Project Final Code

Jiaqi Chen, Yiru Gong, Keming Zhan, Yu Si, Jie Liu

Four assumptions associated with a linear regression:

1.Linearity: The relationship between X and the mean of Y is linear.

2.Homoscedasticity: The variance of residual is the same for any value of X. We can test this assumption later, after fitting the linear model.

3.Independence: Observations are independent of each other.

4.Normality: For any fixed value of X, Y is normally distributed.

Data import and clean

```
data_df=read.csv("./data/cdi.csv") %>%
  mutate(CRM_1000=crimes/pop*1000) %>%  ## add new variable CRM_1000(the crime rate per 1,000 populatio
  dplyr::select(-crimes) %>%
  dplyr::select(-id,-cty,-state) %>%
  mutate(region=as.factor(region))
```

TWO tables of 14 variables

test

test

test

test

test

test

test

test

test

test

test

```
numeric_variable_tb=tibble(Variable=c("area","pop","pop18","pop65","docs","beds","hsgrad","bagrad","pov
                   mean=c(1041,393011,28.6,12.2,988,1459,77.6,21.1,8.72,6.60,18561,7869,57.3),
                   sd=c(1550,601987,4.19,3.99,1790,2289,7.02,7.65,4.66,2.34,4059,12884,27.3)) %>%knitr:

factor_variable_tb=tibble(region=c("1 ( Northeast )","2 ( North central )","3 ( South )","4 ( West )"),
                          Frequency=c(103,108,152,77)) %>% knitr::kable(caption="A Frequency Table")
```

```
numeric_variable_tb
```

Table 1: A Table for Numeric Variables

Variable	mean	sd
area	1041.00	1550.00
pop	393011.00	601987.00
pop18	28.60	4.19
pop65	12.20	3.99
docs	988.00	1790.00
beds	1459.00	2289.00
hsgrad	77.60	7.02
bagrad	21.10	7.65
poverty	8.72	4.66
unemp	6.60	2.34
pcincome	18561.00	4059.00
totalinc	7869.00	12884.00
CRM_1000	57.30	27.30

```
factor_variable_tb
```

Table 2: A Frequency Table

region	Frequency
1 (Northeast)	103
2 (North central)	108
3 (South)	152
4 (West)	77

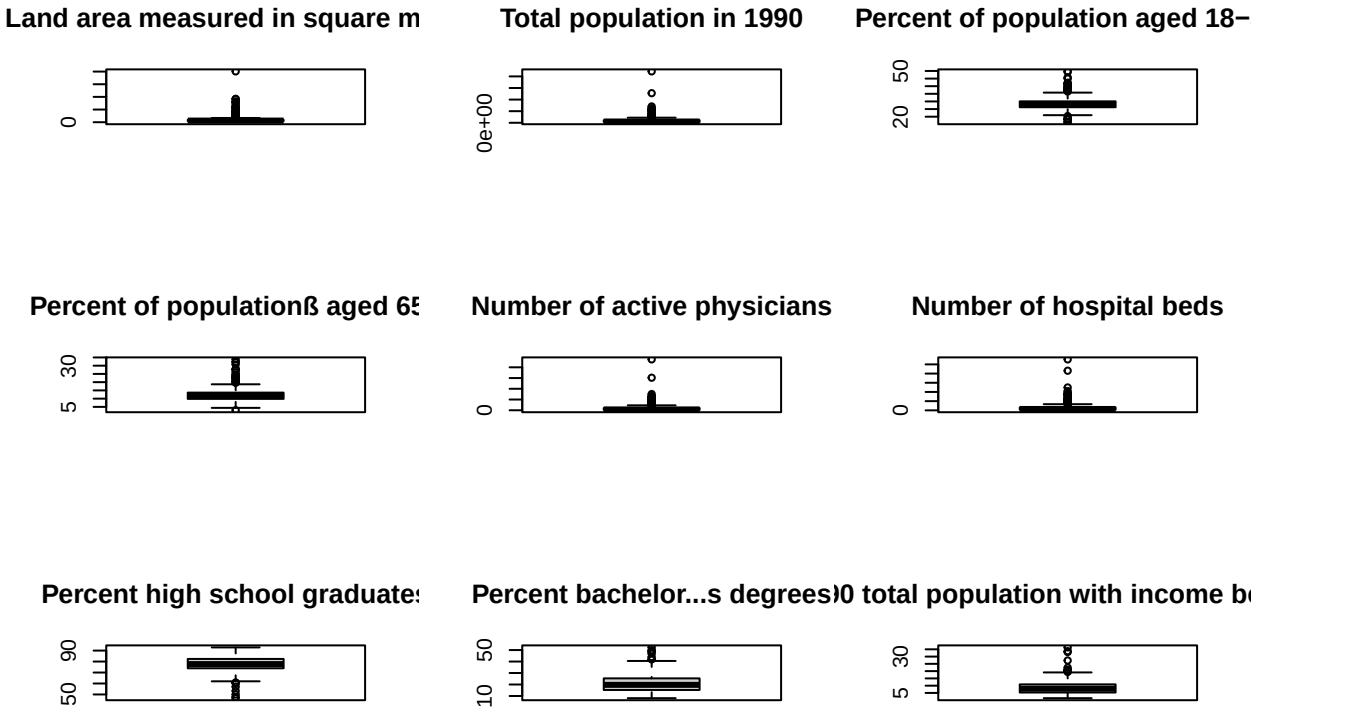
```
## describe(data_df)
```

Checking outliers using boxplots

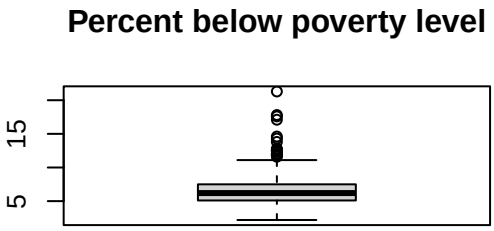
boxplot for each continuous variable

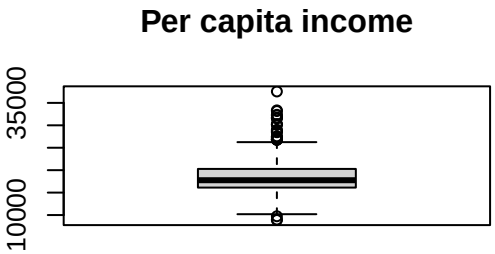
```
par(mfrow=c(3,3))
boxplot(data_df$area,main='Land area measured in square miles')

boxplot(data_df$pop,main='Total population in 1990')

boxplot(data_df$pop18,main='Percent of population aged 18-34')

boxplot(data_df$pop65,main='Percent of populationß aged 65+')

boxplot(data_df$docs,main='Number of active physicians')

boxplot(data_df$beds,main='Number of hospital beds')
```

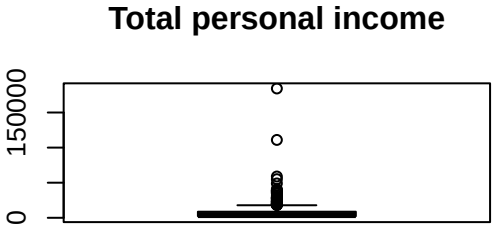
```
boxplot(data_df$hsgrad,main='Percent high school graduates')

boxplot(data_df$bagrad,main='Percent bachelor's degrees')

boxplot(data_df$poverty,main='Percent of 1990 total population with income below poverty level')
```

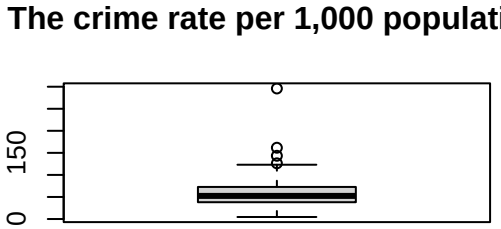



```
par(mfrow=c(2,2))
boxplot(data_df$unemp,main='Percent below poverty level')

boxplot(data_df$pcincome,main='Per capita income')

boxplot(data_df$totalinc,main='Total personal income')

boxplot(data_df$CRM_1000,main='The crime rate per 1,000 population')
```



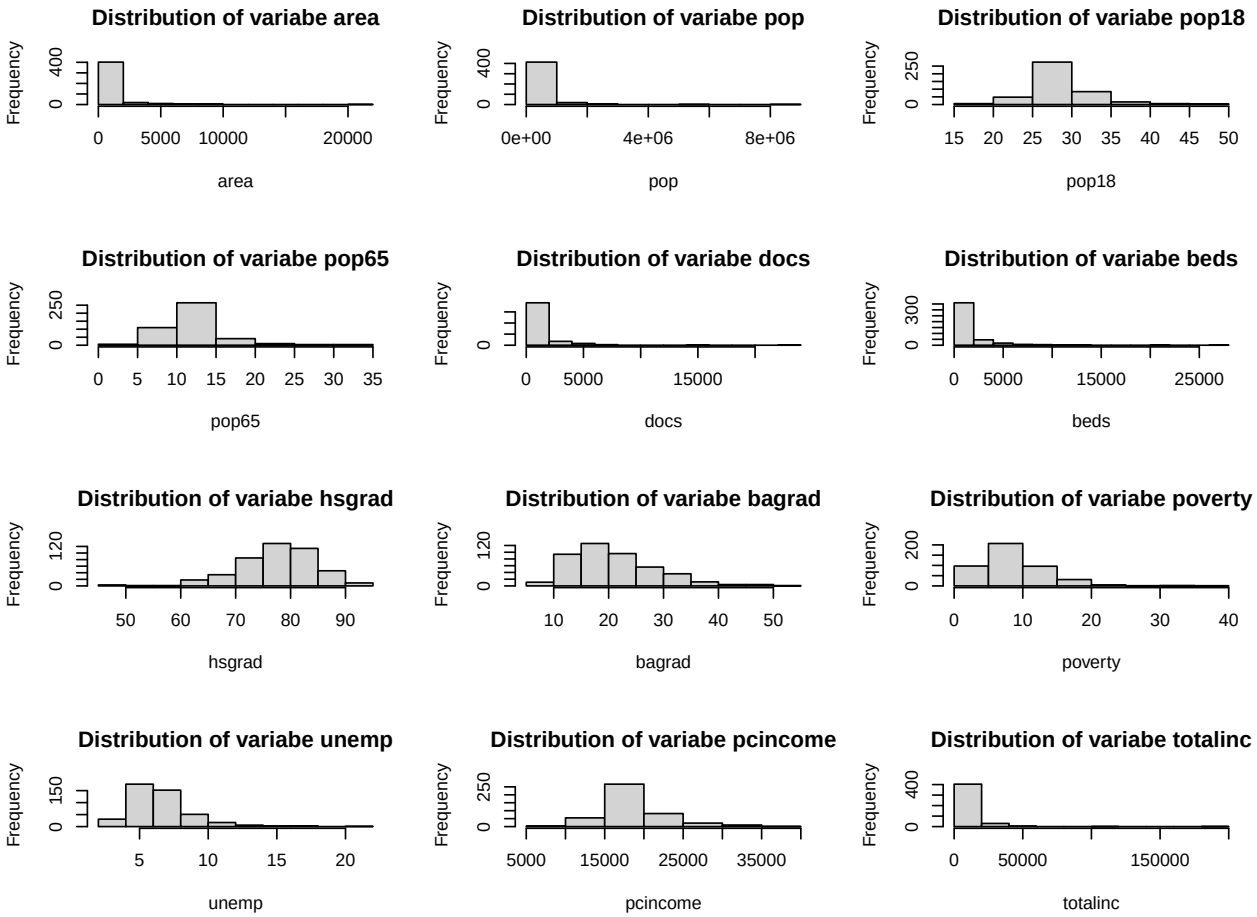






histogram(distribution of 12 independent numeric variables)

```
data_without_region_df=
  data_df %>% dplyr::select(-region,-CRM_1000) ## since histogram cannot be plotted using a factor vari

par(mfrow=c(4,3))

for (n in 1:ncol(data_without_region_df)){
  var = names(data_without_region_df)[n]
  hist(data_without_region_df[,n],xlab = var,main=str_c("Distribution of"," variabe ",var))
}
```



from 12 histograms of each numeric variable, we notice the distribution of 5 histograms are right-skewed, therefore, we need to perform log transformation for these five variables(area,pop,docs,beds and totalinc)

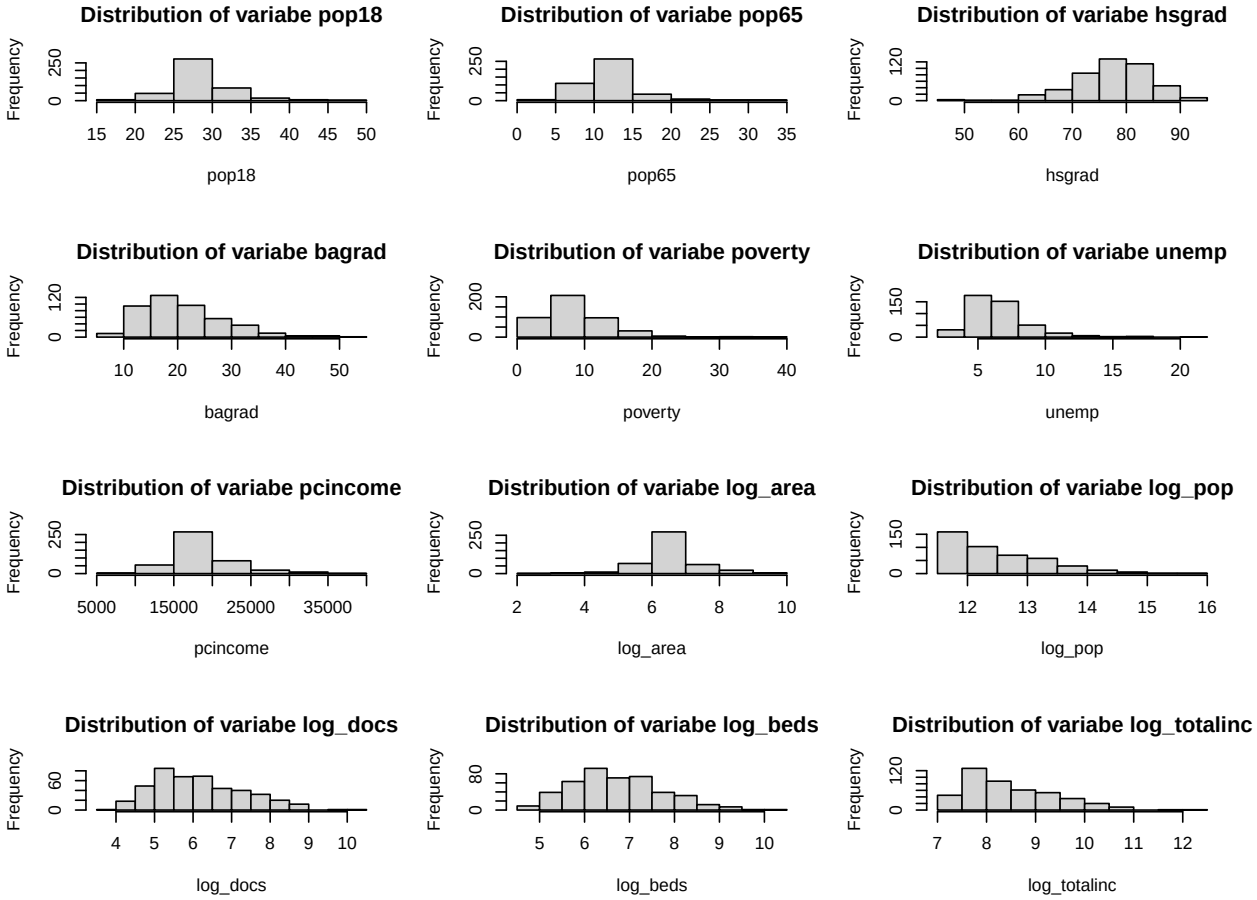
log transformation for five variables whose distribution is right-skewed

```
data_df = data_df %>%
  mutate(log_area = log(area),
         log_pop = log(pop),
         log_docs = log(docs),
         log_beds = log(beds),
         log_totalinc = log(totalinc)
         ) %>%
  dplyr::select(-pop,-area,-docs,-beds,-totalinc)
```

Re-check distribution of independent variables

```
data_without_region_log_CRM1000_df=
  data_df %>% dplyr::select(-region,-CRM_1000)

## histogram after transformation
par(mfrow=c(4,3))
for (n in 1:ncol(data_without_region_log_CRM1000_df)){
  var = names(data_without_region_log_CRM1000_df[n])
  hist(data_without_region_log_CRM1000_df[,n],xlab = var,main=str_c("Distribution of"," variabe ",var))
```

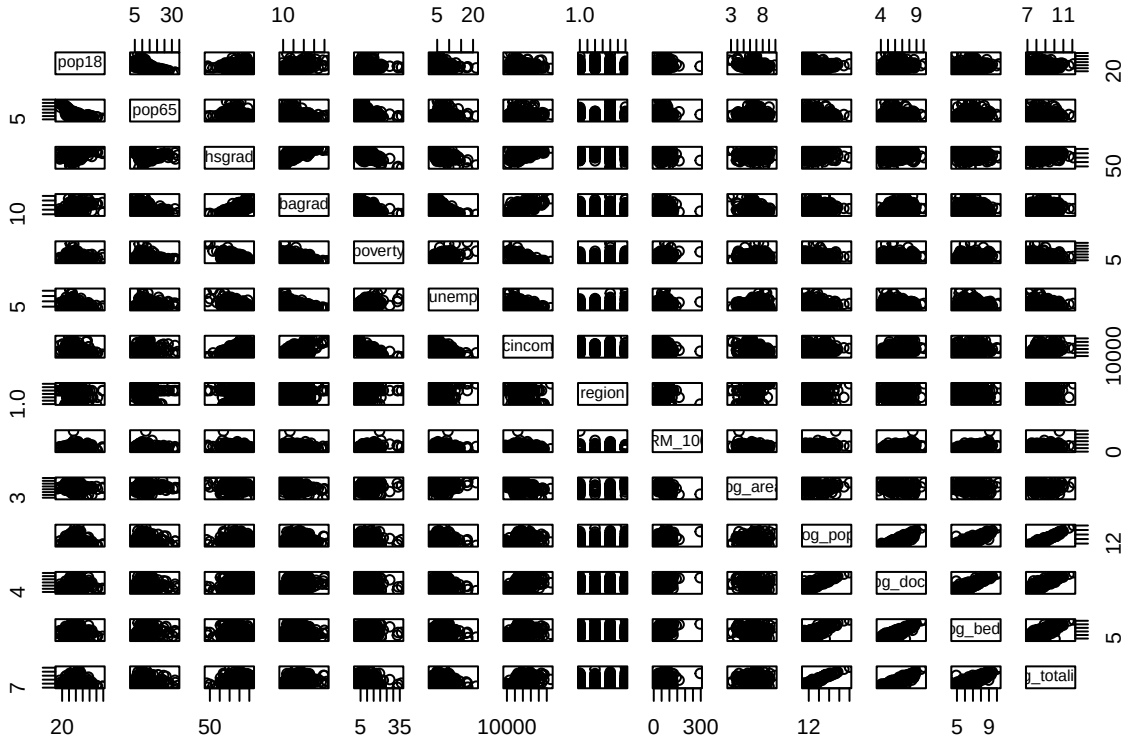
```
}
```



After log transformation, in histograms, we notices except for the variable log_pop, right-skewed problems of four variables mentioned before has been solved.

relationship of two variables(pairs)

```
pair_df=
  data_df

pairs(data_df)
```

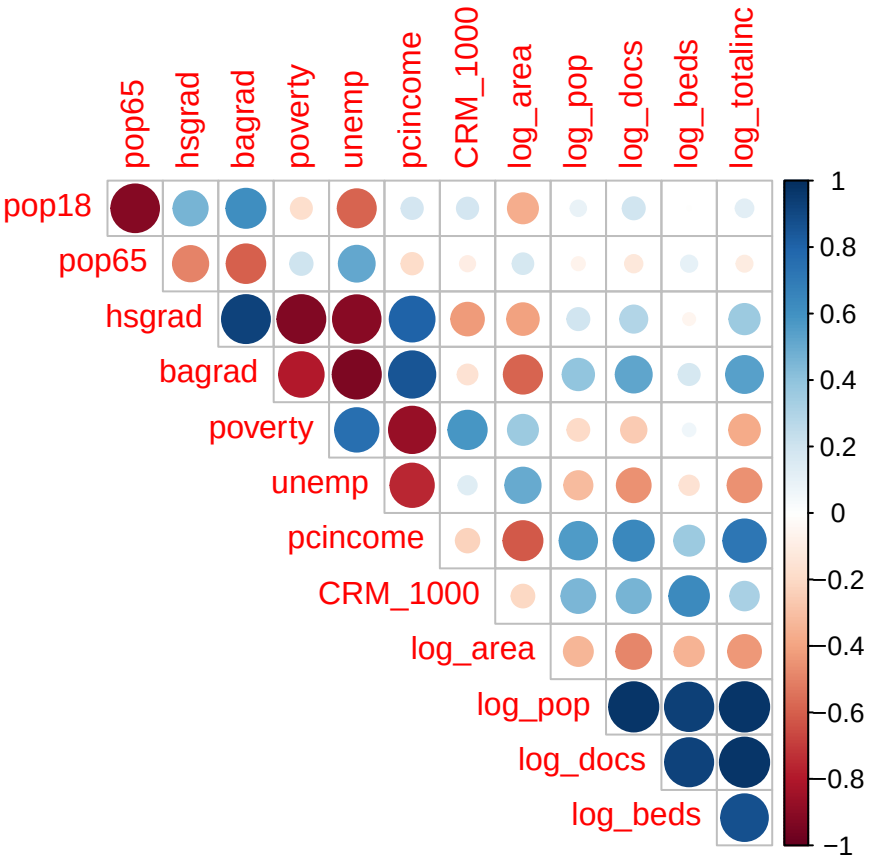


As shown in scatter plots,we can find it seems there is a linear relationship between any two of log_pop,log_docs,log_beds and log_totalinc

Correlation plot

In theory,the correlation between independent variables should be zero.In particular,we expect and are okey with weak to no correlation between independent variables. We also expect that independent variables reflect a high correlation with the target variable.

```
cor_df=
  data_df %>%
  dplyr::select(-region) %>%
  cor() ## Correlation coefficient

corrplot(cor(cor_df), type = "upper", diag = FALSE)
```



from the correlaton plot,we can find independent variables like poverty(0.47),UNEMP(0.42),log_docs(0.443),log_beds(0.493) have a high correlation with the target variable log_CRM_1000. In addition,we also notice independent variables like log_totalinc,log_beds,log_docs and log_pop have a high correlation with each other.

Multicollinearity Assessment

Check collinearity. Stepwise variable selection

ResultL:remove four variables:pop,docs,bed and bagrad

A general rule of thumb for variance inflation factor(VIF): A value of 1 indicates there is no correlation between a given predictor variable and any other predictor variables in the model. A value between 1 and 5 indicates moderate correlation between a given predictor variable and other predictor variables in the model, but this is often not severe enough to require attention. A value greater than 5 indicates potentially severe correlation between a given predictor variable and other predictor variables in the model. In this case, the coefficient estimates and p-values in the regression output are likely unreliable.

```
vif1 = lm(CRM_1000 ~ ., data = data_df)
summary(vif1)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ ., data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.075  -9.633  -0.584   8.187 186.957
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.045e+02  1.506e+02   4.678 3.90e-06 ***
## pop18         9.177e-01  3.408e-01   2.693  0.00737 **
## pop65         8.990e-02  3.260e-01   0.276  0.78289
## hsgrad        2.029e-01  2.707e-01   0.750  0.45389
## bagrad       -2.901e-01  3.063e-01  -0.947  0.34413
## poverty       2.970e+00  4.473e-01   6.640 9.61e-11 ***
## unemp         4.933e-01  5.459e-01   0.904  0.36665
## pcincome     -6.955e-03  1.470e-03  -4.730 3.06e-06 ***
## region2       8.620e+00  2.783e+00   3.098  0.00208 **
## region3       2.472e+01  2.712e+00   9.117  < 2e-16 ***
## region4       2.132e+01  3.397e+00   6.274 8.67e-10 ***
## log_area     -5.066e+00  1.279e+00  -3.962 8.72e-05 ***
## log_pop      -1.598e+02  3.047e+01  -5.246 2.46e-07 ***
## log_docs      1.765e+00  3.263e+00   0.541  0.58881
## log_beds      4.437e+00  2.920e+00   1.519  0.12942
## log_totalinc  1.639e+02  3.091e+01   5.302 1.85e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.08 on 424 degrees of freedom
## Multiple R-squared:  0.5772, Adjusted R-squared:  0.5623
## F-statistic: 38.59 on 15 and 424 DF,  p-value: < 2.2e-16
```

```
vif(vif1) ##  log_totalinc has the highest VIF
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## pop18          2.739703  1        1.655205
## pop65          2.275732  1        1.508553
## hsgrad         4.841235  1        2.200281
## bagrad         7.382145  1        2.717010
## poverty        5.827130  1        2.413945
## unemp          2.187204  1        1.478920
## pcincome      47.844921  1        6.917002
## region         2.843839  3        1.190285
## log_area       1.668359  1        1.291650
## log_pop      778.687397  1       27.904971
## log_docs      18.719261  1        4.326576
## log_beds      11.530197  1        3.395614
## log_totalinc 1033.935497  1       32.154867
```

```
vif2=lm(CRM_1000 ~.-log_totalinc, data = data_df)
summary(vif2)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ . - log_totalinc, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.022  -9.730  -0.881   8.176 192.601
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -6.523e+01  4.127e+01  -1.581 0.114680
## pop18        7.203e-01  3.494e-01   2.062 0.039854 *
## pop65       -1.732e-01  3.324e-01  -0.521 0.602454
## hsgrad       2.765e-01  2.788e-01   0.992 0.321865
## bagrad      -4.246e-01  3.148e-01  -1.349 0.178205
## poverty      1.882e+00  4.099e-01   4.591 5.81e-06 ***
## unemp        6.167e-01  5.625e-01   1.096 0.273562
## pcincome     3.463e-04  5.314e-04   0.652 0.514909
## region2      9.682e+00  2.862e+00   3.383 0.000785 ***
## region3      2.568e+01  2.791e+00   9.204  < 2e-16 ***
## region4      2.243e+01  3.498e+00   6.412 3.83e-10 ***
## log_area    -5.654e+00  1.314e+00  -4.304 2.09e-05 ***
## log_pop      8.866e-01  3.157e+00   0.281 0.778990
## log_docs     4.204e+00  3.332e+00   1.262 0.207735
## log_beds     7.348e+00  2.958e+00   2.484 0.013380 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.65 on 425 degrees of freedom
## Multiple R-squared:  0.5492, Adjusted R-squared:  0.5344
## F-statistic: 36.98 on 14 and 425 DF,  p-value: < 2.2e-16
```

```
vif(vif2) ## remove log_totalinc and log_docs has the highest VIF
```

```
##               GVIF Df GVIF^(1/(2*Df))
## pop18     2.707014  1        1.645301
## pop65     2.222990  1        1.490970
## hsgrad    4.828498  1        2.197384
## bagrad    7.331526  1        2.707679
## poverty   4.600138  1        2.144793
## unemp     2.183230  1        1.477576
## pcincome  5.873363  1        2.423502
## region    2.824312  3        1.188919
## log_area  1.655793  1        1.286776
## log_pop   7.861961  1        2.803919
## log_docs 18.347243  1        4.283368
## log_beds 11.122561  1        3.335050
```

```
vif3=lm(CRM_1000 ~.-log_totalinc -log_docs, data = data_df)
vif(vif3) ## remove log_docs and bagrad has the highest VIF
```

```
##              GVIF Df GVIF^(1/(2*Df))
## pop18    2.673817  1        1.635181
## pop65    2.190305  1        1.479968
## hsgrad   4.828462  1        2.197376
## bagrad   6.805289  1        2.608695
## poverty  4.506901  1        2.122946
## unemp    2.139694  1        1.462769
## pcincome 5.666485  1        2.380438
## region   2.608474  3        1.173269
## log_area 1.632840  1        1.277826
## log_pop  5.526997  1        2.350957
## log_beds 6.233818  1        2.496762
```

```
vif4=lm(CRM_1000 ~ .-log_totalinc -log_docs -bagrad , data = data_df)
vif(vif4) ## remove bagrad and log_beds has the highest VIF.
```

```
##              GVIF Df GVIF^(1/(2*Df))
## pop18    1.981367  1        1.407610
## pop65    2.189620  1        1.479737
## hsgrad   3.279035  1        1.810811
## poverty  3.676560  1        1.917436
## unemp    2.043823  1        1.429623
## pcincome 2.717704  1        1.648546
## region   2.412516  3        1.158097
## log_area 1.624597  1        1.274597
## log_pop  5.519025  1        2.349260
## log_beds 6.169514  1        2.483851
```

```
vif5=lm(CRM_1000 ~ .-log_totalinc -log_docs -bagrad -log_beds, data = data_df)
summary(vif5)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ . - log_totalinc - log_docs - bagrad -
##     log_beds, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.629 -10.796  -0.808   9.955 184.391
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.481e+02  2.714e+01  -5.456 8.25e-08 ***
## pop18        8.304e-01  3.010e-01   2.759 0.006042 **
## pop65        5.470e-01  3.018e-01   1.813 0.070586 .
## hsgrad       1.078e-01  2.352e-01   0.458 0.646943
## poverty      2.558e+00  3.313e-01   7.723 8.16e-14 ***
## unemp       -5.156e-02  5.351e-01  -0.096 0.923284
## pcincome     2.739e-04  3.674e-04   0.745 0.456428
## region2      1.058e+01  2.860e+00   3.699 0.000245 ***
## region3      2.321e+01  2.803e+00   8.279 1.60e-15 ***
## region4      1.999e+01  3.462e+00   5.774 1.49e-08 ***
## log_area    -6.258e+00  1.330e+00  -4.704 3.45e-06 ***
## log_pop      1.333e+01  1.386e+00   9.612  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.09 on 428 degrees of freedom
## Multiple R-squared:  0.5242, Adjusted R-squared:  0.512
## F-statistic: 42.87 on 11 and 428 DF,  p-value: < 2.2e-16
```

```
vif(vif5) ## remove log_beds  and remaining variables' VIFs are between 0 to 5 which indicate remaining
```

```
##              GVIF Df GVIF^(1/(2*Df))
## pop18    1.916638  1        1.384427
## pop65    1.748968  1        1.322486
## hsgrad   3.279022  1        1.810807
## poverty  2.866764  1        1.693152
## unemp    1.885453  1        1.373118
## pcincome 2.679128  1        1.636804
## region   2.172856  3        1.138077
```

```
## log_area 1.620121  1         1.272840
## log_pop  1.446547  1         1.202725
```

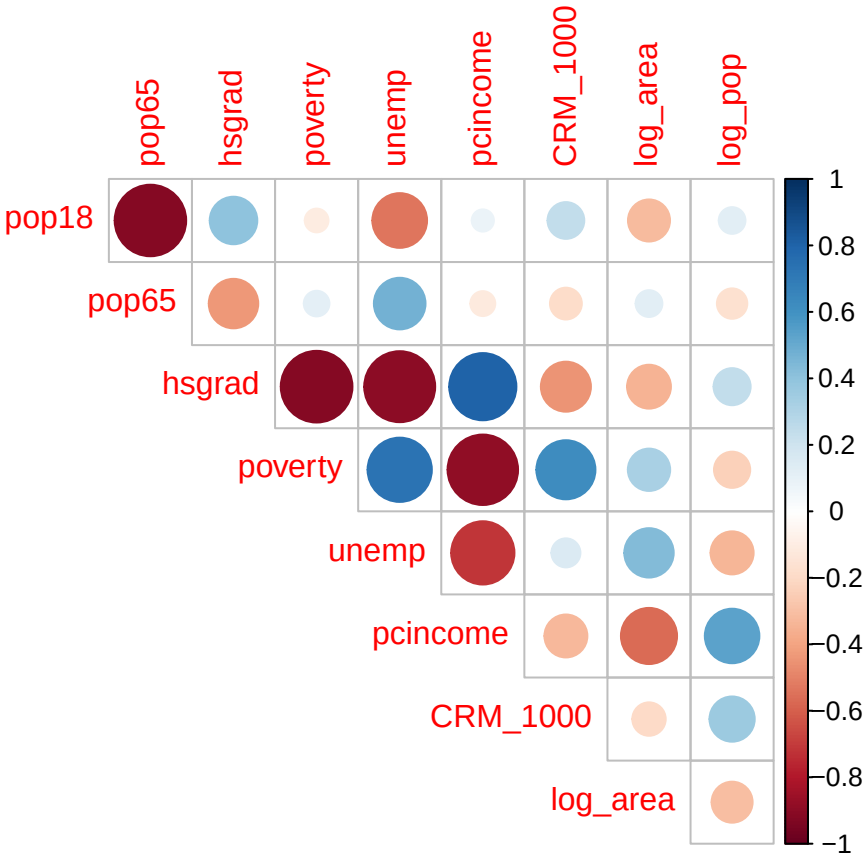
Altogether, We remove four variables(log_totalinc,log_docs,bagrad and log_beds) whose VIFs are higher than 5

```
data_df=data_df %>%
        dplyr::select(-log_totalinc,-log_beds,-log_docs,-bagrad)
```

The remaining 9 variables are pop18, pop65, hsgrad, poverty, unemp, pcincome, region, log_area, log_pop

After log transformation and VIF test, we draw a new correlation plotfrom which we find all absoulte correlation coefficients are less than 0.7,indicating the absence of multicollinearity.

```
cor_df=
  data_df %>%
  dplyr::select(-region) %>%
  cor() ## Correlation coefficient


corrplot(cor(cor_df), type = "upper", diag = FALSE)
```



stepwise model selection

```
mult.fit = lm(CRM_1000 ~ ., data = data_df)

step(mult.fit, direction = 'forward')
```

```
## Start:  AIC=2607.1
## CRM_1000 ~ pop18 + pop65 + hsgrad + poverty + unemp + pcincome +
##     region + log_area + log_pop
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + hsgrad + poverty + unemp +
##     pcincome + region + log_area + log_pop, data = data_df)
##
## Coefficients:
## (Intercept)        pop18        pop65       hsgrad      poverty        unemp
##  -1.481e+02    8.304e-01    5.470e-01    1.078e-01    2.558e+00   -5.156e-02
##    pcincome      region2      region3      region4     log_area      log_pop
##   2.739e-04    1.058e+01    2.321e+01    1.999e+01   -6.258e+00    1.333e+01
```

```
step(mult.fit, direction = 'backward')
```

```
## Start:  AIC=2607.1
## CRM_1000 ~ pop18 + pop65 + hsgrad + poverty + unemp + pcincome +
##     region + log_area + log_pop
##
##             Df Sum of Sq    RSS    AIC
## - unemp      1         3 155980 2605.1
## - hsgrad     1        77 156053 2605.3
## - pcincome   1       202 156179 2605.7
## <none>                   155977 2607.1
## - pop65      1      1197 157174 2608.5
## - pop18      1      2775 158751 2612.9
## - log_area   1      8065 164041 2627.3
## - poverty    1     21734 177710 2662.5
## - region     3     27941 183918 2673.6
## - log_pop    1     33671 189647 2691.1
##
## Step:  AIC=2605.11
## CRM_1000 ~ pop18 + pop65 + hsgrad + poverty + pcincome + region +
##     log_area + log_pop
##
##             Df Sum of Sq    RSS    AIC
## - hsgrad     1       104 156084 2603.4
## - pcincome   1       202 156182 2603.7
## <none>                   155980 2605.1
## - pop65      1      1196 157176 2606.5
## - pop18      1      2820 158800 2611.0
## - log_area   1      8232 164212 2625.7
## - poverty    1     22104 178084 2661.4
## - region     3     30730 186710 2678.2
## - log_pop    1     33725 189705 2689.2
##
## Step:  AIC=2603.41
## CRM_1000 ~ pop18 + pop65 + poverty + pcincome + region + log_area +
##     log_pop
##
##             Df Sum of Sq    RSS    AIC
## - pcincome   1       309 156393 2602.3
## <none>                   156084 2603.4
```

```
## - pop65     1      1138 157222 2604.6
## - pop18     1      3442 159527 2611.0
## - log_area  1      8144 164228 2623.8
## - region    3     31934 188018 2679.3
## - poverty   1     30353 186438 2679.6
## - log_pop   1     33632 189716 2687.3
##
## Step:  AIC=2602.27
## CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop
##
##            Df Sum of Sq    RSS    AIC
## <none>                  156393 2602.3
## - pop65     1      1083 157476 2603.3
## - pop18     1      3242 159635 2609.3
## - log_area  1     10710 167102 2629.4
## - region    3     31748 188141 2677.6
## - poverty   1     43429 199822 2708.1
## - log_pop   1     51043 207436 2724.6
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop, data = data_df)
##
## Coefficients:
## (Intercept)        pop18        pop65      poverty      region2      region3
##   -138.0414       0.8477       0.5171       2.2996      10.6199      23.2782
##     region4     log_area      log_pop
##     20.5980      -6.6625      13.9413
```

```
model = step(mult.fit, direction = 'both')
```

```
## Start:  AIC=2607.1
## CRM_1000 ~ pop18 + pop65 + hsgrad + poverty + unemp + pcincome +
##     region + log_area + log_pop
##
##            Df Sum of Sq    RSS    AIC
## - unemp     1         3 155980 2605.1
## - hsgrad    1        77 156053 2605.3
## - pcincome  1       202 156179 2605.7
## <none>                  155977 2607.1
## - pop65     1      1197 157174 2608.5
## - pop18     1      2775 158751 2612.9
## - log_area  1      8065 164041 2627.3
## - poverty   1     21734 177710 2662.5
## - region    3     27941 183918 2673.6
## - log_pop   1     33671 189647 2691.1
##
## Step:  AIC=2605.11
## CRM_1000 ~ pop18 + pop65 + hsgrad + poverty + pcincome + region +
##     log_area + log_pop
##
##            Df Sum of Sq    RSS    AIC
## - hsgrad    1       104 156084 2603.4
## - pcincome  1       202 156182 2603.7
```

```
## <none>                  155980 2605.1
## - pop65     1      1196 157176 2606.5
## + unemp     1         3 155977 2607.1
## - pop18     1      2820 158800 2611.0
## - log_area  1      8232 164212 2625.7
## - poverty   1     22104 178084 2661.4
## - region    3     30730 186710 2678.2
## - log_pop   1     33725 189705 2689.2
##
## Step:  AIC=2603.41
## CRM_1000 ~ pop18 + pop65 + poverty + pcincome + region + log_area +
##     log_pop
##
##            Df Sum of Sq    RSS    AIC
## - pcincome  1       309 156393 2602.3
## <none>                  156084 2603.4
## - pop65     1      1138 157222 2604.6
## + hsgrad    1       104 155980 2605.1
## + unemp     1        31 156053 2605.3
## - pop18     1      3442 159527 2611.0
## - log_area  1      8144 164228 2623.8
## - region    3     31934 188018 2679.3
## - poverty   1     30353 186438 2679.6
## - log_pop   1     33632 189716 2687.3
##
## Step:  AIC=2602.27
## CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop
##
##            Df Sum of Sq    RSS    AIC
## <none>                  156393 2602.3
## - pop65     1      1083 157476 2603.3
## + pcincome  1       309 156084 2603.4
## + hsgrad    1       211 156182 2603.7
## + unemp     1        51 156342 2604.1
## - pop18     1      3242 159635 2609.3
## - log_area  1     10710 167102 2629.4
## - region    3     31748 188141 2677.6
## - poverty   1     43429 199822 2708.1
## - log_pop   1     51043 207436 2724.6
```

```
broom::tidy(model) %>%
  knitr::kable(digits = c(2,2,2,2,4))
```

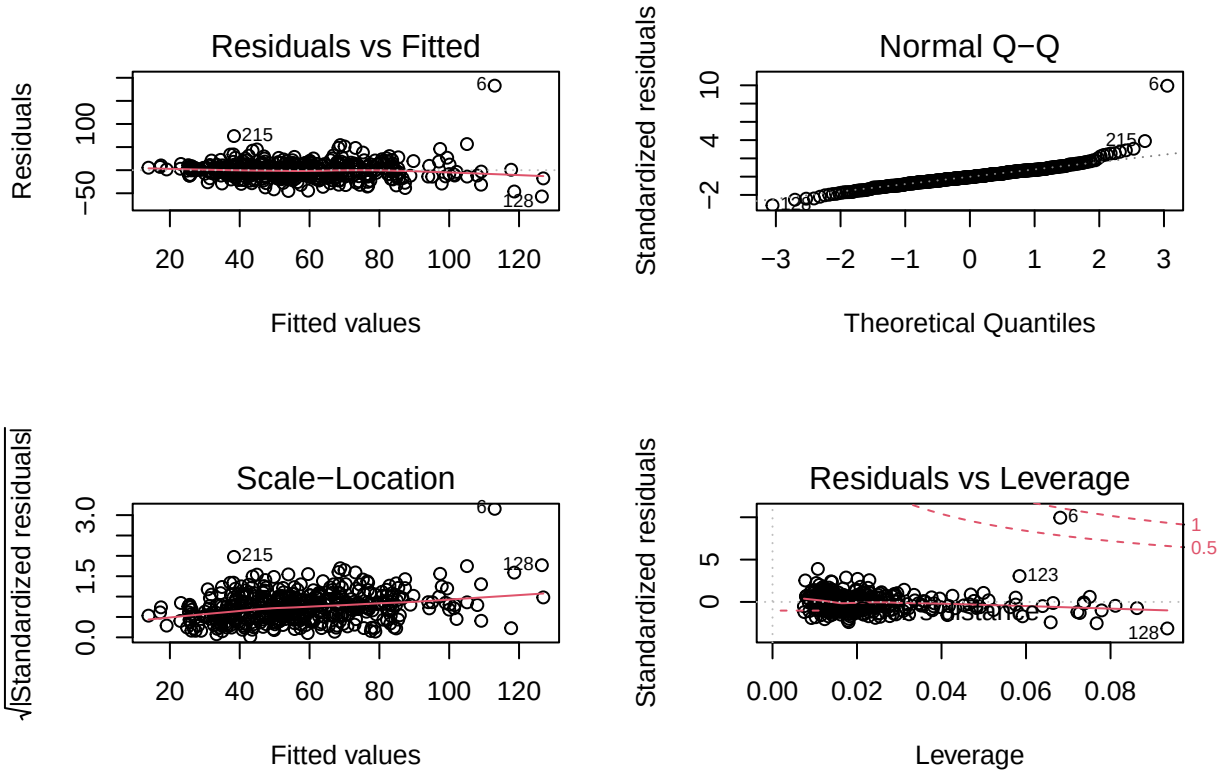
term	estimate	std.error	statistic	p.value
(Intercept)	-138.04	19.80	-6.97	0.0000
pop18	0.85	0.28	2.99	0.0030
pop65	0.52	0.30	1.73	0.0848
poverty	2.30	0.21	10.94	0.0000
region2	10.62	2.70	3.93	0.0001
region3	23.28	2.62	8.89	0.0000
region4	20.60	3.30	6.24	0.0000
log_area	-6.66	1.23	-5.43	0.0000
log_pop	13.94	1.18	11.86	0.0000

Interaction

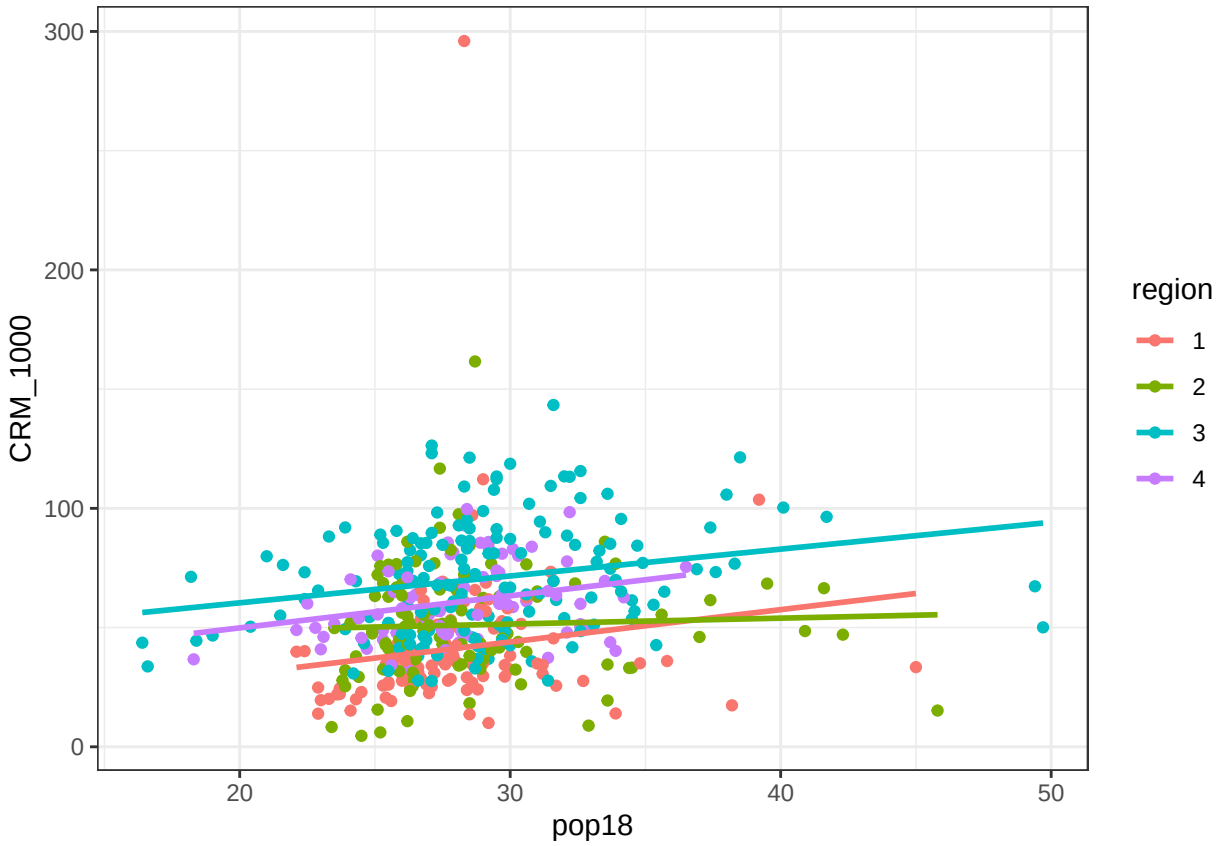
```
# model interaction
model = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop, data = data_df)
summary(model)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.893 -10.478  -0.984  10.070 182.961
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -138.0414    19.8023  -6.971 1.19e-11 ***
## pop18          0.8477     0.2836   2.989  0.00296 **
## pop65          0.5171     0.2993   1.728  0.08475 .
## poverty        2.2996     0.2102  10.940  < 2e-16 ***
## region2       10.6199     2.7047   3.927  0.00010 ***
## region3       23.2782     2.6193   8.887  < 2e-16 ***
## region4       20.5980     3.3023   6.237 1.06e-09 ***
## log_area      -6.6625     1.2264  -5.433 9.30e-08 ***
## log_pop       13.9413     1.1755  11.860  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.05 on 431 degrees of freedom
## Multiple R-squared:  0.523,  Adjusted R-squared:  0.5141
## F-statistic: 59.06 on 8 and 431 DF,  p-value: < 2.2e-16
```

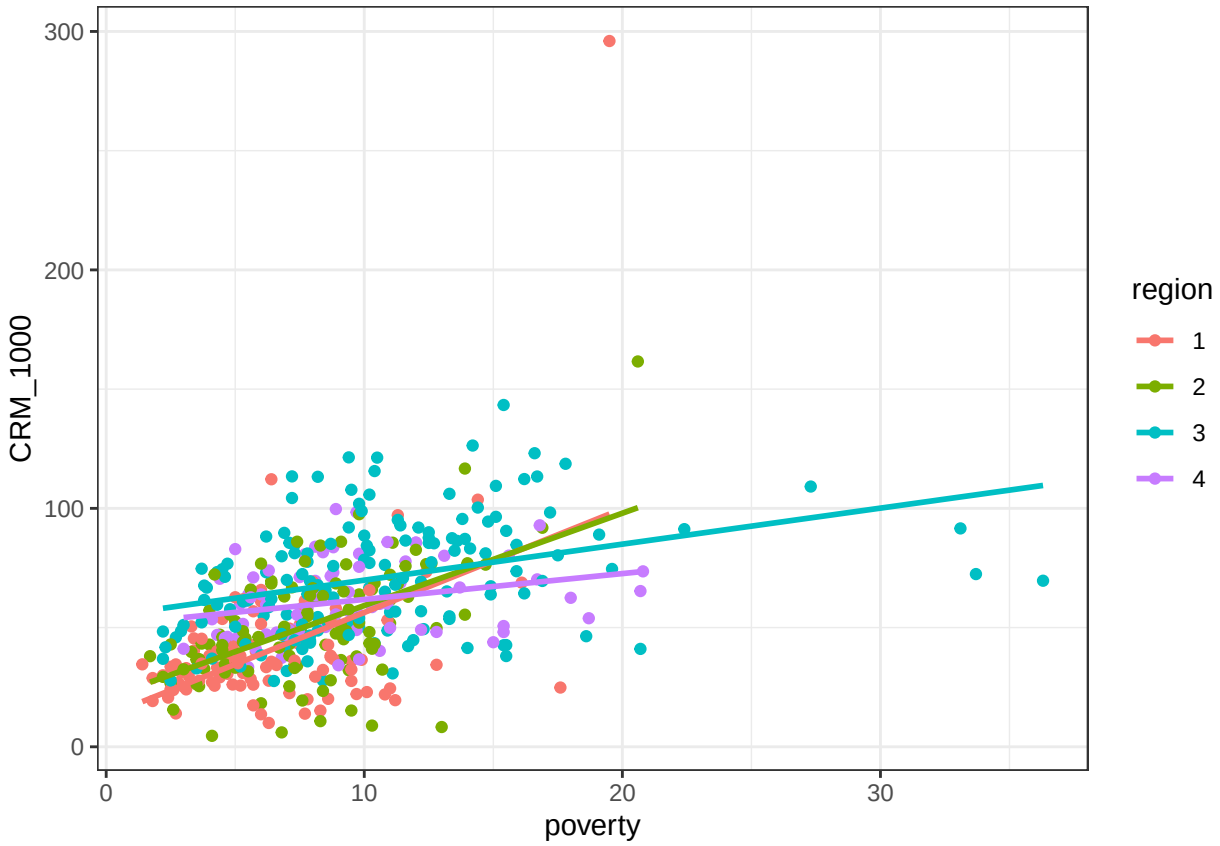
```
par(mfrow=c(2,2)) ## divide the Plots window into the number of rows and columns specified in the brack
plot(model)
```



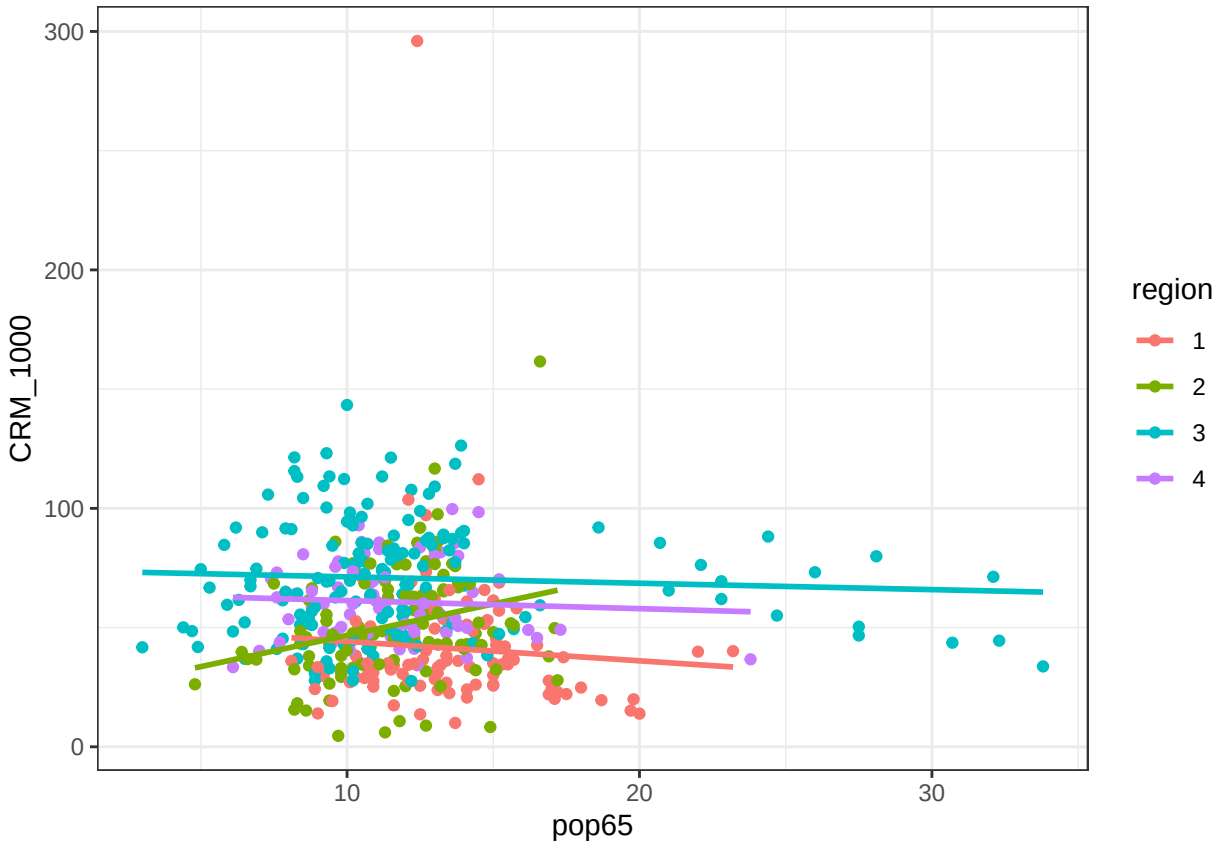

```
par(mfrow=c(1,1))  ## show 4 plots simultaneously.plotting one graph in the entire window, set the para

#1
qplot(x = pop18, y = CRM_1000, color = factor(region), data = data_df) +
  geom_smooth(method = "lm", se=FALSE) +
  theme_bw() +
  labs(x="pop18", y="CRM_1000", color = "region") # not good
```



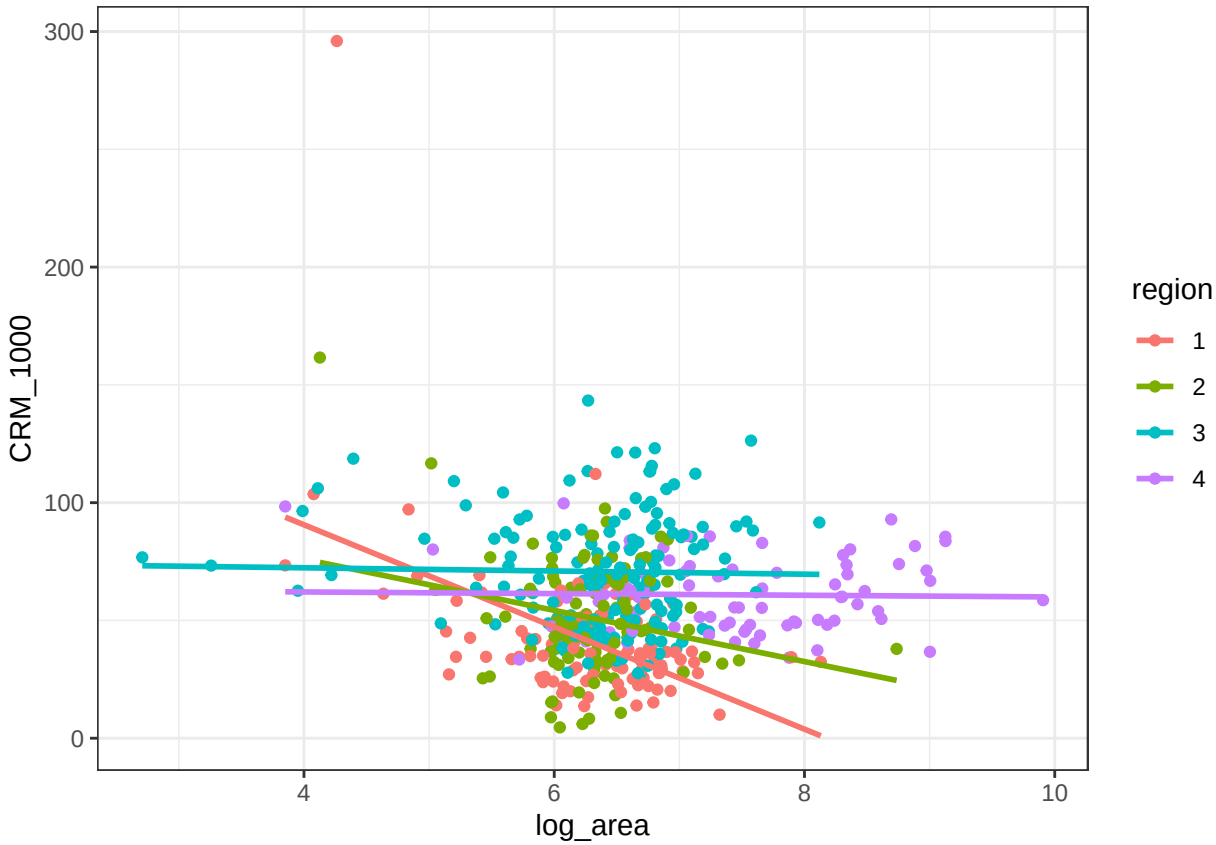

```
#2
qplot(x = poverty, y = CRM_1000, color = factor(region), data = data_df) +
  geom_smooth(method = "lm", se=FALSE) +
  theme_bw() +
  labs(x="poverty", y="CRM_1000", color = "region")
```



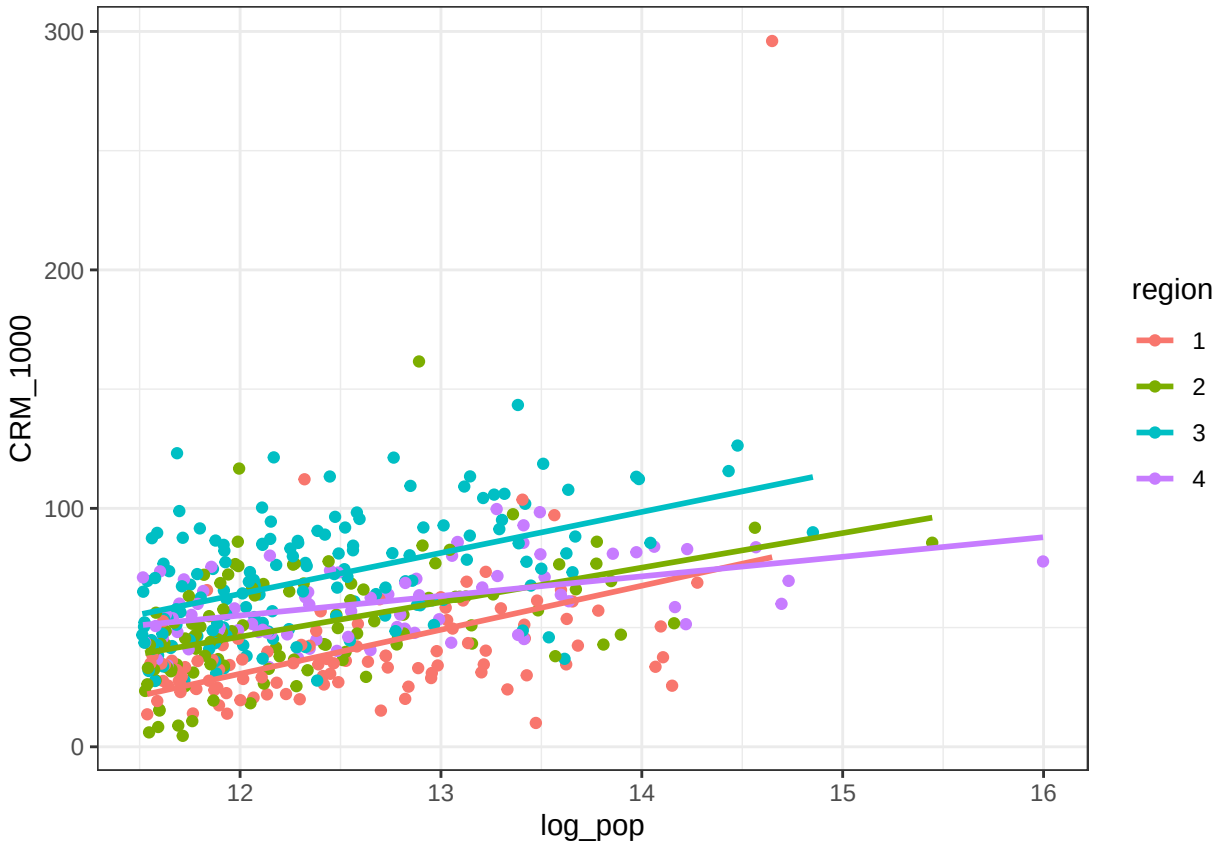

```
#3
qplot(x = pop65, y = CRM_1000, color = factor(region), data = data_df) +
  geom_smooth(method = "lm", se=FALSE) +
  theme_bw() +
  labs(x="pop65", y="CRM_1000", color = "region") #not good
```




```
#4
qplot(x = log_area, y = CRM_1000, color = factor(region), data = data_df) +
  geom_smooth(method = "lm", se=FALSE) +
  theme_bw() +
  labs(x="log_area", y="CRM_1000", color = "region")
```




```
#5
qplot(x = log_pop, y = CRM_1000, color = factor(region), data = data_df) +
  geom_smooth(method = "lm", se=FALSE) +
  theme_bw() +
  labs(x="log_pop", y="CRM_1000", color = "region") #not good
```




```
# model_adj1 = update(model, . ~ . + poverty*region, data=data_df)

model_adj1 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop+ poverty*reg
summary(model_adj1)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * region, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.220 -10.559  -0.369  10.453 163.086
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -143.6838    19.8186  -7.250 1.96e-12 ***
## pop18              0.7427     0.2811   2.642 0.008544 **
## pop65              0.2533     0.3044   0.832 0.405659
## poverty            3.9266     0.5586   7.030 8.22e-12 ***
## region2           14.1447     6.2675   2.257 0.024522 *
## region3           40.1729     5.2324   7.678 1.11e-13 ***
## region4           28.8289     6.7414   4.276 2.34e-05 ***
## log_area          -6.0784     1.2457  -4.880 1.50e-06 ***
## log_pop           13.7737     1.1698  11.774  < 2e-16 ***
## poverty:region2   -0.7942     0.7762  -1.023 0.306794
## poverty:region3   -2.2579     0.6272  -3.600 0.000355 ***
```

```
## poverty:region4   -1.5029     0.7887  -1.906 0.057374 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.78 on 428 degrees of freedom
## Multiple R-squared:  0.5395, Adjusted R-squared:  0.5277
## F-statistic: 45.59 on 11 and 428 DF,  p-value: < 2.2e-16
```

```
anova(model, model_adj1)
```

```
## Analysis of Variance Table
##
## Model 1: CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop
## Model 2: CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +
##     poverty * region
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    431 156393
## 2    428 150957  3    5435.4 5.1369 0.001687 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_adj2 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +log_pop*reg
summary(model_adj2)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + log_pop * region, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.645  -9.975  -1.145   9.659 179.763
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -162.7955    35.4495  -4.592 5.77e-06 ***
## pop18              0.9043     0.2841   3.183  0.00156 **
## pop65              0.5299     0.2978   1.779  0.07587 .
## poverty            2.2677     0.2108  10.758  < 2e-16 ***
## region2           53.8029    43.1083   1.248  0.21268
## region3            1.8302    41.3743   0.044  0.96474
## region4          100.2286    43.7806   2.289  0.02255 *
## log_area          -6.5909     1.2395  -5.318 1.70e-07 ***
## log_pop           15.7509     2.4942   6.315 6.77e-10 ***
## region2:log_pop   -3.4616     3.4526  -1.003  0.31662
## region3:log_pop    1.7653     3.3048   0.534  0.59351
## region4:log_pop   -6.2835     3.4661  -1.813  0.07056 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.95 on 428 degrees of freedom
## Multiple R-squared:  0.5311, Adjusted R-squared:  0.519
## F-statistic: 44.06 on 11 and 428 DF,  p-value: < 2.2e-16
```

```
anova(model, model_adj2)
```

```
## Analysis of Variance Table
##
## Model 1: CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop
## Model 2: CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +
##     log_pop * region
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    431 156393
## 2    428 153738  3    2654.4 2.4632 0.06195 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# model_adj3 = update(model, . ~ . + poverty*log_pop, data=data_df)

model_adj3 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_
summary(model_adj3)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * log_pop, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.175  -9.881  -0.644   9.635 156.188
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.8989    34.5445   0.258 0.796835
## pop18             0.8701     0.2757   3.156 0.001711 **
## pop65             0.5142     0.2909   1.768 0.077792 .
## poverty         -14.2931     3.2457  -4.404 1.34e-05 ***
## region2           9.7723     2.6340   3.710 0.000234 ***
## region3          23.5341     2.5463   9.243  < 2e-16 ***
## region4          20.7983     3.2099   6.479 2.53e-10 ***
## log_area         -5.7992     1.2038  -4.817 2.02e-06 ***
## log_pop           1.7393     2.6419   0.658 0.510674
## poverty:log_pop   1.3228     0.2582   5.122 4.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.51 on 430 degrees of freedom
## Multiple R-squared:  0.5504, Adjusted R-squared:  0.541
## F-statistic: 58.49 on 9 and 430 DF,  p-value: < 2.2e-16
```

```
anova(model, model_adj3)
```

```
## Analysis of Variance Table
##
## Model 1: CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop
## Model 2: CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +
##     poverty * log_pop
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    431 156393
## 2    430 147399  1    8994.3 26.239 4.564e-07 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

leave model, model_adj1, model_adj3

Cross-Validation

```
train = trainControl(method = "cv", number = 5)

model1 = train(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop+ poverty*region,
                   data = data_df,
                   trControl = train,
                   method = 'lm',
                   na.action = na.pass)

model1$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
##       (Intercept)              pop18              pop65            poverty
##         -143.6838             0.7427             0.2533             3.9266
##           region2            region3            region4          log_area
##           14.1447            40.1729            28.8289            -6.0784
##           log_pop  `poverty:region2`  `poverty:region3`  `poverty:region4`
##           13.7737            -0.7942            -2.2579            -1.5029
```

```
print(model1)
```

```
## Linear Regression
##
## 440 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 352, 352, 352, 352, 352
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   19.27728  0.513496  13.7687
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
model3 = train(CRM_1000 ~pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_pop,
                   data = data_df,
                   trControl = train,
                   method = 'lm',
                   na.action = na.pass)

model3$finalModel
```

```
##
## Call:
```

```
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
##       (Intercept)              pop18              pop65            poverty
##            8.8989             0.8701             0.5142           -14.2931
##           region2            region3            region4           log_area
##            9.7723            23.5341            20.7983            -5.7992
##           log_pop  `poverty:log_pop`
##            1.7393             1.3228
```

```
print(model3)
```

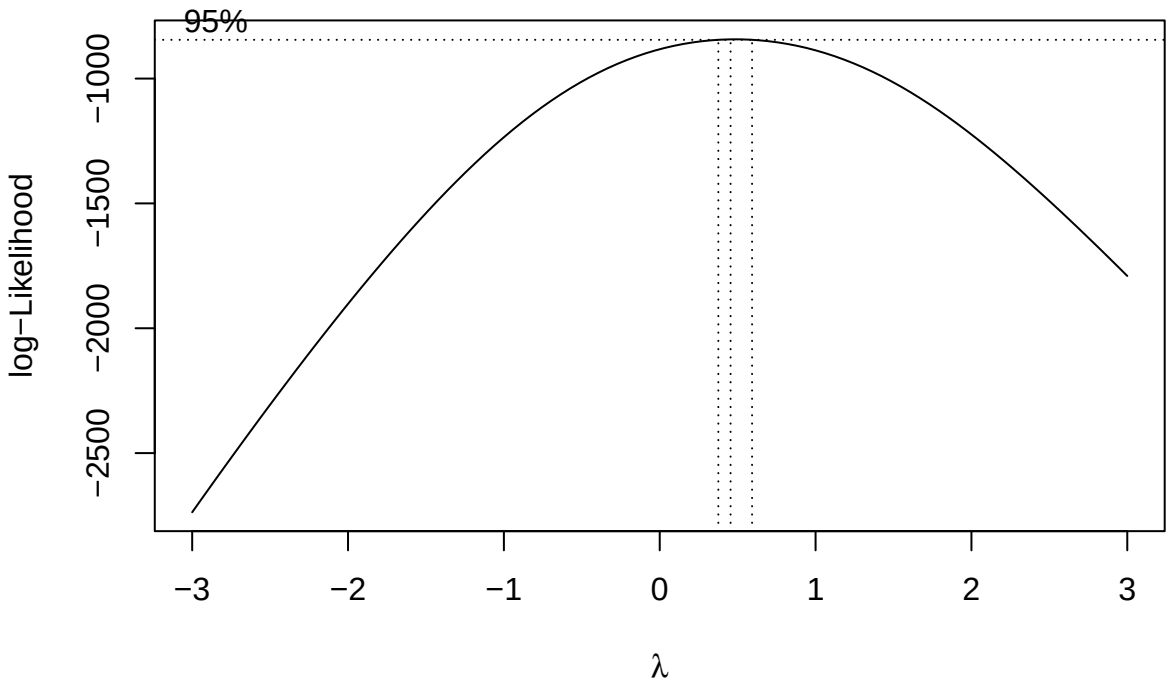
```
## Linear Regression
##
## 440 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 352, 352, 352, 352, 352
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   18.80731  0.5258853  13.2116
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Model3 is selected with lower RMSE and higher R-square.

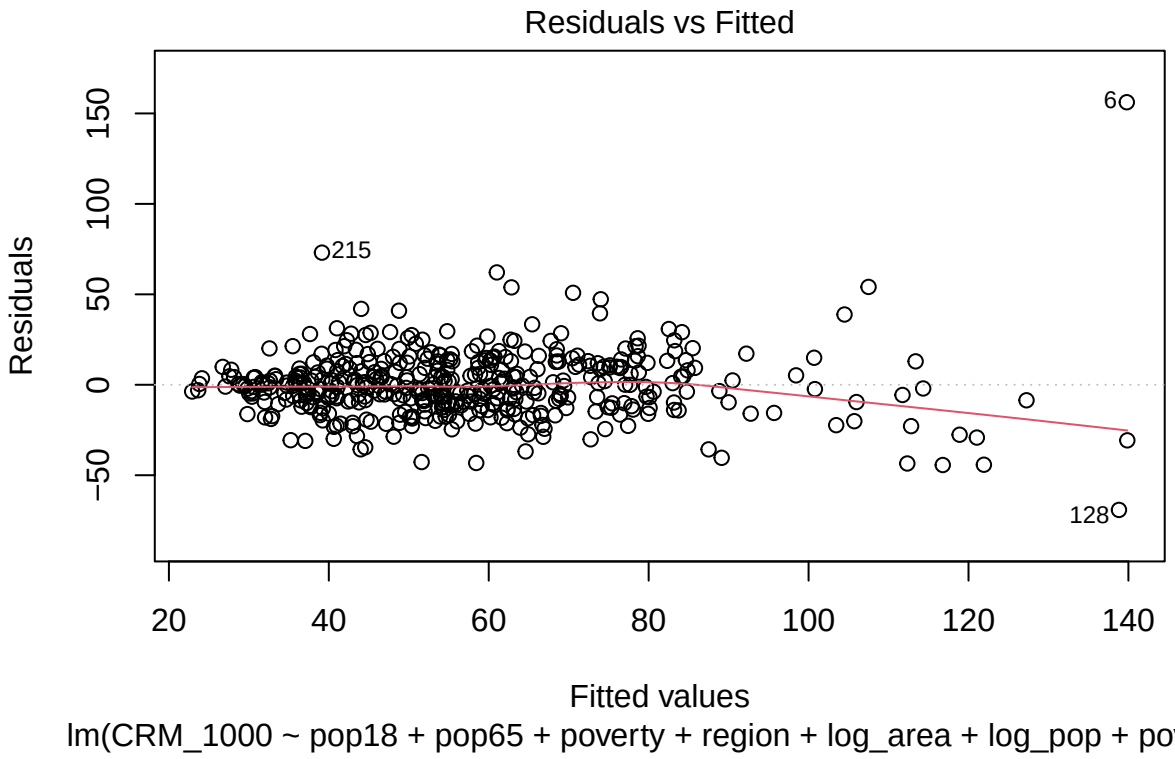
model diagnostic

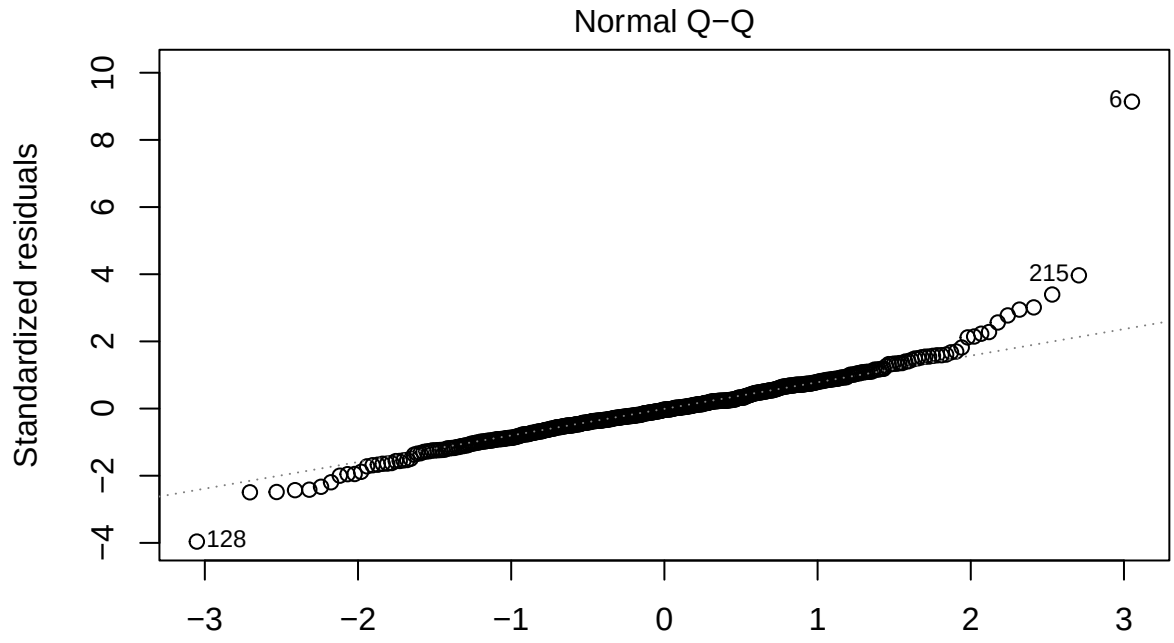
Box-Cox Transformation

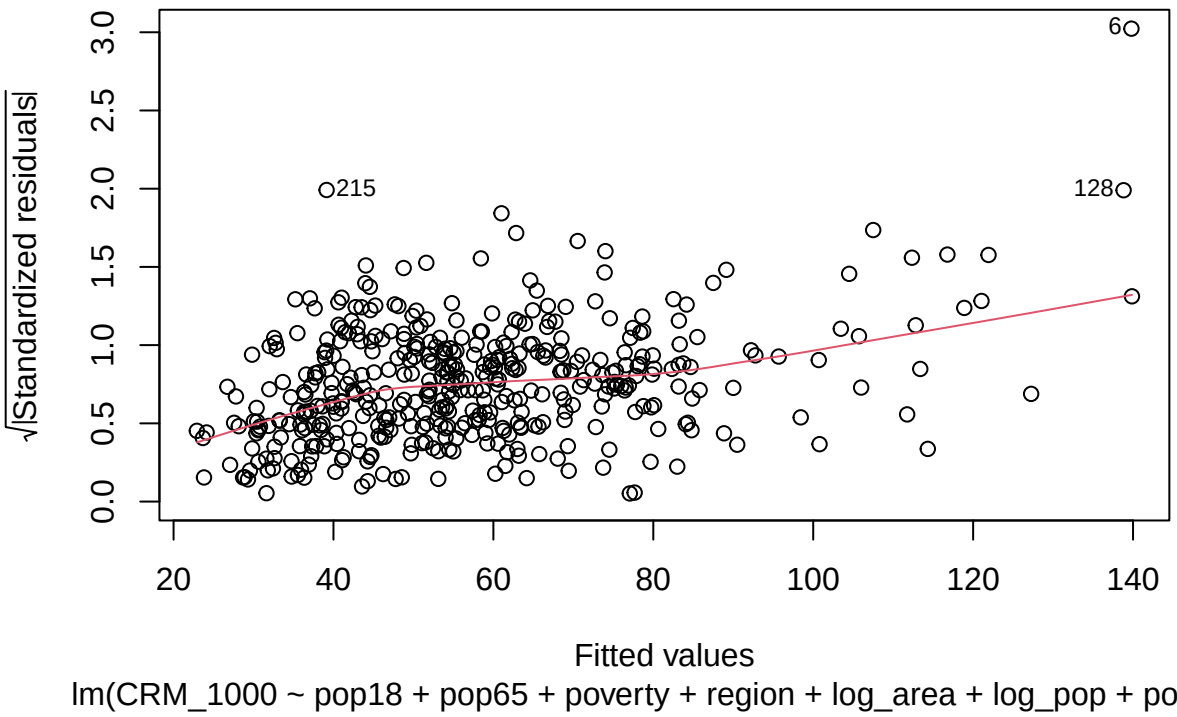
```
# fit model
fit_adj3 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_p

# choose best transformation - choosing from lambda in (-3,3)
boxcox(fit_adj3, lambda = seq(-3, 3, by = 0.25))
```

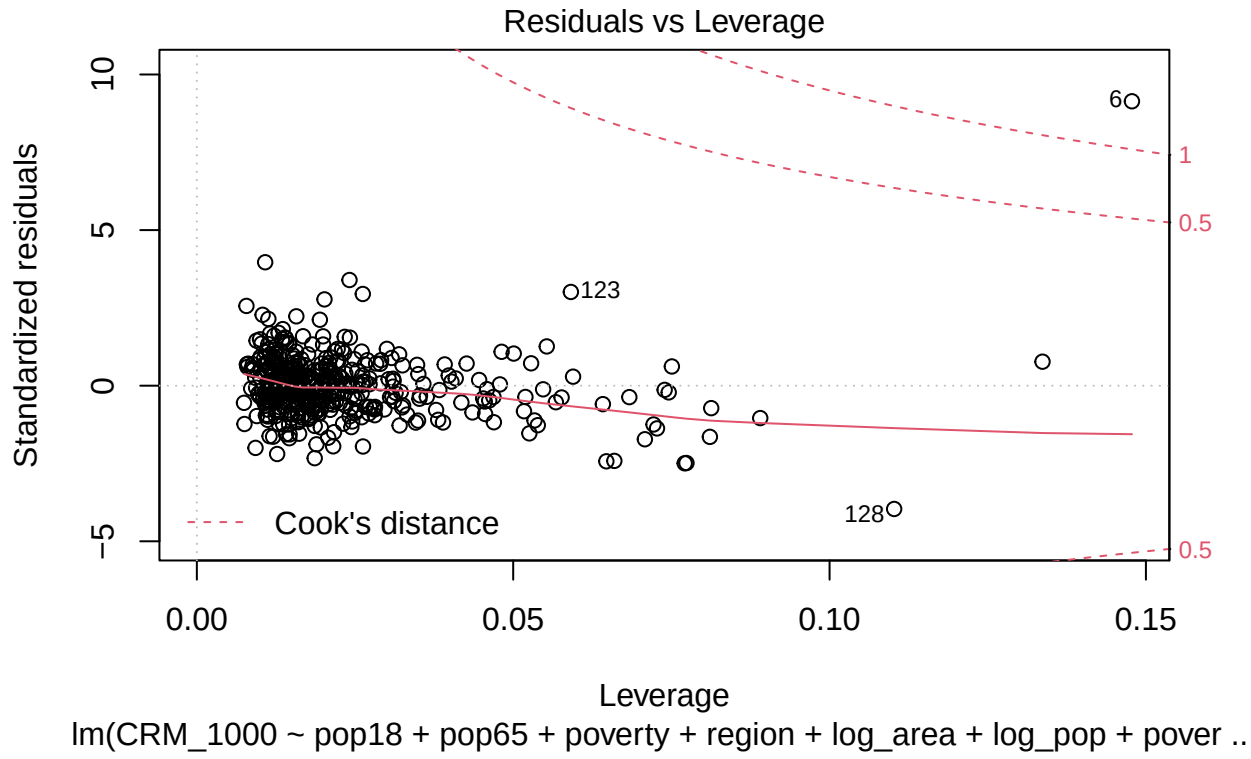



```
# fit multivariate model
mult.fit_adj3 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +poverty*

plot(mult.fit_adj3)
```

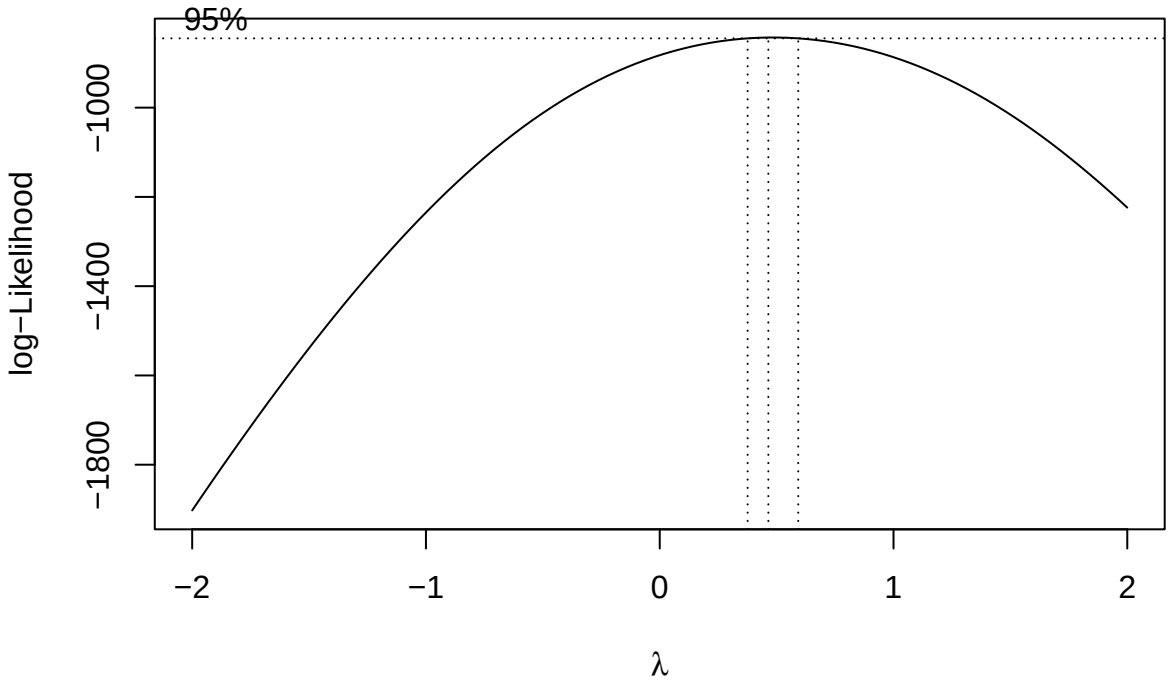



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop + pover ..
6
215
128


Scale−Location
√|Standardized residuals|
Fitted values
lm(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop + pover ..
6
215
128




```
boxcox(mult.fit_adj3)
```



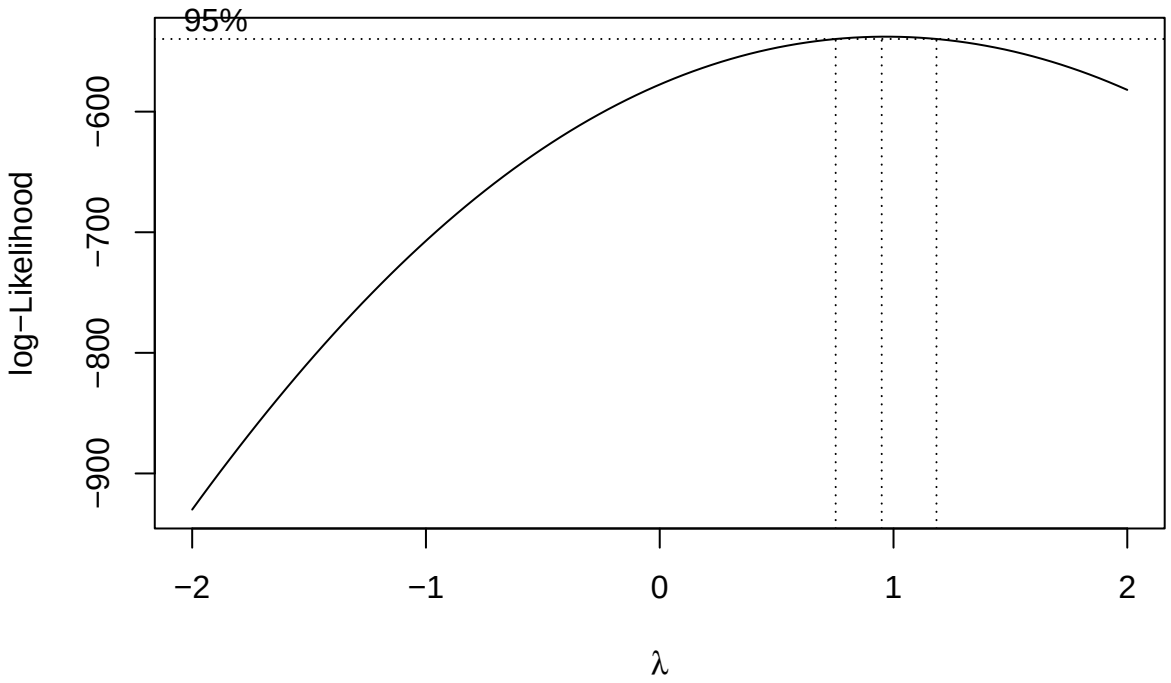
Since the mean is closer to 1/2 in the box-cox plot, we chose square root transformation to create a model with square root values and to compare the model with the original model. In the fit multivariate model, the residual values are normally distributed around 0. In the normal QQ plot, it is linear to a straight line and its residuals are normal.

```
data_sqrt = data_df %>%
  mutate(sqrt_CRM_1000 = sqrt(CRM_1000)) %>%
```

```
  dplyr::select(-CRM_1000)

# fit multivariate model with square root transform ()
sqrt_fit_adj3 = lm(sqrt_CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_p

# check diagnostics
summary(sqrt_fit_adj3)
```

```
##
## Call:
## lm(formula = sqrt_CRM_1000 ~ pop18 + pop65 + poverty + region +
##     log_area + log_pop + poverty * log_pop, data = data_sqrt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0430 -0.6448  0.0096  0.7021  5.7130
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.06449    2.18491  -0.030 0.976467
## pop18            0.06142    0.01744   3.522 0.000473 ***
## pop65            0.03927    0.01840   2.134 0.033371 *
## poverty         -0.50270    0.20529  -2.449 0.014734 *
## region2          0.79874    0.16660   4.794 2.25e-06 ***
## region3          1.76066    0.16105  10.932  < 2e-16 ***
## region4          1.60432    0.20302   7.902 2.32e-14 ***
## log_area        -0.33567    0.07614  -4.409 1.32e-05 ***
## log_pop          0.41409    0.16710   2.478 0.013590 *
## poverty:log_pop  0.05075    0.01633   3.107 0.002013 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.171 on 430 degrees of freedom
## Multiple R-squared:  0.5552, Adjusted R-squared:  0.5459
## F-statistic: 59.63 on 9 and 430 DF,  p-value: < 2.2e-16
```

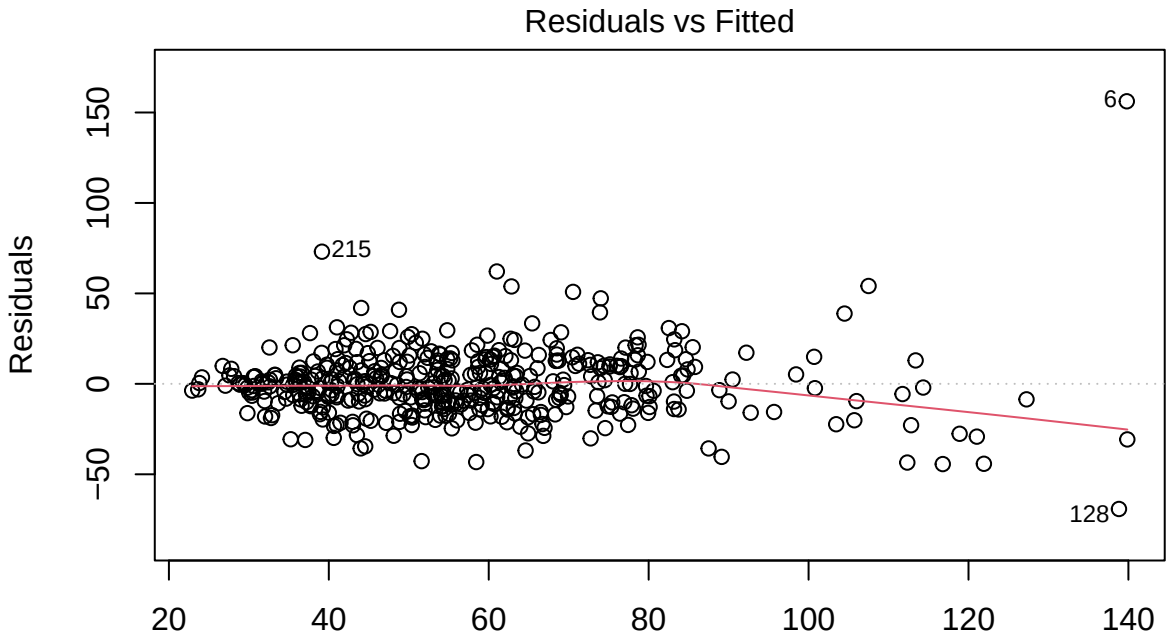
```
boxcox(sqrt_fit_adj3)
```



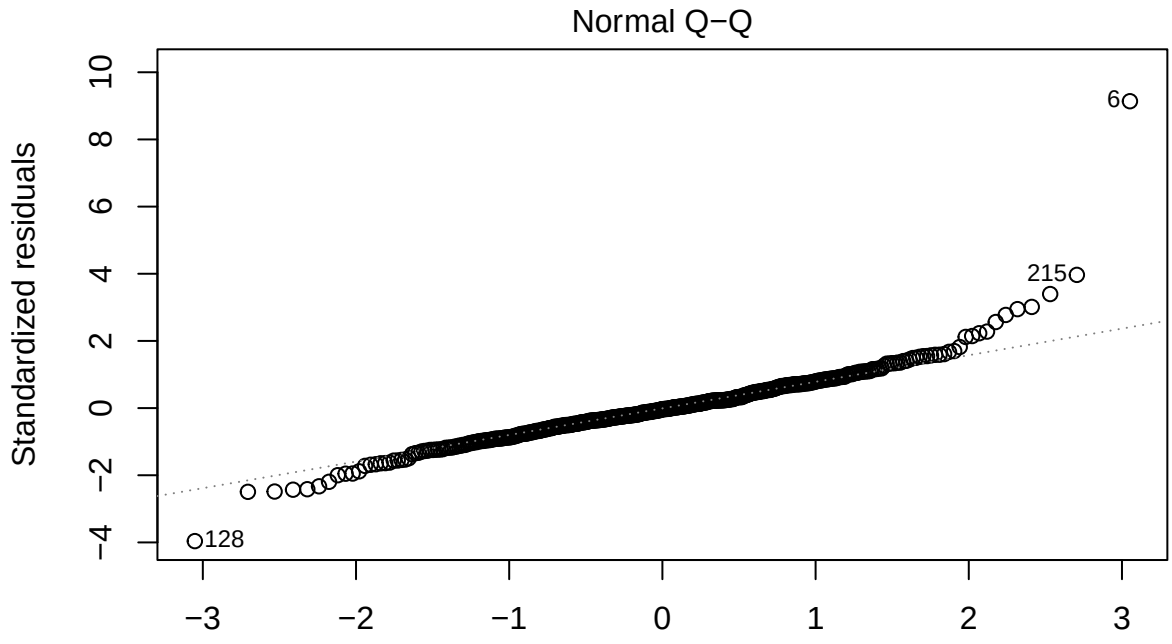
In the box-cox model after square root transformation, the mean is closer to 1.

Residual vs Fitted & QQ Plots

```
# fit model
fit_adj3 = lm(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_pop, data =

# residual vs fitted plot
plot(fit_adj3, which = 1)
```

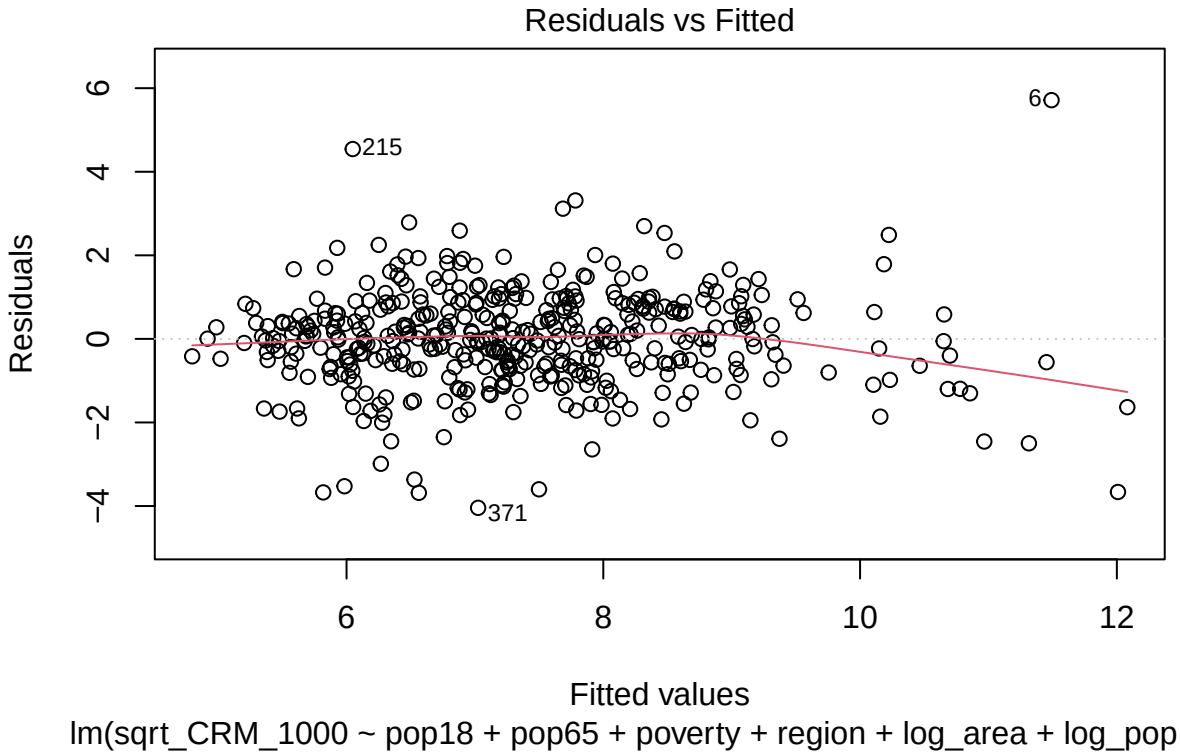



```
# QQ plot
plot(fit_adj3, which = 2)
```

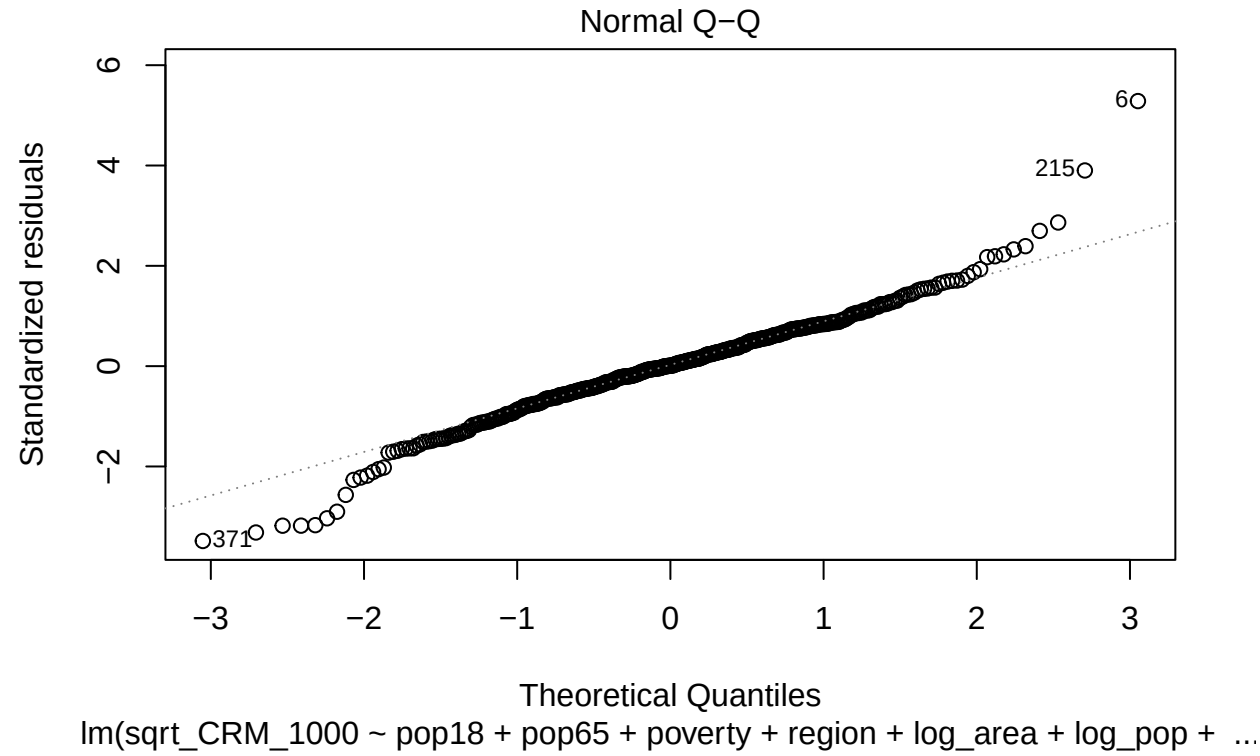



```
# fit model - log transform the outcome
fit3 = lm(sqrt_CRM_1000 ~pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_pop,  data =
```

```r
# residual vs fitted plot
plot(fit3, which = 1)
```




```r
# QQ plot
plot(fit3, which = 2)
```



In this part, we showed four plots: fit model with original data, QQ plot of original model, fit model after square root transforms the outcome, and QQ plot of model with square root values. With comparison of two fit models, fit model with original data is better, since it has random pattern, and residual values are evenly distributed around 0. With comparison of two QQ plots, QQ plot of original model is better, since it is more linear to a straight line and its residuals are normal. The outliers of this fit model are also fewer than the fit model with square root values, which shows the fit model with original value is a better model.

Influential Observations (outliers)

To identify the outliers in this model, and check if they are influential, we first plot the residuals vs leverage plots and calculate the Cook's distance D, which considers the influence of the i^{th} case on all fitted values.

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y_{j(i)}})^2}{pMSE}$$

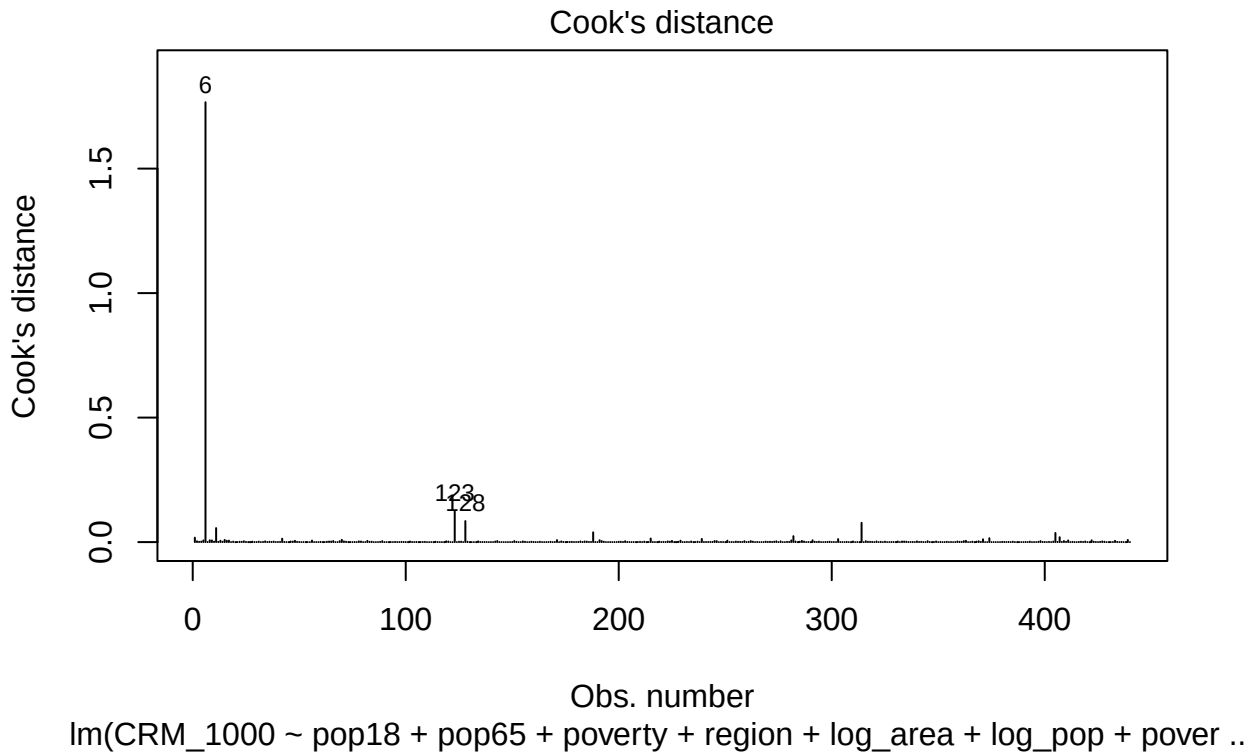
Here we regard $D_i > 4/n$ where $n = 440$ as a threshold of concern.

Model 1

```
# residuals vs leverage plot
model_adj1$call
```

```
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * region, data = data_df)
```

```
fit1 = lm(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop + poverty * region, data = data_df)

plot(fit1, which = 4)
```



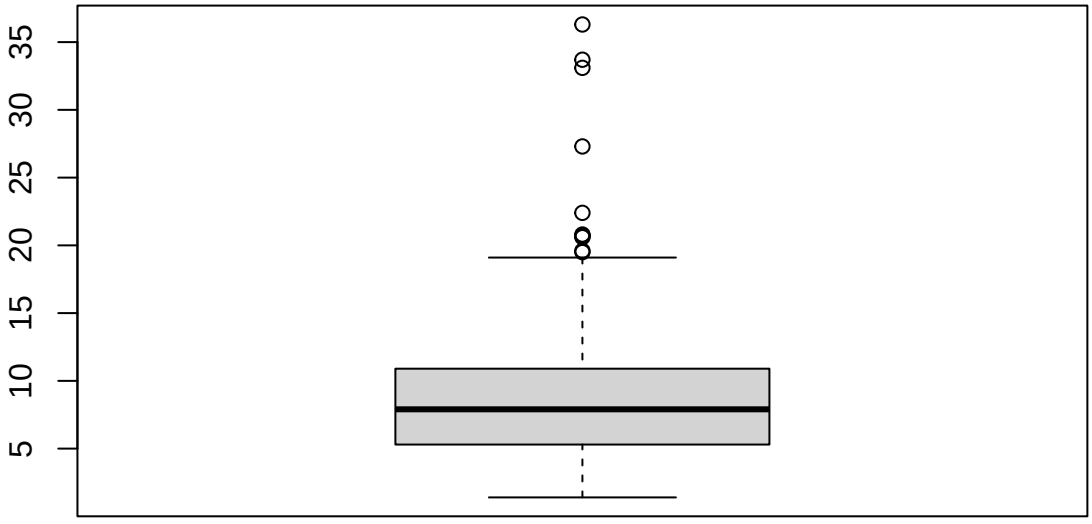
```
data_df[c(6,123,128),]
```

```
##     pop18 pop65 hsgrad poverty unemp pcincome region  CRM_1000 log_area
## 6    28.3  12.4   63.7    19.5   9.5    16803      1 295.98672 4.262680
## 123  28.7  16.6   62.8    20.6   9.0    18113      2 161.59673 4.127134
## 128  26.4  10.1   46.6    36.3  17.6     8899      3  69.64502 7.358194
##      log_pop
## 6   14.64871
## 123 12.89090
## 128 12.85721
```

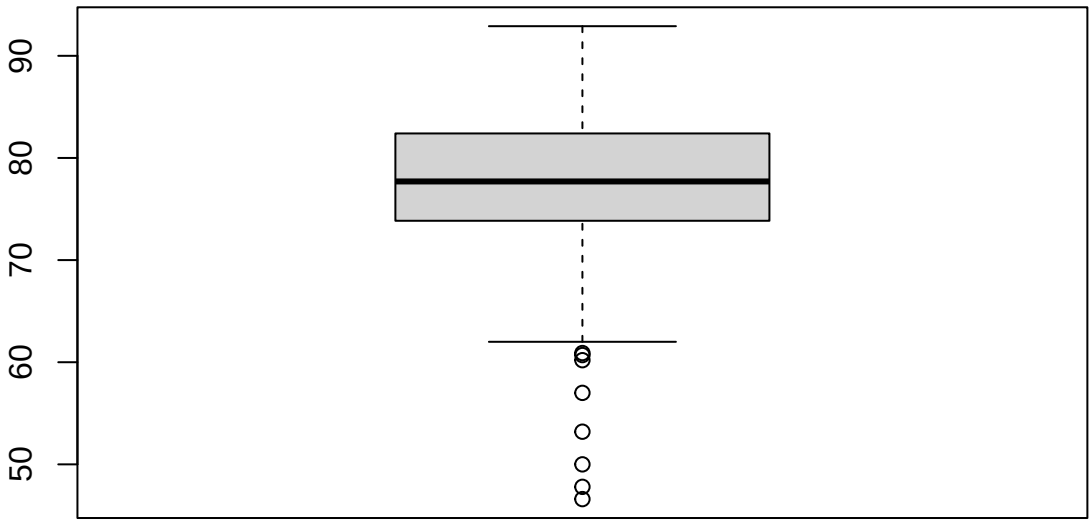
```
summary(data_df)
```

```
##      pop18           pop65           hsgrad         poverty
##  Min.   :16.40   Min.   : 3.000   Min.   :46.60   Min.   : 1.400
##  1st Qu.:26.20   1st Qu.: 9.875   1st Qu.:73.88   1st Qu.: 5.300
##  Median :28.10   Median :11.750   Median :77.70   Median : 7.900
##  Mean   :28.57   Mean   :12.170   Mean   :77.56   Mean   : 8.721
##  3rd Qu.:30.02   3rd Qu.:13.625   3rd Qu.:82.40   3rd Qu.:10.900
##  Max.   :49.70   Max.   :33.800   Max.   :92.90   Max.   :36.300
##      unemp           pcincome     region     CRM_1000          log_area
##  Min.   : 2.200   Min.   : 8899   1:103   Min.   :  4.601   Min.   :2.708
##  1st Qu.: 5.100   1st Qu.:16118   2:108   1st Qu.: 38.102   1st Qu.:6.112
##  Median : 6.200   Median :17759   3:152   Median : 52.429   Median :6.487
##  Mean   : 6.597   Mean   :18561   4: 77   Mean   : 57.286   Mean   :6.517
##  3rd Qu.: 7.500   3rd Qu.:20270           3rd Qu.: 72.597   3rd Qu.:6.853
##  Max.   :21.300   Max.   :37541           Max.   :295.987   Max.   :9.907
##     log_pop
##  Min.   :11.51
##  1st Qu.:11.84
##  Median :12.29
##  Mean   :12.48
##  3rd Qu.:12.99
##  Max.   :16.00
```

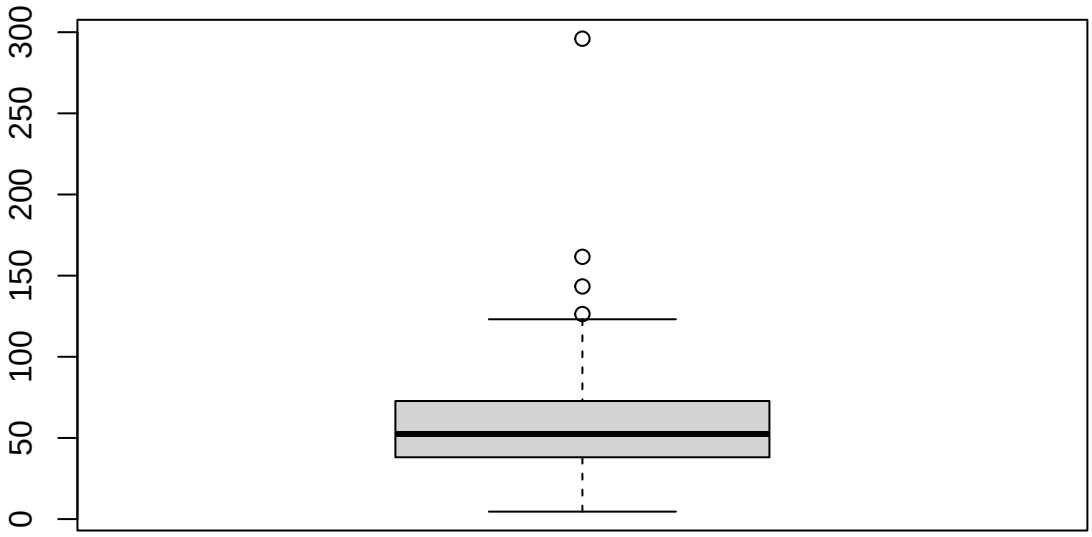
```
# case 128 has extremely high poverty and high hsgrad (Percent high school graduates)
boxplot(data_df$poverty)
```



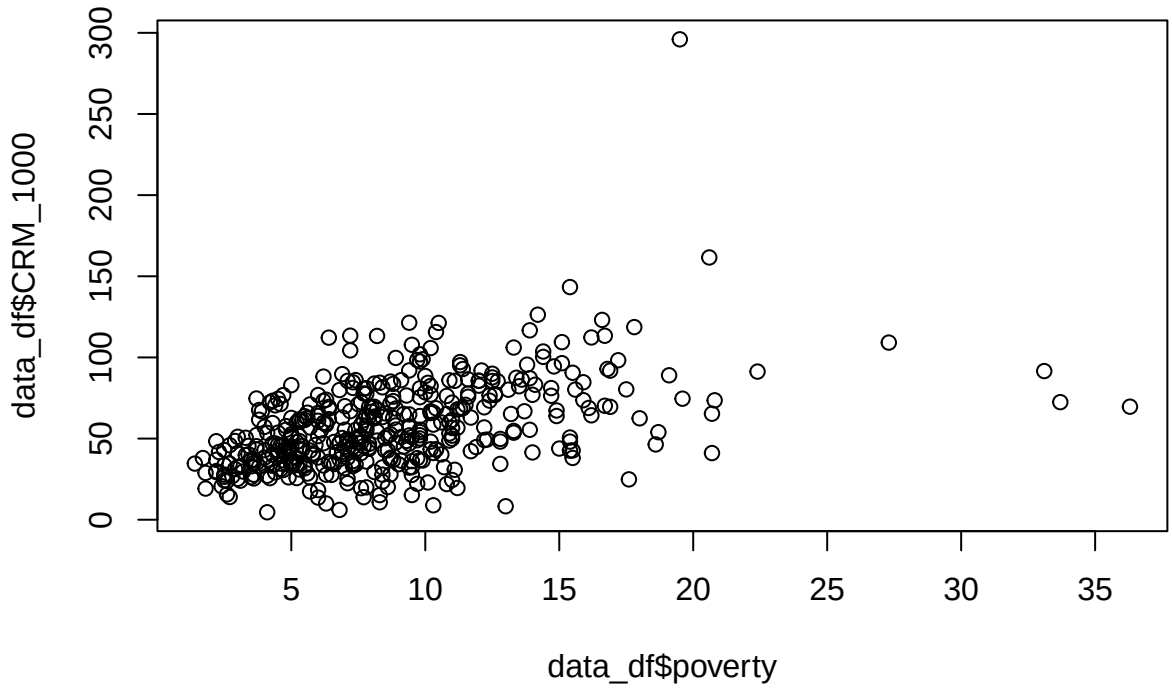
```
boxplot(data_df$hsgrad)
```



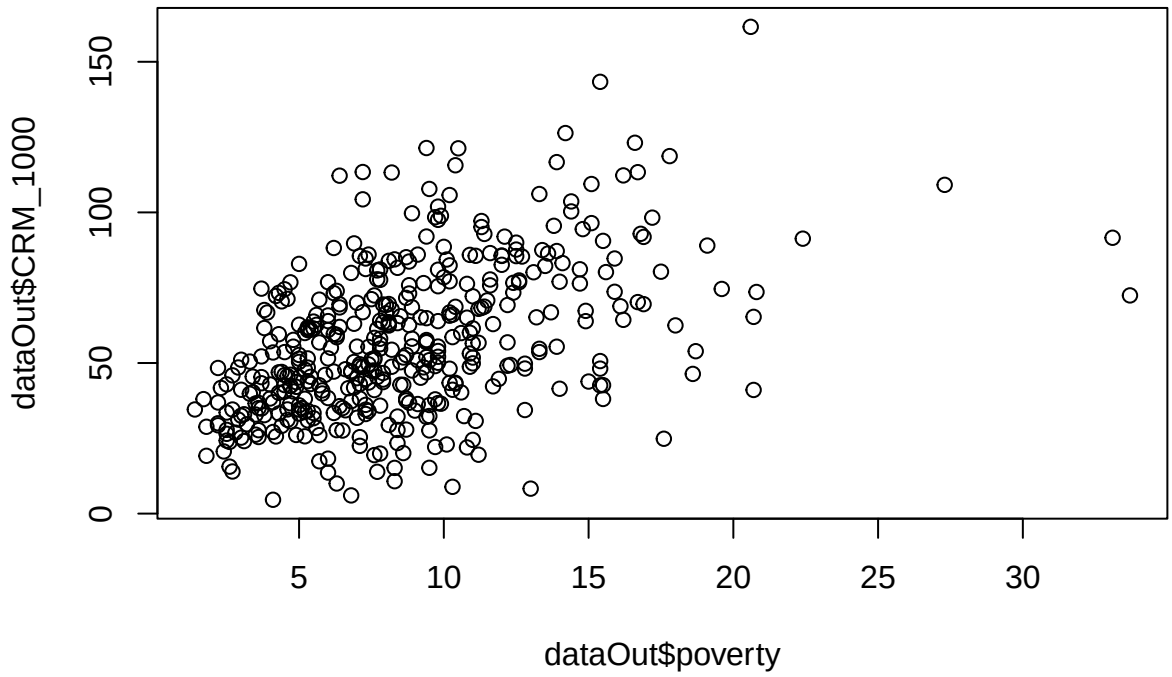

```
# case 6 has extremely high CRM rate
boxplot(data_df$CRM_1000)
```




```
# remove influential points
dataOut = data_df[-c(6,128),]

# plot with and without influential points
plot(data_df$poverty, data_df$CRM_1000)
```




```
plot(dataOut$poverty, dataOut$CRM_1000)
```



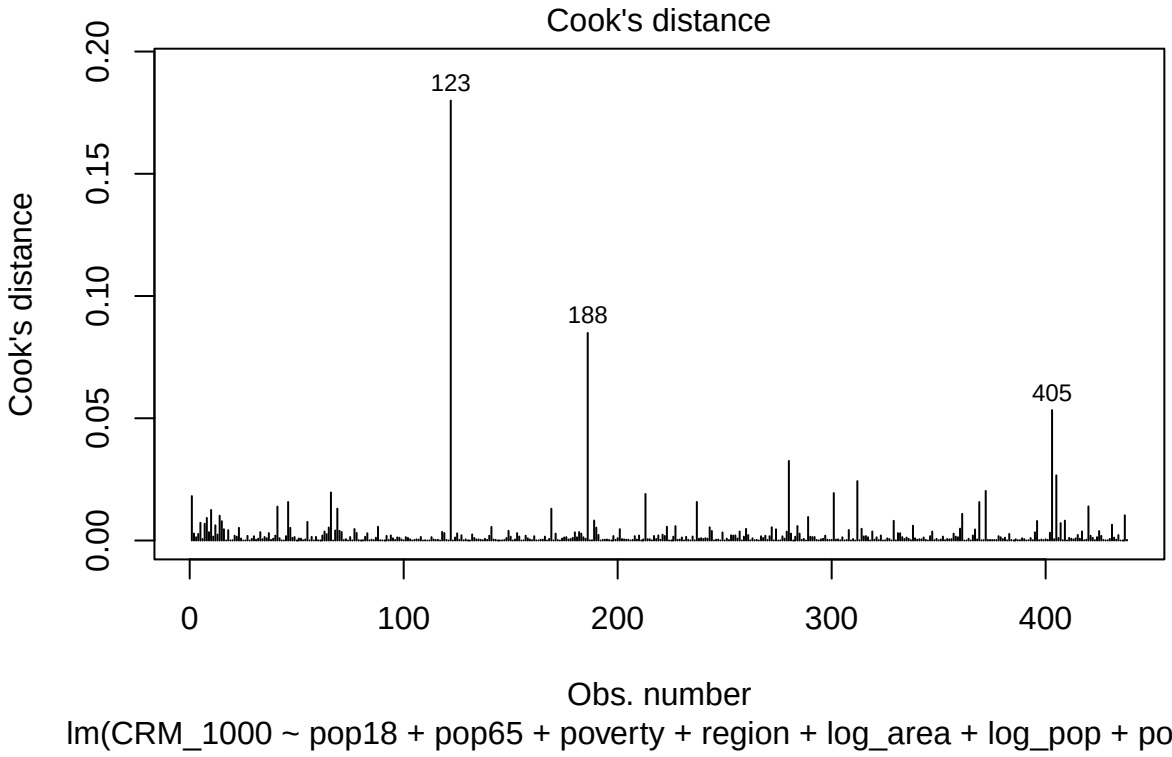

```
# fit model with and without influential points
with1 = lm(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop + poverty * region, data = data_df)

without1 = lm(CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop + poverty * region, data = dataOut)

summary(with1); summary(without1)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * region, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.220 -10.559  -0.369  10.453 163.086
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -143.6838    19.8186  -7.250 1.96e-12 ***
## pop18              0.7427     0.2811   2.642 0.008544 **
## pop65              0.2533     0.3044   0.832 0.405659
## poverty            3.9266     0.5586   7.030 8.22e-12 ***
## region2           14.1447     6.2675   2.257 0.024522 *
## region3           40.1729     5.2324   7.678 1.11e-13 ***
## region4           28.8289     6.7414   4.276 2.34e-05 ***
## log_area          -6.0784     1.2457  -4.880 1.50e-06 ***
## log_pop           13.7737     1.1698  11.774  < 2e-16 ***
## poverty:region2   -0.7942     0.7762  -1.023 0.306794
## poverty:region3   -2.2579     0.6272  -3.600 0.000355 ***
## poverty:region4   -1.5029     0.7887  -1.906 0.057374 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.78 on 428 degrees of freedom
## Multiple R-squared:  0.5395, Adjusted R-squared:  0.5277
## F-statistic: 45.59 on 11 and 428 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * region, data = dataOut)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.583 -10.003   0.137  10.100  75.186
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -130.2557    17.4808  -7.451 5.19e-13 ***
## pop18              0.8959     0.2482   3.609 0.000344 ***
## pop65              0.4674     0.2687   1.740 0.082668 .
## poverty            1.6883     0.5325   3.170 0.001632 **
## region2            1.3202     5.6376   0.234 0.814953
## region3           25.1116     4.8255   5.204 3.04e-07 ***
## region4           17.2484     6.0233   2.864 0.004395 **
## log_area          -4.5771     1.1041  -4.146 4.09e-05 ***
## log_pop           12.3910     1.0387  11.929  < 2e-16 ***
## poverty:region2    1.4634     0.7135   2.051 0.040866 *
## poverty:region3    0.2149     0.5952   0.361 0.718238
## poverty:region4    0.4951     0.7176   0.690 0.490622
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.52 on 426 degrees of freedom
## Multiple R-squared:  0.5702, Adjusted R-squared:  0.5591
## F-statistic: 51.38 on 11 and 426 DF,  p-value: < 2.2e-16
```

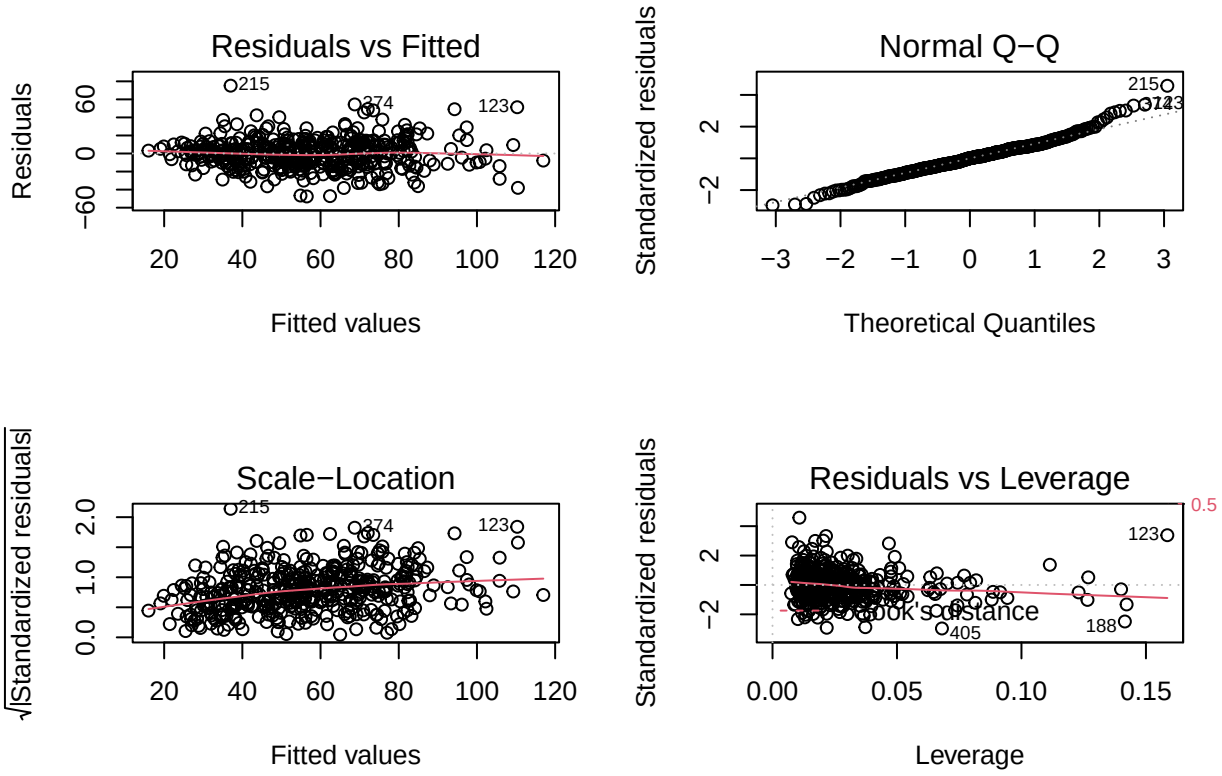
```
### The model without outliers have higher significancy and R-squared value

# check without diagnostics
# par(mfrow=c(2,2))
plot(without1,which=4)
```




```
## further elimination
dataOut[c(123,188,405),]
```

```
##     pop18 pop65 hsgrad poverty unemp pcincome region CRM_1000 log_area  log_pop
## 124  28.8  10.7   81.9     3.1   5.6    23008      1 32.99196 6.690842 12.88681
## 190  31.2  11.9   80.9     4.7   6.8    20259      1 30.62085 6.501290 12.44885
## 407  23.4  14.9   71.8    13.0   6.7    12597      2  8.29362 6.278521 11.59244
```

```
# seems normal in all dimension, therefore kept all cases.

par(mfrow=c(2,2))
plot(without1)
```

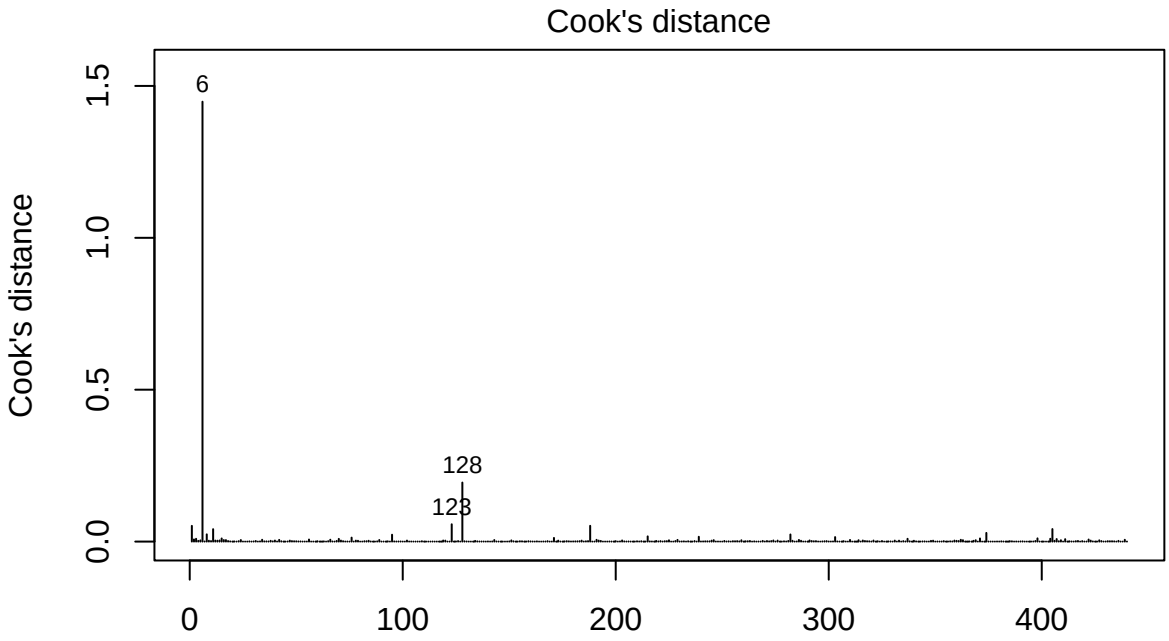


Model 3

```
# residuals vs leverage plot
model_adj3$call
```

```
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * log_pop, data = data_df)
```

```
fit2 = lm(CRM_1000 ~pop18 + pop65 + poverty + region + log_area + log_pop +poverty*log_pop, data = data_

plot(fit2, which = 4)
```




```
data_df[c(6,123,128),]
```

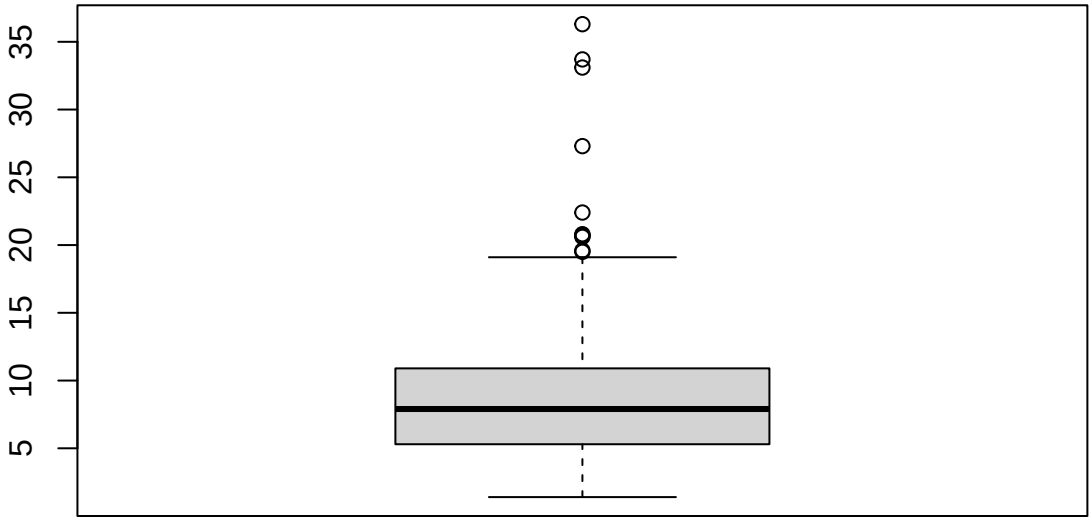
```
##     pop18 pop65 hsgrad poverty unemp pcincome region  CRM_1000 log_area
## 6    28.3  12.4   63.7    19.5   9.5    16803      1 295.98672 4.262680
## 123  28.7  16.6   62.8    20.6   9.0    18113      2 161.59673 4.127134
## 128  26.4  10.1   46.6    36.3  17.6     8899      3  69.64502 7.358194
##      log_pop
## 6   14.64871
## 123 12.89090
## 128 12.85721
```

```
summary(data_df)
```

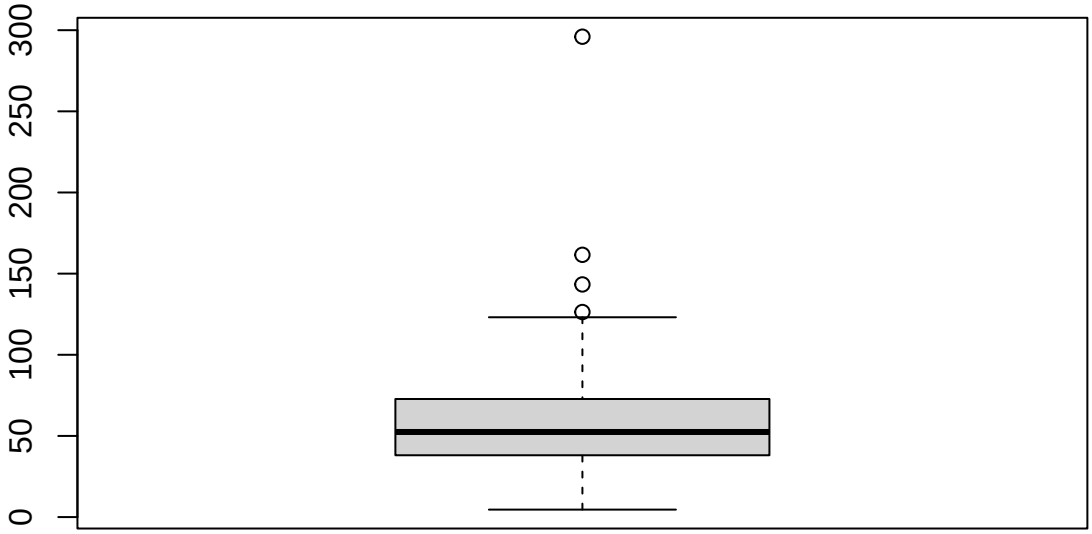
```
##      pop18           pop65           hsgrad         poverty
##  Min.   :16.40   Min.   : 3.000   Min.   :46.60   Min.   : 1.400
##  1st Qu.:26.20   1st Qu.: 9.875   1st Qu.:73.88   1st Qu.: 5.300
##  Median :28.10   Median :11.750   Median :77.70   Median : 7.900
##  Mean   :28.57   Mean   :12.170   Mean   :77.56   Mean   : 8.721
##  3rd Qu.:30.02   3rd Qu.:13.625   3rd Qu.:82.40   3rd Qu.:10.900
##  Max.   :49.70   Max.   :33.800   Max.   :92.90   Max.   :36.300
##      unemp           pcincome     region     CRM_1000          log_area
##  Min.   : 2.200   Min.   : 8899   1:103   Min.   :  4.601   Min.   :2.708
##  1st Qu.: 5.100   1st Qu.:16118   2:108   1st Qu.: 38.102   1st Qu.:6.112
##  Median : 6.200   Median :17759   3:152   Median : 52.429   Median :6.487
##  Mean   : 6.597   Mean   :18561   4: 77   Mean   : 57.286   Mean   :6.517
##  3rd Qu.: 7.500   3rd Qu.:20270           3rd Qu.: 72.597   3rd Qu.:6.853
##  Max.   :21.300   Max.   :37541           Max.   :295.987   Max.   :9.907
##     log_pop
##  Min.   :11.51
##  1st Qu.:11.84
##  Median :12.29
```

```
##  Mean   :12.48
##  3rd Qu.:12.99
##  Max.   :16.00
```

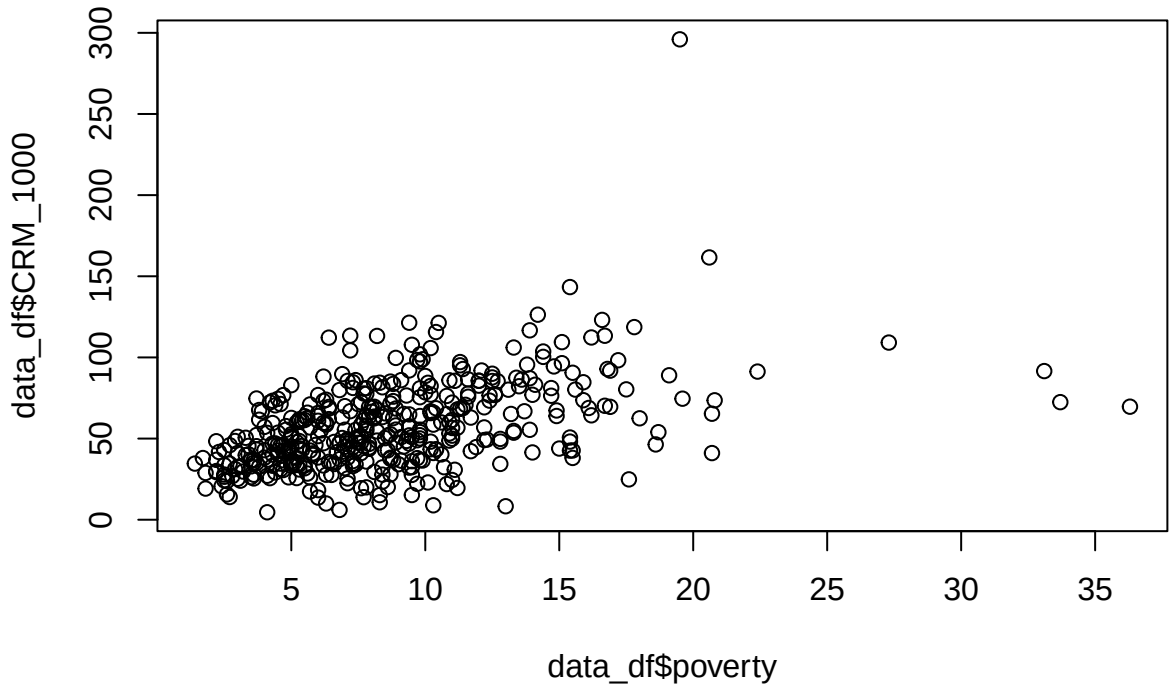
```
# case 128 has extremely high poverty
boxplot(data_df$poverty)
```



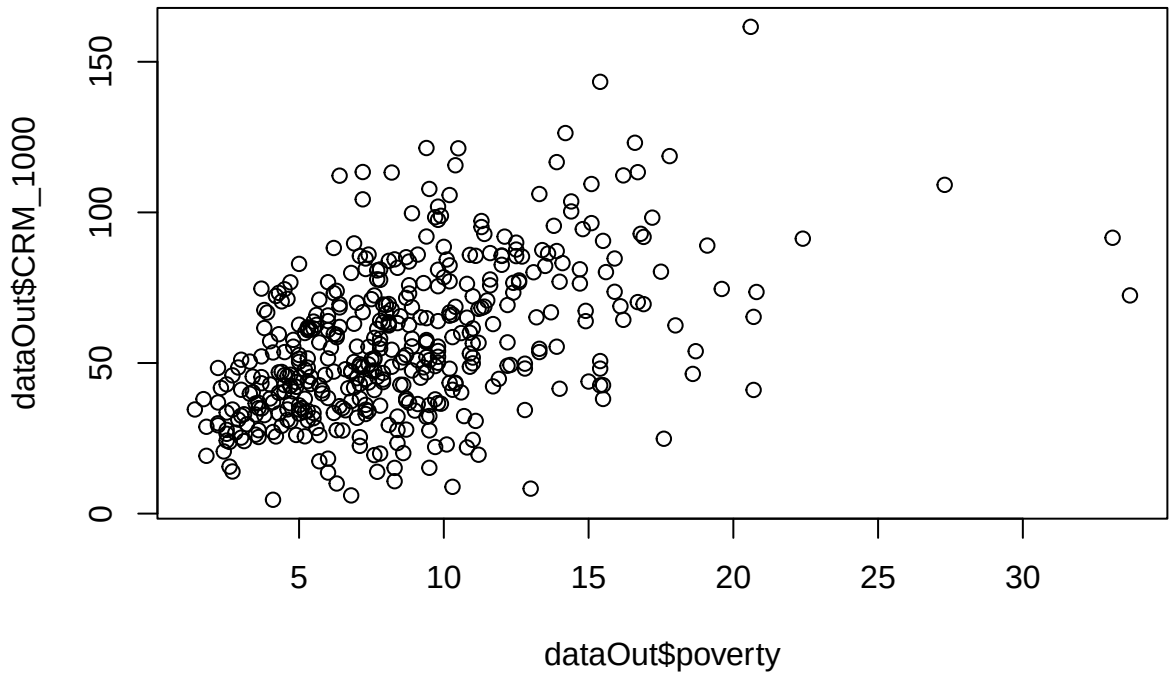

```
# case 6 has extremely high CRM rate
boxplot(data_df$CRM_1000)
```




```
# remove influential points
dataOut = data_df[-c(6,128),]

# plot with and without influential points
plot(data_df$poverty, data_df$CRM_1000)
```




```
plot(dataOut$poverty, dataOut$CRM_1000)
```



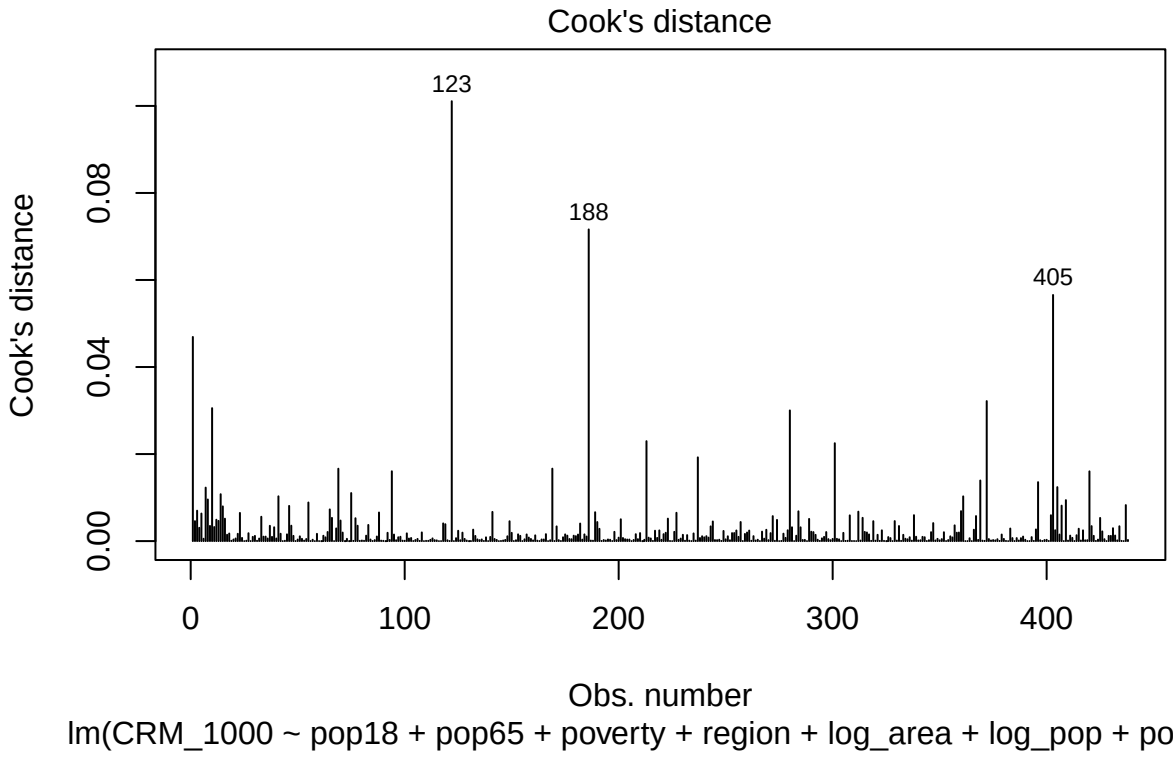

```
# fit model with and without influential points
with3 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop + poverty*log_pop, data = data_df)

without3 = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop + poverty*log_pop, data = dataOut)

summary(with3); summary(without3)
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * log_pop, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.175  -9.881  -0.644   9.635 156.188
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.8989    34.5445   0.258 0.796835
## pop18             0.8701     0.2757   3.156 0.001711 **
## pop65             0.5142     0.2909   1.768 0.077792 .
## poverty         -14.2931     3.2457  -4.404 1.34e-05 ***
## region2           9.7723     2.6340   3.710 0.000234 ***
## region3          23.5341     2.5463   9.243  < 2e-16 ***
## region4          20.7983     3.2099   6.479 2.53e-10 ***
## log_area         -5.7992     1.2038  -4.817 2.02e-06 ***
## log_pop           1.7393     2.6419   0.658 0.510674
## poverty:log_pop   1.3228     0.2582   5.122 4.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.51 on 430 degrees of freedom
## Multiple R-squared:  0.5504, Adjusted R-squared:  0.541
## F-statistic: 58.49 on 9 and 430 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
##     log_pop + poverty * log_pop, data = dataOut)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.727  -9.856  -0.538  10.128  74.624
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -54.5825    31.8111  -1.716 0.086917 .
## pop18             0.9126     0.2455   3.718 0.000228 ***
## pop65             0.5408     0.2591   2.087 0.037463 *
## poverty          -7.0544     3.0298  -2.328 0.020358 *
## region2          11.7562     2.3525   4.997 8.49e-07 ***
## region3          25.6478     2.2754  11.272  < 2e-16 ***
## region4          20.9343     2.8567   7.328 1.17e-12 ***
## log_area         -4.4431     1.0749  -4.133 4.30e-05 ***
## log_pop           5.9334     2.4166   2.455 0.014476 *
## poverty:log_pop   0.7356     0.2425   3.034 0.002564 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.42 on 428 degrees of freedom
## Multiple R-squared:  0.5735, Adjusted R-squared:  0.5645
```

```
## F-statistic: 63.95 on 9 and 428 DF,  p-value: < 2.2e-16
```

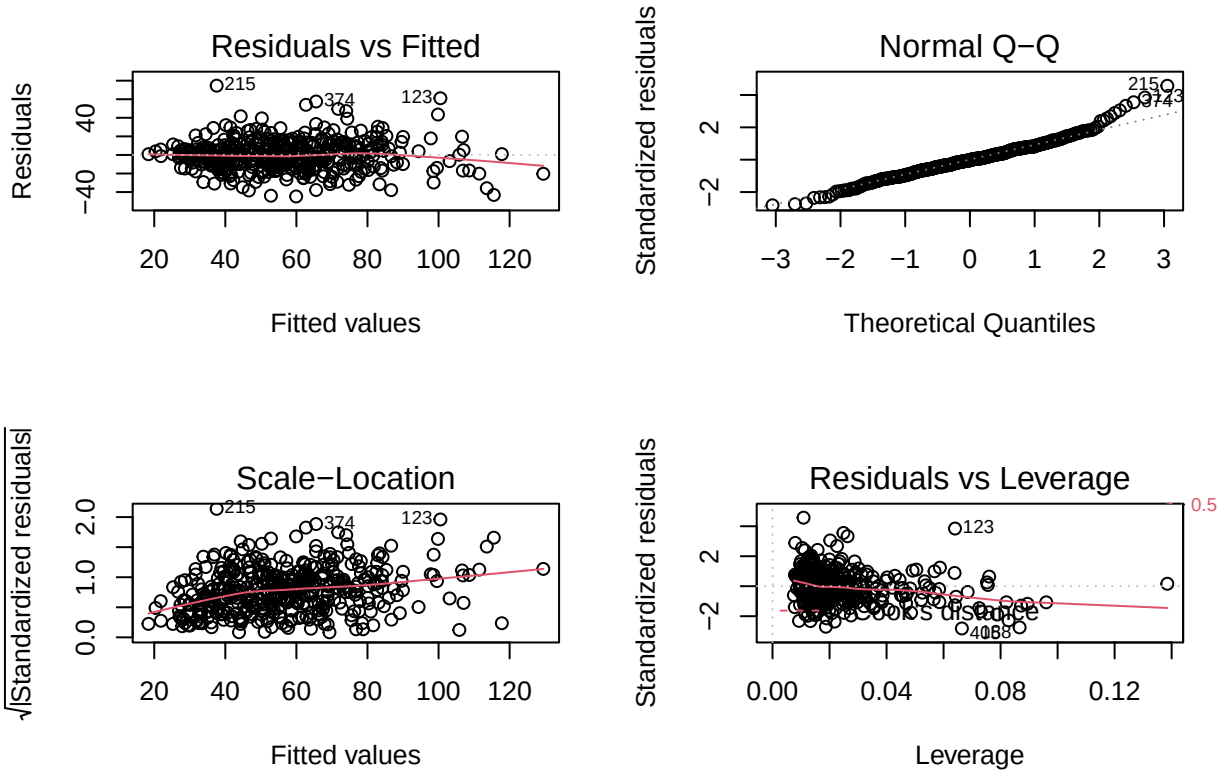
```
### The model without outliers have higher significancy and R-squared value

# check without diagnostics
plot(without3,which=4)
```




```
## further elimination
dataOut[c(123,188,405),]
```

```
##     pop18 pop65 hsgrad poverty unemp pcincome region CRM_1000 log_area  log_pop
## 124  28.8  10.7   81.9     3.1   5.6    23008      1 32.99196 6.690842 12.88681
## 190  31.2  11.9   80.9     4.7   6.8    20259      1 30.62085 6.501290 12.44885
## 407  23.4  14.9   71.8    13.0   6.7    12597      2  8.29362 6.278521 11.59244
```

```
# seems normal in all dimension, therefore kept all cases.

par(mfrow=c(2,2))
plot(without3)
```



model_adj1 and model_adj3 have same outlier situation.

Multicolliearity check

```
model_noint = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop, data = dataOut)
check_collinearity(model_noint)
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##      Term  VIF Increased SE Tolerance
##     pop18 1.72         1.31      0.58
##     pop65 1.74         1.32      0.57
##   poverty 1.18         1.09      0.85
##    region 1.67         1.29      0.60
##  log_area 1.39         1.18      0.72
##   log_pop 1.04         1.02      0.96
```

Final model

```
final = lm(formula = CRM_1000 ~ pop18 + pop65 + poverty + region + log_area +
    log_pop + +poverty*log_pop, data = dataOut)

terms = c("Residual","Percent of population - aged 18-34","Percent of population - aged 65+", "Percent

broom::tidy(final) %>%
```

```
mutate(term = terms) %>%
arrange(p.value) %>%
knitr::kable(caption = "Final Model Estimates",digits = c(2,2,2,2,4))
```

Table 4: Final Model Estimates

term	estimate	std.error	statistic	p.value
Geographic region - South	25.65	2.28	11.27	0.0000
Geographic region - West	20.93	2.86	7.33	0.0000
Geographic region - North Central	11.76	2.35	5.00	0.0000
Log(Land area)	-4.44	1.07	-4.13	0.0000
Percent of population - aged 18-34	0.91	0.25	3.72	0.0002
Interaction between poverty and total population	0.74	0.24	3.03	0.0026
Log(Total population)	5.93	2.42	2.46	0.0145
Percent below poverty level	-7.05	3.03	-2.33	0.0204
Percent of population - aged 65+	0.54	0.26	2.09	0.0375
Residual	-54.58	31.81	-1.72	0.0869