Methow Hatchery Steelhead: Effects of Smolt Age on Age at Maturity and Life-Cycle Survival

Eric Buhle, Chris Tatara, Kevin See, and …

March 7, 2018

Table of Contents

# Overview

This vignette describes the analysis of a mark-recapture study comparing alternative hatchery rearing trajectories for steelhead in the Methow River, Washington. From 2010 to 2015, 176930 steelhead smolts were released from Winthrop National Fish Hatchery after being reared to either age 1 (S1, the traditional method) or 2 (S2, a more typical age for wild-type smolts). The smolts were tagged with passive integrated transponder (PIT) tags which can be detected at arrays such as those in most mainstem Columbia River dams. The objectives are to understand the carryover effects (if any) of age at release on subsequent life history stages, and how these vary from year to year. In particular, we are interested in whether and how smolt age affects survival through the rest of the life cycle, both directly during downstream migration and early ocean residence, and perhaps indirectly during upstream spawning migration if smolt age is related to age at maturity.

One practical and conceptual difficulty is that adult age is only known for fish that were detected while migrating upstream. If adult age is to be used as a covariate of upstream survival, or if the model contains any random terms associated with year of return, the missing observations will be problematic. We could construct an integrated joint model for the survival and adult age data, treating the unobserved ages as latent states to be estimated. The Markov chain Monte Carlo algorithm we’re going to use to fit the models doesn’t handle discrete parameters, though, so the code would have to integrate out the latent states by summation – doable, but tedious and error-prone.

Serendipitously, it turns out that the joint posterior distribution of the integrated model factorizes into independent posteriors for adult age and for survival:

where contains the capture (detection) histories that provide information on survival, and the parameter vectors and indicate the classes of models we will use for adult age and survival, respectively. This separation strategy is valid only because missing adult age values are actually informative. The detection probability for adult salmonids at the mainstem Columbia River dams is effectively 1 (confirmed by model estimates, as we will see below), therefore non-detected adults did not survive to return. Conditional on unobserved adult age, then, the likelihood of the subsequent (non-) detection history is

As a constant, it can be removed from the likelihood. The upshot is that we can dis-integrate the model – that is, just model adult age and survival separately, combine their independent posteriors and get the joint posterior of the integrated model.

# Setup and data

First we’ll load some packages and custom functions.

# setwd(file.path("~","MethowSteelhead"))  
library(lubridate)  
library(gtools)  
library(Hmisc)  
library(vioplot)  
source("vioplot2.R")  
library(yarrr)  
library(rstan)  
library(rstanarm)  
library(loo)  
library(shinystan)  
source("extract1.R")  
source("stan\_mean.R")  
source("sim\_phi\_tot.R")

Next we’ll read in the dataset and clean it up a bit. methowSH is a data frame containing individual capture histories and covariates, and methowSHm contains the same capture histories in m-array format, with several of the mainstem Columbia River dams collapsed into “downstream” and “upstream” detection occasions.

# Read in capture history data (observations are dates)  
methowSH <- read.csv("methowSH.csv", header = T)  
for(i in 8:29)  
 methowSH[,i] <- as.Date(as.character(methowSH[,i]), "%m/%d/%Y")  
  
# "Fix" return year and age at return  
  
# (1) Some tags were detected in adult ladders the same year they were released  
# (usually in April). Change those detections to NA.  
rty\_test <- sweep(apply(methowSH[,13:29], 2, year), 1, methowSH$release\_year, "==")   
rty\_indx <- which(rty\_test & !is.na(rty\_test), arr.ind = T)  
rty\_indx[,"col"] <- rty\_indx[,"col"] + 12  
methowSH[rty\_indx] <- NA  
  
# (2) return\_year and adult\_age are missing for some fish that did, in fact, return as adults.   
# Fill in those missing values.  
rty <- apply(methowSH[,13:29], 1, function(x) ifelse(all(is.na(x)), NA, min(year(x), na.rm = T)))  
rty\_NA <- !is.na(rty) & (is.na(methowSH$return\_year) | is.na(methowSH$adult\_age))  
methowSH$return\_year[rty\_NA] <- rty[rty\_NA]  
methowSH$adult\_age[rty\_NA] <- methowSH$return\_year[rty\_NA] - methowSH$brood\_year[rty\_NA]  
  
# Convert dates to 0/1  
methowSH[,8:29] <- as.numeric(!is.na(methowSH[,8:29]))  
  
# Pool detections from MCJ:TWX (juvenile), TDA:WEA (adult), and MRC:BROOD (adult)  
methowSH <- cbind(methowSH[,1:6],  
 ocean\_age = factor(methowSH$return\_year - methowSH$release\_year),  
 methowSH[,7:12],  
 MCJTWX = as.numeric(apply(methowSH[,c("MCJ","JDJ","BON","TWX")] > 0, 1, any)),  
 methowSH[,13:19],  
 TDAWEA = as.numeric(apply(methowSH[,c("TDA","MCN","PRA",  
 "RIA","RRF","WEA")] > 0, 1, any)),  
 methowSH[,20:29],  
 MRCBRD = as.numeric(apply(methowSH[,c("MRC","MRT","MRW","SPRING",  
 "WFC","LOR","EWC","CRW","BROOD")] > 0,   
 1, any)))  
levels(methowSH$ocean\_age) <- c("1","2+","2+") # very few 3-ocean; group with 2-ocean  
  
# Convert crosstabs of ocean age by release year and smolt age to data frame  
# for use in posterior predictive checking of binomial GLMMs  
methowSHoa <- ftable(release\_year = methowSH$release\_year,   
 smolt\_age = methowSH$smolt\_age,   
 ocean\_age = methowSH$ocean\_age,   
 row.vars = 1:2, col.vars = 3, exclude = NULL)  
methowSHoa <- as.data.frame(as.matrix(methowSHoa))  
methowSHoa <- cbind(t(sapply(strsplit(row.names(methowSHoa), "\_"), identity)),  
 methowSHoa)  
names(methowSHoa) <- c("release\_year","smolt\_age","oa1","oa2plus","mort")  
methowSHoa$release\_year <- as.numeric(as.character(methowSHoa$release\_year))  
methowSHoa <- data.frame(methowSHoa[,1:2], smolt\_age\_num = as.numeric(methowSHoa$smolt\_age),  
 methowSHoa[,3:5])  
row.names(methowSHoa) <- NULL  
  
# Convert to m-array format, including the covariates (some redundant):  
# smolt\_age, brood\_year, release\_year, return\_year, adult\_age, adult\_age\_factor  
# (which combines 4 and 5 and arbitrarily fills NAs)  
methowSHm <- cbind(methowSH[,1:4],   
 return\_year = factor(methowSH$return\_year, exclude = NULL),  
 adult\_age = factor(methowSH$adult\_age, exclude = NULL),  
 adult\_age\_factor = factor(methowSH$adult\_age, exclude = NULL),  
 methowSH[,c("WNFH","RRJ","MCJTWX","BOA", "TDAWEA","LMR","MRCBRD")])  
levels(methowSHm$adult\_age\_factor) <- c("2","3","4+","4+","2")  
methowSHm <- aggregate(tag ~ smolt\_age + brood\_year + release\_year + return\_year + adult\_age +  
 adult\_age\_factor + WNFH + RRJ + MCJTWX + BOA + TDAWEA + LMR + MRCBRD,  
 data = methowSHm, length)  
names(methowSHm)[names(methowSHm)=="tag"] <- "n"  
methowSHm$return\_year <- as.numeric(as.character(methowSHm$return\_year))  
methowSHm$adult\_age <- as.numeric(as.character(methowSHm$adult\_age))

# GLMMs of ocean age

In this part of the analysis we ask whether a smolt’s age at release predicts how long it will spend at sea before maturing (its “ocean age”). To do this, we will collapse ocean age into two classes, 1 and 2+, noting that there are too few 3-salt adults to model as a distinct category.

table(release\_year = methowSH$release\_year,   
 ocean\_age = methowSH$return\_year-methowSH$release\_year)

ocean\_age  
release\_year 1 2 3  
 2010 227 85 0  
 2011 74 51 2  
 2012 133 80 1  
 2013 66 100 0  
 2014 80 65 0  
 2015 5 11 0

With ocean age as a binary outcome we can use hierarchical Bernoulli models, a type of generalized linear mixed-effects model (GLMM), to model the probability of maturing at ocean age 2+ as a function of smolt age while allowing for random interannual fluctuations among release years. To fit these models in a Bayesian framework we will use the [rstanarm](https://cran.r-project.org/web/packages/rstanarm/index.html) package, which is an interface to [Stan](http://mc-stan.org/) and its Hamiltonian Monte Carlo (HMC) sampling algorithm. [This vignette](https://cran.r-project.org/web/packages/rstanarm/vignettes/pooling.html#complete-pooling) specifically discusses Bernoulli GLMMs and the advantages of partial pooling via Bayesian hierarchical models.

We condition on survival to adulthood (and detection during upstream migration, although as we will see later, this occurs with probability close to 1) because only in this case is ocean age known. Thus the sample consists of returning tagged adults, and each observation is an individual steelhead.

## Candidate models

First we fit a “null” or intercept-only model where the intercept varies by release year.

fit\_oa0 <- stan\_glmer(ocean\_age ~ 1 | release\_year,   
 data = methowSH,  
 family = binomial("logit"),  
 prior\_intercept = normal(0,1.5),   
 prior\_covariance = decov(),  
 chains = 4, iter = 2000, warmup = 1000, cores = 4)

summary(fit\_oa0, probs = c(0.025,0.5,0.975))

Model Info:  
  
 function: stan\_glmer  
 family: binomial [logit]  
 formula: ocean\_age ~ 1 | release\_year  
 algorithm: sampling  
 priors: see help('prior\_summary')  
 sample: 4000 (posterior sample size)  
 observations: 980  
 groups: release\_year (6)  
  
Estimates:  
 mean sd 2.5% 50% 97.5%  
(Intercept) -0.2 0.3 -0.7 -0.2 0.4   
b[(Intercept) release\_year:2010] -0.8 0.3 -1.4 -0.7 -0.2   
b[(Intercept) release\_year:2011] -0.1 0.3 -0.8 -0.1 0.4   
b[(Intercept) release\_year:2012] -0.3 0.3 -0.9 -0.3 0.3   
b[(Intercept) release\_year:2013] 0.6 0.3 0.0 0.6 1.1   
b[(Intercept) release\_year:2014] 0.0 0.3 -0.7 0.0 0.6   
b[(Intercept) release\_year:2015] 0.5 0.5 -0.3 0.5 1.5   
Sigma[release\_year:(Intercept),(Intercept)] 0.5 0.5 0.1 0.3 1.7   
mean\_PPD 0.4 0.0 0.4 0.4 0.4   
log-posterior -647.7 2.5 -653.4 -647.4 -643.7   
  
Diagnostics:  
 mcse Rhat n\_eff  
(Intercept) 0.0 1.0 674   
b[(Intercept) release\_year:2010] 0.0 1.0 751   
b[(Intercept) release\_year:2011] 0.0 1.0 1063   
b[(Intercept) release\_year:2012] 0.0 1.0 854   
b[(Intercept) release\_year:2013] 0.0 1.0 836   
b[(Intercept) release\_year:2014] 0.0 1.0 848   
b[(Intercept) release\_year:2015] 0.0 1.0 2055   
Sigma[release\_year:(Intercept),(Intercept)] 0.0 1.0 1282   
mean\_PPD 0.0 1.0 4000   
log-posterior 0.1 1.0 970   
  
For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

The main thing to note here is that all the convergence diagnostics look good. The Gelman-Rubin potential scale reduction factor is essentially perfect and the effective sample size is adequate for all parameters. Next we add a term for the effect of smolt age (S1 or S2) on the probability of returning after 2+ years at sea.

fit\_oa1a <- stan\_glmer(ocean\_age ~ smolt\_age + (1 | release\_year),   
 data = methowSH,  
 family = binomial("logit"),  
 prior = normal(0,3),  
 prior\_intercept = normal(0,1.5),   
 prior\_covariance = decov(),  
 chains = 4, iter = 2000, warmup = 1000, cores = 4)

summary(fit\_oa1a, probs = c(0.025,0.5,0.975))

Model Info:  
  
 function: stan\_glmer  
 family: binomial [logit]  
 formula: ocean\_age ~ smolt\_age + (1 | release\_year)  
 algorithm: sampling  
 priors: see help('prior\_summary')  
 sample: 4000 (posterior sample size)  
 observations: 980  
 groups: release\_year (6)  
  
Estimates:  
 mean sd 2.5% 50% 97.5%  
(Intercept) 0.0 0.3 -0.6 -0.1 0.6   
smolt\_ageS2 -0.3 0.1 -0.6 -0.3 -0.1   
b[(Intercept) release\_year:2010] -0.7 0.3 -1.3 -0.7 -0.2   
b[(Intercept) release\_year:2011] -0.1 0.3 -0.7 0.0 0.5   
b[(Intercept) release\_year:2012] -0.3 0.3 -0.9 -0.3 0.3   
b[(Intercept) release\_year:2013] 0.5 0.3 -0.1 0.5 1.1   
b[(Intercept) release\_year:2014] 0.0 0.3 -0.6 0.0 0.6   
b[(Intercept) release\_year:2015] 0.5 0.4 -0.3 0.4 1.4   
Sigma[release\_year:(Intercept),(Intercept)] 0.5 0.5 0.1 0.3 1.7   
mean\_PPD 0.4 0.0 0.4 0.4 0.4   
log-posterior -646.1 2.7 -652.1 -645.8 -641.9   
  
Diagnostics:  
 mcse Rhat n\_eff  
(Intercept) 0.0 1.0 1207   
smolt\_ageS2 0.0 1.0 3357   
b[(Intercept) release\_year:2010] 0.0 1.0 1252   
b[(Intercept) release\_year:2011] 0.0 1.0 1398   
b[(Intercept) release\_year:2012] 0.0 1.0 1311   
b[(Intercept) release\_year:2013] 0.0 1.0 1387   
b[(Intercept) release\_year:2014] 0.0 1.0 1088   
b[(Intercept) release\_year:2015] 0.0 1.0 2399   
Sigma[release\_year:(Intercept),(Intercept)] 0.0 1.0 1012   
mean\_PPD 0.0 1.0 4000   
log-posterior 0.1 1.0 821   
  
For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

The posterior distribution for the smolt age coefficient is well below zero, so S2 smolts are more likely than their S1 counterparts to mature after only one year at sea. Again the convergence diagnostics look good. This model assumes the effect of smolt age is constant over time, but we could also allow it to vary across release years. With a sample size of only 6 years, we will assume the year-specific intercepts and smolt age coefficients are uncorrelated.

# Specifying independent random effects involves tricking the formula parser into   
# treating `smolt\_age` as a binary 0/1 numeric variable instead of a factor.  
fit\_oa1b <- stan\_glmer(ocean\_age ~ smolt\_age + (smolt\_age\_num || release\_year),   
 data = cbind(methowSH, smolt\_age\_num = as.numeric(methowSH$smolt\_age)),  
 family = binomial("logit"),  
 prior = normal(0,3),  
 prior\_intercept = normal(0,1.5),   
 prior\_covariance = decov(),  
 chains = 4, iter = 2000, warmup = 1000, cores = 4)

summary(fit\_oa1b, probs = c(0.025,0.5,0.975))

Model Info:  
  
 function: stan\_glmer  
 family: binomial [logit]  
 formula: ocean\_age ~ smolt\_age + (smolt\_age\_num || release\_year)  
 algorithm: sampling  
 priors: see help('prior\_summary')  
 sample: 4000 (posterior sample size)  
 observations: 980  
 groups: release\_year (6)  
  
Estimates:  
 mean sd 2.5% 50% 97.5%  
(Intercept) -0.1 0.3 -0.7 -0.1 0.6   
smolt\_ageS2 -0.3 0.2 -0.7 -0.3 0.2   
b[(Intercept) release\_year:2010] -0.2 0.4 -1.1 -0.2 0.5   
b[(Intercept) release\_year:2011] 0.0 0.4 -0.7 0.0 0.8   
b[(Intercept) release\_year:2012] -0.5 0.4 -1.4 -0.4 0.1   
b[(Intercept) release\_year:2013] 0.3 0.4 -0.3 0.3 1.0   
b[(Intercept) release\_year:2014] 0.0 0.3 -0.7 0.0 0.8   
b[(Intercept) release\_year:2015] 0.4 0.5 -0.4 0.3 1.5   
b[smolt\_age\_num release\_year:2010] -0.4 0.3 -0.9 -0.3 0.1   
b[smolt\_age\_num release\_year:2011] -0.1 0.2 -0.6 -0.1 0.4   
b[smolt\_age\_num release\_year:2012] 0.1 0.3 -0.3 0.1 0.7   
b[smolt\_age\_num release\_year:2013] 0.2 0.3 -0.3 0.2 0.7   
b[smolt\_age\_num release\_year:2014] 0.0 0.2 -0.6 0.0 0.4   
b[smolt\_age\_num release\_year:2015] 0.1 0.3 -0.5 0.1 0.8   
Sigma[release\_year:(Intercept),(Intercept)] 0.4 0.5 0.0 0.2 1.7   
Sigma[release\_year:smolt\_age\_num,smolt\_age\_num] 0.2 0.2 0.0 0.1 0.7   
mean\_PPD 0.4 0.0 0.4 0.4 0.4   
log-posterior -655.4 3.7 -663.5 -655.1 -649.0   
  
Diagnostics:  
 mcse Rhat n\_eff  
(Intercept) 0.0 1.0 1708   
smolt\_ageS2 0.0 1.0 1843   
b[(Intercept) release\_year:2010] 0.0 1.0 1970   
b[(Intercept) release\_year:2011] 0.0 1.0 2536   
b[(Intercept) release\_year:2012] 0.0 1.0 1971   
b[(Intercept) release\_year:2013] 0.0 1.0 2178   
b[(Intercept) release\_year:2014] 0.0 1.0 2545   
b[(Intercept) release\_year:2015] 0.0 1.0 2134   
b[smolt\_age\_num release\_year:2010] 0.0 1.0 1588   
b[smolt\_age\_num release\_year:2011] 0.0 1.0 2024   
b[smolt\_age\_num release\_year:2012] 0.0 1.0 1495   
b[smolt\_age\_num release\_year:2013] 0.0 1.0 2342   
b[smolt\_age\_num release\_year:2014] 0.0 1.0 2113   
b[smolt\_age\_num release\_year:2015] 0.0 1.0 2761   
Sigma[release\_year:(Intercept),(Intercept)] 0.0 1.0 2048   
Sigma[release\_year:smolt\_age\_num,smolt\_age\_num] 0.0 1.0 1569   
mean\_PPD 0.0 1.0 4000   
log-posterior 0.1 1.0 1069   
  
For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

There appears to be only modest evidence for a time-varying effect of smolt age. The posterior distributions of the year-specific slopes overlap substantially, and the interannual standard deviation is about half that of the intercept. At the same time, though, the uncertainty in the mean smolt age effect has increased; its 95% credible interval now straddles zero.

## Model selection

We can make a more formal comparison among these three candidate models in terms of their ability to predict new observations by using an information criterion such as [LOO](https://cran.r-project.org/web/packages/loo/vignettes/loo-example.html) (the approximate leave-one-out cross-validation score) implemented in the loo package. Like other information criteria (AIC, DIC, WAIC, et al.), LOO estimates the out-of-sample log predictive density using the same sample to which the model has been fitted. Specifically, the posterior distributions of the pointwise log-likelihoods are resampled using Pareto-smoothed importance sampling (hence the full name PSIS-LOO). This method integrates over posterior parameter uncertainty (avoiding plug-in point estimates, as in DIC) and also provides standard errors for the overall log predictive density summed across observations.

loo\_oa <- list(fit\_oa0 = loo(fit\_oa0), fit\_oa1a = loo(fit\_oa1a), fit\_oa1b = loo(fit\_oa1b))  
compare\_oa <- compare\_models(loos = loo\_oa)  
compare\_oa[,"looic"] <- compare\_oa[,"looic"] - min(compare\_oa[,"looic"])  
round(compare\_oa[,c("looic","se\_looic","p\_loo","se\_p\_loo")], 2)

looic se\_looic p\_loo se\_p\_loo  
fit\_oa1b 0.00 18.72 8.34 0.27  
fit\_oa1a 2.30 18.33 6.40 0.20  
fit\_oa0 6.44 17.92 5.61 0.21

The output shows the LOO information criterion, defined as -2 times the expected log predictive density , expressed as differences from the best score (i.e., the lowest point estimate). It also gives the effective number of parameters , along with the SEs. We can see that although the model with a time-varying intercept and slope is the best, the deviations among models are much smaller than their SEs, suggesting there is little practical difference in predictive utility.

Because LOO is based on the pointwise likelihood, we can also compute pairwise differences between models, which will be more discriminating (have smaller SEs) to the extent that the pointwise likelihoods under different models are positively correlated.

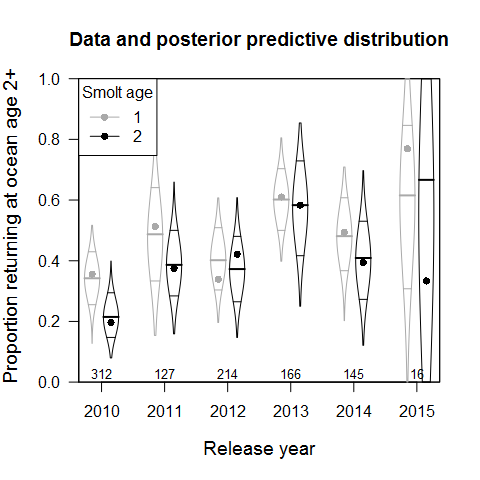
compair\_oa <- vector("list", length(loo\_oa) -1 )   
for(i in 1:length(compair\_oa))  
{  
 names(compair\_oa)[i] <- paste(row.names(compare\_oa)[c(i+1,i)], collapse = " vs. ")  
 compair\_oa[[i]] <- 2\*compare\_models(loos = loo\_oa[row.names(compare\_oa)[c(i+1,i)]])  
 names(compair\_oa[[i]])[1] <- "looic\_diff"  
}  
compair\_oa

$`fit\_oa1a vs. fit\_oa1b`  
  
Model comparison:   
(negative 'elpd\_diff' favors 1st model, positive favors 2nd)   
  
looic\_diff se   
 2.3 3.2   
  
$`fit\_oa0 vs. fit\_oa1a`  
  
Model comparison:   
(negative 'elpd\_diff' favors 1st model, positive favors 2nd)   
  
looic\_diff se   
 4.1 5.1

The SEs of the pairwise differences are indeed much smaller, but they still exceed the differences in . Not only does the time-varying slope not offer substantial improvement over the constant slope; the latter is not decisively better than the null model in terms of predictive utility, despite the strongly negative estimated smolt age effect. Perhaps this should not be too surprising; binary outcomes are an inherently “lossy” measure of an underlying process, so comparisons of predictive performance in this context are relatively insensitive and should be taken with a grain of salt.

## Plots

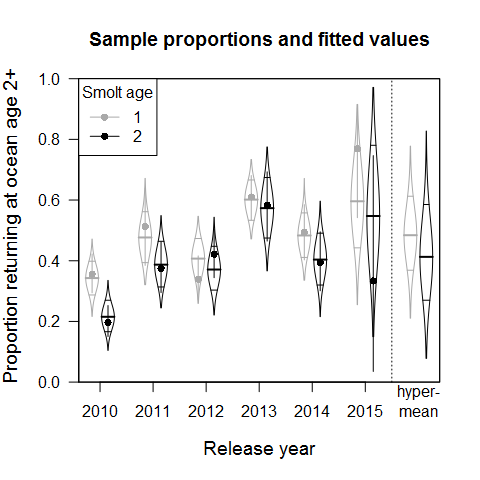
We can explore the implications of the fitted models graphically. One useful approach is to compare the observed data to the [posterior predictive distribution](https://cran.r-project.org/web/packages/rstanarm/vignettes/pooling.html#posterior-predictive-distribution) (PPD), i.e., the modeled distribution of hypothetical replicate data sets conditional on the observed data, integrated over posterior parameter uncertainty. Such posterior predictive checking can help to assess model adequacy and conversely, diagnose lack of fit. Here we’ll do this for the “full” Bernoulli GLMM (with time-varying intercept and smolt age effect).



**Figure S1:** Posterior predictive check for full ocean age GLMM.

Figure S1 shows the observed proportion of S1 and S2 smolts from each release year that matured at ocean age 2+ (the points), along with the corresponding posterior predictive distribution. The violin plots are similar to boxplots, but show the actual shape of the PPD as well as the median (thick line) and 5th and 95th percentiles (thin lines). Numbers above the *x*-axis give the adult sample size from each release year. Because the binary outcomes are conditionally IID given smolt age and year, each group is a sample from a binomial distribution. All the observations fall within the 90% credible intervals, indicating the model is well-calibrated; there is no sign the data are overdispersed relative to the binomial model. The posterior predictive uncertainty is much higher for release year 2015 because, as we saw in the table above, relatively few smolts released that year have returned as adults. The uncertainty reflects both the noisy data generated from the binomial distribution due to the small sample size and the posterior uncertainty in the estimated probability, as we will see in the next plot.

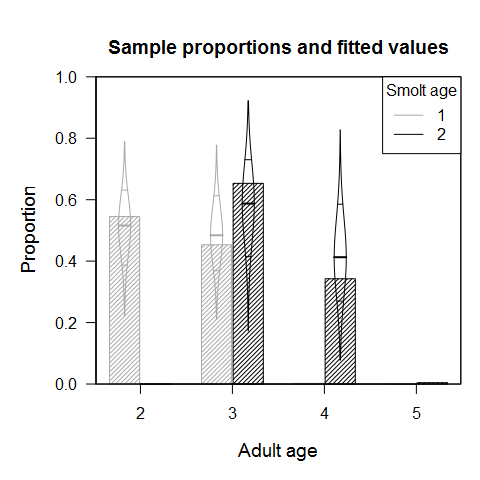
We can also examine the fitted values to better understand the model predictions. In this case, rather than simulating new data from the model, we are interested in the posterior distribution of the expected probability of maturing at ocean age 2+, i.e. the linear predictor inverse-link transformed () to the probability scale.



**Figure S2:** Posterior probability of maturing at ocean age 2+ under the full GLMM with time-varying intercept and slope.

This looks superficially similar to Figure S1, but we have added frequentist binomial 90% confidence intervals around the sample proportions to emphasize that they are also unpooled estimates of the underlying probabilities, and as such can be compared to the partially pooled model-based estimates. Here we can more clearly see how the full GLMM predicts interannual fluctuation in both the overall rate of age-2+ maturation (the time-varying intercept) and the effect of smolt age (the time-varying slope). S1 smolts are more likely than S2 smolts to spend 2+ years at sea, but in most years the difference is small relative to posterior uncertainty. The effect of hierarchical partial pooling is especially evident in 2015, where the sample point estimates indicate a large difference between S1 and S2 but the model down-weights this difference, shrinking it toward the hyper-mean based on its high uncertainty (due to small sample size).

It is simple to convert the model predictions into adult age rather than ocean age, and compare them to the observed adult age distribution of S1 and S2 smolts:



**Figure S3:** Predicted and observed distribution of age at maturity under the full GLMM with time-varying intercept and slope.

Even taking into account the (weak, variable) tendency for S2 smolts to return at a younger ocean age than S1 smolts, the older smolts are still clearly older at maturity. No S1 smolts were observed returning at age 4+, and no anadromous S2’s were observed at adult age 2 (this would amount to a “minijack” strategy of returning to spawn in the same year as outmigration).

# CJS models of survival

We turn now to modeling survival through the entire life cycle, from hatchery release to adult return to the Methow basin. We’ll use the Cormack-Jolly-Seber (CJS) framework to allow for imperfect detection of PIT-tagged fish while accommodating covariate effects (in particular, smolt and adult age) as well as hierarchically time-varying “random effects” on stage-specific survival () and detection () probabilities. We’ll fit these models in [Stan](mc-stan.org) using the [rstan](https://cran.r-project.org/web/packages/rstan/vignettes/rstan.html) package.

The detection “occasions” (really locations) are as follows:

WNFH RRJ MCJTWX BOA TDAWEA LMR MRCBRD.

Juvenile passage facilities at the mainstem Columbia River dams from McNary (MCJ) to Bonneville (BON), and the estuary towed array (TWX), are collapsed into an overall “downstream” corridor. Likewise, the adult ladders from The Dalles (TDA) to Wells (WEA) are collapsed into an “upstream” corridor. The reach from Winthrop National Fish Hatchery (WNFH, the point of release) to Rocky Reach Dam (RRJ) is kept distinct, as is the reach from WEA to the furthest downstream array in the Methow River (LMR), because we are interested in survival in these specific segments. The adult ladder at Bonneville (BOA) is distinct because survival from TWX to BOA serves as an operational definition of smolt-to-adult marine survival (SAR). The collapsed locations upstream of LMR, including detections during hook-and-line broodstock collection, allow us to estimate survival from WEA to the Methow. (As usual in CJS, survival and detection in the final interval are not separately identifiable through the likelihood.)

## Candidate models

### *Complete pooling*

We begin with a standard CJS model, with no covariates and all years pooled (no time-varying random effects). The following Stan code is borrowed, with a few modifications, from Section 15.3 of the [Stan 2.17.0 manual](https://github.com/stan-dev/stan/releases/download/v2.17.0/stan-reference-2.17.0.pdf). This version of the model uses the m-array data format (which the Stan manual calls the “collective CJS model”), consisting of cross-tabulated cell counts of each *unique* observed capture history. A capture history is a vector of binary outcomes indicating whether a tagged individual was detected () or not () on each of occasions.

# Cormack-Jolly-Seber Model (m-array data format)  
  
functions {  
 int first\_capture(int[] y\_i) {  
 for (t in 1:size(y\_i))  
 if(y\_i[t])  
 return t;  
 return 0;  
 }  
   
 int last\_capture(int[] y\_i) {  
 for (t\_rev in 0:(size(y\_i) - 1))   
 {  
 int t;  
 t = size(y\_i) - t\_rev;  
 if(y\_i[t])  
 return t;  
 }  
 return 0;  
 }  
   
 vector prob\_uncaptured(int T, vector p, vector phi) {  
 vector[T] chi;  
   
 chi[T] = 1.0;  
 for (t in 1:(T - 1))   
 {  
 int t\_curr;  
 int t\_next;  
 t\_curr = T - t;  
 t\_next = t\_curr + 1;  
 chi[t\_curr] = (1 - phi[t\_curr]) + phi[t\_curr] \* (1 - p[t\_next]) \* chi[t\_next];  
 }  
 return chi;  
 }  
}  
  
data {  
 int<lower=2> T; # number of capture events (includes marking)  
 int<lower=0> M; # number of unique capture histories  
 int<lower=0,upper=1> y[M,T]; # y[m,t]: history m captured at t  
 int<lower=1> n[M]; # n[m]: number of individuals with capture history y[m,]  
}  
  
transformed data {  
 int<lower=0,upper=T> first[M]; # first capture occasion  
 int<lower=0,upper=T> last[M]; # last capture occasion  
 int<lower=0,upper=T-1> last\_minus\_first[M]; # duh  
   
 for (m in 1:M)  
 {  
 first[m] = first\_capture(y[m,]);  
 last[m] = last\_capture(y[m,]);  
 last\_minus\_first[m] = last[m] - first[m];  
 }  
}  
  
parameters {  
 vector<lower=0,upper=1>[T-1] phi; # survival probabilities  
 vector<lower=0,upper=1>[T] p; # capture probabilities  
}  
  
transformed parameters {  
 vector<lower=0,upper=1>[T] chi; # chi[,t]: Pr[not captured > t | alive at t]  
  
 chi = prob\_uncaptured(T, p, phi);  
}  
  
model {  
 # implied uniform priors:  
 # phi ~ uniform(0,1)  
 # p ~ uniform(0,1)  
   
 # Likelihood of capture history  
 # marginalized over discrete latent states  
 for (m in 1:M)   
 {  
 if (last\_minus\_first[m] > 0) # if history m was recaptured  
 {  
 for(t in (first[m]+1):last[m])  
 {  
 target += n[m] \* log(phi[t-1]); # survival from t - 1 to t  
 target += n[m] \* bernoulli\_lpmf(y[m,t] | p[t]); # observation (captured or not)  
 }  
 }  
 target += n[m] \* log(chi[last[m]]); # Pr[not detected after last[m]]  
 }  
}  
  
generated quantities {  
 real lambda; # phi[T-1] and p[T] not identified, but product is  
 vector[M] LL; # log-likelihood of each capture history  
   
 lambda = phi[T-1] \* p[T];  
   
 # Likelihood of capture history, marginalized over discrete latent states  
 LL = rep\_vector(0,M);  
 for (m in 1:M)   
 {  
 if (last\_minus\_first[m] > 0) # if history m was recaptured  
 {  
 for(t in (first[m]+1):last[m])  
 {  
 LL[m] += n[m] \* log(phi[t-1]); # survival from t - 1 to t  
 LL[m] += n[m] \* bernoulli\_lpmf(y[m,t] | p[t]); # observation (captured or not)  
 }  
 }  
 LL[m] += n[m] \* log(chi[last[m]]); # Pr[not detected after last[m]]  
 }  
}

To fit the model, we call stan() with the appropriate data objects:

cjs\_all <- stan(file = "CJS-marray.stan",  
 data = list(T = 7, M = nrow(methowSHm),   
 y = methowSHm[,c("WNFH","RRJ","MCJTWX",  
 "BOA","TDAWEA","LMR","MRCBRD")],  
 n = methowSHm$n),  
 pars = c("phi","p","lambda","LL"),  
 chains = 3, cores = 3, iter = 2000, warmup = 1000,  
 control = list(adapt\_delta = 0.95, max\_treedepth = 12))

print(cjs\_all, pars = "LL", include = FALSE, prob = c(0.025,0.5,0.975))

Inference for Stan model: CJS-marray.  
3 chains, each with iter=2000; warmup=1000; thin=1;   
post-warmup draws per chain=1000, total post-warmup draws=3000.  
  
 mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat  
phi[1] 0.63 0.00 0.00 0.62 0.63 0.64 2252 1  
phi[2] 0.54 0.00 0.02 0.50 0.54 0.59 1651 1  
phi[3] 0.02 0.00 0.00 0.01 0.02 0.02 1875 1  
phi[4] 0.82 0.00 0.01 0.80 0.82 0.85 3000 1  
phi[5] 0.59 0.00 0.03 0.53 0.58 0.65 2074 1  
phi[6] 0.57 0.00 0.20 0.28 0.53 0.97 1983 1  
p[1] 0.50 0.01 0.29 0.02 0.50 0.97 3000 1  
p[2] 0.47 0.00 0.00 0.46 0.47 0.47 2298 1  
p[3] 0.34 0.00 0.02 0.31 0.34 0.37 1694 1  
p[4] 0.99 0.00 0.00 0.98 0.99 0.99 3000 1  
p[5] 1.00 0.00 0.00 0.99 1.00 1.00 3000 1  
p[6] 0.73 0.00 0.04 0.65 0.73 0.80 2034 1  
p[7] 0.56 0.00 0.21 0.28 0.52 0.97 1973 1  
lambda 0.28 0.00 0.02 0.23 0.28 0.32 2452 1  
lp\_\_ -176515.36 0.09 2.67 -176521.63 -176515.00 -176511.23 981 1  
  
Samples were drawn using NUTS(diag\_e) at Sat Mar 03 12:43:36 2018.  
For each parameter, n\_eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor on split chains (at   
convergence, Rhat=1).

The convergence diagnostics look excellent, but if they didn’t, we could explore the posterior samples using the shinystan package. Note that the detection probability at WNFH p[1] is just a placeholder and does not enter into the likelihood, so its posterior is the same as its prior. Also, as previously noted, survival in the final reach from LMR to MRCBRD (phi[6]) and detection at MRCBRD (p[7]) are not separately identified; only their product lambda is. As we would expect, the lowest stage-specific survival is SAR (phi[3]). Juvenile downstream detection probabilities are moderately high, but detection of adults at Bonneville (p[4]) and the other mainstem Columbia River dams collectively (p[5]) is essentially certain. As we shall see, this result is robust to increasing model complexity. This conveniently allows us to assume that fish not detected as adults did not survive to maturity (or did not return to the Columbia Basin), thereby avoiding having to model adult age as a partially unobserved covariate. Instead we can break the problem into two independent likelihood components: one for adult age (the GLMMs) and another for survival (CJS).

### *Hierarchical CJS models*

Now we consider more complex CJS models that include covariates and time-varying survival and detection probabilities. The cross-tabulation used to define the m-array includes these covariates and grouping factors. This model can be written as

This says that for the -th unique capture history in the m-array (), survival from occasion to () is modeled, on the logit scale, as a regression on a vector of covariates with coefficients , plus a normally distributed random effect associated with the group () to which capture history belongs. The model for the detection probability on occasion is similar; note that the covariates and grouping factors need not be the same for and , nor for different occasions. We also allow a special group ID which causes the parameters for capture histories belonging to that “group” to be fixed at and , respectively. We’ll use this to encode the assumption that tagged fish not detected as returning adults did not survive to maturity, thus the likelihood of their subsequent capture history is 1.

The Stan code for this hierarchical CJS model is an extension of the simple CJS code above.

# Cormack-Jolly-Seber Model (m-array data format) with covariates and  
# random effects on time-specific survival and capture probabilities   
# (possibly different grouping variables for each phi and p)  
# as well as a code that indicates phi = 0, p = 1  
  
functions {  
 int first\_capture(int[] y\_i) {  
 for (t in 1:size(y\_i))  
 if (y\_i[t])  
 return t;  
 return 0;  
 }  
   
 int last\_capture(int[] y\_i) {  
 for (t\_rev in 0:(size(y\_i) - 1))   
 {  
 int t;  
 t = size(y\_i) - t\_rev;  
 if (y\_i[t])  
 return t;  
 }  
 return 0;  
 }  
   
 row\_vector prob\_uncaptured(int T, row\_vector p, row\_vector phi) {  
 row\_vector[T] chi;  
   
 chi[T] = 1.0;  
 for (t in 1:(T - 1))   
 {  
 int t\_curr;  
 int t\_next;  
 t\_curr = T - t;  
 t\_next = t\_curr + 1;  
 chi[t\_curr] = (1 - phi[t\_curr]) + phi[t\_curr] \* (1 - p[t\_next]) \* chi[t\_next];  
 }  
 return chi;  
 }  
}  
  
data {  
 int<lower=2> T; # number of capture events (includes marking)  
 int<lower=0> M; # number of unique capture histories  
 int<lower=1> K; # total number of covariates  
 matrix[M,K] X; # covariates (first column is 1 for intercept)  
 int<lower=0,upper=1> indX\_phi[K,T-1]; # use covariate k for phi[t]?  
 int<lower=0> group\_phi[M,T-1]; # phi group IDs for each unique capture history  
 int<lower=0,upper=1> indX\_p[K,T]; # use covariate k for p[t]?  
 int<lower=0> group\_p[M,T]; # p group IDs for each unique capture history  
 int<lower=0,upper=1> y[M,T]; # y[m,t]: history m captured at t  
 int<lower=1> n[M]; # n[m]: number of individuals with capture history y[m,]  
}  
  
transformed data {  
 int<lower=1> K\_phi; # number of covariates for phi  
 int<lower=1> K\_p; # number of covariates for p  
 int<lower=1> J\_phi; # number of groups for phi  
 int<lower=1> J\_p; # number of groups for p  
 int<lower=0,upper=T> first[M]; # first capture occasion  
 int<lower=0,upper=T> last[M]; # last capture occasion  
 int<lower=0,upper=T-1> last\_minus\_first[M]; # duh  
   
 K\_phi = sum(to\_array\_1d(indX\_phi));  
 K\_p = sum(to\_array\_1d(indX\_p));  
 J\_phi = max(to\_array\_1d(group\_phi));  
 J\_p = max(to\_array\_1d(group\_p));  
   
 for (m in 1:M)  
 {  
 first[m] = first\_capture(y[m,]);  
 last[m] = last\_capture(y[m,]);  
 last\_minus\_first[m] = last[m] - first[m];  
 }  
}  
  
parameters {  
 vector[K\_phi] beta\_vec; # regression coefficients for logit(phi)  
 vector<lower=0>[T-1] sigma; # among-group SDs of logit(phi[,t])  
 matrix[J\_phi,T-1] epsilon\_z; # group-specific random effects on phi (z-scores)  
 vector[K\_p] b\_vec; # regression coefficients for logit(p)  
 vector<lower=0>[T] s; # among-group SDs of logit(p[,t])  
 matrix[J\_p,T] e\_z; # group-specific random effects on p (z-scores)  
}  
  
transformed parameters {  
 matrix[K,T-1] beta; # regression coefficients for logit(phi) with structural zeros  
 matrix[K,T] b; # regression coefficients for logit(p) with structural zeros  
 matrix[M,T-1] phi; # phi[,t]: Pr[alive at t + 1 | alive at t]  
 matrix[M,T] p; # p[,t]: Pr[captured at t | alive at t] (note p[,1] not used in model)  
 matrix[M,T] chi; # chi[,t]: Pr[not captured > t | alive at t]  
 vector[M] LL; # log-likelihood of each capture history  
   
 # Fill in sparse beta and b matrices  
 beta = rep\_matrix(0, K, T-1);  
 b = rep\_matrix(0, K, T);  
   
 {  
 int np\_phi;  
 int np\_p;  
   
 np\_phi = 1;  
 np\_p = 1;  
   
 for(k in 1:K)  
 {  
 for(t in 1:(T-1))  
 if(indX\_phi[k,t])  
 {  
 beta[k,t] = beta\_vec[np\_phi];  
 np\_phi = np\_phi + 1;  
 }  
   
 for(t in 1:T)  
 if(indX\_p[k,t])  
 {  
 b[k,t] = b\_vec[np\_p];  
 np\_p = np\_p + 1;  
 }  
 }  
 }  
   
 # Hierarchical logistic regression for phi and p  
 for(m in 1:M)  
 {  
 for(t in 1:(T-1))  
 {  
 if(group\_phi[m,t] == 0) # special code: fix survival to 0 and detection to 1  
 phi[m,t] = 0;  
 else  
 phi[m,t] = inv\_logit(X[m,] \* beta[,t] + sigma[t] \* epsilon\_z[group\_phi[m,t],t]);  
 }  
   
 for(t in 1:T)  
 {  
 if(group\_p[m,t] == 0)  
 p[m,t] = 1;  
 else  
 p[m,t] = inv\_logit(X[m,] \* b[,t] + s[t] \* e\_z[group\_p[m,t],t]);  
 }  
   
 chi[m,] = prob\_uncaptured(T, p[m,], phi[m,]);  
 }  
   
 # Likelihood of capture history, marginalized over discrete latent states  
 LL = rep\_vector(0,M);  
   
 for (m in 1:M)   
 {  
 if (last\_minus\_first[m] > 0) # if history m was recaptured  
 {  
 for(t in (first[m]+1):last[m])  
 {  
 LL[m] += n[m] \* log(phi[m,t-1]); # survival from t - 1 to t  
 LL[m] += n[m] \* bernoulli\_lpmf(y[m,t] | p[m,t]); # observation (captured or not)  
 }  
 }  
 LL[m] += n[m] \* log(chi[m,last[m]]); # Pr[not detected after last[m]]  
 }  
}  
  
model {  
 # Priors   
   
 # log Jacobian of logit transform for phi[t] intercepts  
 # implies phi[t] ~ Unif(0,1) given all covariates are at their sample means  
 target += log\_inv\_logit(beta\_vec[1:(T-1)]) + log1m\_inv\_logit(beta\_vec[1:(T-1)]);  
 if(K\_phi > T - 1)  
 beta\_vec[T:K\_phi] ~ normal(0,3);   
 sigma ~ normal(0,3);   
 to\_vector(epsilon\_z) ~ normal(0,1); # implies logit(phi[m,t]) ~ N(logit(mu\_phi[t]), sigma);  
 # log Jacobian of logit transform for p[t] intercepts  
 # implies p[t] ~ Unif(0,1) given all covariates are at their sample means  
 target += log\_inv\_logit(b\_vec[1:T]) + log1m\_inv\_logit(b\_vec[1:T]);  
 if(K\_p > T)  
 b\_vec[(T+1):K\_p] ~ normal(0,3);   
 s ~ normal(0,3);   
 to\_vector(e\_z) ~ normal(0,1); # implies logit(p[m,t]) ~ N(logit(mu\_p[t]), s);  
   
 # Likelihood of capture history added to log posterior  
 target += sum(LL);  
}

Next we’ll assemble some data objects that will come in handy for fitting the models. These include mean-centered covariates with missing adult ages arbitrarily filled in (since these values won’t enter into the likelihood) as well as matrices of grouping variables for and at each occasion. We will assume that downstream smolt survivals and SAR are grouped by release year, while upstream survivals are grouped by return year. Similarly, downstream detection probabilities are grouped by release year and upstream ones (including BOA) by return year. We use group 0 to fix phi[,4:6] to 0 and p[,4:7] to 1 for capture histories that were not detected at any adult facilities or upstream arrays.

# Year indices as grouping variables for phi and p random effects  
# Replace NA return years with arbitrary index that will not enter into the likelihood  
# (these capture histories contain no information on upstream survival and detection probs)  
release\_year <- as.numeric(factor(methowSHm$release\_year))  
return\_year <- ifelse(!is.na(methowSHm$return\_year),  
 as.numeric(factor(methowSHm$return\_year)),  
 0) # code NA as zero  
  
# phi: downstream, SAR grouped by release year; upstream grouped by return year  
group\_phi <- cbind(matrix(rep(release\_year, 3), ncol = 3), matrix(rep(return\_year, 3), ncol = 3))  
  
# p: downstream grouped by release year; upstream grouped by return year  
group\_p <- cbind(matrix(rep(release\_year, 3), ncol = 3), matrix(rep(return\_year, 4), ncol = 4))  
  
# Centered predictors for CJS models  
smolt\_age <- scale(as.numeric(methowSHm$smolt\_age), scale=F)  
adult\_age <- na.replace(scale(methowSHm$adult\_age, scale = F), 0) # arbitrary NA value

Finally, we’re ready to fit some models! We begin with a model with no covariate effects, just the time-varying intercepts of and .

cjs\_fixNA0 <- stan(file = "CJS-marray-phiXRE-pXRE-fixNA.stan",  
 data = list(T = 7, M = nrow(methowSHm), K = 1,  
 X = matrix(1, nrow(methowSHm), 1),  
 indX\_phi = matrix(1,1,6),  
 group\_phi = group\_phi,   
 indX\_p = matrix(1,1,7),  
 group\_p = group\_p,  
 y = methowSHm[,c("WNFH","RRJ","MCJTWX",  
 "BOA","TDAWEA","LMR","MRCBRD")],  
 n = methowSHm$n),  
 pars = c("beta","sigma","b","s","LL"),  
 chains = 3, cores = 3, iter = 1500, warmup = 500,  
 control = list(adapt\_delta = 0.95, max\_treedepth = 12))

print(cjs\_fixNA0, pars = "LL", include = FALSE, prob = c(0.025,0.5,0.975))

Inference for Stan model: CJS-marray-phiXRE-pXRE-fixNA.  
3 chains, each with iter=1500; warmup=500; thin=1;   
post-warmup draws per chain=1000, total post-warmup draws=3000.  
  
 mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat  
beta[1,1] 0.54 0.00 0.17 0.17 0.54 0.89 1207 1.00  
beta[1,2] 0.39 0.01 0.31 -0.18 0.36 1.10 1344 1.00  
beta[1,3] -4.16 0.02 0.63 -5.19 -4.25 -2.63 713 1.00  
beta[1,4] 1.56 0.00 0.13 1.31 1.56 1.80 2348 1.00  
beta[1,5] 0.34 0.00 0.21 -0.08 0.33 0.78 2061 1.00  
beta[1,6] 0.73 0.04 1.41 -1.44 0.53 4.18 1392 1.00  
sigma[1] 0.40 0.01 0.19 0.19 0.35 0.89 1355 1.00  
sigma[2] 0.53 0.01 0.42 0.05 0.44 1.64 1343 1.00  
sigma[3] 1.32 0.02 0.61 0.60 1.17 2.86 955 1.00  
sigma[4] 0.17 0.00 0.15 0.01 0.14 0.56 1239 1.00  
sigma[5] 0.35 0.01 0.28 0.02 0.28 1.07 1023 1.00  
sigma[6] 2.20 0.03 1.24 0.46 1.92 5.44 1600 1.00  
b[1,1] -0.01 0.04 1.97 -3.95 -0.03 3.93 3000 1.00  
b[1,2] -0.14 0.01 0.28 -0.67 -0.15 0.46 1244 1.00  
b[1,3] -0.77 0.01 0.31 -1.41 -0.77 -0.15 1363 1.01  
b[1,4] 4.20 0.03 1.04 1.57 4.32 6.05 1267 1.00  
b[1,5] 6.51 0.04 1.63 3.29 6.40 10.18 1826 1.00  
b[1,6] 1.04 0.01 0.42 0.26 1.04 1.90 2037 1.00  
b[1,7] 0.65 0.03 1.36 -1.40 0.44 3.98 1545 1.00  
s[1] 2.34 0.03 1.82 0.07 1.92 6.65 3000 1.00  
s[2] 0.64 0.01 0.27 0.31 0.58 1.39 1362 1.00  
s[3] 0.69 0.01 0.32 0.31 0.62 1.52 1241 1.00  
s[4] 2.16 0.05 1.35 0.22 1.94 5.45 625 1.01  
s[5] 1.60 0.03 1.47 0.05 1.13 5.65 2001 1.00  
s[6] 0.88 0.01 0.50 0.19 0.80 2.08 1345 1.00  
s[7] 2.16 0.03 1.20 0.49 1.90 5.20 1925 1.00  
lp\_\_ -173299.77 0.39 9.98 -173320.13 -173299.62 -173280.27 672 1.01  
  
Samples were drawn using NUTS(diag\_e) at Fri Mar 09 14:56:24 2018.  
For each parameter, n\_eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor on split chains (at   
convergence, Rhat=1).

The regression coefficients beta and b are indexed by covariate (rows, in this case just one for the intercept) and occasion (columns), while the interannual SDs sigma and s are indexed by occasion. The usual caveats about phi[,1] being a placeholder and phi[,6] and p[,7] being jointly nonidentified apply to the corresponding intercepts and SDs as well. We can see that some parameters, such as SAR (sigma[3]), are much more variable (on the unconstrained logit link scale) than others. Also, the adult detection probabilities at BOA (b[1,4]) and the mainstem Columbia (b[1,5]) remain close to 1.

Next we’ll consider an effect of smolt age on the reach-specific juvenile survivals from release to the estuary, and on SAR (phi[,1:3]).

cjs\_fixNA1 <- stan(file = "CJS-marray-phiXRE-pXRE-fixNA.stan",  
 data = list(T = 7, M = nrow(methowSHm), K = 2,  
 X = model.matrix(~ smolt\_age),  
 indX\_phi = rbind(1, c(1,1,1,0,0,0)),  
 group\_phi = group\_phi,   
 indX\_p = rbind(rep(1,7), 0),  
 group\_p = group\_p,  
 y = methowSHm[,c("WNFH","RRJ","MCJTWX",  
 "BOA","TDAWEA","LMR","MRCBRD")],  
 n = methowSHm$n),  
 pars = c("beta","sigma","b","s","LL"),  
 chains = 3, cores = 3, iter = 1500, warmup = 500,  
 control = list(adapt\_delta = 0.95, max\_treedepth = 12))

print(cjs\_fixNA1, pars = "beta", prob = c(0.025,0.5,0.975))

Inference for Stan model: CJS-marray-phiXRE-pXRE-fixNA.  
3 chains, each with iter=1500; warmup=500; thin=1;   
post-warmup draws per chain=1000, total post-warmup draws=3000.  
  
 mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat  
beta[1,1] 0.55 0.01 0.20 0.17 0.55 0.93 530 1  
beta[1,2] 1.67 0.03 1.10 -0.51 1.66 3.89 1228 1  
beta[1,3] -4.20 0.03 0.82 -5.55 -4.31 -2.35 575 1  
beta[1,4] 1.56 0.00 0.12 1.30 1.56 1.81 1849 1  
beta[1,5] 0.33 0.01 0.22 -0.11 0.33 0.77 1069 1  
beta[1,6] 0.69 0.04 1.34 -1.42 0.49 3.73 1025 1  
beta[2,1] 0.19 0.00 0.02 0.16 0.19 0.23 3000 1  
beta[2,2] 0.49 0.00 0.05 0.40 0.49 0.61 3000 1  
beta[2,3] -0.28 0.00 0.07 -0.41 -0.28 -0.15 3000 1  
beta[2,4] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,5] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,6] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
  
Samples were drawn using NUTS(diag\_e) at Fri Mar 09 15:33:20 2018.  
For each parameter, n\_eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor on split chains (at   
convergence, Rhat=1).

The beta and b matrices now have a second row, consisting of the smolt age coefficients. These are only estimated for the first three survivals (and not at all for detection); the others are “switched off” by setting them to zero, hence their diagnostics are NaN. The estimates indicate that S2 smolts had higher survival than S1 smolts from release to RRJ (beta[2,1]) and RRJ to the estuary (beta[2,2]), but they had a *lower* SAR (beta[2,2]). This is perhaps unexpected, and interesting. We might suspect that this comparison is confounded by different lengths of time spent in the ocean. But the GLMMs showed that S1 smolts are generally more likely to spend two years in saltwater before maturing, so if anything they would be expected to have lower SAR; instead, we see the opposite.

Next we’ll consider an effect of adult age on the upstream reach-specific survivals. There are two ways we might represent adult age: as a categorical factor or a continuous variable. The continuous encoding assumes a (logit-) linear relationship between age and survival, while the factor encoding allows nonlinearity (which might arise from size-selective fisheries, for example). We’ll start with the latter. Since there are very few 5-year-old adults

table(release\_year = methowSH$release\_year, adult\_age = methowSH$adult\_age)

adult\_age  
release\_year 2 3 4 5  
 2010 96 184 32 0  
 2011 19 74 33 1  
 2012 74 97 42 1  
 2013 46 92 28 0  
 2014 40 79 26 0  
 2015 3 12 1 0

we’ll use the factor levels 2, 3, 4+, producing 2 contrast coefficients.

cjs\_fixNA2a <- stan(file = "CJS-marray-phiXRE-pXRE-fixNA.stan",  
 data = list(T = 7, M = nrow(methowSHm), K = 3,  
 X = model.matrix(~ adult\_age\_factor, data = methowSHm),  
 indX\_phi = rbind(1, c(0,0,0,1,1,0), c(0,0,0,1,1,0)),  
 group\_phi = group\_phi,   
 indX\_p = rbind(rep(1,7), 0, 0),  
 group\_p = group\_p,  
 y = methowSHm[,c("WNFH","RRJ","MCJTWX",  
 "BOA","TDAWEA","LMR","MRCBRD")],  
 n = methowSHm$n),  
 pars = c("beta","sigma","b","s","LL"),  
 chains = 3, cores = 3, iter = 1500, warmup = 500,  
 control = list(adapt\_delta = 0.95, max\_treedepth = 12))

print(cjs\_fixNA2a, pars = "beta", prob = c(0.025,0.5,0.975))

Inference for Stan model: CJS-marray-phiXRE-pXRE-fixNA.  
3 chains, each with iter=1500; warmup=500; thin=1;   
post-warmup draws per chain=1000, total post-warmup draws=3000.  
  
 mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat  
beta[1,1] 0.54 0.01 0.18 0.17 0.54 0.89 1163 1  
beta[1,2] 0.39 0.01 0.33 -0.14 0.35 1.17 1692 1  
beta[1,3] -4.14 0.03 0.68 -5.18 -4.22 -2.51 453 1  
beta[1,4] 1.45 0.00 0.19 1.07 1.45 1.81 3000 1  
beta[1,5] -0.29 0.01 0.30 -0.88 -0.29 0.27 1769 1  
beta[1,6] 0.64 0.04 1.37 -1.50 0.41 3.96 1387 1  
beta[2,1] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,2] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,3] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,4] 0.17 0.00 0.20 -0.21 0.18 0.55 3000 1  
beta[2,5] 0.83 0.00 0.22 0.42 0.82 1.29 3000 1  
beta[2,6] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,1] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,2] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,3] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,4] 0.14 0.00 0.27 -0.38 0.14 0.67 3000 1  
beta[3,5] 1.00 0.01 0.31 0.42 0.99 1.63 3000 1  
beta[3,6] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
  
Samples were drawn using NUTS(diag\_e) at Fri Mar 09 16:01:18 2018.  
For each parameter, n\_eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor on split chains (at   
convergence, Rhat=1).

Age has no clear effect on survival from BOA to WEA (beta[2:3,4]), but older adults have a clear survival advantage in the reach from WEA to the lower Methow (beta[2:3,5]). Comparing rows 2 and 3, each of which represents the effect of a 1-year increment, it looks like the effect of adult age actually *is* more or less linear. We can therefore fit the model with adult age as a continuous predictor:

cjs\_fixNA2b <- stan(file = "CJS-marray-phiXRE-pXRE-fixNA.stan",  
 data = list(T = 7, M = nrow(methowSHm), K = 2,  
 X = model.matrix(~ adult\_age),  
 indX\_phi = rbind(1, c(0,0,0,1,1,0)),  
 group\_phi = group\_phi,   
 indX\_p = rbind(rep(1,7), 0),  
 group\_p = group\_p,  
 y = methowSHm[,c("WNFH","RRJ","MCJTWX",  
 "BOA","TDAWEA","LMR","MRCBRD")],  
 n = methowSHm$n),  
 pars = c("beta","sigma","b","s","LL"),  
 chains = 3, cores = 3, iter = 1500, warmup = 500,  
 control = list(adapt\_delta = 0.95, max\_treedepth = 12))

print(cjs\_fixNA2b, pars = "beta", prob = c(0.025,0.5,0.975))

Inference for Stan model: CJS-marray-phiXRE-pXRE-fixNA.  
3 chains, each with iter=1500; warmup=500; thin=1;   
post-warmup draws per chain=1000, total post-warmup draws=3000.  
  
 mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat  
beta[1,1] 0.53 0.00 0.18 0.17 0.54 0.85 1239 1.00  
beta[1,2] 0.39 0.01 0.31 -0.12 0.36 1.10 1159 1.00  
beta[1,3] -4.14 0.02 0.65 -5.22 -4.23 -2.47 737 1.01  
beta[1,4] 1.57 0.00 0.12 1.34 1.57 1.81 3000 1.00  
beta[1,5] 0.39 0.01 0.25 -0.11 0.38 0.91 1904 1.00  
beta[1,6] 0.65 0.04 1.36 -1.47 0.43 3.78 1376 1.00  
beta[2,1] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,2] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,3] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,4] 0.06 0.00 0.13 -0.20 0.06 0.31 3000 1.00  
beta[2,5] 0.56 0.00 0.16 0.24 0.55 0.89 3000 1.00  
beta[2,6] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
  
Samples were drawn using NUTS(diag\_e) at Fri Mar 09 16:26:20 2018.  
For each parameter, n\_eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor on split chains (at   
convergence, Rhat=1).

Note that the age coefficients are similar in magnitude to those in the previous model (but more precise), and the qualitative inferences are the same. Lastly we’ll consider a “full” model combining the smolt age and continuous adult age terms.

cjs\_fixNA3 <- stan(file = "CJS-marray-phiXRE-pXRE-fixNA.stan",  
 data = list(T = 7, M = nrow(methowSHm), K = 3,  
 X = model.matrix(~ smolt\_age + adult\_age),  
 indX\_phi = rbind(1, c(1,1,1,0,0,0), c(0,0,0,1,1,0)),  
 group\_phi = group\_phi,   
 indX\_p = rbind(rep(1,7), 0, 0),  
 group\_p = group\_p,  
 y = methowSHm[,c("WNFH","RRJ","MCJTWX","BOA","TDAWEA","LMR","MRCBRD")],  
 n = methowSHm$n),  
 pars = c("beta","sigma","epsilon\_z","b","s","e\_z","LL"),  
 chains = 3, cores = 3, iter = 1500, warmup = 500,  
 control = list(adapt\_delta = 0.95, max\_treedepth = 12))

print(cjs\_fixNA3, pars = "beta", prob = c(0.025,0.5,0.975))

Inference for Stan model: CJS-marray-phiXRE-pXRE-fixNA.  
3 chains, each with iter=1500; warmup=500; thin=1;   
post-warmup draws per chain=1000, total post-warmup draws=3000.  
  
 mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat  
beta[1,1] 0.55 0.01 0.18 0.20 0.55 0.92 872 1  
beta[1,2] 1.68 0.03 1.11 -0.49 1.67 3.86 1044 1  
beta[1,3] -4.28 0.03 0.77 -5.45 -4.38 -2.40 752 1  
beta[1,4] 1.57 0.00 0.13 1.30 1.57 1.81 1940 1  
beta[1,5] 0.39 0.01 0.25 -0.08 0.38 0.94 1638 1  
beta[1,6] 0.64 0.04 1.40 -1.52 0.42 3.93 978 1  
beta[2,1] 0.19 0.00 0.02 0.15 0.19 0.23 3000 1  
beta[2,2] 0.49 0.00 0.05 0.40 0.49 0.60 3000 1  
beta[2,3] -0.28 0.00 0.07 -0.42 -0.28 -0.15 3000 1  
beta[2,4] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,5] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[2,6] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,1] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,2] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,3] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
beta[3,4] 0.06 0.00 0.13 -0.20 0.06 0.31 3000 1  
beta[3,5] 0.56 0.00 0.16 0.26 0.56 0.89 3000 1  
beta[3,6] 0.00 0.00 0.00 0.00 0.00 0.00 3000 NaN  
  
Samples were drawn using NUTS(diag\_e) at Thu Mar 15 17:04:45 2018.  
For each parameter, n\_eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor on split chains (at   
convergence, Rhat=1).

## Model selection

We can use LOO to compare the alternative CJS models, using the pointwise log-likelihoods that we calculated in the Stan code. Here an “observation” is the capture history for an individual tagged fish. The idea is that we’re asking how well the candidate models would do at predicting the data for an unobserved individual, as opposed to an all the individuals in a cell of the m-array (which is an arbitrary post-sampling construct).

# Because all histories within a cell of the m-array have the same likelihood,  
# we only need to call loo() on the posterior samples of the unique log-likelihoods.  
# This is waaaaay faster than calling it on the full individual-level matrix.   
# We can then edit the loo object so it corresponds to the individual history data.  
loo\_cjs <- lapply(list(cjs\_all = cjs\_all,  
 cjs\_fixNA0 = cjs\_fixNA0,   
 cjs\_fixNA1 = cjs\_fixNA1,   
 cjs\_fixNA2a = cjs\_fixNA2a,   
 cjs\_fixNA2b = cjs\_fixNA2b,   
 cjs\_fixNA3 = cjs\_fixNA3),  
 function(obj) {  
 LL <- sweep(as.matrix(obj,"LL"), 2, methowSHm$n, "/") # LL of unique CH  
 LOO <- loo(LL)  
 LOO$pointwise <- LOO$pointwise[rep(1:ncol(LL), times = methowSHm$n),]  
 LOO$pareto\_k <- LOO$pareto\_k[rep(1:ncol(LL), times = methowSHm$n)]  
 LOO[c("elpd\_loo","p\_loo","looic")] <- colSums(LOO$pointwise)  
 LOO[c("se\_elpd\_loo","se\_p\_loo","se\_looic")] <-   
 apply(LOO$pointwise,2,sd)\*sqrt(nrow(LOO$pointwise)) # SE of \*sum\*  
 return(LOO)  
 })  
  
compare\_cjs <- compare(x = loo\_cjs)  
# deltas  
compare\_cjs[,"looic"] <- compare\_cjs[,"looic"] - min(compare\_cjs[,"looic"])  
round(compare\_cjs[,c("looic","se\_looic","p\_loo","se\_p\_loo")], 2)

looic se\_looic p\_loo se\_p\_loo  
cjs\_all 0.00 91.73 3.38 0.31  
cjs\_fixNA0 3.88 94.27 18.86 2.18  
cjs\_fixNA2a 6.63 94.17 20.56 2.10  
cjs\_fixNA2b 7.82 94.32 20.09 2.17  
cjs\_fixNA1 28.63 95.40 17.86 2.08  
cjs\_fixNA3 33.46 95.48 18.89 2.05

We get a warning about the Pareto parameter estimates from the smoothed importance resampling; this can indicate that the likelihood is unduly sensitive to certain observations, violating the exchangeability assumption underlying LOO, but inspection of the individual loo objects reassures us that all values are in the acceptable range. The model ranking appears highly uncertain; the SEs dwarf all but the largest discrepancies. But as we saw with the ocean age GLMMs, pairwise comparisons can be much more discriminating.

compair\_cjs <- vector("list", length(loo\_cjs) - 1)   
for(i in 1:length(compair\_cjs))  
{  
 names(compair\_cjs)[i] <- paste(row.names(compare\_cjs)[c(i+1,i)], collapse = " vs. ")  
 compair\_cjs[[i]] <- 2\*compare(x = loo\_cjs[row.names(compare\_cjs)[c(i+1,i)]])  
 names(compair\_cjs[[i]])[1] <- "looic\_diff"  
}  
compair\_cjs

$`cjs\_fixNA0 vs. cjs\_all`  
looic\_diff se   
 -6484.0 158.4   
  
$`cjs\_fixNA2a vs. cjs\_fixNA0`  
looic\_diff se   
 -10.7 8.5   
  
$`cjs\_fixNA2b vs. cjs\_fixNA2a`  
looic\_diff se   
 2.0 3.7   
  
$`cjs\_fixNA1 vs. cjs\_fixNA2b`  
looic\_diff se   
 -307.7 41.0   
  
$`cjs\_fixNA3 vs. cjs\_fixNA1`  
looic\_diff se   
 -10.3 7.8

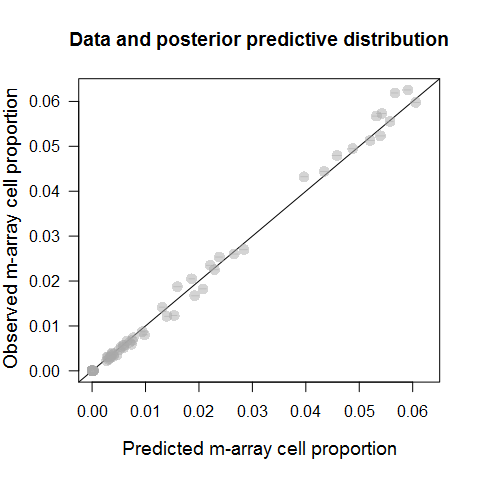
The SEs of the paired comparisons indicate that the full model cjs\_fixNA3 is preferred over the version with only a smolt age effect cjs\_fixNA1, and the latter is overwhelmingly superior to the next-best model cjs\_fixNA2b (linear effect of adult age only), which suggests smolt age is more important for predicting juvenile survival and SAR than adult age is for upstream survival. As we surmised earlier, there is no evidence for nonlinearity in the adult age effect (cjs\_fixNA2a vs. cjs\_fixNA2b). All the time-varying hierarchical models perform vastly better than the one-size-fits-all, complete-pooling cjs\_all model.

## Plots

As with the GLMMs, we can better understand the CJS models by comparing their posterior predictive distribution to the data we actually observed. We choose to focus on the relative frequencies in the m-array formed by cross-tabulating all possible combinations of release year, smolt age, adult age, and capture history. To facilitate this, we’ll create a function to simulate the adult age and capture history for each steelhead smolt in the study, given the fitted GLMM and CJS models. (Another chunk of this function calculates expected cumulative life-cycle survival; we’ll return to this later.)

sim\_phi\_tot <- function(cjs, fit\_oa, iter = NULL, adult\_age, release\_years, return\_years,   
 Tmax = NULL, CH = NULL)   
{  
 # Actual unique years, not factor levels  
 rlsy <- sort(unique(release\_years))  
 rtny <- sort(unique(return\_years))  
 # adult\_age is the centered covariate used in CJS models  
 mean\_adult\_age <- attributes(adult\_age)[["scaled:center"]]  
 aa <- 2:4 - mean\_adult\_age  
 names(aa) <- 2:4 # ignore 5-year-olds  
   
 # CJS  
 TT <- 7 # nr. occasions  
 if(is.null(Tmax)) Tmax <- TT - 1 # calculate survival for occasions 1:Tmax  
 if(is.null(iter)) iter <- length(as.matrix(cjs,"lp\_\_")) # nr. MCMC iters   
 J <- max(length(rlsy), length(rtny)) # max nr. groups   
 K <- 3 # nr. covariates  
 mbeta <- as.matrix(cjs, "beta")[1:iter,]  
 beta <- array(NA, c(iter, TT - 1, K))  
 for(k in 1:K)  
 beta[,,k] <- mbeta[, paste0("beta[", k, ",", 1:(TT-1), "]")]  
 mepsilon\_z <- as.matrix(cjs, "epsilon\_z")[1:iter,]  
 epsilon\_z <- array(NA, c(iter, TT - 1, J))  
 for(j in 1:J)  
 epsilon\_z[,,j] <- mepsilon\_z[, paste0("epsilon\_z[", j, ",", 1:(TT-1), "]")]  
 sigma <- as.matrix(cjs, "sigma")[1:iter,] # iter \* TT - 1  
 epsilon <- sweep(epsilon\_z, 1:2, sigma, "\*") # iter \* TT - 1 \* J  
 b <- as.matrix(cjs, "b")[1:iter, paste0("b[", 1, ",", 1:TT, "]")] # iter \* TT  
 me\_z <- as.matrix(cjs, "e\_z")[1:iter,]  
 e\_z <- array(NA, c(iter, TT, J))  
 for(j in 1:J)  
 e\_z[,,j] <- me\_z[, paste0("e\_z[", j, ",", 1:TT, "]")]  
 s <- as.matrix(cjs, "s")[1:iter,] # iter \* TT  
 e <- sweep(e\_z, 1:2, s, "\*")  
  
 # OA GLMM  
 dat <- data.frame(release\_year = rep(rlsy, each = 2),   
 smolt\_age = c("S1","S2"), smolt\_age\_num = 1:2)  
 prob\_oa2 <- posterior\_linpred(fit\_oa, transform = TRUE, newdata = dat)  
 prob\_oa2 <- prob\_oa2[1:iter,] # GLMM iter > CJS iter  
   
 # Calculate cumulative survival (Gilbert & Rich age notation)  
 phi\_1.1 <- array(NA, c(iter, TT - 1, length(rlsy)))  
 dimnames(phi\_1.1)[[3]] <- rlsy  
 phi\_1.2 <- phi\_1.1  
 phi\_2.1 <- phi\_1.1  
 phi\_2.2 <- phi\_1.1  
 phi\_tot\_S1 <- matrix(NA, iter, length(rlsy))  
 colnames(phi\_tot\_S1) <- rlsy  
 phi\_tot\_S2 <- phi\_tot\_S1  
 p\_x.1 <- array(NA, c(iter, TT, length(rlsy)))  
 dimnames(p\_x.1)[[3]] <- rlsy  
 p\_x.2 <- p\_x.1  
 for(j in 1:length(rlsy))  
 {  
 phi\_1.1[,,j] <- plogis(beta[,,1] + aa["2"] \* beta[,,3] +   
 cbind(epsilon[,1:3,j], epsilon[,4:6,j]))  
 phi\_1.2[,,j] <- plogis(beta[,,1] + aa["3"] \* beta[,,3] +   
 cbind(epsilon[,1:3,j], epsilon[,4:6,j+1]))  
 prob\_1.2 <- prob\_oa2[, dat$release\_year==rlsy[j] & dat$smolt\_age=="S1"]  
 phi\_tot\_S1[,j] <- (1 - prob\_1.2) \* apply(phi\_1.1[,1:Tmax,j], 1, prod) +   
 prob\_1.2 \* apply(phi\_1.2[,1:Tmax,j], 1, prod)  
 phi\_2.1[,,j] <- plogis(beta[,,1] + beta[,,2] + aa["3"] \* beta[,,3] +   
 cbind(epsilon[,1:3,j], epsilon[,4:6,j]))  
 phi\_2.2[,,j] <- plogis(beta[,,1] + beta[,,2] + aa["4"] \* beta[,,3] +   
 cbind(epsilon[,1:3,j], epsilon[,4:6,j+1]))  
 prob\_2.2 <- prob\_oa2[, dat$release\_year==rlsy[j] & dat$smolt\_age=="S2"]  
 phi\_tot\_S2[,j] <- (1 - prob\_2.2) \* apply(phi\_2.1[,1:Tmax,j], 1, prod) +   
 prob\_2.2 \* apply(phi\_2.2[,1:Tmax,j], 1, prod)  
 p\_x.1[,,j] <- plogis(b + cbind(e[,1:3,j], e[,4:7,j]))  
 p\_x.2[,,j] <- plogis(b + cbind(e[,1:3,j], e[,4:7,j+1]))  
 }  
   
 # Generate replicate data from PPD using initial conditions in CH  
 # CH is data frame with ID, release\_year, smolt\_age, adult\_age, return\_year   
 # (latter two can be NA), and capture histories for occasions 1:TT (rows are individuals)  
 if(!is.null(CH))   
 {  
 N <- nrow(CH)  
 CH\_sim <- CH  
 CHm\_sim <- matrix(NA, 2^(TT - 1) \* length(rlsy) \* 2 \* 4, iter) # simulated m-arrays  
 s\_a <- CH\_sim$smolt\_age  
 r\_y <- as.character(CH\_sim$release\_year)  
 for(i in 1:iter)  
 {  
 o\_a <- 1 + rbinom(N, size = 1,   
 prob = prob\_oa2[i, match(paste(r\_y, s\_a),   
 paste(dat$release\_year, dat$smolt\_age))])  
 CH\_sim$adult\_age <- ifelse(s\_a=="S1", 1, 2) + o\_a  
 CH\_sim$return\_year <- CH\_sim$release\_year + o\_a  
 phi <- matrix(NA, N, TT - 1)  
 phi[s\_a=="S1" & o\_a==1,] <- t(phi\_1.1[i,,r\_y[s\_a=="S1" & o\_a==1]])  
 phi[s\_a=="S1" & o\_a==2,] <- t(phi\_1.2[i,,r\_y[s\_a=="S1" & o\_a==2]])  
 phi[s\_a=="S2" & o\_a==1,] <- t(phi\_2.1[i,,r\_y[s\_a=="S2" & o\_a==1]])  
 phi[s\_a=="S2" & o\_a==2,] <- t(phi\_2.2[i,,r\_y[s\_a=="S2" & o\_a==2]])  
 p <- matrix(NA, N, TT)  
 p[o\_a==1,] <- t(p\_x.1[i,,r\_y[o\_a==1]])  
 p[o\_a==2,] <- t(p\_x.2[i,,r\_y[o\_a==2]])  
 z <- cbind(1, matrix(NA, N, TT - 1)) # latent states: alive or dead  
 for(tt in 2:TT)  
 z[,tt] <- z[,tt-1] \* rbinom(N, size = 1, prob = phi[,tt - 1])  
 # observations  
 y <- cbind(1, matrix(rbinom(N \* (TT - 1), size = 1,   
 prob = z[,-1] \* p[,-1]), N, TT - 1))  
 colnames(y) <- names(CH\_sim)[!names(CH\_sim) %in% c("tag","release\_year","smolt\_age",  
 "adult\_age","return\_year")]  
 CH\_sim$adult\_age <- factor(replace(CH\_sim$adult\_age, CH\_sim$TDAWEA==0, NA), exclude = "")  
 CHm\_sim\_i <- aggregate(WNFH ~ .,  
 data = cbind(CH\_sim[,c("release\_year","smolt\_age","adult\_age")], y),   
 length, drop = FALSE)  
 CHm\_sim[,i] <- CHm\_sim\_i$WNFH  
 }  
 CHm\_sim <- cbind(CHm\_sim\_i[,colnames(CHm\_sim\_i) != "WNFH"], CHm\_sim)  
 CHm\_sim$adult\_age <- as.numeric(as.character(CHm\_sim$adult\_age))  
 } else {  
 CHm\_sim <- NULL  
 }  
 return(list(S1 = phi\_tot\_S1, S2 = phi\_tot\_S2, CHm\_sim = CHm\_sim))  
}

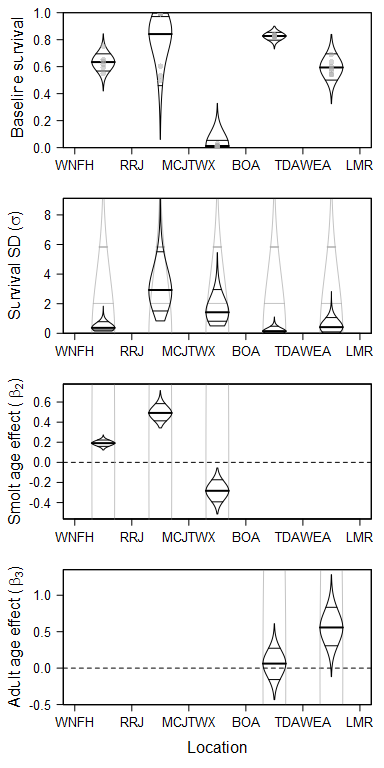
cjs\_ppd\_iter <- 100  
cjs\_ppd <- sim\_phi\_tot(cjs = cjs\_fixNA3, fit\_oa = fit\_oa1b, iter = cjs\_ppd\_iter,   
 adult\_age = adult\_age,release\_years = unique(methowSHm$release\_year),  
 return\_years = unique(na.omit(methowSHm$return\_year)),  
 CH = methowSH[,c("tag","release\_year",  
 "smolt\_age","return\_year","adult\_age",  
 "WNFH","RRJ","MCJTWX","BOA","TDAWEA","LMR","MRCBRD")])



**Figure S4:** Posterior predictive check for full hierarchical CJS model.

Figure S4 shows the observed and predicted proportion of smolts that had a given combination of traits (release year and age) and fate (capture history and, for those detected as adults, adult age). There is both good and bad news here for our overall integrated model. On one hand, the observed proportions and the PPD medians are tightly correlated; the model is accurate and unbiased. On the other hand, almost none of the 90% PPD credible intervals (the horizontal lines, so narrow as to be almost invisible) contain the corresponding observations, suggesting the predictions are not well-calibrated; the data show overdispersion relative to the multinomial model. Because of the large sample size of tagged smolts (*n* = 176930), the underlying survival and capture probabilities are estimated quite precisely *and* the simulated sample proportions are a very precise estimate of the underlying multinomial cell probabilities. Both of these facts make this a sensitive posterior predictive check, capable of discerning even small departures of the data from the model. The model does a good job of capturing the major patterns in the data, but the discrepancies raise interesting questions: what accounts for the extra dispersion, and might some of it be explained by individual-level variation in traits such as body size at release?

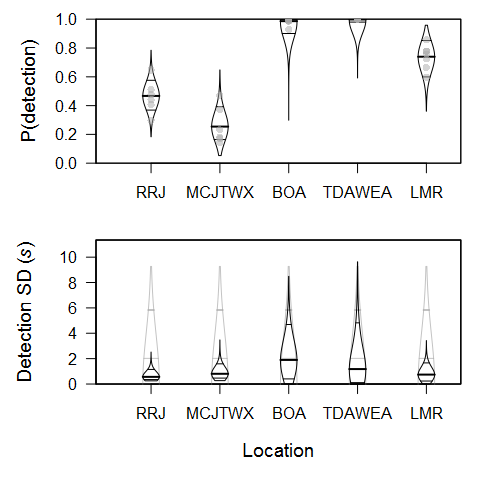
Next we examine the marginal posterior distributions of stage-specific survival parameters in the full model.



**Figure S5:** Estimated hyper-parameters of survival (intercept, regression slopes, and interannual SD) under the full hierarchical CJS model.

In the top panel, corresponding to the intercept(s), the gray points are point estimates (posterior medians) of the year-specific baseline survivals (release year for the first three transitions, return year for the latter two). The *y*-axis in that panel is on the probability scale; the other panels are on the logit scale. The gray violin plots in the background represent the priors, but the prior on the intercept is and is not shown. This lets us see that, for example, two of the interannual SDs are poorly identified by the likelihood and thus are constrained by the prior. (It might seem odd that the SD for SAR, the third stage, has such a long tail when the actual estimates are all so close to zero. But remember the hierarchical model is on the logit scale, which stretches the unit interval by an increasing amount as you approach its endpoints.) The plots of the covariate effects neatly illustrate what we saw in the stanfit summaries: S2 smolts survived better through both downstream migration segments but had lower SAR than S1 smolts, and older adults survived better from Wells Dam to the Methow but not prior to that.

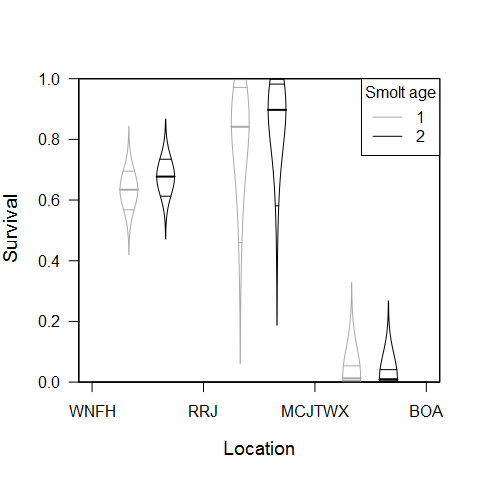
We can make similar plots for the detection parameters in the full model. In this case there are no covariate effects to worry about.



**Figure S6:** Estimated hyper-parameters of detection probability (intercept and interannual SD) under the full hierarchical CJS model.

Again we see the near-perfect detection efficiency at mainstem Columbia adult fish ladders, justifying our approach of fixing survival and detection for nondetected adults. The interannual SDs of these two detection rates are long-tailed and weakly identified by the data; again this can be understood by considering how the logit transformation works. The logit-scale estimates in most years would wander off toward their modes at if not shrunk by their hyper-SDs, which in turn are partly (though not completely) constrained by their priors. The effect of this shrinkage is negligible on the probability scale, which is how enters into the likelihood, so we need not worry about sensitivity to the prior in cases like this.

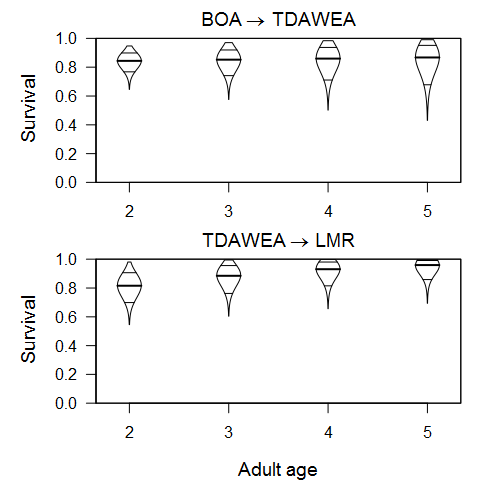
To gain more insight into the reach-specific smolt age effects, we can plot the expected survival for S1 and S2 steelhead in an average year ().



**Figure S7:** Predicted reach-specific survival of S1 and S2 smolts under the full hierarchical CJS model.

The strong support for this covariate notwithstanding, its effect is not terribly impressive on the probability scale compared to the posterior uncertainty (which comes from the intercept or baseline as well as the smolt age coefficient), but nonetheless is clearly in the direction indicated by the coefficient estimates shown in Figure S5.

Likewise, we can plot the predicted effect of adult age on upstream survival in an average year.

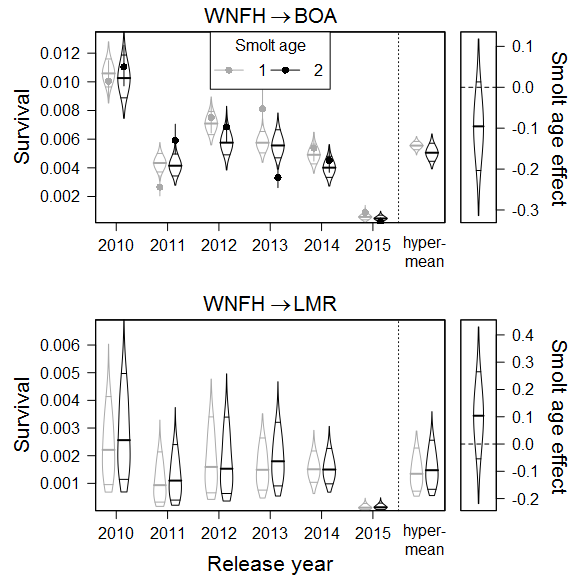


**Figure S8:** Predicted effect of adult age on reach-specific survival during upstream migration under the full hierarchical CJS model.

At any age, adults experience higher mortality in the final mainstem Columbia reach than in the entire upstream migration corridor before that, a much longer distance. However, as we saw in Figure S5, older adults have higher survival in that final reach. This should give S2 smolts an advantage during upstream migration, considering that they tend to be older at maturity (Figure S3).

With that in mind, we can calculate the overall expected survival to adulthood for S1 and S2 smolts accounting for the effects of smolt age on downstream survival, SAR, and age at maturity, and the effect of adult age on upstream survival. These probabilities depend on the parameters of the GLMM and CJS models:

We’ll use the same function we created for the posterior predictive simulations to extract the respective parameters from the fitted models, assemble them in the appropriate sequence, and back-transform them to survival.



**Figure S9:** Predicted survival from release to adult return under the full integrated model (GLMM and CJS)

The top panel shows cumulative survival from release to BOA (the first detection location encountered by returning adults) and the bottom panel shows survival from release to the Methow subbasin. One reason for plotting these separately is that the former can be directly compared to data, thanks to the near-perfect detection efficiency at adult fish ladders in the mainstem Columbia dams. The points show the sample proportion of fish from each release group that were detected as adults, along with frequentist binomial 90% confidence intervals. We can see that the joint survival-age at maturity model does a reasonably good job of predicting the annual cumulative survival rates of S1 and S2 smolts; the posterior 90% credible intervals and the binomial confidence intervals overlap in nearly all cases. However, the model does seem to miss the mark in 2011 and 2013. The smolt age effect in those years was above-average in magnitude, but in opposite directions. The model shrinks these “outliers” toward the mean, represented by the average survivals shown on the right side of the plot and by the average logit-scale difference between S2 and S1 (the smolt age effect, analogous to in the full CJS model) shown in the top right panel.

The other reason for plotting the two cumulative survivals separately is that the comparison reveals how the effect of smolt age on survival reverses during upstream migration, mediated by the effect on adult age. S2 smolts fare worse than S1s up through their return to the Columbia (because even though they have an advantage during downstream migration, they have substantially lower SAR), but by the time they make it home to the Methow, S2s have a slight advantage (because older adults survive better, especially in the final mainstem reach). The evidence for these differences is strong but not overwhelming – there is a 92% posterior probability that the smolt age effect is negative for WNFH BON, and 86% that it is positive for WNFH LMR. Nevertheless, these results indicate complex interactions between life history and demography, and carryover effects of hatchery rearing practices that persist to adulthood.