CSSS 510: Lab 6

Multinomial Logit

2017-11-17

0. Agenda

1. Deriving a likelihood function for the multinomial model
2. Fitting a multinomial logit model using optim() and glm()
3. Simulating predicted values and confidence intervals
4. Evaluating goodness of fit

1. Deriving a likelihood function for multinomial logit

Recall from lecture the multinomial logit model:

$$\Pr(y_i = j|\boldsymbol{x_i}) = \frac{\exp(\boldsymbol{x_i}\boldsymbol{\beta_j})}{\sum_{l=1}^{M} \exp(\boldsymbol{x_i}\boldsymbol{\beta_l})}$$

If we make the identifying assumption that $\boldsymbol{\beta_1} = 0$, then we have

For $j = 1$

$$\Pr(y_i = 1|\boldsymbol{x_i}) = \frac{1}{1 + \sum_{l=2}^{M} \exp(\boldsymbol{x_i}\boldsymbol{\beta_l})}$$

For $j > 2$

$$\Pr(y_i = j|\boldsymbol{x_i}) = \frac{\exp(\boldsymbol{x_i}\boldsymbol{\beta_j})}{1 + \sum_{l=2}^{M} \exp(\boldsymbol{x_i}\boldsymbol{\beta_l})}$$

1. Deriving a likelihood function for multinomial logit

$$\Pr(y_i = j|\boldsymbol{x_i}) = \frac{\exp(\boldsymbol{x_i}\boldsymbol{\beta_j})}{\sum_{l=1}^{M} \exp(\boldsymbol{x_i}\boldsymbol{\beta_l})}$$

In many ways, this is the most tractable model we've seen so far.

1. We assume that $\Pr(y_i = j|x)$ is a function of the linear combination $\boldsymbol{x_i}\boldsymbol{\beta_j}$. One key difference between this and previous models is that $\boldsymbol{\beta_j}$ differs for *each outcome*. We therefore have a set of β for each outcome category.

We then take two simple steps.

2. We take the exponential of $\boldsymbol{x_i}\boldsymbol{\beta_j}$ to ensure that the probabilities are nonnegative.
3. We then divide each $\exp(\boldsymbol{x_i}\boldsymbol{\beta_j})$ and divide it by $\sum_{l=2}^{M} \exp(\boldsymbol{x_i}\boldsymbol{\beta_l})$ to make the probabilities sum to 1.

1. Deriving a likelihood function for multinomial logit

Our model looks very similar to logit but with more than two categories. Indeed, multinomial logit can be thought of as a way to estimate separate logits for all categories all at once.

Like ordered probit, the likelihood function for multinomial logit finds the $\boldsymbol{\beta_j}$ that make the observed data most likely. That is, it maximizes the likelihood that y_{ij} falls into category j. All other categories $\neq$ j are irrelevant.

$$\mathcal{L}(\boldsymbol{\beta_2}, ..., \boldsymbol{\beta_m} | \mathbf{y}, \mathbf{X}) = \prod_{i=1}^{N} \prod_{j=1}^{M} p_{ij}^{y_{ij}}$$

$$\mathcal{L}(\boldsymbol{\beta_2}, ..., \boldsymbol{\beta_m} | \mathbf{y}, \mathbf{X}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\frac{\exp(\boldsymbol{x_i} \boldsymbol{\beta_j})}{1 + \sum_{l=2}^{M} \exp(\boldsymbol{x_i} \boldsymbol{\beta_l})} \right]^{y_{ij}}$$

$$\mathcal{L}(\boldsymbol{\beta_2}, ..., \boldsymbol{\beta_m} | \mathbf{y}, \mathbf{X}) = \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log \frac{\exp(\boldsymbol{x_i} \boldsymbol{\beta_j})}{1 + \sum_{l=2}^{M} \exp(\boldsymbol{x_i} \boldsymbol{\beta_l})}$$

1. Deriving a likelihood function for multinomial logit

Our coefficients can be interpreted in a similar way as logit coefficients: β is the change in the log of the odds of one outcome versus another outcome.

Suppose that we have outcome m and outcome n. We will then have a β that corresponds to the change in the log of the odds of outcome m versus outcome n.

$$\log \frac{\Pr(y = m|\mathbf{x}_c, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_M)}{\Pr(y = n|\mathbf{x}_c, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_M)} = \mathbf{x}_c(\boldsymbol{\beta}_m - \boldsymbol{\beta}_n)$$

If the kth covariate increases by 1, then the log of the odds of category m versus category n increases by the *difference* of their coefficients.

2. Fitting a multinomial logit model

```
rm(list=ls())

# Load data and libraries
library(simcf)           # for mlogit simulators
library(tile)            # for graphics
```

```
## Loading required package: grid
```

```
library(RColorBrewer)    # for colors
library(MASS)            # for mvrnorm()
library(nnet)            # for multinom()

# homemade extractor function for multinom() coef's
coef.multinom <- function(x) {
  nlevel <- length(mlogit.result$lev)
  ncoef <- length(mlogit.result$coefnames)
  coef <- x$wts[(ncoef+2):length(x$wts)]
  coef[-((0:(nlevel-2))*(ncoef+1) + 1)]
}

# Load data (in simcf library)
data(gator)
gator <- as.data.frame(cbind(food, size, female))
```

2. Fitting a multinomial logit model

```
# Estimate MNL using the nnet library

model <- food ~ size + female
mlogit.result <- multinom(model, Hess=TRUE)
```

```
## # weights:  12 (6 variable)
## initial  value 64.818125
## iter  10 value 48.292498
## final  value 48.291021
## converged
```

```
pe <- coef(mlogit.result)
vc <- solve(mlogit.result$Hess)       # var-cov matrix
se <- sqrt(diag(vc))
pe
```

```
## [1]  4.8969032 -2.5259623 -0.7899437 -1.9471694  0.1337791  0.3817674
```

```
se
```

```
## 2:(Intercept)        2:size      2:female 3:(Intercept)        3:size
##     1.7063258     0.8475594     0.7122227     1.5305908     0.5181170
##      3:female
##     0.9083499
```

2. Fitting a multinomial logit model

```
gator$foodnm[gator$food==1]<-"Invertebrates"
gator$foodnm[gator$food==2]<-"Fish"
gator$foodnm[gator$food==3]<-"Other"

model <- foodnm ~ size + female
mlogit.result <- multinom(model, data=gator,Hess=TRUE)
```

```
## # weights:  12 (6 variable)
## initial  value 64.818125
## iter  10 value 48.291150
## final  value 48.291021
## converged
```

```
mlogit.result
```

```
## Call:
## multinom(formula = model, data = gator, Hess = TRUE)
##
## Coefficients:
##               (Intercept)     size    female
## Invertebrates   -4.896281 2.525668 0.7897795
## Other           -6.843614 2.659585 1.1714407
##
## Residual Deviance: 96.58204
## AIC: 108.582
```

2. Simulating predicted values and confidence intervals

```r
# Simulate parameters from predictive distributions
sims <- 10000
simbetas <- mvrnorm(sims,pe,vc)       # draw parameters, using MASS::mvrnorm
simB <- array(NA, dim = c(sims,3,2))  # re-arrange simulates to array format
simB[,,1] <- simbetas[,1:3]           #   for MNL simulation
simB[,,2] <- simbetas[,4:6]

##############################################
# Create full factorial set of counterfactuals
sizerange <- seq(1,4,by=0.1)          # range of counterfactual sizes
femalerange <- c(0,1)                 # range of counterfactual sexes
xhyp1 <- cfFactorial(size = sizerange, female = femalerange)

# Simulate expected probabilities with 68% CI
mlogit.ev1 <- mlogitsimev(xhyp1, simB, ci=0.68)
```

2. Simulating predicted values and confidence intervals

```
#######################################################################
# Plot expected values for all combinations of size and sex in 2 plots

# Get 3 colors
cols <- brewer.pal(3,"Set1")

# Create one trace for each predicted category of the response, and each sex
trace1a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==0],
                    y=mlogit.ev1$pe[xhyp1$x$female==0,1],
                    lower=mlogit.ev1$lower[xhyp1$x$female==0,1,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==0,1,],
                    ci=list(mark="shaded"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==0,],
                                     omit.extrapolated=FALSE),
                    col=cols[1],
                    plot=1
)

trace2a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==0],
                    y=mlogit.ev1$pe[xhyp1$x$female==0,2],
                    lower=mlogit.ev1$lower[xhyp1$x$female==0,2,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==0,2,],
                    ci=list(mark="shaded"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==0,],
                                     omit.extrapolated=FALSE),
                    col=cols[3],
                    plot=1
)
```

2. Simulating predicted values and confidence intervals

```
trace3a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==0],
                    y=mlogit.ev1$pe[xhyp1$x$female==0,3],
                    lower=mlogit.ev1$lower[xhyp1$x$female==0,3,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==0,3,],
                    ci=list(mark="shaded"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==0,],
                                     omit.extrapolated=FALSE),
                    col=cols[2],
                    plot=1
)

trace4a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==1],
                    y=mlogit.ev1$pe[xhyp1$x$female==1,1],
                    lower=mlogit.ev1$lower[xhyp1$x$female==1,1,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==1,1,],
                    ci=list(mark="shaded"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==1,],
                                     omit.extrapolated=FALSE),
                    col=cols[1],
                    plot=2
)
```

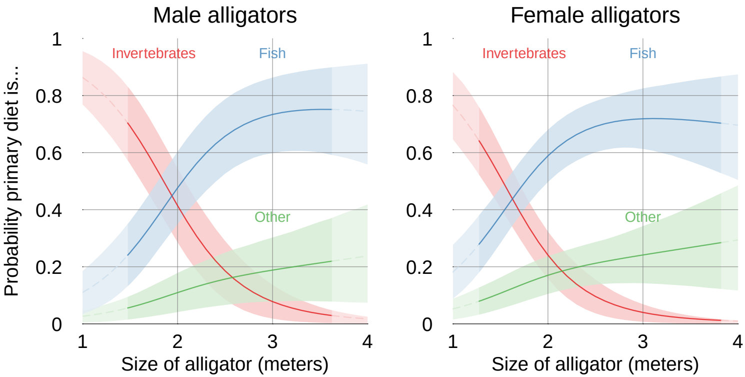
2. Simulating predicted values and confidence intervals

```
trace5a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==1],
                    y=mlogit.ev1$pe[xhyp1$x$female==1,2],
                    lower=mlogit.ev1$lower[xhyp1$x$female==1,2,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==1,2,],
                    ci=list(mark="shaded"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==1,],
                                     omit.extrapolated=FALSE),
                    col=cols[3],
                    plot=2
)

trace6a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==1],
                    y=mlogit.ev1$pe[xhyp1$x$female==1,3],
                    lower=mlogit.ev1$lower[xhyp1$x$female==1,3,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==1,3,],
                    ci=list(mark="shaded"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==1,],
                                     omit.extrapolated=FALSE),
                    col=cols[2],
                    plot=2
)
```

2. Simulating predicted values and confidence intervals

```
linelabels <- textTile(labels=c("Invertebrates",
                                "Fish",
                                "Other"),
                       x=  c(1.75,      3,         3),
                       y=  c(0.95,     0.95,      0.375),
                       col=c(cols[1], cols[2], cols[3]),
                       cex = 0.75,
                       plot=c(1,2)
)


at.x <- c(1,2,3,4)
at.y <- c(0,0.2,0.4,0.6,0.8,1)


# Plot traces using tile
file <- "gatorsEV"
tile(trace1a,
     trace2a,
     trace3a,
     trace4a,
     trace5a,
     trace6a,
     linelabels,
     RxC = c(1,2),
     limits = c(1,4,0,1),
     output = list(outfile=file, width=7),
     xaxis = list(at=at.x),
     yaxis = list(at=at.y, major=FALSE),
     xaxistitle = list(labels="Size of alligator (meters)"),
     yaxistitle = list(type="first", labels="Probability primary diet is...", x=0.1),
     plottitle = list(labels=c("Male alligators","Female alligators"), y=1),
     gridlines = list(type="xy")
)
```

2. Simulating predicted values and confidence intervals



2. Simulating predicted values and confidence intervals

```
##################################################################
## Alternate version with 95% CIs and one plot per trace (6 plots)

# Simulate expected probabilities with 95% CI
mlogit.ev1 <- mlogitsimev(xhyp1, simB, ci=0.95)

# Create one trace for each predicted category of the response, and each sex
trace1a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==0],
                    y=mlogit.ev1$pe[xhyp1$x$female==0,1],
                    lower=mlogit.ev1$lower[xhyp1$x$female==0,1,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==0,1,],
                    ci=list(mark=c("shaded", "dashed")),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==0,],
                                     omit.extrapolated=TRUE),
                    plot=1
)

trace2a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==0],
                    y=mlogit.ev1$pe[xhyp1$x$female==0,2],
                    lower=mlogit.ev1$lower[xhyp1$x$female==0,2,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==0,2,],
                    ci=list(mark="shaded", "dashed"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==0,],
                                     omit.extrapolated=TRUE),
                    plot=3
)
```

2. Simulating predicted values and confidence intervals

```r
trace3a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==0],
                    y=mlogit.ev1$pe[xhyp1$x$female==0,3],
                    lower=mlogit.ev1$lower[xhyp1$x$female==0,3,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==0,3,],
                    ci=list(mark="shaded", "dashed"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==0,],
                                     omit.extrapolated=TRUE),
                    plot=2
)

trace4a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==1],
                    y=mlogit.ev1$pe[xhyp1$x$female==1,1],
                    lower=mlogit.ev1$lower[xhyp1$x$female==1,1,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==1,1,],
                    ci=list(mark="shaded", "dashed"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==1,],
                                     omit.extrapolated=TRUE),
                    plot=4
)
```

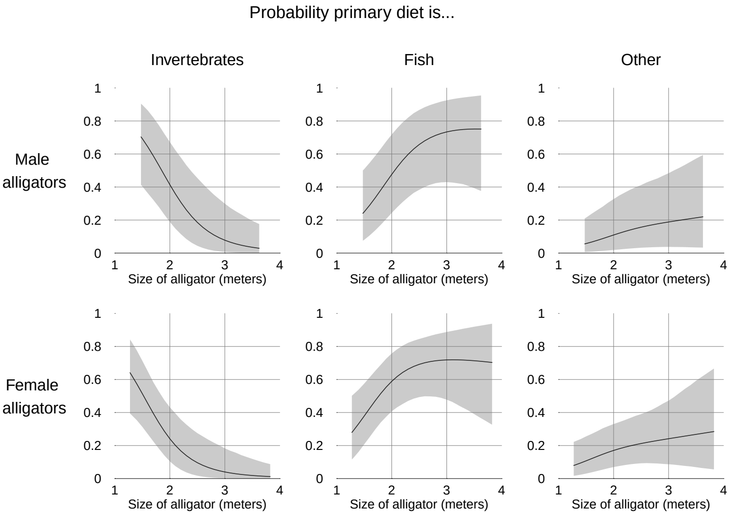
2. Simulating predicted values and confidence intervals

```
trace5a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==1],
                    y=mlogit.ev1$pe[xhyp1$x$female==1,2],
                    lower=mlogit.ev1$lower[xhyp1$x$female==1,2,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==1,2,],
                    ci=list(mark="shaded", "dashed"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==1,],
                                     omit.extrapolated=TRUE),
                    plot=6
)

trace6a <- lineplot(x=xhyp1$x$size[xhyp1$x$female==1],
                    y=mlogit.ev1$pe[xhyp1$x$female==1,3],
                    lower=mlogit.ev1$lower[xhyp1$x$female==1,3,],
                    upper=mlogit.ev1$upper[xhyp1$x$female==1,3,],
                    ci=list(mark="shaded", "dashed"),
                    extrapolate=list(data=cbind(size,female),
                                     cfact=xhyp1$x[xhyp1$x$female==1,],
                                     omit.extrapolated=TRUE),
                    plot=5
)
```

2. Simulating predicted values and confidence intervals

```r
# Plot traces using tile
file <- "gatorsEVsep95"
tile(trace1a,
     trace2a,
     trace3a,
     trace4a,
     trace5a,
     trace6a,
     RxC = c(2,3),
     limits = c(1,4,0,1),
     output = list(file=file, width=10),
     xaxis = list(at=at.x),
     yaxis = list(at=at.y, major=FALSE),
     xaxistitle = list(labels="Size of alligator (meters)"),
     maintitle = list(labels="Probability primary diet is..."),
     rowtitle = list(labels=c("Male\n alligators", "Female\n alligators"), cex=1.25),
     columntitle = list(labels=c("Invertebrates", "Fish", "Other"), cex=1.25),
     height=list(columntitle=5),
     width=list(rowtitle=1.5),
     gridlines = list(type="xy")
)
```

2. Simulating predicted values and confidence intervals



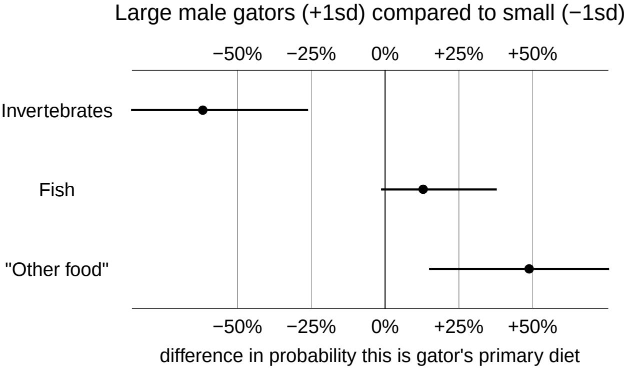
2. Simulating predicted values and confidence intervals

```r
##################################################################
# Create a specific comparison for first diffs and relative risks
# Load data (in simcf library)
data(gator)
gator <- as.data.frame(cbind(food, size, female))
model <- food ~ size + female

xhyp2 <- cfMake(model, gator, nscen=1)
# Scenario 1: Large vs small (male mu + 1sd vs male mu - 1sd), holding male fixed
xhyp2 <- cfName(xhyp2, "Large vs Small Males", scen=1)
xhyp2 <- cfChange(xhyp2, "size",
                  x=mean(gator$size[gator$female==0]) + sd(gator$size[gator$female==0]),
                  xpre=mean(gator$size[gator$female==0]) - sd(gator$size[gator$female==0]),
                  scen=1)
xhyp2 <- cfChange(xhyp2, "female", x=0, xpre=0, scen=1)

# Simulate first differences with 95\% CI
mlogit.fd1 <- mlogitsimfd(xhyp2, simB, ci=0.95)

# Simulate relative risks with 95\% CI
mlogit.rr1 <- mlogitsimrr(xhyp2, simB, ci=0.95)
```

2. Simulating predicted values and confidence intervals

```r
###########################################
# Make ropeladder plot of First Differences

# Make trace of FDs, large vs small males, all categories
traceFD <- ropeladder(x=mlogit.fd1$pe,
                      lower=mlogit.fd1$lower,
                      upper=mlogit.fd1$upper,
                      labels=c("Invertebrates",
                               "Fish",
                               "\"Other food\""),
                      size=0.65,
                      lex=1.75,
                      lineend="square",
                      plot=1
)

# Make reference line trace for first diffs (at 0)
vertmarkFD <- linesTile(x=c(0,0), y=c(0,1), plot=1)

# Set tick marks for x axis
xat <- c(-0.5,-0.25,0, 0.25, 0.5)
xlab <- c("-50%", "-25%", "0%", "+25%", "+50%")

# Make plot with tile
file <- "gatorsFD"
tile(traceFD, vertmarkFD,
     xaxis=list(at=xat, labels=xlab),
     topaxis=list(add=TRUE, at=xat, labels=xlab),
     plottitle=list(labels="Large male gators (+1sd) compared to small (-1sd)"),
     xaxistitle=list(labels="difference in probability this is gator's primary diet"),
     width=list(null=4),
     height=list(xaxistitle=3, plottitle=4),
     gridlines=list(type="xt"),
     output=list(file=file, width=7)
)
```

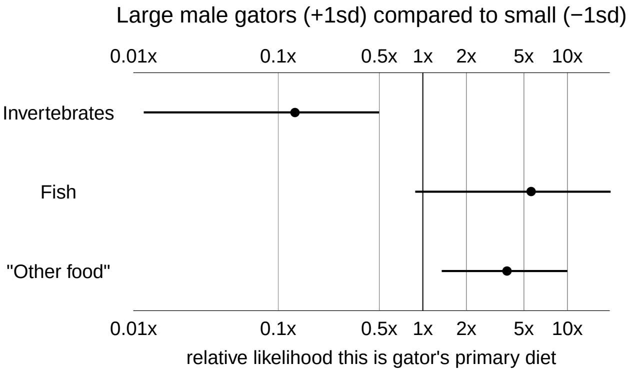
2. Simulating predicted values and confidence intervals



2. Simulating predicted values and confidence intervals

```
##########################################
# Make ropeladder plot of Relative Risks

# Make trace of RRs, large vs small males, all categories
traceRR <- ropeladder(x=mlogit.rr1$pe,
                      lower=mlogit.rr1$lower,
                      upper=mlogit.rr1$upper,
                      labels=c("Invertebrates",
                               "Fish",
                               "\"Other food\""),
                      size=0.65,
                      lex=1.75,
                      lineend="square",
                      plot=1
)

# Make reference line trace for relative risks (at 1)
vertmarkRR <- linesTile(x=c(1,1), y=c(0,1), plot=1)

# Set tick marks for x axis
xat <- c(0.01, 0.1, 0.5, 1, 2, 5, 10 )

# Make plot with tile
file <- "gatorsRR"
tile(traceRR, vertmarkRR,
     xaxis=list(log=TRUE, at=xat, labels=paste0(xat,"x")),
     topaxis=list(add=TRUE, log=TRUE, at=xat, labels=paste0(xat,"x")),
     plottitle=list(labels="Large male gators (+1sd) compared to small (-1sd)"),
     xaxistitle=list(labels="relative likelihood this is gator's primary diet"),
     width=list(null=4),
     height=list(xaxistitle=3, plottitle=4),
     gridlines=list(type="xt"),
     output=list(file=file, width=7)
)
```

2. Simulating predicted values and confidence intervals



2. Simulating predicted values and confidence intervals

```
################################################################
# Create several comparisons for first diffs and relative risks
xhyp3 <- cfMake(model, gator, nscen=5)

# Scenario 1: Large vs small (male mu + 1sd vs male mu - 1sd), holding male fixed
xhyp3 <- cfName(xhyp3, "Large Male (Small Male)", scen=1)
xhyp3 <- cfChange(xhyp3, "size",
                  x=mean(gator$size[gator$female==0]) + sd(gator$size[gator$female==0]),
                  xpre=mean(gator$size[gator$female==0]) - sd(gator$size[gator$female==0]),
                  scen=1)
xhyp3 <- cfChange(xhyp3, "female", x=0, xpre=0, scen=1)

# Scenario 2: Large vs small (female mu + 1sd vs female mu - 1sd), holding female fixed
xhyp3 <- cfName(xhyp3, "Large Female (Small Female)", scen=2)
xhyp3 <- cfChange(xhyp3, "size",
                  x=mean(gator$size[gator$female==1]) + sd(gator$size[gator$female==1]),
                  xpre=mean(gator$size[gator$female==1]) - sd(gator$size[gator$female==1]),
                  scen=2)
xhyp3 <- cfChange(xhyp3, "female", x=1, xpre=1, scen=2)

# Scenario 3: Female vs male holding size fixed at mean of all gators
xhyp3 <- cfName(xhyp3, "Average Female (Average Male)", scen=3)
xhyp3 <- cfChange(xhyp3, "size",
                  x=mean(gator$size),
                  xpre=mean(gator$size),
                  scen=3)
xhyp3 <- cfChange(xhyp3, "female", x=1, xpre=0, scen=3)
```

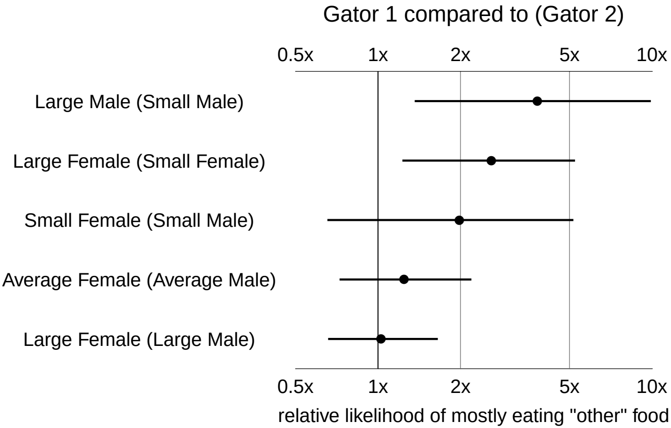
2. Simulating predicted values and confidence intervals

```r
# Scenario 4: Female vs male holding size fixed at high of all gators
xhyp3 <- cfName(xhyp3, "Large Female (Large Male)", scen=4)
xhyp3 <- cfChange(xhyp3, "size",
                  x=mean(gator$size) + sd(gator$size),
                  xpre=mean(gator$size) + sd(gator$size),
                  scen=4)
xhyp3 <- cfChange(xhyp3, "female", x=1, xpre=0, scen=4)

# Scenario 5: Female vs male holding size fixed at low of all gators
xhyp3 <- cfName(xhyp3, "Small Female (Small Male)", scen=5)
xhyp3 <- cfChange(xhyp3, "size",
                  x=mean(gator$size) - sd(gator$size),
                  xpre=mean(gator$size) - sd(gator$size),
                  scen=5)
xhyp3 <- cfChange(xhyp3, "female", x=1, xpre=0, scen=5)


# Simulate first differences with 95\% CI
mlogit.fd2 <- mlogitsimfd(xhyp3, simB, ci=0.95)

# Simulate relative risks with 95\% CI
mlogit.rr2 <- mlogitsimrr(xhyp3, simB, ci=0.95)

sorted <- rev(order(mlogit.rr2$pe[,3]))

# Make trace of RRs, large vs small males, all categories
traceRR2 <- ropeladder(x=mlogit.rr2$pe[sorted,3],
                       lower=mlogit.rr2$lower[sorted,3,1],
                       upper=mlogit.rr2$upper[sorted,3,1],
                       labels=rownames(xhyp3$x)[sorted],
                       size=0.65,
                       lex=1.75,
                       lineend="square",
                       plot=1
)
```

2. Simulating predicted values and confidence intervals

```r
# Make reference line trace for relative risks (at 1)
vertmarkRR2 <- linesTile(x=c(1,1), y=c(0,1), plot=1)

# Set tick marks for x axis
xat <- c(0.5, 1, 2, 5, 10 )

# Make plot with tile
file <- "gatorsRRmultiscen"
tile(traceRR2, vertmarkRR2,
     xaxis=list(log=TRUE, at=xat, labels=paste0(xat,"x")),
     topaxis=list(add=TRUE, log=TRUE, at=xat, labels=paste0(xat,"x")),
     plottitle=list(labels="Gator 1 compared to (Gator 2)"),
     xaxistitle=list(labels="relative likelihood of mostly eating \"other\" food"),
     width=list(null=4),
     height=list(xaxistitle=3, plottitle=4),
     gridlines=list(type="xt"),
     output=list(file=file, width=7)
)
```

2. Simulating predicted values and confidence intervals



2. Evaluating Goodness of Fit

```r
#########################################
## A cross-validated goodness of fit test

## A simple leave-one-out cross-validation function for multinom; returns predicted probs
loocv <- function (obj, model, data) {
  ncat <- length(obj$lev)
  m <- nrow(data)
  form <- model
  loo <- matrix(NA, nrow=m, ncol=ncat)
  for (i in 1:m) {
    i.mlogit <- multinom(model, data=data[-i,])
    loo[i,] <- predict(i.mlogit, newdata = data[i,], type="probs")
  }
  loo
}

predIS <- predict(mlogit.result, type="probs")
predCV <- loocv(mlogit.result, model, gator)

ncat <- 3
predIScat <- apply(predIS, 1, function(x, ncat) order(x)[ncat], ncat=ncat)
predCVcat <- apply(predCV, 1, function(x, ncat) order(x)[ncat], ncat=ncat)

pcpIS <- mean(predIScat==gator$food)
pcpCV <- mean(predCVcat==gator$food)
```

2. Evaluating Goodness of Fit

```
pcpIS
```

```
## [1] 0.2372881
```

```
pcpCV
```

```
## [1] 0.559322
```