participation rate, armed forces size, and physicians per thousand people. After creating a base model, we used several key tests to fine tune our predictions. These tests include one-way step wise, two-way step wise, RIDGE, and LASSO. This gave us 9 total models in which we used R^2 , R^2adj , LL, AIC, BIC, RMSE, MAE as criteria to dictate the highest performing model. We developed an average ranking system which ranks the model based on all the criteria stated above. Using this system, we chose the two-way backward model which had the highest rank overall. Our best fit model for life expectancy, resulting in four predictor variables, had an R^2 adj value of 0.79. This means that our model accounts for a total of 79 percent of variation in life expectancy can be explained by the predictor variables included in our model. The most significant predictor values of life expectancy are minimum wage(t-value = 4.4, p = 0.000) and physicians per thousand people(t-value = 7.198, p-value = 0.000). Additionally, the least indicative variable was surprisingly birth rate(t-value= -1.035, p-value= 0.303).

2 Introduction

The purpose of our study is to determine which countrywide statistics such as GDP, minimum wage, and education enrollment have the greatest influence on the average life expectancy. In order to conduct our study, we used a data set titled “Global Country Information” data set from Kaggle (Elgiriyewithana (2023)) This data set provides several statistics for 195 unique countries all around the world. The data set provides different education statistics, economic information, and environmental factors from every country in the world. For this study in particular, we chose to focus on seven variables in which we believe will be best suited as predictors for human life expectancy. These seven variables include minimum Wage, armed forces size, birth rate, labor force participation rate (LFPR), CO2 emissions, CPI, and physicians per thousand people (PPT). We are interested to see which country characteristics will be the strongest predictor of average life expectancy, measured in years.

All the variables in our model are all continuous and quantitative. For our variables, life expectancy is defined as the average number of years that a newborn is expected to live. We chose this to be variable we are attempting to predict with our other factors. The first predictor variable is armed forces size. With this variable, we chose this variable because a country that spends a lot on the military could potentially find themselves in frequent conflicts which would lower life expectancy, or a country that is developed enough to have a large military may have a better quality of life. Our next variable is birth rate, which we chose because as a higher birth rate could mean a country invests more money into healthcare resources and increases life expectancy or citizens of a country may have a higher birth rate to offset high infant mortality or low life expectancy. We chose labor force participation rate(LFPR) because countries with a higher LFPR may be more developed and thus a higher quality of life and life expectancy. The CO2 emissions variable is the total carbon dioxide emissions in tons. We chose this to analyze this variable as we believe a higher CO2 emissions will have more unhealthy living conditions and therefore decrease life expectancy. We believe the Consumer Price Index (CPI) variable could be interesting as it measures inflation rates and purchasing power. It is possible that a country’s CPI may indicate that state of the economy and its impact on life expectancy. The physicians per thousand people (PPT) variable was picked because we thought if this number was higher there would potentially be a better healthcare system in place to increase the life expectancy of the population. Our final variable is minimum wage measured in U.S. Dollars. We wanted to observe whether or not a raise in minimum wage would reflect access to a better standard of living and thus higher life expectancy.

This study has the potential to be impactful in countries all over the world. With our analysis in mind, governments can do effective planning measures to allocate resources to improving the strongest predictors of life expectancy to ensure the well being of the population. Knowing the most significant factors effecting life expectancy can give humans a path to longer and healthier lives overall.

We used the following packages in our analysis:

xtable (Dahl et al. (2019)), tidyverse (Wickham et al. (2019)), RVAideMemoire (Herve (2023)), ggplot2 (Wickham (2016)), GGally (Schloerke et al. (2021)), caret (Kuhn and Max (2008)), bestglm (McLeod et al. (2020)), patchwork (Pedersen (2023)) lmtest (Hothorn et al. (2022)), boot(Brazzale (2024)), Ecdat (Croissant and Graves (2022)), Glmnet (Friedman et al. (2023)), Mass (Ripley et al. (2024)), RMS(Harrell (2023)),and FSA (Ogle et al. (2023)).

3 Exploratory Data Analysis

Now we will be preforming an exploratory data analysis on our data set. We begin by using the library function to access the correct packages and reading in the data set. For our analysis, we are using the seed of ”0” for reproduce-ability. We chose 0.05 to be our operating significance value.

```r
library(tidyverse)
library(xtable)
library(patchwork)
library(RVAideMemoire)
library(FSA)
library(rms)
library(Stat2Data)
library(MASS)
library(ggplot2)
library(car)
library(GGally)
library(caret)
library(RColorBrewer)
library(glmnet)
library(bestglm)
library(Ecdat)
library(boot)
library(lmtest)

life.df <- read_csv("world.csv")
set.seed(0)
```

Before we start our analysis of the data, we must complete some slight data cleaning, we chose to make these changes to the ”.csv” file itself. The original data set separated the column names using a ”space” key, which is unable to be read by R, in order to remedy this, we replaced all ”spaces” with an ”_” character. Any variables that are measured in percentage had a ”%” character after its value. We chose to remove all of these ”%” characters and change the percentages to continuously quantitative numbers between 0.0 and 1.0, rather than 0.0 and 100. Finally, we chose to omit any countries that contained NA values.

```r
life.df <- na.omit(life.df)

glimpse(life.df)

## Rows: 120
## Columns: 26
## $ Population_density               <dbl> 60, 105, 18, 26, 17, 104, 3, 123, 39,~
## $ Agricultural_Land                <dbl> 0.58, 0.43, 0.17, 0.48, 0.54, 0.59, 0~
## $ Land_Area                        <dbl> 652230, 28748, 2381741, 1246700, 2780~
## $ Armed_Forces_size                <dbl> 323000, 9000, 317000, 117000, 105000,~
## $ Birth_Rate                       <dbl> 32.49, 11.78, 24.28, 40.73, 17.02, 13~
## $ Co2_Emissions                    <dbl> 8672, 4536, 150006, 34693, 201348, 51~
## $ CPI                              <dbl> 149.90, 119.05, 151.36, 261.73, 232.7~
## $ CPI_Change                       <dbl> 0.02, 0.01, 0.02, 0.17, 0.54, 0.01, 0~
## $ Fertility_Rate                   <dbl> 4.47, 1.62, 3.02, 5.52, 2.26, 1.76, 1~
## $ Forested_Area                    <dbl> 0.02, 0.28, 0.01, 0.46, 0.10, 0.12, 0~
## $ Gasoline_Price                   <dbl> 0.70, 1.36, 0.28, 0.97, 1.10, 0.77, 0~
## $ GDP                              <dbl> 1.910135e+10, 1.527808e+10, 1.699882e~
## $ GPEM                             <dbl> 1.04, 1.07, 1.10, 1.14, 1.10, 0.93, 1~
## $ GTEE                             <dbl> 0.10, 0.55, 0.51, 0.09, 0.90, 0.55, 1~
## $ Infant_mortality                 <dbl> 47.9, 7.8, 20.1, 51.6, 8.8, 11.0, 3.1~
## $ Life_expectancy                  <dbl> 64.5, 78.5, 76.7, 60.8, 76.5, 74.9, 8~
## $ Maternal_mortality_ratio         <dbl> 638, 15, 112, 241, 39, 26, 6, 26, 70,~
## $ Minimum_wage                     <dbl> 0.43, 1.12, 0.95, 0.71, 3.35, 0.66, 1~
## $ Out_of_pocket_health_expenditure <dbl> 0.78, 0.57, 0.28, 0.33, 0.18, 0.82, 0~
## $ Physicians_per_thousand          <dbl> 0.28, 1.20, 1.72, 0.21, 3.96, 4.40, 3~
## $ Population                       <dbl> 38041754, 2854191, 43053054, 31825295~
## $ LFPR                             <dbl> 0.49, 0.56, 0.41, 0.78, 0.61, 0.56, 0~
## $ Tax_revenue                      <dbl> 0.09, 0.19, 0.37, 0.09, 0.10, 0.21, 0~
## $ tax_rate                         <dbl> 0.71, 0.37, 0.66, 0.49, 1.06, 0.23, 0~
## $ Unemployment_rate                <dbl> 0.11, 0.12, 0.12, 0.07, 0.10, 0.17, 0~
## $ Urban_population                 <dbl> 9797273, 1747593, 31510100, 21061025,~
```

With our data cleaned and usable, we can now begin our exploratory data analysis.

```
summary_stats <- life.df %>%
  reframe(
    mean = c(mean(Minimum_wage), mean(Armed_Forces_size),
      mean(CPI), mean(Birth_Rate),
      mean(LFPR), mean(Co2_Emissions),
      mean(Physicians_per_thousand), mean(Life_expectancy)
    ),
    median = c(median(Minimum_wage), median(Armed_Forces_size),
      median(CPI), median(Birth_Rate),
      median(LFPR), median(Co2_Emissions),
      median(Physicians_per_thousand), median(Life_expectancy)
    ),
    min = c(min(Minimum_wage), min(Armed_Forces_size),
      min(CPI), min(Birth_Rate),
      min(LFPR), min(Co2_Emissions),
      min(Physicians_per_thousand), min(Life_expectancy)
    ),
    max = c(max(Minimum_wage), max(Armed_Forces_size),
      max(CPI), max(Birth_Rate),
      max(LFPR),max(Co2_Emissions),
      max(Physicians_per_thousand), max(Life_expectancy)
    )
  )

#xtable(summary_stats, caption = "Summary of Country Data Variables")
```

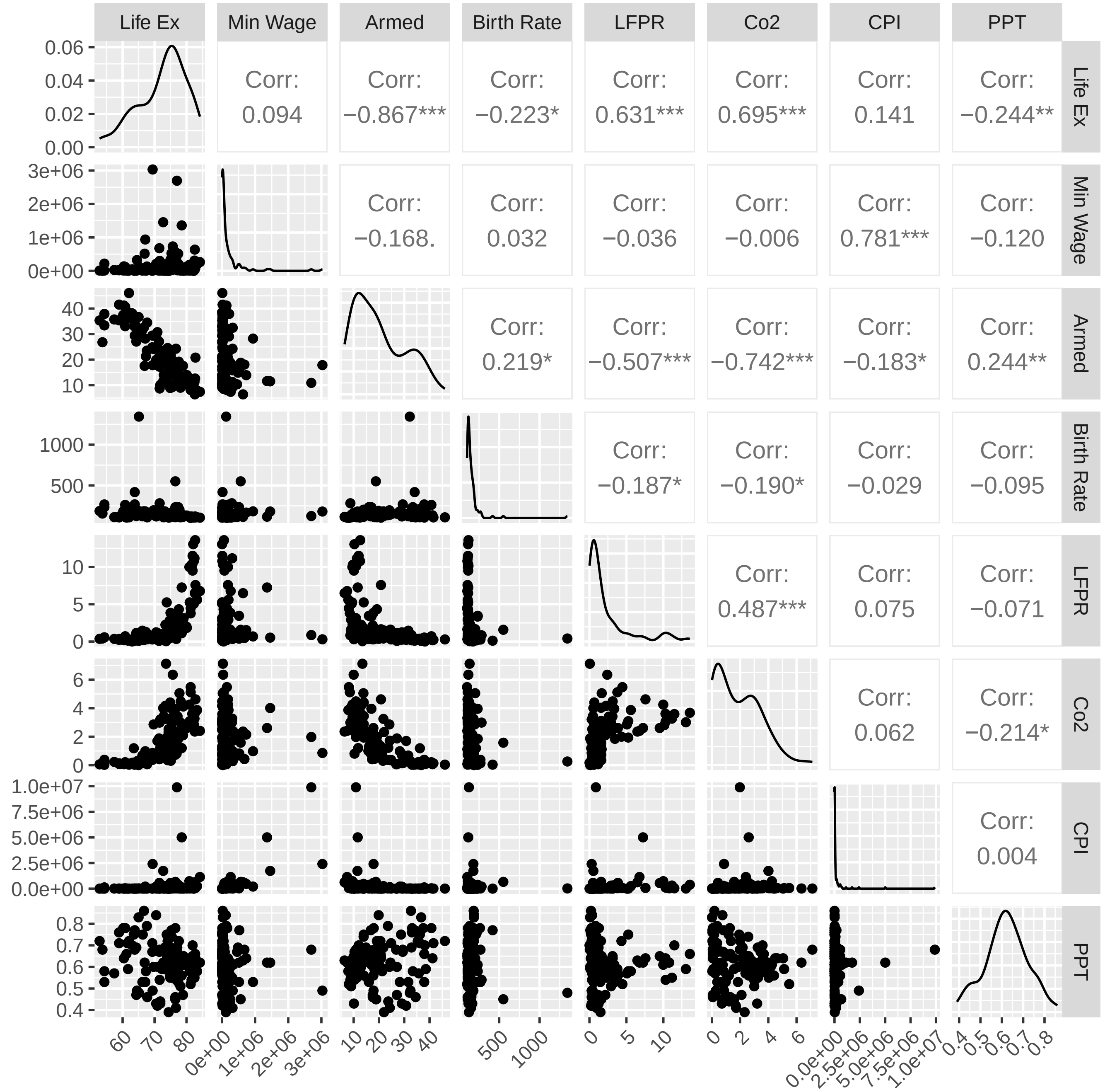
Parameter	μ	Median	Min	Max
Min Wage	2.30	0.98	0.01	13.59
Armed Forces	185900.00	37000.00	1000.00	3031000.00
CPI	160.25	132.11	101.87	1344.19
Birth Rate	20.24	17.83	6.40	46.08
LFPR	0.62	0.62	0.39	0.86
Co2 Emissions	258887.34	24079.00	297.00	9893038.00
PPT	1.87	1.58	0.01	7.12
Life Expectancy	72.46	74.35	52.80	84.20

Table 1: Summary of Country Data Variables

Regarding Minimum Wage, the mean stands at 2.30, with a median of 0.98, spanning from a minimum of 0.01 to a maximum of 13.59. Armed Forces size exhibits a mean of 185,900 and a median of 37,000, with values ranging from 1,000 to 3,031,000. Moving on to CPI, it showcases a mean of 160.25 and a median of 132.11, fluctuating between 101.87 and 1,344.19. Birth Rate, meanwhile, demonstrates a mean of 20.24 and a median of 17.83, with a range spanning from 6.40 to 46.08.LFPR has a mean of 0.62, a median of 0.62, and values spanning from 0.39 to 0.86. CO2 emissions present a mean of 258,887.34 and a median of 24,079, with emissions varying between 297 and 9,893,038. Physician per thousand shows a mean of 1.87 and a median of 1.58, with values ranging from 0.01 to 7.12.Finally, Life expectancy exhibits a mean of 72.46 years and a median of 74.35 years, with a range extending from 52.80 to

84.20 years.

```
custom_labels <- c("Life Ex", "Min Wage", "Armed",
                   "Birth Rate", "LFPR","Co2", "CPI", "PPT")

ggpairs(life.df, columns = c(16,4,5,7,18,20,6,22),
        columnLabels = custom_labels) + theme(axis.text.x =
        element_text(angle=45, hjust = 1))
```



```
cor(life.df$Minimum_wage, life.df$CPI)

## [1] -0.1866092
```

There is a strong positive correlation coefficient of 0.781 between CPI and minimum wage, this implies that as CPI tends to increase, minimum wage increases greatly.

```
cor(life.df$Armed_Forces_size, life.df$CPI)

## [1] 0.03227521
```

There is a moderate negative correlation coefficient of -0.183 between armed forces size and CPI, this implies that as armed forces Size tends to increase, CPI decreases.

```
cor(life.df$Birth_Rate, life.df$Armed_Forces_size)

## [1] -0.1681428
```

There is a moderate positive correlation coefficient of 0.219 between armed forces size and birth rate, this implies that as armed forces size tends to increase, Birth Rate increases.

```
cor(life.df$LFPR, life.df$Armed_Forces_size)

## [1] -0.1196188
```

There is a strong negative correlation coefficient of -0.507 between log1p(LFPR) and armed forces size, this implies that as log1p(LFPR) tends to increase, armed forces size decreases greatly

```
cor(life.df$Co2_Emissions, life.df$Armed_Forces_size)

## [1] 0.7806493
```

There is an extremely strong negative correlation coefficient of -0.742 between Co2 and armed forces size, this implies that as Co2 emissions tends to increase, Armed Forces Size decreases by an extreme amount.

```
cor(life.df$Physicians_per_thousand, life.df$Armed_Forces_size)

## [1] -0.00625427
```

There is a moderate positive correlation coefficient of 0.244 between physicians per thousand people and armed forces size, this implies that as Physicians per thousand people tends to increase, Armed Forces size increases.

```
cor(life.df$LFPR, life.df$Birth_Rate)

## [1] 0.2437697
```

There is a moderate negative correlation coefficient of -0.187 between log1p(LFPR) and Birth rate, this implies that as log1p(LFPR) increases, Birth rate tends to decrease.

```
cor(life.df$Co2_Emissions, life.df$Birth_Rate)

## [1] -0.1829763
```

There is a moderate negative correlation coefficient of -0.190 between Co2 emissions and birth rate, this implies that as CO2 emissions tends to increase, Birth rate tends to decrease.

```
cor(life.df$Co2_Emissions, life.df$LFPR)

## [1] 0.003862391
```

There is a strong positive correlation coefficient of 0.487 between CO2 emissions and log1p(LFPR), this implies that as CO2 emissions tends to increase, log1p(LFPR) increases greatly.

```
cor(life.df$Physicians_per_thousand, life.df$Co2_Emissions)

## [1] 0.06168127
```

There is a moderate negative correlation coefficient of -0.214 between Physicians per thousand people and CO2 emissions, this implies that as physicians per thousand people tends to increase, CO2 emissions decreases.

```
cor(life.df$Life_expectancy, life.df$Armed_Forces_size)

## [1] 0.09352201
```

There is an extremely strong negative correlation coefficient of -0.867 between Life expectancy and Armed forces size, this implies that as life expectancy tends to increase, Armed forces size decreases greatly.

```
cor(life.df$Life_expectancy, life.df$Birth_Rate)

## [1] -0.8672152
```

There is a moderate negative correlation coefficient of -0.223 between Life expectancy and Birth rate, this implies that as life expectancy increases, birth rate tends to decrease.

```
cor(life.df$Life_expectancy, life.df$LFPR)

## [1] -0.2440248
```

There is a strong positive correlation coefficient of 0.631 between Life expectancy and Labor force participation rate, this implies that as life expectancy tends to increase, labor force participation rate increases greatly.

```
cor(life.df$Life_expectancy, life.df$Co2_Emissions)

## [1] 0.14074
```

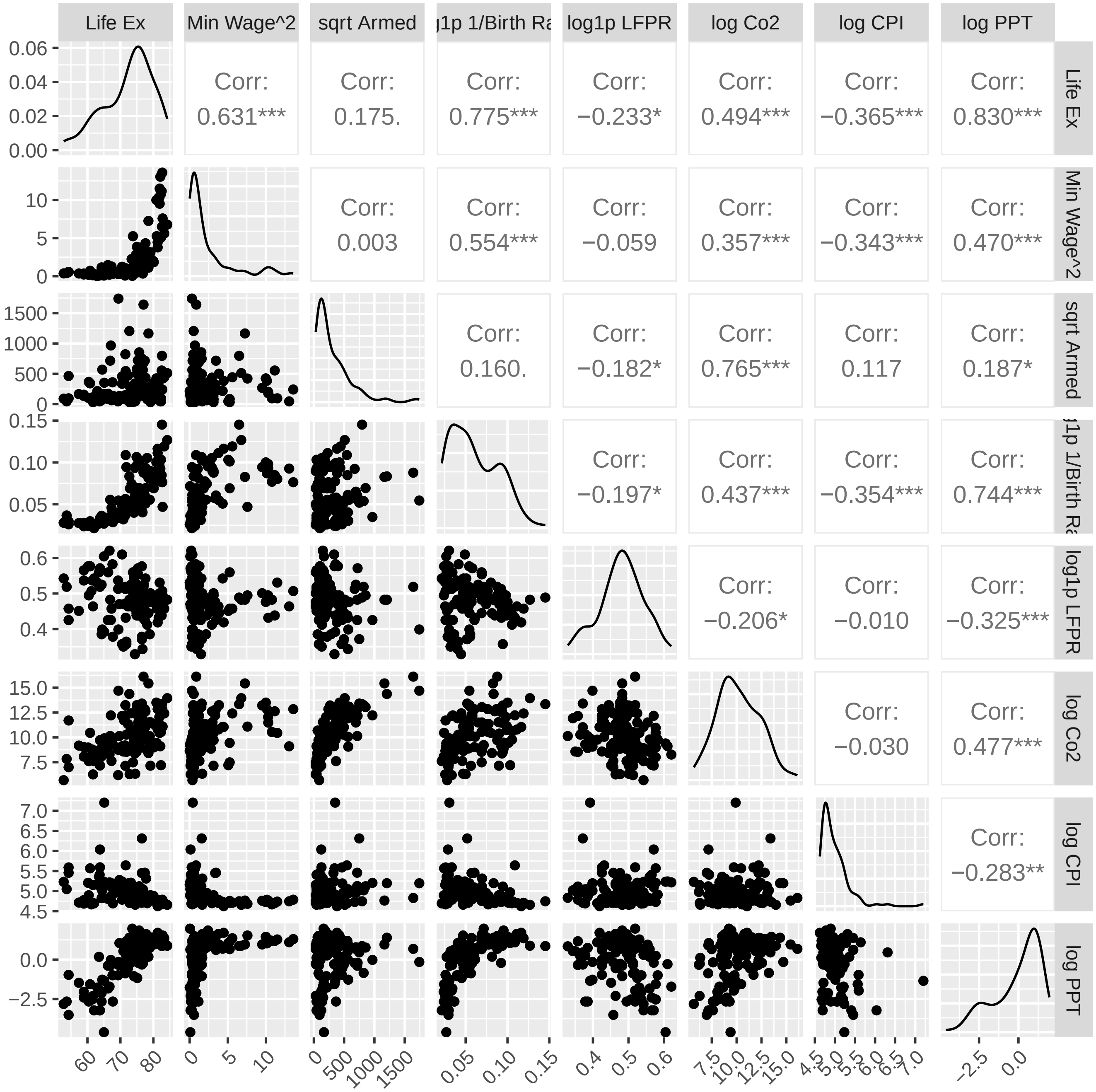
There is a strong positive correlation coefficient of 0.695 between Life expectancy and Co2 emissions, this implies that as life expectancy tends to increase, CO2 emissions increases greatly.

```
cor(life.df$Life_expectancy, life.df$Physicians_per_thousand)

## [1] 0.6946879
```

There is a moderate negative correlation coefficient of -0.244 between life expectancy and physicians per thousand people, this implies that as life expectancy tends to increase, physicians per thousand people tends to decrease.

We preformed several transformations on the data in order to fit assumptions that must be met. We deemed it necessary to assess the correlation between the variables with transformations included.

```
custom_labels <- c("Life Ex", "Min Wage^2", "sqrt Armed",
          "log1p 1/Birth Rate", "log1p LFPR", "log Co2", "log CPI", "log PPT")

data_df <- model.frame(Life_expectancy ~ Minimum_wage^2 + sqrt(Armed_Forces_size) +
                         log1p(1/Birth_Rate) + log1p(LFPR) + log(Co2_Emissions) +
                         log(CPI) + log(Physicians_per_thousand), data = life.df)

ggpairs(data_df, columns = 1:8, columnLabels = custom_labels) + theme(axis.text.x =
        element_text(angle=45, hjust = 1))
```



These correlation coefficients are important to keep in mind when analyzing the impact of these factors on life expectancy.

Given these correlation coefficients, we believe it is best to calculate the VIF for these variables to ensure there is no multicollinearity, this is completed in the next section.

4 Methods

4.1 Fitting ANOVA

We have decided to fit an ANOVA regression using our first-order linear model to determine what variables will be the most significant when predicting life expectancy. Below is our ANOVA for regression hypothesis:

$H_O : \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = \beta_7 = 0$

H_a : at least one $\beta_n \neq 0, n = 1, 2, 3, ..., 7$

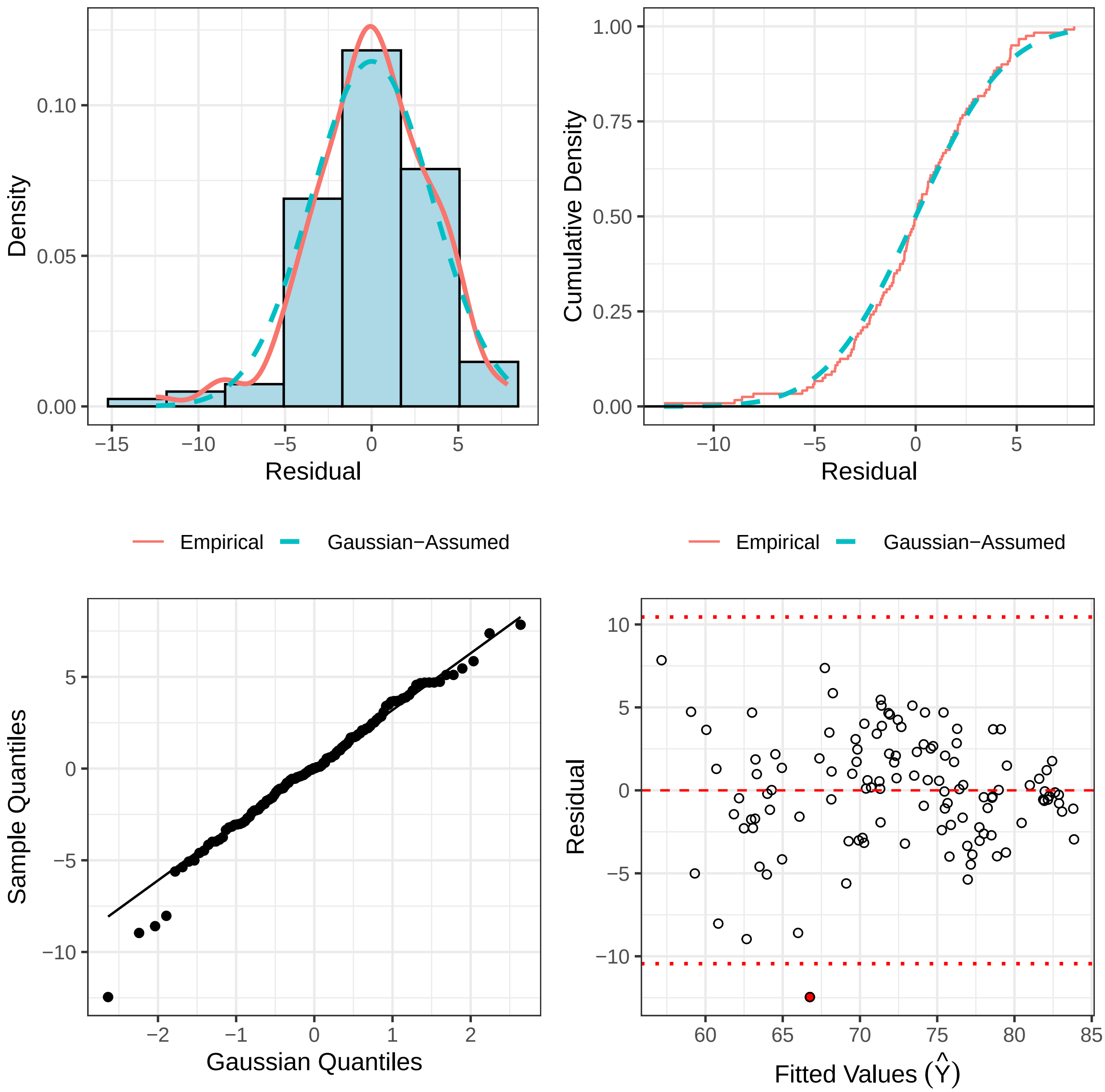
```
anova.df <- anova(lm(Life_expectancy~ Minimum_wage^2 + sqrt(Armed_Forces_size)
                     + log1p(1/Birth_Rate)+ log1p(LFPR) +log(Co2_Emissions)
                     + log(CPI) + log(Physicians_per_thousand), data=life.df))

anova.table<-(xtable(anova.df))
caption(anova.table) <- "Regression summary with standarized predictors"
#anova.table
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Minimum_wage^2	1	2613.20	2613.20	215.49	0.0000
sqrt(Armed_Forces_size)	1	196.67	196.67	16.22	0.0001
log1p(1/Birth_Rate)	1	1554.86	1554.86	128.22	0.0000
log1p(LFPR)	1	49.25	49.25	4.06	0.0463
log(Co2_Emissions)	1	99.54	99.54	8.21	0.0050
log(CPI)	1	41.94	41.94	3.46	0.0656
log(Physicians_per_thousand)	1	640.25	640.25	52.80	0.0000
Residuals	112	1358.21	12.13		

Table 2: Regression summary with standarized predictors

From the ANOVA regression table, there is evidence to support that minimum wage, armed forces size, and birth rate are significant predictor variables for life expectancy. This is with a significance of p-value < 0.0001. This also supplies us with sufficient information to distinguish physician population per thousand, Co2 emissions, and CPI as less significant predictors, as they do not meet our chosen significance level of 0.05.

4.2 First-Order Model and Model Selection



```
## [1] 0.7798118
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	71.5268	6.3918	11.19	0.0000
Minimum_wage^2	0.5203	0.1362	3.82	0.0002
sqrt(Armed_Forces_size)	-0.0004	0.0019	-0.20	0.8420
log1p(1/Birth_Rate)	56.7292	18.2614	3.11	0.0024
log1p(LFPR)	0.3326	5.6358	0.06	0.9530
log(Co2_Emissions)	0.3110	0.3015	1.03	0.3046
log(CPI)	-1.3680	1.0134	-1.35	0.1798
log(Physicians_per_thousand)	2.5713	0.3539	7.27	0.0000

Table 3: Anova Table for regression

The linear regression formula for this model gives insight on how effective each variable is at predicting life expectancy. The further the t-value is from zero,regardless of direction, indicates to us that a variable will be a more effective predictor than one closer to zero.It is clear that minimum wage(t-value = 3.82, p-value = 0.0002), physicians per thousand(t-value = 7.27, p-value = 0.0000), and birth rate(t-value = 3.82, p-value = 0.0002) are all the most significant predictors for life expectancy.Some of the least significant predictor effects in this first model are Labor Force Participation Rate(t-value = 0.06, p-value = 0.9530) and Co2 emissions(t-value = 1.03, p-value = 0.3046). These values are least expected to affect life expectancy which is Suprising.The adjusted R^2 value for our model is 0.7798 This means that approximately 77.981% of the variability of life expectancy is explained by this linear regression model.

The assumptions for regression appear to be met. The response variable, life expectancy, is continuous and quantitative. Life expectancy is measured in the number of years someone lives, which is quantitative and is also continuous since we get older every second. Because this sample contains statistics from all countries worldwide, we can conclude that the data is representative of the population. Sample collection methods are not detailed by the creator of the data set, but given the size of the data randomness is assumed.

To begin, we did a variation of transformation on our data to increase normality. After the transformations, the residuals seem to have constant variance. When looking at the residual plot, there is a clear random dispersal of points across the middle line at 0. Only one point lies outside the banded region, which is acceptable. Finally, there is no break in the pattern we see in the plot.

The residual plots also are approximately normal. The residual density is bell curved and is close to the Gaussian-Assumed curve. The points in the q-q plot follow a straight line except for some skew on the left end.This is not ideal but we can still move forward with this in mind.

We calculated the VIF for both the original and transformed data to ensure there is no multicollinearity.

```r
vif(one.way)
```

```
##                 Minimum_wage      sqrt(Armed_Forces_size)
##                     1.768745                     3.143761
##          log1p(1/Birth_Rate)                  log1p(LFPR)
##                     2.639761                     1.156880
##           log(Co2_Emissions)                     log(CPI)
##                     4.014246                     1.237469
## log(Physicians_per_thousand)
##                     2.650392
```

```r
vif(one.way.pretrans)
```

```
##            Minimum_wage       Armed_Forces_size              Birth_Rate
##                1.447943                2.842946                2.647141
##                    LFPR           Co2_Emissions                     CPI
##                1.139100                2.728368                1.088854
## Physicians_per_thousand
##                2.412176
```

Our rigorous regression analysis, including a VIF test on the model, has yielded reassuring results. All VIF values fall within the optimal range of 1 to 4, indicating that the predictor values do not show signs of multicollinearity. With all the assumptions for regression met, we can be confident in the validity of our results.

```r
intercept.model <- lm(Life_expectancy ~ 1, data = life.df)
one.way.forward <- stepAIC(intercept.model,
        direction = "forward",
        scope = list(upper = one.way),
        trace = FALSE)
one.way.backward <- stepAIC(one.way,
        direction = "backward",
        trace=FALSE)
one.way.both <- stepAIC(intercept.model,
        direction = "both",
        scope = list(upper = one.way),
        trace = FALSE)
```

```
metrics<-data.frame(Model=c("One-Way", "One-Way Backward", "One-Way Forward", "One-Way Both"),
                    R.squared = c(summary(one.way)$r.squared,
                                  summary(one.way.backward)$r.squared,
                                  summary(one.way.forward)$r.squared,
                                  summary(one.way.both)$r.squared),
                    R.adj.squared = c(summary(one.way)$adj.r.squared,
                                      summary(one.way.backward)$adj.r.squared,
                                      summary(one.way.forward)$adj.r.squared,
                                      summary(one.way.both)$adj.r.squared),
                    LL=c(logLik(one.way),
                         logLik(one.way.backward),
                         logLik(one.way.forward),
                         logLik(one.way.both)),
                    AIC=c(AIC(one.way), AIC(one.way.backward),
                          AIC(one.way.forward),
                          AIC(one.way.both)),
                    BIC=c(BIC(one.way), BIC(one.way.backward),
                          BIC(one.way.forward),
                          BIC(one.way.both)),
                    RMSE=c(sqrt(mean(sum(residuals(one.way)^2))),
                           sqrt(mean(sum(residuals(one.way.backward)^2))),
                           sqrt(mean(sum(residuals(one.way.forward)^2))),
                           sqrt(mean(sum(residuals(one.way.both)^2)))),
                    MAE=c(mean(sum(abs(residuals(one.way)))),
                          mean(sum(abs(residuals(one.way.backward)))),
                          mean(sum(abs(residuals(one.way.forward)))),
                                mean(sum(abs(residuals(one.way.both))))),
                    Parameters=c("7","3", "3", "3"))
metric.table <- xtable(metrics)
caption(metric.table) <- "A table of model metrics for our first order models."
#metric.table
```

Model	R.squared	R.adj.squared	LL	AIC	BIC	RMSE	MAE	Parameters
One-Way	0.79	0.78	-315.86	649.72	674.80	36.85	309.01	7
One-Way Backward	0.79	0.78	-317.67	645.33	659.27	37.41	314.01	3
One-Way Forward	0.79	0.78	-317.67	645.33	659.27	37.41	314.01	3
One-Way Both	0.79	0.78	-317.67	645.33	659.27	37.41	314.01	3

Table 4: A table of model metrics for our first order models.

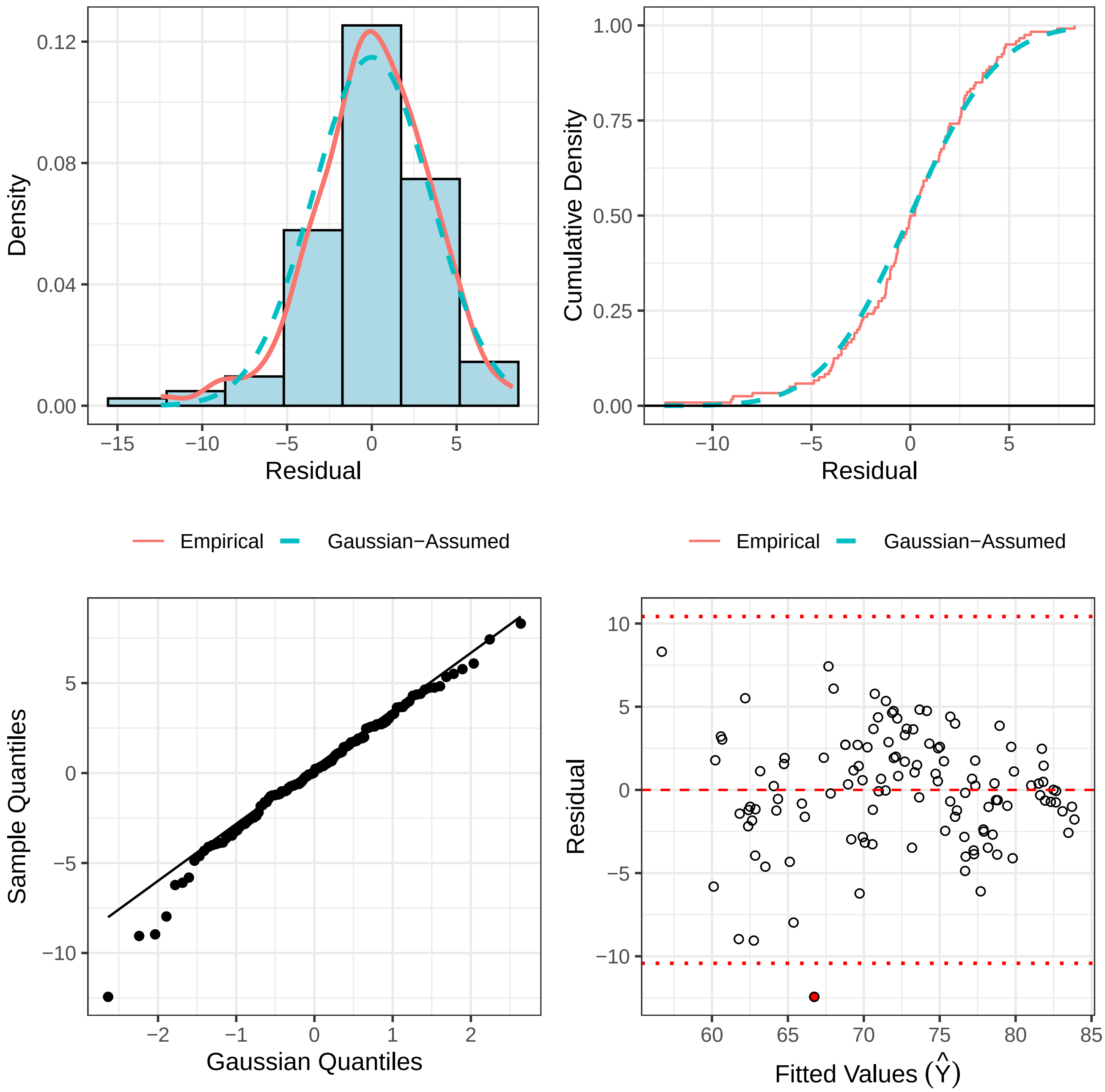
After running the one one way backwards, forward, and both models, it becomes apparent that they chose the same predictor variables which is why the factors are the same. However, you can see that compared to our original model, the step wise models all have a reduction in AIC and BIC values. Additionally, these three new models showed an increase in our R^2 from 0.779 to 0.7809.

```
summtable2<- xtable(summary(one.way.backward))
caption(summtable2)<- ":Regression summary for model selection"
#summtable2
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	67.3979	1.0777	62.54	0.0000
Minimum_wage^2	0.5878	0.1233	4.77	0.0000
log1p(1/Birth_Rate)	62.9137	17.8743	3.52	0.0006
log(Physicians_per_thousand)	2.7029	0.3261	8.29	0.0000

Table 5: Regression summary for model selection

Looking at our one way backward analysis table, minimum wage, birth rate, and physicians per thousand all have a significance effect on life expectancy. Minimum wage and birth rate are both the highest predictor variables with their p-value < 0.0001). There is a negative correlation between birth rate and life expectancy as denoted by the coefficient -0.4199. Physicians per thousand falls behind with a p-value $= 0.0273$. We did anticipate physicians per thousand to be in the final model as more investments in healthcare could increase life expectancy.



```
##              Minimum_wage         log1p(1/Birth_Rate)
##                  1.457842                    2.541638
## log(Physicians_per_thousand)
##                  2.261739
```

With this new model, the residuals still appear to be approximately normal. The residual density bell curve seems to follow the Gaussian assumed closely. We are still able to proceed even though the left tail is pretty long on the residual density curve. The left tail on the q-q plot deviates more than is ideal however, this deviation is still acceptable. Finally the residual plot has constant variance and only one point is a significant outlier. The VIF values are still within the acceptable range of 1-3, therefore we can deduce there is no significant multicollinearity. Thus our model is not over fitted.

4.3 Interaction Analysis

To further understand the relationship between life expectancy our predictor variables we selected, we chose to run a two way ANOVA interaction analysis with our transformed data.

```
two.way <- lm(Life_expectancy ~ Minimum_wage^2 +
                sqrt(Armed_Forces_size) + log1p(1/Birth_Rate) * log1p(LFPR) +
                log(Co2_Emissions) + log(CPI) +
                log(Physicians_per_thousand), data=life.df)

#anova(two.way)

two.way.model <- lm(Life_expectancy ~
                     Minimum_wage^2 * sqrt(Armed_Forces_size) +
                     Minimum_wage^2 * log1p(1/Birth_Rate) +
                     Minimum_wage^2 * log1p(LFPR) +
                     Minimum_wage^2 * log(Co2_Emissions) +
                     Minimum_wage^2 * log(CPI) +
                     Minimum_wage^2 * log(Physicians_per_thousand) +
                     sqrt(Armed_Forces_size) * log1p(1/Birth_Rate) +
                     sqrt(Armed_Forces_size) * log1p(LFPR) +
                     sqrt(Armed_Forces_size) * log(Co2_Emissions) +
                     sqrt(Armed_Forces_size) * log(CPI) +
                     sqrt(Armed_Forces_size) * log(Physicians_per_thousand) +
                     log(CPI) * log1p(1/Birth_Rate) +
                     log(CPI) * log1p(LFPR) +
                     log(CPI) * log(Co2_Emissions) +
                     log(CPI) * log(Physicians_per_thousand) +
                     log(Physicians_per_thousand) * log1p(1/Birth_Rate) +
                     log(Physicians_per_thousand) * log1p(LFPR) +
                     log(Physicians_per_thousand) * log(Co2_Emissions) +
                     log(Co2_Emissions) * log1p(1/Birth_Rate) +
                     log(Co2_Emissions) * log1p(LFPR) +
                     log(Co2_Emissions) * sqrt(Armed_Forces_size) +
                     log(Co2_Emissions) * log(CPI) +
                     log(Co2_Emissions) * log(Physicians_per_thousand) +
                     log1p(LFPR) * log1p(1/Birth_Rate) +
                     log1p(LFPR) * sqrt(Armed_Forces_size) +
                     log1p(LFPR) * log(CPI) +
                     log1p(LFPR) * log(Physicians_per_thousand) +
                     log1p(1/Birth_Rate) * sqrt(Armed_Forces_size) +
                     log1p(1/Birth_Rate) * log(CPI) +
                     log1p(1/Birth_Rate) * log(Physicians_per_thousand),
                     data = life.df)
```

```
#anova(two.way.model)
```

```
two.way.table <- xtable(anova(two.way.model))
caption(two.way.table) <- "ANOVA table for regression of two-way interaction parameters."
#two.way.table
```

Parameter	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Minimum_wage^2	1	2613.20	2613.20	249.75	0.0000
sqrt(Armed_Forces_size)	1	196.67	196.67	18.80	0.0000
log1p(1/Birth_Rate)	1	1554.86	1554.86	148.60	0.0000
log1p(LFPR)	1	49.25	49.25	4.71	0.0326
log(Co2_Emissions)	1	99.54	99.54	9.51	0.0027
log(CPI)	1	41.94	41.94	4.01	0.0483
log(Physicians_per_thousand)	1	640.25	640.25	61.19	0.0000
Minimum_wage^2:sqrt(Armed_Forces_size)	1	0.06	0.06	0.01	0.9400
Minimum_wage^2:log1p(1/Birth_Rate)	1	26.87	26.87	2.57	0.1125
Minimum_wage^2:log1p(LFPR)	1	5.79	5.79	0.55	0.4591
Minimum_wage^2:log(Co2_Emissions)	1	14.08	14.08	1.35	0.2491
Minimum_wage^2:log(CPI)	1	2.03	2.03	0.19	0.6606
Minimum_wage^2:log(Physicians_per_thousand)	1	12.53	12.53	1.20	0.2766
sqrt(Armed_Forces_size):log1p(1/Birth_Rate)	1	0.89	0.89	0.09	0.7707
sqrt(Armed_Forces_size):log1p(LFPR)	1	0.27	0.27	0.03	0.8726
sqrt(Armed_Forces_size):log(Co2_Emissions)	1	14.73	14.73	1.41	0.2385
sqrt(Armed_Forces_size):log(CPI)	1	10.73	10.73	1.03	0.3139
sqrt(Armed_Forces_size):log(Physicians_per_thousand)	1	1.47	1.47	0.14	0.7088
log1p(1/Birth_Rate):log(CPI)	1	10.38	10.38	0.99	0.3220
log1p(LFPR):log(CPI)	1	1.20	1.20	0.11	0.7360
log(Co2_Emissions):log(CPI)	1	0.07	0.07	0.01	0.9344
log(CPI):log(Physicians_per_thousand)	1	14.16	14.16	1.35	0.2478
log1p(1/Birth_Rate):log(Physicians_per_thousand)	1	238.06	238.06	22.75	0.0000
log1p(LFPR):log(Physicians_per_thousand)	1	10.94	10.94	1.05	0.3092
log(Co2_Emissions):log(Physicians_per_thousand)	1	25.69	25.69	2.46	0.1206
log1p(1/Birth_Rate):log(Co2_Emissions)	1	1.30	1.30	0.12	0.7254
log1p(LFPR):log(Co2_Emissions)	1	1.05	1.05	0.10	0.7519
log1p(1/Birth_Rate):log1p(LFPR)	1	13.74	13.74	1.31	0.2549
Residuals	91	952.17	10.46		

Table 6: ANOVA table for regression of two-way interaction parameters.

When analyzing the models summary statistics we concluded that all our independent variables, such as minimum wage (p=0.0), armed forces size (p=0.0), birth rate (p=0.0), LFPR (p=0.0326), Co2 emissions (p=0.0027), CPI (p=0.0483), and physicians per thousand .

The most significant interaction of variables we see form this table are birth rate and physicians per thousand (F=22.75 and p = 0.000). This implies that as birth rate increases, there is a strong correlation to physicians per thousand increasing. This can be explained by less pregnancy complications due to a higher concentration of physicians.

```r
summarytable2way <- xtable(summary(two.way))
caption(summarytable2way) <- "Two-Way interaction model results"
#summarytable2way
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	78.9066	8.5956	9.18	0.0000
Minimum_wage^2	0.4932	0.1375	3.59	0.0005
sqrt(Armed_Forces_size)	-0.0007	0.0019	-0.35	0.7270
log1p(1/Birth_Rate)	-104.1673	127.0502	-0.82	0.4140
log1p(LFPR)	-16.6277	14.3964	-1.15	0.2506
log(Co2_Emissions)	0.3009	0.3008	1.00	0.3193
log(CPI)	-1.2682	1.0136	-1.25	0.2135
log(Physicians_per_thousand)	2.4489	0.3656	6.70	0.0000
log1p(1/Birth_Rate):log1p(LFPR)	356.7258	278.7764	1.28	0.2033

Table 7: Two-Way interaction model results

4.4 Model Selection

As seen in the model above, only one of our two-way interaction parameters has statistical significance. This is between birth-rate and physician population per thousand people, which is logical as a higher amount of physicians should lead to less pregnancy complications. The more of these parameters that are included in the model the higher our R^2 value will become. The more parameters you add to a model the higher R^2 becomes, however, each variable becomes less significant to the outcome of the model and the model will lose reliability. In addition to R^2 we used metrics unaffected by the number of parameters used in the model. These metrics include R_a^2dj, log likelihood (LL), AIC, BIC, RMSE, MAE, LOOCV, LOOCV R^2,LOOCV RMSE, and LOOCV MAE. The numbers of parameters used was excluded as a metric,as it is already encompassed within AIC, BIC, and any metric including R^2. The more metrics we choose to analyze, the easier it will be to determine which model is the most consistently effective and accounts for possible success in only one random metric.

Next, as a method of further analysis we will repeat our forward, backwards, and bidirectional regression models for our two-way interaction data. This has the same assumptions met as the models we created for the one-way data, and we will proceed the same way.

```r
two.way.backward <- stepAIC(two.way,
                             direction = "backward",
                             trace=FALSE)
two.way.forward <- stepAIC(intercept.model,
                            direction = "forward",
                            scope = list(upper = two.way),
                            trace=FALSE)
two.way.both <- stepAIC(intercept.model,
                         direction = "both",
                         scope = list(upper = two.way),
                         trace=FALSE)
```

Now, we have 8 models to choose from. Forward,backward, and bidirectional regressions for One and Two way data, as well as our models for one and two-way interaction. However, many of these two-way models contain redundancy as seen in the following table.

Model	R.squared	R.adj.squared	LL	AIC	BIC	RMSE	MAE	Parameters
Two-Way	0.80	0.78	-314.98	649.96	677.83	36.59	305.23	7
Two-Way Backward	0.79	0.78	-316.40	645.33	659.27	37.02	308.16	5
Two-Way Forward	0.79	0.78	-317.67	645.33	659.27	37.41	314.01	5
Two-Way Both	0.79	0.78	-317.67	645.33	659.27	37.41	314.01	3

Table 8: A table of model metrics for our first order models.

After Metric Analysis we see that the two-way forward and two-way bidirectional regressions are identical, so we can exclude the forward model. In regards to the one-way models, we see that all of the regression models, except the One-way bidirectional regression ,are identical, so we can remove the forward and backward models. Now we have subtracted 3 models from our total bringing us to 5 possible models to choose from. Having 5 models is sufficient for us to pick from, however we would like to increase our options. When looking at AIC on our forward, backward, and bidirectional regression models, we can see that none of the AIC values appear to be particularly low, and we believe that we can achieve a model with a lower value by other means. To further analyze our options we will use a LASSO regression on our interaction regression models. The LASSO regression using the same additional assumptions as other analyses we have used this far, however every different test or analyses can introduce a new bias to your potential results, and LASSO is not immune to this. LASSO (Least Absolute Shrinkage and Selection Operator) regression is a parameter selection technique used to regularize against multicollinearity, over fitting, and interpret-ability. Using cross validation, LASSO finds an ideal tuning parameter (lambda). LASSO accomplishes this by minimizing the sum of the absolute values of the regression coefficients. This allows LASSO to set parameter to coefficient values to zero, giving it the ability to eliminate ineffective parameters and optimize coefficient values at the same time.

```
one.x <- model.matrix(one.way)[, -1]
one.y <- life.df$Life_expectancy
one.way.lasso <- glmnet(one.x, one.y, alpha = 1)
one.way.ridge <- glmnet(one.x, one.y, alpha = 0)
two.x <- model.matrix(two.way)[, -1]
two.y <- life.df$Life_expectancy
two.way.lasso <- glmnet(two.x, two.y, alpha = 1)
two.way.ridge <- glmnet(two.x, two.y, alpha = 0)
```

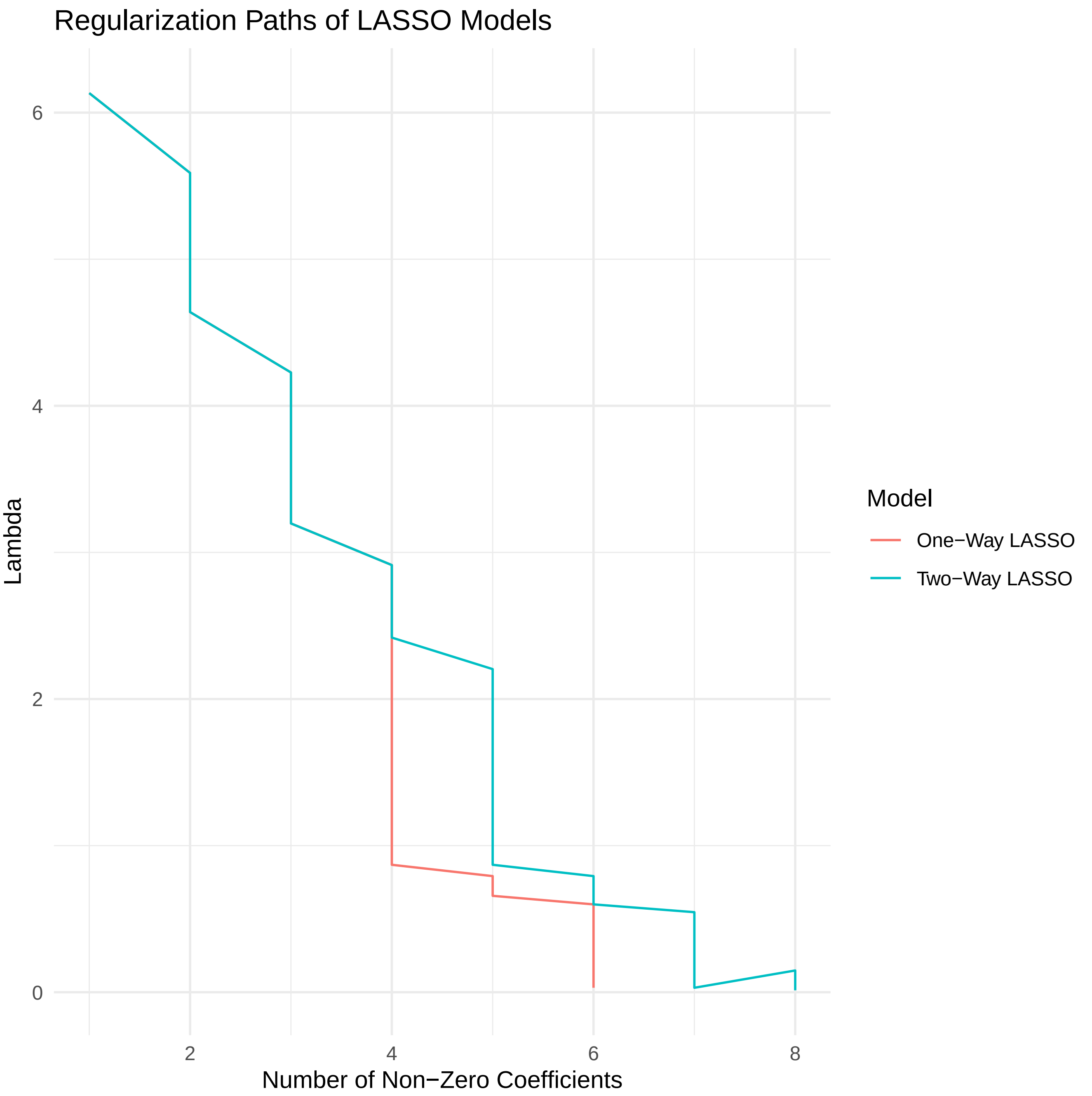
Comparing the lambda values for each model across different numbers of parameters will be useful in identifying bias from the LASSO analysis. We compare them using this code to create a data frame (Fenwick et al. (2023)).

```
extract_coefs <- function(model, model_name) {
  lambda <- model$lambda
  coefs <- coef(model, s = lambda)
  num_nonzero <- numeric(length(lambda))
  for (i in seq_along(lambda)) {
    num_nonzero[i] <- sum(coefs[, i] != 0)
  }
  if (length(lambda) != ncol(coefs)) {
    print("Warning: Length of lambda does not match the number of columns in the coefficient matrix.")
  }
  data.frame(lambda = lambda, num_nonzero = num_nonzero, model = model_name)
}

one.way.lasso.data <- extract_coefs(one.way.lasso, "One-Way LASSO")
two.way.lasso.data <- extract_coefs(two.way.lasso, "Two-Way LASSO")

lasso_data <- rbind(one.way.lasso.data, two.way.lasso.data)
```

```
extract_coefs <- function(model, model_name) { lambda <- model$lambda
coefs <- coef(model, s = lambda)
  num_nonzero <- numeric(length(lambda))
for (i in seq_along(lambda)) {
}
data.frame(lambda = lambda, num_nonzero = num_nonzero - 1, model = model_name) }
one.way.ridge.data <- extract_coefs(one.way.ridge, "One-Way RIDGE")
two.way.ridge.data <- extract_coefs(two.way.ridge, "Two-Way RIDGE")
```

The RIDGE and LASSO regressions make an efficient way to analyze changes in multicollinearity by creating a penalty term lambda to adjust values depending on how many predictor values you have. As you increase the number of parameters, lambda will shrink towards zero.

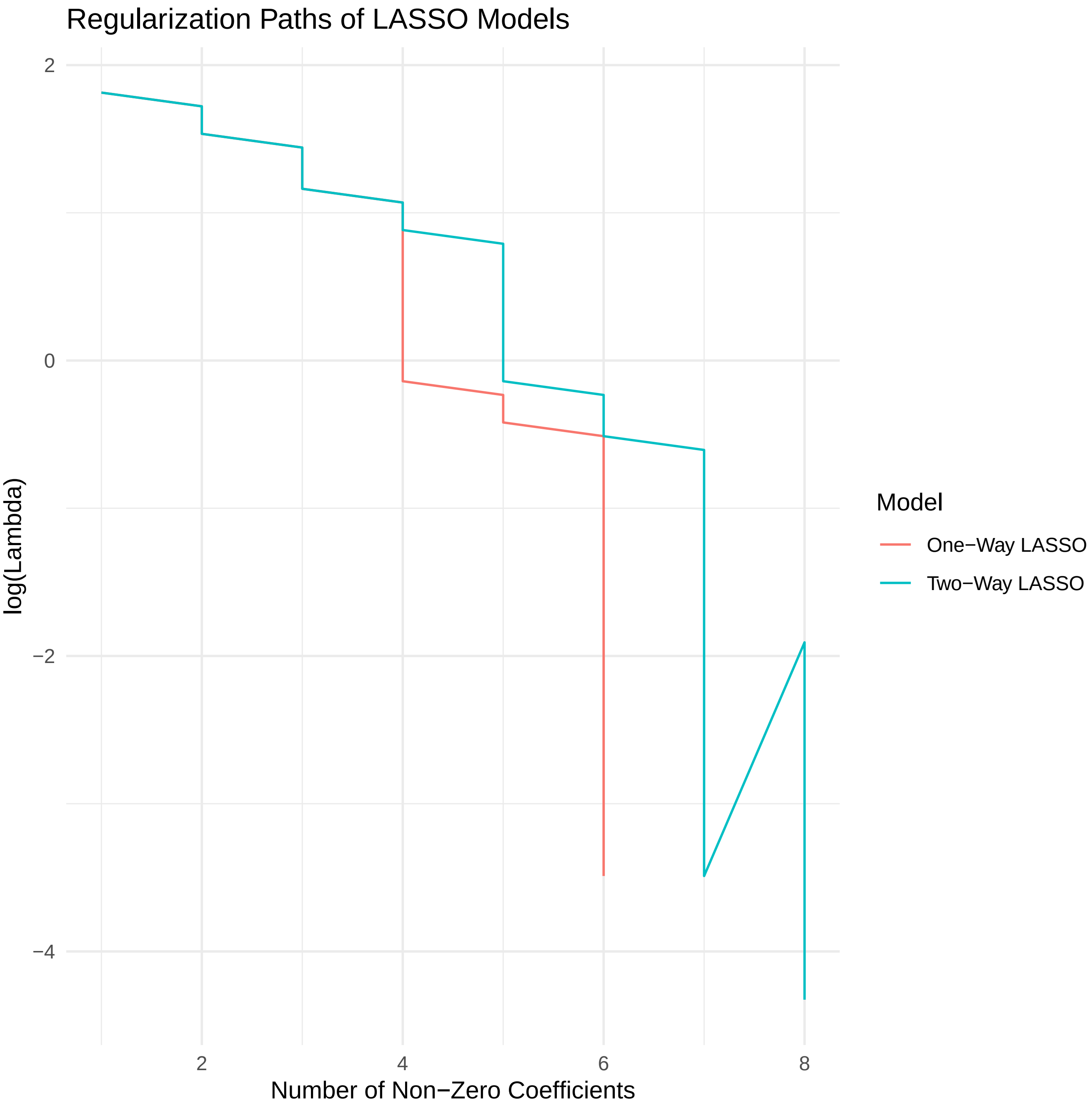
```
lambdaplot<- ggplot(lasso_data, aes(x = num_nonzero, y = lambda, color = model)) +
  geom_line() +
  labs(title = "Regularization Paths of LASSO Models",
       x = "Number of Non-Zero Coefficients",
       y = "Lambda",
       color = "Model") +
  theme_minimal()

lambdaplot
```



The graph above shows the relationship between lambda and non-zero coefficients. The scaling of the Y axis makes it difficult to read the effect of lambda on large numbers of non zero parameters. To fix this we will use a logarithmic transformation on the Lambda values.

```
lambdaplot<- ggplot(lasso_data, aes(x = num_nonzero, y = log(lambda), color = model)) +
  geom_line() +
  labs(title = "Regularization Paths of LASSO Models",
       x = "Number of Non-Zero Coefficients",
       y = "log(Lambda)",
       color = "Model") +
  theme_minimal()
```

```
lambdaplot
```



This graph shows the changes between the number of coefficients more clearly due to the log transformation. We do see a large lambda value for a low number of non-zero coefficients, and a low lambda for the number of coefficients we would like to use, which is between four and six. We can now move on to our next model option, using best subsets.

Best subsets uses a binary selection algorithm to determine which model is the best by essentially brute forcing every combination of possible parameters. Parameters can either be included or not included giving it two distinct options when choosing parameters. Though a limited number of parameters used in a model such as ours, using a function such as bestglm() is feasible as there is a limited computing time. For larger data sets with more quantitative variables and columns will take much longer. The best subsets method is the easiest way to find the best model while doing slightly more complex coding.

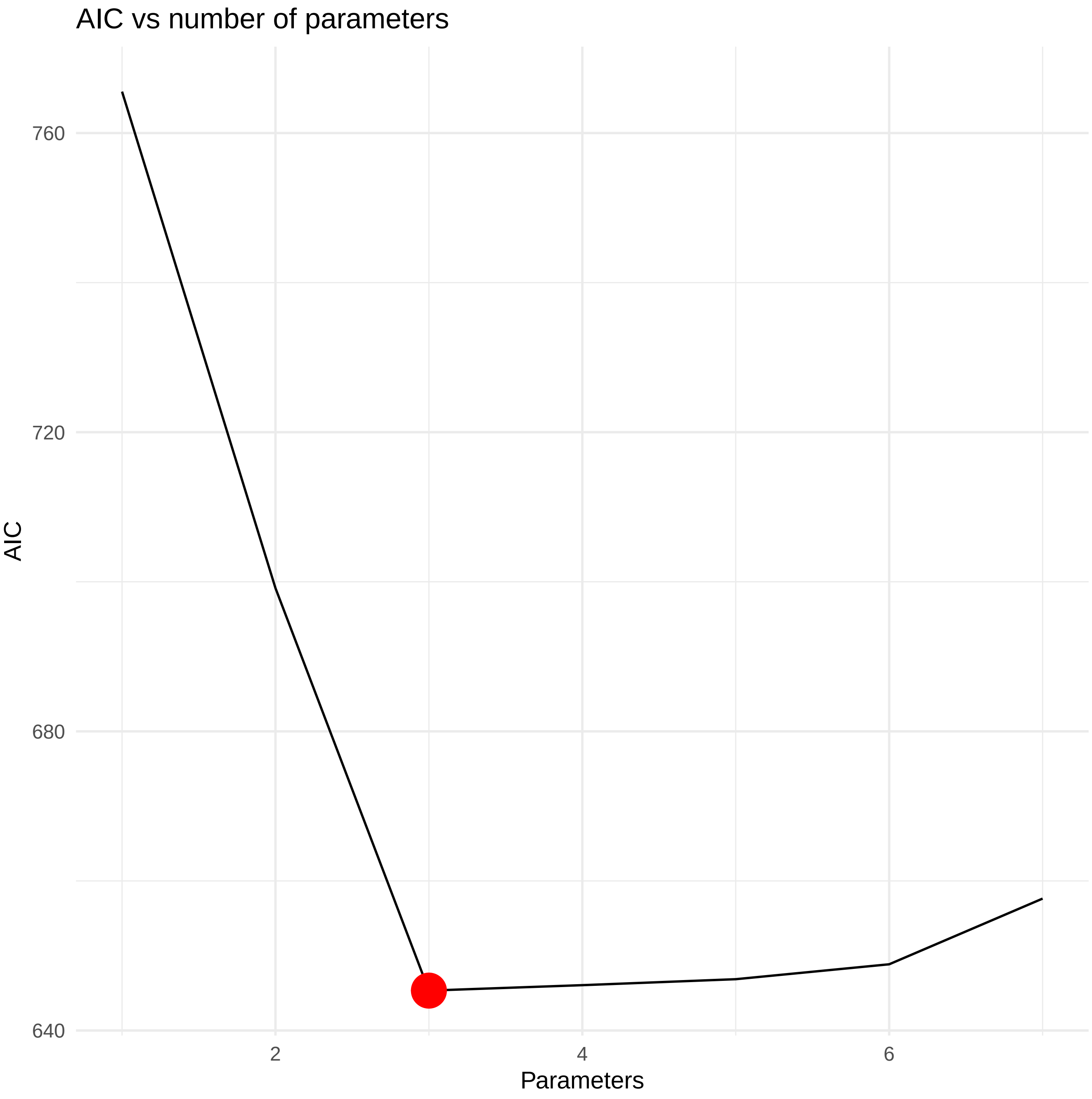
```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
##                  (Intercept)                   Minimum_wage
##                      67.3979                         0.5878
##          log1p.1.Birth_Rate.  log.Physicians_per_thousand.
##                      62.9137                         2.7029
```

This iteration of the bestglm() function uses AIC as its focus metric. Its goal will be to minimize the AIC of each parameter.

```
regsubsets.out <- regsubsets(Life_expectancy ~ Minimum_wage^2 + (log1p(1/(Birth_Rate))) + log(Physicians_pe
#xtable(as.data.frame(summary(regsubsets.out)£outmat))
subset1 <- lm(Life_expectancy ~ Minimum_wage^2, data = life.df)
subset2 <- lm(Life_expectancy ~ Minimum_wage^2 + (log1p(1/(Birth_Rate))),
              data = life.df)
subset3 <- lm(Life_expectancy ~ Minimum_wage^2 + (log1p(1/(Birth_Rate))) +
                log(Physicians_per_thousand), data = life.df)
subset4 <- lm(Life_expectancy ~ Minimum_wage^2 + (log1p(1/(Birth_Rate))) +
                log(Physicians_per_thousand) + log(CPI), data = life.df)
subset5 <- lm(Life_expectancy ~ Minimum_wage^2 + (log1p(1/(Birth_Rate))) +
                log(Physicians_per_thousand) + log(CPI) + sqrt(Armed_Forces_size), data = life.df)
subset6 <- lm(Life_expectancy ~ Minimum_wage^2 + (log1p(1/(Birth_Rate))) +
                log(Physicians_per_thousand) + log(CPI) + sqrt(Armed_Forces_size) + log1p(LFPR), data = lif
subset7 <- lm(Life_expectancy ~ Minimum_wage^2+
                log(Physicians_per_thousand) + log(CPI) + sqrt(Armed_Forces_size) + log1p(LFPR) + log(Co2_E
              data = life.df)
aic_values <- c(AIC(subset1), AIC(subset2), AIC(subset3), AIC(subset4),
                AIC(subset5), AIC(subset6), AIC(subset7))
num_parameters <- c(1,2,3,4,5,6,7)
aic_data <- data.frame(
  Model = paste("Subset", 1:7),
  NumParameters = num_parameters,
  AIC = aic_values
)
lowest_aic_subset <- which.min(aic_data$AIC)
size_multiplier <- ifelse(1:7 == lowest_aic_subset, 5, 1)
```

	Min Wage^2	log1p(1/(Birth))	log(PPT)	log(CPI)	sqrt(Armed)	log1p(LFPR)	log(Co2)
1			*				
2	*		*				
3	*	*	*				
4	*	*	*				*
5	*	*	*	*			*
6	*	*	*	*	*		*
7	*	*	*	*	*	*	*

```
best_params <- aic_data[which.min(aic_data$AIC), ]

plot2 <- ggplot(aic_data, aes(x = NumParameters, y = AIC)) +
  geom_line() +
  geom_point(data = best_params, aes(x = NumParameters, y = AIC), color = "red", size = 7) +  labs(title =
       x = "Parameters",
       y = "AIC") +
  theme_minimal()

plot2
```



Because the model with the lowest AIC was the subset of three parameters, it is our best subset. This is our One-Way Best model. Next we have chosen to conduct Leave-One-Out Cross-Validation (LOOCV) on R^2, RMSE, and MAE in order to analyze actual effectiveness of each model in practical application. Using the same data to train and test a model leads to an overestimate of a models fit and creates an inaccurate idea of how effective a model is. Cross validation allows us to split the data into two parts, one to test the model, and a separate part to train the model. LOOCV splits the data into individual points instead of just into two subgroups. This allows for a more holistic test as every other data point is used to determine the one being tested. When you average the accuracy at each point you get a better idea of the predictive accuracy of the model on new data. R^2, RMSE, and MAE must all be valued highly, and as a result we cannot fairly assign a weight to the three of them.

```
models <- list(
  one.way = one.way,
  one.way.backward = one.way.backward,
  one.way.both = one.way.both,
  two.way = two.way,
  two.way.backward = two.way.backward)

specs <- trainControl(method = "LOOCV")

metrics <- data.frame(
  Model = names(models),
  R.sq = sapply(models, function(model) summary(model)$r.squared),
  R.adj.sq = sapply(models, function(model) summary(model)$adj.r.squared),
  LL = sapply(models, logLik),
  AIC = sapply(models, AIC),
  BIC = sapply(models, BIC),
  RMSE = sapply(models, function(model) sqrt(mean(residuals(model)^2))),
  MAE = sapply(models, function(model) mean(abs(residuals(model)))),
  Cross.R.sq = sapply(models, function(model) train(
    formula(model),
    data = life.df,
    method = "lm",
    trControl = specs,
    na.action = na.omit)$results$Rsquared),
  Cross.RMSE = sapply(models, function(model) train(
    formula(model),
    data = life.df,
    method = "lm",
trControl = specs,na.action = na.omit)$results$RMSE),
Cross.MAE = sapply(models, function(model) train(
  formula(model),
  data = life.df,
  method = "lm",
  trControl = specs,
  na.action = na.omit)$results$MAE),
Parameters = sapply(models, function(model) length(coef(model))))
```

The code above calculates the leave-one-out cross validated value for each metric we had previously acquired for our ANOVA models. The chunks of code below do the same thing for our RIDGE and LASSO regression models. These required manual calculation and were much more tedious and repetitive.

```
owr.cv_fit <- cv.glmnet(one.x, one.y, alpha = 0)
owr.predicted_values <- predict(one.way.ridge,
                                s = owr.cv_fit$lambda.min, newx = one.x)
owr.R2 <- cor(one.y, owr.predicted_values)^2
owr.n <- length(one.y)
owr.p <- ncol(one.x)
owr.R2_adj <- 1 - ((1 - owr.R2) * (owr.n - 1) / (owr.n - owr.p - 1))
owr.k <- sum(coef(one.way.ridge, s = owr.cv_fit$lambda.min) != 0)
owr.model_gaussian <- lm(one.y ~ one.x)
owr.ll <- logLik(owr.model_gaussian)
owr.AIC <- -2 * owr.ll + 2 * owr.k
owr.BIC <- -2 * owr.ll + log(owr.n) * owr.k
owr.RMSE <- RMSE(owr.predicted_values, one.y)
owr.MAE <- MAE(owr.predicted_values, one.y)
owr.predictions <- rep(NA, owr.n)
owr.actuals <- one.y
```

```
for (i in 1:owr.n) {
train_index <- setdiff(1:owr.n, i)
  owr.model_loocv <- glmnet(one.x[train_index, ], one.y[train_index],
                            alpha = 0)
  owr.predictions[i] <- predict(owr.model_loocv, s = owr.cv_fit$lambda.min,newx = one.x[i, , drop = FALSE])
}

owr.R2_loocv <- cor(owr.actuals, owr.predictions)^2
owr.RMSE_loocv <- sqrt(mean((owr.actuals - owr.predictions)^2))
owr.MAE_loocv <- mean(abs(owr.actuals - owr.predictions))

one.way.ridge.values <- list(
  Model = "one.way.ridge",
  R.sq = owr.R2,
  R.adj.sq = owr.R2_adj,
  LL = owr.ll,
  AIC = owr.AIC,
  BIC = owr.BIC,
  RMSE = owr.RMSE,
  MAE = owr.MAE,
  Cross.R.sq = owr.R2_loocv,
  Cross.RMSE = owr.RMSE_loocv,
  Cross.MAE = owr.MAE_loocv,
  Parameters = owr.k
)
```

```
twr.cv_fit <- cv.glmnet(one.x, one.y, alpha = 0)
twr.predicted_values <- predict(one.way.ridge,
                                s = twr.cv_fit$lambda.min, newx = one.x)
twr.R2 <- cor(one.y, twr.predicted_values)^2
twr.n <- length(one.y)
twr.p <- ncol(one.x)
twr.R2_adj <- 1 - ((1 - twr.R2) * (twr.n - 1) / (twr.n - twr.p - 1))
twr.k <- sum(coef(one.way.ridge, s = twr.cv_fit$lambda.min) != 0)
twr.model_gaussian <- lm(one.y ~ one.x)
twr.ll <- logLik(twr.model_gaussian)
twr.AIC <- -2 * twr.ll + 2 * twr.k
twr.BIC <- -2 * twr.ll + log(twr.n) * twr.k
twr.RMSE <- RMSE(twr.predicted_values, one.y)
twr.MAE <- MAE(twr.predicted_values, one.y)
twr.predictions <- rep(NA, twr.n)
twr.actuals <- one.y

for (i in 1:twr.n) {
train_index <- setdiff(1:twr.n, i)
  twr.model_loocv <- glmnet(one.x[train_index, ], one.y[train_index],
                            alpha = 0)
  twr.predictions[i] <- predict(twr.model_loocv, s = twr.cv_fit$lambda.min,newx = one.x[i, , drop = FALSE])
}

twr.R2_loocv <- cor(twr.actuals, twr.predictions)^2
twr.RMSE_loocv <- sqrt(mean((twr.actuals - twr.predictions)^2))
twr.MAE_loocv <- mean(abs(twr.actuals - twr.predictions))

two.way.ridge.values <- list(
  Model = "two.way.ridge",
  R.sq = twr.R2,
  R.adj.sq = twr.R2_adj,
  LL = twr.ll,
  AIC = twr.AIC,
  BIC = twr.BIC,
  RMSE = twr.RMSE,
  MAE = twr.MAE,
  Cross.R.sq = twr.R2_loocv,
  Cross.RMSE = twr.RMSE_loocv,
  Cross.MAE = twr.MAE_loocv,
  Parameters = twr.k
)
```

```
owl.cv_fit <- cv.glmnet(one.x, one.y, alpha = 0)
owl.predicted_values <- predict(one.way.ridge,
                                s = owl.cv_fit$lambda.min, newx = one.x)
owl.R2 <- cor(one.y, owl.predicted_values)^2
owl.n <- length(one.y)
owl.p <- ncol(one.x)
owl.R2_adj <- 1 - ((1 - owl.R2) * (owl.n - 1) / (owl.n - owl.p - 1))
owl.k <- sum(coef(one.way.ridge, s = owl.cv_fit$lambda.min) != 0)
owl.model_gaussian <- lm(one.y ~ one.x)
owl.ll <- logLik(owl.model_gaussian)
owl.AIC <- -2 * owl.ll + 2 * owl.k
owl.BIC <- -2 * owl.ll + log(owl.n) * owl.k
owl.RMSE <- RMSE(owl.predicted_values, one.y)
owl.MAE <- MAE(owl.predicted_values, one.y)
owl.predictions <- rep(NA, owl.n)
owl.actuals <- one.y

for (i in 1:owl.n) {
train_index <- setdiff(1:owl.n, i)
  owl.model_loocv <- glmnet(one.x[train_index, ], one.y[train_index],
                            alpha = 0)
  owl.predictions[i] <- predict(owl.model_loocv, s = owl.cv_fit$lambda.min,newx = one.x[i, , drop = FALSE])
}

owl.R2_loocv <- cor(owl.actuals, owl.predictions)^2
owl.RMSE_loocv <- sqrt(mean((owl.actuals - owl.predictions)^2))
owl.MAE_loocv <- mean(abs(owl.actuals - owl.predictions))

one.way.lasso.values <- list(
  Model = "one.way.lasso",
  R.sq = owl.R2,
  R.adj.sq = owl.R2_adj,
  LL = owl.ll,
  AIC = owl.AIC,
  BIC = owl.BIC,
  RMSE = owl.RMSE,
  MAE = owl.MAE,
  Cross.R.sq = owl.R2_loocv,
  Cross.RMSE = owl.RMSE_loocv,
  Cross.MAE = owl.MAE_loocv,
  Parameters = owl.k
)
```

,

```r
twl.cv_fit <- cv.glmnet(one.x, one.y, alpha = 0)
twl.predicted_values <- predict(one.way.ridge,
                                s = twl.cv_fit$lambda.min, newx = one.x)
twl.R2 <- cor(one.y, twl.predicted_values)^2
twl.n <- length(one.y)
twl.p <- ncol(one.x)
twl.R2_adj <- 1 - ((1 - twl.R2) * (twl.n - 1) / (twl.n - twl.p - 1))
twl.k <- sum(coef(one.way.ridge, s = twl.cv_fit$lambda.min) != 0)
twl.model_gaussian <- lm(one.y ~ one.x)
twl.ll <- logLik(twl.model_gaussian)
twl.AIC <- -2 * twl.ll + 2 * twl.k
twl.BIC <- -2 * twl.ll + log(twl.n) * twl.k
twl.RMSE <- RMSE(twl.predicted_values, one.y)
twl.MAE <- MAE(twl.predicted_values, one.y)
twl.predictions <- rep(NA, twl.n)
twl.actuals <- one.y

for (i in 1:twl.n) {
train_index <- setdiff(1:twl.n, i)
  twl.model_loocv <- glmnet(one.x[train_index, ], one.y[train_index],
                            alpha = 0)
  twl.predictions[i] <- predict(twl.model_loocv, s = twl.cv_fit$lambda.min,newx = one.x[i, , drop = FALSE])
}
twl.R2_loocv <- cor(twl.actuals, twl.predictions)^2
twl.RMSE_loocv <- sqrt(mean((twl.actuals - twl.predictions)^2))
twl.MAE_loocv <- mean(abs(twl.actuals - twl.predictions))

two.way.lasso.values <- list(
  Model = "two.way.lasso",
  R.sq = twl.R2,
  R.adj.sq = twl.R2_adj,
  LL = twl.ll,
  AIC = twl.AIC,
  BIC = twl.BIC,
  RMSE = twl.RMSE,
  MAE = twl.MAE,
  Cross.R.sq = twl.R2_loocv,
  Cross.RMSE = twl.RMSE_loocv,
  Cross.MAE = twl.MAE_loocv,
  Parameters = twl.k
)
```

```r
one.way.ridge.df <- as.data.frame(t(one.way.ridge.values))
one.way.lasso.df <- as.data.frame(t(one.way.lasso.values))
two.way.ridge.df <- as.data.frame(t(two.way.ridge.values))
two.way.lasso.df <- as.data.frame(t(two.way.lasso.values))
metrics <- rbind(metrics, one.way.ridge.df, one.way.lasso.df,
                 two.way.ridge.df, two.way.lasso.df)
for (variable in names(metrics)) { metrics[[variable]] <- unlist(metrics[[variable]])
}
```

```r
cv.table <- xtable(metrics, digits = 4, include.rownames = FALSE)
split_column <- ncol(cv.table) %/% 2

table1 <- cv.table[, 1:split_column]
table2 <- cv.table[, (split_column + 1):ncol(cv.table)]
```

```
xtable1 <- xtable(table1, digits = 4, include.rownames = FALSE)
xtable2 <- xtable(table2, digits = 4, include.rownames = FALSE)
```

Model	R.sq	R.adj.sq	LL	AIC	BIC
one.way	0.7749	0.7630	-320.8177	657.6353	679.9352
one.way.backward	0.7748	0.7670	-320.8481	653.6962	670.4211
one.way.both	0.7636	0.7596	-323.7543	655.5086	666.6586
two.way	0.7749	0.7650	-320.8294	655.6587	675.1712
two.way.backward	0.7748	0.7670	-320.8481	653.6962	670.4211
one.way.ridge	0.7739	0.7619	-320.8177	655.6353	675.1477
one.way.lasso	0.7739	0.7619	-320.8177	655.6353	675.1477
two.way.ridge	0.7739	0.7619	-320.8177	655.6353	675.1477
two.way.lasso	0.7739	0.7619	-320.8177	655.6353	675.1477

Model	RMSE	MAE	Cross.R.sq	Cross.RMSE	Cross.MAE	Parameters
one.way	3.5062	2.6997	0.7450	3.7370	2.8775	7
one.way.backward	3.5071	2.7051	0.7527	3.6782	2.8332	5
one.way.both	3.5931	2.7774	0.7505	3.6923	2.8502	3
two.way	3.5066	2.7038	0.7493	3.7040	2.8555	6
two.way.backward	3.5071	2.7051	0.7527	3.6782	2.8332	5
one.way.ridge	3.5275	2.7100	0.7458	3.7286	2.8690	7
two.way.ridge	3.5275	2.7100	0.7458	3.7286	2.8690	7
one.way.lasso	3.5275	2.7100	0.7458	3.7286	2.8690	7
two.way.lasso	3.5275	2.7100	0.7458	3.7286	2.8690	7

Now, we have a multitude of models to choose from. Our process now will be to organize them in a manner in which we can rank and sort them in order for them to be compared.

4.5 Model Ranking

```
metrics_ranked <- metrics_ordered %>%
  mutate(
    Rank_R.sq = rank(-R.sq),
    Rank_R.adj.sq = rank(-R.adj.sq),
    Rank_LL = rank(-LL),
    Rank_AIC = rank(-AIC),
    Rank_BIC = rank(-BIC),
    Rank_RMSE = rank(-RMSE),
    Rank_MAE = rank(-MAE),
 Rank_Cross.R.sq = rank(-Cross.R.sq),
    Rank_Cross.RMSE = rank(-Cross.RMSE),
    Rank_Cross.MAE = rank(-Cross.MAE)
)
metrics_ranked <- metrics_ranked %>%
  mutate(
    Rank_avg = rowMeans(cbind(Rank_R.sq, Rank_R.adj.sq, Rank_LL,
                              Rank_AIC, Rank_BIC, Rank_RMSE, Rank_MAE,
                              Rank_Cross.R.sq, Rank_Cross.RMSE,
                              Rank_Cross.MAE))
)

columns_to_print <- c("Model", "Rank_R.sq", "Rank_R.adj.sq", "Rank_LL",
                      "Rank_AIC","Rank_BIC", "Rank_RMSE", "Rank_MAE",
                      "Rank_Cross.R.sq", "Rank_Cross.RMSE", "Rank_Cross.MAE",
                      "Rank_avg")
rank.table<- xtable(as.data.frame(metrics_ranked))
```

```r
num_columns <- ncol(rank.table)
columns_per_split <- num_columns %/% 6

remaining_columns <- num_columns %% 6
start_index <- 1

#for (i in 1:6) {
#  end_index <- start_index + columns_per_split - 1
#    if (i <= remaining_columns) {
#    end_index <- end_index + 1
#  }
#
#  table_i <- rank.table[, start_index:end_index]
#
#  xtable_i <- xtable(table_i, digits = 4, include.rownames = FALSE)
#
#  print(xtable_i)
#
#  start_index <- end_index + 1
#}
```

Model	R.sq	R.adj.sq	LL
one.way	0.7928	0.7798	-315.8585
one.way.backward	0.7864	0.7809	-317.6659
one.way.both	0.7864	0.7809	-317.6659
two.way	0.7958	0.7811	-314.9799
two.way.backward	0.7909	0.7817	-316.3988
one.way.ridge	0.7919	0.7789	-315.8585
one.way.lasso	0.7919	0.7789	-315.8585
two.way.ridge	0.7919	0.7789	-315.8585
two.way.lasso	0.7919	0.7789	-315.8585

Model	AIC	BIC	RMSE	MAE
one.way	-649.7170	-674.8044	-3.3643	-2.5751
one.way.backward	-645.3318	-659.2693	-3.4153	-2.6168
one.way.both	-645.3318	-659.2693	-3.4153	-2.6168
two.way	-649.9597	-677.8347	-3.3397	-2.5436
two.way.backward	-646.7976	-666.3100	-3.3795	-2.5680
one.way.ridge	-647.7170	-670.0169	-3.3791	-2.5812
one.way.lasso	-647.7170	-670.0169	-3.3791	-2.5812
two.way.ridge	-647.7170	-670.0169	-3.3791	-2.5812
two.way.lasso	-647.7170	-670.0169	-3.3791	-2.5812

Model	Cross.R.sq	Cross.RMSE	Cross.MAE	Parameters
one.way	0.7625	-3.6055	-2.7605	7
one.way.backward	0.7713	-3.5356	-2.7042	5
one.way.both	0.7713	-3.5356	-2.7042	3
two.way	0.7631	-3.6015	-2.7458	6
two.way.backward	0.7682	-3.5602	-2.7031	5
one.way.ridge	0.7647	-3.5860	-2.7429	7
one.way.lasso	0.7647	-3.5860	-2.7429	7
two.way.ridge	0.7647	-3.5860	-2.7429	7
two.way.lasso	0.7647	-3.5860	-2.7429	7

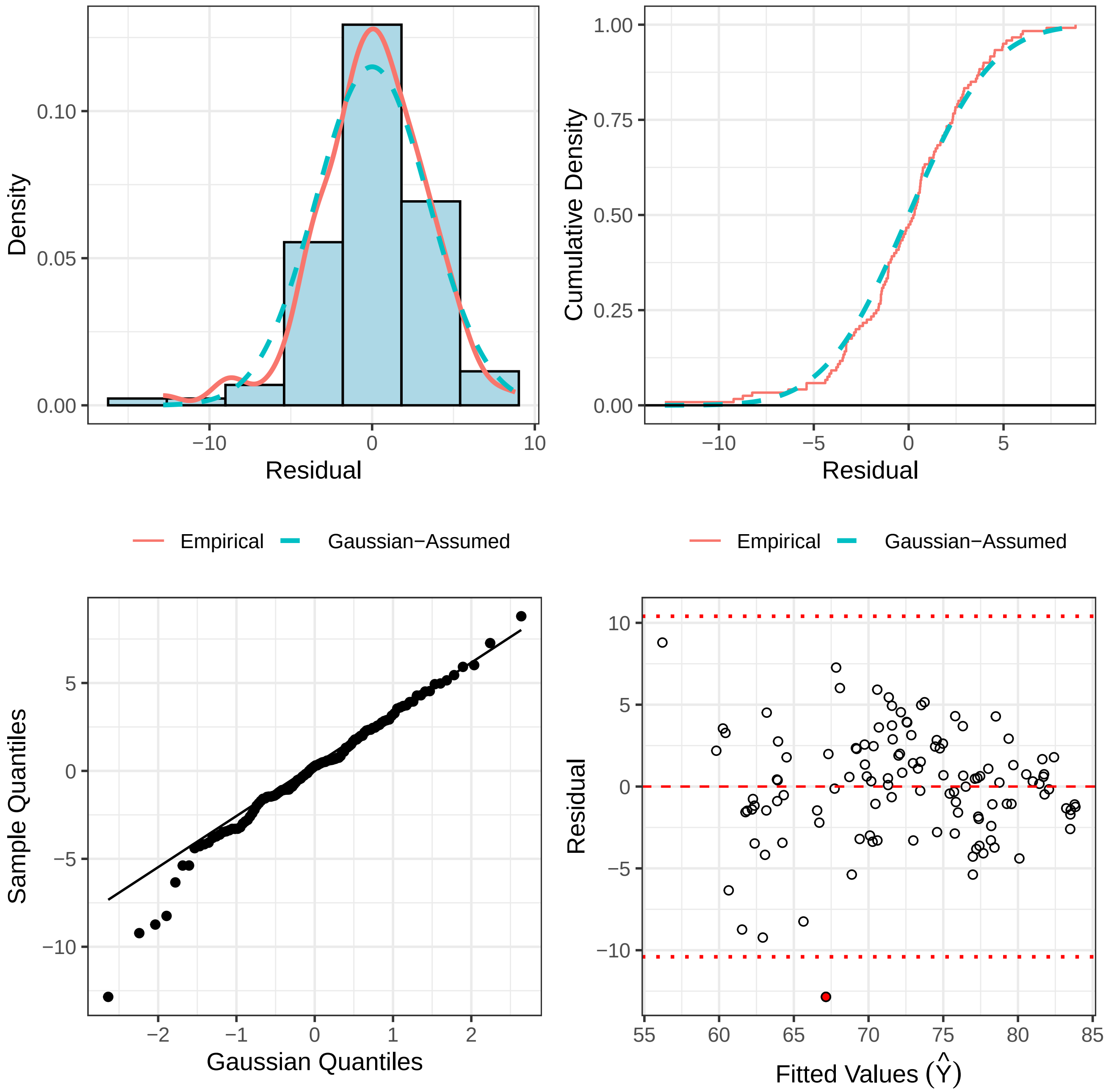
Model	Rank_R.sq	Rank_R.adj.sq	Rank_LL	Rank_AIC
one.way	2.0000	5.0000	4.0000	8.0000
one.way.backward	8.5000	3.5000	8.5000	1.5000
one.way.both	8.5000	3.5000	8.5000	1.5000
two.way	1.0000	2.0000	1.0000	9.0000
two.way.backward	7.0000	1.0000	7.0000	3.0000
one.way.ridge	4.5000	7.5000	4.0000	5.5000
one.way.lasso	4.5000	7.5000	4.0000	5.5000
two.way.ridge	4.5000	7.5000	4.0000	5.5000
two.way.lasso	4.5000	7.5000	4.0000	5.5000

Model	Rank_BIC	Rank_RMSE	Rank_MAE	Rank_Cross.R.sq
one.way	8.0000	2.0000	3.0000	9.0000
one.way.backward	1.5000	8.5000	8.5000	1.5000
one.way.both	1.5000	8.5000	8.5000	1.5000
two.way	9.0000	1.0000	1.0000	8.0000
two.way.backward	3.0000	7.0000	2.0000	3.0000
one.way.ridge	5.5000	4.5000	5.5000	5.5000
one.way.lasso	5.5000	4.5000	5.5000	5.5000
two.way.ridge	5.5000	4.5000	5.5000	5.5000
two.way.lasso	5.5000	4.5000	5.5000	5.5000

```
plotResiduals(two.way.backward)
```

Model	Rank_Cross.RMSE	Rank_Cross.MAE	Rank_avg
one.way	9.0000	9.0000	5.9000
one.way.backward	1.5000	2.5000	4.6000
one.way.both	1.5000	2.5000	4.6000
two.way	8.0000	8.0000	4.8000
two.way.backward	3.0000	1.0000	3.7000
one.way.ridge	5.5000	5.5000	5.3500
one.way.lasso	5.5000	5.5000	5.3500
two.way.ridge	5.5000	5.5000	5.3500
two.way.lasso	5.5000	5.5000	5.3500

Table 9: Table detailing final rankings of models



```
summary(two.way.backward)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ Minimum_wage + log1p(1/Birth_Rate) +
##     log1p(LFPR) + log(Physicians_per_thousand) + log1p(1/Birth_Rate):log1p(LFPR),
##     data = life.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8487  -1.6176   0.2763   2.3052   8.7947
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       76.2789     6.4532  11.820  < 2e-16 ***
## Minimum_wage                       0.5551     0.1253   4.429 2.19e-05 ***
## log1p(1/Birth_Rate)             -128.1966   123.8898  -1.035    0.303
## log1p(LFPR)                      -19.5053    13.9020  -1.403    0.163
## log(Physicians_per_thousand)       2.5497     0.3542   7.198 6.98e-11 ***
## log1p(1/Birth_Rate):log1p(LFPR)  421.7541   270.6108   1.559    0.122
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.467 on 114 degrees of freedom
## Multiple R-squared:  0.7909, Adjusted R-squared:  0.7817
## F-statistic: 86.23 on 5 and 114 DF,  p-value: < 2.2e-16
```

The residuals seem to have constant variance. When looking at the residual plot, there is a clear random dispersal of points across the middle line at 0. Only one point still lies outside the banded region, which is acceptable. Finally, there is no break in the pattern we see in the plot.

The residual plots also are approximately normal. The residual density is bell curved and is close to the Gaussian-Assumed curve. The points in the q-q plot follow a straight line except for some skew that still remains on the left end due to the smaller data set after cleaning.

(Final $R^2 = 0.7817$ p = 0)

5 Conclusion

The top-performing model, determined through a comprehensive ranking system evaluating various metrics including R^2, LL, AIC, BIC, RMSE, and MAE was the two-way backward model, which based of the criteria stated, had the best average rank(Final $R^2 = 0.7817$ p = 0). Within our model, the most significant predictor variables were minimum wage and the number of physicians per one thousand people in the population. Conversely, interaction between birth rate and labor force participation rate, birth rate, and labor force participation rate were found to be less indicative. Each of our significant variables were consistent with expectations derived from exploratory data analyses using ANOVA, however, some variables that were significant in the two-way ANOVA were not significant in our model. Despite their relatively weak predictive power, these variables underscore the importance of socioeconomic factors on international health outcomes which are a major factor in life expectancy.

Reflecting on the project, it becomes apparent that a better strategy for removing NA values would have been beneficial. In the data cleaning process we lost roughly 35 percent of our data set. This is due to some countries failing to report certain socioeconomic statistics due to government overreach. This introduced some bias to our data as many of these countries that do not report data are classified as developing countries.

References

Brazzale, A. (2024). *Functions and datasets for bootstrapping from the book 'Bootstrap Methods and Their Application' by A. C. Davison and D. V. Hinkley.* R Package.

CDC (2023). *Life Expectancy.*

Croissant, Y. and Graves, S. (2022). *Data Sets for Econometrics.* R Package.

Dahl, D. B., Scott, D., Roosen, C., Magnusson, A., and Swinton, J. (2019). *Export Tables to LaTeX or HTML.* R Package.

Elgiriyewithana, N. (2023). Global country information dataset 2023. *Kaggle.*

Fenwick, V., Jacobson, H., and Roberts, J. (2023). *Evaluation of Systolic Blood Pressure for Pregnant Women.*

Friedman, J., Hastie, T., Tibshirani, R., Balasubramanian, N., Tay, K., Simon, N., Qian, J., and Yang, J. (2023). *Lasso and Elastic-Net Regularized Generalized Linear Models.* R Package.

Harrell, Jr., F. E. (2023). *rms: Regression Modeling Strategies.* R package.

Herve, M. (2023). *Testing and Plotting Procedures for Biostatistics.* R Package.

Hothorn, T., Zeileis, A., Farebrother, R., Cummins, C., Millo, G., and Mitchell, D. (2022). *Testing Linear Regression Models.* R Package.

Kuhn, M. and Max, K. (2008). *Building predictive models in R using the caret package.* Journal of Statistical Software.

McLeod, A., Xu, C., and Lai, Y. (2020). *Best Subset GLM and Regression Utilities.* R Package.

Ogle, D. H., Doll, J. C., Wheeler, A. P., and Dinno, A. (2023). *FSA: Simple Fisheries Stock Assessment Methods.* R Package.

Pedersen, L. (2023). The composer of plots.

Ripley, B., Venables, B., Bates, D., Hornik, K., Gebhardt, A., and Firth, D. (2024). *Functions and datasets to support Venables and Ripley, "Modern Applied Statistics with S".* R Package.

Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A., and Crowley, J. (2021). *GGally: Extension to ggplot2.* R package.

Wickham, H. (2016). *Elegant Graphics for Data Analysis.* Springer-Verlag, New York.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., Francois, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Muller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). *Welcome to the tidyverse.* Journal of Open Source Software.