

BM1 Final Project

Yijia Jiang, Yifei Xu, Xinyi Zhou, Hengxuan Ma, Chao Gao

11/16/2021

Purpose

We will be analyzing data from the “County Demographic Information” (CDI) data set, which contains characteristics of 440 counties in the United States collected from 1990-1992. The primary objective of this investigation is to develop insight relevant to predicting the crime rate in counties, namely to summarize as the crime rate per 1,000 population (CRM_1000).

Import the package we need

Data preprocessing

Transfer population variables to per capita variables

```
rm(list = ls())
cdi = read.csv("./data/cdi.csv") %>%
  mutate(crime_rate = crimes/pop,
         pcarea = area/pop,
         pcdocs = docs/pop,
         pcbeds = beds/pop,
         region = relevel(factor(region),ref = 3))
cdi_pc = cdi %>%
  dplyr::select(crime_rate, everything(), -id, -cty, -state, -area, -docs, -beds, -crimes, -pop, -total)
summary(cdi_pc)
```

##	crime_rate	pop18	pop65	hsgrad	
##	Min. :0.004601	Min. :16.40	Min. : 3.000	Min. :46.60	
##	1st Qu.:0.038102	1st Qu.:26.20	1st Qu.: 9.875	1st Qu.:73.88	
##	Median :0.052429	Median :28.10	Median :11.750	Median :77.70	
##	Mean :0.057286	Mean :28.57	Mean :12.170	Mean :77.56	
##	3rd Qu.:0.072597	3rd Qu.:30.02	3rd Qu.:13.625	3rd Qu.:82.40	
##	Max. :0.295987	Max. :49.70	Max. :33.800	Max. :92.90	
##	bagrad	poverty	unemp	pcincome	region
##	Min. : 8.10	Min. : 1.400	Min. : 2.200	Min. : 8899	3:152
##	1st Qu.:15.28	1st Qu.: 5.300	1st Qu.: 5.100	1st Qu.:16118	1:103
##	Median :19.70	Median : 7.900	Median : 6.200	Median :17759	2:108
##	Mean :21.08	Mean : 8.721	Mean : 6.597	Mean :18561	4: 77
##	3rd Qu.:25.32	3rd Qu.:10.900	3rd Qu.: 7.500	3rd Qu.:20270	

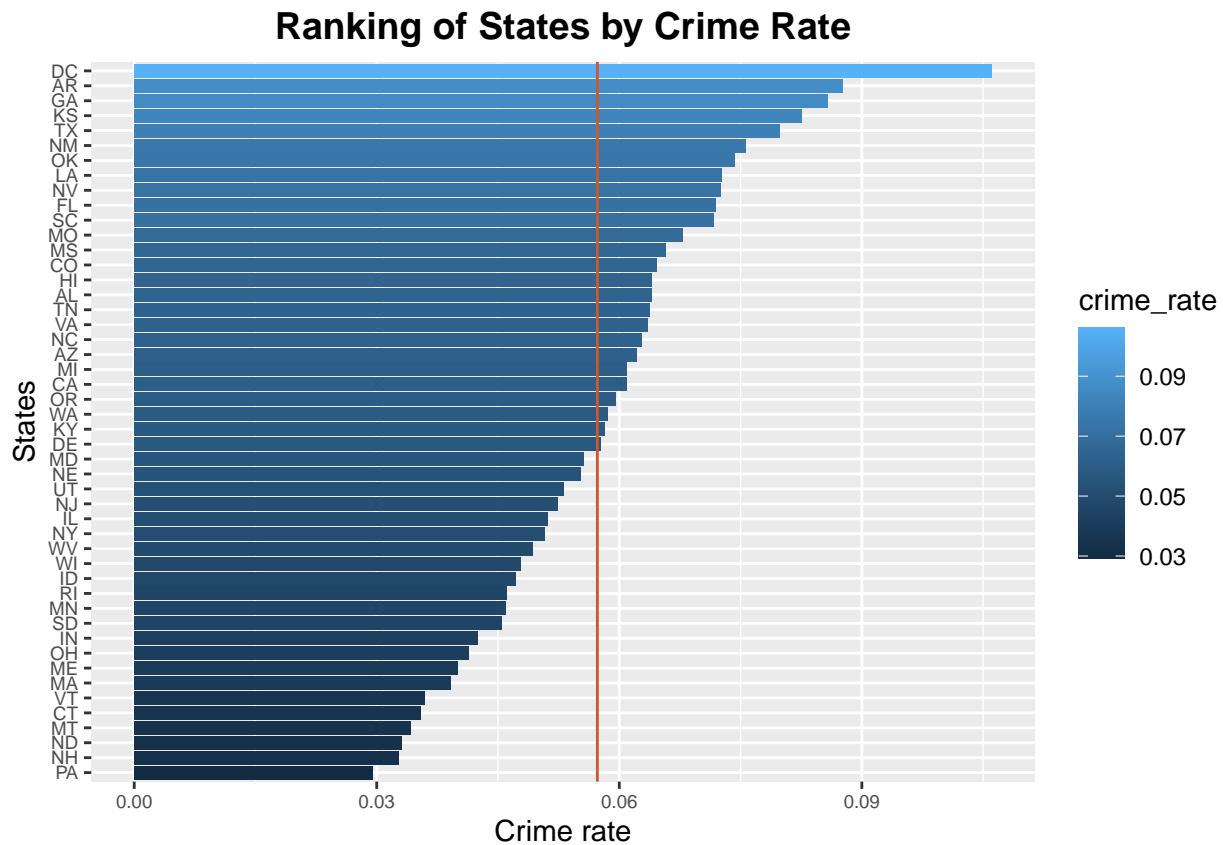
```
## Max. :52.30 Max. :36.300 Max. :21.300 Max. :37541
##      pcarea      pcdocs      pcbeds
## Min. :3.086e-05 Min. :0.0003559 Min. :0.0001649
## 1st Qu.:1.323e-03 1st Qu.:0.0012127 1st Qu.:0.0021972
## Median :2.977e-03 Median :0.0017509 Median :0.0033287
## Mean :4.760e-03 Mean :0.0021230 Mean :0.0036493
## 3rd Qu.:5.199e-03 3rd Qu.:0.0024915 3rd Qu.:0.0045649
## Max. :7.542e-02 Max. :0.0170377 Max. :0.0196982
```

Exploratory Data Analysis

Rank of the crime rate by state

```
cdi_state <- cdi %>%
  group_by(state) %>%
  summarize(crime_rate = mean(crime_rate))
#cdi_state_rank <- cdi_state[order(-rank(cdi_state$crime_rate)),]

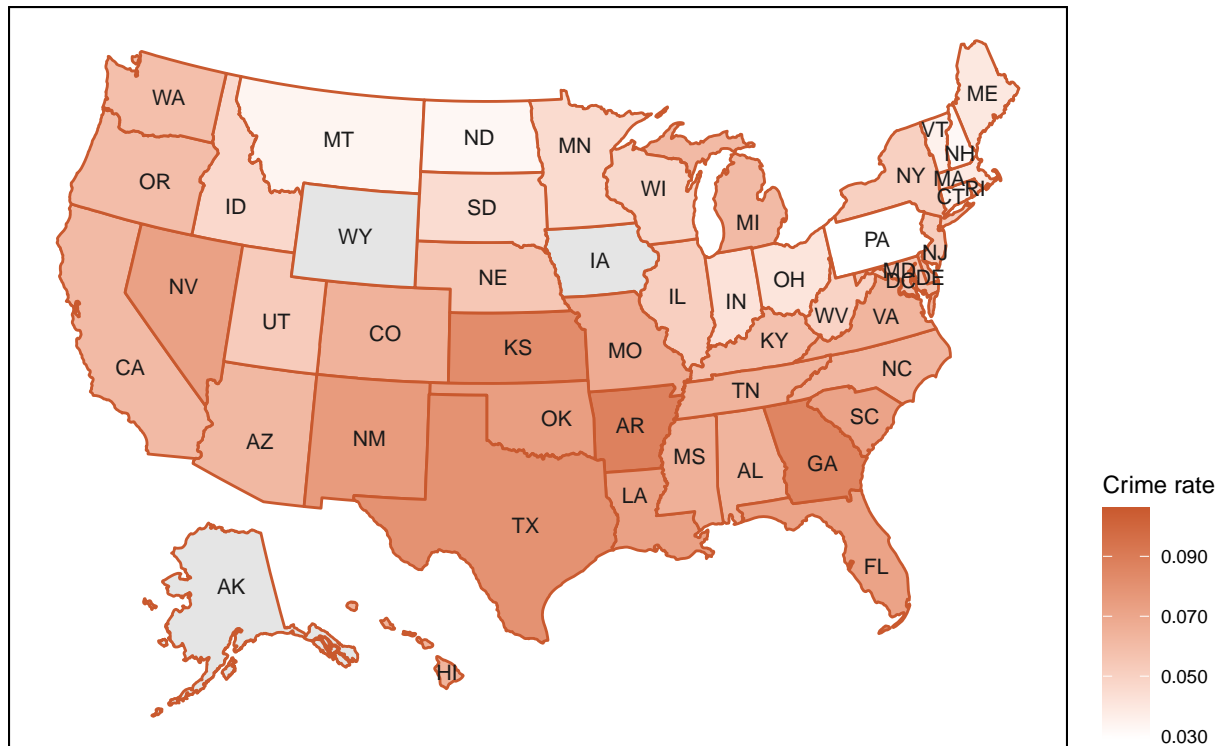
ggplot(cdi_state,aes(x=reorder(state,crime_rate),y=crime_rate,fill=crime_rate)) +
  geom_bar(stat = 'identity')+
  coord_flip() +
  theme_grey() +
  labs(title = 'Ranking of States by Crime Rate',
        y='Crime rate',x='States') +
  geom_hline(yintercept = mean(cdi$crime_rate),color = "#C9592E")+
  theme(plot.title = element_text(hjust = 0.5,size = 14, face = "bold"),
        axis.text=element_text(size=6.5))
```



US crime rate map by state

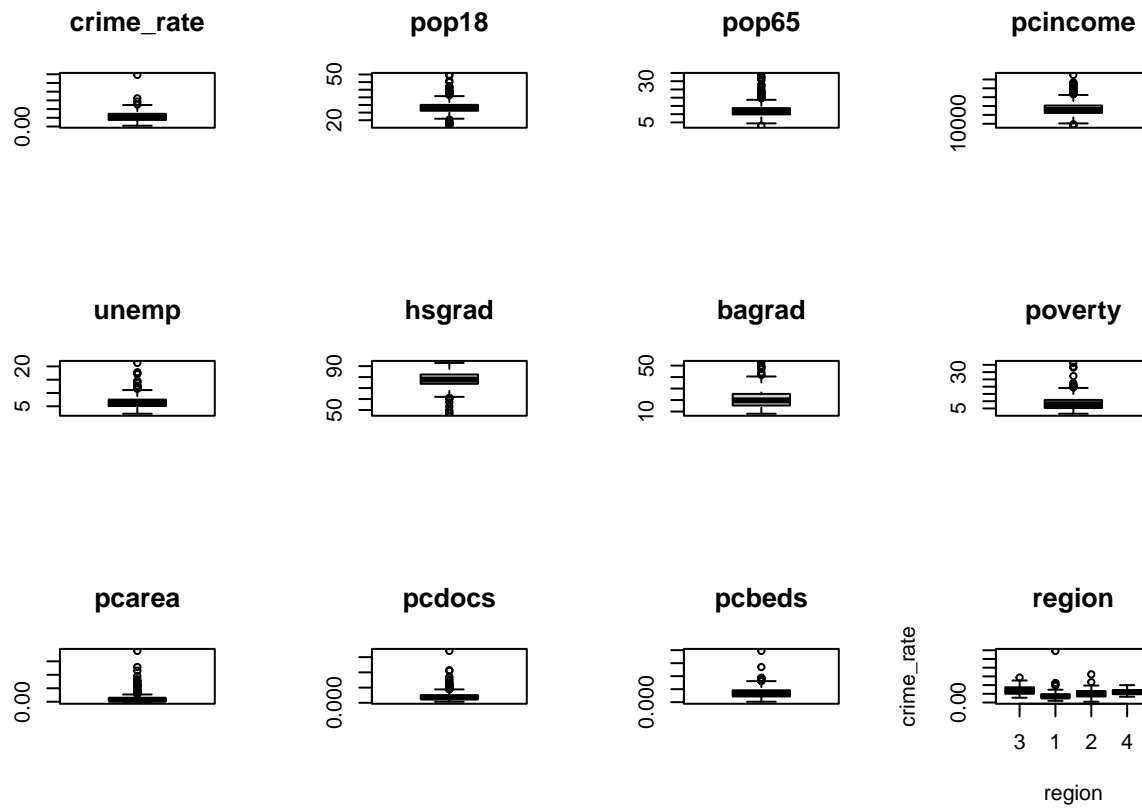
```
p<-plot_usmap(data = cdi_state, values = "crime_rate", color = "#C9592E", size = 0.5,
  labels = TRUE, label_color = "grey10") +
  scale_fill_continuous(low = "white", high = "#C9592E", na.value = "grey90",
    name = "Crime rate", label = scales::comma) +
  theme(legend.position = "right") +
  theme(panel.background = element_rect(colour = "black")) +
  labs(title = "US Crime Rate Map") +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
p$layers[[2]]$aes_params$size <- 2.8
print(p)
```

US Crime Rate Map



Boxplot for each variable

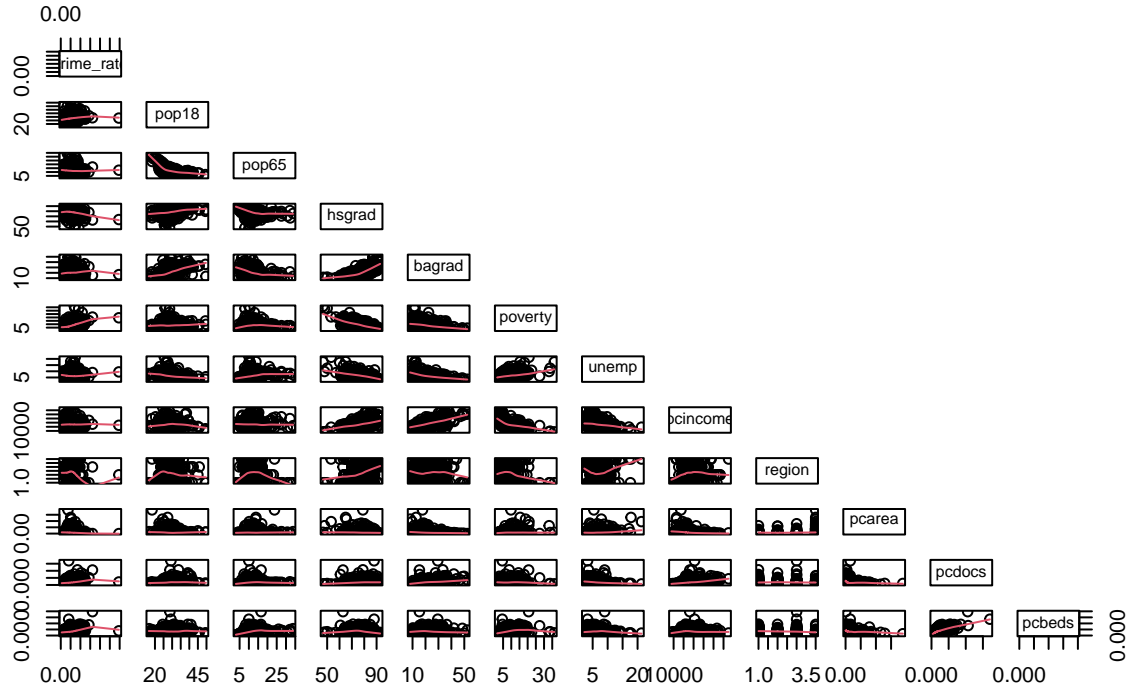
```
par(mfrow=c(3,4))
boxplot(cdi_pc$crime_rate, main = "crime_rate")
boxplot(cdi_pc$pop18, main = "pop18")
boxplot(cdi_pc$pop65, main = "pop65")
boxplot(cdi_pc$pcincome, main = "pcincome")
boxplot(cdi_pc$unemp, main = "unemp")
boxplot(cdi_pc$hsgrad, main = "hsgrad")
boxplot(cdi_pc$bagrad, main = "bagrad")
boxplot(cdi_pc$poverty, main = "poverty")
boxplot(cdi_pc$pcarea, main = "pcarea")
boxplot(cdi_pc$pcdocs, main = "pcdocs")
boxplot(cdi_pc$pcbeds, main = "pcbeds")
boxplot(crime_rate ~ region, data = cdi_pc, main = "region")
```



Scatterplot Matrix

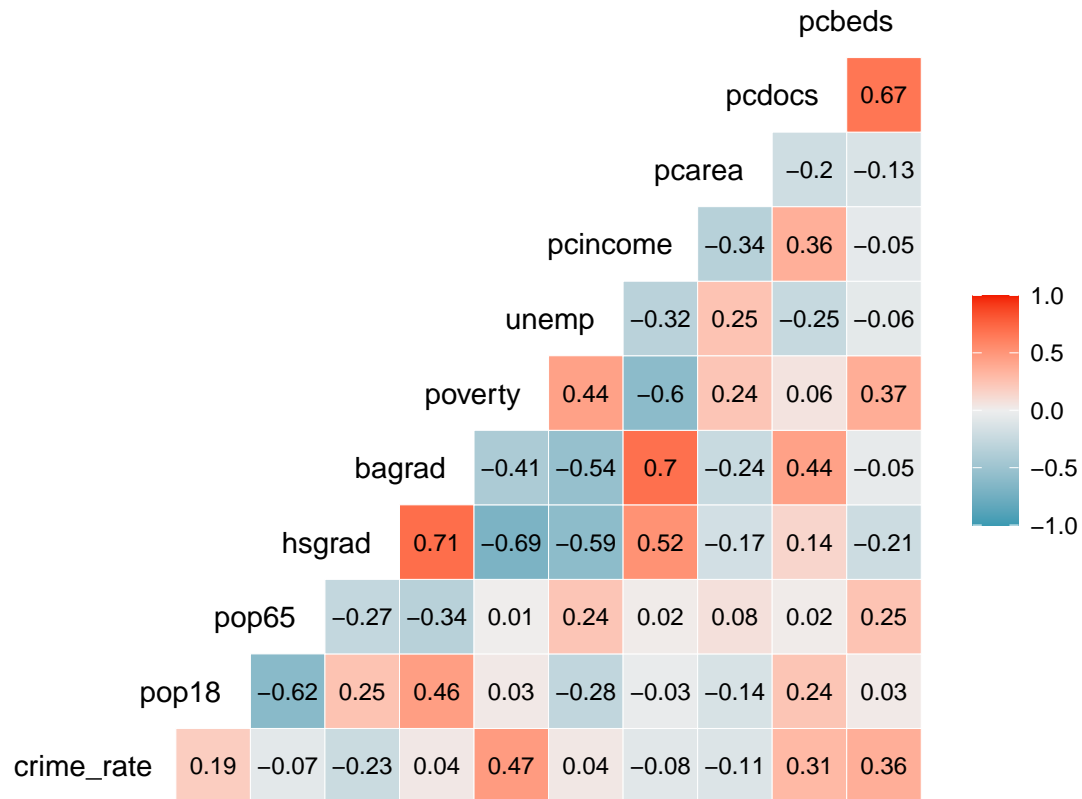
```
pairs(~crime_rate + ., data=cdi_pc, panel = panel.smooth, upper.panel = NULL, main = "Scatterplot Matrix")
```

Scatterplot Matrix



Correlation plot/ Heatmap

```
cdi_pc %>%
  dplyr::select(-region) %>%
  ggcorr(label = TRUE, hjust = 0.9, layout.exp = 2, label_size = 3, label_round = 2)
```



Modelling

Fit regression using all predictors

```
mult_fit = lm(crime_rate ~ ., data = cdi_pc)
summary(mult_fit)
```

```
##
## Call:
## lm(formula = crime_rate ~ ., data = cdi_pc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.049868 -0.010696 -0.000262  0.009115  0.222726
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.110e-02  3.107e-02  -1.645  0.100775
## pop18        1.320e-03  3.721e-04   3.547  0.000433 ***
## pop65        2.234e-04  3.488e-04   0.640  0.522201
## hsgrad       1.678e-04  3.053e-04   0.549  0.582951
## bagrad      -5.330e-04  3.417e-04  -1.560  0.119567
## poverty      2.799e-03  4.323e-04   6.475  2.62e-10 ***
## unemp        3.280e-04  6.065e-04   0.541  0.588897
## pcincome     2.251e-06  5.292e-07   4.254  2.58e-05 ***
```

```
## region1      -2.352e-02  3.025e-03  -7.776 5.71e-14 ***
## region2      -1.491e-02  2.945e-03  -5.064 6.13e-07 ***
## region4       1.269e-03  3.452e-03   0.368 0.713311
## pcarea       -6.520e-01  1.695e-01  -3.847 0.000138 ***
## pcdocs        6.635e-01  1.157e+00   0.574 0.566606
## pcbeds        2.319e+00  9.046e-01   2.564 0.010699 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02033 on 426 degrees of freedom
## Multiple R-squared:  0.4628, Adjusted R-squared:  0.4464
## F-statistic: 28.23 on 13 and 426 DF,  p-value: < 2.2e-16
```

Backwards Elimination

```
mult_fit_back <- step(mult_fit, direction='backward')
```

```
## Start:  AIC=-3414.29
## crime_rate ~ pop18 + pop65 + hsgrad + bagrad + poverty + unemp +
##      pcincome + region + pcarea + pcdocs + pcbeds
##
##           Df Sum of Sq    RSS    AIC
## - unemp     1  0.000121 0.17624 -3416.0
## - hsgrad     1  0.000125 0.17624 -3416.0
## - pcdocs     1  0.000136 0.17625 -3416.0
## - pop65      1  0.000170 0.17629 -3415.9
## <none>                0.17612 -3414.3
## - bagrad     1  0.001006 0.17712 -3413.8
## - pcbeds     1  0.002717 0.17883 -3409.6
## - pop18      1  0.005201 0.18132 -3403.5
## - pcarea     1  0.006118 0.18223 -3401.3
## - pcincome   1  0.007483 0.18360 -3398.0
## - poverty    1  0.017333 0.19345 -3375.0
## - region     3  0.033187 0.20930 -3344.3
##
## Step:  AIC=-3415.99
## crime_rate ~ pop18 + pop65 + hsgrad + bagrad + poverty + pcincome +
##      region + pcarea + pcdocs + pcbeds
##
##           Df Sum of Sq    RSS    AIC
## - hsgrad     1  0.000090 0.17633 -3417.8
## - pcdocs     1  0.000139 0.17638 -3417.6
## - pop65      1  0.000206 0.17644 -3417.5
## <none>                0.17624 -3416.0
## - bagrad     1  0.001199 0.17744 -3415.0
## - pcbeds     1  0.002597 0.17883 -3411.6
## - pop18      1  0.005296 0.18153 -3405.0
## - pcarea     1  0.006032 0.18227 -3403.2
## - pcincome   1  0.008072 0.18431 -3398.3
## - poverty    1  0.020229 0.19647 -3370.2
## - region     3  0.034275 0.21051 -3343.8
##
```



```
## Step: AIC=-3417.76
## crime_rate ~ pop18 + pop65 + bagrad + poverty + pcincome + region +
##      pcarea + pcdocs + pcbeds
##
##           Df Sum of Sq      RSS      AIC
## - pcdocs    1  0.000119 0.17645 -3419.5
## - pop65     1  0.000185 0.17651 -3419.3
## <none>                0.17633 -3417.8
## - bagrad    1  0.001366 0.17769 -3416.4
## - pcbeds    1  0.002791 0.17912 -3412.9
## - pop18     1  0.005207 0.18153 -3407.0
## - pcarea    1  0.006153 0.18248 -3404.7
## - pcincome  1  0.008474 0.18480 -3399.1
## - region    3  0.034432 0.21076 -3345.3
## - poverty   1  0.032602 0.20893 -3345.1
##
## Step: AIC=-3419.47
## crime_rate ~ pop18 + pop65 + bagrad + poverty + pcincome + region +
##      pcarea + pcbeds
##
##           Df Sum of Sq      RSS      AIC
## - pop65     1  0.000185 0.17663 -3421.0
## <none>                0.17645 -3419.5
## - bagrad    1  0.001252 0.17770 -3418.4
## - pop18     1  0.005485 0.18193 -3408.0
## - pcarea    1  0.006170 0.18262 -3406.3
## - pcbeds    1  0.007404 0.18385 -3403.4
## - pcincome  1  0.009338 0.18578 -3398.8
## - poverty   1  0.032979 0.20943 -3346.1
## - region    3  0.034895 0.21134 -3346.1
##
## Step: AIC=-3421.01
## crime_rate ~ pop18 + bagrad + poverty + pcincome + region + pcarea +
##      pcbeds
##
##           Df Sum of Sq      RSS      AIC
## <none>                0.17663 -3421.0
## - bagrad    1  0.001396 0.17803 -3419.5
## - pop18     1  0.005894 0.18252 -3408.6
## - pcarea    1  0.006021 0.18265 -3408.3
## - pcincome  1  0.009446 0.18608 -3400.1
## - pcbeds    1  0.009589 0.18622 -3399.7
## - region    3  0.034710 0.21134 -3348.1
## - poverty   1  0.032938 0.20957 -3347.8
```

```
mult_fit_back
```

```
##
## Call:
## lm(formula = crime_rate ~ pop18 + bagrad + poverty + pcincome +
##      region + pcarea + pcbeds, data = cdi_pc)
##
## Coefficients:
## (Intercept)      pop18      bagrad      poverty      pcincome      region1
```

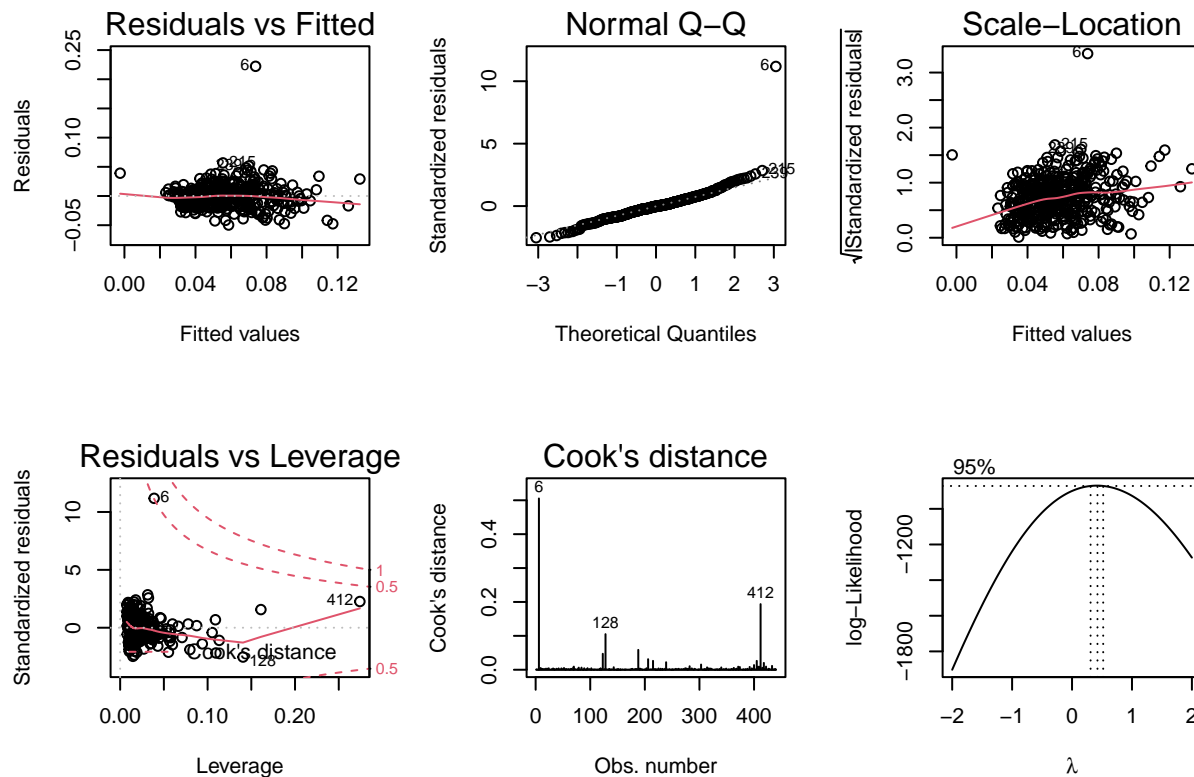
```
## -3.242e-02  1.214e-03 -4.595e-04  2.727e-03  2.288e-06  -2.279e-02
##      region2      region4      pcare      pcbeds
## -1.458e-02  2.190e-03 -6.410e-01  2.750e+00
```

```
crime_rate ~ pop18 + bagrad + poverty + pcincome + region + pcare + pcbeds
```

Model Diagnostics

Create Residuals vs Fitted plot & Normal Q-Q plot & Scale-Location plot & Residuals vs Leverage plot to detect the normality of residuals and outliers

```
par(mfrow=c(2,3))
plot(mult_fit_back)
plot(mult_fit_back, which = 4)
bc = boxcox(mult_fit_back)
```



Diagnose the model without outliers

```
# remove influential points
cdi_pc_out = cdi_pc[-c(6,128,412),]

# fit model with and without influential points
mult_fit_back_without = lm(crime_rate ~ pop18 + bagrad + poverty + pcincome + region + pcare + pcbeds,
```

```
summary(mult_fit_back)
```

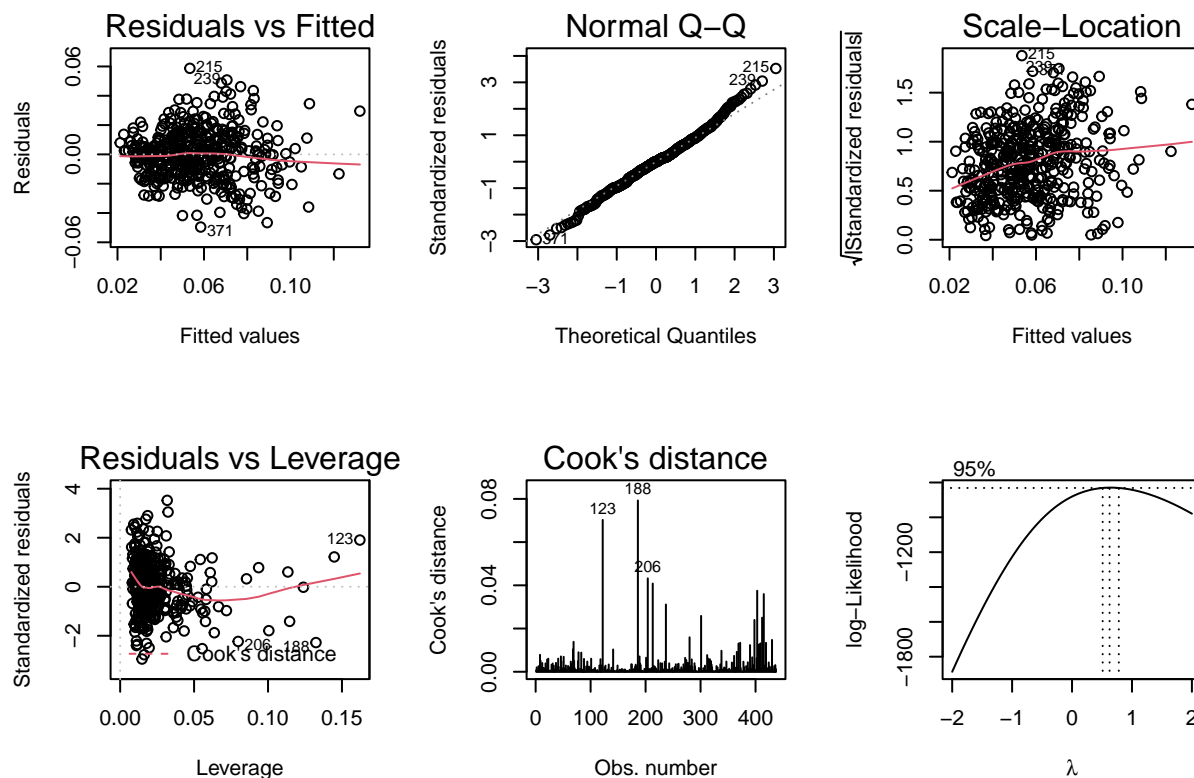
```
##
## Call:
## lm(formula = crime_rate ~ pop18 + bagrad + poverty + pcincome +
##     region + pcarea + pcbeds, data = cdi_pc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.049729 -0.010702 -0.000564  0.009817  0.222166
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.242e-02  1.279e-02  -2.534  0.011638 *
## pop18        1.214e-03  3.206e-04   3.788  0.000174 ***
## bagrad       -4.595e-04  2.492e-04  -1.844  0.065937 .
## poverty      2.727e-03  3.046e-04   8.955 < 2e-16 ***
## pcincome     2.288e-06  4.770e-07   4.795  2.24e-06 ***
## region1     -2.279e-02  2.869e-03  -7.943  1.74e-14 ***
## region2     -1.458e-02  2.711e-03  -5.377  1.25e-07 ***
## region4      2.190e-03  3.170e-03   0.691  0.489979
## pcarea      -6.410e-01  1.674e-01  -3.828  0.000148 ***
## pcbeds       2.750e+00  5.691e-01   4.831  1.89e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02027 on 430 degrees of freedom
## Multiple R-squared:  0.4612, Adjusted R-squared:  0.45
## F-statistic: 40.9 on 9 and 430 DF, p-value: < 2.2e-16
```

```
summary(mult_fit_back_without)
```

```
##
## Call:
## lm(formula = crime_rate ~ pop18 + bagrad + poverty + pcincome +
##     region + pcarea + pcbeds, data = cdi_pc_out)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.049584 -0.010326  0.000031  0.010232  0.058723
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.833e-02  1.069e-02  -2.649  0.00837 **
## pop18        1.258e-03  2.701e-04   4.659  4.25e-06 ***
## bagrad       -4.979e-04  2.099e-04  -2.372  0.01813 *
## poverty      2.432e-03  2.755e-04   8.830 < 2e-16 ***
## pcincome     2.202e-06  3.985e-07   5.526  5.72e-08 ***
## region1     -2.607e-02  2.422e-03 -10.765 < 2e-16 ***
## region2     -1.563e-02  2.273e-03  -6.876  2.20e-11 ***
## region4      2.040e-03  2.673e-03   0.763  0.44568
## pcarea      -7.446e-01  1.614e-01  -4.613  5.24e-06 ***
```

```
## pcbeds      2.927e+00  4.883e-01  5.995 4.35e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01693 on 427 degrees of freedom
## Multiple R-squared:  0.5471, Adjusted R-squared:  0.5375
## F-statistic:  57.3 on 9 and 427 DF,  p-value: < 2.2e-16
```

```
# diagnose the model without outliers
par(mfrow=c(2,3))
plot(mult_fit_back_without)
plot(mult_fit_back_without, which = 4)
bc_without = boxcox(mult_fit_back_without)
```



Box-cox transformation

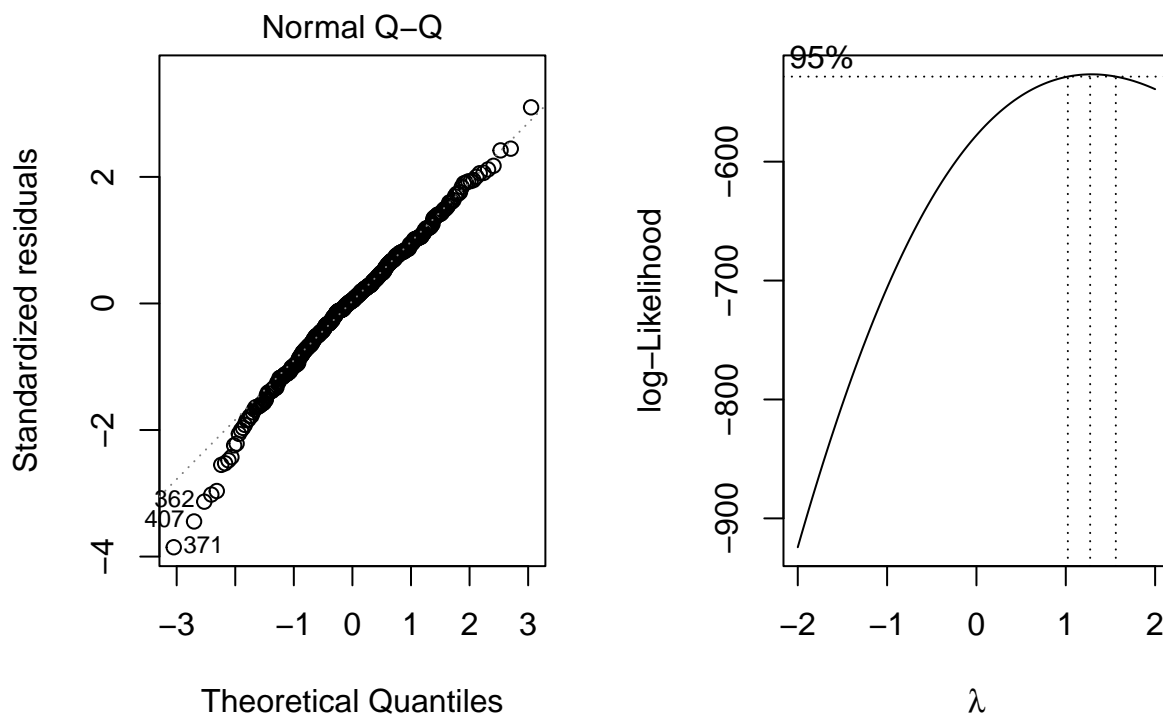
```
(lambda = bc_without$x[which.max(bc_without$y)])
```

```
## [1] 0.6262626
```

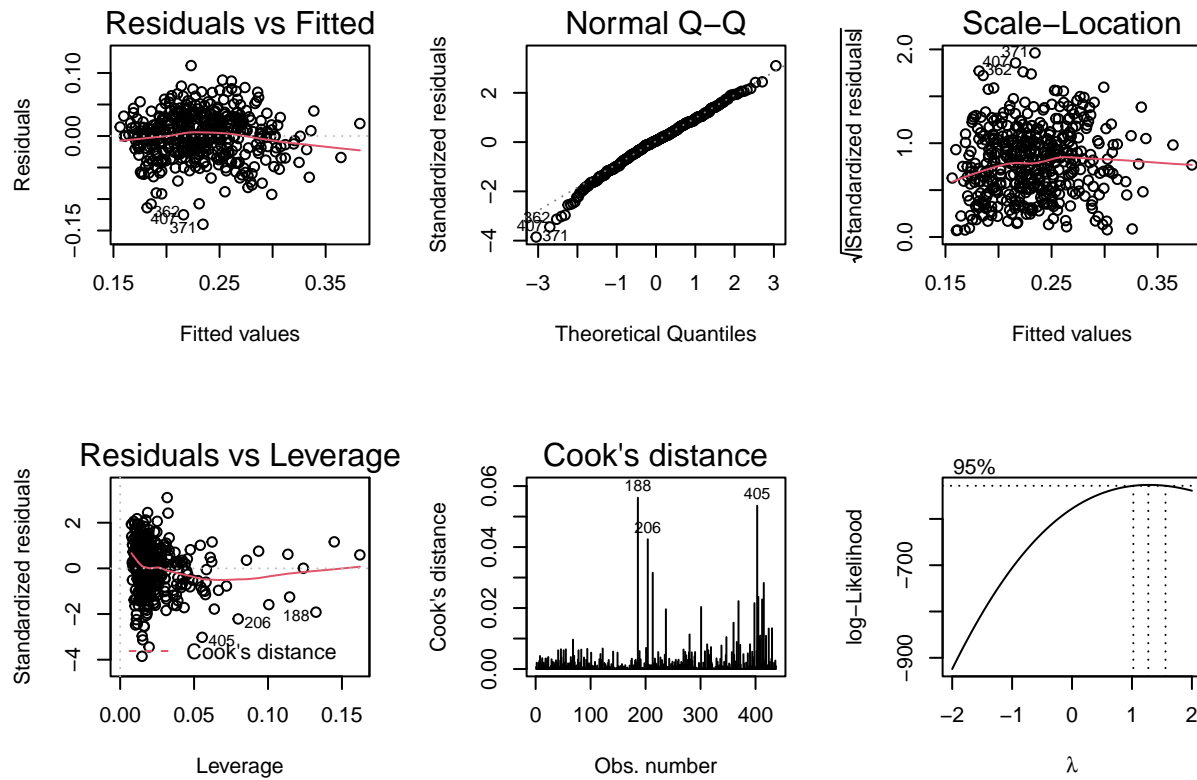
```
#mult_fit_back_without_trans = lm(((crime_rate^lambda-1)/lambda) ~ pop18 + bagrad + poverty + pcincome
mult_fit_back_without_trans = lm(crime_rate^0.5 ~ pop18 + bagrad + poverty + pcincome +
                                region + pcarea + pcbeds, data = cdi_pc_out)
summary(mult_fit_back_without_trans)
```

```
##
## Call:
## lm(formula = crime_rate^0.5 ~ pop18 + bagrad + poverty + pcincome +
##     region + pcarea + pcbeds, data = cdi_pc_out)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.140088 -0.021520  0.001828  0.024375  0.111737
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.665e-02  2.314e-02   2.448  0.0148 *
## pop18        2.657e-03  5.844e-04   4.546 7.13e-06 ***
## bagrad       -1.087e-03  4.542e-04  -2.393  0.0171 *
## poverty      4.835e-03  5.961e-04   8.111 5.36e-15 ***
## pcincome     4.687e-06  8.624e-07   5.435 9.23e-08 ***
## region1     -5.867e-02  5.241e-03 -11.193 < 2e-16 ***
## region2     -3.469e-02  4.920e-03  -7.051 7.17e-12 ***
## region4      6.480e-03  5.784e-03   1.120  0.2632
## pcarea      -1.466e+00  3.493e-01  -4.197 3.29e-05 ***
## pcbeds       5.908e+00  1.057e+00   5.591 4.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03663 on 427 degrees of freedom
## Multiple R-squared:  0.5326, Adjusted R-squared:  0.5227
## F-statistic: 54.05 on 9 and 427 DF, p-value: < 2.2e-16
```

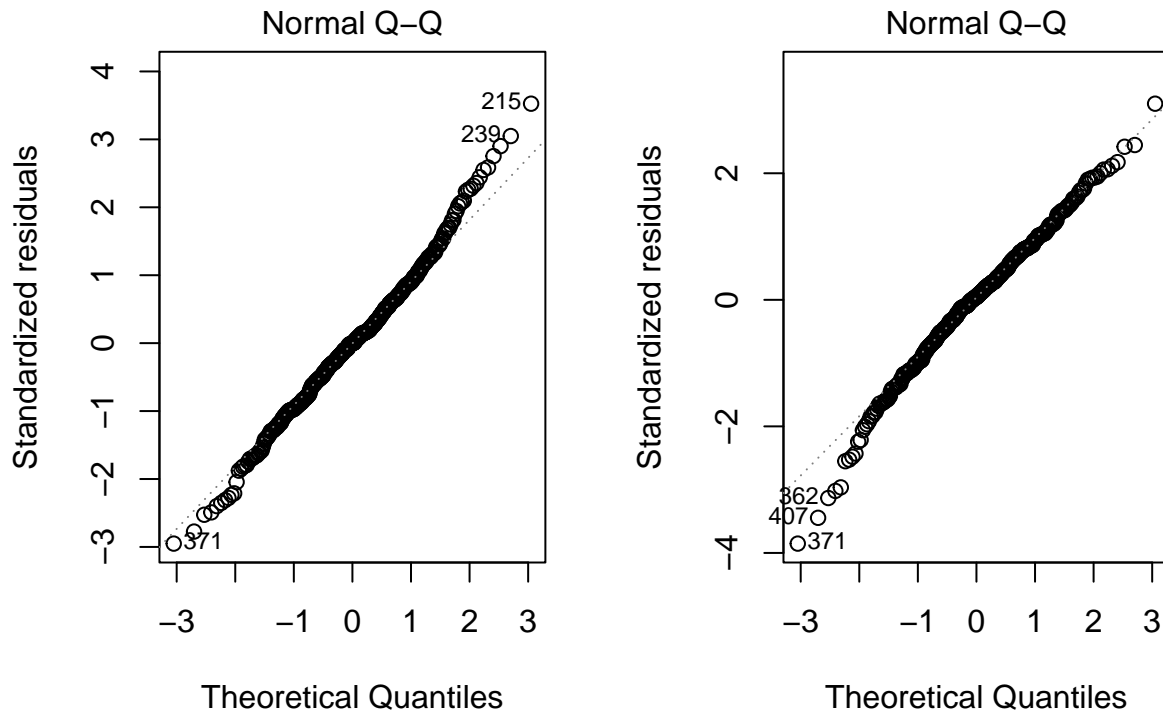
```
par(mfrow = c(1,2))
plot(mult_fit_back_without_trans, which = 2)
boxcox(mult_fit_back_without_trans)
```



```
# Diagnose the model by square root transformation
par(mfrow = c(2,3))
plot(mult_fit_back_without_trans)
plot(mult_fit_back_without_trans, which = 4)
bc_without_trans = boxcox(mult_fit_back_without_trans)
```



```
# Compare the qq plot
par(mfrow = c(1,2))
plot(mult_fit_back_without, which = 2)
plot(mult_fit_back_without_trans, which = 2)
```



Compare the Adjusted R^2

```

rbind(mult_fit_back %>% broom::glance() %>% mutate(model_type = "mult_fit_back"),
      mult_fit_back_without %>% broom::glance() %>% mutate(model_type = "mult_fit_back_without"),
      mult_fit_back_without_trans %>% broom::glance() %>% mutate(model_type = "mult_fit_back_without_trans"),
      dplyr::select(model_type, everything()))

```

```

## # A tibble: 3 x 13
##   model_type      r.squared adj.r.squared  sigma statistic  p.value    df logLik
##   <chr>          <dbl>      <dbl>  <dbl>    <dbl>    <dbl> <dbl> <dbl>
## 1 mult_fit_back    0.461        0.450 0.0203     40.9 1.57e-52     9 1096.
## 2 mult_fit_back_~  0.547        0.538 0.0169     57.3 5.69e-68     9 1167.
## 3 mult_fit_back_~  0.533        0.523 0.0366     54.1 4.34e-65     9  830.
## # ... with 5 more variables: AIC <dbl>, BIC <dbl>, deviance <dbl>,
## #   df.residual <int>, nobs <int>

```

Assessing Multicollinearity

```

# Calculate the variance inflation factor (VIF)
check_collinearity(mult_fit_back_without)

```

```

## # Check for Multicollinearity
##
## Low Correlation
##
##      Term  VIF Increased SE Tolerance

```

```
## pcincome 1.00      1.00      1.00
## pcarea 2.02      1.42      0.49
##
## Moderate Correlation
##
##      Term  VIF Increased SE Tolerance
##  pop18 7.83      2.80      0.13
##  bagrad 6.83      2.61      0.15
##  region 6.15      2.48      0.16
##  pcbeds 6.29      2.51      0.16
##
## High Correlation
##
##      Term  VIF Increased SE Tolerance
##  poverty 13.19      3.63      0.08
```

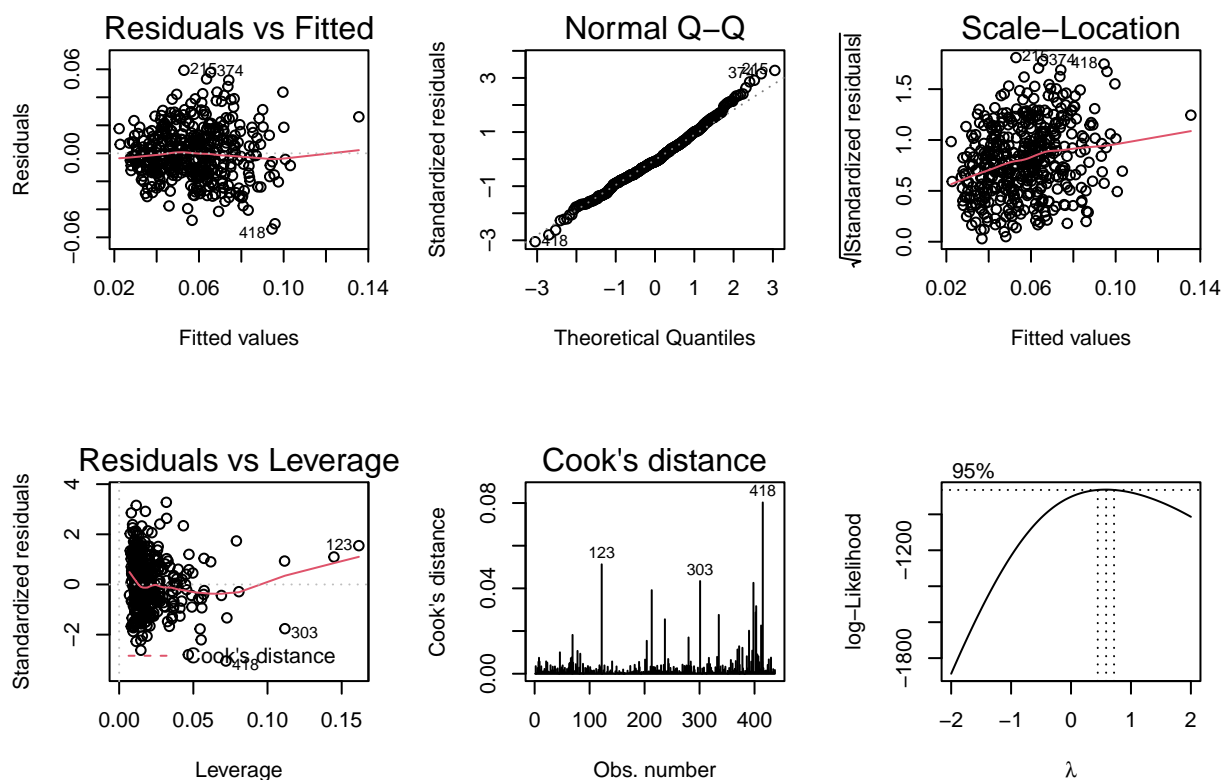
```
# Remove the variable whose vif is larger than 10
```

```
mult_fit_back_without_vif = lm(crime_rate ~ pop18 + bagrad + pcincome + region + pcarea + pcbeds, data = cdi_pc_out)
summary(mult_fit_back_without_vif)
```

```
##
## Call:
## lm(formula = crime_rate ~ pop18 + bagrad + pcincome + region +
##      pcarea + pcbeds, data = cdi_pc_out)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.054061 -0.011870 -0.001622  0.010885  0.059278
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.686e-03  1.088e-02   0.431  0.66696
## pop18        1.442e-03  2.925e-04   4.932 1.17e-06 ***
## bagrad       -6.808e-04  2.269e-04  -3.001  0.00285 **
## pcincome     1.140e-06  4.127e-07   2.764  0.00596 **
## region1     -3.105e-02  2.559e-03 -12.137 < 2e-16 ***
## region2     -2.138e-02  2.366e-03  -9.034 < 2e-16 ***
## region4      1.904e-03  2.903e-03   0.656  0.51234
## pcarea      -5.425e-01  1.735e-01  -3.126  0.00189 **
## pcbeds       5.111e+00  4.573e-01  11.176 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01839 on 428 degrees of freedom
## Multiple R-squared:  0.4644, Adjusted R-squared:  0.4543
## F-statistic: 46.38 on 8 and 428 DF, p-value: < 2.2e-16
```

```
# Diagnose the model removing poverty term
```

```
par(mfrow = c(2,3))
plot(mult_fit_back_without_vif)
plot(mult_fit_back_without_vif, which = 4)
bc_without_trans = boxcox(mult_fit_back_without_vif)
```

```
# Add the interaction
```

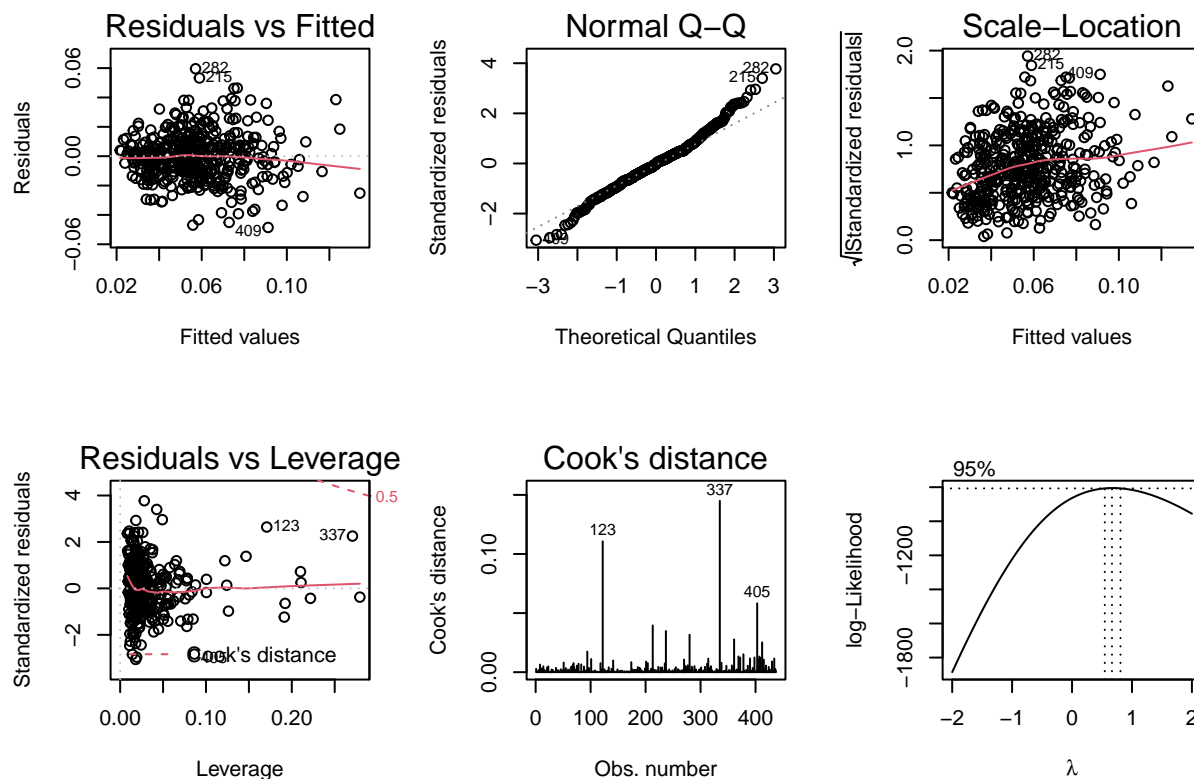
```
mult_fit_back_without_int = lm(crime_rate ~ pop18 + bagrad + pcincome + region + pcarea + pcbeds + poverty)
summary(mult_fit_back_without_int)
```

```
##
## Call:
## lm(formula = crime_rate ~ pop18 + bagrad + pcincome + region +
##      pcarea + pcbeds + poverty + pcincome * poverty + pcincome *
##      bagrad + pop18 * bagrad, data = cdi_pc_out)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.048555 -0.009658 -0.000259  0.007877  0.059566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.851e-02  2.027e-02  -2.393  0.017154 *
## pop18          1.008e-03  5.271e-04   1.912  0.056516 .
## bagrad         1.243e-03  7.670e-04   1.621  0.105802
## pcincome       3.169e-06  9.379e-07   3.378  0.000796 ***
## region1      -2.550e-02  2.321e-03 -10.983 < 2e-16 ***
## region2      -1.526e-02  2.161e-03  -7.061  6.82e-12 ***
## region4      -2.490e-03  2.616e-03  -0.952  0.341857
## pcarea       -4.545e-01  1.582e-01  -2.872  0.004278 **
## pcbeds        1.327e+00  5.209e-01   2.547  0.011210 *
## poverty      -4.214e-04  7.964e-04  -0.529  0.597008
## pcincome:poverty  2.295e-07  5.077e-08   4.521  7.99e-06 ***
## bagrad:pcincome -7.267e-08  2.288e-08  -3.176  0.001600 **
```

```
## pop18:bagrad      -3.205e-06  2.123e-05  -0.151 0.880068
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01601 on 424 degrees of freedom
## Multiple R-squared:  0.5978, Adjusted R-squared:  0.5864
## F-statistic: 52.52 on 12 and 424 DF,  p-value: < 2.2e-16
```

```
# Diagnose the model with interaction term
```

```
par(mfrow = c(2,3))
plot(mult_fit_back_without_int)
plot(mult_fit_back_without_int, which = 4)
bc_without_trans = boxcox(mult_fit_back_without_int)
```



Compare the Adjusted R² again

```
rbind(mult_fit_back %>% broom::glance() %>% mutate(model_type = "mult_fit_back"),
      mult_fit_back_without %>% broom::glance() %>% mutate(model_type = "mult_fit_back_without"),
      mult_fit_back_without_trans %>% broom::glance() %>% mutate(model_type = "mult_fit_back_without_trans"),
      mult_fit_back_without_vif %>% broom::glance() %>% mutate(model_type = "mult_fit_back_without_vif"),
      mult_fit_back_without_int %>% broom::glance() %>% mutate(model_type = "mult_fit_back_without_int"),
      dplyr::select(model_type, everything()))
```

```
## # A tibble: 5 x 13
##   model_type      r.squared adj.r.squared sigma statistic p.value    df logLik
##   <chr>          <dbl>         <dbl>   <dbl>    <dbl>    <dbl> <dbl> <dbl>
```

```
## 1 mult_fit_back      0.461      0.450 0.0203      40.9 1.57e-52      9 1096.
## 2 mult_fit_back_~    0.547      0.538 0.0169      57.3 5.69e-68      9 1167.
## 3 mult_fit_back_~    0.533      0.523 0.0366      54.1 4.34e-65      9 830.
## 4 mult_fit_back_~    0.464      0.454 0.0184      46.4 1.63e-53      8 1131.
## 5 mult_fit_back_~    0.598      0.586 0.0160      52.5 4.09e-76     12 1193.
## # ... with 5 more variables: AIC <dbl>, BIC <dbl>, deviance <dbl>,
## #   df.residual <int>, nobs <int>
```

Model Validation

Compute RMSE by cross-validation

```
set.seed(1234)
cv_df =
  crossv_kfold(cdi_pc_out, k = 5) %>%
  mutate(
    train = map(train, as_tibble),
    test = map(test, as_tibble)) %>%
  mutate(
    mult_fit_back_without = map(train, ~lm(crime_rate ~ pop18 + bagrad + poverty +
      pcincome + region + pcarea + pcbeds, data = .x)),
    mult_fit_back_without_trans = map(train, ~lm(sqrt(crime_rate) ~ pop18 + bagrad + poverty +
      pcincome + region + pcarea + pcbeds, data = .x)),
    mult_fit_back_without_vif = map(train, ~lm(crime_rate ~ pop18 + bagrad +
      pcincome + region + pcarea + pcbeds, data = .x)),
    mult_fit_back_without_int = map(train, ~lm(crime_rate ~ pop18 + bagrad + pcincome + region + pcarea +
      pcincome*poverty + pcincome*bagrad + pop18*bagrad, data = .x)))
  mutate(
    rmse_model1 = map2_dbl(mult_fit_back_without, test, ~rmse(model = .x, data = .y)),
    rmse_model2 = map2_dbl(mult_fit_back_without_trans, test, ~rmse(model = .x, data = .y)),
    rmse_model3 = map2_dbl(mult_fit_back_without_vif, test, ~rmse(model = .x, data = .y)),
    rmse_model4 = map2_dbl(mult_fit_back_without_int, test, ~rmse(model = .x, data = .y)))

cv_df %>%
  dplyr::select(rmse_model1, rmse_model2, rmse_model3, rmse_model4) %>%
  dplyr::summarise_each(funs(mean(., na.rm = TRUE)))
```

```
## Warning: 'summarise_each()' was deprecated in dplyr 0.7.0.
## Please use 'across()' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.
```

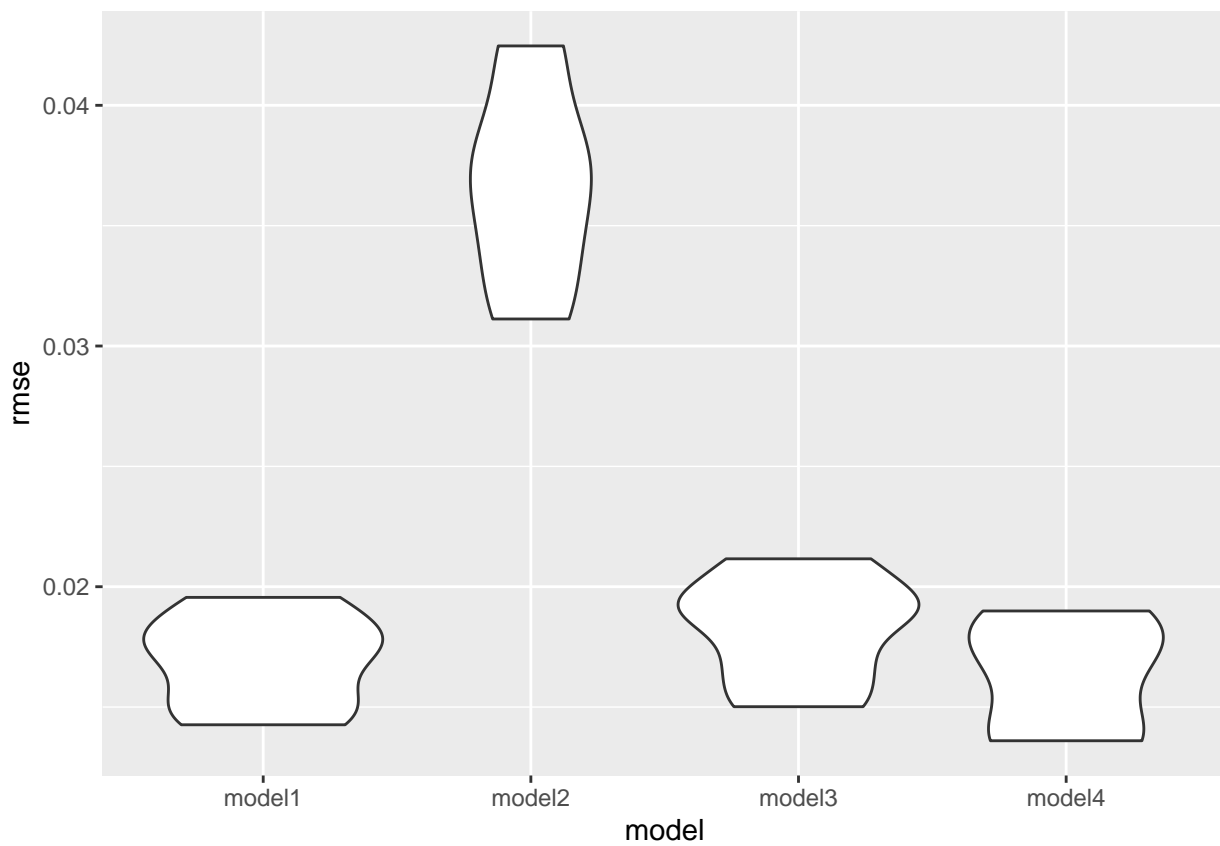
```
## Warning: 'funs()' was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with 'tibble::lst()':
##   tibble::lst(mean, median)
```

```
##
## # Using lambdas
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.

## # A tibble: 1 x 4
##   rmse_model1 rmse_model2 rmse_model3 rmse_model4
##   <dbl>      <dbl>      <dbl>      <dbl>
## 1      0.0169      0.0366      0.0183      0.0163
```

Plot the violin plot

```
cv_df %>%
  dplyr::select(starts_with("rmse")) %>%
  pivot_longer(
    everything(),
    names_to = "model",
    values_to = "rmse",
    names_prefix = "rmse_"
  ) %>%
  mutate(model = fct_inorder(model)) %>%
  ggplot(aes(x = model, y = rmse)) + geom_violin()
```



Thanks for reading!