Supplemental code for Social Scaling of Stressor Impacts

Accompanying code for “Colony size buffers interactions between neonicotinoid exposure and cold stress in bumble bees”

#Queenright colony experiments

Load and clean data for queenright experiments

Load data

```
# Get colony demographic metadata
setwd("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/colonyInf
metadata <- read.csv("csvFiles/colonyMetaData.csv")

### Parse into separate treatments ### initialize
metadata$neonic <- 0
metadata$cold <- 0
metadata$neonic[str_detect(metadata$Treatment, "eonic")] <- 1
metadata$cold[str_detect(metadata$Treatment, "old")] <- 1
metadata$neonic <- as.numeric(metadata$neonic)
metadata$cold <- as.numeric(metadata$cold)

### Load behavioral tracking data
setwd("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/thermalRi
behData <- read.csv("behData_June2022.csv")
behData <- behData[complete.cases(behData[, c("porTimeMoving",
    "movingVelocity")]), ]  ## Remove rows with missing data
# tracking.metadata <- read.csv('trackingMetadata.csv')
tag.data <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors

# Correct name for one misentified colony
tag.data$colony[tag.data$colony == "BB246"] <- "BG246"

### Add size at tagging to metadata by reading taglists
### replace nans with 'worker'
tag.data$caste[is.na(tag.data$caste)] <- "worker"
tag.data$caste[tag.data$caste == ""] <- "worker"
tag.data$caste[tag.data$caste == "Worker"] <- "worker"
tag.data$caste[tag.data$caste == "Queen"] <- "queen"
tag.data$caste[tag.data$caste == "Male"] <- "male"

# Add colony position to tag.data
for (i in 1:length(metadata[, 1])) {
    tag.data$thermalRigPosition[tag.data$colony == metadata$colonyID[i]] <- metadata$positionInThermalR
}
```

```r
tag.data$unique_id <- paste(tag.data$block, tag.data$thermalRigPosition,
    tag.data$id, sep = "_")

for (i in 1:length(metadata[, 1])) {
    # Figure out how many bees in this colony
    metadata$colony.size.at.tagging[i] <- dim(subset(tag.data,
        colony == metadata$colonyID[i]))[1]
    metadata$new_queens[i] <- dim(subset(tag.data, colony ==
        metadata$colonyID[i] & caste == "queen" & founding_queen !=
        "Y"))[1]
    metadata$males[i] <- dim(subset(tag.data, colony == metadata$colonyID[i] &
        caste == "male"))[1]
}

metadata$reproductive <- as.factor((metadata$new_queens + metadata$males) >
    0)

reproductive.vs.size.model <- glm(reproductive ~ log10(colony.size.at.tagging),
    data = subset(metadata, colony.size.at.tagging > 0), family = "binomial")
summary(reproductive.vs.size.model)
```

```
##
## Call:
## glm(formula = reproductive ~ log10(colony.size.at.tagging), family = "binomial",
##     data = subset(metadata, colony.size.at.tagging > 0))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1884  -0.9461  -0.7742   1.3922   1.6155
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -1.4415     0.7391  -1.950   0.0511 .
## log10(colony.size.at.tagging)   0.6478     0.5418   1.196   0.2318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 74.726  on 57  degrees of freedom
## Residual deviance: 73.239  on 56  degrees of freedom
## AIC: 77.239
##
## Number of Fisher Scoring iterations: 4
```

```r
## Load thermal data
thermData <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors
thermData <- thermData[!is.na(thermData$block), ]  #Remove some missing data

behData$colony.id <- paste(behData$block, behData$rigPosition,
    sep = "_")
behData$uniqueID <- paste(behData$colony.id, behData$id, sep = "_")
```

```
## Fill in metadata to individual behavioral and colony
## level thermal data frames
for (i in 1:length(metadata[, 1])) {
    # print(i)
    bd.ind <- behData$rigPosition == metadata$positionInThermalRig[i] &
        behData$block == metadata$experimentalBlock[i]

    if (sum(bd.ind) > 0) {
        behData$neonic[bd.ind] <- as.logical(metadata$neonic[i])
        behData$cold[bd.ind] <- as.logical(metadata$cold[i])
        behData$colony.size.at.tagging[bd.ind] <- metadata$colony.size.at.tagging[i]
        behData$species[bd.ind] <- metadata$Species[i]
        behData$experimental.day[bd.ind] <- floor(behData[bd.ind,
            ]$dateNum) - min(floor(behData[bd.ind, ]$dateNum)) +
            1
        behData$time.from.start[bd.ind] <- behData[bd.ind, ]$dateNum -
            min(behData[bd.ind, ]$dateNum)
        worker.brood.ratio <- unique(behData$colony.size.at.tagging[bd.ind]/behData$numBrood[bd.ind])
        behData$worker.brood.ratio[bd.ind] <- worker.brood.ratio
        # Add reproductive state (binary)
        behData$reproductive[bd.ind] <- metadata$reproductive[i]

        # Identify first exposure
        rs <- behData$rampState[bd.ind]
        tfs <- behData$time.from.start[bd.ind]
        first.cold.time <- tfs[which(rs == 3)[1]]
        pre.exposure <- tfs < first.cold.time
        behData$pre.exposure[bd.ind] <- pre.exposure
        td.ind <- thermData$rigPosition == metadata$positionInThermalRig[i] &
            thermData$block == metadata$experimentalBlock[i]
        thermData$neonic[td.ind] <- as.logical(metadata$neonic[i])
        thermData$cold[td.ind] <- as.logical(metadata$cold[i])
        thermData$colony.size.at.tagging[td.ind] <- metadata$colony.size.at.tagging[i]
        thermData$species[td.ind] <- metadata$Species[i]
        thermData$experimental.day[td.ind] <- floor(thermData[td.ind,
            ]$dateNum) - min(floor(thermData[td.ind, ]$dateNum)) +
            1
        thermData$worker.brood.ratio[td.ind] <- worker.brood.ratio
    } else {
        # print(paste('no tracking data for colony ', i, ',
        # skipping', sep=''))
    }

}

# Remove missing values from thermData
thermData <- thermData[!is.na(thermData$block), ]

# Remove errant readings from thermData
thermData <- subset(thermData, (meanTempBrood_all - nestAirTemp) >
    -2)

# Add unique colony identifier to thermData
```

```r
thermData$colony.id <- paste(thermData$block, thermData$rigPosition,
    sep = "_")
behData$tod.hours <- (behData$dateNum%%1) * 24  #Convert timestamp to daily hours
behData <- subset(behData, framesTracked > 10)  #Select out low-quality tracking

# Variable conversion to factors
behData$id <- as.factor(behData$id)
behData$experimental.day <- as.factor(behData$experimental.day)
behData$neonic <- as.factor(behData$neonic)
behData$cold <- as.factor(behData$cold)

behData$trial <- paste(behData$colony.id, behData$date, sep = "_")

# Add marker for surface nesting species
behData$surfaceNesting <- behData$species == "griseocolis"

for (i in 1:length(tag.data$unique_id)) {
    behData$caste[behData$uniqueID == tag.data$unique_id[i]] <- tag.data$caste[i]
    behData$callow[behData$uniqueID == tag.data$unique_id[i]] <- tag.data$callow[i]
    behData$founding.queen[behData$uniqueID == tag.data$unique_id[i]] <- tag.data$founding_queen[i]
}

# create discrete rampState variable
behData$rampState_ch <- as.character(behData$rampState)
rmps <- as.character(seq(0, 5))
rep_names <- c("floating", "a_before", "ramp_down", "cold", "ramp_up",
    "b_after")
for (i in 1:length(rmps)) {
    behData$rampState_ch[behData$rampState_ch == rmps[i]] <- rep_names[i]
}
behData$rampState_ch <- as.factor(behData$rampState_ch)
```

Perform principal components analysis on within-nest behavioral metrics

```r
# Run principal components analysis on nest behavior
pc.vars <- c("distanceToCenterInstantaneous", "porTimeMoving",
    "movingVelocity", "medianDistancetoClosestBrood", "medianDistanceToClosestWaxpot",
    "medianDistanceToAllBrood", "medianDistanceToAllWaxpots",
    "degreeCentrality", "meanInteractionRate", "spatCorAll",
    "broodInfoRichness", "waxpotInfoRichness", "waxpotOccupancyRate",
    "broodOccupancyRate")

behData <- behData[complete.cases(behData[, pc.vars]), ]

pca.data.sub <- behData[, pc.vars]

# Variable transformation
pca.data.sub$porTimeMoving <- log10(1.05 - pca.data.sub$porTimeMoving)
lg.vars <- c("distanceToCenterInstantaneous", "medianDistancetoClosestBrood",
    "medianDistanceToClosestWaxpot", "medianDistanceToAllBrood",
    "medianDistanceToAllWaxpots", "degreeCentrality", "meanInteractionRate",
    "spatCorAll", "broodInfoRichness", "waxpotInfoRichness")
pca.data.sub[, lg.vars] <- log10(0.5 + pca.data.sub[, lg.vars])
```

```
nest.beh.pca <- prcomp(pca.data.sub, scale = TRUE, center = TRUE)
behData$PC1 <- nest.beh.pca$x[, 1]
behData$PC1 <- -behData$PC1  #NB Switch sign of PC1 to reflect 'spatial centrality' in other findings
behData$PC2 <- nest.beh.pca$x[, 2]
behData$PC3 <- nest.beh.pca$x[, 3]



# Subtract baseline PC1 for each colony
colonies <- unique(behData$colony.id)
baseline.pcs <- data.frame(matrix(nrow = length(colonies), ncol = 3))
colnames(baseline.pcs) <- c("colony", "pc1", "pc2")

for (i in 1:length(colonies)) {
    c.id <- behData$colony.id == colonies[i]
    first.ramp.time <- min(subset(behData, colony.id == colonies[i] &
        rampState > 0)$time.from.start)
    baseline.data <- subset(behData, colony.id == colonies[i] &
        rampState == 0 & time.from.start < first.ramp.time)
    if (length(baseline.data[, 1]) == 0) {
        print("no baseline data")
        print(i)
        print(colonies[i])
    }
    mean.pc1 <- mean(baseline.data$PC1)
    behData$PC1.res[c.id] <- behData$PC1[c.id] - mean.pc1

    mean.pc2 <- mean(baseline.data$PC2)
    behData$PC2.res[c.id] <- behData$PC2[c.id] - mean.pc2
    behData$first.ramp.time <- first.ramp.time

}

# pc loadings visulization prep
pc.loadings <- -nest.beh.pca$rotation[, 1:2]
pc.loadings <- pc.loadings[order(pc.loadings[, 1]), ]
pc.loadings <- as.data.frame(pc.loadings)
pc.loadings$variable <- rownames(pc.loadings)

setwd(output_folder)

pdf("FigS2a.pdf")
ggplot(data = pc.loadings, aes(x = reorder(variable, +PC1), y = PC1,
    fill = PC1)) + geom_bar(stat = "identity") + coord_flip() +
    scale_fill_distiller(palette = "RdYlBu") + theme_minimal()
dev.off()
```

```
## pdf
##   2
```

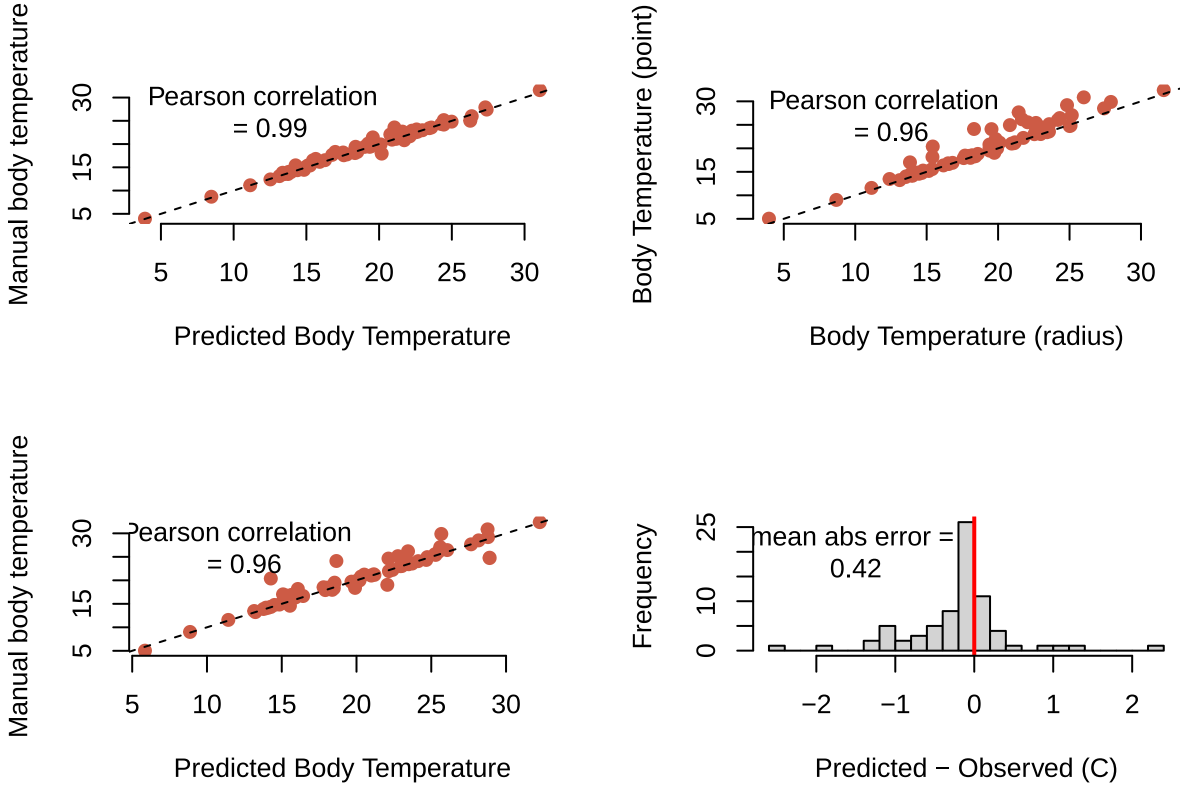
```
pdf("FigS2b.pdf")
ggplot(data = pc.loadings, aes(x = reorder(variable, +PC1), y = PC2,
    fill = PC2)) + geom_bar(stat = "identity") + coord_flip() +
    scale_fill_distiller(palette = "RdYlBu") + theme_minimal()
dev.off()
```

```
## pdf
##   2
```

```
behData$caste <- as.factor(behData$caste)
behData$uniqueID <- as.factor(behData$uniqueID)
behData$colony.id <- as.factor(behData$colony.id)
```

Body temperature validation

```
body_temp_ref_data <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingO

temp_val_plots <- function(body_temp_ref_data) {
    par(mfcol = c(2, 2))

    plot(ground_truth_radius ~ predicted_radius, data = body_temp_ref_data,
        axes = FALSE, pch = 19, col = "coral3", xlab = ("Predicted Body Temperature"),
        ylab = "Manual body temperature")
    abline(a = 0, b = 1, lty = 2)
    axis(1)
    axis(2)
    cc = cor(body_temp_ref_data$ground_truth_radius, body_temp_ref_data$predicted_radius,
        method = "pearson", use = "complete.obs")
    text(12, 27, paste("Pearson correlation\n  = ", signif(cc,
        2), sep = ""))

    plot(ground_truth_pt ~ predicted_pt, data = body_temp_ref_data,
        axes = FALSE, pch = 19, col = "coral3", xlab = ("Predicted Body Temperature"),
        ylab = "Manual body temperature")
    abline(a = 0, b = 1, lty = 2)
    axis(1)
    axis(2)
    cc = cor(body_temp_ref_data$ground_truth_pt, body_temp_ref_data$predicted_pt,
        method = "pearson", use = "complete.obs")
    text(12, 27, paste("Pearson correlation\n  = ", signif(cc,
        2), sep = ""))

    plot(ground_truth_pt ~ ground_truth_radius, data = body_temp_ref_data,
        axes = FALSE, pch = 19, col = "coral3", ylab = ("Body Temperature (point)"),
        xlab = "Body Temperature (radius)")
    abline(a = 0, b = 1, lty = 2)
    axis(1)
    axis(2)
    cc = cor(body_temp_ref_data$ground_truth_pt, body_temp_ref_data$ground_truth_radius,
        method = "pearson", use = "complete.obs")
    text(12, 27, paste("Pearson correlation\n  = ", signif(cc,
        2), sep = ""))

    hist(body_temp_ref_data$predicted_radius - body_temp_ref_data$ground_truth_radius,
        axes = FALSE, xlab = "Predicted - Observed (C)", ylab = "Frequency",
        breaks = 20, main = "")
    axis(2)
    abline(v = 0, lwd = 2, col = "red")
    mean_error = mean(abs(body_temp_ref_data$predicted_radius -
        body_temp_ref_data$ground_truth_radius), na.rm = TRUE)
    text(-1.5, 20, paste("mean abs error = \n", signif(mean_error,
        2), sep = ""))
    axis(1)
}

# Plot inline
temp_val_plots(body_temp_ref_data)
```



```
setwd(output_folder)
# Export figure panel
pdf("FigS10.pdf")
temp_val_plots(body_temp_ref_data)
dev.off()
```

```
## pdf
##   2
```

Aggregate behavioral effects

```
pc1.model <- lmer(PC1 ~ nestTemp + (1 | colony.id/uniqueID) +
    (1 | species) + (1 | block) + (1 | experimental.day), data = subset(behData,
    neonic == FALSE), REML = FALSE)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00663284 (tol = 0.002, component 1)
```

```
pc2.model <- lmer(PC2 ~ nestTemp + (1 | colony.id/uniqueID) +
    (1 | species) + (1 | experimental.day), data = subset(behData,
    neonic == FALSE & rampState > 0), REML = FALSE)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0039169 (tol = 0.002, component 1)
```

```
# NB removed 'block' random effect here to improve model
# convergence

summary(pc1.model)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC1 ~ nestTemp + (1 | colony.id/uniqueID) + (1 | species) + (1 |
##     block) + (1 | experimental.day)
##    Data: subset(behData, neonic == FALSE)
##
##       AIC       BIC    logLik  deviance  df.resid
##  256727.5  256800.0 -128355.7  256711.5     63858
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4296 -0.5577  0.0821  0.6631  4.9196
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 0.74655  0.8640
##  colony.id          (Intercept) 1.32105  1.1494
##  block              (Intercept) 0.29056  0.5390
##  species            (Intercept) 0.15271  0.3908
##  experimental.day   (Intercept) 0.01646  0.1283
##  Residual                       3.16535  1.7791
## Number of obs: 63866, groups:
## uniqueID:colony.id, 716; colony.id, 21; block, 14; species, 4; experimental.day, 3
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)  2.369e+00  4.019e-01  2.596e+00   5.894   0.0145 *
## nestTemp    -7.552e-02  2.705e-03  6.233e+04 -27.920   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##          (Intr)
## nestTemp -0.176
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00663284 (tol = 0.002, component 1)
```

```
summary(pc2.model)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC2 ~ nestTemp + (1 | colony.id/uniqueID) + (1 | species) + (1 |
##     experimental.day)
##    Data: subset(behData, neonic == FALSE & rampState > 0)
##
##      AIC      BIC   logLik deviance df.resid
##  42091.6  42144.0 -21038.8  42077.6    13247
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6740 -0.6812 -0.0525  0.6454  3.9710
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 0.25828  0.5082
##  colony.id          (Intercept) 0.39827  0.6311
##  species            (Intercept) 0.30924  0.5561
##  experimental.day   (Intercept) 0.02034  0.1426
##  Residual                       1.29721  1.1390
## Number of obs: 13254, groups:
## uniqueID:colony.id, 692; colony.id, 21; species, 4; experimental.day, 3
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)  2.429e+00  3.559e-01  2.963e+00   6.826  0.00669 **
## nestTemp    -8.234e-02  2.976e-03  1.256e+04 -27.672  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## nestTemp -0.197
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0039169 (tol = 0.002, component 1)
```

Baseline model - test whether exposure has any effects pre-temperature stress

```
pc1.baseline <- lmer(PC1 ~ neonic * log10(colony.size.at.tagging) +
    (1 | block) + (1 | caste) + (1 | species/colony.id/uniqueID),
    data = subset(behData, rampState == 0 & time.from.start <
        first.ramp.time), REML = FALSE)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00293511 (tol = 0.002, component 1)
```

```
summary(pc1.baseline)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC1 ~ neonic * log10(colony.size.at.tagging) + (1 | block) +
##     (1 | caste) + (1 | species/colony.id/uniqueID)
##    Data: subset(behData, rampState == 0 & time.from.start < first.ramp.time)
##
##      AIC      BIC   logLik deviance df.resid
## 184911.3 184998.8 -92445.7 184891.3    46306
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4108 -0.5596  0.0657  0.6489  4.5172
##
## Random effects:
##  Groups                       Name        Variance Std.Dev.
##  uniqueID:(colony.id:species) (Intercept) 1.09448  1.0462
##  colony.id:species            (Intercept) 0.62475  0.7904
##  block                        (Intercept) 0.43625  0.6605
##  species                      (Intercept) 0.41058  0.6408
##  caste                        (Intercept) 0.02557  0.1599
##  Residual                                 2.96743  1.7226
## Number of obs: 46316, groups:
## uniqueID:(colony.id:species), 1248; colony.id:species, 42; block, 15; species, 4; caste, 3
##
## Fixed effects:
##                                         Estimate Std. Error      df t value
## (Intercept)                               0.1772     0.8286 24.0020   0.214
## neonicTRUE                                1.7811     0.9681 36.6911   1.840
## log10(colony.size.at.tagging)            -0.1442     0.5245 36.9546  -0.275
## neonicTRUE:log10(colony.size.at.tagging)  -1.5602     0.6802 32.8656  -2.294
##                                         Pr(>|t|)
## (Intercept)                               0.8325
## neonicTRUE                                0.0739 .
## log10(colony.size.at.tagging)             0.7850
## neonicTRUE:log10(colony.size.at.tagging)   0.0283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) nnTRUE l10(..
## neonicTRUE  -0.482
## lg10(cl...) -0.840  0.455
## nTRUE:10(..  0.476 -0.955 -0.505
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00293511 (tol = 0.002, component 1)
```

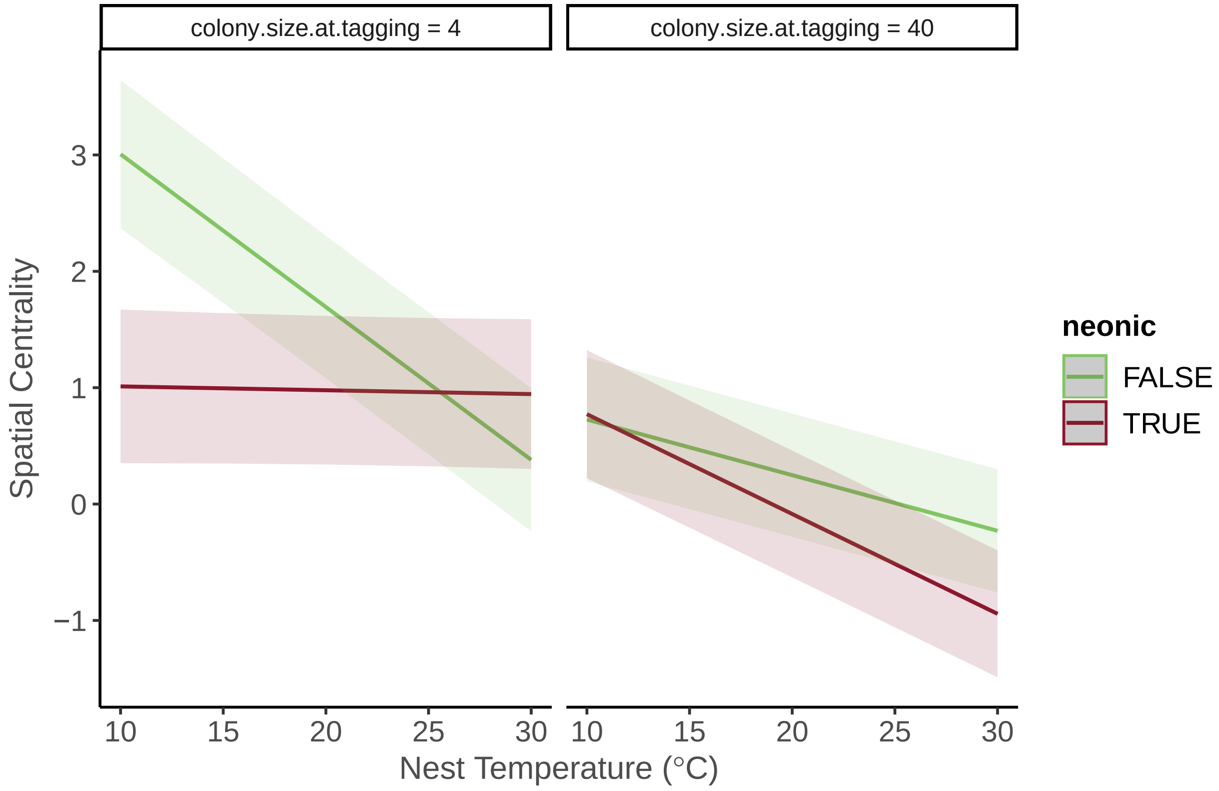
Active response model - effects of treatment during active cooling

```r
pc1.active.response.model <- lmer(PC1 ~ log10(colony.size.at.tagging) *
    neonic * nestTemp + (1 | experimental.day) + (1 | block) +
    (1 | species) + (1 | colony.id/uniqueID) + (1 | caste), data = subset(behData,
    rampState %in% c(1, 2, 3) & caste %in% c("worker", "queen")))

summary(pc1.active.response.model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PC1 ~ log10(colony.size.at.tagging) * neonic * nestTemp + (1 |
##     experimental.day) + (1 | block) + (1 | species) + (1 | colony.id/uniqueID) +
##     (1 | caste)
##    Data: subset(behData, rampState %in% c(1, 2, 3) & caste %in% c("worker",
##     "queen"))
##
## REML criterion at convergence: 49388.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7714 -0.5426  0.0700  0.6337  3.4624
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 1.4624   1.2093
##  colony.id          (Intercept) 0.8682   0.9317
##  block              (Intercept) 0.7168   0.8466
##  species            (Intercept) 1.1039   1.0507
##  experimental.day   (Intercept) 0.3181   0.5640
##  caste              (Intercept) 0.1179   0.3433
##  Residual                       2.5717   1.6036
## Number of obs: 12509, groups:
## uniqueID:colony.id, 1084; colony.id, 42; block, 15; species, 4; experimental.day, 3; caste, 2
##
## Fixed effects:
##                                                Estimate Std. Error
## (Intercept)                                     6.19372    1.37028
## log10(colony.size.at.tagging)                  -3.11416    0.78877
## neonicTRUE                                     -5.50255    1.58819
## nestTemp                                       -0.18153    0.03044
## log10(colony.size.at.tagging):neonicTRUE        3.70027    1.08343
## log10(colony.size.at.tagging):nestTemp          0.08347    0.01856
## neonicTRUE:nestTemp                             0.22789    0.04364
## log10(colony.size.at.tagging):neonicTRUE:nestTemp   -0.16594    0.02801
##                                                        df t value Pr(>|t|)
## (Intercept)                                      32.30369   4.520 7.83e-05
## log10(colony.size.at.tagging)                    62.97617  -3.948 0.000201
## neonicTRUE                                       86.50730  -3.465 0.000828
## nestTemp                                       8174.76380  -5.964 2.56e-09
## log10(colony.size.at.tagging):neonicTRUE         70.52601   3.415 0.001060
## log10(colony.size.at.tagging):nestTemp         8789.86681   4.497 6.99e-06
## neonicTRUE:nestTemp                            8336.95162   5.221 1.82e-07
## log10(colony.size.at.tagging):neonicTRUE:nestTemp 9189.52748  -5.924 3.26e-09
##
```

```
## (Intercept)                                       ***
## log10(colony.size.at.tagging)                     ***
## neonicTRUE                                        ***
## nestTemp                                          ***
## log10(colony.size.at.tagging):neonicTRUE          **
## log10(colony.size.at.tagging):nestTemp            ***
## neonicTRUE:nestTemp                               ***
## log10(colony.size.at.tagging):neonicTRUE:nestTemp ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                (Intr) lg10(...) nnTRUE nstTmp lg10(...):TRUE lg10(...):T nTRUE:
## lg10(cl...)    -0.820
## neonicTRUE     -0.517  0.536
## nestTemp       -0.532  0.554     0.471
## lg10(...):TRUE  0.500 -0.559    -0.962 -0.417
## lg10(...):T     0.524 -0.567    -0.462 -0.980  0.426
## nncTRUE:nsT     0.372 -0.386    -0.662 -0.697  0.609          0.683
## l10(...):TRUE: -0.349  0.376     0.647  0.650 -0.627         -0.663      -0.973
```

```
plot_model(pc1.active.response.model, type = "pred", terms = c("nestTemp",
    "neonic", "colony.size.at.tagging [4, 40]"), ci.lvl = 0.5,
    colors = list(ctrl_color, imid_color), title = "", axis.title = c(expression(paste("Nest Temperatur
        degree, "C)", sep = "")), "Spatial Centrality"))
```



```r
# Export figure panel
setwd(output_folder)
pdf("Fig3_ab.pdf")
plot_model(pc1.active.response.model, type = "pred", terms = c("nestTemp",
    "neonic", "colony.size.at.tagging [4, 40]"), ci.lvl = 0.5,
    colors = list(ctrl_color, imid_color), title = "", axis.title = c(expression(paste("Nest Temperatur
        degree, "C)", sep = "")), "Spatial Centrality (PC1)"),
    show.legend = F)
dev.off()
```

```
## pdf
##   2
```

Recovery model - effects of treatment on behavioral recovery after cold exposure

```r
sub.dat <- subset(behData, rampState %in% c(1, 5) & cold == TRUE)
sub.dat$rampState <- as.factor(sub.dat$rampState)
pc1.recovery.model <- lmer(PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) +
    (1 | colony.id/uniqueID), data = sub.dat, REML = FALSE)
summary(pc1.recovery.model)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) +
##     (1 | colony.id/uniqueID)
##    Data: sub.dat
##
##      AIC      BIC   logLik deviance df.resid
##  13525.0  13592.1  -6751.5  13503.0     3270
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3031 -0.5517  0.0608  0.6219  3.1433
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 1.311    1.145
##  colony.id          (Intercept) 1.534    1.239
##  Residual                       2.858    1.691
## Number of obs: 3281, groups:  uniqueID:colony.id, 560; colony.id, 21
##
## Fixed effects:
##                                                               Estimate
## (Intercept)                                                     1.5541
## rampState_chb_after                                             1.8789
## neonicTRUE                                                     -0.1569
## log10(colony.size.at.tagging)                                  -0.5644
## rampState_chb_after:neonicTRUE                                 -3.5832
## rampState_chb_after:log10(colony.size.at.tagging)              -0.9422
## neonicTRUE:log10(colony.size.at.tagging)                       -0.7793
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)    2.3034
##                                                              Std. Error
## (Intercept)                                                      1.4934
## rampState_chb_after                                              0.4796
## neonicTRUE                                                       2.0750
## log10(colony.size.at.tagging)                                    1.0614
## rampState_chb_after:neonicTRUE                                   0.6491
## rampState_chb_after:log10(colony.size.at.tagging)                0.2953
## neonicTRUE:log10(colony.size.at.tagging)                         1.4793
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)     0.4011
##                                                                     df t value
## (Intercept)                                                    25.0239   1.041
## rampState_chb_after                                          2836.3541   3.917
## neonicTRUE                                                     24.1768  -0.076
## log10(colony.size.at.tagging)                                  22.9009  -0.532
## rampState_chb_after:neonicTRUE                               2879.7773  -5.520
## rampState_chb_after:log10(colony.size.at.tagging)            2844.3204  -3.191
## neonicTRUE:log10(colony.size.at.tagging)                       22.3783  -0.527
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging) 2905.3936   5.743
##                                                              Pr(>|t|)
## (Intercept)                                                   0.30801
## rampState_chb_after                                          9.17e-05 ***
## neonicTRUE                                                    0.94034
## log10(colony.size.at.tagging)                                 0.60002
## rampState_chb_after:neonicTRUE                               3.69e-08 ***
## rampState_chb_after:log10(colony.size.at.tagging)             0.00144 **
## neonicTRUE:log10(colony.size.at.tagging)                      0.60350
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging) 1.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rmpS__ nnTRUE l10(.. rmS__:TRUE rS__:1 nTRUE:
## rmpStt_chb_ -0.217
## neonicTRUE  -0.720  0.156
## lg10(cl...) -0.960  0.180  0.691
## rmpS__:TRUE  0.160 -0.739 -0.219 -0.133
## rS__:10(...  0.211 -0.982 -0.152 -0.181  0.726
## nTRUE:10(..  0.689 -0.129 -0.960 -0.718  0.183      0.130
## rS__:TRUE:1 -0.155  0.723  0.213  0.133 -0.977     -0.736 -0.186
```

```
pc2.recovery.model <- lmer(PC2 ~ rampState_ch * neonic * colony.size.at.tagging +
    (1 | colony.id/uniqueID), data = sub.dat, REML = FALSE)
summary(pc2.recovery.model)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula:
## PC2 ~ rampState_ch * neonic * colony.size.at.tagging + (1 | colony.id/uniqueID)
##    Data: sub.dat
##
##      AIC      BIC   logLik deviance df.resid
##  10348.6  10415.6  -5163.3  10326.6     3270
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -3.0779 -0.6323 -0.0433  0.6046  4.1242 
## 
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 0.3291   0.5737  
##  colony.id          (Intercept) 0.3041   0.5515  
##  Residual                       1.1444   1.0698  
## Number of obs: 3281, groups:  uniqueID:colony.id, 560; colony.id, 21
## 
## Fixed effects:
##                                                         Estimate Std. Error
## (Intercept)                                            5.663e-01  3.226e-01
## rampState_chb_after                                   -6.916e-01  1.301e-01
## neonicTRUE                                             3.163e-01  4.221e-01
## colony.size.at.tagging                                -1.253e-02  8.076e-03
## rampState_chb_after:neonicTRUE                         3.090e-01  1.724e-01
## rampState_chb_after:colony.size.at.tagging             8.203e-03  2.467e-03
## neonicTRUE:colony.size.at.tagging                      1.073e-02  9.989e-03
## rampState_chb_after:neonicTRUE:colony.size.at.tagging -6.766e-03  3.056e-03
##                                                               df t value
## (Intercept)                                            2.789e+01   1.755
## rampState_chb_after                                    2.952e+03  -5.316
## neonicTRUE                                             2.800e+01   0.749
## colony.size.at.tagging                                 2.221e+01  -1.552
## rampState_chb_after:neonicTRUE                         2.970e+03   1.792
## rampState_chb_after:colony.size.at.tagging             2.978e+03   3.326
## neonicTRUE:colony.size.at.tagging                      2.239e+01   1.075
## rampState_chb_after:neonicTRUE:colony.size.at.tagging  3.008e+03  -2.214
##                                                       Pr(>|t|)    
## (Intercept)                                           0.090162 .  
## rampState_chb_after                                   1.14e-07 ***
## neonicTRUE                                            0.459943    
## colony.size.at.tagging                                0.134858    
## rampState_chb_after:neonicTRUE                        0.073178 .  
## rampState_chb_after:colony.size.at.tagging            0.000893 ***
## neonicTRUE:colony.size.at.tagging                     0.293959    
## rampState_chb_after:neonicTRUE:colony.size.at.tagging 0.026905 *  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Correlation of Fixed Effects:
##             (Intr) rmpS__ nnTRUE cln... rmS__:TRUE rS__:. nTRUE:
## rmpStt_chb_ -0.278                                                 
## neonicTRUE  -0.764  0.213                                          
## clny.sz.t.t -0.806  0.175  0.616                                   
## rmpS__:TRUE  0.210 -0.755 -0.291 -0.132                            
## rmpSt__:...  0.242 -0.899 -0.185 -0.183  0.679                     
## nncTRUE:...  0.652 -0.142 -0.776 -0.809  0.178      0.148          
## rS__:TRUE:. -0.195  0.726  0.244  0.147 -0.865     -0.807 -0.197
```

```
# Export panels
setwd(output_folder)
pdf("Fig3_cd.pdf")
```

```
plot_model(pc1.recovery.model, type = "pred", terms = c("rampState_ch",
    "neonic", "colony.size.at.tagging [4, 40]]"), ci.lvl = 0.5,
    colors = list(ctrl_color, imid_color), title = "", axis.title = c("Observation Period",
        "Spatial Centrality (PC1)"), show.legend = F)
dev.off()
```

```
## pdf
##   2
```

#Worker group models Characterize treatment effects by initial behavioral groups (bees that are on vs. off the nest at beginning of trials)

```
### look at effects during ramp down by whether bees were
### already on the nest
sub.dat <- subset(behData, rampState %in% c(1, 2, 3) & cold ==
    TRUE)  #Subset to data in initial cold exposure

colony.size.cuttoff <- 30
# Define workers that were on nest at ambient temperatures
sd <- subset(sub.dat, rampState < 2 & nestTemp > 23)
tmp <- tapply(sd$medianDistancetoClosestBrood, sd$uniqueID, median,
    na.rm = TRUE)  #Get averages in this time period for all bees

incubator.list <- names(tmp)[which(tmp < pixel_conversion)]
non.incubator.list <- names(tmp)[which(tmp > pixel_conversion *
    2)]

# Model responses with 'incubator' as an explicit
# interaction term
sub.dat$incubator <- NA
sub.dat$incubator[sub.dat$uniqueID %in% incubator.list] <- "inc"
sub.dat$incubator[sub.dat$uniqueID %in% non.incubator.list] <- "ninc"

pc1.response.model.by.incubator <- lmer(PC1 ~ nestTemp * neonic *
    incubator + (1 | colony.id/uniqueID), data = subset(sub.dat,
    caste == "worker" & incubator %in% c("inc", "ninc") & colony.size.at.tagging <
        colony.size.cuttoff), REML = FALSE)
summary(pc1.response.model.by.incubator)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC1 ~ nestTemp * neonic * incubator + (1 | colony.id/uniqueID)
##    Data: subset(sub.dat, caste == "worker" & incubator %in% c("inc", "ninc") &
##     colony.size.at.tagging < colony.size.cuttoff)
##
##      AIC      BIC   logLik deviance df.resid
##   5558.0   5616.2  -2768.0   5536.0     1454
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8545 -0.4826  0.0250  0.5353  3.5859
##
```

```
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 1.2910   1.1362
##  colony.id          (Intercept) 0.3442   0.5867
##  Residual                       2.2086   1.4862
## Number of obs: 1465, groups:  uniqueID:colony.id, 101; colony.id, 13
##
## Fixed effects:
##                                   Estimate Std. Error         df t value
## (Intercept)                      3.516e+00  5.224e-01  4.187e+01   6.731
## nestTemp                        -5.559e-02  1.755e-02  1.372e+03  -3.168
## neonicTRUE                      -2.911e-01  6.644e-01  3.523e+01  -0.438
## incubatorninc                   -4.336e-02  7.086e-01  3.859e+02  -0.061
## nestTemp:neonicTRUE              7.875e-03  2.182e-02  1.371e+03   0.361
## nestTemp:incubatorninc          -1.207e-01  2.917e-02  1.405e+03  -4.136
## neonicTRUE:incubatorninc        -3.829e+00  9.483e-01  3.316e+02  -4.038
## nestTemp:neonicTRUE:incubatorninc  1.064e-01  3.730e-02  1.426e+03   2.851
##                                 Pr(>|t|)
## (Intercept)                     3.58e-08 ***
## nestTemp                         0.00157 **
## neonicTRUE                       0.66401
## incubatorninc                    0.95124
## nestTemp:neonicTRUE              0.71828
## nestTemp:incubatorninc          3.74e-05 ***
## neonicTRUE:incubatorninc        6.71e-05 ***
## nestTemp:neonicTRUE:incubatorninc  0.00442 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) nstTmp nnTRUE incbtr nsT:TRUE nstTm: nTRUE:
## nestTemp    -0.613
## neonicTRUE  -0.786  0.482
## incubatrnnc -0.559  0.449  0.440
## nstTmp:TRUE  0.493 -0.804 -0.597 -0.361
## nstTmp:ncbt  0.363 -0.601 -0.285 -0.750  0.483
## nncTRUE:ncb  0.418 -0.335 -0.537 -0.747  0.417    0.560
## nstTm:TRUE: -0.284  0.470  0.344  0.587 -0.585   -0.782 -0.743
```

```
setwd("/Users/jamescrall/Dropbox/Work/_Writing/2022_SocialScaling/Figures")
pdf("Fig5A.pdf")
dat <- subset(sub.dat, rampState == 3 & bodyTemp_localAv > 10 &
    colony.size.at.tagging < colony.size.cuttoff)
beeswarm(bodyTemp_localAv ~ incubator, data = dat, pch = 19,
    cex = 0.75, col = c("darkgoldenrod2", "mediumpurple3"), axes = FALSE,
    ann = FALSE)
bxplot(bodyTemp_localAv ~ incubator, data = dat, add = TRUE,
    probs = 0.5)
axis(2)
dev.off()
```

```
## pdf
##   2
```

```
pdf("Fig5B.pdf")
plot_model(pc1.response.model.by.incubator, type = "pred", terms = c("nestTemp",
    "neonic", "incubator"), ci.lvl = 0.5, colors = list(ctrl_color,
    imid_color), axis.title = c(expression(paste("Nest Temperature (",
    degree, "C)", sep = "")), "Spatial Centrality (PC1)"), show.legend = F,
    title = "")
dev.off()
```

```
## pdf
##   2
```

```
# Subset data
sub.dat <- subset(behData, rampState %in% c(1, 5) & cold == TRUE)
sub.dat$rampState <- as.factor(sub.dat$rampState)

# Separate model for incubators (bees on nest)
pc1.recovery.model.incubators <- lmer(PC1 ~ rampState_ch * neonic *
    log10(colony.size.at.tagging) + experimental.day + (1 | uniqueID) +
    (1 | colony.id), data = subset(sub.dat, caste == "worker" &
    uniqueID %in% incubator.list), REML = FALSE)
summary(pc1.recovery.model.incubators)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) +
##     experimental.day + (1 | uniqueID) + (1 | colony.id)
##    Data: subset(sub.dat, caste == "worker" & uniqueID %in% incubator.list)
##
##      AIC      BIC   logLik deviance df.resid
##   3681.4   3744.3  -1827.7   3655.4      926
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0867 -0.4323  0.1868  0.6505  2.5209
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  uniqueID  (Intercept) 0.2770   0.5263
##  colony.id (Intercept) 0.2106   0.4589
##  Residual              2.6034   1.6135
## Number of obs: 939, groups:  uniqueID, 151; colony.id, 15
##
## Fixed effects:
##                                                               Estimate
## (Intercept)                                                    3.07585
## rampState_chb_after                                           -0.03334
## neonicTRUE                                                    -0.49768
## log10(colony.size.at.tagging)                                 -0.97400
## experimental.day2                                              0.32133
## experimental.day3                                              0.31286
## rampState_chb_after:neonicTRUE                                 0.40138
## rampState_chb_after:log10(colony.size.at.tagging)             -0.18455
## neonicTRUE:log10(colony.size.at.tagging)                       0.23139
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)  -0.37846
##                                                              Std. Error
## (Intercept)                                                     1.04321
## rampState_chb_after                                             0.77982
## neonicTRUE                                                      1.27024
## log10(colony.size.at.tagging)                                   0.60943
## experimental.day2                                               0.47598
## experimental.day3                                               0.49287
## rampState_chb_after:neonicTRUE                                  1.04133
## rampState_chb_after:log10(colony.size.at.tagging)               0.47605
## neonicTRUE:log10(colony.size.at.tagging)                        0.84266
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)    0.64992
##                                                                     df t value
## (Intercept)                                                   32.12358   2.948
## rampState_chb_after                                          859.05955  -0.043
## neonicTRUE                                                    14.59760  -0.392
## log10(colony.size.at.tagging)                                 13.26451  -1.598
## experimental.day2                                            796.08981   0.675
## experimental.day3                                            780.81870   0.635
## rampState_chb_after:neonicTRUE                               865.26567   0.385
## rampState_chb_after:log10(colony.size.at.tagging)            863.94173  -0.388
## neonicTRUE:log10(colony.size.at.tagging)                      10.36874   0.275
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging) 867.44245  -0.582
##                                                              Pr(>|t|)
## (Intercept)                                                   0.00591 **
## rampState_chb_after                                           0.96591
## neonicTRUE                                                    0.70086
## log10(colony.size.at.tagging)                                 0.13353
## experimental.day2                                             0.49981
## experimental.day3                                             0.52576
## rampState_chb_after:neonicTRUE                                0.70000
## rampState_chb_after:log10(colony.size.at.tagging)             0.69836
## neonicTRUE:log10(colony.size.at.tagging)                      0.78902
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)  0.56050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rmpS__ nnTRUE l10(.. expr.2 expr.3 rmS__:TRUE rS__:1 nTRUE:
## rmpStt_chb_ -0.477
## neonicTRUE  -0.625  0.391
## lg10(cl...) -0.862  0.478  0.715
## exprmntl.d2 -0.437 -0.024 -0.081 -0.017
## exprmntl.d3 -0.437  0.013 -0.087 -0.017  0.966
## rmpS__:TRUE  0.328 -0.755 -0.502 -0.359  0.090  0.051
## rS__:10(...  0.462 -0.982 -0.375 -0.473  0.021 -0.024  0.743
## nTRUE:10(..  0.574 -0.340 -0.963 -0.725  0.118  0.129  0.448      0.334
## rS__:TRUE:1 -0.285  0.726  0.484  0.349 -0.140 -0.097 -0.974     -0.739 -0.456
```

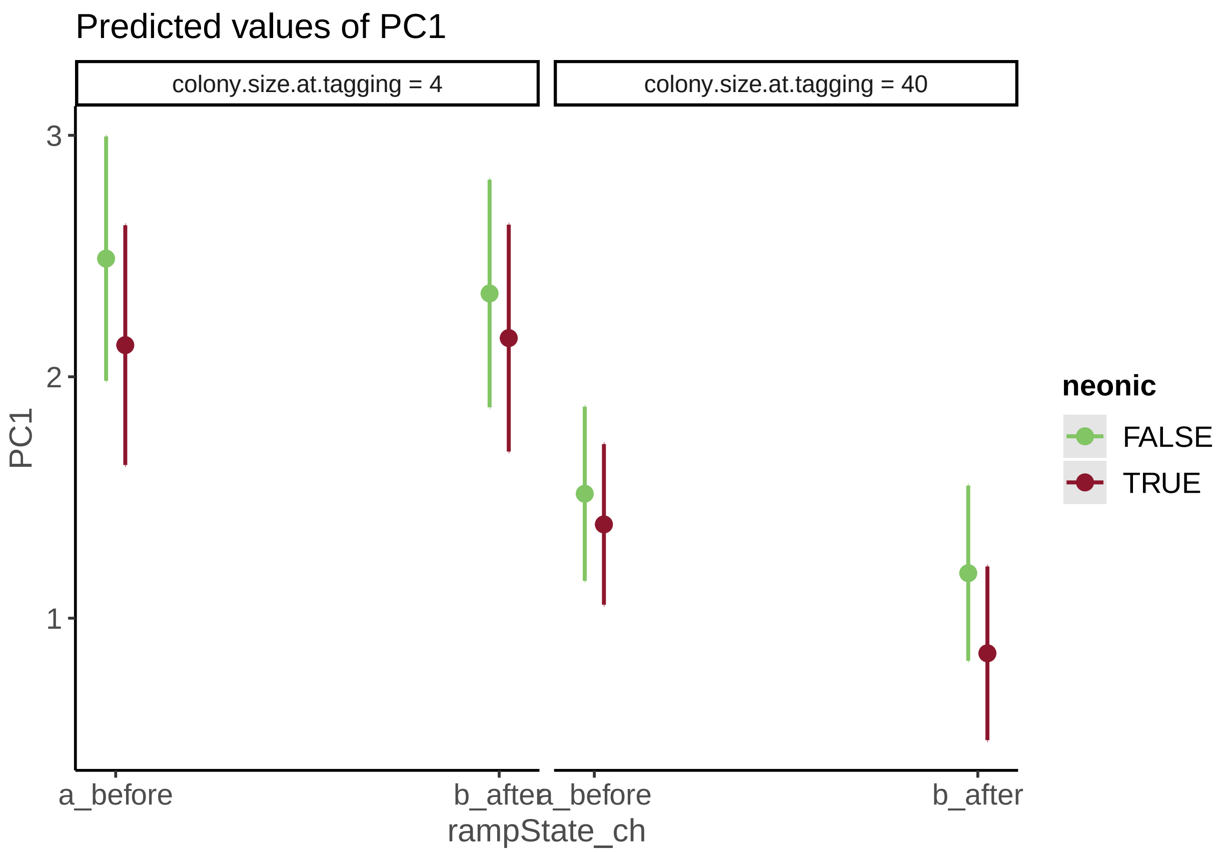
```
pc1.recovery.model.non.incubators <- lmer(PC1 ~ rampState_ch *
    neonic * log10(colony.size.at.tagging) + experimental.day +
    (1 | uniqueID) + (1 | colony.id), data = subset(sub.dat,
    caste == "worker" & uniqueID %in% non.incubator.list), REML = FALSE)
summary(pc1.recovery.model.non.incubators)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) +
##     experimental.day + (1 | uniqueID) + (1 | colony.id)
##    Data: subset(sub.dat, caste == "worker" & uniqueID %in% non.incubator.list)
##
##      AIC      BIC   logLik deviance df.resid
##   3457.5   3519.0  -1715.8   3431.5      823
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.73891 -0.71961 -0.05607  0.57924  3.14171
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  uniqueID  (Intercept) 1.1973   1.0942
##  colony.id (Intercept) 0.1511   0.3887
##  Residual              2.8797   1.6970
## Number of obs: 836, groups:  uniqueID, 144; colony.id, 15
##
## Fixed effects:
##                                                              Estimate
## (Intercept)                                                    0.3028
## rampState_chb_after                                            2.4213
## neonicTRUE                                                    -1.1893
## log10(colony.size.at.tagging)                                 -0.5204
## experimental.day2                                             -0.7614
## experimental.day3                                             -0.9498
## rampState_chb_after:neonicTRUE                                -6.4569
## rampState_chb_after:log10(colony.size.at.tagging)             -0.5644
## neonicTRUE:log10(colony.size.at.tagging)                       0.0589
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)   3.8719
##                                                              Std. Error
## (Intercept)                                                      1.4757
## rampState_chb_after                                              1.1863
## neonicTRUE                                                       1.8935
## log10(colony.size.at.tagging)                                    0.9297
## experimental.day2                                                0.3654
## experimental.day3                                                0.3990
## rampState_chb_after:neonicTRUE                                   1.4472
## rampState_chb_after:log10(colony.size.at.tagging)                0.7546
## neonicTRUE:log10(colony.size.at.tagging)                         1.1963
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging)     0.9026
##                                                                    df t value
## (Intercept)                                                   23.9353   0.205
## rampState_chb_after                                          747.4004   2.041
## neonicTRUE                                                    15.9790  -0.628
## log10(colony.size.at.tagging)                                 15.2236  -0.560
## experimental.day2                                            658.6795  -2.084
## experimental.day3                                            637.0708  -2.381
## rampState_chb_after:neonicTRUE                               745.7581  -4.462
## rampState_chb_after:log10(colony.size.at.tagging)            748.9345  -0.748
## neonicTRUE:log10(colony.size.at.tagging)                      11.3623   0.049
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging) 748.4433   4.290
##                                                               Pr(>|t|)
## (Intercept)                                                     0.8392
## rampState_chb_after                                             0.0416 *
## neonicTRUE                                                      0.5388
## log10(colony.size.at.tagging)                                   0.5838
## experimental.day2                                               0.0376 *
## experimental.day3                                               0.0176 *
## rampState_chb_after:neonicTRUE                                9.38e-06 ***
## rampState_chb_after:log10(colony.size.at.tagging)               0.4547
## neonicTRUE:log10(colony.size.at.tagging)                        0.9616
## rampState_chb_after:neonicTRUE:log10(colony.size.at.tagging) 2.02e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) rmpS__ nnTRUE l10(.. expr.2 expr.3 rmS__:TRUE rS__:1 nTRUE:
## rmpStt_chb_ -0.478
## neonicTRUE  -0.730  0.373
## lg10(cl...) -0.950  0.468  0.753
## exprmntl.d2 -0.213 -0.050 -0.067 -0.029
## exprmntl.d3 -0.193 -0.012 -0.119 -0.046  0.907
## rmpS__:TRUE  0.382 -0.815 -0.488 -0.389  0.088  0.088
## rS__:10(...  0.459 -0.983 -0.355 -0.462  0.044 -0.002  0.800
## nTRUE:10(..  0.724 -0.359 -0.979 -0.784  0.088  0.142  0.463      0.351
## rS__:TRUE:1 -0.367  0.820  0.466  0.391 -0.113 -0.097 -0.982     -0.832 -0.456
```

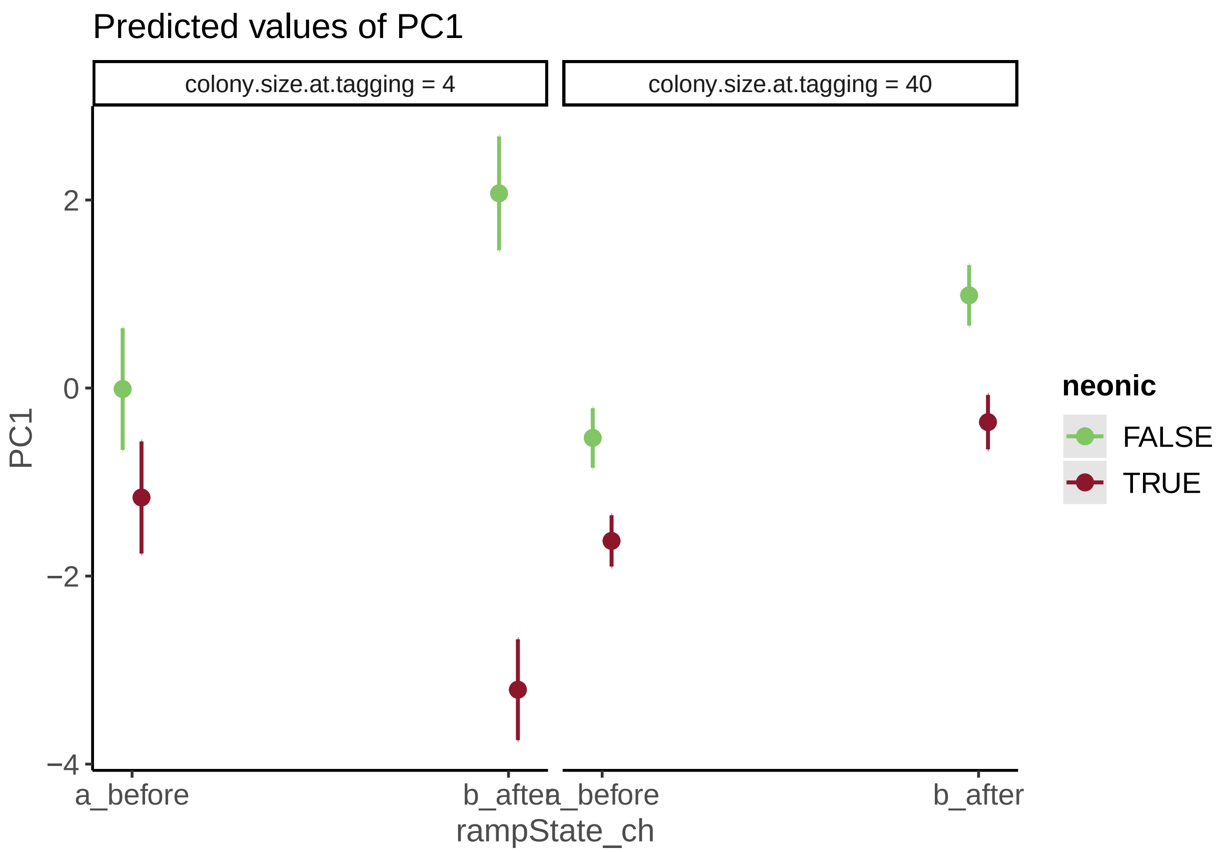
```
# Explicit model for incubator interaction effect, subset
# to small colonies
sub.dat$incubator <- NA
sub.dat$incubator[sub.dat$uniqueID %in% incubator.list] <- "inc"
sub.dat$incubator[sub.dat$uniqueID %in% non.incubator.list] <- "ninc"

pc1.recovery.model.by.incubator <- lmer(PC1 ~ rampState_ch *
    neonic * incubator + (1 | colony.id/uniqueID), data = subset(sub.dat,
    caste == "worker" & incubator %in% c("inc", "ninc") & colony.size.at.tagging <
        colony.size.cuttoff), REML = FALSE)
summary(pc1.recovery.model.by.incubator)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite’s
##   method [lmerModLmerTest]
## Formula: PC1 ~ rampState_ch * neonic * incubator + (1 | colony.id/uniqueID)
##    Data: subset(sub.dat, caste == "worker" & incubator %in% c("inc", "ninc") &
##     colony.size.at.tagging < colony.size.cuttoff)
##
##      AIC      BIC   logLik deviance df.resid
##   2965.8   3016.2  -1471.9   2943.8      714
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8747 -0.5014  0.0561  0.5411  2.9187
##
## Random effects:
##  Groups             Name        Variance Std.Dev.
##  uniqueID:colony.id (Intercept) 0.7474   0.8645
##  colony.id          (Intercept) 0.6445   0.8028
##  Residual                       2.8696   1.6940
## Number of obs: 725, groups:  uniqueID:colony.id, 101; colony.id, 13
##
## Fixed effects:
##                                              Estimate Std. Error       df
## (Intercept)                                    2.4337     0.4881  20.4189
## rampState_chb_after                           -0.4278     0.2958 645.9634
## neonicTRUE                                    -0.2821     0.6471  18.0846
## incubatorninc                                 -3.6775     0.5274 190.8373
## rampState_chb_after:neonicTRUE                 0.1318     0.3914 650.7818
## rampState_chb_after:incubatorninc              1.9429     0.4659 656.6091
## neonicTRUE:incubatorninc                      -0.8337     0.7123 176.6817
## rampState_chb_after:neonicTRUE:incubatorninc  -1.4016     0.5824 656.0326
##                                              t value Pr(>|t|)
## (Intercept)                                    4.986 6.69e-05 ***
## rampState_chb_after                           -1.446   0.1486
## neonicTRUE                                    -0.436   0.6680
## incubatorninc                                 -6.972 4.96e-11 ***
## rampState_chb_after:neonicTRUE                 0.337   0.7364
## rampState_chb_after:incubatorninc              4.170 3.45e-05 ***
## neonicTRUE:incubatorninc                      -1.170   0.2434
## rampState_chb_after:neonicTRUE:incubatorninc  -2.407   0.0164 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) rmpS__ nnTRUE incbtr rmS__:TRUE rmS__: nTRUE:
## rmpStt_chb_ -0.408
## neonicTRUE  -0.754  0.308
## incubatrnnc -0.489  0.375  0.369
## rmpS__:TRUE  0.309 -0.756 -0.400 -0.283
## rmpStt_ch_:  0.242 -0.633 -0.182 -0.562  0.478
## nncTRUE:ncb  0.362 -0.277 -0.504 -0.740  0.365      0.416
## rmS__:TRUE: -0.193  0.506  0.253  0.450 -0.671     -0.800 -0.524
```

```
# Plot models
plot_model(pc1.recovery.model.incubators, type = "pred", terms = c("rampState_ch",
    "neonic", "colony.size.at.tagging [4, 40]]"), ci.lvl = 0.5,
    colors = list(ctrl_color, imid_color))
```



```
plot_model(pc1.recovery.model.non.incubators, type = "pred",
    terms = c("rampState_ch", "neonic", "colony.size.at.tagging [4, 40]]"),
    ci.lvl = 0.5, colors = list(ctrl_color, imid_color))
```



```
setwd(output_folder)

pdf("Fig5C.pdf")
plot_model(pc1.recovery.model.by.incubator, type = "pred", terms = c("rampState_ch",
    "neonic", "incubator"), ci.lvl = 0.5, colors = list(ctrl_color,
    imid_color), axis.title = c("Observation Period", "Spatial Centrality (PC1)"),
    show.legend = F, title = "Recovery response by worker group")
dev.off()
```

```
## pdf
##   2
```

```
# Export panels
pdf("FigS3.pdf")
a <- plot_model(pc1.recovery.model.incubators, type = "pred",
    terms = c("rampState_ch", "neonic", "colony.size.at.tagging [4, 40]]"),
    ci.lvl = 0.5, colors = list(ctrl_color, imid_color), axis.title = c("Observation Period",
        "Spatial Centrality (PC1)"), show.legend = F, title = "On-nest workers")

b <- plot_model(pc1.recovery.model.non.incubators, type = "pred",
    terms = c("rampState_ch", "neonic", "colony.size.at.tagging [4, 40]]"),
    ci.lvl = 0.5, colors = list(ctrl_color, imid_color), axis.title = c("Observation Period",
        "Spatial Centrality (PC1)"), show.legend = F, title = "Off-nest workers")
grid.arrange(a, b, nrow = 1)
dev.off()
```

```
## pdf
##   2
```

Brood surface temperature analysis

```
thermData$neonic <- as.factor(thermData$neonic)  #Convert to factor

dat <- subset(thermData, tempRampState %in% c(1, 2, 3) & cold ==
    TRUE)
brood.temp.model <- lmer(meanTempBrood_all ~ nestAirTemp * neonic *
    log10(colony.size.at.tagging) + (1 | species/colony.id) +
    (1 | experimental.day), data = dat, REML = FALSE)

rem.ind <- which(abs(residuals(brood.temp.model)) > 4)  #Remove two outliers
dat.no <- dat[-rem.ind, ]
brood.temp.modelno.outliers <- lmer(meanTempBrood_all ~ nestAirTemp *
    neonic * log10(colony.size.at.tagging) + (1 | species/colony.id) +
    (1 | block), data = dat.no)
# No qualitative effect of removing outliers - retain
# original model


# Recovery model
dat <- subset(thermData, tempRampState %in% c(1, 5) & cold ==
    TRUE)
dat$tempRampState <- as.factor(dat$tempRampState)
brood.temp.recovery.model <- lmer(meanTempBrood_all ~ tempRampState *
    neonic * log10(colony.size.at.tagging) + (1 | colony.id),
    data = dat)


# Control model for comparison
dat <- subset(thermData, tempRampState %in% c(1, 5) & cold ==
    FALSE)
dat$tempRampState <- as.factor(dat$tempRampState)
brood.temp.recovery.model.ctl <- lmer(meanTempBrood_all ~ tempRampState +
    neonic + log10(colony.size.at.tagging) + (1 | colony.id),
    data = dat)


summary(brood.temp.model)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula:
## meanTempBrood_all ~ nestAirTemp * neonic * log10(colony.size.at.tagging) +
##     (1 | species/colony.id) + (1 | experimental.day)
##    Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##   1946.6   2000.0   -961.3   1922.6      624
##
```

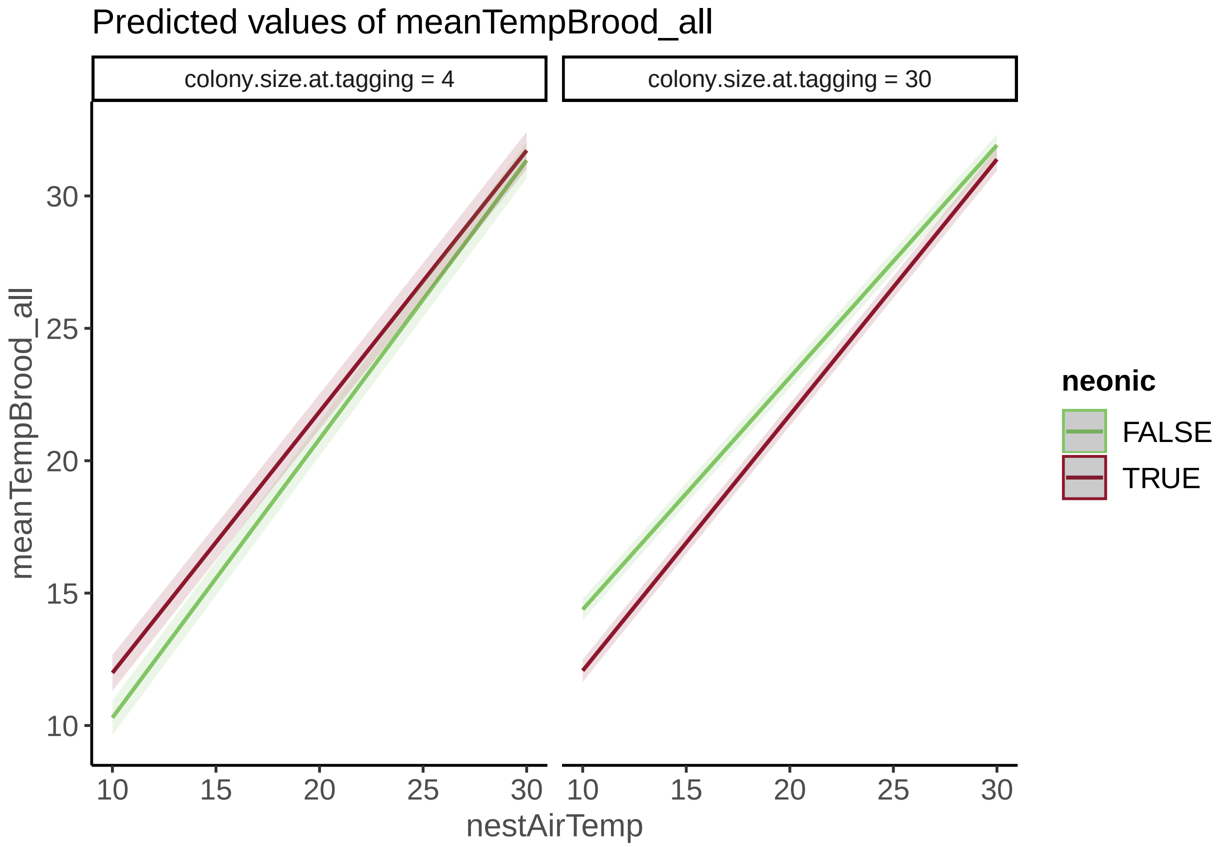
```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.2071 -0.5925 -0.0430  0.5243  4.4022
##
## Random effects:
##  Groups            Name        Variance Std.Dev.
##  colony.id:species (Intercept) 1.5439   1.2425
##  species           (Intercept) 0.3092   0.5561
##  experimental.day  (Intercept) 0.1293   0.3595
##  Residual                      1.0563   1.0278
## Number of obs: 636, groups:
## colony.id:species, 20; species, 4; experimental.day, 3
##
## Fixed effects:
##                                                       Estimate Std. Error
## (Intercept)                                           -4.24856    1.70365
## nestAirTemp                                            1.17337    0.04667
## neonicTRUE                                             6.16911    2.53017
## log10(colony.size.at.tagging)                          6.67527    1.17965
## nestAirTemp:neonicTRUE                                -0.17242    0.06885
## nestAirTemp:log10(colony.size.at.tagging)             -0.20074    0.03345
## neonicTRUE:log10(colony.size.at.tagging)              -6.34302    1.85403
## nestAirTemp:neonicTRUE:log10(colony.size.at.tagging)   0.17696    0.05189
##                                                             df t value Pr(>|t|)
## (Intercept)                                           35.62000  -2.494 0.017423
## nestAirTemp                                          615.47005  25.140  < 2e-16
## neonicTRUE                                            33.68491   2.438 0.020191
## log10(colony.size.at.tagging)                         30.73865   5.659 3.34e-06
## nestAirTemp:neonicTRUE                               615.90154  -2.504 0.012521
## nestAirTemp:log10(colony.size.at.tagging)            615.17944  -6.001 3.34e-09
## neonicTRUE:log10(colony.size.at.tagging)              35.82445  -3.421 0.001573
## nestAirTemp:neonicTRUE:log10(colony.size.at.tagging) 616.27945   3.410 0.000692
##
## (Intercept)                                          *
## nestAirTemp                                          ***
## neonicTRUE                                           *
## log10(colony.size.at.tagging)                        ***
## nestAirTemp:neonicTRUE                               *
## nestAirTemp:log10(colony.size.at.tagging)            ***
## neonicTRUE:log10(colony.size.at.tagging)             **
## nestAirTemp:neonicTRUE:log10(colony.size.at.tagging) ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) nstArT nnTRUE l10(.. nsAT:TRUE nAT:10 nTRUE:
## nestAirTemp -0.480
## neonicTRUE  -0.673  0.323
## lg10(cl...) -0.931  0.474  0.630
## nstArT:TRUE  0.327 -0.676 -0.483 -0.321
## nAT:10(...)  0.458 -0.954 -0.308 -0.497  0.645
## nTRUE:10(..  0.636 -0.301 -0.963 -0.649  0.479     0.315
## nAT:TRUE:10 -0.297  0.612  0.465  0.320 -0.960    -0.642 -0.500
```

```
summary(brood.temp.recovery.model)
```

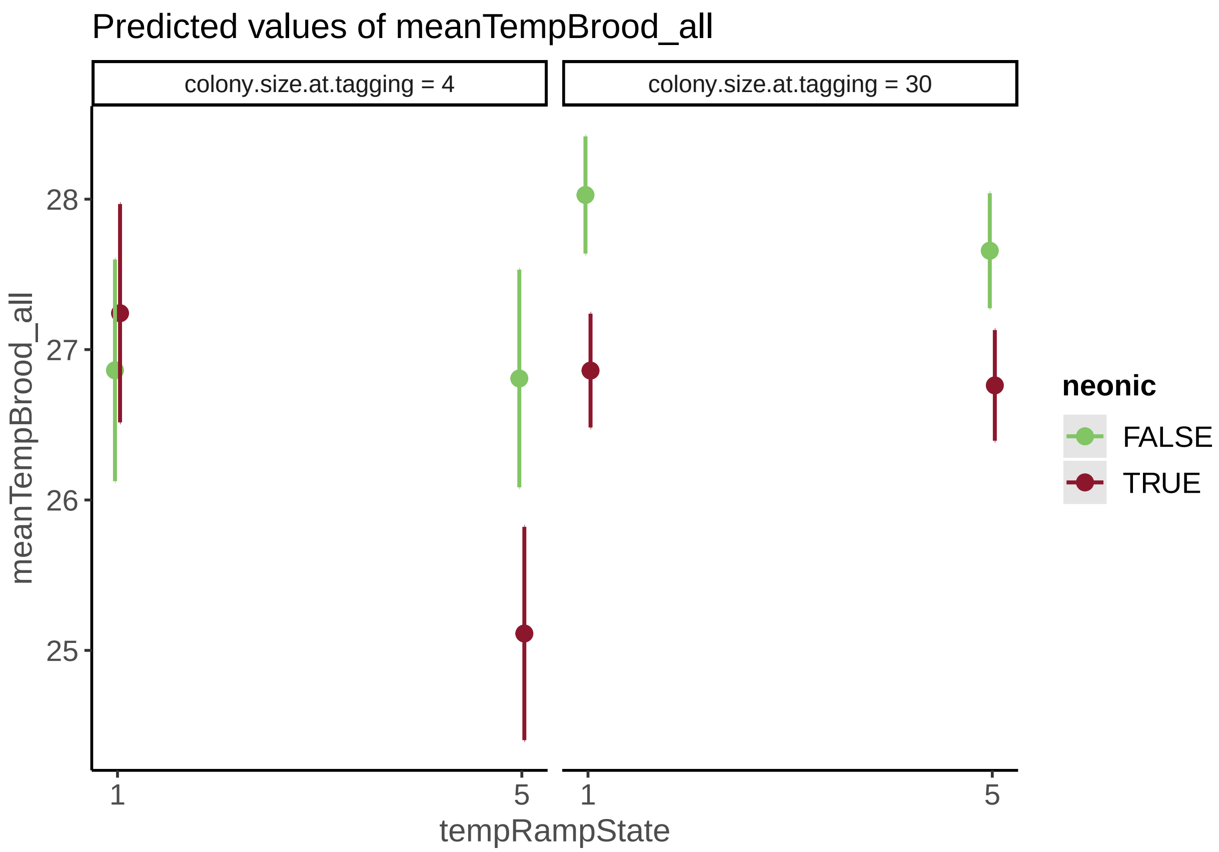
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## meanTempBrood_all ~ tempRampState * neonic * log10(colony.size.at.tagging) +
##     (1 | colony.id)
##    Data: dat
##
## REML criterion at convergence: 614.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0748 -0.5206  0.0313  0.4608  3.3561
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  colony.id (Intercept) 2.605    1.6140
##  Residual              0.701    0.8373
## Number of obs: 222, groups:  colony.id, 21
##
## Fixed effects:
##                                                      Estimate Std. Error
## (Intercept)                                           26.0603     1.8385
## tempRampState5                                         0.1631     0.6470
## neonicTRUE                                             1.4439     2.5934
## log10(colony.size.at.tagging)                          1.3320     1.3388
## tempRampState5:neonicTRUE                             -3.6901     0.9158
## tempRampState5:log10(colony.size.at.tagging)          -0.3614     0.4792
## neonicTRUE:log10(colony.size.at.tagging)              -1.7679     1.8971
## tempRampState5:neonicTRUE:log10(colony.size.at.tagging)   2.6825     0.6908
##                                                            df t value
## (Intercept)                                           18.0818  14.175
## tempRampState5                                       198.2826   0.252
## neonicTRUE                                            18.4717   0.557
## log10(colony.size.at.tagging)                         18.2007   0.995
## tempRampState5:neonicTRUE                            198.2539  -4.029
## tempRampState5:log10(colony.size.at.tagging)         198.6021  -0.754
## neonicTRUE:log10(colony.size.at.tagging)              18.8889  -0.932
## tempRampState5:neonicTRUE:log10(colony.size.at.tagging) 198.4540   3.883
##                                                      Pr(>|t|)
## (Intercept)                                          3.10e-11 ***
## tempRampState5                                        0.80120
## neonicTRUE                                            0.58438
## log10(colony.size.at.tagging)                         0.33280
## tempRampState5:neonicTRUE                            7.97e-05 ***
## tempRampState5:log10(colony.size.at.tagging)          0.45165
## neonicTRUE:log10(colony.size.at.tagging)              0.36317
## tempRampState5:neonicTRUE:log10(colony.size.at.tagging)  0.00014 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##              (Intr) tmpRS5 nnTRUE l10(.. tmRS5:TRUE tRS5:1 nTRUE:
## tempRmpStt5 -0.223
## neonicTRUE  -0.709  0.158
## lg10(cl...) -0.957  0.213  0.678
## tmpRS5:TRUE  0.157 -0.707 -0.234 -0.151
## tRS5:10(...  0.209 -0.956 -0.148 -0.220  0.676
## nTRUE:10(..  0.675 -0.150 -0.958 -0.706  0.230      0.155
## tRS5:TRUE:1 -0.145  0.663  0.223  0.153 -0.960     -0.694 -0.239
```

```
# summary(brood.temp.recovery.model.ctl)

plot_model(brood.temp.model, type = "pred", terms = c("nestAirTemp",
    "neonic", "colony.size.at.tagging [4, 30]"), ci.lvl = 0.5,
    colors = list(ctrl_color, imid_color))
```



```
plot_model(brood.temp.recovery.model, type = "pred", terms = c("tempRampState",
    "neonic", "colony.size.at.tagging [4, 30]"), ci.lvl = 0.5,
    colors = list(ctrl_color, imid_color))
```



```
# Export panels
setwd(output_folder)
pdf("FigS5.pdf")
plot_model(brood.temp.model, type = "pred", terms = c("nestAirTemp",
    "neonic", "colony.size.at.tagging [4, 40]"), colors = list(ctrl_color,
    imid_color), axis.title = c("Nest Temperature (C0", "Spatial Centrality (PC1)"),
    show.legend = F, title = "Brood temperature during cold exposure")
dev.off()
```

```
## pdf
##   2
```

```
pdf("FigS6.pdf")
plot_model(brood.temp.recovery.model, type = "pred", terms = c("tempRampState",
    "neonic", "colony.size.at.tagging [4, 40]"), colors = list(ctrl_color,
    imid_color), axis.title = c("Observation Period", "Spatial Centrality (PC1)"),
    show.legend = F, title = "Brood temperature during recovery")
dev.off()
```

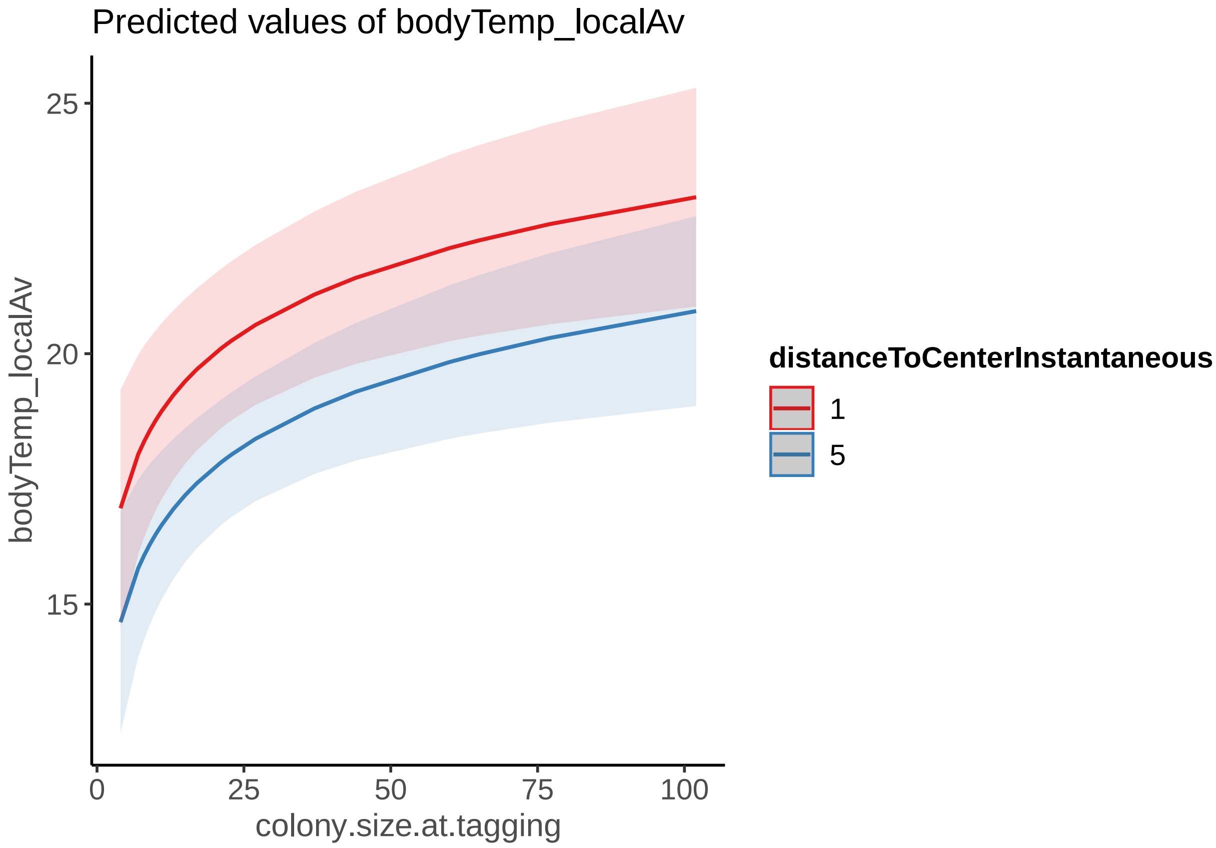
```
## pdf
##   2
```

Proximate behavioral mechanisms

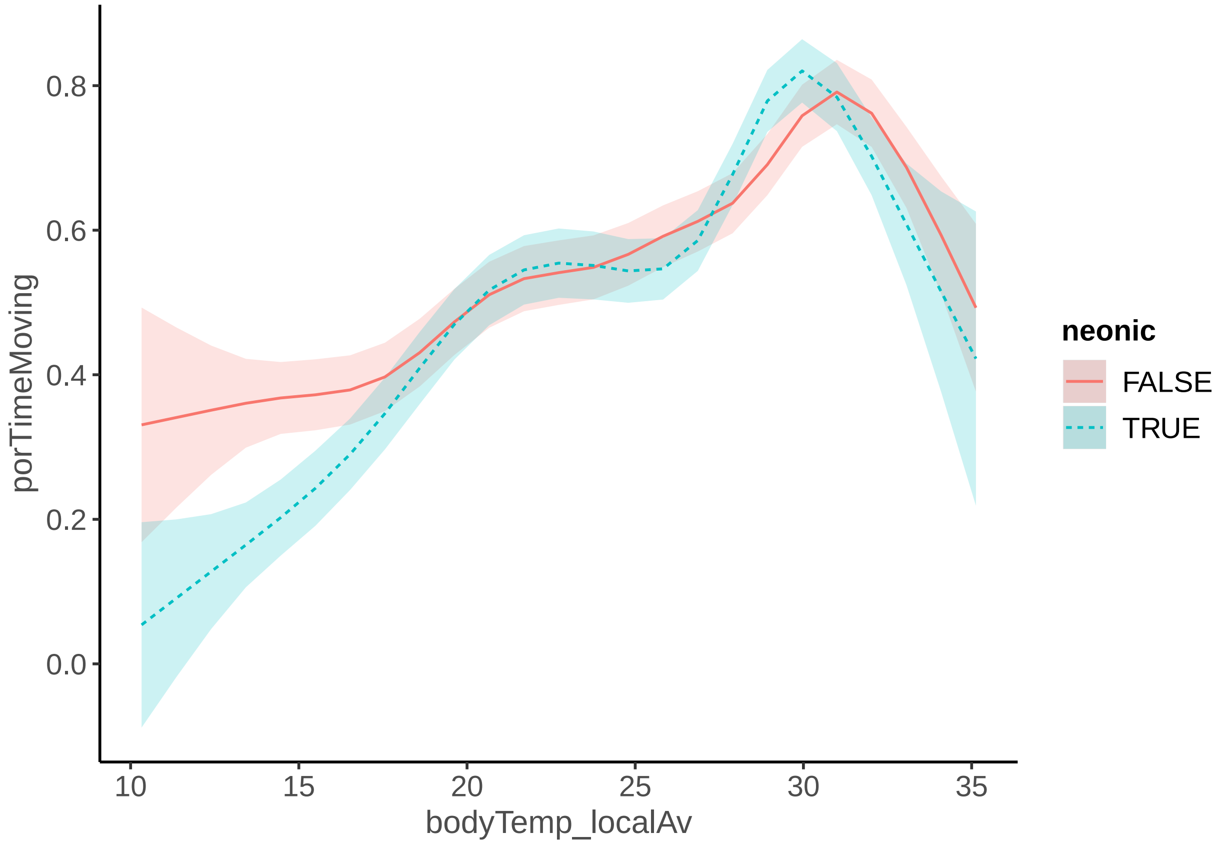
```
# temp vs distance vs colony size Create subset of data
# during cold exposure
dist.sub <- subset(behData, cold == TRUE & rampState == 3 & bodyTemp_localAv >
    10)
dist.sub$distanceToCenterInstantaneous <- dist.sub$distanceToCenterInstantaneous/pixel_conversion
dist.sub$medianDistancetoClosestBrood <- dist.sub$medianDistancetoClosestBrood/pixel_conversion

temp.model <- lmer(bodyTemp_localAv ~ neonic + distanceToCenterInstantaneous +
    log10(colony.size.at.tagging) + (distanceToCenterInstantaneous |
    colony.id) + (1 | colony.id/uniqueID), data = dist.sub)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues

## Warning: Model failed to converge with 1 negative eigenvalue: -1.0e-01
```

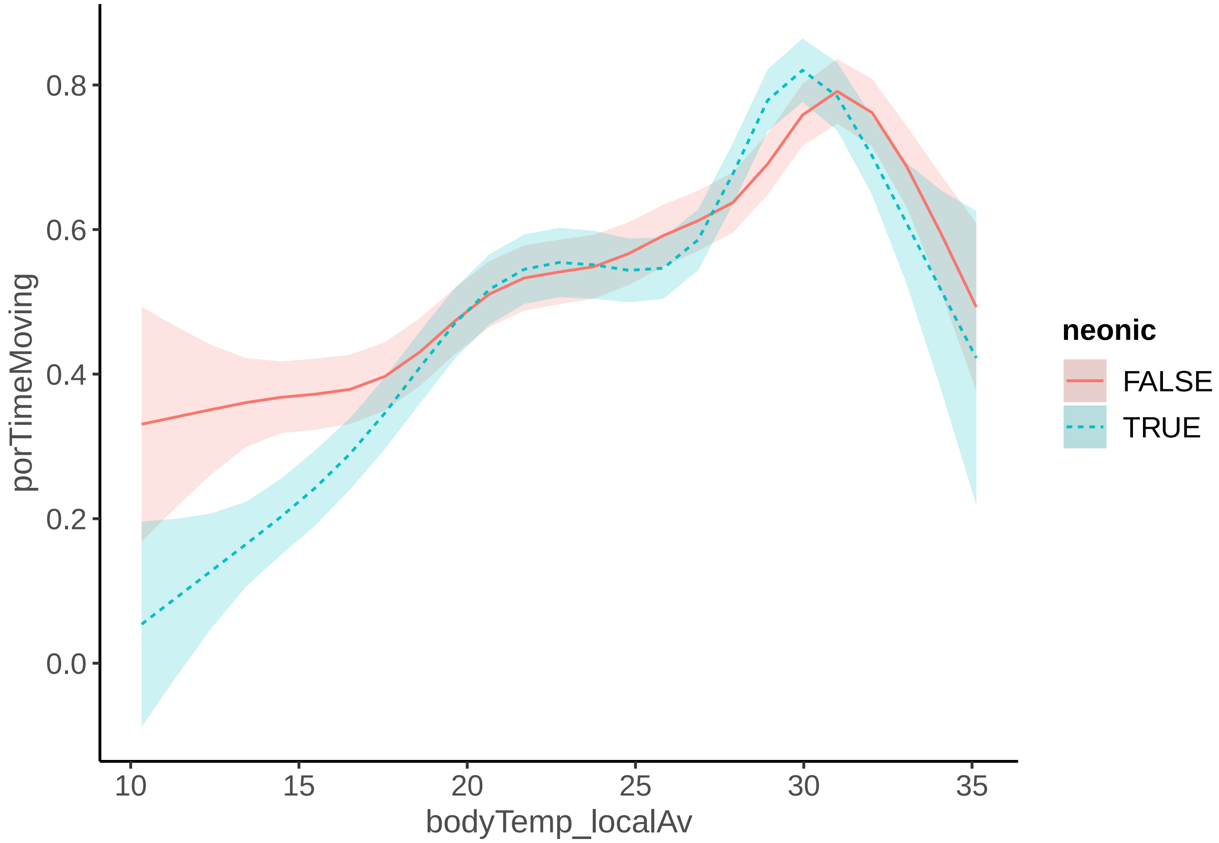
```
plot_model(temp.model, type = "pred", terms = c("colony.size.at.tagging",
    "distanceToCenterInstantaneous[1,5]"))
```



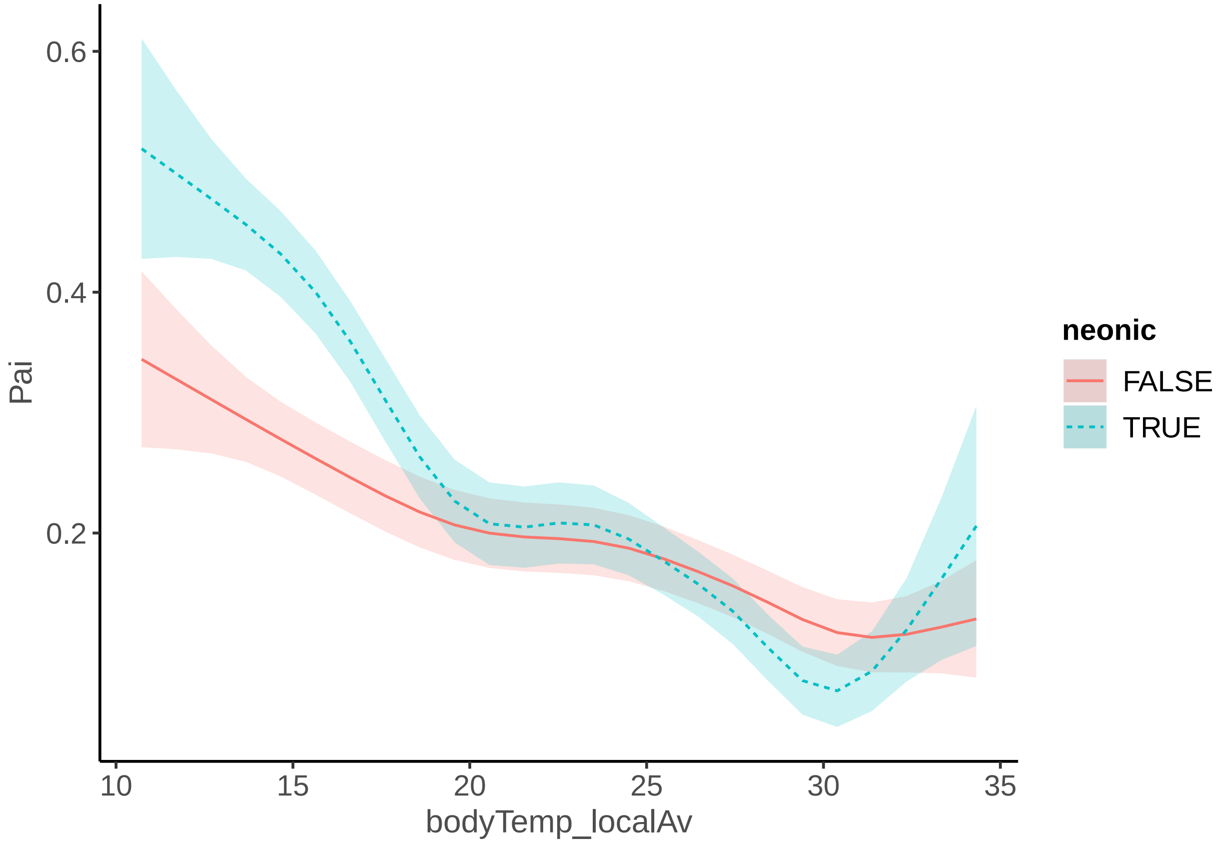
```
# Overall activity by
activity.body.temp.model <- bam(porTimeMoving ~ s(colony.id,
    bs = "re") + s(uniqueID, bs = "re") + s(bodyTemp_localAv,
    by = neonic), data = subset(behData, bodyTemp_localAv > 10 &
    rampState > 0))
plot_smooths(activity.body.temp.model, bodyTemp_localAv, neonic)
```



```
pc1.body.temp.model <- bam(PC1 ~ s(colony.id, bs = "re") + s(uniqueID,
    bs = "re") + s(bodyTemp_localAv, by = neonic), data = subset(behData,
    bodyTemp_localAv > 10 & rampState > 0))
plot_smooths(activity.body.temp.model, bodyTemp_localAv, neonic)
```



```r
# Activity state switching vs body temperature
pai.body.temp.model <- bam(Pai ~ s(colony.id, bs = "re") + s(uniqueID,
    bs = "re") + s(bodyTemp_localAv, by = neonic), data = subset(behData,
    bodyTemp_localAv > 10 & rampState %in% c(1, 2, 3)))
plot_smooths(pai.body.temp.model, bodyTemp_localAv, neonic)
```



```
### Figure 3
pdf("Fig4_a.pdf")
# plot_smooths(body.temp.by.dist.model,
# distanceToCenterInstantaneous, small.colony)
plot_model(temp.model, type = "pred", terms = c("colony.size.at.tagging",
    "distanceToCenterInstantaneous[1,5]"))
dev.off()
```

```
## pdf
##   2
```

```
pdf("Fig4_b.pdf")
plot_smooths(pai.body.temp.model, bodyTemp_localAv, neonic)
dev.off()
```

```
## pdf
##   2
```

```
pdf("Fig4_c.pdf")
plot_smooths(activity.body.temp.model, bodyTemp_localAv, neonic)
dev.off()
```

```
## pdf
##   2
```

```
# Colony size vs. activity
colony.sizes <- aggregate(behData$colony.size.at.tagging, list(behData$colony.id),
    FUN = mean)[, 2]
mean.activity <- aggregate(behData$porTimeMoving, list(behData$colony.id),
    FUN = mean)[, 2]
activity.size.model <- gam(mean.activity ~ s(colony.sizes))
summary(activity.size.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean.activity ~ s(colony.sizes)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.70759    0.01701   41.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(colony.sizes)   1      1 4.744  0.0354 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0837   Deviance explained = 10.6%
## GCV = 0.012762  Scale est. = 0.012154  n = 42
```

```
# Colony size vs. interaction rate
colony.sizes <- aggregate(behData$colony.size.at.tagging, list(behData$colony.id),
    FUN = mean)[, 2]
interaction.rate <- aggregate(behData$meanInteractionRate * behData$colony.size.at.tagging,
    list(behData$colony.id), FUN = mean)[, 2]
interaction.size.model <- gam(interaction.rate ~ s(colony.sizes))
summary(interaction.size.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## interaction.rate ~ s(colony.sizes)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.46364    0.02902   15.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(colony.sizes)   1      1 79.16  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.656   Deviance explained = 66.4%
## GCV = 0.037141  Scale est. = 0.035373  n = 42
```

```
setwd(output_folder)
png("FigS8.png", width = 640, height = 300)
a <- plot_smooths(activity.size.model, colony.sizes) + xlab("Colony Size") +
    ylab("Mean Activity")
b <- plot_smooths(interaction.size.model, colony.sizes) + xlab("Colony Size") +
    ylab("Interaction Rate (interactions/frame)")
grid.arrange(a, b, nrow = 1)
dev.off()
```

```
## pdf
##   2
```

Microcolony size-manipulation experiments

Load and clean data

```
block_1 <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
block_2 <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
block_3 <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
metadata <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
mc_data <- rbind(block_1, block_2, block_3)
head(mc_data)
```

```
##   X mean_min_distance_to_brood num_brood_interactions mean_dist_other_bees
## 1 0                   1.617829                      0             2.430178
## 2 1                   6.518312                      0             8.073369
## 3 2                   1.622784                      0             3.783518
## 4 3                   0.817224                     73             4.001826
## 5 4                   1.681611                      0             4.743056
## 6 5                   6.861974                      0             6.188042
##   total_social_interactions countable_interaction_frames activity
## 1                        10                           22        0
## 2                         0                           18        0
## 3                        67                          153        0
## 4                        67                          170        0
## 5                         0                           44        0
## 6                         0                           22        0
##   speed_when_moving distance_to_social_center frames_tracked tag_ids
## 1                NA                 0.5888912              5      16
## 2                NA                 7.6145221              6      17
## 3                NA                 0.5811764             62      22
## 4                NA                 0.4717314             73      33
## 5                NA                 3.3990098             14      36
## 6                NA                 5.5870597              5      37
##                            video_file block colony           timestamp
## 1 microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 2 microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 3 microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 4 microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 5 microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 6 microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
##   time_num nest_temp
## 1 272.6542    16.583
## 2 272.6542    16.583
## 3 272.6542    16.583
## 4 272.6542    16.583
## 5 272.6542    16.583
## 6 272.6542    16.583
```

```r
### Add metadata to data
for (i in 1:length(metadata[, 1])) {
    idx = mc_data$block == metadata$block[i] & mc_data$colony ==
        metadata$colony_number[i]
    mc_data$treatment[idx] <- metadata$treatment[i]
    mc_data$colony_size[idx] <- metadata$colony_size[i]
    mc_data$source_colony[idx] <- metadata$source_colony[i]


}

# Convert to factors
mc_data$block <- as.factor(mc_data$block)
mc_data$colony_size <- as.factor((mc_data$colony_size))
mc_data$source_colony <- as.factor(mc_data$source_colony)
mc_data$unique_colony <- paste(mc_data$block, mc_data$colony,
    sep = "_")
mc_data$tag_ids <- as.factor(mc_data$tag_ids)
mc_data$unique.id <- paste(mc_data$tag_ids, mc_data$unique_colony,
    sep = "_")
mc_data$unique.id <- as.factor(mc_data$unique.id)
### Clean up mc_data Clear out faulty temperature readings
mc_data$nest_temp[mc_data$nest_temp < 5 | mc_data$nest_temp >
    35] <- NaN
# Log transform speed
mc_data$speed_when_moving <- log10(mc_data$speed_when_moving)

# Square root transform brood interactions
mc_data$num_brood_interactions <- sqrt(mc_data$num_brood_interactions)
# hist(mc_data$mean_dist_other_bees)
# hist(mc_data$mean_min_distance_to_brood, breaks = 100)
# hist(mc_data$distance_to_social_center)
# hist(mc_data$num_brood_interactions^(1/2))
# hist(mc_data$activity)
# hist(log10(1+mc_data$total_social_interactions))
# hist(log10(mc_data$speed_when_moving))
# hist(mc_data$frames_tracked)
# Get tag lists from experiments
mc_taglist <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStresso
# Convert to factors
attach(mc_taglist)
valid.ids <- paste(tag.ID, block, colony, sep = "_")
detach(mc_taglist)
mc_data <- subset(mc_data, unique.id %in% valid.ids & frames_tracked >
    20)
dim(mc_data)  #Size of data -> # of unique observations
```

```
## [1] 106164     22
```

```
length(unique(mc_data$unique.id))  #unique bees
```

```
## [1] 237
```

```
# Remove two sets of erroneous data
rm.ind.1 <- mc_data$unique_colony == "1_12" & mc_data$time_num <
    270  #Some erroneous timestamps
rm.ind.2 <- mc_data$unique_colony == "2_3" & mc_data$time_num >
    280  #Period  of artifically high readings, removing
rm.ind <- rm.ind.1 | rm.ind.2
mc_data <- mc_data[!rm.ind, ]

# Remove some abnormally high temp readings (a few hundred)
# mc_data <- subset(mc_data, nest_temp < 28) Separate pre
# and post
cols <- unique(mc_data$unique_colony)
rampThresh <- 20

define_ramp_states <- function(sd, plt = FALSE) {
    minT <- min(sd$nest_temp, na.rm = TRUE)
    minT_time <- sd$time_num[sd$nest_temp == minT][1]
    sd$rampState[sd$nest_temp > 20 & sd$time_num < minT_time] <- "pre"
    sd$rampState[sd$nest_temp > 20 & sd$time_num > minT_time] <- "post"
    sd$rampState[sd$nest_temp <= 20] <- "ramp"

    sd$cl[sd$ramp == "pre"] <- "coral1"
    sd$cl[sd$ramp == "ramp"] <- "cadetblue"
    sd$cl[sd$ramp == "post"] <- "coral4"
    plt_data <- subset(sd, rampState %in% c("pre", "ramp", "post"))
    if (plt == TRUE) {
        plot(nest_temp ~ time_num, data = plt_data, col = plt_data$cl,
            pch = 19, cex = 0.5)
        title(unique(sd$unique_colony))
    }
    return(sd$rampState)

}


for (i in 1:length(cols)) {
    cur.ind <- mc_data$unique_colony == cols[i]
    sd <- mc_data[cur.ind, ]
    rampState <- define_ramp_states(sd)
    mc_data$rampState[cur.ind] <- rampState
}

# mc_data <- subset(mc_data, rampState %in% c('pre',
# 'ramp'))
```

PCA analysis

```
spat.vars <- c("mean_dist_other_bees", "mean_min_distance_to_brood",
    "distance_to_social_center", "total_social_interactions",
    "num_brood_interactions")

pc.dat <- mc_data[, spat.vars]
cmplt_ind <- complete.cases(pc.dat)
pc.dat <- pc.dat[cmplt_ind, ]
mc_data <- mc_data[cmplt_ind, ]
mc.nest.beh.pca <- prcomp(pc.dat, scale = TRUE, center = TRUE)
mc_data$pc1 <- -mc.nest.beh.pca$x[, 1]  #Inverse sign for consistency with other analysis, so higher pc

head(mc_data)
```

```
##     X mean_min_distance_to_brood num_brood_interactions mean_dist_other_bees
## 3   2                  1.6227837               0.000000             3.783518
## 4   3                  0.8172240               8.544004             4.001826
## 8   7                  2.7178458               0.000000             5.169691
## 11 10                  0.9112662               8.774964             4.162098
## 12 11                  1.1629358               8.831761             4.142163
## 14 13                  3.9095601               0.000000             5.309894
##    total_social_interactions countable_interaction_frames  activity
## 3                         67                          153 0.0000000
## 4                         67                          170 0.0000000
## 8                          0                          170 0.0000000
## 11                        17                          108 0.8409091
## 12                        20                          140 0.7391304
## 14                         0                          140 0.0000000
##    speed_when_moving distance_to_social_center frames_tracked tag_ids
## 3                 NA                 0.5811764             62      22
## 4                 NA                 0.4717314             73      33
## 8                 NA                 1.9373203             73      47
## 11        -0.2440059                 2.2086506             47      22
## 12        -0.6191297                 0.7177818             72      33
## 14                NA                 2.1745005             74      47
##                              video_file block colony           timestamp
## 3  microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 4  microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 8  microcol_1-2021-09-30_15-42-01.mjpeg     1      1 2021-09-30_15-42-01
## 11 microcol_1-2021-09-30_12-42-01.mjpeg     1      1 2021-09-30_12-42-01
## 12 microcol_1-2021-09-30_12-42-01.mjpeg     1      1 2021-09-30_12-42-01
## 14 microcol_1-2021-09-30_12-42-01.mjpeg     1      1 2021-09-30_12-42-01
##    time_num nest_temp treatment colony_size source_colony unique_colony
## 3  272.6542    16.583    neonic           4             3           1_1
## 4  272.6542    16.583    neonic           4             3           1_1
## 8  272.6542    16.583    neonic           4             3           1_1
## 11 272.5292    22.529    neonic           4             3           1_1
## 12 272.5292    22.529    neonic           4             3           1_1
## 14 272.5292    22.529    neonic           4             3           1_1
##    unique.id rampState       pc1
## 3     22_1_1      ramp 2.0437801
## 4     33_1_1      ramp 2.8073377
## 8     47_1_1      ramp 0.7142319
## 11    22_1_1       pre 1.9910808
## 12    33_1_1       pre 2.2503496
## 14    47_1_1       pre 0.4409615
```

```
# plot(pc1~nest_temp, data = mc_data, pch = 19, cex = 0.1,
# col=rgb(0.6,0.2, 0.3, 0.1))
```

##Models Statistical models

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: pc1 ~ nest_temp + (1 | block/colony/tag_ids) + (1 | source_colony)
##    Data: subset(mc_data, treatment == "control")
##
##      AIC      BIC   logLik deviance df.resid
## 164423.5 164485.1 -82204.8 164409.5    48829
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3389 -0.6082  0.0115  0.6053  4.3088
##
## Random effects:
##  Groups                Name        Variance Std.Dev.
##  tag_ids:(colony:block) (Intercept) 1.0101   1.0050
##  colony:block           (Intercept) 0.1925   0.4388
##  block                  (Intercept) 3.7515   1.9369
##  source_colony          (Intercept) 1.2065   1.0984
##  Residual                           1.6743   1.2939
## Number of obs: 48836, groups:
## tag_ids:(colony:block), 115; colony:block, 15; block, 3; source_colony, 3
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)  1.826e+00  1.308e+00  3.579e+00   1.395    0.243
## nest_temp   -3.998e-02  1.350e-03  4.874e+04 -29.608   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## nest_temp -0.020
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite’s
##   method [lmerModLmerTest]
## Formula: pc1 ~ nest_temp * treatment * colony_size + (1 | block) + (1 |
##     colony/unique.id) + (1 | source_colony) + (1 | rampState)
##    Data: subset(mc_data, rampState %in% c("pre", "ramp"))
##
##      AIC      BIC   logLik deviance df.resid
## 127522.8 127643.6 -63747.4 127494.8    41302
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.1534 -0.5489 -0.0097  0.5379  4.9599
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  unique.id:colony (Intercept) 1.19429  1.0928
##  colony           (Intercept) 0.16517  0.4064
##  source_colony    (Intercept) 0.57625  0.7591
##  block            (Intercept) 1.77078  1.3307
##  rampState        (Intercept) 0.01119  0.1058
##  Residual                     1.24396  1.1153
## Number of obs: 41316, groups:
## unique.id:colony, 235; colony, 13; source_colony, 3; block, 3; rampState, 2
##
## Fixed effects:
##                                          Estimate Std. Error         df
## (Intercept)                             2.143e+00  9.285e-01  4.133e+00
## nest_temp                              -2.587e-02  3.509e-03  4.481e+03
## treatmentneonic                        -1.380e+00  2.777e-01  2.335e+02
## colony_size16                          -1.065e+00  3.077e-01  1.154e+02
## nest_temp:treatmentneonic               1.209e-02  3.624e-03  4.112e+04
## nest_temp:colony_size16                 5.604e-03  3.262e-03  4.111e+04
## treatmentneonic:colony_size16           1.348e+00  3.825e-01  1.403e+02
## nest_temp:treatmentneonic:colony_size16 -1.905e-02  4.470e-03  4.112e+04
##                                        t value Pr(>|t|)
## (Intercept)                              2.308 0.080087 .
## nest_temp                               -7.372 1.99e-13 ***
## treatmentneonic                         -4.969 1.30e-06 ***
## colony_size16                           -3.462 0.000752 ***
## nest_temp:treatmentneonic                3.335 0.000854 ***
## nest_temp:colony_size16                  1.718 0.085771 .
## treatmentneonic:colony_size16            3.523 0.000576 ***
## nest_temp:treatmentneonic:colony_size16  -4.262 2.03e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##             (Intr) nst_tm trtmnt cln_16 nst_t: n_:_16 tr:_16
## nest_temp   -0.065
## treatmntnnc -0.136  0.128
## colony_sz16 -0.150  0.118  0.533
## nst_tmp:trt  0.036 -0.567 -0.221 -0.109
## nst_tmp:_16  0.040 -0.636 -0.134 -0.180  0.600
## trtmntn:_16  0.108 -0.094 -0.722 -0.742  0.161  0.144
## nst_tm::_16 -0.029  0.463  0.179  0.131 -0.811 -0.730 -0.198
```

```
## boundary (singular) fit: see help('isSingular')
```

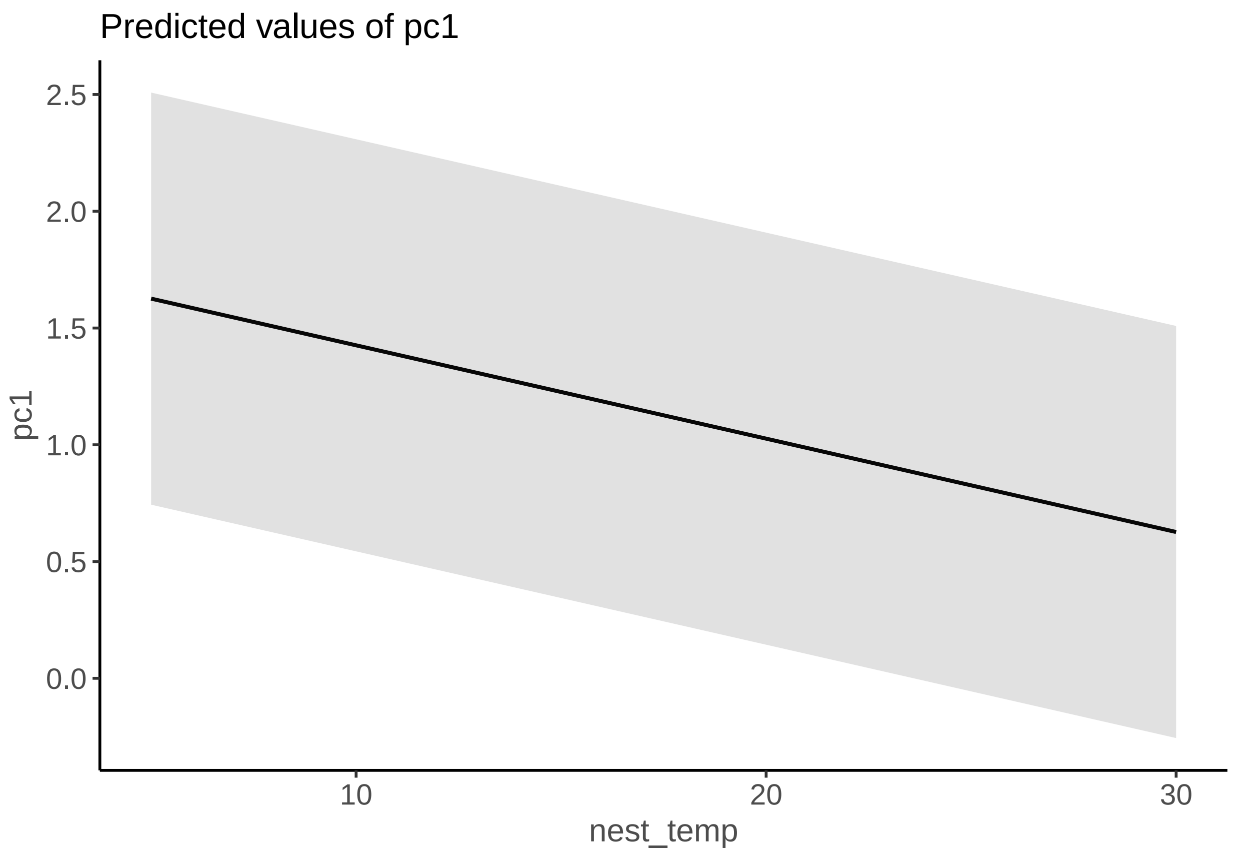
```
## Warning: Model failed to converge with 1 negative eigenvalue: -3.4e+00
```

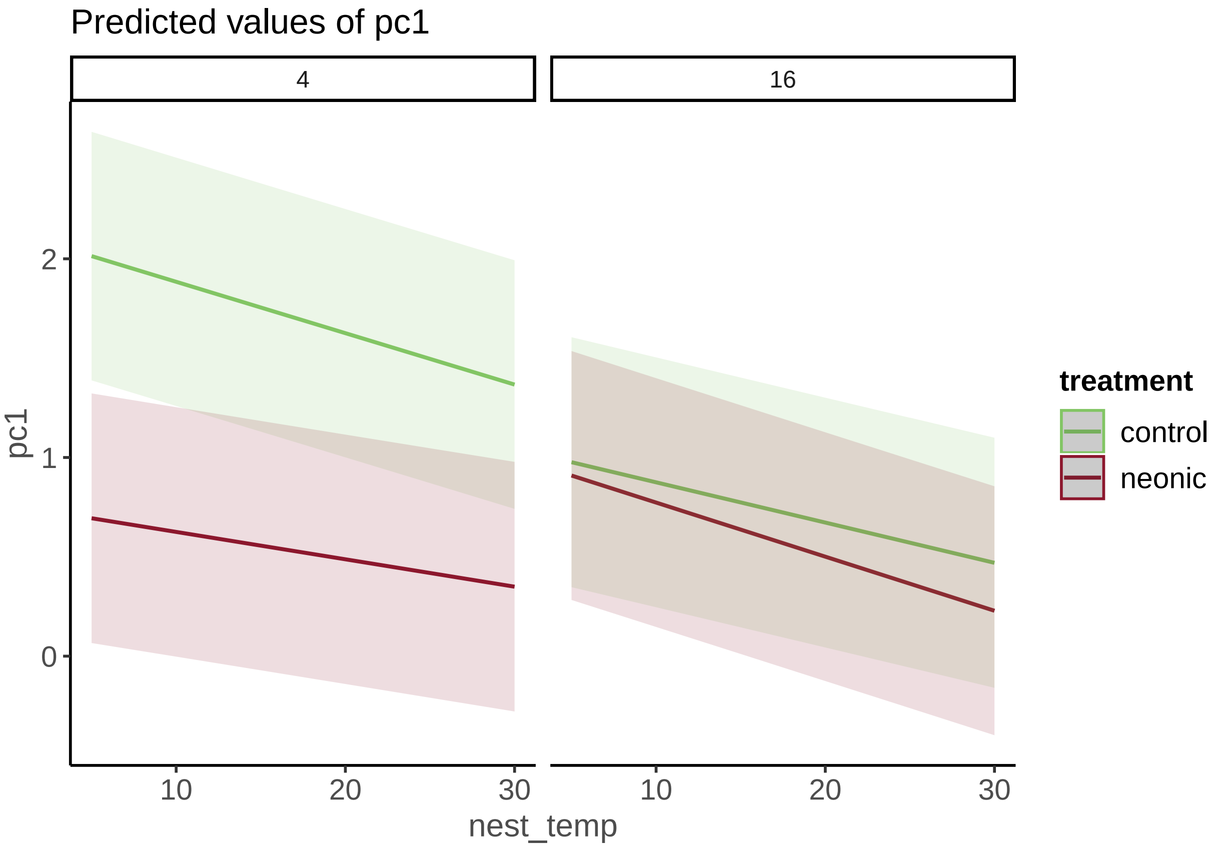
```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: pc1 ~ treatment * nest_temp + (1 | block) + (1 | colony/tag_ids) +
##     (1 | source_colony)
##    Data: subset(mc_data, colony_size == "16" & rampState %in% c("pre",
##     "ramp"))
##
##      AIC      BIC   logLik deviance df.resid
##  88857.3  88931.2 -44419.7  88839.3    26961
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8269 -0.5958 -0.0280  0.5949  4.4549
##
## Random effects:
##  Groups         Name        Variance  Std.Dev.
##  tag_ids:colony (Intercept) 1.231e+00 1.1094960
##  colony         (Intercept) 2.128e+00 1.4587787
##  block          (Intercept) 4.523e-01 0.6725650
##  source_colony  (Intercept) 1.293e-08 0.0001137
##  Residual                   1.536e+00 1.2394476
## Number of obs: 26970, groups:
## tag_ids:colony, 139; colony, 7; block, 3; source_colony, 2
##
## Fixed effects:
##                            Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)                1.570e+00  6.835e-01  8.767e+00   2.298   0.0479 *
## treatmentneonic           -1.761e+00  9.689e-02  2.359e+04 -18.174   <2e-16 ***
## nest_temp                 -3.349e-02  2.088e-03  2.685e+04 -16.034   <2e-16 ***
## treatmentneonic:nest_temp -5.245e-03  2.904e-03  2.686e+04  -1.806   0.0709 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) trtmnt nst_tm
## treatmntnnc -0.067
## nest_temp   -0.052  0.378
## trtmntnnc:_  0.038 -0.531 -0.719
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: pc1 ~ treatment * nest_temp + (1 | block) + (1 | colony/tag_ids) +
##     (1 | source_colony)
##    Data: subset(mc_data, colony_size == "4" & rampState %in% c("pre",
##     "ramp"))
```

```
##
##      AIC      BIC   logLik deviance df.resid
##  39842.9  39911.1 -19912.5  39824.9    14337
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.5019 -0.5376  0.0238  0.5764  4.6610
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  tag_ids:colony (Intercept) 0.5426   0.7366
##  colony         (Intercept) 0.4876   0.6983
##  source_colony  (Intercept) 2.2853   1.5117
##  block          (Intercept) 5.5371   2.3531
##  Residual                   0.9133   0.9557
## Number of obs: 14346, groups:
## tag_ids:colony, 79; colony, 13; source_colony, 3; block, 3
##
## Fixed effects:
##                            Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)               2.418e+00  1.640e+00  4.664e+00   1.474    0.205
## treatmentneonic          -1.244e+00  1.886e-01  9.467e+01  -6.596 2.40e-09 ***
## nest_temp                -3.855e-02  2.281e-03  1.428e+04 -16.906  < 2e-16 ***
## treatmentneonic:nest_temp  1.248e-02  3.105e-03  1.428e+04   4.020 5.85e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) trtmnt nst_tm
## treatmntnnc -0.074
## nest_temp   -0.023  0.209
## trtmntnnc:_  0.017 -0.280 -0.735
```

```
## $nest_temp
```


Predicted values of pc1
pc1
2.5
2.0
1.5
1.0
0.5
0.0
10
20
30
nest_temp



```
## pdf
##   2
```

```
## $nest_temp
```

```
## pdf
##   2
```

Colony demography

Load and clean data

##Growth models

```
gen.model <- glmer(total ~ log10(colony.size.at.tagging) * neonic *
    cold + growth.length + (1 | experimentalBlock) + (1 | Species),
    data = subset(demog.data, colony.size.at.tagging > 2 & alive.at.tagging ==
        1 & Species != "perplexus"), family = "poisson")
```

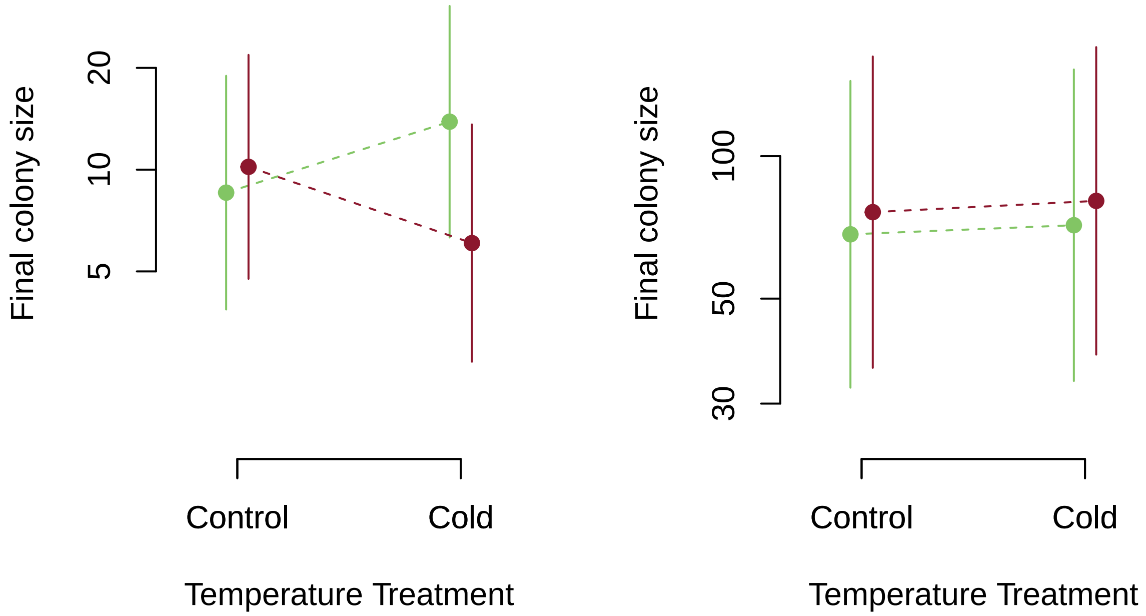
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00922731 (tol = 0.002, component 1)
```

```r
gen.model.exclude.large.colonies <- glmer(total ~ log10(colony.size.at.tagging) *
    neonic * cold + growth.length + (1 | experimentalBlock) +
    (1 | Species), data = subset(demog.data, colony.size.at.tagging >
    2 & colony.size.at.tagging < 30 & alive.at.tagging == 1 &
    Species != "perplexus"), family = "poisson")

gr.model <- lmer(growth.rate ~ log10(colony.size.at.tagging) *
    neonic * cold + (1 | experimentalBlock) + (1 | Species),
    data = subset(demog.data, colony.size.at.tagging > 2 & colony.size.at.tagging <
        30 & alive.at.tagging == 1 & Species != "perplexus"))


# ineline plot
par(mfcol = c(1, 2))
out <- plot_model(gen.model.exclude.large.colonies, type = "pred",
    terms = c("cold", "neonic", "colony.size.at.tagging [4]"),
    transform = "exp")
custom_plot(out, c(0.2, 1.6), c(5, 10, 20))
```

```
## [1] 0.2 1.6
## [1]  5 10 20
## [1] 0.9319349 1.0082704 1.1418005 0.7827807
```

```r
out <- plot_model(gen.model.exclude.large.colonies, type = "pred",
    terms = c("cold", "neonic", "colony.size.at.tagging [30]"),
    transform = "exp")
custom_plot(out, c(1.4, 2.4), c(30, 50, 100))
```

```
## [1] 1.4 2.4
## [1]  30  50 100
## [1] 1.834809 1.881678 1.854045 1.905451
```



```
# Figure 2: growth
setwd(output_folder)
pdf("Fig2.pdf", width = 4, height = 8)
par(mfcol = c(2, 1))
out <- plot_model(gen.model.exclude.large.colonies, type = "pred",
    terms = c("cold", "neonic", "colony.size.at.tagging [4]"),
    transform = "exp")
custom_plot(out, c(0.2, 1.6), c(5, 10, 20))
```

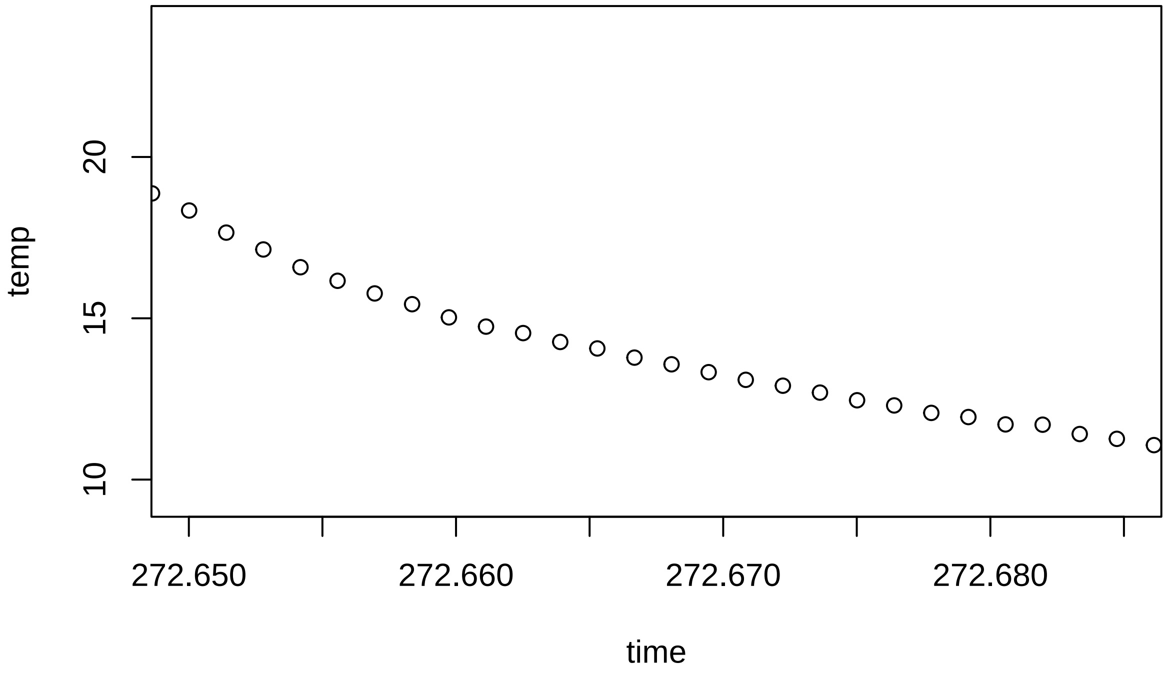
```
## [1] 0.2 1.6
## [1]  5 10 20
## [1] 0.9319349 1.0082704 1.1418005 0.7827807
```

```
out <- plot_model(gen.model.exclude.large.colonies, type = "pred",
    terms = c("cold", "neonic", "colony.size.at.tagging [30]"),
    transform = "exp")
custom_plot(out, c(1.4, 2.4), c(30, 50, 100))
```

```
## [1] 1.4 2.4
## [1]  30  50 100
## [1] 1.834809 1.881678 1.854045 1.905451
```

```
dev.off()
```

```
## pdf
##   2
```



```
## [1] 0.14156
```

```
# Confirm achieved nest temperature in manipulations above
tmp <- subset(behData, cold == TRUE & rampState == 3)  #Create a subset with data in cold-exposed colon

# Nest temperatures
nest.tmps <- aggregate(tmp$nestTemp, by = list(tmp$date), FUN = mean)
colnames(nest.tmps) <- c("timestamp", "nest_temp")
mean(nest.tmps$nest_temp)
```

```
## [1] 14.52631
```

```
median(nest.tmps$nest_temp)
```

```
## [1] 14.3125
```

```
# Chamber temps
ch.tmps <- aggregate(tmp$chamberTemp, by = list(tmp$date), FUN = mean)
colnames(ch.tmps) <- c("timestamp", "nest_temp")
mean(ch.tmps$nest_temp)
```

```
## [1] 10.14476
```

```
median(ch.tmps$nest_temp)
```

```
## [1] 9.75
```

```
setwd("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/AppletonT
filelist <- list.files(pattern = ".csv")

par(mfcol = c(4, 1))
for (i in 1:length(filelist)) {

    tempdata <- read.csv(filelist[i], skip = 1)
    tempdata <- tempdata[, 1:6]
    colnames(tempdata) <- c("Num", "Time", "air.temp", "ground.temp",
        "shallow.soil.temp", "deep.soil.temp")
    tempdata$air.temp.diff <- NaN
    tempdata$air.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$air.temp)
    tempdata$ground.temp.diff <- NaN
    tempdata$ground.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$ground.temp)
    tempdata$shallow.soil.temp.diff <- NaN
    tempdata$shallow.soil.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$shallow.soil.temp)
    tempdata$deep.soil.temp.diff <- NaN
    tempdata$deep.soil.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$deep.soil.temp)
    tempdata$unit <- i
    if (i == 1) {
        m.dat <- tempdata
    } else {
        m.dat <- rbind(m.dat, tempdata)
    }
}


temp.data <- gather(m.dat[, c("Time", "air.temp", "ground.temp",
    "shallow.soil.temp", "deep.soil.temp", "unit")], key = "location",
    value = "Temperature", air.temp, ground.temp, shallow.soil.temp,
    deep.soil.temp)
## Single colony
diff.data <- gather(m.dat[, c("Time", "air.temp.diff", "ground.temp.diff",
    "shallow.soil.temp.diff", "deep.soil.temp.diff", "unit")],
    key = "location", value = "Temperature", air.temp.diff, ground.temp.diff,
    shallow.soil.temp.diff, deep.soil.temp.diff)

# Clean out errant readings and nans
temp.data$Temperature[temp.data$Temperature < -10 | temp.data$Temperature >
    50] <- NA
diff.data$Temperature[diff.data$Temperature < -5 | diff.data$Temperature >
    5] <- NA

temp.data$location <- factor(temp.data$location, levels = c("air.temp",
    "ground.temp", "shallow.soil.temp", "deep.soil.temp"))
diff.data$location <- factor(diff.data$location, levels = c("air.temp.diff",
    "ground.temp.diff", "shallow.soil.temp.diff", "deep.soil.temp.diff"))
```

```
diff.data$Temperature <- abs(diff.data$Temperature)
# hist(temp.data$Temperature) hist(diff.data$Temperature)

setwd(output_folder)
pdf("FigS7.pdf")
par(mfcol = c(2, 1))
p1 <- ggplot(temp.data, aes(x = location, y = Temperature, fill = location)) +
    geom_violin() + geom_boxplot(width = 0.1, outlier.shape = NA) +
    ylim(0, 40) + geom_hline(yintercept = 14.3) + scale_fill_discrete(labels = c("Air",
    "Surface", "2 cm belowground", "10 cm belowground"), name = "Temp probe location") +
    theme(legend.position = "none")
p2 <- ggplot(diff.data, aes(x = Temperature, fill = location)) +
    geom_histogram(aes(y = ..density..)) + xlim(0, 0.5) + scale_fill_discrete(labels = c("Air",
    "Surface", "2 cm belowground", "10 cm belowground"), name = "Temp probe location")
grid.arrange(p1, p2)
```

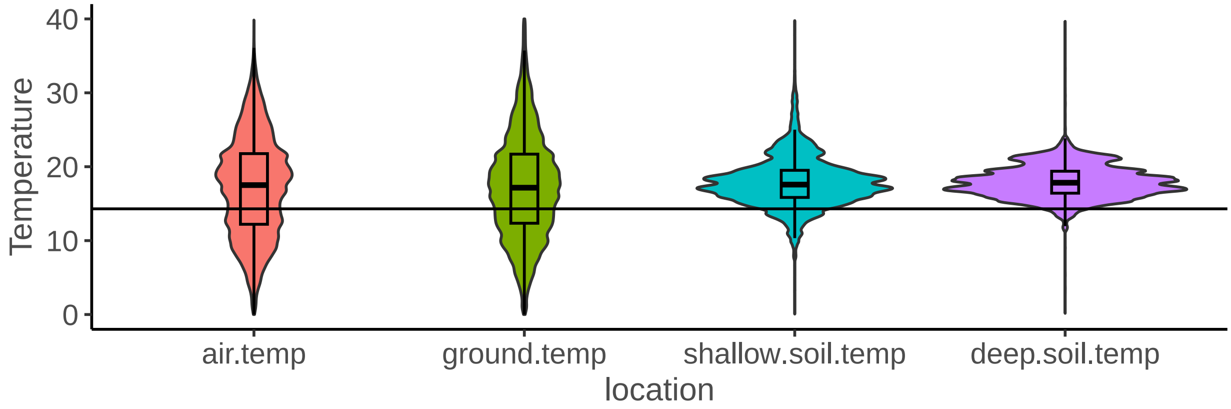
```
## Warning: Removed 12637 rows containing non-finite values (`stat_ydensity()`).

## Warning: Removed 12637 rows containing non-finite values (`stat_boxplot()`).

## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 100721 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 8 rows containing missing values (`geom_bar()`).
```

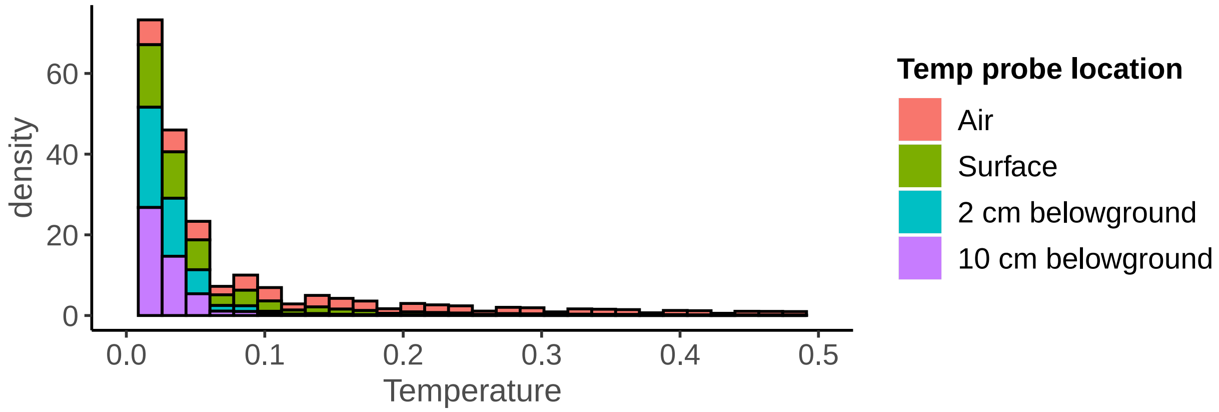
```
dev.off()
```

```
## pdf
##   2
```

```
# Plot for console
grid.arrange(p1, p2)
```

```
## Warning: Removed 12637 rows containing non-finite values (`stat_ydensity()`).

## Warning: Removed 12637 rows containing non-finite values (`stat_boxplot()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 100721 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 8 rows containing missing values (`geom_bar()`).
```





```
sm <- diff.data %>%
    group_by(location) %>%
    summarize(quant50 = quantile(Temperature, probs = 0.5, na.rm = TRUE),
        quant95 = quantile(Temperature, probs = 0.95, na.rm = TRUE),
        quant99 = quantile(Temperature, probs = 0.99, na.rm = TRUE))
print(sm)
```

```
## # A tibble: 4 x 4
##   location               quant50 quant95 quant99
##   <fct>                    <dbl>   <dbl>   <dbl>
## 1 air.temp.diff            0.220   1.58     2.64
## 2 ground.temp.diff         0.0400  0.460    1.82
## 3 shallow.soil.temp.diff   0.0200  0.0800   0.190
## 4 deep.soil.temp.diff      0.0200  0.0700   0.130
```

```
# Get range of dates
dates <- unique(parse_date_time(m.dat$Time, "mdy HMS"))
min(dates)
```

```
## [1] "2019-08-08 05:54:02 UTC"
```

```
max(dates)
```

```
## [1] "2019-10-02 12:59:35 UTC"
```

```
difftime(min(dates), max(dates))
```

```
## Time difference of -55.29552 days
```