Where to See Poecile: Understanding Swiss Willow Tit Occupancy

Misha Tseitlin

24 March 2024

Project 3: Occupancy Modelling

School of Mathematics and Statistics

in partial fulfilment of the requirements for
MT5751: Estimating Animal Abundance and Biodiversity

Executive Summary

Willow tits (*Poecile montanus*) etc etc.

Introduction

Willow tits are a well-studied resident Eurasian breeding bird species seeing substantial population declines in some areas like Britain and Finland.[1,2] Divided into several subspecies, Swiss alpine (*montanus*) and willow (*rhenanus* and *salicarius*) tits have experienced fewer pressures and maintained their population better than their international compatriots. Even so, alpine tits have proven especially resilient in contrast to more vulnerable willow tits, possibly caused by interspecific and intraspecific competition, changing vegetation onset, and rising temperatures.[3,4] Some environmental drivers are particularly clear: whether via its species name *montanus* or its French common name (la mésange boréale), we have long known about the bird's dependence on mountainous, forested habitats. Averse to competition, the species shows strong density-dependence and preference for forest edge habitats and thus particularly responds to land use changes.[5] Using occupancy modelling, we unpack specific drivers of willow tit occupancy and detectability, provide habitat management recommendations, and predict national-level as well as some quadrant-specific occurrence.

Methods

The Swiss programme MHB (Monitoring Häufige Brutvögel) annually surveys 267 squares nationwide to understand bird populations: we use a pre-processed subset of 237 quadrants with detected willow tits across three survey occasions (two for high-altitude quadrants) and various possible factors including elevation, forest cover, survey duration, survey route length, and survey day.[6] Data quality may be imperfect—MHB documentation lists surveys as running from mid-April to mid-July, but our "day" measure allegedly starts at the start of the year which suggests surveying outside the breeding season from January to April.[3] (Table 1) For analysis, we z-score elevation, convert forest cover to a proportion, and assume that survey day measures the date from 1 April rather than 1 January. As we lack spatial quadrant locations, we forgo initial exploration of detected willow tit distribution and skip directly to occupancy.

Hierarchical occupancy modelling provides the best method of separating out tit detectability p from occupancy ψ; we need to determine which of our observations stemmed from observer patterns versus the true underlying state process. The resulting two-tier model looks across i quadrants and j survey occasions to estimate p and ψ using a logistic relationship $ln(\frac{x}{1-x})$ for probabilities:

$$\begin{aligned} logit(\psi_i) = \beta_0 + \beta_1 Elevation_i + \beta_2 Elevation_i^2 + \beta_3 Forest_i + \beta_4 Forest_i^2 \\ + \beta_5 Forest_i Elevation_i^2 + \beta_6 Forest_i^2 Elevation_i^2 \end{aligned}$$

$$logit(p_{ij}) = \alpha_0 + \alpha_1 Day_{ij} + \alpha_2 Duration_{ij} + \alpha_3 Forest_{ij} + \alpha_4 Forest_{ij} Day_{ij}$$

Detection and state functions, based on only four covariates, demonstrate the complexity of fitting highly non-linear relationships using this linear (GLM) framework fitted using `unmarked` in `R`.[7,8] To account for clear non-linearities in forest cover and elevation, we manually and automatically compared all models containing potentially relevant covariates.[9,10] Overall, adding non-linearity substantially improved model fit measured by AIC. (Table 2) Finally, we tackled model suitability using Pearson's chi-squared via bootstrap estimation: MacKenzie-Bailey goodness-of-fit tests measured the chance of seeing our data given the model via a p-value and resulting overdispersion $\hat{c}$.[11] To correct for imperfect fit, we inflated prediction errors by $\hat{c}$.[12]

Results

Overall, model fit is sub-par but passable yet suggests unmodelled heterogeneity—a violation of occupancy modelling assumptions. On one hand, we understand including covariates like forest*day interactions for detection (i.e., foliage thickness impedes observer visibility) and occupancy impacts from squared relationships for elevation (i.e., optimal mountain habitat below the tree line) and forests (i.e., higher interspecific competition and less deadwood in aggressively-forested areas). The joint relationship between both forest and elevation further approximates various components of ecosystem suitability like climate and diet.[13] Nonetheless, we also avoid overfitting despite AIC suggesting elevation-dependent detection. Elevation indeed affects cloud cover, heat, and humidity and therefore detection (by proxy) but imperfectly captures their effects.[14,15] Instead, identifiability issues inhibit parameter-specific inference for only minor predictive improvements.[16] Thus, our model balances predictive and inferential analysis goals by imperfectly match the data but better generalising beyond anomalous 2014 heat and vegetation. (Figure 1)

Given model non-linearity for parameter estimates, we examine partial and predictive effects for inference and management: spatial predictions do not differentiate between the southern Swiss Alps' alpine tit and the willow tit in northwestern Jura and central pre-Alps and likely represent the confusion between complex environmental parameters—intermediate elevation and high forest in the former, and intermediate vegetation and high elevation in the latter. (Figure 2) Overall, we struggle to conclude anything beyond tit avoidance of low-lying river valleys and the Central Plateau; estimated occupancy peaks above 50% only around Jura Vadois in the west, upslope of Rhone and Ticino river valleys in the south, and above Lake Lucerne's surrounding Pre-Alps. (Figure 3)

GR (Geographic Region)

Table 1: Data Summary Statistics Split by First Occasion Detection

Variable	**Overall**, N = 237	**Undetected**, N = 170	**Detected**, N = 67
Occasion 2 Detected, # (%)	61 (26%)	9 (5.4%)	52 (78%)
Occasion 3 Detected, # (%)	54 (28%)	7 (5.2%)	47 (80%)
Occasion 1 Duration (min)			
Mean (SD)	229 (62)	214 (57)	266 (58)
[Min, Max]	[95, 390]	[95, 390]	[120, 390]
Occasion 2 Duration (min)			
Mean (SD)	232 (63)	217 (59)	270 (57)
[Min, Max]	[90, 391]	[90, 391]	[120, 375]
Occasion 3 Duration (min)			
Mean (SD)	232 (66)	217 (64)	265 (56)
[Min, Max]	[85, 406]	[85, 406]	[110, 360]
Length (km)			
Mean (SD)	5.10 (1.35)	5.15 (1.46)	4.99 (1.04)
[Min, Max]	[1.20, 9.40]	[1.20, 9.40]	[2.90, 8.50]
Day of Occasion 1			
Mean (SD)	38 (21)	35 (21)	45 (19)
[Min, Max]	[13, 91]	[13, 91]	[13, 87]
Day of Occasion 2			
Mean (SD)	57 (19)	55 (19)	63 (17)
[Min, Max]	[29, 102]	[29, 102]	[37, 97]
Day of Occasion 3			
Mean (SD)	69 (13)	66 (11)	75 (14)
[Min, Max]	[42, 107]	[42, 101]	[47, 107]
Forest Cover (%)			
Mean (SD)	35 (28)	30 (28)	47 (23)
[Min, Max]	[0, 98]	[0, 98]	[2, 95]
Elevation (m)			
Mean (SD)	1,183 (646)	1,047 (665)	1,526 (442)
[Min, Max]	[250, 2,750]	[250, 2,750]	[380, 2,310]

Table 2: Fitted Model Comparisons

Detection	**Occupancy**	**AIC**
p(day + dur + forest + forest*day + elev)	psi(elev + elev^2*(forest + forest^2))	382.4294
p(day + dur + forest + forest*day)	psi(elev + elev^2*(forest + forest^2))	391.1644
p(day + dur + forest + forest^2)	psi(elev + elev^2 + forest + forest^2)	392.1461
p(day + dur + forest + forest*day)	psi(elev)	392.6263
p(day + dur + forest + forest*day)	psi((elev + elev^2)*(forest + forest^2))	394.7457
p(day + dur + forest)	psi(elev + elev^2 + forest + forest^2)	400.9895
p(day + dur + forest)	psi(elev + forest)	405.8298
p(.)	psi(.)	528.9870

Note. Models include survey day, survey duration, forest cover, and elevation as well as their squared and interaction terms. Interaction terms imply including base terms (e.g., elev*forest includes elev and forest).

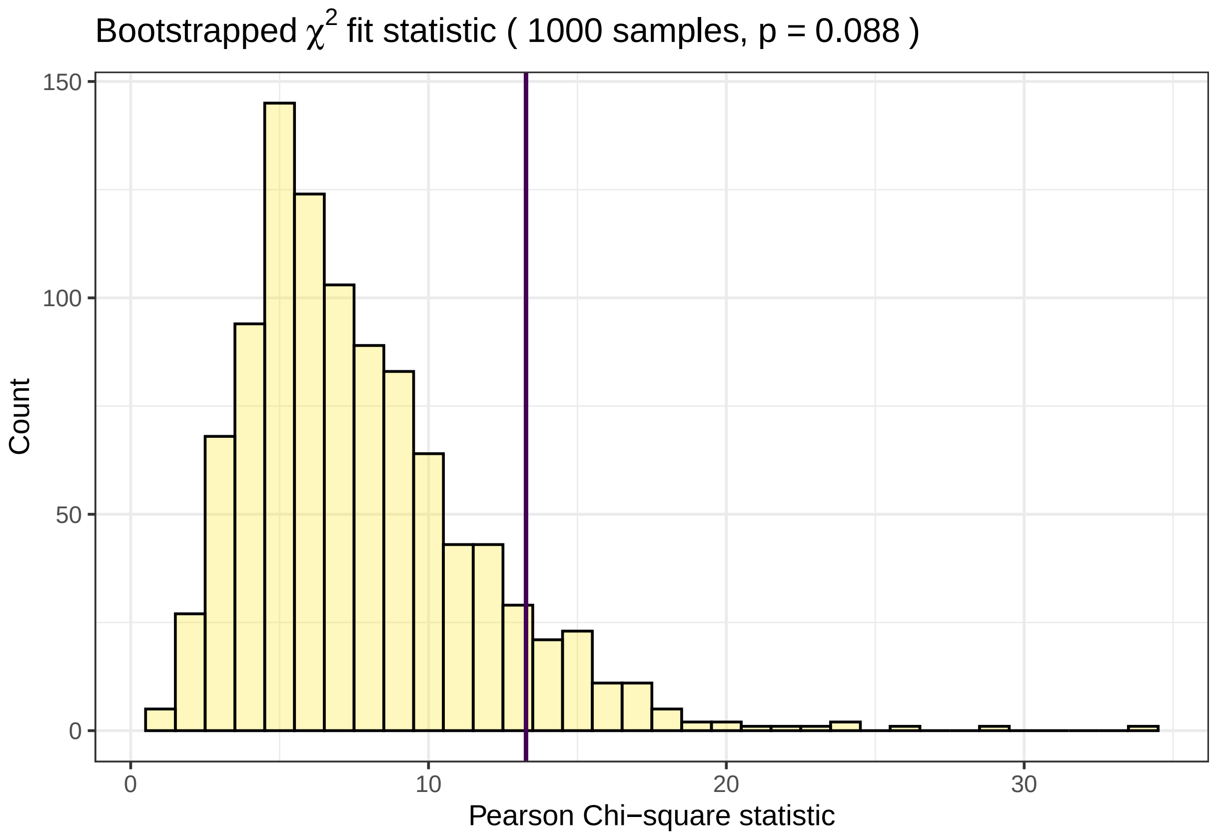

Figure 1: Goodness-of-fit suggests 'good' fit at the 5% level but somewhat atypical model predictions.

Code Appendix

```
library(statsecol)
library(unmarked)
library(tidyverse)
library(MuMIn)
library(AICcmodavg)
library(viridisLite)
library(latex2exp)
# read in data
data(willow)
# prepare the data how I need it
willowNum <- willow %>% mutate(forestsq = forest^2,
                              iLength = 1/length) %>%
  mutate_all(as.numeric) %>%
  rownames_to_column("id")

willowSum <- willowNum %>% mutate(forestP = forest*100,
                                  elevR = 1182.574 + elev*646.333,
                                  y.1 = as.factor(y.1),
                                  y.2 = as.factor(y.2),
                                  y.3 = as.factor(y.3)) %>%
  select(-c("id","elev","elevsq","forest","forestsq","iLength"))
```

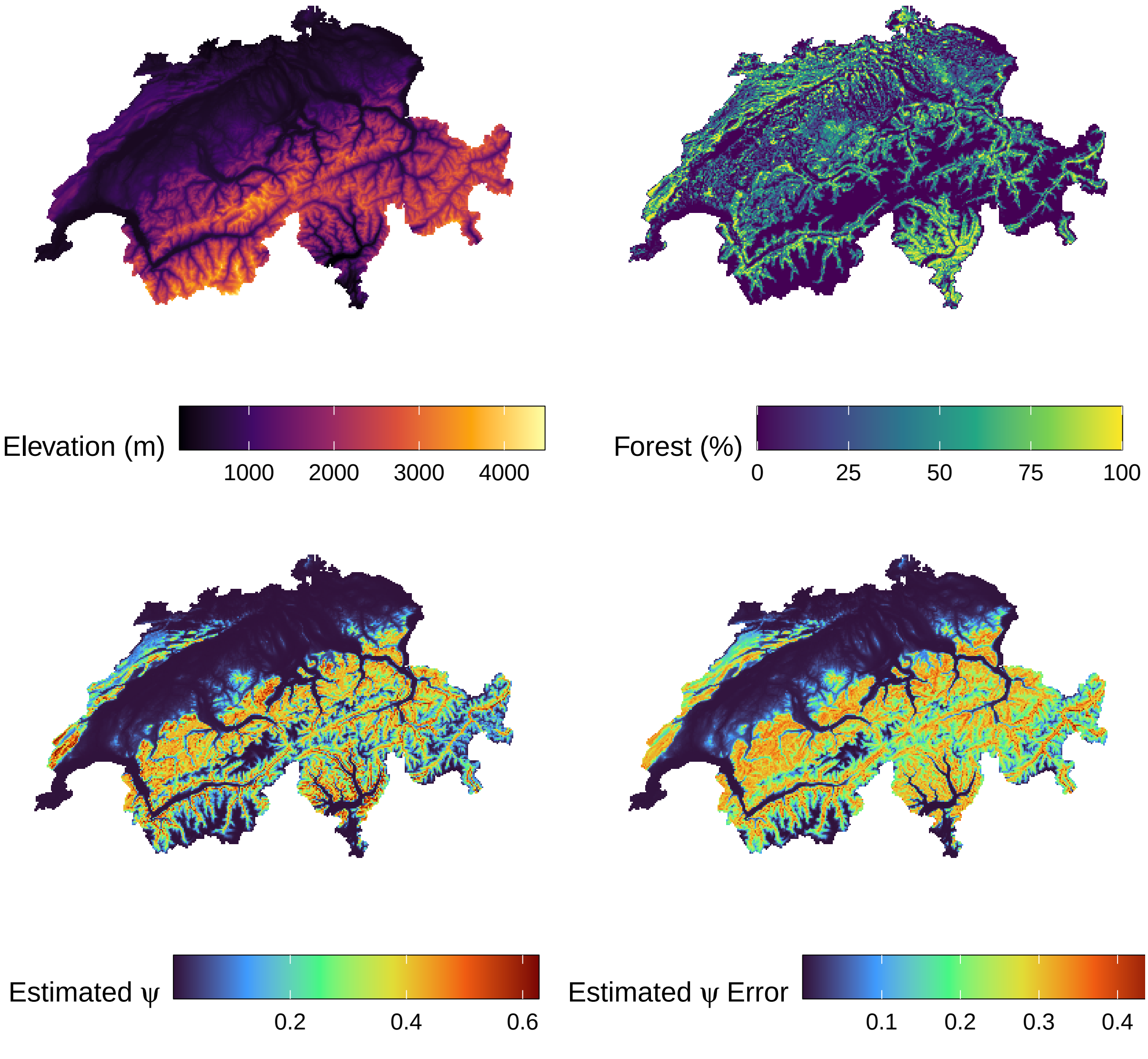

Figure 2: Spatial trends of elevation, forest cover, and estimated occupancy across Switzerland.

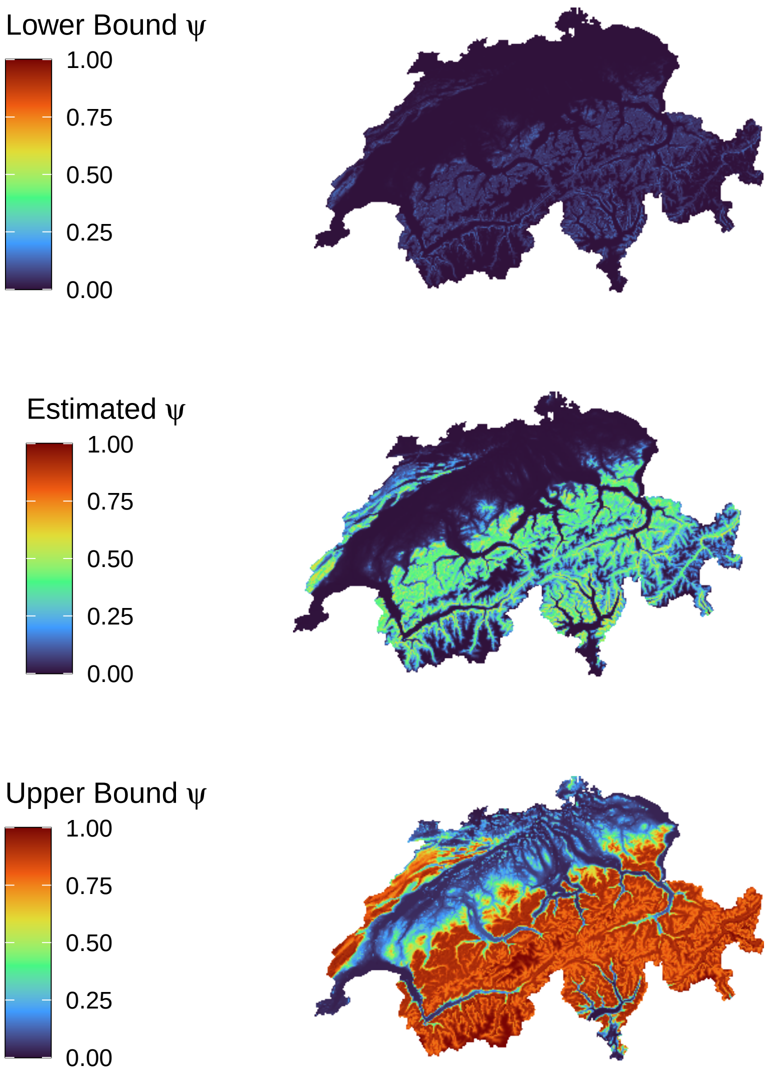

Figure 3: Mean, upper, and lower bound predicted occupancy demonstrating extreme uncertainty and few high-confidence areas.

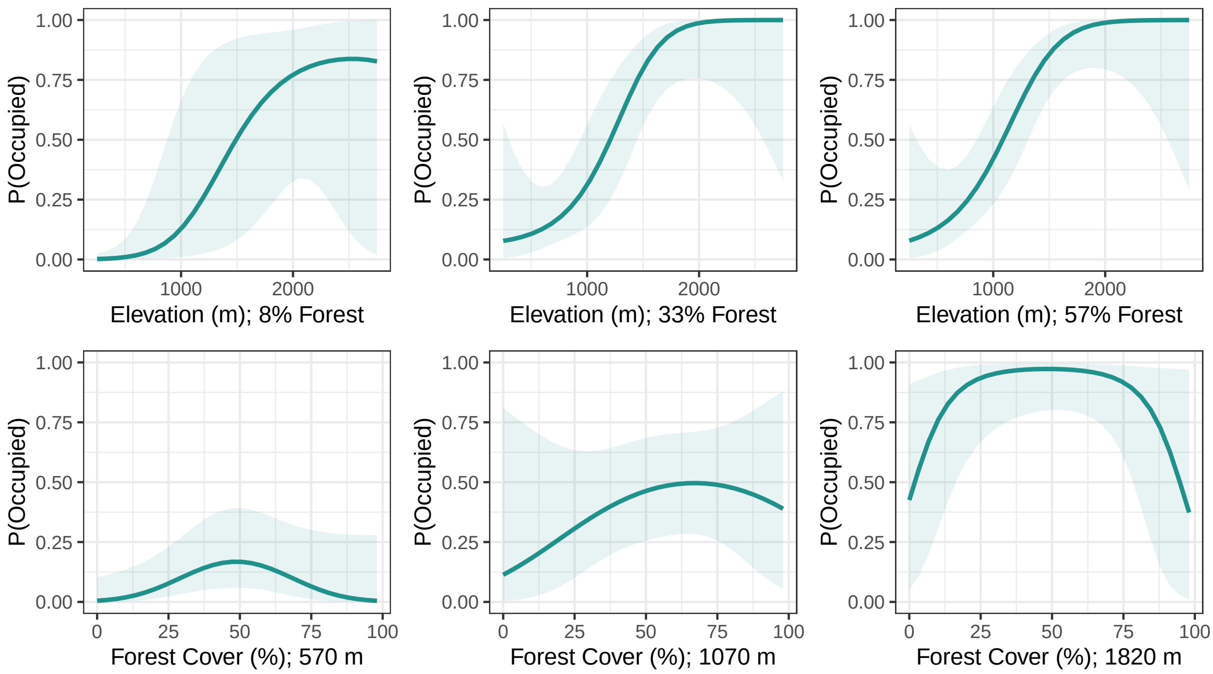

Figure 4: Elevation and forest effects under 25th, median, and 75th percentile values of the non-axis variable. Elevation is essential to occupancy but highly uncertain at extreme values, while forest cover's effects are clearer but with elevation-dependent magnitudes.

```
# brief data visualisation
hist(willowNum$forest)
hist(willowNum$elev)
hist(willowNum$length)
# no clear relationship here, except that there's no forest at very high-elevation areas
ggplot(willowNum, aes(x = forest, y = elev)) + geom_point()
# no occurrence at lower elevations, higher occurrence at higher elevations
# no real shifts between time periods
ggplot(willowNum, aes(x = elev, fill = y.1)) + geom_histogram() + facet_wrap(~y.1) + theme(l
ggplot(willowNum, aes(x = elev, fill = y.2)) + geom_histogram() + facet_wrap(~y.2) + theme(l
ggplot(willowNum, aes(x = elev, fill = y.3)) + geom_histogram() + facet_wrap(~y.3) + theme(l
# complex forest relationship w/ no occurrence at predominantly low forest cover and an inve
# no shifts between time periods again
ggplot(willowNum, aes(x = forest, fill = y.1)) + geom_histogram() + facet_wrap(~y.1) + theme
ggplot(willowNum, aes(x = forest, fill = y.2)) + geom_histogram() + facet_wrap(~y.2) + theme
ggplot(willowNum, aes(x = forest, fill = y.3)) + geom_histogram() + facet_wrap(~y.3) + theme



willowUnm <- unmarkedFrameOccu(
  y = willowNum[,c("y.1","y.2","y.3")],
  # we are interested in breeding bird occupancy for the WHOLE breeding season, so days (wh
  siteCovs = data.frame(elev = willowNum$elev,
                        elev2 = willowNum$elevsq,
                        forest = willowNum$forest,
                        forest2 = willowNum$forestsq,
                        iLength = willowNum$iLength),
```

```r
  obsCovs = list(day = willowNum[,c("day1","day2","day3")],
                 dur = willowNum[,c("dur1","dur2","dur3")],
                 intensity = willowNum[,c("intensity1","intensity2","intensity3")],
                 length = willowNum[,c("length","length","length")],
                 iLength = willowNum[,c("iLength","iLength","iLength")],
                 forest = willowNum[,c("forest","forest","forest")],
                 forest2 = willowNum[,c("forestsq","forestsq","forestsq")],
                 elev = willowNum[,c("elev","elev","elev")],
                 elev2 = willowNum[,c("elevsq","elevsq","elevsq")])
)
summary(willowUnm)

# some more plotting
hist(willowUnm@obsCovs$day)
hist(willowUnm@obsCovs$dur)
hist(willowUnm@obsCovs$intensity)


# null model
m0 <- occu(~1 ~1, data = willowUnm)

# full model
# intensity:length interaction is just dur and thus not included
mfull <- occu(formula = ~day + dur + intensity + length + day*dur + dur*intensity + day*leng
              ~elev + elev2 + forest + forest2 + elev*forest + elev2*forest2 + elev*forest2
              data = willowUnm) #the data object

summary(mfull)

# all three agree on a few things
# elev2 and elev:forest are mutually exclusive
# p(day) is not worth including
# all p interaction terms are useless
mDredgeB <- dredge(mfull, rank = "BIC")
mDredgeA <- dredge(mfull, rank = "AIC")
mDredgeAc <- dredge(mfull, rank = "AICc")

mBIC <- occu(formula = ~dur # p formula
             ~elev + forest + forest^2 + elev*forest, #psi formula
             data = willowUnm) #the data object
summary(mBIC)

# we missed a few variables, mainly forests which could affect visibility by surveyors (eve
mfullAct <- occu(formula = ~day + dur + intensity + length + day*dur + dur*intensity + day*l
              ~elev + elev2 + forest + forest2 + elev*forest + elev2*forest2 + elev*forest2
              data = willowUnm) #the data object

summary(mfullAct)

# all three agree on a few things
# elev2 and elev:forest are mutually exclusive
```

```r
# p(day) is not worth including
# all p interaction terms are useless
mDredgeB2 <- dredge(mfullAct, rank = "BIC")
mDredgeA2 <- dredge(mfullAct, rank = "AIC")
mDredgeAc2 <- dredge(mfullAct, rank = "AICc")

# create some representative models to display
mAlt <- occu(formula = ~day + dur + forest  # p formula
             ~elev + forest, #psi formula
             data = willowUnm) #the data object

mAlt1 <- occu(formula = ~day + dur + forest # p formula
              ~elev + elev2 + forest + forest2, #psi formula
              data = willowUnm) #the data object

mAlt2 <- occu(formula = ~day + dur + forest + forest2 # p formula
              ~elev + elev2 + forest + forest2, #psi formula
              data = willowUnm) #the data object

mOptm_Alt <- occu(formula = ~day + dur + forest + forest*day # p formula
             ~elev, #psi formula
             data = willowUnm) #the data object

# recording visibility/leafing may be a good way to increase separability and avoid forest
mOptm <- occu(formula = ~day + dur + forest + forest*day # p formula
              ~elev + elev2 + forest + forest2 + elev2*forest + elev2*forest2 , #psi formul
              data = willowUnm) #the data object

mAlt3 <- occu(formula = ~day + dur + forest + forest*day # p formula
              ~elev + elev2 + forest + forest2 + elev*forest + elev*forest2 + elev2*forest
              data = willowUnm) #the data object

#this is by far the best model, but elevation in the detection function is probably too muc
mOverpred <- occu(formula = ~day + dur + forest + forest*day + elev # p formula
              ~elev + elev2 + forest + forest2 + elev2*forest + elev2*forest2 , #psi formul
              data = willowUnm) #the data object

# combine some representative models for displaying
fl <- fitList(
  "p(.)                                           psi(.)"
  "p(day + dur + forest)                          psi(elev + forest)"
  "p(day + dur + forest)                          psi(elev + elev^2 + forest + forest^2)"
  "p(day + dur + forest + forest^2)               psi(elev + elev^2 + forest + forest^2)"
  "p(day + dur + forest + forest*day)             psi(elev)"
  "p(day + dur + forest + forest*day)             psi(elev + elev^2*(forest + forest^2))"
  "p(day + dur + forest + forest*day)             psi((elev + elev^2)*(forest + forest^2))"
  "p(day + dur + forest + forest*day + elev)      psi((elev + elev^2)*(forest + forest^2))"

# model output table to format
ms <- modSel(fl)
```

```
# full model summary
summary(mOptm)
# state model
mOptm@estimates@estimates$state
#detection model
mOptm@estimates@estimates$det

# test for VIF?
# so much strutural collinearity that this probably doesn't matter
vif(mOptm, type = "state")
vif(mOptm, type = "det")

# GOF for best model....it's barely passable
# code below is parallelised, be careful all ye who lack 10 free cores
gof.boot <- mb.gof.test(mOptm, nsim = 1000, ncores = 10)
# save this and re-use output out of pity for my computer
write_rds(gof.boot, file = "gofBootstrap.rds")
# p-values generally vary between 0.4 and 0.9, but the difference to c-hat is negligible
# 10000 cores confirms a p-value around 0.7 but the plot is too ugly so this is not it
# gof.boot <- mb.gof.test(mOptm, nsim = 10000, ncores = 10)



# repeat for the overfit model
# fit is so temptingly good...but it just doesn't make sense
gof.boot.test <- mb.gof.test(mOverpred, nsim = 10000, ncores = 10)
write_rds(gof.boot.test, file = "gofBootstrapOverfit.rds")

# read in the saved for analysis
gof.boot <- read_rds("gofBootstrap.rds")
# even so, our model is very much closer to the tail of the chi^2 distribution and we're qu
ggplot() +
  geom_histogram(data = data.frame(t.star = gof.boot$t.star),
                 aes(x=t.star), color="black", fill="#fde725", alpha = 0.3, binwidth = 1) +
  geom_vline(aes(xintercept = gof.boot$chi.square), linewidth = 0.8, color = "#440154") +
  xlab("Pearson Chi-square statistic") +
  ylab("Count") +
  theme_bw() +
  ggtitle(bquote("Bootstrapped"~chi^2~"fit statistic (1000 samples, p ="~.(gof.boot$p.valu)

data(Switzerland)
gelev <- ggplot(data = Switzerland, aes(x=x, y=y,fill=elevation)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "B") +
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
```

```r
        legend.position = "bottom") +
  labs(x = "",
       y = "",
       fill = "Elevation (m)") +
  guides(fill = guide_colorbar(# draw border around the legend
                               frame.colour = "black",
                               barwidth = 10)) +
  coord_fixed()
gfor <- ggplot(data = Switzerland, aes(x=x, y=y,fill=forest)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "D") +
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        legend.position="bottom") +
  labs(x = "",
       y = "",
       fill = "Forest (%)")  +
  guides(fill = guide_colorbar(# draw border around the legend
    frame.colour = "black",
    barwidth = 10)) +
  coord_fixed()

for_pred <- data.frame(elev = (Switzerland$elevation - 1182.574)/646.333,      # convert or
                       elev2 = ((Switzerland$elevation - 1182.574)/646.333)^2, # convert or
                       forest = Switzerland$forest/100,           #want prop not %
                       forest2 = Switzerland$forest/100,          #want prop not %
                       X = Switzerland$x,                         #keep the coordinates
                       Y = Switzerland$y)                         #keep the coordinates
cowplot::plot_grid(gelev,gfor,nrow=2)

willowPredSDM <- modavgPred(list(mOptm), # top model
                            newdata = for_pred, #spatially indexed data frame
                            parm.type = "psi",  #predict from state model
                            c.hat = gof.boot$c.hat.est) #inflate SEs using Royle & Kery meth
#add data to predictions manually
willow_sdm <- for_pred %>% mutate(Predicted = willowPredSDM$mod.avg.pred,
                                  SE = willowPredSDM$uncond.se,
                                  lower = willowPredSDM$lower.CL,
                                  upper = willowPredSDM$upper.CL)

gpredM_1 <- ggplot(data = willow_sdm, aes(x=X, y=Y,fill=Predicted)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "H") +
  # add actual observations if we have x,y data
  # geom_point(data = willowNum, aes(x=X, y=Y)) +
```

```r
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        legend.position="bottom") +
  labs(x = "",
       y = "",
       fill = TeX(r'(Estimated $\psi$)'))  +
  guides(fill = guide_colorbar(# draw border around the legend
    frame.colour = "black",
    barwidth = 10)) +
  coord_fixed()

gpredE <- ggplot(data = willow_sdm, aes(x=X, y=Y,fill=SE)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "H") +
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        legend.position="bottom") +
  labs(x = "",
       y = "",
       fill = TeX(r'(Estimated $\psi$ Error)'))  +
  guides(fill = guide_colorbar(# draw border around the legend
    frame.colour = "black",
    barwidth = 10)) +
  coord_fixed()

cowplot::plot_grid(gelev,gfor,gpredM_1,gpredE,nrow=2)

gpredL <- ggplot(data = willow_sdm, aes(x=X, y=Y,fill=lower)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "H",
                       limits= c(0,1)) +
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        legend.position="left") +
  labs(x = "",
       y = "",
       fill = TeX(r'(Lower Bound $\psi$)')) +
```

```r
  guides(fill = guide_colorbar(# draw border around the legend
    frame.colour = "black")) +
  coord_fixed()

gpredM_2 <- ggplot(data = willow_sdm, aes(x=X, y=Y,fill=Predicted)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "H",
                       limits= c(0,1)) +
  # add actual observations if we have x,y data
  # geom_point(data = willowNum, aes(x=X, y=Y)) +
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        legend.position="left") +
  labs(x = "",
       y = "",
       fill = TeX(r'(Estimated $\psi$)'))  +
  guides(fill = guide_colorbar(# draw border around the legend
    frame.colour = "black")) +
  coord_fixed()

gpredH <- ggplot(data = willow_sdm, aes(x=X, y=Y,fill=upper)) +
  geom_raster() +
  scale_fill_viridis_c(direction = 1,
                       option = "H",
                       limits= c(0,1)) +
  theme_bw() +
  theme(axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        legend.position="left") +
  labs(x = "",
       y = "",
       fill = TeX(r'(Upper Bound $\psi$)'))  +
  guides(fill = guide_colorbar(# draw border around the legend
    frame.colour = "black")) +
  coord_fixed()

cowplot::plot_grid(gpredL, gpredM_2, gpredH,nrow=3)



#----------------------------------------------------------------------------------------#
#psi ~ elev | mean(forest)
pred_psi_eleL <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
```

```
                                      max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                      length = 30),
                           forest = quantile(probs = 0.25, willowUnm@siteCovs$forest, na.rm=
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleL <- modavgPred(list(mOptm), newdata = pred_psi_eleL, parm.type = "psi", c.hat = 
pred_psi_eleL <- pred_psi_eleL %>% mutate(Predicted = predPsiEleL$mod.avg.pred,
                                          SE = predPsiEleL$uncond.se,
                                          lower = predPsiEleL$lower.CL,
                                          upper = predPsiEleL$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleL <- ggplot(data = pred_psi_eleL, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

pred_psi_eleM <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                      max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                      length = 30),
                           forest = median(willowUnm@siteCovs$forest, na.rm=TRUE)) %>%
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleM <- modavgPred(list(mOptm), newdata = pred_psi_eleM, parm.type = "psi", c.hat = 
pred_psi_eleM <- pred_psi_eleM %>% mutate(Predicted = predPsiEleM$mod.avg.pred,
                                          SE = predPsiEleM$uncond.se,
                                          lower = predPsiEleM$lower.CL,
                                          upper = predPsiEleM$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleM <- ggplot(data = pred_psi_eleM, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

pred_psi_eleH <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                      max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                      length = 30),
                           forest = quantile(probs = 0.75, willowUnm@siteCovs$forest, na.rm=
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleH <- modavgPred(list(mOptm), newdata = pred_psi_eleH, parm.type = "psi", c.hat = 
pred_psi_eleH <- pred_psi_eleH %>% mutate(Predicted = predPsiEleH$mod.avg.pred,
                                          SE = predPsiEleH$uncond.se,
                                          lower = predPsiEleH$lower.CL,
                                          upper = predPsiEleH$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleH <- ggplot(data = pred_psi_eleH, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

#--------------------------------------------------------------------------------------#
```

```
#psi ~ for | mean(elev)
pred_psi_forL <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                      max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                      length = 30),
                           elev = quantile(probs = 0.25, willowUnm@siteCovs$elev, na.rm=TRUE
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForL <- modavgPred(list(mOptm), newdata = pred_psi_forL, parm.type = "psi", c.hat =
pred_psi_forL <- pred_psi_forL %>% mutate(Predicted = predPsiForL$mod.avg.pred,
                                          SE = predPsiForL$uncond.se,
                                          lower = predPsiForL$lower.CL,
                                          upper = predPsiForL$upper.CL,
                                          forestP = forest*100)
ggpsiforL <- ggplot(data = pred_psi_forL, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

pred_psi_forM <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        length = 30),
                           elev = median(willowUnm@siteCovs$elev, na.rm=TRUE)) %>%
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForM <- modavgPred(list(mOptm), newdata = pred_psi_forM, parm.type = "psi", c.hat =
pred_psi_forM <- pred_psi_forM %>% mutate(Predicted = predPsiForM$mod.avg.pred,
                                          SE = predPsiForM$uncond.se,
                                          lower = predPsiForM$lower.CL,
                                          upper = predPsiForM$upper.CL,
                                          forestP = forest*100)
ggpsiforM <- ggplot(data = pred_psi_forM, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

pred_psi_forH <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        length = 30),
                           elev = quantile(probs = 0.75, willowUnm@siteCovs$elev, na.rm=TRUE
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForH <- modavgPred(list(mOptm), newdata = pred_psi_forH, parm.type = "psi", c.hat =
pred_psi_forH <- pred_psi_forH %>% mutate(Predicted = predPsiForH$mod.avg.pred,
                                    SE = predPsiForH$uncond.se,
                                    lower = predPsiForH$lower.CL,
                                    upper = predPsiForH$upper.CL,
                                    forestP = forest*100)
ggpsiforH <- ggplot(data = pred_psi_forH, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()
```

```
cowplot::plot_grid(ggpsieleL, ggpsieleM, ggpsieleH, ggpsiforL, ggpsiforM, ggpsiforH, nrow=2)

#------------------------------------------------------------------------------------------#
#psi ~ elev | quantile(forest)
pred_psi_eleL <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       length = 30),
                            forest = quantile(probs = 0.25, willowUnm@siteCovs$forest, na.rm=
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleL <- modavgPred(list(mOptm), newdata = pred_psi_eleL, parm.type = "psi", c.hat = g
pred_psi_eleL <- pred_psi_eleL %>% mutate(Predicted = predPsiEleL$mod.avg.pred,
                                          SE = predPsiEleL$uncond.se,
                                          lower = predPsiEleL$lower.CL,
                                          upper = predPsiEleL$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleL <- ggplot(data = pred_psi_eleL, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

pred_psi_eleM <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       length = 30),
                            forest = median(willowUnm@siteCovs$forest, na.rm=TRUE)) %>%
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleM <- modavgPred(list(mOptm), newdata = pred_psi_eleM, parm.type = "psi", c.hat = g
pred_psi_eleM <- pred_psi_eleM %>% mutate(Predicted = predPsiEleM$mod.avg.pred,
                                          SE = predPsiEleM$uncond.se,
                                          lower = predPsiEleM$lower.CL,
                                          upper = predPsiEleM$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleM <- ggplot(data = pred_psi_eleM, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

pred_psi_eleH <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       length = 30),
                            forest = quantile(probs = 0.75, willowUnm@siteCovs$forest, na.rm=
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleH <- modavgPred(list(mOptm), newdata = pred_psi_eleH, parm.type = "psi", c.hat = g
pred_psi_eleH <- pred_psi_eleH %>% mutate(Predicted = predPsiEleH$mod.avg.pred,
                                          SE = predPsiEleH$uncond.se,
                                          lower = predPsiEleH$lower.CL,
                                          upper = predPsiEleH$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
```

```
ggpsieleH <- ggplot(data = pred_psi_eleH, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

#------------------------------------------------------------------------------------------#
#psi ~ for | quantile(elev)
pred_psi_forL <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                      max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                      length = 30),
                           elev = quantile(probs = 0.25, willowUnm@siteCovs$elev, na.rm=TRUE
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForL <- modavgPred(list(mOptm), newdata = pred_psi_forL, parm.type = "psi", c.hat = g
pred_psi_forL <- pred_psi_forL %>% mutate(Predicted = predPsiForL$mod.avg.pred,
                                          SE = predPsiForL$uncond.se,
                                          lower = predPsiForL$lower.CL,
                                          upper = predPsiForL$upper.CL,
                                          forestP = forest*100)
ggpsiforL <- ggplot(data = pred_psi_forL, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

pred_psi_forM <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        length = 30),
                           elev = median(willowUnm@siteCovs$elev, na.rm=TRUE)) %>%
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForM <- modavgPred(list(mOptm), newdata = pred_psi_forM, parm.type = "psi", c.hat = g
pred_psi_forM <- pred_psi_forM %>% mutate(Predicted = predPsiForM$mod.avg.pred,
                                          SE = predPsiForM$uncond.se,
                                          lower = predPsiForM$lower.CL,
                                          upper = predPsiForM$upper.CL,
                                          forestP = forest*100)
ggpsiforM <- ggplot(data = pred_psi_forM, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

pred_psi_forH <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                        length = 30),
                           elev = quantile(probs = 0.75, willowUnm@siteCovs$elev, na.rm=TRUE
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForH <- modavgPred(list(mOptm), newdata = pred_psi_forH, parm.type = "psi", c.hat = g
pred_psi_forH <- pred_psi_forH %>% mutate(Predicted = predPsiForH$mod.avg.pred,
                                    SE = predPsiForH$uncond.se,
                                    lower = predPsiForH$lower.CL,
```

```
                                    upper = predPsiForH$upper.CL,
                                    forestP = forest*100)
ggpsiforH <- ggplot(data = pred_psi_forH, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

cowplot::plot_grid(ggpsieleL, ggpsieleM, ggpsieleH, ggpsiforL, ggpsiforM, ggpsiforH, nrow=2)



#------------------------------------------------------------------------------------#
#p ~ dur | median(day & forest)
pred_p_dur <- data.frame(dur = seq(min(willowUnm@obsCovs$dur, na.rm=TRUE),
                                   max(willowUnm@obsCovs$dur, na.rm=TRUE),
                                   length = 30),
                         day = median(willowUnm@obsCovs$day, na.rm=TRUE),
                         forest = median(willowUnm@obsCovs$forest, na.rm=TRUE))
predPDur <- modavgPred(list(mOptm), newdata = pred_p_dur, parm.type = "detect", c.hat = gof
pred_p_dur <- pred_p_dur %>% mutate(Predicted = predPDur$mod.avg.pred,
                                    SE = predPDur$uncond.se,
                                    lower = predPDur$lower.CL,
                                    upper = predPDur$upper.CL,
                                    forestP = forest*100)
pDurPlot <- ggplot(data = pred_p_dur, aes(x = dur, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Duration (min)") + ylim(0,1) + theme_bw()
#------------------------------------------------------------------------------------#
#p ~ day | median(dur & forest)
pred_p_dayL <- data.frame(day = seq(min(willowUnm@obsCovs$day, na.rm=TRUE),
                                    max(willowUnm@obsCovs$day, na.rm=TRUE),
                                    length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          forest = quantile(probs = 0.25, willowUnm@obsCovs$forest, na.rm=TR
predPDayL <- modavgPred(list(mOptm), newdata = pred_p_dayL, parm.type = "detect", c.hat = go
pred_p_dayL <- pred_p_dayL %>% mutate(Predicted = predPDayL$mod.avg.pred,
                                      SE = predPDayL$uncond.se,
                                      lower = predPDayL$lower.CL,
                                      upper = predPDayL$upper.CL,
                                      forestP = forest*100)
pDayPlotL <- ggplot(data = pred_p_dayL, aes(x = day, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Day (8% Forested)") + ylim(0,1) + theme_bw()
#p ~ day | Q2(dur & forest)
pred_p_day <- data.frame(day = seq(min(willowUnm@obsCovs$day, na.rm=TRUE),
                                   max(willowUnm@obsCovs$day, na.rm=TRUE),
                                   length = 30),
                         dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                         forest = quantile(probs = 0.5, willowUnm@obsCovs$forest, na.rm=TRUE
```

```
predPDay <- modavgPred(list(mOptm), newdata = pred_p_day, parm.type = "detect", c.hat = gof
pred_p_day <- pred_p_day %>% mutate(Predicted = predPDay$mod.avg.pred,
                                    SE = predPDay$uncond.se,
                                    lower = predPDay$lower.CL,
                                    upper = predPDay$upper.CL,
                                    forestP = forest*100)
pDayPlot <- ggplot(data = pred_p_day, aes(x = day, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Day (33% Forested)") + ylim(0,1) + theme_bw()
#p ~ day | Q4(dur & forest)
pred_p_dayH <- data.frame(day = seq(min(willowUnm@obsCovs$day, na.rm=TRUE),
                                   max(willowUnm@obsCovs$day, na.rm=TRUE),
                                   length = 30),
                         dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                         forest = quantile(probs = 0.75, willowUnm@obsCovs$forest, na.rm=TRU
predPDayH <- modavgPred(list(mOptm), newdata = pred_p_dayH, parm.type = "detect", c.hat = go
pred_p_dayH <- pred_p_dayH %>% mutate(Predicted = predPDayH$mod.avg.pred,
                                      SE = predPDayH$uncond.se,
                                      lower = predPDayH$lower.CL,
                                      upper = predPDayH$upper.CL,
                                      forestP = forest*100)
pDayPlotH <- ggplot(data = pred_p_dayH, aes(x = day, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Day (57% Forested)") + ylim(0,1) + theme_bw()
#-----------------------------------------------------------------------------------------#
#p ~ for | Q2(day), median(dur)
pred_p_forL <- data.frame(forest = seq(min(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       max(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          day = quantile(probs = 0.25, willowUnm@obsCovs$day, na.rm=TRUE))
predPForL <- modavgPred(list(mOptm), newdata = pred_p_forL, parm.type = "detect", c.hat = go
pred_p_forL <- pred_p_forL %>% mutate(Predicted = predPForL$mod.avg.pred,
                                      SE = predPForL$uncond.se,
                                      lower = predPForL$lower.CL,
                                      upper = predPForL$upper.CL,
                                      forestP = forest*100)
pForPlotL <- ggplot(data = pred_p_forL, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Forest (%) on Day 38") + ylim(0,1) + theme_bw()
#p ~ for | median(dur & day)
pred_p_for <- data.frame(forest = seq(min(willowUnm@obsCovs$forest, na.rm=TRUE),
                                      max(willowUnm@obsCovs$forest, na.rm=TRUE),
                                   length = 30),
                         dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                         day = quantile(probs = 0.5, willowUnm@obsCovs$day, na.rm=TRUE))
predPFor <- modavgPred(list(mOptm), newdata = pred_p_for, parm.type = "detect", c.hat = gof
pred_p_for <- pred_p_for %>% mutate(Predicted = predPFor$mod.avg.pred,
```

```
                                          SE = predPFor$uncond.se,
                                          lower = predPFor$lower.CL,
                                          upper = predPFor$upper.CL,
                                          forestP = forest*100)
pForPlot <- ggplot(data = pred_p_for, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Forest (%) on Day 52") + ylim(0,1) + theme_bw()
#p ~ for | Q4(day), median(dur)
pred_p_forH <- data.frame(forest = seq(min(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       max(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          day = quantile(probs = 0.75, willowUnm@obsCovs$day, na.rm=TRUE))
predPForH <- modavgPred(list(mOptm), newdata = pred_p_forH, parm.type = "detect", c.hat = go
pred_p_forH <- pred_p_forH %>% mutate(Predicted = predPForH$mod.avg.pred,
                                      SE = predPForH$uncond.se,
                                      lower = predPForH$lower.CL,
                                      upper = predPForH$upper.CL,
                                      forestP = forest*100)
ggplot(data = pred_p_forH, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Forest (%) on Day 72") + ylim(0,1) + theme_bw()

# all together
cowplot::plot_grid(pDayPlotL, pDayPlot, pDayPlotH, pForPlotL, pForPlot, pForPlotH, nrow=2)
pDurPlot


# generate predictions for 4 quadrants of interest
willowPred <- willowNum %>% filter(id %in% c(25, 62, 150, 203)) %>%
  rename(elev2 = elevsq,
         forest2 = forestsq)
# predicting occurrence from the state process
predQuads <- modavgPred(list(mOptm), newdata = willowPred, parm.type = "psi", c.hat = gof.bo
willowRes <- willowPred %>% mutate(Predicted = predQuads$mod.avg.pred,
                                   SE = predQuads$uncond.se,
                                   lower = predQuads$lower.CL,
                                   upper = predQuads$upper.CL,
                                   elev = round((646.333*elev+1182.574),0)) %>%
  select(-c(elev2, forest2, iLength))
```

```
## APPENDIX PART 2: Testing Overfit Model Parameter Effects
#--------------------------------------------------------------------------------------#
#psi ~ elev | mean(forest)
pred_psi_eleL <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       length = 30),
                            forest = quantile(probs = 0.25, willowUnm@siteCovs$forest, na.rm
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleL <- modavgPred(list(mOverpred), newdata = pred_psi_eleL, parm.type = "psi", c.hat
pred_psi_eleL <- pred_psi_eleL %>% mutate(Predicted = predPsiEleL$mod.avg.pred,
                                          SE = predPsiEleL$uncond.se,
                                          lower = predPsiEleL$lower.CL,
                                          upper = predPsiEleL$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleL <- ggplot(data = pred_psi_eleL, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

pred_psi_eleM <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       length = 30),
                            forest = median(willowUnm@siteCovs$forest, na.rm=TRUE)) %>%
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleM <- modavgPred(list(mOverpred), newdata = pred_psi_eleM, parm.type = "psi", c.hat
pred_psi_eleM <- pred_psi_eleM %>% mutate(Predicted = predPsiEleM$mod.avg.pred,
                                          SE = predPsiEleM$uncond.se,
                                          lower = predPsiEleM$lower.CL,
                                          upper = predPsiEleM$upper.CL,
                                          elevR = 1182.574 + elev*646.333)
ggpsieleM <- ggplot(data = pred_psi_eleM, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

pred_psi_eleH <- data.frame(elev = seq(min(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       max(willowUnm@siteCovs$elev, na.rm=TRUE),
                                       length = 30),
                            forest = quantile(probs = 0.75, willowUnm@siteCovs$forest, na.rm
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiEleH <- modavgPred(list(mOverpred), newdata = pred_psi_eleH, parm.type = "psi", c.hat
pred_psi_eleH <- pred_psi_eleH %>% mutate(Predicted = predPsiEleH$mod.avg.pred,
                                          SE = predPsiEleH$uncond.se,
```

```
                                        lower = predPsiEleH$lower.CL,
                                        upper = predPsiEleH$upper.CL,
                                        elevR = 1182.574 + elev*646.333)
ggpsieleH <- ggplot(data = pred_psi_eleH, aes(x = elevR, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Elevation (m)") + ylim(0,1) + theme_bw()

#-------------------------------------------------------------------------------------#
#psi ~ for | mean(elev)
pred_psi_forL <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                         max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                         length = 30),
                            elev = quantile(probs = 0.25, willowUnm@siteCovs$elev, na.rm=TRU
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForL <- modavgPred(list(mOverpred), newdata = pred_psi_forL, parm.type = "psi", c.hat
pred_psi_forL <- pred_psi_forL %>% mutate(Predicted = predPsiForL$mod.avg.pred,
                                        SE = predPsiForL$uncond.se,
                                        lower = predPsiForL$lower.CL,
                                        upper = predPsiForL$upper.CL,
                                        forestP = forest*100)
ggpsiforL <- ggplot(data = pred_psi_forL, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

pred_psi_forM <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                         max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                         length = 30),
                            elev = median(willowUnm@siteCovs$elev, na.rm=TRUE)) %>%
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForM <- modavgPred(list(mOverpred), newdata = pred_psi_forM, parm.type = "psi", c.hat
pred_psi_forM <- pred_psi_forM %>% mutate(Predicted = predPsiForM$mod.avg.pred,
                                        SE = predPsiForM$uncond.se,
                                        lower = predPsiForM$lower.CL,
                                        upper = predPsiForM$upper.CL,
                                        forestP = forest*100)
ggpsiforM <- ggplot(data = pred_psi_forM, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

pred_psi_forH <- data.frame(forest = seq(min(willowUnm@siteCovs$forest, na.rm=TRUE),
                                         max(willowUnm@siteCovs$forest, na.rm=TRUE),
                                         length = 30),
                            elev = quantile(probs = 0.75, willowUnm@siteCovs$elev, na.rm=TRU
  mutate(elev2 = elev^2,
         forest2 = forest^2)
predPsiForH <- modavgPred(list(mOverpred), newdata = pred_psi_forH, parm.type = "psi", c.hat
```

```
pred_psi_forH <- pred_psi_forH %>% mutate(Predicted = predPsiForH$mod.avg.pred,
                                          SE = predPsiForH$uncond.se,
                                          lower = predPsiForH$lower.CL,
                                          upper = predPsiForH$upper.CL,
                                          forestP = forest*100)
ggpsiforH <- ggplot(data = pred_psi_forH, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#21918c", alpha=0.1) +
  geom_line(size=1,color="#21918c") +
  ylab("P(Occupied)") + xlab("Forest Cover (%)") + ylim(0,1) + theme_bw()

cowplot::plot_grid(ggpsieleL, ggpsieleM, ggpsieleH, ggpsiforL, ggpsiforM, ggpsiforH, nrow=2)

#------------------------------------------------------------------------------------------#
#p ~ elev | median(day & forest & dur)
pred_p_elev <- data.frame(elev = seq(min(willowUnm@obsCovs$elev, na.rm=TRUE),
                                     max(willowUnm@obsCovs$elev, na.rm=TRUE),
                                     length = 30),
                          day = median(willowUnm@obsCovs$day, na.rm=TRUE),
                          forest = median(willowUnm@obsCovs$forest, na.rm=TRUE),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE))
predPEle <- modavgPred(list(mOverpred), newdata = pred_p_elev, parm.type = "detect", c.hat =
pred_p_elev <- pred_p_elev %>% mutate(Predicted = predPEle$mod.avg.pred,
                                      SE = predPEle$uncond.se,
                                      lower = predPEle$lower.CL,
                                      upper = predPEle$upper.CL,
                                      forestP = forest*100)
ggplot(data = pred_p_elev, aes(x = elev, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Elevation") + ylim(0,1) + theme_bw()
#------------------------------------------------------------------------------------------#
#p ~ dur | median(day & forest)
pred_p_dur <- data.frame(dur = seq(min(willowUnm@obsCovs$dur, na.rm=TRUE),
                                   max(willowUnm@obsCovs$dur, na.rm=TRUE),
                                   length = 30),
                         day = median(willowUnm@obsCovs$day, na.rm=TRUE),
                         forest = median(willowUnm@obsCovs$forest, na.rm=TRUE),
                         elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPDur <- modavgPred(list(mOverpred), newdata = pred_p_dur, parm.type = "detect", c.hat =
pred_p_dur <- pred_p_dur %>% mutate(Predicted = predPDur$mod.avg.pred,
                                    SE = predPDur$uncond.se,
                                    lower = predPDur$lower.CL,
                                    upper = predPDur$upper.CL,
                                    forestP = forest*100)
ggplot(data = pred_p_dur, aes(x = dur, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Duration (min)") + ylim(0,1) + theme_bw()
#------------------------------------------------------------------------------------------#
#p ~ day | Q2(forest) + median(everything else)
pred_p_dayL <- data.frame(day = seq(min(willowUnm@obsCovs$day, na.rm=TRUE),
```

```
                                    max(willowUnm@obsCovs$day, na.rm=TRUE),
                                    length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          forest = quantile(probs = 0.25, willowUnm@obsCovs$forest, na.rm=TR
                          elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPDayL <- modavgPred(list(mOverpred), newdata = pred_p_dayL, parm.type = "detect", c.hat
pred_p_dayL <- pred_p_dayL %>% mutate(Predicted = predPDayL$mod.avg.pred,
                                      SE = predPDayL$uncond.se,
                                      lower = predPDayL$lower.CL,
                                      upper = predPDayL$upper.CL,
                                      forestP = forest*100)
pDayPlotL <- ggplot(data = pred_p_dayL, aes(x = day, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Day (8% Forested)") + ylim(0,1) + theme_bw()
#p ~ day | Q2(dur & forest)
pred_p_day <- data.frame(day = seq(min(willowUnm@obsCovs$day, na.rm=TRUE),
                                   max(willowUnm@obsCovs$day, na.rm=TRUE),
                                   length = 30),
                         dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                         forest = quantile(probs = 0.5, willowUnm@obsCovs$forest, na.rm=TRUE
                         elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPDay <- modavgPred(list(mOverpred), newdata = pred_p_day, parm.type = "detect", c.hat =
pred_p_day <- pred_p_day %>% mutate(Predicted = predPDay$mod.avg.pred,
                                    SE = predPDay$uncond.se,
                                    lower = predPDay$lower.CL,
                                    upper = predPDay$upper.CL,
                                    forestP = forest*100)
pDayPlot <- ggplot(data = pred_p_day, aes(x = day, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Day (33% Forested)") + ylim(0,1) + theme_bw()
#p ~ day | Q4(forest) + median(everything else)
pred_p_dayH <- data.frame(day = seq(min(willowUnm@obsCovs$day, na.rm=TRUE),
                                    max(willowUnm@obsCovs$day, na.rm=TRUE),
                                    length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          forest = quantile(probs = 0.75, willowUnm@obsCovs$forest, na.rm=TR
                          elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPDayH <- modavgPred(list(mOverpred), newdata = pred_p_dayH, parm.type = "detect", c.hat
pred_p_dayH <- pred_p_dayH %>% mutate(Predicted = predPDayH$mod.avg.pred,
                                      SE = predPDayH$uncond.se,
                                      lower = predPDayH$lower.CL,
                                      upper = predPDayH$upper.CL,
                                      forestP = forest*100)
pDayPlotH <- ggplot(data = pred_p_dayH, aes(x = day, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Survey Day (57% Forested)") + ylim(0,1) + theme_bw()
#-------------------------------------------------------------------------------------#
#p ~ for | Q2(day), median(everything else)
```

```r
pred_p_forL <- data.frame(forest = seq(min(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       max(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          day = quantile(probs = 0.25, willowUnm@obsCovs$day, na.rm=TRUE),
                          elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPForL <- modavgPred(list(mOverpred), newdata = pred_p_forL, parm.type = "detect", c.hat
pred_p_forL <- pred_p_forL %>% mutate(Predicted = predPForL$mod.avg.pred,
                                      SE = predPForL$uncond.se,
                                      lower = predPForL$lower.CL,
                                      upper = predPForL$upper.CL,
                                      forestP = forest*100)
pForPlotL <- ggplot(data = pred_p_forL, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Forest (%) on Day 38") + ylim(0,1) + theme_bw()
#p ~ for | median(everything
pred_p_for <- data.frame(forest = seq(min(willowUnm@obsCovs$forest, na.rm=TRUE),
                                      max(willowUnm@obsCovs$forest, na.rm=TRUE),
                                      length = 30),
                         dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                         day = quantile(probs = 0.5, willowUnm@obsCovs$day, na.rm=TRUE),
                         elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPFor <- modavgPred(list(mOverpred), newdata = pred_p_for, parm.type = "detect", c.hat =
pred_p_for <- pred_p_for %>% mutate(Predicted = predPFor$mod.avg.pred,
                                    SE = predPFor$uncond.se,
                                    lower = predPFor$lower.CL,
                                    upper = predPFor$upper.CL,
                                    forestP = forest*100)
pForPlot <- ggplot(data = pred_p_for, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Forest (%) on Day 52") + ylim(0,1) + theme_bw()
#p ~ for | Q4(day), median(others)
pred_p_forH <- data.frame(forest = seq(min(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       max(willowUnm@obsCovs$forest, na.rm=TRUE),
                                       length = 30),
                          dur = median(willowUnm@obsCovs$dur, na.rm=TRUE),
                          day = quantile(probs = 0.75, willowUnm@obsCovs$day, na.rm=TRUE),
                          elev = median(willowUnm@obsCovs$elev, na.rm=TRUE))
predPForH <- modavgPred(list(mOverpred), newdata = pred_p_forH, parm.type = "detect", c.hat
pred_p_forH <- pred_p_forH %>% mutate(Predicted = predPForH$mod.avg.pred,
                                      SE = predPForH$uncond.se,
                                      lower = predPForH$lower.CL,
                                      upper = predPForH$upper.CL,
                                      forestP = forest*100)
ggplot(data = pred_p_forH, aes(x = forestP, y = Predicted)) +
  geom_ribbon(aes(ymin=lower, ymax=upper), fill="#440154", alpha=0.1) +
  geom_line(size=1,color="#440154") +
  ylab("P(Detected)") + xlab("Forest (%) on Day 72") + ylim(0,1) + theme_bw()
```

```r
# all together
cowplot::plot_grid(pDayPlotL, pDayPlot, pDayPlotH, pForPlotL, pForPlot, pForPlotH, nrow=2)
```

References

1. Kumpula, S., Vatka, E., Orell, M. & Rytkönen, S. Effects of forest management on the spatial distribution of the willow tit (*Poecile montanus*). *Forest Ecology and Management* **529**, 120694 (2023).
2. Parry, W. & Broughton, R. K. Nesting behaviour and breeding success of Willow Tits Poecile montanus in north-west England. *Ringing & Migration* **33**, 75–85 (2018).
3. Knaus, P. *et al. Swiss Breeding Bird Atlas 2013–2016. Distribution and Population Trends of Birds in Switzerland and Liechtenstein.* (Swiss Ornithological Institute, Sempach, 2018).
4. Riecke, T. V., Gibson, D., Kéry, M. & Schaub, M. Sharing detection heterogeneity information among species in community models of occupancy and abundance can strengthen inference. *Ecology and Evolution* **11**, 18125–18135 (2021).
5. Bani, L., Luppi, M., Rocchia, E., Dondina, O. & Orioli, V. Winners and losers: How the elevational range of breeding birds on Alps has varied over the past four decades due to climate and habitat changes. *Ecology and Evolution* **9**, 1289–1305 (2019).
6. Sutherland, C. Statsecol: A Package Assocaited With The Statistical Ecology MSc. (2024).
7. Kellner, K. F. *et al.* The unmarked R package: Twelve years of advances in occurrence and abundance modelling in ecology. *Methods in Ecology and Evolution* **14**, 1408–1415 (2023).
8. R Core Team. R: A Language and Environment for Statistical Computing. (2023).
9. Bartón, K. MuMIn: Multi-Model Inference. (2023).
10. Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**, 716–723 (1974).
11. Mazerolle, M. J. AICcmodavg: Model selection and multimodel inference based on (Q)AIC(c). (2023).
12. Kéry, M. & Royle, J. A. *Applied Hierarchical Modeling in Ecology: Analysis of Distribution, Abundance and Species Richness in R and BUGS.* vol. 1 (Elsevier/AP, Academic Press is an imprint of Elsevier, Amsterdam ; Boston, 2016).
13. Vanhinsbergh, D., Fuller, R. J. & Noble, D. *A Review of Possible Causes of Recent Changes in Populations of Woodland Birds in Britain.* (British Trust for Ornithology, Norfolk, 2003).
14. Rottler, E., Kormann, C., Francke, T. & Bronstert, A. Elevation-dependent warming in the Swiss Alps 1981–2017: Features, forcings and feedbacks. *International Journal of Climatology* **39**, 2556–2568 (2019).
15. HOHNSBEIN, J., PIEKARSKI, C., KAMPMANN, B. & NOACK, Th. Effects of heat on visual acuity. *Ergonomics* **27**, 1239–1246 (1984).
16. Cole, D. *Parameter Redundancy and Identifiability.* (Chapman; Hall/CRC, New York, 2020). doi:10.1201/9781315120003.