

Lynx EE figures

Jonathan Cummings

February 26, 2016

Contents

Introduction	1
Lynx Status	1
Representation	1
Redundancy	2
Resiliency	6

Introduction

This R markdown file contains code to process the results of the October 13-15 Lynx expert elicitation workshop. The input data are the responses collected in response to the questions asked of the expert panel at the workshop. See the Canada Lynx Expert Elicitation Workshop Report for a description of the workshop and the questions asked. What follows is a description of the r code used to generate summary figures to accompany the workshop report.

Lynx Status

Expert responses were collected for each of the 3Rs used in species status assessment. The code below is grouped by the 3Rs, Representation, Redundancy, and Resiliency.

R initialization

Prior to analyzing the data r is initialized with packages needed for the response summaries. *####* Load packages

```
# Clear workspace
rm(list=ls())

# Load libraries
# ----Packages for Manipulating & Visualizing Data-----
library(reshape2) # package for manipulating data
library(ggplot2) # package for plotting data
```

Representation

There are no figures associated with the responses to the representation questions.

Redundancy

This section processes the responses to redundancy questions for lynx populations in the lower 48 US states.

```
# Responses to redundancy question 3, entered by hand from the response files
redundancy3<-data.frame(expert<-c(1,2,3,4,5,6,7,8,9,10),Q3<-c(1,1,10,10,0.1,0.01,1,1,1,5))
# Name the data columns for data on question 3
names(redundancy3)<-c("Expert","Q3")

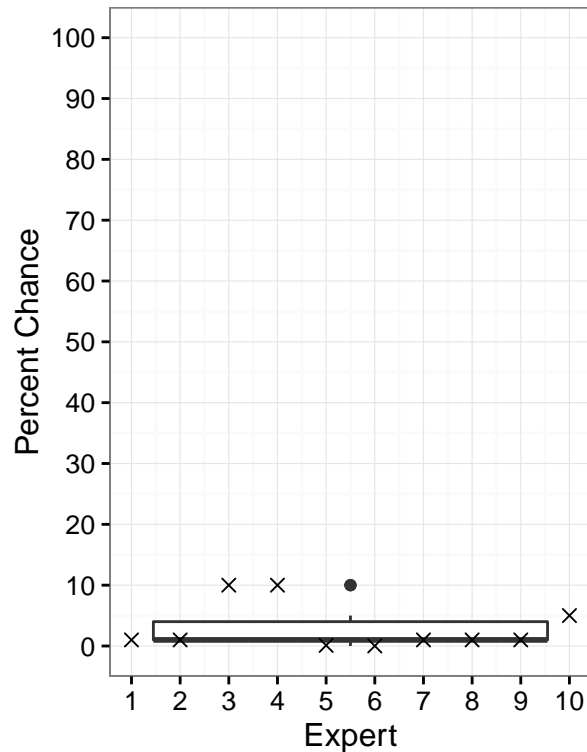
# Create the plot for the responses to redundancy question 3
plot.Q3<-ggplot(redundancy3,aes(Expert,Q3))
Q3<-plot.Q3 + geom_boxplot() + geom_point(size=2,shape=4) +
  scale_x_continuous(breaks=seq(1,10,1)) + theme_bw()+
  ylab("Percent Chance")+xlab("Expert")+scale_y_continuous(limits=c(0,100),breaks = seq(0,100,10))+
  ggtitle("Q3. Chance a geographic unit is eliminated\nby a catastrophic event") +
  coord_fixed(0.125)

# ggplot is the plotting function, geom_boxplot creates the boxplot and geom_jitter adds data points to

#Plot
Q3
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

Q3. Chance a geographic unit is eliminated by a catastrophic event



```

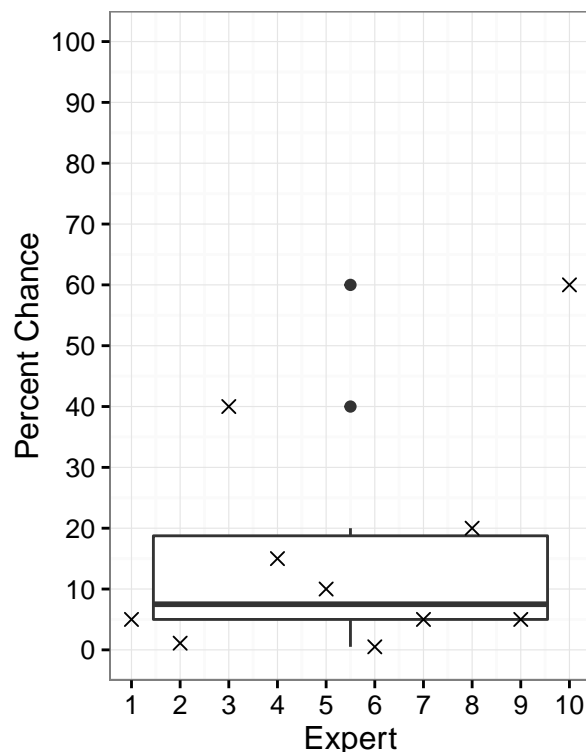
# Responses to redundancy question 4, entered by hand from the response files
redundancy4<-data.frame(expert<-c(1,2,3,4,5,6,7,8,9,10),Q4<-c(5,1.1,40,15,10,0.5,5,20,5,60))
# Name the data columns for data on question 4
names(redundancy4)<-c("Expert","Q4")

# Create the plot for the responses to redundancy question 4
plot.Q4<-ggplot(redundancy4,aes(expert,Q4))
Q4<-plot.Q4 + geom_boxplot() + geom_point(size=2,shape=4)+
  scale_x_continuous(breaks=seq(1,10,1)) + theme_bw() +
  ylab("Percent Chance")+xlab("Expert")+scale_y_continuous(limits=c(0,100),breaks = seq(0,100,10))+
  ggtitle("Q4. Chance a geographic unit is eliminated\nby a series of catastrophic events") +
  coord_fixed(0.125)
# Same plot a for question 3, different data
# Plot
Q4

```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

Q4. Chance a geographic unit is eliminated by a series of catastrophic events



```

# Responses to redundancy question 5, entered by hand from the response files
redundancy5<-data.frame(expert<-rep(c(1,2,3,4,5,6,7,8,9,10),4),
  point<-c(rep("longest",10),rep("shortest",10),rep("most likely",10),rep("confid
  value<-c(100,300,60,NA,100,50,25,NA,100,200,10,15,15,1,25,20,15,15,20,15,
    40,100,35,10,50,30,20,50,30,55,50,80,5,100,75,90,90,40,50,50))
# Name the data columns for data on question 4
names(redundancy5)<-c("expert","point","value")

```

```

# Manipulate data to ease calculations
redundancy.df<-dcast(redundancy5,expert~point)

# Compute 95% confidence bound for shortest time period
redundancy.df$shortest.95<-round(((redundancy.df$shortest-redundancy.df$most)/redundancy.df$confidence)*.95)
# Compute 95% confidence bound for longest time period
redundancy.df$longest.95<- round(((redundancy.df$longest-redundancy.df$most)/redundancy.df$confidence)*.95)

# Manipulate data again to ease figure production
redundancy5<-melt(redundancy.df,id.vars="expert")
redundancy5$value[redundancy5$value<0]<-0
redundancy5$variable<-factor(redundancy5$variable)
redundancy5$variable<-factor(redundancy5$variable,levels(redundancy5$variable)[c(6,4,3,2,5,1)])

# Create the boxplot for the responses to redundancy question 5
plot.Q5.box<-ggplot(redundancy5[c(11:40),],aes(x=expert,y=value))
Q5.box<-plot.Q5.box + geom_boxplot() +geom_point(size=2,shape=4) + ylab("Years until reestablishment") +
  scale_y_continuous(limits=c(0,300),breaks = seq(0,300,25)) + scale_x_continuous(breaks = seq(0,10))+f
#Plot
Q5.box

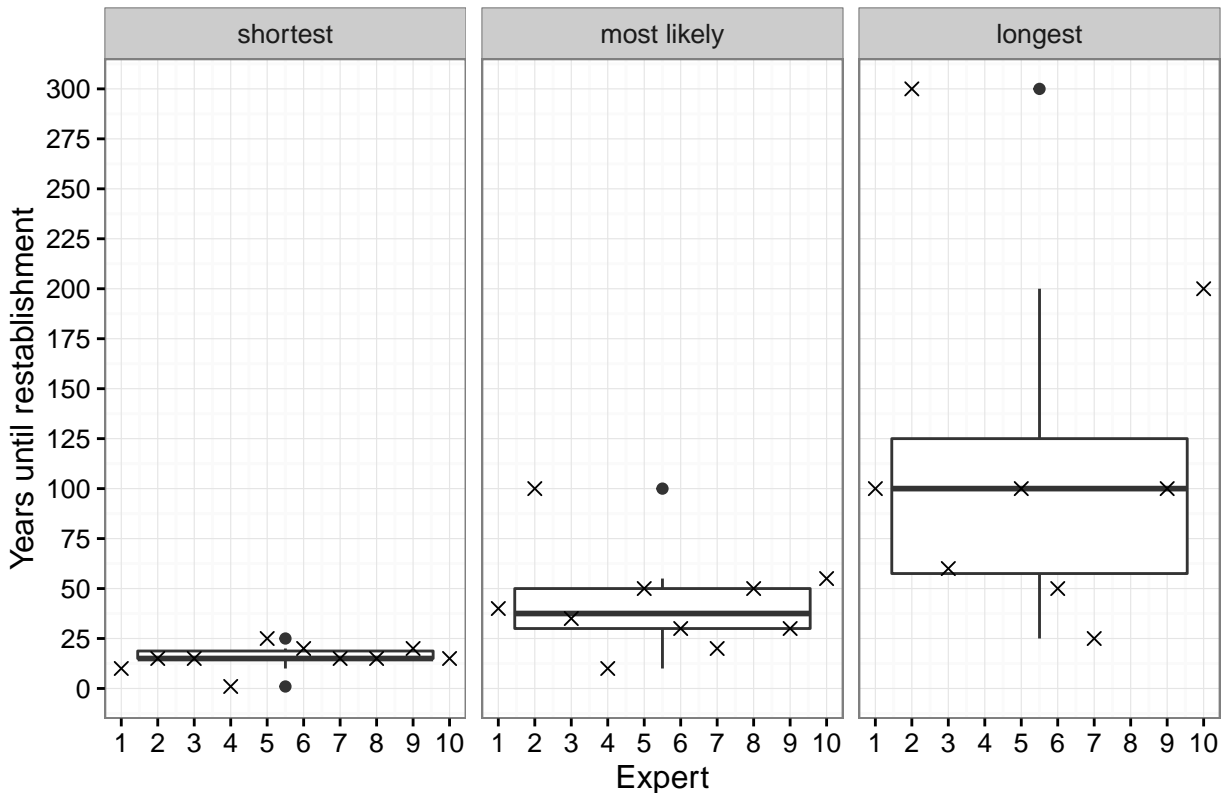
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

Q5. Years for reestablishment by time period



```
# replace large and NA values with 350 for plotting purposes
```

```
redundancy5$value[is.na(redundancy5$value)]<-350
```

```
redundancy5$value[redundancy5$value>350]<-350
```

```
# Create the 95% confidence interval lineplot for the responses to redundancy question 5
```

```
plot.Q5<-ggplot(redundancy5[c(21:30,41:60),],aes(x=expert,y=value,group=variable))
```

```
Q5<-plot.Q5 + geom_line(aes(colour=variable)) + geom_point(aes(colour=variable)) +
```

```
  ylab("Years until reestablishment") +
```

```
  scale_x_continuous(limits=c(1,10),breaks = seq(1,10,1))+theme_bw() +
```

```
  scale_colour_discrete(name="time period",labels=c("longest plausible","most plausible",
```

```
    "shortest plausible")) +
```

```
  scale_y_continuous(limits=c(0,350),breaks = seq(0,350,25)) + annotate("text", x = 3.4, y = 345,
```

```
    label = "never", colour="red") +
```

```
  annotate("text", x = 4.4, y = 345,label = "510", colour="red") +
```

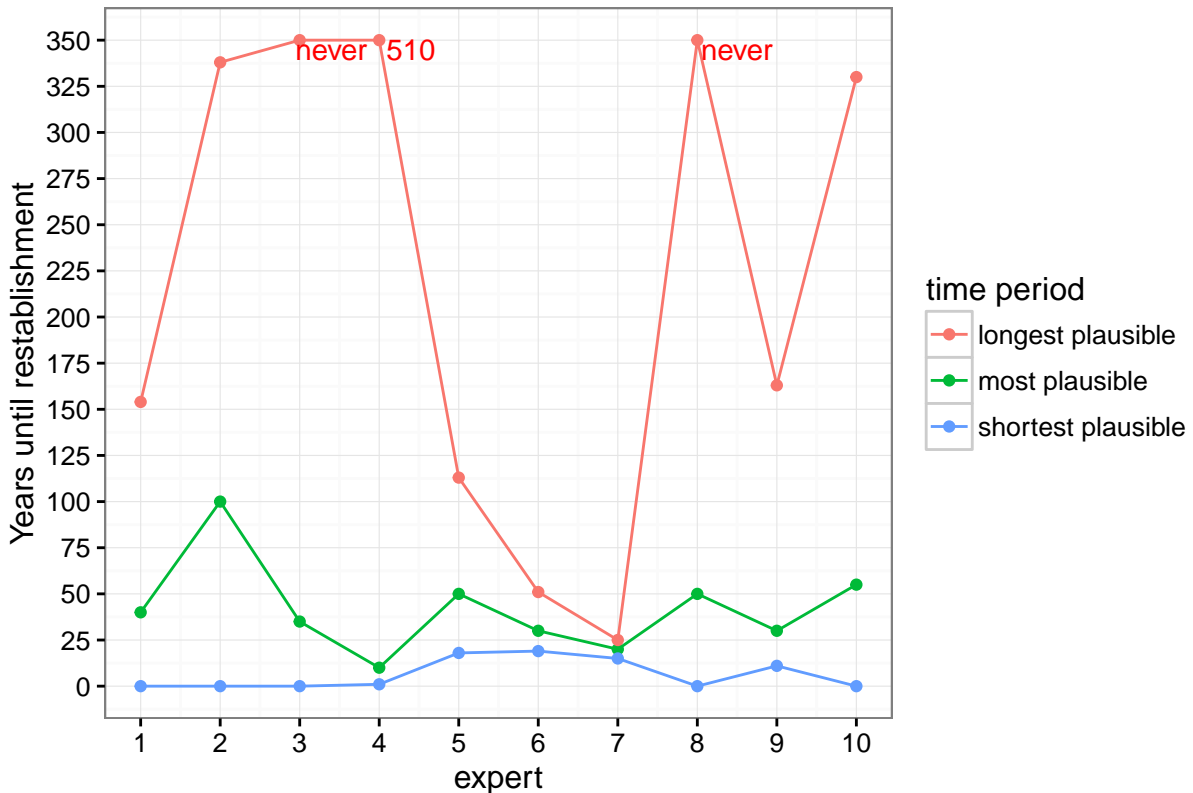
```
  annotate("text", x = 8.5, y = 345,label = "never", colour="red") +
```

```
  ggtitle("Q5. Years for reestablishment by expert - 95% confidence interval")
```

```
#Plot
```

```
Q5
```

5. Years for reestablishment by expert – 95% confidence interval



Resiliency

This section processes the responses to the resiliency questions for lynx populations in the lower 48 US states.

Resiliency by Geographic Unit

This section reads in the resiliency response file, reorganizes that data, and produces a summary figure of the responses for each geographic unit.

```
# read in resiliency data and organize it
resiliency.data<-read.csv("Resiliency Responses.csv") # read in data file from workhop
# name the columns of data
names(resiliency.data)<-c("Geographic.Unit","Expert","Time.Period","Highest","Most Likely","Lowest")
# reorganize the data and reassign names
resiliency.data<-melt(resiliency.data,c("Geographic.Unit","Expert","Time.Period"))
names(resiliency.data)<-c("Geographic.Unit","Expert","Time.Period","Probability","value")
# make the expert column a factor for use in plotting
resiliency.data$Expert<-as.factor(resiliency.data$Expert)

# Filter by geographic unit
r.ME<-resiliency.data[resiliency.data$Geographic.Unit=="Maine/NE",]
r.MN<-resiliency.data[resiliency.data$Geographic.Unit=="MN/Lakes States",]
r.MT<-resiliency.data[resiliency.data$Geographic.Unit=="Northwest MT/ NE Idaho",]
r.WA<-resiliency.data[resiliency.data$Geographic.Unit=="Washington",]
```

```

r.GYA<-resiliency.data[resiliency.data$Geographic.Unit=="Greater Yellowstone Area",]
r.CO<-resiliency.data[resiliency.data$Geographic.Unit=="Colorado",]

# Create plot for each geographic unit
#Maine/NE (ME)
res.plot.ME <- ggplot(r.ME, aes(y=value, x=Time.Period,)) + theme_bw() +
  scale_x_continuous(breaks=c(2015,2025,2050,2100)) + scale_y_continuous(limits=c(0,
  1),breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) +
  ylab("Probability of Persistence")+xlab("Year") +
  stat_summary(geom="ribbon", fun.ymin="min", fun.ymax="max",fill="grey90") +
  geom_point(data=r.ME,aes(x=Time.Period,y=value, shape=Probability,
  colour=Probability,group=interaction(Probability,Expert))) +
  scale_shape_manual(values=c(5,19,5)) +
  stat_summary(data=r.ME[r.ME$Probability=="Highest",],geom="line",
  fun.y="median",linetype="dashed",colour="red") +
  stat_summary(data=r.ME[r.ME$Probability=="Lowest",],geom="line",
  fun.y="median",linetype="dashed",colour="blue") +
  stat_summary(data=r.ME[r.ME$Probability=="Most Likely",],geom="line",
  fun.y="median",linetype="dashed",colour="green") +
  stat_summary(data=r.ME[r.ME$Probability=="Most Likely",],geom="point",
  fun.y="median",shape=4,size=4) + ggtitle("Maine")
#MN/Lakes States(MN)
res.plot.MN <- ggplot(r.MN, aes(y=value, x=Time.Period,)) + theme_bw() +
  scale_x_continuous(breaks=c(2015,2025,2050,2100)) + scale_y_continuous(limits=c(0,
  1),breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) +
  ylab("Probability of Persistence") + xlab("Year") + stat_summary(geom="ribbon",
  fun.ymin="min", fun.ymax="max",fill="grey90") +
  geom_point(data=r.MN,aes(x=Time.Period,y=value, shape=Probability,
  colour=Probability,group=interaction(Probability,Expert))) +
  scale_shape_manual(values=c(5,19,5)) +
  stat_summary(data=r.MN[r.MN$Probability=="Highest",],geom="line",
  fun.y="median",linetype="dashed",colour="red") +
  stat_summary(data=r.MN[r.MN$Probability=="Lowest",],geom="line",
  fun.y="median",linetype="dashed",colour="blue") +
  stat_summary(data=r.MN[r.MN$Probability=="Most Likely",],geom="line",
  fun.y="median",linetype="dashed",colour="green") +
  stat_summary(data=r.MN[r.MN$Probability=="Most Likely",],geom="point",
  fun.y="median",shape=4,size=4) + ggtitle("Minnesota")
#Northwest MT/ NE Idaho
res.plot.MT <- ggplot(r.MT, aes(y=value, x=Time.Period,)) + theme_bw() +
  scale_x_continuous(breaks=c(2015,2025,2050,2100))+scale_y_continuous(limits=c(0,
  1),breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) +
  ylab("Probability of Persistence")+xlab("Year") + stat_summary(geom="ribbon",
  fun.ymin="min", fun.ymax="max",fill="grey90") + geom_point(data=r.MT,
  aes(x=Time.Period,y=value, shape=Probability, colour=Probability,
  group=interaction(Probability,Expert))) + scale_shape_manual(values=c(5,19,5)) +
  stat_summary(data=r.MT[r.MT$Probability=="Highest",],geom="line",
  fun.y="median",linetype="dashed",colour="red") +
  stat_summary(data=r.MT[r.MT$Probability=="Lowest",],geom="line",
  fun.y="median",linetype="dashed",colour="blue") +
  stat_summary(data=r.MT[r.MT$Probability=="Most Likely",],geom="line",
  fun.y="median",linetype="dashed",colour="green") +
  stat_summary(data=r.MT[r.MT$Probability=="Most Likely",],geom="point",

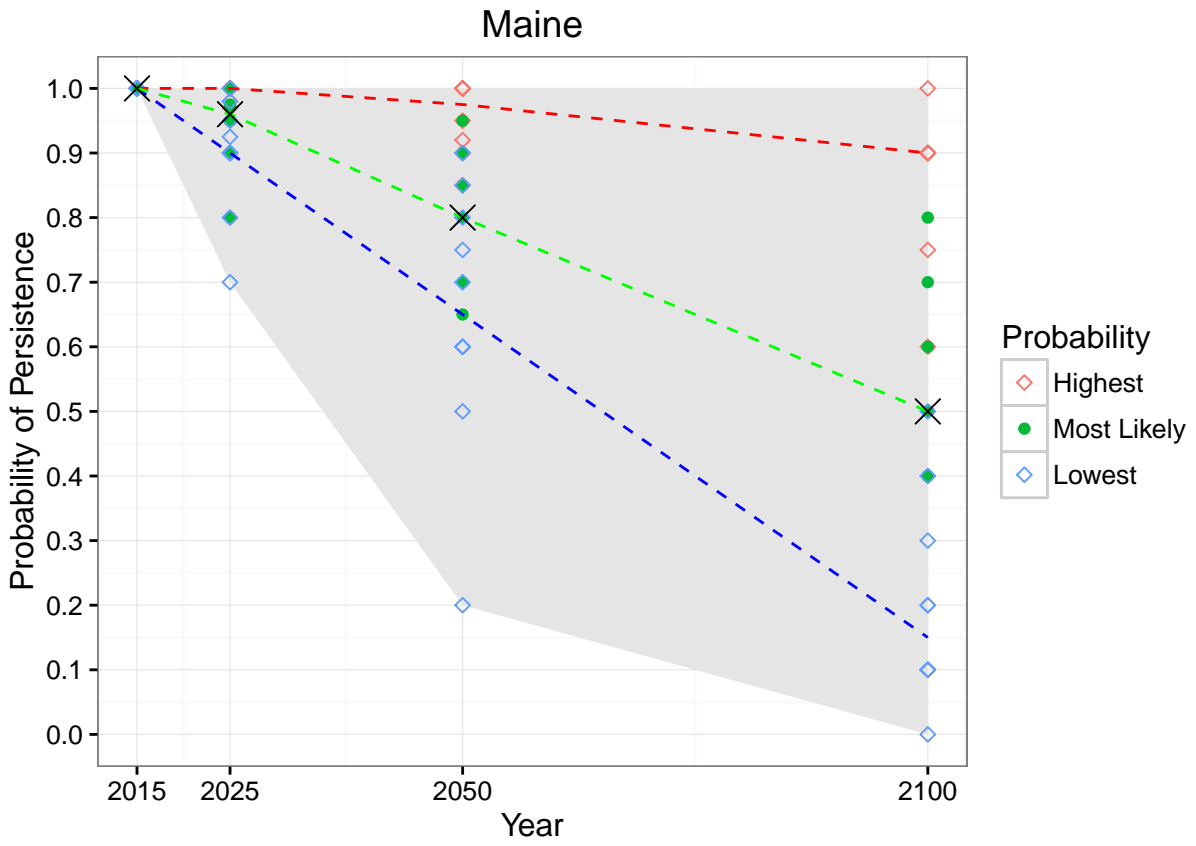
```

```

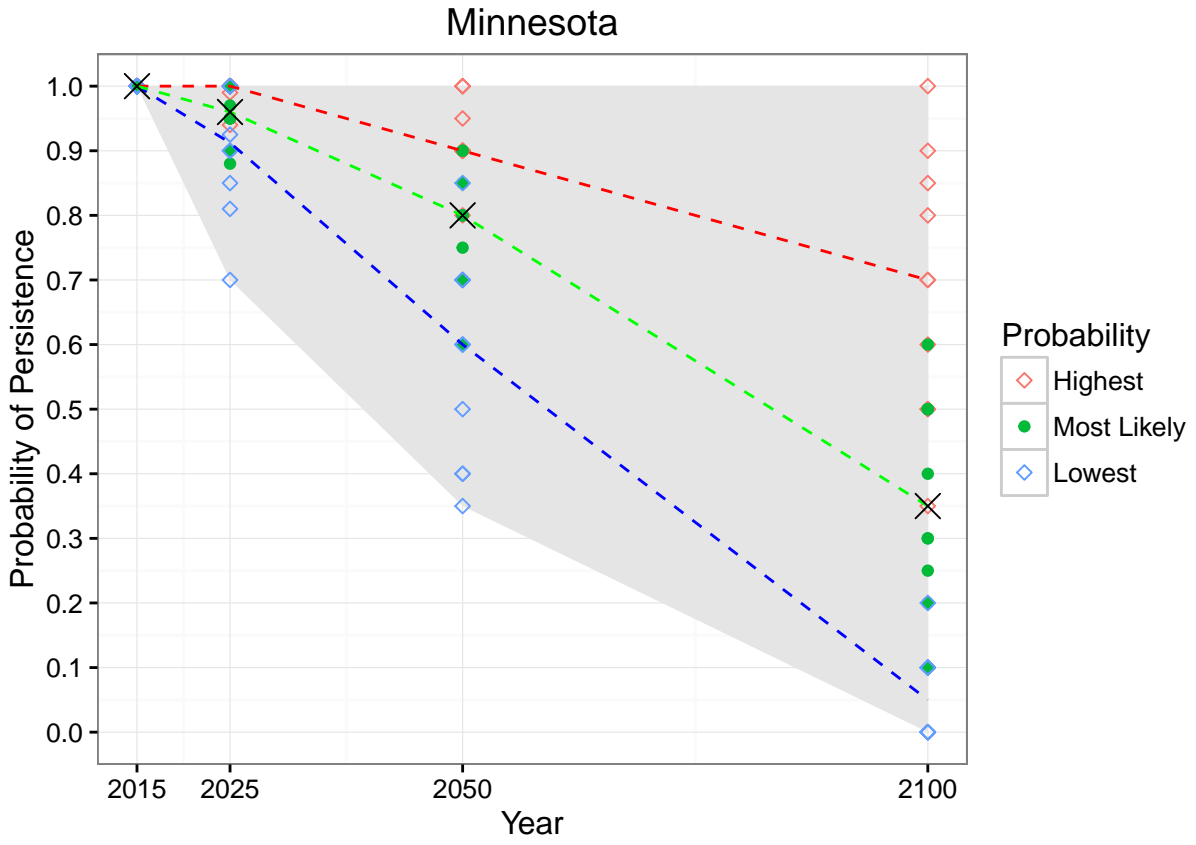
    fun.y="median",shape=4,size=4) + ggtitle("NW Montana & NE Idaho")
#Washington
res.plot.WA <- ggplot(r.WA, aes(y=value, x=Time.Period,)) + theme_bw() +
  scale_x_continuous(breaks=c(2015,2025,2050,2100)) + scale_y_continuous(limits=c(0,
  1),breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) +
  ylab("Probability of Persistence") + xlab("Year") +
  stat_summary(geom="ribbon", fun.ymin="min", fun.ymax="max",fill="grey90") +
  geom_point(data=r.WA,aes(x=Time.Period,y=value, shape=Probability,
  colour=Probability,group=interaction(Probability,Expert))) +
  scale_shape_manual(values=c(5,19,5)) +
  stat_summary(data=r.WA[r.WA$Probability=="Highest",],geom="line",
  fun.y="median",linetype="dashed",colour="red") +
  stat_summary(data=r.WA[r.WA$Probability=="Lowest",],geom="line",
  fun.y="median",linetype="dashed",colour="blue") +
  stat_summary(data=r.WA[r.WA$Probability=="Most Likely",],geom="line",
  fun.y="median",linetype="dashed",colour="green")+
  stat_summary(data=r.WA[r.WA$Probability=="Most Likely",],geom="point",
  fun.y="median",shape=4,size=4) + ggtitle("Washington")
#Greater Yellowstone Area
res.plot.GYA <- ggplot(r.GYA, aes(y=value, x=Time.Period,)) + theme_bw() +
  scale_x_continuous(breaks=c(2015,2025,2050,2100)) + scale_y_continuous(limits=c(0,
  1),breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) +
  ylab("Probability of Persistence")+xlab("Year") + stat_summary(geom="ribbon",
  fun.ymin="min", fun.ymax="max",fill="grey90") + geom_point(data=r.GYA,
  aes(x=Time.Period,y=value, shape=Probability,colour=Probability,
  group=interaction(Probability,Expert))) + scale_shape_manual(values=c(5,19,5)) +
  stat_summary(data=r.GYA[r.GYA$Probability=="Highest",],geom="line",
  fun.y="median",linetype="dashed",colour="red") +
  stat_summary(data=r.GYA[r.GYA$Probability=="Lowest",],geom="line",
  fun.y="median",linetype="dashed",colour="blue") +
  stat_summary(data=r.GYA[r.GYA$Probability=="Most Likely",],geom="line",
  fun.y="median",linetype="dashed",colour="green") +
  stat_summary(data=r.GYA[r.GYA$Probability=="Most Likely",],geom="point",
  fun.y="median",shape=4,size=4) + ggtitle("Greater Yellowstone Area")
#Colorado
res.plot.CO <- ggplot(r.CO, aes(y=value, x=Time.Period,)) + theme_bw() +
  scale_x_continuous(breaks=c(2015,2025,2050,2100)) + scale_y_continuous(limits=c(0,
  1),breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)) +
  ylab("Probability of Persistence") + xlab("Year") +
  stat_summary(geom="ribbon", fun.ymin="min", fun.ymax="max",fill="grey90") +
  geom_point(data=r.CO,aes(x=Time.Period,y=value, shape=Probability,
  colour=Probability,group=interaction(Probability,Expert))) +
  scale_shape_manual(values=c(5,19,5)) +
  stat_summary(data=r.CO[r.CO$Probability=="Highest",],geom="line",
  fun.y="median",linetype="dashed",colour="red") +
  stat_summary(data=r.CO[r.CO$Probability=="Lowest",],geom="line",
  fun.y="median",linetype="dashed",colour="blue") +
  stat_summary(data=r.CO[r.CO$Probability=="Most Likely",],geom="line",
  fun.y="median",linetype="dashed",colour="green") +
  stat_summary(data=r.CO[r.CO$Probability=="Most Likely",],geom="point",
  fun.y="median",shape=4,size=4) + ggtitle("Colorado")
#show plots

```

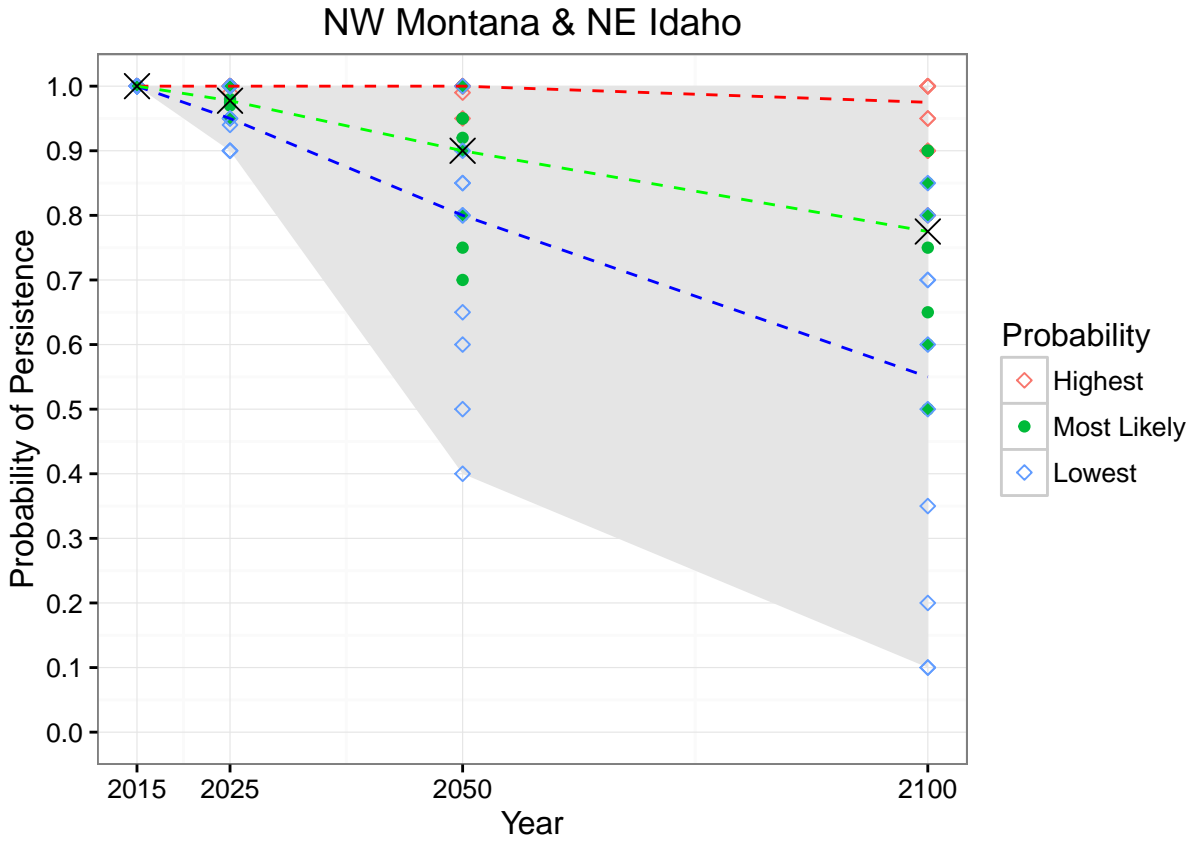

res.plot.ME



res.plot.MN

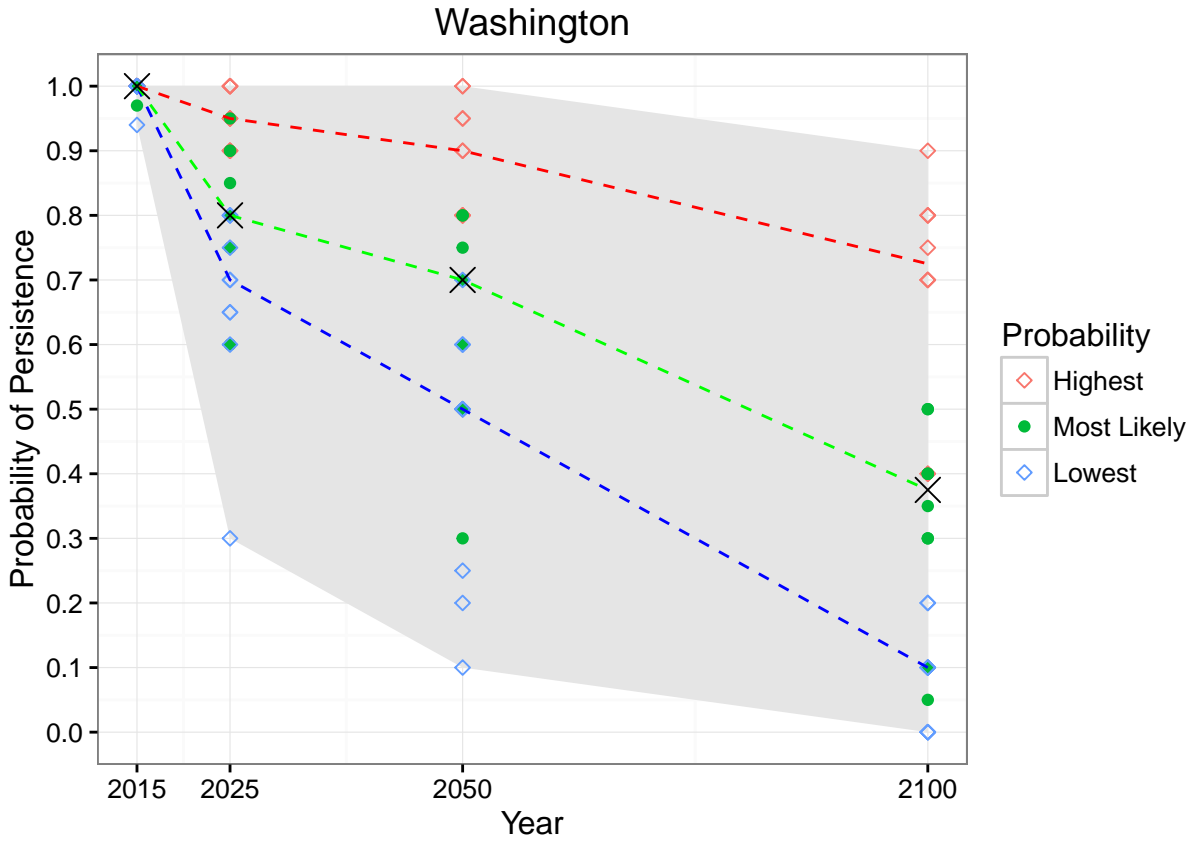


res.plot.MT



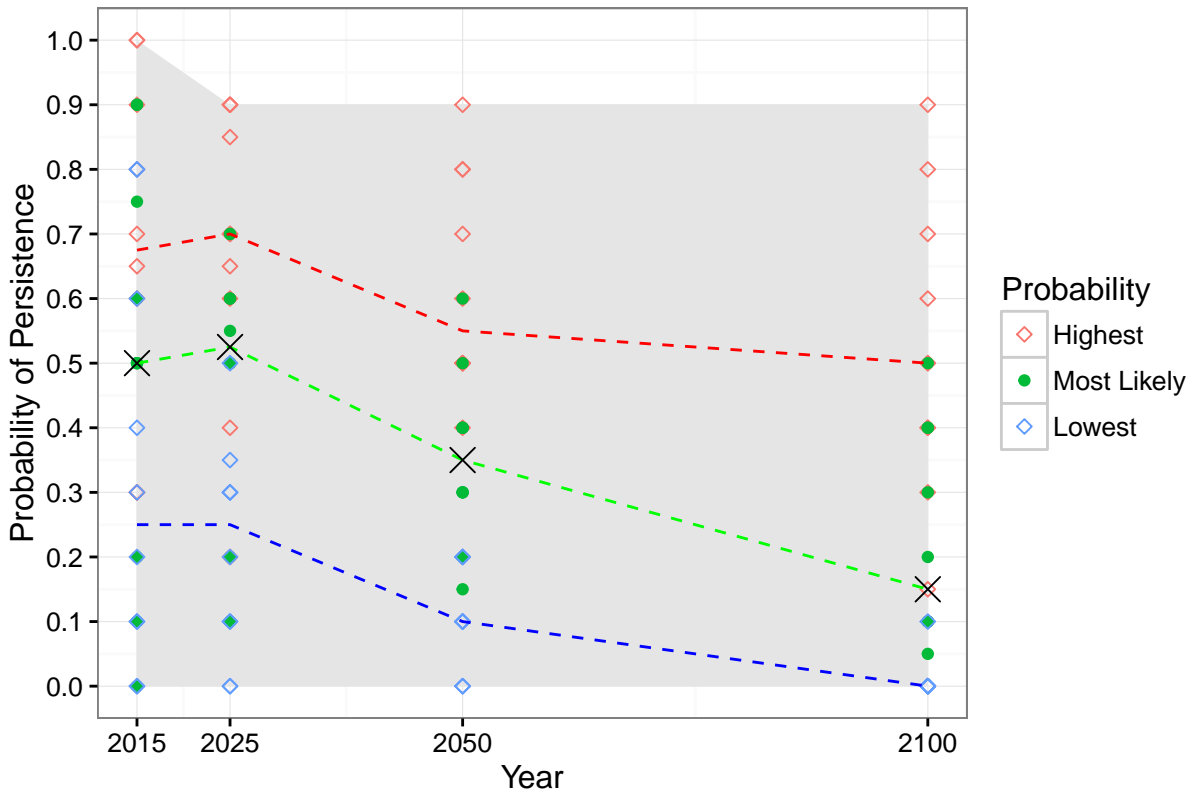
```
res.plot.WA
```

```
## Warning: Removed 3 rows containing non-finite values (stat_summary).
## Warning: Removed 1 rows containing non-finite values (stat_summary).
## Warning: Removed 1 rows containing non-finite values (stat_summary).
## Warning: Removed 1 rows containing non-finite values (stat_summary).
## Warning: Removed 1 rows containing non-finite values (stat_summary).
## Warning: Removed 3 rows containing missing values (geom_point).
```

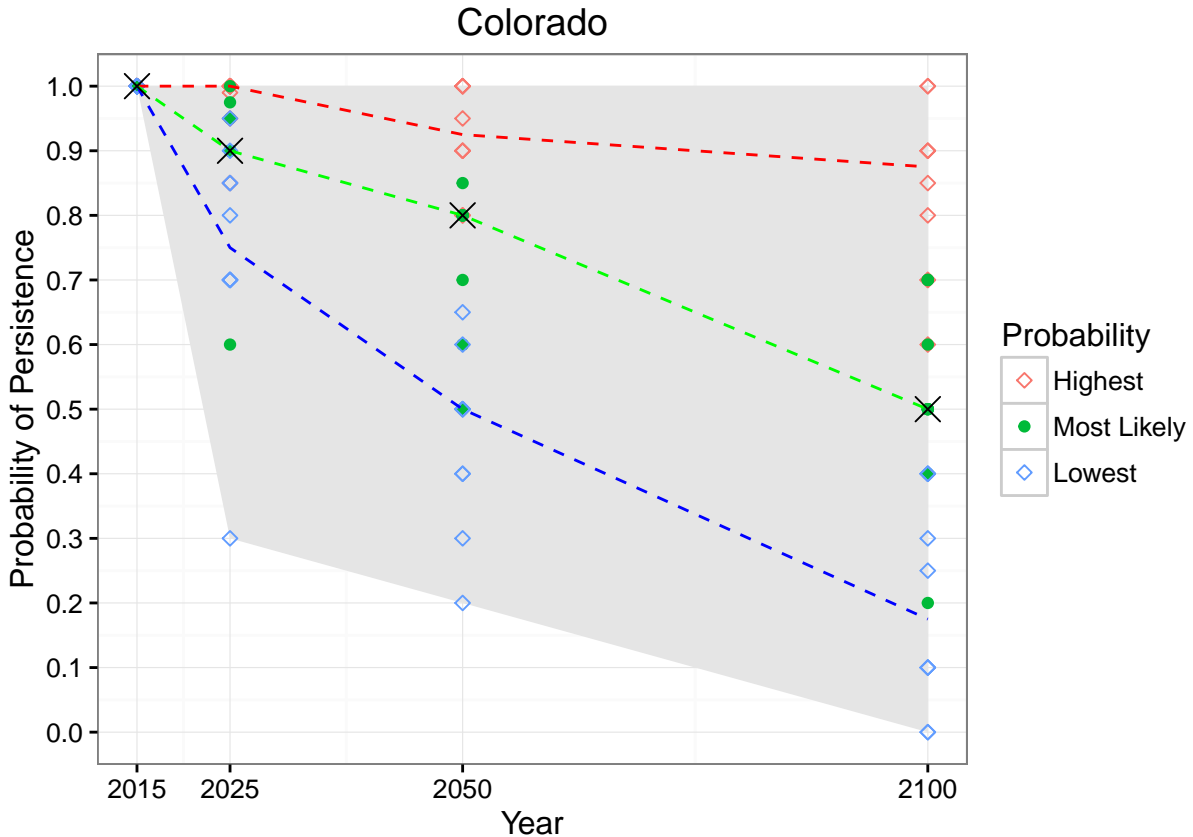


res.plot.GYA

Greater Yellowstone Area



res.plot.CO



Resiliency across Geographic Units This section uses the binomial probability of persistence in each unit and combines them to determine the resulting probability of persistence across units. Some terminology: “Highest_High” is the probability of persistence generated by selecting the highest probability of persistence across experts from the highest probability response in each geographic unit.

“Median_High” is the probability of persistence generated by selecting the median probability of persistence across experts from the highest probability response in each geographic unit. “Highest_Likely” is the probability of persistence generated by selecting the highest probability of persistence across experts from the most likely probability response in each geographic unit. “Median_Likely” is the probability of persistence generated by selecting the median probability of persistence across experts from the most likely probability response in each geographic unit. “Lowest_Likely” is the probability of persistence generated by selecting the lowest probability of persistence across experts from the most likely probability response in each geographic unit. “Median_Low” is the probability of persistence generated by selecting the median probability of persistence across experts from the lowest probability response in each geographic unit. “Lowest_Low” is the probability of persistence generated by selecting the lowest probability of persistence across experts from the lowest probability response in each geographic unit.

```
# Create summary graph across geographic units
# create empty vector to bind results to
r.summary<-0

# loop through data to compute median of response and find max and min response across experts for each
for (i in unique(resiliency.data$Geographic.Unit)){ # loop through geographic units
  for (j in unique(resiliency.data$Time.Period)){ # loop through time periods
    for (k in unique(resiliency.data$Probability)){ # loop through probability (high, low, most likely)
      # Create new array containing the geographic unit, time period, probability, and the median, max,
      r.summary<-rbind(r.summary,c(i,j,k,"median",
        median(resiliency.data[resiliency.data$Geographic.Unit==i&
```

```

    resiliency.data$Time.Period==j&resiliency.data$Probability==k,"value"),
    na.rm=T),"max",
  max(resiliency.data[resiliency.data$Geographic.Unit==i&
    resiliency.data$Time.Period==j&resiliency.data$Probability==k,"value"],
    na.rm=T),"min",
  min(resiliency.data[resiliency.data$Geographic.Unit==i&
    resiliency.data$Time.Period==j&resiliency.data$Probability==k,"value"],
    na.rm=T)))
  }
}
}

# Remove empty first row
r.summary<-r.summary[-1,]
# Convert to data.frame and name columns
r.summary<-data.frame(Geographic.Unit=r.summary[,1],Time.Period=r.summary[,2],
  Probability=r.summary[,3],median=r.summary[,5],max=r.summary[,7],min=r.summary[,9])
# Organize for easier processing and graphing
r.summary<-melt(r.summary,id.vars=c("Geographic.Unit","Time.Period","Probability"),
  measure.vars=c("median","max","min"))

```

```

## Warning: attributes are not identical across measure variables; they will
## be dropped

```

```

# Remove unwanted rows
r.summary.big<-r.summary[!(r.summary$Probability=="Highest"&r.summary$variable=="min"),]
r.summary.big<-r.summary.big[!(r.summary.big$Probability=="Lowest"&r.summary.big$variable=="max"),]
# order the data by time period, variable, and probability
r.summary.big<-r.summary.big[order(r.summary.big$Time.Period,r.summary.big$variable,r.summary.big$Probab
# number the rows
row.names(r.summary.big)<-1:nrow(r.summary.big)

# Compute the independent binomial probabilities of a total number of persisting geographic units based
convolve.binomial <- function(p) {
  # p is a vector of probabilities of Bernoulli distributions.
  # here p is response for the probability of each geographic unit persisting
  # The convolution of these distributions is returned as a vector
  # `z` where z[i] is the probability of i-1, i=1, 2, ..., length(p)+1.
  n <- length(p) + 1
  z <- c(1, rep(0, n-1))
  sapply(p, function(q) {z <- (1-q)*z + q*(c(0, z[-n])); q})
  z
}

n<-1:nrow(r.summary.big) # variable for rows in results
#Create a list to store results
p.results<-list(name=character(0),p=matrix(NA,max(n)/6,7))
r.summary.big[<-lapply(r.summary.big,as.character) # convert the list structure to character
# Name the results in the first element of the list and assign the results of the convolve function to
for (i in n[seq(1, length(n), 6)]){
  j<-(i+5)/6
  p.results$name[j]<-paste(r.summary.big[i,2],".",r.summary.big[i,3],".",r.summary.big[i,4])
  p.results$p[j,]<-convolve.binomial(as.numeric(r.summary.big[i:(i+5),5]))
}

```

```

}

#Manipulate the results
# flip the resulting persistence probabilities from the convolve function horizontally to order from 6
p.results$p<-p.results$p[,ncol(p.results$p):1]
# Name the columns
colnames(p.results$p)<-c(6,5,4,3,2,1,0)
# Name the rows
rownames(p.results$p)<-p.results$name
# drop the name portion of the list and retain the matrix only
p.results<-p.results$p
# Compute cumulative probabilities of persistence
c.results <- t(apply(p.results, 1, cumsum))

p.results<-melt(p.results) # Reorganize the results
names(p.results)<-c("name", "Units", "Probability") # add names to the columns
# edit the text in the results, extracting the time period
Time.Period <- sub(" .*", "", p.results[,1])
# combine them
p.results<-cbind(p.results,Time.Period)
# edit the text of the first column
p.results[,1]<-gsub(" . ", ". ", p.results[,1])
p.results[,1]<-substr(p.results[,1],6,100)
p.results[,1]<-gsub(" ", "", p.results[,1])
# label the result probability type
p.results$Variable<-rep("Probability",nrow(p.results))

# Repeat for the cumulative probability results
c.results<-melt(c.results)
names(c.results)<-c("name", "Units", "Probability")
Time.Period <- sub(" .*", "", c.results[,1])
c.results<-cbind(c.results,Time.Period)
c.results[,1]<-gsub(" . ", ". ", c.results[,1])
c.results[,1]<-substr(c.results[,1],6,100)
c.results[,1]<-gsub(" ", "", c.results[,1])
c.results$Variable<-rep("Cumulative Probability",nrow(c.results))

# Combine the results
res.summary.data<-rbind(p.results,c.results)
# Convert the unit column to a factor, change the names, and order the factors for the figure
res.summary.data$Units<-as.factor(res.summary.data$Units)
res.summary.data[res.summary.data$name=="Highest.max", "name"]<-"Highest_High"
res.summary.data[res.summary.data$name=="Highest.median", "name"]<-"Median_High"
res.summary.data[res.summary.data$name=="MostLikely.max", "name"]<-"Highest_Likely"
res.summary.data[res.summary.data$name=="MostLikely.median", "name"]<-"Median_Likely"
res.summary.data[res.summary.data$name=="MostLikely.min", "name"]<-"Lowest_Likely"
res.summary.data[res.summary.data$name=="Lowest.median", "name"]<-"Median_Low"
res.summary.data[res.summary.data$name=="Lowest.min", "name"]<-"Lowest_Low"
res.summary.data$name<-factor(res.summary.data$name)
res.summary.data$name<-factor(res.summary.data$name,levels(res.summary.data$name)[c(1,5,2,6,3,7,4)])

# For all summaries
res.summary.data.all<-res.summary.data

```



```

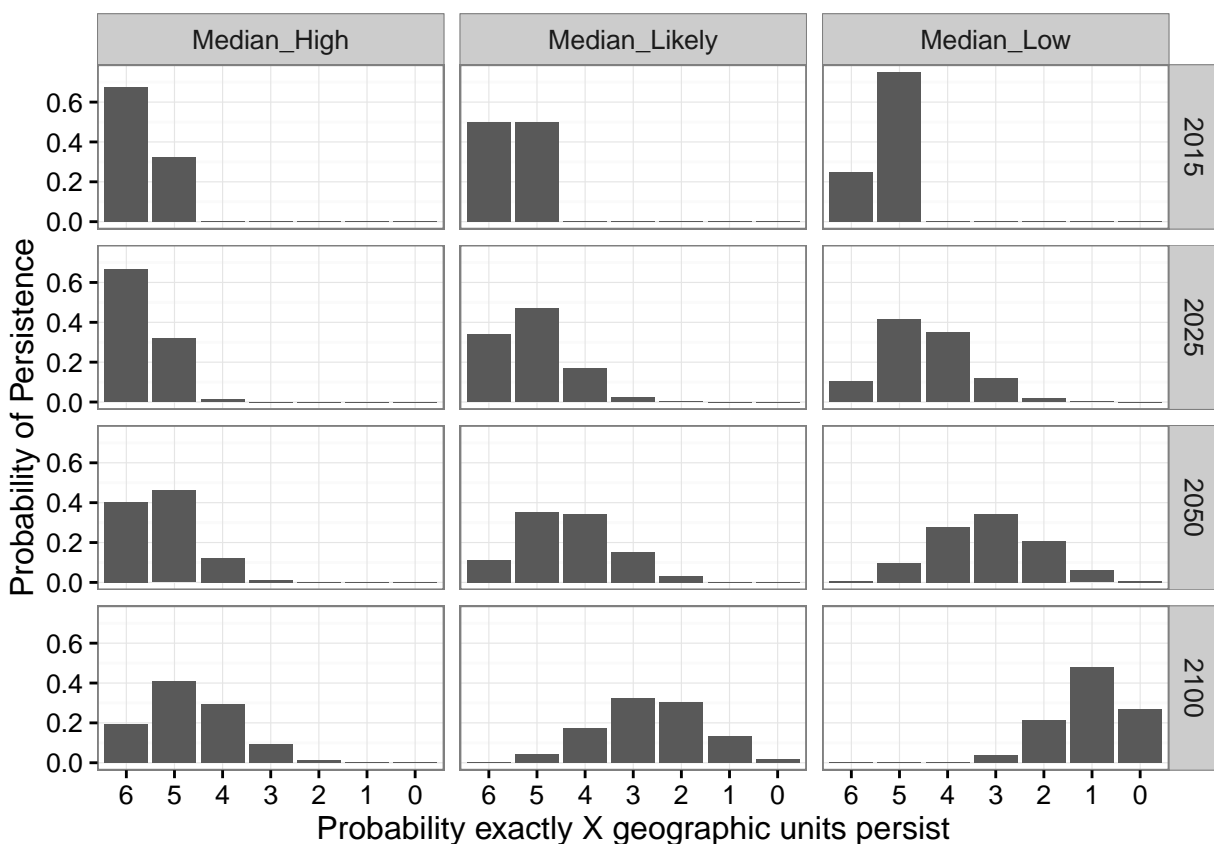
# Summaries for medians only
res.summary.data<-rbind(res.summary.data[res.summary.data$name=="Median_Low",],res.summary.data[res.sum

# Plot the probability of persistence across the six geographic units
res.plot.summary.p <- ggplot(data=res.summary.data[res.summary.data$Variable
=="Probability",],aes(y=Probability,x=Units)) + geom_bar(stat="identity") +
facet_grid(Time.Period~name) + scale_x_discrete(limits=c("6","5","4","3","2","1",
"0"),labels=c("6","5","4","3","2","1","0")) +
xlab("Probability exactly X geographic units persist") + ylab("Probability of Persistence") + theme_b

# select the data for the probability density figure, assign the x and y variables for plotting, make a

# Show the figure
res.plot.summary.p

```



```

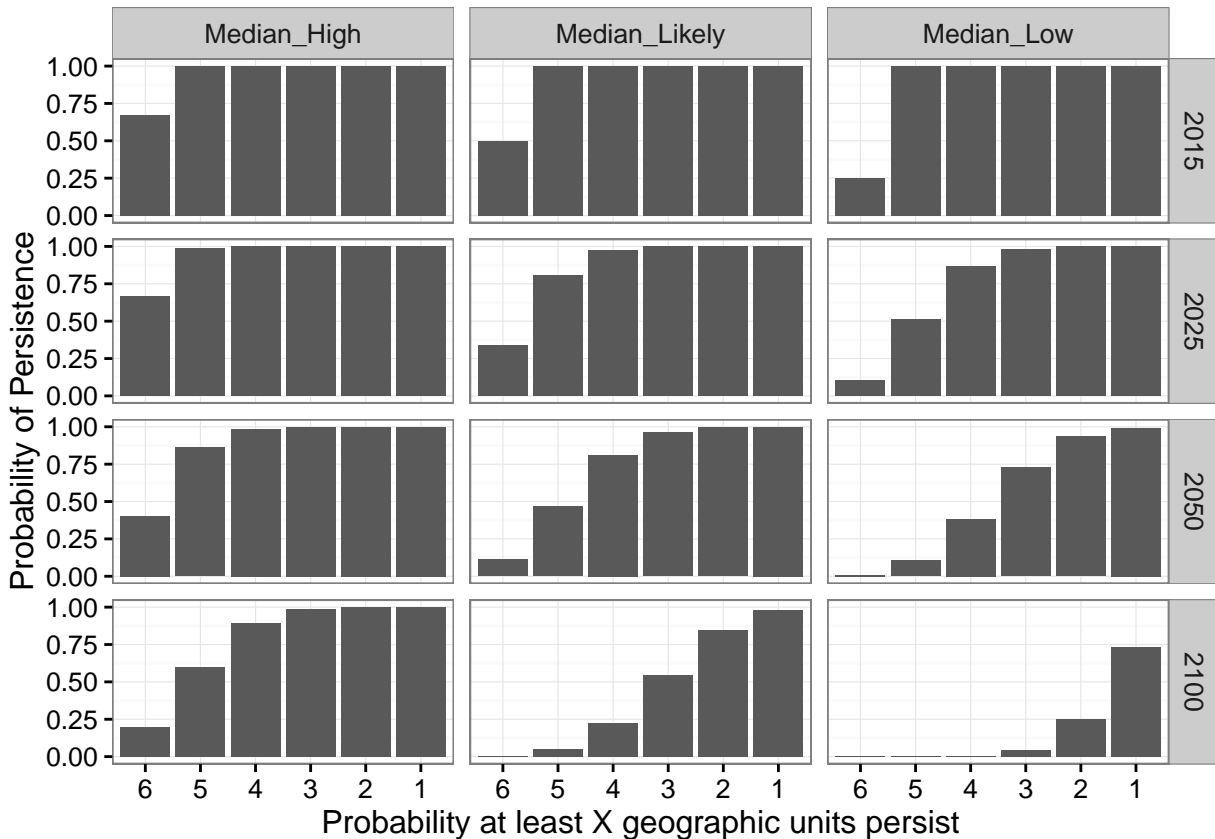
# Plot the cumulative probability of persistence across the six geographic units
res.plot.summary.cp <- ggplot(data=res.summary.data[res.summary.data$Variable=="
"Cumulative Probability",],aes(Probability,x=Units)) + geom_bar(stat="identity")+
facet_grid(Time.Period~name) + scale_x_discrete(limits=c("6","5","4","3","2",
"1"),labels=c("6","5","4","3","2","1")) +
xlab("Probability at least X geographic units persist") + ylab("Probability of Persistence") + theme_b

# As above, but select the cumulative probability data and limit the x axis to positive values

# Show the figure
res.plot.summary.cp

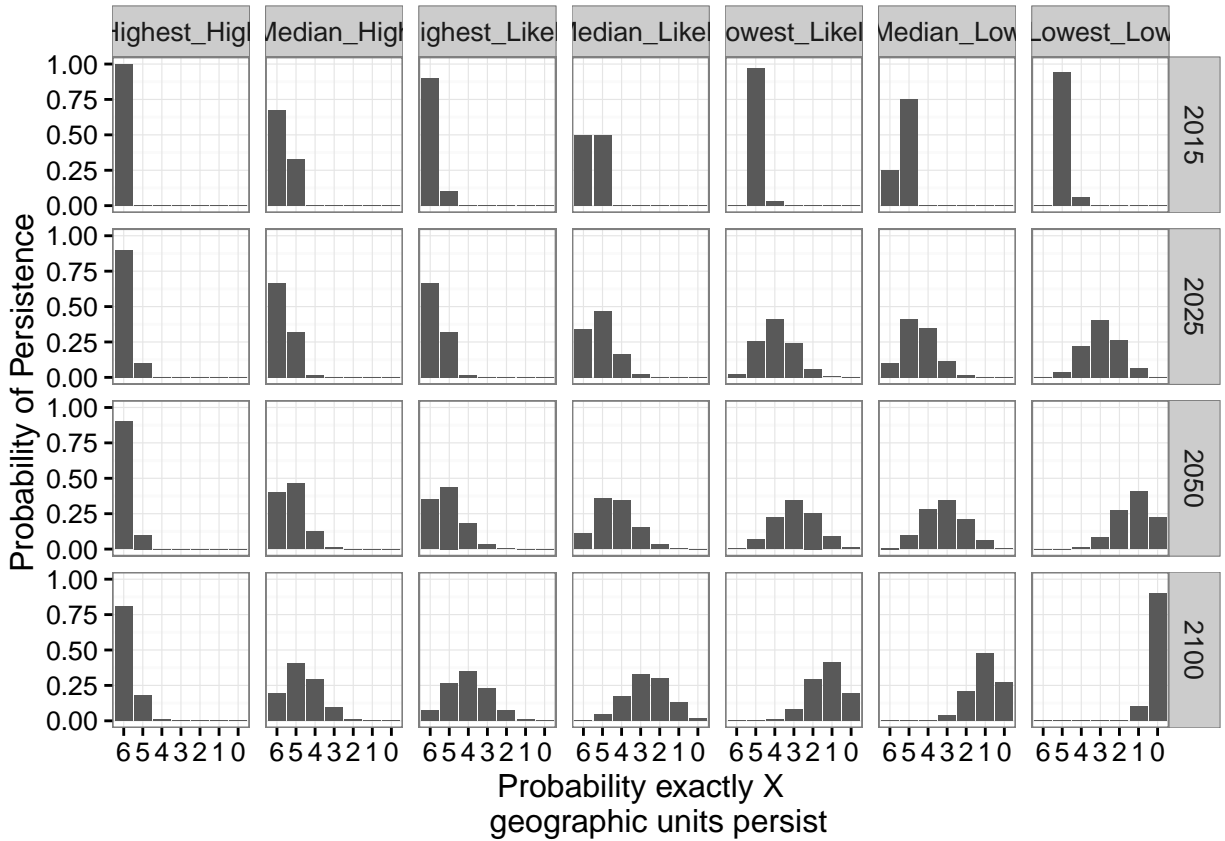
```

Warning: Removed 12 rows containing missing values (position_stack).



```
# Plots for all summaries. This produces the same figures as the sections above, but with all summaries.
# Plot the probability of persistence across the six geographic units
res.plot.summary.all.p <- ggplot(data=res.summary.data.all[res.summary.data.all$Variable
=="Probability",],aes(y=Probability,x=Units)) + geom_bar(stat="identity") +
  facet_grid(Time.Period~name) + scale_x_discrete(limits=c("6","5","4","3","2","1",
  "0"),labels=c("6","5","4","3","2","1","0")) + xlab("Probability exactly X
  geographic units persist")+ylab("Probability of Persistence") + theme_bw()
# select the data for the probability density figure, assign the x and y variables for plotting, make a

# Show the figure
res.plot.summary.all.p
```



```
# Plot the cumulative probability of persistence across the six geographic units
res.plot.summary.all.cp <- ggplot(data=res.summary.data.all[res.summary.data.all$Variable==
  "Cumulative Probability",],aes(Probability,x=Units)) + geom_bar(stat="identity")+
  facet_grid(Time.Period~name) + scale_x_discrete(limits=c("6","5","4","3","2",
  "1"),labels=c("6","5","4","3","2","1")) + xlab("Probability at least X geographic
  units persist") + ylab("Probability of Persistence") + theme_bw()
# As above, but select the cumulative probability data and limit the x axis to positive values

# Show the figure
res.plot.summary.all.cp
```

Warning: Removed 28 rows containing missing values (position_stack).

