Assignment_4

2022-11-05

#PART A

##Problem -1 Importing all the necessary packages

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

```
library(modelr)
library(purrr)
```

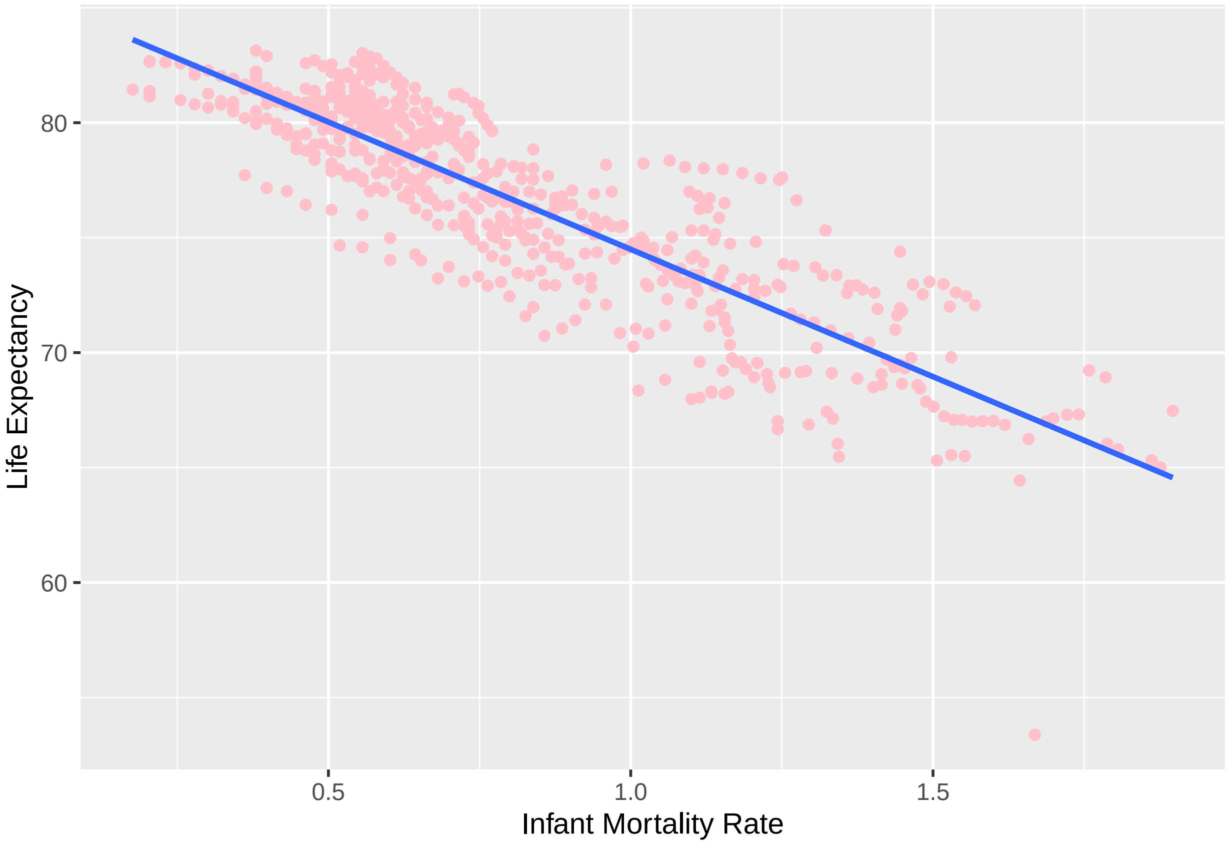
Merging all the files to get the final dataset

```
life_expectancy <- read.csv('gapminder/countries-etc-datapoints/ddf--datapoints--life_expectancy_years--

infant_mortality_rate <- read.csv('gapminder/countries-etc-datapoints/ddf--datapoints--infant_mortality_

murder <- read.csv('gapminder/countries-etc-datapoints/ddf--datapoints--murder_per_100000_people--by--g

gdp <- read.csv('gapminder/countries-etc-datapoints/ddf--datapoints--gdppercapita_us_inflation_adjusted-

medical <- read.csv('gapminder/countries-etc-datapoints/ddf--datapoints--medical_doctors_per_1000_people

poverty_rate <- read.csv('gapminder/countries-etc-datapoints/ddf--datapoints--poverty_percent_people_bel

data <- merge(life_expectancy,infant_mortality_rate,by=c('geo','time'))
data <- merge(data,murder, by = c('geo','time'))
data <- merge(data,gdp, by = c('geo','time'))
data <- merge(data,medical, by = c('geo','time'))
data <- merge(data,poverty_rate, by = c('geo','time'))
```

Visualizing each predictor v/s Life Expectancy

```
y = data$life_expectancy_years
x = log10(data$infant_mortality_rate_per_1000_births)
plt <- ggplot(data, aes(x=x,y=y),alpha=0.1)
plt + geom_point(colour = 'pink')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+labs(y="Life Expecta
```

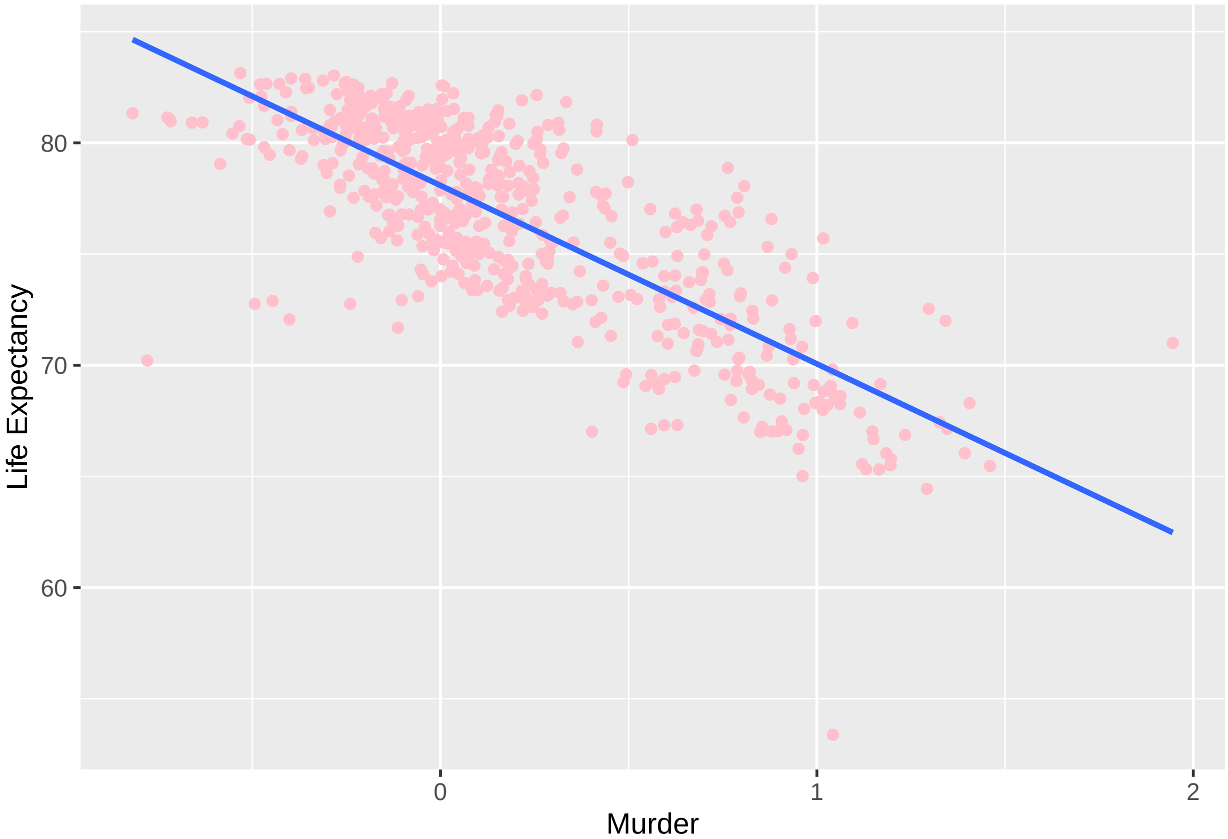
```
## `geom_smooth()` using formula 'y ~ x'
```



`Infant-mortality-rate` was right skewed therefore I used `log-transformation` on it. Mortality has a strong negative correlation with life expectancy.

```
y = data$life_expectancy_years
x = log10(data$murder_per_100000_people)
plt <- ggplot(data, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'pink')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Life Expec
```

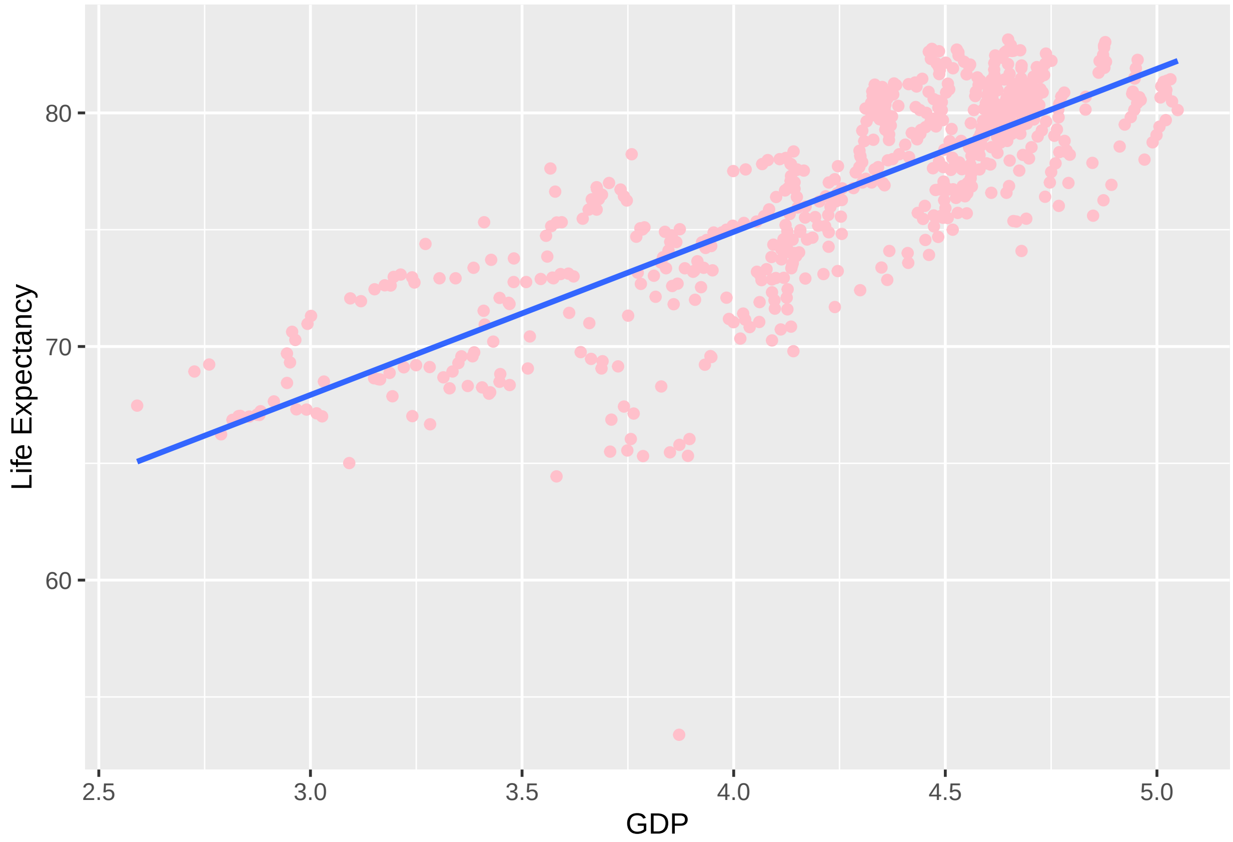
```
## `geom_smooth()` using formula 'y ~ x'
```



Murder was right skewed therefore I used log-transformation on it. Murder has a negative correlation with life expectancy.

```
x = data$life_expectancy_years
y = log10(data$gdppercapita_us_inflation_adjusted)
plt <- ggplot(data, aes(x=y,y=x,alpha(0.1)))
plt + geom_point(colour = 'pink')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Life Expec
```

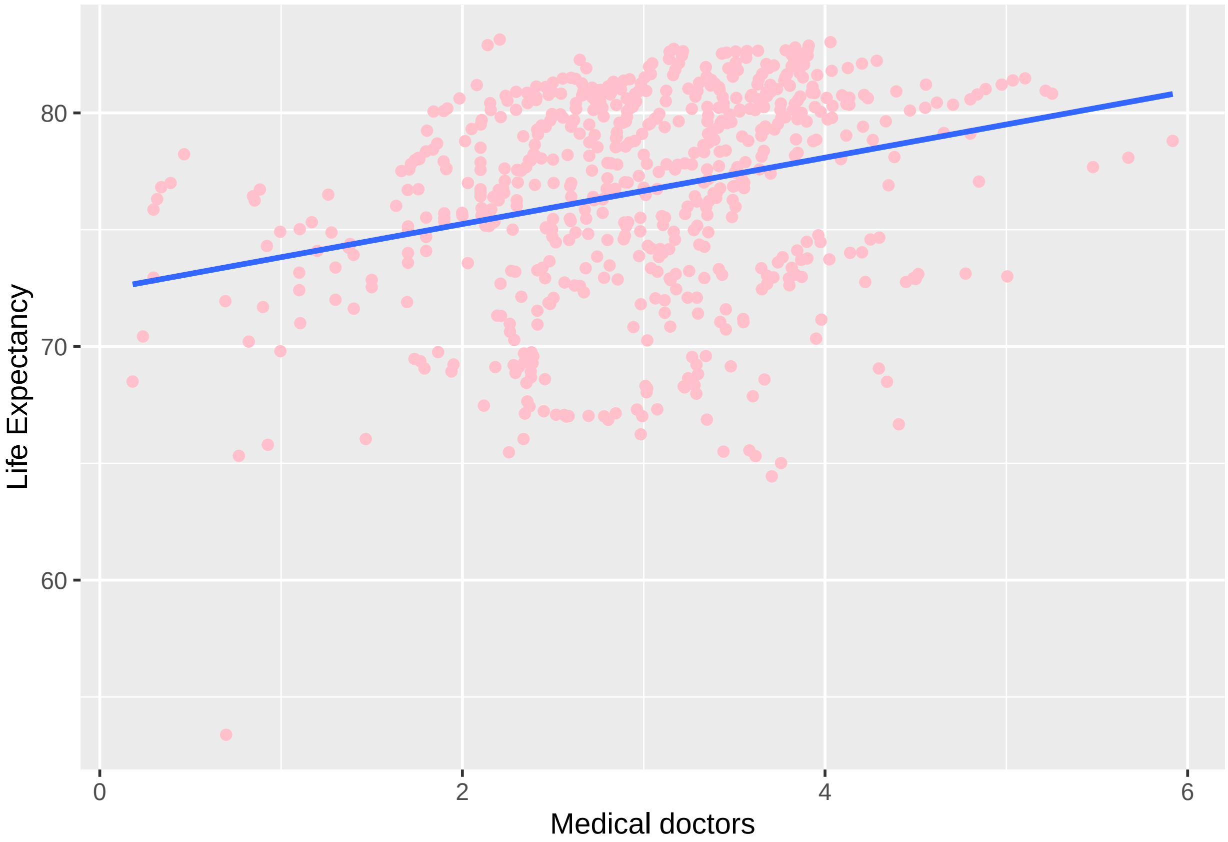
```
## `geom_smooth()` using formula 'y ~ x'
```



`GDP` was right skewed therefore I used `log-transformation` on it. GDP has a positive correlation with life expectancy.

```
y = data$life_expectancy_years
x = data$medical_doctors_per_1000_people
plt <- ggplot(data, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'pink')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Life Expec
```

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

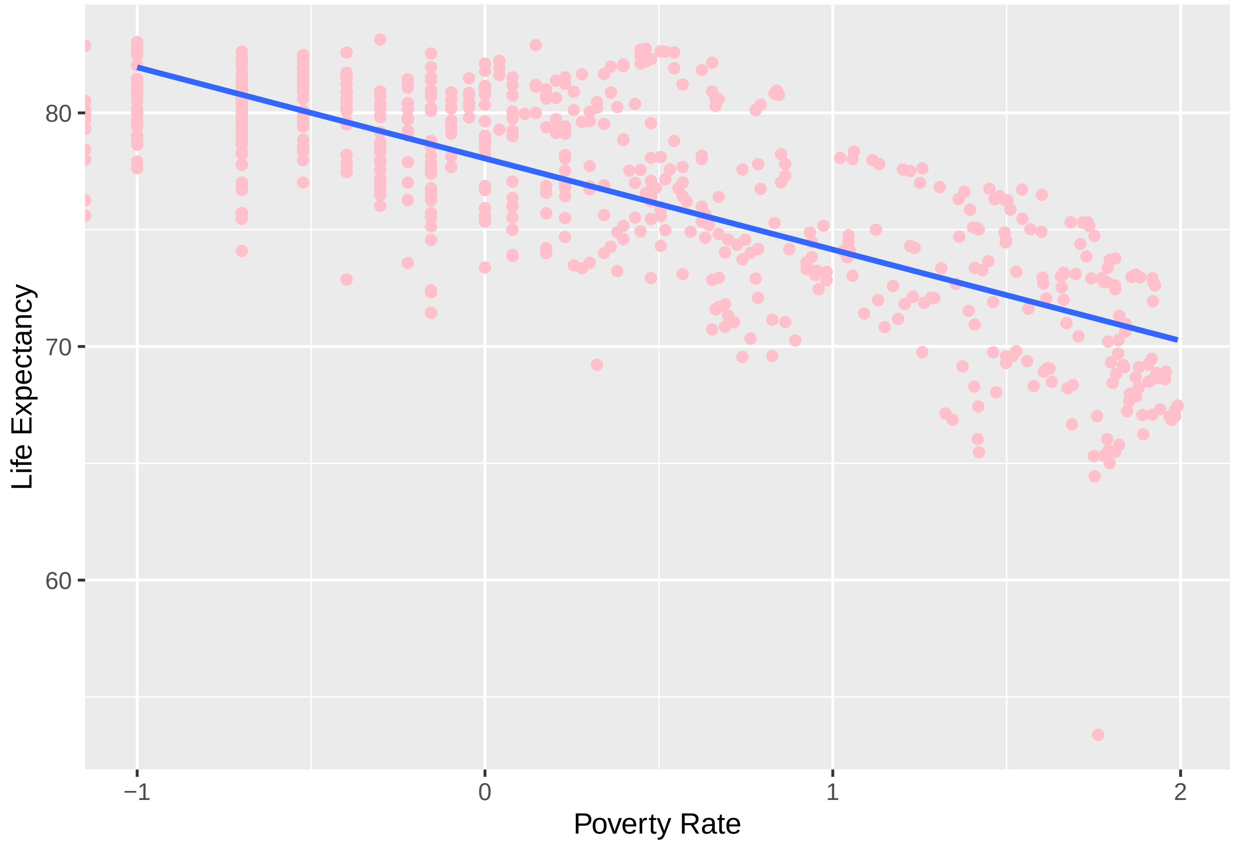


Medical Doctors has a slight positive correlation with life expectancy.

```
y = data$life_expectancy_years
x = log10(data$poverty_percent_people_below_550_a_day)
plt <- ggplot(data, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'pink')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Life Expec
```

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

```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```



`Poverty Rate` was right skewed therefore I used `log-transformation` on it. Poverty Rate has a negative correlation with life expectancy.

##Probelm-2

From the above plots taking Infant mortality rate with log transformation was obvious choice so I used it to build the model

```
lin_reg <- lm(formula = data$life_expectancy_years ~ log10(data$infant_mortality_rate_per_1000_births))

summary(lin_reg)
```

```
##
## Call:
## lm(formula = data$life_expectancy_years ~ log10(data$infant_mortality_rate_per_1000_births))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7098  -1.2938   0.1379   1.3695   5.9011
##
## Coefficients:
##                                                    Estimate Std. Error t value
## (Intercept)                                         85.5675     0.2184  391.71
## log10(data$infant_mortality_rate_per_1000_births) -11.0752     0.2477  -44.71
##                                                   Pr(>|t|)
## (Intercept)                                         <2e-16 ***
## log10(data$infant_mortality_rate_per_1000_births)   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

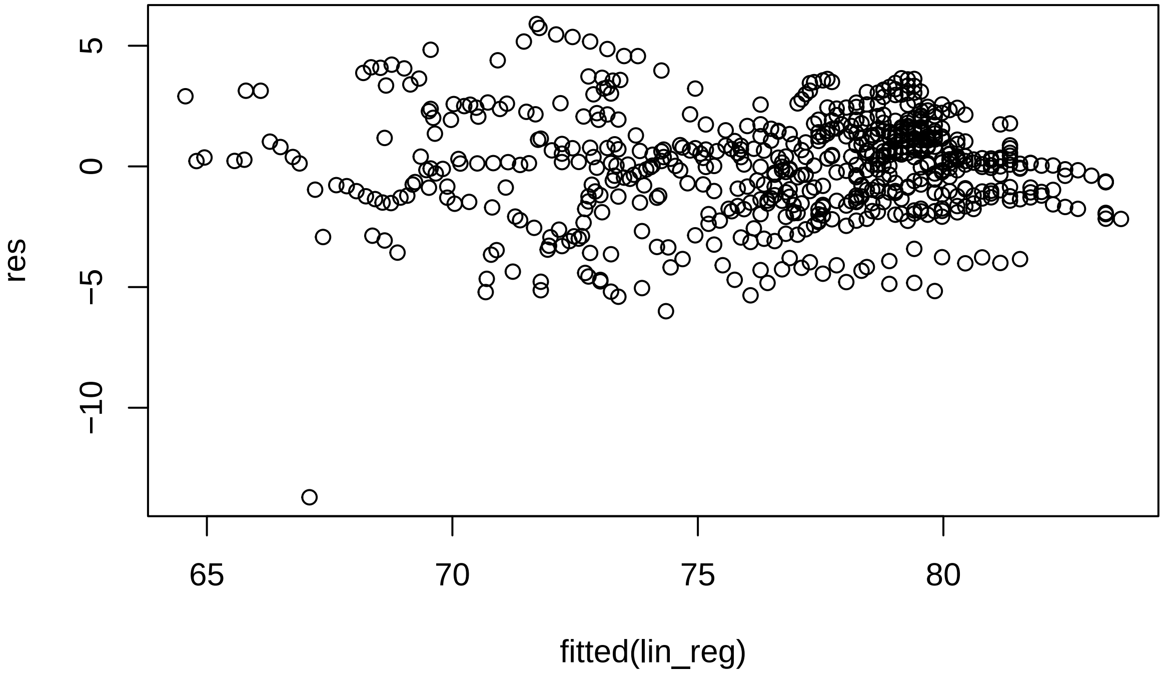
```
##
## Residual standard error: 2.178 on 626 degrees of freedom
## Multiple R-squared:  0.7615, Adjusted R-squared:  0.7611
## F-statistic:  1999 on 1 and 626 DF,  p-value: < 2.2e-16
```

```
res <- resid(lin_reg)
```

I got `R-squared` error of 0.7615 using this predictor

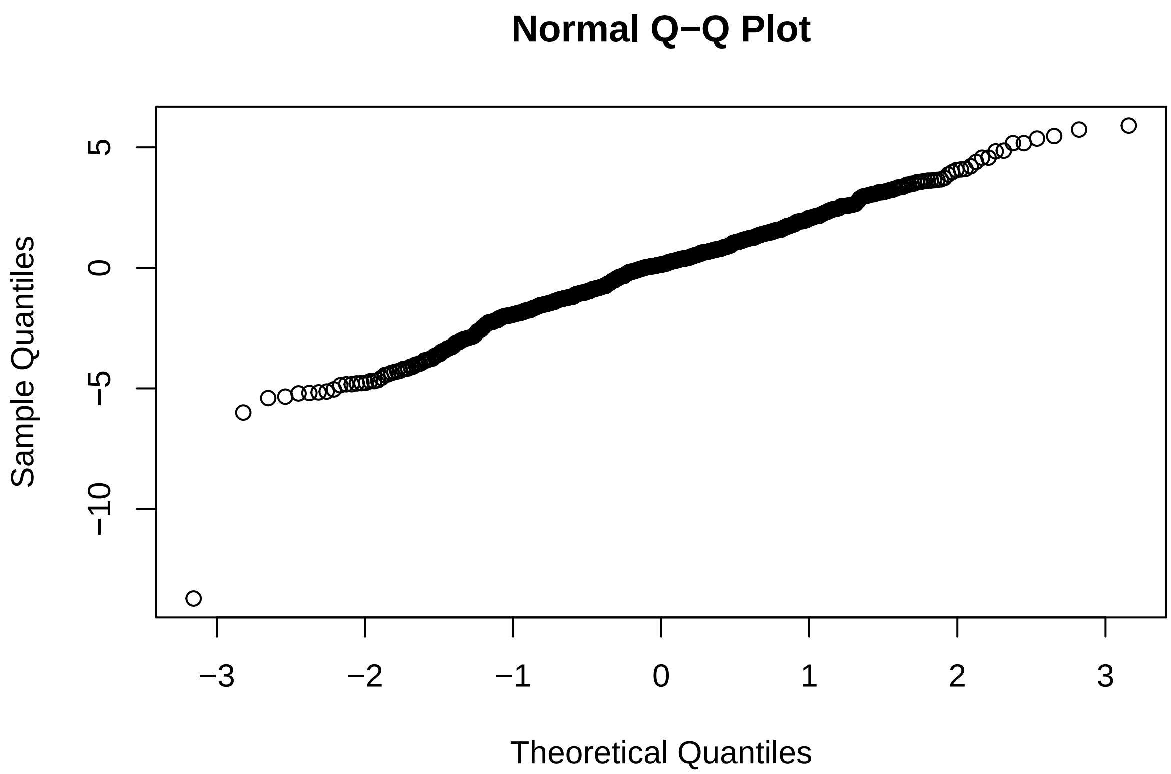
Now lets visualize the residual to see if we did it correctly First we will compare produced residual v/s the fitted plot

```
plot(fitted(lin_reg), res)
```



Lets also look at the Q-Q plot

```
qqnorm(res)
```



From the above plot we can see that in mortality we get one outlier so lets remove that outlier and check again.

```
data_res <- data %>%
  add_residuals(lin_reg,"resid")

dataNoOutliers <- data_res %>% filter(resid>(-10))
```

Now lets re run the whole regression with mortality feature again and see the change in scores

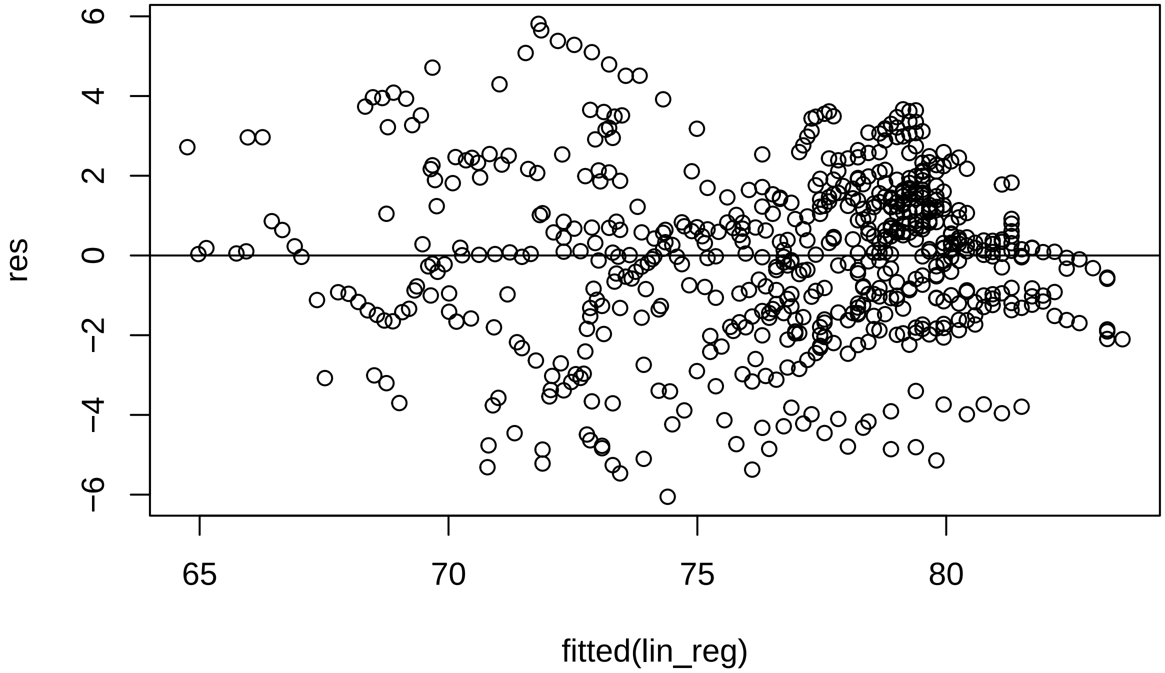
```
lin_reg <- lm(formula = dataNoOutliers$life_expectancy ~ log10(dataNoOutliers$infant_mortality_rate))


summary(lin_reg)
```

```
##
## Call:
## lm(formula = dataNoOutliers$life_expectancy ~ log10(dataNoOutliers$infant_mortality_rate))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.054 -1.317  0.105  1.385  5.811
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                   85.4648     0.2121  402.97
## log10(dataNoOutliers$infant_mortality_rate) -10.9210     0.2410  -45.31
##                                             Pr(>|t|)
## (Intercept)                                   <2e-16 ***
## log10(dataNoOutliers$infant_mortality_rate)   <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
```

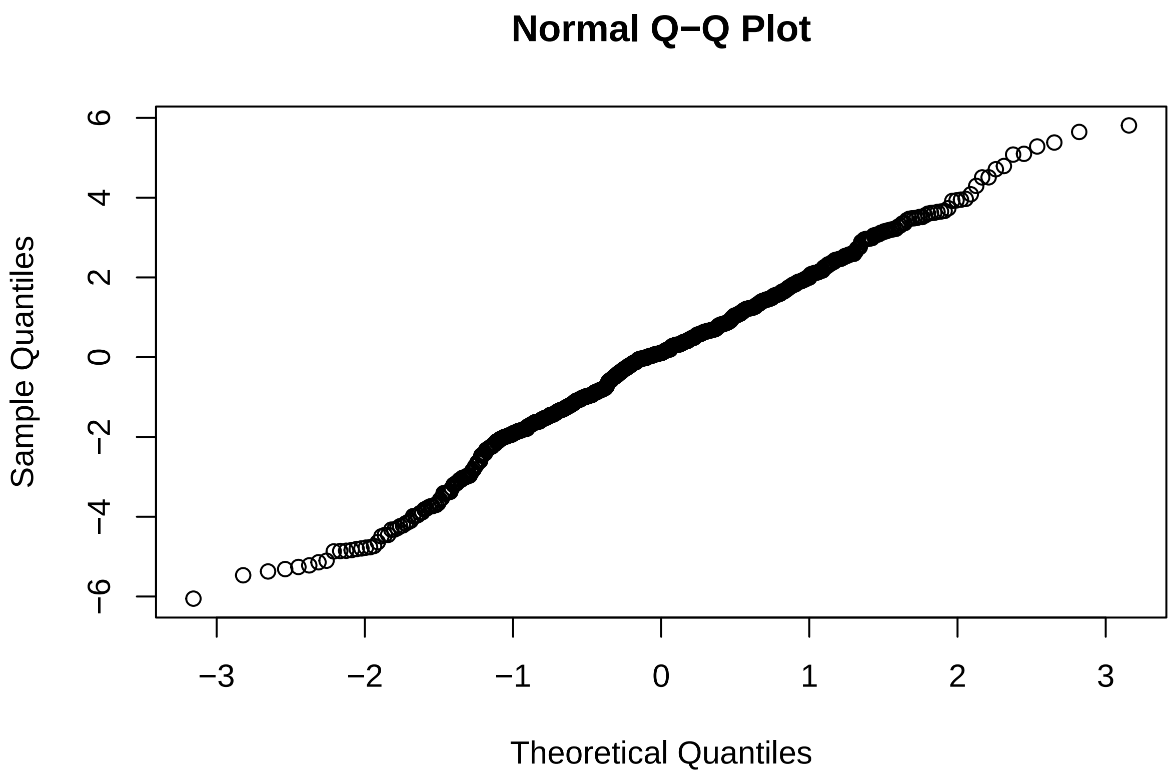
As we can see from the model summary we get better `R-squared` value now lets do the model diagonosis again,

```
res <- resid(lin_reg)

plot(fitted(lin_reg),res)
abline(0,0)
```



Much better now lets also look at the Q-Q plot

```
qqnorm(res)
```



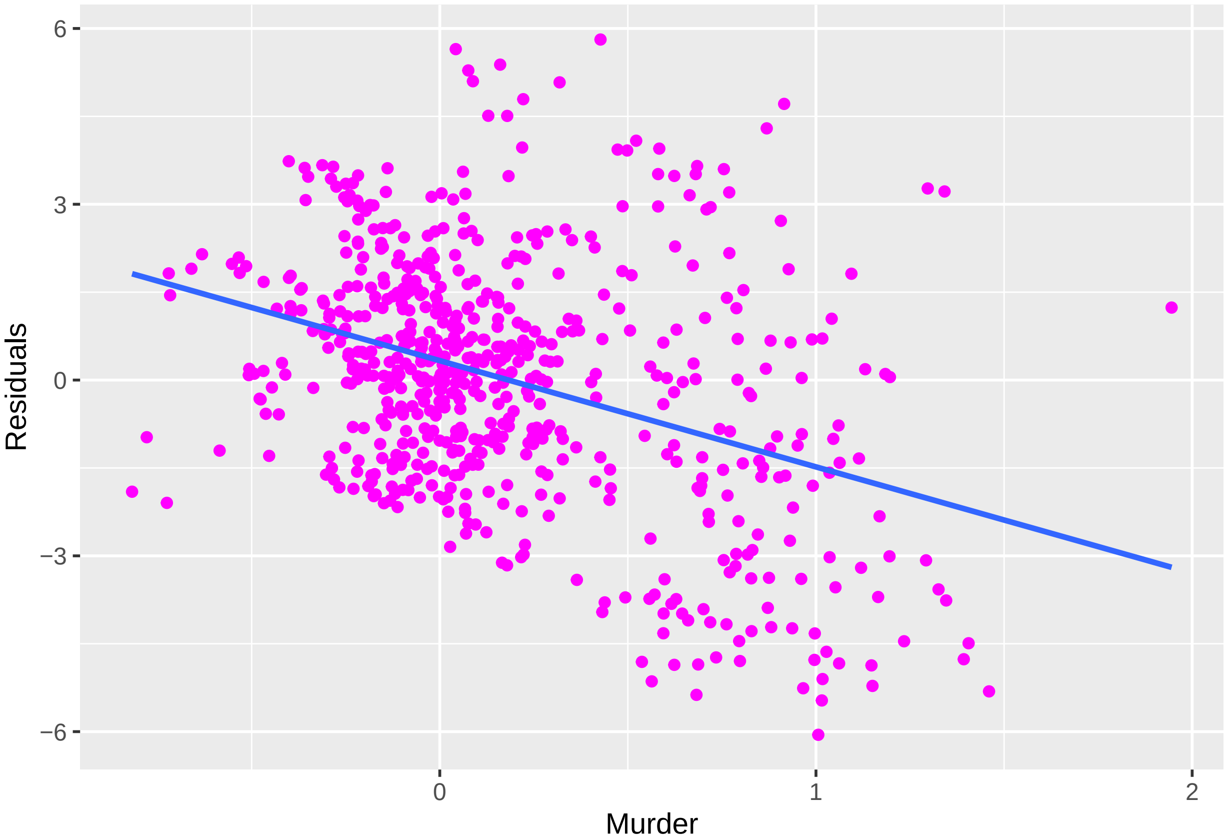
The Q-Q plot is pretty better compared to earlier after model diagnosis

Problem-3

Lets look for relationship between residuals and unused variables to pick another predictor

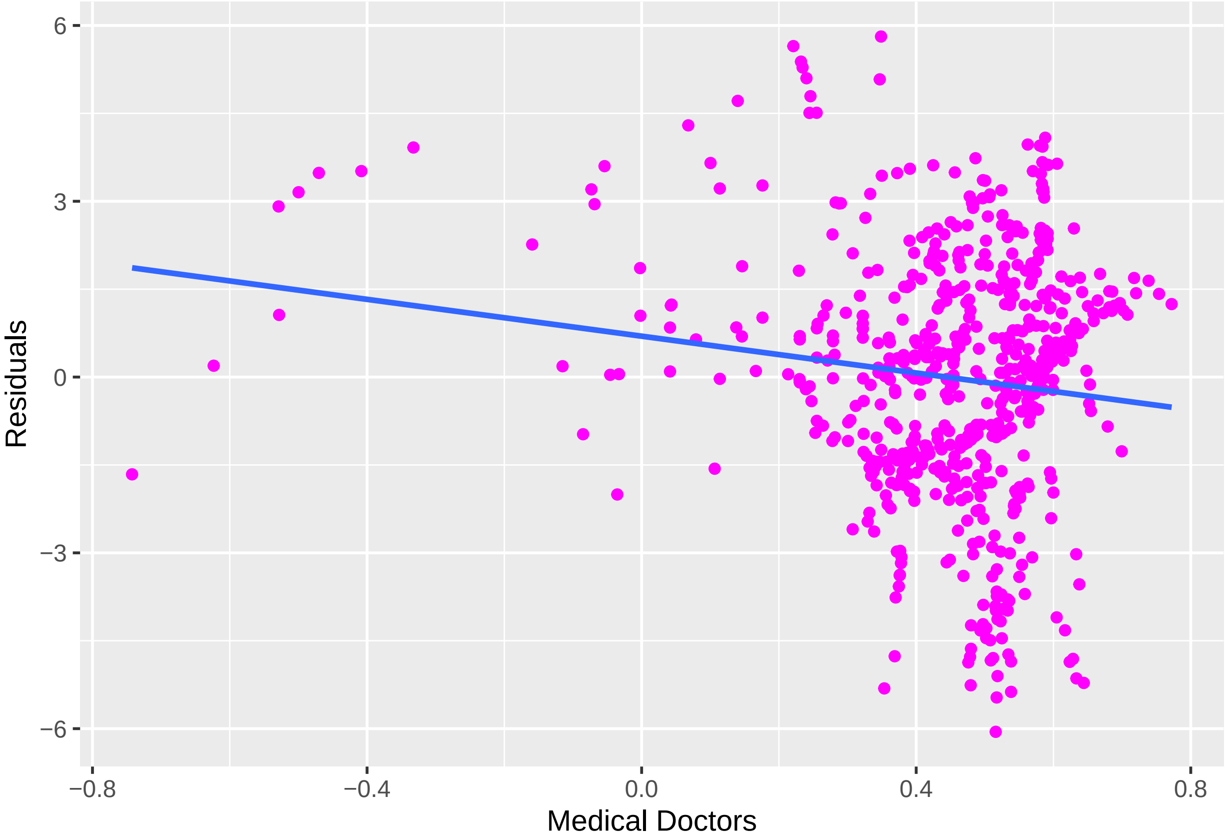
```
x = log10(dataNoOutliers$murder_per_100000_people)
y = res
plt <- ggplot(dataNoOutliers, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'magenta')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Residual
```

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



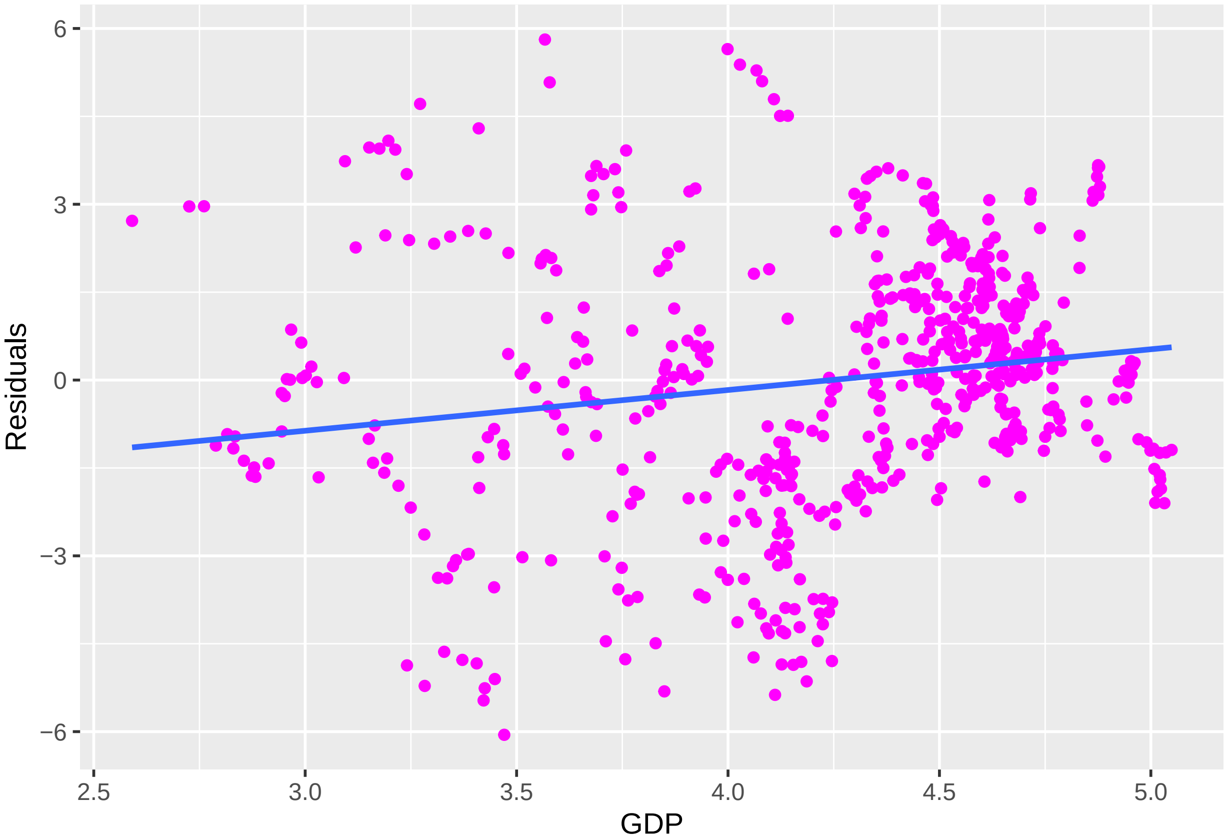
```
x = log10(dataNoOutliers$medical_doctors_per_1000_people)
y = res
plt <- ggplot(dataNoOutliers, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'magenta')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Residual
```

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



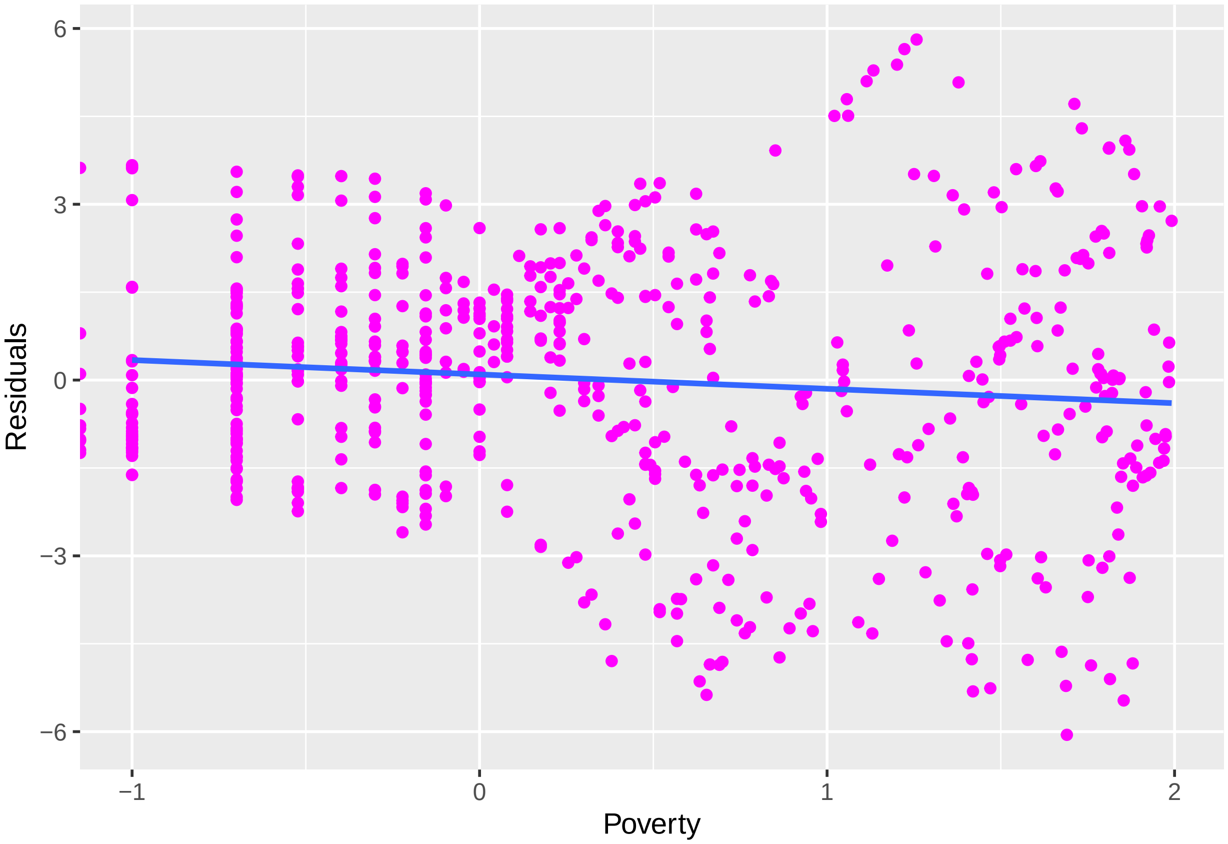
```
x = log10(dataNoOutliers$gdppercapita_us_inflation_adjusted)
y = res
plt <- ggplot(dataNoOutliers, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'magenta')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Residual
```

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



```
x = log10(dataNoOutliers$poverty_percent_people_below_550_a_day)
y = res
plt <- ggplot(dataNoOutliers, aes(x=x,y=y,alpha(0.1)))
plt + geom_point(colour = 'magenta')+ geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+ labs(y="Residual
```

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

## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```



Based on the above plots we can confirm that with murder rate and residuals have highest positive correlation among other features. Therefore we can select murder as our second feature to fit linear regression line.

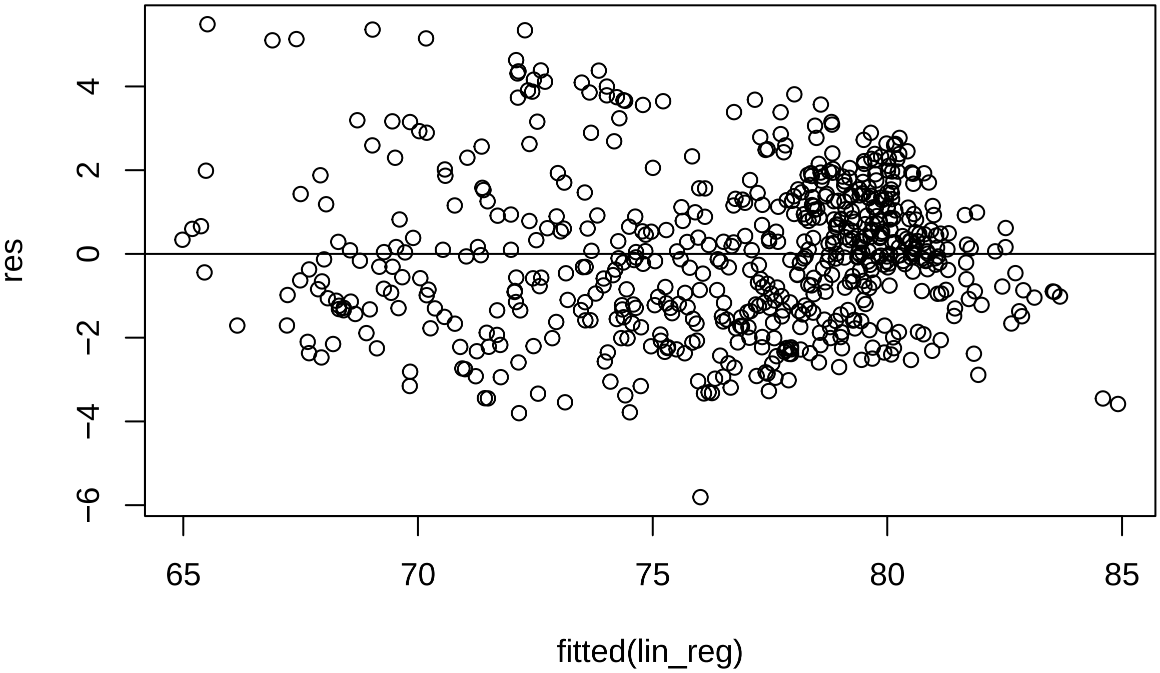
```
lin_reg <- lm(formula = dataNoOutliers$life_expectancy_years ~ log10(dataNoOutliers$infant_mortality_rat
                log10(dataNoOutliers$murder_per_100000_people))

summary(lin_reg)
```

```
##
## Call:
## lm(formula = dataNoOutliers$life_expectancy_years ~ log10(dataNoOutliers$infant_mortality_rate_per_10
##     log10(dataNoOutliers$murder_per_100000_people))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8085 -1.3207 -0.0484  1.1241  5.4854
##
## Coefficients:
##                                                        Estimate Std. Error
## (Intercept)                                             83.6914     0.2165
## log10(dataNoOutliers$infant_mortality_rate_per_1000_births)  -7.9391     0.2859
## log10(dataNoOutliers$murder_per_100000_people)          -3.4762     0.2306
##                                                        t value Pr(>|t|)
## (Intercept)                                             386.62   <2e-16
## log10(dataNoOutliers$infant_mortality_rate_per_1000_births)  -27.77   <2e-16
## log10(dataNoOutliers$murder_per_100000_people)          -15.08   <2e-16
##
## (Intercept)                                            ***
## log10(dataNoOutliers$infant_mortality_rate_per_1000_births) ***
## log10(dataNoOutliers$murder_per_100000_people)              ***
## ---
## 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
```

```
res <- resid(lin_reg)
```

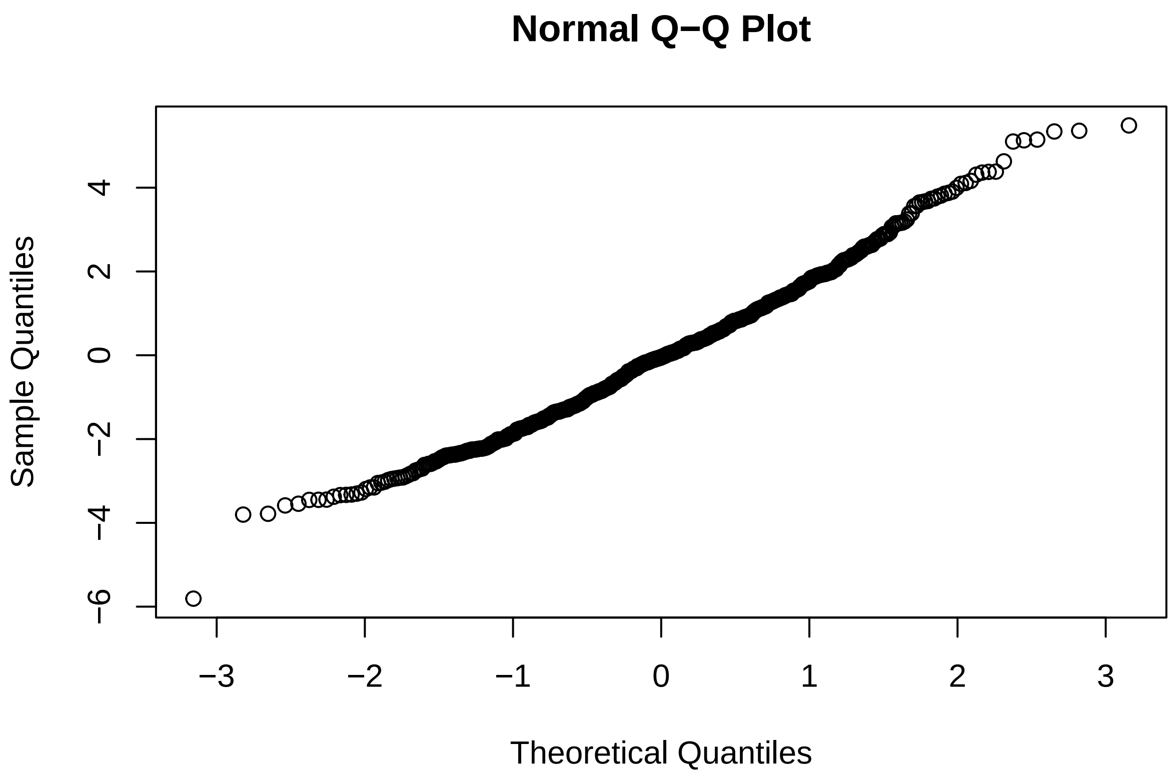
As we can can see the `R-squared` value has also gone up from 0.7671 to 0.8219 after including `Murder` feature along with `infant-mortality-rate` Now lets visualize the residual to see if we did it correctly First we will compare produced residual v/s the fitted plot

```
plot(fitted(lin_reg), res)
abline(0,0)
```



Lets also look at the Q-Q plot

```
qqnorm(res)
```



After model Diagnosis everything looks fine and works with the standard assumption too.

PART B

Problem-4

Now we have to perform k(10) fold cross validation

```
keeps <- c('life_expectancy_years',"infant_mortality_rate_per_1000_births","murder_per_100000_people")
dataNoOutliers_kept = dataNoOutliers[keeps]


dataNoOutliers_ <- resample_partition(dataNoOutliers,
                                    p=c(train=0.8,

                                    test=0.2))






dataNoOutliers_train <- dataNoOutliers[-dataNoOutliers_$test$idx,]

dataNoOutliers_cv <- crossv_kfold(dataNoOutliers_kept, k=10)
```

We get our dataset for cross validation.

```
dataNoOutliers_cv
```

```
## # A tibble: 10 x 3
##    train                test                .id
##    <named list>         <named list>        <chr>
##  1 <resample [564 x 3]> <resample [63 x 3]> 01
##  2 <resample [564 x 3]> <resample [63 x 3]> 02
##  3 <resample [564 x 3]> <resample [63 x 3]> 03
##  4 <resample [564 x 3]> <resample [63 x 3]> 04
##  5 <resample [564 x 3]> <resample [63 x 3]> 05
##  6 <resample [564 x 3]> <resample [63 x 3]> 06
##  7 <resample [564 x 3]> <resample [63 x 3]> 07
##  8 <resample [565 x 3]> <resample [62 x 3]> 08
##  9 <resample [565 x 3]> <resample [62 x 3]> 09
## 10 <resample [565 x 3]> <resample [62 x 3]> 10
```

the `do_cv` function will create folds and perform linear regression on it and return the RMSE value

```
set.seed(2)
do_cv <- function(formula) {
  dataNoOutliers_cv %>%
    mutate(fit = map(train,
                     ~ lm(formula, data = .)),
         rmse = map2_dbl(fit, test, ~ rmse(.x, .y))) %>%
    summarize(cv_rmse = mean(rmse)) %>%
    pull(cv_rmse)
}

do_cv(life_expectancy_years ~ log10(infant_mortality_rate_per_1000_births)+
                log10(murder_per_100000_people))
```

```
## [1] 1.802958
```

We get cross validation error of 1.806 lets check cross validation of model from Problem-3

```
rmse(lin_reg,dataNoOutliers)
```

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

Compared to RMSE from model 3 the cross validation error was slightly higher this could. Though cross validation error should be less one of the plausible reason for it could be model overfitting and given the length of the dataset is less too.

Problem-5

Lets do step wise selection

The following function performs step wise model selections and gives rmse as the output.

```
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
}
```

Initialize the model to NULL.

```
model <- NULL
```

We see some -Inf values in poverety and when taking log of it and thus removing those as well.

```
# Got inf values of poverty rate when took log of it and thus removing those values first.
dataNoOutliers <- dataNoOutliers %>% filter(log(poverty_percent_people_below_550_a_day)!=-Inf)


dataNoOutliers_ <- resample_partition(dataNoOutliers,
                                    p=c(train=0.5,test=0.25,valid=0.25))

preds <- "1"
cands <- c("log(infant_mortality_rate_per_1000_births)","log(murder_per_100000_people)",
           "log(gdppercapita_us_inflation_adjusted)","medical_doctors_per_1000_people",
           "log(poverty_percent_people_below_550_a_day)")




s1 <- step1("life_expectancy_years", preds, cands, dataNoOutliers_)
model <- c(model, attr(s1, "best"))

s1
```

```
##  log(infant_mortality_rate_per_1000_births)
##                                    2.092334
##               log(murder_per_100000_people)
##                                    2.915392
##     log(gdppercapita_us_inflation_adjusted)
##                                    2.381477
##             medical_doctors_per_1000_people
##                                    4.365193
## log(poverty_percent_people_below_550_a_day)
##                                    2.738942
## attr(,"best")
## log(infant_mortality_rate_per_1000_births)
##                                   2.092334
```

```
preds <- "log(infant_mortality_rate_per_1000_births)"
cands <- c("log(murder_per_100000_people)",
           "log(gdppercapita_us_inflation_adjusted)","medical_doctors_per_1000_people",
           "log(poverty_percent_people_below_550_a_day)")




s1 <- step1("life_expectancy_years", preds, cands, dataNoOutliers_)
model <- c(model, attr(s1, "best"))
s1
```

```
##               log(murder_per_100000_people)
##                                    1.796553
##     log(gdppercapita_us_inflation_adjusted)
##                                    2.028338
##             medical_doctors_per_1000_people
##                                    2.097350
## log(poverty_percent_people_below_550_a_day)
##                                    2.102282
## attr(,"best")
## log(murder_per_100000_people)
##                        1.796553
```

```
preds <- c("log(infant_mortality_rate_per_1000_births)", "log(murder_per_100000_people)")
cands <- c(
           "log(gdppercapita_us_inflation_adjusted)","medical_doctors_per_1000_people",
           "log(poverty_percent_people_below_550_a_day)")



s1 <- step1("life_expectancy_years", preds, cands, dataNoOutliers_)
model <- c(model, attr(s1, "best"))
s1
```

```
##     log(gdppercapita_us_inflation_adjusted)
##                                    1.762101
##             medical_doctors_per_1000_people
##                                    1.814000
## log(poverty_percent_people_below_550_a_day)
##                                    1.809307
## attr(,"best")
## log(gdppercapita_us_inflation_adjusted)
##                                1.762101
```

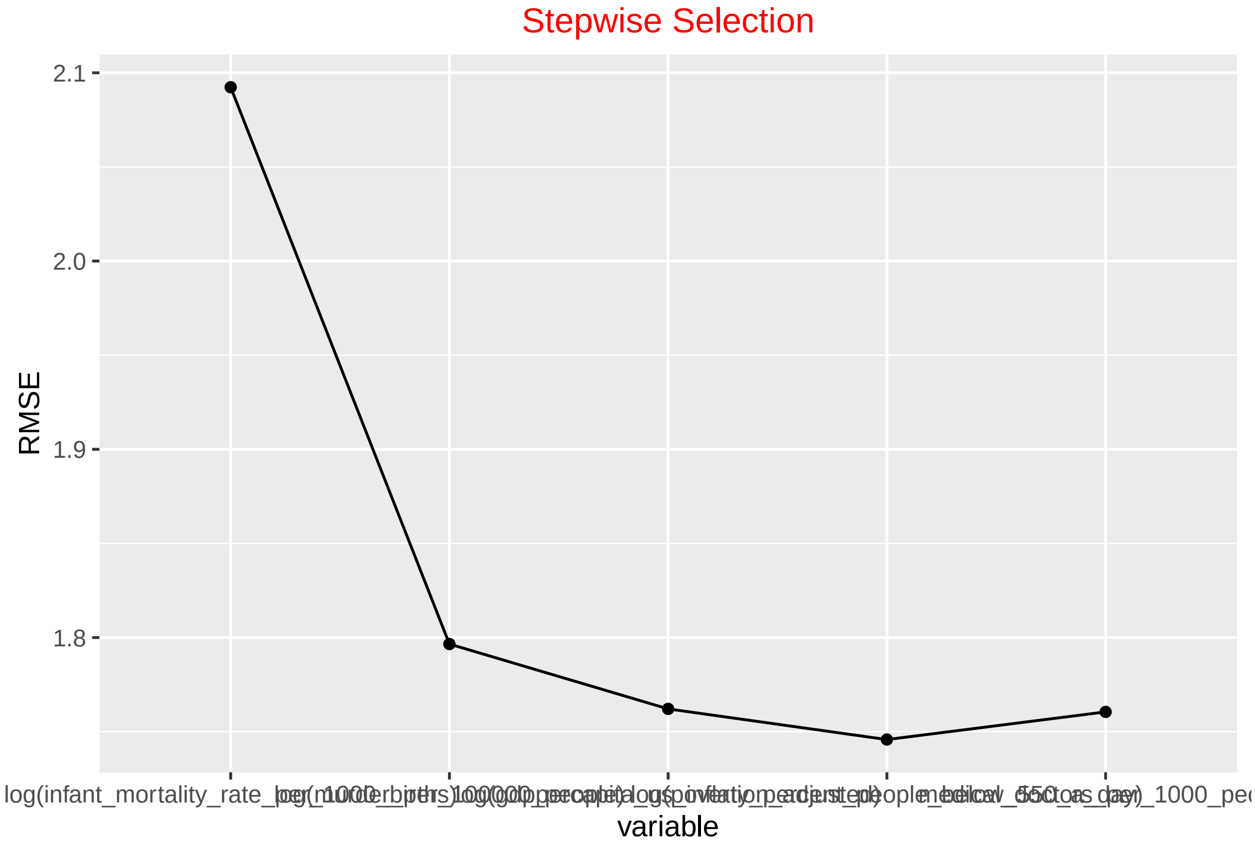
```
preds <- c("log(infant_mortality_rate_per_1000_births)", "log(murder_per_100000_people)","log(gdppercap
cands <- c("medical_doctors_per_1000_people",
           "log(poverty_percent_people_below_550_a_day)")

s1 <- step1("life_expectancy_years", preds, cands, dataNoOutliers_)
model <- c(model, attr(s1, "best"))
s1
```

```
##             medical_doctors_per_1000_people
##                                    1.774315
## log(poverty_percent_people_below_550_a_day)
##                                    1.745803
## attr(,"best")
## log(poverty_percent_people_below_550_a_day)
##                                    1.745803
```

```
preds <- c("log(infant_mortality_rate_per_1000_births)", "log(murder_per_100000_people)","log(gdppercap
"log(poverty_percent_people_below_550_a_day)")
cands <- c("medical_doctors_per_1000_people")



s1 <- step1("life_expectancy_years", preds, cands, dataNoOutliers_)
model <- c(model, attr(s1, "best"))
s1
```

```
## medical_doctors_per_1000_people
##                        1.760505
## attr(,"best")
## medical_doctors_per_1000_people
##                        1.760505
```

```
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 Selection")+
  theme(plot.title=element_text(hjust=0.5, color="red"))
```



Since adding all five candidate predictors lowers down the rmse and thus taking all five candidate predictors.

```
fit4 <- lm(life_expectancy_years ~ log10(infant_mortality_rate_per_1000_births) + log10(murder_per_10000
```

```
rmse(fit4,data = dataNoOutliers)
```

```
## [1] 1.702172
```

RMSE for our model is lower than our previous model.