5章 GLMの尤度比検定と検定の非対称性

In [1]:

```
d <- read.csv('data3a.csv')
fit1 <- glm(y ~ 1, data=d, family=poisson)
fit2 <- glm(y ~ x, data=d, family=poisson)
fit1$deviance - fit2$deviance
```

4.51394107885181

PB法（パラメトリックブートストラップ法）・・・乱数生成に基づくアプローチ

In [2]:

```
d$y.rnd = rpois(100, lambda = mean(d$y))
fit1 <- glm(y.rnd ~ 1, data=d, family=poisson)
fit2 <- glm(y.rnd ~ x, data=d, family=poisson)
fit1$deviance - fit2$deviance
```

2.50757487354798

以上、帰無仮説（一定モデル）を仮定して、逸脱度の差を計算するのが、PB方の1ステップに相当する。これを繰り返すことで、逸脱度の差の分布を得ることができる。

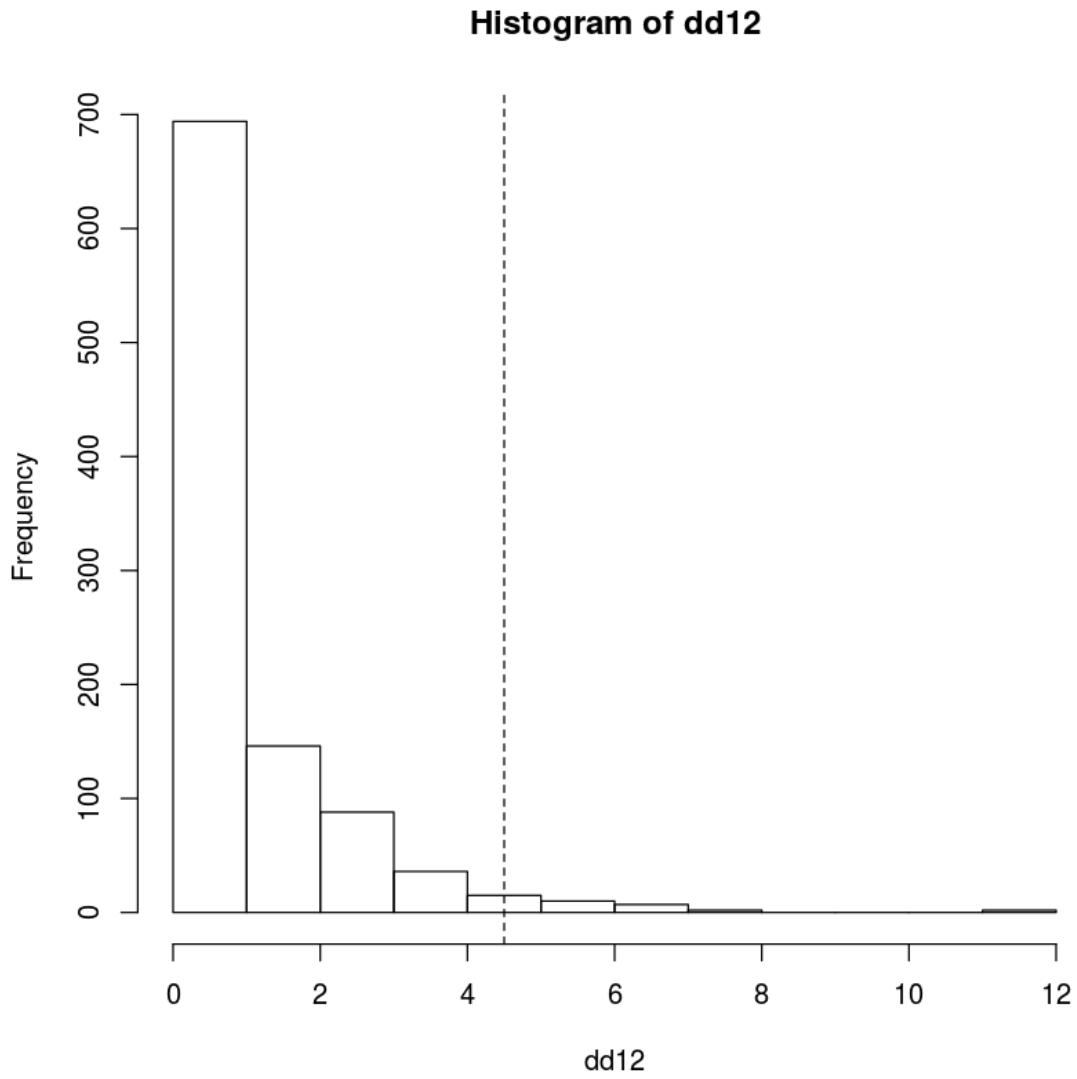
In [3]:

```
get.dd <- function(d) {
  n.sample <- nrow(d)
  y.mean <- mean(d$y)
  d$y.rnd <- rpois(n.sample, lambda = y.mean)
  fit1 <- glm(y.rnd ~ 1, data=d, family=poisson)
  fit2 <- glm(y.rnd ~ x, data=d, family=poisson)
  fit1$deviance - fit2$deviance
}

pb <- function(d, n.bootstrap) {
  replicate(n.bootstrap, get.dd(d))
}

dd12 <- pb(d, n.bootstrap = 1000)
summary(dd12)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
 0.000001  0.088812  0.436136  0.955636  1.290047 11.809232 
```

In [4]:

```
hist(dd12)
abline(v=4.5, lty = 2)
```



In [5]:

```
sum(dd12 >= 4.5)
```

30

In [6]:

```
P <- 28 / 1000
P
```

0.028

P < 0.05 なので、有意差がある。
帰無仮説（一定モデル）は棄却されて x モデルが残るので、これを採択する。

In [7]:

```
quantile(dd12, 0.95)
```

95%: 3.56874973804344

χ^2分布を使った近似計算法

In [8]:

```
fit1 <- glm(y ~ 1, data=d, family=poisson)
fit2 <- glm(y ~ x, data=d, family=poisson)
anova(fit1, fit2, test = "Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
99	89.50694	NA	NA	NA
98	84.99300	1	4.513941	0.03361969

χ^2 分布近似はサンプルサイズが大きい場合に有効な近似計算。

サンプルサイズが小さい場合は、PB法を使ってシミュレーションするのが良い。
等分散正規分布が仮定できる場合には、t分布（平均の差）やF分布（分散比）を
使った方が正確な結果を得ることができる。

ただし、あくまで P 値は P 値であって、決して効果の大きさを言うものではない（それは効果量 effect size の仕事）

6章 GLMの応用範囲をひろげる

In [9]:

```
d <- read.csv('data4a.csv')
summary(d)
d
```

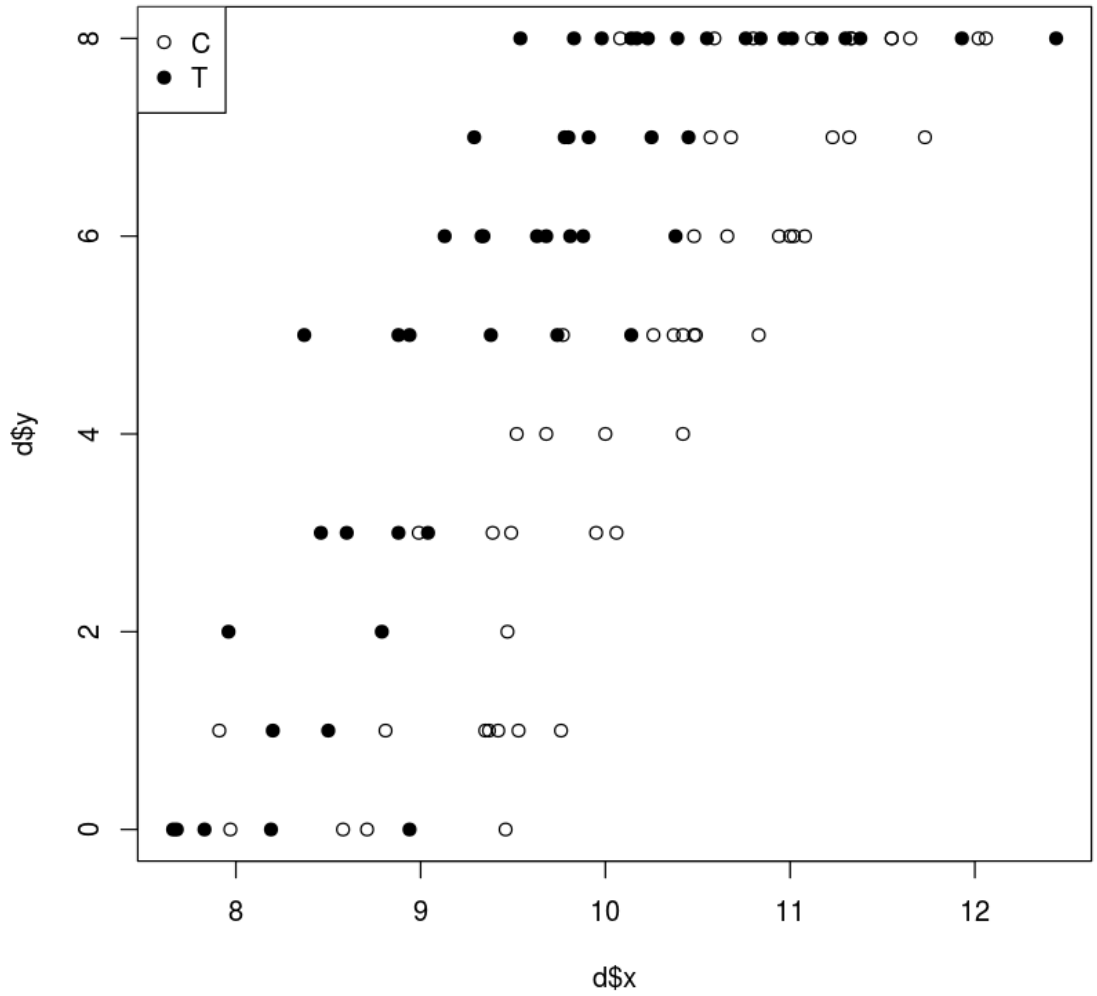
```
     N          y             x         f
Min.   :8   Min.   :0.00   Min.   : 7.660   C:50
1st Qu.:8   1st Qu.:3.00   1st Qu.: 9.338   T:50
Median :8   Median :6.00   Median : 9.965
Mean   :8   Mean   :5.08   Mean   : 9.967
3rd Qu.:8   3rd Qu.:8.00   3rd Qu.:10.770
Max.   :8   Max.   :8.00   Max.   :12.440
```

N	y	x	f
8	1	9.76	C
8	6	10.48	C
8	5	10.83	C
8	6	10.94	C
8	1	9.37	C
8	1	8.81	C
8	3	9.49	C
8	6	11.02	C
8	0	7.97	C
8	8	11.55	C
8	0	9.46	C
8	2	9.47	C
8	0	8.71	C
8	5	10.42	C
8	3	10.06	C
8	6	11.00	C
8	3	9.95	C
8	4	9.52	C
8	5	10.26	C
8	8	11.33	C
8	5	9.77	C
8	8	10.59	C
8	1	9.35	C
8	4	10.00	C
8	1	9.53	C
8	8	12.06	C
8	4	9.68	C
8	7	11.32	C
8	5	10.48	C
8	5	10.37	C
⋮	⋮	⋮	⋮
8	8	9.83	T
8	0	7.66	T
8	6	9.33	T

N	y	x	f
8	1	8.20	T
8	8	9.54	T
8	8	10.55	T
8	6	9.88	T
8	6	9.34	T
8	6	10.38	T
8	6	9.63	T
8	8	12.44	T
8	8	10.17	T
8	7	9.29	T
8	8	11.17	T
8	6	9.13	T
8	2	8.79	T
8	0	8.19	T
8	7	10.25	T
8	8	11.30	T
8	8	10.84	T
8	8	10.97	T
8	3	8.60	T
8	7	9.91	T
8	8	11.38	T
8	8	10.39	T
8	7	10.45	T
8	0	8.94	T
8	5	8.94	T
8	8	10.14	T
8	1	8.50	T

In [10]:

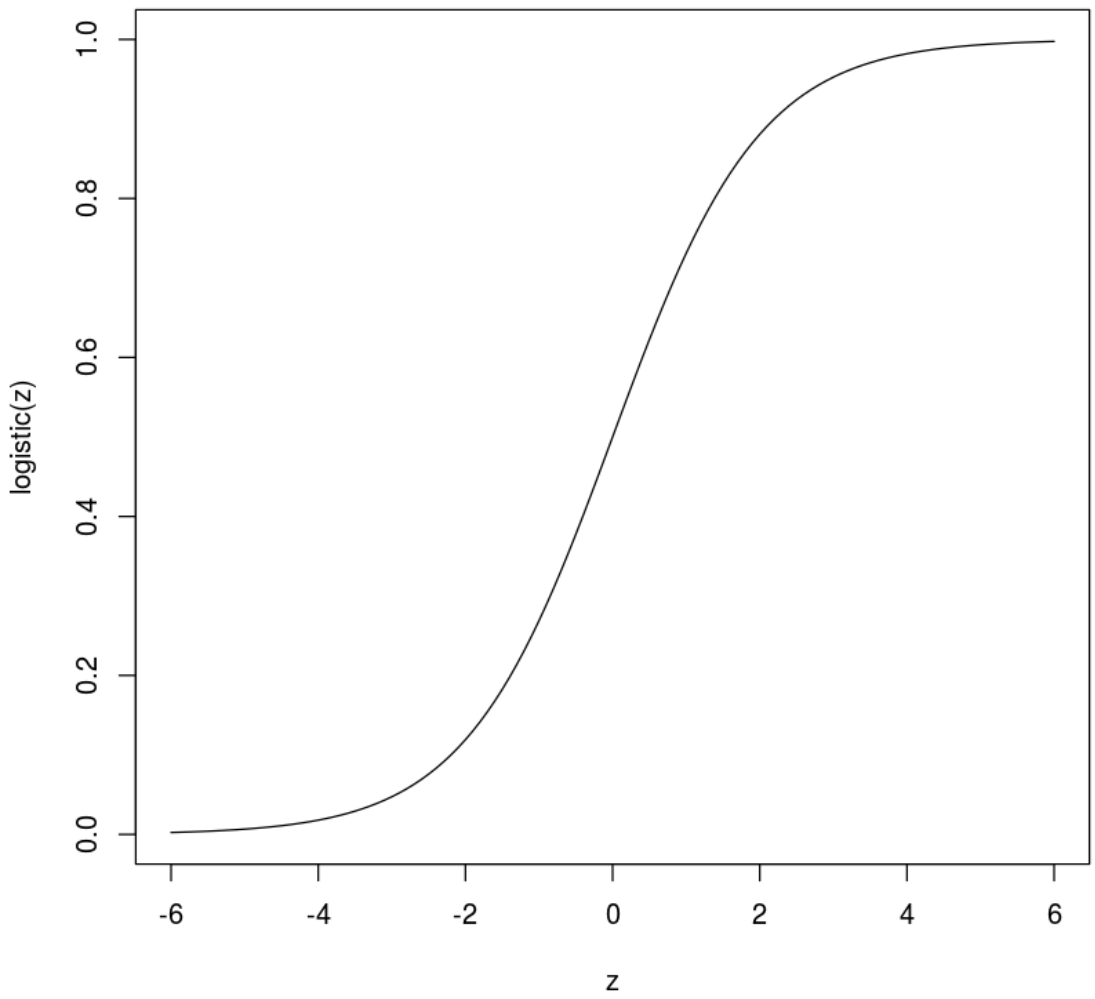
```
plot(d$x, d$y, pch = c(21, 19)[d$f])
legend("topleft", legend=c("C", "T"), pch=c(21, 19))
```



ロジスティック関数の概観

In [11]:

```
logistic <- function(z) 1 / (1 + exp(-z))
z <- seq(-6, 6, 0.1)
plot(z, logistic(z), type="l")
```



ロジスティック関数の逆関数がロジット関数。$logit(q_i) = \log \frac{q_i}{1-q_i}$

In [12]:

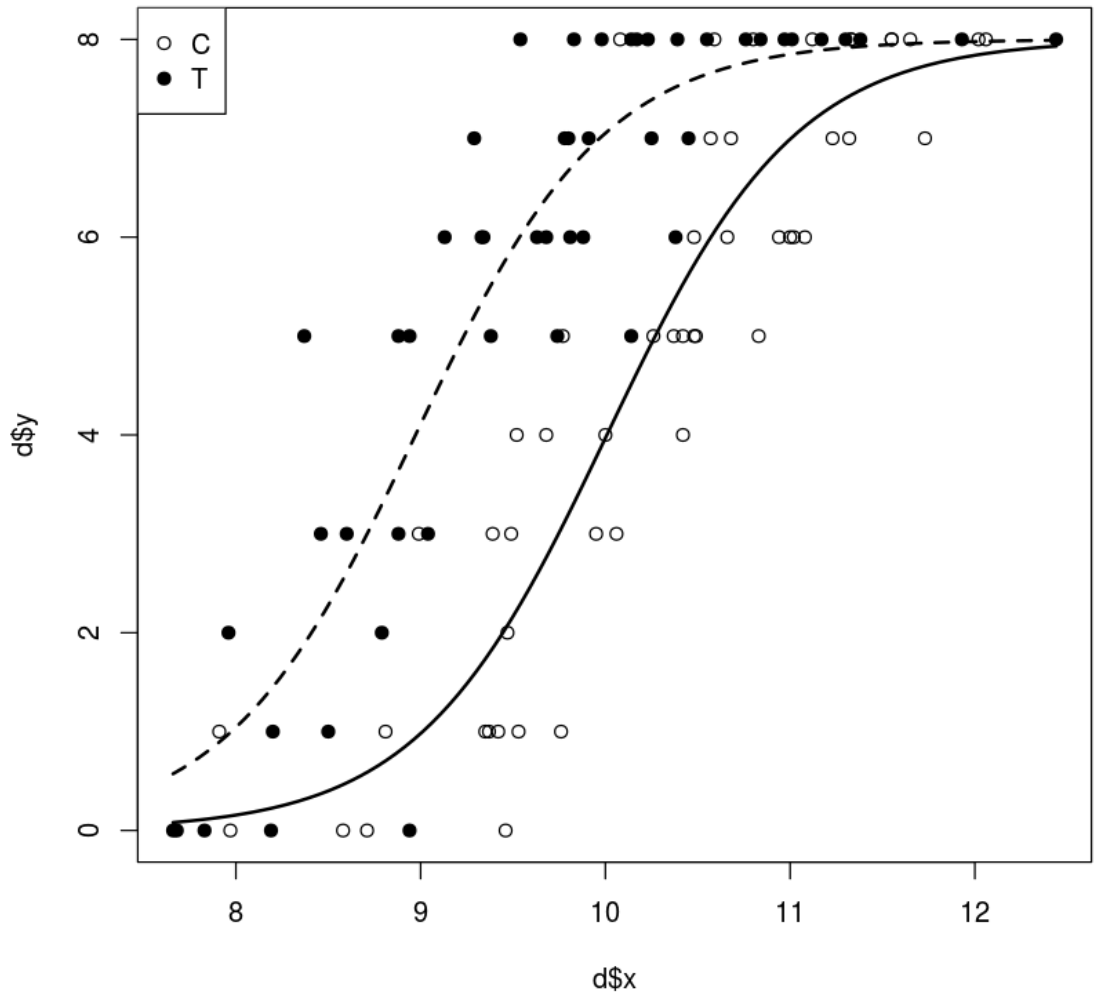
```
fit <- glm(cbind(y, N-y) ~ x + f, data = d, family = binomial)
fit
```

```
Call:  glm(formula = cbind(y, N - y) ~ x + f, family = binomial, data = d)

Coefficients:
(Intercept)          x         fT
   -19.536       1.952        2.022

Degrees of Freedom: 99 Total (i.e. Null);  97 Residual
Null Deviance:          499.2
Residual Deviance: 123        AIC: 272.2
```

応答変数が cbind(y, N-y)、family が binomial である点に注意。

In [13]:

```
plot(d$x, d$y, pch = c(21, 19)[d$f])
xx <- seq(min(d$x), max(d$x), length=100)
lines(xx, logistic(-19.536 + 1.952 * xx) * d$N, lwd = 2)
lines(xx, logistic(-19.536 + 1.952 * xx + 2.022) * d$N, lwd = 2, lty=2)
legend("topleft", legend=c("C", "T"), pch=c(21, 19))
```



AIC によるモデル選択

In [14]:

```
library(MASS)
stepAIC(fit)
```

```
Start:  AIC=272.21
cbind(y, N - y) ~ x + f

      Df Deviance    AIC
<none>      123.03 272.21
- f     1   217.17 364.35
- x     1   490.58 637.76

Call:  glm(formula = cbind(y, N - y) ~ x + f, family = binomial, data = d)

Coefficients:
(Intercept)          x         fT
   -19.536       1.952       2.022

Degrees of Freedom: 99 Total (i.e. Null);  97 Residual
Null Deviance:          499.2
Residual Deviance: 123      AIC: 272.2
```

交互作用を考える

In [15]:

```
fit <- glm(cbind(y, N-y) ~ x + f + x : f, data = d, family = binomial)
fit
```

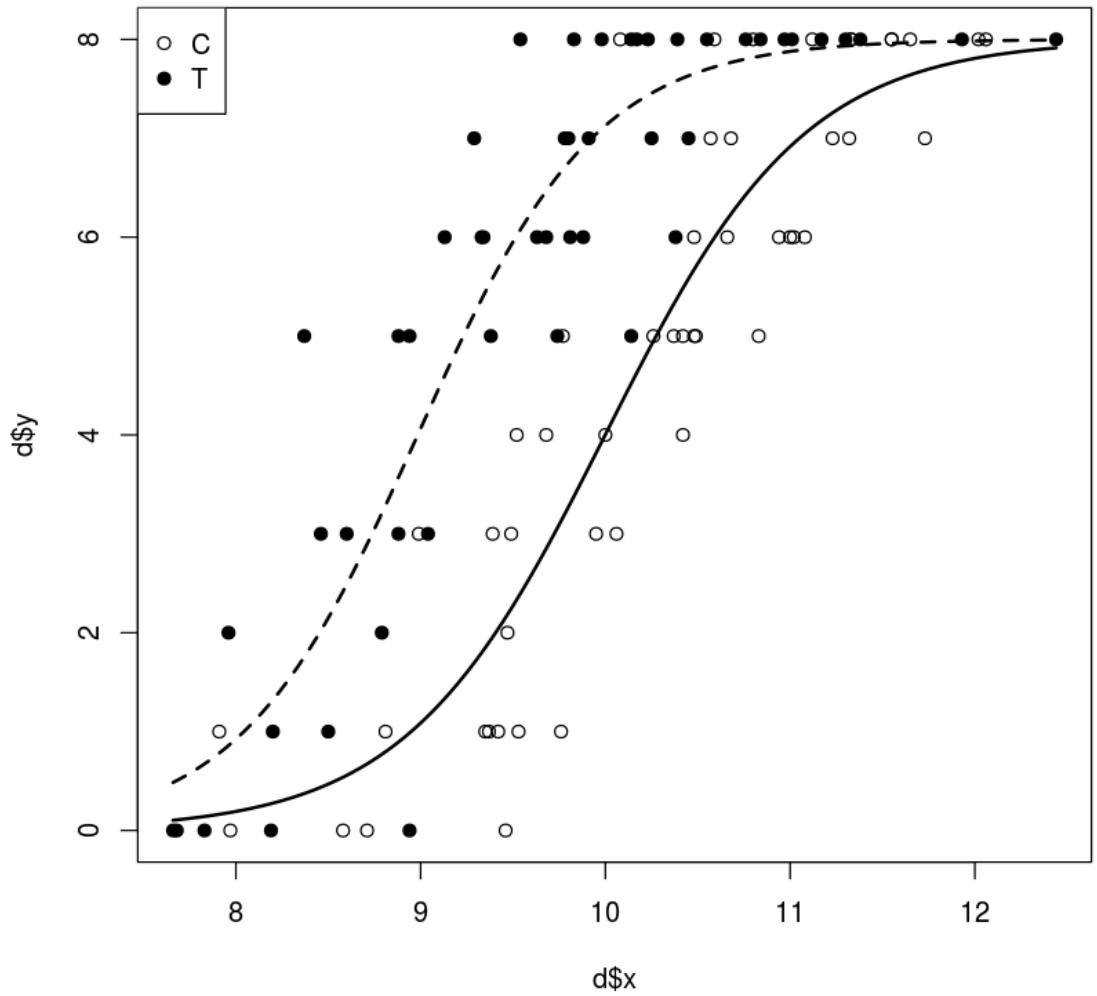
```
Call:  glm(formula = cbind(y, N - y) ~ x + f + x:f, family = binomial,
    data = d)

Coefficients:
(Intercept)          x         fT        x:fT
  -18.52332     1.85251    -0.06376     0.21634

Degrees of Freedom: 99 Total (i.e. Null);  96 Residual
Null Deviance:          499.2
Residual Deviance: 122.4    AIC: 273.6
```

交互作用によって、fT が大きく変わったように見えるが、実際に式を展開すると、そんなに変わらない。

交互作用も考慮した図を示す。

In [16]:

```
plot(d$x, d$y, pch = c(21, 19)[d$f])
xx <- seq(min(d$x), max(d$x), length=100)
lines(xx, logistic(-18.52332 + 1.85251 * xx) * d$N, lwd = 2)
lines(xx, logistic(-18.52332 + 1.85251 * xx -0.06376 + 0.21634 * xx) * d$N, lwd = 2, lty=2)
legend("topleft", legend=c("C", "T"), pch=c(21, 19))
```



オフセット項の導入

In [17]:

```
d <- read.csv('data4b.csv')
d
```

y	x	A
57	0.68	10.3
64	0.27	15.6
49	0.46	10.0
64	0.45	14.9
82	0.74	14.0
29	0.15	9.6
37	0.50	11.8
33	0.57	6.8
61	0.79	11.8
46	0.57	9.5
61	0.80	10.5
30	0.62	6.8
45	0.84	6.4
21	0.36	6.0
39	0.49	11.9
43	0.57	10.1
54	0.55	12.0
24	0.13	9.0
33	0.65	7.1
39	0.62	7.4
53	0.52	12.4
44	0.64	8.3
50	0.42	12.1
52	0.67	8.4
59	0.52	11.4
33	0.19	11.5
38	0.58	9.0
46	0.48	10.5
30	0.22	9.6
35	0.24	11.9
⋮	⋮	⋮
54	0.50	10.3
76	0.59	14.8
70	0.52	14.2

y	x	A
36	0.33	9.8
38	0.23	10.1
48	0.41	12.3
36	0.38	10.6
43	0.49	11.2
69	0.72	13.3
55	0.44	13.6
51	0.41	13.4
26	0.15	9.4
62	0.62	12.0
42	0.99	5.7
46	0.50	9.9
67	0.75	12.9
67	0.65	13.4
68	0.55	12.8
19	0.46	4.9
47	0.44	10.0
41	0.62	9.0
69	0.22	16.8
33	0.05	11.4
46	0.43	11.2
74	0.58	17.4
57	0.76	9.8
49	0.17	12.5
95	0.98	11.4
27	0.54	5.4
71	0.47	13.5

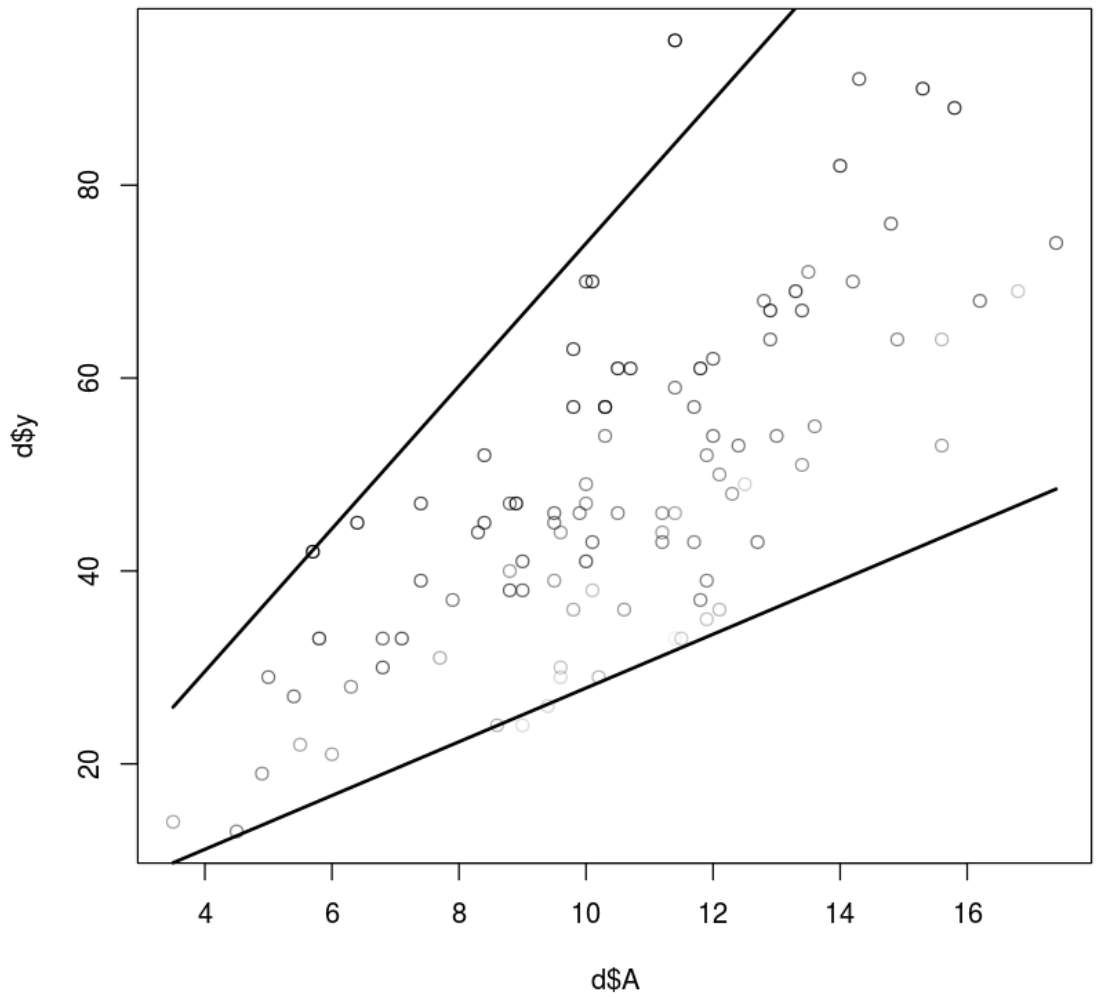
In [18]:

```
glm(y ~ x, offset = log(A), family=poisson, data=d)
```

```
Call:  glm(formula = y ~ x, family = poisson, data = d, offset = log(A))

Coefficients:
(Intercept)          x
     0.9731       1.0383

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:          261.5
Residual Deviance: 81.61     AIC: 650.3
```

In [19]:

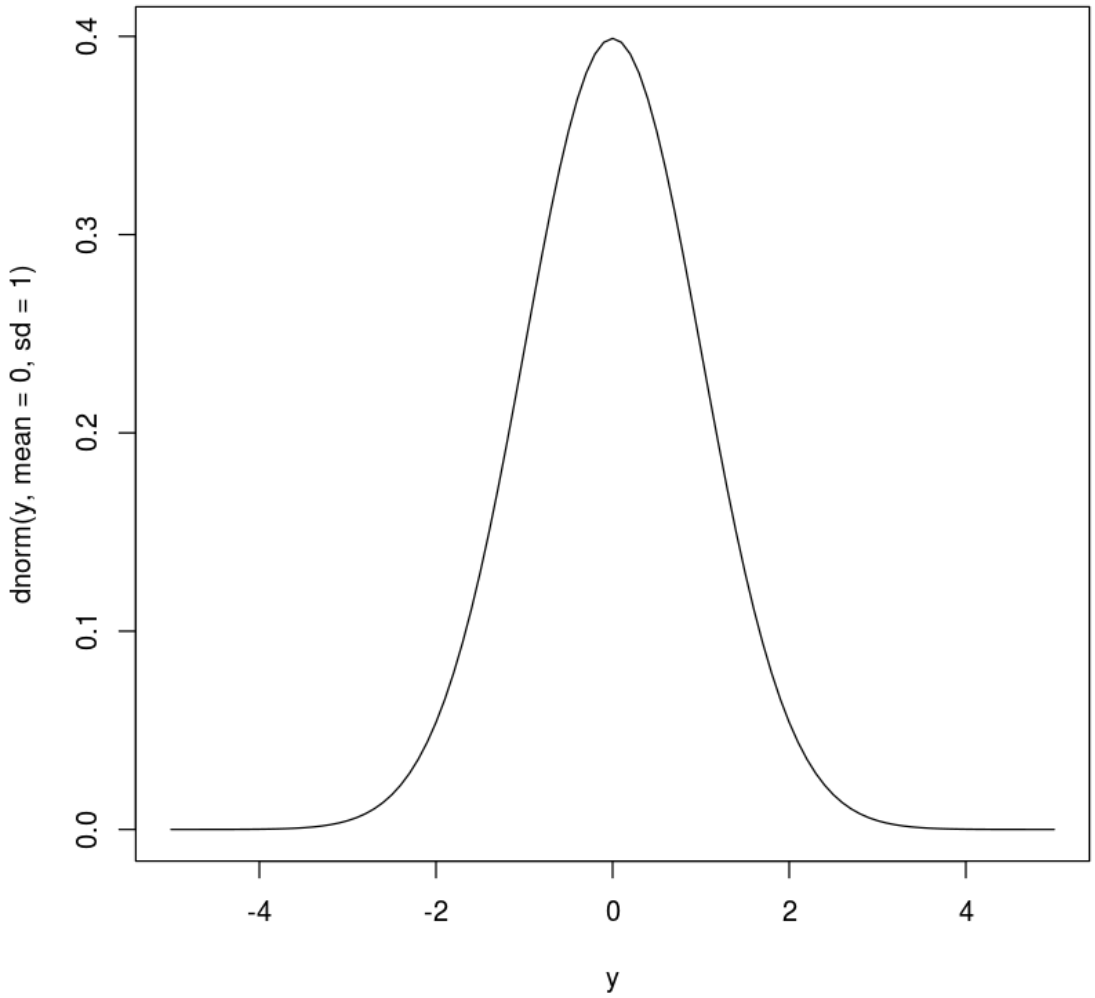
```
plot(d$A, d$y, col=rgb(0, 0, 0, alpha=d$x))
xx <- seq(min(d$A), max(d$A), length=100)
lines(xx, exp(0.9731 + 1.0383 * min(d$x) + log(xx)), lwd = 2)
lines(xx, exp(0.9731 + 1.0383 * max(d$x) + log(xx)), lwd = 2)
```



正規分布を使ってみる

In [20]:

```
y <- seq(-5, 5, 0.1)
plot(y, dnorm(y, mean = 0, sd = 1), type="l")
```



平均$\mu = 0$, 標準偏差$\sigma = 1$の正規分布で、
$1.2 \leq y \leq 1.8$ なる確率を計算したい場合は、下記のようにする。

In [21]:

```
pnorm(1.8, 0, 1) - pnorm(1.2, 0, 1)
```

0.0791393511087825

長方形で近似するという手もある。

In [22]:

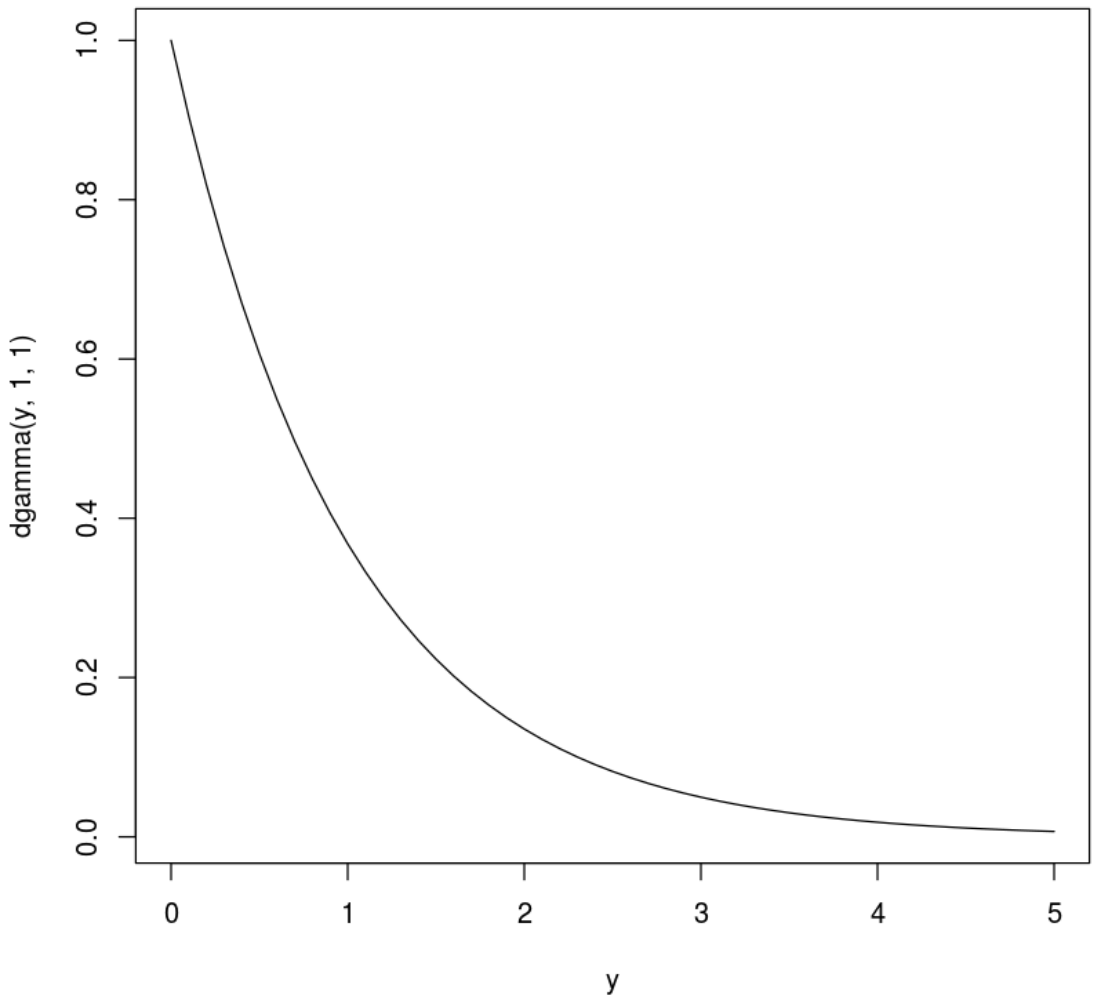
```
(1.8 - 1.2) * dnorm(1.5, mean=0, sd=1)
```

0.0777105573995351

ガンマ分布を使ってみる

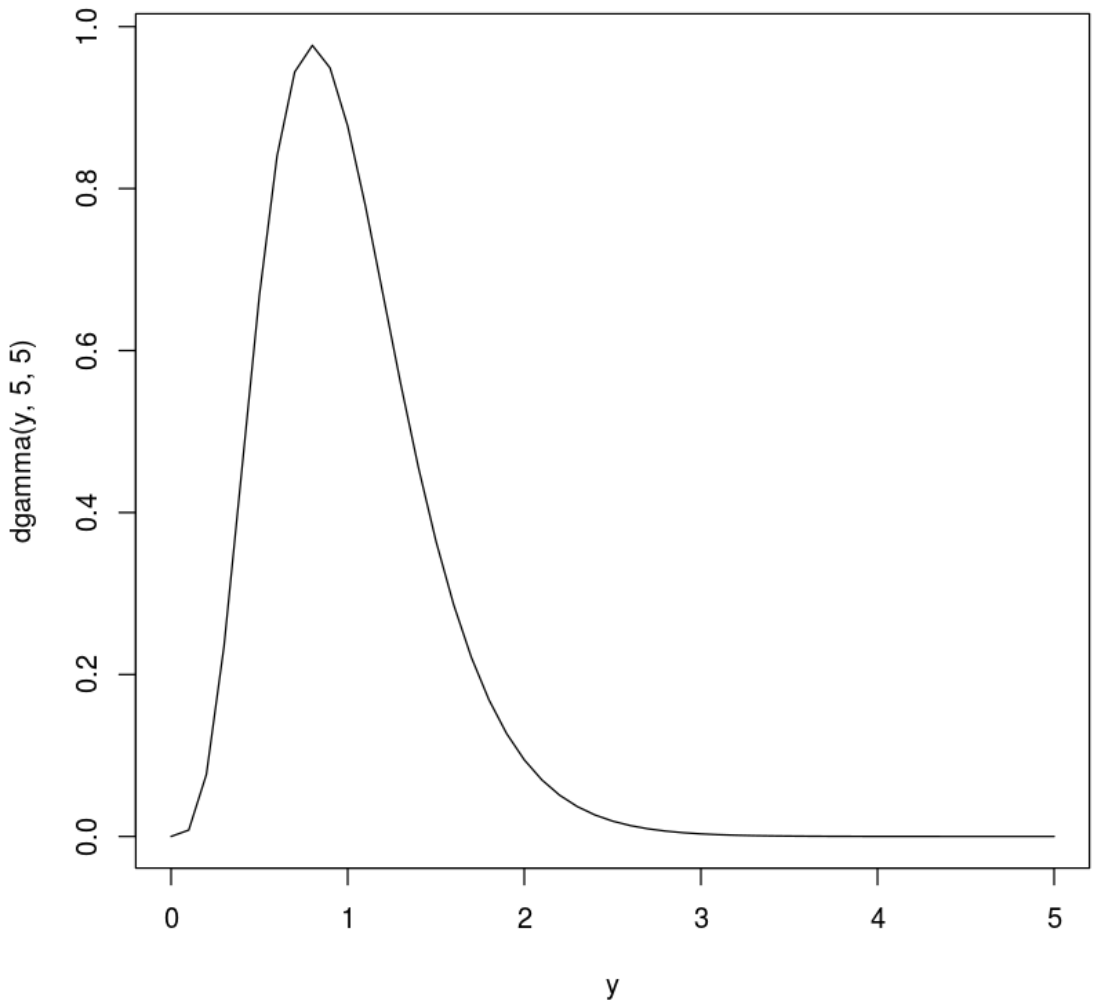
In [23]:

```
y <- seq(0, 5, 0.1)
plot(y, dgamma(y, 1, 1), type="l")
```



In [24]:

```
plot(y, dgamma(y, 5, 5), type="l")
```



In [25]:

```
d <- read.csv('data4c.csv')
d
```

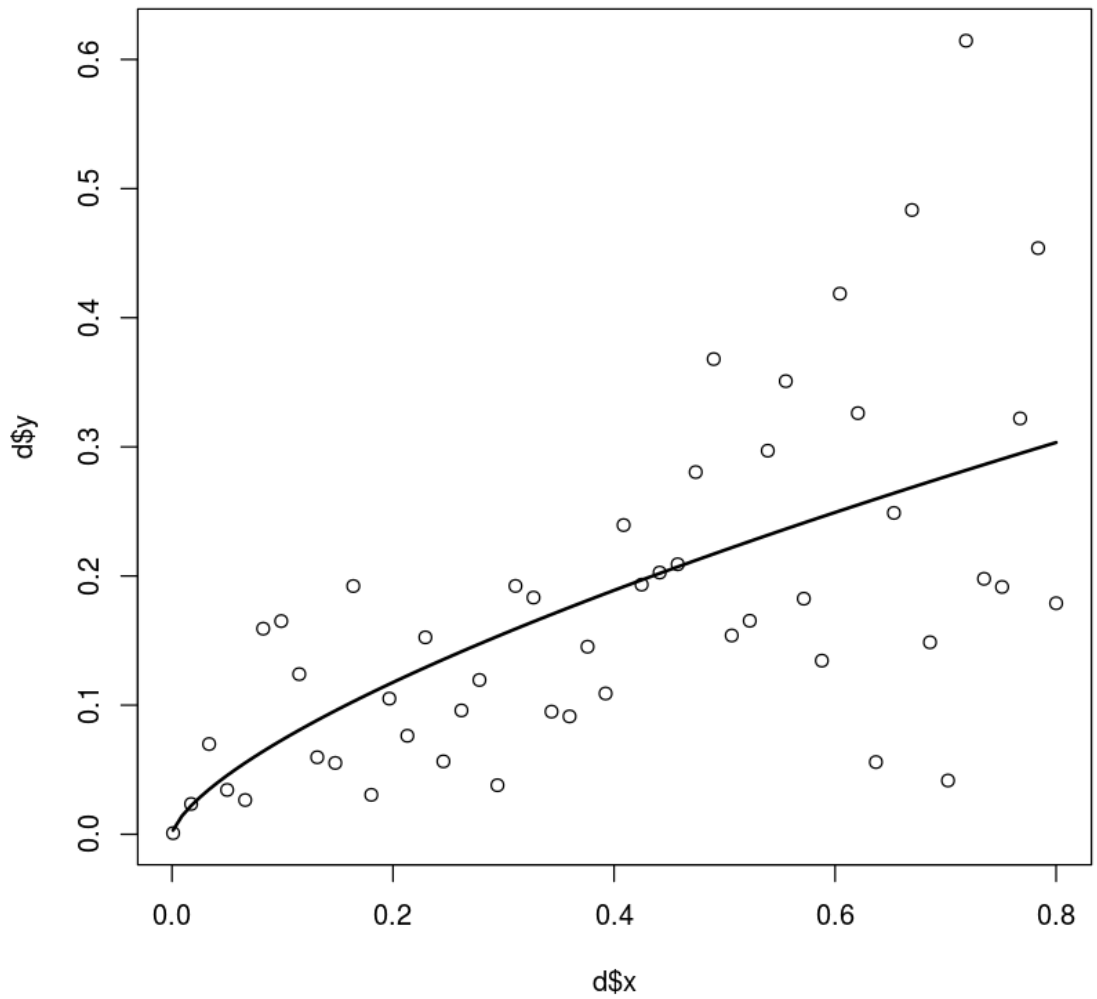
x	y
0.00100000	0.0008873584
0.01730612	0.0234652100
0.03361224	0.0698755600
0.04991837	0.0343402500
0.06622449	0.0265204000
0.08253061	0.1592148000
0.09883673	0.1650784000
0.11514290	0.1240029000
0.13144900	0.0596455000
0.14775510	0.0552452700
0.16406120	0.1922147000
0.18036740	0.0305346200
0.19667350	0.1050614000
0.21297960	0.0762759200
0.22928570	0.1524998000
0.24559180	0.0564525600
0.26189800	0.0959048800
0.27820410	0.1194827000
0.29451020	0.0379757900
0.31081630	0.1923055000
0.32712250	0.1833032000
0.34342860	0.0949801800
0.35973470	0.0912946500
0.37604080	0.1452413000
0.39234690	0.1090217000
0.40865310	0.2394446000
0.42495920	0.1933050000
0.44126530	0.2026793000
0.45757140	0.2090867000
0.47387750	0.2804644000
0.49018370	0.3679867000
0.50648980	0.1539353000
0.52279590	0.1653358000
0.53910200	0.2970396000

x	y
0.55540820	0.3508911000
0.57171430	0.1824436000
0.58802040	0.1344559000
0.60432650	0.4186217000
0.62063270	0.3261179000
0.63693880	0.0558967700
0.65324490	0.2488526000
0.66955100	0.4834365000
0.68585710	0.1486624000
0.70216330	0.0416329800
0.71846940	0.6145356000
0.73477550	0.1978336000
0.75108160	0.1915292000
0.76738780	0.3220099000
0.78369390	0.4539203000
0.80000000	0.1788959000

In [26]:

```
fit <- glm(y ~ log(x), family = Gamma(link = "log"), data = d)
fit
plot(d$x, d$y)
xx <- seq(min(d$x), max(d$x), length=100)
lines(xx, exp(-1.0403 + 0.6833 * log(xx)), lwd = 2)
```

```
Call:  glm(formula = y ~ log(x), family = Gamma(link = "log"), data = d)

Coefficients:
(Intercept)      log(x)
   -1.0403       0.6833

Degrees of Freedom: 49 Total (i.e. Null);  48 Residual
Null Deviance:          35.37
Residual Deviance: 17.25     AIC: -110.9
```



7章 一般化線形混合モデル(GLMM)

まずは、データを読み込む

In [27]:

```
d <- read.csv('data5.csv')
summary(d)
```

```
       N          y             x          id       
 Min.   :8   Min.   :0.00   Min.   :2   Min.   :  1.00  
 1st Qu.:8   1st Qu.:1.00   1st Qu.:3   1st Qu.: 25.75  
 Median :8   Median :3.00   Median :4   Median : 50.50  
 Mean   :8   Mean   :3.81   Mean   :4   Mean   : 50.50  
 3rd Qu.:8   3rd Qu.:7.00   3rd Qu.:5   3rd Qu.: 75.25  
 Max.   :8   Max.   :8.00   Max.   :6   Max.   :100.00  
```

生存種子数 y は二項分布に従うとして、fitting を行う。

In [28]:

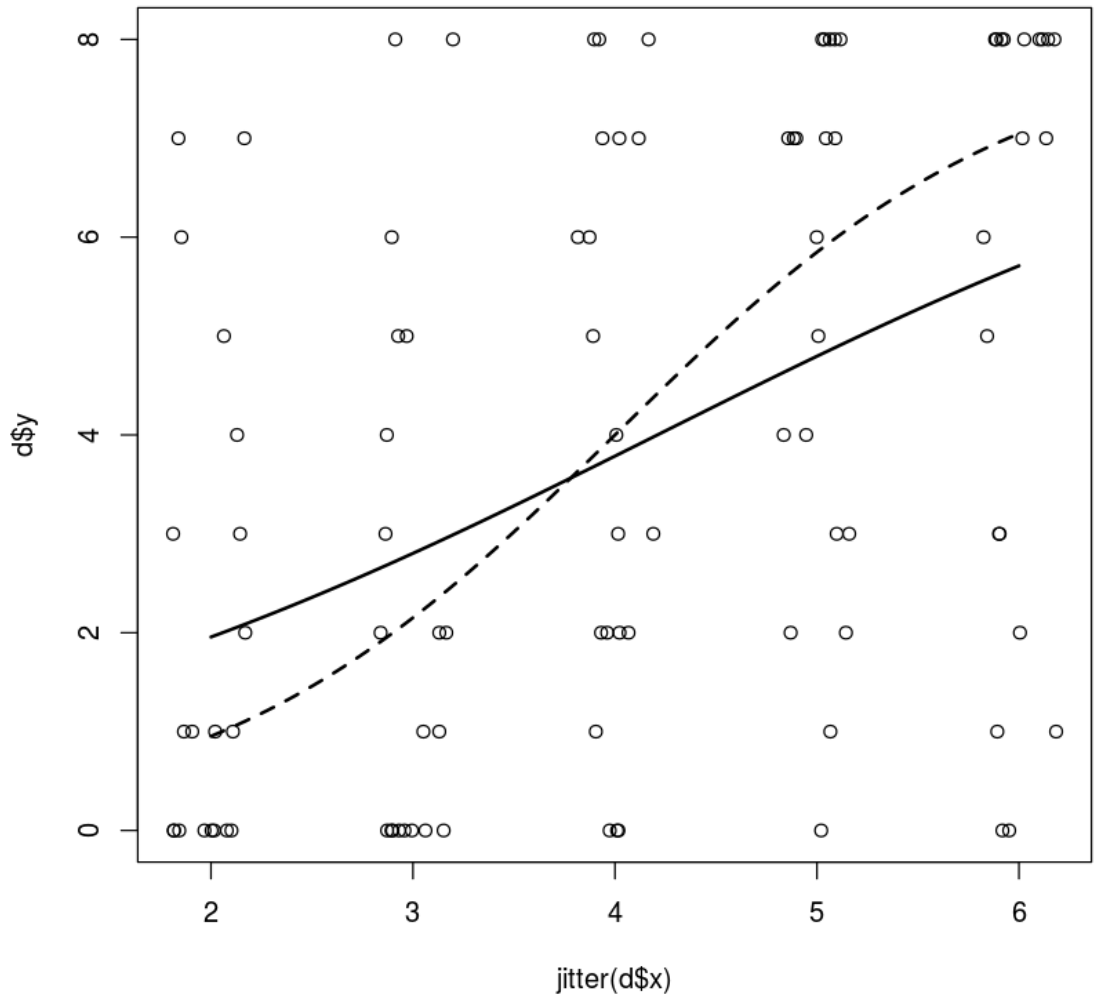
```
fit <- glm(cbind(y, N-y) ~ x, data=d, family=binomial)
fit
```

```
Call:  glm(formula = cbind(y, N - y) ~ x, family = binomial, data = d)

Coefficients:
(Intercept)            x  
    -2.1487       0.5104  

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:          607.4 
Residual Deviance: 513.8    AIC: 649.6
```

fitting は行われたが、真の傾き$\hat{\beta}_2 = 1$と比べると推定値$\beta_2 = 0.51$は小さい。

In [29]:

```
logistic <- function(z) 1 / (1 + exp(-z))
plot(jitter(d$x), d$y)
xx <- seq(min(d$x), max(d$x), length=100)
lines(xx, logistic(-2.1487 + 0.5104 * xx)*d$N, lwd=2)
lines(xx, logistic(-4 + 1 * xx)*d$N, lwd=2, lty=2)
```

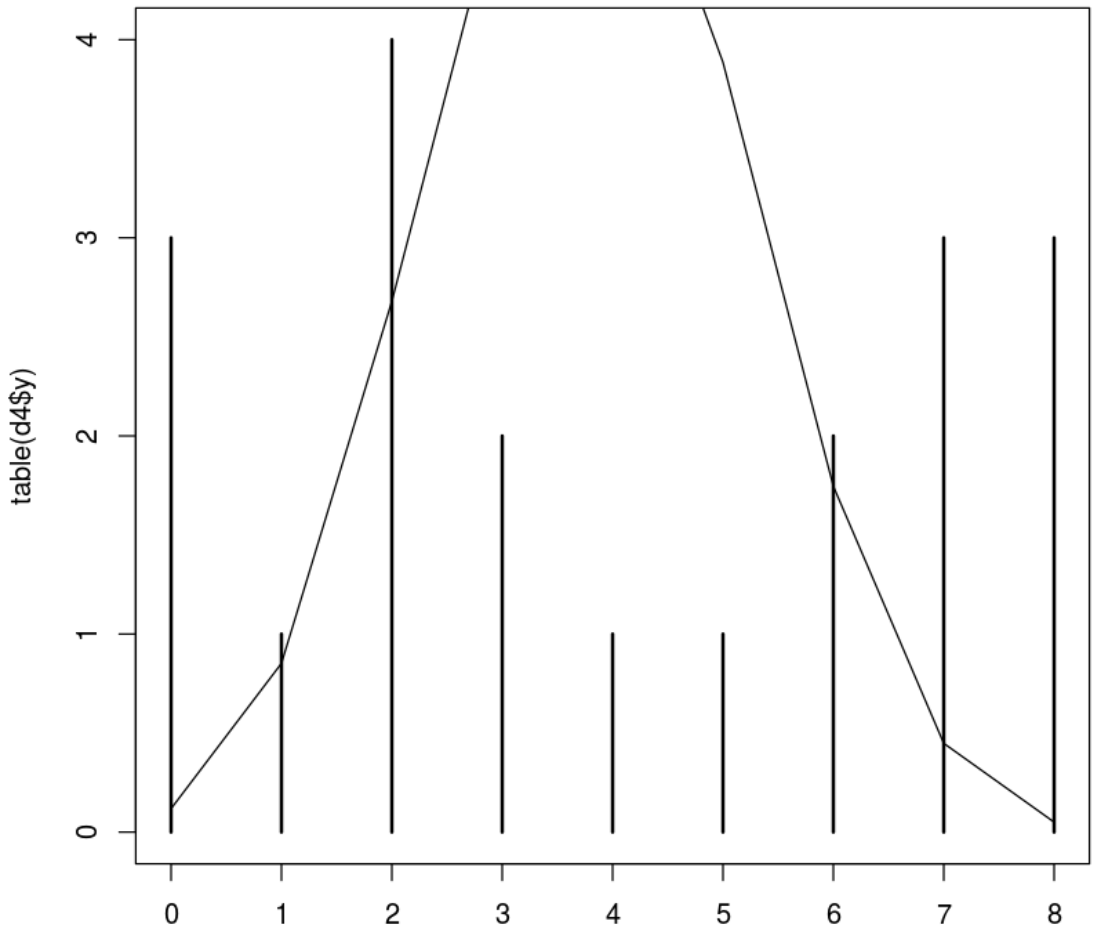


$x_i = 4$である個体に絞って見てみると、
本来は線で示した二項分布に乗るはずだが、実際は大きく異なる分布を見せている。

In [30]:

```
d4 <- d[d$x==4,]
summary(d4)
plot(table(d4$y))
y <- seq(0, 8, 1)
lines(y, dbinom(y, mean(d$N), logistic(-2.1487 + 0.5104 * 4)) * length(d4$y))
```

```
       N          y             x          id       
 Min.   :8   Min.   :0.00   Min.   :4   Min.   :41.00  
 1st Qu.:8   1st Qu.:2.00   1st Qu.:4   1st Qu.:45.75  
 Median :8   Median :3.50   Median :4   Median :50.50  
 Mean   :8   Mean   :4.05   Mean   :4   Mean   :50.50  
 3rd Qu.:8   3rd Qu.:7.00   3rd Qu.:4   3rd Qu.:55.25  
 Max.   :8   Max.   :8.00   Max.   :4   Max.   :60.00  
```



だいぶ二項分布とは異なる。

過分散と個体差

In [31]:

```
c(mean(d4$y), var(d4$y))
```

4.05 8.36578947368421

平均は 4.05/8=0.5 くらいで、分散は 8 0.5 (1-0.5) = 2 くらいになるはずなのに、
実際の分散は 8.4 と大きい。 個体が持つ要因によって予想よりも分散が大きくなってしまっている。
→実際の現象ではよく起こること。 そこで、個体差を表すパラメタ r_i を線形予測子に導入する。

In [32]:

```
install.packages("glmmML")
```

Warning message in install.packages("glmmML"):
“installation of package ‘glmmML’ had non-zero exit status”Updating HTML index of p
ackages in '.Library'
Making 'packages.html' ... done

In [33]:

```
library(glmmML)
glmmML(cbind(y, N-y) ~ x, data=d, family=binomial, cluster=id)
```

Error in library(glmmML): there is no package called ‘glmmML’
Traceback:

1. library(glmmML)
2. stop(txt, domain = NA)