Lynx EE figures

Jonathan Cummings

March 6, 2018

Introduction

This R markdown file contains code to process the results of the October 13-15 Lnyx 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
library(gridExtra) # package for manipulating the plotting window
library(plyr)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=80),tidy=TRUE)
```

Representation

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

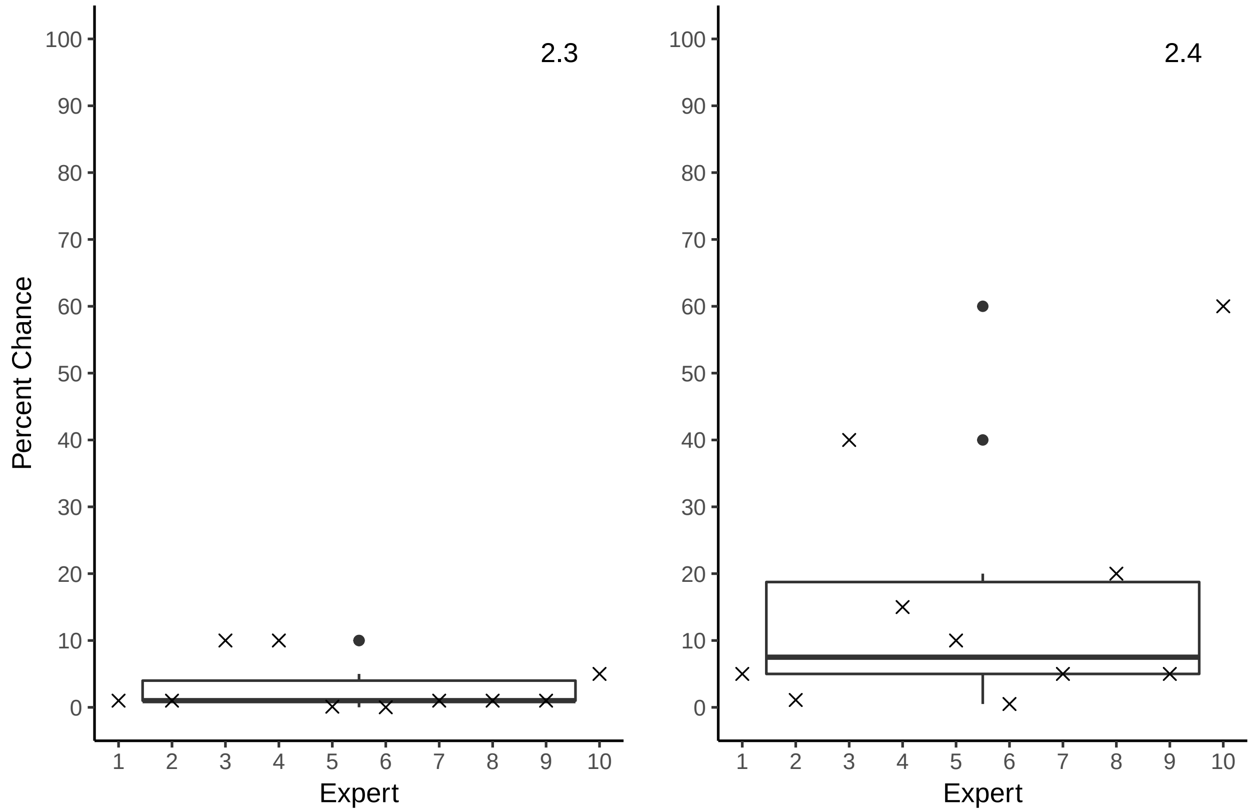
Redundancy

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

Likelihood of catastrophic events

This code produces figures 2 regarding the likelihood of catastrophic events.

```
# Responses to redundancy question 3, entered by hand from the response record
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() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.margin = grid::unit(c(0, 0, 0, 0), "mm")) + ylab("Percent Chance") + xlab("Expert") +
    scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, 10)) + annotate("text",
    x = 9.25, y = 98, label = "2.3") + coord_fixed(0.125)
# ggplot is the ploting function, geom_boxplot creates the boxplot and geom_point
# adds data points to the figure.  Theme controls the format of the plot, ylab
# and xlab and axis labels, scale_y_continuous controls the axis range, and
# annotate adds a figure number

# Responses to redundancy question 4, entered by hand from the response record
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("") + theme(panel.border = element_blank(), panel.grid.major = element_
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.margin = grid::unit(c(0, 0, 0, 0), "mm")) + xlab("Expert") + scale_y_continuous(limits = c(0,
    100), breaks = seq(0, 100, 10)) + annotate("text", x = 9.25, y = 98, label = "2.4") +
    coord_fixed(0.125)
# Same plot a for question 3, different data

# Arrange and print the plots as a single figure
grid.arrange(Q3, Q4, ncol = 2)
```

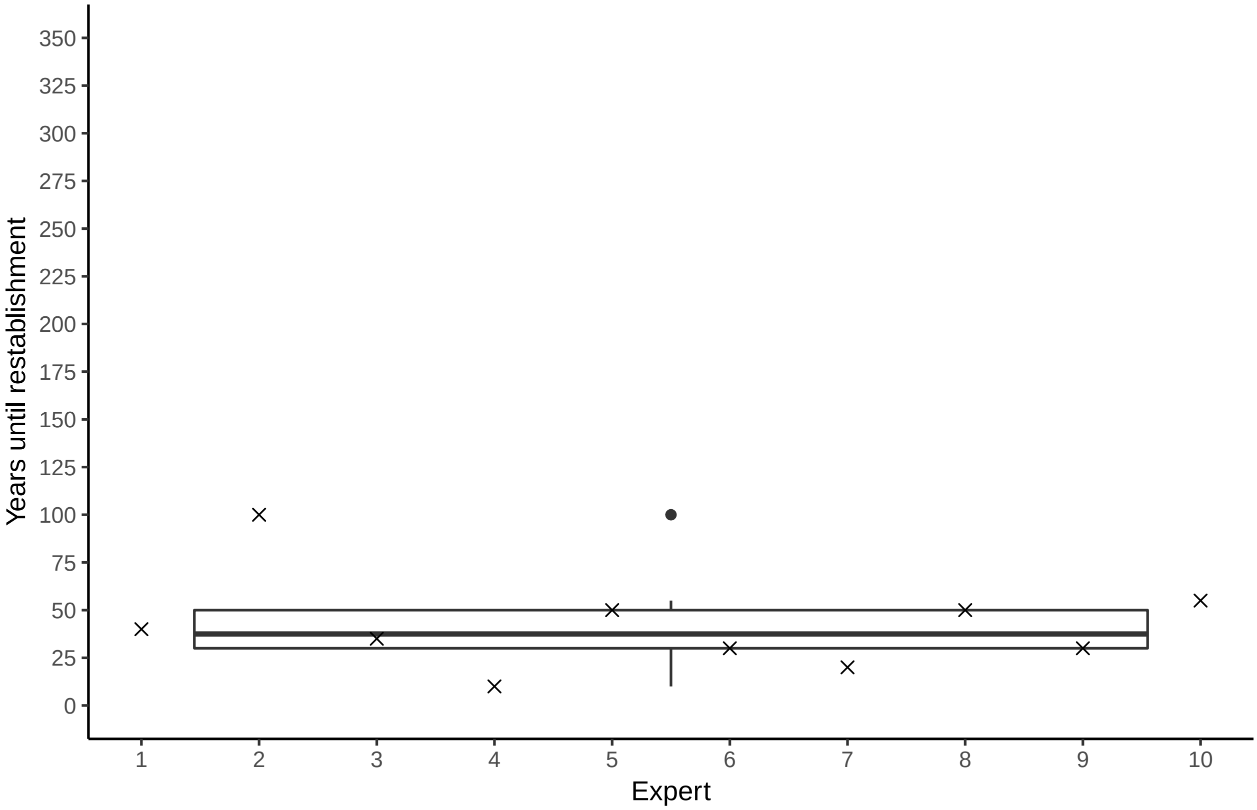


Median recovery time following a catastrophic event

This section of code produces figure 3 and a summary table used to supply text for the manuscript regarding the time required for a geographic unit to recover floowing a catastrophic event.

```
# 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("confidence", 10)), 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 redundancy question 5
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$most)/redundancy.df$confide
    95 + redundancy.df$most)
# Compute 95% confidence bound for longest time period
redundancy.df$longest.95 <- round(((redundancy.df$longest - redundancy.df$most)/redundancy.df$confidenc
    95 + redundancy.df$most)

# Manipulate data again to ease figure production
redundancy5 <- melt(redundancy.df, id.vars = "expert")
redundancy5$value[redundancy5$value < 0] <- 0
```

```r
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(21:30), ], aes(x = expert, y = value))
Q5.box <- plot.Q5.box + geom_boxplot() + geom_point(size = 2, shape = 4) + ylab("Years until restablishm
    xlab("Expert") + theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blan
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.margin = grid::unit(c(0, 0, 0, 0), "mm")) + scale_y_continuous(limits = c(0,
    350), breaks = seq(0, 350, 25)) + scale_x_continuous(breaks = seq(0, 10))
# Similar to the formating for Q3 & Q4

# Print the plot
Q5.box
```




```r
# Create a table from a series of columns summaries to display the responses to
# question 5
tableQ5 <- matrix(c(median(redundancy5[redundancy5$variable == "most likely", "value"]),
    median(redundancy5[redundancy5$variable == "shortest.95", "value"]), median(redundancy5[redundancy5$
        "longest.95", "value"], na.rm = T)), nrow = 1)
tableQ5 <- data.frame(tableQ5)
names(tableQ5) <- c("Median Years for Reestablishement", "Median Lower 95% CI", "Median Upper 95% CI")

# Display the table
kable(tableQ5)
```

Median Years for Reestablishement	Median Lower 95% CI	Median Upper 95% CI
37.5	0.5	158.5

Resiliency

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

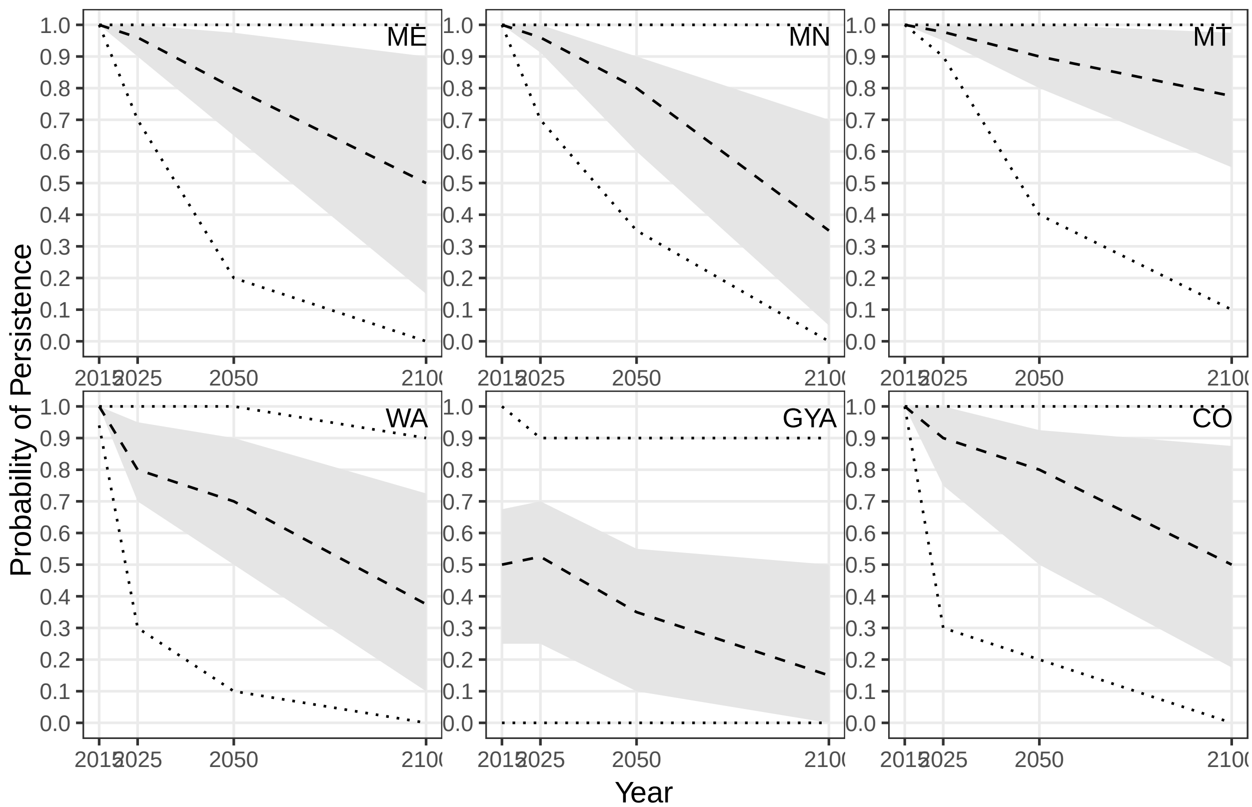
Resiliency by Geographic Unit

This section of code reads in the resiliency response file, reorganizes that data, and produces a summary figure (figure 4) of the responses for each geographic unit.

```
# read in resiliency data and organize it read in data file from workhop
resiliency.data <- read.csv("Resiliency Responses.csv")
# 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", ]

# Compute Median value by Geographic Unit, Probabiliy, and Time Period
r.medians <- ddply(resiliency.data, .(Geographic.Unit, Probability, Time.Period),
    summarise, median = median(value, na.rm = TRUE))
r.medians <- dcast(r.medians, Geographic.Unit + Time.Period ~ Probability)

# Filter by geographic unit
rm.ME <- r.medians[r.medians$Geographic.Unit == "Maine/NE", ]
rm.MN <- r.medians[r.medians$Geographic.Unit == "MN/Lakes States", ]
rm.MT <- r.medians[r.medians$Geographic.Unit == "Northwest MT/ NE Idaho", ]
rm.WA <- r.medians[r.medians$Geographic.Unit == "Washington", ]
rm.GYA <- r.medians[r.medians$Geographic.Unit == "Greater Yellowstone Area", ]
rm.CO <- r.medians[r.medians$Geographic.Unit == "Colorado", ]

# Create plot for each geographic unit

# Plot the probability of persistence responses at each time point by first
# providing the y and x values to plot, the scale of the x and y axis.
# Geom_ribbon creates the confidence interval and geom_point displays the
```

```
# responses from each expert. The stat_summary lines create lines on the plot for
# the bounds across experts, and the median of the high and low responses.

# Maine/NE (ME)
res.plot.ME <- ggplot(r.ME, aes(y = value, x = Time.Period)) + theme_bw() + scale_x_continuous(breaks =
    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)) + geom_ribbon(data = rm.ME, aes(y = Most.Likely,
    ymin = Lowest, ymax = Highest), fill = "grey90") + stat_summary(data = r.ME[r.ME$Probability ==
    "Highest", ], geom = "line", fun.y = "max", linetype = "dotted") + stat_summary(data = r.ME[r.ME$Pro
    "Lowest", ], geom = "line", fun.y = "min", linetype = "dotted") + stat_summary(data = r.ME[r.ME$Prob
    "Most.Likely", ], geom = "line", fun.y = "median", linetype = "dashed") + annotate("text",
    x = 2095, y = 0.965, label = "ME") + theme(axis.title.x = element_blank(), axis.title.y = element_bl
    panel.grid.minor = element_blank(), plot.margin = grid::unit(c(0, 0, 0.75, 0),
        "mm"))

# MN/Lakes States(MN)
res.plot.MN <- ggplot(r.MN, aes(y = value, x = Time.Period)) + theme_bw() + scale_x_continuous(breaks =
    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)) + geom_ribbon(data = rm.MN, aes(y = Most.Likely,
    ymin = Lowest, ymax = Highest), fill = "grey90") + stat_summary(data = r.MN[r.MN$Probability ==
    "Highest", ], geom = "line", fun.y = "max", linetype = "dotted") + stat_summary(data = r.MN[r.MN$Pro
    "Lowest", ], geom = "line", fun.y = "min", linetype = "dotted") + stat_summary(data = r.MN[r.MN$Prob
    "Most.Likely", ], geom = "line", fun.y = "median", linetype = "dashed") + annotate("text",
    x = 2095, y = 0.965, label = "MN") + theme(legend.position = "none", axis.title.x = element_blank()
    axis.title.y = element_blank(), panel.grid.minor = element_blank(), plot.margin = grid::unit(c(0,
        0, 0.75, 0), "mm"))

# Northwest MT/ NE Idaho
res.plot.MT <- ggplot(r.MT, aes(y = value, x = Time.Period)) + theme_bw() + scale_x_continuous(breaks =
    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)) + geom_ribbon(data = rm.MT, aes(y = Most.Likely,
    ymin = Lowest, ymax = Highest), fill = "grey90") + stat_summary(data = r.MT[r.MT$Probability ==
    "Highest", ], geom = "line", fun.y = "max", linetype = "dotted") + stat_summary(data = r.MT[r.MT$Pro
    "Lowest", ], geom = "line", fun.y = "min", linetype = "dotted") + stat_summary(data = r.MT[r.MT$Prob
    "Most.Likely", ], geom = "line", fun.y = "median", linetype = "dashed") + annotate("text",
    x = 2095, y = 0.965, label = "MT") + theme(legend.position = "none", axis.title.x = element_blank()
    axis.title.y = element_blank(), panel.grid.minor = element_blank(), plot.margin = grid::unit(c(0,
        0, 0.75, 0), "mm"))

# Washington
res.plot.WA <- ggplot(r.WA, aes(y = value, x = Time.Period)) + theme_bw() + scale_x_continuous(breaks =
    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)) + geom_ribbon(data = rm.WA, aes(y = Most.Likely,
    ymin = Lowest, ymax = Highest), fill = "grey90") + stat_summary(data = r.WA[r.WA$Probability ==
    "Highest", ], geom = "line", fun.y = "max", linetype = "dotted") + stat_summary(data = r.WA[r.WA$Pro
    "Lowest", ], geom = "line", fun.y = "min", linetype = "dotted") + stat_summary(data = r.WA[r.WA$Prob
    "Most.Likely", ], geom = "line", fun.y = "median", linetype = "dashed") + annotate("text",
    x = 2095, y = 0.965, label = "WA") + theme(legend.position = "none", axis.title.x = element_blank()
    axis.title.y = element_blank(), panel.grid.minor = element_blank(), plot.margin = grid::unit(c(0,
        0, 0.75, 0), "mm"))

# Greater Yellowstone Area
res.plot.GYA <- ggplot(r.GYA, aes(y = value, x = Time.Period)) + theme_bw() + scale_x_continuous(breaks
```

```r
    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)) + geom_ribbon(data = rm.GYA, aes(y = Most.Likely,
    ymin = Lowest, ymax = Highest), fill = "grey90") + stat_summary(data = r.GYA[r.GYA$Probability ==
    "Highest", ], geom = "line", fun.y = "max", linetype = "dotted") + stat_summary(data = r.GYA[r.GYA$P
    "Lowest", ], geom = "line", fun.y = "min", linetype = "dotted") + stat_summary(data = r.GYA[r.GYA$Pr
    "Most.Likely", ], geom = "line", fun.y = "median", linetype = "dashed") + annotate("text",
    x = 2095, y = 0.965, label = "GYA") + theme(legend.position = "none", axis.title.x = element_blank(
    axis.title.y = element_blank(), panel.grid.minor = element_blank(), plot.margin = grid::unit(c(0,
        0, 0.75, 0), "mm"))

# Colorado
res.plot.CO <- ggplot(r.CO, aes(y = value, x = Time.Period)) + theme_bw() + scale_x_continuous(breaks =
    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)) + geom_ribbon(data = rm.CO, aes(y = Most.Likely,
    ymin = Lowest, ymax = Highest), fill = "grey90") + stat_summary(data = r.CO[r.CO$Probability ==
    "Highest", ], geom = "line", fun.y = "max", linetype = "dotted") + stat_summary(data = r.CO[r.CO$Pro
    "Lowest", ], geom = "line", fun.y = "min", linetype = "dotted") + stat_summary(data = r.CO[r.CO$Prob
    "Most.Likely", ], geom = "line", fun.y = "median", linetype = "dashed") + annotate("text",
    x = 2095, y = 0.965, label = "CO") + theme(legend.position = "none", axis.title.x = element_blank()
    axis.title.y = element_blank(), panel.grid.minor = element_blank(), plot.margin = grid::unit(c(0,
        0, 0.75, 0), "mm"))

# Plot
grid.arrange(res.plot.ME, res.plot.MN, res.plot.MT, res.plot.WA, res.plot.GYA, res.plot.CO,
    ncol = 3, left = "Probability of Persistence", bottom = "Year")
```



Resiliency across Geographic Units

This section uses the binonomial probability of persistence in each unit and combines them to determine the resulting probability of persistence across units to produce figures 5 and 6.

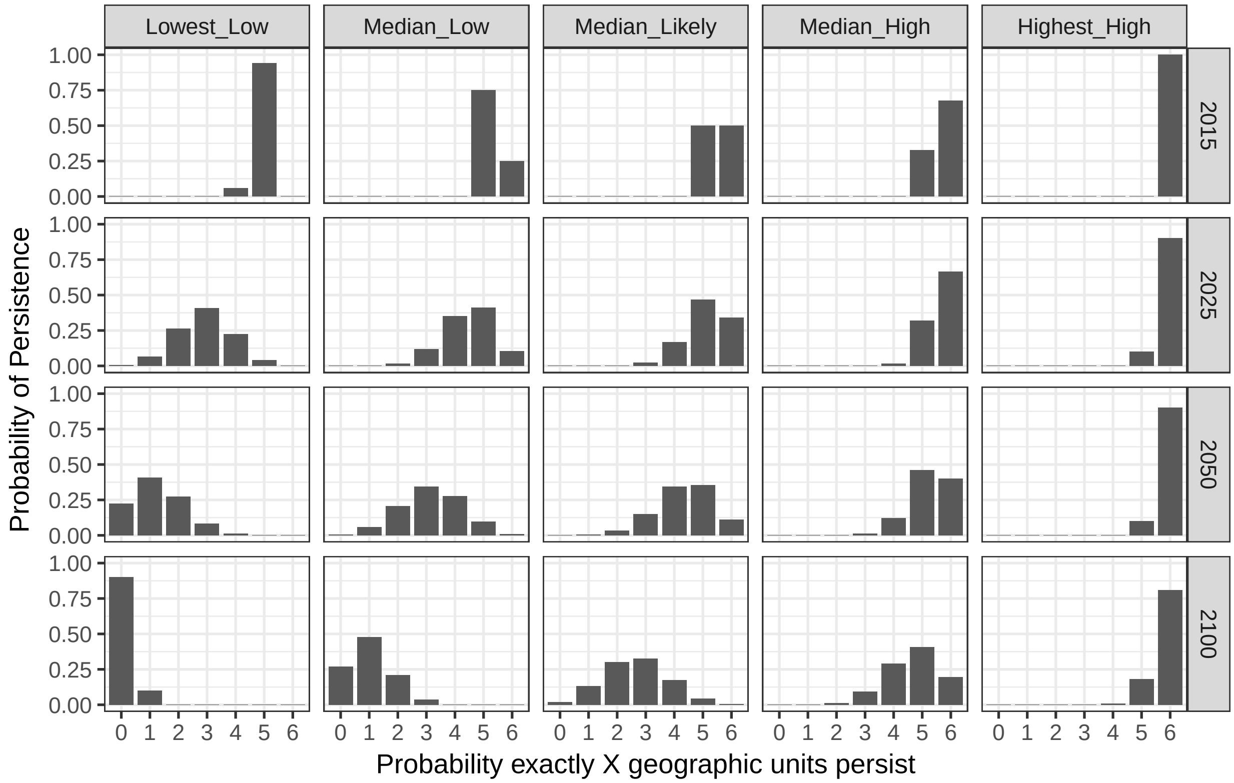
Some terminology used in the plot:

"Highest_High" — 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" — the probability of persistence generated by selecting the median probability of persistence across experts from the highest probability response in each geographic unit. "Median_Likely" — the probability of persistence generated by selecting the median probability of persistence across experts from the most likely probability response in each geographic unit. "Median_Low" — 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" — the probability of persistence generated by selecting the lowest probability of persistence across experts from the lowest probability response in each geographic unit.

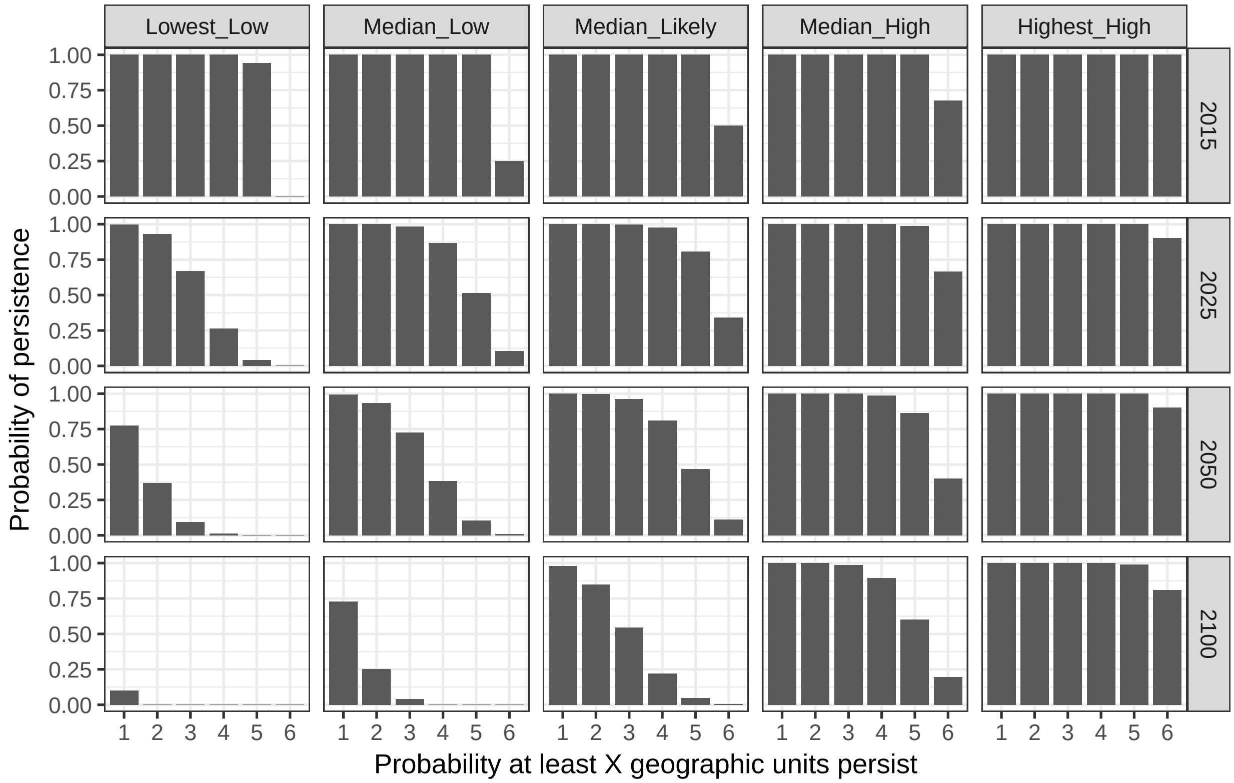
```
# Summaries of median probabilities create empty vector to bind results to
r.summary <- 0

# loop through data to compute median and find max and min
for (i in unique(resiliency.data$Geographic.Unit)) {
    for (j in unique(resiliency.data$Time.Period)) {
        for (k in unique(resiliency.data$Probability)) {
            r.summary <- rbind(r.summary, c(i, j, k, "median", median(resiliency.data[resiliency.data$Ge
                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"))

# 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"), ]
r.summary.big <- r.summary.big[!(r.summary.big$Probability == "Most.Likely" & r.summary.big$variable ==
    "min"), ]
r.summary.big <- r.summary.big[!(r.summary.big$Probability == "Most.Likely" & r.summary.big$variable ==
    "max"), ]
# order the data by time period, variable, and probability
```

```r
r.summary.big <- r.summary.big[order(r.summary.big$Time.Period, r.summary.big$variable,
    r.summary.big$Probability), ]
# number the rows
row.names(r.summary.big) <- 1:nrow(r.summary.big)


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))
# convert the list structure to character
r.summary.big[] <- lapply(r.summary.big, as.character)
# Name the results in the first element of the list and assign the results of the
# convolve function to the for (i in n[seq(1, length(n), 6)]){
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 persisting to 1
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])
```

```r
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)
# Change the names, create a data.frame for each figure, Convert the unit column
# to a factor, and order the factors for the figure
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$name == "Most.Likely.median", "name"] <- "Median_Likely"
res.summary.data[res.summary.data$name == "Lowest.median", "name"] <- "Median_Low"
res.summary.data[res.summary.data$name == "Lowest.min", "name"] <- "Lowest_Low"

# Separate the data by probablity of a given number of units persisting and the
# cumulative probability that at least a given number persist
res.summary.p <- res.summary.data[res.summary.data$Variable == "Probability", ]
res.summary.cp <- res.summary.data[res.summary.data$Variable == "Cumulative Probability",
    ]

# Create factors and sort data for plotting
res.summary.p$Units <- factor(res.summary.p$Units, levels = c("0", "1", "2", "3",
    "4", "5", "6"))
res.summary.p$name <- factor(res.summary.p$name, levels = c("Lowest_Low", "Median_Low",
    "Median_Likely", "Median_High", "Highest_High"))
res.summary.p <- res.summary.p[order(res.summary.p[, 4], res.summary.p[, 1], res.summary.p[,
    2]), ]

# Create plot
res.plot.summary.p <- ggplot(data = res.summary.p, aes(y = Probability, x = Units)) +
    geom_bar(stat = "identity", width = 0.85) + facet_grid(Time.Period ~ name) +
    scale_x_discrete(limits = c("0", "1", "2", "3", "4", "5", "6"), labels = c("0",
        "1", "2", "3", "4", "5", "6")) + 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; select the data for the probability density figure, assign the x and
# y variables for ploting, make a bar graph, add the faceting, define the x axis,
# label the axes, and set the theme

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




```r
# Repeate for cumulative probablity plot
res.summary.cp$Units <- factor(res.summary.cp$Units, levels = c("0", "1", "2", "3",
    "4", "5", "6"))
res.summary.cp$name <- factor(res.summary.cp$name, levels = c("Lowest_Low", "Median_Low",
    "Median_Likely", "Median_High", "Highest_High"))
res.summary.cp <- res.summary.cp[order(res.summary.cp[, 4], res.summary.cp[, 1],
    res.summary.cp[, 2]), ]

res.plot.summary.cp <- ggplot(data = res.summary.cp, aes(y = Probability, x = Units)) +
    geom_bar(stat = "identity", width = 0.85) + facet_grid(Time.Period ~ name) +
    scale_x_discrete(limits = c("1", "2", "3", "4", "5", "6"), labels = c("1", "2",
        "3", "4", "5", "6")) + xlab("Probability at least X geographic units persist") +
    ylab("Probability of persistence") + theme_bw()

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


Lowest_Low
Median_Low
Median_Likely
Median_High
Highest_High
2015
2025
2050
2100
1.00
0.75
0.50
0.25
0.00
1 2 3 4 5 6
Probability of persistence
Probability at least X geographic units persist