Management Plan Figures

Dean Wilkinson

02/03/2022

Introduction

In this document I will be creating the figures that I will be using for my Amami rabbit management plan report. For graphs I will be using the packages **ggplot2** and **patchwork** but to create tables I will be using the package **knitr** and **kableExtra**.

```
library(ggplot2)
library(patchwork)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.1.2
```

```
library(kableExtra)
library(popdemo)
```

```
## Warning: package 'popdemo' was built under R version 4.1.2
```

```
## Welcome to popdemo! This is version 1.3-0
## Use ?popdemo for an intro, or browseVignettes('popdemo') for vignettes
## Citation for popdemo is here: doi.org/10.1111/j.2041-210X.2012.00222.x
## Development and legacy versions are here: github.com/iainmstott/popdemo
```

Estimating Rabbit Age Figures

The age-stages used in the matrix modelling were done based on the weight of individuals in each age-stage. I will summarise the catagories below:

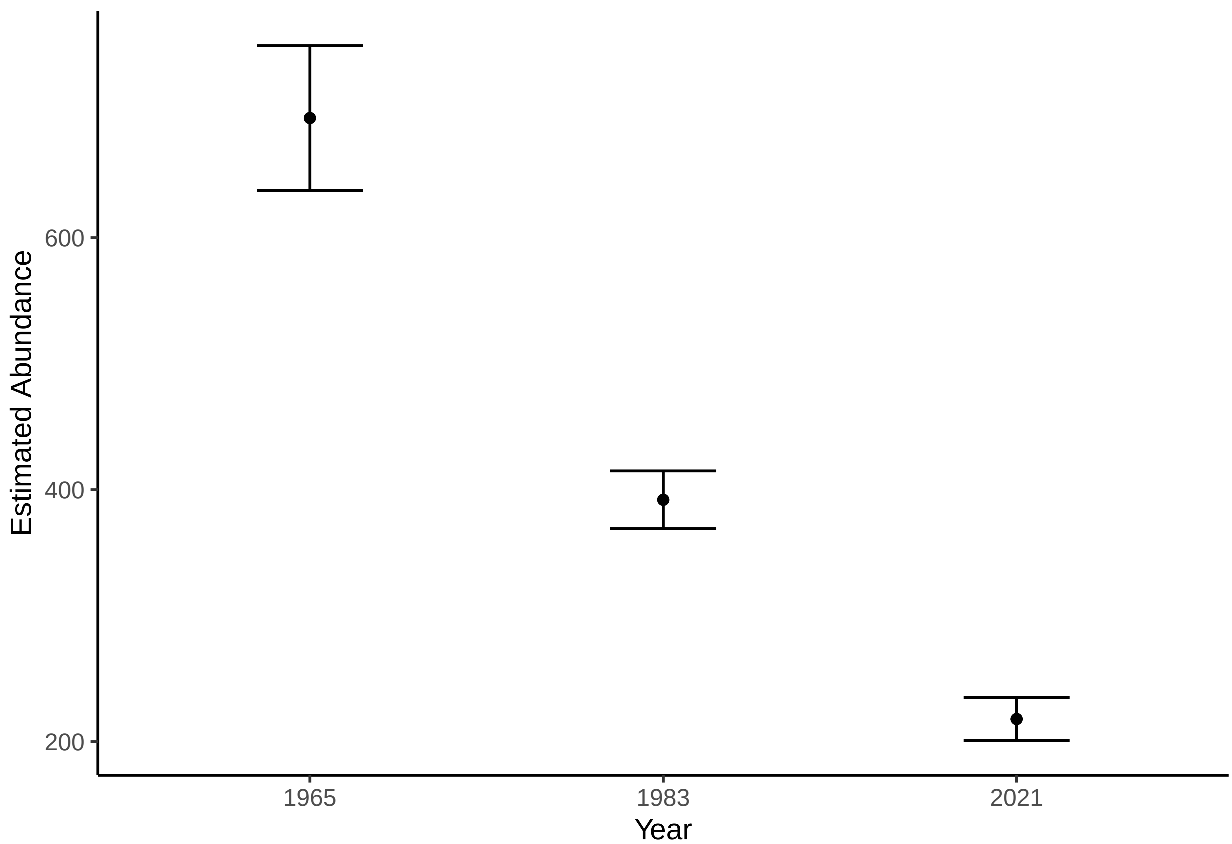
- 1yr-old: 0-2kg
- 2yr-old: 2-4kg
- adults: >4kg

```
# Create the data
tab.age = data.frame(Age = c("1yr Olds", "2yr Olds", "Adults"),
                     Weight = c("0-2", "2-4", "> 4"))

# Create the table
#kable(tab.age) %>%
#  kable_styling("striped") %>%
#  save_kable("Age-Stage Criteria.png")
```

Estimating Abundance Figures

The estimates for Amami rabbit population sizes for 1965, 1983 and 2021 are 695 +/- 29.3, 392 +/- 11.7, and 218 +/- 8.7. I will now make a point boxplot.

```
#Create the data
N = c(695, 392, 218)
Year = factor(c(1965, 1983, 2021))
se = c(29.3, 11.7, 8.7)
Abund = data.frame(N = N, Year = Year, se = se)

#Plot the data
p.Abund = ggplot(data = Abund) +
  geom_point(aes(x = Year, y = N)) +
  geom_errorbar(aes(Year, ymin = N - 1.96*se, ymax = N + 1.96*se), width = 0.3) +
  labs(x = "Year", y = "Estimated Abundance") +
  theme_classic()
p.Abund
```




```
#Save the plot
#ggsave("Abundance.png")
```

Next, I want to create tables for the model coefficients for the best-fitting models used to estimate the abundances in the graph above as well as the models used to estimate age-stage specific abundances.

First I will create the data to turn into tables.

```r
tab.1965 = data.frame(Statistics = c("Mean", "SE", "lwr", "upr"),
                      p1 = c(0.17, 0.01, 0.14, 0.20),
                      p2 = c(0.39, 0.01, 0.37, 0.42),
                      f0 = c(211, 29, 161, 277),
                      N = c(695, 29, 645, 761)
                      )
tab.1983 = data.frame(Statistics = c("Mean", "SE", "lwr", "upr"),
                      p = c(0.53, 0.02, 0.49, 0.56),
                      c = c(0.74, 0.02, 0.70, 0.78),
                      f0 = c(41, 18, 18, 94),
                      N = c(392, 18, 369, 445)
                      )
tab.adults = data.frame(Statistics = c("Mean", "SE", "lwr", "upr"),
                        p1 = c(0.36, 0.05, 0.27, 0.45),
                        p2 = c(0.45, 0.03, 0.38, 0.51),
                        f0 = c(20, 8, 10, 42),
                        N = c(105, 8, 92, 126)
                        )
tab.1yrold = data.frame(Statistics = c("Mean", "SE", "lwr", "upr"),
                        p = c(0.57, 0.03, 0.51, 0.64),
                        f0 = c(5, 1, 3, 8),
                        N = c(66, 1, 64, 69)
                        )
tab.2yrold = data.frame(Statistics = c("Mean", "SE", "lwr", "upr"),
                        p = c(0.49, 0.04, 0.41, 0.58),
                        f0 = c(6, 4, 2, 18),
                        N = c(47, 2, 43, 59)
                        )
```

Now I will create the tables and save them in my directory.

```r
#kable(tab.1965, row.names = FALSE, format = 'html') %>%
#  kable_styling("striped") %>%
#  save_kable("1965 Model.png")
#kable(tab.1983, row.names = FALSE, format = 'html') %>%
#  kable_styling("striped") %>%
#  save_kable("1983 Model.png")
#kable(tab.adults, row.names = FALSE, format = 'html') %>%
#  kable_styling("striped") %>%
#  save_kable("2021 Adults Model.png")
#kable(tab.1yrold, row.names = FALSE, format = 'html') %>%
#  kable_styling("striped") %>%
#  save_kable("2021 1yr old Model.png")
#kable(tab.2yrold, row.names = FALSE, format = 'html') %>%
#  kable_styling("striped") %>%
#  save_kable("2021 2yr old Model.png")
```

This concludes the "Estimating Abundance Figures" section, I will now move on to the "Estimating Survival Figures" section of the document.

Estimating Survival Figures

In this section I will plot the model predictions of the survival analysis for the Amami rabbit. For this I will create a line plot of how age specific survival changes with the distance to the nearest house with pets and how age specific survival changes with the presence and absence of java mongoose as this was the best fitting model. The model equation is given below:

logit(phi) = 0.145 + 0.784*2-yr-olds + 1.013*adults - 0.014*house - 0.151*mongoose

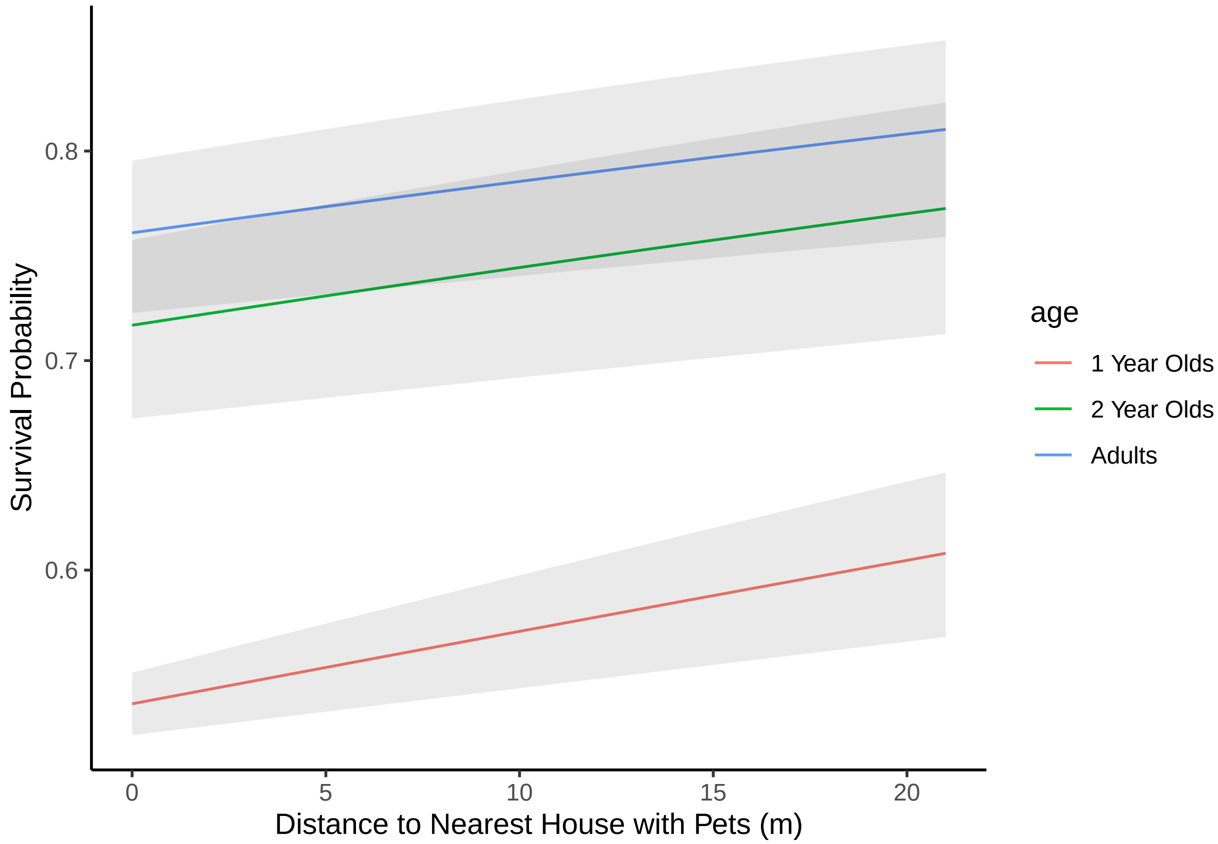
1yr_old SE = 0.06 2yr_old SE = 0.15 adult SE = 0.14 house SE = 0.005 mongoose SE = 0.17

First, the effect of **house** data.

```
#Create the predictor
house.dist = seq(from = 0, to = 21)

#Simulate responses
adult.trend = plogis(0.145 + 0.784*0 + 1.013*1 + 0.014*house.dist)
adult.lwr = plogis((0.145-0.06) + 0.784*0 + (1.013-0.14)*1 + (0.014-0.005)*house.dist)
adult.upr = plogis((0.145+0.06) + 0.784*0 + (1.013+0.14)*1 + (0.014+0.005)*house.dist)
Two_yr.trend = plogis(0.145 + 0.784*1 + 1.013*0 + 0.014*house.dist)
Two_yr.lwr = plogis((0.145-0.06) + (0.784-0.15)*1 + (1.013-0.14)*0 + (0.014-0.005)*house.dist)
Two_yr.upr = plogis((0.145+0.06) + (0.784+0.15)*1 + (1.013-0.14)*0 + (0.014+0.005)*house.dist)
One_yr.trend = plogis(0.145 + 0.784*0 + 1.013*0 + 0.014*house.dist)
One_yr.lwr = plogis((0.145-0.06) + 0.784*0 + (1.013-0.14)*0 + (0.014-0.005)*house.dist)
One_yr.upr = plogis((0.145+0.06) + 0.784*0 + (1.013+0.14)*0 + (0.014+0.005)*house.dist)

#Create the data frame
House.ads = data.frame( phi = adult.trend,
                        house.dist,
                        phi.lwr = adult.lwr,
                        phi.upr = adult.upr,
                        age = "Adults")
House.1s = data.frame( phi = One_yr.trend,
                       house.dist,
                       phi.lwr = One_yr.lwr,
                       phi.upr = One_yr.upr,
                       age = "1 Year Olds")
House.2s = data.frame( phi = Two_yr.trend,
                       house.dist,
                       phi.lwr = Two_yr.lwr,
                       phi.upr = Two_yr.upr,
                       age = "2 Year Olds")
House = rbind(House.ads, House.2s, House.1s)
```

Now, i will create the plot.

```
p.House = ggplot(data = House) +
  geom_line(aes(x = house.dist, y = phi, colour = age)) +
  geom_ribbon(aes(x = house.dist, ymin = phi.lwr, ymax = phi.upr,
                  group = age), alpha = 0.1) +
  labs(x = "Distance to Nearest House with Pets (m)", y = "Survival Probability") +
  theme_classic()
p.House
```



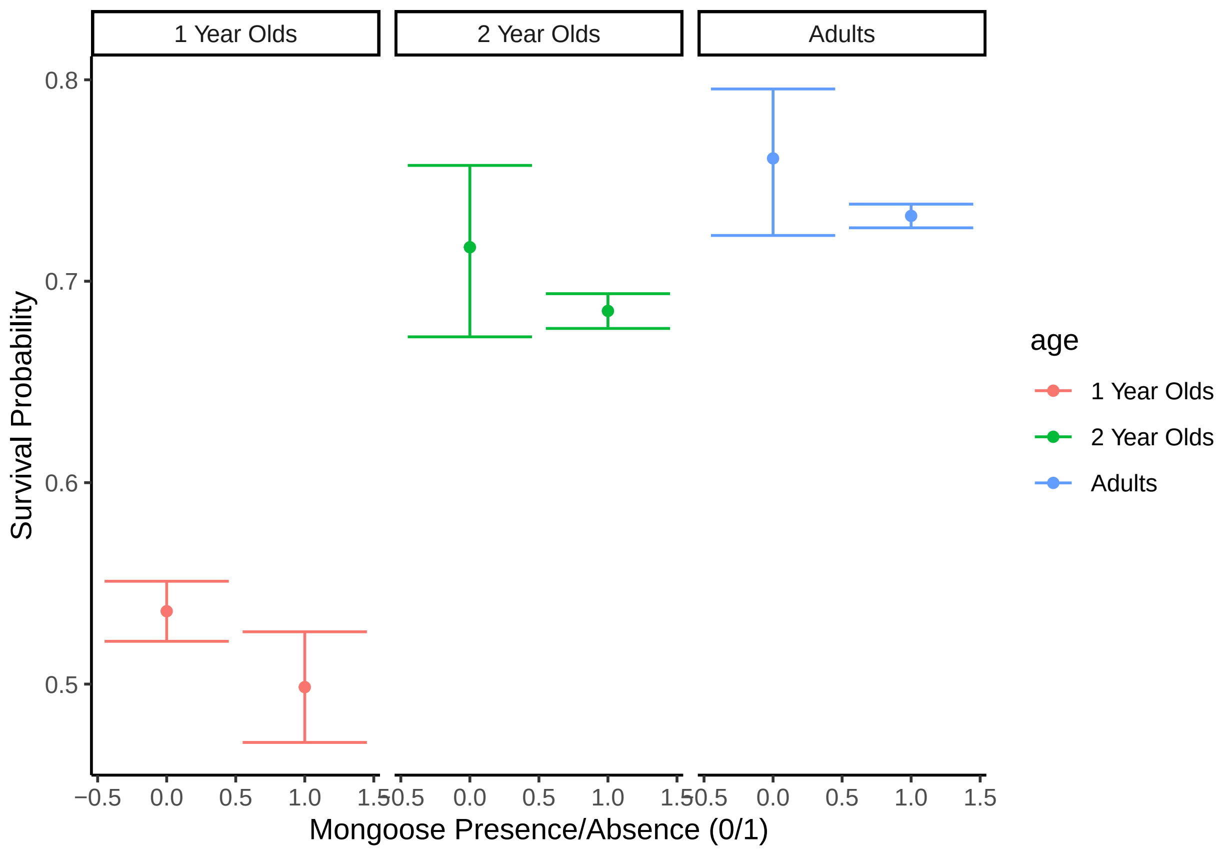
Now I will do the same for the effect of mongooose presence on age specific survival.

```
#Create the predictor
mongoose = c(0,1)

#Simulate responses
adult.trend2 = plogis(0.145 + 0.784*0 + 1.013*1 - 0.151*mongoose)
adult.lwr2 = plogis((0.145-0.06) + 0.784*0 + (1.013-0.14)*1 - (0.151-0.17)*mongoose)
adult.upr2 = plogis((0.145+0.06) + 0.784*0 + (1.013+0.14)*1 - (0.151+0.17)*mongoose)
Two_yr.trend2 = plogis(0.145 + 0.784*1 + 1.013*0 - 0.151*mongoose)
Two_yr.lwr2 = plogis((0.145-0.06) + (0.784-0.15)*1 + (1.013-0.14)*0 - (0.151-0.17)*mongoose)
Two_yr.upr2 = plogis((0.145+0.06) + (0.784+0.15)*1 + (1.013-0.14)*0 - (0.151+0.17)*mongoose)
One_yr.trend2 = plogis(0.145 + 0.784*0 + 1.013*0 - 0.151*mongoose)
One_yr.lwr2 = plogis((0.145-0.06) + 0.784*0 + (1.013-0.14)*0 - (0.151-0.17)*mongoose)
One_yr.upr2 = plogis((0.145+0.06) + 0.784*0 + (1.013+0.14)*0 - (0.151+0.17)*mongoose)

#Create the data frame
Mongoose.ads = data.frame( phi = adult.trend2,
                        mongoose,
                        phi.lwr = adult.lwr2,
                        phi.upr = adult.upr2,
                        age = "Adults")
Mongoose.1s = data.frame( phi = One_yr.trend2,
                       mongoose,
                       phi.lwr = One_yr.lwr2,
                       phi.upr = One_yr.upr2,
```

```
                       age = "1 Year Olds")
Mongoose.2s = data.frame( phi = Two_yr.trend2,
                       mongoose,
                       phi.lwr = Two_yr.lwr2,
                       phi.upr = Two_yr.upr2,
                       age = "2 Year Olds")
Mongoose = rbind(Mongoose.ads, Mongoose.2s, Mongoose.1s)
```

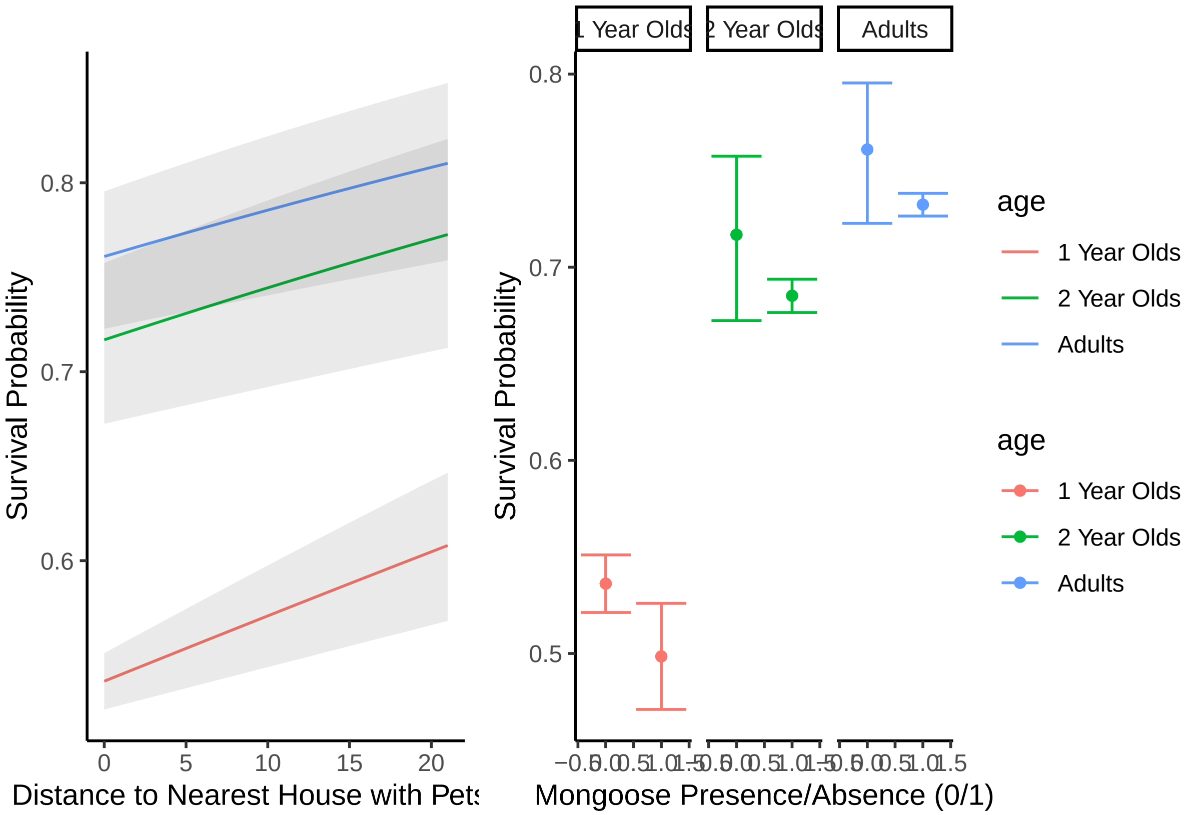
Now for the plot.

```
p.Mongoose = ggplot(data = Mongoose) +
  geom_point(aes(x = mongoose, y = phi, colour = age)) +
  geom_errorbar(aes(x = mongoose, y = phi, colour = age,
                    ymin = phi.lwr, ymax = phi.upr)) +
  facet_grid(~ age) +
  labs(x = "Mongoose Presence/Absence (0/1)", y = "Survival Probability") +
  theme_classic()
p.Mongoose
```



Finally, I will combine the plots into one panel.

```
p.House + p.Mongoose + plot_layout(guides = "collect")
```




```
#ggsave("Estimating Survival.png")
```

Finally, I will create a table of model coefficents to put in the appendix of my management plan report.

```
# Create the data
tab.surv = data.frame(Parameter = c("Intercept", "2yr-Olds", "Adults", "House", "Mongoose", "p"),
                      Estimate = round(plogis(c(0.14, 0.78, 1.01, 0.01, -0.15, 0.41)), digits = 2),
                      SE = round(plogis(c(0.06, 0.15, 0.14, 0.01, 0.17, 0.06)), digits = 2),
                      lwr = round(plogis(c(0.03, 0.49, 0.74, 0.01, -0.48, 0.30)), digits = 2),
                      upr = round(plogis(c(0.26, 1.08, 1.28, 0.02, 0.18, 0.52)), digits = 2))
# Create the table
#kable(tab.surv, row.names = FALSE, format = 'html') %>%
#  kable_styling("striped") %>%
#  save_kable("Survival Model.png")
```

This concludes the "Estimating Survival Figures" section.

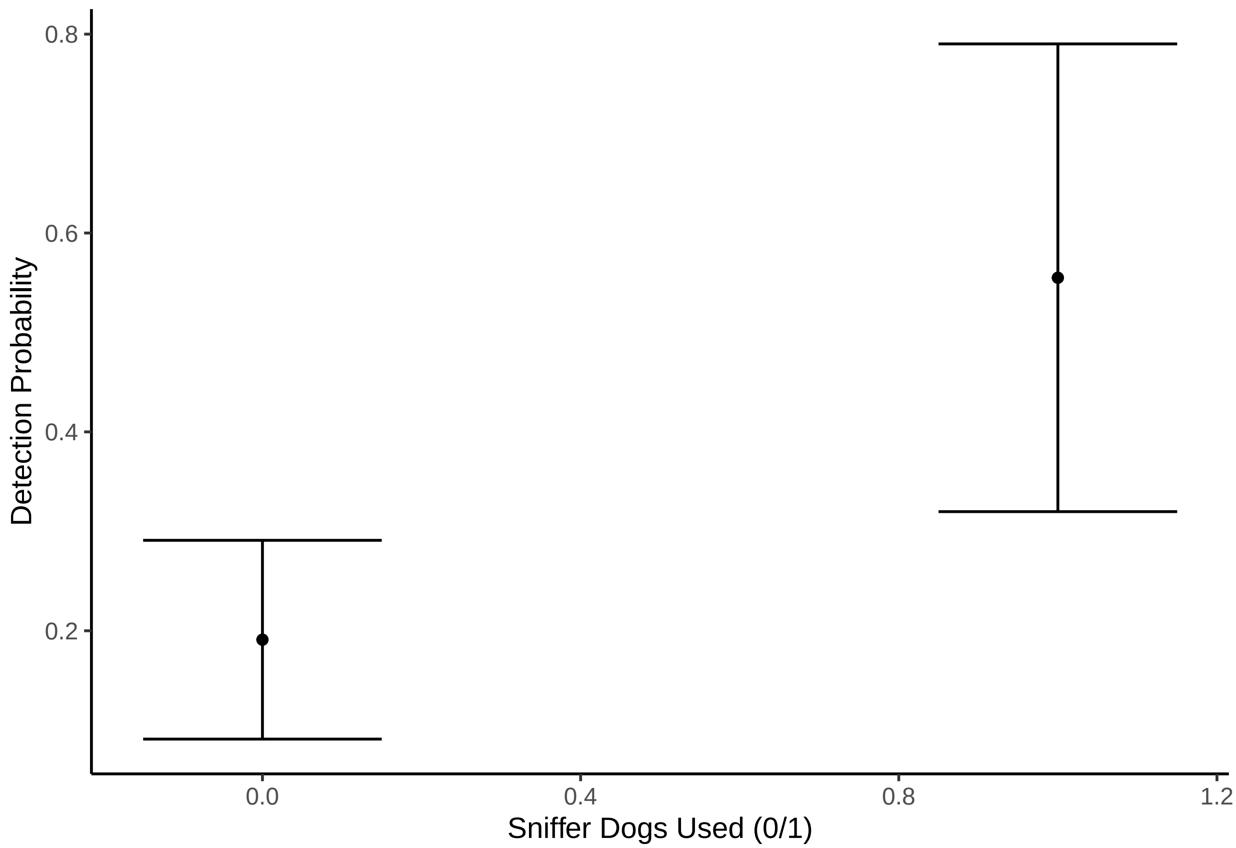
Mongoose Occupancy Figures

In this section I will be plotting the outputs of the occupancy models I ran, but also the predicted occupancy over time. I will then create model coefficent tables for the occupancy models.

First, I will create the plot for the detection propability model, here there is only one predictor **Sniffer_Dogs** and was a factor representing whether they were used (1), or not (0) for detecting mongoose. Beta parameter estimates are given below:

- Intercept: 0.191 +/- 0.051
- Sniffer_Dogs: 0.364 +/- 0.069

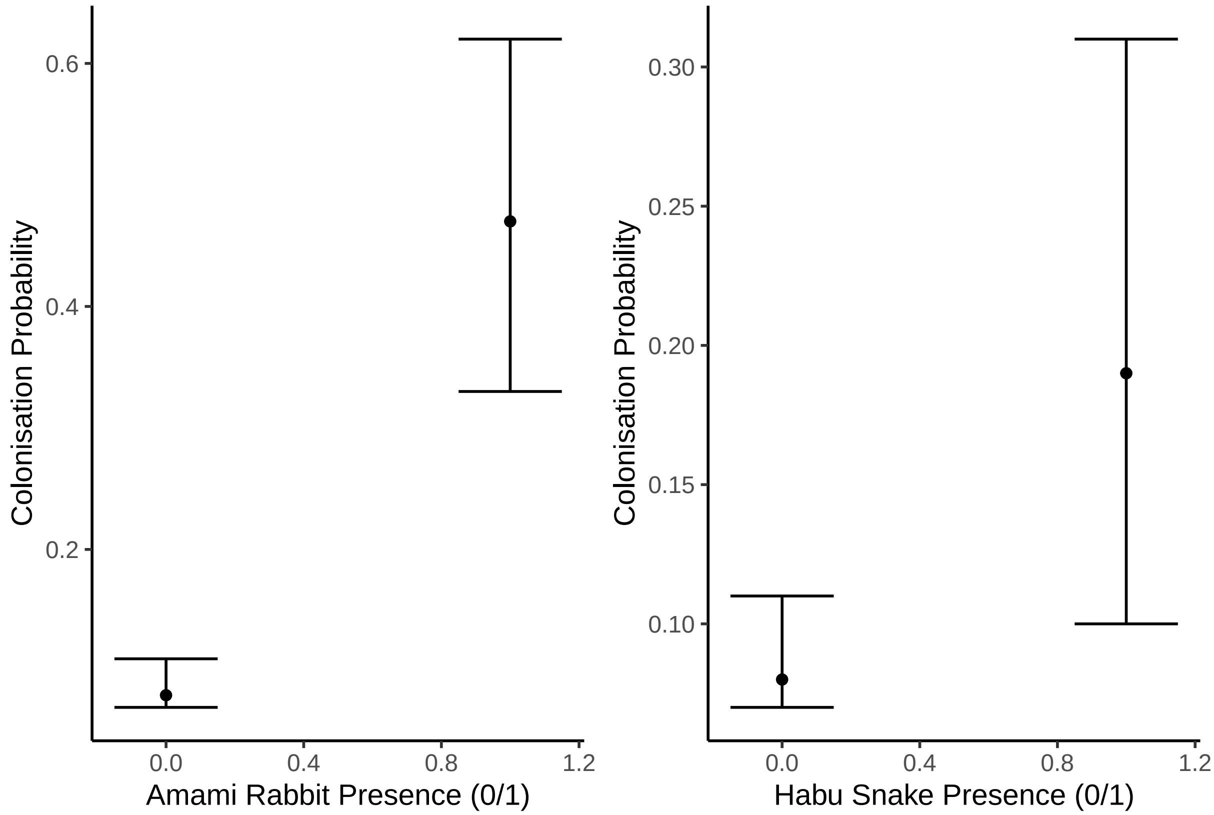
```
# Create the data
Sniffer_Dogs = c(0,1)
P.trend = 0.191 + 0.364*Sniffer_Dogs
P.lwr = (0.191 - 1.96*0.051) + (0.364 - 1.96*0.069)*Sniffer_Dogs
P.upr = (0.191 + 1.96*0.051) + (0.364 + 1.96*0.069)*Sniffer_Dogs
Detect_prob = data.frame(P.trend, Sniffer_Dogs, P.lwr, P.upr)

# Create the plot
p.Prob = ggplot(data = Detect_prob) +
  geom_point(aes(x = Sniffer_Dogs, y = P.trend)) +
  geom_errorbar(aes(x = Sniffer_Dogs, ymin = P.lwr, ymax = P.upr), width = 0.3) +
  labs(x = "Sniffer Dogs Used (0/1)", y = "Detection Probability") +
  theme_classic()
p.Prob
```



The plot looks as it should. I will now move on to plotting the predictions of the local colonisation probability model where we have two predictors **Rabbits** and **Snakes**. They are both factors and relate to whether the species was present (1) or absent (0). Beta parameter estimates are given below:

- Intercept: -2.386 +/- 0.133
- Rabbits: 2.270 +/- 0.178
- Snakes: 0.904 +/- 0.212

```r
# Create the data
Rabbits = c(0,1)
Snakes = c(0,1)

Col.trend1 = -2.386 + 2.270*Rabbits
Col.lwr1 = (-2.386 - 1.96*0.133) + (2.270 - 1.96*0.178)*Rabbits
Col.upr1 = (-2.386 + 1.96*0.133) + (2.270 + 1.96*0.178)*Rabbits

Col.trend2 = -2.386 + 0.904*Snakes
Col.lwr2 = (-2.386 - 1.96*0.133) + (0.904 - 1.96*0.212)*Snakes
Col.upr2 = (-2.386 + 1.96*0.133) + (0.904 + 1.96*0.212)*Snakes

Col_prob1 = data.frame(Col.trend = round(plogis(Col.trend1), digits = 2),
                       Col.lwr = round(plogis(Col.lwr1), digits = 2),
                       Col.upr = round(plogis(Col.upr1), digits = 2),
                       Rabbits = Rabbits)
Col_prob2 = data.frame(Col.trend = round(plogis(Col.trend2), digits = 2),
                       Col.lwr = round(plogis(Col.lwr2), digits = 2),
                       Col.upr = round(plogis(Col.upr2), digits = 2),
                       Snakes = Snakes)

#Create individual plots
p.Col1 = ggplot(data = Col_prob1) +
  geom_point(aes(x = Rabbits, y = Col.trend)) +
  geom_errorbar(aes(x = Rabbits, ymin = Col.lwr, ymax = Col.upr), width = 0.3) +
  labs(x = "Amami Rabbit Presence (0/1)", y = "Colonisation Probability") +
  theme_classic()
p.Col2 = ggplot(data = Col_prob2) +
  geom_point(aes(x = Snakes, y = Col.trend)) +
  geom_errorbar(aes(x = Snakes, ymin = Col.lwr, ymax = Col.upr), width = 0.3) +
  labs(x = "Habu Snake Presence (0/1)", y = "Colonisation Probability") +
  theme_classic()

# Combine the plots
p.Col = p.Col1 + p.Col2
p.Col
```



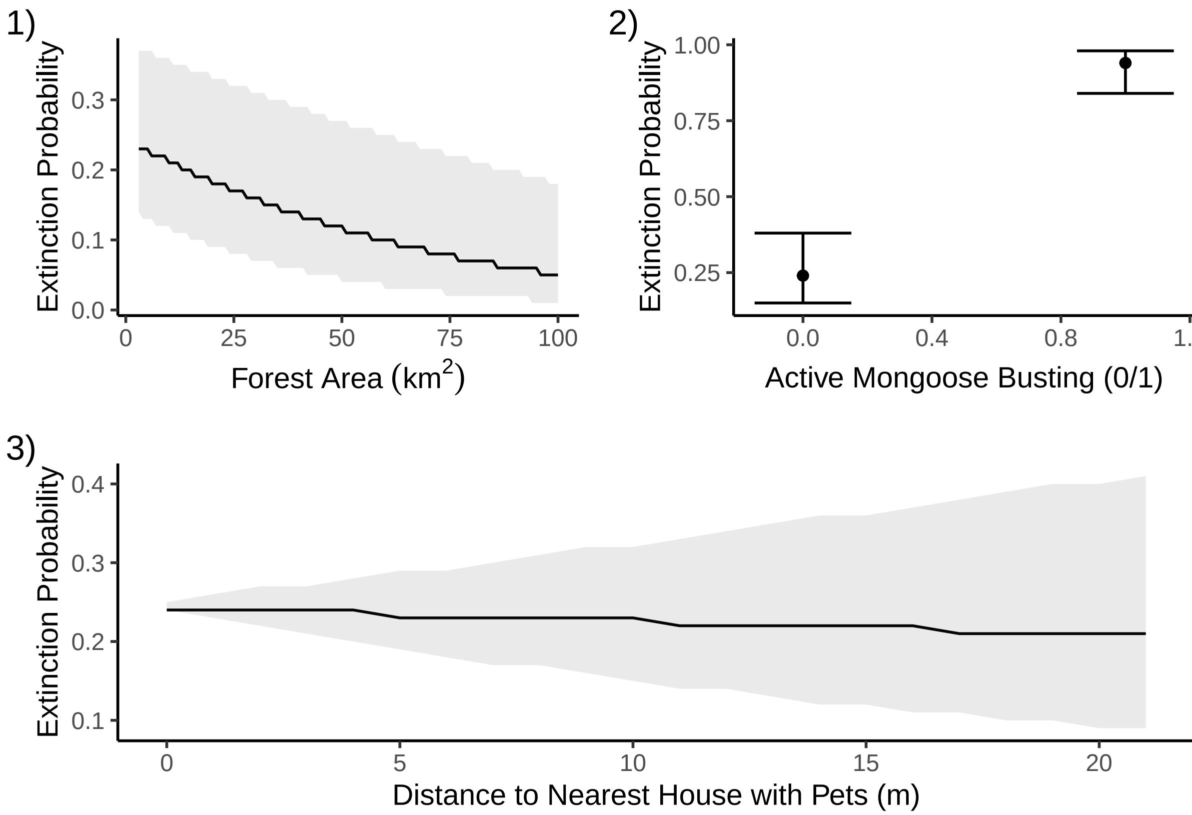

```
#ggsave("Colonisation Probability Model.png")
```

The nested plot looks as it should so I will now move on to plotting the predictions from the extinction probability model. In this model we have 3 predictors, **Forest_Area**, **Busters** and **House** where **Forest_Area** is the total forest area of the sampling site, **Busters** is whether a mongoose busting operation was active (1) or not (0) and **Home** is the distance to the nearest house with pets at the study site. The beta parameter estimates are given below:

- Intercept: -1.133 +/- 0.327
- Forest_Area: -0.018 +/- 0.004
- Busters: 3.924 +/- 0.255
- House: -0.010 +/- 0.023

```
# Create the data
Forest_Area = seq(from = 3, to = 100)
Busters = c(0,1)
Home = seq(from = 0, to = 21)

Ext.trend1 = -1.133 -0.018*Forest_Area
Ext.lwr1 = (-1.133 - 1.96*0.327) + (-0.018 - 1.96*0.004)*Forest_Area
Ext.upr1 = (-1.133 + 1.96*0.327) + (-0.018 + 1.96*0.004)*Forest_Area

Ext.trend2 = -1.133 + 3.924*Busters
Ext.lwr2 = (-1.133 - 1.96*0.327) + (3.924 - 1.96*0.255)*Busters
Ext.upr2 = (-1.133 + 1.96*0.327) + (3.924 + 1.96*0.255)*Busters
```

```
Ext.trend3 = -1.133 - 0.010*Home
Ext.lwr3 = (-1.133 - 1.96*0.023) + (-0.010 - 1.96*0.023)*Home
Ext.upr3 = (-1.133 + 1.96*0.023) + (-0.010 + 1.96*0.023)*Home

Ext_prob1 = data.frame(Ext.trend = round(plogis(Ext.trend1), digits = 2),
                       Ext.lwr = round(plogis(Ext.lwr1), digits = 2),
                       Ext.upr = round(plogis(Ext.upr1), digits = 2),
                       Forest_Area = Forest_Area)
Ext_prob2 = data.frame(Ext.trend = round(plogis(Ext.trend2), digits = 2),
                       Ext.lwr = round(plogis(Ext.lwr2), digits = 2),
                       Ext.upr = round(plogis(Ext.upr2), digits = 2),
                       Busters = Busters)
Ext_prob3 = data.frame(Ext.trend = round(plogis(Ext.trend3), digits = 2),
                       Ext.lwr = round(plogis(Ext.lwr3), digits = 2),
                       Ext.upr = round(plogis(Ext.upr3), digits = 2),
                       Home = Home)

# Create the individual plots
p.Ext1 = ggplot(data = Ext_prob1) +
  geom_line(aes(x = Forest_Area, y = Ext.trend)) +
  geom_ribbon(aes(x = Forest_Area, ymin = Ext.lwr, ymax = Ext.upr), alpha = 0.1) +
  labs(x = bquote("Forest Area" ~ (km^2)), y = "Extinction Probability") +
  theme_classic()
p.Ext2 = ggplot(data = Ext_prob2) +
  geom_point(aes(x = Busters, y = Ext.trend)) +
  geom_errorbar(aes(x = Busters, ymin = Ext.lwr, ymax = Ext.upr), width = 0.3) +
  labs(x = "Active Mongoose Busting (0/1)", y = "Extinction Probability") +
  theme_classic()
p.Ext3 = ggplot(data = Ext_prob3) +
  geom_line(aes(x = Home, y = Ext.trend)) +
  geom_ribbon(aes(x = Home, ymin = Ext.lwr, ymax = Ext.upr), alpha = 0.1) +
  labs(x = "Distance to Nearest House with Pets (m)", y = "Extinction Probability") +
  theme_classic()

# Create the combined plot
p.Ext = (p.Ext1 | p.Ext2)/p.Ext3
p.Ext + plot_annotation(tag_levels = "1", tag_suffix = ")")
```




```
#ggsave("Extinction Probability Model.png")
```

The plot looks as it should I will now move on to plotting how occupancy **psi** has changed over time. This requires all 3 of the previous models plus the initial occupancy. The "TokunoshimaOcc.txt" file contains occupancy data on Tokuno-Shima on 5 sites from 1987 to 2021. I will first read in the data then make a **Year** variable.

```
Occupancy = read.table("E:/Wildlife Conservation and Management/Z04541/TokunoshimaOcc.txt",
                       header = TRUE, sep = "", stringsAsFactors = TRUE)
Occupancy$Year = factor(rep(1987:2020, each = 5))
summary(Occupancy)
```

```
##             Site           Cpsi              Year
##  Tokunoshima:170   Min.   :0.0044   1987   :  5
##                    1st Qu.:0.0080   1988   :  5
##                    Median :0.1167   1989   :  5
##                    Mean   :0.4472   1990   :  5
##                    3rd Qu.:1.0000   1991   :  5
##                    Max.   :1.0000   1992   :  5
##                                     (Other):140
```

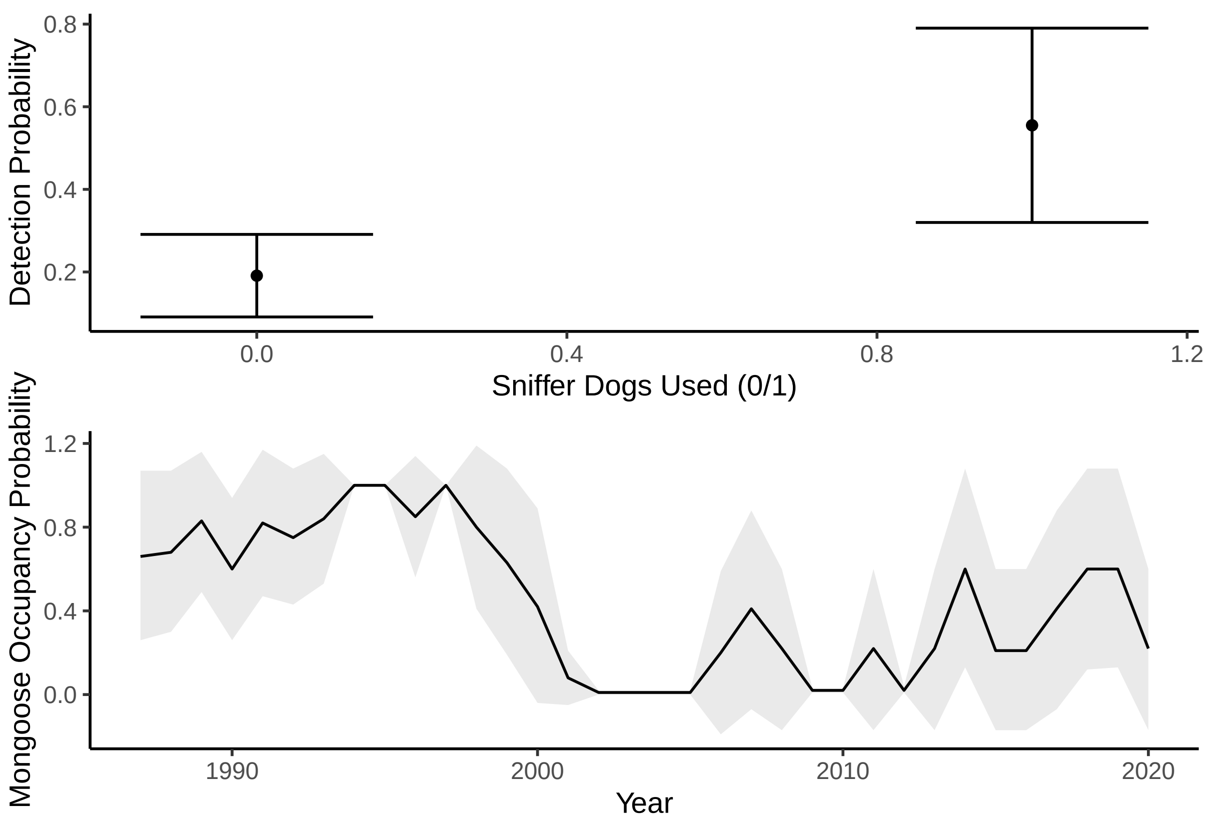
```
str(Occupancy)
```

```
## 'data.frame':    170 obs. of  3 variables:
```

```
##  $ Site: Factor w/ 1 level "Tokunoshima": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Cpsi: num  1 1 1 0.157 0.157 ...
##  $ Year: Factor w/ 34 levels "1987","1988",..: 1 1 1 1 1 2 2 2 2 2 ...
```

Now I will plot the occupancy data for Tokuno-Shima

```
# Create the data
Year = 1987:2020
Occ.trend = tapply(Occupancy$Cpsi, Occupancy$Year, FUN = mean)
Occ.se = tapply(Occupancy$Cpsi, Occupancy$Year,
                FUN = function(x) sd(x)/sqrt(length(x)))
Occ = data.frame(psi = round(Occ.trend, digits = 2),
                 psi.lwr = round(Occ.trend-1.96*Occ.se, digits = 2),
                 psi.upr = round(Occ.trend+1.96*Occ.se, digits = 2),
                 Year = Year)

#Create the plot
p.Occ = ggplot(data = Occ) +
  geom_line(aes(x = Year, y = psi)) +
  geom_ribbon(aes(x = Year, ymin = psi.lwr, ymax = psi.upr), alpha = 0.1) +
  labs(x = "Year", y = "Mongoose Occupancy Probability") +
  theme_classic()

#Create a combined plot with detection probability
p.Occ_Prob = p.Prob / p.Occ
p.Occ_Prob
```



```r
#ggsave("Conditional Occupancy Model.png")
```

Finally, I will create a table of back-transformed parameter estimates. Beta estimates are given below:

- psi0: 0.697 +/- 0.258
- gamma0: 2.386 +/- 0.133
- Rabbits: 2.270 +/- 0.178
- Snakes: 0.904 +/- 0.212
- epsilon0: -1.133 +/- 0.327
- Forest_Area: -0.018 +/- 0.004
- Busters: 3.924 +/- 0.255
- House: -0.010 +/- 0.023
- p0: 0.191 +/- 0.051
- Sniffer_Dogs: 0.364 +/- 0.069

```r
 # Create the data
tab.Occ = data.frame(Model = c("psi", rep("gam", times = 3), rep("eps", times = 4), rep("p", times = 2)
                     Variable = c(rep("Int", times = 2), "Rabbits", "Snakes", "Int", "Forest_Area",
                                  "Busters", "House", "Int", "Sniffer_Dogs"),
                     Estimate = round(plogis(c(0.697, 2.386, 2.270, 0.904, -1.133, -0.018, 3.924,
                                               -0.010, 0.191, 0.364)), digits = 3),
                     SE = round(plogis(c(0.258, 0.133, 0.178, 0.212, 0.327, 0.004, 0.255, 0.023,
                                         0.051, 0.069)), digits = 2))
tab.Occ$lwr = tab.Occ$Estimate - 1.96*tab.Occ$SE
tab.Occ$upr = tab.Occ$Estimate + 1.96*tab.Occ$SE

# Create the table
#kable(tab.Occ) %>%
#  kable_styling("striped") %>%
#  save_kable("Occupancy Model.png")
```

I also want to create a table detailing what variables were used for what occupancy sub-model.

```r
#Create the data
tab = data.frame(Initial.Occupancy = c("Constant", "", ""),
                 Detection.Prob = c("Sniffer Dogs", "", ""),
                 Colonisation.Prob = c("Rabits", "Snakes", ""),
                 Extinction.Prob = c("Forest Area", "Busters", "House"))
colnames(tab) = c("Initial Occupancy", "Detection Probability", "Colonisation Probability",
                  "Extinction Probability")
#Create the table
#kable(tab) %>%
#  kable_styling("striped") %>%
#  save_kable("Occupancy Model Variables.png")
```

This concludes the "Estimating Mongoose Occupancy Figures" section.

Matrix Model Figures

Finally, I will make a table of the action, corresponding lambda, and 95% confidence intervals.

```r
# Create the data
#lambda.current = eigs(A.avg, what = "lambda")
#lambda.action1 = eigs(A1.avg, what = "lambda")
#lambda.action2 = eigs(A2.avg, what = "lambda")
#lambda.current.lwr = eigs(A.lwr, what = "lambda")
#lambda.action1.lwr = eigs(A1.lwr, what = "lambda")
#lambda.action2.lwr = eigs(A2.lwr, what = "lambda")
#lambda.current.upr = eigs(A.upr, what = "lambda")
#lambda.action1.upr = eigs(A1.upr, what = "lambda")
#lambda.action2.upr = eigs(A2.upr, what = "lambda")

#tab.matrix = data.frame(Action = c("None", "Action1", "Action2"),
#                         Lambda = round(c(lambda.current, lambda.action1, lambda.action2), digits = 2),
#                         lwr = round(c(lambda.current.lwr, lambda.action1.lwr, lambda.action2.lwr), dig
#                         upr = round(c(lambda.current.upr, lambda.action1.upr, lambda.action2.upr), dig

# Create the table
#kable(tab.matrix) %>%
#  kable_styling("striped") %>%
#  save_kable("Lambda.png")
```

This concludes the "Matrix Modelling Figures" section and the document.

```r
rm(list = ls())
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] popdemo_1.3-1    kableExtra_1.3.4 knitr_1.37       patchwork_1.1.1
## [5] ggplot2_3.3.5
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.29         purrr_0.3.4       lattice_0.20-45
##  [5] colorspace_2.0-2  vctrs_0.3.8       generics_0.1.1    expm_0.999-6
##  [9] htmltools_0.5.2   viridisLite_0.4.0 yaml_2.2.1        utf8_1.2.2
## [13] rlang_0.4.11      pillar_1.6.4      glue_1.6.1        withr_2.4.3
## [17] DBI_1.1.2         lifecycle_1.0.1   stringr_1.4.0     munsell_0.5.0
## [21] gtable_0.3.0      rvest_1.0.2       evaluate_0.14     labeling_0.4.2
```

```
## [25] fastmap_1.1.0     fansi_1.0.2       highr_0.9         scales_1.1.1
## [29] webshot_0.5.2     farver_2.1.0      systemfonts_1.0.3 digest_0.6.28
## [33] stringi_1.7.6     dplyr_1.0.7       grid_4.1.1        tools_4.1.1
## [37] magrittr_2.0.1    tibble_3.1.6      crayon_1.4.2      pkgconfig_2.0.3
## [41] ellipsis_0.3.2    Matrix_1.3-4      xml2_1.3.3        assertthat_0.2.1
## [45] rmarkdown_2.11    svglite_2.0.0     httr_1.4.2        rstudioapi_0.13
## [49] R6_2.5.1          compiler_4.1.1
```