

Statistical Details of the Lizard Data Analysis

March 14, 2017

Initial Setup

In the hidden R code chunk, I load (and install, if not already installed) needed packages and set up working directory. The working directory should have a sub-folder named `Data` for raw data file(s) and a folder named `Figures` for exporting figures.

Once the data are imported, we change the column names for the six categories:

```
> lizards <- read.csv(paste(dataDIR, "lizard light data_better bins.csv", sep = "/"))
>
> names(lizards)[5:10] <- c("Buried", "Sheltered", "Shade", "PtShade", "PtSun",
+   "FullSun")
> ## lizards$HomeElev <- ordered(lizards$HomeElev,
> ## levels=rev(levels(lizards$HomeElev)))
```

Fitting Multinomial Regression

The function `multinom` from package `nnet` is used. The function takes a matrix as the response variable.

```
> PID <- c(5:10)
> multinom.better <- multinom(as.matrix(lizards[, PID]) ~ Sex * HomeElev * Treatment,
+   data = lizards)
```

```
## # weights:  54 (40 variable)
## initial  value 93209.119349
## iter   10 value 84362.181199
## iter   20 value 84148.116584
## iter   30 value 83957.084387
## iter   40 value 83861.702132
## final   value 83853.974279
## converged
```

```
> ## print summary
> summary(multinom.better, corr = FALSE)
```

```
## Call:
## multinom(formula = as.matrix(lizards[, PID]) ~ Sex * HomeElev *
##   Treatment, data = lizards)
##
## Coefficients:
##              (Intercept)           SexM HomeElevlow TreatmentTrans
## Sheltered -0.3044227 -0.05920824 -0.7960815   -0.6641483
## Shade     -1.5063763  0.19036682 -0.6628925   -0.5026080
## PtShade    1.1955848 -0.65360111 -0.6269809   -1.1013122
## PtSun      0.8233942  0.01692115 -0.6861753   -0.7248518
## FullSun    0.8084113 -0.06180556 -1.1759609   -1.1871881
##
##           SexM:HomeElevlow SexM:TreatmentTrans HomeElevlow:TreatmentTrans
## Sheltered      0.09149641          1.269461          2.113318
## Shade          0.08102729          1.293035          1.212780
```

```
## PtShade          0.41076611          2.511661          1.341389
## PtSun            -0.42729978          1.668673          1.366047
## FullSun         -0.54778982          1.392339          2.391675
## SexM:HomeElevlow:TreatmentTrans
## Sheltered              -1.172025
## Shade                 -1.303312
## PtShade              -2.491693
## PtSun                -1.701519
## FullSun              -1.262198
##
## Std. Errors:
## (Intercept)      SexM HomeElevlow TreatmentTrans
## Sheltered  0.07300650 0.1872646  0.08292164  0.09044522
## Shade      0.11165503 0.2648229  0.12739986  0.13819886
## PtShade    0.05428524 0.1491122  0.05959387  0.06667070
## PtSun      0.05705709 0.1439222  0.06307101  0.06892435
## FullSun    0.05718831 0.1457588  0.06491436  0.07211340
## SexM:HomeElevlow SexM:TreatmentTrans HomeElevlow:TreatmentTrans
## Sheltered      0.1962104          0.3450624          0.10560522
## Shade          0.2785539          0.4462069          0.16521459
## PtShade        0.1538469          0.2750247          0.07910790
## PtSun          0.1501708          0.2744021          0.08181067
## FullSun        0.1547623          0.2935605          0.08568472
## SexM:HomeElevlow:TreatmentTrans
## Sheltered              0.3540958
## Shade                 0.4635524
## PtShade              0.2822835
## PtSun                0.2826648
## FullSun              0.3026759
##
## Residual Deviance: 167707.9
## AIC: 167787.9
```

The fitted model uses the first category (Buried) as the baseline. Statistical significance of the estimated coefficients is determined by comparing the estimated values and their respective estimation standard error. When the standard error is less than half of the absolute value of the respective coefficients, the coefficients are statistically different from 0 at roughly a significance level of 0.05. In a multinational regression, an effect is “statistically significant” if at least one category-specific coefficient is different from 0. That is, when examine the summary table, we check the five values under each column (an effect) and their respective standard error; if at least one of the five values is different from 0, the effect represented by the column is deemed “statistically significant.”

However, the statistical significance concept, in this case, is meaningless because the coefficients are in log-odds ratio scale. When the baseline category is changed, the coefficients and their uncertainty will be changed, too. For example, we can use **Sheltered** category as the baseline:

```
> PID2 <- c(6, 5, 7:10)
> multinom.better2 <- multinom(as.matrix(lizards[, PID2]) ~ Sex * HomeElev * Treatment,
+ data = lizards)
```

```
## # weights: 54 (40 variable)
## initial value 93209.119349
## iter 10 value 84558.863744
## iter 20 value 84395.859846
## iter 30 value 84096.311035
## iter 40 value 83869.825966
```

```

## final value 83853.974319
## converged
> ## print summary
> summary(multinom.better2, corr = FALSE)

## Call:
## multinom(formula = as.matrix(lizards[, PID2]) ~ Sex * HomeElev *
## Treatment, data = lizards)
##
## Coefficients:
## (Intercept) SexM HomeElevlow TreatmentTrans
## Buried 0.3043563 0.059511984 0.7961447 0.66419181
## Shade -1.2019966 0.249926871 0.1333044 0.16164601
## PtShade 1.4998828 -0.593959515 0.1691981 -0.43709040
## PtSun 1.1277358 0.076306268 0.1099698 -0.06068318
## FullSun 1.1127666 -0.002556308 -0.3797959 -0.52298159
## SexM:HomeElevlow SexM:TreatmentTrans HomeElevlow:TreatmentTrans
## Buried -0.09181590 -1.26977693 -2.1133550
## Shade -0.01085723 0.02351407 -0.9006370
## PtShade 0.31884205 1.24146759 -0.7719997
## PtSun -0.51899329 0.39880110 -0.7472792
## FullSun -0.63950608 0.12260296 0.2782888
## SexM:HomeElevlow:TreatmentTrans
## Buried 1.17233838
## Shade -0.13111836
## PtShade -1.31893360
## PtSun -0.52905914
## FullSun -0.08973009
##
## Std. Errors:
## (Intercept) SexM HomeElevlow TreatmentTrans SexM:HomeElevlow
## Buried 0.07300445 0.1872660 0.08291975 0.09044316 0.1962116
## Shade 0.11520086 0.2761838 0.13345016 0.14554870 0.2922888
## PtShade 0.06125135 0.1684881 0.07162669 0.08081770 0.1775311
## PtSun 0.06372065 0.1639171 0.07454444 0.08268652 0.1743592
## FullSun 0.06383807 0.1655341 0.07611021 0.08536261 0.1783314
## SexM:TreatmentTrans HomeElevlow:TreatmentTrans
## Buried 0.3450355 0.10560335
## Shade 0.4435239 0.17294152
## PtShade 0.2707525 0.09419434
## PtSun 0.2701220 0.09647522
## FullSun 0.2895641 0.09978111
## SexM:HomeElevlow:TreatmentTrans
## Buried 0.3540696
## Shade 0.4621670
## PtShade 0.2801040
## PtSun 0.2804903
## FullSun 0.3006467
##
## Residual Deviance: 167707.9
## AIC: 167787.9

```

The estimated coefficients and their standard errors are different from the same of the previous model. Mathematically, these two models are identical with respect to the multinational model parameters (the

proportions of time a lizard spends in each of the six categories). Consequently, we need to derive and present these proportions.

Model Prediction

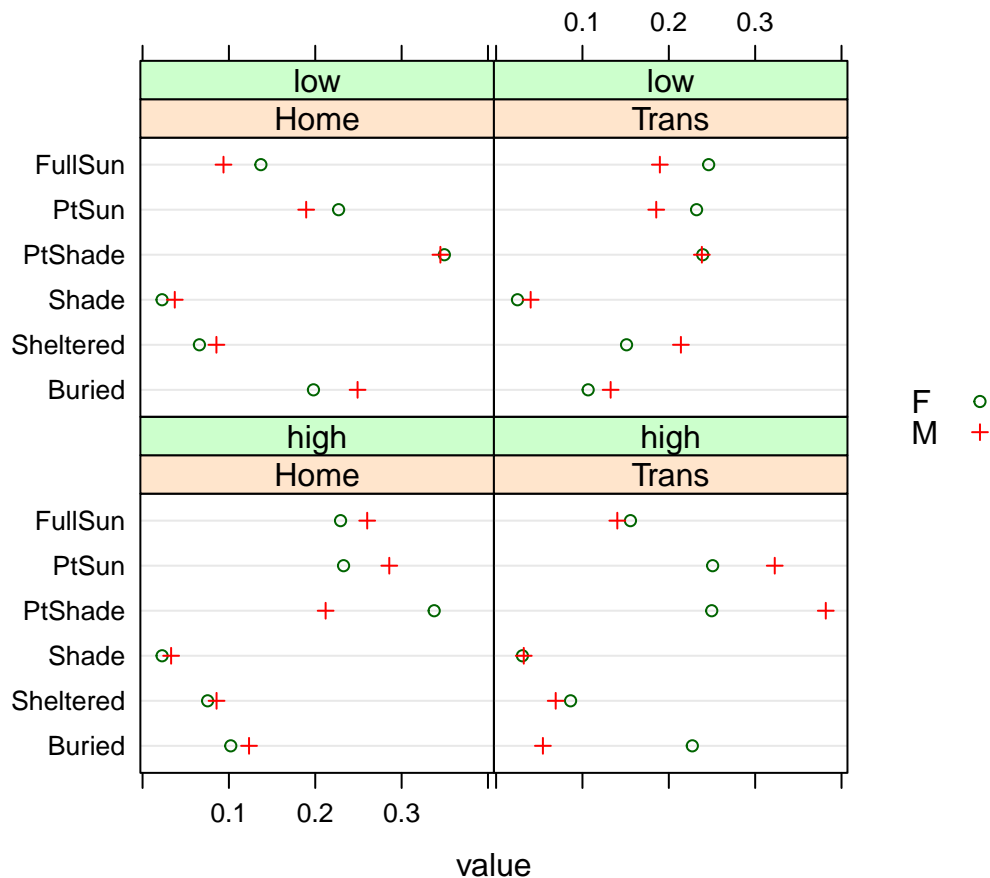
Using the generic function `predict`, we derive the model predicted proportions for the eight unique combinations of the three factors:

```
> pp <- predict(multinom.better, type = "probs", newdata = as.data.frame(table(lizards[,
+   c("Sex", "HomeElev", "Treatment")))[, 1:3], se.fit = T)
> pp2 <- predict(multinom.better2, type = "probs", newdata = as.data.frame(table(lizards[,
+   c("Sex", "HomeElev", "Treatment")))[, 1:3], se.fit = T)
```

We see that the model predicted proportions from the two models are the same.

To present the result, we use conditional plots from the package `lattice`.

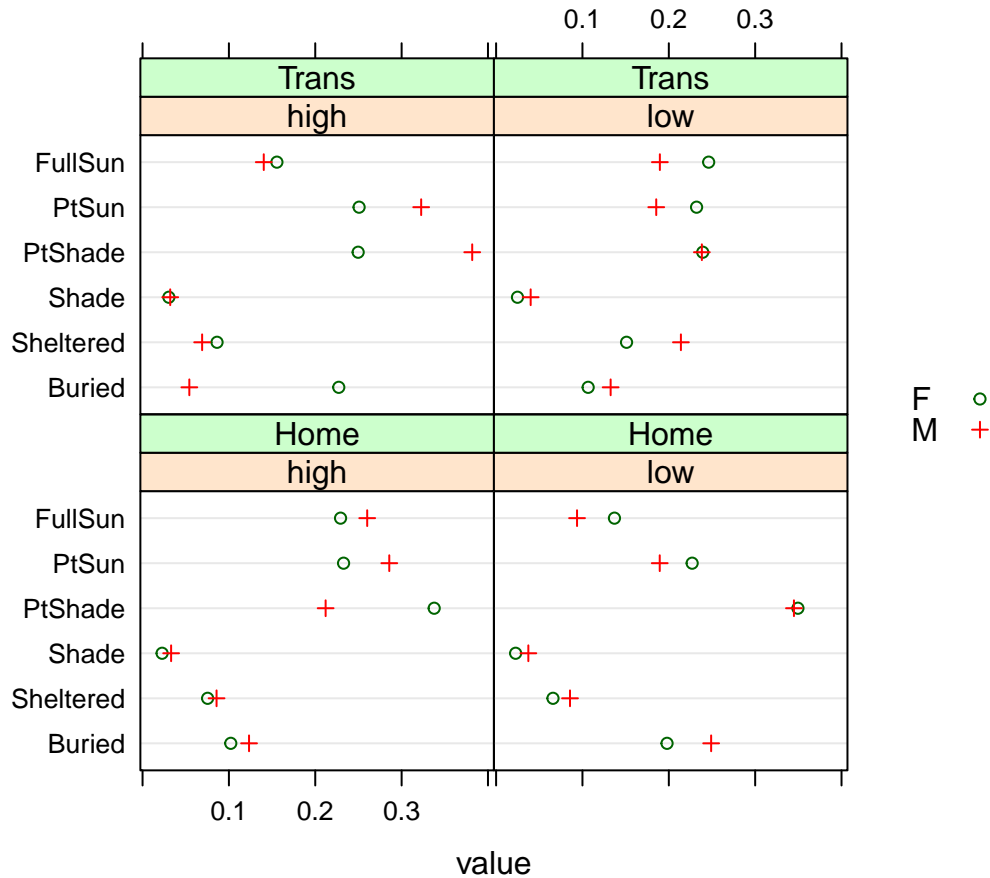
```
> pp <- cbind(as.data.frame(table(lizards[, c("Sex", "HomeElev", "Treatment")))[,
+   1:3], as.data.frame(pp))
> pp.molten <- melt(pp, id.var = c("Sex", "HomeElev", "Treatment"))
> ## pdf(paste(plotDIR, 'fitted.pdf', sep='/'), height=5, width=5.5)
> key <- simpleKey(levels(pp.molten$Sex), space = "right")
> dotplot(variable ~ value | Treatment * HomeElev, data = pp.molten, key = key,
+   groups = Sex)
```



```
> ## dev.off()
```

The above figure is focused on the difference in behavior between male and female lizards. At the same time, it is also effective for comparing the two populations of lizards (high versus low habitat) because we can easily tell the difference when two groups are lined up vertically. We can change the layout of the figure to emphasize the difference of the treatment:

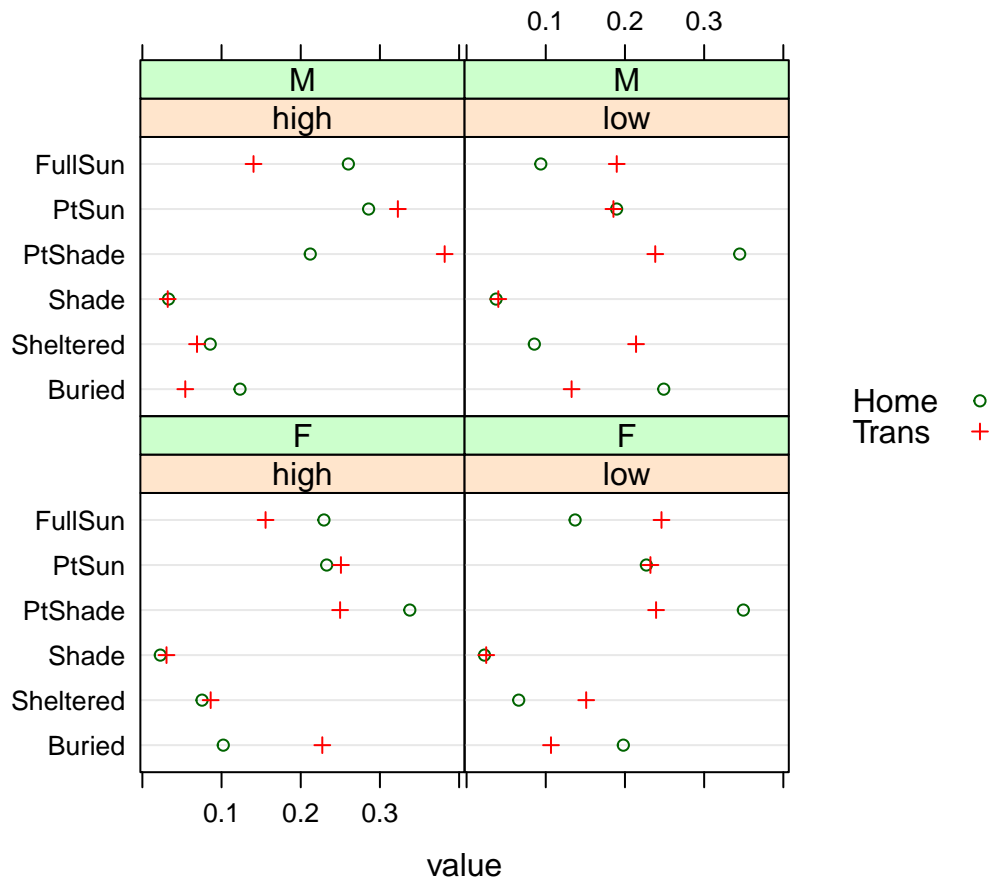
```
> ## pdf(paste(plotDIR, 'fitted2.pdf', sep='/'), height=5, width=5.5)
> key <- simpleKey(levels(pp.molten$Sex), space = "right")
> dotplot(variable ~ value | HomeElev * Treatment, data = pp.molten, key = key,
+         groups = Sex)
```



```
> ## dev.off()
```

Likewise, we can highlight the effect of treatment by using `Treatment` as the group variable:

```
> ## pdf(paste(plotDIR, 'fitted3.pdf', sep='/'), height=5, width=5.5)
> key <- simpleKey(levels(pp.molten$Treatment), space = "right")
> dotplot(variable ~ value | HomeElev * Sex, data = pp.molten, key = key, groups = Treatment)
```



```
> ## dev.off()
```

Simulation for Propagating Estimation Uncertainty

There is no analytic method for converting the estimation uncertainty with respect to the model coefficients (in the log-odds scale) to the uncertainty in the probability (proportion) scale. When using the function `predict` for models other than the simple linear regression models, the option `se.fit=T` is ignored. A Monte Carlo simulation program is provided by (Qian 2016) (Chapter 9) for this purpose. The function is reproduced here:

```
> ## function for generating random samples of model coefficients assuming
> ## estimated model coefficients have a multivariate normal distribution
>
> sim.multinom <- function(M, n.sims = NULL) {
+   ## M: a multinomial model object of class 'multinom' n.sims: number of Monte
+   ## Carlo simulations
+   packages(rv)
+   ## a package for random variate simulation and calculation
+   if (is.null(n.sims))
+     n.sims <- getnsims() else setnsims(n.sims)
+   ## setting simulation numbers to be either user supplied or rv package
+   ## default (2500)
+   object.class <- class(M)
+   if (object.class[1] != "multinom")
+     stop("Not a multinom object")
+ }
```

```

+
+   summ <- summary(M)
+   beta.hat <- as.vector(t(coef(M)))
+   V.beta <- vcov(M)
+   k <- length(beta.hat)
+   beta <- array(NA, c(n.sims, k))
+   lbs <- labels(coef(M))
+   dnmnm <- character()
+   for (i in 1:length(lbs[[1]])) dnmnm <- c(dnmnm, paste(lbs[[1]][i], lbs[[2]],
+     sep = ":"))
+   dimnames(beta) <- list(NULL, dnmnm)
+   beta <- mvrnorm(n.sims, beta.hat, V.beta)
+   return(beta)
+ }

```

The function returns a array of model coefficients. For this model, model coefficients are presented as a 5 by 8 matrix. The simulation program return random samples of the same matrix. We used the package `rv` to process the simulation results.

```

> sim.Better <- rvsims(sim.multinom(multinom.better, 5000))
> ## generating random samples of model coefficients and store them as an rv
> ## object

```

The following code chunk is for calculating the predicted proportions using `rv` objects. There are 40 (5×8) model coefficients and 48 (6×8) proportions. Each is represented by a random variable (a vector of 5000 random samples).

```

> ## design matrix for prediction
> sx <- as.numeric(pp$Sex == "M")
> hl <- as.numeric(pp$HomeElev == "low")
> tr <- as.numeric(pp$Treatment == "Trans")
>
> X <- cbind(1, sx, hl, tr, sx * hl, sx * tr, hl * tr, sx * hl * tr)
>
> sim.Better <- rvmatrix(sim.Better, nrow = 5, ncol = 8, byrow = T)
>
> Xb11 <- sum(X[1, ] * sim.Better[1, ])
> Xb12 <- sum(X[1, ] * sim.Better[2, ])
> Xb13 <- sum(X[1, ] * sim.Better[3, ])
> Xb14 <- sum(X[1, ] * sim.Better[4, ])
> Xb15 <- sum(X[1, ] * sim.Better[5, ])
>
> Xb21 <- sum(X[2, ] * sim.Better[1, ])
> Xb22 <- sum(X[2, ] * sim.Better[2, ])
> Xb23 <- sum(X[2, ] * sim.Better[3, ])
> Xb24 <- sum(X[2, ] * sim.Better[4, ])
> Xb25 <- sum(X[2, ] * sim.Better[5, ])
>
> Xb31 <- sum(X[3, ] * sim.Better[1, ])
> Xb32 <- sum(X[3, ] * sim.Better[2, ])
> Xb33 <- sum(X[3, ] * sim.Better[3, ])
> Xb34 <- sum(X[3, ] * sim.Better[4, ])
> Xb35 <- sum(X[3, ] * sim.Better[5, ])
>
> Xb41 <- sum(X[4, ] * sim.Better[1, ])

```

```

> Xb42 <- sum(X[4, ] * sim.Better[2, ])
> Xb43 <- sum(X[4, ] * sim.Better[3, ])
> Xb44 <- sum(X[4, ] * sim.Better[4, ])
> Xb45 <- sum(X[4, ] * sim.Better[5, ])
>
> Xb51 <- sum(X[5, ] * sim.Better[1, ])
> Xb52 <- sum(X[5, ] * sim.Better[2, ])
> Xb53 <- sum(X[5, ] * sim.Better[3, ])
> Xb54 <- sum(X[5, ] * sim.Better[4, ])
> Xb55 <- sum(X[5, ] * sim.Better[5, ])
>
> Xb61 <- sum(X[6, ] * sim.Better[1, ])
> Xb62 <- sum(X[6, ] * sim.Better[2, ])
> Xb63 <- sum(X[6, ] * sim.Better[3, ])
> Xb64 <- sum(X[6, ] * sim.Better[4, ])
> Xb65 <- sum(X[6, ] * sim.Better[5, ])
>
> Xb71 <- sum(X[7, ] * sim.Better[1, ])
> Xb72 <- sum(X[7, ] * sim.Better[2, ])
> Xb73 <- sum(X[7, ] * sim.Better[3, ])
> Xb74 <- sum(X[7, ] * sim.Better[4, ])
> Xb75 <- sum(X[7, ] * sim.Better[5, ])
>
> Xb81 <- sum(X[8, ] * sim.Better[1, ])
> Xb82 <- sum(X[8, ] * sim.Better[2, ])
> Xb83 <- sum(X[8, ] * sim.Better[3, ])
> Xb84 <- sum(X[8, ] * sim.Better[4, ])
> Xb85 <- sum(X[8, ] * sim.Better[5, ])
>
> pRV <- rvmatrix(0, nrow = 8, ncol = 6)
>
> denomsum <- 1 + exp(Xb11) + exp(Xb12) + exp(Xb13) + exp(Xb14) + exp(Xb15)
> pRV[1, 1] <- 1/denomsum
> pRV[1, 2] <- exp(Xb11)/denomsum
> pRV[1, 3] <- exp(Xb12)/denomsum
> pRV[1, 4] <- exp(Xb13)/denomsum
> pRV[1, 5] <- exp(Xb14)/denomsum
> pRV[1, 6] <- exp(Xb15)/denomsum
>
> denomsum <- 1 + exp(Xb21) + exp(Xb22) + exp(Xb23) + exp(Xb24) + exp(Xb25)
> pRV[2, 1] <- 1/denomsum
> pRV[2, 2] <- exp(Xb21)/denomsum
> pRV[2, 3] <- exp(Xb22)/denomsum
> pRV[2, 4] <- exp(Xb23)/denomsum
> pRV[2, 5] <- exp(Xb24)/denomsum
> pRV[2, 6] <- exp(Xb25)/denomsum
>
> denomsum <- 1 + exp(Xb31) + exp(Xb32) + exp(Xb33) + exp(Xb34) + exp(Xb35)
> pRV[3, 1] <- 1/denomsum
> pRV[3, 2] <- exp(Xb31)/denomsum
> pRV[3, 3] <- exp(Xb32)/denomsum
> pRV[3, 4] <- exp(Xb33)/denomsum
> pRV[3, 5] <- exp(Xb34)/denomsum

```



```

> pRV[3, 6] <- exp(Xb35)/denomsum
>
> denomsum <- 1 + exp(Xb41) + exp(Xb42) + exp(Xb43) + exp(Xb44) + exp(Xb45)
> pRV[4, 1] <- 1/denomsum
> pRV[4, 2] <- exp(Xb41)/denomsum
> pRV[4, 3] <- exp(Xb42)/denomsum
> pRV[4, 4] <- exp(Xb43)/denomsum
> pRV[4, 5] <- exp(Xb44)/denomsum
> pRV[4, 6] <- exp(Xb45)/denomsum
>
> denomsum <- 1 + exp(Xb51) + exp(Xb52) + exp(Xb53) + exp(Xb54) + exp(Xb55)
> pRV[5, 1] <- 1/denomsum
> pRV[5, 2] <- exp(Xb51)/denomsum
> pRV[5, 3] <- exp(Xb52)/denomsum
> pRV[5, 4] <- exp(Xb53)/denomsum
> pRV[5, 5] <- exp(Xb54)/denomsum
> pRV[5, 6] <- exp(Xb55)/denomsum
>
> denomsum <- 1 + exp(Xb61) + exp(Xb62) + exp(Xb63) + exp(Xb64) + exp(Xb65)
> pRV[6, 1] <- 1/denomsum
> pRV[6, 2] <- exp(Xb61)/denomsum
> pRV[6, 3] <- exp(Xb62)/denomsum
> pRV[6, 4] <- exp(Xb63)/denomsum
> pRV[6, 5] <- exp(Xb64)/denomsum
> pRV[6, 6] <- exp(Xb65)/denomsum
>
> denomsum <- 1 + exp(Xb71) + exp(Xb72) + exp(Xb73) + exp(Xb74) + exp(Xb75)
> pRV[7, 1] <- 1/denomsum
> pRV[7, 2] <- exp(Xb71)/denomsum
> pRV[7, 3] <- exp(Xb72)/denomsum
> pRV[7, 4] <- exp(Xb73)/denomsum
> pRV[7, 5] <- exp(Xb74)/denomsum
> pRV[7, 6] <- exp(Xb75)/denomsum
>
> denomsum <- 1 + exp(Xb81) + exp(Xb82) + exp(Xb83) + exp(Xb84) + exp(Xb85)
> pRV[8, 1] <- 1/denomsum
> pRV[8, 2] <- exp(Xb81)/denomsum
> pRV[8, 3] <- exp(Xb82)/denomsum
> pRV[8, 4] <- exp(Xb83)/denomsum
> pRV[8, 5] <- exp(Xb84)/denomsum
> pRV[8, 6] <- exp(Xb85)/denomsum

```

Using the summary function from `rv`, we convert the random variable object to a matrix of mean and selected percentiles, so that we can graphically present the estimated means and their 95% confidence intervals.

```

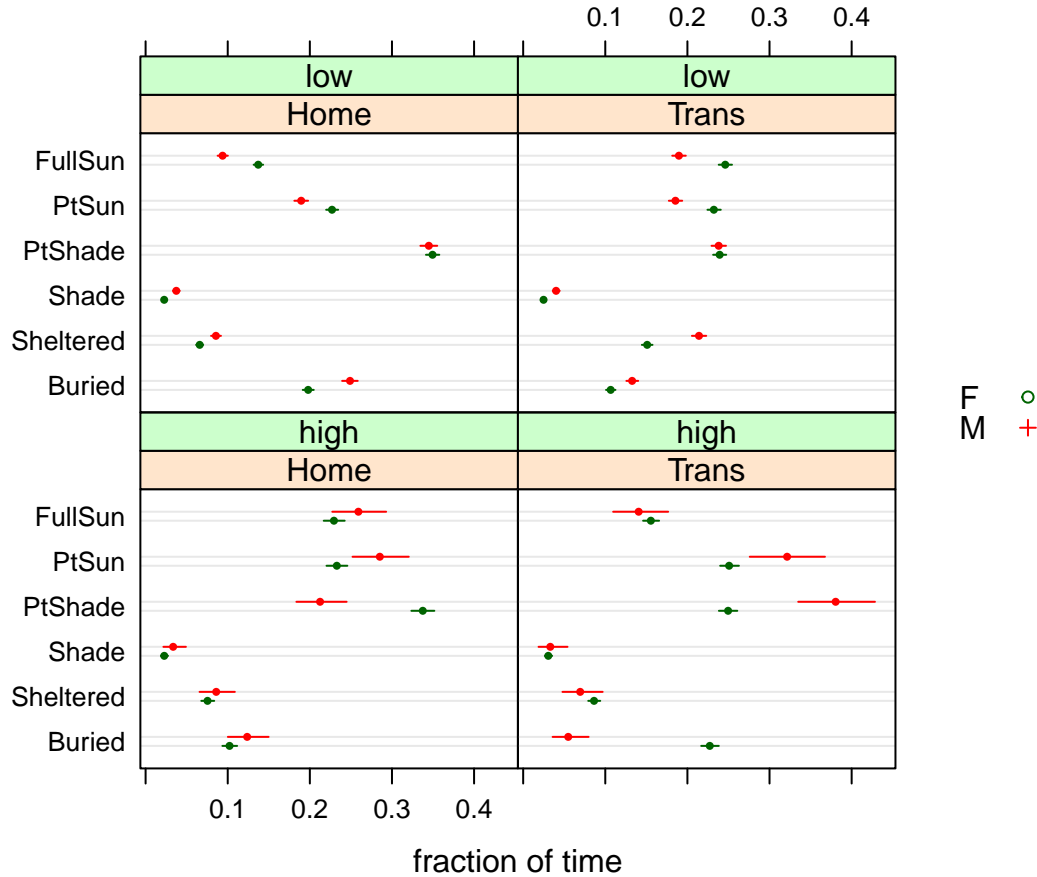
> dataPlot <- as.data.frame(cbind(pp.molten, summary(pRV)))
>
> # pdf(paste(plotDIR, 'fittedUncas.pdf', sep='/'), height=5, width=5.75)
> my.panel <- function(x, y, subscripts, group.number, col, ...) {
+   myjitter <- c(-0.1, 0.1)
+   panel.dotplot(x, as.numeric(y) + myjitter[group.number], cex = 0.5, col = col)
+   ## panel.grid()
+   panel.segments(dataPlot[subscripts, 9], as.numeric(y) + myjitter[group.number],
+     dataPlot[subscripts, 13], as.numeric(y) + myjitter[group.number], col = col)
+ }

```

```

+
+ }
+
+ > key <- simpleKey(levels(dataPlot$Sex), space = "right")
+ > dotplot(variable ~ mean | Treatment * HomeElev, data = dataPlot, key = key,
+   groups = Sex, col = key$points$col, panel = my.panel, xlab = "fraction of time",
+   xlim = range(dataPlot[, c(9, 13)]) + c(-0.025, 0.025))

```



```

> # dev.off()

```

We can use this figure to compare the difference between male and female. When the two confidence intervals do not overlap, we can safely conclude that the male and female fractions are statistically different (at $\alpha = 0.05$). However, because of the strong correlation often associated with regression model coefficients, we cannot easily decide whether the difference is statistically different from 0 when the confidence intervals do not overlap. Using the simulation results, we can directly calculate the difference and quantify the confidence intervals of the differences.

```

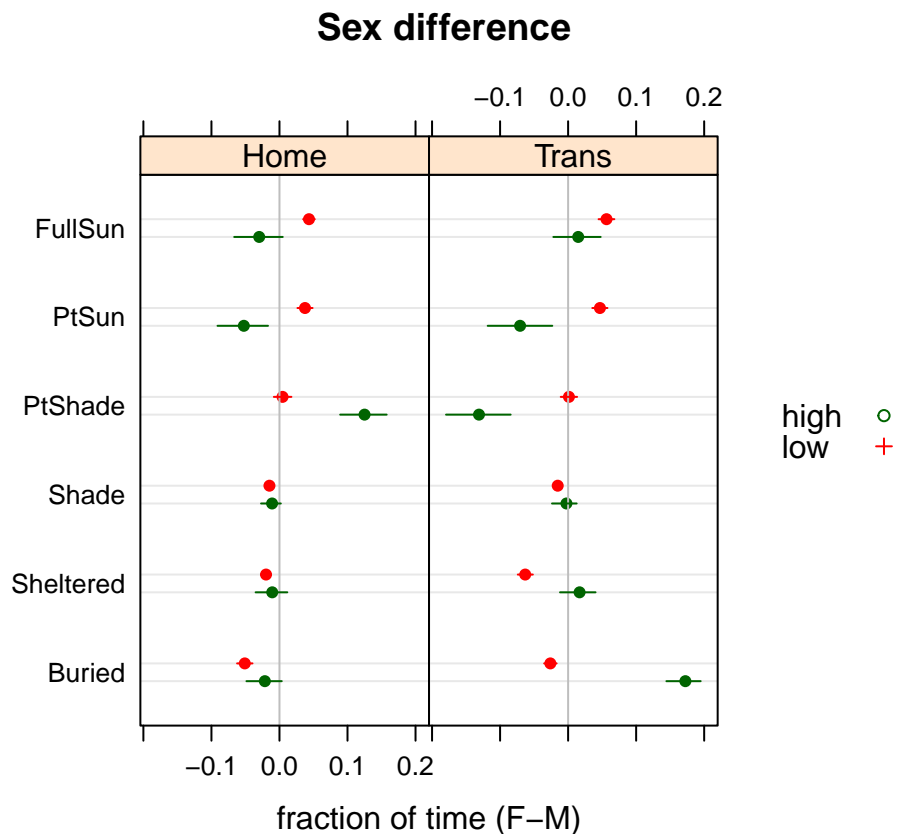
> ### Sex difference
> dataPlotSex <- as.data.frame(cbind(pp.molten[seq(1, 48, 2), ], summary(pRV[seq(1,
+   8, 2), ] - pRV[seq(2, 8, 2), ])))
>
>
> my.panel <- function(x, y, subscripts, group.number, col, ...) {
+   myjitter <- c(-0.1, 0.1)
+   panel.dotplot(x, as.numeric(y) + myjitter[group.number], cex = 0.75, col = col)
+   ## panel.grid()

```

```

+   panel.abline(v = 0, col = "gray")
+   panel.segments(dataPlotSex[subscripts, 9], as.numeric(y) + myjitter[group.number],
+     dataPlotSex[subscripts, 13], as.numeric(y) + myjitter[group.number],
+     col = col)
+ }
+
> # pdf(paste(plotDIR, 'sexdiff1.pdf', sep='/'), height=4.5, width=5)
> key <- simpleKey(levels(dataPlotSex$HomeElev), space = "right")
> dotplot(variable ~ mean | Treatment, data = dataPlotSex, key = key, groups = HomeElev,
+   col = key$points$col, panel = my.panel, xlab = "fraction of time (F-M)",
+   main = "Sex difference", xlim = range(dataPlotSex[, c(9, 13)]) + c(-0.025,
+   0.025))

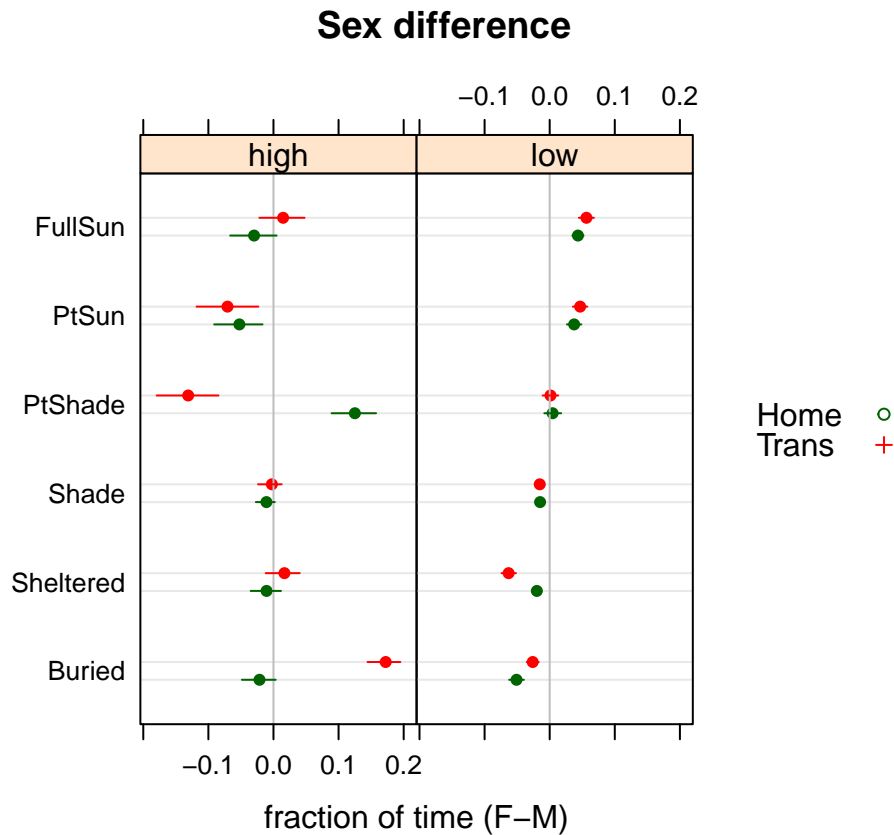
```



```

> # dev.off()
>
> ## pdf(paste(plotDIR, 'sexdiff2.pdf', sep='/'), height=4.5, width=5)
> key <- simpleKey(levels(dataPlotSex$Treatment), space = "right")
> dotplot(variable ~ mean | HomeElev, data = dataPlotSex, key = key, groups = Treatment,
+   col = key$points$col, panel = my.panel, xlab = "fraction of time (F-M)",
+   main = "Sex difference", xlim = range(dataPlotSex[, c(9, 13)]) + c(-0.025,
+   0.025))

```



```
> ## dev.off()
```

In some cases, we may not be interested in the sex differences. We can fit the multinomial regression without the sex effect and redraw the figures accordingly.

```
> multinom.better2 <- multinom(as.matrix(lizards[, PID])) ~ HomeElev * Treatment,
+   data = lizards)
```

```
## # weights: 30 (20 variable)
## initial value 93209.119349
## iter 10 value 84838.876990
## iter 20 value 84574.985918
## final value 84194.183972
## converged
```

```
> sim.Better2 <- rvsims(sim.multinom(multinom.better2, 5000))
>
> pp2 <- predict(multinom.better2, type = "probs", newdata = as.data.frame(table(lizards[,
+   c("HomeElev", "Treatment")))[, 1:2], se.fit = T)
>
> pp2 <- cbind(as.data.frame(table(lizards[, c("HomeElev", "Treatment")))[, 1:2],
+   as.data.frame(pp2))
>
> pp2.molten <- melt(pp2, id.var = c("HomeElev", "Treatment"))
> ## plots without Sex:
> hl <- as.numeric(pp2$HomeElev == "low")
> tr <- as.numeric(pp2$Treatment == "Trans")
>
> X <- cbind(1, hl, tr, hl * tr)
```

```

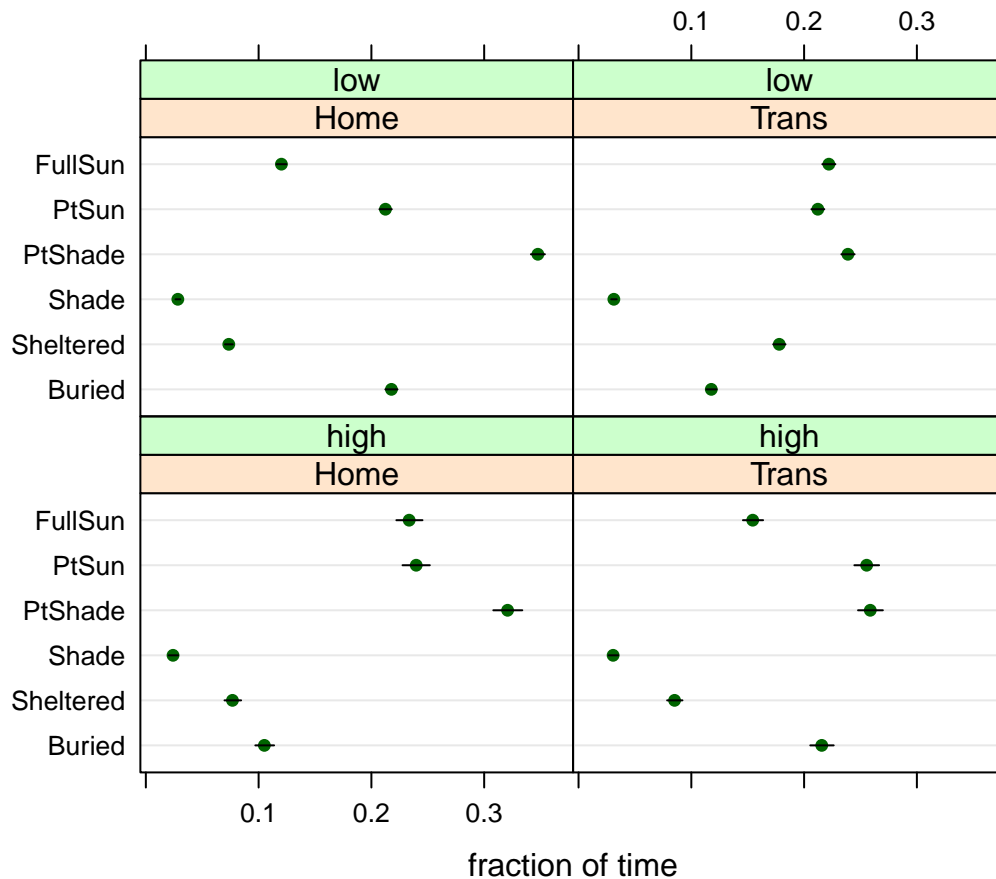
>
> sim.Better2 <- rvmatrix(sim.Better2, nrow = 5, ncol = 4, byrow = T)
>
> Xb11 <- sum(X[1, ] * sim.Better2[1, ])
> Xb12 <- sum(X[1, ] * sim.Better2[2, ])
> Xb13 <- sum(X[1, ] * sim.Better2[3, ])
> Xb14 <- sum(X[1, ] * sim.Better2[4, ])
> Xb15 <- sum(X[1, ] * sim.Better2[5, ])
>
> Xb21 <- sum(X[2, ] * sim.Better2[1, ])
> Xb22 <- sum(X[2, ] * sim.Better2[2, ])
> Xb23 <- sum(X[2, ] * sim.Better2[3, ])
> Xb24 <- sum(X[2, ] * sim.Better2[4, ])
> Xb25 <- sum(X[2, ] * sim.Better2[5, ])
>
> Xb31 <- sum(X[3, ] * sim.Better2[1, ])
> Xb32 <- sum(X[3, ] * sim.Better2[2, ])
> Xb33 <- sum(X[3, ] * sim.Better2[3, ])
> Xb34 <- sum(X[3, ] * sim.Better2[4, ])
> Xb35 <- sum(X[3, ] * sim.Better2[5, ])
>
> Xb41 <- sum(X[4, ] * sim.Better2[1, ])
> Xb42 <- sum(X[4, ] * sim.Better2[2, ])
> Xb43 <- sum(X[4, ] * sim.Better2[3, ])
> Xb44 <- sum(X[4, ] * sim.Better2[4, ])
> Xb45 <- sum(X[4, ] * sim.Better2[5, ])
>
> pRV <- rvmatrix(0, nrow = 4, ncol = 6)
>
> denomsum <- 1 + exp(Xb11) + exp(Xb12) + exp(Xb13) + exp(Xb14) + exp(Xb15)
> pRV[1, 1] <- 1/denomsum
> pRV[1, 2] <- exp(Xb11)/denomsum
> pRV[1, 3] <- exp(Xb12)/denomsum
> pRV[1, 4] <- exp(Xb13)/denomsum
> pRV[1, 5] <- exp(Xb14)/denomsum
> pRV[1, 6] <- exp(Xb15)/denomsum
>
> denomsum <- 1 + exp(Xb21) + exp(Xb22) + exp(Xb23) + exp(Xb24) + exp(Xb25)
> pRV[2, 1] <- 1/denomsum
> pRV[2, 2] <- exp(Xb21)/denomsum
> pRV[2, 3] <- exp(Xb22)/denomsum
> pRV[2, 4] <- exp(Xb23)/denomsum
> pRV[2, 5] <- exp(Xb24)/denomsum
> pRV[2, 6] <- exp(Xb25)/denomsum
>
> denomsum <- 1 + exp(Xb31) + exp(Xb32) + exp(Xb33) + exp(Xb34) + exp(Xb35)
> pRV[3, 1] <- 1/denomsum
> pRV[3, 2] <- exp(Xb31)/denomsum
> pRV[3, 3] <- exp(Xb32)/denomsum
> pRV[3, 4] <- exp(Xb33)/denomsum
> pRV[3, 5] <- exp(Xb34)/denomsum
> pRV[3, 6] <- exp(Xb35)/denomsum
>

```

```

> denomsum <- 1 + exp(Xb41) + exp(Xb42) + exp(Xb43) + exp(Xb44) + exp(Xb45)
> pRV[4, 1] <- 1/denomsum
> pRV[4, 2] <- exp(Xb41)/denomsum
> pRV[4, 3] <- exp(Xb42)/denomsum
> pRV[4, 4] <- exp(Xb43)/denomsum
> pRV[4, 5] <- exp(Xb44)/denomsum
> pRV[4, 6] <- exp(Xb45)/denomsum
>
>
> dataPlot <- as.data.frame(cbind(pp2.molten, summary(pRV)))
>
> ## pdf(paste(plotDIR, 'fittedUncasNS.pdf', sep='/'), height=5, width=5.75)
> my.panel <- function(x, y, subscripts, ...) {
+   panel.dotplot(x, as.numeric(y))
+   ## panel.grid()
+   panel.segments(dataPlot[subscripts, 8], as.numeric(y), dataPlot[subscripts,
+   12], as.numeric(y))
+ }
>
> dotplot(variable ~ mean | Treatment * HomeElev, data = dataPlot, panel = my.panel,
+   xlab = "fraction of time", xlim = range(dataPlot[, c(8, 12)]) + c(-0.025,
+   0.025))

```



```

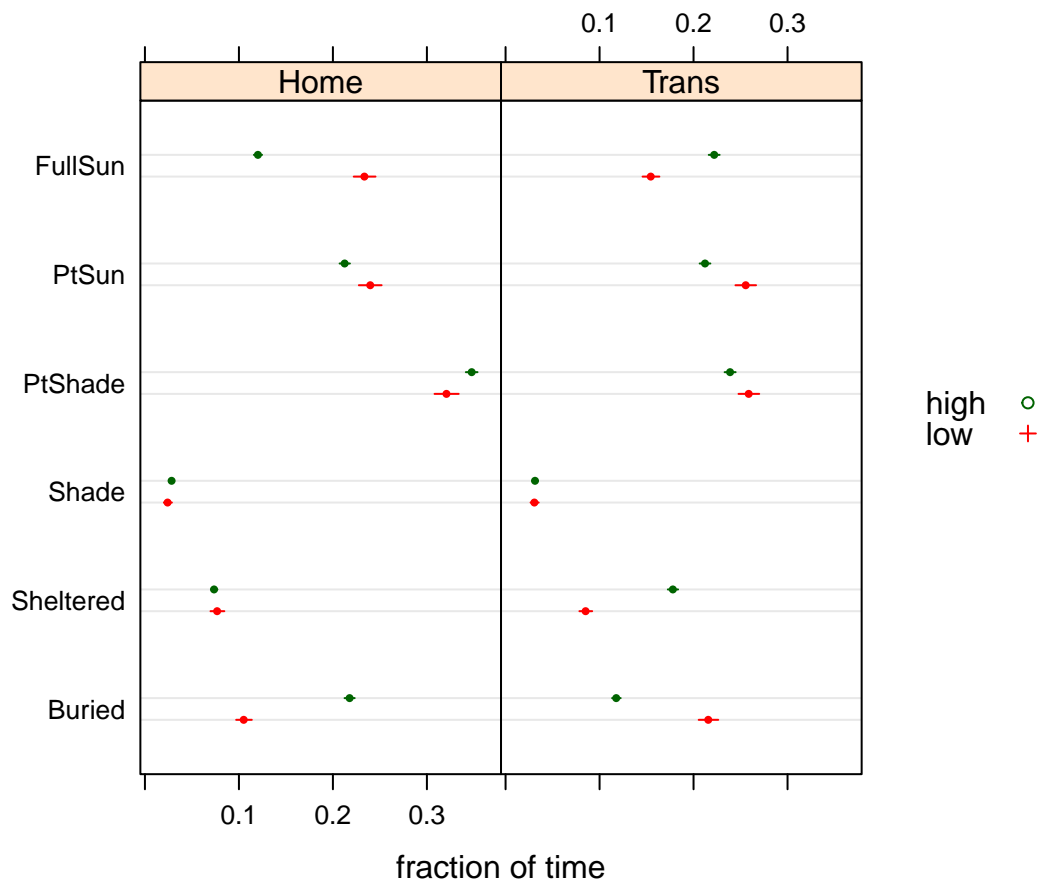
> ## dev.off()
>

```

```

>
>
> my.panel <- function(x, y, subscripts, group.number, col, ...) {
+   myjitter <- c(-0.1, 0.1)
+   panel.dotplot(x, as.numeric(y) + myjitter[group.number], cex = 0.5, col = col)
+   ## panel.grid() panel.abline(v=0, col='gray')
+   panel.segments(dataPlot[subscripts, 8], as.numeric(y) + myjitter[group.number],
+     dataPlot[subscripts, 12], as.numeric(y) + myjitter[group.number], col = col)
+ }
>
> ## pdf(paste(plotDIR, 'inteff1.pdf', sep='/'), height=4.5, width=5)
>
> key <- simpleKey(text = levels(dataPlot$HomeElev), space = "right")
> dotplot(variable ~ mean | Treatment, data = dataPlot, key = key, groups = HomeElev,
+   col = rev(key$points$col), panel = my.panel, xlab = "fraction of time",
+   xlim = range(dataPlot[, c(8, 12)]) + c(-0.025, 0.025))

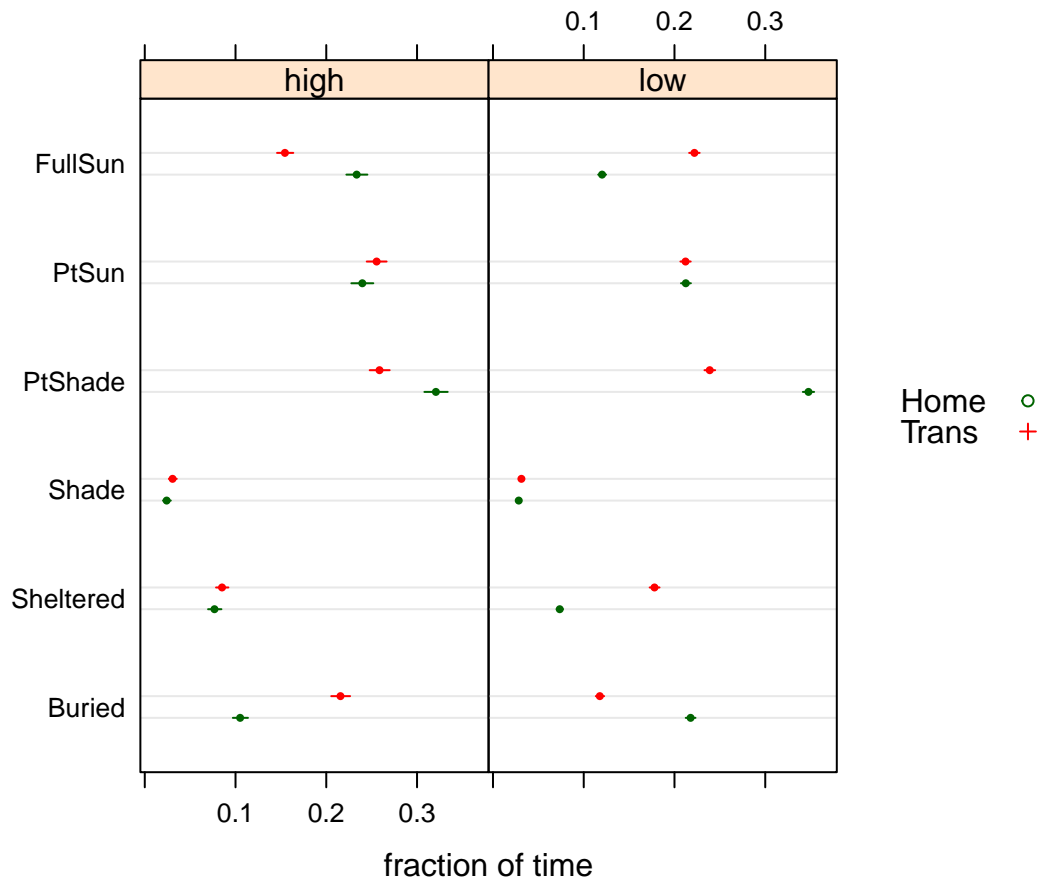
```



```

> ## dev.off()
>
> ## pdf(paste(plotDIR, 'inteff2.pdf', sep='/'), height=4.5, width=5)
> key <- simpleKey(levels(dataPlot$Treatment), space = "right")
> dotplot(variable ~ mean | HomeElev, data = dataPlot, key = key, groups = Treatment,
+   col = key$points$col, panel = my.panel, xlab = "fraction of time", xlim = range(dataPlot[,
+   c(8, 12)]) + c(-0.025, 0.025))

```



```
> ## dev.off()
```

Revision

During the review, we were asked to describe the behavior of these lizards when they were sunning. That is, what time do they bask when they were classified as in full sun. Using the raw time-stamp data, we added a variable for those basking lizards to group them based on the time of full sun exposure (morning, afternoon, and evening) (specific hours).

```
> dayhours <- read.csv(paste(dataDIR, "lizard light data_time bins.csv", sep = "/"))
> moltern <- melt(dayhours, id = c("LizardID", "Sex", "HomeElev", "Treatment",
+   "TimeBin"), measure = "HoFullSun")
> daycast <- dcast(moltern, LizardID + Sex + HomeElev + Treatment ~ TimeBin)
> daycast2 <- dcast(moltern, LizardID + HomeElev + Treatment ~ TimeBin, sum)
```

The model

```
> pid <- c(5, 4, 6)
> day_mln <- multinom(as.matrix(daycast2[, pid]) ~ HomeElev * Treatment, data = daycast2)

## # weights: 15 (8 variable)
## initial value 9908.384231
## iter 10 value 9661.974545
## final value 9660.693084
## converged
```



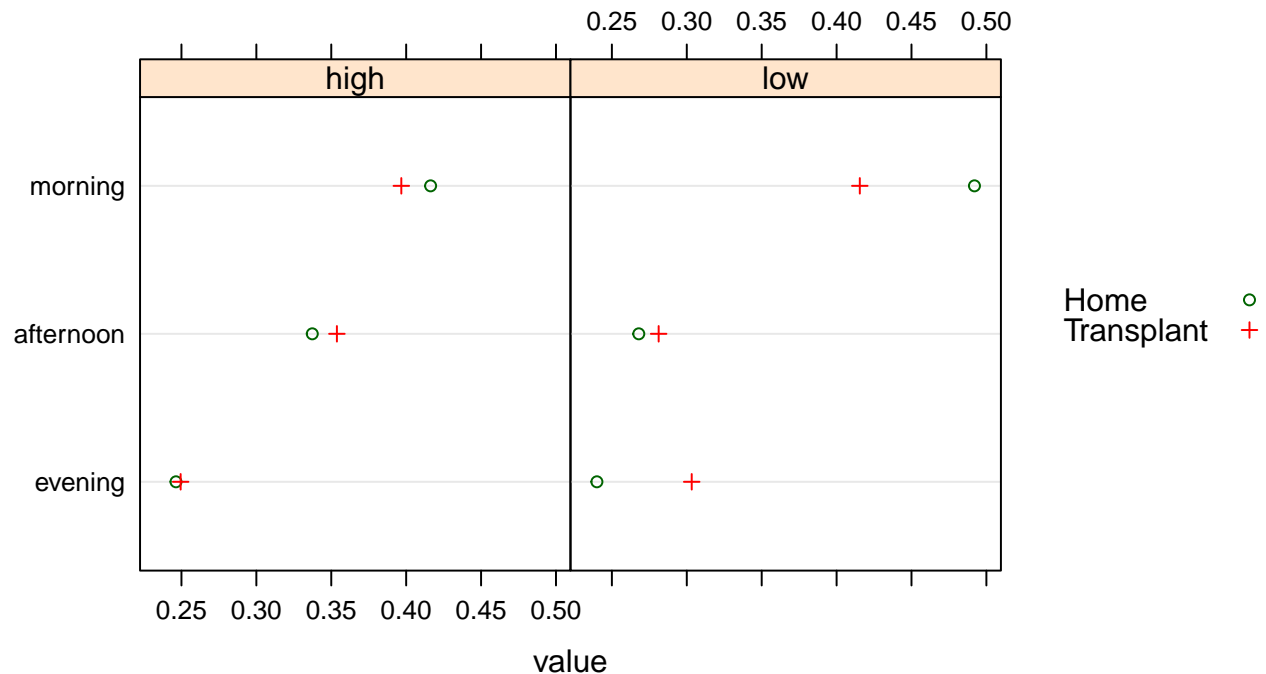
```

> summary(day_mln)

## Call:
## multinom(formula = as.matrix(daycast2[, pid]) ~ HomeElev * Treatment,
##          data = daycast2)
##
## Coefficients:
##          (Intercept) HomeElevlow TreatmentTransplant
## afternoon  0.3143190 -0.2041246          0.03500339
## morning    0.5246614  0.1932505          -0.06060565
##          HomeElevlow:TreatmentTransplant
## afternoon                    -0.2207597
## morning                      -0.3425066
##
## Std. Errors:
##          (Intercept) HomeElevlow TreatmentTransplant
## afternoon 0.07764561 0.09537418          0.1156663
## morning   0.07447254 0.08918262          0.1121006
##          HomeElevlow:TreatmentTransplant
## afternoon                    0.1342457
## morning                      0.1276145
##
## Residual Deviance: 19321.39
## AIC: 19337.39

> pp <- predict(day_mln, type = "prob", newdata = as.data.frame(table(daycast2[,
+   c("HomeElev", "Treatment")))[, -3]))
> pp <- cbind(as.data.frame(table(daycast2[, c("HomeElev", "Treatment")))[, -3],
+   as.data.frame(pp))
> pp.molten <- melt(pp, id.var = c("HomeElev", "Treatment"))
> ## pdf(paste(plotDIR, 'fittedFullSun.pdf', sep='/'), height=5, width=5.5)
> key <- simpleKey(levels(pp.molten$Treatment), space = "right")
> dotplot(variable ~ value | HomeElev, data = pp.molten, key = key, groups = Treatment)

```



```
> ## dev.off()
```

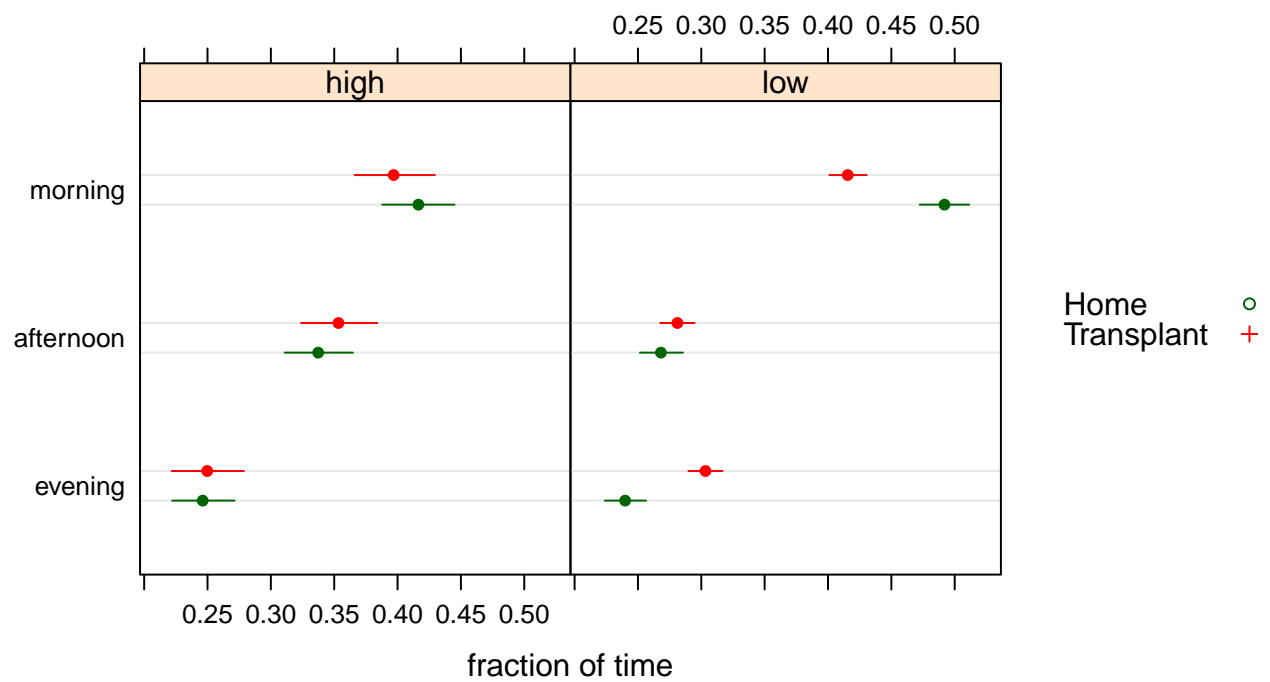
Simulation

```
> sim.cast2 <- rvsims(sim.multinom(day_mln, 5000))
>
> ## plots without Sex:
> hl <- as.numeric(pp$HomeElev == "low")
> tr <- as.numeric(pp$Treatment == "Transplant")
>
> X <- cbind(1, hl, tr, hl * tr)
>
> sim.cast2 <- rvmatrix(sim.cast2, nrow = 2, ncol = 4, byrow = T)
>
> Xb11 <- sum(X[1, ] * sim.cast2[1, ])
> Xb12 <- sum(X[1, ] * sim.cast2[2, ])
>
> Xb21 <- sum(X[2, ] * sim.cast2[1, ])
> Xb22 <- sum(X[2, ] * sim.cast2[2, ])
>
> Xb31 <- sum(X[3, ] * sim.cast2[1, ])
> Xb32 <- sum(X[3, ] * sim.cast2[2, ])
>
> Xb41 <- sum(X[4, ] * sim.cast2[1, ])
> Xb42 <- sum(X[4, ] * sim.cast2[2, ])
>
> pRV <- rvmatrix(0, nrow = 4, ncol = 3)
>
> denomsum <- 1 + exp(Xb11) + exp(Xb12)
> pRV[1, 1] <- 1/denomsum
> pRV[1, 2] <- exp(Xb11)/denomsum
> pRV[1, 3] <- exp(Xb12)/denomsum
>
```

```

> denomsum <- 1 + exp(Xb21) + exp(Xb22)
> pRV[2, 1] <- 1/denomsum
> pRV[2, 2] <- exp(Xb21)/denomsum
> pRV[2, 3] <- exp(Xb22)/denomsum
>
> denomsum <- 1 + exp(Xb31) + exp(Xb32)
> pRV[3, 1] <- 1/denomsum
> pRV[3, 2] <- exp(Xb31)/denomsum
> pRV[3, 3] <- exp(Xb32)/denomsum
>
> denomsum <- 1 + exp(Xb41) + exp(Xb42)
> pRV[4, 1] <- 1/denomsum
> pRV[4, 2] <- exp(Xb41)/denomsum
> pRV[4, 3] <- exp(Xb42)/denomsum
>
>
> dataPlot <- as.data.frame(cbind(pp.molten, summary(pRV)))
>
> ## pdf(paste(plotDIR, 'fittedUncasFullSun.pdf', sep='/'), height=5,
> ## width=5.75)
> key <- simpleKey(levels(dataPlot$Treatment), space = "right")
> dotplot(variable ~ mean | HomeElev, data = dataPlot, key = key, groups = Treatment,
+   col = key$points$col, panel = function(x, y, subscripts, group.number, col,
+   ...) {
+     myjitter <- c(-0.1, 0.1)
+     panel.dotplot(x, as.numeric(y) + myjitter[group.number], cex = 0.75,
+       col = col)
+     panel.abline(v = 0, col = "gray")
+     panel.segments(dataPlot[subscripts, 8], as.numeric(y) + myjitter[group.number],
+       dataPlot[subscripts, 12], as.numeric(y) + myjitter[group.number],
+       col = col)
+   }, xlab = "fraction of time", xlim = range(dataPlot[, c(8, 12)]) + c(-0.025,
+   0.025))

```



```
> ## dev.off()
```

Qian, S.S. 2016. *Environmental and Ecological Statistics with R*. 2nd ed. Chapman; Hall/CRC Press.