HM4

Mu Cheng

3/15/2021

Problem1

Step1: Call the Original Dataframe of the Life_Expectancy

```
getwd()
```

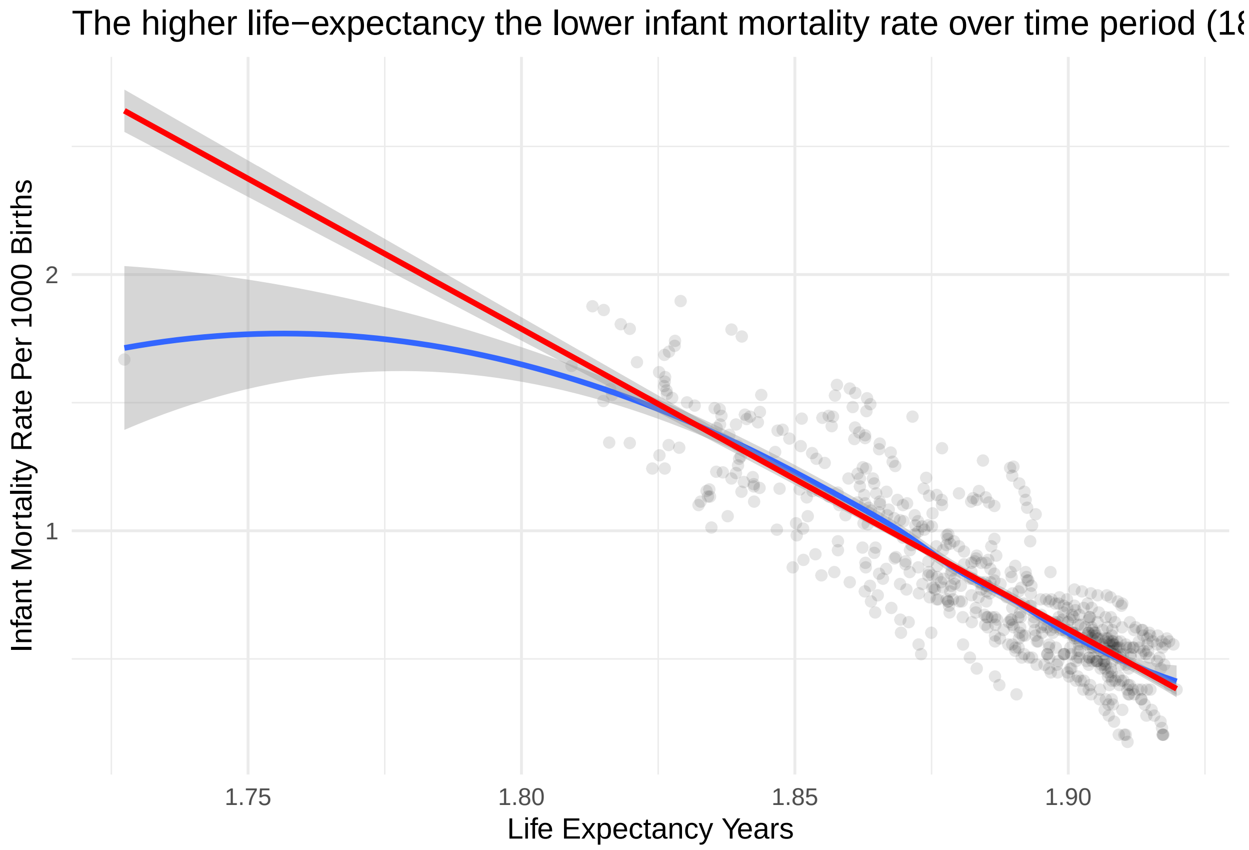
```
## [1] "/Users/mucheng/Desktop/5110/HM4"
```

```
setwd("~/Desktop/5110/HM4/ddf--gapminder--systema_globalis-master/countries-etc-datapoints")
library("readr")
library(ggplot2)
library(RSQLite)
library(dbplyr)
library(DBI)

life_expectancy_ori <- read_csv("ddf--datapoints--life_expectancy_years--by--geo--time.csv")
life_expectancy_ori <- na.omit(life_expectancy_ori)


setwd("~/Desktop/5110/HM4/ddf--gapminder--systema_globalis-master/countries-etc-datapoints")
infant_mortality_ori <- read_csv(
  "ddf--datapoints--infant_mortality_rate_per_1000_births--by--geo--time.csv")
infant_mortality_ori <- na.omit(infant_mortality_ori)


setwd("~/Desktop/5110/HM4/ddf--gapminder--systema_globalis-master/countries-etc-datapoints")
murder_ori <- read_csv(
  "ddf--datapoints--murder_per_100000_people--by--geo--time.csv")
murder_ori <- na.omit(murder_ori)


setwd("~/Desktop/5110/HM4/ddf--gapminder--systema_globalis-master/countries-etc-datapoints")
GDP_inflation_ori <- read_csv(
  "ddf--datapoints--total_gdp_us_inflation_adjusted--by--geo--time.csv")
GDP_inflation_ori <- na.omit(GDP_inflation_ori)


setwd("~/Desktop/5110/HM4/ddf--gapminder--systema_globalis-master/countries-etc-datapoints")
MD_ori <- read_csv(
  "ddf--datapoints--medical_doctors_per_1000_people--by--geo--time.csv")
MD_ori <- na.omit(MD_ori)


setwd("~/Desktop/5110/HM4/ddf--gapminder--systema_globalis-master/countries-etc-datapoints")
```

```
poverty_rate_ori <- read_csv(
  "ddf--datapoints--poverty_percent_people_below_550_a_day--by--geo--time.csv")
poverty_rate_ori <- na.omit(poverty_rate_ori)


con <- dbConnect(SQLite(), ":memory:")
dbWriteTable(con, "life_expectancy_ori", life_expectancy_ori)
dbWriteTable(con, "infant_mortality_ori", infant_mortality_ori)
dbWriteTable(con, "murder_ori", murder_ori)
dbWriteTable(con, "GDP_inflation_ori", GDP_inflation_ori)
dbWriteTable(con, "MD_ori", MD_ori)
dbWriteTable(con, "poverty_rate_ori", poverty_rate_ori)

LE_data1 <-dbGetQuery(con, "SELECT DISTINCT
infant_mortality_ori.geo,
infant_mortality_ori.time,
infant_mortality_ori.infant_mortality_rate_per_1000_births,
life_expectancy_ori.life_expectancy_years
FROM life_expectancy_ori
INNER JOIN infant_mortality_ori
WHERE infant_mortality_ori.geo = life_expectancy_ori.geo
AND infant_mortality_ori.time=life_expectancy_ori.time")
dbWriteTable(con, "LE_data1", LE_data1)


LE_data2 <-dbGetQuery(con, "SELECT DISTINCT
murder_ori.geo,
murder_ori.time,
murder_ori.murder_per_100000_people,
GDP_inflation_ori.total_gdp_us_inflation_adjusted
FROM GDP_inflation_ori
INNER JOIN murder_ori
WHERE murder_ori.geo=GDP_inflation_ori.geo
AND murder_ori.time=GDP_inflation_ori.time")
dbWriteTable(con, "LE_data2", LE_data2)


LE_data3 <-dbGetQuery(con, "SELECT DISTINCT
MD_ori.geo,
MD_ori.time,
MD_ori.medical_doctors_per_1000_people,
poverty_rate_ori.poverty_percent_people_below_550_a_day
FROM poverty_rate_ori
INNER JOIN MD_ori
WHERE MD_ori.geo = poverty_rate_ori.geo
AND MD_ori.time=poverty_rate_ori.time")
dbWriteTable(con, "LE_data3", LE_data3)


LE_data4 <- dbGetQuery(con, "SELECT
LE_data1.geo,
LE_data1.time,
LE_data1.infant_mortality_rate_per_1000_births,
LE_data1.life_expectancy_years,
LE_data2.murder_per_100000_people,
LE_data2.total_gdp_us_inflation_adjusted
FROM LE_data1
JOIN LE_data2
WHERE LE_data2.geo=LE_data1.geo
AND LE_data2.time=LE_data1.time
")
dbWriteTable(con, "LE_data4", LE_data4)

LE_data <- dbGetQuery(con, "SELECT
LE_data4.geo,
LE_data4.time,
LE_data4.infant_mortality_rate_per_1000_births,
LE_data4.life_expectancy_years,
LE_data4.murder_per_100000_people,
LE_data4.total_gdp_us_inflation_adjusted,
LE_data3.medical_doctors_per_1000_people,
LE_data3.poverty_percent_people_below_550_a_day
FROM LE_data4
JOIN LE_data3
WHERE LE_data3.geo=LE_data4.geo
AND LE_data3.time=LE_data4.time
")
dbWriteTable(con, "LE_data", LE_data)
```

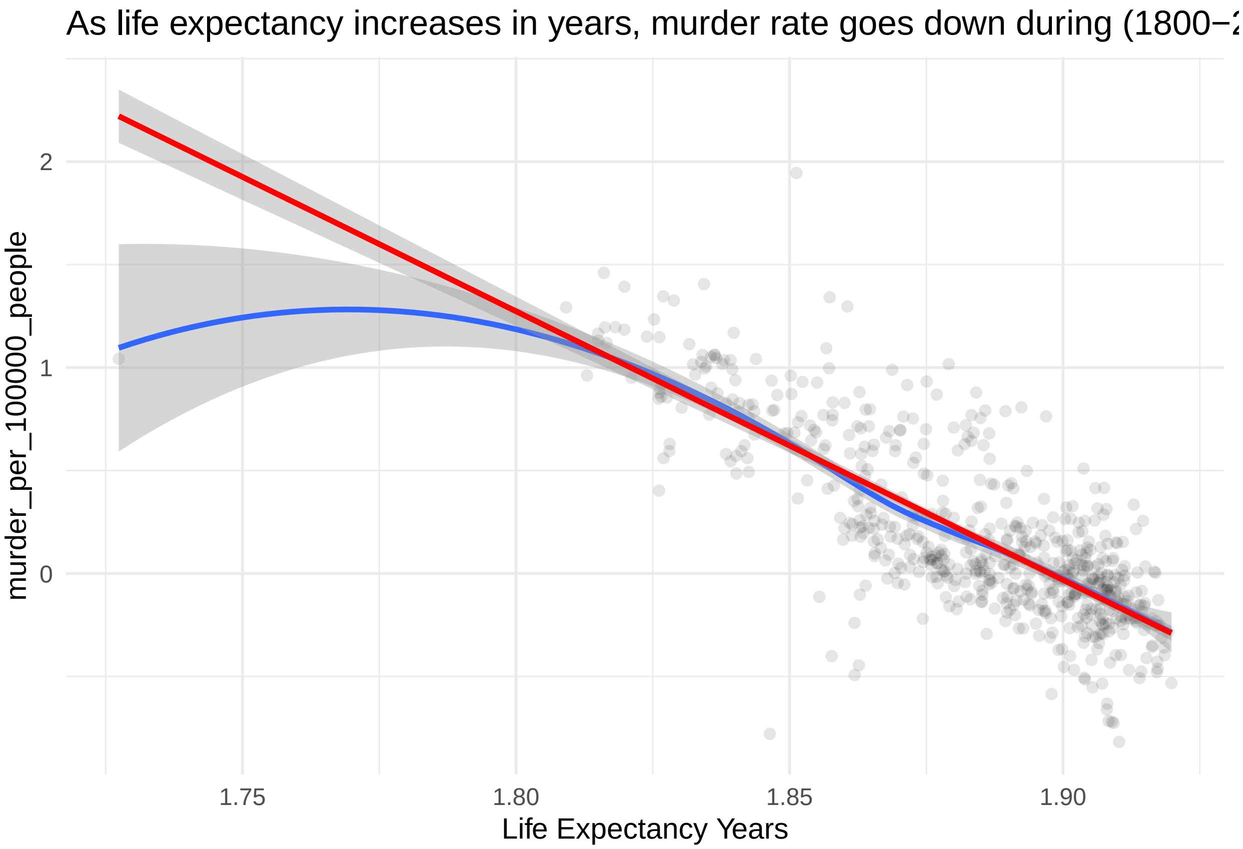
le1: life__expectancy__years against infant__mortality__rate__per__1000__births. Log10 transformation on both variables.

```
le1<- ggplot(LE_data, mapping=aes(x=log10(life_expectancy_years),
                                  y=log10(infant_mortality_rate_per_1000_births))) +
  geom_point(alpha=0.1) + geom_smooth(method = 'loess', formula='y ~ x') + geom_smooth(method="lm", colo
  labs(x="Life Expectancy Years",
       y="Infant Mortality Rate Per 1000 Births",
       title=
"The higher life-expectancy the lower infant mortality rate over time period (1800-2015)") +
  theme_minimal()
le1
```



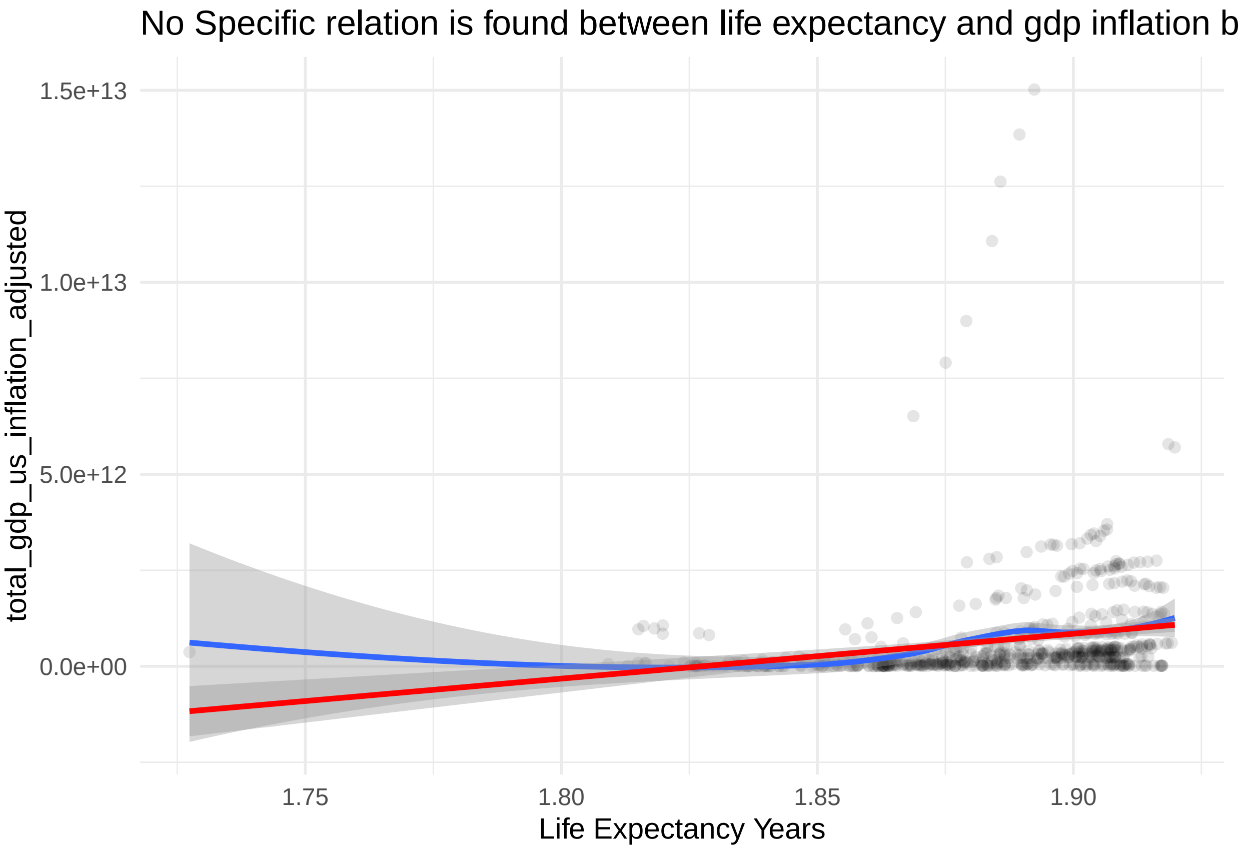
le2: life_expectancy_years against murder_per_100000_people. Log10 transformations on both variables.

```
le2<- ggplot(LE_data, mapping=aes(x=log10(life_expectancy_years),
                                  y=log10(murder_per_100000_people))) +
  geom_point(alpha=0.1) + geom_smooth(method = 'loess', formula='y ~ x') +  geom_smooth(method="lm", col
  labs(x="Life Expectancy Years",
       y="murder_per_100000_people",
       title=
"As life expectancy increases in years, murder rate goes down during (1800-2015)") + theme_minimal()
le2
```



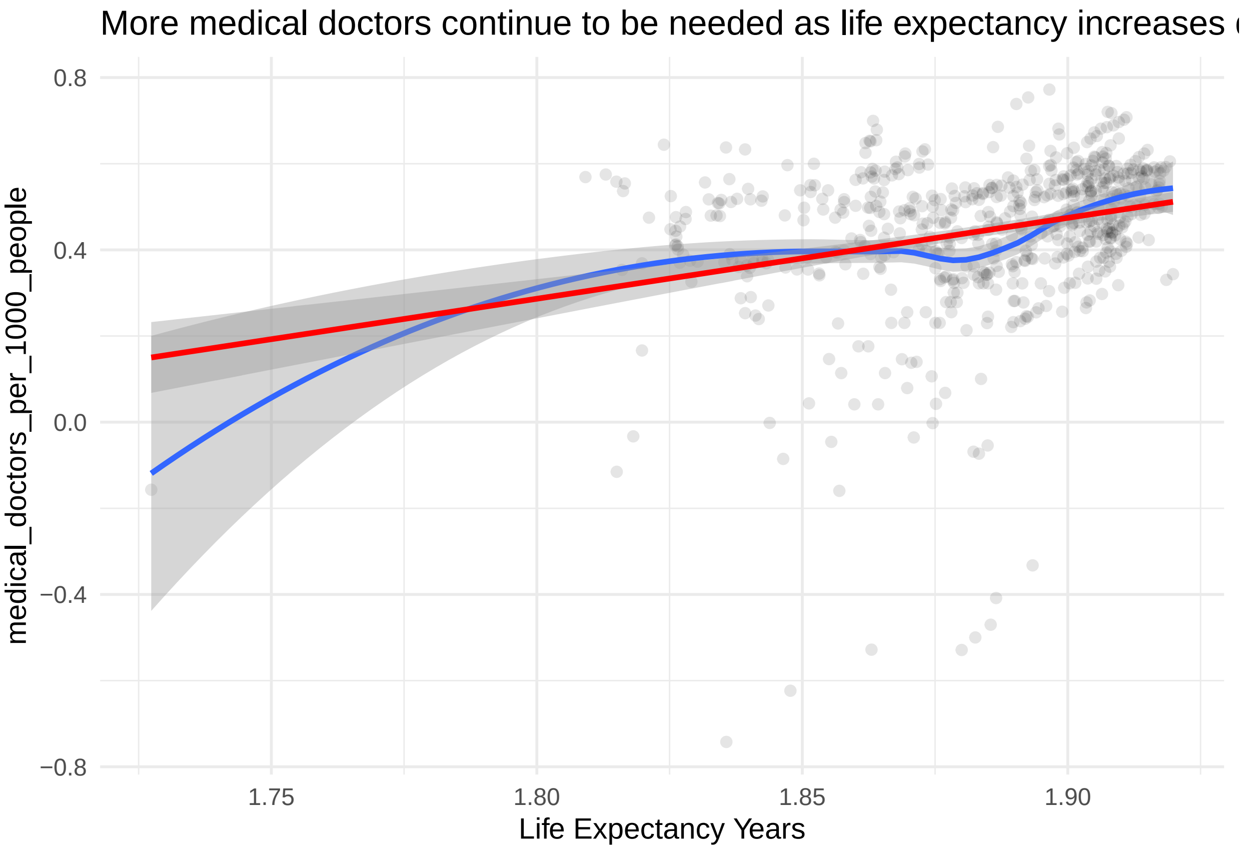
le3: life_expectancy_years against total_gdp_us_inflation_adjusted. Log10 transformations on x variable.

```
le3 <- ggplot(LE_data, mapping=aes(x=log10(life_expectancy_years),
                                   y=total_gdp_us_inflation_adjusted)) +
  geom_point(alpha=0.1) + geom_smooth(method = 'loess', formula='y ~ x') +  geom_smooth(method="lm", col
  labs(x="Life Expectancy Years",
       y="total_gdp_us_inflation_adjusted",
       title=
"No Specific relation is found between life expectancy and gdp inflation between (1800-2015)") + theme_m
le3
```



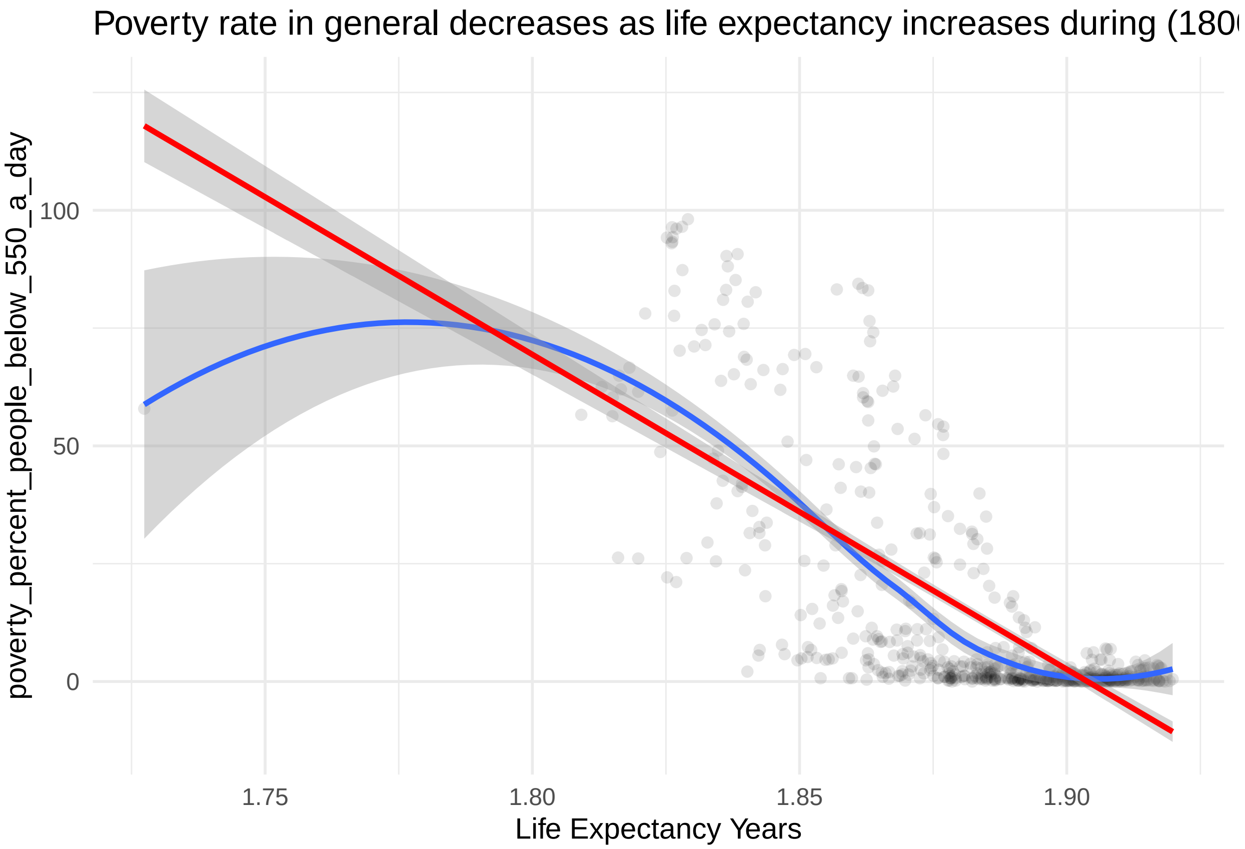
le4: life_expectancy_years against medical_doctors_per_1000_people. Log10 transformations on both variables.

```
le4 <- ggplot(LE_data, mapping=aes(x=log10(life_expectancy_years),
                                  y=log10(medical_doctors_per_1000_people))) +
  geom_point(alpha=0.1) + geom_smooth(method = 'loess', formula='y ~ x') +  geom_smooth(method="lm", col
  labs(x="Life Expectancy Years",
       y="medical_doctors_per_1000_people",
       title=
"More medical doctors continue to be needed as life expectancy increases during (1800-2015)") + theme_mi
le4
```



le5: life_expectancy_years against poverty_percent_people_below_550_a_day. Log10 transformations on x variable.

```
le5 <- ggplot(LE_data, mapping=aes(x=log10(life_expectancy_years),
                                  y=poverty_percent_people_below_550_a_day)) +
  geom_point(alpha=0.1) + geom_smooth(method = 'loess', formula='y ~ x') +
  geom_smooth(method="lm", color="red") +
  labs(x="Life Expectancy Years",
       y="poverty_percent_people_below_550_a_day",
       title=
"Poverty rate in general decreases as life expectancy increases during (1800-2015)") +
  theme_minimal()
le5
```



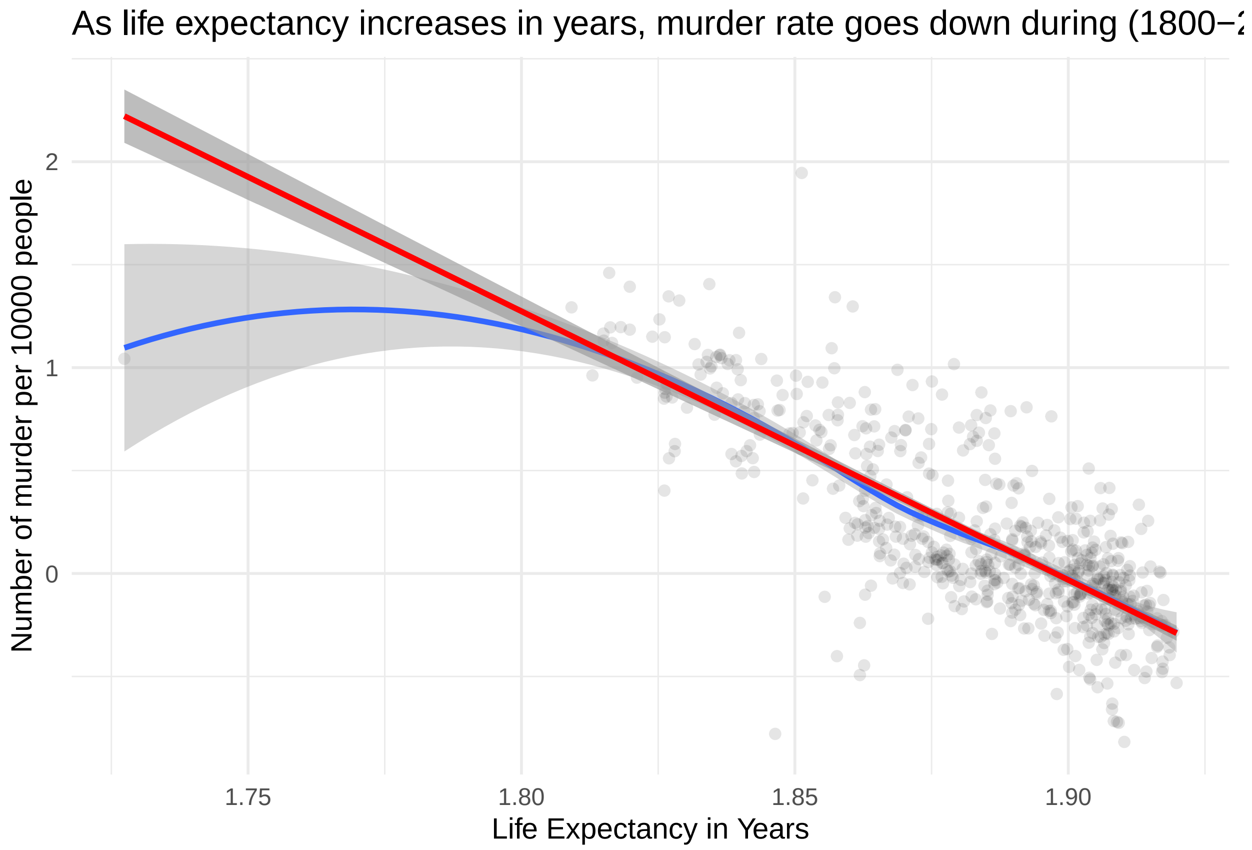
Problem2

Based on previous graphs, the response variable, Murder__per__10000__people, has the best linearity among all the plots. Therefore, Murder__per__10000__people is utilized here as a predictor.

Residual graph demonstrates a clear pattern(increasing line) when plotting without log transformation, with r squared value around 0.3115. It's in violation of model assumptions(linearity) Therefore, adjustment of adding log10() to both variables is made. Then the value of r squared is better with 0.6112. After making the adjustments, the plotted residual graph with a better model (fit1) is then normal in demonstration.

Step 1. Compare better fit vs. bad fit's summary stats.

```
library(tidyverse)
library(modelr)
library(broom)

murder_graph <- ggplot(LE_data, aes(x=log10(life_expectancy_years),
                                    y=log10(murder_per_100000_people))) +
  geom_point(alpha=0.1) + geom_smooth() + geom_smooth(method="lm", color="red") +
  labs(x="Life Expectancy in Years",
       y="Number of murder per 10000 people",
    title="As life expectancy increases in years, murder rate goes down during (1800-2015)") +
  geom_smooth(method="lm", color="red") + theme_minimal()
murder_graph
```



```r
fit1 <- lm(log10(life_expectancy_years) ~ log10(murder_per_100000_people),
           data=LE_data)
summary(fit1)
```

```
##
## Call:
## lm(formula = log10(life_expectancy_years) ~ log10(murder_per_100000_people),
##     data = LE_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.115893 -0.011049  0.000667  0.011518  0.050269
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      1.8921275  0.0007043 2686.50   <2e-16 ***
## log10(murder_per_100000_people) -0.0468496  0.0014932  -31.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01625 on 626 degrees of freedom
## Multiple R-squared:  0.6113, Adjusted R-squared:  0.6107
## F-statistic: 984.5 on 1 and 626 DF,  p-value: < 2.2e-16
```

```r
glance(fit1)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared  sigma statistic   p.value    df logLik    AIC    BIC
##       <dbl>         <dbl>  <dbl>     <dbl>     <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1     0.611         0.611 0.0163      984. 1.45e-130     1  1697. -3388. -3375.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
fitbad <- lm(life_expectancy_years ~ murder_per_100000_people,
           data=LE_data)
summary(fitbad)
```

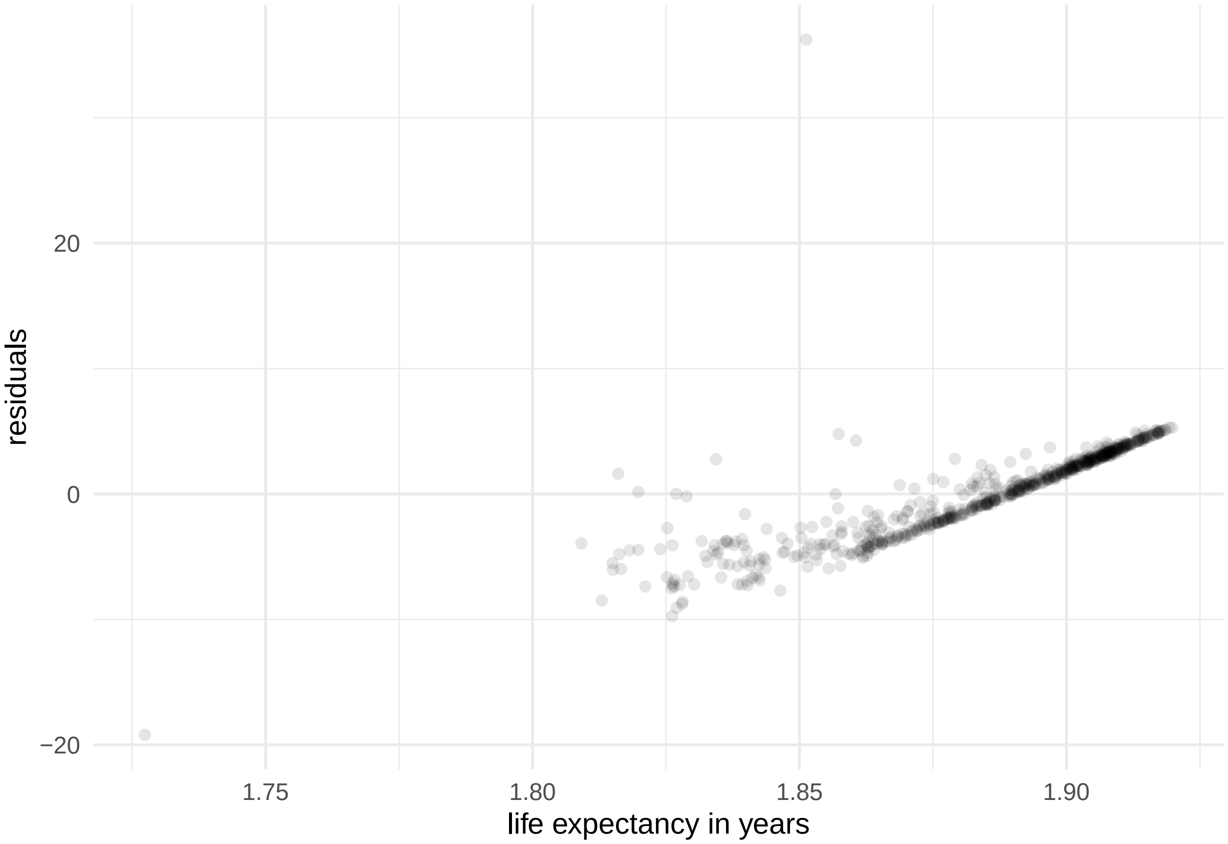
```
##
## Call:
## lm(formula = life_expectancy_years ~ murder_per_100000_people,
##     data = LE_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.203  -2.598   0.592   2.802  36.236
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              77.99415    0.16909  461.25   <2e-16 ***
## murder_per_100000_people -0.49028    0.02913  -16.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.7 on 626 degrees of freedom
## Multiple R-squared:  0.3115, Adjusted R-squared:  0.3104
## F-statistic: 283.3 on 1 and 626 DF,  p-value: < 2.2e-16
```

```
glance(fitbad)
```

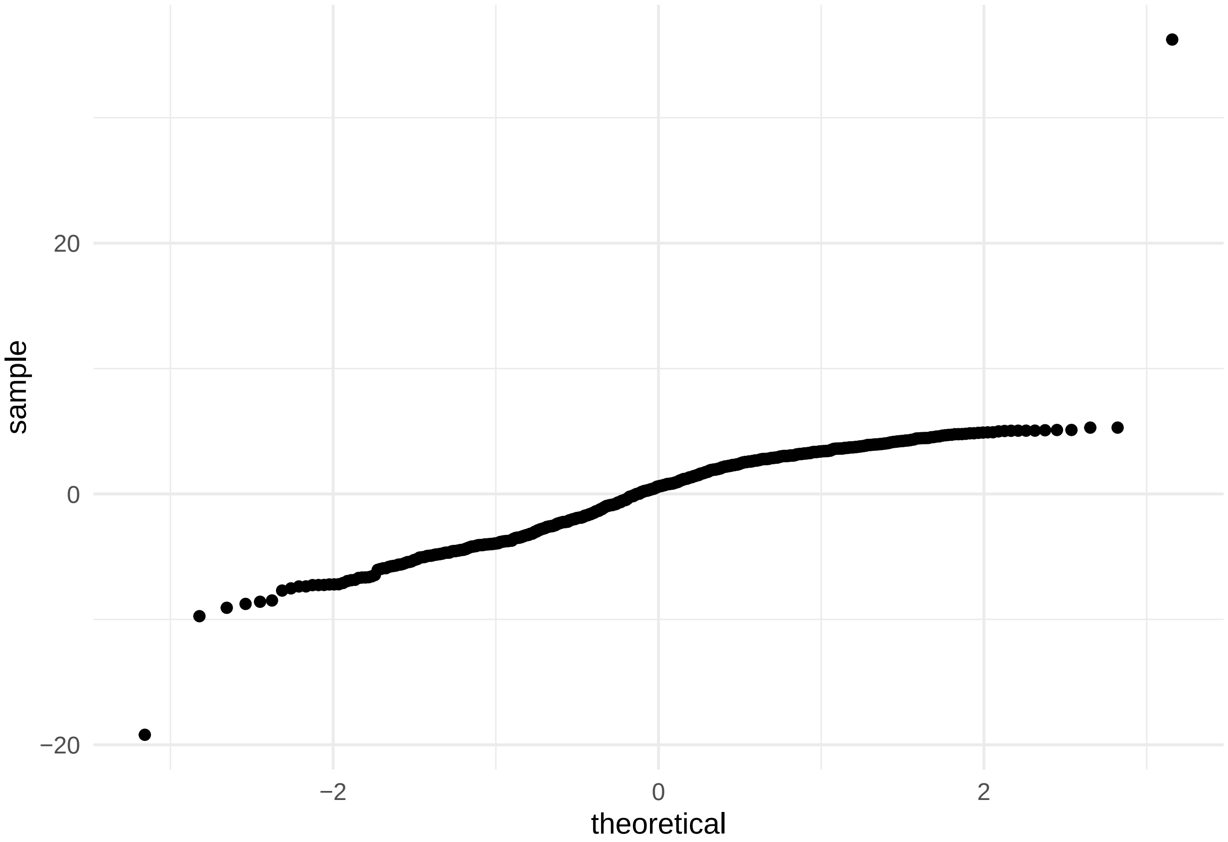
```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.312         0.310  3.70      283. 1.03e-52     1 -1712. 3429. 3443.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Step 2. Compare both fit and bad models on both residual plots, then check the normality of the two residuals by using qq_plot, finally check the residual outlier with olsrr package.

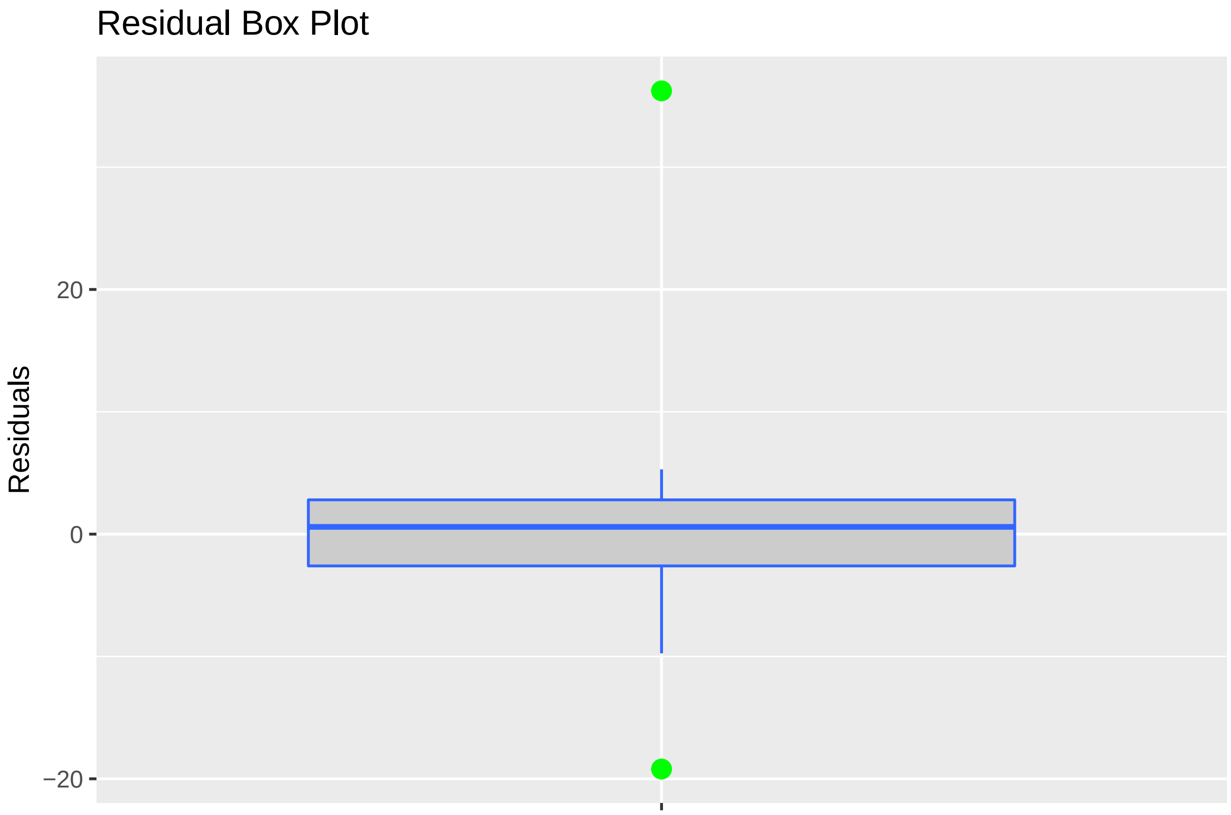
```
library("olsrr")
bad_residual <- LE_data %>%
  add_residuals(fitbad, "resid") %>%
  ggplot(aes(x=log10(life_expectancy_years))) +
  geom_point(aes(y=resid), alpha=0.1) +
  labs(x="life expectancy in years", y="residuals") +
  theme_minimal()
bad_residual
```



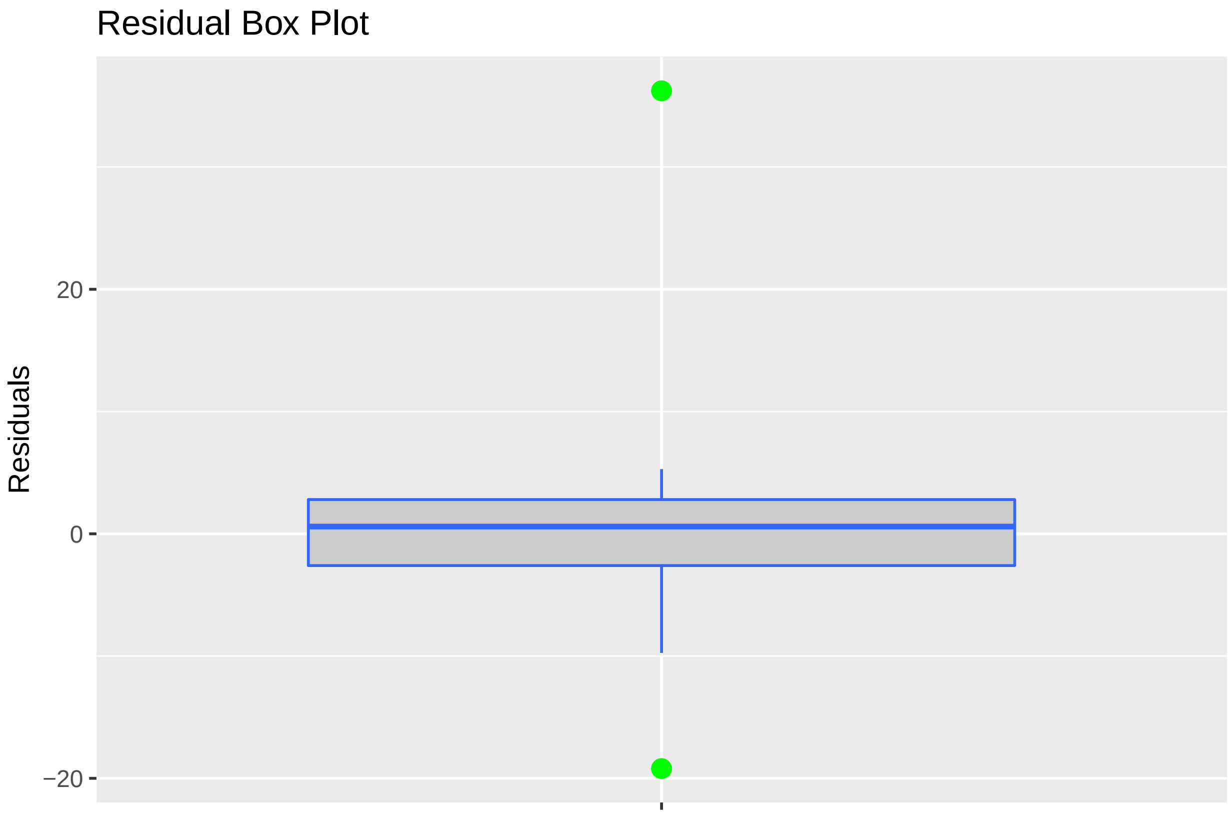
```
bad_residual_qq <- LE_data %>%
  add_residuals(fitbad, "resid") %>%
  ggplot(aes(sample=resid)) +
  geom_qq() +
  theme_minimal()
bad_residual_qq
```



```
bad_residual_box <- ols_plot_resid_box(fitbad)
```

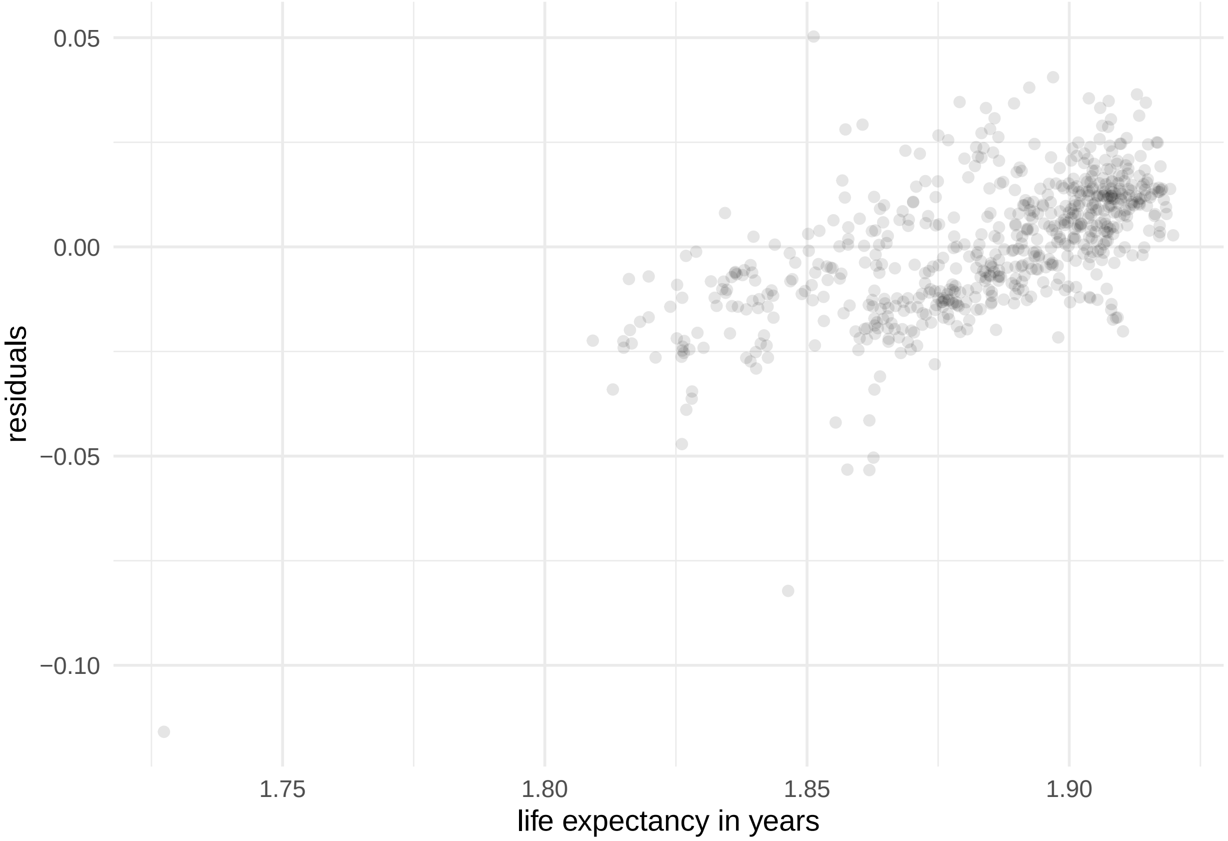


```
bad_residual_box
```



Residual Box Plot

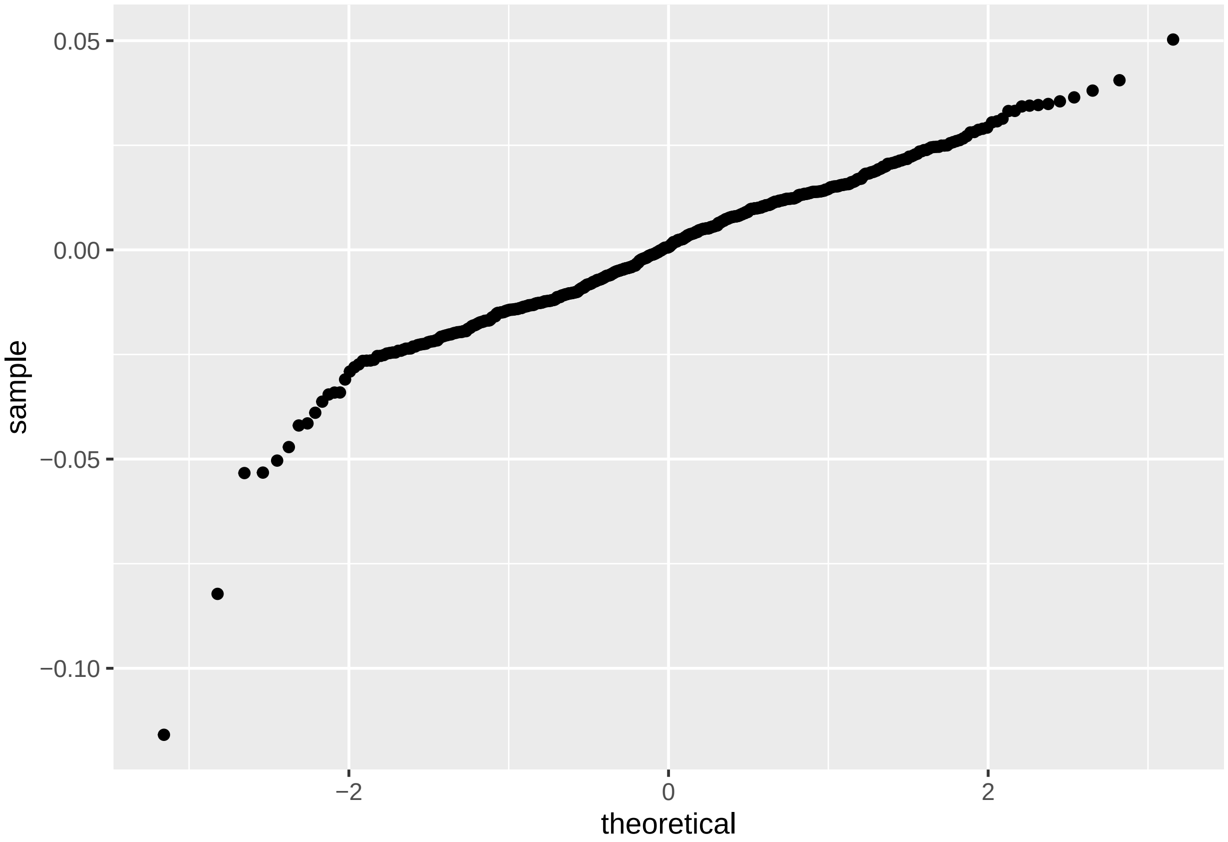
```
better_residual <- LE_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=log10(life_expectancy_years))) +
  geom_point(aes(y=resid), alpha=0.1) +
  labs(x="life expectancy in years", y="residuals") +
  theme_minimal()
better_residual
```



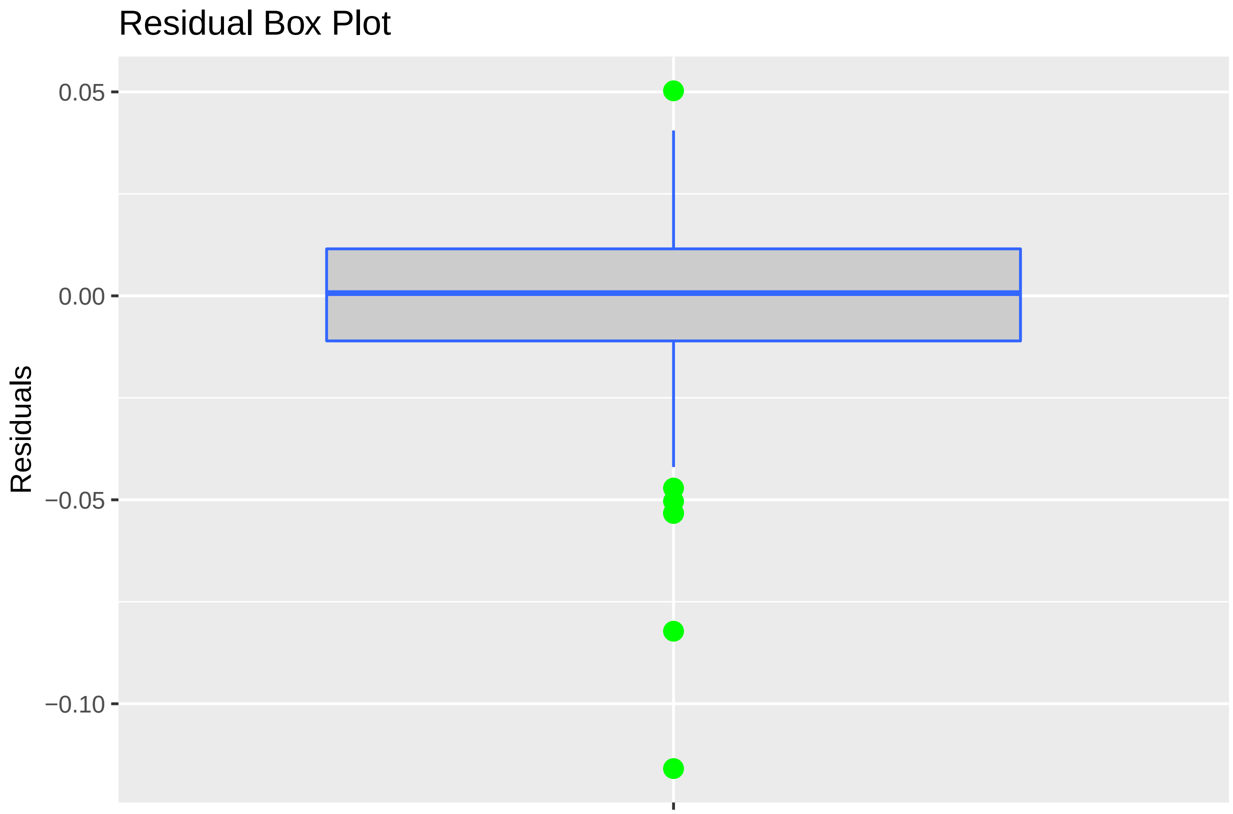
```
better_residual_qq <- LE_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(sample=resid)) +
  geom_qq() + labs(y="sample", x="theoretical")
  theme_minimal()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.border              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

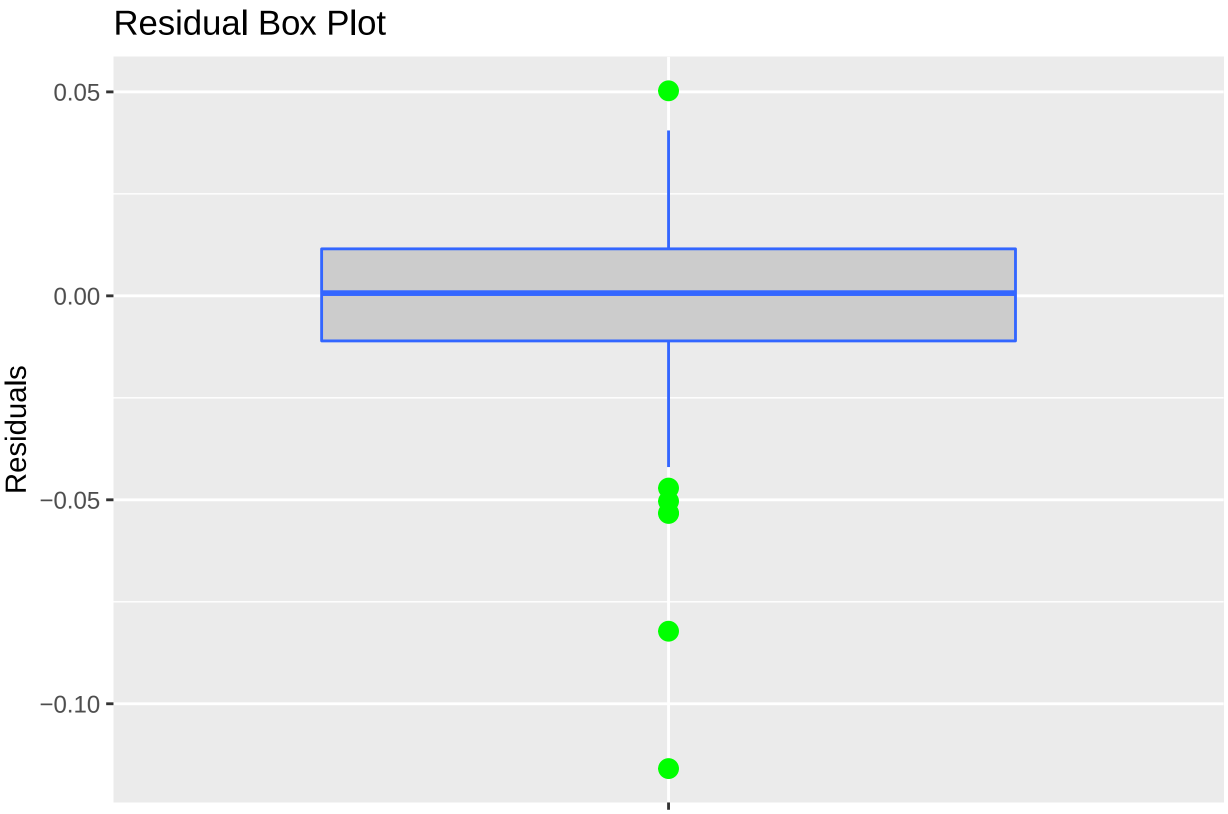
```
better_residual_qq
```



```
better_residual_box <- ols_plot_resid_box(fit1)
```

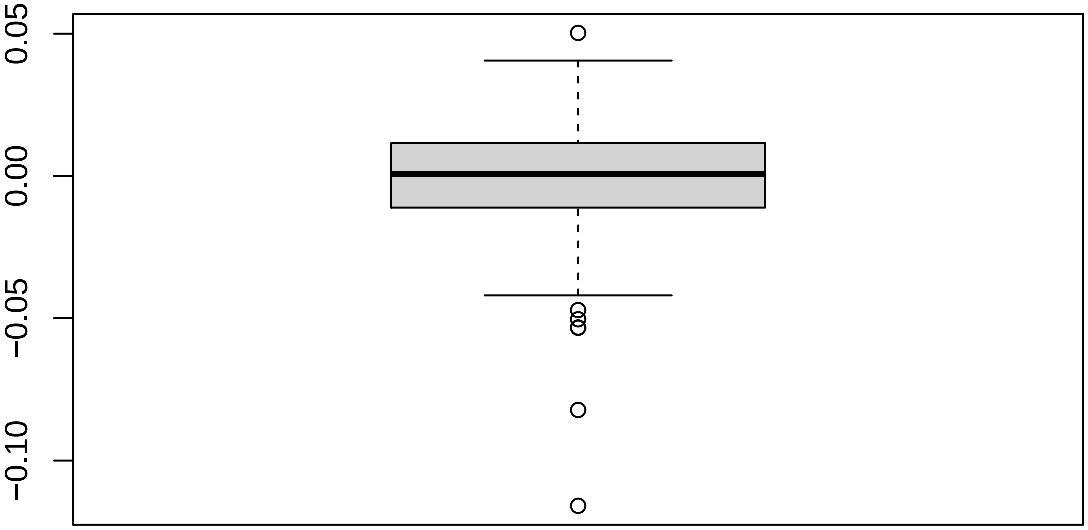


```
better_residual_box
```



Removing outliers. TO BE REMOVED. Not so much of the difference.

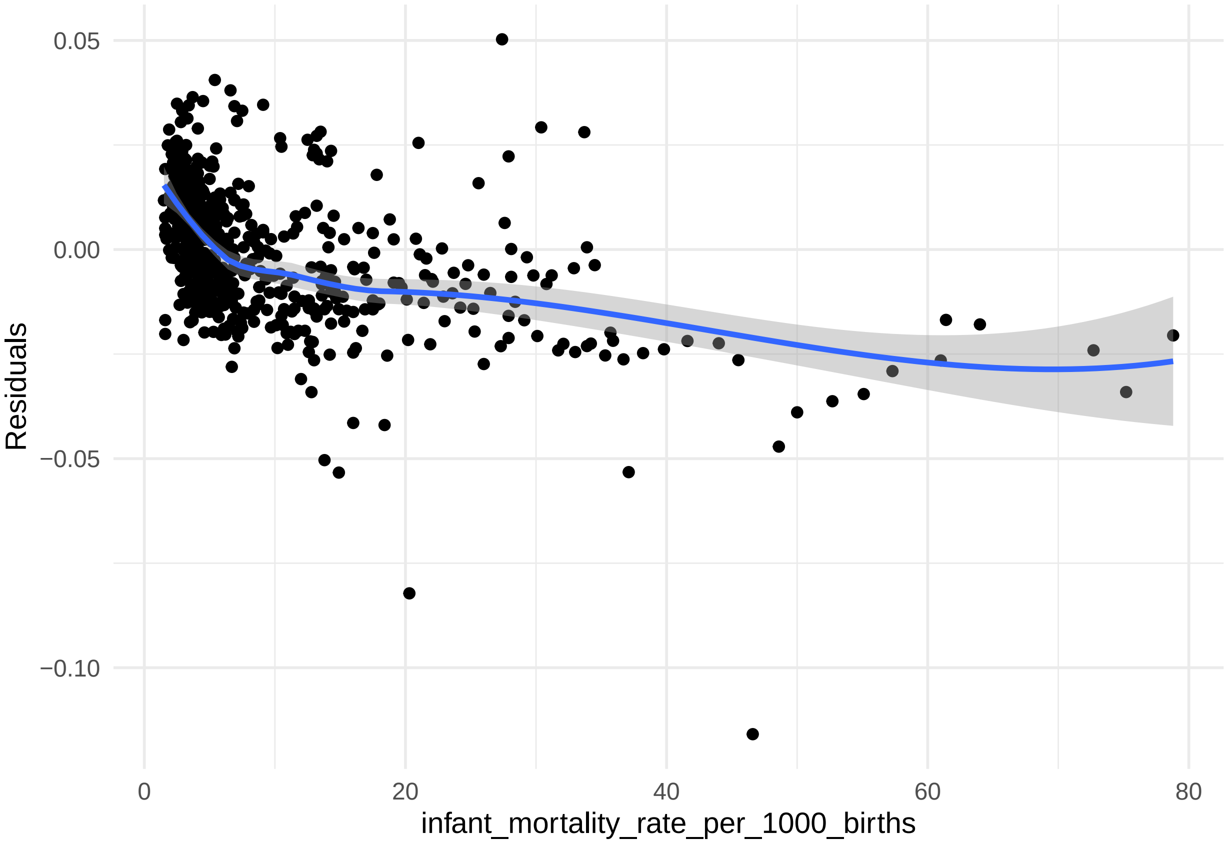
```
better_residual_removedOutlier <- fit1[!fit1 %in% better_residual_box$out]
boxplot(x=better_residual_removedOutlier$residuals,
        y=better_residual_removedOutlier$residuals)
```



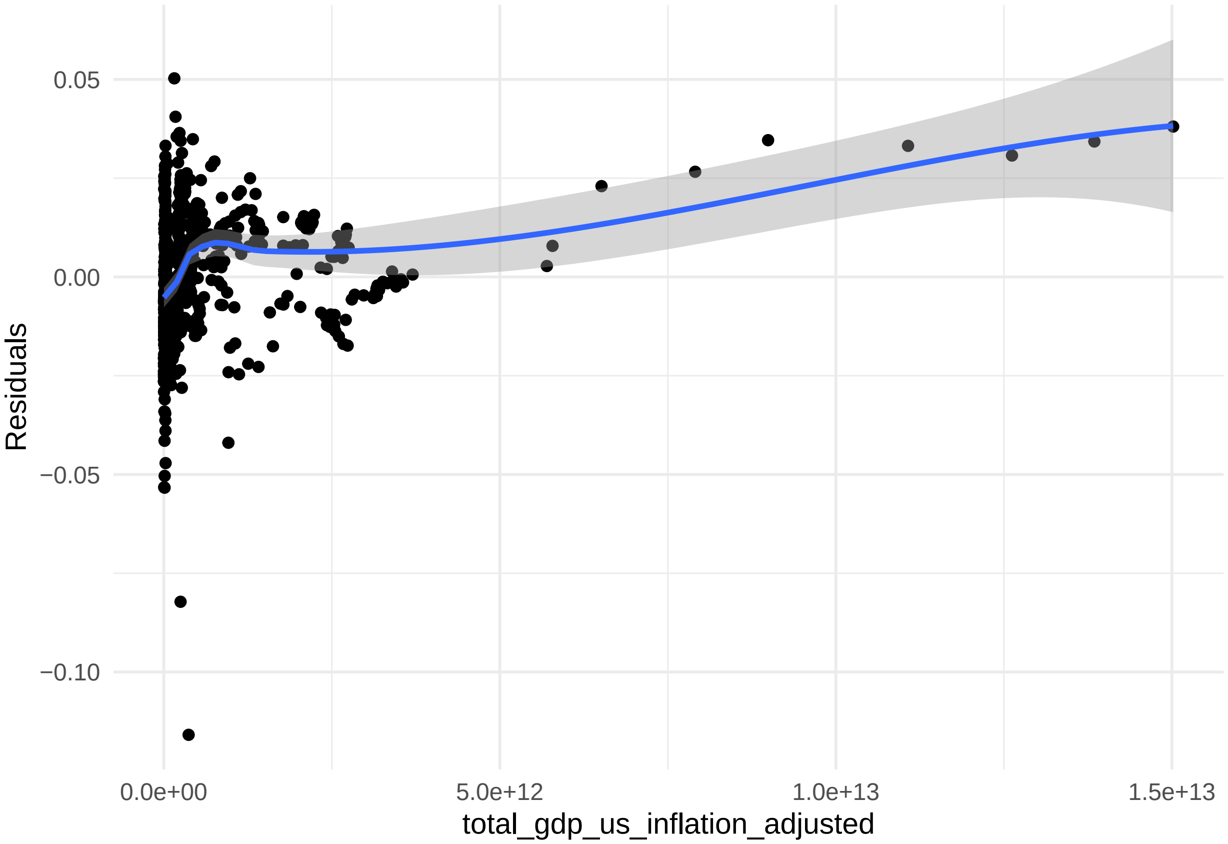
Problem3

Using total_gdp_us_inflation_adjusted as the one additional predictor after comparing the 4 different predictors. The adjusted Rˆ2 increases to 0.6311 from 0.6112.

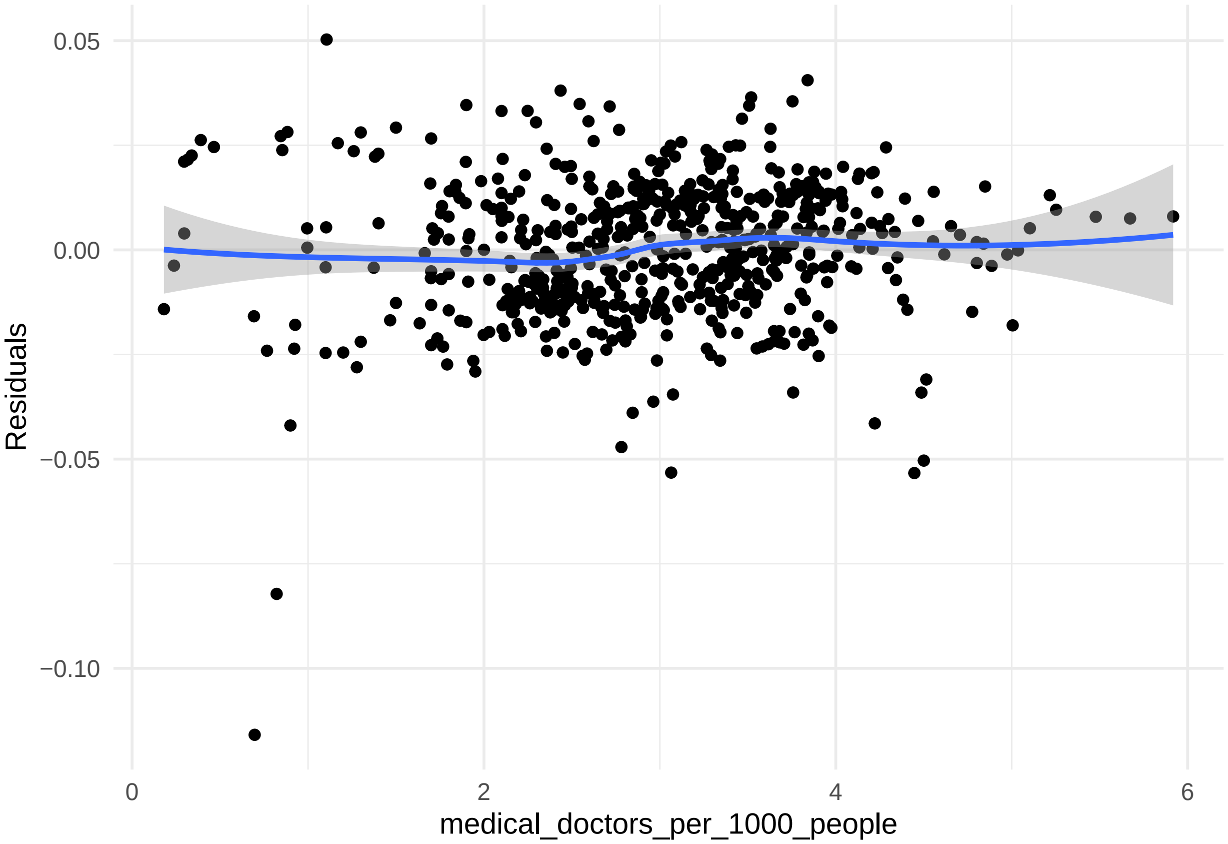
```
pred_infant <- LE_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=infant_mortality_rate_per_1000_births, y=resid)) +
  geom_point() + geom_smooth() +
  labs(x="infant_mortality_rate_per_1000_births", y="Residuals") +
  theme_minimal()
pred_infant
```



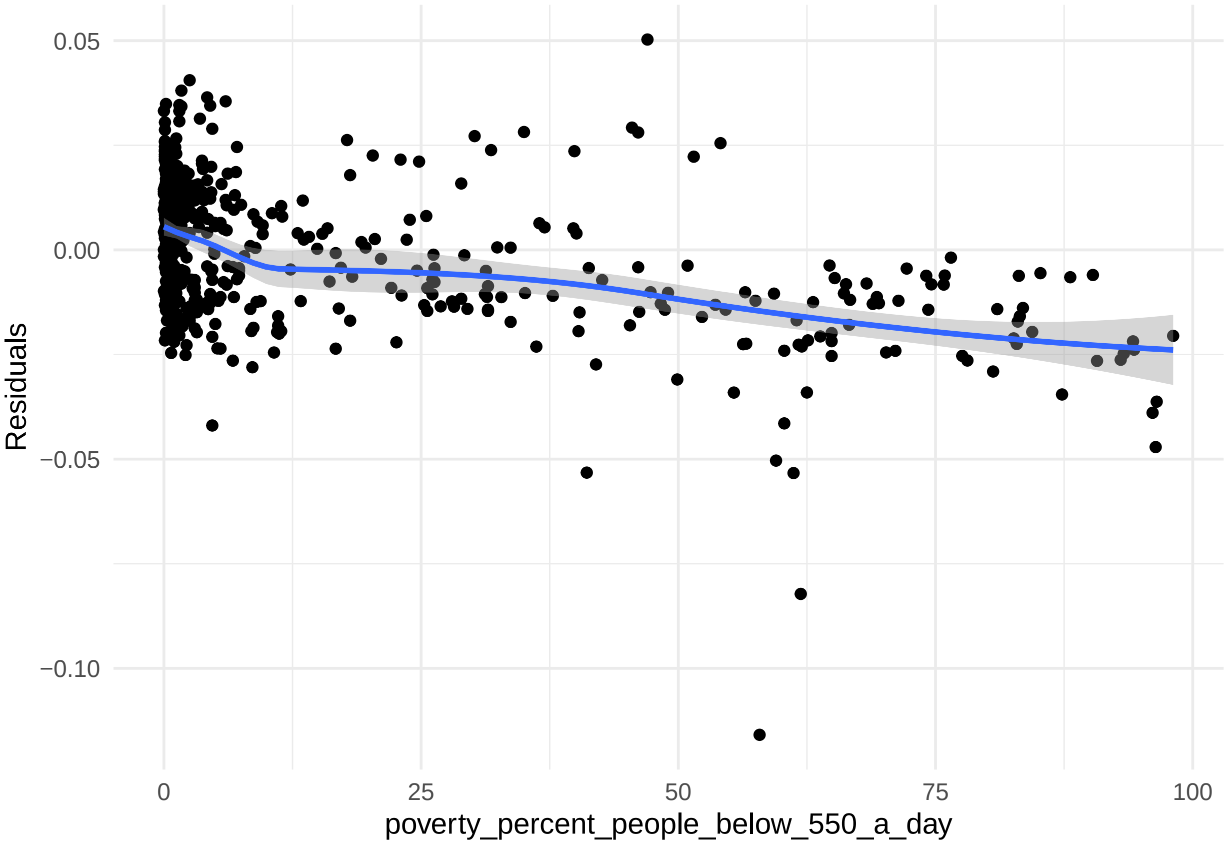
```
pred_gdp <- LE_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=total_gdp_us_inflation_adjusted, y=resid)) +
  geom_point() + geom_smooth() +
  labs(x="total_gdp_us_inflation_adjusted", y="Residuals") +
  theme_minimal()
pred_gdp
```



```
pred_dc <- LE_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=medical_doctors_per_1000_people, y=resid)) +
  geom_point() + geom_smooth() +
  labs(x="medical_doctors_per_1000_people", y="Residuals") +
  theme_minimal()
pred_dc
```



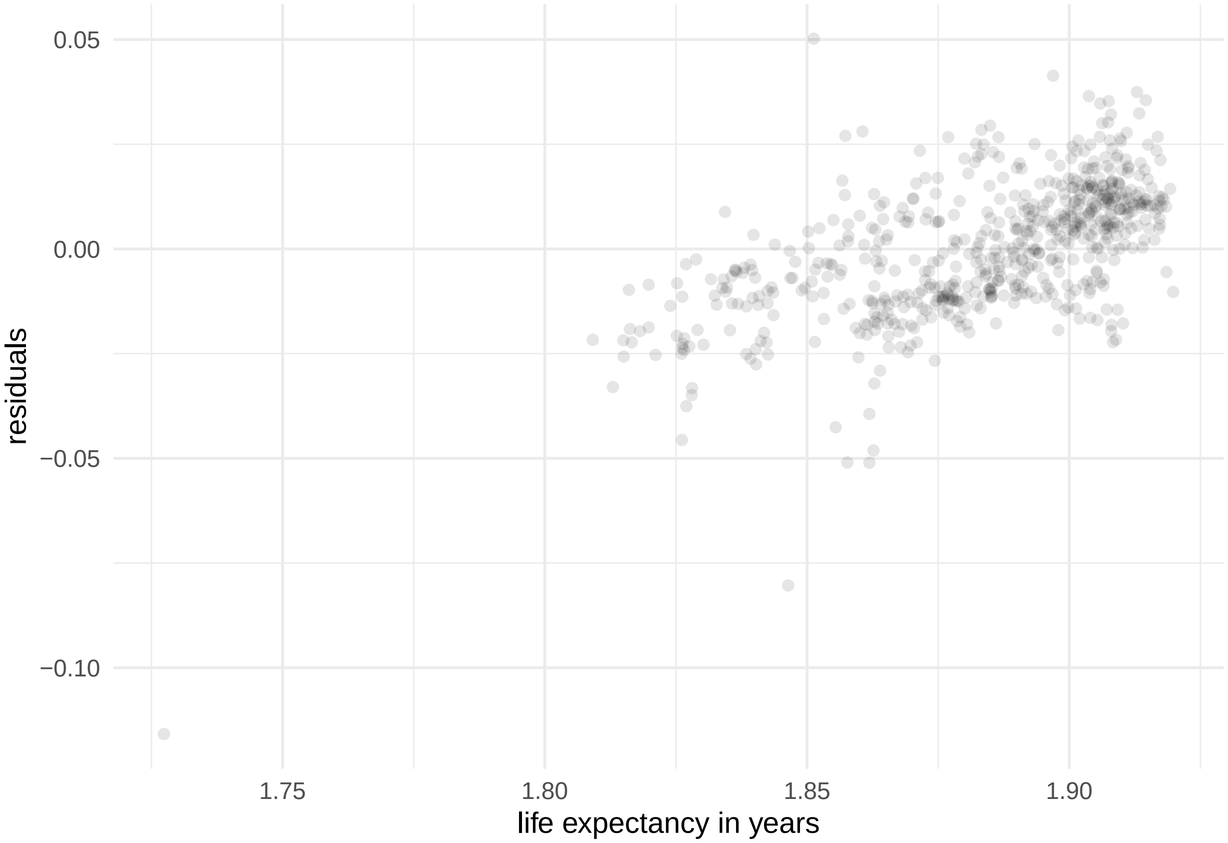
```
pred_poverty <- LE_data %>%
  add_residuals(fit1, "resid") %>%
  ggplot(aes(x=poverty_percent_people_below_550_a_day, y=resid)) +
   geom_point() + geom_smooth() +
  labs(x="poverty_percent_people_below_550_a_day", y="Residuals") +
  theme_minimal()
pred_poverty
```



```r
fit2 <- lm(log10(life_expectancy_years) ~ log10(murder_per_100000_people) +
             total_gdp_us_inflation_adjusted, data=LE_data)
summary(fit2)
```

```
##
## Call:
## lm(formula = log10(life_expectancy_years) ~ log10(murder_per_100000_people) +
##     total_gdp_us_inflation_adjusted, data = LE_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.115818 -0.010728  0.000948  0.010728  0.050190
##
## Coefficients:
##                                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                      1.890e+00  7.567e-04 2497.815  < 2e-16 ***
## log10(murder_per_100000_people) -4.604e-02  1.460e-03  -31.543  < 2e-16 ***
## total_gdp_us_inflation_adjusted  2.699e-15  4.517e-16    5.975 3.86e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01582 on 625 degrees of freedom
## Multiple R-squared:  0.6323, Adjusted R-squared:  0.6311
## F-statistic: 537.4 on 2 and 625 DF,  p-value: < 2.2e-16
```

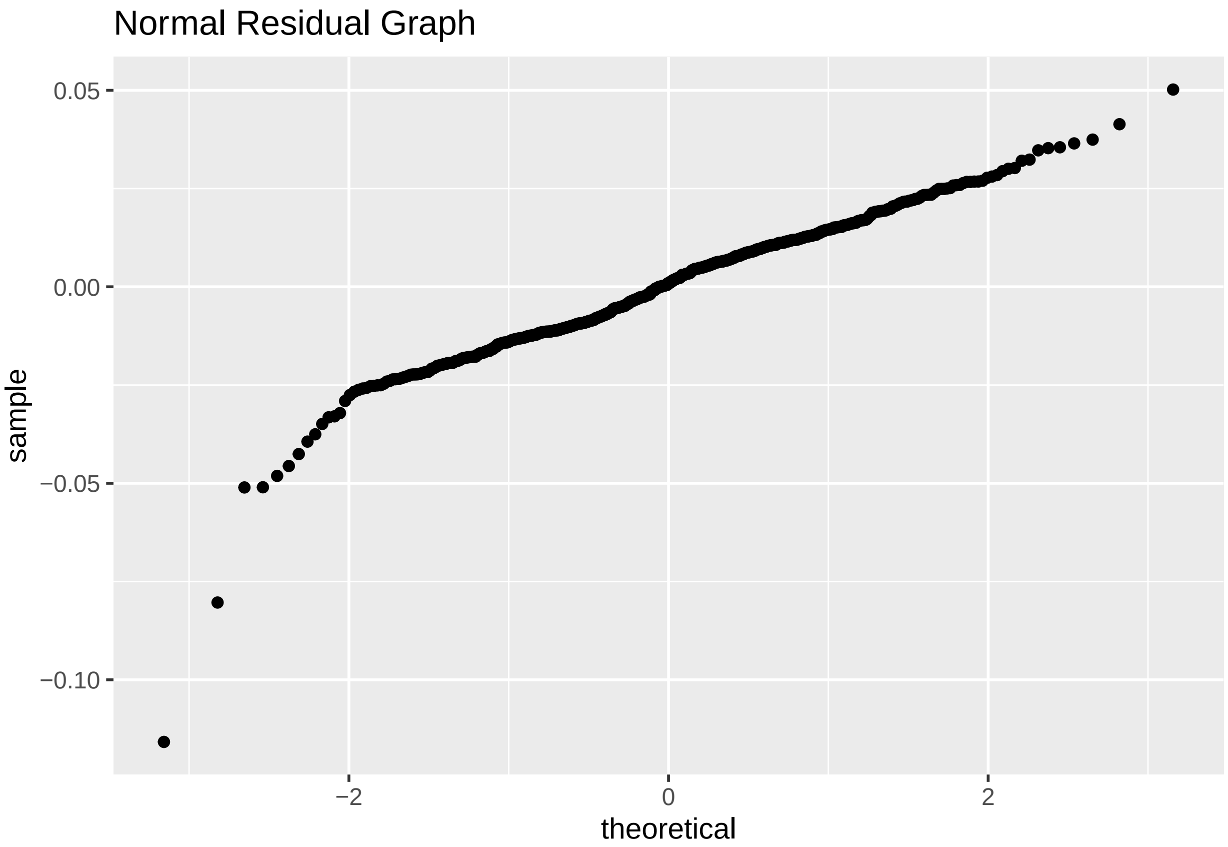
```r
better_residual <- LE_data %>%
  add_residuals(fit2, "resid") %>%
  ggplot(aes(x=log10(life_expectancy_years))) +
  geom_point(aes(y=resid), alpha=0.1) +
  labs(x="life expectancy in years", y="residuals") +
  theme_minimal()
better_residual
```



```
better_residual_qq <- LE_data %>%
  add_residuals(fit2, "resid") %>%
  ggplot(aes(sample=resid)) +
  geom_qq() + labs(y="sample", x="theoretical", title="Normal Residual Graph")
  theme_minimal()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.border              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```
better_residual_qq
```



Problem4

The better model from problem3 has a smaller cv RMSE at 0.01565241 whereas the original model has a larger cv RMSE at 0.0161054. It's not surprising as we know the more variables we add the better model will show a smaller CV RMSE.

```
LE_data_cv <- crossv_kfold(LE_data, 10)
LE_data_cv
```

```
## # A tibble: 10 x 3
##    train        test         .id
##    <named list> <named list> <chr>
##  1 <resample>   <resample>   01
##  2 <resample>   <resample>   02
##  3 <resample>   <resample>   03
##  4 <resample>   <resample>   04
##  5 <resample>   <resample>   05
##  6 <resample>   <resample>   06
##  7 <resample>   <resample>   07
##  8 <resample>   <resample>   08
##  9 <resample>   <resample>   09
## 10 <resample>   <resample>   10
```

```
cv_better_model <- LE_data_cv %>%
  mutate(fit = purrr::map(train,
                          ~lm(log10(life_expectancy_years) ~
                                log10(murder_per_100000_people) +
             total_gdp_us_inflation_adjusted, data=.)),
         rmse=purrr::map2_dbl(fit, test, ~rmse(.x, .y)))
cv_better_model
```

```
## # A tibble: 10 x 5
##    train        test         .id   fit             rmse
##    <named list> <named list> <chr> <named list>  <dbl>
##  1 <resample>   <resample>   01    <lm>         0.0145
##  2 <resample>   <resample>   02    <lm>         0.0157
##  3 <resample>   <resample>   03    <lm>         0.0147
##  4 <resample>   <resample>   04    <lm>         0.0142
##  5 <resample>   <resample>   05    <lm>         0.0209
##  6 <resample>   <resample>   06    <lm>         0.0172
##  7 <resample>   <resample>   07    <lm>         0.0145
##  8 <resample>   <resample>   08    <lm>         0.0135
##  9 <resample>   <resample>   09    <lm>         0.0181
## 10 <resample>   <resample>   10    <lm>         0.0135
```

```
cv_original <- LE_data_cv %>%
  mutate(fit = purrr::map(train,
                          ~lm(log10(life_expectancy_years) ~
                                log10(murder_per_100000_people), data=.)),
         rmse=purrr::map2_dbl(fit, test, ~rmse(.x, .y)))
cv_original
```

```
## # A tibble: 10 x 5
##    train        test         .id   fit             rmse
##    <named list> <named list> <chr> <named list>  <dbl>
##  1 <resample>   <resample>   01    <lm>         0.0158
##  2 <resample>   <resample>   02    <lm>         0.0161
##  3 <resample>   <resample>   03    <lm>         0.0147
##  4 <resample>   <resample>   04    <lm>         0.0141
##  5 <resample>   <resample>   05    <lm>         0.0209
##  6 <resample>   <resample>   06    <lm>         0.0184
##  7 <resample>   <resample>   07    <lm>         0.0149
##  8 <resample>   <resample>   08    <lm>         0.0136
##  9 <resample>   <resample>   09    <lm>         0.0184
## 10 <resample>   <resample>   10    <lm>         0.0144
```

```
mean(cv_better_model$rmse)
```

```
## [1] 0.01567992
```

```
mean(cv_original$rmse)
```

```
## [1] 0.01613559
```

Problem5

Partitioning

```
set.seed(10)
LE_data_part <- resample_partition(LE_data,
                                   p=c(train=0.5,
                                       valid=0.25,
                                       test=0.25))
LE_data_part
```

```
## $train
## <resample [313 x 8]> 11, 12, 19, 22, 24, 25, 26, 28, 29, 30, ...
##
## $valid
## <resample [157 x 8]> 2, 3, 4, 5, 7, 8, 10, 13, 17, 18, ...
##
## $test
## <resample [158 x 8]> 1, 6, 9, 14, 15, 16, 23, 39, 45, 46, ...
```

```
step1 <- function(response, predictors, candidates, partition)
{
  rhs <- paste0(paste0(predictors, collapse="+"), "+", candidates)
  formulas <- lapply(paste0(response, "~", rhs), as.formula)
  rmses <- sapply(formulas,
                  function(fm) rmse(lm(fm, data=partition$train),
                                    data=partition$valid))
  names(rmses) <- candidates
  attr(rmses, "best") <- rmses[which.min(rmses)]
  rmses
}

model <- NULL
```

step1

```
preds <- "1"
cands <- c("log10(infant_mortality_rate_per_1000_births)",
           "log10(murder_per_100000_people)",
           "total_gdp_us_inflation_adjusted",
           "log10(medical_doctors_per_1000_people)",
           "poverty_percent_people_below_550_a_day")
s1 <- step1("log10(life_expectancy_years)", preds, cands, LE_data_part)

model <- c(model, attr(s1, "best"))
s1
```

```
## log10(infant_mortality_rate_per_1000_births)
##                                   0.01398428
##              log10(murder_per_100000_people)
##                                   0.01991172
##              total_gdp_us_inflation_adjusted
##                                   0.02735699
##       log10(medical_doctors_per_1000_people)
##                                   0.02669927
##       poverty_percent_people_below_550_a_day
##                                   0.01903142
## attr(,"best")
## log10(infant_mortality_rate_per_1000_births)
##                                   0.01398428
```

Step2 adding log10(infant_mortality_rate_per_1000_births)

```
preds <- "log10(infant_mortality_rate_per_1000_births)"
cands <- c("log10(murder_per_100000_people)",
           "total_gdp_us_inflation_adjusted",
           "log10(medical_doctors_per_1000_people)",
           "poverty_percent_people_below_550_a_day")
```

```
s2 <- step1("log10(life_expectancy_years)", preds, cands, LE_data_part)

model <- c(model, attr(s2, "best"))
s2
```

```
##        log10(murder_per_100000_people)        total_gdp_us_inflation_adjusted
##                             0.01323456                             0.01373380
## log10(medical_doctors_per_1000_people) poverty_percent_people_below_550_a_day
##                             0.01440880                             0.01402275
## attr(,"best")
## log10(murder_per_100000_people)
##                      0.01323456
```

Step 3 Adding "log10(murder__per__100000__people)"

```
preds <- c("log10(infant_mortality_rate_per_1000_births)",
"log10(murder_per_100000_people)")
cands <- c("total_gdp_us_inflation_adjusted",
           "log10(medical_doctors_per_1000_people)",
           "poverty_percent_people_below_550_a_day")
s3 <- step1("log10(life_expectancy_years)", preds, cands, LE_data_part)

model <- c(model, attr(s3, "best"))
s3
```

```
##        total_gdp_us_inflation_adjusted log10(medical_doctors_per_1000_people)
##                             0.01298784                             0.01365805
## poverty_percent_people_below_550_a_day
##                             0.01312733
## attr(,"best")
## total_gdp_us_inflation_adjusted
##                      0.01298784
```

Step 4 Adding total__gdp__us__inflation__adjusted

```
preds <- c("log10(infant_mortality_rate_per_1000_births)",
"log10(murder_per_100000_people)", "total_gdp_us_inflation_adjusted")
cands <- c("log10(medical_doctors_per_1000_people)",
           "poverty_percent_people_below_550_a_day")
s4 <- step1("log10(life_expectancy_years)", preds, cands, LE_data_part)

model <- c(model, attr(s4, "best"))
s4
```

```
## log10(medical_doctors_per_1000_people) poverty_percent_people_below_550_a_day
##                             0.01344301                             0.01294285
## attr(,"best")
## poverty_percent_people_below_550_a_day
##                             0.01294285
```

Step 5 Adding poverty__percent__people__below__550__a__day. Then stop at step 5 due to the increasing RMSE value of log(medical__doctors__per__1000__people).

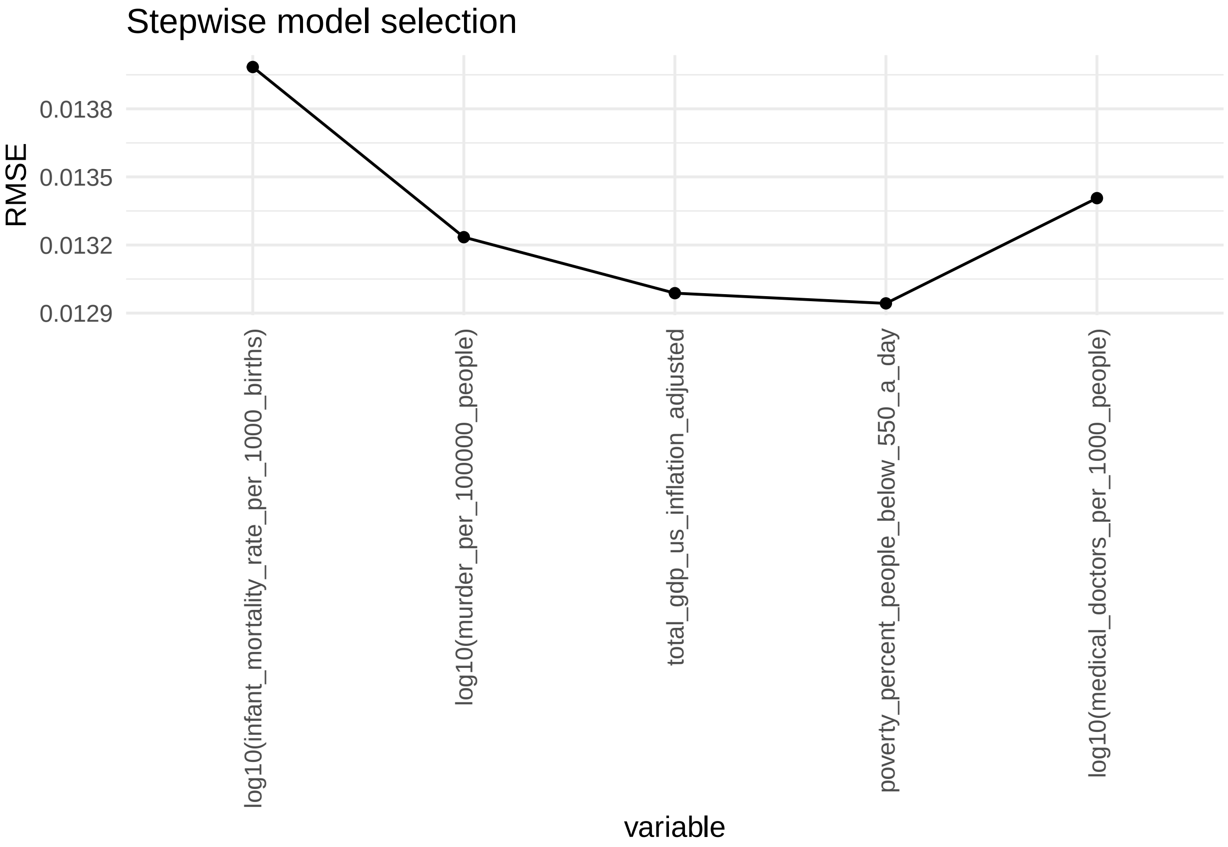
```
preds <- c("log10(infant_mortality_rate_per_1000_births)",
"log10(murder_per_100000_people)", "total_gdp_us_inflation_adjusted",
"poverty_percent_people_below_550_a_day")
cands <- c("log10(medical_doctors_per_1000_people)")
s5 <- step1("log10(life_expectancy_years)", preds, cands, LE_data_part)
```

```
model <- c(model, attr(s5, "best"))
s5
```

```
## log10(medical_doctors_per_1000_people)
##                             0.01340655
## attr(,"best")
## log10(medical_doctors_per_1000_people)
##                             0.01340655
```

Plotting

```
step_model <- tibble(index=seq_along(model),
                     variable=factor(names(model), levels=names(model)),
                     RMSE=model)

ggplot(step_model, aes(y=RMSE)) +
  geom_point(aes(x=variable)) +
  geom_line(aes(x=index)) +
  labs(title="Stepwise model selection") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=90, vjust=0.5,  hjust=1))
```


Stepwise model selection
RMSE
log10(infant_mortality_rate_per_1000_births)
log10(murder_per_100000_people)
total_gdp_us_inflation_adjusted
poverty_percent_people_below_550_a_day
log10(medical_doctors_per_1000_people)
variable