common garden tank analysis

luke reding

September 16, 2016

Contents

Before we begin...

- LL: large male only tanks
- SS: small male only tanks
- INT: intermediate male only tanks
- LS: large and small male tanks
- FF: female only tanks
- *total courtship* refers to any sort of courtship-esque event by any male. (We should have Rachel E give us a rigourous def of what she was looking for w/r/t courtship).
- *total aggression* refers to any sort of aggressive or chase event, regardless of fish identites

to do:

- break things apart by round / replicate

```
library(magrittr)
library(rjson)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```r
# helper function: returns NA if the feild doesn't exist, otherwise returns the value
ret <- function(entry){
  if(is.null(entry)){
    return("NA")
  }
  else{
    return(entry)
  }
}

# bitchin' colors
g <- c(0.9649929502500952, 0.9531129905215493, 0.9191510752734625, 0.8593537612656912, 0.881760421173651
greens <- c()
for(i in seq(1, length(g), by = 3)){
  greens %<>% c(rgb(g[i], g[1+i], g[2+i]))
}
greens <- colorRampPalette(greens)

di <- c("#0072B2", "#009E73", "#D55E00", "#CC79A7", "#F0E442", "#65B4E9")
diverging <- function(n){
  return(rep(di,10)[1:n])
}

 theme
```

```
## function (..., complete = FALSE, validate = TRUE)
## {
##     elements <- list(...)
##     if (!is.null(elements$axis.ticks.margin)) {
##         warning("`axis.ticks.margin` is deprecated. Please set `margin` property ",
##             " of `axis.text` instead", call. = FALSE)
##         elements$axis.ticks.margin <- NULL
##     }
##     if (validate) {
##         mapply(validate_element, elements, names(elements))
##     }
##     structure(elements, class = c("theme", "gg"), complete = complete,
##         validate = validate)
## }
## <environment: namespace:ggplot2>
```

```
theme_clean <- function(font_size = 18, font_family = "", line_size = .5) {
  half_line <- font_size / 2
  small_rel <- 0.857
  small_size <- small_rel * font_size

  theme_grey(base_size = font_size, base_family = font_family) %+replace%
    theme(
      rect              = element_rect(fill = "transparent", colour = NA, color = NA, size = 0, linetyp
      text              = element_text(family = font_family, face = "plain", colour = "black",
                                       size = font_size, hjust = 0.5, vjust = 0.5, angle = 0, lineheigh
                                       margin = ggplot2::margin(), debug = FALSE),
      axis.text         = element_text(colour = "black", size = small_size),
      #axis.title        = element_text(face = "bold"),
      axis.text.x       = element_text(margin = ggplot2::margin(t = small_size / 4), vjust = 1),
      axis.text.y       = element_text(margin = ggplot2::margin(r = small_size / 4), hjust = 1),
      axis.title.x      = element_text(
        margin = ggplot2::margin(t = small_size / 2, b = small_size / 4)
      ),
      axis.title.y      = element_text(
        angle = 90,
        margin = ggplot2::margin(r = small_size / 2, l = small_size / 4),
      ),
      axis.ticks        = element_line(colour = "black", size = line_size),
      axis.line.x       = element_line(colour = "black", size = line_size),
      axis.line.y       = element_line(colour = "black", size = line_size),
      legend.key        = element_blank(),
      legend.margin     = grid::unit(0.1, "cm"),
      legend.key.size   = grid::unit(1, "lines"),
      legend.text       = element_text(size = rel(small_rel)),
      #    legend.position   = c(-0.03, 1.05),
      # legend.justification = c("left", "right"),
      panel.background  = element_blank(),
      panel.border      = element_blank(),
      panel.grid.major  = element_blank(),
      panel.grid.minor  = element_blank(),
      strip.text        = element_text(size = rel(small_rel)),
      strip.background  = element_rect(fill = "grey80", colour = "grey50", size = 0),
      plot.background   = element_blank(),
      plot.title        = element_text(face = "bold",
                                       size = font_size,
                                       margin = ggplot2::margin(b = half_line))
    )
}
```

Import json files.

```
files <- list.files("/Users/lukereding/Documents/common_garden/data", pattern = "*.json", full.names = T
```

We have scored 318 of these short, 10-second videos.

reading in the `json` files

Now that we've got the list of the file names, we'll read them in one by one, extract various quantities, and organize everything into a data frame.

A single json file looks something like this:

```
rjson::fromJSON(file=files[5])
```

```
## $focal_juvenile_locations
## $focal_juvenile_locations[[1]]
## [1] 331 541
##
## $focal_juvenile_locations[[2]]
## [1] 720 475
##
## $focal_juvenile_locations[[3]]
## [1] 909 988
##
## $focal_juvenile_locations[[4]]
## [1] 669 968
##
## $focal_juvenile_locations[[5]]
## [1] 305 972
##
## $focal_juvenile_locations[[6]]
## [1] 364 992
##
##
## $large_vs_female
## [1] "0"
##
## $number_small_male
## [1] 0
##
## $large_vs_large
## [1] "0"
##
## $pairwise_distance_females
## [1] 423.7
##
## $tank_id
## [1] "INT1"
##
## $small_courting
## [1] "0"
##
## $small_vs_female
## [1] "0"
##
## $pairwise_distance_large_males
## [1] 497.32
##
## $total_fish
```

```
## [1] 13
##
## $comments
## [1] ""
##
## $large_courting
## [1] "0"
##
## $large_male_locations
## $large_male_locations[[1]]
## [1] 666 611
##
## $large_male_locations[[2]]
## [1] 624 803
##
## $large_male_locations[[3]]
## [1] 951 829
##
## $large_male_locations[[4]]
## [1] 1435  846
##
##
## $male_chased_juvenile
## [1] "0"
##
## $small_male_locations
## list()
##
## $pairwise_distance_small_males
## [1] "NA"
##
## $pairewise_distance_juvs
## [1] 466.24
##
## $female_vs_male
## [1] "1"
##
## $date
## [1] "03-20-2016"
##
## $time_of_clip
## [1] "1min"
##
## $number_focal
## [1] 6
##
## $intermediate_courting
## [1] "0"
##
## $int_vs_female
## [1] "1"
##
## $model_female_locations
## $model_female_locations[[1]]
```

```
## [1] 608 868
##
## $model_female_locations[[2]]
## [1] 568 911
##
## $model_female_locations[[3]]
## [1] 1191  947
##
##
## $int_vs_int
## [1] "0"
##
## $female_vs_female
## [1] "0"
##
## $large_vs_small
## [1] "0"
##
## $number_large_male
## [1] 4
##
## $number_model_female
## [1] 3
##
## $video_name
## [1] "/Volumes/THE MOLLUSK/common_garden_weekly_videos/short_videos/03_20_2016_INT1_1min.mp4"
```

```
library(rjson)

n <- length(files)
# create data frame
df <- data.frame("video_name" = character(n),
                 "large_vs_large" =integer(n),
                 "large_vs_small"= integer(n),
                 "int_vs_int"= integer(n),
                 "large_vs_female"= integer(n),
                 "small_vs_female"= integer(n),
                 "int_vs_female"= integer(n),
                 "female_vs_female"= integer(n),
                 "female_vs_male"= integer(n),
                 "large_courting"= integer(n),
                 "intermediate_courting"= integer(n),
                 "small_courting"= integer(n),
                 "number_focal"= integer(n),
                 "number_large_male"= integer(n),
                 "number_small_male"= integer(n),
                 "number_model_female"= integer(n),
                 "male_chased_juvenile"= integer(n),
                 "tank_id"= character(n),
                 "pairwise_distance_large_males"= double(n),
                 "pairwise_distance_small_males"= double(n),
                 "pairwise_distance_females"= double(n),
                 "pairewise_distance_juvs"= double(n),
                 "total_fish"= integer(n),
```

```
                  "date"= character(n),
                 "comments" = character(n),
                 "small_vs_small" = integer(n),
                 "observer" = character(n),
                 stringsAsFactors=FALSE)



for(i in 1:length(files)){
  # read in data
  json_data <- rjson::fromJSON(file=files[i])
  # print(i)
  # extract the data
  df$video_name[i] <-  ret(json_data$video_name) %>% as.character
  df$large_vs_large[i] <-  ret(json_data$large_vs_large) %>% as.numeric
  df$large_vs_small[i] <-  ret(json_data$large_vs_small)%>% as.numeric
  df$int_vs_int[i] <-  ret(json_data$int_vs_int)%>% as.numeric
  df$large_vs_female[i] <-  ret(json_data$large_vs_female)%>% as.numeric
  df$small_vs_female[i] <-  ret(json_data$small_vs_female)%>% as.numeric
  df$int_vs_female[i] <-  ret(json_data$int_vs_female)%>% as.numeric
  df$female_vs_female[i] <-  ret(json_data$female_vs_female)%>% as.numeric
  df$female_vs_male[i] <-  ret(json_data$female_vs_male)%>% as.numeric
  df$large_courting[i] <-  ret(json_data$large_courting)%>% as.numeric
  df$intermediate_courting[i] <-  ret(json_data$intermediate_courting)%>% as.numeric
  df$small_courting[i] <-  ret(json_data$small_courting)%>% as.numeric
  df$number_focal[i] <-  ret(json_data$number_focal)%>% as.numeric
  df$number_large_male[i] <-  ret(json_data$number_large_male)%>% as.numeric
  df$number_small_male[i] <-  ret(json_data$number_small_male)%>% as.numeric
  df$number_model_female[i] <-  ret(json_data$number_model_female)%>% as.numeric
  df$male_chased_juvenile[i] <-  ret(json_data$male_chased_juvenile)%>% as.numeric
  df$tank_id[i] <-  ret(json_data$tank_id) %>% as.character
  df$pairwise_distance_large_males[i] <-  ret(json_data$pairwise_distance_large_males)%>% as.numeric
  df$pairwise_distance_small_males[i] <-  ret(json_data$pairwise_distance_small_males)%>% as.numeric
  df$pairwise_distance_females[i] <-  ret(json_data$pairwise_distance_females)%>% as.numeric
  df$pairewise_distance_juvs[i] <-  ret(json_data$pairewise_distance_juvs)%>% as.numeric
  df$total_fish[i] <-  ret(json_data$total_fish)%>% as.numeric
  df$date[i] <-  ret(json_data$date) %>% as.character
  df$comments[i] <-  ret(json_data$comments) %>% as.character
  df$observer[i] <- ret(json_data$observer) %>% as.character
  df$small_vs_small[i] <- ret(json_data$small_vs_small) %>% as.character

}

# get treatment
df$treatment <- gsub("[[:digit:]]","", df$tank_id)

# get total courtship events:
df %<>% mutate(total_courtship = large_courting + small_courting + intermediate_courting)

# get overall (total) aggression
df %<>% mutate(total_aggression = large_vs_large + large_vs_small + int_vs_int + large_vs_female + int_

# make column that denotes the date / tank combo
```

```
df %<>% unite(col = id, tank_id, date, remove = FALSE)
```

wrangling

Now I define a couple functions that are useful went collapsing all the trials from a given day into a single row. This isn't a particularly elegant wasy of doing things, but it works.

```
# function to use to collapse all the videos from one day into a single row
avg_if_numeric <- function(x){
  if(!is.character(x)){
    return(mean(x, na.rm=T))
  }
  else{
    if(length(levels(factor(x))) > 1){
      warning("you are trying to collapse non-matching characters. will take the first one, but beware.
    }
    return(x[1])
  }
}

sum_if_numeric <- function(x){
  if(!is.character(x)){
    return(sum(x, na.rm=T))
  }
  else{
    if(length(levels(factor(x))) > 1){
      warning("you are trying to collapse non-matching characters. will take the first one, but beware.
    }
    return(x[1])
  }
}
```

I now create two data frame. `df_avg` contains average behaviors from one of these videos. `df` contains sums.

Why do we need two data frames?

For some of the videos, for whatever reason, we only have two instead of three short videos. If we want to use all the data we have, we should use `df_avg` because data from tanks where we only have two videos are comparable to tanks where we have three videos. If we want to be conservative and make sure we have a totally complete three short videos for each tank / week, we can use `df.`

```
# to get the sum total of each behavior for each day
# exclude videos for which there are fewer than three json files
# first get the names of the videos for which there are 3 videos analyzed
good_videos = df %>%
  group_by(id) %>%
  tally %>%
  filter(n ==3) %>%
  .$id
df$good_video <- ifelse(df$id %in% good_videos, TRUE, FALSE)

# the dataframe containing the averages
```

```
df_avg <- df %>%
  group_by(id) %>%
  summarise_each(funs(if(is.numeric(.)) mean(., na.rm = TRUE) else first(.)))

# the more complete dataframe containing the sums
df %<>%
  filter(good_video == TRUE) %>%
  group_by(id) %>%
  summarise_each(funs(if(is.numeric(.)) sum(., na.rm = TRUE) else first(.)))

df$date %<>% as.Date(format = "%m-%d-%Y")
df_avg$date %<>% as.Date(format = "%m-%d-%Y")
df_avg$treatment %<>% factor
df$treatment %<>% factor
```

df_avg has 111 rows while df has 96, meaning that we're missing some data for 15 videos.

---

visualizing the dataset

We can use the function define below to (1) do a permutational ANOVA, (2) do Tukey post-hoc testing, and (3) add labels to the plots showing which groups are signficantly different from other groups:

(things to add: overall anova significance)

```
generate_label_df <- function(x, y, dataframe){
  require(lmPerm)
  require(multcompView)
  require(magrittr)
  arguments <- as.list(match.call())
  # do the anova
  model <- aovp(eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
  print(summary(model))
  if(summary(model)[[1]][5][1,] > 0.05){
    warning("the ANOVA is not significant")
  }
HSD <- TukeyHSD(model, ordered = FALSE, conf.level = 0.95)
  print(HSD)

  # Extract labels and factor levels from Tukey post-hoc
  Tukey.levels <- HSD[[1]][,4]
  Tukey.labels <- multcompLetters(Tukey.levels)['Letters']
  plot.labels <- names(Tukey.labels[['Letters']])

  boxplot.df <- split(dataframe, eval(arguments$x, dataframe)) %>%
    lapply(., function(z) eval(arguments$y, z)) %>%
    lapply(., max, na.rm=T) %>%
    unlist %>%
    add(((range(.)[2] - range(.)[1])*0.07))

  boxplot.df <- data.frame(
```

```
    height = boxplot.df,
    names(boxplot.df)
  )
  names(boxplot.df)[2] <- "plot.labels"

  # Create a data frame out of the factor levels and Tukey's homogenous group letters
  plot.levels <- data.frame(plot.labels,
                            labels = Tukey.labels[['Letters']],
                            stringsAsFactors = FALSE)

  # Merge it with the labels
  labels.df <- merge(plot.levels, boxplot.df, sort = FALSE)
  print(labels.df)

  return(labels.df)
}
```

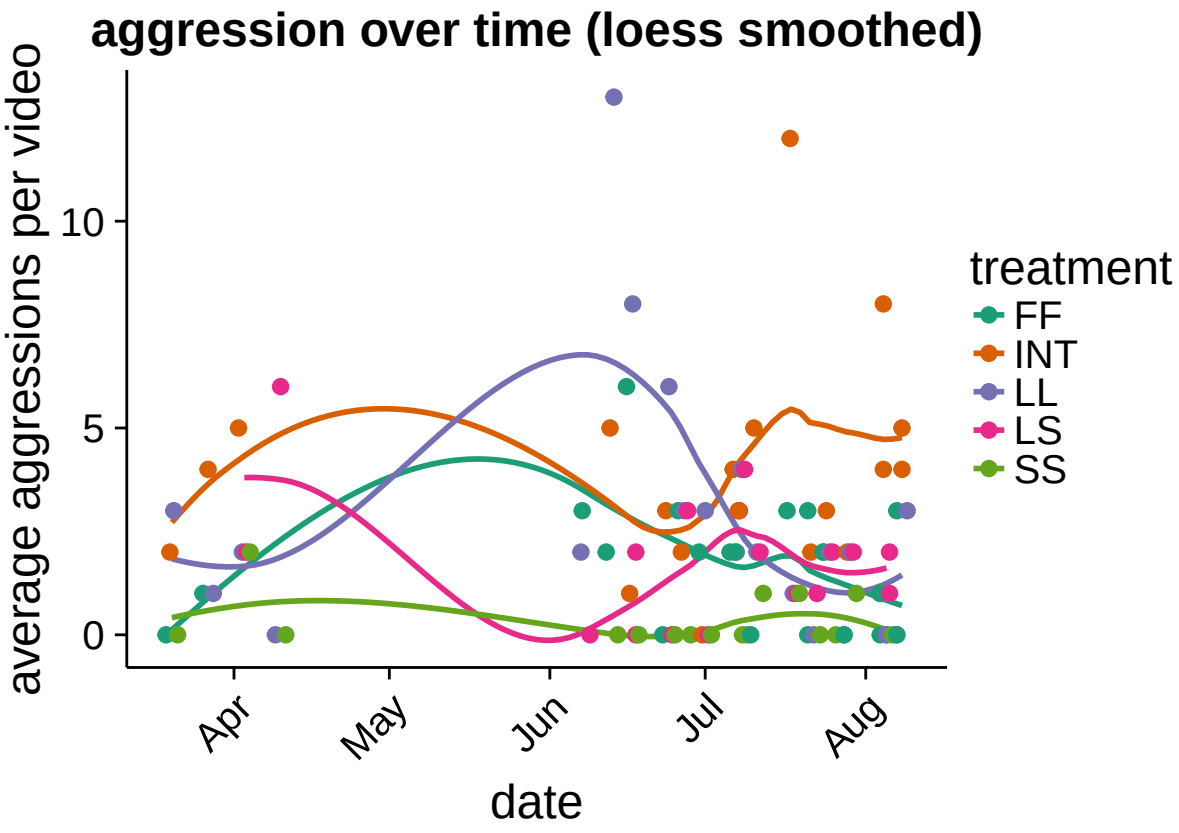
Example useage:

```
ggplot(df_avg, aes(y=total_courtship, x=treatment)) +
  geom_boxplot() +
  geom_text(data = generate_label_df(x = treatment, y = total_courtship,df_avg),
            aes(x = plot.labels, y = height, label = labels)
  ) +
  theme_minimal()
```

time trends

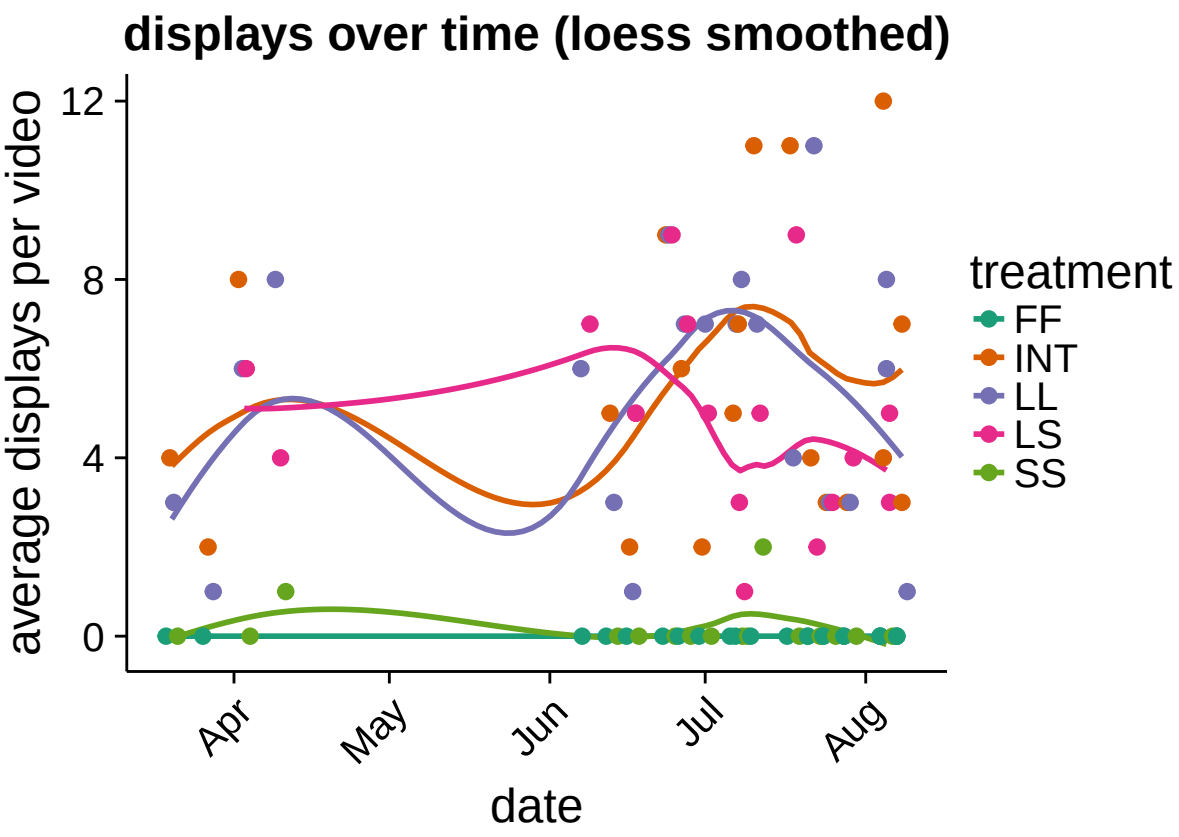
```
# for adding sample sizes to plots
give.n <- function(x){
  return(c(y = median(x)*1.02, label = length(x)))
  # experiment with the multiplier to find the perfect position
}

ggplot(df, aes(date, total_aggression, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=3)) +
  geom_smooth(se=F)+
  theme_clean() +
  geom_point(size = 2.4, position = position_dodge(width=3)) +
  ylab("average aggressions per video") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("aggression over time (loess smoothed)")
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

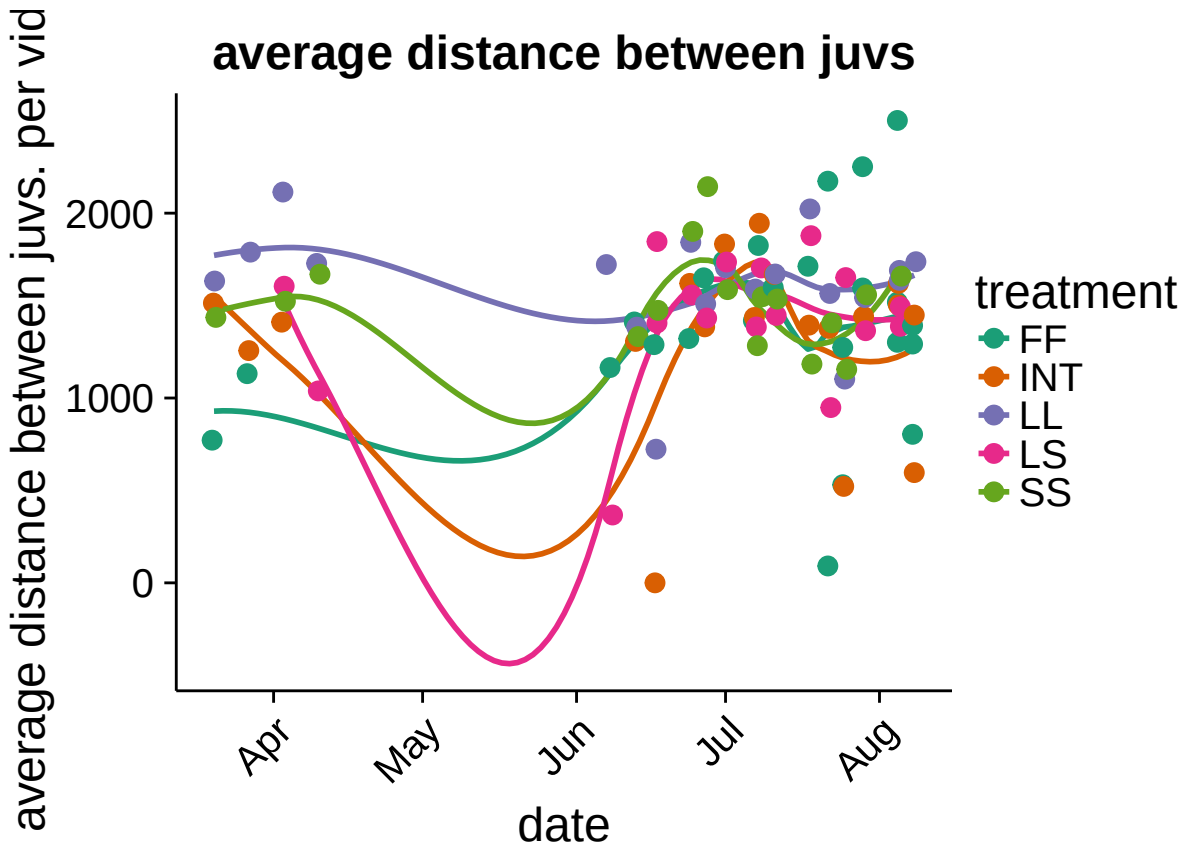



```
ggplot(df, aes(date, total_courtship, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=3)) +
  theme_clean() +
  geom_smooth(se=F)+
  geom_point(size = 2.4, position = position_dodge(width=3)) +
  ylab("average displays per video") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("displays over time (loess smoothed)")
```

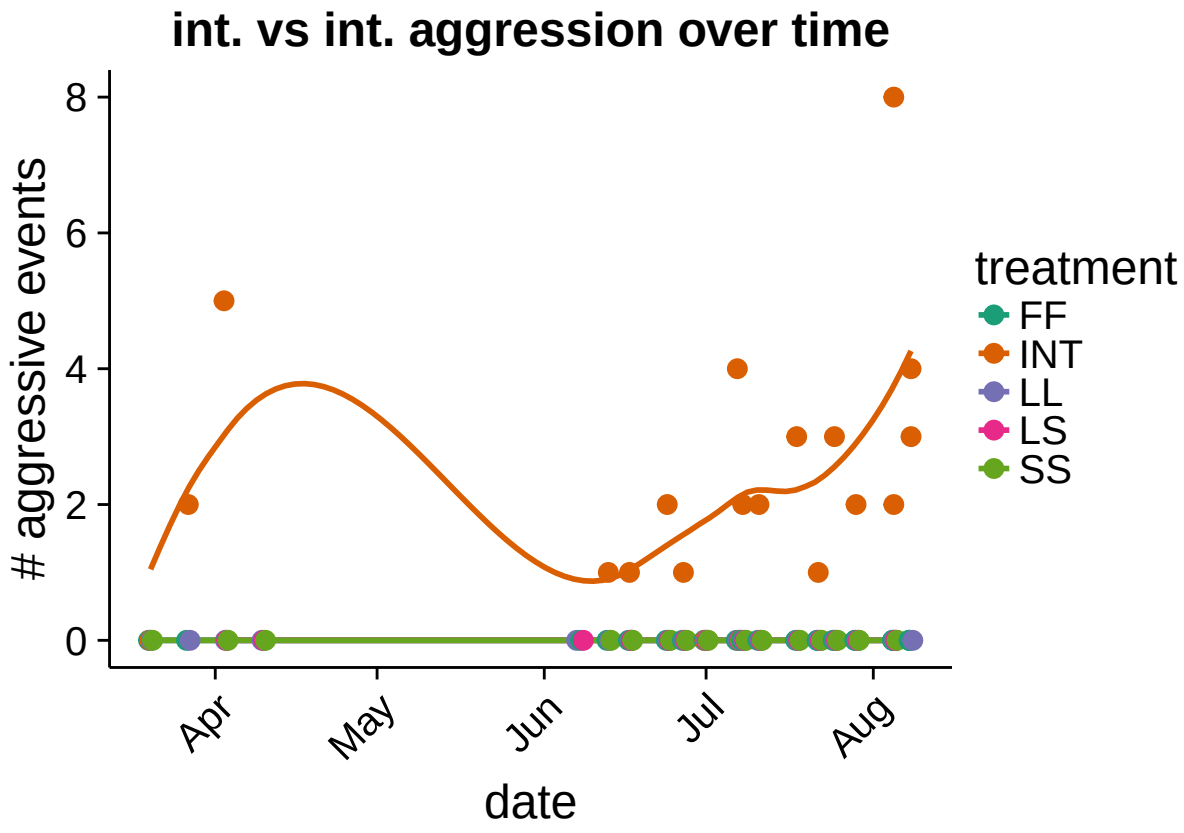
```
## Warning: position_dodge requires non-overlapping x intervals
```



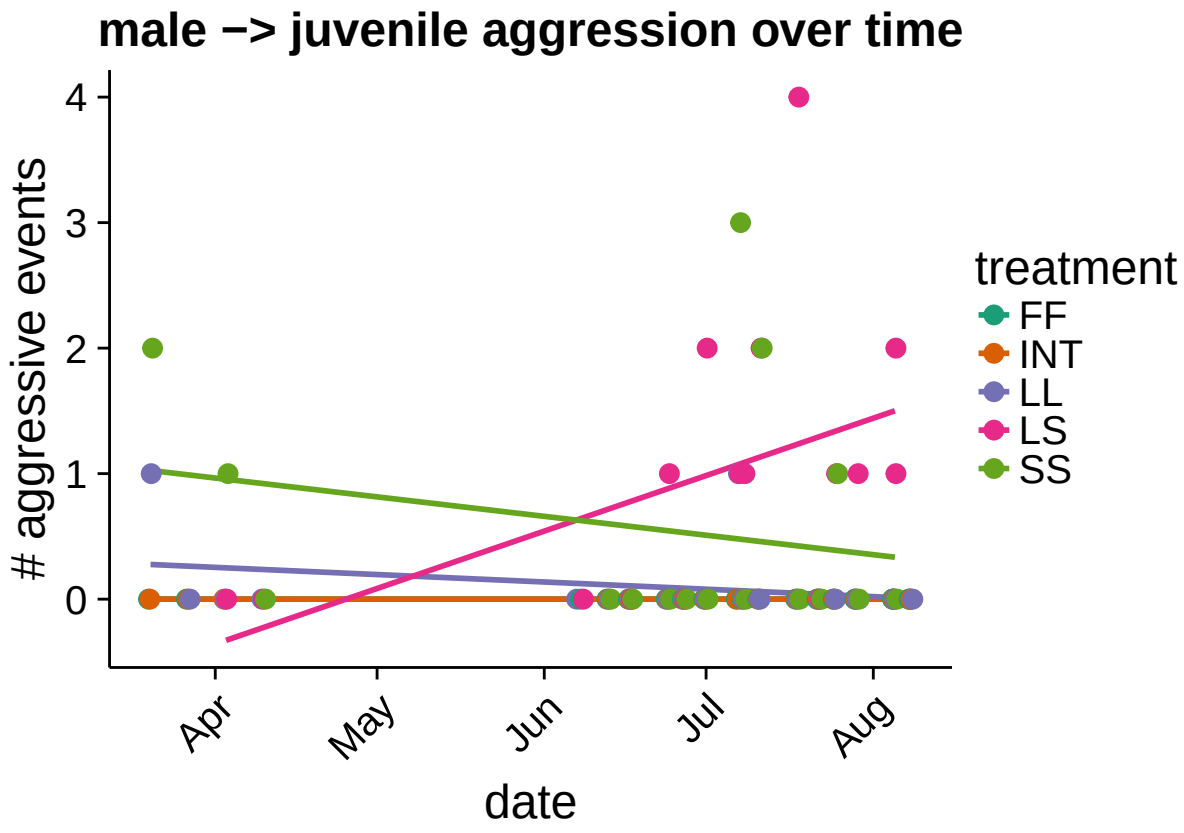

```
ggplot(df, aes(date, pairewise_distance_juvs, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=1)) +
  theme_clean() +
  geom_smooth(se=F)+
  geom_point(size=3, position = position_dodge(width=1)) +
  ylab("average distance between juvs. per video") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("average distance between juvs")
```



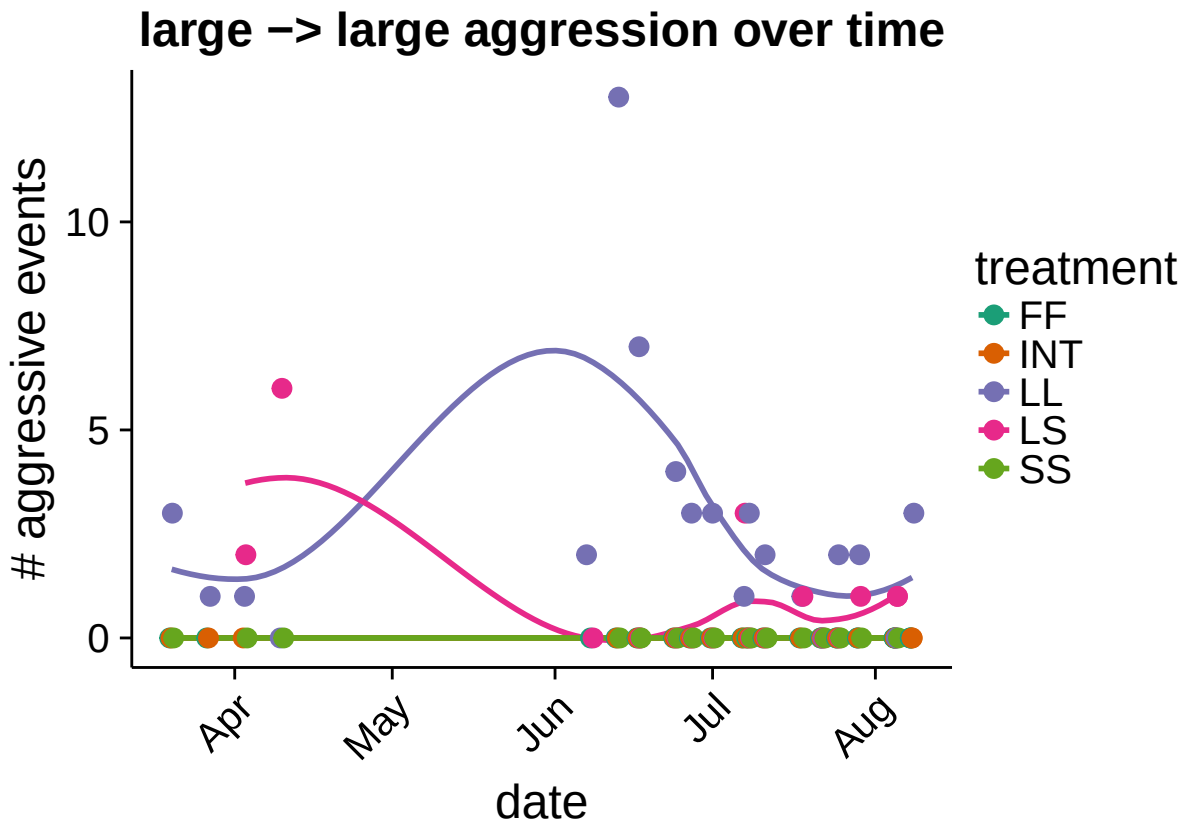

```
ggplot(df, aes(date, int_vs_int, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=1)) +
  theme_clean() +
  geom_smooth(se=F)+
  geom_point(size=3, position = position_dodge(width=1)) +
  ylab("# aggressive events") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("int. vs int. aggression over time")
```



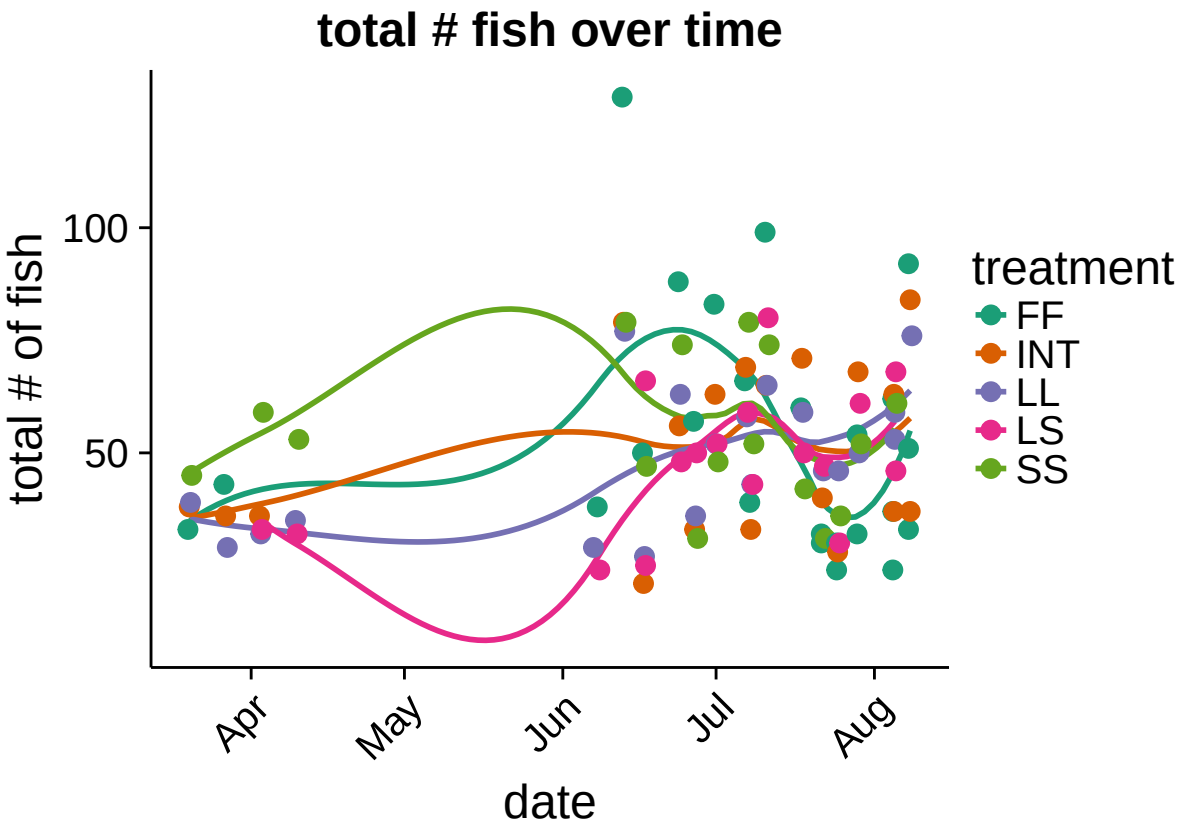

```
ggplot(df, aes(date, male_chased_juvenile, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=1)) +
  theme_clean() +
  geom_smooth(se=F, method="lm")+
  geom_point(size=3, position = position_dodge(width=1)) +
  ylab("# aggressive events") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("male -> juvenile aggression over time")
```




```
ggplot(df, aes(date, large_vs_large, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=1)) +
  theme_clean() +
  geom_smooth(se=F)+
  geom_point(size=3, position = position_dodge(width=1)) +
  ylab("# aggressive events") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("large -> large aggression over time")
```




```
ggplot(df, aes(date, total_fish, color = treatment)) +
  # geom_line(size=1.25, position = position_dodge(width=1)) +
  theme_clean() +
  geom_smooth(se=F)+
  geom_point(size=3, position = position_dodge(width=1)) +
  ylab("total # of fish") +
  # scale_color_manual(values=greens(5)[2:5]) +
  scale_colour_brewer(palette = "Dark2") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_date() +
  ggtitle("total # fish over time")
```



figures for the grant

sums of behaviors

```
require(cowplot)
```

```
## Loading required package: cowplot
```

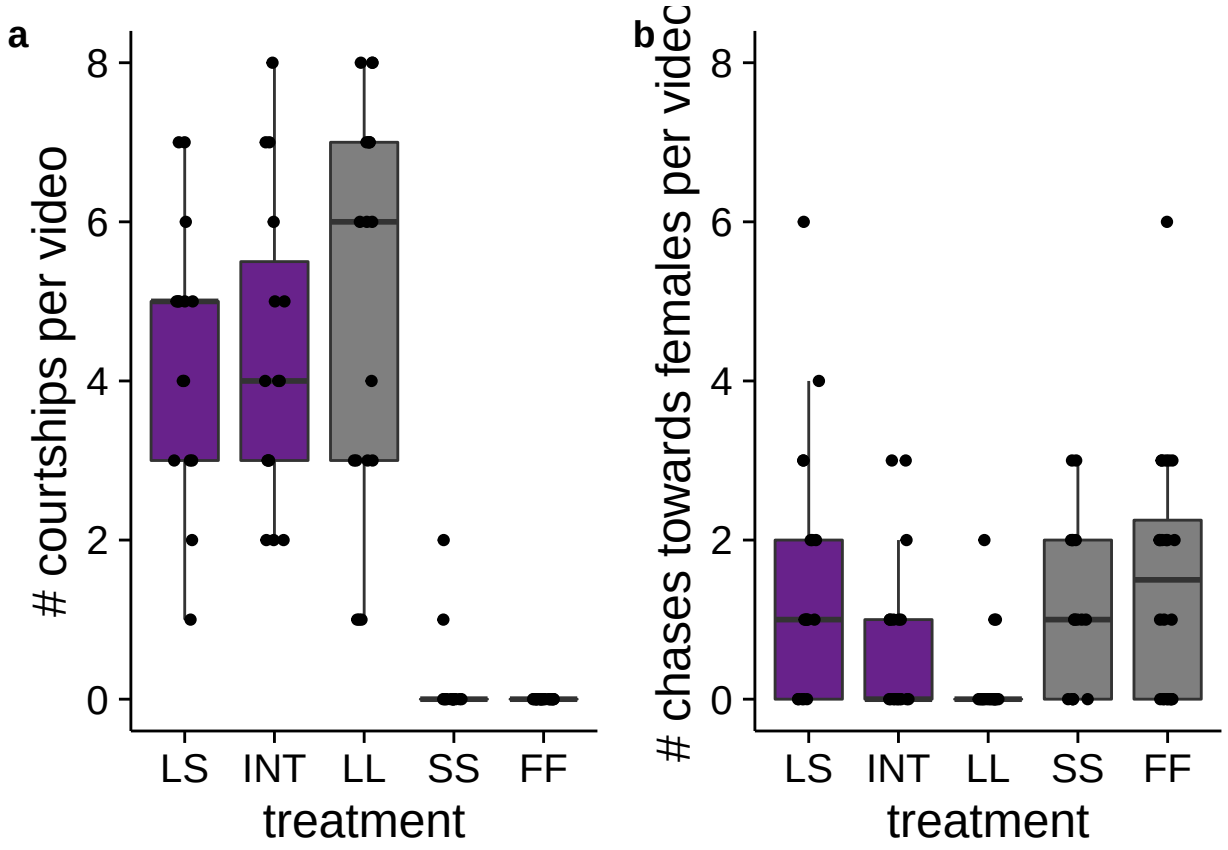
```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```
df$treatment %<>% factor(levels = c("LS", "INT", "LL","SS","FF"))

courts <- df %>%
  filter(good_video == TRUE) %>%
  mutate(aggresion_towards_females = large_vs_female + small_vs_female + int_vs_female + female_vs_fema
  ggplot(aes(treatment, total_courtship)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  # geom_jitter(width=0.3, height=0.15, aes(size = aggresion_towards_females)) +
  geom_jitter(width=0.3, height=0) +
```

```
  scale_fill_manual(values=c("darkorchid4","darkorchid4",rep("grey50",3)), guide=F) +
  # scale_size(name = "chases towards\nfemales") +
  ylab("# courtships per video") +
  # ggtitle("number of courtship events per video") +
  theme_clean() +
  ylim(c(0,8))

towards_females <- df %>%
  filter(good_video == TRUE) %>%
  mutate(aggression_towards_males = large_vs_large + large_vs_small + int_vs_int + female_vs_male) %>%
  mutate(aggresion_towards_females = large_vs_female + small_vs_female + int_vs_female + female_vs_fema
  ggplot(aes(treatment, aggresion_towards_females)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  # geom_jitter(width=0.3, height=0.15, aes(size = total_courtship)) +
  geom_jitter(width=0.3, height=0) +
  scale_fill_manual(values=c("darkorchid4","darkorchid4",rep("grey50",3)), guide=F) +
  # scale_size(name = "average displays\nper video") +
  ylab("# chases towards females per video") +
  theme_clean() +
  ylim(c(0,8)) +
  theme(legend.justification=c(0,1.1), legend.position=c(0.05,1))

(x <- plot_grid(courts, towards_females, labels = c("a","b")))
```

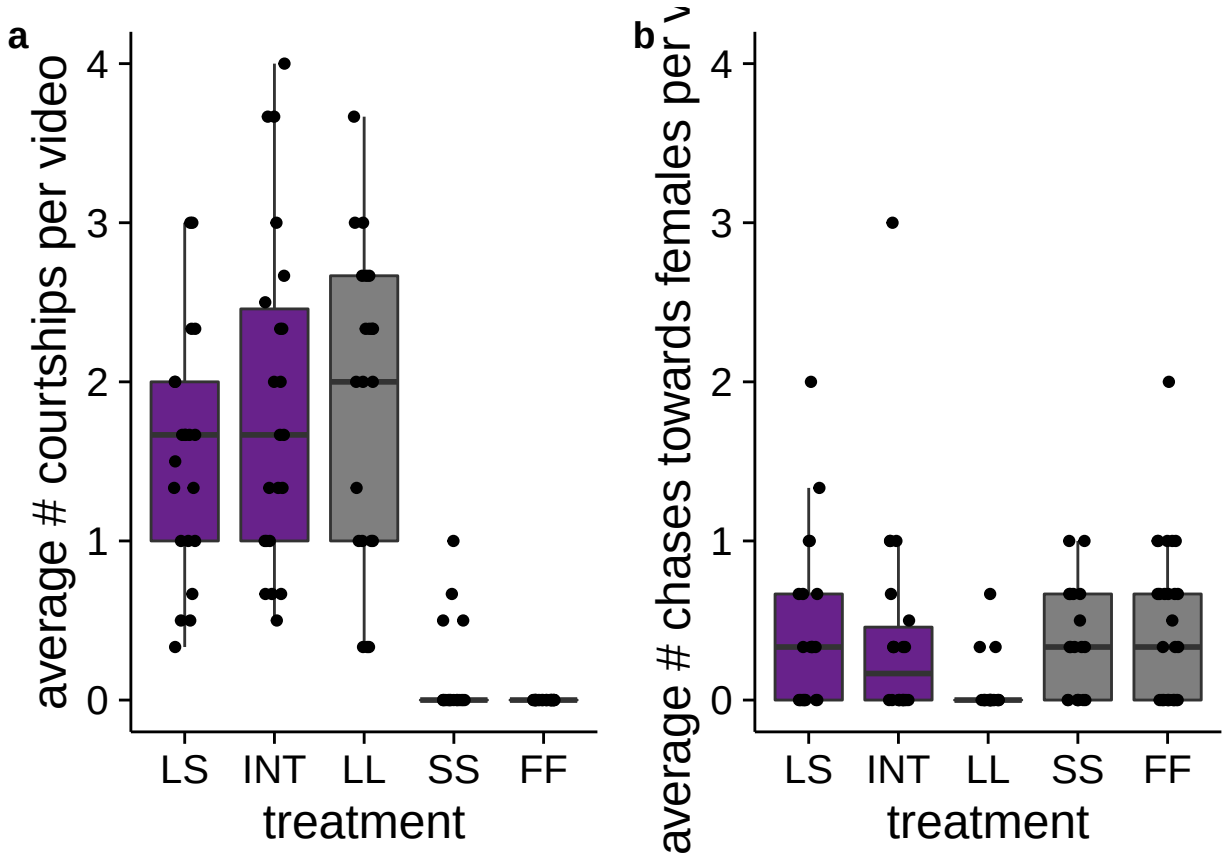
```
## Warning: Removed 8 rows containing non-finite values (stat_boxplot).

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

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## Warning: Removed 1 rows containing missing values (geom_point).
```



averages of behaviors

```
df_avg$treatment %<>% factor(levels = c("LS", "INT", "LL","SS","FF"))

courts <- df_avg %>%
  mutate(aggresion_towards_females = large_vs_female + small_vs_female + int_vs_female + female_vs_fema
  ggplot(aes(treatment, total_courtship)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  # geom_jitter(width=0.3, height=0.15, aes(size = aggresion_towards_females)) +
  geom_jitter(width=0.3, height=0) +
  scale_fill_manual(values=c("darkorchid4","darkorchid4",rep("grey50",3)), guide=F) +
  # scale_size(name = "chases towards\nfemales") +
  ylab("average # courtships per video") +
  # ggtitle("number of courtship events per video") +
  theme_clean() +
  ylim(c(0,4))

towards_females <- df_avg %>%
  mutate(aggression_towards_males = large_vs_large + large_vs_small + int_vs_int + female_vs_male) %>%
  mutate(aggresion_towards_females = large_vs_female + small_vs_female + int_vs_female + female_vs_fema
  ggplot(aes(treatment, aggresion_towards_females)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  # geom_jitter(width=0.3, height=0.15, aes(size = total_courtship)) +
  geom_jitter(width=0.3, height=0) +
  scale_fill_manual(values=c("darkorchid4","darkorchid4",rep("grey50",3)), guide=F) +
```

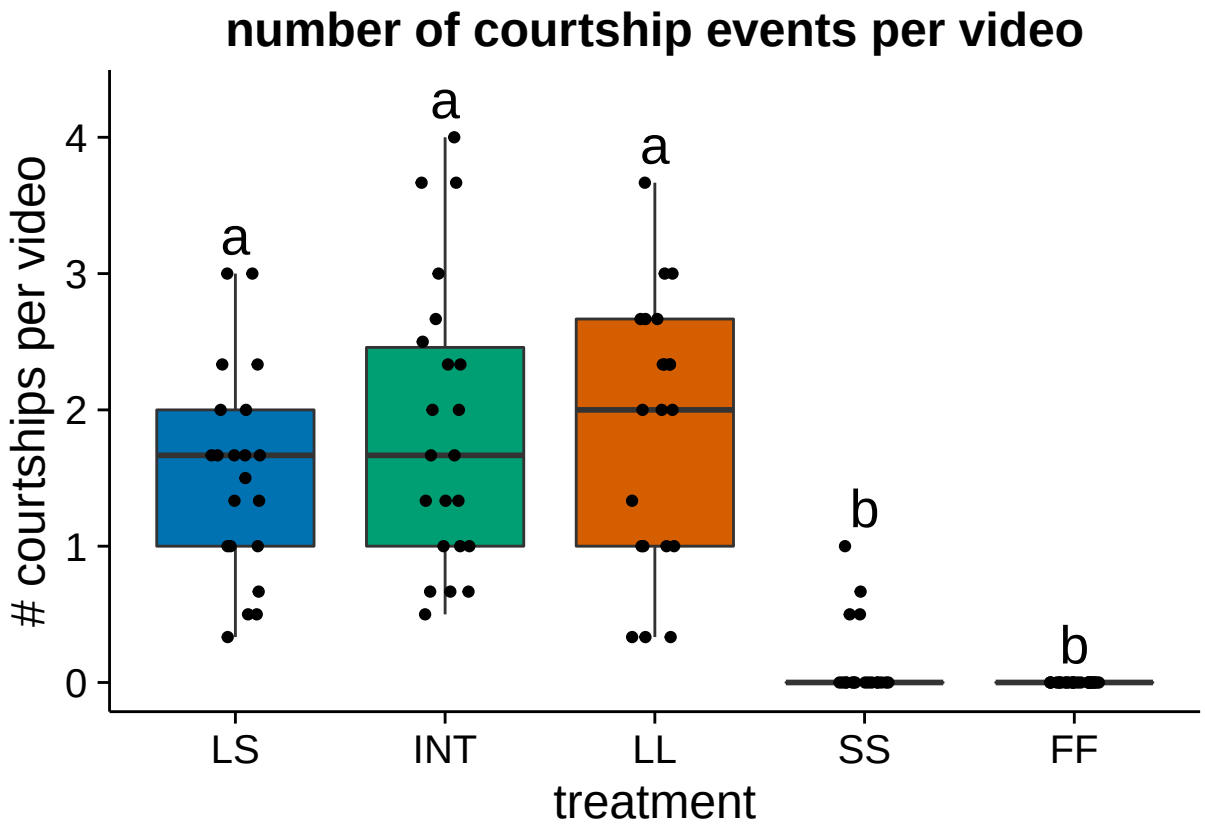
```
  # scale_size(name = "average displays\nper video") +
  ylab("average # chases towards females per video") +
  theme_clean() +
  ylim(c(0,4)) +
  theme(legend.justification=c(0,1.1), legend.position=c(0.05,1))

(x <- plot_grid(courts, towards_females, labels=c("a","b")))
```



other

```
df_avg %>%
  ggplot(aes(treatment, total_courtship)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  geom_jitter(width=0.3, , height=0) +
  scale_fill_manual(values=diverging(5), guide=F) +
  # scale_fill_brewer(palette = "Dark2", guide= F) +
  scale_size(name = "total aggressions") +
  ylab("# courtships per video") +
  ggtitle("number of courtship events per video") +
  theme_clean() +
  geom_text(data = generate_label_df(x = treatment, y = total_courtship, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

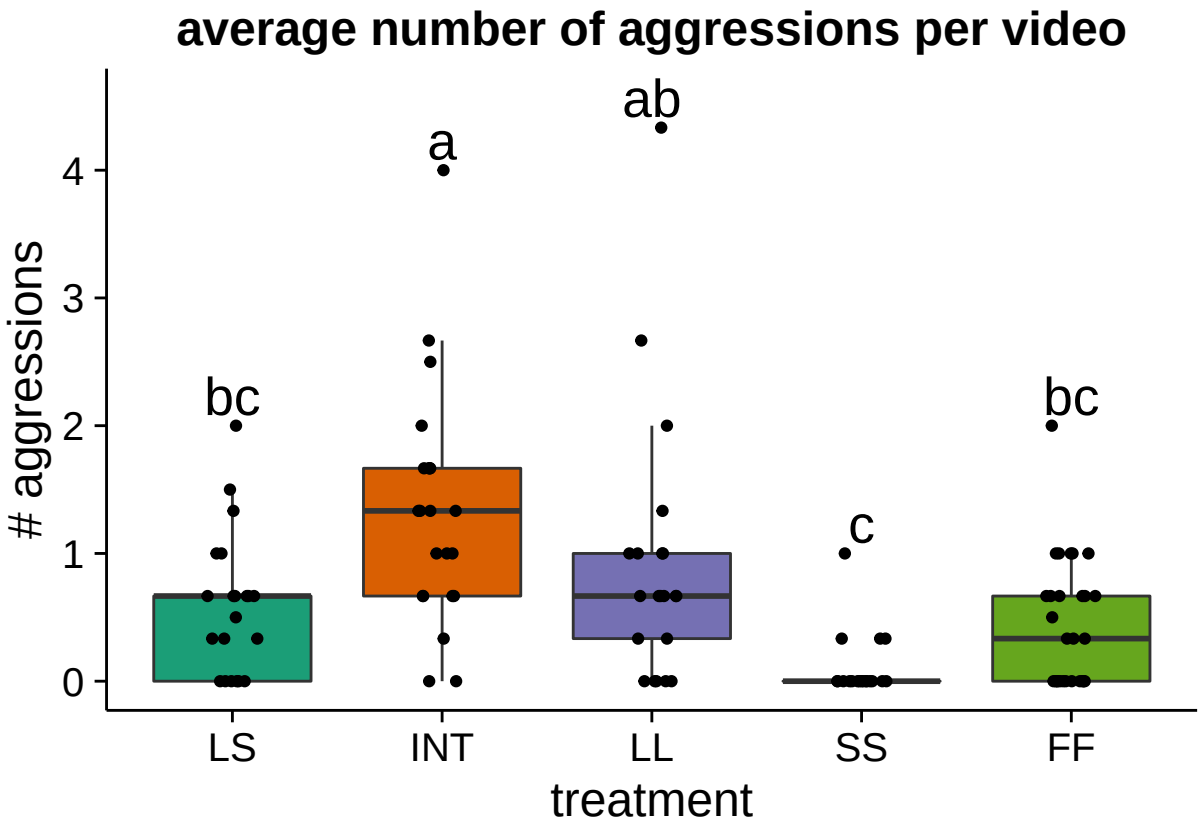
```
## Loading required package: lmPerm

## Loading required package: multcompView

## [1] "Settings:  unique SS "
## Component 1 :
##                               Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4   80.237   20.0592 5000 < 2.2e-16 ***
## Residuals                    106   55.135    0.5201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##                diff        lwr        upr     p adj
## INT-LS  0.331890332 -0.2787441  0.9425248 0.5593615
## LL-LS   0.341269841 -0.2764242  0.9589639 0.5433401
## SS-LS  -1.391395155 -2.0251359 -0.7576544 0.0000002
## FF-LS  -1.531746032 -2.1095459 -0.9539461 0.0000000
## LL-INT  0.009379509 -0.6012550  0.6200140 0.9999992
## SS-INT -1.723285486 -2.3501474 -1.0964236 0.0000000
## FF-INT -1.863636364 -2.4338830 -1.2933897 0.0000000
## SS-LL  -1.732664996 -2.3664057 -1.0989243 0.0000000
## FF-LL  -1.873015873 -2.4508158 -1.2952160 0.0000000
## FF-SS  -0.140350877 -0.7352745  0.4545727 0.9654019
##
##   plot.labels labels   height
## 1         INT      a 4.280000
## 2          LL      a 3.946667
## 3          SS      b 1.280000
## 4          FF      b 0.280000
## 5          LS      a 3.280000
```




```
df_avg %>%
  ggplot(aes(treatment, total_aggression)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  geom_jitter(width=0.3, , height=0) +
  scale_size(name = "number of\ndisplays") +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  ylab("# aggressions") +
  ggtitle("average number of aggressions per video") +
  theme_clean()+
  geom_text(data = generate_label_df(x = treatment, y = total_aggression, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

```
## [1] "Settings:  unique SS "
## Component 1 :
##                              Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4   18.217    4.5542 5000 < 2.2e-16 ***
## Residuals                    106   52.996    0.5000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
```

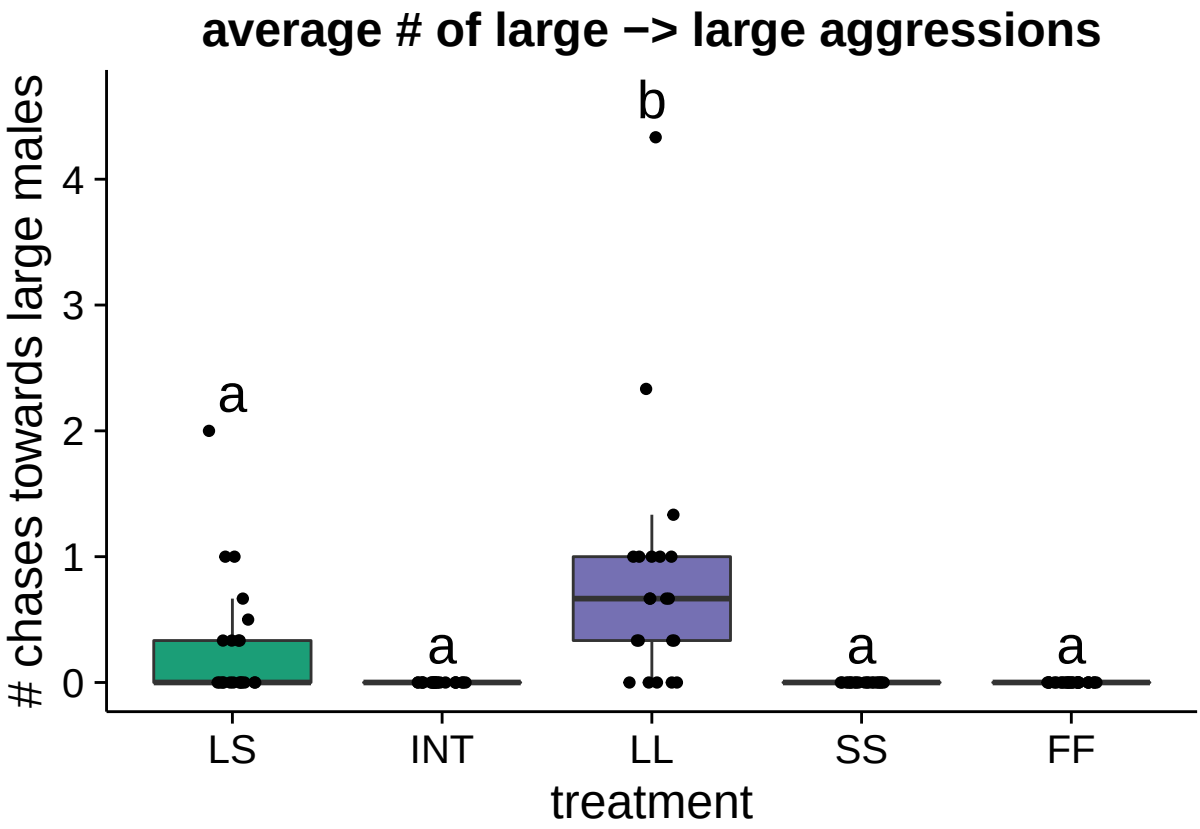
```
##              diff        lwr        upr     p adj
## INT-LS  0.7384560  0.1397805  1.3371315 0.0076917
## LL-LS   0.3174603 -0.2881365  0.9230572 0.5938714
## SS-LS  -0.4820384 -1.1033677  0.1392908 0.2058445
## FF-LS  -0.1408730 -0.7073570  0.4256110 0.9582205
## LL-INT -0.4209957 -1.0196712  0.1776799 0.2969979
## SS-INT -1.2204944 -1.8350796 -0.6059093 0.0000025
## FF-INT -0.8793290 -1.4384077 -0.3202503 0.0002824
## SS-LL  -0.7994987 -1.4208280 -0.1781695 0.0047757
## FF-LL  -0.4583333 -1.0248173  0.1081507 0.1713129
## FF-SS   0.3411654 -0.2421069  0.9244378 0.4859893
##
##   plot.labels labels   height
## 1         INT      a 4.233333
## 2          LL     ab 4.566667
## 3          SS      c 1.233333
## 4          FF     bc 2.233333
## 5          LS     bc 2.233333
```




```
df_avg %>%
  ggplot(aes(treatment, large_vs_large)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  geom_jitter(width=0.3, height=0) +
  scale_size(name = "number of\ndisplays") +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  ylab("# chases towards large males") +
  ggtitle("average # of large -> large aggressions") +
```

```
  theme_clean() +
  geom_text(data = generate_label_df(x = treatment, y = large_vs_large, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

```
## [1] "Settings:  unique SS "
## Component 1 :
##                              Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4   10.799   2.69965 5000 < 2.2e-16 ***
## Residuals                    106   24.587   0.23196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##                 diff         lwr         upr     p adj
## INT-LS -3.095238e-01 -0.71730249  0.09825488 0.2249778
## LL-LS   5.000000e-01  0.08750695  0.91249305 0.0092647
## SS-LS  -3.095238e-01 -0.73273275  0.11368513 0.2590932
## FF-LS  -3.095238e-01 -0.69537572  0.07632810 0.1781060
## LL-INT  8.095238e-01  0.40174512  1.21730249 0.0000025
## SS-INT -3.330669e-16 -0.41861527  0.41861527 1.0000000
## FF-INT -1.923065e-16 -0.38080788  0.38080788 1.0000000
## SS-LL  -8.095238e-01 -1.23273275 -0.38631487 0.0000060
## FF-LL  -8.095238e-01 -1.19537572 -0.42367190 0.0000006
## FF-SS   1.407604e-16 -0.39728704  0.39728704 1.0000000
##
##   plot.labels labels    height
## 1         INT      a 0.3033333
## 2          LL      b 4.6366667
## 3          SS      a 0.3033333
## 4          FF      a 0.3033333
## 5          LS      a 2.3033333
```



Here and below I'm going to try a flexible approach in terms of analyzing things statistically. I'm using a permutation (=non-parametric) linear model using `lmp` and looking for post-doc differences between treatments using `glht` from the `multcomp` package.

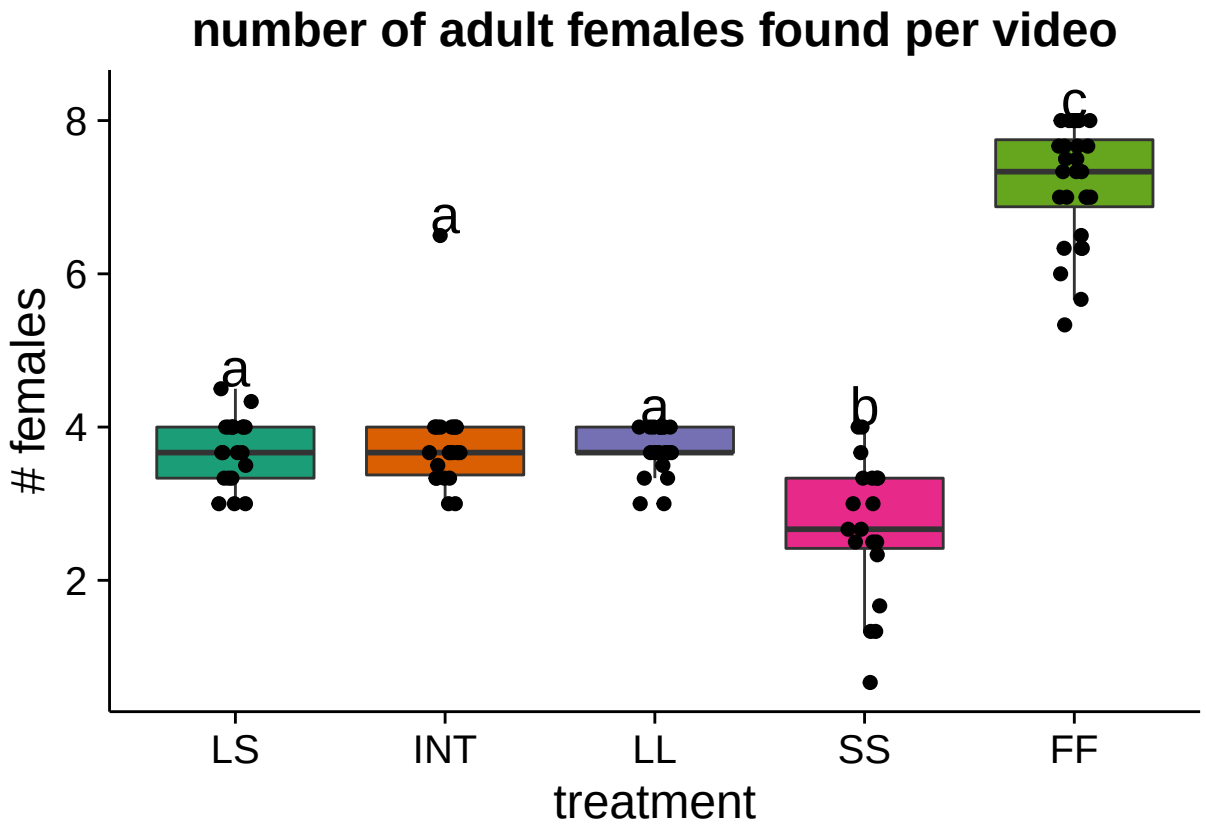
```
require(lmPerm)
require(multcomp)
```

```
## Loading required package: multcomp

## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
# df_avg %$%
# lmp(large_vs_large ~ treatment, data = .) %>%
#   glht(linfct=mcp(treatment="Tukey")) %>%
#   summary

df_avg %>%
  ggplot(aes(treatment, number_model_female)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  geom_jitter(width=0.2, size=2, height=0) +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  ylab("# females") +
  ggtitle("number of adult females found per video") +
  theme_clean() +
  scale_size_continuous(range = c(1,6)) +
  geom_text(data = generate_label_df(x = treatment, y = number_model_female, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

```
## [1] "Settings:  unique SS "
## Component 1 :
##                              Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4  300.599    75.150 5000 < 2.2e-16 ***
## Residuals                    106   46.824     0.442
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##                diff        lwr        upr     p adj
## INT-LS  0.073593074 -0.4891411  0.6363273 0.9962405
## LL-LS  -0.007936508 -0.5771765  0.5613035 0.9999995
## SS-LS  -1.021303258 -1.6053312 -0.4372753 0.0000408
## FF-LS   3.470238095  2.9377628  4.0027134 0.0000000
## LL-INT -0.081529582 -0.6442638  0.4812046 0.9944176
## SS-INT -1.094896332 -1.6725850 -0.5172076 0.0000074
## FF-INT  3.396645022  2.8711305  3.9221595 0.0000000
## SS-LL  -1.013366750 -1.5973947 -0.4293388 0.0000476
## FF-LL   3.478174603  2.9456993  4.0106499 0.0000000
## FF-SS   4.491541353  3.9432856  5.0397971 0.0000000
##
##   plot.labels labels height
## 1         INT      a   6.78
## 2          LL      a   4.28
## 3          SS      b   4.28
## 4          FF      c   8.28
## 5          LS      a   4.78
```



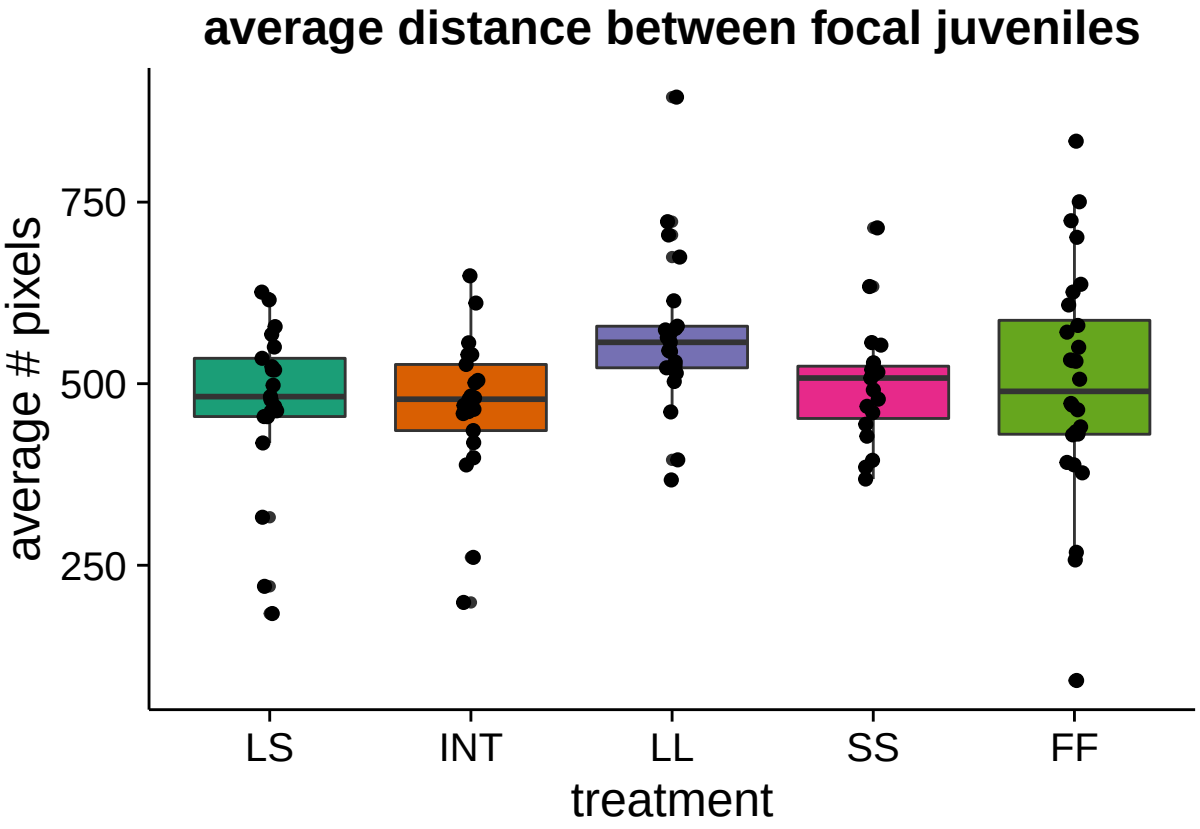

```
# df_avg %$%
#   lmp(number_model_female ~ treatment, data = .) %>%
#   glht(linfct=mcp(treatment="Tukey")) %>%
#   summary

require(viridis)
```

```
## Loading required package: viridis
```

```
df_avg %>%
  ggplot(aes(treatment, pairewise_distance_juvs, fill = treatment)) +
  geom_boxplot() +
  geom_jitter(size = 2, width = 0.1, height=0) +
  theme_clean() +
  ylab("average # pixels") +
  ggtitle("average distance between focal juveniles") +
  scale_fill_brewer(palette = "Dark2", guide= F)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

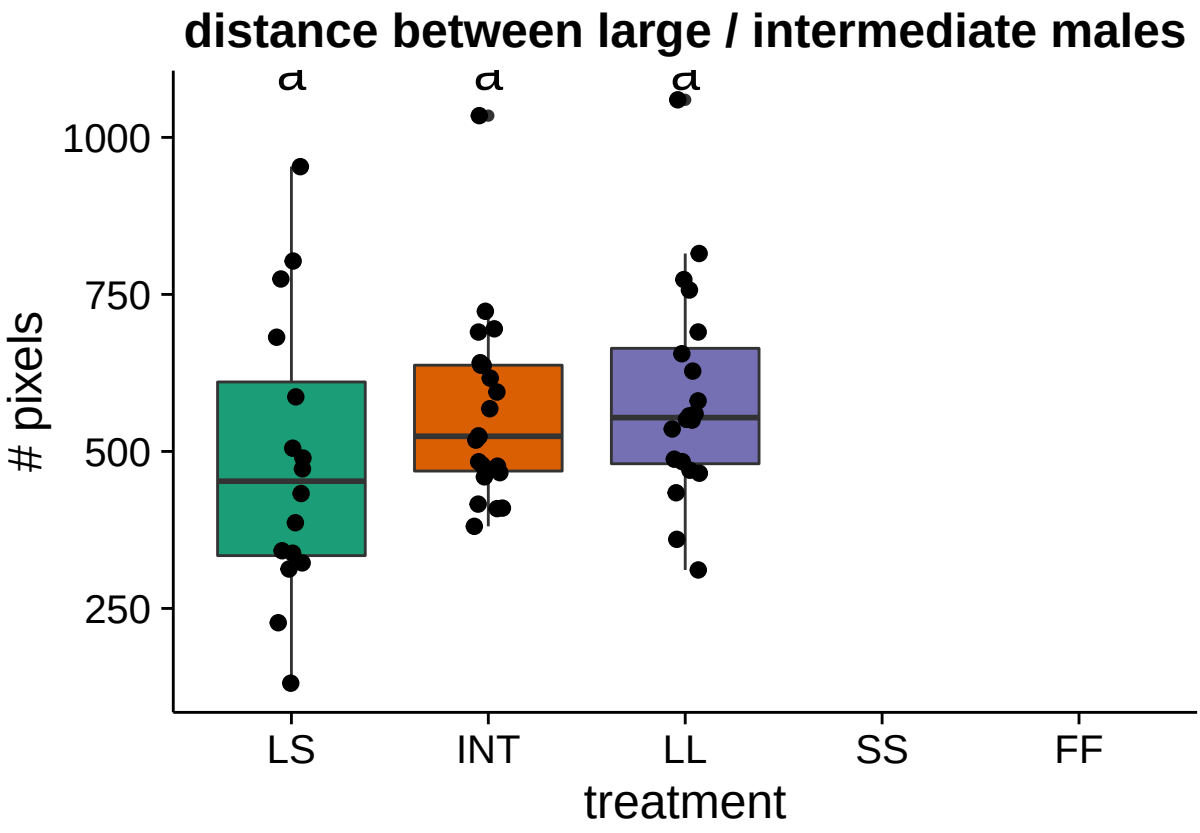
```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
# df_avg %$%
#   lmp(pairewise_distance_juvs ~ treatment, data = .) %>%
#   glht(linfct=mcp(treatment="Tukey")) %>%
#   summary

df_avg %>%
  group_by(treatment) %>%
  ggplot(aes(treatment, pairwise_distance_large_males)) +
  geom_boxplot(aes(fill=treatment)) +
  geom_jitter(width=0.2, size=2.4, height=0) +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  scale_colour_continuous(low="grey80", high="forestgreen") +
  ylab("# pixels") +
  ggtitle("distance between large / intermediate males") +
  theme_clean() +
  geom_text(data = generate_label_df(x = treatment, y = pairwise_distance_large_males, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

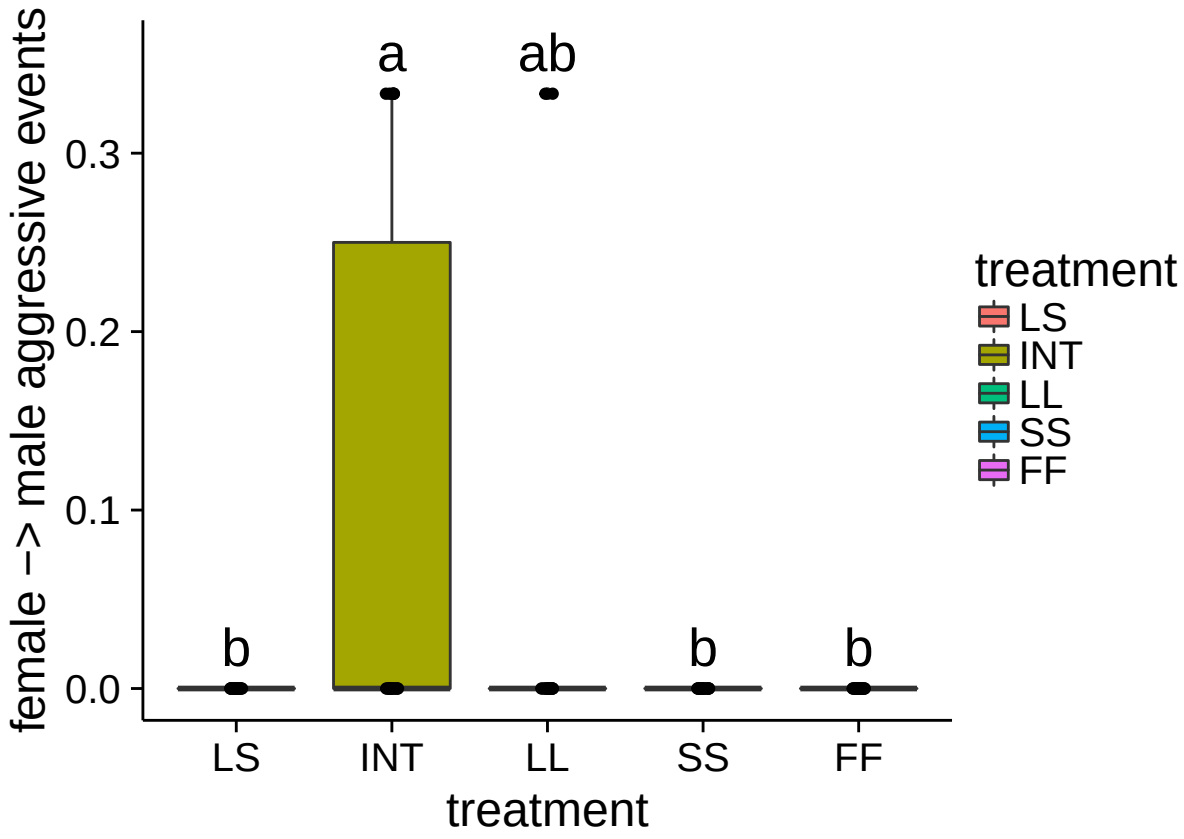
```
## [1] "Settings:  unique SS "
## Component 1 :
##                              Df R Sum Sq R Mean Sq Iter Pr(Prob)
## eval(arguments$x, dataframe)  2    97598     48799 1974   0.2958
## Residuals                    55  1770863     32198

## Warning in generate_label_df(x = treatment, y =
## pairwise_distance_large_males, : the ANOVA is not significant
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##             diff        lwr      upr     p adj
## INT-LS  77.92986  -64.08164 219.9414 0.3892882
## LL-LS  101.28706  -43.68333 246.2575 0.2208169
## LL-INT  23.35720 -110.17972 156.8941 0.9069618

## Warning in FUN(X[[i]], ...): no non-missing arguments to max; returning -
## Inf

## Warning in FUN(X[[i]], ...): no non-missing arguments to max; returning -
## Inf

##   plot.labels labels height
## 1         INT      a    Inf
## 2          LL      a    Inf
## 3          LS      a    Inf

## Warning: Removed 53 rows containing non-finite values (stat_boxplot).

## Warning: Removed 53 rows containing missing values (geom_point).
```



```r
# df_avg %$%
#   lmp(pairwise_distance_large_males ~ treatment, data = .) %>%
#   glht(linfct=mcp(treatment="Tukey")) %>%
#   summary
df_avg %>%
  ggplot(aes(y=female_vs_male, x=treatment)) +
    geom_boxplot(aes(fill = treatment)) +
    theme_clean() +
  geom_jitter(width =0.1, height=0) +
    xlab("treatment") +
    ylab("female -> male aggressive events") +
    geom_text(data = generate_label_df(x = treatment, y = female_vs_male, df_avg),
              aes(x = plot.labels, y = height, label = labels),
              size = 7)
```

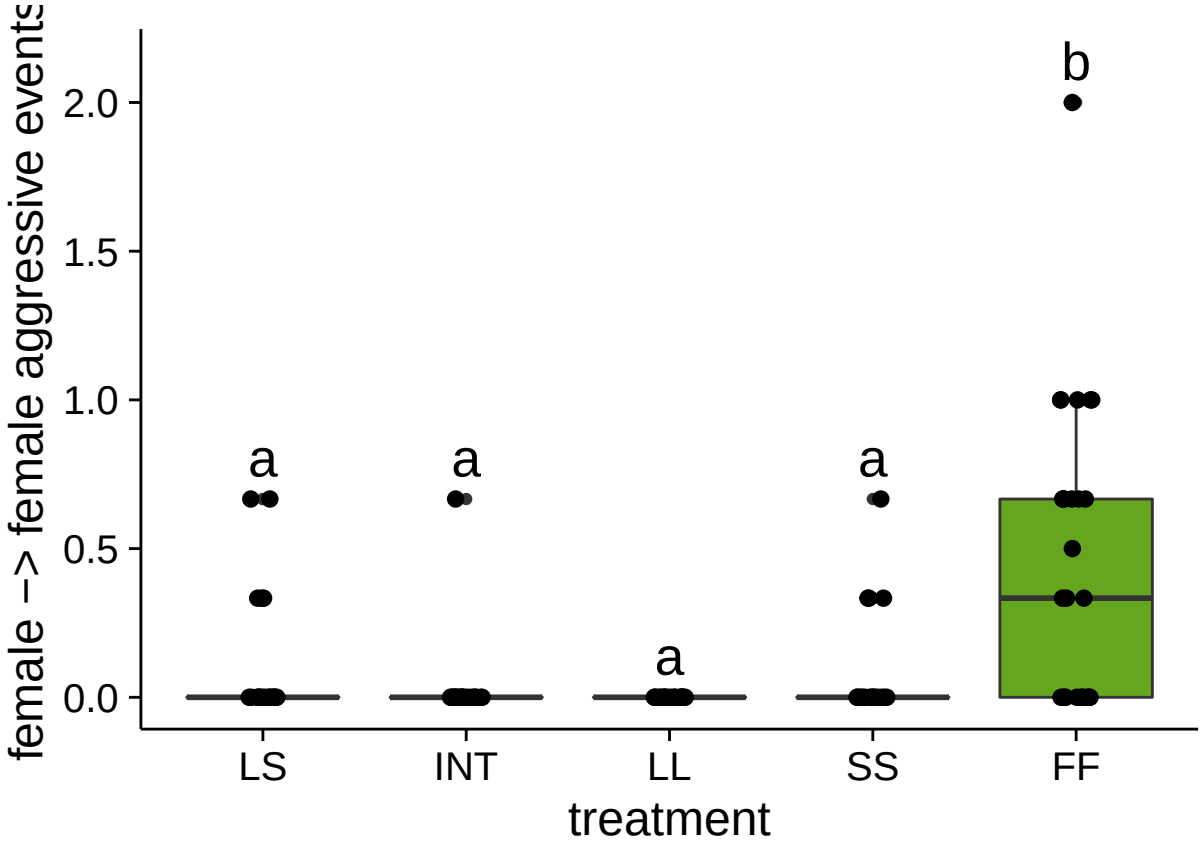
```
## [1] "Settings:  unique SS "
## Component 1 :
##                               Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4  0.13892  0.034730 5000 < 2.2e-16 ***
## Residuals                    106  0.68591  0.006471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##                  diff         lwr          upr     p adj
## INT-LS  9.090909e-02  0.02280055  0.159017636 0.0030637
## LL-LS   3.174603e-02 -0.03714992  0.100641985 0.7047381
## SS-LS  -7.285839e-17 -0.07068576  0.070685757 1.0000000
## FF-LS   3.134893e-17 -0.06444626  0.064446263 1.0000000
## LL-INT -5.916306e-02 -0.12727160  0.008945486 0.1204159
## SS-INT -9.090909e-02 -0.16082760 -0.020990583 0.0042226
## FF-INT -9.090909e-02 -0.15451288 -0.027305299 0.0012311
## SS-LL  -3.174603e-02 -0.10243179  0.038939725 0.7242106
## FF-LL  -3.174603e-02 -0.09619230  0.032700232 0.6499208
## FF-SS   1.042073e-16 -0.06635620  0.066356196 1.0000000
##
##   plot.labels labels     height
## 1         INT      a 0.35666667
## 2          LL     ab 0.35666667
## 3          SS      b 0.02333333
## 4          FF      b 0.02333333
## 5          LS      b 0.02333333
```




```
ggplot(df_avg, aes(y=female_vs_female, x=treatment)) +
  geom_boxplot(aes(fill = treatment)) +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  theme_clean()+
  geom_jitter(width=0.2, size=2.4, height=0) +
  xlab("treatment") +
  ylab("female -> female aggressive events") +
  geom_text(data = generate_label_df(x = treatment, y = female_vs_female, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

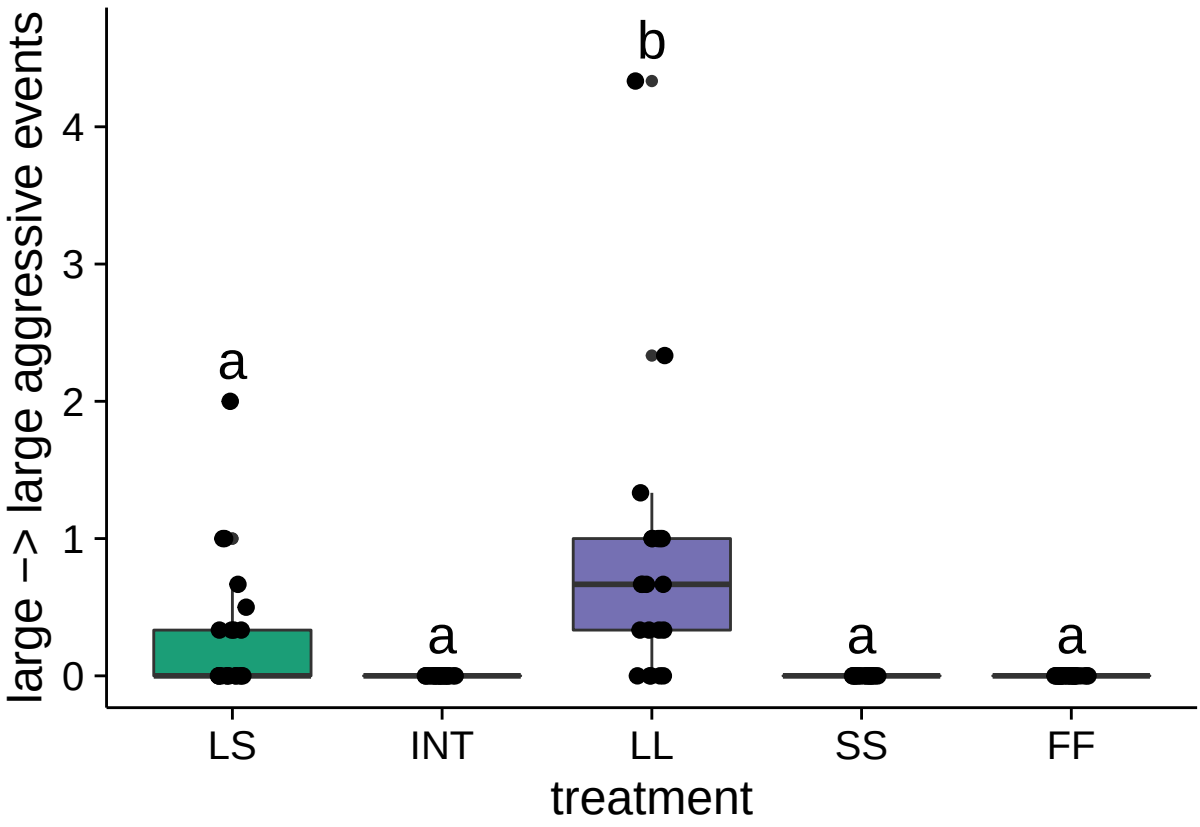
```
## [1] "Settings:  unique SS "
## Component 1 :
##                              Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4   3.3511   0.83778 5000 < 2.2e-16 ***
## Residuals                    106   8.6884   0.08197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##               diff        lwr       upr     p adj
## INT-LS -0.08080808 -0.3232120 0.1615959 0.8865682
```

```
## LL-LS  -0.11111111 -0.3563175 0.1340953 0.7175499
## SS-LS  -0.02339181 -0.2749683 0.2281846 0.9990088
## FF-LS   0.33531746  0.1059479 0.5646870 0.0008891
## LL-INT -0.03030303 -0.2727070 0.2121009 0.9968440
## SS-INT  0.05741627 -0.1914295 0.3062620 0.9680674
## FF-INT  0.41612554  0.1897544 0.6424967 0.0000145
## SS-LL   0.08771930 -0.1638571 0.3392957 0.8690419
## FF-LL   0.44642857  0.2170590 0.6757981 0.0000040
## FF-SS   0.35870927  0.1225421 0.5948765 0.0004970
##
##   plot.labels labels    height
## 1         INT      a 0.8066667
## 2          LL      a 0.1400000
## 3          SS      a 0.8066667
## 4          FF      b 2.1400000
## 5          LS      a 0.8066667
```




```
ggplot(df_avg, aes(y=large_vs_large, x=treatment)) +
  geom_boxplot(aes(fill = treatment)) +
  theme_clean() +
  geom_jitter(width=0.2, size=2.4, height=0) +
  xlab("treatment") +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  ylab("large -> large aggressive events") +
  geom_text(data = generate_label_df(x = treatment, y = large_vs_large, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

```
## [1] "Settings:  unique SS "
## Component 1 :
##                               Df R Sum Sq R Mean Sq Iter  Pr(Prob)
## eval(arguments$x, dataframe)   4   10.799   2.69965 5000 < 2.2e-16 ***
## Residuals                    106   24.587   0.23196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##                 diff         lwr         upr     p adj
## INT-LS -3.095238e-01 -0.71730249  0.09825488 0.2249778
## LL-LS   5.000000e-01  0.08750695  0.91249305 0.0092647
## SS-LS  -3.095238e-01 -0.73273275  0.11368513 0.2590932
## FF-LS  -3.095238e-01 -0.69537572  0.07632810 0.1781060
## LL-INT  8.095238e-01  0.40174512  1.21730249 0.0000025
## SS-INT -3.330669e-16 -0.41861527  0.41861527 1.0000000
## FF-INT -1.923065e-16 -0.38080788  0.38080788 1.0000000
## SS-LL  -8.095238e-01 -1.23273275 -0.38631487 0.0000060
## FF-LL  -8.095238e-01 -1.19537572 -0.42367190 0.0000006
## FF-SS   1.407604e-16 -0.39728704  0.39728704 1.0000000
##
##   plot.labels labels    height
## 1         INT      a 0.3033333
## 2          LL      b 4.6366667
## 3          SS      a 0.3033333
## 4          FF      a 0.3033333
## 5          LS      a 2.3033333
```




```
ggplot(df_avg, aes(total_courtship, total_aggression)) +
  geom_point() +
  facet_wrap(~ treatment) +
  theme_minimal() +
  ggtitle("relationship between courtship and aggression by treatment") +
  xlab("total # courtship events") +
  ylab("total # aggressive events") +
  geom_smooth(se=F)
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 2.5e-05
```

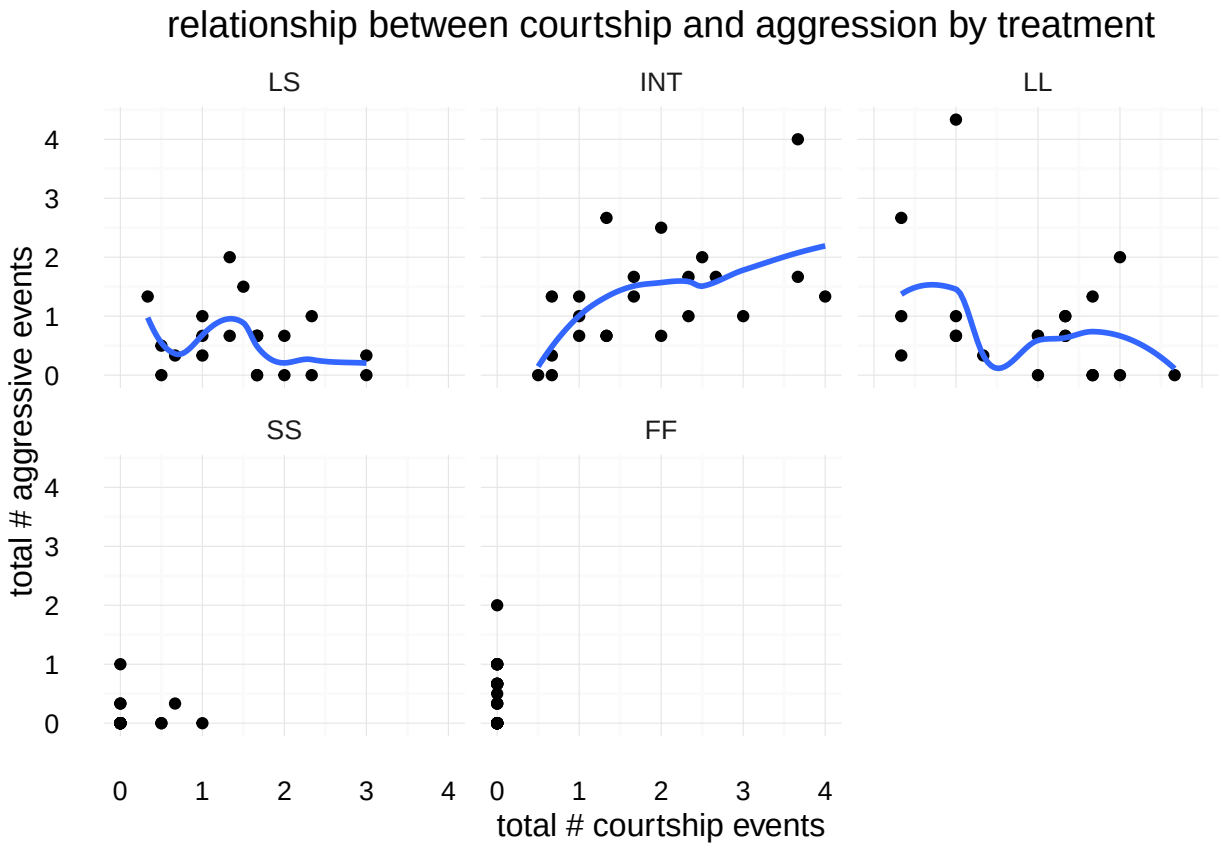
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.25

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)
```




```
df_avg %$%
lmp(total_aggression ~ total_courtship * treatment, data = .)  %>%
anova
```

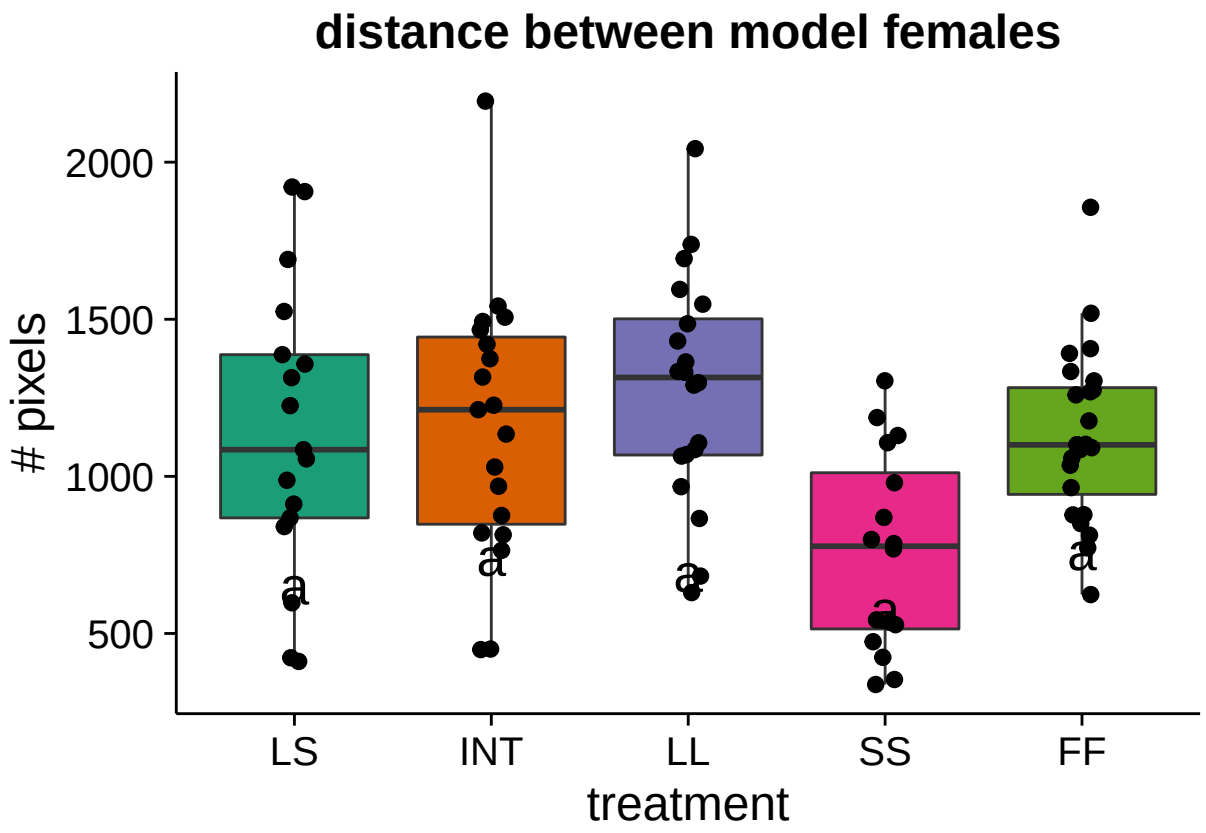
```
## [1] "Settings:  unique SS : numeric variables centered"
## Analysis of Variance Table
##
## Response: total_aggression
##                            Df R Sum Sq R Mean Sq Iter Pr(Prob)
## total_courtship             1    0.002   0.00218   51   1.0000
## treatment                   4    7.745   1.93616 5000   0.0012 **
## total_courtship:treatment   3    8.447   2.81582 5000   <2e-16 ***
## Residuals                 102   44.488   0.43616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
df %>%
  ggplot(aes(treatment, pairwise_distance_females)) +
  geom_boxplot(aes(fill=treatment), outlier.shape = NA) +
  geom_jitter(width=0.2, size=2.4, height=0) +
  scale_colour_continuous(low="grey80", high="forestgreen") +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  ylab("# pixels") +
  ggtitle("distance between model females") +
  theme_clean() +
  geom_text(data = generate_label_df(x = treatment, y = pairwise_distance_females, df_avg),
            aes(x = plot.labels, y = height, label = labels),
            size = 7)
```

```
## [1] "Settings:  unique SS "
## Component 1 :
##                               Df R Sum Sq R Mean Sq Iter Pr(Prob)
## eval(arguments$x, dataframe)   4    48107     12027  356   0.6124
## Residuals                    106  1862400     17570
```

```
## Warning in generate_label_df(x = treatment, y =
## pairwise_distance_females, : the ANOVA is not significant
```

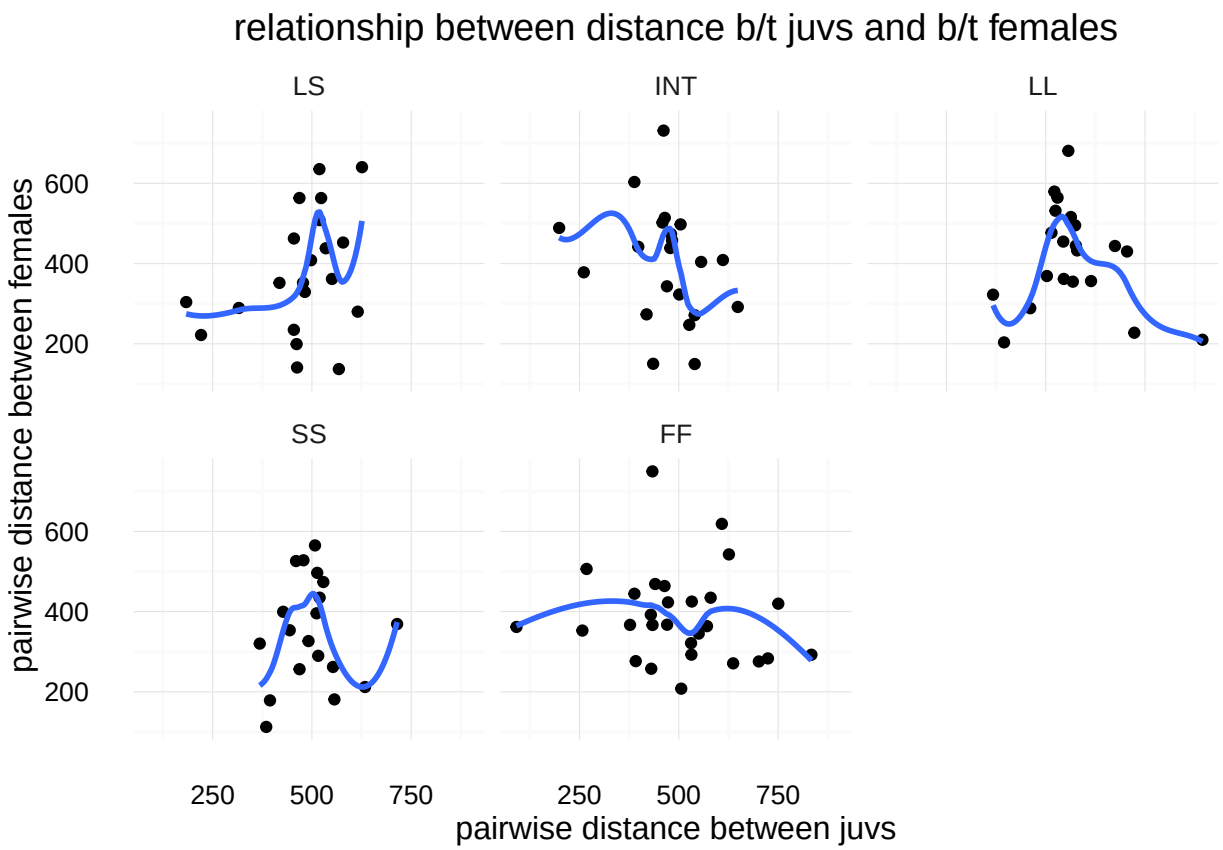
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aovp(formula = eval(arguments$y, dataframe) ~ eval(arguments$x, dataframe), data = dataframe)
##
## $`eval(arguments$x, dataframe)`
##              diff        lwr       upr     p adj
## INT-LS  23.793289  -88.43583 136.02241 0.9765466
## LL-LS   41.465497  -72.06111 154.99211 0.8485707
## SS-LS  -23.298533 -139.77438  93.17731 0.9810787
## FF-LS   14.059096  -92.13532 120.25352 0.9960578
## LL-INT  17.672207  -94.55691 129.90133 0.9923186
## SS-INT -47.091822 -162.30339  68.11975 0.7879630
## FF-INT  -9.734194 -114.54039  95.07200 0.9990131
## SS-LL  -64.764029 -181.23987  51.71181 0.5370311
## FF-LL  -27.406401 -133.60082  78.78802 0.9523553
## FF-SS   37.357628  -71.98397 146.69923 0.8772425
##
##   plot.labels labels   height
## 1         INT      a 744.3441
## 2          LL      a 693.9008
## 3          SS      a 577.9425
## 4          FF      a 762.4774
## 5          LS      a 653.2074
```



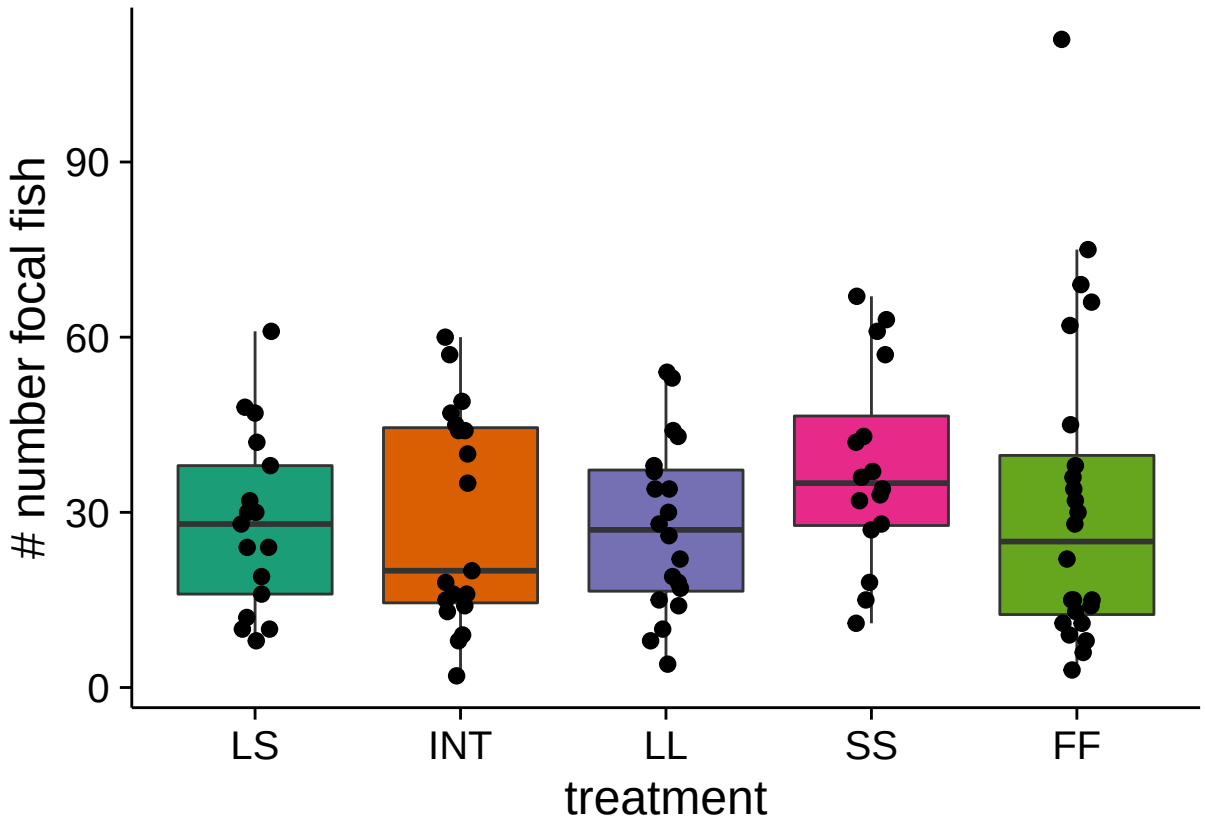

```
  # df_avg %$%
  # lmp(pairwise_distance_females ~ treatment, data = .) %>%
  # glht(linfct=mcp(treatment="Tukey")) %>%
  # summary

ggplot(df_avg, aes(pairewise_distance_juvs, pairwise_distance_females)) +
  geom_point() +
  facet_wrap(~ treatment) +
  theme_minimal() +
  ggtitle("relationship between distance b/t juvs and b/t females") +
  xlab("pairwise distance between juvs") +
  ylab("pairwise distance between females") +
  geom_smooth(se=F)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
df %>%
  ggplot(aes(treatment, number_focal)) +
  geom_boxplot(aes(fill=treatment), outlier.shape=NA) +
  geom_jitter(width=0.2, size=2.4, height=0) +
  scale_fill_brewer(palette = "Dark2", guide= F) +
  ylab("# number focal fish") +
  theme_clean()
```




```
ggplot(df_avg, aes(total_aggression, pairewise_distance_juvs)) +
  geom_point() +
  facet_wrap(~ treatment) +
  theme_clean() +
  xlab("total aggression") +
  ylab("pairwise distance between juvs") +
  geom_smooth(se=F)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.33333
```

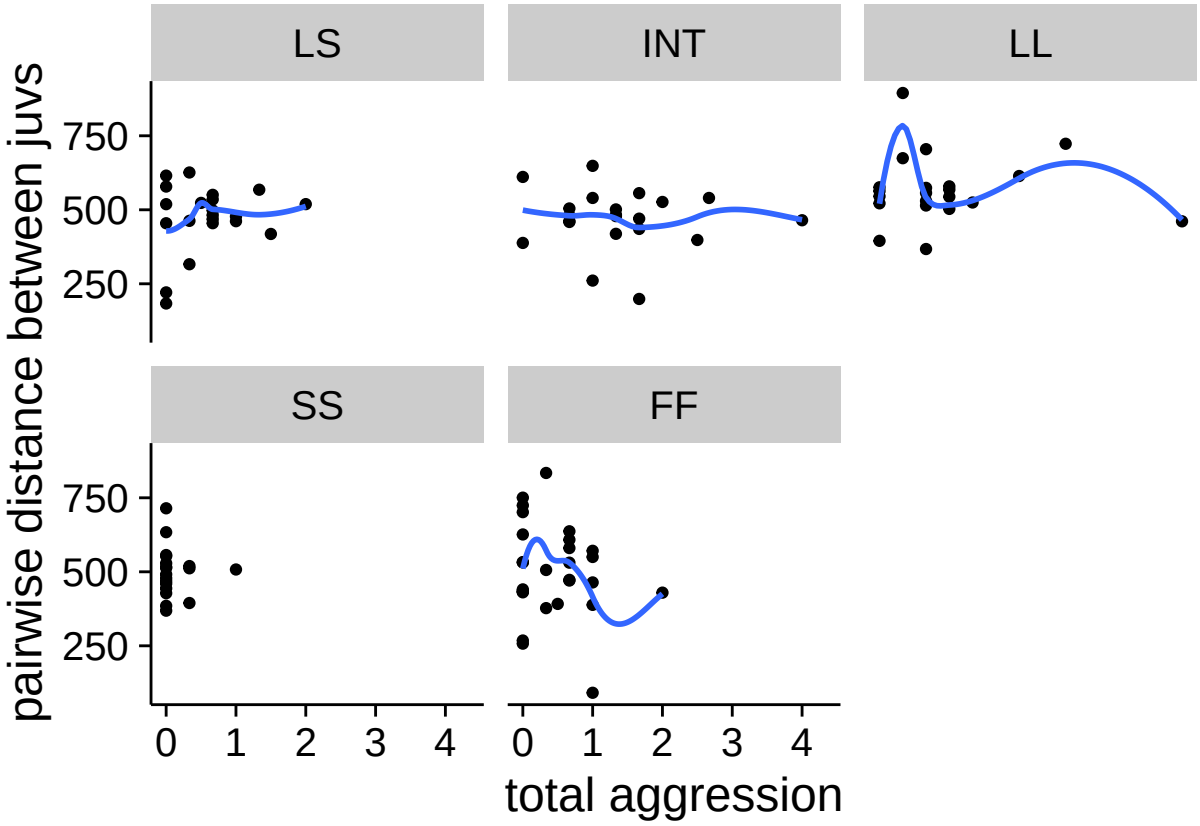
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.33333
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.1019e-16
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 2.5e-05
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.33333

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.33333

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.1019e-16

## Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(df_avg, aes(male_chased_juvenile, pairewise_distance_juvs)) +
  geom_point() +
  facet_wrap(~ treatment) +
  theme_clean() +
  xlab("male chased juvenile") +
  ylab("pairwise distance between juvs") +
  geom_smooth(se=F)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.0066667
```

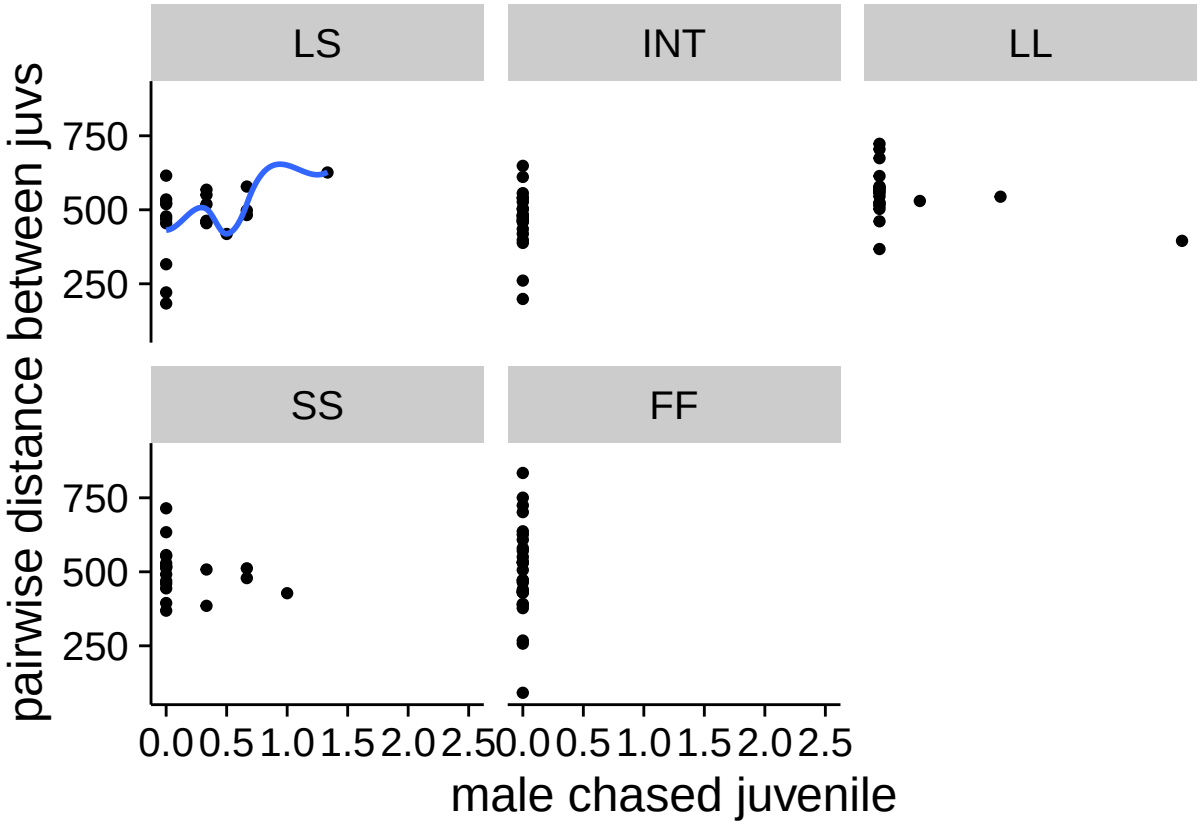
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.34
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.11111
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at -0.0125
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 0.00015625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.0125

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.0125

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at -0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 2.5e-05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.11111

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)

## Warning: Removed 2 rows containing missing values (geom_point).
```




```
ggplot(df_avg, aes(total_aggression, total_fish)) +
  geom_point() +
  theme_clean() +
  xlab("total # aggressive events") +
  ylab("total # of fish") +
  geom_smooth(se=F) +
  facet_wrap(~ treatment)
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.33333

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.33333

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.1019e-16

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at -0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 2.5e-05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.33333

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.33333

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.1019e-16
```