Logistic Regression

1. Logistic Regression, One Predictor
2. Inference: Estimating the Parameters
3. Multiple Logistic Regression
4. AIC and BIC in Logistic Regression
5. Target Marketing: Tabloid Data
6. Log Odds
7. Multinomial Logit

1. Logistic Regression, One Predictor

To start off as simply as possible, we will first consider the case where we have a binary y and one numeric x.

Lets' look at the Default data:

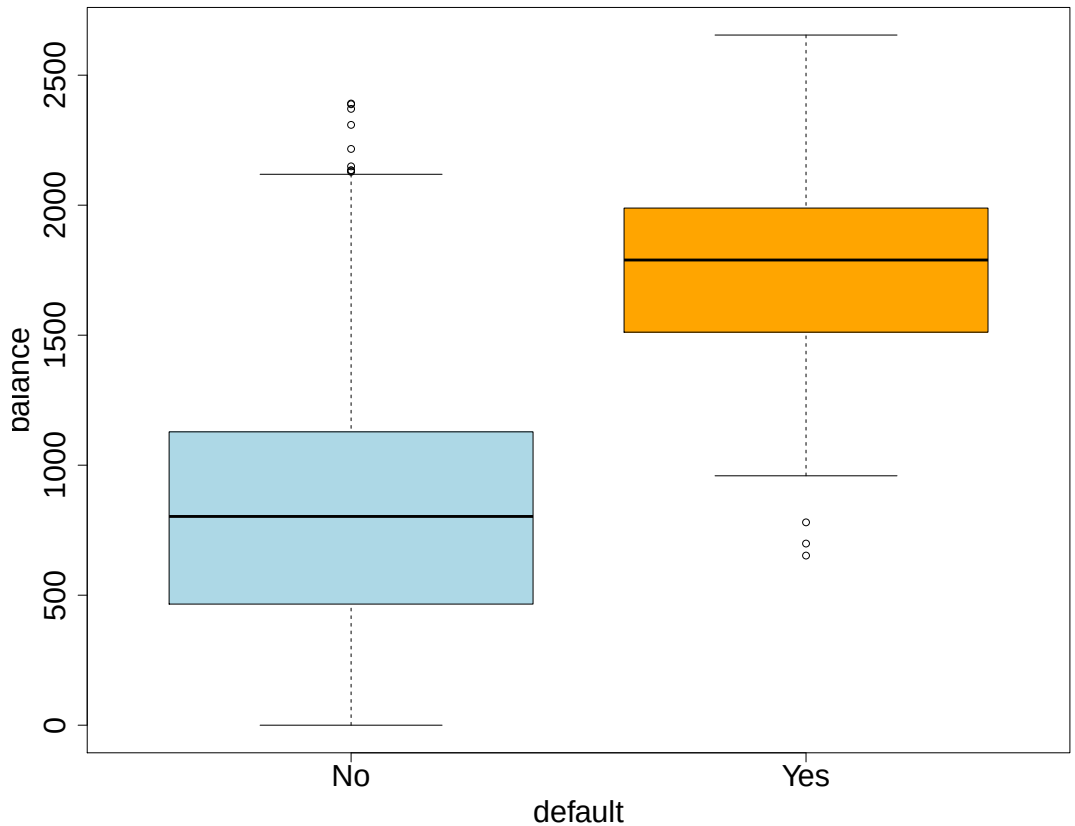
- y:
  whether or not a customer defaults on their credit card (No or Yes).
- x:
  The average balance that the customer has remaining on their credit card after making their monthly payment.
- 10,000 observations, 333 defaults (.0333 default rate).

Let's look at the data.

Divide the data into two groups, one group has y=No and other other group has y=Yes.

Use boxplots to display the x=balance values in each subgroup.

The balance values are bigger for the default y=Yes observations!



Logistic regression uses the power of linear modeling and estimates $Pr(Y = y \mid x)$ by using a two step process:

- Step 1:

  apply a linear function to x: $x \rightarrow \eta = \beta_0 + \beta_1 x$.

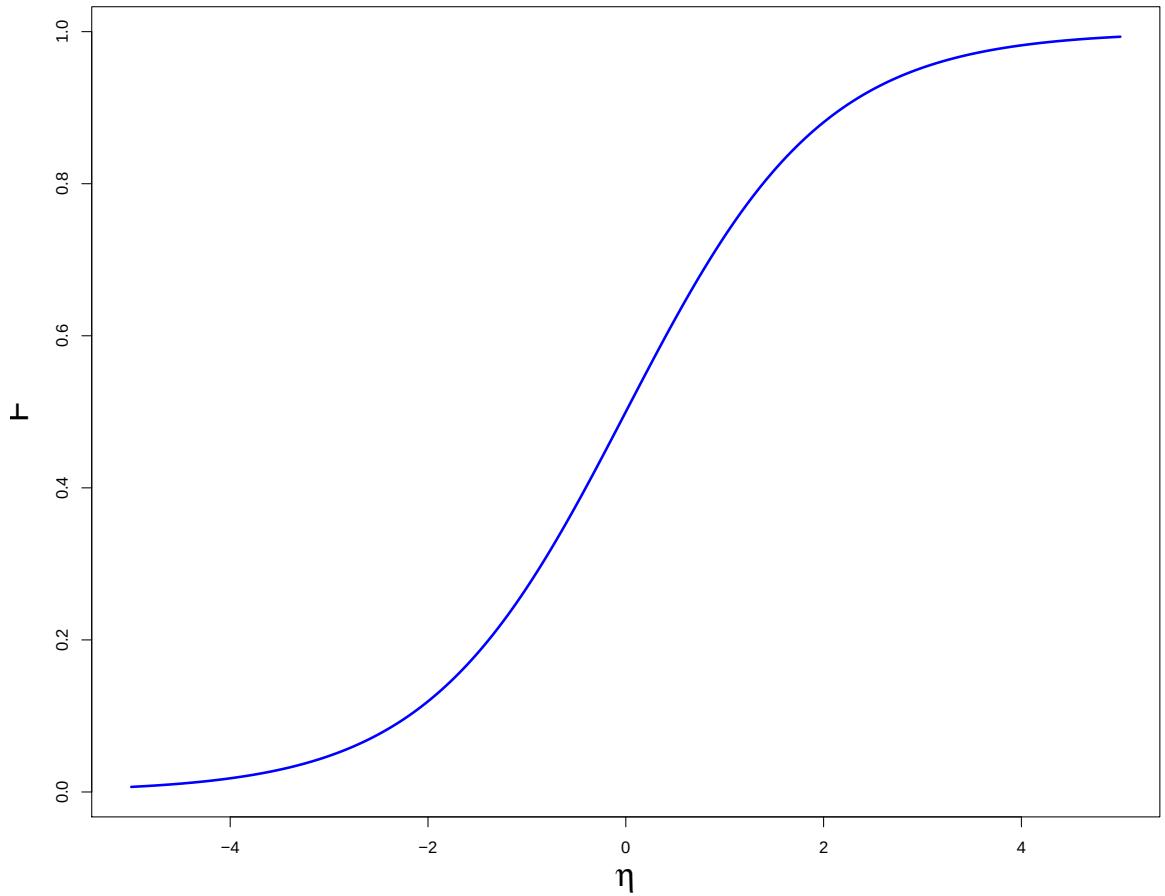
- Step 2:

  apply the *logistic function* F,
  to η to get a number between 0 and 1.

  $P(Y = 1 \mid x) = F(\eta)$.

The logistic function:

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



The key idea is that $F(\eta)$ is always between 0 and 1 so we can use it as a probability.

Note that F is increasing, so if η goes up $P(Y = 1 \mid x)$ goes up.

$$F(-3) = .05,\ F(-2) = .12,\ F(-1) = .27$$

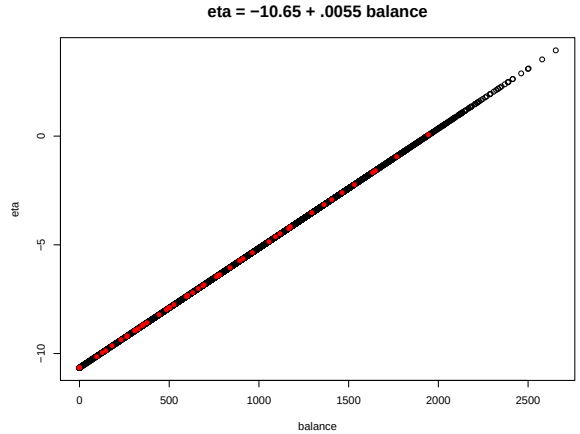
$$F(0) = .5$$

$$F(1) = .73,\ F(2) = .88,\ F(3) = .95$$

Logistic fit to the y=default, x=balance data.

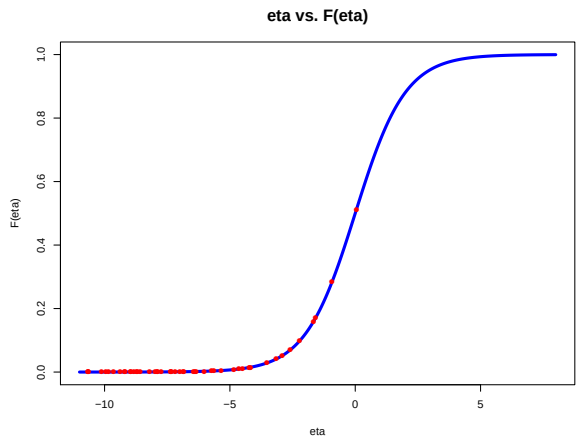
First, logistic looks for a linear function of x it can feed into the logistic function.

Here we have
$\eta = -10.65 + .0055\,x$.

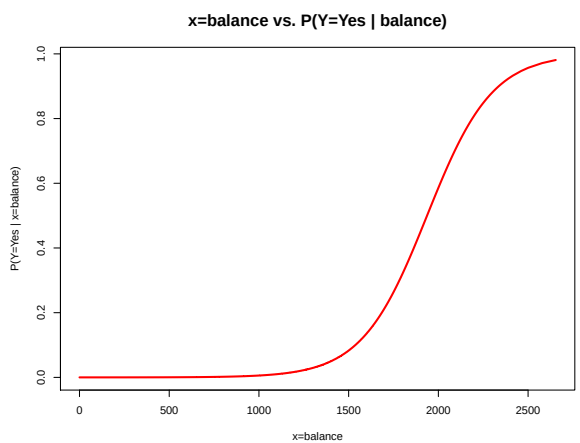


Next we feed the η values into the logistic function.

100 randomly sampled observations are plotted with red dots.



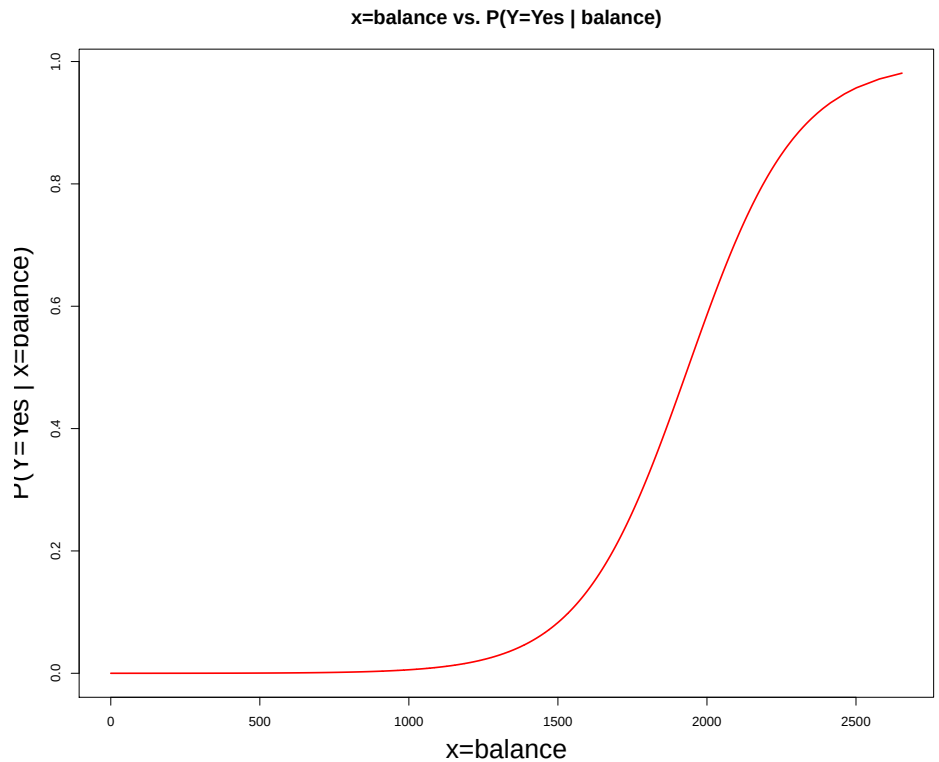
We can combine the two steps together and plot
x=balance vs.
$P(Y = Yes \mid x) = F(-10.65 + .0055\,x)$.



Logistic Regression:

Combining the two steps, our logistic regression model is:

$$P(Y = 1 \mid X = x) = F(\beta_0 + \beta_1 x).$$



2. Inference: Estimating the Parameters

Logistic regression gives us a formal parametric statistical model (like linear regression with normal errors).

Our model is:

$$Y_i \sim \text{Bernoulli}(p_i), \quad p_i = F(\beta_0 + \beta_1 x_i).$$

Our model has two parameters β_0 and β_1 which we can estimate given data.

We usually assume that given the parameters and the x_i, the Y_i are independent.

To estimate the parameters, we usually use *maximum likelihood*.

That is, we choose the parameter values that make the data we have seen most likely.

Let p_y be a simplified notation for $P(Y = y \mid x)$.
In the logistic model, p_y depends on (β_0, β_1)
(where we have "supressed" x).

$$p_y = p_y(\beta_0, \beta_1) = \begin{cases} P(Y = 1 \mid x) = F(\beta_0 + \beta_1 x) & Y = 1 \\ P(Y = 0 \mid x) = 1 - F(\beta_0 + \beta_1 x) & Y = 0 \end{cases}$$

For our logistic model, the probability of the $Y_i = y_i$ given x_i, $i = 1, 2, \ldots, n$ as a function of β_0 and β_1 is

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} p_{y_i}(\beta_0, \beta_1).$$

So, we estimate (β_0, β_1) by choosing the values that optimize the likelihood function $L(\beta_0, \beta_1)$!!

This optimization has to be done numerically using an iterative technique (Newton's Method!!).

The problem is convex and the optimization usually converges pretty fast.

Some fairly complex statistical theory gives us standard errors for our estimates from which we can get confidence intervals and hypothesis test for β_0 and β_1.

Here is the logistic regression output for our y=default, x= balance example.

The MLE of β_0 is $\hat{\beta}_0 = -10.65$.

The MLE of β_1 is $\hat{\beta}_1 = .0055$.

Given x=balance $= 2000$, $\eta = -10.65 + .0055 * 2000 = 0.35$

$\hat{\beta}_1 > 0$ suggests larger balances are associated with higher risk of default.

$P(default) =$
$P(Y = 1 \mid x = 2000) =$
exp(.35)/(1+exp(.35))
=0.59.

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <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: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
AIC: 1600.5

Number of Fisher Scoring iterations: 8
```

Confidence Interval for β_1:
$\hat{\beta}_1 \pm 2\text{se}(\hat{\beta}_1)$.
$.0055 \pm 2(.00022)$.

Test $H_0 : \beta_1 = \beta_1^0$
$z = \frac{\hat{\beta}_1 - \beta_1^0}{\text{se}(\hat{\beta}_1)}$.
If H_0 is true, z should look like a standard normal draw.

$\frac{.0055-0}{.00022} = 25$,
big for a standard normal
$\Rightarrow$
reject the null that $\beta_1 = 0$.

Similar for β_0.

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <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: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
AIC: 1600.5

Number of Fisher Scoring iterations: 8
```

Fisher Scoring iterations:

It took 8 iterations of the optimization for convergence.

Deviance:

The deviance is $-2log(L(\hat{\beta}_0, \hat{\beta}_1))$.
Twice -2 times the log of the maximized likelihood.

For numerical and theoretical reasons it turns out to be easier to work with the log of the likelihood than the likelihood itself.
Taking the log turns all the products into sums.

A big likelihood is good, so a small deviance is good.

Null and Residual Deviance:

The Residual deviance is just the one you get by plugging the MLE's of β_0 and β_1 into the likelihood.

The Null deviance is what you get by setting $\beta_1 = 0$ and then optimizing the likelihood over β_0 alone.

You can see that the deviance is a lot smaller when we don't restrict β_1 to be 0!!

Deviance as a sum of losses:

If we let

$$\hat{p}_y = p_y(\hat{\beta}_0, \hat{\beta}_1),$$

then the deviance is

$$\sum_{i=1}^{n} -2 \log(\hat{p}_{y_i}).$$

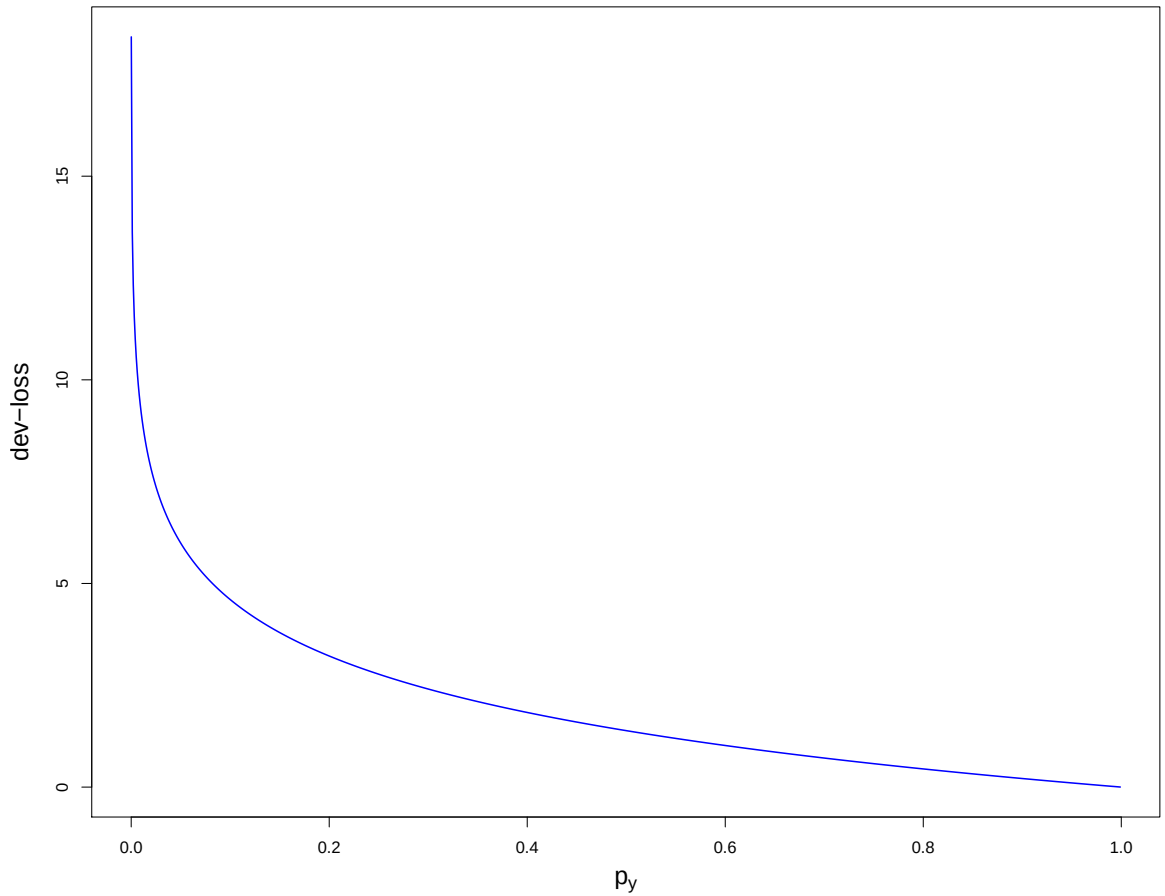
The sum over observations of -2 times the log of the estimated probability of the y you got.

p_y is the probability we assign to Y turning out to be y.
We can think of $-2\log(p_y)$ as our *loss*.

This is
p_y versus $-2\log(p_y)$.

When y happens, the bigger we said p_y is the better off we are, the lower our loss.

If y happens and we said p_y is small, we really get a big loss -that's fair!!



We have used deviance as an *out of sample* loss function, just as we have used sum of squared errors for a numeric Y.

Deviance In the Default-Balance Regression:

Let's look at the numbers we have been talking about in our x=balance, y=default logistic regression.

Here are the numbers printed out for observations 135-140 (out of 10,000).

We chose this range to look at because it includes one of the occasional ones.

	x=balance	eta	p(y=1|x)	y=default	py	deviance
135	1216.4515	-3.9622	0.0187	0	0.9813	0.0377
136	598.4614	-7.3604	0.0006	0	0.9994	0.0013
137	1486.9981	-2.4745	0.0777	1	0.0777	5.1106
138	943.7963	-5.4615	0.0042	0	0.9958	0.0085
139	0.0000	-10.6513	0.0000	0	1.0000	0.0000
140	996.2761	-5.1729	0.0056	0	0.9944	0.0113

For example, for i=135:

η = -10.65 + .0055*1216.4515 = -3.959517

```
    x=balance      eta p(y=1|x) y=default     py deviance
135 1216.4515  -3.9622   0.0187         0 0.9813   0.0377
136  598.4614  -7.3604   0.0006         0 0.9994   0.0013
137 1486.9981  -2.4745   0.0777         1 0.0777   5.1106
138  943.7963  -5.4615   0.0042         0 0.9958   0.0085
139    0.0000 -10.6513   0.0000         0 1.0000   0.0000
140  996.2761  -5.1729   0.0056         0 0.9944   0.0113
```

For i=135:
η = -10.65 + .0055*1216.4515 = -3.959517
$p(y = 1 \mid x)$ = exp(-3.96)/(1+exp(-3.96)) = 0.01870651.
$y = 0 \Rightarrow p_y = 1 - p(y = 1 \mid x) = 1 - 0.01870651 = 0.9812935$. deviance = -2*log(0.9812935) = 0.03776736.

For i=137, things are different because now y=1.
In this case $p_y = p(y = 1 \mid x)$.
Now the y that happened was given a smaller chance of happening (.077) so we have the much bigger deviance loss of -2*log(.0777) = 5.11.

The "residual deviance" in the R output is just the sum of all the deviance numbers for each observation = ...0377 + .0013 + 5.11 + ...

What is the "null deviance"?

If you did not know x (or if x had nothing to do with y) you would have to go with the marginal $P(Y = 1)$ which we could estimate to be the sample percentage: .0333.

Then the deviance should be -2log(.0333) for an observation which with y=1 and -2log(.9667) for an observation with y=0.

-2*(9667*log(.9667) + 333*log(.0333)) = 2920.65
(as in the R output).

3. Multiple Logistic Regression

We can extend our logistic model to several numeric x by letting η be a linear combination of the x's instead of just a linear function of one x:

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

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.

Inference is as in the $p = 1$ case discussed previously except now our likelihood will depend on $(\beta_0, \beta_1, \ldots, \beta_p)$ instead of just (β_0, β_1).

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
```

Everything is analogous to when we had one x.

The estimates are MLE.

Confidence intervals are estimate +/- 2 standard errors.

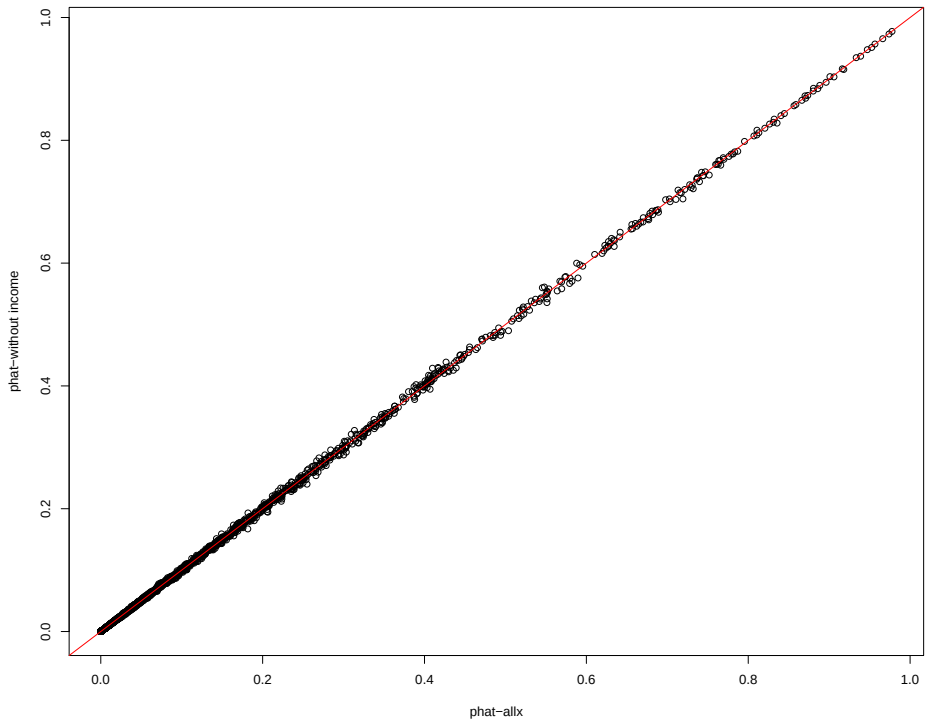
Z-stats are (estimate-proposed)/se.

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.

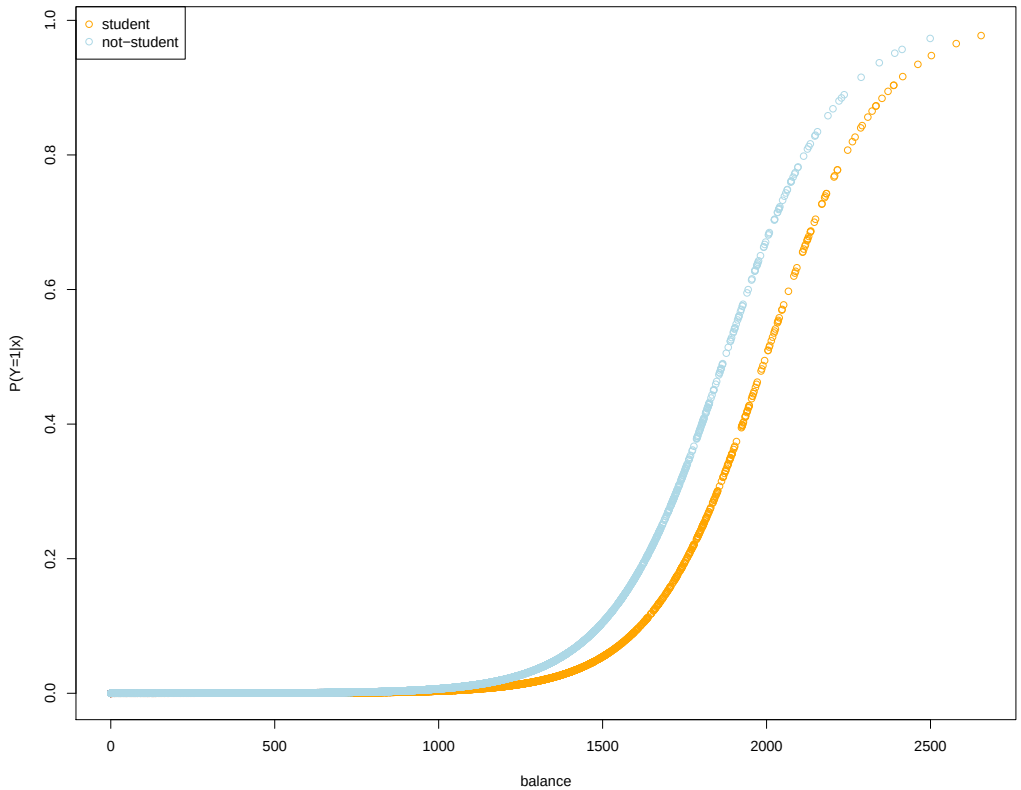
Here is the output using balance and student.

```
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.



Confounding Example:

The Default data gives us a nice example of "confounding".

Suppose we do a logistic regression using only `student`.

Here the coefficient for the student dummy is positive, suggesting that a student is more likely to default.

But, in the multiple logistic regression, the coefficient for student was -.7 and we saw the a student was less likely to default at any fixed level of `balance`.

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2970  -0.2970  -0.2434  -0.2434   2.6585

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
studentYes   0.40489    0.11502    3.52 0.000431 ***
---
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: 2908.7  on 9998  degrees of freedom
AIC: 2912.7

Number of Fisher Scoring iterations: 6
```

How can this be?

This is the sort of thing where our intuition from linear multiple regression can carry over to logistic regression. Since both methods start by mapping a p dimensional x down to just one number, the have some basic features in common. That is a nice thing about using logistic regression.

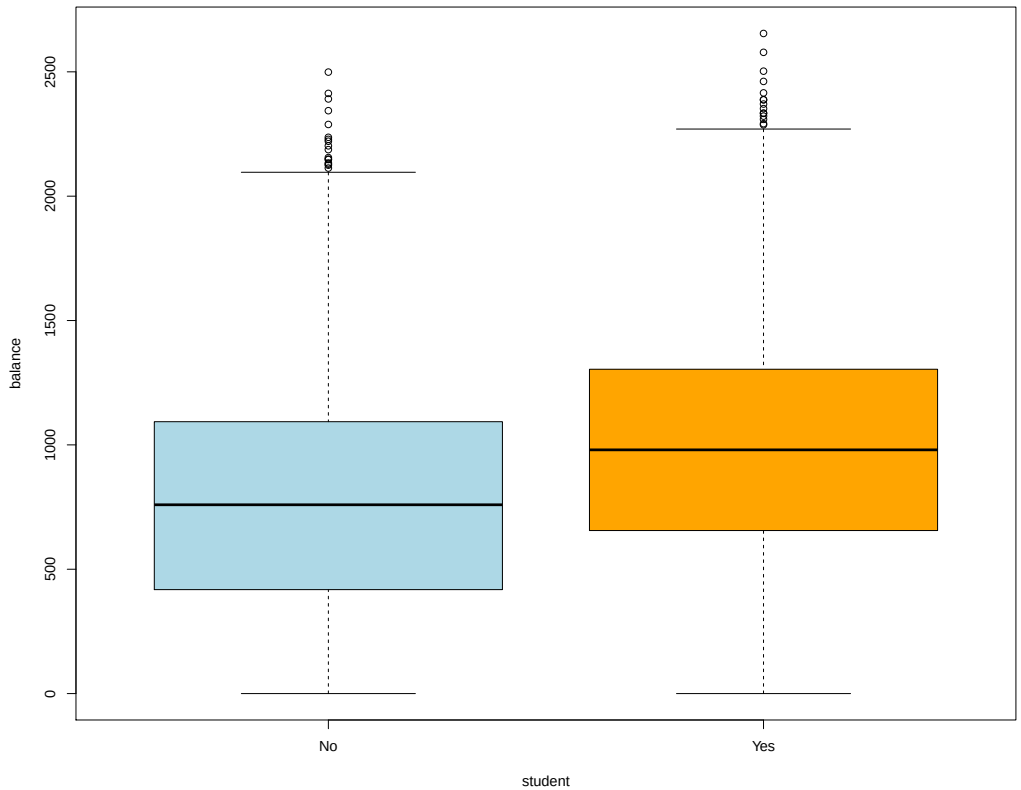
We know that when x's are correlated the coefficients for old x's can change when we add new x's to a model.

Are `balance` and `student` "correlated"?

Here is a plot of `balance` vs. `student`.
We can see the are related.

If all you know is that a credit card holder is a student, then (in the background) they are more likely to have a larger balance and hence more likely to default.

But, if you know the balance, a student is less likely to default than a non-student.



4. 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.

We have reported logistic regression results for a variety of choices for x.

- balance:
  Residual deviance: 1596.5, AIC: 1600.5
- balance + student + income:
  Residual deviance: 1571.5, AIC: 1579.5
- balance + student:
  Residual deviance: 1571.7, AIC: 1577.7
- student:
  Residual deviance: 2908.7, AIC: 2912.7

A smaller residual deviance indicates a better fit.
But, it can only get smaller when you add variables!

The deviance is just -2 times the maximized log likelihood. When you add x variables the maximized likelihood can only get bigger so the deviance can only get smaller.

If you have more coefficients to optimize over you can only do better since you can set them to zero if you want.

This is analogous to the fact that in linear multiple regression R^2 can only go up when you add x's.

AIC is analogous to adjusted R^2 in that it penalizes you for adding variables.

Rather than choosing the model with the smallest deviance, some people advocate choosing the model with the smallest AIC value:

$$AIC = -2log(\hat{L}) + 2(p+1) = deviance + 2(p+1),$$

where $\hat{L}$ is maximized likelihood and p is the number of xs (we add 1 for the intercept).

The idea is that as you add variables (the model gets more complex), deviance goes down but 2*(p+1) goes up.

The suggestion is to pick the model with the smallest AIC.

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 !!!

5. Target Marketing: Tabloid Data

A large retailor wants to explore the predictibility of reponse to a tabloid mailing.

If they mail a tabloid to a customer in their data-base, can they predict whether or not the customer will respond by making a purchase?

The dependent variable is 1 if they buy something, 0 if they do not.

They tried to come up with x's based on past purchasing behaviour.

They came up with lots of variables, we'll just look at four:

- nTab: number of past tabloid orders.
- moCbook: months since last tabloid order.
- iRecMer1 : 1/ months since last order in merchandise category 1.
- llDol: log of the dollar value of past purchases

The data for these variables is obtained from the companies operational data base.

Rec is for "recency", big `months` is low recency.

The decision to log the dollar value and use 1/recency is based on past experience the retailor has had with this kind of data.

The retailor decided to perform an experiment by randomly picking 10,000 households to mail the tabloid to.

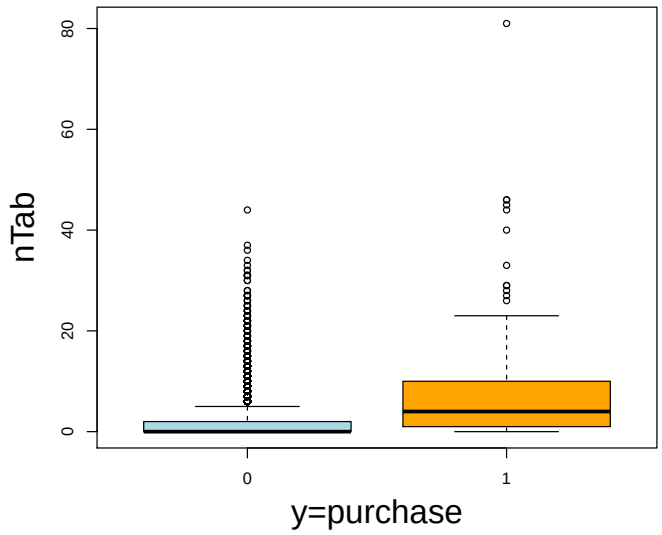
Here is a summary of the data:

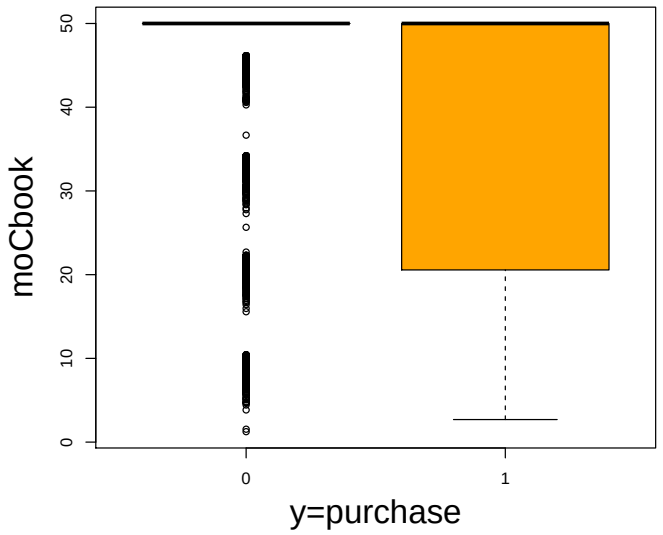
```
purchase      nTab           moCbook          iRecMer1          llDol
0:9742   Min.   : 0.000   Min.   : 1.248   Min.   :0.01961   Min.   :-2.303
1: 258   1st Qu.: 0.000   1st Qu.:50.000   1st Qu.:0.01961   1st Qu.:-2.303
         Median : 0.000   Median :50.000   Median :0.01961   Median :-2.303
         Mean   : 1.857   Mean   :47.597   Mean   :0.09362   Mean   :-1.387
         3rd Qu.: 2.000   3rd Qu.:50.000   3rd Qu.:0.07398   3rd Qu.:-2.303
         Max.   :81.000   Max.   :50.000   Max.   :0.96819   Max.   : 7.310
```

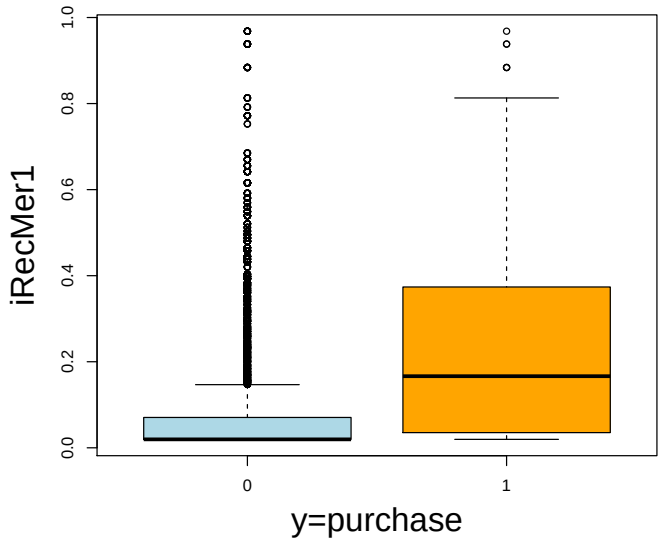
Notice that the precentage of households that make a purchase is pretty small!!

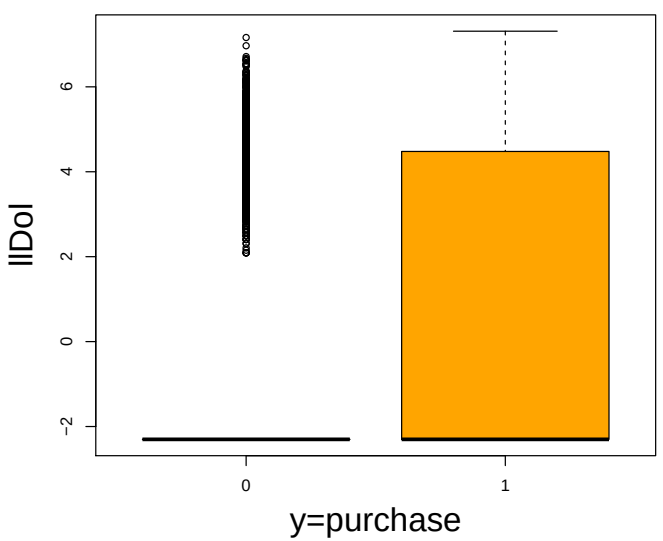
$258/10000 = 0.0258$.

Here is y plotted vs. each of the four x's.









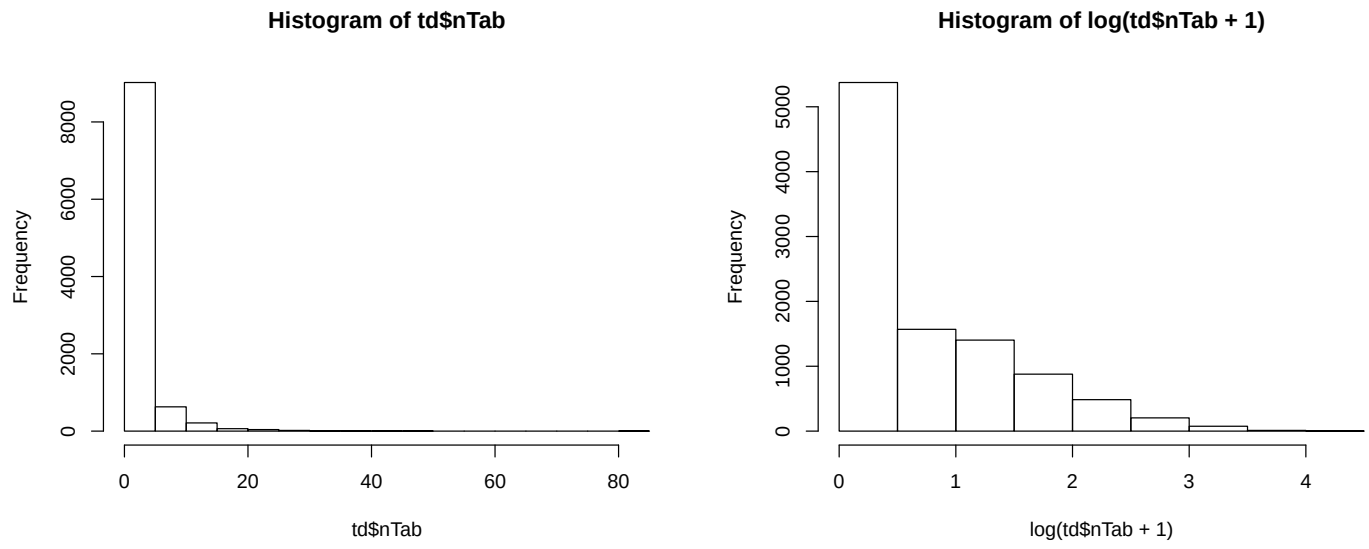
seems promising !!

Here is the logit output from the regression of y=balance on the four x's.

```
Call:
glm(formula = purchase ~ nTab + moCbook + iRecMer1 + llDol, family = binomial,
    data = td)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5007  -0.1883  -0.1612  -0.1568   2.9680

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.621312   0.256668 -10.213  < 2e-16 ***
nTab         0.055299   0.012088   4.575 4.77e-06 ***
moCbook     -0.032486   0.005268  -6.167 6.98e-10 ***
iRecMer1     1.726883   0.312821   5.520 3.38e-08 ***
llDol        0.078418   0.026299   2.982  0.00287 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2396.5  on 9999  degrees of freedom
Residual deviance: 2063.5  on 9995  degrees of freedom
AIC: 2073.5

Number of Fisher Scoring iterations: 7
```

Residual AIC is much smaller then NULL: looks promising.

The nTab variable is very right skewed.



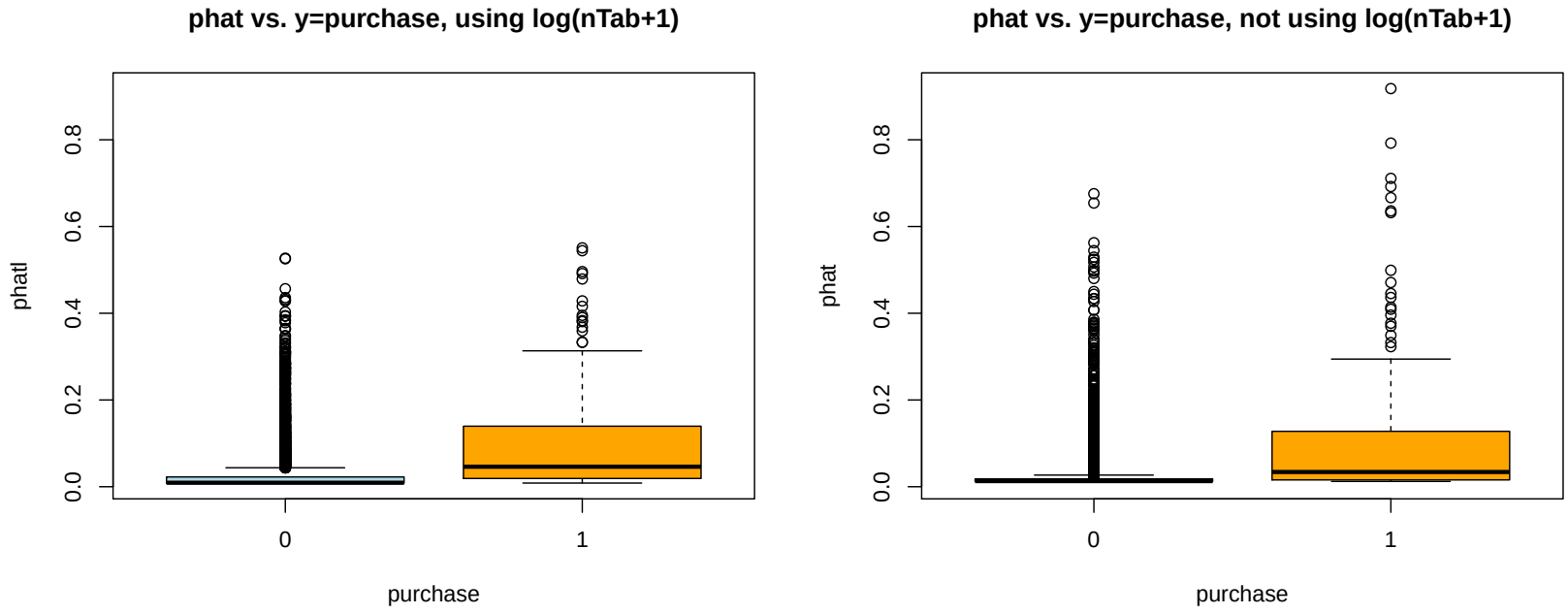
Let's try taking the log of nTab+1.
We have to add 1 because a lot of the nTab values are 0.

Here is the logit output where we have transformed nTab (nTab $\Rightarrow$ log(nTab+1)).

```
Call:
glm(formula = purchase ~ nTablog + moCbook + iRecMer1 + llDol,
    family = binomial, data = td)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2237  -0.2107  -0.1339  -0.1299   3.0917

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.378193   0.284447 -11.876  < 2e-16 ***
nTablog      0.722401   0.097044   7.444 9.76e-14 ***
moCbook     -0.027145   0.005066  -5.358 8.40e-08 ***
iRecMer1     1.247988   0.323153   3.862 0.000113 ***
llDol        0.025953   0.026079   0.995 0.319660
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2396.5  on 9999  degrees of freedom
Residual deviance: 2031.9  on 9995  degrees of freedom
AIC: 2041.9

Number of Fisher Scoring iterations: 7
```

The AIC is now 2032 as opposed to 2064 without the transformation *suggesting* that this might be a good idea.

Here is y=purchase vs. the fitted $\hat{p}(x)$ for the models with and without the transformation.



Easy to see there is some fit, but not easy to see that the transformation helped as suggested by AIC.

Maybe the deviance likes the fact that with the transformation, we don't get big $\hat{p}(x)$'s.

6. Log Odds

Logistic regression and linear regression both start with the same first key step: take a possibly high dimensional x and map it down to a single number using a linear combination of the components of x.

This reduces the dimension from p down to 1!!

Linear regression adds noise, while logistic regression just maps the single number to a probability.

$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots \beta_p x_{ip}$.

Linear Regression: $Y_i = \eta_i + \epsilon_i$.
Logistic Regression: $Y_i \sim Bernoulli(p_i), p_i = F(\eta_i)$.

Linear regression (without transformations!) is interpretable.

The non-linear F in logistic regression makes it somewhat less interpretable.

We can invert F to express our model in terms of the *odds ratio* $\frac{p}{1-p}$.

$$p = F(\eta) = \frac{e^\eta}{1+e^\eta} \Rightarrow \log(\frac{p}{1-p}) = \eta.$$

So, we can write the logistic model as:

$$Y_i \sim Bernoulli(p_i),$$

$$\log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots \beta_p x_{ip}.$$

Some people find this more interpretable.

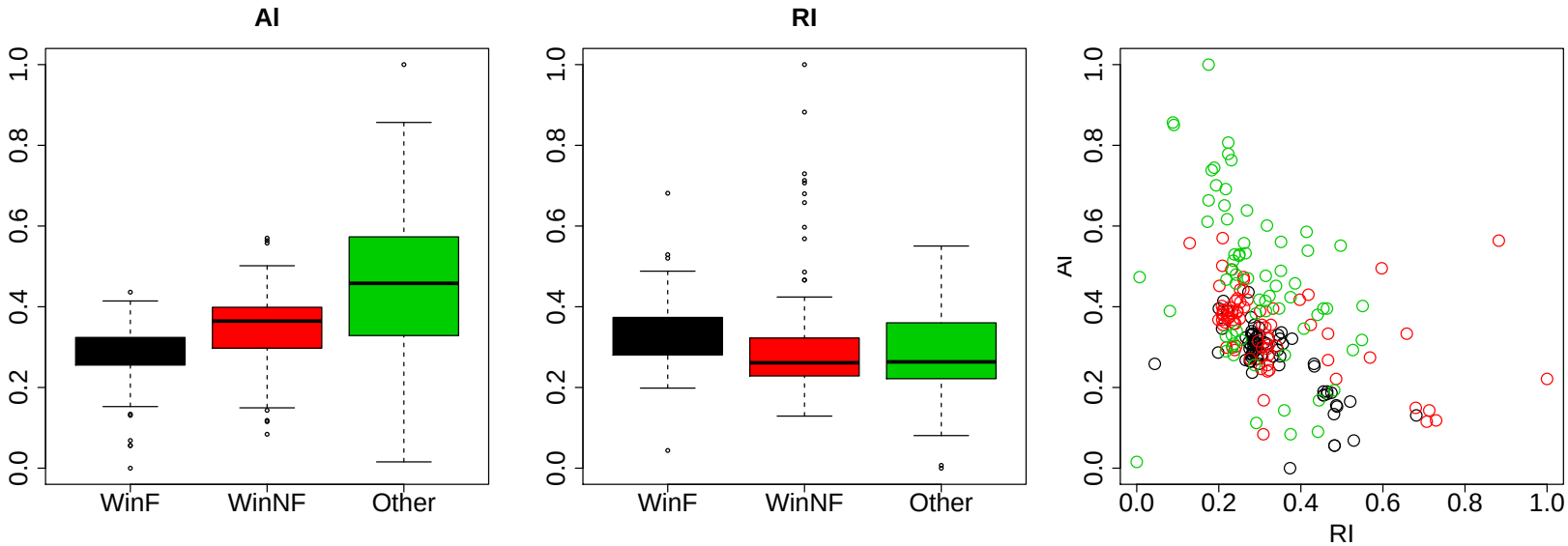
7. Multinomial Logit

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.

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



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.

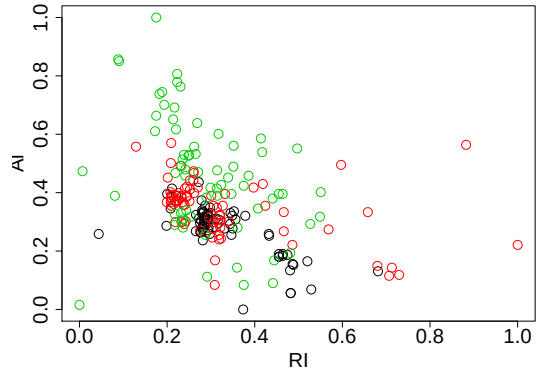
Here is ouput from fitting a multinomial logit using the forensic glass data.
(R package nnet, function multinom).

```
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
```

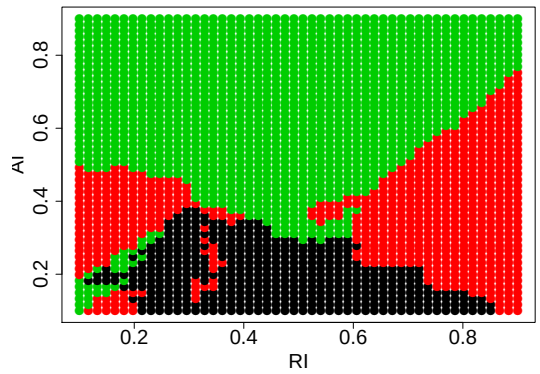
The first β has been set to 0.

Note: 402.6627+12 = 414.6627

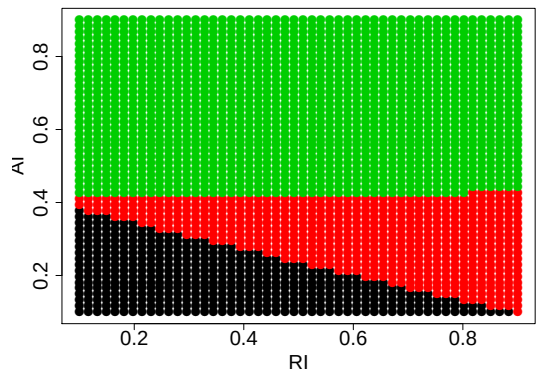
plot of the data.



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

$$\eta_1 = 0$$

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

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

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

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

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

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

When RI increases, with AI held fixed, Y=Other and Y=WinNF become more likely than Y=WinF and Y=WinNF becomes very slightly more likely relative to Other.

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

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

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

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.