Can GDP be predicted by standard of living factors?

Erin Xu & Dora Dong

2025-05-19

Load data and check if needed to clean

```r
all_data <- read.csv("life_expectancy.csv", na.strings = c("", "NA"))
anyNA(all_data)
```

```
## [1] TRUE
```

```r
sum(is.na(all_data))
```

```
## [1] 2563
```

```r
colSums(is.na(all_data))
```

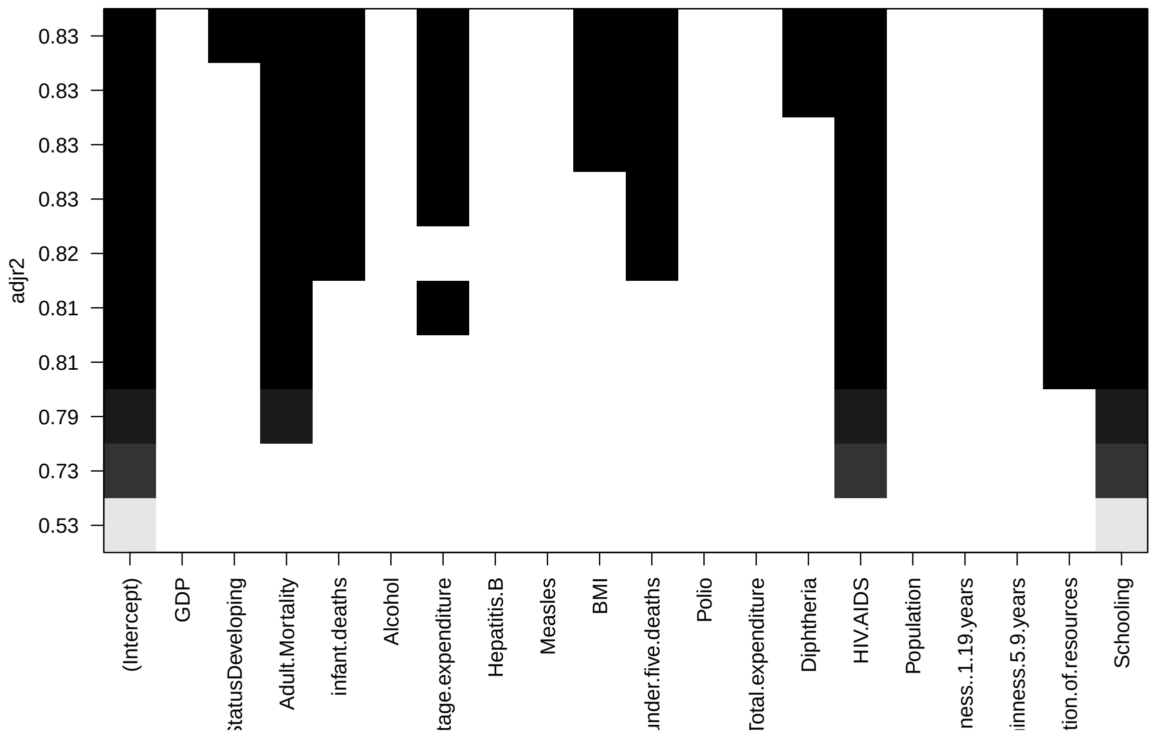
```
##                         Country                            Year
##                               0                               0
##                          Status                 Life.expectancy
##                               0                              10
##                 Adult.Mortality                   infant.deaths
##                              10                               0
##                         Alcohol          percentage.expenditure
##                             194                               0
##                     Hepatitis.B                         Measles
##                             553                               0
##                             BMI               under.five.deaths
##                              34                               0
##                           Polio               Total.expenditure
##                              19                             226
##                      Diphtheria                        HIV.AIDS
##                              19                               0
##                             GDP                      Population
##                             448                             652
##             thinness..1.19.years              thinness.5.9.years
##                              34                              34
## Income.composition.of.resources                       Schooling
##                             167                             163
```

Decide predictors

```r
all_data <- read.csv("life_expectancy.csv")
all_data <- na.omit(all_data)

all_data$Status <- as.factor(all_data$Status)


# Clean model
full_model <- lm(GDP ~ Life.expectancy + Status + Adult.Mortality +
                   infant.deaths + Alcohol + percentage.expenditure +
                   Hepatitis.B + Measles + BMI + under.five.deaths + Polio +
                   Total.expenditure + Diphtheria + HIV.AIDS + Population +
                   thinness..1.19.years + thinness.5.9.years +
                   Income.composition.of.resources + Schooling,
                 data = all_data)
summary(full_model)

step_model <- step(full_model, direction = "both")
summary(step_model)
```

Best subset selection

```r
best_model <- regsubsets(Life.expectancy ~ GDP + Status + Adult.Mortality +
                           infant.deaths + Alcohol +
                           percentage.expenditure + Hepatitis.B + Measles +
                           BMI + under.five.deaths + Polio +
                           Total.expenditure + Diphtheria + HIV.AIDS +
                           Population + thinness..1.19.years +
                           thinness.5.9.years + Income.composition.of.resources
                          + Schooling, data = all_data, nvmax = 10)

# Summary and plot
best_summary <- summary(best_model)
plot(best_model, scale = "adjr2")
```



We propose the following multiple linear regression model:

$$GDP = E(log(GDP) + e =$$

$$b_0 + b_1 PercentageExpenditure + b_2 Polio + b_3 Population + b_4 IncomeCompositionofResources + b_5 Schooling + b_6 Status$$

Get the response and predictors.

```
all_data <- read.csv("life_expectancy.csv")
all_data <- na.omit(all_data)
all_data$log_GDP <- log(all_data$GDP)

response <- all_data$log_GDP
x0 <- all_data$percentage.expenditure
x1 <- all_data$Polio
x2 <- all_data$Population
x3 <- all_data$Income.composition.of.resources
x4 <- all_data$Schooling

all_data$Status <- as.factor(all_data$Status)

model <- lm(response ~ Status + x0 + x1 + x2 + x3 + x4, data = all_data)
summary(model)
```

```
##
## Call:
## lm(formula = response ~ Status + x0 + x1 + x2 + x3 + x4, data = all_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3838 -0.6281  0.3300  0.8680  2.4999
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.656e+00  2.280e-01  16.037  < 2e-16 ***
## StatusDeveloping   5.719e-02  1.064e-01   0.538    0.591
## x0                 3.862e-04  2.030e-05  19.019  < 2e-16 ***
## x1                -7.981e-04  1.459e-03  -0.547    0.585
## x2                -1.866e-10  4.348e-10  -0.429    0.668
## x3                 1.398e+00  2.733e-01   5.115 3.51e-07 ***
## x4                 2.086e-01  1.871e-02  11.152  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.241 on 1642 degrees of freedom
## Multiple R-squared:  0.5001, Adjusted R-squared:  0.4983
## F-statistic: 273.8 on 6 and 1642 DF,  p-value: < 2.2e-16
```

We estimate the deterministic model as

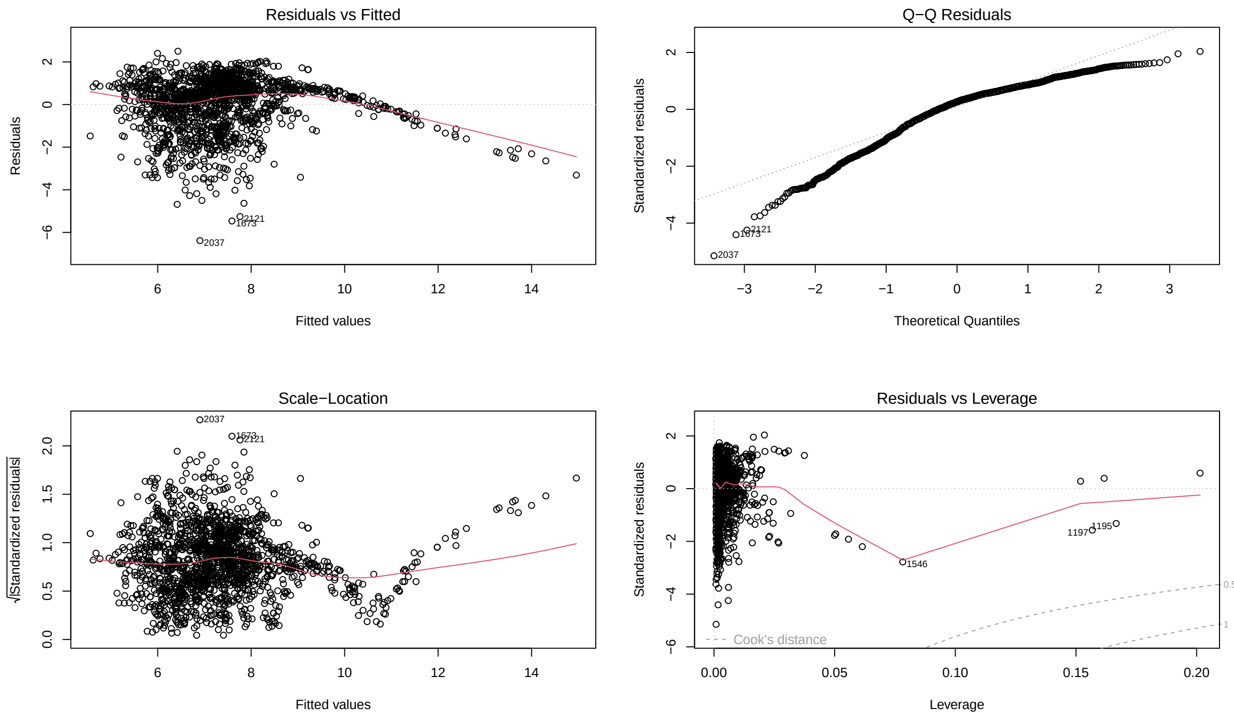
$$G\hat{D}P = exp(\hat{b_0} + \hat{b_1} PercentageExpenditure + \hat{b_2} Polio +$$

$$\hat{b_3} Population + \hat{b_4} IncomeCompositionofResources + \hat{b_5} Schooling + \hat{b_6} Status)$$

by using the *lm* function to find the values of the coefficients that minimize the RSS.

Residual plots

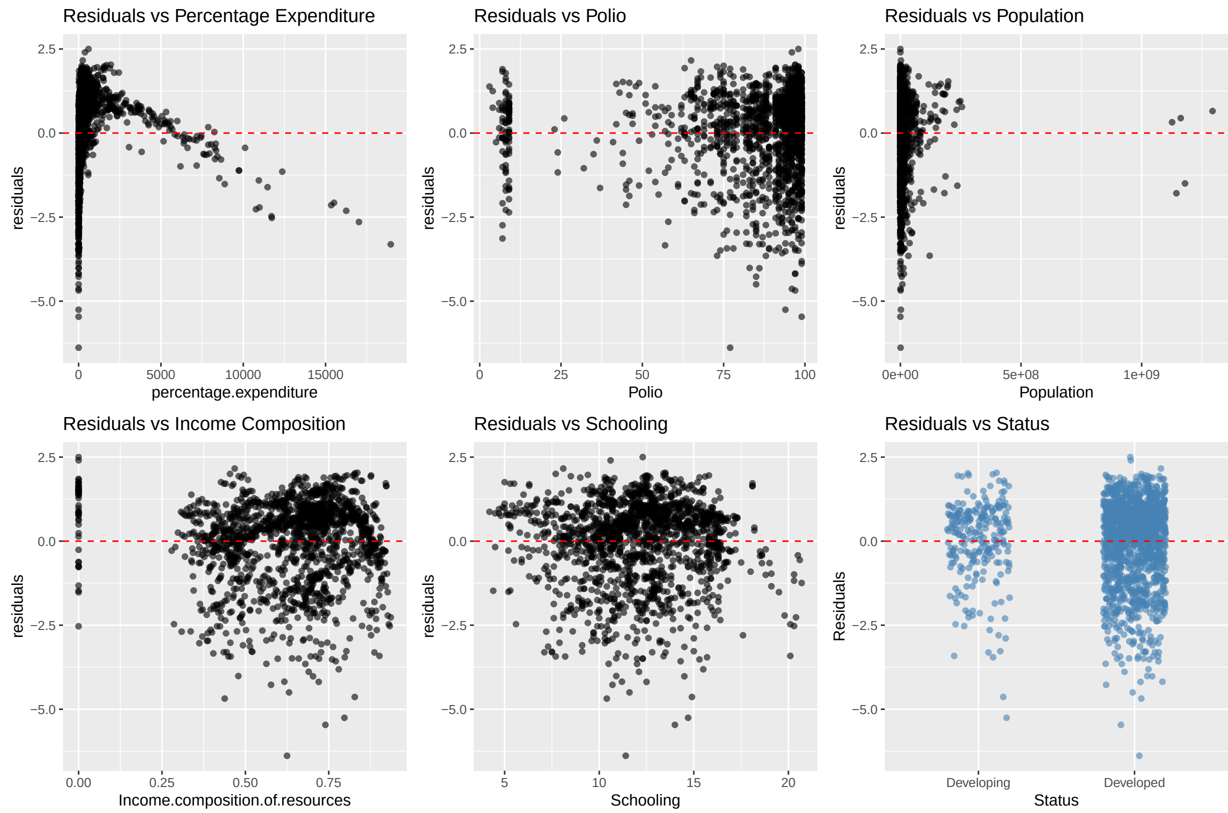
```
par(mfrow = c(2, 2))
plot(model)
```



Residual VS each predictor (Regression assumptions)

```
all_data$residuals <- resid(model)
all_data$fitted <- fitted(model)

p1 <- ggplot(all_data, aes(x = `percentage.expenditure`, y = residuals)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Percentage Expenditure")

p2 <- ggplot(all_data, aes(x = Polio, y = residuals)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Polio")

p3 <- ggplot(all_data, aes(x = Population, y = residuals)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Population")

p4 <- ggplot(all_data, aes(x = `Income.composition.of.resources`,
                           y = residuals)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Income Composition")
```

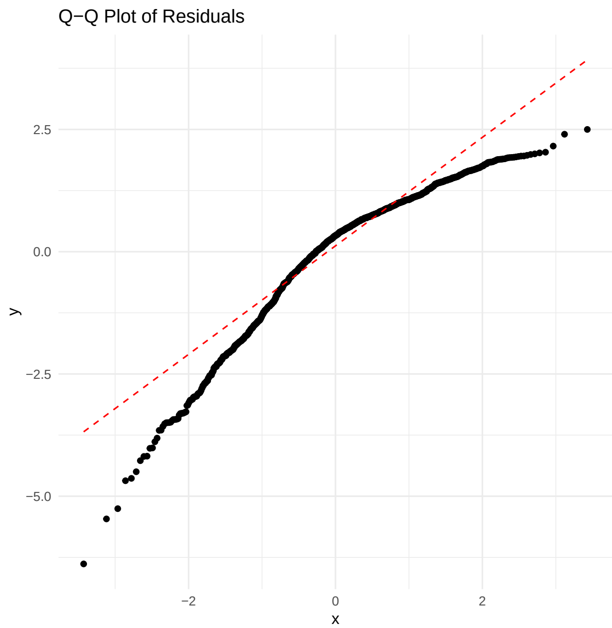
```r
p5 <- ggplot(all_data, aes(x = Schooling, y = residuals)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Schooling")

p6 <- ggplot(all_data, aes(x = as.factor(Status), y = residuals)) +
  geom_jitter(width = 0.2, alpha = 0.6, color = "steelblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  scale_x_discrete(labels = c("Developing", "Developed")) +
  labs(title = "Residuals vs Status",
       x = "Status",
       y = "Residuals")

(p1 | p2 | p3) / (p4 | p5 | p6)
```

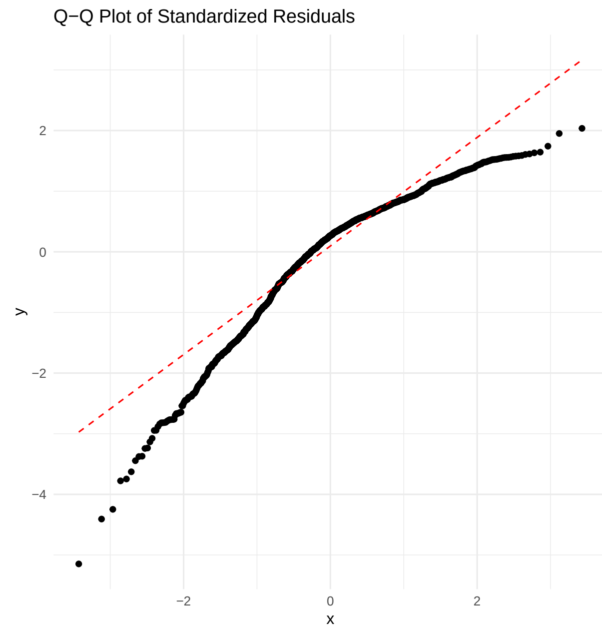


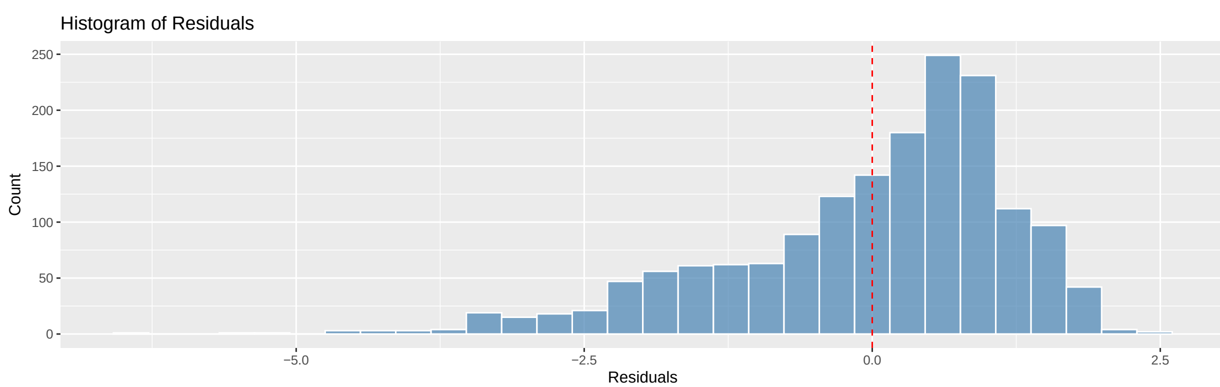
Residual VS Fitted plot (Regression assumptions)

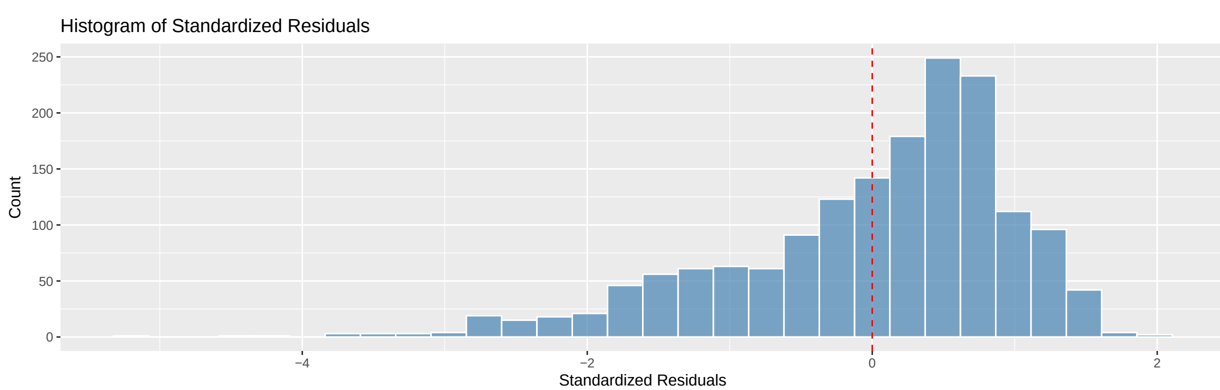
```r
# Store residuals only once (already done earlier)
all_data$residuals <- resid(model)
all_data$std_residuals <- rstandard(model)

# Q-Q Plot
qq_plot <- ggplot(all_data, aes(sample = residuals)) +
```

```r
  stat_qq() +
  stat_qq_line(color = "red", linetype = "dashed") +
  labs(title = "Q-Q Plot of Residuals") +
  theme_minimal() +  # Optional: adds a cleaner theme that may help with rendering
  theme(aspect.ratio = 1)  # Makes the plot square, which often helps QQ plots display correctly

# Standardized Q-Q Plot
std_qq_plot <- ggplot(all_data, aes(sample = std_residuals)) +
  stat_qq() +
  stat_qq_line(color = "red", linetype = "dashed") +
  labs(title = "Q-Q Plot of Standardized Residuals") +
  theme_minimal() +
  theme(aspect.ratio = 1)

# Histogram
hist_plot <- ggplot(all_data, aes(x = residuals)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white", alpha = 0.7) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Count")

std_hist_plot <- ggplot(all_data, aes(x = std_residuals)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white", alpha = 0.7) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Histogram of Standardized Residuals",
       x = "Standardized Residuals",
       y = "Count")


(qq_plot | std_qq_plot) / (hist_plot / std_hist_plot)
```


Q−Q Plot of Residuals
y
x
2.5
0.0
−2.5
−5.0
−2
0
2


Q−Q Plot of Standardized Residuals
y
x
2
0
−2
−4
−2
0
2


Histogram of Residuals
Count
Residuals
250
200
150
100
50
0
−5.0
−2.5
0.0
2.5


Histogram of Standardized Residuals
Count
Standardized Residuals
250
200
150
100
50
0
−4
−2
0
2