Reporting on the Southwest Climate Decisions Project

Matthew A. Williamson

Wed Apr 20 10:50:50 2016

Introduction

This document details our analysis of contemporary usage of available climate information for federal land management decisions. For this analysis, we focus on those decisions where respondents indicated that climate information was relevant to the decision being considered. We then evaluate which of the existing products are used most frequently (and the attributes of the decisions in which they were used). In addition, we consider the motivations for not using climate information despite its suggested relevance.

```
pacman::p_load(tidyr, plyr, dplyr, ggplot2, reshape2, extrafont, rmarkdown,
  RColorBrewer, vegan, extrafont)
setwd("C:/ClimateDecisionAnalysis/SG_Download/")
#setwd("/Users/matthewwilliamson/Google Drive/Sync/ClimateDecisionAnalysis/SG_Download/")
orig.data <- as.data.frame(read.csv("Data_CheckAsColumn_080515.csv",
              na.strings=c("", "NA")), stringsAsFactors=FALSE)
#Remove non-participants

master <- orig.data[grep("Yes",
          orig.data$Do.you.wish.to.participate.in.the.survey.), ]
#Remove non-land managers

m <- master[grep("Yes",
      master$Do.you.work.for.a.public.natural.resource.management.agency.), ]

# developing the dataframe for analysis
nepa.1 <- m[,c(1, 71)]
nepa.2 <- m[,c(1, 72)]
nepa.1$DecisionNum <- 1
nepa.2$DecisionNum <- 2
#Agency worked for at time of decision
nepa.1 <- cbind(nepa.1, m[,73])
colnames(nepa.1) <- c("ID", "DecisionName","DecisionNum", "AgencyWorked")
nepa.2 <- cbind(nepa.2, m[,423])
colnames(nepa.2) <-c("ID", "DecisionName","DecisionNum", "AgencyWorked")
# Only a subset of the decisions were actually appropriate for using climate
#information
nepa.1 <- cbind(nepa.1, m[,207])
colnames(nepa.1)[5] <- "ClimRelevance"
nepa.2 <- cbind(nepa.2, m[,563])
colnames(nepa.2)[5] <- "ClimRelevance"

d <- rbind(nepa.1, nepa.2)
# eliminating responses that did not actually describe a decision
dt.1 <- d[!is.na(d$DecisionName), ]
dt.1 <- droplevels(dt.1)
dt.2 <- dt.1[!(dt.1$DecisionName == 'never done it' |
        dt.1$DecisionName == 'NEPA2' |
```

```
        dt.1$DecisionName == 'Do not wish to answer' |
        dt.1$DecisionName == '2' | dt.1$DecisionName == 'Zero' |
        dt.1$DecisionName == 'not involved in NEPA ever' |
        dt.1$DecisionName == 'NONE YET'|
        dt.1$DecisionName == 'None that are completed' |
        dt.1$DecisionName == 'None'| dt.1$DecisionName == 'none' |
        dt.1$DecisionName == 'NEPA 1' | dt.1$DecisionName == 'N/A' |
        dt.1$DecisionName == 'n/a' |
        dt.1$DecisionName == 'Do not wish to Answer' |
        dt.1$DecisionName == '0' | dt.1$DecisionName == '1'),]
dt.2 <- droplevels(dt.2)
```

```
gtheme <- theme(legend.position = "none", title = element_text(colour="black",
          size=12, face="bold", family = "Garamond"),
          axis.text = element_text(size=12, colour = "black"),
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title = element_text(size=12, face="bold", family = "Garamond"),
          line = element_line(size=1),
          panel.background = element_rect(fill = "white"),
          axis.line = element_line(color="black"))
```

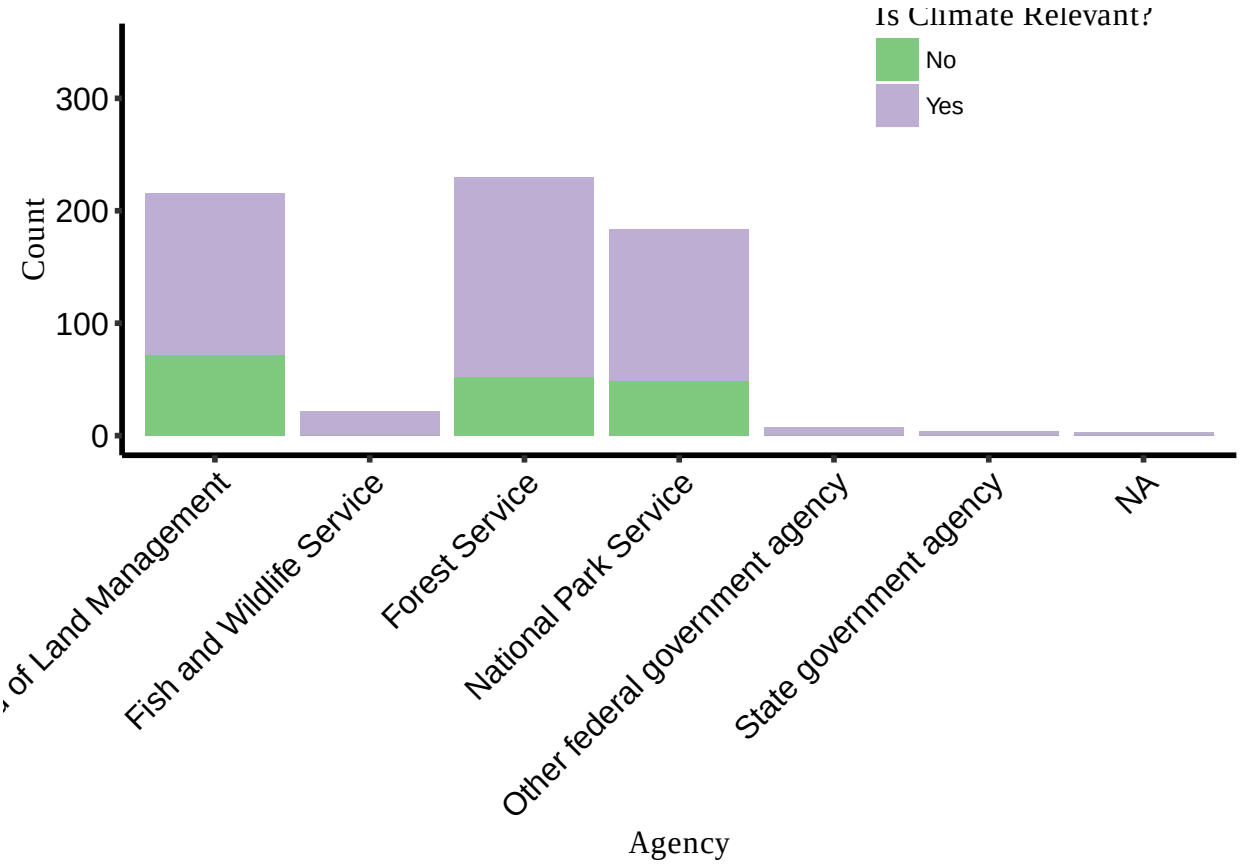
‘‘‘

Determining usage of currently available climate products

How many decisions were identified as climate-relevant

At this point, the dataset contains all of the “named” decisions (a total of 1055). This does not mean that respondents provided all of the information for all of the decisions. Justification for removing decisions from this dataset will be provided as removal becomes necessary.

```
p1 <- ggplot(dt.2, aes(x=AgencyWorked, fill=ClimRelevance,
  order=-as.numeric(ClimRelevance))) + geom_bar() +
  scale_fill_brewer(palette = "Accent",
  guide=guide_legend(title = "Is Climate Relevant?")) +
  ylab("Count") + xlab("Agency") + gtheme + theme(legend.position = c(.8,.9))
print(p1)
```



‘‘‘

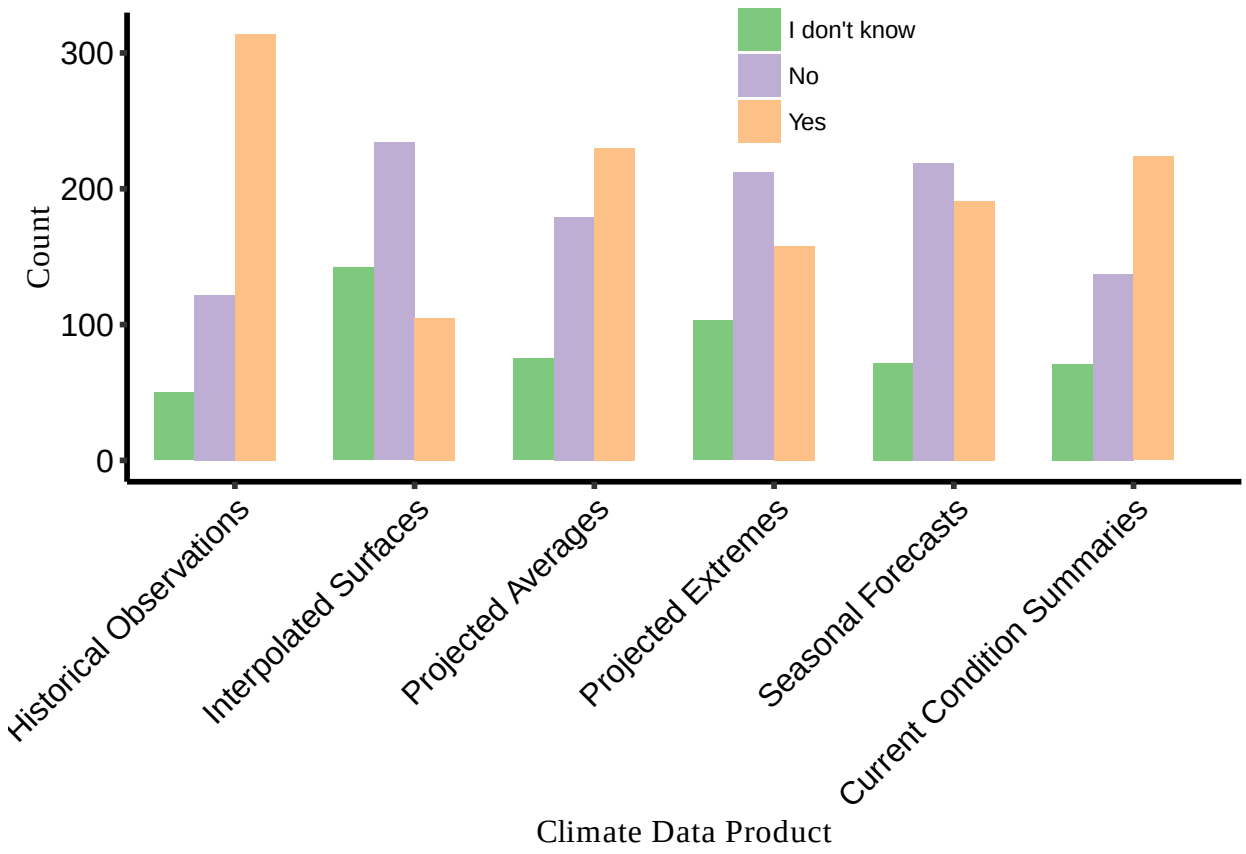
Which products were used (given that climate was relevant)?

Here we take only the decisions where respondents indicated that climate was relevant and determine which of several classes of products were used.

```
# generating the climate use dataset
dec <- subset(dt.2, ClimRelevance == "Yes")
clim.use.1 <- m[,c(1,208:213)]
#setup levels for melt
clim.use.1.lvl <- c(levels(clim.use.1[,2]), levels(clim.use.1[,3]),
                    levels(clim.use.1[,4]), levels(clim.use.1[,5]),
                    levels(clim.use.1[,6]), levels(clim.use.1[,7]))
lvl <- unique(clim.use.1.lvl)

clim.use.1[,2:7] <- lapply(clim.use.1[,2:7], factor, levels=lvl)
#melt
clim.use.1.melt <- melt(clim.use.1, id.vars = "Response.ID",
                        value.name = "Used", factorsAsStrings=F)

#Fix variables so they can be comparable to Resources Affected values for merge
clim.use.1.melt$variable <- sub("\\.For.*$", "", clim.use.1.melt$variable)
#Did not remove incomplete cases - may want to re-visit
colnames(clim.use.1.melt)[1] <- "ID"
colnames(clim.use.1.melt)[2] <- "ClimDataset"
```

```
clim.use.1.melt$DecisionNum <- 1
clim.use.2 <- m[,c(1,564:569)]
clim.use.2 <- as.data.frame(clim.use.2)
#setup levels for melt
clim.use.2.lvl <- c(levels(clim.use.2[,2]), levels(clim.use.2[,3]),
                    levels(clim.use.2[,4]), levels(clim.use.2[,5]),
                    levels(clim.use.2[,6]), levels(clim.use.2[,7]))
lvl <- unique(clim.use.2.lvl)

clim.use.2[,2:7] <- lapply(clim.use.2[,2:7], factor, levels=lvl)
#melt
clim.use.2.melt <- melt(clim.use.2, id.vars = "Response.ID",
                    value.name = "Used", factorsAsStrings=F)

#Fix variables so they can be comparable to Resources Affected values for merge
clim.use.2.melt$variable <- sub("\\.For.*$", "", clim.use.2.melt$variable)
#Did not remove incomplete cases - may want to re-visit
colnames(clim.use.2.melt)[1] <- "ID"
colnames(clim.use.2.melt)[2] <- "ClimDataset"
clim.use.2.melt$DecisionNum <- 2
clim.use <- rbind(clim.use.1.melt, clim.use.2.melt)
decision.climate <- merge(dec, clim.use, by = c("ID", "DecisionNum"))
decision.climate$ClimDataset <- as.factor(decision.climate$ClimDataset)
levels(decision.climate$ClimDataset) <- c("Historical Observations",
          "Interpolated Surfaces", "Projected Averages", "Projected Extremes",
          "Seasonal Forecasts", "Current Condition Summaries")
```

```
p2 <- ggplot(decision.climate, aes(x=ClimDataset, fill=Used,
      order=-as.numeric(Used))) + geom_bar(position = "dodge") +
      scale_fill_brewer(palette = "Accent",
      guide=guide_legend(title = "Did you use climate information?")) +
      ylab("Count") + xlab("Climate Data Product") + gtheme +
      theme(legend.position = c(.75,.9))
print(p2)
```



‘‘‘

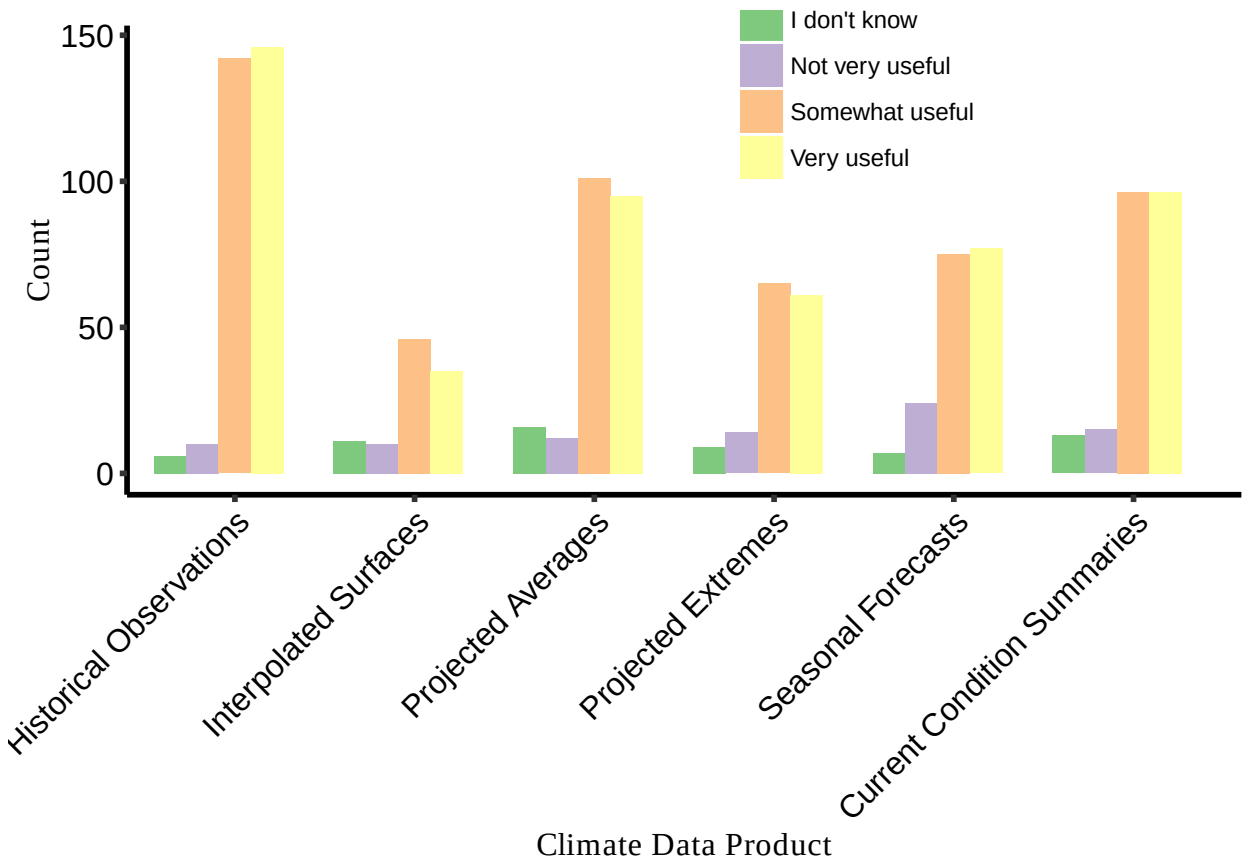
How useful were the datasets that were actually used?

Here we asked respondents to identify how useful the datasets were for the decision they were describing.

```
use <- subset(decision.climate, Used == "Yes")
clim.utility.1 <- m[,c(1,214:219)]
clim.utility.1 <- as.data.frame(clim.utility.1)
#setup levels for melt
clim.utility.1.lvl <- c(levels(clim.utility.1[,2]), levels(clim.utility.1[,3]),
                        levels(clim.utility.1[,4]), levels(clim.utility.1[,5]),
                        levels(clim.utility.1[,6]), levels(clim.utility.1[,7]))
lvl <- unique(clim.utility.1.lvl)

clim.utility.1[,2:7] <- lapply(clim.utility.1[,2:7], factor, levels=lvl)
#melt
clim.utility.1.melt <- melt(clim.utility.1, id.vars = "Response.ID",
                          value.name = "ClimUtility", factorsAsStrings=F)

#Fix variables so they can be comparable to Resources Affected values for merge
clim.utility.1.melt$variable <- sub("\\.How.*$", "",
                                    clim.utility.1.melt$variable)
#Did not remove incomplete cases - may want to re-visit
colnames(clim.utility.1.melt)[1] <- "ID"
colnames(clim.utility.1.melt)[2] <- "ClimDataset"
```

```
clim.utility.1.melt$DecisionNum <- 1

#Utility of ClimDataset
clim.utility.2 <- m[,c(1,570:575)]
clim.utility.2 <- as.data.frame(clim.utility.2)
#setup levels for melt
clim.utility.2.lvl <- c(levels(clim.utility.2[,2]), levels(clim.utility.2[,3]),
                        levels(clim.utility.2[,4]), levels(clim.utility.2[,5]),
                        levels(clim.utility.2[,6]), levels(clim.utility.2[,7]))
lvl <- unique(clim.utility.2.lvl)

clim.utility.2[,2:7] <- lapply(clim.utility.2[,2:7], factor, levels=lvl)
#melt
clim.utility.2.melt <- melt(clim.utility.2, id.vars = "Response.ID",
                        value.name = "ClimUtility", factorsAsStrings=F)

#Fix variables so they can be comparable to Resources Affected values for merge
clim.utility.2.melt$variable <- sub("\\.How.*$", "",
                                    clim.utility.2.melt$variable)
#Did not remove incomplete cases - may want to re-visit
colnames(clim.utility.2.melt)[1] <- "ID"
colnames(clim.utility.2.melt)[2] <- "ClimDataset"
clim.utility.2.melt$DecisionNum <- 2

utility <- rbind(clim.utility.1.melt, clim.utility.2.melt)
utility$ClimDataset <- as.factor(utility$ClimDataset)
levels(utility$ClimDataset) <- c("Historical Observations",
            "Interpolated Surfaces", "Projected Averages","Projected Extremes",
            "Seasonal Forecasts", "Current Condition Summaries")
utility.climate <- merge(use, utility, by = c("ID", "DecisionNum",
                          "ClimDataset"))
```

```
p3 <- ggplot(utility.climate, aes(x=ClimDataset, fill=ClimUtility)) +
      geom_bar(position = "dodge") + scale_fill_brewer(palette = "Accent",
      guide=guide_legend(title = "How useful was the information?")) +
      ylab("Count") + xlab("Climate Data Product") + gtheme +
      theme(legend.position = c(0.75, 0.9))
print(p3)
```



‘‘

What were the major impediments to using datasets that might have otherwise been useful?

Here we are interested in focusing on the reasons for not using a dataset that the respondent suggested would be useful in hopes of understanding how we might improve the utility of existing datasets. Thus, we are interested in those decisions where climate was deemed to be relevant, but a particular dataset was not used.

```
#select decisions where climate was relevant, but datasets not Used
no.use <- subset(decision.climate, Used == "No")

#Historical Weather observations
hwo.imped.1 <- m[,c(1,220:224)]
hwo.imped.1 <- as.data.frame(hwo.imped.1)
colnames(hwo.imped.1) <- c("ID", "Historical.weather.observations1",
        "Historical.weather.observations2", "Historical.weather.observations3",
        "Historical.weather.observations4", "Historical.weather.observations5")
#set up levels for melt
hwo.imped.1.lvl <- c(levels(hwo.imped.1[,2]), levels(hwo.imped.1[,3]),
                     levels(hwo.imped.1[,4]), levels(hwo.imped.1[,5]),
                     levels(hwo.imped.1[,6]))
lvl <- unique(hwo.imped.1.lvl)

hwo.imped.1[,2:6] <- lapply(hwo.imped.1[,2:6], factor, levels=lvl)
hwo.imped.1.melt <- melt(hwo.imped.1, id.vars = "ID", value.name = "Impediment",
                     factorsAsStrings=F)
```

```
hwo.imped.1.melt$variable <- "Historical Observations"
colnames(hwo.imped.1.melt)[2] <- "ClimDataset"
hwo.imped.2 <- m[,c(1,576:580)]
hwo.imped.2 <- as.data.frame(hwo.imped.2)
colnames(hwo.imped.2) <- c("ID", "Historical.weather.observations1",
        "Historical.weather.observations2", "Historical.weather.observations3",
        "Historical.weather.observations4", "Historical.weather.observations5")
#set up levels for melt
hwo.imped.2.lvl <- c(levels(hwo.imped.2[,2]), levels(hwo.imped.2[,3]),
                     levels(hwo.imped.2[,4]), levels(hwo.imped.2[,5]),
                     levels(hwo.imped.2[,6]))
lvl <- unique(hwo.imped.2.lvl)

hwo.imped.2[,2:6] <- lapply(hwo.imped.2[,2:6], factor, levels=lvl)
hwo.imped.2.melt <- melt(hwo.imped.2, id.vars = "ID", value.name = "Impediment",
                     factorsAsStrings=F)
hwo.imped.2.melt$variable <- "Historical Observations"
colnames(hwo.imped.2.melt)[2] <- "ClimDataset"

#Interpolated Surfaces
isc.imped.1 <- m[,c(1,225:229)]
isc.imped.1 <- as.data.frame(isc.imped.1)
colnames(isc.imped.1) <- c("ID",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.1",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.2",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.3",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.4",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.5")
#set up levels for melt
isc.imped.1.lvl <- c(levels(isc.imped.1[,2]), levels(isc.imped.1[,3]),
                     levels(isc.imped.1[,4]), levels(isc.imped.1[,5]),
                     levels(isc.imped.1[,6]))
lvl <- unique(isc.imped.1.lvl)

isc.imped.1[,2:6] <- lapply(isc.imped.1[,2:6], factor, levels=lvl)
isc.imped.1.melt <- melt(isc.imped.1, id.vars = "ID", value.name = "Impediment",
                     factorsAsStrings=F)
isc.imped.1.melt$variable <- "Interpolated Surfaces"
colnames(isc.imped.1.melt)[2] <- "ClimDataset"
isc.imped.2 <- m[,c(1,581:585)]
isc.imped.2 <- as.data.frame(isc.imped.2)
colnames(isc.imped.2) <- c("ID",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.1",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.2",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.3",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.4",
              "Interpolated.surfaces.of.historical.climate.data..e.g...PRISM.5")
#set up levels for melt
isc.imped.2.lvl <- c(levels(isc.imped.2[,2]), levels(isc.imped.2[,3]),
                     levels(isc.imped.2[,4]), levels(isc.imped.2[,5]),
                     levels(isc.imped.2[,6]))
lvl <- unique(isc.imped.2.lvl)
```

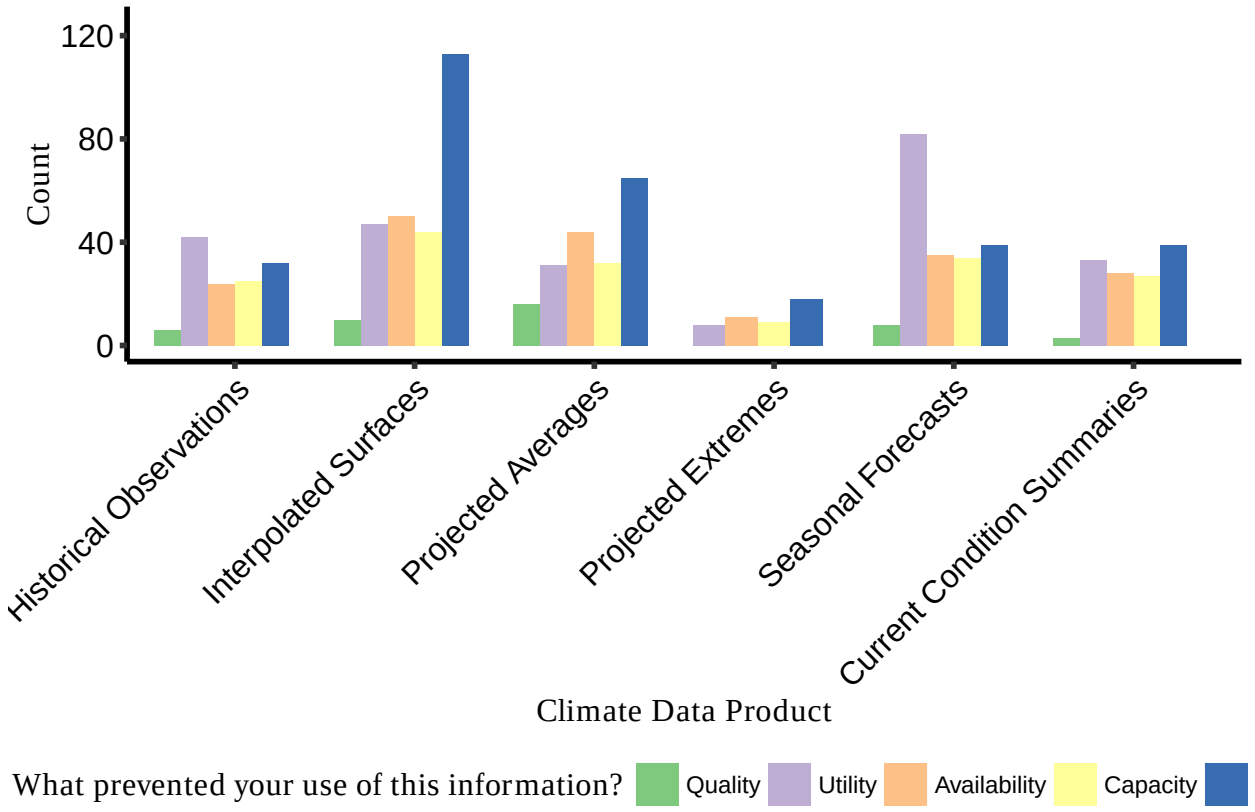
```
isc.imped.2[,2:6] <- lapply(isc.imped.2[,2:6], factor, levels=lvl)
isc.imped.2.melt <- melt(isc.imped.2, id.vars = "ID", value.name = "Impediment",
                    factorsAsStrings=F)
isc.imped.2.melt$variable <- "Interpolated Surfaces"
colnames(isc.imped.2.melt)[2] <- "ClimDataset"

#Current condition Summaries
scc.imped.1 <- m[,c(1,230:234)]
scc.imped.1 <- as.data.frame(scc.imped.1)
colnames(scc.imped.1) <- c("ID",
              "Summaries.of.current.climate.conditions..drought.indices.or.
              climate.anomaly.maps.1", "Summaries.of.current.climate.conditions
              ..drought.indices.or.climate.anomaly.maps.2", "Summaries.of.
              current.climate.conditions..drought.indices.or.climate.
              anomaly.maps.3","Summaries.of.current.climate.conditions..drought.
              indices.or.climate.anomaly.maps.4", "Summaries.of.current.climate.
              conditions..drought.indices.or.climate.anomaly.maps.5")
#set up levels for melt
scc.imped.1.lvl <- c(levels(scc.imped.1[,2]), levels(scc.imped.1[,3]),
                    levels(scc.imped.1[,4]), levels(scc.imped.1[,5]),
                    levels(scc.imped.1[,6]))
lvl <- unique(scc.imped.1.lvl)

scc.imped.1[,2:6] <- lapply(scc.imped.1[,2:6], factor, levels=lvl)
scc.imped.1.melt <- melt(scc.imped.1, id.vars = "ID", value.name = "Impediment",
                    factorsAsStrings=F)
scc.imped.1.melt$variable <- "Current Condition Summaries"
colnames(scc.imped.1.melt)[2] <- "ClimDataset"
scc.imped.2 <- m[,c(1,586:590)]
scc.imped.2 <- as.data.frame(scc.imped.2)
colnames(scc.imped.2) <- c("ID",
              "Summaries.of.current.climate.conditions..drought.indices.or.
              climate.anomaly.maps.1","Summaries.of.current.climate.conditions
              ..drought.indices.or.climate.anomaly.maps.2", "Summaries.of.
              current.climate.conditions..drought.indices.or.climate.
              anomaly.maps.3","Summaries.of.current.climate.conditions..drought.
              indices.or.climate.anomaly.maps.4", "Summaries.of.current.climate.
              conditions..drought.indices.or.climate.anomaly.maps.5")
#set up levels for melt
scc.imped.2.lvl <- c(levels(scc.imped.2[,2]), levels(scc.imped.2[,3]),
                    levels(scc.imped.2[,4]), levels(scc.imped.2[,5]),
                    levels(scc.imped.2[,6]))
lvl <- unique(scc.imped.2.lvl)

scc.imped.2[,2:6] <- lapply(scc.imped.2[,2:6], factor, levels=lvl)
scc.imped.2.melt <- melt(scc.imped.2, id.vars = "ID", value.name = "Impediment",
                    factorsAsStrings=F)
scc.imped.2.melt$variable <- "Current Condition Summaries"
colnames(scc.imped.2.melt)[2] <- "ClimDataset"

#Seasonal Outlooks
mso.imped.1 <- m[,c(1,235:239)]
mso.imped.1 <- as.data.frame(mso.imped.1)
```

```
colnames(mso.imped.1) <- c("ID", "Monthly.or.seasonal.outlooks1",
            "Monthly.or.seasonal.outlooks2", "Monthly.or.seasonal.outlooks3",
            "Monthly.or.seasonal.outlooks4", "Monthly.or.seasonal.outlooks5")
#set up levels for melt
mso.imped.1.lvl <- c(levels(mso.imped.1[,2]), levels(mso.imped.1[,3]),
                     levels(mso.imped.1[,4]), levels(mso.imped.1[,5]),
                     levels(mso.imped.1[,6]))
lvl <- unique(mso.imped.1.lvl)

mso.imped.1[,2:6] <- lapply(mso.imped.1[,2:6], factor, levels=lvl)
mso.imped.1.melt <- melt(mso.imped.1, id.vars = "ID", value.name = "Impediment",
                         factorsAsStrings=F)
mso.imped.1.melt$variable <- "Seasonal Forecasts"
colnames(mso.imped.1.melt)[2] <- "ClimDataset"
mso.imped.2 <- m[,c(1,591:595)]
mso.imped.2 <- as.data.frame(mso.imped.2)
colnames(mso.imped.2) <- c("ID", "Monthly.or.seasonal.outlooks1",
            "Monthly.or.seasonal.outlooks2", "Monthly.or.seasonal.outlooks3",
            "Monthly.or.seasonal.outlooks4", "Monthly.or.seasonal.outlooks5")
#set up levels for melt
mso.imped.2.lvl <- c(levels(mso.imped.2[,2]), levels(mso.imped.2[,3]),
                     levels(mso.imped.2[,4]), levels(mso.imped.2[,5]),
                     levels(mso.imped.2[,6]))
lvl <- unique(mso.imped.2.lvl)

mso.imped.2[,2:6] <- lapply(mso.imped.2[,2:6], factor, levels=lvl)
mso.imped.2.melt <- melt(mso.imped.2, id.vars = "ID", value.name = "Impediment",
                     factorsAsStrings=F)
mso.imped.2.melt$variable <- "Seasonal Forecasts"
colnames(mso.imped.2.melt)[2] <- "ClimDataset"

#Projected Averages
lta.imped.1 <- m[,c(1,240:244)]
lta.imped.1 <- as.data.frame(lta.imped.1)
colnames(lta.imped.1) <- c("ID",
            "Long.term....10.years.out..projections.of.climate.averages1",
            "Long.term....10.years.out..projections.of.climate.averages2",
            "Long.term....10.years.out..projections.of.climate.averages3",
            "Long.term....10.years.out..projections.of.climate.averages4",
            "Long.term....10.years.out..projections.of.climate.averages5")
#set up levels for melt
lta.imped.1.lvl <- c(levels(lta.imped.1[,2]), levels(lta.imped.1[,3]),
                     levels(lta.imped.1[,4]), levels(lta.imped.1[,5]),
                     levels(lta.imped.1[,6]))
lvl <- unique(lta.imped.1.lvl)

lta.imped.1[,2:6] <- lapply(lta.imped.1[,2:6], factor, levels=lvl)
lta.imped.1.melt <- melt(lta.imped.1, id.vars = "ID", value.name = "Impediment",
                           factorsAsStrings=F)
lta.imped.1.melt$variable <- "Projected Averages"
colnames(lta.imped.1.melt)[2] <- "ClimDataset"

lta.imped.2 <- m[,c(1,596:600)]
```

```
lta.imped.2 <- as.data.frame(lta.imped.2)
colnames(lta.imped.2) <- c("ID",
            "Long.term....10.years.out..projections.of.climate.averages1",
            "Long.term....10.years.out..projections.of.climate.averages2",
            "Long.term....10.years.out..projections.of.climate.averages3",
            "Long.term....10.years.out..projections.of.climate.averages4",
            "Long.term....10.years.out..projections.of.climate.averages5")
#set up levels for melt
lta.imped.2.lvl <- c(levels(lta.imped.2[,2]), levels(lta.imped.2[,3]),
                     levels(lta.imped.2[,4]), levels(lta.imped.2[,5]),
                     levels(lta.imped.2[,6]))
lvl <- unique(lta.imped.2.lvl)

lta.imped.2[,2:6] <- lapply(lta.imped.2[,2:6], factor, levels=lvl)
lta.imped.2.melt <- melt(lta.imped.2, id.vars = "ID", value.name = "Impediment",
                     factorsAsStrings=F)
lta.imped.2.melt$variable <- "Projected Averages"
colnames(lta.imped.2.melt)[2] <- "ClimDataset"

#long-term Extremes
lte.imped.1 <- m[,c(1,245:249)]
lte.imped.1 <- as.data.frame(lte.imped.1)
colnames(lte.imped.1) <- c("ID",
              "Long.term....10.years.out..projections.of.climate.extremes1",
              "Long.term....10.years.out..projections.of.climate.extremes2",
              "Long.term....10.years.out..projections.of.climate.extremes3",
              "Long.term....10.years.out..projections.of.climate.extremes4",
              "Long.term....10.years.out..projections.of.climate.extremes5")
#set up levels for melt
lte.imped.1.lvl <- c(levels(lte.imped.1[,2]), levels(lte.imped.1[,3]),
                     levels(lte.imped.1[,4]), levels(lte.imped.1[,5]),
                     levels(lte.imped.1[,6]))
lvl <- unique(lte.imped.1.lvl)

lte.imped.1[,2:6] <- lapply(lte.imped.1[,2:6], factor, levels=lvl)
lte.imped.1.melt <- melt(lte.imped.1, id.vars = "ID", value.name = "Impediment",
                       factorsAsStrings=F)
lte.imped.1.melt$variable <- "Projected Extremes"
colnames(lte.imped.1.melt)[2] <- "ClimDataset"

lte.imped.2 <- m[,c(1,601:605)]
lte.imped.2 <- as.data.frame(lte.imped.2)
colnames(lte.imped.2) <- c("ID",
              "Long.term....10.years.out..projections.of.climate.extremes1",
              "Long.term....10.years.out..projections.of.climate.extremes2",
              "Long.term....10.years.out..projections.of.climate.extremes3",
              "Long.term....10.years.out..projections.of.climate.extremes4",
              "Long.term....10.years.out..projections.of.climate.extremes5")
#set up levels for melt
lte.imped.2.lvl <- c(levels(lte.imped.2[,2]), levels(lte.imped.2[,3]),
                     levels(lte.imped.2[,4]), levels(lte.imped.2[,5]),
                     levels(lte.imped.2[,6]))
lvl <- unique(lte.imped.2.lvl)
```

```
lte.imped.2[,2:6] <- lapply(lte.imped.2[,2:6], factor, levels=lvl)
lte.imped.2.melt <- melt(lte.imped.2, id.vars = "ID", value.name = "Impediment",
                    factorsAsStrings=F)
lte.imped.2.melt$variable <- "Projected Extremes"
colnames(lte.imped.2.melt)[2] <- "ClimDataset"


clim.imped.1 <- rbind(hwo.imped.1.melt, isc.imped.1.melt, scc.imped.1.melt,
                mso.imped.1.melt, lta.imped.1.melt, lte.imped.1.melt)
clim.imped.2 <- rbind(hwo.imped.2.melt, isc.imped.2.melt, scc.imped.2.melt,
                mso.imped.2.melt, lta.imped.2.melt, lte.imped.2.melt)
clim.imped.1$ClimDataset <- as.factor(clim.imped.1$ClimDataset)
clim.imped.2$ClimDataset <- as.factor(clim.imped.2$ClimDataset)

clim.imped.1$DecisionNum <- 1
clim.imped.2$DecisionNum <- 2
clim.imped <- rbind(clim.imped.1, clim.imped.2)
impediment <- merge(no.use, clim.imped, by = c("ID", "DecisionNum",
              "ClimDataset"))
```

```
p4 <- ggplot(impediment, aes(x=ClimDataset, fill=Impediment)) +
      geom_bar(position = "dodge") + scale_fill_brewer(palette = "Accent",
      guide=guide_legend(title = "What prevented your use of this information?")) +
      ylab("Count") + xlab("Climate Data Product") + gtheme +
      theme(legend.position = "bottom") + ylim(c(0,125))
print(p4)
```



Developing a classification of decision types for identifying important predictors of climate data usage #‘ Land management decisions are often designed to achieve multiple objectives, implemented at multiple scales, and are governed under different procedural rules. We sought to develop a multivariate classification of these decisions as a means to reduce the dimensions of this variability into a “decision type” that could be used as a predictor in subsequent analyses.

```
dec <- dt.2
po.1 <- m[,c(1,121:143)]
po.2 <- m[,c(1,478:500)]
po.1$DecisionNum <- 1
po.2$DecisionNum <- 2
names(po.1) <- gsub("\\.Please.*$", "", names(po.1))
names(po.1) <- gsub("\\.i.e.*$", "", names(po.1))
names(po.1) <- gsub("\\.e.g.*$", "", names(po.1))
names(po.2) <- gsub("\\.Please.*$", "", names(po.2))
names(po.2) <- gsub("\\.i.e.*$", "", names(po.2))
names(po.2) <- gsub("\\.e.g.*$", "", names(po.2))
objectives <- rbind(po.1, po.2)
colnames(objectives)[1] <- "ID"

for (i in 2:24) {
  levels(objectives[,i]) <- c(levels(objectives[,i]), 0, 1)
}
objectives[, 2:24][!is.na(objectives[, 2:24])] <- 1
objectives[, 2:24][is.na(objectives[, 2:24])] <- 0
objectives <- droplevels(objectives)
dec.obj<-merge(dec, objectives, by = c("ID", "DecisionNum"))
```

```
ra.1 <- m[,c(1,145:156)]
ra.2 <- m[,c(1,502:513)]
ra.1$DecisionNum <- 1
ra.2$DecisionNum <- 2
names(ra.1) <- gsub("\\.Please.*$", "", names(ra.1))
names(ra.1) <- gsub("\\.i.e.*$", "", names(ra.1))
names(ra.1) <- gsub("\\.e.g.*$", "", names(ra.1))
names(ra.2) <- gsub("\\.Please.*$", "", names(ra.2))
names(ra.2) <- gsub("\\.i.e.*$", "", names(ra.2))
names(ra.2) <- gsub("\\.e.g.*$", "", names(ra.2))

colnames(ra.2)[12] <- "Fish.and.or.wildlife"

resources <- rbind(ra.1, ra.2)
colnames(resources)[1] <- "ID"

for (i in 2:13) {
  levels(resources[,i]) <- c(levels(resources[,i]), 0, 1)
}
resources[, 2:13][!is.na(resources[, 2:13])] <- 1
resources[, 2:13][is.na(resources[, 2:13])] <- 0
resources <- droplevels(resources)
dec.res <- merge(dec, resources, by = c("ID", "DecisionNum"))
```

“‘

the following code applies two clustering methods (Ward’s, Average Linkage), several distance measures ’(“jaccard”, “raup”,“binomial”,“horn”,“mountford”), and several cluster numbers, to cluster decisions based on 1) primary resources affected (res), 2) primary objectives of the plan (obj), and 3) a combination of resources and objectives (resobj).Ultimately, the Raup-Ward 10 cluster solution based on objectives was used as the basis for additional analyses.

```
require(reshape)
```

```
## Loading required package: reshape

##
## Attaching package: 'reshape'

## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast

## The following object is masked from 'package:dplyr':
##
##     rename

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

## The following object is masked from 'package:tidyr':
##
##     expand
```

```
res <- dec.res
obj <- dec.obj

##get list of resource types
varlist_res = colnames(res[6:17])
##get list of objective types
varlist_obj = colnames(obj[6:28])
#convert factors to integers
res[,6:17] <- apply(res[,6:17], 2, function(x) as.numeric(as.character(x)))
obj[,6:28] <- apply(obj[,6:28], 2, function(x) as.numeric(as.character(x)))
##get number of resources identified (max of 3)
res$res_sum<-rowSums(res[,6:17])
##get number of objectives identified (max of 3)
obj$obj_sum<-rowSums(obj[,6:28])

#combine resources and objectives
resobj <- merge(res, obj, by=colnames(res)[1:5])

filteredres=subset(res, res$res_sum > 0)
filteredobj=subset(obj, obj$obj_sum > 0)
filteredresobj=subset(resobj, resobj$res_sum > 0 & resobj$obj_sum > 0)


##Create similarity indices##
simind<-c("jaccard", "raup","binomial","horn","mountford")
vegdistsresobj<-as.list(vector(mode = "logical", length = length(simind)))
vegdistsres<-as.list(vector(mode = "logical", length = length(simind)))
vegdistsobj<-as.list(vector(mode = "logical", length = length(simind)))

#generate distance matrices
for (j in 1: length(simind)){
  vegdistsresobj[[j]]<-vegdist(filteredresobj[,6:42], method=simind[j],
    binary=TRUE, diag=FALSE, upper=FALSE, na.rm = TRUE)#based on presence/absence
  vegdistsres[[j]]<-vegdist(filteredres[,6:17], method=simind[j], binary=TRUE,
    diag=FALSE, upper=FALSE, na.rm = TRUE)##based on presence/absence##
  vegdistsobj[[j]]<-vegdist(filteredobj[,6:28], method=simind[j], binary=TRUE,
    diag=FALSE, upper=FALSE, na.rm = TRUE)##based on presence/absence##
}

#Cluster data using Ward's and Average linkage methods for each similarity
#index type##
clustresobj<-as.list(vector(mode = "logical", length = length(vegdistsresobj)))
clustres<-as.list(vector(mode = "logical", length = length(vegdistsres)))
clustobj<-as.list(vector(mode = "logical", length = length(vegdistsobj)))

for(j in 1:length(vegdistsresobj)){
  clustresobj[[j]]<-hclust(vegdistsresobj[[j]],method="ward")
  clustresobj[[j+5]]<-hclust(vegdistsresobj[[j]],method="average")
  clustres[[j]]<-hclust(vegdistsres[[j]],method="ward")
  clustres[[j+5]]<-hclust(vegdistsres[[j]],method="average")
  clustobj[[j]]<-hclust(vegdistsobj[[j]],method="ward")
  clustobj[[j+5]]<-hclust(vegdistsobj[[j]],method="average")
 }
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
##create breaks to assign clusters##
cutsresobj<-as.list(vector(mode = "logical", length = length(clustresobj)))
cutsres<-as.list(vector(mode = "logical", length = length(clustres)))
cutsobj<-as.list(vector(mode = "logical", length = length(clustobj)))

for ( j in 1: length(clustresobj)){
    cutsresobj[[j]]<-cutree(clustresobj[[j]], k=7, h=null)
    cutsresobj[[j+10]]<-cutree(clustresobj[[j]], k=8, h=null)
    cutsresobj[[j+20]]<-cutree(clustresobj[[j]], k=9, h=null)
    cutsresobj[[j+30]]<-cutree(clustresobj[[j]], k=10, h=null)
    cutsresobj[[j+40]]<-cutree(clustresobj[[j]], k=11, h=null)
    cutsresobj[[j+50]]<-cutree(clustresobj[[j]], k=12, h=null)
    cutsresobj[[j+60]]<-cutree(clustresobj[[j]], k=13, h=null)
    cutsresobj[[j+70]]<-cutree(clustresobj[[j]], k=14, h=null)
    cutsresobj[[j+80]]<-cutree(clustresobj[[j]], k=15, h=null)

    cutsres[[j]]<-cutree(clustres[[j]], k=7, h=null)
    cutsres[[j+10]]<-cutree(clustres[[j]], k=8, h=null)
    cutsres[[j+20]]<-cutree(clustres[[j]], k=9, h=null)
    cutsres[[j+30]]<-cutree(clustres[[j]], k=10, h=null)
    cutsres[[j+40]]<-cutree(clustres[[j]], k=11, h=null)
    cutsres[[j+50]]<-cutree(clustres[[j]], k=12, h=null)
    cutsres[[j+60]]<-cutree(clustres[[j]], k=13, h=null)
    cutsres[[j+70]]<-cutree(clustres[[j]], k=14, h=null)
    cutsres[[j+80]]<-cutree(clustres[[j]], k=15, h=null)

    cutsobj[[j]]<-cutree(clustobj[[j]], k=7, h=null)
    cutsobj[[j+10]]<-cutree(clustobj[[j]], k=8, h=null)
    cutsobj[[j+20]]<-cutree(clustobj[[j]], k=9, h=null)
    cutsobj[[j+30]]<-cutree(clustobj[[j]], k=10, h=null)
    cutsobj[[j+40]]<-cutree(clustobj[[j]], k=11, h=null)
    cutsobj[[j+50]]<-cutree(clustobj[[j]], k=12, h=null)
    cutsobj[[j+60]]<-cutree(clustobj[[j]], k=13, h=null)
    cutsobj[[j+70]]<-cutree(clustobj[[j]], k=14, h=null)
    cutsobj[[j+80]]<-cutree(clustobj[[j]], k=15, h=null)
    cutsobj[[j+90]]<-cutree(clustobj[[j]], k=16, h=null)
    cutsobj[[j+100]]<-cutree(clustobj[[j]], k=17, h=null)
```

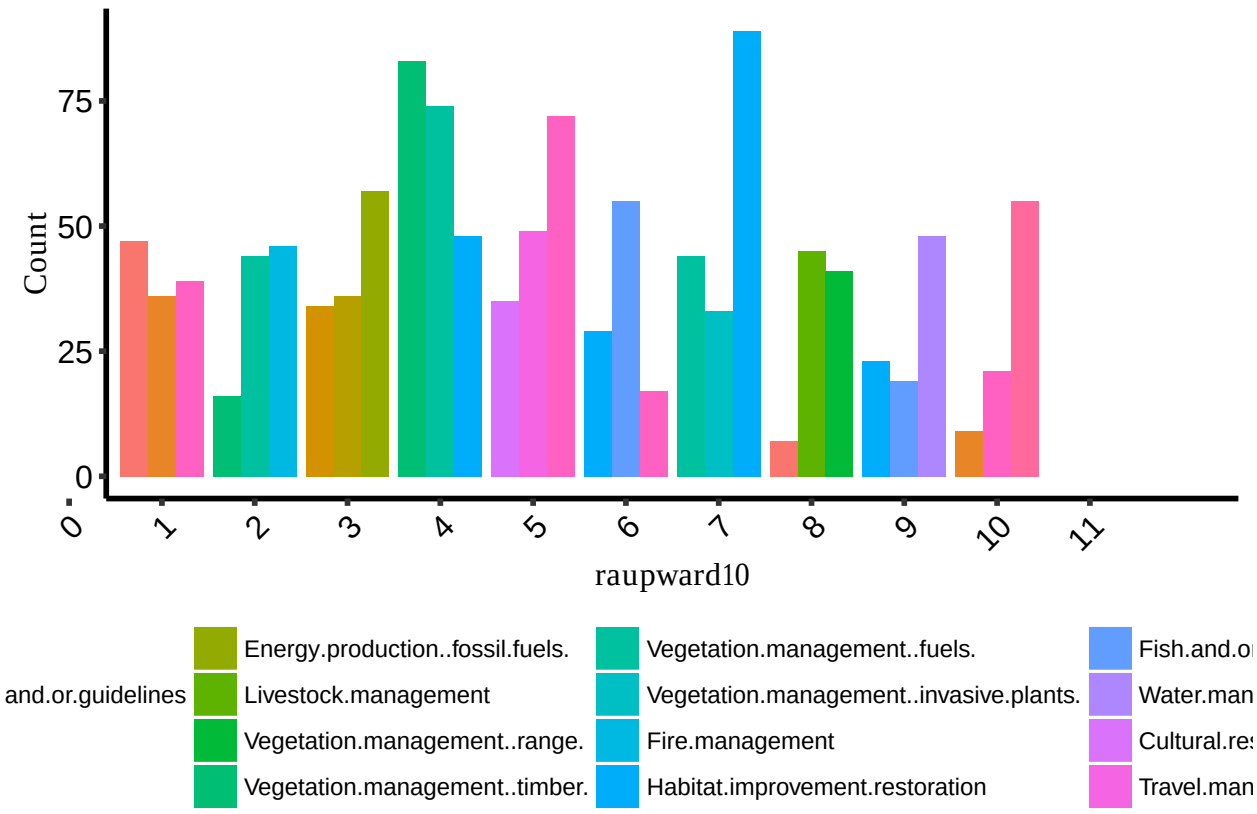
```
    cutsobj[[j+110]]<-cutree(clustobj[[j]], k=18, h=null)
    cutsobj[[j+120]]<-cutree(clustobj[[j]], k=19, h=null)
    cutsobj[[j+130]]<-cutree(clustobj[[j]], k=20, h=null)
}


##convert to dataframe and name columns
clustresultsresobj=as.data.frame(cutsresobj)
clustresultsres=as.data.frame(cutsres)
clustresultsobj=as.data.frame(cutsobj)

colnames(clustresultsresobj)<-c("jacward7", "raupward7","binward7","hornward7",
     "mountward7","jacave7", "raupave7","binave7","hornave7","mountave7",
      "jacward8", "raupward8","binward8","hornward8","mountward8","jacave8",
      "raupave8","binave8","hornave8","mountave8", "jacward9", "raupward9",
      "binward9","hornward9","mountward9","jacave9", "raupave9","binave9",
      "hornave9","mountave9","jacward10", "raupward10","binward10",
      "hornward10","mountward10","jacave10", "raupave10","binave10",
      "hornave10","mountave10", "jacward11", "raupward11","binward11",
      "hornward11","mountward11","jacave11", "raupave11","binave11",
      "hornave11","mountave11", "jacward12", "raupward12","binward12",
      "hornward12","mountward12","jacave12", "raupave12","binave12",
      "hornave12","mountave12", "jacward13", "raupward13","binward13",
      "hornward13","mountward13","jacave13", "raupave13","binave13",
      "hornave13","mountave13", "jacward14", "raupward14","binward14",
      "hornward14","mountward14","jacave14", "raupave14","binave14",
      "hornave14","mountave14", "jacward15", "raupward15","binward15",
      "hornward15","mountward15","jacave15", "raupave15","binave15",
      "hornave15","mountave15")

colnames(clustresultsres)<-c("jacward7", "raupward7","binward7","hornward7",
     "mountward7","jacave7", "raupave7","binave7","hornave7","mountave7",
     "jacward8", "raupward8","binward8","hornward8","mountward8","jacave8",
     "raupave8","binave8","hornave8","mountave8","jacward9", "raupward9",
     "binward9","hornward9","mountward9","jacave9", "raupave9","binave9",
     "hornave9","mountave9", "jacward10", "raupward10","binward10",
     "hornward10","mountward10","jacave10", "raupave10","binave10",
     "hornave10","mountave10","jacward11", "raupward11","binward11",
     "hornward11","mountward11","jacave11", "raupave11","binave11",
     "hornave11","mountave11","jacward12", "raupward12","binward12",
     "hornward12","mountward12","jacave12", "raupave12","binave12",
     "hornave12","mountave12", "jacward13", "raupward13","binward13",
     "hornward13","mountward13","jacave13", "raupave13","binave13",
     "hornave13","mountave13","jacward14", "raupward14","binward14",
     "hornward14","mountward14","jacave14", "raupave14","binave14",
     "hornave14","mountave14","jacward15", "raupward15","binward15",
     "hornward15","mountward15","jacave15", "raupave15","binave15",
     "hornave15","mountave15")

colnames(clustresultsobj)<-c("jacward7", "raupward7","binward7","hornward7",
     "mountward7","jacave7", "raupave7","binave7","hornave7","mountave7",
     "jacward8", "raupward8","binward8","hornward8","mountward8","jacave8",
     "raupave8","binave8","hornave8","mountave8","jacward9", "raupward9",
```

```
      "binward9","hornward9","mountward9","jacave9", "raupave9","binave9",
      "hornave9","mountave9","jacward10", "raupward10","binward10",
      "hornward10","mountward10","jacave10", "raupave10","binave10","hornave10",
      "mountave10","jacward11", "raupward11","binward11","hornward11",
      "mountward11","jacave11", "raupave11","binave11","hornave11","mountave11",
      "jacward12", "raupward12","binward12","hornward12","mountward12",
      "jacave12", "raupave12","binave12","hornave12","mountave12","jacward13",
      "raupward13","binward13","hornward13","mountward13","jacave13",
      "raupave13","binave13","hornave13","mountave13","jacward14", "raupward14",
      "binward14","hornward14","mountward14","jacave14", "raupave14","binave14",
      "hornave14","mountave14","jacward15", "raupward15","binward15",
      "hornward15","mountward15","jacave15", "raupave15","binave15","hornave15",
      "mountave15", "jacward16", "raupward16","binward16","hornward16",
      "mountward16","jacave16", "raupave16","binave16","hornave16","mountave16",
      "jacward17", "raupward17","binward17","hornward17","mountward17",
      "jacave17", "raupave17","binave17","hornave17","mountave17","jacward18",
      "raupward18","binward18","hornward18","mountward18","jacave18",
      "raupave18","binave18","hornave18","mountave18","jacward19", "raupward19",
      "binward19","hornward19","mountward19","jacave19", "raupave19","binave19",
      "hornave19","mountave19","jacward20", "raupward20","binward20",
      "hornward20","mountward20","jacave20", "raupave20","binave20","hornave20",
      "mountave20")
##bind cluster results to original file
filteredclustresobj<-cbind(filteredresobj, clustresultsresobj)
filteredclustres<-cbind(filteredres, clustresultsres)
filteredclustobj<-cbind(filteredobj, clustresultsobj)

dec.type <- filteredclustobj[,c(1:28,61)]
d <- melt(dec.type[,6:29], id.vars="raupward10")
ds <- ddply(d, .(raupward10, variable), summarise,
        freq = sum(value))
d.summary <- ds %>%
      group_by(raupward10) %>%
      top_n(n=3, wt=freq)
d.summary <- as.data.frame(d.summary)

dec.typ <- dec.type[,c(1,2,29)]

detach("package:reshape", unload=TRUE)
```

‘‘‘

In order to aid in the interpretation of the various decision types, we created plots of the top three objectives within each decision type.

```
p5 <- ggplot(d.summary, aes(x=raupward10, y = freq, fill=variable)) +
              geom_bar(stat="identity", position = "dodge") +
              scale_colour_brewer(type = "qual", palette = 2,
              guide=guide_legend(title = "Objectives")) +
              scale_x_discrete("raupward10") + ylab("Count") + gtheme +
              theme(legend.position = "bottom")
print(p5)
```




```
#Load the clean and complete decision set
dec <- merge(dt.2, dec.typ, by=c("ID","DecisionNum"))
#Attach Agency Role
role.1 <- m[,c(1,75)]
role.2 <- m[,c(1,425)]
colnames(role.1)<- c("ID", "AgencyRole")
colnames(role.2) <- c("ID", "AgencyRole")
role.1$DecisionNum <- 1
role.2$DecisionNum <- 2
#remove the hyperlink markup
role.1$AgencyRole <- sub("<.*?>", "", role.1$AgencyRole)
role.1$AgencyRole <- sub("</span>", "", role.1$AgencyRole)
role.2$AgencyRole <- sub("<.*?>", "", role.2$AgencyRole)
role.2$AgencyRole <- sub("</span>", "", role.2$AgencyRole)
role <- rbind(role.1, role.2)
decR <- merge(dec, role, by = c("ID", "DecisionNum"))
decR$AgencyRole <- as.factor(decR$AgencyRole)

#Attach Spatial scales
scale.1 <- m[,c(1,111)]
scale.2 <- m[,c(1,461)]
colnames(scale.1) <- c("ID","SpatialScale")
colnames(scale.2) <- c("ID","SpatialScale")
scale.1$DecisionNum <- 1
scale.2$DecisionNum <- 2
#remove markup
```

```
scale.1$SpatialScale <- sub("&gt;", "Greater than ", scale.1$SpatialScale)
scale.1$SpatialScale <- sub("&lt;", "Less than ", scale.1$SpatialScale)
scale.2$SpatialScale <- sub("&gt;", "Greater than ", scale.2$SpatialScale)
scale.2$SpatialScale <- sub("&lt;", "Less than ", scale.2$SpatialScale)
scale <- rbind(scale.1,scale.2)
#convert I don't know to NA and create an ordered variable
scale$SpatialScale <- as.factor(scale$SpatialScale)
scale$SpatialScale <- factor(scale$SpatialScale,
                      exclude = levels(scale$SpatialScale)[5],
                      levels=levels(scale$SpatialScale)[c(6,1,2,3,4)])
```

```
## Warning in as.vector(exclude, typeof(x)): NAs introduced by coercion
```

```
scale$SpatialScale <- ordered(scale$SpatialScale)
decRS <- merge(decR, scale, by = c("ID", "DecisionNum"))

#Plan type
pt.1 <- m[,c(1,119,120)]
pt.2 <- m[,c(1,476, 477)]
colnames(pt.1) <- c("ID","PlanType","Other")
colnames(pt.2) <- c("ID","PlanType","Other")
pt.1$DecisionNum <- 1
pt.2$DecisionNum <- 2
pt <- rbind(pt.1, pt.2)
pt$PlanType <- as.factor(with( pt,
                ifelse( PlanType == "Other (please enter in Comments below)",
                NA, as.character(PlanType))))
pt$Other <- NULL

decRSP <- merge(decRS, pt, by=c("ID","DecisionNum"))


#Attach NEPA Type (Programmatic EIS, EIS, EA, etc.)
nt.1 <- m[,c(1,118)]
nt.2 <- m[,c(1,475)]
colnames(nt.1) <- c("ID","NEPAType")
colnames(nt.2) <- c("ID","NEPAType")
nt.1$DecisionNum <- 1
nt.2$DecisionNum <- 2
nt <- rbind(nt.1, nt.2)
decRSPN <- merge(decRSP, nt, by=c("ID", "DecisionNum"))

#Were emissions analyzed
em.1 <- m[,c(1,175,176)]
em.2 <- m[,c(1,532,533)]
colnames(em.1) <- c("ID","Emissions", "CCImplications")
colnames(em.2) <- c("ID","Emissions", "CCImplications")
em.1$DecisionNum <- 1
em.2$DecisionNum <- 2
em <- rbind(em.1, em.2)
decRSPNE <- merge(decRSPN, em, by=c("ID","DecisionNum"))

#Constraints
```

```
const.1 <- m[,c(1,202:206)]
const.2 <- m[,c(1, 558:562)]
colnames(const.1) <- c("ID", "Inst.Const", "SP.Const", "Econ.Const",
                       "Inform.Const", "Fin.Const")
colnames(const.2) <- c("ID", "Inst.Const", "SP.Const", "Econ.Const",
                         "Inform.Const", "Fin.Const")
const.1$DecisionNum <- 1
const.2$DecisionNum <- 2
const <- rbind(const.1, const.2)

const$Inst.Const <- gsub(" +"," ",gsub("^ +","",gsub("[^a-zA-Z0-9 ]","",
                      const$Inst.Const)))
const$SP.Const <- gsub(" +"," ",gsub("^ +","",gsub("[^a-zA-Z0-9 ]","",
                    const$SP.Const)))
const$Econ.Const <- gsub(" +"," ",gsub("^ +","",gsub("[^a-zA-Z0-9 ]","",
                      const$Econ.Const)))
const$Inform.Const <- gsub(" +"," ",gsub("^ +","",gsub("[^a-zA-Z0-9 ]","",
                        const$Inform.Const)))
const$Fin.Const <- gsub(" +"," ",gsub("^ +","",gsub("[^a-zA-Z0-9 ]","",
                    const$Fin.Const)))
const[,2:6] <- lapply(const[,2:6], as.factor)
const[,2:6] <- lapply(const[,2:6], ordered)
decRSPNEC <- merge(decRSPNE, const, by =c("ID", "DecisionNum"))

#Add resources affected
res.aff <- res[,c(1,2,6:17)]
decRSPNECRA <- merge(decRSPNEC, res.aff, by = c("ID", "DecisionNum"))

#Timeframe analyzed
time.1 <- m[,c(1,112:117)]
time.2 <- m[,c(1,469:474)]
time.1$DecisionNum <- 1
time.2$DecisionNum <- 2
colnames(time.1) <- c("ID", "Short", "Moderate", "Long", "Very Long",
                      "I dont know", "None of the above", "DecisionNum")
colnames(time.2) <- c("ID", "Short", "Moderate", "Long", "Very Long",
                        "I dont know", "None of the above", "DecisionNum")
time <- rbind(time.1,time.2)
#setup levels for melt
time.lvl <- c(levels(time[,2]), levels(time[,3]), levels(time[,4]),
              levels(time[,5]), levels(time[,6]), levels(time[,7]))
lvl <- unique(time.lvl)

#apply levels to all columns
time[,2:7] <- lapply(time[,2:7], factor, levels=lvl)
#melt
time.melt <- melt(time, id.vars = c("ID","DecisionNum"),
              value.name = "TempScale", factorsAsStrings=F)

#take melted dataframe and prepare for merge
time.sum <- time.melt[complete.cases(time.melt),]
time.sum$variable <- NULL
decRSPNECRAT <- merge(decRSPNECRA, time.sum, by=c("ID","DecisionNum"))
```

```
decRSPNECRAT$TempScale <- factor(decRSPNECRAT$TempScale,
                        exclude = levels(decRSPNECRAT$TempScale)[5:6],
                        levels=levels(decRSPNECRAT$TempScale)[c(1,2,3,4)])
```

```
## Warning in as.vector(exclude, typeof(x)): NAs introduced by coercion
```

```
decRSPNECRAT$TempScale <- ordered(decRSPNECRAT$TempScale)
decRSPNECRAT$ClimRelevance<-ifelse(decRSPNECRAT$ClimRelevance == "Yes", 1, 0)
decRSPNECRAT$raupward10 <- as.factor(decRSPNECRAT$raupward10)

preds <- decRSPNECRAT
levels(preds$CCImplications) <- c("I don't know", "No, I was involved in the analysis but did not analy
                                  "No, I was involved in the analysis but did not analyze these effects."
                                  "Yes, I was involved in the analysis and analyzed these effects.")
preds[,15:26] <- lapply(preds[,15:26], as.factor)
```

‘‘‘

Fitting Boosted Regression Trees

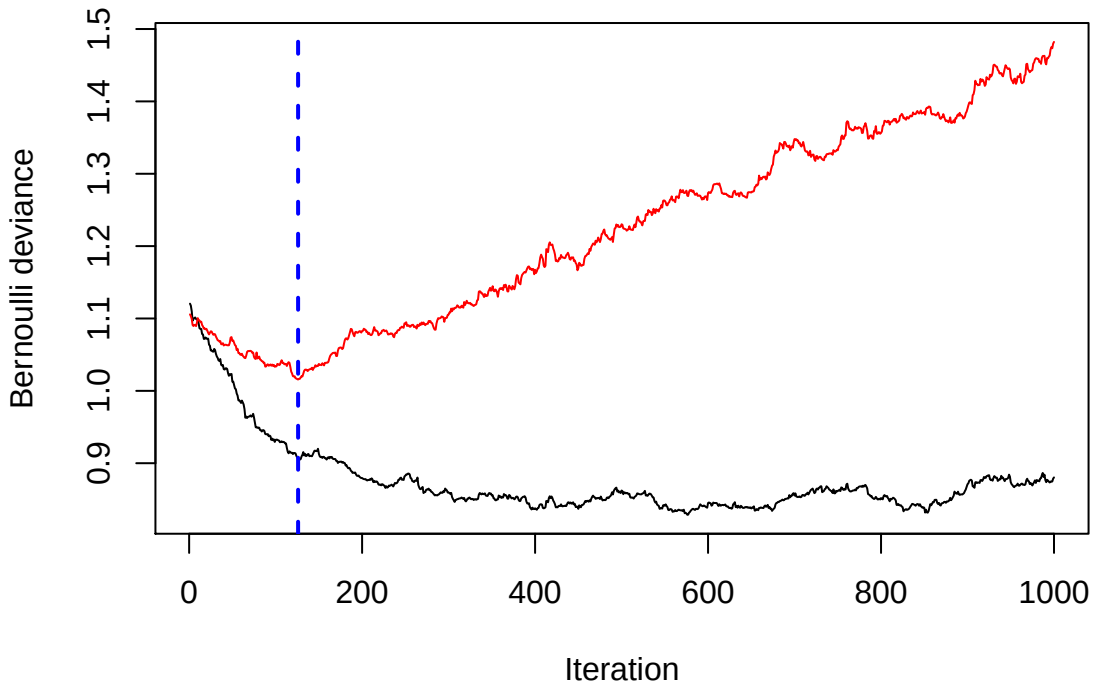
Attributes of a decision that lead to climate being relevant

```
library(gbm)
set.seed(082980)
preds <- preds[,c(5,4,6:30)]
preds <- preds[!is.na(preds$ClimRelevance),]

clm.gbm.1 <- gbm(ClimRelevance ~ . ,
                        data = preds, distribution = "bernoulli",
                        n.trees=1000, shrinkage = 0.05,
                        interaction.depth = 5, bag.fraction = 0.05,
                        train.fraction = 0.5, mFeatures = 3, n.minobsinnode = 5,
                        cv.folds = 4, keep.data= TRUE, verbose = TRUE,
                        n.cores =1)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.1205          1.1057     0.0500    0.0066
##      2        1.1177          1.1032     0.0500    0.0014
##      3        1.1100          1.0994     0.0500    0.0028
##      4        1.0970          1.0911     0.0500    0.0052
##      5        1.0992          1.0896     0.0500   -0.0017
##      6        1.1005          1.0910     0.0500   -0.0008
##      7        1.1014          1.0912     0.0500   -0.0006
##      8        1.0993          1.0895     0.0500    0.0009
##      9        1.0985          1.0939     0.0500   -0.0003
##     10        1.0982          1.1005     0.0500   -0.0011
##     20        1.0720          1.0857     0.0500    0.0003
##     40        1.0301          1.0627     0.0500    0.0020
##     60        0.9832          1.0497     0.0500    0.0006
##     80        0.9493          1.0458     0.0500   -0.0004
```

```
##    100        0.9323          1.0337     0.0500   -0.0023
##    120        0.9127          1.0203     0.0500    0.0004
##    140        0.9103          1.0300     0.0500   -0.0004
##    160        0.9072          1.0389     0.0500   -0.0019
##    180        0.9003          1.0700     0.0500   -0.0004
##    200        0.8795          1.0826     0.0500   -0.0002
##    220        0.8702          1.0780     0.0500   -0.0000
##    240        0.8724          1.0800     0.0500   -0.0007
##    260        0.8739          1.0881     0.0500   -0.0016
##    280        0.8604          1.0938     0.0500    0.0005
##    300        0.8595          1.1053     0.0500   -0.0005
##    320        0.8473          1.1214     0.0500    0.0003
##    340        0.8572          1.1323     0.0500    0.0001
##    360        0.8543          1.1420     0.0500   -0.0010
##    380        0.8530          1.1590     0.0500    0.0003
##    400        0.8378          1.1673     0.0500   -0.0011
##    420        0.8520          1.2023     0.0500    0.0004
##    440        0.8443          1.1821     0.0500    0.0004
##    460        0.8481          1.1882     0.0500   -0.0009
##    480        0.8494          1.2230     0.0500   -0.0005
##    500        0.8622          1.2302     0.0500   -0.0024
##    520        0.8574          1.2357     0.0500   -0.0002
##    540        0.8470          1.2511     0.0500   -0.0015
##    560        0.8334          1.2636     0.0500   -0.0005
##    580        0.8344          1.2766     0.0500   -0.0015
##    600        0.8450          1.2735     0.0500   -0.0044
##    620        0.8429          1.2720     0.0500   -0.0011
##    640        0.8384          1.2699     0.0500    0.0006
##    660        0.8381          1.2918     0.0500   -0.0006
##    680        0.8453          1.3299     0.0500   -0.0001
##    700        0.8553          1.3480     0.0500    0.0011
##    720        0.8641          1.3251     0.0500   -0.0006
##    740        0.8637          1.3280     0.0500    0.0005
##    760        0.8706          1.3705     0.0500   -0.0054
##    780        0.8675          1.3646     0.0500   -0.0000
##    800        0.8518          1.3565     0.0500   -0.0017
##    820        0.8436          1.3724     0.0500   -0.0012
##    840        0.8412          1.3824     0.0500    0.0004
##    860        0.8465          1.3825     0.0500   -0.0018
##    880        0.8522          1.3779     0.0500   -0.0023
##    900        0.8709          1.3900     0.0500   -0.0021
##    920        0.8802          1.4214     0.0500   -0.0003
##    940        0.8793          1.4359     0.0500    0.0005
##    960        0.8689          1.4339     0.0500   -0.0001
##    980        0.8768          1.4596     0.0500   -0.0027
##   1000        0.8804          1.4822     0.0500   -0.0010

## CV:1

## CV:2

## CV:3

## CV:4
```

```
best.iter <- gbm.perf(clm.gbm.1,method="test",plot.it=TRUE)
```

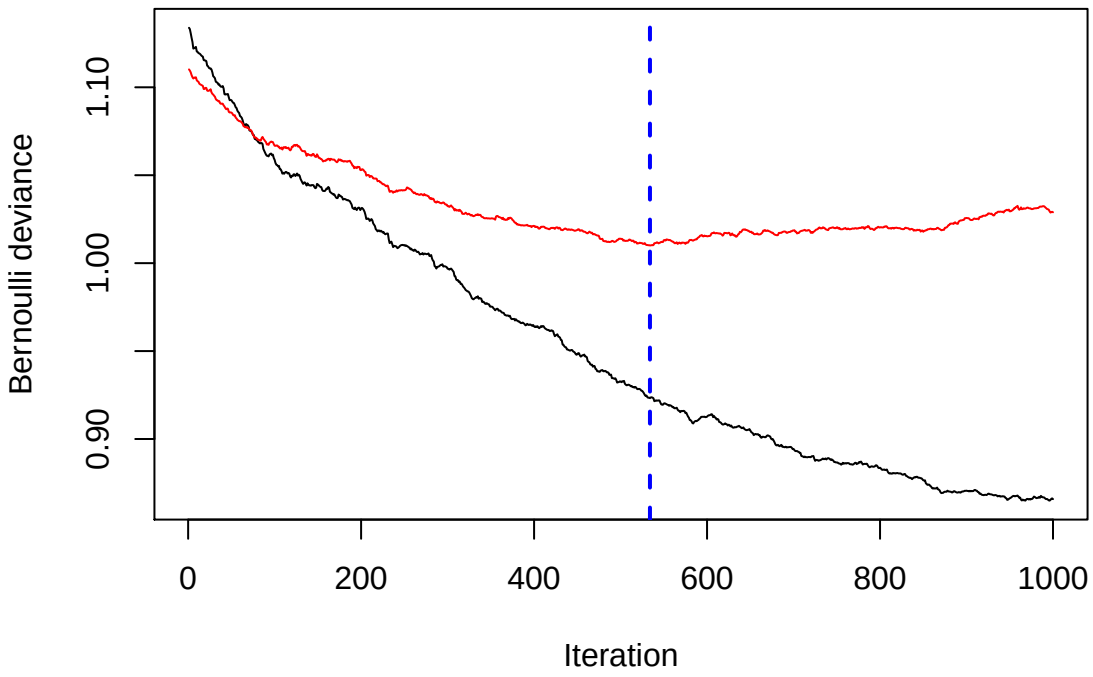



```
clm.gbm.2 <- gbm(ClimRelevance ~ . ,
                         data = preds, distribution = "bernoulli",
                         n.trees=1000, shrinkage = 0.01,
                         interaction.depth = 5, bag.fraction = 0.05,
                         train.fraction = 0.5, mFeatures = 3, n.minobsinnode = 5,
                         cv.folds = 4, keep.data= TRUE, verbose = TRUE,
                         n.cores =1)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.1338          1.1102     0.0100    0.0003
##      2        1.1331          1.1095     0.0100    0.0003
##      3        1.1307          1.1085     0.0100    0.0011
##      4        1.1281          1.1069     0.0100    0.0010
##      5        1.1258          1.1057     0.0100    0.0010
##      6        1.1220          1.1050     0.0100    0.0016
##      7        1.1222          1.1056     0.0100   -0.0002
##      8        1.1226          1.1055     0.0100   -0.0004
##      9        1.1230          1.1057     0.0100   -0.0003
##     10        1.1203          1.1039     0.0100    0.0013
##     20        1.1152          1.0997     0.0100   -0.0001
##     40        1.1007          1.0907     0.0100   -0.0004
##     60        1.0837          1.0807     0.0100    0.0007
##     80        1.0697          1.0702     0.0100    0.0000
```

```
##    100        1.0587          1.0668     0.0100    0.0005
##    120        1.0497          1.0662     0.0100   -0.0007
##    140        1.0442          1.0614     0.0100    0.0002
##    160        1.0424          1.0589     0.0100   -0.0005
##    180        1.0364          1.0577     0.0100   -0.0002
##    200        1.0310          1.0529     0.0100   -0.0006
##    220        1.0186          1.0467     0.0100   -0.0004
##    240        1.0097          1.0406     0.0100   -0.0001
##    260        1.0073          1.0404     0.0100    0.0001
##    280        1.0042          1.0366     0.0100   -0.0002
##    300        0.9968          1.0324     0.0100    0.0002
##    320        0.9855          1.0283     0.0100    0.0001
##    340        0.9779          1.0263     0.0100    0.0008
##    360        0.9730          1.0262     0.0100   -0.0001
##    380        0.9674          1.0220     0.0100    0.0003
##    400        0.9640          1.0212     0.0100    0.0003
##    420        0.9618          1.0201     0.0100   -0.0002
##    440        0.9506          1.0193     0.0100    0.0001
##    460        0.9469          1.0174     0.0100   -0.0000
##    480        0.9385          1.0126     0.0100    0.0002
##    500        0.9327          1.0139     0.0100    0.0000
##    520        0.9283          1.0112     0.0100    0.0000
##    540        0.9217          1.0114     0.0100   -0.0001
##    560        0.9181          1.0122     0.0100    0.0003
##    580        0.9108          1.0132     0.0100    0.0005
##    600        0.9127          1.0153     0.0100   -0.0001
##    620        0.9085          1.0172     0.0100   -0.0004
##    640        0.9061          1.0181     0.0100    0.0002
##    660        0.9021          1.0163     0.0100    0.0003
##    680        0.8964          1.0160     0.0100    0.0002
##    700        0.8944          1.0186     0.0100   -0.0001
##    720        0.8897          1.0172     0.0100   -0.0002
##    740        0.8891          1.0188     0.0100   -0.0002
##    760        0.8863          1.0200     0.0100   -0.0002
##    780        0.8855          1.0201     0.0100    0.0003
##    800        0.8837          1.0203     0.0100   -0.0002
##    820        0.8807          1.0200     0.0100   -0.0001
##    840        0.8783          1.0189     0.0100   -0.0002
##    860        0.8722          1.0192     0.0100    0.0003
##    880        0.8701          1.0216     0.0100    0.0000
##    900        0.8706          1.0257     0.0100   -0.0001
##    920        0.8680          1.0270     0.0100   -0.0000
##    940        0.8674          1.0292     0.0100   -0.0001
##    960        0.8673          1.0310     0.0100    0.0000
##    980        0.8666          1.0312     0.0100    0.0002
##   1000        0.8658          1.0289     0.0100    0.0002

## CV:1

## CV:2

## CV:3

## CV:4
```

```
best.iter <- gbm.perf(clm.gbm.2,method="test",plot.it=TRUE)
```



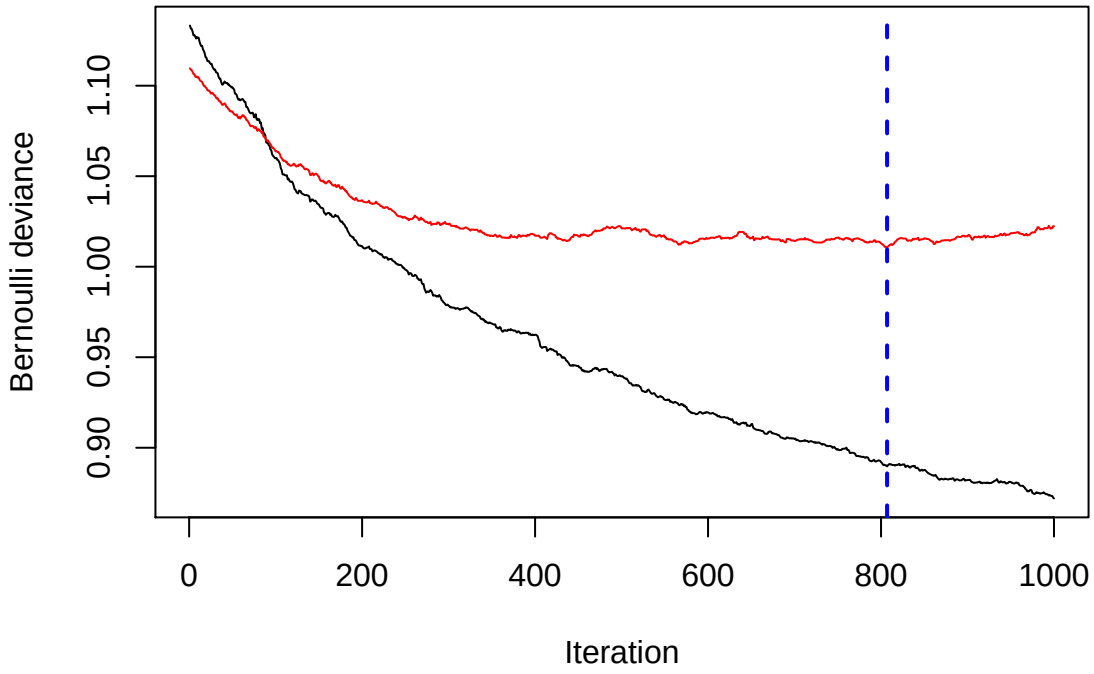

```
#This appears to be the best model
clm.gbm.3 <- gbm(ClimRelevance ~ . ,
                         data = preds, distribution = "bernoulli",
                         n.trees=1000, shrinkage = 0.01,
                         interaction.depth = 3, bag.fraction = 0.05,
                         train.fraction = 0.5, mFeatures = 3, n.minobsinnode = 5,
                         cv.folds = 4, keep.data= TRUE, verbose = TRUE,
                         n.cores =1)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.1332          1.1096     0.0100    0.0006
##      2        1.1315          1.1088     0.0100    0.0009
##      3        1.1314          1.1086     0.0100    0.0000
##      4        1.1299          1.1076     0.0100    0.0005
##      5        1.1278          1.1065     0.0100    0.0009
##      6        1.1280          1.1063     0.0100   -0.0002
##      7        1.1273          1.1051     0.0100    0.0003
##      8        1.1260          1.1048     0.0100    0.0005
##      9        1.1269          1.1051     0.0100   -0.0005
##     10        1.1265          1.1044     0.0100    0.0001
##     20        1.1155          1.0988     0.0100    0.0001
##     40        1.1014          1.0898     0.0100   -0.0006
##     60        1.0922          1.0829     0.0100   -0.0003
```

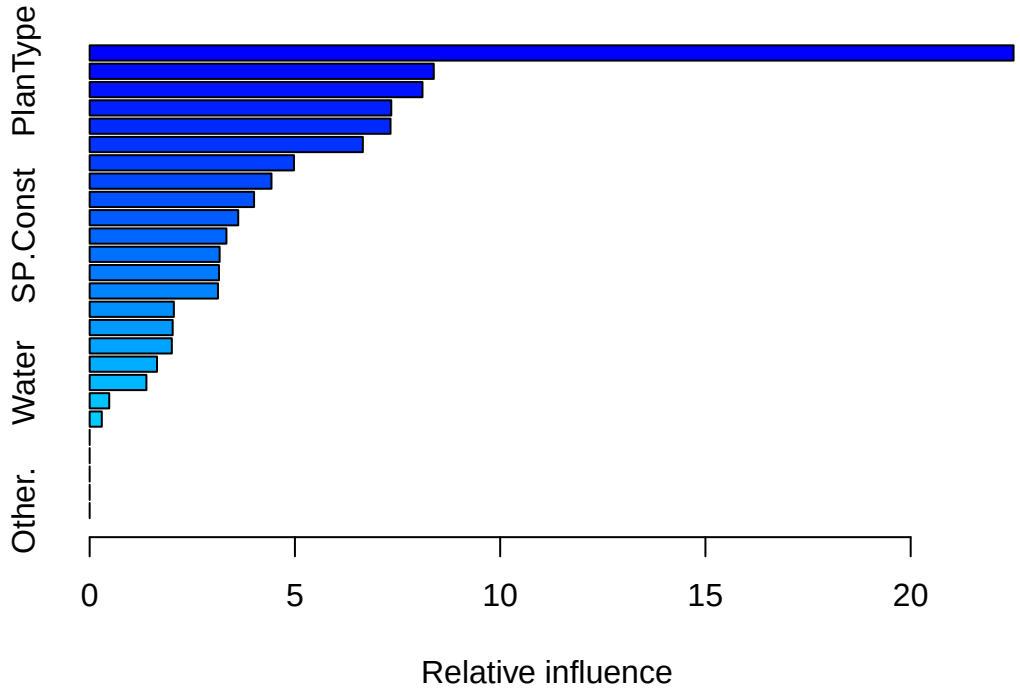
```
##     80        1.0816          1.0760     0.0100   -0.0004
##    100        1.0599          1.0637     0.0100   -0.0001
##    120        1.0461          1.0563     0.0100    0.0002
##    140        1.0359          1.0506     0.0100    0.0011
##    160        1.0296          1.0467     0.0100   -0.0003
##    180        1.0229          1.0427     0.0100    0.0003
##    200        1.0114          1.0367     0.0100   -0.0001
##    220        1.0073          1.0344     0.0100   -0.0000
##    240        1.0017          1.0286     0.0100   -0.0000
##    260        0.9949          1.0275     0.0100   -0.0002
##    280        0.9865          1.0231     0.0100   -0.0000
##    300        0.9785          1.0234     0.0100    0.0001
##    320        0.9776          1.0217     0.0100   -0.0004
##    340        0.9706          1.0185     0.0100    0.0002
##    360        0.9665          1.0169     0.0100    0.0000
##    380        0.9642          1.0165     0.0100   -0.0004
##    400        0.9624          1.0171     0.0100   -0.0002
##    420        0.9542          1.0179     0.0100   -0.0001
##    440        0.9457          1.0148     0.0100    0.0003
##    460        0.9422          1.0173     0.0100   -0.0001
##    480        0.9435          1.0211     0.0100   -0.0000
##    500        0.9394          1.0212     0.0100   -0.0001
##    520        0.9344          1.0206     0.0100    0.0000
##    540        0.9286          1.0165     0.0100    0.0002
##    560        0.9247          1.0140     0.0100    0.0001
##    580        0.9194          1.0130     0.0100    0.0002
##    600        0.9195          1.0155     0.0100   -0.0001
##    620        0.9167          1.0160     0.0100    0.0001
##    640        0.9138          1.0192     0.0100   -0.0002
##    660        0.9097          1.0151     0.0100   -0.0001
##    680        0.9072          1.0150     0.0100    0.0001
##    700        0.9048          1.0142     0.0100    0.0001
##    720        0.9034          1.0144     0.0100    0.0001
##    740        0.9009          1.0150     0.0100   -0.0000
##    760        0.9001          1.0160     0.0100   -0.0004
##    780        0.8950          1.0143     0.0100   -0.0002
##    800        0.8921          1.0129     0.0100    0.0002
##    820        0.8907          1.0152     0.0100   -0.0002
##    840        0.8889          1.0154     0.0100    0.0004
##    860        0.8848          1.0139     0.0100   -0.0000
##    880        0.8828          1.0143     0.0100   -0.0001
##    900        0.8821          1.0167     0.0100   -0.0001
##    920        0.8806          1.0172     0.0100   -0.0001
##    940        0.8810          1.0173     0.0100   -0.0000
##    960        0.8793          1.0183     0.0100    0.0004
##    980        0.8754          1.0204     0.0100   -0.0003
##   1000        0.8720          1.0223     0.0100    0.0001

## CV:1

## CV:2

## CV:3
```

```
## CV:4
```

```
best.iter <- gbm.perf(clm.gbm.3,method="test",plot.it=TRUE)
```



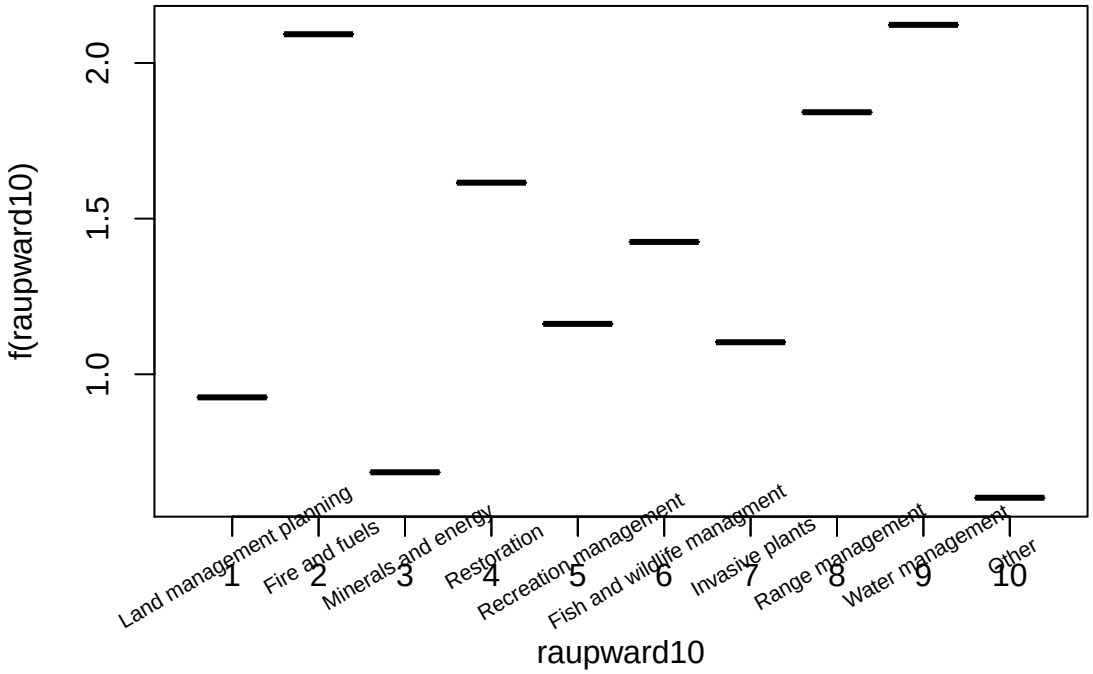

```
summary(clm.gbm.2)
```



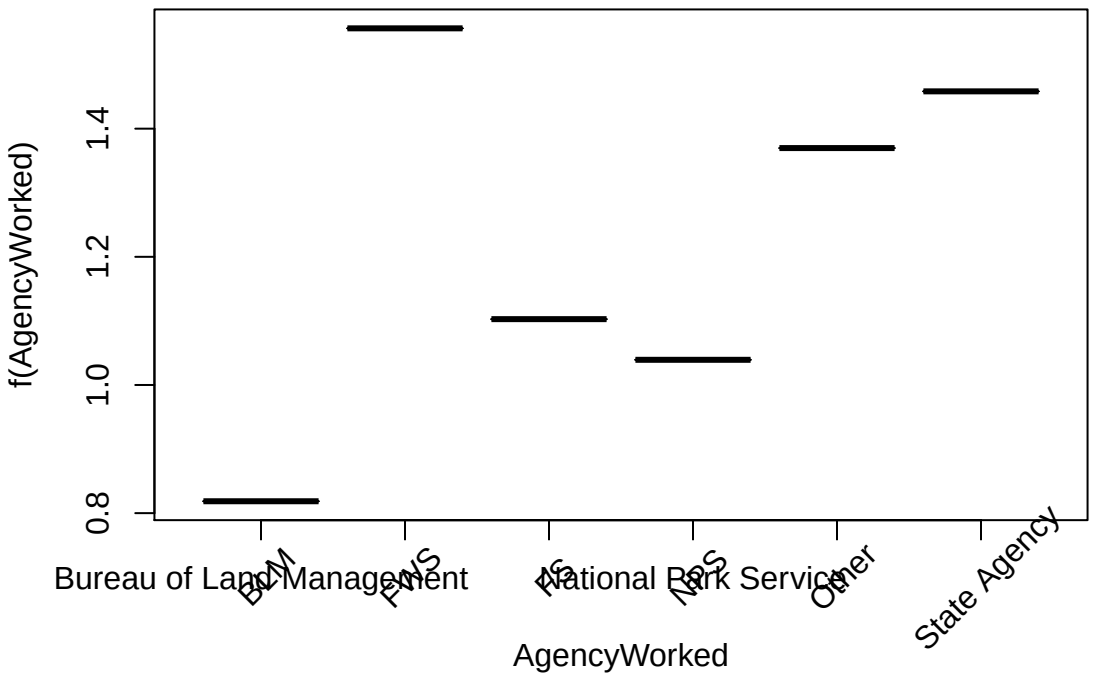

```
##                                    var    rel.inf
## raupward10                   raupward10 22.5061709
## PlanType                       PlanType  8.3824645
## Emissions                     Emissions  8.1075160
## AgencyWorked               AgencyWorked  7.3470894
## SpatialScale               SpatialScale  7.3287418
## CCImplications           CCImplications  6.6549521
## NEPAType                       NEPAType  4.9766440
## Econ.Const                   Econ.Const  4.4293603
## TempScale                     TempScale  4.0046433
## Fish.and.or.wildlife Fish.and.or.wildlife  3.6186739
## SP.Const                       SP.Const  3.3320497
## Vegetation                   Vegetation  3.1658748
## Fin.Const                     Fin.Const  3.1500258
## Inst.Const                   Inst.Const  3.1257100
## Recreational               Recreational  2.0510036
## Inform.Const               Inform.Const  2.0225225
## Cultural.Historical   Cultural.Historical  2.0011826
## Soils                             Soils  1.6400689
## Water                             Water  1.3826820
## Visual.Scenic             Visual.Scenic  0.4770361
## AgencyRole                   AgencyRole  0.2955878
## Paleontological         Paleontological  0.0000000
## Minerals.Geology       Minerals.Geology  0.0000000
## Fossil.Fuels               Fossil.Fuels  0.0000000
## Air                                 Air  0.0000000
```

```
## Other.                          Other.  0.0000000
```

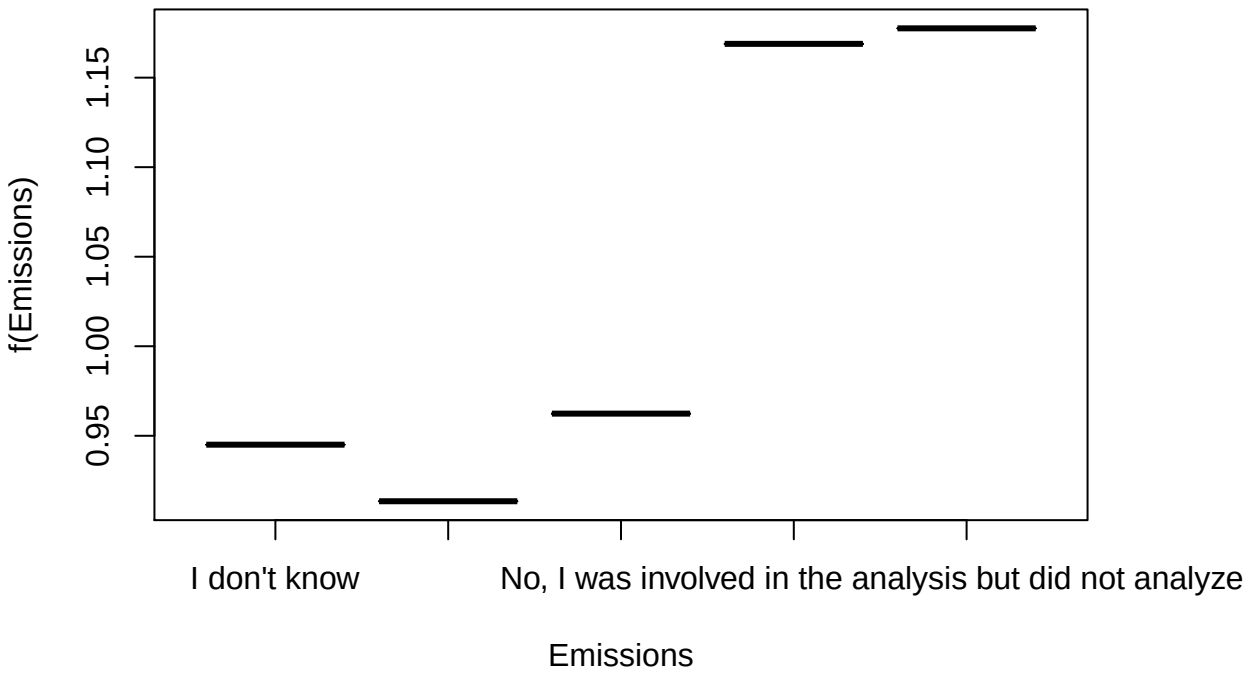
```
plot(clm.gbm.2, i ="raupward10")
lablist1 <- c("Land management planning", "Fire and fuels",
              "Minerals and energy", "Restoration", "Recreation management",
              "Fish and wildlife managment", "Invasive plants", "Range management",
              "Water management", "Other")
axis(1, at=seq(1, 10, by=1), labels = FALSE)
text(seq(1, 10, by=1), par("usr")[3] - 0.05, labels = lablist1, srt = 30, pos = 1, xpd = TRUE, cex = 0.
```



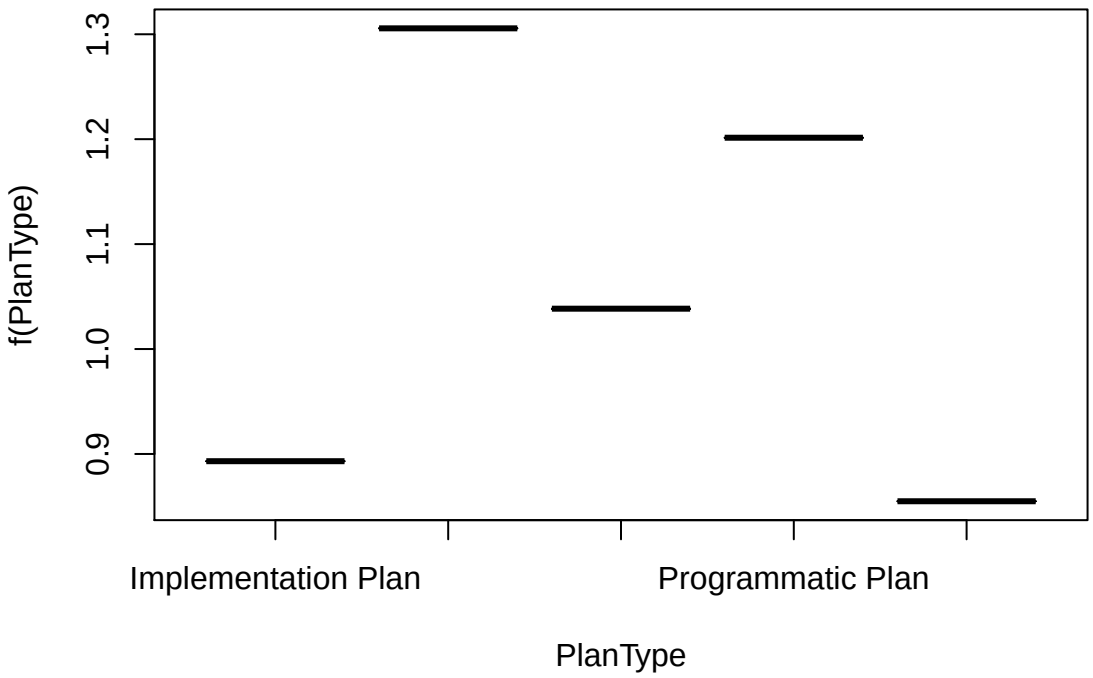

```
plot(clm.gbm.2, i="AgencyWorked")
lablist2 <- as.vector(c("BLM", "FWS", "FS", "NPS", "Other","State Agency"))
axis(1, at=seq(1, 6, by=1), labels = FALSE)
text(seq(1, 6, by=1), par("usr")[3] - 0.05,labels = lablist2, srt = 45, pos = 1, xpd = TRUE)
```



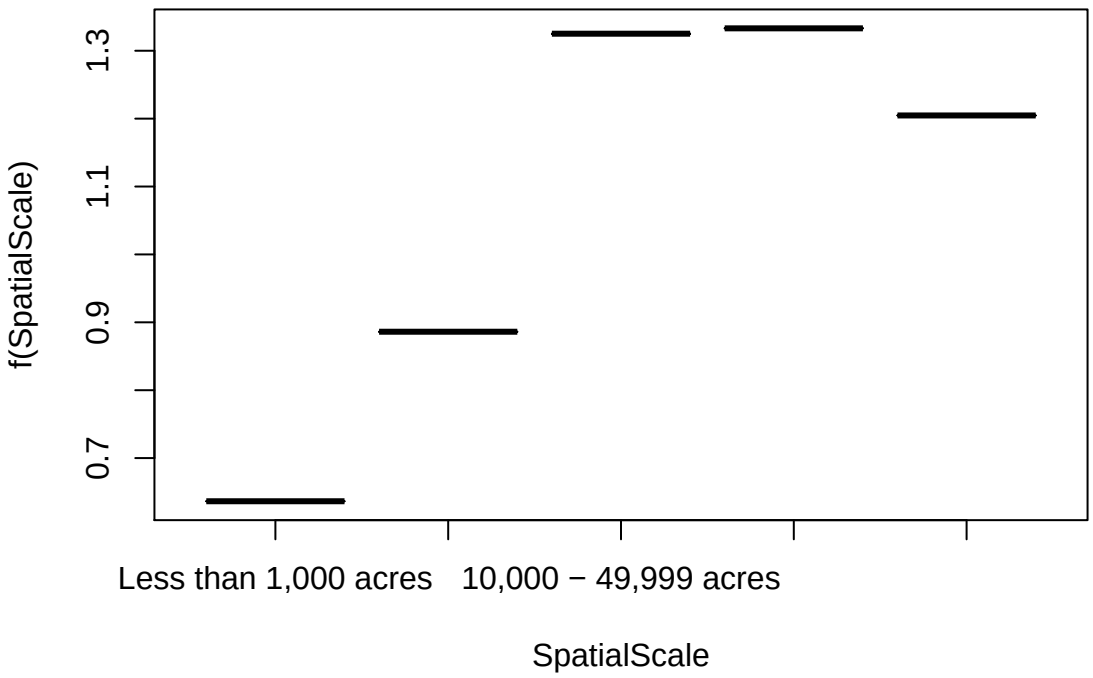

```
plot(clm.gbm.2, i="Emissions")
```




```
plot(clm.gbm.2, i="PlanType")
```




```
plot(clm.gbm.2, i="SpatialScale")
```



#‘ Gradient Bosted Regression of climate use

```
use.model <- dcast(decision.climate, ID + DecisionNum + DecisionName +
              AgencyWorked + ClimRelevance ~ ClimDataset, value.var = "Used")
colnames(use.model)[6:11] <- c("HistObs", "InterpSurf", "ProjAvg","ProjExt",
                               "SeasFcst", "CurCondSum")

use.model$used <- with(use.model, ifelse(HistObs == "Yes" | InterpSurf =="Yes" |
                         ProjAvg == "Yes" | ProjExt == "Yes" |
                         SeasFcst == "Yes" | CurCondSum == "Yes", 1, 0))
um <- use.model[,c(1,2,12)]
#combine used data with predictors
um.preds <- merge(decRSPNECRAT, um, by=c("ID", "DecisionNum"))
um.preds[,18:29] <- lapply(um.preds[,18:29], as.factor)
levels(um.preds$CCImplications) <- c("I don't know", "No, I was involved in the analysis but did not ana
                               "No, I was involved in the analysis but did not analyze these effects."
                               "Yes, I was involved in the analysis and analyzed these effects.")
#prepare dataset for gbm
um.preds <- um.preds[,c(31,4,6:30)]
um.preds <- um.preds[!is.na(um.preds$used),]

cluse.gbm.1 <- gbm(used ~ . ,
                       data = um.preds, distribution = "bernoulli",
                       n.trees=1000, shrinkage = 0.05,
                       interaction.depth = 5, bag.fraction = 0.10,
                       train.fraction = 0.6, mFeatures = 3, n.minobsinnode = 5,
                       cv.folds = 4, keep.data= TRUE, verbose = TRUE,
```

```
                        n.cores =1)
best.iter <- gbm.perf(cluse.gbm.1,method="test",plot.it=TRUE)

cluse.gbm.2 <- gbm(used ~ . ,
                        data = um.preds, distribution = "bernoulli",
                        n.trees=1000, shrinkage = 0.01,
                        interaction.depth = 5, bag.fraction = 0.10,
                        train.fraction = 0.6, mFeatures = 3, n.minobsinnode = 5,
                        cv.folds = 4, keep.data= TRUE, verbose = TRUE,
                        n.cores =1)
best.iter <- gbm.perf(cluse.gbm.2,method="test",plot.it=TRUE)

cluse.gbm.3 <- gbm(used ~ . ,
                        data = um.preds, distribution = "bernoulli",
                        n.trees=1000, shrinkage = 0.01,
                        interaction.depth = 3, bag.fraction = 0.10,
                        train.fraction = 0.6, mFeatures = 3, n.minobsinnode = 5,
                        cv.folds = 4, keep.data= TRUE, verbose = TRUE,
                        n.cores =1)
best.iter <- gbm.perf(cluse.gbm.3,method="test",plot.it=TRUE)

summary(cluse.gbm.3)
par(mfrow=c(2,2), xaxt = "n")
plot(cluse.gbm.3, i ="raupward10")
lablist1 <- c("Land management planning", "Fire and fuels",
            "Minerals and energy", "Restoration", "Recreation management",
            "Fish and wildlife managment", "Invasive plants", "Range management",
            "Water management", "Other")
axis(1, at=seq(1, 10, by=1), labels = FALSE)
text(seq(1, 10, by=1), par("usr")[3] - 0.05, labels = lablist1, srt = 30, pos = 1, xpd = TRUE, cex = 0.
plot(cluse.gbm.3, i="CCImplications")
lablist4 <- c("I don't know", "No", "Yes")
axis(1, at=seq(1, 6, by=1), labels = FALSE)
text(seq(1, 6, by=1), par("usr")[3] - 0.05,labels = lablist4, srt = 45, pos = 1, xpd = TRUE)

plot(cluse.gbm.3 i="Emissions")
plot(cluse.gbm.3, i="PlanType")
plot(cluse.gbm.3, i="SpatialScale")

‘‘‘
```