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(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")
#  )
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: 3 x 6
##   treatment count  mean    sd median   IQR
##   <fct>     <int> <dbl> <dbl>  <dbl> <dbl>
```

```
## 1 excite       11 12.4   4.51  12.4   3.79
## 2 control       7 -5.20  4.21  -3.31  3.13
## 3 inhib         7 -9.84  5.61  -8.65  4.19
```

```
# Plot it:
# choose comparisons to plot
# http://www.sthda.com/english/articles/32-r-graphics-essentials/132-plot-grouped-data-box-plot-bar-plo
my_comps <- list( c("control","inhib"),c("excite","control"),
                  c("excite","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, 35)) +
  scale_color_manual(values = c("#ED2024","#3953A4","#0C8140"))+
  ggtitle("Wrist Max x")+
  stat_compare_means(comparisons=my_comps, label="p.signif") # + # default is Wilcox
```

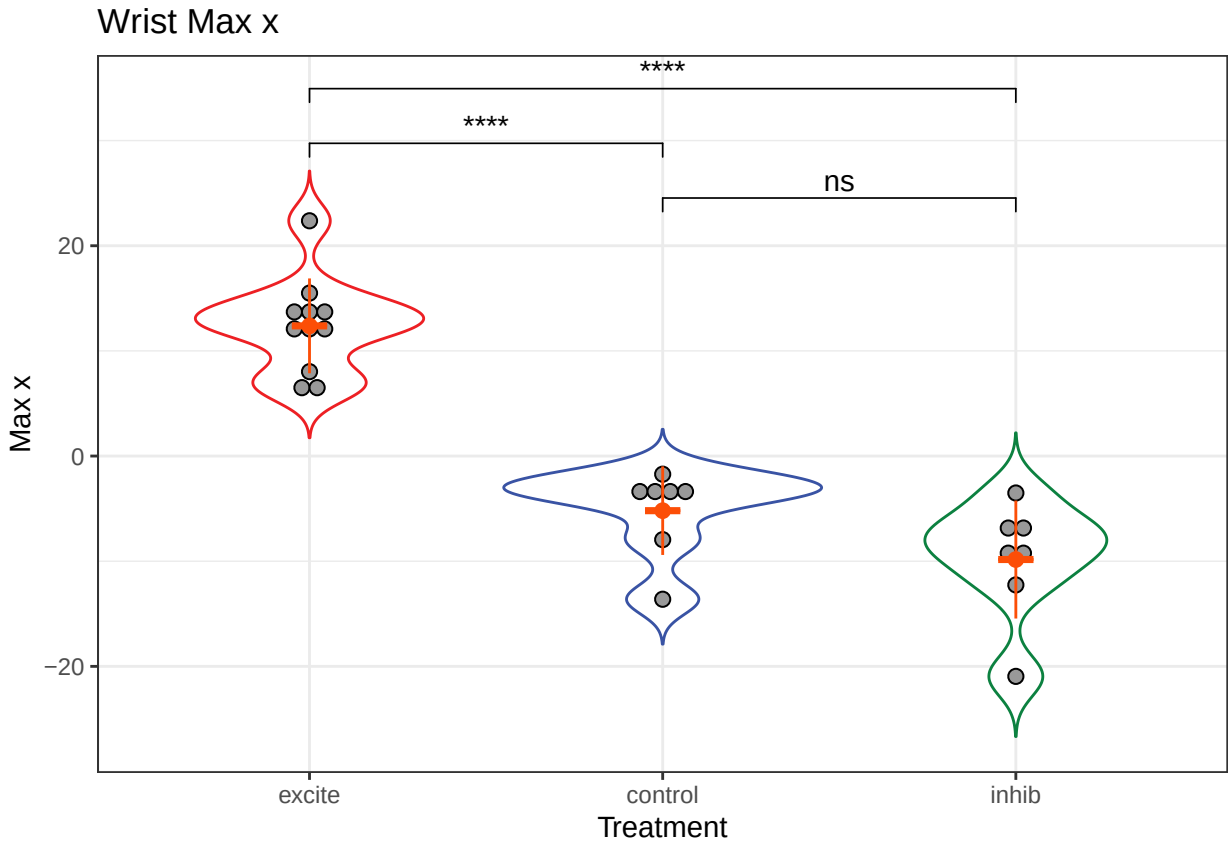
```
## Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
## i Please use the `fun` argument instead.
```

```
## 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.
```

```
## [1] FALSE
```

```
## 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 = 18.878, df = 2, p-value = 7.955e-05
```

```
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
## control 9.4e-05 -
## inhib   9.4e-05 0.073
##
## 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: 3 x 6
##   treatment count  mean    sd median   IQR
##   <fct>     <int> <dbl> <dbl>  <dbl> <dbl>
## 1 excite       11 20.7   8.79 19.5   12.9
## 2 control       7 -3.94  6.35 -3.37   7.02
## 3 inhib         7 -1.45  7.00 -0.945  9.72
```

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

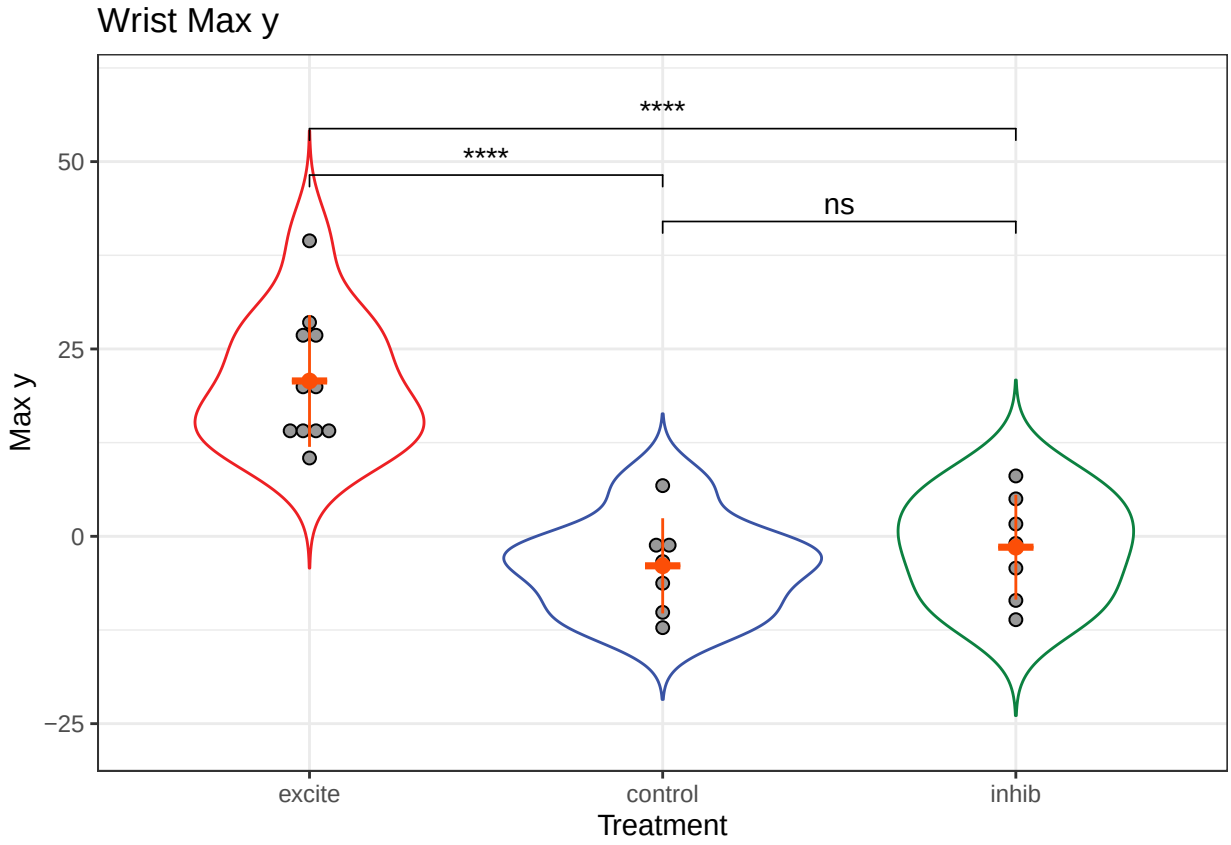
```
##
##  Kruskal-Wallis rank sum test
##
## data:  .$y and .$treatment
## Kruskal-Wallis chi-squared = 17.992, df = 2, p-value = 0.0001239
```

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

```
## [1] FALSE
```

```
## 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 = 17.992, df = 2, p-value = 0.0001239
```

```
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
## control 9.4e-05 -
## inhib   9.4e-05 0.46
##
## P value adjustment method: BH
```