Приложение 6

# Базовая Линейная регрессия подходов к западной Камчатке от запаса

## Повторим настройку модели для рис. 5

Рис. 5 в статье - Регрессионная модель для оценки численности подходов горбуши западной Камчатки на основе линейной зависимости «родители – потомство» по данным 1990–2023 гг. с коэффициентами в табл. 4.

Загрузим данные.

options(OutDec= ',')  
(pinks <- as.data.frame(readxl::read\_excel('data.xlsx', sheet = 'Запад')))

## LnR LnS F1 F2 F3 F4  
## 1 2,8617902 2,6764735 -0,58 -1,15 -0,31 0,80  
## 2 -0,4993485 0,3504456 0,85 1,19 -0,94 0,54  
## 3 4,6370578 1,8988185 1,96 0,52 0,17 -0,39  
## 4 -0,5653405 -0,7318880 -0,88 2,59 0,38 0,83  
## 5 4,3870833 4,3955843 -0,11 -0,74 0,62 0,54  
## 6 -0,1425988 -0,6901517 -0,66 -1,18 0,38 0,13  
## 7 4,8550929 3,8605192 2,27 -0,34 -0,11 0,12  
## 8 -2,5181073 -0,3926723 -0,87 -0,02 1,37 0,65  
## 9 4,4318234 3,7404883 -1,84 1,12 -0,89 -0,67  
## 10 0,3126565 -3,1332717 -1,68 -0,86 0,22 0,14  
## 11 4,4226264 3,0428353 -1,68 -0,27 -0,73 0,52  
## 12 2,4359368 0,1949087 0,11 -1,64 -0,60 -0,23  
## 13 4,5195526 3,8329798 0,63 -0,90 0,37 -0,28  
## 14 3,1771537 2,4148418 0,24 -0,03 -0,10 -0,72  
## 15 4,2104648 3,9461539 -0,42 0,35 0,75 0,03  
## 16 2,7206985 2,8741859 -0,85 -0,16 0,98 -0,27  
## 17 4,2696263 3,4338582 0,09 0,02 0,12 -0,03  
## 18 1,4475568 1,2923408 -1,85 -1,32 1,70 -0,08  
## 19 4,6493878 3,6622531 -0,49 0,66 0,72 -0,05  
## 20 1,6198396 -2,1336866 -2,21 0,41 -0,02 -0,12  
## 21 5,0660509 3,8379889 -2,37 -0,51 0,41 -1,06  
## 22 0,4541537 -0,1839228 -2,60 -0,07 0,62 0,01  
## 23 2,0373166 3,3571758 -1,56 -0,17 1,59 0,15  
## 24 1,5260563 -1,5575573 0,34 -0,77 1,47 -0,37  
## 25 4,2654928 1,0986123 0,98 -1,50 0,17 -0,69  
## 26 2,4647039 1,0473190 -0,63 -0,37 1,54 0,47  
## 27 5,8579332 2,9957323 -0,62 -1,38 1,71 0,15  
## 28 4,1269731 1,4350845 -0,24 -1,38 0,82 0,84  
## 29 4,6128407 4,7238417 0,01 -2,19 0,70 -0,72  
## 30 5,4337220 3,0189604 -1,33 -0,21 0,38 -0,38  
## 31 3,1876328 3,1135153 -0,94 -1,94 2,23 -0,21  
## 32 5,3717769 4,5185224 -2,38 -0,41 0,97 -0,17

Простая линейная регрессия (ordinary LM), как в статье.

Lm1 <- lm(LnR~LnS, pinks)  
summary(Lm1)

##   
## Call:  
## lm(formula = LnR ~ LnS, data = pinks)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3,7143 -0,7566 0,0989 0,8971 2,0531   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1,4985 0,3260 4,597 7,25e-05 \*\*\*  
## LnS 0,7699 0,1147 6,713 1,94e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1  
##   
## Residual standard error: 1,35 on 30 degrees of freedom  
## Multiple R-squared: 0,6003, Adjusted R-squared: 0,587   
## F-statistic: 45,06 on 1 and 30 DF, p-value: 1,938e-07

anova(Lm1)

## Analysis of Variance Table  
##   
## Response: LnR  
## Df Sum Sq Mean Sq F value Pr(>F)   
## LnS 1 82,155 82,155 45,064 1,938e-07 \*\*\*  
## Residuals 30 54,692 1,823   
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

Рассчитаем доверительный интервал возвратов в LM в результате 1000 кратной перевыборки.

library(simpleboot)

## Simple Bootstrap Routines (1.1-7)

citation('simpleboot')

## To cite package 'simpleboot' in publications use:  
##   
## Peng RD (2019). \_simpleboot: Simple Bootstrap Routines\_. R package  
## version 1.1-7, <https://CRAN.R-project.org/package=simpleboot>.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {simpleboot: Simple Bootstrap Routines},  
## author = {Roger D. Peng},  
## year = {2019},  
## note = {R package version 1.1-7},  
## url = {https://CRAN.R-project.org/package=simpleboot},  
## }  
##   
## ATTENTION: This citation information has been auto-generated from the  
## package DESCRIPTION file and may need manual editing, see  
## 'help("citation")'.

set.seed(123)  
lboot <- lm.boot(Lm1, R = 1000, # число перевыборок  
 new.xpts=data.frame(LnS=log(seq(min(exp(pinks$LnS)\*0.9),max(exp(pinks$LnS)\*1.1),by=0.1))))  
summary(lboot)

## BOOTSTRAP OF LINEAR MODEL (method = rows)  
##   
## Original Model Fit  
## ------------------  
## Call:  
## lm(formula = LnR ~ LnS, data = pinks)  
##   
## Coefficients:  
## (Intercept) LnS   
## 1,4985 0,7699   
##   
## Bootstrap SD's:  
## (Intercept) LnS   
## 0,4520684 0,1322709

perc(lboot)

## (Intercept) LnS  
## 2,5% 0,4698431 0,5733948  
## 97,5% 2,2547510 1,0877489

Посмотрим на статистику коэффициентов.

Lm1bootBetas <- t(samples(lboot, 'coef'))  
psych::describe(Lm1bootBetas)

## vars n mean sd median trimmed mad min max range skew  
## (Intercept) 1 1000 1,46 0,45 1,50 1,48 0,40 -0,26 2,63 2,89 -0,55  
## LnS 2 1000 0,79 0,13 0,77 0,78 0,12 0,41 1,36 0,95 0,78  
## kurtosis se  
## (Intercept) 0,66 0,01  
## LnS 1,00 0,00

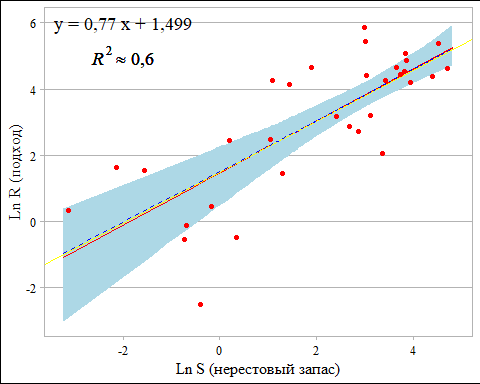
Очевидно, что общее смещение не сильно гуляет вокруг нуля.

Рассчитаем 95% доверительный интервал оценок в диапазоне наблюдений.

Lm1bootFit <- as.data.frame(t(samples(lboot, 'fitted')))  
Lm1quants = apply(Lm1bootFit, 2, quantile, probs = c(0.025,0.5,0.975))  
Lm1mu = rbind(Lm1quants, apply(Lm1bootFit, 2, mean), lboot$new.xpts[,1])  
Lm1forPlot = t(Lm1mu)  
colnames(Lm1forPlot) <- c('LCI','Med','HCI','Avg','LnS')

Построим график.

R2=round(cor(fitted(Lm1),pinks$LnR)^2,2)  
library(ggplot2)  
library(ggthemes)  
windowsFonts('Times New Roman' = windowsFont('Times New Roman'))  
ggplot(Lm1forPlot, aes(x=LnS, y=Avg))+  
 geom\_ribbon(aes(ymin=LCI,ymax=HCI), fill='lightblue')+  
 geom\_line(col='red')+ # среднее  
 geom\_abline(intercept=coef(Lm1)[1], slope=coef(Lm1)[2], col='yellow')+  
 geom\_line(aes(y=Med), col='blue', lty=2)+ # медиана  
 geom\_point(data=pinks, aes(x=LnS,y=LnR), col='red')+  
 annotate(geom = 'text',family='Times New Roman',size = 5, x=-2,y=6,  
 label = paste('y =',round(coef(Lm1)[2],3),'x +',round(coef(Lm1)[1],3)))+  
 annotate(geom = 'text',family='Times New Roman',size = 5, x=-2,y=5,  
 label = bquote(italic(R)^2%~~%.(R2)))+  
 xlab('Ln S (нерестовый запас)')+  
 ylab('Ln R (подход)')+  
 theme\_calc(base\_size = 12, base\_family = 'Times New Roman')



Красная прямая - по средним из перевыборки, синяя - по медианам из перевыборки, а жёлтая прямая - по коэффициентам из оценки линейной модели. Все они почти совпали.

Сохраним полученный рисунок в векторном формате для свежих версий Word и данные для него.

ggsave('fig05.svg', width = 13.5, height = 8.5, units = 'cm')  
writexl::write\_xlsx(as.data.frame(Lm1forPlot), path = 'fig05.xlsx')

## Диагностика Lm1

Проверим основные допущения линейной модели.

gvlma::gvlma(Lm1)

##   
## Call:  
## lm(formula = LnR ~ LnS, data = pinks)  
##   
## Coefficients:  
## (Intercept) LnS   
## 1,4985 0,7699   
##   
##   
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS  
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:  
## Level of Significance = 0,05   
##   
## Call:  
## gvlma::gvlma(x = Lm1)   
##   
## Value p-value Decision  
## Global Stat 3,3190 0,5059 Assumptions acceptable.  
## Skewness 1,8076 0,1788 Assumptions acceptable.  
## Kurtosis 0,1395 0,7088 Assumptions acceptable.  
## Link Function 0,8678 0,3516 Assumptions acceptable.  
## Heteroscedasticity 0,5041 0,4777 Assumptions acceptable.

Тест пройден.

Проверим есть нарушения i.i.d. по времени? Для этого загрузим пакет анализа временных рядов.

library(fpp3)  
citation('fpp3')

## To cite package 'fpp3' in publications use:  
##   
## Hyndman R (2023). \_fpp3: Data for "Forecasting: Principles and  
## Practice" (3rd Edition)\_. R package version 0.5,  
## <https://CRAN.R-project.org/package=fpp3>.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {fpp3: Data for "Forecasting: Principles and Practice" (3rd Edition)},  
## author = {Rob Hyndman},  
## year = {2023},  
## note = {R package version 0.5},  
## url = {https://CRAN.R-project.org/package=fpp3},  
## }

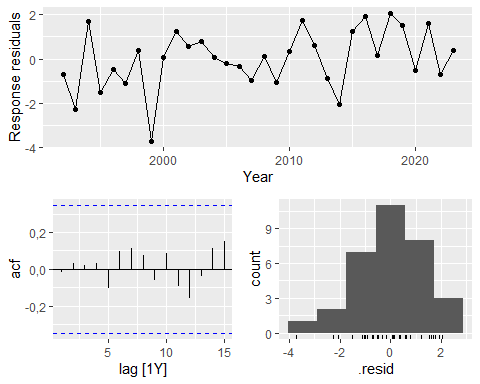
Посмотрим есть ли нужда в авторегрессии в невязках простой модели?

ts1 <- pinks |> tibble(Year=1992:2023) |> as\_tsibble(index=Year)  
fit1 <- ts1 |> model(ARIMA(LnR ~ 1+LnS))  
report(fit1)

## Series: LnR   
## Model: LM w/ ARIMA(0,0,0) errors   
##   
## Coefficients:  
## LnS intercept  
## 0,7699 1,4985  
## s.e. 0,1110 0,3156  
##   
## sigma^2 estimated as 1,823: log likelihood=-53,98  
## AIC=113,96 AICc=114,82 BIC=118,36

Нет. Примерно равно белому шуму ARIMA(0,0,0). В чём можно убедиться на графике ACF.

fit1 |> gg\_tsresiduals(type='response')



Проверим необходимость трансформации через перекрёстную проверку в соответствующем пакете.

library(cv)  
citation('cv')

## To cite package 'cv' in publications use:  
##   
## Fox J, Monette G (2024). \_cv: Cross-Validating Regression Models\_. R  
## package version 2.0.0, <https://CRAN.R-project.org/package=cv>.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {cv: Cross-Validating Regression Models},  
## author = {John Fox and Georges Monette},  
## year = {2024},  
## note = {R package version 2.0.0},  
## url = {https://CRAN.R-project.org/package=cv},  
## }

Заодно трансформируем предиктор S натуральным логарифмом заранее, чтобы была проверена ещё и модель по формуле 4 (LnR = b1 + b2×S + b3×LnS)

(D=transform(pinks, S=exp(LnS)))

## LnR LnS F1 F2 F3 F4 S  
## 1 2,8617902 2,6764735 -0,58 -1,15 -0,31 0,80 14,533750  
## 2 -0,4993485 0,3504456 0,85 1,19 -0,94 0,54 1,419700  
## 3 4,6370578 1,8988185 1,96 0,52 0,17 -0,39 6,678000  
## 4 -0,5653405 -0,7318880 -0,88 2,59 0,38 0,83 0,481000  
## 5 4,3870833 4,3955843 -0,11 -0,74 0,62 0,54 81,092000  
## 6 -0,1425988 -0,6901517 -0,66 -1,18 0,38 0,13 0,501500  
## 7 4,8550929 3,8605192 2,27 -0,34 -0,11 0,12 47,490000  
## 8 -2,5181073 -0,3926723 -0,87 -0,02 1,37 0,65 0,675250  
## 9 4,4318234 3,7404883 -1,84 1,12 -0,89 -0,67 42,118550  
## 10 0,3126565 -3,1332717 -1,68 -0,86 0,22 0,14 0,043575  
## 11 4,4226264 3,0428353 -1,68 -0,27 -0,73 0,52 20,964600  
## 12 2,4359368 0,1949087 0,11 -1,64 -0,60 -0,23 1,215200  
## 13 4,5195526 3,8329798 0,63 -0,90 0,37 -0,28 46,200000  
## 14 3,1771537 2,4148418 0,24 -0,03 -0,10 -0,72 11,188000  
## 15 4,2104648 3,9461539 -0,42 0,35 0,75 0,03 51,736000  
## 16 2,7206985 2,8741859 -0,85 -0,16 0,98 -0,27 17,711000  
## 17 4,2696263 3,4338582 0,09 0,02 0,12 -0,03 30,996000  
## 18 1,4475568 1,2923408 -1,85 -1,32 1,70 -0,08 3,641300  
## 19 4,6493878 3,6622531 -0,49 0,66 0,72 -0,05 38,949000  
## 20 1,6198396 -2,1336866 -2,21 0,41 -0,02 -0,12 0,118400  
## 21 5,0660509 3,8379889 -2,37 -0,51 0,41 -1,06 46,432000  
## 22 0,4541537 -0,1839228 -2,60 -0,07 0,62 0,01 0,832000  
## 23 2,0373166 3,3571758 -1,56 -0,17 1,59 0,15 28,708000  
## 24 1,5260563 -1,5575573 0,34 -0,77 1,47 -0,37 0,210650  
## 25 4,2654928 1,0986123 0,98 -1,50 0,17 -0,69 3,000000  
## 26 2,4647039 1,0473190 -0,63 -0,37 1,54 0,47 2,850000  
## 27 5,8579332 2,9957323 -0,62 -1,38 1,71 0,15 20,000000  
## 28 4,1269731 1,4350845 -0,24 -1,38 0,82 0,84 4,200000  
## 29 4,6128407 4,7238417 0,01 -2,19 0,70 -0,72 112,600000  
## 30 5,4337220 3,0189604 -1,33 -0,21 0,38 -0,38 20,470000  
## 31 3,1876328 3,1135153 -0,94 -1,94 2,23 -0,21 22,500000  
## 32 5,3717769 4,5185224 -2,38 -0,41 0,97 -0,17 91,700000

настроим полную модель из табл. 6 (рис. 7) в статье.

LmF4 <- lm(LnR~LnS+F1+F2+F3+F4, D)  
summary(LmF4)

##   
## Call:  
## lm(formula = LnR ~ LnS + F1 + F2 + F3 + F4, data = D)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2,86350 -0,78163 0,05405 0,84075 2,32538   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1,71208 0,36487 4,692 7,57e-05 \*\*\*  
## LnS 0,69743 0,11386 6,125 1,78e-06 \*\*\*  
## F1 0,09143 0,20022 0,457 0,6517   
## F2 -0,34184 0,25587 -1,336 0,1931   
## F3 -0,32176 0,32091 -1,003 0,3253   
## F4 -0,89125 0,49870 -1,787 0,0856 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1  
##   
## Residual standard error: 1,283 on 26 degrees of freedom  
## Multiple R-squared: 0,6875, Adjusted R-squared: 0,6273   
## F-statistic: 11,44 on 5 and 26 DF, p-value: 6,571e-06

anova(LmF4)

## Analysis of Variance Table  
##   
## Response: LnR  
## Df Sum Sq Mean Sq F value Pr(>F)   
## LnS 1 82,155 82,155 49,9409 1,667e-07 \*\*\*  
## F1 1 0,826 0,826 0,5020 0,48491   
## F2 1 3,231 3,231 1,9639 0,17292   
## F3 1 2,610 2,610 1,5867 0,21899   
## F4 1 5,254 5,254 3,1939 0,08558 .   
## Residuals 26 42,771 1,645   
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

Настроим простейшую модель без общего смещения

Lm0 <- lm(LnR~LnS-1, D)  
summary(Lm0)

##   
## Call:  
## lm(formula = LnR ~ LnS - 1, data = D)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2,0748 -0,1809 0,4740 2,0732 4,0286   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## LnS 1,1289 0,1079 10,47 1,07e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1  
##   
## Residual standard error: 1,734 on 31 degrees of freedom  
## Multiple R-squared: 0,7794, Adjusted R-squared: 0,7723   
## F-statistic: 109,6 on 1 and 31 DF, p-value: 1,066e-11

anova(Lm0)

## Analysis of Variance Table  
##   
## Response: LnR  
## Df Sum Sq Mean Sq F value Pr(>F)   
## LnS 1 329,45 329,45 109,56 1,066e-11 \*\*\*  
## Residuals 31 93,22 3,01   
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

Сравним их пока на полном наборе без трансформаций по критерию Шварца-Байеса

BIC(Lm0,Lm1,LmF4)

## df BIC  
## Lm0 2 131,9596  
## Lm1 3 118,3608  
## LmF4 7 124,3564

На полных данных и без трансформаций оптимальная модель - это Lm1.

anova(Lm1,LmF4)

## Analysis of Variance Table  
##   
## Model 1: LnR ~ LnS  
## Model 2: LnR ~ LnS + F1 + F2 + F3 + F4  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 30 54,692   
## 2 26 42,771 4 11,921 1,8116 0,1569

Итак, полная модель LmF4, опубликованная в табл. 6 (рис. 7) в статье не имеет значимого преимущества перед простой лог-линейной модели возврата от запаса.

Оценим необходимые трансформации

cvs <- cv(selectTransStepAIC,data = D, working.model = LmF4, k=2,  
 predictors = c('F1', 'F2', 'F3', 'F4'),   
 AIC = FALSE, criterion = medAbsErr,   
 ncores =10, family ='yjPower',reps=10  
)

## R RNG seed set to 765131

## R RNG seed set to 491514

## R RNG seed set to 986400

## R RNG seed set to 149

## R RNG seed set to 325368

## R RNG seed set to 347526

## R RNG seed set to 218730

## R RNG seed set to 700956

## R RNG seed set to 686968

## R RNG seed set to 808927

cvs

##   
## Replicate 1:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 1,295455  
## full-sample criterion = 0,9424243   
##   
## Replicate 2:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 1,017686  
## full-sample criterion = 0,9424243   
##   
## Replicate 3:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 1,22235  
## full-sample criterion = 0,9424243   
##   
## Replicate 4:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 0,9695555  
## full-sample criterion = 0,9424243   
##   
## Replicate 5:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 0,7923924  
## full-sample criterion = 0,9424243   
##   
## Replicate 6:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 1,324715  
## full-sample criterion = 0,9424243   
##   
## Replicate 7:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 1,531773  
## full-sample criterion = 0,9424243   
##   
## Replicate 8:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 0,9489635  
## full-sample criterion = 0,9424243   
##   
## Replicate 9:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 0,7629264  
## full-sample criterion = 0,9424243   
##   
## Replicate 10:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 0,921619  
## full-sample criterion = 0,9424243   
##   
## Average:  
## 2-Fold Cross Validation  
## criterion: medAbsErr  
## cross-validation criterion = 1,098445 (0,2479793)  
## full-sample criterion = 0,9424243

Посмотрим на все перезапуски

for (i in seq(10)) {  
 print('-----------------------------------------------------------------------')  
 print(paste('-------------------------------',i,'------------------------------------'))  
 compareFolds(cvs[[i]])  
}

## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 1 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 1,128610 1,403745   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F2 F3 F4  
## Fold 1 2,239 1,000 1,000 1,000 1,000 0,558 -0,385 -0,544   
## Fold 2 0,883 1,000 1,000 1,000 1,000 1,026 -2,03  
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 2 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 0,902658 1,303508   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F3 F4  
## Fold 1 1,201 1,000 1,000 1,000 1,000 0,773 -1,91  
## Fold 2 1,899 1,000 1,000 1,000 1,000 0,734 -0,544   
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 3 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 0,6924447 1,2587990   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F4  
## Fold 1 1,869 1,000 1,000 1,000 1,000 0,712   
## Fold 2 0,988 1,000 1,000 1,000 1,000 0,815 -2,07  
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 4 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 1,6143440 0,6757318   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS  
## Fold 1 -0,012 1,000 1,000 1,000 1,000 1,21  
## Fold 2 2,149 1,000 1,000 1,000 1,000 0,59  
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 5 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 0,9259645 0,6595534   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F4  
## Fold 1 1,381 1,000 1,000 1,000 1,000 0,833 -1,14  
## Fold 2 1,504 1,000 1,000 1,000 1,000 0,753   
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 6 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 1,324715 1,262248   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F1 F2 F3  
## Fold 1 1,865 1,000 1,000 1,000 1,000 0,732 0,334   
## Fold 2 1,626 1,000 1,000 1,000 1,000 0,696 -0,977 -0,864  
## F4  
## Fold 1 -0,99  
## Fold 2   
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 7 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 0,9529533 1,9787915   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F3  
## Fold 1 1,428 1,000 0,000 1,000 1,000 0,760   
## Fold 2 2,359 1,000 1,000 1,000 1,000 0,934 -2,1  
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 8 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 1,2609747 0,8421259   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F4  
## Fold 1 0,662 0,000 1,000 1,000 1,000 1,072   
## Fold 2 1,781 1,000 1,000 1,000 1,000 0,608 -1,54  
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 9 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 0,5589293 1,0772658   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F4  
## Fold 1 1,839 1,000 1,000 1,000 1,000 0,676   
## Fold 2 1,303 1,000 1,000 1,000 1,000 0,783 -1,3  
## [1] "-----------------------------------------------------------------------"  
## [1] "------------------------------- 10 ------------------------------------"  
## CV criterion by folds:  
## fold.1 fold.2   
## 0,6998628 1,0738956   
##   
## Coefficients by folds:  
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 LnS F4  
## Fold 1 2,221 1,000 1,000 1,000 1,000 0,638   
## Fold 2 1,123 1,000 1,000 1,000 1,000 0,710 -1,41

Итак, во всех перезапусках трансформация факторов F1-F4 (lam.F1…lam.F4=1) была не нужна. Во всех перезапусках оставался постоянно только один предиктор - LnS, а сочетания дополнительных предикторов зависело от зерна генератора псевдослучайных чисел. Однако, нигде все четыре (F1-F4) предиктора вместе не были выбраны в оптимальной модели. Следовательно, предпочтителен ансамблевый подход.

Загрузим пакет для усреднения моделей по Байесу

library(BAS)  
citation('BAS')

## To cite package 'BAS' in publications use:  
##   
## Clyde, Merlise (2023) BAS: Bayesian Variable Selection and Model  
## Averaging using Bayesian Adaptive Sampling, R package version 1.7.1  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {{BAS}: Bayesian Variable Selection and Model Averaging using Bayesian  
## Adaptive Sampling},  
## author = {Merlise Clyde},  
## year = {2023},  
## note = {R package version 1.7.1},  
## }

Посмотрим на полную модель со всеми предикторами.

LMfullBAS <- bas.lm(LnR ~ LnS + S + F1 + F2 + F3 + F4,data=D,  
 modelprior = uniform(), initprobs = 'eplogp')  
summary(LMfullBAS)

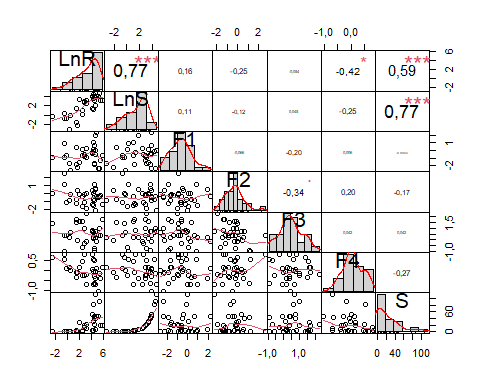
## P(B != 0 | Y) model 1 model 2 model 3 model 4 model 5  
## Intercept 1,0000000 1,00000 1,0000000 1,0000000 1,0000000 1,0000000  
## LnS 0,9993462 1,00000 1,0000000 1,0000000 1,0000000 1,0000000  
## S 0,1758481 0,00000 0,0000000 0,0000000 0,0000000 0,0000000  
## F1 0,2000456 0,00000 0,0000000 0,0000000 0,0000000 1,0000000  
## F2 0,3116866 0,00000 0,0000000 1,0000000 1,0000000 0,0000000  
## F3 0,2303024 0,00000 0,0000000 0,0000000 0,0000000 0,0000000  
## F4 0,5747962 1,00000 0,0000000 1,0000000 0,0000000 1,0000000  
## BF NA 1,00000 0,7213284 0,3385163 0,3100468 0,2579528  
## PostProbs NA 0,21620 0,1559000 0,0732000 0,0670000 0,0558000  
## R2 NA 0,65500 0,6003000 0,6691000 0,6252000 0,6624000  
## dim NA 3,00000 2,0000000 4,0000000 3,0000000 4,0000000  
## logmarg NA 10,98827 10,6616091 9,9050867 9,8172379 9,6332912

LMfullBAS

##   
## Call:  
## bas.lm(formula = LnR ~ LnS + S + F1 + F2 + F3 + F4, data = D,   
## modelprior = uniform(), initprobs = "eplogp")  
##   
##   
## Marginal Posterior Inclusion Probabilities:   
## Intercept LnS S F1 F2 F3 F4   
## 1,0000 0,9993 0,1758 0,2000 0,3117 0,2303 0,5748

Порог более 50% вероятности необходимости включения в модель преодолели только LnS и F4, а самая низкая вероятность включения в модель у S. Посмотрим на их отношения по отдельности.

PerformanceAnalytics::chart.Correlation(D)



Повторим настройку без S.

LMfullBAS2 <- bas.lm(LnR ~ LnS + F1 + F2 + F3 + F4,data=D,  
 modelprior = uniform(), initprobs = 'eplogp')  
summary(LMfullBAS2)

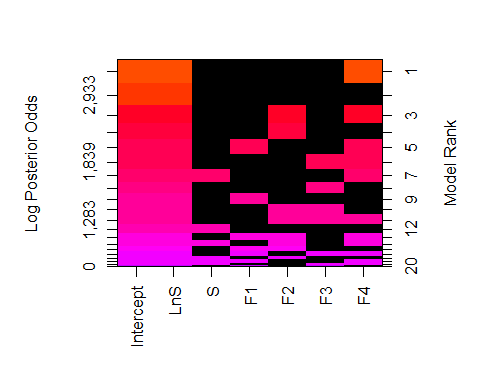
## P(B != 0 | Y) model 1 model 2 model 3 model 4 model 5  
## Intercept 1,0000000 1,00000 1,0000000 1,0000000 1,0000000 1,0000000  
## LnS 0,9999807 1,00000 1,0000000 1,0000000 1,0000000 1,0000000  
## F1 0,1971842 0,00000 0,0000000 0,0000000 0,0000000 1,0000000  
## F2 0,3062056 0,00000 0,0000000 1,0000000 1,0000000 0,0000000  
## F3 0,2261538 0,00000 0,0000000 0,0000000 0,0000000 0,0000000  
## F4 0,5690144 1,00000 0,0000000 1,0000000 0,0000000 1,0000000  
## BF NA 1,00000 0,7213284 0,3385163 0,3100468 0,2579528  
## PostProbs NA 0,26230 0,1892000 0,0888000 0,0813000 0,0677000  
## R2 NA 0,65500 0,6003000 0,6691000 0,6252000 0,6624000  
## dim NA 3,00000 2,0000000 4,0000000 3,0000000 4,0000000  
## logmarg NA 10,98827 10,6616091 9,9050867 9,8172379 9,6332912

LMfullBAS2

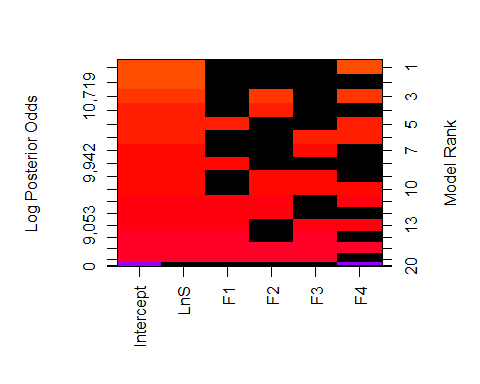
##   
## Call:  
## bas.lm(formula = LnR ~ LnS + F1 + F2 + F3 + F4, data = D, modelprior = uniform(),   
## initprobs = "eplogp")  
##   
##   
## Marginal Posterior Inclusion Probabilities:   
## Intercept LnS F1 F2 F3 F4   
## 1,0000 1,0000 0,1972 0,3062 0,2262 0,5690

Сравним конфигурации до упрощения и после

image(LMfullBAS)



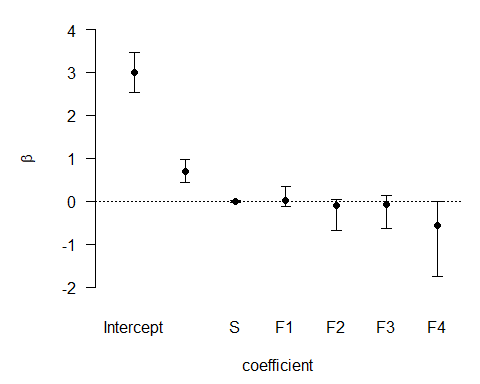
image(LMfullBAS2)



Верхушки предпочтительных конфигураций не изменились.

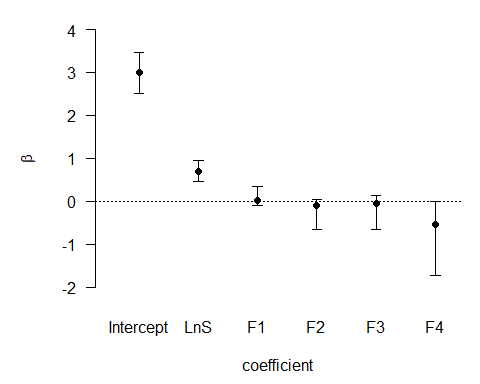
Сравним диапазоны общих доверительных интервалов из множества конфигураций.

plot(confint(coef(LMfullBAS, estimator = 'BMA')))



## NULL

plot(confint(coef(LMfullBAS2, estimator = 'BMA')))



## NULL

Очевидно, что коэффициент S незначимо отличался от 0, а его исключение не расширило доверительные интервалы оставшихся параметров.

confint(coef(LMfullBAS, estimator = 'BMA'))

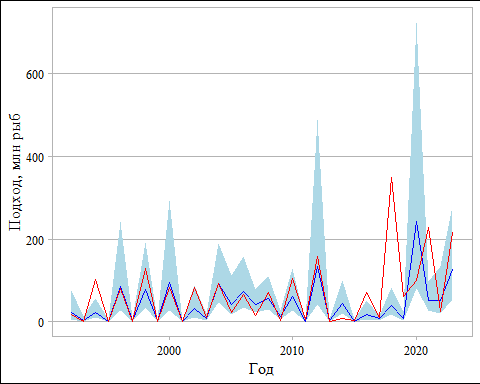
## 2,5% 97,5% beta  
## Intercept 2,52793559 3,473265043 2,9886751613  
## LnS 0,44506680 0,974630997 0,7058264783  
## S -0,01826477 0,008963439 -0,0006178704  
## F1 -0,09245836 0,378319352 0,0252113667  
## F2 -0,64218405 0,055512198 -0,0968006296  
## F3 -0,62425363 0,135020114 -0,0613929299  
## F4 -1,74531701 0,000000000 -0,5546676599  
## attr(,"Probability")  
## [1] 0,95  
## attr(,"class")  
## [1] "confint.bas"

confint(coef(LMfullBAS2, estimator = 'BMA'))

## 2,5% 97,5% beta  
## Intercept 2,51995631 3,470095698 2,98867516  
## LnS 0,46667263 0,936589581 0,70171707  
## F1 -0,08675487 0,372100170 0,02514938  
## F2 -0,62416123 0,069986706 -0,09487005  
## F3 -0,61403409 0,152216256 -0,06029550  
## F4 -1,74369172 0,001078747 -0,54823612  
## attr(,"Probability")  
## [1] 0,95  
## attr(,"class")  
## [1] "confint.bas"

Посмотрим на результаты усреднённых моделей из топ-10 конфигураций

BMA <- predict(LMfullBAS2, estimator = "BMA", se.fit = TRUE, top=10)  
conf.fit <- confint(BMA, parm = "mean")  
conf.fitDF <- data.frame(LCI=conf.fit[,1],HCI=conf.fit[,2],Avg=conf.fit[,3])  
expconf.fit <- exp(conf.fitDF)  
expconf.fit$Years=1992:2023  
pinks$Years=expconf.fit$Years  
BMA\_R2=round(cor(conf.fitDF$Avg,pinks$LnR)^2,3)  
library(ggplot2)  
library(ggthemes)  
ggplot(expconf.fit, aes(x=Years, y=Avg))+  
 geom\_ribbon(aes(ymin=LCI,ymax=HCI), fill='lightblue')+  
 geom\_line(col='blue')+ # среднее  
 geom\_line(data=pinks, aes(x=Years,y=exp(LnR)), col='red')+  
 xlab('Год')+ylab('Подход, млн рыб')+  
 theme\_calc(base\_size = 12, base\_family = 'Times New Roman')

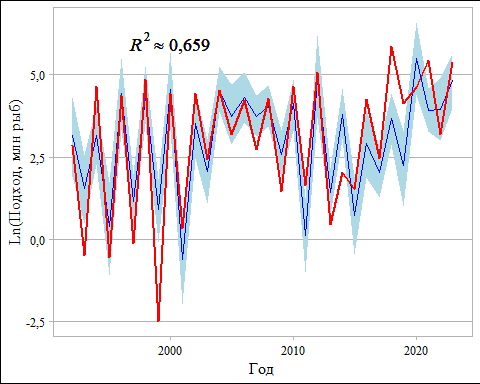


Красная кривая - данные, а синяя с фоном - их аппроксимация с доверительным интервалом.

Посмотрим в лог-масштабе.

conf.fitDF$Years=1992:2023  
ggplot(conf.fitDF, aes(x=Years, y=Avg))+  
 geom\_ribbon(aes(ymin=LCI,ymax=HCI), fill='lightblue')+  
 geom\_line(col='blue')+ # среднее  
 geom\_line(data=pinks, aes(x=Years,y=LnR), col='red',lwd=1)+  
 annotate(geom = 'text',family='Times New Roman',size = 5, x=2000,y=6,  
 label = bquote(italic(R)^2%~~%.(BMA\_R2)))+  
 xlab('Год')+ylab('Ln(Подход, млн рыб)')+  
 theme\_calc(base\_size = 12, base\_family = 'Times New Roman')

## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'language'



Посмотрим на BIC различных конфигураций.

tLMR <- lm(LnR ~ LnS+S,data=D)  
tLM1 <- lm(LnR ~ LnS,data=D)  
tLM2 <- lm(LnR ~ LnS+F2+F3+F4,data=D)  
tLM3 <- lm(LnR ~ LnS+F3+F4,data=D)  
tLM4 <- lm(LnR ~ LnS+F4,data=D)  
tLMfull <- lm(LnR ~ LnS + S + F1 + F2 + F3 + F4,data=D)  
BIC(tLMR,tLM1,tLM2,tLM3,tLM4,tLMfull)

## df BIC  
## tLMR 4 121,8142  
## tLM1 3 118,3608  
## tLM2 6 121,1463  
## tLM3 5 120,0418  
## tLM4 4 117,1166  
## tLMfull 8 127,5774

Оптимальная модель на полном наборе по BIC включает только LnS и F4, также, как и модель, выбираемая по Байесову подходу. Доверительные интервалы её коэффициентов:

confint(coef(LMfullBAS2, estimator = 'HPM'))

## 2,5% 97,5% beta  
## Intercept 2,528679 3,44867177 2,9886752  
## LnS 0,460270 0,90908245 0,6846762  
## F1 0,000000 0,00000000 0,0000000  
## F2 0,000000 0,00000000 0,0000000  
## F3 0,000000 0,00000000 0,0000000  
## F4 -1,969220 -0,03199801 -1,0006091  
## attr(,"Probability")  
## [1] 0,95  
## attr(,"class")  
## [1] "confint.bas"

## Сравнение с зависимостью по Рикеру

Перейдём к зависимости по Рикеру в пакете FSA.

library(FSA)

## Registered S3 methods overwritten by 'FSA':  
## method from  
## confint.boot car   
## hist.boot car

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

##   
## Attaching package: 'FSA'

## The following object is masked from 'package:simpleboot':  
##   
## perc

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: 0x000002e15b674468>  
## <environment: 0x000002e15b67c9a8>

Получим стартовые значения для оптимизатора.

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

## $a  
## [1] 3,971341  
##   
## $b  
## [1] 0,0135286

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.

srR <- nls(LnR~log(rckr(S,a,b)),data=pinks,start=svR)  
# Оценки параметров с доверительным интервалом из профиля правдоподобия  
cbind(estimates=coef(srR),confint(srR))

## Waiting for profiling to be done...

## estimates 2,5% 97,5%  
## a 3,9713409 2,067698820 7,63821793  
## b 0,0135286 -0,004013633 0,03107083

bootR <- nlstools::nlsBoot(srR)  
# Оценки параметров с доверительным интервалом из перевыборки  
cbind(estimates=coef(srR),confint(bootR))

## estimates 95% LCI 95% UCI  
## a 3,9713409 2,145501734 7,02134817  
## b 0,0135286 -0,002189514 0,02909692

Доверительный интервал b пересекает 0 (Он может быть лишним). Проверка значимости фактора плотности.

ind <- srFuns('independence')  
svI <- srStarts(R~S,data=pinks,type='independence')  
srI <- nls(LnR~log(ind(S,a)),data=pinks,start=svI)  
# Отношение правдоподобия  
lrt(srI,com=srR)

## Loading required namespace: lmtest

## 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 -55,9971 30 -54,7260 1 -1,2712 2,5423 0,1108

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# дополнительная проверка суммы квадратов  
extraSS(srI,com=srR)

## Model 1: LnR ~ log(ind(S, a))  
## Model A: LnR ~ log(rckr(S, a, b))   
##   
## DfO RSSO DfA RSSA Df SS F Pr(>F)  
## 1vA 31 62,0339 30 57,2962 1 4,7377 2,4806 0,1257

По обоим тестам выходит, что фактор плотности имеет низкую статистическую значимость (p > 0.1).

Опишем модель прямой пропорции.

# Оценки параметров с доверительным интервалом из профиля правдоподобия  
round(coef(srI),3)

## a   
## 2,866

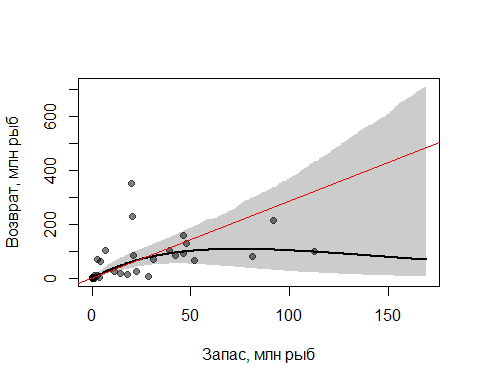
round(confint(srI),3)

## Waiting for profiling to be done...

## 2,5% 97,5%   
## 1,721 4,774

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, R  
LCI <- UCI <- numeric(length(x))  
for(i in 1:length(x)) { # 95% доверительные интервалы  
tmp <- apply(bootR$coefboot,MARGIN=1,FUN=rckr,S=x[i])  
LCI[i] <- quantile(tmp,0.025)  
UCI[i] <- quantile(tmp,0.975)  
}  
ylmts <- range(c(pR,LCI,UCI,pinks$R))  
xlmts <- range(c(x,pinks$S))  
plot(R ~ S,data=pinks,xlim=xlmts,ylim=ylmts,col='white',  
ylab='Возврат, млн рыб', xlab='Запас, млн рыб')  
polygon(c(x,rev(x)),c(LCI,rev(UCI)),col='gray80',border=NA)  
points(R~S,data=pinks,pch=19,col=rgb(0,0,0,1/2))  
lines(pR~x,lwd=2)  
abline(a=0,b=coef(srI), col = 'red')



В целом, прямая пропорция находится полностью внутри доверительного интервала зависимости по Рикеру.

## Добавим F4 в модель Рикера.

rckr2 <- function(S,X,a,b=NULL,c=NULL) {  
if (length(a)>1) { # all values in a argument  
c <- a[3]  
b <- a[2]  
a <- a[1]  
}  
a\*S\*exp(-b\*S+c\*X)  
}  
  
tmp <- lm(log(R/S)~S+F4,data=pinks)  
tmp <- coef(tmp)  
svR2 <- list(a=exp(tmp[[1]]),b=-tmp[[2]],c=tmp[[3]])  
srR2 <- nls(LnR~log(rckr2(S,F4,a,b,c)),data=pinks,start=svR2)  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Отношение правдоподобия  
lrt(srI,srR,com=srR2)

## Model 1: LnR ~ log(ind(S, a))  
## Model 2: LnR ~ log(rckr(S, a, b))  
## Model A: LnR ~ log(rckr2(S, F4, a, b, c))   
##   
## DfO logLikO DfA logLikA Df logLik Chisq Pr(>Chisq)   
## 1vA 31 -55,9971 29 -52,6087 2 -3,3885 6,7769 0,03376 \*  
## 2vA 30 -54,7260 29 -52,6087 1 -2,1173 4,2346 0,03961 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#дополнительная проверка суммы квадратов  
extraSS(srI,srR,com=srR2)

## Model 1: LnR ~ log(ind(S, a))  
## Model 2: LnR ~ log(rckr(S, a, b))  
## Model A: LnR ~ log(rckr2(S, F4, a, b, c))   
##   
## DfO RSSO DfA RSSA Df SS F Pr(>F)   
## 1vA 31 62,0339 29 50,1943 2 11,8396 3,4202 0,04638 \*  
## 2vA 30 57,2962 29 50,1943 1 7,1019 4,1031 0,05209 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

По отношению правдоподобия выходит, что учёт F4 имеет статистически значимое преимущество.

coef(srR2)

## a b c   
## 4,3785625 0,0182639 -1,0161433

Эффект F4 в среднем отрицательный, но каков его доверительный интервал?

#Оценки параметров с доверительным интервалом из профиля правдоподобия  
cbind(estimates=coef(srR2),confint(srR2))

## Waiting for profiling to be done...

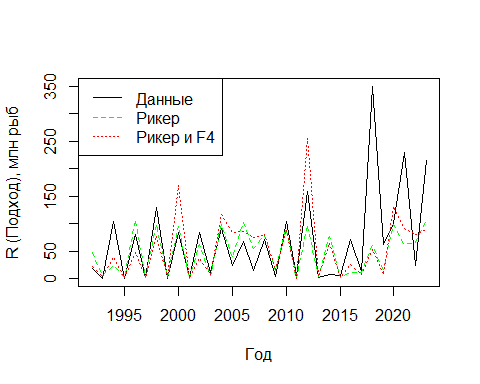
## estimates 2,5% 97,5%  
## a 4,3785625 2,3315726667 8,231946746  
## b 0,0182639 0,0008699258 0,035657877  
## c -1,0161433 -2,0421234140 0,009836851

bootR2 <- nlstools::nlsBoot(srR2)  
#Оценки параметров с доверительным интервалом из перевыборки  
cbind(estimates=coef(srR2),confint(bootR2))

## estimates 95% LCI 95% UCI  
## a 4,3785625 2,575636170 7,87227707  
## b 0,0182639 0,002906627 0,03437635  
## c -1,0161433 -1,942399989 -0,10534157

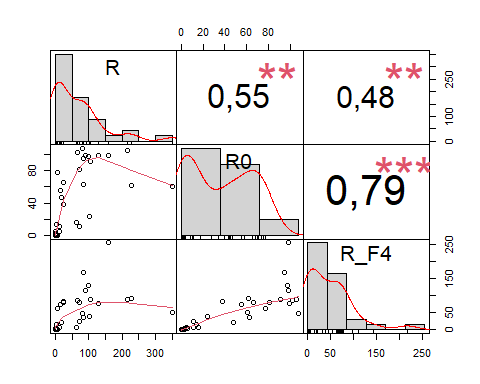
Сравним прогнозы по модели Рикера (R0) и по дополненной F4 (R\_F4).

pinks$R0 <- rckr(pinks$S,a=coef(srR))  
pinks$R\_F4 <- rckr2(pinks$S,pinks$F4,a=coef(srR2))  
pinks$Year = 1992:2023  
plot(R~Year, pinks, type='l', xlab='Год', ylab = 'R (Подход), млн рыб')  
with(pinks, lines(Year, R0, col='green', lty=2))  
with(pinks, lines(Year, R\_F4, col='red', lty=3))  
legend('topleft', legend=c('Данные', 'Рикер', 'Рикер и F4'),   
 col = c('black','green','red'), lty=1:3)



Связь с исходными данными прогнозов.

PerformanceAnalytics::chart.Correlation(pinks[,c('R','R0','R\_F4')])



Дополнение модели Рикера фактором F4 ухудшает связь наблюдённых подходов (R) с прогнозируемыми (R\_F4).

with(pinks, cor.test(R,R0))

##   
## Pearson's product-moment correlation  
##   
## data: R and R0  
## t = 3,5961, df = 30, p-value = 0,001143  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0,2475123 0,7533573  
## sample estimates:  
## cor   
## 0,5488382

with(pinks, cor.test(R,R\_F4))

##   
## Pearson's product-moment correlation  
##   
## data: R and R\_F4  
## t = 3,0032, df = 30, p-value = 0,005346  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0,1586940 0,7103842  
## sample estimates:  
## cor   
## 0,4807835

## Дополним модель Рикера всеми 4 факторами.

Допустим, что полная модель всё же лучше.

rckr4 <- function(S,X1,X2,X3,X4, a, b=NULL,c=NULL,d=NULL,e=NULL,f=NULL) {  
if (length(a)>1) { # all values in a argument  
a <- a[1]; b <- a[2]; c <- a[3]; d <- a[4]; e <- a[5]; f <- a[6]  
}  
a\*S\*exp(-b\*S+c\*X1+d\*X2+e\*X3+f\*X4)  
}  
  
tmp2 <- lm(log(R/S)~S+F1+F2+F3+F4,data=pinks)  
tmp2 <- coef(tmp2)  
svR4 <- list(a=exp(tmp2[[1]]),b=-tmp2[[2]],c=tmp2[[3]],d=tmp2[[4]],e=tmp2[[5]],f=tmp2[[6]])  
srR4 <- nls(LnR~log(rckr4(S,F1,F2,F3,F4,a,b,c,d,e,f)),data=pinks,start=svR4)  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Отношение правдоподобия  
lrt(srR,com=srR4)

## Model 1: LnR ~ log(rckr(S, a, b))  
## Model A: LnR ~ log(rckr4(S, F1, F2, F3, F4, a, b, c, d, e, f))   
##   
## DfO logLikO DfA logLikA Df logLik Chisq Pr(>Chisq)  
## 1vA 30 -54,7260 26 -51,0945 4 -3,6315 7,2629 0,1226

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#дополнительная проверка суммы квадратов  
extraSS(srR,com=srR4)

## Model 1: LnR ~ log(rckr(S, a, b))  
## Model A: LnR ~ log(rckr4(S, F1, F2, F3, F4, a, b, c, d, e, f))   
##   
## DfO RSSO DfA RSSA Df SS F Pr(>F)  
## 1vA 30 57,296 26 45,662 4 11,634 1,6561 0,1904

Очевидно, что значимых преимуществ у полной модели с 4 факторами перед классической по Рикеру нет.