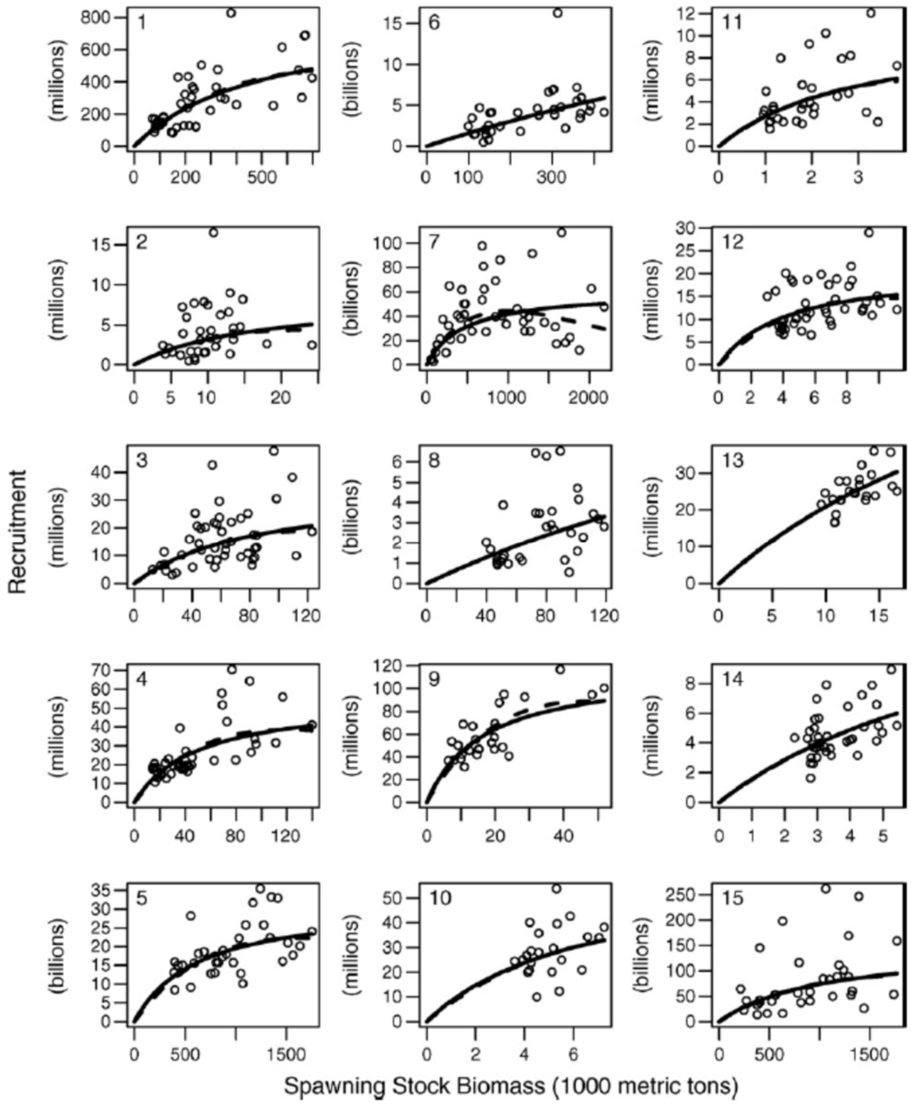
Lecture 8: Stock Recruitment models



Noel Cadigan

CFER

Centre for Fisheries Ecosystems Research

F6004 Lecture 8 Outline

Stock Recruitment models (Haddon Chapter 9)

1) Importance of stock-recruit models in sustainable fisheries management
2) Properties of a good stock-recruit relationship
3) Important life-history processes
4) Stock-recruit data uncertainties
5) Beverton-Holt, Ricker, Hockey-Stick models
6) Estimation
7) Nonparametric stock-recruit model
    a. Details of the Scam() package
8) Auto-correlated errors

Fogarty, M. J., and O’Brien, L. 2009. Recruitment in marine fish populations. In Fish Reproductive Biology: Implications for Assessment and Management. Ed. by T. Jakobsen, M. Fogarty, B. A. Megrey, and E. Moksness. Wiley-Blackwell.

Needle, C.L., 2002. Recruitment models: diagnosis and prognosis. Reviews in Fish Biology and Fisheries, 11: 95-111.

Stock Recruitment models

- Stock production (i.e. yield to fishery) primarily involves (1) recruitment of new individuals to the stock and (2) growth of individuals in the stock (see lecture 6)
- We looked at spawner per recruit (SPR) earlier
- This lecture deals with the recruit per spawner (RPS) relationship
- In this section we focus on the mathematical description, and estimation, of recruitment

Recruitment of individuals

- Stock-recruit models deal with how many new fish can we expect to enter the exploited population as a function of parental stock size
- this is important when estimating sustainable levels of fishing mortality (i.e. Fmsy, etc)
- recruitment over-fishing – taking more fish than the stock is producing. The parental stock size is reduced to a low level in which it cannot reproduce enough to replace the deaths

leads to stock collapse

Stock-recruit relationship

- Understanding the relationship between parental stock size (*S*) and subsequent reproduction and recruitment (*R*) of juveniles to a fishery is widely recognized as a fundamental component of sustainable fisheries management (Quinn and Deriso, 1999).
- Stock-recruit (SR) relationships are used to “longer-term” project future fish population dynamics in response to proposed management actions

Stock-recruit relationship

- and to determine management reference points (RP’s; Needle, 2002), esp. MSY but also Blim’s.
- Many fisheries are managed using RPs, where prescribed actions should occur when stock size or fishing mortality rates transgress RPs.
- RPs are widely considered an essential part of well-managed fisheries (e.g. Hilborn and Stokes, 2010).
- SR relationships play an important role when determining what the stock size and level of fishing should be

Stock-recruit relationship

- Reliable SR models are therefore important for successful fisheries management.
- However, there are other important factors that determine how much recruitment occurs (e.g. physical environments, predators).
- Therefore, SR relationships are notoriously difficult to model.
- There is usually much noise to deal with, and it can be difficult to detect recruitment over-fishing

Properties of a good SR relationship

1. Passes through the origin i.e. *S=0* implies *R=0*. That is, no parents (*S*) means no recruitment (*R*)

2. *R>0* when *S* is large

3. *R/S* non-increasing as *S* increases.

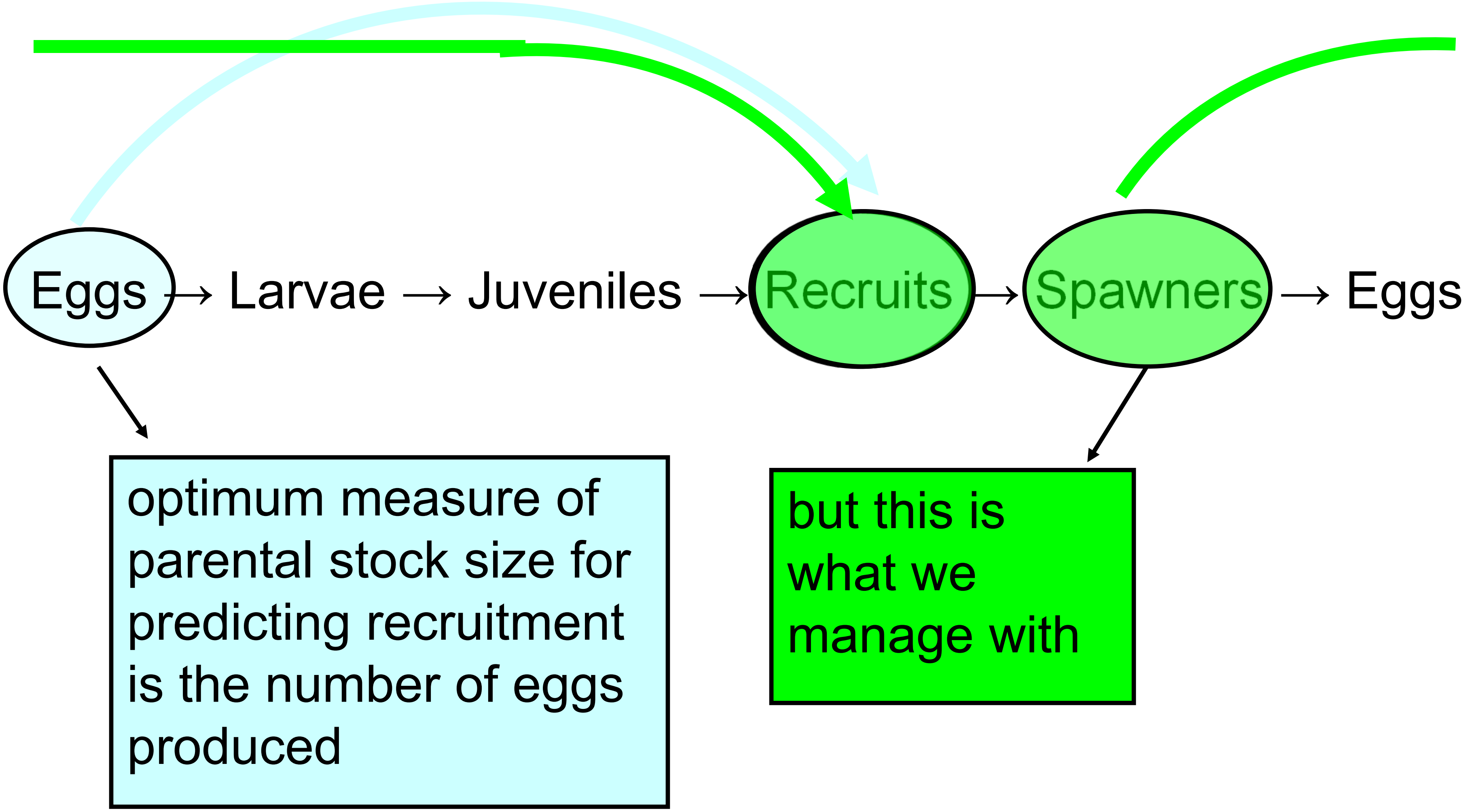
4. *R > {1 - exp(-M)}S*

also

5. *R* should be a smooth function of *S*

6. *R(S)* should not depend on time?? often does

The SR relationship

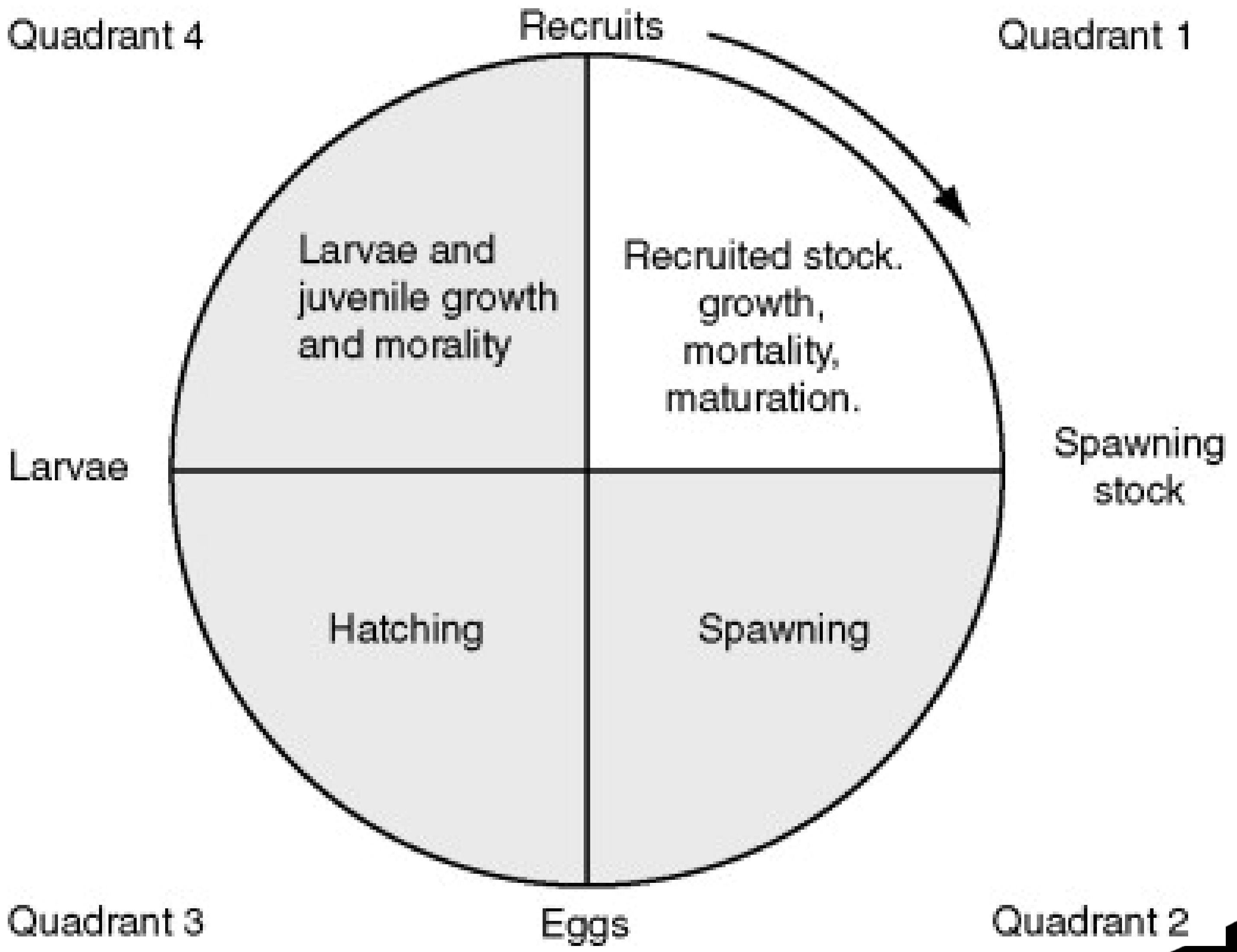


Eggs → Larvae → Juveniles → Recruits → Spawners → Eggs

optimum measure of parental stock size for predicting recruitment is the number of eggs produced

but this is what we manage with

Paulik Diagram



The data - *S*

1. Average fecundity at age
2. number of females *x* average fecundity = egg production
3. biomass of mature females
4. index of population size

by order of reliability

The data - *R*

- The number of fish that first become vunerable to capture by fishing
- This tends to be a gradual process, with increasing vunerability to fishing with age
- It can change over time
- Sometimes recruitment can mean a younger age, especially when the surveys catch smaller sizes than the fishery
- Hence definition of *R* is case specific

The data

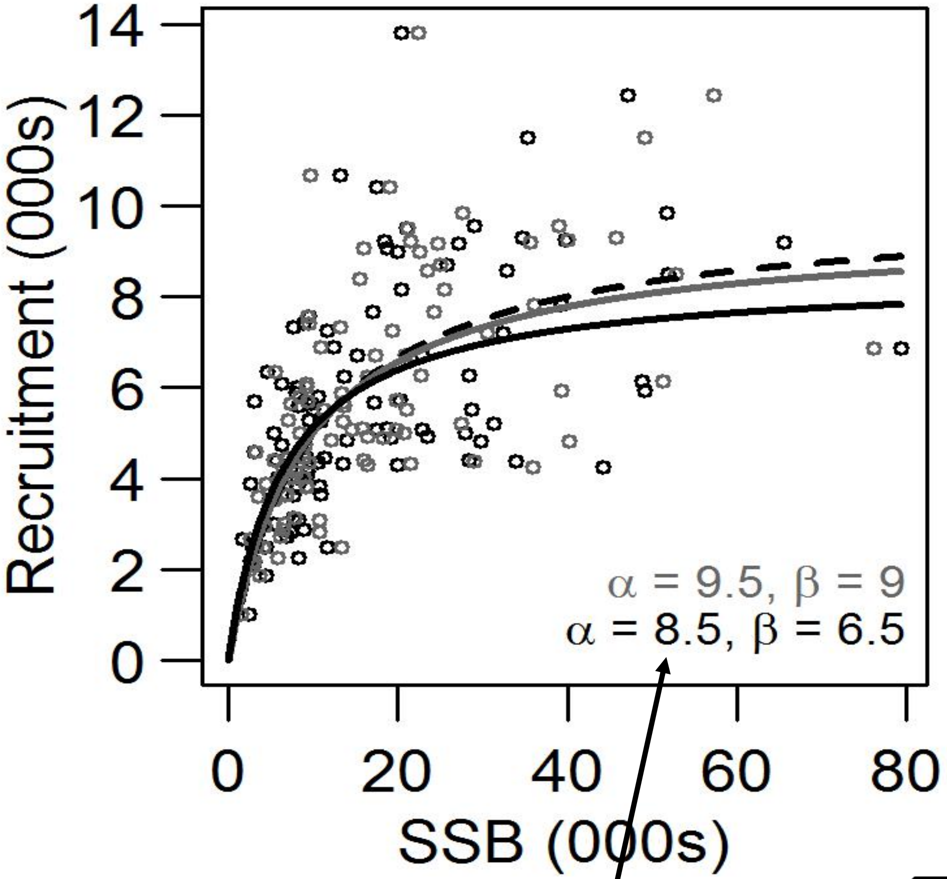
- Typically SR data are outputs from a stock assessment (SA) model
- Not really data, but estimates of population quantities
- Sometimes the SR data are derived directly from surveys (i.e. survey SSB, and survey recruitment)
- Still not data, but estimates
- Survey SR “data” have different dependencies than SA model SR “data”
- Sometimes the SR model is estimated within the SA model, but that’s a topic of current research

Measurement error in S

- Of course this happens, and it can obscure the SR relationship
- Measurement error can result in biased estimates of important SR parameters
- Haddon has text on this

Stock size measurement error (ME)

- Simulation: 100 SR observations generated from a BH model (black dashed line) with α=10, β=10.
- Black symbols indicate the observed data points and grey symbols indicate the data points with true SSB values.
- Estimated SR models are shown as solid lines with colors corresponding to the data used to estimate model parameters
- Max R (Rmax) under-estimated and slope-at-origin (Sao) over-estimated when there is ME in S



est slope-at-origin =1.3
True = 1

Measurement error in S

- There is no “best” solution to deal with measurement error
- Need additional information to determine measurement error variance
- Should anticipate auto-correlation in *S* measurement errors

Measurement and process errors

- There will usually be substantial measurement error (ME) in *R* and *S*.

- ME in *S* is usually less than the ME in *R* because *S* is usually derived as the sum of estimates of biomass-at-age times maturity-at-age, and

- age-specific biomass errors will tend to cancel in the sum (i.e. *S*) especially when there are many age classes in a stock.

- A justification for the common estimation methods used to fit SR functions – described later

Measurement and process errors

- there can be changes in the reproductive potential of a population, or its potential to produce recruitment, due to changes in maturation, sex ratio, and fecundity
- Hence, there are additional process errors (PE’s) in *S* as a proxy for reproductive potential
- There are substantial PE’s or environmental variability that affects the **low** survival of egg and larvae and ultimately how much *R* is derived from *S* parents.
- A challenge to untangle the ME and PE’s (for F6005)

Recruitment

- Hjort (1914): events during the early life stages were likely to be major determinants of *R* to fishery
- there appeared to be limited effect of *S* on future yield
- strong cohorts occurred when spawning occurred at a time "to ensure an adequate supply of nourishment for the young larvae at a stage when this was required"
- Also, vagaries in ocean currents might be detrimental to survival
- Pre-recruit mortality can be very large and very variable

Beverton-Holt SR model

- Based on the assumption that egg production, N_o, is proportional to S, $N_o = fS$
- The Beverton-Holt (BH) SR model is derived from the logistic growth model (earlier Lecture)

$$N(t) = \frac{KN_o \exp(rt)}{K - N_o + N_o \exp(rt)}$$

$$= \frac{KN_o \exp(rt)}{K + N_o \{\exp(rt) - 1\}}$$

Beverton-Holt SR model

- If $R = N(t_o)$ then, using $N_o = fS$,

$$R = N(t_o) = \frac{KN_o \exp(rt_o)}{K + N_o\{\exp(rt_o) - 1\}}$$

$$= \frac{KfS \exp(rt_o)}{K + fS\{\exp(rt_o) - 1\}}$$

$$= \frac{\alpha S}{\beta + S}$$

$\alpha = Kexp(rt_o)/\{exp(rt_o) -1\}$

$\beta = K/\{exp(rt_o) -1\}f$

$$R = \frac{\alpha^* S}{1 + \beta^* S}$$

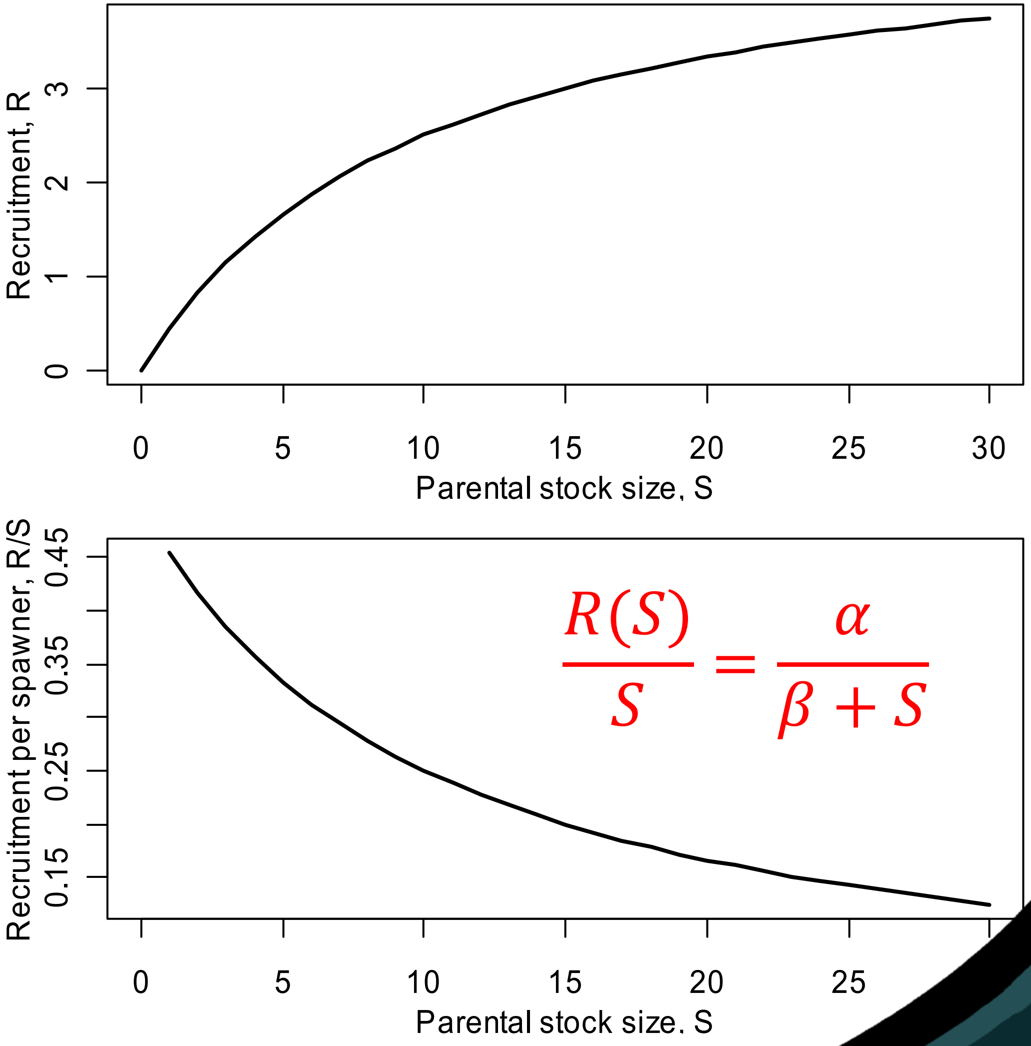
sometimes formulated like this

Beverton-Holt SR model

$$\lim_{S\to\infty} R(S) = \alpha$$

$$\frac{dR(S)}{dS} = \frac{\alpha\beta}{(\beta+S)^2}$$

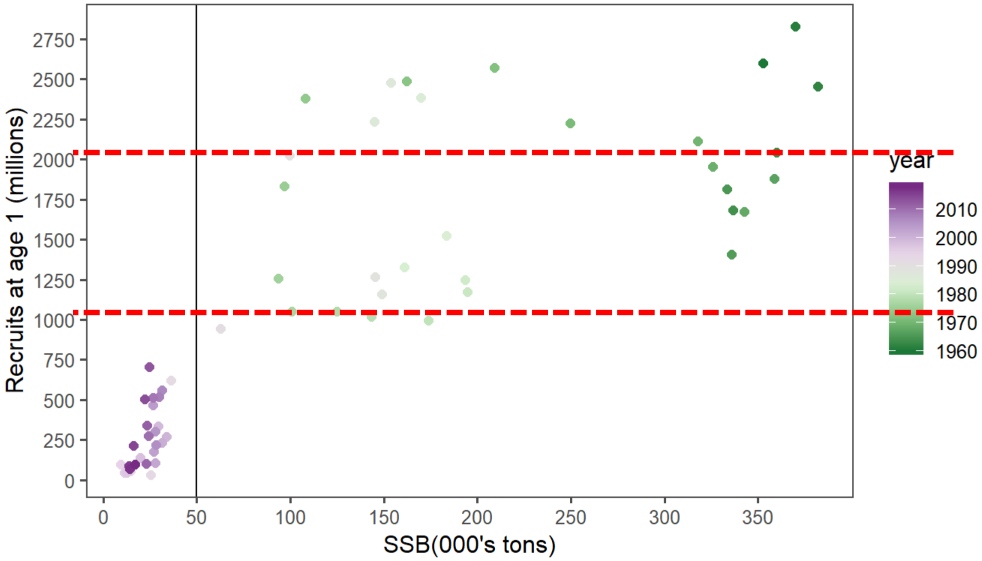
$$\left.\frac{dR(S)}{dS}\right|_{S=0} = \frac{\alpha}{\beta}$$



$$\frac{R(S)}{S} = \frac{\alpha}{\beta+S}$$

Beverton-Holt properties

- Maximum recruitment is *Rmax* = *α*
- The stock size that gives 50% of *Rmax* (S50) is *β*
- The slope at the origin (Sao) is *α/β*.



Ricker SR model

- Is based on the simple exponential growth model
- and is defined by the differential equation

$$\frac{dN}{dt} = -(a + bS)N$$

- If the dependence of *S* on *N* is ignored then the solution to the differential equation is

$$N(t) = N_o \exp\{-(a + bS)t\}$$

Ricker SR model

– If *$R = N(t_o)$* then, using *$N_o = fS$,*

$$R = N_o \exp\{-at_o - bt_o S\}$$

$$= fS \exp\{-at_o\} \exp\{-bt_o S\}$$

$$= \alpha S \exp(-\beta S)$$

$\alpha = fexp(at_o)$

$\beta = bt_o$

$$R = S \exp(\alpha^* - \beta S)$$

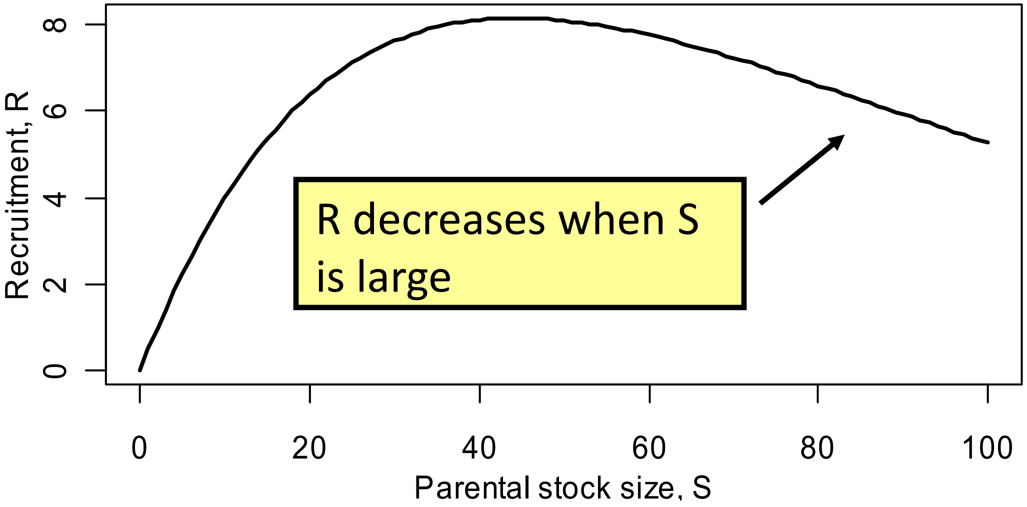
sometimes formulated like this

Ricker SR model

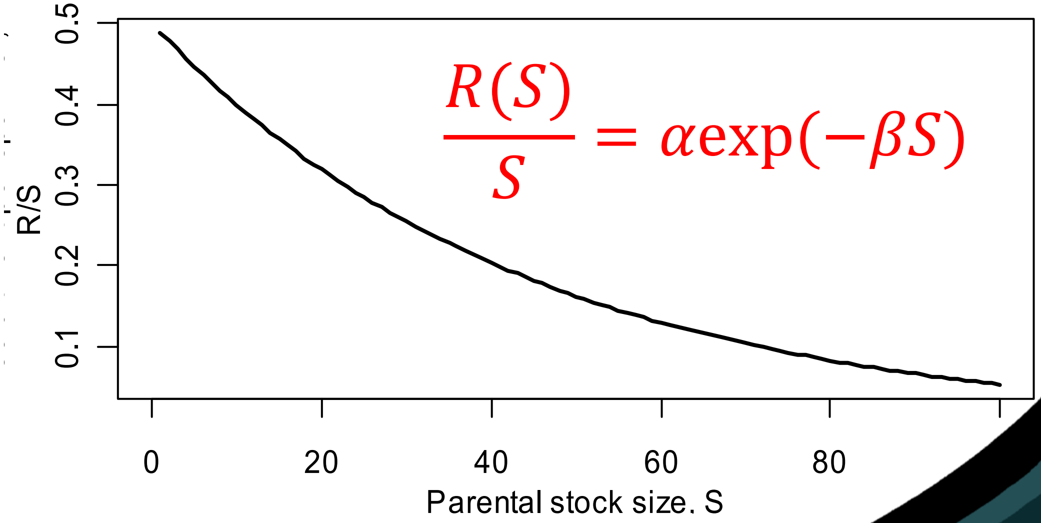
$$\lim_{S\to\infty} R(S) = 0$$

$$\frac{dR(S)}{dS} = \alpha e^{-\beta S}\left(1-\beta S\right)$$

$$\left.\frac{dR(S)}{dS}\right|_{S=0} = \alpha$$



R decreases when S is large



$$\frac{R(S)}{S} = \alpha\exp(-\beta S)$$

Ricker properties

- Maximum recruitment is *Rmax = α/βexp(1)*
- This occurs at *Smax = 1/β*
- The **lower** stock size that gives 50% of *Rmax* (S50) does not have a closed-form solution, but is approximately *1/(4.44xβ)*
- The slope at the origin (Sao) is *α*.

BH Estimation

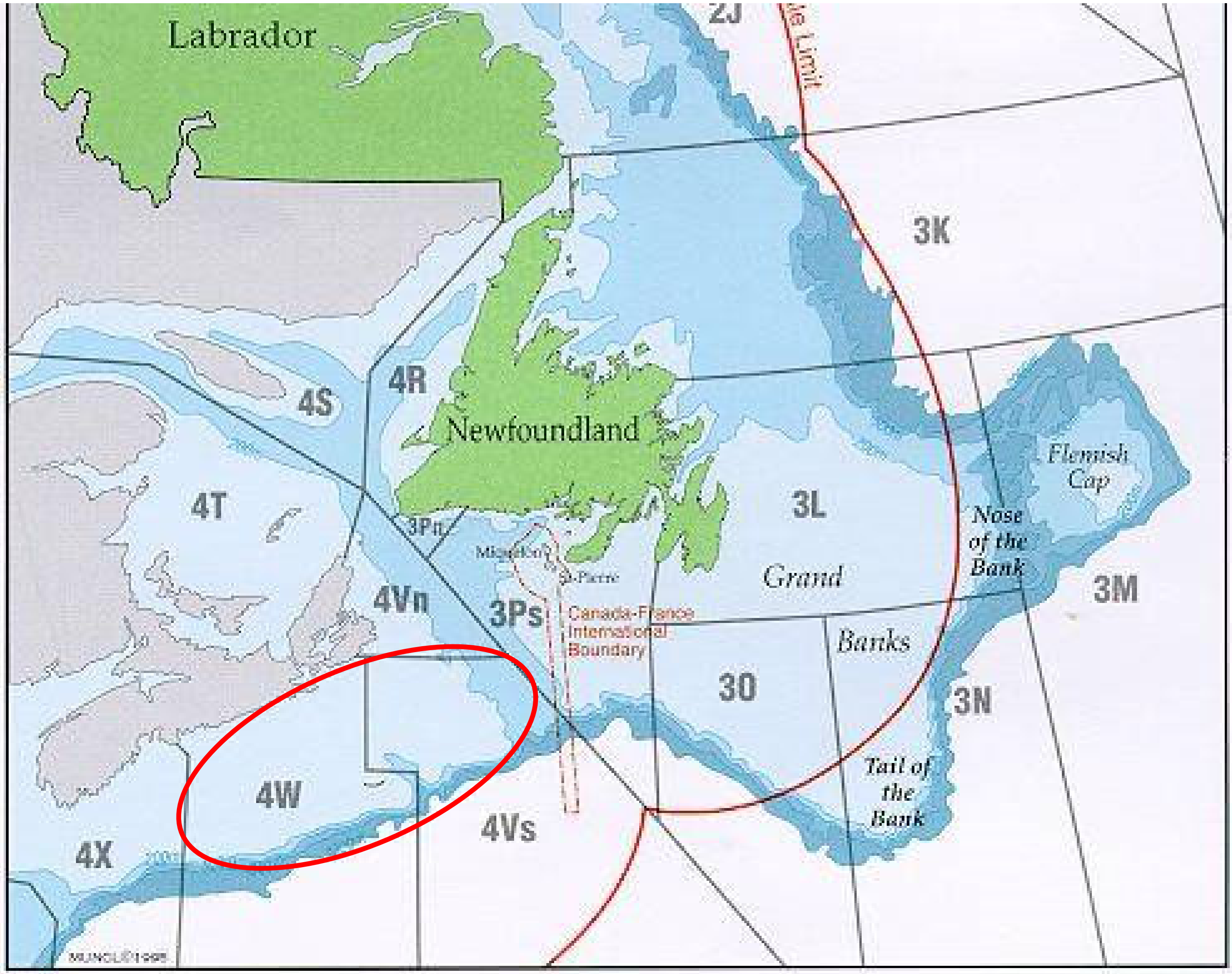
- Assume we have *n* observations of parental stock size *S* and recruitment *R*.

- It is common practise to fit SR models assuming lognormal error in *R*, and no error in *S*

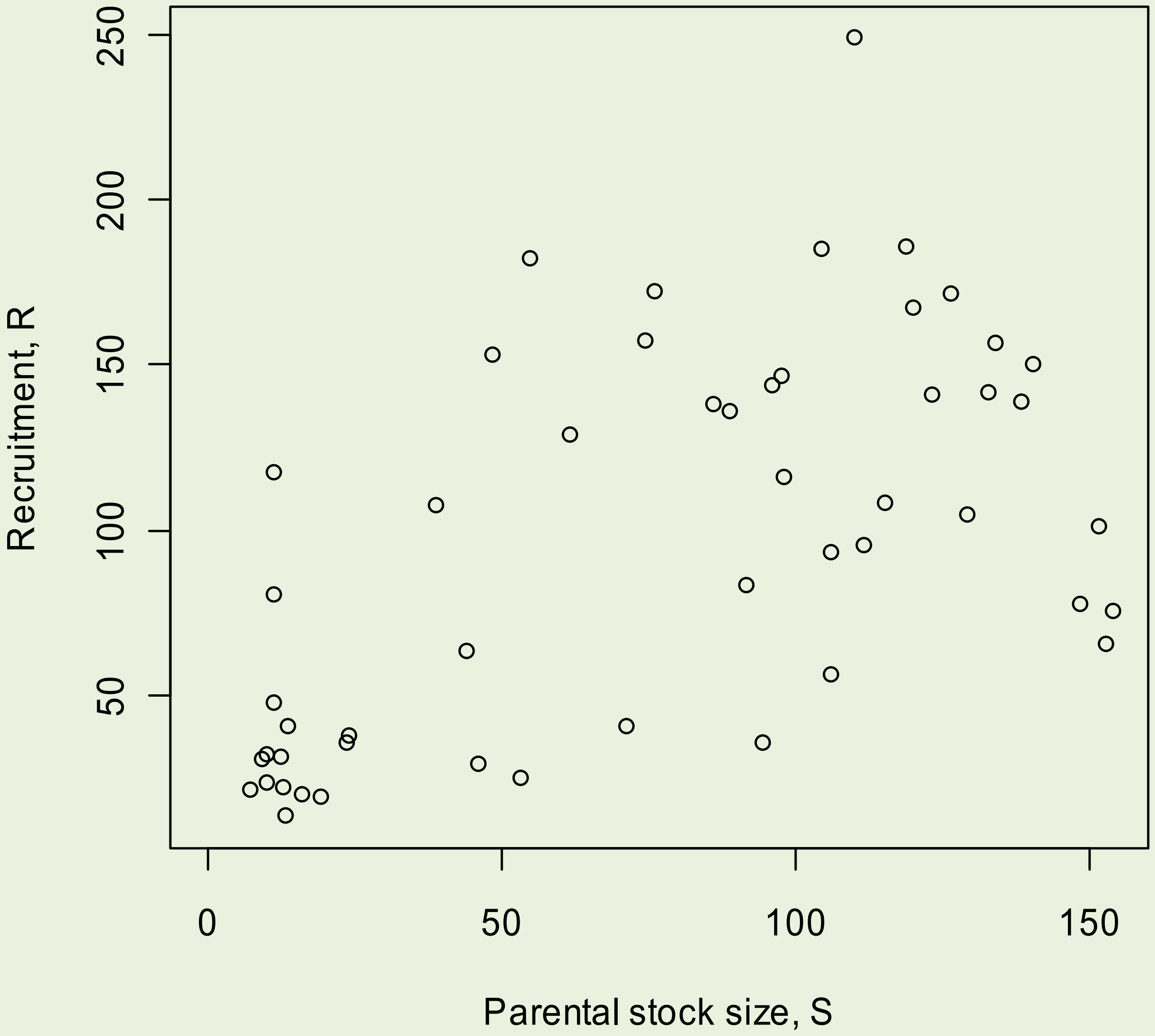
$$R = \mu(S)\exp(\varepsilon) \qquad \varepsilon \sim N(0, \sigma^2{}_{\varepsilon})$$

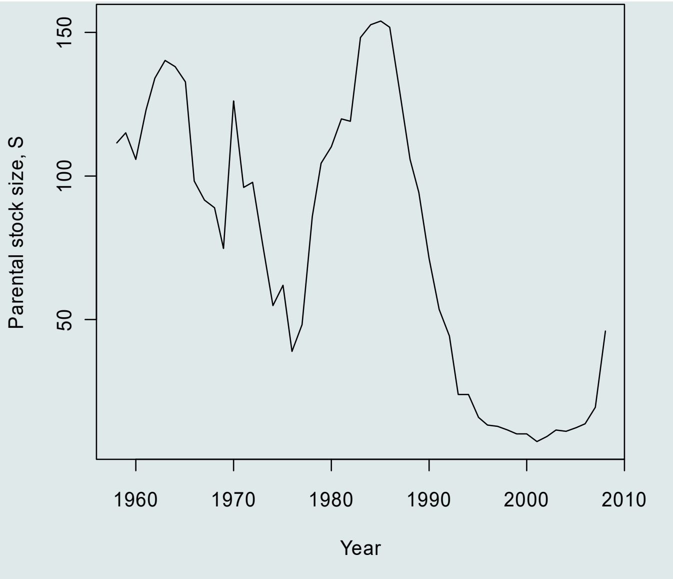
$$\mu(S) = \alpha S / (\beta + S)$$

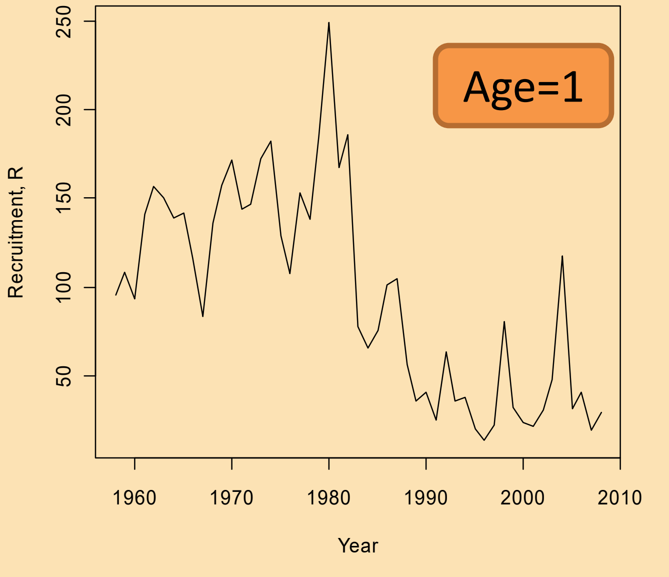
4VsW cod


2J
Limit
Labrador
3K
4R
4S
Newfoundland
Flemish Cap
4T
3L
3Pn
Miquelon
St-Pierre
Nose of the Bank
Grand
Banks
4Vn
3Ps
Canada-France International Boundary
3M
30
3N
4W
Tail of the Bank
4Vs
4X
MUNCL©1998

Some 4VsW cod SR data







R lognormal estimation of BH for 4VsW cod

```
> fname = c(‘SR.txt')
> sr.data = read.table(fname,header=T)
> names(sr.data)
[1] "year" "ssb"  "rec"
>
```

read data, and store in sr.data

```
sr.data$logrec = log(sr.data$rec)
sr.data$logssb = log(sr.data$ssb)
```

create log of ssb and rec, and store

```
init.Rmax = 150
init.S50 = 50
init.alpha= init.Rmax
init.beta = init.S50
```

initial values for nls. Rmax and S50 obtained by visually inspecting SR scatter plot on the previous slide

BH LN Estimation

– The statistical model we can use in nls is

$$\log(R) = \log\{\mu(S)\} + \varepsilon$$

$$\varepsilon \sim N(0, \sigma^{2}{}_{\varepsilon})$$

$$\begin{aligned}\log\{\mu(S)\} &= \log(\alpha) + \log(S) \\ &\quad - \log(\beta + S)\end{aligned}$$

R LN estimation of BH for 4VsW cod

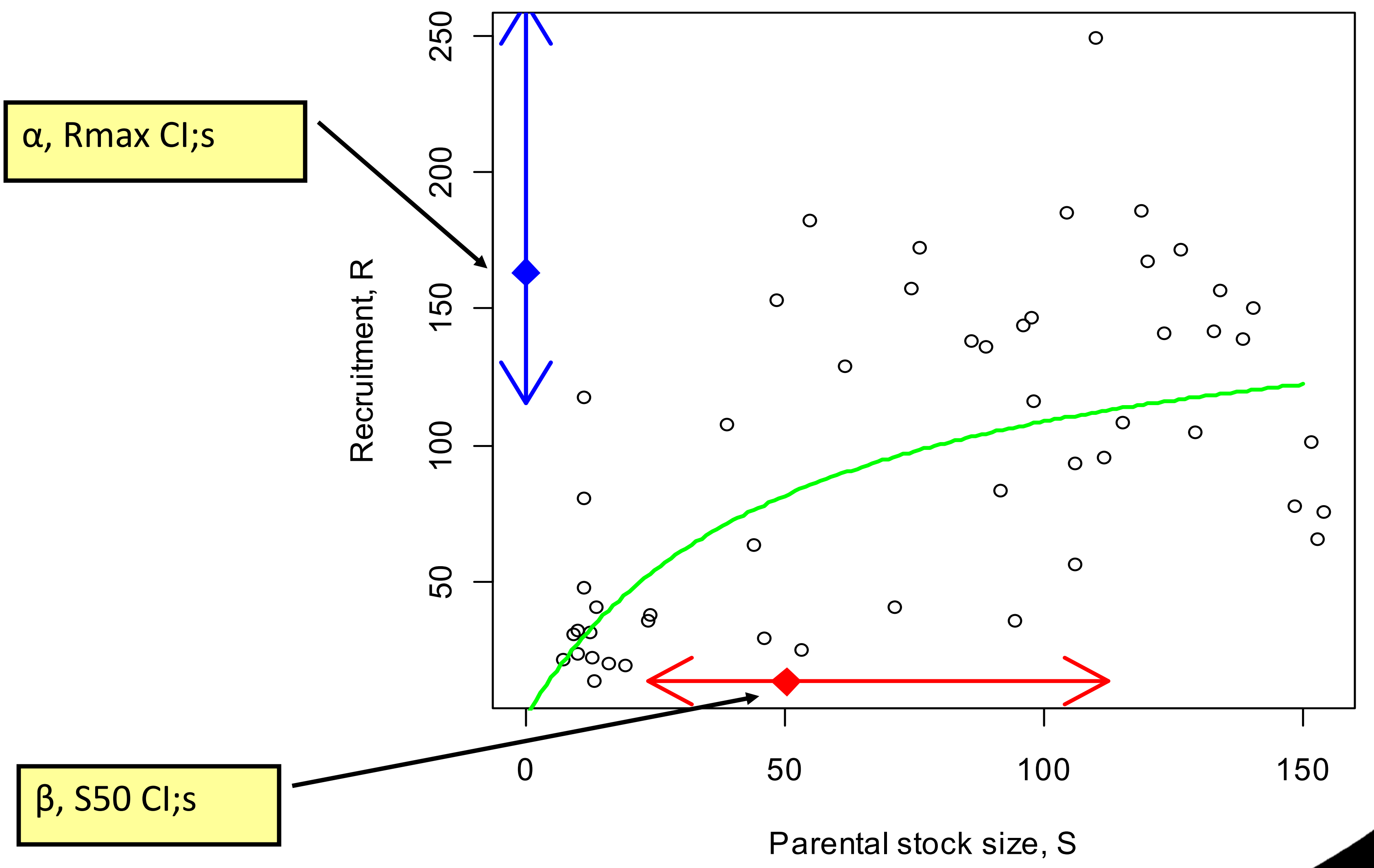
Take log’s on both sides

```
> BH.fit <- nls(logrec ~ log(alpha) + logssb - log(beta + ssb),
+           algorithm="port",lower=c(0,0),data=sr.data,
+           start = list(beta = init.beta,alpha = init.alpha))
> summary(BH.fit)

Formula: logrec ~ log(alpha) + logssb - log(beta + ssb)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
beta     50.34      18.41   2.734  0.00869 **
alpha   163.32      31.41   5.200 3.88e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5615 on 49 degrees of freedom

Algorithm "port", convergence message: relative convergence (4)
```

```
> confint(BH.fit)
Waiting for profiling to be done...
           2.5%      97.5%
beta    23.81713  112.4705
alpha  115.41694  262.1919
```

Wide CI’s

BH Predict for 4VsW cod



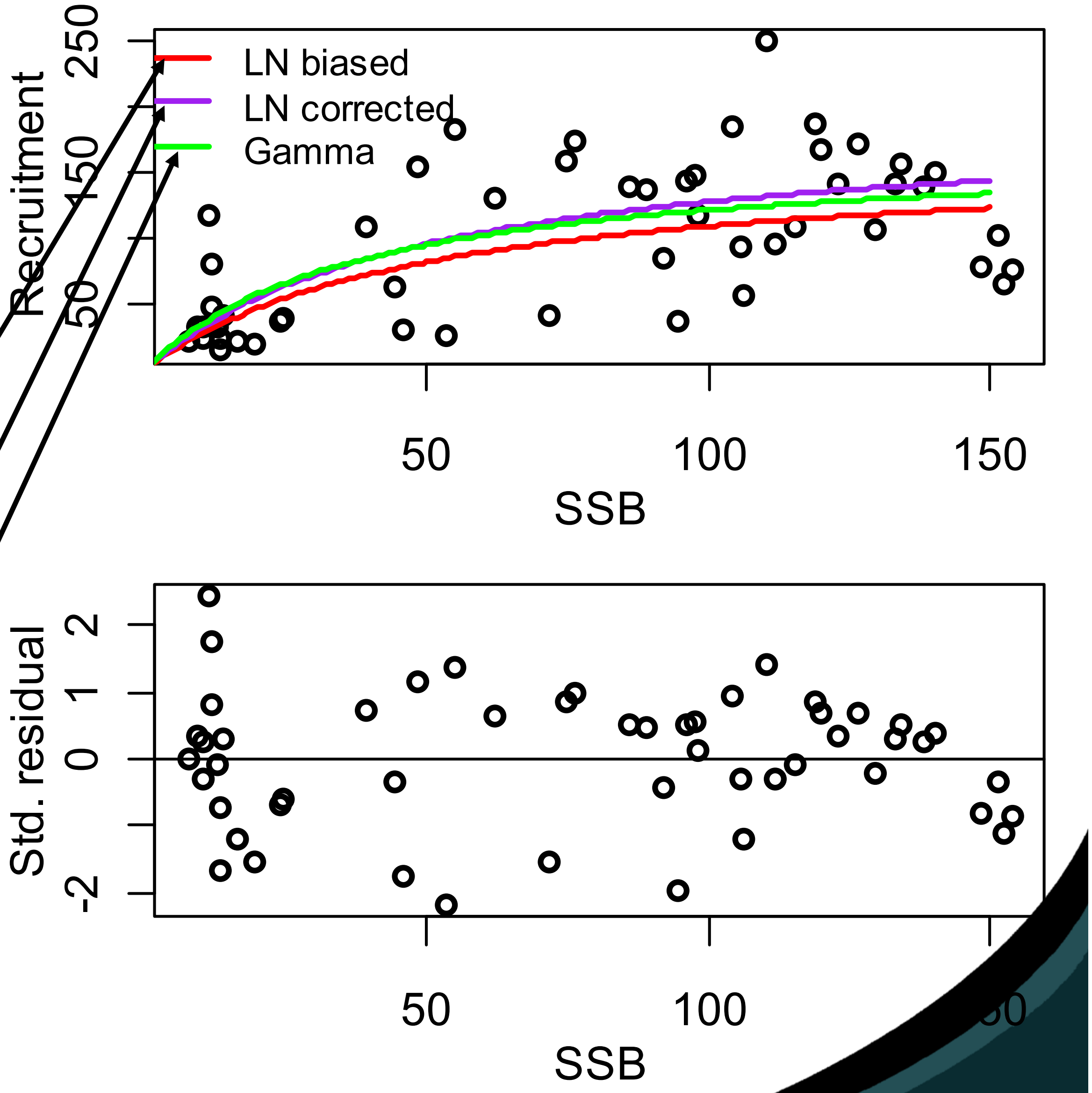
BH Predict and residuals for 4VsW cod

```
ssb.pred = 0:150
BH.predict =
predict(BH.fit,list(ssb=ssb.pred,
logssb=log(ssb.pred)))
```

exp(BH.predict)

exp(BH.predict+(se.resid**2)/2)

BH.predict.gam

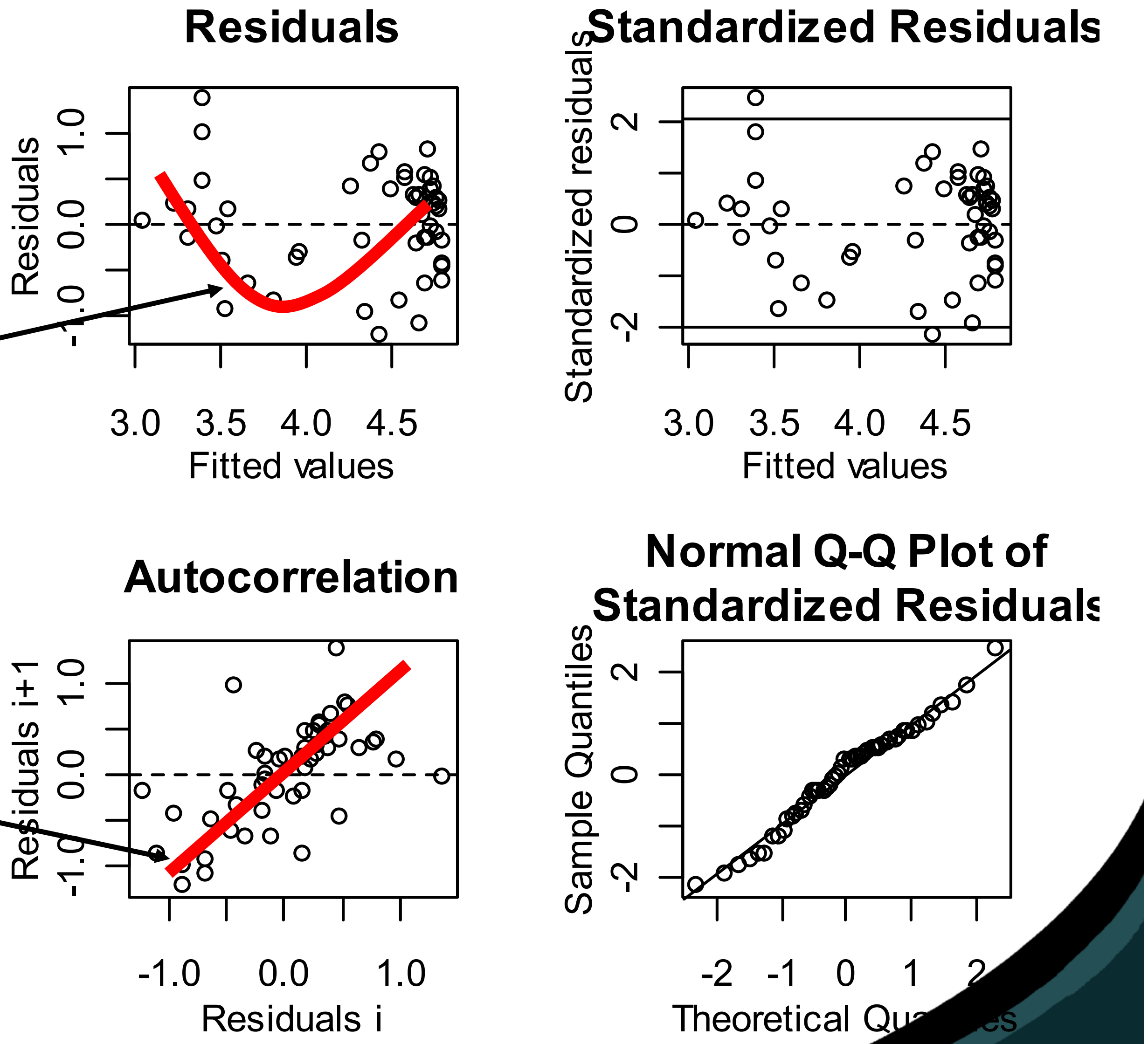


$$E(R) = \mu(S)\exp(\sigma^2/2)$$

Predicted SR based on Gamma mle’s usually similar to transformation bias corrected lognormal mle’s

BH residuals for 4VsW cod

```
nr <- nlsResiduals(BH.fit)
par(mar=c(3.5,3,3,2),
    mgp=c(2,1,0))
plot(nr, which = 0)
```

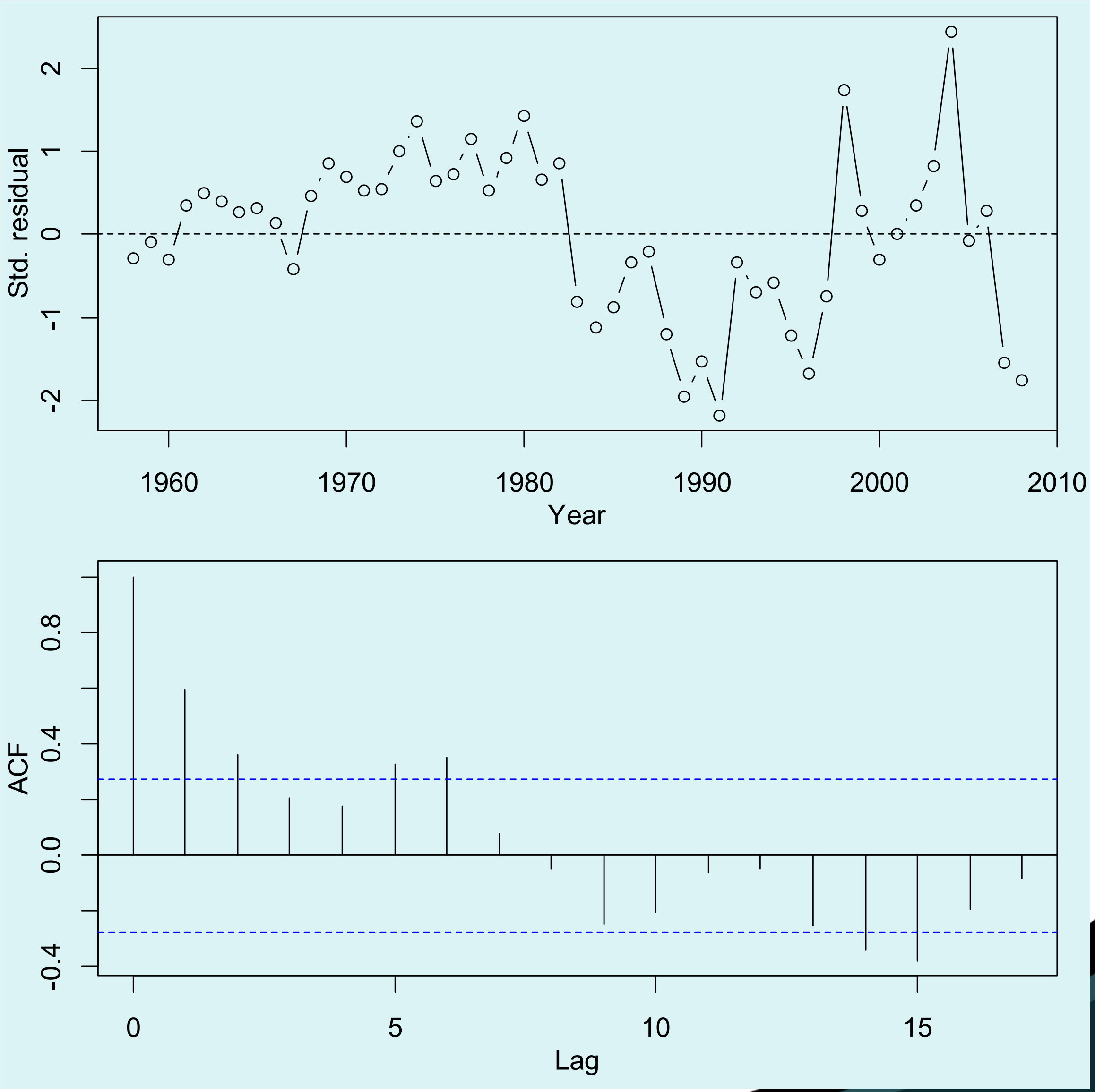


Not good

Not good

BH residuals for 4VsW cod

Not good



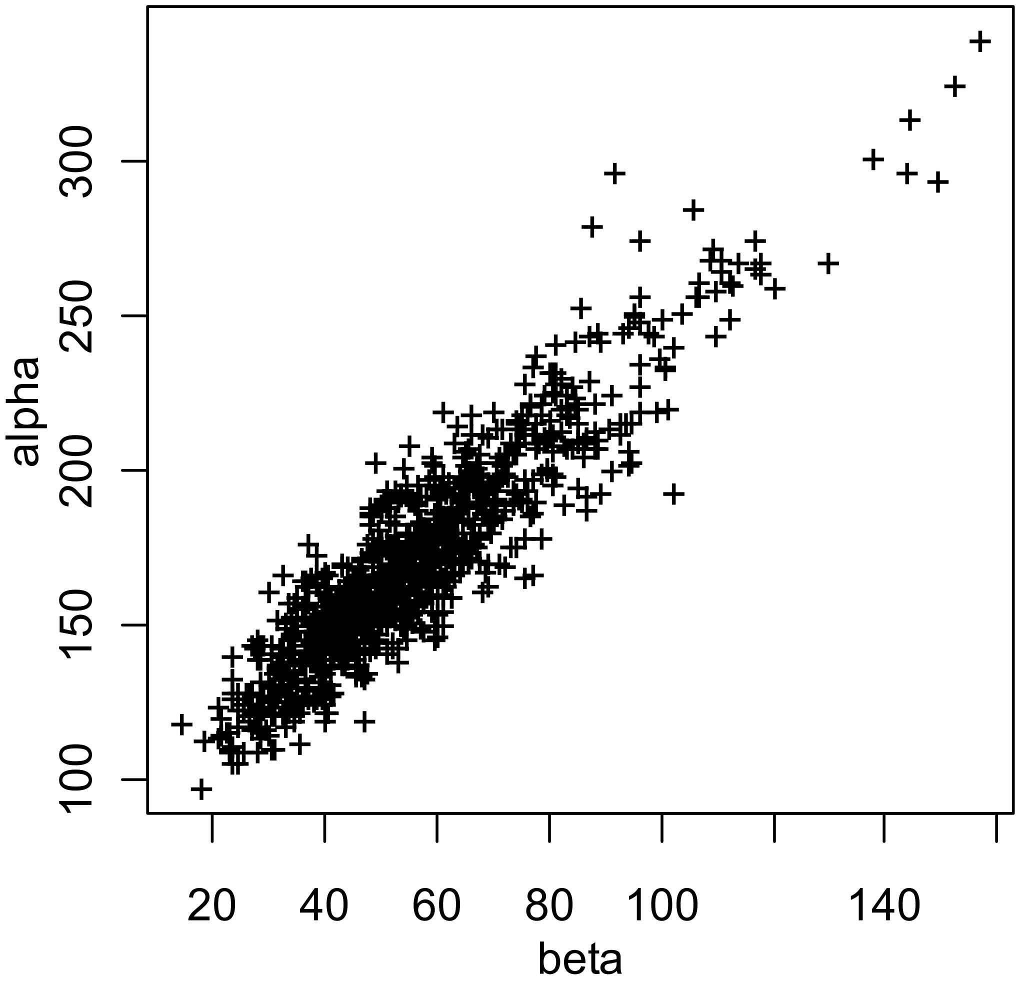
nls BH residual diagnostics for 4VsW cod

```
> test.nlsResiduals(nr)


------
        Shapiro-Wilk normality test

data:  stdres
W = 0.9802, p-value = 0.5463



------
        Runs Test

data:  as.factor(run)
Standard Normal = -4.3583, p-value = 1.31e-05
alternative hypothesis: two.sided
```

P-value >> 0.05
Std. residuals are
seem N(0,1)

P-value << 0.05
autocorrelation in residuals

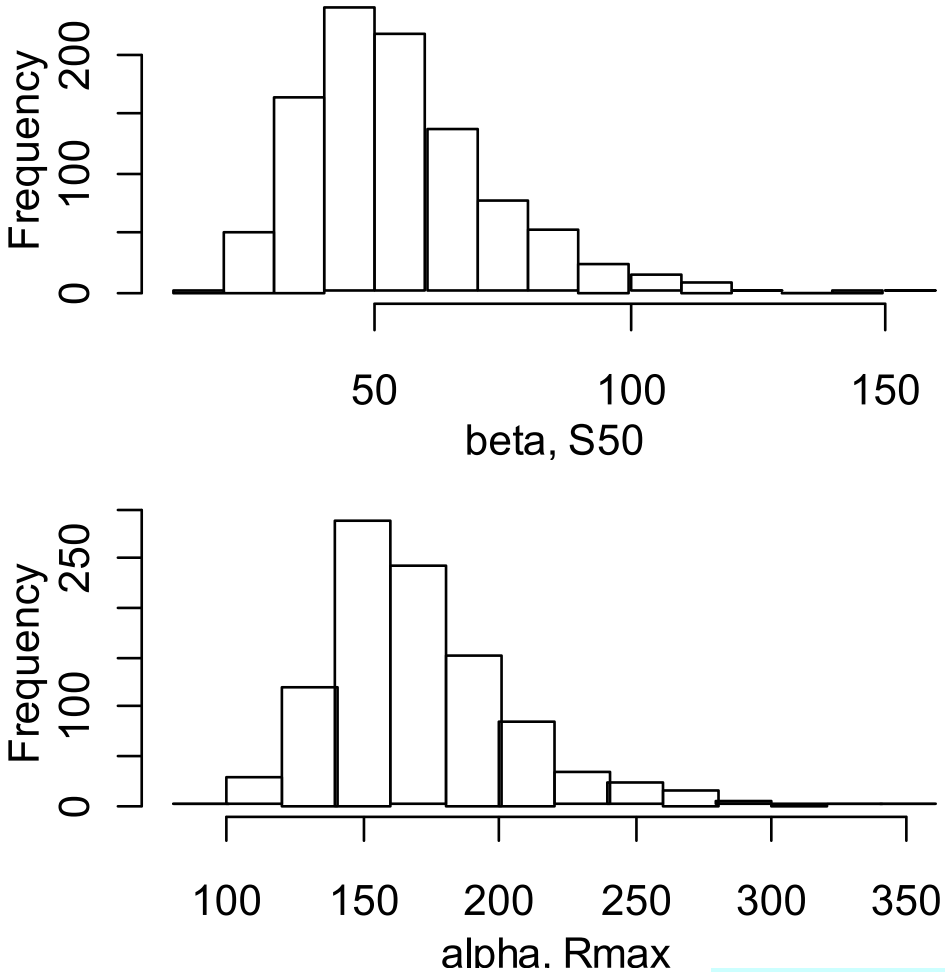
BH bootstrap for 4VsW cod



```
> boo <- nlsBoot(BH.fit)
> plot(boo)
```

There is high correlation in the BH parameter estimates

BH bootstrap for 4VsW cod



```
par(mfrow=c(2,1),mar=c(3,3,1,1),mgp=c(2,1,0))
hist(boo$coef[,1],xlab="beta, S50",main='')
hist(boo$coef[,2],xlab="alpha, Rmax",main='')
```

BH Bootstrap CI’s for 4VsW cod

```
------
Bootstrap estimates
     beta     alpha
 51.83972 164.71769


------
Bootstrap confidence intervals
               2.5%      97.5%
beta     25.95528  105.5379
alpha 119.10793  257.6426
```

These are percentile intervals

```
> confint(BH.fit)
Waiting for profiling to be done...
             2.5%       97.5%
beta    23.81713  112.4705
alpha 115.41694 262.1919
```

Not as wide as profile likelihood CI’s

Ricker LN Estimation

- The statistical model we can use in nls is

$$\log(R) = \log\{\mu(S)\} + \varepsilon$$

$$\varepsilon \sim N(0, \sigma^2{}_{\varepsilon})$$

$$\log\{\mu(S)\} = \log(\alpha) + \log(S) - \beta S$$

R LN estimation of RK for 4VsW cod

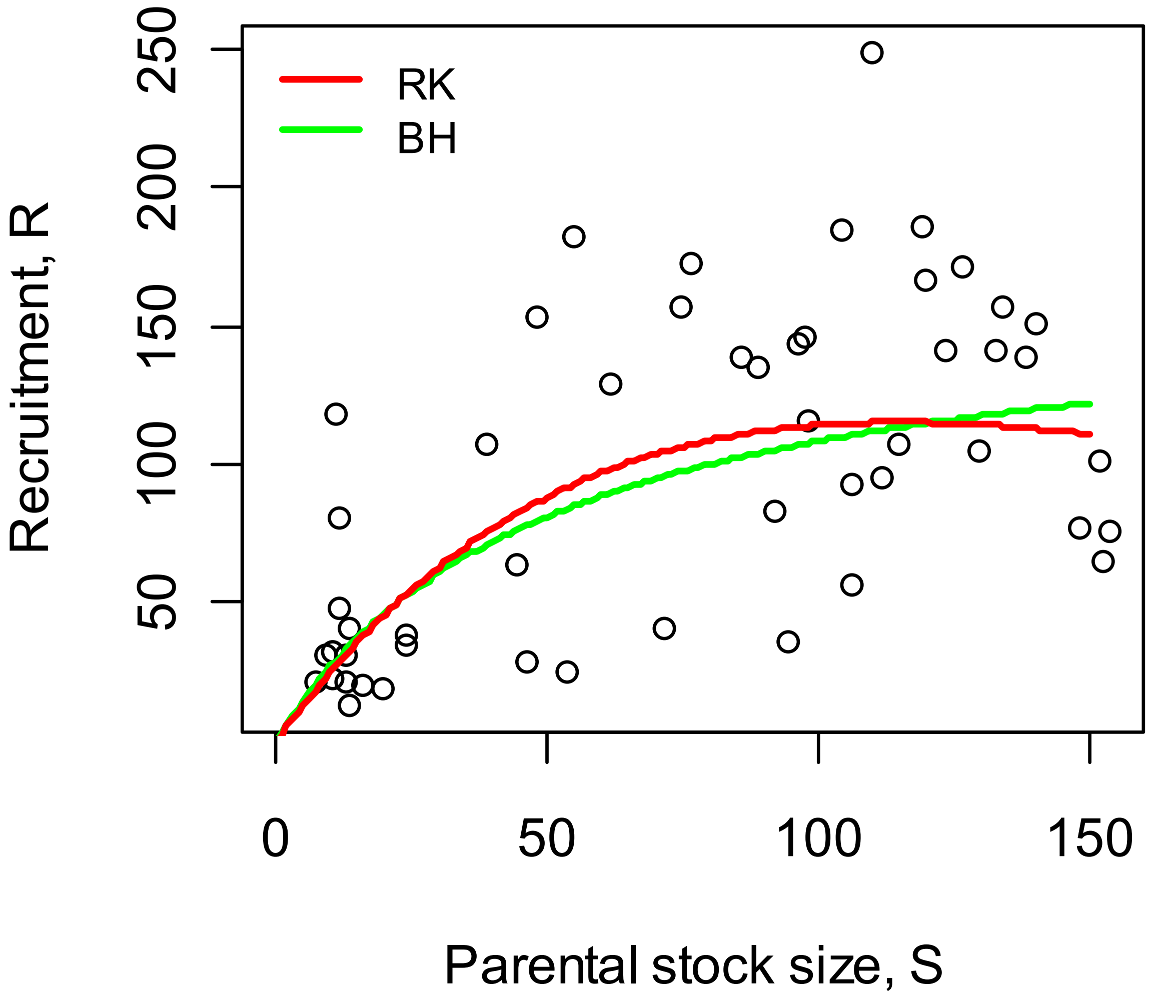
```
> RK.fit <- nls(logrec ~ log(alpha) + logssb - beta*ssb,
+          algorithm="port",lower=c(0,0),data=sr.data,
+          start = list(beta = init.beta,alpha = init.alpha))
> summary(RK.fit)

Formula: logrec ~ log(alpha) + logssb - beta * ssb

Parameters:
      Estimate Std. Error t value Pr(>|t|)
beta  0.008701   0.001621   5.366 2.18e-06 ***
alpha 2.729209   0.397738   6.862 1.09e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5642 on 49 degrees of freedom

Algorithm "port", convergence message: relative convergence (4)
```

```
init.beta = 1/(4.44*init.S50)
init.alpha= init.Rmax*init.beta*exp(1)
```

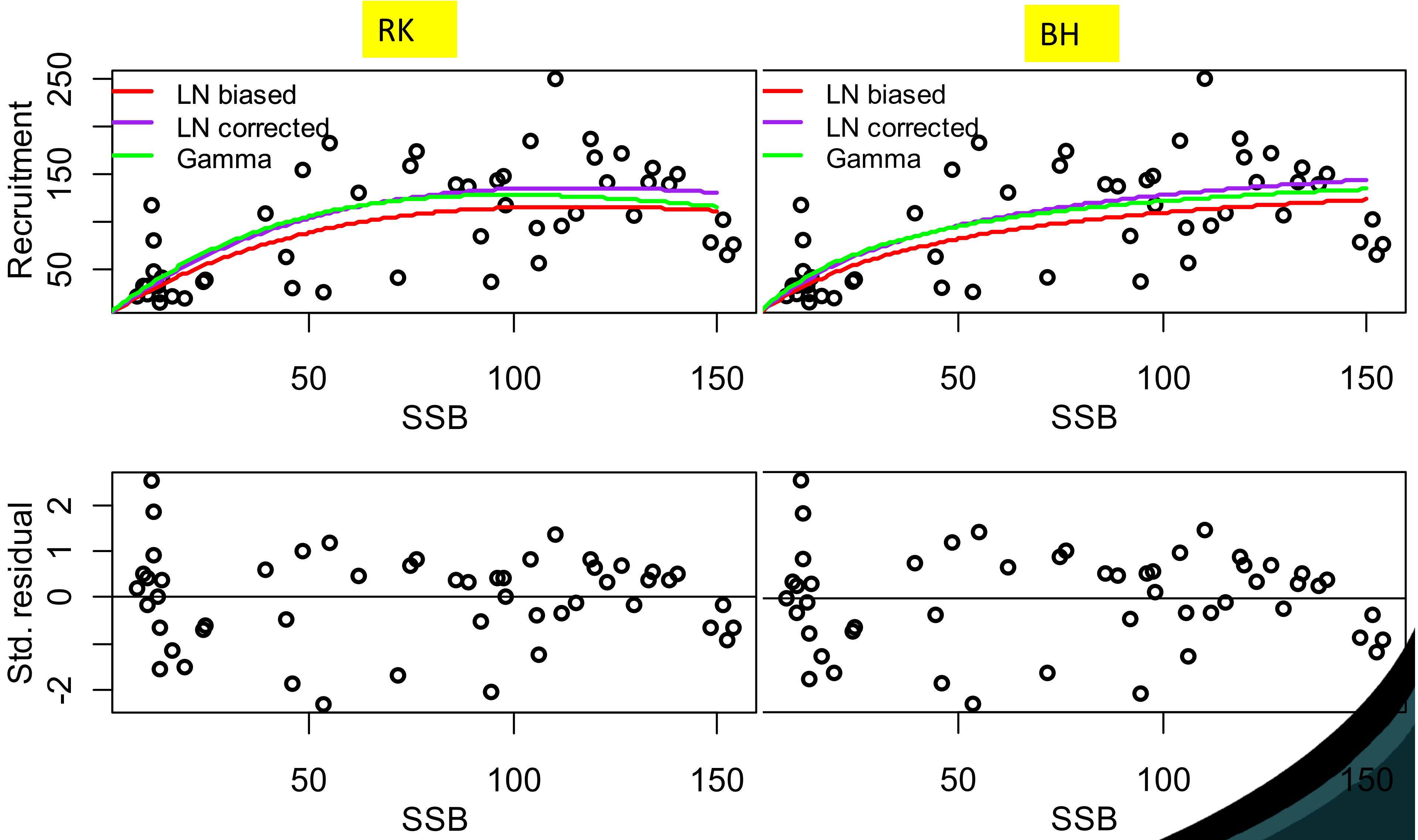
```
> confint(RK.fit)
Waiting for profiling to be done...
              2.5%        97.5%
Beta   0.005442704 0.01195957
alpha 2.036531567 3.65798284
```

About the same as BH

BH & RK Predict for 4VsW cod



BH & RK Predict and residuals



Ricker LN Estimation

- How can we get confidence intervals for S50 and Rmax?
- Recall the estimate of Rmax is

$$R_{\max} = \frac{\hat{\alpha}}{\hat{\beta}\exp(1)}$$

- and S50 is the "lower" solution to

$$R(S50) = 0.5R_{\max}$$

Ricker LN Bootstrap

- We can easily get the bootstrap distribution for Rmax, and with a little work we can get it for S50

```
boo.Rmax = boo$coef[,2]/(boo$coef[,1]*exp(1));
```

Ricker LN Bootstrap

```
ricker <-function(parm,s){
  alpha <- parm[2]
  beta <- parm[1];
  ret = alpha*s*exp(-beta*s);
  return(ret)
}
```

```
fS50 <- function(parm){
  hRmax = 0.5*parm[2]/(parm[1]*exp(1));
  temp = uniroot(function(x) hRmax-ricker(parm,x),
        lower = 0, upper = 1/parm[1])
  return(temp$root)
}
```

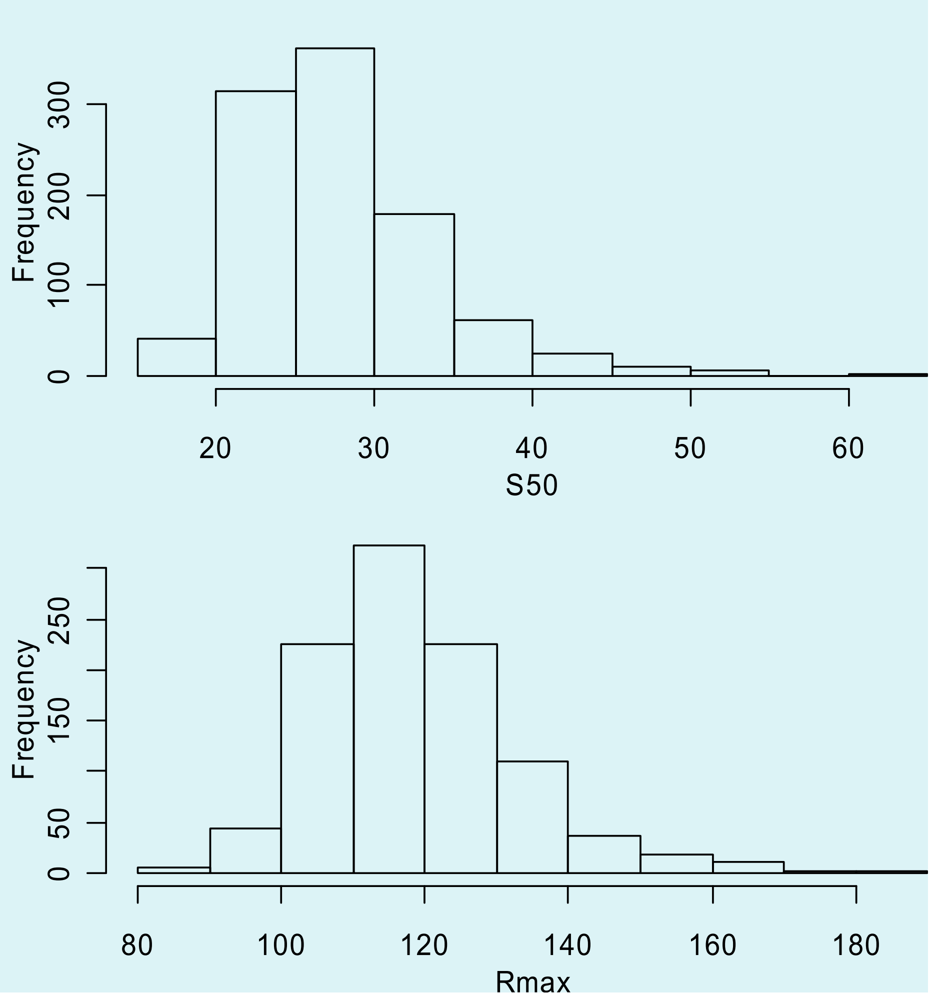
Smax=1/β

```
boo.S50 = apply(boo$coef,1,fS50);
```

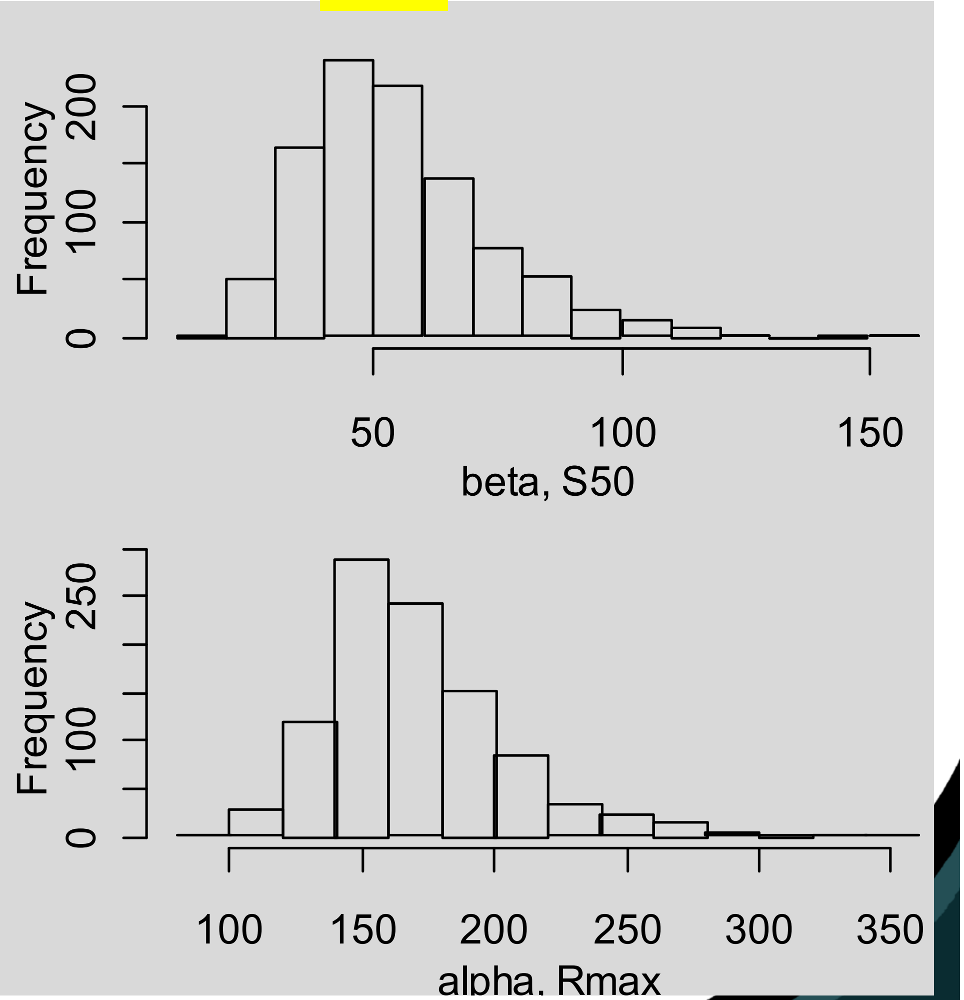
```
new.boo = boo
new.boo$coef =cbind(boo$coef,boo.Rmax,boo.S50)
colnames(new.boo$coef) = c(colnames(boo$coef),"Rmax","S50")
qRmax = quantile(boo.Rmax,probs=c(0.5,0.025,0.975))
qS50 = quantile(boo.S50,probs=c(0.5,0.025,0.975))
new.boo$bootCI = rbind(boo$bootCI,qRmax,qS50)
rownames(new.boo$bootCI) = c(rownames(boo$bootCI),"Rmax","S50")
```

Compare bootstrap for 4VsW cod

RK



BH



Compare Bootstrap CI’s for 4VsW cod

RK

```
------
Bootstrap estimates
      beta            alpha              Rmax         S50
8.638431e-03 2.740930e+00 1.169061e+02 2.685222e+01

------
Bootstrap confidence intervals
                2.5%             97.5%
beta       0.005426038     0.01193137
alpha     2.072166053     3.58811253
Rmax  97.561211411 152.39662878
S50     19.441276567   42.74962801
```

BH

```
------
Bootstrap estimates
    beta        alpha
 51.83972 164.71769

------
Bootstrap confidence intervals
               2.5%      97.5%
beta     25.95528  105.5379
alpha 119.10793  257.6426
```

S50 → beta

Rmax → alpha

S50 and Rmax are quite different for BH and RK SR models

Other formulations

- There are other formulations of SR models, and many are described in the Needle paper I sent.
- The “steepness” formulation described in Haddon is commonly used in some regions (US, NZ, AUS, …)
- A segmented regression (“Hockey-stick”) model is sometimes used
- And some nonparametric approaches

Hockey-stick model

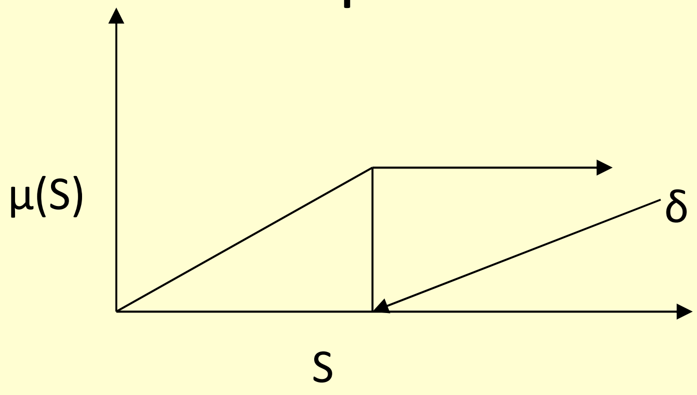
- The "Hockey-stick" (HS) model is sometimes used

$$\mu(S) = \begin{cases} \alpha S, & S < \delta \\ \alpha\delta, & S \geq \delta \end{cases}$$

- A segmented regression model

- There is specialized software to get mle's

- because this model is not smooth and parameter-differentiable everywhere



Smooth HS

- A slightly smoothed version was described in Mesnil and Rochet (2010)
- $\mu(s) = \alpha\left[s + \sqrt{\delta^2 + \gamma^2/4} - \sqrt{(s-\delta)^2 + \gamma^2/4}\right]$
- where $\gamma^2 = 0.1$.
- Can show that

  Sao = 2α, Rmax $\cong 2\alpha\delta$, and $S_{50\%}$ = δ/2.
- γ is fixed (not a parameter) and governs the smoothness of the curve at the break point

LN estimation of HS

```
Formula: logrec ~ log(alpha) + log(ssb + sqrt(delta^2 + gam2by4) - sqrt((ssb -
    delta)^2 + gam2by4))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
alpha    1.215      0.187    6.51  3.9e-08 ***
delta   45.156      8.928    5.06  6.3e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.583 on 49 degrees of freedom

Algorithm "port", convergence message: both X-convergence and relative
convergence (5)
```

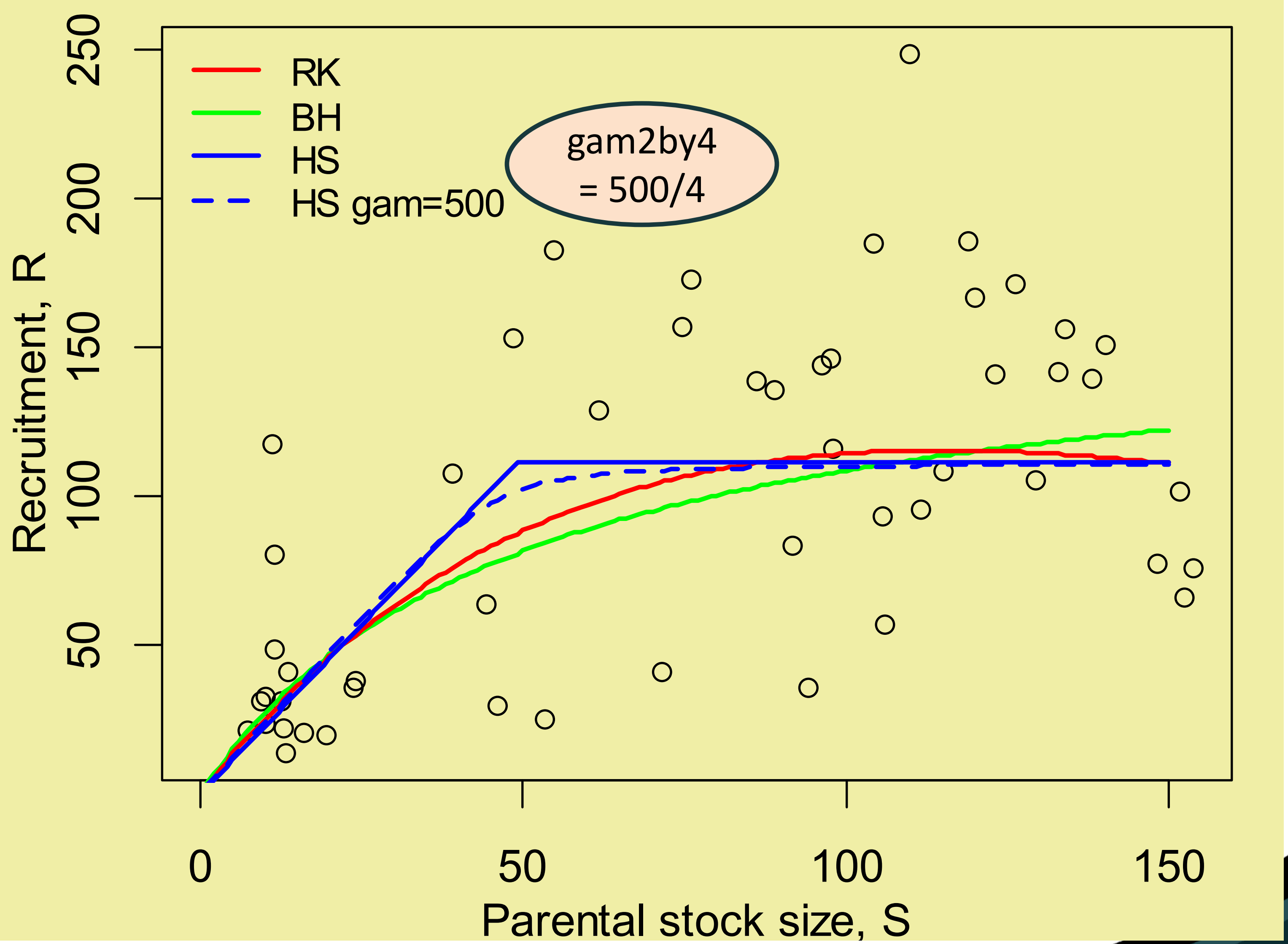
```
init.alpha= 0.5*init.Rmax/init.S50
init.delta = init.S50
gam2by4 = 0.1/4
```

a little higher than BH/RK

Waiting for profiling to be done...

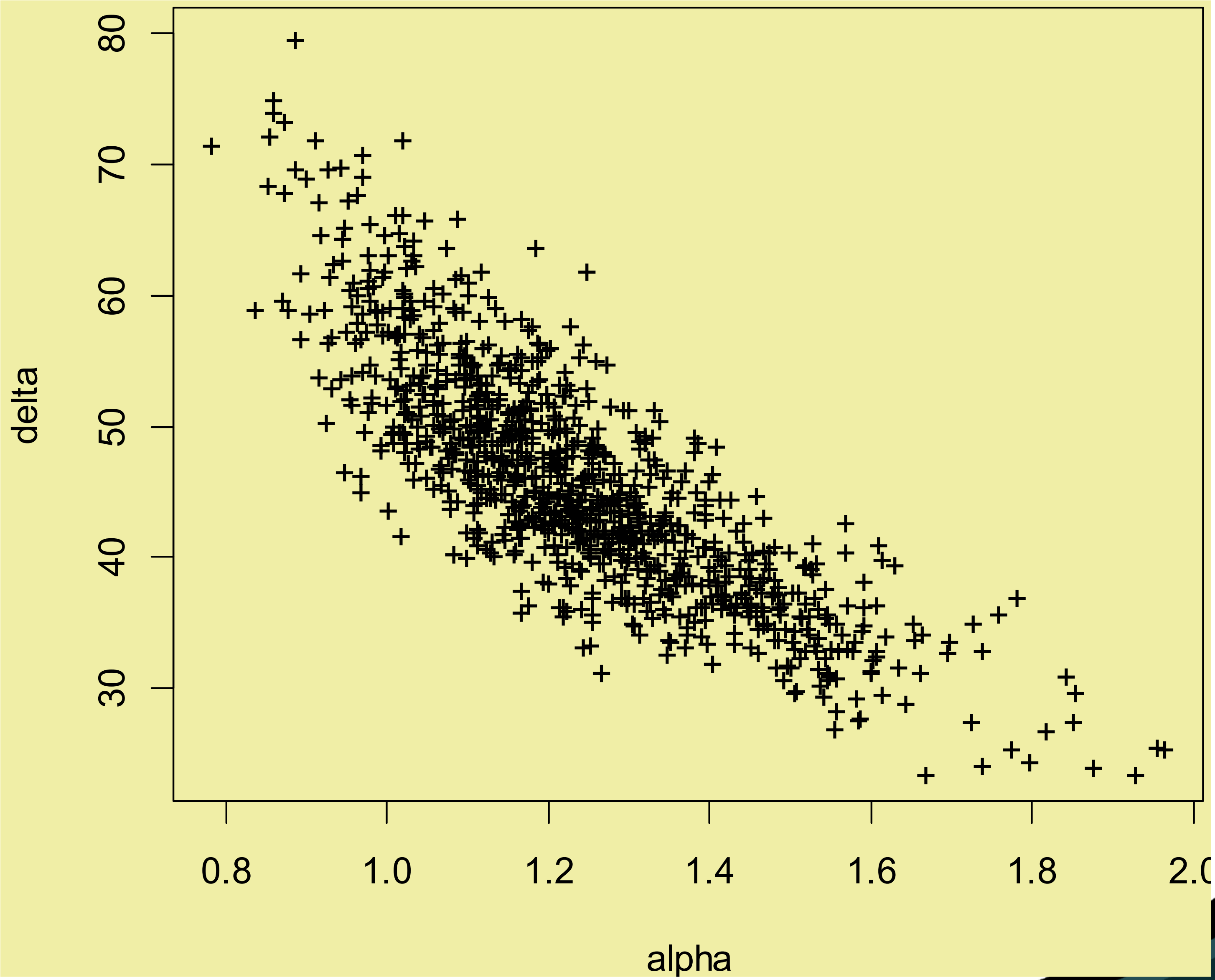
```
> HS.profile
        2.5% 97.5%
alpha  0.869  1.78
delta 27.340 72.14
```

BH, RK, & HS Predict for 4VsW cod



gam2by4
= 500/4

HS bootstrap



Nonparametric SR for MSY RP’s

- SR data are usually noisy and different models often fit the data almost equally as well.
- different models may have substantial impact on MSY reference points (RP's).
- Characterizing this uncertainty on MSY RP’s is important.

Cadigan, N.G., 2012. Fitting a non-parametric stock–recruitment model in R that is useful for deriving MSY reference points and accounting for model uncertainty. *ICES Journal of Marine Science*, *70*(1), pp.56-67.

SR models and MSY RP’s

- Most SR models have a compensatory mortality property (CMP) in which the recruitment rate (*R/S*) declines monotonically as *S* increases; that is

$$\frac{d\{\mu(s)/s\}}{ds} < 0 \text{ for all } s$$

- CMP ensures that for any level of *F* there is a unique equilibrium stock size in projections.
- CMP ensures the equilibrium yield curve (i.e. Catch versus F) is continuous with a well-defined maximum at Fmsy

Determining MSY RP's

- Basing RP's on a nonparametric SR model may more fully account for model uncertainty.
- But too much flexibility can cause difficulties
- I used the *scam* (shape constrained additive models) function in R (Pya, 2012) to fit nonparametric SR curves with CMP, and to derive "SR-nonparametric" yield curves.
- Details about *scam* are provided in the appendix

A nonparametric SR model using Scam

- R code: *scam(log.recruit ~ s(stock.size,bs="mpd") + offset(log(stock.size)),data=dat)*
- mpd: monotonic decreasing P-splines.
- The *s(.)* term represents $\log\{\mu(s)/s\}$ which is a monotonic decreasing function of *S*.
- The offset term is added to *s(.)* so that the right-hand side of the *scam* equation represents $\log\{\mu(s)\}$.

log(R/S) = g(S) => log(R) = g(S) + log(S)

log(S) – offset, like an effect with a slope fixed at 1

Over-smoothed SCAM

```
> sr.data$offset = log(sr.data$ssb)
> scam.fit <- scam(logrec ~ s(ssb,k=5,bs="mpd",m=2) +
offset(offset),
+
family=gaussian(link="identity"),data=sr.data,sp=0.000001)
> summary(scam.fit)
```

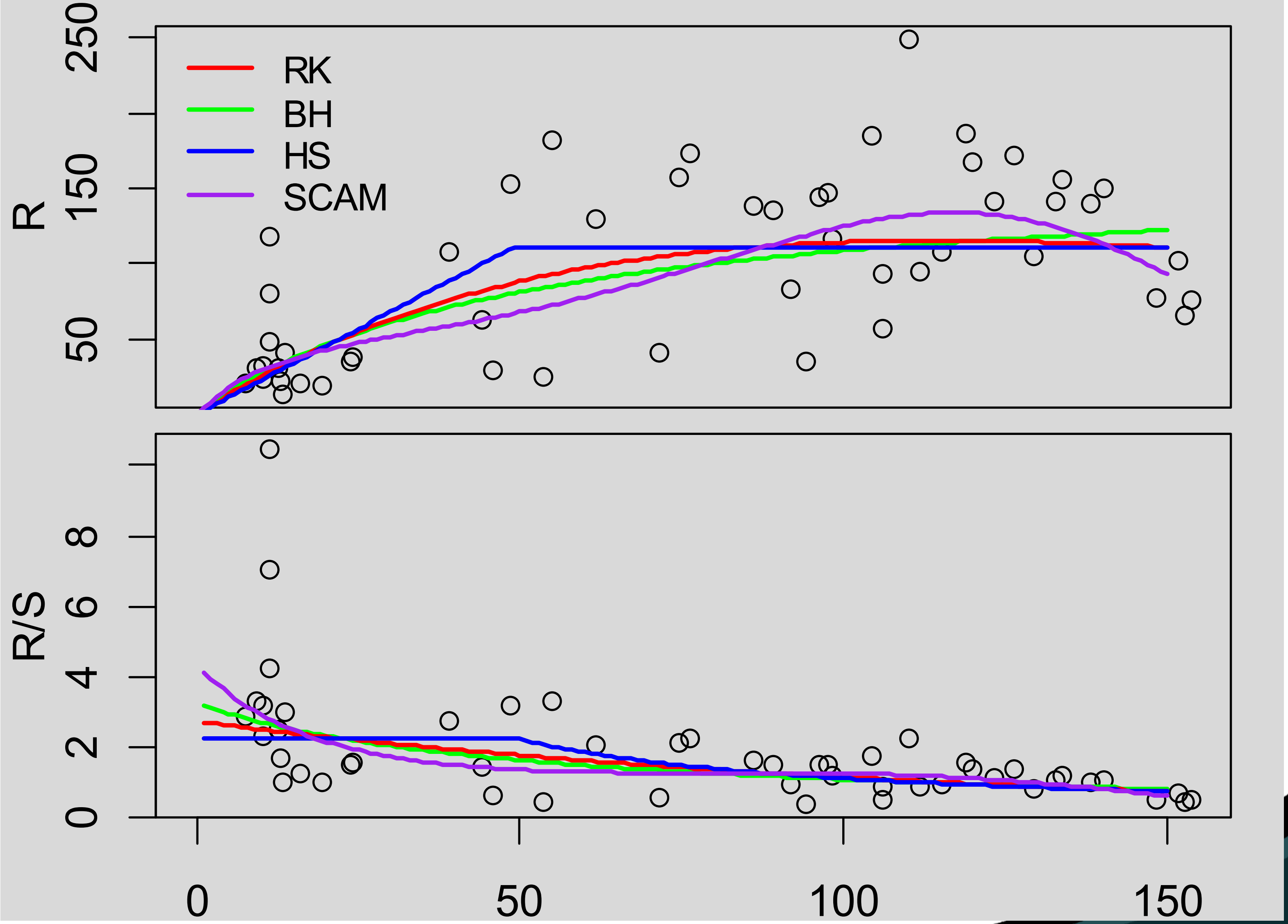
```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.81       1.34    4.35    7e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
        edf Ref.df     F p-value
s(ssb) 2.02   2.04 17.7   1e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.5107   Deviance explained =   53%
GCV score =   Scale est. = 0.29511    n = 51
```

Smoothing parameters

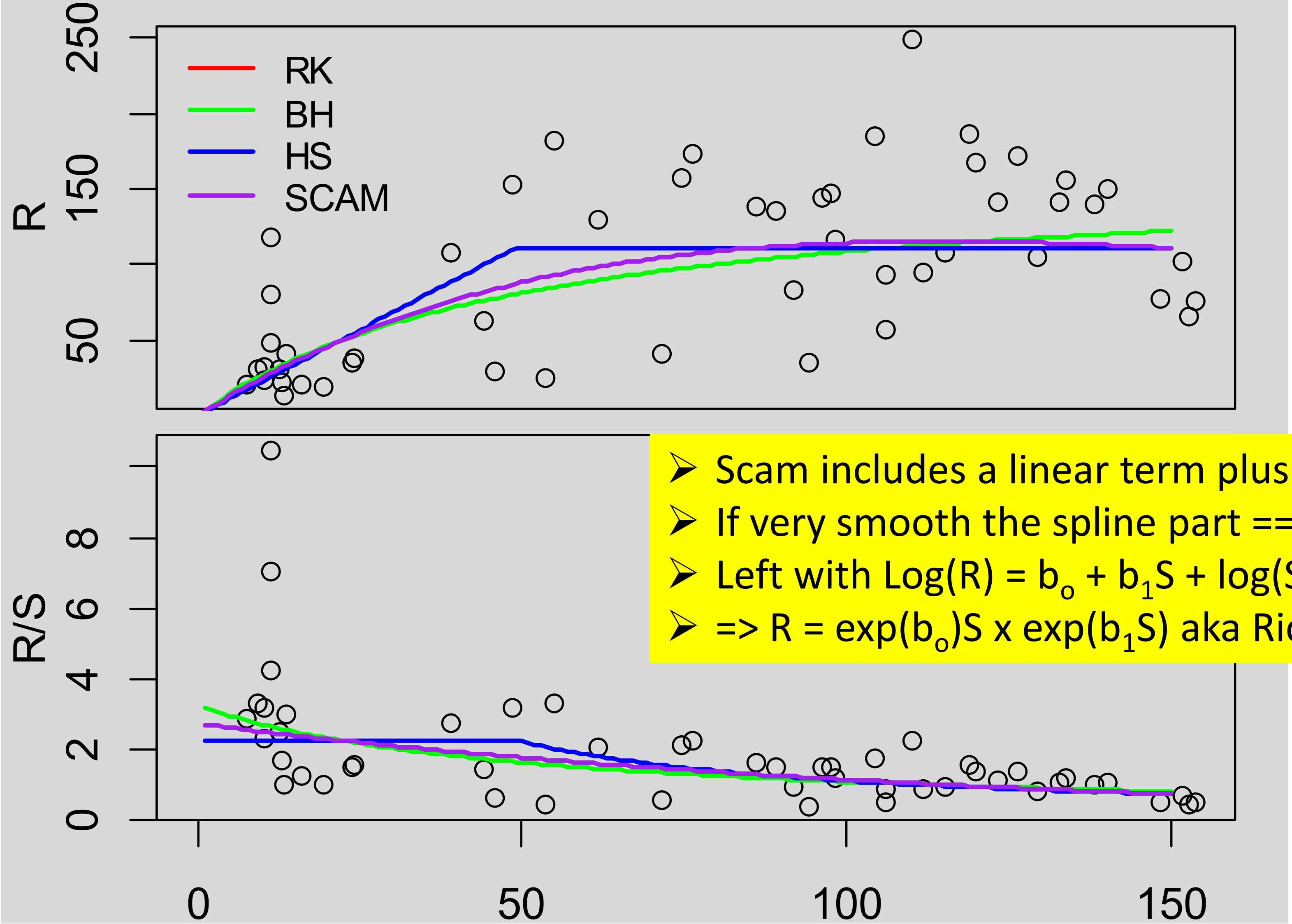
Sp=1e-6 small and results in “over-smoothing”

sp=NULL indicates that smoothing parameters should be estimated

Over-smoothed SCAM



Default-smoothed SCAM



- ➢ Scam includes a linear term plus spline part
- ➢ If very smooth the spline part ==0 (f(x)=0)
- ➢ Left with Log(R) = b_o + b_1S + log(S)
- ➢ => R = exp(b_o)S x exp(b_1S) aka Ricker model

Default-smoothed SCAM

```
> scam.fit <- scam(logrec ~ s(ssb,k=5,bs="mpd",m=2) + offset(offset),
+        family=gaussian(link="identity"),data=sr.data)
> summary(scam.fit)
```

```
Parametric coefficients:
           Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.577      0.243     6.5    4e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
       edf Ref.df    F p-value
s(ssb)   1      1 28.7 1.7e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.4724   Deviance explained = 48.3%
GCV score = 0.33137  Scale est. = 0.31837    n = 51

BFGS termination condition:
1.89e-05
```

Just linear part

BH scam model

- For the BH, $\mu(s) = {}^{\alpha s}/_{\beta+s} \Longrightarrow \mu^{-1}(s) = \frac{\beta}{\alpha} s^{-1} + \alpha^{-1}$
- A scam() formulation that is BH when the spline part is zero is:

```
sr.data$ssb_inv = 1/sr.data$ssb
scamBH.fit <- scam(rec ~ s(ssb_inv,k=5,bs="mpi",m=2),
      family=Gamma(link="inverse"),data=sr.data)
summary(scamBH.fit)
scamBH.predict = predict(scamBH.fit,list(ssb_inv=1/ssb.pred),type = "response")
```

- Gamma BH:

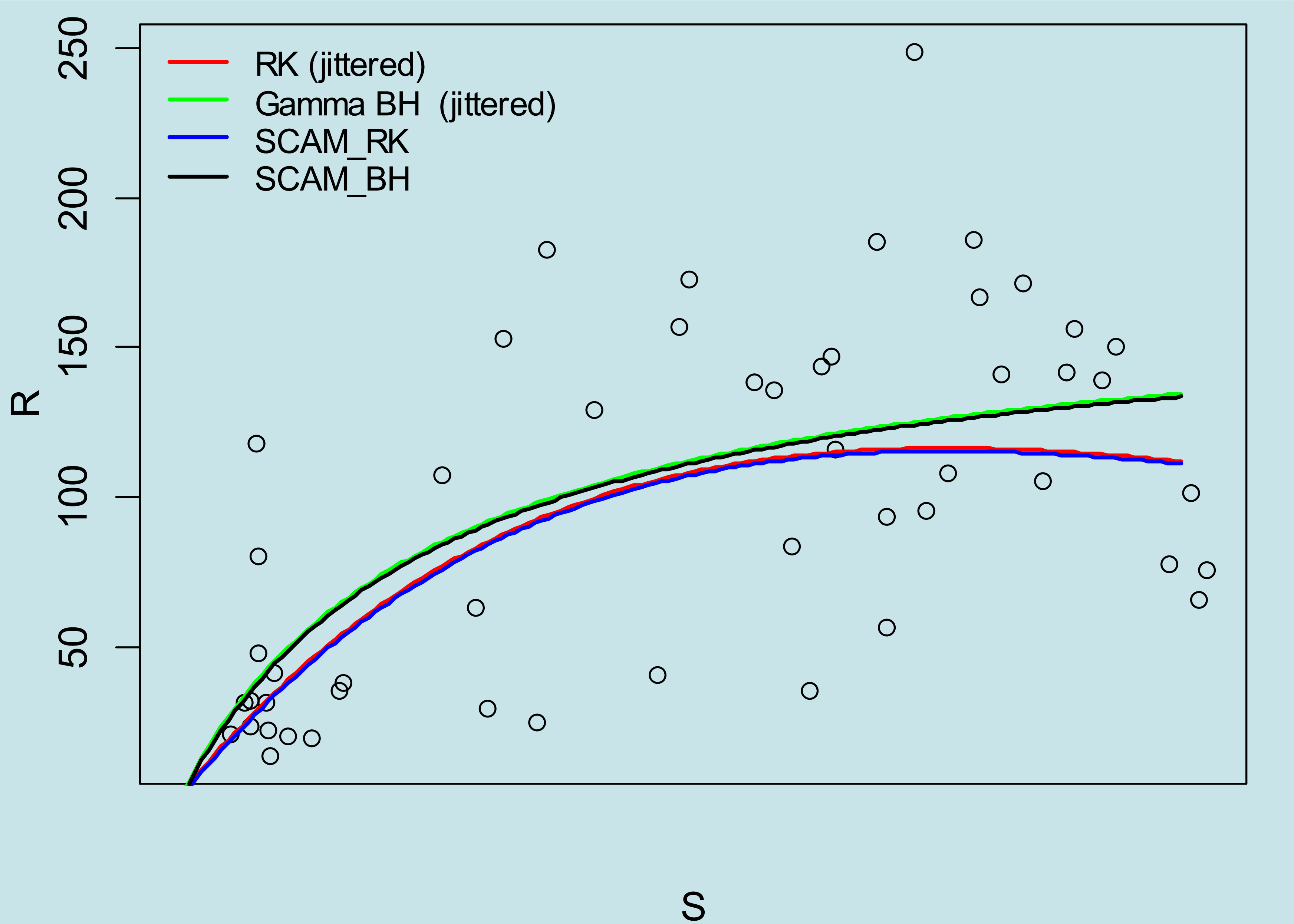
```
BH.fit <- glm(rec ~ ssb_inv, family = Gamma(link="inverse"), data = sr.data)
summary(BH.fit)
BH.predict = predict(BH.fit,list(ssb_inv=1/ssb.pred),type = "response")
```

BH scam model

```
> summary(scamBH.fit)
Family: Gamma
Link function: inverse

Formula:
rec ~ s(ssb_inv, k = 5, bs = "mpi", m = 2)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.00753    0.00366   -2.05    0.045 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0

Approximate significance of smooth terms:
           edf Ref.df    F p-value
s(ssb_inv)   1      1 21.6 2.3e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.3957   Deviance explained =   42%
GCV score = 0.3123  Scale est. = 0.30005    n = 51
```

```
>  summary(BH.fit)
Call:
glm(formula = rec ~ ssb_inv, family = Gamma(link =
"inverse"), data = sr.data)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.00594    0.00101    5.87  3.8e-07 ***
ssb_inv      0.23270    0.05033    4.62  2.8e-05 ***
---
(Dispersion parameter for Gamma family taken to be 0.303)

Null deviance: 25.330  on 50  degrees of freedom
Residual deviance: 14.703  on 49  degrees of freedom
AIC: 530
```

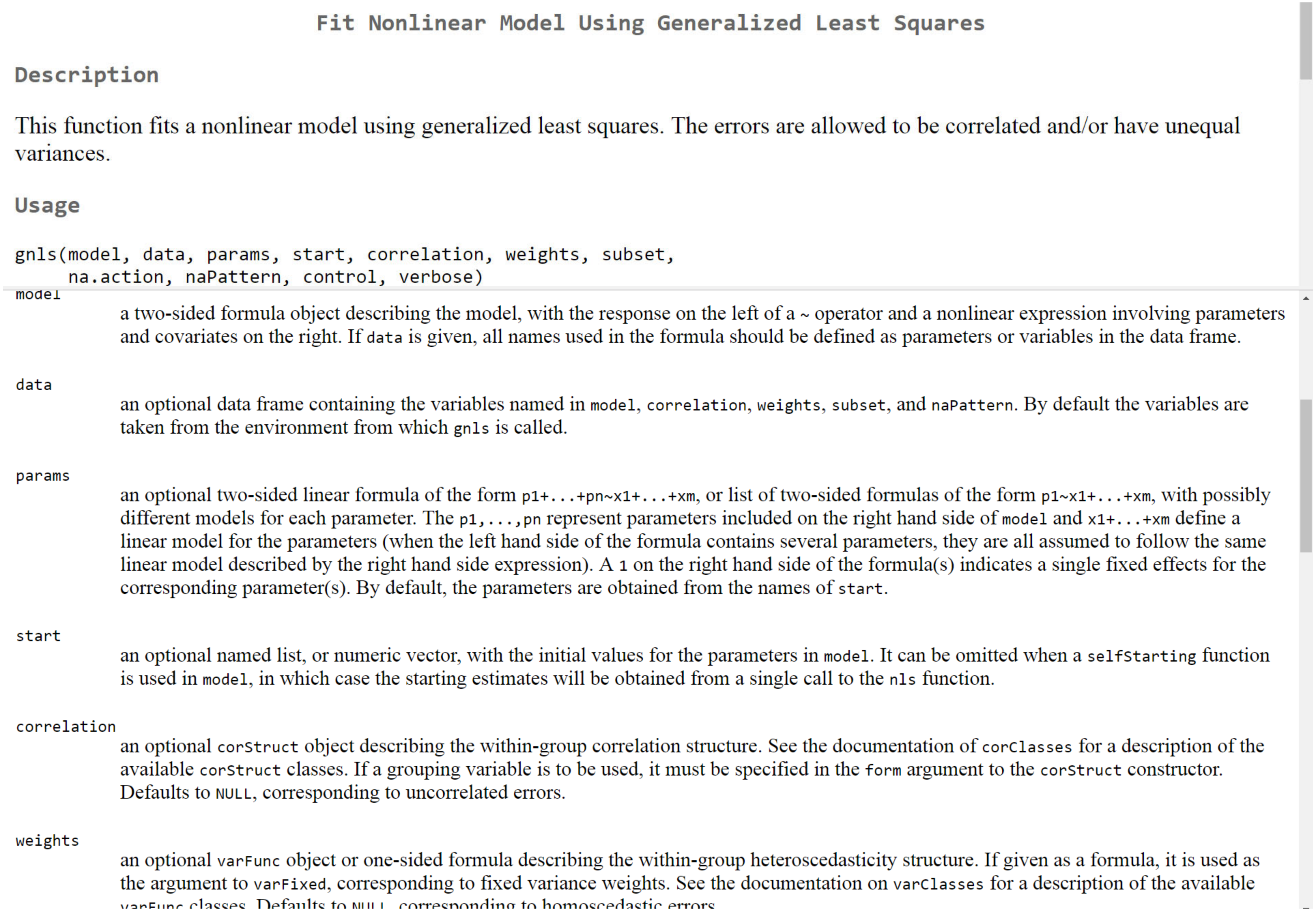
BH scam model



The scam model

- This nonparametric scam() SR model is not a major improvement
- Predictions can be too strange
- There was some interest in this at first, but this has waned (only 21 cites for a 2013ish paper) – not many people use this
- Most of the reasons I went into detail was to provide information on shape constrained smoothing spline models
- Which could be useful for growth curves, maturity curves, selectivity, catchability, etc.
- More important to account for temporal non-stationarity in the stock-recruit relationship:

Autocorrelated Errors



Fit Nonlinear Model Using Generalized Least Squares

Description

This function fits a nonlinear model using generalized least squares. The errors are allowed to be correlated and/or have unequal variances.

Usage

```
gnls(model, data, params, start, correlation, weights, subset,
     na.action, naPattern, control, verbose)
```

`model`
: a two-sided formula object describing the model, with the response on the left of a ~ operator and a nonlinear expression involving parameters and covariates on the right. If `data` is given, all names used in the formula should be defined as parameters or variables in the data frame.

`data`
: an optional data frame containing the variables named in `model`, `correlation`, `weights`, `subset`, and `naPattern`. By default the variables are taken from the environment from which `gnls` is called.

`params`
: an optional two-sided linear formula of the form `p1+...+pn~x1+...+xm`, or list of two-sided formulas of the form `p1~x1+...+xm`, with possibly different models for each parameter. The `p1,...,pn` represent parameters included on the right hand side of `model` and `x1+...+xm` define a linear model for the parameters (when the left hand side of the formula contains several parameters, they are all assumed to follow the same linear model described by the right hand side expression). A `1` on the right hand side of the formula(s) indicates a single fixed effects for the corresponding parameter(s). By default, the parameters are obtained from the names of `start`.

`start`
: an optional named list, or numeric vector, with the initial values for the parameters in `model`. It can be omitted when a `selfStarting` function is used in `model`, in which case the starting estimates will be obtained from a single call to the `nls` function.

`correlation`
: an optional `corStruct` object describing the within-group correlation structure. See the documentation of `corClasses` for a description of the available `corStruct` classes. If a grouping variable is to be used, it must be specified in the `form` argument to the `corStruct` constructor. Defaults to `NULL`, corresponding to uncorrelated errors.

`weights`
: an optional `varFunc` object or one-sided formula describing the within-group heteroscedasticity structure. If given as a formula, it is used as the argument to `varFixed`, corresponding to fixed variance weights. See the documentation on `varClasses` for a description of the available `varFunc` classes. Defaults to `NULL`, corresponding to homoscedastic errors.

Autocorrelated Errors

```
>   BH.arfit <- gnls(logrec ~ log(alpha) + logssb - log(beta + ssb),data=sr.data,
+           start = list(beta = init.beta,alpha = init.alpha),
+           correlation = corAR1(form=~year))
>   summary(BH.arfit)
```

```
Generalized nonlinear least squares fit
Model: logrec ~ log(alpha) + logssb - log(beta
+ ssb)
Data: sr.data
AIC BIC logLik
67.7 75.4 -29.8

Correlation Structure: AR(1)
Formula: ~year
Parameter estimate(s):
Phi
0.632

Coefficients:
Value Std.Error t-value p-value
beta     38.7 25.7 1.50 0.1389
alpha 139.9 45.4 3.08 0.0034
```

```
> AIC(BH.fit)
[1] 89.8
```

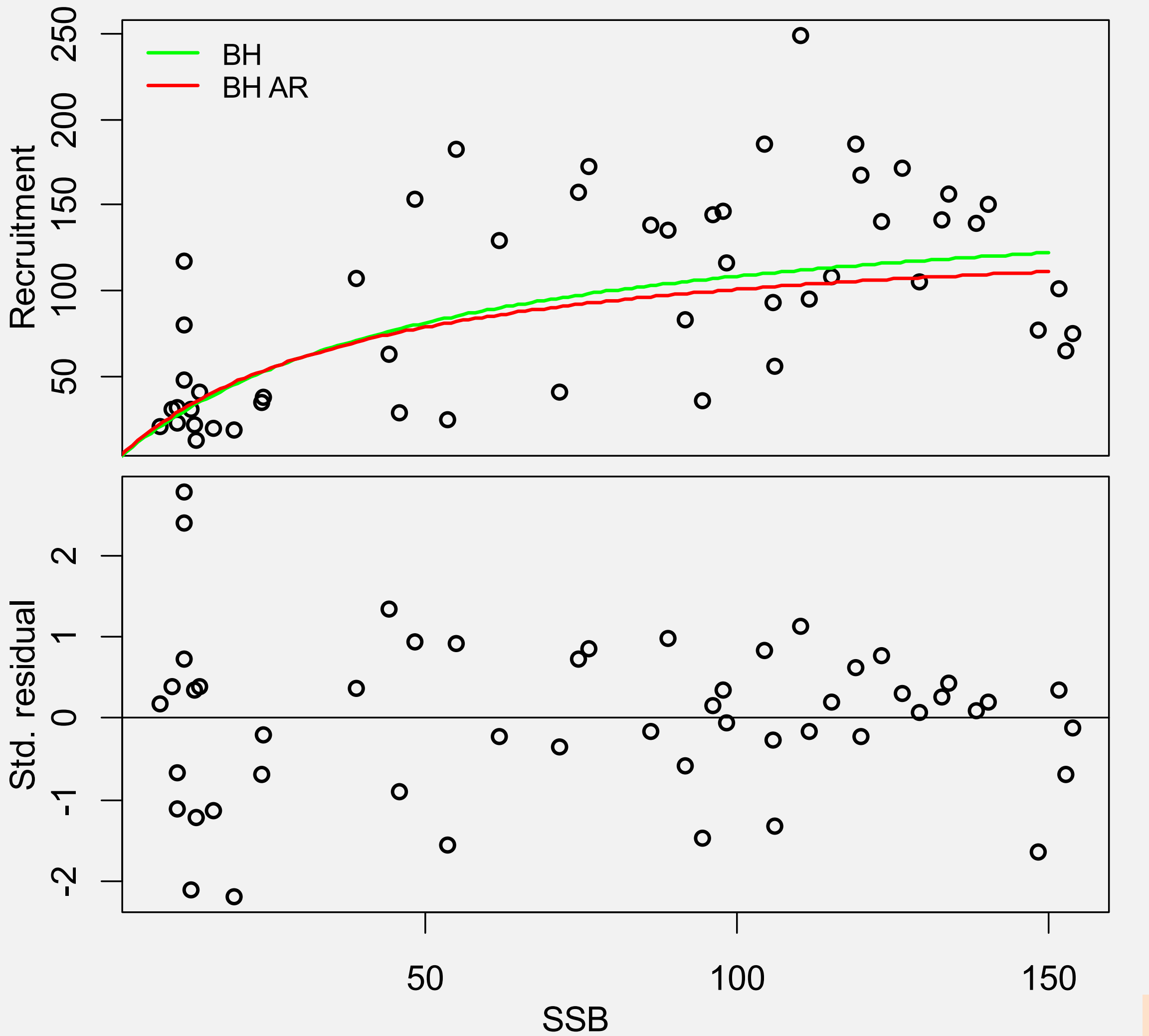
```
> logLik(BH.fit)
'log Lik.' -41.9
(df=3)
```

```
>   coef(BH.fit)
beta alpha
50.3 163.3
```

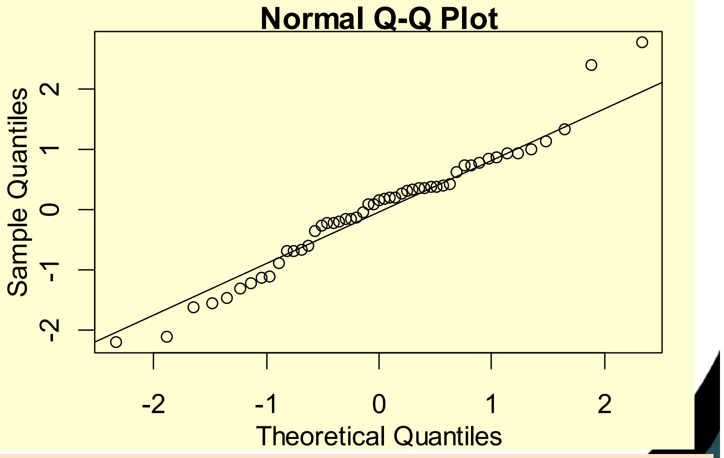
```
Correlation:
beta
alpha 0.865

Standardized residuals:
Min Q1 Med Q3 Max
-2.092 -0.619 0.193 0.716 2.314

Residual standard error: 0.569
Degrees of freedom: 51 total; 49 residual
```

BH.fit: Residual standard error: 0.562 on 49 degrees of freedom

Autocorrelated Errors

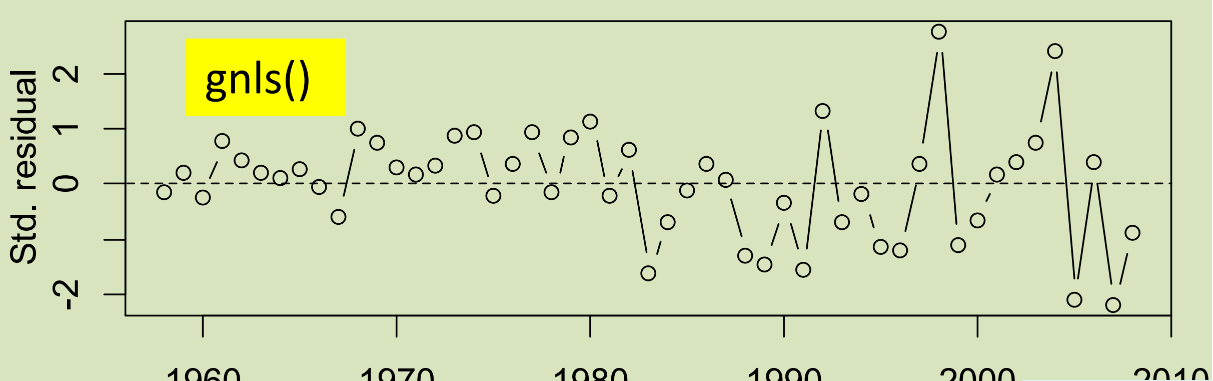


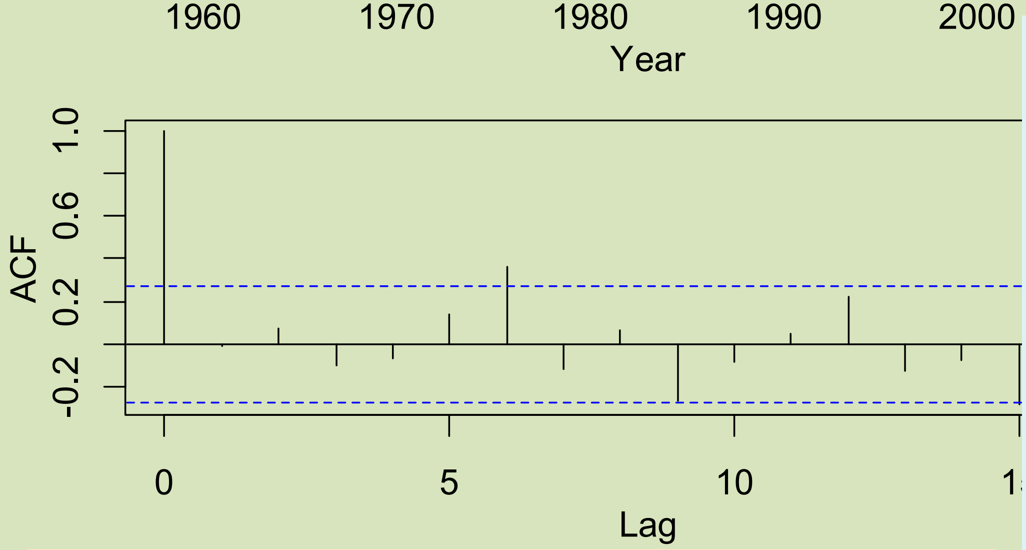
```
> shapiro.test(resid.s)

        Shapiro-Wilk normality test

data:  resid.s
W = 1, p-value = 0.2
```



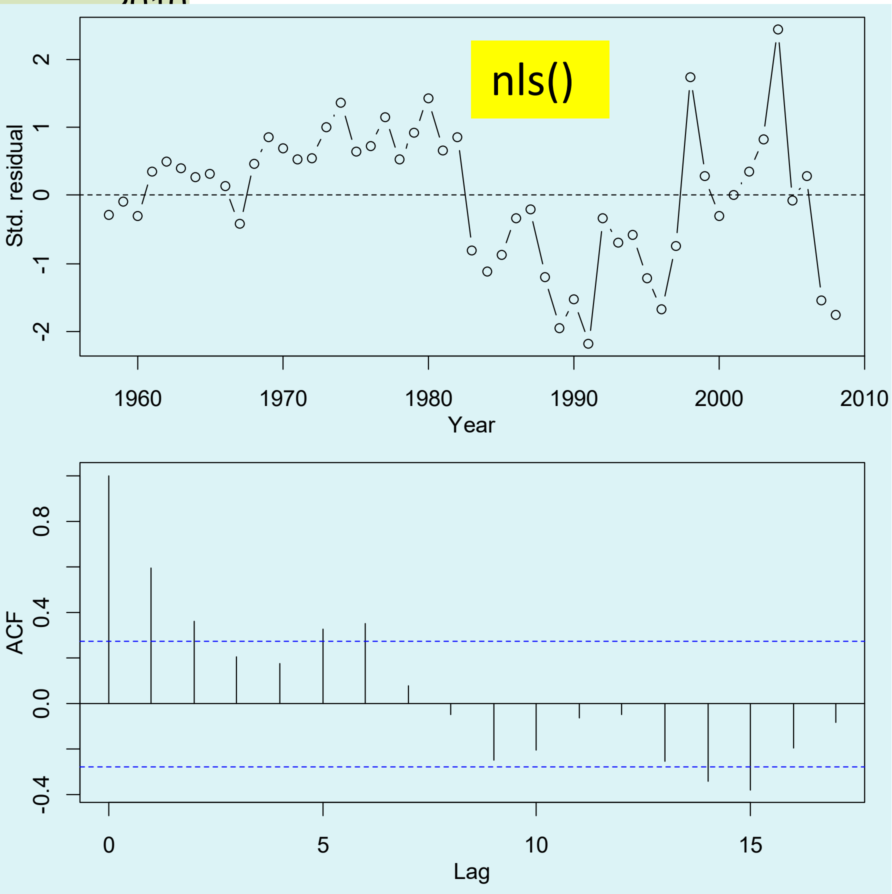
```
> par(mar=c(3,3,1,1),mgp=c(2,1,0))
> qqnorm(resid.s)
> qqline(resid.s)
```

Autocorrelated Errors





```
Standard Normal = -4.3583,
p-value = 1.31e-05
alternative hypothesis:
two.sided
```



```
> library(randtests) ; runs.test(resid.s)
      Runs Test
data: resid.s
statistic = -0.3, runs = 20, n1 = 20, n2 = 20, n =
50, p-value = 0.8
alternative hypothesis: nonrandomness
```

Autocorrelated Errors

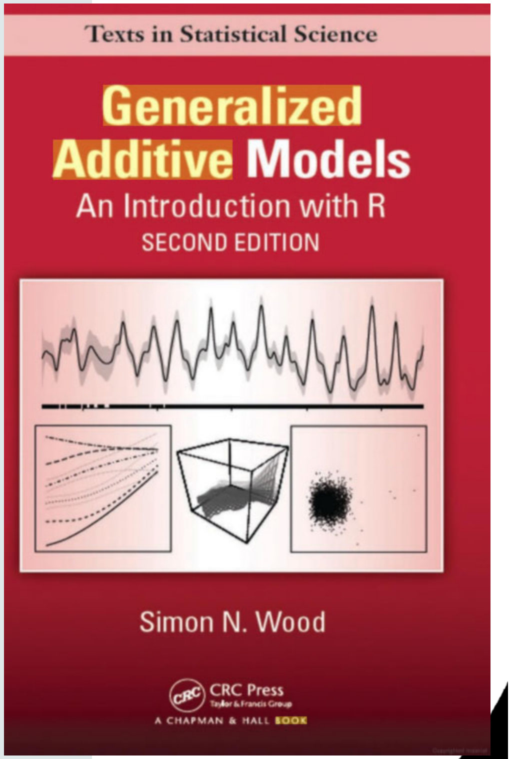
- We could use the autocorrelation to improve short-term recruitment predictions
- Hope to do this in F6005
- But this is far enough for F6004

Appendix: The scam package

- The *scam* package is similar to the *mgcv*(*gam)* package (Wood, 2006, 2011) for fitting GAMs (generalized additive models),

- except that *scam* allows for a variety of shape constraints on the component functions of the linear predictor of the GAM.

- The constraints involve monotonicity (increasing or decreasing) with options to specify convex or concave shapes



The scam model

- use spline smoothers for nonparametric regression,
- data are assumed to be independent and from an exponential family distribution
- with mean μ that is a partially linear function of covariates

The scam model

- $g(\mu) = \theta_o + \sum_{k=1}^{p_l} \theta_k x_k^* + \sum_{j=1}^{p_n} f_j(x_j)$
- g is a one-to-one link function
- the $f_j(x_j)$'s are unknown smooth functions of the x_j covariates
- θ_k's and $f_j(x_j)$'s must be estimated
- In our SR case $p_l = 0\ and\ p_n = 1$ and $g(\mu) = \theta_o + f(x)$

The scam model

- spline approximation: $f(x) = \sum_{i=1}^{q} B_i^m(x)\,\gamma_i$
- $\gamma_1,\dots,\gamma_q$ are unknown spline parameters to estimate
- $B_1^m(x)$,…,$B_q^m(x)$ are known spline basis functions
- *scam* package uses B-spline basis functions
- knots determine the location and shape of the polynomials

The scam model

- *scam* package assumes the knots are evenly spaced and cover the range of *x*
- The shape constraints are based on evenly spaced knots, which would be otherwise tricky to specify.
- number of knots (q) should be large to avoid over-smoothing/under-fitting; however,
- this could result in over-fitting of the data

The scam model

- A penalty function is used to control the variation in the *γ*‘s
- A smoothing parameter determines the contribution of the smoothing penalty function to the total fit function
- *scam* uses generalized cross-validation or the Akaike information criterion to determine the value of the smoothing parameter.
- P-splines are penalized B-splines

The scam model

- *scam* uses various shape restrictions on the γ spline parameters to ensure the correct shape restrictions on *f*
- For example, say *f(x)* is strictly monotone increasing in *x.*
- since all B-spline basis functions are nonnegative, then a sufficient condition for this shape constraint (i.e. monotone increasing) is $\gamma_i > \gamma_{i-1}$

The scam model

- achieved by redefining the γ parameters as
- $\gamma_1 = \beta_1, \gamma_j = \beta_1 + \sum_{i=2}^{j} \exp(\beta_i),$ for $j = 2,\ldots,q$.
- β_i's are unconstrained parameters to estimate.
- The penalty function is based on the squared differences of $\beta_2,\ldots\ \beta_q$.
- However, the shape restrictions themselves add much smoothing and
- model results are usually not that sensitive to the choice of smoothing parameter.