cede

Malcolm Haddon

2018-06-22

Table of Contents

Catch Effort and Data Exploration

When working with relatively minor commercial species the data available is typically less comprehensive than for species that might be considered to be the economic drivers of a fishery. Nevertheless, data exploration, perhaps through plotting up different variables and how they might change through the years, can often be informative about changes in any fishery for a particular species. The **cede** R package (Catch Effort and Data Exploration) includes an array of functions that should assist with such data exploration. If a species' fishery includes CPUE data then plots of the distribution of catches, effort, and CPUE (perhaps as Log(CPUE)) can be helpful in the interpretation of such CPUE, especially if there is sufficient data to allow for CPUE standardization. **cede** now includes various functions that can also assist with CPUE standardization. All these functions are described below with examples of their use.

There should be no expectation that the functions to be used in the standardization of CPUE constitute anything like a complete treatment. This vignette only provide a very brief introduction or pointer to get people started. There are many aspects not considered (e.g. how or whether to treat zeros). This vignette remains a draft and if you find errors, omissions, or obscurities do please let me know (see DESCRIPTION for email address). In addition if you wish to reference this package when writing your SAFS assessment you can obtain one by typing `citation("cede")` into the console, which will give you the latest version.

Data Exploration

The main data set included with **cede** is called *sps* and contains typical fisheries data from a scalefish fishery. It is there mainly to assist with learning the operation of the different functions. Generally it would be better to use your own data but if you consider the *sps* data set you will gain an understanding of a typical format.

```
data(sps)
kable(sps[1:6,],digits=c(0,0,0,0,3,3,0,0,2,0))
```

Year	Month	Vessel	catch_kg	Long	Lat	Depth	DayNight	Effort	Zone
2004	4	1	220	145.117	-43.067	125	N	4.00	1
2004	4	1	280	145.250	-43.233	130	M	3.66	1
2004	4	1	180	145.150	-43.083	115	D	3.50	1
2004	4	1	70	145.233	-43.217	120	N	4.75	1
2004	4	1	200	145.100	-43.033	120	M	4.75	1
2004	4	1	100	145.767	-43.683	130	M	2.01	1

```
cat("\n")

properties(sps)

##           Index isNA Unique     Class        Min     Max    Example
## Year          1    0     12   numeric 2003.00000 2014.00       2004
## Month         2    0     12   numeric    1.00000   12.00          4
## Vessel        3    0     23   numeric    1.00000   27.00          1
## catch_kg      4    0    442   numeric    1.00000 4500.00        220
## Long          5    0    447   numeric  144.11667  146.30   145.1167
## Lat           6    0    512   numeric  -45.83333  -40.75  -43.06667
## Depth         7    0    191   numeric    2.00000  366.00        125
## DayNight      8    0      3 character    0.00000    0.00          N
## Effort        9    0    377   numeric    0.16000    9.66          4
## Zone         10    0      3   numeric    1.00000    3.00          1
```

The *properties* function categorizes the contents of a data.frame, counting the number of NAs in each variable, if any, listing their class, their minimum and maximum (if applicable) and finally printing an example of the contents. I find this function quite useful when beginning to use a different data.frame. Generally I refer to variables within a data.frame by their names so it is important to know if they are capitalized or not as well as knowing exactly which variables are present.

Once we have our data available for analysis it is often a good idea to find ways to summarize how they vary relative to one another. With fisheries data it is common to want to know how different factors influence the total catch and whether these vary by year. Typically one might use the R function *tapply* to conduct such examinations. To simplify this use one can use the *tapsum* function from within **cede**.

The seasonality of catches can be indicative of the typical behaviour of the fishery within a year.

```
kable(tapsum(sps,"catch_kg","Year","Month"),digits=c(1,1,1,1,1,1,1,1,1
,1,1,1))
```

	1	2	3	4	5	6	7	8	9	10	11	12
2003	33.6	26.0	37.3	30.4	14.7	3.7	4.8	11.6	14.5	5.1	6.4	33.8
2004	73.7	66.2	52.7	100.8	55.9	18.3	12.7	22.8	8.4	9.4	30.1	21.6
2005	114.9	83.9	35.0	37.4	7.3	15.1	11.8	6.1	4.1	13.3	13.9	36.0
2006	79.8	53.1	45.8	27.4	0.3	1.8	2.8	3.1	0.4	5.1	9.2	55.7
2007	31.8	60.1	27.3	1.5	13.6	4.6	2.5	0.8	0.3	0.2	7.0	20.6
2008	76.3	21.6	33.0	5.5	2.1	0.7	1.3	0.5	0.2	3.2	6.4	14.1
2009	16.7	25.4	9.5	2.5	2.4	0.7	0.6	2.0	0.7	6.7	18.2	11.2
2010	40.9	22.5	11.4	2.0	0.3	0.5	1.8	2.3	1.4	1.6	0.7	4.4
2011	25.0	38.6	10.6	6.3	2.7	3.2	1.5	2.7	2.1	2.2	5.1	23.5
2012	35.3	49.4	24.9	6.4	2.9	2.6	5.1	1.6	1.6	3.4	4.4	13.1
2013	47.3	48.8	41.0	11.0	17.1	0.3	2.3	0.5	1.3	6.6	6.3	6.4
2014	11.0	10.3	21.5	12.1	6.4	11.0	8.1	15.5	3.9	3.8	26.6	49.4

Here we have examined the catch by zone where the zones are in sequence along the coast (or they would be if this was a real fisheries data).

```
tapsum(sps,"catch_kg","Year","Zone")

##             1          2      3
## 2003  94.6190  98.06400 29.197
## 2004 215.2230 210.47900 46.804
## 2005 112.7670 216.02300 50.079
## 2006  82.4370 120.29100 81.663
## 2007  42.7560  91.46240 36.161
## 2008  51.9840  93.81300 19.020
## 2009  33.9920  33.62310 28.931
## 2010  11.8070  18.71400 59.165
## 2011  37.1840  79.41725  6.892
## 2012  55.2330  65.35600 30.263
## 2013  50.3015  83.81800 54.848
## 2014  46.6240  81.44250 51.455
```

We are not limited to summarizing catch but, for example could also look at the distribution of effort as total number of hours (note the change to the default value of div so that the total number of hours is not divided by 1000). By pointing the function call to a new object one can then plot the results.

```
effbyyr <- tapsum(sps,"Effort","Year","Zone",div=1.0)
effbyyr

##            1       2      3
## 2003 2473.36 1998.01 724.13
## 2004 3558.32 2541.13 709.58
## 2005 2095.92 2750.78 639.01
## 2006 2001.37 2055.52 941.46
## 2007 1192.94 1279.45 481.96
## 2008 1426.79 1072.82 495.61
## 2009  877.81  739.13 488.86
## 2010  471.06  493.39 691.16
## 2011  855.54 1185.06 293.93
## 2012 1278.07  981.93 508.41
## 2013 1323.23  960.89 816.47
## 2014 1036.63 1222.02 681.42
```

```
# plotprep(width=7,height=4.5)
ymax <- max(effbyyr,na.rm=TRUE)
label <- colnames(effbyyr)
yrs <- as.numeric(rownames(effbyyr))
par(mfrow=c(1,1),mai=c(0.45,0.45,0.05,0.05))
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)
plot(yrs,effbyyr[,label[1]],type="l",lwd=2,col=1,ylim=c(0,ymax),
     ylab="Total Effort (Hours) by Zone per Year",xlab="",
     panel.first=grid())
lines(yrs,effbyyr[,label[2]],lwd=2,col=2)
lines(yrs,effbyyr[,label[3]],lwd=3,col=3)
legend("topright",label,col=c(1,2,3),lwd=3,bty="n",cex=1.25)
```

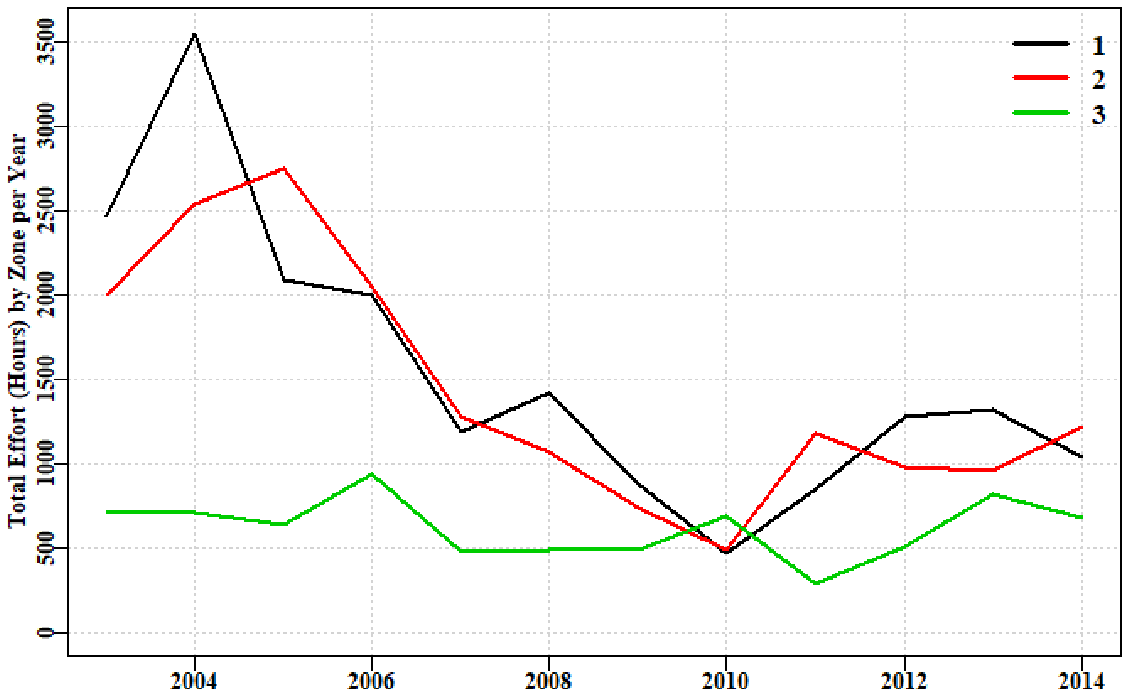


Figure 1. A plot of total effort by zone, showing that a visual illustration can often more easily highlight changes in a fishery's dynamics.

DayNight is another factor that can have large consequences for catches and catch rates. Check the description of the *sps* data set using `?sps

```
tapsum(sps,"catch_kg","Year","DayNight")

##              D        M       N
## 2003  80.54300  81.3930 59.9440
## 2004 226.67300 153.7910 92.0420
## 2005 157.21800 133.5640 88.0870
## 2006 127.24900 104.6120 52.5300
## 2007  72.13700  61.5024 36.7400
## 2008  75.67900  56.9030 32.2350
## 2009  35.10710  34.7680 26.6710
## 2010  39.00500  25.8060 24.8750
## 2011  46.14625  44.6535 32.6935
## 2012  52.92000  59.4950 38.4370
## 2013  72.16750  66.8170 49.9830
## 2014  52.40750  64.0420 63.0720
```

One of the most influential factors within each fishery is the vessel doing the catching. Often this is also a reflection of the skipper of the vessel as well as the relative performance of the boat itself. Nevertheless, it is often the case the vessel name is the only information available about the vessel's relative fishing power. It is possible to pay special attention to catch-per-vessel, although the following analysis is more general than that and can be applied to, for example, catch-by-month relative to Depth Category.

```
cbv <- tapsum(sps,"catch_kg","Vessel","Year") # often more vessels
than years
total <- rowSums(cbv,na.rm=TRUE)
cbv1 <- cbv[order(total),]   # sort by total catch
kable(cbv1,digits=c(1,1,1,1,1,1,1,1,1,1,1,1))
```

	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
21	0.1											
27						0.2						
16	0.1		0.2									
20	0.8											
24			0.2		0.6							
23			1.2									
19	0.0						0.1	0.0	0.8	0.4	0.3	0.2
17	2.5											
25								0.1	0.1		0.7	2.8
11	0.0	3.7	7.5									
4	0.3	11.0										
12	3.0	4.9	1.2	3.0			0.1					
14	6.2	0.3	0.6	4.6	6.1	1.2	0.5				0.0	0.1
10	9.8	18.0	22.8	22.0								
9	1.2	9.3	1.4	6.1	15.0	6.4	2.1	1.9	5.8	3.2	12.5	13.2
6	19.4	41.7	3.7	13.8								
8	38.6	32.1	17.2									
5	0.1	27.7	40.8	29.5								9.2
13	8.6	8.2	10.1	4.3	26.7	9.5	1.1	4.0	0.3	9.3	8.9	17.1
3	31.6	15.7	39.9	32.8	21.7	16.3	12.3	25.8	20.0	21.4	41.7	15.0
7	45.3	37.8	61.5	49.7	14.1	20.7	11.5	6.2	6.3	16.2	23.0	3.2
1	1.7	107.3	73.1	32.6	23.3	25.1	18.3	17.1	30.7	42.9	31.4	80.2
2	52.5	154.8	97.3	86.0	63.0	85.4	50.5	34.6	59.5	57.5	70.5	38.4

Obviously some vessels will be much more influential than others simply because they catch a great deal more than others and hence introduce many more records into the database.

```
# plotprep(width=8,height=6) # not needed in the vignette
to <- turnover(cbv1)
yearBubble(cbv1,ylabel="sqrt(catch-per-vessel)",
           diam=0.125,txt=c(2,3,4,5),hline=TRUE)
```

Figure 2. This hypothetical fishery is clearly dominated by four or five vessels with numerous minor players. Additionally, before 2007 there were a few more productive fishers present (this reflects the structural adjustment in the Commonwealth from which this simulated data derives). The optional horizontal lines merely delineate the individual vessels. The top two rows of numbers is the total catch per year and the bottom row of numbers is the number of vessels reporting in each year.

It is likely that if the data from the bottom nine vessels were omitted there would be no effect on any results as their catches are so minor in a relative sense. It is clear those vessels are merely casual occurrences within the fishery.

While the main vessels were reasonably consistent in terms of reporting from this fishery other vessels came and went. To summarize such activity one can use the *turnover* function which sumarizes the year-to-year changes in which vessels report being active.

```
print(to)

##      Continue Leave Start Total
## 2003       19     0     0    19
## 2004       14     5     0    14
## 2005       13     1     3    16
## 2006       11     5     0    11
## 2007        7     4     1     8
## 2008        7     1     1     8
## 2009        7     1     2     9
## 2010        7     2     1     8
## 2011        8     0     0     8
## 2012        7     1     0     7
## 2013        7     0     2     9
## 2014        9     0     1    10
```

The Continue column lists how many continued from the preceding year, the Leave column designates how many left relative to the previous year, while the Start column is literally how many started reporting in that year. The Total is the total reporting in each year. No attempt is made to follow individual vessels.

The Addition of CPUE data

You will have noticed that the data came with catch and effort but not CPUE, so we need to calculate that. In the following I test for the presence of zeros in the catch and effort to avoid generating errors of division (divide by zero errors will stop the analysis) and when taking logs. In fact, as the *properties* call showed there were no *NA* values, but is remains worth checking. While we are adding CPUE we can also group the depth data into depth classes to provide that option when standardizing the CPUE data.

```
sps$CE <- NA     # make space in the data.frame
sps$LnCE <- NA
pick <- which((sps$catch_kg > 0) & (sps$Effort > 0))
sps$CE[pick] <- sps$catch_kg[pick]/sps$Effort[pick]
sps$LnCE[pick] <- log(sps$CE)   # natural log-transformation
# categorize Depth
range(sps$Depth,na.tm=TRUE)     # to aid selection of depth class
width

## [1]   1 366

sps$DepCat <- NA
sps$DepCat <- trunc(sps$Depth/25) * 25
table(sps$DepCat)

##
##    0   25   50   75  100  125  150  175  200  225  250  275  300
325  350
##    6   19  224 1569 4583 3593 1393   74   66   21   15   21    7
5    7
```

It is clear from the summary of records by depth that most of the fishing occurs in waters of 150 meters or less.

Tables of numbers are very informative but sometimes it is much easier to gain a visual impression of patterns in one’s data by plotting them. Typically, with fisheries data, one might plot each variable, such as catch, effort, log(CPUE), depth, etc, by year to see whether changes have occurred through time. Such changes might adversely affect any analysis applied so it is always a good idea to examine (explore) one’s data before using it. **cede** provides a function *histyear* that an plot a histogram of a selected variable by year.

```
outH <- histyear(sps,Lbound=-1.75,Rbound=8.5,inc=0.25,pickvar="LnCE",
                 years="Year",varlabel="log(CPUE)",plots=c(4,3))
```

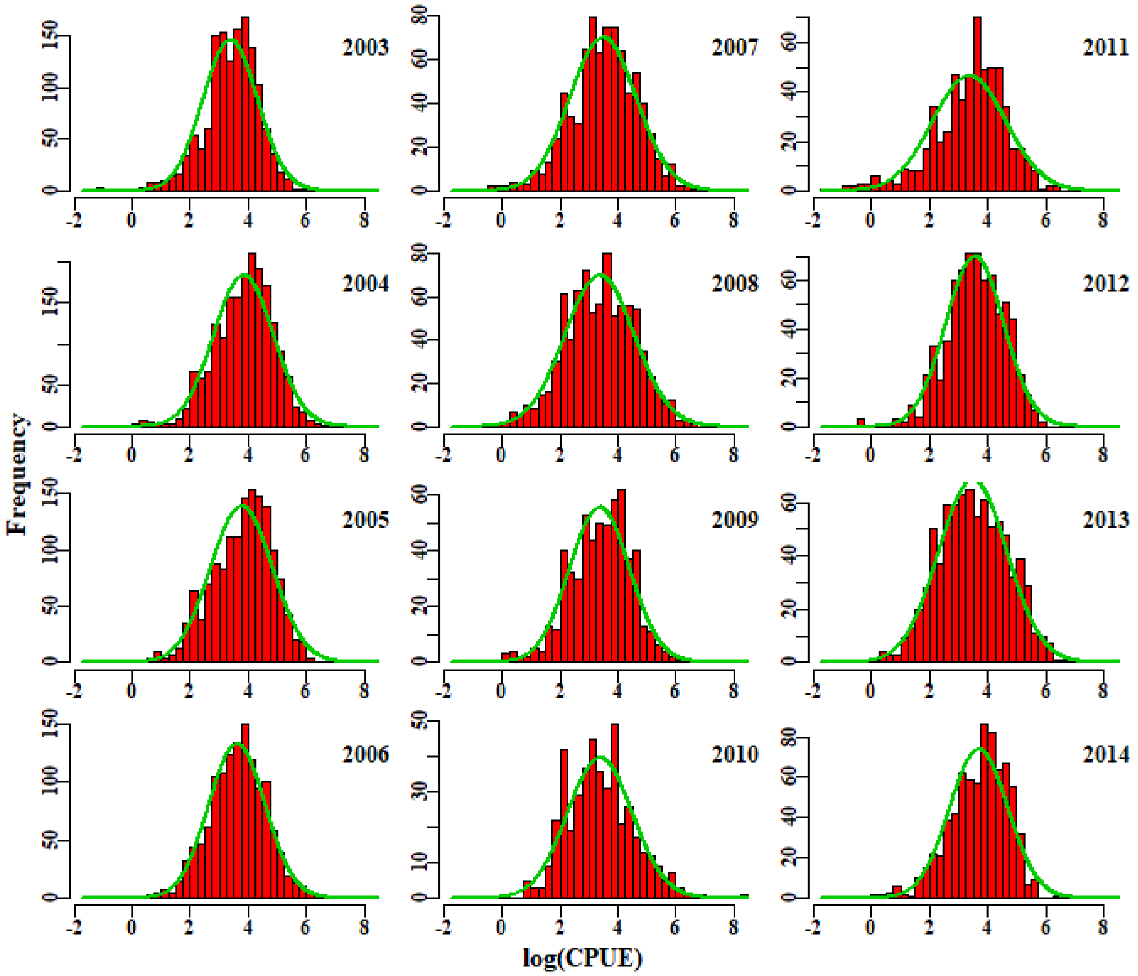


Figure 3. The distribution of the log(CPUE) each year for which data is available. The green lines are fitted normal distributions there for reference (log-transformation should normalize log-normal data).

```
outH <- histyear(sps,Lbound=0,Rbound=375,inc=12.5,pickvar="Depth",
                 years="Year",varlabel="Depth
(m)",plots=c(4,3),vline=120)
```

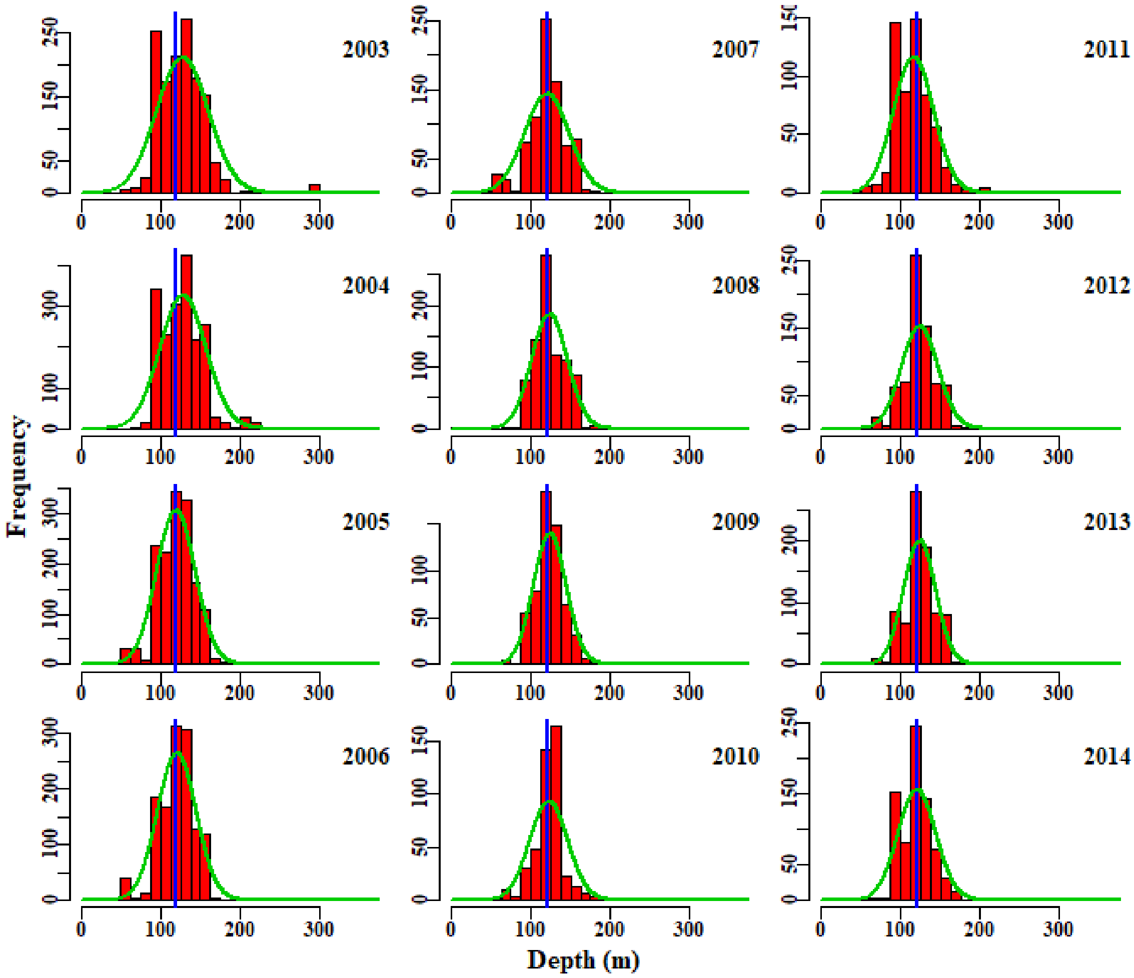


Figure 4. The distribution of reported mean depth of fishing each year. The green lines are fitted normal distributions there for reference, the blue lines are merely reference lines to ease comparisons between years.

```
outH <- histyear(sps,Lbound=0,Rbound=10,inc=0.25,pickvar="Effort",
                 years="Year",varlabel="Effort
(Hrs)",plots=c(4,3),vline=NA)
```

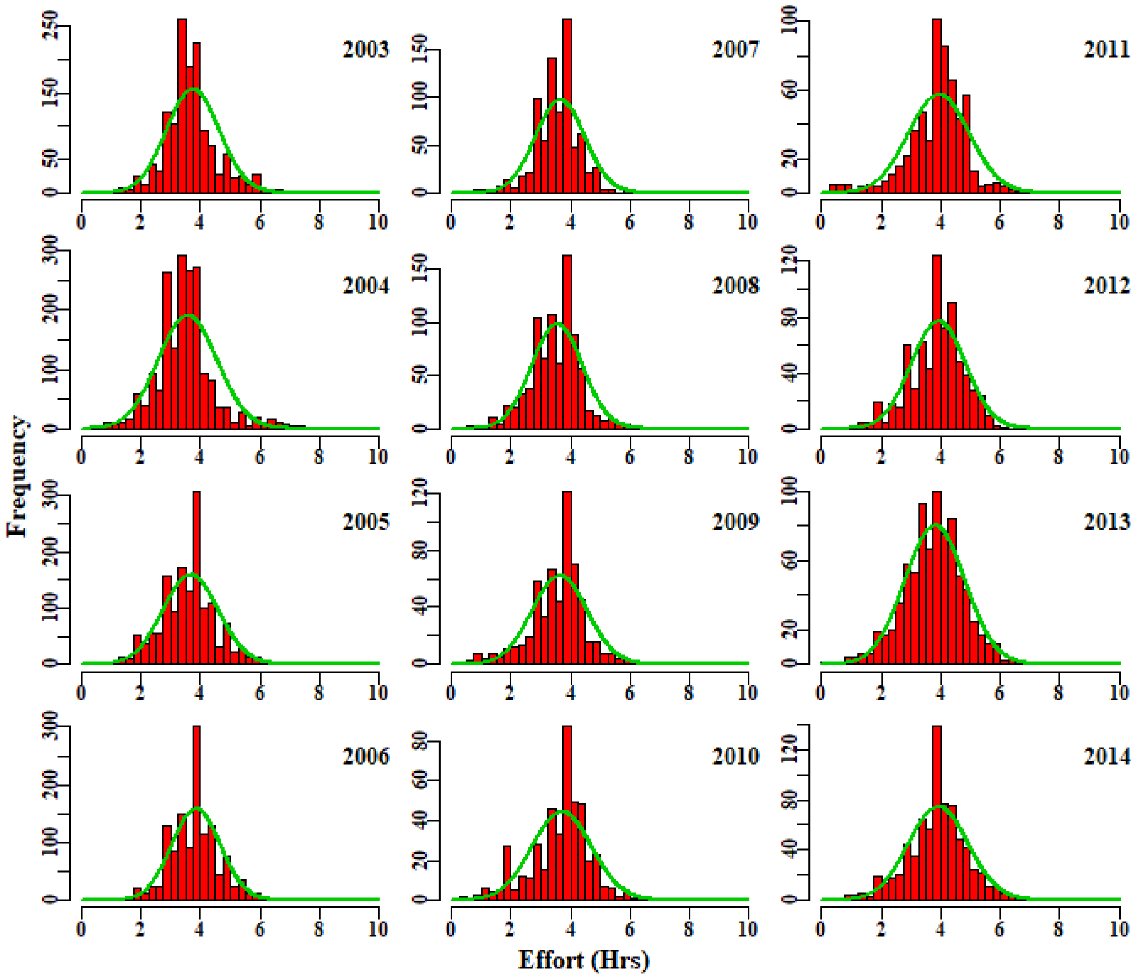


Figure 5. The distribution of reported Effort each year. The green lines are fitted normal distributions there for reference. Note the spikes of reporting four hours.

Spikes can be seen in each of the graphs and the question needs to arise whether this is due to rounding by the fishers or is a real phenomenon. In fact, unless dealing with counts of fish caught (quite possible in some fisheries) then rounding invariably occurs when estimating catches but also in effort.

```
par(mfrow=c(1,1),mai=c(0.45,0.45,0.05,0.05))
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)
plot(sps$Effort,sps$catch_kg,type="p",pch=16,col=rgb(1,0,0,1/5),
     ylim=c(0,500),xlab="Effort (Hrs)",ylab="Catch (Kg)")
abline(h=0.0,col="grey")
```

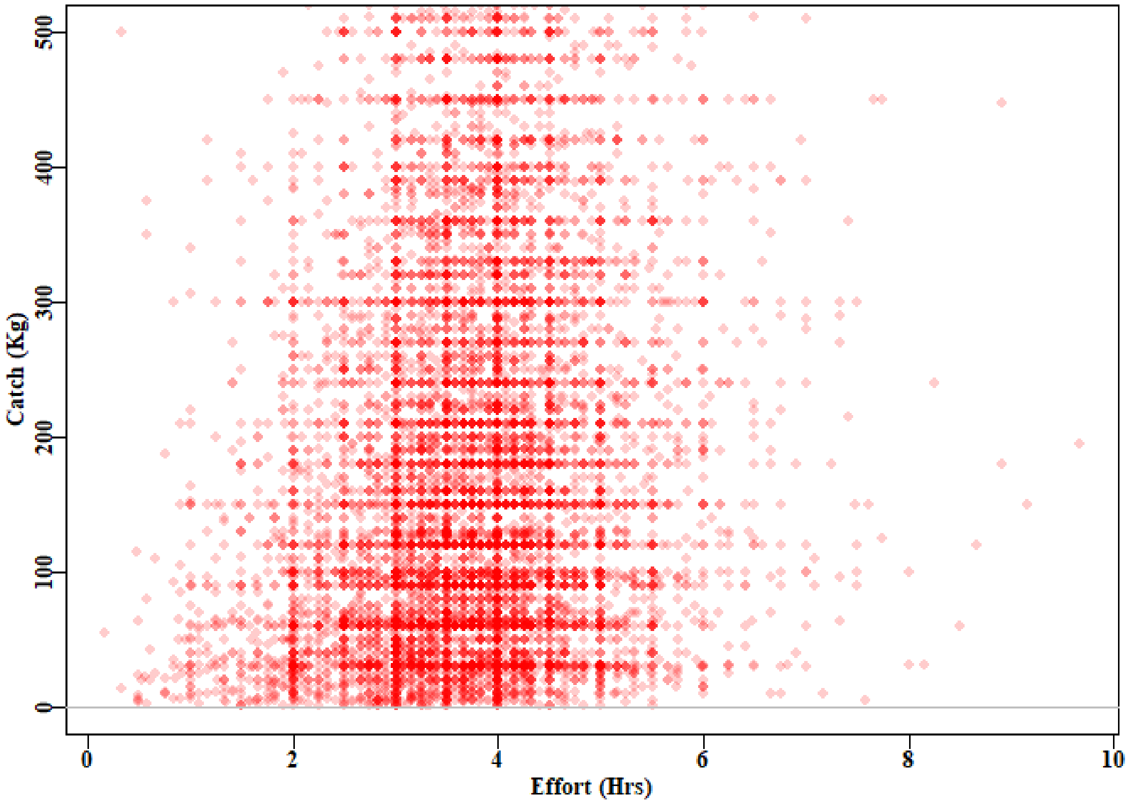


Figure 6. A plot of catch against effort for each record in the *sps* data.frame. The catch axis has been truncated at 500 kg so as to allow the rounding of catches to be less compressed and more visually obvious. It should be clear there is rounding at every half hour between 2 - 6 hours. In addition, there is rounding at about 30 kg steps from 30 - 300 kg, with other categories above that. The 30-33kg rounding reflects a belief that a standard fish bin contains about 30-33Kg of fish.

The uneven grid like nature of the catch and effort data is reflected in the CPUE data, which might make one skeptical about the notion of a predictive model attempting to predict such values. While the residuals that are the basis of th statistical model fitting might be smoother in their distribution they do derive from a comparison of smooth predicted values with the grouped observed values, so any results are likely to be uncertain and to under-estimate any inherent variation.

Despite such problems it is possible to derive useful information from fisheries data. It is generally recognized that fisheries data in general is noisy and potentially contains many errors, especially when considering the less important species that fall into the data-poor category. Neverntheless, the challenge remains of attempting to obtain useful and useable information from analysing such data.

Plotting Sketch Maps of Lat-Long data

Since the advent of GPS and GPS plotters very many fishers use there equipment and fisheries departments have started to ask for precise location data accordingly. If such latitude and longitude data are available it is often informative to plot such data as a literal map to illustrate the focus and range of a fishery. **cede** also provides the capacity to generate such sketch maps instead of using a full GIS. The idea here is not to conduct detailed spatial analyses, for which a GIS is better suited. Instead the idea is simply to gain a rapid impression of the operation of a fishery. Of course, care needs to be taken with such plots as they very obviously contain confidential information (such as exactly where fishers have been operating). This is especially important when there are very few fishers involved in a fishery. So while such images may not be able to be displayed in meetings they remain useful for data exploration purposes.

```
leftlong <- 143.0;  rightlong <- 150.0
uplat <- -40.0;  downlat <- -44.6
plotaus(leftlong,rightlong,uplat,downlat,gridon=1.0)
addpoints(sps,intitle="Location of Catches")

## 11603 1 4500

plotLand(incol="blue")
```

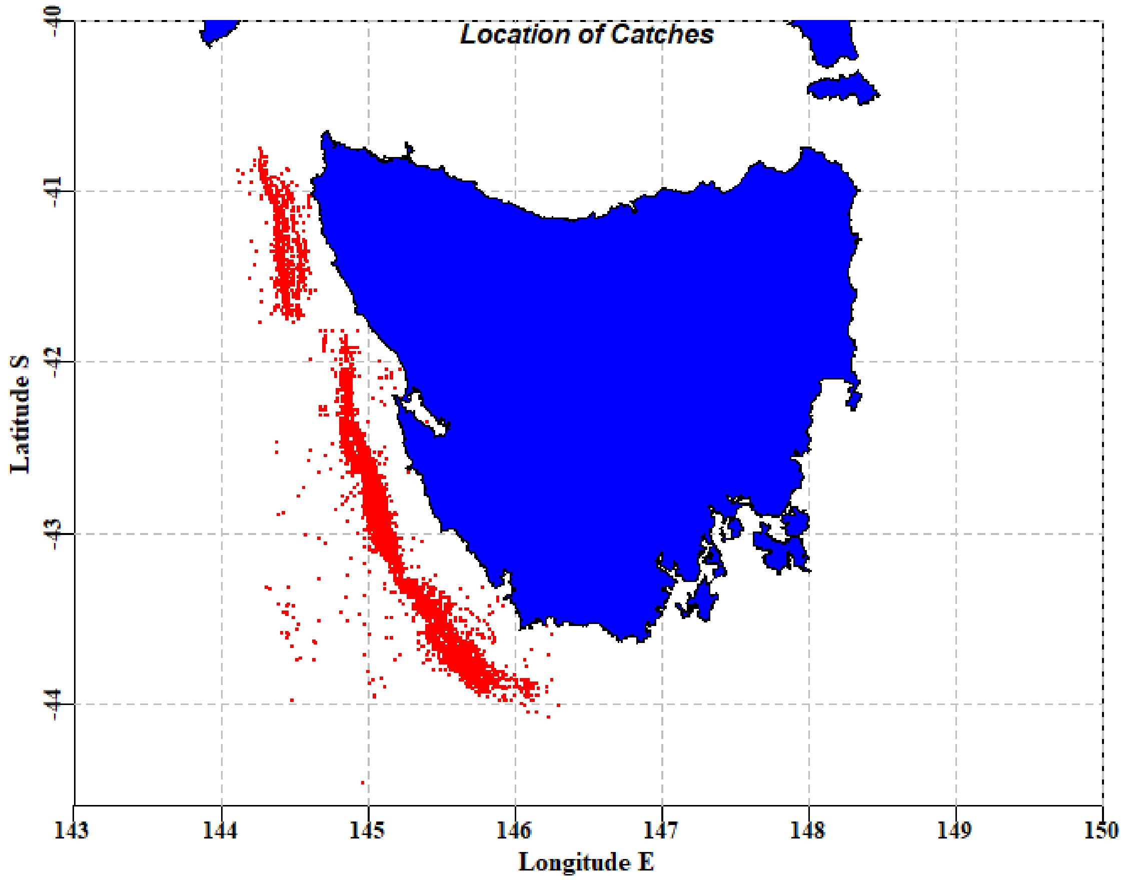


Figure 7. A sketch map of the the Lat Long data within the *sps* data set. There are clearly a number of points reported to be out oer the abyssal plain, but the majority of points define the range of the fishery.

Rather than show individual points it is also possible, by using the function *plotpolys*, to aggragate catches into different geographical sub-divisions (e.g. 0.25 or 0.5 degree squares, definable with the *gridon* parameter). If these are coloured relative to the density of total catches the locations where most of the yield of a fishery derives from becomes apparent. The output, from the function includes the plotting but also the sub-divisions used and the counts of each of those sub-divisions. The final plotting of the land is merely to provide a tidy plot.

```
leftlong <- 143.0;  rightlong <- 150.0
uplat <- -40.0;  downlat <- -44.6
plotaus(leftlong,rightlong,uplat,downlat,gridon=1.0)
plotpolys(sps,leftlong,rightlong,uplat,downlat,gridon=0.2,leg="left",
          intitle="0.2 degree squares",mincount=2)
```

```
## subdiv      87.057 8.7057 0.87057 0.087057
## counthot    0 0 0 0
## 494.8624      500 250 100 50 10 5 1 0.001
## countpoly   0 2 6 6 11 4 16 31
```

```
plotLand(incol="pink")
```

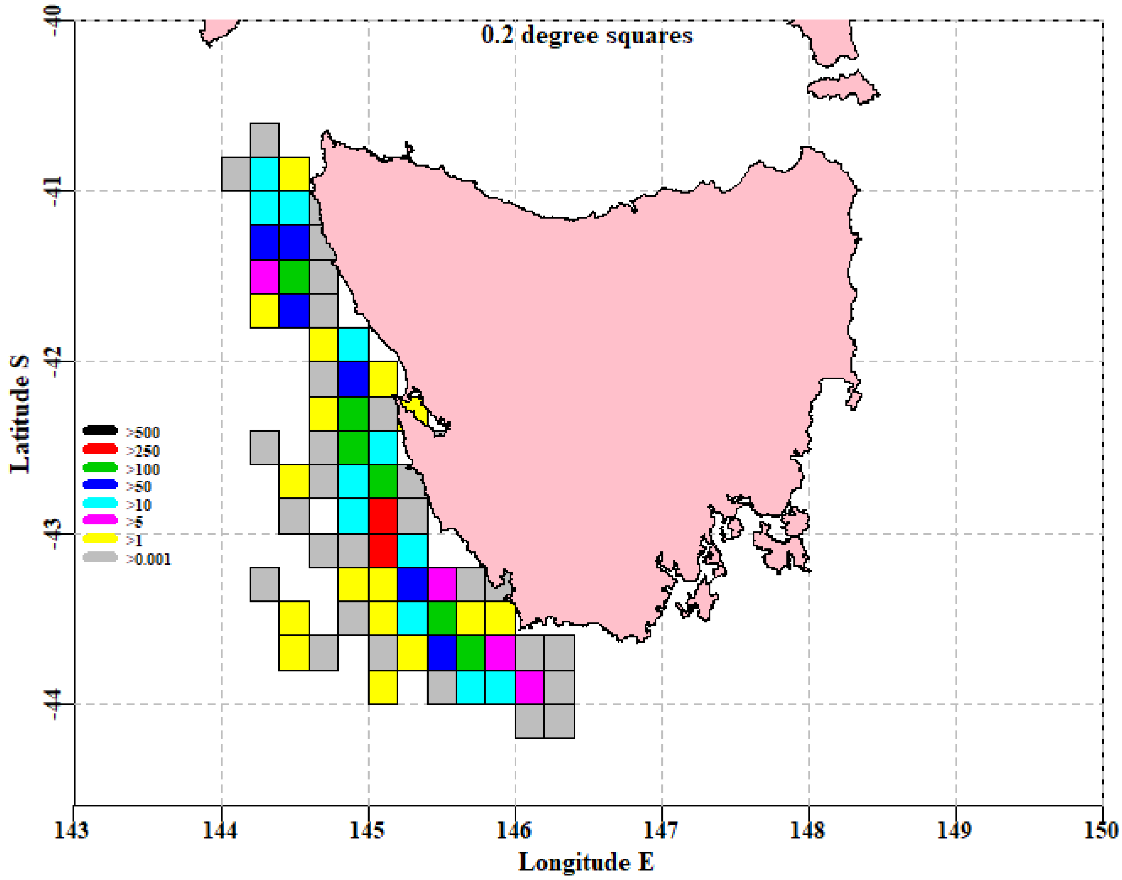


Figure 8. A sketch map of the the Lat Long data within the *sps* data set with catches aggregated into 0.2 degree squares. By requiring at least 2 records in each square before inclusion some of the deeper water extraneous records have been eliminated (although not all). The red, green, and roayl blue squares denote the areas generating the greatest yields.

Such sketch maps can be helpful, especially when plotting single year's of data to illustrate how the extent of a fishery varies through time. There are obvious limitations. There is no formal map projection, one merely alters the width and height of the plot until the visual representation of the land looks acceptable. In addition there are islands missing so as to limit the size of the underlying coastal definition data set (to see this try entering *head(cede:::aus,30)* into the console).

CPUE Standardization

Introduction

If one were to search online for CPUE standardization it would quickly become apparent that this is a very large subject with many alternative approaches and strategies. Here I will introduce two approaches that use General and Generalized Linear Models (LMs and GLMs) and that use Generalized Additive Models (GAMS). This will only be a brief introduction to the subject but the hope is that such an introduction would enable users to explore further and develop approaches best suited to their own fisheries.

Commercial catch and effort (CPUE) data are used in very many fishery stock assessments in Australia as an index of relative abundance. Using CPUE in this way assumes there is a direct relationship between catch rates and exploitable biomass. However, many other factors can influence catch rates, including vessel, gear(fishing method), depth, season, area, and time of fishing (e.g. day or night). The use of CPUE as an index of relative abundance requires the removal of the effects of variation due to changes in factors other than stock biomass, on the assumption that what remains will provide a better estimate of the underlying biomass dynamics. This process of adjusting the time series for the effects of other factors is known as standardization and the accepted way of doing this is to use some statistical modelling procedure that focuses attention onto the annual average catch rates adjusted for the variation in the averages brought about by all the other factors identified. Idiosyncrasies between species and methods across Australia means that each fishery/stock for which standardized catch rates are required entails its own set of conditions and selection of data.

The Limits of Standardization

The use of commercial CPUE as an index of the relative abundance of exploitable biomass can be misleading when there are factors that significantly influence CPUE but cannot be accounted for in a statistical standardization analysis. Over the last few decades the management of many Australian fisheries have undergone significant changes. For example, in the Commonwealth fisheries there was the introduction of the quota management system into the SESSF in 1992, and the introduction of the Harvest Strategy Policy (HSP) and associated structural adjustment in 2005 - 2007. The combination of limited quotas and the HSP is now controlling catches in such a way that many fishers have been altering their fishing behaviour to take into account the availability of quota and their own access to quota needed to land the species taken in the mixed species SESSF.

Methods

Initial Data Selection

Fisheries data is often noisy and can contain obvious errors and mistakes (e.g. an inshore species repotedly being caught in 6000 m of water). The data exploration mentioned earlier should allow one to defensibly select data for further analysis. Often such data selection is aimed at identifying records that represent typical activities in each fishery concerned. In particular some selection criteria are aimed at focussing on records where the species is being targeted. For example, most species have a depth range within which they are typically caught. Ideally, an agreed depth range should be used so that it becomes standard to select data records between some minimum and maximum dapth range. A second example relates to one vessel in the SESSF catching a particular species by a particular gear having catch rates 10 - 20 times those of other vessels fishing in the same places at the same time. Further exploration indicated that the vessel concerned had misunderstood how to fill in the log book so their data was removed from subsequent analysis. Whatever decisions are made about the selection of data, each choice should be defensible and it should be possible to present the evidence for the selection made (e.g. illustrate extreme values, typical depth ranges, unusual vessels).

Once a defensible set of data records have been selected there are other modifications needed. At its most besic a linear model is very similar to a regression analysis. If you imagine conducting a regression of Log(CPUE) against Year so as to evaluate how those catch rates have changed through time then all that would come out would be a single line having two parameters, an intercept and gradient. There are only two parameters because it would treat the factor 'Year' as a continuous variable. What we actually want is a separate index for each year, we need to treat the 'Year' factor as a categorical factor rather than as a continuous variable. Below we will illustrate the use of using all categorical factors and then a different illustration showing how to include a continuous variable such as depth, into a standardization.

Standardization

The use of *properties* indicates that in the *sps* data set contains six clear factors: Year, Month, Vessel, Depth or DepCat, DayNight, and Zone. The Zone factor is a subdivision of mainly the Latitude factor although longitude is also in there to a lesser extent.

First we need to convert some of the factors into categorical factors. For the *sps* data set there are six factor. It is good practice not to over-write your original data.frame so here the *sps* name is slightly modified to *sps1*.

```
kable(properties(sps),digits=c(0,0,0,0,6,6,6))
```

	Index	isNA	Unique	Class	Min	Max	Example
Year	1	0	12	numeric	2003.000000	2014.000000	2004
Month	2	0	12	numeric	1.000000	12.000000	4
Vessel	3	0	23	numeric	1.000000	27.000000	1
catch_kg	4	0	442	numeric	1.000000	4500.000000	220
Long	5	0	447	numeric	144.116670	146.300000	145.1167
Lat	6	0	512	numeric	-45.833330	-40.750000	-43.06667
Depth	7	0	191	numeric	2.000000	366.000000	125
DayNight	8	0	3	character	0.000000	0.000000	N
Effort	9	0	377	numeric	0.160000	9.660000	4
Zone	10	0	3	numeric	1.000000	3.000000	1
CE	11	0	3624	numeric	0.222222	4140.000000	55
LnCE	12	0	3596	numeric	-1.504077	8.328451	4.007333
DepCat	13	0	15	numeric	0.000000	350.000000	125

```
labelM <- c("Year","Zone","Vessel","Month","DayNight","DepCat")
sps1 <- makecategorical(labelM,sps)
kable(properties(sps1),digits=c(0,0,0,0,6,6,6))
```

	Index	isNA	Unique	Class	Min	Max	Example
Year	1	0	12	factor	0.000000	0.000000	2004
Month	2	0	12	factor	0.000000	0.000000	4
Vessel	3	0	23	factor	0.000000	0.000000	1
catch_kg	4	0	442	numeric	1.000000	4500.000000	220
Long	5	0	447	numeric	144.116670	146.300000	145.1167
Lat	6	0	512	numeric	-45.833330	-40.750000	-43.06667
Depth	7	0	191	numeric	2.000000	366.000000	125
DayNight	8	0	3	factor	0.000000	0.000000	N
Effort	9	0	377	numeric	0.160000	9.660000	4
Zone	10	0	3	factor	0.000000	0.000000	1
CE	11	0	3624	numeric	0.222222	4140.000000	55

LnCE	12	0	3596	numeric	-1.504077	8.328451	4.007333
DepCat	13	0	15	factor	0.000000	0.000000	125

Note that after using *makecategorical*, the factors of interest within *sps1* are now listed as factors rather than numeric, and that is enough to alter the analysis to something more like an anova than a regression analysis so that we obtain a parameter for each level of the factors used.

```
labelM <- c("Year","Zone","Vessel","Month","DayNight","DepCat")
sps1 <- makecategorical(labelM,sps)
mod <- makeonemodel(labelM)
mod

## LnCE ~ Year + Zone + Vessel + Month + DayNight + DepCat
## <environment: 0x0000000010ae1da0>

class(mod)

## [1] "formula"
```

Each of the standardization methods we will use require that each statistical model to be examined needs to be a **formula**. If you enter *makeonemodel*, without brackets, into the R console you will see the final `form <- as.formula(form)`, which achieves this requirement.

If we are going to use a simple linear model then we can proceed using the function *dosingle* (try ?dosingle or just dosingle). We point the output of this function to the *out* object because there is an enormous amount of information generated, you can see this by using just *str(out)*.

```
labelM <- c("Year","Zone","Vessel","Month","DayNight","DepCat")
sps1 <- makecategorical(labelM,sps)
mod <- makeonemodel(labelM)
out <- dosingle(mod,sps1)
str(out,max.level=1)

## List of 7
##  $ Results  : num [1:12, 1:2] 0.855 1.351 1.26 1.077 0.949 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ StErr    : num [1:12, 1:2] 0 0.0377 0.0399 0.0413 0.0473 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ Optimum  : num 2
##  $ modelcoef: num [1:63, 1:4] 3.8949 0.3697 0.1962 -0.0137 -0.2159
...
##   ..- attr(*, "dimnames")=List of 2
##  $ optModel :List of 13
##   ..- attr(*, "class")= chr "lm"
##  $ modelG   :List of 13
##   ..- attr(*, "class")= chr "lm"
##  $ years    : Factor w/ 12 levels "2003","2004",..: 1 2 3 4 5 6 7 8
9 10 ...
```

One of the components of the *out* object is the *optModel*, which, not surprisingly, represents the optimum model. It is possible to run the generic functions *summary* and *anova*. The *summary* function (`summary(out)`) will generate the parameters (on the log-scale) and a few other details. the *anova* function determines the significance of each factor.

```r
anova(out$optModel)
```

```
## Analysis of Variance Table
##
## Response: LnCE
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## Year         11   371.6   33.78  35.072 < 2.2e-16
## Zone          2   809.2  404.58 420.019 < 2.2e-16
## Vessel       22   374.6   17.03  17.675 < 2.2e-16
## Month        11   281.3   25.57  26.546 < 2.2e-16
## DayNight      2   223.4  111.69 115.950 < 2.2e-16
## DepCat       14   432.7   30.91  32.087 < 2.2e-16
## Residuals 11540 11115.7    0.96
```

The Mean Year Estimates

For the lognormal model the expected back-transformed year effect involves a bias-correction to account for the log-normality; this then focuses on the mean of the distribution rather than the median:

$$CPUE_t = e^{(\gamma_t + \sigma_t^2/2)}$$

where γ_t is the Year coefficient for year t and σ_t is the standard deviation of the log transformed data (obtained from the analysis). The year coefficients were all divided by the average of all the Year coefficients to simplify the visual comparison of catch rate changes.

$$CE_t = \frac{CPUE_t}{(\sum CPUE_t)/n}$$

where $CPUE_t$ is the yearly coefficients from the standardization, $(CPUE_t)/n$ is the arithmetic average of the yearly coefficients, n is the number of years of observations, and CE_t is the final time series of yearly index of relative abundance.

All of this can be obtained in two ways. Within the *out* object there is the *Results* matrix which contains both the geometric mean estimates along with the optimum statistical model. *StErr* within *out* contains the standard error estimates for each of those.

```r
cbind(out$Results,out$StErr)
```

```
##           Year   optimum       Year    optimum
## 2003 0.8551563 1.0803147 0.00000000 0.00000000
## 2004 1.3506682 1.5644932 0.03774809 0.03629678
## 2005 1.2600506 1.3154374 0.03988160 0.03897411
## 2006 1.0769724 1.0665040 0.04131560 0.04084128
## 2007 0.9487208 0.8715249 0.04731264 0.04691162
## 2008 0.8429911 0.8105254 0.04670561 0.04604055
## 2009 0.8422759 0.8031273 0.05292345 0.05183217
## 2010 0.8511003 0.8000413 0.05818629 0.05697959
## 2011 0.8379600 0.7449298 0.05251125 0.05168512
## 2012 1.0175412 0.9594120 0.04949327 0.04872046
## 2013 0.9505466 0.9162404 0.04723900 0.04669282
## 2014 1.1660166 1.0674496 0.04849594 0.04877541
```

Alternatively, if all the details are wanted there is another function *getfact*, which provides those extra details.

```r
kable(getfact(out$optModel,"Year"),digits=c(4,4,4,4,4,4))
```

	Coeff	SE	LogCE	Scaled	t value	Prob
Year	1.0000	0.0000	0.0000	1.0803		
Year2004	1.4482	0.0363	0.3697	1.5645	10.1841	0.0000
Year2005	1.2176	0.0390	0.1962	1.3154	5.0330	0.0000
Year2006	0.9872	0.0408	-0.0137	1.0665	-0.3355	0.7373
Year2007	0.8067	0.0469	-0.2159	0.8715	-4.6015	0.0000
Year2008	0.7503	0.0460	-0.2884	0.8105	-6.2637	0.0000
Year2009	0.7434	0.0518	-0.2978	0.8031	-5.7462	0.0000
Year2010	0.7406	0.0570	-0.3020	0.8000	-5.2996	0.0000
Year2011	0.6895	0.0517	-0.3731	0.7449	-7.2178	0.0000
Year2012	0.8881	0.0487	-0.1199	0.9594	-2.4604	0.0139
Year2013	0.8481	0.0467	-0.1658	0.9162	-3.5513	0.0004
Year2014	0.9881	0.0488	-0.0132	1.0674	-0.2700	0.7872

The standardizations provide parameters for each level of each factor except the first level in each case. These are all assumed to have a log-trasformed value of 0.0 (= 1.0 on the nominal scale). All the other parameters (when log-normal errors are used, are proportional to the first level). Thus the LogCE column is the output from the standardization. The bias-adjusted transformation back to the nominal scale is described in the equations above. The 'Scaled' column is the same as the 'Coeff' column except it is has been divided through by the mean of the series. This sets the average value to 1.0, which permits simple visual comparison with other time-series. The 'SE' column provides the basis for generating the log-normally distributed confidence intervals.

```r
# plotprep(width=7,height=4.5)
plotstand(out,bars=TRUE)
```

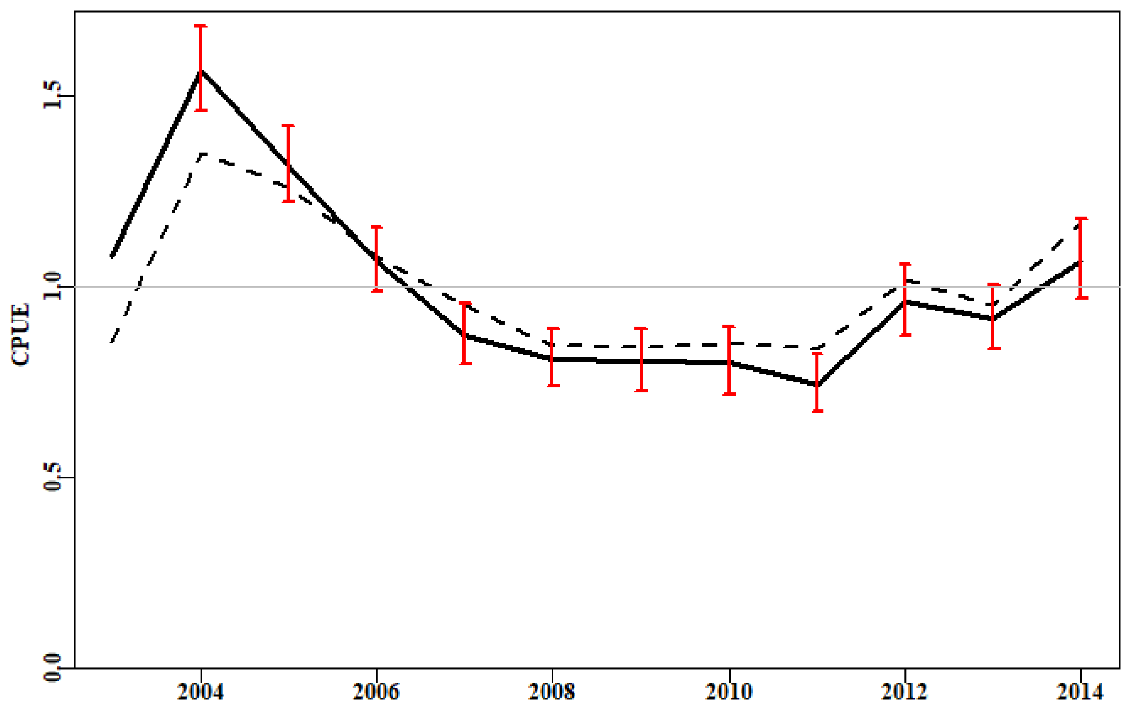


Figure 9. The standardization of the cpue data within the *sps* data set. The dashed line is the geometric mean CPUE while the solid line with 95% confidence intervals is the

standardized CPUE. In places the difference between the standardized CPUE and the geometric mean CPUE is greater than the 95% log-normal confidence intervals.

One issue with this plot is that the scale makes little sense to Industry members who are more used to the nominal scale at which they operate personally. Given that the average of both the geometric mean and the optimum model is 1.0, both can be multiplied by a constant to rescale the plots. If we calculate the geoemtric mean CPUE for the whole fishery we can use that as a multiplier and that will place each time-series on a recognizable nominal scale. This can be done using the function *geomean* and include the *geo* option of *plotstand*.

```
# plotprep(width=7,height=4.5)
geom <-geomean(sps$CE)
plotstand(out,bars=TRUE,geo=geom)
```

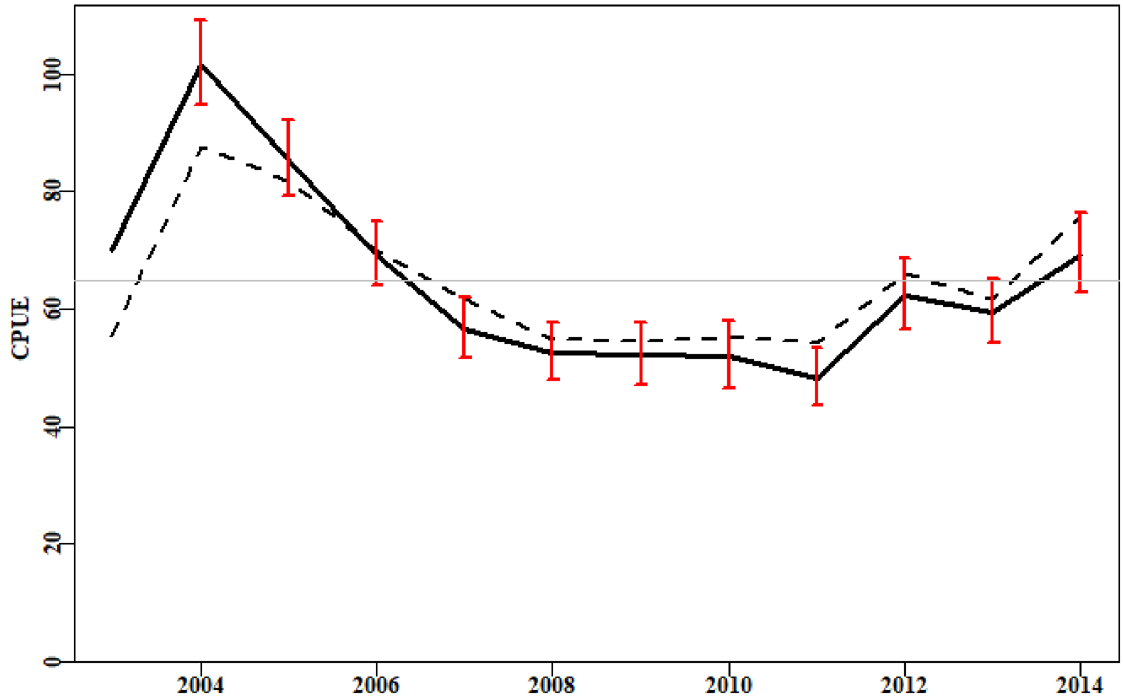


Figure 10. The standardization of the cpue data within the *sps* data set. The dashed line is the geometric mean CPUE while the solid line with 95%. These trajectories both have an average of the overall geometric mean CPUE.

It is often helpful to examine the standardizations as they increase in complexity so that the relative influence of each factor becomes more clear. However, to do this would require a little more R code.

```
# first make a matrix to hold the results
labelM <- c("Year","Zone","Vessel","Month","DayNight","DepCat")
columns <- c("adjR2","incR2","RSS","MSS","Npar","nobs","AIC")
nummod <- length(labelM)
results <- as.data.frame(matrix(0,nrow=nummod,ncol=length(columns),
                         dimnames=list(labelM,columns)))
for (i in 1:nummod) {  # sequentially build the models
   mod <- makeonemodel(labelM[1:i])  # When i = 1 LnCE ~ Year
   out <- dosingle(mod,sps1)
   outsum <- summary(out$optModel)
   aov <- anova(out$optModel)    #  Extract a range of results
```

```
    RSS <- tail(aov$"Sum Sq",1)
    df <- aov$Df
    nobs <- sum(df) + 1
    numfact <- length(df) - 1
    npars <- sum(df[1:numfact]) + 1
    AIC <- nobs * log(RSS/nobs) + (2 * npars)
    results[i,] <- c(outsum$adj.r.squared,NA,RSS,sum(aov$"Sum Sq") -
RSS,npars,nobs,AIC)
}
results[2:nummod,"incR2"] <- results[2:nummod,"adjR2"]-
results[1:(nummod-1),"adjR2"]
round(results,4)
```

```
##              adjR2  incR2      RSS       MSS Npar  nobs       AIC
## Year       0.0264     NA 13236.77  371.6062   12 11603 1552.5217
## Zone       0.0857 0.0594 12427.62 1180.7589   14 11603  824.6352
## Vessel     0.1116 0.0259 12053.06 1555.3196   36 11603  513.5497
## Month      0.1315 0.0199 11771.79 1836.5924   47 11603  261.5701
## DayNight   0.1478 0.0163 11548.41 2059.9662   49 11603   43.2834
## DepCat     0.1788 0.0309 11115.70 2492.6736   63 11603 -371.8236
```

By looking at the increments to the adjusted-R2 it is clear that the factor *DepCat* has a larger impact on the variation accounted for than even *Vessel*, so strictly the analysis should be repeated after re-ordering the different factors within labelM. The *AIC* column identifies the optimum combination of factors with the smallest value indicating the optimum. It would be worthwhile repeating the analysis with the re-ordering. Typically, if one plots each standardization on the same plot, typically, while the later factors can be statistically significant, their effect upon the trajectory of the standardized CPUE can be minimal or appear to contribute mainly noise. If the standardization is to be used within an assessment it is the trend that matters so those final few factors may only have a minor effect.

Alternative Standardization Strategies

So far we have only considered General Linear Models (which with log-normal errors give the same results as simple linear models). If we wish to use alternative residual error structures then it would be necessary to use true GLMs (as in Generalized Linear Models). These would be necessary if, for example, there was a wish to attempt using perhaps a Gamma distribution instead of log-normal, then would need to use different syntax. The standard approach when using the Gamma distribution would be to use a log-link in the GLM. In such cases then the dependent variable would then be *CE* rather than *LnCE*. The functions described so far are designed for use with log-normal residual errors that need a bias-correction Gamma residual erros do not require such a bias-correction so we will need to work directly with the estimated coefficients.

```
labelM <-
c("Year","Zone","Vessel","Month","DayNight","DepCat","Month:Zone")
sps1 <- makecategorical(labelM,sps)
mod <- makeonemodel(labelM,dependent="CE")

model4 <- glm(mod,family=Gamma(link="log"),data=sps1)
m4 <- summary(model4)$coefficients  # combine these with empty first
year
yrval <- rbind(c(0,0,0,0),m4[grep("Year",rownames(m4)),])
gamres <- cbind(yrval,exp(yrval[,"Estimate"]))
```

```
rownames(gamres) <- 2003:2014
gamres

##          Estimate Std. Error     t value     Pr(>|t|)
## 2003  0.000000000 0.00000000  0.00000000 0.000000e+00 1.0000000
## 2004  0.333489137 0.04141384  8.05260025 8.901627e-16 1.3958299
## 2005  0.232479826 0.04412451  5.26872275 1.398554e-07 1.2617250
## 2006  0.003738450 0.04620812  0.08090461 9.355192e-01 1.0037454
## 2007 -0.123979553 0.05282090 -2.34716857 1.893353e-02 0.8833979
## 2008 -0.139742192 0.05204407 -2.68507408 7.261757e-03 0.8695824
## 2009 -0.226946891 0.05850235 -3.87927840 1.053499e-04 0.7969631
## 2010 -0.079275668 0.06427291 -1.23342264 2.174433e-01 0.9237852
## 2011 -0.147106194 0.05832505 -2.52217851 1.167642e-02 0.8632023
## 2012 -0.080927349 0.05495370 -1.47264600 1.408738e-01 0.9222607
## 2013 -0.002500419 0.05314454 -0.04704940 9.624747e-01 0.9975027
## 2014  0.023195520 0.05528536  0.41955987 6.748148e-01 1.0234666

plotstand(out,bars=TRUE)
lines(2003:2014,exp(yrval[,"Estimate"]),lwd=2,col=4)
```

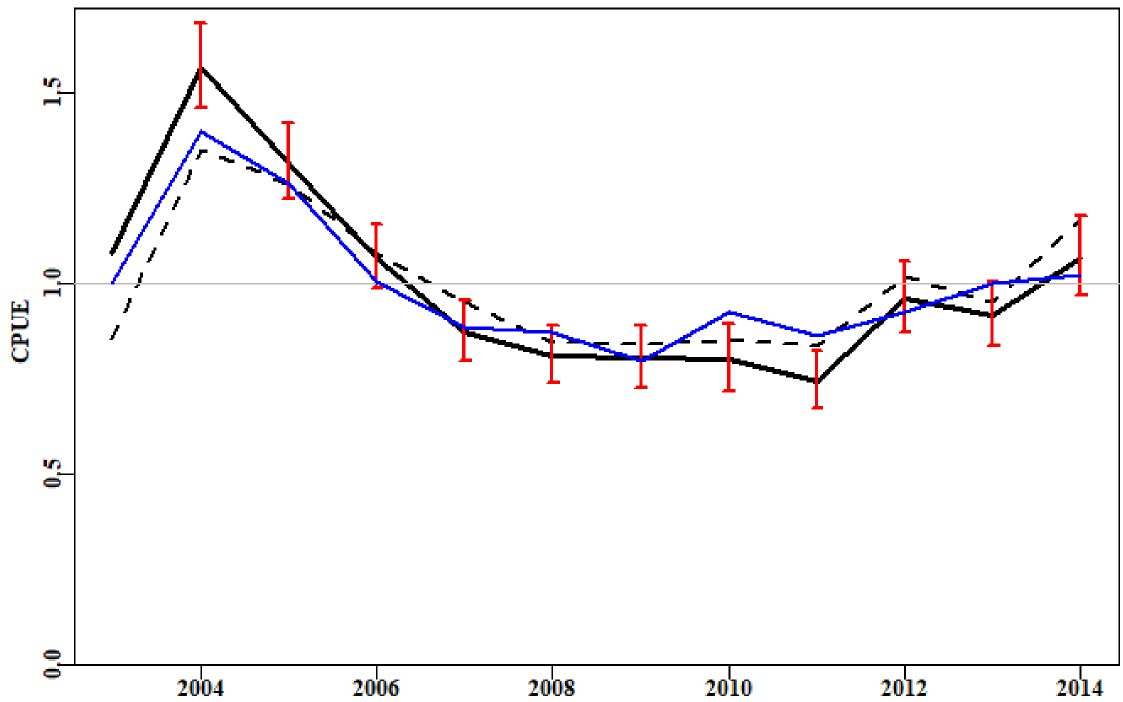


Figure 11. The standardization of the cpue data within the *sps* data set comparing a LM using log-normal with a GLM using Gamma residual errors. The dashed line is the geometric mean CPUE while the solid black line with 95% is the log-normal error standardization. Finally, the blue line is the Gamma error standaridzation.

The Use of GAMs

Generalized Additive Models are an extension of GLMs in which at least some of the factors are replaced by fitting smooth surfaces to some of the factors that are considered to have a non-linear relationship with catch rates.

In order to run them, however, it is necessary to install a number of additional R packages. As an example, we could use a GAM to add a smoother to the Lat - Long data in the sps data set. We would actually use the sps1 data set as the remaining categorical factors are also included in the analysis. A possible workflow might involve the following code.

```
# install and call these R packages and their dependencies
library(nlme)
library(mgcv)
library(gamm4)
# note the use of gam rather than lm or glm (see the examples in ?gam
for more
# details.
modelGam <- gam(LnCE ~ s(Long,Lat) + Year + Zone + Vessel + Month +
                   DayNight + DepCat, data = sps1)
anova(modelGam)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## LnCE ~ s(Long, Lat) + Year + Zone + Vessel + Month + DayNight +
##     DepCat
##
## Parametric Terms:
##           df       F p-value
## Year      11  54.145  <2e-16
## Zone       2   0.778   0.459
## Vessel    22  16.549  <2e-16
## Month     11  26.175  <2e-16
## DayNight   2 112.842  <2e-16
## DepCat    14   9.269  <2e-16
##
## Approximate significance of smooth terms:
##                edf Ref.df     F p-value
## s(Long,Lat) 26.96  28.69 19.14  <2e-16
```

We should not be surprised that the *Zone* factor is no longer singificant. By including the Lat - Long surface including the *Zone* factor becomes redundant so we should really repeat the analysis without *Zone* included.

```
modelGam <- gam(LnCE ~ s(Long,Lat) + Year + Vessel + Month +
                   DayNight + DepCat, data = sps1)
anova(modelGam)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## LnCE ~ s(Long, Lat) + Year + Vessel + Month + DayNight + DepCat
##
## Parametric Terms:
##           df       F p-value
## Year      11  54.039  <2e-16
## Vessel    22  16.635  <2e-16
## Month     11  26.237  <2e-16
## DayNight   2 112.859  <2e-16
```

```
## DepCat   14   9.297  <2e-16
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(Long,Lat) 27.14  28.74 26.65  <2e-16
```

We can use the *getfact* function to extract the results we need. The Coeff column contains the LogCE transformed back to the linear scale and Scaled is the Coeff re-scaled to a mean of 1.0. Once again if it is desired to scale this to the nominal CPUE from the fishery so as to improve communication with Industry and managers then we can use *geomean* to estimate the overal geometric mean to re-scale the 'Scaled' column to something more meaningful to industry members.

```
answer <- getfact(modelGam,"Year")
opti <- answer[,"Scaled"]
round(answer,5)

##              Coeff      SE    LogCE  Scaled  t value    Prob
## Year       1.00000 0.00000  0.00000 1.11779       NA      NA
## Year2004 1.43698 0.03557  0.36191 1.60624 10.17322 0.00000
## Year2005 1.18326 0.03826  0.16754 1.32263  4.37886 0.00001
## Year2006 0.94500 0.04009 -0.05737 1.05631 -1.43110 0.15243
## Year2007 0.76003 0.04616 -0.27546 0.84955 -5.96784 0.00000
## Year2008 0.72439 0.04523 -0.32344 0.80972 -7.15074 0.00000
## Year2009 0.70430 0.05095 -0.35185 0.78725 -6.90523 0.00000
## Year2010 0.69154 0.05594 -0.37039 0.77300 -6.62073 0.00000
## Year2011 0.66716 0.05075 -0.40601 0.74575 -7.99956 0.00000
## Year2012 0.85665 0.04794 -0.15587 0.95755 -3.25124 0.00115
## Year2013 0.82559 0.04588 -0.19271 0.92283 -4.19995 0.00003
## Year2014 0.94059 0.04789 -0.06240 1.05138 -1.30282 0.19266
```

We can gain an impression of the surface fitted to the Lat - Long data using the *plot* function, which recognizes the output from a gam and can react accordingly.

```
#plotprep(width=4.5,height=7)
 plot(modelGam,ylim=c(-44.5,-
40),xlim=c(143.5,146.5),se=FALSE,xlab="",ylab="")
 title(ylab=list("Latitude", cex=1.0, font=7),
       xlab=list("Longitude", cex=1.0, font=7))
 plotLand("pink")
```

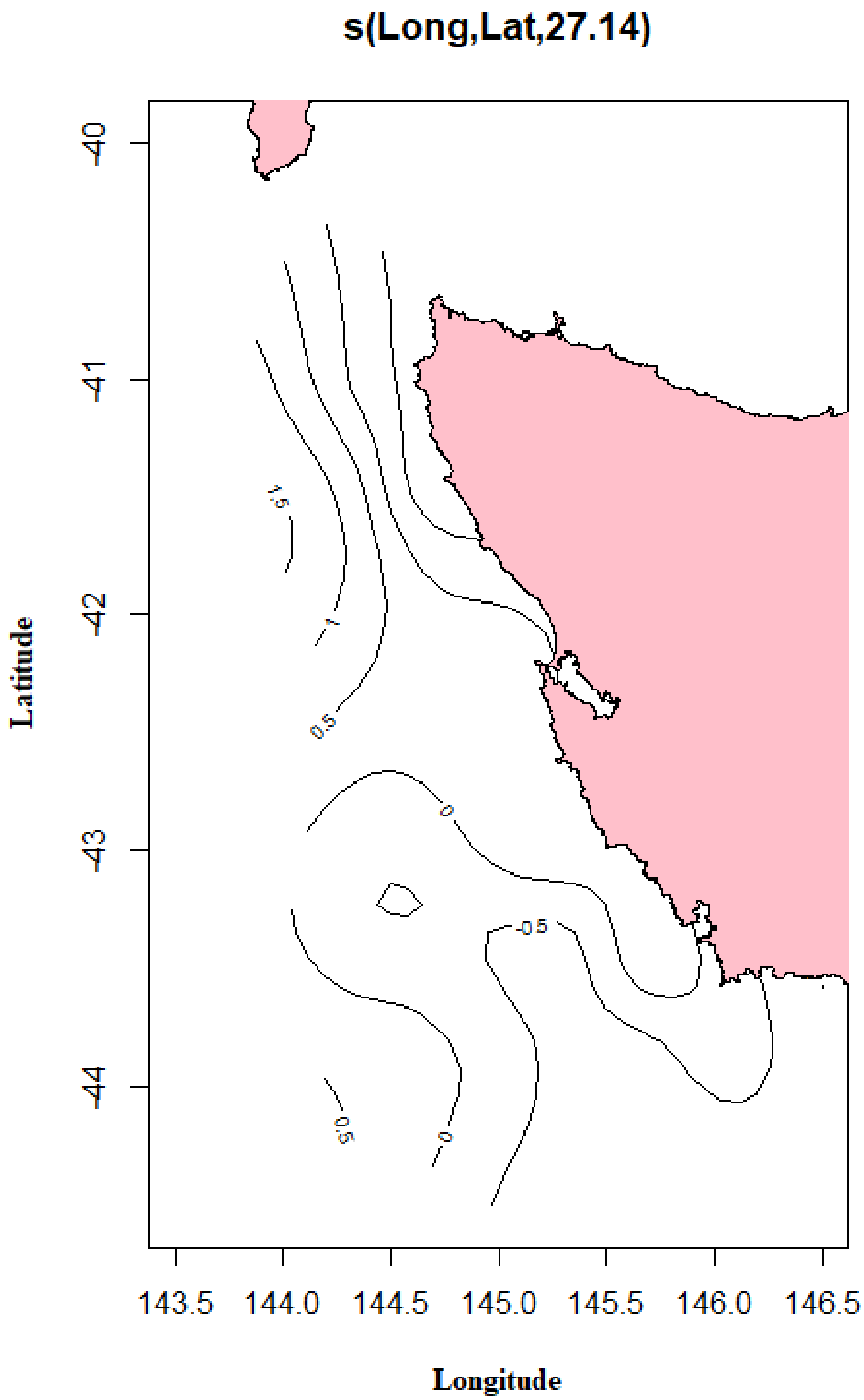
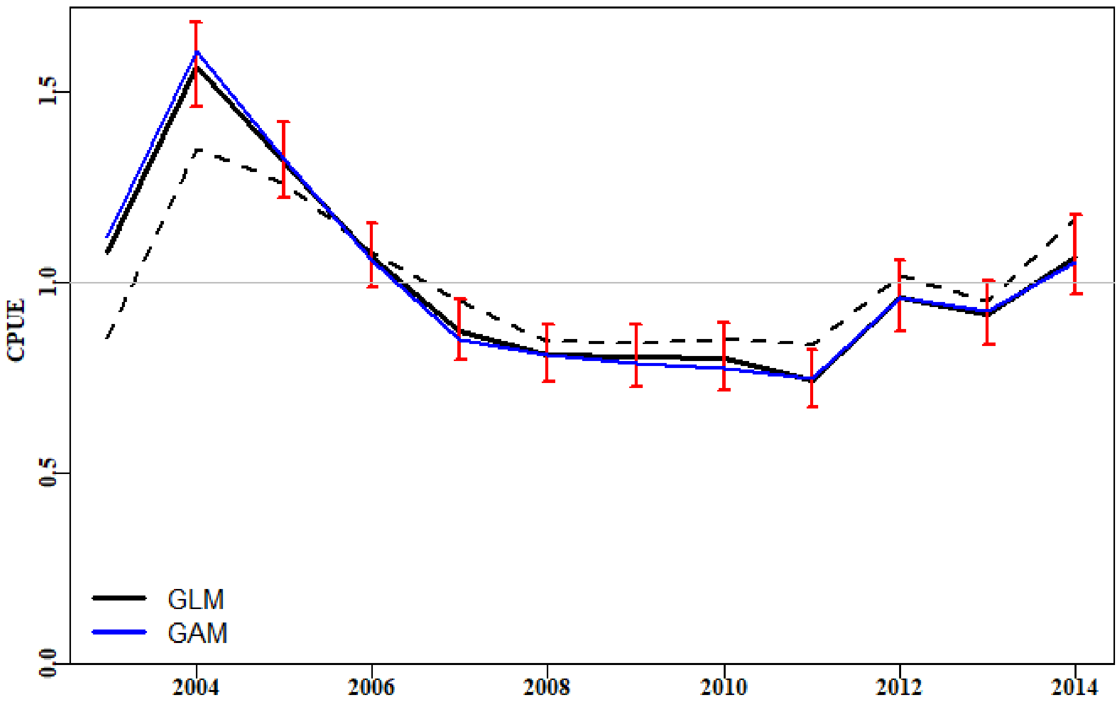


Figure 12. A plot of the surface fitted to the output from the gam function.

The effect on the year parameters is what we are really interested in for the purposes of stock assessment and we can compare the outcome of the GAM with the previous GLM.

```
#plotprep(width=7,height=4.5)
plotstand(out,bars=TRUE)
lines(facttonum(out$years),opti,col=4,lwd=2)
legend("bottomleft",c("GLM","GAM"),col=c(1,4),lwd=3,bty="n",cex=1.2)
```


CPUE
0.0
0.5
1.0
1.5
2004
2006
2008
2010
2012
2014
GLM
GAM