FPA2020 Lecture 12: Several regression models with examples

Toshihide Kitakado

June 1, 2020

Contents

```
library(mgcv)
library(ggplot2)
library(fields)
library(marmap)
```

	Distributional assumption	Regression component	R function
Normal linear model	Normal	Linear	"lm"
Normal nonlinear model	Normal	Nonlinear	"nls"
Generalized linear model (GLM)	Exponential family (Normal, Gamma, Binomial, Poisson etc)	Linear through "a link function"	"glm"
Additive model	Normal	Nonparametric	"gam"
Generalized additive model (GAM)	Exponential family (Normal, Gamma, Binomial, Poisson etc)	Nonparametric through "a link function"	"gam"

[40%]

1 Linear model (1) allometric analysis

1.1 Reading the data

```
Data <- read.csv("Data/Rainbowtrout.csv")
names(Data)
```

```
## [1] "Length" "Weight"
```

```
head(Data, 5)
```

```
##   Length Weight
## 1    7.5     10
## 2    9.2      9
## 3    9.5     25
## 4    9.7     35
## 5   10.5     20
```

```
tail(Data, 3)
```

```
##    Length Weight
## 37   48.7   1700
```

```
## 38   49.2   1290
## 39   51.7   1650
```

```
Length <- Data$Length
Weight <- Data$Weight
```

1.2 Analyzig the data

1.2.1 Definition of model

We will use the following allometric function:

$$W = aL^b$$

1.2.2 Simple linear model

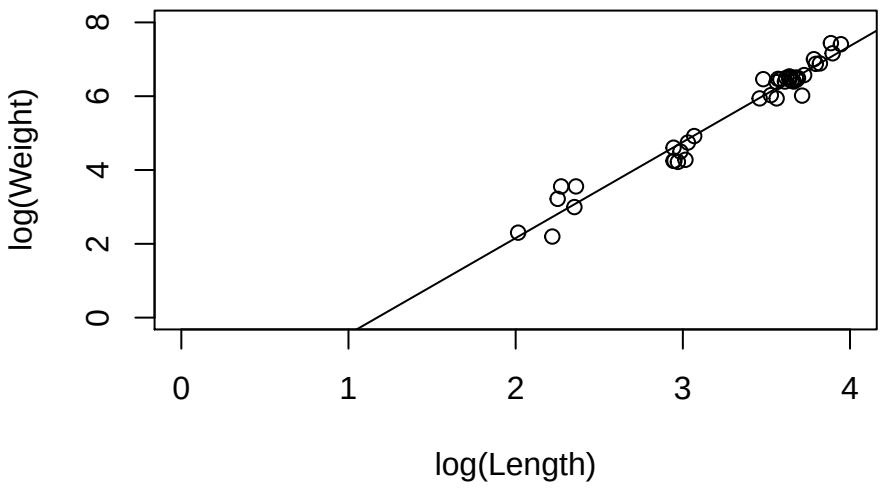
We then conduct a simple linear regression analysis with the following simple linear model:

$$\log W = \log a + b \log L.$$

```
res <- lm(log(Weight)~log(Length))
summary(res)
```

```
##
## Call:
## lm(formula = log(Weight) ~ log(Length))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59889 -0.08977 -0.00832  0.14493  0.69146
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.04898    0.28442  -10.72 6.67e-13 ***
## log(Length)  2.60235    0.08474   30.71  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.286 on 37 degrees of freedom
## Multiple R-squared:  0.9622, Adjusted R-squared:  0.9612
## F-statistic: 943.1 on 1 and 37 DF,  p-value: < 2.2e-16
```

```
plot(log(Length), log(Weight), xlim=c(0,4), ylim=c(0,8))
abline(res)
```



1.2.3 Parameter estimates

```
names(res)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
coef(res)
```

```
## (Intercept) log(Length)
##   -3.048982    2.602353
```

```
alpha.est <- as.numeric(coef(res)[1])
beta.est <- as.numeric(coef(res)[2])

a.est <- exp(alpha.est)
b.est <- beta.est
Est <- c(a.est, b.est)
Est
```

```
## [1] 0.04740718 2.60235273
```

1.2.4 Confidence intervals

```
confint(res)
```

```
##                 2.5 %    97.5 %
## (Intercept) -3.625272 -2.472691
## log(Length)  2.430651  2.774054
```

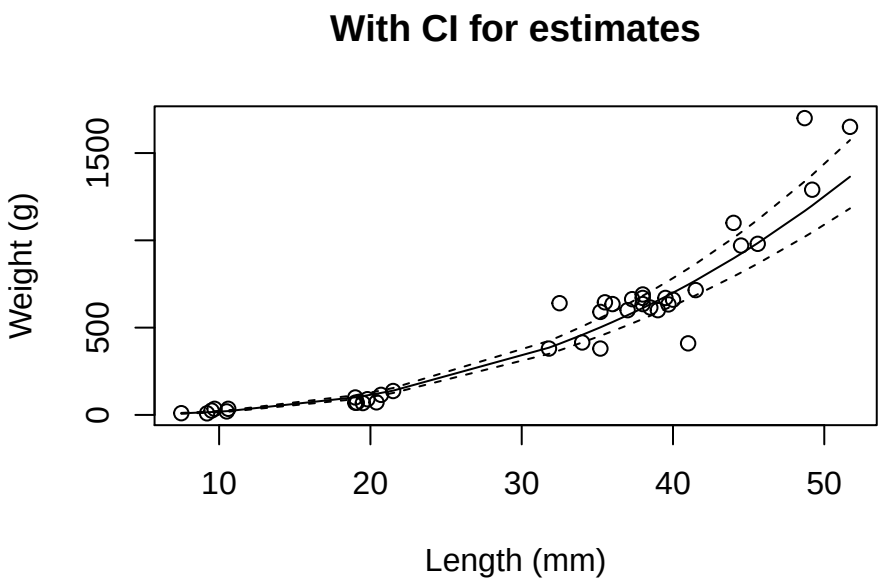
```
alpha.ci <- confint(res)[1,]
beta.ci <- confint(res)[2,]
#alpha.ci
#beta.ci
```

```
a.ci <- exp(alpha.ci)
b.ci <- beta.ci
CI <- rbind(a.ci, b.ci)
#CI

Out <- cbind(Est, CI)
row.names(Out) <- c("a", "b")
Out
```

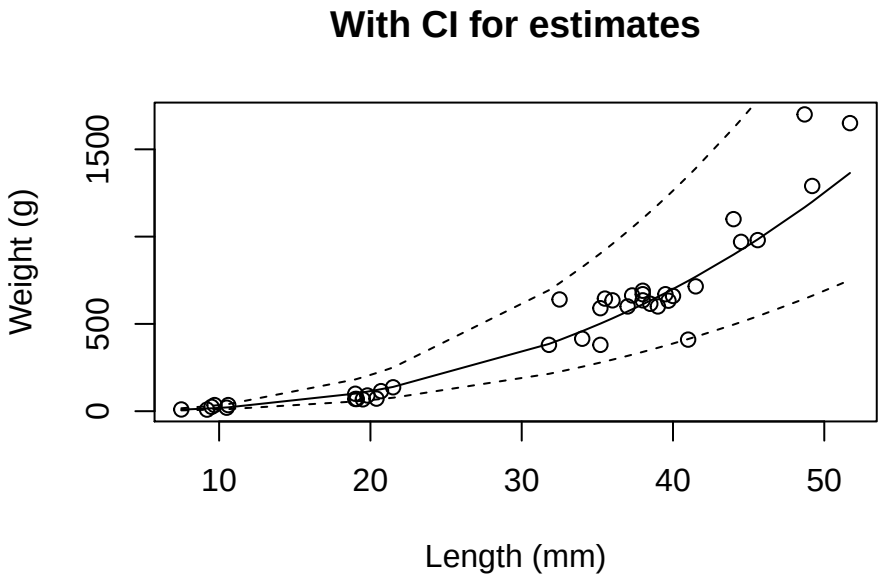
```
##          Est      2.5 %     97.5 %
## a 0.04740718 0.02664184 0.08435757
## b 2.60235273 2.43065137 2.77405408
```

1.3 Showing the results

```
plot(Length, Weight, xlab="Length (mm)",
     ylab="Weight (g)", main="With CI for estimates")
pred <- exp(predict(res, int="c", level = 0.95))
matlines(Length, pred, lty=c(1,2,2), col=1)
```




```
plot(Length, Weight, xlab="Length (mm)",
     ylab="Weight (g)", main="With CI for estimates")
pred <- exp(predict(res, int="p", level = 0.95))
matlines(Length, pred, lty=c(1,2,2), col=1)
```



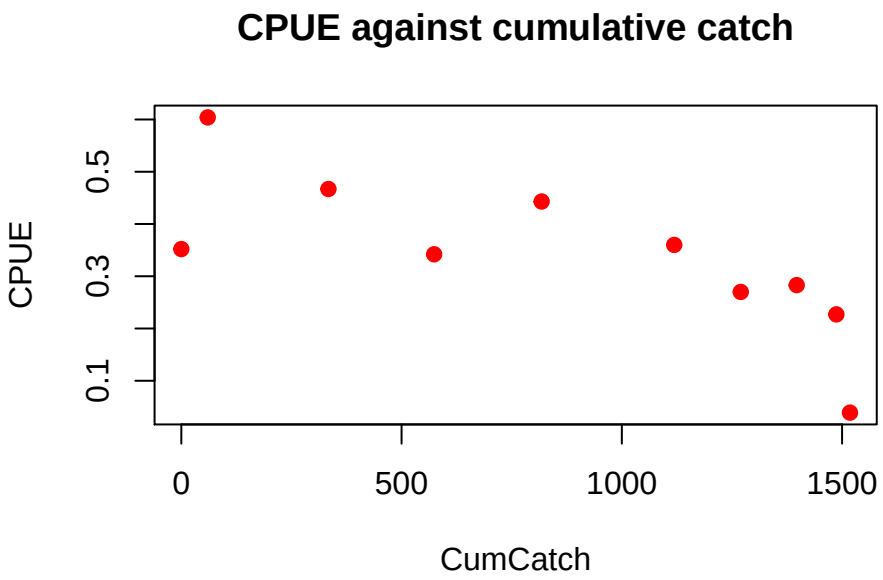
2 Linear model (2) de Lury method

2.1 Reading the data

```
Data <- read.csv("Data/DeLury_R.csv", header=T)
Data
```

```
##    Period Catch Effort  CPUE CumCatch CumEffort
## 1       1    60  170.6 0.352        0       0.0
## 2       2   274  453.8 0.604       60     170.6
## 3       3   240  513.4 0.467      334     624.4
## 4       4   244  714.4 0.342      574    1137.8
## 5       5   301  679.1 0.443      818    1852.2
## 6       6   151  419.9 0.360     1119    2531.3
## 7       7   127  470.3 0.270     1270    2951.2
## 8       8    90  318.4 0.283     1397    3421.5
## 9       9    31  136.8 0.227     1487    3739.9
## 10     10     7  177.6 0.039     1518    3876.7
```

```
CPUE <- Data$CPUE
CumCatch <- Data$CumCatch
CumEffort <- Data$CumEffort

plot(CumCatch, CPUE, pch=19, col="red",
     main="CPUE against cumulative catch")
```

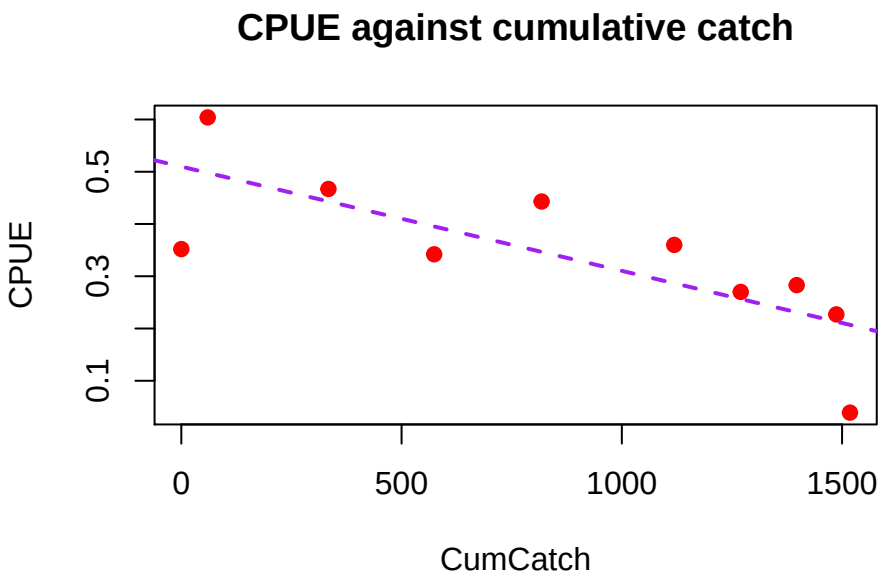


2.2 Analyzing data by De Lury method 1

```
Res.DeLury1 <- lm(CPUE~CumCatch)
summary(Res.DeLury1)
```

```
##
## Call:
## lm(formula = CPUE ~ CumCatch)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.16798 -0.03658  0.01883  0.06804  0.10617
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.098e-01  6.013e-02   8.478 2.87e-05 ***
## CumCatch    -1.995e-04  5.883e-05  -3.391  0.00949 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1034 on 8 degrees of freedom
## Multiple R-squared:  0.5897, Adjusted R-squared:  0.5384
## F-statistic:  11.5 on 1 and 8 DF,  p-value: 0.00949
```

```
plot(CumCatch, CPUE, pch=19, col="red",
     main="CPUE against cumulative catch")
abline(Res.DeLury1, col="purple", lwd=2, lty=2)
```




```
Para1 <- coef(Res.DeLury1)
Para1
```

```
##  (Intercept)     CumCatch
##  0.509796545 -0.000199483
```

```
q.est1 <- (-1)*Para1[2]
N.est1 <- Para1[1]/q.est1

data.frame(q.est1, N.est1)
```

```
##               q.est1   N.est1
## CumCatch 0.000199483 2555.589
```

2.3 Analyzing data by De Lury method 2

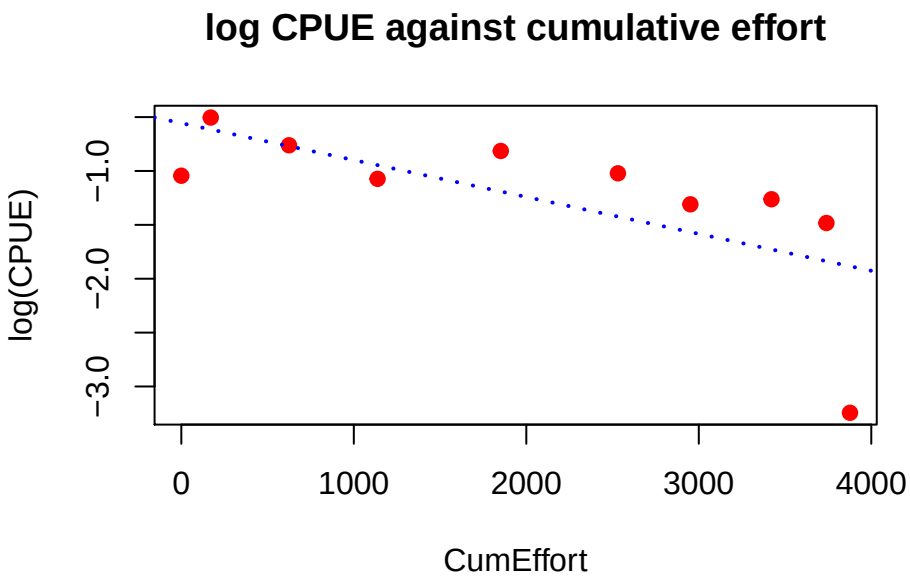
2.3.1 Ignoring M (M=0)

```
Res.DeLury2 <- lm(log(CPUE)~CumEffort)
summary(Res.DeLury2)
```

```
##
## Call:
## lm(formula = log(CPUE) ~ CumEffort)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36018 -0.09308  0.18410  0.37092  0.46580
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5562541  0.3303927  -1.684   0.1308
## CumEffort   -0.0003425  0.0001338  -2.560   0.0336 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5946 on 8 degrees of freedom
## Multiple R-squared:  0.4503, Adjusted R-squared:  0.3816
## F-statistic: 6.553 on 1 and 8 DF,  p-value: 0.03365
```

```
plot(CumEffort, log(CPUE), pch=19, col="red",
     main="log CPUE against cumulative effort")
abline(Res.DeLury2, col="blue", lwd=2, lty=3)
```




```
Para2 <- coef(Res.DeLury2)
Para2
```

```
##   (Intercept)     CumEffort
## -0.5562540837 -0.0003424973
```

```
q.est2 <- (-1)*Para2[2]
N.est2 <- exp(Para2[1])/q.est2

data.frame(q.est2, N.est2)
```

```
##                  q.est2   N.est2
## CumEffort 0.0003424973 1674.036
```

2.3.2 Try to estimate unknown M

```
Period1 <- Data$Period-1

Res.DeLury3 <- lm(log(CPUE) ~ CumEffort + Period1)
summary(Res.DeLury3)
```

```
##
## Call:
## lm(formula = log(CPUE) ~ CumEffort + Period1)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82722 -0.05662  0.16564  0.29739  0.38859
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.2169073  0.3107284  -0.698   0.5077
## CumEffort    0.0014243  0.0008032   1.773   0.1195
## Period1     -0.8726329  0.3929920  -2.220   0.0618 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4869 on 7 degrees of freedom
## Multiple R-squared:  0.6775, Adjusted R-squared:  0.5853
## F-statistic: 7.352 on 2 and 7 DF,  p-value: 0.01905
```

```
Para3 <- coef(Res.DeLury3)
Para3
```

```
##  (Intercept)    CumEffort      Period1
## -0.216907302  0.001424257 -0.872632934
```

```
q.est3 <- (-1)*Para3[2]
N.est3 <- exp(Para3[1])/q.est3
M.est3 <- (-1)*Para3[3]

data.frame(q.est3, N.est3, M.est3)
```

```
##                 q.est3  N.est3    M.est3
## CumEffort -0.001424257 -565.21 0.8726329
```

Does not make sense….

2.3.3 With M (known, fixed at 0.1)

```
Res.DeLury4 <-
  lm(log(CPUE)~CumEffort+offset(-Period1*0.1))

summary(Res.DeLury4)
```

```
##
## Call:
## lm(formula = log(CPUE) ~ CumEffort + offset(-Period1 * 0.1))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28395 -0.06134  0.17919  0.35625  0.43419
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5173664  0.3152973  -1.641    0.139
## CumEffort   -0.0001400  0.0001277  -1.097    0.305
```

```
##
## Residual standard error: 0.5674 on 8 degrees of freedom
## Multiple R-squared:  0.4752, Adjusted R-squared:  0.4096
## F-statistic: 7.244 on 1 and 8 DF,  p-value: 0.02744
```

```
Para4 <- coef(Res.DeLury4)
Para4
```

```
##   (Intercept)     CumEffort
## -0.5173663947 -0.0001400347
```

```
q.est4 <- (-1)*Para4[2]
N.est4 <- exp(Para4[1])/q.est4

data.frame(q.est4, N.est4)
```

```
##                 q.est4   N.est4
## CumEffort 0.0001400347 4256.718
```

2.3.4 With M (known, fixed at 0.01)

```
Res.DeLury5 <-
  lm(log(CPUE)~CumEffort+offset(-Period1*0.01))
summary(Res.DeLury5)
```

```
##
## Call:
## lm(formula = log(CPUE) ~ CumEffort + offset(-Period1 * 0.01))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3526 -0.0899  0.1836  0.3700  0.4626
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5523653  0.3288333   -1.68   0.1315
## CumEffort   -0.0003223  0.0001332   -2.42   0.0418 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5918 on 8 degrees of freedom
## Multiple R-squared:  0.4527, Adjusted R-squared:  0.3842
## F-statistic: 6.616 on 1 and 8 DF,  p-value: 0.03302
```

```
Para5 <- coef(Res.DeLury5)
Para5
```

```
##  (Intercept)    CumEffort
## -0.552365315 -0.000322251
```

```
q.est5 <- (-1)*Para5[2]
N.est5 <- exp(Para5[1])/q.est5
data.frame(q.est5, N.est5)
```

```
##                q.est5   N.est5
```

```
## CumEffort 0.000322251 1786.144
```

```
Method <- 1:5
M <- c("NA", 0, round(M.est3,4), 0.1, 0.01)
N.est <- c(N.est1,N.est2,N.est3,N.est4,N.est5)
data.frame(Method, M, N.est)
```

```
##   Method      M    N.est
## 1      1     NA 2555.589
## 2      2      0 1674.036
## 3      3 0.8726 -565.210
## 4      4    0.1 4256.718
## 5      5   0.01 1786.144
```

3 Generalized linear model (1) Sextual maturity model

3.1 Generalized linear model (GLM)

Why "generalized"
- Distribution assumption (not necessarily "normal distribution")
- Binomial, Poisson, Gamma distributions etc.
- Regression covariates are linked with a "link function"

GLM in R
- "glm"" (almost same syntax as "lm")
- Estimate, CI, testing, AIC, ...

3.2 Maturity as a function of age

3.2.1 Definition

Assumption
- You are analysing "age at sexual maturity (ASM)" of a fish species
- You observe the numbers of mature and immature individuals by age
- Your primary interest is the estimation of "age at 50% maturity (ASM50)"

Structural model
- Maturity is a funciton of age, $p(a)$
- $p(a)$ should be a monotonically increasing function of a like a sigmoid function:

1. Logistic function

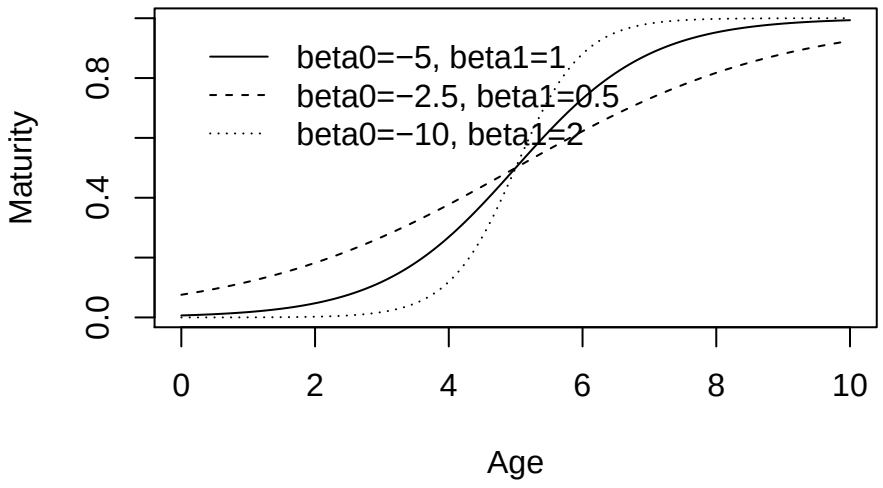
$$p(a) = \frac{e^{\beta_0+\beta_1 a}}{1+e^{\beta_0+\beta_1 a}} \implies \log\frac{p(a)}{1-p(a)} = \beta_0 + \beta_1 a$$

2. Probit function

$$p(a) = \int_{-\infty}^{a} f(z)dz$$

```
# Definition of logistic function (the parameters are beta0 and beta1)
LogisticFn <- function(x,beta0,beta1) {1.0/(1.0+exp(-beta0-beta1*x))}
Age <- seq(0,10,0.01)

# beta0=-5, beta1=1
```

```
plot(Age, LogisticFn(Age,-5,1), type="l", ylab="Maturity", lty=1)

# beta0=-2.5, beta1=0.5
points(Age, LogisticFn(Age,-2.5,0.5), type="l", lty=2)

# beta0=-10, beta1=2
points(Age, LogisticFn(Age,-10,2), type="l", lty=3)

# legends
legend(0, 1, lty=1:3, bty="n",
 legend=c("beta0=-5, beta1=1","beta0=-2.5, beta1=0.5","beta0=-10, beta1=2"))
```



3.2.2 More about logistic function

- Let $ASM50$ be the age at 50% maturity as $p(ASM50) = 0.5$.

$$ASM50 = -\frac{\beta_0}{\beta_1}$$

3.3 Statistical model and generation of example data

Distributional assumption
- Two state (mature/immmature) -> Binomial distribution
- N_a: number of sampled individuals of age a
- Y_a: number of mature individuals among N_a individuals
- $p(a)$: maturity (probability) in age a
- $Y_a \sim Bin(N_a, p(a))$

```
set.seed <- 2020

Amin <- 1
Amax <- 10
AgeRange <- Age <- Amin:Amax
```

```
NAgeClass<- length(AgeRange)

# True beta0=-5, beta1=1
Nvec <- rep(10, NAgeClass)
pvec <- LogisticFn(AgeRange, -5, 1)
pvec
```

```
##  [1] 0.01798621 0.04742587 0.11920292 0.26894142 0.50000000 0.73105858
##  [7] 0.88079708 0.95257413 0.98201379 0.99330715
```

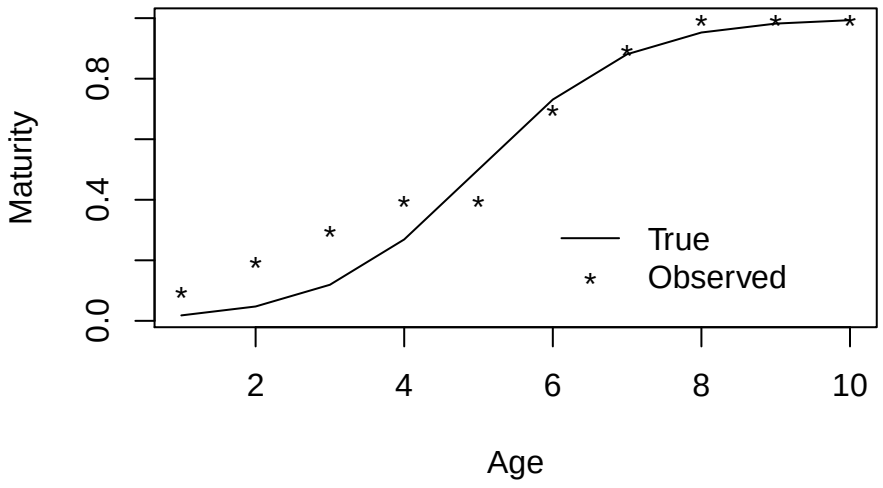
```
# Generation of simulation data
Yvec <- rbinom(NAgeClass, Nvec, pvec)
Yvec
```

```
##  [1]  1  2  3  4  4  7  9 10 10 10
```

```
Maturity.obs <- Yvec/Nvec
Maturity.obs
```

```
##  [1] 0.1 0.2 0.3 0.4 0.4 0.7 0.9 1.0 1.0 1.0
```

```
# Plotting the data
plot(Age, LogisticFn(Age,-5,1), type="l", ylab="Maturity")
points(Age, Maturity.obs, pch="*")
legend(6, 0.4, lty=1:0, pch=c("","*"), legend=c("True","Observed"), bty="n")
```



3.4 Statistical estimation of maturity curve using "glm" function

3.4.1 Estimation

```
Res.glm.logit <- glm(cbind(Yvec,Nvec-Yvec)~Age, family=binomial)
Res.glm.logit
```

```
##
## Call:  glm(formula = cbind(Yvec, Nvec - Yvec) ~ Age, family = binomial)
```

```
##
## Coefficients:
## (Intercept)          Age
##     -3.3441       0.7517
##
## Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
## Null Deviance:       60.24
## Residual Deviance: 4.783     AIC: 25.79
```

```
# Summary
summary(Res.glm.logit)
```

```
##
## Call:
## glm(formula = cbind(Yvec, Nvec - Yvec) ~ Age, family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.28791  -0.01088   0.34994   0.55175   1.15732
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.3441     0.7162  -4.669 3.02e-06 ***
## Age           0.7517     0.1418   5.301 1.15e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 60.2359  on 9  degrees of freedom
## Residual deviance:  4.7833  on 8  degrees of freedom
## AIC: 25.788
##
## Number of Fisher Scoring iterations: 4
```

```
# Parameter estimates
Res.glm.logit$coef
```

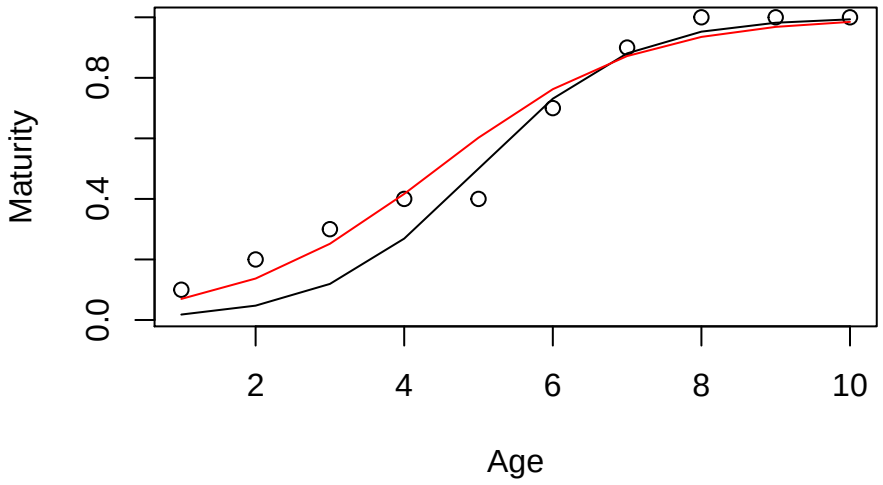
```
## (Intercept)         Age
##  -3.3440646   0.7517381
```

```
beta0.est <- Res.glm.logit$coef[1]
beta1.est <- Res.glm.logit$coef[2]

# CI of the paramters
confint(Res.glm.logit)
```

```
##                 2.5 %    97.5 %
## (Intercept) -4.907731 -2.065668
## Age          0.503086  1.066000
```

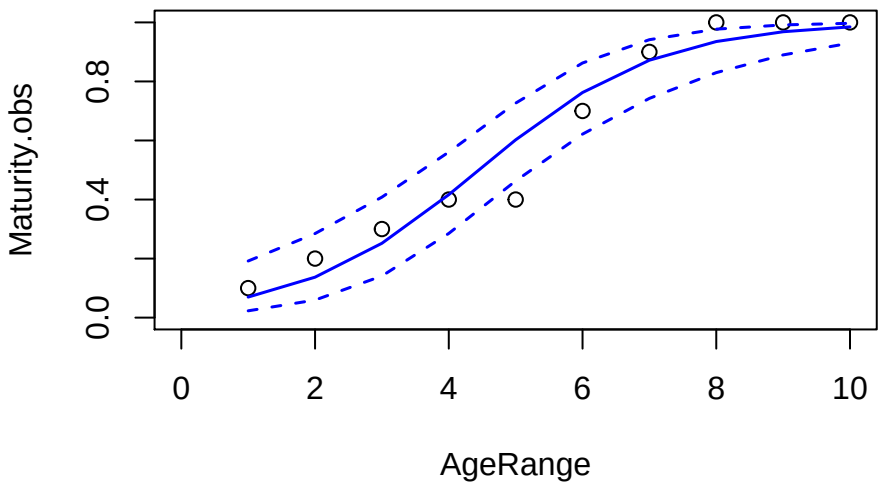
```
# Visual presentation of results
plot(Age, LogisticFn(Age,-5,1), type="l", ylab="Maturity")
```

```
points(Age, Maturity.obs)
points(Age, LogisticFn(Age,beta0.est,beta1.est), type="l", col="red")
```



3.4.2 Confidence interval for the curve

```
# Estimates
pred.se<-predict (Res.glm.logit, se.fit=TRUE)  #Linear predictor

#Linear predictor's CI
Lower <- pred.se$fit - qnorm(0.975, 0, 1)* pred.se$se.fit
Upper <- pred.se$fit + qnorm(0.975, 0, 1)* pred.se$se.fit

#Logistic transformation of CI
logit.t<-function(x){ 1/(1+exp(-x)) }
pred.logit<- sapply(pred.se$fit, logit.t)
Lower.logit <- sapply(Lower, logit.t)
Upper.logit<- sapply(Upper, logit.t)
pcon.logit<-cbind(pred.logit, Lower.logit, Upper.logit)
plot(AgeRange, Maturity.obs, xlim=c(0,10), ylim=c(0,1))
matlines(AgeRange, pcon.logit, lty=c(1,2,2), col="blue", lwd=1.5)
```



3.5 Confidence interval for your parmeter of interest (ASM)

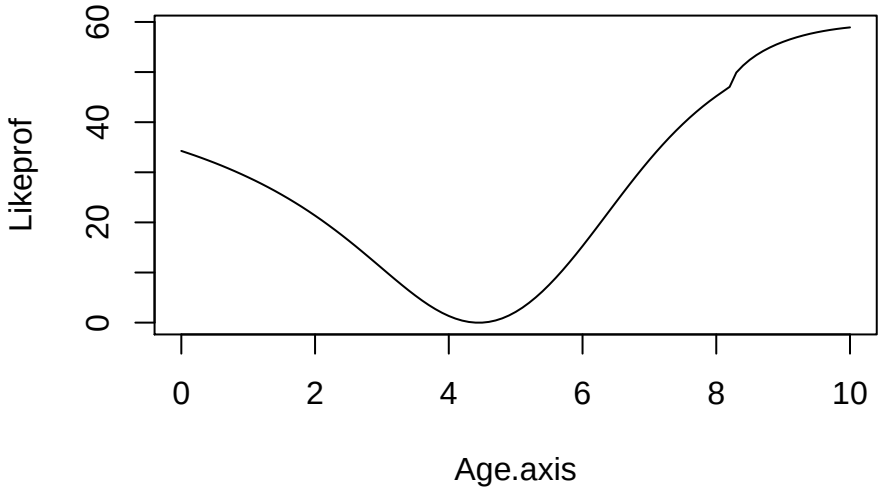
Method 1. Delta method (skipped)

Method 2. Profile likelihood

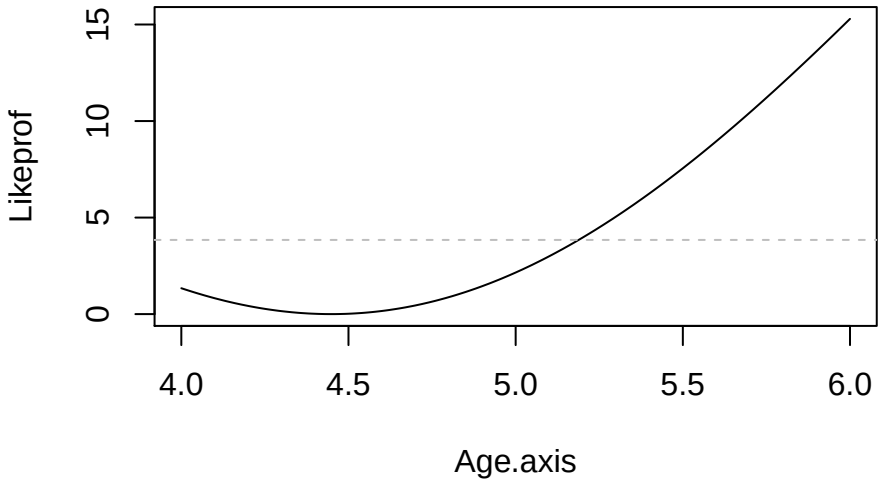
```
#Maturity function with respect to ASM50 and ASM95
MaturityFn <- function(a,ASM50,ASM95){
  beta0 <- -log(19)*ASM50/(ASM95-ASM50)
  beta1 <- log(19)/(ASM95-ASM50)
  LogisticFn(a,beta0,beta1)
}

ProfileASM50 <- function(ASM50)
{
 NLL <- function(ASM95)
 {
  pp <- MaturityFn(AgeRange,ASM50,ASM95)
  obj <- (-1.0)*sum(dbinom(Yvec,Nvec,pp,log=T))
  obj
 }
 tmp <- optim(11,NLL,method="BFGS")
 as.numeric((-1.0)*tmp$value)
}
```

Profile likelihood function:

```
Age.axis <- seq(0,10,0.1)
Res.prof <- unlist(lapply(Age.axis, ProfileASM50))
Likeprof <- -2*(Res.prof - max(Res.prof))
plot(Age.axis, Likeprof, type="l")
```




```
Age.axis <- seq(4,6,0.01)
Res.prof <- unlist(lapply(Age.axis, ProfileASM50))
Likeprof <- -2*(Res.prof - max(Res.prof))
plot(Age.axis, Likeprof, type="l")
abline(qchisq(0.95,1),0, lty=2, col="gray")
```



3.6 Statistical test

Assumption
- There is a traditional knowledge about ASM50 (e.g. H0: ASM50=6)
- You have to check if this knowldege is still valid or not
- So, you are now testing H0 against H1: ASM50 $\neq$ 6

Likelihood ratio test
- Generic method for statistical test
- Applicable in theory if likelihood functions for H0 and H1 are eplicitly written

In general, if H0 is relevant to 1-dimentional quantity (say H0: $\theta = \theta_0$),

$$T(\theta_0) = -2\log\frac{L(\theta_0)}{L(\hat{\theta})} \longrightarrow \chi^2(1) T \leq \chi^2(1, 1-\alpha) \Rightarrow \text{H0 is accepted}$$

```
ProfileASM50(6)
```

```
## [1] -18.53992
```

```
max(Res.prof)
```

```
## [1] -10.89384
```

```
Tstat <- -2*(ProfileASM50(6)-max(Res.prof))
Tstat
```

```
## [1] 15.29215
```

3.7 Model selection

AIC
- A sort of estimate of distance from the true model to estimated model
- c-AIC
- ...
- AIC = -2*maxloglikelihood* + *2*(#paramters)

```
Res.glm.logit <- glm(cbind(Yvec,Nvec-Yvec)~Age, family=binomial)
Res.glm.logit$aic
```

```
## [1] 25.78767
```

```
Res.glm.probit <- glm(cbind(Yvec,Nvec-Yvec)~Age, family=binomial(link=probit))
Res.glm.probit$aic
```

```
## [1] 25.13434
```

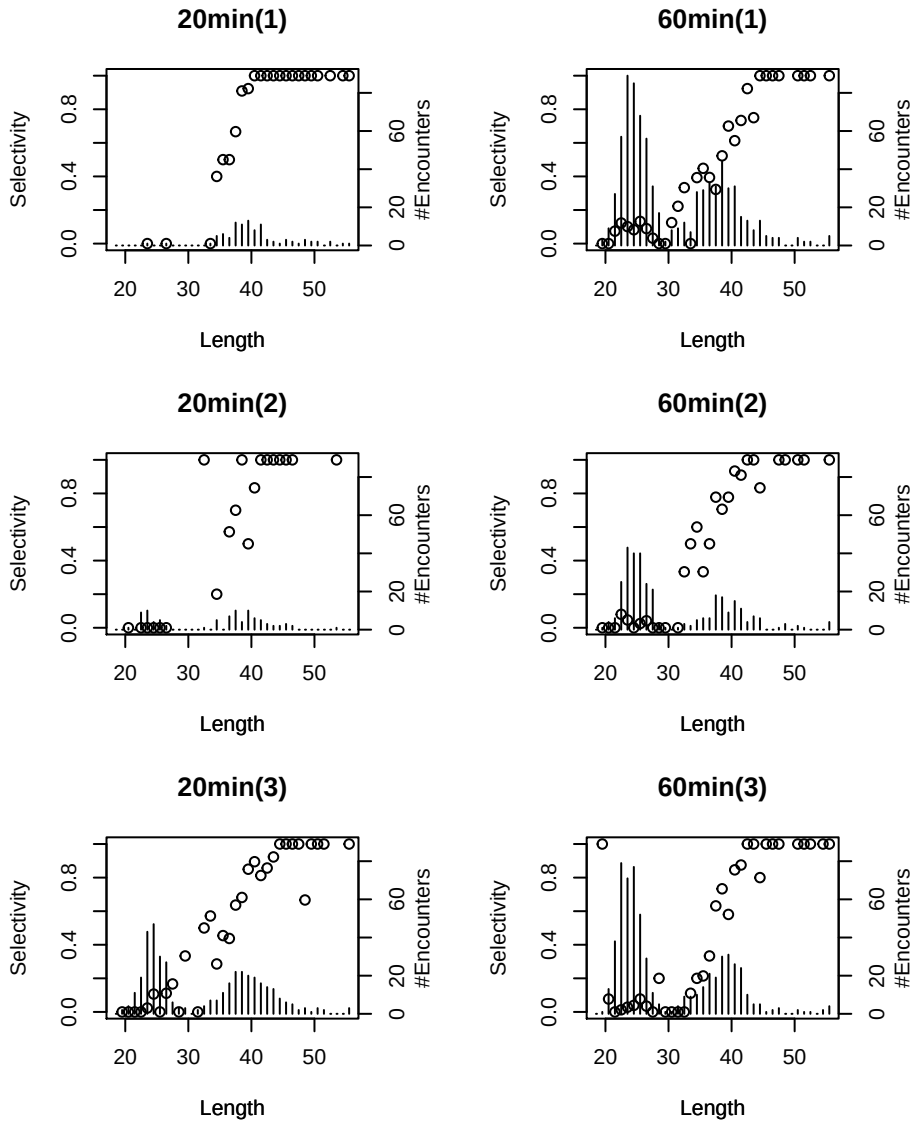
4 Generalized linear model (2) Gear selectivity

4.1 Data

We will use a data set from a covernet experiment.

```
Data <- read.csv("Data/Covernet.csv", header=T)
Length <- Data$Length
Codend <- Data[,seq(2,12,2)]
Cover <- Data[,seq(3,13,2)]
Encounter <- Codend + Cover

NN <- Codend + Cover
Selectivity <- Codend/Encounter
Duration <- c(20,20,20,60,60,60)
Repeat <- c(1,2,3,4,5,6)
Text <- c("20min(1)","20min(2)","20min(3)","60min(1)","60min(2)","60min(3)")
```

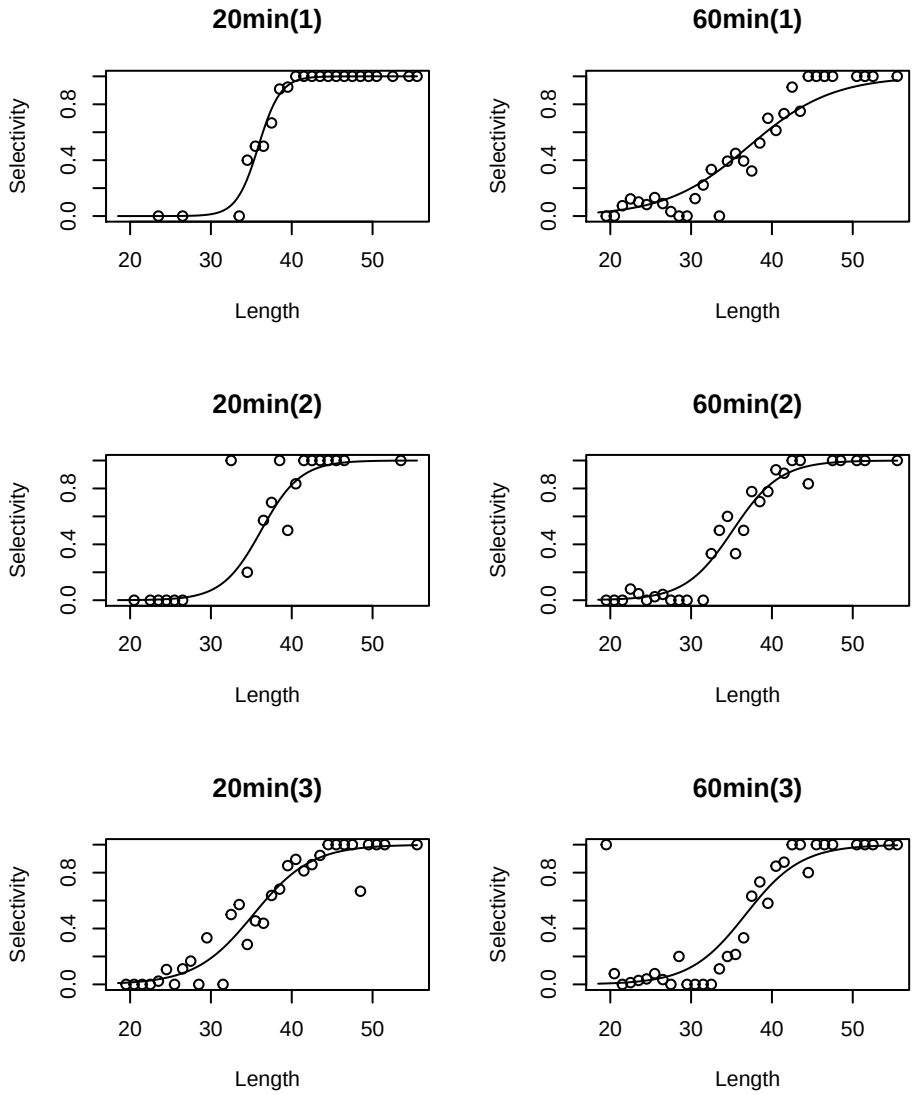
```
par(mfcol=c(3,2))
Ntrial <- 6
for(i in 1:Ntrial)
{
 par(mar=c(4,4,4,5))
 plot(Length, Selectivity[,i], ylim=c(0,1), ylab="Selectivity", main=Text[i])
 par(new=T)
 plot(Length, NN[,i], type="h", axes=F, ylab="", ylim=c(1,max(NN)))
 axis(4); mtext("#Encounters", side=4, line=2, cex=0.7)
}
```



4.2 GLM analysis by haul

```
Res.glm0 <- NULL
Para0 <- array(NA, c(Ntrial, 2))
colnames(Para0) <- c("a","b")
```

```
Retention <- function(x,para){ 1/(1+exp(-para[1]-para[2]*x)) }

par(mfcol=c(3,2))

for(i in 1:Ntrial){
 Res.glm0[[i]] <- glm(cbind(Codend[,i],Cover[,i])~Length, family=binomial)
 Para0[i,] <- Res.glm0[[i]]$coefficients
 plot(Length, Selectivity[,i], ylim=c(0,1), ylab="Selectivity", main=Text[i])
 curve(Retention(x,Para0[i,]), add=TRUE)
}
```




```
Para0
```

```
##               a         b
## [1,] -26.361227 0.7347294
## [2,] -14.946438 0.4126321
## [3,]  -9.819376 0.2785152
## [4,]  -7.204277 0.1938873
## [5,] -11.980199 0.3407501
```

```
## [6,] -10.661597 0.2901800
```

```
sum(as.numeric(lapply(Res.glm0, AIC)))
```

```
## [1] 385.9945
```

4.3 GLM analysis together (but same as above)

```
Codend.vec <- as.vector(as.matrix(Codend))
Cover.vec <- as.vector(as.matrix(Cover))
Length.vec <- rep(Length, Ntrial)
Repeat.vec <- factor(rep(Repeat, each=length(Length)))

CC <- cbind(Codend.vec,Cover.vec)
Res.glm1 <- glm(CC ~ -1+Repeat.vec+Length.vec*Repeat.vec, family=binomial)
Res.glm1
```

```
##
## Call:  glm(formula = CC ~ -1 + Repeat.vec + Length.vec * Repeat.vec,
##     family = binomial)
##
## Coefficients:
##            Repeat.vec1             Repeat.vec2             Repeat.vec3
##               -26.3612                -14.9464                 -9.8194
##            Repeat.vec4             Repeat.vec5             Repeat.vec6
##                -7.2043                -11.9802                -10.6616
##             Length.vec  Repeat.vec2:Length.vec  Repeat.vec3:Length.vec
##                 0.7347                 -0.3221                 -0.4562
## Repeat.vec4:Length.vec  Repeat.vec5:Length.vec  Repeat.vec6:Length.vec
##                -0.5408                 -0.3940                 -0.4445
##
## Degrees of Freedom: 173 Total (i.e. Null);  161 Residual
## Null Deviance:       1776
## Residual Deviance: 149   AIC: 386
```

4.4 GLM analysis more

```
Duration.vec <- factor(rep(Duration, each=length(Length)))

Res.glm2 <- glm(CC ~ -1+Duration.vec+Length.vec*Duration.vec, family=binomial)
Res.glm2
```

```
##
## Call:  glm(formula = CC ~ -1 + Duration.vec + Length.vec * Duration.vec,
##     family = binomial)
##
## Coefficients:
##            Duration.vec20             Duration.vec60
##                -10.86003                   -8.95278
##                Length.vec  Duration.vec60:Length.vec
##                   0.30982                   -0.06467
##
```

```
## Degrees of Freedom: 173 Total (i.e. Null);  169 Residual
## Null Deviance:       1776
## Residual Deviance: 189.8     AIC: 410.8
```

```
Res.glm3 <- glm(CC ~ -1+Duration.vec+Length.vec, family=binomial)
Res.glm3
```

```
##
## Call:  glm(formula = CC ~ -1 + Duration.vec + Length.vec, family = binomial)
##
## Coefficients:
## Duration.vec20  Duration.vec60      Length.vec
##        -9.0478         -9.4523          0.2596
##
## Degrees of Freedom: 173 Total (i.e. Null);  170 Residual
## Null Deviance:       1776
## Residual Deviance: 196.3     AIC: 415.3
```

```
Res.glm4 <- glm(CC ~ Length.vec*Duration.vec, family=binomial)
Res.glm4
```

```
##
## Call:  glm(formula = CC ~ Length.vec * Duration.vec, family = binomial)
##
## Coefficients:
##               (Intercept)                 Length.vec
##                  -10.86003                    0.30982
##            Duration.vec60  Length.vec:Duration.vec60
##                   1.90725                   -0.06467
##
## Degrees of Freedom: 172 Total (i.e. Null);  169 Residual
## Null Deviance:       1513
## Residual Deviance: 189.8     AIC: 410.8
```

```
Res.glm5 <- glm(CC ~ Length.vec, family=binomial)
Res.glm5
```

```
##
## Call:  glm(formula = CC ~ Length.vec, family = binomial)
##
## Coefficients:
## (Intercept)   Length.vec
##     -9.4615       0.2632
##
## Degrees of Freedom: 172 Total (i.e. Null);  171 Residual
## Null Deviance:       1513
## Residual Deviance: 204.7     AIC: 421.7
```

```
Res.glm6 <- glm(CC ~ -1+Repeat.vec+Length.vec*Duration.vec, family=binomial)
Res.glm6
```

```
##
## Call:  glm(formula = CC ~ -1 + Repeat.vec + Length.vec * Duration.vec,
```

```
##     family = binomial)
##
## Coefficients:
##               Repeat.vec1                Repeat.vec2
##                 -10.23583                  -10.87205
##               Repeat.vec3                Repeat.vec4
##                 -10.73100                   -8.99041
##               Repeat.vec5                Repeat.vec6
##                  -8.75276                   -9.09720
##                Length.vec             Duration.vec60
##                   0.30420                         NA
## Length.vec:Duration.vec60
##                  -0.05818
##
## Degrees of Freedom: 173 Total (i.e. Null);  165 Residual
## Null Deviance:       1776
## Residual Deviance: 184.3     AIC: 413.3
```

```
Res.glm7 <- glm(CC ~ -1+Duration.vec+Length.vec:Repeat.vec, family=binomial)
Res.glm7
```

```
##
## Call:  glm(formula = CC ~ -1 + Duration.vec + Length.vec:Repeat.vec,
##     family = binomial)
##
## Coefficients:
##         Duration.vec20          Duration.vec60  Length.vec:Repeat.vec1
##               -10.6943                 -8.9828                  0.3176
## Length.vec:Repeat.vec2  Length.vec:Repeat.vec3  Length.vec:Repeat.vec4
##                 0.3005                  0.3025                  0.2439
## Length.vec:Repeat.vec5  Length.vec:Repeat.vec6
##                 0.2555                  0.2441
##
## Degrees of Freedom: 173 Total (i.e. Null);  165 Residual
## Null Deviance:       1776
## Residual Deviance: 182.1     AIC: 411.1
```

```
lapply(list(Res.glm1,Res.glm2,Res.glm3,Res.glm4,Res.glm5,Res.glm6,Res.glm7), AIC)
```

```
## [[1]]
## [1] 385.9945
##
## [[2]]
## [1] 410.8121
##
## [[3]]
## [1] 415.3129
##
## [[4]]
## [1] 410.8121
##
## [[5]]
```

```
## [1] 421.7207
##
## [[6]]
## [1] 413.3131
##
## [[7]]
## [1] 411.0692
```

5 Generalized linear mixed-effect model (GLMM)

```
#install.packages("lme4")
library(lme4)
```

5.1 Maturity analysis revisied

Assumption
- You are analysing "age at sexual maturity (ASM)" of a fish species
- "Every year"", you observe the numbers of mature and immature individuals by age
- Your primary interest is if age at 50% maturity (ASM50) changes over years

Structural model
- Maturity is a funciton of age and y, $p(y,a)$
- $p(y,a)$ is a monotonically increasing function of a given year y

Logistic function

$$p(y,a) = \frac{e^{\beta_{y,0}+\beta_1 a}}{1+e^{\beta_{y,0}+\beta_1 a}} \implies \log \frac{p(y,a)}{1-p(y,a)} = \beta_{y,0} + \beta_1 a$$

5.2 Random intercept: Data generation with random year effect

Assumption
- Intercept β_0 randomly changes across years
- This implies that ASM50 also randomly changes accordingly

$$\frac{p(y,a)}{1-p(y,a)} = \beta_{y,0} + \beta_1 \beta_{y,0} \sim N(\mu, \sigma^2)$$

$ASM50_y = -\frac{\beta_{y,0}}{\beta_1}$

```
Nyear <- 15

Nmat <- array(10, c(Nyear,NAgeClass))
pmat <- array(NA, c(Nyear,NAgeClass))
Ymat <- array(NA, c(Nyear,NAgeClass))

mu <- (-1.0)*5
sigma <- 0.5

beta0.vec <- rnorm(Nyear, mu, sigma)
ASM.true <- -beta0.vec/1
ASM.true
```

```
##  [1] 5.183563 5.592128 5.251982 5.239722 5.668037 4.562885 3.983649 5.047555
##  [9] 5.770512 4.873124 4.686536 4.590178 5.160573 5.484169 5.204213
```

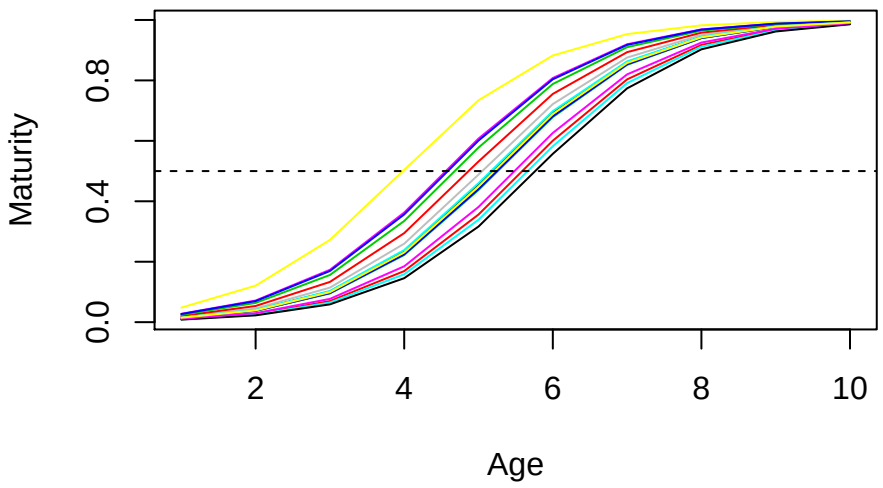
```
for(y in 1:Nyear){
  pmat[y,]<-LogisticFn(AgeRange, beta0.vec[y], 1)
  Ymat[y,]<-rbinom(NAgeClass, Nmat[y,], pmat[y,])
}

Data <- data.frame(
  Mature=as.vector(Ymat),
  Immature=as.vector(Nmat-Ymat),
  Age=rep(AgeRange,Nyear),
  Year=sort(rep(1:15,NAgeClass))
)
head(Data)
```

```
##   Mature Immature Age Year
## 1      0       10   1    1
## 2      0       10   2    1
## 3      0       10   3    1
## 4      0       10   4    1
## 5      0       10   5    1
## 6      0       10   6    1
```

```
tail(Data)
```

```
##     Mature Immature Age Year
## 145     10        0   5   15
## 146     10        0   6   15
## 147     10        0   7   15
## 148     10        0   8   15
## 149     10        0   9   15
## 150     10        0  10   15
```

```
plot(AgeRange, pmat[1,], xlab="Age", ylab="Maturity", type="l")
for(y in 1:Nyear) points(AgeRange, pmat[y,], type="l", col=y)
abline(0.5, 0, lty=2)
```



5.3 Estimation

```
#library(lme4)
Res.glmer1.logit <-
glmer(cbind(Mature, Immature) ~ Age + (1|Year), family=binomial, data=Data)
summary(Res.glmer1.logit)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(Mature, Immature) ~ Age + (1 | Year)
##    Data: Data
##
##      AIC      BIC   logLik deviance df.resid
##    428.7    437.8   -211.4    422.7      147
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2427 -0.6191  0.0751  0.5823  3.4327
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Year   (Intercept) 8.94     2.99
## Number of obs: 150, groups:  Year, 15
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.16811    0.79909   0.210    0.833
## Age          0.03122    0.02780   1.123    0.261
##
## Correlation of Fixed Effects:
```

```
##     (Intr)
## Age -0.191
```

```
Res.glmer2.logit <-
glmer(cbind(Mature, Immature) ~ 1 + (Age|Year), family=binomial, data=Data)
summary(Res.glmer2.logit)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(Mature, Immature) ~ 1 + (Age | Year)
##    Data: Data
##
##      AIC      BIC   logLik deviance df.resid
##    429.1    441.1   -210.5    421.1      146
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6129 -0.6269  0.0332  0.5826  3.5446
##
## Random effects:
##  Groups Name        Variance  Std.Dev. Corr
##  Year   (Intercept) 12.406593 3.52230
##         Age          0.008549 0.09246  -0.94
## Number of obs: 150, groups:  Year, 15
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.6043     0.6688   0.903    0.366
```

```
Res.glmer3.logit <-
glmer(cbind(Mature, Immature) ~  (1+Age|Year), family=binomial, data=Data)
summary(Res.glmer3.logit)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(Mature, Immature) ~ (1 + Age | Year)
##    Data: Data
##
##      AIC      BIC   logLik deviance df.resid
##    429.1    441.1   -210.5    421.1      146
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6129 -0.6269  0.0332  0.5826  3.5446
##
## Random effects:
##  Groups Name        Variance  Std.Dev. Corr
##  Year   (Intercept) 12.406593 3.52230
##         Age          0.008549 0.09246  -0.94
## Number of obs: 150, groups:  Year, 15
```

```
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.6043     0.6688   0.903    0.366
```

6 Nonlinear regression: growth formula

6.1 Reading and visualization of the data

```
Data <- read.csv("Data/Alfonsino.csv")
head(Data)
```

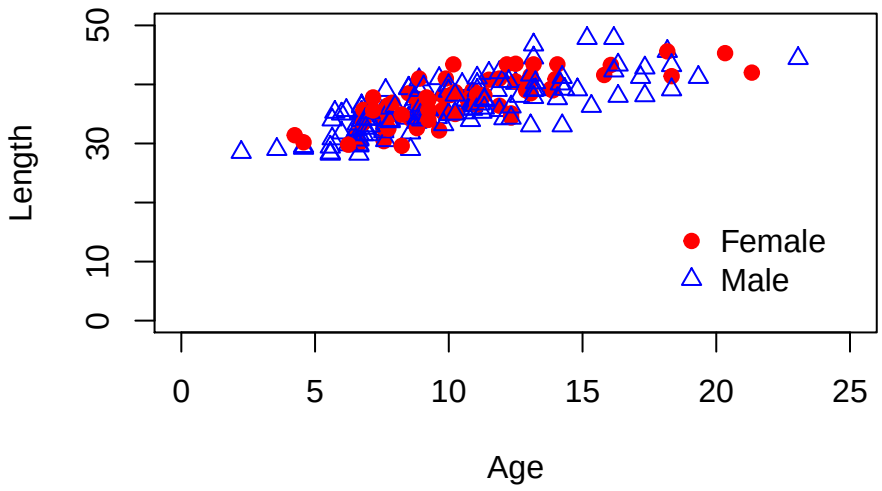
```
##    Age Length Sex
## 1 10.5   40.0   M
## 2 12.5   40.2   M
## 3 12.5   40.5   F
## 4  8.5   34.3   M
## 5  6.5   31.4   F
## 6  8.5   39.4   M
```

```
names(Data)
```

```
## [1] "Age"    "Length" "Sex"
```

```
Age <- Data$Age
Length <- Data$Length
Sex <- as.numeric(Data$Sex)
```

```
mark <- c(19, 2)
color <- c("red","blue")
plot(Length~Age, pch=mark[Sex],
     col=color[Sex], xlim=c(0,25),
     ylim=c(0,50))
legend(18,20,pch=mark,col=color,
       legend=c("Female","Male"), bty="n")
```



6.2 Model

We will employ a famous growth curve, which the von Bertalanffy growth curve defined as

$$L_t = L_\infty(1 - exp(-K(t - t0)))$$

```
growth.VB <- function(t, Linf, K, t0){
  Linf*(1-exp(-K*(t-t0)))
}
```

6.3 Estimation of parameters under an assumption of common parameters between female and male

```
start <- list(Linf=max(Length), K=0.1, t0=0)

res.000 <- nls(Length~growth.VB(t=Age, Linf, K, t0), start=start)

summary(res.000)
```

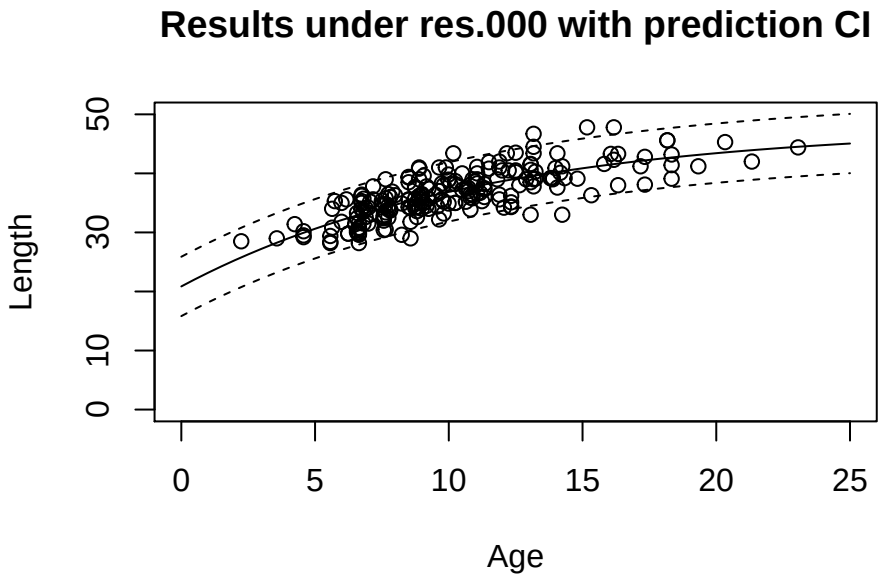
```
##
## Formula: Length ~ growth.VB(t = Age, Linf, K, t0)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Linf 47.95842    2.91479  16.453  < 2e-16 ***
## K     0.08936    0.02413   3.704  0.00027 ***
## t0   -6.38637    2.14489  -2.977  0.00324 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.564 on 215 degrees of freedom
##
```

```
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 5.63e-06
```

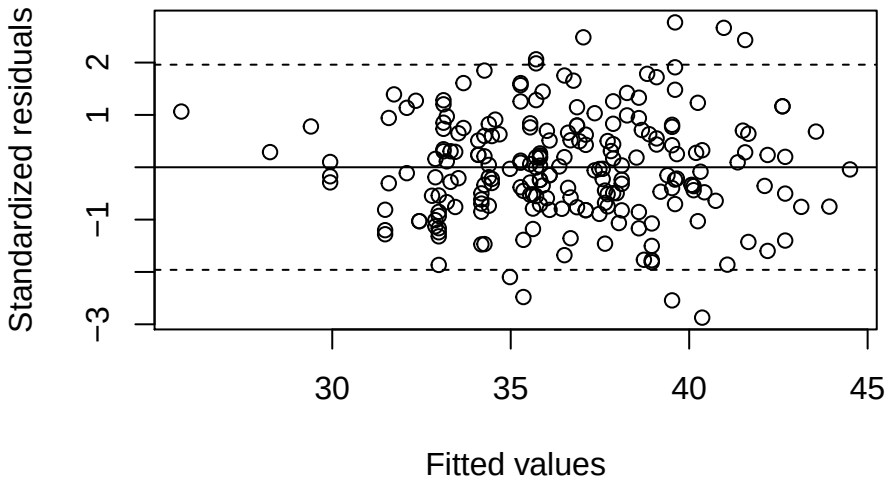
```
confint(res.000)
```

```
##              2.5%      97.5%
## Linf  44.10249477 58.3766050
## K      0.04465578  0.1360594
## t0   -12.34296242 -3.3029532
```

```
# Visual presentation
taxis <- seq(0,25,0.5); tlen <- length(taxis);
pred.000 <- predict(res.000, list(Age=taxis))
plot(Length~Age,xlim=c(0,25),ylim=c(0,50), main="Results under res.000 with prediction CI")
points(taxis, pred.000, type="l")
points(taxis, pred.000-1.96*sigma(res.000), type="l", lty=2)
points(taxis, pred.000+1.96*sigma(res.000), type="l", lty=2)
```




```
# Diagnostics
e <- residuals(res.000)/sigma(res.000)
plot(fitted(res.000), e, xlab="Fitted values", ylab="Standardized residuals")
abline(0,0); abline(-1.96,0,lty=2); abline(1.96,0,lty=2);
```



6.4 Estimation of parameters under an assumption of different parameters between female and male

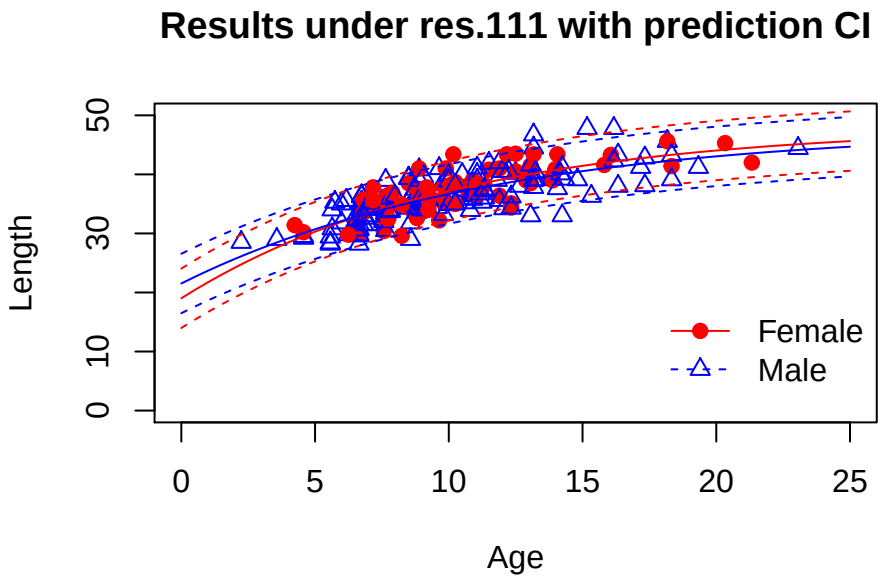
```
# Setting a set of initial values for female and male simultaneously
start <- list(Linf=rep(max(Length),2),K=c(0.1,0.1),t0=c(0,0))

# Estimation
res.111 <- nls(Length~growth.VB(t=Age,Linf[Sex], K[Sex], t0[Sex]), start=start)

# Showing the estimation summary
summary(res.111)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf[Sex], K[Sex], t0[Sex])
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## Linf1 48.15050    4.52391  10.644  < 2e-16 ***
## Linf2 47.70053    3.75965  12.687  < 2e-16 ***
## K1     0.09797    0.04315   2.270  0.02418 *
## K2     0.08646    0.02995   2.887  0.00429 **
## t01   -5.12851    3.38334  -1.516  0.13106
## t02   -6.94091    2.80413  -2.475  0.01410 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.567 on 212 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 3.178e-06
```

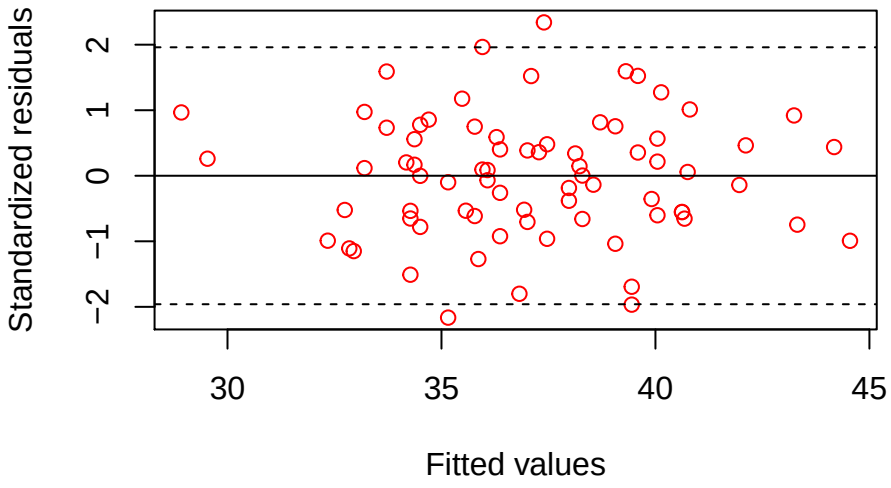
```
# Confidence interval for the growth paramters
confint(res.111)
```

```
##               2.5%      97.5%
## Linf1  43.09967561 81.7105984
## Linf2  43.18112305 66.9658334
## K1      0.02164258  0.1833475
## K2      0.03031102  0.1453427
## t01   -18.02360053 -0.9526554
## t02   -16.13978541 -3.1310768
```

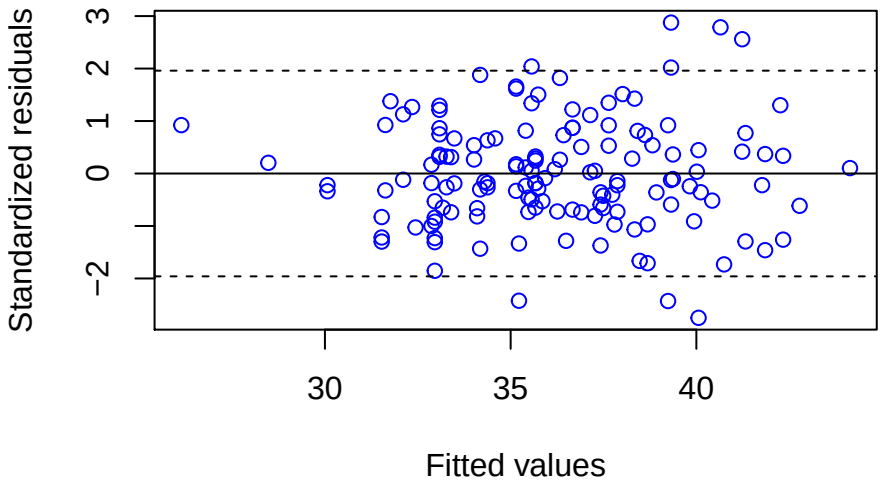
```
# Visual presentation
taxis <- seq(0,25,0.5); tlen <- length(taxis);
pred.female.111 <- predict(res.111, list(Age=taxis, Sex=rep(1,tlen)))
pred.male.111 <- predict(res.111, list(Age=taxis, Sex=rep(2,tlen)) )

plot(Length~Age,pch=mark[Sex],col=color[Sex],xlim=c(0,25),ylim=c(0,50),
     main="Results under res.111 with prediction CI")
points(taxis, pred.female.111, type="l", col="red", lty=1)
points(taxis, pred.female.111-1.96*sigma(res.111), type="l", col="red", lty=2)
points(taxis, pred.female.111+1.96*sigma(res.111), type="l", col="red", lty=2)
points(taxis, pred.male.111, type="l", col="blue", lty=1)
points(taxis, pred.male.111-1.96*sigma(res.111), type="l", col="blue", lty=2)
points(taxis, pred.male.111+1.96*sigma(res.111), type="l", col="blue", lty=2)
legend(18,20,pch=mark,col=color,lty=c(1,2),legend=c("Female","Male"),bty="n")
```




```
# Diagnostics for female
e <- residuals(res.111)/sigma(res.111)
plot(fitted(res.111)[Sex==1], e[Sex==1], col="red",
     xlab="Fitted values", ylab="Standardized residuals")
abline(0,0); abline(-1.96,0,lty=2); abline(1.96,0,lty=2);
```




```
# Diagnostics for male
plot(fitted(res.111)[Sex==2], e[Sex==2], col="blue",
     xlab="Fitted values", ylab="Standardized residuals")
abline(0,0); abline(-1.96,0,lty=2); abline(1.96,0,lty=2)
```



6.5 Other models

6.5.1 Model with res.100

```
start <- list(Linf=rep(max(Length),2), K=0.1, t0=0)
res.100 <- nls(Length~growth.VB(t=Age,Linf[Sex], K, t0), start=start)
summary(res.100)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf[Sex], K, t0)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## Linf1 48.54689    3.08114  15.756  < 2e-16 ***
## Linf2 47.90648    3.00673  15.933  < 2e-16 ***
## K      0.08754    0.02420   3.618 0.000371 ***
## t0    -6.58214    2.21800  -2.968 0.003343 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.559 on 214 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 7.413e-06
```

6.5.2 Model with res.010

```
start <- list(Linf=max(Length), K=c(0.1, 0.1), t0=0)
res.010 <- nls(Length~growth.VB(t=Age,Linf, K[Sex], t0), start=start)
summary(res.010)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf, K[Sex], t0)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Linf 48.32799    3.11385  15.520  < 2e-16 ***
## K1    0.08744    0.02459   3.555 0.000465 ***
## K2    0.08513    0.02378   3.580 0.000426 ***
## t0   -6.73140    2.25591  -2.984 0.003177 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.561 on 214 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 5.365e-06
```

6.5.3 Model with res.001

```
start <- list(Linf=max(Length), K=0.1, t0=c(0,0))
res.001 <- nls(Length~growth.VB(t=Age,Linf, K, t0[Sex]), start=start)
summary(res.001)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf, K, t0[Sex])
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
```

```
## Linf 48.39942    3.15358  15.347  < 2e-16 ***
## K     0.08529    0.02402   3.551 0.000473 ***
## t01  -7.03254    2.32792  -3.021 0.002827 **
## t02  -6.67567    2.25953  -2.954 0.003483 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.564 on 214 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 1.406e-06
```

6.5.4 Model with res.110

```
start <- list(Linf=rep(max(Length),2), K=rep(0.1,2), t0=0)
res.110 <- nls(Length~growth.VB(t=Age,Linf[Sex], K[Sex], t0), start=start)
summary(res.110)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf[Sex], K[Sex], t0)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## Linf1 49.66398    3.71760  13.359  < 2e-16 ***
## Linf2 46.99869    2.88861  16.270  < 2e-16 ***
## K1     0.08431    0.02384   3.536 0.000498 ***
## K2     0.09291    0.02562   3.627 0.000359 ***
## t0    -6.34966    2.16198  -2.937 0.003679 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.562 on 213 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 3.422e-06
```

6.5.5 Model with res.011

```
start <- list(Linf=max(Length), K=rep(0.1,2), t0=rep(0,2))
res.011 <- nls(Length~growth.VB(t=Age,Linf, K[Sex], t0[Sex]), start=start)
summary(res.011)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf, K[Sex], t0[Sex])
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Linf 47.90369    2.88967  16.578  < 2e-16 ***
## K1    0.10029    0.02990   3.354 0.000942 ***
## K2    0.08491    0.02261   3.755 0.000224 ***
## t01  -4.96476    2.45415  -2.023 0.044323 *
```

```
## t02  -7.07769    2.26390  -3.126 0.002017 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.561 on 213 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 4.91e-06
```

6.5.6 Model with res.101

```
start <- list(Linf=rep(max(Length),2), K=0.1, t0=rep(0,2))
res.101 <- nls(Length~growth.VB(t=Age,Linf[Sex], K, t0[Sex]), start=start)
summary(res.101)
```

```
##
## Formula: Length ~ growth.VB(t = Age, Linf[Sex], K, t0[Sex])
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## Linf1 48.97747    3.13603  15.618  < 2e-16 ***
## Linf2 47.23314    2.86083  16.510  < 2e-16 ***
## K      0.09044    0.02452   3.688 0.000287 ***
## t01   -5.73684    2.22748  -2.575 0.010687 *
## t02   -6.58757    2.19411  -3.002 0.002999 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.561 on 213 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 7.627e-06
```

6.5.7 Comparison of models

```
AIC(res.000)
```

```
## [1] 1034.122
```

```
AIC(res.100)
```

```
## [1] 1034.308
```

```
AIC(res.010)
```

```
## [1] 1034.706
```

```
AIC(res.001)
```

```
## [1] 1035.137
```

```
AIC(res.110)
```

```
## [1] 1035.71
```

```
AIC(res.101)
```

```
## [1] 1035.587
```

```
AIC(res.011)
```

```
## [1] 1035.538
```

```
AIC(res.111)
```

```
## [1] 1037.531
```

6.5.8 Conclusion

Based on the comparison of AIC values, the best model is "res.000", which means that there are no sex-difference in the growth of Alfonsino.

7 Generalized linear model GLM: PCB data

7.1 Data

We will use lake trout data with PCB contamination shown in Qian (2016).

```
pcb.data <- read.csv("Data/laketrout2.csv", header=T)[-130,] #remove length=0
pcb.data$lenclass <- cut(pcb.data$length, breaks=c(0,20,25,30,50))
pcb.data$period <- cut(pcb.data$year-1900, breaks=c(73,80,90,100))
pcb.data$loglen <- log(pcb.data$length)

kable(rbind(head(pcb.data), tail(pcb.data)))
```

	length	pcb	n	lgpcb	year	y	lenclass	period	loglen
1	29.9	31.3	1	3.443618	1974	0	(25,30]	(73,80]	3.397858
2	29.5	7.9	1	2.066863	1974	0	(25,30]	(73,80]	3.384390
3	27.0	26.7	1	3.284664	1974	0	(25,30]	(73,80]	3.295837
4	26.3	8.3	1	2.116255	1974	0	(25,30]	(73,80]	3.269569
5	27.4	11.3	1	2.424803	1974	0	(25,30]	(73,80]	3.310543
6	27.2	12.6	1	2.533697	1974	0	(25,30]	(73,80]	3.303217
643	29.0	5.4	1	1.686399	2000	26	(25,30]	(90,100]	3.367296
644	30.0	3.6	1	1.280934	2000	26	(25,30]	(90,100]	3.401197
645	30.6	4.1	1	1.410987	2000	26	(30,50]	(90,100]	3.421000
646	31.8	7.1	1	1.960095	2000	26	(30,50]	(90,100]	3.459466
647	26.0	3.0	1	1.098612	2000	26	(25,30]	(90,100]	3.258097
648	30.8	4.2	1	1.435085	1998	24	(30,50]	(90,100]	3.427515

```
str(pcb.data)
```

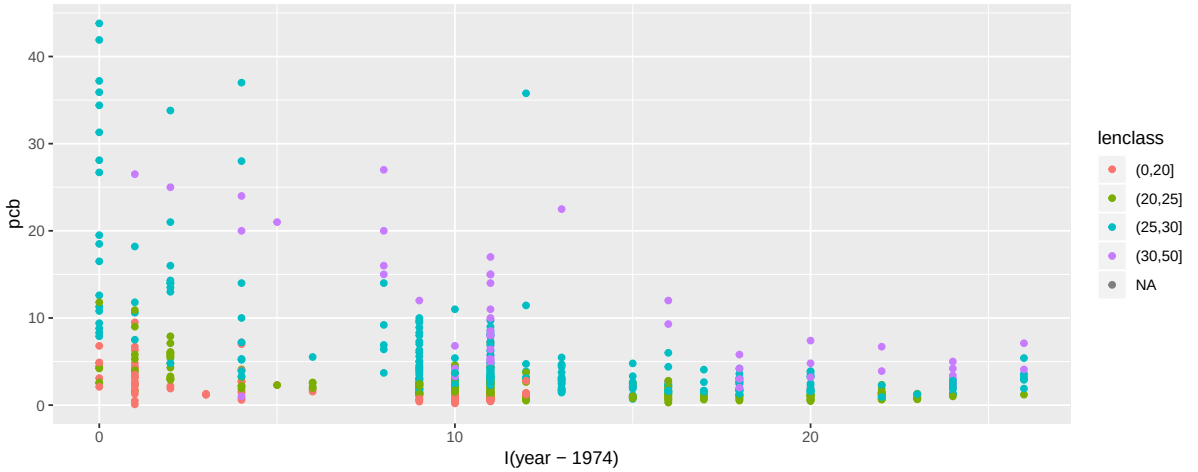
```
## 'data.frame':    647 obs. of  9 variables:
##  $ length  : num  29.9 29.5 27 26.3 27.4 27.2 26.2 26.6 20.8 19.3 ...
##  $ pcb     : num  31.3 7.9 26.7 8.3 11.3 12.6 8.8 10.8 4.4 4.8 ...
##  $ n       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ lgpcb   : num  3.44 2.07 3.28 2.12 2.42 ...
##  $ year    : int  1974 1974 1974 1974 1974 1974 1974 1974 1974 1974 ...
##  $ y       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ lenclass: Factor w/ 4 levels "(0,20]","(20,25]",..: 3 3 3 3 3 3 3 3 2 1 ...
##  $ period  : Factor w/ 3 levels "(73,80]","(80,90]",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ loglen  : num  3.4 3.38 3.3 3.27 3.31 ...
```

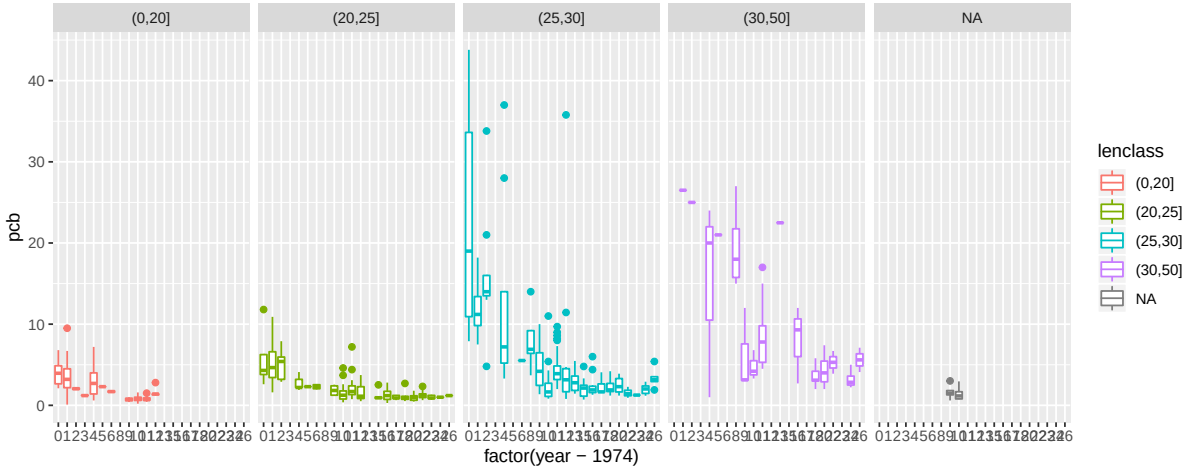
Questions are
- the PCB level in the trout has been monotonically increasing or decreasing or has a nonliner pattern ?
- the PCB level in the trout is proportional to the body? - Or any other factors impacting the pcb level?

7.2 Visual presentation of data

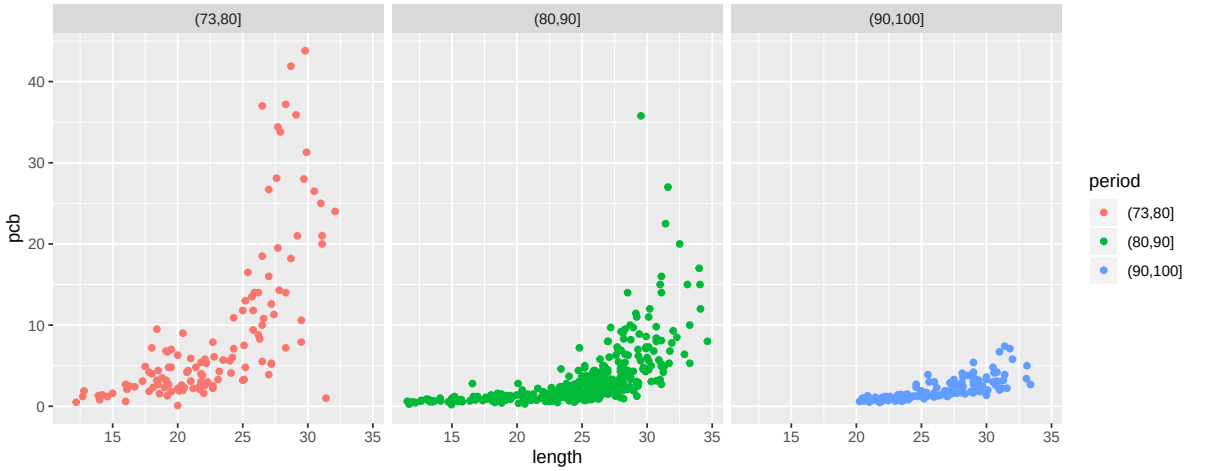
```
ggplot(pcb.data, aes(x=I(year-1974), y=pcb, color=lenclass)) + geom_point()
```



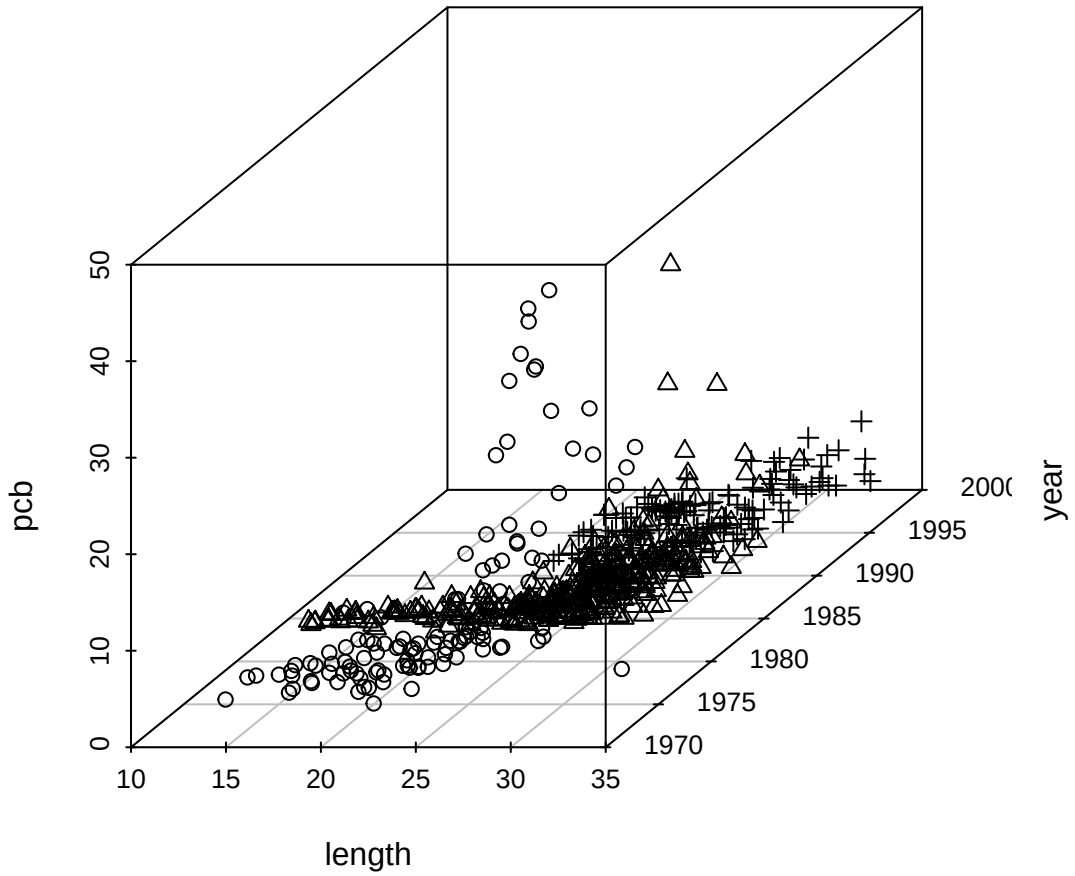

```
ggplot(pcb.data, aes(x=factor(year-1974), y=pcb, color=lenclass)) +
  geom_boxplot() + facet_wrap(lenclass~., ncol=5)
```




```
ggplot(pcb.data, aes(x=length, y=pcb, color=period)) +  geom_point() +
  facet_wrap(period~., ncol=3)
```




```
#install.packages("scatterplot3d")
library(scatterplot3d)
scatterplot3d(pcb.data[, c(1,5,2)], pch=as.numeric(pcb.data$period))
```



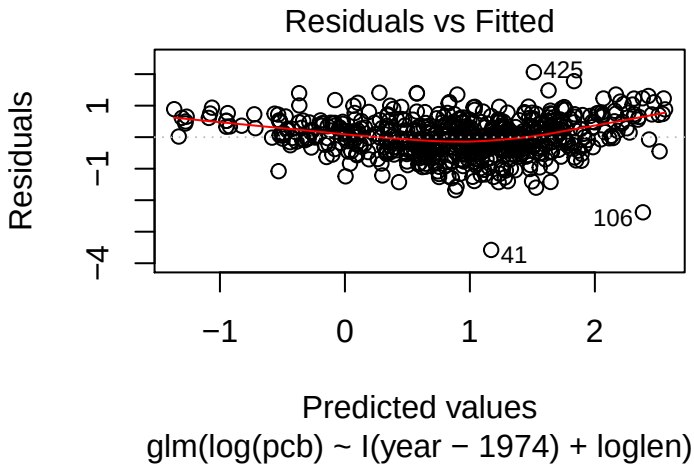
```
#install.packages("rgl")
library(rgl)
#knit_hooks$set(webgl = hook_webgl)
plot3d(pcb.data[,c("length", "year", "pcb")],
       col = rainbow(3)[factor(pcb.data$lenclass)])
```

7.3 Application linear models

```
res.pcb.lm1 <- res <- glm(log(pcb) ~ I(year-1974) + loglen, data=pcb.data)
summary(res.pcb.lm1)
```

```
##
## Call:
## glm(formula = log(pcb) ~ I(year - 1974) + loglen, data = pcb.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5783  -0.3497  -0.0060   0.3612   2.0653
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -8.485177   0.364087  -23.30   <2e-16 ***
## I(year - 1974) -0.084078   0.003858  -21.79   <2e-16 ***
## loglen          3.251149   0.116617   27.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3561125)
##
##     Null deviance: 587.01  on 631  degrees of freedom
## Residual deviance: 223.99  on 629  degrees of freedom
##   (15 observations deleted due to missingness)
## AIC: 1146
##
## Number of Fisher Scoring iterations: 2
```

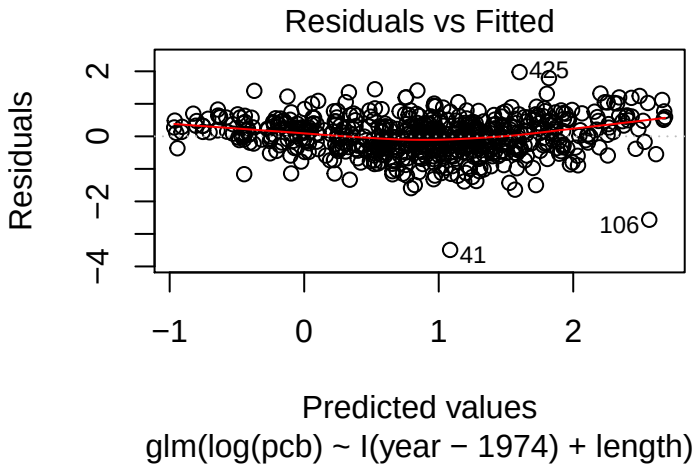
```
plot(res.pcb.lm1,1)
```




```
aic.lm1 <- AIC(res.pcb.lm1)
dev.lm1 <- (res$null.deviance-res$deviance)/res$null.deviance

res.pcb.lm2 <- res <- glm(log(pcb) ~ I(year-1974) + length, data=pcb.data)
summary(res)
```

```
##
## Call:
## glm(formula = log(pcb) ~ I(year - 1974) + length, data = pcb.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4927  -0.3309   0.0172   0.3538   1.9764
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.874012   0.123834  -15.13   <2e-16 ***
## I(year - 1974) -0.084882   0.003694  -22.98   <2e-16 ***
## length          0.152184   0.005060   30.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3265182)
##
##     Null deviance: 587.01  on 631  degrees of freedom
## Residual deviance: 205.38  on 629  degrees of freedom
##   (15 observations deleted due to missingness)
## AIC: 1091.2
##
## Number of Fisher Scoring iterations: 2
```

```
plot(res,1)
```

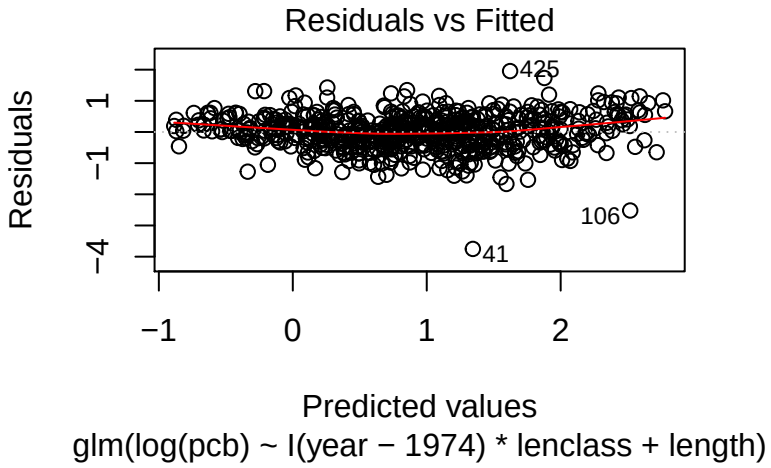



```
aic.lm2 <- AIC(res)
dev.lm2 <- (res$null.deviance-res$deviance)/res$null.deviance

res.pcb.lm3 <- res <- glm(log(pcb) ~ I(year-1974)*lenclass + length, data=pcb.data)
summary(res)
```

```
##
## Call:
## glm(formula = log(pcb) ~ I(year - 1974) * lenclass + length,
##     data = pcb.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7524  -0.3274   0.0418   0.3464   1.9549
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -1.705250   0.276630  -6.164 1.27e-09 ***
## I(year - 1974)                 -0.098617   0.013256  -7.439 3.37e-13 ***
## lenclass(20,25]                -0.581390   0.154659  -3.759 0.000187 ***
## lenclass(25,30]                -0.222001   0.189711  -1.170 0.242364
## lenclass(30,50]                -0.447858   0.278241  -1.610 0.107991
## length                          0.157417   0.014332  10.984  < 2e-16 ***
## I(year - 1974):lenclass(20,25]  0.027242   0.014651   1.859 0.063446 .
## I(year - 1974):lenclass(25,30]  0.007083   0.014407   0.492 0.623131
## I(year - 1974):lenclass(30,50]  0.030978   0.017184   1.803 0.071909 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3136239)
##
##     Null deviance: 587.01  on 631  degrees of freedom
## Residual deviance: 195.39  on 623  degrees of freedom
##   (15 observations deleted due to missingness)
```

```
## AIC: 1071.6
##
## Number of Fisher Scoring iterations: 2
```

```
plot(res,1)
```




```
aic.lm3 <- AIC(res)
dev.lm3 <- (res$null.deviance-res$deviance)/res$null.deviance

table.lm <- data.frame(
  Model=paste("LM",1:3),
  AIC=c(aic.lm1,aic.lm2,aic.lm3),
  Deviance.explained=c(dev.lm1,dev.lm2,dev.lm3)
)
kable(table.lm)
```

Model	AIC	Deviance.explained
LM 1	1145.986	0.6184132
LM 2	1091.153	0.6501246
LM 3	1071.631	0.6671468

7.4 Application of additive models

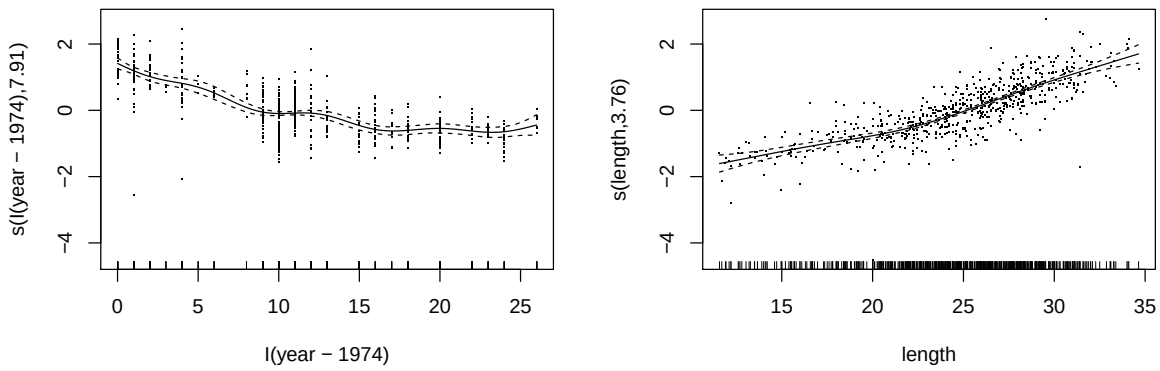
We will use the GAM but with a normal distribution. In this we shall call just "additive model".

```
res.pcb.gam1 <- res <- gam(log(pcb)~s(I(year-1974))+s(length), data=pcb.data)
summary(res)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcb) ~ s(I(year - 1974)) + s(length)
##
## Parametric coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.90542    0.02019   44.84   <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(I(year - 1974)) 7.91  8.685  88.6  <2e-16 ***
## s(length)         3.76  4.694 243.2  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.723   Deviance explained = 72.8%
## GCV = 0.2629  Scale est. = 0.25763   n = 632
```

```
par(mfrow=c(1,2))
plot(res,1)
```




```
aic.gam1 <- AIC(res)
dev.gam1 <- summary(res)$dev.expl

res.pcb.gam2 <- res <-
  gam(log(pcb)~s(I(year-1974),by=lenclass)+s(length), data=pcb.data)
summary(res)
```

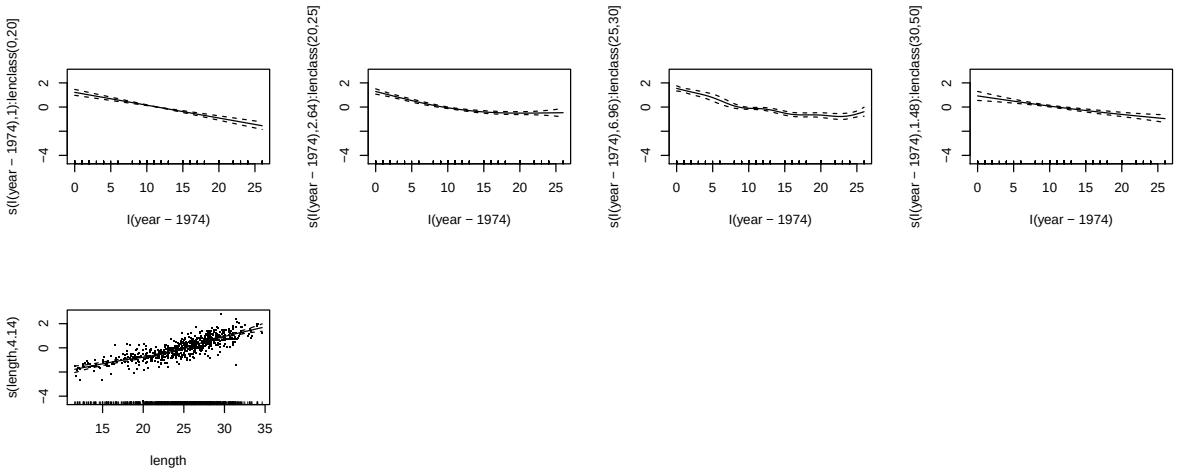
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcb) ~ s(I(year - 1974), by = lenclass) + s(length)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.88917    0.02176   40.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Approximate significance of smooth terms:
##                                    edf Ref.df      F  p-value
## s(I(year - 1974)):lenclass(0,20]  1.000  1.000  99.13  < 2e-16 ***
## s(I(year - 1974)):lenclass(20,25] 2.637  3.228  64.69  < 2e-16 ***
## s(I(year - 1974)):lenclass(25,30] 6.958  7.998  52.81  < 2e-16 ***
## s(I(year - 1974)):lenclass(30,50] 1.481  1.815  27.37 5.59e-11 ***
## s(length)                         4.145  5.159 202.59  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.726   Deviance explained = 73.3%
## GCV = 0.26248  Scale est. = 0.25533   n = 632
```

```
par(mfrow=c(2,4))
plot(res,1)
aic.gam2 <- AIC(res)
dev.gam2 <- summary(res)$dev.expl

res.pcb.gam3 <- res <-
  gam(log(pcb)~s(I(year-1974))+s(length,by=period), data=pcb.data)
summary(res)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcb) ~ s(I(year - 1974)) + s(length, by = period)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.90668    0.02084   43.51   <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(I(year - 1974))          7.786  8.610  82.58  <2e-16 ***
## s(length):period(73,80]    5.000  6.058  50.58  <2e-16 ***
## s(length):period(80,90]    3.231  4.054 189.91  <2e-16 ***
## s(length):period(90,100] 1.000  1.000 119.95  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.733   Deviance explained =   74%
## GCV = 0.25575  Scale est. = 0.24846   n = 632
```

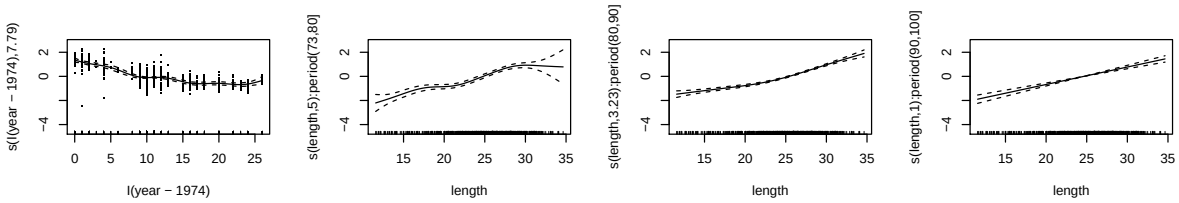
```
par(mfrow=c(2,4))
```




```
plot(res,1)
aic.gam3 <- AIC(res)
dev.gam3 <- summary(res.pcb.gam3)$dev.expl

res.pcb.gam4 <- res <-
  gam(log(pcb)~s(I(year-1974),by=lenclass)+s(length,by=period), data=pcb.data)
summary(res)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcb) ~ s(I(year - 1974), by = lenclass) + s(length, by = period)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.89372    0.02819    31.7   <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(I(year - 1974)):lenclass(0,20]   3.628  4.223  2.295  0.05277 .
## s(I(year - 1974)):lenclass(20,25] 2.403  2.978 34.000  < 2e-16 ***
## s(I(year - 1974)):lenclass(25,30] 7.373  8.316 19.388  < 2e-16 ***
## s(I(year - 1974)):lenclass(30,50] 2.680  3.333  6.380  0.00019 ***
## s(length):period(73,80]           6.653  7.670  7.345 4.94e-09 ***
## s(length):period(80,90]           2.957  3.738 77.363  < 2e-16 ***
## s(length):period(90,100]          1.000  1.000 36.340 2.82e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.733   Deviance explained = 74.4%
## GCV = 0.25985  Scale est. = 0.24846   n = 632
```

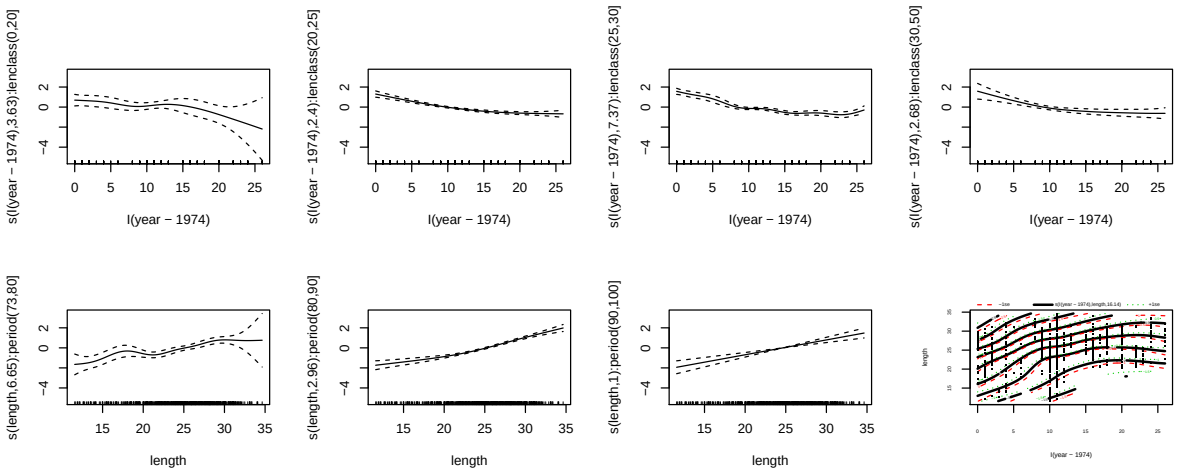
```
par(mfrow=c(2,4))
```




```
plot(res,1)
aic.gam4 <- AIC(res)
dev.gam4 <- summary(res)$dev.expl

res.pcb.gam5 <- res <- gam(log(pcb)~s(I(year-1974),length), data=pcb.data)
summary(res)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcb) ~ s(I(year - 1974), length)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.90542    0.02007   45.11   <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(I(year - 1974),length) 16.14  21.03 79.59  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.726   Deviance explained = 73.3%
## GCV = 0.26166  Scale est. = 0.25456   n = 632
```

```
plot(res,1)
```



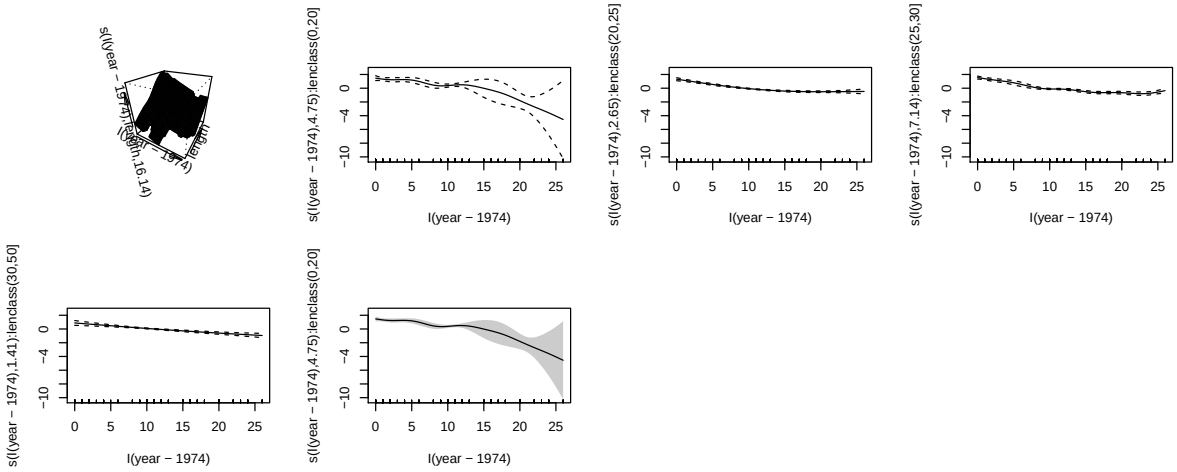

```
plot(res,select=1,se=T,rug=T,resid=T,scheme=T)
aic.gam5 <- AIC(res)
dev.gam5 <- summary(res)$dev.expl

res.pcb.gam6 <- res <- gam(log(pcb)~s(I(year-1974),by=lenclass)+length, data=pcb.data)
summary(res)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcb) ~ s(I(year - 1974), by = lenclass) + length
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.500537   0.178547  -19.61   <2e-16 ***
## length       0.176013   0.006809   25.85   <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(I(year - 1974)):lenclass(0,20]  4.751  5.291 31.23  < 2e-16 ***
## s(I(year - 1974)):lenclass(20,25] 2.655  3.250 66.87  < 2e-16 ***
## s(I(year - 1974)):lenclass(25,30] 7.145  8.150 51.15  < 2e-16 ***
## s(I(year - 1974)):lenclass(30,50] 1.412  1.715 28.25 1.09e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.722   Deviance explained =   73%
## GCV = 0.26583  Scale est. = 0.25827   n = 632
```

```
plot(res,1)
plot(res,select=1,se=T,rug=T,resid=T,scheme=T)
aic.gam6 <- AIC(res)
dev.gam6 <- summary(res)$dev.expl
```

```
table.gam <- data.frame(
  Model=paste("GAM",1:6),
  AIC=c(aic.gam1,aic.gam2,aic.gam3,aic.gam4,aic.gam5,aic.gam6),
  Deviance.explained=c(dev.gam1,dev.gam2,dev.gam3,dev.gam4,dev.gam5,dev.gam6)
)
kable(table.gam)
```

Model	AIC	Deviance.explained
GAM 1	950.9369	0.7281867
GAM 2	949.7123	0.7325922
GAM 3	933.2561	0.7401208
GAM 4	942.5629	0.7442189
GAM 5	947.7287	0.7333636
GAM 6	957.6747	0.7298356




```
#install.packages("mgcViz",dependencies=T)
library(mgcViz)
res.pcb.gam5 <- getViz(res.pcb.gam5)
plotRGL(sm(res.pcb.gam5,1), residuals = TRUE)
```

8 Generalized adddtive model: (very) simple spatial mapping

8.1 Visual presentation of mackerel data

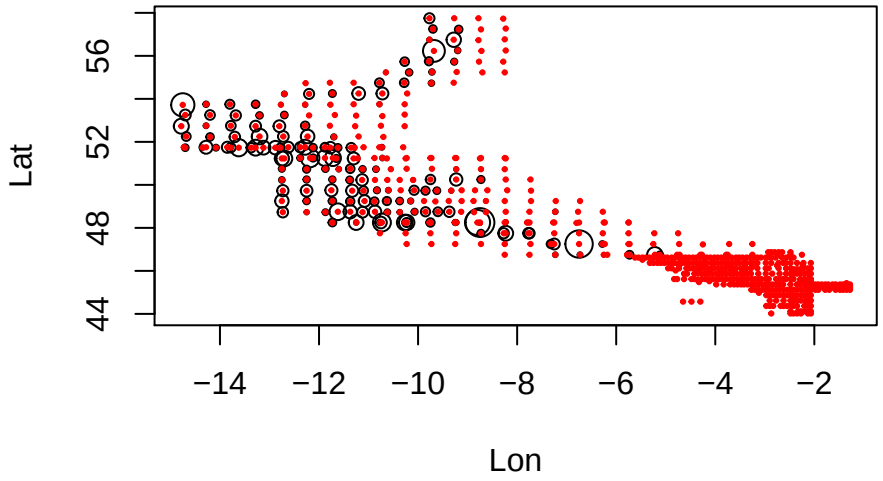
The data which we are using as an example is loaded as follows:

```
data(mack, package="gamair")
str(mack)
```

```
## 'data.frame':    634 obs. of  16 variables:
##  $ egg.count  : num  0 0 0 1 4 3 0 1 0 0 ...
##  $ egg.dens   : num  0 0 0 18.3 90.2 ...
##  $ b.depth    : num  4342 4334 4286 1438 166 ...
##  $ lat        : num  44.6 44.6 44.6 44 44 ...
##  $ lon        : num  -4.65 -4.48 -4.3 -2.87 -2.07 -2.13 -2.27 -2.35 -2.42 -2.48 ...
##  $ time       : num  8.23 9.68 10.9 19.33 8.78 ...
##  $ salinity   : num  35.7 35.7 35.7 35.7 NA ...
```

```
##  $ flow       : num  417 405 377 420 354 373 375 364 375 362 ...
##  $ s.depth    : num  104 98 101 98 101 100 100 102 100 100 ...
##  $ temp.surf  : num  15 15.4 15.9 16.6 16.7 16.6 17.1 17.1 17.3 16.9 ...
##  $ temp.20m   : num  15 15.4 15.9 16.6 16.7 16.6 17.1 17.1 17.3 16.9 ...
##  $ net.area   : num  0.242 0.242 0.242 0.242 0.242 0.242 0.242 0.242 0.242 0.242 ...
##  $ country    : Factor w/ 4 levels "EN","fr","IR",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ vessel     : Factor w/ 4 levels "CIRO","COSA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ vessel.haul: num  22 23 24 93 178 179 181 182 183 184 ...
##  $ c.dist     : num  0.84 0.859 0.893 0.396 0.04 ...
```

```
loc <- cbind(mack$lon, mack$lat)
plot(loc, cex=2*mack$egg.count/max(mack$egg.count), xlab="Lon", ylab="Lat")
points(loc, pch=19, cex=0.3, col="red")
```



The assumption of modelling is like below (sorry for simple text…) but this is a conventional GAM with the spatial correlation and environmental covariates.

$$Y_i \sim Pois(\lambda_i)$$

$$\log \lambda_i = log(net.area) + s(temp.20m) + s(lon, lat)$$

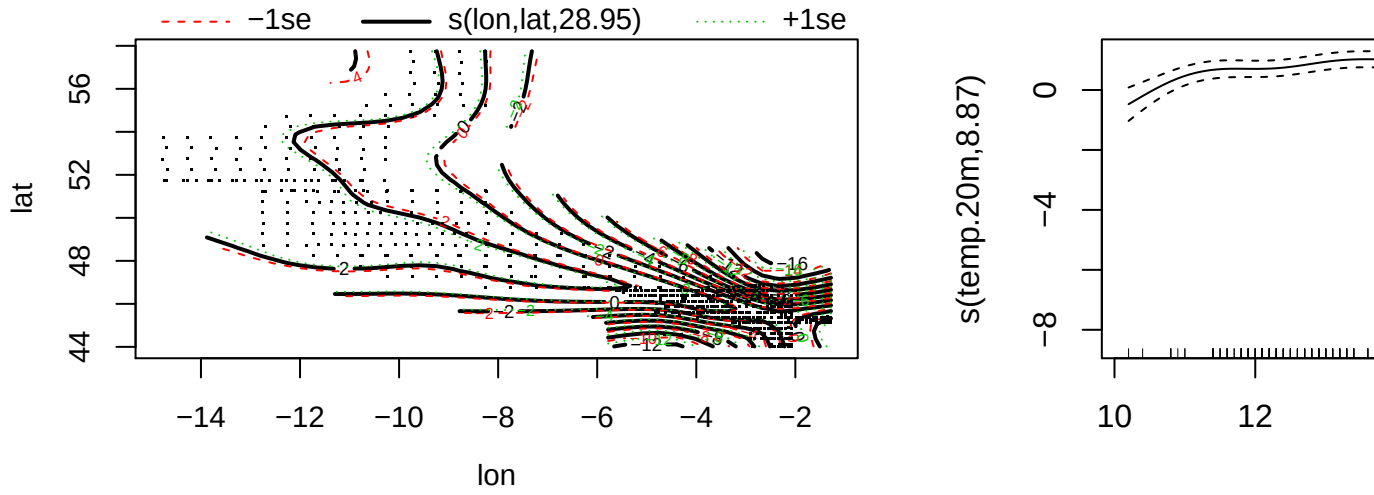
8.2 GAM analysis

```
Res.gam <- gam(egg.count~offset(net.area)-1+s(lon,lat)+s(temp.20m),
               family="poisson", data=mack)

summary(Res.gam)
```

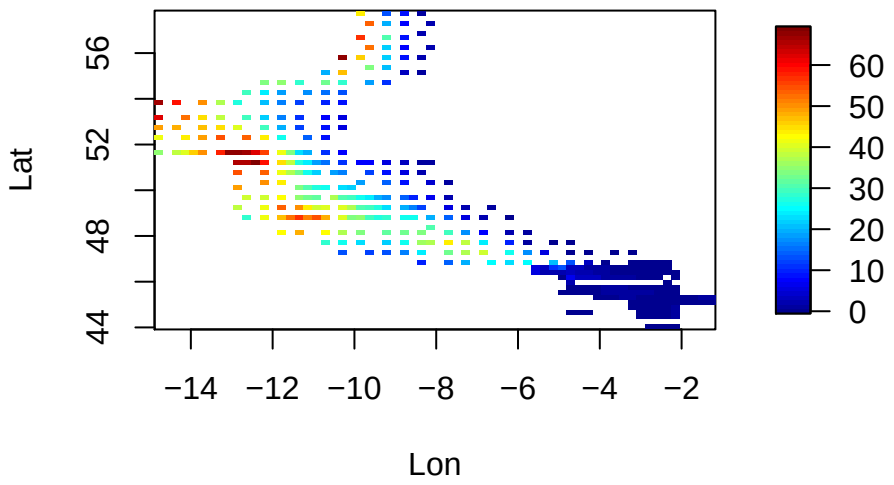
```
##
## Family: poisson
## Link function: log
```

```
##
## Formula:
## egg.count ~ offset(net.area) - 1 + s(lon, lat) + s(temp.20m)
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq p-value
## s(lon,lat)  28.947 28.999   3154  <2e-16 ***
## s(temp.20m)  8.871  8.989    276  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.644   Deviance explained = 76.9%
## UBRE =  5.833  Scale est. = 1         n = 634
```

```
zz <- Res.gam$fitted.values

plot(Res.gam)
```

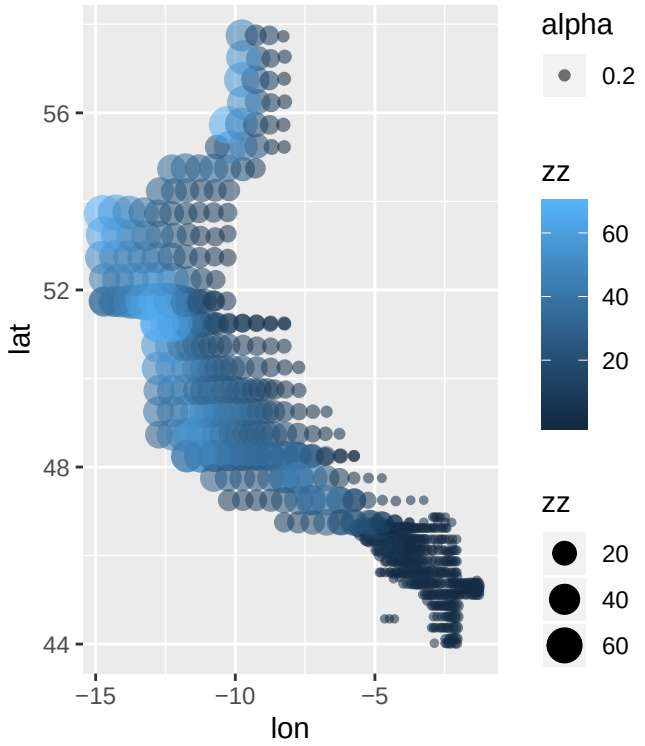



```
quilt.plot(mack$lon, mack$lat, zz, xlab="Lon", ylab="Lat")
```



8.3 Mapping of results

```
ggplot(mack) +
  geom_point(aes(x=lon,y=lat,alpha=0.2,size=zz,col=zz)) +
  coord_quickmap()
```




```
library(marmap)
Lon.range = c(-15, 0)
Lat.range = c(44, 60)
#dat <- getNOAA.bathy(Lon.range[1],Lon.range[2],Lat.range[1],Lat.range[2],res=5)
```

```
#autoplot(dat, geom=c("r", "c"), colour="white", size=0.1)+
# scale_fill_etopo()+
# labs(x="Longitude",y="Latitude",fill="Depth") + ggtitle("Map") +
#  geom_point(data=mack,aes(x=lon,y=lat,alpha=0.2,size=zz,col=zz))+
# coord_quickmap()
```

```
Res.gam <- getViz(Res.gam)
plotRGL(sm(Res.gam,1), residuals = TRUE)
```