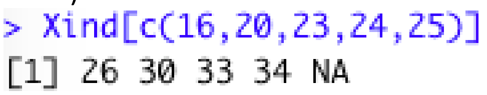
Corn data

Current Problem: glm detects the separation (i.e. our estimator is at infinity) but our method, glmdr does not. Furthermore, we want to see if detect_separation (from brglm2 package) can detect the separation in this case. I firstly mention few findings in corndata.R in part 1 then discuss what I did in part 2.

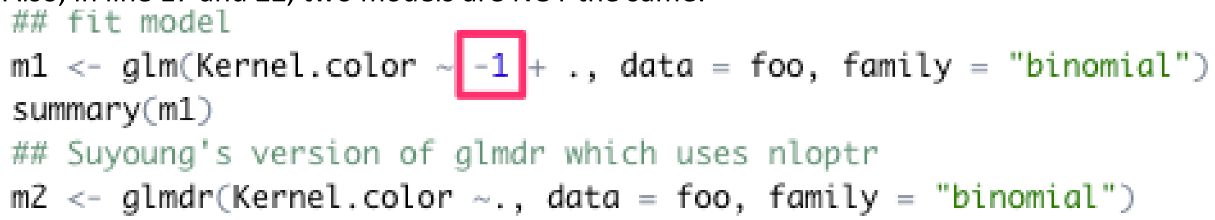
Part 1: corndata.R
To my understanding, the problem setting is that we want to fit the model using glmdr then fit the model again based on the subset that excludes problematic points (i.e. linearity == FALSE). In this case, our model should not show the complete separation as we have already removed problematic points.

Given dataset (Combined_Final_Product) contains 1547 x 34 and we use the subset that its column starts from 11 to 34 (i.e. 24 columns. X matrix in the line 12 in the corndata.R). Then, we remove problematic columns (I am not sure this is what you want but to my understanding you want to remove NA columns in coefficients from glm). When we fit the model with this X matrix, coefficients for S6_82186661 (15th), S6_82218044 (19th), S6_82243861 (22nd), S6_82243874 (23rd) and S6_82243883 (24th) are “NA” (Based on my knowledge, this happens due to the multicollinearity – maybe I assume this is reason why you checked eigenvalue of X^TX matrix?). However, I think there is indexing problem in Xind – line 14 (it should not show “NA”):



```
> Xind[c(16,20,23,24,25)]
[1] 26 30 33 34 NA
```

Also, in line 17 and 22, two models are NOT the same:



```
## fit model
m1 <- glm(Kernel.color ~ -1 + ., data = foo, family = "binomial")
summary(m1)
## Suyoung's version of glmdr which uses nloptr
m2 <- glmdr(Kernel.color ~., data = foo, family = "binomial")
```

That is, m1 does not have intercept term, meanwhile, m2 have the intercept term (but this may not be important to the next step).

After all, I think problem in this code comes from our model matrix, foo, which fails to remove columns that have an exact collinearity problem.

Part 2: corndata2.R
In this file, I removed the duplicated column for multicollinearity and compare glm, glmdr and detect separation (as of now, author separated detect_separation function from brglm2 and created the new package called, “detectseparation.” Thus, you may need to install this package from CRAN to run my code properly).

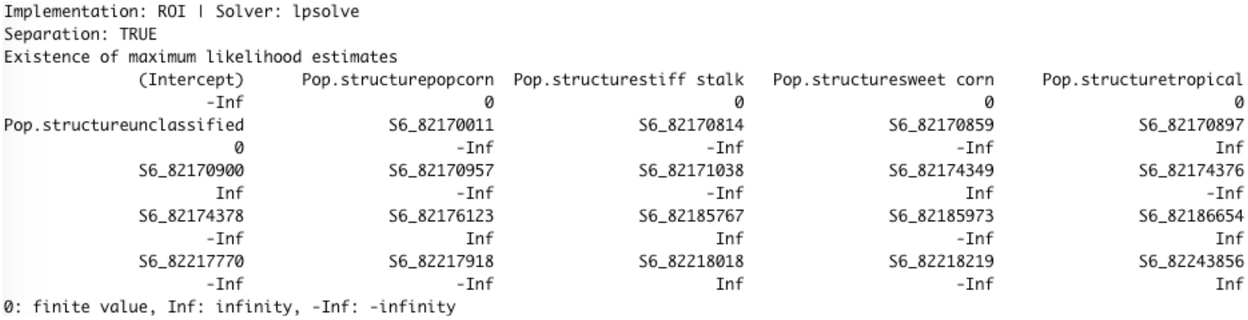
Firstly, I checked which columns have the same values (line 14-18 in corndata2.R) to confirm my interpretation (multicollinearity) in the glm function. It turns out 1) 14th and 15th 2) 18th and 19th

3) 21, 22, 23, and 24th columns have the same value. Thus, I removed 15, 19, 22, 23, 24th column from our model matrix and created the foo while column-wise binding kernel.color and pop.structure.

Q: Without statistical or mathematical knowledge, is it okay to drop the column even though they have the same values considering we are handling *genetic* data? If yes, which one should we drop in each case (e.g. 21, 22, 23, 24th columns have the all same values but they are different gene, then among them, which one is the most important in the inference?)

To detect the complete separation, I firstly fit the model using 1) glm (named as xyz_glm) then 2) send them to the infinity by changing maxit and epsilon in the control from glm function (named as xyz_inf). If any of estimators goes to infinity, we can see there is a complete separation problem. If all estimators are close or the same between 1) and 2), then there is no separation problem.
I used 3) detect_separation function (named _sep) and 4) glmdr (named _glmdr).

Interestingly, result from the detect_separation says that all parameters expect the "Pop.structure" lie at infinity.

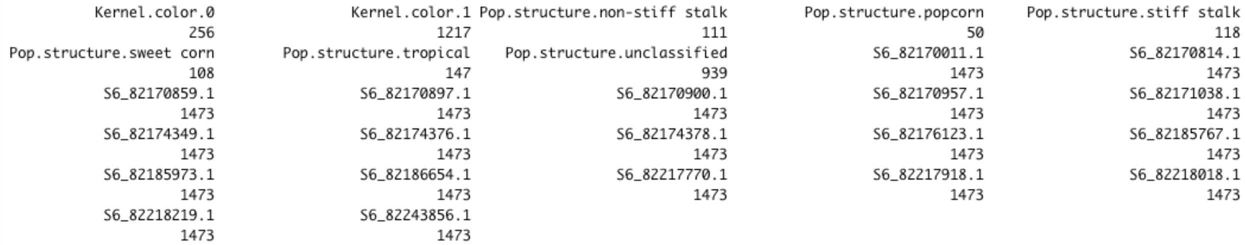


```
Implementation: ROI | Solver: lpsolve
Separation: TRUE
Existence of maximum likelihood estimates
              (Intercept)      Pop.structurepopcorn Pop.structurestiff stalk  Pop.structuresweet corn     Pop.structuretropical
                     -Inf                         0                        0                        0                         0
Pop.structureunclassified                S6_82170011              S6_82170814              S6_82170859               S6_82170897
                        0                      -Inf                     -Inf                     -Inf                       Inf
              S6_82170900                S6_82170957              S6_82171038              S6_82174349               S6_82174376
                      Inf                      -Inf                     -Inf                      Inf                      -Inf
              S6_82174378                S6_82176123              S6_82185767              S6_82185973               S6_82186654
                     -Inf                       Inf                      Inf                     -Inf                       Inf
              S6_82217770                S6_82217918              S6_82218018              S6_82218219               S6_82243856
                     -Inf                      -Inf                      Inf                     -Inf                       Inf
0: finite value, Inf: infinity, -Inf: -infinity
```

Our glmdr shows that there is the complete separation and there are **74** problematic points (in corndata.R, we saw only 3 problematic points).
Therefore, combining results from 3) and 4), suppose we fit the model EXCLUDING the 74 problematic points we identified in 4), we will be able to estimate the parameter for Pop.structure (other columns will be NA) ⇔ we should NOT see separation problem.

Using subset (linearity == TURE), as we expected, we can estimate parameters of pop structure (line 39 – 40, and there is no warning message (glm.fit: fitted probabilities numerically 0 or 1 occurred) as well, although it does not guarantee the exist of the infinite estimator). Notice that m2_inf (line 42 - 43) still shows different results with m2_glm even though there is no complete separation. This is because our data, foo[m1_glmdr$linearity,] contain the all same value in every column except the pop structure (line 44).

(Continue on the next page)



```
          Kernel.color.0             Kernel.color.1 Pop.structure.non-stiff stalk      Pop.structure.popcorn  Pop.structure.stiff stalk
                     256                       1217                           111                         50                        118
Pop.structure.sweet corn     Pop.structure.tropical    Pop.structure.unclassified              S6_82170011.1              S6_82170814.1
                     108                        147                           939                       1473                       1473
           S6_82170859.1              S6_82170897.1                 S6_82170900.1              S6_82170957.1              S6_82171038.1
                    1473                       1473                          1473                       1473                       1473
           S6_82174349.1              S6_82174376.1                 S6_82174378.1              S6_82176123.1              S6_82185767.1
                    1473                       1473                          1473                       1473                       1473
           S6_82185973.1              S6_82186654.1                 S6_82217770.1              S6_82217918.1              S6_82218018.1
                    1473                       1473                          1473                       1473                       1473
           S6_82218219.1              S6_82243856.1
                    1473                       1473
```

** Column_name.value and number of frequencies. E.g. Kernel.color.0: 256 => in Kernel.color column there are 256 zeros (0).*

In 3) and 4), we can see both functions say MLE exists in the conventional sense in the OM. Therefore, all results reach the same conclusion.