

Technical Reports

```
library(readxl)
library(glmdr)
library(brglm2)
library(arm)
library(binom)
library(ggplot2)
library(logistf)
library(foreach)
library(doParallel)
```

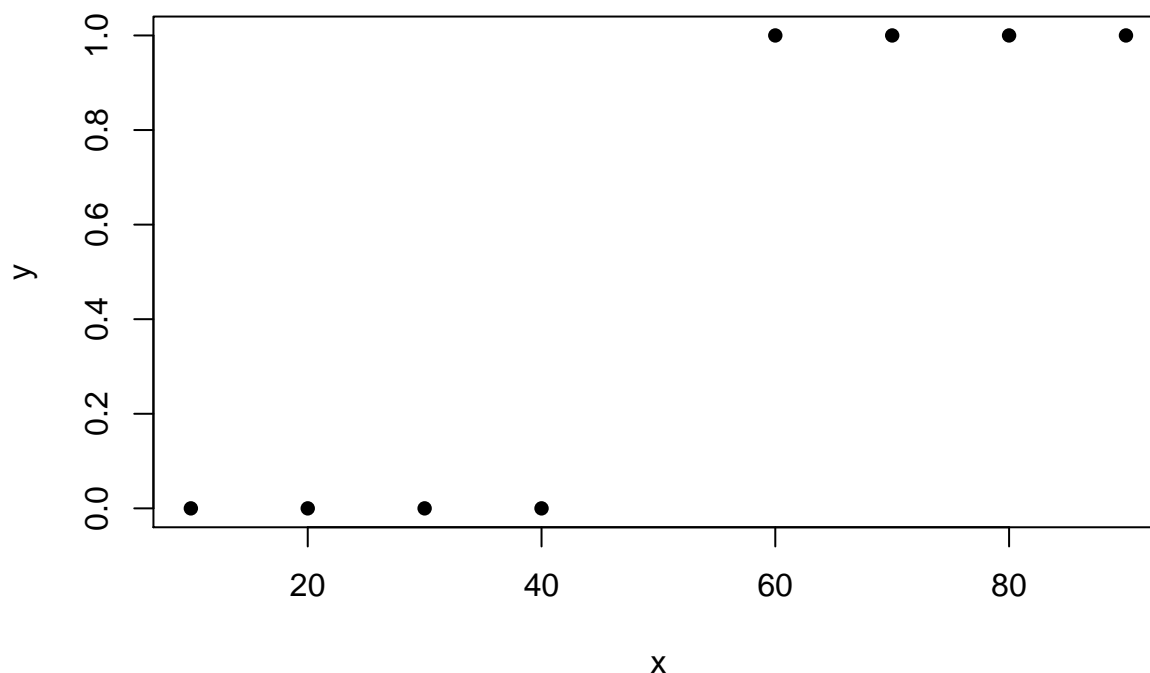
Agresti Complete Separation Example

Motivation

```
x <- c(10,20,30,40,60,70,80,90)
y <- c(0,0,0,0,1,1,1,1)
mod1 <- glm(y~x,family="binomial")
```

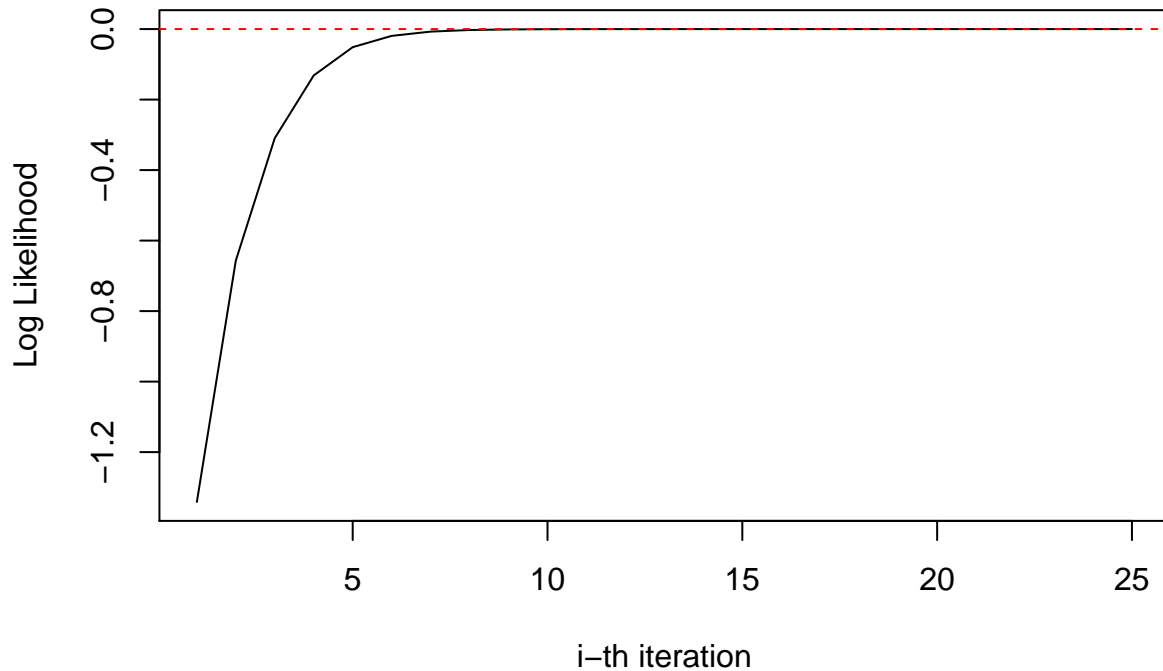
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
plot(y~x, pch=16)
```

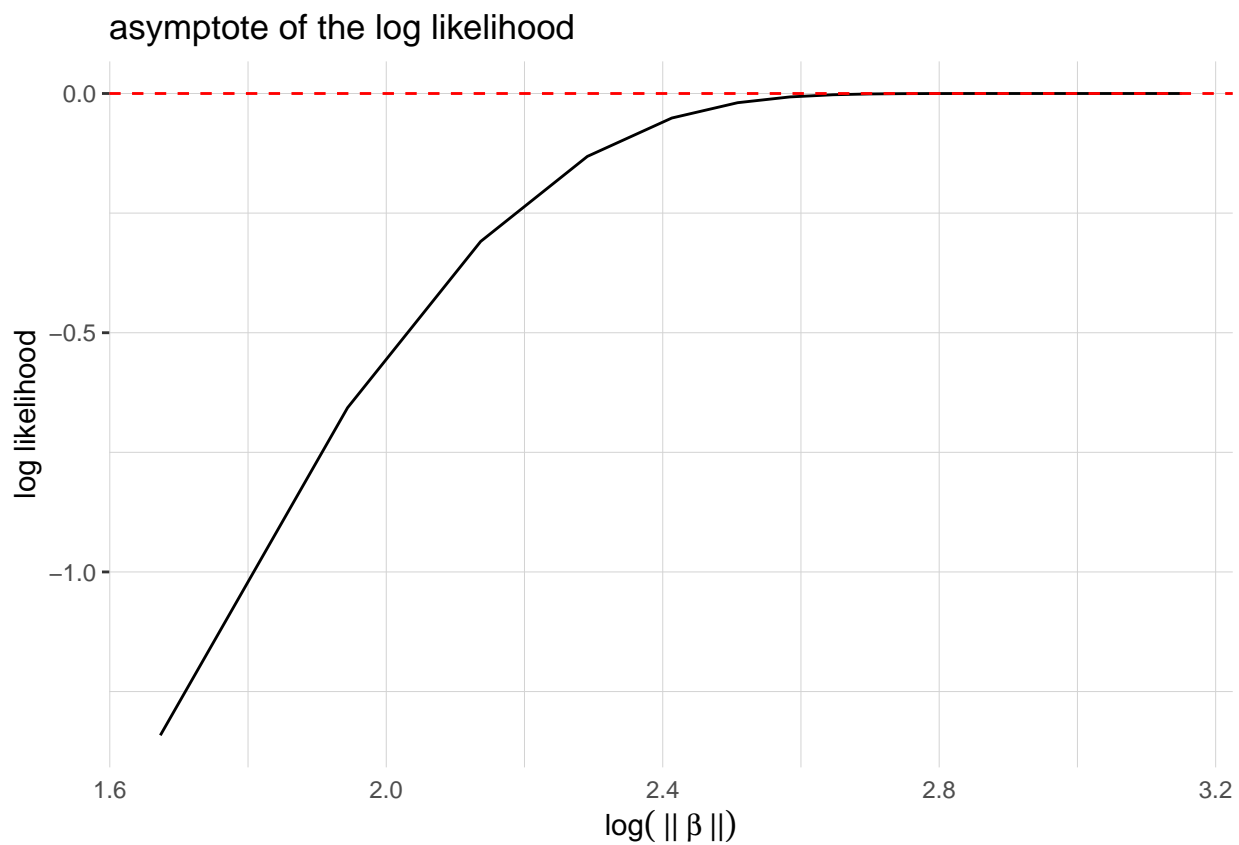


```
n <- mod1$iter
lst <- rep(NA,n)
for(i in 1:n){
  lst[i] <- suppressWarnings(logLik(glm(y~x, family="binomial", control = list(maxit=i,eps=1e-20))))
}
```

```
plot(lst,ylab="Log Likelihood",xlab="i-th iteration",type="l")
abline(a =0, b =0, lty=2, col ="red")
```

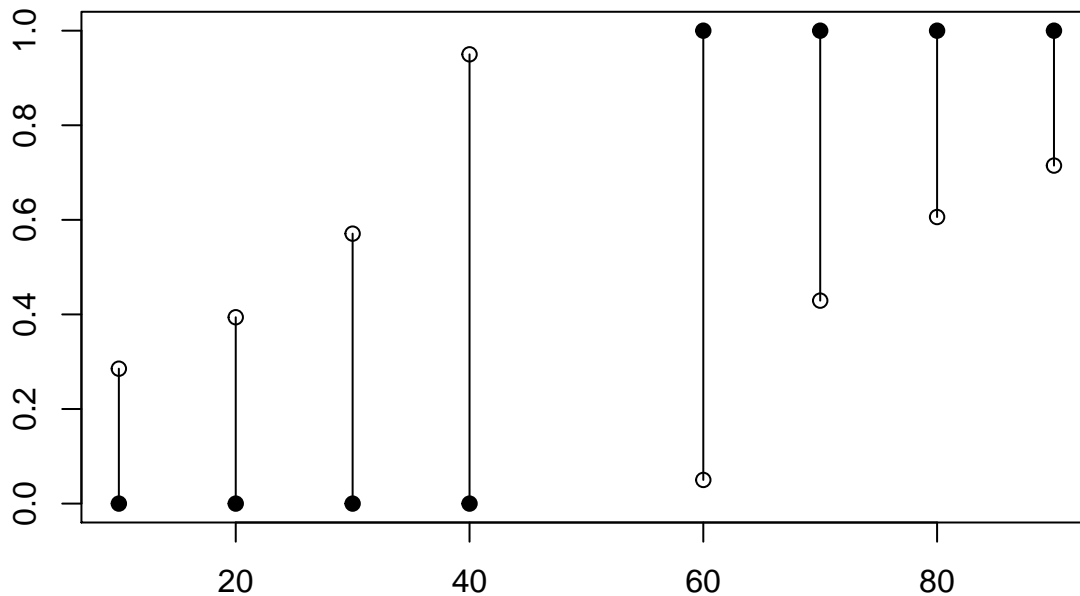


```
asymptote <- t(sapply(1:30, function(iter){
  m1 <- suppressWarnings(glm(y ~ x, family = "binomial", control = list(maxit = iter, epsilon = 1e-20)))
  c(sqrt(log(crossprod(coef(m1)))), logLik(m1))
})))
asymptote <- as.data.frame(asymptote)
ggplot(asymptote) +
  ggtitle("asymptote of the log likelihood") +
  labs(x= expression(log(~"||"~beta~"||")), y= "log likelihood") +
  geom_line(aes(x = V1, y = V2), col = "black") +
  geom_abline(intercept = 0, slope = 0, lty = 2, col = "red") +
  theme(legend.position="bottom", panel.background = element_blank(),
        legend.key = element_rect(fill = "white"),
        axis.ticks.x = element_blank(),
        panel.grid.major.x = element_line("lightgrey", size = 0.15),
        panel.grid.major.y = element_line("lightgrey", size = 0.15),
        panel.grid.minor.x = element_line("lightgrey", size = 0.07),
        panel.grid.minor.y = element_line("lightgrey", size = 0.07))
```



Estimation

```
glmdr_mod <- glmdr(y~x,family="binomial")
mod_CI <- inference(glmdr_mod)
plot(x, y, ylim = c(0,1), pch = 16, ylab = "", xlab = "")
points(x, mod_CI[, 1])
points(x, mod_CI[, 2])
segments(x, mod_CI[, 1], x, mod_CI[, 2])
```



Comparison

```

bayes_mod <- bayesglm(y~x,family="binomial")
glmldr_mod <- glmldr(y~x,family="binomial")
brglm_mod <- glm(y~x, family = "binomial", method = "brglmFit", type = "MPL_Jeffreys")

mean((predict(bayes_mod,type="response") >= 0.5) == y)

## [1] 1

bayes_CI <- binom.confint(predict(bayes_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
bayes_length <- sum(as.matrix(bayes_CI) %*% c(-1,1))

mean((predict(brglm_mod,type="response") >= 0.5) == y)

## [1] 1

brglm_CI <- binom.confint(predict(brglm_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
brglm_length <- sum(as.matrix(brglm_CI) %*% c(-1,1))

mean((predict(glmldr_mod$om,type="response") >= 0.5) == y)

## [1] 1

glmldr_CI <- binom.confint(predict(glmldr_mod$om,type="response"),n=1,methods="wilson")[,c(5,6)]
glmldr_length <- sum(as.matrix(glmldr_CI) %*% c(-1,1))

cbind(glmldr_length,bayes_length,brglm_length)

##      glmldr_length bayes_length brglm_length
## [1,]      6.347605      6.627746      6.680239

```

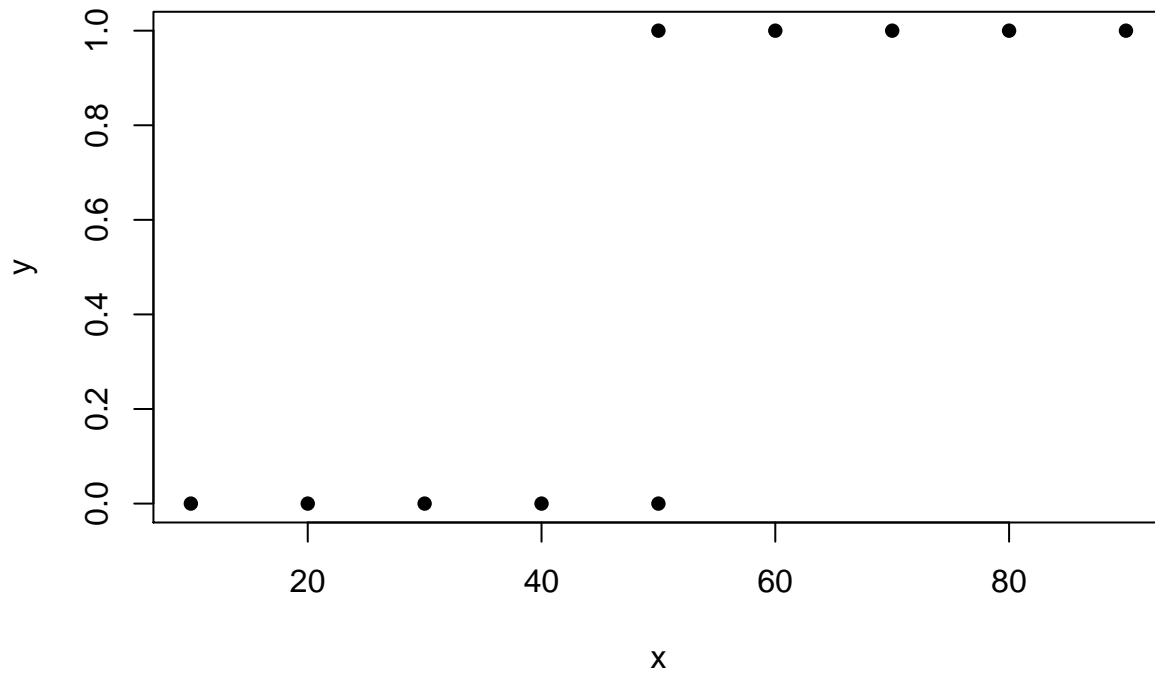
Agresti Quasi-Complete Separation Example

Motivation

```
x <- c(10,20,30,40,60,70,80,90,50,50)
y <- c(0,0,0,0,1,1,1,1,0,1)
mod1 <- glm(y~x,family="binomial")

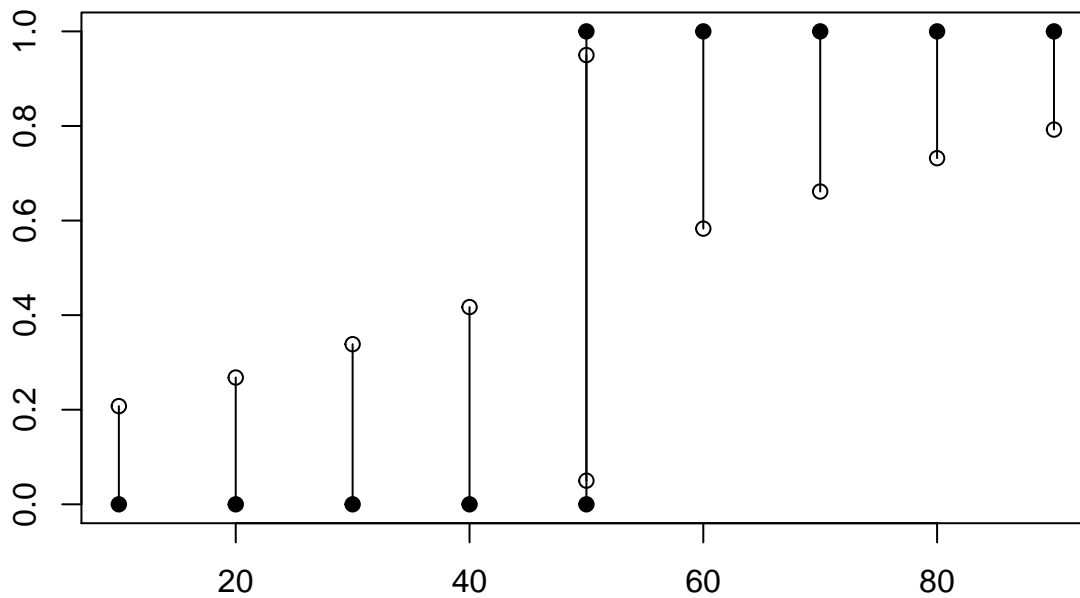
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
summary(mod1)

##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.177    0.000    0.000    0.000    1.177
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -98.158   39288.592  -0.002    0.998
## x              1.963     785.772   0.002    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 13.8629  on 9  degrees of freedom
## Residual deviance:  2.7726  on 8  degrees of freedom
## AIC: 6.7726
##
## Number of Fisher Scoring iterations: 21
plot(y~x, pch=16)
```



Estimation

```
glmdr_mod <- glmdr(y~x,family="binomial")
mod_CI <- inference(glmdr_mod)
mod_CI <- rbind(mod_CI,c(0,0.95),c(0.05,1))
plot(x, y, ylim = c(0,1), pch = 16, ylab = "", xlab = "")
points(x, mod_CI[, 1])
points(x, mod_CI[, 2])
segments(x, mod_CI[, 1], x, mod_CI[, 2])
```



Comparison

```
bayes_mod <- bayesglm(y~x,family="binomial")
glmdr_mod <- glmdr(y~x,family="binomial")
brglm_mod <- glm(y~x, family = "binomial", method = "brglmFit", type = "MPL_Jeffreys")

mean((predict(bayes_mod,type="response") >= 0.5) == y)

## [1] 0.9

bayes_CI <- binom.confint(predict(bayes_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
bayes_length <- sum(as.matrix(bayes_CI) %*% c(-1,1))

mean((predict(brglm_mod,type="response") >= 0.5) == y)

## [1] 0.9

brglm_CI <- binom.confint(predict(brglm_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
brglm_length <- sum(as.matrix(brglm_CI) %*% c(-1,1))

mean(c((predict(glmdr_mod$om,type="response")[!glmdr_mod$linearity] >= 0.5) == y[!glmdr_mod$linearity]
      ,(predict(glmdr_mod$lcm,type="response") >= 0.5) == y[glmdr_mod$linearity]))

## [1] 0.9

glmdr_CI <- binom.confint(predict(glmdr_mod$om,type="response")[!glmdr_mod$linearity],n=1,methods="wilson")
glmdr_CI <- rbind(glmdr_CI,binom.confint(predict(glmdr_mod$lcm,type="response"),n=1,methods="wilson"),
glmdr_length <- sum(as.matrix(glmdr_CI) %*% c(-1,1))

cbind(glmdr_length,bayes_length,brglm_length)

##      glmdr_length bayes_length brglm_length
## [1,]      8.129122      8.398411      8.42598
```

Quadratic Example

Motivation

```
data(quadratic)
mod <- glm(y~x+I(x^2),data=quadratic,family="binomial")

## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

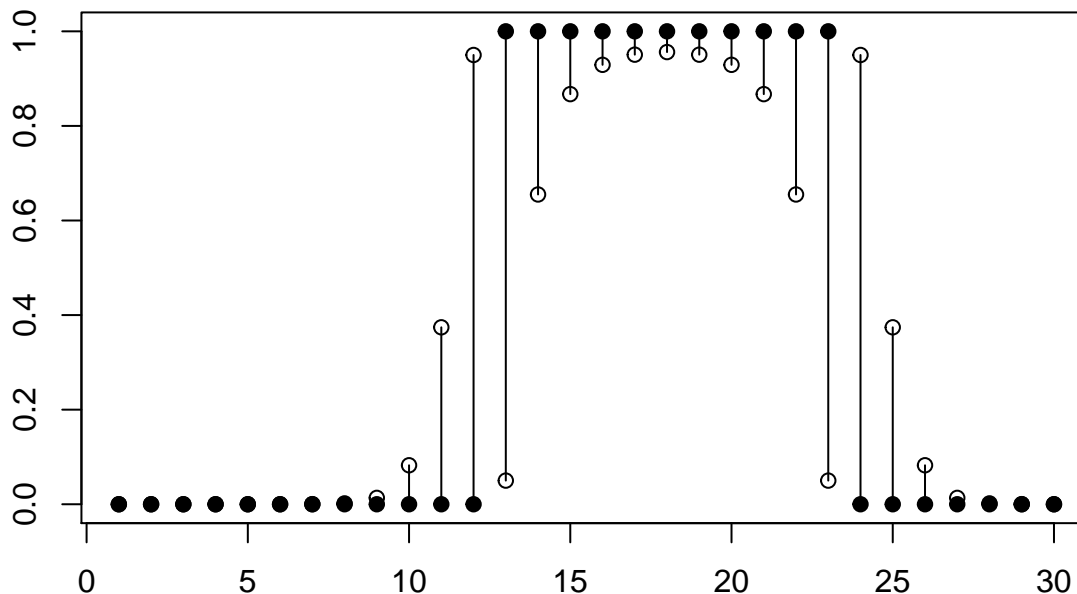
summary(mod)

##
## Call:
## glm(formula = y ~ x + I(x^2), family = "binomial", data = quadratic)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -5.401e-05 -2.100e-08 -2.100e-08  2.100e-08  5.637e-05
##
```

```
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1084.702  420692.854  -0.003   0.998
## x           133.047   51514.292   0.003   0.998
## I(x^2)       -3.696    1431.369  -0.003   0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3.9429e+01  on 29  degrees of freedom
## Residual deviance: 1.1902e-08  on 27  degrees of freedom
## AIC: 6
##
## Number of Fisher Scoring iterations: 25
```

Estimation

```
glmldr_mod <- glmldr(y~x+I(x^2),data=quadratic,family="binomial")
mod_CI <- inference(glmldr_mod)
with(quadratic,plot(x, y, ylim = c(0,1), pch = 16,
  ylab = "", xlab = ""))
with(quadratic,points(x, mod_CI[, 1]))
with(quadratic,points(x, mod_CI[, 2]))
with(quadratic,segments(x, mod_CI[, 1], x, mod_CI[, 2]))
```



Comparison

```
bayes_mod <- bayesglm(y~x+I(x^2),data=quadratic,family="binomial")
glmldr_mod <- glmldr(y~x+I(x^2),data=quadratic,family="binomial")
brglm_mod <- glm(y~x+I(x^2),data=quadratic,family = "binomial", method = "brglmFit", type = "MPL_Jeffrey")

## Warning: brglmFit: fitted probabilities numerically 0 or 1 occurred
```



```

mean((predict(bayes_mod,type="response") >= 0.5) == quadratic$y)

## [1] 1
bayes_CI <- binom.confint(predict(bayes_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
bayes_length <- sum(as.matrix(bayes_CI) %*% c(-1,1))

mean((predict(brglm_mod,type="response") >= 0.5) == quadratic$y)

## [1] 1
brglm_CI <- binom.confint(predict(brglm_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
brglm_length <- sum(as.matrix(brglm_CI) %*% c(-1,1))

mean((predict(glmdr_mod$om,type="response") >= 0.5) == quadratic$y)

## [1] 1
glmdr_CI <- binom.confint(predict(glmdr_mod$om,type="response"),n=1,methods="wilson")[,c(5,6)]
glmdr_length <- sum(as.matrix(glmdr_CI) %*% c(-1,1))

cbind(glmdr_length,bayes_length,brglm_length)

##      glmdr_length bayes_length brglm_length
## [1,]      23.80352      24.70393      24.32923

```

Endometrial

Motivation

```

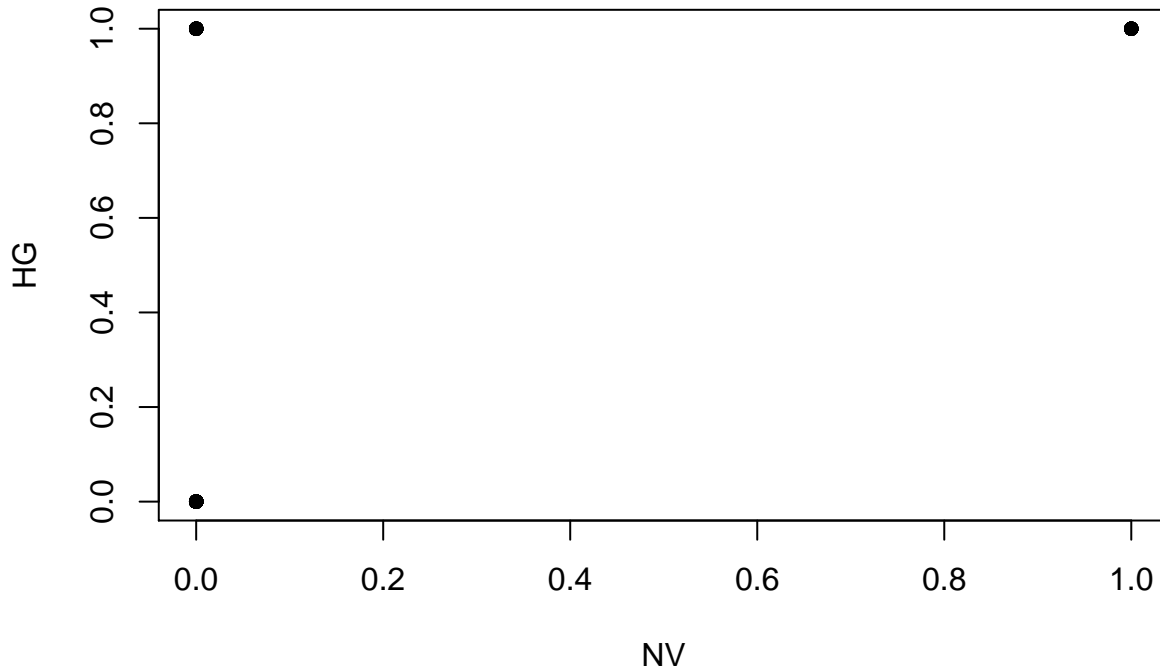
data(endometrial)
mod <- glm(HG~.,data=endometrial,family="binomial")
summary(mod)

##
## Call:
## glm(formula = HG ~ ., family = "binomial", data = endometrial)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50137  -0.64108  -0.29432   0.00016   2.72777
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.30452    1.63730   2.629 0.008563 **
## NV            18.18556   1715.75089   0.011 0.991543
## PI            -0.04218    0.04433  -0.952 0.341333
## EH            -2.90261    0.84555  -3.433 0.000597 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 104.903  on 78  degrees of freedom

```

```
## Residual deviance: 55.393 on 75 degrees of freedom
## AIC: 63.393
##
## Number of Fisher Scoring iterations: 17
```

```
with(endometrial,plot(HG~NV,pch=16))
```



Estimation

```
glmdr_mod <- glmdr(HG~.,data=endometrial,family="binomial")
inference(glmdr_mod)
```

```
##      lower upper
## 22 0.7071451    1
## 23 0.7432730    1
## 24 0.9205964    1
## 25 0.8325905    1
## 26 0.9518366    1
## 48 0.5479196    1
## 49 0.5467740    1
## 50 0.9160397    1
## 51 0.8703443    1
## 71 0.9205150    1
## 75 0.8703894    1
## 76 0.8277338    1
## 78 0.8231611    1
```

Comparison

```

bayes_mod <- bayesglm(HG~.,data=endometrial,family="binomial")
glmdr_mod <- glmdr(HG~.,data=endometrial,family="binomial")
brglm_mod <- glm(HG~.,data=endometrial,family = "binomial", method = "brglmFit", type = "MPL_Jeffreys")

mean((predict(bayes_mod,type="response") >= 0.5) == endometrial$HG)

## [1] 0.8860759

bayes_CI <- binom.confint(predict(bayes_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
bayes_length <- sum(as.matrix(bayes_CI) %*% c(-1,1))

mean((predict(brglm_mod,type="response") >= 0.5) == endometrial$HG)

## [1] 0.8860759

brglm_CI <- binom.confint(predict(brglm_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
brglm_length <- sum(as.matrix(brglm_CI) %*% c(-1,1))

mean(c((predict(glmdr_mod$om,type="response")[!glmdr_mod$linearity] >= 0.5) == endometrial$HG[!glmdr_mod$linearity],
(predict(glmdr_mod$lcm,type="response") >= 0.5) == endometrial$HG[glmdr_mod$linearity]))

## [1] 0.8860759

glmdr_CI <- binom.confint(predict(glmdr_mod$om,type="response")[!glmdr_mod$linearity],n=1,methods="wilson")
glmdr_CI <- rbind(glmdr_CI,binom.confint(predict(glmdr_mod$lcm,type="response"),n=1,methods="wilson"),
glmdr_length <- sum(as.matrix(glmdr_CI) %*% c(-1,1))

cbind(glmdr_length,bayes_length,brglm_length)

##      glmdr_length bayes_length brglm_length
## [1,]      66.22537      66.5582      66.63425

```

Maize data

Data Cleaning

```

corn_dat <- readxl::read_xlsx("./Combined_Final_Product.xlsx")
names(corn_dat)[c(10,8)] <- c("Kernel.color","Pop.structure")
Xind <- 11:ncol(corn_dat)
foo <- corn_dat[, c(10,8,Xind[-c(15,19,22,23,24)])]
foo$Pop.structure <- factor(foo$Pop.structure)
dat <- data.frame(foo)

```

Motivation

Univariate Case

When we fit the logistic model one by one (univariate case), there is no separation.

```

n <- ncol(dat)
ret_vec <- rep(TRUE,n) #TRUE: separation does not exist / FALSE: separation exists
Y <- foo$Kernel.color
for(i in 2:n){

```

```

X <- unlist(dat[,i])
mod <- glm(Y ~ X, family="binomial")
ret_vec[(i-1)] <- is.null(mod$linearity)
}
all(ret_vec)

```

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

Multivariate Case

```
mod <- glm(Kernel.color ~ ., data=dat, family="binomial")
```

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

```
summary(mod)
```

```
##
## Call:
## glm(formula = Kernel.color ~ ., family = "binomial", data = dat)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7971   0.2270   0.5638   0.5638   1.5387
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.034e+02  2.968e+04  -0.010   0.9918
## Pop.structurepopcorn    1.464e+00  1.068e+00   1.370   0.1706
## Pop.structurestiff stalk    1.219e+00  6.804e-01   1.791   0.0733 .
## Pop.structuresweet corn    1.128e+00  6.810e-01   1.656   0.0978 .
## Pop.structuretropical    -3.246e+00  3.911e-01  -8.300 <2e-16 ***
## Pop.structureunclassified -6.691e-01  3.597e-01  -1.860   0.0629 .
## S6_82170011    -3.226e+03  1.178e+05  -0.027   0.9781
## S6_82170814     1.051e+02  6.299e+03   0.017   0.9867
## S6_82170859    -2.986e+03  1.459e+05  -0.020   0.9837
## S6_82170897     1.556e+02  5.719e+03   0.027   0.9783
## S6_82170900     2.427e+03  1.308e+05   0.019   0.9852
## S6_82170957    -5.946e+01  5.803e+03  -0.010   0.9918
## S6_82171038     5.741e+02  4.685e+04   0.012   0.9902
## S6_82174349    -1.011e+02  1.667e+04  -0.006   0.9952
## S6_82174376     1.725e+02  1.795e+04   0.010   0.9923
## S6_82174378     1.282e+02  1.692e+04   0.008   0.9940
## S6_82176123     1.663e+02  3.685e+04   0.005   0.9964
## S6_82185767     1.215e+03  5.106e+04   0.024   0.9810
## S6_82185973    -4.137e+02  2.006e+04  -0.021   0.9835
## S6_82186654     2.548e+02  2.366e+04   0.011   0.9914
## S6_82217770    -2.419e+02  9.601e+03  -0.025   0.9799
## S6_82217918    -2.272e+03  1.040e+05  -0.022   0.9826
## S6_82218018     4.045e+02  7.478e+04   0.005   0.9957
## S6_82218219    -1.664e+02  1.330e+04  -0.013   0.9900
## S6_82243856     4.169e+03  1.555e+05   0.027   0.9786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1547.1  on 1546  degrees of freedom
## Residual deviance: 1092.6  on 1522  degrees of freedom
## AIC: 1142.6
##
## Number of Fisher Scoring iterations: 22
```

Estimation

```
glmdr_mod <- glmdr(Kernel.color~.,data=dat,family="binomial")
inference(glmdr_mod)
```

##		lower	upper
## 8		0.98813949	1.000000e+00
## 31		0.00000000	6.237754e-04
## 34		0.00000000	4.741097e-06
## 35		0.00000000	9.500000e-01
## 53		0.00000000	9.500000e-01
## 54		0.00000000	1.164959e-06
## 55		0.00000000	9.499997e-01
## 60		0.00000000	9.499996e-01
## 63		0.00000000	9.499997e-01
## 66		0.00000000	5.671587e-03
## 70		0.78224613	1.000000e+00
## 107		0.00000000	9.500000e-01
## 117		0.00000000	9.499995e-01
## 126		0.05000002	1.000000e+00
## 128		0.05000018	1.000000e+00
## 129		0.33368737	1.000000e+00
## 133		0.00000000	1.786882e-10
## 135		0.00000000	5.899264e-08
## 138		0.00000000	9.499980e-01
## 148		0.00000000	9.499999e-01
## 149		0.00000000	2.137789e-07
## 190		0.70612654	1.000000e+00
## 191		0.05000005	1.000000e+00
## 235		0.99847863	1.000000e+00
## 244		0.99997116	1.000000e+00
## 249		0.00000000	9.499988e-01
## 266		0.05000001	1.000000e+00
## 454		0.00000000	1.743899e-03
## 456		0.05000000	1.000000e+00
## 458		0.00000000	6.446680e-04
## 476		0.00000000	2.690309e-05
## 497		0.00000000	9.499993e-01
## 581		0.00000000	9.500000e-01
## 592		0.00000000	5.444221e-05
## 601		0.05000000	1.000000e+00
## 605		0.00000000	9.499999e-01
## 626		0.00000000	1.783803e-11
## 696		0.00000000	9.500000e-01

```
## 710 0.00000000 9.500000e-01
## 774 0.00000000 8.239179e-11
## 775 0.00000000 9.500000e-01
## 778 0.00000000 9.499999e-01
## 779 0.00000000 9.500000e-01
## 886 0.00000000 5.171483e-09
## 890 0.00000000 5.974480e-05
## 905 0.00000000 9.500000e-01
## 915 0.05000012 1.000000e+00
## 919 0.00000000 9.499999e-01
## 920 0.00000000 9.500000e-01
## 922 0.00000000 7.341575e-10
## 923 0.00000000 5.670205e-05
## 932 0.00000000 9.836097e-17
## 939 0.05000000 1.000000e+00
## 947 0.05000008 1.000000e+00
## 954 0.05000000 1.000000e+00
## 1017 0.05000000 1.000000e+00
## 1073 0.05000003 1.000000e+00
## 1116 0.00000000 9.499999e-01
## 1121 0.00000000 9.500000e-01
## 1127 0.00000000 6.159062e-07
## 1210 0.00000000 4.740907e-08
## 1305 0.00000000 3.706127e-05
## 1313 0.00000000 8.404463e-03
## 1319 0.00000000 5.665611e-01
## 1332 0.05000007 1.000000e+00
## 1379 0.00000000 9.500000e-01
## 1412 0.00000000 9.499999e-01
## 1428 0.00000000 7.455165e-06
## 1431 0.05000006 1.000000e+00
## 1435 0.00000000 4.999734e-05
## 1440 0.00000000 9.499999e-01
## 1485 0.00000000 1.236760e-11
## 1517 0.05000007 1.000000e+00
## 1520 0.00000000 9.499999e-01
```

Comparison

```
bayes_mod <- bayesglm(Kernel.color~.,data=dat,family="binomial")
glmldr_mod <- glmldr(Kernel.color~.,data=dat,family="binomial")
brglm_mod <- glm(Kernel.color~.,data=dat, family = "binomial", method = "brglmFit", type = "MPL_Jeffrey")

## Warning: brglmFit: algorithm did not converge
## Warning: brglmFit: fitted probabilities numerically 0 or 1 occurred
#brglm does not converge
logistif_mod <- logistf(Kernel.color~.,data=dat, family = "binomial")
lm_mod <- lm(Kernel.color~.,data=dat)

mean((predict(bayes_mod,type="response") >= 0.5) == dat$Kernel.color)

## [1] 0.8707175
```

```

bayes_CI <- binom.confint(predict(bayes_mod,type="response"),n=1,methods="wilson")[,c(5,6)]
bayes_length <- sum(as.matrix(bayes_CI) %>% c(-1,1))

logistif_pred <- c(invlogit(model.matrix(glmldr_mod$om) %>% logistif_mod$coefficients))
mean((logistif_pred >= 0.5) == dat$Kernel.color)

## [1] 0.8700711

logistf_CI <- binom.confint(logistif_pred,n=1,methods="wilson")[,c(5,6)]
logistf_length <- sum(as.matrix(logistf_CI) %>% c(-1,1))

mean(c((predict(glmldr_mod$om,type="response") >= 0.5)[!glmldr_mod$linearity] == dat$Kernel.color[!glmldr_

## [1] 0.8713639

glmldr_CI <- binom.confint(predict(glmldr_mod$om,type="response")[!glmldr_mod$linearity],n=1,methods="wilson")
glmldr_CI <- rbind(glmldr_CI,binom.confint(predict(glmldr_mod$lcm,type="response"),n=1,methods="wilson"),
glmldr_length <- sum(as.matrix(glmldr_CI) %>% c(-1,1))

lm_pred <- predict(lm_mod,se.fit=TRUE)
mean( (lm_pred$fit >= 0.5) == dat$Kernel.color)

## [1] 0.8681319

lm_CI <- cbind(lm_pred$fit - (1.96 * lm_pred$se.fit),lm_pred$fit + (1.96 * lm_pred$se.fit))
lm_length <- sum(as.matrix(lm_CI) %>% c(-1,1))

cbind(glmldr_length,bayes_length,logistf_length,lm_length)

##      glmldr_length bayes_length logistf_length lm_length
## [1,]      1293.014      1294.965      1296.37    152.3245

```

Prediction & Comparison

```

n_sample <- nrow(dat)
total_mat_glmdr_test <- matrix(NA,nrow=n_sample,ncol=8)
total_mat_bayes_test <- matrix(NA,nrow=n_sample,ncol=8)
total_mat_logistf_test <- matrix(NA,nrow=n_sample,ncol=8)
total_mat_linear_test <- matrix(NA,nrow=n_sample,ncol=8)

numCores <- detectCores()
registerDoParallel(numCores)
y_idx <- which(names(glmldr_mod$om$data) == paste(formula(glmldr_mod$om))[2])
#glmldr
ret <- foreach(i=1:n_sample) %dopar% {
  training <- dat[-i,]
  testing_X <- dat[i,-y_idx,drop=FALSE]
  testing_Y <- dat[i,y_idx]
  glmldr_mod <- glmldr(Kernel.color~.,data=training, family="binomial")
  ret2 <- predict(glmldr_mod,newdata = testing_X, alpha=0.05,crit="AICc",interval = "wilson")
  total_mat_glmdr_test[i,1] <- i
  total_mat_glmdr_test[i,4] <- sum(ifelse(rowMeans(ret2)>=0.5,1,0) == testing_Y & testing_Y == 1) #true
  total_mat_glmdr_test[i,5] <- sum(ifelse(rowMeans(ret2)>=0.5,1,0) == testing_Y & testing_Y == 0) #true
  total_mat_glmdr_test[i,6] <- sum(ifelse(rowMeans(ret2)>=0.5,1,0) != testing_Y & testing_Y == 0) #false
}

```

```

total_mat_glmldr_test[i,7] <- sum(iffelse(rowMeans(ret2)>=0.5,1,0) != testing_Y & testing_Y == 1) #false
total_mat_glmldr_test[i,8] <- sum(total_mat_glmldr_test[i,4:7])
total_mat_glmldr_test[i,]
}
total_mat_glmldr_test <- do.call(rbind,ret)

```

#bayesglm

```

ret <- foreach(i=1:n_sample) %dopar% {
  training <- dat[-i,]
  testing_X <- dat[i,-y_idx,drop=FALSE]
  testing_Y <- dat[i,y_idx]
  bayesglm_mod <- bayesglm(Kernel.color~.,data=training, family="binomial")
  ret2 <- predict(bayesglm_mod, newdata= testing_X, type="response")
  total_mat_bayes_test[i,1] <- i
  total_mat_bayes_test[i,4] <- sum((iffelse(ret2>=0.5,1,0) == testing_Y) & (testing_Y == 1)) #true positive
  total_mat_bayes_test[i,5] <- sum((iffelse(ret2>=0.5,1,0) == testing_Y) & (testing_Y == 0)) #true negative
  total_mat_bayes_test[i,6] <- sum((iffelse(ret2>=0.5,1,0) != testing_Y) & (testing_Y == 0)) #false positive
  total_mat_bayes_test[i,7] <- sum((iffelse(ret2>=0.5,1,0) != testing_Y) & (testing_Y == 1)) #false negative
  total_mat_bayes_test[i,8] <- sum(total_mat_bayes_test[i,4:7])
  total_mat_bayes_test[i,]
}
total_mat_bayes_test <- do.call(rbind,ret)

```

#logistf

```

start_time <- proc.time()
ret <- foreach(i=1:8) %dopar% {
  training <- dat[-i,]
  testing_X <- dat[i,-y_idx,drop=FALSE]
  testing_Y <- dat[i,y_idx]
  logistf_mod <- logistf(Kernel.color~.,data=training, family="binomial")
  new_dat <- rbind(logistf_mod$data,as.data.frame(cbind("Kernel.color"=testing_Y,testing_X),col.names=c("Kernel.color",testing_X)))
  ret2 <- c(invlogit(tail(model.matrix(Kernel.color~.,data=new_dat),1) %*% logistf_mod$coefficients))
  total_mat_logistf_test[i,1] <- i
  total_mat_logistf_test[i,4] <- sum((iffelse(ret2>=0.5,1,0) == testing_Y) & (testing_Y == 1)) #true positive
  total_mat_logistf_test[i,5] <- sum((iffelse(ret2>=0.5,1,0) == testing_Y) & (testing_Y == 0)) #true negative
  total_mat_logistf_test[i,6] <- sum((iffelse(ret2>=0.5,1,0) != testing_Y) & (testing_Y == 0)) #false positive
  total_mat_logistf_test[i,7] <- sum((iffelse(ret2>=0.5,1,0) != testing_Y) & (testing_Y == 1)) #false negative
  total_mat_logistf_test[i,8] <- sum(total_mat_logistf_test[i,4:7])
  total_mat_logistf_test[i,]
}
proc.time()-start_time
total_mat_logistf_test <- do.call(rbind,ret)

```

#OLS

```

ret <- foreach(i=1:n_sample) %dopar% {
  training <- dat[-i,]
  testing_X <- dat[i,-y_idx,drop=FALSE]
  testing_Y <- dat[i,y_idx]
  lm_mod <- lm(Kernel.color~.,data=training, family="binomial")
  ret2 <- predict(lm_mod, newdata= testing_X, type="response")
  total_mat_linear_test[i,1] <- i
  total_mat_linear_test[i,4] <- sum((iffelse(ret2>=0.5,1,0) == testing_Y) & (testing_Y == 1)) #true positive
  total_mat_linear_test[i,5] <- sum((iffelse(ret2>=0.5,1,0) == testing_Y) & (testing_Y == 0)) #true negative
  total_mat_linear_test[i,6] <- sum((iffelse(ret2>=0.5,1,0) != testing_Y) & (testing_Y == 0)) #false positive

```



```

total_mat_linear_test[i,7] <- sum((ifelse(ret2>=0.5,1,0) != testing_Y) & (testing_Y == 1)) #false neg
total_mat_linear_test[i,8] <- sum(total_mat_linear_test[i,4:7])
total_mat_linear_test[i,]
}
total_mat_linear_test <- do.call(rbind,ret)

sum(colSums(total_mat_glmdr_test[,4:7])[1:2]) / sum(colSums(total_mat_glmdr_test[,4:7])) # glmdr
sum(colSums(total_mat_bayes_test[,4:7])[1:2]) / sum(colSums(total_mat_bayes_test[,4:7])) # bayesglm
sum(colSums(total_mat_logistf_test[,4:7])[1:2]) / sum(colSums(total_mat_logistf_test[,4:7])) # logistf
sum(colSums(total_mat_linear_test[,4:7])[1:2]) / sum(colSums(total_mat_linear_test[,4:7])) # linear

```