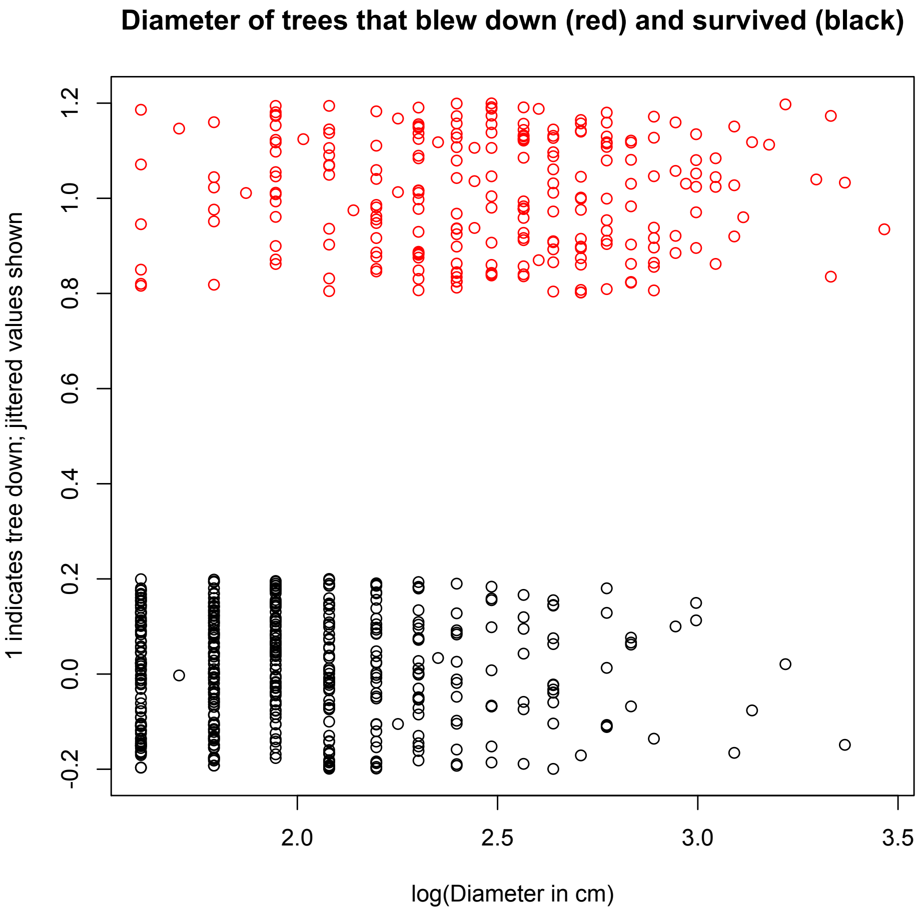
228.371: Logistic Regression

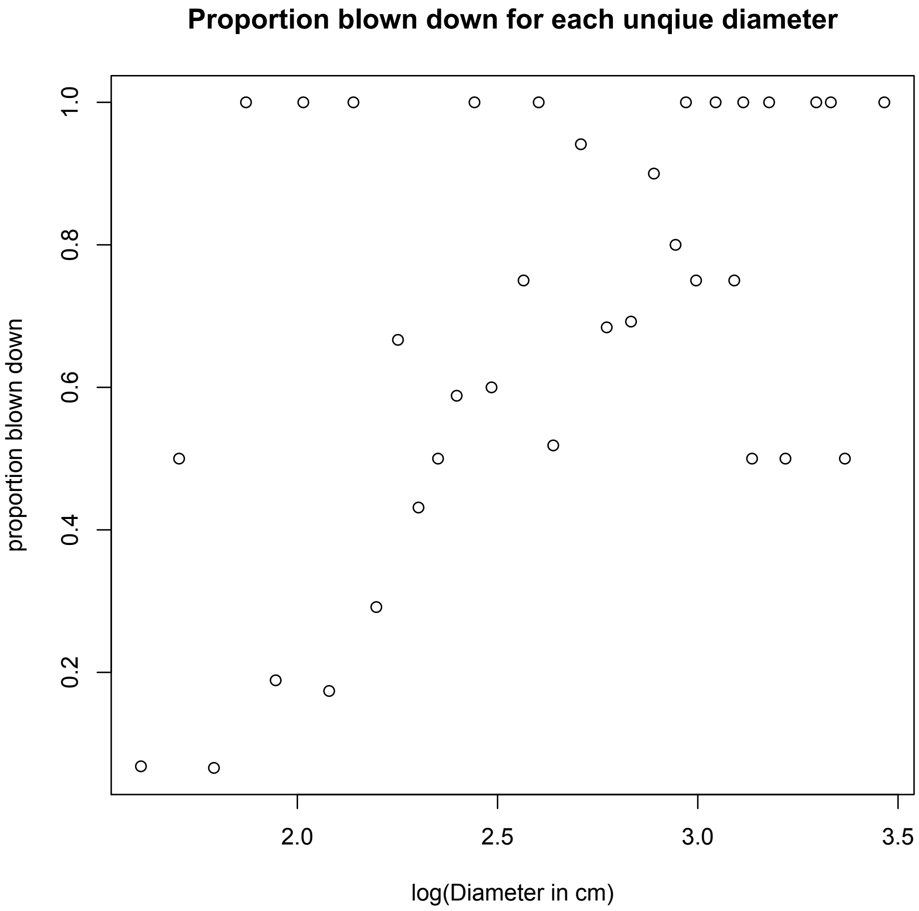
Week 8

1 Motivation

Consider the *blowdown* data, which records, for each of 659 Balsam Fir trees from the Boundary Waters Canoe area wilderness, the tree's diameter, and whether the tree blew down in a 1999 storm with 90mph winds (with a 1 indicating the tree was blown down, and a 0 indicating it was not). The data have been jittered so that you can get an idea of the density of points at each coordinate pair. Is a linear model suitable?

What if we record a proportion for each diameter range? Note that each point now represents a different number of observations. What do we know about the variance of sample proportions, and how does it compare with the linear regression assumptions? What about proportions that are observed as 0 or 1?


Diameter of trees that blew down (red) and survived (black)
1 indicates tree down; jittered values shown
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
2.0
2.5
3.0
3.5
log(Diameter in cm)


Proportion blown down for each unqiue diameter
proportion blown down
0.2
0.4
0.6
0.8
1.0
2.0
2.5
3.0
3.5
log(Diameter in cm)

2 Generalized linear models

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

$$E(Y_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} ...$$

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

$$g(E(Y_i)) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} ...$$

When Y is a dichotomous (0/1) variable, $E(Y_i)$ is the proportion of 1's (successes). We want a function that will map p falling into the (0,1) interval (left hand side of the equation) to the real line (right hand side of the equation). The most common choice of link, and the one we will focus on, is the *Logit function*:

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

3 Model Fitting

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

$$\prod_{i=1}^{n} p_i^{Y_i}(1-p_i)^{1-Y_i}$$

$$logit(p_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} ...$$

This is called "maximum likelihood." In practice, the maximization is done by an iterative procedure that stops when change at each iteration drops below some threshold (note R output about number of Fisher scoring iterations). For the *blowdown* data:

```
> summary(glm(y~log(D), family=binomial))

Call:
glm(formula = y ~ log(D), family = binomial)

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(D)        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
```

Asymptotically (with large amounts of data), the maximum likelihood estimates have a normal distribution, which is where the z-scores and p-values given in the table come from.

In the previous model fit, each observation is treated as if it has a unique x value. We can see this from the null deviance degrees of freedom, which equals the number of data points-1, indicating the saturated model has a separate p_i for each data point. This is what we expect when the data are continuously measured–eg, if the diameter were measured to the nearest millimeter. In fact, in this data there are several observations for most$x$ values (diameters are recorded to the nearest half centimeter) so we can aggregate over predictor classes. R will take the data as a table of ‘successes’ (1’s) and ‘failures’ (0’s).

```
>table(D,y)[,2:1]->successes
>uniqueD<-sort(unique(D))
>mod3<-glm(successes~log(uniqueD), family=binomial)
> summary(mod3)

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
```

Note the estimates and standard errors are identical when the data are summarized in this way. The difference between the null and residual deviance measures the utility of our model; while the raw values are different, their difference (and the corresponding difference in degrees of freedom) are the same.

4 Model interpretation

Consider increasing log diameter from 1 to 2. How does the probability of being knocked over in a storm increase? According to our model, for log(diameter)= 1 this probability is:

```
> exp(-7.8925+3.2643)/(1+exp(-7.8925+3.2643))
[1] 0.009677759
```

and at log(diameter)=2

```
> exp(-7.8925+3.2643*2)/(1+exp(-7.8925+3.2643*2))
[1] 0.2036072
```

(Note that the interpretation of these numbers as probabilities (or *risks*) for trees of a certain diameter depends on the study being a sample from the population of all trees at a particular x-value. More on this later.)

However, if we consider increasing the diameter to three, the change in probability is is different, whether considered as the raw increase or % increase.

```
>  exp(-7.8925+3.2643*3)/(1+exp(-7.8925+3.2643*3))
[1] 0.8699368
```

The graph relating the probability of blowdown to the log diameter looks like this:

What is increasing linearly is the *log odds*:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1$$

The odds

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

are multiplied by a factor of $\exp(\beta_1)$ for each unit of increase of X_1.

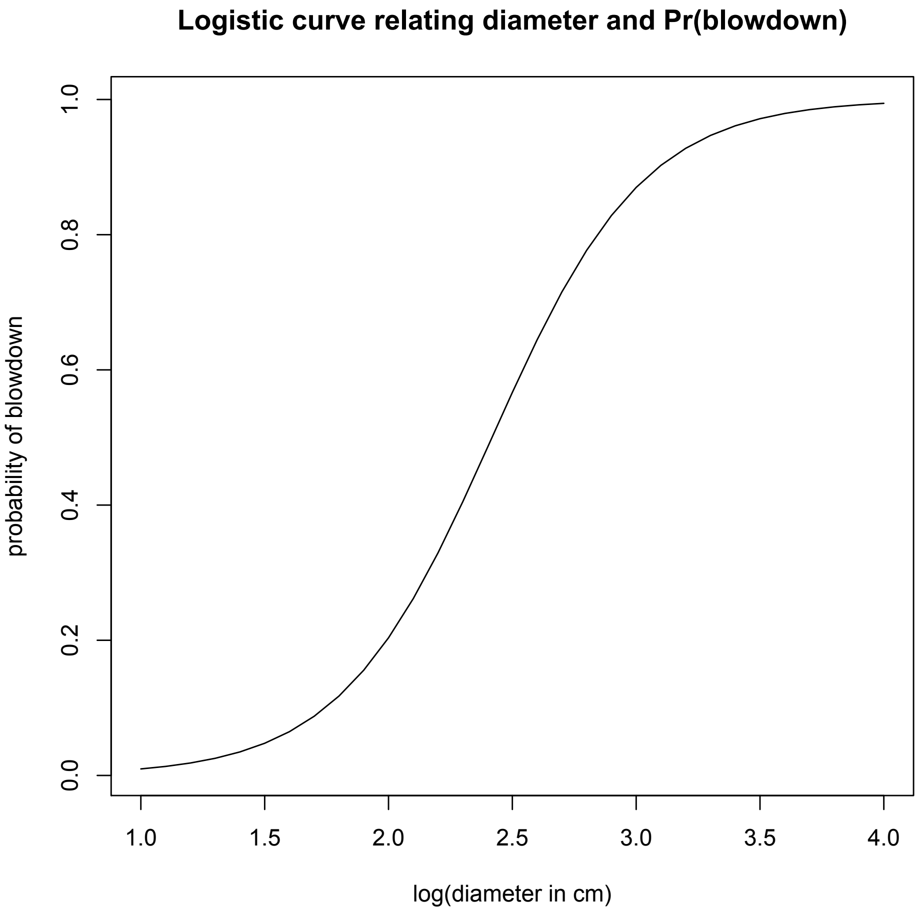
5 Residuals

The command plot(model) produces a series of four plots that are closely related to those produced for linear regression. However, these plots are much more useful if our data are treated as a proportion of successes for each x value. Compare the following, produced for the two model fits above:

When the x-values are treated as unique, one trend is traced out for the points where $Y = 1$, and another where $Y = 0$. The other fitting method is much preferred when possible.

When the predicted values are shown in these plots, they are the predicted logit(p_i), not p_i–thus their values outside the 0-1 interval! The plots show two different types of residuals, *deviance residuals* and *Pearson residuals.* Deviance residuals (or some function of them) are used in the first three plots. Deviance residuals make use of the fact that the deviance can be thought of as a sum over each observation:

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


Logistic curve relating diameter and Pr(blowdown)
probability of blowdown
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
log(diameter in cm)

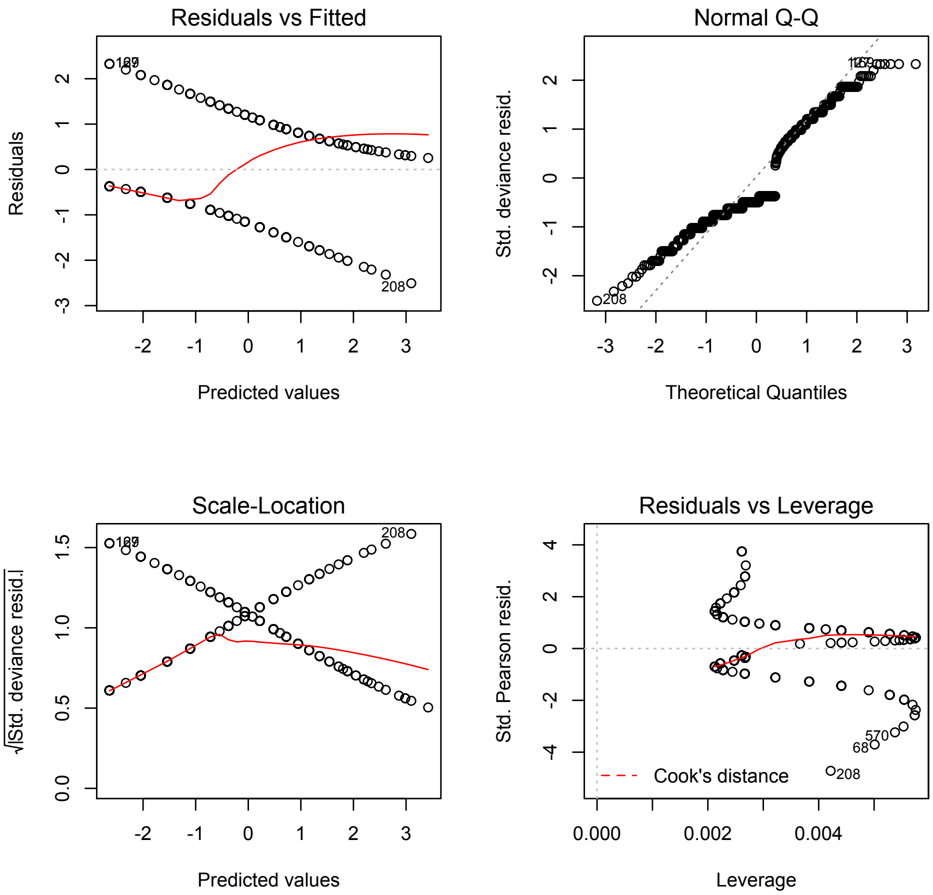


Figure 1: Residual plots when x values are treated as continuous and unique.

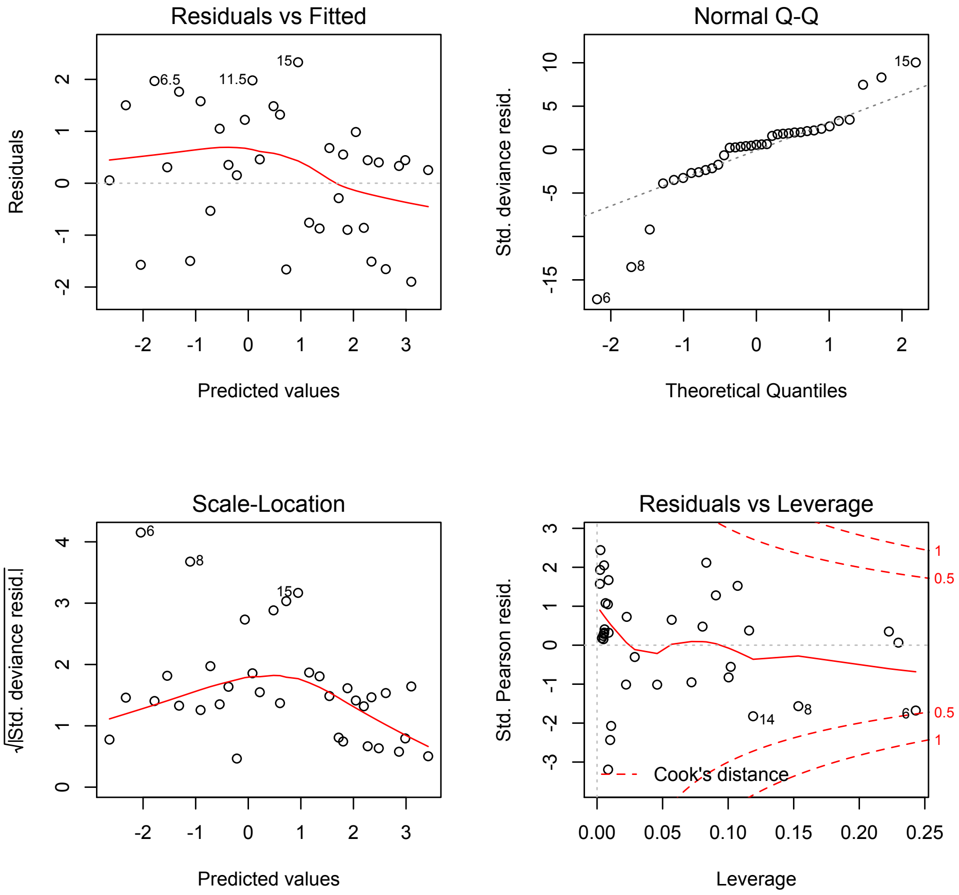


Figure 2: Residual plots when the responses have been aggregated over common x values

or we can aggregate over observations with the same set of X-values, and treat our data as the

$$\sum_{1}^{m} -2\times[y_j \times (log(\hat{\mu}_j) - y_j/n_j) + (n_j - y_j)\,(log(1-\hat{\mu}_j) - log((n_j - y_j)/n_j))]$$

We then take the square root of the individual terms and give them the sign associated with the conventional residual. Plots with these residuals that show an outlier or trend suggest problems with the fit to (respectively) single data points or the data as a whole. The standardized deviance residual accounts for the fact that the points do not all have the same number of observations. Thus, although the deviance residual for D=6 is -1.57, similar to other values, it represents many observations (91) so when standardized it shows up as an outlier.

Standardized Pearson residuals are similar to the standardized residuals for linear regression. However, because the variance changes depending on p and n_x in a logistic model, the factor used to standardize changes for each point:

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

where $logit(\mu_i) = \beta_0 + \beta_1 x_1 i + \ldots$. Pearson's residuals may have a skewed distribution, but should be free of obvious trends (including increases in variance) when plotted against predictors, fitted values, or (more frequently) the fitted values on the linear scale.

To get these residuals in R we can use the commands:

```
>residuals(mod3)->resid.dev  # default is the deviance residuals
>residuals(mod3, type="pearson")->resid.pearson
```

Note that these are **not** the residuals stored in the model object.

The Leverages and Cooks distances shown in the fourth panel are interpreted in the same way as their linear regression counter parts. However, their computation has been modified to account for the differing variances of different observations. In particular, extreme X values, which give values of p close to zero or one, are down-weighted, so unlike their linear regression counter parts they are not particularly influential.

6 Model Utility, Model Adequacy

The “null deviance” given in the model summary is the deviance when only the intercept is used in the model. If the larger model does not improve on this null model, the difference in deviances will have a Chi-squared distribution with degrees of freedom equal to the difference in degrees of freedom for the two deviances.

```
Null deviance: 250.856  on 34  degrees of freedom
Residual deviance:  49.891  on 33  degrees of freedom
> pchisq(250.856-49.891,34-33, lower=FALSE)
[1] 1.286026e-45
```

Thus, we have convincing evidence that our predictor improves on the null model.

When we have multiple observations per x-value, the ‘saturated’ model is also a valid model that can be considered. For the kth x value, the saturated model sets $p_k = \sum_{i \in x_i = k} Y_i / n_k$, providing a perfect fit. We can imagine treating our model as the null model, and seeing if the saturated model (deviance 0, degrees of freedom 0) makes a significant improvement.

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

Here a big value (larger than 0.05) would suggest our model fits well. We are not quite there for this data set.

7 Multiple Logistic Regression

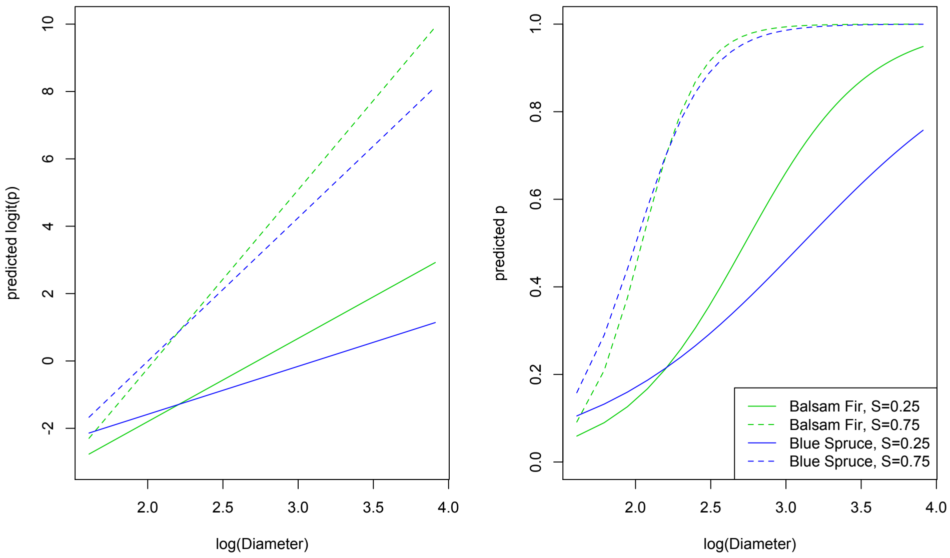
We now consider a larger data set related to tree blow-down, which includes 2 species (given in the SPP variable), and also data on ’local severity’ of the storm: the percentage of trees in the surrounding area that are down. As in linear regression, 2e can use “step” (in this case backwards stepwise) to start with the largest model we would like to consider and remove unnecessary terms.

```
> step(glm(y~log(D)*SPP*S ,family='binomial', data=blow.sub))->blowmod
Start:  AIC=1485.73
y ~ log(D) * SPP * S
```

```
               Df Deviance    AIC
- log(D):SPP:S  1   1470.8 1484.8
<none>              1469.7 1485.7

Step:  AIC=1484.8
y ~ log(D) + SPP + S + log(D):SPP + log(D):S + SPP:S

             Df Deviance    AIC
- SPP:S       1   1470.8 1482.8
<none>            1470.8 1484.8
- log(D):SPP  1   1478.6 1490.6
- log(D):S    1   1512.3 1524.3

Step:  AIC=1482.84
y ~ log(D) + SPP + S + log(D):SPP + log(D):S

             Df Deviance    AIC
<none>            1470.8 1482.8
- log(D):SPP  1   1478.7 1488.7
- log(D):S    1   1517.0 1527.0
```

It appears we can get away without the three way interaction, and without the severity/species interaction. But what do the remaining terms tell us? As with the complicated response surface models we looked at previously, looking at some predicted values can give us a sense of what's happening. Let's look at a range of diameter values, for each of the two different species, and with 'severity' fixed at 0.25 and 0.75.

```
>newdata.BF<-data.frame(D=rep(seq(5, 50, 1),2), S=rep(c(0.25, 0.75),
each=46), SPP=rep("BF", 46*2))
>newdata.BS<-data.frame(D=rep(seq(5, 50, 1),2), S=rep(c(0.25, 0.75),
each=46), SPP=rep("BS", 46*2))
>predBF.lin<-predict.glm(blowmod, newdata.BF)
>predBF.prob<-predict.glm(blowmod,newdata.BF, type='response')
>predBS.lin<-predict.glm(blowmod, newdata.BS)
>predBS.prob<-predict.glm(blowmod,newdata.BS, type='response')
> range(c(predBF.lin,predBS.lin))
[1] -2.766544  9.918014
```



```
> range(c(predBF.prob,predBS.prob))
[1] 0.05915907 0.99995072
> par(mfrow=c(1,2))
> plot(log(newdata.BF$D[1:46]),predBF.lin[1:46], type=’l’,
ylim=c(-3,10), xlab=’log(Diameter)’, ylab=’predicted logit(p)’,col=3)
> lines(log(newdata.BF$D[47:92]),predBF.lin[47:92], lty=2,col=3)
> lines(log(newdata.BS$D[47:92]),predBS.lin[47:92], lty=2,col=4)
> lines(log(newdata.BS$D[1:46]),predBS.lin[1:46],col=4)
> plot(log(newdata.BF$D[1:46]),predBF.prob[1:46],type=’l’,
 ylim=c(0,1), xlab=’log(Diameter)’, ylab=’predicted p’,col=3)
> lines(log(newdata.BF$D[47:92]),predBF.prob[47:92], lty=2,col=3)
>lines(log(newdata.BS$D[47:92]),predBS.prob[47:92], lty=2,col=4)
>lines(log(newdata.BS$D[1:46]),predBS.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"))
```

8 Analysis of Variance

The following experiment was performed to look at how soaking time and heating time affected the probability that a steel ingot was ready to roll. There are three different Heating Times and five different soaking times.

```
   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
```

As in a linear analysis of variance , we first want to assess whether there is an interaction between the factors.

```
> mod1<-glm(cbind(NotReady,Ready)~as.factor(SoakingTime)+as.factor(HeatingTime),
             data=ingots, family=’binomial’)
> mod2<-glm(cbind(NotReady,Ready)~as.factor(SoakingTime)*as.factor(HeatingTime),
             data=ingots, family=’binomial’)
> anova(mod1, mod2)
Analysis of Deviance Table

Model 1: cbind(NotReady, Ready) ~ as.factor(SoakingTime) + as.factor(HeatingTime)
Model 2: cbind(NotReady, Ready) ~ as.factor(SoakingTime) * as.factor(HeatingTime)
  Resid. Df Resid. Dev Df Deviance
1         8     8.3763
2         0     0.0000  8   8.3763
> pchisq(8.3763,8,lower=FALSE)
[1] 0.3976013
```

Note that the deviance is zero for the larger model (with interaction); this is always the case, as a separate value of p is computed for each combination of x values. In this case there is no evidence of an interaction. We now examine the individual factors:

```
> anova(mod1)
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NotReady, Ready)

Terms added sequentially (first to last)


                       Df Deviance Resid. Df Resid. Dev
NULL                                      14    17.1738
as.factor(SoakingTime)  4   2.1901        10    14.9837
as.factor(HeatingTime)  2   6.6074         8     8.3763
> pchisq(6.6,2,lower=FALSE)
[1] 0.03688317
> pchisq(2.19,4,lower=FALSE)
[1] 0.7008605
```

Soaking time does not appear to be important, and heating time does. Note tht the deviance differences and corresponding p-values depend on the order in which the terms are considered. The test for Heating time vs the null model is not quite significant.

```
> anova(mod3)
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(NotReady, Ready)

Terms added sequentially (first to last)
```

```
                       Df Deviance Resid. Df Resid. Dev
NULL                                      14     17.174
as.factor(HeatingTime)  2   5.8856        12     11.288
> pchisq(5.8856,2, lower=FALSE)
[1] 0.05271791
```

We can also try using heating time as a linear predictor rather than a factor, which will gain us a degree of freedom. In this case we will want to examine the model for lack of fit.

```
> summary(mod4)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3887  -0.8000  -0.3319  -0.1202   1.9658

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.19464    1.44324  -4.292 1.77e-05 ***
HeatingTime  0.11618    0.05838   1.990   0.0466 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 17.174  on 14  degrees of freedom
Residual deviance: 11.883  on 13  degrees of freedom
AIC: 27.478

Number of Fisher Scoring iterations: 6

> pchisq(17.174-11.883,1, lower=FALSE)
[1] 0.02143591
> pchisq(11.883, 13, lower=FALSE)
[1] 0.5372605
```

Table 1: Plum Tree Data.

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

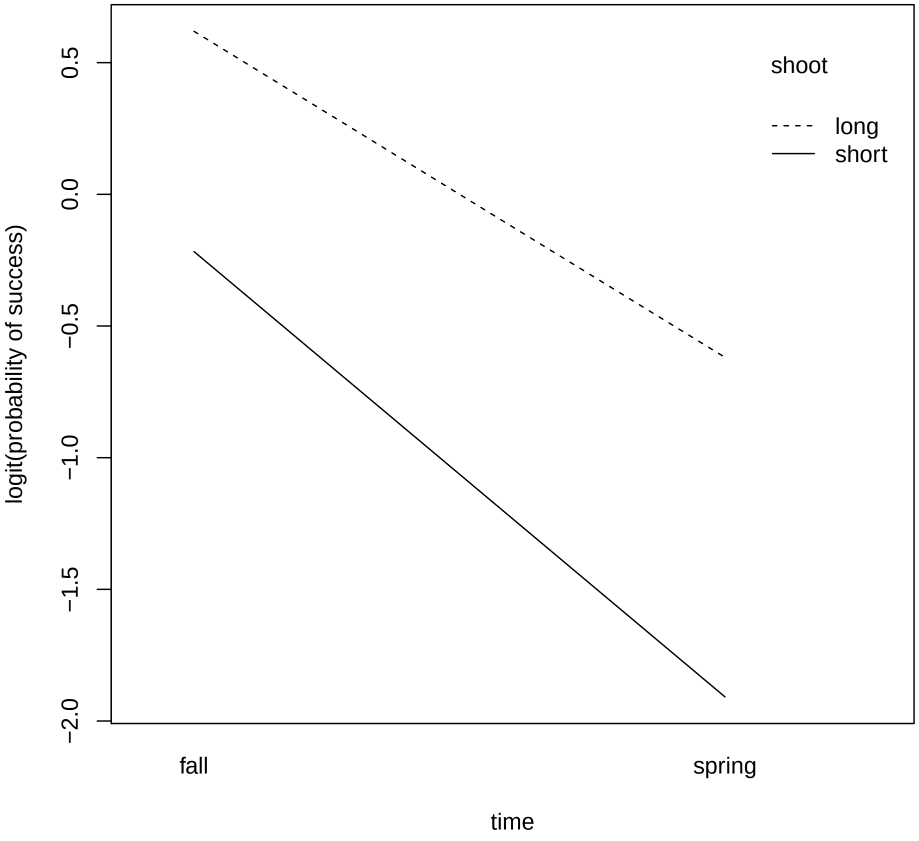
In this formulation, Heating time is a significant predictor whether using the Wald test (in the table) or a likelihood ratio test–note however the p-values are different. The large p-value for the residual deviance suggests that the fit is adequate.

When there are some successes and some failures under each set of conditions, we can visulize the interaction as we do for a linear regression, using the logit of the observed probabilities (the problem with conditions that give all successes or all failures is that the logit is infinity or negative infinity). We will consider a further example where this is possible.

Plum trees are propagated by grafting onto a root stock (thus the plum producing part is essentially a clone of the original tree). The following experiment considers how two factors affect the success of grafting: the length of the shoot that is grafted, and the time of year the grafting is done. Table 1 gives the number of trials and successes under each condition. Note that the model with an interaction will always have deviance 0: a different proportion has been fit for each line in the table above.

```
> 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"))
> summary(glm(plum~shoot*time, family="binomial"))

Call:
glm(formula = plum ~ shoot * time, family = "binomial")

Deviance Residuals:
[1]  0  0  0  0
```

```
Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)             0.6190     0.1353   4.574 4.78e-06 ***
shootshort             -0.8366     0.1876  -4.460 8.19e-06 ***
timespring             -1.2381     0.1914  -6.469 9.87e-11 ***
shootshort:timespring  -0.4527     0.3009  -1.505    0.132
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1.5102e+02  on 3  degrees of freedom
Residual deviance: 7.8160e-14  on 0  degrees of freedom
AIC: 30.742

Number of Fisher Scoring iterations: 3
> summary(glm(plum~shoot+time, family="binomial"))

Call:
glm(formula = plum ~ shoot + time, family = "binomial")

Deviance Residuals:
      1        2        3        4
-0.6966   0.6966   0.6642  -0.9393

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.7138     0.1217   5.867 4.45e-09 ***
shootshort   -1.0177     0.1455  -6.995 2.64e-12 ***
timespring   -1.4275     0.1465  -9.747  < 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: 151.0193  on 3  degrees of freedom
Residual deviance:   2.2938  on 1  degrees of freedom
AIC: 31.036
```



```
Number of Fisher Scoring iterations: 3
```

Again it appears the interaction is not necessary, and that both factors are significant. This can be visualized with an interaction plot; note that this is on the transformed scale:

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

9 Case-Control Data

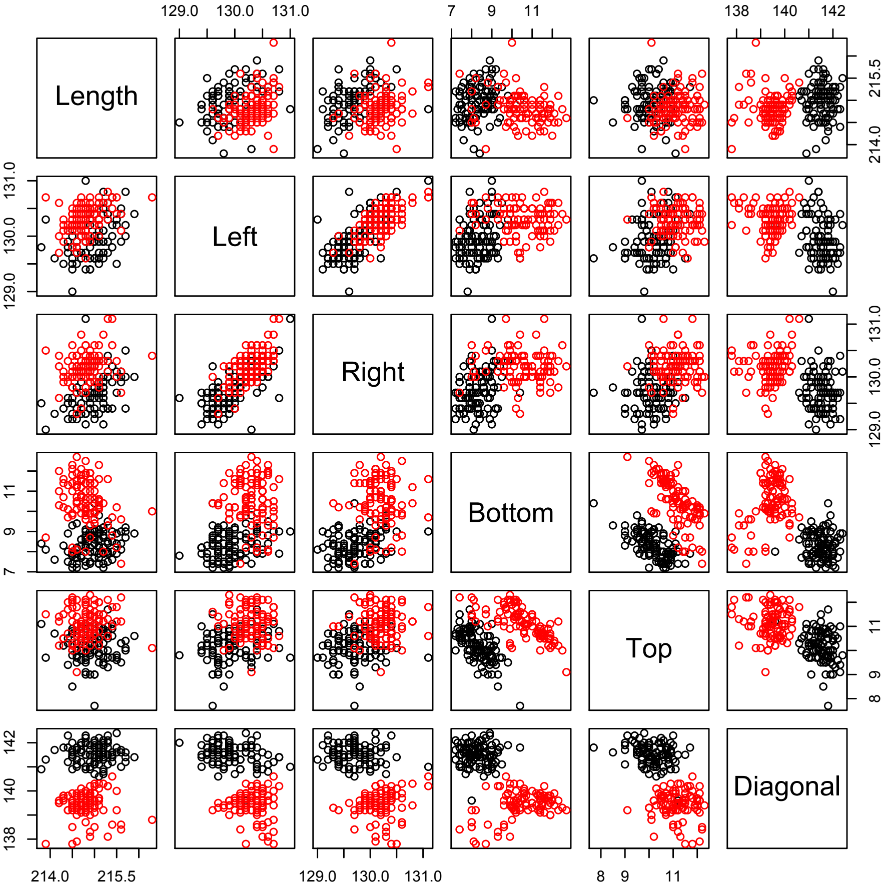
Sometimes, the events we are interested in are rare events. In this case, it can be difficult to obtain precise estimates of β_1; that was one of the difficulties with the 'ingots ready for rolling' example. One way of dealing with this is to enrich the sample with the events of interest; for instance, one interesting event for every normal event. However, we then must be cautious in interpreting the result. In this situation, $\beta_1 \dots \beta_k$ will be the same as for a random sample of data, and their estimates will be more accurate. We will be able to make accurate decisions about which predictors are important, and how much influence they have. However, β_0 will be different–very different–from the random sampling case, and the fitted probabilities will **not** reflect the probability of seeing the event of interest in the original population.

A data set of this structure is in `banknote.txt`, which contains 6 measurements on 100 counterfeit swiss banknotes (Y=0) and 100 genuine banknotes (Y=1). Exploratory analysis on this data might include pairwise scatterplots of the variables, with the real and counterfeit notes in different colors:

```
pairs(banknote[,1:6],col=banknote$Y+1)
```

We see that it seems like it should be possible to separate the two groups. It also seems like not all variables will be necessary to do this. In fact, trying to fit all six predictors leads to convergence problems. The final model selected by forwards stepwise selection achieves perfect separation, but also looks unstable–note the very large coefficients and very large standard errors.

```
> summary(mod1)

Call:
glm(formula = Y ~ Diagonal + Bottom, family = "binomial", data = banknote)

Deviance Residuals:
      Min          1Q      Median          3Q         Max
-5.51e-04  -2.00e-08   0.00e+00   2.00e-08   6.58e-04

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
```


Length
Left
Right
Bottom
Top
Diagonal
129.0
130.0
131.0
7
9
11
138
140
142
215.5
214.0
214.0
215.5
129.0
130.0
131.0
8
9
11

```
Bottom          688.7    37796.3   0.018    0.985

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2.7726e+02  on 199  degrees of freedom
Residual deviance: 1.0424e-06  on 197  degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 25
```

The panel with Diagonal and Top has the clearest separation so we will try that model.

```
> summary(mod2)

Call:
glm(formula = Y ~ Diagonal + Top, family = "binomial", data = banknote)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5749  -0.0275  -0.0003   0.0464   0.7423

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
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.259  on 199  degrees of freedom
Residual deviance:  18.187  on 197  degrees of freedom
AIC: 24.187

Number of Fisher Scoring iterations: 9
```

It’s not clear that including `top` is improving the model. The model with just diagonal as a predictor gives:

```
> summary(mod0)

Call:
glm(formula = Y ~ Diagonal, family = "binomial", data = banknote)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5320  -0.0397  -0.0008   0.0440   1.5341

Coefficients:
            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 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.259  on 199  degrees of freedom
Residual deviance:  21.109  on 198  degrees of freedom
AIC: 25.109

Number of Fisher Scoring iterations: 9
```

Imagine classifying the points from this data set. The fitted p for each point is the probability (in this 50/50 data set) that the note is genuine. Let’s draw a line on the plot where the fitted $p = 0.5$; this would be a sensible place to change our decision from ’counterfeit’ to ’genuine.’ Note that logit(0.5)=0, so we are looking for the values of Diagonal and Top where

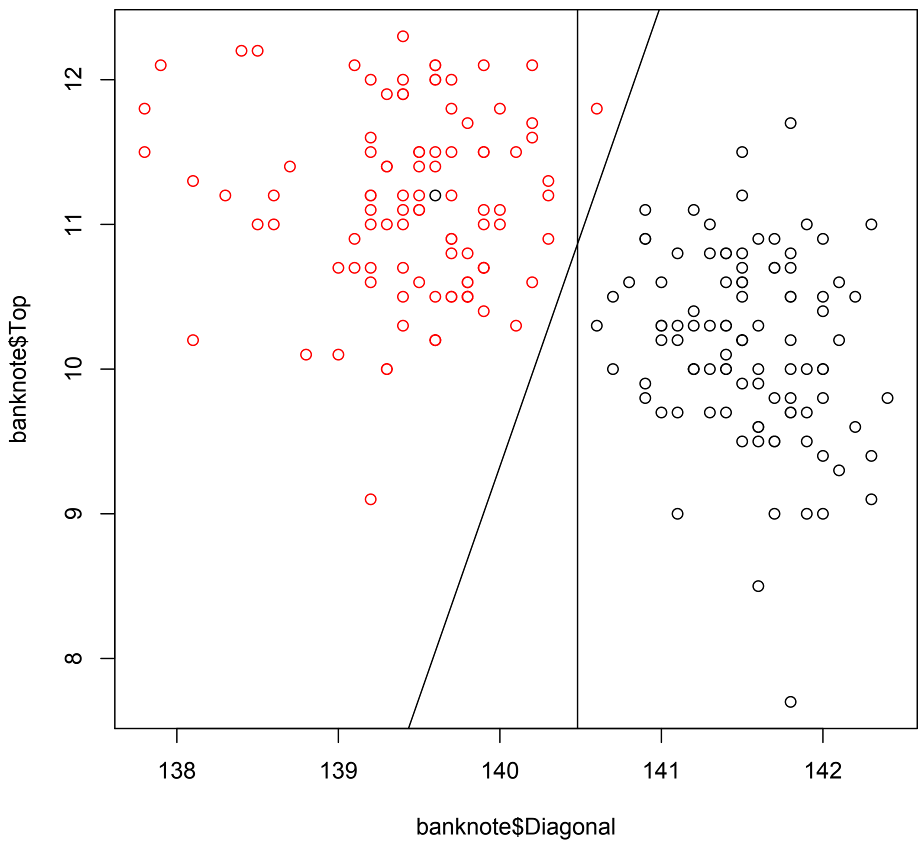
$$0 = 884.197 - 6.450 \times Diagonal + 2.016 \times Top;$$

rearranging we get

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

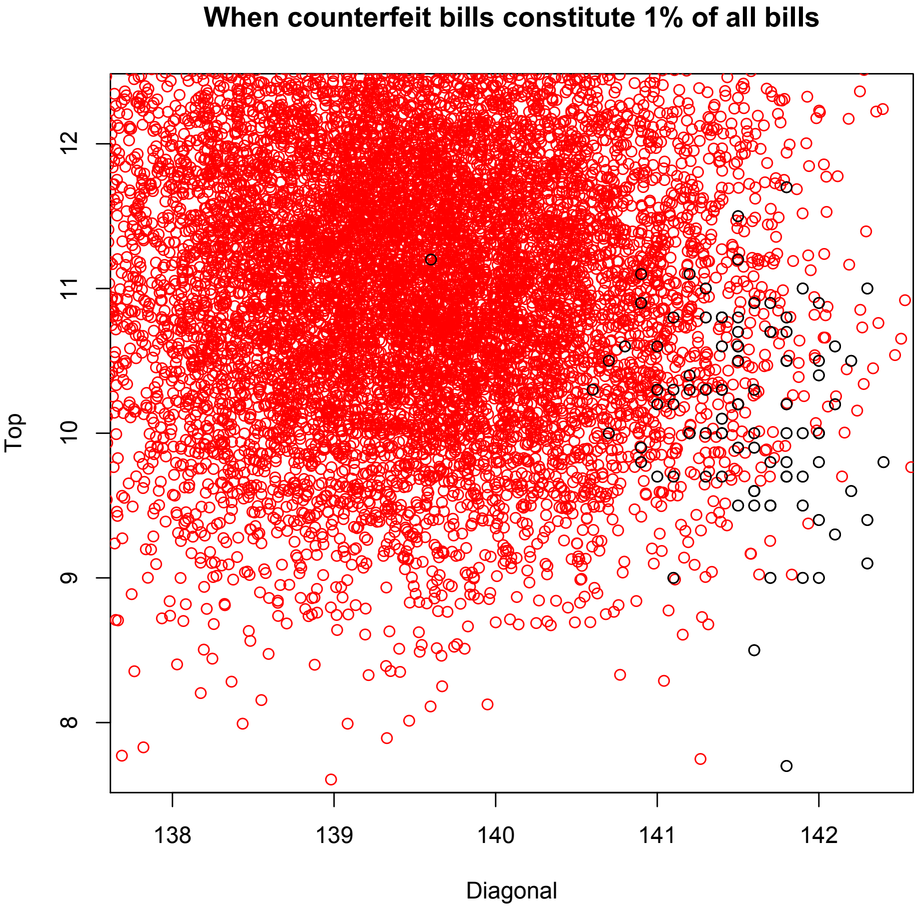
For the single variable model, we solve

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



giving *Diagonal* = 140.48. We now visualize these lines:

It is important to remember the artificial structure of our data. Reality looks more like this:


When counterfeit bills constitute 1% of all bills
Top
Diagonal