# Comment:

#

# A sophisticated example of adding further output to a basic plot.

#

# Most of the functions defined are just for calculating values

# relevant to the data analysis.

#

# The function plotPars() is the one of interest for seeing how

# the drawing of the plot is done.

#

params <- function(N, breaks, p=seq(0.001, 1, length=100)) {

list(N=N, T=1/breaks, p=p, q=1-p)

}

pdfcomp <- function(comp, params) {

n <- params$T

p <- params$p

q <- params$q

y <- round(comp/n)

choose(n, comp)\*p^comp\*q^(n-comp) / (1 - q^n)

}

# Expected num sherds (for a vessel) [=completeness]

expcomp <- function(params) {

params$T\*params$p/(1-params$q^params$T)

}

# Variance of num sherds (for a vessel)

varcomp <- function(params) {

n <- params$T

p <- params$p

q <- params$q

# From Johnson & Kotz

(n\*p\*q / (1 - q^n)) - (n^2\*p^2\*q^n / (1 - q^n)^2)

# n^2 times Thomas Yee's formula

# n^2\*((p\*(1 + p\*(n - 1)) / (n\*(1 - q^n))) - (p^2 / (1 - q^n)^2))

}

# Expected value of completeness (for a sample of vessels)

expmeancomp <- function(params) {

expcomp(params)

}

# Variance of completeness (for a sample of vessels)

# Use the expected number of vessels in sample as denominator

varmeancomp <- function(params) {

varcomp(params)/(numvess(params))

}

numvess <- function(params) {

params$N\*(1-params$q^params$T)

}

ecomp <- function(p, T, comp) {

q <- 1 - p

T\*p/(1 - q^T) - comp

}

estN <- function(comp, broke, n) {

T <- 1/broke

n / (1 - (1 - uniroot(ecomp, c(0.00001, 1), T=T, comp=comp)$root)^T)

}

nvessscale <- function(params, xlim, ylim, new=TRUE) {

if (new)

par(new=TRUE)

plot(0:1, c(1, params$N), type="n", axes=!new, ann=FALSE,

xlim=xlim, ylim=ylim)

}

compscale <- function(params, xlim, ylim, new=TRUE) {

if (new)

par(new=TRUE)

plot(0:1, c(1, params$T), type="n", axes=!new, ann=FALSE,

xlim=xlim, ylim=ylim)

}

lowerCI <- function(p, N, breaks, lb) {

params <- params(N, breaks, p)

expmeancomp(params) - 2\*sqrt(varmeancomp(params)) - lb

}

upperCI <- function(p, N, breaks, lb) {

params <- params(N, breaks, p)

expmeancomp(params) + 2\*sqrt(varmeancomp(params)) - lb

}

critP <- function(comp, params) {

c(uniroot(lowerCI, c(0.00001, 1), N=params$N,

breaks=1/params$T, lb=max(comp))$root,

if (upperCI(0.00001, params$N, 1/params$T, min(comp)) > 0) 0

else uniroot(upperCI, c(0.00001, 1), N=params$N,

breaks=1/params$T, lb=min(comp))$root)

}

anncomp <- function(params, comp, xlim, ylim, cylim) {

cp <- critP(comp, params)

nv <- numvess(params(params$N, 1/params$T, cp))

nvessscale(params, xlim, ylim)

polygon(c(cp[2], cp[2], 0, 0, cp[1], cp[1]),

c(0, nv[2], nv[2], nv[1], nv[1], 0),

col="grey90", border=NA)

text(0, nv[1], paste(round(nv[1]),

" (", round(100\*nv[1]/params$N), "%)", sep=""),

adj=c(0, 0), col="grey")

text(0, nv[2], paste(round(nv[2]),

" (", round(100\*nv[2]/params$N), "%)", sep=""),

adj=c(0, 1), col="grey")

compscale(params, xlim, cylim)

segments(1, min(comp), cp[2], comp, col="grey")

segments(1, max(comp), cp[1], comp, col="grey")

text(1, comp, paste(comp, collapse="-"), adj=c(1, 0), col="grey")

}

plotPars <- function(params, comp, xlim=NULL, ylim=NULL) {

mean <- expmeancomp(params)

var <- 2\*sqrt(varmeancomp(params))

lb <- mean - var

ub <- mean + var

par(mar=c(5, 4, 4, 4))

if (is.null(ylim))

cylim <- ylim

else

cylim <- c(1 + ((ylim[1] - 1)/(params$N - 1))\*(params$T - 1),

1 + ((ylim[2] - 1)/(params$N - 1))\*(params$T - 1))

nvessscale(params, xlim, ylim, new=FALSE)

compscale(params, xlim, cylim)

polygon(c(params$p, rev(params$p)), c(lb, rev(ub)),

col="grey90", border=NA)

anncomp(params, comp, xlim, ylim, cylim)

nvessscale(params, xlim, ylim)

mtext("Number of Vessels", side=2, line=3)

mtext("Sampling Fraction", side=1, line=3)

lines(params$p, numvess(params))

par(new=TRUE)

compscale(params, xlim, cylim)

mtext("Completeness", side=4, line=3)

axis(4)

lines(params$p, mean, lty="dashed")

lines(params$p, lb, lty="dotted")

lines(params$p, ub, lty="dotted")

mtext(paste("N = ", round(params$N),

" brokenness = ", round(1/params$T, 3), sep=""),

side=3, line=2)

}

par(cex=0.8, mar=c(3, 3, 3, 3))

p6 <- params(estN(1.2, 0.5, 200), 0.5)

plotPars(p6, 1.2)

nvessscale(p6, NULL, NULL)

pcrit <- 1 - (1 - 200/estN(1.2, 0.5, 200))^(1/p6$T)

lines(c(0, pcrit), c(200, 200))

lines(c(pcrit, pcrit), c(200, 0))