# Random Grid Results

*Alex Tidd*

*09 November, 2018*

```
library(reshape)
library(plyr)
```

```
##
## Attaching package: 'plyr'

## The following objects are masked from 'package:reshape':
##
##     rename, round_any
```

```
library(tinytex)
#library(dplyr)

library(ggpubr)
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.3.2

## Loading required package: magrittr

##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':
##
##     mutate
```

```
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 3.3.2

## Loading required package: nlme

## This is mgcv 1.8-22. For overview type 'help("mgcv-package")'.
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.3.2
```

```
library(RColorBrewer)
```

```
load("/Users/alextidd/Dropbox/mydas/results/empd-results2.RData")
```

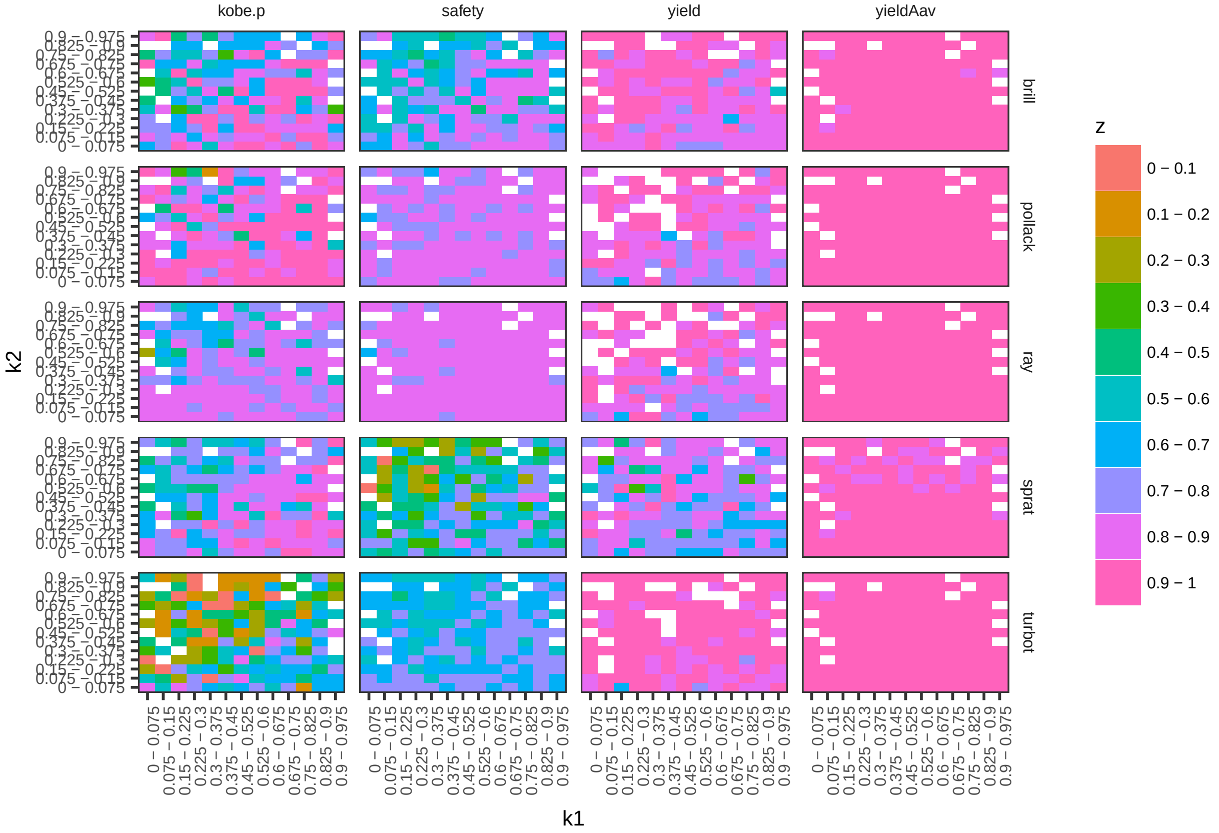
```
empd_pm$kobe.p=empd_pm$kobe.n/45
empd_pm$yieldAav = pmin(0.5,empd_pm$yieldAav)
empd_pm$yieldAav = 1 - empd_pm$yieldAav
test=melt(empd_pm, id.vars=c("spp","k1","k2"),measure.vars=c("safety","kobe.p","yield","yieldAav"))
```

### remove brackets resulting from cut

```
out=NULL
for (i in c("brill","turbot","ray","pollack","sprat")){
       for (k in c("safety","kobe.p","yield","yieldAav")) {
  fld  = subset(test, variable==k & spp==i)
  fld$spp=i
  fld$objective=k
  x = cut(fld$k1, seq(0,1, 0.075))
  y = cut(fld$k2,seq(0,1,0.075))

  x = gsub(","," - ",x,fixed=TRUE)
  fld$k1 = gsub("\\(|\\]","",x)
  y = gsub(","," - ",y,fixed=TRUE)
  fld$k2 = gsub("\\(|\\]","",y)

  fld=ddply(fld, .(spp,k1,k2, objective), summarise, z1=quantile(value,probs=0.5, na.rm=TRUE))
  z = cut(fld$z1,seq(0,1,0.1))
  z <- gsub(","," - ",z,fixed=TRUE)
  fld$z <- gsub("\\(|\\]","",z)

  out=rbind(out,fld)
       }
}
```

## Blocky image

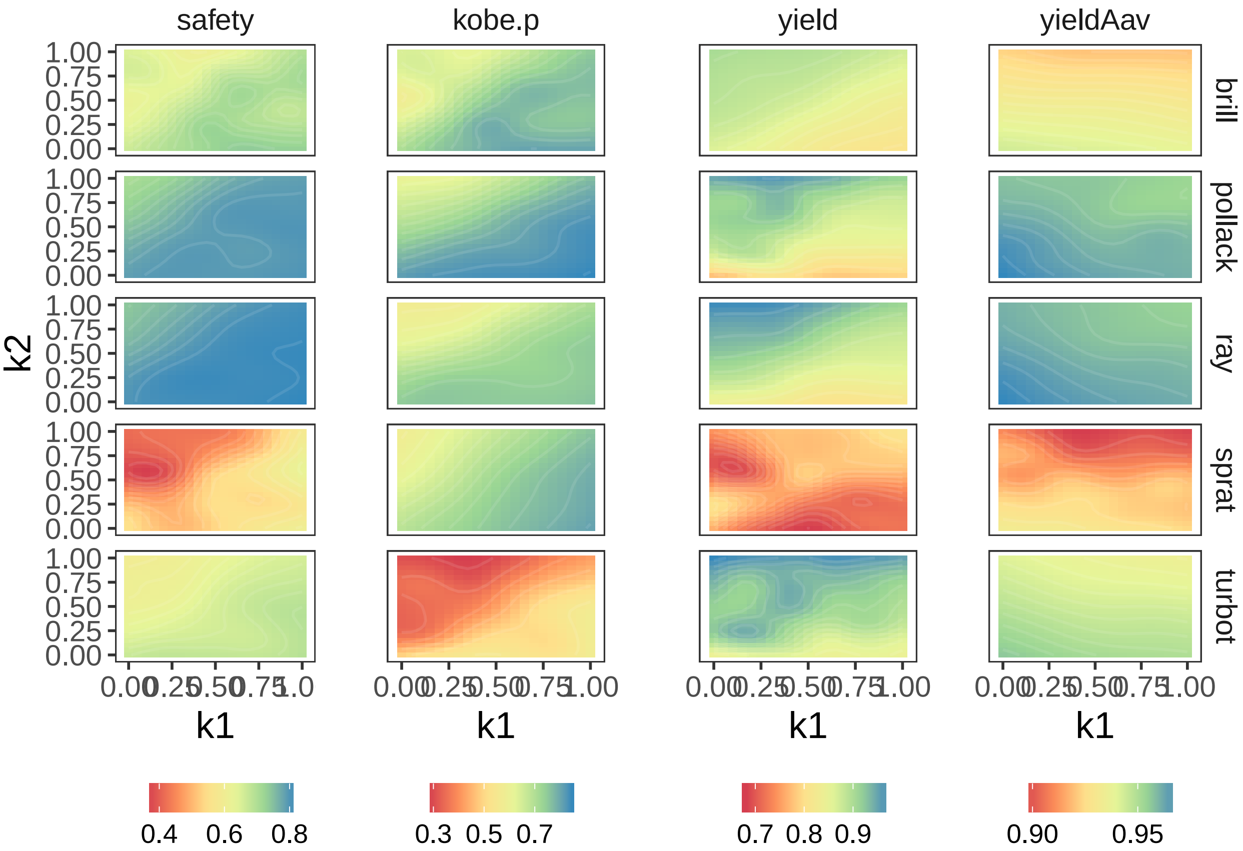
```
ggplot(na.omit(out) ,aes(k1,k2)) +
  geom_tile(aes(fill=z))+
  #scale_fill_manual("values",values=brewer.pal(9,"YlOrRd"))+
  facet_grid(spp~objective)+theme_bw()+
  theme(text = element_text(size=8),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        strip.background = element_blank(),
        axis.text.x = element_text(angle = 90, hjust = 1))
```



## modelled display using GAM

```
pairwise = list()
for (i in c("safety","kobe.p","yield","yieldAav")){
  for (j in c("brill","turbot","ray","pollack","sprat")){
    a    = subset(test, spp==j & variable==i)
    spl1 = gam(value ~ s(k1, k2, bs = 'sos'), data = a)
    # fine grid, coarser is faster
    datmat2 <- data.frame(expand.grid(k1 = seq(0, 1, 0.05), k2= seq(0, 1, 0.05)))
    resp = predict(spl1, datmat2, type = "response")
    datmat2$value <- resp
    datmat2$spp = j
    datmat2$objective = i
    pairwise=rbind(pairwise, datmat2)
  }
}
```

```
#myPalette = colorRampPalette(brewer.pal(6, "Greys"))
pairwise1 = list()
for (k in c("safety","kobe.p","yield","yieldAav")) {
  if(k %in% c("safety")){
  pairwise1[[k]] =ggplot(subset(pairwise, objective==k)) +
    aes(x = k1, y = k2, z = value, fill = value)  +
    geom_tile() +

    geom_contour(color = "white", alpha = 0.1) +
    scale_fill_distiller("",palette="Spectral", na.value="white", direction=1,breaks=c(0.4,0.6,0.8)) +
    #geom_contour(color = "white", alpha = 0.1) +
    #scale_fill_gradientn("",colours=myPalette(4), breaks=c(0.4,0.6,0.8))+
    theme_bw()+facet_grid(spp~objective) +
    theme(text = element_text(size=14),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          legend.position="bottom",
          legend.key.size = unit(0.75,"line"),
          legend.text=element_text(size=10),
          strip.background = element_blank(),
          strip.text.y = element_blank(),
          plot.margin = unit(c(0, 0, 0, 0), "cm")) + ylab("k2")+xlab("k1")
  }
if(k %in% c("kobe.p")){
  pairwise1[[k]] =ggplot(subset(pairwise, objective==k)) +
    aes(x = k1, y = k2, z = value, fill = value)  +
    geom_tile() +

    geom_contour(color = "white", alpha = 0.1) +
    scale_fill_distiller("",palette="Spectral", na.value="white", direction=1, breaks=c(0.3,0.5,0.7)) +
    #geom_contour(color = "white", alpha = 0.1) +
    #scale_fill_gradientn("",colours=myPalette(4), breaks=c(0.3,0.5,0.7))+
    theme_bw()+facet_grid(spp~objective) +
    theme(text = element_text(size=14),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          legend.position="bottom",
          legend.key.size = unit(0.75,"line"),
          legend.text=element_text(size=10),
          strip.background = element_blank(),
          strip.text.y = element_blank(),
          axis.text.y=element_blank(),
          axis.ticks.y=element_blank(),
          plot.margin = unit(c(0, 0.3, 0, 0.3), "cm")) +xlab("k1")+ ylab("")#t,r,b,l
}
  if(k %in% c("yield")){
    pairwise1[[k]] =ggplot(subset(pairwise, objective==k)) +
      aes(x = k1, y = k2, z = value, fill = value)  +
      geom_tile() +

      geom_contour(color = "white", alpha = 0.1) +
      scale_fill_distiller("",palette="Spectral", na.value="white", direction=1) +
      #geom_contour(color = "white", alpha = 0.1) +
      #scale_fill_gradientn("",colours=myPalette(4))+
      theme_bw()+facet_grid(spp~objective) +
      theme(text = element_text(size=14),
            panel.grid.major = element_blank(),
            panel.grid.minor = element_blank(),
            legend.position="bottom",
            legend.key.size = unit(0.75,"line"),
            legend.text=element_text(size=10),
            strip.background = element_blank(),
            strip.text.y = element_blank(),
            axis.text.y=element_blank(),
            axis.ticks.y=element_blank(),
            plot.margin = unit(c(0, 0.3, 0, 0.3), "cm")) +xlab("k1")+ ylab("")
  }
  if(k %in% c("yieldAav")) {
  pairwise1[[k]] =ggplot(subset(pairwise, objective==k)) +
    aes(x = k1, y = k2, z = value, fill = value)  +
    geom_tile() +
    geom_contour(color = "white", alpha = 0.1) +
    scale_fill_distiller("",palette="Spectral", na.value="white", direction=1, breaks=c(0.9,0.95)) +
    #geom_contour(color = "white", alpha = 0.1) +
    #scale_fill_gradientn("",colours=myPalette(4), breaks=c(0.9,0.95))+
    theme_bw()+facet_grid(spp~objective) +
    theme(text = element_text(size=14),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          legend.position="bottom",
          legend.key.size = unit(0.75,"line"),
          legend.text=element_text(size=10),
          strip.background = element_blank(),
          axis.text.y=element_blank(),
          axis.ticks.y=element_blank(),
          plot.margin = unit(c(0, 0, 0, 0), "cm")) +xlab("k1")+ ylab("")
  }
}
ggarrange(plotlist = pairwise1, ncol=4)
```

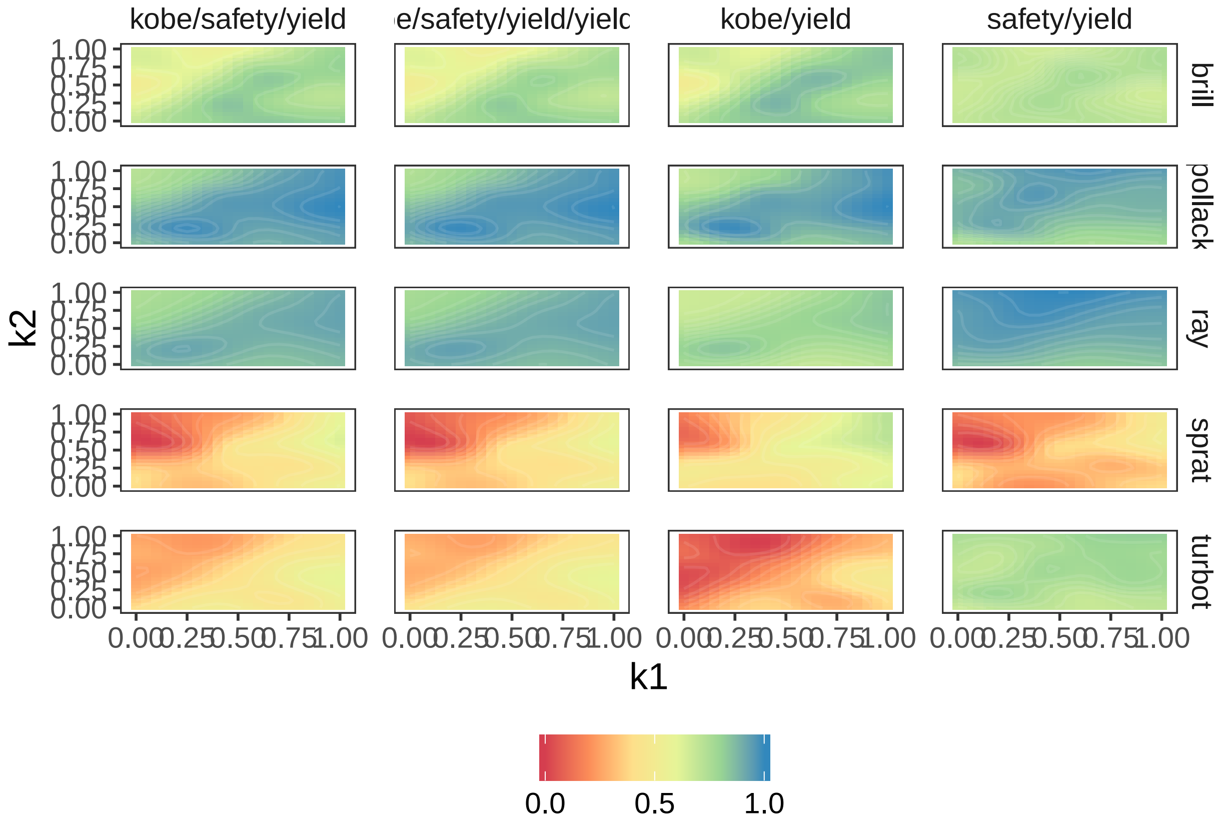


## utilities

```
pair=reshape( pairwise, idvar = c("spp","k1","k2"), v.names = "value", timevar = "objective",
              direction = "wide")
colnames(pair)[4:7] = c("safety","kobe.p","yield","yieldAav")

pair$k_y=apply(pair[c(5,6)],c(1),sum,na.rm=t)/2
pair$s_y=apply(pair[c(4,6)],c(1),sum,na.rm=t)/2
pair$k_s_y=apply(pair[c(4:6)],c(1),sum,na.rm=t)/3
pair$k_s_y_y=apply(pair[c(4:7)],c(1),sum,na.rm=t)/4


pair2=melt(pair, id.vars=c("spp","k1","k2"), measure.vars=c("k_y","s_y","k_s_y","k_s_y_y"))
pair2$variable=as.character(pair2$variable)
pair2$variable[pair2$variable=="k_y"] = "kobe/yield"
pair2$variable[pair2$variable=="s_y"] = "safety/yield"
pair2$variable[pair2$variable=="k_s_y"] = "kobe/safety/yield"
pair2$variable[pair2$variable=="k_s_y_y"] = "kobe/safety/yield/yieldvar"
minMax<-function(x,na.rm=TRUE) (x-min(x,na.rm=na.rm))/diff(range(x,na.rm=na.rm))
te=ddply(pair2,.(variable),transform, var2=minMax(value))
```

### utilities plots

```
ggplot(te) +
  aes(x = k1, y = k2, z = var2, fill = var2) +
  geom_tile() +
  #coord_equal() +
  #geom_contour(color = "white", alpha = 0.5) +
  #scale_fill_distiller("",palette="Spectral", na.value="white", direction=1) +
  geom_contour(color = "white", alpha = 0.1) +
  scale_fill_distiller("",palette="Spectral", na.value="white", direction=1, breaks=c(0, 0.5, 1))+
  #scale_fill_gradientn("utility",colours=myPalette(4), breaks=c(0, 0.5, 1))+
  theme_bw()+facet_grid(spp~variable) +
  theme(text = element_text(size=14),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        strip.background = element_blank(),
        legend.position="bottom",panel.spacing = unit(1, "lines")
        ) + ylab("k2")+xlab("k1")
```



```
#ggsave(filename='/Users/alextidd/Documents/fig6.png',last_plot(),dpi=300, units='in',width=12,height=1
```

## Author information

**Alex Tidd**. emperorfish@gmail.com

## Acknowledgements

This vignette and many of the methods documented in it were developed under the MyDas project funded by the Irish exchequer and EMFF 2014-2020. The overall aim of MyDas is to develop and test a range of assessment models and methods to establish Maximum Sustainable Yield (MSY) reference points (or proxy MSY reference points) across the spectrum of data-limited stocks.