# Chapter 16 Proportion data | Chapter 17 Binary response variables | Chapter 18 Generalized additive models

*Qianqian Shan*

*June 7, 2017*

## Chapater 16 Proportion Data

Count data on proportions.

### Analysis of data on one and two proportions

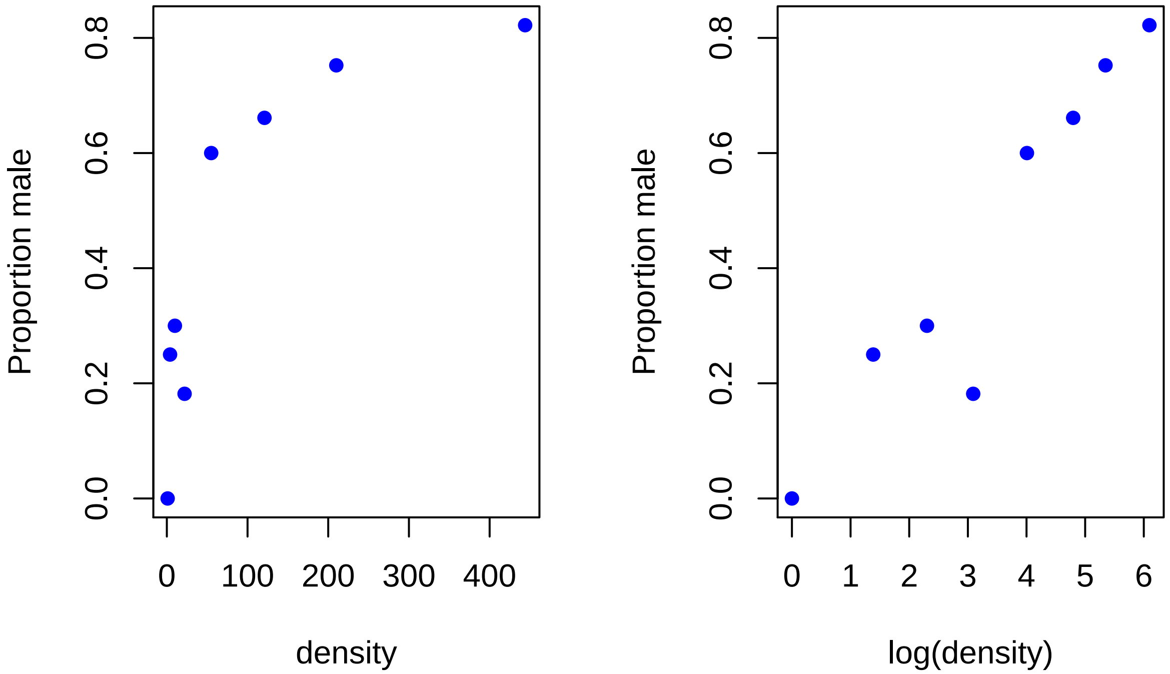
- Comparisons of one binomial proportion with a constant , use `binom.test`.
- Comparison of two samples of proportion data, use `prop.test`.

$ln(\frac{p}{1-p}) = a + bx$ wiht a linear predictor, logit transformation of $p$.

```r
# logistic regression with binomial errors
numbers <- read.table("sexratio.txt", header = TRUE)
attach(numbers)
head(numbers)
```

```
##   density females males
## 1       1       1     0
## 2       4       3     1
## 3      10       7     3
## 4      22      18     4
## 5      55      22    33
## 6     121      41    80
```

```r
# overview of data
par(mfrow=c(1,2))
# male ratio
p <- males/(males + females)
plot(density, p, ylab = "Proportion male", pch = 16, col = "blue")
# log(density)
plot(log(density), p, ylab = "Proportion male", pch = 16, col = "blue")
```



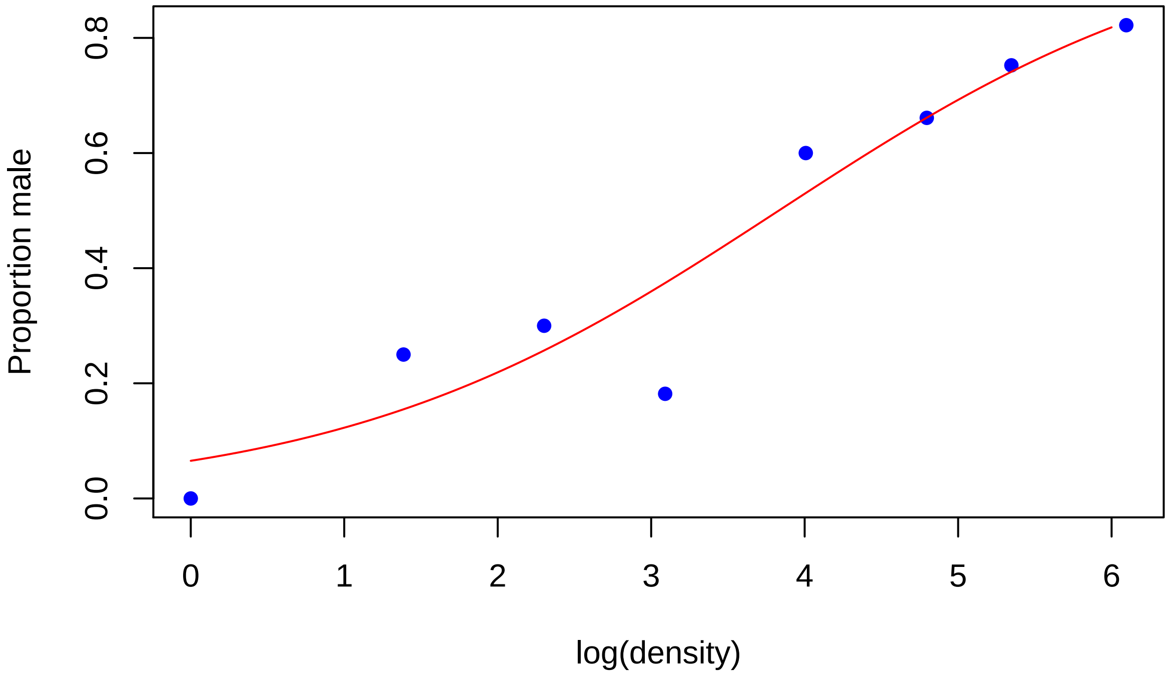
```r
par(mfrow= c(1, 2))

# glm with binomial errors
y <- cbind(males, females)

model <- glm(y ~ density, family = binomial)

summary(model)
```

```
##
## Call:
## glm(formula = y ~ density, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4619  -1.2760  -0.9911   0.5742   1.8795
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.0807368  0.1550376   0.521    0.603
## density     0.0035101  0.0005116   6.862 6.81e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 71.159  on 7  degrees of freedom
## Residual deviance: 22.091  on 6  degrees of freedom
## AIC: 54.618
##
## Number of Fisher Scoring iterations: 4
```

```r
# there is substantial overdipsersion as deviance is much bigger than the df


# fit log density
model2 <- glm(y ~ log(density), family = binomial)
summary(model2)
```

```
##
## Call:
## glm(formula = y ~ log(density), family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9697  -0.3411   0.1499   0.4019   1.0372
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.65927    0.48758  -5.454 4.92e-08 ***
## log(density)  0.69410    0.09056   7.665 1.80e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 71.1593  on 7  degrees of freedom
## Residual deviance:  5.6739  on 6  degrees of freedom
## AIC: 38.201
##
## Number of Fisher Scoring iterations: 4
```

```r
par(mfrow = c(1, 1))

# plot the fitted line
xv <- seq(0, 6, 0.01)
yv <- predict(model2, list(density = exp(xv)), type = "response") # type = response
plot(log(density), p, ylab = "Proportion male", pch = 16, col = "blue")
lines(xv, yv, col = "red")
```



```
detach(numbers)



# we want to know what kills the 50%(y, dead) ,
# i.e., use y to predict x and work out  a standard error on the x axis
data <- read.table("bioassay.txt", header = TRUE)
attach(data)
head(data)
```

```
##   dose dead batch
## 1    1    2   100
## 2    3   10    90
## 3   10   40    98
## 4   30   96   100
## 5  100   98   100
```

```
y <- cbind(dead, batch - dead)
model <- glm(y ~ log(dose), family = binomial)

library(MASS)
dose.p(model, p = c(0.5, 0.9, 0.95))
```

```
##                 Dose         SE
## p = 0.50: 2.306981 0.07772065
## p = 0.90: 3.425506 0.12362080
## p = 0.95: 3.805885 0.15150043
```

```
# dose.p(obj, cf = 1:2, p = 0.5)
# dose.p calibrates binomial assays, generalizing the calculation of LD50.
detach(data)



# proportion data with categorical explanatory variabels
```

```
germination <- read.table("germination.txt", header = TRUE)
attach(germination)
names(germination)
```

```
## [1] "count"     "sample"    "Orobanche" "extract"
```

```
y <- cbind(count, sample - count)

levels(Orobanche)
```

```
## [1] "a73" "a75"
```

```
levels(extract)
```

```
## [1] "bean"     "cucumber"
```

```
# factorial analysis
model <- glm(y ~ Orobanche * extract, binomial)
summary(model)
```

```
## 
## Call:
## glm(formula = y ~ Orobanche * extract, family = binomial)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.01617  -1.24398   0.05995   0.84695   2.12123  
## 
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)                   -0.4122     0.1842  -2.238   0.0252 *
## Orobanchea75                  -0.1459     0.2232  -0.654   0.5132  
## extractcucumber                0.5401     0.2498   2.162   0.0306 *
## Orobanchea75:extractcucumber   0.7781     0.3064   2.539   0.0111 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 98.719  on 20  degrees of freedom
## Residual deviance: 33.278  on 17  degrees of freedom
## AIC: 117.87
## 
## Number of Fisher Scoring iterations: 4
```

```
# approximate dispersion parameter
sum(summary(model)$deviance.resid^2)/summary(model)$df.residual
```

```
## [1] 1.957517
```

```
# use quasi-binomial
model <- glm(y ~ Orobanche * extract, family = quasibinomial)
summary(model)
```

```
## 
## Call:
## glm(formula = y ~ Orobanche * extract, family = quasibinomial)
## 
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.01617  -1.24398   0.05995   0.84695   2.12123
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -0.4122     0.2513  -1.640   0.1193
## Orobanchea75                  -0.1459     0.3045  -0.479   0.6379
## extractcucumber                0.5401     0.3409   1.584   0.1315
## Orobanchea75:extractcucumber   0.7781     0.4181   1.861   0.0801 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.861832)
##
##     Null deviance: 98.719  on 20  degrees of freedom
## Residual deviance: 33.278  on 17  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
# update model
model2 <- update(model, ~ . - Orobanche:extract)

anova(model, model2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ Orobanche * extract
## Model 2: y ~ Orobanche + extract
##   Resid. Df Resid. Dev Df Deviance      F  Pr(>F)
## 1        17     33.278
## 2        18     39.686 -1  -6.4081 3.4418 0.08099 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model: quasibinomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                         20     98.719
## Orobanche  1    2.544        19     96.175  1.1954    0.2887
## extract    1   56.489        18     39.686 26.5412 6.692e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Orobanche factor seems not significant in model2
model3 <- update(model2, ~ . - Orobanche)
```

```
anova(model2, model3, test = "F") # minimal adequate
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ Orobanche + extract
## Model 2: y ~ extract
##   Resid. Df Resid. Dev Df Deviance      F Pr(>F)
## 1        18     39.686
## 2        19     42.751 -1   -3.065 1.4401 0.2457
```

```
coef(model3)
```

```
##     (Intercept) extractcucumber
##      -0.5121761       1.0574031
```

```
a <- coef(model3)[1]
b <- coef(model3)[2]

# the p for the first extract type
1/(1+1/(exp(a)))
```

```
## (Intercept)
##   0.3746835
```

```
# p for the second extract type
1/(1+1/(exp(a + b)))
```

```
## (Intercept)
##   0.6330275
```

```
# make prediction
tapply(predict(model3, type = "response"), extract, mean)
```

```
##      bean  cucumber
## 0.3746835 0.6330275
```

```
# the average of raw proportions
as.vector(tapply(count,extract,sum))/as.vector(tapply(sample,extract,sum))
```

```
## [1] 0.3746835 0.6330275
```

```
# The average of proportions is the total counts over the total samples,
# NOT averaging the raw proportions one by one

detach(germination)
```

### Analysis of covariance with binomial data

Data with both continuous and categorical explanatory variables.

```
props <- read.table("flowering.txt", header = TRUE)
attach(props)
names(props)
```

```
## [1] "flowered" "number"   "dose"     "variety"
```

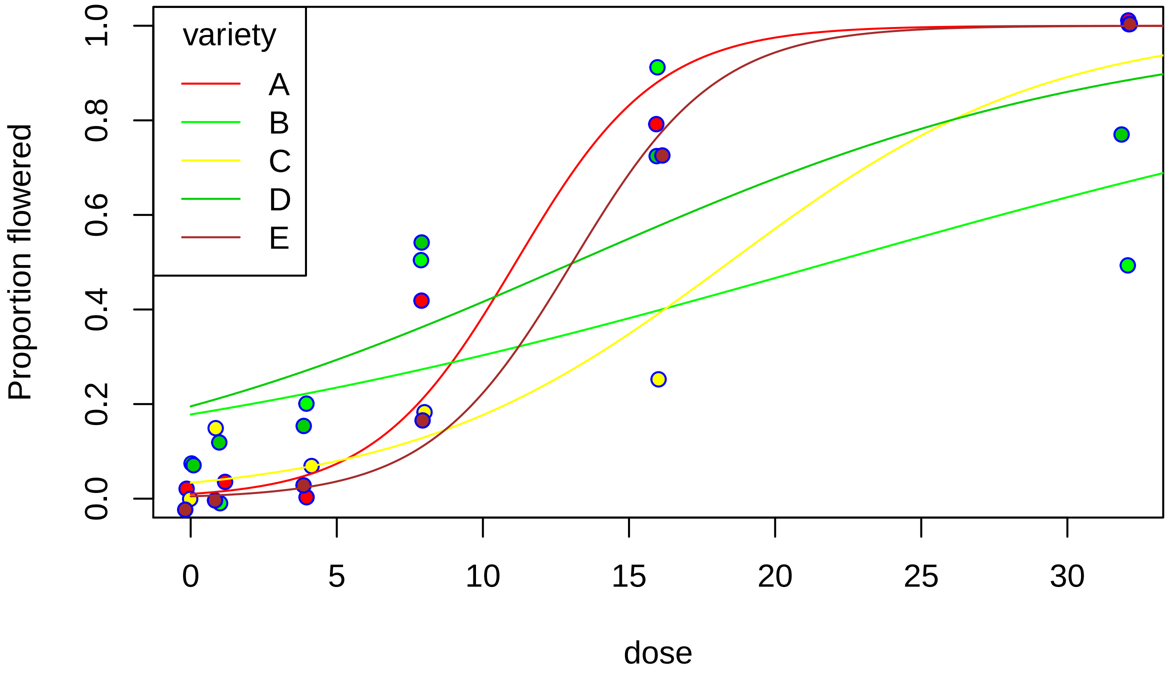
```
# dose continuous, variety categorical
y <- cbind(flowered, number - flowered)
```

```
pf <- flowered/number
pfc <- split(pf, variety)
dc <- split(dose, variety)
plot(dose, pf, type = "n", ylab = "Proportion flowered")
points(jitter(dc[[1]]), jitter(pfc[[1]]), pch = 21, col = "blue", bg = "red")
points(jitter(dc[[2]]), jitter(pfc[[2]]), pch = 21, col = "blue", bg = "green")

points(jitter(dc[[3]]), jitter(pfc[[3]]), pch = 21, col = "blue", bg = "yellow")
points(jitter(dc[[4]]), jitter(pfc[[4]]), pch = 21, col = "blue", bg = "green3")
points(jitter(dc[[5]]), jitter(pfc[[5]]), pch = 21, col = "blue", bg = "brown")


# fit maximal model
model1 <- glm(y ~ dose * variety, family = binomial)
summary(model1) # overdispersion
```

```
## 
## Call:
## glm(formula = y ~ dose * variety, family = binomial)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.6648  -1.1200  -0.3769   0.5735   3.3299  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   -4.59165    1.03215  -4.449 8.64e-06 ***
## dose           0.41262    0.10033   4.113 3.91e-05 ***
## varietyB       3.06197    1.09317   2.801 0.005094 ** 
## varietyC       1.23248    1.18812   1.037 0.299576    
## varietyD       3.17506    1.07516   2.953 0.003146 ** 
## varietyE      -0.71466    1.54849  -0.462 0.644426    
## dose:varietyB -0.34282    0.10239  -3.348 0.000813 ***
## dose:varietyC -0.23039    0.10698  -2.154 0.031274 *  
## dose:varietyD -0.30481    0.10257  -2.972 0.002961 ** 
## dose:varietyE -0.00649    0.13292  -0.049 0.961057    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 303.350  on 29  degrees of freedom
## Residual deviance:  51.083  on 20  degrees of freedom
## AIC: 123.55
## 
## Number of Fisher Scoring iterations: 5
```

```
# plot fitted curve
xv <- seq(0, 35, 0.1)
vn <- rep("A", length(xv))
yv <- predict(model1, list(variety = factor(vn), dose = xv), type = "response")
lines(xv, yv, col = "red")
vn <- rep("B", length(xv))
yv <- predict(model1, list(variety = factor(vn), dose = xv), type = "response")
```

```
lines(xv, yv, col = "green")
vn <- rep("C", length(xv))
yv <- predict(model1, list(variety = factor(vn), dose = xv), type = "response")
lines(xv, yv, col = "yellow")
vn <- rep("D", length(xv))
yv <- predict(model1, list(variety = factor(vn), dose = xv), type = "response")
lines(xv, yv, col = "green3")

vn <- rep("E", length(xv))
yv <- predict(model1, list(variety = factor(vn), dose = xv), type = "response")
lines(xv, yv, col = "brown")
legend("topleft", legend = c("A", "B", "C", "D", "E"), title = "variety",
            lty = rep(1, 5), col = c("red", "green", "yellow", "green3", "brown"))
```



```
tapply(pf, list(dose, variety), mean)
```

```
##            A          B          C          D         E
## 0  0.0000000 0.08333333 0.00000000 0.06666667 0.0000000
## 1  0.0000000 0.00000000 0.14285714 0.11111111 0.0000000
## 4  0.0000000 0.20000000 0.06666667 0.15789474 0.0000000
## 8  0.4000000 0.50000000 0.17647059 0.53571429 0.1578947
## 16 0.8181818 0.90000000 0.25000000 0.73076923 0.7500000
## 32 1.0000000 0.50000000 1.00000000 0.77777778 1.0000000
```

```
detach(props)
```

**Sumamry**: we have proportion data doesn't necessarily mean that the data will be well described by the logistic model.

### Converting complex contingency tables to proportions

Remove the need for all of the nuisance variables that are involved in complex contingency table modeling.

```r
lizards <- read.table("lizards.txt", header = TRUE)
attach(lizards)
head(lizards)
```

```
##    n   sun height  perch    time  species
## 1 20 Shade   High  Broad Morning opalinus
## 2 13 Shade    Low  Broad Morning opalinus
## 3  8 Shade   High Narrow Morning opalinus
## 4  6 Shade    Low Narrow Morning opalinus
## 5 34   Sun   High  Broad Morning opalinus
## 6 31   Sun    Low  Broad Morning opalinus
```

```r
sorted <- lizards[order(species, sun, height, perch, time), ]
levels(species) # two levels
```

```
## [1] "grahamii" "opalinus"
```

```r
head(sorted)
```

```
##    n   sun height  perch      time  species
## 41 4 Shade   High  Broad Afternoon grahamii
## 33 1 Shade   High  Broad   Mid.day grahamii
## 25 2 Shade   High  Broad   Morning grahamii
## 43 3 Shade   High Narrow Afternoon grahamii
## 35 1 Shade   High Narrow   Mid.day grahamii
## 27 3 Shade   High Narrow   Morning grahamii
```

```r
dim(sorted) # 1-24 one species, 25-48 another species
```

```
## [1] 48  6
```

```r
short <- sorted[1:24, ]

names(short)[1] <- "Ag" # the original "n" column
names(short)
```

```
## [1] "Ag"      "sun"     "height"  "perch"   "time"    "species"
```

```r
head(short)
```

```
##    Ag   sun height  perch      time  species
## 41  4 Shade   High  Broad Afternoon grahamii
## 33  1 Shade   High  Broad   Mid.day grahamii
## 25  2 Shade   High  Broad   Morning grahamii
## 43  3 Shade   High Narrow Afternoon grahamii
## 35  1 Shade   High Narrow   Mid.day grahamii
## 27  3 Shade   High Narrow   Morning grahamii
```

```r
# delete the last column, i.e., the species
short <- short[, -6]
head(short)
```

```
##    Ag   sun height  perch      time
## 41  4 Shade   High  Broad Afternoon
## 33  1 Shade   High  Broad   Mid.day
## 25  2 Shade   High  Broad   Morning
## 43  3 Shade   High Narrow Afternoon
## 35  1 Shade   High Narrow   Mid.day
## 27  3 Shade   High Narrow   Morning
```

```r
new.lizards <- data.frame(sorted$n[25:48], short)

names(new.lizards)[1] <- "Ao"
head(new.lizards)
```

```
##    Ao Ag   sun height  perch      time
## 41  4  4 Shade   High  Broad Afternoon
## 33  8  1 Shade   High  Broad   Mid.day
## 25 20  2 Shade   High  Broad   Morning
## 43  5  3 Shade   High Narrow Afternoon
## 35  4  1 Shade   High Narrow   Mid.day
## 27  8  3 Shade   High Narrow   Morning
```

```r
# create new columns Ao Ag to replace the original "n" column
# deleted the speices column

detach(lizards)
rm(short, sorted)
attach(new.lizards)

names(new.lizards)
```

```
## [1] "Ao"     "Ag"     "sun"    "height" "perch"  "time"
```

```r
y <- cbind(Ao, Ag)

model1 <- glm(y ~ sun * height * perch * time, family = binomial)

model2 <- step(model1)
```

```
## Start:  AIC=102.82
## y ~ sun * height * perch * time

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                         Df   Deviance    AIC
## - sun:height:perch:time  1 2.1800e-10 100.82
## <none>                     3.5826e-10 102.82

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=100.82
## y ~ sun + height + perch + time + sun:height + sun:perch + height:perch +
##     sun:time + height:time + perch:time + sun:height:perch +
##     sun:height:time + sun:perch:time + height:perch:time

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                     Df Deviance     AIC
## - sun:height:time    2   0.4416  97.266
## - sun:perch:time     2   0.8101  97.634
## - height:perch:time  2   3.2217 100.046
## <none>                   0.0000 100.824
## - sun:height:perch   1   2.7088 101.533
##
```

```
## Step:  AIC=97.27
## y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     sun:time + height:time + perch:time + sun:height:perch + 
##     sun:perch:time + height:perch:time
## 
##                      Df Deviance    AIC
## - sun:perch:time     2   1.0713 93.896
## <none>                   0.4416 97.266
## - height:perch:time  2   4.6476 97.472
## - sun:height:perch   1   3.1113 97.936
## 
## Step:  AIC=93.9
## y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     sun:time + height:time + perch:time + sun:height:perch + 
##     height:perch:time
## 
##                      Df Deviance    AIC
## - sun:time           2   3.3403 92.165
## <none>                   1.0713 93.896
## - sun:height:perch   1   3.3016 94.126
## - height:perch:time  2   5.7906 94.615
## 
## Step:  AIC=92.16
## y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     height:time + perch:time + sun:height:perch + height:perch:time
## 
##                      Df Deviance    AIC
## <none>                   3.3403 92.165
## - sun:height:perch   1   5.8273 92.651
## - height:perch:time  2   8.5418 93.366
```

```r
model3 <- update(model2,~. - height:perch:time)
model4 <- update(model2,~. - sun:height:perch)
anova(model2,model3,test="Chi")
```

```
## Analysis of Deviance Table
## 
## Model 1: y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     height:time + perch:time + sun:height:perch + height:perch:time
## Model 2: y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     height:time + perch:time + sun:height:perch
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1         7     3.3403                       
## 2         9     8.5418 -2  -5.2014  0.07422 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model2,model4,test="Chi")
```

```
## Analysis of Deviance Table
## 
## Model 1: y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     height:time + perch:time + sun:height:perch + height:perch:time
## Model 2: y ~ sun + height + perch + time + sun:height + sun:perch + height:perch + 
##     height:time + perch:time + height:perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         7     3.3403
## 2         8     5.8273 -1   -2.487   0.1148
```

```r
model5 <- glm(y~(sun+height+perch+time)^2-sun:time,binomial)

model6 <- update(model5,~. - sun:height)
anova(model5,model6,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ (sun + height + perch + time)^2 - sun:time
## Model 2: y ~ sun + height + perch + time + sun:perch + height:perch +
##     height:time + perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        10     10.903
## 2        11     13.254 -1  -2.3511   0.1252
```

```r
model7 <- update(model5,~. - sun:perch)
anova(model5,model7,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ (sun + height + perch + time)^2 - sun:time
## Model 2: y ~ sun + height + perch + time + sun:height + height:perch +
##     height:time + perch:time
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        10     10.903
## 2        11     10.927 -1 -0.023597   0.8779
```

```r
model8 <- update(model5,~. - height:perch)
anova(model5,model8,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ (sun + height + perch + time)^2 - sun:time
## Model 2: y ~ sun + height + perch + time + sun:height + sun:perch + height:time +
##     perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        10     10.903
## 2        11     11.143 -1 -0.24006   0.6242
```

```r
model9 <- update(model5,~. - time:perch)
anova(model5,model9,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ (sun + height + perch + time)^2 - sun:time
## Model 2: y ~ sun + height + perch + time + sun:height + sun:perch + height:perch +
##     height:time
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1        10     10.903
## 2        12     10.909 -2 -0.0058263   0.9971
```

```r
model10 <- update(model5,~. - time:height)
anova(model5,model10,test="Chi")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: y ~ (sun + height + perch + time)^2 - sun:time
## Model 2: y ~ sun + height + perch + time + sun:height + sun:perch + height:perch +
##     perch:time
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        10     10.903
## 2        12     11.760 -2 -0.85679   0.6516
```

```
model11 <- glm(y~sun+height+perch+time,binomial)
summary(model11)
```

```
##
## Call:
## glm(formula = y ~ sun + height + perch + time, family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.66015  -0.37800   0.04488   0.62644   1.48717
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.2079     0.3536   3.416 0.000634 ***
## sunSun       -0.8473     0.3224  -2.628 0.008585 **
## heightLow     1.1300     0.2571   4.395 1.11e-05 ***
## perchNarrow  -0.7626     0.2113  -3.610 0.000306 ***
## timeMid.day   0.9639     0.2816   3.423 0.000619 ***
## timeMorning   0.7368     0.2990   2.464 0.013730 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.102  on 22  degrees of freedom
## Residual deviance: 14.205  on 17  degrees of freedom
## AIC: 83.029
##
## Number of Fisher Scoring iterations: 4
```

```
# combine levels
t2 <- time
levels(t2)[c(2,3)] <- "other"
levels(t2)
```

```
## [1] "Afternoon" "other"
```

```
model12 <- glm(y~sun+height+perch+t2,binomial)
anova(model11,model12,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ sun + height + perch + time
## Model 2: y ~ sun + height + perch + t2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        17     14.205
## 2        18     15.023 -1 -0.81863   0.3656
```

```
summary(model12)
```

```
##
## Call:
## glm(formula = y ~ sun + height + perch + t2, family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.59707  -0.37407   0.06965   0.64616   1.53004
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.1595     0.3484   3.328 0.000874 ***
## sunSun       -0.7872     0.3159  -2.491 0.012722 *
## heightLow     1.1188     0.2566   4.360  1.3e-05 ***
## perchNarrow  -0.7485     0.2104  -3.557 0.000375 ***
## t2other       0.8717     0.2611   3.338 0.000844 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.102  on 22  degrees of freedom
## Residual deviance: 15.023  on 18  degrees of freedom
## AIC: 81.847
##
## Number of Fisher Scoring iterations: 4
```

```
detach(new.lizards)
rm(y)
```

# Chapter 17 Binary response variables

Steps:

1. Create a single vector containing 0s and 1s as response variables.
2. Use `glm` with `family = binomial`.
3. Consider changing the link function from default logit to complementary log-log.
4. Fit the model in the usual way.
5. Test significance by deletion of terms from the maximal model, and compare the change in deviance with chi-squared.

```
island <- read.table("isolation.txt", header = TRUE)
attach(island)
names(island)
```

```
## [1] "incidence" "area"      "isolation"
```

```
# incidence is 1 or 0

# maximal
model1 <- glm(incidence ~ area * isolation, family = binomial)
```

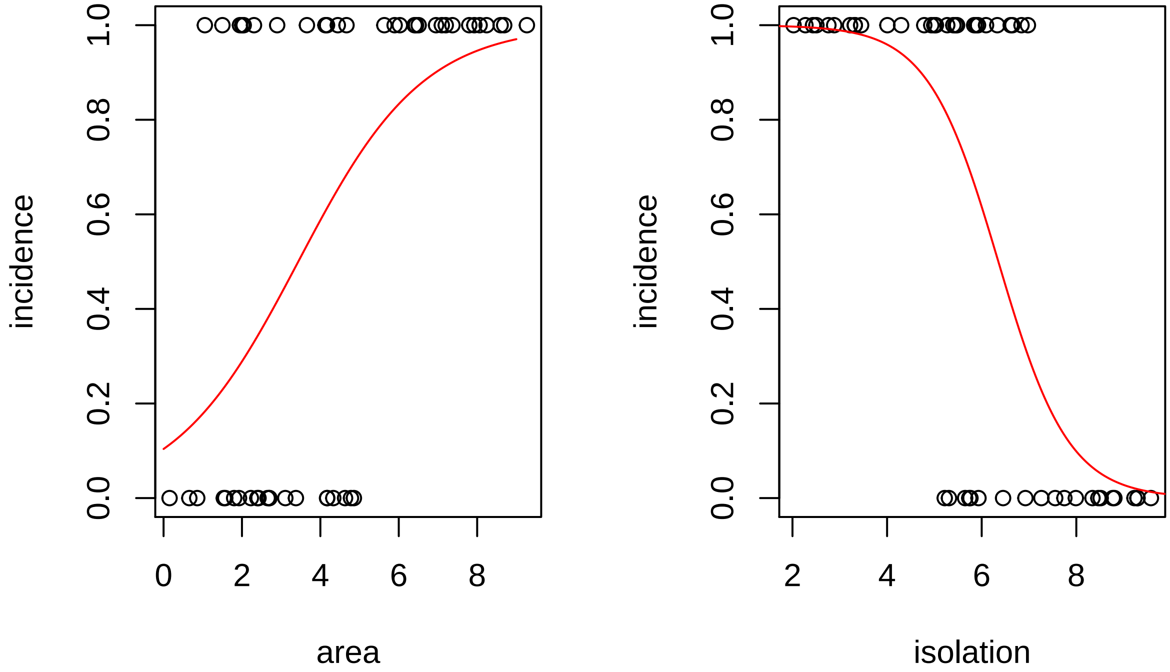
```r
# w/o interaction
model2 <- glm(incidence ~ area + isolation, family = binomial)

anova(model1, model2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: incidence ~ area * isolation
## Model 2: incidence ~ area + isolation
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        46     28.252
## 2        47     28.402 -1 -0.15043   0.6981
```

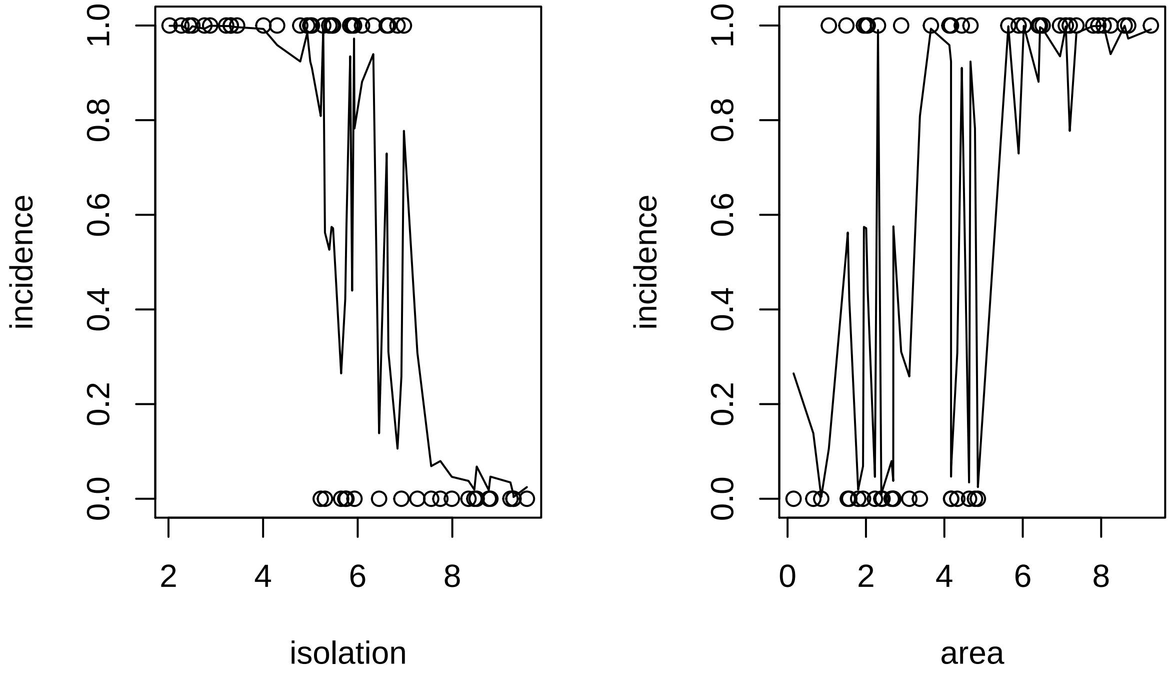
```r
summary(model2)
```

```
##
## Call:
## glm(formula = incidence ~ area + isolation, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8189  -0.3089   0.0490   0.3635   2.1192
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   6.6417     2.9218   2.273  0.02302 *
## area          0.5807     0.2478   2.344  0.01909 *
## isolation    -1.3719     0.4769  -2.877  0.00401 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 28.402  on 47  degrees of freedom
## AIC: 34.402
##
## Number of Fisher Scoring iterations: 6
```

```r
# plot fitted lines against each separately variable
modela <- glm(incidence ~ area, family = binomial)
modeli <- glm(incidence ~ isolation, family = binomial)

par(mfrow=c(1, 2))
xv <- seq(0, 9, 0.01)
yv <- predict(modela, list(area = xv), type = "response")
plot(area, incidence)
lines(xv, yv, col = "red")
xv2 <- seq(0, 10, 0.01)
yv2 <- predict(modeli, list(isolation = xv2), type = "response")
plot(isolation, incidence)
lines(xv2, yv2, col = "red")
```



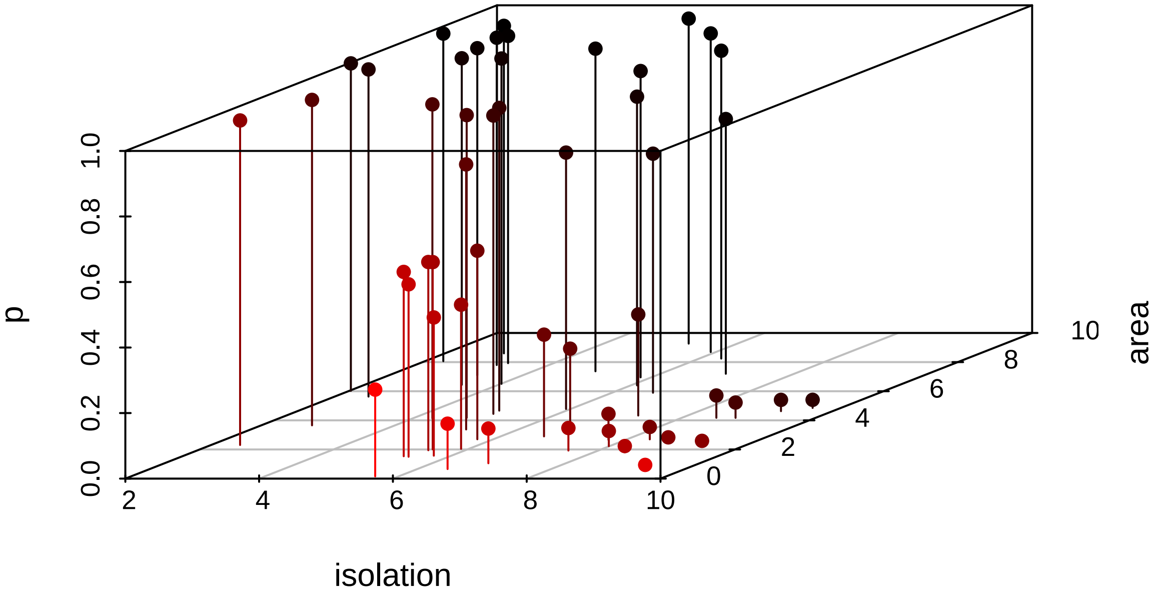
```r
plot(isolation, incidence)
lines(isolation[order(isolation)], predict(model2, type = "response")[order(isolation)])
plot(area, incidence)
lines(area[order(area)], predict(model2, type = "response")[order(area)])
```



```r
par(mfrow = c(1, 1))




library(scatterplot3d)
```

```
## Warning: package 'scatterplot3d' was built under R version 3.3.2
```

```
s3d <- scatterplot3d(x = isolation, y = area, z = predict(model2, type = "response"),
                     pch = 16, highlight.3d = TRUE, type = "h", zlab = "p")
```



```
detach(island)
```

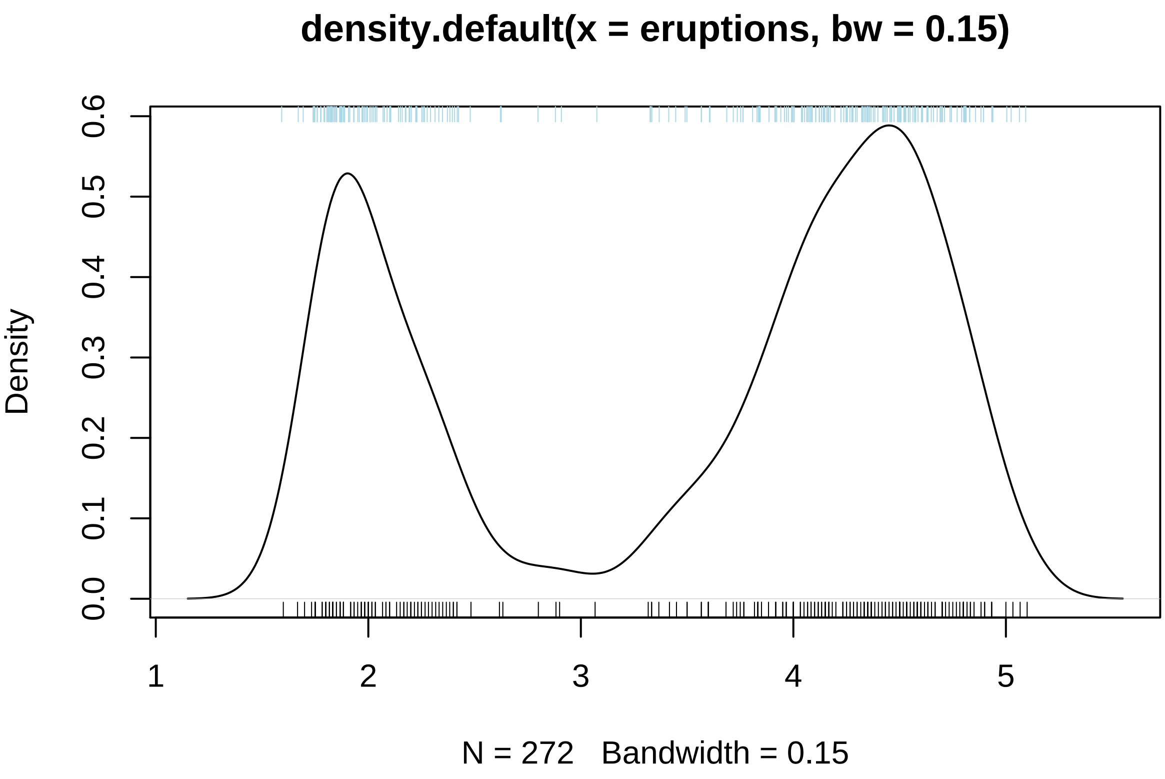
## Graphical tests of the fit of the logistic

**Rugs** are one-dimentional addition to the bottom(or top) of the plot showing the locations of the data points along x axis.

```
occupy <- read.table("occupation.txt", header = TRUE)
attach(occupy)
names(occupy)
```

```
## [1] "resources" "occupied"
```

```
# use of rug
with(faithful, {
    plot(density(eruptions, bw = 0.15))
    rug(eruptions)
    rug(jitter(eruptions, amount = 0.01), side = 3, col = "light blue")
})
```



```r
plot(resources, occupied, type = "n")
rug(jitter(resources[occupied == 0]))
rug(jitter(resources[occupied == 1]), side = 3)

model <- glm(occupied ~ resources, family = binomial)
xv <- 0:1000
yv <- predict(model, list(resources = xv), type = "response")
lines(xv, yv, col = "red")

# cut up the ranked values on x axis into five categories and
# then work out the mean and standard error of the proportions
# of each group
cutr <- cut(resources, 5)
head(cutr)
```

```
## [1] (13.2,209] (13.2,209] (13.2,209] (13.2,209] (13.2,209] (13.2,209]
## Levels: (13.2,209] (209,405] (405,600] (600,795] (795,992]
```

```r
tapply(occupied, cutr, sum) # number of observations in each group
```

```
## (13.2,209]  (209,405]  (405,600]  (600,795]  (795,992]
##          0         10         25         26         31
```

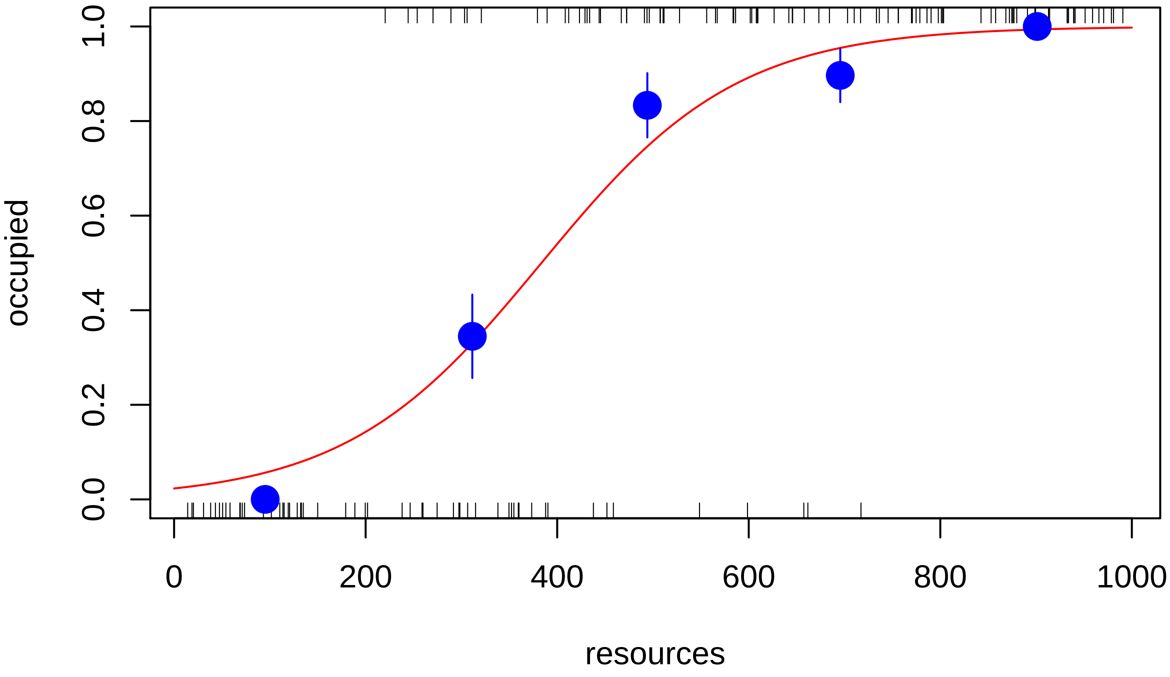
```r
table(cutr)
```

```
## cutr
## (13.2,209]  (209,405]  (405,600]  (600,795]  (795,992]
##         31         29         30         29         31
```

```r
# empirical probabilities
probs <- tapply(occupied, cutr, sum)/table(cutr)
probs
```

```
## (13.2,209]  (209,405]  (405,600]  (600,795]  (795,992]
##  0.0000000  0.3448276  0.8333333  0.8965517  1.0000000
```

```
probs <- as.vector(probs)

# mean values of each group as the x values of the empirical probabilities
resmeans <- tapply(resources, cutr, mean)
resmeans <- as.vector(resmeans)

points(resmeans, probs, pch = 16, cex = 2, col = "blue")

# standard error of each point by se = sqrt(prob * (1 - prob) / n)
se <- sqrt(probs * (1 - probs)/table(cutr))

up <- probs + as.vector(se)
down <- probs - as.vector(se)
for (i in 1:5) {
  lines(c(resmeans[i], resmeans[i]), c(up[i], down[i]), col = "blue")
}
```



```
detach(occupy)
```

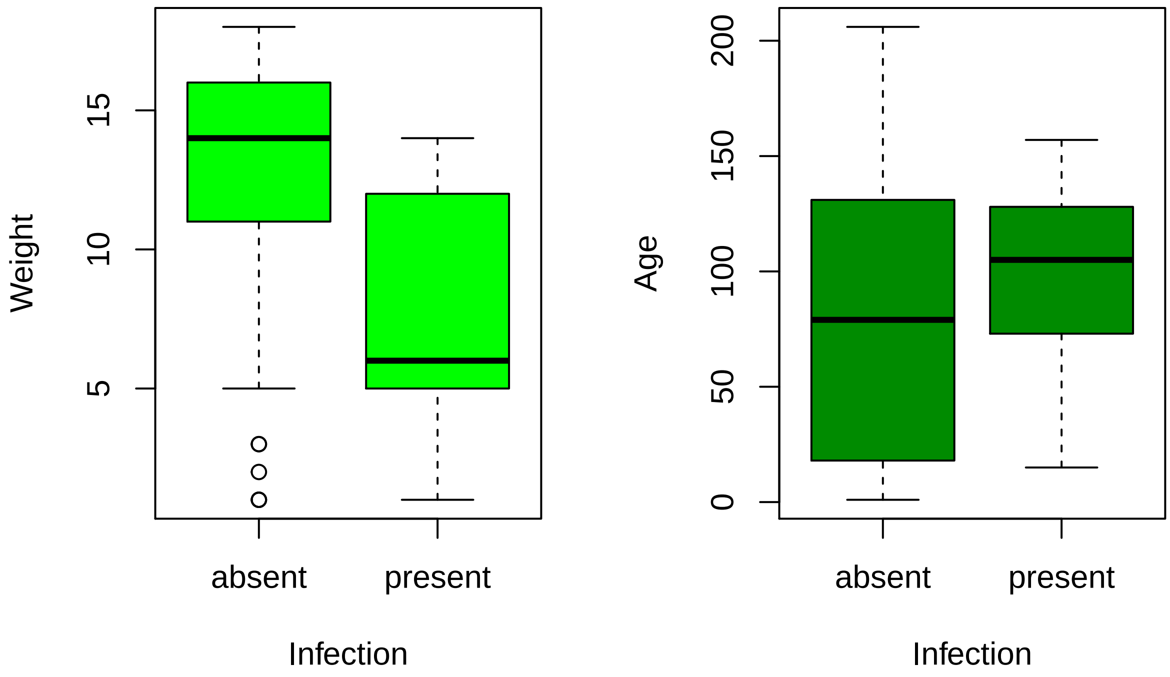
## ANCOVA with binary response variable

```
infection <- read.table("infection.txt", header = TRUE)
attach(infection)
names(infection)
```

```
## [1] "infected" "age"      "weight"   "sex"
```

```
# infected is binary response
# age , weight are continous
# sex categorical
```

```
par(mfrow=c(1,2))
plot(infected, weight, xlab = "Infection", ylab = "Weight", col = "green")
plot(infected, age, xlab = "Infection", ylab = "Age", col = "green4")
```



```
par(mfrow = c(1, 1))

# relationship with gender
table(infected, sex)
```

```
##           sex
## infected   female male
##   absent       17   47
##   present      11    6
```

```
# maximal model
model <- glm(infected ~ age * weight * sex, family = binomial)
summary(model)
```

```
## 
## Call:
## glm(formula = infected ~ age * weight * sex, family = binomial)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1767  -0.5359  -0.2494  -0.1691   2.3149  
## 
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.109124   1.375388  -0.079    0.937
## age                   0.024128   0.020874   1.156    0.248
## weight               -0.074156   0.147678  -0.502    0.616
## sexmale              -5.969109   4.278066  -1.395    0.163
## age:weight           -0.001977   0.002006  -0.985    0.325
## age:sexmale         0.038086   0.041325   0.922    0.357
## weight:sexmale      0.213830   0.343265   0.623    0.533
## age:weight:sexmale -0.001651   0.003419  -0.483    0.629
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 55.706  on 73  degrees of freedom
## AIC: 71.706
##
## Number of Fisher Scoring iterations: 6
```

```
# use step
model2 <- step(model)
```

```
## Start:  AIC=71.71
## infected ~ age * weight * sex
##
##                  Df Deviance    AIC
## - age:weight:sex  1   55.943 69.943
## <none>                55.706 71.706
##
## Step:  AIC=69.94
## infected ~ age + weight + sex + age:weight + age:sex + weight:sex
##
##              Df Deviance    AIC
## - weight:sex  1   56.122 68.122
## - age:sex     1   57.828 69.828
## <none>            55.943 69.943
## - age:weight  1   58.674 70.674
##
## Step:  AIC=68.12
## infected ~ age + weight + sex + age:weight + age:sex
##
##              Df Deviance    AIC
## <none>            56.122 68.122
## - age:sex     1   58.142 68.142
## - age:weight  1   58.899 68.899
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = infected ~ age + weight + sex + age:weight + age:sex,
##     family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1599  -0.5643  -0.2230  -0.1359   2.3490
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.391566   1.265230  -0.309   0.7570
## age          0.025764   0.014921   1.727   0.0842 .
## weight      -0.036494   0.128993  -0.283   0.7772
## sexmale     -3.743771   1.791962  -2.089   0.0367 *
## age:weight  -0.002221   0.001365  -1.627   0.1038
## age:sexmale  0.020464   0.015232   1.343   0.1791
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 56.122  on 75  degrees of freedom
## AIC: 68.122
##
## Number of Fisher Scoring iterations: 6
```

```
# interactions not significant, use update to simplify
model3 <- update(model2, ~.-age:weight)
anova(model2, model3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: infected ~ age + weight + sex + age:weight + age:sex
## Model 2: infected ~ age + weight + sex + age:sex
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        75     56.122
## 2        76     58.899 -1   -2.777  0.09562 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#
model4 <- update(model2, ~.-age:sex)
anova(model2, model4, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: infected ~ age + weight + sex + age:weight + age:sex
## Model 2: infected ~ age + weight + sex + age:weight
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        75     56.122
## 2        76     58.142 -1  -2.0203   0.1552
```

```
# test the main effects
model5 <- glm(infected ~ age + weight + sex, family = binomial)
summary(model5)
```

```
##
## Call:
## glm(formula = infected ~ age + weight + sex, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9481  -0.5284  -0.3120  -0.1437   2.2525
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.609369   0.803288   0.759 0.448096
## age          0.012653   0.006772   1.868 0.061701 .
## weight      -0.227912   0.068599  -3.322 0.000893 ***
## sexmale     -1.543444   0.685681  -2.251 0.024388 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 59.859  on 77  degrees of freedom
## AIC: 67.859
##
## Number of Fisher Scoring iterations: 5
```

```
# age is not significant in the overall model, however, is marginally significant


# fit quadratic terms for the continous variables to test non-linearity
model6 <- glm(infected ~ age + weight + sex + I(weight^2) + I(age^2), family = binomial)
summary(model6) # significant
```

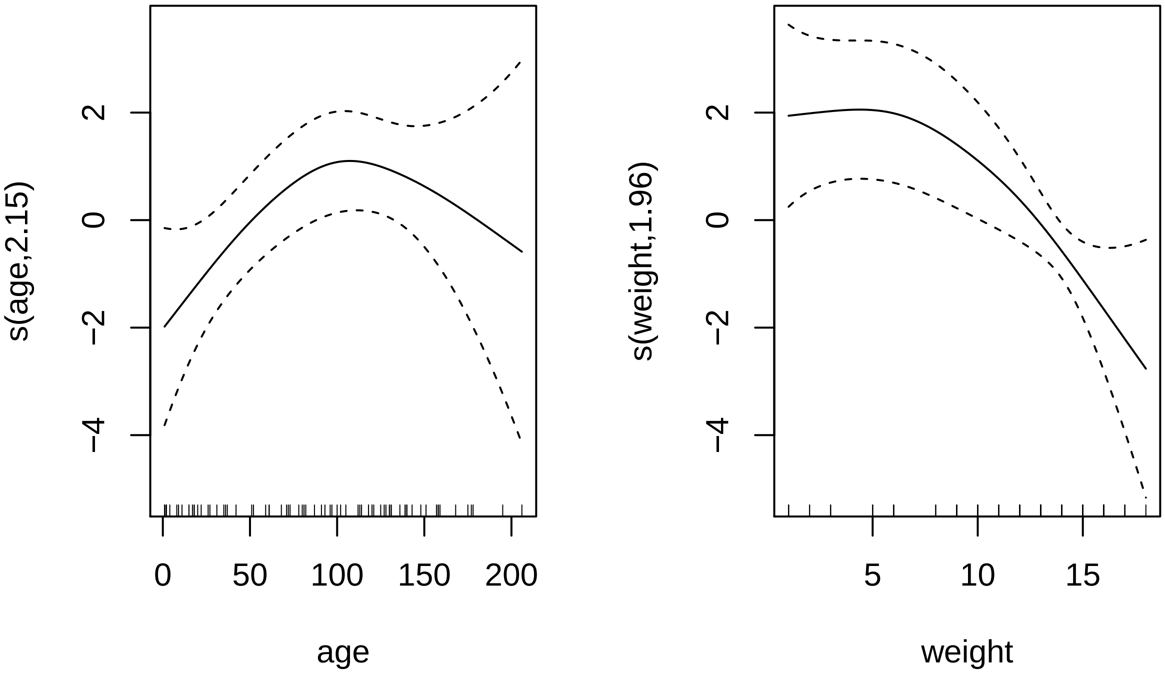
```
##
## Call:
## glm(formula = infected ~ age + weight + sex + I(weight^2) + I(age^2),
##     family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.70226  -0.44412  -0.19584  -0.02505   2.36653
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.4475839  1.7978359  -1.918   0.0552 .
## age          0.0829364  0.0360205   2.302   0.0213 *
## weight       0.4466284  0.3372352   1.324   0.1854
## sexmale     -1.2203683  0.7683288  -1.588   0.1122
## I(weight^2) -0.0415128  0.0209677  -1.980   0.0477 *
## I(age^2)    -0.0004009  0.0002004  -2.000   0.0455 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 48.620  on 75  degrees of freedom
## AIC: 60.62
##
## Number of Fisher Scoring iterations: 6
```

```
# looking at the non-linearities in more detail,

# see if we can do better with other kinds of models such as
# non-parametric smoothers, piecewise linear models or step functions


# gam
```

```
library(mgcv)
```

```
## Loading required package: nlme

## This is mgcv 1.8-16. For overview type 'help("mgcv-package")'.
```

```
model7 <- gam(infected ~ sex + s(age) + s(weight), family = binomial)
par(mfrow=c(1,2))
plot.gam(model7)
```



```
par(mfrow = c(1, 1))

# piecewise linear with threshold from above plots by lowest residual deviance
model8 <- glm(infected ~ sex + age + I(age^2) + I((weight - 12) * (weight > 12)),
              family = binomial)
summary(model8)
```

```
##
## Call:
## glm(formula = infected ~ sex + age + I(age^2) + I((weight - 12) *
##     (weight > 12)), family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.56653  -0.38639  -0.09629  -0.01089   2.24920
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.7511382  1.3678824  -2.011   0.0443 *
## sexmale                         -1.2864683  0.7349201  -1.750   0.0800 .
## age                              0.0798629  0.0348184   2.294   0.0218 *
## I(age^2)                        -0.0003892  0.0001955  -1.991   0.0465 *
## I((weight - 12) * (weight > 12)) -1.3547520  0.5350853  -2.532   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 48.687  on 76  degrees of freedom
## AIC: 58.687
##
## Number of Fisher Scoring iterations: 7
```

```
# minimal adequate
model9 <- glm(infected ~ age + I(age^2) + I((weight - 12) * (weight > 12)), family = binomial)
summary(model9)
```

```
##
## Call:
## glm(formula = infected ~ age + I(age^2) + I((weight - 12) * (weight >
##     12)), family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.42301  -0.50141  -0.13277  -0.01416   2.11658
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -3.1207552  1.2665593  -2.464   0.0137 *
## age                               0.0765784  0.0323376   2.368   0.0179 *
## I(age^2)                         -0.0003843  0.0001846  -2.081   0.0374 *
## I((weight - 12) * (weight > 12)) -1.3511706  0.5134681  -2.631   0.0085 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 51.953  on 77  degrees of freedom
## AIC: 59.953
##
## Number of Fisher Scoring iterations: 7
```

```
detach(infection)
```

## Binary response with pseudoreplication

- General linear mixed effects model
- Only use the data measured the last (or any specified)
- Convert to proportions and use binomial or quasi-binomial family within glm

```
library(MASS)
attach(bacteria)
names(bacteria)
```

```
## [1] "y"    "ap"   "hilo" "week" "ID"   "trt"
```

```r
table(y)
```

```
## y
##   n   y
##  43 177
```

```r
# yes or no for infection

table(y, trt) # three treatments
```

```
##    trt
## y   placebo drug drug+
##   n      12   18    13
##   y      84   44    49
```

```r
#  random effects defined by the round brackets
# and the "given" operator to separate the continuous
# random effect(week) from the categorical random effect
# (ID)
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'lme4'

## The following object is masked from 'package:nlme':
##
##     lmList
```

```r
model1 <- glmer(y ~ trt + (week | ID), family = binomial)
summary(model1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: y ~ trt + (week | ID)
##
##      AIC      BIC   logLik deviance df.resid
##    209.2    229.6    -98.6    197.2      214
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7547  0.1835  0.2550  0.3989  1.3075
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  ID     (Intercept) 0.14772  0.3843
##         week        0.06236  0.2497   1.00
## Number of obs: 220, groups:  ID, 50
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.6192     0.5376   4.872  1.1e-06 ***
## trtdrug      -1.2183     0.6669  -1.827   0.0677 .
## trtdrug+     -0.5288     0.7057  -0.749   0.4537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) trtdrg
## trtdrug  -0.694
## trtdrug+ -0.633  0.509
```

```
# week random effect not significant
# fixed effects not significant

# remove the dependence of infection on week
model2 <- glmer(y ~ trt + (1|ID), family = binomial)
anova(model1, model2)
```

```
## Data: NULL
## Models:
## model2: y ~ trt + (1 | ID)
## model1: y ~ trt + (week | ID)
##        Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
## model2  4 214.32 227.90 -103.162   206.32
## model1  6 209.21 229.57  -98.603   197.21 9.1182      2    0.01047 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# accept model1


# combine drug and drug+
drugs <- factor(1 + (trt != "placebo"))
table(y, drugs)
```

```
##    drugs
## y    1  2
##   n 12 31
##   y 84 93
```

```
model3 <- glmer(y ~ drugs + (week|ID), family = binomial)
summary(model3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: y ~ drugs + (week | ID)
##
##      AIC      BIC   logLik deviance df.resid
##    208.2    225.2    -99.1    198.2      215
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7785  0.1816  0.2499  0.3966  1.2955
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  ID     (Intercept) 0.19639  0.4432
##         week        0.05911  0.2431   1.00
## Number of obs: 220, groups:  ID, 50
```

```
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.6416     0.5482   4.819 1.45e-06 ***
## drugs2       -0.8987     0.6020  -1.493    0.135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
## drugs2 -0.762
```

```
# sample size too small to demonstrate the significance of its efficiency

table(y, trt)
```

```
##    trt
## y   placebo drug drug+
##   n      12   18    13
##   y      84   44    49
```

```
# wrong way to do proportion test due to the pseudo replication
# as seen above, the effect is not significant, however significant here
prop.test(c(12, 18, 13), c(96, 62, 62)) # the second argument is the total
```

```
##
##  3-sample test for equality of proportions without continuity
##  correction
##
## data:  c(12, 18, 13) out of c(96, 62, 62)
## X-squared = 6.6585, df = 2, p-value = 0.03582
## alternative hypothesis: two.sided
## sample estimates:
##    prop 1    prop 2    prop 3
## 0.1250000 0.2903226 0.2096774
```

```
# one way to deal with pseudo replication is to only
# use the data from the end of the experiment

# check if there are obs that are measured twice within a week
head(table(ID, week))
```

```
##      week
## ID    0 2 4 6 11
##   X01 1 1 1 0  1
##   X02 1 1 0 1  1
##   X03 1 1 1 1  1
##   X04 1 1 1 1  1
##   X05 1 1 1 1  1
##   X06 1 1 1 0  1
```

```
any(table(ID, week) > 1) # no
```

```
## [1] FALSE
```

```
# fit model with a subset of data
model <- glm(y ~ trt, family = binomial, subset = (week == 11))
summary(model)
```

```
## 
## Call:
## glm(formula = y ~ trt, family = binomial, subset = (week == 11))
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.7941  -1.4823   0.6681   0.9005   0.9005  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)  
## (Intercept)   1.3863     0.5590   2.480   0.0131 *
## trtdrug      -0.6931     0.8292  -0.836   0.4032  
## trtdrug+     -0.6931     0.8292  -0.836   0.4032  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 51.564  on 43  degrees of freedom
## Residual deviance: 50.569  on 41  degrees of freedom
## AIC: 56.569
## 
## Number of Fisher Scoring iterations: 4
```

```
# combine drug levels
drugs <- factor(1 + (trt == "placebo"))


table(drugs[week == 11])
```

```
## 
##  1  2 
## 24 20
```

```
# refit
model <- glm(y ~ drugs, family = binomial, subset = (week == 11))
summary(model)
```

```
## 
## Call:
## glm(formula = y ~ drugs, family = binomial, subset = (week == 
##     11))
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.7941  -1.4823   0.6681   0.9005   0.9005  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.6931     0.4330   1.601    0.109
## drugs2        0.6931     0.7071   0.980    0.327
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
```

```
##     Null deviance: 51.564  on 43  degrees of freedom
## Residual deviance: 50.569  on 42  degrees of freedom
## AIC: 54.569
##
## Number of Fisher Scoring iterations: 4
```

```
# not significant drug effect



# convert the data into proportions so each patient have one proportion
dss <- data.frame(table(trt, ID))
head(dss)
```

```
##       trt  ID Freq
## 1 placebo X01    4
## 2    drug X01    0
## 3   drug+ X01    0
## 4 placebo X02    0
## 5    drug X02    0
## 6   drug+ X02    4
```

```
# only select the treatment and patients combination with Freq > 0
tss <- dss[dss[, 3] > 0, ]$trt
ys <- table(y, ID)
yv <- cbind(ys[2, ], ys[1, ])


# fit
model <- glm(yv ~ tss, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = yv ~ tss, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5061  -0.7907   0.9326   1.1556   1.8519
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.9459     0.3086   6.306 2.87e-10 ***
## tssdrug      -1.0521     0.4165  -2.526   0.0115 *
## tssdrug+     -0.6190     0.4388  -1.411   0.1583
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 86.100  on 49  degrees of freedom
## Residual deviance: 79.444  on 47  degrees of freedom
## AIC: 130.9
##
## Number of Fisher Scoring iterations: 4
```

```r
# overdispersion

# refit
model <- glm(yv ~ tss, family = quasibinomial)
summary(model)
```

```
##
## Call:
## glm(formula = yv ~ tss, family = quasibinomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5061  -0.7907   0.9326   1.1556   1.8519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9459     0.3837   5.071 6.62e-06 ***
## tssdrug      -1.0521     0.5180  -2.031   0.0479 *
## tssdrug+     -0.6190     0.5457  -1.134   0.2624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.546375)
##
##     Null deviance: 86.100  on 49  degrees of freedom
## Residual deviance: 79.444  on 47  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```r
# combine two drug effects
tss2 <- factor(1 + (tss == "placebo"))
model <- glm(yv ~ tss2, family = quasibinomial)
summary(model)
```

```
##
## Call:
## glm(formula = yv ~ tss2, family = quasibinomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5061  -0.7356   0.9643   1.1556   1.6961
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0986     0.2582   4.255 9.64e-05 ***
## tss22         0.8473     0.4629   1.830   0.0734 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.55006)
##
##     Null deviance: 86.100  on 49  degrees of freedom
## Residual deviance: 80.523  on 48  degrees of freedom
## AIC: NA
```

```
##
## Number of Fisher Scoring iterations: 4
```

```
# no significant drug effects, consistent with mixed effects models
detach(bacteria)
```

# Chapter 18 Generalized Additive Models

Useful when we have no a *priori* reason to choose a particular parametric model.

All error families allowed with `glm` are available, `update` , `predict`, `summary`, `anova` and so on are also available.

- s(x, z) will do isotropic smooth.
- s(x, z) + s(z, w) is allowed for overlapping terms.
- te(x, z, k = 6) (example k) smooths interactions of any number of variables via scale invariant tensor product smooths.
- s(z, bs = "cr", k = 6) (example) do smoothing with cubic regression spline(cr), while the default is "tp".

**Technical aspects**:

- The degree of smoothness of model terms is estimated as part of the fitting
- Isotropic or scale-invariant smooths of any number of variables are available as model terms
- Confidence or credible intervals are readily available for any quantity predicted using a fitted model
- In `mgcv`, `gam` solves the smoothing parameter estimation by using

1. the generalized cross validation(GCV): $GCV = \frac{nD}{(n-d.f.)^2}$.
2. unbiased risk estimator(UBRE) when $\phi$ is known: $UBRE = \frac{D}{n} + 2\phi\frac{d.f.}{n} - \phi$.

See `?gam.method` for more details.

## Non-parametric smoothers

- `loess`
- `tree`

```
soay <- read.table("soaysheep.txt", header = TRUE)
attach(soay)
names(soay)
```

```
## [1] "Year"       "Population" "Delta"
```

```
# Delta is the yearly change, population is the density


plot(Population, Delta, pch = 21, col = "green", bg = "red")

model <- loess(Delta ~ Population)
# loess : Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

summary(model)
```

```
## Call:
## loess(formula = Delta ~ Population)
##
## Number of Observations: 44
## Equivalent Number of Parameters: 4.66
## Residual Standard Error: 0.2616
## Trace of smoother matrix: 5.11  (exact)
##
## Control settings:
##   span     :  0.75
##   degree   :  2
##   family   :  gaussian
##   surface  :  interpolate      cell = 0.2
##   normalize:  TRUE
##  parametric:  FALSE
## drop.square:  FALSE
```

```
# draw smoothed line
xv <- seq(600, 2000, 1)
yv <- predict(model, data.frame(Population = xv))
lines(xv, yv, col = "red") # looks like a step function

rm(xv, yv)


# use tree to determine the threshold for splitting the data into low and high density parts
library(tree)
thresh <- tree(Delta ~ Population)
print(thresh)
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 44 5.2870  0.006208
##    2) Population < 1289.5 25 0.8596  0.226500
##      4) Population < 1009.5 13 0.2364  0.277600 *
##      5) Population > 1009.5 12 0.5525  0.171200
##       10) Population < 1059.5 5 0.1631  0.072120 *
##       11) Population > 1059.5 7 0.3053  0.241900 *
##    3) Population > 1289.5 19 1.6180 -0.283700
##      6) Population < 1459 9 0.7917 -0.349500 *
##      7) Population > 1459 10 0.7519 -0.224400 *
```

```
# plot(thresh)

th <- 1289.5 # threshold


model2 <- aov(Delta ~ (Population > th))
summary(model2)
```

```
##                  Df Sum Sq Mean Sq F value   Pr(>F)
## Population > th  1  2.810   2.810   47.63 2.01e-08 ***
## Residuals       42  2.477   0.059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

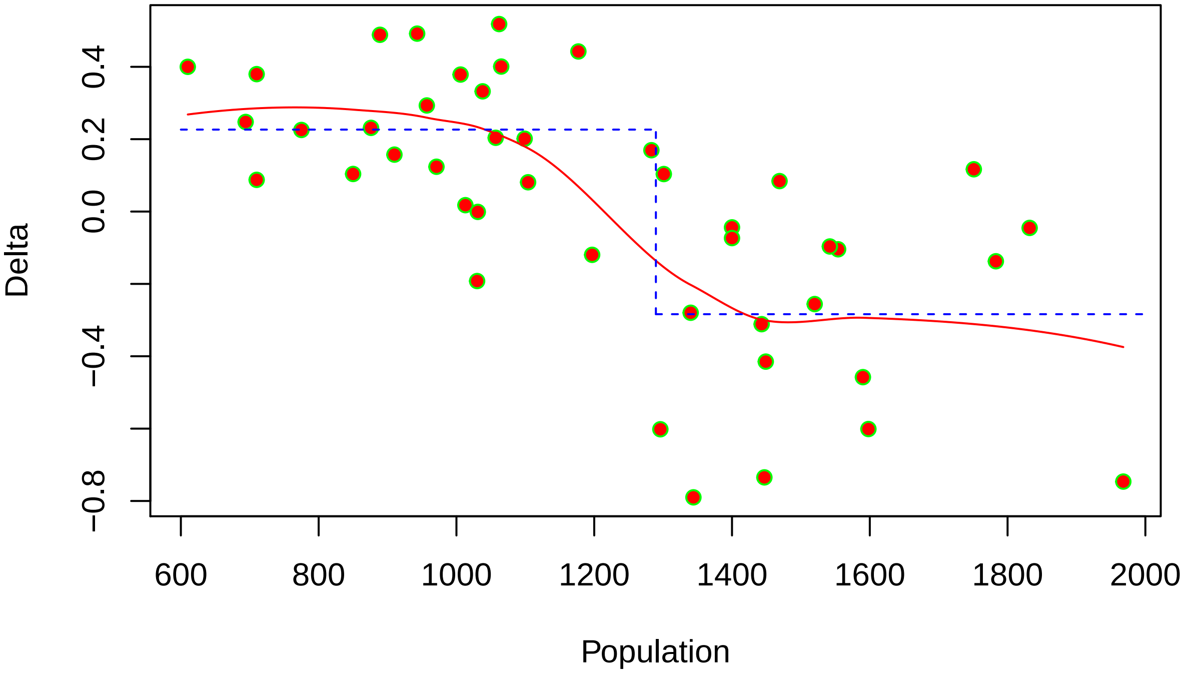
```
tail(Delta, 2) # the 45th data is NA , remove it
```

```
## [1] -0.7463679         NA
```

```
tapply(Delta[-45], (Population[-45] > th), mean)
```

```
##      FALSE       TRUE
##  0.2265084 -0.2836616
```

```
# add step functions
lines(c(600, th), c(0.2265, 0.2265), lty = 2, col = "blue")
lines(c(th, 2000), c(-0.2837, -0.2837), lty = 2, col = "blue")
lines(c(th, th), c(-0.2837, 0.2265), lty = 2, col = "blue")
```



```
# Three parameters (two averages and a threshold) in step function,
# 4.66 df for loess,
# parsimony favours the step function
detach(soay)
```

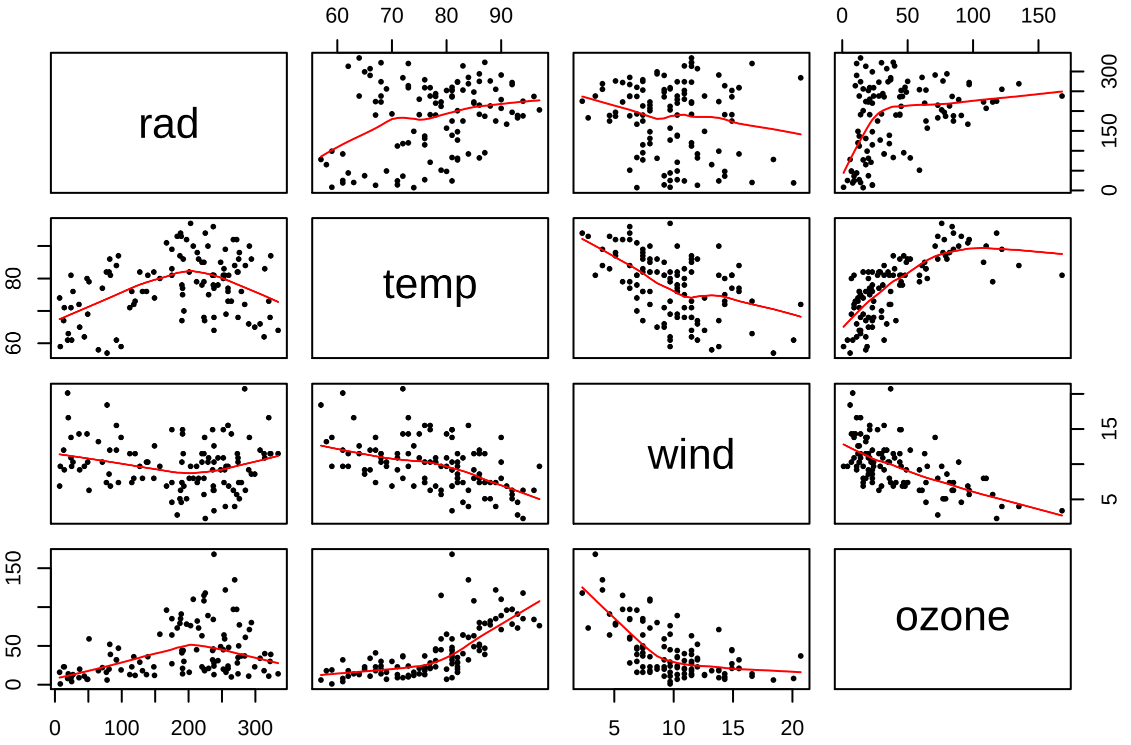
### Generalized additive models

gam is used.

```
ozone.data <- read.table("ozone.data.txt", header = TRUE)
attach(ozone.data)
names(ozone.data)
```

```
## [1] "rad"   "temp"  "wind"  "ozone"
```

```
# ozone is y , the other three are continuous variables
# inspect the data with non parametric loess
pairs(ozone.data, panel = function(x, y) {points(x, y, pch = 16, cex = 0.6); lines(lowess(x, y), col =
```



```
# fit all variables with non parametric smoothers s()
# s() does not evaluate a (spline) smooth - it exists purely to help set up a model using spline based 

model <- gam(ozone ~ s(rad) + s(temp) + s(wind))
summary(model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ozone ~ s(rad) + s(temp) + s(wind)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    42.10       1.66   25.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df      F  p-value
## s(rad)  2.763  3.451  3.964   0.0085 **
## s(temp) 3.841  4.762 11.612 8.19e-09 ***
## s(wind) 2.918  3.666 13.770 1.39e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =  0.724   Deviance explained = 74.8%
## GCV =    338  Scale est. = 305.96    n = 111
```

```r
# add interaction term using update
model2 <- update(model, ~ . + s(wind, temp))
summary(model2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ozone ~ s(rad) + s(temp) + s(wind) + s(wind, temp)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.099      1.361   30.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(rad)         1.389  1.667 5.799  0.0126 *
## s(temp)        1.000  1.000 0.000  0.9892
## s(wind)        5.613  6.482 2.492  0.0244 *
## s(wind,temp) 18.246 27.000 2.805 8.5e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.814   Deviance explained = 85.9%
## GCV = 272.66  Scale est. = 205.72    n = 111
```

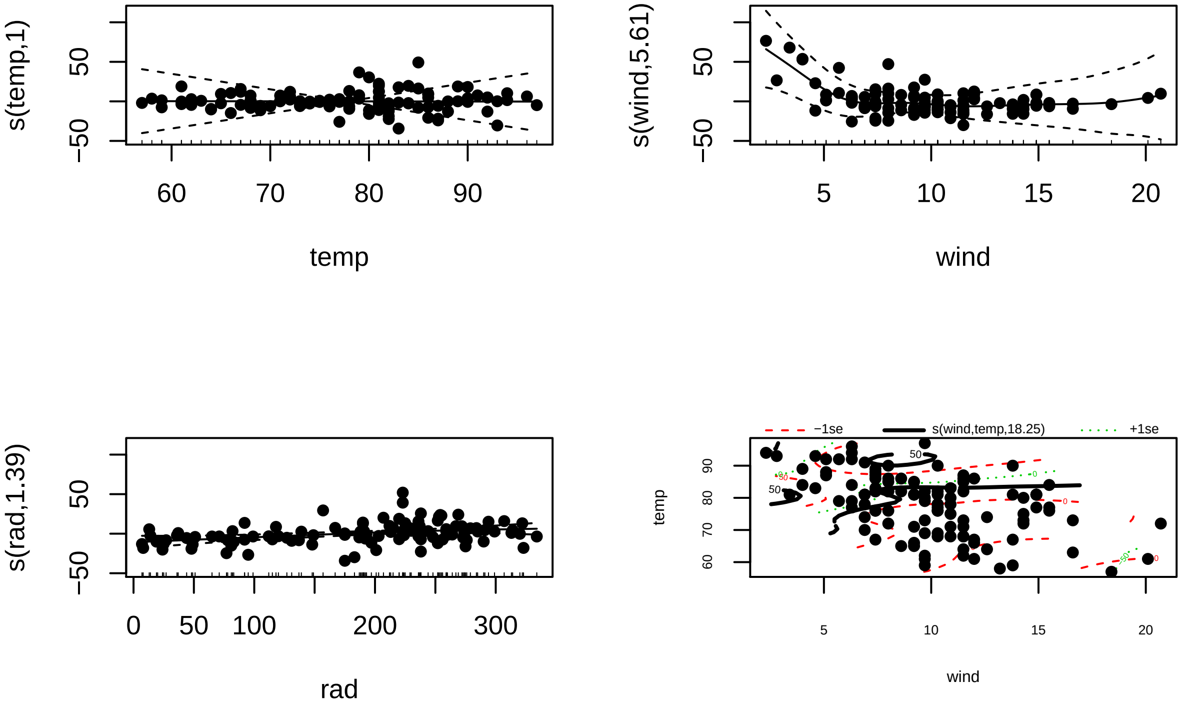
```r
# write out the model
model3 <- gam(ozone ~ s(temp) + s(wind) + s(rad) + s(wind,temp))
summary(model3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## ozone ~ s(temp) + s(wind) + s(rad) + s(wind, temp)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.099      1.361   30.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(temp)        1.000  1.000 0.000  0.9892
## s(wind)        5.613  6.482 2.492  0.0244 *
## s(rad)         1.389  1.667 5.799  0.0126 *
## s(wind,temp) 18.246 27.000 2.805 8.5e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.814   Deviance explained = 85.9%
## GCV = 272.66  Scale est. = 205.72    n = 111
```

```
anova(model2, model3) # these two models should be the same
```

```
## Analysis of Deviance Table
##
## Model 1: ozone ~ s(rad) + s(temp) + s(wind) + s(wind, temp)
## Model 2: ozone ~ s(temp) + s(wind) + s(rad) + s(wind, temp)
##   Resid. Df Resid. Dev        Df   Deviance
## 1    78.791      17230
## 2    78.791      17230 1.62e-12 6.5484e-11
```

```
par(mfrow=c(2, 2))
plot(model3, residuals = TRUE, pch = 16)
```



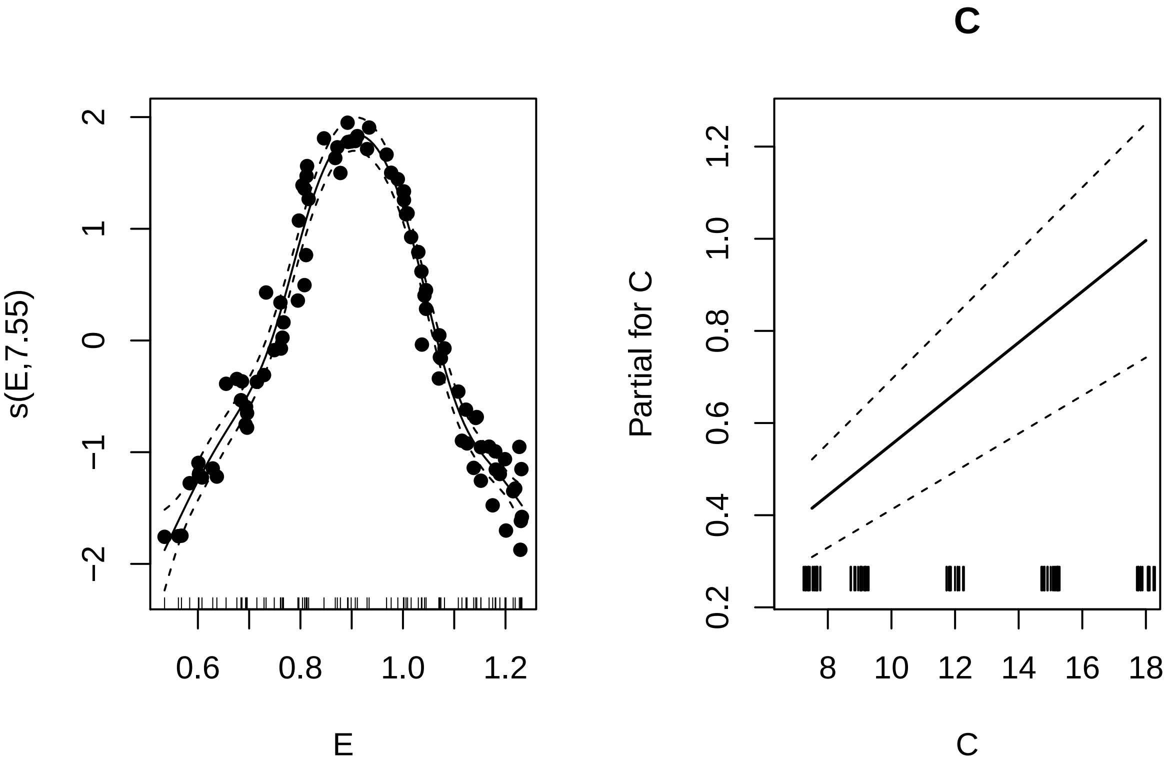
```
par(mfrow = c(1, 1))

detach(ozone.data)
```

## An example with strongly humped data

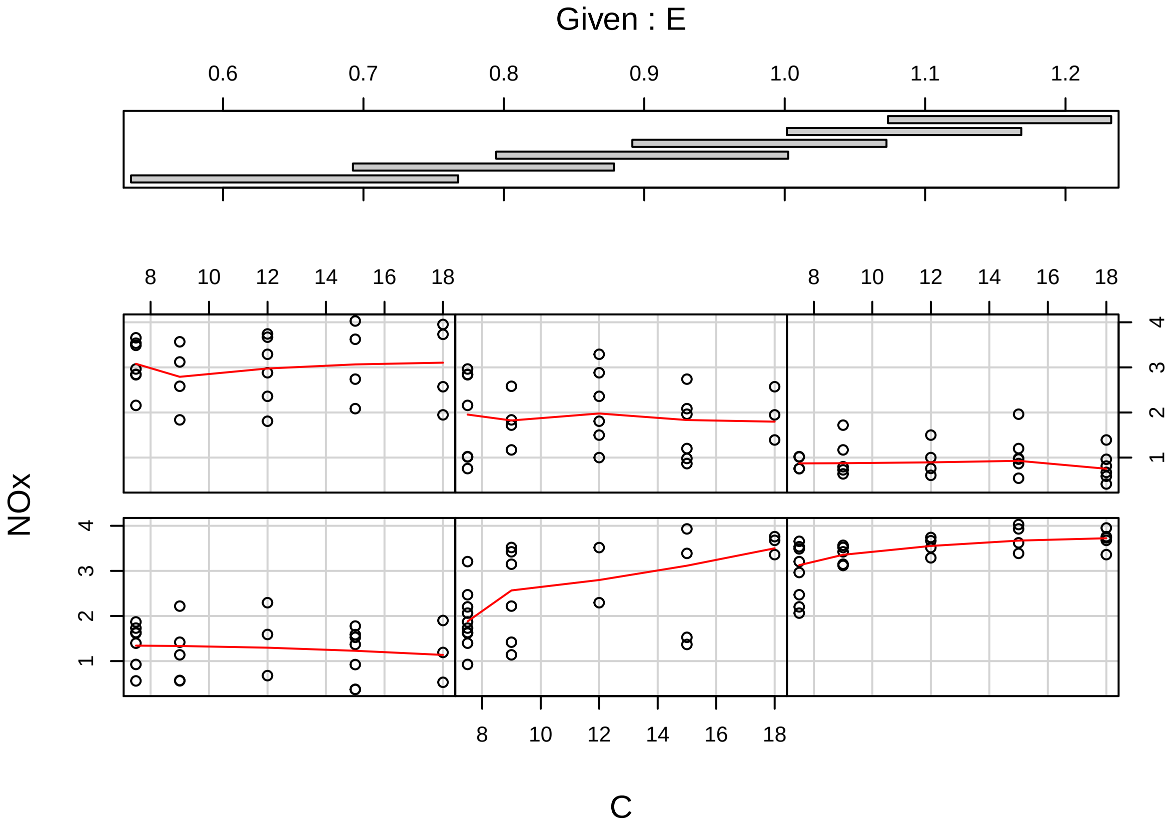
```
# install.packages("SemiPar")
library(SemiPar)
data(ethanol)
attach(ethanol)
head(ethanol)
```

```
##     NOx  C     E
## 1 3.741 12 0.907
## 2 2.295 12 0.761
## 3 1.498 12 1.108
## 4 2.881 12 1.016
## 5 0.760 12 1.189
## 6 3.120  9 1.001
```

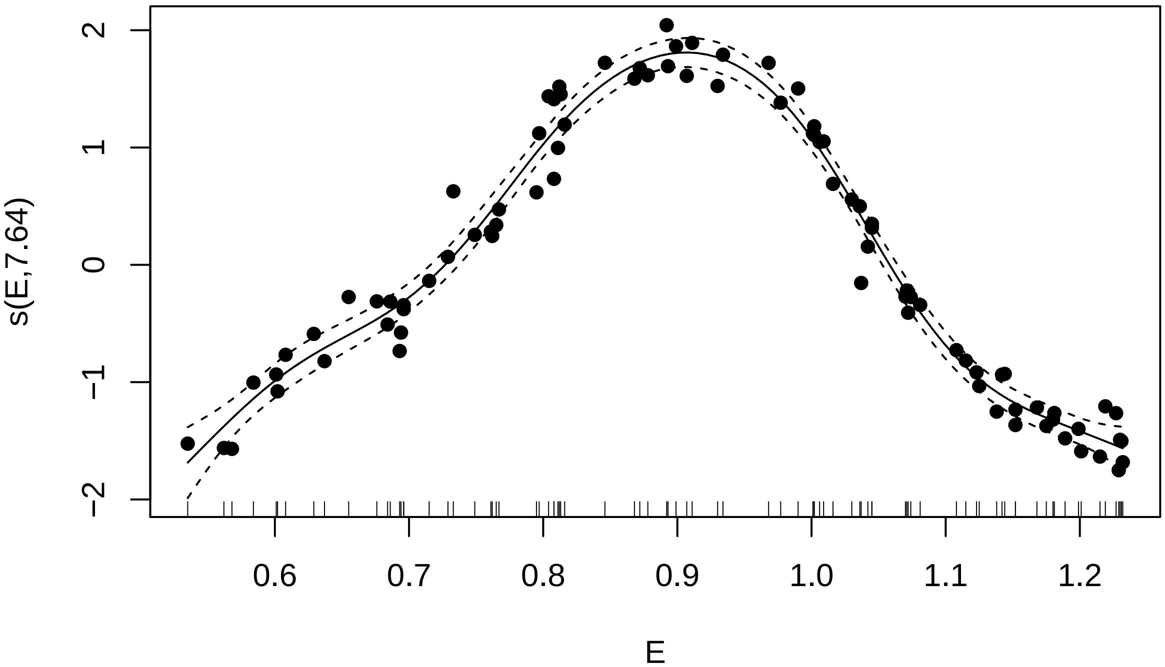
```
# NOx is y

# fit E as smoothed term and C as parametric term
model <- gam(NOx ~ s(E) + C)

par(mfrow=c(1,2))
plot.gam(model, residuals = T, pch = 16, all.terms = T)
```

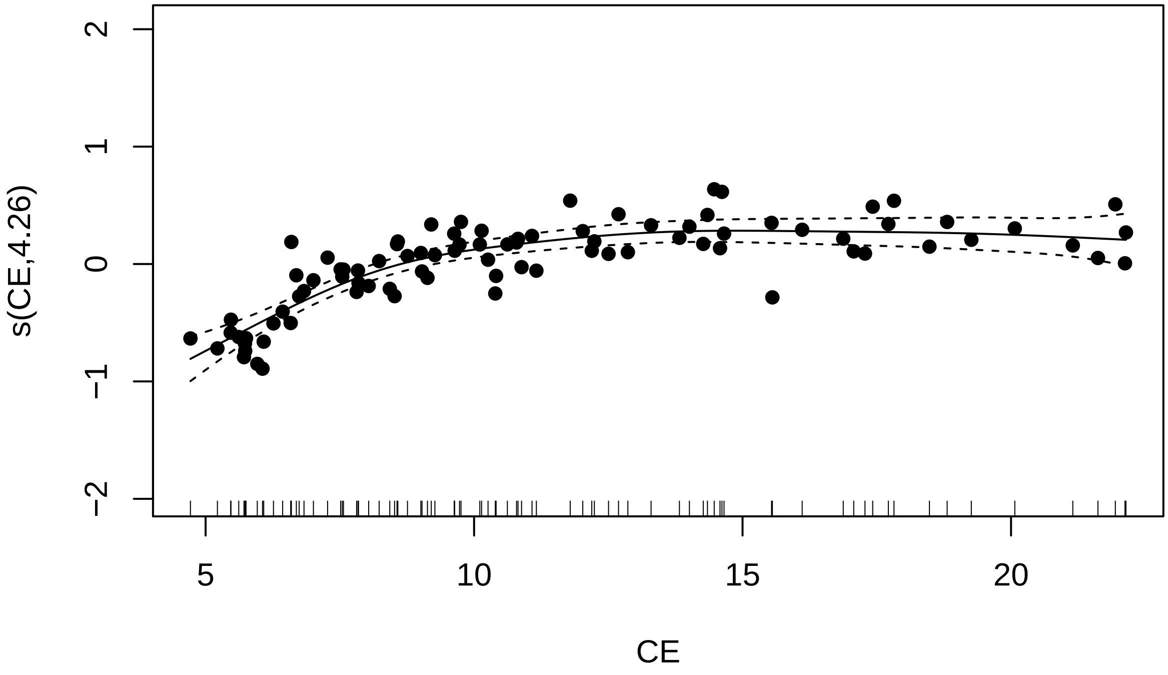


```
coplot(NOx ~ C | E, panel = panel.smooth) # only panel 2 has a pronounced effect
```



```
par(mfrow = c(1, 1))


# add interaction term without C
CE <- E * C
model2 <- gam(NOx ~ s(E) + s(CE))


plot.gam(model2, residuals = TRUE, pch = 16, all.terms = T)
```





```
summary(model2) # significant
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## NOx ~ s(E) + s(CE)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.95737    0.02126   92.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(E)  7.636  8.509 270.17  <2e-16 ***
## s(CE) 4.261  5.224  25.25  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.969   Deviance explained = 97.3%
## GCV = 0.0466  Scale est. = 0.039771  n = 88
```

```
detach(ethanol)
```

### Generalized additive models with binary data

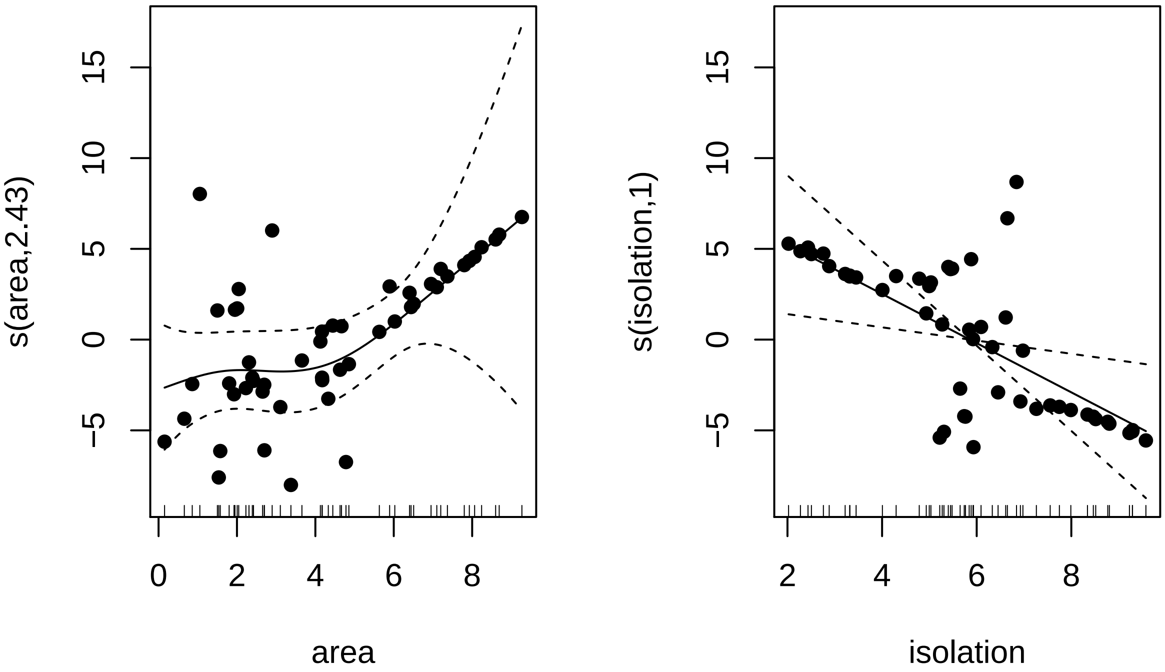
```
attach(island)
names(island)
```

```
## [1] "incidence" "area"      "isolation"
```

```
model3 <- gam(incidence ~ s(area) + s(isolation), family = binomial)
summary(model3) # area not significant
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## incidence ~ s(area) + s(isolation)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.6371     0.9898   1.654   0.0981 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df Chi.sq p-value
## s(area)      2.429  3.066  3.455 0.32945
## s(isolation) 1.000  1.000  7.480 0.00624 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.63   Deviance explained = 63.1%
## UBRE = -0.32096  Scale est. = 1         n = 50
```

```
par(mfrow=c(1, 2))
plot.gam(model3, residuals = TRUE, pch = 16)
```



```r
# fit isolation alone
model4 <- gam(incidence ~ s(isolation), family = binomial)
anova(model3, model4, test = "Chisq") # model3 preferred
```

```
## Analysis of Deviance Table
##
## Model 1: incidence ~ s(area) + s(isolation)
## Model 2: incidence ~ s(isolation)
##   Resid. Df Resid. Dev       Df Deviance Pr(>Chi)
## 1    44.934     25.094
## 2    45.191     29.127 -0.25709   -4.033 0.007425 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# fit area as parametric term
model5 <- gam(incidence ~ area + s(isolation), family = binomial)
summary(model5) # significant
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## incidence ~ area + s(isolation)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3928     0.9002  -1.547   0.1218
## area          0.5807     0.2478   2.344   0.0191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq p-value
## s(isolation)   1      1  8.276 0.00402 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.597   Deviance explained = 58.3%
## UBRE = -0.31196  Scale est. = 1         n = 50
```

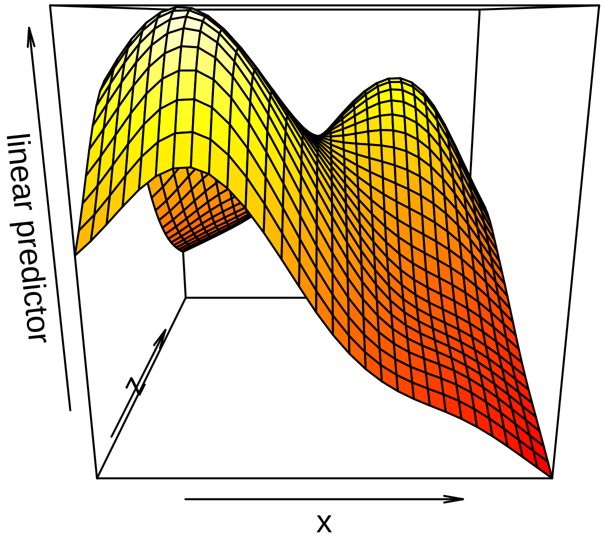
```
detach(island)
```

**Summary**: a term can appear to be significant when entered as a parametric term but not when as a non-parametric term.

## Three-dimensional graphic output from gam

vis.gam is used when there are two continuous explanatory variables. It produces perspective or contour plot views of gam model predictions, fixing all but the values in view to the values supplied in cond.

```
test1 <- function(x, z, sx = 0.3, sz = 0.4) {
  (pi**sx*sz) * (1.2 * exp(- (x - 0.2)^2/sx^2 - (z - 0.3)^2/sz^2) +
                 0.8*exp(- (x - 0.7)^2/sx^2 - (z - 0.8)^2/sz^2))
}



n <- 500
x <- runif(n); z <- runif(n); # random variables from unif(0, 1)
y <- test1(x, z) + rnorm(n) * 0.1
b4 <- gam(y ~ s(x, z))
vis.gam(b4)
```



```
# z axis is the linear predictor
```