```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(modelr)
```

```
## Warning: package 'modelr' was built under R version 4.1.2
```

P1: We would like to build a model for predicting life expectancy. Create a data frame that includes onlycomplete cases (no missing values) and includes columns for country code, year, and the followingresponse + predictors. Visualize life expectancy versus the five candidate predictors, transforming variables as necessary, anddescribe their relationships.

```
file_path <-"D:/Education/MS DS NEU/IDMP/Assignments/HW4/ddf--gapminder--systema_globalis-master/countr
lifeexp_path <- "ddf--datapoints--life_expectancy_years--by--geo--time"
lifeexp <-read_csv(file.path(file_path,paste0(lifeexp_path, ".csv")))
```

```
## Rows: 56130 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, life_expectancy_years

##
## 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.
```

```
inf_mort_path <- "ddf--datapoints--infant_mortality_rate_per_1000_births--by--geo--time"
inf_mort <-read_csv(file.path(file_path,paste0(inf_mort_path, ".csv")))
```

```
## Rows: 13654 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, infant_mortality_rate_per_1000_births

##
## 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
murder_path <- "ddf--datapoints--murder_per_100000_people--by--geo--time"
murder_rate <-read_csv(file.path(file_path,paste0(murder_path, ".csv")))
```

```
## Rows: 3166 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, murder_per_100000_people

##
## 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
gdp_path <- "ddf--datapoints--gdppercapita_us_inflation_adjusted--by--geo--time"
gdp <-read_csv(file.path(file_path,paste0(gdp_path, ".csv")))
```

```
## Rows: 9427 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, gdppercapita_us_inflation_adjusted

##
## 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
doctors_path <- "ddf--datapoints--medical_doctors_per_1000_people--by--geo--time"
doc_data <-read_csv(file.path(file_path,paste0(doctors_path, ".csv")))
```

```
## Rows: 4705 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, medical_doctors_per_1000_people

##
## 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
poverty_path <- "ddf--datapoints--poverty_percent_people_below_550_a_day--by--geo--time"
poverty_data <-read_csv(file.path(file_path,paste0(poverty_path, ".csv")))
```

```
## Rows: 1685 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, poverty_percent_people_below_550_a_day

##
## 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
data_df <- lifeexp%>%
  inner_join(inf_mort)%>%
  inner_join(murder_rate)%>%
  inner_join(gdp)%>%
  inner_join(doc_data)%>%
  inner_join(poverty_data)%>%
  rename(lifeexp=life_expectancy_years,
         inf_mort=infant_mortality_rate_per_1000_births,
         murder_rate=murder_per_100000_people,
         gdp=gdppercapita_us_inflation_adjusted,
         doc_data=medical_doctors_per_1000_people,
         poverty_data=poverty_percent_people_below_550_a_day)
```

```
## Joining, by = c("geo", "time")
```

```
## Joining, by = c("geo", "time")
## Joining, by = c("geo", "time")
## Joining, by = c("geo", "time")
## Joining, by = c("geo", "time")
```

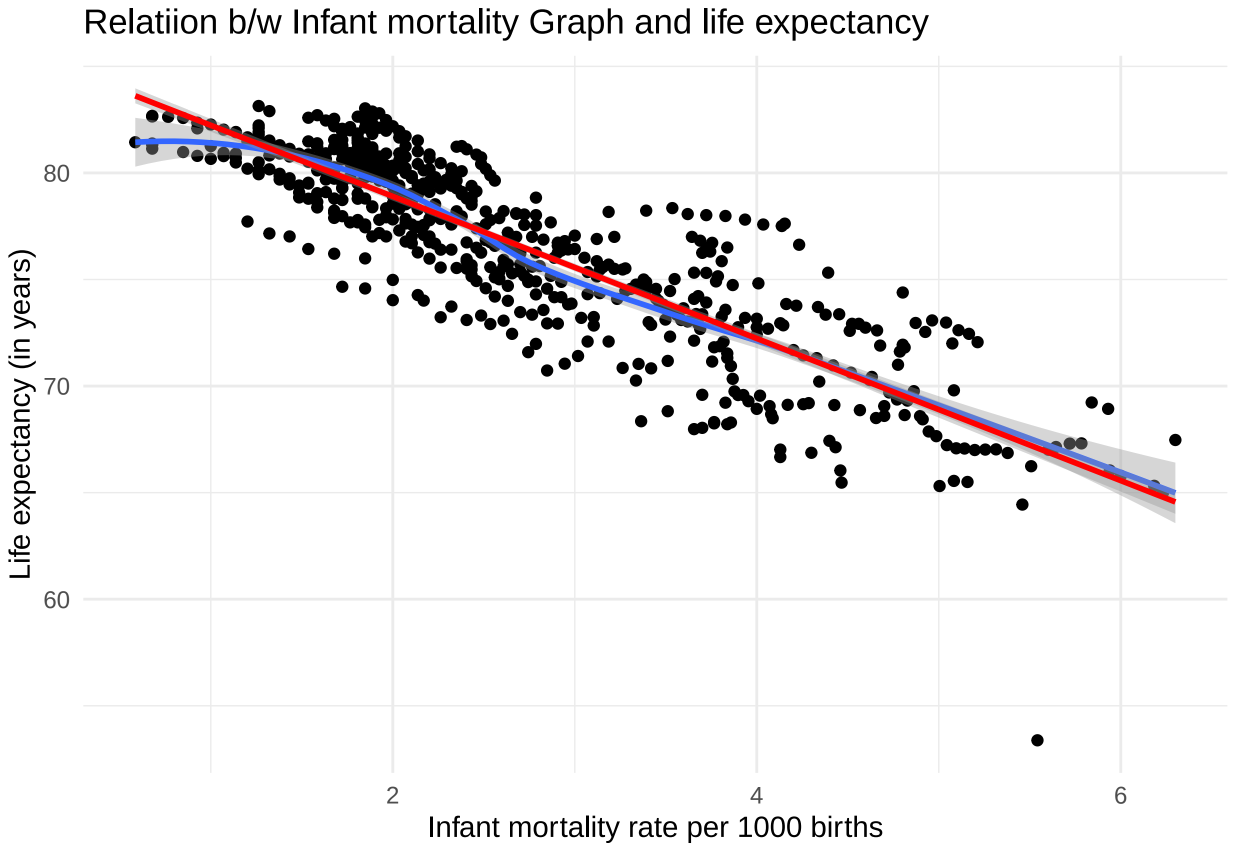
```
data_df
```

```
## # A tibble: 628 x 8
##    geo    time lifeexp inf_mort murder_rate   gdp doc_data poverty_data
##    <chr> <dbl>   <dbl>    <dbl>       <dbl> <dbl>    <dbl>        <dbl>
##  1 alb    1996    74.4     27.9        8.23 1870.    1.38          51.5
##  2 alb    2002    75.3     21          7.40 2573.    1.17          54.1
##  3 arg    1986    71.8     28.1        5.89 7214.    2.98           4.9
##  4 arg    1992    72.6     22.8        4.70 7157.    2.65          14.9
##  5 arg    1995    73.4     20.8        4.22 7667.    2.68          20.5
##  6 arm    1999    71.9     27.9        2.58 1317.    0.693         83.2
##  7 arm    2001    72.6     25.3        1.76 1547.    2.62          84.4
##  8 arm    2002    72.7     24.2        2.25 1761.    2.56          83.5
##  9 arm    2003    72.9     23          1.82 2018.    2.46          83
## 10 arm    2008    73.8     17.9        1.69 3628.    2.74          53.6
## # ... with 618 more rows
```

```
ggplot(data_df,aes(x=log2(inf_mort),y=lifeexp))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Infant mortality rate per 1000 births",y="Life expectancy (in years)",title="Relatiion b/w In
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

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

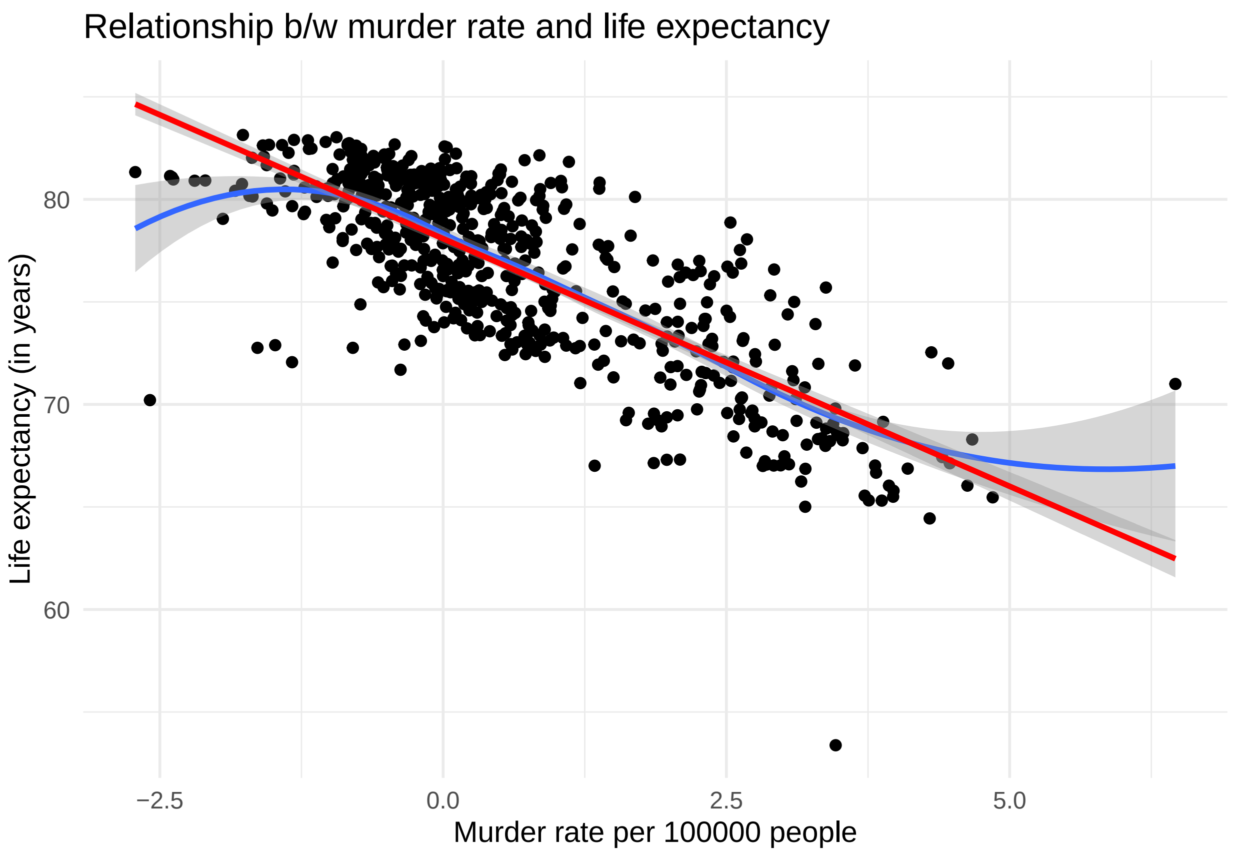


#### There is a negative relationship between Infant mortality rate and Life Expectancy

```
ggplot(data_df,aes(x=log2(murder_rate),y=lifeexp))+
  geom_point()+geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Murder rate per 100000 people",y="Life expectancy (in years)",title="Relationship b/w murder
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

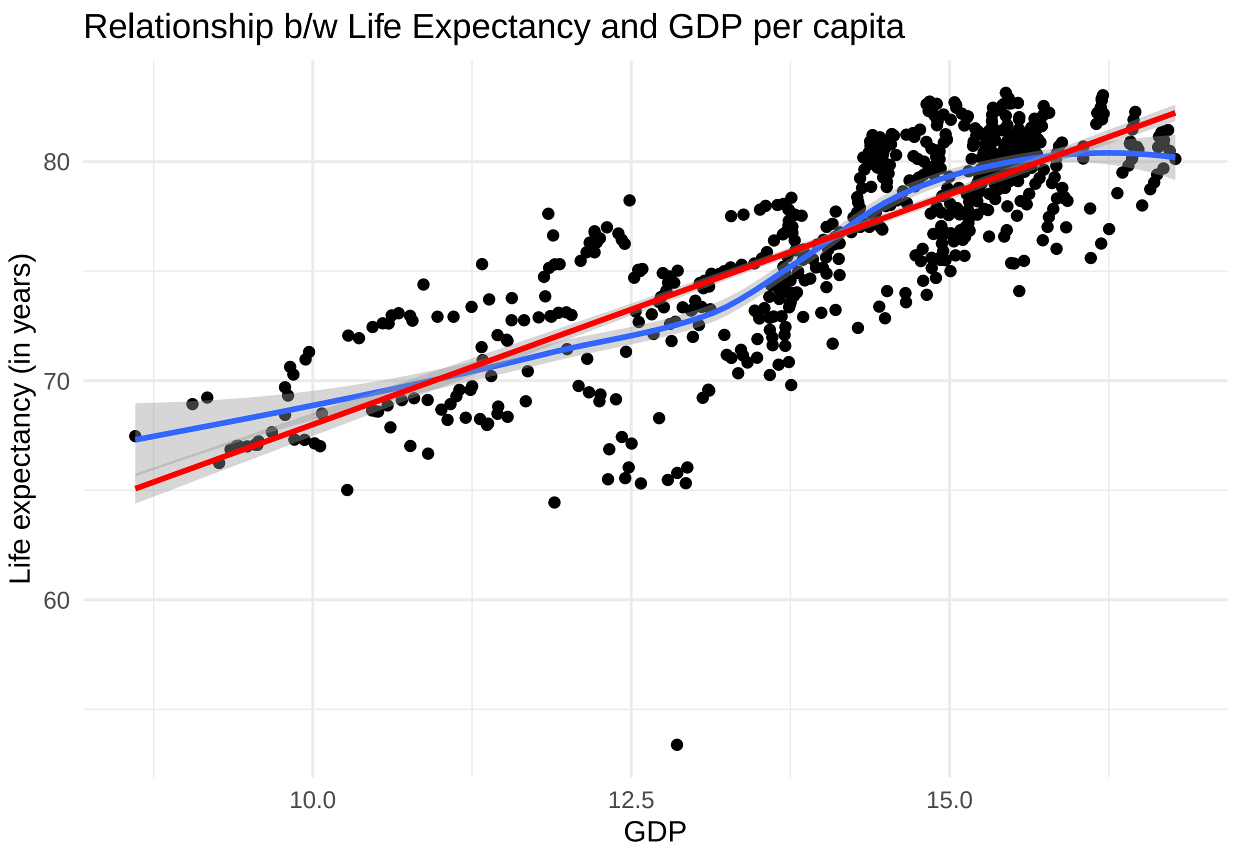
```
## `geom_smooth()` using formula 'y ~ x'
```



#### We can observe a negative relationship between Murder rate and Life expectancy

```
ggplot(data_df,aes(x=log2(gdp),y=lifeexp))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+labs(x="GDP",y="Life expectancy (in years)",title="Relationship
  theme_minimal()
```

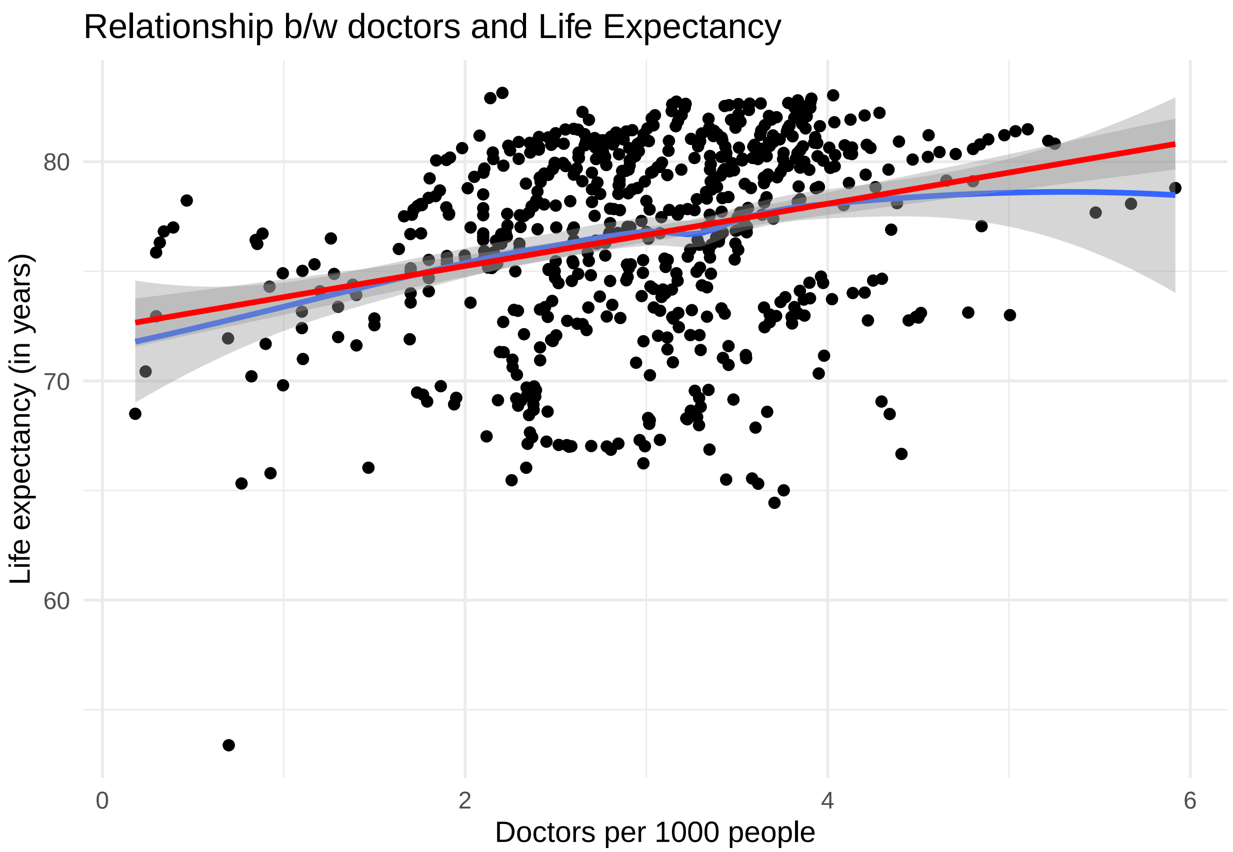
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

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



#### We can observer a positive relationship between GDP and Life expectancy

```
ggplot(data_df,aes(x=doc_data,y=lifeexp))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Doctors per 1000 people",y="Life expectancy (in years)",title="Relationship b/w doctors and Li
  theme_minimal()
```

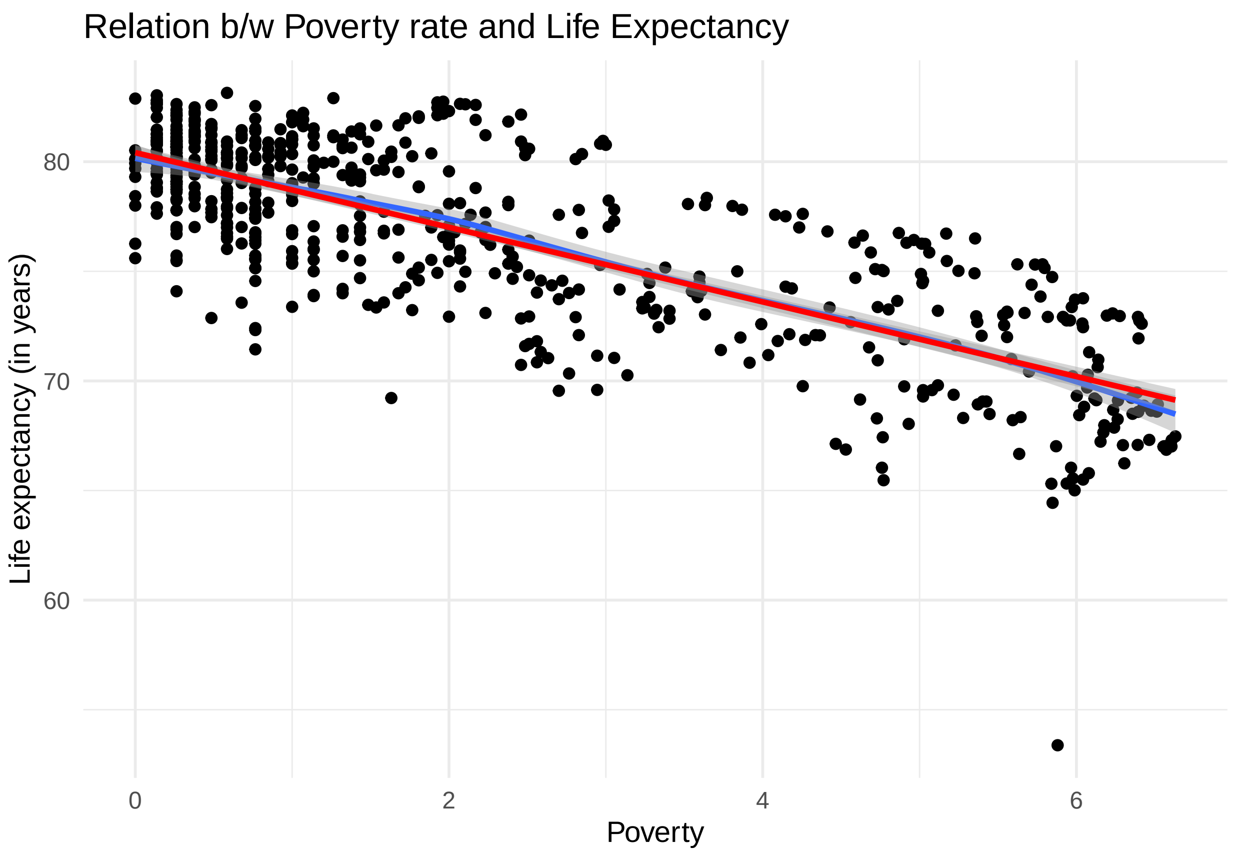
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

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



```
ggplot(data_df,aes(x=log2(1+poverty_data),y=lifeexp))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Poverty",y="Life expectancy (in years)",title="Relation b/w Poverty rate and Life Expectancy"
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

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



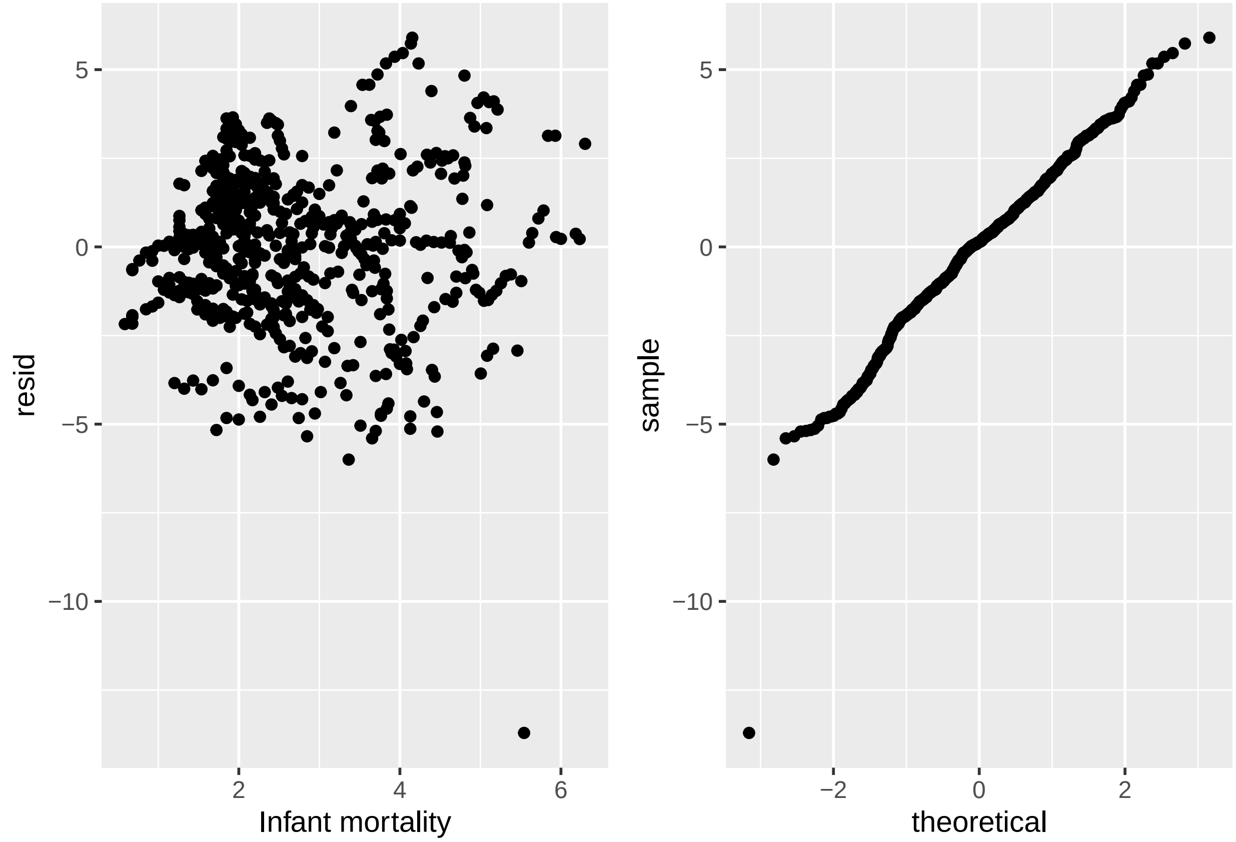
#### We can oobserve a negative relationship between Life Expectancy and poverty data

P2: Build a linear regression model for life expectancy using a single predictor, justifying your choice basedonly on the visualizations from Problem 1. Then use residual plots to perform model diagnostics.Comment on any outliers or violations of model assumptions you notice in the residual plots. If necessary,fix the issue, re-model the model, and perform model diagnostics again.

```
model1 <-lm(lifeexp~ log2(inf_mort), data=data_df)
summary(model1)
```

```
##
## Call:
## lm(formula = lifeexp ~ log2(inf_mort), data = data_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7098  -1.2938   0.1379   1.3695   5.9011
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    85.56749    0.21845  391.71   <2e-16 ***
## log2(inf_mort) -3.33396    0.07457  -44.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.178 on 626 degrees of freedom
## Multiple R-squared:  0.7615, Adjusted R-squared:  0.7611
## F-statistic:  1999 on 1 and 626 DF,  p-value: < 2.2e-16
```

```
graph_1 <- data_df%>%
  add_residuals(model1, "resid")%>%
  ggplot(aes(x=log2(inf_mort),y=resid))+
  geom_point()+labs(x="Infant mortality")
graph_2 <- data_df%>%
  add_residuals(model1, "resid")%>%
  ggplot(aes(sample=resid))+
  geom_qq()
gridExtra::grid.arrange(graph_1, graph_2, ncol=2)
```



```
outliers <- data_df%>%
  add_residuals(model1, "resid")%>%
  filter(resid< -10)
outliers
```

```
## # A tibble: 1 x 9
##   geo    time lifeexp inf_mort murder_rate   gdp doc_data poverty_data resid
##   <chr> <dbl>   <dbl>    <dbl>       <dbl> <dbl>    <dbl>        <dbl> <dbl>
## 1 zaf    2008    53.4     46.6        11.0 7432.    0.697         57.9 -13.7
```

```
data_df2 <-anti_join(data_df, outliers, by=c("geo", "time"))
model2 <-lm(lifeexp~ log2(inf_mort), data=data_df2)
summary(model2)
```

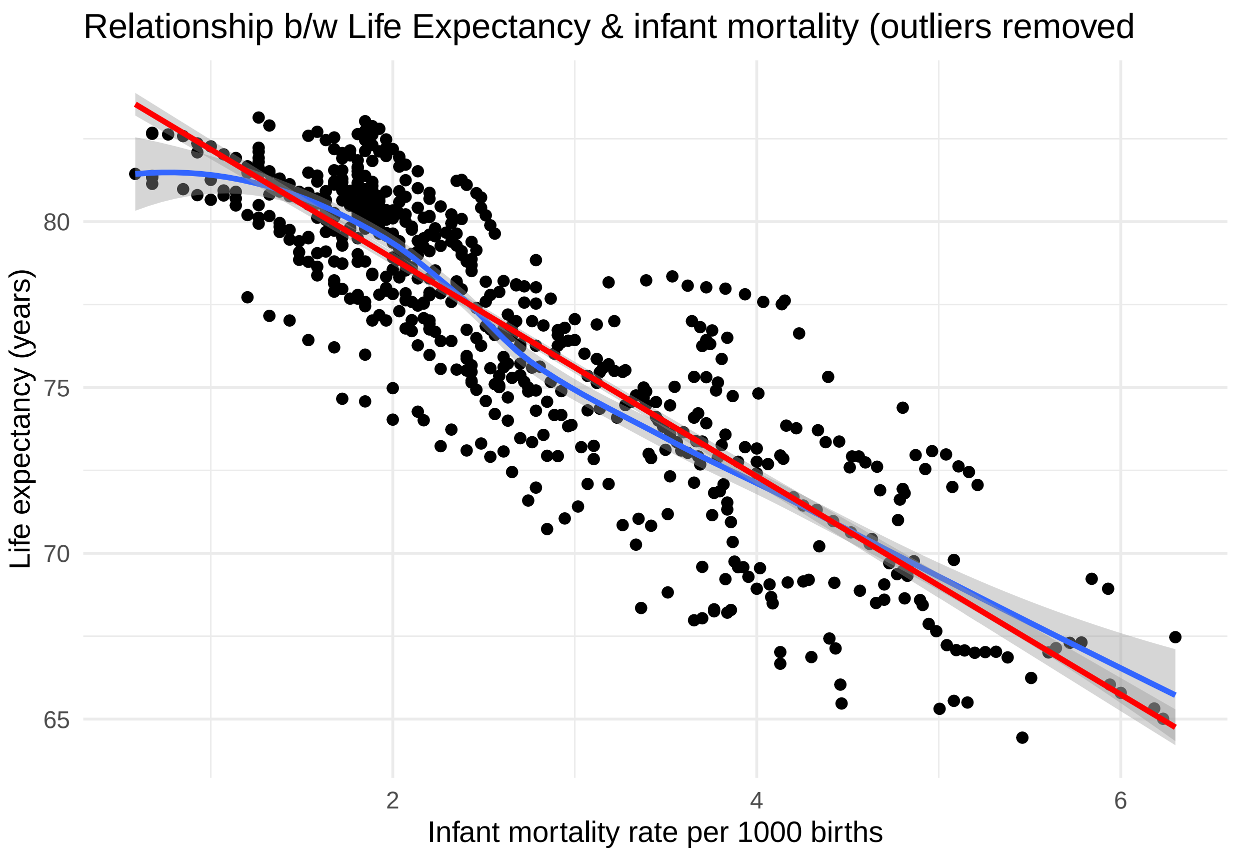
```
##
```

```
## Call:
## lm(formula = lifeexp ~ log2(inf_mort), data = data_df2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.054 -1.317  0.105  1.385  5.811
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    85.46482    0.21209  402.97   <2e-16 ***
## log2(inf_mort) -3.28755    0.07255  -45.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.108 on 625 degrees of freedom
## Multiple R-squared:  0.7667, Adjusted R-squared:  0.7663
## F-statistic:  2053 on 1 and 625 DF,  p-value: < 2.2e-16
```

```
ggplot(data_df2,aes(x=log2(inf_mort),y=lifeexp))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Infant mortality rate per 1000 births",y="Life expectancy (years)",title="Relationship b/w Li
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

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

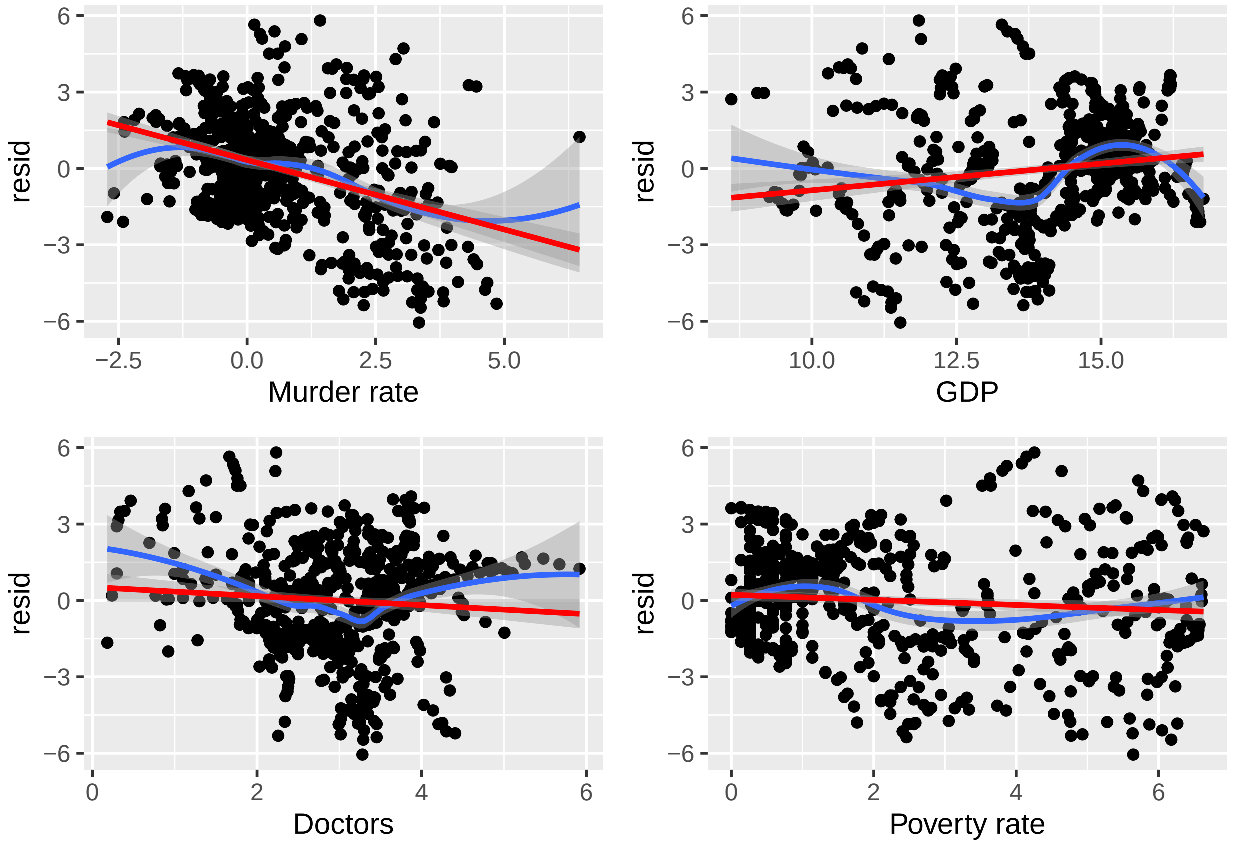


#### Observation is same as before but no outliers.

P3: Use residual plots to determine if any other candidate predictors should be added to your model fromProblem 2. If so, add up to one additional predictor to the model, and then perform model diagnosticson the new model.

```
graph_1 <- data_df2%>%
  add_residuals(model2, "resid")%>%
  ggplot(aes(x=log2(murder_rate),y=resid))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Murder rate")
graph_2 <- data_df2%>%
  add_residuals(model2, "resid")%>%
  ggplot(aes(x=log2(gdp),y=resid))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="GDP")
graph_3 <- data_df2%>%
  add_residuals(model2, "resid")%>%
  ggplot(aes(x=doc_data,y=resid))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Doctors")
graph_4 <- data_df2%>%
  add_residuals(model2, "resid")%>%
  ggplot(aes(x=log2(1+poverty_data),y=resid))+
  geom_point()+
  geom_smooth()+
  geom_smooth(method="lm", color="red")+
  labs(x="Poverty rate")
gridExtra::grid.arrange(graph_1, graph_2, graph_3, graph_4)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

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

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

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

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

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

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

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

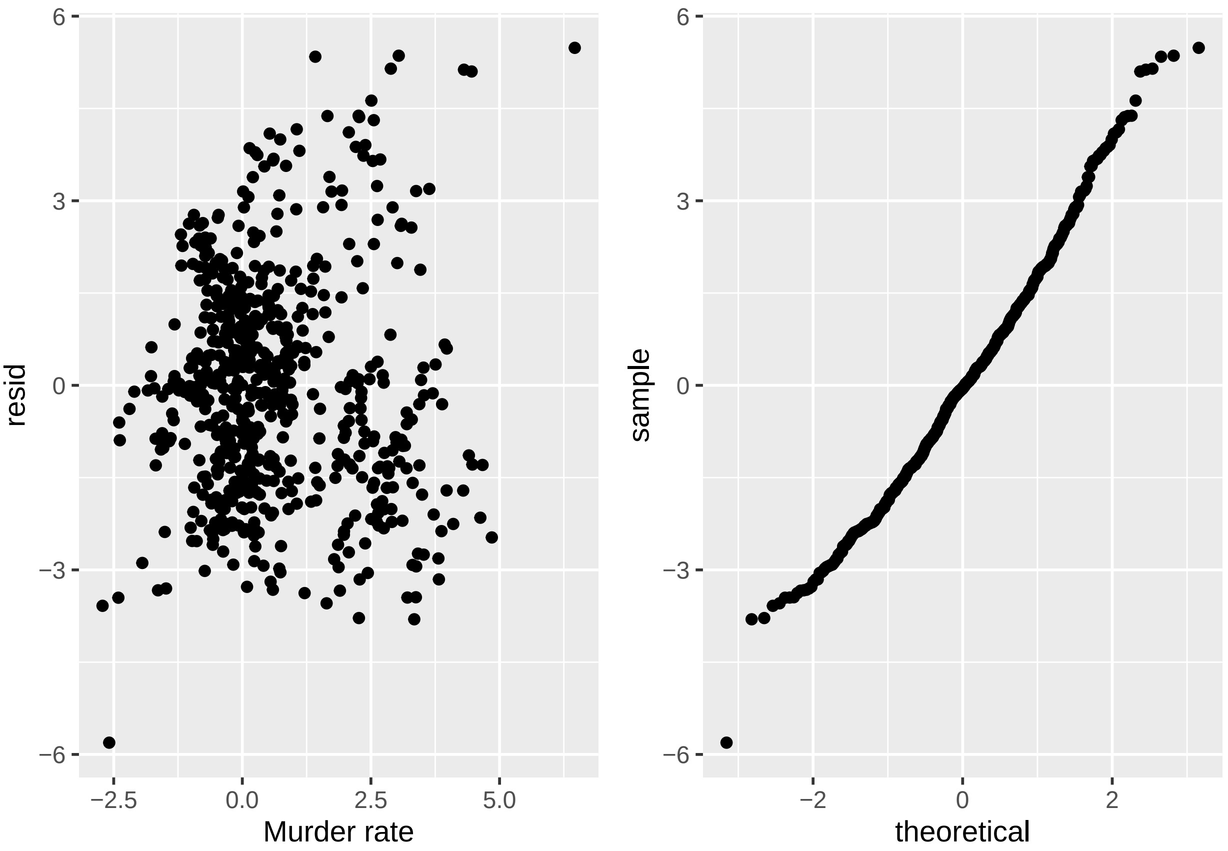


#### Only murder rate showcased a negative relationship while the others were completely random and need not be included in the model. Only murder_rate can be included.

```
model3 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate), data=data_df2)
summary(model3)
```

```
##
## Call:
## lm(formula = lifeexp ~ log2(inf_mort) + log2(murder_rate), data = data_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8085 -1.3207 -0.0484  1.1241  5.4854
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       83.69144    0.21647  386.62   <2e-16 ***
## log2(inf_mort)    -2.38991    0.08607  -27.77   <2e-16 ***
## log2(murder_rate) -1.04643    0.06940  -15.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.806 on 624 degrees of freedom
## Multiple R-squared:  0.829,  Adjusted R-squared:  0.8284
## F-statistic:  1512 on 2 and 624 DF,  p-value: < 2.2e-16
```

```
graph_1 <- data_df2%>%
  add_residuals(model3, "resid")%>%
  ggplot(aes(x=log2(murder_rate),y=resid))+
  geom_point()+
  labs(x="Murder rate")
graph_2 <- data_df2%>%
  add_residuals(model3, "resid")%>%
  ggplot(aes(sample=resid))+
  geom_qq()
gridExtra::grid.arrange(graph_1, graph_2, ncol=2)
```



#### Both the graph shows no significant abnormalities and can be considered an ideal random scatter plot and it is the same case with the residuals. An outlier is there but since it is not extremely abnormal, we can leave it or keep it.

P4: Using the full dataset (minus any outliers you removed), perform reproducible 10-fold cross-validation onyour model from Problem 3. Report the cross-validated RMSE, as well as the RMSE of the model fromProblem 3 on the data used to train it. Which RMSE is larger? Is this surprising, and why?

```
set.seed(2020)
cv_df <-crossv_kfold(data_df2, k=10)
cv_df <-mutate(cv_df,model =map(train,~ lm(lifeexp~ log2(inf_mort)+ log2(murder_rate),data = .)),
              rmse =map2_dbl(model, test,~ rmse(.x, .y)))

mean(cv_df$rmse)
```

Set up seed value and perform k fold cross validation and print final CV value

```
## [1] 1.811058
```

```
rmse(model3, data_df2)
```

```
## [1] 1.802156
```

The CV RMSE is slightly larger than the regular RMSE. And its not that surprising, because of the test and training data differences.

P5: Reproducibly partition the dataset (minus any outliers) into a training, validation, and test set using a50/25/25 split. Keeping any transformations you found to be appropriate in Problem 1, perform stepwisemodel selection to build a predictive model for life expectancy using RMSE as the selection criterion.Show the RMSEs at each step and note which variable is being added/dropped, and then report theRMSE of the selected model on the test set.

```
set.seed(2020)
dfpart <-resample_partition(data_df2, p=c(train=0.5, valid=0.25, test=0.25))
```

Divide the Data

```
mean_1 <-lm(lifeexp~ log2(inf_mort), data=dfpart$train)
mean_2 <-lm(lifeexp~ log2(murder_rate), data=dfpart$train)
mean_3 <-lm(lifeexp~ log2(gdp), data=dfpart$train)
mean_4 <-lm(lifeexp~doc_data, data=dfpart$train)
mean_5 <-lm(lifeexp~ log2(1+poverty_data), data=dfpart$train)

rmse(mean_1, dfpart$valid)
```

Step 1:

```
## [1] 2.010126
```

```
rmse(mean_2, dfpart$valid)
```

```
## [1] 2.914585
```

```
rmse(mean_3, dfpart$valid)
```

```
## [1] 2.482432
```

```
rmse(mean_4, dfpart$valid)
```

```
## [1] 4.051756
```

```
rmse(mean_5, dfpart$valid)
```

```
## [1] 2.745777
```

```
mean_12 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate), data=dfpart$train)
mean_13 <-lm(lifeexp~ log2(inf_mort)+ log2(gdp), data=dfpart$train)
mean_14 <-lm(lifeexp~ log2(inf_mort)+doc_data, data=dfpart$train)
mean_15 <-lm(lifeexp~ log2(inf_mort)+ log2(1+poverty_data), data=dfpart$train)
rmse(mean_12, dfpart$valid)
```

Step 2:

```
## [1] 1.818872
```

```
rmse(mean_13, dfpart$valid)
```

```
## [1] 1.948741
```

```
rmse(mean_14, dfpart$valid)
```

```
## [1] 2.020564
```

```
rmse(mean_15, dfpart$valid)
```

```
## [1] 2.04759
```

```
mean_123 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate)+ log2(gdp),data=dfpart$train)
mean_124 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate)+doc_data,data=dfpart$train)
mean_125 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate)+ log2(1+poverty_data),data=dfpart$train)
rmse(mean_123, dfpart$valid)
```

Step 3:

```
## [1] 1.74326
```

```
rmse(mean_124, dfpart$valid)
```

```
## [1] 1.815282
```

```
rmse(mean_125, dfpart$valid)
```

```
## [1] 1.820939
```

```
mean_1234 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate)+log2(gdp)+doc_data,data=dfpart$train)
mean_1235 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate)+log2(gdp)+ log2(1+poverty_data),data=dfpart$t
rmse(mean_1234, dfpart$valid)
```

Step 4:

```
## [1] 1.744491
```

```
rmse(mean_1235, dfpart$valid)
```

```
## [1] 1.716341
```

```
mean_12354 <-lm(lifeexp~ log2(inf_mort)+ log2(murder_rate)+log2(gdp)+ log2(1+poverty_data)+doc_data,dat
rmse(mean_12354, dfpart$valid)
```

Step 5:

```
## [1] 1.71726
```

```
rmse(mean_1235, dfpart$test)
```

There is no need to add the doctors as the RMSE increased.

```
## [1] 1.78397
```