

SEPP-601

Claire Atkins

2023-04-28

Packages used:

- library(glmmTMB)
- library(summarytools)
- library(arm)
- library(Weighted.Desc.Stat)
- library(magrittr)
- library(dplyr)
- library(ggeffects)
- library(FSA)
- library(gdata)
- library(car)
- library(olsrr)
- library(dplyr)
- library(jtools)
- library(ggplot2)
- library(grid)
- library(readxl)
- library(gridExtra)
- library(sjPlot)
- library(sjmisc)
- library(effects)
- library(MASS)
- library(GGally)

1. Data manipulation

```
#create "success" column where TP/TN = 1 and FN = 0
```

```

real_data <- real_data %>%
mutate(success = case_when(
  detection == "TP" ~ "1",
  detection == "TN" ~ "1",
  detection == "FN" ~ "0"))
#create "det_levels" column where detection types are set as levels 1, 2, 3
real_data <- real_data %>%
  mutate (det_levels = case_when(
    detection == "TP" ~ "1",
    detection == "TN" ~ "2",
    detection == "FN" ~ "3"))
real_data$det_levels <- as.factor(real_data$det_levels)

#create "present" column where a decoy is present in search area = 1
real_data <- real_data %>%
  mutate (present = case_when (
    sa_density >= "1" ~ "1",
    sa_density == "0" ~ "0"
  ))

#create column where a decoy is present in the enclosure to determine likelihood
that a decoy will occur in the search area
real_data <- real_data %>%
  mutate (ex_present = case_when (
    ex_density >= "1" ~ "1",
    ex_density == "0" ~ "0"
  ))

#subset of df where "present" = 1
present <- subset(real_data, present == 1)
present$det_levels <- as.factor(present$det_levels)

#subset of df where "ex_present" = 1
ex_present <- subset(real_data, ex_present == 1)

#subset of df without FN - decoys were detected during a sweep even if individual
decoys were missed along the way
rem.FN <- subset(real_data, detection == c("TP", "TN"))
rem.FN <- rem.FN %>%
  mutate(det_levels = case_when (
    det_levels == "1" ~ "0",
    det_levels == "0" ~ "1"
  ))

```

2. Frequencies in raw data

```
#how often did each detection type occur?
```

```
freq(real_data$detection)
```

```
## Frequencies
## real_data$detection
## Type: Character
##
##           Freq  % Valid  % Valid Cum.  % Total  % Total Cum.
## -----
##           FN     4     9.52         9.52     9.52         9.52
##           TN    31    73.81        83.33    73.81        83.33
##           TP     7    16.67       100.00   16.67       100.00
##          <NA>     0     0.00         0.00    0.00       100.00
##          Total   42   100.00       100.00  100.00       100.00
```

```
#how often was each size class detected or missed?
```

```
real_data %>%
  count(size_class, detection)
```

```
##   size_class detection  n
## 1      ad      FN      1
## 2      ad      TP      2
## 3     egg      FN      1
## 4     egg      TP      1
## 5     juv      FN      1
## 6    none      TN     31
## 7     sub      FN      1
## 8     sub      TP      4
```

```
#how often did each detection type occur per survey type?
```

```
real_data %>%
  count(survey_type, detection)
```

```
##   survey_type detection  n
## 1      quad      FN      2
## 2      quad      TN      1
## 3      quad      TP      3
## 4  rand_quad      FN      1
## 5  rand_quad      TN     13
## 6  rand_quad      TP      2
## 7  rand_tran      FN      1
## 8  rand_tran      TN     17
```

```
## 9   rand_tran      TP  2
```

```
#how often were searches successful in each survey method?
```

```
real_data %>%  
  count(survey_type, success)
```

```
##   survey_type success  n  
## 1      quad      0    2  
## 2      quad      1    4  
## 3  rand_quad      0    1  
## 4  rand_quad      1   15  
## 5  rand_tran      0    1  
## 6  rand_tran      1   19
```

```
#what is the range, mean, and standard deviation of effort per search area?
```

```
range(real_data$effort)
```

```
## [1] 0.3333333 4.4000000
```

```
mean(real_data$effort)
```

```
## [1] 1.340344
```

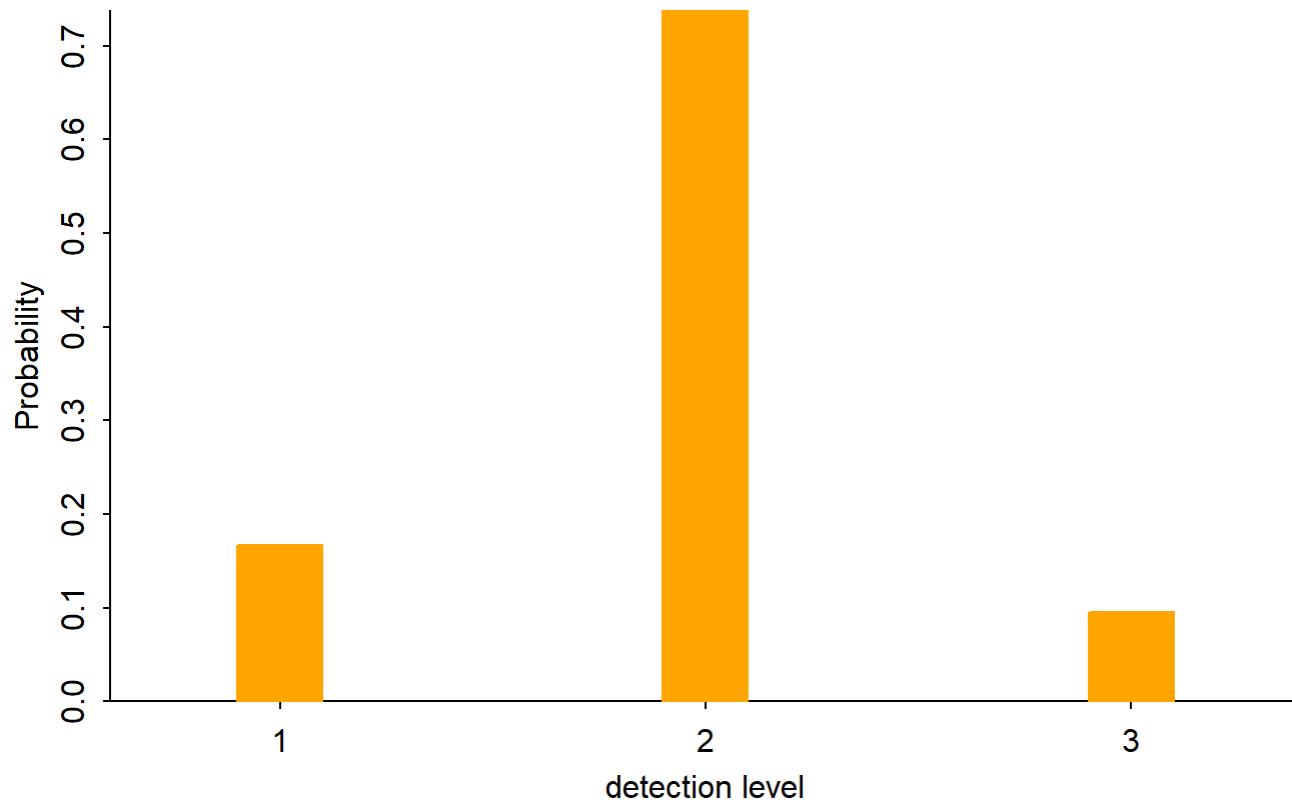
```
sd(real_data$effort)
```

```
## [1] 0.9069789
```

```
#what are the probabilities of each detection level, where:
```

```
# 1 = true positive, 2 = true negative, 3 = false negative
```

```
discrete.histogram(real_data$det_levels,  
                   xlab = "detection level", prob.col = "orange")
```



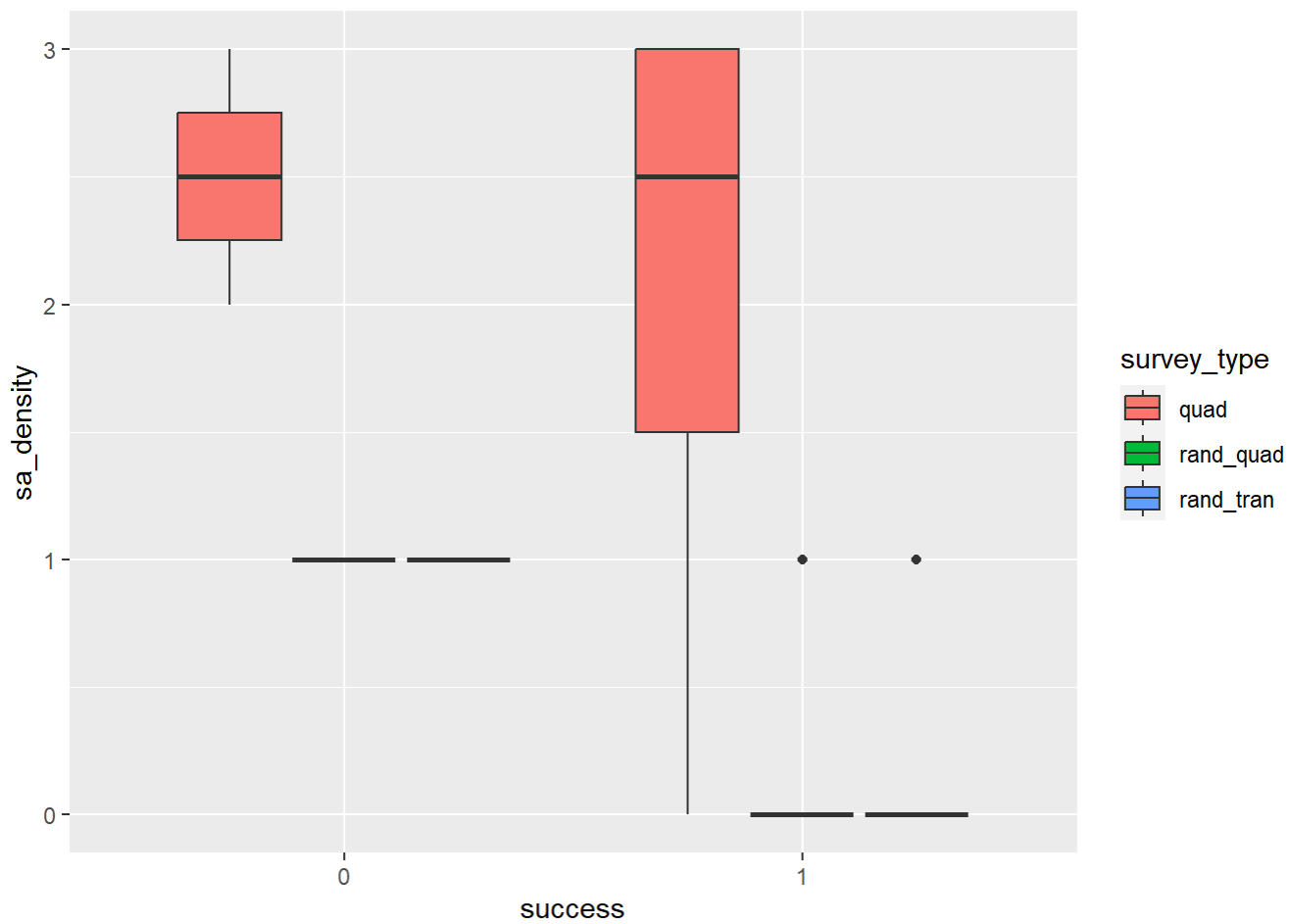
#when at least one decoy is present in the exclosure, how often does each survey type include a decoy in the search area?

```
ex_present %>%
  count(survey_type, present)
```

```
##   survey_type present  n
## 1      quad         0  1
## 2      quad         1  5
## 3  rand_quad         0 13
## 4  rand_quad         1  3
## 5  rand_tran         0  5
## 6  rand_tran         1  3
```

3. Raw data visualization

```
plot2 <- ggplot(real_data, aes(success, sa_density)) + geom_boxplot(
  aes(fill = survey_type)
)
plot2
```

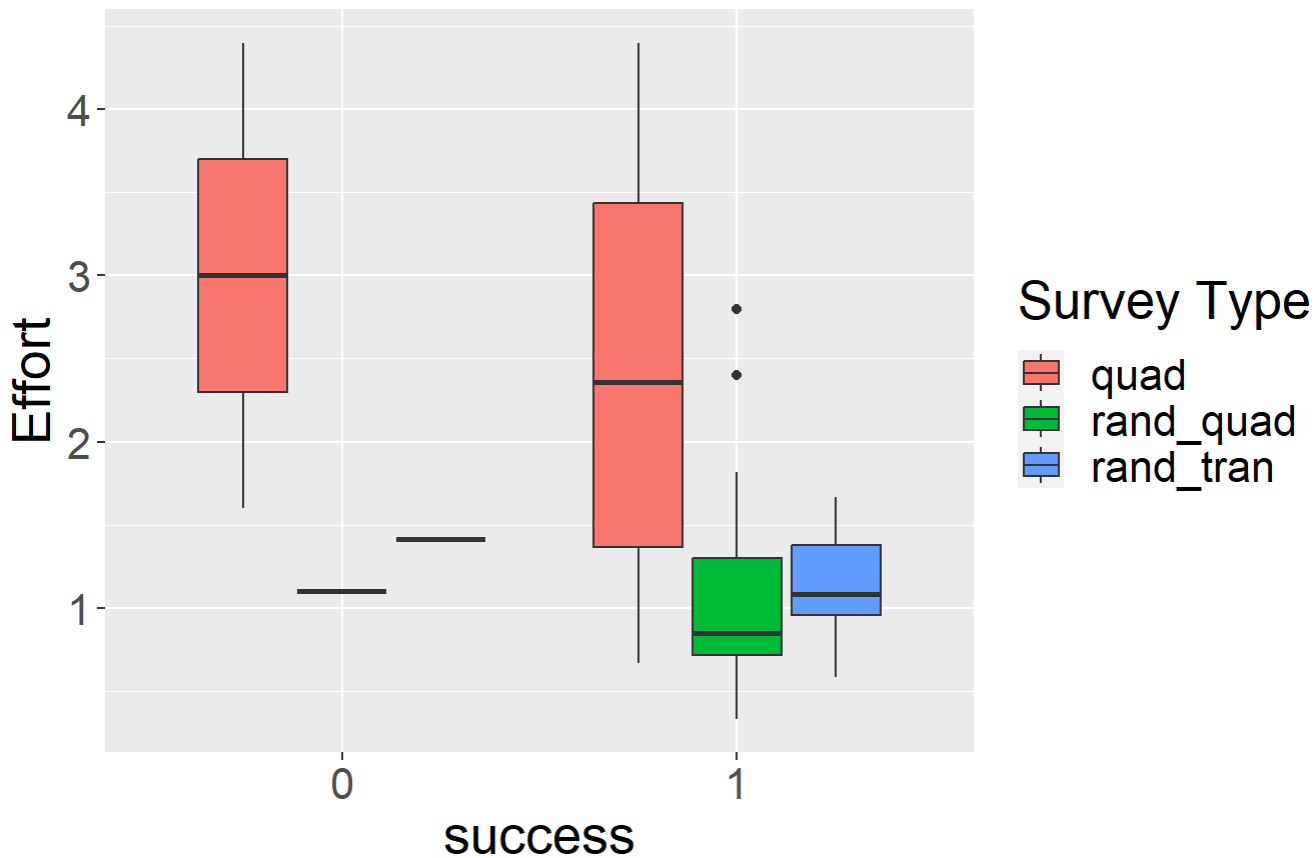


```

effort.plot <- ggplot(real_data, aes(success, effort)) +
  geom_boxplot(aes(fill = survey_type)) +
  theme(text = element_text(size = 20)) +
  labs(title = "Effort expended by survey type", y = "Effort", x = "success") +
  labs(fill = "Survey Type")
effort.plot

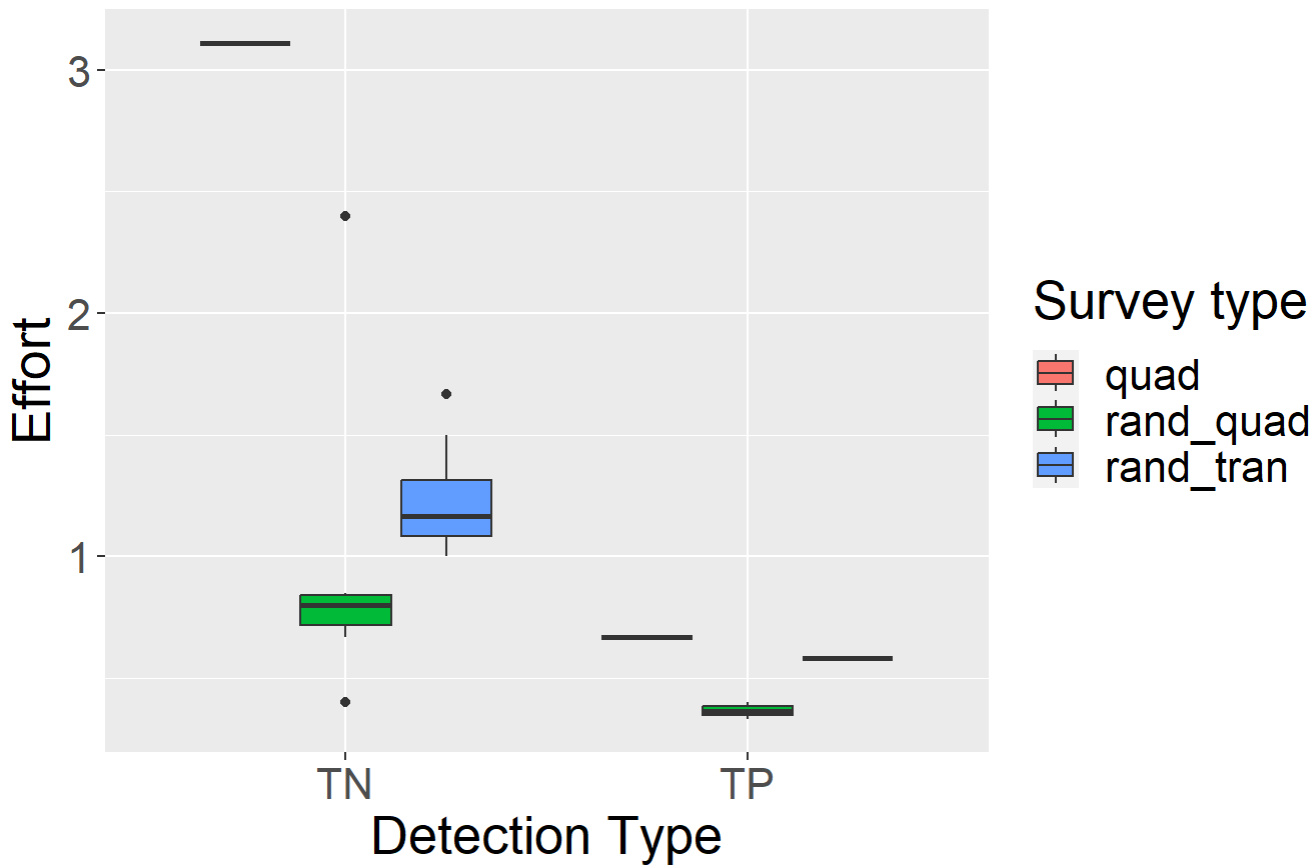
```

Effort expended by survey type



```
detect.plot <- ggplot(rem.FN, aes(detection, effort)) +
  geom_boxplot(aes (fill = survey_type)) +
  theme(text = element_text(size = 20)) +
  labs(title = "Effort expended by detection type", y= "Effort", x = "Detection T
ype") +
  labs(fill = "Survey type")
detect.plot
```

Effort expended by detection type



4. Preliminary models

```
real_data$success = as.integer(real_data$success)
present$success = as.integer(present$success)
rem.FN$success = as.integer(rem.FN$success)
rem.FN$det_levels = as.integer(rem.FN$det_levels)

#is success per search area influenced by effort?
test <- glm(success ~ sweep_effort, family = "binomial", data = real_data)
summary(test)
```

```
##
## Call:
## glm(formula = success ~ sweep_effort, family = "binomial", data = real_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2711  0.2095  0.3682  0.4655  0.7995
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)      0.8535      0.9068      0.941      0.347
## sweep_effort    0.1847      0.1216      1.519      0.129
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 26.417 on 41 degrees of freedom
## Residual deviance: 23.711 on 40 degrees of freedom
## AIC: 27.711
##
## Number of Fisher Scoring iterations: 6
```

```
#is success per search area, when a decoy is present, influenced by effort?
test.p <- glm(success ~ sweep_effort, family = "binomial", data = present)
summary(test.p)
```

```
##
## Call:
## glm(formula = success ~ sweep_effort, family = "binomial", data = present)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6929  -1.1044   0.6859   0.8413   1.3357
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.4940     1.1908   1.255   0.210
## sweep_effort  -0.2085     0.2102  -0.992   0.321
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 14.421 on 10 degrees of freedom
## Residual deviance: 13.369 on 9 degrees of freedom
## AIC: 17.369
##
## Number of Fisher Scoring iterations: 4
```

```
#is success per search area influenced by survey type?
test2 <- glm(success ~ survey_type, family = "binomial", data = real_data)
summary(test2)
```

```
##
## Call:
## glm(formula = success ~ survey_type, family = "binomial", data = real_data)
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -2.4478    0.3203    0.3203    0.3593    0.9005
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.6931     0.8660   0.800   0.4235
## survey_typerand_quad  2.0149     1.3478   1.495   0.1349
## survey_typerand_tran  2.2513     1.3426   1.677   0.0936 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 26.417  on 41  degrees of freedom
## Residual deviance: 23.060  on 39  degrees of freedom
## AIC: 29.06
##
## Number of Fisher Scoring iterations: 5
```

```
#is success per search area influenced by size class when a decoy is present?
test3 <- glm(success ~ size_class, family = "binomial", data = present)
summary(test3)
```

```
##
## Call:
## glm(formula = success ~ size_class, family = "binomial", data = present)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -1.7941   -0.5888    0.6681    0.7843    1.1774
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.6931     1.2247   0.566   0.571
## size_classegg   -0.6931     1.8708  -0.371   0.711
## size_classjuv  -18.2592  3956.1805  -0.005   0.996
## size_classsub    0.6931     1.6583   0.418   0.676
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 14.421  on 10  degrees of freedom
## Residual deviance: 11.596  on   7  degrees of freedom
## AIC: 19.596
```

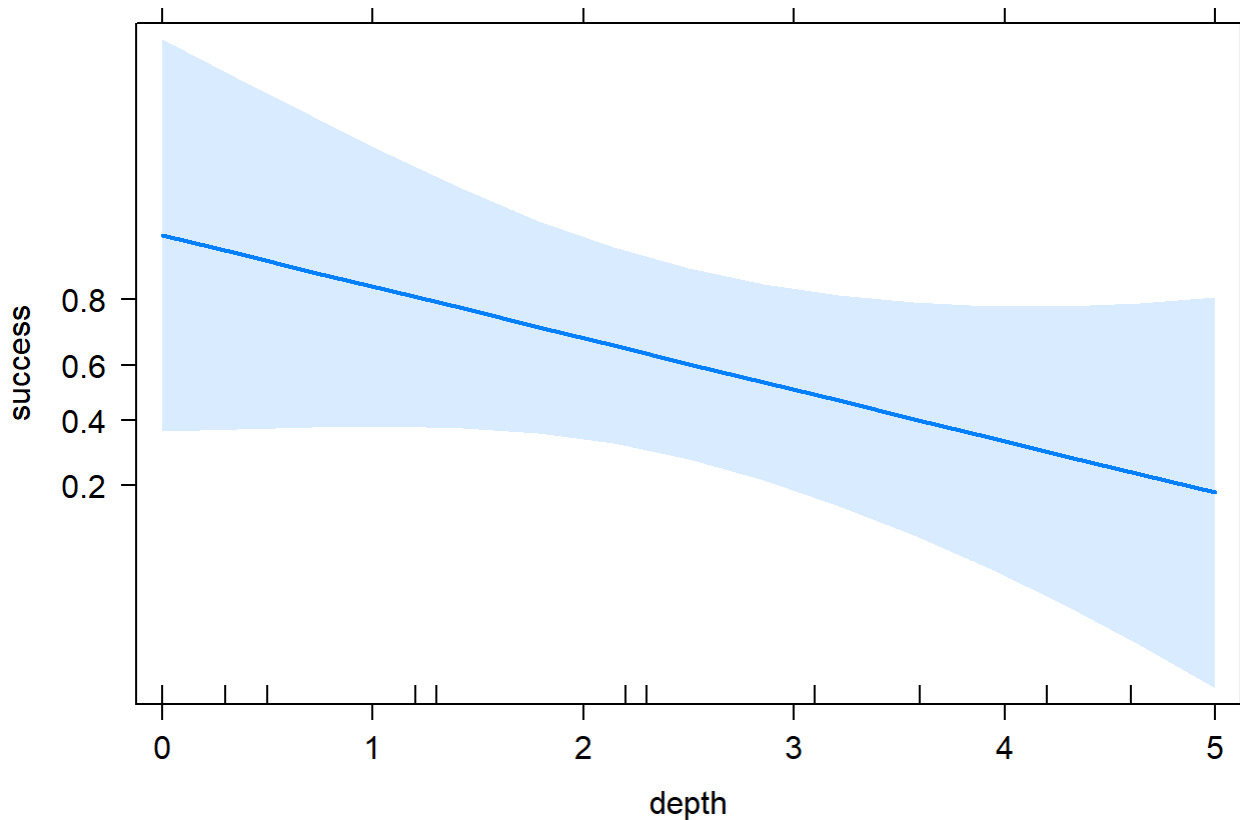
```
##  
## Number of Fisher Scoring iterations: 16
```

```
#is success per search area influenced by depth of the hidden decoy?  
surv.mod <- glm(success ~ depth, family = "binomial", data = present)  
summary(surv.mod)
```

```
##  
## Call:  
## glm(formula = success ~ depth, family = "binomial", data = present)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max  
## -1.7772  -0.7828   0.4772   0.7850   1.3583  
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)    2.3456     1.4867   1.578   0.115  
## depth         -0.7670     0.5201  -1.475   0.140  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
##      Null deviance: 14.421  on 10  degrees of freedom  
## Residual deviance: 11.643  on  9  degrees of freedom  
## AIC: 15.643  
##  
## Number of Fisher Scoring iterations: 4
```

```
plot(allEffects(surv.mod))
```

depth effect plot



```
#is success predicted by density of decoys hidden in the search area?
test.pres_all <- glm(success ~ sa_density, family = "binomial", data = present)
summary(test.pres_all)
```

```
##
## Call:
## glm(formula = success ~ sa_density, family = "binomial", data = present)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4381  -1.4060   0.9374   0.9466   0.9741
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.64254    1.40624   0.457   0.648
## sa_density  -0.04788    0.72475  -0.066   0.947
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 14.421  on 10  degrees of freedom
## Residual deviance: 14.416  on  9  degrees of freedom
```

```
## AIC: 18.416
##
## Number of Fisher Scoring iterations: 4
```

```
# no significant effect of exclosure density on successful searching
dens.mod <- glm(success ~ ex_density, family = "binomial", data = present)
summary(dens.mod)
```

```
##
## Call:
## glm(formula = success ~ ex_density, family = "binomial", data = present)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4823  -1.4006   0.9005   0.9695   0.9695
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.14618     3.28633   0.044   0.965
## ex_density   0.04558     0.35649   0.128   0.898
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 14.421  on 10  degrees of freedom
## Residual deviance: 14.404  on   9  degrees of freedom
## AIC: 18.404
##
## Number of Fisher Scoring iterations: 4
```

```
#is effort determined by sweep number?
eff.mod <- glm(effort ~ sweep_number, data = real_data)
summary(eff.mod)
```

```
##
## Call:
## glm(formula = effort ~ sweep_number, data = real_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.0408  -0.5390  -0.2300   0.2188   3.0461
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    1.39441    0.36986    3.770 0.000528 ***
## sweep_number -0.02027    0.12812   -0.158 0.875063
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.8426485)
##
##      Null deviance: 33.727  on 41  degrees of freedom
## Residual deviance: 33.706  on 40  degrees of freedom
## AIC: 115.95
##
## Number of Fisher Scoring iterations: 2
```

```
#is effort influenced by depth of litter?
eff.depth <- glm(effort ~ depth, data = present)
summary(eff.depth)
```

```
##
## Call:
## glm(formula = effort ~ depth, data = present)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31949  -0.98521  -0.32380   0.03954   2.82835
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.71949     0.79297   2.168  0.0583 .
## depth       -0.06428     0.30375  -0.212  0.8371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.363424)
##
##      Null deviance: 21.377  on 10  degrees of freedom
## Residual deviance: 21.271  on  9  degrees of freedom
## AIC: 44.47
##
## Number of Fisher Scoring iterations: 2
```

```
#is effort influenced by survey type?
effort.type <-glm(effort ~ survey_type, data = real_data)
summary(effort.type)
```

```
##
## Call:
## glm(formula = effort ~ survey_type, data = real_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.96296  -0.41354  -0.05833   0.34245   1.77037
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.6296     0.3072   8.561 1.72e-10 ***
## survey_typerand_quad -1.5244     0.3602  -4.232 0.000136 ***
## survey_typerand_tran -1.4880     0.3502  -4.248 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.566139)
##
##      Null deviance: 33.727  on 41  degrees of freedom
## Residual deviance: 22.079  on 39  degrees of freedom
## AIC: 100.18
##
## Number of Fisher Scoring iterations: 2
```

```
#look at this
#model of searches with decoys present in search area; Much higher occurrence of
successful searches (TN) when exclosure density = 0
test.pres <- glm(success ~ ex_density, family = "binomial", data = real_data)
summary(test.pres)
```

```
##
## Call:
## glm(formula = success ~ ex_density, family = "binomial", data = real_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0682   0.1976   0.2224   0.5007   0.7724
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.9268     1.4189   2.768 0.00565 **
## ex_density    -0.2392     0.1510  -1.584 0.11317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 26.417 on 41 degrees of freedom
## Residual deviance: 22.994 on 40 degrees of freedom
## AIC: 26.994
##
## Number of Fisher Scoring iterations: 6
```

```
#likelihood of success responding to aspect
```

```
aspect.mod <-glm (success ~ aspect, family = "binomial", data = real_data)
summary (aspect.mod)
```

```
##
## Call:
## glm(formula = success ~ aspect, family = "binomial", data = real_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3831  0.2559  0.3246  0.5130  0.9220
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.64634    1.81070  -0.357   0.721
## aspect      0.05154    0.03376   1.527   0.127
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 26.417 on 41 degrees of freedom
## Residual deviance: 23.899 on 40 degrees of freedom
## AIC: 27.899
##
## Number of Fisher Scoring iterations: 5
```

```
#likelihood of success - length + depth of decoys
```

```
sweep.lik <- glm(success ~ length + depth, family = "binomial", data = present)
summary(sweep.lik)
```

```
##
## Call:
## glm(formula = success ~ length + depth, family = "binomial",
##      data = present)
##
```

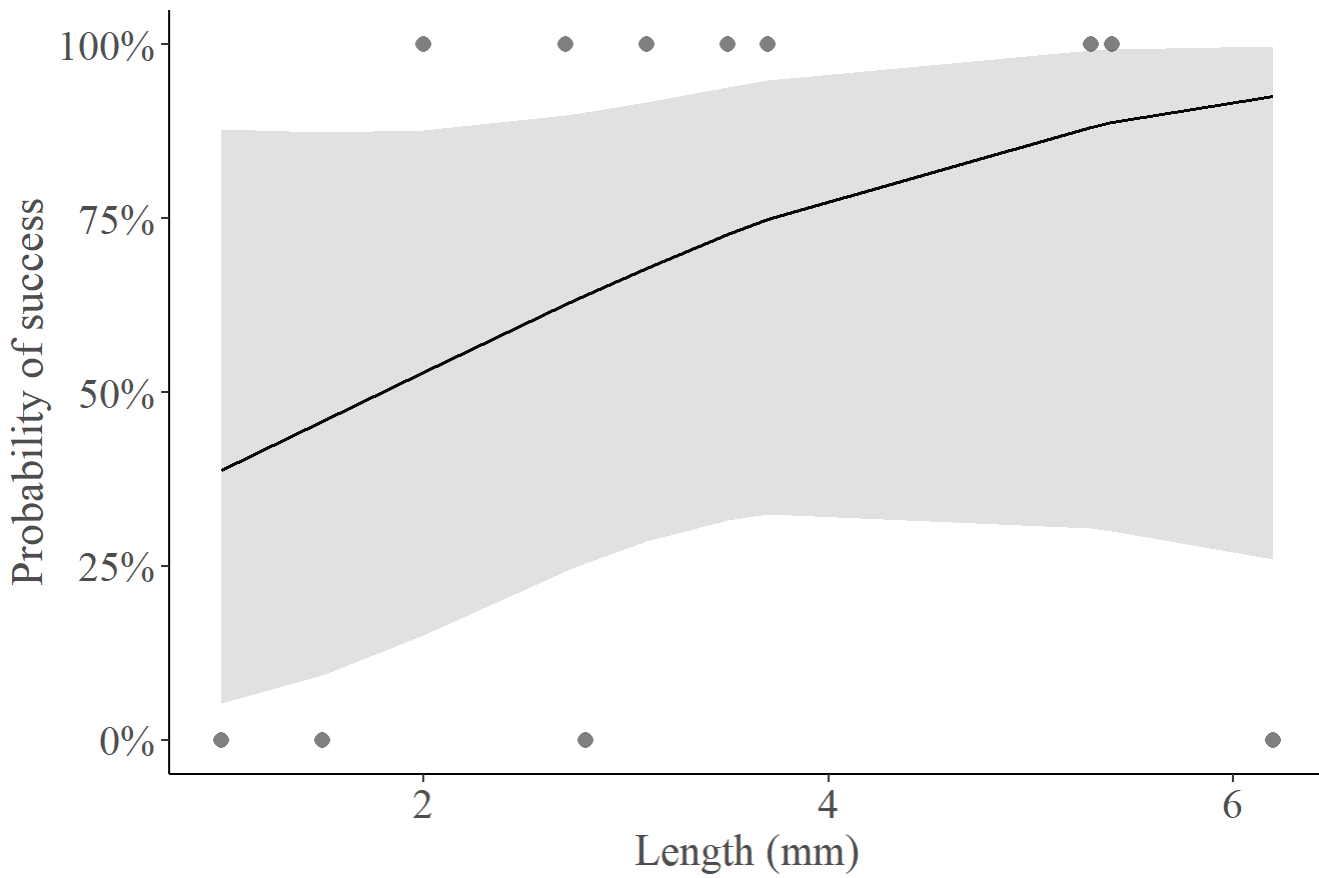
```
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -1.4714   -0.7689    0.3191    0.5929    1.8007
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.1401     1.8822   0.606   0.545
## length         0.5739     0.4876   1.177   0.239
## depth        -1.0247     0.6626  -1.546   0.122
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 14.421  on 10  degrees of freedom
## Residual deviance: 10.061  on   8  degrees of freedom
## AIC: 16.061
##
## Number of Fisher Scoring iterations: 5
```

5. Model visualization

```
set_theme(base = theme_classic(), #To remove the background color and the grids
          theme.font = 'serif',   #To change the font type
          axis.title.size = 1.5,  #To change axis title size
          axis.textsize.x = 1.2,  #To change x axis text size
          axis.textsize.y = 1.2,  #To change y axis text size
          title.size = 2,
          )

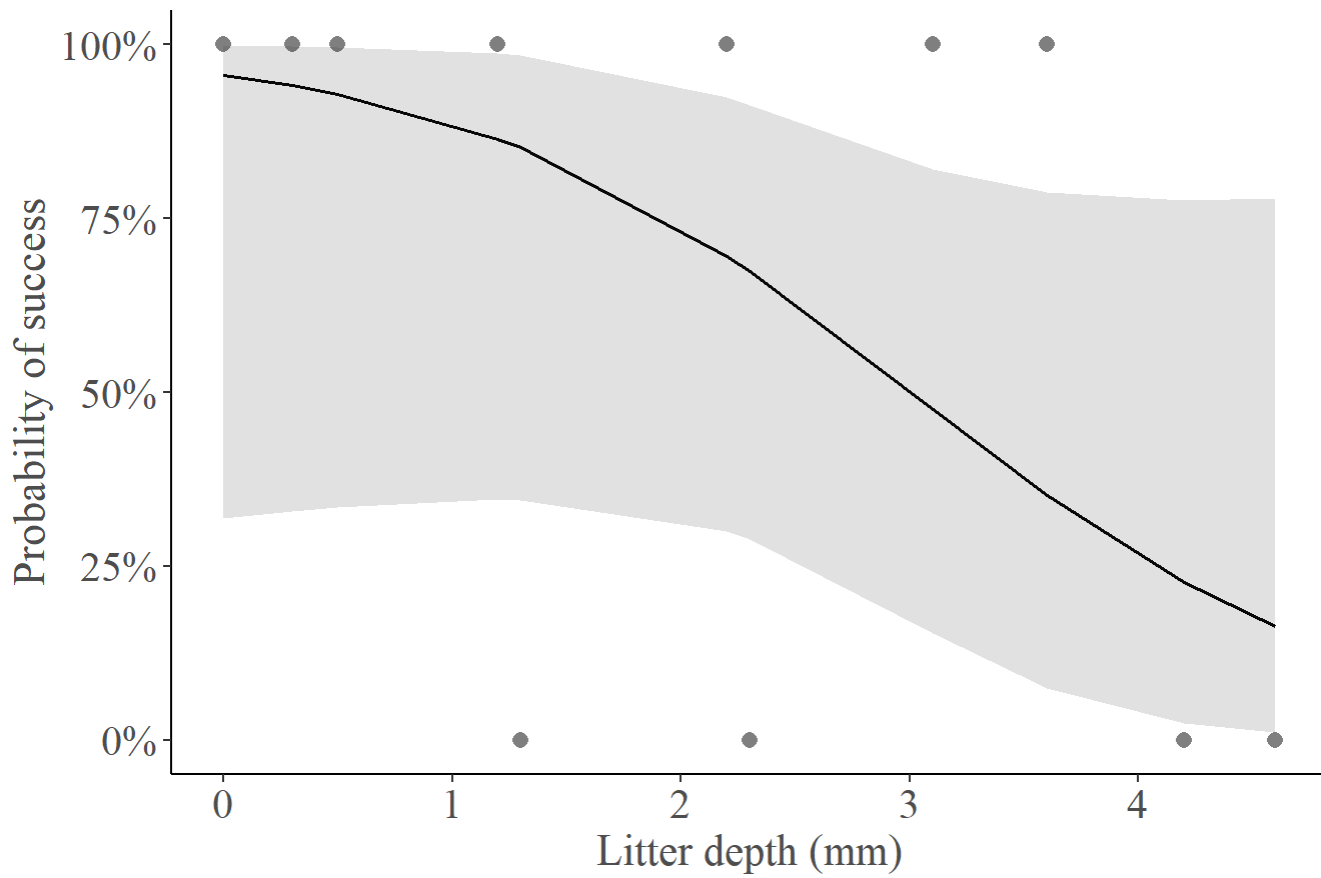
plot_model(sweep.lik, type = "pred", title = "Probability of success per decoy size",
           axis.title = c("Length (mm)", "Probability of success"), terms = "length",
           show.data = TRUE, p.val = "wald")
```

Probability of success per decoy size



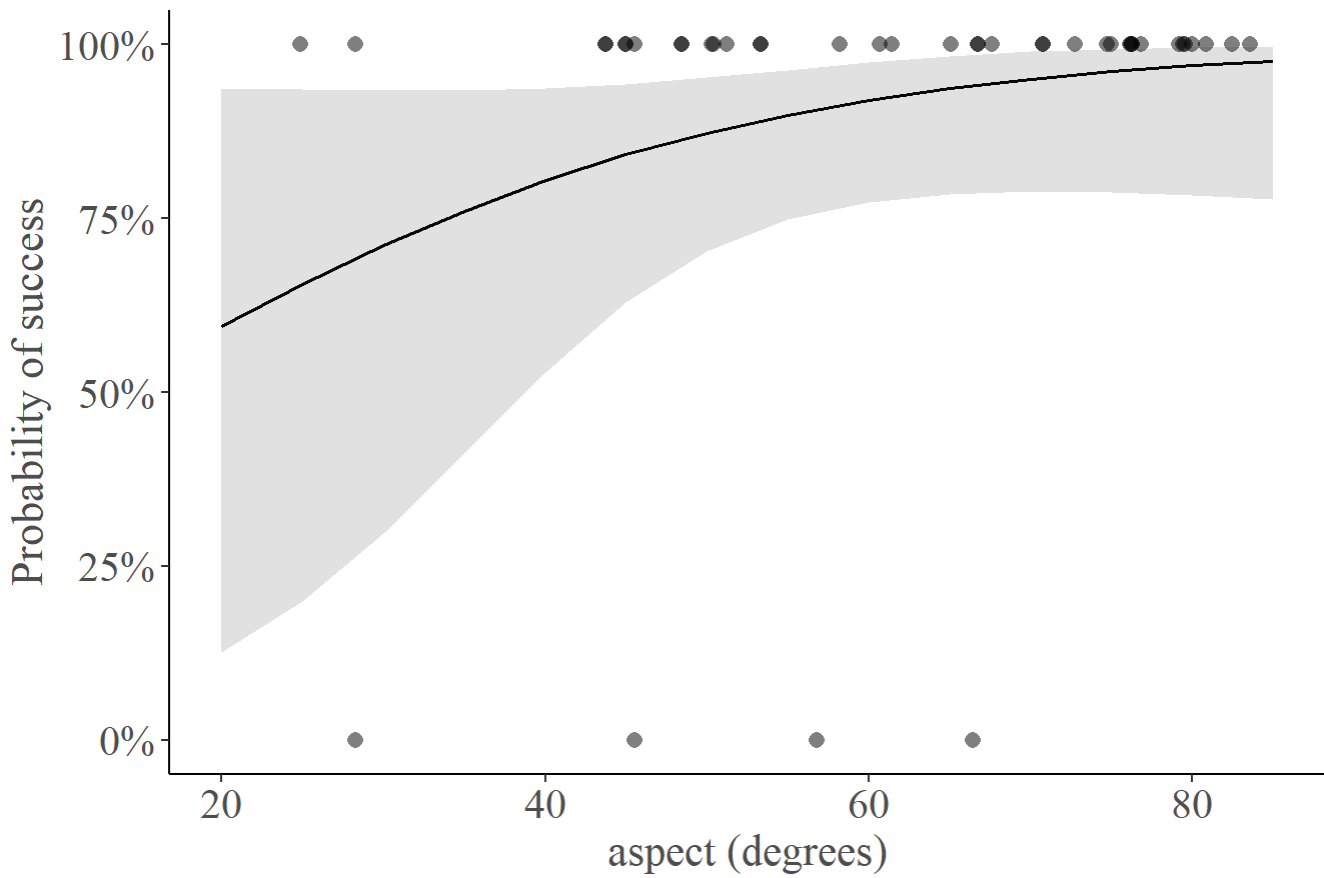
```
plot_model(sweep.lik, type = "pred", title = "Probability of success per litter d  
epth", axis.title = c("Litter depth (mm)", "Probability of success"), terms = "de  
pth", show.data = TRUE, p.val = "wald")
```

Probability of success per litter depth



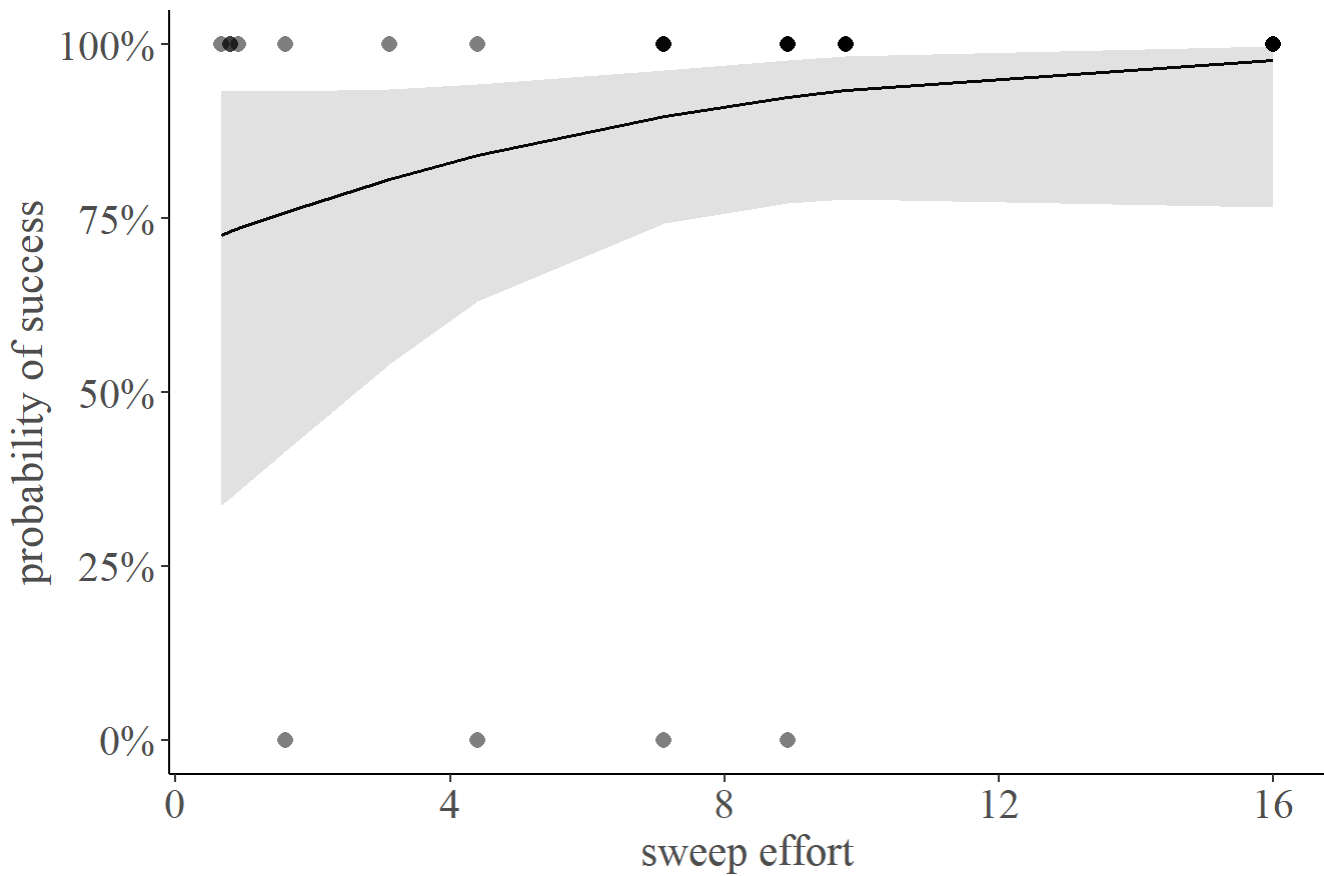
```
plot_model(aspect.mod, type = "pred", title = "Probability of success per aspect",  
axis.title = c("aspect (degrees)", "Probability of success"), terms = "aspect",  
show.data = TRUE, p.val = "wald")
```

Probability of success per aspect



```
#probability of success per survey effort (all searches)
plot_model(test, type = "pred", title = " Effect of total sweep effort on success
",
           axis.title = c("sweep effort", "probability of success"),
           show.data = TRUE, show.intercept = TRUE, terms = "sweep_effort[all]")
```

Effect of total sweep effort on success



```
#probability of success per survey type (all searches)
plot_model(test2, type = "pred", title = " Effect of survey type on success",
  axis.title = c("survey type", "probability of success"),
  show.data = TRUE, show.intercept = TRUE, terms = "survey_type[all]")
```

Effect of survey type on success

