# kobe: **R** tools for Tuna Management Advice

## Laurence Kell
ICCAT

---

### Abstract

Scientific advice within the tuna Regional Fisheries Management Organisations (tRF-MOs) is based the common Kobe framework. Advice summarises the probabilities of biomass being greater than and fishing mortality less than the levels that can support maximum sustainable yield (MSY) for a range of management options. This requires estimates of current status relative to reference points and projections for the different management options. The `kobe` package can read in results from a variety of stock assessment packages and summarise them in the Kobe format.

*Keywords*: R, tuna, RFMO, advice, fisheries, stock assessment, management.

---

# Contents

# 1. Introduction

Scientific Advice within the tuna Regional Fisheries Management Organisations (tRFMOs) is based on the Kobe advice framework. Advice is based on ensuring that they is a low risk of fishing mortality exceeding $F_{MSY}$ and biomass falling below $B_{MSY}$. This requires the calculation of the probabilities of $F < F_{MSY}$ and $B > B_{MSY}$ for current stock status and a range of management options, generally total allowable catches (TAC).

This requires, a stock assessment, estimates of reference points and stock projections, which can be performed in a variety of software packages. The `kobe` package reads in results from the different input and output file formats and summarises them in the Kobe advice format.

The package provides methods for creating data.frames holding time series of $F/F_{MSY}$ and $SSB/B_{MSY}$ and for summarising, plotting and tablulation using using the **plyr**, **ggplot2** and **tabular** packages.

# 2. Data

Estimates of stock status and exploitation level relative to MSY reference points can be obtained from a variety of stock assessment methods based on a variety of assumptions. For example biomass dynamic models which provide estimates of total biomass and harvest rate or age based models which provide estimates of spawning stock biomass and fishing mortality.

The different stock assessment methods have mainly been implemented as executable programs with a variety of input and output files in text format. This can make it difficult to

summarise results across methods, evaluate the consequences of the different assumption and to formulate advice in a common framework. Therefore in the **kobe** there are methods to read text files into a data.frames summarising stock (biomass or SSB) and harvest (fishing mortality or harvest rate) by scenarios (e.g. assessment run), management option (e.g. TACs) used in projecions and replicates from bootstraps or Monte Carlo Markov Chain simulations.

Currently there are methods for reading in stock assessment text files for ASPIC, Adapt, SS3 and MFCL (i.e. `kobeAspic`, `kobe2box`, `kobeSS3` and `kobeMFCL`).

Here we demonstrate how to read in outputs from `ASPIC`.

### 2.1. kobeAspic

Results from `ASPIC` are written to files with extensions that identify their contents, i.e. bootstrapped assessment results are found in `.bio` and projections based on these in `.prb` files. These is a `.prb` file for each TAC, so to use `kobeAspic` requires specifying a single `.bio` file, multiple `.prb` files and a directory where they are found.

Reading in the bootstrapped assessment

```
> ### Results from ASPIC bootstrapped assessment
> bio   ="http://www.iccat.int/stocka/Models/ASPIC/albs/2011/run2/aspic.bio"
> assmt =kobeAspic(bio)

> head(assmt)

  iter year    stock     harvest      bmsy      fmsy
1    1 1956 1.800000 0.0004169473 110448.7 0.2479588
2    1 1957 1.873086 0.0139635494 110448.7 0.2479588
3    1 1958 1.915449 0.0198786446 110448.7 0.2479588
4    1 1959 1.940032 0.0891123383 110448.7 0.2479588
5    1 1960 1.928930 0.2009482466 110448.7 0.2479588
6    1 1961 1.880085 0.2110163113 110448.7 0.2479588
```

and a projection for a single TAC

```
> ## Results from an ASPIC Projection
> prb   ="http://www.iccat.int/stocka/Models/ASPIC/albs/2011/run2/aspic_15000.prb"
> prj1  =kobeAspic(bio,prb)

> tail(prj1)

      iter year     stock   harvest     bmsy      fmsy
71151  500 2020 0.7997950 0.7527672 206883.8 0.1173888
71161  500 2021 0.8408805 0.7162019 206883.8 0.1173888
71171  500 2022 0.8835242 0.6819915 206883.8 0.1173888
71181  500 2023 0.9273448 0.6502398 206883.8 0.1173888
71191  500 2024 0.9719148 0.6209882 206883.8 0.1173888
71201  500 2025 1.0167761 0.0000000 206883.8 0.1173888
```

By default `kobeAspic` returns all the simulations, however in addition four types of summary output can also be returned; these are specified by the `what` argument. As well as `sim` (the default of everthing), options are `trks` interquartiles and medians of stock and harvest; `pts` bootstrapped values in last year of assessment; `smry` probability of being in kobe phase plot quadrants and `wrms` randomly selected bootstrap runs.

If a single value is passed via `what` then a data.frame is returned, if more than one then list of data.frames is returned; useful when producing different plots and summaries.

To read in the results from projections for a range of TACs first create a vector of files

```
> ## Projections
> TACs=seq(15000,35000,5000)
> prb ="http://www.iccat.int/stocka/Models/ASPIC/albs/2011/run2/aspic_"
> prb =paste(prb,TACs,".prb",sep="")
```

In this case it is better to return subsets of the data for plotting

```
> ## Results
> prj=kobeAspic(bio,prb,what=c("pts","trks","smry"))

> class(prj)

[1] "list"

> names(prj)

[1] "pts"  "trks" "smry"

> ## add TAC column to data.frame
> prj=llply(prj, transform, TAC=TACs[X1])
```

`pts` are the simulation results from the last year in the assessment; `trks` summarise the time series by providing the medians and interquartiles.

```
> head(prj$trks)

  X1 year Percentile    stock      harvest   TAC
1  1 1956        75% 1.800002 0.0004401433 15000
2  1 1957        75% 1.899722 0.0148261963 15000
3  1 1958        75% 1.944386 0.0211279973 15000
4  1 1959        75% 1.963508 0.0944787954 15000
5  1 1960        75% 1.937559 0.2118392354 15000
6  1 1961        75% 1.879321 0.2214004342 15000
```

and `smry` provides the probabilities of being overfished or subject to overfishing.

```
> head(prj$smry)
```

```
  X1 year    stock        harvest red yellow green overFished overFishing   TAC
1  1 1956 1.800000 0.000415844   0      0     1          0           0 15000
2  1 1957 1.873657 0.013922865   0      0     1          0           0 15000
3  1 1958 1.916143 0.019820714   0      0     1          0           0 15000
4  1 1959 1.940661 0.088869350   0      0     1          0           0 15000
5  1 1960 1.929268 0.200428382   0      0     1          0           0 15000
6  1 1961 1.875850 0.210521265   0      0     1          0           0 15000
```

There is also an example data set in `kobe` e.g.

```
> data(sims)
```

for use in the following examples

# 3. Plotting

**kobe** `ggplot2` for plotting. This ia a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics allowing the basic plots to be easily modified.

## 3.1. ggplot2

There are two interfaces `qplot` e.g.

```
qplot(x=year,y=stock,data=prj$wrms)
```

or `ggplot2` e.g.

```
> ggplot(assmt)                                        +
+   geom_hline(aes(yintercept=1),col="red",size=2)     +
+   geom_line( aes(year,stock,group=iter,col=iter))    +
+   theme(legend.position="none")
```
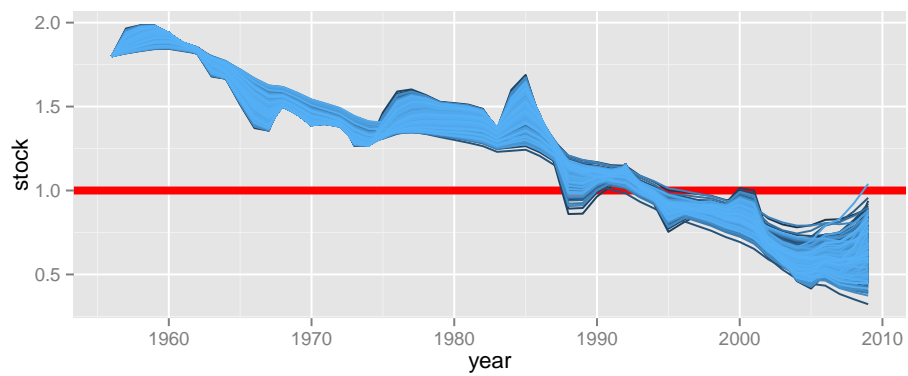


Figure 1: Plots of time series of biomass by bootstrap

The syntax of `qplot` is similar to `plot` in base graphi cs and so it is initially easier to use. While `ggplot` provides more flexibility, where as in the above example, extra elements can be added to the basic plots.

For example here we plot the historic and projected times series of biomass and harvest by TAC for the interquartiles and medians in `trks`

```
> ### tracks
> ggplot(subset(prj$trks,year<=2020))                          +
+   geom_line(aes(year,stock,  linetype=Percentile),col="blue") +
+   geom_line(aes(year,harvest,linetype=Percentile),col= "red") +
+   scale_linetype_manual(values=c(2,1,2))                     +
+   coord_cartesian(ylim=c(0,3))                               +
+   facet_wrap(~TAC,ncol=2)
```
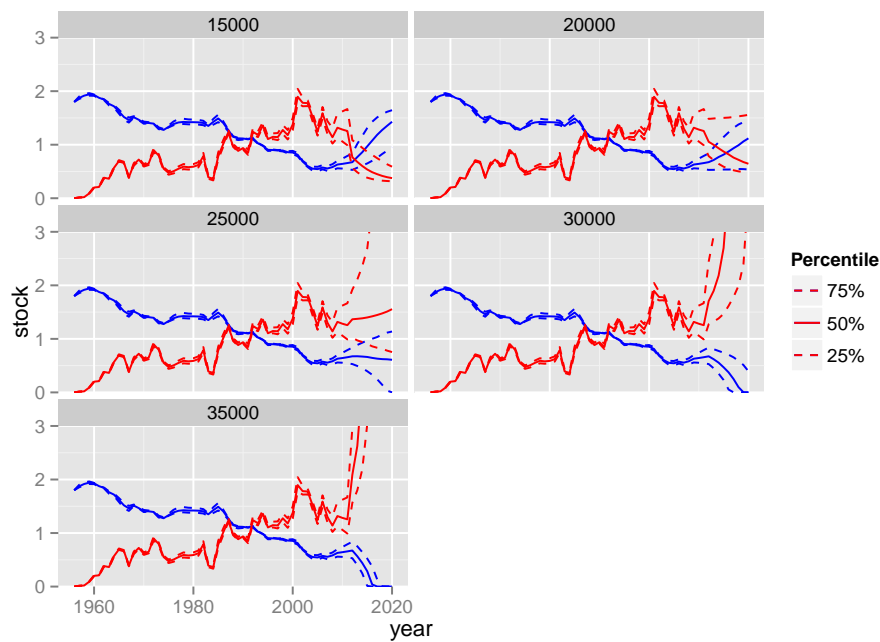
Figure 2: Historic and projected times series of biomass (blue) and harvest (red) by TAC for the interquartiles and medians

### 3.2. Kobe Framework

The Kobe framework is based on a phase plot where $F : F_{MSY}$ is plotted against $Biomass :$ $B_{MSY}$; quadrants are colour coded i.e. green (not overfished, no overfishing), red quadrant (i.e. overfished and overfishing) or yellow (otherwise). Since the main management objective is to keep a stock in the green quadrant projections are performed to evaluate the consequences of different TACs. The Kobe II strategy matrice (K2SM) summarise the probabilities for different levels of TAC across multiple years of being in the green quadrant.

*Phase Plot*

An example phase plot is shown in figure **??**

`ggplot2` produces an object, this can be saved for later use or modified by adding additional layers later. This means it is easy to conduct analyses on the fly and format them for publication when required.

Often the phase plot will have summarises multiple assessments; for example where several scenarios are conducted in order to reflect uncertainty about model structure. The example data set contains results from multiple assessments.

```
> data(sims)
> head(sims)

    Run iter year    stock       harvest   TAC
1 run02    1 1956 1.800000 0.0004169473 15000
2 run02    1 1957 1.873086 0.0139635494 15000
3 run02    1 1958 1.915449 0.0198786446 15000
4 run02    1 1959 1.940032 0.0891123383 15000
5 run02    1 1960 1.928930 0.2009482466 15000
6 run02    1 1961 1.880085 0.2110163113 15000
```

`sims` contains data by `run` and `TAC`. For each subset of the data.frame it will be neccessary to apply a function, e.g. to calculate the median of a time series across iterations, and then combine the results back into a summary data frame. **kobe** therefore contains methods to make it easier to use the **plyr** package which implements the split-apply-combine strategy for **R**. Allowing you to split up a big data structure into homogeneous pieces, apply a function to each piece and then combine all the results back together.

`kobeTrks` calculates percentiles of a time series with replicates (e.g. from a bootstrapped assessment). First we subset the data to extract only the historic assessment data, then calculate the medians of `stock` and `harvest` e.g.

```
> dat =subset(sims,year<=2010 & TAC==15000)
> trks=ddply(dat,.(Run,year,TAC), function(x) kobeTrks(x$stock,x$harvest,prob=c(0.5)))
> head(trks)

    Run year   TAC Percentile    stock      harvest
1 run02 1956 15000        50% 1.800000 0.0004210195
2 run02 1957 15000        50% 1.869179 0.0141144108
```

```
> kp=kobePhase(subset(sims, year==2010 & TAC==15000)) +
+          geom_point(aes(stock,harvest,group=Run,col=Run))
> kp
```
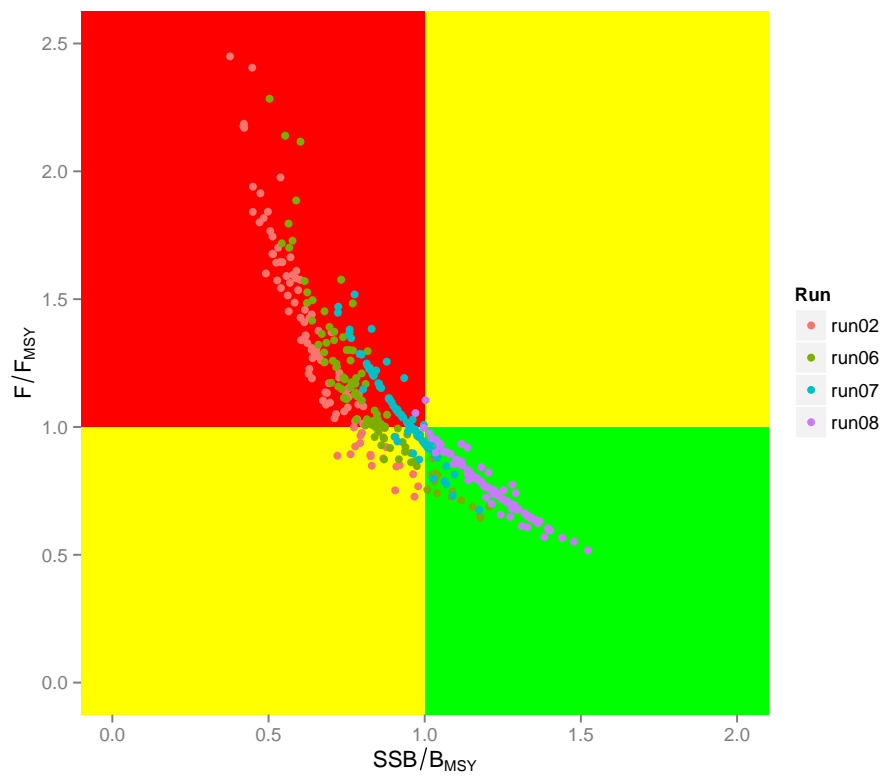


Figure 3: Phase plot of fishng mortality and stock status reletive to $F_{MSY}$ and $B_{MSY}$

```
3 run02 1958 15000          50% 1.910588 0.0200946618
4 run02 1959 15000          50% 1.935521 0.0900287395
5 run02 1960 15000          50% 1.926317 0.2027898363
6 run02 1961 15000          50% 1.877325 0.2127734574
```

We can then add the medians of the historic assessments to the phase plots by adding layers to the `ggplot2` object `kp`, i.e. `geom_path` adds an extra layer plotting the time series medians and `geom_point` the medians in the last assessment year. We then plot the results by assessment using `facet_wrap` to split them into multiple panels. Finally we get rid of the legend for `run` since runs are plotted by panel.

The phase plots show the cross correlations between stock and harvest, but many points overlay each other so it is hard to determine the actual probabilities or densities. To overcome this difficulty contours showing the bivariate probabilities can be plotted using `kobeP` e.g.

```
> kp2 = kp + geom_path(aes(x,y,group=level),colour="blue",
+                 data=ddply(subset(sims,year==2010 & TAC==15000),.(Run), function(pts
+                 facet_wrap(~Run) +
+                 theme(legend.position = "none")
```

```
> kp + geom_path( aes(stock,harvest,group=Run,col=Run), data=trks) +
+      geom_point(aes(stock,harvest,group=Run), data=subset(trks,year==2010),col="cyan",si
+      facet_wrap(~Run) +
+      theme(legend.position = "none")
```
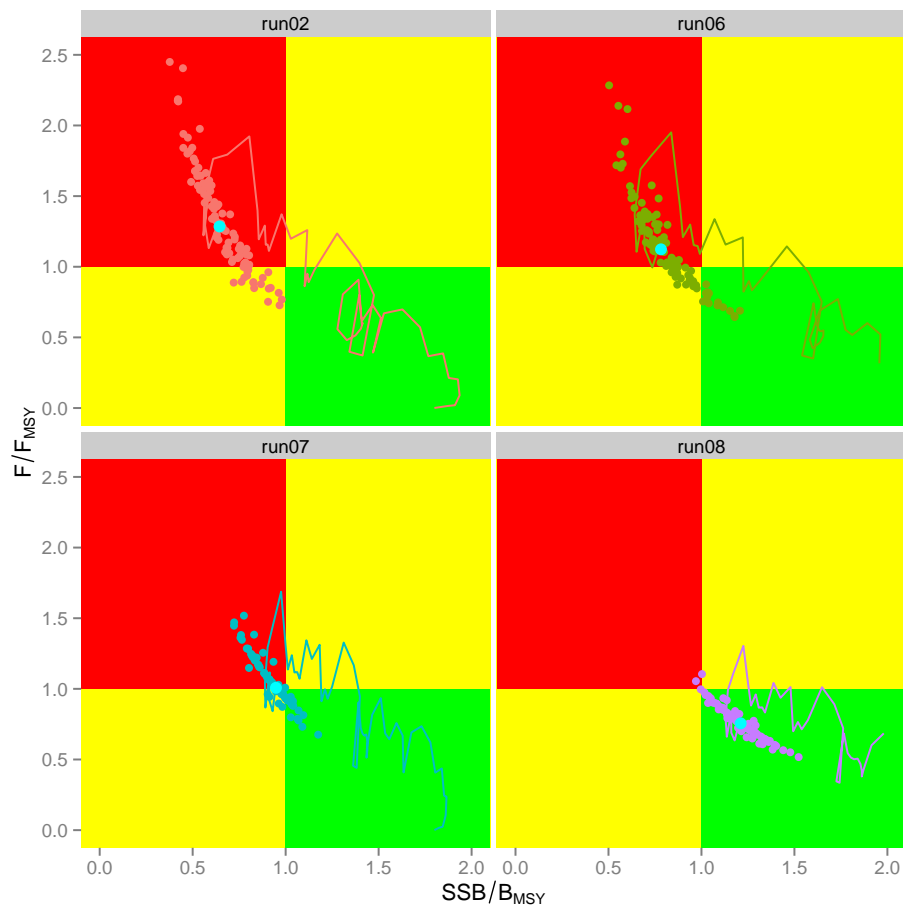


Figure 4: Phase plot of fishng mortality and stock status reletive to $F_{MSY}$ and $B_{MSY}$, large point and lines are the medians from the assessment and the panels correspond to each run.
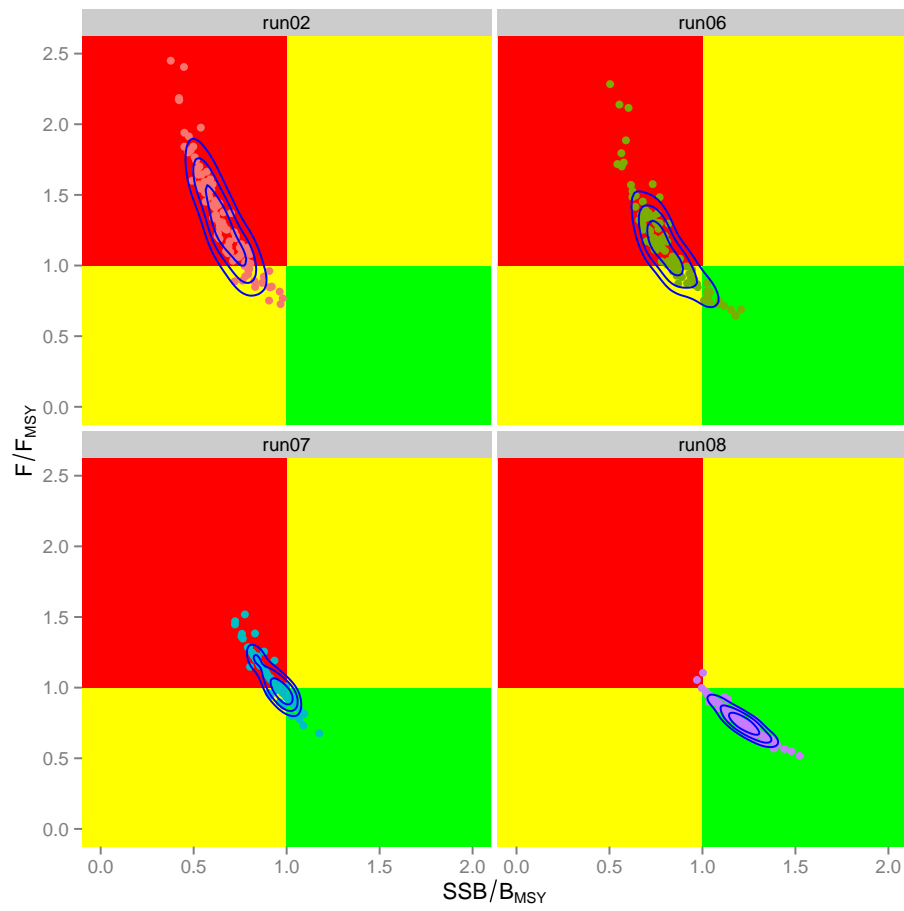
```
> print(kp2)
```



Figure 5: Phase plot of fishng mortality and stock status reletive to $F_{MSY}$ and $B_{MSY}$, large point and lines are the medians from the assessment and the panels correspond to each run.

Alternatively the marginal densities can be plotted e.g.

These show the values overlayed and so do not show the combined density. The same data can be also plotted by stacking them on top of one another e.g.

The phase and marginal density plots can be combined using `kobePhasemar` e.g.

```
> pts =subset(sims, year==2010 & TAC==15000)
> # stock density plot
> ggplot(pts) +
+   geom_density(aes(x=stock, y= ..count.., group=Run, fill=Run, alpha=0.4))
```
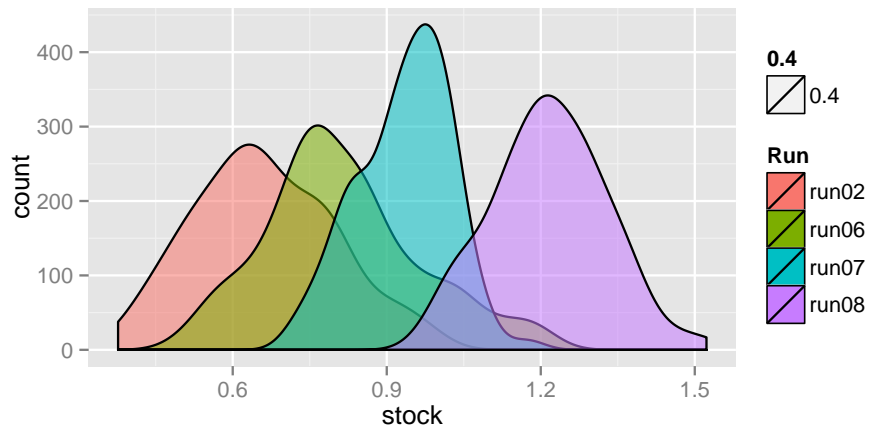


```
> ggplot(pts) +
+   geom_density(aes(x=stock, y=..count.., group=Run, fill=Run),
+                       fill="grey", col=grey(.9), position = "stack")
```
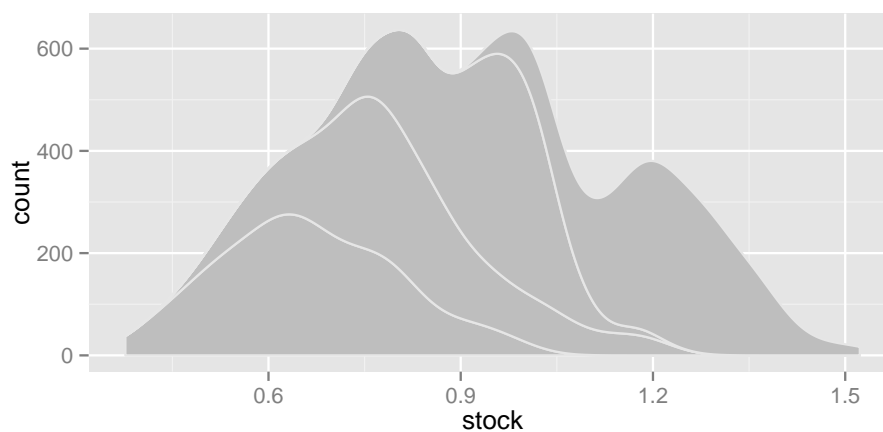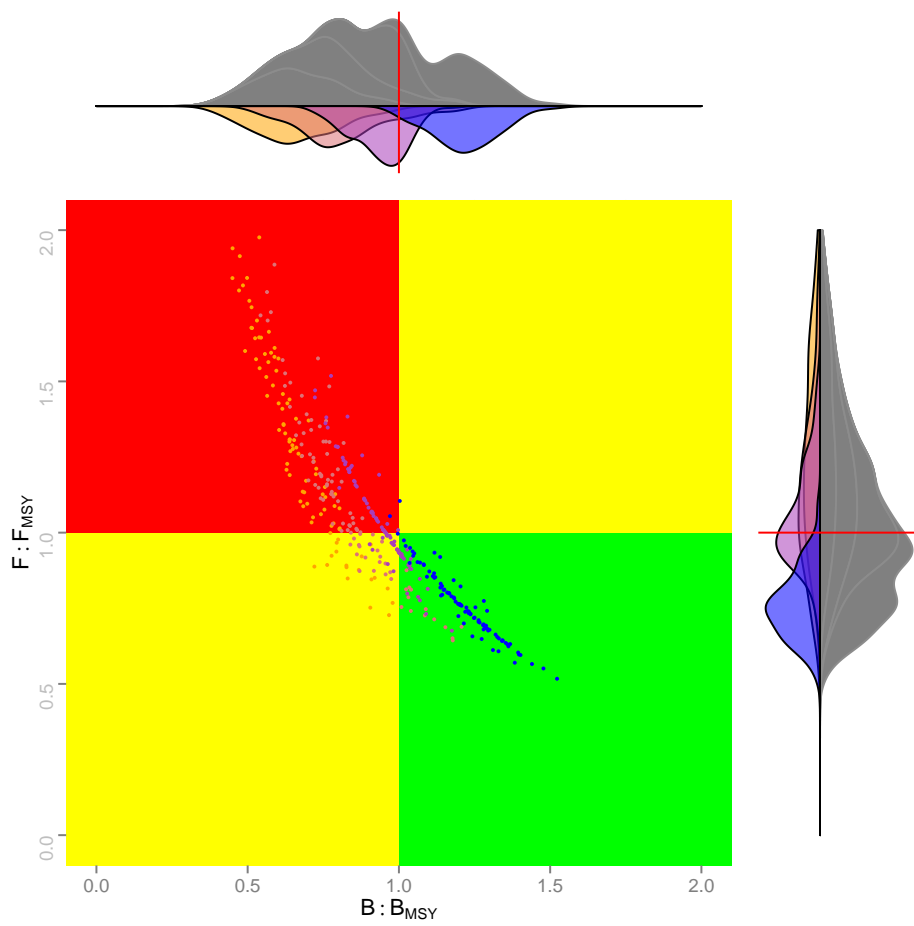
```
> ### Bespoke Stuff ###
> print(kobePhaseMar(transform(pts,group=Run)))
```

```
$harvest
```

```
$stock
```

```
$phase
```

*Pie Charts*

Although pie charts have their critics ...

*Kobe II Strategy Matrix*

# 4. Tabulation

## 4.1. Kobe II Strategy Matrix

Three Kobe matrices Tables **??**, **??** and **??** summarise the probabilities (by the ranges of 50-59 %, 60- 69 %, 70-79 %, 80-89 % and greater or equal to 90 %) for different levels of catch across multiple years of

- Biomass or SSB being greater than $B_{MSY}$ ;

- Fishing Mortality or Harvest Rate being less than $F_{MSY}$; and

- the combined probability of Biomass or SSB being greater than $B_{MSY}$ and Fishing Mortality or Harvest Rate being less than $F_{MSY}$

```
> # Making sure we can source code from github
> source("http://www.r-statistics.com/wp-content/uploads/2012/01/source_https.r.txt")
> # Reading in the function for using tabular on a cast_df object:
> source_https("https://raw.github.com/talgalili/R-code-snippets/master/tabular.cast_df.r"
> t.=ddply(subset(sims,year %in% 2013:2022),.(year,TAC),  kobeSmry)
> #t.=ddply(subset(sims,year %in% 2013:2022),.(year,TAC),  kobeSmry)[,c("year","TAC","gree
> #t.=melt(t.,id=c("TAC","year"))
>
> k2smTab=list()
> k2smTab[[1]]=cast(subset(t., year %in% 2013:2022),TAC~year,value="underFishing")
> k2smTab[[2]]=cast(subset(t., year %in% 2013:2022),TAC~year,value="underFished")
> k2smTab[[3]]=cast(subset(t., year %in% 2013:2022),TAC~year,value="green")
```

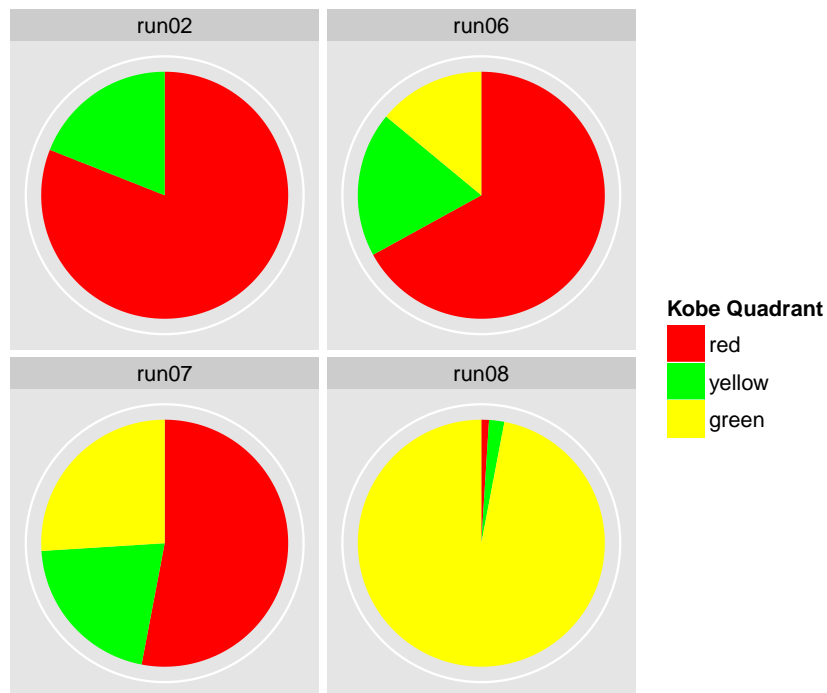| TAC | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|---|---|---|---|
| 15000 | 0.9175 | 0.9425 | 0.9575 | 0.9650 | 0.9675 | 0.9700 | 0.9725 | 0.9725 | 0.9750 | 0.9750 |
| 20000 | 0.7275 | 0.7575 | 0.7725 | 0.7825 | 0.7850 | 0.7925 | 0.8025 | 0.8250 | 0.8275 | 0.8400 |
| 25000 | 0.4075 | 0.4200 | 0.4250 | 0.4250 | 0.4275 | 0.4425 | 0.4425 | 0.4475 | 0.4500 | 0.4500 |
| 30000 | 0.1625 | 0.1475 | 0.1375 | 0.1300 | 0.1125 | 0.1050 | 0.0900 | 0.0775 | 0.0675 | 0.0650 |
| 35000 | 0.0325 | 0.0175 | 0.0125 | 0.0075 | 0.0075 | 0.0050 | 0.0050 | 0.0025 | 0.0025 | 0.0025 |

```
\begin{tabular}{lcccccccccc}
\hline
TAC  & 2013 & 2014 & 2015 & 2016 & 2017 & 2018 & 2019 & 2020 & 2021 & 2022 \\
\hline
15000  & $0.4925$ & $0.6050$ & $0.6625$ & $0.7375$ & $0.7800$ & $0.8000$ & $0.830$ & $0.87
```

```
> ### Pies ###
> pie.dat=ddply(subset(sims,year==2010 & TAC==15000),.(Run),kobeSmry,o=T)
> pie.dat=ddply(melt(pie.dat,id.vars="Run"),.(Run,variable), function(x) data.frame(value=
> ## pie charts
> ggplot(subset(pie.dat,value>0), aes(x =factor(1), y=value, fill = variable)) +
+   geom_bar(width = 1) +
+   coord_polar(theta="y") +
+   labs(fill='Kobe Quadrant') + xlab('') + ylab('')         +
+   scale_fill_manual(values=c("red","green","yellow"))      +
+   facet_wrap(~Run)                                         +
+   scale_x_discrete(breaks=NULL)                            +
+   scale_y_continuous(breaks=NULL)
```
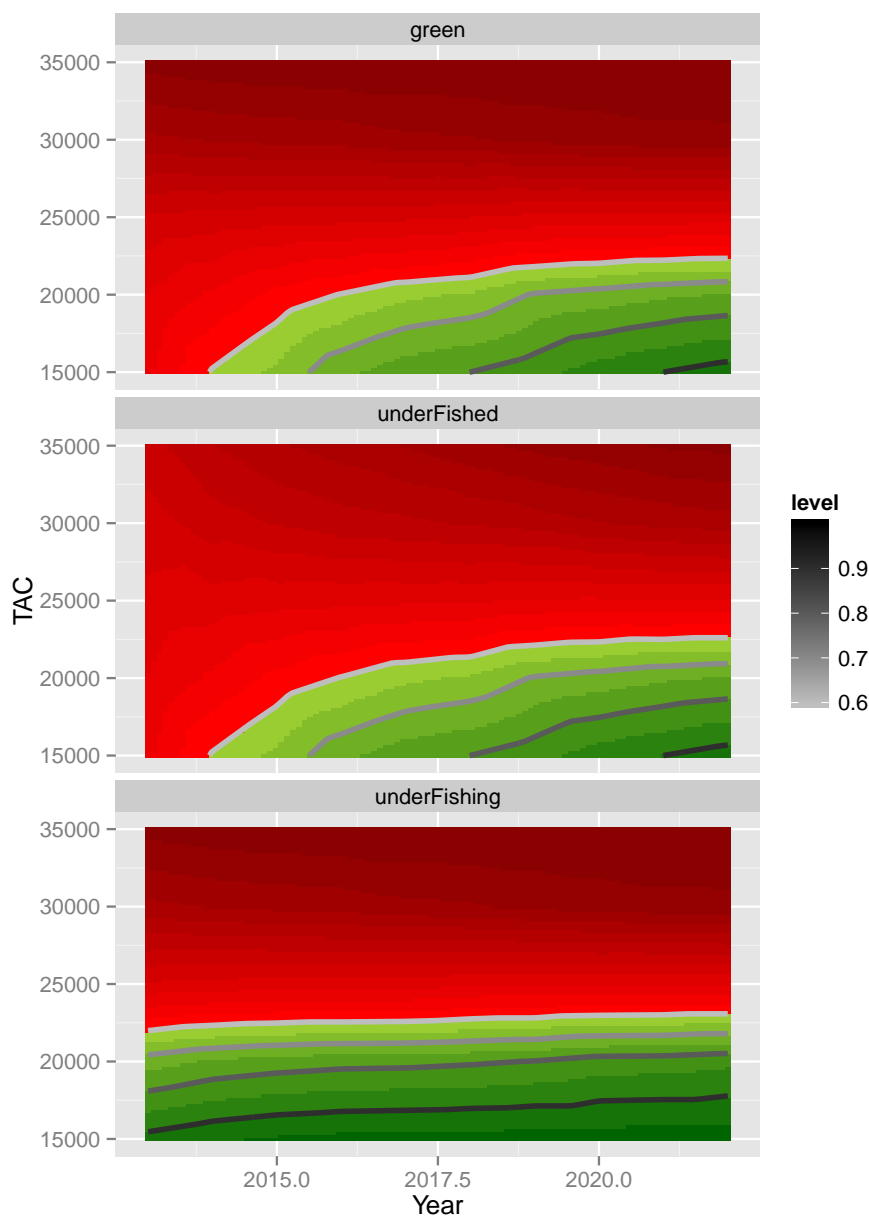
```
> kobe2012=subset(sims,year %in% 2013:2022)
> pdat=subset(ddply(kobe2012,.(year,TAC),kobeSmry),select=c(year,TAC,green,underFished,und
> pdat=melt(pdat,id.vars=c("year","TAC"))
> pdat=ddply(pdat, .(variable), function(x) kobeInterp(data.frame(x$year,x$TAC,x$value)))
> col.=c(colorRampPalette(c("red4","red"))(12),colorRampPalette(c("yellowgreen","darkgreen
> k2p = ggplot(aes(x=x,y=y,z=w),data=pdat)                        +
+           geom_tile(aes(x,y,fill=z))                            +
+           scale_fill_manual(values=col.,guide="none")    +
+           stat_contour(aes(colour = ..level..),size=1.2,  breaks=c(0.6,0.7,0.8,0.9))
+           scale_colour_gradient(low="grey", high="black", breaks=c(0.6,0.7,0.8,0.9),lab
+           facet_wrap(~variable,ncol=1)                          +
+           xlab("Year")+ylab("TAC")
> k2p
```

```
20000  & $0.4525$ & $0.5100$ & $0.5650$ & $0.6025$ & $0.6400$ & $0.6575$ & $0.705$ & $0.72
25000  & $0.3600$ & $0.3700$ & $0.3800$ & $0.3825$ & $0.3900$ & $0.4000$ & $0.415$ & $0.42
30000  & $0.1625$ & $0.1475$ & $0.1375$ & $0.1300$ & $0.1125$ & $0.1050$ & $0.090$ & $0.07
35000  & $0.0325$ & $0.0175$ & $0.0125$ & $0.0075$ & $0.0075$ & $0.0050$ & $0.005$ & $0.00
\hline
\end{tabular}
```

| TAC | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|---|---|---|---|
| 15000 | 0.4925 | 0.6050 | 0.6625 | 0.7375 | 0.7800 | 0.8000 | 0.830 | 0.8750 | 0.9000 | 0.9225 |
| 20000 | 0.4525 | 0.5100 | 0.5650 | 0.6025 | 0.6400 | 0.6575 | 0.705 | 0.7225 | 0.7425 | 0.7550 |
| 25000 | 0.3600 | 0.3700 | 0.3800 | 0.3825 | 0.3900 | 0.4000 | 0.415 | 0.4200 | 0.4225 | 0.4250 |
| 30000 | 0.1625 | 0.1475 | 0.1375 | 0.1300 | 0.1125 | 0.1050 | 0.090 | 0.0775 | 0.0675 | 0.0650 |
| 35000 | 0.0325 | 0.0175 | 0.0125 | 0.0075 | 0.0075 | 0.0050 | 0.005 | 0.0025 | 0.0025 | 0.0025 |

**Affiliation:**

Laurence Kell
ICCAT Secretariat
C/Corazón de María, 8.
28002 Madrid
Spain

E-mail: Laurie.Kell@iccat.int