228.371 - Statistical Modelling for Engineers and Technologists

Week 8. Logistic Regression

Dr. Daniel Walsh
IIMS 3.07 x 41032
d.c.walsh@massey.ac.nz

Semester One - 2015

Logistic Regression

- Linear regression was concerned with modelling a *continuous* response variable with continuous or discrete predictor variables.
- Logistic regression is concerned with modelling a *dichotomous* (or binary) response variable with continuous or discrete predictor variables.

Example: `fir.txt`

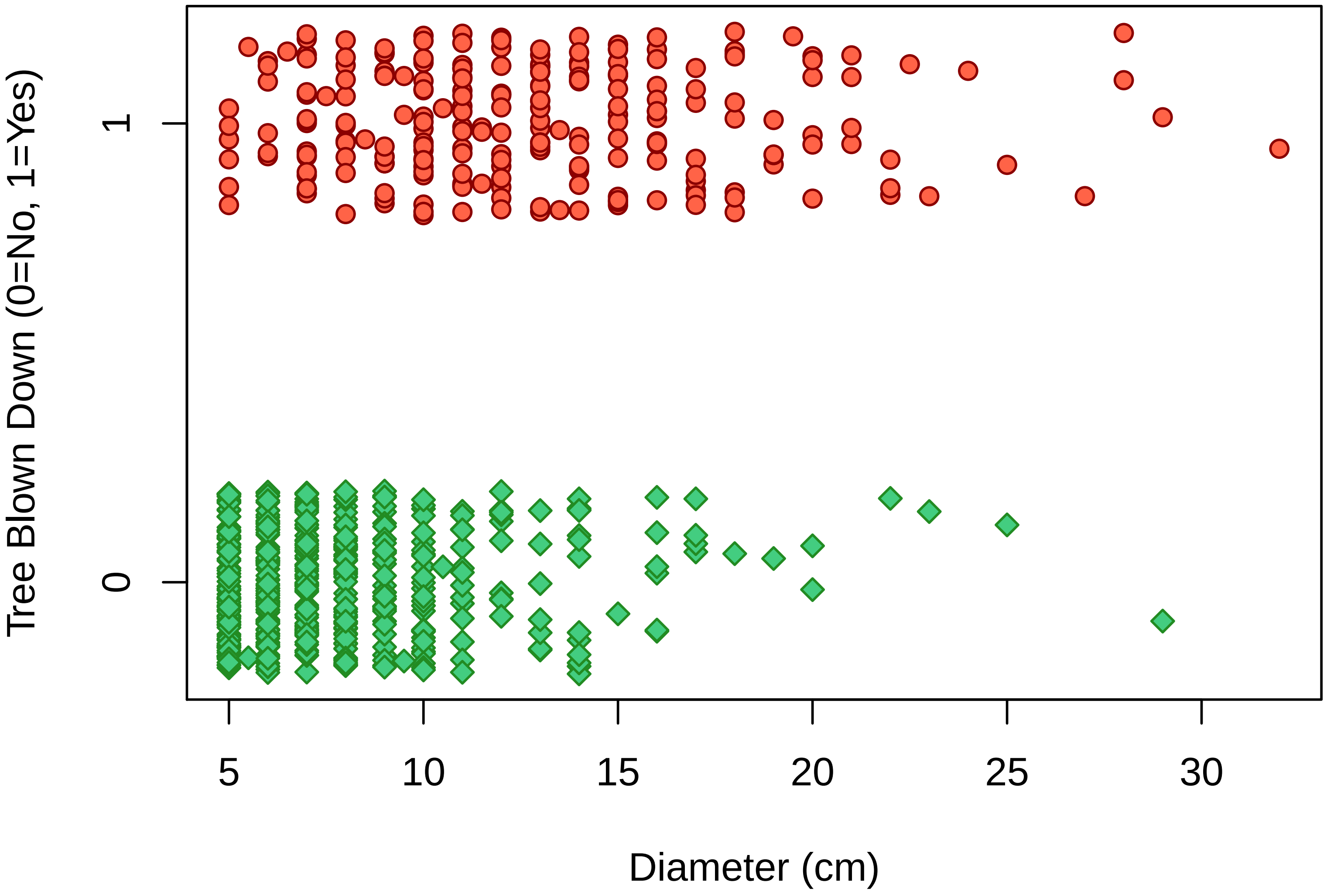
Records details of 659 *balsam fir* trees from Boundary Water Canoe area wilderness, the tree's diameter, and whether or not the tree blow down in a 1999 storm with 90mph winds (1 indicates tree was blown down, 0 not).

```
fir <- read.table (file="Data/fir.txt", header=TRUE)
summary (fir)
```

```
      dia              y         
 Min.   : 5.0   Min.   :0.0000  
 1st Qu.: 6.0   1st Qu.:0.0000  
 Median : 8.0   Median :0.0000  
 Mean   : 9.7   Mean   :0.3536  
 3rd Qu.:12.0   3rd Qu.:1.0000  
 Max.   :32.0   Max.   :1.0000  
```

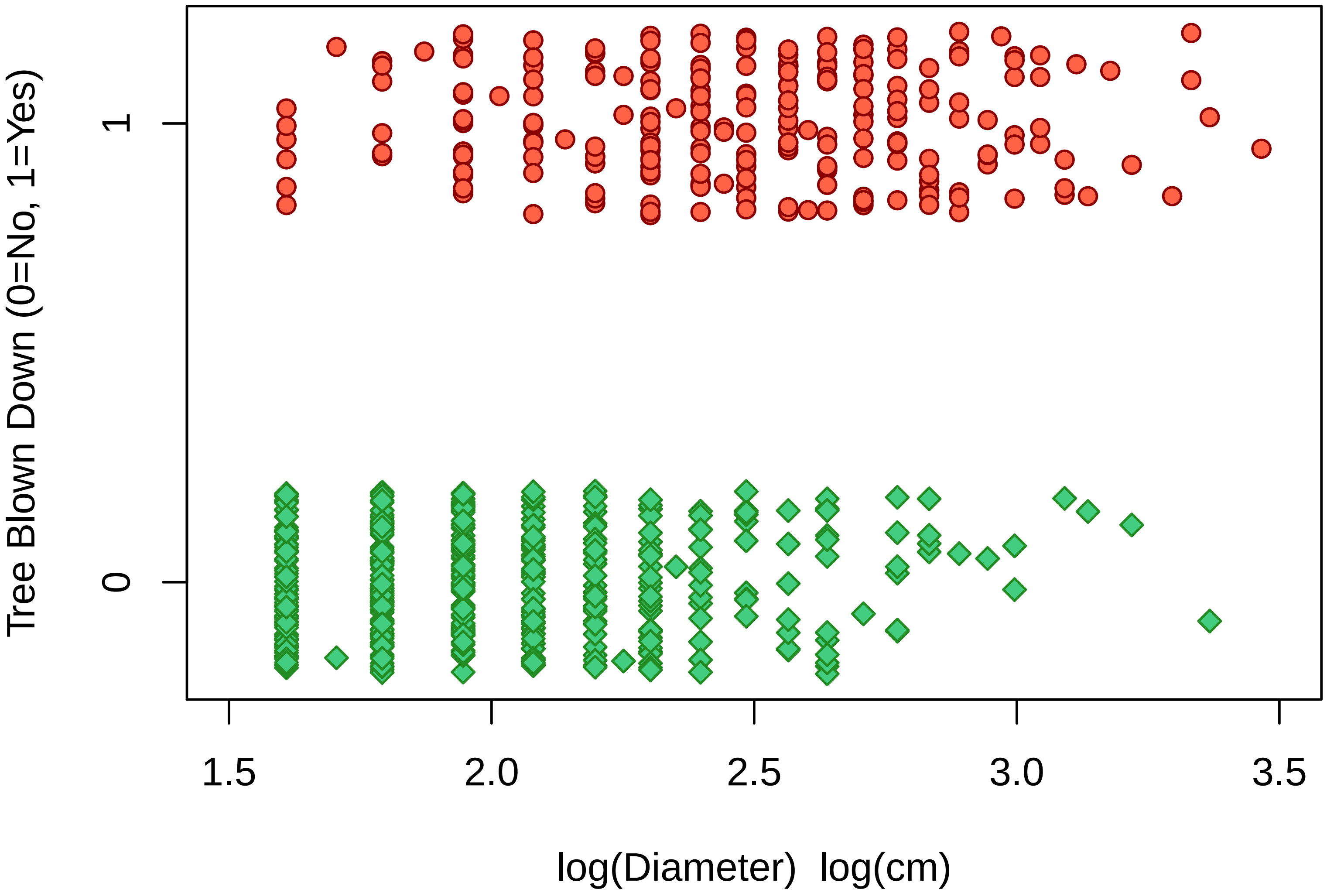
Example: `fir.txt`

- Proportion of tree blown down increases with diameter.

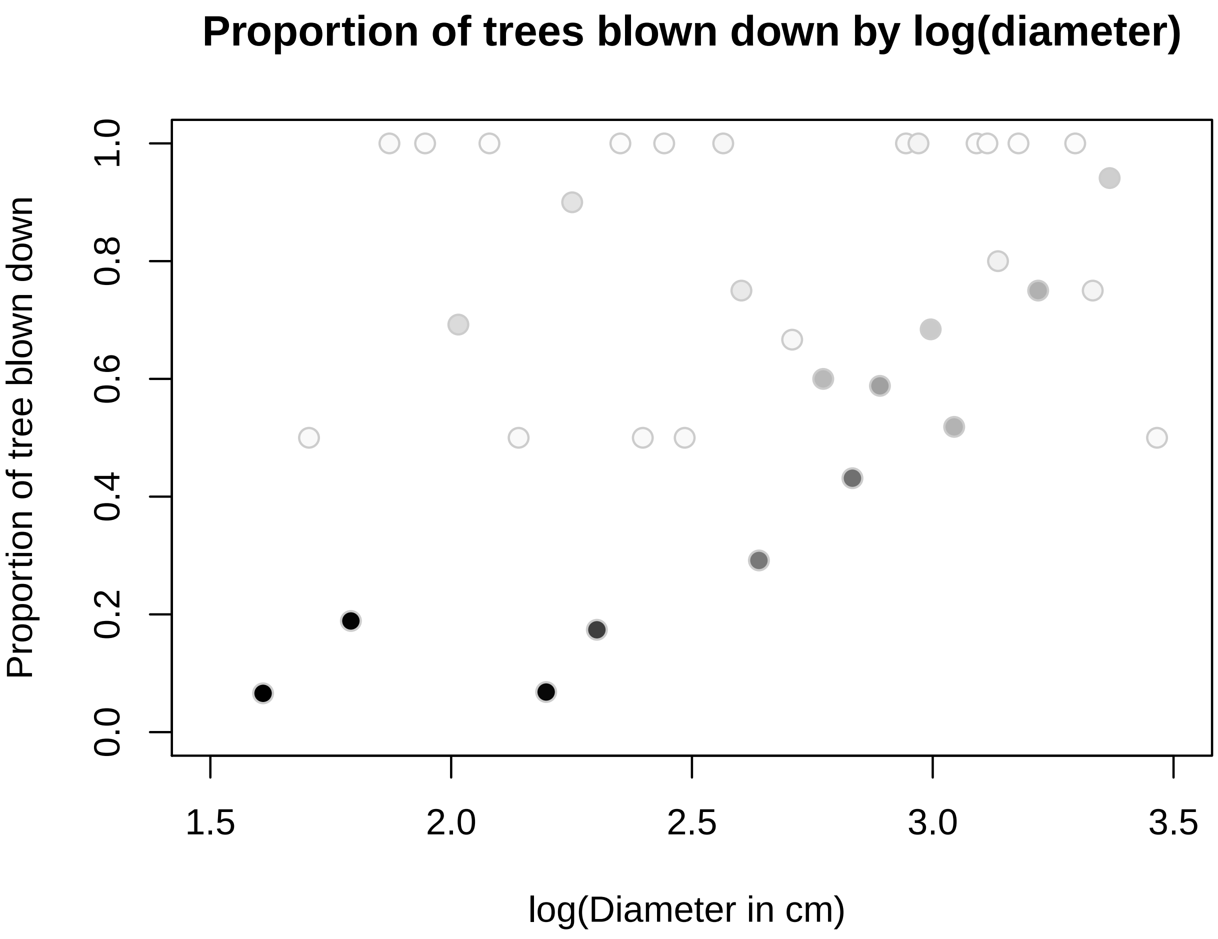


Example: `fir.txt`

- Diameter is left skewed - Log transform is taken.



Example: `fir.txt`



- Color indicates number of observations (the lighter the fewer).
- Moderate positive trend present.

Generalised linear model

The linear regression model assumes that the **mean** of a variable is linearly related to the predictors:

$$\mathbb{E}(y_i) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_{qi}$$

In a **generalised** linear model, a **function** of the mean is linearly related to the predictors:

$$g(\mathbb{E}(y_i)) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_{qi}$$

The function $g(\cdot)$ is called the **link** function.

Logistic regression

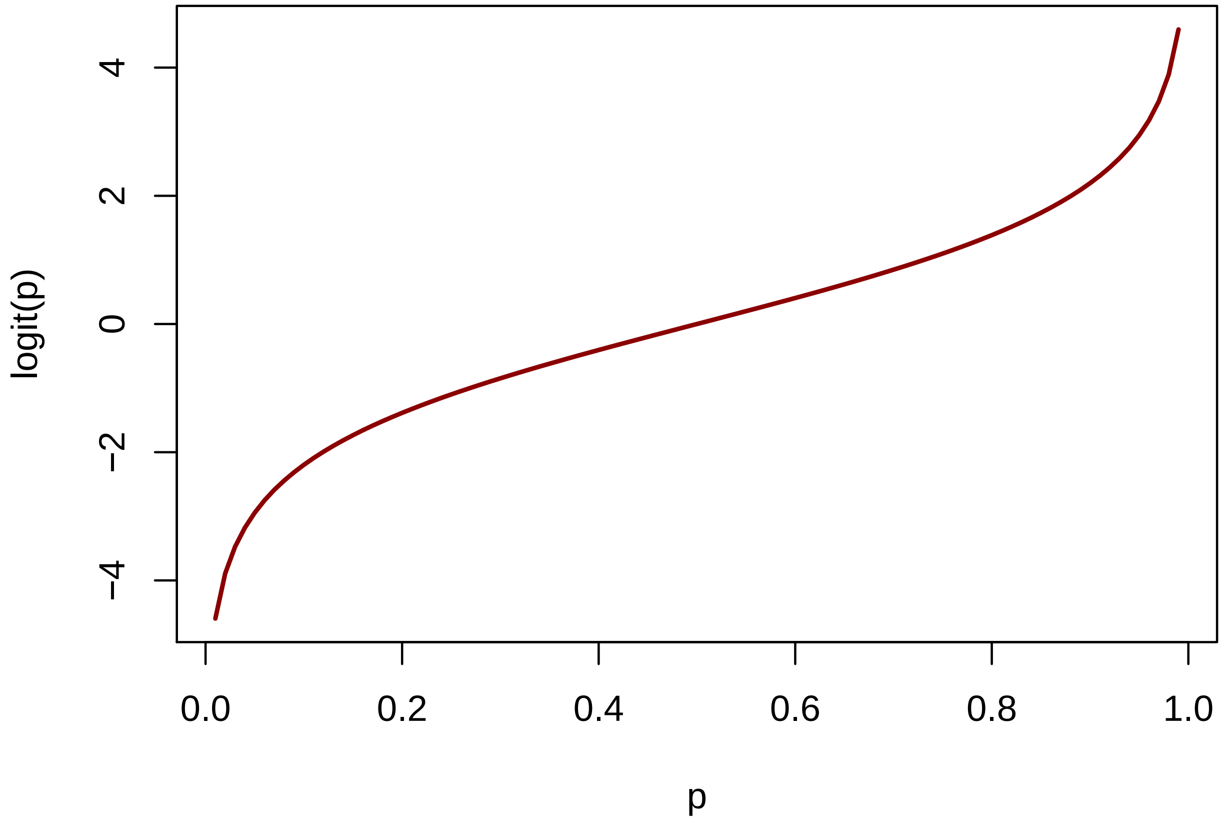
- When y is a dichotomous (0/1) variable, $\mathbb{E}(y_i)$ is the proportion (p) of 1's (successes). We want a function that will map p falling into the (0,1) interval (LHS of the equation) to the real line (RHS of the equation).

$$g(\mathbb{E}(y_i)) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_{qi}$$

- The most common choice of link, and the one we will focus on, is the *logit* function:

$$\mathrm{logit}\,(p) = \log\left(\frac{p}{1-p}\right)$$

Logit function



$$\lim_{p \to 1} \operatorname{logit}(p) = \infty$$

$$\lim_{p \to 0} \operatorname{logit}(p) = -\infty$$

Logistic Regression in R

```
 m1 <- glm (y ~ log(dia), data=fir, family=binomial)
 summary (m1)
```

```
Call:
glm(formula = y ~ log(dia), family = binomial, data = fir)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5073  -0.7565  -0.4936   0.8096   2.3272

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -7.8925     0.6325  -12.48   <2e-16 ***
log(dia)      3.2643     0.2761   11.82   <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: 856.21  on 658  degrees of freedom
Residual deviance: 655.24  on 657  degrees of freedom
AIC: 659.24

Number of Fisher Scoring iterations: 4
```

Where do the fitted βs come from?

- The β's are chosen to maximise the probability of the observed data:

$$P(\boldsymbol{y} \mid \boldsymbol{\beta}) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1-y_i}$$

$$\text{logit}\,(p_i) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_q x_{q_i}$$

- This is called "maximum likelihood."
- We estimate the parameters ($\hat{\beta}_i$'s) so as to maximise the probability (or likelihood) of the observed data ($\boldsymbol{y}$).
- This is an iterative procedure.

Interpretation

Note:

$$\mu = \log\left(\frac{p}{1-p}\right) \Rightarrow p = \frac{e^{\mu}}{1+e^{\mu}}$$

As in linear regression, non-zero β_j suggests x_j is a useful predictor. Suppose `dia=5` hence log(`dia`)$=1.61$.

Then the probability the tree has blown down is:

```
 exp(-7.8925+3.2643*log(5))/(1+exp(-7.8925+3.2643*log(5)))
[1] 0.06668194
```

Alternatively:

```
 mu <- predict.glm (m1, newdata=data.frame(dia=5))
 mu
[1] -2.638832
 exp(mu)/(1+exp(mu))
[1] 0.06668071
```

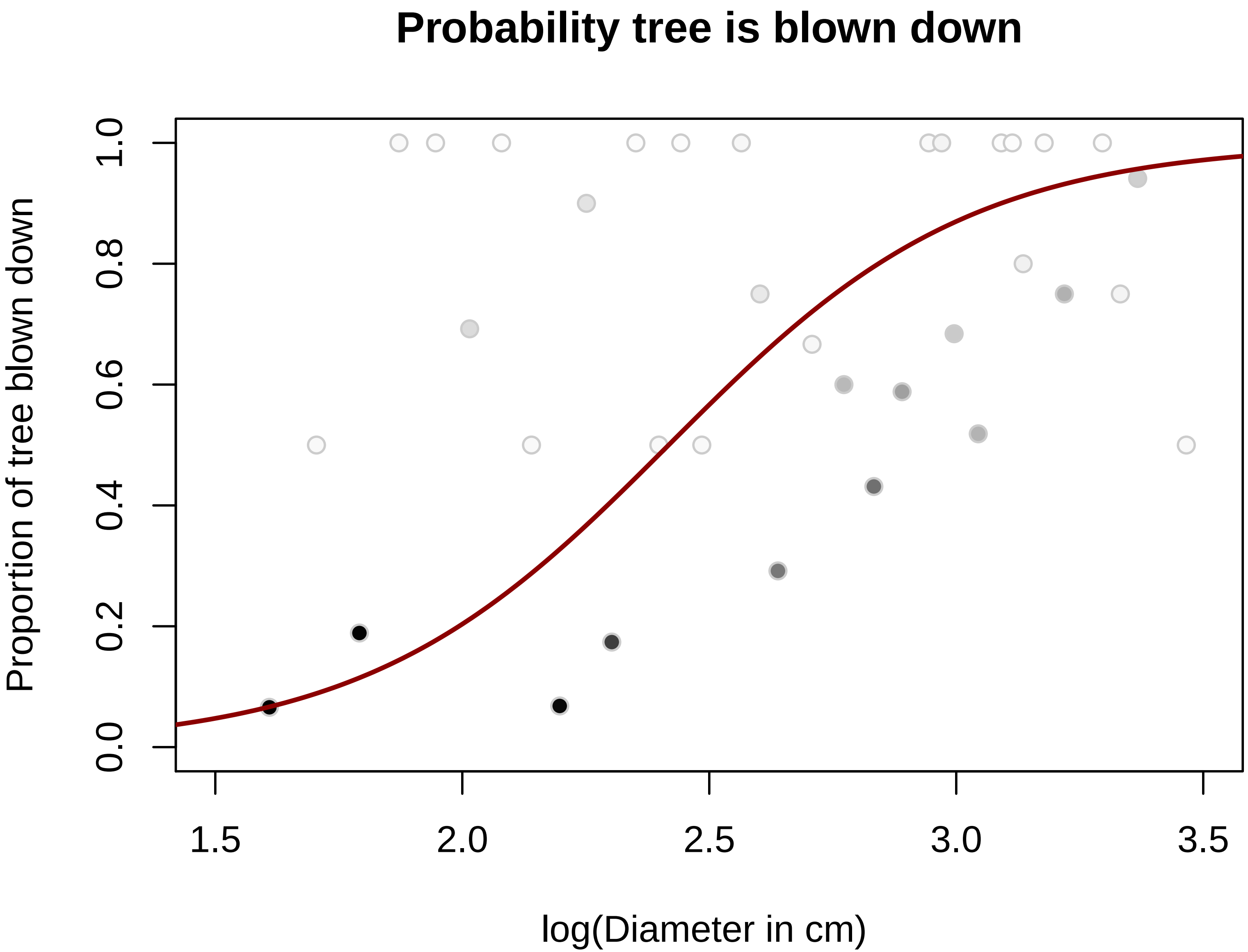
Interpretation

However, the relationship between changes in x (log(diameter)) and changes in $\mathbb{E}(y)$ is more complicated. e.g. for $x = 2.5, 3.0, 3.5$

```
 mu <- predict.glm (m1, newdata=data.frame(dia=exp(c(2.5,3.0,3.5))))
 exp(c(2.5,3.0,3.5))
[1] 12.18249 20.08554 33.11545
 mu
        1         2         3
0.2681993 1.9003320 3.5324646
 exp(mu)/(1+exp(mu))
        1         2         3
0.5666508 0.8699291 0.9715975
```

Changing x by 0.5 (or any constant amount) has a different effect if we are changing it from 2.5 to 3.0 or 3.0 to 3.5.

Predicted probabilities



Odds and Log Odds

What is increasing linearly is the *log odds*: (Note: we have remove the observation subscript i for simplicity).

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_q x_q$$

The odds,

$$\frac{p}{1-p,}$$

are multiplied by a factor of $\exp(\beta_j)$ for each unit of increase of x_j.

Binned Data

We can also analyse the data by binning the response variable by the predictor.

```
successes <- table(fir$dia, fir$y)[,2:1]
uniqueD <- sort(unique(fir$dia))
```

```
cbind (successes, uniqueD)
      1  0 uniqueD
5     6 82     5.0
5.5   1  1     5.5
6     6 85     6.0
6.5   1  0     6.5
7    17 73     7.0
7.5   1  0     7.5
8    12 57     8.0
8.5   1  0     8.5
9    14 34     9.0
9.5   2  1     9.5
10   22 29    10.0
10.5  1  1    10.5
11   20 14    11.0
11.5  3  0    11.5
12   15 10    12.0
13   21  7    13.0
13.5  2  0    13.5
14   14 13    14.0
15   16  1    15.0
16   13  6    16.0
17    9  4    17.0
18    9  1    18.0
19    4  1    19.0
19.5  1  0    19.5
20    6  2    20.0
...
```

```
 m2 <- glm(successes~log(uniqueD), family=binomial)
 summary (m2)
```

```
Call:
glm(formula = successes ~ log(uniqueD), family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8980  -0.8097   0.3527   1.1355   2.3303

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -7.8925     0.6325  -12.48   <2e-16 ***
log(uniqueD)   3.2643     0.2761   11.82   <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: 250.856  on 34  degrees of freedom
Residual deviance:  49.891  on 33  degrees of freedom
AIC: 117.52

Number of Fisher Scoring iterations: 4
```

What has and hasn't changed?

Unchanged:

- Estimates and Standard errors
- Difference between null and residual Deviance and corresponding degrees of freedom

Changed:

- Raw deviances/degrees of freedom
- Residuals

Deviance

Deviance is a measure of fit of the model to the data. Lower deviance means better fit. We can calculate by summing over observations (where y_i is 0/1 (success/failure) of observation i):

$$Deviance = \sum_{i}^{n} -2 \times [y_i \log(\hat{\mu}_i) + (1 - y_i) \log(1 - \hat{\mu}_i)]$$

Or by aggregate over observations with the same set of x-values (y_j is the number of successes and n_j is the number of observations, corresponding to the jth value of x) :

$$\begin{aligned} Deviance = \sum_{1}^{m} -2 \times [&y_j (\log(\hat{\mu}_j) - y_j/n_j) + \\ &(n_j - y_j)(\log(1 - \hat{\mu}_j) - \log((n_j - y_j)/n_j))] \end{aligned}$$

The null deviance (deviance under the null hypothesis) uses a single μ; the residual deviance uses the μ given by our model.

Model Utility

We want to test whether our predictions are better with predictor x.

- Null hypothesis: Null model (each observation has same probability of success) is adequate;
- Test statistic: difference in deviances.
- Null distribution of the test statistic: Chi-squared distribution with degrees of freedom, $df = df_{Null} - df_{model}$.

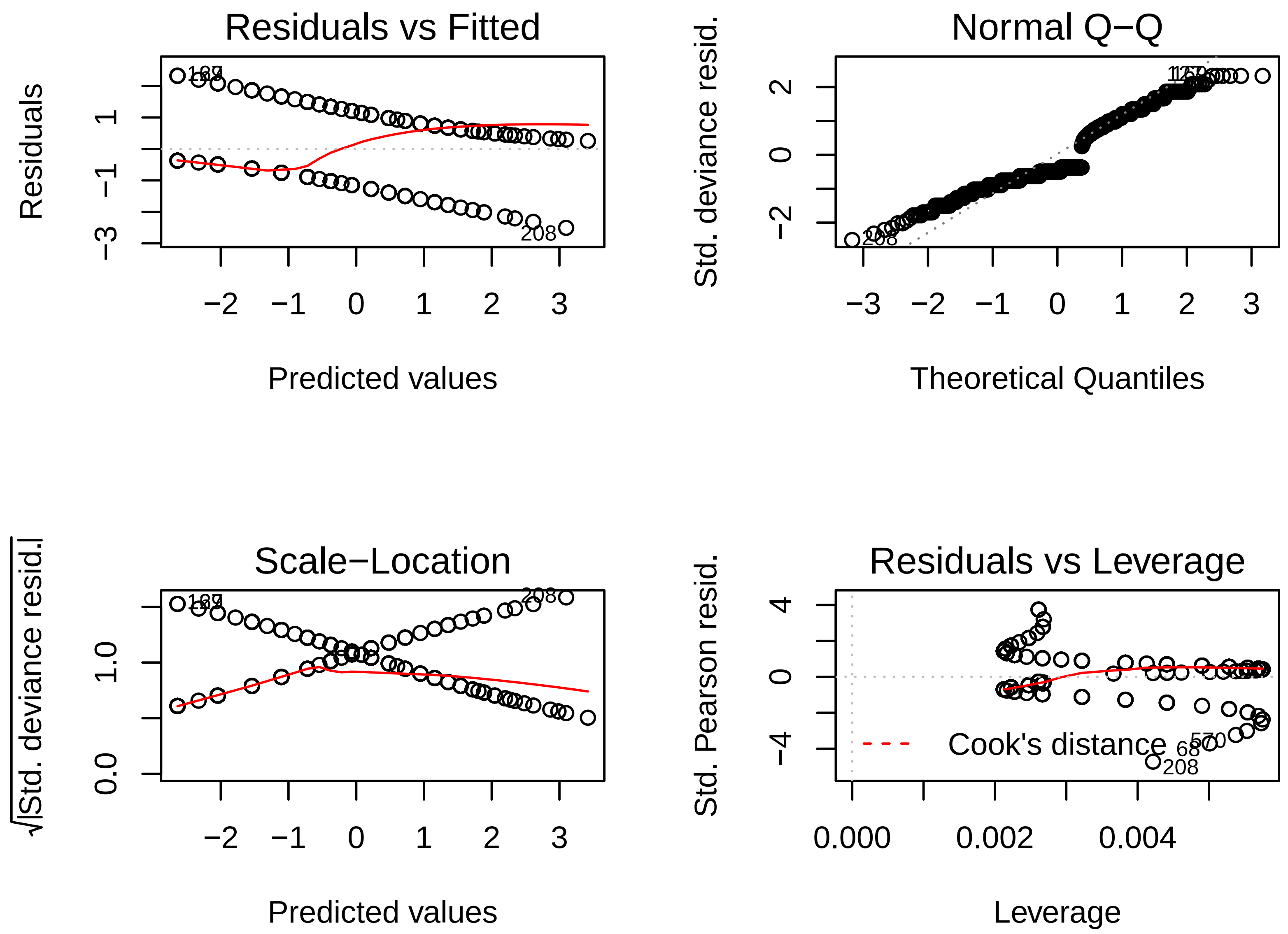
For the `fir.txt` data:

Deviances are given when model is printed. The p-value to reject null model can be calculated with `pchisq()`.

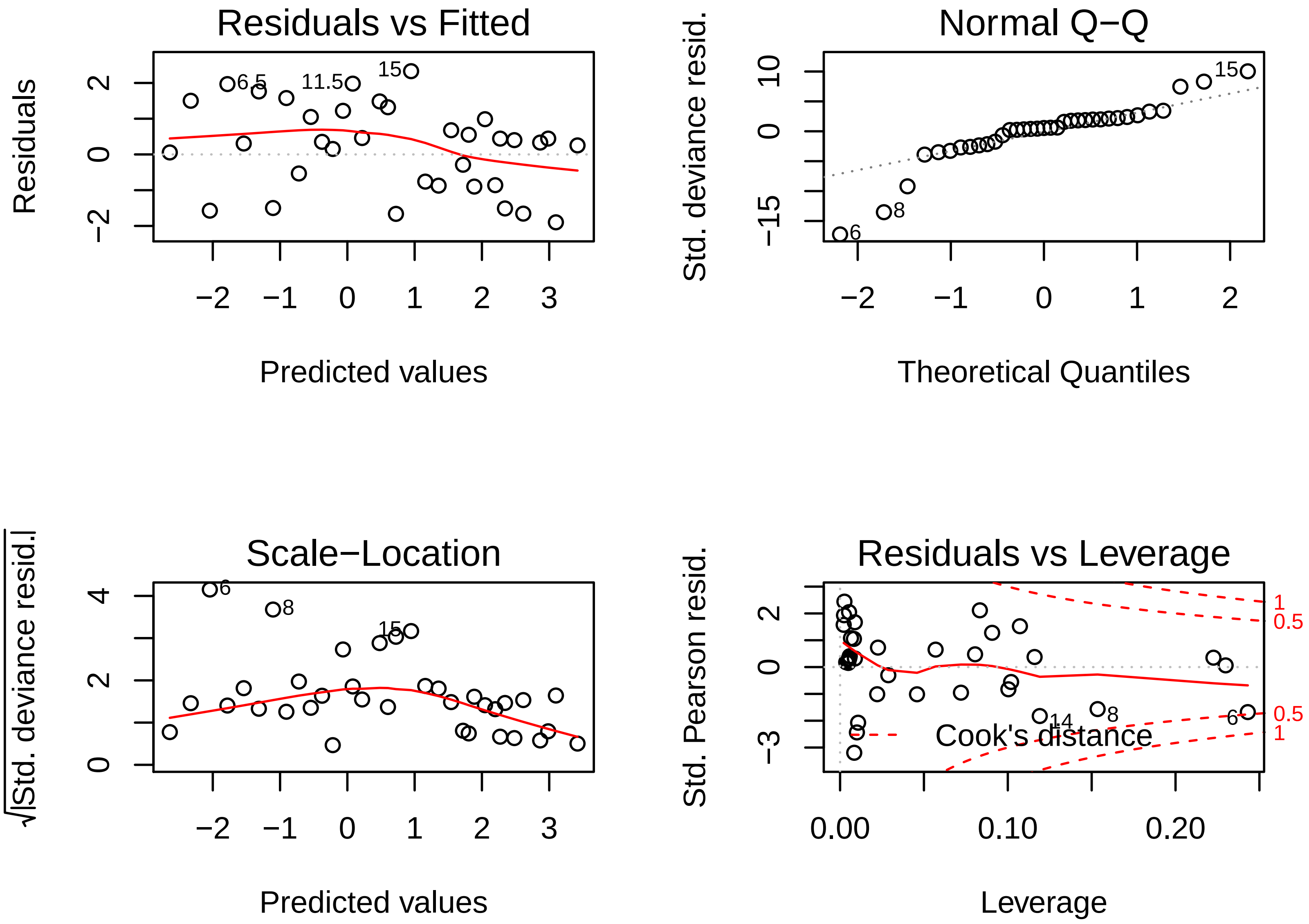
```
 m2
Call:  glm(formula = successes ~ log(uniqueD), family = binomial)

Coefficients:
 (Intercept)  log(uniqueD)
      -7.892         3.264

Degrees of Freedom: 34 Total (i.e. Null);  33 Residual
Null Deviance:     250.9
Residual Deviance: 49.89  AIC: 117.5
 ## P-value
 pchisq(q=250.856-49.891, df=34-33, lower=FALSE)
[1] 1.286026e-45
```

- This is the same regardless of how we fit the model (`m1` vs `m2`).
- Unlike linear regression, it is not identical to the z-test shown for the coefficient of x.

Diagnostic Plots - Individual Observations



Diagnostic Plots - Aggregated Data



Residuals

- Deviance residuals:

$$\sqrt{-2\left[y_j\left(\log(\hat{\mu}_j)-\frac{y_j}{n_j}\right)+(n_j-y_j)\left(\log(1-\hat{\mu}_j)-\log(\frac{n_j-y_j}{n_j})\right)\right]}$$

  with the usual sign.
- Pearson Residuals:

$$r_p=\frac{y_i-\hat{\mu}_i}{\sqrt{\mu_i(1-\mu_i)/n_i}}$$

- Fitted values plotted are $\mathrm{logit}\,(\hat{\mu}_i)=\beta_o+\beta_x x_{1i}\dots$ (logit scale not probability scale)

We are looking for:

- No outliers.
- No trends.
- Standardised Pearson's residual may have skewed distribution, but should be free of trends.
- No points of high influence (outside Cook's distance contours).
- Cook's distance and leverage interpreted the same way as in linear regression; calculated differently (accounting for differing variances). No longer extreme x-values that are most influential.

Model Adequacy

- 'Saturated model' uses the sample proportion for each unique set of x values. Deviance, degrees of freedom are zero.
- Only sensible when there are multiple observation per predictor set.
- Can use deviance to see if this improves on fitted model.
- If not, our model fits as well as can be hoped.
- E.g. for the tree fall data, residual deviance for logistic regression with a single predictor is 49.891.

```
  pchisq(49.891-0,33-0, lower=FALSE)
 [1] 0.0299041
```

  We do not have a "well fitting model" just yet.

Multiple Logistic Regression

- The various complications that we considered for linear regression are also possible for logistic regression.
- Consider an enlarged tree data set:
  - More trees-two species, Balsam Fir and Blue Spruce
  - Measurement of "local wind severity" - proportion of trees in that area knocked down.
  - Consider interactions/multiplicative relationships

Example: firspruce.txt

The larger data set contains two species of tree.

```
firspruce <- read.table (file="Data/firspruce.txt", header=TRUE)
summary (firspruce)
     dia           severity             y            species
Min.   : 5.00   Min.   :0.02421   Min.   :0.0000   fir   :659
1st Qu.: 8.00   1st Qu.:0.21894   1st Qu.:0.0000   spruce:970
Median :11.50   Median :0.40087   Median :0.0000
Mean   :12.31   Mean   :0.41698   Mean   :0.4696
3rd Qu.:15.00   3rd Qu.:0.59286   3rd Qu.:1.0000
Max.   :51.00   Max.   :0.98327   Max.   :1.0000
```

Example: `firspruce.txt`: Backwards Stepwise Selection

```
 m3 <- step(glm(y ~ log(dia) * species * severity,
           family='binomial', data=firspruce))
```

```
Start:  AIC=1485.73
y ~ log(dia) * species * severity

                            Df Deviance    AIC
- log(dia):species:severity  1   1470.8 1484.8
<none>                           1469.7 1485.7

Step:  AIC=1484.8
y ~ log(dia) + species + severity + log(dia):species + log(dia):severity +
    species:severity

                    Df Deviance    AIC
- species:severity   1   1470.8 1482.8
<none>                   1470.8 1484.8
- log(dia):species   1   1478.6 1490.6
- log(dia):severity  1   1512.3 1524.3

Step:  AIC=1482.84
y ~ log(dia) + species + severity + log(dia):species + log(dia):severity

                    Df Deviance    AIC
<none>                   1470.8 1482.8
- log(dia):species   1   1478.7 1488.7
- log(dia):severity  1   1517.0 1527.0
```

How do we picture what this model looks like?

Example: `firspruce.txt`: Model summary

```
 summary (m3)
Call:
glm(formula = y ~ log(dia) + species + severity + log(dia):species +
    log(dia):severity, family = "binomial", data = firspruce)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1952  -0.6564  -0.3659   0.6912   2.4035

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)             -4.6969     1.0318  -4.552 5.31e-06 ***
log(dia)                 1.0537     0.4385   2.403  0.01627 *
speciesspruce            2.3090     0.8907   2.592  0.00953 **
severity                -8.1858     2.0918  -3.913 9.10e-05 ***
log(dia):speciesspruce  -1.0459     0.3778  -2.768  0.00564 **
log(dia):severity        5.6687     0.8759   6.472 9.67e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2252.3  on 1628  degrees of freedom
Residual deviance: 1470.8  on 1623  degrees of freedom
AIC: 1482.8

Number of Fisher Scoring iterations: 5
```

Prediction

Plot the relationship between probability of blow down and diameter; use colours/line types to show effect of Species and local severity.

```
DatFir <- data.frame (dia=rep(seq(5, 50, 1),2),
                      severity=rep(c(0.25, 0.75), each=46),
                      species=rep("fir", 46*2))
DatSpruce <-data.frame(dia=rep(seq(5, 50, 1),2),
                       severity=rep(c(0.25, 0.75), each=46),
                       species=rep("spruce", 46*2))
summary (DatFir)
```

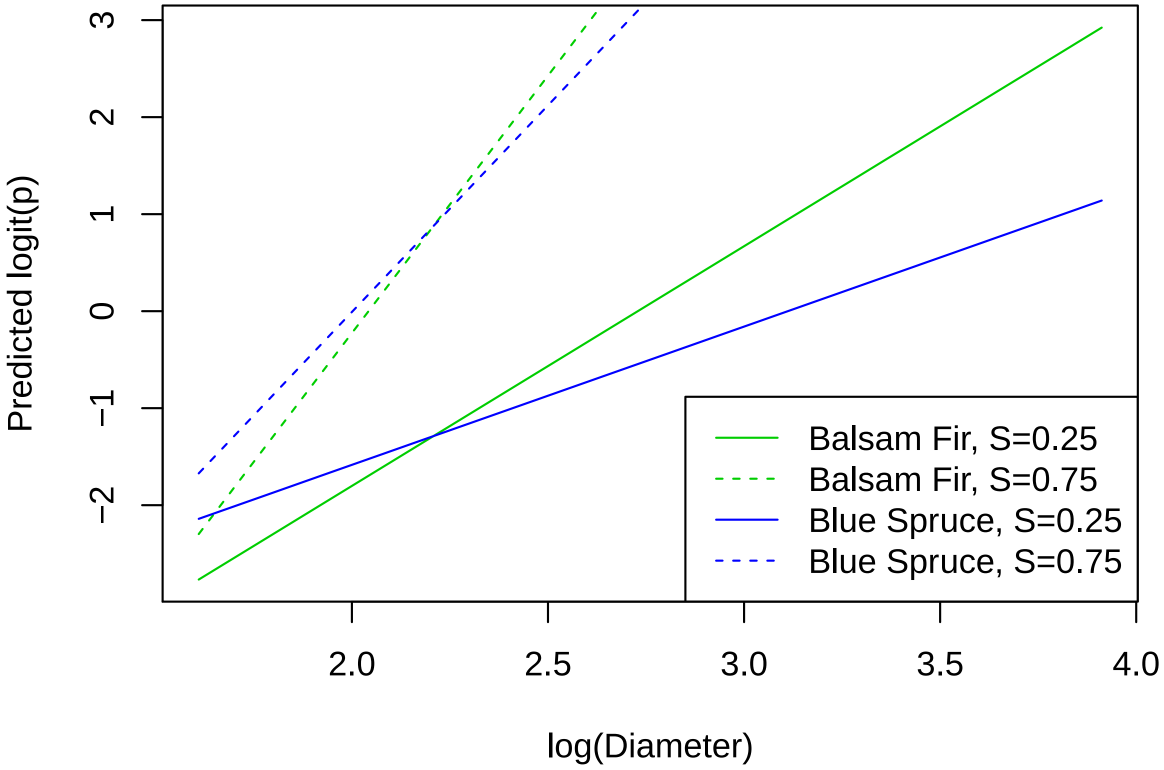
```
     dia          severity    species
Min.   : 5.0   Min.   :0.25   fir:92
1st Qu.:16.0   1st Qu.:0.25
Median :27.5   Median :0.50
Mean   :27.5   Mean   :0.50
3rd Qu.:39.0   3rd Qu.:0.75
Max.   :50.0   Max.   :0.75
```

```
summary (DatSpruce)
```

```
     dia          severity      species
Min.   : 5.0   Min.   :0.25   spruce:92
1st Qu.:16.0   1st Qu.:0.25
Median :27.5   Median :0.50
Mean   :27.5   Mean   :0.50
3rd Qu.:39.0   3rd Qu.:0.75
Max.   :50.0   Max.   :0.75
```

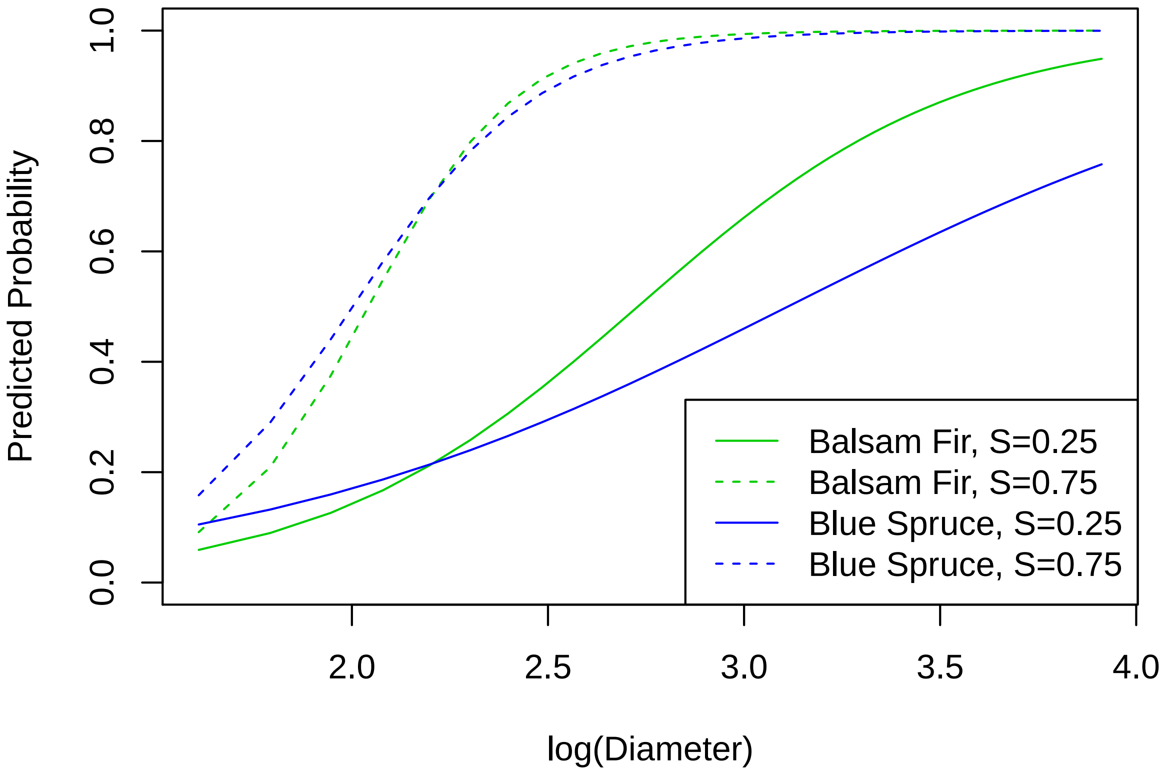
```
pred.Fir.lin     <- predict.glm (m3, DatFir)
pred.Fir.prob    <- predict.glm (m3, DatFir, type='response')
pred.Spruce.lin  <- predict.glm (m3, DatSpruce)
pred.Spruce.prob <- predict.glm (m3, DatSpruce, type='response')
```

Plots produced



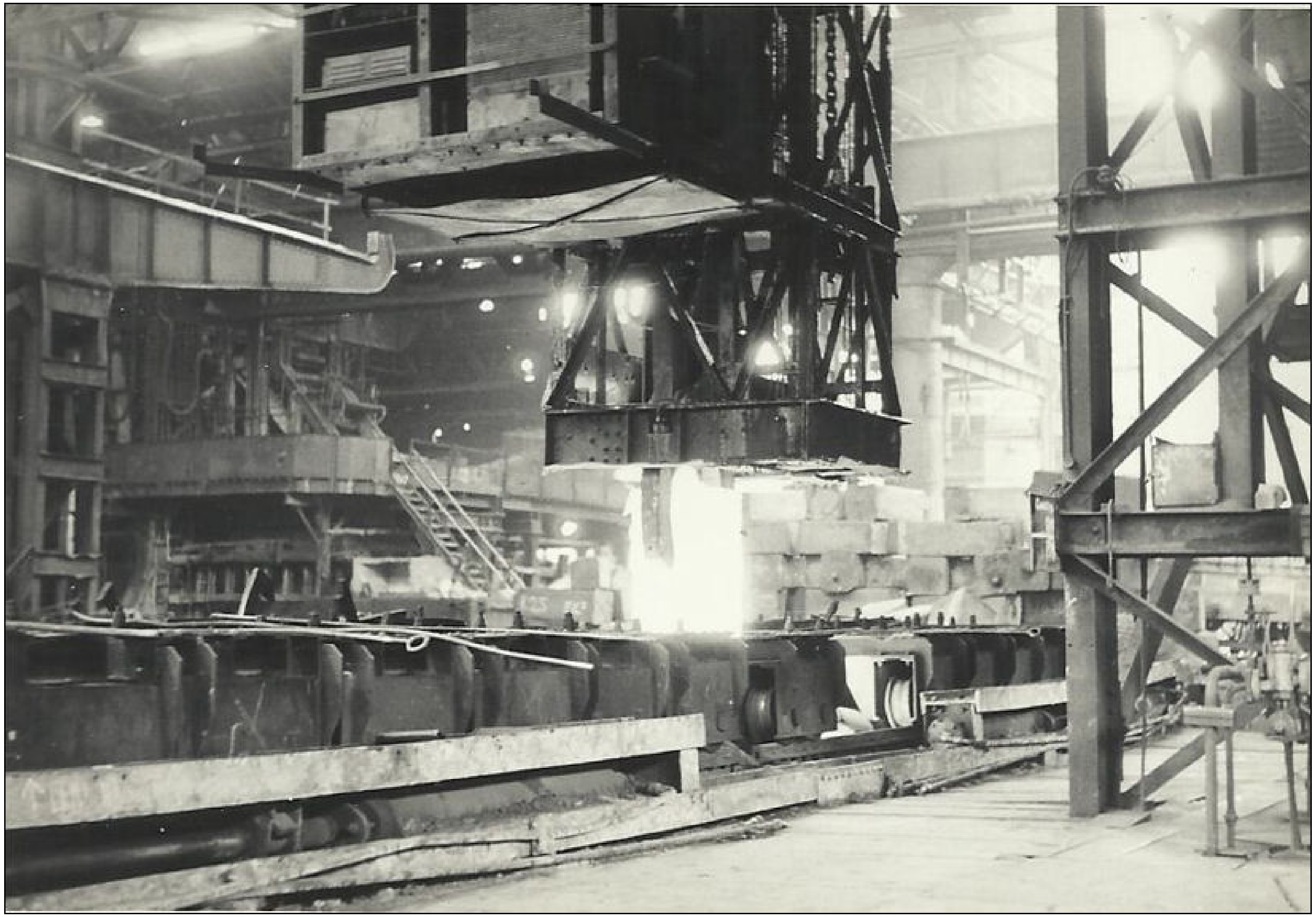
```
plot(log(DatFir$dia[1:46]),pred.Fir.lin[1:46], type="l",
     xlab='log(Diameter)', ylab='Predicted logit(p)',col=3)
lines(log(DatFir$dia[47:92]), pred.Fir.lin[47:92], lty=2,col=3)
lines(log(DatSpruce$dia[47:92]),pred.Spruce.lin[47:92], lty=2,col=4)
lines(log(DatSpruce$dia[1:46]), pred.Spruce.lin[1:46],  col=4)
legend("bottomright", lty=c(1,2,1,2), col=c(3,3,4,4),
       legend=c("Balsam Fir, S=0.25", "Balsam Fir, S=0.75",
           "Blue Spruce, S=0.25", "Blue Spruce, S=0.75"))
```

Plots produced

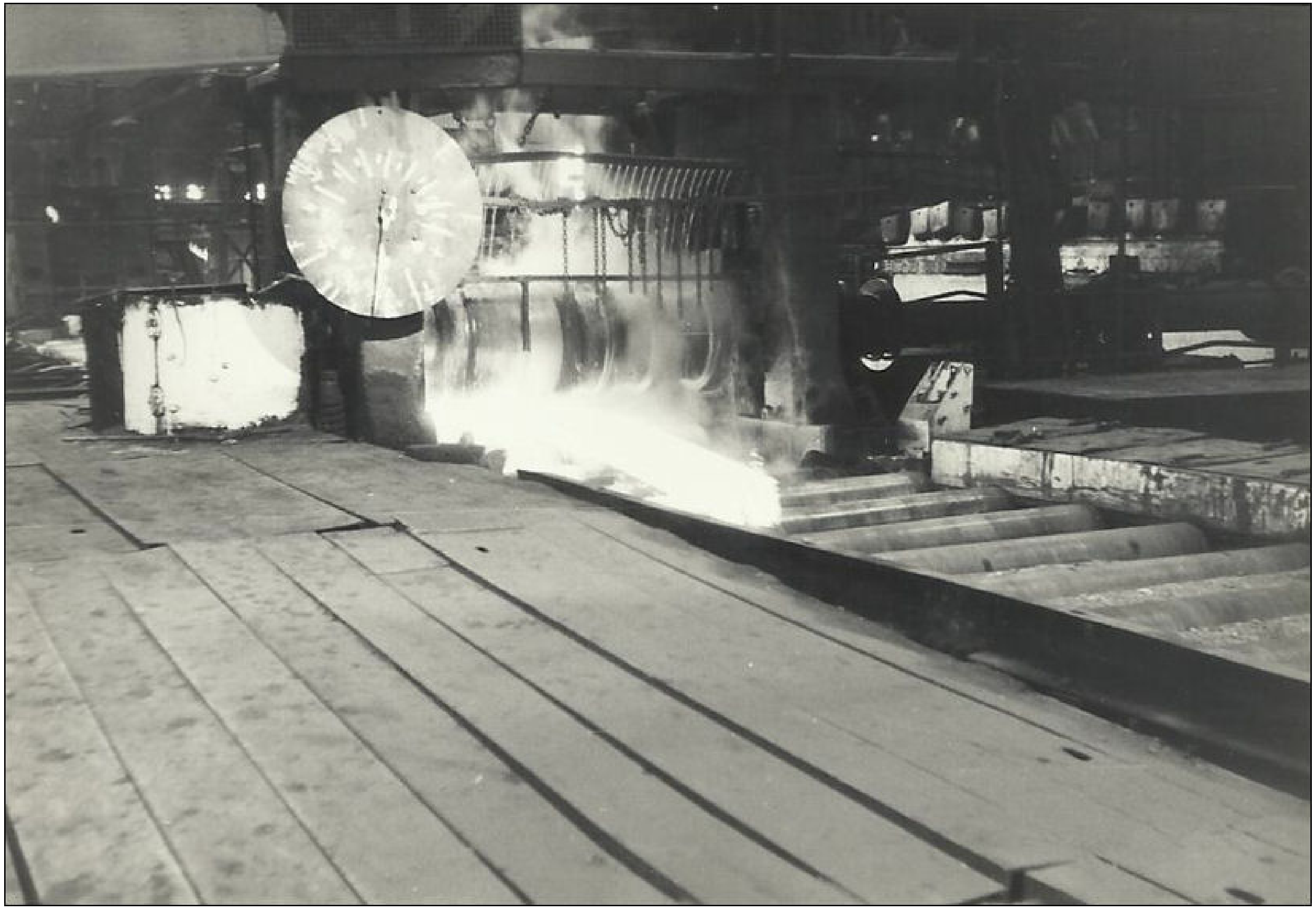


```
plot(log(DatFir$dia[1:46]),pred.Fir.prob[1:46], type="l",
     ylim=c(0,1), xlab='log(Diameter)', ylab='Predicted Probability',col=3)
lines(log(DatFir$dia[47:92]), pred.Fir.prob[47:92], lty=2,col=3)
lines(log(DatSpruce$dia[47:92]),pred.Spruce.prob[47:92], lty=2,col=4)
lines(log(DatSpruce$dia[1:46]), pred.Spruce.prob[1:46],  col=4)
legend("bottomright", lty=c(1,2,1,2), col=c(3,3,4,4),
       legend=c("Balsam Fir, S=0.25", "Balsam Fir, S=0.75",
           "Blue Spruce, S=0.25", "Blue Spruce, S=0.75"))
```

Ingots ready for rolling



Ingots ready for rolling



Example: `ingots.txt`

- Three different soaking times and five different heating times
- Between 7 and 56 observations per combination of factors
- Small number of "successes" (ingot not ready).

Example: `ingots.txt` - ANOVA

- Treat levels as factors to start with
- As with regular ANOVA we first want to assess whether there is an interaction-will use difference in deviance test
- Model with interaction always provides a perfect fit (saturated model)
- Between 7 and 56 observations per combination of factors
- Small number of “successes” (ingot not ready).

Example: ingots.txt - Data

```
ingots <- read.table (file="Data/ingots.txt", header=TRUE)
ingots
```

```
   HeatingTime SoakingTime NotReady Ready
1            7         1.0        0    10
2            7         1.7        0    17
3            7         2.2        0     7
4            7         2.8        0    12
5            7         4.0        0     9
6           14         1.0        0    31
7           14         1.7        0    43
8           14         2.2        2    31
9           14         2.8        0    31
10          14         4.0        0    19
11          27         1.0        1    55
12          27         1.7        4    40
13          27         2.2        0    21
14          27         2.8        1    21
15          27         4.0        1    15
16          51         1.0        3    10
17          51         1.7        0     1
18          51         2.2        0     1
19          51         4.0        0     1
```

Example: ingots.txt - Anova

```
m4a <- glm (cbind(NotReady, Ready) ~ factor(SoakingTime) + factor(HeatingTime),
            data=ingots, family="binomial")
m4b <- glm (cbind(NotReady, Ready) ~ factor(SoakingTime) * factor(HeatingTime),
            data=ingots, family="binomial")
anova (m4a, m4b, test ="Chisq")
```

```
Analysis of Deviance Table

Model 1: cbind(NotReady, Ready) ~ factor(SoakingTime) + factor(HeatingTime)
Model 2: cbind(NotReady, Ready) ~ factor(SoakingTime) * factor(HeatingTime)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        11     11.274
2         0      0.000 11   11.274   0.4206
```

Fit of additive model is OK.

Example: ingots.txt - Anova

```
 anova (m4a, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NotReady, Ready)

Terms added sequentially (first to last)


                    Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                                   18     25.395
factor(SoakingTime)  4   1.0402        14     24.355 0.903633
factor(HeatingTime)  3  13.0810        11     11.274 0.004465 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-values will depend on order.

Example: ingots.txt - Anova

```
 m4c <- glm( cbind(NotReady, Ready) ~
            as.factor(HeatingTime)+as.factor(SoakingTime),
            data=ingots, family="binomial")
 anova(m4c,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NotReady, Ready)

Terms added sequentially (first to last)


                       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                                      18     25.395
as.factor(HeatingTime)  3  12.7100        15     12.685 0.005308 **
as.factor(SoakingTime)  4   1.4113        11     11.274 0.842234
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-values will depend on order.

Example: `ingots.txt` - Heat as a linear predictor

Heating Time is a (borderline) significant predictor; could be used as a linear predictor to gain a degree of freedom.

```
 m5a <- glm(cbind(NotReady, Ready) ~  factor(HeatingTime), data=ingots, family="binomial")
 m5b <- glm(cbind(NotReady, Ready) ~  HeatingTime, data=ingots, family="binomial")
 anova(m5a, m5b, test="Chisq") ## Shows linear predictor works just as well
```

```
Analysis of Deviance Table

Model 1: cbind(NotReady, Ready) ~ factor(HeatingTime)
Model 2: cbind(NotReady, Ready) ~ HeatingTime
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        15     12.685
2        17     13.782 -2  -1.0961   0.5781
```

```
 anova(m5b, test="Chisq")       ## Shows p-value associated with linear predictor
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NotReady, Ready)

Terms added sequentially (first to last)


            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                           18     25.395
HeatingTime  1   11.614        17     13.782 0.0006546 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example: ingots.txt - Coefficients

We should also have a look at the actual coefficients to see if Heating Time is good or bad:

```
summary(m5b)
```

```
Call:
glm(formula = cbind(NotReady, Ready) ~ HeatingTime, family = "binomial",
    data = ingots)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.27249  -0.79572  -0.51483  -0.09311   1.71887

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.41518    0.72754  -7.443 9.83e-14 ***
HeatingTime  0.08070    0.02236   3.610 0.000307 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 25.395  on 18  degrees of freedom
Residual deviance: 13.782  on 17  degrees of freedom
AIC: 32.109

Number of Fisher Scoring iterations: 5
```

Longer Heating Time seems to increase the probability of the Ingot being not ready.

Another ANOVA example

Time and shoot length used in grafting plum trees:

length	time	success	trials
long	autumn	156	240
long	spring	84	240
short	autumn	107	240
short	spring	31	240

```
plum  <- c(156,84, 107,31)
plum  <- cbind(plum, 240-plum)
shoot <- factor(c("long", "long",   "short", "short"))
time  <- factor(c("fall", "spring", "fall",  "spring"))
```

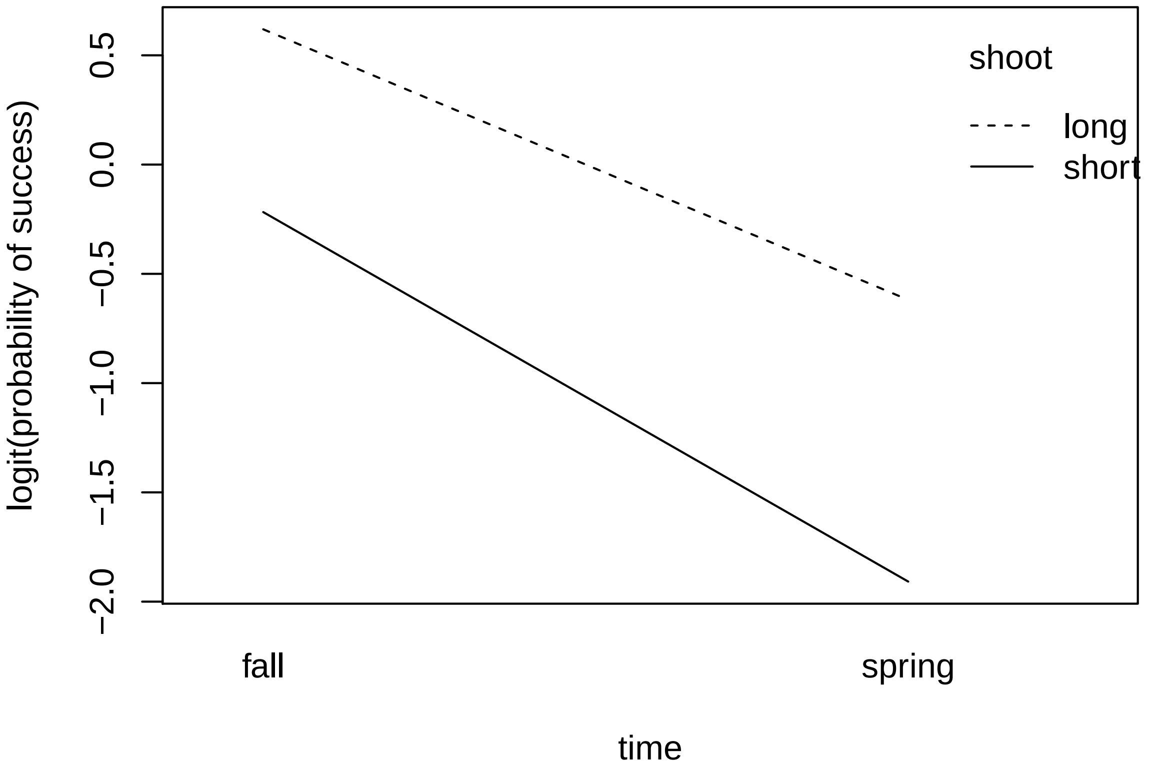
Another ANOVA example

```
 m6 <- glm(plum ~ shoot*time, family="binomial")
 anova(m6, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: plum

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                           3    151.019
shoot       1   45.837         2    105.182 1.285e-11 ***
time        1  102.889         1      2.294 < 2.2e-16 ***
shoot:time  1    2.294         0      0.000    0.1299
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No interaction term needed.

Interaction plot

If there are no groups with all successes or all failures, it can be useful to make an interaction plot on the logit scale.



```
logit <- function (p) { log(p/(1-p)) }
interaction.plot (time, shoot, logit(plum[,1]/240),
                  ylab="logit(probability of success)")
```

Which are the favourable conditions?

```
 m7 <- glm (plum ~ time + shoot, family="binomial")
 summary (m7)
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)   0.7138     0.1217   5.867 4.45e-09 ***
timespring   -1.4275     0.1465  -9.747  < 2e-16 ***
shootshort   -1.0177     0.1455  -6.995 2.64e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

Spring and short shoots decrease the probability of success.

Case Control

- Rather than sampling from the population, enrich the number of 'Cases' (usually units with problems)
- Useful when:
  - Cases are rare AND
  - Measuring the predictor variables is expensive (time, money, destructive testing).
- β_1 to β_k are the same (and more accurately estimated) as in the case where a population sample is taken. Useful for understanding what predictors are important
- β_0 is not the same... cannot compute probabilities that pertain to samples from the population.

An industrial example

- Taken from “Effective Monitoring of Processes with Parts Per Million Defective: A Hard Problem!”
- Exhaust valve seats force fitted by insertion into the head of an engine. Four per engine.
- Expected defective rate is less than 50 PPM (cases rare).
- Quality of fit judged by visual inspection using feeler gauges (inspection expensive).
- 25 defective seat insertions collected over time; other three in each engine used as controls. (“Pareto analysis” showed no evidence poor fit depends on location.) 100 observations total.

An industrial example

- Eleven explanatory variables (Measurements of force taken during the automated insertion process, dimensional and physical characteristics of the valve seat and pocket in the head where the seat is inserted).
- Logistic regression used to predict defectives; three variables identified as important predictors (Inference focused on $\beta_1 \ldots$)
- Quality control began charting these three continuous variables (x-bar chart, s-chart) to improve the quality of this part of the process.

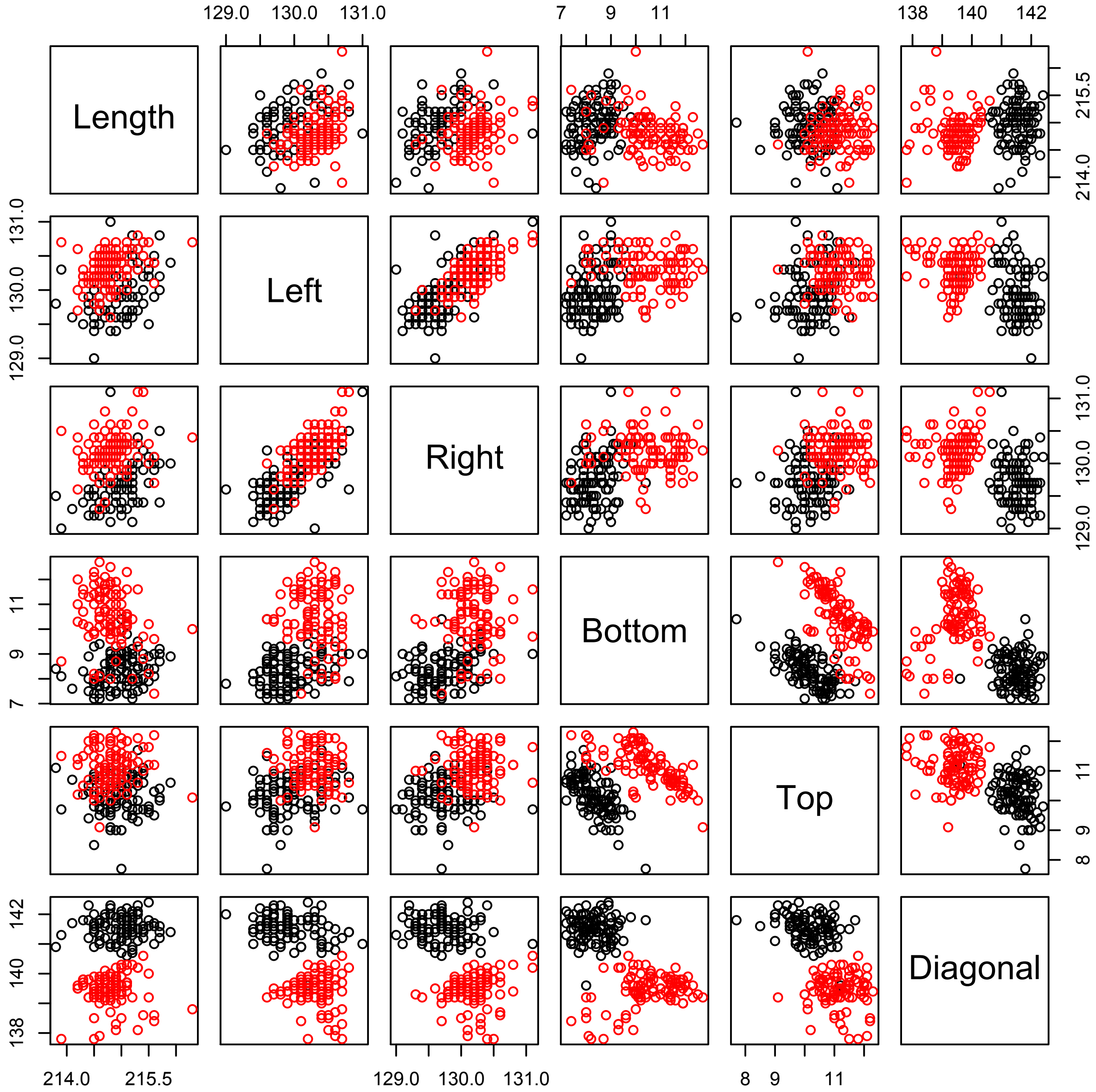
Example with data

- 100 counterfeit and 100 legitimate swiss banknotes.
- 6 measurements (Length, Width, Diagonal, border widths) to use as predictors

```
banknote <- read.table ("Data/banknote.txt", header=TRUE)
summary (banknote)
```

```
    Length           Left           Right           Bottom
Min.   :213.8   Min.   :129.0   Min.   :129.0   Min.   : 7.200
1st Qu.:214.6   1st Qu.:129.9   1st Qu.:129.7   1st Qu.: 8.200
Median :214.9   Median :130.2   Median :130.0   Median : 9.100
Mean   :214.9   Mean   :130.1   Mean   :130.0   Mean   : 9.418
3rd Qu.:215.1   3rd Qu.:130.4   3rd Qu.:130.2   3rd Qu.:10.600
Max.   :216.3   Max.   :131.0   Max.   :131.1   Max.   :12.700
     Top           Diagonal            Y
Min.   : 7.70   Min.   :137.8   Min.   :0.0
1st Qu.:10.10   1st Qu.:139.5   1st Qu.:0.0
Median :10.60   Median :140.4   Median :0.5
Mean   :10.65   Mean   :140.5   Mean   :0.5
3rd Qu.:11.20   3rd Qu.:141.5   3rd Qu.:1.0
Max.   :12.30   Max.   :142.4   Max.   :1.0
```

Predictor Variables



Model Fitting

- Model fitting algorithm does not converge for the model with all 6 predictors.
- Try forward stepwise. Model with `Diagonal` and `Bottom` selected.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   99422.9  5433597.5   0.018    0.985
Diagonal       -751.8    41093.2  -0.018    0.985
Bottom          688.7    37796.3   0.018    0.985
```

- Perfect separation, unstable fit.

Model Fitting

- Try the two variables that give the best visual separation in the pairs plot, `Top` and `Diagonal`.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  884.197    215.637   4.100 4.12e-05 ***
Diagonal      -6.450      1.544  -4.178 2.94e-05 ***
Top            2.016      1.326   1.520    0.129
```

- Not clear 'top' is helping. Model with just Diagonal:

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  989.547    219.465   4.509 6.52e-06 ***
Diagonal      -7.044      1.562  -4.508 6.54e-06 ***
```

Model Fitting

- How to visualise the difference?.
- p=0.5 would be a reasonable place to change decision about the note being real/counterfeit. What does this look like in predictor space?
- p=0.5 means logit(p)=0.

  $$0 = 884.197 - 6.450 \times Diagonal + 2.016 \times Top;$$

  rearranging we get

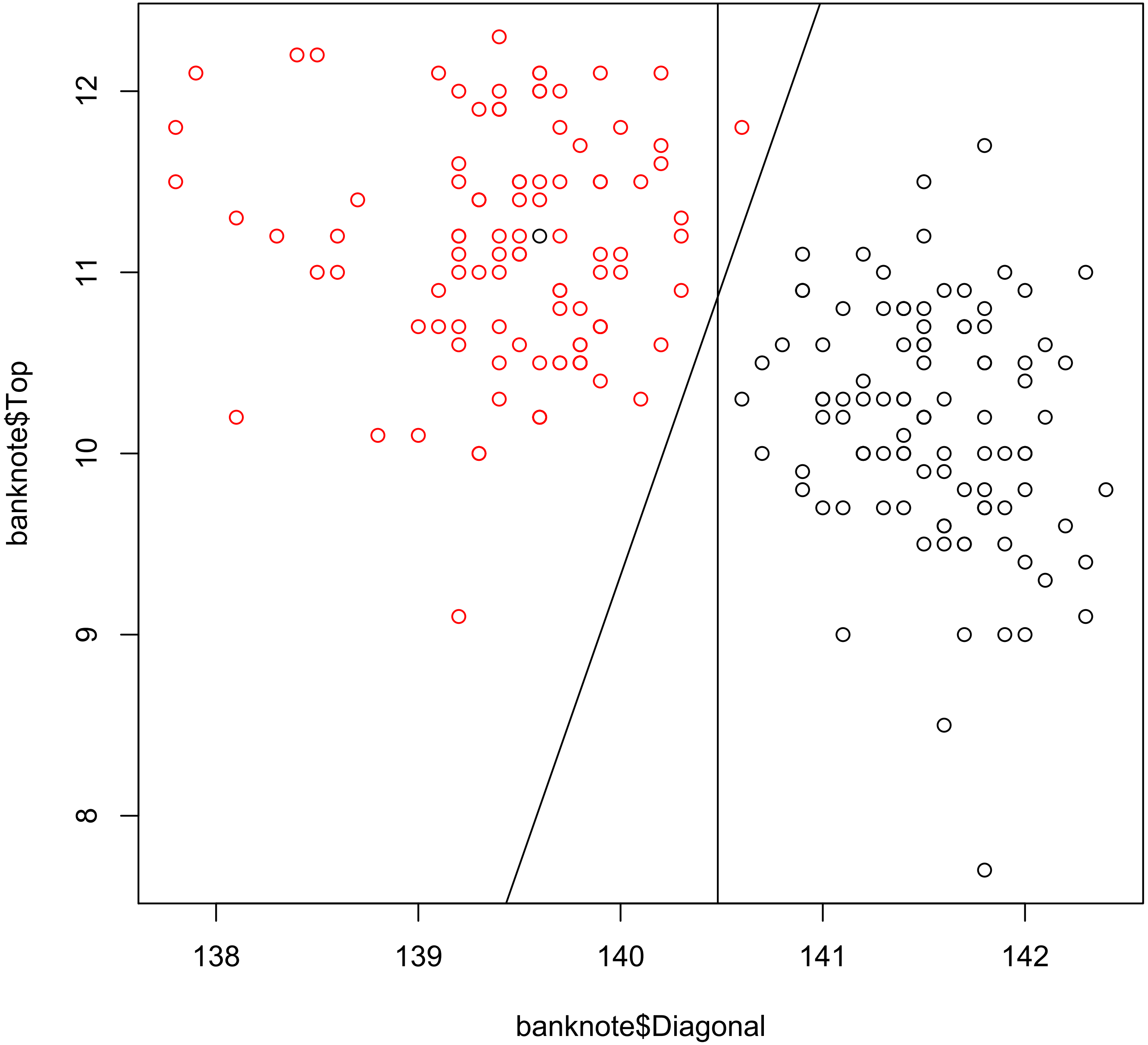
  $$Top = -438.5898 + 3.199405 \times Diagonal.$$

- For the singe variable model:

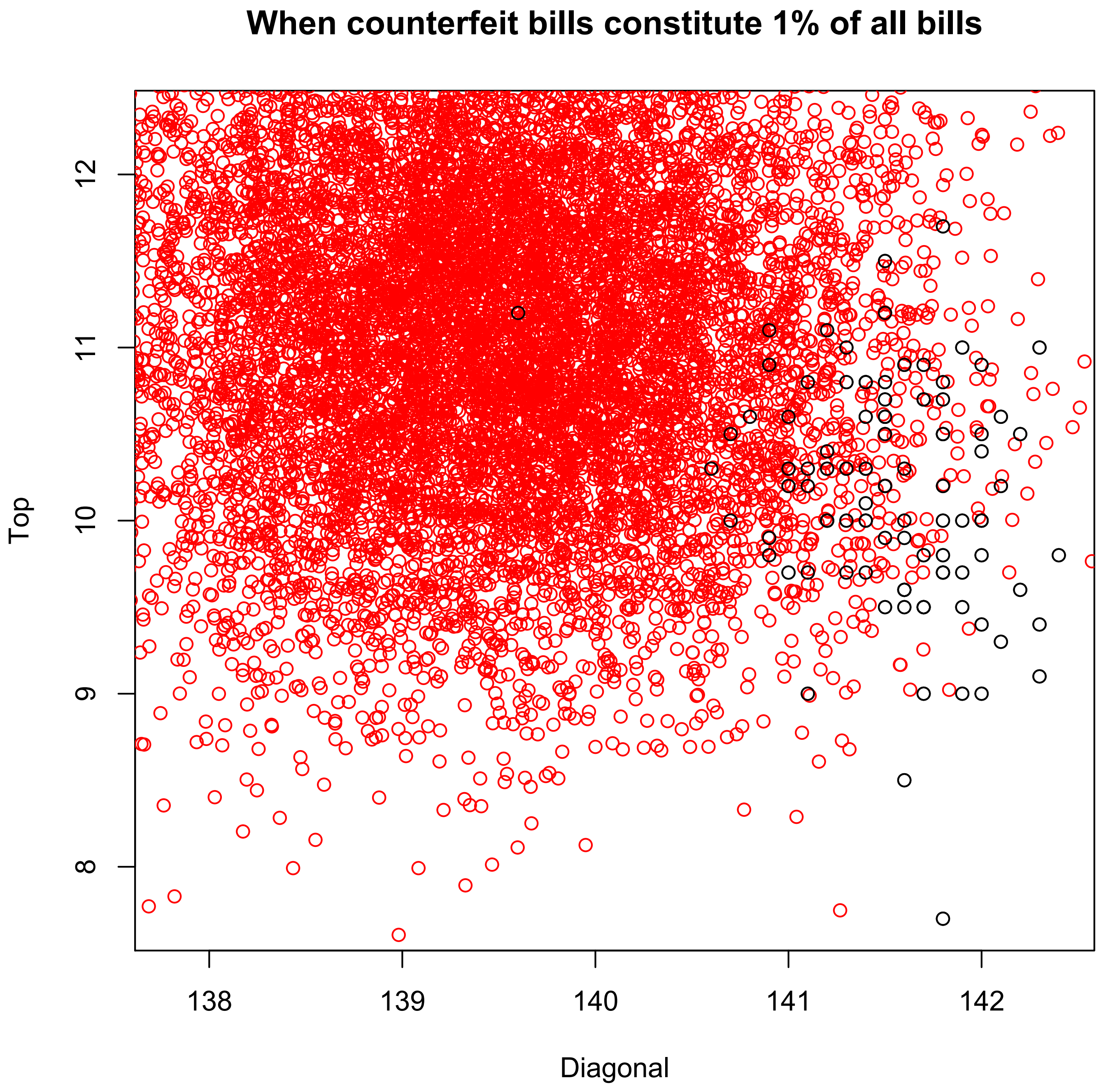
  $$0 = 989.547 - 7.044 \times Diagonal$$

  giving $Diagonal = 140.48$. Warnings about "fitted probabilities of 0 or 1."

Model Fitting



Back in real life...


When counterfeit bills constitute 1% of all bills
Top
Diagonal