

# Sampling Design

author: Derek Corcoran

Last update: 2015-04-29 First we load the spatial packages

```
library("raster", lib.loc="/R/win-library/3.2")

## Loading required package: sp

library("rasterVis", lib.loc="/R/win-library/3.2")

## Loading required package: lattice
## Loading required package: latticeExtra
## Loading required package: RColorBrewer

library("maps", lib.loc="/R/win-library/3.2")
library("maptools", lib.loc="/R/win-library/3.2")

## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry      computations in maptools depend on gpclib
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()

library("rgdal", lib.loc="/R/win-library/3.2")

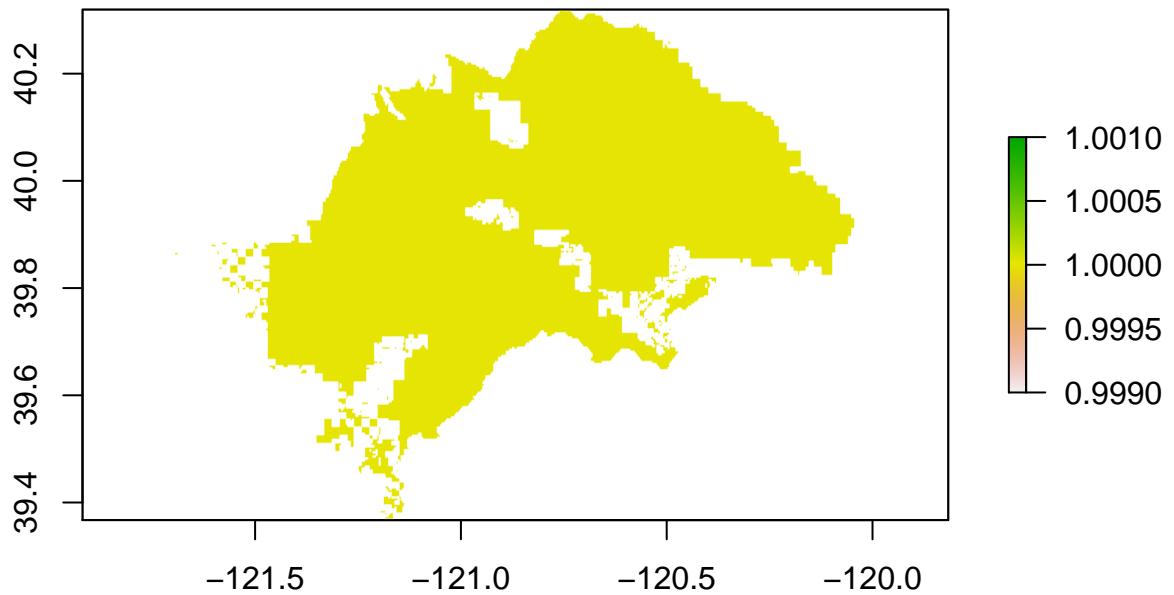
## rgdal: version: 0.9-2, (SVN revision 526)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.11.2, released 2015/02/10
## Path to GDAL shared files: C:/Users/usuario/Documents/R/win-library/3.2/rgdal/gdal
## GDAL does not use iconv for recoding strings.
## Loaded PROJ.4 runtime: Rel. 4.9.1, 04 March 2015, [PJ_VERSION: 491]
## Path to PROJ.4 shared files: C:/Users/usuario/Documents/R/win-library/3.2/rgdal/proj
```

Then we read the needed rasters

```
PNF<- readGDAL("C:/Users/usuario/Bats_California/layers/PNF.asc")

## C:/Users/usuario/Bats_California/layers/PNF.asc has GDAL driver AAIGrid
## and has 250 rows and 434 columns

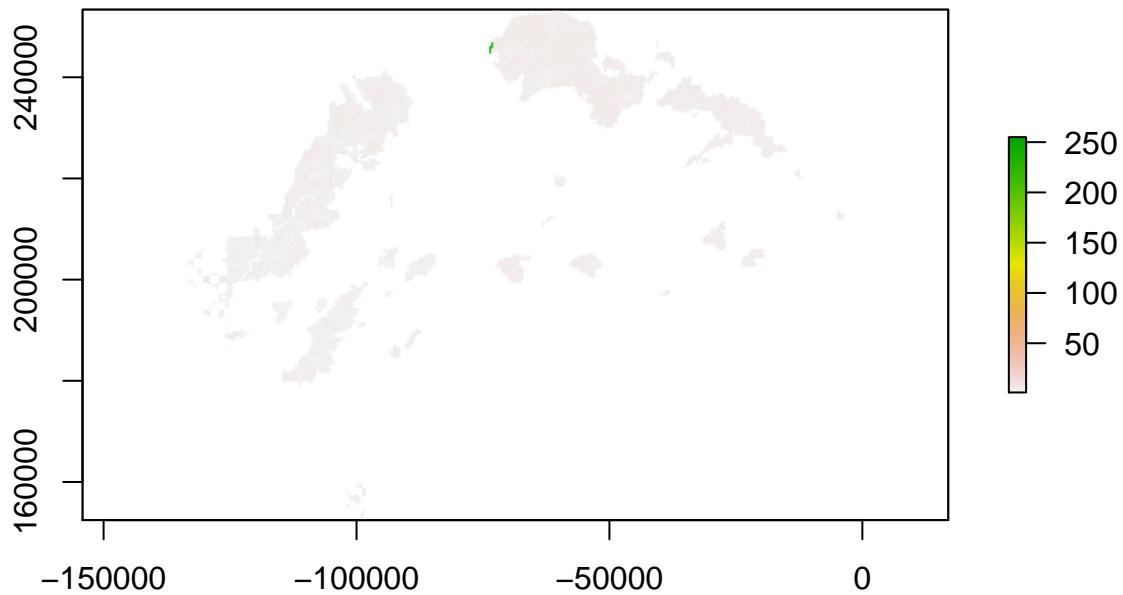
PNF<-raster (PNF)
plot(PNF)
```



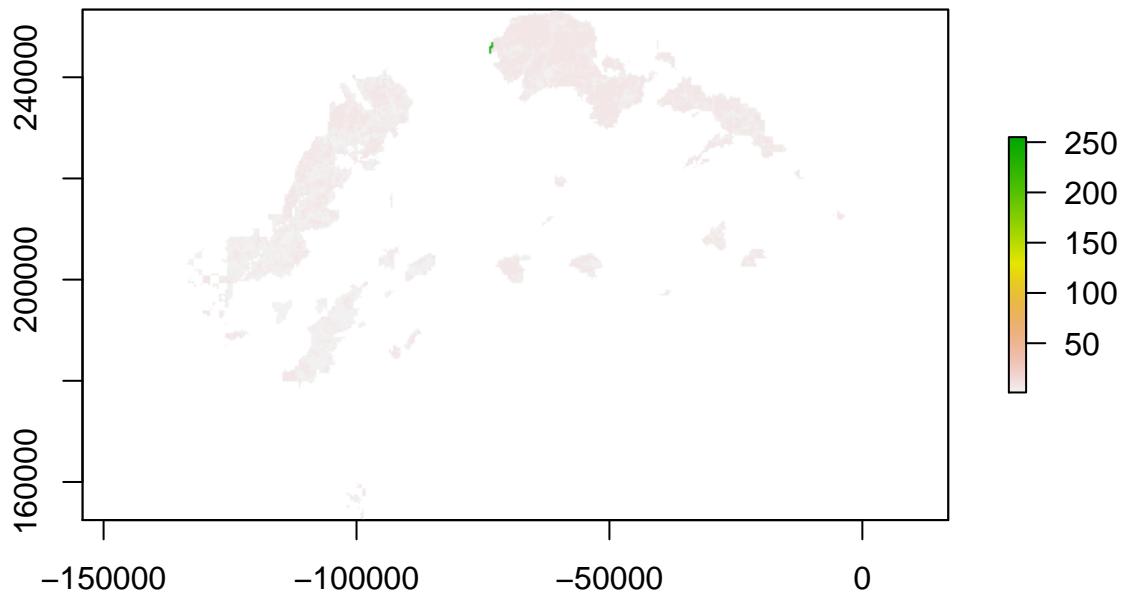
```
bc <- readGDAL("C:/Users/usuario/Bats_California/layers/burn_canopy.asc")
```

```
## C:/Users/usuario/Bats_California/layers/burn_canopy.asc has GDAL driver AAIGrid  
## and has 250 rows and 322 columns
```

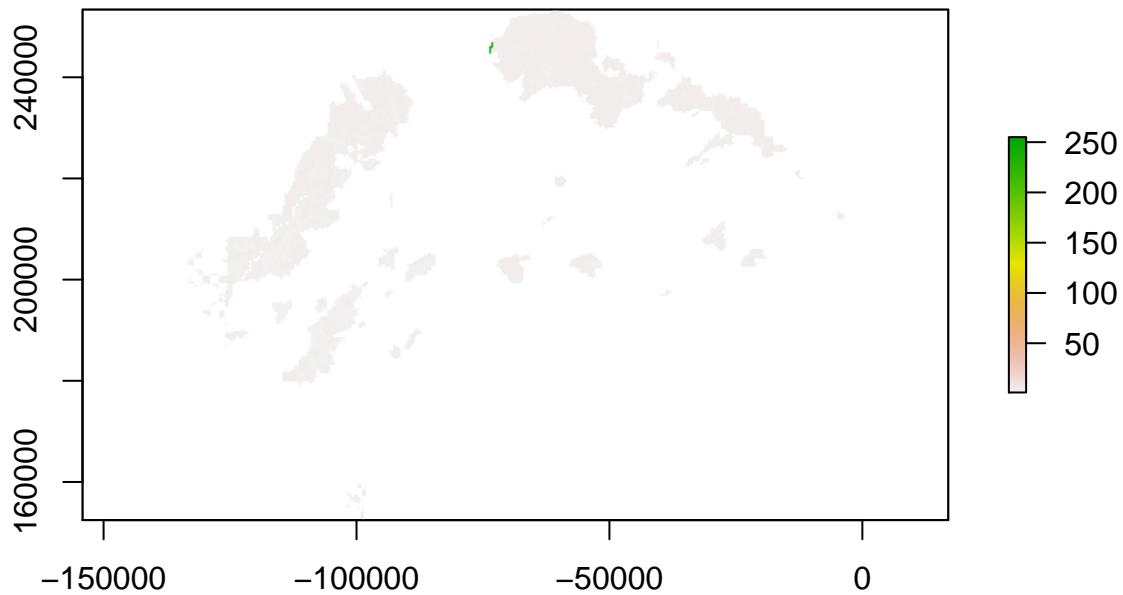
```
bc<-raster (bc)  
plot(bc)
```



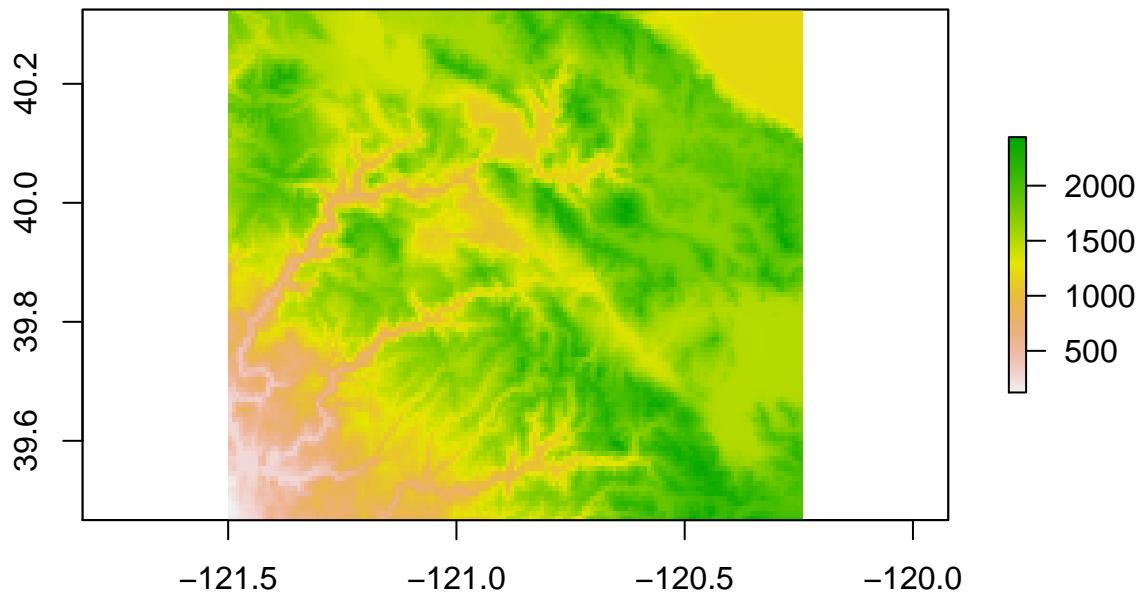
```
bb <- readGDAL("C:/Users/usuario/Bats_California/layers/burn_basal.asc")  
  
## C:/Users/usuario/Bats_California/layers/burn_basal.asc has GDAL driver AAIGrid  
## and has 250 rows and 322 columns  
  
bb<-raster (bb)  
plot(bb)
```



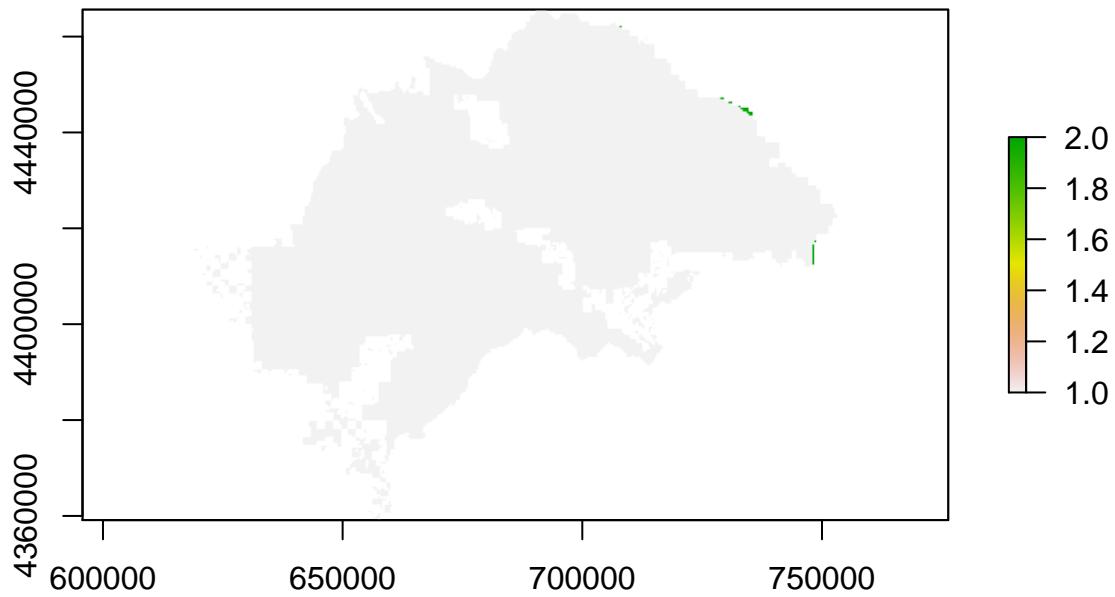
```
bs <- readGDAL("C:/Users/usuario/Bats_California/layers/burn_severity.asc")  
  
## C:/Users/usuario/Bats_California/layers/burn_severity.asc has GDAL driver AAIGrid  
## and has 250 rows and 322 columns  
  
bs<-raster (bs)  
plot(bs)
```



```
topo <- readGDAL("C:/Users/usuario/Bats_California/layers/plumastopo.asc")  
  
## C:/Users/usuario/Bats_California/layers/plumastopo.asc has GDAL driver AAIGrid  
## and has 103 rows and 151 columns  
  
topo<-raster (topo)  
plot(topo)
```

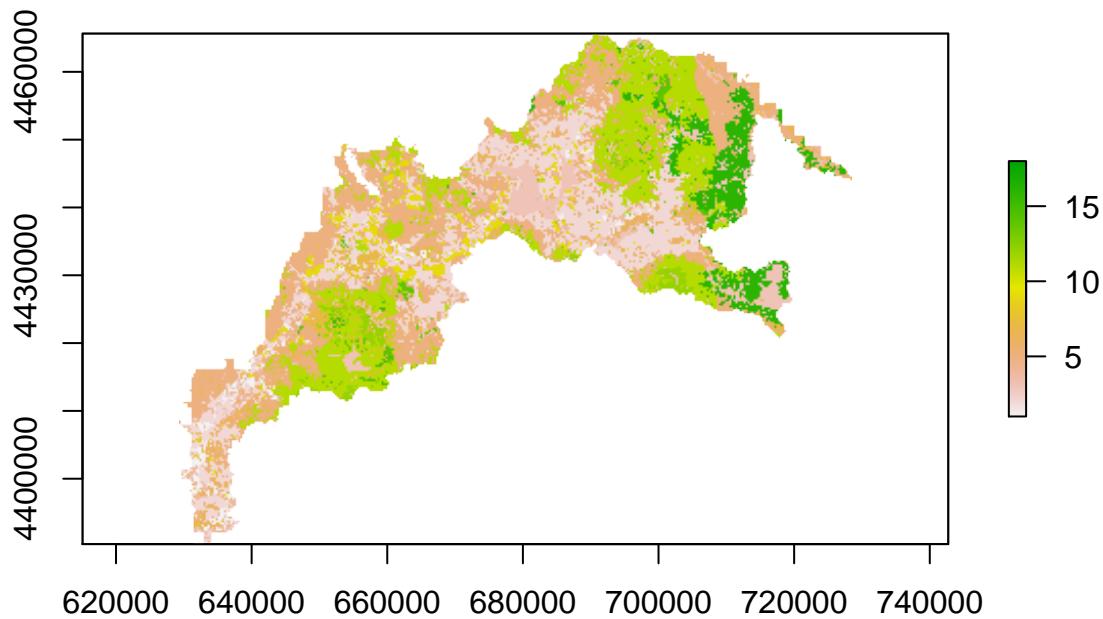


```
Vegetation_existing <- readGDAL("C:/Users/usuario/Bats_California/layers/Vegetation_existing.asc")  
  
## C:/Users/usuario/Bats_California/layers/Vegetation_existing.asc has GDAL driver AAIGrid  
## and has 250 rows and 314 columns  
  
Vegetation_existing<-raster (Vegetation_existing)  
plot(Vegetation_existing)
```

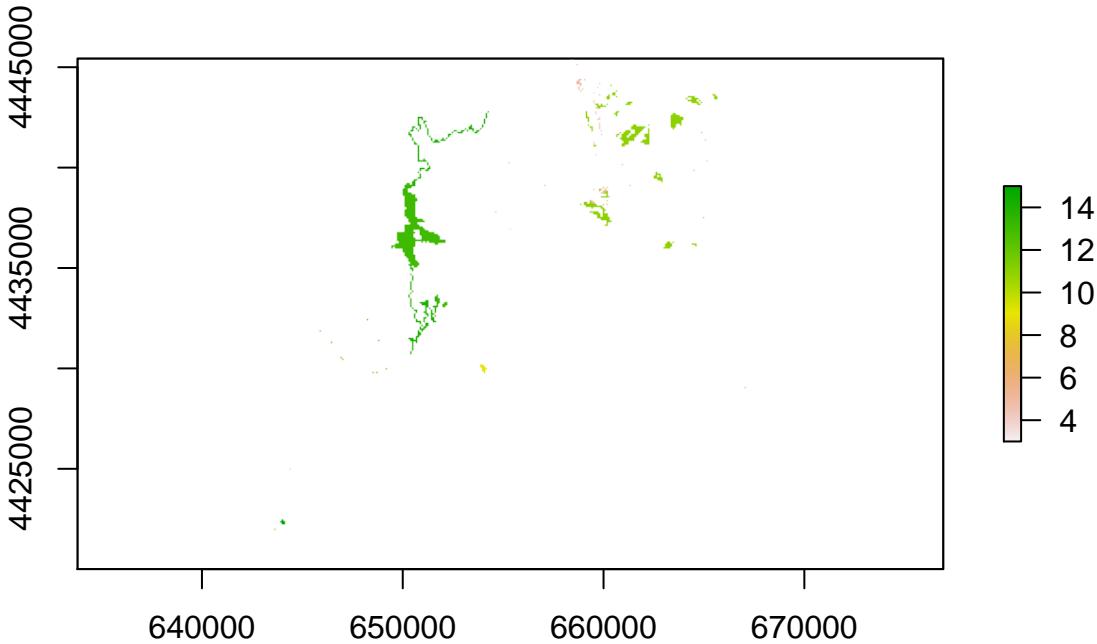


```
FireReturnIntervalDeparture <- readGDAL("C:/Users/usuario/Bats_California/layers/FireReturnIntervalDeparture.asc")
## C:/Users/usuario/Bats_California/layers/FireReturnIntervalDeparture.asc has GDAL driver AAIGrid
## and has 250 rows and 329 columns

FireReturnIntervalDeparture<-raster (FireReturnIntervalDeparture)
plot(FireReturnIntervalDeparture)
```



```
TreatmentsStorrie <- readGDAL("C:/Users/usuario/Bats_California/layers/TreatmentsStorrie.asc")  
  
## C:/Users/usuario/Bats_California/layers/TreatmentsStorrie.asc has GDAL driver AAIGrid  
## and has 271 rows and 250 columns  
  
TreatmentsStorrie<-raster (TreatmentsStorrie)  
plot(TreatmentsStorrie)
```



## Change outlayers and extract NAs

In order to classify the raster we will get rid of unnecessary outlayers, and change NAs to 0

```
df.bb <- data.frame(id=c(NA,1,2,3,4,5,6,7,255), v=c(0,1,2,3,4,5,6,7,8))
bb1 <- subs(bb, df.bb, subswithNA=FALSE)
df.bs <- data.frame(id=c(NA,1,2,3,4,255), v=c(0,1,2,3,4,5))
bs1 <- subs(bs, df.bs, subswithNA=FALSE)
df.bc <- data.frame(id=c(NA,1,2,3,4,5,255), v=c(0,1,2,3,4,5,6))
bc1 <- subs(bc, df.bc, subswithNA=FALSE)
```

## Put all rasters in the same projection

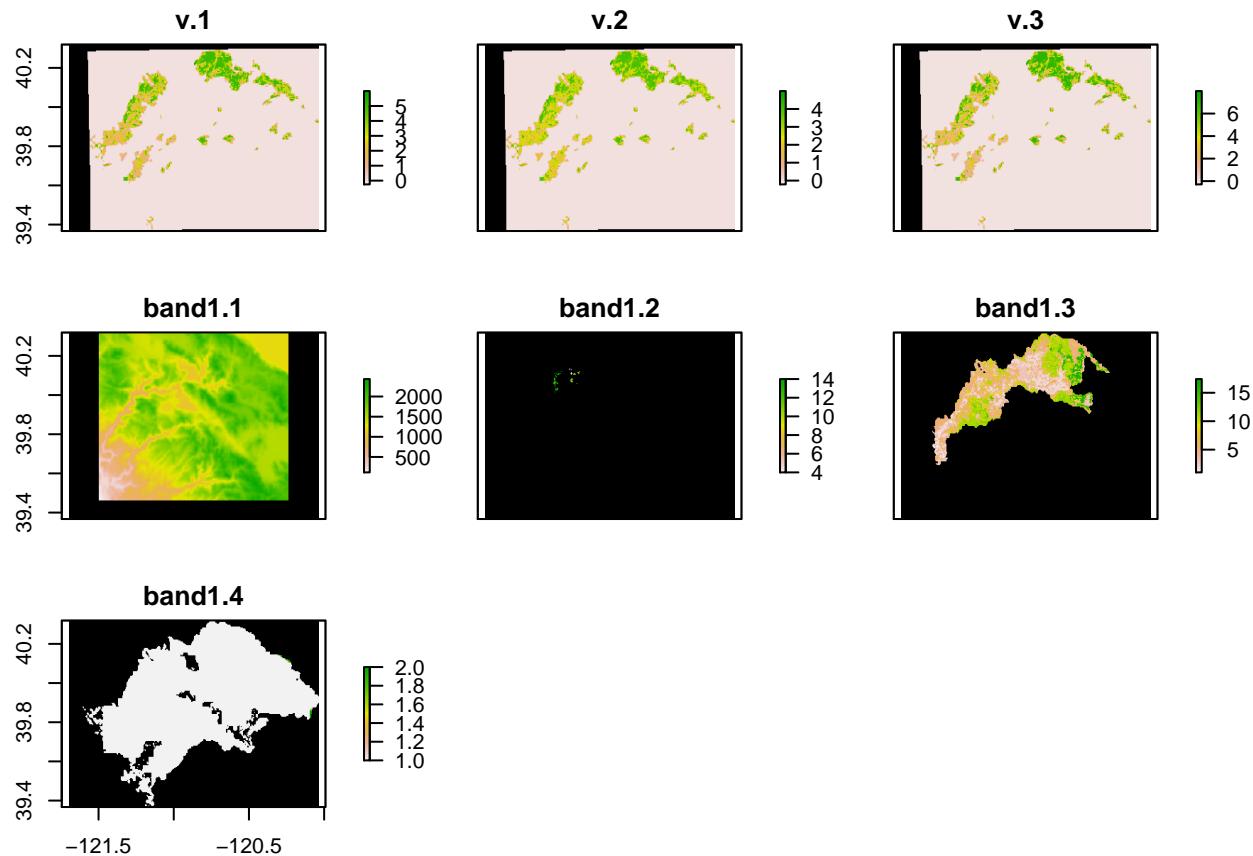
```
bc<-projectRaster(bc1, PNF)
bb<-projectRaster(bb1, PNF)
bs<-projectRaster(bs1, PNF)
Vegetation_existing<-projectRaster(Vegetation_existing, PNF)
FireReturnIntervalDeparture<-projectRaster(FireReturnIntervalDeparture, PNF)
TreatmentsStorrie<-projectRaster(TreatmentsStorrie, PNF)
```

Put them all in the same resolution and size

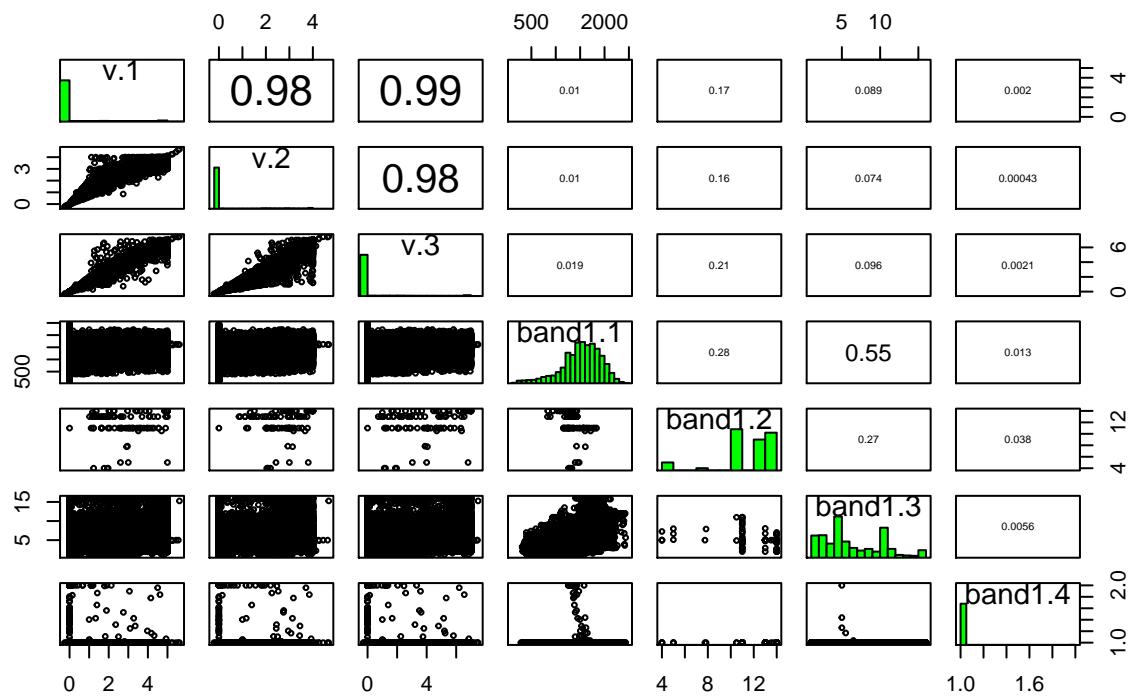
```
bc<-resample(bc, PNF)
bb<-resample(bb, PNF)
bs<-resample(bs, PNF)
Vegetation_existing<-resample(Vegetation_existing, PNF)
FireReturnIntervalDeparture<-resample(FireReturnIntervalDeparture, PNF)
TreatmentsStorrie<-resample(TreatmentsStorrie, PNF)
topo<-resample(topo,PNF)
```

Check for correlation between rasters

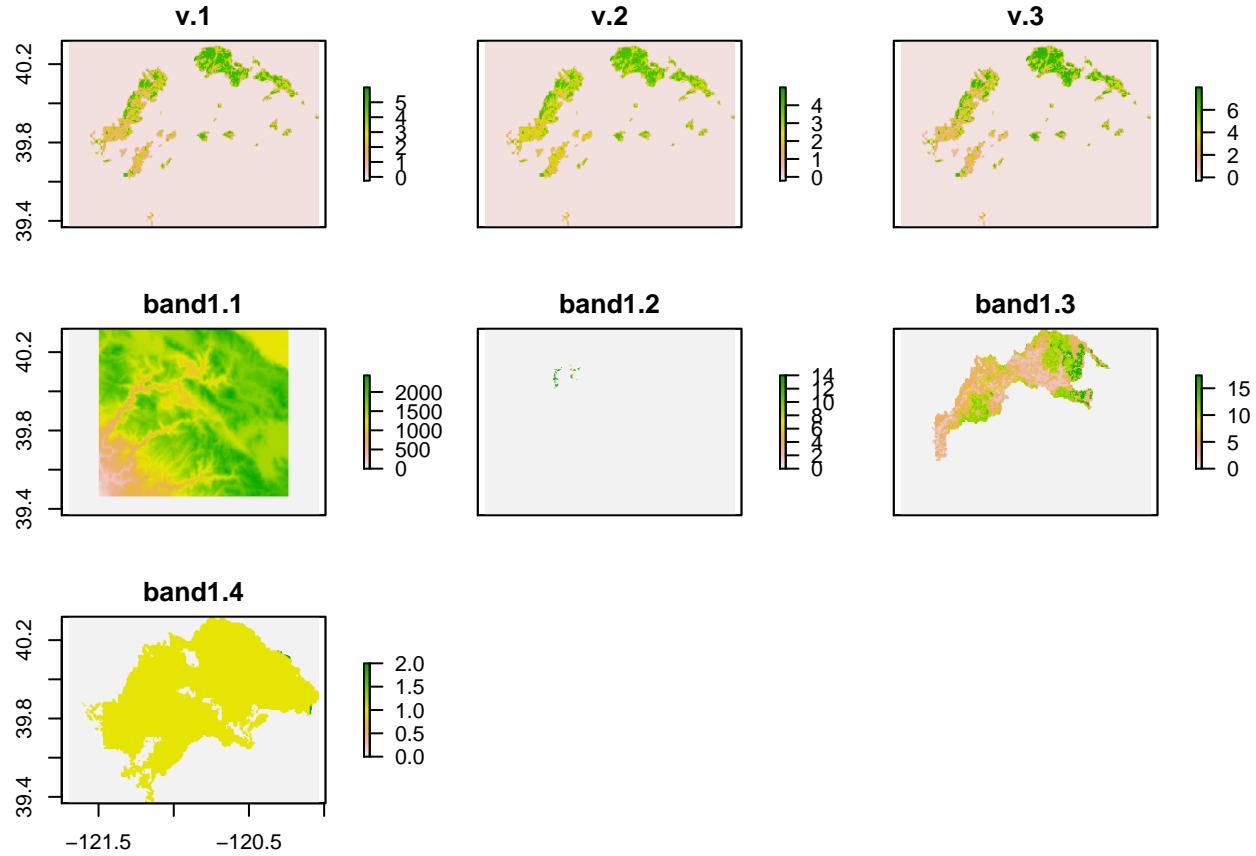
```
burn <- stack(bc1, bs1, bb1)
AllLayers <-stack(bc, bs, bb, topo,TreatmentsStorrie,FireReturnIntervalDeparture, Vegetation_existing)
plot (AllLayers, colNA="black")
```



```
pairs(AllLayers)
```



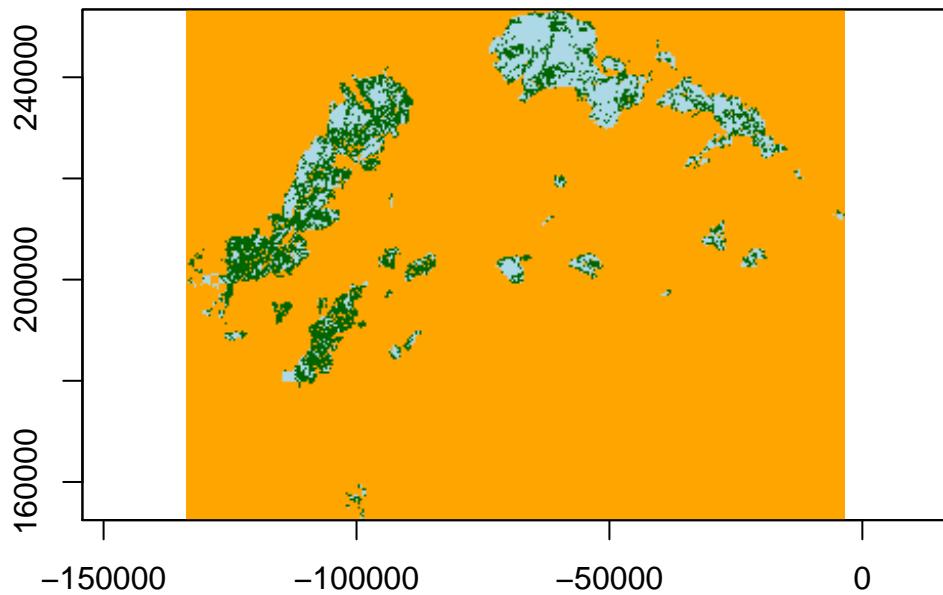
```
AllLayers[is.na(AllLayers)] <- 0
plot(AllLayers)
```



## Clasification example

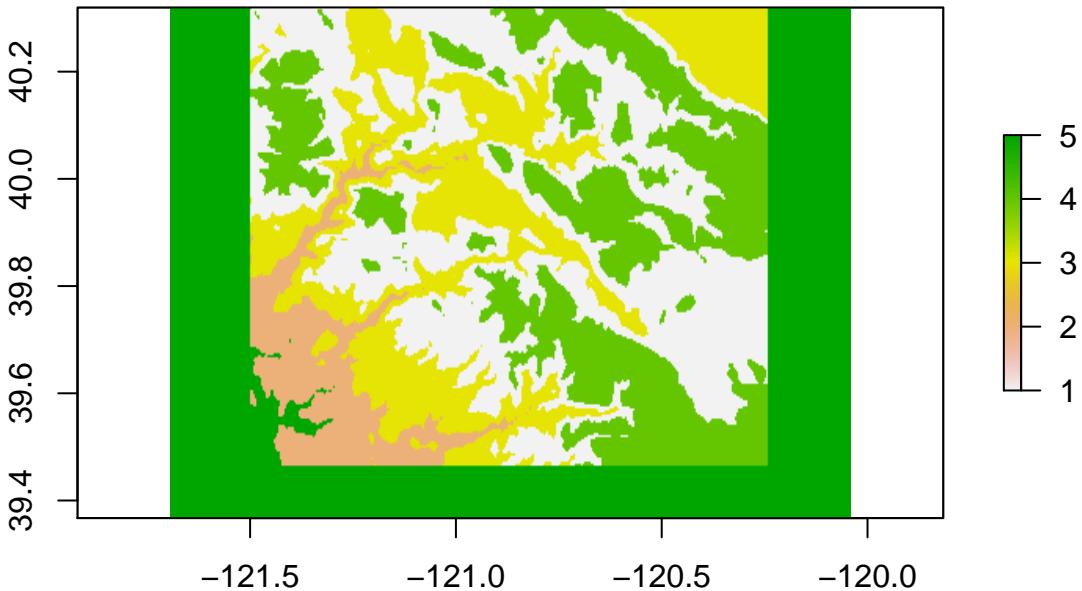
Even though we shouldn't classify using 3 layers of such high classification we will use the RasterBrick of the three burn classifications to exemplify how we will divide the area into areas of similar characteristics. Here we will ask R to use kmeans to sort the area into 3 types of habitat using the abovementioned rasterbrick:

```
##      v.1 v.2 v.3
## [1,]  0  0  0
## [2,]  0  0  0
## [3,]  0  0  0
## [4,]  0  0  0
## [5,]  0  0  0
## [6,]  0  0  0
```



now with every layer

```
##      v.1 v.2 v.3 band1.1 band1.2 band1.3 band1.4
## [1,]    0   0   0      0      0      0      0
## [2,]    0   0   0      0      0      0      0
## [3,]    0   0   0      0      0      0      0
## [4,]    0   0   0      0      0      0      0
## [5,]    0   0   0      0      0      0      0
## [6,]    0   0   0      0      0      0      0
```



```
##          layer
## Min.      1
## 1st Qu.   1
## Median   4
## 3rd Qu.   5
## Max.     5
## NA's     0
```

More info on how to do this classification in <https://geoscripting-wur.github.io/AdvancedRasterAnalysis/>

## Extract Random points from each habitat type

```
df.class.4 <- data.frame(id=c(1,2,3,4,5), v=c(NA,NA,NA,4,NA))
class4 <- subs(classes2, df.class.4, subswithNA=FALSE)
points4<-sampleRandom(class4,10, na.rm=TRUE, xy=TRUE)
df.class.3 <- data.frame(id=c(1,2,3,4,5), v=c(NA,NA,3,NA,NA))
class3 <- subs(classes2, df.class.3,subswithNA=FALSE)
points3<-sampleRandom(class3,10, na.rm=TRUE, xy=TRUE)
plot(classes2)
points (points4, col= "black")
points (points3, col= "red")
```

