STAT 425 Case Study 1

Zachary Ryan (zmryan2) & Sam Burch (sgburch2)

2022-10-17

Preliminary

```
cdi = read.csv('CDI.txt', header = FALSE, sep='')

head(cdi)
```

```
##   V1          V2 V3   V4      V5   V6   V7    V8    V9    V10  V11  V12  V13
## 1  1 Los_Angeles CA 4060 8863164 32.1  9.7 23677 27700 688936 70.0 22.3 11.6
## 2  2        Cook IL  946 5105067 29.2 12.4 15153 21550 436936 73.4 22.8 11.1
## 3  3      Harris TX 1729 2818199 31.3  7.1  7553 12449 253526 74.9 25.4 12.5
## 4  4   San_Diego CA 4205 2498016 33.5 10.9  5905  6179 173821 81.9 25.3  8.1
## 5  5      Orange CA  790 2410556 32.6  9.2  6062  6369 144524 81.2 27.8  5.2
## 6  6       Kings NY   71 2300664 28.3 12.4  4861  8942 680966 63.7 16.6 19.5
##   V14   V15    V16 V17
## 1 8.0 20786 184230   4
## 2 7.2 21729 110928   2
## 3 5.7 19517  55003   3
## 4 6.1 19588  48931   4
## 5 4.8 24400  58818   4
## 6 9.5 16803  38658   1
```

```
dim(cdi)
```

```
## [1] 440  17
```

```
names(cdi) = c('id', 'county', 'state', 'land_area', 'pop', 'pop_rate_young',
               'pop_rate_old', 'active_physicians', 'hospital_beds',
               'serious_crimes', 'hs_grad_rate', 'bachelor_deg_rate',
               'below_poverty_rate', 'unemployment_rate', 'per_cap_income',
               'personal_income', 'geo_region')
head(cdi)
```

```
##   id      county state land_area     pop pop_rate_young pop_rate_old
## 1  1 Los_Angeles    CA      4060 8863164           32.1          9.7
## 2  2        Cook    IL       946 5105067           29.2         12.4
## 3  3      Harris    TX      1729 2818199           31.3          7.1
## 4  4   San_Diego    CA      4205 2498016           33.5         10.9
## 5  5      Orange    CA       790 2410556           32.6          9.2
## 6  6       Kings    NY        71 2300664           28.3         12.4
##   active_physicians hospital_beds serious_crimes hs_grad_rate bachelor_deg_rate
## 1             23677         27700         688936         70.0              22.3
## 2             15153         21550         436936         73.4              22.8
## 3              7553         12449         253526         74.9              25.4
## 4              5905          6179         173821         81.9              25.3
## 5              6062          6369         144524         81.2              27.8
## 6              4861          8942         680966         63.7              16.6
##   below_poverty_rate unemployment_rate per_cap_income personal_income
## 1               11.6               8.0          20786          184230
## 2               11.1               7.2          21729          110928
## 3               12.5               5.7          19517           55003
## 4                8.1               6.1          19588           48931
## 5                5.2               4.8          24400           58818
## 6               19.5               9.5          16803           38658
##   geo_region
## 1          4
## 2          2
## 3          3
## 4          4
## 5          4
## 6          1
```

```
dim(cdi)
```

```
## [1] 440  17
```

Pre-Testing

```
cdi$hospital_beds_rate = cdi$hospital_beds/cdi$pop
cdi$serious_crimes_rate = cdi$serious_crimes/cdi$pop
cdi = cdi[, -c(9, 10)]

df_1 = cdi[-c(1, 2, 3)]

cor(df_1[, -5])
```

```
##                        land_area          pop pop_rate_young pop_rate_old
## land_area            1.000000000  0.173083353    -0.05487812  0.005770871
## pop                  0.173083353  1.000000000     0.07837212 -0.029037393
## pop_rate_young      -0.054878125  0.078372117     1.00000000 -0.616309639
## pop_rate_old         0.005770871 -0.029037393    -0.61630964  1.000000000
## hs_grad_rate        -0.098598111 -0.017426900     0.25058429 -0.268251758
## bachelor_deg_rate   -0.137237736  0.146813850     0.45609703 -0.339228765
## below_poverty_rate   0.171343348  0.038019509     0.03397551  0.006578474
## unemployment_rate    0.199209277  0.005351703    -0.27852706  0.236309411
## per_cap_income      -0.187715132  0.235610188    -0.03164843  0.018590706
## personal_income      0.127074261  0.986747626     0.07116151 -0.022733151
## geo_region           0.362868243  0.069437072     0.05241407 -0.173291567
## hospital_beds_rate  -0.141233520  0.020301218     0.02952439  0.247147869
## serious_crimes_rate  0.042948447  0.280099222     0.19056876 -0.066533283
##                     hs_grad_rate bachelor_deg_rate below_poverty_rate
## land_area            -0.09859811       -0.13723774        0.171343348
## pop                  -0.01742690        0.14681385        0.038019509
## pop_rate_young        0.25058429        0.45609703        0.033975512
## pop_rate_old         -0.26825176       -0.33922877        0.006578474
## hs_grad_rate          1.00000000        0.70778672       -0.691750483
## bachelor_deg_rate     0.70778672        1.00000000       -0.408423848
## below_poverty_rate   -0.69175048       -0.40842385        1.000000000
## unemployment_rate    -0.59359579       -0.54090691        0.436947236
## per_cap_income        0.52299613        0.69536186       -0.601725039
## personal_income       0.04335573        0.22223013       -0.038739339
## geo_region           -0.01005506        0.02029897        0.270984846
## hospital_beds_rate   -0.21116247       -0.04541826        0.371398926
## serious_crimes_rate  -0.22641291        0.03830458        0.471844218
##                     unemployment_rate per_cap_income personal_income
## land_area                 0.199209277    -0.18771513     0.127074261
## pop                       0.005351703     0.23561019     0.986747626
## pop_rate_young           -0.278527058    -0.03164843     0.071161515
## pop_rate_old              0.236309411     0.01859071    -0.022733151
## hs_grad_rate             -0.593595788     0.52299613     0.043355729
## bachelor_deg_rate        -0.540906913     0.69536186     0.222230125
## below_poverty_rate        0.436947236    -0.60172504    -0.038739339
## unemployment_rate         1.000000000    -0.32214439    -0.033876330
## per_cap_income           -0.322144395     1.00000000     0.347681610
## personal_income          -0.033876330     0.34768161     1.000000000
## geo_region               -0.054378572    -0.22249375     0.037685456
## hospital_beds_rate       -0.062487824    -0.05355004     0.006323904
## serious_crimes_rate       0.041846579    -0.08024417     0.228155749
##                      geo_region hospital_beds_rate serious_crimes_rate
## land_area            0.36286824       -0.141233520          0.04294845
## pop                  0.06943707        0.020301218          0.28009922
## pop_rate_young       0.05241407        0.029524392          0.19056876
## pop_rate_old        -0.17329157        0.247147869         -0.06653328
## hs_grad_rate        -0.01005506       -0.211162472         -0.22641291
## bachelor_deg_rate    0.02029897       -0.045418264          0.03830458
## below_poverty_rate   0.27098485        0.371398926          0.47184422
## unemployment_rate   -0.05437857       -0.062487824          0.04184658
## per_cap_income      -0.22249375       -0.053550037         -0.08024417
```

```
## personal_income      0.03768546        0.006323904          0.22815575
## geo_region           1.00000000       -0.113622302          0.34275842
## hospital_beds_rate  -0.11362230        1.000000000          0.36445047
## serious_crimes_rate  0.34275842        0.364450470          1.00000000
```

We created rate metrics for beds and crimes by dividing them by pop.

Personal income high correlation with pop (0.987), small correlation with others. We will remove pop since it was also used to create the hospital beds and serious crimes rate variables. Leave county, state out because the same info in geo_region

```
df_1 = df_1[,-2]

cor(df_1[, -4])
```

```
##                      land_area pop_rate_young pop_rate_old hs_grad_rate
## land_area           1.000000000    -0.05487812  0.005770871  -0.09859811
## pop_rate_young     -0.054878125     1.00000000 -0.616309639   0.25058429
## pop_rate_old        0.005770871    -0.61630964  1.000000000  -0.26825176
## hs_grad_rate       -0.098598111     0.25058429 -0.268251758   1.00000000
## bachelor_deg_rate  -0.137237736     0.45609703 -0.339228765   0.70778672
## below_poverty_rate  0.171343348     0.03397551  0.006578474  -0.69175048
## unemployment_rate   0.199209277    -0.27852706  0.236309411  -0.59359579
## per_cap_income     -0.187715132    -0.03164843  0.018590706   0.52299613
## personal_income     0.127074261     0.07116151 -0.022733151   0.04335573
## geo_region          0.362868243     0.05241407 -0.173291567  -0.01005506
## hospital_beds_rate -0.141233520     0.02952439  0.247147869  -0.21116247
## serious_crimes_rate  0.042948447     0.19056876 -0.066533283  -0.22641291
##                    bachelor_deg_rate below_poverty_rate unemployment_rate
## land_area                -0.13723774        0.171343348        0.19920928
## pop_rate_young            0.45609703        0.033975512       -0.27852706
## pop_rate_old             -0.33922877        0.006578474        0.23630941
## hs_grad_rate              0.70778672       -0.691750483       -0.59359579
## bachelor_deg_rate         1.00000000       -0.408423848       -0.54090691
## below_poverty_rate       -0.40842385        1.000000000        0.43694724
## unemployment_rate        -0.54090691        0.436947236        1.00000000
## per_cap_income            0.69536186       -0.601725039       -0.32214439
## personal_income           0.22223013       -0.038739339       -0.03387633
## geo_region                0.02029897        0.270984846       -0.05437857
## hospital_beds_rate       -0.04541826        0.371398926       -0.06248782
## serious_crimes_rate       0.03830458        0.471844218        0.04184658
##                    per_cap_income personal_income  geo_region
## land_area             -0.18771513     0.127074261  0.36286824
## pop_rate_young        -0.03164843     0.071161515  0.05241407
## pop_rate_old           0.01859071    -0.022733151 -0.17329157
## hs_grad_rate           0.52299613     0.043355729 -0.01005506
## bachelor_deg_rate      0.69536186     0.222230125  0.02029897
## below_poverty_rate    -0.60172504    -0.038739339  0.27098485
## unemployment_rate     -0.32214439    -0.033876330 -0.05437857
## per_cap_income         1.00000000     0.347681610 -0.22249375
## personal_income        0.34768161     1.000000000  0.03768546
## geo_region            -0.22249375     0.037685456  1.00000000
## hospital_beds_rate    -0.05355004     0.006323904 -0.11362230
## serious_crimes_rate   -0.08024417     0.228155749  0.34275842
##                    hospital_beds_rate serious_crimes_rate
## land_area                -0.141233520          0.04294845
## pop_rate_young            0.029524392          0.19056876
## pop_rate_old              0.247147869         -0.06653328
## hs_grad_rate             -0.211162472         -0.22641291
## bachelor_deg_rate        -0.045418264          0.03830458
## below_poverty_rate        0.371398926          0.47184422
## unemployment_rate        -0.062487824          0.04184658
## per_cap_income           -0.053550037         -0.08024417
## personal_income           0.006323904          0.22815575
## geo_region               -0.113622302          0.34275842
```

```
## hospital_beds_rate        1.000000000         0.36445047
## serious_crimes_rate       0.364450470         1.00000000
```

Now that all correlations are under absolute value of 0.9 we should be able to start doing testing-based model selection without collinearity impacting the p-values.

Initial Testing-Based Model Selection

```
mlr_full = lm(active_physicians ~ ., df_1)
summary(mlr_full)
```

```
##
## Call:
## lm(formula = active_physicians ~ ., data = df_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1515.49  -230.03   -11.33   173.15  2895.33
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -6.238e+02  6.623e+02  -0.942   0.3468
## land_area           -2.198e-02  1.563e-02  -1.407   0.1603
## pop_rate_young       1.083e+01  8.255e+00   1.312   0.1902
## pop_rate_old         4.902e+00  7.492e+00   0.654   0.5133
## hs_grad_rate        -8.442e+00  6.126e+00  -1.378   0.1689
## bachelor_deg_rate    1.735e+01  6.971e+00   2.488   0.0132 *
## below_poverty_rate   1.700e+01  9.903e+00   1.716   0.0868 .
## unemployment_rate   -2.971e+00  1.288e+01  -0.231   0.8177
## per_cap_income      -6.308e-03  1.188e-02  -0.531   0.5956
## personal_income      1.305e-01  1.911e-03  68.306   <2e-16 ***
## geo_region          -3.577e+00  2.665e+01  -0.134   0.8933
## hospital_beds_rate   1.442e+05  1.424e+04  10.122   <2e-16 ***
## serious_crimes_rate -2.414e+01  1.045e+03  -0.023   0.9816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 445.4 on 427 degrees of freedom
## Multiple R-squared:  0.9398, Adjusted R-squared:  0.9381
## F-statistic: 555.2 on 12 and 427 DF,  p-value: < 2.2e-16
```

Individual t-test show bachelor_deg_rate, personal_income, and hospital_beds_rate to be statistically significant, with α = .05. The F-test shows p-value of ~0, which leads to the conclusion that at least one β is not equal to 0. Note that land_area, pop_rate_young, hs_grad_rate, and below_poverty_rate have relatively low p-values. Also, the most significant are personal_income, and hospital_beds_rate (with p-values ~0)

Let's now consider a model where only the predictors mentioned above are used.

```r
mlr_red_1 = lm(active_physicians ~ land_area + pop_rate_young +
                 hs_grad_rate + bachelor_deg_rate +
                 below_poverty_rate + personal_income + hospital_beds_rate,
               data=df_1)
summary(mlr_red_1)
```

```
##
## Call:
## lm(formula = active_physicians ~ land_area + pop_rate_young +
##     hs_grad_rate + bachelor_deg_rate + below_poverty_rate + personal_income +
##     hospital_beds_rate, data = df_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1505.01  -231.30    -5.93   170.66  2877.10
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.076e+02  4.319e+02  -1.638  0.10210
## land_area          -2.163e-02  1.455e-02  -1.487  0.13781
## pop_rate_young      1.051e+01  5.926e+00   1.774  0.07675 .
## hs_grad_rate       -7.828e+00  5.602e+00  -1.397  0.16299
## bachelor_deg_rate   1.469e+01  4.520e+00   3.249  0.00125 **
## below_poverty_rate  1.775e+01  7.130e+00   2.490  0.01316 *
## personal_income     1.301e-01  1.740e-03  74.810  < 2e-16 ***
## hospital_beds_rate  1.470e+05  1.178e+04  12.475  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 443.2 on 432 degrees of freedom
## Multiple R-squared:  0.9397, Adjusted R-squared:  0.9387
## F-statistic:   961 on 7 and 432 DF,  p-value: < 2.2e-16
```

```r
anova(mlr_red_1, mlr_full)
```

```
## Analysis of Variance Table
##
## Model 1: active_physicians ~ land_area + pop_rate_young + hs_grad_rate +
##     bachelor_deg_rate + below_poverty_rate + personal_income +
##     hospital_beds_rate
## Model 2: active_physicians ~ land_area + pop_rate_young + pop_rate_old +
##     hs_grad_rate + bachelor_deg_rate + below_poverty_rate + unemployment_rate +
##     per_cap_income + personal_income + geo_region + hospital_beds_rate +
##     serious_crimes_rate
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    432 84859564
## 2    427 84690592  5    168972 0.1704 0.9735
```

Here, our null stated the reduced model is adequate, while the alternate stated it is not. With a p-value of .97 >>> α = .05 (much greater), we can say the reduced model (mlr_red_1) is adequate!

Now, let's take this one step further and only use the predictors that had a p-value < 0.1.

```
mlr_red_2 = lm(active_physicians ~ bachelor_deg_rate + personal_income
               + below_poverty_rate + hospital_beds_rate + pop_rate_young,
               data=df_1)
summary(mlr_red_2)
```

```
##
## Call:
## lm(formula = active_physicians ~ bachelor_deg_rate + personal_income +
##     below_poverty_rate + hospital_beds_rate + pop_rate_young,
##     data = df_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1474.05  -238.65   -13.41   172.72  2955.36
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.309e+03  1.522e+02  -8.601  < 2e-16 ***
## bachelor_deg_rate   1.132e+01  3.667e+00   3.088  0.00214 **
## personal_income     1.301e-01  1.695e-03  76.787  < 2e-16 ***
## below_poverty_rate  2.177e+01  5.638e+00   3.861  0.00013 ***
## hospital_beds_rate  1.511e+05  1.153e+04  13.107  < 2e-16 ***
## pop_rate_young      1.026e+01  5.928e+00   1.731  0.08410 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 444.6 on 434 degrees of freedom
## Multiple R-squared:  0.939,  Adjusted R-squared:  0.9383
## F-statistic:  1336 on 5 and 434 DF,  p-value: < 2.2e-16
```

```
anova(mlr_red_2, mlr_red_1)
```

```
## Analysis of Variance Table
##
## Model 1: active_physicians ~ bachelor_deg_rate + personal_income + below_poverty_rate +
##     hospital_beds_rate + pop_rate_young
## Model 2: active_physicians ~ land_area + pop_rate_young + hs_grad_rate +
##     bachelor_deg_rate + below_poverty_rate + personal_income +
##     hospital_beds_rate
##   Res.Df      RSS Df Sum of Sq     F  Pr(>F)
## 1    434 85800497
## 2    432 84859564  2    940933 2.395 0.09238 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mlr_red_2, mlr_full)
```

```
## Analysis of Variance Table
##
## Model 1: active_physicians ~ bachelor_deg_rate + personal_income + below_poverty_rate +
##     hospital_beds_rate + pop_rate_young
## Model 2: active_physicians ~ land_area + pop_rate_young + pop_rate_old +
##     hs_grad_rate + bachelor_deg_rate + below_poverty_rate + unemployment_rate +
##     per_cap_income + personal_income + geo_region + hospital_beds_rate +
##     serious_crimes_rate
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    434 85800497
## 2    427 84690592  7   1109905 0.7994 0.5881
```

Here, both nulls state the reduced model (mlr_red_2) is adequate, while the alternates state it is not. With both partial F-tests producing p-values higher than α = 0.05, we can conclude mlr_red_2 is adequate compared to the prior two models.

Finally we will test out removing pop_rate_young which had a p-value of 0.08>α=0.05

```
mlr_red_3 = lm(active_physicians ~ bachelor_deg_rate + personal_income
               + below_poverty_rate + hospital_beds_rate,
               data=df_1)
summary(mlr_red_3)
```

```
##
## Call:
## lm(formula = active_physicians ~ bachelor_deg_rate + personal_income +
##     below_poverty_rate + hospital_beds_rate, data = df_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1513.66  -209.41   -13.76   191.11  2970.08
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.103e+03  9.499e+01 -11.612  < 2e-16 ***
## bachelor_deg_rate   1.460e+01  3.149e+00   4.634 4.74e-06 ***
## personal_income     1.300e-01  1.696e-03  76.628  < 2e-16 ***
## below_poverty_rate  2.444e+01  5.435e+00   4.496 8.90e-06 ***
## hospital_beds_rate  1.500e+05  1.154e+04  13.001  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 445.7 on 435 degrees of freedom
## Multiple R-squared:  0.9386, Adjusted R-squared:  0.938
## F-statistic:  1661 on 4 and 435 DF,  p-value: < 2.2e-16
```

```
anova(mlr_red_3, mlr_red_2)
```

```
## Analysis of Variance Table
##
## Model 1: active_physicians ~ bachelor_deg_rate + personal_income + below_poverty_rate +
##     hospital_beds_rate
## Model 2: active_physicians ~ bachelor_deg_rate + personal_income + below_poverty_rate +
##     hospital_beds_rate + pop_rate_young
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    435 86393121
## 2    434 85800497  1    592624 2.9976 0.0841 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, the null states the reduced model (mlr_red_3) is adequate, while the alternate states it is not (pop_rate_young is required). With the p-value 0.084 greater than α = .05, we can say the reduced model (mlr_red_3) is adequate when compared to mlr_red_2! Thus, this is the best model out of the 4 models we tested.

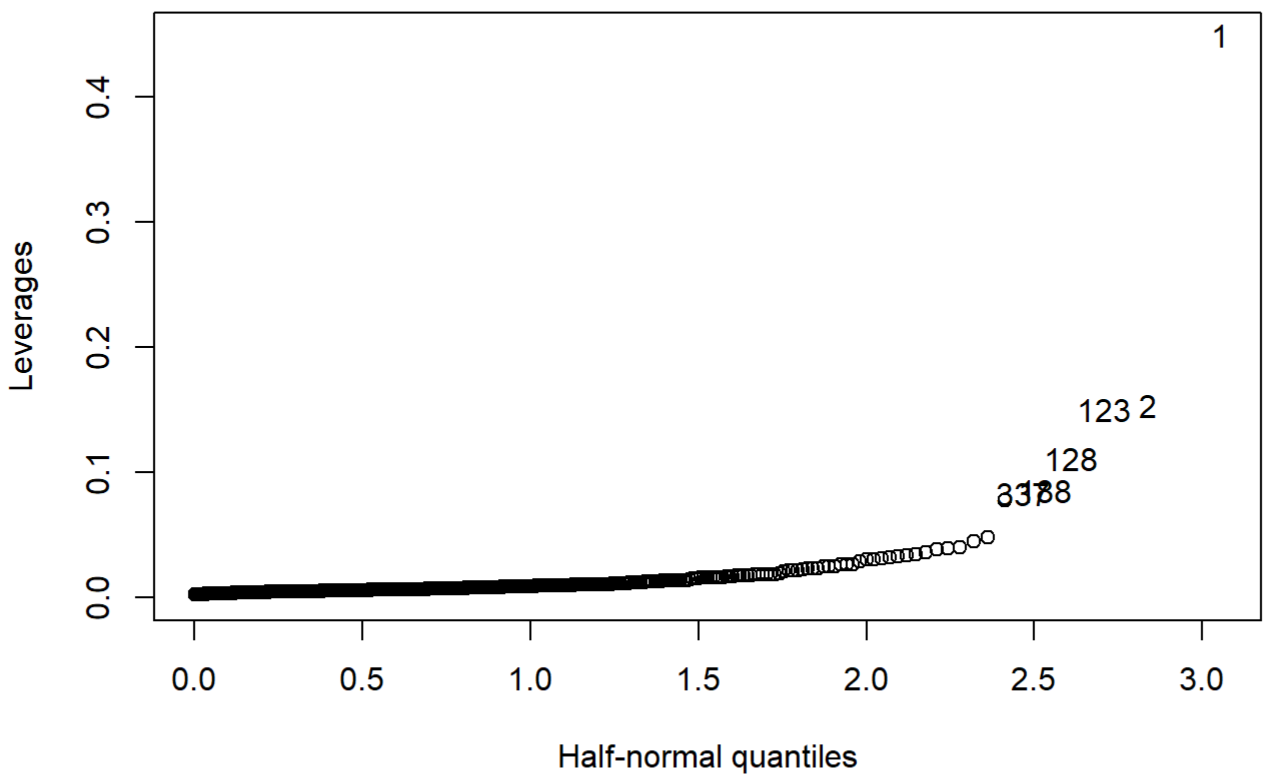
Unusual Observations

High Leverage Points (HLPs)

```
cdi.leverages = lm.influence(mlr_red_3)$hat
head(cdi.leverages)
```

```
##          1          2          3          4          5          6
## 0.44956429 0.15389090 0.03488152 0.02639698 0.03908954 0.03017191
```

```
library(faraway)
halfnorm(cdi.leverages, nlab=6, labs=as.character(1:length(cdi.leverages)), ylab="Leverages")
```




```
n = dim(cdi)[1];
p = length(variable.names(mlr_red_3));
(2*p/n)
```

```
## [1] 0.02272727
```

```
cdi.leverages.high = cdi.leverages[cdi.leverages > (2*p/n)]
(cdi.leverages.high = sort(abs(cdi.leverages.high), decreasing = TRUE))
```

```
##          1          2        123        128        188        337        418
## 0.44956429 0.15389090 0.15022731 0.11133179 0.08604913 0.08331015 0.07813890
##         95        272         42          5        396         48          3
## 0.04825668 0.04510643 0.03998096 0.03908954 0.03875937 0.03649826 0.03488152
##        248        334        392        363        400          6         76
## 0.03371716 0.03335194 0.03181376 0.03153155 0.03078272 0.03017191 0.02912249
##         70          4        404        214        168        206         67
## 0.02645637 0.02639698 0.02626699 0.02473183 0.02471154 0.02463776 0.02374233
##        422          8
## 0.02320030 0.02318085
```

```
length(cdi.leverages.high)
```

```
## [1] 30
```

This tells us there are 30 HLPs.

Trying to find Good vs Bad HLPs:

```
IQR_ap = IQR(cdi$active_physicians)

QT1_ap = quantile(cdi$active_physicians, .25)
QT3_ap = quantile(cdi$active_physicians, .75)

lower_lim = QT1_ap - IQR_ap
upper_lim = QT3_ap + IQR_ap

vector_lim = c(lower_lim, upper_lim)
vector_lim
```

```
##     25%     75%
## -670.50 1889.25
```

```
cdi.highlev = cdi[cdi.leverages > (2*p/n), ]

cdi.highlev_lower = cdi.highlev[cdi.highlev$active_physicians < vector_lim[1], ]
cdi.highlev_upper = cdi.highlev[cdi.highlev$active_physicians > vector_lim[2], ]
cdi.highlev2 = rbind(cdi.highlev_lower, cdi.highlev_upper)
cdi.highlev2
```

```
##      id         county state land_area     pop pop_rate_young pop_rate_old
## 1     1    Los_Angeles    CA      4060 8863164           32.1          9.7
## 2     2           Cook    IL       946 5105067           29.2         12.4
## 3     3         Harris    TX      1729 2818199           31.3          7.1
## 4     4      San_Diego    CA      4205 2498016           33.5         10.9
## 5     5         Orange    CA       790 2410556           32.6          9.2
## 6     6          Kings    NY        71 2300664           28.3         12.4
## 8     8          Wayne    MI       614 2111687           27.4         12.5
## 48   48     Montgomery    MD       495  757027           28.6         10.2
## 67   67        Suffolk    MA        59  663906           39.2         12.1
## 70   70         Fulton    GA       529  648951           31.6         10.0
## 95   95        Orleans    LA       181  496938           28.3         13.0
## 123 123 St._Louis_City    MO        62  396685           28.7         16.6
## 168 168      Washtenaw    MI       710  282937           39.5          7.5
##     active_physicians hs_grad_rate bachelor_deg_rate below_poverty_rate
## 1               23677         70.0              22.3               11.6
## 2               15153         73.4              22.8               11.1
## 3                7553         74.9              25.4               12.5
## 4                5905         81.9              25.3                8.1
## 5                6062         81.2              27.8                5.2
## 6                4861         63.7              16.6               19.5
## 8                3823         70.0              13.7               16.9
## 48               4635         90.6              49.9                2.7
## 67               5674         75.4              27.7               14.4
## 70               3368         77.8              31.6               15.4
## 95               2500         68.1              22.4               27.3
## 123              4189         62.8              15.3               20.6
## 168              2188         87.2              41.9                6.4
##     unemployment_rate per_cap_income personal_income geo_region
## 1                 8.0          20786          184230          4
## 2                 7.2          21729          110928          2
## 3                 5.7          19517           55003          3
## 4                 6.1          19588           48931          4
## 5                 4.8          24400           58818          4
## 6                 9.5          16803           38658          1
## 8                10.0          17461           36872          2
## 48                3.3          30081           22772          3
## 67                8.7          23150           15369          1
## 70                5.3          22819           14808          3
## 95                6.1          16578            8238          3
## 123               9.0          18113            7185          2
## 168               6.0          22782            6446          2
##     hospital_beds_rate serious_crimes_rate
## 1          0.003125295          0.07773026
## 2          0.004221296          0.08558869
## 3          0.004417360          0.08996029
## 4          0.002473563          0.06958362
## 5          0.002642129          0.05995463
## 6          0.003886704          0.29598672
## 8          0.004494037          0.09185926
## 48         0.001990682          0.04590853
## 67         0.009269385          0.10364118
```

```
## 70         0.008871240          0.14334672
## 95         0.008085516          0.10914440
## 123        0.019698249          0.16159673
## 168        0.006114435          0.06844987
```

```
nrow(cdi.highlev2)
```

```
## [1] 13
```

13 of the 30 are Bad HLPs.

Outliers

```
cdi.resid = rstudent(mlr_red_3)

bonferroni_cv = qt(.05/(2*n), n-p-1)
bonferroni_cv
```

```
## [1] -3.895092
```

```
cdi.resid.sorted = sort(abs(cdi.resid), decreasing = TRUE)[1:10]
print(cdi.resid.sorted)
```

```
##       50       11       67       48       53        9       28        5
## 7.099874 6.354710 6.233780 3.915906 3.848537 3.632280 3.458266 3.260094
##       15       17
## 3.256509 2.966872
```

```
cdi.outliers = cdi.resid.sorted[abs(cdi.resid.sorted) > abs(bonferroni_cv)]
print(cdi.outliers)
```

```
##       50       11       67       48
## 7.099874 6.354710 6.233780 3.915906
```

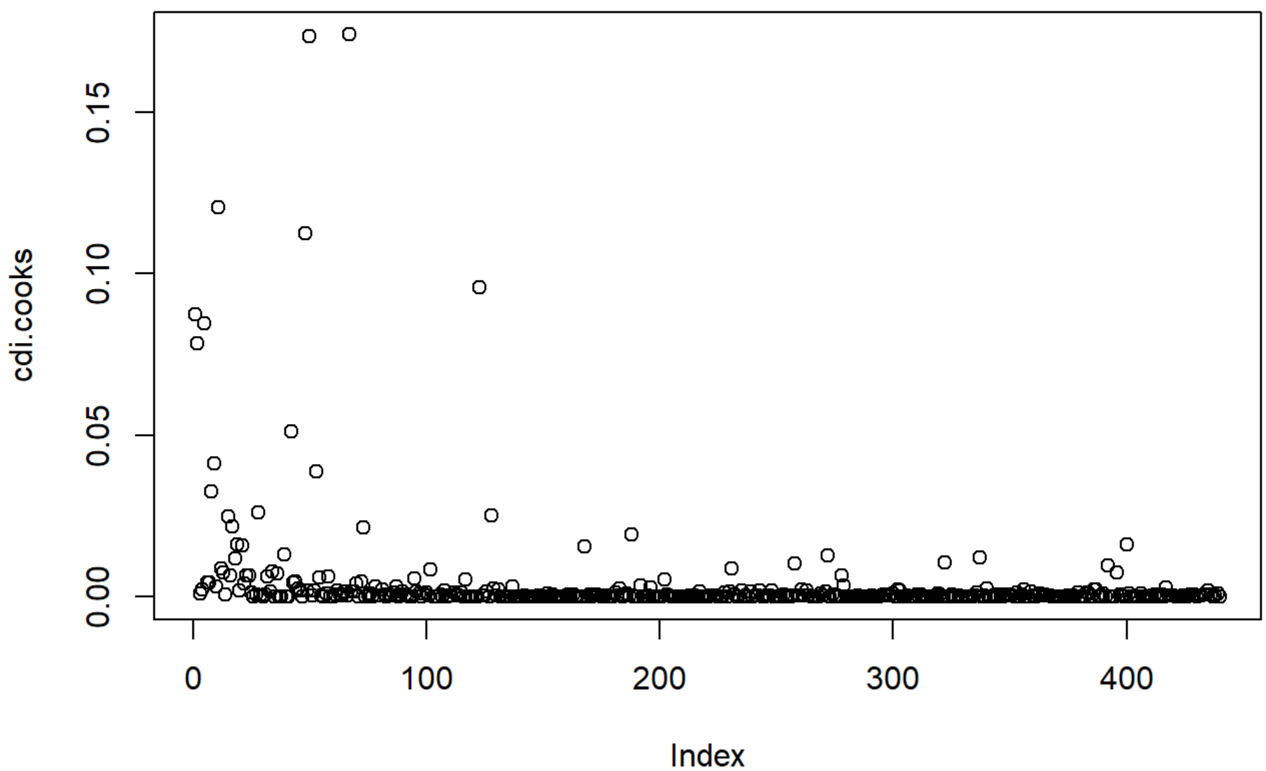
4 outliers.

Highly Influential Points (HIPs)

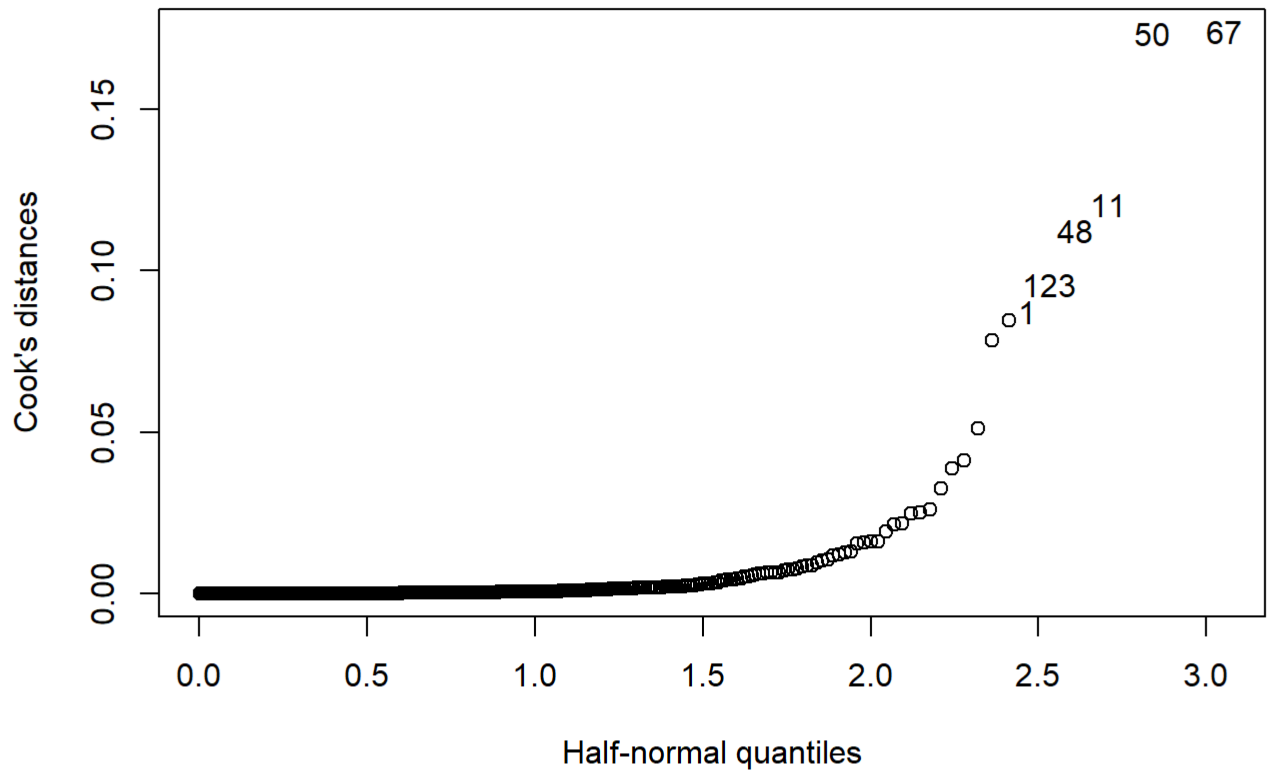
```
cdi.cooks = cooks.distance(mlr_red_3)
sort(cdi.cooks, decreasing = TRUE)[1:10]
```

```
##          67          50          11          48         123           1           5
## 0.17387954 0.17324210 0.12037705 0.11246926 0.09581769 0.08747078 0.08459801
##           2          42           9
## 0.07832728 0.05112855 0.04123943
```

```
plot(cdi.cooks)
```




```
halfnorm(cdi.cooks, 6, labs=as.character(1:length(cdi.cooks)), ylab="Cook's distances")
```

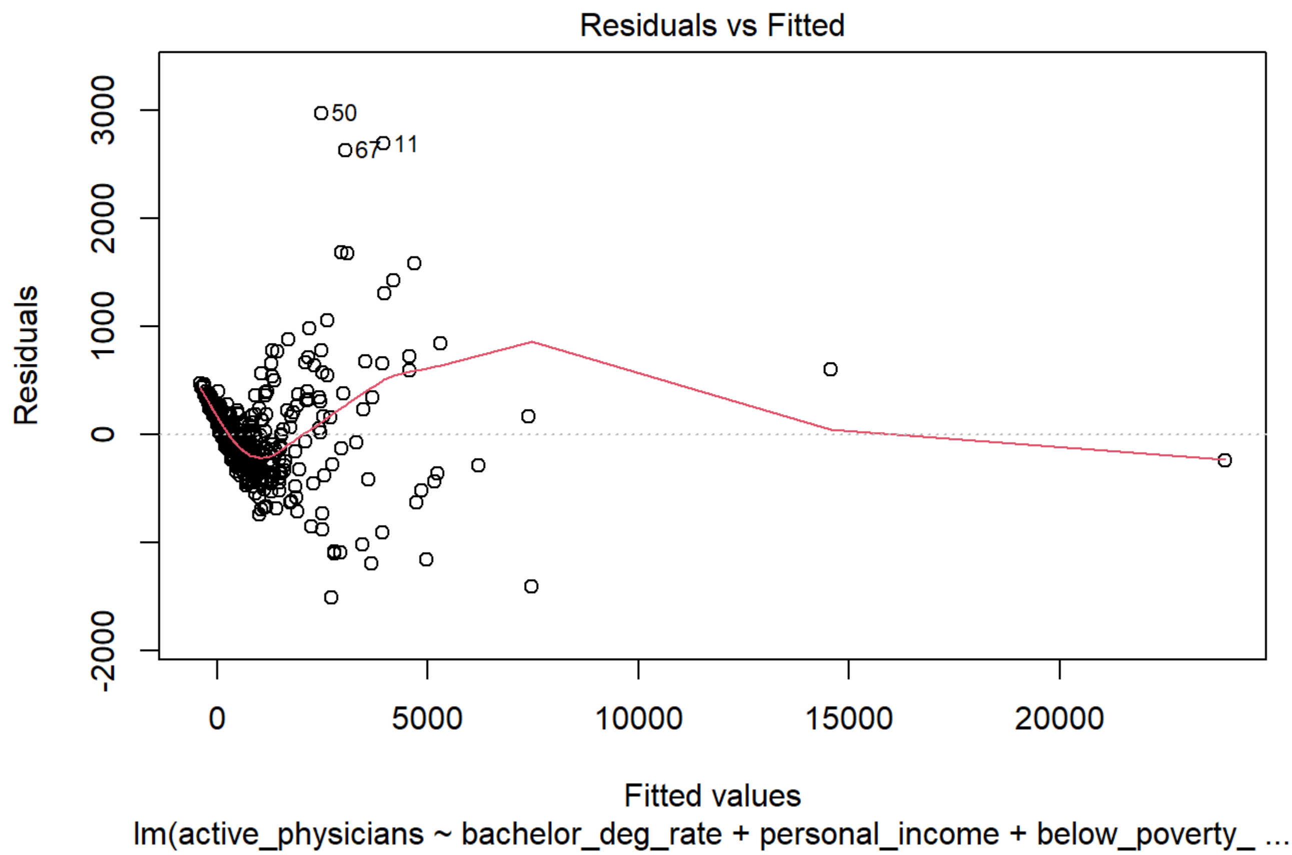


No HIPs because CD < 1.

Checking Model Assumptions

Constant Variance

```
plot(mlr_red_3, which = 1)
```




```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

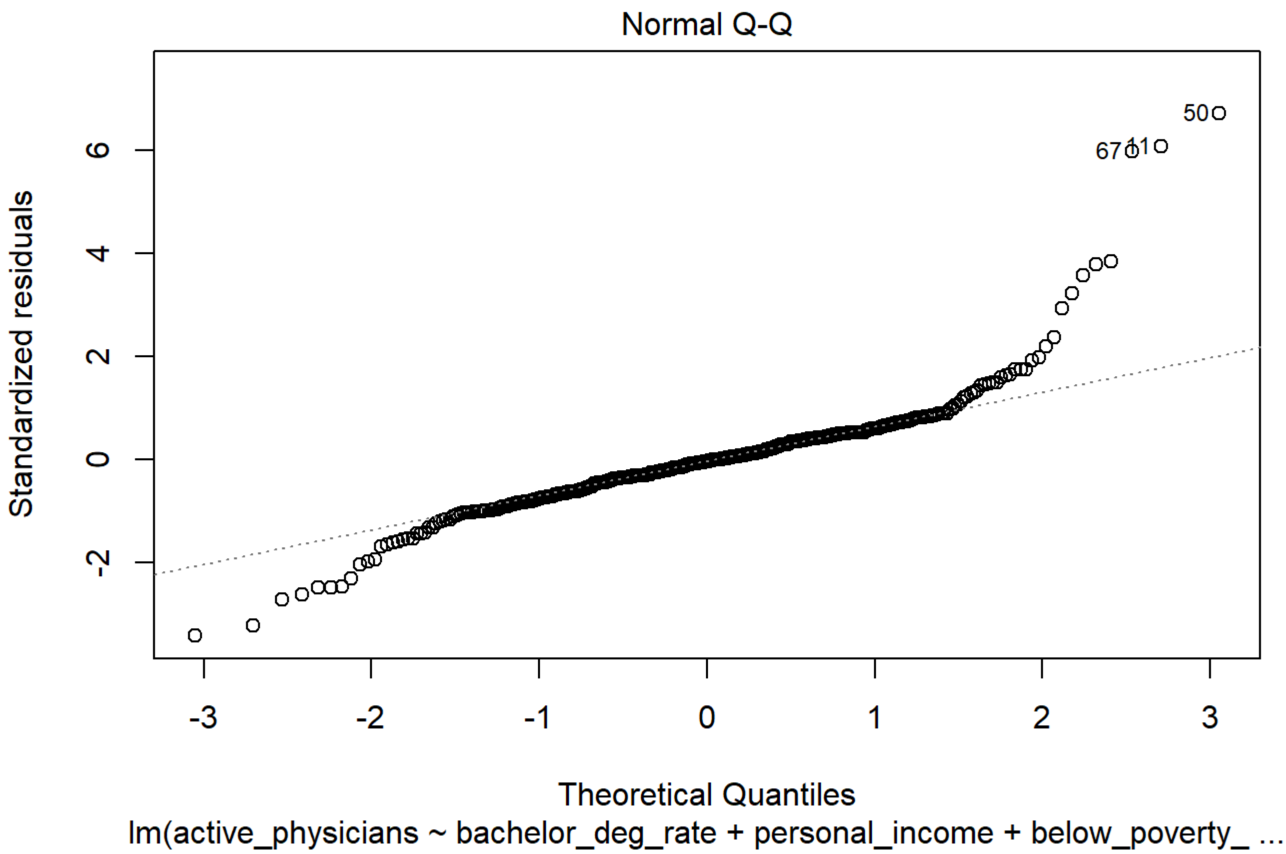
```
bptest(mlr_red_3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mlr_red_3
## BP = 45.319, df = 4, p-value = 3.413e-09
```

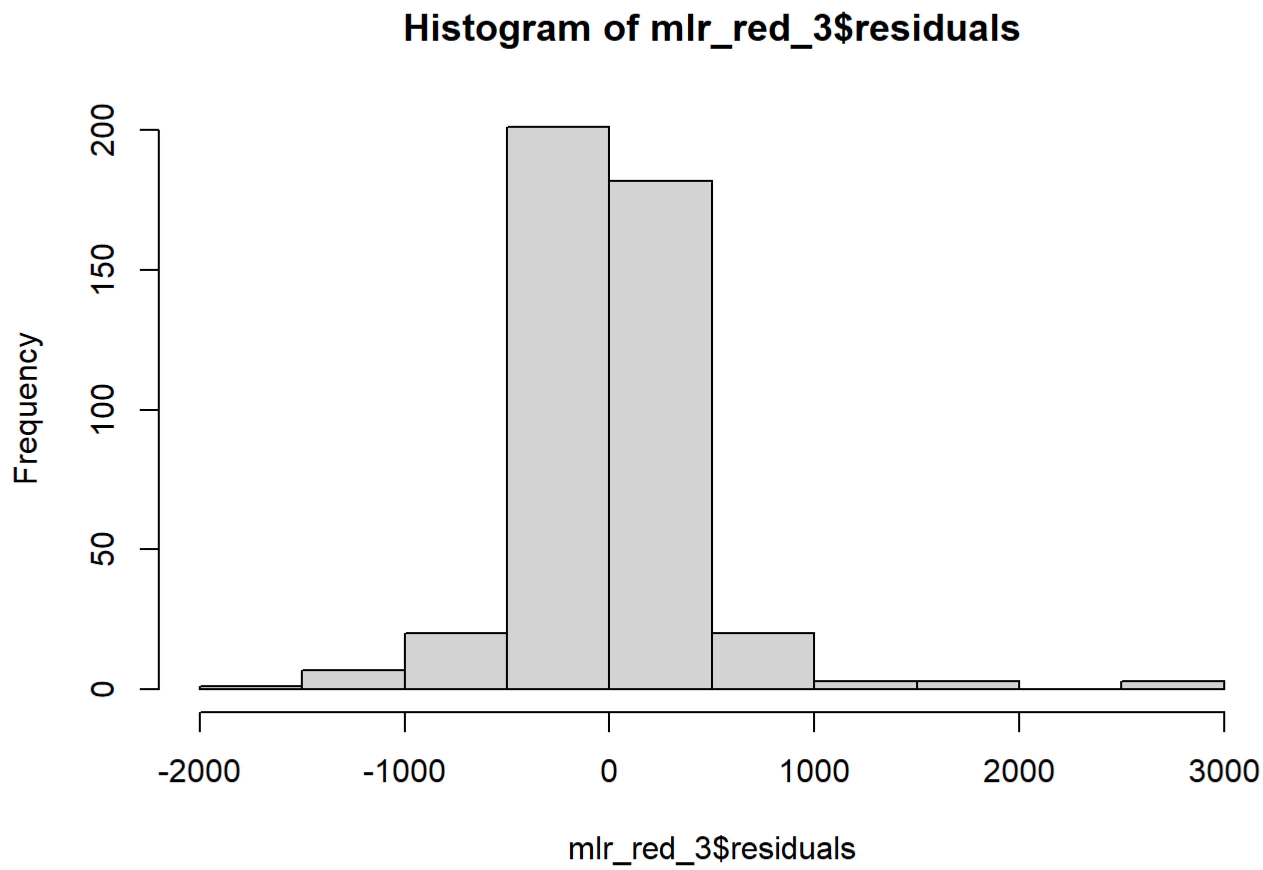
Constant variance is NOT satisfied –> TRANSFORM

Normality

```
plot(mlr_red_3, which = 2)
```




```
hist(mlr_red_3$residuals)
```




```
ks.test(mlr_red_3$residuals, y = 'pnorm')
```

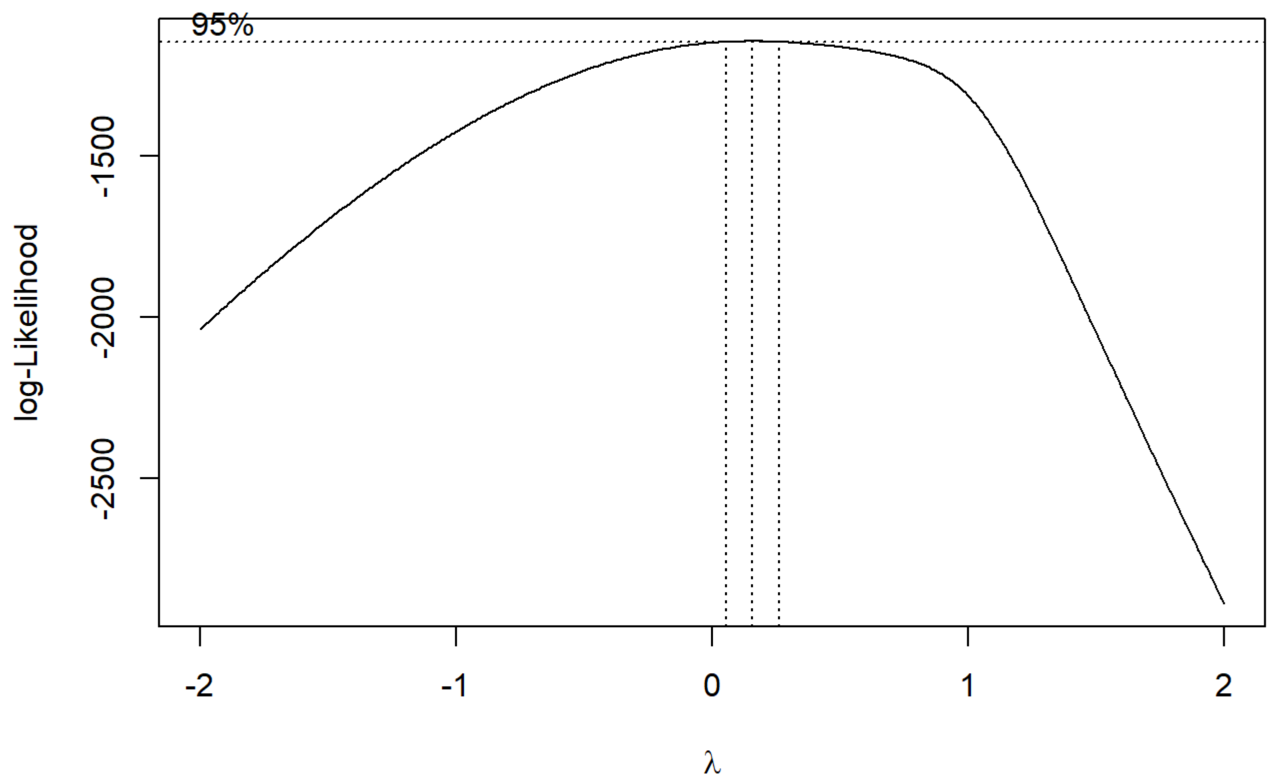
```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  mlr_red_3$residuals
## D = 0.51136, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Normal assumption not satisfied –> TRANSFORM

We will perform further diagnostics after attempting transformations to satisfy the constant variance/normality assumptions

Box-Cox Transformation

```
library(MASS)
bc_full = boxcox(mlr_red_3, lambda=seq(-2,2, length=400))
```




```
lambda <- bc_full$x[which.max(bc_full$y)]
lambda
```

```
## [1] 0.1553885
```

For better interpretability, we will choose lambda of 0 (Y=log(Y))

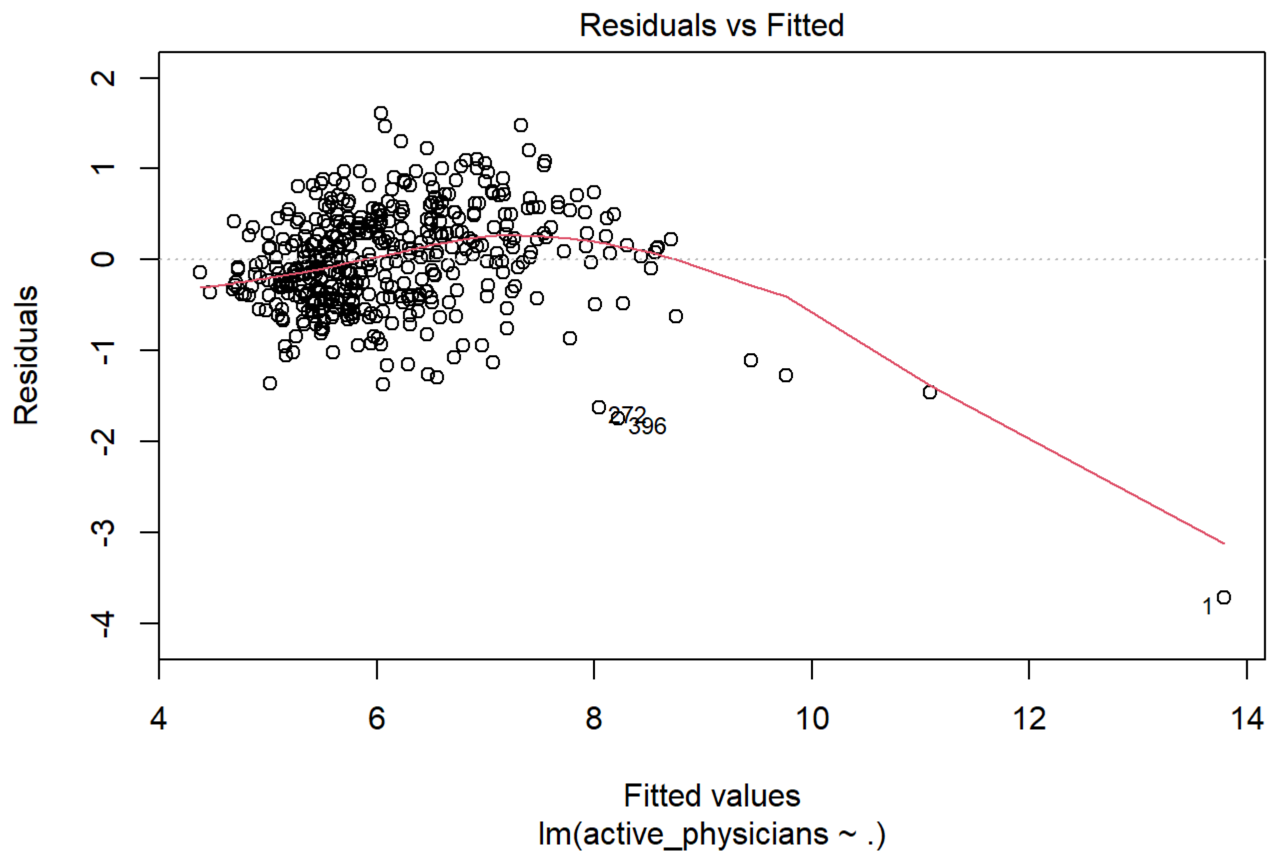
```
df_2 = df_1
df_2$active_physicians = log(df_2$active_physicians)

mlr_full_bc = lm(active_physicians ~ .,
               data=df_2)
```

Re-testing Assumptions After Box-Cox Transformation

Constant Variance

```
plot(mlr_full_bc, which = 1)
```



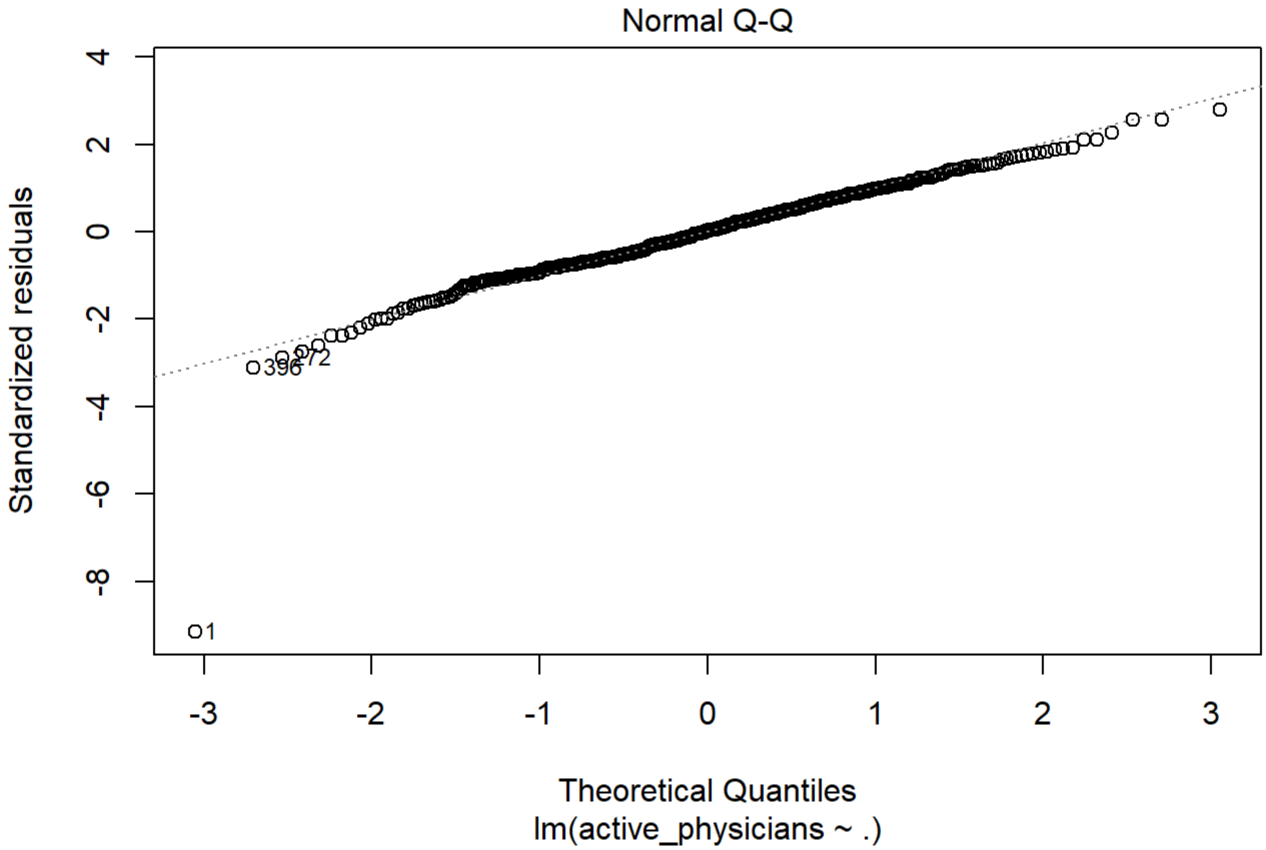

```
bptest(mlr_full_bc)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mlr_full_bc
## BP = 191.07, df = 12, p-value < 2.2e-16
```

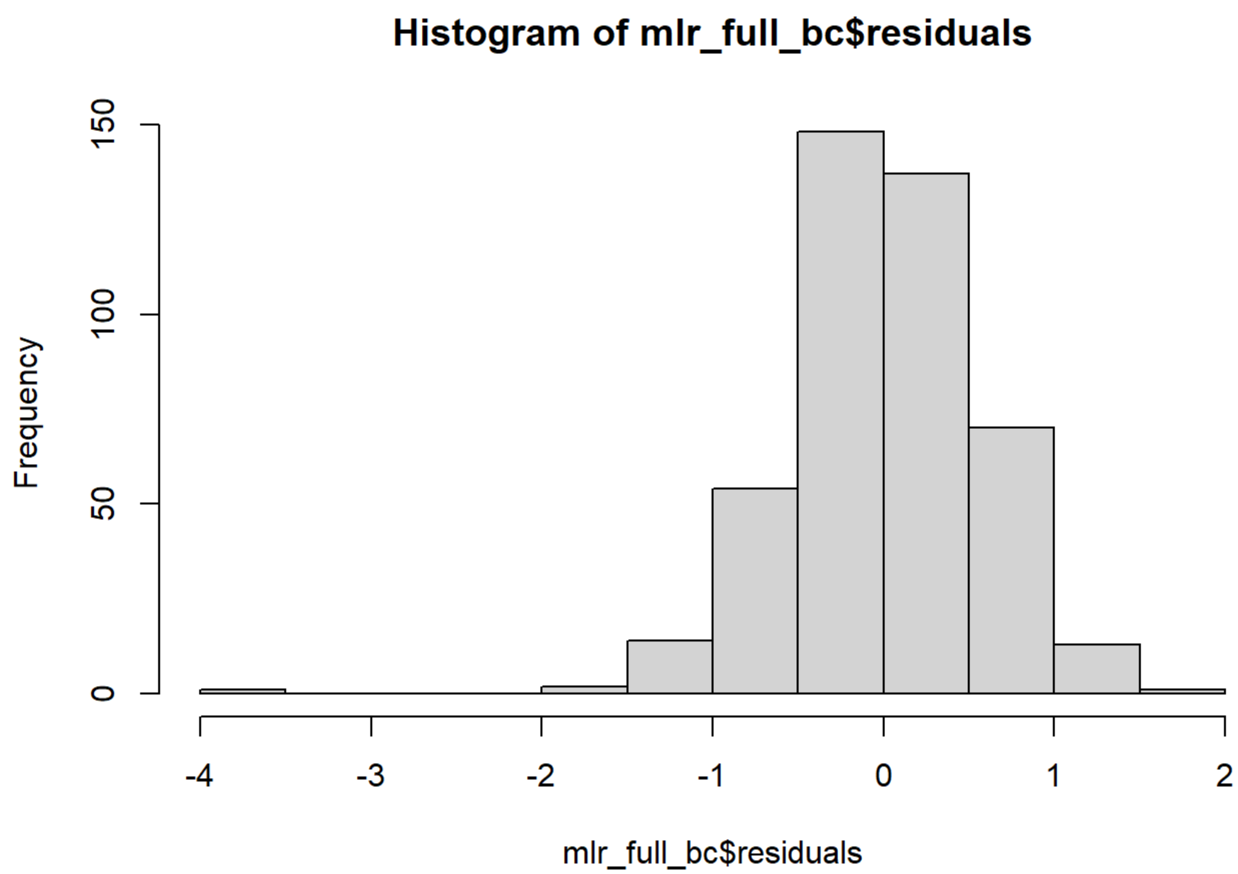
Constant variance is still NOT satisfied –> TRANSFORM

Normality

```
plot(mlr_full_bc, which = 2)
```




```
hist(mlr_full_bc$residuals)
```



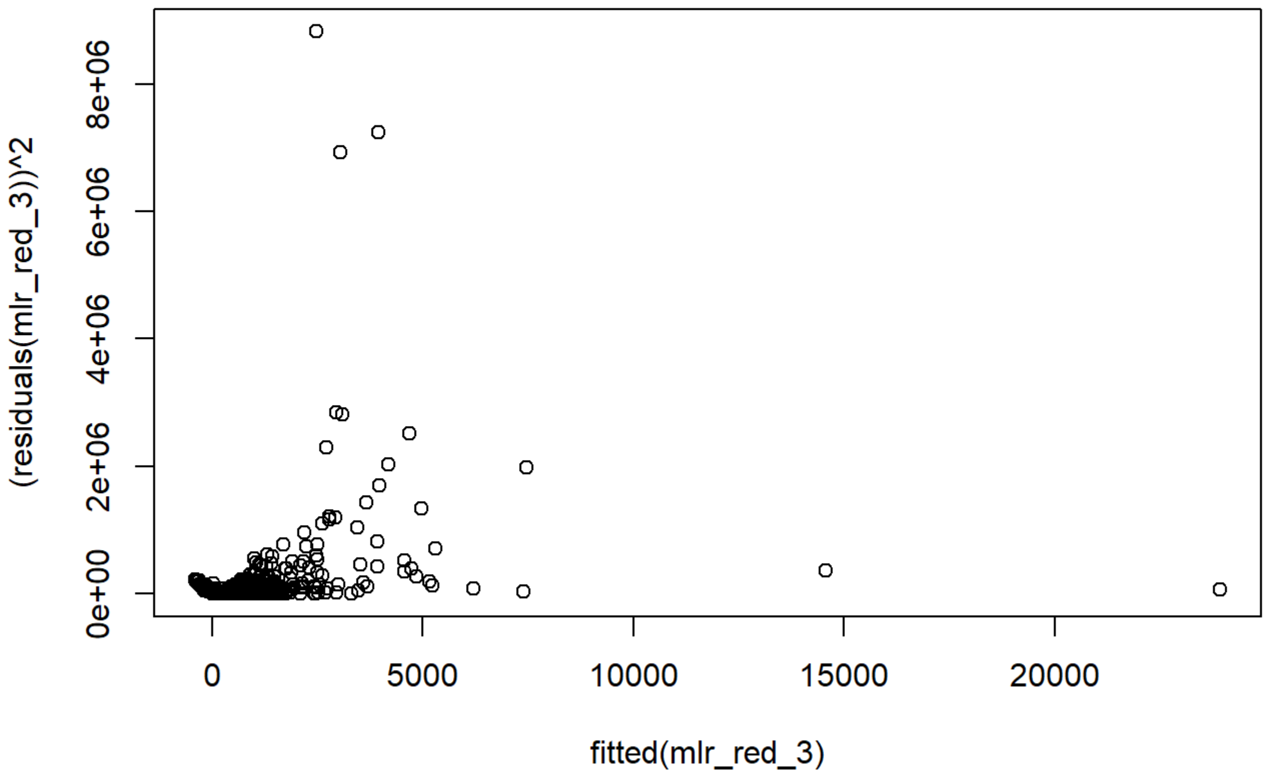

```
ks.test(mlr_full_bc$residuals, y = 'pnorm')
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  mlr_full_bc$residuals
## D = 0.16938, p-value = 2.17e-11
## alternative hypothesis: two-sided
```

Normal assumption not satisfied –> The Box-Cox transformation failed to fix the deviation from the normality assumption, so we will instead attempt to fix the Constant Variance Assumption with a variance stabilizing transformation.

Variance Stabilizing Transformation

```
plot(x=fitted(mlr_red_3),y=(residuals(mlr_red_3))^2)
```



Since our box-cox transformation was already a log(Y) transformation, and the squared residuals vs fitted values plot does not show a linear relationship, we will try to use the 1/Y transformation to stabilize variance.

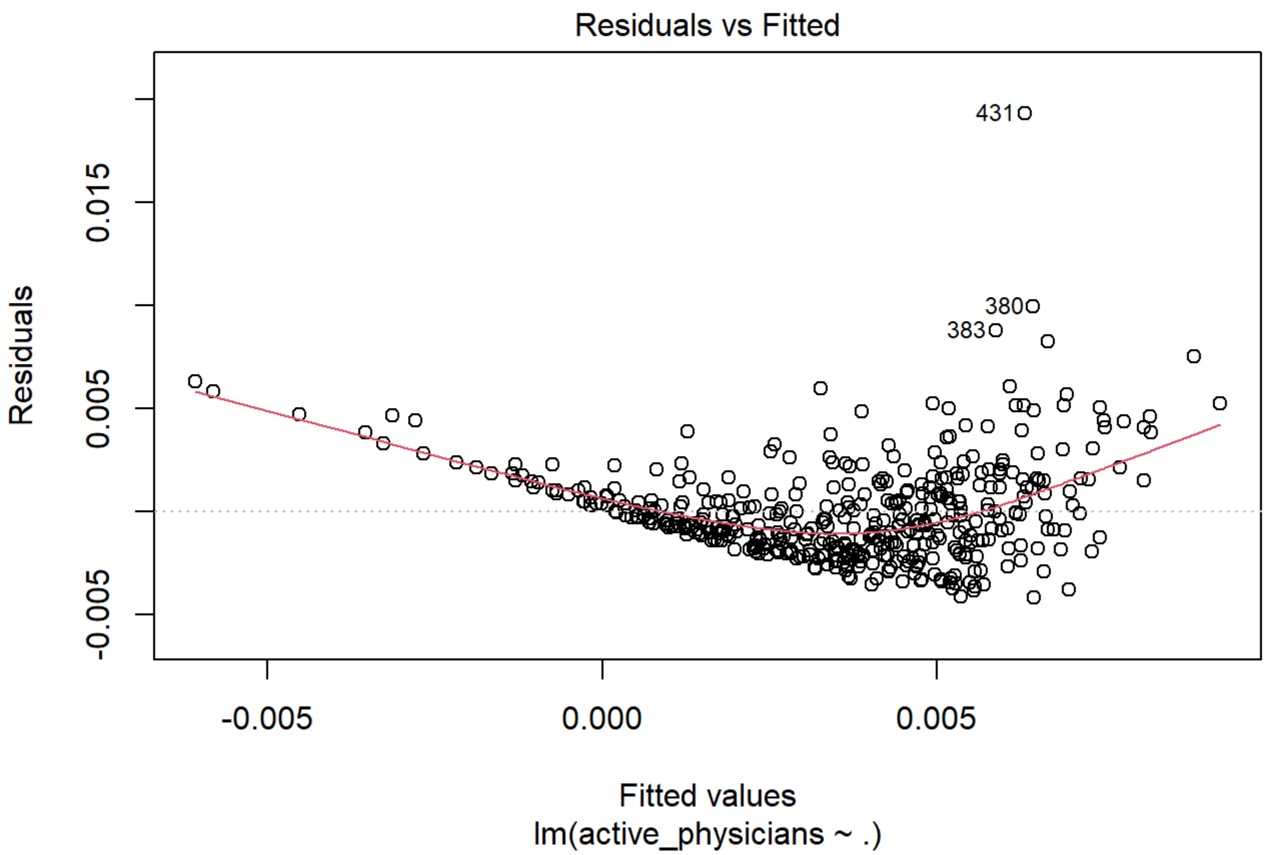
```
df_3 = df_1
df_3$active_physicians = 1/(df_3$active_physicians)

mlr_full_vs = lm(active_physicians ~ .,
                 data=df_3)
```

Re-testing Assumptions After Variance Stabilizing Transformation

Constant Variance

```
plot(mlr_full_vs, which = 1)
```



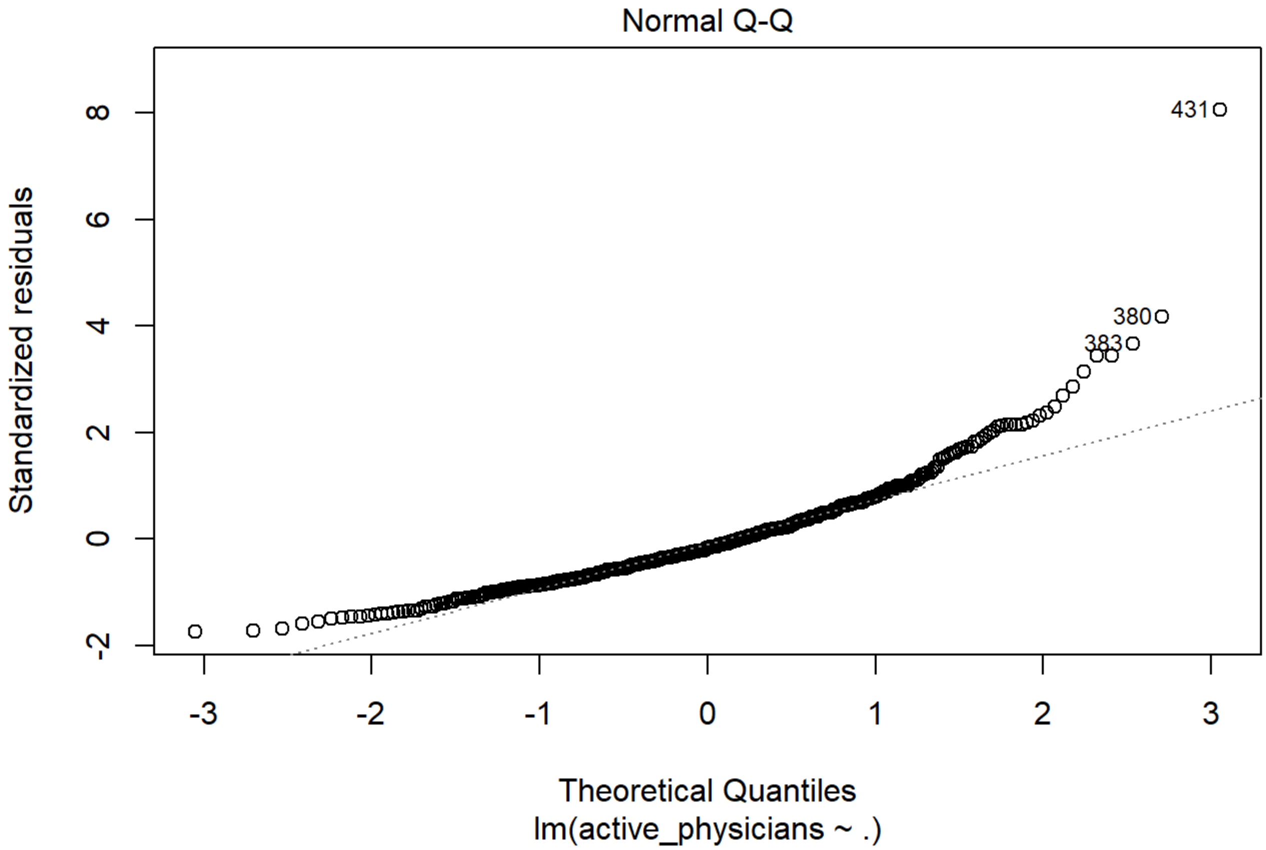

```
bptest(mlr_full_vs)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mlr_full_vs
## BP = 14.579, df = 12, p-value = 0.2653
```

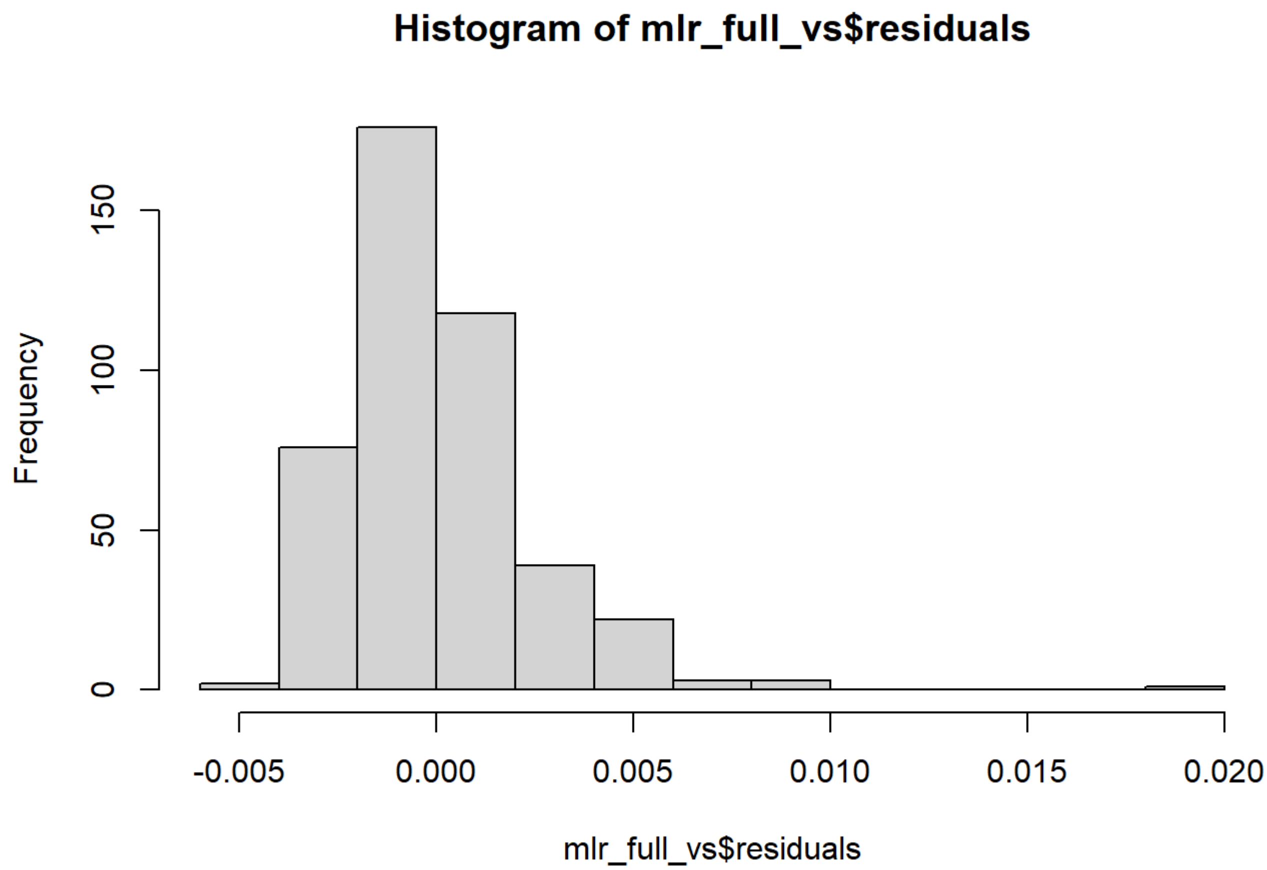
Constant variance is satisfied based on the Breusch-Pagan Test since p-val=0.265>0.05. Will note that the residual plot does not look ideal, so there may still be some issue with homoscedasticity or other assumptions in the model.

Normality

```
plot(mlr_full_vs, which = 2)
```




```
hist(mlr_full_vs$residuals)
```




```
ks.test(mlr_full_vs$residuals, y = 'pnorm')
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  mlr_full_vs$residuals
## D = 0.49833, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Normal assumption not satisfied –> Will need to note this, but since Box-Cox transformation did not work, we can’t do anything to solve this issue

Model Selection After Variance Stabilizing Transformation

```
summary(mlr_full_vs)
```

```
##
## Call:
## lm(formula = active_physicians ~ ., data = df_3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.004178 -0.001597 -0.000400  0.001110  0.019328
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.069e-02  3.621e-03   5.714 2.07e-08 ***
## land_area           -2.279e-07  8.543e-08  -2.668  0.00793 **
## pop_rate_young      -1.571e-05  4.513e-05  -0.348  0.72791
## pop_rate_old        -1.302e-04  4.096e-05  -3.180  0.00158 **
## hs_grad_rate        -6.504e-05  3.349e-05  -1.942  0.05278 .
## bachelor_deg_rate   -1.162e-04  3.811e-05  -3.050  0.00243 **
## below_poverty_rate  -5.538e-05  5.414e-05  -1.023  0.30694
## unemployment_rate   -5.083e-05  7.041e-05  -0.722  0.47073
## per_cap_income      -1.717e-07  6.492e-08  -2.644  0.00849 **
## personal_income     -4.833e-08  1.045e-08  -4.627 4.93e-06 ***
## geo_region           1.599e-04  1.457e-04   1.098  0.27294
## hospital_beds_rate  -4.319e-01  7.787e-02  -5.547 5.11e-08 ***
## serious_crimes_rate -3.115e-02  5.712e-03  -5.455 8.33e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002435 on 427 degrees of freedom
## Multiple R-squared:  0.4936, Adjusted R-squared:  0.4794
## F-statistic: 34.69 on 12 and 427 DF,  p-value: < 2.2e-16
```

Individual t-test show land_area, pop_rate_old, bachelor_deg_rate, per_cap_income, personal_income, hospital_beds_rate, and serial_crimes_rate to be statistically significant, with alpha = .05. The F-test shows p-value of ~0, which leads to the conclusion that at least one β is not equal to 0. Note that hs_grad_rate has a relatively low p-value close to 0.05.

Consider a model with only relatively low p-values

```
mlr_red_1_vs = lm(active_physicians ~ land_area + pop_rate_old +
                  hs_grad_rate + bachelor_deg_rate +
                  per_cap_income + personal_income +
                  hospital_beds_rate + serious_crimes_rate ,
                data=df_3)
summary(mlr_red_1_vs)
```

```
##
## Call:
## lm(formula = active_physicians ~ land_area + pop_rate_old + hs_grad_rate +
##     bachelor_deg_rate + per_cap_income + personal_income + hospital_beds_rate +
##     serious_crimes_rate, data = df_3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0046151 -0.0015622 -0.0004263  0.0010698  0.0193731
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.700e-02  1.892e-03   8.986  < 2e-16 ***
## land_area           -2.167e-07  7.934e-08  -2.731 0.006578 **
## pop_rate_old        -1.212e-04  3.512e-05  -3.450 0.000615 ***
## hs_grad_rate        -3.172e-05  2.556e-05  -1.241 0.215274
## bachelor_deg_rate   -1.256e-04  2.923e-05  -4.295 2.16e-05 ***
## per_cap_income      -1.494e-07  4.682e-08  -3.190 0.001527 **
## personal_income     -4.937e-08  1.034e-08  -4.775 2.47e-06 ***
## hospital_beds_rate  -4.660e-01  6.728e-02  -6.926 1.59e-11 ***
## serious_crimes_rate -3.079e-02  5.027e-03  -6.125 2.05e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002434 on 431 degrees of freedom
## Multiple R-squared:  0.4892, Adjusted R-squared:  0.4798
## F-statistic: 51.61 on 8 and 431 DF,  p-value: < 2.2e-16
```

```
n.iter = 2000;
fstats = numeric(n.iter);
for(i in 1:n.iter){
  new_df_3 = df_3;

  new_df_3[, c(2,7,8,11)] = df_3[sample(440), c(2,7,8,11)];

  model = lm(active_physicians ~ ., data = new_df_3);
  fstats[i] = summary(model)$fstat[1]
}
length(fstats[fstats > summary(mlr_full_vs)$fstat[1]])/n.iter
```

```
## [1] 0.432
```

For the permutation test, our null states the reduced model is adequate, while the alternate states it is not. With a p-value much greater than >>> a = .05 (much greater), we can say the reduced model (mlr_red_1_vs) is adequate!

Only using the predictors that had a p-value < α = .05 (all except hs_grad_rate).

```r
mlr_red_2_vs =lm(active_physicians ~ land_area + pop_rate_old
                 + bachelor_deg_rate + per_cap_income + personal_income
                 + hospital_beds_rate + serious_crimes_rate, data=df_3)
summary(mlr_red_2_vs)
```

```
##
## Call:
## lm(formula = active_physicians ~ land_area + pop_rate_old + bachelor_deg_rate +
##     per_cap_income + personal_income + hospital_beds_rate + serious_crimes_rate,
##     data = df_3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0041808 -0.0016214 -0.0004456  0.0010532  0.0191154
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.486e-02  7.785e-04  19.091  < 2e-16 ***
## land_area           -2.185e-07  7.938e-08  -2.752 0.006172 **
## pop_rate_old        -1.194e-04  3.512e-05  -3.401 0.000734 ***
## bachelor_deg_rate   -1.452e-04  2.459e-05  -5.904 7.18e-09 ***
## per_cap_income      -1.530e-07  4.676e-08  -3.271 0.001158 **
## personal_income     -4.788e-08  1.028e-08  -4.659 4.24e-06 ***
## hospital_beds_rate  -4.553e-01  6.677e-02  -6.819 3.11e-11 ***
## serious_crimes_rate -2.920e-02  4.865e-03  -6.003 4.11e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002435 on 432 degrees of freedom
## Multiple R-squared:  0.4874, Adjusted R-squared:  0.4791
## F-statistic: 58.69 on 7 and 432 DF,  p-value: < 2.2e-16
```

```r
n.iter = 2000;
fstats = numeric(n.iter);
for(i in 1:n.iter){
  new_df_3 = df_3;

  new_df_3[, c(2,5,7,8,11)] = df_3[sample(440), c(2,5,7,8,11)];

  model = lm(active_physicians ~ ., data = new_df_3);
  fstats[i] = summary(model)$fstat[1]
}
length(fstats[fstats > summary(mlr_full_vs)$fstat[1]])/n.iter
```

```
## [1] 0.3835
```

Here, the permutation test produces a p-value of 0.399 > 0.05, so the reduced model (mlr_red_2_vs) is adequate. Since we have ensured the model is adequate with permutation tests and each predictor is statistically significant (p-val<0.05) with the T-tests from the summary, mlr_red_2 is our final model.

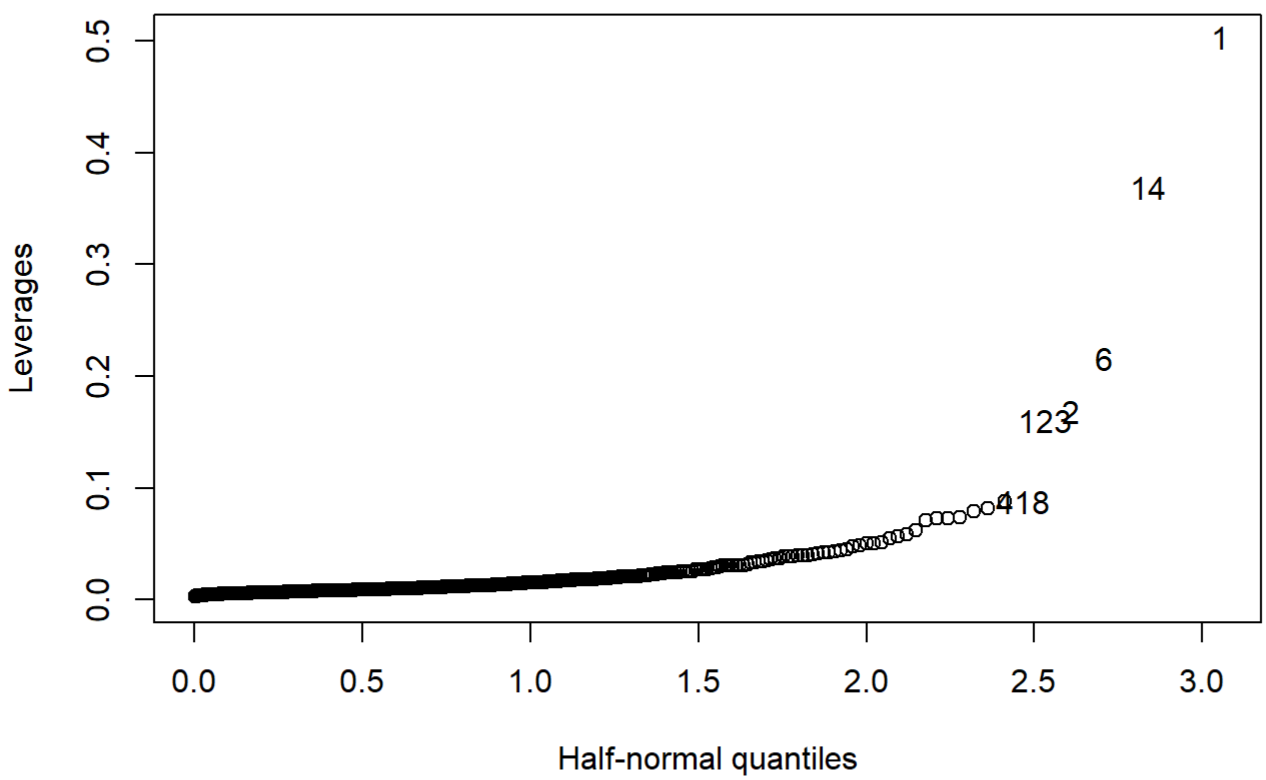
Checking Unusual Observations again after Transformation and Model Selection

High Leverage Points (HLPs)

```
cdi.leverages = lm.influence(mlr_red_2_vs)$hat
head(cdi.leverages)
```

```
##          1          2          3          4          5          6
## 0.50344452 0.16864347 0.03953810 0.03388089 0.04195758 0.21651601
```

```
halfnorm(cdi.leverages, nlab=6, labs=as.character(1:length(cdi.leverages)), ylab="Leverages")
```




```
n = dim(cdi)[1];
p = length(variable.names(mlr_red_2_vs));
(2*p/n)
```

```
## [1] 0.03636364
```

```
cdi.leverages.high = cdi.leverages[cdi.leverages > (2*p/n)]
(cdi.leverages.high = sort(abs(cdi.leverages.high), decreasing = TRUE))
```

```
##          1         14          6          2        123        418        412
## 0.50344452 0.36951821 0.21651601 0.16864347 0.16027987 0.08840355 0.08820001
##        398        171         65        173          7        436        206
## 0.08197445 0.07860391 0.07387957 0.07320644 0.07315395 0.07063183 0.06166381
##         49         85        437        392        229        363        272
## 0.05850053 0.05628863 0.05473369 0.05091430 0.05018751 0.05005902 0.04858737
##        235        396        191         34         23          5        239
## 0.04749656 0.04532947 0.04429083 0.04321225 0.04267428 0.04195758 0.04135765
##        405         42         19          3        201        293         48
## 0.04074790 0.03956324 0.03953914 0.03953810 0.03875765 0.03865217 0.03844934
##         70        215
## 0.03726306 0.03707556
```

```
length(cdi.leverages.high)
```

```
## [1] 37
```

This tells us there are 37 HLPs.

Trying to find Good vs Bad HLPs:

```
IQR_ap = IQR(df_3$active_physicians)

QT1_ap = quantile(df_3$active_physicians, .25)
QT3_ap = quantile(df_3$active_physicians, .75)

lower_lim = QT1_ap - IQR_ap
upper_lim = QT3_ap + IQR_ap

vector_lim = c(lower_lim, upper_lim)
vector_lim
```

```
##          25%          75%
## -0.003541485  0.009978723
```

```
cdi.highlev = df_3[cdi.leverages > (2*p/n), ]

cdi.highlev_lower = cdi.highlev[cdi.highlev$active_physicians < vector_lim[1], ]
cdi.highlev_upper = cdi.highlev[cdi.highlev$active_physicians > vector_lim[2], ]
cdi.highlev2 = rbind(cdi.highlev_lower, cdi.highlev_upper)
cdi.highlev2
```

```
##     land_area pop_rate_young pop_rate_old active_physicians hs_grad_rate
## 436       478           16.4         30.7        0.01020408         70.5
##     bachelor_deg_rate below_poverty_rate unemployment_rate per_cap_income
## 436               9.7                7.9               8.2          13919
##     personal_income geo_region hospital_beds_rate serious_crimes_rate
## 436            1407          3        0.002868022          0.04365327
```

```
nrow(cdi.highlev2)
```

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

1 of the 37 are Bad HLPs.

Outliers

```
cdi.resid = rstudent(mlr_red_2_vs)

bonferroni_cv = qt(.05/(2*n), n-p-1)
bonferroni_cv
```

```
## [1] -3.895342
```

```
cdi.resid.sorted = sort(abs(cdi.resid), decreasing = TRUE)[1:10]
print(cdi.resid.sorted)
```

```
##      431      380        1      383      435      271      123      417
## 8.551415 4.292989 3.637206 3.546899 3.499647 3.222480 2.890446 2.635461
##      404      291
## 2.606816 2.377980
```

```
cdi.outliers = cdi.resid.sorted[abs(cdi.resid.sorted) > abs(bonferroni_cv)]
print(cdi.outliers)
```

```
##      431      380
## 8.551415 4.292989
```

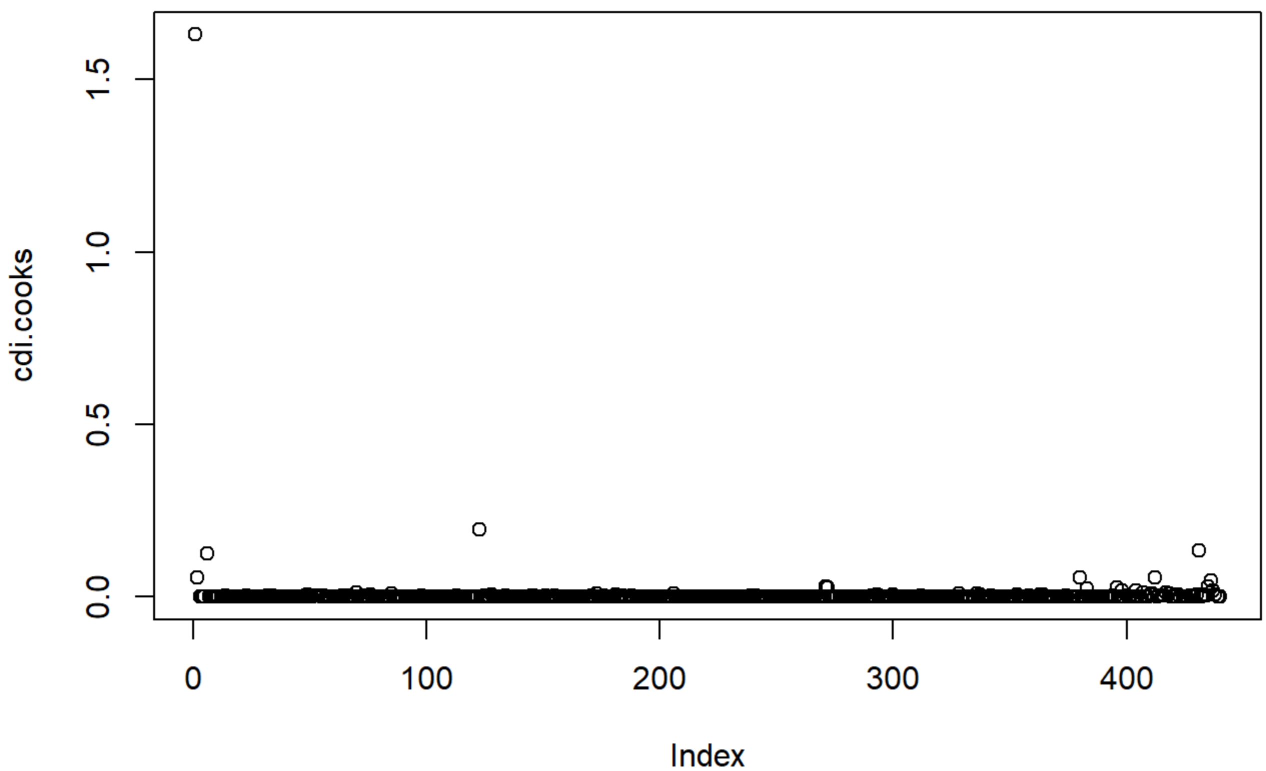
2 outliers.

Highly Influential Points (HIPs)

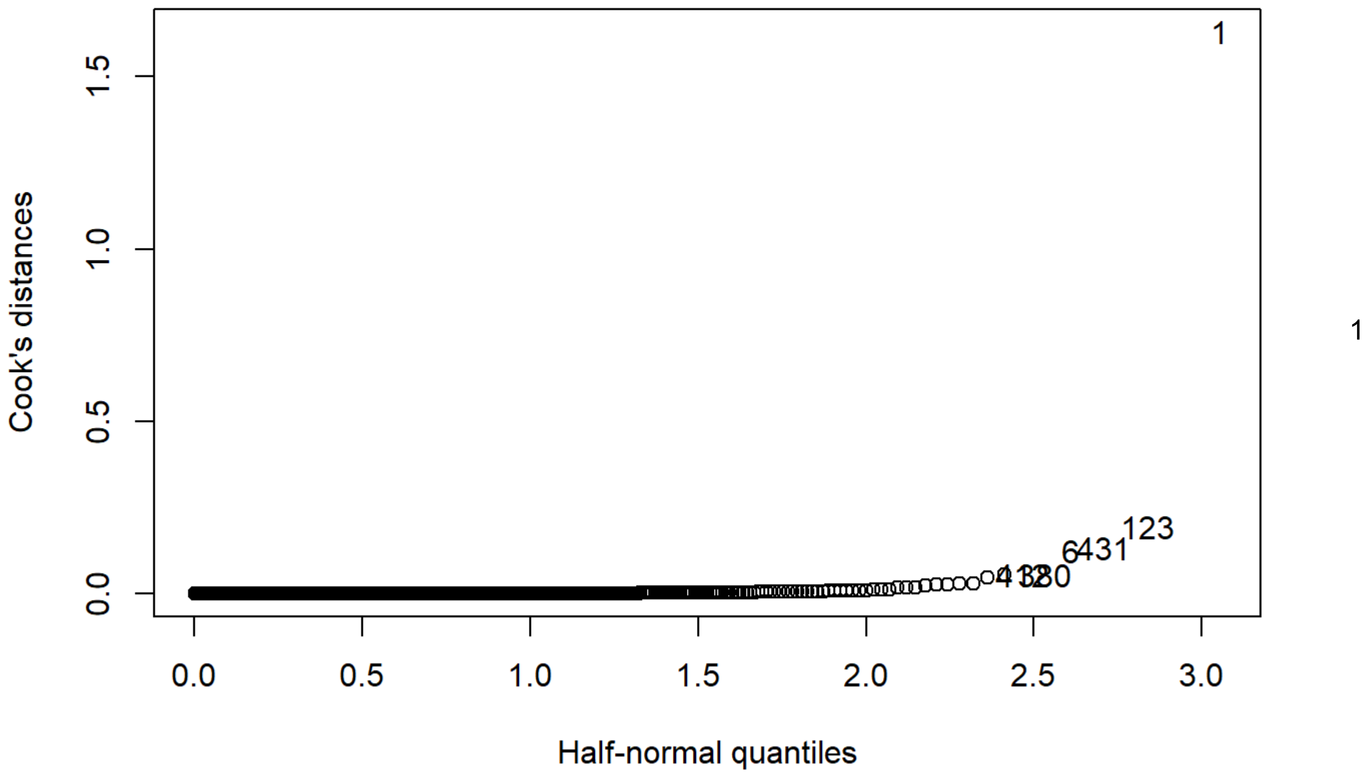
```
cdi.cooks = cooks.distance(mlr_red_2_vs)
sort(cdi.cooks, decreasing = TRUE)[1:10]
```

```
##          1        123        431          6        380        412          2
## 1.63044527 0.19599840 0.13397702 0.12400481 0.05700219 0.05596764 0.05456233
##        436        435        271
## 0.04837291 0.02939470 0.02854255
```

```
plot(cdi.cooks)
```




```
halfnorm(cdi.cooks, 6, labs=as.character(1:length(cdi.cooks)), ylab="Cook's distances")
```



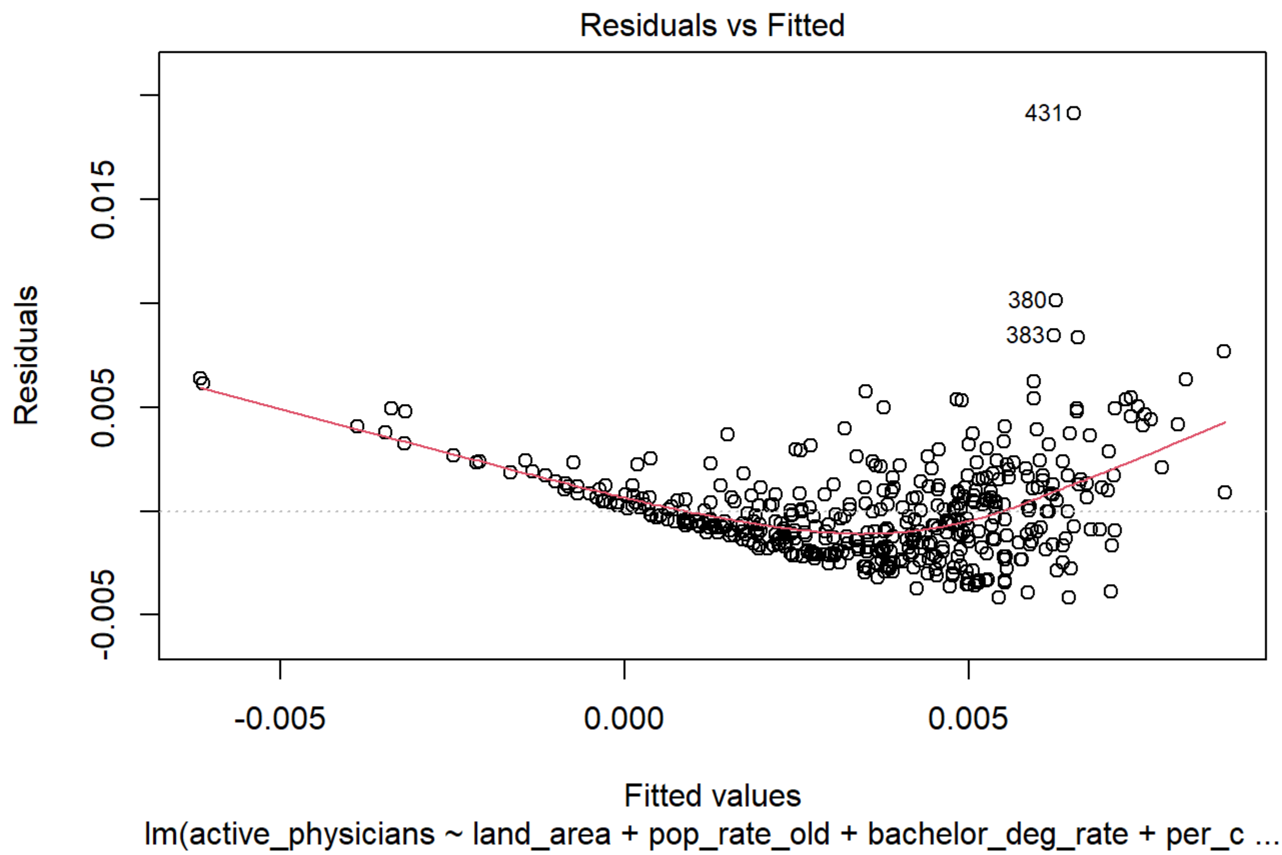
Highly Influential Point (observation 1 with CD = 1.63 > 1)

None of the observations are outliers, bad HLPs, AND HIPs, and we have no access to an industry expert. So we will NOT drop these observations.

Re-testing Assumptions After Variance Stabilizing Transformation and Model Selection

Constant Variance

```
plot(mlr_red_2_vs, which = 1)
```



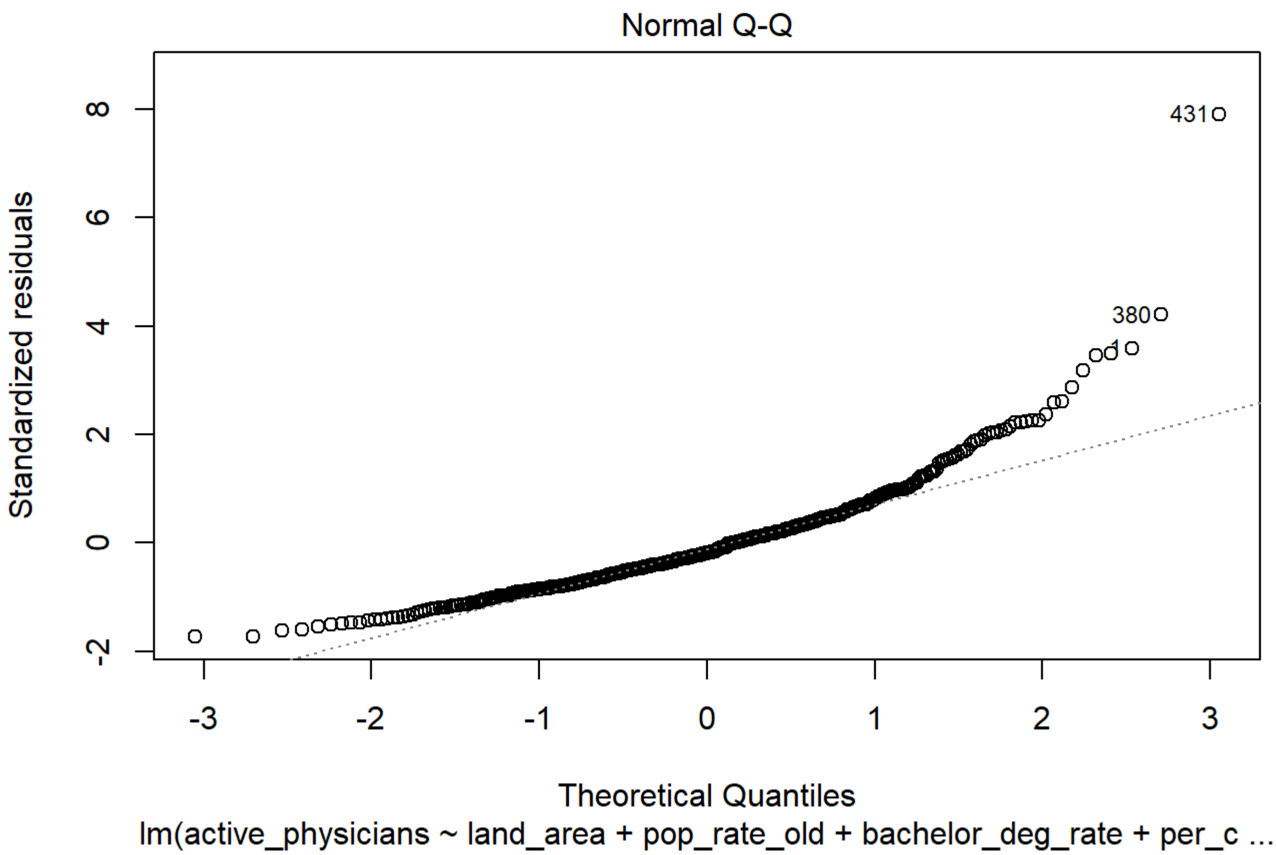

```
bptest(mlr_red_2_vs)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mlr_red_2_vs
## BP = 13.56, df = 7, p-value = 0.05958
```

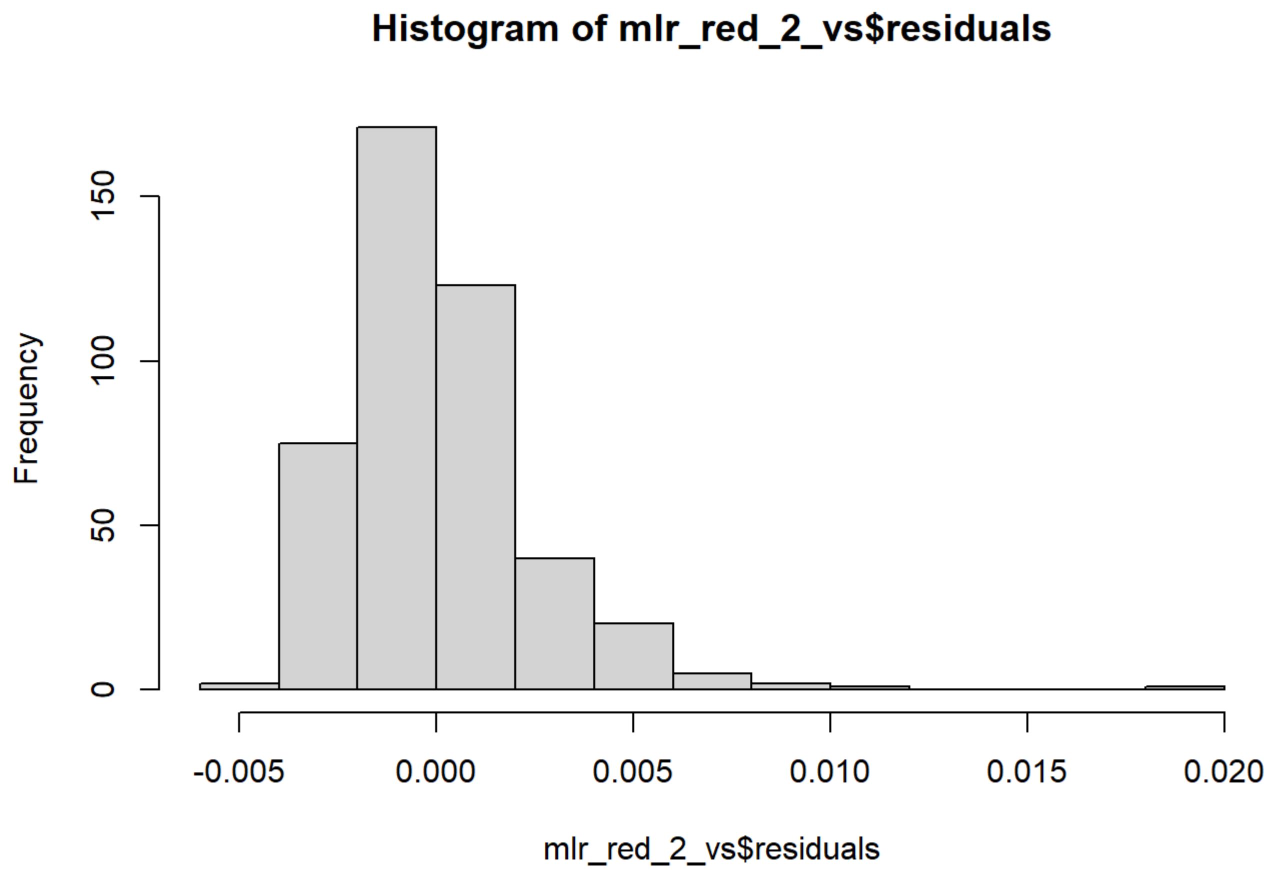
Constant variance is satisfied based on the Breusch-Pagan Test since p-val=0.06>0.05 = α Will note that the residual plot still does not look ideal, so there may still be some issue with homoscedasticity or other assumptions in the model.

Normality

```
plot(mlr_red_2_vs, which = 2)
```




```
hist(mlr_red_2_vs$residuals)
```




```
ks.test(mlr_red_2_vs$residuals, y = 'pnorm')
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  mlr_red_2_vs$residuals
## D = 0.49833, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Normal assumption still not satisfied –> Will need to note this, but since Box-Cox transformation did not work, we can't do anything to solve this issue.

Linearity Assumption

```
summary(mlr_red_2_vs)
```

```
##
## Call:
## lm(formula = active_physicians ~ land_area + pop_rate_old + bachelor_deg_rate +
##     per_cap_income + personal_income + hospital_beds_rate + serious_crimes_rate,
##     data = df_3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0041808 -0.0016214 -0.0004456  0.0010532  0.0191154
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.486e-02  7.785e-04  19.091  < 2e-16 ***
## land_area           -2.185e-07  7.938e-08  -2.752 0.006172 **
## pop_rate_old        -1.194e-04  3.512e-05  -3.401 0.000734 ***
## bachelor_deg_rate   -1.452e-04  2.459e-05  -5.904 7.18e-09 ***
## per_cap_income      -1.530e-07  4.676e-08  -3.271 0.001158 **
## personal_income     -4.788e-08  1.028e-08  -4.659 4.24e-06 ***
## hospital_beds_rate  -4.553e-01  6.677e-02  -6.819 3.11e-11 ***
## serious_crimes_rate -2.920e-02  4.865e-03  -6.003 4.11e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002435 on 432 degrees of freedom
## Multiple R-squared:  0.4874, Adjusted R-squared:  0.4791
## F-statistic: 58.69 on 7 and 432 DF,  p-value: < 2.2e-16
```

```
y.land_area = update(mlr_red_2_vs, .~. -land_area)$res
x.land_area = lm(land_area ~ pop_rate_old + bachelor_deg_rate
                 + per_cap_income + personal_income + hospital_beds_rate
                 + serious_crimes_rate, data = df_3)$res
plot(x.land_area, y.land_area, xlab="land_area Residuals",
     ylab="Active Physicians Residuals", col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

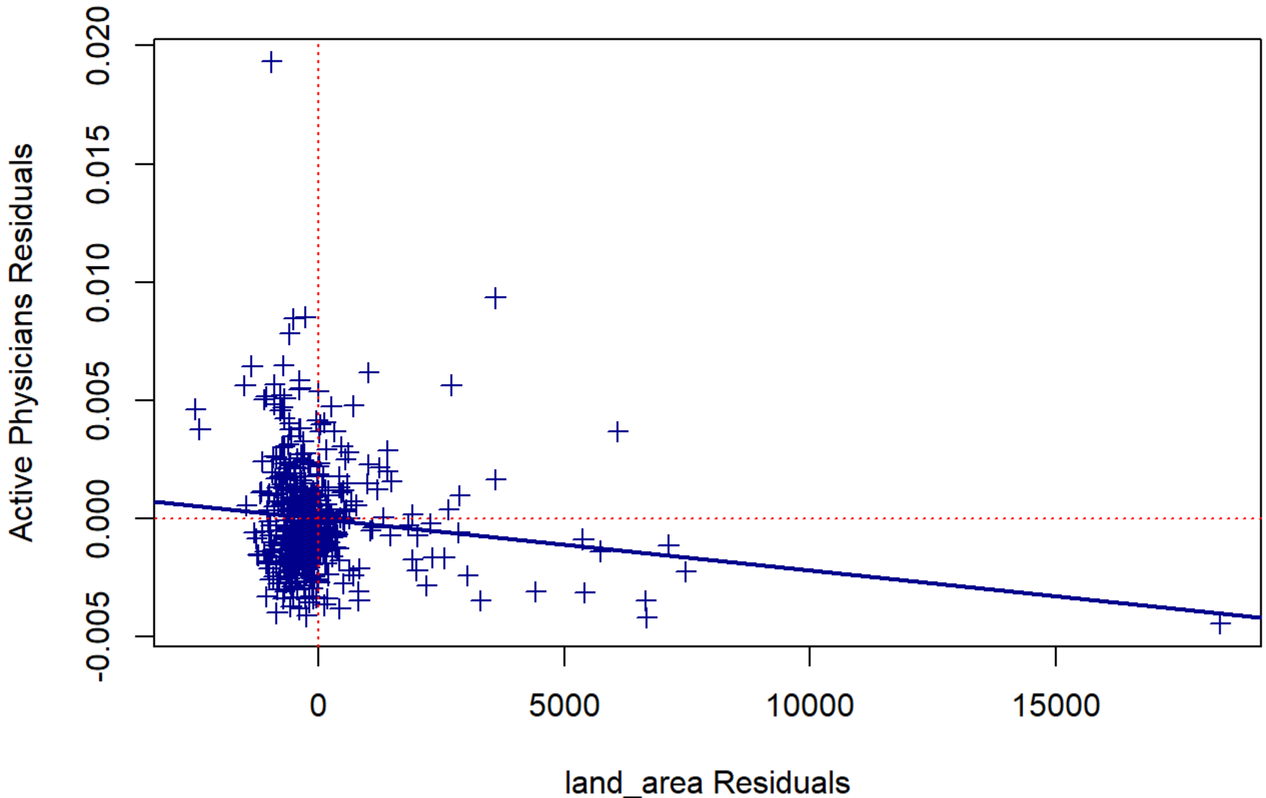
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.land_area ~ x.land_area), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```




```
y.pop_rate_old = update(mlr_red_2_vs, .~. -pop_rate_old)$res
x.pop_rate_old = lm(pop_rate_old ~ land_area + bachelor_deg_rate
                    + per_cap_income + personal_income
                    + hospital_beds_rate + serious_crimes_rate,
                    data = df_3)$res
plot(x.pop_rate_old, y.pop_rate_old, xlab="pop_rate_old Residuals",
     ylab="Active Physicians Residuals",
     col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

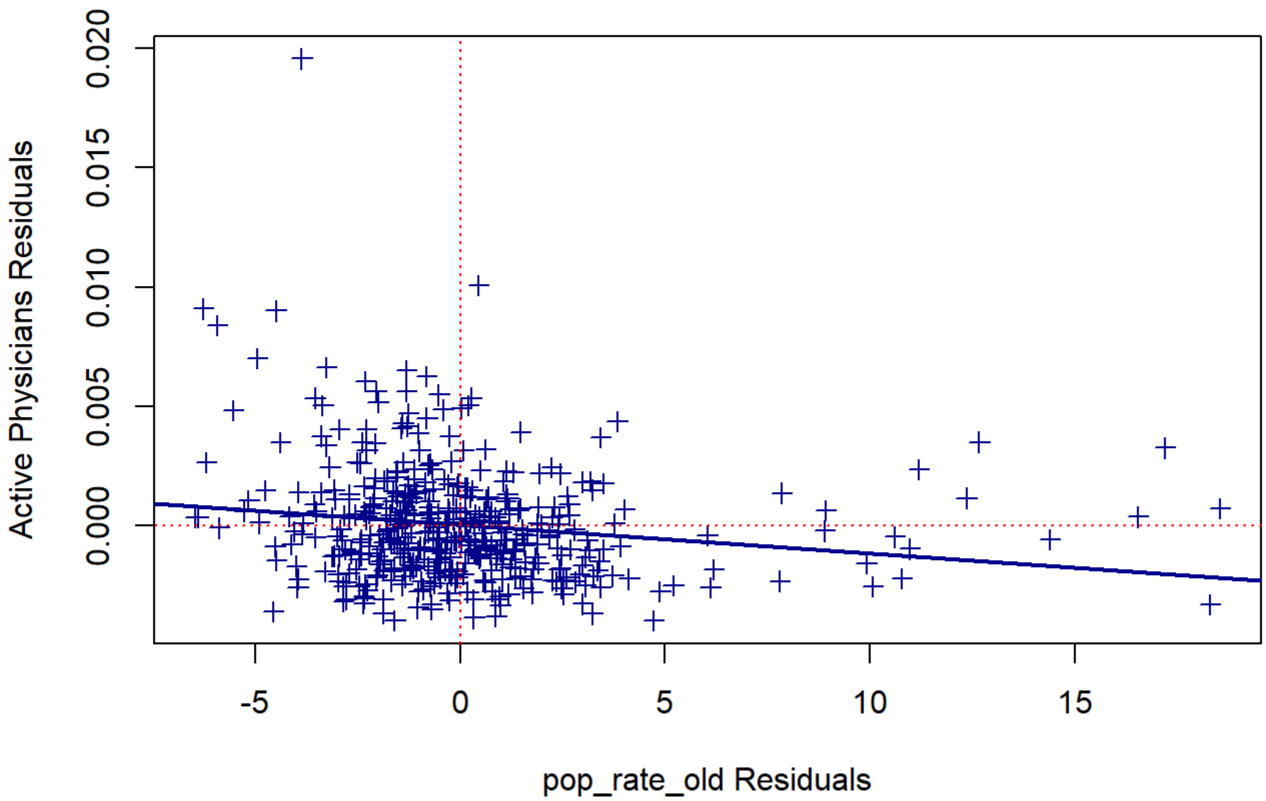
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.pop_rate_old ~ x.pop_rate_old), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```




```
y.bachelor_deg_rate = update(mlr_red_2_vs, .~. -bachelor_deg_rate)$res
x.bachelor_deg_rate = lm(bachelor_deg_rate ~ land_area + pop_rate_old
                        + per_cap_income + personal_income + hospital_beds_rate
                        + serious_crimes_rate, data = df_3)$res
plot(x.bachelor_deg_rate, y.bachelor_deg_rate,
     xlab="bachelor_deg_rate Residuals",
     ylab="Active Physicians Residuals",
     col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

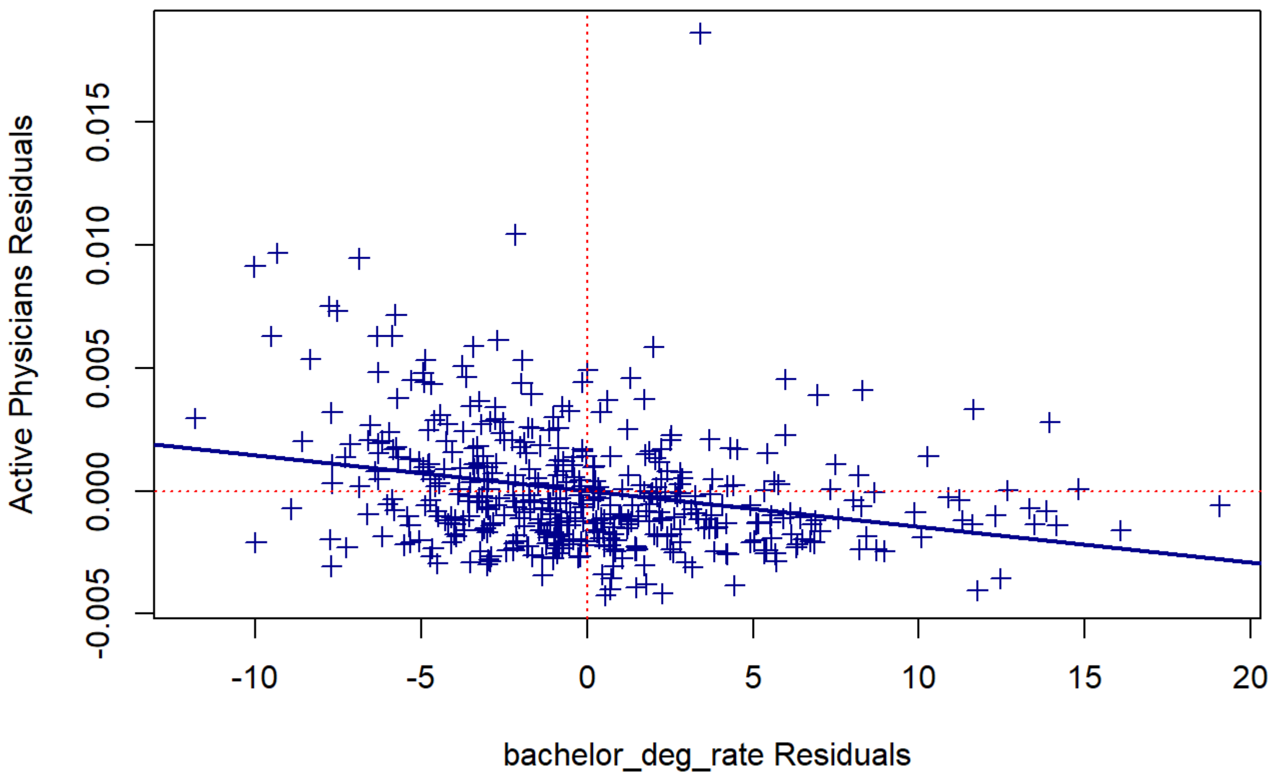
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.bachelor_deg_rate ~ x.bachelor_deg_rate), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```



```
y.per_cap_income  = update(mlr_red_2_vs, .~. -per_cap_income)$res
x.per_cap_income  = lm(per_cap_income  ~ land_area + pop_rate_old
                       + bachelor_deg_rate + personal_income
                       + hospital_beds_rate + serious_crimes_rate,
                       data = df_3)$res
plot(x.per_cap_income, y.per_cap_income, xlab="per_cap_income Residuals",
     ylab="Active Physicians Residuals",
     col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

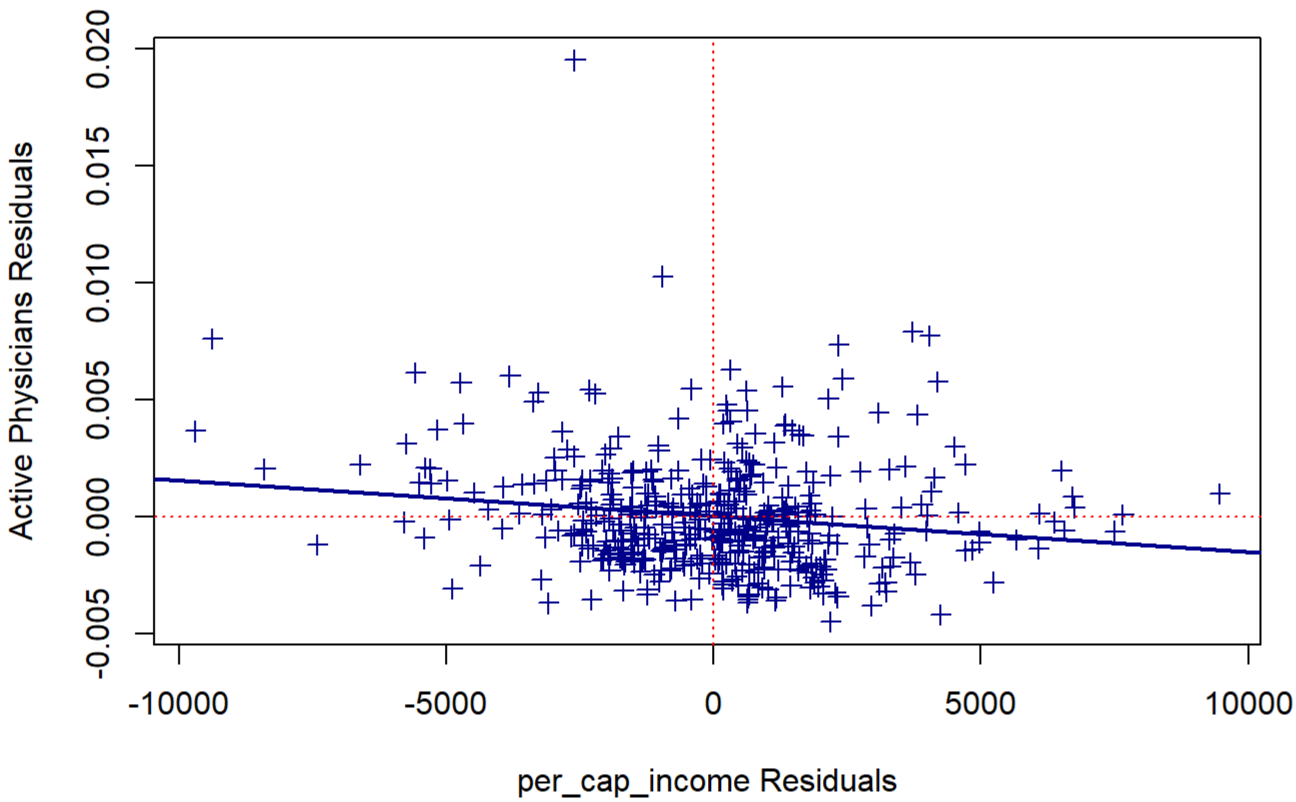
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.per_cap_income  ~ x.per_cap_income ), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```




```
y.personal_income = update(mlr_red_2_vs, .~. -personal_income)$res
x.personal_income = lm(personal_income ~ land_area + pop_rate_old
                       + bachelor_deg_rate + per_cap_income
                       + hospital_beds_rate + serious_crimes_rate,
                       data = df_3)$res
plot(x.personal_income, y.personal_income, xlab="personal_income Residuals",
     ylab="Active Physicians Residuals",
     col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

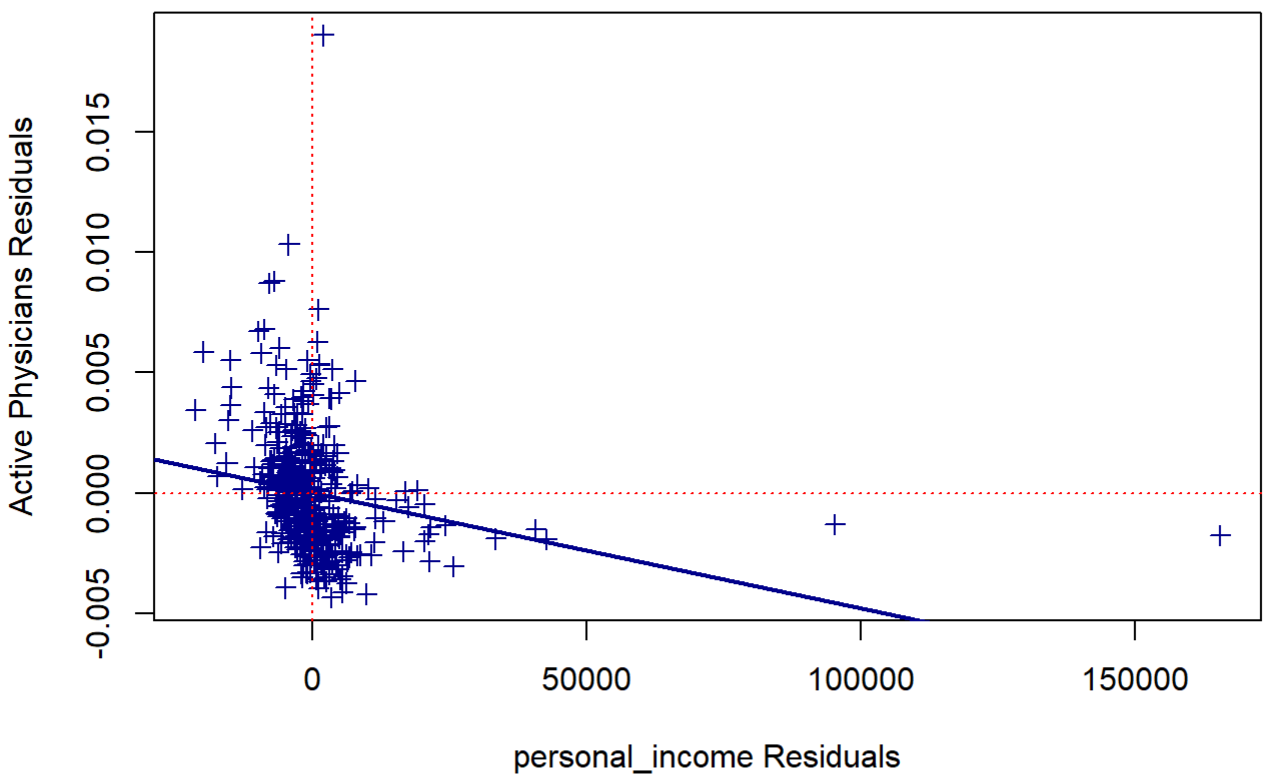
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.personal_income ~ x.personal_income), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```




```
y.hospital_beds_rate = update(mlr_red_2_vs, .~. -hospital_beds_rate)$res
x.hospital_beds_rate = lm(hospital_beds_rate ~ land_area + pop_rate_old
                          + bachelor_deg_rate + per_cap_income
                          + personal_income + serious_crimes_rate,
                          data = df_3)$res
plot(x.hospital_beds_rate, y.hospital_beds_rate,
     xlab="hospital_beds_rate Residuals", ylab="Active Physicians Residuals",
     col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

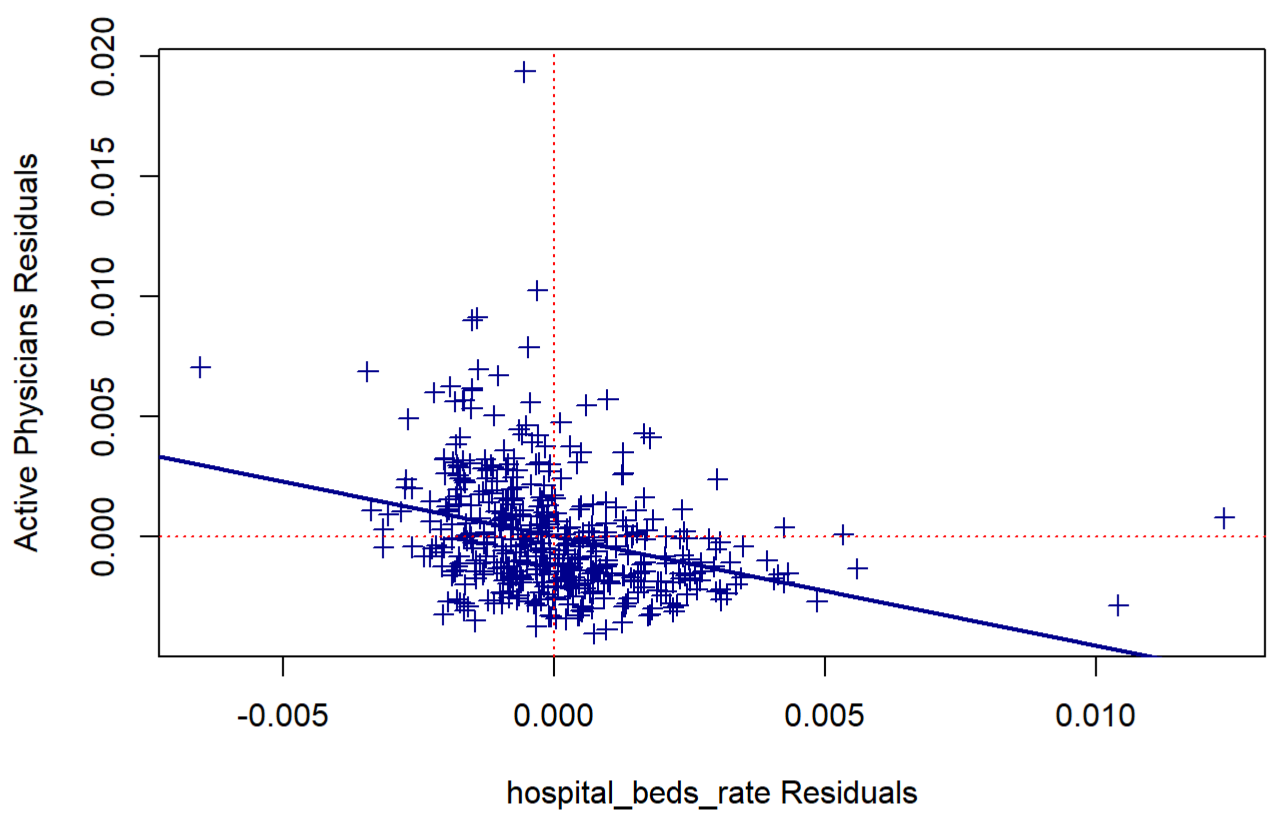
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.hospital_beds_rate ~ x.hospital_beds_rate), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```




```
y.serious_crimes_rate = update(mlr_red_2_vs, .~. -serious_crimes_rate)$res
x.serious_crimes_rate = lm(serious_crimes_rate ~ land_area + pop_rate_old
                        + bachelor_deg_rate + per_cap_income
                        + personal_income + hospital_beds_rate,
                        data = df_3)$res
plot(x.serious_crimes_rate, y.serious_crimes_rate, xlab="serious_crimes_rate Residuals", ylab="A
ctive Physicians Residuals",
     col='Darkblue', pch=3, size=3)
```

```
## Warning in plot.window(...): "size" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter
```

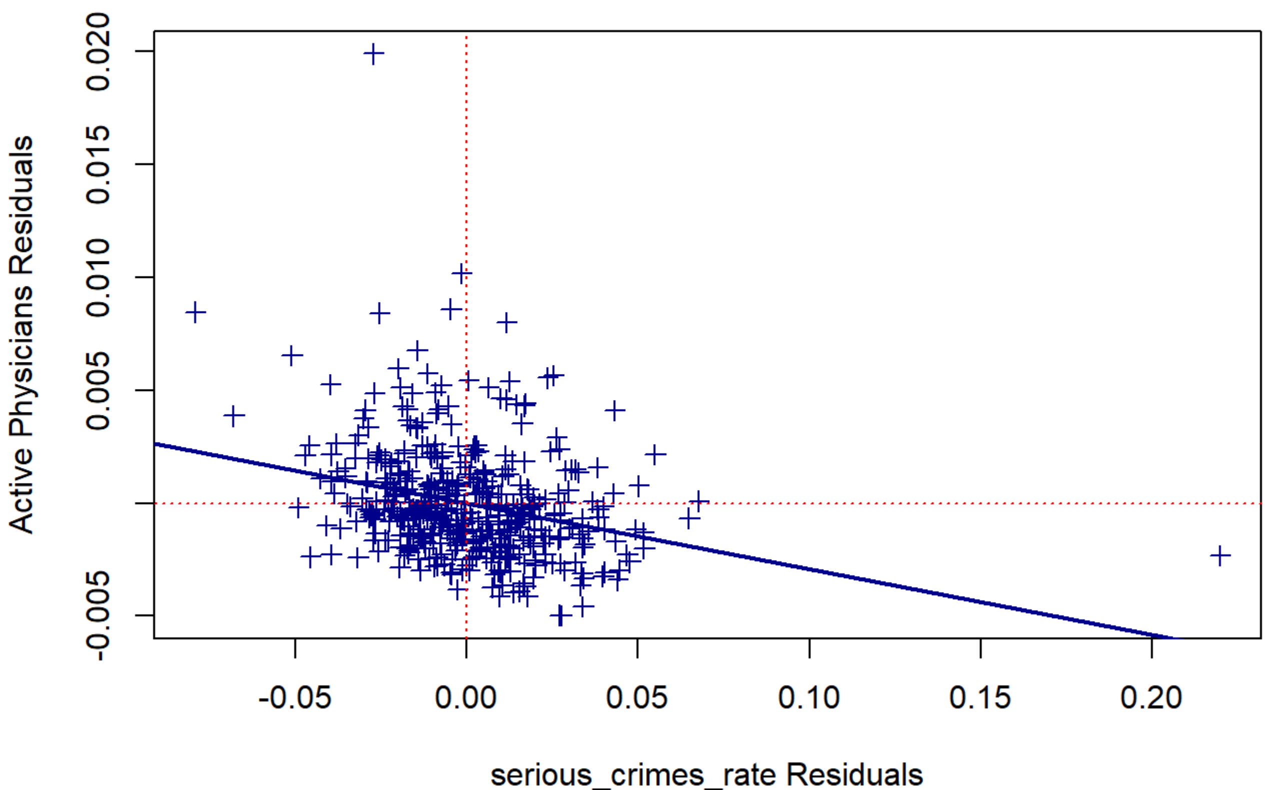
```
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a
## graphical parameter
```

```
## Warning in box(...): "size" is not a graphical parameter
```

```
## Warning in title(...): "size" is not a graphical parameter
```

```
abline(lm(y.serious_crimes_rate ~ x.serious_crimes_rate), col='Darkblue', lwd=2)
abline(v = 0, col="red", lty=3)
abline(h = 0, col="red", lty=3)
```



Since all of the plots show points approximately randomly scattered around the regression line, we can conclude that the linearity assumption is satisfied for the chosen model (mlr_red_2_vs). If we were to pick which predictors most likely have nonlinear relationship with active physicians they would be personal income and land area due to the way the points are clustered.

Collinearity

```
x = model.matrix(mlr_red_2_vs)[,-1]
dim(x)
```

```
## [1] 440   7
```

```
x = x - matrix(apply(x,2, mean), 440, 7, byrow=TRUE)
x = x / matrix(apply(x, 2, sd), 440, 7, byrow=TRUE)

eigenvalues.x = eigen(t(x) %*% x)
eigenvalues.x$val
```

```
## [1] 863.73219 640.34139 530.98491 462.98048 284.76636 204.76954  85.42513
```

```
sqrt(eigenvalues.x$val[1]/eigenvalues.x$val[7])
```

```
## [1] 3.179777
```

Since the condition number 3,18<30, we can conclude there is not significant collinearity in our chosen model.