The results below are generated from an R script.

```
---
title: "Preliminary analysis of growing degree days (GDD) for Mainland Feldmark ecosystem"
author: "Ferrer-Paris, J.R."
date: "February 10, 2021"
output:
  pdf_document: default
  word_document: default
editor_options:
  chunk_output_type: console
---

```{r setup, include=FALSE}

## load libraries
require(ncdf4)
require(chron)
require(raster)
require(sf)
require(abind)
require(units)

## set up data directories and other variables

SCRIPTDIR <- sprintf("/home/%s/proyectos/UNSW/australian-alps-climate-change",system("whoami",intern=T))
MAPS <- sprintf("/srv/scratch/%s/gisdata/aust-alps",system("whoami",intern=T))
RDATA <- sprintf("/srv/scratch/%s/gisdata/aust-alps/Rdata",system("whoami",intern=T))
PERIOD <- "1990-2009"
MODEL <- "CCCMA3.1"
PRM <- "R2"

```

Target ecosystem is Feldmark, we use the shapefile with the final map:

```{r feldmark shape, include=TRUE}
arch <- sprintf("%s/all states feldmark/final outputs/all_states_feldmark_min.shp",MAPS)
eco.xy <- st_read(arch)
```

Climate data is in coarse cells (10 km²), while ecosystems are very small (<40 ha)

```{r feldmark area, include=TRUE}
eco.area <- st_area(eco.xy)
units(eco.area) <- with(ud_units, ha)
##units(eco.area) <- with(ud_units, km^2)
hist(eco.area)
```

Thus we can simplify the analysis using the centroids of the ecosystem polygons. We could use the areas
```

```{r feldmark centroids, include=TRUE}
eco.centroids <- st_coordinates(st_transform(st_centroid(eco.xy),crs = "EPSG:4326"))

```

Load GDD data for one model and scenario and aggregate GDD values per cell using inverse distance weight

```{r calculate GDD, include=TRUE}
if (!exists("gdd.eco")) {
   gdd.eco <- data.frame(eco.centroids)

   for (PERIOD in c("1990-2009","2020-2039","2060-2079")) {
      load(sprintf("%s/%s-%s-%s.rda",RDATA,PERIOD,MODEL,PRM))

      for (yy in unique(GDD$year)) {
         ss <- subset(GDD, year %in% yy & n>360 & lon>(min(eco.centroids[,1])-1) & lon<(max(eco.centroic
         if(nrow(ss)>0) {

         dst1 <- pointDistance(ss[,2:1],eco.centroids,lonlat=T,allpairs=T)

         #Inverse distance weighting
         dst1[dst1>50000] <- NA
         w <- 1/dst1^3
         gdd.eco[yy] <- apply(w,2,function(x) sum(x*ss$GDD,na.rm=T)/sum(x,na.rm=T))

         }
      }
   }
}

valid.eco <- apply(!is.na(gdd.eco[,-(1:2)]),1,sum)>10
gdd.eco <- gdd.eco[valid.eco,]
head(gdd.eco)

```

For one location, we can calculate the trend in annual GDD for the whole time period, and interpolate th

```{r interpolate values, include=TRUE}
y <- unlist(gdd.eco[1,-(1:2)])
x <- as.numeric(colnames(gdd.eco)[-(1:2)])
mdl <- lm(y~x,subset=y>0)
IV <- predict(mdl,data.frame(x=2000))
FV <- predict(mdl,data.frame(x=2050))

plot(x,y,type='n',xlab="Year",ylab="GDD")
rect(2000,-3000,2050,3000,col="palegoldenrod",border="palegoldenrod")
points(x,y,col="maroon",pch=1.2)
abline(mdl,lty=2)
points(2000,IV,pch=19,cex=1.6,type="p")
points(2050,FV,pch=19,cex=1.6,type="p")
```

````
text(2000,IV*.9,sprintf("Initial value = %0.2f",IV),adj=c(0,1))
text(2050,FV*1.1,sprintf("Final value = %0.2f",FV),adj=c(1,1))
```

If we set an *arbitrary collapse* value of $GDD[collapse]=2000$, the relative severity for this location

```{r relative severity, include=TRUE}

CT <- 2000

(FV-IV)/(CT-IV)

```

We can now repeat this for all locations, and calculate the relative severity for all units:

```{r extent and severity, include=TRUE}
x <- as.numeric(colnames(gdd.eco)[-(1:2)])
CT <- 2000
eco.RS <- data.frame()
for (k in 1:nrow(gdd.eco)) {
   y <- unlist(gdd.eco[k,-(1:2)])
   mdl <- lm(y~x,subset=y>0)
   IV <- predict(mdl,data.frame(x=2000))
   FV <- predict(mdl,data.frame(x=2050))
   eco.RS <- rbind(eco.RS,data.frame(k,RS=(FV-IV)/(CT-IV)))
}
summary(eco.RS$RS)
```

In this case, relative severity is > 50% for all localities (>80% extent), thus the resulting category w


We could use reference data to estimate a more appropriate collapse threshold. For example *Hakea microc

```{r Hakea microcarpa, include=TRUE}
arch <- sprintf("%s/Hakea-microcarpa-locs.csv",MAPS)
Hm.xy <- unique(read.table(arch,head=F))
coordinates(Hm.xy) <- 1:2
proj4string(Hm.xy) <-  '+proj=utm +zone=55 +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
Hm.ll <- spTransform(Hm.xy,"+init=epsg:4326")
```

The occurrence records are very widespread, in Tasmania the distribution of *Hakea microcarpa* has littl

```{r distribution all, include=TRUE}
plot(Hm.ll,col=2,cex=.5)
points(eco.centroids,cex=.5)
```

But in the mainland the Feldmark has a very restricted distribution (2 cells from the NARCLiM domain), a

```{r distribution mainland, include=TRUE}
````

```
plot(Hm.ll,col=2,cex=.5,ylim=c(-37,-36),xlim=c(148,149))
points(unique(GDD[,2:1]),pch=22,cex=8)
points(eco.centroids,cex=.5)
```

We can calculate the expected GDD values for the locations with *Hakea microcarpa*:

```{r GDD Hakea, include=TRUE}
if (!exists("gdd.col")) {
   xys2 <- coordinates(Hm.ll)
   gdd.col <- data.frame(xys2)

   for (PERIOD in c("1990-2009")) {
      load(sprintf("%s/%s-%s-%s.rda",RDATA,PERIOD,MODEL,PRM))

      for (yy in unique(GDD$year)) {
         ss <- subset(GDD, year %in% yy & n>360 & lon>(min(eco.centroids[,1])-1) & lon<(max(eco.centroid
         if(nrow(ss)>0) {
            dst2 <- pointDistance(ss[,2:1],xys2,lonlat=T,allpairs=T)
            #Inverse distance weighting
            dst2[dst2>50000] <- NA
            w <- 1/dst2^3
            gdd.col[yy] <- apply(w,2,function(x) sum(x*ss$GDD,na.rm=T)/sum(x,na.rm=T))
         }
      }
   }
}
head(gdd.col)

```

But there is considerable overlap and no clear threshold to separate the ecosystem from the areas with p

```{r GDD Hakea vs Feldmark, include=TRUE}
x <- as.numeric(colnames(gdd.col)[-(1:2)])
y <- as.numeric(colnames(gdd.eco)[-(1:2)])

boxplot(gdd.col[,-(1:2)],at=x,col=grey(.6),border=grey(.4))
matpoints(y,t(gdd.eco[,-(1:2)]),pch=1,col="darkgreen",cex=1)

```

Error: attempt to use zero-length variable name
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

R version 4.0.2 (2020-06-22)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: CentOS Linux 7 (Core)
##
Matrix products: default
BLAS/LAPACK: /apps/intel/Composer/compilers_and_libraries_2020.0.166/linux/mkl/lib/intel64_lin/libmkl
```
```

```
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.29
##
## loaded via a namespace (and not attached):
## [1] compiler_4.0.2 magrittr_1.5   tools_4.0.2    stringi_1.4.6  highr_0.8
## [6] stringr_1.4.0  xfun_0.21      evaluate_0.14
```

```r
Sys.time()
```

```
## [1] "2021-02-11 17:25:47 AEDT"
```