Приложение 2

# Базовая Линейная регрессия подходов к восточной Камчатке от запаса

## Повторим настройку модели для рис. 4

Рис. 4 в статье - оценка численности подходов горбуши северо-восточной Камчатки на основе линейной зависимости «родители – потомство» по данным 1990–2023 гг. с коэффициентами в табл. 3.

Загрузим данные.

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

## R S F1 F2 F3 F4
## 1 7.140649 11.525700 0.37 -0.58 0.82 -0.11
## 2 60.338725 37.522150 0.45 -1.02 -0.72 -0.64
## 3 14.927016 2.831773 1.15 -0.44 -0.34 -0.36
## 4 75.380797 21.661500 -1.51 -0.93 0.72 -0.08
## 5 13.734330 5.760974 1.22 0.05 -0.15 -0.55
## 6 76.558736 38.471000 0.07 -1.41 1.22 -1.14
## 7 11.282888 5.871850 1.80 0.38 1.55 0.20
## 8 93.287568 24.832490 -1.70 1.81 -0.40 -1.05
## 9 2.028027 6.373000 -2.24 0.60 -0.75 0.06
## 10 40.817047 20.176125 -1.45 0.52 -0.97 0.40
## 11 4.402056 1.247745 -1.98 2.28 -1.23 -0.71
## 12 95.721893 11.189550 -0.21 0.79 0.31 -0.04
## 13 21.741115 2.847350 -0.17 -2.17 0.86 0.47
## 14 91.900381 57.864500 0.02 0.26 0.70 0.86
## 15 29.188146 17.623575 -0.99 0.57 0.53 0.80
## 16 98.307516 57.103850 -1.54 0.05 1.16 0.61
## 17 7.520554 19.704501 -0.97 1.20 1.49 0.18
## 18 191.368398 49.556350 -1.87 -0.83 3.34 -0.33
## 19 10.661200 1.995106 0.33 -1.02 1.28 0.88
## 20 185.184616 75.753500 -2.45 0.05 1.69 -0.87
## 21 28.990000 5.768310 -2.50 0.51 1.46 0.67
## 22 36.000000 44.363343 -2.99 0.69 0.84 0.77
## 23 47.570000 13.770000 -1.00 1.93 1.57 -0.46
## 24 128.474000 11.000000 0.76 -1.23 2.51 0.10
## 25 95.630000 25.320000 0.97 -1.38 0.99 -0.17
## 26 190.248000 90.500000 -0.80 -1.75 1.06 0.78
## 27 134.500000 41.310000 -0.18 -1.23 2.10 -0.49
## 28 301.000000 74.000000 -0.35 -1.78 2.67 0.59
## 29 28.270000 41.390000 0.13 0.86 0.77 0.31
## 30 263.800000 102.300000 -1.04 -2.44 1.30 0.63
## 31 51.531000 13.240000 -1.96 -0.65 2.16 -0.25
## 32 257.853000 104.895000 -2.29 1.82 0.47 -0.66

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

Lm1 <- lm(R~S, pinks)
summary(Lm1)

##
## Call:
## lm(formula = R ~ S, data = pinks)
##
## Residuals:
## Min 1Q Median 3Q Max
## -76.954 -23.396 -1.406 15.988 119.366
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.2412 11.4928 0.717 0.479
## S 2.3431 0.2612 8.971 5.39e-10 \*\*\*
## ---
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.94 on 30 degrees of freedom
## Multiple R-squared: 0.7284, Adjusted R-squared: 0.7194
## F-statistic: 80.47 on 1 and 30 DF, p-value: 5.389e-10

anova(Lm1)

## Analysis of Variance Table
##
## Response: R
## Df Sum Sq Mean Sq F value Pr(>F)
## S 1 155369 155369 80.475 5.389e-10 \*\*\*
## Residuals 30 57919 1931
## ---
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Рассчитаем доверительный интервал возвратов в LM в результате 1000 кратной перевыборки в интервале от 1 до 120 млн рыб в запасе (S).

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(S=seq(120)) # значения S от 1 до 120 для перевыборки
 )
summary(lboot)

## BOOTSTRAP OF LINEAR MODEL (method = rows)
##
## Original Model Fit
## ------------------
## Call:
## lm(formula = R ~ S, data = pinks)
##
## Coefficients:
## (Intercept) S
## 8.241 2.343
##
## Bootstrap SD's:
## (Intercept) S
## 8.5575558 0.2763834

perc(lboot)

## (Intercept) S
## 2.5% -8.372056 1.759118
## 97.5% 25.333941 2.890325

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 7.83 8.56 7.77 7.68 8.18 -13.80 44.62 58.41 0.22
## S 2 1000 2.34 0.28 2.34 2.35 0.24 0.62 3.21 2.59 -0.48
## kurtosis se
## (Intercept) 0.17 0.27
## S 2.45 0.01

Очевидно, что общее смещение сильно гуляет вокруг нуля (оно НЕ нужно), но мы вернёмся к этому позже.

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','S')

Построим график.

R2=round(cor(fitted(Lm1),pinks$R)^2,2)
library(ggplot2)
library(ggthemes)
windowsFonts('Times New Roman' = windowsFont('Times New Roman'))
options(OutDec= ',')
ggplot(Lm1forPlot, aes(x=S, 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=S,y=R), col='red')+
 annotate(geom = 'text',family='Times New Roman',size = 5, x=60,y=350,
 label = paste('y =',round(coef(Lm1)[2],3),'x +',round(coef(Lm1)[1],3)))+
 annotate(geom = 'text',family='Times New Roman',size = 5, x=50,y=250,
 label = bquote(italic(R)^2%~~%.(R2)))+
 xlab('S (нерестовый запас), млн рыб')+
 ylab('R (подход), млн рыб')+
 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 совпали.

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

ggsave('fig04.svg', width = 13.5, height = 8.5, units = 'cm')
writexl::write\_xlsx(as.data.frame(Lm1forPlot), path = 'fig04.xlsx')

## Диагностика Lm1

Проверим основные допущения линейной модели.

gvlma::gvlma(Lm1)

##
## Call:
## lm(formula = R ~ S, data = pinks)
##
## Coefficients:
## (Intercept) S
## 8,241 2,343
##
##
## 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 9,5445 0,04884 Assumptions NOT satisfied!
## Skewness 2,8433 0,09175 Assumptions acceptable.
## Kurtosis 1,0429 0,30715 Assumptions acceptable.
## Link Function 0,6976 0,40361 Assumptions acceptable.
## Heteroscedasticity 4,9607 0,02593 Assumptions NOT satisfied!

Тест НЕ пройден по общей статистике и стабильности дисперсии (есть гетероскедастичность, т.е. дисперсия изменяется в связи с мат. ожиданием). Это логично, т.к. функция связи для исключительно положительных значений шумных данных должна была быть логарифмом!

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(R ~ 1+S))
report(fit1)

## Series: R
## Model: LM w/ ARIMA(0,0,0) errors
##
## Coefficients:
## S intercept
## 2,3431 8,2412
## s.e. 0,2529 11,1279
##
## sigma^2 estimated as 1931: log likelihood=-165,42
## AIC=336,85 AICc=337,7 BIC=341,24

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

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



Сравним с аналогичной моделью без общего смещения

fit2 <- ts1 |> model(ARIMA(R ~ S))
report(fit2)

## Series: R
## Model: LM w/ ARIMA(0,0,0) errors
##
## Coefficients:
## S
## 2,4812
## s.e. 0,1724
##
## sigma^2 estimated as 1900: log likelihood=-165,7
## AIC=335,39 AICc=335,8 BIC=338,32

Все информационные критерии (AIC,AICc,BIC) ниже у модели без общего смещения, следовательно, этот свободный член в модели лишний.

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, LnS=log(S)))

## R S F1 F2 F3 F4 LnS
## 1 7,140649 11,525700 0,37 -0,58 0,82 -0,11 2,4445793
## 2 60,338725 37,522150 0,45 -1,02 -0,72 -0,64 3,6249314
## 3 14,927016 2,831773 1,15 -0,44 -0,34 -0,36 1,0409031
## 4 75,380797 21,661500 -1,51 -0,93 0,72 -0,08 3,0755365
## 5 13,734330 5,760974 1,22 0,05 -0,15 -0,55 1,7511066
## 6 76,558736 38,471000 0,07 -1,41 1,22 -1,14 3,6499047
## 7 11,282888 5,871850 1,80 0,38 1,55 0,20 1,7701697
## 8 93,287568 24,832490 -1,70 1,81 -0,40 -1,05 3,2121529
## 9 2,028027 6,373000 -2,24 0,60 -0,75 0,06 1,8520703
## 10 40,817047 20,176125 -1,45 0,52 -0,97 0,40 3,0045000
## 11 4,402056 1,247745 -1,98 2,28 -1,23 -0,71 0,2213378
## 12 95,721893 11,189550 -0,21 0,79 0,31 -0,04 2,4149803
## 13 21,741115 2,847350 -0,17 -2,17 0,86 0,47 1,0463887
## 14 91,900381 57,864500 0,02 0,26 0,70 0,86 4,0581041
## 15 29,188146 17,623575 -0,99 0,57 0,53 0,80 2,8692375
## 16 98,307516 57,103850 -1,54 0,05 1,16 0,61 4,0448715
## 17 7,520554 19,704501 -0,97 1,20 1,49 0,18 2,9808471
## 18 191,368398 49,556350 -1,87 -0,83 3,34 -0,33 3,9031104
## 19 10,661200 1,995106 0,33 -1,02 1,28 0,88 0,6906969
## 20 185,184616 75,753500 -2,45 0,05 1,69 -0,87 4,3274846
## 21 28,990000 5,768310 -2,50 0,51 1,46 0,67 1,7523792
## 22 36,000000 44,363343 -2,99 0,69 0,84 0,77 3,7924135
## 23 47,570000 13,770000 -1,00 1,93 1,57 -0,46 2,6224923
## 24 128,474000 11,000000 0,76 -1,23 2,51 0,10 2,3978953
## 25 95,630000 25,320000 0,97 -1,38 0,99 -0,17 3,2315946
## 26 190,248000 90,500000 -0,80 -1,75 1,06 0,78 4,5053499
## 27 134,500000 41,310000 -0,18 -1,23 2,10 -0,49 3,7211046
## 28 301,000000 74,000000 -0,35 -1,78 2,67 0,59 4,3040651
## 29 28,270000 41,390000 0,13 0,86 0,77 0,31 3,7230393
## 30 263,800000 102,300000 -1,04 -2,44 1,30 0,63 4,6279097
## 31 51,531000 13,240000 -1,96 -0,65 2,16 -0,25 2,5832426
## 32 257,853000 104,895000 -2,29 1,82 0,47 -0,66 4,6529598

настроим полную модель из табл. 5 (рис. 6) в статье.

LmF4 <- lm(R~S+F1+F2+F3+F4, D)
summary(LmF4)

##
## Call:
## lm(formula = R ~ S + F1 + F2 + F3 + F4, data = D)
##
## Residuals:
## Min 1Q Median 3Q Max
## -60,011 -24,563 -1,865 14,538 96,828
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2,1903 11,4152 -0,192 0,8493
## S 2,1138 0,2635 8,021 1,69e-08 \*\*\*
## F1 -1,7069 6,6645 -0,256 0,7999
## F2 -8,7807 7,2362 -1,213 0,2359
## F3 17,2939 7,3886 2,341 0,0272 \*
## F4 -21,1220 12,1726 -1,735 0,0945 .
## ---
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1
##
## Residual standard error: 39,09 on 26 degrees of freedom
## Multiple R-squared: 0,8138, Adjusted R-squared: 0,778
## F-statistic: 22,72 on 5 and 26 DF, p-value: 9,861e-09

anova(LmF4)

## Analysis of Variance Table
##
## Response: R
## Df Sum Sq Mean Sq F value Pr(>F)
## S 1 155369 155369 101,6997 1,778e-10 \*\*\*
## F1 1 624 624 0,4081 0,52851
## F2 1 5420 5420 3,5477 0,07086 .
## F3 1 7555 7555 4,9455 0,03506 \*
## F4 1 4600 4600 3,0110 0,09454 .
## Residuals 26 39721 1528
## ---
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

Настроим простейшую модель без общего смещения

Lm0 <- lm(R~S-1, D)
summary(Lm0)

##
## Call:
## lm(formula = R ~ S - 1, data = D)
##
## Residuals:
## Min 1Q Median 3Q Max
## -74,426 -19,535 0,373 19,419 117,392
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## S 2,4812 0,1751 14,17 4,38e-15 \*\*\*
## ---
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1
##
## Residual standard error: 43,59 on 31 degrees of freedom
## Multiple R-squared: 0,8662, Adjusted R-squared: 0,8619
## F-statistic: 200,7 on 1 and 31 DF, p-value: 4,383e-15

anova(Lm0)

## Analysis of Variance Table
##
## Response: R
## Df Sum Sq Mean Sq F value Pr(>F)
## S 1 381406 381406 200,7 4,383e-15 \*\*\*
## Residuals 31 58912 1900
## ---
## 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 2 338,3216
## Lm1 3 341,2435
## LmF4 7 343,0367

На полных данных и без трансформаций оптимальная модель - это простейшая Lm0. Кстати, её коэффициент детерминации выше (0,87), чем у более сложной модели (0,81).

anova(Lm0,LmF4)

## Analysis of Variance Table
##
## Model 1: R ~ S - 1
## Model 2: R ~ S + F1 + F2 + F3 + F4
## Res.Df RSS Df Sum of Sq F Pr(>F)
## 1 31 58912
## 2 26 39721 5 19191 2,5124 0,05533 .
## ---
## Signif. codes: 0 '\*\*\*' 0,001 '\*\*' 0,01 '\*' 0,05 '.' 0,1 ' ' 1

Итак, полная модель LmF4, опубликованная в табл. 5 (рис. 6) в статье не имеет значимого преимущества перед простой прямо пропорциональной связью возврата от запаса.

Оценим необходимые трансформации

cvs <- cv(selectTransStepAIC,data = D, working.model = LmF4, k=2,
 predictors = c('S', 'F1', 'F2', 'F3', 'F4'),
 response = 'R', 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 = 25,33133
## full-sample criterion = 20,59514
##
## Replicate 2:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 21,19245
## full-sample criterion = 20,59514
##
## Replicate 3:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 37,2555
## full-sample criterion = 20,59514
##
## Replicate 4:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 44,42801
## full-sample criterion = 20,59514
##
## Replicate 5:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 21,88203
## full-sample criterion = 20,59514
##
## Replicate 6:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 29,05246
## full-sample criterion = 20,59514
##
## Replicate 7:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 22,39061
## full-sample criterion = 20,59514
##
## Replicate 8:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 34,879
## full-sample criterion = 20,59514
##
## Replicate 9:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 31,43576
## full-sample criterion = 20,59514
##
## Replicate 10:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 26,83416
## full-sample criterion = 20,59514
##
## Average:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 29,09206 (7,304097)
## full-sample criterion = 20,59514

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
## 31,96535 22,99210
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F2 F3
## Fold 1 -8,917 1,000 1,000 1,000 1,000 0,000 0,500 6,295 -3,345 2,36
## Fold 2 1,327 1,000 1,000 1,000 1,000 0,000 0,000 0,836
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 2 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 18,61324 25,18012
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F3
## Fold 1 -6,818 1,000 1,000 1,000 1,000 0,000 0,500 6,623
## Fold 2 1,227 1,000 1,000 1,000 1,000 0,000 0,000 0,831 0,21
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 3 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 43,14315 32,40645
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F1
## Fold 1 1,268 1,000 1,000 1,000 1,000 0,000 0,000 0,668 -0,338
## Fold 2 -2,940 1,000 1,000 1,000 1,000 0,000 0,500 4,538
## F2 F3 F4
## Fold 1 -0,487
## Fold 2 2,981 -5,87
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 4 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 22,17904 78,47686
##
## Coefficients by folds:
## (Intercept) F4 lam.F1 lam.F2 lam.F3 lam.F4 lam.S
## Fold 1 0,678 -0,620 1,000 1,000 1,000 1,000 0,000
## Fold 2 -150,095 34,377 1,000 1,000 1,000 1,000 0,000
## lambda S F2 F3
## Fold 1 0,000 0,965 -0,340
## Fold 2 1,000 65,155 34,7
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 5 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 25,07585 20,25354
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F3
## Fold 1 -6,50 1,00 1,00 1,00 1,00 0,00 0,50 6,98
## Fold 2 -4,54 1,00 1,00 1,00 1,00 0,00 0,50 4,96 3,34
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 6 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 28,73045 36,55431
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F2 F3
## Fold 1 0,940 1,000 1,000 1,000 1,000 0,000 0,000 0,750 0,68
## Fold 2 1,159 1,000 1,000 1,000 1,000 0,000 0,000 0,900 -0,234
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 7 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 41,40229 20,78785
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F3
## Fold 1 -5,321 1,000 1,000 1,000 1,000 0,000 0,500 5,479 2,38
## Fold 2 1,531 1,000 1,000 1,000 1,000 0,000 0,000 0,788
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 8 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 27,95470 43,01409
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S F1
## Fold 1 3,964 1,000 1,000 1,000 1,000 1,000 0,500 0,280
## Fold 2 1,627 1,000 1,000 1,000 1,000 0,000 0,000 0,791 0,182
## F2 F3 F4
## Fold 1 -1,846 -3,11
## Fold 2 0,324
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 9 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 26,55007 39,84689
##
## Coefficients by folds:
## (Intercept) lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S
## Fold 1 14,689 1,000 1,000 1,000 1,000 1,000 1,000 2,225
## Fold 2 1,022 1,000 1,000 1,000 1,000 0,000 0,000 0,830
## F2 F3 F4
## Fold 1 -23,189 -45,4
## Fold 2 0,353
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 10 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 24,01487 28,15248
##
## Coefficients by folds:
## (Intercept) F2 lam.F1 lam.F2 lam.F3 lam.F4 lam.S lambda S
## Fold 1 1,469 -0,394 1,000 1,000 1,000 1,000 0,000 0,000 0,848
## Fold 2 -8,756 -2,299 1,000 1,000 1,000 1,000 0,000 0,500 6,405
## F1 F4
## Fold 1 -0,74
## Fold 2 -2,029

Итак, во всех перезапусках трансформация факторов F1-F4 (lam.F1…lam.F4=1) была не нужна, а нужна была везде для S логарифмическая трансформация (lam.S=0). В большинстве случаев требовалась логарифмическая трансформация R (lambda=0). Среди предикторов ни разу не остался в конфигурации модели F1, а остальные не сохраняли стабильную конфигурацию. Повторим тестирование, учитывая эти выводы.

D2=transform(D, LnR=log(R))
LmFull <- lm(LnR~LnS+S+F2+F3+F4,D2)
cvsFULL <- cv(selectTransStepAIC,data = D2, working.model = LmFull, k=2,
 predictors = c('S', 'F2', 'F3', 'F4'), AIC = FALSE, criterion = medAbsErr,
 ncores =10, family ='yjPower',reps=10
)

## R RNG seed set to 963940

## R RNG seed set to 432378

## R RNG seed set to 678172

## R RNG seed set to 16824

## R RNG seed set to 356189

## R RNG seed set to 148865

## R RNG seed set to 21507

## R RNG seed set to 631826

## R RNG seed set to 831195

## R RNG seed set to 109315

cvsFULL

##
## Replicate 1:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4592217
## full-sample criterion = 0,3072241
##
## Replicate 2:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,7794918
## full-sample criterion = 0,3072241
##
## Replicate 3:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,3933158
## full-sample criterion = 0,3072241
##
## Replicate 4:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4624348
## full-sample criterion = 0,3072241
##
## Replicate 5:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,6798963
## full-sample criterion = 0,3072241
##
## Replicate 6:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,7064283
## full-sample criterion = 0,3072241
##
## Replicate 7:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,5133765
## full-sample criterion = 0,3072241
##
## Replicate 8:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4147315
## full-sample criterion = 0,3072241
##
## Replicate 9:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4267644
## full-sample criterion = 0,3072241
##
## Replicate 10:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,3612499
## full-sample criterion = 0,3072241
##
## Average:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,5141939 (0,1410879)
## full-sample criterion = 0,3072241

Посмотрим на все перезапуски

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

## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 1 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,9338254 0,3591499
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S F3 F4 LnS S
## Fold 1 0,417 1,000 1,000 1,000 1,000 0,418 -0,393 1,010
## Fold 2 1,227 1,000 1,000 1,000 0,000 0,89
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 2 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,6726818 0,8095999
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S S F3 LnS
## Fold 1 1,002 1,000 1,000 1,000 0,000 0,774 0,540
## Fold 2 -1,543 1,000 1,000 1,000 0,000 9,512 -8,01
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 3 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,5761796 0,3603094
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S S F3
## Fold 1 1,107 1,000 1,000 1,000 0,000 0,749 0,49
## Fold 2 1,069 1,000 1,000 1,000 0,000 0,953
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 4 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,3827554 0,4736470
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S F2 F3 LnS S
## Fold 1 1,857 1,000 1,000 1,000 0,000 0,305 0,631
## Fold 2 0,753 1,000 1,000 1,000 0,000 -0,409 0,94
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 5 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,8011633 0,5085816
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S F3 F4 LnS S
## Fold 1 1,033 1,000 1,000 1,000 0,000 0,598 -0,602 0,64
## Fold 2 1,692 1,000 1,000 1,000 0,000 0,757
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 6 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,8896110 0,6581541
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S LnS F2 F3 F4 S
## Fold 1 -2,237 1,000 1,000 1,000 0,000 -9,781 0,482 -0,531 11,3
## Fold 2 1,713 0,000 1,000 1,000 0,000 0,638 -0,300
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 7 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,4938313 0,5133765
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S F2 F3 F4 LnS S
## Fold 1 1,160 1,000 1,000 1,000 0,000 0,342 0,818
## Fold 2 0,978 1,000 1,000 1,000 0,000 -0,440 -0,716 0,91
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 8 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,5963274 0,3536992
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S F2 F3 LnS S
## Fold 1 0,520 1,000 1,000 1,000 0,000 0,381 0,981
## Fold 2 1,448 1,000 1,000 1,000 0,000 -0,236 0,78
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 9 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,2973835 0,5239986
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S S F3
## Fold 1 0,583 1,000 1,000 1,000 0,000 0,939 0,41
## Fold 2 1,129 1,000 1,000 1,000 0,000 0,876
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 10 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,3729937 0,3504191
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 lam.S S F2
## Fold 1 1,395 1,000 1,000 1,000 0,000 0,837
## Fold 2 1,083 1,000 1,000 1,000 0,000 0,860 -0,31

В тех редких случаях, когда в оптимальной конфигурации модели, настроенной на подвыборке, находятся одновременно 2 предиктора запаса (LnS и S), как предложено по формуле 4, тогда модель всё равно будет оптимальнее при трансформации оставшегося предиктора S логарифмом (lam.S=0). Следовательно, добавлять его нет никакого смысла.

Снова повторим проверку, упростив полную модель.

LmFull2 <- lm(LnR~LnS+F2+F3+F4,D2)
cvsFULL2 <- cv(selectTransStepAIC,data = D2, working.model = LmFull2, k=2,
 predictors = c('F2', 'F3', 'F4'), AIC = FALSE, criterion = medAbsErr,
 ncores =10, family ='yjPower',reps=10
)

## R RNG seed set to 948668

## R RNG seed set to 657334

## R RNG seed set to 978291

## R RNG seed set to 245912

## R RNG seed set to 307404

## R RNG seed set to 466218

## R RNG seed set to 252980

## R RNG seed set to 895955

## R RNG seed set to 461473

## R RNG seed set to 240311

cvsFULL2

##
## Replicate 1:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,6505135
## full-sample criterion = 0,3645214
##
## Replicate 2:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,5366753
## full-sample criterion = 0,3645214
##
## Replicate 3:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,5439826
## full-sample criterion = 0,3645214
##
## Replicate 4:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,6552364
## full-sample criterion = 0,3645214
##
## Replicate 5:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,6099749
## full-sample criterion = 0,3645214
##
## Replicate 6:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4282108
## full-sample criterion = 0,3645214
##
## Replicate 7:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,5477263
## full-sample criterion = 0,3645214
##
## Replicate 8:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4843874
## full-sample criterion = 0,3645214
##
## Replicate 9:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,4589376
## full-sample criterion = 0,3645214
##
## Replicate 10:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,3980021
## full-sample criterion = 0,3645214
##
## Average:
## 2-Fold Cross Validation
## criterion: medAbsErr
## cross-validation criterion = 0,5421964 (0,09217378)
## full-sample criterion = 0,3645214

Посмотрим на все перезапуски

for (i in seq(10)) {
 print('-----------------------------------------------------------------------')
 print(paste('-------------------------------',i,'------------------------------------'))
 compareFolds(cvsFULL2[[i]])
}

## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 1 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,6474108 0,6517567
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F2 F3 F4
## Fold 1 1,453 1,000 1,000 1,000 0,722
## Fold 2 1,586 1,000 1,000 1,000 0,744 -0,189 0,199 -0,32
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 2 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,6387862 0,4676157
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F3
## Fold 1 1,178 1,000 1,000 1,000 0,766 0,41
## Fold 2 1,459 1,000 1,000 1,000 0,785
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 3 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,4405046 0,5577506
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F3
## Fold 1 1,394 1,000 1,000 1,000 0,849
## Fold 2 1,739 1,000 1,000 1,000 0,531 0,55
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 4 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,4242063 0,9372681
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F2
## Fold 1 1,726 1,000 1,000 1,000 0,781
## Fold 2 0,358 1,000 1,000 1,000 1,082 -0,51
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 5 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,6011365 0,6099749
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F2 F4
## Fold 1 0,945 1,000 1,000 1,000 0,910 -0,461 -0,94
## Fold 2 1,978 1,000 1,000 1,000 0,652
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 6 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,4521758 0,3767937
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F3
## Fold 1 1,835 1,000 1,000 1,000 0,736
## Fold 2 0,369 1,000 1,000 1,000 0,975 0,41
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 7 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,5853314 0,5014802
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F3
## Fold 1 1,500 1,000 1,000 1,000 0,741
## Fold 2 1,377 1,000 1,000 1,000 0,713 0,47
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 8 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,5085568 0,3776938
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F2
## Fold 1 1,387 1,000 1,000 1,000 0,854
## Fold 2 1,440 1,000 1,000 1,000 0,742 -0,49
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 9 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,3909500 0,6650045
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F3
## Fold 1 1,525 1,000 1,000 1,000 0,796
## Fold 2 1,246 1,000 1,000 1,000 0,702 0,48
## [1] "-----------------------------------------------------------------------"
## [1] "------------------------------- 10 ------------------------------------"
## CV criterion by folds:
## fold.1 fold.2
## 0,5305981 0,3446072
##
## Coefficients by folds:
## (Intercept) lam.F2 lam.F3 lam.F4 LnS F2
## Fold 1 1,270 1,000 1,000 1,000 0,779 -0,36
## Fold 2 1,576 1,000 1,000 1,000 0,824

Итак, мы снова убедились, что всем факторам (F2-F4) не требуется трансформация. Во всех перезапусках оставался постоянно только один предиктор - LnS, а сочетания дополнительных предикторов зависело от зерна генератора псевдослучайных чисел. Однако, нигде все три (F2-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=D2,
 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 1,000000 1,0000000 1,0000000 1,0000000
## LnS 0,8504196 1,00000 1,000000 1,0000000 1,0000000 1,0000000
## S 0,3800648 0,00000 0,000000 0,0000000 0,0000000 1,0000000
## F1 0,1849107 0,00000 0,000000 0,0000000 0,0000000 0,0000000
## F2 0,3903793 0,00000 0,000000 1,0000000 1,0000000 0,0000000
## F3 0,5410551 1,00000 0,000000 0,0000000 1,0000000 1,0000000
## F4 0,2656981 0,00000 0,000000 0,0000000 0,0000000 0,0000000
## BF NA 1,00000 0,818078 0,7960744 0,4101629 0,4029967
## PostProbs NA 0,12880 0,105400 0,1025000 0,0528000 0,0519000
## R2 NA 0,65250 0,600900 0,6468000 0,6712000 0,6708000
## dim NA 3,00000 2,000000 3,0000000 4,0000000 4,0000000
## logmarg NA 10,88348 10,682683 10,6554178 9,9922796 9,9746535

LMfullBAS

##
## Call:
## bas.lm(formula = LnR ~ LnS + S + F1 + F2 + F3 + F4, data = D2,
## modelprior = uniform(), initprobs = "eplogp")
##
##
## Marginal Posterior Inclusion Probabilities:
## Intercept LnS S F1 F2 F3 F4
## 1,0000 0,8504 0,3801 0,1849 0,3904 0,5411 0,2657

Порог более 50% вероятности необходимости включения в модель преодолели только LnS и F3, а самая низкая вероятность включения в модель у F1. Посмотрим на их отношения по отдельности.

PerformanceAnalytics::chart.Correlation(D2)



Очевидно, что F1 также связан с F2, как и F2 с LnR, поэтому F1 не нужен. Повторим настройку без F1.

LMfullBAS2 <- bas.lm(LnR ~ LnS + S + F2 + F3 + F4,data=D2,
 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 1,000000 1,0000000 1,0000000 1,0000000
## LnS 0,8513335 1,00000 1,000000 1,0000000 1,0000000 1,0000000
## S 0,3740904 0,00000 0,000000 0,0000000 0,0000000 1,0000000
## F2 0,3935239 0,00000 0,000000 1,0000000 1,0000000 0,0000000
## F3 0,5391455 1,00000 0,000000 0,0000000 1,0000000 1,0000000
## F4 0,2615407 0,00000 0,000000 0,0000000 0,0000000 0,0000000
## BF NA 1,00000 0,818078 0,7960744 0,4101629 0,4029967
## PostProbs NA 0,15800 0,129300 0,1258000 0,0648000 0,0637000
## R2 NA 0,65250 0,600900 0,6468000 0,6712000 0,6708000
## dim NA 3,00000 2,000000 3,0000000 4,0000000 4,0000000
## logmarg NA 10,88348 10,682683 10,6554178 9,9922796 9,9746535

LMfullBAS2

##
## Call:
## bas.lm(formula = LnR ~ LnS + S + F2 + F3 + F4, data = D2, modelprior = uniform(),
## initprobs = "eplogp")
##
##
## Marginal Posterior Inclusion Probabilities:
## Intercept LnS S F2 F3 F4
## 1,0000 0,8513 0,3741 0,3935 0,5391 0,2615

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 F1 незначимо отличался от 0, а его исключение не расширило доверительные интервалы оставшихся параметров.

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

## 2,5% 97,5% beta
## Intercept 3,527980661 4,09662519 3,824598361
## LnS 0,000000000 0,95757558 0,577753736
## S 0,000000000 0,03418216 0,006530755
## F1 -0,099209475 0,18883295 0,006620820
## F2 -0,394185051 0,00781573 -0,074431095
## F3 -0,000226244 0,51192152 0,150550595
## F4 -0,592690483 0,06240892 -0,067821746
## attr(,"Probability")
## [1] 0,95
## attr(,"class")
## [1] "confint.bas"

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

## 2,5% 97,5% beta
## Intercept 3,5324400639 4,1042260668 3,824598361
## LnS 0,0000000000 0,9580944826 0,580218174
## S -0,0011529591 0,0331795829 0,006418993
## F2 -0,3806447668 0,0003252312 -0,074776970
## F3 -0,0009469883 0,5046347607 0,150754481
## F4 -0,5763512395 0,0779797023 -0,066839153
## 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$Years
BMA\_R2=round(cor(expconf.fit$Avg,pinks$R)^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=R), col='red')+
 xlab('Год')+ylab('Подход, млн рыб')+
 annotate(geom = 'text',family='Times New Roman',size = 5, x=2000,y=450,
 label = bquote(italic(R)^2%~~%.(BMA\_R2)))+
 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=log(R)), col='red',lwd=1)+
 xlab('Год')+ylab('Ln(Подход, млн рыб)')+
 theme\_calc(base\_size = 12, base\_family = 'Times New Roman')



Посмотрим на BIC различных конфигураций.

tLMR <- lm(LnR ~ LnS+S,data=D2)
tLM1 <- lm(LnR ~ LnS,data=D2)
tLM2 <- lm(LnR ~ LnS+F2+F3+F4,data=D2)
tLM3 <- lm(LnR ~ LnS+F3+F4,data=D2)
tLM4 <- lm(LnR ~ LnS+F3,data=D2)
tLMfull <- lm(LnR ~ LnS + S + F1 + F2 + F3 + F4,data=D2)
BIC(tLMR,tLM1,tLM2,tLM3,tLM4,tLMfull)

## df BIC
## tLMR 4 89,06260
## tLM1 3 86,78968
## tLM2 6 89,13519
## tLM3 5 88,19499
## tLM4 4 85,82984
## tLMfull 8 94,26588

Оптимальная модель на полном наборе по BIC включает только LnS и F3, также, как и модель, выбираемая по Байесову подходу. Доверительные интервалы её коэффициентов:

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

## 2,5% 97,5% beta
## Intercept 3,5424697963 4,1067269 3,8245984
## LnS 0,4682774197 0,9666573 0,7174674
## S 0,0000000000 0,0000000 0,0000000
## F2 0,0000000000 0,0000000 0,0000000
## F3 -0,0002848536 0,5555222 0,2776187
## F4 0,0000000000 0,0000000 0,0000000
## attr(,"Probability")
## [1] 0,95
## attr(,"class")
## [1] "confint.bas"

Однако даже в самой вероятной модели (HPM) доверительный интервал коэффициента F3 пересекает 0, что ставит под сомнение его надёжность.

## 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: 0x0000019ffa02bba0>
## <environment: 0x0000019ffa02e1d0>

Получим стартовые значения для оптимизатора.

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

## $a
## [1] 2,721085
##
## $b
## [1] 0,00341424

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=D2,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 2,72108494 1,734038466 4,2705977
## b 0,00341424 -0,006828118 0,0136566

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

## estimates 95% LCI 95% UCI
## a 2,72108494 1,706726441 4,08383017
## b 0,00341424 -0,005834134 0,01327465

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

ind <- srFuns('independence')
svI <- srStarts(R~S,data=D2,type='independence')
srI <- nls(LnR~log(ind(S,a)),data=D2,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 -39,17867 30 -38,93338 1 -0,24529 0,4906 0,4837

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
# 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 21,68294 30 21,35306 1 0,32988 0,4635 0,5012

По обоим тестам выходит, что фактор плотности имеет низкую статистическую значимость (p > 0.4).

Опишем модель прямой пропорции.

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

## a
## 2,436

round(confint(srI),3)

## Waiting for profiling to be done...

## 2,5% 97,5%
## 1,802 3,293

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')



Очевидно, что зависимость по Рикеру почти прямая, а найденный угловой коэффициент прямой пропорции визуально даже лучше подходит к наблюдениям в области высоких значений. В целом, прямая пропорция находится полностью внутри доверительного интервала зависимости по Рикеру.

## Добавим F3 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+F3,data=pinks)
tmp <- coef(tmp)
svR2 <- list(a=exp(tmp[[1]]),b=-tmp[[2]],c=tmp[[3]])
srR2 <- nls(LnR~log(rckr2(S,F3,a,b,c)),data=D2,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, F3, a, b, c))
##
## DfO logLikO DfA logLikA Df logLik Chisq Pr(>Chisq)
## 1vA 31 -39,1787 29 -37,5101 2 -1,6685 3,3371 0,18852
## 2vA 30 -38,9334 29 -37,5101 1 -1,4233 2,8465 0,09157 .
## ---
## 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, F3, a, b, c))
##
## DfO RSSO DfA RSSA Df SS F Pr(>F)
## 1vA 31 21,6829 29 19,5357 2 2,1473 1,5938 0,2204
## 2vA 30 21,3531 29 19,5357 1 1,8174 2,6979 0,1113

По отношению правдоподобия выходит, что учёт F3 не имеет статистически значимого преимущества.

coef(srR2)

## a b c
## 2,364541420 0,005609201 0,233442021

Эффект F3 в среднем положительный, 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 2,364541420 1,474043885 3,79340810
## b 0,005609201 -0,004737022 0,01595542
## c 0,233442021 -0,057234520 0,52411856

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

## estimates 95% LCI 95% UCI
## a 2,364541420 1,463156902 3,48839227
## b 0,005609201 -0,003167128 0,01544066
## c 0,233442021 -0,010460703 0,49979167

Он (с) пересёк ноль, также как и фактор плотности (b).

Сравним прогнозы по модели Рикера (R0) и по дополненной F3 (R\_F3).

pinks$R0 <- rckr(pinks$S,a=coef(srR))
pinks$R\_F3 <- rckr2(pinks$S,pinks$F2,a=coef(srR2))
pinks$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\_F3, col='red', lty=3))
legend('topleft', legend=c('Данные', 'Рикер', 'Рикер и F3'),
 col = c('black','green','red'), lty=1:3)



Связь с исходными данными прогнозов.

PerformanceAnalytics::chart.Correlation(pinks[,c('R','R0','R\_F3')])



Дополнение модели Рикера фактором F3 ухудшает связь наблюдённых подходов (R) с прогнозируемыми (R\_F3).

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

##
## Pearson's product-moment correlation
##
## data: R and R0
## t = 8,4854, df = 30, p-value = 1,81e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0,6951239 0,9194898
## sample estimates:
## cor
## 0,8401723

with(pinks, cor.test(R,R\_F3))

##
## Pearson's product-moment correlation
##
## data: R and R\_F3
## t = 4,5874, df = 30, p-value = 7,449e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0,3780336 0,8095341
## sample estimates:
## cor
## 0,6420887

## 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=D2,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 -38,9334 26 -35,9577 4 -2,9757 5,9514 0,2028

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
#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 21,3531 26 17,7292 4 3,6239 1,3286 0,2857

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