New Final Project 2

Harrison Plate

4/28/2022

```
#install.packages("openintro")
library(readr)
cia_factbook = read_csv("cia_factbook.csv")
```

```
## Rows: 259 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): country
## dbl (10): area, birth_rate, death_rate, infant_mortality_rate, internet_user...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#View(airline_delay)
```

```
head(cia_factbook)
```

```
## # A tibble: 6 x 11
##   country           area birth_rate death_rate infant_mortality_~ internet_users
##   <chr>            <dbl>      <dbl>      <dbl>              <dbl>          <dbl>
## 1 Russia        17098242       11.9      13.8                7.08       40853000
## 2 Canada         9984670       10.3       8.31               4.71       26960000
## 3 United States  9826675       13.4       8.15               6.17      245000000
## 4 China          9596960       12.2       7.44              14.8       389000000
## 5 Brazil         8514877       14.7       6.54              19.2        75982000
## 6 Australia      7741220       12.2       7.07               4.43       15810000
## # ... with 5 more variables: life_exp_at_birth <dbl>,
## #   maternal_mortality_rate <dbl>, net_migration_rate <dbl>, population <dbl>,
## #   population_growth_rate <dbl>
```

```
#require(MASS)
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
library(car)
```

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

```
library(magrittr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#library(HMisc)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.1.3
```

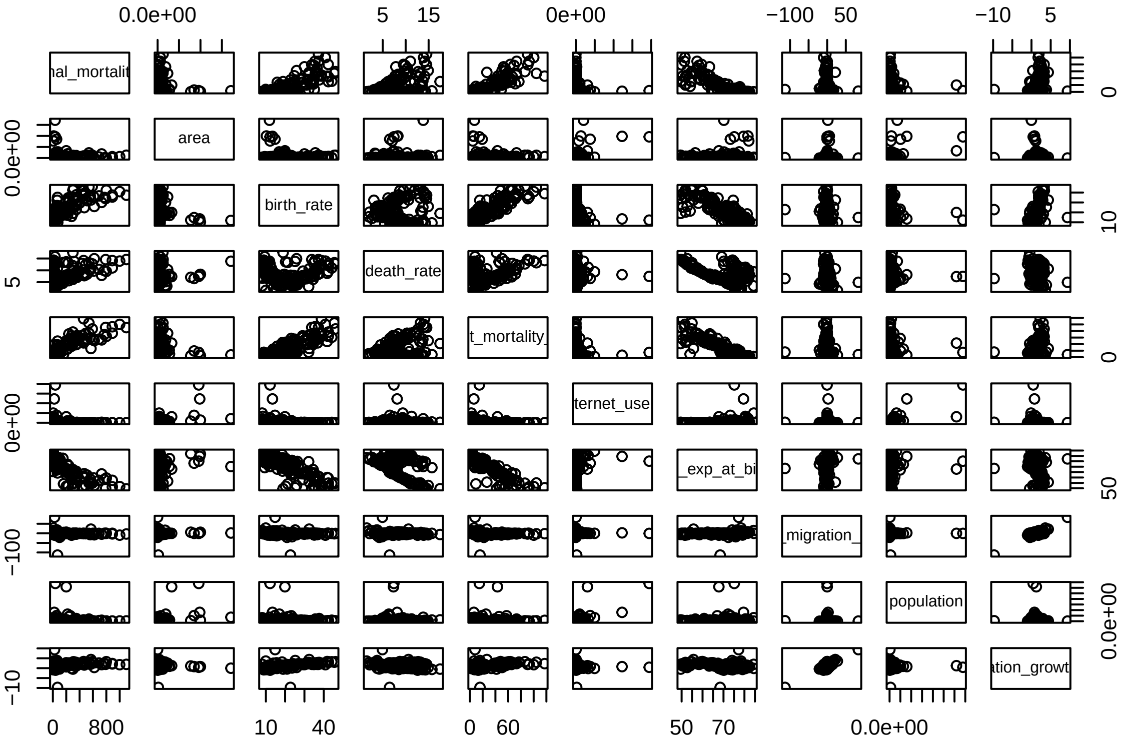
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

```
df1 = na.omit(cia_factbook)

#data cleaning
#omit all of the na's. Not sure what else to do with them
```

```
pairs(df1[c(8, 2, 3, 4, 5, 6, 7, 9, 10, 11)])
```



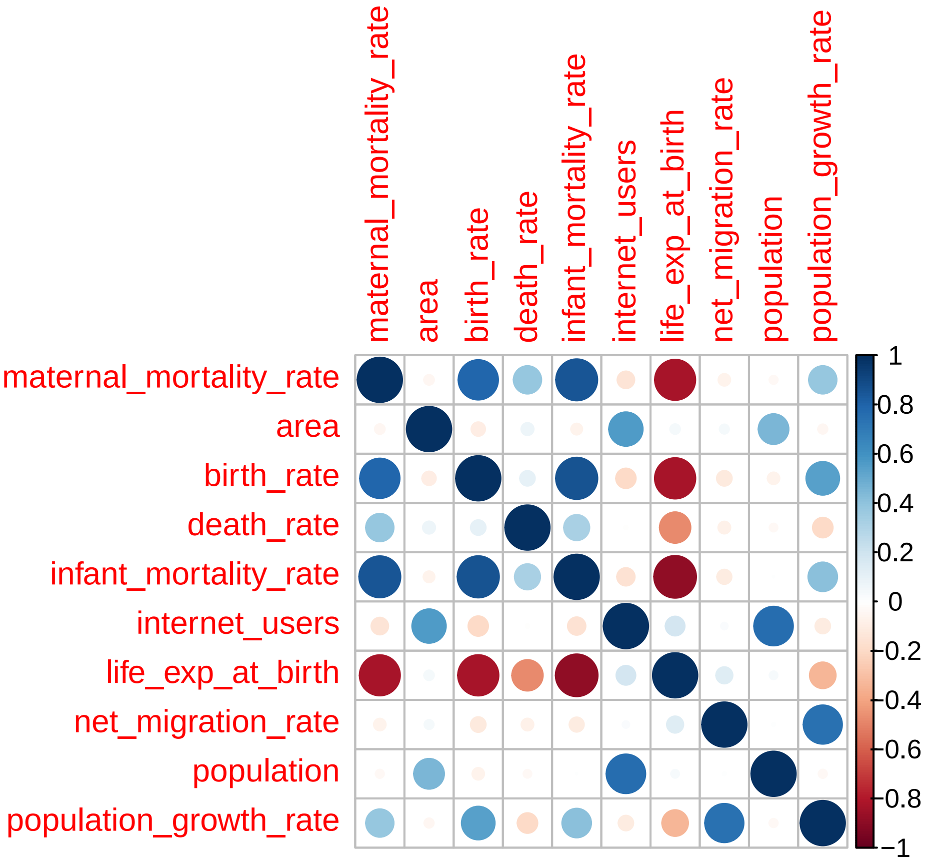
```
#paired scatterplot with our response at the top
```

From looking at our matrix scatter we can see that many predictive variables have a linear correlation with Maternity Mortality Rate.

```
df_int1 = data.frame(df1)
df_int2 = df_int1 %>% select(-country)
df_final = df_int2 %>% select(maternal_mortality_rate, everything())

#we made some intermediate data frames so that we could remove country and add our response, maternal m
```

```
corrplot(cor(df_final))
```



```
#correlation matrix
```

By inspecting our at our correlation matrix, we come across many predictors being correlated to Maternity Mortality rate. As well some predictors being correlated within themselves such as birthrate and infant mortality rate, and birthrate and life expectancy rate.

```
df1$continent = df1$country


df2 <- df1 %>% mutate(continent = recode(continent,  "Russia" = 3, "Canada" = 1, "United States" = 1, "

#we made a new dataframe to group the countries into continents, but we aren't using it
```

```
df2$fcontinent = as.factor(df2$continent)

#as factored continent, but we're not using it
```

β_1 = area

β_2 = internet_users

β_3 = death_rate

β_4 = infant_mortality_rate

β_5 = life_exp_at_birth

β_6 = birth_rate

β_7 = net_migration_rate

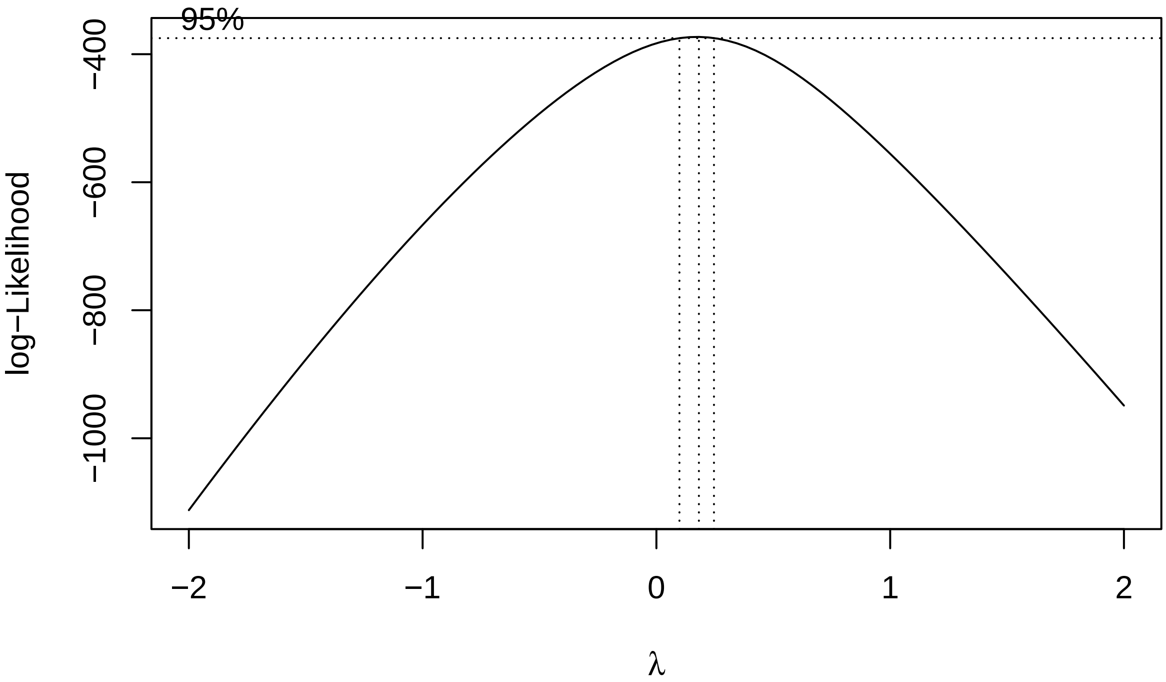
β_8 = population

β_9 = population_growth_rate

$H_0 : \beta_j = 0$

$H_A : \beta_j \neq 0$ for at least 1

```
lmfull = lm(maternal_mortality_rate ~ area + internet_users + death_rate + infant_mortality_rate + life
boxcox(lmfull)
```



```
#original model with all the predictors present
```

```
summary(powerTransform(lmfull))
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.1738        0.17       0.0989       0.2487
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                              LRT df       pval
## LR test, lambda = (0) 19.85563  1 8.3516e-06
##
## Likelihood ratio test that no transformation is needed
##                              LRT df       pval
## LR test, lambda = (1) 365.0084  1 < 2.22e-16
```

```
#power transform/boxcox
```

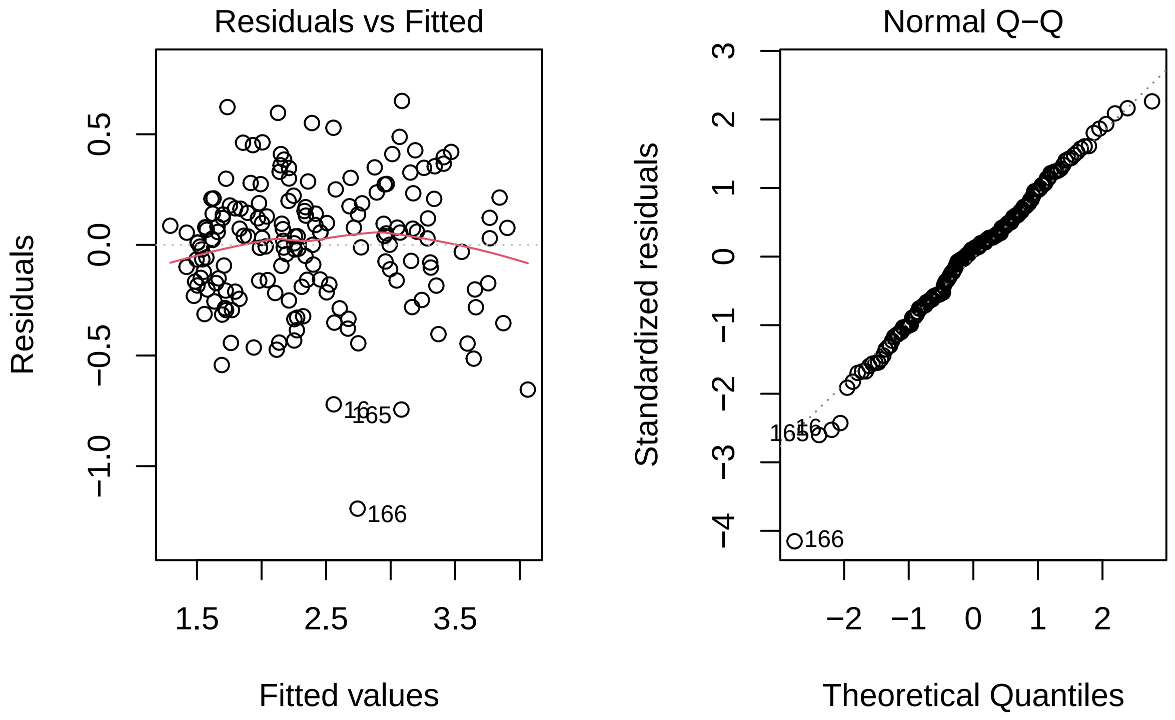
The boxcox/powertransform is telling us to do a 0.2 transformation

```
lmlogfull = lm((maternal_mortality_rate)^(1/5) ~ area + internet_users + death_rate + infant_mortality_
summary(lmlogfull)
```

```
##
## Call:
## lm(formula = (maternal_mortality_rate)^(1/5) ~ area + internet_users +
##     death_rate + infant_mortality_rate + life_exp_at_birth +
##     birth_rate + net_migration_rate + population + population_growth_rate,
##     data = df_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.19200 -0.17924  0.02994  0.17049  0.65045
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.424e+00  6.300e-01   8.609 5.26e-15 ***
## area                    8.479e-09  1.344e-08   0.631  0.52908
## internet_users         -6.503e-10  1.046e-09  -0.622  0.53497
## death_rate             -1.465e-01  7.584e-01  -0.193  0.84703
## infant_mortality_rate   8.605e-03  2.293e-03   3.754  0.00024 ***
## life_exp_at_birth      -4.534e-02  6.976e-03  -6.499 8.93e-10 ***
## birth_rate              1.175e-01  7.576e-01   0.155  0.87697
## net_migration_rate      1.031e-01  7.579e-01   0.136  0.89195
## population              2.473e-10  2.471e-10   1.001  0.31836
## population_growth_rate -1.048e+00  7.578e+00  -0.138  0.89019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2902 on 167 degrees of freedom
## Multiple R-squared:  0.8496, Adjusted R-squared:  0.8415
## F-statistic: 104.8 on 9 and 167 DF,  p-value: < 2.2e-16
```

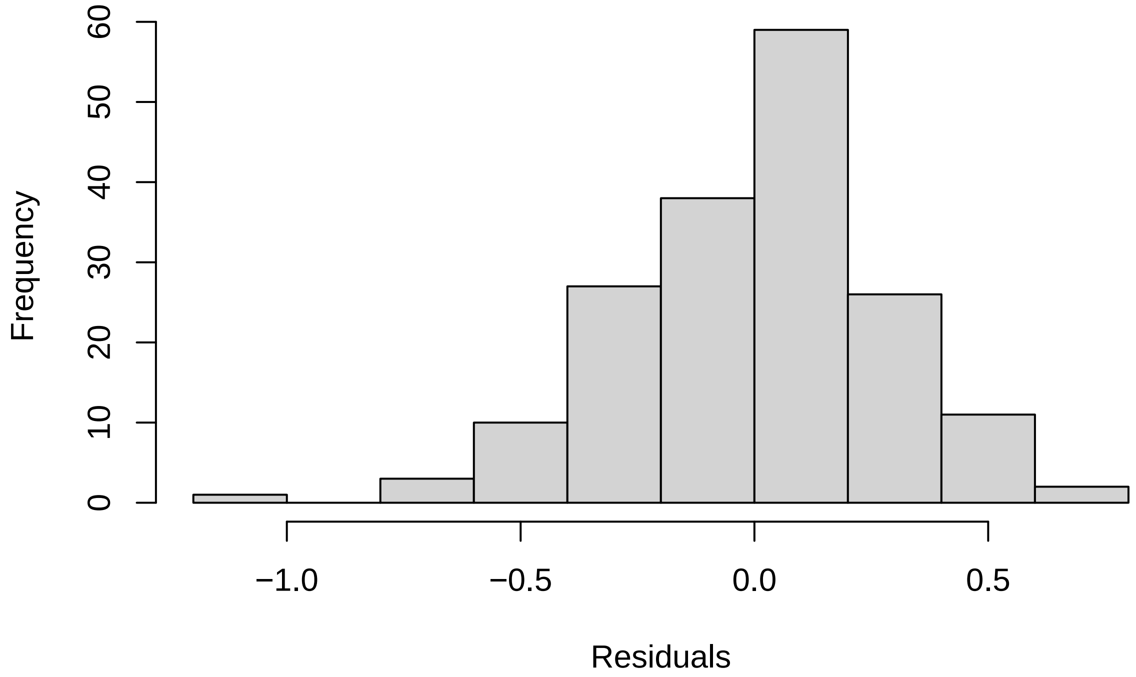
```
#our model with a 0.2 transformation
```

```
par(mfrow = c(1, 2))
plot(lmlogfull, 1:2)
```



```
#checking assumptions
```

```
hist(resid(lmlogfull), main = "", xlab = "Residuals")
```



```
#checking assumptions
```

Assumptions:

Linearity: The assumption has been satisfied because of the good looking QQ plot with the dots near the line.

Independence of Errors: The assumption has been satisfied because of the residuals vs fitted graph, where there is no correlation.

Normality of Errors: The residuals most be approximately normally distributed. This is proven by the QQ plot (can also use a histogram of the residuals), which we can see is normally distributed since the points are close to the line.

Equal Variances: This is proven by the residuals vs fitted graph. The variance of residuals are the same across all values on the x-axis. The graph shows no pattern, so the assumption has been met.

```
lmlogstep = step(lmlogfull)
```

```
## Start:  AIC=-428.31
## (maternal_mortality_rate)^(1/5) ~ area + internet_users + death_rate +
##     infant_mortality_rate + life_exp_at_birth + birth_rate +
##     net_migration_rate + population + population_growth_rate
##
##                          Df Sum of Sq    RSS     AIC
## - net_migration_rate      1    0.0016 14.062 -430.29
## - population_growth_rate  1    0.0016 14.062 -430.29
## - birth_rate              1    0.0020 14.062 -430.28
## - death_rate              1    0.0031 14.063 -430.27
## - internet_users          1    0.0325 14.093 -429.90
## - area                    1    0.0335 14.094 -429.89
## - population              1    0.0843 14.144 -429.25
## <none>                                14.060 -428.31
## - infant_mortality_rate   1    1.1862 15.246 -415.97
## - life_exp_at_birth       1    3.5563 17.616 -390.40
##
## Step:  AIC=-430.29
## (maternal_mortality_rate)^(1/5) ~ area + internet_users + death_rate +
##     infant_mortality_rate + life_exp_at_birth + birth_rate +
##     population + population_growth_rate
##
##                          Df Sum of Sq    RSS     AIC
## - internet_users          1    0.0319 14.094 -431.89
## - area                    1    0.0345 14.096 -431.85
## - population_growth_rate  1    0.0699 14.132 -431.41
## - population              1    0.0836 14.145 -431.24
## <none>                                14.062 -430.29
## - birth_rate              1    0.5150 14.577 -425.92
## - infant_mortality_rate   1    1.2020 15.264 -417.77
## - death_rate              1    1.5626 15.624 -413.64
## - life_exp_at_birth       1    3.6023 17.664 -391.92
##
## Step:  AIC=-431.89
## (maternal_mortality_rate)^(1/5) ~ area + death_rate + infant_mortality_rate +
##     life_exp_at_birth + birth_rate + population + population_growth_rate
##
##                          Df Sum of Sq    RSS     AIC
## - area                    1    0.0163 14.110 -433.68
## - population              1    0.0529 14.146 -433.22
## - population_growth_rate  1    0.0683 14.162 -433.03
## <none>                                14.094 -431.89
## - birth_rate              1    0.4961 14.590 -427.76
## - infant_mortality_rate   1    1.2199 15.313 -419.19
## - death_rate              1    1.6619 15.755 -414.16
## - life_exp_at_birth       1    3.8136 17.907 -391.50
##
## Step:  AIC=-433.68
## (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     life_exp_at_birth + birth_rate + population + population_growth_rate
##
##                          Df Sum of Sq    RSS     AIC
## - population_growth_rate  1    0.0655 14.175 -434.86
## - population              1    0.1057 14.216 -434.36
## <none>                                14.110 -433.68
## - birth_rate              1    0.5002 14.610 -429.52
## - infant_mortality_rate   1    1.2073 15.317 -421.15
## - death_rate              1    1.6467 15.757 -416.14
## - life_exp_at_birth       1    3.8058 17.916 -393.41
##
## Step:  AIC=-434.86
## (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     life_exp_at_birth + birth_rate + population
##
##                         Df Sum of Sq    RSS     AIC
## - population             1    0.1065 14.282 -435.54
## <none>                                14.175 -434.86
## - birth_rate             1    0.4372 14.613 -431.49
## - infant_mortality_rate  1    1.1859 15.361 -422.64
## - death_rate             1    1.5812 15.757 -418.14
## - life_exp_at_birth      1    3.8801 18.055 -394.04
##
## Step:  AIC=-435.54
## (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     life_exp_at_birth + birth_rate
##
##                         Df Sum of Sq    RSS     AIC
## <none>                                14.282 -435.54
## - birth_rate             1    0.3830 14.665 -432.85
## - infant_mortality_rate  1    1.2956 15.578 -422.17
## - death_rate             1    1.6637 15.945 -418.03
## - life_exp_at_birth      1    3.9079 18.190 -394.73
```

```
#step
```

The only 4 predictors that have a significant effect on the response are birth_rate, infant_mortality_rate, death_rate, and life_exp_at_birth.

```
summary(lmlogstep)
```

```
##
## Call:
## lm(formula = (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     life_exp_at_birth + birth_rate, data = df_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2042 -0.1742  0.0360  0.1739  0.6626
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.547843   0.614250   9.032 3.36e-16 ***
## death_rate            -0.042858   0.009575  -4.476 1.38e-05 ***
## infant_mortality_rate  0.008784   0.002224   3.950 0.000114 ***
## life_exp_at_birth     -0.046587   0.006791  -6.860 1.19e-10 ***
## birth_rate             0.011681   0.005439   2.148 0.033144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2882 on 172 degrees of freedom
## Multiple R-squared:  0.8472, Adjusted R-squared:  0.8437
## F-statistic: 238.5 on 4 and 172 DF,  p-value: < 2.2e-16
```

If the null hypothesis is true, we would expect the F value to be close to 1. The F-statistic is 238.5 with the p value of < 2.2e-16. Since the F value is not near 1 and the p value is less than the significance level of $\alpha = 0.05$, we reject H_0 and we know that the data claims there is a relationship between the response, maternal_mortality_rate, and at least one predictor in the model.

```
s1 = summary(lmlogfull)
s2 = summary(lmlogstep)
s1$adj.r.squared
```

```
## [1] 0.841506
```

```
s2$adj.r.squared
```

```
## [1] 0.8436858
```

```
#we are comparing R^2 of the model before and after step
```

We wanted to observe if there is a change in our Adjusted R^2, by comparing both Adjusted R^2 we notice an increase of the Adjust R^2 for the reduced model, by .002. 84.36858% of the variability for (maternal_mortality_rate)^(1/5) is determined by the model.

```
anova(lmlogstep, lmlogfull)
```

```
## Analysis of Variance Table
##
## Model 1: (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     life_exp_at_birth + birth_rate
## Model 2: (maternal_mortality_rate)^(1/5) ~ area + internet_users + death_rate +
##     infant_mortality_rate + life_exp_at_birth + birth_rate +
##     net_migration_rate + population + population_growth_rate
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    172 14.282
## 2    167 14.060  5    0.2218 0.5269 0.7557
```

The p-value is 0.7557, which is higher than $\alpha = 0.05$. Therefore, we fail to reject the null hypothesis and we know that β_1 (area), β_2 (internet_users), β_7 (net_migration_rate), β_8 (population), and β_9 (population_growth_rate) have no impact on our model, so we can remove them and use the reduced model.

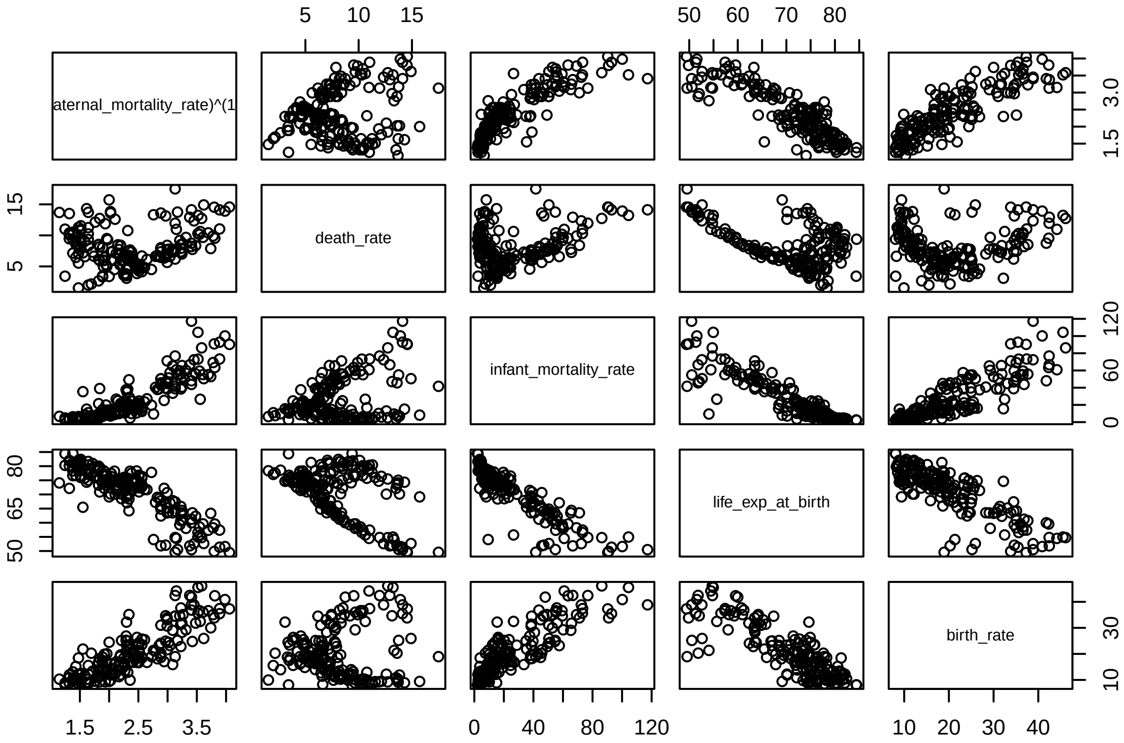
```
AIC(lmlogstep, lmlogfull)
```

```
##           df      AIC
## lmlogstep  6 68.76710
## lmlogfull 11 75.99666
```

```
#comparing AIC of the model before and after the step
```

The stepped model has a lower AIC, and a lower AIC means the better the regression model fits the data.

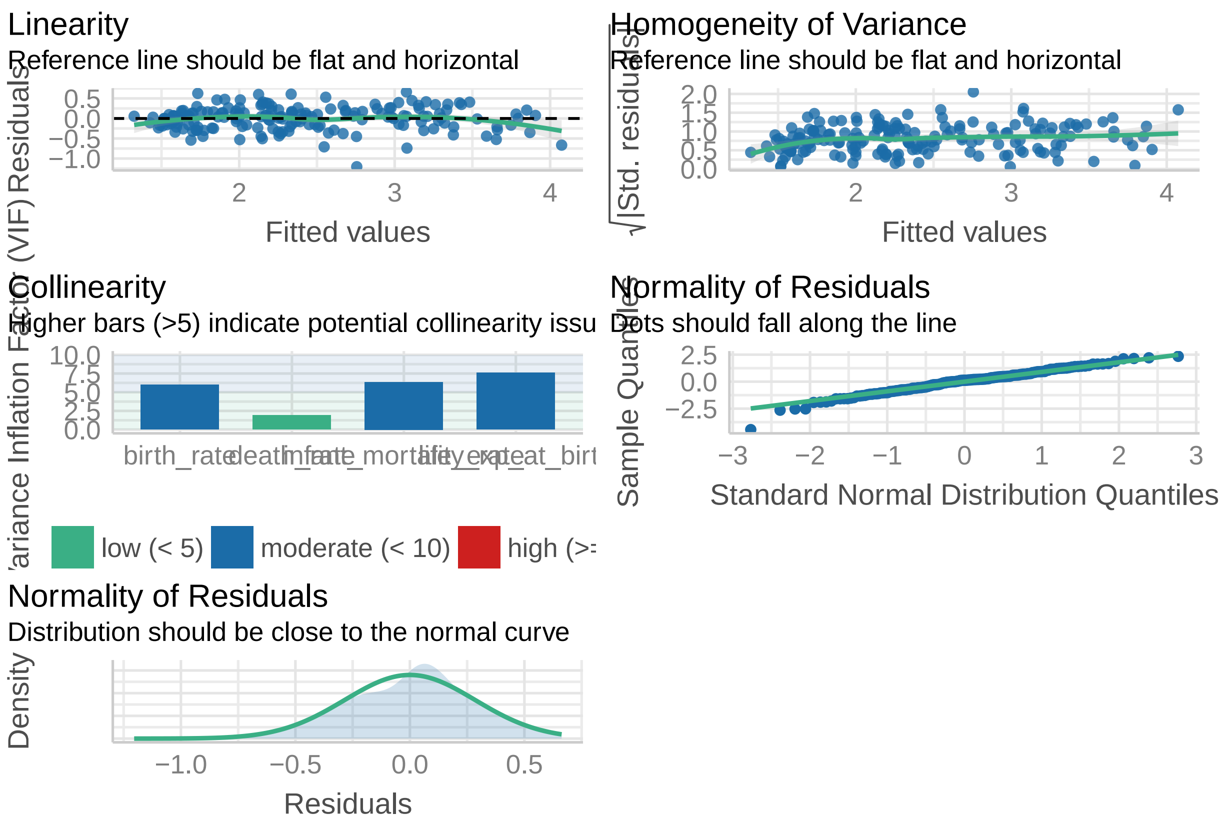
```
pairs((maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
    life_exp_at_birth + birth_rate, data = df_final)
```



```
#pairs plot after 0.2 transformation on the response
```

Things are looking generally linear here!

```
performance::check_model(lmlogstep)
```



The assumptions still look good, but we notice some collinearity issues we need to address.

```
#we need to figure out which predictor to drop in order to address the collinearity issue, so we do a r

y = df_final$maternal_mortality_rate^(1/5)
x = data.matrix(df_final[, c("death_rate", "infant_mortality_rate", "life_exp_at_birth", "birth_rate")]]

model = glmnet(x, y, alpha = 0)
summary(model)
```

```
##           Length Class     Mode
## a0        100    -none-    numeric
## beta      400    dgCMatrix S4
## df        100    -none-    numeric
## dim         2    -none-    numeric
## lambda    100    -none-    numeric
## dev.ratio 100    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        4    -none-    call
## nobs        1    -none-    numeric
```

```
#loading data in for ridge regression
```

```
cv_model = cv.glmnet(x, y, alpha = 0)
```

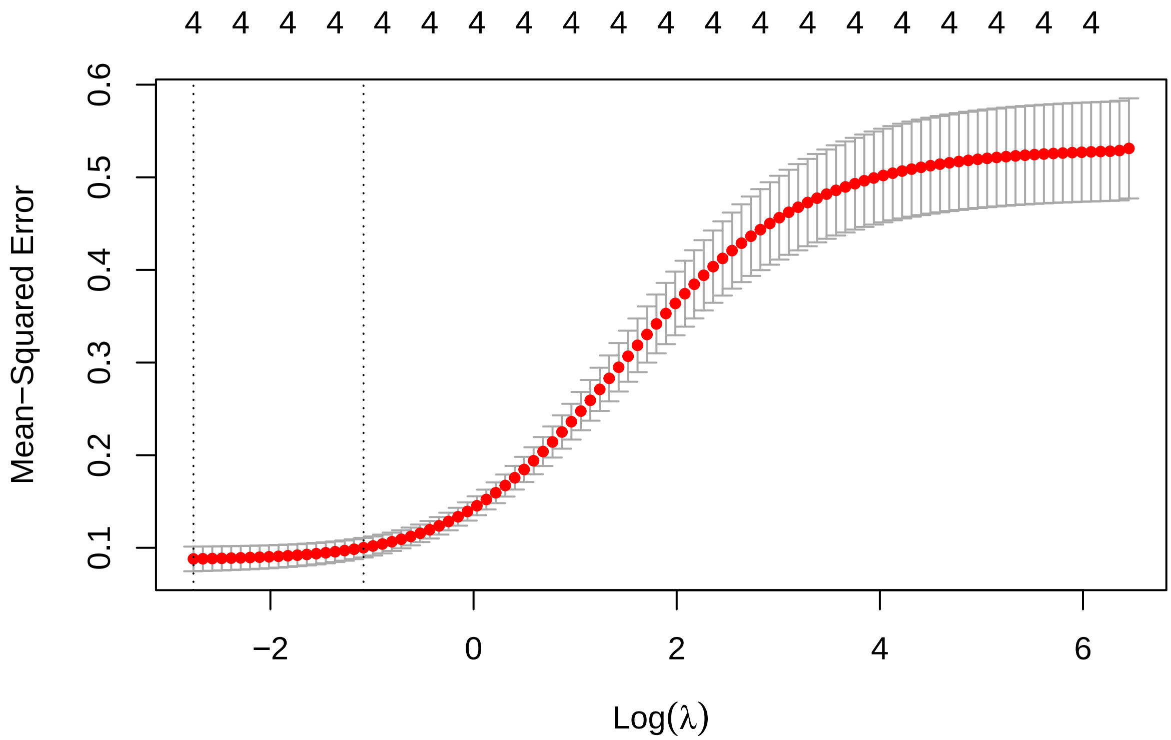
```
best_lambda = cv_model$lambda.min
best_lambda
```

```
## [1] 0.06342977
```

```
#we are trying to find a lambda value that produces the lowest MSE
```

The lambda value that minimizes the test MSE is 0.06342977. The lowest MSE produces the best model.

```
plot(cv_model)
```



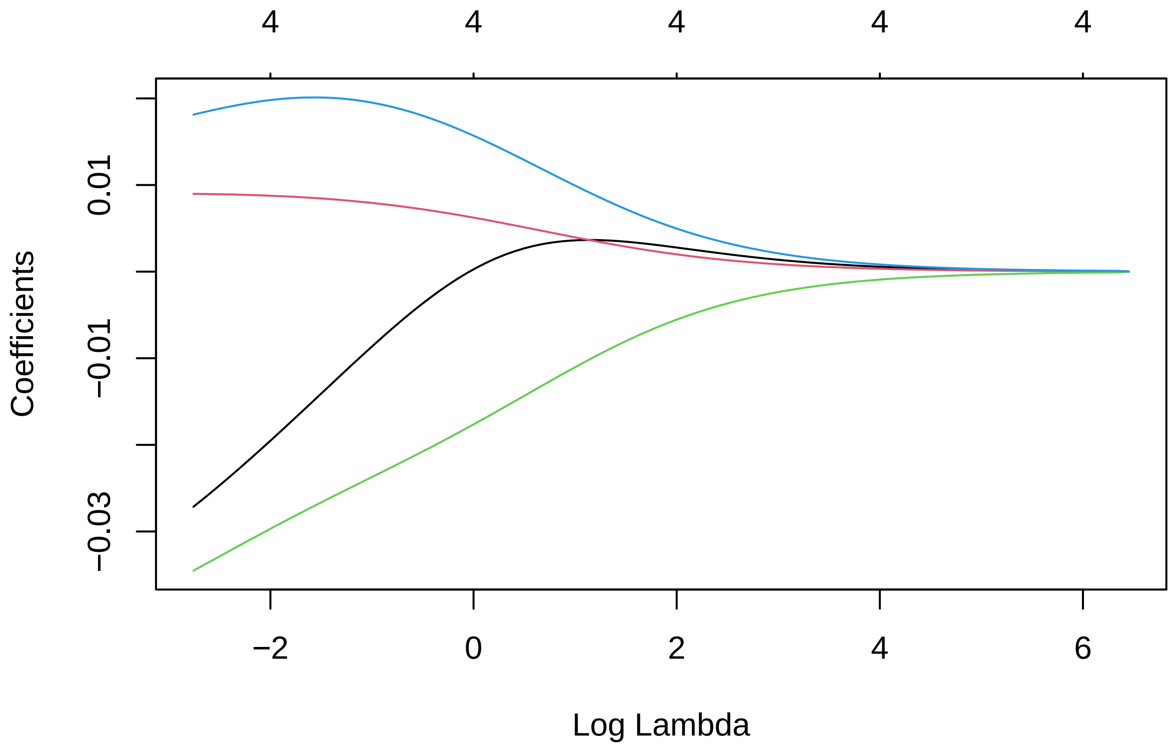
```
#visualization for finding the best lambda
```

```
best_model = glmnet(x, y, alpha = 0, lambda = best_lambda)
coef(best_model)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                                 s0
## (Intercept)            4.431683368
## death_rate            -0.027119913
## infant_mortality_rate  0.008939495
## life_exp_at_birth     -0.034548217
## birth_rate             0.018178334
```

```
#using the best lambda to find the coefficients so that we know what column to drop to address the coll
```

```
plot(model, xvar = "lambda")
```



```
#ridge trace plot
```

The green line on the ridge trace plot represents life_exp_at_birth. It has the coefficient that's furthest away from 0 (-0.034548217), which means it is the least important predictor in our model.

```
lmlogstep_no_col = lm((maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate + birth_rate
summary(lmlogstep_no_col)
```

```
##
## Call:
## lm(formula = (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     birth_rate, data = df_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00872 -0.21522  0.02784  0.21176  0.83474
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.384471   0.106733  12.971  < 2e-16 ***
## death_rate            -0.004806   0.008782  -0.547    0.585
## infant_mortality_rate  0.015740   0.002227   7.068 3.71e-11 ***
## birth_rate             0.029925   0.005339   5.605 8.08e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3243 on 173 degrees of freedom
## Multiple R-squared:  0.8054, Adjusted R-squared:  0.8021 
## F-statistic: 238.7 on 3 and 173 DF,  p-value: < 2.2e-16
```

```r
#new model without life_exp_at_birth, which the ridge plot told us to drop
```

From looking at the summary statistics of our data, we notice that one predictive variable in our regression model has a p-value greater than $\alpha = 0.05$. We continue to see if we can remove death_rate as a predictive variable in our model.

```r
anova(lmlogstep_no_col, lmlogstep)
```

```
## Analysis of Variance Table
## 
## Model 1: (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate + 
##     birth_rate
## Model 2: (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate + 
##     life_exp_at_birth + birth_rate
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1    173 18.190                                  
## 2    172 14.282  1    3.9079 47.063 1.189e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is 1.189e-10, which is lower than $\alpha = 0.05$. Therefore, we reject the null hypothesis and we know that β_5 (life_exp_at_birth) does have an impact on our model. However, we chose to remove it because we want to fix the collinearity problem. We will do the step function again; there is a problem with the p-value for death_rate.

```r
lmlogstep_no_col2 = step(lmlogstep_no_col)
```

```
## Start:  AIC=-394.73
## (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate + 
##     birth_rate
## 
##                         Df Sum of Sq    RSS     AIC
## - death_rate             1    0.0315 18.221 -396.42
## <none>                               18.190 -394.73
## - birth_rate             1    3.3032 21.493 -367.19
## - infant_mortality_rate  1    5.2521 23.442 -351.83
## 
## Step:  AIC=-396.42
## (maternal_mortality_rate)^(1/5) ~ infant_mortality_rate + birth_rate
## 
##                         Df Sum of Sq    RSS     AIC
## <none>                               18.221 -396.42
## - birth_rate             1    4.0569 22.278 -362.84
## - infant_mortality_rate  1    6.1802 24.401 -346.73
```

```r
#we step it again
```

The step function tells us to remove death_rate, which agrees with the fact that the p-value is so high.

```r
s3 = summary(lmlogstep_no_col)
s4 = summary(lmlogstep_no_col2)
s3$adj.r.squared
```

```
## [1] 0.8020654
```

```
s4$adj.r.squared
```

```
## [1] 0.8028622
```

Our R^2 does improve once we remove the column with a high p-value, death_rate, from the model. The p-value was greater than $\alpha = 0.05$, which means it is not a significant predictor variable in our model.

```
anova(lmlogstep_no_col2, lmlogstep_no_col)
```

```
## Analysis of Variance Table
##
## Model 1: (maternal_mortality_rate)^(1/5) ~ infant_mortality_rate + birth_rate
## Model 2: (maternal_mortality_rate)^(1/5) ~ death_rate + infant_mortality_rate +
##     birth_rate
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    174 18.221
## 2    173 18.190  1   0.03149 0.2995 0.5849
```

The p-value is 0.5849, which is higher than $\alpha = 0.05$. Therefore, we fail to reject the null hypothesis and we know that β_3 (death_rate) has no significant impact on our model, so we can remove it and use the reduced model.

```
AIC(lmlogstep_no_col2, lmlogstep_no_col)
```

```
##                   df      AIC
## lmlogstep_no_col2  4 107.8835
## lmlogstep_no_col   5 109.5773
```

The second stepped model has a lower AIC, and a lower AIC means the better the regression model fits the data.

```
summary(lmlogstep_no_col2)
```

```
##
## Call:
## lm(formula = (maternal_mortality_rate)^(1/5) ~ infant_mortality_rate +
##     birth_rate, data = df_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00088 -0.20711  0.02856  0.21028  0.81003
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.339008   0.066876  20.022  < 2e-16 ***
## infant_mortality_rate 0.015183   0.001976   7.682 1.09e-12 ***
## birth_rate            0.030970   0.004976   6.224 3.51e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3236 on 174 degrees of freedom
## Multiple R-squared:  0.8051, Adjusted R-squared:  0.8029
## F-statistic: 359.4 on 2 and 174 DF,  p-value: < 2.2e-16
```

```
#FINAL MODEL!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

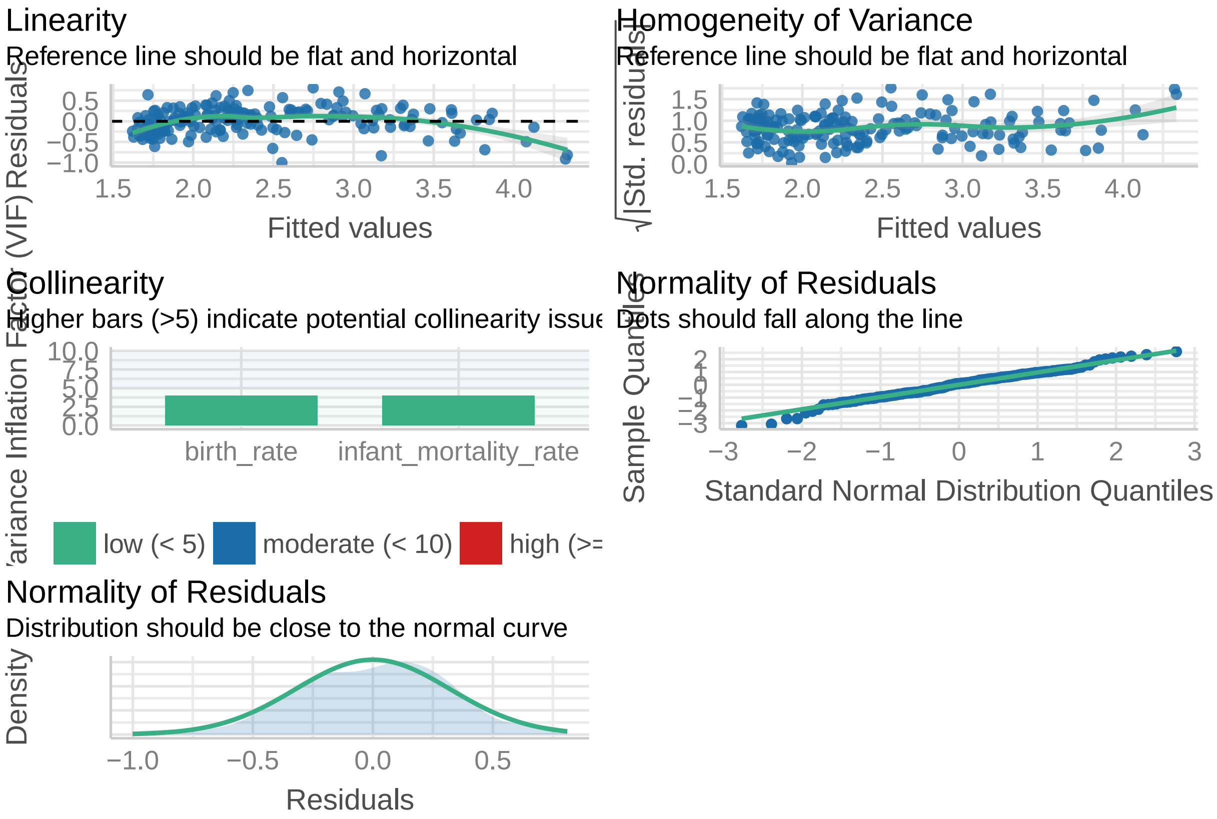
If the null hypothesis is true, we would expect the F value to be close to 1. The F-statistic is 359.4 with the p value of < 2.2e-16. Since the F value is not near 1 and the p value is less than the significance level of $\alpha = 0.05$, we reject H_0 and we know that the data claims there is a relationship between the response, maternal_mortality_rate, and at least one predictor in the model.

$$Maternal\widehat{Mortality}Rate^{1/5} = 1.339008 + 0.015183(InfantMortalityRate) + 0.030970(BirthRate)$$

A one unit increase in infant_mortality_rate (1 more death per 1,000 live births), with the other predictor (birth_rate) held fixed, is associated with an increase in maternal_moratality_rate by $(0.015183)^5$ units, which equals 8.06841002E-10 units, which can be interpreted as 8.06841002E-10 more deaths (where the death is related to pregnancy or birth) per 100,000 live births.

A one unit increase in birth_rate (1 birth per 1000 people), with the other predictor (infant_mortality_rate) held fixed, is associated with an increase in maternal_moratality_rate by $(0.030970)^5$ units, which equals 2.84908907E-8 units, which can be interpreted as 2.84908907E-8 more deaths (where the death is related to pregnancy or birth) per 100,000 live births.

```
performance::check_model(lmlogstep_no_col2)
```



```
shapiro.test(resid(lmlogstep_no_col2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lmlogstep_no_col2)
## W = 0.99015, p-value = 0.2621
```

H_0 : The data is normally distributed

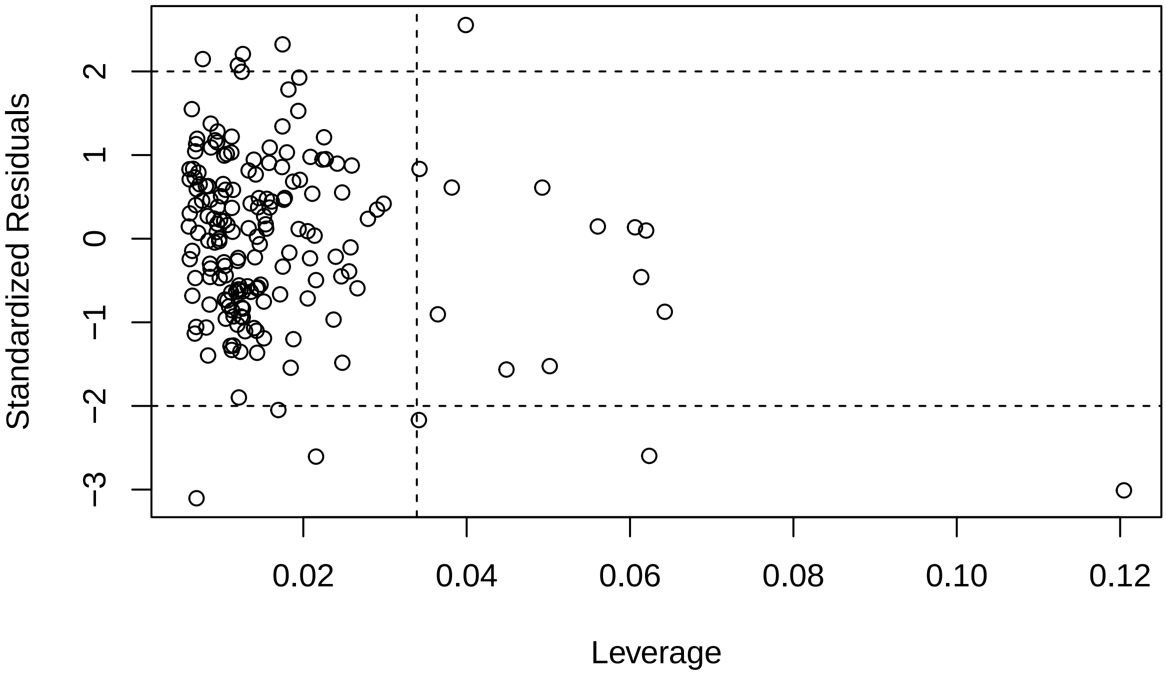
H_A : The data is not normally distributed

A W value that's greater than 0.95 indicates that the data is normal. Also, the p-value is greater than

$\alpha = 0.05$, which means we fail to reject H_0 :, and we know the data is normally distributed.

```
p <- 2
n <- nrow(df_final)

plot(hatvalues(lmlogstep_no_col2), rstandard(lmlogstep_no_col2),
xlab = 'Leverage', ylab = 'Standardized Residuals')
abline(v = 2*(p+1)/n, lty=2) #cutoff for leverage points
abline(h = c(-2, 2), lty = 2) #cutoff for outliers
```



```
high_sr = which(abs(rstandard(lmlogstep_no_col2)) > 2)
df_final[high_sr,]
```

```
##      maternal_mortality_rate    area birth_rate death_rate infant_mortality_rate
## 13                       220 1904569      17.04       6.34                 25.16
## 14                       730 1861484      30.01       7.87                 52.86
## 16                        21 1648195      18.23       5.94                 39.00
## 21                       540 1240192      45.53      13.22                104.34
## 37                       460  652230      38.84      14.12                117.23
## 56                       570  390757      32.47      10.62                 26.55
## 70                       300  274200      42.42      11.96                 76.80
## 79                       280  214969      15.90       7.30                 33.56
## 131                      620   30355      25.92      14.91                 50.48
## 165                       70     964      35.12       7.45                 49.16
## 166                        9     811      21.85       7.18                 35.37
##      internet_users life_exp_at_birth net_migration_rate population
## 13         20000000             72.17              -1.18  253609643
## 14         4200000             63.32              -4.36   35482233
## 16         8214000             70.89              -0.08   80840713
## 21          249800             54.95              -2.33   16455903
## 37         1000000             50.49              -1.83   31822848
## 56         1423000             55.68              21.78   13771721
## 70          178100             54.78               0.00   18365123
## 79          189600             67.81              -9.67     735554
## 131          76800             52.65              -7.62    1942008
## 165          26700             64.22              -8.79     190428
## 166           7800             65.47              -2.86     104488
##     population_growth_rate
## 13                    0.95
## 14                    1.78
## 16                    1.22
## 21                    3.00
## 37                    2.29
## 56                    4.36
## 70                    3.05
## 79                   -0.11
## 131                   0.34
## 165                   1.89
## 166                   1.18
```

```
high_leverage = which(abs(hatvalues(lmlogstep_no_col2)) > .033)
df_final[high_leverage,]
```

```
##     maternal_mortality_rate    area birth_rate death_rate infant_mortality_rate
## 19                     1100 1284000      37.29      14.56                 90.30
## 20                      590 1267000      46.12      12.73                 86.27
## 21                      540 1240192      45.53      13.22                104.34
## 28                      460  947300      36.82       8.20                 43.74
## 35                      440  752618      42.46      12.92                 66.62
## 37                      460  652230      38.84      14.12                117.23
## 39                     1000  637657      40.87      13.91                100.14
## 40                      890  622984      35.45      14.11                 92.86
## 56                      570  390757      32.47      10.62                 26.55
## 70                      300  274200      42.42      11.96                 76.80
## 75                      310  241038      44.17      10.97                 60.82
## 93                      460  118484      41.80       8.74                 48.01
## 128                     790   36125      33.83      14.54                 90.92
## 136                     800   27830      42.33       9.54                 63.44
## 174                      64     360      32.20       3.09                 15.46
##     internet_users life_exp_at_birth net_migration_rate population
## 19          168100             49.44              -3.54   11412107
## 20          115900             54.74              -0.58   17466172
## 21          249800             54.95              -2.33   16455903
## 28          678000             61.24              -0.57   49639138
## 35          816200             51.83              -0.72   14638505
## 37         1000000             50.49              -1.83   31822848
## 39          106000             51.58              -9.51   10428043
## 40           22600             51.35               0.00    5277959
## 56         1423000             55.68              21.78   13771721
## 70          178100             54.78               0.00   18365123
## 75         3200000             54.46              -0.76   35918915
## 93          716400             59.99               0.25   17377468
## 128          37100             49.87              0.00    1693398
## 136         157800             59.55              0.00   10395931
## 174        1379000             74.64              0.00    1816379
##     population_growth_rate
## 19                    1.92
## 20                    3.28
## 21                    3.00
## 28                    2.80
## 35                    2.88
## 37                    2.29
## 39                    1.75
## 40                    2.13
## 56                    4.36
## 70                    3.05
## 75                    3.24
## 93                    3.33
## 128                   1.93
## 136                   3.28
## 174                   2.91
```