

# Beispiel 4.2

Bernd Gruber

2022-08-29

## Single season Occupancy

Dieses Beispiel verwendet das R-paket “unmarked” fuer eine Demonstration wie die Belegungswahrscheinlichkeit nach MacKenzie berechnet werden kann.

Als erstes muss das unmarked package geladen werden. Der folgende Code, schaut nach, ob es nicht schon installiert ist und installiert es gegebenenfalls.

```
## unmarked  ggplot2
##      TRUE      TRUE
```

Fuer dieses Beispiel verwenden wir Daten von Koalas. **[needs to be explained in more detail]**

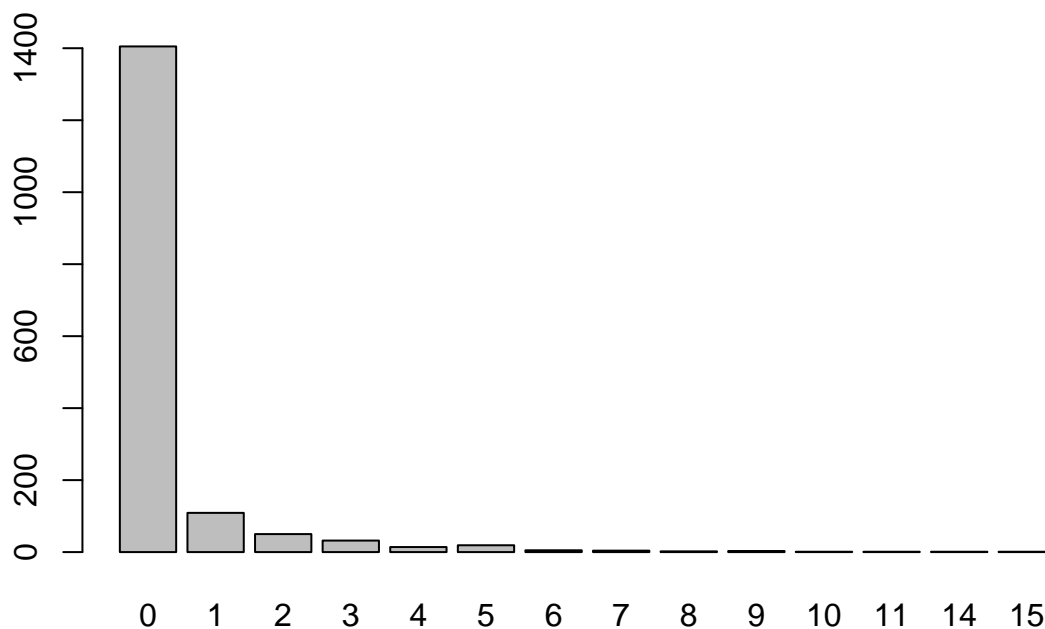
```
fn <- system.file('extdata', "koala_ss_occ.csv", package = "UlmerBuch")
#1647 sites mit 30 Besuchen pro site
koalas <- read.csv(fn, row.names = "site")
```

### 1. Koala an/abwesend

```
dim(koalas)  #
```

```
## [1] 1647  30
```

```
#Summierte Anzahl der Koala-Pellets unter allen 30 Baeumen
barplot(table(rowSums(koalas)))
```



## Aufbereiten der Daten in das unmarked format

Wir muessen die Daten noch umformatieren mittels der `unmarkedFrameOccu` function.

```
sample.koalas <- unmarkedFrameOccu( y = as.matrix(koalas))
```

Unmarked hat eine Summary funktion, welches die Daten uebersichtlich zusammenfasst

```
summary(sample.koalas)
```

```
## unmarkedFrame Object
##
## 1647 sites
## Maximum number of observations per site: 30
## Mean number of observations per site: 30
## Sites with at least one detection: 242
##
## Tabulation of y observations:
##      0      1
## 48803  607
```

## Einfaches Belegungsmodell ohne Kovariable

```
occu.m1 <- occu(formula = ~1 #detection keine Kovariable
                  ~1 #occupancy keine Kovariable
                  , data = sample.koalas)
```

Die normale Zusammenfassung zeigt die Schätzungen auf der logit-scale, was nicht allzusehr weiterhilft.

```
summary(occu.m1)

##
## Call:
## occu(formula = ~1 ~ 1, data = sample.koalas)
##
## Occupancy (logit-scale):
##   Estimate      SE      z    P(>|z|)
##    -1.64 0.0722 -22.7 1.74e-114
##
## Detection (logit-scale):
##   Estimate      SE      z    P(>|z|)
##    -2.5 0.0489 -51.1      0
##
## AIC: 5511.573
## Number of sites: 1647
## optim convergence code: 0
## optim iterations: 34
## Bootstrap iterations: 0
```

Mittels der predict function, koennen wir aber die Belegung in Anwesend/Abwesend umwandeln.

```
predict(occu.m1, newdata = data.frame(site = 1), type = "state")

##   Predicted      SE    lower    upper
## 1 0.1622074 0.00981139 0.1438839 0.1823674
```

Als erstes Ergebniss erfahren wir dass etwa 16% aller site besetzt waren.

Weiterhin moechten wir noch die Nachweiswahrscheinlichkeit p berechnen. Da wir in unserem Model die Nachweiswahrscheinlichkeit als konstant fuer alle Sites angenommen haben, ist es egal fuer welche site wir die Nachweiswahrscheinlichkeit berechnen.

```
predict(occu.m1, newdata = data.frame(site = 1), type = "det")

##   Predicted      SE    lower    upper
## 1 0.07573597 0.003425192 0.06928993 0.08272837
```

## Occupancy mit Kovariablen

```
fn <- system.file('extdata','foodtrees.csv', package = "UlmerBuch")

food <- read.csv(fn, row.names = "site")
summary(food)
```

```
##      foodtrees
## Min.      : 0.00
## 1st Qu.:17.00
## Median :23.00
## Mean    :21.53
## 3rd Qu.:27.00
## Max.    :30.00
```

## Umformatierung in ein Beispiel mit Kovariable

Das Beispiel ist ein einfaches Beispiel (nur eine Kovariable fuer die Sites und keine fuer die Erfassungen (=Observer). Zum Beispiel koennte das Wetter die Nachweiswahrscheinlichkeiten beeinflussen.

```
sample.koalas_cov <- unmarkedFrameOccu( y = as.matrix(koalas), siteCovs = food)
summary(sample.koalas_cov)
```

```
## unmarkedFrame Object
##
## 1647 sites
## Maximum number of observations per site: 30
## Mean number of observations per site: 30
## Sites with at least one detection: 242
##
## Tabulation of y observations:
##      0      1
## 48803   607
##
## Site-level covariates:
##      foodtrees
## Min.      : 0.00
## 1st Qu.:17.00
## Median :23.00
## Mean    :21.53
## 3rd Qu.:27.00
## Max.    :30.00
```

```
occu.m2 <- occu(formula =~1 # Erfassung (=detection)
                ~ foodtrees #Belegung (=occupancy)
                , data =sample.koalas_cov)

# Zusammenfassung
summary(occu.m2)
```

```
##
## Call:
```

```
## occu(formula = ~1 ~ foodtrees, data = sample.koalas_cov)
##
## Occupancy (logit-scale):
##      Estimate      SE      z  P(>|z|)
## (Intercept) -2.4420 0.2685 -9.10 9.39e-20
## foodtrees    0.0363 0.0115  3.16 1.56e-03
##
## Detection (logit-scale):
##      Estimate      SE      z  P(>|z|)
##      -2.5 0.0489 -51.1      0
##
## AIC: 5502.884
## Number of sites: 1647
## optim convergence code: 0
## optim iterations: 54
## Bootstrap iterations: 0
```

In der Zusammenfassung koennen wir den “Erklaerungswert” der Variablen foodtrees erkennen ( $p=1.56e-03 < 0.01$ ), welches es signifikant geschaetzt wird. Fuer die weitere Untersuchgen des Effekts (die Schaetzung des Effekts von 0.0363 bedeutet dass ein positiver Zusammenhang besteht, das heist eine Site ist eher besetzt, je mehr Nahrungsbaeume vorhanden sind). Wie bevor verwenden wir die predict function und tragen die Anzahl foodtrees gegen die Belegungswahrscheinlichkeit auf.

```
predict_m2_food <- predict(occu.m2, newdata = data.frame(foodtrees= 0:30), type="state")
```

```
ggplot(data = predict_m2_food, aes(x = 0:30, y = Predicted)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "gray") +
  stat_smooth(method = "loess", col = "black", se = FALSE) +
  labs(x = "Foodtrees", y = "Belegungswahrscheinlichkeit") +
  theme_classic()
```

