Predicting of life expectancy

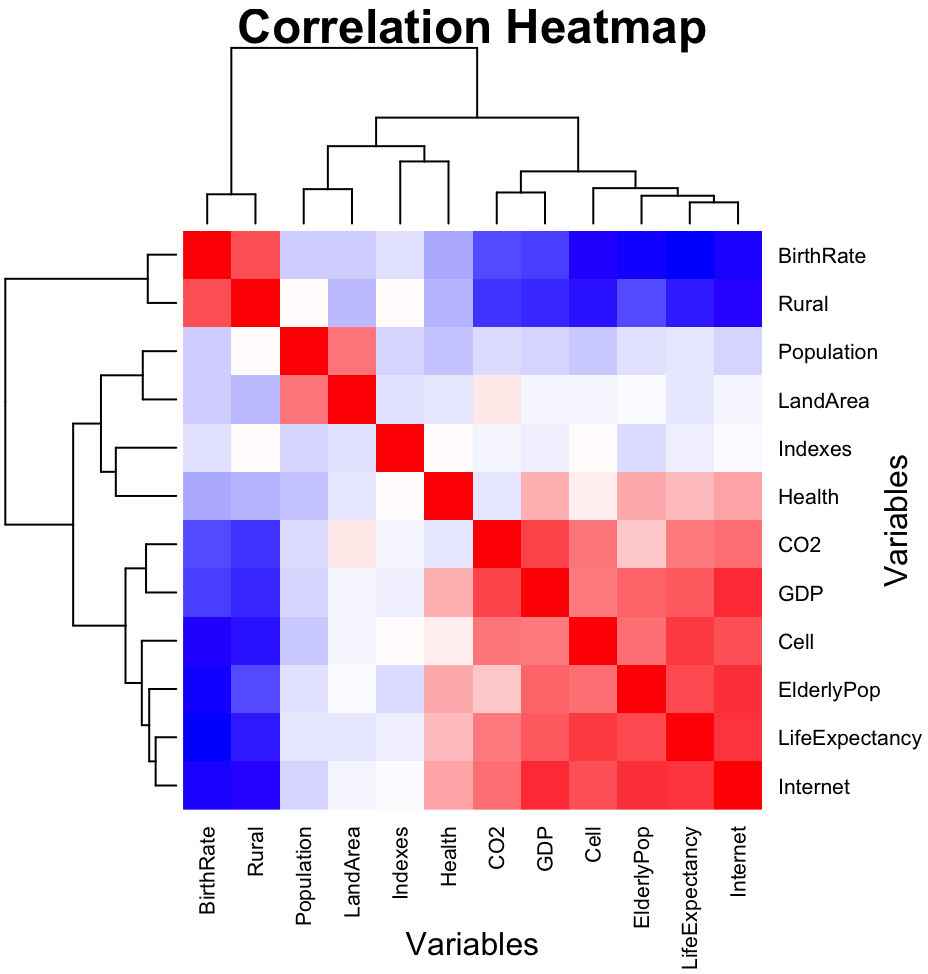
2023-12-02

```
library("car") # we will need this later on for vif
```

```
## Loading required package: carData
```

```
countries_temp = read.csv("/Users/darianlee/Downloads/countries.csv")
countries_temp_no_nans = countries_temp[complete.cases(countries_temp), ] # get rid of r
ows with NAs
set.seed(6037) # a seed I don't think anyone else is gonna choose. The 787th prime numbe
r
n = nrow(countries_temp) # to ensure its 80% of the original and not the nan version
random_indices = sample(1:n, size = 0.8 * n) # taking our random sample of indices
countries = countries_temp_no_nans[random_indices, ]
countries$Indexes = random_indices # residual plot labels points of interest by their in
dexes in the superset model, which are out of order in our random subset. Having these n
umbers in a column that I can easily search through (the 0th column is not easily iterab
le) makes finding these points and analysizing them much easier


# we will use this function when preparing our residual plots for each of the individual
predictors vs life expenctancy to ensure that our plots are zoomed in on the bulk of the
data
remove_outliers = function(data, y) {
  print(length(data) == length(y))
  q = quantile(data, c(0.25, 0.75), na.rm = TRUE)
  iqr = q[2] - q[1]
  lower_limit = q[1] - 1.5 * iqr
  upper_limit = q[2] + 1.5 * iqr
  indices_no_outliers <- which(data >= lower_limit & data <= upper_limit)
  data_no_outliers = data[indices_no_outliers]
  y_matching_indexes = y[indices_no_outliers]
  return(list(data_no_outliers = data_no_outliers, y_no_outliers = y_matching_indexes))
}

# this function lets us retire the numeric columns so that we can loop through only thes
e columns when making our graphs
get_numeric = function(numeric_cols){
for (name in colnames(countries)){
    data_temp = countries[, name]
    if (is.numeric(data_temp)) {
      numeric_cols = c(numeric_cols, name)
    }
  }

  return (numeric_cols)
}

empty_vector = c()
numerics = get_numeric(empty_vector)

# gives us all the numeric columns in countries
countries_numeric = countries[numerics]

# this will help us visualize the correlation between each predictor and LifeExpectancy
as well as between each predictor and each other
correlation_matrix = cor(countries_numeric)
```

```
heatmap(
  correlation_matrix,
  main = "Correlation Heatmap",
  xlab = "Variables",
  ylab = "Variables",
  col = colorRampPalette(c("blue", "white", "red"))(100),
  symm = TRUE,
  scale = "none",
  margins = c(5, 5),
  cexRow = 0.8,
  cexCol = 0.8
)
```




```
# we ignore the indexes column. This is just to help us later because the residual plot points out points by their indexes, and having them in a column that I can easily search through (the 0th column is not easily iteratable) makes analysis much easier
```

```
print(correlation_matrix)
```

```
##                    LandArea    Population       Rural       Health    Internet
## LandArea        1.000000000  0.4858090507 -0.12036802  0.003159548  0.03237619
## Population      0.485809051  1.0000000000  0.07615895 -0.104117390 -0.05524191
## Rural          -0.120368023  0.0761589515  1.00000000 -0.146848657 -0.65999325
## Health          0.003159548 -0.1041173905 -0.14684866  1.000000000  0.34150818
## Internet        0.032376189 -0.0552419094 -0.65999325  0.341508182  1.00000000
## BirthRate      -0.071323090 -0.0656442598  0.61920908 -0.183227720 -0.71516838
## ElderlyPop      0.060228024 -0.0128832440 -0.48657967  0.323521324  0.73648692
## LifeExpectancy  0.003863450 -0.0001930919 -0.63057850  0.261128704  0.71822225
## CO2             0.130224174 -0.0280363932 -0.55293927  0.011045571  0.50528641
## GDP             0.035119220 -0.0606622930 -0.60215668  0.310526716  0.75668082
## Cell            0.048088151 -0.0870936708 -0.64389206  0.113058507  0.61359954
## Indexes        -0.014321692 -0.0439008016  0.07780162  0.082039977  0.06089897
##                   BirthRate  ElderlyPop LifeExpectancy         CO2         GDP
## LandArea       -0.07132309  0.06022802   0.0038634504  0.13022417  0.03511922
## Population     -0.06564426 -0.01288324  -0.0001930919 -0.02803639 -0.06066229
## Rural           0.61920908 -0.48657967  -0.6305785000 -0.55293927 -0.60215668
## Health         -0.18322772  0.32352132   0.2611287035  0.01104557  0.31052672
## Internet       -0.71516838  0.73648692   0.7182222515  0.50528641  0.75668082
## BirthRate       1.00000000 -0.75748660  -0.8634066314 -0.47828244 -0.52776085
## ElderlyPop     -0.75748660  1.00000000   0.6291083634  0.22327095  0.54174782
## LifeExpectancy -0.86340663  0.62910836   1.0000000000  0.47661933  0.57816627
## CO2            -0.47828244  0.22327095   0.4766193284  1.00000000  0.64953938
## GDP            -0.52776085  0.54174782   0.5781662706  0.64953938  1.00000000
## Cell           -0.68466732  0.50428808   0.6994220628  0.49501699  0.46973287
## Indexes        -0.02015301 -0.02905853   0.0262863803  0.03542248  0.02926984
##                       Cell     Indexes
## LandArea        0.04808815 -0.01432169
## Population     -0.08709367 -0.04390080
## Rural          -0.64389206  0.07780162
## Health          0.11305851  0.08203998
## Internet        0.61359954  0.06089897
## BirthRate      -0.68466732 -0.02015301
## ElderlyPop      0.50428808 -0.02905853
## LifeExpectancy  0.69942206  0.02628638
## CO2             0.49501699  0.03542248
## GDP             0.46973287  0.02926984
## Cell            1.00000000  0.08433674
## Indexes         0.08433674  1.00000000
```

from this correlation matrix, life expectancy seems most heavily correlated with birthrate and cell. Birthrate seems to have high correlation with elderly pop and cell as well, which could be a potential issue. Elderly pop in particular has a higher correlation with birthrate than it does with life expectancy, which is concerning

```
# preparing the plots
goal_temp = countries$LifeExpectancy
countries_numeric_X = countries_numeric[, -which(names(countries_numeric) == "LifeExpect
ancy")]


get_plots = function(countries_numeric, goal_temp) {
  CIs = list()
  five_num = list()

# I am going to output them in a pdf file because I think it is useful to have large plo
ts, but I also don't want it to muddy the code. I will attach this pdf in the appendix
  pdf("single_regression.pdf", width = 16, height = 12) # putting all the graphs into a
pdf so that it doesnt muddy the code

  for (name in colnames(countries_numeric)) {
    if (!(name == "Indexes")){ # no need to make a plot of the indexes


    data_temp = countries_numeric[, name]
    five_num[[name]] = fivenum(data_temp)

    data_goal_vector = remove_outliers(data_temp, goal_temp)

    data = data_goal_vector$data_no_outliers
    goal = data_goal_vector$y_no_outliers
    par(fmrow= c(4,4))
    scatter = plot(x = data, y = goal,
                   main = paste("Scatter Plot of", name, "vs. LifeExpectancy"))


    lm_temp = lm(goal ~ data)
    ci = confint(lm_temp)
    CIs[[name]] = ci["data", ]

    resid = plot(lm_temp, which = 1,
                 main = paste("Residuals vs. Fitted for", name))


    qq = plot(lm_temp, which = 2,
              main = paste("QQ Plot for Residuals of", name))
    }
  }



# the graphs will go to the pdfs, but the CIs and 5 number summaries don't take up much
space, so we will show those in R
output = list("cis" = CIs, "fivenum" = five_num)
  return(output)
}
```

```
par(fmrow = c(1,1))
```

```
## Warning in par(fmrow = c(1, 1)): "fmrow" is not a graphical parameter
```

```
results = get_plots(countries_numeric_X, goal_temp)
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

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

```
## Warning in par(fmrow = c(4, 4)): "fmrow" is not a graphical parameter
```

```
five_num = results$fivenum
CIs = results$cis

#I'm not totally sure why this prints "TRUE", but its not hurting anything
```

```
CIs
```

```
## $LandArea
##         2.5 %        97.5 %
## -1.952345e-05 -1.015409e-05
##
## $Population
##      2.5 %     97.5 %
## -0.2098837  0.0479860
##
## $Rural
##      2.5 %     97.5 %
## -0.3416452 -0.2271395
##
## $Health
##     2.5 %    97.5 %
## 0.2147197 0.9806143
##
## $Internet
##     2.5 %    97.5 %
## 0.2360136 0.3248954
##
## $BirthRate
##      2.5 %     97.5 %
## -0.8986967 -0.7418984
##
## $ElderlyPop
##    2.5 %   97.5 %
## 1.031755 1.554411
##
## $CO2
##    2.5 %   97.5 %
## 1.193008 1.949832
##
## $GDP
##        2.5 %       97.5 %
## 0.0007007508 0.0011219928
##
## $Cell
##     2.5 %    97.5 %
## 0.1353108 0.1896191
```

```
five_num
```

```
## $LandArea
## [1]       50    27695   141655   557195 16376870
##
## $Population
## [1]    0.0640    2.7075    7.8610   27.1640 1324.6550
##
## $Rural
## [1]  0.0 29.1 43.3 64.2 89.6
##
## $Health
## [1]  0.70  7.25 10.85 13.80 26.10
##
## $Internet
## [1]  0.2  4.4 20.4 44.4 90.5
##
## $BirthRate
## [1]  8.30 12.90 20.55 30.15 53.50
##
## $ElderlyPop
## [1]  1.00  3.35  4.95 11.35 20.10
##
## $CO2
## [1]  0.02262046  0.53476368  2.28172955  7.23175904 49.05058379
##
## $GDP
## [1]    192.1238   1223.3342   4461.6757  13598.2445 105437.6707
##
## $Cell
## [1]   1.238454  59.432395  93.858131 121.968969 189.818770
```

population seems like the only one with a possible 0 slope. No numbers in the 5 number summary stand out as obvious errors. Looking at the scatter plots, internet, elderly pop, CO^2, GDP all seem to have non linear relationships in simple regression. Additionally, birthrate seems to have the best model for a single predictor. Thus we will proceed with building the model

```
library(leaps)

# for now, lets ignore the potentially important fact that Birthrate and elderly pop hav
e high correlation, and then re-assess later based on how our model looks

predictors = c("LandArea", "Rural", "Population", "Health", "Internet", "BirthRate", "El
derlyPop", "CO2", "GDP", "Cell")
temp_df = data.frame(
  LifeExpectancy = countries$LifeExpectancy,
  LandArea = countries$LandArea,
  Rural = countries$Rural,
  Population = countries$Population,
  Health = countries$Health,
  Internet = countries$Internet,
  BirthRate = countries$BirthRate,
  ElderlyPop = countries$ElderlyPop,
  CO2 = countries$CO2,
  GDP = countries$GDP,
  Cell = countries$Cell
)



max_predictors = length(predictors)
temp_model = regsubsets(LifeExpectancy ~ ., data = temp_df, nbest = 1, nvmax = max_predi
ctors)
summary_model = summary(temp_model)
summary_model
```

```
## Subset selection object
## Call: regsubsets.formula(LifeExpectancy ~ ., data = temp_df, nbest = 1,
##     nvmax = max_predictors)
## 10 Variables  (and intercept)
##            Forced in Forced out
## LandArea       FALSE      FALSE
## Rural          FALSE      FALSE
## Population     FALSE      FALSE
## Health         FALSE      FALSE
## Internet       FALSE      FALSE
## BirthRate      FALSE      FALSE
## ElderlyPop     FALSE      FALSE
## CO2            FALSE      FALSE
## GDP            FALSE      FALSE
## Cell           FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           LandArea Rural Population Health Internet BirthRate ElderlyPop CO2
## 1  ( 1 )  " "      " "   " "        " "    " "      "*"       " "        " "
## 2  ( 1 )  " "      " "   " "        " "    " "      "*"       " "        " "
## 3  ( 1 )  " "      " "   " "        " "    " "      "*"       " "        " "
## 4  ( 1 )  " "      " "   " "        " "    "*"      "*"       "*"        " "
## 5  ( 1 )  " "      " "   " "        "*"    "*"      "*"       "*"        " "
## 6  ( 1 )  " "      " "   " "        "*"    "*"      "*"       "*"        " "
## 7  ( 1 )  " "      " "   " "        "*"    "*"      "*"       "*"        "*"
## 8  ( 1 )  "*"      " "   " "        "*"    "*"      "*"       "*"        "*"
## 9  ( 1 )  "*"      "*"   " "        "*"    "*"      "*"       "*"        "*"
## 10  ( 1 ) "*"      "*"   "*"        "*"    "*"      "*"       "*"        "*"
##           GDP Cell
## 1  ( 1 )  " " " "
## 2  ( 1 )  " " "*"
## 3  ( 1 )  "*" "*"
## 4  ( 1 )  " " "*"
## 5  ( 1 )  " " "*"
## 6  ( 1 )  "*" "*"
## 7  ( 1 )  "*" "*"
## 8  ( 1 )  "*" "*"
## 9  ( 1 )  "*" "*"
## 10  ( 1 ) "*" "*"
```

```
summary_model$rsq
```

```
##  [1] 0.7454710 0.7675398 0.7819538 0.7886662 0.7972912 0.8002816 0.8034048
##  [8] 0.8053987 0.8061243 0.8062167
```

```
summary_model$cp
```

```
##  [1] 35.945737 22.343616 14.153278 11.407821  7.310140  7.196007  6.987988
##  [8]  7.578340  9.065319 11.000000
```

```
summary_model$adjr2
```

```
##  [1] 0.7437277 0.7643335 0.7774112 0.7827547 0.7901536 0.7917829 0.7935750
##  [8] 0.7941986 0.7934803 0.7920720
```

```
top_adjR2 = order(-summary_model$adjr2)[1:4]
top_Cp = order(summary_model$cp)[1:4]
top_r2 = order(-summary_model$rsq)[1:4]

top_adjR2
```

```
## [1]  8  7  9 10
```

```
top_Cp
```

```
## [1] 7 6 5 8
```

```
top_r2
```

```
## [1] 10  9  8  7
```

so far the best looking models are 7 and 8 because they score in the top 4 for all three important categories

```
summary_model$rsq[7]
```

```
## [1] 0.8034048
```

```
summary_model$cp[7]
```

```
## [1] 6.987988
```

```
summary_model$adjr2[7]
```

```
## [1] 0.793575
```

```
cat("\n\n")
```

```
summary_model$rsq[8]
```

```
## [1] 0.8053987
```

```
summary_model$cp[8]
```

```
## [1] 7.57834
```

```
summary_model$adjr2[8]
```

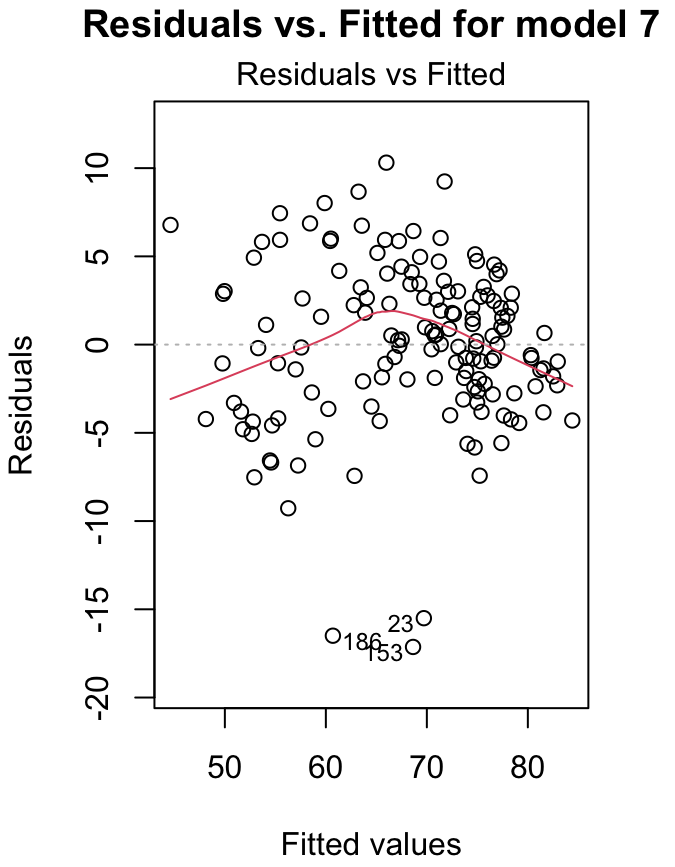
```
## [1] 0.7941986
```

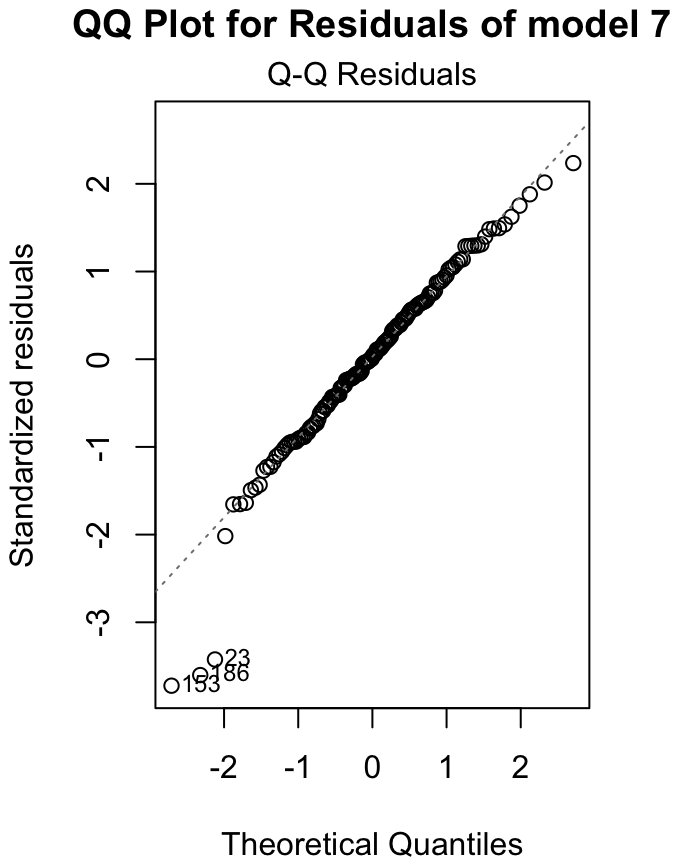
```
#because the R^2 and adjusted R^2 are very similar, but the Mallows' Cp for model 7 is v
ery very close to the number of predictors
```

```
model7 = lm(LifeExpectancy ~ Health + Internet + BirthRate + ElderlyPop + CO2 + GDP + Ce
ll, data = countries_numeric)

model7
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ Health + Internet + BirthRate +
##     ElderlyPop + CO2 + GDP + Cell, data = countries_numeric)
##
## Coefficients:
## (Intercept)       Health     Internet    BirthRate   ElderlyPop          CO2
##   8.110e+01    1.944e-01    5.458e-02   -7.368e-01   -4.694e-01   -1.304e-01
##         GDP         Cell
##   8.076e-05    3.861e-02
```

```
par(mfrow = c(1, 2))
resid1 = plot(model7, which = 1,
                main = paste("Residuals vs. Fitted for model 7"))


qq1 = plot(model7, which = 2,
              main = paste("QQ Plot for Residuals of model 7"))
```






```
residuals_1 = resid(model7)
Q1 = quantile(residuals_1, 0.25)
Q3 = quantile(residuals_1, 0.75)
IQR = Q3 - Q1

threshold = 1.5
outliers = residuals_1 < (Q1 - threshold * IQR) | residuals_1 > (Q3 + threshold * IQR)

print(residuals_1[outliers])
```

```
##        23       153       186
## -15.50285 -17.13335 -16.49638
```

```
summary(model7)
```

```
##
## Call:
## lm(formula = LifeExpectancy ~ Health + Internet + BirthRate +
##     ElderlyPop + CO2 + GDP + Cell, data = countries_numeric)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.1333  -2.6660   0.1091   2.9109  10.3103
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.110e+01  3.044e+00  26.646  < 2e-16 ***
## Health       1.944e-01  9.675e-02   2.010  0.04640 *
## Internet     5.458e-02  3.009e-02   1.813  0.07190 .
## BirthRate   -7.368e-01  6.979e-02 -10.557  < 2e-16 ***
## ElderlyPop  -4.694e-01  1.472e-01  -3.189  0.00176 **
## CO2         -1.304e-01  8.742e-02  -1.491  0.13812
## GDP          8.076e-05  3.967e-05   2.036  0.04364 *
## Cell         3.861e-02  1.268e-02   3.045  0.00278 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.656 on 140 degrees of freedom
## Multiple R-squared:  0.8034, Adjusted R-squared:  0.7936
## F-statistic: 81.73 on 7 and 140 DF,  p-value: < 2.2e-16
```

```
#there were 3 big outliers on the residual graph. I will analysize these points in case they are measurement errors
print(residuals_1[outliers])
```

```
##        23       153       186
## -15.50285 -17.13335 -16.49638
```

lets inspect these variables

```
# this is where having our indexes column pays off



  pdf("scatter_plots_with_highlights.pdf", width = 16, height = 12) # putting all the graphs into a pdf so that it doesnt muddy the code


highlight_indices = c(23, 153, 186)
only_problem_rows = countries[which(countries_numeric$Indexes %in% highlight_indices), ]
only_problem_rows
```

```
##          Country Code LandArea Population Rural Health Internet BirthRate
## 23      Botswana  BOT   566730      1.921  40.4   16.6      6.2      24.5
## 153 South Africa  RSA  1214470     48.793  39.3   10.4      8.6      22.0
## 186     Zimbabwe  ZIM   386850     12.463  62.7    3.2     11.4      29.9
##     ElderlyPop LifeExpectancy       CO2       GDP      Cell Indexes
## 23         3.7           54.2 2.4761538 7402.9310 117.76162      23
## 153        4.4           51.5 8.9332027 7275.3440 100.76153     153
## 186        4.0           44.2 0.7288916  594.5215  59.65897     186
```

```
goal = countries_numeric$LifeExpectancy
for (name in colnames(countries_numeric)) {
  if (!(name == "Indexes")) {
    data = countries_numeric[, name]

    scatter = plot(x = data, y = goal,
                   main = paste("Scatter Plot of", name, "vs. LifeExpectancy"))

    problem_x = only_problem_rows[, name]
    problem_y = only_problem_rows$LifeExpectancy
    points(x = problem_x, y = problem_y, col = "red", pch = 20)
  }
}

# looking at the graph, it looks like these points may be outliers in have a low life ex
pectancy to birthrate ratio, which according to our heat map is one of the strongest cor
related columns


# lets see if they really are outliers
```

```
# judging by the scatter plots, these are countries where the birth rate is low, but the
life expectancy is also low. Lets see if there is anything statistically weird about the
se points by comparing their ratios to the other countries. I also included CO2 and cell
because the points also look lower than normal on the scatter plots for these variables

countries$ratioCell = countries$Cell/countries$LifeExpectancy
countries$ratioBirth = countries$BirthRate/countries$LifeExpectancy
countries$ratioCO2 = countries$CO2/countries$LifeExpectancy

# forward orders
head(countries[order(countries$ratioCell), ], 5)
```

```
##             Country Code LandArea Population Rural Health Internet BirthRate
## 118          Myanmar  MYA   653520     49.563  67.4    0.7      0.2      20.5
## 89  Korea, Dem. Rep.  PRK   120410     23.819  37.3    4.0      2.0      13.7
## 54           Eritrea  ERI   101000      4.927  79.3    3.0      4.1      37.0
## 151  Solomon Islands  SOL    27990      0.511  82.0   14.4      2.0      30.4
## 56          Ethiopia  ETH  1000000     80.713  83.0   11.5      0.4      38.2
##     ElderlyPop LifeExpectancy        CO2       GDP     Cell Indexes  ratioCell
## 118        5.5           61.6 0.27038609 1325.9524 1.238454     118 0.02010478
## 89         9.4           67.2 3.24836505 1723.0542 1.774069      89 0.02639984
## 54         2.4           59.5 0.08375326  402.9575 3.526578      54 0.05927022
## 151        3.1           66.3 0.38810241 1261.0388 5.574675     151 0.08408257
## 56         3.1           55.2 0.08945206  358.2541 7.856936      56 0.14233579
##     ratioBirth    ratioCO2
## 118  0.3327922 0.004389385
## 89   0.2038690 0.048338766
## 54   0.6218487 0.001407618
## 151  0.4585219 0.005853732
## 56   0.6920290 0.001620508
```

```
head(countries[order(countries$ratioBirth), ], 5)
```

```
##                   Country Code LandArea Population Rural Health Internet
## 64                Germany  GER   348630     82.110  26.4   18.0     78.1
## 9                 Austria  AUT    82450      8.337  32.8   15.8     72.9
## 90            Korea, Rep.  KOR    96920     48.607  18.5   12.3     80.2
## 83                  Italy  ITA   294140     59.832  31.9   13.6     44.4
## 22 Bosnia and Herzegovina  BIH    51200      3.773  52.6   14.0     34.7
##    BirthRate ElderlyPop LifeExpectancy       CO2       GDP      Cell Indexes
## 64       8.3       20.0           79.7  9.580545 40152.219 127.97677      64
## 9        9.3       17.0           80.2  8.123597 45209.396 145.99132       9
## 90       9.4       10.4           79.8 10.475246 31362.259 103.87159      90
## 83       9.6       20.1           81.2  7.439455 33916.877 135.57412      83
## 22       9.1       13.8           75.1  8.286827  4408.838  80.14842      22
##    ratioCell ratioBirth  ratioCO2
## 64  1.605731  0.1041405 0.1202076
## 9   1.820341  0.1159601 0.1012917
## 90  1.301649  0.1177945 0.1312687
## 83  1.669632  0.1182266 0.0916189
## 22  1.067223  0.1211718 0.1103439
```

```
head(countries[order(countries$ratioCO2), ], 5)
```

```
##             Country Code LandArea Population Rural Health Internet BirthRate
## 28          Burundi  BDI    25680      8.074  89.6   11.8      0.8      34.5
## 1        Afghanistan  AFG   652230     29.021  76.0    3.7      1.7      46.5
## 107            Mali  MLI  1220190     12.706  67.8   11.1      1.6      42.6
## 39  Congo, Dem. Rep.  COD  2267050     64.257  66.0   17.5      0.5      44.9
## 34             Chad  CHA  1259200     10.914  73.3   13.8      1.2      45.7
##     ElderlyPop LifeExpectancy        CO2      GDP     Cell Indexes ratioCell
## 28         2.8           50.4 0.02262046 192.1238 13.72445      28 0.2723105
## 1          2.2           43.9 0.02503483 501.4709 37.80711       1 0.8612098
## 107        2.3           48.4 0.04108260 601.9196 47.66381     107 0.9847895
## 39         2.6           47.6 0.04507820 199.2718 17.21302      39 0.3616181
## 34         2.9           48.7 0.04646669 675.8290 23.28566      34 0.4781451
##     ratioBirth     ratioCO2
## 28   0.6845238 0.0004488186
## 1    1.0592255 0.0005702695
## 107  0.8801653 0.0008488141
## 39   0.9432773 0.0009470210
## 34   0.9383984 0.0009541414
```

```
# reverse orders

head(countries[order(countries$ratioCell, decreasing = TRUE), ], 5)
```

```
##               Country Code LandArea Population Rural Health Internet BirthRate
## 143       Saudi Arabia  KSA  2000000     24.807  17.6    8.4     31.3      23.4
## 115         Montenegro  MNE    13450      0.622  39.8   13.6     41.0      12.1
## 139 Russian Federation  RUS 16376870    141.950  27.2    9.2     32.0      12.1
## 130             Panama  PAN    74340      3.399  26.8   13.5     27.6      20.6
## 183            Vietnam  VIE   310070     86.211  72.2    9.3     24.2      17.2
##     ElderlyPop LifeExpectancy       CO2       GDP     Cell Indexes ratioCell
## 143        2.9           73.1 16.569065 15835.936 187.8615     143  2.569924
## 115       12.8           74.1  3.100589  6510.105 185.2761     115  2.500352
## 139       13.3           67.8 12.037008 10439.642 167.6820     139  2.473186
## 130        6.4           75.7  2.029156  7588.894 184.7167     130  2.440115
## 183        6.3           74.4  1.496485  1224.191 177.1409     183  2.380926
##     ratioBirth   ratioCO2
## 143  0.3201094 0.22666300
## 115  0.1632928 0.04184330
## 139  0.1784661 0.17753699
## 130  0.2721268 0.02680524
## 183  0.2311828 0.02011405
```

```
head(countries[order(countries$ratioBirth, decreasing = TRUE), ], 5)
```

```
##             Country Code LandArea Population Rural Health Internet BirthRate
## 1       Afghanistan  AFG   652230     29.021  76.0    3.7      1.7      46.5
## 125           Niger  NIG  1266700     14.704  83.5   14.8      0.5      53.5
## 185          Zambia  ZAM   743390     12.620  64.6   15.3      5.5      42.9
## 39  Congo, Dem. Rep.  COD  2267050     64.257  66.0   17.5      0.5      44.9
## 34             Chad  CHA  1259200     10.914  73.3   13.8      1.2      45.7
##     ElderlyPop LifeExpectancy        CO2       GDP     Cell Indexes ratioCell
## 1          2.2           43.9 0.02503483  501.4709 37.80711       1 0.8612098
## 125        2.0           51.4 0.05887499  357.7122 24.53329     125 0.4773013
## 185        3.0           45.4 0.15254961 1252.6957 38.26972     185 0.8429453
## 39         2.6           47.6 0.04507820  199.2718 17.21302      39 0.3616181
## 34         2.9           48.7 0.04646669  675.8290 23.28566      34 0.4781451
##     ratioBirth     ratioCO2
## 1    1.0592255 0.0005702695
## 125  1.0408560 0.0011454278
## 185  0.9449339 0.0033601236
## 39   0.9432773 0.0009470210
## 34   0.9383984 0.0009541414
```

```
head(countries[order(countries$ratioCO2, decreasing = TRUE), ], 5)
```

```
##                 Country Code LandArea Population Rural Health Internet
## 137               Qatar  QAT    11590      1.281   4.4    6.8     25.5
## 91               Kuwait  KUW    17820      2.728   1.6    6.1     36.7
## 25     Brunei Darussalam  BRU     5270      0.392  25.2    7.0     68.0
## 176 United Arab Emirates  UAE    83600      4.485  22.1    8.9     72.0
## 7                 Aruba  ARU      180      0.105  53.2    9.5     22.8
##     BirthRate ElderlyPop LifeExpectancy      CO2       GDP     Cell Indexes
## 137      12.1        1.1           75.9 49.05058 68793.782 132.4349     137
## 91       17.7        2.1           78.0 30.11476 33994.412 160.7757      91
## 25       19.8        3.3           77.4 27.53860  8672.831 109.0705      25
## 176      14.0        1.0           77.7 24.98403 39624.702 145.4535     176
## 7        11.7        9.2           74.7 21.68383 20984.560 122.6183       7
##     ratioCell ratioBirth  ratioCO2
## 137  1.744860  0.1594203 0.6462528
## 91   2.061227  0.2269231 0.3860867
## 25   1.409179  0.2558140 0.3557959
## 176  1.871989  0.1801802 0.3215447
## 7    1.641477  0.1566265 0.2902789
```

none of these points are showing up as clear outliers in any category. Removing these points is not justified

Lets see if there are any outliers in the highly predictive predictors that may be influencing our results

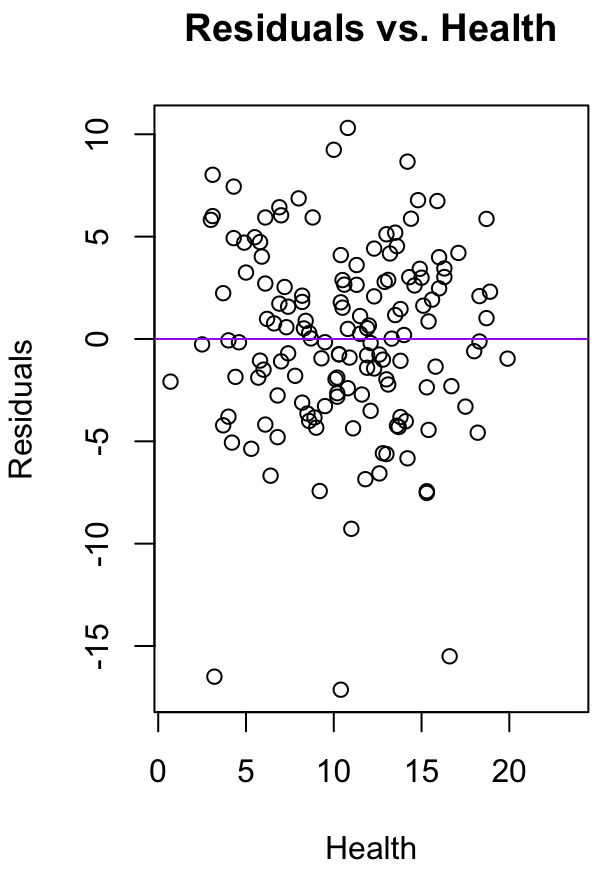
```
# I am going to modify the function from earlier because I don't want it to change the s
ize of the columns

important_x = countries$BirthRate
q = quantile(important_x, c(0.25, 0.75), na.rm = TRUE)
iqr = q[2] - q[1]
lower_limit = q[1] - 1.5 * iqr
upper_limit = q[2] + 1.5 * iqr

new_x = c()
for (x in important_x){
  if (x < lower_limit | x > upper_limit){
    print("outlier found!")
    new_x = c(new_x, NA)
  }
  else{
    new_x = c(new_x, x)
  }
}
```

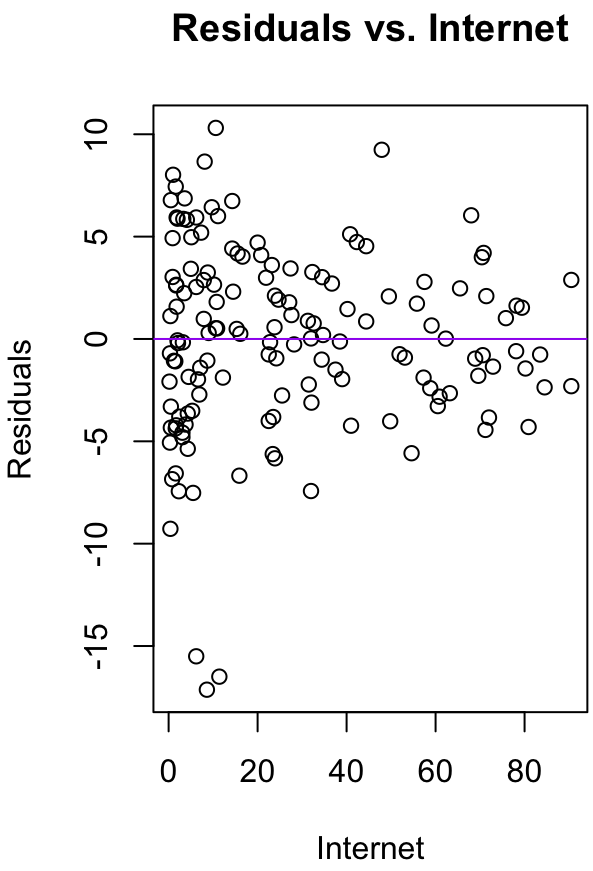
```
important_x = countries$ElderlyPop
q = quantile(important_x, c(0.25, 0.75), na.rm = TRUE)
iqr = q[2] - q[1]
lower_limit = q[1] - 1.5 * iqr
upper_limit = q[2] + 1.5 * iqr

new_x = c()
for (x in important_x){
  if (x < lower_limit | x > upper_limit){
    print("outlier found!")
    new_x = c(new_x, NA)
  }
  else{
    new_x = c(new_x, x)
  }
}
```

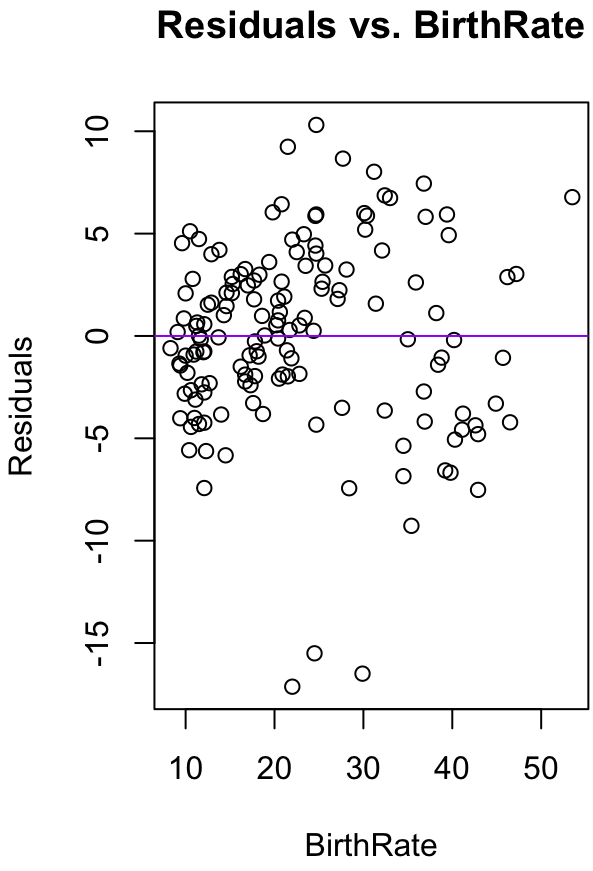
No outliers were found

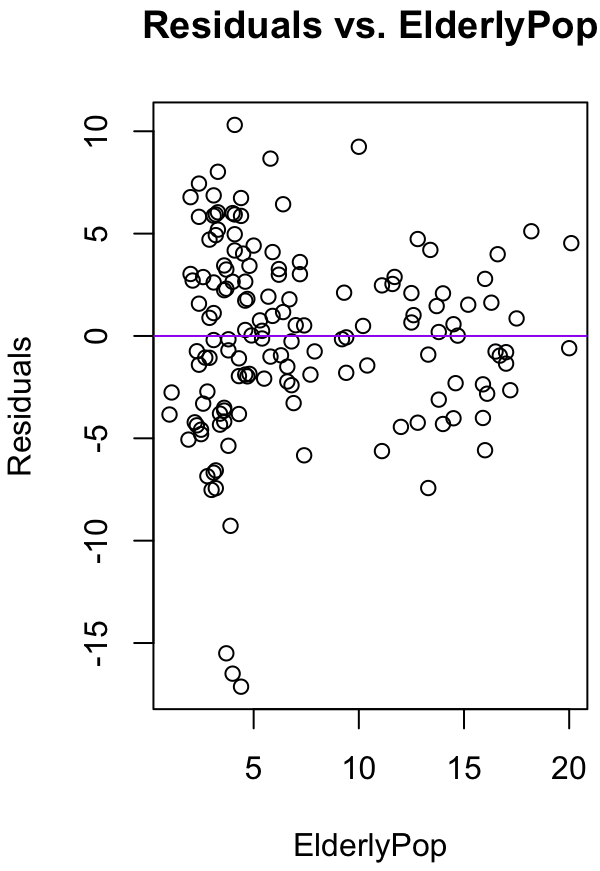
Lets plot the residuals against each predictor, zooming in on the area where the bulk of the data is

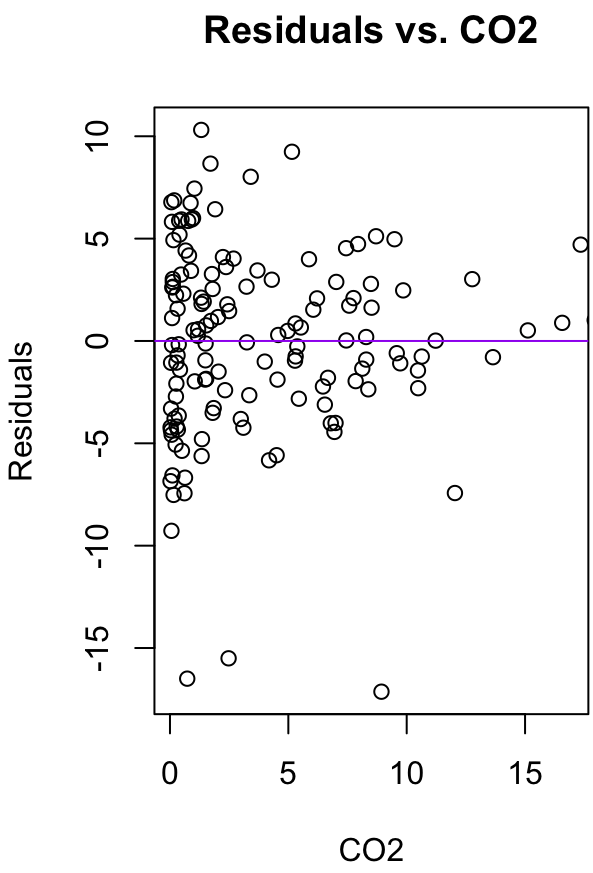
```
par(mfrow = c(1, 2))
predictors = c("Health", "Internet", "BirthRate", "ElderlyPop", "CO2", "GDP", "Cell")

for (predictor in predictors) {
  data = countries[[predictor]]
  q = quantile(data, c(0.25, 0.75), na.rm = TRUE)
  iqr = q[2] - q[1]
  lower_limit = q[1] - 1.5 * iqr
  upper_limit = q[2] + 1.5 * iqr

  x_limit_min = max(c(min(data), lower_limit))
  x_limit_max = min(c(max(data), upper_limit))


  plot(countries[[predictor]], residuals(model7), main = paste("Residuals vs.", predicto
r),
       xlab = predictor, ylab = "Residuals", xlim = c(x_limit_min, x_limit_max))
    abline(h = 0, col = "purple")
    identify(countries[[predictor]], countries$LifeExpectancy, labels = rownames(countri
es))



}
```

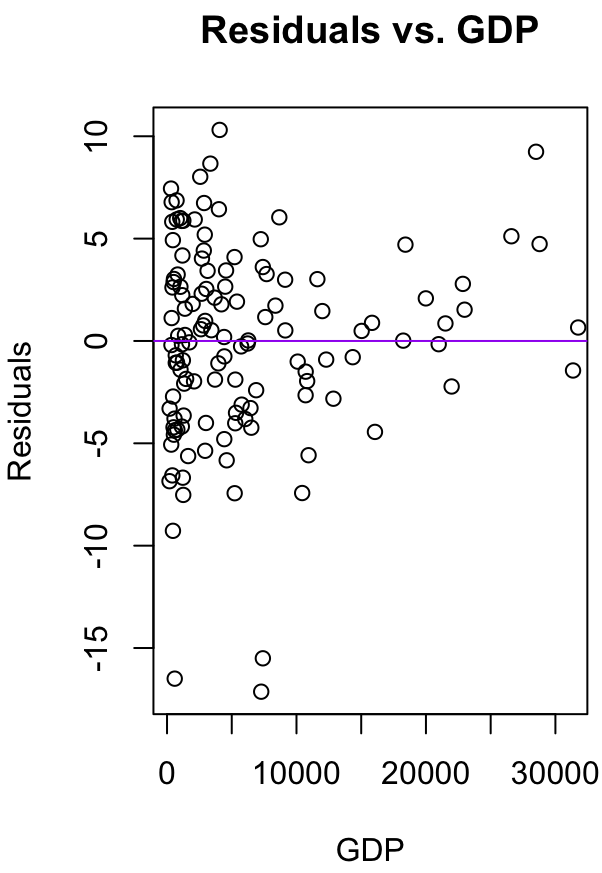

Residuals vs. Health
Residuals
Health
10
5
0
-5
-10
-15
0
5
10
15
20


Residuals vs. Internet
Residuals
Internet
10
5
0
-5
-10
-15
0
20
40
60
80

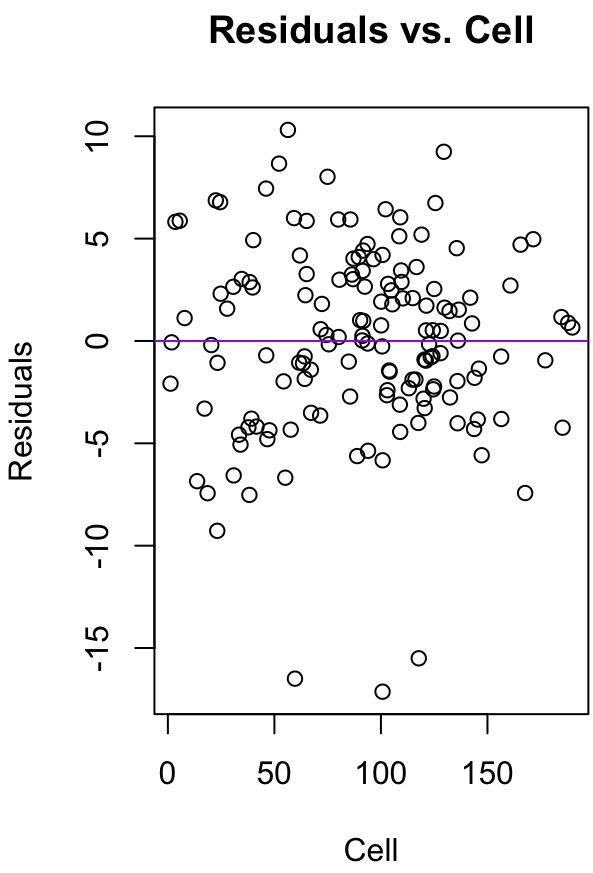

Residuals vs. BirthRate
Residuals
BirthRate
10
5
0
-5
-10
-15
10
20
30
40
50


Residuals vs. ElderlyPop
Residuals
ElderlyPop
10
5
0
-5
-10
-15
5
10
15
20






```
par(mfrow = c(1, 1))
```



There are no clear non linear relationships in these X variables, expect perhaps in GDP, where there is sligfht U trend in the residuals. We will try transforming this variable to see if it will improve our overall model.

```
countries$GDP_log = log(countries$GDP)
```

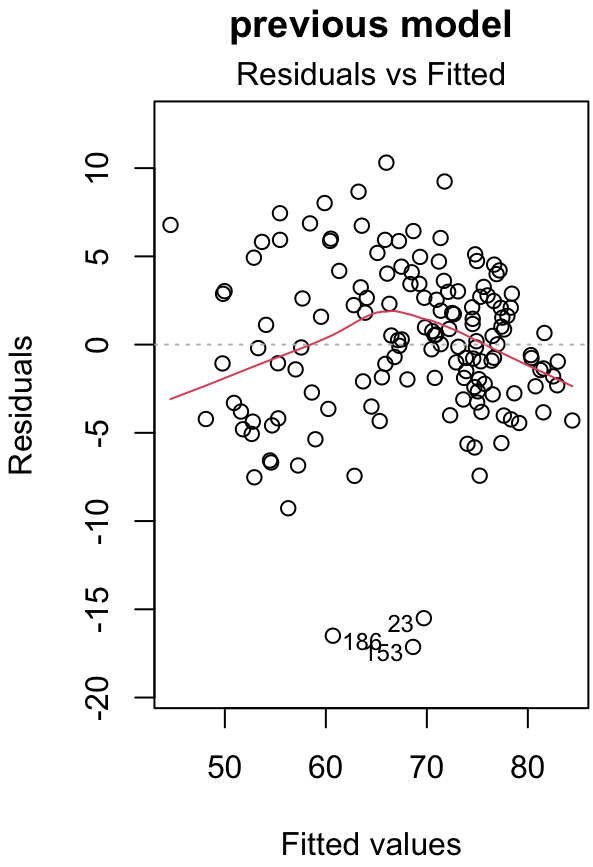
```
model7_log = lm(LifeExpectancy ~ Health + Internet + BirthRate + ElderlyPop + CO2 + GDP_
log + Cell, data = countries)



print("new summary (log)")
```

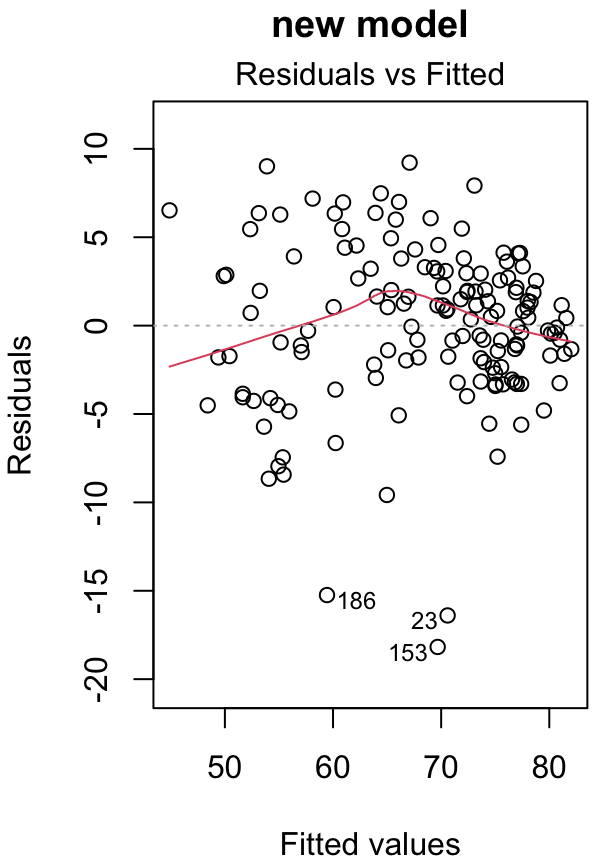
```
## [1] "new summary (log)"
```

```
print(summary(model7_log))
```

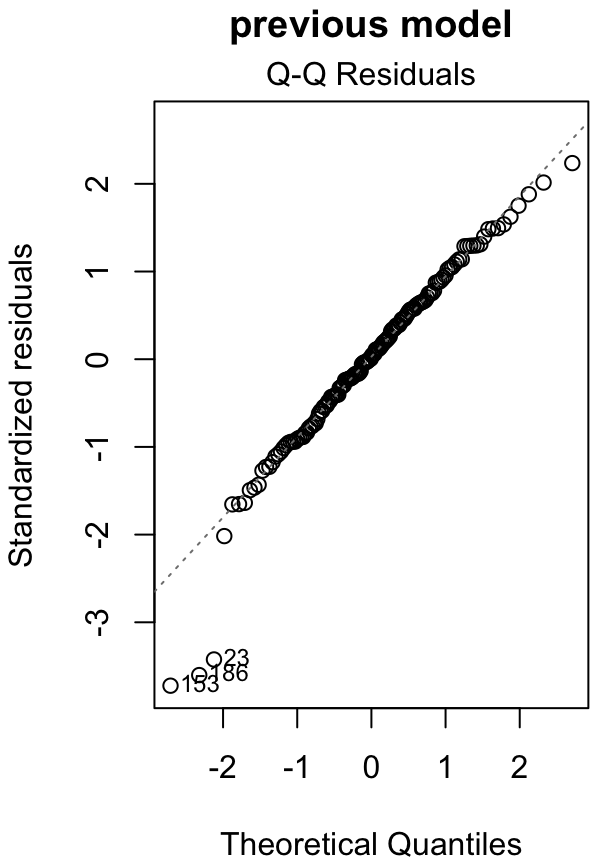
```
##
## Call:
## lm(formula = LifeExpectancy ~ Health + Internet + BirthRate +
##     ElderlyPop + CO2 + GDP_log + Cell, data = countries)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.1778  -2.3543   0.4403   2.8919   9.2191
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 66.78344    5.70127  11.714  < 2e-16 ***
## Health       0.16668    0.09732   1.713  0.08898 .
## Internet     0.04792    0.02966   1.615  0.10847
## BirthRate   -0.64016    0.07312  -8.755 5.94e-15 ***
## ElderlyPop  -0.41233    0.14294  -2.885  0.00454 **
## CO2         -0.13027    0.08152  -1.598  0.11229
## GDP_log      1.70924    0.65293   2.618  0.00982 **
## Cell         0.02591    0.01321   1.961  0.05187 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.613 on 140 degrees of freedom
## Multiple R-squared:  0.807,  Adjusted R-squared:  0.7974
## F-statistic: 83.64 on 7 and 140 DF,  p-value: < 2.2e-16
```

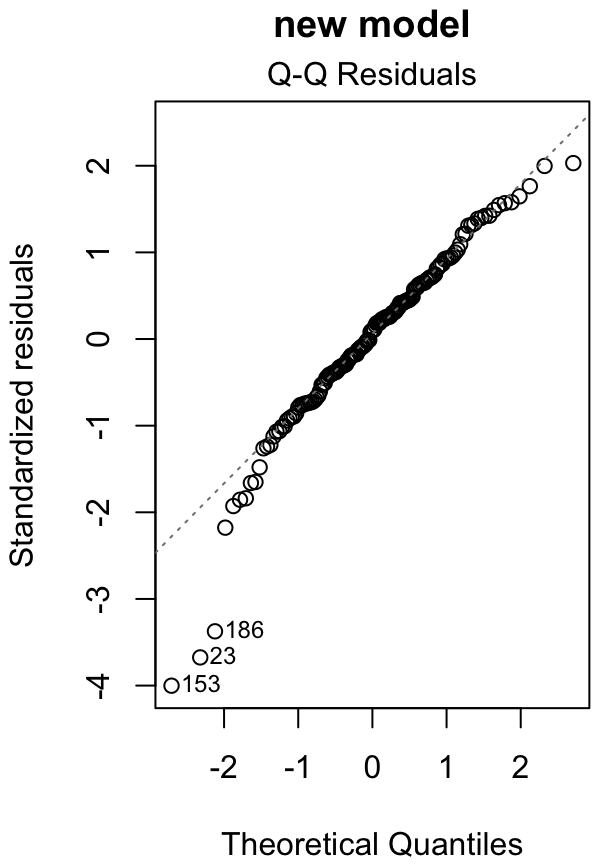
```
par(mfrow = c(1, 2))



resid2 = plot(model7, which = 1, main = "previous model")
resid3 = plot(model7_log, which = 1, main = "new model")
```






```
qq2 = plot(model7, which = 2, main = "previous model")
qq3 = plot(model7_log, which = 2, main = "new model")
```

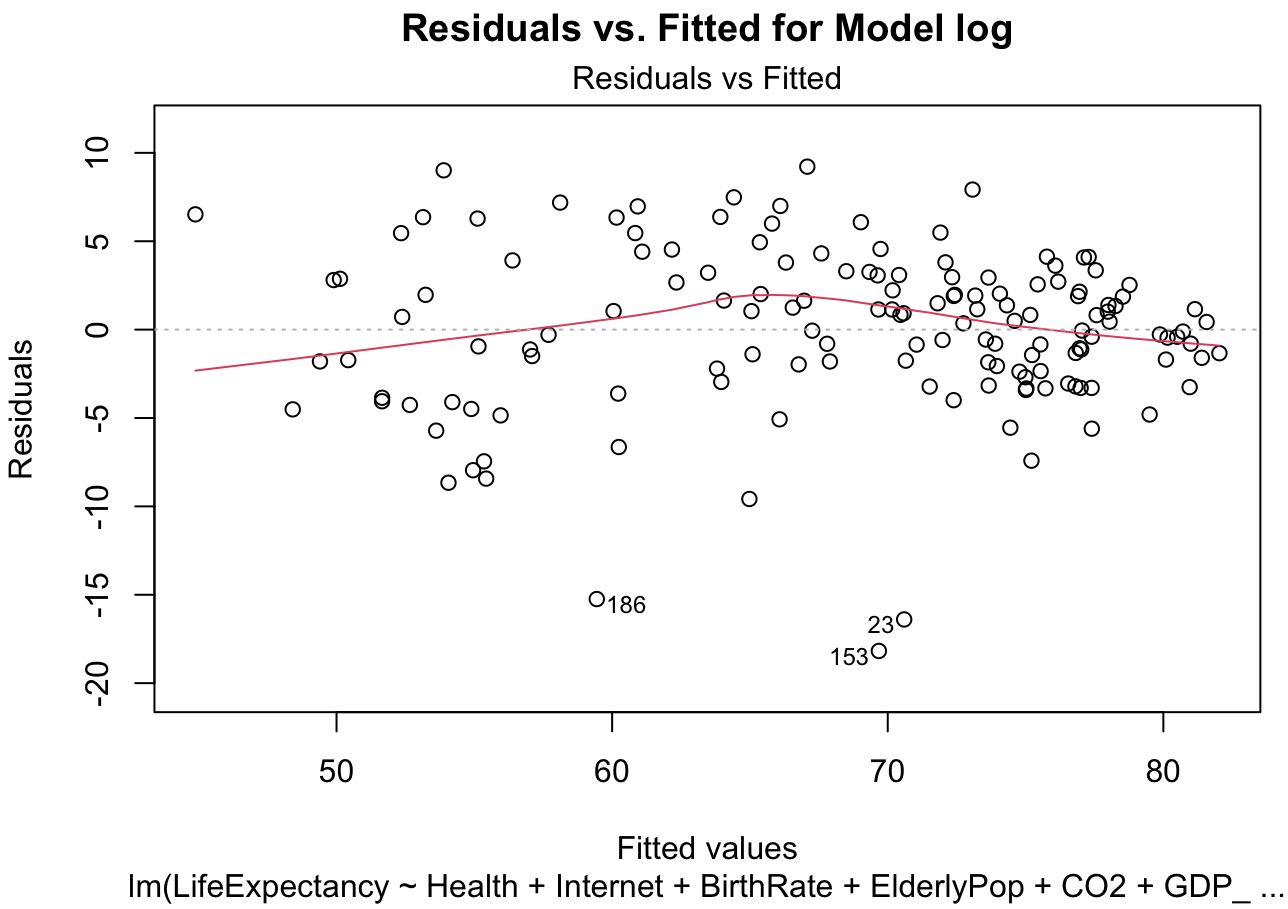




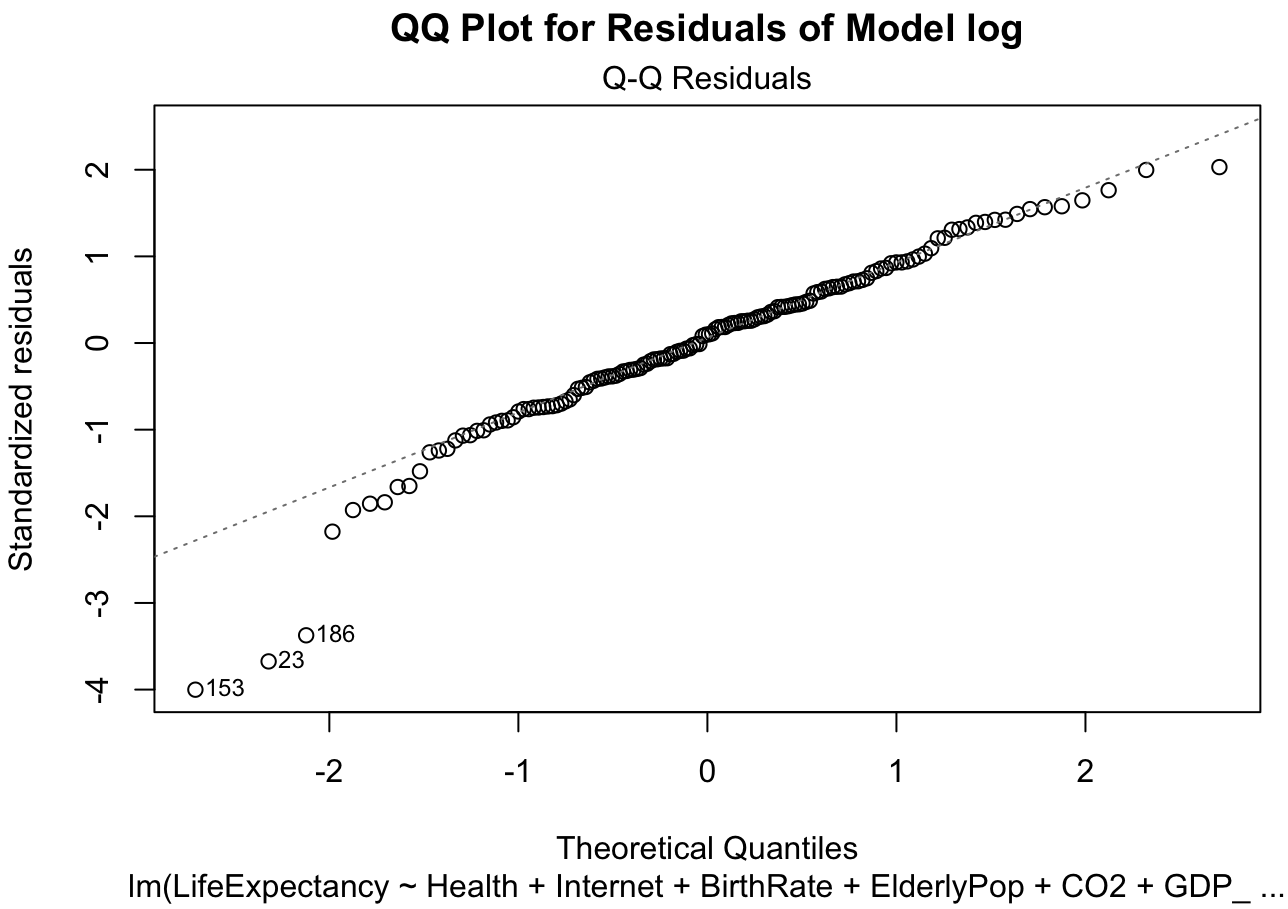

```
par(mfrow = c(1, 1))
```

It seems to make the residual plot more centered at 0, but it made the errors slightly less normal

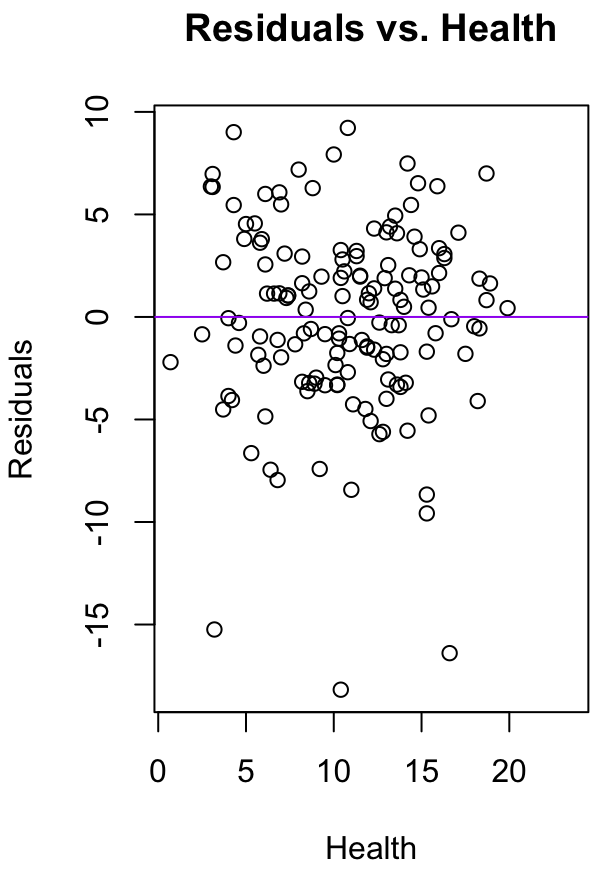
```
# to see the plots larger
resid3 = plot(model7_log, which = 1, main = "Residuals vs. Fitted for Model log")
```

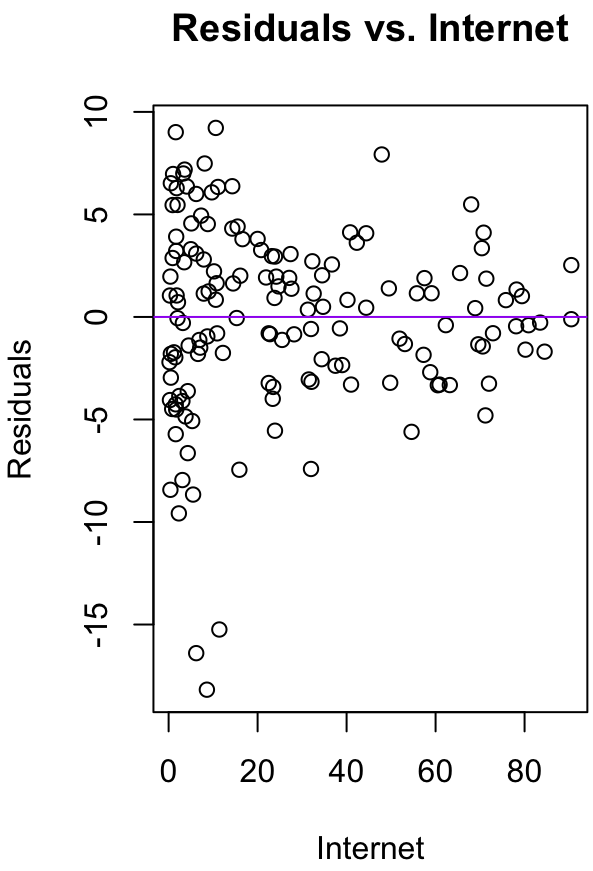


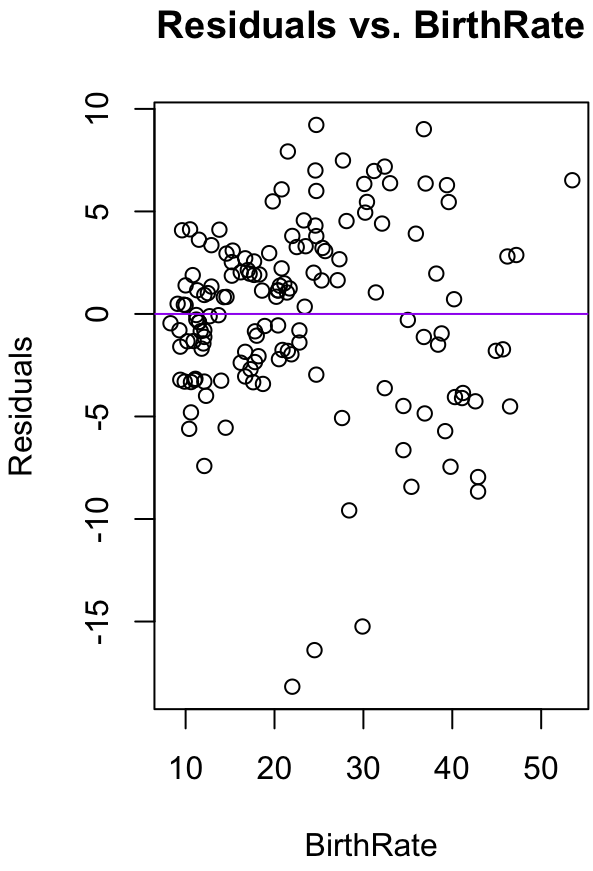

```
qq3 = plot(model7_log, which = 2, main = "QQ Plot for Residuals of Model log")
```

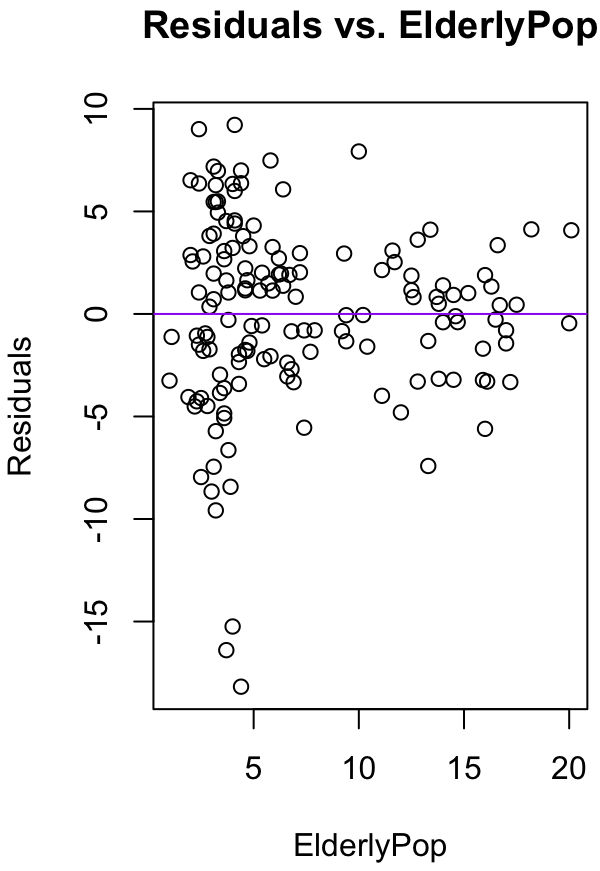

QQ Plot for Residuals of Model log
Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(LifeExpectancy ~ Health + Internet + BirthRate + ElderlyPop + CO2 + GDP_ ...
186
23
153

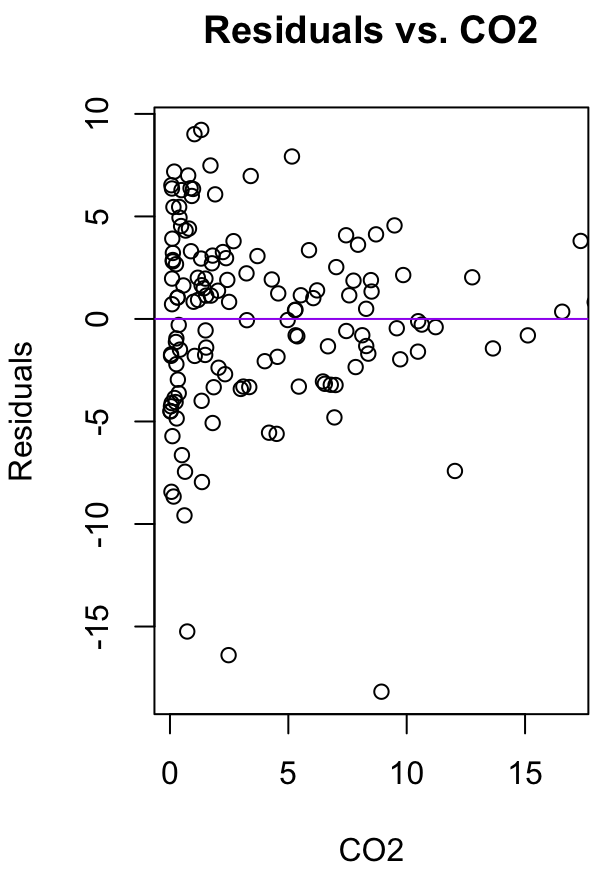
```
# lets see each of the predictors plotted against the residuals once again
par(mfrow = c(1, 2))
predictors = c("Health", "Internet", "BirthRate", "ElderlyPop", "CO2", "GDP_log", "Cel
l")

for (predictor in predictors) {
  data = countries[[predictor]]
  q = quantile(data, c(0.25, 0.75), na.rm = TRUE)
  iqr = q[2] - q[1]
  lower_limit = q[1] - 1.5 * iqr
  upper_limit = q[2] + 1.5 * iqr

  x_limit_min = max(c(min(data), lower_limit))
  x_limit_max = min(c(max(data), upper_limit))


  plot(countries[[predictor]], residuals(model7_log), main = paste("Residuals vs.", pred
ictor),
       xlab = predictor, ylab = "Residuals", xlim = c(x_limit_min, x_limit_max))
    abline(h = 0, col = "purple")
    identify(countries[[predictor]], countries$LifeExpectancy, labels = rownames(countri
es))



}
```

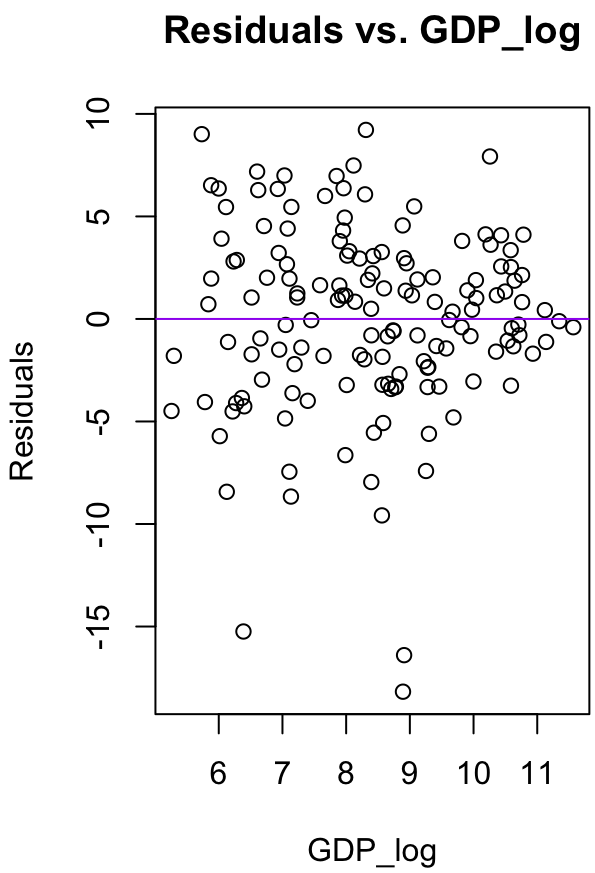

Residuals vs. Health
Residuals
Health
10
5
0
-5
-10
-15
0
5
10
15
20


Residuals vs. Internet
Residuals
Internet
10
5
0
-5
-10
-15
0
20
40
60
80

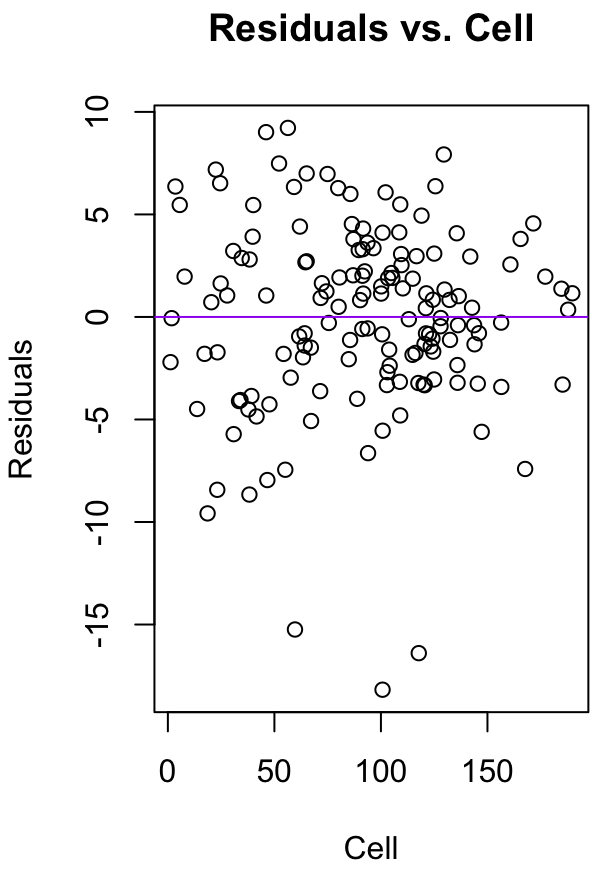

Residuals vs. BirthRate
Residuals
BirthRate
10
5
0
-5
-10
-15
10
20
30
40
50


Residuals vs. ElderlyPop
Residuals
ElderlyPop
10
5
0
-5
-10
-15
5
10
15
20






```
par(mfrow = c(1, 1))
```

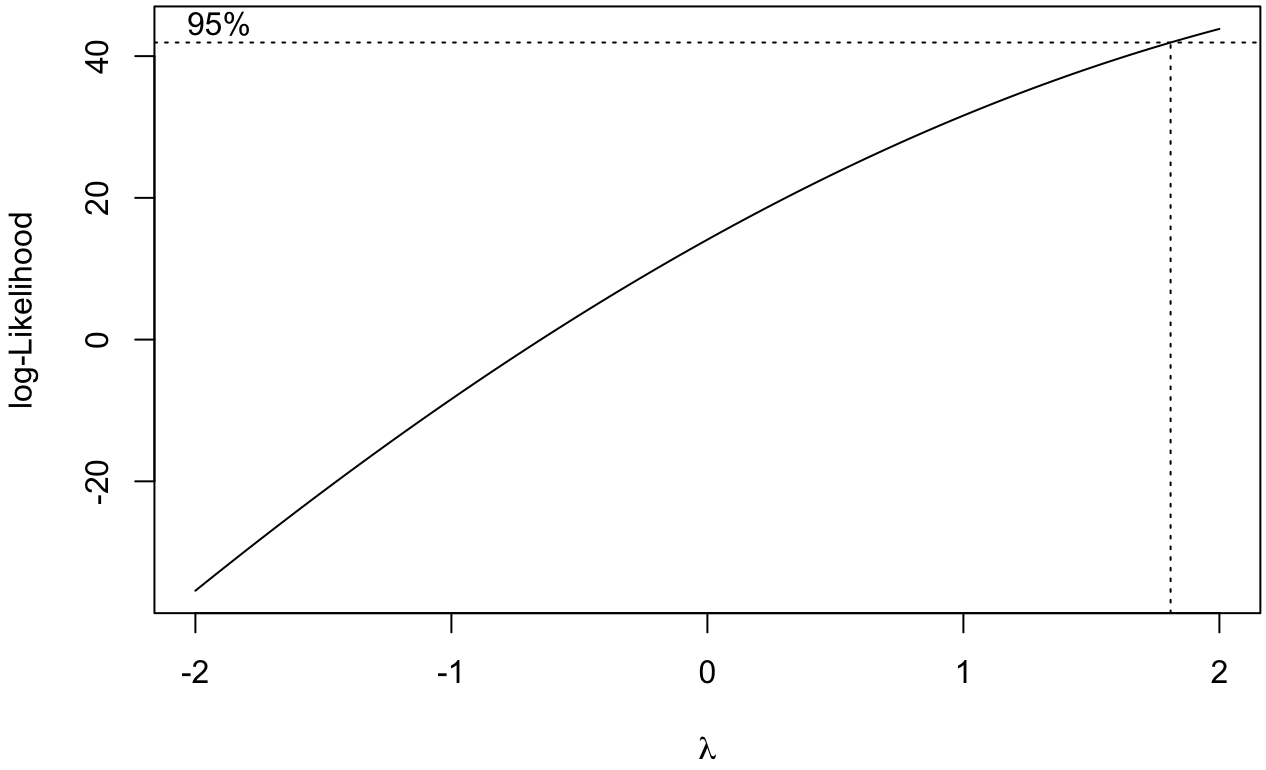


these

plots look much better!

But now the data is slightly non normal. Lets see if transforming the y would help

```
library(MASS)
par(mfrow = c(1,1))
predictor_data = countries[c("Health", "Internet", "BirthRate", "ElderlyPop", "CO2", "GD
P_log", "Cell")]

result = boxcox(countries$LifeExpectancy ~ ., data = predictor_data, lambda = seq(-2, 2,
by = 0.1))
```



our

results indicate that a square transformation of y is most appropriate

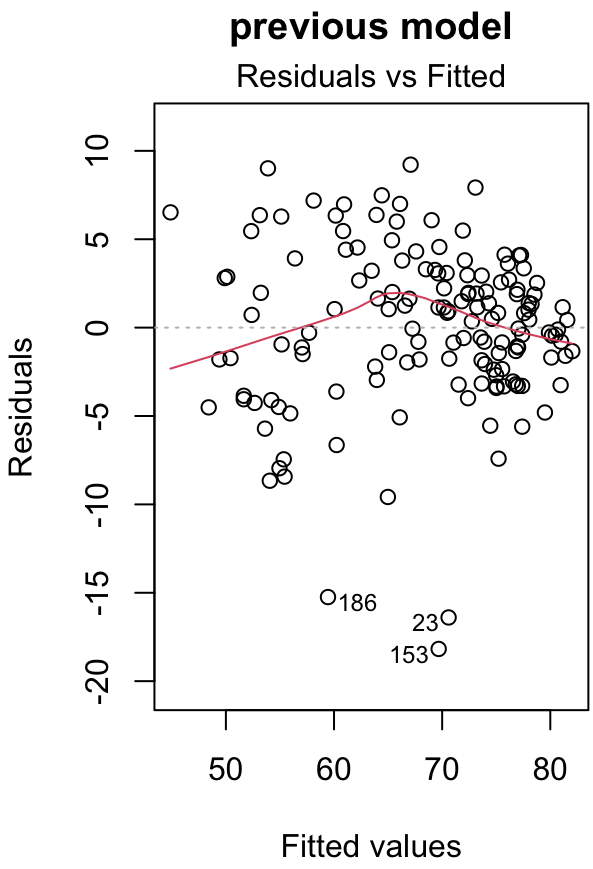
```
countries$LifeExpectancySquared = countries$LifeExpectancy^2


model7_ysquared = lm(LifeExpectancySquared ~ Health + Internet + BirthRate + ElderlyPop
+ CO2 + GDP_log + Cell, data = countries)



print("new summary (y squared)")
```

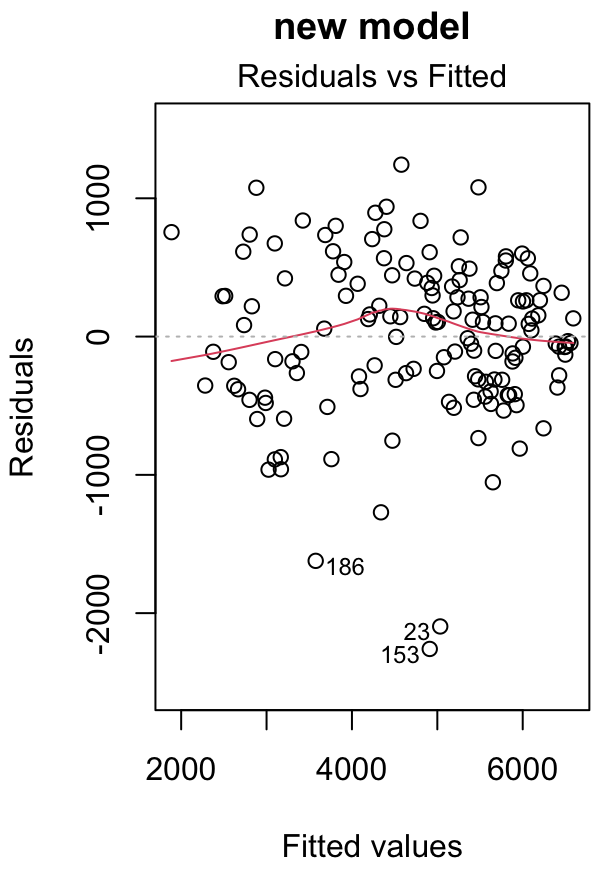
```
## [1] "new summary (y squared)"
```

```
print(summary(model7_ysquared))
```

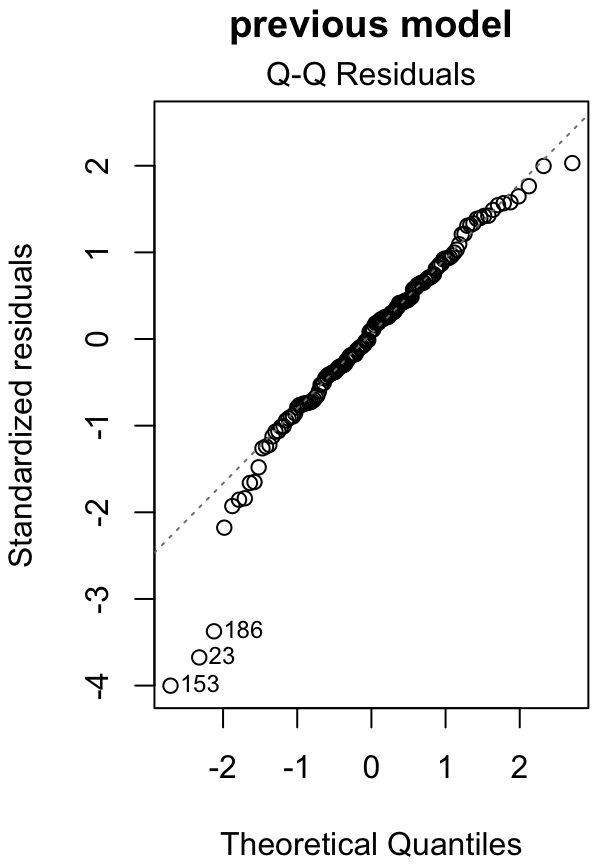
```
##
## Call:
## lm(formula = LifeExpectancySquared ~ Health + Internet + BirthRate +
##     ElderlyPop + CO2 + GDP_log + Cell, data = countries)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2258.92  -316.23    70.33   383.09  1243.28
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3977.560    709.460   5.606 1.06e-07 ***
## Health        21.978     12.111   1.815  0.07170 .
## Internet       7.415      3.691   2.009  0.04650 *
## BirthRate    -72.869      9.099  -8.009 4.04e-13 ***
## ElderlyPop   -43.310     17.787  -2.435  0.01615 *
## CO2          -15.157     10.144  -1.494  0.13739
## GDP_log      253.192     81.249   3.116  0.00222 **
## Cell           3.165      1.644   1.925  0.05629 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 574 on 140 degrees of freedom
## Multiple R-squared:  0.8199, Adjusted R-squared:  0.8109
## F-statistic: 91.04 on 7 and 140 DF,  p-value: < 2.2e-16
```

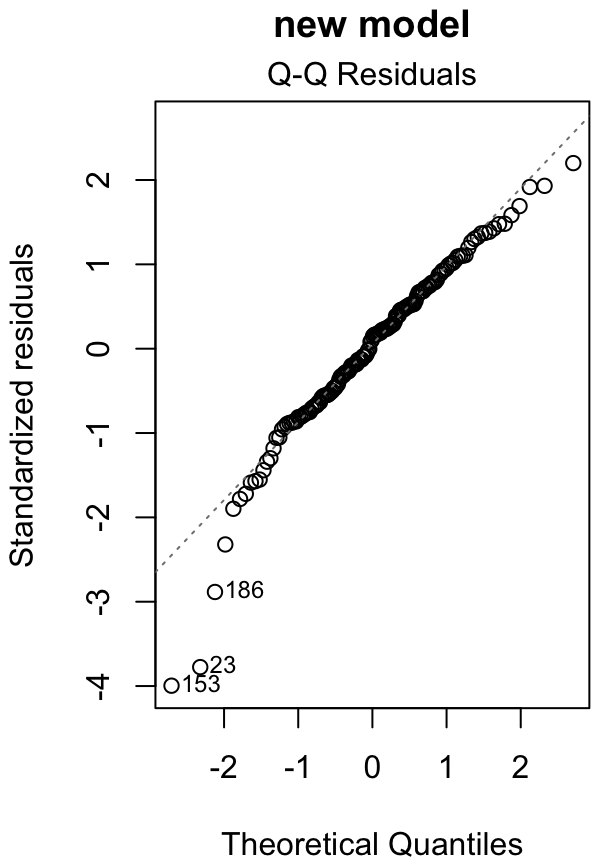
```
par(mfrow = c(1, 2))



resid2 = plot(model7_log, which = 1, main = "previous model")
resid3 = plot(model7_ysquared, which = 1, main = "new model")
```






```
qq2 = plot(model7_log, which = 2, main = "previous model")
qq3 = plot(model7_ysquared, which = 2, main = "new model")
```

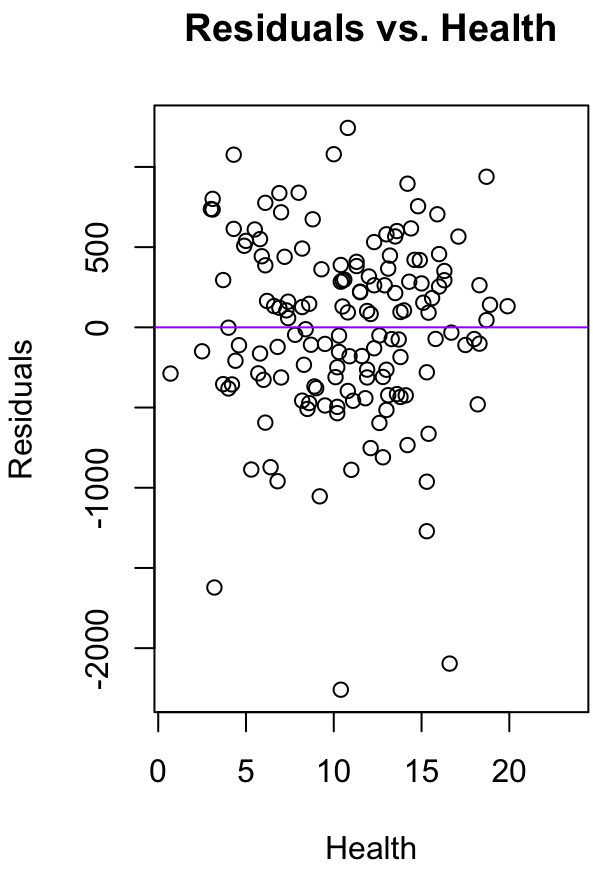


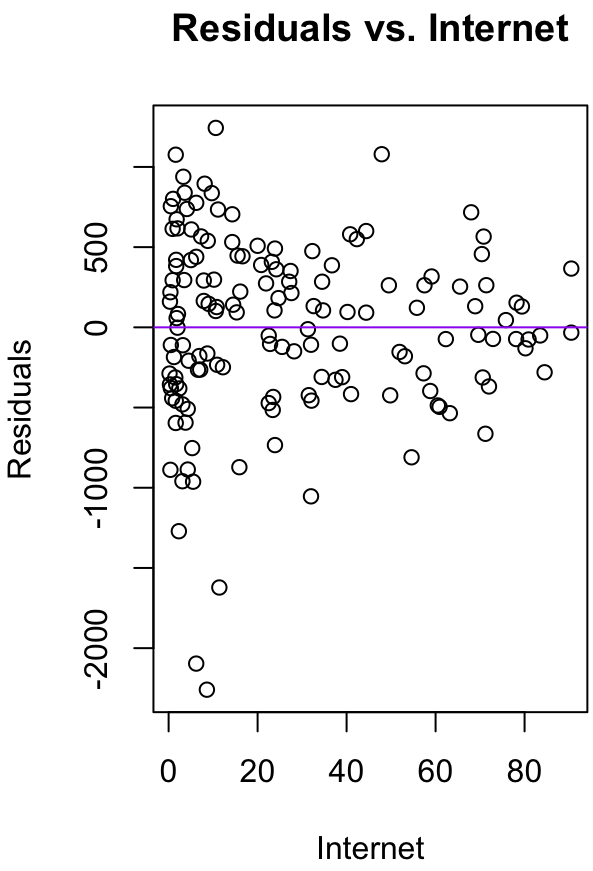


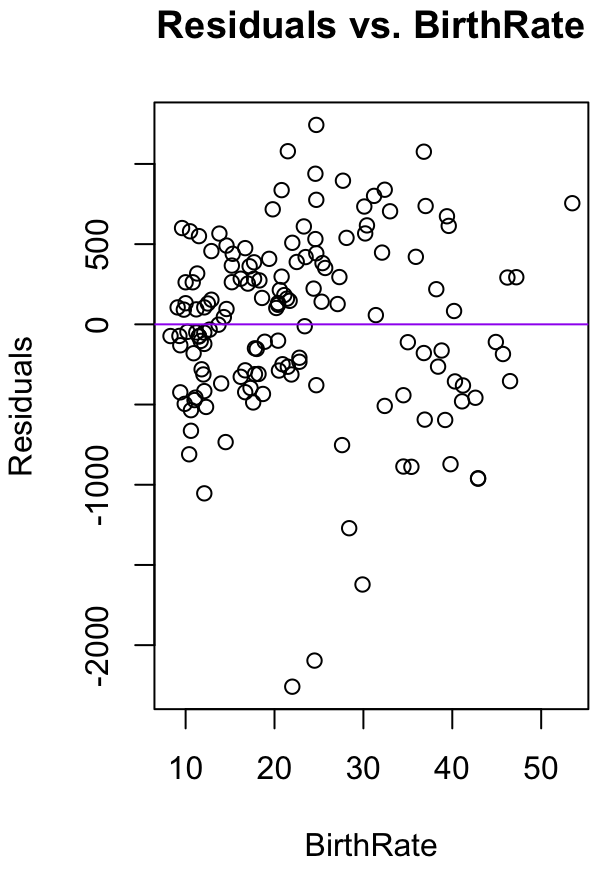

```
par(mfrow = c(1, 1))
```

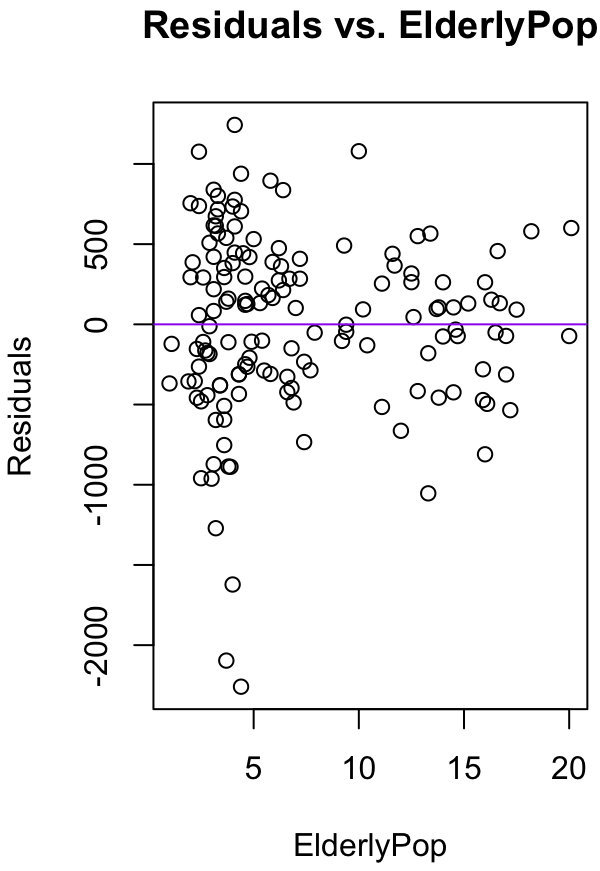
This model looks a bit better! It has brought some of our big outliers closer to the line in the qq plot. But the errors still look slightly non normal. Lets try box cox again to see if we should transform y a second time

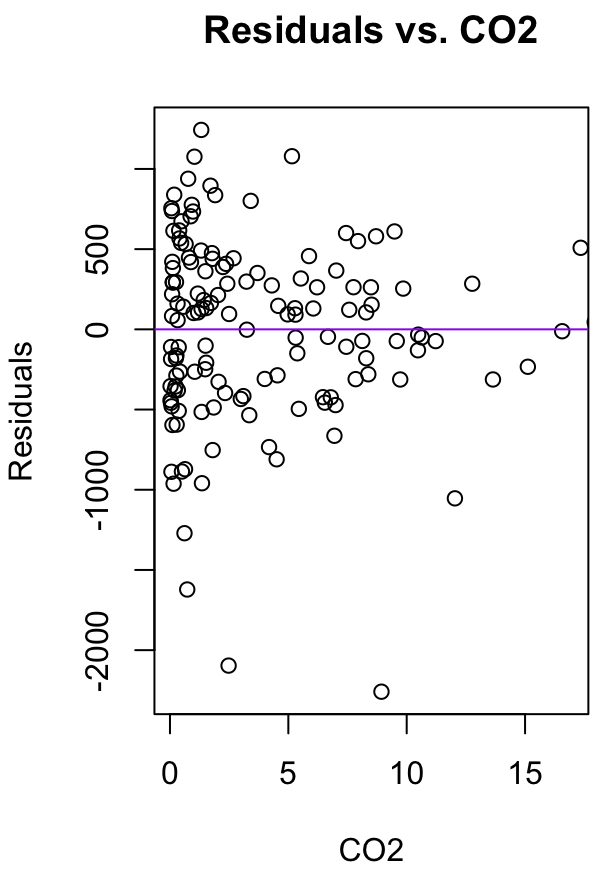
```
# first, lets make sure all our predictors are still looking good
par(mfrow = c(1, 2))
predictors = c("Health", "Internet", "BirthRate", "ElderlyPop", "CO2", "GDP_log", "Cel
l")

for (predictor in predictors) {
  data = countries[[predictor]]
  q = quantile(data, c(0.25, 0.75), na.rm = TRUE)
  iqr = q[2] - q[1]
  lower_limit = q[1] - 1.5 * iqr
  upper_limit = q[2] + 1.5 * iqr

  x_limit_min = max(c(min(data), lower_limit))
  x_limit_max = min(c(max(data), upper_limit))


  plot(countries[[predictor]], residuals(model7_ysquared), main = paste("Residuals vs.",
predictor),
       xlab = predictor, ylab = "Residuals", xlim = c(x_limit_min, x_limit_max))
    abline(h = 0, col = "purple")
    identify(countries[[predictor]], countries$LifeExpectancy, labels = rownames(countri
es))



}
```

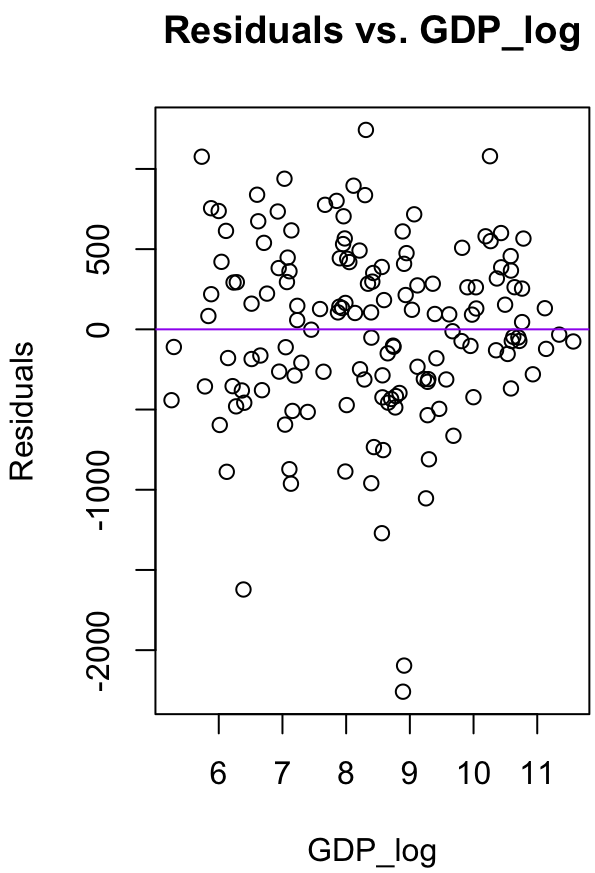

Residuals vs. Health
Residuals
500
0
-1000
-2000
0
5
10
15
20
Health


Residuals vs. Internet
Residuals
500
0
-1000
-2000
0
20
40
60
80
Internet

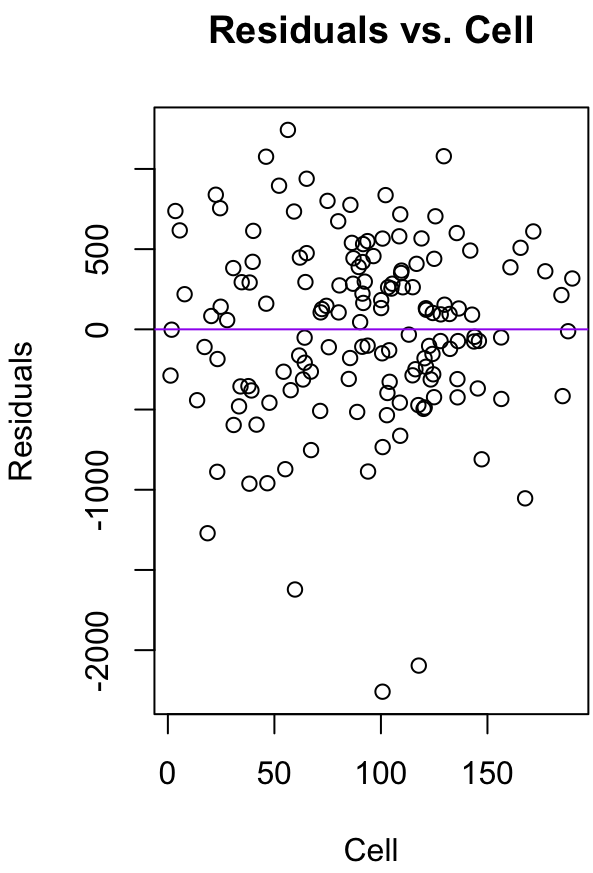

Residuals vs. BirthRate
Residuals
500
0
-1000
-2000
10
20
30
40
50
BirthRate


Residuals vs. ElderlyPop
Residuals
500
0
-1000
-2000
5
10
15
20
ElderlyPop





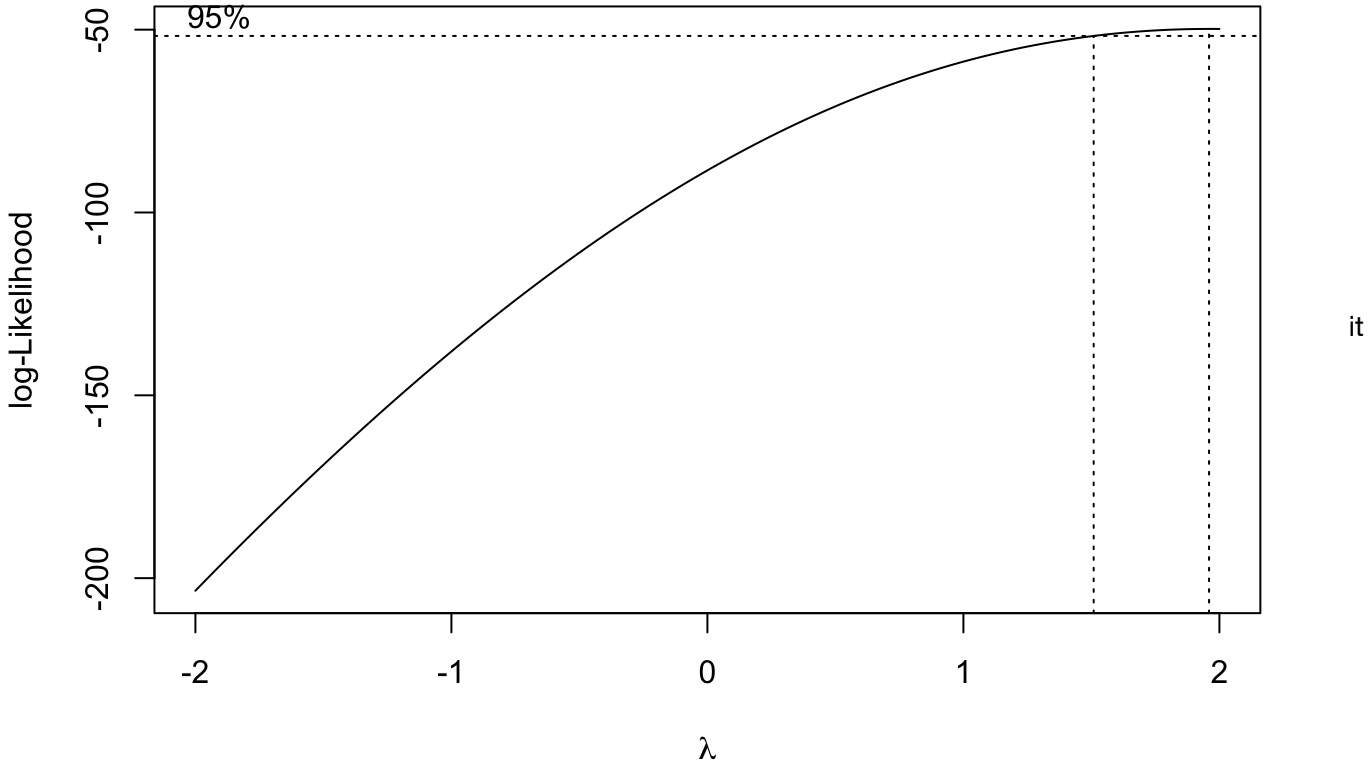

```
par(mfrow = c(1, 1))
```



Still

look good

```
# lets see if transforming y again will help normalize our errors
library(MASS)
predictor_data = countries[c("Health", "Internet", "BirthRate", "ElderlyPop", "CO2", "GD
P_log", "Cell")]
result = boxcox(countries$LifeExpectancySquared ~ ., data = predictor_data, lambda = seq
(-2, 2, by = 0.1))
```



looks like squaring y is the best transformation

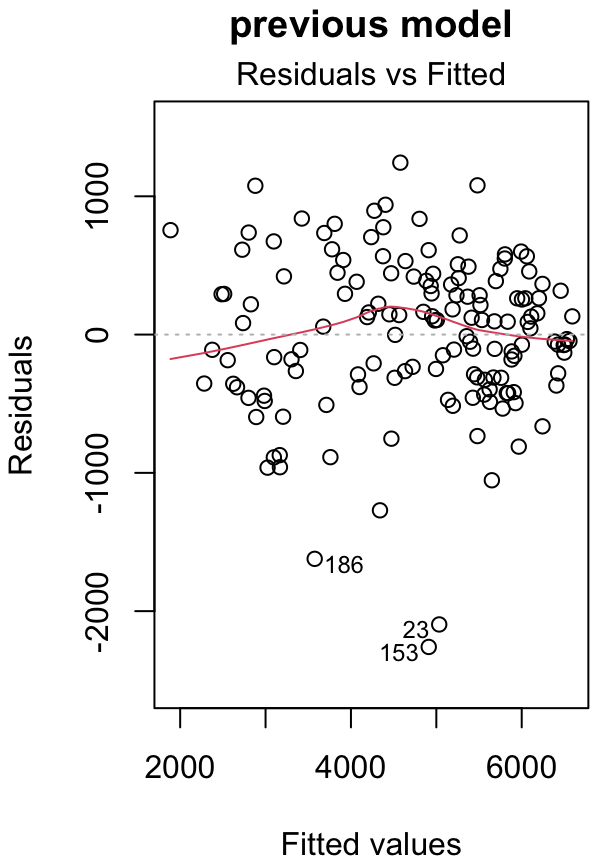
```
countries$LifeExpectancy_4 = countries$LifeExpectancySquared^2


model7_y4 = lm(LifeExpectancy_4 ~ Health + Internet + BirthRate + ElderlyPop + CO2 + GDP
_log + Cell, data = countries)



print("new summary (y squared)")
```

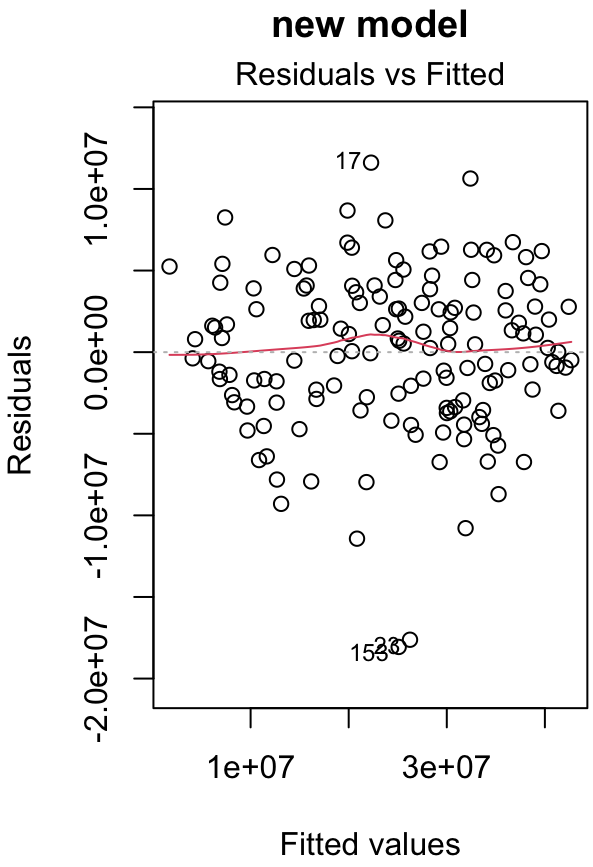
```
## [1] "new summary (y squared)"
```

```
print(summary(model7_y4))
```

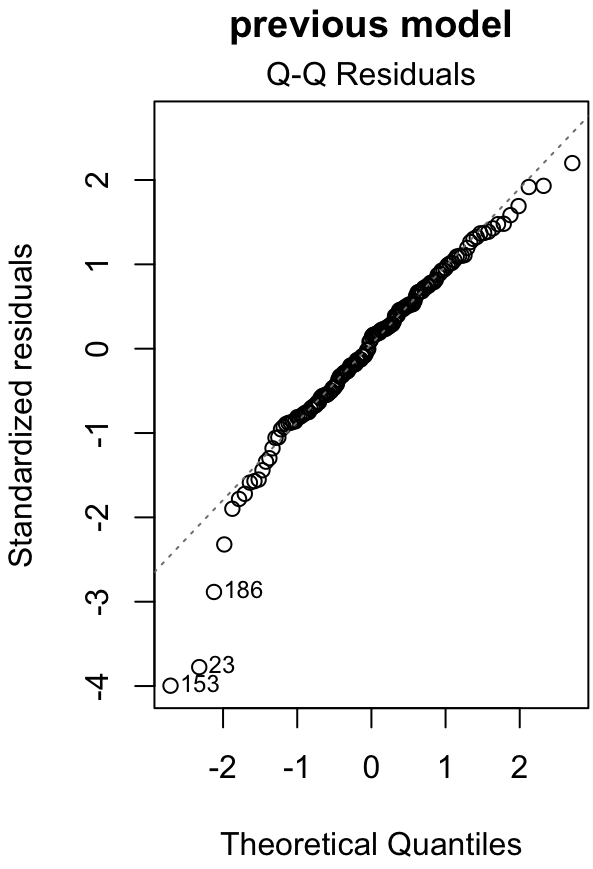
```
##
## Call:
## lm(formula = LifeExpectancy_4 ~ Health + Internet + BirthRate +
##     ElderlyPop + CO2 + GDP_log + Cell, data = countries)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -18062145  -2983201    369779   3113564  11610832
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6839746    6100289   1.121  0.26412
## Health        201280     104134   1.933  0.05527 .
## Internet       85234      31740   2.685  0.00812 **
## BirthRate    -466233      78234  -5.959 1.95e-08 ***
## ElderlyPop   -201758     152944  -1.319  0.18927
## CO2          -106846      87226  -1.225  0.22266
## GDP_log      2832657     698623   4.055 8.30e-05 ***
## Cell           23235      14140   1.643  0.10258
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4936000 on 140 degrees of freedom
## Multiple R-squared:  0.8306, Adjusted R-squared:  0.8222
## F-statistic: 98.08 on 7 and 140 DF,  p-value: < 2.2e-16
```

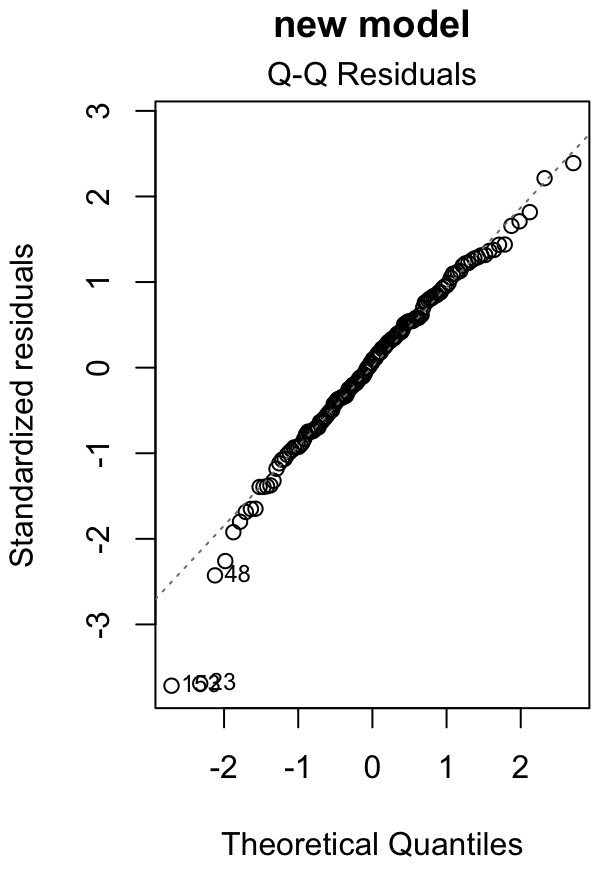
```
par(mfrow = c(1, 2))



resid2 = plot(model7_ysquared, which = 1, main = "previous model")
resid3 = plot(model7_y4, which = 1, main = "new model")
```






```
qq2 = plot(model7_ysquared, which = 2, main = "previous model")
qq3 = plot(model7_y4, which = 2, main = "new model")
```

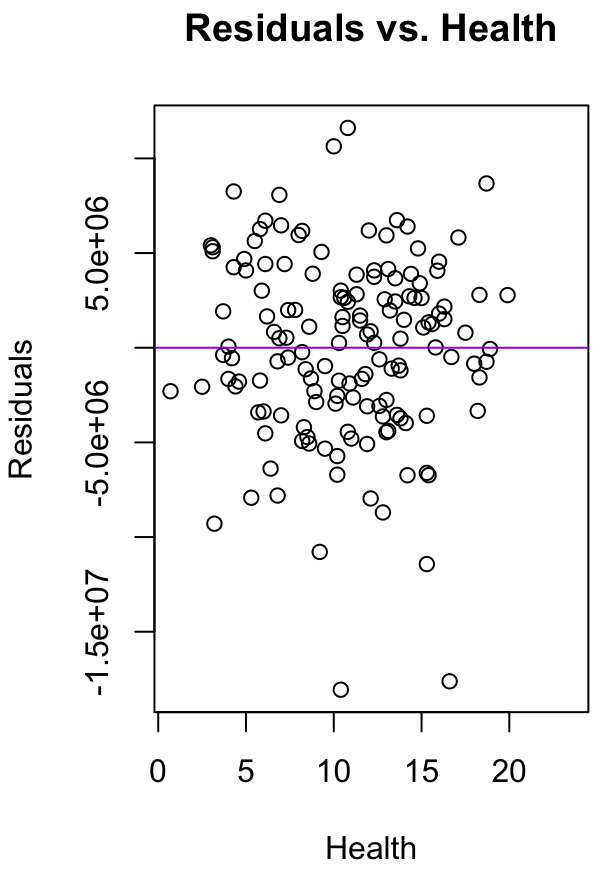


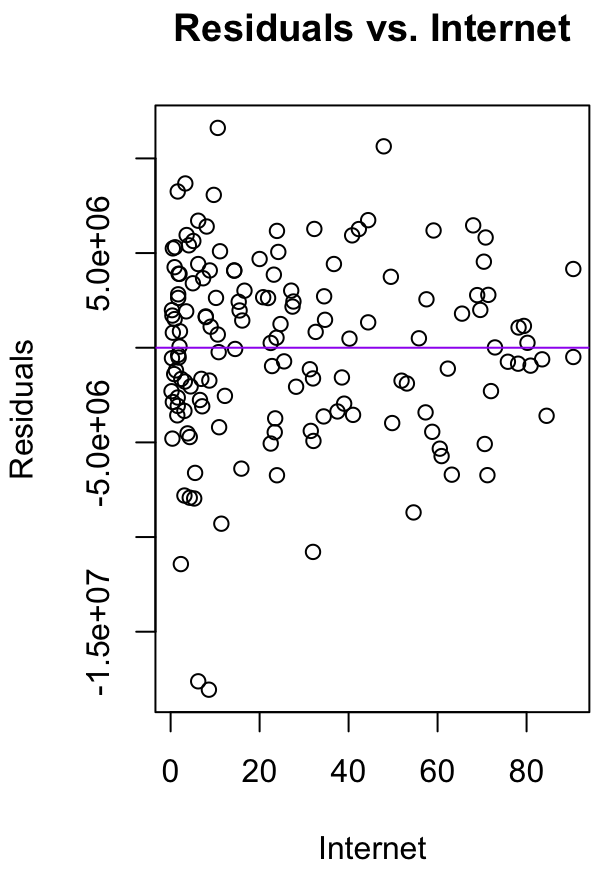


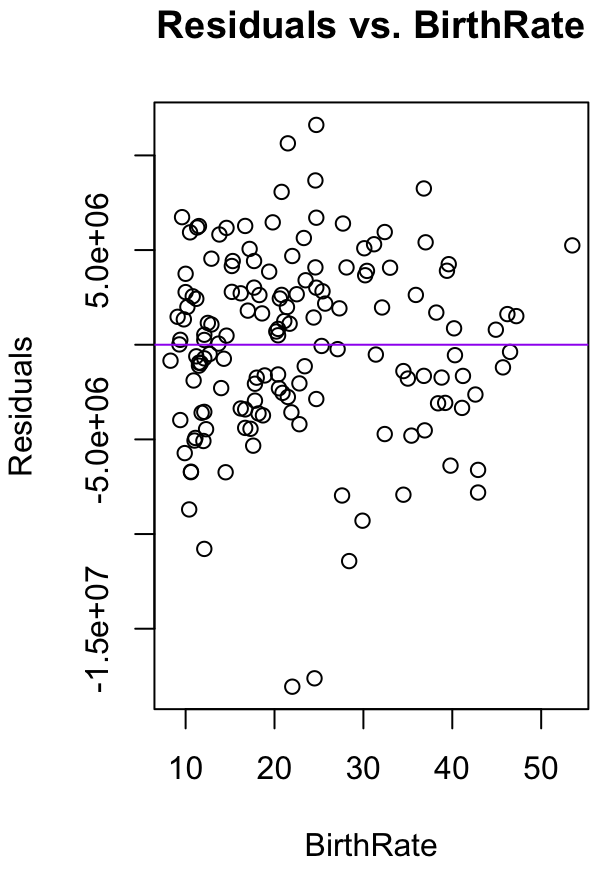

```
par(mfrow = c(1, 1))
```

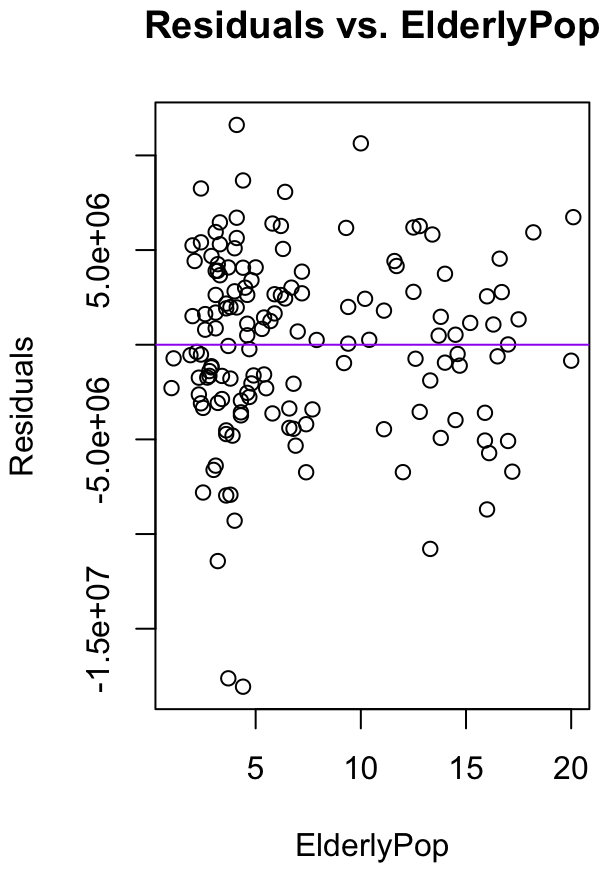
The errors look more normal!

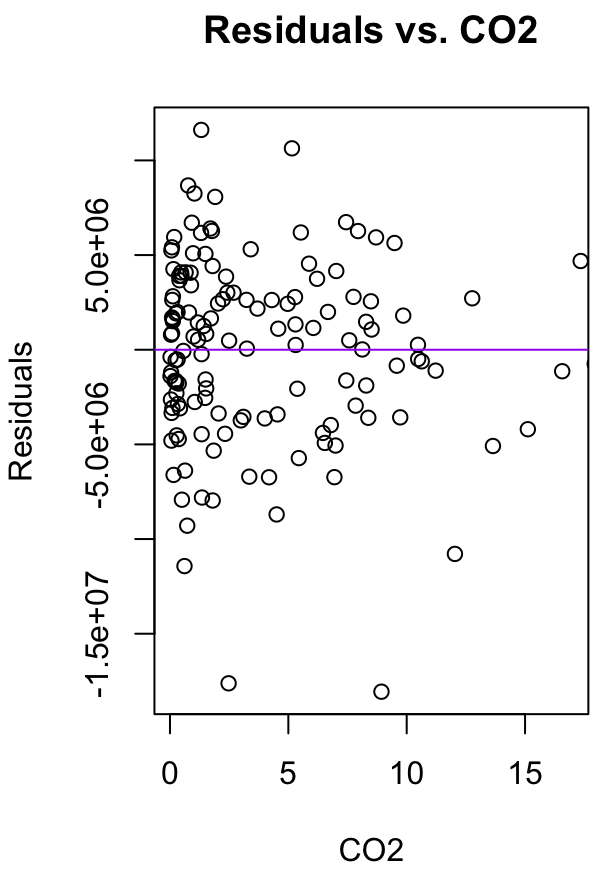
```
par(mfrow = c(1, 2))
predictors = c("Health", "Internet", "BirthRate", "ElderlyPop", "CO2", "GDP_log", "Cel
l")

for (predictor in predictors) {
  data = countries[[predictor]]
  q = quantile(data, c(0.25, 0.75), na.rm = TRUE)
  iqr = q[2] - q[1]
  lower_limit = q[1] - 1.5 * iqr
  upper_limit = q[2] + 1.5 * iqr

  x_limit_min = max(c(min(data), lower_limit))
  x_limit_max = min(c(max(data), upper_limit))


  plot(countries[[predictor]], residuals(model7_y4), main = paste("Residuals vs.", predi
ctor),
       xlab = predictor, ylab = "Residuals", xlim = c(x_limit_min, x_limit_max))
    abline(h = 0, col = "purple")
    identify(countries[[predictor]], countries$LifeExpectancy, labels = rownames(countri
es))



}
```

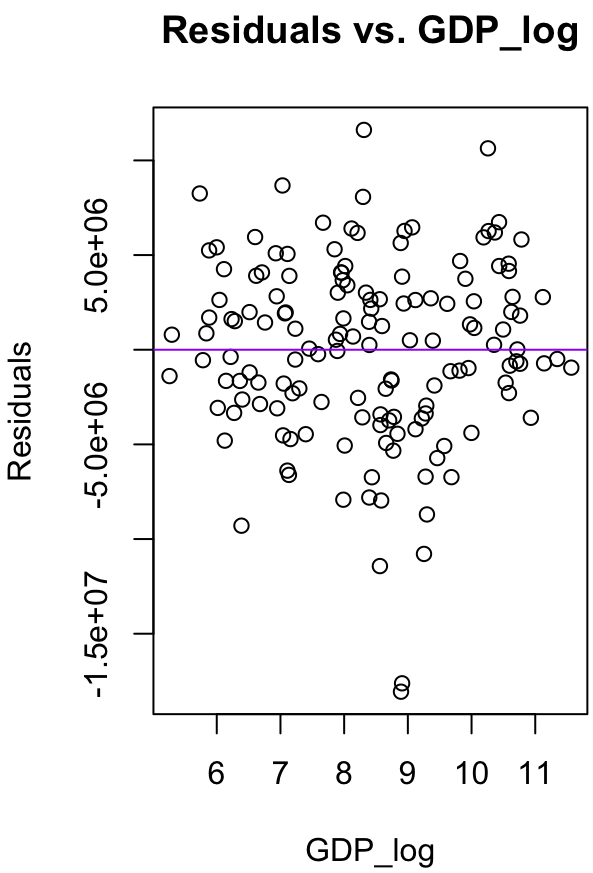

Residuals vs. Health
Residuals
5.0e+06
-5.0e+06
-1.5e+07
0
5
10
15
20
Health


Residuals vs. Internet
Residuals
5.0e+06
-5.0e+06
-1.5e+07
0
20
40
60
80
Internet

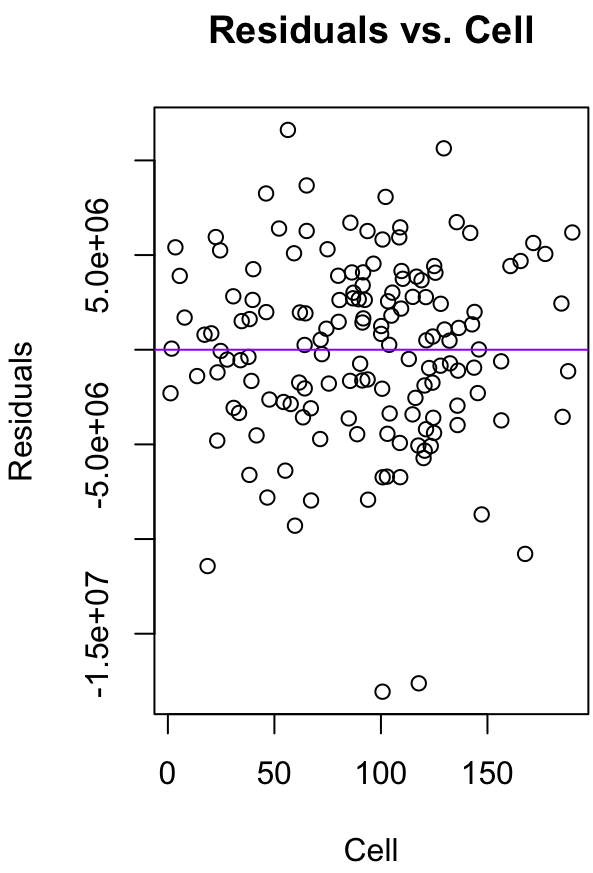

Residuals vs. BirthRate
Residuals
5.0e+06
-5.0e+06
-1.5e+07
10
20
30
40
50
BirthRate


Residuals vs. ElderlyPop
Residuals
5.0e+06
-5.0e+06
-1.5e+07
5
10
15
20
ElderlyPop






```
par(mfrow = c(1, 1))
```




```
# lets find mallow_cp
library("olsrr")
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:MASS':
##
##     cement
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
fullmodel = lm(LifeExpectancy_4 ~ GDP_log + BirthRate + LandArea + Population + Rural +
Health + Internet + ElderlyPop + CO2 + Cell, data = countries)

print("mallows CP:")
```

```
## [1] "mallows CP:"
```

```
ols_mallows_cp(model7_y4, fullmodel)
```

```
## [1] 9.038934
```

```
print("p+1 = 7 (we are looking for the marlows cp closest to p+1")
```

```
## [1] "p+1 = 7 (we are looking for the marlows cp closest to p+1"
```

```
# lets do one final check that this is the best model

library(leaps)

predictors = c("LandArea", "Rural", "Population", "Health", "Internet", "BirthRate", "El
derlyPop", "CO2", "GDP_log", "Cell")
temp_df = data.frame(
  LifeExpectancy = countries$LifeExpectancy_4,
  LandArea = countries$LandArea,
  Rural = countries$Rural,
  Population = countries$Population,
  Health = countries$Health,
  Internet = countries$Internet,
  BirthRate = countries$BirthRate,
  ElderlyPop = countries$ElderlyPop,
  CO2 = countries$CO2,
  GDP = countries$GDP_log,
  Cell = countries$Cell
)



max_predictors = length(predictors)
temp_model = regsubsets(LifeExpectancy ~ ., data = temp_df, nbest = 1, nvmax = max_predi
ctors)
summary_model = summary(temp_model)
summary_model
```

```
## Subset selection object
## Call: regsubsets.formula(LifeExpectancy ~ ., data = temp_df, nbest = 1,
##     nvmax = max_predictors)
## 10 Variables  (and intercept)
##            Forced in Forced out
## LandArea       FALSE      FALSE
## Rural          FALSE      FALSE
## Population     FALSE      FALSE
## Health         FALSE      FALSE
## Internet       FALSE      FALSE
## BirthRate      FALSE      FALSE
## ElderlyPop     FALSE      FALSE
## CO2            FALSE      FALSE
## GDP            FALSE      FALSE
## Cell           FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##          LandArea Rural Population Health Internet BirthRate ElderlyPop CO2
## 1  ( 1 )  " "      " "   " "        " "    " "      " "       " "        " "
## 2  ( 1 )  " "      " "   " "        " "    " "      "*"       " "        " "
## 3  ( 1 )  " "      " "   " "        " "    "*"      "*"       " "        " "
## 4  ( 1 )  "*"      " "   " "        " "    "*"      "*"       " "        " "
## 5  ( 1 )  "*"      " "   " "        "*"    "*"      "*"       " "        " "
## 6  ( 1 )  "*"      " "   " "        "*"    "*"      "*"       " "        " "
## 7  ( 1 )  "*"      " "   " "        "*"    "*"      "*"       "*"        " "
## 8  ( 1 )  "*"      " "   " "        "*"    "*"      "*"       "*"        "*"
## 9  ( 1 )  "*"      " "   "*"        "*"    "*"      "*"       "*"        "*"
## 10  ( 1 ) "*"      "*"   "*"        "*"    "*"      "*"       "*"        "*"
##          GDP Cell
## 1  ( 1 )  "*" " "
## 2  ( 1 )  "*" " "
## 3  ( 1 )  "*" " "
## 4  ( 1 )  "*" " "
## 5  ( 1 )  "*" " "
## 6  ( 1 )  "*" "*"
## 7  ( 1 )  "*" "*"
## 8  ( 1 )  "*" "*"
## 9  ( 1 )  "*" "*"
## 10  ( 1 ) "*" "*"
```

```
summary_model$rsq
```

```
##  [1] 0.7485828 0.8104417 0.8198888 0.8250582 0.8291629 0.8325690 0.8335240
##  [8] 0.8347216 0.8352329 0.8354733
```

```
summary_model$cp
```

```
##  [1] 65.352923 15.843602  9.977026  7.672564  6.254620  5.418374  6.623146
##  [8]  7.625907  9.200188 11.000000
```

```
summary_model$adjr2
```

```
##  [1] 0.7468608 0.8078271 0.8161365 0.8201647 0.8231475 0.8254443 0.8252002
##  [8] 0.8252092 0.8244872 0.8234640
```

```
top_adjR2 = order(-summary_model$adjr2)[1:4]
top_Cp = order(summary_model$cp)[1:4]
top_r2 = order(-summary_model$rsq)[1:4]

top_adjR2
```

```
## [1] 6 8 7 9
```

```
top_Cp
```

```
## [1] 6 5 7 8
```

```
top_r2
```

```
## [1] 10  9  8  7
```

6 7 and 8 look most promising

```
summary_model$rsq[6]
```

```
## [1] 0.832569
```

```
summary_model$cp[6]
```

```
## [1] 5.418374
```

```
summary_model$adjr2[6]
```

```
## [1] 0.8254443
```

```
cat("\n\n")
```

```
summary_model$rsq[7]
```

```
## [1] 0.833524
```

```
summary_model$cp[7]
```

```
## [1] 6.623146
```

```
summary_model$adjr2[7]
```

```
## [1] 0.8252002
```

```
cat("\n\n")
```

```
summary_model$rsq[8]
```

```
## [1] 0.8347216
```

```
summary_model$cp[8]
```

```
## [1] 7.625907
```

```
summary_model$adjr2[8]
```

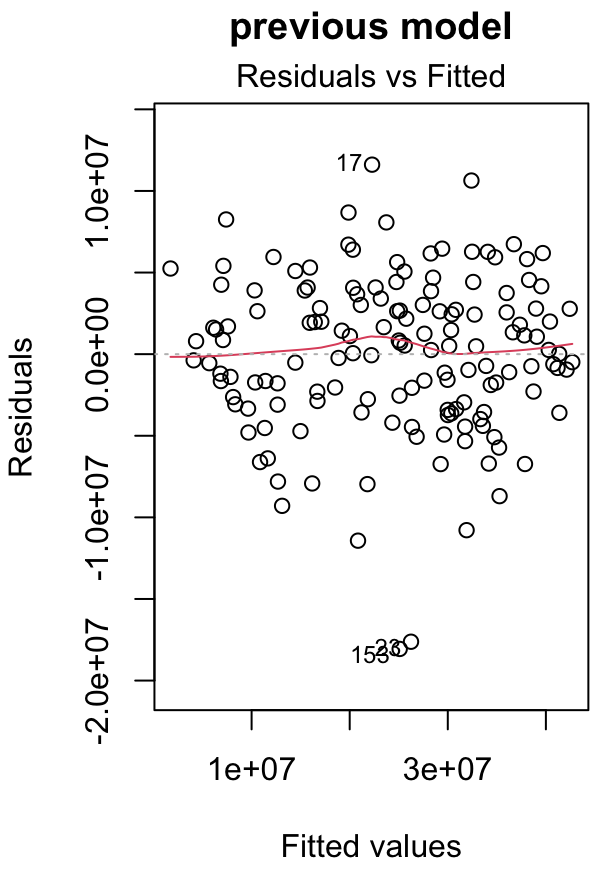
```
## [1] 0.8252092
```

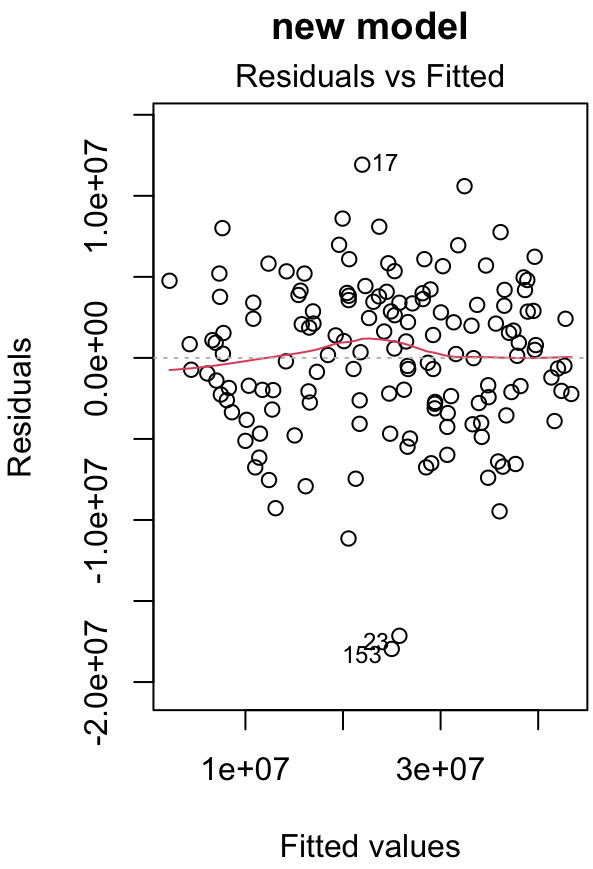
```
# model 6 looks the best because the R^2 and adjusted R^2 values for all predictors are very similar, but the cp value is a lot lower. Also, model 6 does not include elderly population, which was the predictor we flagged as potentially having too high of multicolinearity with birthrate. Thus our final model will be less influenced by multicolinearity
```

```
model6 = lm(LifeExpectancy_4 ~ LandArea + Health + Internet + BirthRate + GDP_log + Cell, data = countries)



print(summary(model6))
```

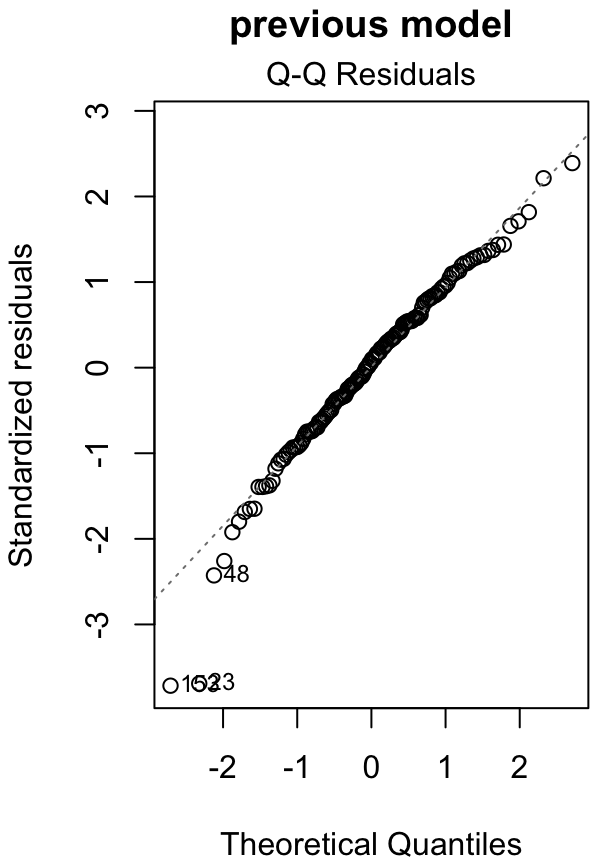
```
##
## Call:
## lm(formula = LifeExpectancy_4 ~ LandArea + Health + Internet +
##     BirthRate + GDP_log + Cell, data = countries)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -17957550  -2761641    303157   3422204  11922985
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.249e+06  5.331e+06   1.172   0.2431
## LandArea    -4.211e-01  2.075e-01  -2.029   0.0443 *
## Health       2.063e+05  9.894e+04   2.085   0.0388 *
## Internet     6.458e+04  2.846e+04   2.269   0.0248 *
## BirthRate   -4.243e+05  6.586e+04  -6.443 1.72e-09 ***
## GDP_log      2.643e+06  6.093e+05   4.338 2.72e-05 ***
## Cell         2.368e+04  1.398e+04   1.694   0.0925 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4890000 on 141 degrees of freedom
## Multiple R-squared:  0.8326, Adjusted R-squared:  0.8254
## F-statistic: 116.9 on 6 and 141 DF,  p-value: < 2.2e-16
```

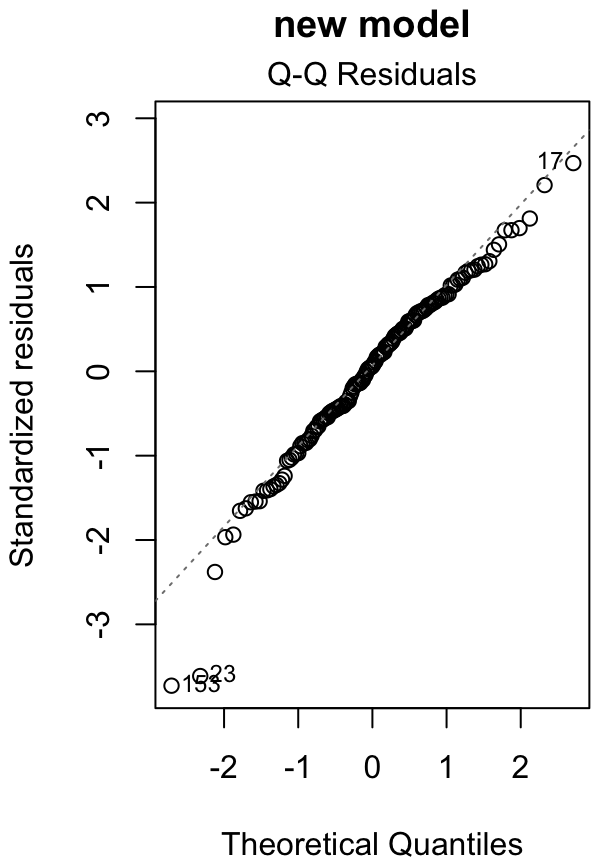
```
par(mfrow = c(1, 2))



resid2 = plot(model7_y4, which = 1, main = "previous model")
resid3 = plot(model6, which = 1, main = "new model")
```






```
qq2 = plot(model7_y4, which = 2, main = "previous model")
qq3 = plot(model6, which = 2, main = "new model")
```






```
par(mfrow = c(1, 1))

print("mallows CP:")
```

```
## [1] "mallows CP:"
```

```
ols_mallows_cp(model6, fullmodel)
```

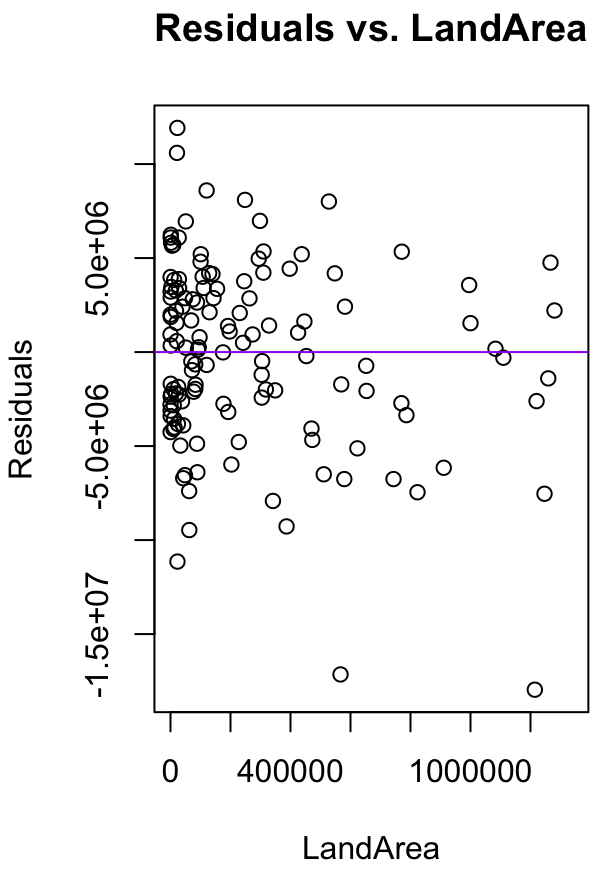
```
## [1] 5.418374
```

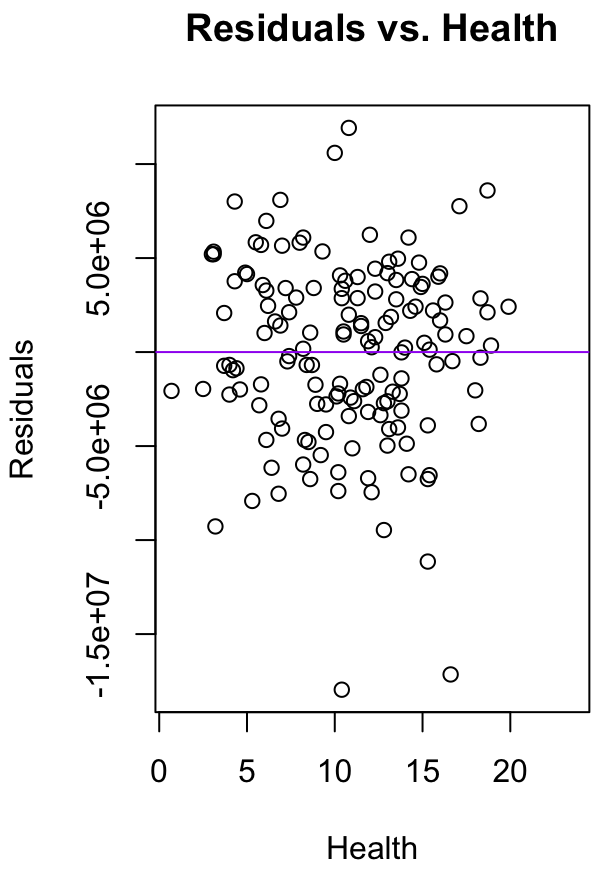
```
print("p+1 = 6 (we are looking for the marlows cp closest to p+1")
```

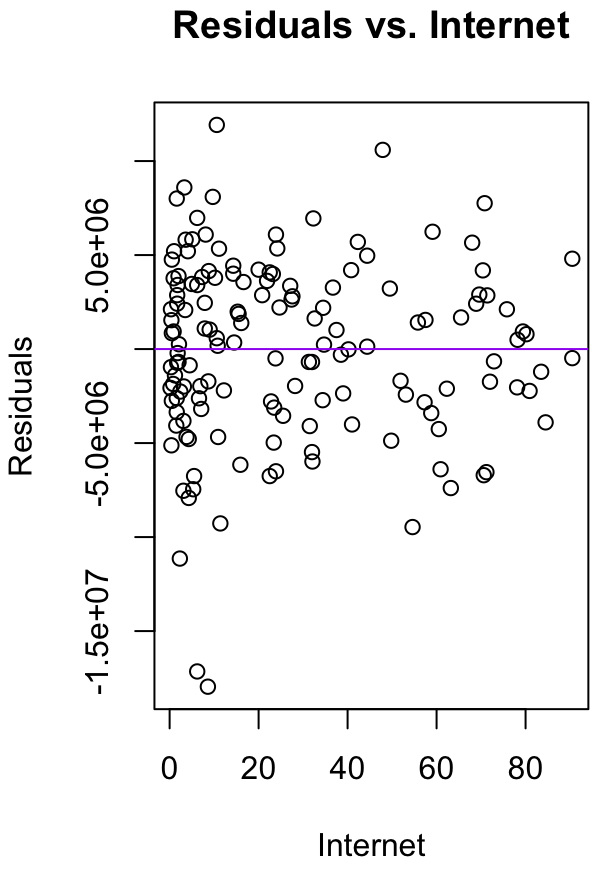
```
## [1] "p+1 = 6 (we are looking for the marlows cp closest to p+1"
```

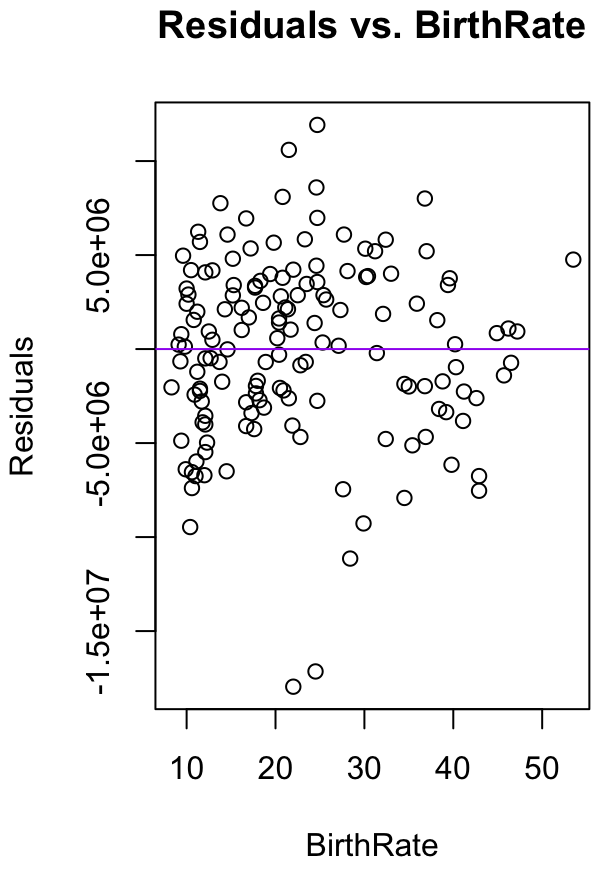
Residuals are centered at 0 with constant error, errors look normal, and our R^2 and adjusted R^2 are higher! this is all very impressive

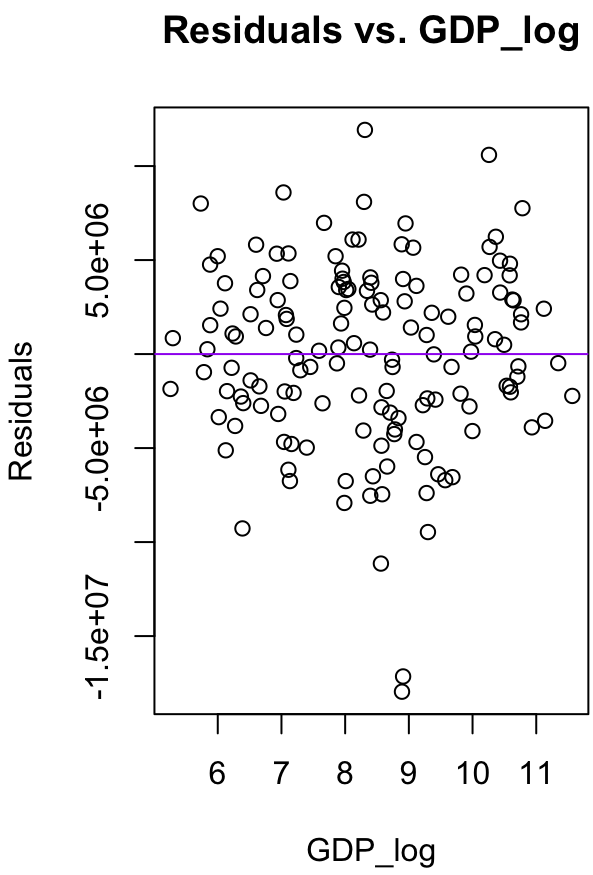
```
# one last check that our predictors are linear
par(mfrow = c(1, 2))
predictors = c("LandArea", "Health", "Internet", "BirthRate", "GDP_log", "Cell")

for (predictor in predictors) {
  data = countries[[predictor]]
  q = quantile(data, c(0.25, 0.75), na.rm = TRUE)
  iqr = q[2] - q[1]
  lower_limit = q[1] - 1.5 * iqr
  upper_limit = q[2] + 1.5 * iqr

  x_limit_min = max(c(min(data), lower_limit))
  x_limit_max = min(c(max(data), upper_limit))


  plot(countries[[predictor]], residuals(model6), main = paste("Residuals vs.", predicto
r),
       xlab = predictor, ylab = "Residuals", xlim = c(x_limit_min, x_limit_max))
    abline(h = 0, col = "purple")
    identify(countries[[predictor]], countries$LifeExpectancy_4, labels = rownames(count
ries))



}
```

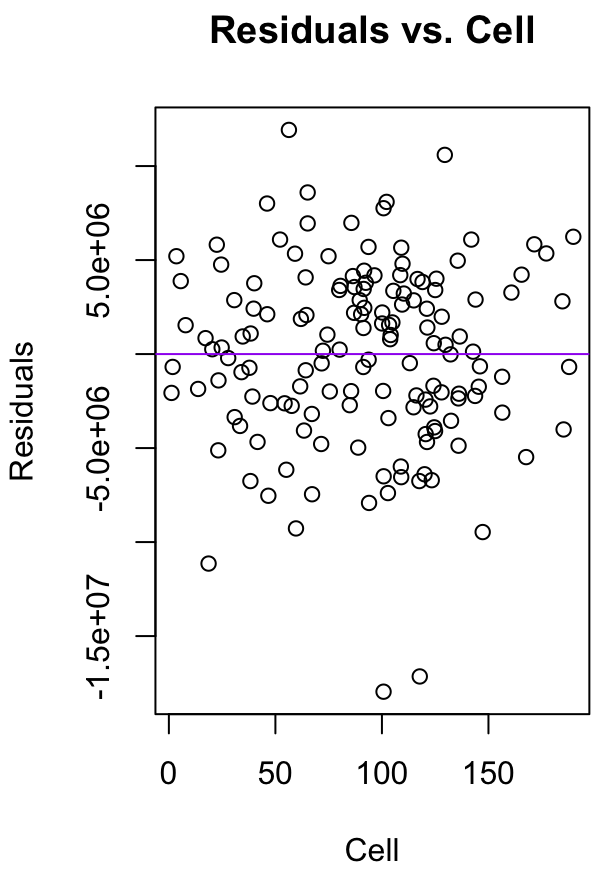

Residuals vs. LandArea
Residuals
5.0e+06
-5.0e+06
-1.5e+07
0
400000
1000000
LandArea


Residuals vs. Health
Residuals
5.0e+06
-5.0e+06
-1.5e+07
0
5
10
15
20
Health


Residuals vs. Internet
Residuals
5.0e+06
-5.0e+06
-1.5e+07
0
20
40
60
80
Internet

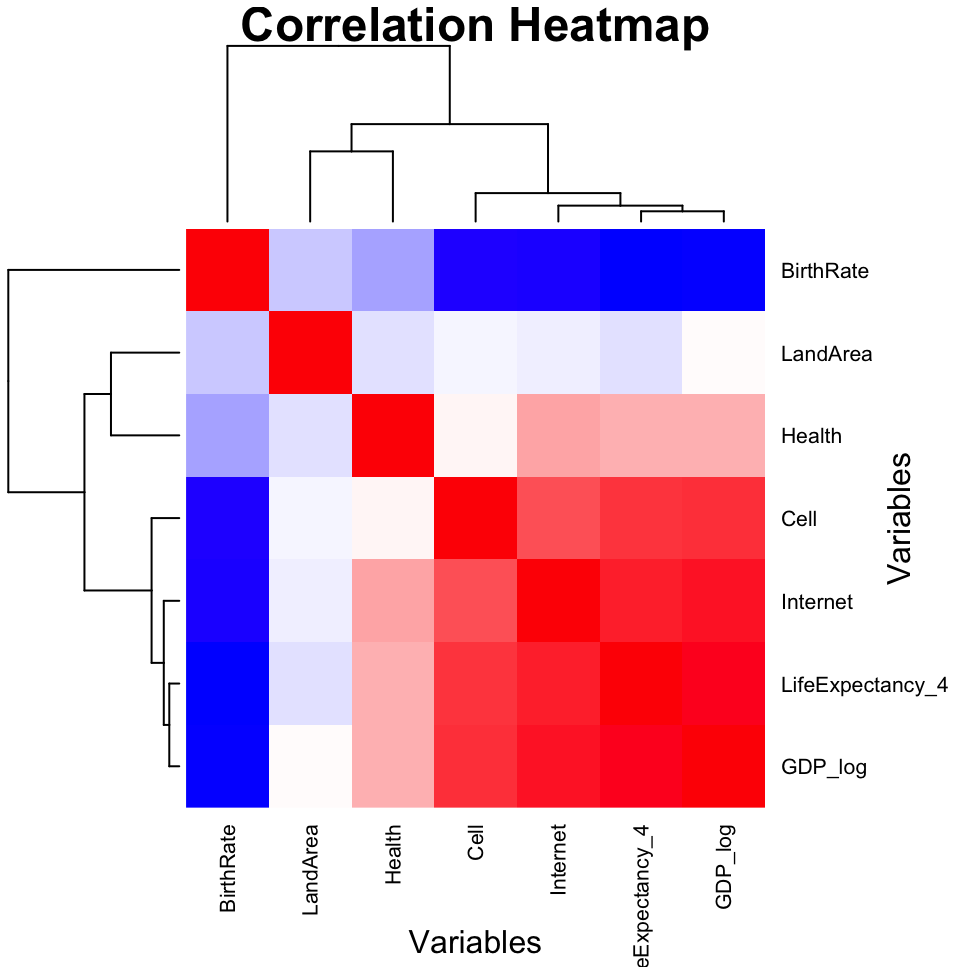

Residuals vs. BirthRate
Residuals
5.0e+06
-5.0e+06
-1.5e+07
10
20
30
40
50
BirthRate






```
par(mfrow = c(1, 1))
```

```
# we will check the colinearity of our final model one last time

countries_final_subset = countries[c("LifeExpectancy_4", "GDP_log", "Cell", "Internet",
"BirthRate", "LandArea", "Health")]
correlation_matrix = cor(countries_final_subset)


heatmap(
  correlation_matrix,
  main = "Correlation Heatmap",
  xlab = "Variables",
  ylab = "Variables",
  col = colorRampPalette(c("blue", "white", "red"))(100),
  symm = TRUE,
  scale = "none",
  margins = c(5, 5),
  cexRow = 0.8,
  cexCol = 0.8
)
```




```
correlation_matrix
```

```
##                 LifeExpectancy_4     GDP_log        Cell    Internet
## LifeExpectancy_4      1.000000000  0.86520682  0.70552999  0.79166776
## GDP_log               0.865206818  1.00000000  0.73323063  0.83077168
## Cell                  0.705529986  0.73323063  1.00000000  0.61359954
## Internet              0.791667756  0.83077168  0.61359954  1.00000000
## BirthRate            -0.844298170 -0.80547534 -0.68466732 -0.71516838
## LandArea             -0.004946161  0.08154779  0.04808815  0.03237619
## Health                0.313299535  0.30088291  0.11305851  0.34150818
##                   BirthRate     LandArea       Health
## LifeExpectancy_4 -0.84429817 -0.004946161  0.313299535
## GDP_log          -0.80547534  0.081547786  0.300882911
## Cell             -0.68466732  0.048088151  0.113058507
## Internet         -0.71516838  0.032376189  0.341508182
## BirthRate         1.00000000 -0.071323090 -0.183227720
## LandArea         -0.07132309  1.000000000  0.003159548
## Health           -0.18322772  0.003159548  1.000000000
```

```
vif(model6)
```

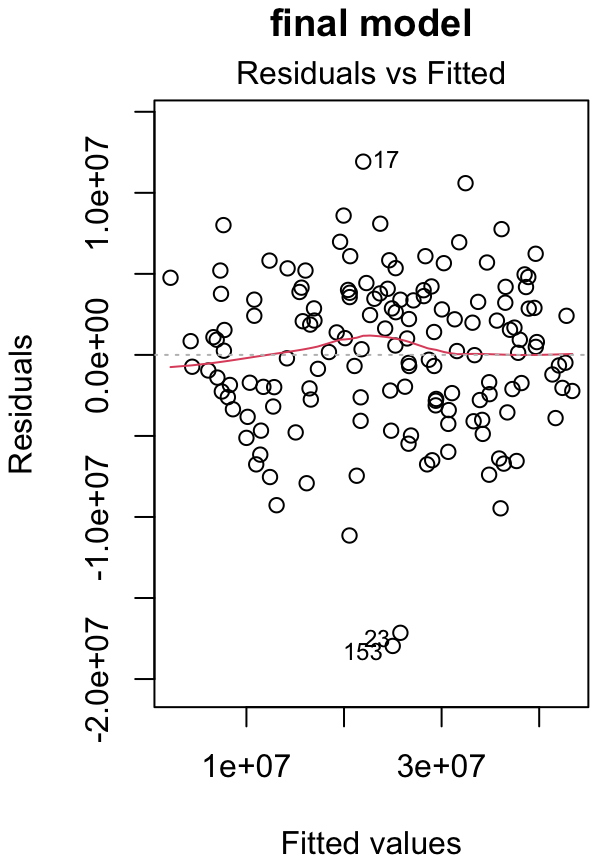
```
##  LandArea    Health  Internet BirthRate   GDP_log      Cell
##  1.011784  1.178212  3.428507  3.102221  5.380795  2.338750
```

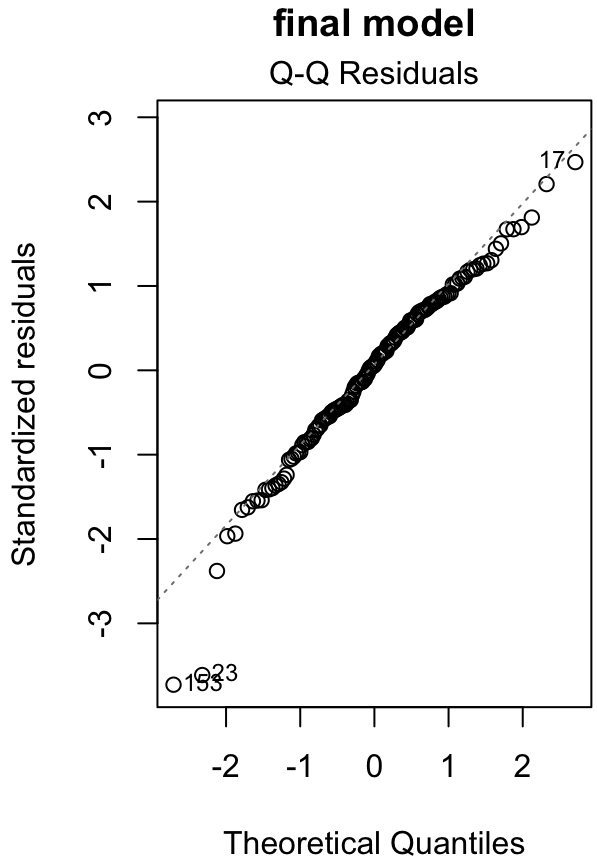
```
# technically a value greater than 5 is potentially concerning, but this value is very v
ery close to 5. Also, because this is our most important predictor, removing it would li
kely be very detremental to our model. It is also noted that even our highest value is s
till very close to 5, and under 5 is considered medium. Assuming that the model is used
mostly for predictive purposes, these vif values should not introduce bias into our pred
ictions. Other applications of the model may require these cmulticolinarity values to re
assess, but that is outside the scope of this analysis
```

```
print(min(countries$LifeExpectancy))
```

```
## [1] 43.9
```

```
par(mfrow = c(1, 2))
finalmodel = plot(model6, which = 1, main = "final model")
finalqq = plot(model6, which = 2, main = "final model")
```





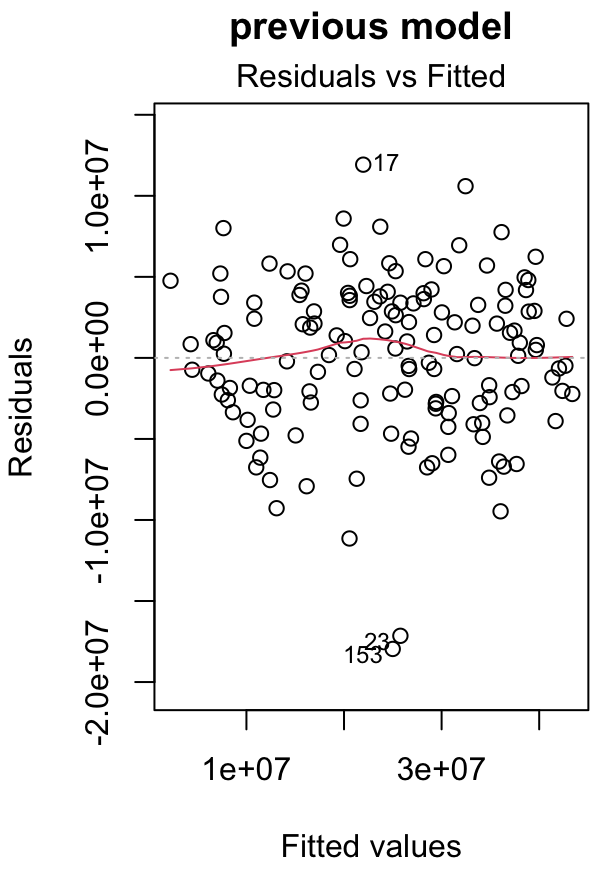
Note: above concludes the process of building the final model. The rest is experimentions into ways to improve

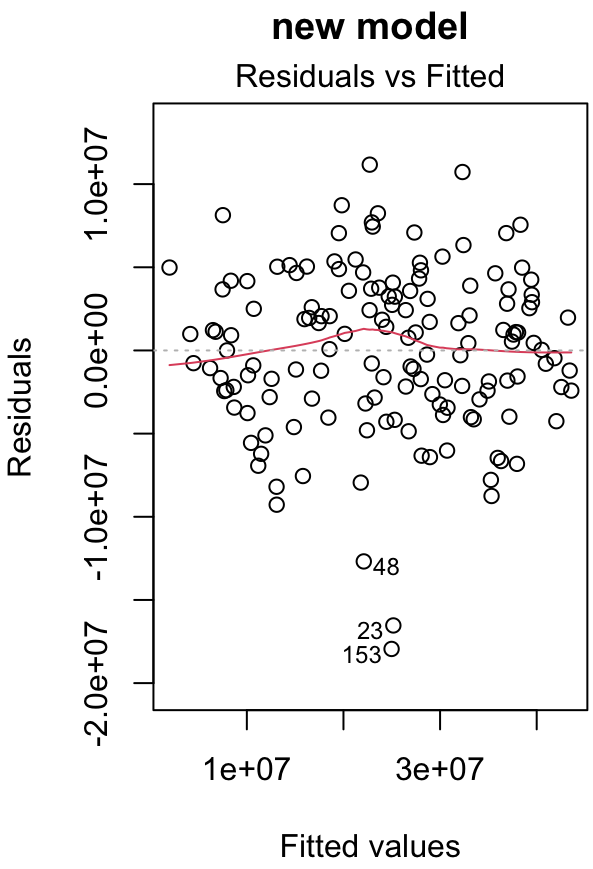
multicolinearity, none of which were determined to improve the model

```
# lets try removing cell


modelnocor = lm(LifeExpectancy_4 ~ LandArea + Health + GDP_log + Internet + BirthRate, d
ata = countries)

print(summary(modelnocor))
```

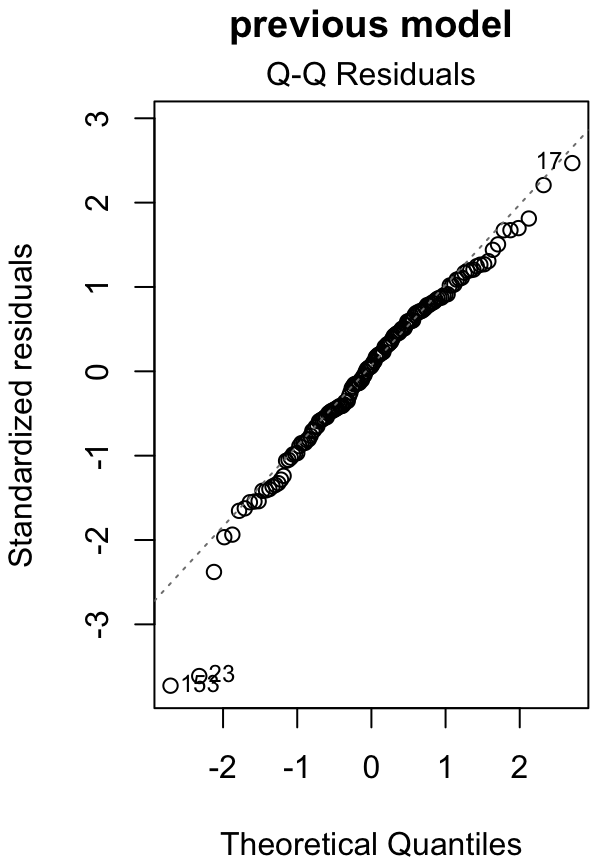
```
##
## Call:
## lm(formula = LifeExpectancy_4 ~ LandArea + Health + GDP_log +
##     Internet + BirthRate, data = countries)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -17945521  -2847800    446210   3577765  11169624
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.033e+06  5.364e+06   1.125   0.2626
## LandArea    -4.292e-01  2.088e-01  -2.056   0.0417 *
## Health       1.820e+05  9.854e+04   1.847   0.0668 .
## GDP_log      3.024e+06  5.701e+05   5.303 4.26e-07 ***
## Internet     6.481e+04  2.864e+04   2.263   0.0252 *
## BirthRate   -4.485e+05  6.471e+04  -6.932 1.34e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4922000 on 142 degrees of freedom
## Multiple R-squared:  0.8292, Adjusted R-squared:  0.8231
## F-statistic: 137.8 on 5 and 142 DF,  p-value: < 2.2e-16
```

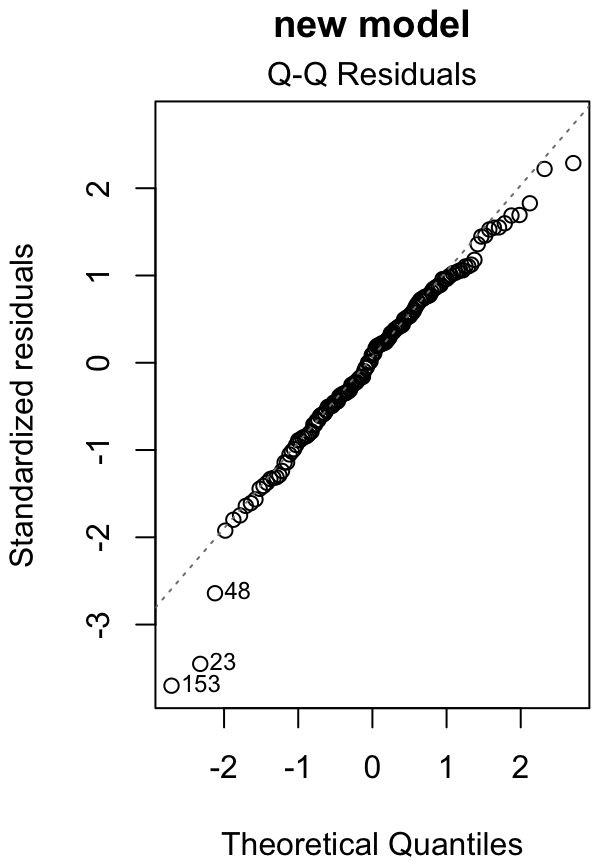
```
par(mfrow = c(1, 2))



resid2 = plot(model6, which = 1, main = "previous model")
resid3 = plot(modelnocor, which = 1, main = "new model")
```






```
qq2 = plot(model6, which = 2, main = "previous model")
qq3 = plot(modelnocor, which = 2, main = "new model")
```






```
vif(modelnocor)
```

```
##  LandArea    Health   GDP_log  Internet BirthRate
##  1.011244  1.153415  4.649537  3.428425  2.956083
```

```
model_summary = summary(modelnocor)

print("mallows CP:")
```

```
## [1] "mallows CP:"
```

```
ols_mallows_cp(modelnocor, fullmodel)
```

```
## [1] 6.25462
```

```
print("p+1 = 5 (we are looking for the marlows cp closest to p+1")
```

```
## [1] "p+1 = 5 (we are looking for the marlows cp closest to p+1"
```

All the vif values are less than 5 now, but the errors are less normal now and less centered at 0. I don’t think this trade off is worth it. The R^2, adjusted R^2 and marlows CP value are also worse now. Thus I conclude that model6 is the best model