COVID-19 fatality rate

Sumiya Ganbaatar

10/8/2021

```
#Importing the GDP data
df_gdp <- read.csv("/Users/bayarmaaorsoo/Desktop/Projects/COVID19 - Fatality Rate/gdp-per-capita-worldba
#Choosing the year 2020 because that's the latest data
df_gdp1 <- df_gdp %>% filter(Year == 2020)
#Dropping columns "Code" and "Year" as we won't use them.
df_gdp1 <- subset(df_gdp1, select = -c(Code, Year))
#Changing column names
df_gdp1 <- df_gdp1 %>% rename(Country = Entity, GDP_per_capita = GDP.per.capita..PPP..constant.2017.int
str(df_gdp1)
```

```
## 'data.frame':    224 obs. of  2 variables:
##  $ Country       : chr  "Afghanistan" "Africa Eastern and Southern" "Africa Western and Central" "Al
##  $ GDP_per_capita: num  1979 3388 4003 13295 10682 ...
```

```
data_covid <- read_csv("/Users/bayarmaaorsoo/Desktop/Projects/COVID19 - Fatality Rate/COVID data by JHU
```

```
## Rows: 137396 Columns: 67
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): iso_code, continent, location, tests_units
## dbl  (62): total_cases, new_cases, new_cases_smoothed, total_deaths, new_dea...
## date  (1): date
```

```
##
## 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.
```

```
#Selecting necessary data
data_covid <- data_covid[data_covid$date == '2021-12-02',]
variables <- c('location', 'total_cases', 'total_deaths')
data_covid <- data_covid[variables]
# Creating column fatality rate
data_covid<- data_covid %>% mutate(Fatality_rate = (total_deaths/total_cases)*100)
#Changing a column name
colnames(data_covid)[1] <- "Country"
```

```r
#Joining two datatables
data <- inner_join(x= df_gdp1, y = data_covid, by = "Country")

#Dropping country Vanuatu because there are only 6 cases and 1 deaths
data <- subset(data, Country != 'Vanuatu')
```

```r
#Importing government effectiveness data
data_gov <- read.csv("/Users/bayarmaaorsoo/Desktop/Projects/COVID19 - Fatality Rate/Goverment effectiver
#Selecting only country name and score
data_gov <- subset(data_gov, select = c(Country.Name, X2020..YR2020.))
#Changing column names
colnames(data_gov) <- c("Country","Gov_index")
#Inner joining with the main data
df <- inner_join(x = data, y = data_gov, by = 'Country')
#Converting data type chr to double
df$Gov_index <- as.double(df$Gov_index)
```

```r
#Importing age data
data_age80 <- read_csv("/Users/bayarmaaorsoo/Desktop/Projects/COVID19 - Fatality Rate/Age_over_80.csv")
```

```
## Rows: 184 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country
## dbl (3): Rank, Value, Year
```

```
##
## 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.
```

```r
#Selecting necessary data
data_age80 <- subset(data_age80, select = c(Country, Value))
#Changing column names
colnames(data_age80) <- c("Country","age80")
#Joining df dataset
df <- inner_join(x= data_age80, y = df, by = "Country")
```

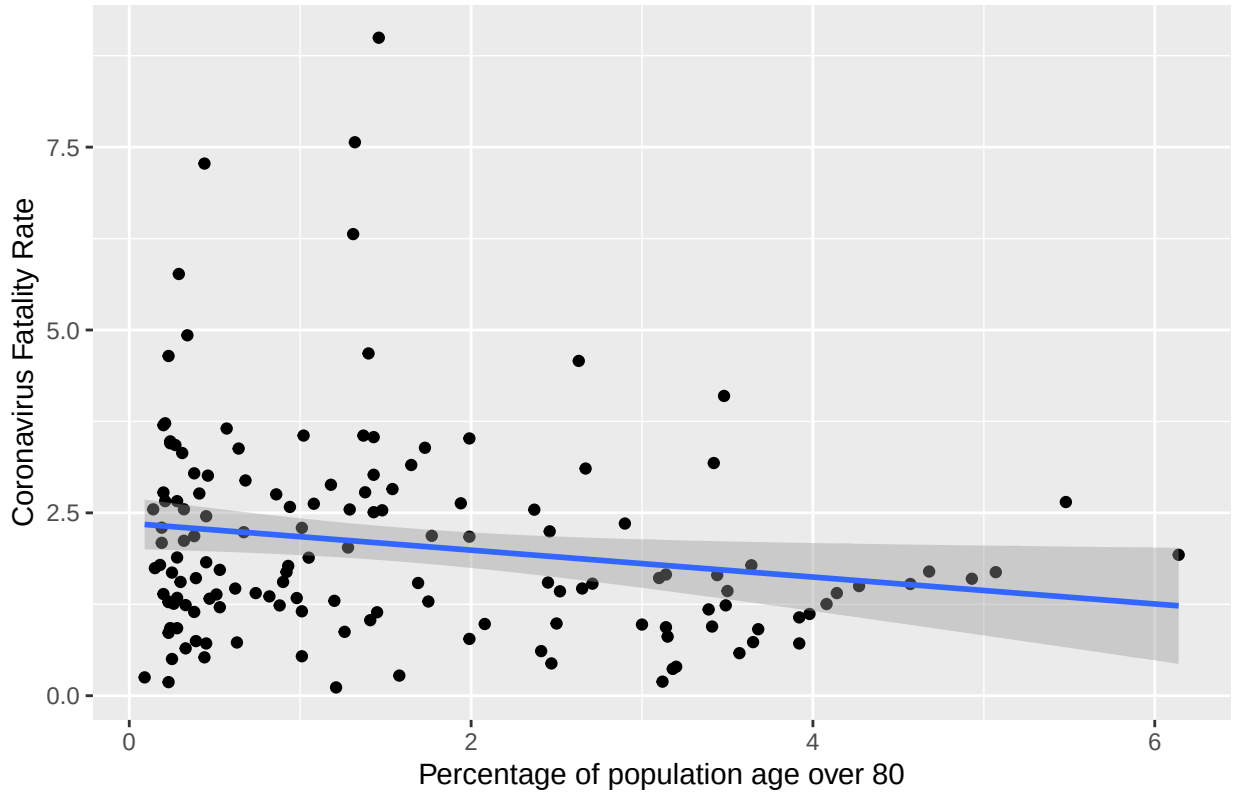
```
# Exploratory Data Analysis
```

```r
ggplot(aes(x = age80, y = (Fatality_rate)), data = df) +geom_point() + geom_smooth(method='lm') + labs(t
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

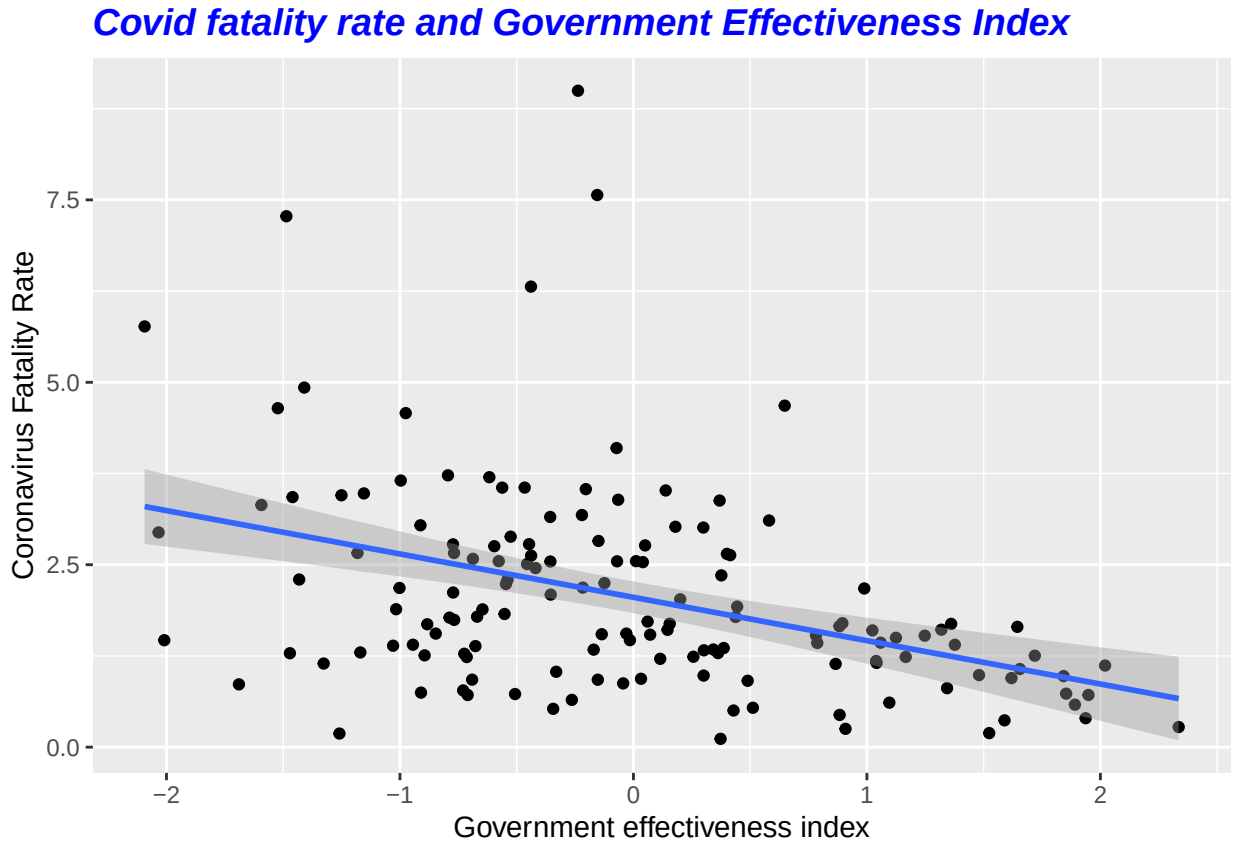



```
ggplot(aes(x = Gov_index, y = (Fatality_rate)), data = df) +geom_point() + geom_smooth(method='lm') + la
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

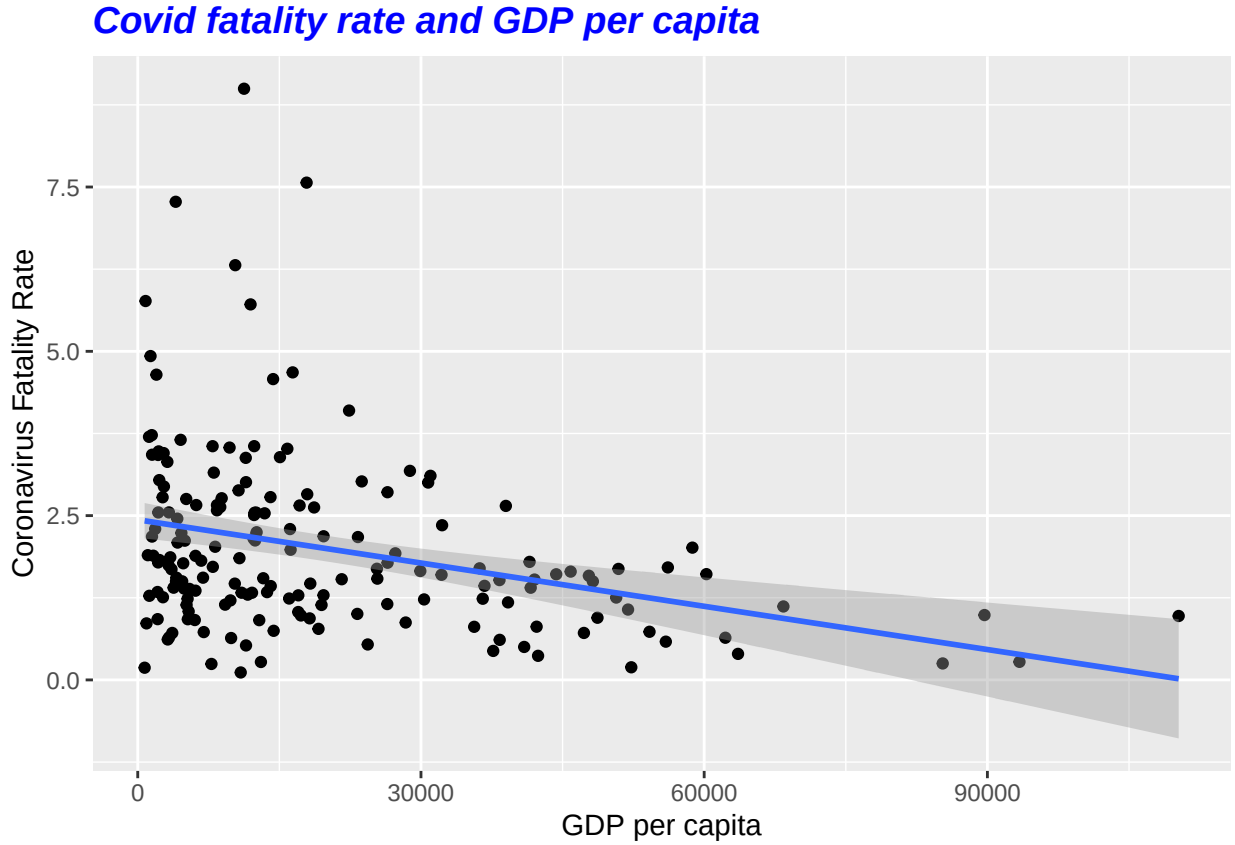



```
ggplot(aes(x = GDP_per_capita, y = (Fatality_rate)), data = data) +geom_point() + geom_smooth(method='lm
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```



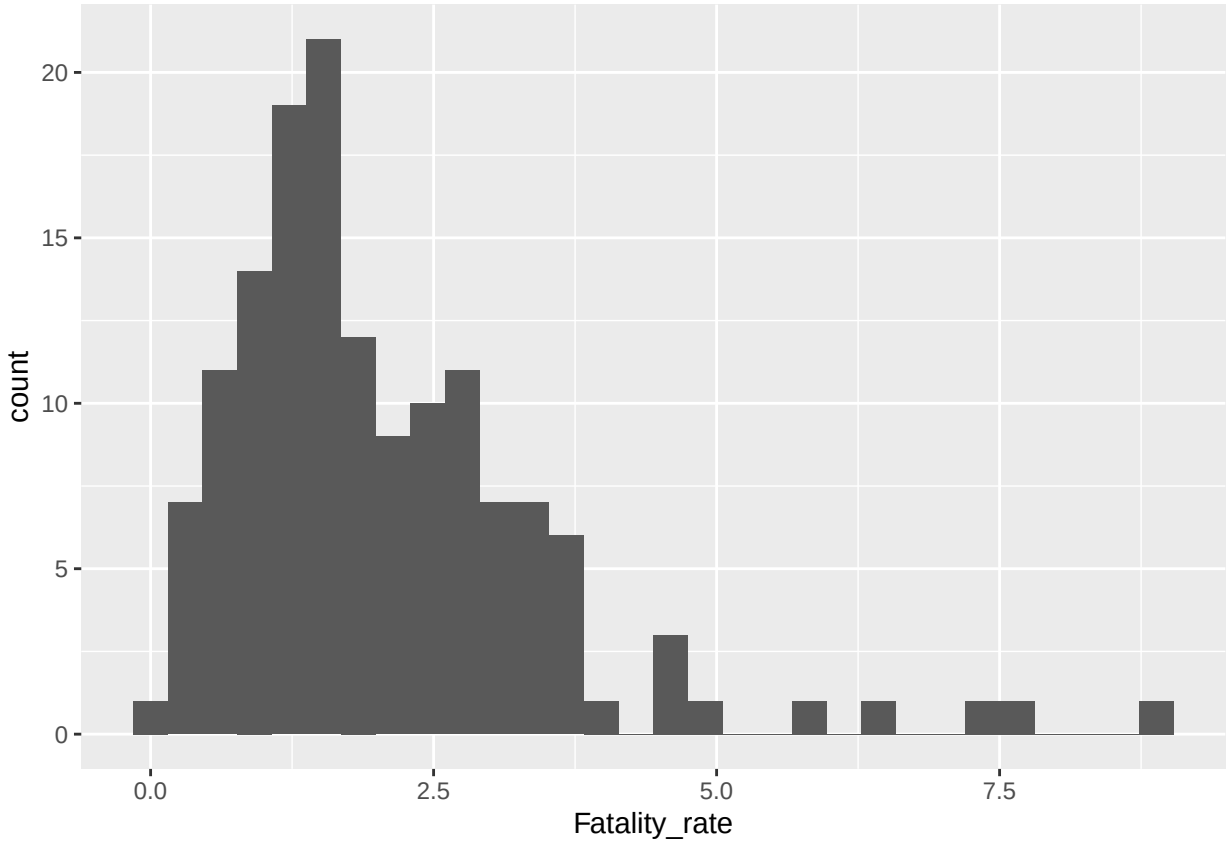

```
#Histogram
summary(df$Fatality_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.1136  1.1543  1.6837  2.0673  2.6607  8.9957       3
```

```
ggplot(df, aes(x = Fatality_rate))  + geom_histogram()
```

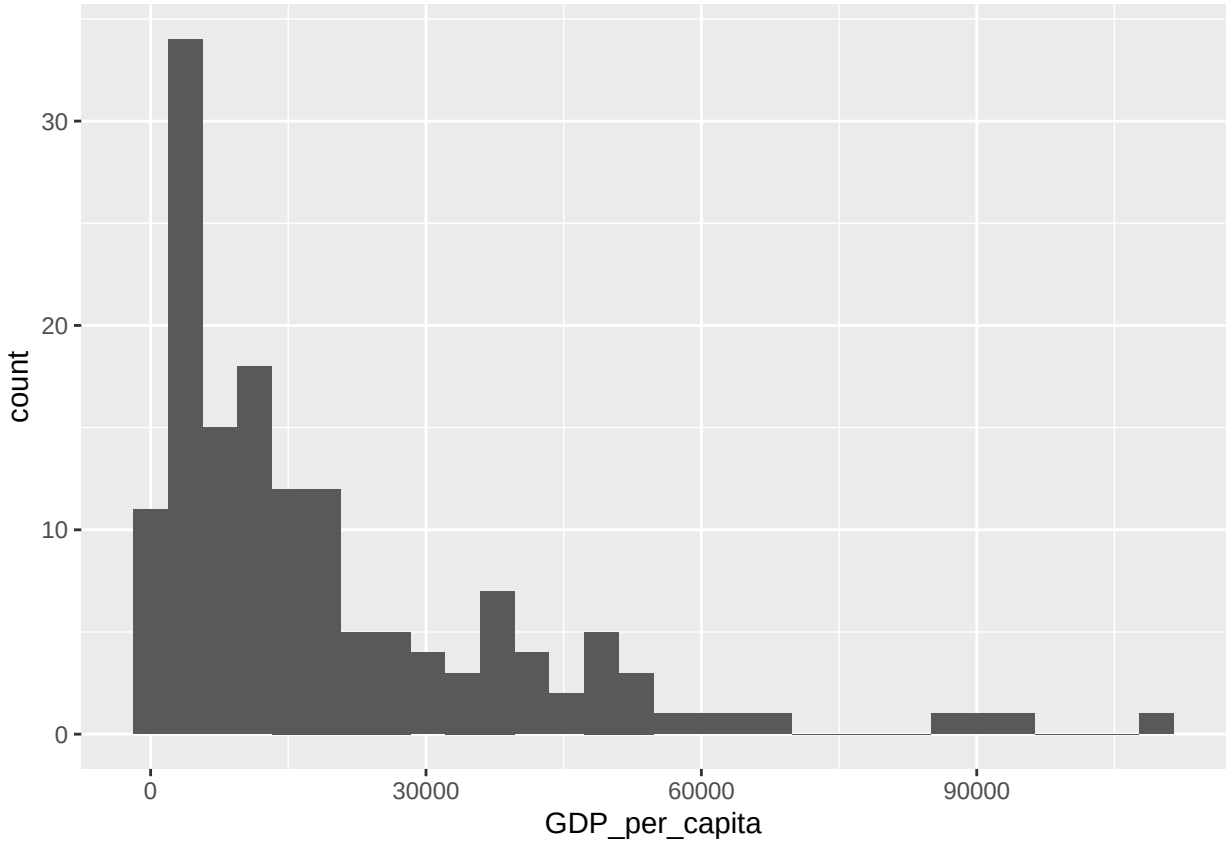
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



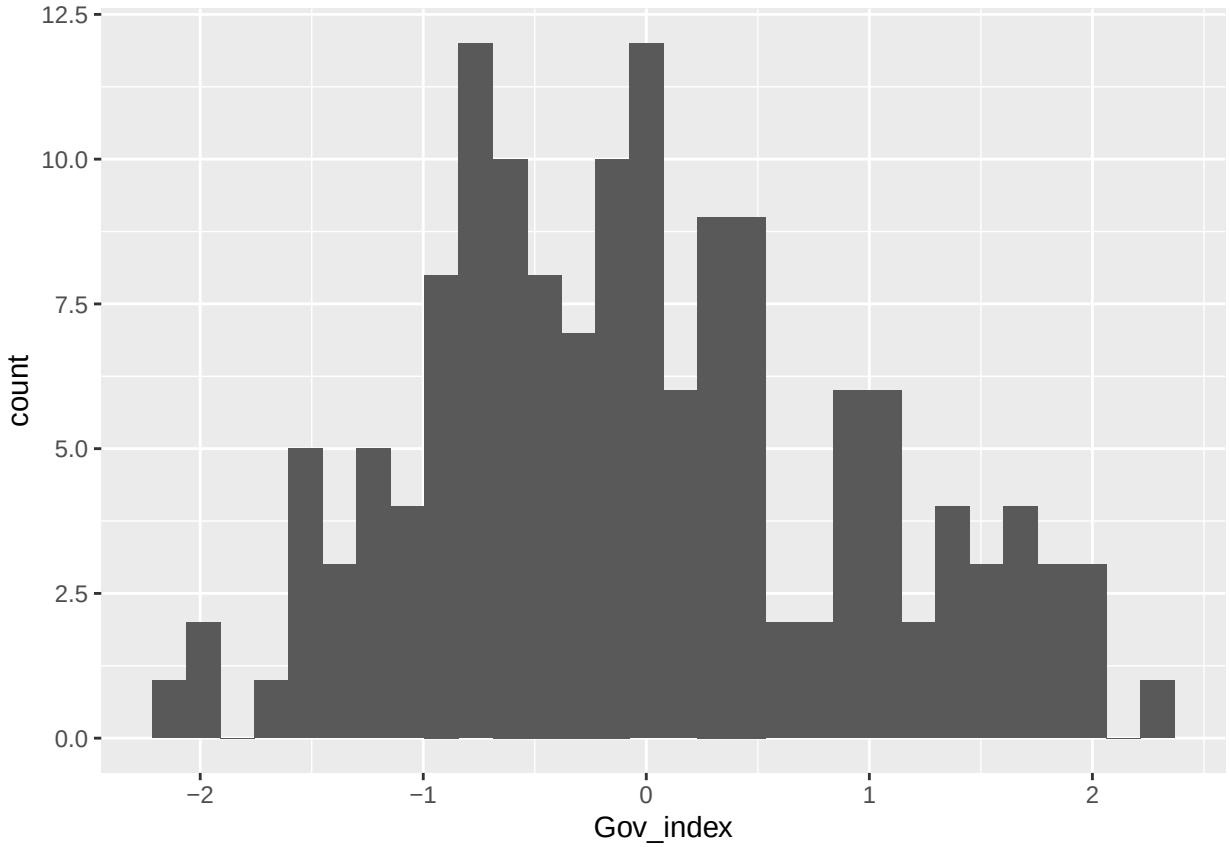

```
ggplot(df, aes(x = GDP_per_capita)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



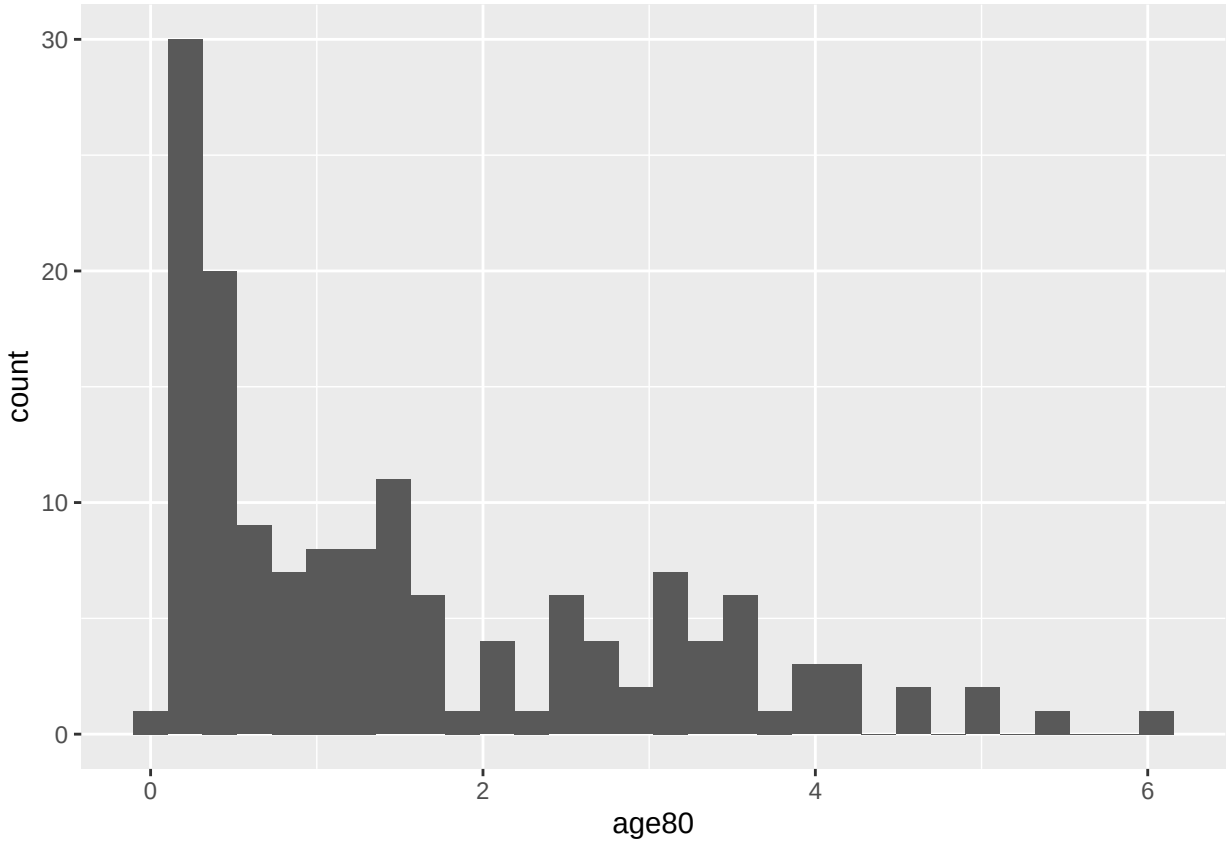

```
ggplot(df, aes(x = Gov_index)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(df, aes(x = age80)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

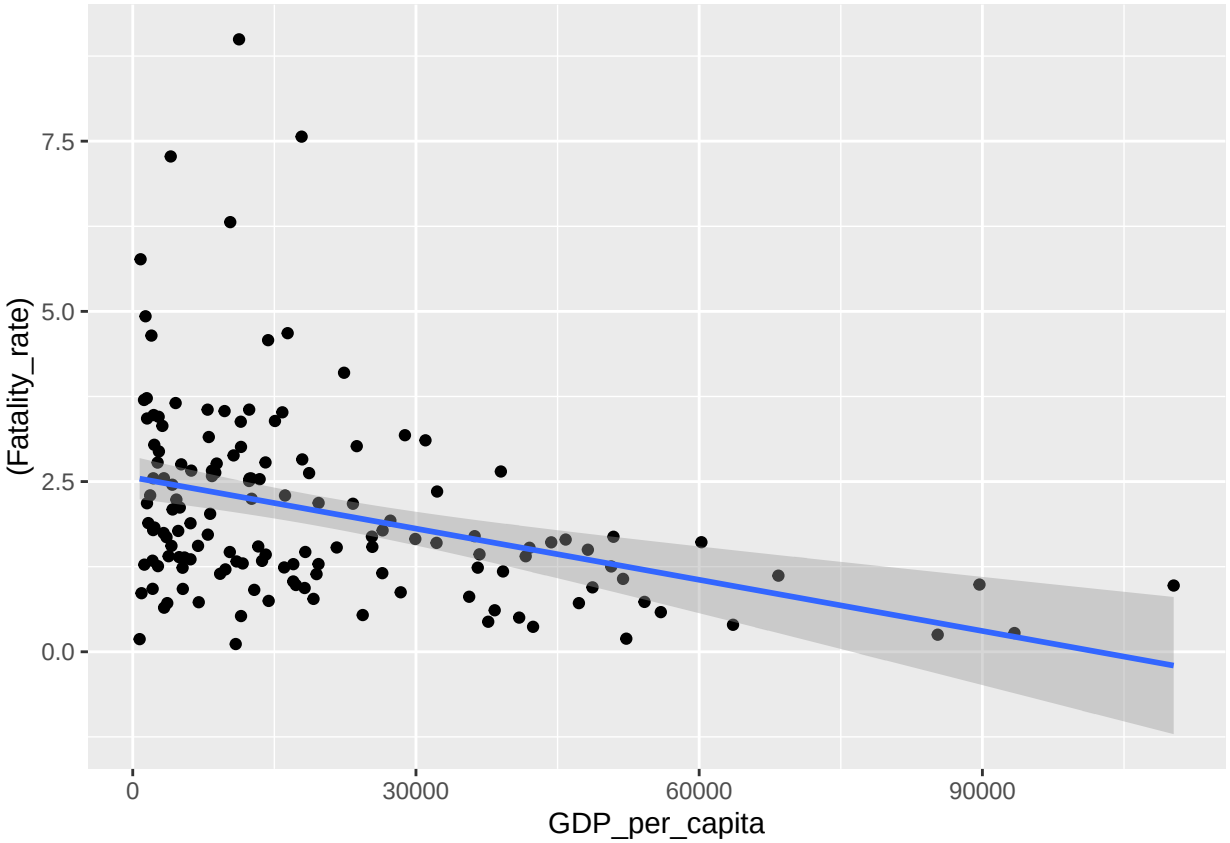



```
ggplot(aes(x = GDP_per_capita, y = (Fatality_rate)), data = df) +geom_point() + geom_smooth(method='lm'
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

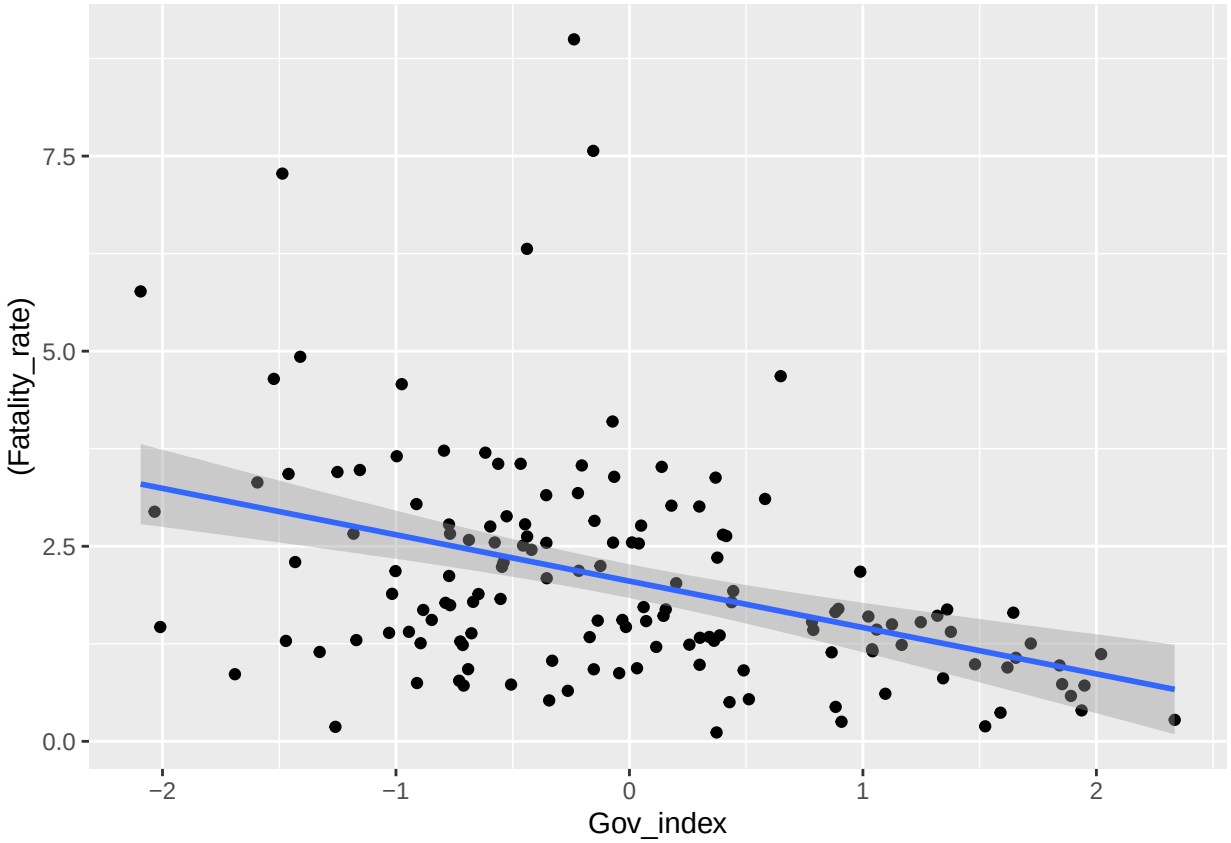



```
ggplot(aes(x = Gov_index, y = (Fatality_rate)), data = df) +geom_point() + geom_smooth(method='lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

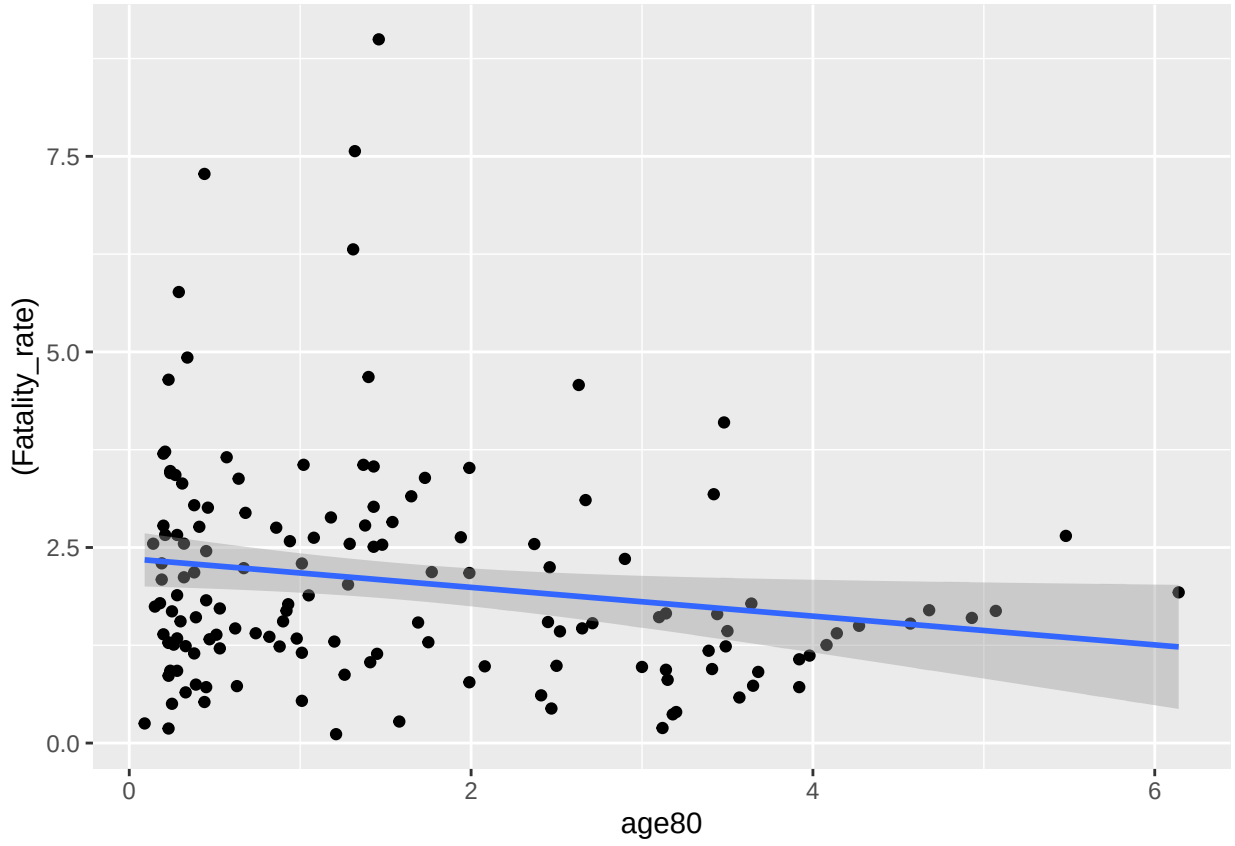



```
ggplot(aes(x = age80, y = (Fatality_rate)), data = df) +geom_point() + geom_smooth(method='lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

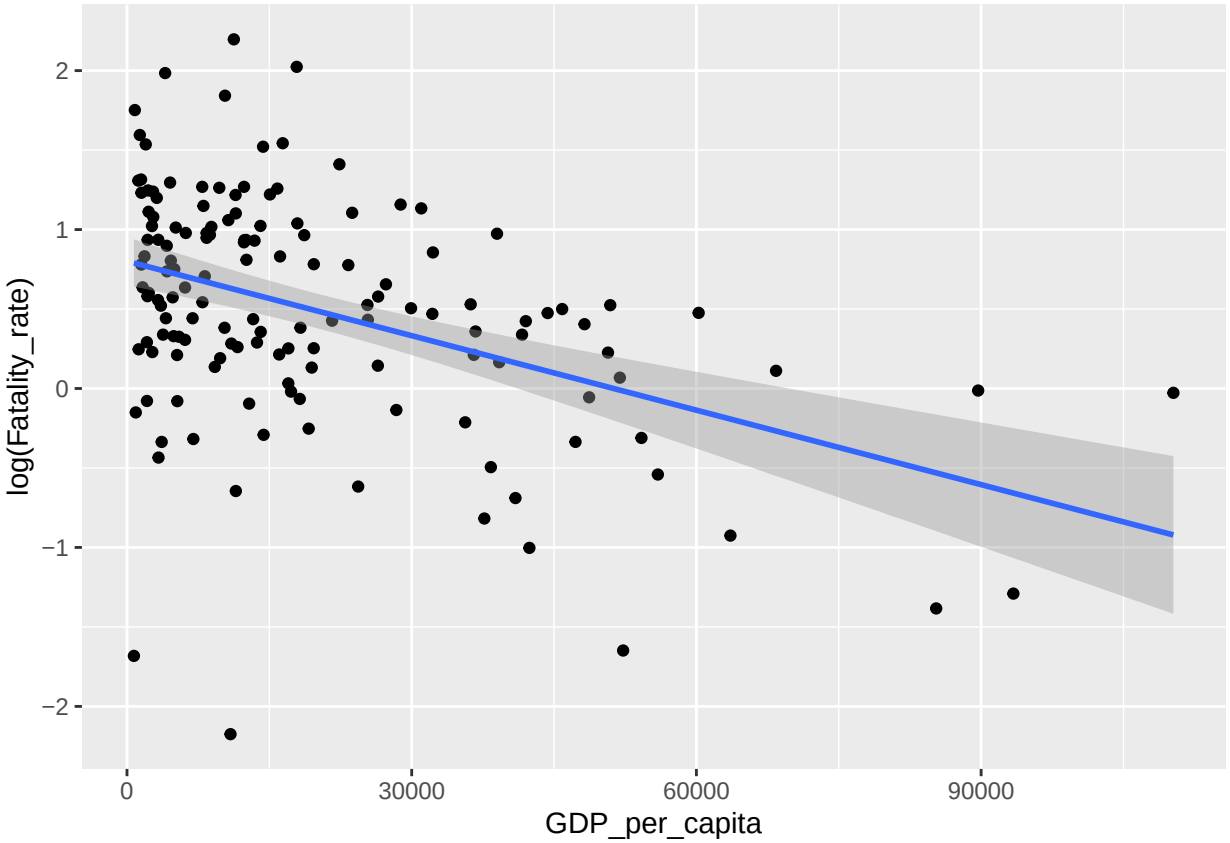



```
#Scatterplots with logs
ggplot(aes(x = GDP_per_capita, y = log(Fatality_rate)), data = df) +geom_point() + geom_smooth(method='l
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

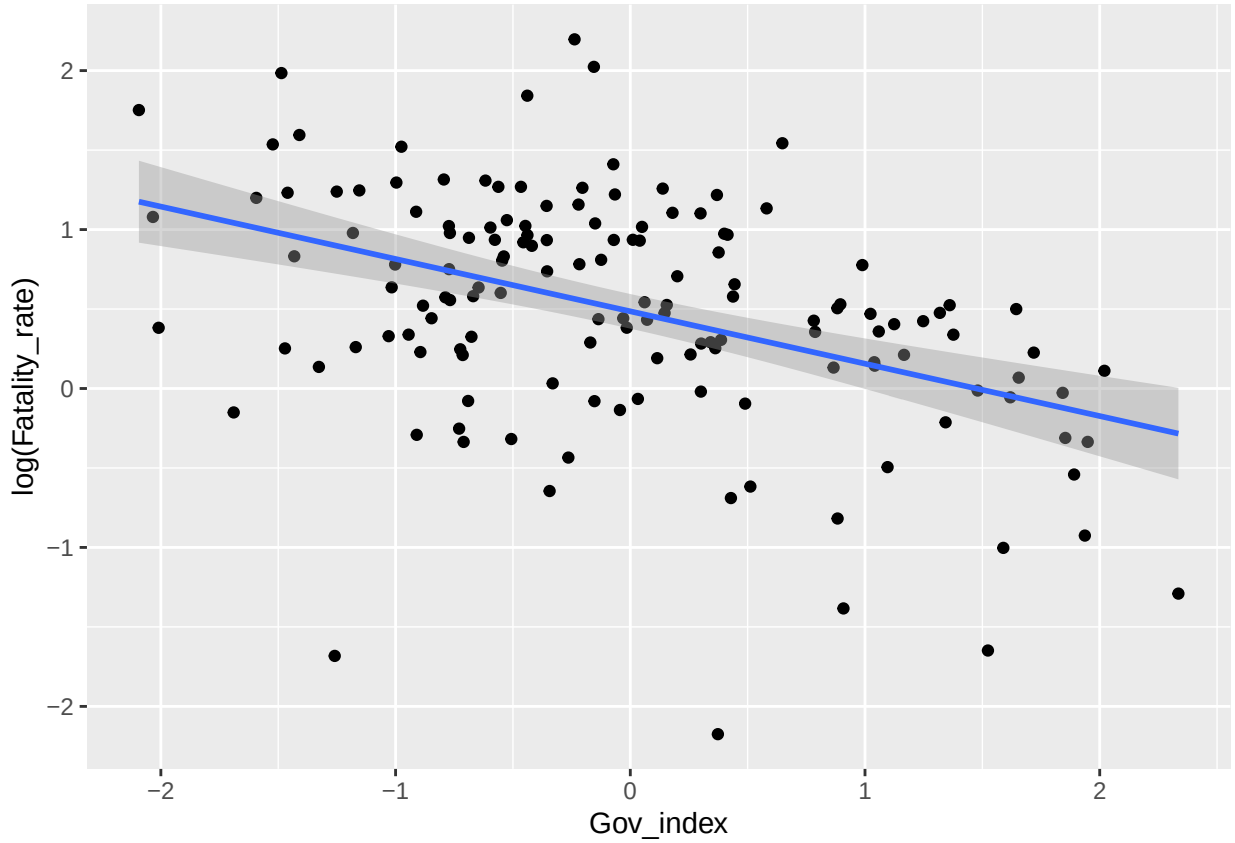



```
ggplot(aes(x = Gov_index, y = log(Fatality_rate)), data = df) +geom_point() + geom_smooth(method='lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
#Cleaning data
df <- df[df$Fatality_rate != 0,]
#Removing null values
df <- df[complete.cases(df$Fatality_rate),]
```

```
# Fitting models
```

```
df <- na.omit(df)
modelfull <-lm(Fatality_rate ~ age80+Gov_index+GDP_per_capita, data= df);summary(modelfull)
```

```
##
## Call:
## lm(formula = Fatality_rate ~ age80 + Gov_index + GDP_per_capita,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5779 -0.7523 -0.1681  0.4851  6.7342
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.863e+00  2.629e-01   7.084 6.12e-11 ***
## age80           2.338e-01  1.116e-01   2.095  0.03800 *
## Gov_index      -6.782e-01  2.147e-01  -3.159  0.00194 **
## GDP_per_capita -9.187e-06  9.237e-06  -0.995  0.32165
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.308 on 141 degrees of freedom
## Multiple R-squared:  0.1883, Adjusted R-squared:  0.171
## F-statistic:  10.9 on 3 and 141 DF,  p-value: 1.742e-06
```

```
bestmodel <- lm(Fatality_rate ~ age80+Gov_index, data= df);summary(bestmodel)
```

```
##
## Call:
## lm(formula = Fatality_rate ~ age80 + Gov_index, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5997 -0.7268 -0.1454  0.4677  6.7795
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7027     0.2080   8.185 1.41e-13 ***
## age80         0.2184     0.1105   1.976   0.0501 .
## Gov_index    -0.8201     0.1605  -5.110 1.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.308 on 142 degrees of freedom
## Multiple R-squared:  0.1826, Adjusted R-squared:  0.1711
## F-statistic: 15.86 on 2 and 142 DF,  p-value: 6.06e-07
```

```
#Fitting models
model1 <- lm(log(Fatality_rate)~GDP_per_capita, data = df); summary(model1)
```

```
##
## Call:
## lm(formula = log(Fatality_rate) ~ GDP_per_capita, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80528 -0.39494  0.07854  0.44165  1.57172
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.009e-01  7.659e-02  10.457  < 2e-16 ***
## GDP_per_capita -1.562e-05  2.703e-06  -5.778 4.54e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6632 on 143 degrees of freedom
## Multiple R-squared:  0.1893, Adjusted R-squared:  0.1836
## F-statistic: 33.39 on 1 and 143 DF,  p-value: 4.538e-08
```

```r
model2 <- lm(log(Fatality_rate)~Gov_index, data = df); summary(model2)
```

```
##
## Call:
## lm(formula = log(Fatality_rate) ~ Gov_index, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5829 -0.3226  0.1102  0.4004  1.6330
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.48555    0.05512   8.809 3.83e-15 ***
## Gov_index   -0.32944    0.05715  -5.765 4.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6635 on 143 degrees of freedom
## Multiple R-squared:  0.1886, Adjusted R-squared:  0.1829
## F-statistic: 33.23 on 1 and 143 DF,  p-value: 4.845e-08
```

```r
model3 <- lm(log(Fatality_rate)~Gov_index+GDP_per_capita, data = df); summary(model3)
```

```
##
## Call:
## lm(formula = log(Fatality_rate) ~ Gov_index + GDP_per_capita,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6729 -0.3635  0.1136  0.3972  1.5916
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.602e-01  1.074e-01   6.149 7.47e-09 ***
## Gov_index      -1.802e-01  9.718e-02  -1.855   0.0657 .
## GDP_per_capita -8.689e-06  4.599e-06  -1.889   0.0609 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6576 on 142 degrees of freedom
## Multiple R-squared:  0.2085, Adjusted R-squared:  0.1973
## F-statistic:  18.7 on 2 and 142 DF,  p-value: 6.184e-08
```

```r
data_control2 <- trainControl(method = "LOOCV") # Use Leave One Out.
train(log(Fatality_rate) ~ age80+Gov_index+GDP_per_capita,
      data = df,
      trControl = data_control2,
      method = "lm",
      na.action = na.pass)
```

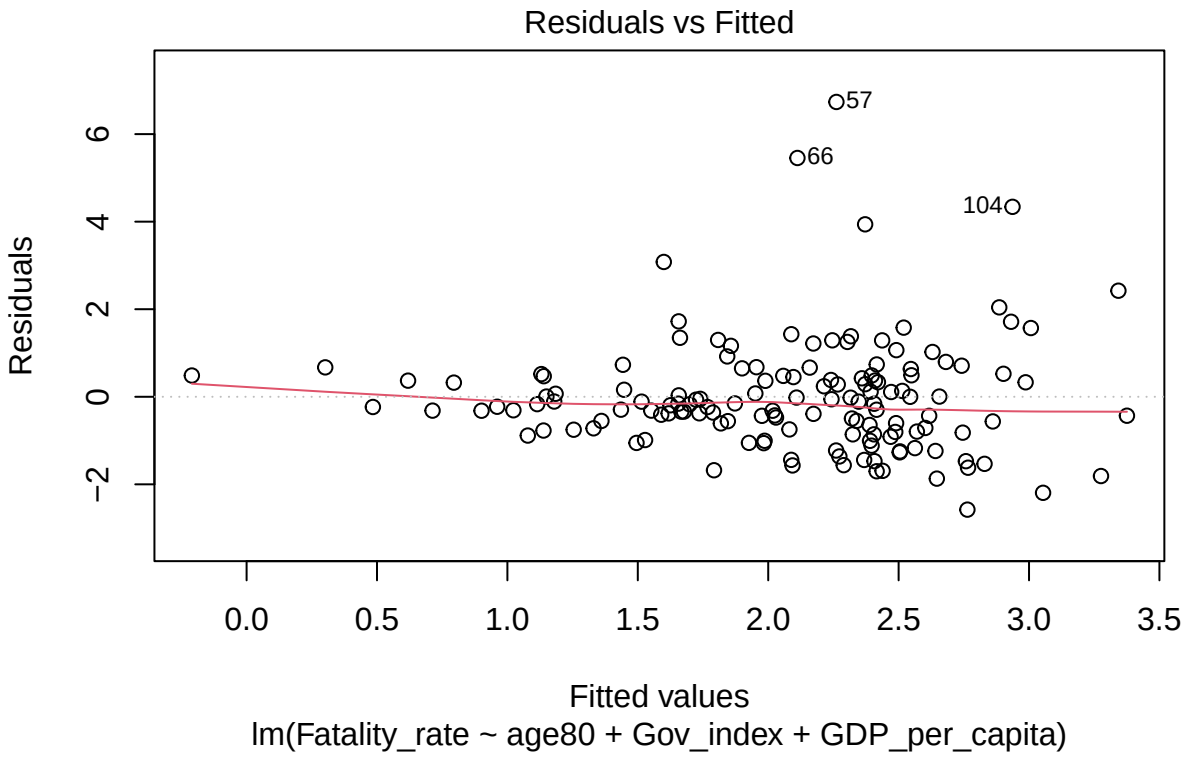
```
## Linear Regression
```

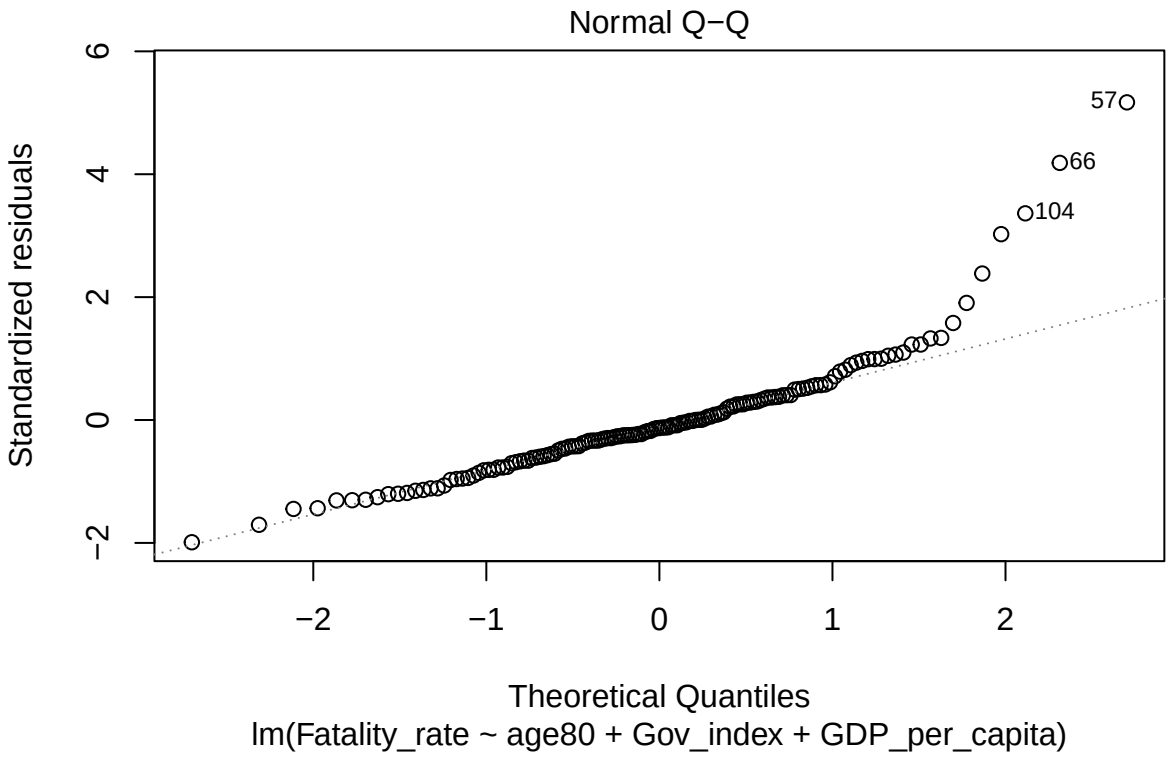
```
##
## 145 samples
##   3 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.6426236  0.2301614  0.4790895
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

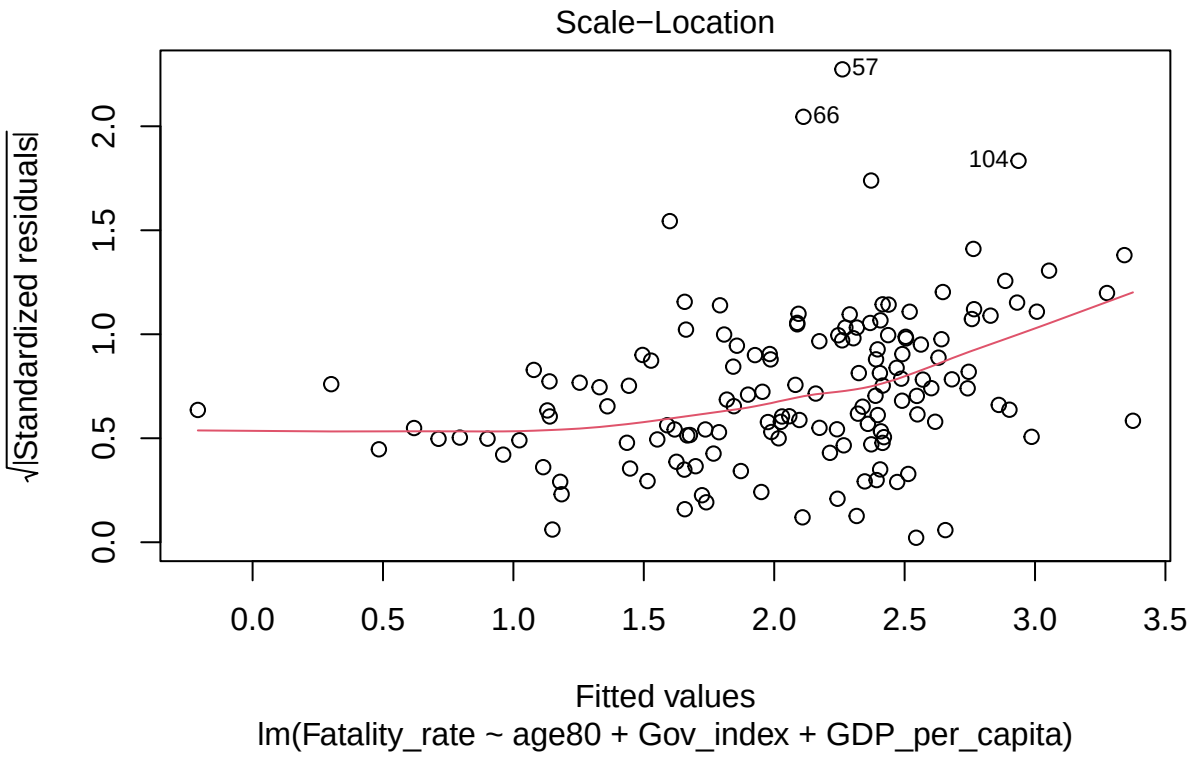
```
train((Fatality_rate) ~ age80+Gov_index+GDP_per_capita,
      data = df,
      trControl = data_control2,
      method = "lm",
      na.action = na.pass)
```

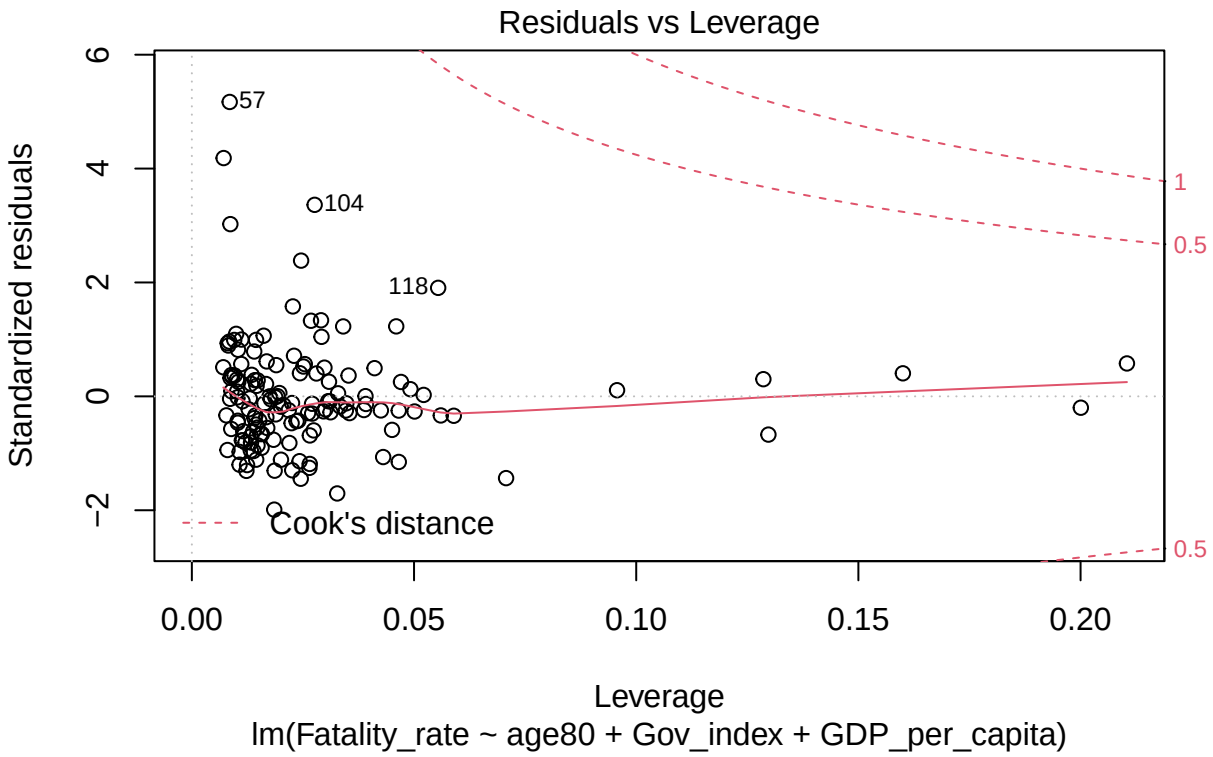
```
## Linear Regression
##
## 145 samples
##   3 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.315896  0.1571839  0.9028299
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
plot(modelfull)
```

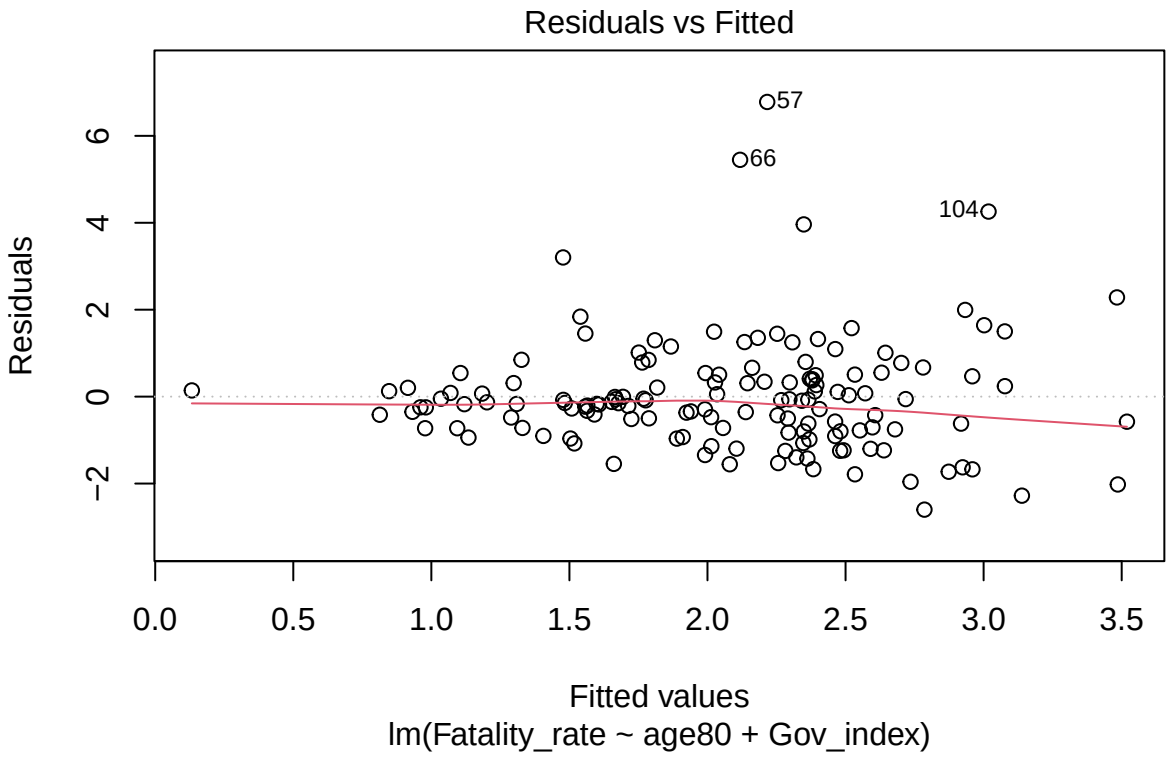

Residuals vs Fitted
57
66
104
Residuals
6
4
2
0
−2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Fitted values
lm(Fatality_rate ~ age80 + Gov_index + GDP_per_capita)

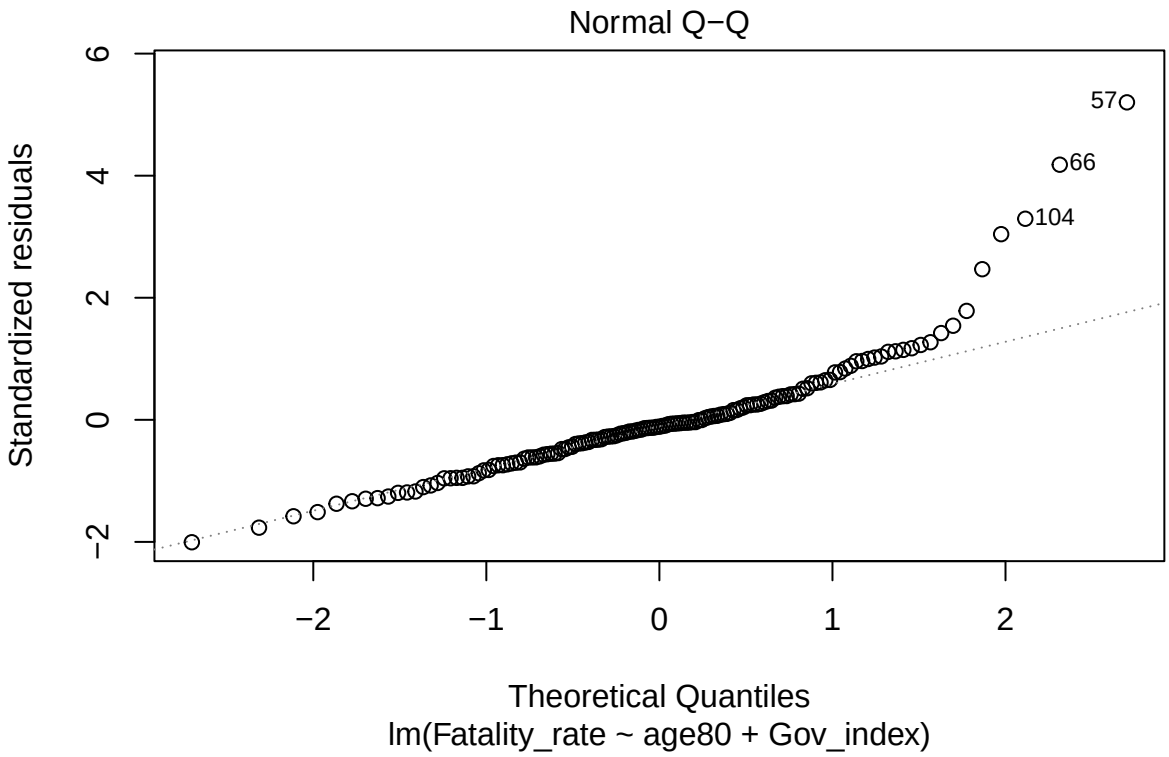

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Fatality_rate ~ age80 + Gov_index + GDP_per_capita)
57
66
104
−2
0
2
4
6
−2
−1
0
1
2

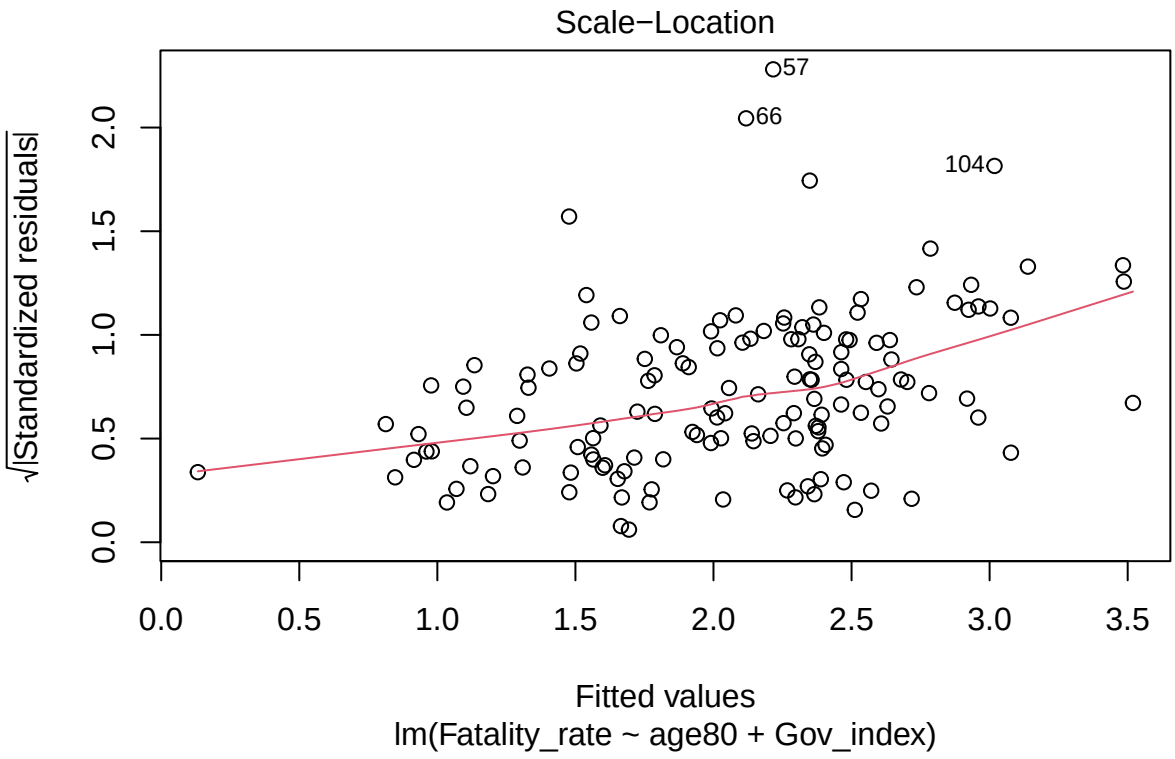

Scale−Location
57
66
104
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Fitted values
lm(Fatality_rate ~ age80 + Gov_index + GDP_per_capita)

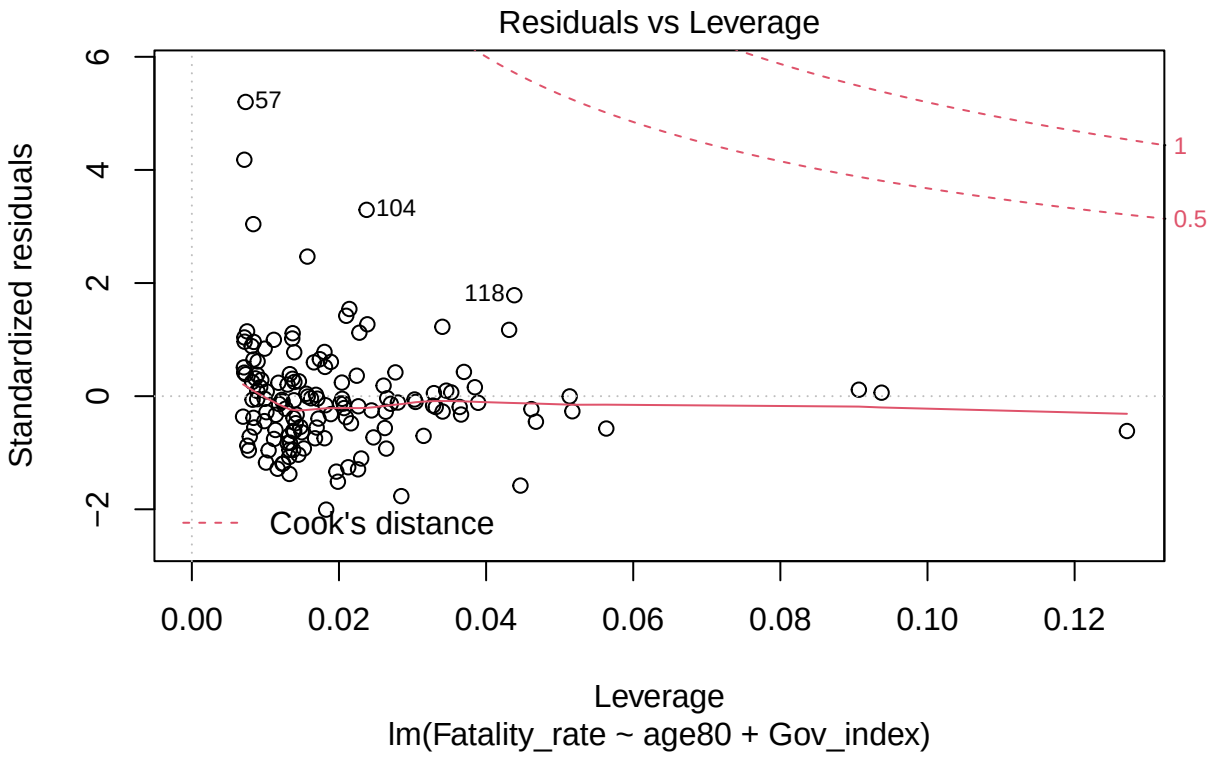



```
plot(bestmodel)
```


Residuals vs Fitted
57
66
104
Residuals
6
4
2
0
−2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Fitted values
lm(Fatality_rate ~ age80 + Gov_index)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Fatality_rate ~ age80 + Gov_index)
57
66
104
−2
0
2
4
6
−2
−1
0
1
2


Scale−Location
57
66
104
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Fitted values
lm(Fatality_rate ~ age80 + Gov_index)


Residuals vs Leverage
Standardized residuals
6
4
2
0
−2
57
104
118
1
0.5
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Leverage
lm(Fatality_rate ~ age80 + Gov_index)