Bootstrapped quantiles

The use of bootstrapping to estimate the required number of iterations for a given quantile precision when conducting MSE

12 July, 2019

## Setup

# packages  
library(FLCore)  
library(ggplotFL)  
library(reshape2)  
# functions  
source("../R/qboot.R")  
source("../R/qbootplot.R")  
nboot <- 300

## Introduction

The proposed bootstrapping methodology calculates the relative standard error and confidence intervals for quantile estimates of a stochastic simulation. The approach is of interest for application to management strategy evaluations (MSE) in determining the number of samples required for a given level of precision. In particular, the approach can be used to optimize the search of a given harvest control parameter space given a cost function (e.g. maximum yield, minimum interannual yield variation), while avoiding poorly scoring regions after a lower number of iterations (e.g. risk of spawning stock biomass going below a given threshold).

The bootstrapping approach allows for two main outputs:

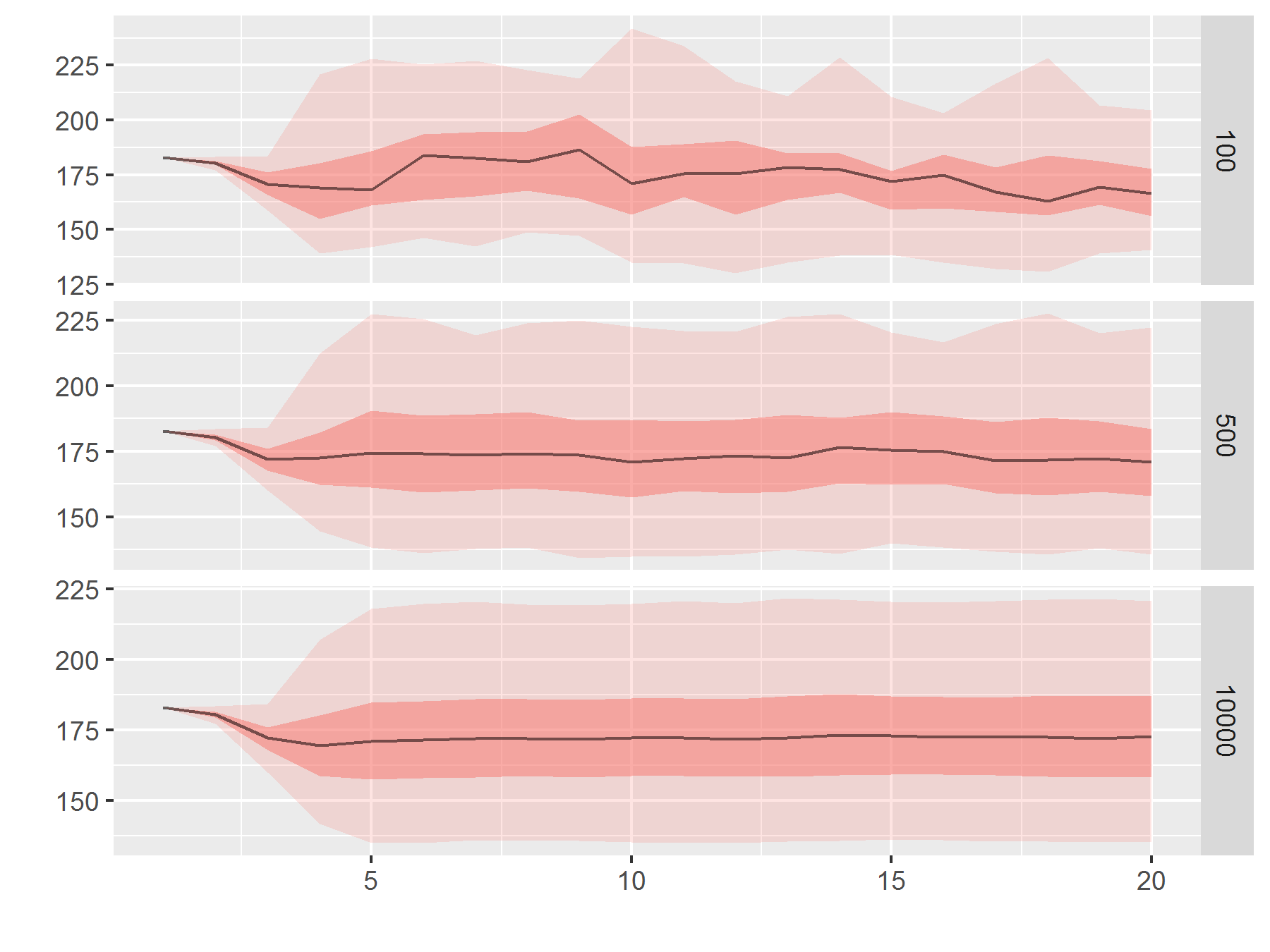
1. Quantile estimates by year and number of iterations, associated standard error and confidence limits
2. Prediction of the number of iterations required for a given level of relative standard error (standard error / mean \* 100).

The first output provides a robust estimation quantile estimates by number of iterations through boostrapping with replacement and provides two measures of the confidence of the estimate. The first is standard error (), assuming a roughly normal distribution of the sample estimates as dictated by the central limit theorem. Thus, confidence intervals could be calculated with and cumulative distribution function. A second approach to determining confidence intervals is a direct estimation from the estimation distribution. This is also provided in the form of confidence limits.

The latter output is based on the known relationship between *se* and sample size (), , where . The qboot function uses relative standard error () in fitting this relationship in order to standardize among years of the data set since, in the case of MSE, short-term values may be of a different magnitude than long-term equilibrium values. Thus, using the linear relationship log(niter)~log(rsa), where *niter* are the number of samples used in the bootstrap estimation, the model is able to approximate the number of iterations required in order to achieve a given level of *rsa*.

## A synthetic example

load(file="../data/ssb.RData")  
L <- FLQuants("100" = ssb[,,,,,1:50], "500" = ssb[,,,,,1:500], "10000" = ssb)  
plot(L)

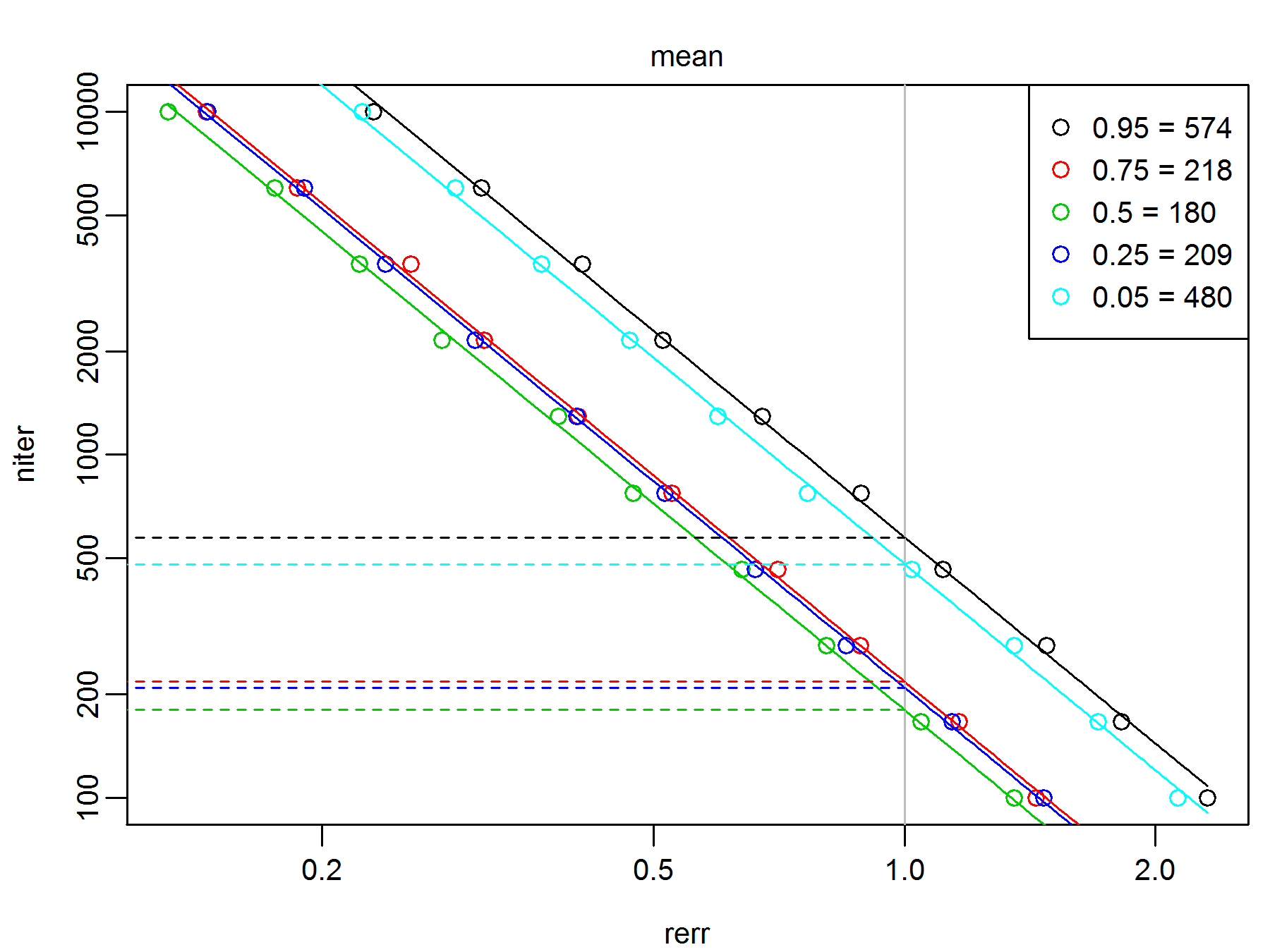


**Figure 1.** Quantile distributions of spawning stock biomass for a synthetic stock by differing number of iterations. Darker shaded areas shows the 25% and 75% quantiles, with median indicated by a solid line, and 5% and 95% quantiles as dashed lines.

## Predicting number of iterations for a given error level

The following shows mean relative standard error across years by defined quantile and number of iterations. This is related to ‘Prob1’ risk estimation.

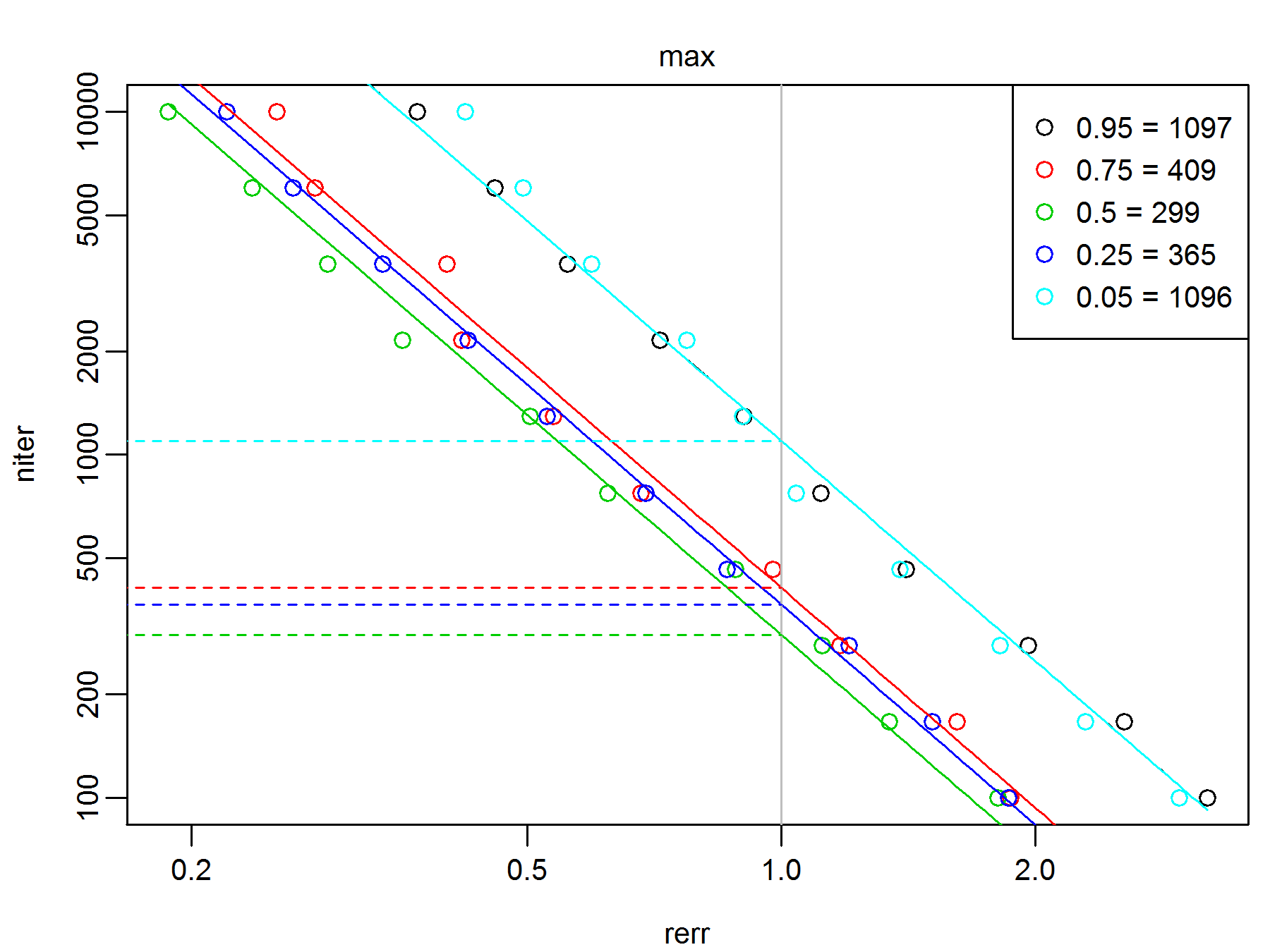
# Convert to matrix (year by iteration)  
X <- array(ssb, dim = dim(ssb)[c(2,6)])  
dimnames(X) <- dimnames(ssb)[c(2,6)] # add dim names  
X <- t(X) # transpose to make iters rows  
  
set.seed(1)  
# run bootstrapping, model mean  
res <- qboot(X, nboot = nboot, aggfun = "mean", rerrTarg = 1, verbose = FALSE)  
par(mar=c(3,3,2,0.5), ps=10, mgp = c(2,0.5,0))  
qbootplot(res)  
mtext("mean", side = 3, line = 0.25)



**Figure 2.** Log-log plot of mean bootstrapped relative standard error (rse) versus number of iterations by quantile. Predicted linear regressions are shown by solid lines, and predicted number of iterations needed for the target error level are shown by dashed lines, with values in the legend.

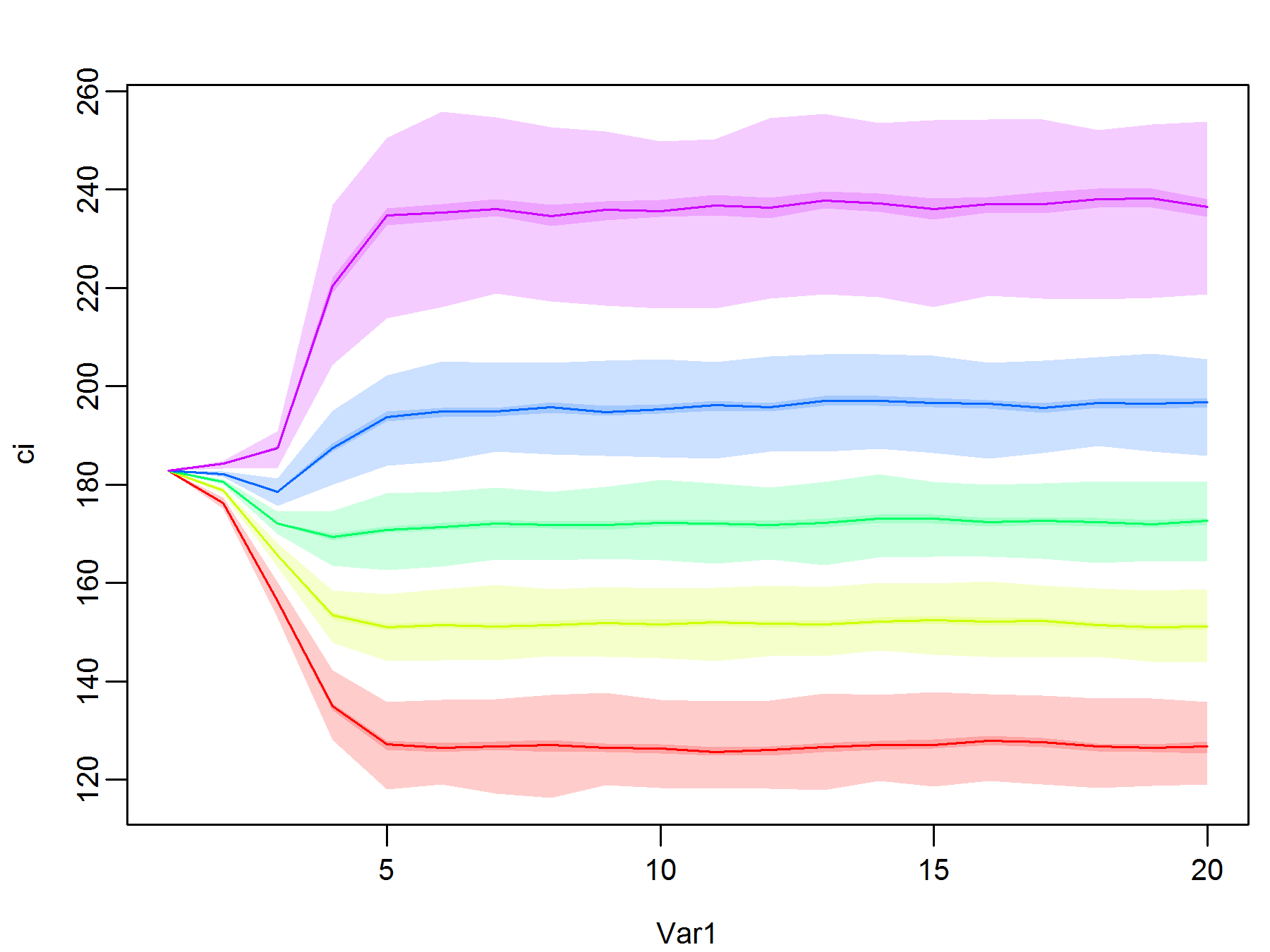
The following shows max rse across years by defined quantile and number of iterations. This is related to ‘Prob3’ risk estimation.

set.seed(1)  
# run bootstrapping, model max  
res <- qboot(X, nboot = nboot, aggfun = "max", rerrTarg = 1, verbose = FALSE)  
par(mar=c(3,3,2,0.5), ps=10, mgp = c(2,0.5,0))  
qbootplot(res)  
mtext("max", side = 3, line = 0.25)



**Figure 3.** Log-log plot of max bootstrapped relative error (rse) versus number of iterations by quantile. Predicted linear regressions are shown by solid lines, and predicted number of iterations needed for the target error level are shown by dashed lines, with values in the legend.

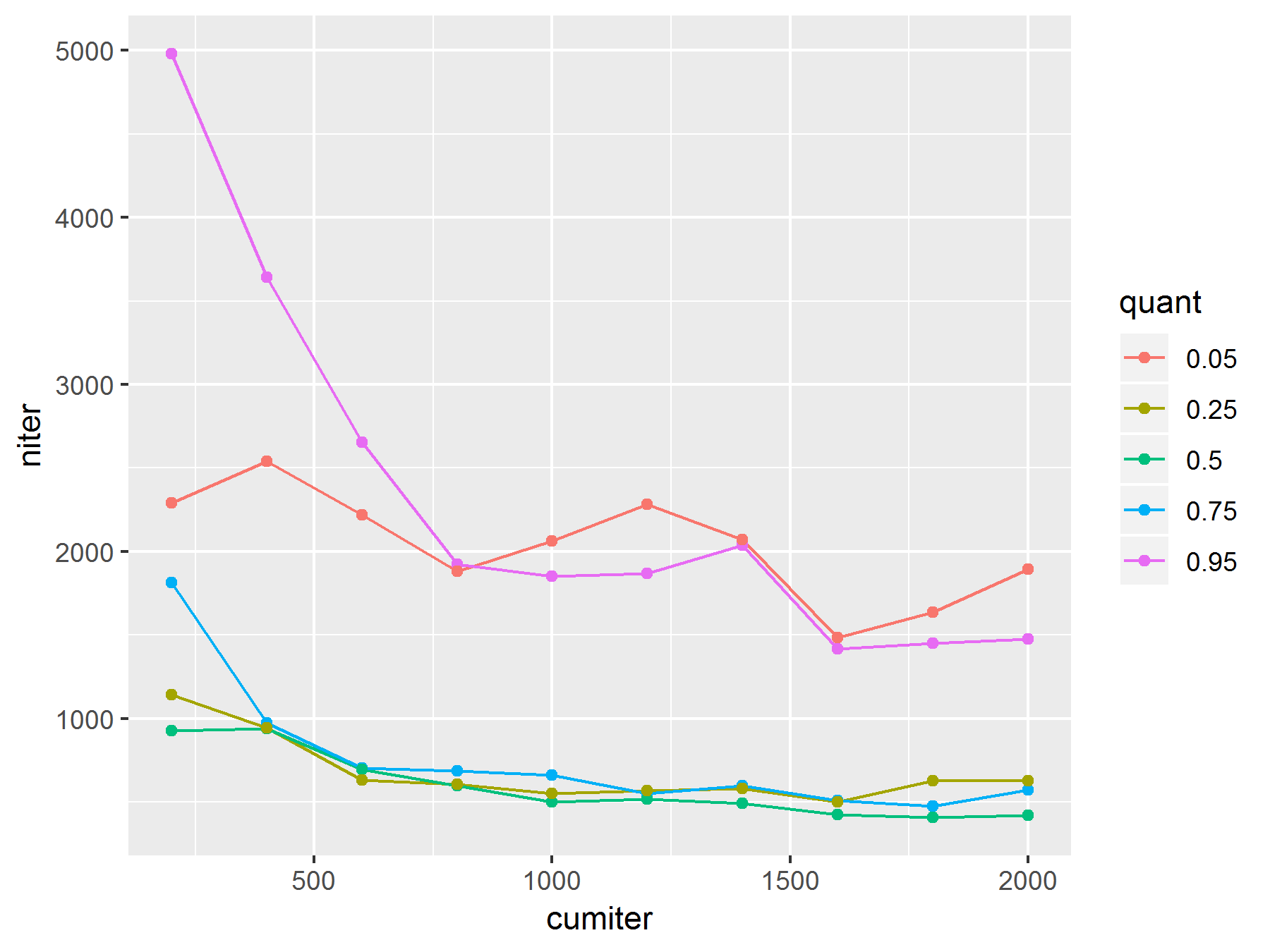
par(mar=c(3,3,2,0.5), ps=10, mgp = c(2,0.5,0))  
# make dataframe of subset of ci data  
res2 <- expand.grid(dimnames(res$est))  
res2$niter <- factor(round(as.numeric(as.character(res2$niter))))  
res2$Var1 <- as.numeric(as.character(res2$Var1))  
res2$est <- c(res$est)  
res2$cilow <- c(res$cilow)  
res2$ciup <- c(res$ciup)  
# res2$cilow <- c(res$cilow)  
# res2$ciup <- c(res$ciup)  
res2 <- subset(res2, niter %in% levels(res2$niter)[c(1,10,20)])  
  
res3 <- as.data.frame(rbind(  
 as.matrix(res2[,c("Var1", "quant", "niter", "cilow")]),  
 as.matrix(res2[rev(seq(nrow(res2))),c("Var1", "quant", "niter", "ciup")])))  
names(res3) <- c("Var1", "quant", "niter", "ci")  
res3$Var1 <- as.numeric(as.character(res3$Var1))  
res3$ci <- as.numeric(as.character(res3$ci))  
res3$niter <- factor(res3$niter,   
 levels = ac(sort(as.numeric(ac(levels(res3$niter))))) )  
  
Qs <- apply(X, 2, quantile, prob = as.numeric(levels(res3$quant)))  
plot(ci ~ Var1, data = res3, ylim = range(ci), t="n")  
COL = rainbow(5)  
for(i in seq(levels(res3$quant))){  
 for(j in seq(levels(res3$niter))){  
 dat.ij = subset(res3, niter == levels(niter)[j] &   
 quant == levels(quant)[i])  
 polygon(x = dat.ij$Var1, y = dat.ij$ci,   
 col = adjustcolor(COL[i], 0.2), border = NA)  
 }  
 lines(as.numeric(colnames(Qs)), Qs[i,], col=COL[i])  
}



**Figure 4.** True quantiles (lines; 5, 25, 50, 75, & 95%) and their confidence intervals (95%, shaded area) for varying sample sizes (n = 100, 10000).

Still need to understand this…

cumiter <- seq(200, 2000, 200) # c(250, 500, 750, 1000, 2000, 5000)  
res <- vector("list", length(cumiter))  
set.seed(1)  
for(i in seq(res)){  
 set.seed(1)  
 res.i <- qboot(X[seq(cumiter[i]),], nboot = nboot, aggfun = "max",   
 rerrTarg = 1, verbose = FALSE)  
 res.i$nsim$cumiter <- cumiter[i]  
 res[[i]] <- res.i$nsim  
}  
res <- do.call("rbind", res)  
  
ggplot(res, aes(x = cumiter, y = niter, group = quant, colour = quant)) +   
 geom\_line() + geom\_point()



**Figure 5.** Influence of cumulative sample size on predicted number of iterations needed for a given error level (1%).

## Software Versions

* R version 3.6.0 (2019-04-26)
* FLCore: 2.6.13
* ggplotFL: 2.6.6
* reshape2: 1.4.3
* **Compiled**: 2019-Jul-12