Generalized Linear Models and Logistic Regression

Rob McCulloch

1. Introduction
2. Exponential Family
3. Generalized Linear Models
4. Newton's Method for GLMs
5. Logistic Regression
6. L2 and L1 Regularized GLM's
7. Simulated Example
8. We8There
9. Multinomial Logit

1. Introduction

We have studied approaches for using linear models.

Our approaches involved designing paths from simple models to complex models and the use of out-of-sample predictive performance to choose a point on the path.

In linear models "simple" meant started with with a large number of potential predictor variables and the "regularizing" the fit by shrinking coefficients to 0. The Lasso allowed us to shrink all the way to 0.

In this section we extend these ideas to generalizations of the linear model.

We will look at *generalized linear models*, or *GLMs*, which is a fundamental framework for extending linear modeling to non-numeric responses.

We will focus on logistic regression which is the GLM for a binary Y.

We will use iterative optimization, in particular Newton's method, to fit these models.

2. Exponential Family

Consider a single numeric response Y.

Suppose

$$P(Y \in A) = \int_A e^{\theta y - c(\theta)} \, dF(y)$$

$$\begin{aligned} P(Y \in A) &= \int_A e^{\theta y - c(\theta)} \, f(y) \, dy \quad y \text{ continuous} \\ &= \sum_{y \in A} e^{\theta y - c(\theta)} \, p(y) \quad y \text{ discrete} \end{aligned}$$

Then, the distribution of Y is of the *exponential family* form with *natural parameter* θ.

Example:

Suppose $Y \sim N(\theta, 1)$.

Then

$$\begin{aligned} p(y|\theta) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\theta)^2} \\ &= e^{\theta y - \frac{1}{2}\theta^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} \end{aligned}$$

$f(y)$ is the standard normal density and $c(\theta) = \frac{1}{2}\theta^2$.

Example:

$Y \sim Bern(p)$.

$$\begin{aligned} p(y) &= p^y (1-p)^{(1-y)}, \quad y \in \{0,1\} \\ &= \left(\frac{p}{1-p}\right)^y (1-p) \\ &= e^{y \log(\frac{p}{1-p})} e^{-\log(\frac{1}{1-p})} \end{aligned}$$

$$\theta = \log(\tfrac{p}{1-p}) \Rightarrow p = \frac{e^\theta}{1+e^\theta}, \quad 1-p = \frac{1}{1+e^\theta}.$$

$$p(y) = e^{y\,\theta - log(1+e^\theta)}$$

Note: θ is the *log of the odds ratio*.

Moments and Exponential Family:

$$e^{c(\theta)} = \int e^{\theta y} \, dF(y)$$

Now diffentiate both sides:

$$e^{c(\theta)} c'(\theta) = \int y \, e^{\theta y} \, dF(y)$$

Now multiply both sides by $e^{-c(\theta)}$

$$c'(\theta) = \int y \, e^{\theta y - c(\theta)} \, dF(y)$$

So,

$$c'(\theta) = E(Y|\theta).$$

$$e^{c(\theta)} c'(\theta) = \int y \, e^{\theta y} \, dF(y)$$

Now diffentiate both sides:

$$c''(\theta) e^{c(\theta)} + e^{c(\theta)} [c'(\theta)]^2 = \int y^2 \, e^{\theta y} \, dF(y)$$

Now multiply both sides by $e^{-c(\theta)}$

$$c''(\theta) + [E(Y|\theta)]^2 = \int y^2 \, e^{\theta y - c(\theta)} \, dF(y)$$

$$c''(\theta) = E(Y^2|\theta) - [E(Y|\theta)]^2 = Var(Y|\theta).$$

Example:

Suppose $Y \sim N(\theta, 1)$.

Then $c(\theta) = \frac{1}{2}\theta^2$.

$c'(\theta) = \theta$.

$c''(\theta) = 1$.

Example:

$Y \sim Bern(p)$.

$c(\theta) = \log(1 + e^{\theta})$.

$c'(\theta) = \frac{e^{\theta}}{1+e^{\theta}} = p$

$c''(\theta) = \frac{(1+e^{\theta})e^{\theta} - (e^{\theta})^2}{(1+e^{\theta})^2}$

$= p - p^2$.
$= p(1 - p)$.

Likelihood for Exponential Family:

$$Y_i \sim e^{y\theta - c(\theta)} f(y), \quad i = 1, 2, \ldots, n, \quad iid$$

$$\begin{aligned} p(y_1, y_2, \ldots, y_n \mid \theta) &= \prod_{i=1}^{n} e^{y_i\theta - c(\theta)} f(y_i) \\ &= e^{(\sum y_i)\theta - nc(\theta)} \prod f(y_i) \end{aligned}$$

Note that $\sum y_i$ is a *sufficient statistic*.

All you need to know about $(y_1, y_2, \ldots, y_n)$ to compute the likelihood is $S = \sum y_i$!

MLE for Exponential Family:

The likeklihood for Y_i iid from an exponential family is

$$L(\theta) \propto e^{S\,\theta - n\,c(\theta)}, \text{ where }, S = \sum y_i.$$

So the log likelihood is

$$logL(\theta) = S\,\theta - n\,c(\theta) + \text{constant}.$$

The FOC is:

$$S = n\,c'(\theta), \text{ or }, c'(\theta) = \frac{S}{n} = \bar{Y}.$$

So,

$$\hat{\theta} = (c')^{-1}(\bar{Y}).$$

Like a moment estimator: find the θ which makes the mean $E(Y|\theta)$ equal the observed sample mean.

3. Generalized Linear Models

Exponential family gives us a class of models for

$$p(y|\theta).$$

In a "regression context" where we want the distribution of Y to depend on x we can let

$$\theta = g(x).$$

If we want x to effect y only through a linear function of x we can let

$$\theta = h(x'\beta)$$

where now h is just a function $R \to R$.

In general, given exponential family with natural parameter θ we have $\mu = c'(\theta)$, let $\eta = x'\beta$, and then specify a *link function* g such that

$$g(\mu) = \eta,$$

or

$$c'(\theta) = g^{-1}(x'\beta).$$

We will keep things simple and only consider the canonical link

$$\eta = \theta = x'\beta,$$

but the general case is not too much harder.

Example:

$$Y \sim N(\mu, 1), \quad \theta = \mu = \eta = x'\beta.$$

(note: for $Y \sim N(\mu, \sigma^2)$ we have to extend our treatment to include another parameter).

Example:

$Y \sim Bern(p)$

$\theta = log(p/(1-p)) = x'\beta$

$p = \frac{e^\eta}{1+e^\eta}, \quad \theta = \eta = x'\beta.$

this is exactly logistic regression !!

4. Newton's Method for GLMs

We want to compute the MLE (maximize the likelihood).

We will develop the standard approach for a GLM with canonical link.

This is just an application of Newton's method.

Newton's Method:

Suppose we want to minimize $g(x)$ where $x \in R^k$.

The method is iterative.

Start at x:

- Do a second order Taylor approximation to g around x.
- Minimize the approximate function. Since it is quadratic, this can be done in closed form with basic matrix operations.
- update x to be the new minimum.
- Repeat until converge.

$x \in R^k$.

$g : R^k \to R$.

g' is the gradient, the row vector of partial derivatives $\frac{\partial g(x)}{\partial x_j}$.

g'' is the symmetric matrix of cross derivatives (the Hessian) $\frac{\partial^2 g}{\partial x_j \, \partial x_i}$.

$g(x) \approx g(x_o) + g'(x_o)(x - x_o) + \frac{1}{2}(x - x_o)' g''(x_o)(x - x_o)$.

FOC applied to approximation:

$g'(x_o) + (x - x_o)' g''(x_o) = 0$.

$(x - x_o)' = -g'(x_o) g''(x_o)^{-1}$.

$x^* = x_o - g''(x_o)^{-1} g'(x_o)'$.

Newton iteration:
$x_o \to x_o - g''(x_o)^{-1} g'(x_o)'$.

Note:

A symmetric matrix A is positive definite if $x'Ax > 0, \forall\, x$.

Note:

If we are maximizing or minimizing we do the same thing!!

But, if we are minimizing we would like g'' to be positive definite.

If it is pd eveywhere, then g is convex and we know a convex function has a global minimum.

If we are maximizing we want a negative definite Hessian corresponding to a concave function.

GLM MLE:

Let's minimize the negative of the log likelihood using Newton's method.

We need to compute the first and second derivatives.

The Model:

$$p(y|x,\beta) \propto e^{yx^T\beta - c(x^T\beta)}.$$

The Likelihood

Given data $(x_i, y_i), i = 1, 2, \ldots, n$,

$$\theta_i = x_i^T \beta.$$

The likelihood is proportional to:

$$\prod_{i=1}^{n} e^{y_i (x_i^T \beta) - c(x_i^T \beta)}$$

So if L denotes the log-likelihood

$$-L(\beta) = \sum_{i=1}^{n} [c(x_i^T \beta) - y_i (x_i^T \beta)]$$

Derivatives for $c(x'\beta)$:

$$\frac{\partial}{\partial \beta_i} = c'(x'\beta)x_i.$$

$$\frac{\partial^2}{\partial \beta_i \partial \beta_j} = c''(x'\beta)x_i x_j.$$

$$\nabla c = c'(x'\beta)x' = m(x'\beta)x'.$$

$$H = c''(x'\beta)xx' = V(x'\beta)xx'.$$

The Gradient and the Hessian

$-L = \sum_{i=1}^{n} c(x_i' \ beta) - y_i(x_i'\beta)$

$$\begin{aligned} -\nabla L &= \sum m(x_i'\beta)x_i' - y_i x_i' \\ &= \sum x_i'(m(x_i'\beta) - y_i) \\ &= (m - y)'X \end{aligned}$$

$$\begin{aligned} H &= \sum V(x_i'\beta)x_i x_i' \\ &= X'VX \end{aligned}$$

where $V = diag(V(x_i'\beta))$.

Newton's Method for a GLM

$$\begin{aligned}\beta_{i+1} &= \beta_i - [X^T V X]^{-1}[X^T(m-y)] \\ &= \beta_i + [X^T V X]^{-1}[X^T(y-m)]\end{aligned}$$

Weighted Least Squares

Model:

$$Y_i = x_i'\beta + \epsilon_i, \quad \epsilon_i \sim N(0, 1/w_i).$$

If w_i is big, the variance of the error is small which means that observation has a lot of *information*.

$W = diag(w_i)$ is the matrix with $W_{ii} = w_i$ and $W_{ij} = 0, \ i \neq j$.

$p(y \mid X, \beta, w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} w_i^{1/2} e^{-\frac{1}{2} w_i (y_i - x_i'\beta)^2}$.

$-2logL = \sum w_i (y_i - x_i'\beta)^2 + c$.

$$\min_{\beta} \sum w_i (y_i - x_i'\beta)^2$$

$$\begin{aligned} \nabla &= -2 \sum w_i (y_i - x_i'\beta) x_i' \\ &= (-2X'W(y - X\beta))' \end{aligned}$$

FOC:

$$X'Wy = X'WX\beta$$

$$\hat{\beta} = (X'WX)^{-1} X'Wy.$$

IRLS

If we construct the "working vector"

$$z = X\beta_i + V^{-1}(y - m).$$

Then, with $W = V$ we have

$$(X'VX)^{-1}X'Vz = [(X'VX)^{-1}X'V][X\beta_i + V^{-1}(y - m)] = \beta_i + (X'VX)^{-1}X'(y - m).$$

So, the Newton update is obtained by doing a weighted regression of z on X.

IRLS: "Iteratively Reweighted Least Squares".

5. Logistic Regression

Let's look at a simple example of logistic regression.

```
> library(ISLR)
> data(Default)
> attach(Default)
> head(Default)
  default student   balance    income
1      No      No  729.5265 44361.625
2      No     Yes  817.1804 12106.135
3      No      No 1073.5492 31767.139
4      No      No  529.2506 35704.494
5      No      No  785.6559 38463.496
6      No     Yes  919.5885  7491.559
```

Can you predict who will default give characteristics of the account?

Multiple Logistic Regression

The logistic model:

- Step 1:
  $\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_p x_p$.
- Step 2:
  $P(Y = 1 \mid x = (x_1, x_2, \ldots, x_p)) = F(\eta)$.

$$F(\eta) = \frac{e^{\eta}}{1 + e^{\eta}}.$$

Or, in one step, our model is:

$$Y_i \sim \text{Bernoulli}(p_i), \quad p_i = F(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots \beta_p x_{ip}).$$

Our first step keeps some of the structure we are used to in linear regression.

We combine the x's together into one weighted sum that we hope will capture all the information they provide about y.

We then turn the combination into a probability by applying F.

Our parameter vector is $(\beta_0, \beta_1, \ldots, \beta_p)$.

The Default Data, More than One x

Here is the logistic regression output using all three x's in the data set: balance, income, and student.
student is coded up as a factor, so R automatically turns it into a dummy.

```
Call:
glm(formula = default ~ balance + student + income, family = binomial,
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4691  -0.1418  -0.0557  -0.0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
income       3.033e-06  8.203e-06   0.370  0.71152
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

Note:

```
Number of Fisher Scoring iterations: 8
```

It took 8 iterations for the maximization to converge !!!

The estimates are MLE.

Confidence intervals are estimate +/- 2 standard errors.
e.g for `studentYes` coefficient : -.65 +/- 2(.24) = -.65 +/- .5

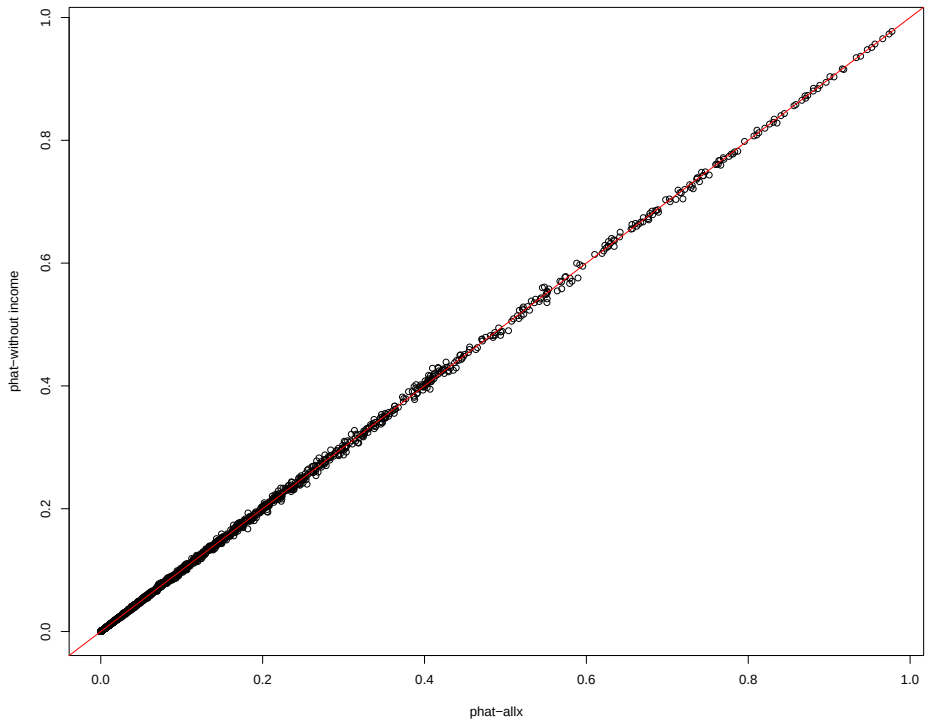
Z-stats are (estimate-proposed)/se.

To test whether the coefficient for income is 0, we have z = (3.033-0)/8.203 = .37, so we fail to reject.

The p-value is 2*P(Z < -.37) = 2*pnorm(-.37) = 0.7113825.

So, the output suggests we may not need `income`.

Here is a plot of the fitted probabilities with and without `income` in the model.



We get almost the same probabilities, so, as a practical matter, income does not change the fit.

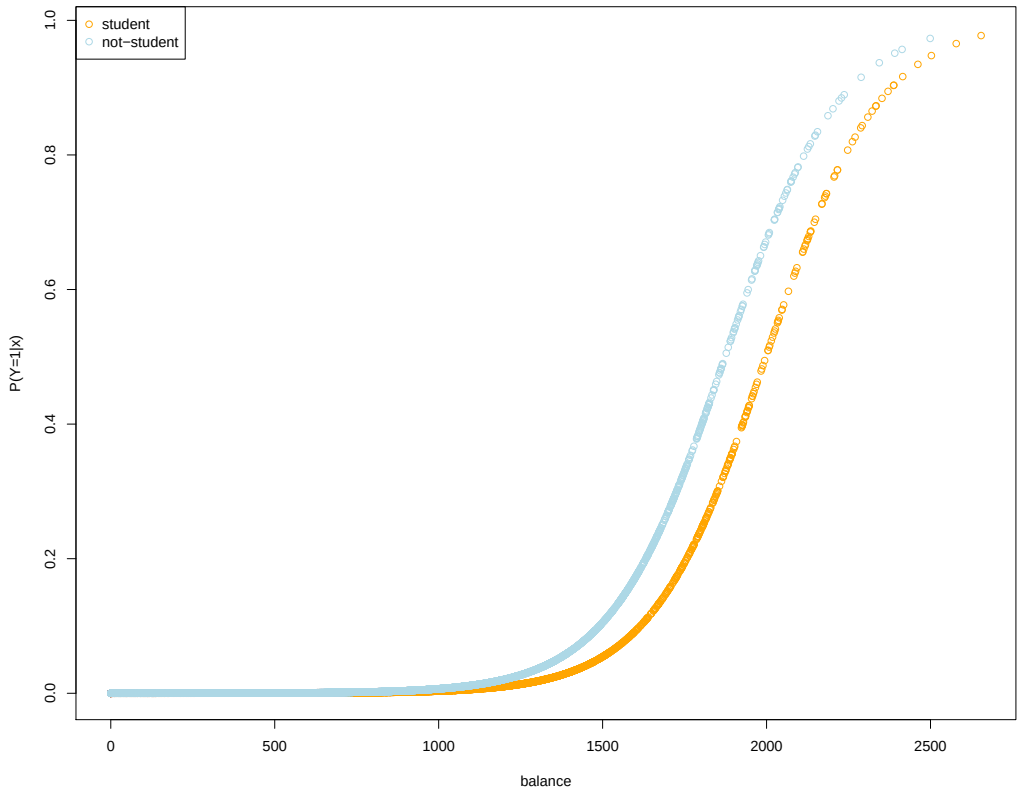
Here is the output using balance and student.

```
Call:
glm(formula = default ~ balance + student, family = binomial,
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4578  -0.1422  -0.0559  -0.0203   3.7435

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.075e+01  3.692e-01 -29.116  < 2e-16 ***
balance      5.738e-03  2.318e-04  24.750  < 2e-16 ***
studentYes  -7.149e-01  1.475e-01  -4.846 1.26e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.7  on 9997  degrees of freedom
AIC: 1577.7

Number of Fisher Scoring iterations: 8
```

With just `balance` and `student` in the model, we can plot $P(Y = 1 \mid x)$ vs. x.

The orange points are for the students and the blue are for the non-students.

In both cases the probability of default increases with the balance, but at any fixed balance, a student is less likely to default.



AIC and BIC in Logistic Regression

In the logistic regression model, the *deviance* is just -2 times the logLikelihood (usually evaluated at the mle).

$$L(\beta) = \prod_{i=1}^{n} P(Y = y_i | X = x_i, \beta)$$

For logistic regression, $P(Y = 1|x, \beta) = F(x'\beta)$, $F(\eta) = \frac{\exp(\eta)}{(1+\exp(\eta))}$.

Given an estimate (usually a MLE) $\hat{\beta}$,

$$deviance = -2 \sum_{i=1}^{n} log(P(Y = y_i | X = x_i, \hat{\beta}))$$

The better the fit of the model, the bigger the likelihood, the smaller the deviance.

AIC for the Default example:

A parameter (a coefficient) costs 2.

- balance:
  Residual deviance: 1596.5, AIC: 1600.5 = 1593.5+2*(2).
- balance + student + income:
  Residual deviance: 1571.5, AIC: 1579.5 = 1571.5 +2*(4).
- balance + student:
  Residual deviance: 1571.7, AIC: 1577.7 = 1571.7 + 2*(3).
- student:
  Residual deviance: 2908.7, AIC: 2912.7 = 2908.7 + 2*(2).

$\Rightarrow$ pick balance+student

BIC:

BIC is an alternative to AIC, but the penalty is different.

$$BIC = deviance + log(n) * (p + 1)$$

$\log(n)$ tends to be bigger than 2, so BIC has a bigger penalty, so it suggest smaller models than AIC.

BIC for the Default example:

$\log(10000) = 9.21034$.

A parameter (a coefficient) costs 9.2.

- balance:
  1596.5, BIC: $= 1593.5 + 9.2*(2) = 1611.9$.
- balance + student + income:
  BIC: $= 1571.5 + 9.2*(4) = 1608.3$.
- balance + student:
  BIC: $= 1571.7 + 9.2*(3) = 1599.3$.
- student:
  BIC: $= 2908.7 + 9.2*(2) = 2927.1$.

$\Rightarrow$ pick balance+student

Which is better, AIC or BIC??

nobody knows.

`R` prints out AIC, which suggests you might want to use it, but a lot of people like the fact that BIC suggests simpler models.

A lot of academic papers report both AIC and BIC and if they pick the same model are happy with that. Lame.

Checking the out of sample performance is safer !!!

6. L2 and L1 Regularized GLM's

For linear models, we found that regularized versions gave us a very nice way to explore the bias-variance tradeoff.

We added a penalty term whose weighting parameter λ controlled the extent to which the coefficients were shrunk towards 0.

We would like to be able to do the same thing with GLMs.

We want *regularized logistic regression*.

L2:

L2 is easy since we can just add the penalty in and then adjust our gradient and Hessian.

$$\underset{\beta}{\text{minimize}} - L(\beta) + \frac{\lambda}{2} \sum \beta_j^2$$

This can be done by Newton's method where we just add $\lambda \beta_j$ to the gradient and λI to the Hessian.

In the GLM case we may want to include the intercept but make no adjustment to the gradient and Hessian part corresponding to the intercept so that it is not shrunk.

L1:

Note: this follows section 16.5 of Efron and Hastie closely.

Now we have both a non quadratic loss *and* a non differentiable penalty.

$$\underset{\beta}{\text{minimize}} - \frac{1}{n} L(\beta) + \lambda \sum |\beta_j|$$

Again, the intercept is in L but not in the penalty.

λ Grid:

As in the linear case we know that

$$\lambda_{\max} = \max_j | < x_j, y - \bar{y}\mathbb{1} > |.$$

Recall that will all the slopes equal to 0, we are just back in simple exponential family and the mle of the mean is just the sample mean. For logistic regression this will just be the sample fractions of ones since $Y \in \{0, 1\}$.

A grid of values equally spaced on the log scale starting at $\lambda_{\max}$ and going down to a small $\varepsilon\lambda_{\max}$ can be used.

Coordinate Descent:

- For each value of λ cycle throught the β_j finding the optimal value conditional on the others. Cycle through until convergence.
- Start at λ_{max} and then decrease λ by going down the grid of values.
  As λ decrease more β_j will become active.
- At each λ value, start the cyclic iteration at values guessed from the solution obtained from the previous (slightly larger) λ.

Proximal Newton Approach:

To optimize over β_j given the rest, first approximate the minus log likelihood so that it can be cast as a weighted least squares problem.

Then our problem looks like:

$$\min_{\beta_j} \frac{1}{2n} \sum w_i \left(z_i - \beta_0 - x_i^T \beta\right)^2 + \lambda \sum |\beta_k|.$$

As in IRLS, the z is the working vector and the w are the V computed at the current β.

With $r_i = z_i - \beta_0 - \sum_{k \neq j} x_{ik}\beta_k$ our problem is:

$$\min_{\beta_j} \frac{1}{2n} \sum w_i \left(r_i - x_{ij}\beta_j\right)^2 + \lambda \left|\beta_j\right|.$$

$$\min_{\beta_j} \frac{1}{2n} \sum w_i \left(r_i - x_{ij}\beta_j\right)^2 + \lambda \left|\beta_j\right|.$$

This is problem is easily solved using the usual soft threholding type of solution. For example we can rewrite it as:

$$\min_{\beta_j} \frac{1}{2n} \sum \left(\sqrt{w_i}\, r_i - \sqrt{w_i}\, x_{ij}\beta_j\right)^2 + \lambda \left|\beta_j\right|.$$

and then it looks exactly like our linear one x problem with $y_i = \sqrt{w_i}\, r_i$ and $x_i = \sqrt{w_i}\, x_{ij}$.

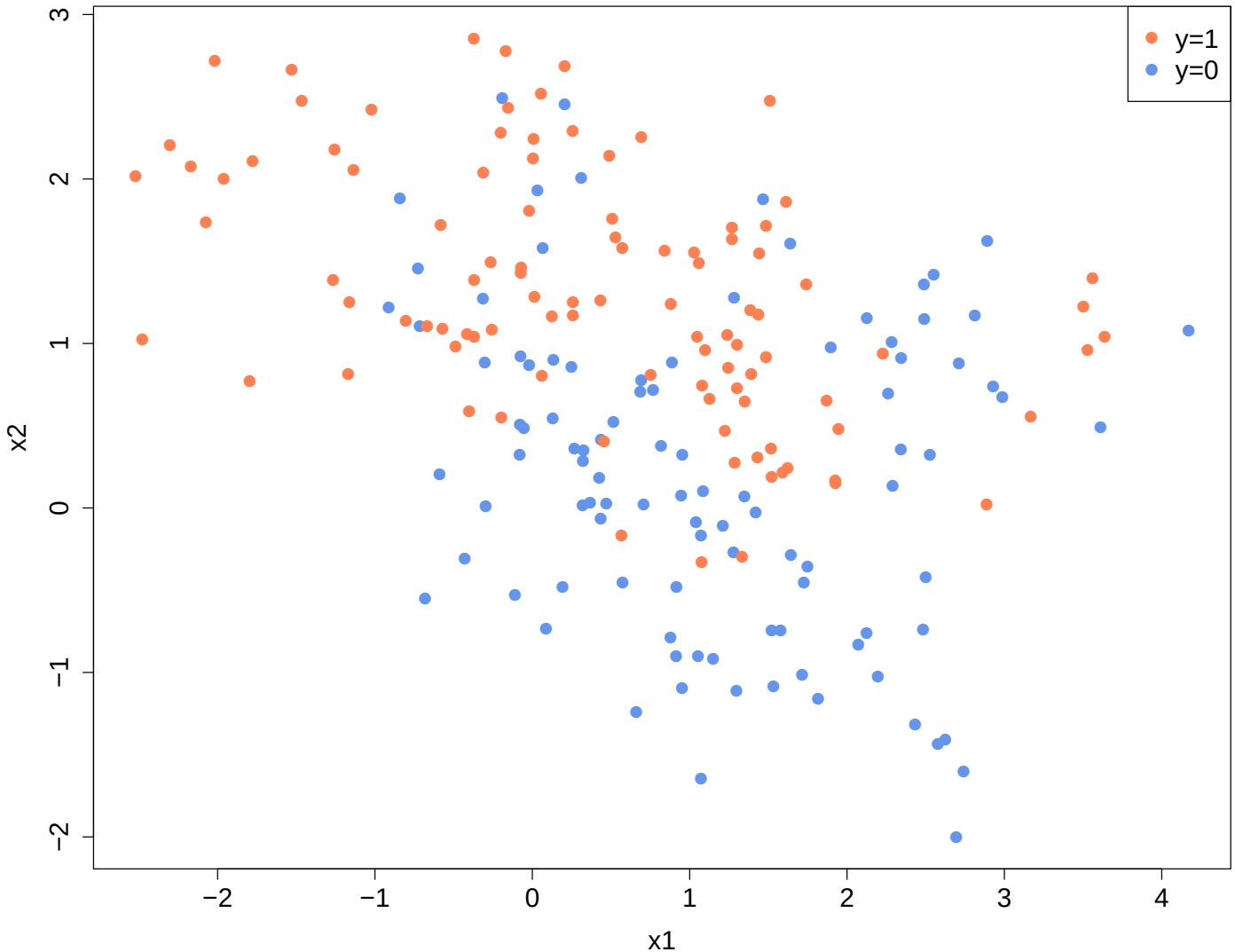
7. Simulated Example

Let's look at a simulated example from the "Elements of Statistical Learning" (`library(ElemStatLearn)`) R package.

> *Description*
> *This is a simulated mixture example with 200 instances and two classes. 100 members in each class.*

We have two numeric x's creatively called x_1 and x_2 and one binary outcome y.

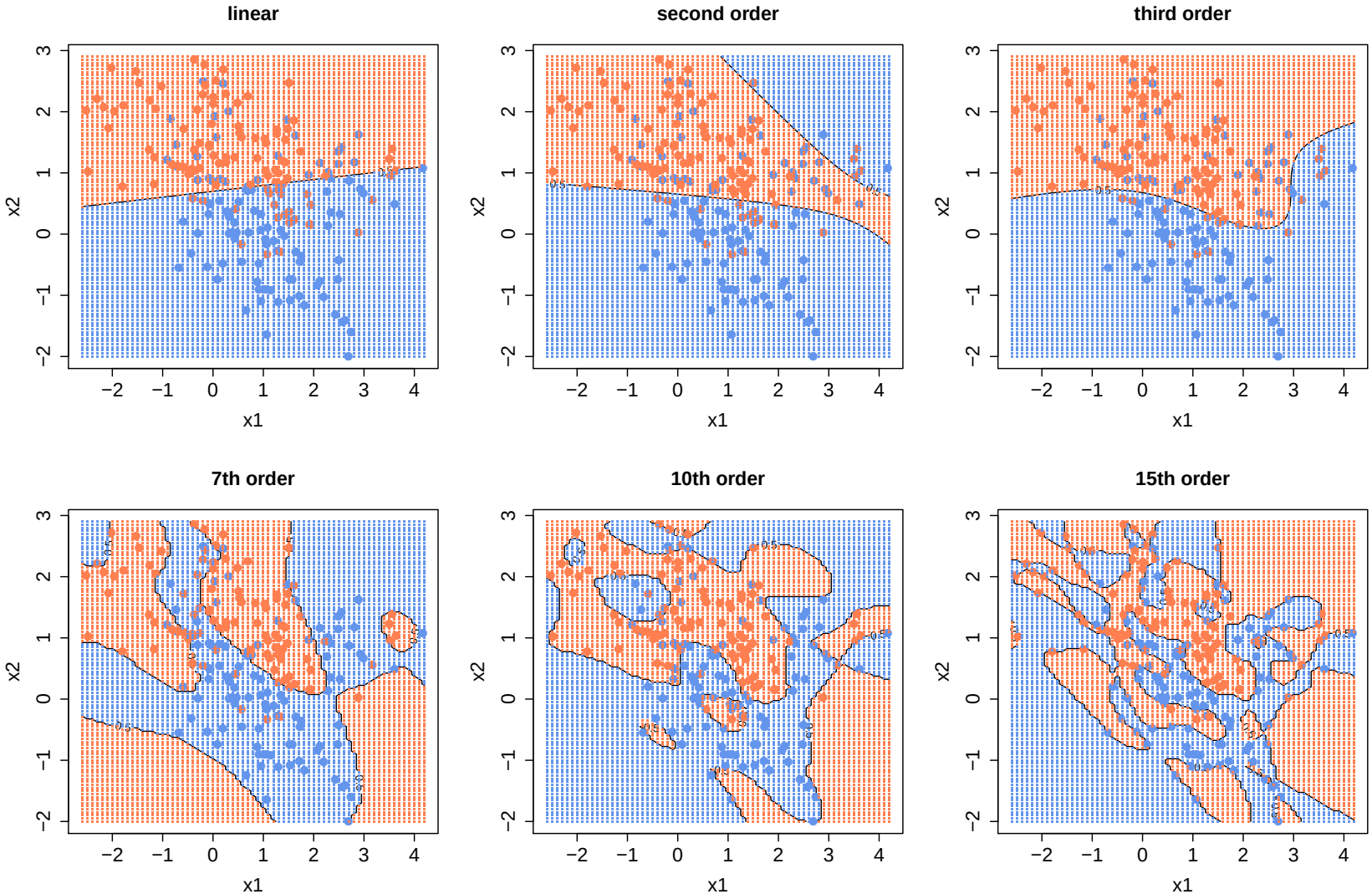
Plot the data: two numeric x and a binary y.



Now let's run logistic regression so y on x, throwing in more and more polynomials in x_1 and x_2.

We use the R function `poly` to construct all the polynomials.

Decision Boundary plots of logit fits with varying degrees of polynomial terms thrown in.



which one do you like the best ???

Ok, now let's try the Lasso.

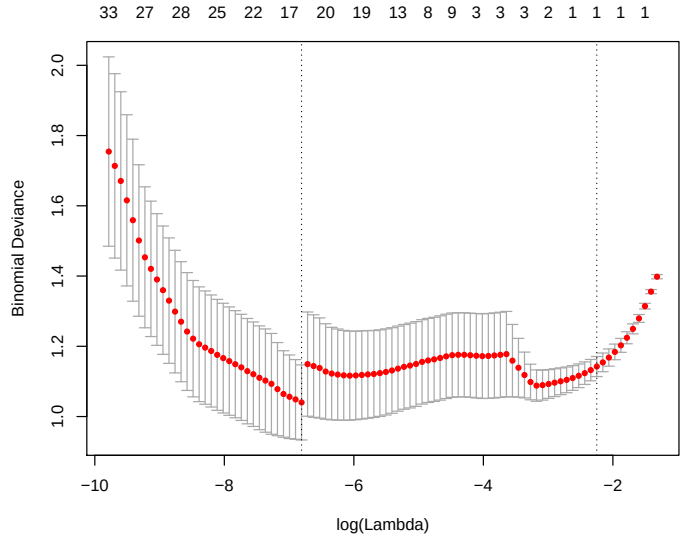
We will throw in all the 15th order terms.

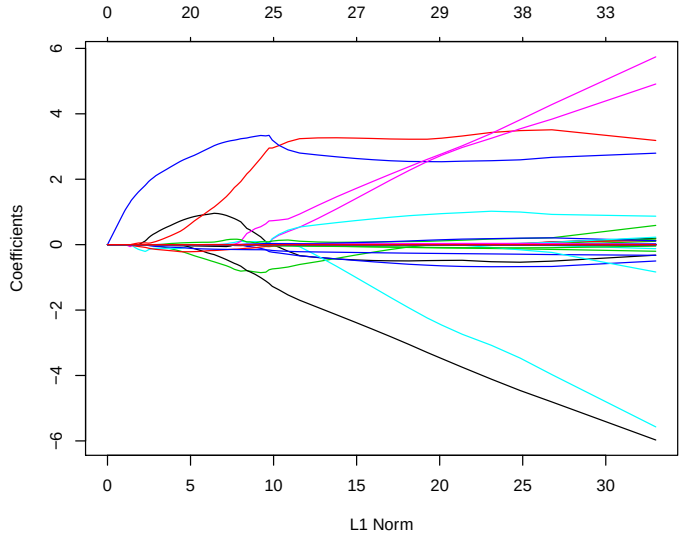
This gives us 135 x's !!!!!!!

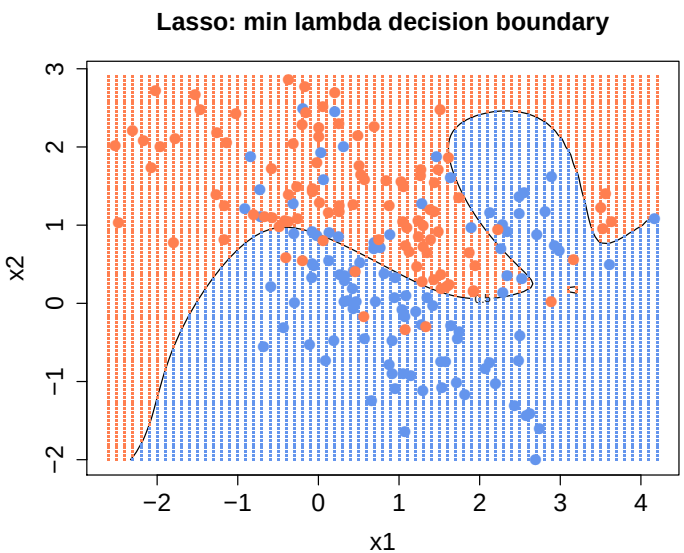
```
> 15*14/2 + 2*15
[1] 135
```

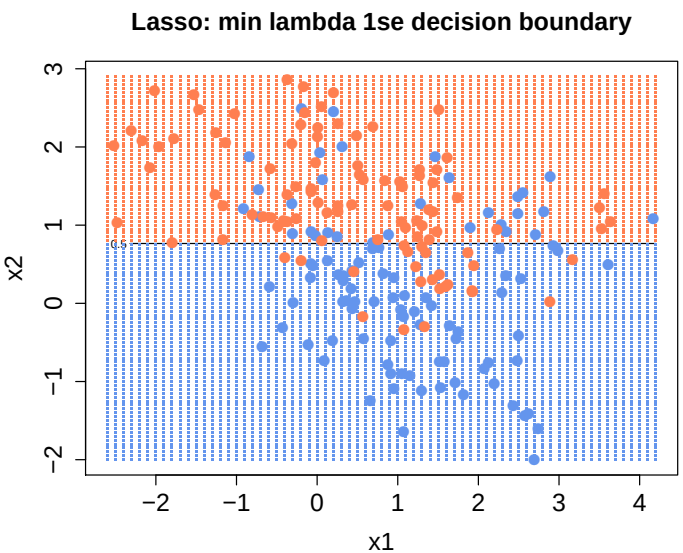
Of course, the fit using all of these is overfit.

But, can we use the Lasso to explore the possibility that some subset of the 135 is good?









Both of these look reasonable.

We got to try using all 135 x's, but the cv process pulls us back to something more restrained.

What transformations did the Lasso choose?

Here are the non-zero coefficients from lambda.min.

```
  (Intercept)         x.1.0         x.2.0         x.3.0         x.7.0
-2.495723e+00  9.241824e-01  1.000302e+00 -4.584982e-01  1.136215e-03
        x.8.0        x.15.0         x.0.1         x.3.1         x.2.2
 1.362927e-04 -3.150041e-08  2.922533e+00 -4.609735e-02 -2.478831e-01
        x.3.2         x.8.2         x.1.3         x.7.3         x.8.3
-1.885774e-01  3.671665e-04  7.296183e-02  1.149181e-04  1.348425e-04
        x.0.4         x.2.7        x.0.15
-1.329917e-01  4.745585e-03  1.255572e-06
```

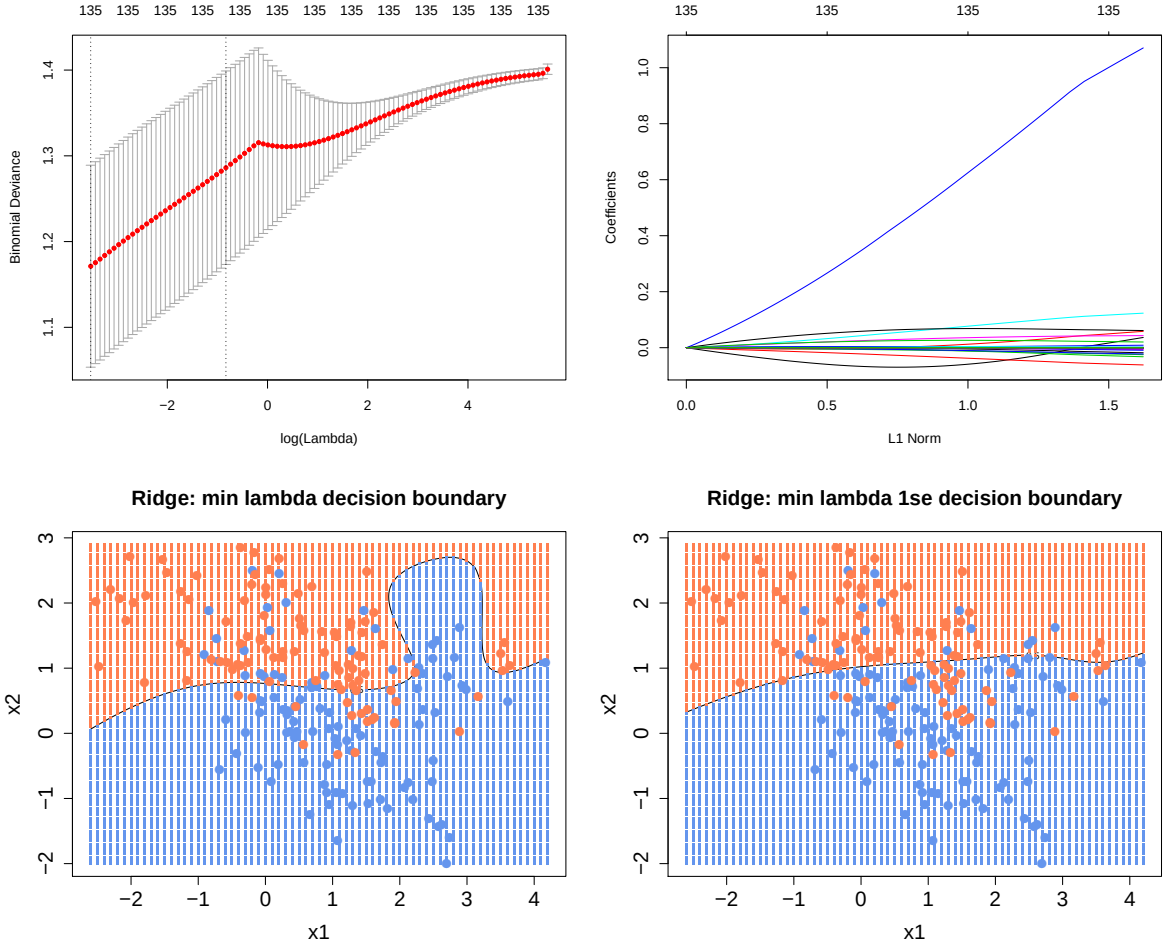
Wow, $x_1^8 \times x_3^2$ comes in, not obvious!!

Here are the non-zero coefficients from lambda.1se.

```
(Intercept)       x.0.1
 -0.5352522   0.6994122
```

And the simple model is just use x_2 which is clearly a reasonable choice give the plot of the data.

Here are the results for Ridge regression.



What x do you think the big coefficient is for?

Note that the default choice of λ grid does not give us the bottom of the U.

Here are the largest absolute values of the Ridge coefficients at lambda.min.

```
      x.0.1 (Intercept)       x.1.1       x.0.2       x.1.2       x.2.0
1.029880622 0.829649432 0.119507976 0.062205441 0.059446543 0.053133363
      x.2.1       x.2.2       x.1.0       x.3.0       x.3.2       x.0.3
0.042759327 0.030234643 0.024597247 0.023377217 0.021276924 0.021120811
      x.3.1       x.0.4       x.1.3       x.3.3       x.3.4       x.2.3
0.016565883 0.008242392 0.008133289 0.006167556 0.002953760 0.002925370
      x.0.5       x.4.2
0.002846589 0.002766778
```

So, x_2 (same as `x.0.1`) dominates the fit.

8. We8There

Each observation consists of a restaurant review.

We also have the Zagat ratings for each restaurant.

We want to see how the Zagat rating relates to reviews.

What kind of customer review correspond to a good rating?

We have 6,166 observations.

Here are the Zagat ratings summarized:

```
   1    2    3    4    5
 615  493  638 1293 3127
```

We will dichotomize the outcome by making y 1 if the overall rating is greater than 3 and 0 otherwise.

```
y
   0    1
1746 4420
```

What is our x?

We want to use the information in the text of the restaurant reviews.

How do we convert the text of a review into a numeric x??

Bag of Words:

The most common way to look at a text document is to treat it as a bag of words.

That is, we look all the words in all the documents and then make a list of words or "terms" of interest.

For example the word "the" will probably occur with some frequency, but that may not be deemed to be of interest.

Given the list of terms, we just count the number of times each term occurs in the text of the document.

Note that this approach ignores the *word order*, the document is treated as a "bag of words".

bigrams:

A bi-gram is just a pair of words appearing in sequence.

For example, the pair of words "good food" appearing in sequence may mean more than "good" or "food" by themselves.

We will make a list of bigrams and then x will the number of times each bigram occurs in the text of a review.

The dimension of our X matrix is:

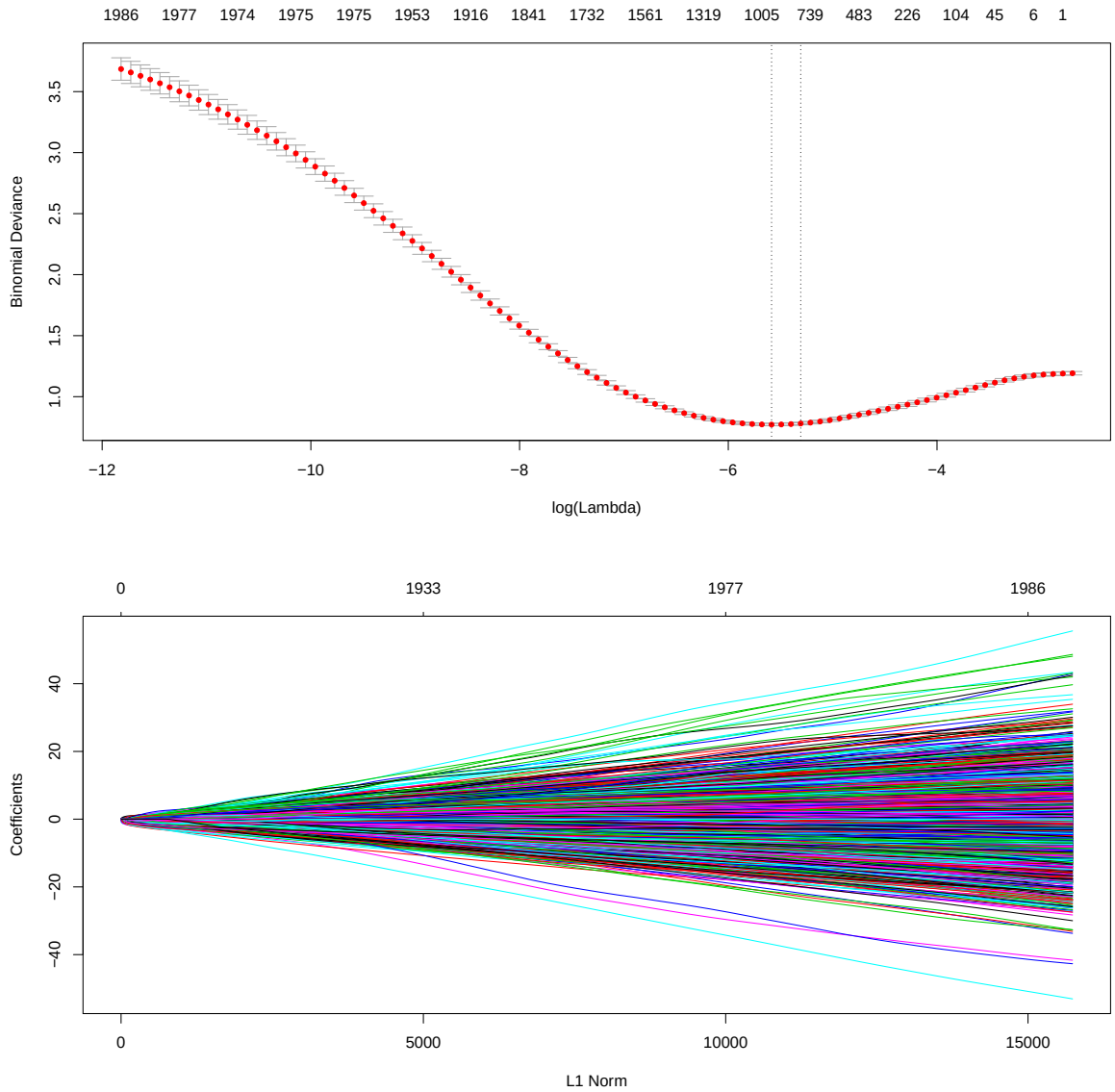
```
[1] 6166 2640
```

There are 2,640 bigrams in our list of ones we are counting.

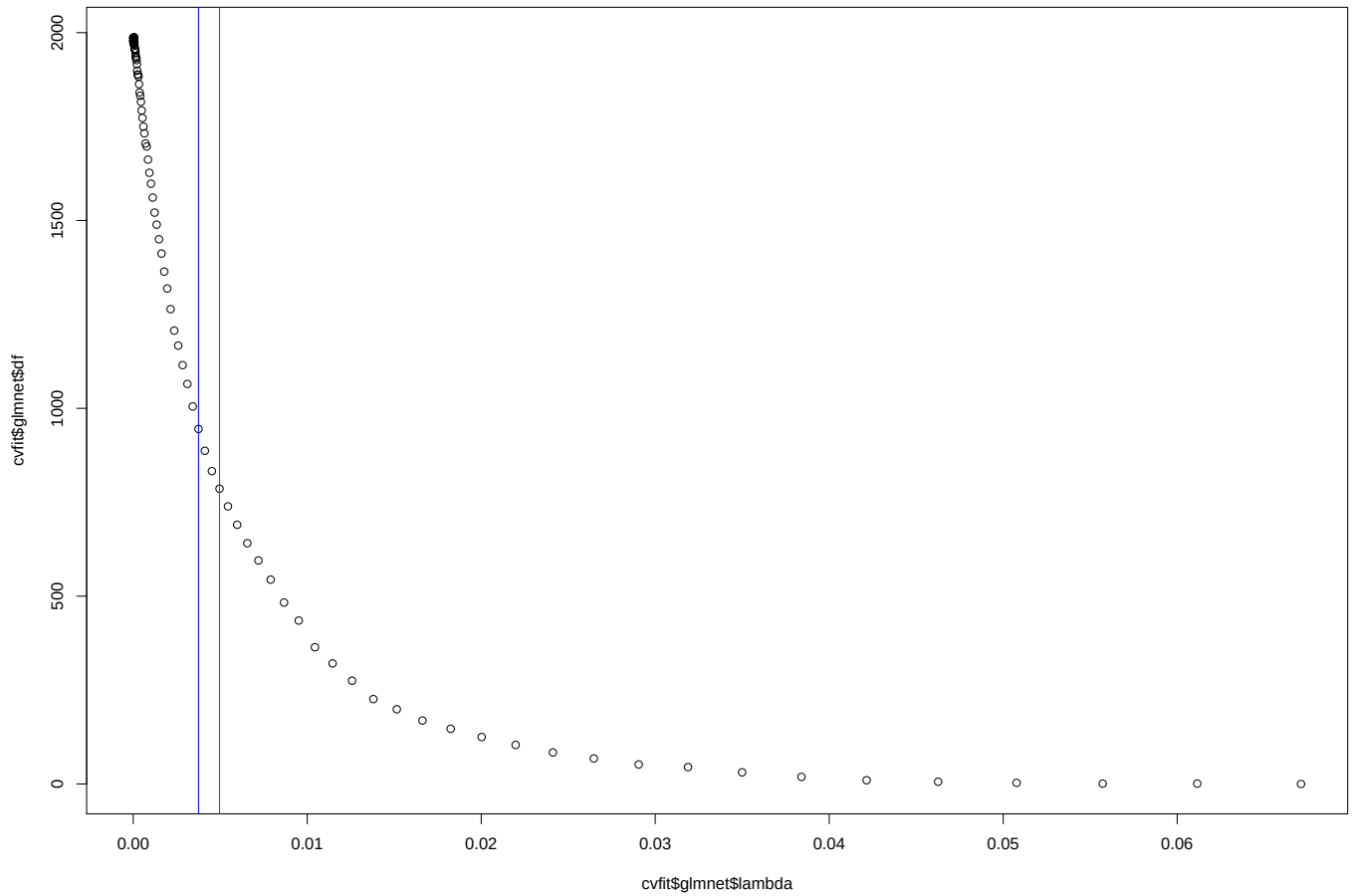
In our first observation, the bigrams with a non-zero count are:

```
 even though larg portion  mouth water     red sauc    babi back     back rib
           1             1            1            1            1            1
chocol mouss veri satisfi
           1             1
```

Here is the Lasso fit:



Here is the number of non-zero coefficients plotted against λ.



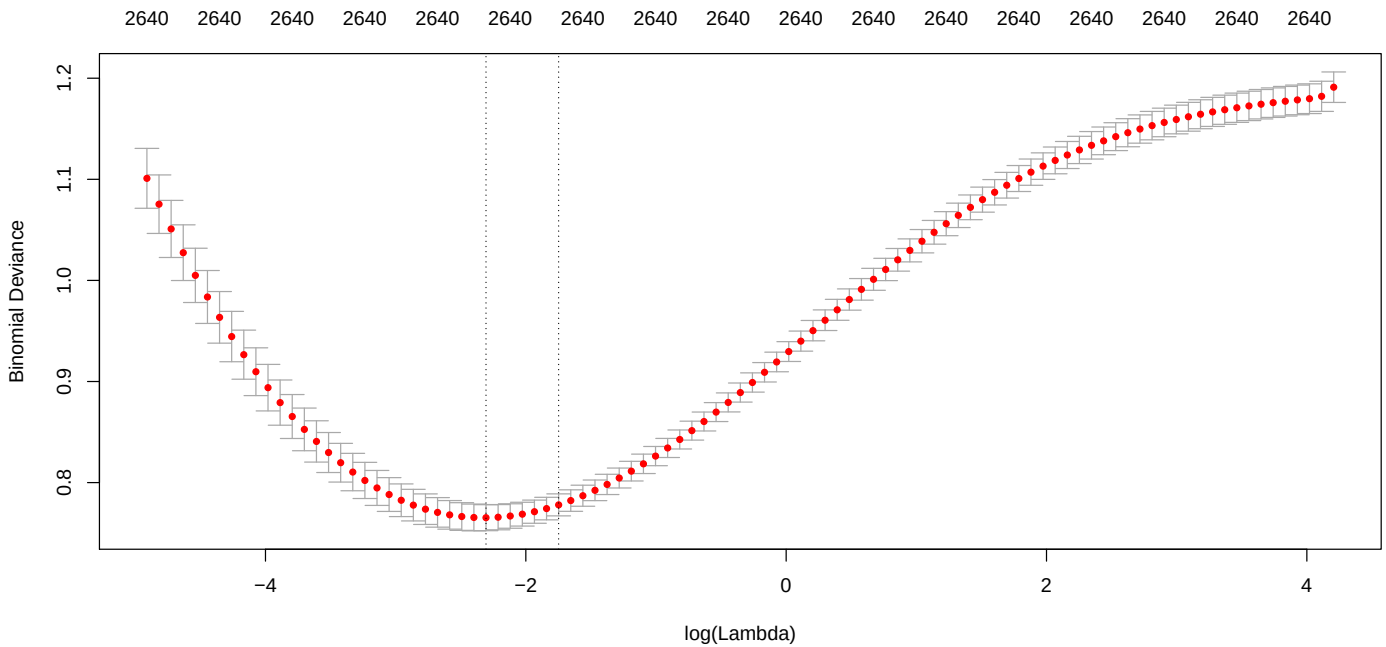
Lines drawn at `lambda.min` and `lambda.1se`.

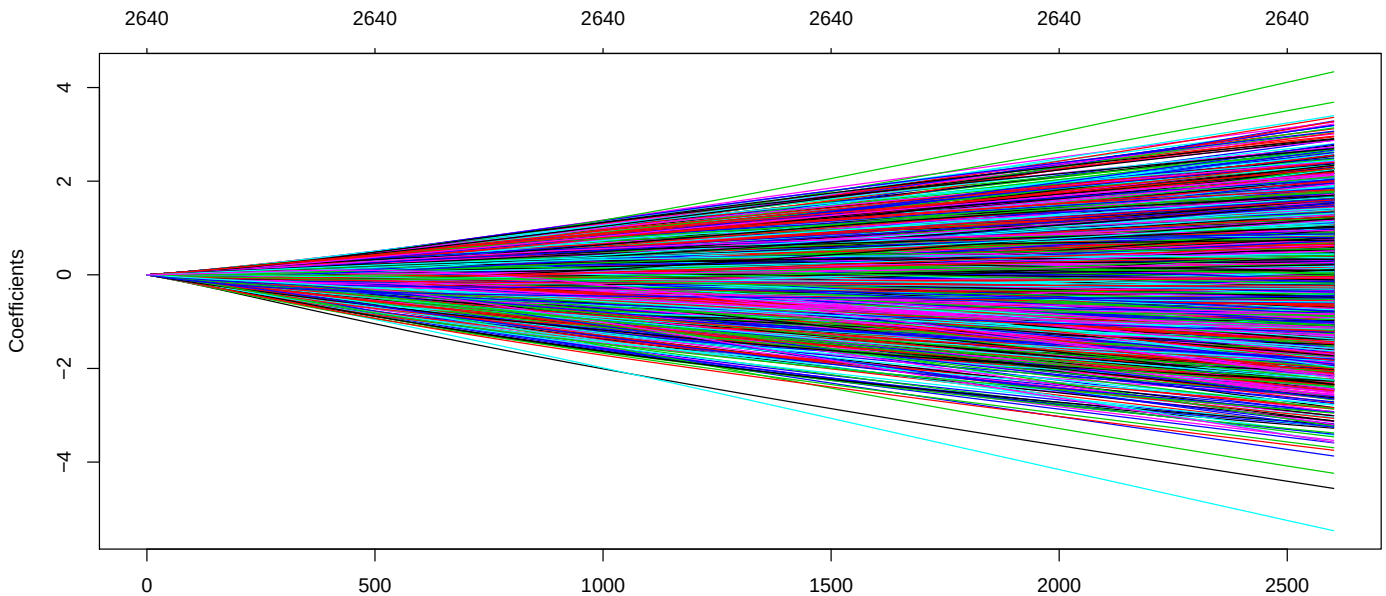
Here are the big positive and big negative coefficients.

```
Big positive coefficients:
 [1] "between two"       "can wait"          "beef sandwich"
 [4] "friend help"       "best meal"         "high recommend"
 [7] "cannot wait"       "melt mouth"        "food delici"
[10] "portion huge"      "wonder experi"     "veri uniqu"
[13] "pretti quick"      "can beat"          "full bar"
[16] "look out"          "absolut best"      "definit recommend"
[19] "(Intercept)"       "now live"          "quit few"
[22] "well worth"        "better most"       "never bad"
[25] "out world"         "great food"        "wall cover"
[28] "up good"           "great experi"      "absolut delici"
 [1] 1.6235757 1.3860122 1.3672099 1.3413956 1.2976254 1.2863832 1.1719763
 [8] 1.1502664 1.1288480 1.1032885 1.0835135 1.0807977 1.0533985 1.0492868
[15] 1.0435863 1.0429624 1.0308216 0.9933000 0.9842953 0.9840951 0.9834227
[22] 0.9633459 0.9578412 0.9512289 0.9375038 0.9236589 0.9104929 0.9073509
[29] 0.9026439 0.8981764


Big negative coefficients:
 [1] "over cook"       "waitress seem"   "just ok"         "bad food"
 [5] "food poison"     "servic ok"       "servic slow"     "mediocr food"
 [9] "one worst"       "wast money"      "food okay"       "mani option"
[13] "anoth chanc"     "veri disappoint" "food averag"     "terribl servic"
[17] "veri slow"       "veri poor"       "veri bland"      "quick lunch"
[21] "never go"        "gone down"       "servic terribl"  "food terribl"
[25] "never return"    "stay away"       "far better"      "mediocr best"
[29] "veri rude"       "extrem rude"
 [1] -1.484979 -1.501437 -1.526967 -1.528732 -1.540570 -1.569690 -1.585480
 [8] -1.591799 -1.642964 -1.652540 -1.672776 -1.673099 -1.685418 -1.704577
[15] -1.752020 -1.770758 -1.780688 -1.792626 -1.803219 -1.831518 -1.899799
[22] -2.025952 -2.035374 -2.044341 -2.087831 -2.097159 -2.267862 -2.277634
[29] -2.322275 -2.524030
```

Not surprising that it is bad to be “extrem rude”.

Here are Ridge results.





```
Big positive coefficients:
 [1] "(Intercept)"    "coconut shrimp" "look out"       "veri uniqu"
 [5] "year food"      "servic attent"  "friend help"    "well food"
 [9] "everyth menu"   "half shell"     "veri cozi"      "absolut best"
[13] "absolut delici" "serv hot"       "staff make"     "same peopl"
[17] "fair price"     "can beat"       "between two"    "excel price"
[21] "salad great"    "cannot wait"    "restaur make"   "time re"
[25] "back home"      "best meal"      "food superb"    "thorough enjoy"
[29] "reserv suggest" "keep go"
 [1] 0.9126494 0.6568706 0.6541469 0.6532986 0.6495907 0.6485409 0.6415978
 [8] 0.6413095 0.6396719 0.6378453 0.6257804 0.6232927 0.6146492 0.6085780
[15] 0.6063171 0.6015257 0.6008904 0.5989794 0.5986393 0.5957714 0.5942530
[22] 0.5889521 0.5856139 0.5836239 0.5766060 0.5764588 0.5733635 0.5724580
[29] 0.5715484 0.5708517


Big negative coefficients:
 [1] "one worst"       "anoth chanc"     "just anoth"      "day befor"
 [5] "veri disappoint" "quick lunch"     "food mediocr"    "veri poor"
 [9] "wast money"      "servic slow"     "terribl servic"  "servic ok"
[13] "got bill"        "stay away"       "food terribl"    "mani option"
[17] "veri limit"      "servic terribl"  "complain manag"  "veri bland"
[21] "just ok"         "mediocr food"    "food averag"     "veri rude"
[25] "food okay"       "veri slow"       "mediocr best"    "gone down"
[29] "far better"      "extrem rude"
 [1] -0.8566305 -0.8635339 -0.8699492 -0.8787698 -0.8881075 -0.8920423
 [7] -0.9201352 -0.9373039 -0.9438574 -0.9541782 -0.9581192 -0.9590981
[13] -0.9771998 -0.9841832 -0.9843469 -0.9920617 -1.0010190 -1.0014385
[19] -1.0104797 -1.0179680 -1.0222027 -1.0260030 -1.0503238 -1.0819638
[25] -1.0859973 -1.0939149 -1.1249573 -1.1382567 -1.2025776 -1.2955933
```

9. Multinomial Logit

The problem where Y is a binary outcome is very common.

But how do we extend logistic regression to the multinomial outcome case?

Let's look at the forensic glass data and use two of the x's (so we can plot) and all three of the outcomes.

Example, Forensic Glass:

Can you tell what kind of glass it was from measurements on the broken shards??

Y: glass type, 3 categories.

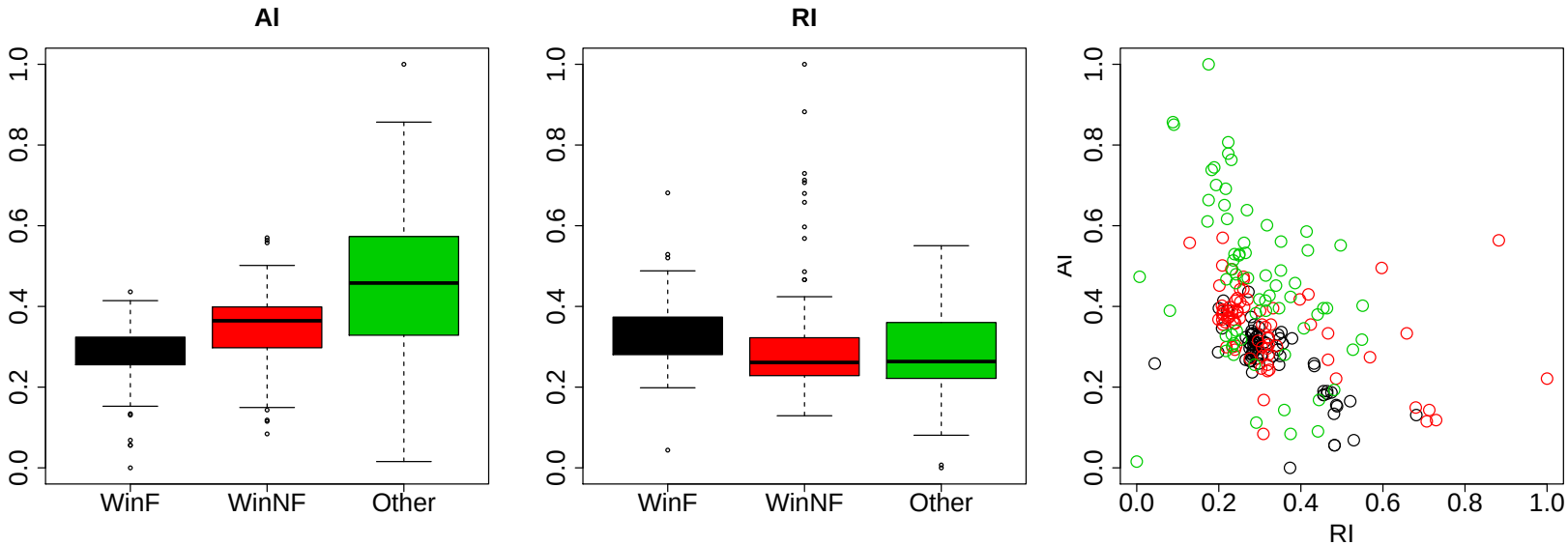
$Y \in S = \{\text{WinF}, \text{WinNF}, \text{Other}\}$.

WinF: float glass window
WinNF: non-float window
Other.

x: 3 numeric x's:
$x_1 = \text{RI}$: refractive index
$x_2 = \text{Al}$
$x_3 = \text{Na}$

Let's use two of the x's (so we can plot) and all three of the outcomes.

Here is the three outcome Y plotted agains the two x's x=(RI,AI).



kNN:

Before we go into the linear multinomial model, let's just note that KNN for classification is obvious.

Given test x and training (x_i, y_i):

Numeric Y:

- find the k training observations with x_i closest to x.
- predict y with the average of the y values for the neighbors.

Categorical Y:

- find the k training observations with x_i closest to x.
- predict Y with the most frequent of the y values for the neighbors.
- estimate $P(Y = y \mid x)$ with the proportion of neighbors having $Y = y$.

Multinomial Logit:

The multinomial logit model for $Y \in \{1, 2, \ldots, C\}$ is

$$P(Y = j|x) \propto \exp(x'\beta_j), \quad j = 1, 2, \ldots, C.$$

Or,

$$P(Y = j|x) = \frac{\exp(x'\beta_j)}{\sum_{j=1}^{C} \exp(x'\beta_j)}$$

So, each category gets a linear (affine) function of x !!!

Softmax Function:

For $x = (x_1, x_2, \ldots, x_C)$ The softmax function is

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_j e^{x_j}}$$

Exponentiate and then normalize.

Takes a vector of real numbers x_j and maps them to a probability vector.

We use this a lot.

Identification:

Suppose we add β to each β_j.

$$\begin{aligned} P(Y=j|x) &= \frac{\exp(x'(\beta_j+\beta)}{\sum \exp(x'(\beta_j+\beta))} \\ &= \frac{\exp(x'\beta)\exp(x'\beta_j)}{\exp(x'\beta)\sum \exp(x'\beta_j)} \\ &= \frac{\exp(x'\beta_j)}{\sum \exp(x'\beta_j)} \end{aligned}$$

So, if we add any vector to all the β_j we get the exact some model!!

In this case we say the the set of parameters $\{\beta_j\}$ is not identified in that two different sets can give you the exact same likelihood.

The common identification strategy is to pick one of the β_j and set it equal to 0. Usually, it is either the "first" or the "last" β_j.

Note that as usual x may be $(1, x_2, \ldots, x_p)'$, that is we have included an intercept.

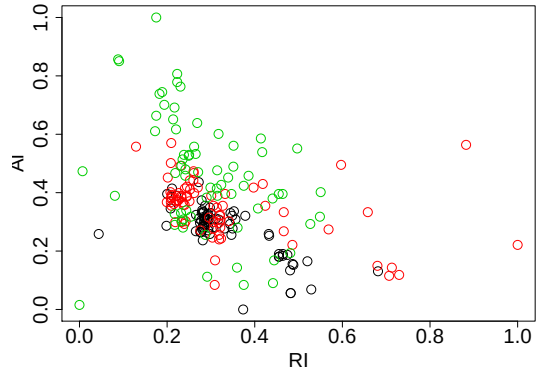
Here is ouput from fitting a multinomial logit using the forensic glass data.
(R package nnet, function multinom).

```
Call:
multinom(formula = y ~ ., data = ddf, maxit = 1000)

Coefficients:
      (Intercept)       RI        Al
WinNF   -3.277900 2.819056  7.861631
Other   -5.651027 2.718534 13.616921

Std. Errors:
      (Intercept)       RI       Al
WinNF    1.030785 1.610635 2.049922
Other    1.165932 1.872040 2.263372

Residual Deviance: 402.6627
AIC: 414.6627
```

The first β has been set to 0.

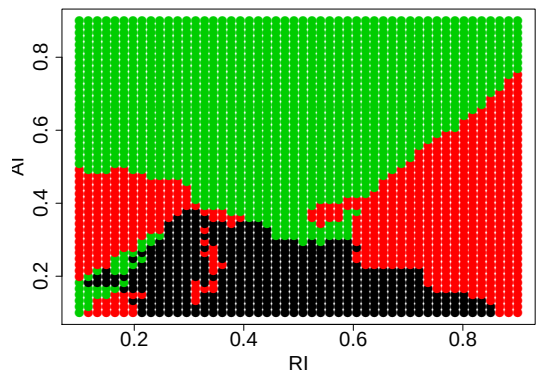
Note: 402.6627+12 = 414.6627

plot of the data.

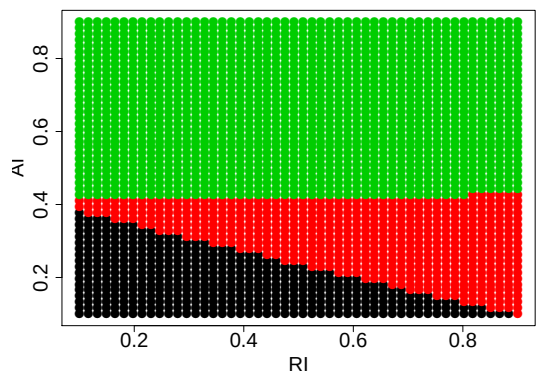


The color indicates the most probable outcome. black=WinF, red=WinNF, green=Other.

Most probable from knn. k=20.



Most probable from multinomial logit.



So, all but one class gets it's own coefficent vector.

```
Coefficients:
      (Intercept)       RI         Al
WinNF   -3.277900 2.819056  7.861631
Other   -5.651027 2.718534 13.616921

> table(y)
y
 WinF WinNF Other
   70    76    68
```

The coefficent vector for WinF has been set to 0.
$\beta_1 = (0.0, 0)$, for WinF
$\beta_2 = (-3.277900, 2.819056, 7.861631)$, for WinNF.
$\beta_3 = (-5.651027, 2.718534, 13.616921)$, for Other.

Let

$$\eta_1 = 0$$

$$\eta_2 = -3.28 + 2.82RI + 7.86AI$$

$$\eta_3 = -5.65 + 2.72RI + 13.62AI$$

$$P(Y = WinF = 1|x) = \frac{1}{1 + e^{\eta_2} + e^{\eta_3}}$$

$$P(Y = WinNF = 2|x) = \frac{e^{\eta_2}}{1 + e^{\eta_2} + e^{\eta_3}}$$

$$P(Y = Other = 3|x) = \frac{e^{\eta_3}}{1 + e^{\eta_2} + e^{\eta_3}}$$

When both AI and RI are small, the negative intercepts mean Y=WinF=1 is more likely.
When AI increases, Y=Other, becomes much more likely because of the large 13.62 coefficient.

$$P(Y = WinF = 1|x) = \frac{1}{1 + e^{\eta_2} + e^{\eta_3}}$$

$$P(Y = WinNF = 2|x) = \frac{e^{\eta_2}}{1 + e^{\eta_2} + e^{\eta_3}}$$

$$P(Y = Other = 3|x) = \frac{e^{\eta_3}}{1 + e^{\eta_2} + e^{\eta_3}}$$

Note

$$P(Y = i|x)/P(Y = j|x) = e^{\eta_i - \eta_j}$$

and the log odds is just $\eta_i - \eta_j$ with one of the η set to 0.

So the large difference in coefficients for AI (13.62-7.86) tells us the as AI increases the odds for 3=Other vs 2=WinNF will change quite a bit in favor of 3=Other.