Приложение 7

## Ортогональная регрессия по западной части

Загрузим данные.

pinks <- readxl::read\_excel('data.xlsx', sheet = 'Запад')  
(pinks <- transform(pinks, S=exp(LnS), R=exp(LnR)))

## LnR LnS F1 F2 F3 F4 S R  
## 1 2.8617902 2.6764735 -0.58 -1.15 -0.31 0.80 14.533750 17.49281481  
## 2 -0.4993485 0.3504456 0.85 1.19 -0.94 0.54 1.419700 0.60692593  
## 3 4.6370578 1.8988185 1.96 0.52 0.17 -0.39 6.678000 103.24014815  
## 4 -0.5653405 -0.7318880 -0.88 2.59 0.38 0.83 0.481000 0.56816667  
## 5 4.3870833 4.3955843 -0.11 -0.74 0.62 0.54 81.092000 80.40555556  
## 6 -0.1425988 -0.6901517 -0.66 -1.18 0.38 0.13 0.501500 0.86710185  
## 7 4.8550929 3.8605192 2.27 -0.34 -0.11 0.12 47.490000 128.39262407  
## 8 -2.5181073 -0.3926723 -0.87 -0.02 1.37 0.65 0.675250 0.08061204  
## 9 4.4318234 3.7404883 -1.84 1.12 -0.89 -0.67 42.118550 84.08460000  
## 10 0.3126565 -3.1332717 -1.68 -0.86 0.22 0.14 0.043575 1.36705185  
## 11 4.4226264 3.0428353 -1.68 -0.27 -0.73 0.52 20.964600 83.31481481  
## 12 2.4359368 0.1949087 0.11 -1.64 -0.60 -0.23 1.215200 11.42651852  
## 13 4.5195526 3.8329798 0.63 -0.90 0.37 -0.28 46.200000 91.79451852  
## 14 3.1771537 2.4148418 0.24 -0.03 -0.10 -0.72 11.188000 23.97840741  
## 15 4.2104648 3.9461539 -0.42 0.35 0.75 0.03 51.736000 67.38785185  
## 16 2.7206985 2.8741859 -0.85 -0.16 0.98 -0.27 17.711000 15.19092963  
## 17 4.2696263 3.4338582 0.09 0.02 0.12 -0.03 30.996000 71.49490985  
## 18 1.4475568 1.2923408 -1.85 -1.32 1.70 -0.08 3.641300 4.25271151  
## 19 4.6493878 3.6622531 -0.49 0.66 0.72 -0.05 38.949000 104.52097485  
## 20 1.6198396 -2.1336866 -2.21 0.41 -0.02 -0.12 0.118400 5.05227972  
## 21 5.0660509 3.8379889 -2.37 -0.51 0.41 -1.06 46.432000 158.54697436  
## 22 0.4541537 -0.1839228 -2.60 -0.07 0.62 0.01 0.832000 1.57484001  
## 23 2.0373166 3.3571758 -1.56 -0.17 1.59 0.15 28.708000 7.67000000  
## 24 1.5260563 -1.5575573 0.34 -0.77 1.47 -0.37 0.210650 4.60000000  
## 25 4.2654928 1.0986123 0.98 -1.50 0.17 -0.69 3.000000 71.20000000  
## 26 2.4647039 1.0473190 -0.63 -0.37 1.54 0.47 2.850000 11.76000000  
## 27 5.8579332 2.9957323 -0.62 -1.38 1.71 0.15 20.000000 350.00000000  
## 28 4.1269731 1.4350845 -0.24 -1.38 0.82 0.84 4.200000 61.99000000  
## 29 4.6128407 4.7238417 0.01 -2.19 0.70 -0.72 112.600000 100.77000000  
## 30 5.4337220 3.0189604 -1.33 -0.21 0.38 -0.38 20.470000 229.00000000  
## 31 3.1876328 3.1135153 -0.94 -1.94 2.23 -0.21 22.500000 24.23100000  
## 32 5.3717769 4.5185224 -2.38 -0.41 0.97 -0.17 91.700000 215.24500000

Перейдём к зависимости по Рикеру в пакете FSA.

library(FSA)

## ## FSA v0.9.5. See citation('FSA') if used in publication.  
## ## Run fishR() for related website and fishR('IFAR') for related book.

citation('FSA')

## To cite package 'FSA' in publications use:  
##   
## Ogle DH, Doll JC, Wheeler AP, Dinno A (2023). \_FSA: Simple Fisheries  
## Stock Assessment Methods\_. R package version 0.9.5,  
## <https://CRAN.R-project.org/package=FSA>.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {FSA: Simple Fisheries Stock Assessment Methods},  
## author = {Derek H. Ogle and Jason C. Doll and A. Powell Wheeler and Alexis Dinno},  
## year = {2023},  
## note = {R package version 0.9.5},  
## url = {https://CRAN.R-project.org/package=FSA},  
## }

Получим функцию Рикера из FSA.

(rckr <- srFuns('Ricker'))

## function (S, a, b = NULL)   
## {  
## if (length(a) > 1) {  
## b <- a[[2]]  
## a <- a[[1]]  
## }  
## a \* S \* exp(-b \* S)  
## }  
## <bytecode: 0x0000023dc76214c0>  
## <environment: 0x0000023dc7627a00>

Эта функция явно подразумевает нормальный аддитивный шум. Получим стартовые значения для оптимизатора.

(svR <- srStarts(R ~ S, data=pinks, type='Ricker'))

## $a  
## [1] 3.971341  
##   
## $b  
## [1] 0.0135286

Загрузим пакет для ортогональной нелинейной регрессии

library(onls)

## Loading required package: minpack.lm

citation('onls')

## To cite package 'onls' in publications use:  
##   
## Andrej-Nikolai Spiess (2022). \_onls: Orthogonal Nonlinear  
## Least-Squares Regression\_. R package version 0.1-2,  
## <https://CRAN.R-project.org/package=onls>.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {onls: Orthogonal Nonlinear Least-Squares Regression},  
## author = {{Andrej-Nikolai Spiess}},  
## year = {2022},  
## note = {R package version 0.1-2},  
## url = {https://CRAN.R-project.org/package=onls},  
## }

Предпочтительная практика - мультипликативыный шум (подгонка модели на лог-трансорфмированных данных) [Quinn II & Deriso, 1999; Hilborn & Walters, 2001], что уже можно считать давно принятой традицией [Subbey et al.,2014], поэтому получим оценку искомой связи в лог масштабе. НО ЗДЕСЬ УЖЕ БУДЕТ ОРТОГОНАЛЬНАЯ регрессия!

srR <- onls(LnR~log(rckr(S,a,b)),data=pinks,start=svR)

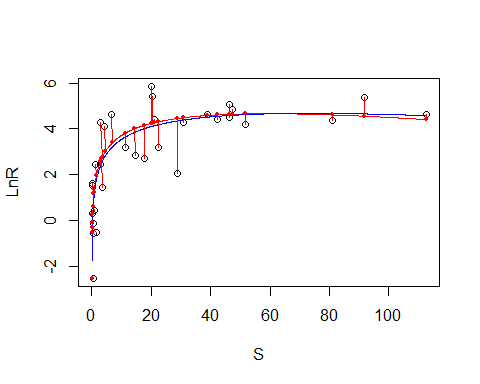
## Obtaining starting parameters from ordinary NLS...  
## Passed...  
## Conditions for `info = 1' and `info = 2' both hold.   
## Optimizing orthogonal NLS...  
## Passed... Conditions for `info = 1' and `info = 2' both hold.

print(srR)

## Nonlinear orthogonal regression model  
## model: LnR ~ log(rckr(S, a, b))  
## data: pinks  
## a b   
## 4.86983 0.01686   
## vertical residual sum-of-squares: 194  
## orthogonal residual sum-of-squares: 27.07  
## PASSED: 32 out of 32 fitted points are orthogonal.  
##   
## Number of iterations to convergence: 4   
## Achieved convergence tolerance: 1.49e-08

График

plot(srR)



Статистика

summary(srR)

##   
## Formula: LnR ~ log(rckr(S, a, b))  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)  
## a 4.86983 2.86949 1.697 0.100  
## b 0.01686 0.01581 1.066 0.295  
##   
## Residual standard error of vertical distances: 2.543 on 30 degrees of freedom  
## Residual standard error of orthogonal distances: 0.9499 on 30 degrees of freedom  
##   
## Number of iterations to convergence: 4   
## Achieved convergence tolerance: 1.49e-08

confint(srR)

## 2.5% 97.5%  
## a 2.731160765 8.78274879  
## b 0.003248549 0.03054111

Ошибка b НЕ выше его ожидания, поэтому его доверительный интервал НЕ пересекает 0. Нужна проверка значимости фактора плотности (т.к. он (b) может быть лишним).

ind <- srFuns('independence')  
svI <- srStarts(R~S,data=pinks,type='independence')  
srI <- onls(LnR~log(ind(S,a)),data=pinks,start=svI)

## Obtaining starting parameters from ordinary NLS...  
## Passed...  
## Relative error in the sum of squares is at most `ftol'.   
## Optimizing orthogonal NLS...  
## Passed... Relative error in the sum of squares is at most `ftol'.

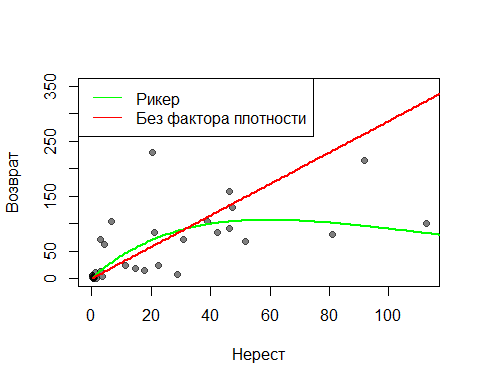
# Отношение правдоподобия  
lrt(srI,com=srR)

## Model 1: LnR ~ log(ind(S, a))  
## Model A: LnR ~ log(rckr(S, a, b))   
##   
## DfO logLikO DfA logLikA Df logLik Chisq Pr(>Chisq)   
## 1vA 31 -77.102 30 -74.240 1 -2.862 5.7241 0.01673 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Выходит, что фактор плотности имеет ВЫСОКУЮ статистическую значимость (p < 0.02).

Визуализируем связь по Рикеру.

x <- seq(min(pinks$S)\*.5, max(pinks$S)\*1.5,length.out=199) # many S for prediction  
pR <- rckr(x,a=coef(srR)) # прогнозируемый возврат по Рикеру, R  
pI <- ind(x,a=coef(srI)) # прогнозируемый возврат без фактора плотности (b), R  
plot(R ~ S,data=pinks,col='white',ylab='Возврат', xlab='Нерест')  
points(R~S,data=pinks,pch=19,col=rgb(0,0,0,1/2))  
lines(pR~x,lwd=2, col='green')  
lines(pI~x,lwd=2, col='red')  
legend('topleft', legend=c('Рикер', 'Без фактора плотности'),   
 col = c('green','red'), lty=1)



Модель без фактора плотности с коэффициентами от ортогональной регрессии.

summary(srI)

##   
## Formula: LnR ~ log(ind(S, a))  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## a 2.868 1.387 2.068 0.0471 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error of vertical distances: 2.736 on 31 degrees of freedom  
## Residual standard error of orthogonal distances: 1.029 on 31 degrees of freedom  
##   
## Number of iterations to convergence: 2   
## Achieved convergence tolerance: 1.49e-08

confint(srI)

## 2.5% 97.5%   
## 1.887189 4.378447

plot(srI)

