Restoration Prioritization

Melanie Dickie

This repository documents the process used to prioritize areas for caribou habitat restoration in boreal Alberta. The spatial layers required to prioritize across all boreal caribou ranges in Alberta are large, and are not well suited to analyses within R and sharing on GitHub. We therefore provide data pre-processed using ArcGIS. The process can be used to include additional prioritization criteria, modify weightings, or adapt to other systems.

## Load needed packages

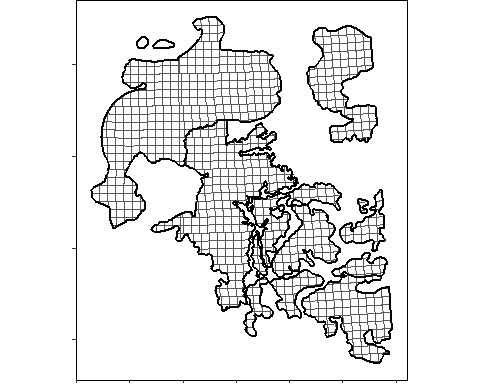
library(sf)  
library(raster)  
library(here)  
library(mapview)  
library(ggplot2)  
library(viridis)  
library(plyr)  
library(dplyr)

## Set Up Area of Interest

Fist, identify landscape units to be prioritized

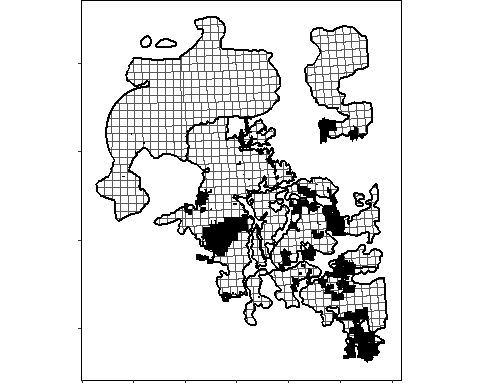
We prioritized areas within Alberta’s boreal caribou ranges. We used townships as landscape units. Alberta has a gridded system of approximately 6 by 6 mile “townships”, which represent an appropriate scale for our landscape units.

Twps<-st\_read(here::here("CleanInput\_Data", "PrioritizationData.shp"))  
Range<-st\_read(here::here("CleanInput\_Data", "ABCaribouRanges.shp"))  
Range<-subset(Range, RANGE == "Cold Lake" | RANGE == "ESAR" | RANGE== "WSAR" |   
 RANGE == "Red Earth" | RANGE == "Richardson")  
  
ggplot() +  
 geom\_sf(data = Twps, fill =NA, size=0.5) +  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



Some areas are not feasible to conduct habitat restoration, and should be excluded from all calculations. For example, we removed areas within the energy sector’s oil and gas project boundaries that are being actively developed, are planned for development in the near future, or are managed under separate regulatory processes. We use the Alberta Energy Regulator’s Scheme Approval mapper (<https://extmapviewer.aer.ca/AERSchemeApprovalArea/Index.html>), and collaborated with Canada’s Oil Sands Innovation Alliance to confirm the boundaries.

Scheme<-st\_read(here::here("CleanInput\_Data", "OS\_Project\_Boundaries\_2021-caribouRg.shp"))  
#These scheme boundaries are clipped out of the townships during calculations  
  
ggplot() +  
 geom\_sf(data = Twps, fill =NA, size=0.5) +  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+  
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())

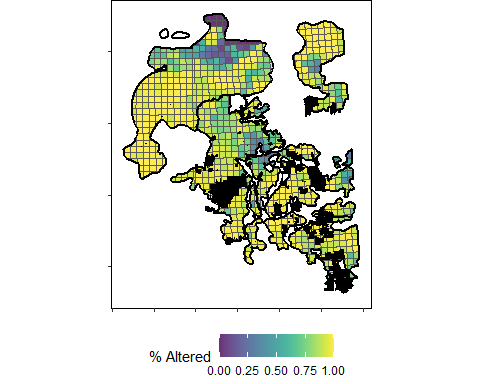


## Set Up “Benefit” layers

The primary benefit was defined as the gain in unaltered caribou habitat Gain in Unaltered habitat was calculated as: current % altered (ABMI Human Footprint Inventory, buffered by 500m) - future % altered following restoration (ABMI Human Footprint Inventory with seismic lines removed, buffered by 500m). ABMI’s Human Footprint Inventory is a very large shapefile, so cannot be shown here. Full data can be accessed at <https://abmi.ca/home/data-analytics/da-top/da-product-overview/Human-Footprint-Products/HF-inventory.html>. When calculating the Gain in Unaltered habitat, make sure input alteration layers are clipped larger than the area of interest (i.e. buffer by 500m) to account for alteration outside of caribou range that have a buffer extending into the range.

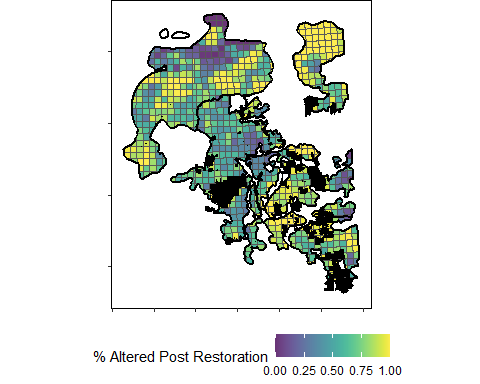
## Calculate Current % Altered

ggplot() +  
 geom\_sf(data = Twps, aes(fill =P\_d\_wf), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "% Altered")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+  
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



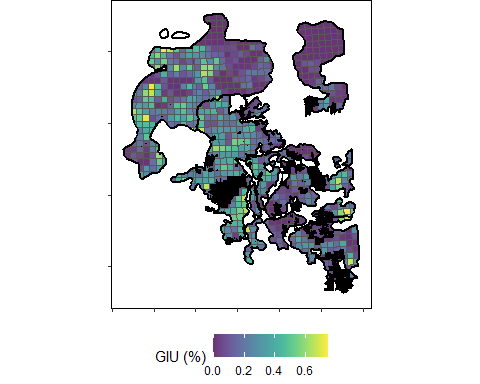
## Calculate % Altered Following Restoration

ggplot() +  
 geom\_sf(data = Twps, aes(fill =P\_d\_nswf), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "% Altered Post Restoration")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+  
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



## Calculate Gain in Unaltered Habitat

##Subtract % altered following restoration from current % altered  
Twps$GIU\_area<-Twps$A\_d\_wf-Twps$A\_d\_nswf  
Twps$GIU<-Twps$P\_d\_wf-Twps$P\_d\_nswf  
  
##Rounding errors when dissolving create some townships with very small "decreases" in gain in unaltered post restoration.  
##These are rounding errors and are set to zero.  
Twps$GIU\_area[Twps$GIU\_area<0] <- 0  
Twps$GIU[Twps$GIU<0] <- 0  
  
ggplot() +  
 geom\_sf(data = Twps, aes(fill=as.numeric(GIU)), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "GIU (%)")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())

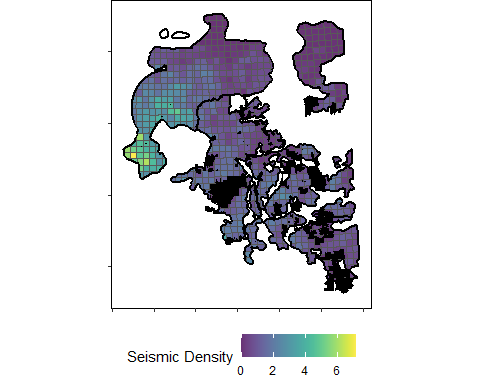


## Import “Cost” layers

We estimated cost using the density of conventional seismic lines to be restored. Seismic lines are the most common feature restored within caribou range. We used density as a metric of cost instead of an actual dollar value because the actual cost varies based on additional operational decisions that can not be incorporated effectively at this scale. Operational level decisions are described within the manuscript.

Conventional seismic line density was calculated using the ABMI Human Footprint Inventory Linear Feature layer

ggplot() +  
 geom\_sf(data = Twps, aes(fill=as.numeric(SeisDens)), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "Seismic Density")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



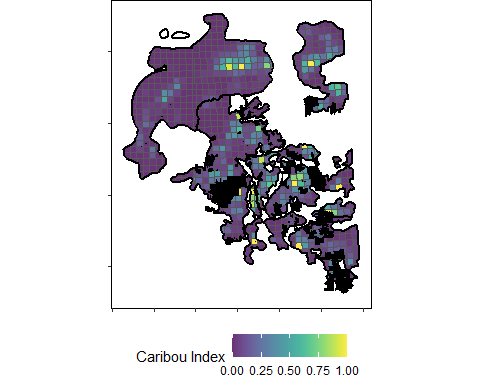
## Set Up “Weighting” layers

We down-weighted areas likely to be altered in the future, but increased the weight of areas of high value to caribou.

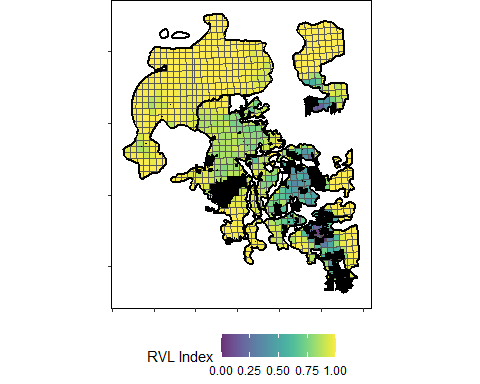
We assumed areas with high values in the Resource Value Layer (RVL) are more likely to be altered in the future. Therefore, we weighted townships by the inverse rank of the normalized RVL using summarized RVL from CAPP (2016). RVL data are confidential, and as such only the index is available.

We also assumed that areas with high caribou use, from GPS locations, are of higher value to caribou. Again, caribou use data are confidential, and only the index is available.

ggplot() +  
 geom\_sf(data = Twps, aes(fill=as.numeric(C\_Index)), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "Caribou Index")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



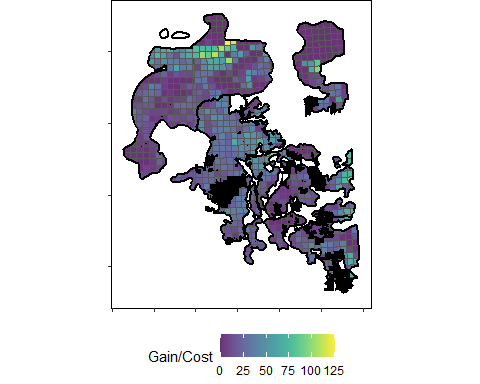
ggplot() +  
 geom\_sf(data = Twps, aes(fill=as.numeric(RVL\_Index)), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "RVL Index")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



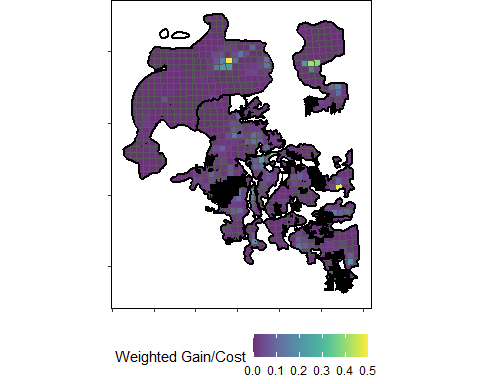
## Rank Townships For Restoration

Now we can calculate the Gain in Unaltered habitat by cost, weight by RVL and caribou indices, and rank into 5 equal bins.

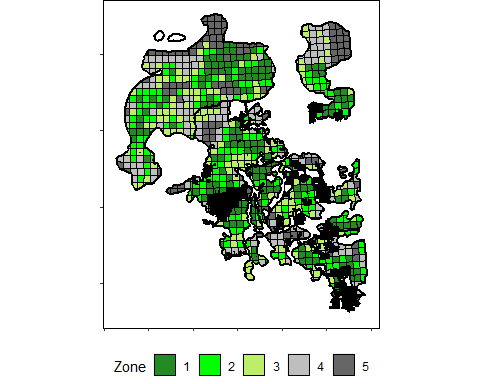
## Calculate "Gain Per Cost", or GIU / Cost  
Twps$GPC<-(Twps$GIU\_area/Twps$TWP\_area\*100)/Twps$SeisDens  
#As above, rounding errors and no GIU set to zero  
Twps$GPC[is.na(Twps$GPC)] <- 0  
Twps$GPC[!is.finite(Twps$GPC)] <- 0  
  
ggplot() +  
 geom\_sf(data = Twps, aes(fill=as.numeric(GPC)), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "Gain/Cost")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



## Weight by RVL and caribou use  
##First, normalize GPC so all three are equally weighted (all on a scale from 0 to 1)  
Twps$GPCNorm = (Twps$GPC-min(Twps$GPC))/(max(Twps$GPC)-min(Twps$GPC))  
Twps$GPC\_RVL<-Twps$GPCNorm\*Twps$RVL\_Index  
Twps$GPC\_RVL\_C<-Twps$GPC\_RVL\*Twps$C\_Index  
  
ggplot() +  
 geom\_sf(data = Twps, aes(fill=as.numeric(GPC\_RVL\_C)), size=0.5) +  
 scale\_fill\_viridis(alpha=0.8, name = "Weighted Gain/Cost")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



## Divide into zones  
Zones <-ddply (Twps, .(RANGE), mutate, Zones = 6 - ntile(GPC\_RVL\_C,5))  
Twps$Zone<-Zones$Zones[match(Twps$RANGE\_LINK , Zones$RANGE\_LINK)]  
Twps$Zone<-as.factor(Twps$Zone)  
ggplot() +  
 geom\_sf(data = Twps, aes(fill=Zone), colour="black", size=0.5) +  
 scale\_fill\_manual(values=c("forestgreen","green1", "darkolivegreen2","gray", "gray40"), name="Zone")+  
 geom\_sf(data = Range, fill=NA, size=1, colour="black")+  
 geom\_sf(data = Scheme, fill = "black", colour=NA)+   
 theme(panel.grid.major = element\_blank(),  
 panel.background = element\_rect(fill = "white"),   
 panel.border = element\_rect(fill = NA),  
 axis.text = element\_blank(),  
 axis.title = element\_blank(),  
 plot.margin=unit(c(0,0,0,0),"mm"),  
 legend.position = "bottom",  
 legend.background = element\_blank())



#st\_write(Twps,here::here("CleanInput\_Data", "PrioritizationZones.shp"))

## Calculate Area Altered as Restoration Progresses

##Zone 1:  
Twps$NZone<-as.numeric(Twps$Zone)  
RestFireZ1<-bind\_rows(aggregate(subset(Twps, NZone>1)$A\_d\_wf, by=list(Range=subset(Twps, NZone>1)$RANGE), FUN=sum), aggregate(subset(Twps, NZone==1)$A\_d\_nswf, by=list(Range=subset(Twps, NZone==1)$RANGE), FUN=sum)) %>%  
 group\_by(Range) %>%  
 summarise\_all(funs(sum(., na.rm = TRUE)))  
##Zone 2:  
RestFireZ2<-bind\_rows(aggregate(subset(Twps, NZone>2)$A\_d\_wf, by=list(Range=subset(Twps, NZone>2)$RANGE), FUN=sum), aggregate(subset(Twps, NZone<=2)$A\_d\_nswf, by=list(Range=subset(Twps, NZone<=2)$RANGE), FUN=sum)) %>%  
 group\_by(Range) %>%  
 summarise\_all(funs(sum(., na.rm = TRUE)))  
##Zone 3:  
RestFireZ3<-bind\_rows(aggregate(subset(Twps, NZone>3)$A\_d\_wf, by=list(Range=subset(Twps, NZone>3)$RANGE), FUN=sum), aggregate(subset(Twps, NZone<=3)$A\_d\_nswf, by=list(Range=subset(Twps, NZone<=3)$RANGE), FUN=sum)) %>%  
 group\_by(Range) %>%  
 summarise\_all(funs(sum(., na.rm = TRUE)))  
##Zone 4:  
RestFireZ4<-bind\_rows(aggregate(subset(Twps, NZone>4)$A\_d\_wf, by=list(Range=subset(Twps, NZone>4)$RANGE), FUN=sum), aggregate(subset(Twps, NZone<=4)$A\_d\_nswf, by=list(Range=subset(Twps, NZone<=4)$RANGE), FUN=sum)) %>%  
 group\_by(Range) %>%  
 summarise\_all(funs(sum(., na.rm = TRUE)))  
##Zone 5:  
RestFireZ5<-aggregate(Twps$A\_d\_nswf, by=list(Range=Twps$RANGE), FUN=sum)  
  
##Calculate caribou range area:  
RangeArea<-aggregate(Twps$TWP\_area, by=list(Range=Twps$RANGE), FUN=sum)  
  
##Calculate current area disturbed per range for reference  
CurrentArea<-aggregate(Twps$A\_d\_wf, by=list(Range=Twps$RANGE), FUN=sum)  
  
## Combine and clean  
RestWithFire<-cbind(RestFireZ1, RestFireZ2, RestFireZ3, RestFireZ4, RestFireZ5, RangeArea, CurrentArea)  
RestWithFire<-RestWithFire[, -c(3, 5, 7, 9, 11, 13, 15)]  
RestWithFire<-RestWithFire %>% rename(Zone1=x, Zone2=x.1, Zone3=x.2,Zone4=x.3, Zone5=x.4, Total=x.5, Current=x.6)  
RestWithFire$Current<-RestWithFire$Current/RestWithFire$Total\*100  
RestWithFire$Zone1P<-RestWithFire$Zone1/RestWithFire$Total\*100  
RestWithFire$Zone2P<-RestWithFire$Zone2/RestWithFire$Total\*100  
RestWithFire$Zone3P<-RestWithFire$Zone3/RestWithFire$Total\*100  
RestWithFire$Zone4P<-RestWithFire$Zone4/RestWithFire$Total\*100  
RestWithFire$Zone5P<-RestWithFire$Zone5/RestWithFire$Total\*100  
  
View(RestWithFire[, c(1, 8:13)])  
#write.csv(RestWithFire[, c(1, 8:13)],here::here("CleanInput\_Data", "AlterationProgression.csv"))

## Calibrate Area Altered to ECCC

plotdata<-RestWithFire[, c(1, 8:13)]  
plotdata<-tidyr::gather(plotdata, Zone, Altered, Current:Zone5P)  
colkdata<-subset(plotdata,Range == "COLK")  
esardata<-subset(plotdata,Range == "ESAR")  
rededata<-subset(plotdata,Range == "REDE")  
richdata<-subset(plotdata,Range == "RICH")  
wsardata<-subset(plotdata,Range == "WSAR")  
colkdata$CalAlt <- 2.0006743 + 0.8929205\*colkdata$Altered  
esardata$CalAlt<--11.902466 + 1.021092\*esardata$Altered  
rededata$CalAlt<-3.5949225 + 0.7213575\*rededata$Altered  
richdata$CalAlt<--44.350332 + 1.383939\*richdata$Altered  
wsardata$CalAlt<--10.5390497 + 0.8954351\*wsardata$Altered  
plotdata<-rbind(colkdata, esardata, rededata, richdata, wsardata)

## Plot Area Altered as Restoration Progresses

plotdata<-tidyr::gather(plotdata, Type, Percent, Altered:CalAlt)  
plotdata$Range<-as.character(plotdata$Range)  
plotdata <- within(plotdata, Range[Range == 'COLK'] <- 'Cold Lake')  
plotdata <- within(plotdata, Range[Range == 'ESAR'] <- 'ESAR')  
plotdata <- within(plotdata, Range[Range == 'WSAR'] <- 'WSAR')  
plotdata <- within(plotdata, Range[Range == 'REDE'] <- 'Red Earth')  
plotdata <- within(plotdata, Range[Range == 'RICH'] <- 'Richardson')  
plotdata <- within(plotdata, Type[Type == 'Altered'] <- 'ABMI')  
plotdata <- within(plotdata, Type[Type == 'CalAlt'] <- 'ECCC')  
plotdata$Zone = substr(plotdata$Zone,5,nchar(plotdata$Zone)-1)  
plotdata <- within(plotdata, Zone[Zone == 'en'] <- 'C')  
plotdata$Zone <- factor(plotdata$Zone, levels = c("C", "1", "2", "3", "4", "5"))  
  
ggplot() +   
 geom\_col(data = plotdata, aes(x = Zone, y = Percent, fill=Type), colour="black", position="dodge")+  
 scale\_fill\_manual(name = "", values = c("grey40", "grey90"))+  
 theme\_bw() +   
 geom\_hline(yintercept=35, linetype="dashed",size=1.25)+  
 xlab("Zone") +  
 ylab("Percent Altered (%)") +  
 theme(axis.text.x = element\_text(size=12), axis.title = element\_text(size=12) ) +   
 theme(axis.text.y = element\_text(size=12)) +   
 theme(panel.grid.minor=element\_blank(), panel.grid.major=element\_blank()) +   
 theme(legend.text = element\_text(size = 12),legend.title = element\_text(size = 12))+  
 facet\_grid(~Range)+  
 theme(strip.text.x = element\_text(size = 12), strip.text.y = element\_text(size = 12), legend.position="top")

