

# rupertreach

A. Spence

2/22/2022

## Rupert reaching data for PA grant

Let's recreate one of our python box plots to show we've loaded the data.

Testlink <http://spencelab.com>.

### Get started

```
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
```

```
library(tidyr)  
library(readr)  
library(ggplot2)  
library(ggpubr)  
library(car)
```

```
## Loading required package: carData  
##  
## Attaching package: 'car'  
## The following object is masked from 'package:dplyr':  
##  
##   recode
```

```
library(Hmisc)
```

```
## Loading required package: lattice  
## Loading required package: survival  
## Loading required package: Formula  
##  
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##   src, summarize
```

```
## The following objects are masked from 'package:base':
##
##   format.pval, units
```

```
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:stats':
##
##   filter
```

```
library(emmeans)
library(nlme)
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##   collapse
```

```
library(stringr)
library(ggforce)
```

```
theme_update(plot.title = element_text(hjust = 0.5))
# maybe the plain BW background looks better than the grey stuff
# https://www.datanovia.com/en/lessons/combine-multiple-ggplots-into-a-figure/
#
theme_set(theme_bw())
#+
#   theme(legend.position = "top")
# )
setwd("/Users/aspence/rupertreaching/newdata/")
dfx <- read.csv("alldata_x.csv")
dfy <- read.csv("alldata_y.csv")
```

Prepare data, aggregate and stats

```
dfx$treatment <- factor(dfx$treatment)
dfx$treatment <- relevel(dfx$treatment, 'excite')
dfxt <- dfx %>% filter(time==0.4)
dfxt %>% group_by(treatment) %>%
  summarise(
    count = n(),
    mean = mean(x, na.rm = TRUE),
    sd = sd(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    IQR = IQR(x, na.rm = TRUE)
  )
```

```
## # A tibble: 4 x 6
##   treatment count    mean    sd median    IQR
```

```
##   <fct>      <int> <dbl> <dbl> <dbl> <dbl>
## 1 excite      13  10.7  5.63  11.8  7.04
## 2 control     11  -3.75  5.06 -3.56  4.11
## 3 inhib       7 -11.7  6.41 -11.7  6.30
## 4 norm        4   9.83  2.87  11.1  2.04
```

*# Plot it:*

*# choose comparisons to plot*

*# <http://www.sthda.com/english/articles/32-r-graphics-essentials/132-plot-grouped-data-box-plot-bar-plot>*

Let's get some sample sizes for the caption:

```
unique(dfxt$rat)
```

```
## [1] "ihr5"      "er1"      "er3"      "er6"      "ihrGS2"   "DDinhib"  "JRnf1"
## [8] "JRnf2"     "VVme"     "MGc2"     "MGc1"     "MGc3"
```

*# num unique rats:*

```
length(unique(dfxt$rat))
```

```
## [1] 12
```

*# num reaches per rat*

```
xtabs(~rat,data=dfxt)
```

```
## rat
## DDinhib      er1      er3      er6      ihr5  ihrGS2  JRnf1  JRnf2  MGc1  MGc2
##          3       7       4       7       1       3       1       3       2       1
##      MGc3  VVme
##          1       2
```

*# num reaches by treatment type*

```
xtabs(~treatment,data=dfxt)
```

```
## treatment
##  excite control  inhib  norm
##      13      11       7     4
```

*# sample size aggregated per rat vs treatment:*

```
dfxta <- dfxt %>% group_by(rat) %>%
  summarise(
    treatment = first(treatment),
    count = n(),
    mean = mean(x, na.rm = TRUE),
    sd = sd(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    IQR = IQR(x, na.rm = TRUE)
  )
```

*# sample size after aggregating by rat:*

```
xtabs(~treatment,dfxta)
```

```
## treatment
##  excite control  inhib  norm
##      3       4       3     2
```

Do the 4 way analysis

```
my_comps <- list( c("control","inhib"),c("excite","control"),
                  c("excite","inhib"), c("control","norm"), c("excite","norm"), c("norm","inhib") )
dfxt %>% ggplot(aes(x = treatment, y = x,color = treatment))+
```

```

#geom_boxplot(aes(color = treatment), show.legend = FALSE)+
geom_violin(trim = FALSE,show.legend=FALSE) +
geom_dotplot(
  binaxis='y', stackdir='center',
  color = "black", fill = "#999999",
) +
stat_summary(
  fun.data="mean_sdl", fun.args = list(mult=1),
  geom = "pointrange", color = "#FC4E07", size = 0.4
)+
stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07"
#geom_sina(aes(color = treatment), show.legend = FALSE)+
ylab("Max x") +
xlab("Treatment") +
coord_cartesian(ylim = c(-27, 55)) +
scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140", "#000000"))+
ggtitle("Wrist Max x")+
stat_compare_means(comparisons=my_comps, label="p.signif") # + # default is Wilcoxon

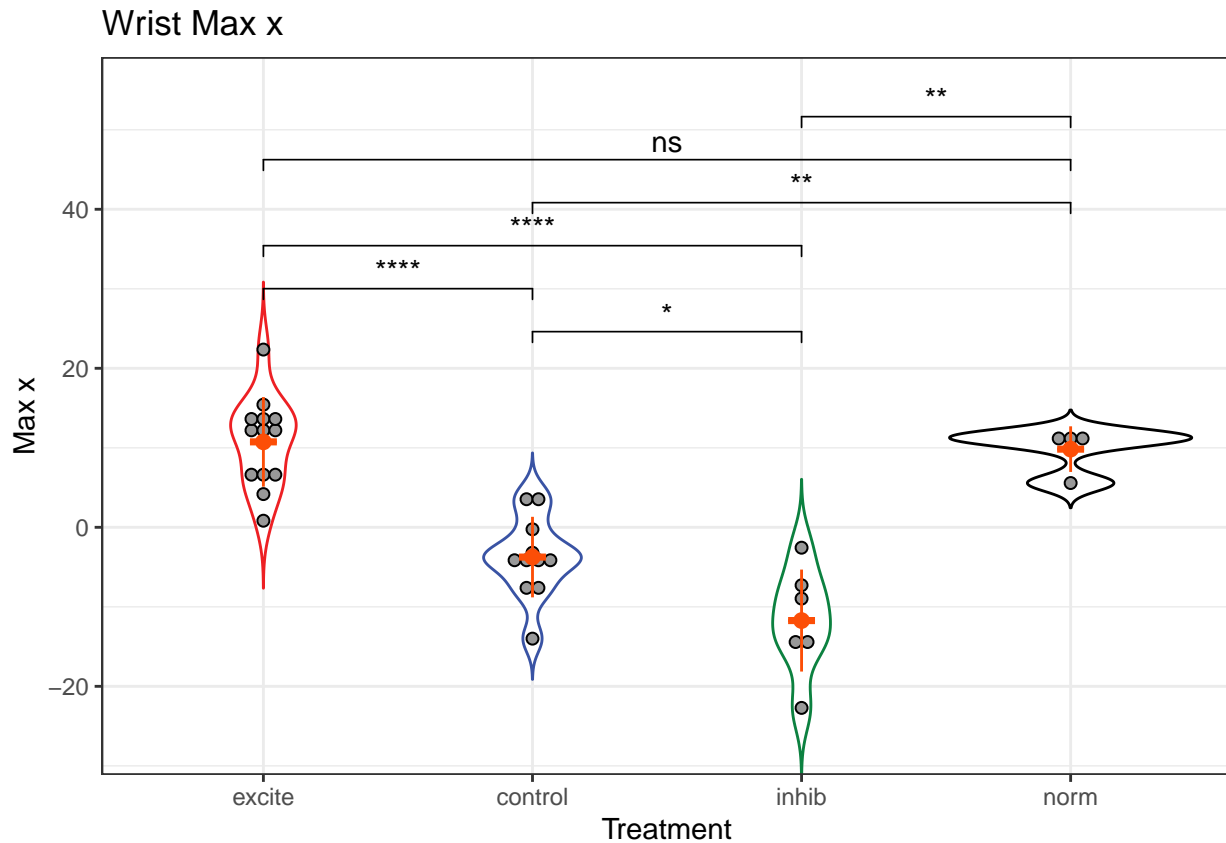
```

## Warning: The `fun.y` argument of `stat\_summary()` is deprecated as of ggplot2 3.3.0.  
## i Please use the `fun` argument instead.

## Warning: The `fun.ymin` argument of `stat\_summary()` is deprecated as of ggplot2 3.3.0.  
## i Please use the `fun.min` argument instead.

## Warning: The `fun.ymax` argument of `stat\_summary()` is deprecated as of ggplot2 3.3.0.  
## i Please use the `fun.max` argument instead.

## Bin width defaults to 1/30 of the range of the data. Pick better value with  
## `binwidth`.



```
# take out the global comparison, because it's implied that significant
# since we are doing post hoc tests
#stat_compare_means(aes(group = treatment), label = "p.signif", label.y=30,
#                      method='kruskal.test' ) # actually default is kruskal
```

```
# stat_compare_means(aes(group = treatment),
#                      method='kruskal.test' )
```

```
ggsave('xbox.pdf',width=2.5,height=3,units='in')
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```

```
kruskal.test(x ~ treatment, data = dfxt)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: x by treatment
## Kruskal-Wallis chi-squared = 26.428, df = 3, p-value = 7.758e-06
```

```
pairwise.wilcox.test(dfxt$x, dfxt$treatment,
                     p.adjust.method = "BH")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: dfxt$x and dfxt$treatment
##
```

```
##           excite  control  inhib
## control 3.4e-05 -          -
## inhib   7.7e-05 0.0185  -
## norm    0.3504 0.0029 0.0091
##
## P value adjustment method: BH
```

```
dfy$treatment <- factor(dfy$treatment)
dfy$treatment <- relevel(dfy$treatment, 'excite')
dfyt <- dfy %>% filter(time==0.4)
dfyt %>% group_by(treatment) %>%
  summarise(
    count = n(),
    mean = mean(y, na.rm = TRUE),
    sd = sd(y, na.rm = TRUE),
    median = median(y, na.rm = TRUE),
    IQR = IQR(y, na.rm = TRUE)
  )
```

```
## # A tibble: 4 x 6
##   treatment count  mean    sd median  IQR
##   <fct>      <int> <dbl> <dbl> <dbl> <dbl>
## 1 excite        13 20.5   8.21  19.8  12.7
## 2 control        11 -6.18  8.71  -6.13 10.1
## 3 inhib         7  -3.23 9.95  -3.26 14.3
## 4 norm          4  5.04  3.71   3.85  2.70
```

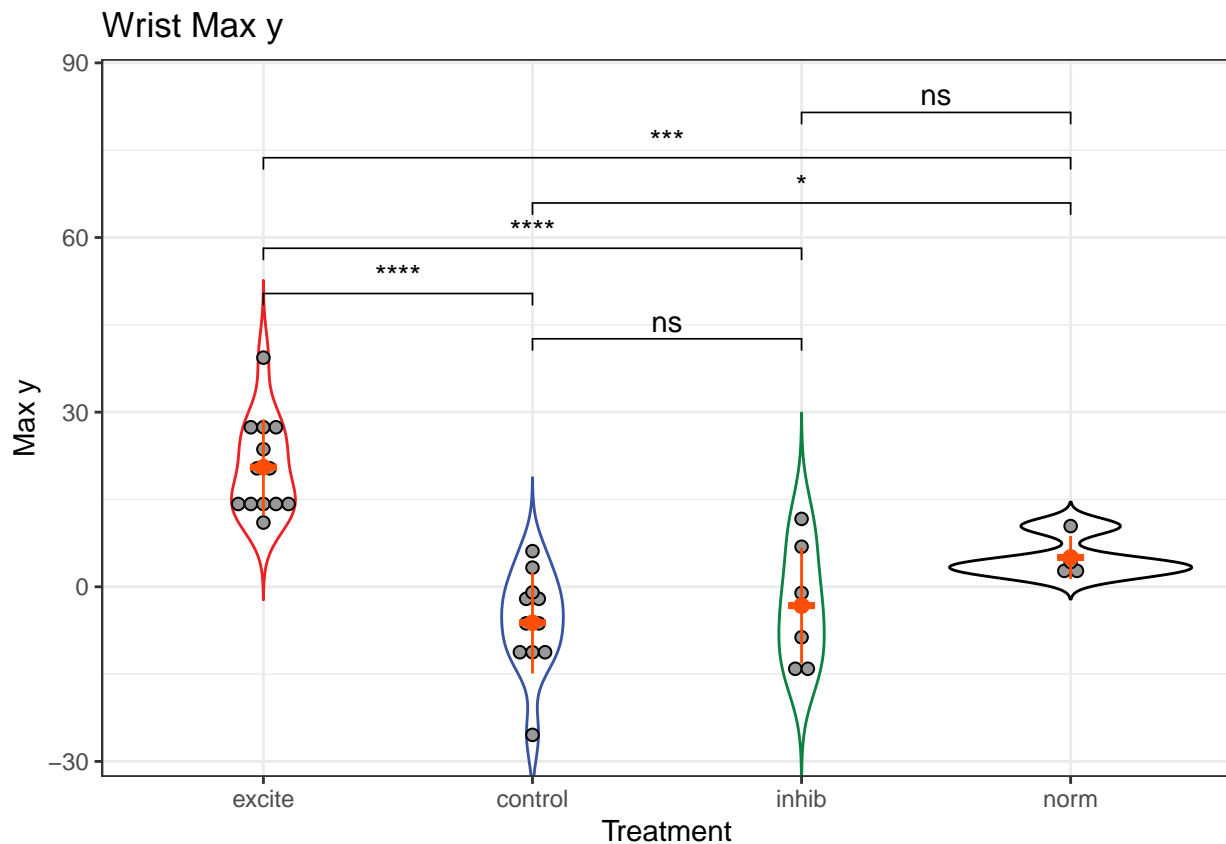
```
# let's try to add the overbars and stars
# Statistical test
stat.test <- dfyt %>%
  kruskal.test(x=.$y, g=.$treatment) %>%
  add_significance()
stat.test
```

```
##
## Kruskal-Wallis rank sum test
##
## data:  .$y and  .$treatment
## Kruskal-Wallis chi-squared = 25.369, df = 3, p-value = 1.292e-05
```

```
#stat.test <- tidy(stat.test %>% add_xy_position(x = "treatment"))
dfyt %>% ggplot(aes(x = treatment, y = y, color=treatment))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE, show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")
ylab("Max y") +
  xlab("Treatment") +
  coord_cartesian(ylim = c(-27, 85)) +
```

```
scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140", "#000000")) +
ggtitle("Wrist Max y") +
stat_compare_means(comparisons=my_comps, label="p.signif") # + # default is Wilcoxon
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('ybox.pdf',width=2.5,height=3,units='in')
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```

```
kruskal.test(y ~ treatment, data = dfyt)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: y by treatment
## Kruskal-Wallis chi-squared = 25.369, df = 3, p-value = 1.292e-05
```

```
pairwise.wilcox.test(dfyt$y, dfyt$treatment,
p.adjust.method = "BH")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: dfyt$y and dfyt$treatment
```

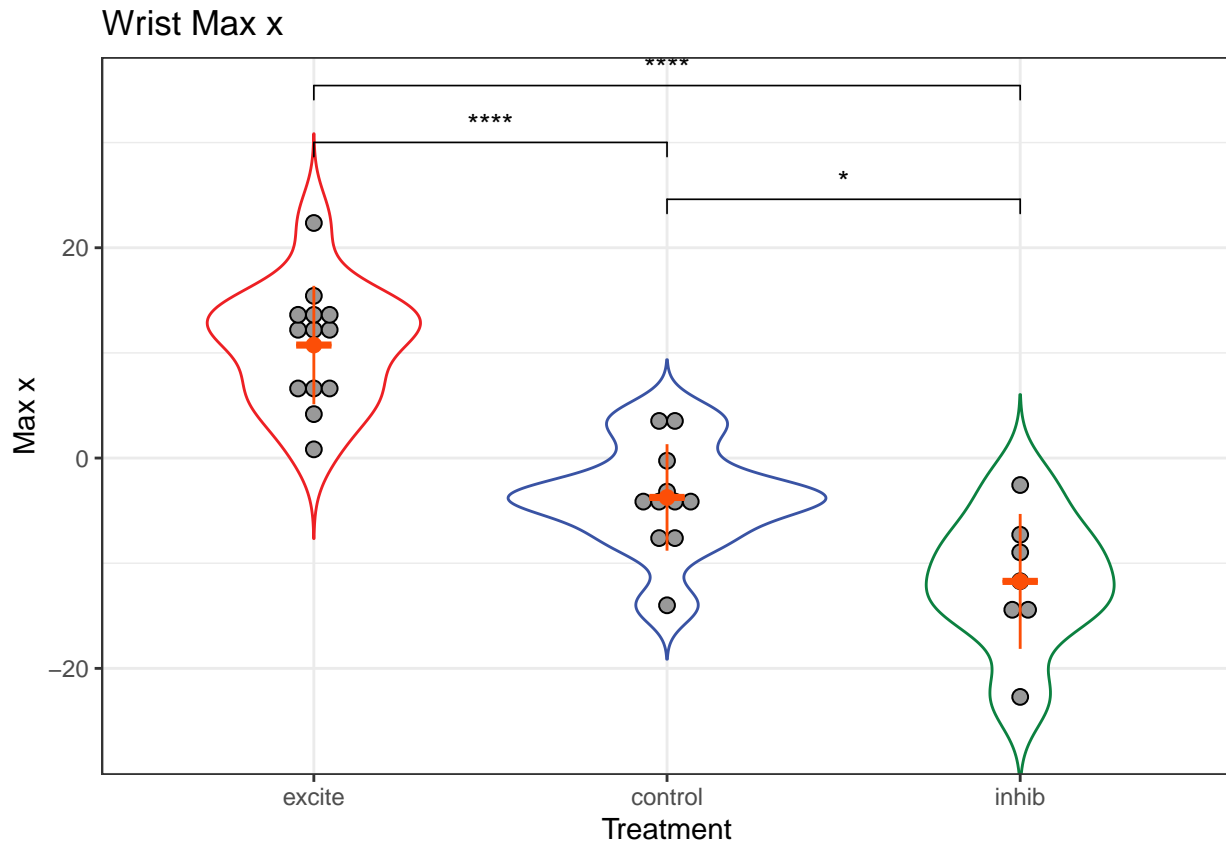
```
##
##          excite  control  inhib
## control 4.8e-06 -          -
## inhib   0.00015 0.86011 -
## norm    0.00168 0.02637 0.27636
##
## P value adjustment method: BH
```

Ok this code can be used with dropped norm level. TODO: Add that drop...

```
# TODO: DROP THE TREATMENT LEVEL NORM AND LET THIS PROCEED...
# Let's keep this code as is if we want to drop the norm level and proceed...
my_comps <- list( c("control","inhib"),c("excite","control"),
                  c("excite","inhib") )
dfxt %>% filter(treatment != "norm") %>% ggplot(aes(x = treatment, y = x,color = treatment))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07"
  #geom_sina(aes(color = treatment), show.legend = FALSE)+
  ylab("Max x") +
  xlab("Treatment") +
  coord_cartesian(ylim = c(-27, 35)) +
  scale_color_manual(values = c("#ED2024","#3953A4","#0C8140"))+
  ggtitle("Wrist Max x")+
  stat_compare_means(comparisons=my_comps, label="p.signif") # + # default is Wilcoxon
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```





```
# take out the global comparison, because it's implied that significant
# since we are doing post hoc tests
#stat_compare_means(aes(group = treatment), label = "p.signif", label.y=30,
#                    method='kruskal.test' ) # actually default is kruskal

# stat_compare_means(aes(group = treatment),
#                    method='kruskal.test' )

ggsave('xbox.pdf',width=2.5,height=3,units='in')

## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.

kruskal.test(x ~ treatment, data = dfxt)

##
## Kruskal-Wallis rank sum test
##
## data: x by treatment
## Kruskal-Wallis chi-squared = 26.428, df = 3, p-value = 7.758e-06

pairwise.wilcox.test(dfxt$x, dfxt$treatment,
                     p.adjust.method = "BH")

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: dfxt$x and dfxt$treatment
##
```

```
##           excite  control  inhib
## control 3.4e-05 -          -
## inhib   7.7e-05 0.0185 -
## norm    0.3504 0.0029 0.0091
##
## P value adjustment method: BH
```

y coord box plot

```
dfy$treatment <- factor(dfy$treatment)
dfy$treatment <- relevel(dfy$treatment, 'excite')
dfyt <- dfy %>% filter(time==0.4)
dfyt %>% group_by(treatment) %>%
  summarise(
    count = n(),
    mean = mean(y, na.rm = TRUE),
    sd = sd(y, na.rm = TRUE),
    median = median(y, na.rm = TRUE),
    IQR = IQR(y, na.rm = TRUE)
  )
```

```
## # A tibble: 4 x 6
##   treatment count  mean    sd median  IQR
##   <fct>      <int> <dbl> <dbl> <dbl> <dbl>
## 1 excite      13 20.5   8.21  19.8  12.7
## 2 control     11 -6.18  8.71  -6.13 10.1
## 3 inhib        7 -3.23  9.95  -3.26 14.3
## 4 norm         4  5.04  3.71   3.85  2.70
```

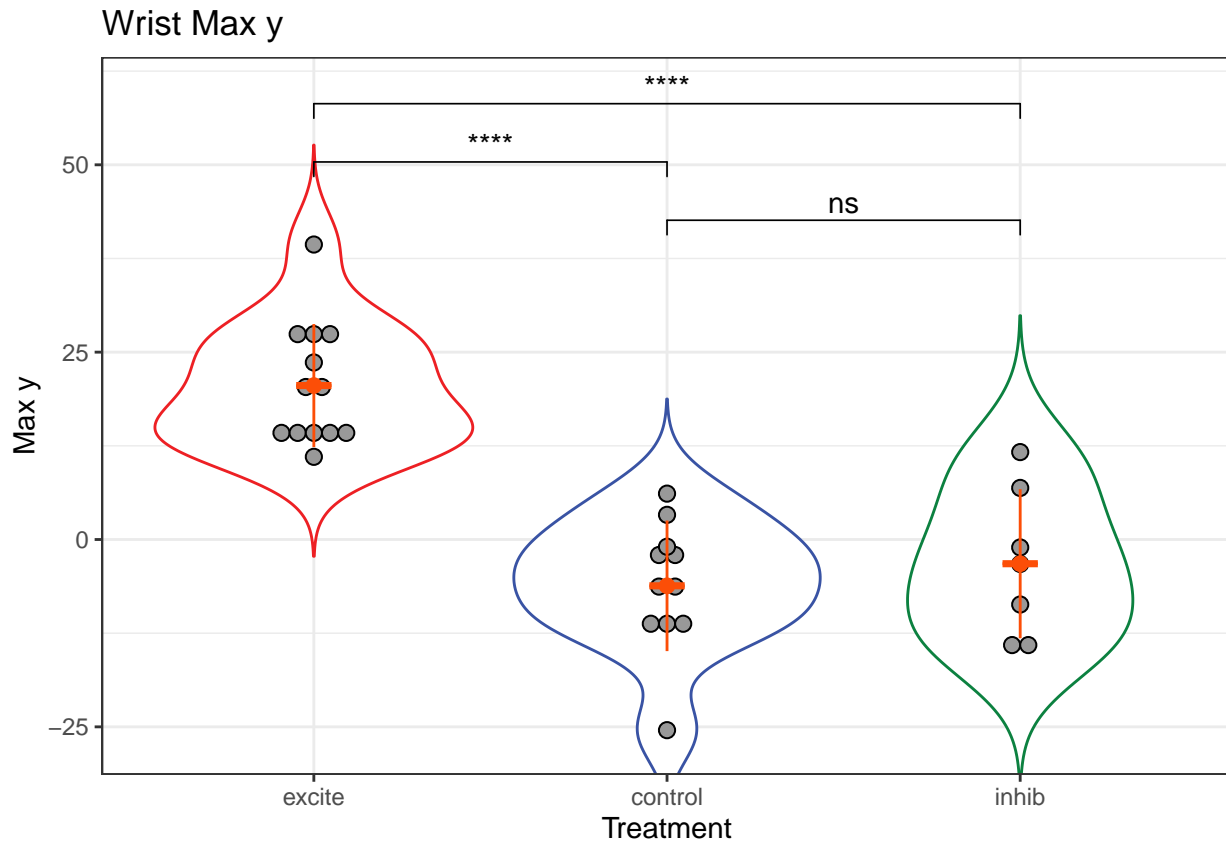
```
# let's try to add the overbars and stars
# Statistical test
stat.test <- dfyt %>%
  kruskal.test(x=.$y, g=.$treatment) %>%
  add_significance()
stat.test
```

```
##
## Kruskal-Wallis rank sum test
##
## data:  .$y and .$treatment
## Kruskal-Wallis chi-squared = 25.369, df = 3, p-value = 1.292e-05
```

```
#stat.test <- tidy(stat.test) %>% add_xy_position(x = "treatment")
dfyt %>% filter(treatment != "norm") %>% ggplot(aes(x = treatment, y = y, color=treatment))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE, show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07"
  )+
  ylab("Max y") +
  xlab("Treatment") +
```

```
coord_cartesian(ylim = c(-27, 60)) +
scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140"))+
ggtitle("Wrist Max y") +
stat_compare_means(comparisons=my_comps, label="p.signif") # + # default is Wilcoxon
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('ybox.pdf',width=2.5,height=3,units='in')
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```

```
kruskal.test(y ~ treatment, data = dfyt)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: y by treatment
## Kruskal-Wallis chi-squared = 25.369, df = 3, p-value = 1.292e-05
```

```
pairwise.wilcox.test(dfyt$y, dfyt$treatment,
p.adjust.method = "BH")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
```

```
## data: dfyt$y and dfyt$treatment
##
##      excite  control  inhib
## control 4.8e-06 -      -
## inhib   0.00015 0.86011 -
## norm    0.00168 0.02637 0.27636
##
## P value adjustment method: BH
```

Ok let's update with velocities, reach duration, deltaX, deltaY plots. Four way with norm

```
md<-read_csv("reachfiles_generated_wvelocities.csv",col_types = cols(type=col_factor())) %>% filter(good)
```

```
## New names:
## * `` -> `...1`
```

```
md$type <- relevel(md$type,'control')
md$type <- relevel(md$type,'excite')
my_comps4 <- list( c("control","inhib"),c("excite","control"),
                  c("excite","inhib"), c("control","norm"), c("excite","norm"), c("norm","inhib") )
```

```
md3<-read_csv("reachfiles_generated_wvelocities.csv",col_types = cols(type=col_factor())) %>% filter(good)
```

```
## New names:
## * `` -> `...1`
```

```
md3$type <- relevel(md3$type,'control')
md3$type <- relevel(md3$type,'excite')
my_comps3 <- list( c("control","inhib"),c("excite","control"),
                  c("excite","inhib") )
```

```
md %>% ggplot(aes(x = type, y = velx, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1,
              color = "#FC4E07", )+
  ylab("X Velocity") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024","#3953A4","#0C8140","#000000"))+
  ggtitle("X Velocity vs Treatment") +
  stat_compare_means(comparisons=my_comps, label="p.signif") # + # default is Wilcox
```

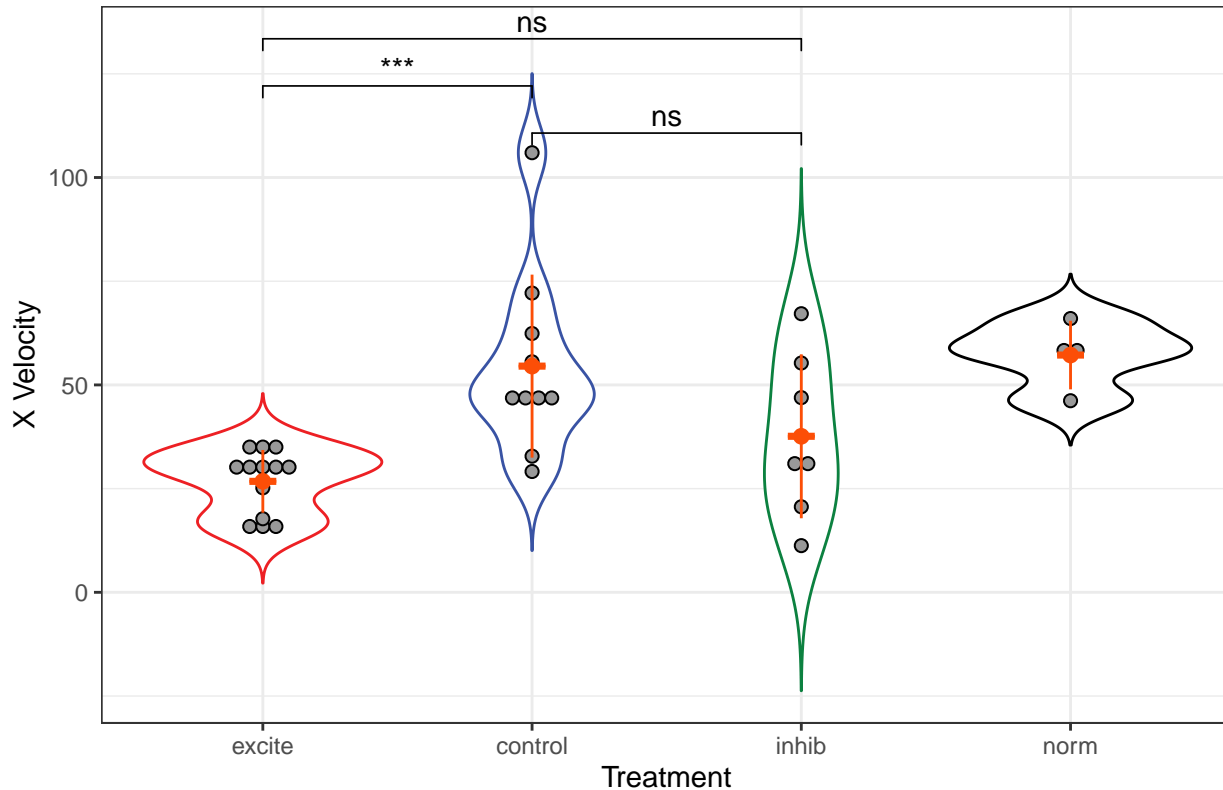
```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```

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

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

### X Velocity vs Treatment



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('xvel.pdf',width=2.5,height=3,units='in')
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

3 way Ok let's update with velocities, reach duration, deltaX, deltaY plots.

```
md3 %>% ggplot(aes(x = type, y = velx, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
```

```

    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")
  ylab("X Velocity") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140"))+
  ggtitle("X Velocity vs Treatment") +
  stat_compare_means(comparisons=my_comps3, label="p.signif") # + # default is Wilcox

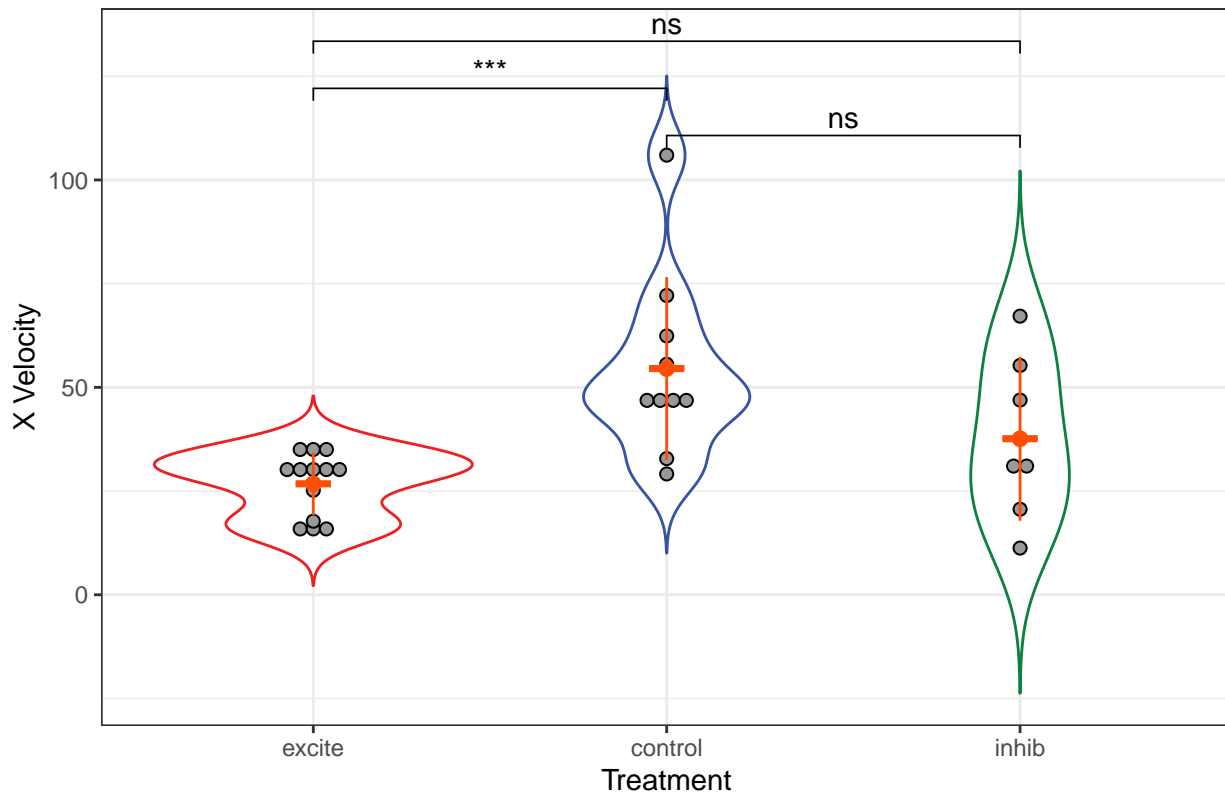
```

```

## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).

```

### X Velocity vs Treatment



```

#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('xvel3.pdf',width=2.5,height=3,units='in')

```

```

## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with

```

```

## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
and now y four way
md %>% ggplot(aes(x = type, y = -vely, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07"
  ylab("Y Velocity") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140", "#000000"))+
  ggtitle("Y Velocity vs Treatment") +
  stat_compare_means(comparisons=my_comps4, label="p.signif") # + # default is Wilcoxon

## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).

```

Violin plot showing Y Velocity for four treatments: excite, control, inhib, and norm. The y-axis ranges from 0 to 120. Individual data points are overlaid on the violins. Statistical significance is indicated by brackets: excite vs control (ns), excite vs inhib (\*), excite vs norm (\*\*), control vs inhib (ns), control vs norm (ns), and inhib vs norm (\*\*).

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

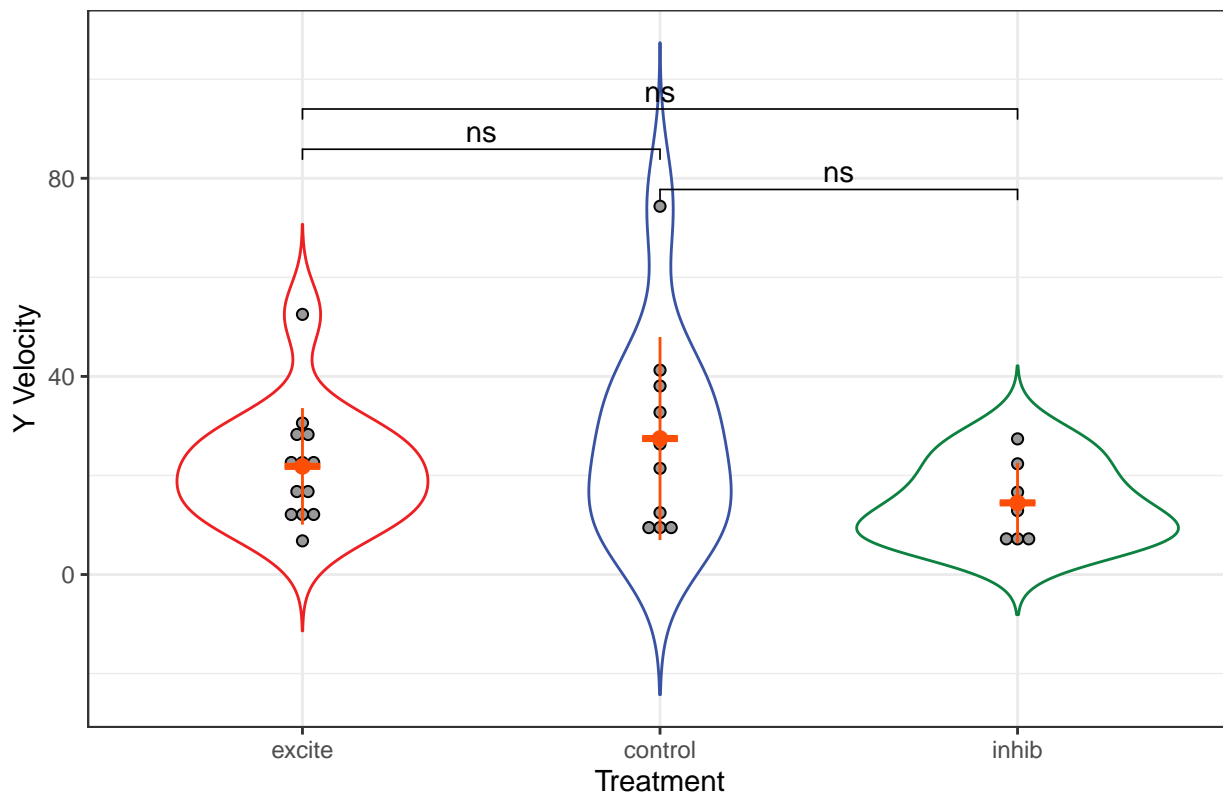
```
md3 %>% ggplot(aes(x = type, y = -vely, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE, show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")
```



```
ylab("Y Velocity") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140"))+
  ggtitle("Y Velocity vs Treatment") +
  stat_compare_means(comparisons=my_comps3, label="p.signif") # + # default is Wilcoxon
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

### Y Velocity vs Treatment



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('yvel3.pdf',width=2.5,height=3,units='in')
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
```

```

## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
and now dur 4 way
md %>% ggplot(aes(x = type, y = rchdur, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07"
  ylab("Reach Duration (s)") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140", "#000000"))+
  ggtitle("Reach Duration vs Treatment") +
  stat_compare_means(comparisons=my_comps4, label="p.signif") # + # default is Wilcox

## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

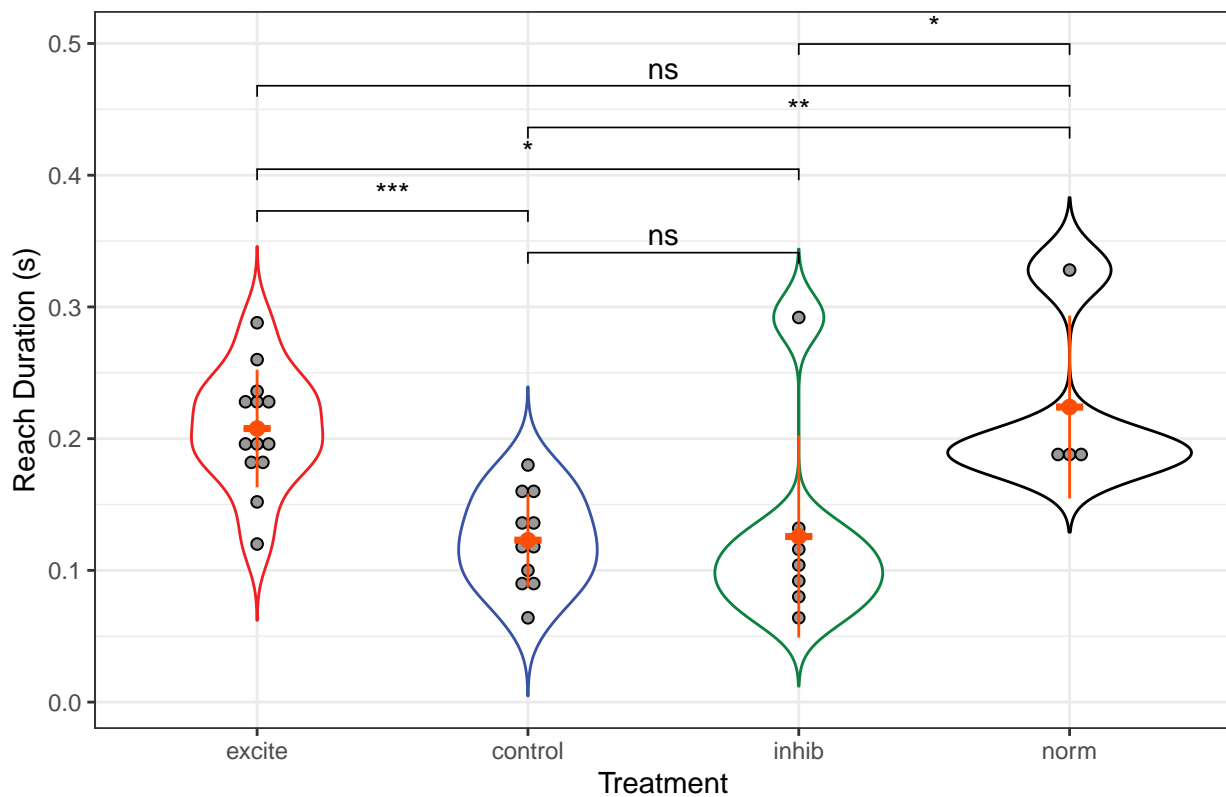
## Warning in wilcox.test.default(c(0.16, 0.116, 0.18, 0.092, 0.12, 0.1, 0.088, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.16, 0.116, 0.18, 0.092, 0.12, 0.1, 0.088, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.328, 0.184, 0.192, 0.192), c(0.104, 0.292, :
## cannot compute exact p-value with ties

```

## Reach Duration vs Treatment



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('reachdur.pdf',width=2.5,height=3,units='in')
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.16, 0.116, 0.18, 0.092, 0.12, 0.1, 0.088, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.16, 0.116, 0.18, 0.092, 0.12, 0.1, 0.088, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.328, 0.184, 0.192, 0.192), c(0.104, 0.292, :
## cannot compute exact p-value with ties
```

and now dur

```
md3 %>% ggplot(aes(x = type, y = rchdur, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
```

```

    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  ) +
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")
ylab("Reach Duration (s)") +
xlab("Treatment") +
#coord_cartesian(ylim = c(-27, 60)) +
scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140")) +
ggtitle("Reach Duration vs Treatment") +
stat_compare_means(comparisons=my_comps3, label="p.signif") # + # default is Wilcox

```

```

## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.

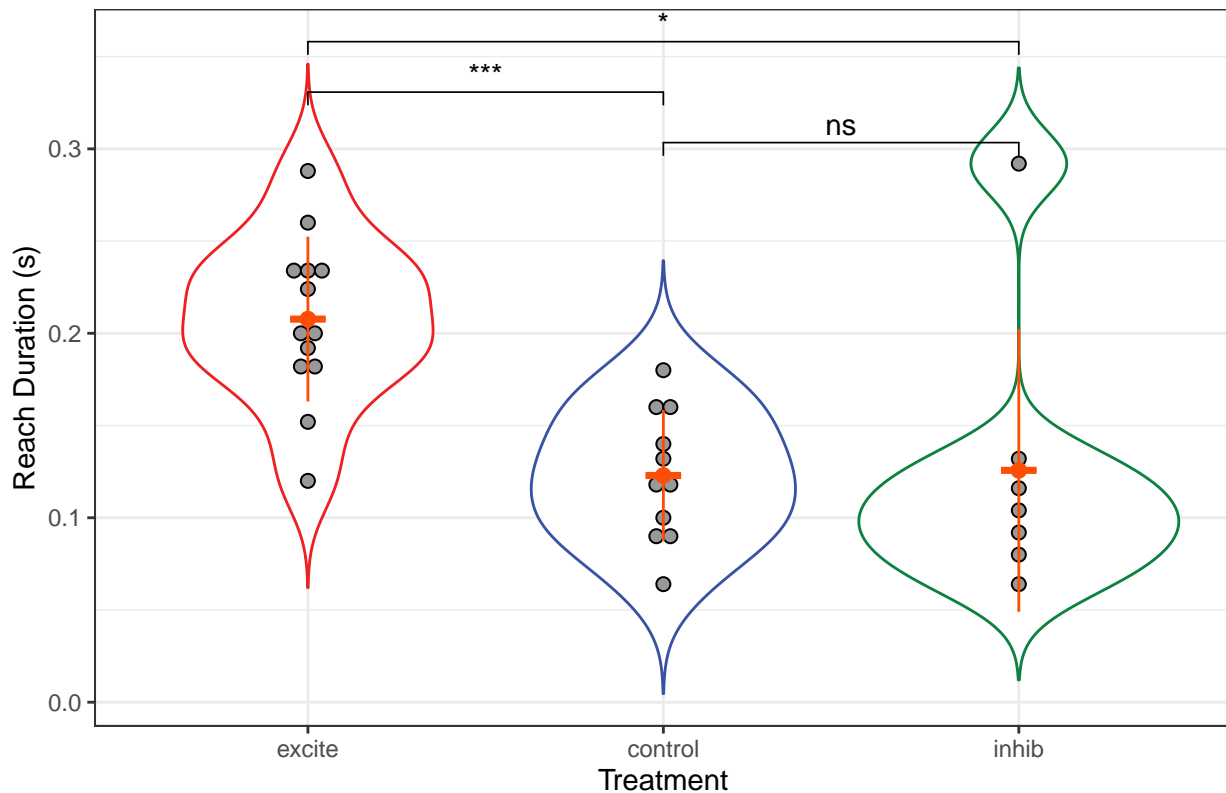
## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0.16, 0.116, 0.18, 0.092, 0.12, 0.1, 0.088, :
## cannot compute exact p-value with ties

```

## Reach Duration vs Treatment



```

#stat_compare_means(aes(group = treatment), label = "p.signif",
#
#                    method='kruskal.test', hide.ns = TRUE )

```

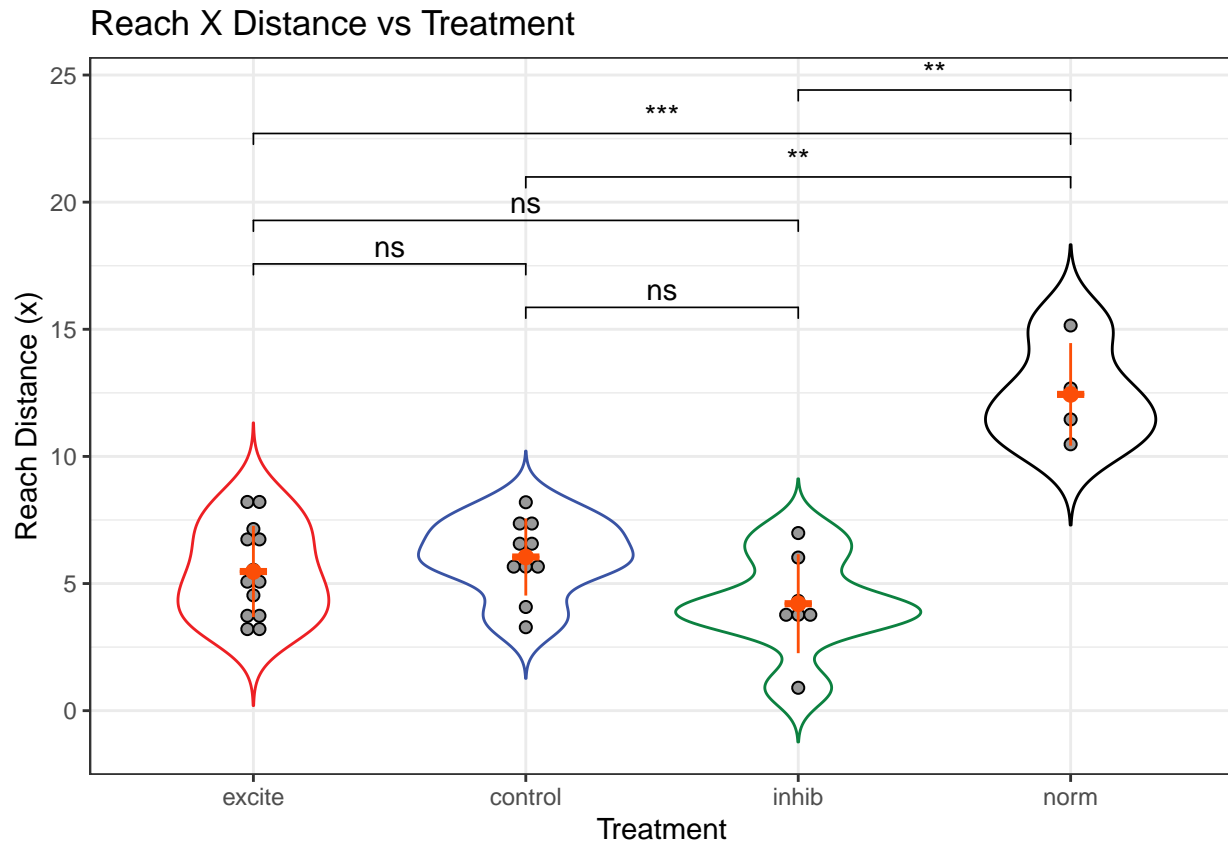
```
ggsave('reachdur3.pdf',width=2.5,height=3,units='in')
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with  
## `binwidth`.  
  
## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :  
## cannot compute exact p-value with ties  
  
## Warning in wilcox.test.default(c(0.152, 0.26, 0.2, 0.232, 0.192, 0.18, 0.12, :  
## cannot compute exact p-value with ties  
  
## Warning in wilcox.test.default(c(0.16, 0.116, 0.18, 0.092, 0.12, 0.1, 0.088, :  
## cannot compute exact p-value with ties
```

and now deltax

```
md %>% ggplot(aes(x = type, y = dx, color=type))+  
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+  
  geom_violin(trim = FALSE,show.legend=FALSE) +  
  geom_dotplot(  
    binaxis='y', stackdir='center',  
    color = "black", fill = "#999999",  
  ) +  
  stat_summary(  
    fun.data="mean_sdl", fun.args = list(mult=1),  
    geom = "pointrange", color = "#FC4E07", size = 0.4  
  )+  
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")  
ylab("Reach Distance (x)") +  
  xlab("Treatment") +  
  #coord_cartesian(ylim = c(-27, 60)) +  
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140", "#000000"))+  
  ggtitle("Reach X Distance vs Treatment") +  
  stat_compare_means(comparisons=my_comps4, label="p.signif") # + # default is Wilcox
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).  
## Bin width defaults to 1/30 of the range of the data. Pick better value with  
## `binwidth`.  
  
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).  
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).  
## Removed 1 rows containing non-finite values (`stat_summary()`).  
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('xdistance.pdf',width=2.5,height=3,units='in')
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

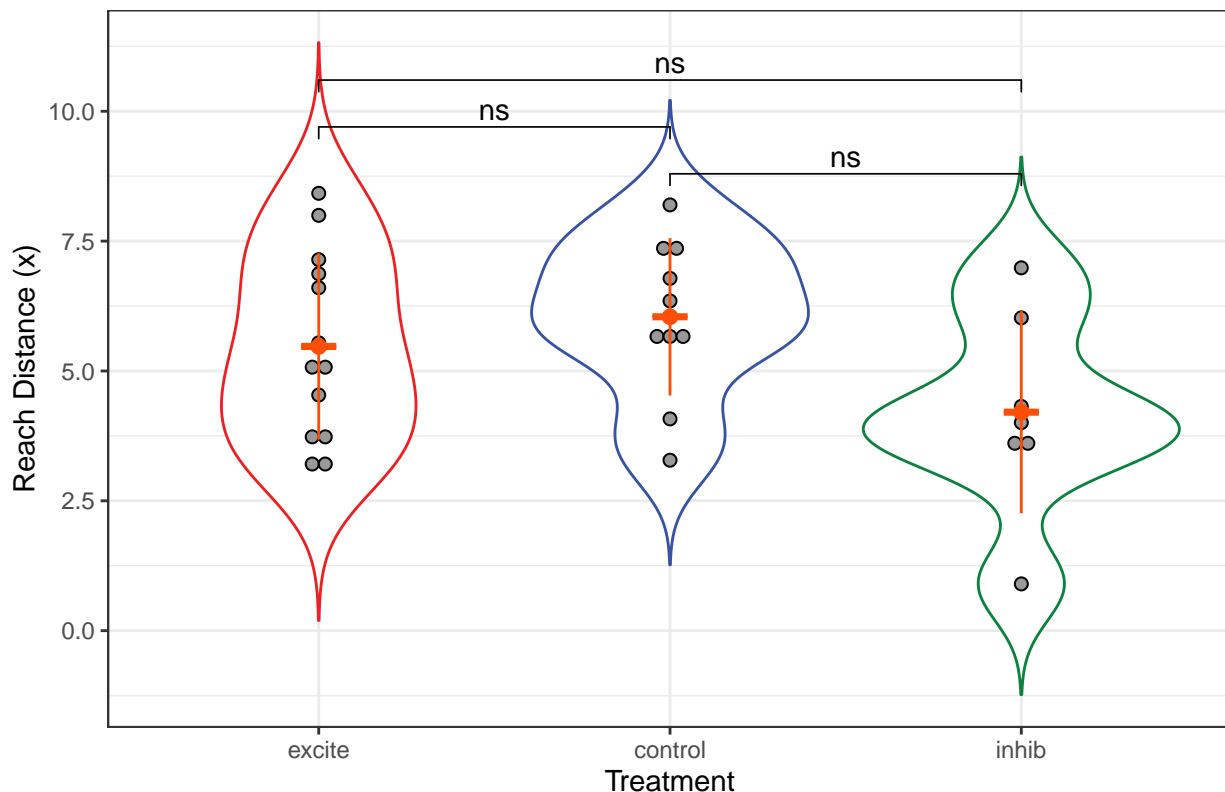
and now deltax3way

```
md3 %>% ggplot(aes(x = type, y = dx, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")
```

```
ylab("Reach Distance (x)") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140"))+
  ggtitle("Reach X Distance vs Treatment") +
  stat_compare_means(comparisons=my_comps3, label="p.signif") # + # default is Wilcoxon
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

### Reach X Distance vs Treatment



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('xdistance3.pdf',width=2.5,height=3,units='in')
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
```

```

## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
and now deltax
md %>% ggplot(aes(x = type, y = dy, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07"
  ylab("Reach Distance (y)") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140", "#000000"))+
  ggtitle("Reach Y Distance vs Treatment") +
  stat_compare_means(comparisons=my_comps4, label="p.signif") # + # default is Wilcoxon

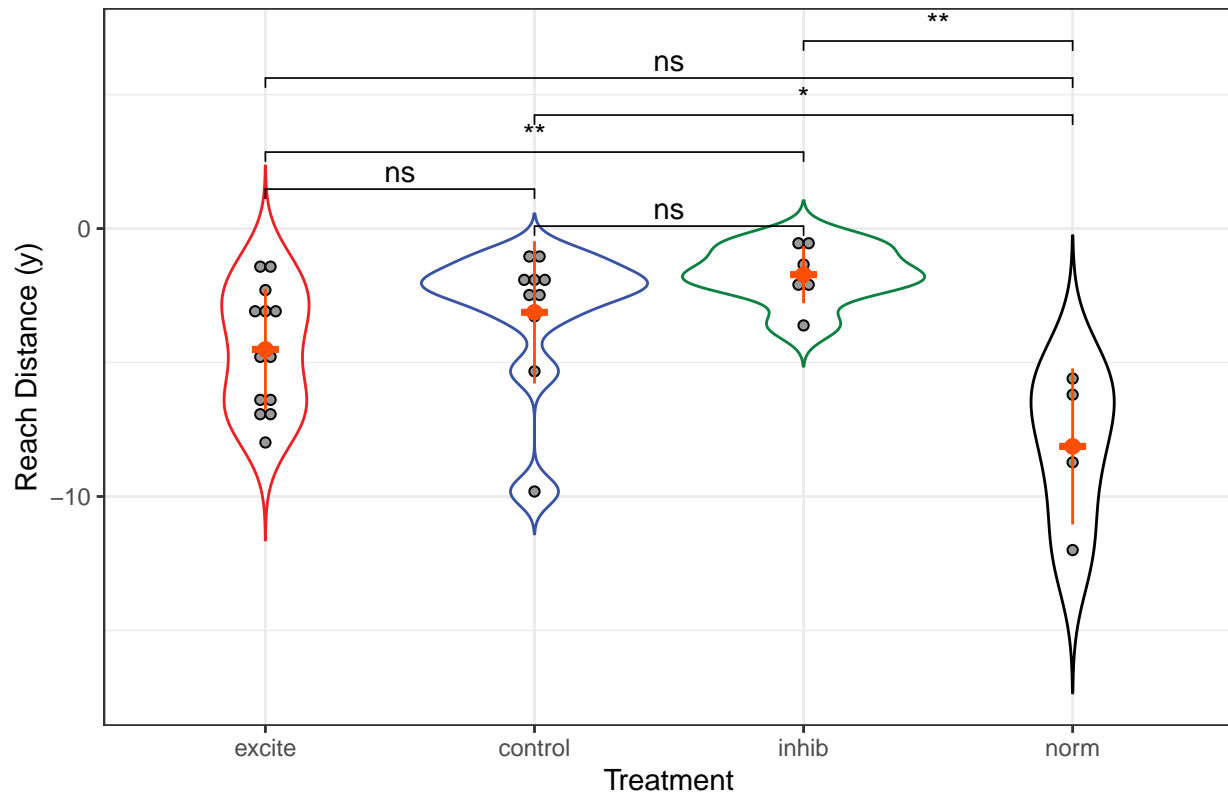
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.

## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).

```



## Reach Y Distance vs Treatment



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('ydistance.pdf',width=2.5,height=3,units='in')
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

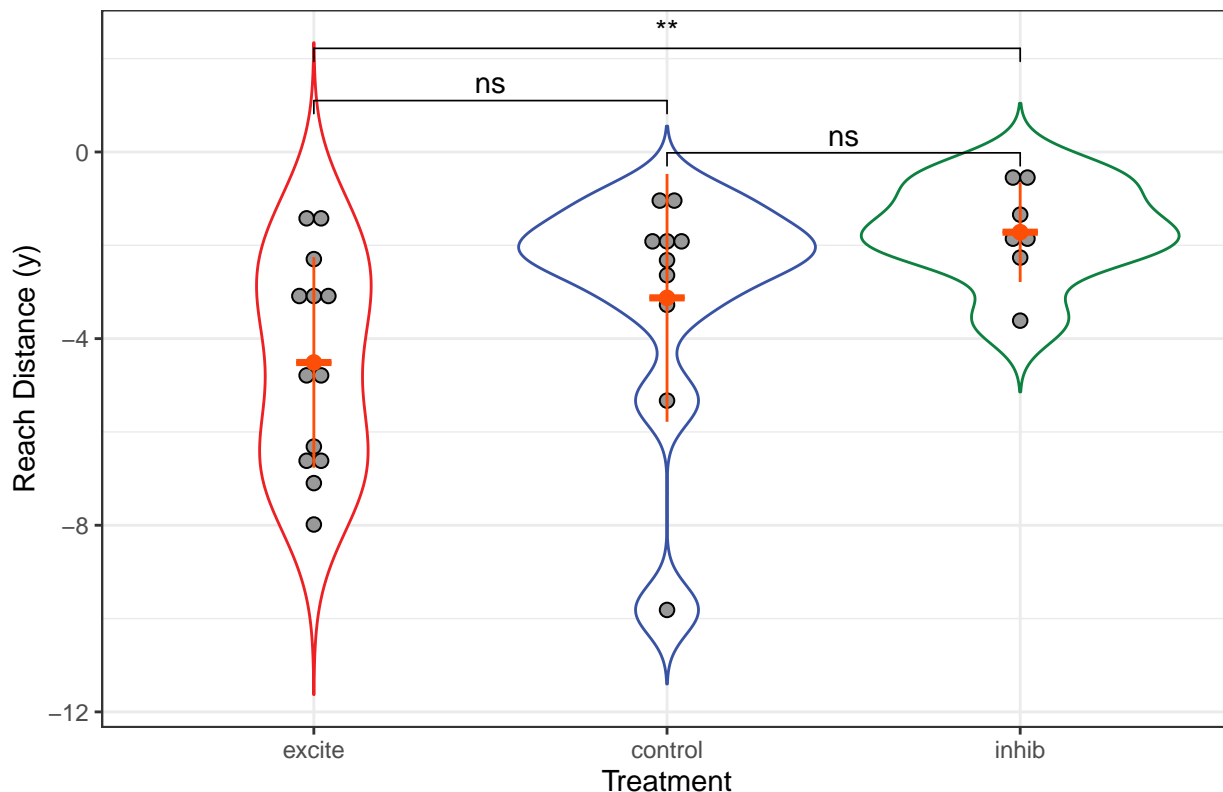
and now deltax3way

```
md3 %>% ggplot(aes(x = type, y = dy, color=type))+
  #geom_boxplot(aes(color = treatment), show.legend = FALSE)+
  geom_violin(trim = FALSE,show.legend=FALSE) +
  geom_dotplot(
    binaxis='y', stackdir='center',
    color = "black", fill = "#999999",
  ) +
  stat_summary(
    fun.data="mean_sdl", fun.args = list(mult=1),
    geom = "pointrange", color = "#FC4E07", size = 0.4
  )+
  stat_summary(fun.y= mean, fun.ymin=mean, fun.ymax=mean, geom="crossbar", width=0.1, color = "#FC4E07")
```

```
ylab("Reach Distance (y)") +
  xlab("Treatment") +
  #coord_cartesian(ylim = c(-27, 60)) +
  scale_color_manual(values = c("#ED2024", "#3953A4", "#0C8140"))+
  ggtitle("Reach Y Distance vs Treatment") +
  stat_compare_means(comparisons=my_comps3, label="p.signif") # + # default is Wilcox
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
## Removed 1 rows containing non-finite values (`stat_summary()`).
## Warning: Removed 1 rows containing non-finite values (`stat_signif()`).
```

### Reach Y Distance vs Treatment



```
#stat_compare_means(aes(group = treatment), label = "p.signif",
#                    method='kruskal.test', hide.ns = TRUE )
ggsave('ydistance3.pdf',width=2.5,height=3,units='in')
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_ydensity()`).
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
## Warning: Removed 1 rows containing missing values (`stat_bindot()`).
## Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
```

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