R Notebook

```
col_NA_counts =
  t(data_full[, lapply(.SD, purrr::compose(sum, is.na))]) %>%
  data.table(keep.rownames = "var") %>%
  setorderv("V1", order = -1) %>%
  with(setNames(split(var, group(V1)), paste0("#NA = ", unique(V1))))

data_NA.free = na.omit(data_full[, .SD, .SDcols = -unlist(col_NA_counts[1:2])])

head(col_NA_counts, 3)
```

```
## $`#NA = 71293`
## [1] "rest_stops"
##
## $`#NA = 60832`
## [1] "Mileage.M" "Mileage.I" "Mileage.U" "Mileage.S" "Mileage.O" "Mileage.C"
##
## $`#NA = 12036`
##  [1] "Restaurants and Other Eating Places"
##  [2] "Offices of Physicians"
##  [3] "Personal Care Services"
##  [4] "Religious Organizations"
##  [5] "Automotive Repair and Maintenance"
##  [6] "Offices of Dentists"
##  [7] "Other Amusement and Recreation Industries"
##  [8] "Offices of Other Health Practitioners"
##  [9] "Depository Credit Intermediation"
## [10] "Elementary and Secondary Schools"
## [11] "Agencies, Brokerages, and Other Insurance Related Activities"
## [12] "Child Day Care Services"
## [13] "Health and Personal Care Stores"
## [14] "Gasoline Stations"
## [15] "Grocery Stores"
## [16] "Building Material and Supplies Dealers"
## [17] "Sporting Goods, Hobby, and Musical Instrument Stores"
## [18] "Clothing Stores"
## [19] "Automotive Parts, Accessories, and Tire Stores"
## [20] "Museums, Historical Sites, and Similar Institutions"
## [21] "Traveler Accommodation"
## [22] "Other Miscellaneous Store Retailers"
## [23] "Automobile Dealers"
## [24] "General Merchandise Stores, including Warehouse Clubs and Supercenters"
## [25] "Other Financial Investment Activities"
## [26] "Home Health Care Services"
## [27] "Other Professional, Scientific, and Technical Services"
## [28] "Lessors of Real Estate"
## [29] "Accounting, Tax Preparation, Bookkeeping, and Payroll Services"
## [30] "Offices of Real Estate Agents and Brokers"
## [31] "Wired and Wireless Telecommunications Carriers"
## [32] "Used Merchandise Stores"
## [33] "Furniture Stores"
## [34] "Office Supplies, Stationery, and Gift Stores"
## [35] "Other Personal Services"
## [36] "Justice, Public Order, and Safety Activities"
## [37] "Beer, Wine, and Liquor Stores"
## [38] "Electronics and Appliance Stores"
## [39] "Services to Buildings and Dwellings"
## [40] "Jewelry, Luggage, and Leather Goods Stores"
## [41] "Postal Service"
## [42] "Florists"
## [43] "Specialty Food Stores"
## [44] "Other Motor Vehicle Dealers"
## [45] "Home Furnishings Stores"
```

```
## [46] "Outpatient Care Centers"
## [47] "Personal and Household Goods Repair and Maintenance"
## [48] "Colleges, Universities, and Professional Schools"
## [49] "Medical and Diagnostic Laboratories"
## [50] "Other Schools and Instruction"
```

```
cat("nrow full merged dataset ", nrow(data_full))
```

```
## nrow full merged dataset  72734
```

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

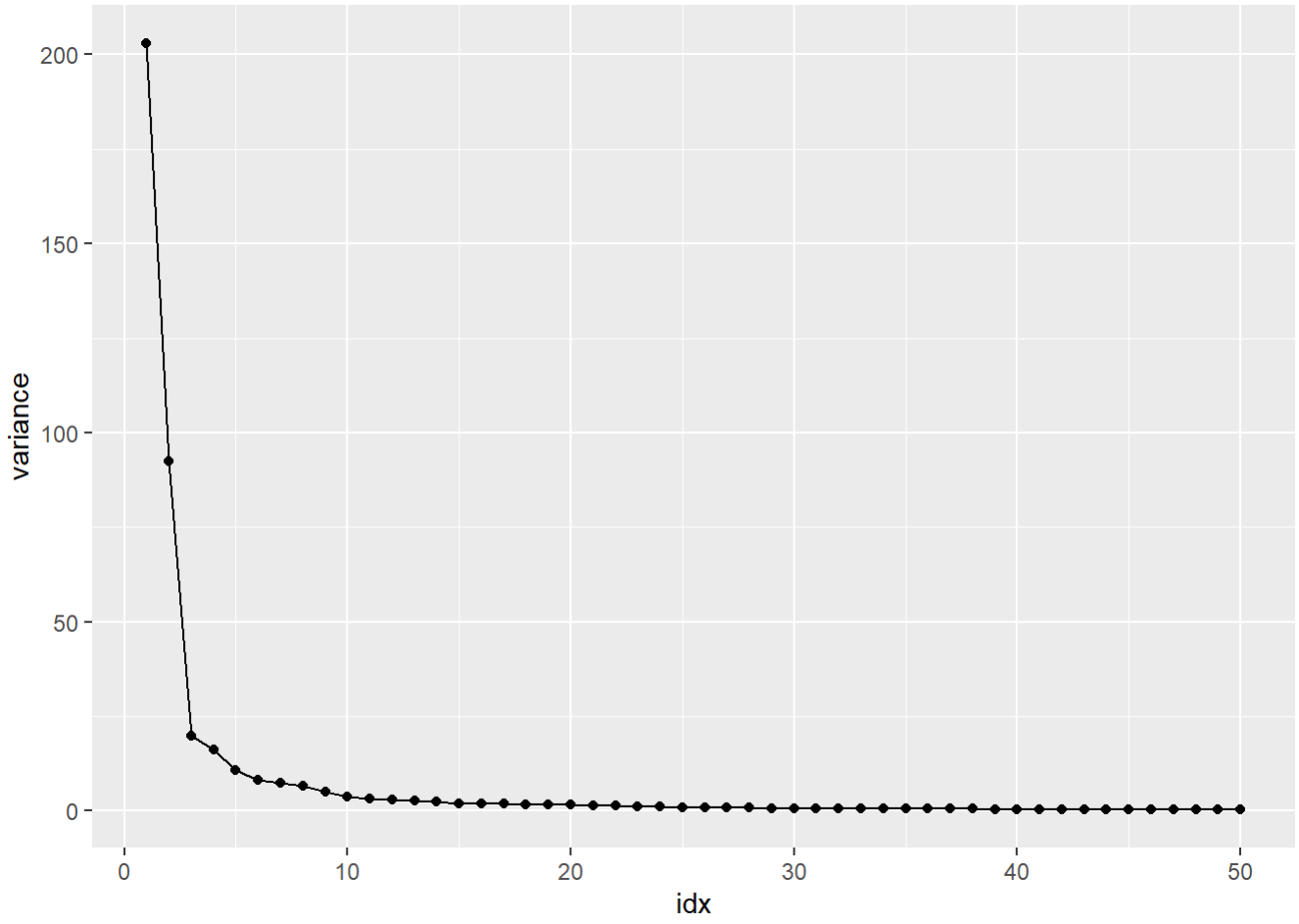
```
cat("\nnrow NA.free dataset ", nrow(data_NA.free))
```

```
##
## nrow NA.free dataset  59251
```

```
data_POI_PCA =
  copy(data_NA.free) |>
  subset(select = col_NA_counts[[3]]) |>
  prcomp()

setDT(data_POI_PCA["sdev"])[,c("idx", "variance") := .(.I, sdev^2)] %>%
    qplot(x = idx, y = variance, geom = c("line", "point"), data = .)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```




```
cat("By keeping the POI data's first", which.min(c(summary(data_POI_PCA)[[6]][3,]) < 0.9), "Prin
cipal Components we retain over 90% of the entire dataset's variance")
```

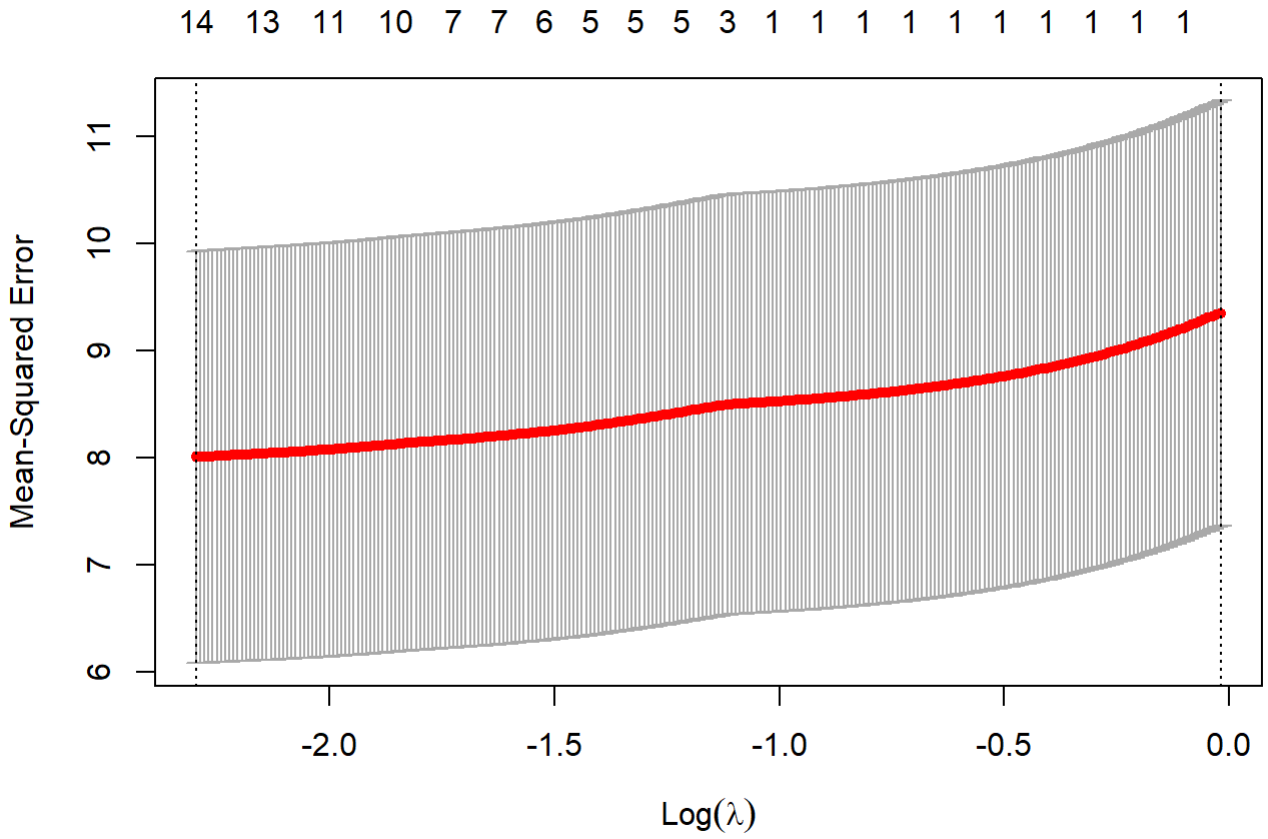
```
## By keeping the POI data's first 10 Principal Components we retain over 90% of the entire data
set's variance
```

```
data_POI_PCA = cbind(copy(data_NA.free)[, .SD, .SDcols = -col_NA_counts[[3]]], data_POI_PCA$x[,
1:10])

good_data_x = copy(data_POI_PCA)[, !data_POI_PCA[, names(data_POI_PCA) %like% "EV|N.Stations"
                                                 | sapply(.SD, is.character)], with = FALSE]
good_data_y = data_POI_PCA$N.Stations

cv_tune.lasso_model = suppressMessages(suppressWarnings(
  cv.glmnet(x = data.matrix(good_data_x),
            y = good_data_y,
            nlambda = 1000,
            nfolds = 500,
            pmax = 15,
            parallel = TRUE)))


plot(cv_tune.lasso_model)
```




```
lasso_model = glmnet(x = good_data_x, y = good_data_y, lambda =  cv_tune.lasso_model$lambda.min)
coef(lasso_model)
```

```
## 107 x 1 sparse Matrix of class "dgCMatrix"
##                                       s0
## (Intercept)               4.630099e-01
## GEOID                     .
## STATEFP                   .
## COUNTYFP                  .
## TRACTCE                   .
## incentives                1.784075e-04
## laws and regulations      1.160540e-02
## avg_gasprice_2012         .
## avg_gasprice_2013         .
## avg_gasprice_2014         .
## avg_gasprice_2015         .
## avg_gasprice_2016         .
## avg_gasprice_2017         .
## avg_gasprice_2018         .
## avg_gasprice_2019         .
## avg_gasprice_2020         .
## avg_gasprice_2021         .
## avg_gasprice_2022         .
## 2016_dem_proportion       .
## 2020_dem_proportion       .
## 2019_affectweather        .
## 2019_citizens             .
## 2019_fundrenewables       .
## 2019_rebates              .
## 2019_supportRPS           .
## 2020_affectweather        .
## 2020_citizens             .
## 2020_fundrenewables       .
## 2020_rebates              .
## 2020_supportRPS           .
## 2021_affectweather        .
## 2021_citizens             .
## 2021_fundrenewables       .
## 2021_rebates              .
## 2021_supportRPS           .
## NAME                      .
## ALAND                     .
## AWATER                    .
## meters                    .
## miles                     .
## tract                     .
## pop                       .
## male                      .
## female                    .
## age                       .
## male.age                  .
## female.age                .
## white                     .
## black                     .
## indian.alaskan            .
```

```
## asian                     1.258567e-04
## pacific                   8.076338e-04
## other                     .
## two.or.more               .
## white.not.hisp            .
## hisp                      .
## white.hisp                .
## black.hisp                .
## households                .
## i10orless                 .
## i10to14                   .
## i15to19                   .
## i20to24                   .
## i25to29                   .
## i30to34                   .
## i35to39                   .
## i40to44                   .
## i45to49                   .
## i50to59                   .
## i60to74                   .
## i75to99                   .
## i100to124                 .
## i125to149                 .
## i150to199                 .
## i200ormore                .
## hh.income                 .
## house.value               3.175085e-08
## male.p                    .
## female.p                  .
## white.p                   .
## black.p                   .
## asian.p                   .
## hisp.p                    .
## white.not.hisp.p          .
## white.hisp.p              .
## black.hisp.p              .
## other.p                   .
## rescaled.house.value      .
## hh.income.and.house       .
## tot.hh.income             .
## tot.house.value           4.773651e-11
## tot.hh.income.and.house   .
## pop.density              -1.017751e-06
## hh.density                .
## income.density            .
## house.value.density       .
## house.and.income.density  .
## PC1                       5.972147e-02
## PC2                       2.276431e-02
## PC3                      -3.879226e-03
## PC4                      -1.774374e-02
## PC5                       2.025201e-02
```

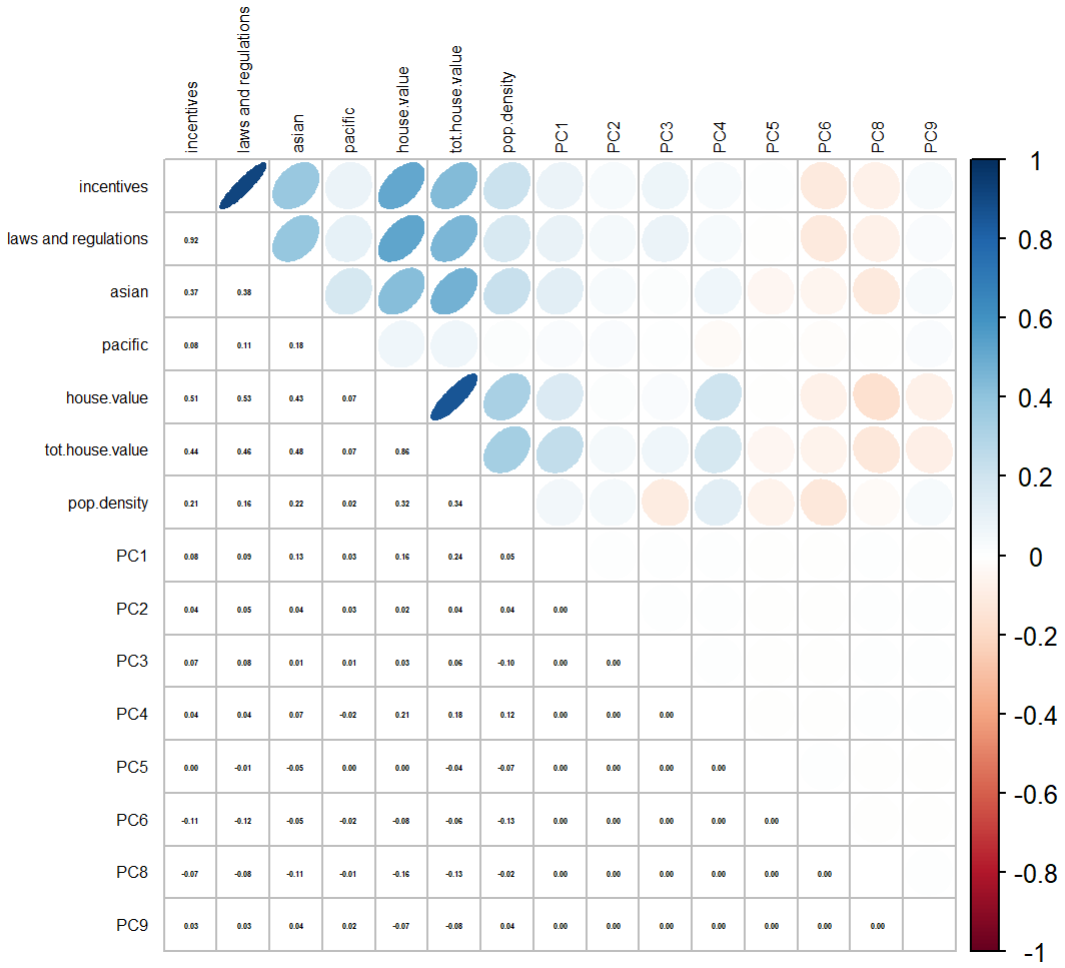
```
## PC6                      2.768476e-02
## PC7                      .
## PC8                     -7.147830e-02
## PC9                     -1.034405e-01
## PC10                     .
```

```
vars_keep = rownames(coef(lasso_model))[which(as.matrix(coef(lasso_model)) != 0)][-1]

data_keep = copy(data_POI_PCA)[, ..vars_keep]
head(data_keep)
```

```
##    incentives laws and regulations asian pacific house.value tot.house.value
## 1:          6                    5     0       0      113800        96843800
## 2:          6                    5    27       0       73200        66246000
## 3:          6                    5   202       0      150000       271200000
## 4:          6                    5   159       0      115900       138384600
## 5:          6                    5    93       0      170100       309752100
## 6:          6                    5    24       0      624000       891072000
##    pop.density       PC1         PC2        PC3         PC4         PC5
## 1:    28.07754 -9.403059  -1.8071066 -0.6430269  0.2903328  0.41308390
## 2:  3694.73212 -6.956766  -0.7302009  2.6760786  3.2546229  1.09361943
## 3:  1941.46112  5.112207   5.8014093  8.1883866 -2.3647980 -0.08114553
## 4:  3434.30165 -8.269829  -1.7368756  1.7945799  1.3176952  2.04098765
## 5:  1797.12235  4.082185   8.7304709 -5.1970221 -3.1047612 -2.84671442
## 6:  3162.35238 83.862734 -48.6152724 -1.8869280 -1.0987838 -0.94048102
##           PC6       PC8         PC9
## 1:  0.2166986 0.4398403  -0.2083111
## 2: -1.8059097 2.8836054  -0.5504837
## 3: -1.9218956 0.8868957   1.3497244
## 4:  2.8974549 1.2135671   0.9080462
## 5:  0.9058239 1.6232189   0.1564409
## 6:  9.6390092 2.3642791 -13.7833699
```

```
corrplot.mixed(
  cor(data_keep),
  lower.col = "black",
  upper = "ellipse",
  number.cex = .25,
  tl.col = "black",
  tl.pos = "lt",
  tl.cex = .5
)
```

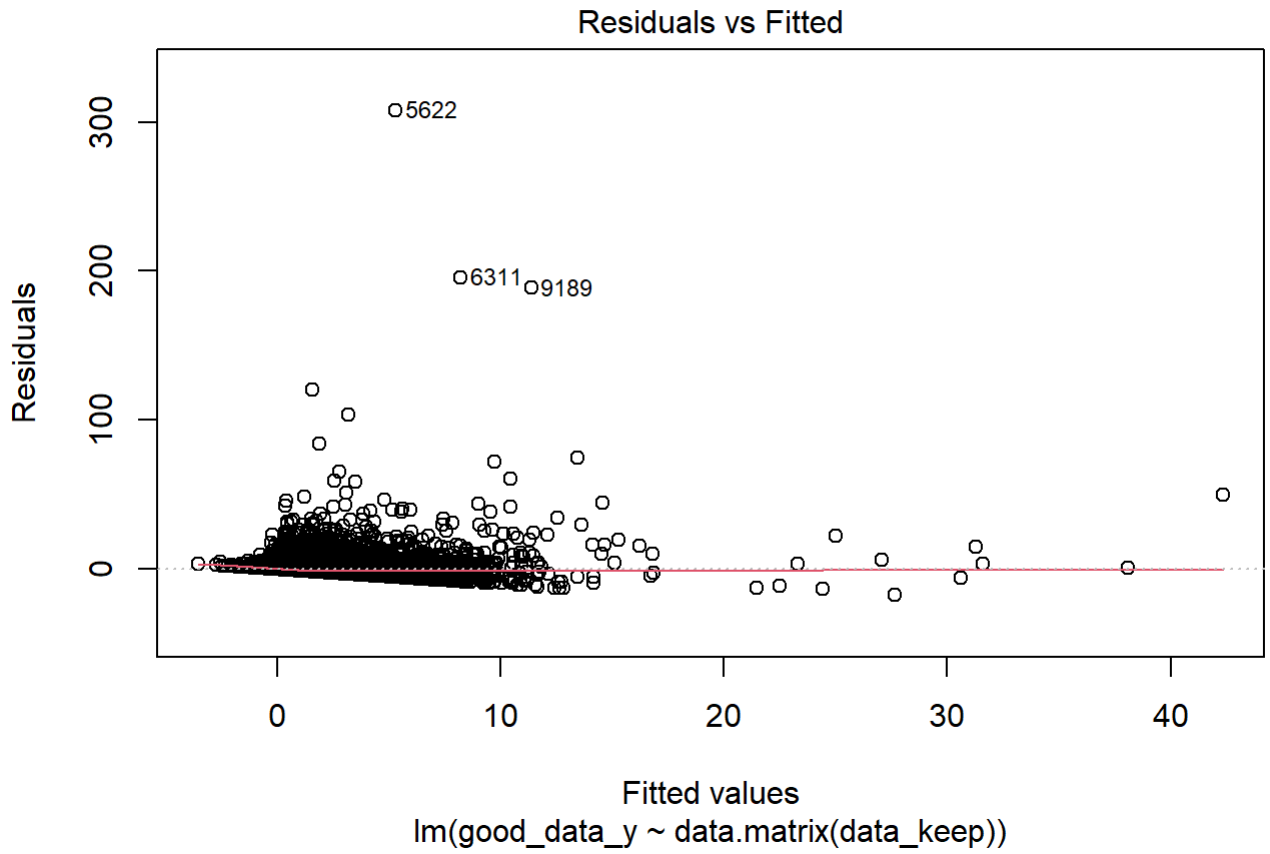


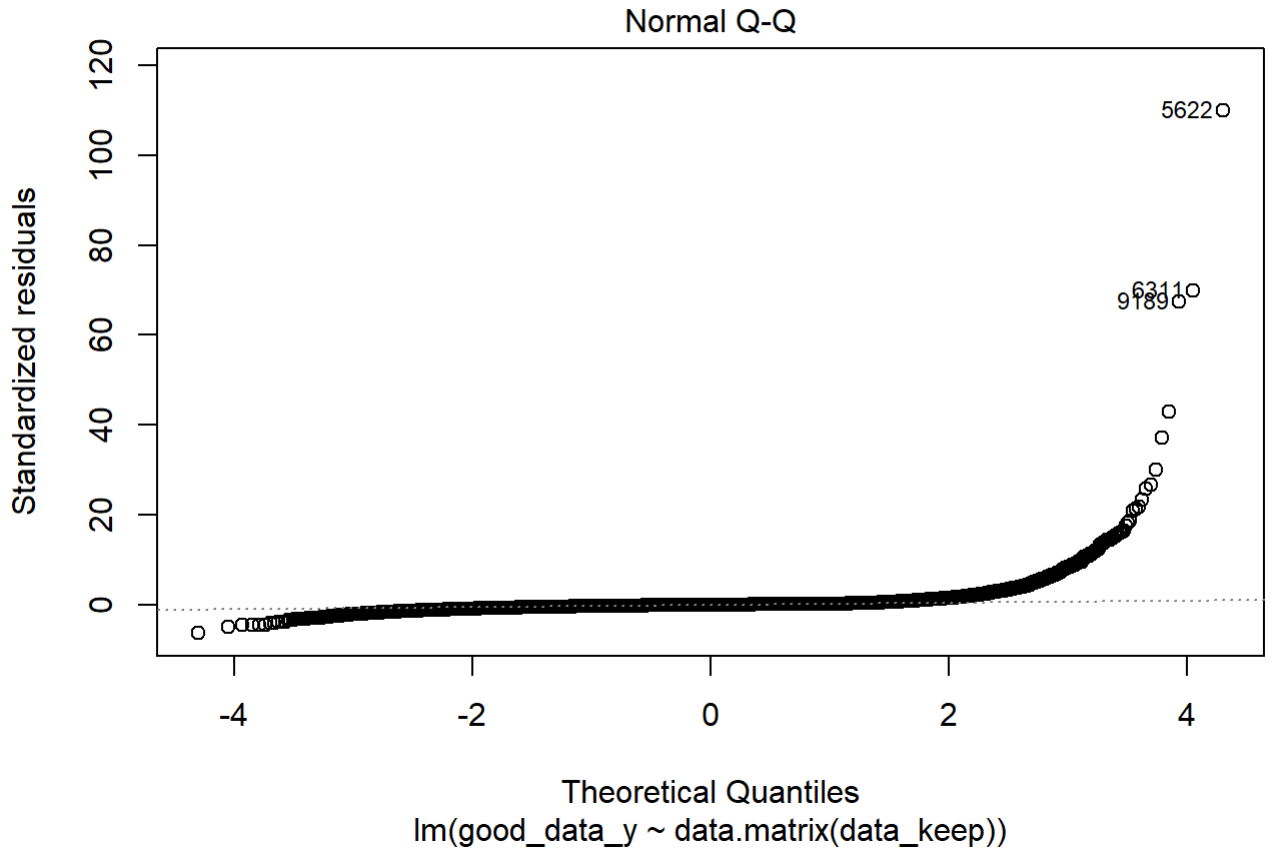

```
GGally::ggpairs(dplyr::sample_n(data_keep, 100)) +
  theme(
    plot.title = element_text(face = 'bold',
                              size = 10,
                              hjust = 0.5, margin = margin(b = 20)),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

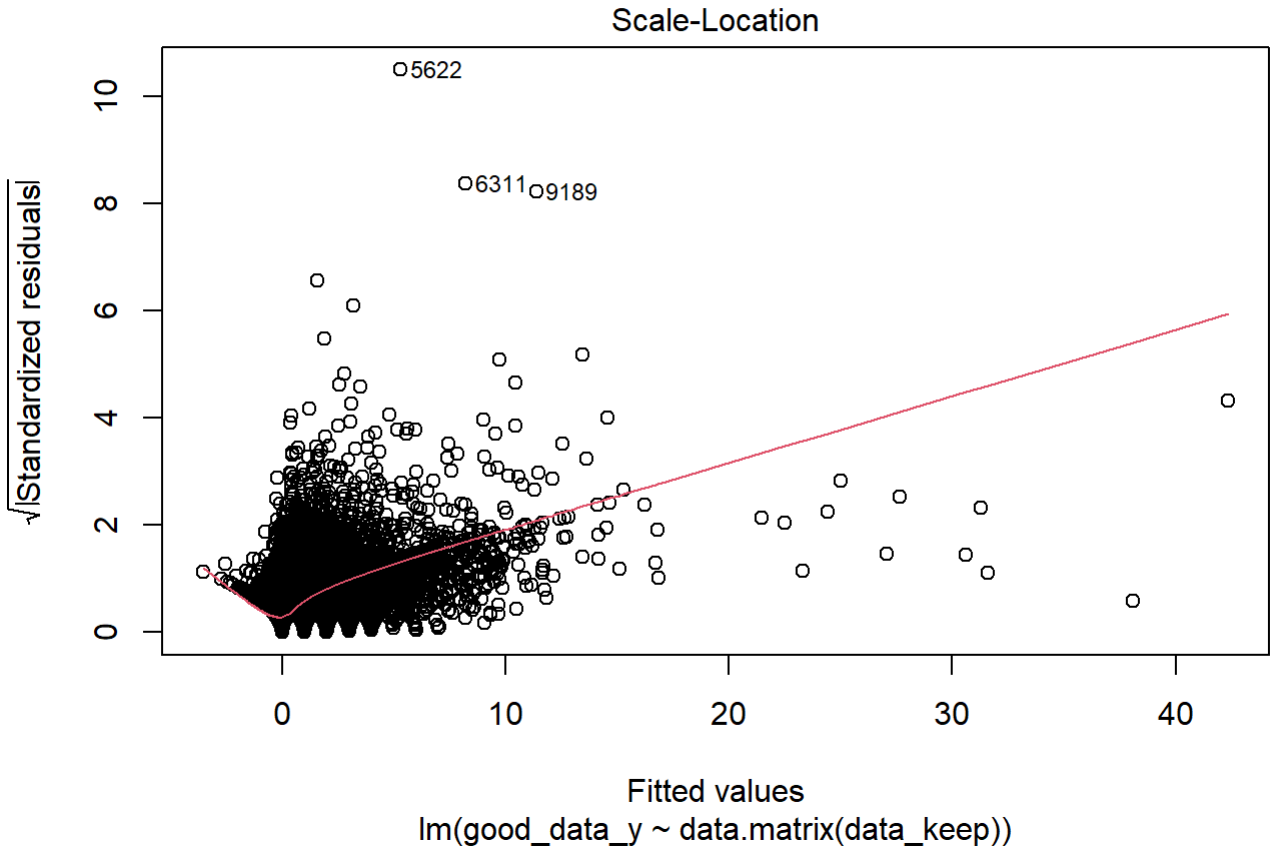

```
lm_model = lm(good_data_y ~ data.matrix(data_keep))
summary(lm_model)
```

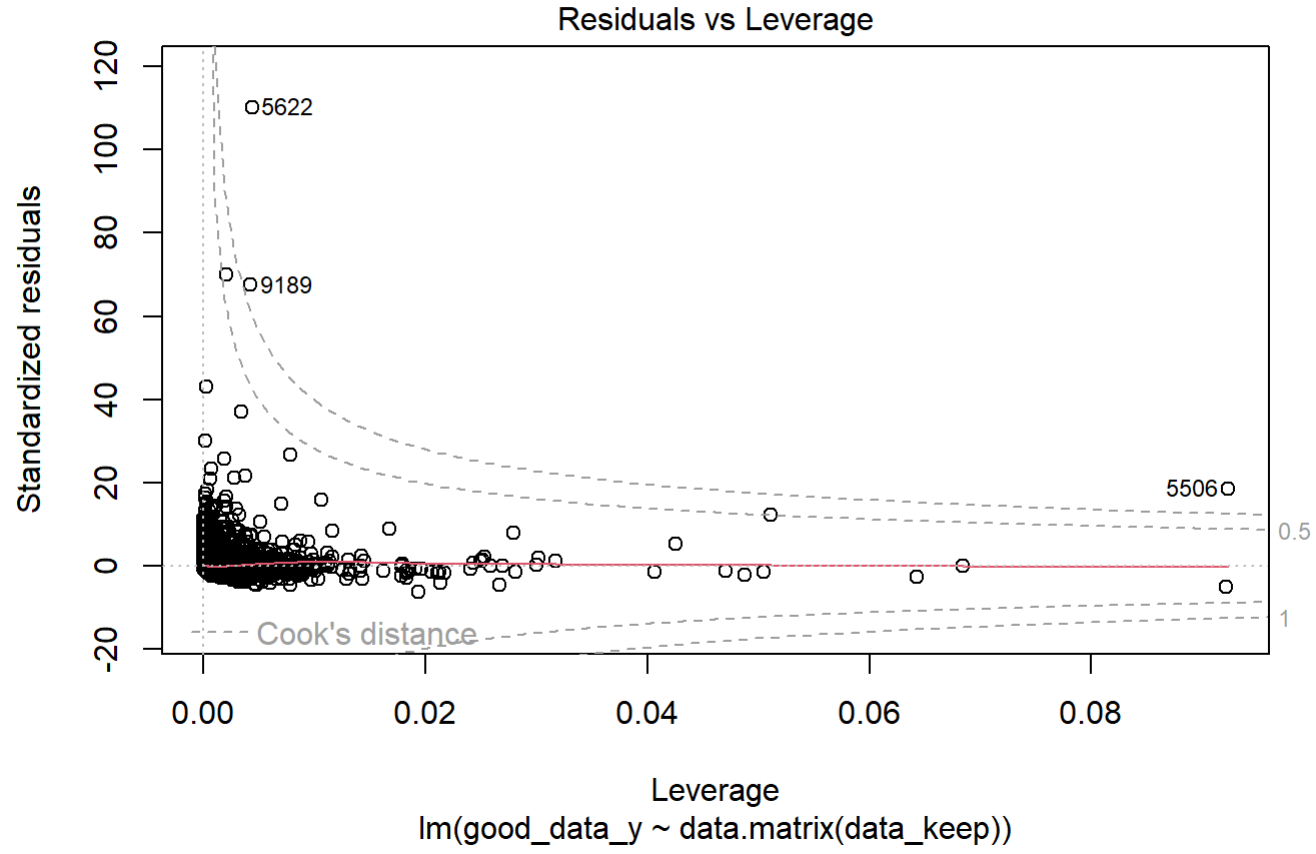
```
##
## Call:
## lm(formula = good_data_y ~ data.matrix(data_keep))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.671  -0.671  -0.130   0.221 307.691
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                              3.421e-01  1.841e-02  18.584
## data.matrix(data_keep)incentives         4.246e-03  1.162e-03   3.654
## data.matrix(data_keep)laws and regulations  9.497e-03  2.051e-03   4.631
## data.matrix(data_keep)asian              2.392e-04  3.069e-05   7.796
## data.matrix(data_keep)pacific            2.796e-03  2.884e-04   9.695
## data.matrix(data_keep)house.value        1.715e-07  9.607e-08   1.785
## data.matrix(data_keep)tot.house.value    1.706e-10  5.220e-11   3.268
## data.matrix(data_keep)pop.density       -1.466e-05  1.169e-06 -12.542
## data.matrix(data_keep)PC1                6.494e-02  8.392e-04  77.383
## data.matrix(data_keep)PC2                3.289e-02  1.201e-03  27.391
## data.matrix(data_keep)PC3               -3.236e-02  2.621e-03 -12.348
## data.matrix(data_keep)PC4               -4.358e-02  2.944e-03 -14.803
## data.matrix(data_keep)PC5                4.951e-02  3.534e-03  14.008
## data.matrix(data_keep)PC6                6.277e-02  4.069e-03  15.425
## data.matrix(data_keep)PC8               -1.025e-01  4.547e-03 -22.540
## data.matrix(data_keep)PC9               -1.456e-01  5.221e-03 -27.884
##                                         Pr(>|t|)
## (Intercept)                              < 2e-16 ***
## data.matrix(data_keep)incentives        0.000258 ***
## data.matrix(data_keep)laws and regulations 3.64e-06 ***
## data.matrix(data_keep)asian             6.51e-15 ***
## data.matrix(data_keep)pacific            < 2e-16 ***
## data.matrix(data_keep)house.value       0.074237 .
## data.matrix(data_keep)tot.house.value   0.001084 **
## data.matrix(data_keep)pop.density        < 2e-16 ***
## data.matrix(data_keep)PC1                < 2e-16 ***
## data.matrix(data_keep)PC2                < 2e-16 ***
## data.matrix(data_keep)PC3                < 2e-16 ***
## data.matrix(data_keep)PC4                < 2e-16 ***
## data.matrix(data_keep)PC5                < 2e-16 ***
## data.matrix(data_keep)PC6                < 2e-16 ***
## data.matrix(data_keep)PC8                < 2e-16 ***
## data.matrix(data_keep)PC9                < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.8 on 59235 degrees of freedom
## Multiple R-squared:  0.1617, Adjusted R-squared:  0.1614
## F-statistic: 761.4 on 15 and 59235 DF,  p-value: < 2.2e-16
```

```
plot(lm_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(good_data_y ~ data.matrix(data_keep))
5622
6311
9189
0
100
200
300
0
10
20
30
40


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(good_data_y ~ data.matrix(data_keep))
5622
6311
9189
0
20
40
60
80
100
120
-4
-2
0
2
4


Scale-Location
√|Standardized residuals|
5622
6311
9189
0
2
4
6
8
10
0
10
20
30
40
Fitted values
lm(good_data_y ~ data.matrix(data_keep))


Residuals vs Leverage
Standardized residuals
5622
9189
5506
0.5
1
Cook's distance
-20
0
20
40
60
80
100
120
0.00
0.02
0.04
0.06
0.08
Leverage
lm(good_data_y ~ data.matrix(data_keep))

```
logit_model = glm(as.logical(good_data_y) ~ data.matrix(data_keep), family = binomial(link = "lo
git"))
```

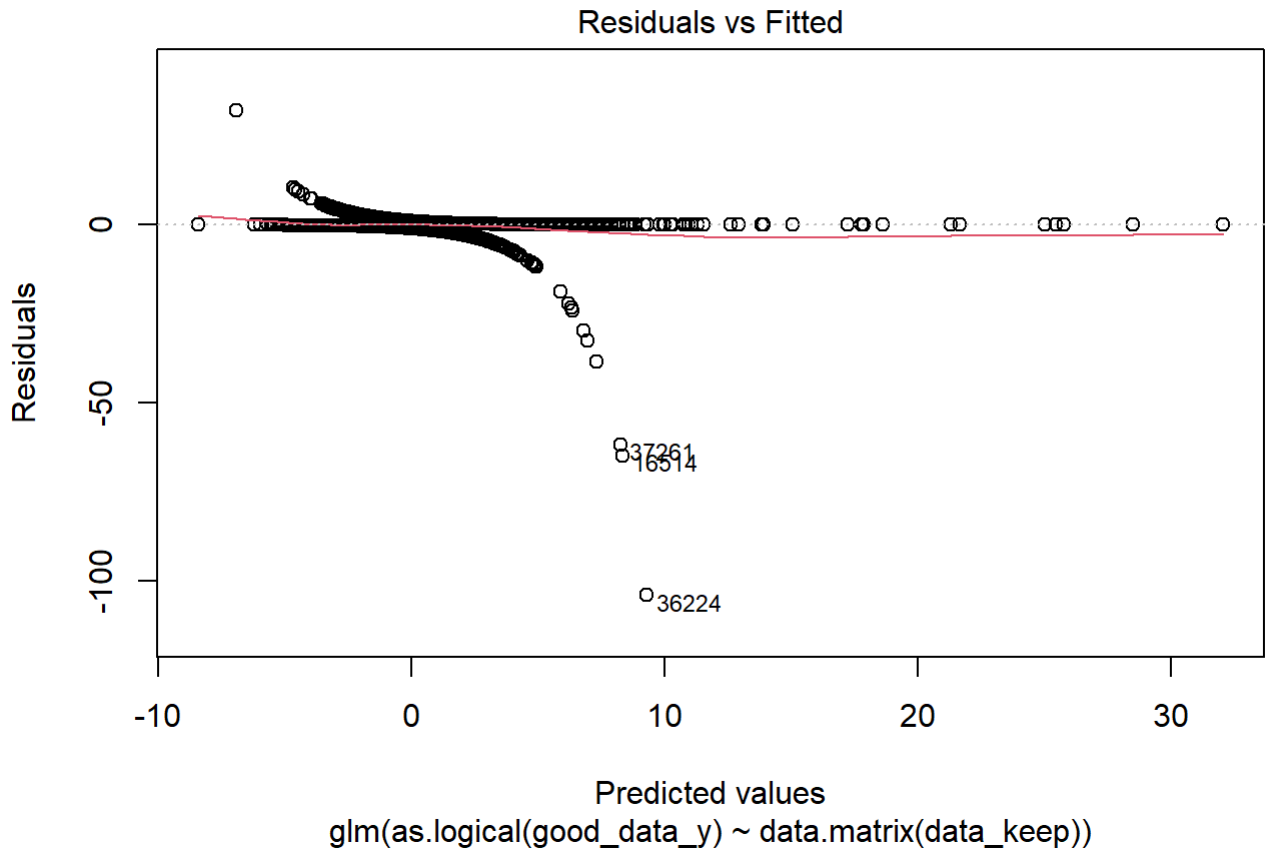
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

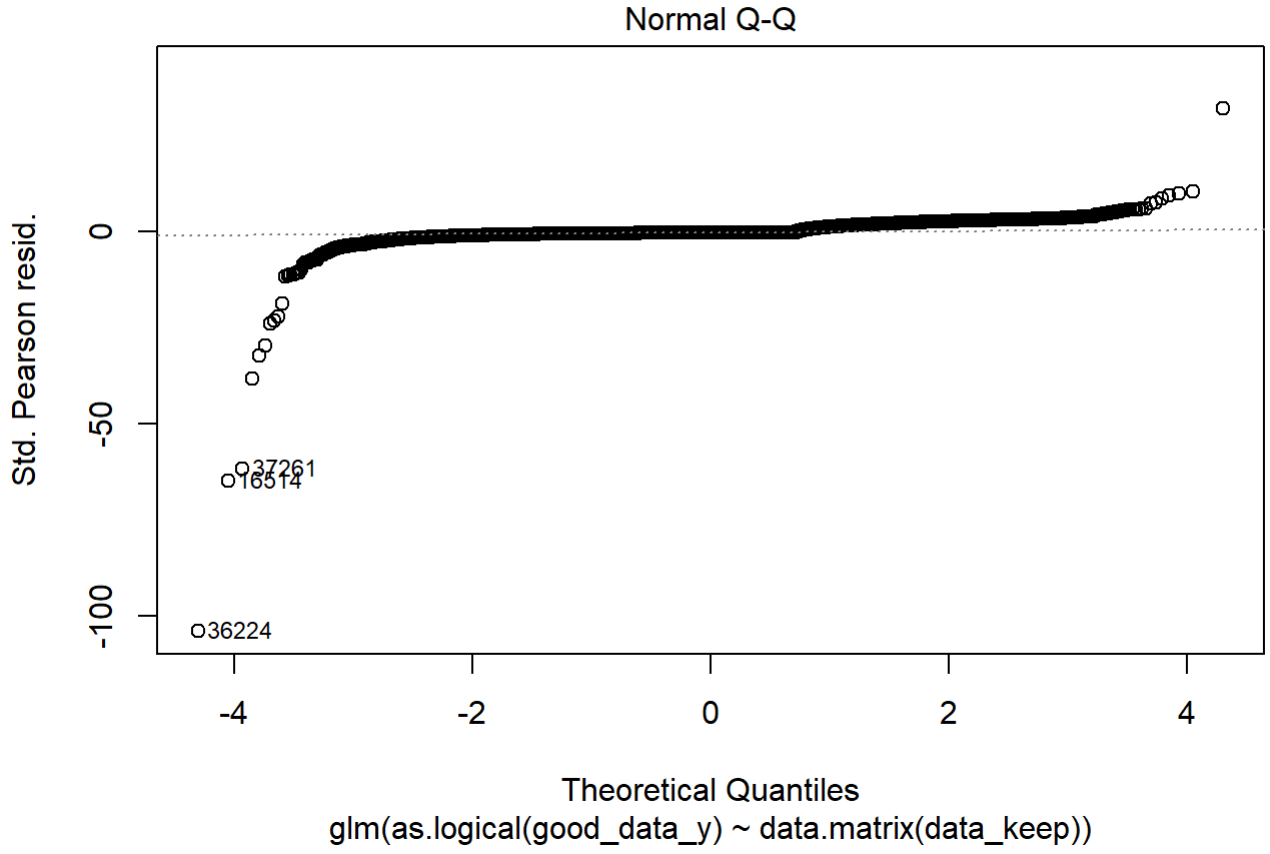
```
summary(logit_model)
```

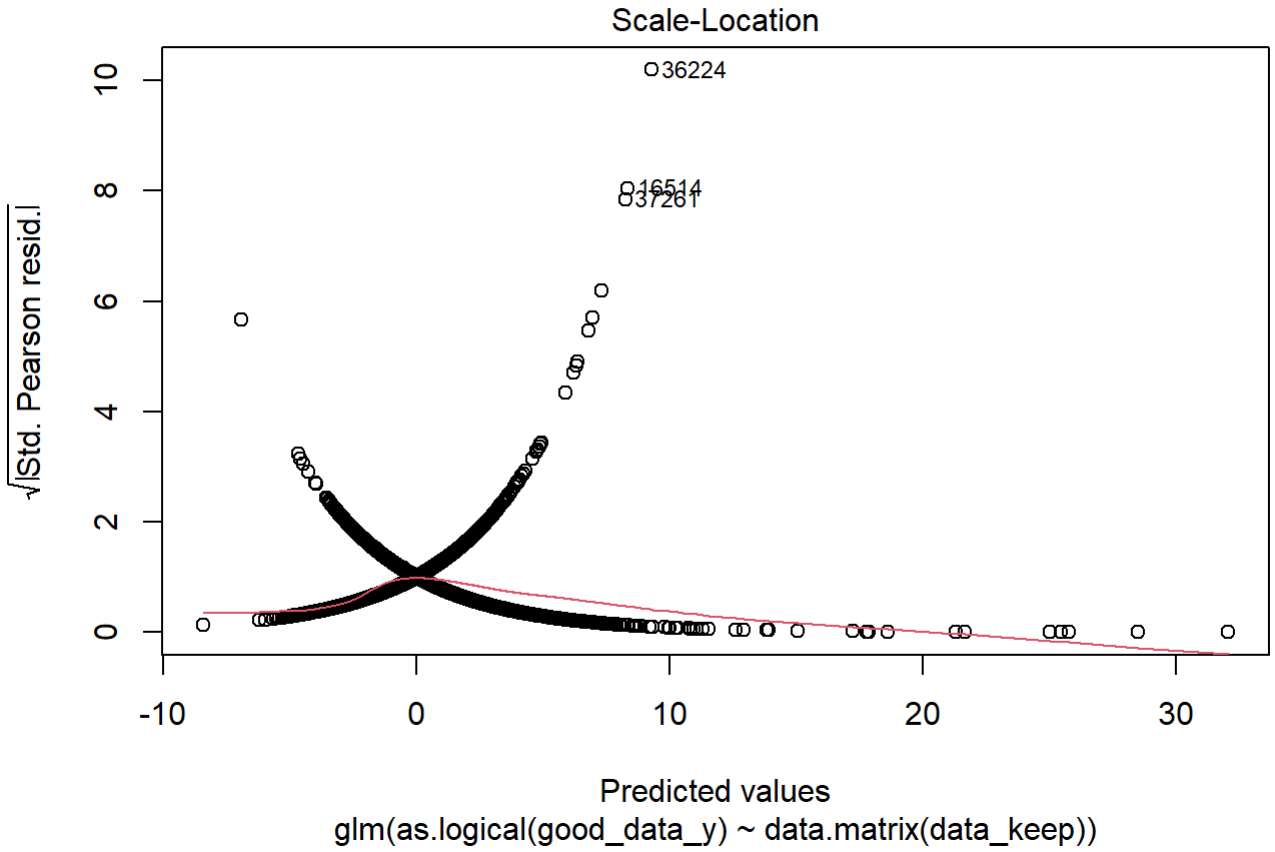
```
##
## Call:
## glm(formula = as.logical(good_data_y) ~ data.matrix(data_keep),
##     family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3101  -0.6504  -0.5121  -0.3886   3.7241
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             -1.675e+00  1.772e-02 -94.561
## data.matrix(data_keep)incentives        -5.653e-03  1.046e-03  -5.402
## data.matrix(data_keep)laws and regulations  2.218e-02  1.875e-03  11.832
## data.matrix(data_keep)asian              4.469e-05  2.652e-05   1.685
## data.matrix(data_keep)pacific           -2.214e-05  2.608e-04  -0.085
## data.matrix(data_keep)house.value        3.058e-07  8.486e-08   3.603
## data.matrix(data_keep)tot.house.value    4.307e-10  4.734e-11   9.098
## data.matrix(data_keep)pop.density       -2.515e-05  1.493e-06 -16.851
## data.matrix(data_keep)PC1                6.339e-02  9.464e-04  66.982
## data.matrix(data_keep)PC2                3.303e-02  1.204e-03  27.441
## data.matrix(data_keep)PC3               -1.651e-02  2.679e-03  -6.162
## data.matrix(data_keep)PC4               -4.055e-02  2.853e-03 -14.210
## data.matrix(data_keep)PC5                2.211e-02  3.889e-03   5.685
## data.matrix(data_keep)PC6                4.255e-02  3.832e-03  11.105
## data.matrix(data_keep)PC8               -5.184e-02  4.354e-03 -11.905
## data.matrix(data_keep)PC9               -7.145e-02  5.041e-03 -14.176
##                                         Pr(>|z|)
## (Intercept)                              < 2e-16 ***
## data.matrix(data_keep)incentives        6.58e-08 ***
## data.matrix(data_keep)laws and regulations  < 2e-16 ***
## data.matrix(data_keep)asian             0.091955 .
## data.matrix(data_keep)pacific           0.932340
## data.matrix(data_keep)house.value       0.000314 ***
## data.matrix(data_keep)tot.house.value    < 2e-16 ***
## data.matrix(data_keep)pop.density        < 2e-16 ***
## data.matrix(data_keep)PC1                < 2e-16 ***
## data.matrix(data_keep)PC2                < 2e-16 ***
## data.matrix(data_keep)PC3               7.17e-10 ***
## data.matrix(data_keep)PC4                < 2e-16 ***
## data.matrix(data_keep)PC5               1.31e-08 ***
## data.matrix(data_keep)PC6                < 2e-16 ***
## data.matrix(data_keep)PC8                < 2e-16 ***
## data.matrix(data_keep)PC9                < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64063  on 59250  degrees of freedom
## Residual deviance: 54230  on 59235  degrees of freedom
## AIC: 54262
```

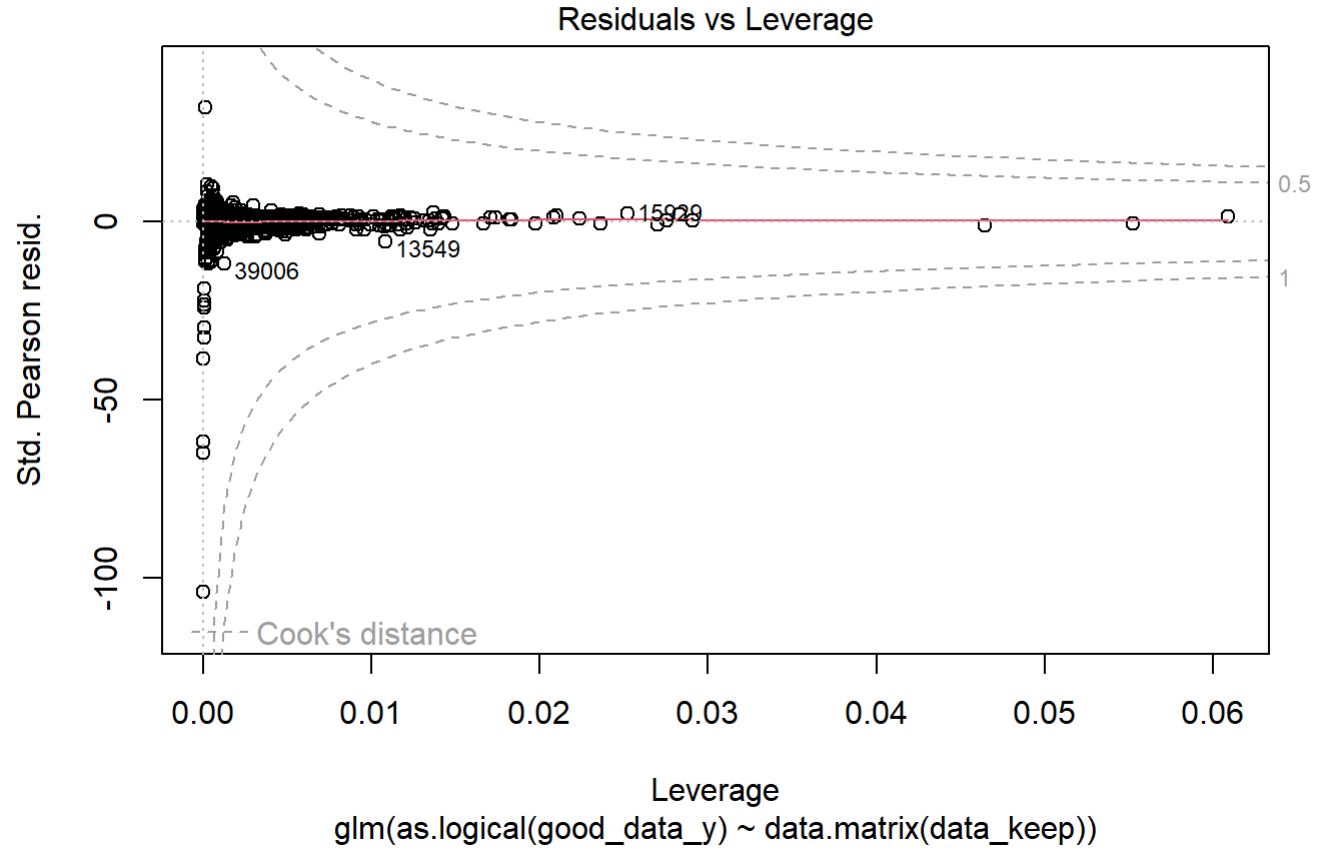
```
##
## Number of Fisher Scoring iterations: 5
```

```
plot(logit_model)
```


Residuals vs Fitted
Residuals
0
-50
-100
37261
16514
36224
-10
0
10
20
30
Predicted values
glm(as.logical(good_data_y) ~ data.matrix(data_keep))


Normal Q-Q
Std. Pearson resid.
0
-50
-100
37261
16514
36224
-4
-2
0
2
4
Theoretical Quantiles
glm(as.logical(good_data_y) ~ data.matrix(data_keep))


Scale-Location
√|Std. Pearson resid.|
36224
16514
37261
0
2
4
6
8
10
-10
0
10
20
30
Predicted values
glm(as.logical(good_data_y) ~ data.matrix(data_keep))


Residuals vs Leverage
Std. Pearson resid.
39006
13549
15929
0.5
1
Cook's distance
0
-50
-100
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Leverage
glm(as.logical(good_data_y) ~ data.matrix(data_keep))

```
data_keep_int = data_keep[, data_keep[, !sapply(.SD, is.double)], with = FALSE]

zip_model = pscl::zeroinfl(good_data_y ~ data.matrix(data_keep_int), dist = "negbin")
summary(zip_model)
```

```
##
## Call:
## pscl::zeroinfl(formula = good_data_y ~ data.matrix(data_keep_int), dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.4552 -0.3958 -0.3267 -0.2369 42.6874
##
## Count model coefficients (negbin with log link):
##                                              Estimate Std. Error  z value
## (Intercept)                                -6.403e-01  2.061e-02  -31.064
## data.matrix(data_keep_int)incentives       -2.852e-03  1.152e-03   -2.476
## data.matrix(data_keep_int)laws and regulations  2.223e-02  2.085e-03   10.660
## data.matrix(data_keep_int)asian             3.317e-04  2.056e-05   16.136
## data.matrix(data_keep_int)pacific           1.360e-03  2.378e-04    5.719
## Log(theta)                                 -1.565e+00  1.528e-02 -102.406
##                                             Pr(>|z|)
## (Intercept)                                  < 2e-16 ***
## data.matrix(data_keep_int)incentives          0.0133 *
## data.matrix(data_keep_int)laws and regulations  < 2e-16 ***
## data.matrix(data_keep_int)asian              < 2e-16 ***
## data.matrix(data_keep_int)pacific           1.07e-08 ***
## Log(theta)                                   < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                                              Estimate Std. Error z value
## (Intercept)                                  0.759671   0.082930   9.160
## data.matrix(data_keep_int)incentives       -0.036532   0.006822  -5.355
## data.matrix(data_keep_int)laws and regulations -0.061263   0.011218  -5.461
## data.matrix(data_keep_int)asian            -0.032758   0.003409  -9.611
## data.matrix(data_keep_int)pacific          -0.021663   0.010611  -2.042
##                                             Pr(>|z|)
## (Intercept)                                  < 2e-16 ***
## data.matrix(data_keep_int)incentives        8.55e-08 ***
## data.matrix(data_keep_int)laws and regulations 4.73e-08 ***
## data.matrix(data_keep_int)asian              < 2e-16 ***
## data.matrix(data_keep_int)pacific             0.0412 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 0.209
## Number of iterations in BFGS optimization: 85
## Log-likelihood: -5.506e+04 on 11 Df
```