

The plight of a plover: viability of an important Snowy Plover population with flexible brood care in Mexico

Luke J. Eberhart-Phillips, Medardo Cruz-López, Guillermo Fernández, Rene Beamonte-Barrientos, Tamás Székely, Martín Alejandro Serrano-Meneses, and Clemens Küpper

In this document we provide all the necessary code for reproducing the analyses presented in our paper. To access the dataset and Rmarkdown file, please download this GitHub repository. Simply follow the link and click on *Download ZIP* on the right-hand side of the page. An explanation of the files in the repository can be found in the Readme file. Please don't hesitate to contact Luke at `luke.eberhart[at]gmail.com` if you have any questions.

Prerequisites:

- For running the complete code you need a `files` subfolder containing the raw data downloaded from `Data_files` folder provided in the GitHub repository.
- The following packages are needed for analysis and can be easily installed from CRAN by uncommenting the `install.packages` functions:

```
# install.packages(RMark)
# install.packages(stringr)
# install.packages(ggplot2)
# install.packages(extrafont)
# install.packages(dplyr)
# install.packages(reshape2)
# install.packages(popbio)

library(RMark)
library(stringr)
library(ggplot2)
library(extrafont)
library(dplyr)
library(reshape2)
library(popbio)
```

Work flow of analysis presented and datasets required

- 1) Nest survival analysis (dataset: `nest_data.txt`)
- 2) Chick survival analysis (dataset: `chick_data.txt`)
- 3) Fledgling and adult survival analysis (dataset: `fledgling_adult_data.txt`)
- 4) Fecundity analysis (dataset: `fecundity_data.txt`)
- 5) Matrix model validation
- 6) Matrix model forecasting
- 7) Sensitivity analysis

Explanation of datasets:

- **nest_fate.txt**: each row is a single nest. Definitions of variables:
FirstFound – Day of the season when nest was found
LastPresent – Last day of the season the nest was seen alive
LastChecked – Last day of the season the nest was checked
Fate – 1 = was seen alive after the 25th day of incubation, 0 = failed before 25th day of incubation
Freq – Frequency of nests with these data
NestAge – Age of a nest on the first occasion
Year – Year that nest was monitored (between 2006 and 2012)
std_time – Standardized laying date (earlier than average nests are negative, late nests are positive).
 Note: maximum duration of breeding season is 85 days.
- **chick_data.txt**: each row is a single individual. Definitions of variables:
ch – capture history of a given chick over the brooding period. 1 = seen, 0 = not seen, . = no survey effort given on that day (no data)
Year – Year that chick was monitored (between 2006 and 2012)
Day_of_Season – Day of season that chick hatched
- **fledgling_adult_data.txt**: each row is a single individual. Definitions of variables:
ch – capture history of a given individual each year between 2006 and 2012. 1 = seen, 0 = not seen
age – Adult or juvenile [NOTE: this describes the stage that the individual was initially ringed]
- **fecundity_data.txt**: each row is a single male. Definitions of variables:
Year – Year of data collection (between 2006 and 2012)
Bird_ID – Unique number assigned to each male individual
Eggs – number of eggs tended by a given male in a given year

Nest survival model selection and analysis

```
# import the nest history file
nest_fate <-
  read.table("Data_files/nest_data.txt",
             header=T)

# define year as a factor
nest_fate$Year <- as.factor(nest_fate$Year)

# define the number of occasions
occ <- max(nest_fate$LastChecked)

# create processed RMARK data format as NestSurvival with Year as group
nest_fate.processed <- RMark::process.data(nest_fate, model = "Nest",
                                           nocc = occ,
                                           groups = c("Year"))

# create the design data
nest_fate.dd1 <- RMark::make.design.data(nest_fate.processed)

# add a new variable to the design data that is the quadratic transformation of
# time
time <- c(0:(occ-1))
Quadratic <- time^2
```

```

quad_time <- data.frame(time, Quadratic)
quad_time$time <- c(1:occ)
nest_fate.ddl$S <-
  RMark::merge_design.covariates(nest_fate.ddl$S, quad_time,
                                bygroup = FALSE, bytime = TRUE)

# Create model list as function
Ceuta_nest_survival <- function()
{

  # Specify models to test
  # constant daily survival rate (DSR)
  S.dot <-
    list(formula = ~1)

  # DSR varies by year
  S.year <-
    list(formula = ~Year)

  # DSR varies with Time (i.e., linearly across the season)
  S.Time <-
    list(formula = ~Time)

  # DSR varies quadratically with Time
  S.Quadratic_Time <-
    list(formula = ~Quadratic)

  # DSR varies with standardized time
  S.Std_time <-
    list(formula = ~std_time)

  # DSR varies with nest age
  S.Age <-
    list(formula = ~NestAge)

  # DSR varies year and time
  S.yearTime <-
    list(formula = ~Year + Time)

  # DSR varies year and time
  S.yearxTime <-
    list(formula = ~Year * Time)

  # DSR varies with year and nest age
  S.yearAge <-
    list(formula = ~Year + NestAge)

  # DSR varies with year and nest age
  S.yearxAge <-
    list(formula = ~Year * NestAge)

  # DSR varies with time and nest age
  S.TimeAge <-

```

```

    list(formula = ~Time + NestAge)

# DSR varies with time and nest age
S.TimexAge <-
  list(formula = ~Time * NestAge)

# DSR varies with year, habitat, time and nest age
S.global <-
  list(formula = ~Year + Time + NestAge)

# DSR varies with year, habitat, time and nest age
S.globalx <-
  list(formula = ~Year * Time * NestAge)

# specify to run as a nest survival model in program MARK
cml <- RMark::create.model.list("Nest")

# run model list in MARK. Supress generation of MARK files.
model.list <- RMark::mark.wrapper(cml,
                                   data = nest_fate.processed, ddl = nest_fate.ddl,
                                   threads = 4, brief = TRUE, delete = TRUE)

# store completed model list
return(model.list)
}

# run the model list
Ceuta_nest_survival_run <- Ceuta_nest_survival()

# examine AIC table
Ceuta_nest_survival_run

# Extract estimates of survival from top model
Nest_survival_reals <- Ceuta_nest_survival_run[[4]]$results$real

# wrangle dataframe to get annual estimates (grouped by "Year")
Groups <- data.frame(
  str_split_fixed(rownames(Nest_survival_reals), " ", n = 4))
Nest_survival_reals <- cbind(Groups, Nest_survival_reals)
Nest_survival_reals$Year <- unlist(substr(Nest_survival_reals$X2, 2, 5))

# summarize the seasonal estimates by averaging within Year
Yearly_nest_survival_avgs <- Nest_survival_reals %>%
  group_by(Year) %>%
  summarise_each(funs(mean))

# calculate the apparent nest survival by raising daily nest survival to ^25
Yearly_nest_survival_avgs$Year_apparent <-
  Yearly_nest_survival_avgs$estimate^25
Yearly_nest_survival_avgs$Year_apparent_lcl <-
  Yearly_nest_survival_avgs$lcl^25
Yearly_nest_survival_avgs$Year_apparent_ucl <-
  Yearly_nest_survival_avgs$ucl^25

```

```

# clean dataframe
Yearly_nest_survival_avgs <- Yearly_nest_survival_avgs[, -c(2:11)]
Yearly_nest_survival_avgs$Age <- "Egg"

```

Chick survival model selection and analysis

```

# Import chick capture history file
Ceuta_chicks <-
  read.table("Data_files/chick_data.txt",
            header = TRUE, colClasses = c("character", "factor", "numeric"))

# Create processed RMARK data format as CJS with "Year" as group
Ceuta_chicks.proc <- process.data(Ceuta_chicks, model = "CJS",
                                groups = c("Year"))

# Create the design data
Ceuta_chicks.ddl <- make.design.data(Ceuta_chicks.proc)

# create a quadratic time variable
time <- c(0:(Ceuta_chicks.proc$nocc[1]-1))
Quadratic <- time^2
Cubic <- time^3
quad_time <- data.frame(time, Quadratic, Cubic)
Ceuta_chicks.ddl$p <- merge_design.covariates(Ceuta_chicks.ddl$Phi,
                                             quad_time, bygroup=FALSE, bytime=TRUE)
Ceuta_chicks.ddl$Phi <- merge_design.covariates(Ceuta_chicks.ddl$Phi,
                                             quad_time, bygroup=FALSE, bytime=TRUE)

# First assess the best model structure of encounter probability (p) while
# keeping survival (Phi) constant
Ceuta_chicks_p=function()
{
  # Constant Phi
  Phi.dot <-
    list(formula = ~1)

  # Constant p
  p.dot <-
    list(formula = ~1)

  # p varies linearly with chick age
  p.Time <-
    list(formula = ~Time)

  # p varies quadratically with chick age
  p.Quadratic <-
    list(formula = ~Quadratic)

  # p varies across year
  p.year <-
    list(formula = ~Year)
}

```

```

# p varies across the hatch date
p.hatchdate <-
  list(formula = ~Day_of_Season)

# interaction between year and age
p.year.x.Time <-
  list(formula = ~Year * Time)

# interaction between year and quadratic age
p.year.x.Quadratic <-
  list(formula = ~Year * Quadratic)

# interaction between year and hatch date
p.year.x.hatchdate <-
  list(formula = ~Year * Day_of_Season)

# additive effect between year and age
p.year.Time <-
  list(formula = ~Year + Time)

# additive effect between year and quadratic age
p.year.Quadratic <-
  list(formula = ~Year + Quadratic)

# additive effect between year and hatch date
p.year.hatchdate <-
  list(formula = ~Year + Day_of_Season)

# create model list for all a priori models above that begin with Phi. or p.
cml <- create.model.list("CJS")

# runs model list in program MARK
model.list <- mark.wrapper(model.list = cml, data = Ceuta_chicks.proc,
                           ddl = Ceuta_chicks.ddl,
                           threads = 4, brief = TRUE, delete = TRUE)

# store completed model list
return(model.list)
}

# Run model selection to determine best structure of p
Ceuta_chicks_p_run <- Ceuta_chicks_p()

# inspect p AIC table to find top model
Ceuta_chicks_p_run # best structure: p(~Year * Day_of_Season)

# Find best structure of Phi while using p(~Year * Day_of_Season)
Ceuta_chicks_Phi <- function()
{
  # Constant Phi
  Phi.dot <-
    list(formula = ~1)

```

```

# Phi varies with chick age
Phi.Time <-
  list(formula = ~Time)

# Phi varies quadratically with chick age
Phi.Quadratic <-
  list(formula = ~Quadratic)

# Phi varies by year
Phi.year <-
  list(formula = ~Year)

# Phi varies by hatch date
Phi.hatchdate <-
  list(formula = ~Day_of_Season)

# interaction between hatch date and age
Phi.hatchdate.x.Time <-
  list(formula = ~Day_of_Season * Time)

# interaction between hatch date and quadratic age
Phi.hatchdate.x.Quadratic <-
  list(formula = ~Day_of_Season * Quadratic)

# interaction between year and hatch date
Phi.year.x.hatchdate <-
  list(formula = ~Year * Day_of_Season)

# additive effect of hatch date and age
Phi.hatchdate.Time <-
  list(formula = ~Day_of_Season + Time)

# additive effect of hatch date and quadratic age
Phi.hatchdate.Quadratic <-
  list(formula = ~Day_of_Season + Quadratic)

# best structure of p determined from previous analysis
p.year.x.hatchdate <-
  list(formula = ~Year * Day_of_Season)

# create model list for all a priori models above that begin with Phi. or p.
cml <- create.model.list("CJS")

# runs model list in program MARK
model.list <- mark.wrapper(model.list = cml, data = Ceuta_chicks.proc,
                           ddl = Ceuta_chicks.ddl,
                           threads = 4, brief = TRUE, delete = TRUE)

# store completed model list
return(model.list)
}

# Run model selection to determine best structure of Phi

```

```

Ceuta_chicks_Phi_run <- Ceuta_chicks_Phi()

# inspect Phi AIC table to find top model
Ceuta_chicks_Phi_run # best structure: Phi(~Year * Day_of_Season)

# Extract estimates of survival from top model
Chick_survival_reals <- Ceuta_chicks_Phi_run[[10]]$results$real

# wrangle dataframe to get annual estimates (grouped by "Year")
Groups <- data.frame(str_split_fixed(rownames(Chick_survival_reals), " ", n = 5))
Chick_survival_reals <- cbind(Groups, Chick_survival_reals)
Chick_survival_reals$Year <- unlist(substr(Chick_survival_reals$X2, 2, 5))
Chick_survival_reals <- Chick_survival_reals[which(Chick_survival_reals$X1 == "Phi"),]

# calculate the apparent fledging success by raising daily chick survival to ~25
Chick_survival_reals$Year_apparent <- Chick_survival_reals$estimate^25
Chick_survival_reals$Year_apparent_lcl <- Chick_survival_reals$lcl^25
Chick_survival_reals$Year_apparent_ucl <- Chick_survival_reals$ucl^25

# clean dataframe
Chick_survival_reals <- Chick_survival_reals[, -c(1:11)]
Chick_survival_reals$Age <- "Chick"
row.names(Chick_survival_reals) <- NULL

```

Fledgling and adult survival model selection and analysis

```

# Import fledgling and adult capture history file
Ceuta_snpl_FA <- read.table("Data_files/fledgling_adult_data.txt",
                           header = TRUE, colClasses = c("character", "factor"))

# Create processed RMARK data format as CJS with 1 group (age initially ringed)
# and starting at year 2006
Ceuta_snpl_FA.proc <- process.data(Ceuta_snpl_FA, model = "CJS",
                                   groups = c("age"), begin.time = 2006,
                                   age.var = 1, initial.age = c(1, 0))

# Create the design data
Ceuta_snpl_FA.ddl <- make.design.data(Ceuta_snpl_FA.proc)

# adds firstyear/adult age field to design data in column "age"
Ceuta_snpl_FA.ddl <- add.design.data(Ceuta_snpl_FA.proc, Ceuta_snpl_FA.ddl,
                                     "Phi", "age", bins = c(0, 1, 7),
                                     right = FALSE, name = "age",
                                     replace = TRUE)

# create a dummy field called marked.as.adult which is 0 for the group initially
# ringed as juvenile and 1 for the group marked as adults.
Ceuta_snpl_FA.ddl$Phi$marked.as.adult <- 0
Ceuta_snpl_FA.ddl$Phi$marked.as.adult[Ceuta_snpl_FA.ddl$Phi$initial.age.class == "A"] <- 1
Ceuta_snpl_FA.ddl$p$marked.as.adult <- 0
Ceuta_snpl_FA.ddl$p$marked.as.adult[Ceuta_snpl_FA.ddl$p$initial.age.class == "A"] <- 1

```



```

# First determine the best structure of p
Ceuta_FA_p <- function()
{
  # constant Phi
  Phi.dot <-
    list(formula = ~1)

  # p varies by stage
  p.age <-
    list(formula = ~age) #p(age(.))

  # constant p
  p.dot <-
    list(formula = ~1) #p(.)

  # p varies annually
  p.time <-
    list(formula = ~time) #p(t)

  # interaction between annual variation and stage
  p.agetime <-
    list(formula = ~age * time)

  # additive effects of stage and annual variation
  p.age_time <-
    list(formula = ~age + time)

  # creates model list for all a priori models above that begin with Phi. or p.
  cml <- create.model.list("CJS")

  # runs model list in program MARK
  model.list <- mark.wrapper(cml, data = Ceuta_snpl_FA.proc, ddl = Ceuta_snpl_FA.ddl,
    threads = 4, brief = TRUE, delete = TRUE)

  # store completed model list
  return(model.list)
}

# Run models to determine best structure of p
Ceuta_FA_p_run <- Ceuta_FA_p()

# Inspect model output table ranked by AIC
Ceuta_FA_p_run # Best strucure was: p(~age * time)

# determine the nest structure of Phi while using p(~age * time)
Ceuta_FA_Phi <- function()
{
  # additive effects of stage and annual variaiton
  Phi.age_time <-
    list(formula = ~age + time) #Phi(age(t))

  # interaction between stage and year
  Phi.agetime <-

```

```

    list(formula = ~age * time)

# Phi varies by stage
Phi.age <-
  list(formula = ~age) #Phi(age(.))

# constant Phi
Phi.dot <-
  list(formula = ~1) #p = Phi(.)

# Phi varies by year
Phi.time <-
  list(formula = ~time) #Phi(t)

# Phi varies by stage
p.age <-
  list(formula = ~age * time) #p(age(.))

# creates model list for all a priori models above that begin with Phi. or p.
cml <- create.model.list("CJS")

# runs model list in MARK
model.list <- mark.wrapper(cml, data = Ceuta_snpl_FA.proc, ddl = Ceuta_snpl_FA.ddl,
  threads = 4, brief = TRUE, delete = TRUE)

# store completed model list
return(model.list)
}

# Run models
Ceuta_FA_Phi_run <- Ceuta_FA_Phi()

# Inspect model output table ranked by AIC
Ceuta_FA_Phi_run

# Extract estimates of survival from top model
Year_Age_reals <- Ceuta_FA_Phi_run[[3]]$results$real

# wrangle dataframe to get annual estimates (grouped by "Year")
Groups <- data.frame(str_split_fixed(rownames(Year_Age_reals), " ", n = 5))
Year_Age_reals <- cbind(Groups, Year_Age_reals)
Year_Age_reals$Year <- unlist(substr(Year_Age_reals$X5, 2, 5))
Year_Age_reals <- Year_Age_reals[which(Year_Age_reals$X1 == "Phi"),]
Year_Age_reals$Age <- unlist(str_extract_all(Year_Age_reals$X2, "[AJ]"))

# clean dataframe
Year_Juv_reals <- Year_Age_reals[which(Year_Age_reals$Age == "J"),]
Year_Juv_reals <- Year_Juv_reals[,c(12, 6, 8, 9)]
rownames(Year_Juv_reals) <- NULL
Year_Juv_reals$Age <- "Fledgling"
colnames(Year_Juv_reals) <-
  c("Year", "Year_apparent", "Year_apparent_lcl", "Year_apparent_ucl", "Age")

```

```

# clean dataframe
Year_Adult_reals <- Year_Age_reals[which(Year_Age_reals$Age == "A"),]
Year_Adult_reals <- Year_Adult_reals[,c(12, 6, 8, 9)]
row.names(Year_Adult_reals) <- NULL
Year_Adult_reals$Age <- "Adult"
colnames(Year_Adult_reals) <-
  c("Year", "Year_apparent", "Year_apparent_lcl", "Year_apparent_ucl", "Age")

```

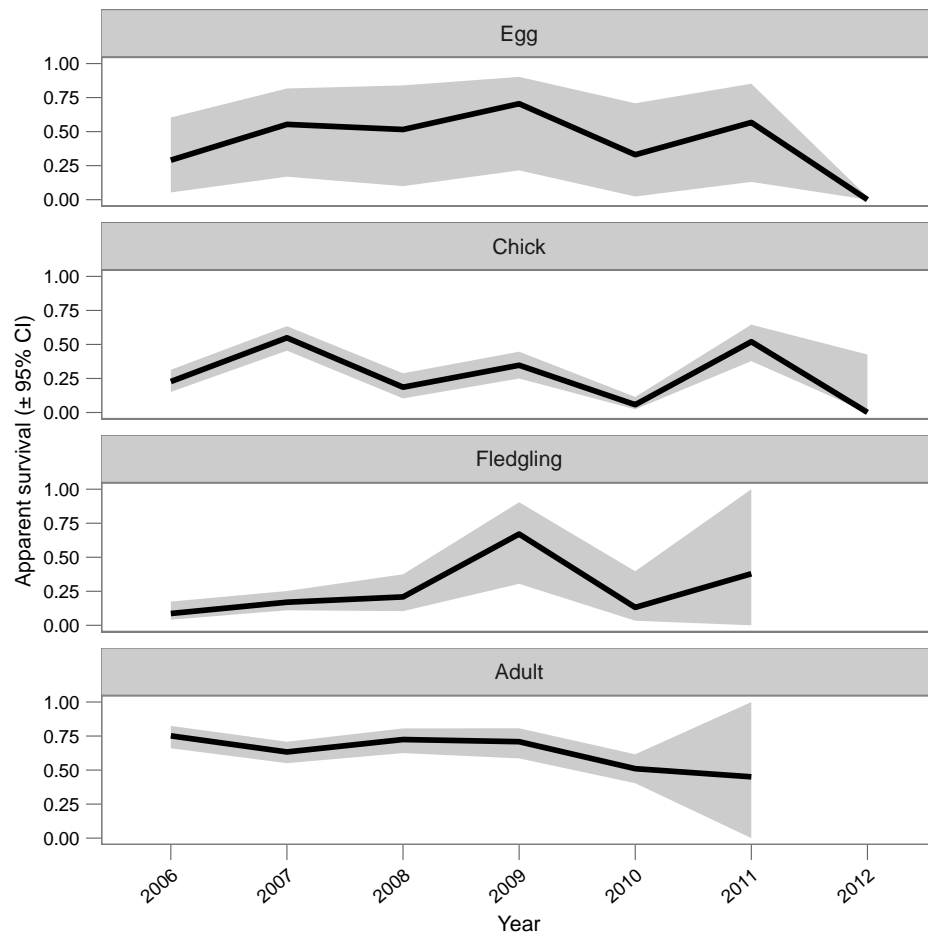
Vizualize annual variation in stage-specific survival (Figure 1b)

```

# combine all estimates into one dataframe
All_survival_rates <-
  rbind(Yearly_nest_survival_avgs, Chick_survival_reals,
        Year_Juv_reals, Year_Adult_reals)
All_survival_rates$Age <-
  factor(All_survival_rates$Age,
        levels = c("Egg", "Chick", "Fledgling", "Adult"))

# plot the trends while facetting by Age class
cbPalette <- c("#000000", "#000000", "#000000", "#000000")
All_survival_rates_plot <-
  ggplot(All_survival_rates, aes(x = Year, y = Year_apparent, group = Age)) +
    theme_bw() +
    geom_line(size = 1, aes(colour = Age)) +
    geom_ribbon(aes(ymin = Year_apparent_lcl, ymax = Year_apparent_ucl, fill = Age),
              alpha = 0.2) +
    theme(text=element_text(family="Arial"),
          legend.position = "none",
          axis.title.x = element_text(size=8, vjust=-0.1),
          axis.text.x = element_text(angle=40, size=7, vjust=1, hjust=1.01),
          axis.title.y = element_text(size=8, vjust=1.2),
          axis.text.y = element_text(size=7),
          panel.grid.major = element_blank(),
          strip.text = element_text(size=8),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
          axis.ticks.length = unit(0.2, "cm"),
          axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
    xlab("Year") +
    scale_colour_manual(values=cbPalette, name="Age") +
    scale_fill_manual(values=cbPalette, name="Age") +
    ylab("Apparent survival ( $\pm$  95% CI)") +
    facet_wrap(~ Age, ncol = 1)
All_survival_rates_plot

```



Calculate male annual fecundity

```
# Import male reproductive success dataset
male_fecundity <- read.table("Data_files/fecundity_data.txt",
                             header = TRUE,
                             colClasses = c("factor", "factor", "numeric"))

# calculate avg total eggs tended per male in each year between 2006 and 2012
male_fecundity_annual_summary <- Rmisc::summarySE(male_fecundity,
                                                  measurevar = "Eggs",
                                                  groupvars = c("Year")) [, c(1, 3, 6)]

# Consolidate all annual vital rates into a dataframe
male_fecundity_annual_summary$lcl <-
  male_fecundity_annual_summary$Eggs - male_fecundity_annual_summary$ci
male_fecundity_annual_summary$ucl <-
  male_fecundity_annual_summary$Eggs + male_fecundity_annual_summary$ci
male_fecundity_annual_summary$Stage <- "Fecundity"
male_fecundity_annual_summary <- male_fecundity_annual_summary[, -c(3)]
colnames(male_fecundity_annual_summary) <- c("Year", "Rate", "lcl", "ucl", "Stage")
colnames(All_survival_rates) <- c("Year", "Rate", "lcl", "ucl", "Stage")
All_vital_rates <- data.frame(rbind(All_survival_rates, male_fecundity_annual_summary))
```

Construct annual matrices and grand matrix

```
# Extract annual Nest Survival vital rates:
Nest_2006 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2006"), c("Rate")]
Nest_2007 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2007"), c("Rate")]
Nest_2008 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2008"), c("Rate")]
Nest_2009 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2009"), c("Rate")]
Nest_2010 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2010"), c("Rate")]
Nest_2011 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2011"), c("Rate")]
Nest_2012 <- All_vital_rates[which(All_vital_rates$Stage == "Egg" &
                                   All_vital_rates$Year == "2012"), c("Rate")]

# Extract annual Chick Survival vital rates:
Chick_2006 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2006"), c("Rate")]
Chick_2007 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2007"), c("Rate")]
Chick_2008 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2008"), c("Rate")]
Chick_2009 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2009"), c("Rate")]
Chick_2010 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2010"), c("Rate")]
Chick_2011 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2011"), c("Rate")]
Chick_2012 <- All_vital_rates[which(All_vital_rates$Stage == "Chick" &
                                   All_vital_rates$Year == "2012"), c("Rate")]

# Extract annual fledgling Survival vital rates:
Fledgling_2006 <- All_vital_rates[which(All_vital_rates$Stage == "Fledgling" &
                                       All_vital_rates$Year == "2006"), c("Rate")]
Fledgling_2007 <- All_vital_rates[which(All_vital_rates$Stage == "Fledgling" &
                                       All_vital_rates$Year == "2007"), c("Rate")]
Fledgling_2008 <- All_vital_rates[which(All_vital_rates$Stage == "Fledgling" &
                                       All_vital_rates$Year == "2008"), c("Rate")]
Fledgling_2009 <- All_vital_rates[which(All_vital_rates$Stage == "Fledgling" &
                                       All_vital_rates$Year == "2009"), c("Rate")]
Fledgling_2010 <- All_vital_rates[which(All_vital_rates$Stage == "Fledgling" &
                                       All_vital_rates$Year == "2010"), c("Rate")]
Fledgling_2011 <- All_vital_rates[which(All_vital_rates$Stage == "Fledgling" &
                                       All_vital_rates$Year == "2011"), c("Rate")]

# Extract annual Adult Survival vital rates:
Adult_2006 <- All_vital_rates[which(All_vital_rates$Stage == "Adult" &
                                   All_vital_rates$Year == "2006"), c("Rate")]
Adult_2007 <- All_vital_rates[which(All_vital_rates$Stage == "Adult" &
                                   All_vital_rates$Year == "2007"), c("Rate")]
```

```

Adult_2008 <- All_vital_rates[which(All_vital_rates$Stage == "Adult" &
                                   All_vital_rates$Year == "2008"), c("Rate")]
Adult_2009 <- All_vital_rates[which(All_vital_rates$Stage == "Adult" &
                                   All_vital_rates$Year == "2009"), c("Rate")]
Adult_2010 <- All_vital_rates[which(All_vital_rates$Stage == "Adult" &
                                   All_vital_rates$Year == "2010"), c("Rate")]
Adult_2011 <- All_vital_rates[which(All_vital_rates$Stage == "Adult" &
                                   All_vital_rates$Year == "2011"), c("Rate")]

# Extract annual Male Fecundity vital rates:
Fecundity_2006 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2006"), c("Rate")]
Fecundity_2007 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2007"), c("Rate")]
Fecundity_2008 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2008"), c("Rate")]
Fecundity_2009 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2009"), c("Rate")]
Fecundity_2010 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2010"), c("Rate")]
Fecundity_2011 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2011"), c("Rate")]
Fecundity_2012 <- All_vital_rates[which(All_vital_rates$Stage == "Fecundity" &
                                   All_vital_rates$Year == "2012"), c("Rate")]

# Define the life-stages of the matrix model
stages <- c("1st_year", "Adult")

# Construct grand avg. population projection matrix
Grand_vital_rates <- All_vital_rates %>%
  group_by(Stage) %>%
  summarise(avg = mean(Rate))
Grand_vital_rates <- data.frame(Grand_vital_rates)

Ceuta_Grand_Avg <- matrix(c(0,
                          Grand_vital_rates[5, 2],
                          (Grand_vital_rates[1, 2]*Grand_vital_rates[2, 2]*
                           Grand_vital_rates[3, 2]),
                          Grand_vital_rates[4, 2]),
                          nrow = 2, byrow = TRUE,
                          dimnames = list(stages, stages))

# overall population growth rate (lambda) derived from grand matrix
eigen(Ceuta_Grand_Avg)$values[1]

# Construct annual population projection matrices
Ceuta_2007 <- matrix(c(0,
                      Fecundity_2007,
                      (Nest_2007*Chick_2007*Fledgling_2006),
                      Adult_2006),
                      nrow = 2, byrow = TRUE,
                      dimnames = list(stages, stages))

```

```

Ceuta_2008 <- matrix(c(0,
                      Fecundity_2008,
                      (Nest_2008*Chick_2008*Fledgling_2007),
                      Adult_2007),
                    nrow = 2, byrow = TRUE,
                    dimnames = list(stages, stages))

Ceuta_2009 <- matrix(c(0,
                      Fecundity_2009,
                      (Nest_2009*Chick_2009*Fledgling_2008),
                      Adult_2008),
                    nrow = 2, byrow = TRUE,
                    dimnames = list(stages, stages))

Ceuta_2010 <- matrix(c(0,
                      Fecundity_2010,
                      (Nest_2010*Chick_2010*Fledgling_2009),
                      Adult_2009),
                    nrow = 2, byrow = TRUE,
                    dimnames = list(stages, stages))

Ceuta_2011 <- matrix(c(0,
                      Fecundity_2011,
                      (Nest_2011*Chick_2011*Fledgling_2010),
                      Adult_2010),
                    nrow = 2, byrow = TRUE,
                    dimnames = list(stages, stages))

Ceuta_2012 <- matrix(c(0,
                      Fecundity_2012,
                      (Nest_2012*Chick_2012*Fledgling_2011),
                      Adult_2011),
                    nrow = 2, byrow = TRUE,
                    dimnames = list(stages, stages))

# Organize annual matrices into a list
Ceuta_Annual_Matrices <- list(Ceuta_2007, Ceuta_2008, Ceuta_2009,
                             Ceuta_2010, Ceuta_2011, Ceuta_2012)

```

Model validation diagnostics

```

# Stochastic projection function that projects population growth for "tmax"
# time steps by randomly choosing matrices from "matrices" (a list of
# pre-defined matrices). The function runs "nreps" number of iterations,
# and starts with a population distribution of "n0". The extinction
# probability is calculated as the number of iterations that fell below
# the pre-defined quasi-extinction threshold of "Quasi_Ex".
# NOTE: This function was modified from the "stoch.projection()" function in the
# popbio package (Chris Stubben 2016)

stoch.projection.luke <-

```

```

function (matrices, n0, tmax = 50, nreps = 5000, Quasi_Ex = 0)
{
  Adults <- matrix(numeric(nreps * tmax), nrow = nreps)
  Eggs <- matrix(numeric(nreps * tmax), nrow = nreps)
  Adults[,1] <- n0[2]
  Eggs[,1] <- n0[1]
  for (i in 1:nreps) {
    A <- sample(matrices, (tmax-1), replace = TRUE)
    for (j in 1:(tmax-1)) {
      B <- A[[j]] %*% c(Eggs[i,j], Adults[i,j])
      Eggs[i,(j+1)] <- B[1]
      Adults[i,(j+1)] <- B[2]
    }
  }
  Adults_CV <- apply(Adults, 1, sd)/apply(Adults, 1, mean)
  Adults_Final <- Adults[,tmax]
  Adults_melted <- melt(t(Adults))
  Adults_ExProb <- sum(Adults_Final < Quasi_Ex)/nreps
  colnames(Adults_melted) <- c("Year", "Iteration", "Adults")
  Sim_Avg <- Adults_melted %>%
    group_by(Year) %>%
    summarise(mean(Adults))
  Adults <- list(Adults_CV = Adults_CV, Adults_ExProb = Adults_ExProb,
               Adults_Final = Adults_Final, Adults_melted = Adults_melted,
               Sim_Avg = Sim_Avg)
  Adults
}

# Store the actual population trend observed between 2006 and 2012
Actual <- c(204, 190, 98, 102, 131, 96, 58)

# Store the stage distribution observed in 2006 and 2012 and name the elements
n2006 <- c(454, 204)
names(n2006) <- c("Eggs", "Adults")
n2012 <- c(99, 58)
names(n2012) <- c("Eggs", "Adults")

# Run the stochastic projection over the seven-year period
Seven_year_sim <-
  stoch.projection.luke(matrices = Ceuta_Annual_Matrices, n0 = n2006,
                       tmax = 7, nreps = 10000, Quasi_Ex = 2)

# Specify the years in the melted dataframe
Seven_year_sim$Adults_melted$Year <-
  c("2006", "2007", "2008", "2009", "2010", "2011", "2012")

# group the iterations by year and calculate the annual averages
Sim_Avg <- Seven_year_sim$Adults_melted %>%
  group_by(Year) %>%
  summarise(mean(Adults))

# group the iterations by year and calculate the annual IQRs
Sim_summary <- Seven_year_sim$Adults_melted %>%

```



```

group_by(Year) %>%
  summarise(IQR(Adults))

# calculate the upper and lower 50% boundaries of the simulation
Sim_summary$upper <- Sim_Avg$`mean(Adults)`+Sim_summary$IQR(Adults)/2
Sim_summary$lower <- Sim_Avg$`mean(Adults)`-Sim_summary$IQR(Adults)/2

# calculate the average population size for each year across all iterations
Sim_summary$Sim <- Sim_Avg$`mean(Adults)`

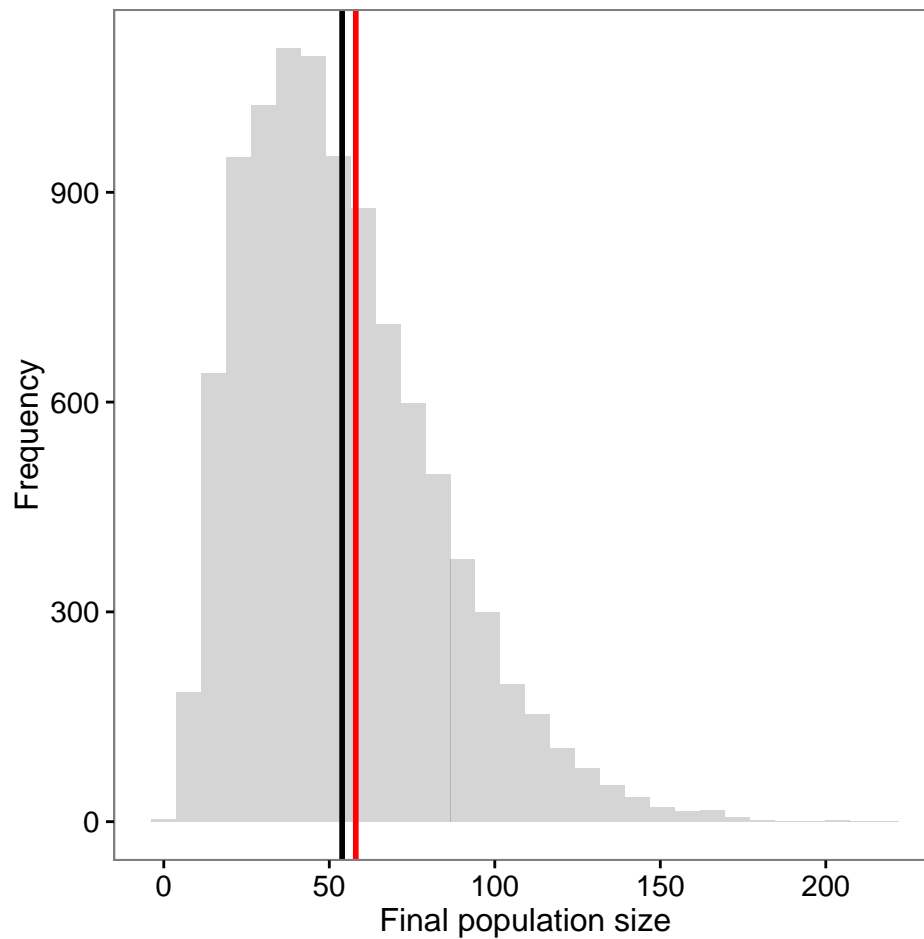
# Tidy up summary dataframe
Sim_summary <- Sim_summary[,-2]
Sim_summary$Actual <- c(204, 190, 98, 102, 131, 96, 58)
Sim_summary <- melt(Sim_summary)
colnames(Sim_summary) <- c("Year", "Iteration", "Adults")

# Calculate the mean final population size of the simulation
Actual_final <- Actual[length(Actual)]
Final_mean <- mean(Seven_year_sim$Adults_Final)
Final_SD <- sd(Seven_year_sim$Adults_Final)

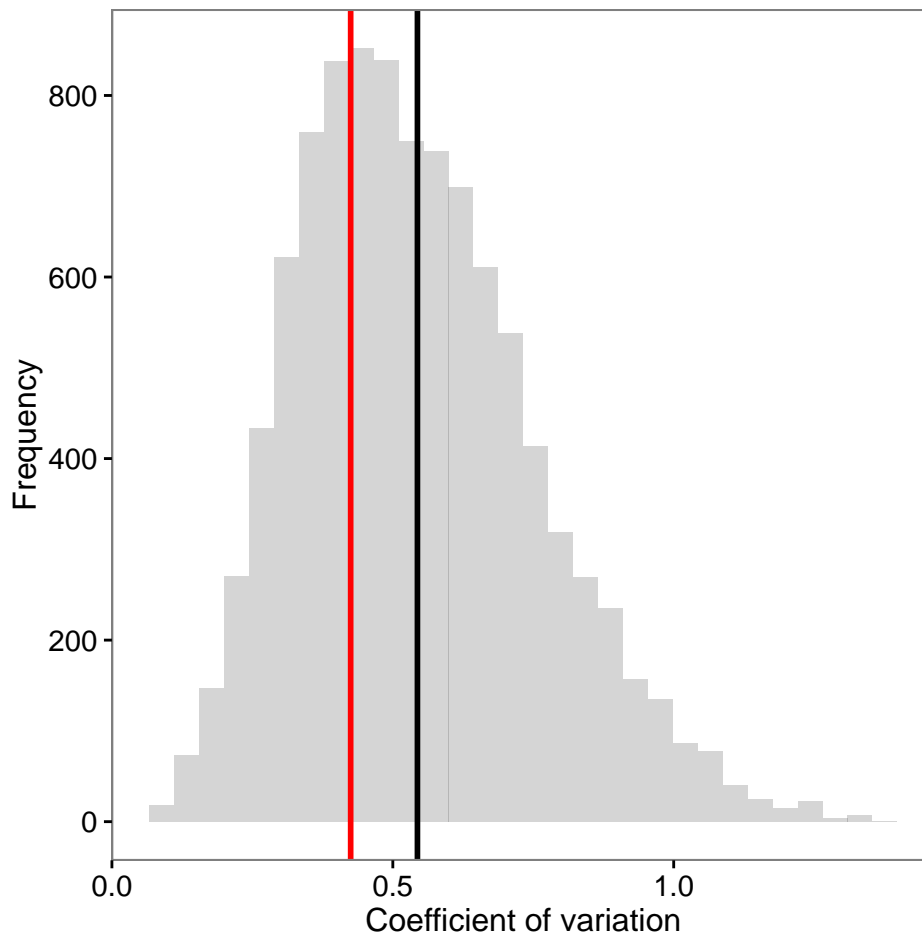
CV_actual <- sd(Actual)/mean(Actual)
CV_mean <- mean(Seven_year_sim$Adults_CV)
CV_sd <- sd(Seven_year_sim$Adults_CV)

# Histogram plot of final population sizes with verticle lines illustrating
# actual final population size and the mean of the simulation
Final_Population_Plot <-
  ggplot(NULL) +
    theme_bw() +
    geom_histogram(aes(x = Seven_year_sim$Adults_Final), alpha = 0.25) +
    geom_vline(xintercept = Actual_final, color = "red", size=1) +
    geom_vline(xintercept = Final_mean, color = "black", size=1) +
    theme(text=element_text(family="Arial"),
          axis.title.x = element_text(size=12, vjust=-0.1),
          axis.text.x = element_text(size=11),
          axis.title.y = element_text(size=12, vjust=1.2),
          axis.text.y = element_text(size=11),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          strip.text = element_blank(),
          strip.background = element_blank()) +
    xlab("Final population size") +
    ylab("Frequency")
Final_Population_Plot

```



```
# Histogram plot of coefficient of variations with verticle lines illustrating
# actual final population size and the mean of the simulation
CV_Plot <-
  ggplot(NULL) +
    theme_bw() +
    geom_histogram(aes(x = Seven_year_sim$Adults_CV), alpha = 0.25) +
    geom_vline(xintercept = CV_actual, color = "red", size=1) +
    geom_vline(xintercept = CV_mean, color = "black", size=1) +
    theme(text=element_text(family="Arial"),
          axis.title.x = element_text(size=12, vjust=-0.1),
          axis.text.x = element_text(size=11),
          axis.title.y = element_text(size=12, vjust=1.2),
          axis.text.y = element_text(size=11),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          strip.text = element_blank(),
          strip.background = element_blank()) +
    xlab("Coefficient of variation") +
    ylab("Frequency")
CV_Plot
```

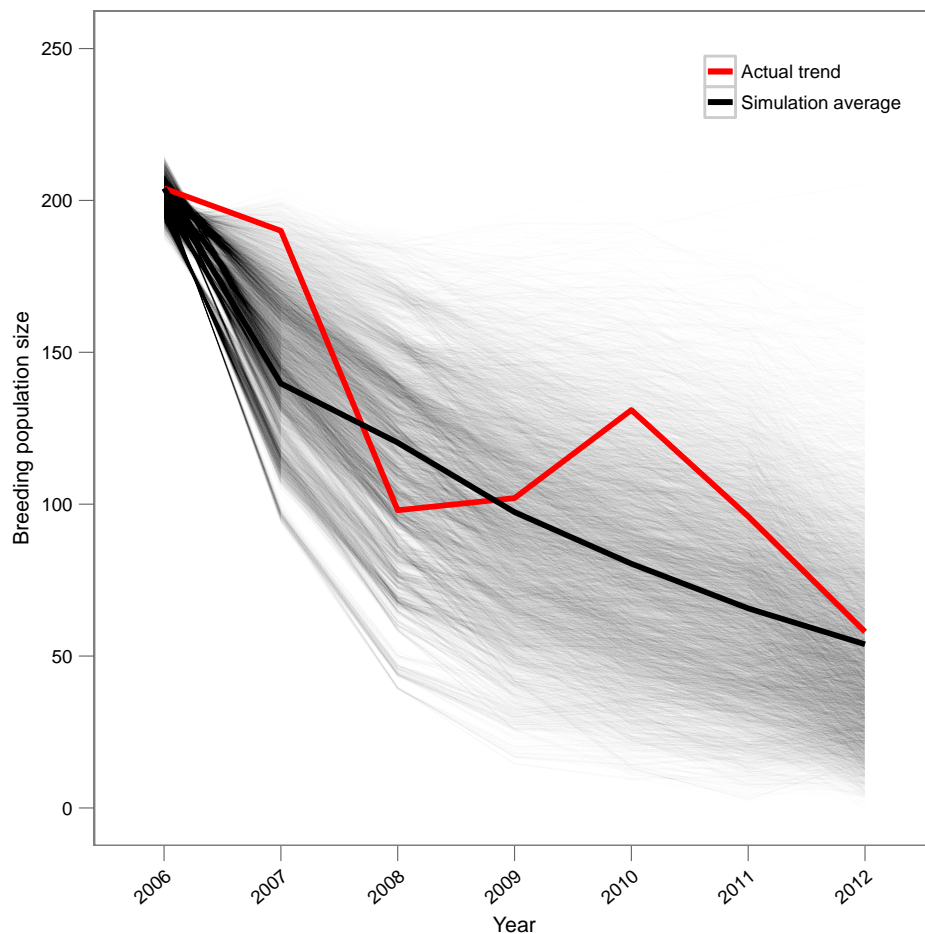


```
# Line plot of simulation trajectories with trend lines illustrating
# actual final population dynamics between 2006 to 2012 and the mean of the
# simulation
Stochastic_simulation_plot <-
  ggplot(NULL) +
    theme_bw() +
    geom_line(data = Seven_year_sim$Adults_melted,
              aes(x = Year, y = Adults, group = Iteration),
              size = 0.25, alpha = 0.008, stat = "smooth",
              method = "lm", formula = y ~ poly(x,3)) +
    geom_line(data = Sim_summary[c(22:28),],
              aes(x = Year, y = Adults, group = Iteration, color = Iteration),
              size = 1, alpha = 1) +
    geom_line(data = Sim_summary[c(15:21),],
              aes(x = Year, y = Adults, group = Iteration, color = Iteration),
              size = 1, alpha = 1) +
    scale_colour_manual("", labels = c("Actual trend", "Simulation average"),
                        values = c("red", "black")) +
    theme(text=element_text(family="Arial"),
          legend.position = c(1, 1),
          legend.justification = c(1, 1),
          legend.background = element_rect(fill=NA),
          legend.text = element_text(size=7),
```

```

legend.title = element_blank(),
legend.key.height = unit(0.8,"line"),
legend.key.width = unit(0.8,"line"),
axis.title.x = element_text(size=8, vjust=-0.1),
axis.text.x = element_text(angle=40, size=7, vjust=1, hjust=1.01),
axis.title.y = element_text(size=8, vjust=1.2),
axis.text.y = element_text(size=7),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
axis.ticks.length = unit(0.2, "cm"),
axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
  xlab("Year") +
  ylab("Breeding population size") +
  scale_y_continuous(limits = c(0, 250))
Stochastic_simulation_plot

```



Forecast stochastic simulation

```

#~~~~~#
# Future extinction probability analysis

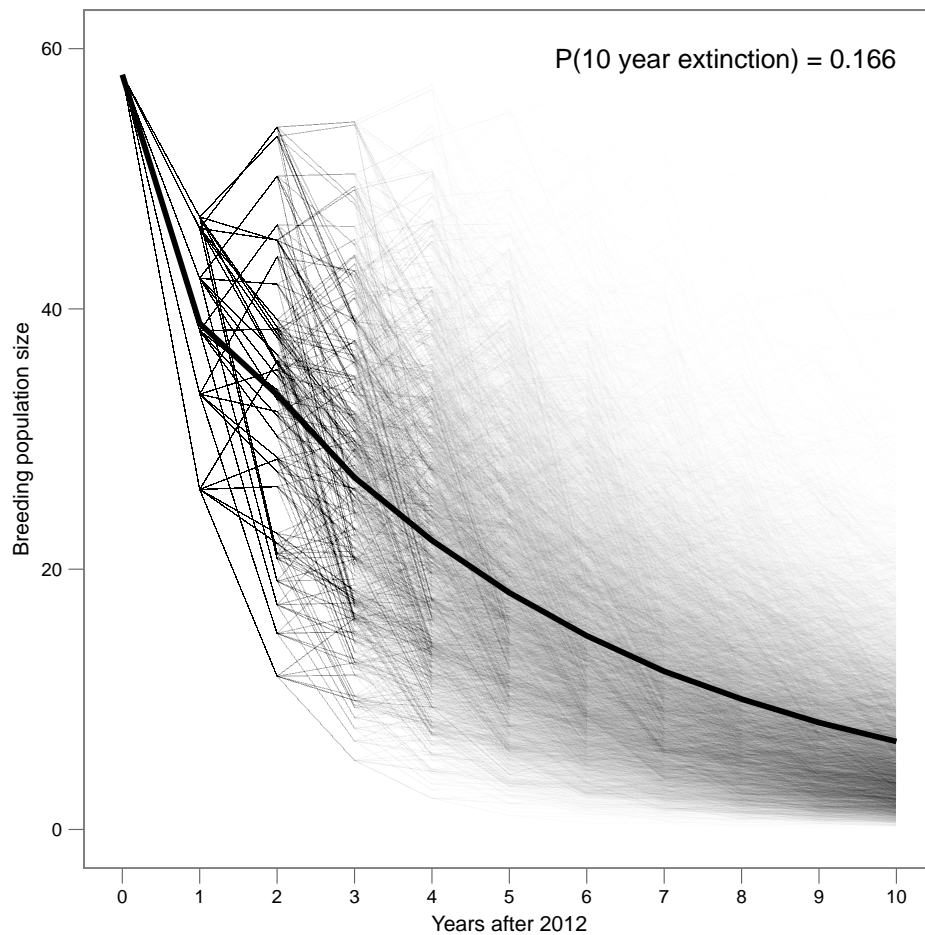
```

```

#~~~~~#
# Run the stochastic projection for 10, 25, and 50 years into the future
ten_Year_Stoch_Proj <- stoch.projection.luke(matrices = Ceuta_Annual_Matrices,
                                             n0 = n2012, tmax = 11,
                                             nreps = 10000, Quasi_Ex = 2)
twentyfive_Year_Stoch_Proj <- stoch.projection.luke(matrices = Ceuta_Annual_Matrices,
                                                    n0 = n2012, tmax = 26,
                                                    nreps = 10000, Quasi_Ex = 2)
fifty_Year_Stoch_Proj <- stoch.projection.luke(matrices = Ceuta_Annual_Matrices,
                                              n0 = n2012, tmax = 51,
                                              nreps = 10000, Quasi_Ex = 2)

# Line plots of simulation trajectories with trend lines illustrating
# the predicted population dynamics post-2012 for 10, 25, and 50 years into
# the future
ten_Year_Stoch_Proj_plot <- {
  ggplot(NULL) +
    theme_bw() +
    geom_line(data = ten_Year_Stoch_Proj$Adults_melted,
              aes(x = (Year-1), y = Adults, group = Iteration),
              size = 0.25, alpha = 0.008) +
    geom_line(data = ten_Year_Stoch_Proj$Sim_Avg,
              aes(x = (Year-1), y = `mean(Adults)`),
              size = 1, alpha = 1) +
    scale_colour_manual("", values = c("black", "black")) +
    theme(text=element_text(family="Arial"),
          axis.title.x = element_text(size=8, vjust=-0.1),
          axis.text.x = element_text(size=7),
          axis.title.y = element_text(size=8, vjust=1.2),
          axis.text.y = element_text(size=7),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
          axis.ticks.length = unit(0.2, "cm"),
          axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
    ylab("Breeding population size") +
    xlab("Years after 2012") +
    annotate("text", x = 10, y = 60, vjust = 1, hjust = 1,
             label = "P(10 year extinction) = 0.166", size = 3.5, family="Arial") +
    scale_y_continuous(limits = c(0, 60)) +
    scale_x_continuous(limits = c(0, 10), breaks = 0:10)
}
ten_Year_Stoch_Proj_plot

```

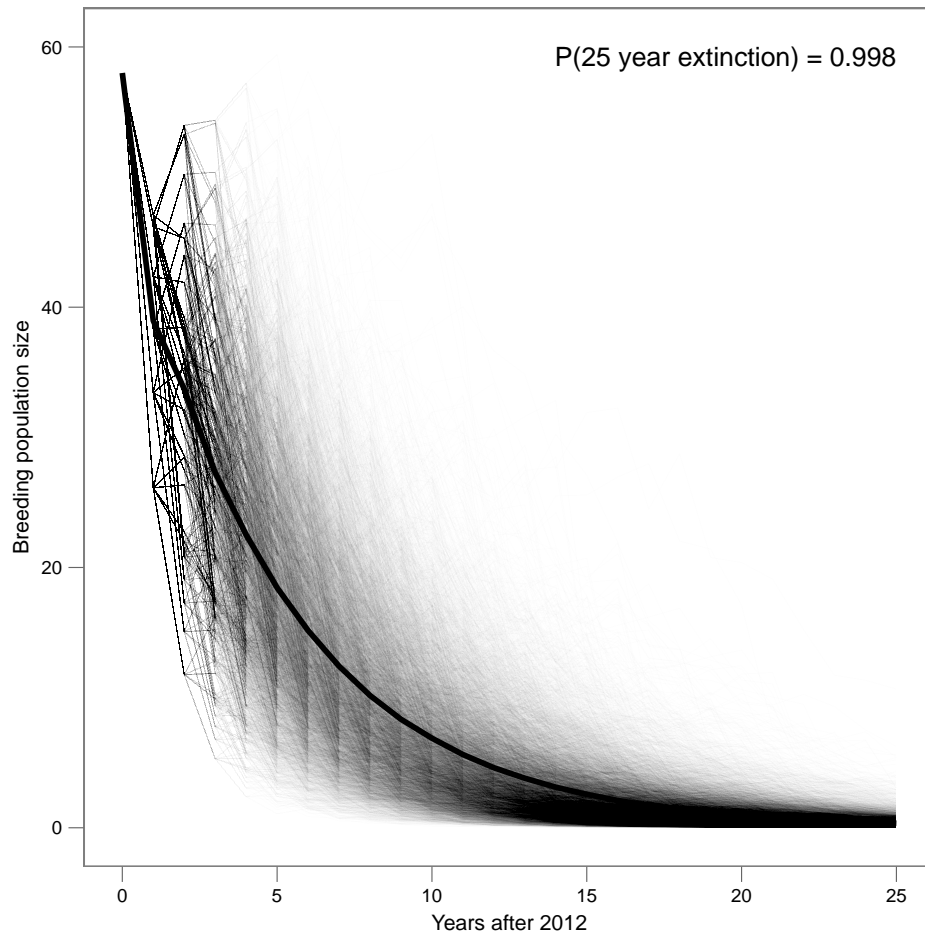


```
twtyfive_Year_Stoch_Proj_plot <- {
  ggplot(NULL) +
    theme_bw() +
    geom_line(data = twtyfive_Year_Stoch_Proj$Adults_melted,
              aes(x = (Year-1), y = Adults, group = Iteration),
              size = 0.25, alpha = 0.008) +
    geom_line(data = twtyfive_Year_Stoch_Proj$Sim_Avg,
              aes(x = (Year-1), y = `mean(Adults)`),
              size = 1, alpha = 1) +
    scale_colour_manual("", values = c("black", "black")) +
    theme(text=element_text(family="Arial"),
          axis.title.x = element_text(size=8, vjust=-0.1),
          axis.text.x = element_text(size=7),
          axis.title.y = element_text(size=8, vjust=1.2),
          axis.text.y = element_text(size=7),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
          axis.ticks.length = unit(0.2, "cm"),
          axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
    ylab("Breeding population size") +
    xlab("Years after 2012") +
    annotate("text", x = 25, y = 60, vjust = 1, hjust = 1,
```

```

    label = "P(25 year extinction) = 0.998", size = 3.5, family="Arial") +
    scale_y_continuous(limits = c(0, 60)) +
    scale_x_continuous(limits = c(0, 25), breaks = c(0,5,10,15,20,25))
  }
twtyfive_Year_Stoch_Proj_plot

```



```

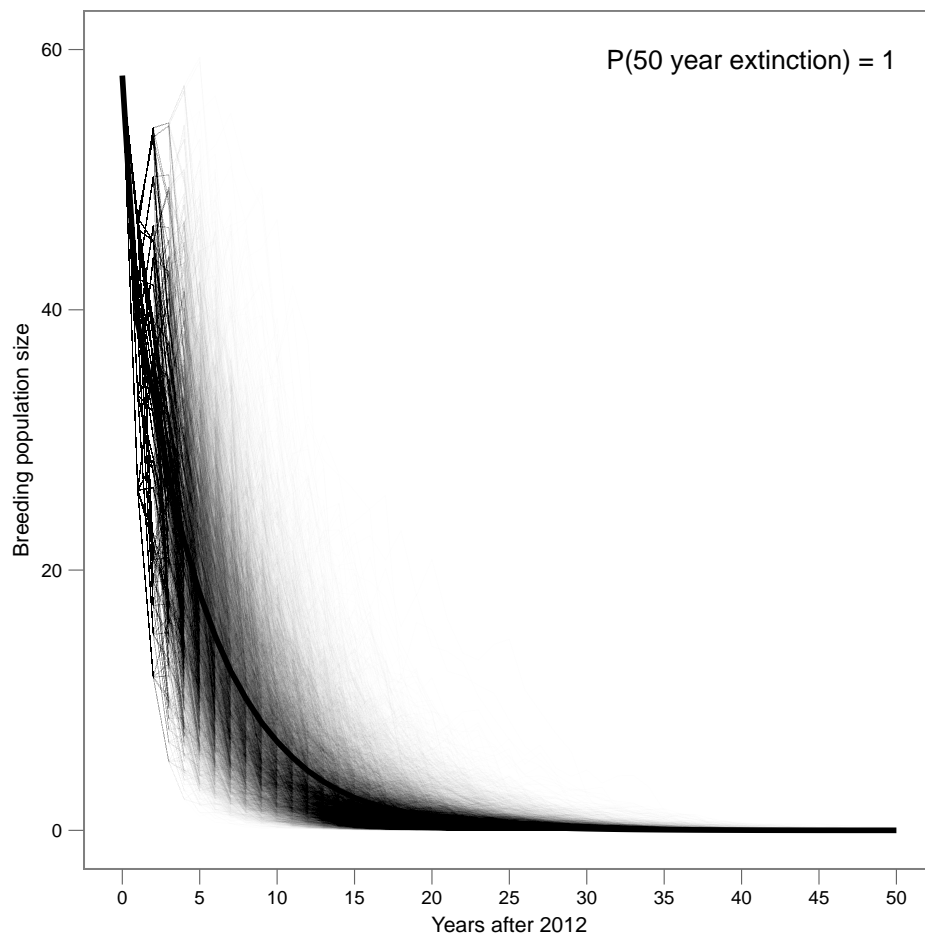
fifty_Year_Stoch_Proj_plot <- {
  ggplot(NULL) +
    theme_bw() +
    geom_line(data = fifty_Year_Stoch_Proj$Adults_melted,
              aes(x = (Year-1), y = Adults, group = Iteration),
              size = 0.25, alpha = 0.008) +
    geom_line(data = fifty_Year_Stoch_Proj$Sim_Avg,
              aes(x = (Year-1), y = `mean(Adults)`),
              size = 1, alpha = 1) +
    scale_colour_manual("", values = c("black", "black")) +
    theme(text=element_text(family="Arial"),
          axis.title.x = element_text(size=8, vjust=-0.1),
          axis.text.x = element_text(size=7),
          axis.title.y = element_text(size=8, vjust=1.2),
          axis.text.y = element_text(size=7),
          panel.grid.major = element_blank(),

```

```

    panel.grid.minor = element_blank(),
    axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
    axis.ticks.length = unit(0.2, "cm"),
    axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
  ylab("Breeding population size") +
  xlab("Years after 2012") +
  annotate("text", x = 50, y = 60, vjust = 1, hjust = 1,
    label = "P(50 year extinction) = 1", size = 3.5, family="Arial") +
  scale_y_continuous(limits = c(0, 60)) +
  scale_x_continuous(limits = c(0, 50),
    breaks = c(0,5,10,15,20,25,30,35,40,45,50))
}
fifty_Year_Stoch_Proj_plot

```



Sensitivity analysis

```

# Lower-level vital rate sensitivities
# Set up the matrix and lower-level elements
Ceuta.vr<-list(Egg = Grand_vital_rates[1, 2], Chick = Grand_vital_rates[2, 2],
  Fledg. = Grand_vital_rates[3, 2], Adult = Grand_vital_rates[4, 2],
  Fecund. = Grand_vital_rates[5, 2])

```



```

Ceuta.el<-expression(0, Fecund.,
                    Egg*Chick*Fledg., Adult )

# check relative responses of growth to perturbations of vital rates
# NOTE: code modified from the vitalsens() example in the
# popbio package (Chris Stubben 2016)
n <- length(Ceuta.vr)
vr <- seq(0,1,.1)
vrсен <- matrix(numeric(n*length(vr)), ncol = n, dimnames = list(vr, names(Ceuta.vr)))
for (h in 1:n)
{
  Ceuta.vr2<-list(Egg = Grand_vital_rates[1, 2], Chick = Grand_vital_rates[2, 2],
                 Fledg. = Grand_vital_rates[3, 2], Adult = Grand_vital_rates[4, 2],
                 Fecund. = Grand_vital_rates[5, 2])
  for (i in 1:length(vr))
  {
    Ceuta.vr2[[h]] <- vr[i]
    A <- matrix(sapply(Ceuta.el, eval, Ceuta.vr2 , NULL),
               nrow = sqrt(length(Ceuta.el)), byrow = TRUE)
    vrсен[i, h] <- max(Re(eigen(A)$values))
  }
}

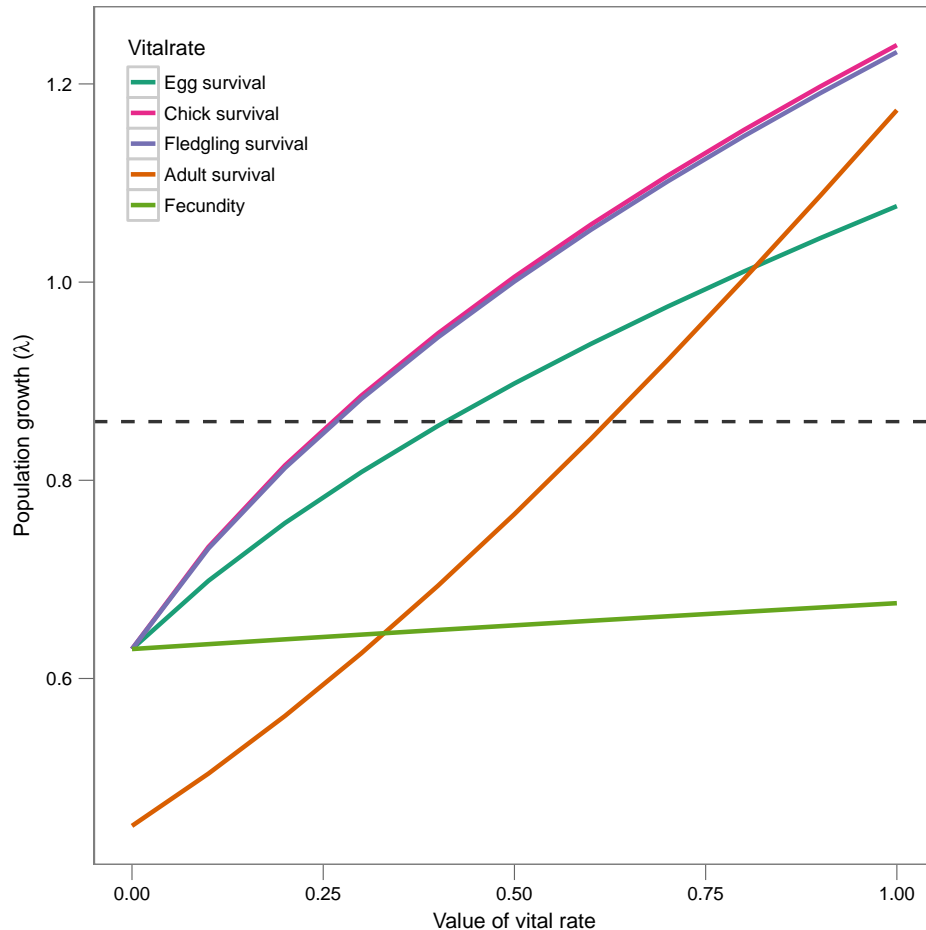
# Sensitivity plot
colnames(vrсен) <-
  c("Egg survival", "Chick survival", "Fledgling survival",
    "Adult survival", "Fecundity")
Sensitivities <- melt(vrсен)
colnames(Sensitivities) <- c("Perturbation", "Vitalrate", "Sensitivity")
cbPalette <- c("#1b9e77", "#e7298a", "#7570b3", "#d95f02", "#66a61e")
Sensitivity_plot <-
  ggplot(Sensitivities, aes(x = Perturbation, y = Sensitivity, group = Vitalrate)) +
    theme_bw() +
    geom_line(size = 0.8, aes(colour = Vitalrate)) +
    geom_hline(yintercept = 0.8592153, color = "black", size = 0.6, linetype = "dashed",
              alpha = 0.8) +
    theme(text=element_text(family = "Arial"),
          legend.position = c(0, 1),
          legend.justification = c(0, 1),
          legend.text=element_text(size = 7),
          legend.title=element_text(size = 8),
          legend.key.height=unit(0.8, "line"),
          legend.key.width=unit(0.8, "line"),
          legend.background = element_rect(fill = NA),
          axis.title.x = element_text(size = 8, vjust=-0.1),
          axis.text.x = element_text(size = 7),
          axis.title.y = element_text(size = 8, vjust = 1.2),
          axis.text.y = element_text(size = 7),
          panel.grid.minor = element_blank(),
          panel.grid.major = element_blank(),
          axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
          axis.ticks.length = unit(0.2, "cm"),
          axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +

```

```

xlab("Value of vital rate") +
ylab(expression(paste("Population growth (", lambda, ")"))) +
scale_colour_manual(values=cbPalette, name="Vitalrate")
Sensitivity_plot

```



```

# check relative proportional responses of growth to perturbations of vital rates
# NOTE: code modified from the vitalsens() example in the
# popbio package (Chris Stubben 2016)
vr <- seq(0,2,.1)
vrсен <- matrix(numeric(n*length(vr)), ncol = n, dimnames = list(vr, names(Ceuta.vr)))
vrelas <- matrix(numeric(n*length(vr)), ncol = n, dimnames = list(vr, names(Ceuta.vr)))
for (h in 1:n)
{
  for (i in 1:length(vr))
  {
    Ceuta.vr2 <- list(Egg = Grand_vital_rates[1, 2], Chick = Grand_vital_rates[2, 2],
                     Fledg. = Grand_vital_rates[3, 2], Adult = Grand_vital_rates[4, 2],
                     Fecund. = Grand_vital_rates[5, 2])
    Ceuta.vr2[[h]] <- vr[i] * Ceuta.vr2[[h]]
    A <- matrix(sapply(Ceuta.el, eval, Ceuta.vr2, NULL), nrow = sqrt(length(Ceuta.el)),
                byrow = TRUE)
    vrelas[i, h] <- max(Re(eigen(A)$values))/eigen(Ceuta_Grand_Avg)$values[1]
  }
}

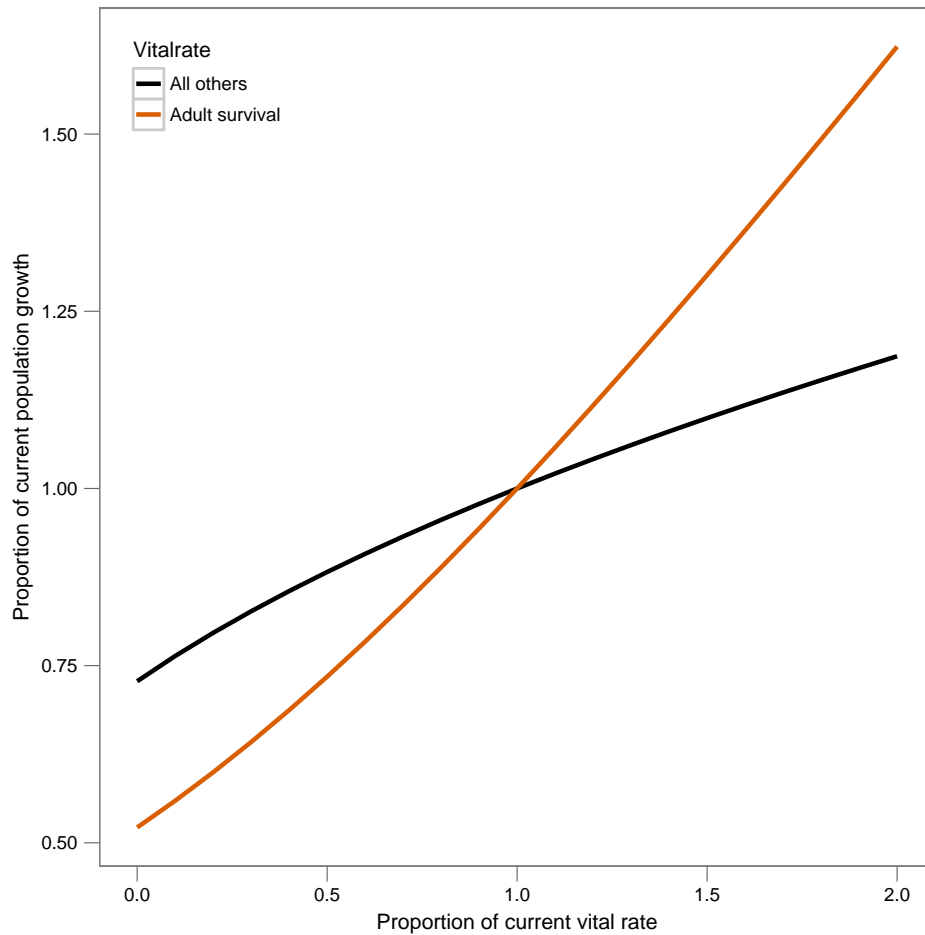
```

```

}
}

# Elasticity plot
reduced_vrelas <- vrelas[, c(1, 4)]
colnames(reduced_vrelas) <- c("All others", "Adult survival")
Elasticities <- melt(reduced_vrelas)
colnames(Elasticities) <- c("Perturbation", "Vitalrate", "Elasticity")
cbPalette <- c("#000000", "#d95f02")
Elasticity_plot <-
  ggplot(Elasticities, aes(x = Perturbation, y = Elasticity, group = Vitalrate)) +
    theme_bw() +
    geom_line(size = 0.8, aes(colour = Vitalrate)) +
    theme(text=element_text(family="Arial"),
          legend.position = c(0, 1),
          legend.justification = c(0, 1),
          legend.text = element_text(size=7),
          legend.title = element_text(size=8),
          legend.key.height = unit(0.8,"line"),
          legend.key.width = unit(0.8,"line"),
          legend.background = element_rect(fill = NA),
          axis.title.x = element_text(size = 8, vjust = -0.1),
          axis.text.x = element_text(size = 7),
          axis.title.y = element_text(size = 8, vjust = 1.2),
          axis.text.y = element_text(size = 7),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
          axis.ticks.length = unit(0.2, "cm"),
          axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
    xlab("Proportion of current vital rate") +
    ylab("Proportion of current population growth") +
    scale_colour_manual(values=cbPalette, name="Vitalrate")
Elasticity_plot

```



```
# then derive sensitivities and elasticities
x <- popbio::vitalsens(Ceuta.el, Ceuta.vr)
x$Vital_rate <- as.factor(rownames(x))
x_melt <- melt(x[, c(2:4)])
x_melt$Vital_rate <-
  factor(x_melt$Vital_rate,
    levels = c("Egg", "Chick", "Fledg.", "Adult", "Fecund.))
x_melt$variable <- ifelse(x_melt$variable == "sensitivity", "Sensitivity", "Elasticity")
cbPalette <- c("#1b9e77", "#d95f02", "#7570b3", "#e7298a", "#66a61e")
Sensitivity_Elasticity_histogram <-
  ggplot(x_melt, aes(x = Vital_rate, y = value, fill = variable)) +
    theme_bw() +
    geom_bar(stat = "identity", position = position_dodge()) +
    theme(text = element_text(family="Arial"),
      legend.position = c(0, 1),
      legend.justification = c(0, 1),
      legend.text=element_text(size = 8),
      legend.title=element_blank(),
      legend.key.height = unit(0.8,"line"),
      legend.key.width=unit(0.8,"line"),
      legend.background = element_rect(fill=NA),
      axis.title.x = element_text(size=8, vjust=-0.1),
      axis.text.x = element_text(size=7),
```

```

axis.title.y = element_text(size=8, vjust=1.2),
axis.text.y = element_text(size=7),
panel.grid.minor = element_blank(),
panel.grid.major = element_blank(),
axis.ticks.x = element_line(size = 0.2, colour = "grey40"),
axis.ticks.length = unit(0.2, "cm"),
axis.ticks.y = element_line(size = 0.2, colour = "grey40")) +
xlab("Vital rate") +
ylab("Sensitivity or elasticity of population growth rate") +
scale_fill_brewer(palette = "Set1")
Sensitivity_Elasticity_histogram

```

