

# Calculation of survey indices using the fishvice R-library

Einar Hjørleifsson

March 15, 2014

## Preamble

The primary objective of this document is to describe functions and scripts in the R-package *fishvice* that should produce same results as are currently are obtained via the shell scripts.

- R version 3.0.2 (2013-09-25), x86\_64-redhat-linux-gnu
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, utils
- Other packages: data.table 1.8.8, fishvise 0.01, fjlst 1.0, geo 1.2-1, ggplot2 0.9.3.1, knitr 1.2, lubridate 1.3.0, mapdata 2.2-2, maps 2.3-2, plyr 1.8, RColorBrewer 1.0-5, reshape2 1.2.2, scales 0.2.3, stringr 0.6.2
- Loaded via a namespace (and not attached): colorspace 1.2-2, dichromat 2.0-0, digest 0.6.3, evaluate 0.4.4, formatR 0.8, grid 3.0.2, gtable 0.1.2, labeling 0.2, MASS 7.3-27, munsell 0.4.2, proto 0.3-10, tools 3.0.2

```
print("This document was created in knitr")
```

```
[1] "This document was created in knitr"
```

# Length based survey indices

## Current practice

The standardized spring survey (SMB) indices have heteroforth been calculated using a bunch of R-shell script. The base location is:

```
/net/hafkaldi/u2/reikn/Splus5/SMB.
```

In this directory the mother of all following scripts is `allarteg.sh` (see Appendix). The first step in the script is to move to a specific species directory and then run from there `BIOVISIT_R.sh` and for some stocks `BIOVISIT.fastar_R.sh` (see Appendix). In principle these `BIOVISIT` scripts calculate the abundance and biomass indices and cv for each length class by strata and aggregate those them according various specified areas. The calculation for each length class is performed via a loop within the R-script but the year loop is run within the shell script.

The output of the `BIOVISIT_R.sh` are two *data.frames* that are located in the binary *.RData* file of the respective species directories. Below is an example how these *data.frames* are accessed, taking cod as an example:

```
# attach('/net/hafkaldi/export/u2/reikn/Splus5/SMB/TORSKUR/.RData')
# str(base.visit) str(aggr.visit)
```

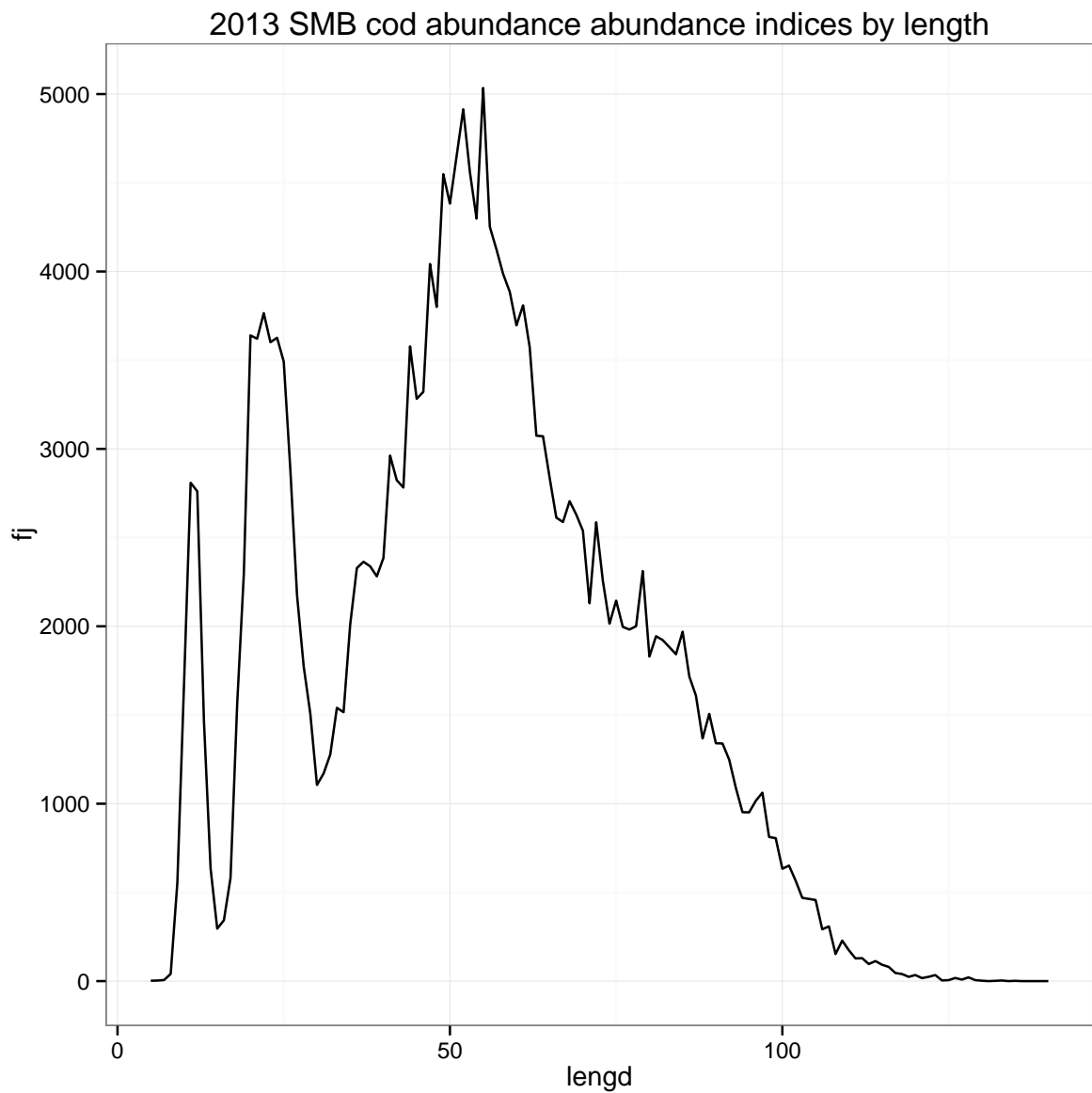
The `base.visit` contains indices values for each strata while `aggr.visit` contains indices values compiled for various areas. In both *data.frames* the indices are calculated relative for each length class for each year. The common columns are:

- `fj` Abundance indices of a given length class
- `cv.fj` Abundance coefficient of variation
- `bio.staerri` Biomass indices equal to and greater than a given length (`lengd`)
- `cv.bio.staerri` Coefficient of variation for biomass indices equal to and greater than a given length (`lengd`)
- `fj.minni` Abundance indices equal to or less than a given length (`lengd`)
- `cv.fj.minni` Coefficient of variation for biomass indices equal to and less than a given length (`lengd`)
- `lengd` The length class
- `ar` The year
- `bio` Biomass of a given length class

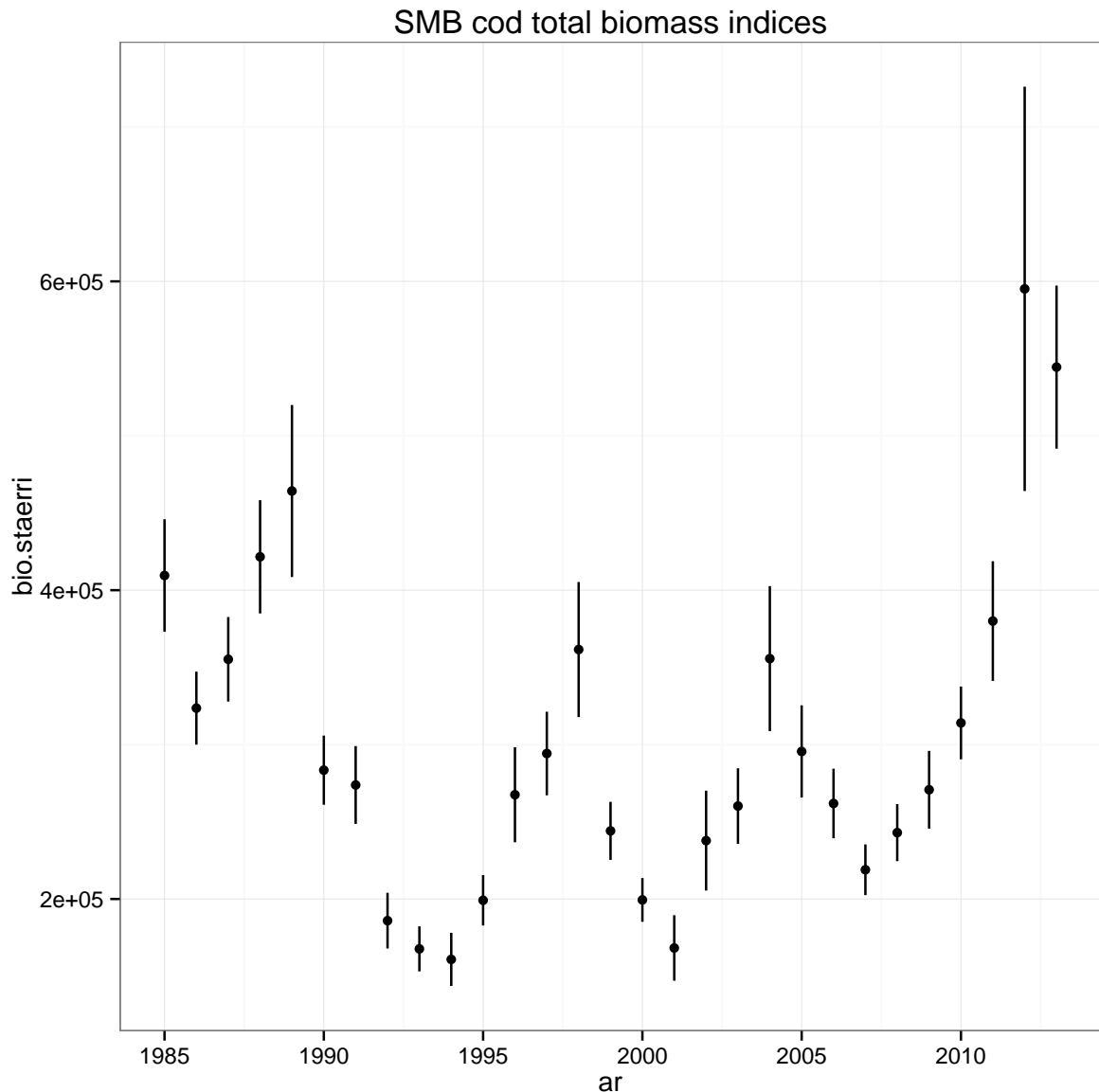
The `base.visit` in addition contains the field `strata` representing the strata and the `aggr.visit` contains the field `svaedi` containing the name of area over which the statified indices have been aggregated and the field `svaedisnr` containing a numerical value representing the area.

Here two examples how these data are often represented (using the cod as an example):

```
i <- smb_cod_aggr$ar %in% 2013 & smb_cod_aggr$svaedi %in% "Heild"
length2013.official <- smb_cod_aggr[i, ]
ggplot(length2013.official) + geom_line(aes(lengd, fj)) + labs(title = "2013 SMB cod abundance abunda
```



```
i <- smb_cod_aggr$lengd %in% 5 & smb_cod_aggr$svaedi %in% "Heild"
biomass.official <- smb_cod_aggr[i, ]
ggplot(biomass.official) + geom_pointrange(aes(ar, bio.staerri, ymin = bio.staerri *
  (1 - cv.bio.staerri), ymax = bio.staerri * (1 + cv.bio.staerri))) + labs(title = "SMB cod total b
```



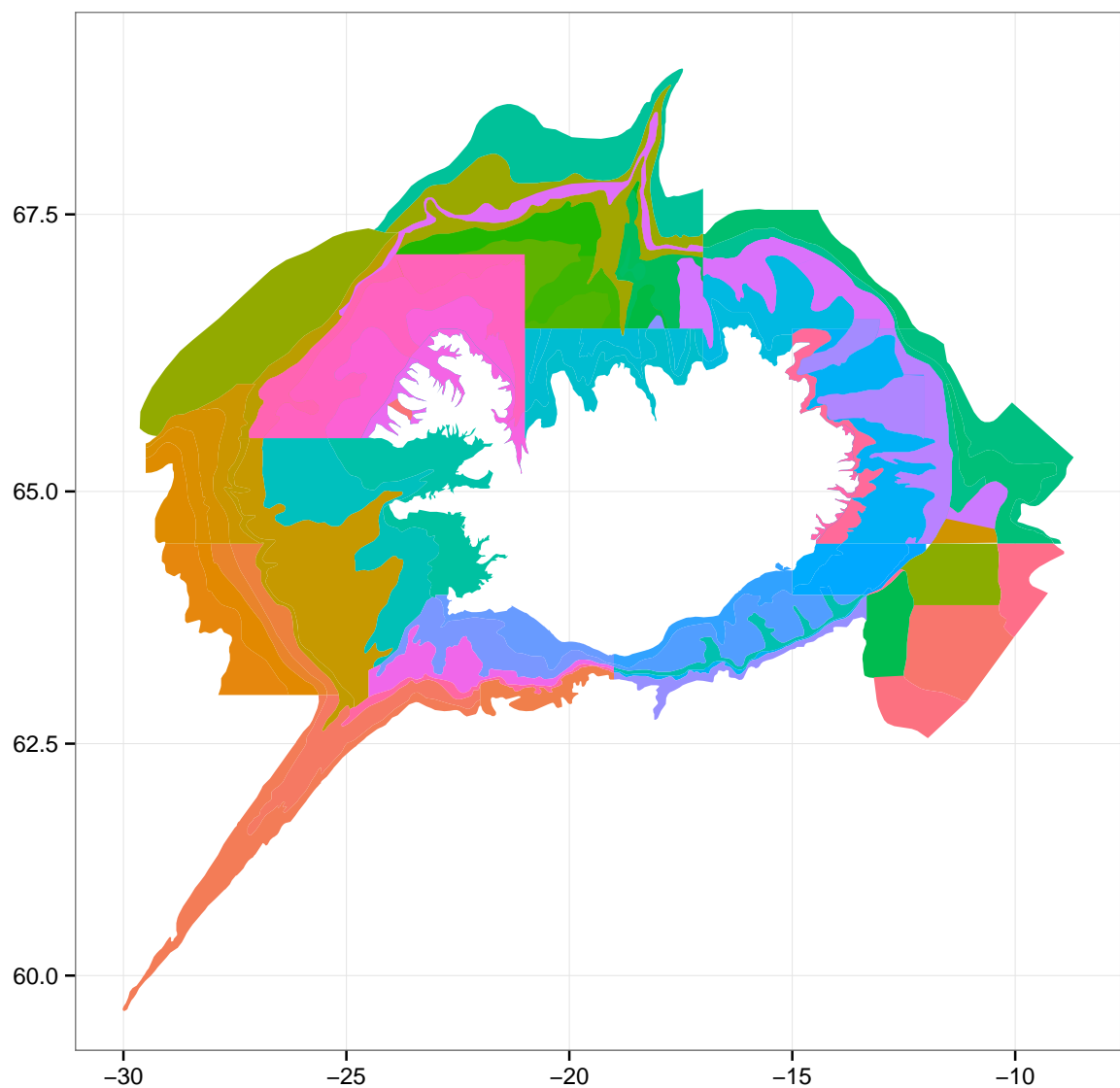
```
# detach('file:/net/hafkaldi/export/u2/reikn/Splus5/SMB/TORSKUR/.RData')
detach("file:/home/einarhj/r/Pakkar/fishvise_setup/SMB_TORSKUR.RData")

## Error:  invalid 'name' argument
```

## What is under the hood?

In the `BIOVISIT_R.sh` various functions and data are used to compile the length based indices. These are generally accessed by attaching certain binary files that reside on the network. In the library *fishvise* attempt has been made to include these objects (functions and data), the objective being to ...

```
tmp <- ldply(STRATAS)
ggplot(tmp, aes(lon, lat, group = .id, fill = .id)) + geom_polygon() + theme(legend.position = "none")
axis.title.x = element_blank(), axis.title.y = element_blank()) + coord_map()
```



## The fishvice library

The following describes steps that can be used to obtain calculation of survey indices using the *fishvice* library.

### Cod 2013 SMB abundance by length class

1. Load the library:

```
require(fishvice)
```

2. Specify the year and the species and indicate if the indices should be calculated by sex:

```
yr <- 2013
TEG <- 1
KYN <- FALSE
```

- (a) Set the length-weight coefficient:

```
lwcoeff <- c(0.01, 3)
```

(b) Or one could obtain them from a data object stored in the fishvice library:

```
lwcoeff <- LWCOEFF[[as.character(TEG)]]
lwcoeff
## [1] 0.01 3.00
```

(a) Set the length classes for which the indices should be compiled:

```
lengthClass <- c(5:140)
```

(b) Or one could obtain them from a data object stored in the fishvice library:

```
lengthClass <- LENGDIR[[as.character(TEG)]]
```

3. Subset the data:

Here we will use the station list stored in the fishvice library (smbSTODVAR)

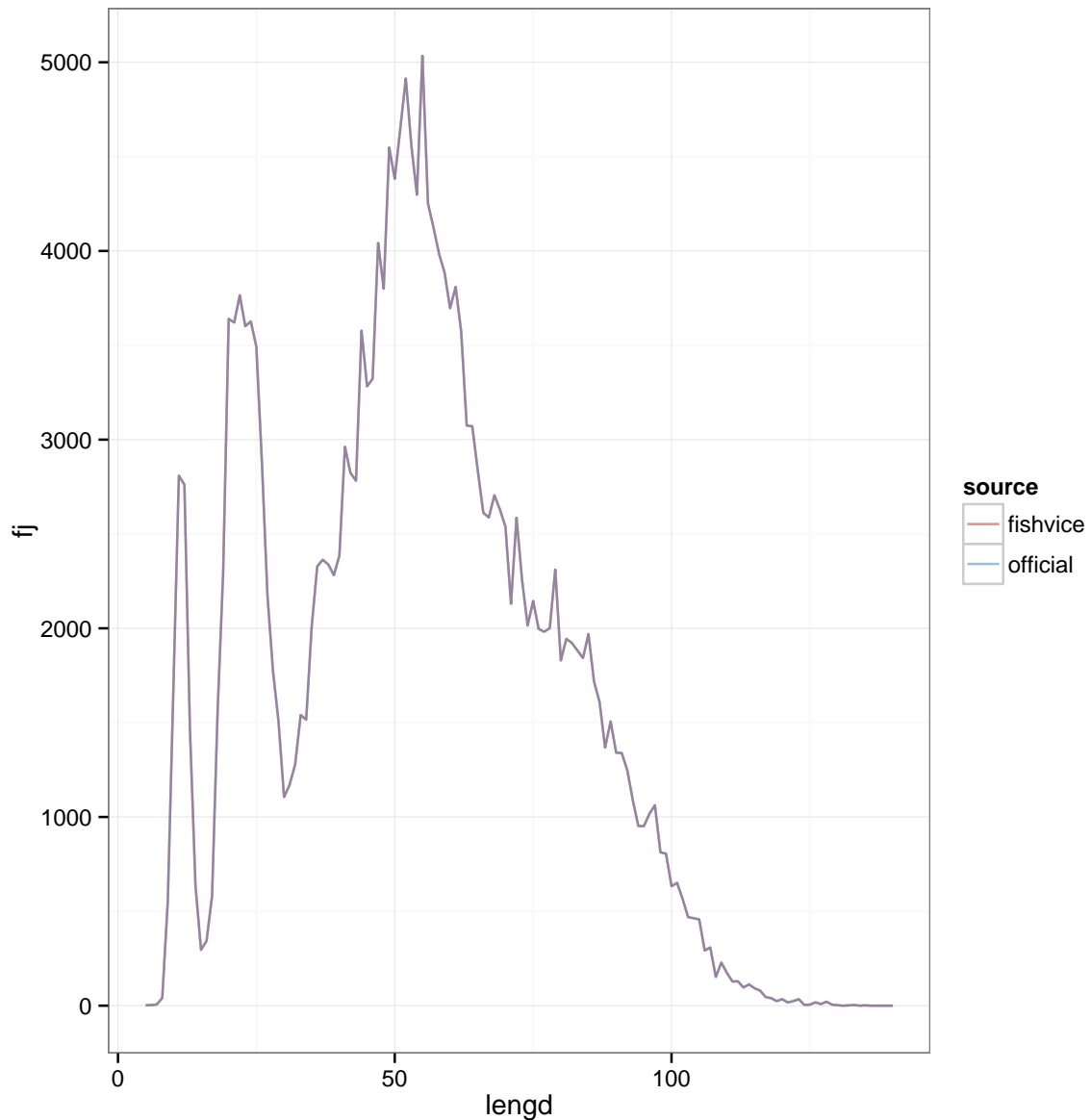
```
require(fjlst)
Oracle <- FALSE
st <- smbSTODVAR[[yr - 1984]][smbSTODVAR[[yr - 1984]]$tognumber %in% 1:39, ]
st <- st[, c("synis.id", "ar", "toglengd", "newstrata")]
le <- lesa.lengdir(st$synis.id, TEG, col.names = "kyn")
nu <- lesa.numer(st$synis.id, TEG)
le <- Skala.med.toldum(le, nu)
le$fj.all$[is.na(le$fj.all)] <- 0
le$bio <- le$fj.all * lwcoeff[1] * le$lengd^lwcoeff[2]/1e+06 #tonn
le$fj.all$ <- le$fj.all/1000 # thousands
```

4. Calculate the index:

```
d <- bioIndex(lengths = le, stations = st, sex = KYN, lenClass = lengthClass,
  yr = yr, lwcoeff = lwcoeff)
```

Lets look at the results and make comparison with the 'offical' calculation (see above):

```
d1 <- d$aggr[d$aggr$svaedi %in% "Heild", ]
d1$source <- "fishvice"
length2013.official$source <- "official"
d <- rbind(d1, length2013.official)
ggplot(d, aes(lengd, fj, colour = source)) + geom_line(alpha = 0.5) + scale_colour_brewer(palette = "magma")
```



### Calculating biomass indices for all the years

The example shows how to calculate the total biomass index for cod. Because we are interested in the total biomass indices only we calculate the only values for the 5 cm length class. This being so because the value we are interested in is `bio.staerri` than 5 cm. The steps done here are similar to that described above with the addition of creating a loop over the years:

```
TEG <- 1
KYN <- FALSE
lwcoeff <- LWCOEFF[[as.character(TEG)]]
lengdir <- LENGDIR[[as.character(TEG)]]
Oracle <- FALSE
years <- c(1985:2013)
for (j in 1:length(years)) {
  st <- smbSTODVAR[[years[j] - 1984]][smbSTODVAR[[years[j] - 1984]]$tognumber %in%
    1:39, ]
  le <- lesa.lengdir(st$synis.id, TEG, col.names = "kyn")
  nu <- lesa.numer(st$synis.id, TEG)
  le <- Skala.med.toldum(le, nu)
  le$fj.all[is.na(le$fj.all)] <- 0
}
```



```

le$bio <- le$fj.alls * lwcoeff[1] * le$lengd~lwcoeff[2]/1e+06 #tonn
le$fj.alls <- le$fj.alls/1000 # thousands
tmp <- bioIndex(lengths = le, stations = st, sex = KYN, lenClass = 5, yr = years[j],
  lwcoeff = lwcoeff)
if (j == 1)
  d1 <- tmp$aggr[tmp$aggr$svaedi %in% "Heild", ]
if (j > 1)
  d1 <- rbind(d1, tmp$aggr[tmp$aggr$svaedi %in% "Heild", ])
}

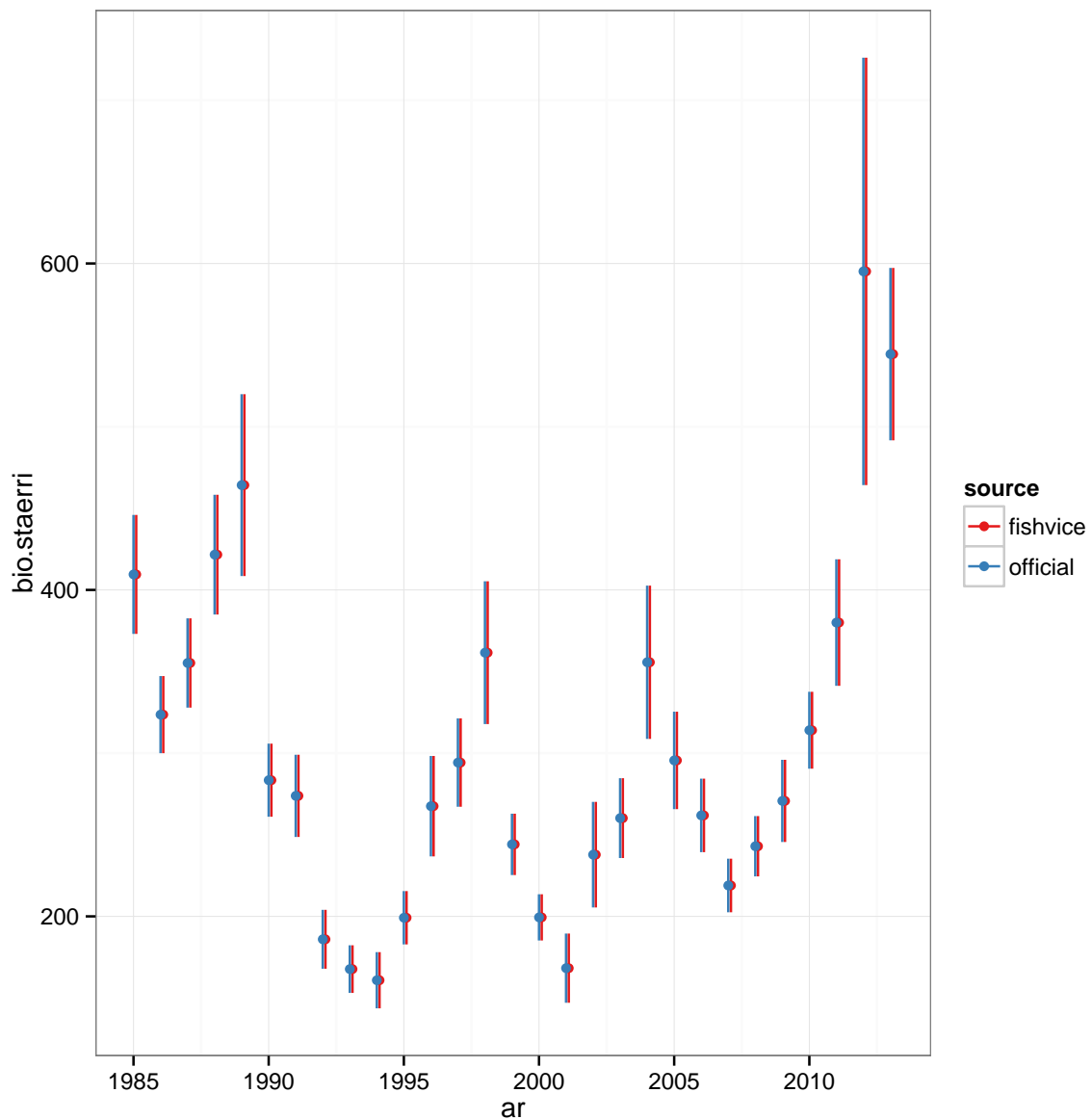
```

Lets look at the results and make comparison with the 'official' calculation (see above):

```

d1$bio.staerri <- d1$bio.staerri/1000
d1$source <- "fishvice"
d1$ar <- d1$ar + 0.1
biomass.official$bio.staerri <- biomass.official$bio.staerri/1000
biomass.official$source <- "official"
d <- rbind(d1, biomass.official)
ggplot(d) + geom_pointrange(aes(ar, bio.staerri, ymin = bio.staerri * (1 - cv.bio.staerri),
  ymax = bio.staerri * (1 + cv.bio.staerri), colour = source)) + scale_colour_brewer(palette = "Set1")

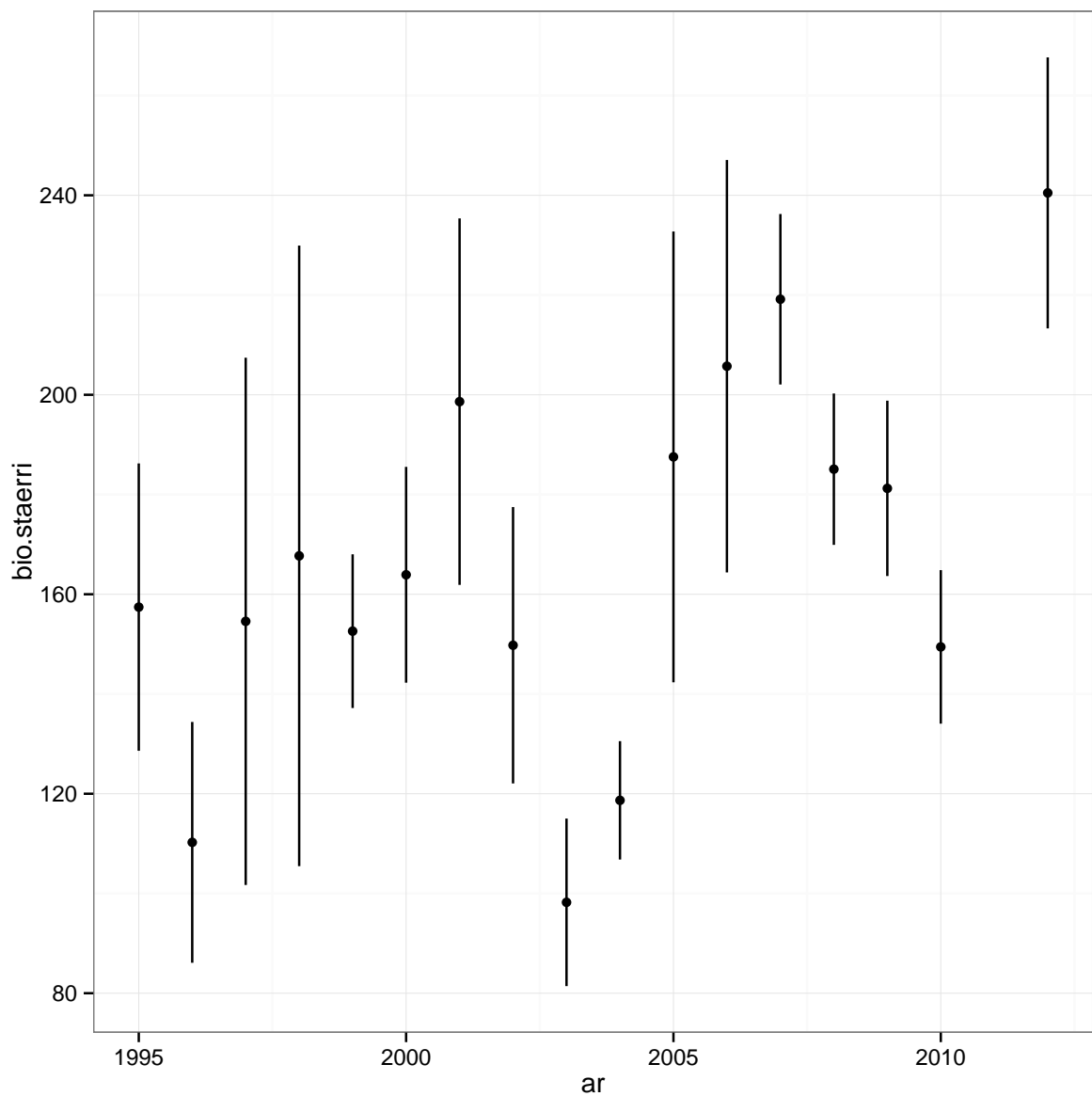
```



Lets try the fall survey:

```
TEG <- 1
KYN <- FALSE
lwcoeff <- LWCOEFF[[as.character(TEG)]]
lengdir <- LENGDIR[[as.character(TEG)]]
Oracle <- FALSE
years <- c(1995:2010, 2012)
for (j in 1:length(years)) {
  st <- smhSTODVAR[[years[j] - 1994]][smhSTODVAR[[years[j] - 1994]]$tognumer %in%
    1:39, ]
  le <- lesa.lengdir(st$synis.id, TEG, col.names = "kyn")
  nu <- lesa.numer(st$synis.id, TEG)
  if (nrow(nu) == 1)
    nu <- rbind(nu, nu) # Vegna villu í skala.med.toldum
  if (nrow(le) > 0) {
    le <- Skala.med.toldum(le, nu)
    le$fj.all$[is.na(le$fj.all)] <- 0
    le$bio <- le$fj.all * lwcoeff[1] * le$lengd~lwcoeff[2]/1e+06 #tonn
    le$fj.all <- le$fj.all/1000 # thousands
  }
  if (nrow(le) == 0) {
    le <- data.frame(synis.id = st$synis.id[1:2], lengd = lengdir[1:2],
      kyn = c(1, 1), fj.all = c(0, 0), bio = c(0, 0))
  }
  tmp <- bioIndex(lengths = le, stations = st, sex = KYN, lenClass = 5, yr = years[j],
    lwcoeff = lwcoeff)
  if (j == 1)
    d2 <- tmp$aggr[tmp$aggr$svaedi %in% "Heild", ]
  if (j > 1)
    d2 <- rbind(d2, tmp$aggr[tmp$aggr$svaedi %in% "Heild", ])
}

d2$bio.staerri <- d2$bio.staerri/1000
ggplot(d2) + geom_pointrange(aes(ar, bio.staerri, ymin = bio.staerri * (1 -
  cv.bio.staerri), ymax = bio.staerri * (1 + cv.bio.staerri)))
```



```
smh1995_2012 <- d2
```

Both on the same graph:

```
d1$ar <- d1$ar + 0.25
d1$sur <- "smb"
d2$ar <- d2$ar + 0.75
d2$sur <- "smh"
d <- rbind(d1, d2)

## Error: numbers of columns of arguments do not match

ggplot(d) + geom_pointrange(aes(ar, bio.staerri, ymin = bio.staerri * (1 - cv.bio.staerri),
  ymax = bio.staerri * (1 + cv.bio.staerri), colour = sur))

## Error: object 'sur' not found
```

# Appendix

Scripts used in calculation of spring survey indices

## allarteg.sh

The shell script `allarteg.sh` resides in `/net/hafkaldi/u2/reikn/Splus5/SMB:`

```
echo "torskur"
cd TORSKUR
../BIOVISIT_R.sh
../BIOVISIT.fastar_R.sh
echo "ysa"
cd ../YSA
../BIOVISIT_R.sh
../BIOVISIT.fastar_R.sh
echo "karfi"
cd ../KARFI
../BIOVISIT.fastar_R.sh
../BIOVISIT_R.sh
echo "langlura"
cd ../LANGLURA
../BIOVISIT.fastar_R.sh
../BIOVISIT_R.sh
cd ../SKARKOLI
../BIOVISIT.fastar_R.sh
../BIOVISIT_R.sh
echo "keila"
cd ../KEILA
../BIOVISIT_R.sh
echo "langa"
cd ../LANGA
../BIOVISIT_R.sh
echo "hlyri"
cd ../HLYRI
../BIOVISIT_R.sh
echo "blalanga"
cd ../BLALANGA
../BIOVISIT_R.sh
echo "tylura"
cd ../TYLURA
../BIOVISIT_R.sh
echo "steinbitur"
cd ../STEINBITUR
../BIOVISIT_R.sh
echo "ufsi"
cd ../UFSI
../BIOVISIT_R.sh
../BIOVISIT.fastar_R.sh
echo "sandkoli"
cd ../SANDKOLI
../BIOVISIT_R.sh
echo "tindaskata"
cd ../TINDASKATA
../BIOVISIT_R.sh
echo "luda"
cd ../LUDA
../BIOVISIT_R.sh
echo "djupkarfi"
```

```

cd ../DJUPKARFI
../BIOVISIT_R.sh
echo "skrapflura"
cd ../SKRAPFLURA
../BIOVISIT_R.sh
echo "skotuselur"
cd ../SKOTUSELUR
../BIOVISIT_R.sh
echo "gulllax"
cd ../GULLLAX
../BIOVISIT_R.sh
echo "HROGNKELSI"
cd ../HROGNKELSI
../BIOVISIT_R.sh
cd ../GRALUDA
../BIOVISIT_R.sh

```

## BIOVISIT\_R.sh

The shell script `allarteg.sh` resides in `/net/hafkaldi/u2/reikn/Splus5/SMB:`

```

for yr in 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013
do
echo '
year <- 999
print(year)
Rattach("../")
Rattach("../GEOMETRY.NEW")
lwcoeff <- LWCOEFF[[as.character(TEG)]]
lengdir <- LENGDIR[[as.character(TEG)]]
st1 <- STODVAR$y999[STODVAR$y999$tognumber %in% 1:39,]
tmp <- lesa.lengdir(st1$synis.id,TEG,col.names="kyn")
tmp1 <- lesa.numer(st1$synis.id,TEG)
tmp <- Skala.med.toldum(tmp,tmp1)
i <- is.na(tmp$fj.all)
tmp$fj.all[i] <- 0
tmp$bio <- tmp$fj.all*lwcoeff[1]*tmp$lengd~lwcoeff[2]/1e6 # tonn
tmp$fj.all <- tmp$fj.all/1e3 # þúsundir

for( i in 1:length(lengdir)) {
print(i)
tmp1 <- tmp[tmp$lengd==lengdir[i],]
if(nrow(tmp1) > 0) {
x <- apply.shrink(tmp1$fj.all,tmp1$synis.id,sum)
names(x) <- c("synis.id","fj")
st <- join(st1[,c("newstrata","toglengd","synis.id")],x,"synis.id",set=0)
}
else {
st <- st1[,c("newstrata","toglengd","synis.id")]
st$fj <- rep(0,nrow(st))
}

if(KYN) {
tmp1 <- tmp[tmp$lengd==lengdir[i] & tmp$kyn==1 & !is.na(tmp$kyn),]
if(nrow(tmp1) > 0) {
x <- apply.shrink(tmp1$fj.all,tmp1$synis.id,sum)
names(x) <- c("synis.id","fjhaenga")
st <- join(st,x,"synis.id",set=0)
}
}
}

```

```

}

else {
  st$fjhaenga <- rep(0,nrow(st))
}
tmp1 <- tmp[tmp$lengd==lengdir[i] & tmp$kyn==2 & !is.na(tmp$kyn),]
if(nrow(tmp1) > 0) {
  x <- apply.shrink(tmp1$fj.all, tmp1$synis.id, sum)
  names(x) <- c("synis.id", "fjhrygna")
  st <- join(st, x, "synis.id", set=0)
}
else {
  st$fjhrygna <- rep(0,nrow(st))
}
}

tmp1 <- tmp[tmp$lengd >= lengdir[i],]
if(nrow(tmp1) > 0) {
  x <- apply.shrink(tmp1$bio, tmp1$synis.id, sum)
  names(x) <- c("synis.id", "bioge")
  st <- join(st, x, "synis.id", set=0)
}
else {
  st$bioge <- rep(0,nrow(st))
}

tmp1 <- tmp[tmp$lengd <= lengdir[i],]
if(nrow(tmp1) > 0) {
  x <- apply.shrink(tmp1$fj.all, tmp1$synis.id, sum)
  names(x) <- c("synis.id", "fjle")
  st <- join(st, x, "synis.id", set=0)
}
else {
  st$fjle <- rep(0,nrow(st))
}

tmp.visit <- Calc.index(st, "fj")
tmp.biovisit <- Calc.index(st, "bioge")
tmp.seidavisit <- Calc.index(st, "fjle")
if(KYN) {
  tmp.haengavisit <- Calc.index(st, "fjhaenga")
  tmp.hrygnuvisit <- Calc.index(st, "fjhrygna")
}

tmp.visit1 <- tmp.visit$result[, c("strata", "total", "cv")]
names(tmp.visit1)[2:3] <- c("fj", "cv.fj")
tmp.visit1$bio.staerri <- tmp.biovisit$result[, "total"]
tmp.visit1$cv.bio.staerri <- tmp.biovisit$result[, "cv"]
tmp.visit1$fj.minni <- tmp.seidavisit$result[, "total"]
tmp.visit1$cv.fj.minni <- tmp.seidavisit$result[, "cv"]
if(KYN) {
  tmp.visit1$fj.haenga <- tmp.haengavisit$result[, "total"]
  tmp.visit1$cv.fj.haenga <- tmp.haengavisit$result[, "cv"]
  tmp.visit1$fj.hrygna <- tmp.hrygnuvisit$result[, "total"]
  tmp.visit1$cv.fj.hrygna <- tmp.hrygnuvisit$result[, "cv"]
}

tmp.visit2 <- tmp.visit$aggr.output[, c("total", "cv")]
names(tmp.visit2) <- c("fj", "cv.fj")

```

```

tmp.visit2$bio.staerri <- tmp.biovisit$aggr.output[, "total"]
tmp.visit2$cv.bio.staerri <- tmp.biovisit$aggr.output[, "cv"]
tmp.visit2$fj.minni <- tmp.seidavisit$aggr.output[, "total"]
tmp.visit2$cv.fj.minni <- tmp.seidavisit$aggr.output[, "cv"]
if(KYN) {
tmp.visit2$fj.haenga <- tmp.haengavisit$aggr.output[, "total"]
tmp.visit2$cv.fj.haenga <- tmp.haengavisit$aggr.output[, "cv"]
tmp.visit2$fj.hrygna <- tmp.hrygnuvisit$aggr.output[, "total"]
tmp.visit2$cv.fj.hrygna <- tmp.hrygnuvisit$aggr.output[, "cv"]
}

tmp.visit2$svaedi <- dimnames(tmp.visit2)[[1]]
tmp.visit2$svaedisnr <- 1:nrow(tmp.visit2)
dimnames(tmp.visit2)[[1]] <- 1:nrow(tmp.visit2)
tmp.visit1$lengd <- rep(lengdir[i], nrow(tmp.visit1))
tmp.visit2$lengd <- rep(lengdir[i], nrow(tmp.visit2))
tmp.visit1$ar <- rep(999, nrow(tmp.visit1))
tmp.visit2$ar <- rep(999, nrow(tmp.visit2))
if(i == 1) {
tmp.visit1a <- tmp.visit1
tmp.visit2a <- tmp.visit2
}
else {
tmp.visit1a <- rbind(tmp.visit1a, tmp.visit1)
tmp.visit2a <- rbind(tmp.visit2a, tmp.visit2)
}
}
if(year == 1985) {
base.visit <- tmp.visit1a
aggr.visit <- tmp.visit2a
}
if(year != 1985) {
base.visit <- rbind(base.visit, tmp.visit1a)
aggr.visit <- rbind(aggr.visit, tmp.visit2a)
}
' | sed "s/999/$yr/g" | R --save > /dev/null
done

echo '
Rattach("..")
Rattach("../GEOMETRY.NEW")
lwcoeff <- LWCOEFF[[as.character(TEG)]]
# tonn því fjöldi er þegar / 1000 en lengd-þyngdar gefur grömm.
aggr.visit$bio <- aggr.visit$fj*lwcoeff[1]*aggr.visit$lengd^lwcoeff[2]/1e3
base.visit$bio <- base.visit$fj*lwcoeff[1]*base.visit$lengd^lwcoeff[2]/1e3
' | R --save > /dev/null

```