DataPrep - Working for Water plots

Jasper Slingsby, Adam M. Wilson and Glenn R. Moncrieff

This script assembles various environmental layers into a common 30m grid for the Cape Peninsula. It also calculates veg age based on the fire data.

Index raster

Import raster of an index grid (`ig`) to spatially connect all the datasets.

```
ig=raster(paste0(datadir,"clean/indexgrid_landsat_30m.grd"))
```

Vegetation

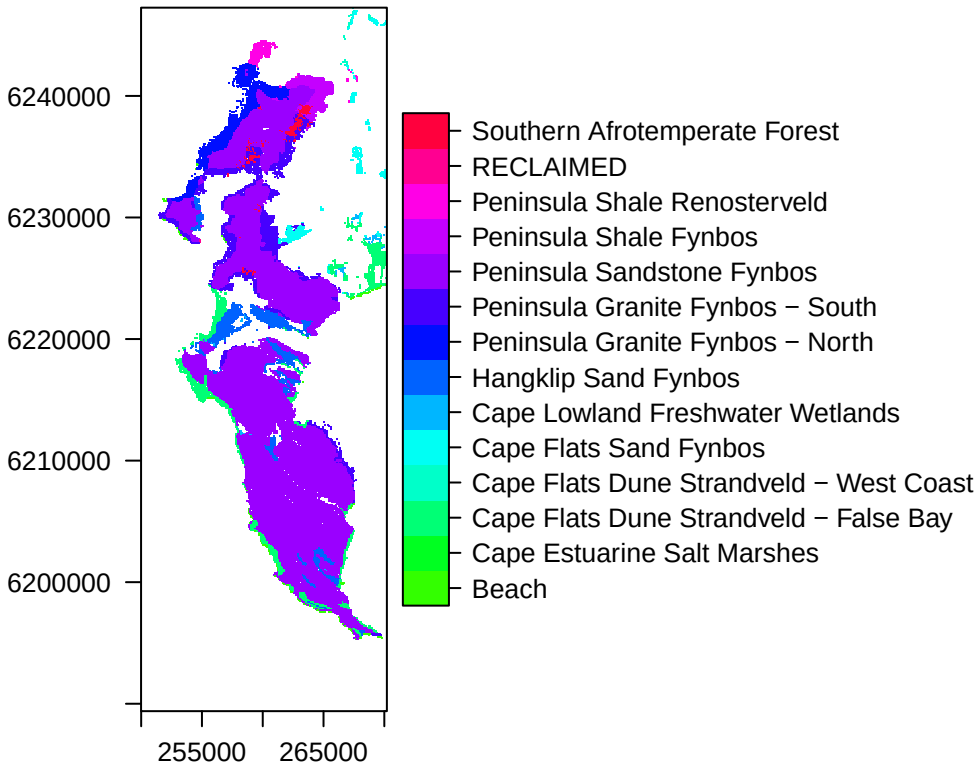
```
rv=readOGR(dsn=paste0(datadir,"raw/VegLayers/Vegetation_Indigenous_Remnants"), layer="Vegetation_Indige
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/glennmoncrieff/Documents/GIS/Postfire_test/data/raw/VegLayers/Vegetation_Indigenous_
## with 3428 features
## It has 7 fields
```

```
#remnant veg layer - readOGR() reads shapefiles
#rv; summary(rv$National_); summary(rv$Subtype); summary(rv$Community); levels(rv@data$National_)
rv_meta=data.frame(1:length(levels(rv@data$National_)), levels(rv@data$National_)) #save VegType metada
colnames(rv_meta)=c("ID", "code") #rename columns
write.csv(rv_meta, "data/vegtypecodes.csv", row.names=F)

# reproject to the CRS of the Landsat index grid (UTM 34S)
rv=spTransform(rv,CRS(proj4string(ig)))
```

Extract the national veg types from the veg layer into a 30m raster based on the index grid

```
rvrfile="data/vegtypes_landsat_30m.tif"
## note the if(!file.exists)) first checks if the file already exists so you don't rerun this everytime
if(!file.exists(rvrfile))
  rvr=rasterize(rv, ig, field=c("National_"), fun="max",file=rvrfile) #get national veg type for each c
## read it back in and 'factorize' it
rvr=raster(rvrfile)
rvr=as.factor(rvr)
rv_meta$code=as.character(rv_meta$code)
levels(rvr)=rv_meta[levels(rvr)[[1]]$ID,]
levelplot(rvr,col.regions=rainbow(nrow(rv_meta),start=.3))
```



Count number of veg types for each cell (i.e. ID mixed cells)

```
rvcfile="data/count_vegtypes_landsat_30m.tif"
if(!file.exists(rvcfile))
  rvc=rasterize(rv, ig, field=c("National_"), fun="count",file=rvcfile)
rvc=raster(rvcfile)
```

Are there any mixed cells?

```
table(values(rvc))
```

```
##
##      1      2
## 337337   5617
```

load WfW data

```
alien_plots<-read.csv(paste0(datadir,"clean/alienplots.csv"),stringsAsFactors = FALSE)
alien_xy<-cbind(as.numeric(alien_plots$x),as.numeric(alien_plots$y))
alien_xy<-SpatialPoints(alien_xy,crs(ig))
alien_data<-SpatialPointsDataFrame(alien_xy,alien_plots[,4:8])


##load plot clearing data
alien_n <- readOGR(dsn=paste0(datadir,"raw/AlienPlots/Nbal Analysis_TMNP_20160314"),"Nbal Analysis_TMNP
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/glennmoncrieff/Documents/GIS/Postfire_test/data/raw/AlienPlots/Nbal Analysis_TMNP_20
## with 306 features
## It has 123 fields
```

```
alien_c <- readOGR(dsn=paste0(datadir,"raw/AlienPlots/Nbal Analysis_TMNP_20160314"),"Nbal Analysis_TMNP
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/glennmoncrieff/Documents/GIS/Postfire_test/data/raw/AlienPlots/Nbal Analysis_TMNP_20
## with 436 features
## It has 123 fields
```

```
alien_s <- readOGR(dsn=paste0(datadir,"raw/AlienPlots/Nbal Analysis_TMNP_20160314"),"Nbal Analysis_TMNP
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/glennmoncrieff/Documents/GIS/Postfire_test/data/raw/AlienPlots/Nbal Analysis_TMNP_20
## with 206 features
## It has 123 fields
```

```
alien_n<-spTransform(alien_n,CRS(proj4string(ig)))
alien_c<-spTransform(alien_c,CRS(proj4string(ig)))
alien_s<-spTransform(alien_s,CRS(proj4string(ig)))

nn<-seq(1,nrow(alien_n@data),by=1)
nc<-seq(max(nn)+1,max(nn)+nrow(alien_c@data),by=1)
ns<-seq(max(nc)+1,max(nc)+nrow(alien_s@data),by=1)

nn1 <- spChFIDs(alien_n, as.character(nn))
nc1 <- spChFIDs(alien_c, as.character(nc))
ns1 <- spChFIDs(alien_s, as.character(ns))

temp<-spRbind(nn1,nc1)
vegclear<-spRbind(temp,ns1)
```

Fire data

```
fi=readOGR(dsn=paste0(datadir,"raw/Fire"), layer="CapePenFires") #Cape Peninsula fires history layers 1
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/glennmoncrieff/Documents/GIS/Postfire_test/data/raw/Fire", layer: "CapePenFires"
## with 4578 features
## It has 9 fields
```

```
fi=spTransform(fi,CRS(proj4string(ig)))

### Extract fire history data and convert to a 30m raster
fi$STARTDATE[which(fi$STARTDATE==196201001)]=19620101#fix an anomalous date...

#Raster showing total numbers of fires in each grid cell
```

```
ficfile="data/fires_number_1962to2007_landsat_30m.tif"
if(!file.exists(ficfile))
    fic=rasterize(fi, ig, field=c("YEAR"), fun="count",file=ficfile)

fic=raster(ficfile)
```

Rasterize fire data into annual fire maps

```
## Set up a "Days since 1 January 1960" column on the fire polygons
sdate=fi$STARTDATE #get the unique list of years in which fires occurred
sdate[which(substr(sdate, 5, 8)=="0000")]=sdate[which(substr(sdate, 5, 8)=="0000")]+1231 #set those fir

ddate=as.Date(as.character(sdate), format="%Y%m%d")
sdate=as.numeric(ddate-as.Date("19600101", format="%Y%m%d"))
fi$Day=sdate
fi$DayDate=ddate
sdate=sort(unique(sdate))
```

NDVI Compositing

```
getNDVI=function(file,datefile,prefix){
  ndvi=stack(paste0(datadir,"raw/NDVI/",file))
  NAvalue(ndvi)=0
offs(ndvi)=-2
gain(ndvi)=.001
dates=as.Date(read.csv(paste0(datadir,"raw/NDVI/",datefile),header=F)[,1])
names(ndvi)=paste0(prefix,sub("-","",dates))
ndvi=setZ(ndvi,dates)
}
```

Now use the function to read in the data and add the relevant metadata.

```
l4=getNDVI(file="2016045_v1g_LT4_L1T_TOA_daily__1988-03-08-1992-11-14.tif",
           datefile="LT4_L1T_TOA2.csv",prefix="L4_")
l5=getNDVI(file="2016045_v1g_LT5_L1T_TOA_daily__1984-06-09-2011-04-17.tif",
           datefile="LT5_L1T_TOA2.csv",prefix="L5_")
l7=getNDVI(file="2016045_v1g_LE7_L1T_TOA_daily__1999-08-30-2016-03-21.tif",
           datefile="LE7_L1T_TOA2.csv",prefix="L7_")
l8=getNDVI(file="2016045_v1g_LC8_L1T_TOA_daily__2013-05-24-2016-03-13.tif",
           datefile="LC8_L1T_TOA2.csv",prefix="L8_")
```

Let's check out one of the LANDSAT objects. Raster provides a summary by just typing the object's name:

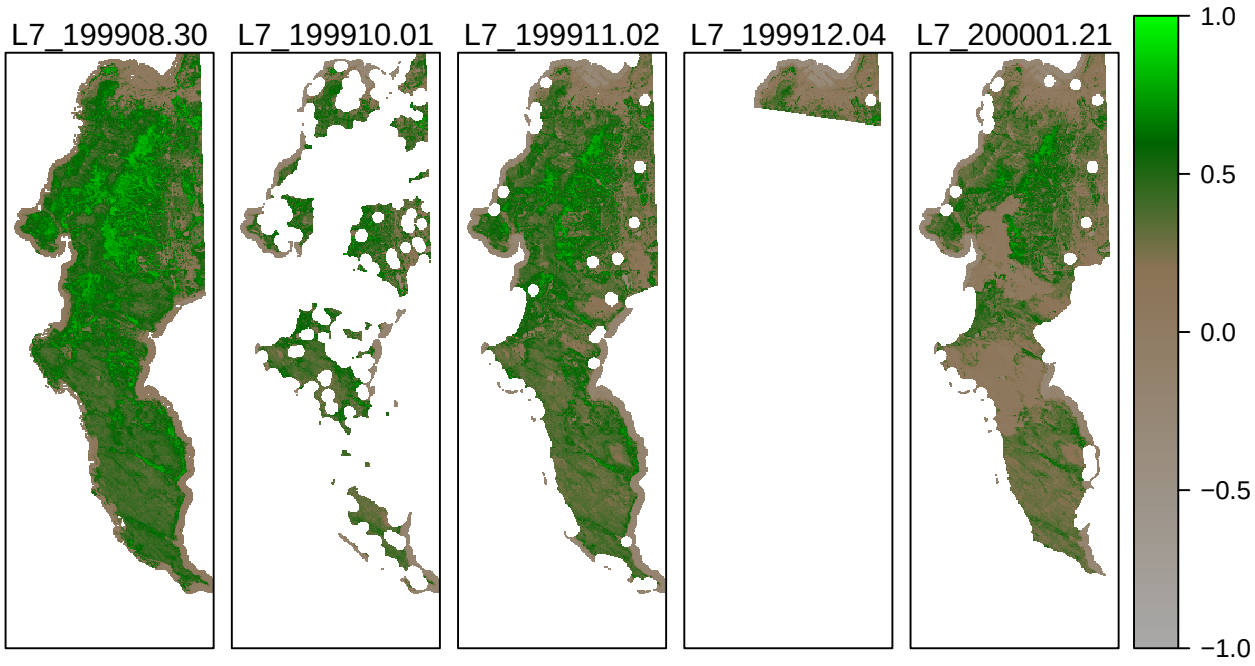
```
l7
```

```
## class       : RasterStack
## dimensions  : 1929, 674, 1300146, 229  (nrow, ncol, ncell, nlayers)
## resolution  : 30, 30  (x, y)
```

```
## extent      : 249990, 270210, 6189390, 6247260  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=34 +south +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
## names       : L7_199908.30, L7_199910.01, L7_199911.02, L7_199912.04, L7_200001.21, L7_200004.10, L7
## min values  :      -34.768,      -34.768,      -34.768,      -34.768,      -34.768,      -34.768,
## max values  :       30.767,       30.767,       30.767,       30.767,       30.767,       30.767,
## time        : 1999-08-30 - 2016-03-21 (range)
```

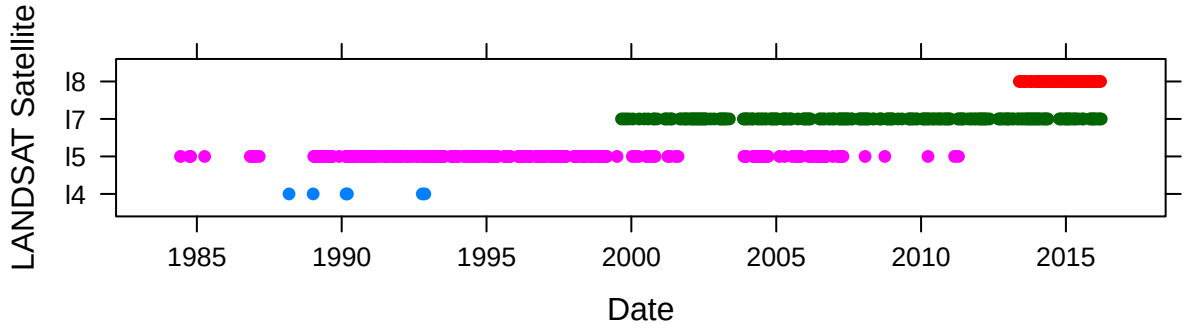
And a plot of a few different dates:

```
yearind=which(getZ(l7)%in%getZ(l7)[1:5])
levelplot(l7[[yearind]],col.regions=cndvi()$col,cuts=length(cndvi()$at),at=cndvi()$at,layout=c(length(y
```



There is some temporal overlap between sensors, let's look at that:

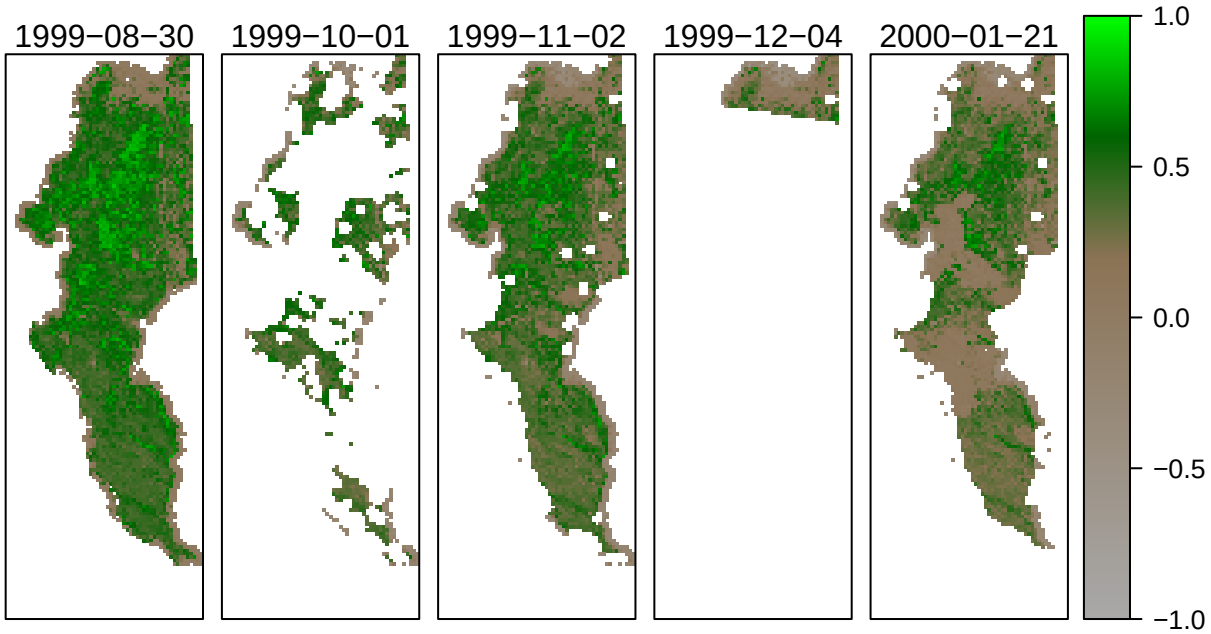
```
tl=melt(list(l4=getZ(l4),l5=getZ(l5),l7=getZ(l7),l8=getZ(l8)))
xyplot(as.factor(L1)~value,data=tl,pch=16,groups=as.factor(L1),asp=.15,lwd=5,ylab="LANDSAT Satellite",x
```



There are several ways these data could be combined.
The individual scenes could be assessed for quality (cloud contamination, etc.), sensors could be weighted by sensor quality (newer=better?).
Today, we'll simply combine (stack) all the available observations for each pixel.

```
ndates=unique(tl$value)
## write that to disk for later use
write.table(ndates,file="data/ndates.csv",row.names=F,col.names=F,sep=",")

### concatenate all sensors to a single raster stack (this just links the filenames)
undvi=stack(l4,l5,l7,l8)
undvi=setZ(undvi,c(getZ(l4),getZ(l5),getZ(l7),getZ(l8)))
### Sort them by date (z)
ndvi=undvi[[order(getZ(undvi))]]

ndvifile="data/ndvi_wfw.Rdata"
if(!file.exists(ndvifile)){
save(ndvi,file=ndvifile)
# ## create a new ndvi layer
# ndvifile="data/ndvi_landsat_30m.tif"
# if(!file.exists(ndvifile)){
# writeRaster(ndvi,filename=ndvifile)
# }
# ## load it
# ndvi=raster(ndvifile)

}
load(ndvifile)
```

```
yearind=which(getZ(ndvi)%in%getZ(l7)[1:5])

levelplot(ndvi[[yearind]],col.regions=cndvi()$col,cuts=length(cndvi()$at),
          at=cndvi()$at,margin=F,scales=list(draw=F),
          names.attr=getZ(ndvi)[yearind],maxpixels=1e4)
```



Data Compilation

Select domain of interest

Here we will define the subset of cells that we will explore further. In this case it is the WfW plots. You can fiddle with these settings to include fewer (or more) cells. If your computer is slow, you may want to subset this further.

load spatial data masking

```
## load data for masking
cover=raster(paste0(datadir,"clean/landcover2009_landsat_30m.gri"))

maskfile="data/mask_landsat_30m.tif"
if(!file.exists(maskfile)){
    mask=overlay(cover,fic,fun=function(x,y) x==1&y>0,filename=maskfile)
}
mask=raster(maskfile)

## load additional covariate data
tmax=raster(paste0(datadir,"clean/Tmax_jan_mean.gri"))
tmin=raster(paste0(datadir,"clean/Tmin_jul_mean.gri"))
tpi=raster(paste0(datadir,"clean/tpi500.gri"))
dem=raster(paste0(datadir,"clean/dem_landsat_30m.gri"))
janrad=raster(paste0(datadir,"clean/janrad.gri"))
julrad=raster(paste0(datadir,"clean/julrad.gri"))
aspect=raster(paste0(datadir,"clean/aspect.gri"))

### Make a dataframe of all spatial data
## Beware, this approach will only work if your data are all in identical projection/grid/etc.
maskids=extract(mask,alien_xy,cellnumbers=TRUE)
maskids=maskids[,1]

cleardat=data.frame(
  id=extract(ig, maskids),
  coordinates(ig)[maskids,],
  extract(vegclear,alien_xy) #maskids wont work here becasue vegclear is not a raster
)

sdat=data.frame(
  id=extract(ig, maskids),
  coordinates(ig)[maskids,],
  veg=extract(rvr, maskids),
  cover=extract(cover, maskids),
  tmax=extract(tmax, maskids),
  tmin=extract(tmin, maskids),
  janrad=extract(janrad, maskids),
  julrad=extract(julrad, maskids),
  aspect=extract(aspect, maskids),
  dem=extract(dem, maskids),
  tpi=extract(tpi, maskids)
 # firecount=extract(fic, maskids)
```

```
)

sdat<-cbind(sdat,alien_data@data)
kable(head(sdat))
```

id	x	y	veg	cover	tmax	tmin	janrad	julrad	aspect	dem	tpi
510630	262335	6224535	16	1	26.75148	8.804244	8564.083	3034.583	4.348300	295.9442	-5.746557
503866	261615	6224835	16	1	27.39490	8.695924	8577.000	2056.750	2.866099	310.1948	3.405393
509959	262425	6224565	16	1	27.25568	9.311305	8444.250	3097.167	4.679824	326.9142	34.642132
488295	259545	6225525	16	1	24.47923	7.349550	8333.167	2424.000	2.976255	535.4271	64.628049
508605	262245	6224625	16	1	27.00466	8.638811	8858.917	3093.083	4.129409	286.9676	-14.465683
481553	259485	6225825	16	1	25.35269	7.169548	8704.250	2204.333	2.700882	560.4637	39.230657

Reshape temporal data

It's often easier to work with data in 'long' format where there is one row for each observation and another column indicating what the observation is. Let's `melt` the data to 'long' format.

```
#extract ndvi data
ftdatw="data/tdatw.Rdata"
if(!file.exists(ftdatw)){

tdatw=data.frame(
  id=extract(ig, maskids),
#  extract(age, maskids),
  extract(ndvi,maskids)
  )
save(tdatw,file=ftdatw)
}

load(ftdatw)
kable(tdatw[1:10,1:10])
```

id	L5_198406.09	L5_198409.29	L5_198410.15	L5_198504.09	L5_198611.06	L5_198611.22	L5_198612.24
510630	0.457	0.116	-2.000	0.377	-2	-2	0.423
503866	0.439	0.103	-2.000	0.358	-2	-2	0.341
509959	0.339	0.078	-2.000	0.126	-2	-2	0.222
488295	0.682	0.074	-2.000	0.529	-2	-2	-2.000
508605	0.546	0.103	0.385	0.342	-2	-2	0.397
481553	0.509	0.055	-2.000	0.348	-2	-2	-2.000
521371	0.649	0.203	0.535	0.596	-2	-2	0.631
532137	0.629	-2.000	0.322	0.383	-2	-2	0.364
530791	0.610	-2.000	0.566	0.599	-2	-2	0.527
475386	0.523	-2.000	-2.000	0.523	-2	-2	0.377

```
tdatl=melt(tdatw,id.var="id")
tdatln=cbind.data.frame(lab=levels(tdatl$variable),
                        do.call(rbind,strsplit(as.character(levels(tdatl$variable)),"_")))
tdatln$type=ifelse(tdatln[,"1"]%in%c("L4","L5","L7","L8"),"ndvi","age")
```

```
tdatl[,c("type","date")]=tdatln[match(tdatl$variable,tdatln$lab),4:3]
tdatl$miss=paste(substr(as.character(tdatl$variable), 1, 2))

tdat=dcast(tdatl,id+date+miss~type,value.var="value")
## convert date to proper format
n = 5 #where to insert separator

tdat$date=paste(substr(as.character(tdat$date), 1, 5-1), ".", substr(as.character(tdat$date), n, nchar(a
## convert year from a factor to numeric
tdat$date=as.Date(as.character(tdat$date),"%Y.%m.%d")

kable(head(tdat))
```

id	date	miss	ndvi
319827	1984-06-09	L5	0.761
319827	1984-09-29	L5	0.269
319827	1984-10-15	L5	-2.000
319827	1985-04-09	L5	0.596
319827	1986-11-06	L5	0.498
319827	1986-11-22	L5	-2.000

Extract fire data for WfW plots and calculates age in days

```
#extract fire data for each wfw plot
fi_plot <- extract(fi,alien_xy)
#create a seq of date from earliest NDVI to most recent
alldate <- seq.Date(min(tdat$date),max(tdat$date),by="day")

firefile="data/fire.age_wfw.Rdata"
if(!file.exists(firefile)){

fire.age <- matrix(,length(alien_xy),length(alldate))

#loop through plots
for (i in 1:length(alien_xy)){
  # get fire dates for each pixel
  dates.b <- filter(fi_plot, point.ID == i) %>% select(DayDate)

  #loop through days - ouch
  for (j in 1:length(alldate)){
    if(any(!is.na(dates.b)) && ((alldate[j]-min(dates.b[,1],na.rm=T))>=0)){
    fire.diff <- alldate[j] - dates.b[,1]
    fire.diff.pos <- fire.diff[which(fire.diff>=0)]
    fire.age[i,j] <- min(fire.diff.pos)
    }
  }
}
fire.age<-as.data.frame(fire.age)
names(fire.age) <- alldate
save(fire.age,file=firefile)
```

```
}

load(firefile)


#create data frame
fi.plot.dat <- data.frame(
              id=extract(ig, alien_xy),
              fire.age)
names(fi.plot.dat) <- c("id",as.character(alldate))

fi.age<-melt(fi.plot.dat,id.vars="id")
fi.age$variable<-as.Date(fi.age$variable)
```

join temporal datasets

```
#join NDVI and age data
tdat.all<-left_join(tdat, fi.age, by = c("id" = "id", "date" = "variable"))
tdat.all.com<-tdat.all[complete.cases(tdat.all),]
## check it out
kable(head(tdat.all),row.names = F)
```

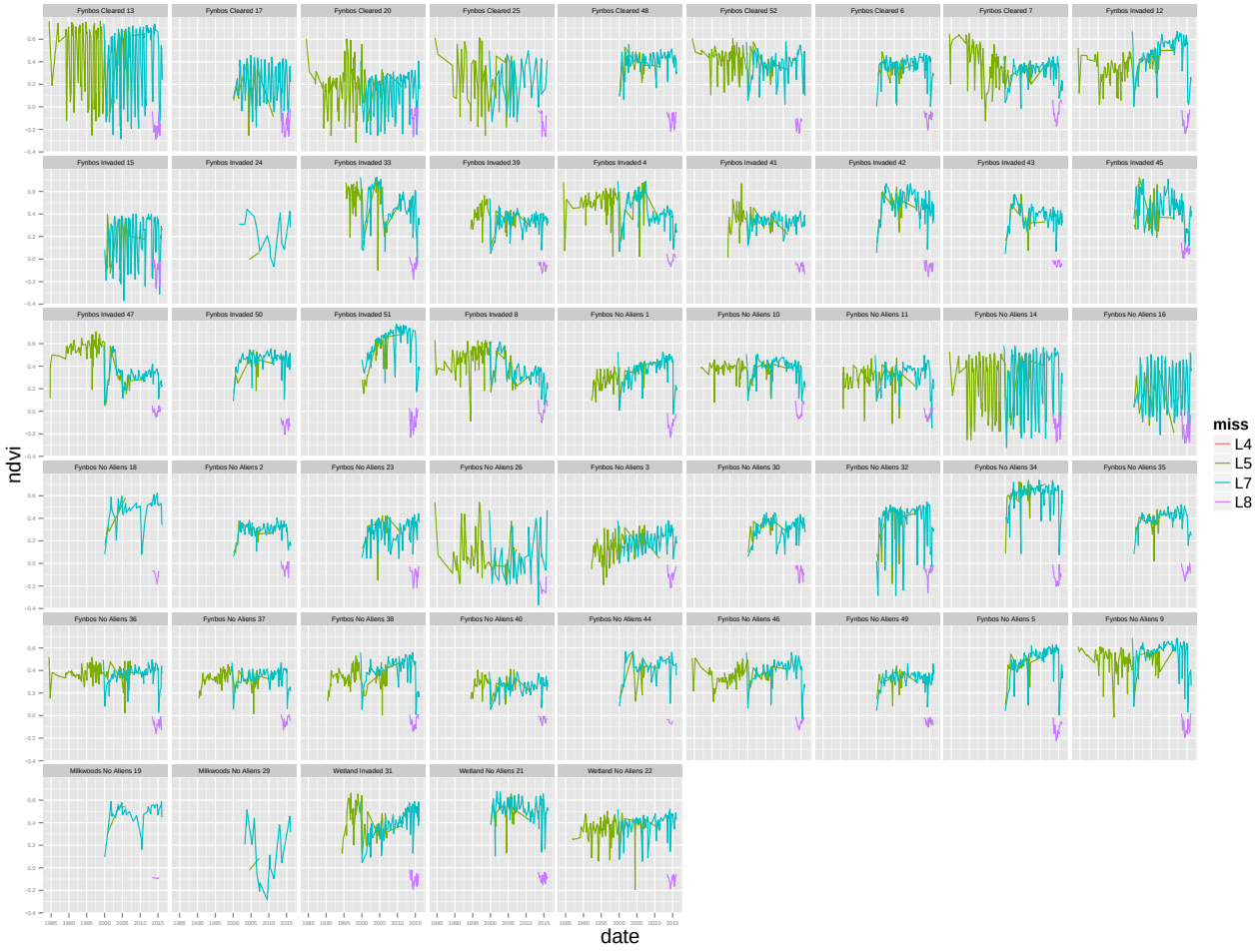
id	date	miss	ndvi	value
319827	1984-06-09	L5	0.761	NA
319827	1984-09-29	L5	0.269	NA
319827	1984-10-15	L5	-2.000	NA
319827	1985-04-09	L5	0.596	NA
319827	1986-11-06	L5	0.498	NA
319827	1986-11-22	L5	-2.000	NA

```
#join clearing data
clearsmall <- select(cleardat, id,point.ID,First_Date, First_End_, Second_End, Third_End_, Fourth_End, F
clearsmall$First_Date <- as.Date(clearsmall$First_Date, format="%Y%m%d")
clearsmall$First_End_ <- as.Date(clearsmall$First_End_, format="%Y%m%d")
clearsmall$Second_End <- as.Date(clearsmall$Second_End, format="%Y%m%d")
clearsmall$Third_End_ <- as.Date(clearsmall$Third_End_, format="%Y%m%d")
clearsmall$Fourth_End <- as.Date(clearsmall$Fourth_End, format="%Y%m%d")
clearsmall$Fifth_End_ <- as.Date(clearsmall$Fifth_End_, format="%Y%m%d")

#matrix with a row for every plot and columns up to max fires
fmat<-matrix(,nrow=length(unique(fi_plot$point.ID)),ncol= max(count(fi_plot,point.ID)[,2]))
fnum<-numeric(length(unique(fi_plot$point.ID)))

for(i in  1:length(unique(fi_plot$point.ID))){
  temp <- filter(fi_plot, point.ID == i) %>% select(DayDate)
  temp <- as.character(temp[,1])
  fmat[i,1:length(temp)]<-temp
  fnum[i] <- i
}
```

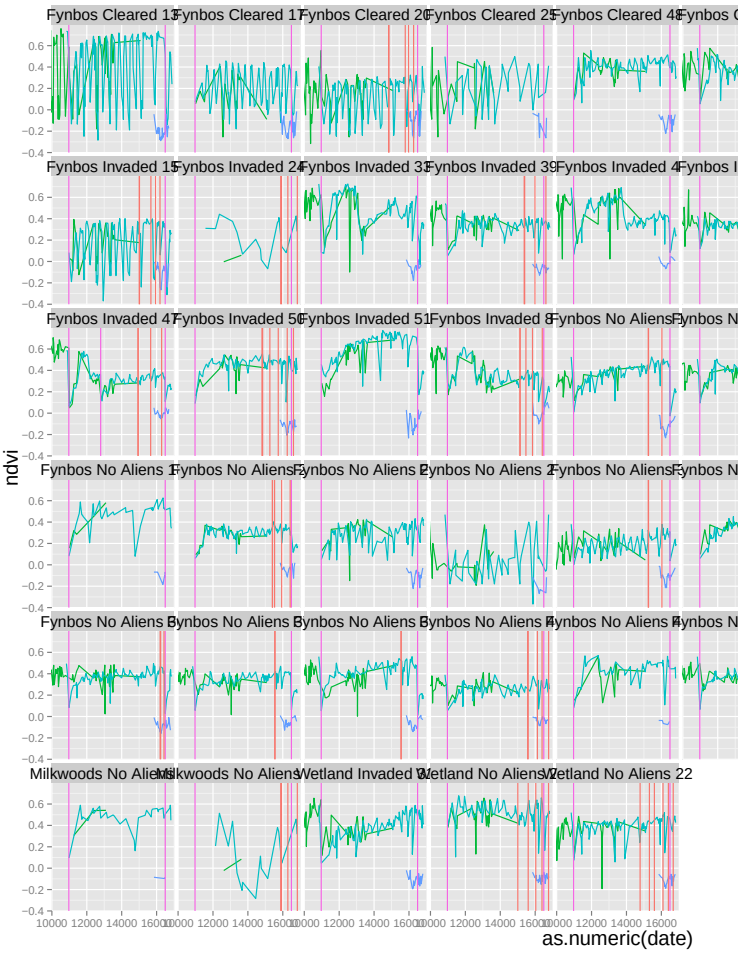
```
fmat<-as.data.frame(fmat)
fmat <- cbind(fnum,fmat)
names(fmat) <- c("point.ID","fire1","fire2","fire3","fire4")

fmat$fire1 <- as.Date(fmat$fire1, format="%Y-%m-%d")
fmat$fire2 <- as.Date(fmat$fire2, format="%Y-%m-%d")
fmat$fire3 <- as.Date(fmat$fire3, format="%Y-%m-%d")
fmat$fire4 <- as.Date(fmat$fire4, format="%Y-%m-%d")


clear.all.dat <- left_join(clearsmall,fmat,by="point.ID")
```

Create some plots

Time series by dates

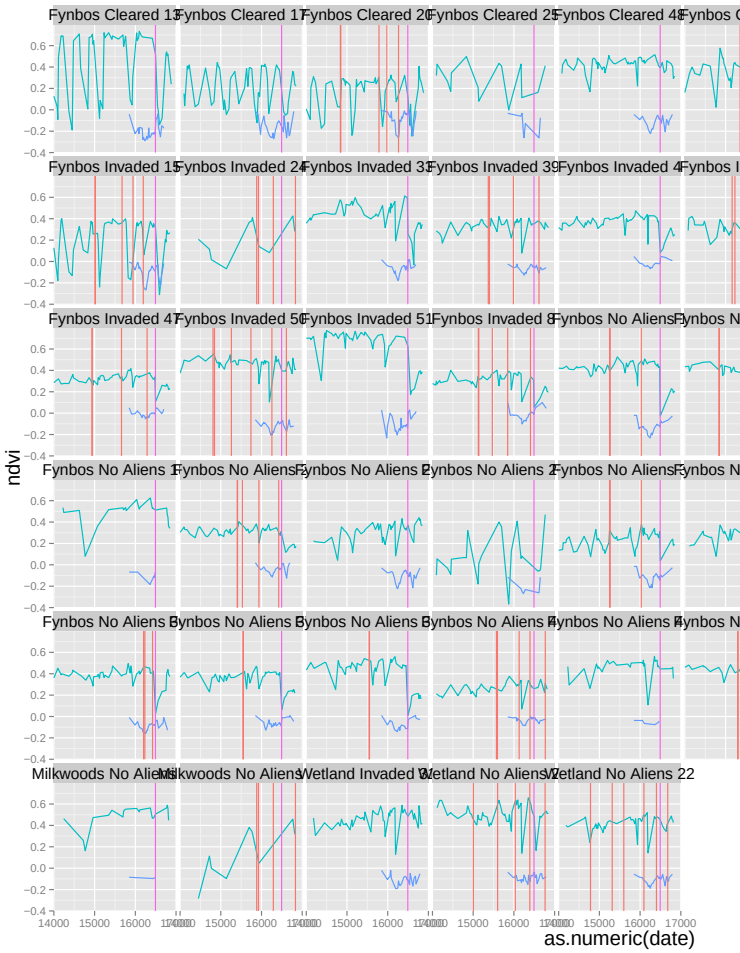


Time series with fires and clearing



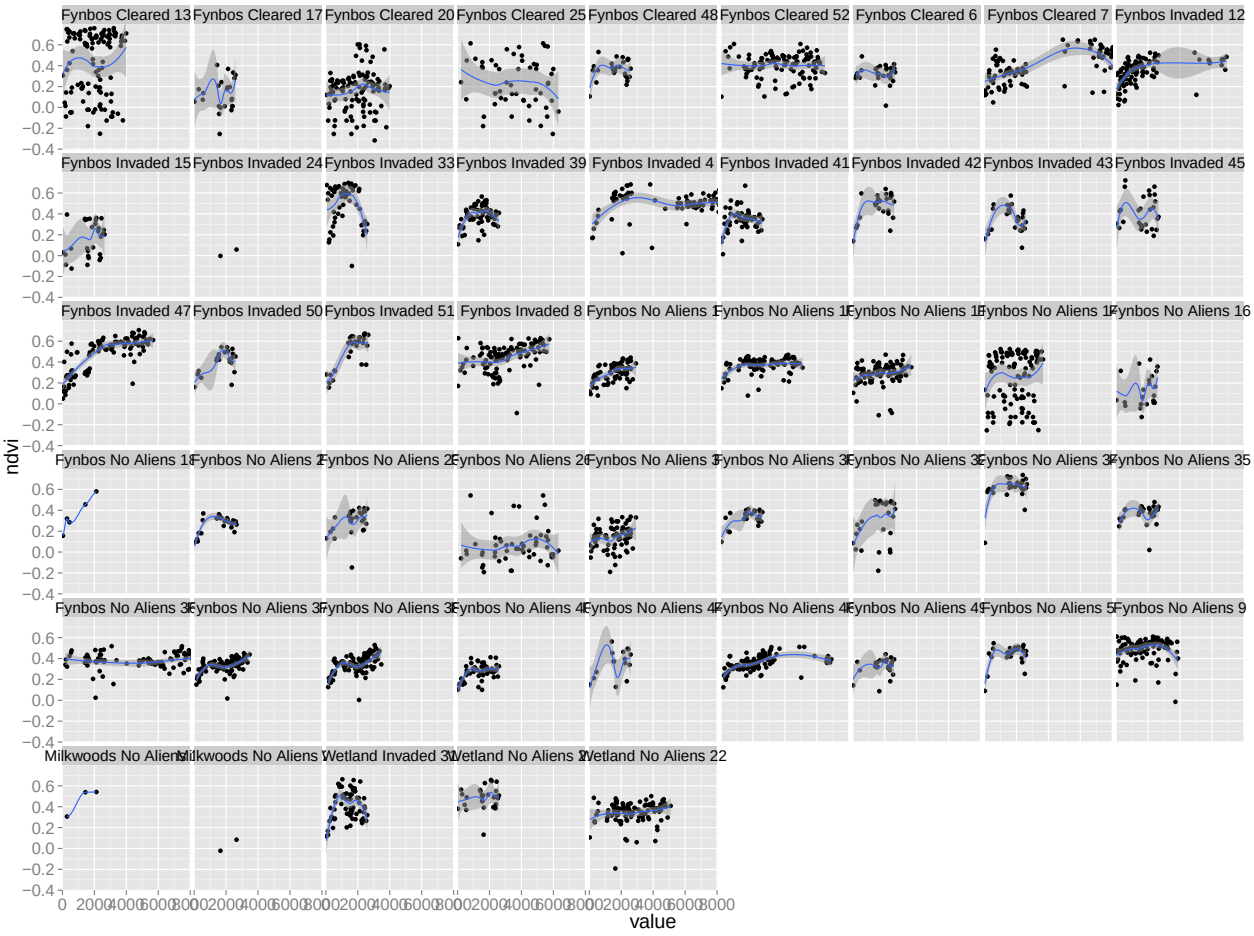
read vertical lines show clearing, purple vertical lines show fires

Time series zoomed to clearing data



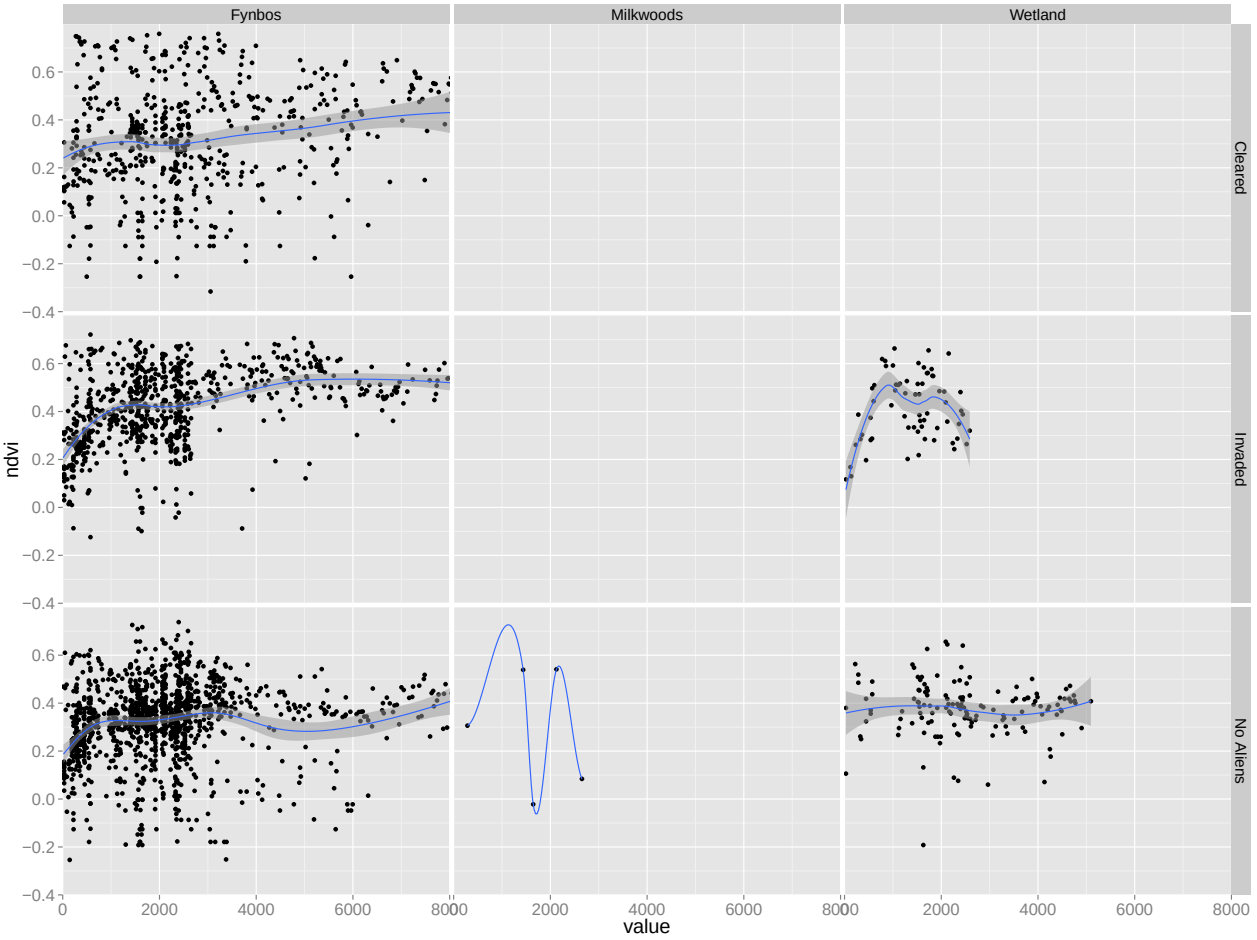
read vertical lines show clearing, purple vertical lines show fires

Recovery for all plots



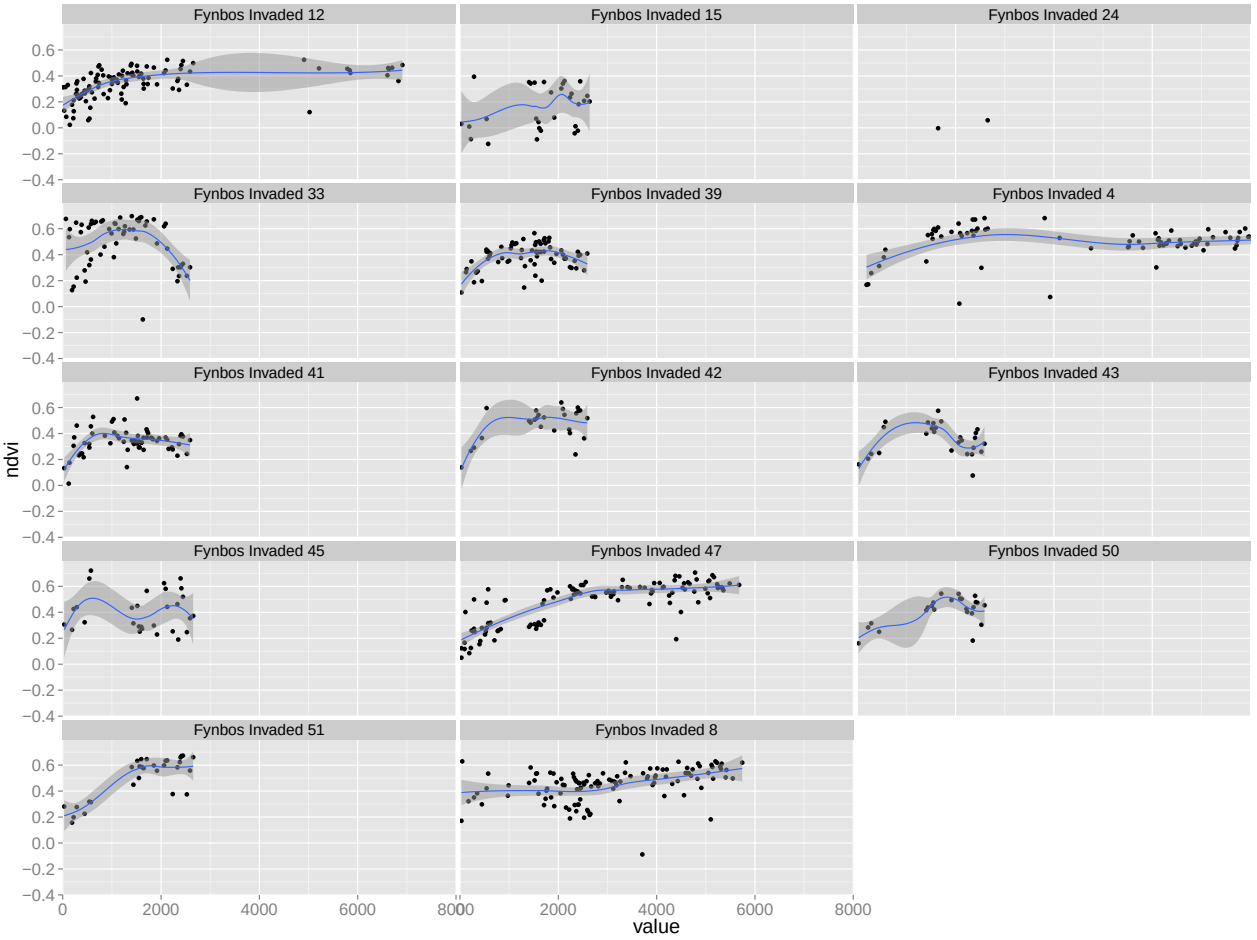
showing only L5

Recovery by vegetation type



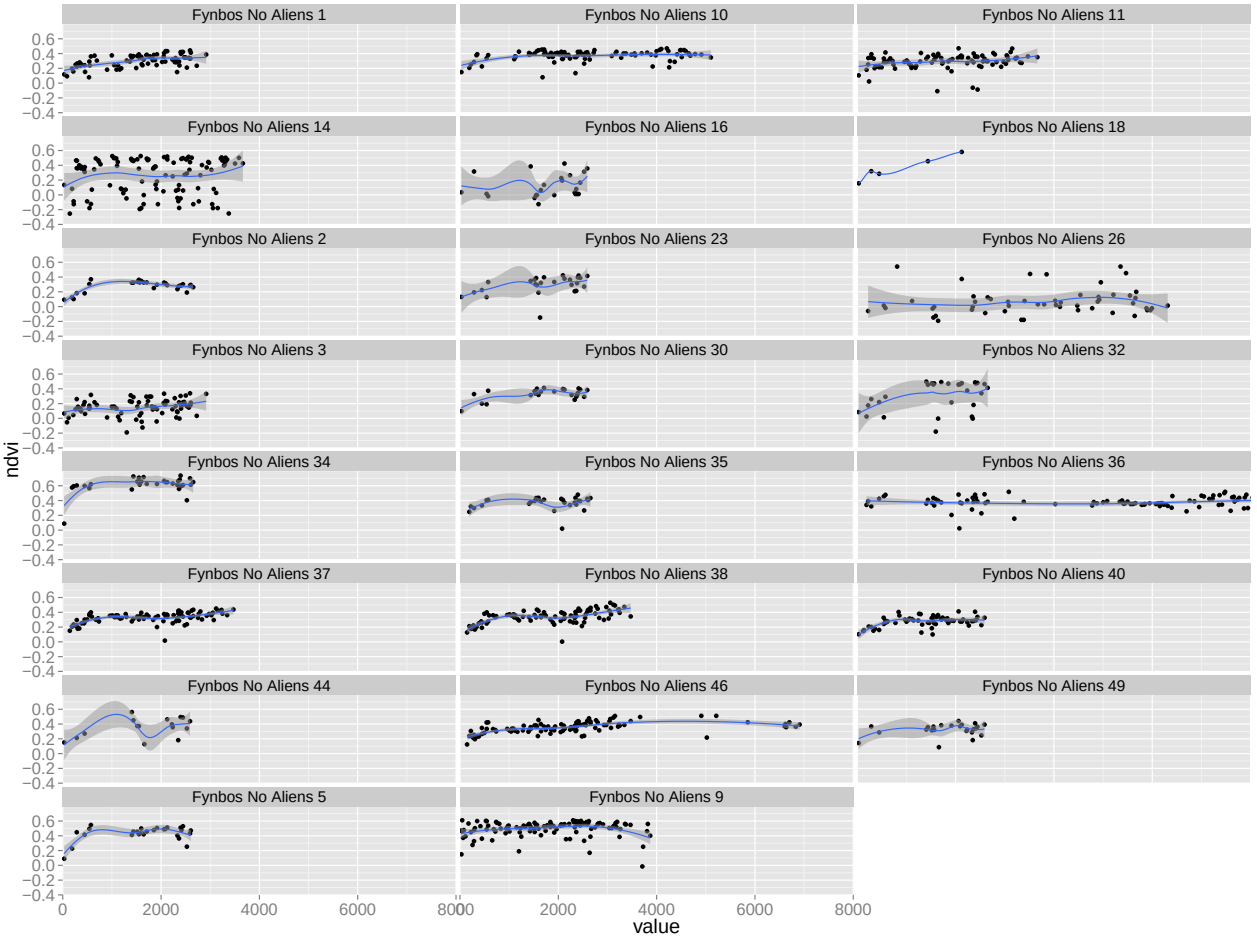
showing only L5

Fynbos invaded plots



showing only L5

Fynbos plots with no aliens



showing only L5

Save R data object for later use.

```
## drop the 'wide' version
save(clear.all.dat,tdat.all,file="data/modeldata_nofire_wfw.Rdata")
```