

```
#####
### stevens_lemur_rcode.R
### Created by Jeffrey R. Stevens on 4 Mar 2011 (jeffrey.r.stevens@gmail.com),
### finalized on 20 Apr 2011
### Summary: This script calculates descriptive statistics and generates figures
### for the analysis of lemur intertemporal choice (Stevens & Mülhroff submitted).
### Instructions: Place this file and the data files (stevens_[comparative/lemur]_data.csv)
### in the same directory. Create a folder called "figures". Set the R
### working directory to this directory. At the R command prompt, type
### > source("stevens_rcode.R")
### This will run the script, adding all of the calculated variables to the
### workspace and saving PDF versions of the figures in the figures directory.
### Uses: This script can be reproduced and modified for personal and scientific use.
### Data files: Description of the data columns:
### stevens_lemur_data
### date - date of session
### time - time of session
### subject - name of subject
### species - species of subject
### sex - sex of subject
### main - flag for whether in the main experiment (1) or the training (0)
### lt - delay to large amount (adjusting)
### free - free-choice trial (1) or forced-choice trial (0)
### trial - trial number within a session
### L - amount presented on left side
### R - amount presented on right side
### choice_side - side chosen by subject
### completed - flag representing whether the session was valid (1) or not (0)
### time_available - time at which food became available for subject
### start_eat - time at which subject put first food in mouth
### end_eat - time at which subject put last food in mouth
### stevens_comparative_data
### subject - name of subject
### species - species of subject
### type - specific description of species
### latin_name - scientific name of species
### exp_cond - experimental condition: 2v6 is two vs. six food items, lv3 is one vs. three food items, and other is a different
adjusting delay scheme
### indiff - mean adjusting delay at indifference for each subject (primary dependent variable for comparative analysis)
### body_wt - body mass of specific individual or mean for species
### indiff_ref - reference for mean adjusting delay at indifference data
### body_wt_ref - reference for body mass data
#####
#####
### Load libraries, data, and R version
#####
rm(list=ls())
library(ape)
ape.ver <- installed.packages()["ape", "Version"]

# clears all variables
# needed for independent contrasts
# get ape version
```

```

library(beeswarm)
beeswarm.ver <- installed.packages()[["beeswarm", "Version"]]
library(xtable)
xtable.ver <- installed.packages()[["xtable", "Version"]]
ver <- getRversion()

choice_data <- read.csv("data/stevens_lemur_data.csv")
comparative <- read.csv("data/stevens_comparative_data.csv")

#####
### Calculate mean adjusted delay to indifference
#####

## Clean and distill data
choice_data$date <- as.Date(choice_data$date, "%d-%b-%y") # convert dates to proper date format
choice_data <- choice_data[which(choice_data$subject == "Gustav" & choice_data$end_eat == ""), ] #
Gustav was run beyond his indifference point according to our criteria, so we stop analysing choice data after 26 May 2008
freechoice <- subset(choice_data, (main == 1 & free == 1 & completed == 1)) # select data from the main experiment (not numerical
discrimination) using completed free-choice trials
freechoice$choice <- ifelse(freechoice$choice_side == "-", NA, ifelse(freechoice$R == 6, ifelse(freechoice$choice_side == "R", 1, 0),
ifelse(freechoice$choice_side == "L", 1, 0))) # create choice column that codes choosing the larger reward as 1 and the smaller
reward as 0 (currently coded as choice side)
freechoice <- freechoice[, c(1, 3, 4, 5, 7, 9, 12, 13, 14, 19)] # select only necessary columns of data

## Calculate the number of sessions and create column of session numbers
maxrows <- length(freechoice[,1])
currentsubject <- "0"
subjectcounter <- 0
subjsessions <- data.frame(rep(0, 5), rep(0, 5))
names(subjsessions) <- c("subject", "totalsessions")

for(i in 1:maxrows) {
  if(i == maxrows) {
    freechoice[i, 11] <- session
    subjectcounter <- subjectcounter + 1
    subjsessions[subjectcounter, ] <- c(currentsubject, session)
  } else {
    if(freechoice[i, 2] == currentsubject) {
      if(freechoice[i, 1] == freechoice[i - 1, 1]) {
        freechoice[i, 11] <- session
      } else {
        session <- session + 1
        freechoice[i, 11] <- session
      }
    } else {
      if(i > 1) {
        subjectcounter <- subjectcounter + 1
      }
    }
  }
}

# needed for beeswarm plot
# get beeswarm version
# needed to create latex table
# get xtable version
# get R version

# read in lemur choice and time data
# read in comparative intertemporal choice data from other primates

# if the subject name does not match the current subject
# if it is not the first row
# increment the subject counter

```

```

subjsessions[subjcounter, ] <- c(currentsubject, session) # assign current subject and current session to subjsessions
currentsubject <- freechoice[i, 2] # update the new current subject
session <- 1 # initialize the session to 1
freechoice[i, 11] <- session # assign the final column of freechoice to the current session
}
else {
  currentsubject <- freechoice[i, 2] # if it is the first row
  session <- 1 # update the current subject with the first subject
  freechoice[i, 11] <- session # initialize the session to 1
}
}
}

names(freechoice)[names(freechoice)=="V11"] <- "session" # rename final column as session number
write.csv(freechoice, file = "data/stevens_choice_data.csv") # write to file
subjsessions$subject <- levels(currentsubject) # assign subject names to session data

## Calculate temporal variables of latency and handling time
times <- subset(choice_data, end_eat != "") # select choice_data with time data (no empty data)
times$time_available <- strptime(times$time_available, "%H:%M:%S") # convert time_available data to proper time format
times$start_eat <- strptime(times$start_eat, "%H:%M:%S") # convert start_eat data to proper time format
times$end_eat <- strptime(times$end_eat, "%H:%M:%S") # convert end_eat data to proper time format
times$latency <- difftime(times$start_eat, times$time_available) # assign difference between time_available and start_eat to latency
times$handling <- difftime(times$end_eat, times$start_eat) # assign difference between start_eat and end_eat to handling
times$total_time <- times$latency + times$handling # sum latency and handling for total time
times$size <- times$L + times$R # assign reward amount to size
write.csv(times, file = "data/stevens_times_data.csv") # write to file
subj_times <- aggregate(cbind(latency, handling, total_time) ~ subject + size, # subject + size,
  data = times, FUN = mean) # aggregate times by subject and reward amount

## Extract data used to calculate indifference points for each subject
blumchen <- subset(freechoice, (subject == "Blumchen" & session > subjsessions[1, 2] - 5)) # select last 5 sessions before
indifference for subject
gustav <- subset(freechoice, (subject == "Gustav" & session > subjsessions[2, 2] - 5))
ole <- subset(freechoice, (subject == "Ole" & session > subjsessions[3, 2] - 5))
puppi <- subset(freechoice, (subject == "Püppi" & session > subjsessions[4, 2] - 5))
uta <- subset(freechoice, (subject == "Uta" & session > subjsessions[5, 2] - 5))
titration <- rbind(blumchen, gustav, ole, puppi, uta) # concatenate last 5 sessions of data for
all subjects

## Generate data frame of times for each subject
subj_data <- aggregate(titration$t, by = list(titration$subject), FUN = mean) # aggregate long delay by subject
names(subj_data) <- c("subject", "ldelay")
subj_data$species <- c("R", "BW", "BW", "BW", "BL") # create column of species
subj_data$sex <- c("F", "M", "M", "F", "F") # create column of sex
subj_data$ldelay <- round(subj_data$ldelay, 2) # round long delay values
subj_data$delay <- c(rep(0, 5)) # assign all short delays to 0.1
subj_data <- subj_data[, c(1, 3, 4, 5, 2)] # reorder columns
subj_data$amt <- subj_times[1:5, 2] # assign small amount
subj_data$amt <- subj_times[6:10, 2] # assign large amount

```

```

subj_data$iti <- c(rep(30, 5))
subj_data$latency <- round(subj_times[1:5, 3], 2)
subj_data$latency <- round(subj_times[6:10, 3], 2)
subj_data$handling <- round(subj_times[1:5, 4], 2)
subj_data$handling <- round(subj_times[6:10, 4], 2)
subj_data$totaltime <- round(subj_times[1:5, 5], 2)
subj_data$totaltime <- round(subj_times[6:10, 5], 2)
# calculate short-term rate for small and large rewards
subj_data$ratepred <- (((subj_data$delay + subj_data$handling) * subj_data$amt) - subj_data$handling # includes
only handling time
subj_data$ratepred2 <- (((subj_data$delay + subj_data$totaltime) * subj_data$amt) - subj_data$totaltime # includes
both handling time and latency
# calculate long-term rate for small and large rewards
subj_data$ratepred <- (((subj_data$delay + subj_data$handling + subj_data$iti) * subj_data$amt) / subj_data$amt) - (subj_data
$handling + subj_data$iti) # includes only handling time
subj_data$ratepred2 <- (((subj_data$delay + subj_data$totaltime + subj_data$iti) * subj_data$amt) / subj_data$amt) - (subj_data
$totaltime + subj_data$iti) # includes both handling time and latency
subj_data$sessions <- subj_sessions[, 2]
mean_subj_data <- round(mean(subj_data), 2)
subj_data <- rbind(subj_data, mean_subj_data)
subj_data$subject <- as.character(subj_data$subject)
subj_data[6, 1] <- "Mean"
subj_data$subject <- as.factor(subj_data$subject)

## Create LaTeX table
tex_data <- subj_data
tex_data$subject <- as.character(tex_data$subject)
tex_data[1, 1] <- "Blumchen"
tex_data[4, 1] <- "Puppi"
names(tex_data) <- c("Subject", "Species", "Sex", "Short delay", "Long delay", "Small amount", "Large amount", "III", "Small latency",
"Large latency", "Small handling", "Large handling", "Small total time", "Large total time", "Small short-term rate", "Large short-term
rate", "Small long-term rate", "Large long-term rate", "Sessions")
subj_table <- xtable(tex_data[, c(1, 2, 3, 19, 6, 7, 8, 4, 5, 9, 10, 11, 12)], digits = c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)) # create LaTeX table
align(subj_table) <- c('r', 'p{10mm}', 'p{8mm}', '>{\centering p{3mm}}', '>{\centering p{8mm}}', '>{\centering p{8mm}}', '>{\centering p{12mm}}', 'p
{8mm}', 'c', '>{\centering p{7mm}}', '>{\centering p{7mm}}', '>{\centering p{10mm}}', '>{\centering p{10mm}}', '>{\centering p{12mm}}', 'p
{12mm}}')

#####
### Comparative analysis
#####

## Generate plot of indifference points across species
comparative$species <- factor(as.character(comparative$species),
  levels = c("Pigeon", "Rat", "Lemur", "Tamarin", "Marmoset", "Capuchin", "Spider monkey", "Macaque", "Gorilla", "Orangutan", "Bonobo",
"Chimpanzee"))
comp_2v6 <- subset(comparative, exp_cond == "2v6")
comp_2v6 <- droplevels(comp_2v6)
comp_other <- subset(comparative, exp_cond != "2v6")
comp_other <- droplevels(comp_other)
levels(comp_other$species) <- c("Pigeon", "Rat", "Capuchin", "Spider \nmonkey", "Macaque", "Gorilla", "Orangutan") # include line

```

```
break for spider monkey label
```

```
# generate plots for 2v6 species and for other species
pdf(file = "figures/stevens_fig2.pdf", width = 11, height = 7)
par(mfcol = c(1, 2), las = 2, mai = c(1.3, 0.9, 0, 0))
comp_2v6.bw <- boxplot(indiff ~ species, data = comp_2v6, range = 0, col = "#0000ff22",
  ylab = "Mean adjusted delay at indifference (s)", xlab = NULL, ylim = c(0, 160),
  cex = 1.5, cex.axis = 1.2, cex.lab = 1.5, lwd = 0.5, boxwex = 0.5, outlier = F
)
condition
mean.com <- tapply(comp_2v6$indiff, comp_2v6$species, mean)
points(seq(comp_2v6.bw$n), mean.com, pch = 17, cex = 1.5, lwd = 2)
beeswarm(indiff ~ species, data = comp_2v6, pch = 1, cex = 1.15, col = "grey30", add = T) # overlay beeswarm plot of individual data
points
text(0.5, 160, labels = "A", cex = 1.5)

comp_other.bw <- boxplot(indiff ~ species, data = comp_other, range = 0, col = "#0000ff22",
  ylab = NULL, xlab = NULL, ylim = c(0, 160),
  cex = 1.5, cex.axis = 1.2, cex.lab = 1.5, lwd = 0.5, boxwex = 0.5, outlier = F
)
other conditions
mean.com <- tapply(comp_other$indiff, comp_other$species, mean)
points(seq(comp_other.bw$n), mean.com, pch = 17, cex = 1.5, lwd = 2)
beeswarm(indiff ~ species, data = comp_other, pch = 1, cex = 1.15, col = "grey30", add = T) # overlay beeswarm plot of individual data
points
text(0.5, 160, labels = "B", cex = 1.5)
dev.off()

## Correlate body mass and indifference points
comp.m <- aggregate(cbind(indiff, body_wt) ~ species, data = comparative, FUN = mean) # calculate mean indifference points and body
masses for all species
comp.m <- comp.m[order(comp.m$body_wt),] # reorder by body mass
comp.m$body_wt <- comp.m$body_wt / 1000 # convert g to kg
comp.m$logbody_wt <- log(comp.m$body_wt) # create log body mass column
comp.m$logindiff <- log(comp.m$indiff) # create log indifference point column
comp.lm <- lm(logindiff ~ logbody_wt, data = comp.m) # calculate linear regression of log body mass and
indifference points
comp.cor <- cor.test(comp.m$logbody_wt, comp.m$logindiff, method = "spearman") # calculate correlation test of log body mass and
indifference points
comp.r <- comp.cor$estimate # extract r from correlation
comp.p <- comp.cor$p.value # extract p-value from correlation
comp.n <- length(comp.m$species) # extract sample size from correlation

## Plot scatterplot and regression lines for body mass and indifference point comparison
pdf(file = "figures/stevens_fig3_raw.pdf", width = 9, height = 8)
par(mai = c(1, 1, 0.1, 0.1), mgp = c(3.5, 1, 0))
indiff_wt <- plot(indiff ~ body_wt, data = comp.m, log = "xy", # plot scatterplot
  ylab = "Mean adjusted delay at indifference (s)", xlab = "Body mass (kg)",
  ylim = c(4, 130), xlim = c(0.25, 150), las = 1, cex.lab = 1.8, pch = 1.5, pch = 16, cex = 1.4)
abline(lm(log10(indiff) ~ log10(body_wt), data = comp.m), lwd = 2)
```

```

text(comp.m$indiff ~ comp.m$body_wt, labels = comp.m$species, cex = 1.4, adj = c(0,1.8)) # add species names as labels
dev.off()

## Calculate phylogenetically independent contrasts
# use phylogeny from 10kTrees and Timetree
species <- "((((Macaque:29.999999,(Gorilla:10.123284,
(Bonobo:1.978625,Chimpanzee:1.978625):8.14466):5.079264,Orangutan:15.202547):14.797453):19.653115,(Spider_monkey:22.775847,
(Marmoset:16.314556,Tamarin:16.314556):4.76502,Capuchin:21.079576):1.696271):26.877268):22.633503,Lemur:72.286619):103.7,Rat:177):324.8,Pigeon:
species.tree.raw <- read.tree(text = species)
species.tree2 <- rotate(species.tree.raw, node = 21)
species.tree3 <- rotate(species.tree2, node = 19)
species.tree4 <- rotate(species.tree3, node = 17)
species.tree <- rotate(species.tree4, node = 20)

body_wt <- comp.m$logbody_wt[c(10,9,12,11,7,3,2,5,8,6,1,4)]
indiff <- comp.m$logindiff[c(10,9,12,11,7,3,2,5,8,6,1,4)]
indiff.contrast <- pic(indiff, species.tree)
body_wt.contrast <- pic(body_wt, species.tree)
indiff.contrast[which(body_wt.contrast < 0)] <- indiff.contrast[which(body_wt.contrast < 0)] * -1 # multiply indiffrence points
associated with negative log body mass contrasts by -1
body_wt.contrast <- abs(body_wt.contrast)
contrast.lm <- lm(indiff.contrast ~ body_wt.contrast - 1)
contrast.pval <- summary(contrast.lm)$coefficients[4]
contrast.fstat <- summary(contrast.lm)$fstatistic[1]
contrast.df1 <- summary(contrast.lm)$fstatistic[2]
contrast.df2 <- summary(contrast.lm)$fstatistic[3]

## Plot phylogeny
pdf(file = "figures/stevens_figS1.pdf", width = 9)
plot(species.tree, label.offset = 5)
dev.off()

## Plot scatterplot of contrasts
pdf(file = "figures/stevens_figS2.pdf", width = 9, height = 8)
par(mai = c(1.1, 1.2, 0.2, 0.2), mgp = c(4, 1, 0))
plot(indiff.contrast ~ body_wt.contrast,
      ylab = "Log mean adjusted delay contrasts", xlab = "Log body mass contrasts",
      cex.lab = 1.8, cex.axis = 1.5, pch = 16, cex = 1.6, las = 1)
abline(contrast.lm, lwd = 3)
dev.off()

# plot scatterplot
# plot regression line
# take absolute value of negative log body mass contrasts
# calculate regression (through the origin) of contrasts
# extract p-value
# extract F-statistic
# extract degrees of freedom (numerator)
# extract degrees of freedom (denominator)

```