```r
###################################################
### kaighobadi_stevens_rcode.R
### Created by Jeffrey R. Stevens on 1 Sept 2012 (jeffrey.r.stevens@gmail.com),
### Summary: This script calculates statistics and generates figures for the
###      analysis of women's intertemporal choice data across their ovulatory cycle.
### Instructions: Place this file and the data files (kaighobadi_stevens_data[2/3].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("kaighobadi_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:
###  kaighobadi_stevens_data2--intertemporal and risky choice data
###   subject - participant number
###   age - participant age
###   condition - experimental condition (experimental [male image] or control [neutral landscape image])
###   fertility - fertility status (peak or low)
###   date - date of experiment
###   time - time of day for experiment
###   session - session number (1 or 2)
###   task - order of experimental tasks
###   tasktype - experimental task (Discounting [intertemporal choice] or Risk [risky choice])
###   block - block number for blocks of questions
###   switchpt - indifference point calculated for question
###   amount1 - small amount
###   time1 - short delay
###   amount2 - large amount
###   time2 - long delay
###   prepost - flag for whether question occurs before or after exposure to images
###  kaighobadi_stevens_data3--attractiveness ratings
###   Subject - participant number
###   Fertility - fertility status (peak or low)
###   Image - image number with code for male image ("man") or landscape image ("L")
###   Rating - participant rating of image from 0 (very unattractive) to 9 (very attractive)
###   RT - response time--time between exposure to image and rating response (in ms)
###################################################


######################
# Clear variables, load libraries, include external functions, define local functions
######################
rm(list=ls())                        # clear all variables

library(epicalc, quietly = T)        # for aggregate with multiple functions
library(Hmisc, quietly = T)          # for xYplots
library(latticeExtra, quietly = T)  # for layer overlaying of plots
library(boot)                        # for bootstrapping CIs
library(nparLD)                      # for non-parametric repeated measures stats
```

```
######################
# Data input and preparation
######################
within <- read.csv("data/ov_indiff_within.csv")                              # input data
within$ffertility <- factor(within$fertility, labels = c("peak", "low"))            # create factor for fertility
within$fcondition <- factor(within$condition, labels = c("control", "experimental"))# create factor for condition
within$fprepost <- factor(within$prepost, labels = c("pre", "post"))                # create factor for prepost

######################
# Intertemporal choice
######################
disc <- subset(within, tasktype == "Discounting")               # select intertemporal choice (discounting) questions
disc <- subset(disc, substr(block, 8, 8) == "7")                # select data rows with the same question
disc$time2 <- as.character(disc$time2)                          # change time2 column to characters
disc_preexposure <- subset(disc, fprepost == "pre")             # extract pre-exposure questions
disc_preexposure$subject <- as.factor(disc_preexposure$subject) # convert subject to factor

###########
# Indifference point data
###########
disc_indiff_subj <- aggregate(disc_preexposure$switchpt, by = list(disc_preexposure$ffertility,  disc_preexposure$subject), FUN = c("mean",
"median", "length", "sd"))          # aggregate indifference points by subject, prepost, fertility, and condition
names(disc_indiff_subj) <- c("fertility", "subject", "indiff", "median", "n", "sd")
disc_indiff <- aggregate(disc_indiff_subj$indiff, by  = list(disc_indiff_subj$fertility), FUN = c("mean", "median", "length", "sd"))  #
aggregate indifference points by fertility
names(disc_indiff) <- c("fertility", "indiff", "median", "N", "sd")

boot_indiff_peak <- boot(subset(disc_indiff_subj, fertility == "peak")$indiff, function(u,i) mean(u[i]), R = 999)   # bootstrap peak
fertility data
boot_indiff_low <- boot(subset(disc_indiff_subj, fertility == "low")$indiff, function(u,i) mean(u[i]), R = 999)     # bootstrap low
fertility data
indiff_peak_ci <- boot.ci(boot_indiff_peak, type = "norm")                                                          # calculate 95%
confidence intervals
indiff_low_ci <- boot.ci(boot_indiff_low, type = "norm")                                                            # calculate 95%
confidence intervals
disc_indiff$ubci <- c(indiff_peak_ci$normal[3], indiff_low_ci$normal[3])                                            # add CIs to dataframe
disc_indiff$lbci <- c(indiff_peak_ci$normal[2], indiff_low_ci$normal[2])                                            # add CIs to dataframe

subject_col <- c("#0072B2", "#D55E00", "#009E73", "#E69F00", "#F0E442", "#CC79A7")  # create list of colors
subject_lty <- c(rep(c(2, 2), times = 7))                                           # create list of line types

disc_indiff_plot <- stripplot(jitter(switchpt) ~ ffertility, groups = subject, data = disc_preexposure,
    pch = 4, cex = 2, xlab = "Fertility", ylab = "Indifference point", aspect = 0.65,
    par.settings = list(axis.text = list(cex = 2), par.xlab.text = list(cex = 2.5),
        par.ylab.text = list(cex = 2.5), layout.heights = list(strip = 2)),
    panel = function(x, y, ...) {
        mean.values <<- tapply(y, x, mean, na.rm=T)                 # calculates means
        panel.stripplot(x, y, type = "b", lty = subject_lty, ...)
        panel.average(x, y, fun = mean, lwd = 4, col = "black", ...)# plot line connecting means
        panel.points(mean.values, pch = 16, cex = 2, col = "black") # plot means as diamonds
```

```
    }
)
addWithinCI <- layer_(panel.arrows(x, lbci[subscripts], x, ubci[subscripts], col = 'black', length = 0,
        unit = "native", angle = 90, code = 3, lwd = 2), data = disc_indiff, under = FALSE) # add layer for bootstrapped CIs
png(file = "figures/disc_indifference_within.png", width = 900, height = 600)               # create PNG file
plot(disc_indiff_plot + addWithinCI)
dev.off()

## Parametric statistics
disc_indiff_aov <- aov(switchpt ~ ffertility + Error(subject/ffertility), data = disc_preexposure)  # within-subjects ANOVA
shapiro.test(residuals(disc_indiff_aov$subject))                                                   # test assumption of normality of
residuals (failed)

## Non-parametric statistics
wilcox.test(switchpt ~ ffertility, data = disc_preexposure, paired = T)                            # Wilcoxan signed rank test

###########
# Difference score data
###########
disc_diff_subj <- aggregate(disc$switchpt, by = list(disc$ffertility, disc$fcondition, disc$subject), FUN = "diff")  # create difference
score (post - pre) as a function of fertility, condition, and subject
names(disc_diff_subj) <- c("fertility", "condition", "subject", "diff")
disc_diff_raw <- aggregate(disc_diff_subj$diff, by  = list(disc_diff_subj$fertility, disc_diff_subj$condition), FUN = c("mean", "median",
"length", "sd"))  # aggregate difference score by fertility and condition
names(disc_diff_raw) <- c("fertility", "condition", "diff", "median", "N", "sd")
disc_diff_expt_all <- subset(disc_diff_subj, condition == "experimental")   # create subset of experimental condition data
disc_diff_expt <- aggregate(disc_diff_expt_all$diff, by  = list(disc_diff_expt_all$fertility, disc_diff_expt_all$condition), FUN = c
("mean", "median", "length", "sd")) # aggregate experimental data by fertility and condition
disc_diff_control_all <- subset(disc_diff_subj, condition == "control")     # create subset of control condition data
disc_diff_control <- aggregate(disc_diff_control_all$diff, by  = list(disc_diff_control_all$fertility, disc_diff_control_all$condition),
FUN = c("mean", "median", "length", "sd")) # aggregate control data by fertility and condition
names(disc_diff_expt) <- names(disc_diff_control) <- c("ferility", "condition", "mean", "median", "N", "sd")
disc_diff <- data.frame(condition = c("experimental", "experimental", "control", "control"), rbind(disc_diff_expt, disc_diff_control)) #
combine experimental and control condition
disc_diff$diff <- disc_diff_raw[c(3, 4, 1, 2), 3]  # append raw difference means
boot_diff_peak_exp <- boot(subset(disc_diff_subj, fertility == "peak" & condition == "experimental")$diff, function(u,i) mean(u[i]), R =
999)    # bootstrap peak fertility/experimental data
boot_diff_peak_cont <- boot(subset(disc_diff_subj, fertility == "peak" & condition == "control")$diff, function(u,i) mean(u[i]), R =
999)        # bootstrap peak fertility/control data
boot_diff_low_exp <- boot(subset(disc_diff_subj, fertility == "low" & condition == "experimental")$diff, function(u,i) mean(u[i]), R =
999)        # bootstrap low fertility/experimental data
boot_diff_low_cont <- boot(subset(disc_diff_subj, fertility == "low" & condition == "control")$diff, function(u,i) mean(u[i]), R =
999)            # bootstrap low fertility/control data
diff_peak_exp_ci <- boot.ci(boot_diff_peak_exp, type = "norm")  # calculate 95% confidence intervals
diff_peak_cont_ci <- boot.ci(boot_diff_peak_cont, type = "norm")# calculate 95% confidence intervals
diff_low_exp_ci <- boot.ci(boot_diff_low_exp, type = "norm")    # calculate 95% confidence intervals
diff_low_cont_ci <- boot.ci(boot_diff_low_cont, type = "norm")  # calculate 95% confidence intervals
disc_diff$ubci <- c(diff_peak_exp_ci$normal[3], diff_low_exp_ci$normal[3], diff_peak_cont_ci$normal[3], diff_low_cont_ci$normal[3]) # add
CIs to dataframe
disc_diff$lbci <- c(diff_peak_exp_ci$normal[2], diff_low_exp_ci$normal[2], diff_peak_cont_ci$normal[2], diff_low_cont_ci$normal[2]) # add
```

```
CIs to dataframe

disc_diff_plot <- stripplot(jitter(diff, amount = 0.5) ~ fertility | condition, groups = subject, data = disc_diff_subj,
    pch = 4, cex = 2, subscripts = T, xlab = "Fertility", ylab = "Difference score (Post - Preexposure)", #main = "Intertemporal choice
(within subjects)",
    strip = strip.custom(factor.levels = c("Neutral Image", "Male Image"), par.strip.text = list(cex = 2)),
    par.settings = list(axis.text = list(cex = 2), par.xlab.text = list(cex = 2.5),
        par.ylab.text = list(cex = 2.5), layout.heights = list(strip = 2)),
    panel = function(x, y, ...) {
#       x2 <<- tapply(as.numeric(x), x, mean)
        mean.values <<- tapply(y, x, mean, na.rm=T)                     # calculates means
        panel.stripplot(x, y, type = "b", lty = subject_lty, ...)
        panel.average(x, y, fun = mean, lwd = 4, col = "black", ...)    # plot line connecting means
        panel.points(mean.values, pch = 16, cex = 1.5, col = "black")   # plot means as diamonds
        panel.abline(h = 0, lty = 2)                                    # mark 0 line
    }
)
addWithinCI <- layer_(panel.arrows(x, lbci[subscripts], x, ubci[subscripts], col = 'black', length = 0.3,  subscripts = T, monunit =
"native", angle = 0, code = 3, lwd = 2), data = disc_diff, under = FALSE)                                          # add layer for
bootstrapped CIs
png(file = "figures/disc_diff_score_within.png", width = 900, height = 700) # create PNG file
plot(disc_diff_plot + addWithinCI)
dev.off()

## Parametric stats
disc_diff_aov <- aov(diff ~ condition * fertility + Error(subject/(fertility)), data = disc_diff_subj) # mixed effect ANOVA
# shapiro.test(residuals(disc_diff_aov$subject)) # test assumption of normality of residuals (failed--unbalanced data)

## Non-parametric stats
my.t<- c(2:1)                  # create fertility predictions
my.pat <- rbind(c(2:1), c(2:1)) # create fertility predictions
disc_npar <- f1.ld.f1(y = disc_diff_subj$diff, time = disc_diff_subj$fertility, group = disc_diff_subj$condition, subject = disc_diff_subj
$subject, w.t=my.t, w.pat=my.pat, description = F)              # conduct non-parametric analysis

disc_diff_npaov <- disc_npar$ANOVA.test         # extract statistical test
disc_diff_pairedcomp <- disc_npar$pattern.time  # extract pairwise comparison

## Effect size of interaction effect (calculates Glass's delta [using low ferility standard deviation as control standard deviation]
because standard deviations differed slightly between high and low fertility conditions)
disc_diff_effect_size <- (disc_diff$mean[1] - disc_diff$mean[2])/disc_diff$sd[2]

######################
# Risky choice
######################
risk <- subset(within, tasktype == "Risk")      # select intertemporal choice (discounting) questions
risk <- subset(risk, substr(block, 8, 8) == "4")    # select data rows with the same question
risk_preexposure <- subset(risk, fprepost == "pre") # extract pre-exposure questions
risk_preexposure$subject <- as.factor(risk_preexposure$subject)

###########
```

```
# Indifference point data
###########

risk_indiff_subj <- aggregate(risk_preexposure$switchpt, by = list(risk_preexposure$ffertility,  risk_preexposure$subject), FUN = c("mean",
"median", "length", "sd"))          # aggregate indifference points by subject, prepost, fertility, and condition
names(risk_indiff_subj) <- c("fertility", "subject", "indiff", "median", "n", "sd")
risk_indiff <- aggregate(risk_indiff_subj$indiff, by  = list(risk_indiff_subj$fertility), FUN = c("mean", "median", "length", "sd"))  #
aggregate indifference points by fertility
names(risk_indiff) <- c("fertility", "indiff", "median", "N", "sd")

boot_indiff_peak <- boot(subset(risk_indiff_subj, fertility == "peak")$indiff, function(u,i) mean(u[i]), R = 999)   # bootstrap peak
fertility data
boot_indiff_low <- boot(subset(risk_indiff_subj, fertility == "low")$indiff, function(u,i) mean(u[i]), R = 999)     # bootstrap low
fertility data
indiff_peak_ci <- boot.ci(boot_indiff_peak, type = "norm")                                                          # calculate 95%
confidence intervals
indiff_low_ci <- boot.ci(boot_indiff_low, type = "norm")                                                            # calculate 95%
confidence intervals
risk_indiff$ubci <- c(indiff_peak_ci$normal[3], indiff_low_ci$normal[3])                                            # add CIs to dataframe
risk_indiff$lbci <- c(indiff_peak_ci$normal[2], indiff_low_ci$normal[2])                                            # add CIs to dataframe

risk_indiff_plot <- stripplot(jitter(switchpt) ~ ffertility, groups = subject, data = risk_preexposure,
    aspect = 0.65, pch = 4, cex = 2, xlab = "Fertility", ylab = "Indifference point", #main = "Risky choice (within subjects)",
    strip = strip.custom(factor.levels = c("Neutral Image", "Male Image"), par.strip.text = list(cex = 2)),
    par.settings = list(axis.text = list(cex = 2), par.xlab.text = list(cex = 2.5),
        par.ylab.text = list(cex = 2.5), layout.heights = list(strip = 2)),
    panel = function(x, y, ...) {
        x2 <- tapply(as.numeric(x), x, mean)
        mean.values <<- tapply(y, x, mean, na.rm=T)                         # calculates means
        panel.stripplot(x, y, type = "b", lty = subject_lty, ...)
        panel.average(x, y, fun = mean, lwd = 4, col = "black", ...)    # plot line between means
        panel.points(mean.values, pch = 16, cex = 1.5, col = "black")   # plot means as diamonds
    }
)
addWithinCI <- layer_(panel.segments(x, lbci[subscripts], x, ubci[subscripts], col = 'black', length = 0,
        unit = "native", angle = 90, code = 3, lwd = 2), data = risk_indiff, under = FALSE)  # add layer for bootstrapped CIs
png(file = "figures/risk_indifference_within.png", width = 900, height = 600)
plot(risk_indiff_plot + addWithinCI)
dev.off()

## Parametric stats
risk_preexp_aov <- aov(switchpt ~ ffertility + Error(subject/ffertility), data = risk_preexposure)  # within-subjects ANOVA
shapiro.test(residuals(risk_preexp_aov$subject))                # test assumption of normality of residuals (passed)
summary(risk_preexp_aov) # no main effects or interactions

## Non-parametric stats
wilcox.test(switchpt ~ ffertility, data = risk_preexposure, paired = T)

###########
# Difference score data
```

```
###########

risk_diff_subj <- aggregate(risk$switchpt, by = list(risk$ffertility, risk$fcondition, risk$subject), FUN = "diff")  # create difference
score (post - pre) as a function of fertility, condition, and subject
names(risk_diff_subj) <- c("fertility", "condition", "subject", "diff")
risk_diff_raw <- aggregate(risk_diff_subj$diff, by  = list(risk_diff_subj$fertility, risk_diff_subj$condition), FUN = c("mean", "median",
"length", "sd"))  # aggregate difference score by fertility and condition
names(risk_diff_raw) <- c("fertility", "condition", "diff", "median", "N", "sd")
risk_diff_expt_all <- subset(risk_diff_subj, condition == "experimental")   # create subset of experimental condition data
risk_diff_expt <- aggregate(risk_diff_expt_all$diff, by  = list(risk_diff_expt_all$fertility, risk_diff_expt_all$condition), FUN = c
("mean", "median", "length", "sd")) # aggregate experimental data by fertility and condition
risk_diff_control_all <- subset(risk_diff_subj, condition == "control")     # create subset of control condition data
risk_diff_control <- aggregate(risk_diff_control_all$diff, by  = list(risk_diff_control_all$fertility, risk_diff_control_all$condition),
FUN = c("mean", "median", "length", "sd")) # aggregate control data by fertility and condition
names(risk_diff_expt) <- names(risk_diff_control) <- c("ferility", "condition", "mean", "median", "N", "sd")
risk_diff <- data.frame(condition = c("experimental", "experimental", "control", "control"), rbind(risk_diff_expt, risk_diff_control)) #
combine experimental and control condition
risk_diff$diff <- risk_diff_raw[c(3, 4, 1, 2), 3]  # append raw difference means
boot_diff_peak_exp <- boot(subset(risk_diff_subj, fertility == "peak" & condition == "experimental")$diff, function(u,i) mean(u[i]), R =
999)    # bootstrap peak fertility/experimental data
boot_diff_peak_cont <- boot(subset(risk_diff_subj, fertility == "peak" & condition == "control")$diff, function(u,i) mean(u[i]), R =
999)        # bootstrap peak fertility/control data
boot_diff_low_exp <- boot(subset(risk_diff_subj, fertility == "low" & condition == "experimental")$diff, function(u,i) mean(u[i]), R =
999)        # bootstrap low fertility/experimental data
boot_diff_low_cont <- boot(subset(risk_diff_subj, fertility == "low" & condition == "control")$diff, function(u,i) mean(u[i]), R =
999)            # bootstrap low fertility/control data
diff_peak_exp_ci <- boot.ci(boot_diff_peak_exp, type = "norm")  # calculate 95% confidence intervals
diff_peak_cont_ci <- boot.ci(boot_diff_peak_cont, type = "norm")# calculate 95% confidence intervals
diff_low_exp_ci <- boot.ci(boot_diff_low_exp, type = "norm")    # calculate 95% confidence intervals
diff_low_cont_ci <- boot.ci(boot_diff_low_cont, type = "norm")  # calculate 95% confidence intervals
risk_diff$ubci <- c(diff_peak_exp_ci$normal[3], diff_low_exp_ci$normal[3], diff_peak_cont_ci$normal[3], diff_low_cont_ci$normal[3]) # add
CIs to dataframe
risk_diff$lbci <- c(diff_peak_exp_ci$normal[2], diff_low_exp_ci$normal[2], diff_peak_cont_ci$normal[2], diff_low_cont_ci$normal[2]) # add
CIs to dataframe

risk_diff_plot <- stripplot(jitter(diff) ~ fertility | condition, groups = subject, data = risk_diff_subj,
    pch = 4, cex = 2, subscripts = T, xlab = "Fertility", ylab = "Difference score (Post - Preexposure)", # main = "Risky choice (within
subjects)",
    strip = strip.custom(factor.levels = c("Neutral Image", "Male Image"), par.strip.text = list(cex = 2)),
    par.settings = list(axis.text = list(cex = 2), par.xlab.text = list(cex = 2.5),
        par.ylab.text = list(cex = 2.5), layout.heights = list(strip = 2)),
    panel = function(x, y, ...) {
        x2 <<- tapply(as.numeric(x), x, mean)
        mean.values <<- tapply(y, x, mean, na.rm=T)                     # calculates means
        panel.stripplot(x, y, type = "b", lty = subject_lty, ...)
        panel.average(x, y, fun = mean, lwd = 4, col = "black", ...)    # plot line between means
        panel.points(mean.values, pch = 16, cex = 1.5, col = "black")   # plot means as diamonds
        panel.abline(h = 0, lty = 2)                                    # mark 0 line
    }
)
```

```
addWithinCI <- layer_(panel.segments(x, lbci[subscripts], x, ubci[subscripts], col = 'black', length = 0,
        unit = "native", angle = 0, code = 3, lwd = 2), data = risk_diff, under = FALSE)  # add layer for bootstrapped CIs
png(file = "figures/risk_diff_score_within.png", width = 900, height = 700)
plot(risk_diff_plot + addWithinCI)
dev.off()

## Inferential stats
risk_diff_aov <- aov(diff ~ condition * fertility + Error(subject/(fertility)), data = risk_diff_subj) # mixed-effects ANOVA
# shapiro.test(residuals(risk_diff_aov$subject)) # test assumption of normality of residuals (failed--unbalanced data)

## Non-parametric stats
risk_npar <- f1.ld.f1(y = risk_diff_subj$diff, time = risk_diff_subj$fertility, group = risk_diff_subj$condition, subject = risk_diff_subj
$subject, description = F)  # conduct non-parametric analysis
risk_diff_npaov <- risk_npar$ANOVA.test     # extract ANOVA-like statistics

######################
# Attractiveness ratings
######################

rating_all <- read.csv("data/ov_rating_data.csv")                                  # input data
rating_all <- subset(rating_all, subject != 2, drop = T)
rating_all$ffertility <- factor(rating_all$fertility, labels = c("peak", "low"))       # create factor for fertility
ratings <- aggregate(rating_all$rating, by = list(rating_all$ffertility, rating_all$subject), FUN = "mean") # aggregate rating data by
fertility and subject
names(ratings) <- c("fertility", "subject", "rating")

rating_disc <- merge(disc_diff_subj, ratings)                                           # merge discounting and rating data frames
# rating_disc <- subset(rating_disc, subject != 2, drop = T)
rating_disc_peak_exp <- subset(rating_disc, fertility == "peak" & condition == "experimental")  # subset peak and experimental data
rating_disc_low_exp <- subset(rating_disc, fertility == "low" & condition == "experimental")    # subset low and experimental data
rating_disc_peak_cont <- subset(rating_disc, fertility == "peak" & condition == "control")      # subset peak and control data
rating_disc_low_cont <- subset(rating_disc, fertility == "low" & condition == "control")        # subset low and control data

## Ratings fertility by condition
rating_raw <- aggregate(rating_disc$rating, by  = list(rating_disc$condition, rating_disc$fertility), FUN = c("mean", "length", "sd"))  #
aggregate rating by fertility and condition
names(rating_raw) <- c("condition", "fertility", "rating", "N", "sd")
boot_rating_peak_exp <- boot(subset(rating_disc, fertility == "peak" & condition == "experimental")$rating, function(u,i) mean(u[i]), R =
999)    # bootstrap peak fertility/experimental data
boot_rating_peak_cont <- boot(subset(rating_disc, fertility == "peak" & condition == "control")$rating, function(u,i) mean(u[i]), R = 999)
 # bootstrap peak fertility/control data
boot_rating_low_exp <- boot(subset(rating_disc, fertility == "low" & condition == "experimental")$rating, function(u,i) mean(u[i]), R =
999)    # bootstrap low fertility/experimental data
boot_rating_low_cont <- boot(subset(rating_disc, fertility == "low" & condition == "control")$rating, function(u,i) mean(u[i]), R = 999)
   # bootstrap low fertility/control data
rating_peak_exp_ci <- boot.ci(boot_rating_peak_exp, type = "norm")  # calculate 95% confidence intervals
rating_peak_cont_ci <- boot.ci(boot_rating_peak_cont, type = "norm")# calculate 95% confidence intervals
rating_low_exp_ci <- boot.ci(boot_rating_low_exp, type = "norm")    # calculate 95% confidence intervals
rating_low_cont_ci <- boot.ci(boot_rating_low_cont, type = "norm")  # calculate 95% confidence intervals
rating_raw$ubci <- c(rating_peak_cont_ci$normal[3], rating_peak_exp_ci$normal[3], rating_low_cont_ci$normal[3], rating_low_exp_ci$normal
```

```
[3])    # add CIs to dataframe
rating_raw$lbci <- c(rating_peak_cont_ci$normal[2], rating_peak_exp_ci$normal[2], rating_low_cont_ci$normal[2], rating_low_exp_ci$normal
[2])    # add CIs to dataframe
rating_raw$ci <- rating_raw$ubci - rating_raw$rating

## Ratings fertility
rating_fert <- aggregate(rating_disc$rating, by  = list(rating_disc$fertility), FUN = c("mean", "length", "sd"))    # aggregate ratings by
fertility
names(rating_fert) <- c("fertility", "rating", "N", "sd")
boot_rating_peak <- boot(subset(rating_disc, fertility == "peak")$rating, function(u,i) mean(u[i]), R = 999)        # bootstrap peak
fertility data
boot_rating_low <- boot(subset(rating_disc, fertility == "low")$rating, function(u,i) mean(u[i]), R = 999)          # bootstrap low
fertility data
rating_peak_ci <- boot.ci(boot_rating_peak, type = "norm")                  # calculate 95% confidence intervals
rating_low_ci <- boot.ci(boot_rating_low, type = "norm")                    # calculate 95% confidence intervals
rating_fert$ubci <- c(rating_peak_ci$normal[3], rating_low_ci$normal[3])    # add CIs to dataframe
rating_fert$lbci <- c(rating_peak_ci$normal[2], rating_low_ci$normal[2])    # add CIs to dataframe
rating_fert$ci <- rating_fert$ubci - rating_fert$rating

## Ratings condition
rating_cond <- aggregate(rating_disc$rating, by  = list(rating_disc$condition), FUN = c("mean", "length", "sd"))        # aggregate
ratings by fertility
names(rating_cond) <- c("condition", "rating", "N", "sd")
boot_rating_exp <- boot(subset(rating_disc, condition == "experimental")$rating, function(u,i) mean(u[i]), R = 999)     # bootstrap
experimental condition data
boot_rating_cont <- boot(subset(rating_disc, condition == "control")$rating, function(u,i) mean(u[i]), R = 999)     # bootstrap control
condition data
rating_exp_ci <- boot.ci(boot_rating_exp, type = "norm")                    # calculate 95% confidence intervals
rating_cont_ci <- boot.ci(boot_rating_cont, type = "norm")                  # calculate 95% confidence intervals
rating_cond$ubci <- c(rating_cont_ci$normal[3], rating_exp_ci$normal[3])    # add CIs to dataframe
rating_cond$lbci <- c(rating_cont_ci$normal[2], rating_exp_ci$normal[2])    # add CIs to dataframe
rating_cond$ci <- rating_cond$ubci - rating_cond$rating
```