R Output from Estimation Procedure

Lines in RED represents text copied and pasted into R from in98estimation.txt (via a word/text processor).

Lines in BLUE represent output from the R statistical software

Key things to remember when starting this procedure.

1. It is necessary to execute the weight adjustment procedure prior to this procedure. If already accomplished, skip to step 3.
2. Make sure that R has the correct directory information for your analysis folder. Usually accomplished by placing a "short-cut" of R in the analysis folder. That folder should also include a sub-folder "original_data", containing files inbr98.txt and siteevaluation98.txt
3. Load the package "psurvey.analysis" into R
4. Output files will appear in the analysis folder

```
> # File: in98estimation.txt
> # Purpose: Estimation for Indiana Upper Wabash River Basin Biological Status
>
> # If R workspace from weight adjustment is not currently present or was not
saved (if saved then have R load it), then read in designstatus file
> # Note that designstatus file contains the data results as well as design
status information
>
> designstatus <- read.table('designstatus.csv',sep=',',header=TRUE)
> names(designstatus)
 [1] "SiteID"              "SiteName"            "County"
 [4] "RF3RCHID"            "MAP24K"              "MAP100K"
 [7] "MAP250K"             "Long.dd"             "Lat.dd"
[10] "Stratum"             "MDCaty"              "Weight"
[13] "StrahlerOrder"       "OECO.96"             "SiteIDF"
[16] "Basin"               "HUC14Code"           "StrahlerOrder1"
[19] "StatusB"             "Comments"            "BiologicalSampleDate"
[22] "QHEI.Score"          "QHEI.Status"         "IBI.Score"
[25] "IBI.Status"          "xmarinus"            "ymarinus"
[28] "final.wt"
>
> # Set up data frames for estimation functions
> # Set up data frames for sites, subpop, dsgn, and data
> # sites identifies which sites to use in estimation
> # Determine what StatusB and StatusWC codes are used
> table(designstatus$StatusB)

  LD   NT   OT   PB SCNB   TS   UK
  19    9    2    7   14   48    1

> # Note that all sites were evaluated
> # All sites will be used to estimate stream length in each category
> # sites must have SiteId and column indicating if site will be used
> sites <- data.frame(designstatus$SiteID,rep(TRUE,nrow(designstatus)))
> names(sites) <- c('SiteID','Use')
>
> # Define subpopulations for which estimates are desired
> # In this case only for entire Upper Wabash Basin
> subpop <- designstatus[,c('SiteID','Basin')]
>
> # Specify design information
```

```
> dsgn <- designstatus[,c('SiteID','Stratum','final.wt','xmarinus','ymarinus')]
> dimnames(dsgn)[[2]] <- c('SiteID','Stratum','wgt','xcoord','ycoord')
>
>
> # Specify data for estimation of Site Evaluation categories
> # Add a variable that assigns categories to Target or NonTarget streams
> StatusTNT <- designstatus$StatusB
> levels(StatusTNT) <- list(T=c('TS','LD','OT','SCNB','PB'),NT=c('NT','UK'))
> data <- data.frame(designstatus[,c('SiteID','StatusB')],StatusTNT)
> names(data) <- c('SiteID','StatusB','StatusTNT')
>
> # Estimate Site Evaluation categories
> # Set known stratum size (stream length) for basin
> # Note that names matches the Stratum names in designstatus.
> stratasize <- sum(designstatus$final.wt)
> names(stratasize) <- 3
> sitestatussum <- cat.analysis.fcn(sites,subpop,dsgn,data,
+ type='Status',conf=95,vartype='Local', R=stratasize)
> # Write results out as csv file to read in spreadsheet
> write.table(sitestatussum, file = "sitestatussum.csv", sep = ",",col.names=NA)
>
>
> # Do estimates for IBI and QHEI status
> # select which sites to use - only TS-target sampled sites
> sites <- data.frame(designstatus$SiteID,designstatus$StatusB=='TS')
> names(sites) <- c('SiteID','Use')
> # Can use same subpop as before
> subpop <- designstatus[,c('SiteID','Basin')]
> # Can use same dsgn as before
> dsgn <- designstatus[,c('SiteID','Stratum','final.wt','xmarinus','ymarinus')]
> dimnames(dsgn)[[2]] <- c('SiteID','Stratum','wgt','xcoord','ycoord')
> # Set up the data data frame
> data <- designstatus[,c('SiteID','IBI.Status','QHEI.Status')]
> # Calculate the estimates: note no longer know the size of the target stream
length
> ibiqheisum <- cat.analysis.fcn(sites,subpop,dsgn,data,
+ type='Category',conf=95,vartype='Local')
> # Write results out as csv file to read in spreadsheet
> write.table(ibiqheisum, file = " ibiqheisum.csv", sep = ",",col.names=NA)
>
>
> # Complete continuous indicator estimates
> # select which sites to use - only TS-target sampled sites
> sites <- data.frame(designstatus$SiteID,designstatus$StatusB=='TS')
> names(sites) <- c('SiteID','Use')
> # Can use same subpop as before
> subpop <- designstatus[,c('SiteID','Basin')]
> # Can use same dsgn as before
> dsgn <- designstatus[,c('SiteID','Stratum','final.wt','xmarinus','ymarinus')]
> dimnames(dsgn)[[2]] <- c('SiteID','Stratum','wgt','xcoord','ycoord')
> # Set up the data data frame
> data <- designstatus[,c('SiteID','IBI.Score','QHEI.Score')]
> dimnames(data)[[2]] <- c('SiteID','IBI Score','QHEI Score')
> scoressum <- cont.analysis.fcn(sites,subpop,dsgn,data,
+ pctval=c(5, 10, 25, 50, 75, 90, 95),
+ conf=95 ,vartype='Local')
>
```

```
>
> # Write results out as csv file to read in spreadsheet
> write.table(scoressum$CDF, file = "scoressumcdf.csv", sep = ",",col.names=NA)
> write.table(scoressum$Pct, file = "scoressumpct.csv", sep = ",",col.names=NA)
> write.table(scoressum$Tot, file = "scoressumtot.csv", sep = ",",col.names=NA)
>
> # Can use R to Plot CDF estimates using function: cdfplot.fcn
> #File: cdfplot.fcn.r
> #Programmer: Tony Olsen
> #Date: Sept 23, 2002
> # Input:
> #cdfest - dataframe with
> #x-value for cdf plot
> #y-value (cdf estimate)
> #lower confidence bound
> #upper confidence bound
> #
> # Output:
> #plot of cdf with confidence bounds
> cdfplot.fcn <- function(cdfest,prop=T,...){
+
+ if(prop == T) {
+ plot(cdfest[,1],cdfest[,2]  ,
+ type='l',ylim=c(0,100),...)
+ tvalue <- cdfest[,2]>=5 & cdfest[,2]<=95
+ }
+ else {
+ plot(cdfest[,1],cdfest[,2],type='l',...)
+ tvalue <- cdfest[,2]>=0.05*max(cdfest[,2]) &
+ cdfest[,2]<=0.95*max(cdfest[,2])
+ }
+ value <- cdfest[,1][tvalue]
+ upper <- cdfest[,4][tvalue]
+ lower <- cdfest[,3][tvalue]
+ lines(value,lower,lty=2)
+ lines(value,upper,lty=2)
+
+ legend(x=min(cdfest[,1]),y=max(cdfest[,2]),
+ legend=c('CDF estimate','95% Confidence Limits'),
+ lty=c(1,2), bty='n', cex=.7)
+ }
>
> pdf('results.pdf',width=8,height=11.5)
> op <- par(mfrow=c(2,2))
> # Plot IBI Score CDFs for percent and km stream length
> tst <- scoressum$CDF$Indicator=='IBI Score'
> cdf <- scoressum$CDF[tst,c('Value','Estimate.P','LCB95Pct.P','UCB95Pct.P')]
> xlab <- 'IBI Score'
> ylab <- 'Percent Stream Length'
> cdfplot.fcn(cdf,prop=T,xlab=xlab,ylab=ylab)
> title('Upper Wabash Basin: IBI Score CDF Estimate')
>
> tst <- scoressum$CDF$Indicator=='IBI Score'
> cdf <- scoressum$CDF[tst,c('Value','Estimate.U','LCB95Pct.U','UCB95Pct.U')]
> xlab <- 'IBI Score'
> ylab <- 'Stream Length (km)'
> cdfplot.fcn(cdf,prop=F,xlab=xlab,ylab=ylab)
```

```
> title('Upper Wabash Basin: IBI Score CDF Estimate')
>
> # Plot QHEI Score CDFs for percent and km stream length
> tst <- scoressum$CDF$Indicator=='QHEI Score'
> cdf <- scoressum$CDF[tst,c('Value','Estimate.P','LCB95Pct.P','UCB95Pct.P')]
> xlab <- 'QHEI Score'
> ylab <- 'Percent Stream Length'
> cdfplot.fcn(cdf,prop=T,xlab=xlab,ylab=ylab)
> title('Upper Wabash Basin: QHEI Score CDF Estimate')
>
> tst <- scoressum$CDF$Indicator=='QHEI Score'
> cdf <- scoressum$CDF[tst,c('Value','Estimate.U','LCB95Pct.U','UCB95Pct.U')]
> xlab <- 'QHEI Score'
> ylab <- 'Stream Length (km)'
> cdfplot.fcn(cdf,prop=F,xlab=xlab,ylab=ylab)
> title('Upper Wabash Basin: QHEI Score CDF Estimate')
>
> par(op)
> graphics.off()
```