Raincloud Plots with the ggrain package

Nicholas Judd, Jordy van Langen & Rogier Kievit

The `geom_rain()` function

- handles as many rainclouds as you wish and can overlap them by a group
- connects within-subject observations longitudinally with lines using `id.long.var` argument
- colors dots by a covariate using the `cov` argument
- handles likert data by adding y-jittering with `likert = TRUE`
- changes orientation with `+ coord_flip()`

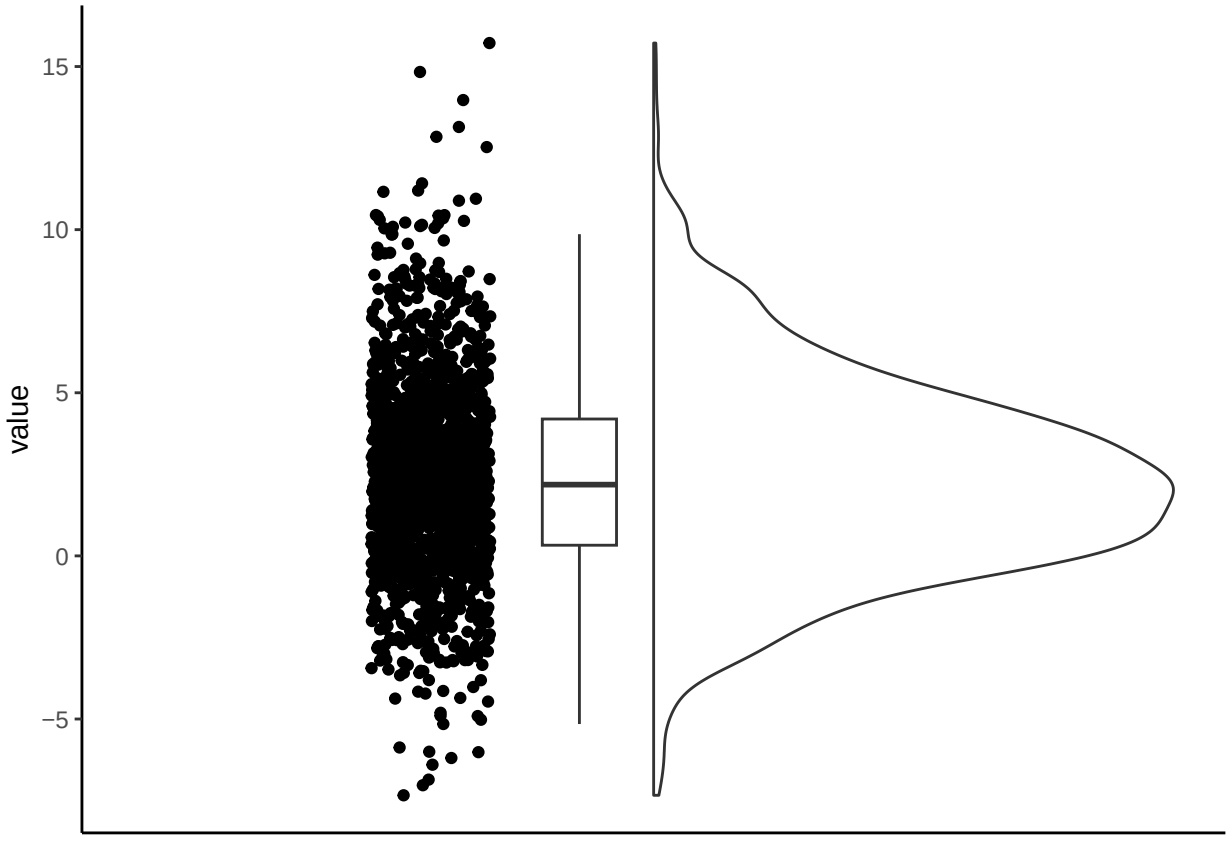
All individual elements of the plots can be edited, these are split into aesthetic and positioning arguments that are supplied by lists. For example the boxplot's can be edited with `boxplot.args` and `boxplot.args.pos`, yet the others can also be edited by substituting for their name, i.e. `point/violin/line`. When you supply a list the defaults are overwritten so you may need to re-add them. To see the defaults run `?geom_rain`.

Introduction

We will look at some aging data that has two groups (young and old) measured four times.

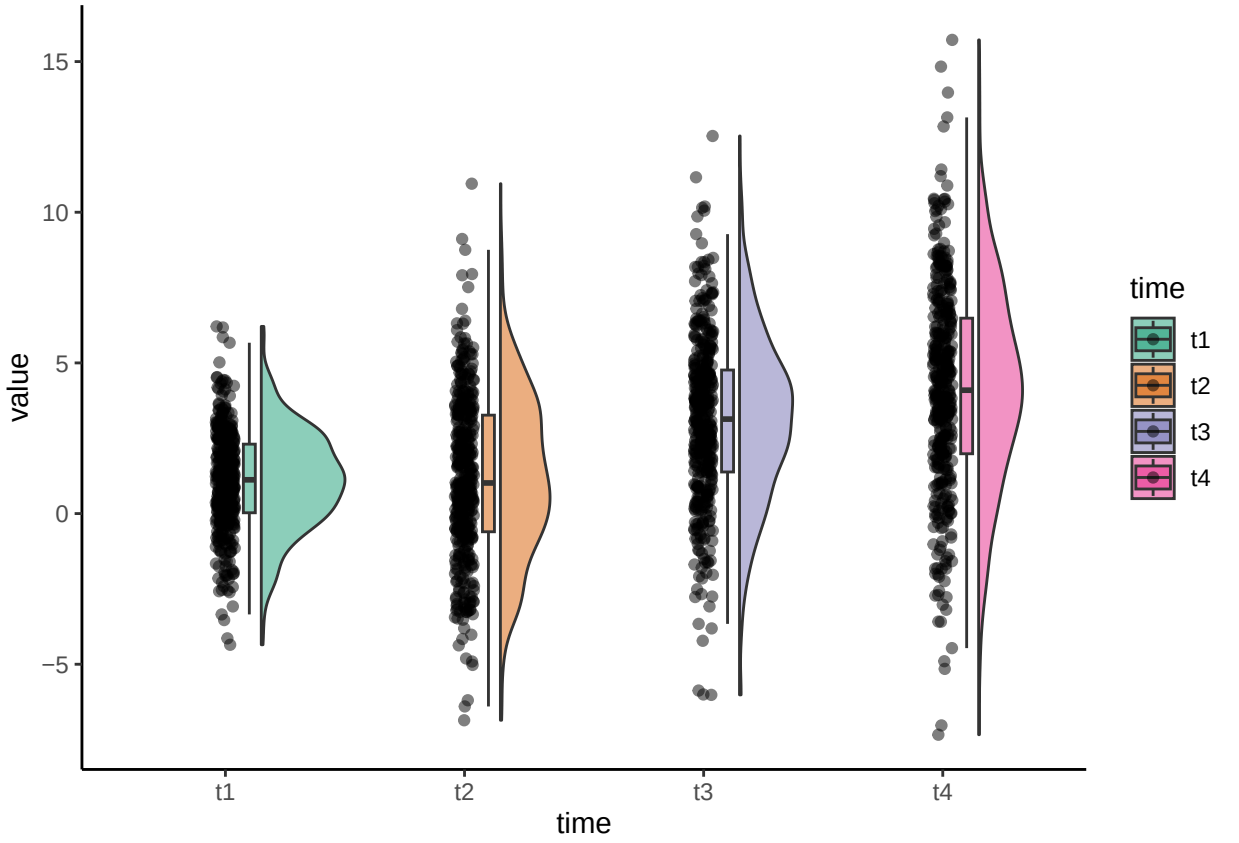
Here is our first plot that is just simply all the values in the aging data set. For the function to work the value you want to plot **must be given to the y argument** in ggplot. You can than flip the plot with `+ coord_flip()` as we demonstrate below.

```
ggplot(aging, aes(1, value)) +
  geom_rain() +
  theme_classic() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(), axis.ticks.x = element_blank())
```



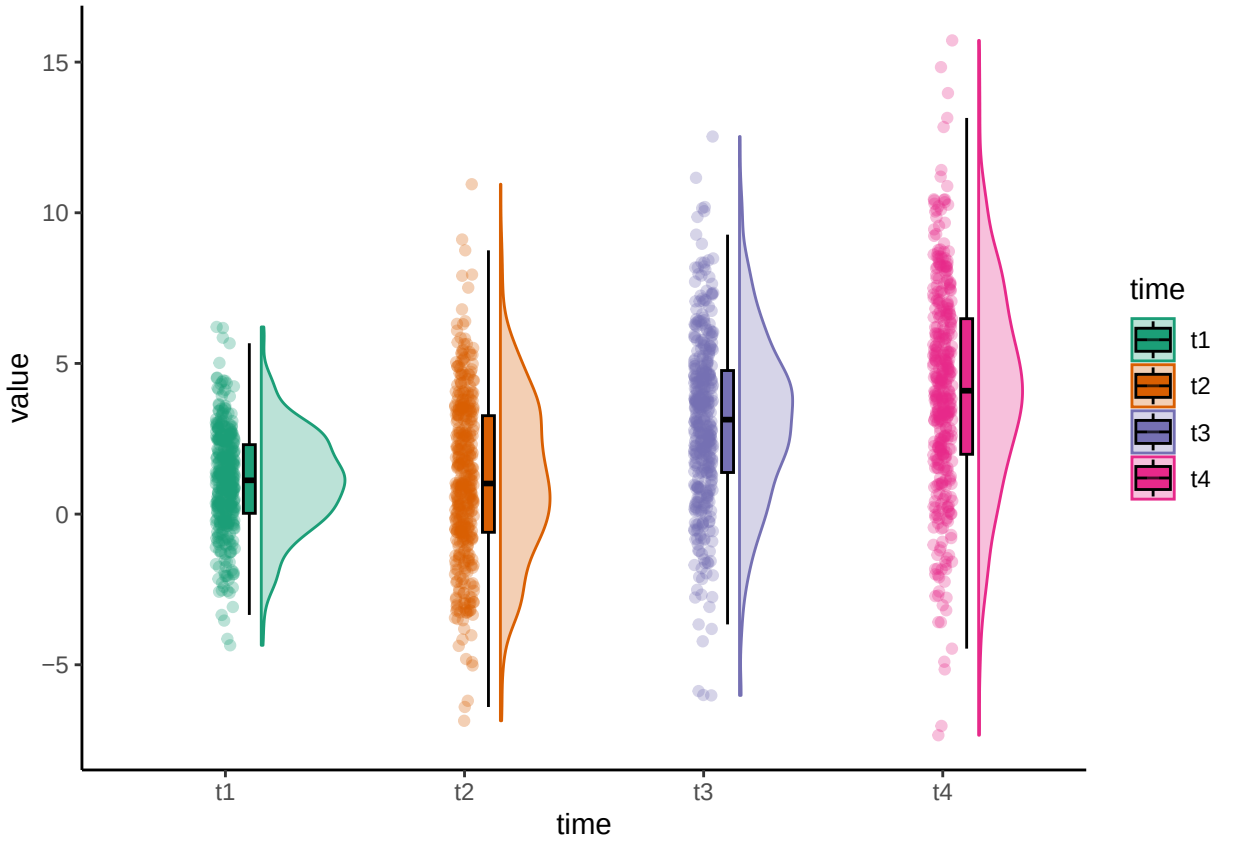
Let's see what is happening over the four time points

```
ggplot(aging, aes(time, value, fill = time)) +
  geom_rain(alpha = .5) +
  theme_classic() +
  scale_fill_brewer(palette = 'Dark2')
```



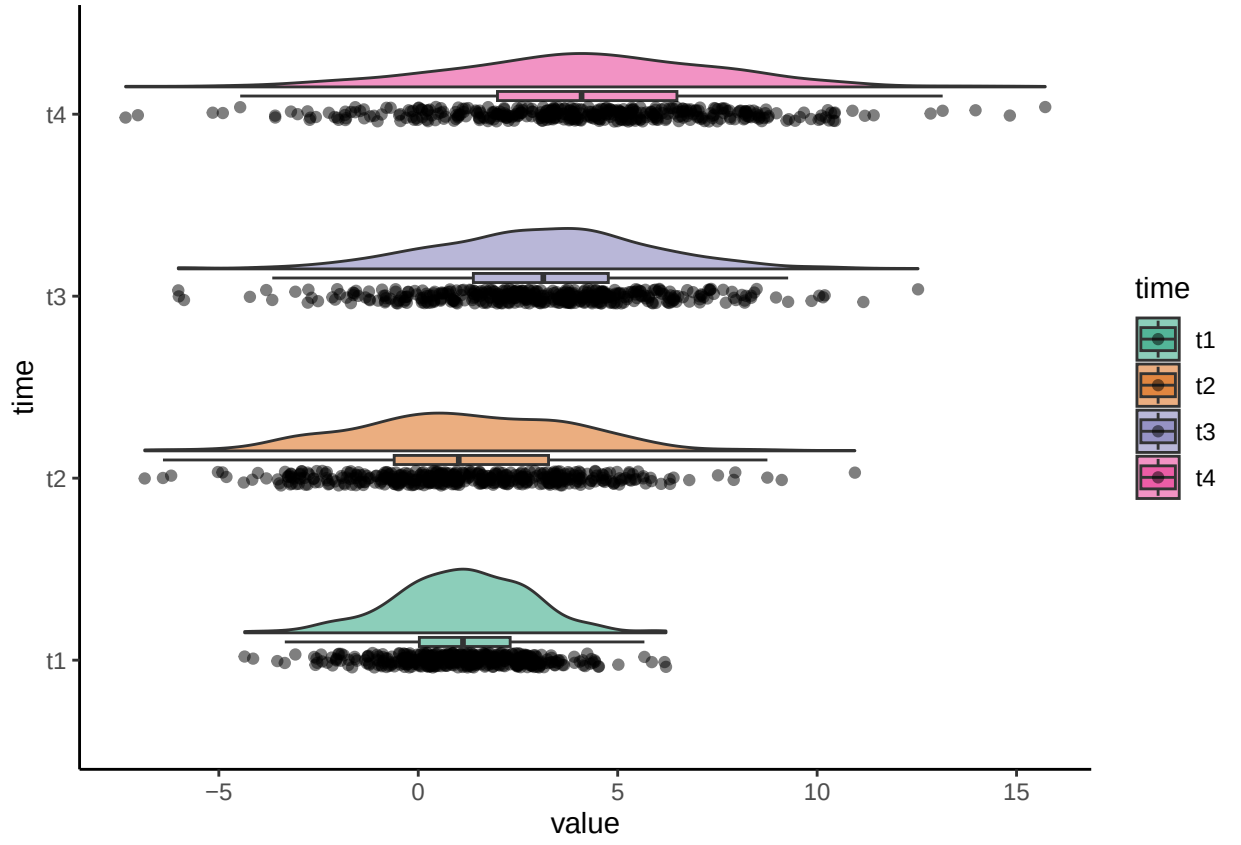
Let's color the dots by time, we do this by adding `color = time` to ggplot. The default behavior of `geom_boxplot` is to color the lines showing the median and IQR. Therefore, we need to add a `boxplot.args` for color to be black along with re-adding the default to show no outliers.

```
ggplot(aging, aes(time, value, fill = time, color = time)) +
  geom_rain(alpha = .3,
            boxplot.args = list(color = "black", outlier.shape = NA)) +
  theme_classic() +
  scale_fill_brewer(palette = 'Dark2') +
  scale_color_brewer(palette = 'Dark2')
```



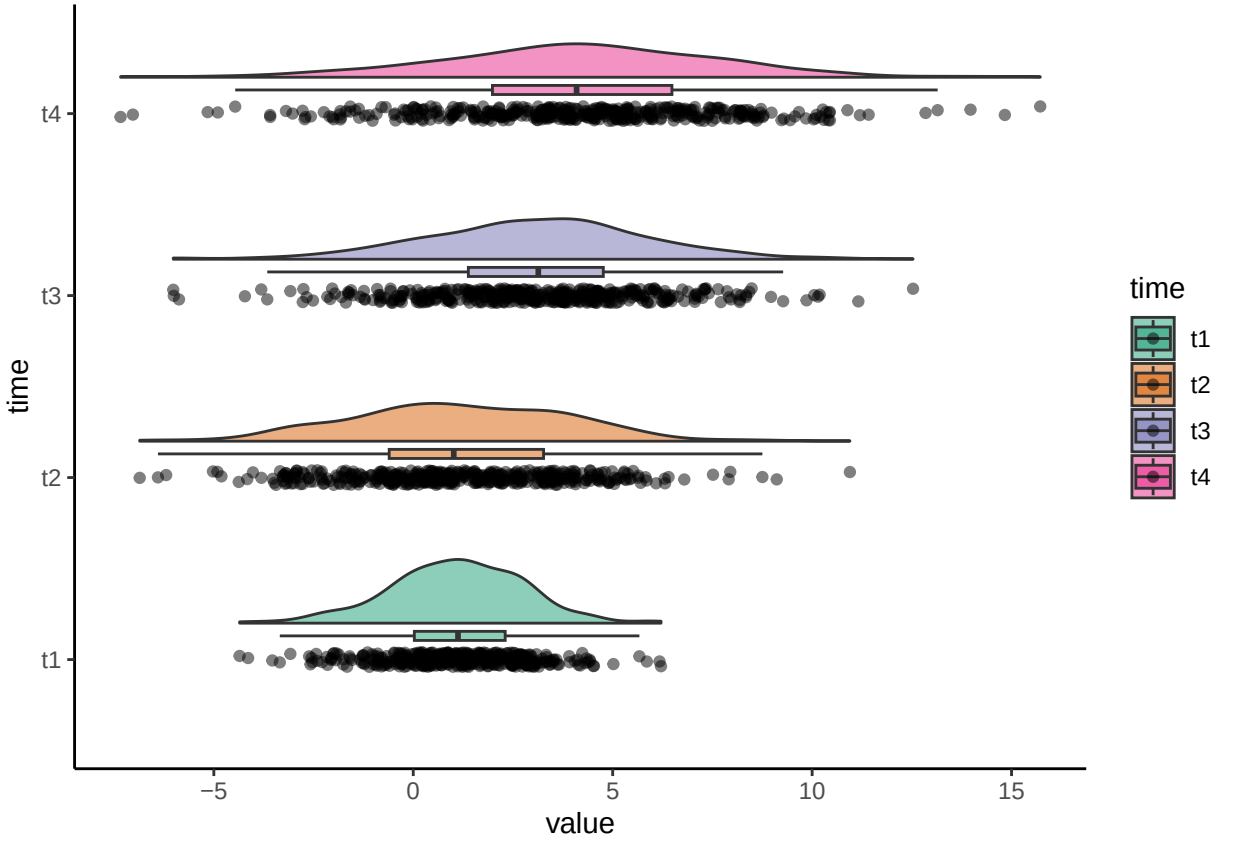
We can flip the plots by adding `coord_flip()`

```
ggplot(aging, aes(time, value, fill = time)) +
  geom_rain(alpha = .5) +
  theme_classic() +
  scale_fill_brewer(palette = 'Dark2') +
  coord_flip()
```



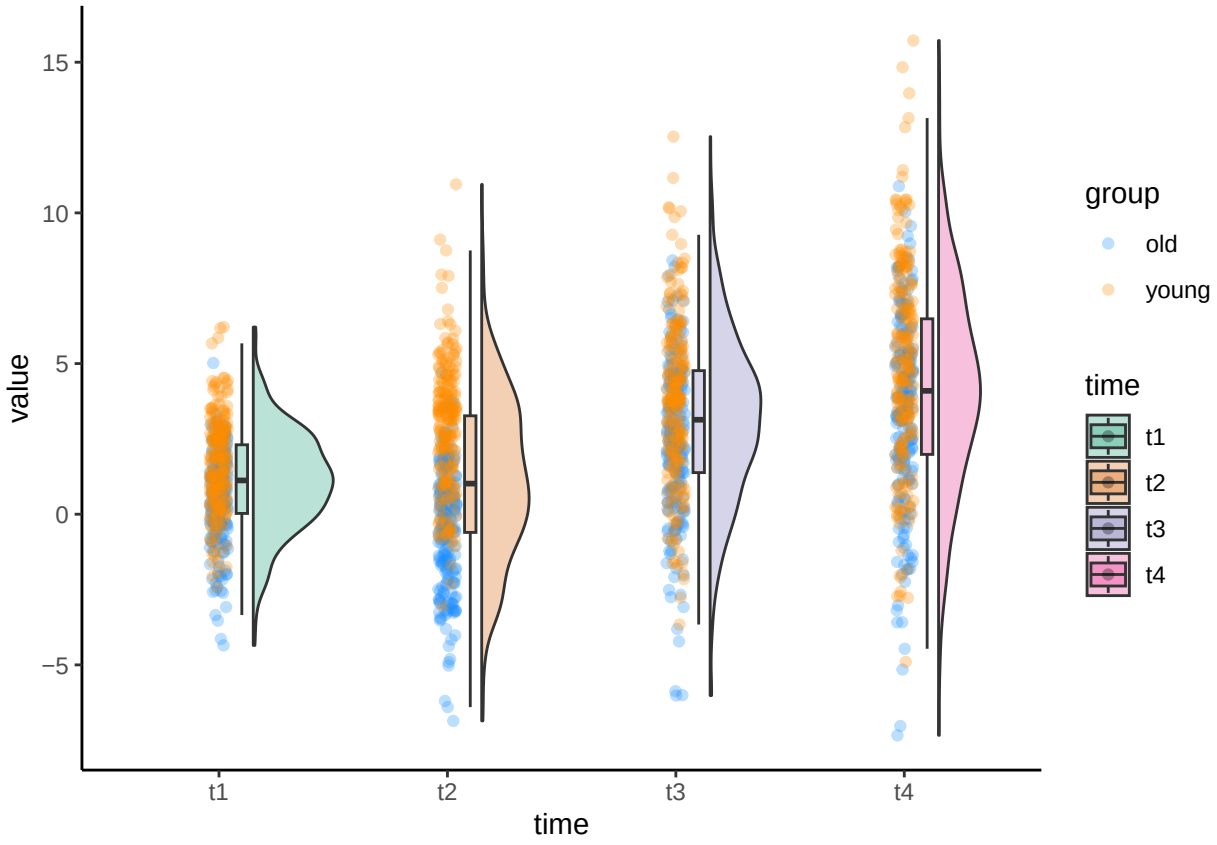
This plot is a bit crammed, lets spread stuff out using the `boxplot.args.pos` & `violin.args.pos` arguments.

```
ggplot(aging, aes(time, value, fill = time)) +
  geom_rain(alpha = .5,
            boxplot.args.pos = list(
              width = 0.05, position = position_nudge(x = 0.13)),
            violin.args.pos = list(
              side = "r",
              width = 0.7, position = position_nudge(x = 0.2))) +
  theme_classic() +
  scale_fill_brewer(palette = 'Dark2') +
  coord_flip()
```



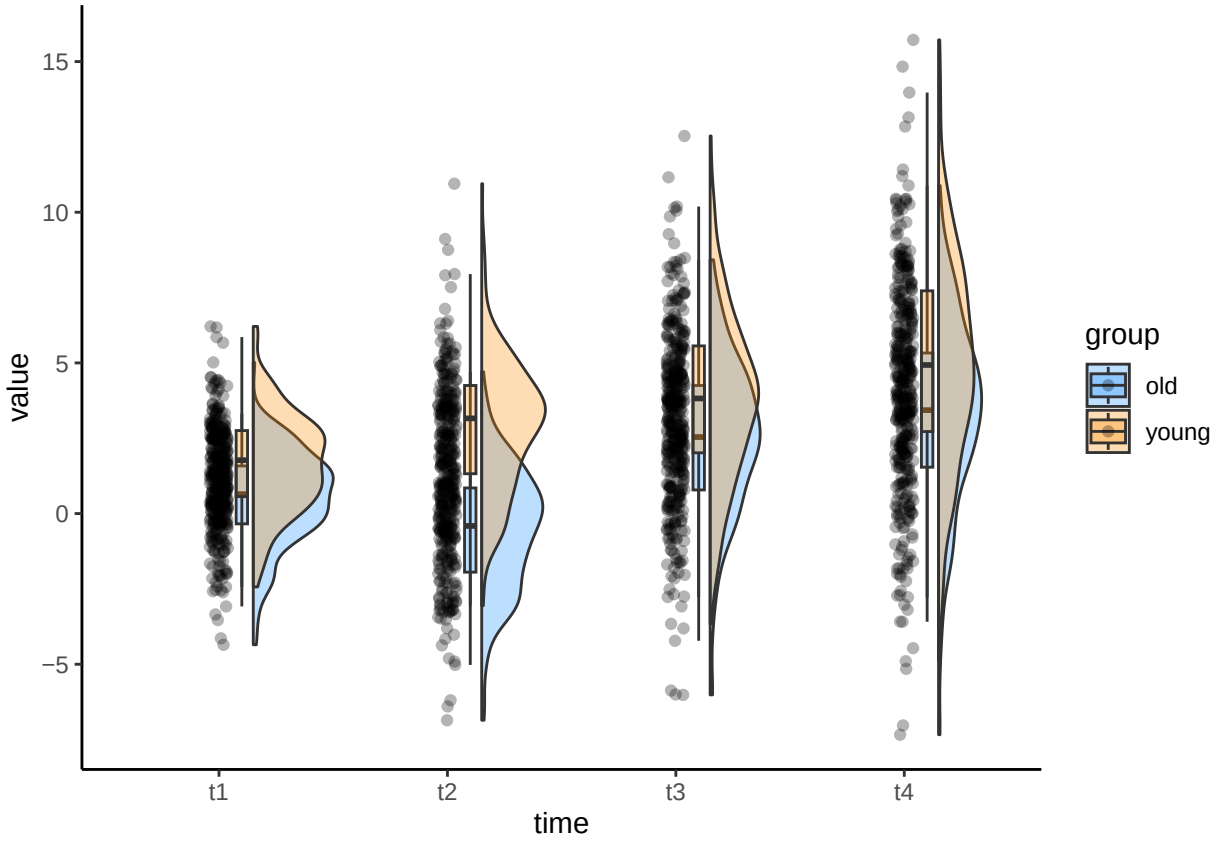
Instead of coloring the dots by time, lets see how our two groups are doing (old & young people). We can do this by adding them as a covariate with the `cov` argument.

```
ggplot(aging, aes(time, value, fill = time)) +
  geom_rain(alpha = .3,
            cov = "group") +
  theme_classic() +
  scale_fill_brewer(palette = 'Dark2') +
  scale_color_manual(values=c("dodgerblue", "darkorange"))
```



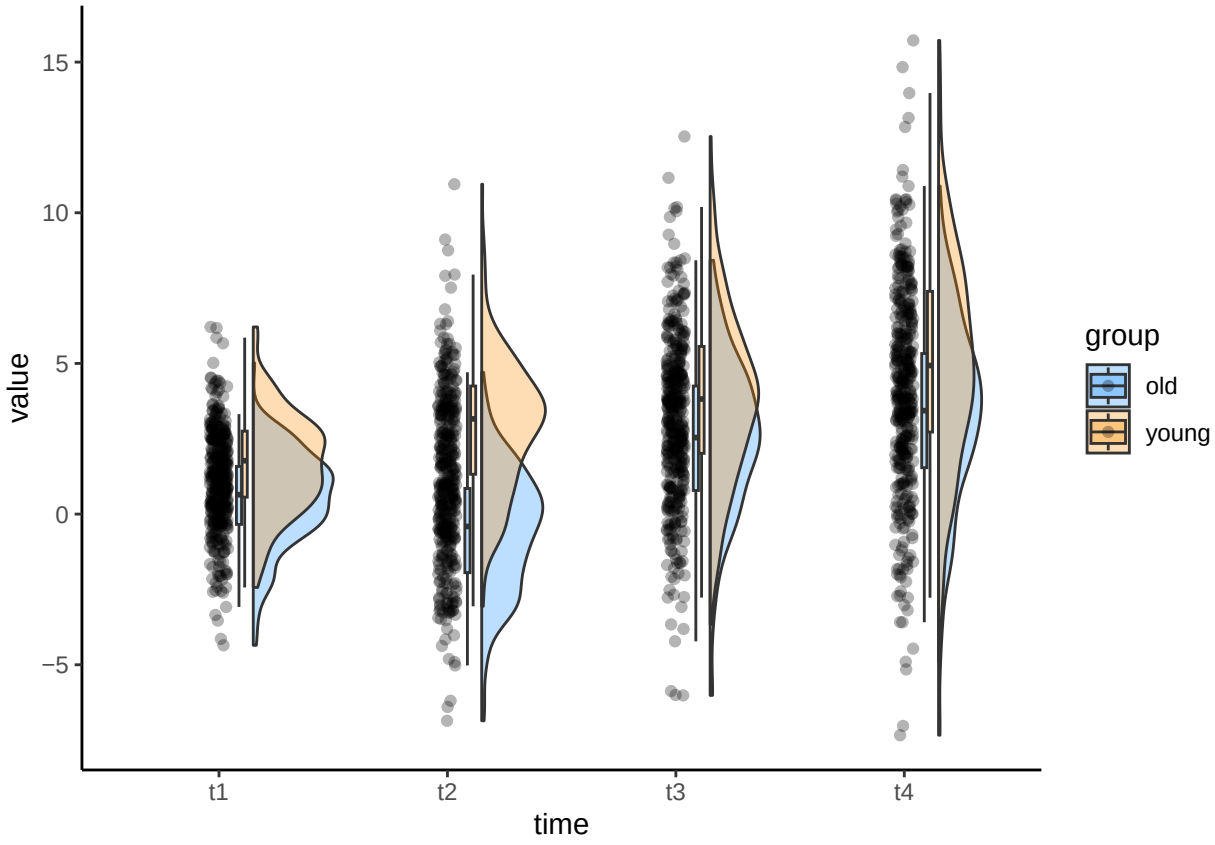
These are distinct groups it would be much more informative to plot them separately!

```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(alpha = .3) +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange"))
```



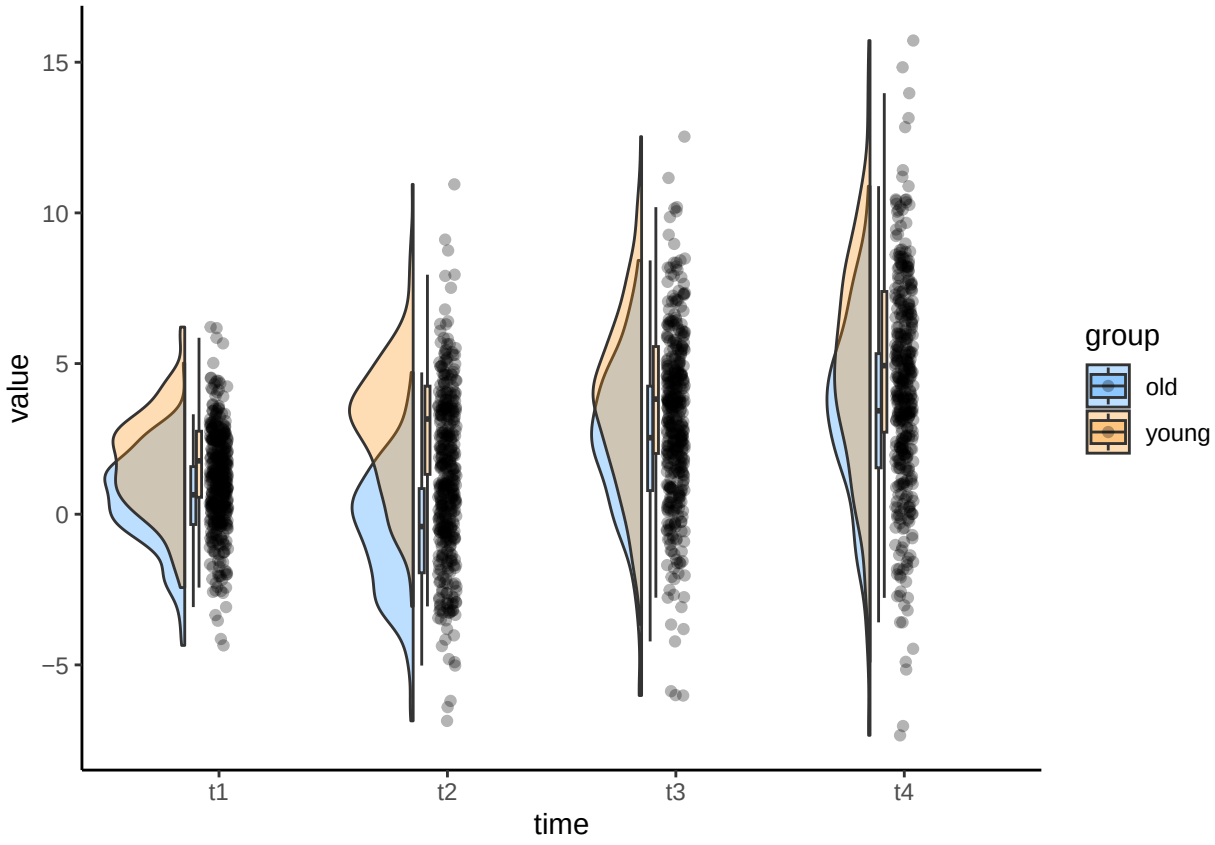
We can't really see the boxplots so lets dodge them while also nudging, we can do this with the boxplot position arg `boxplot.args.pos`. Remember to readd the defaults if you would like to keep them!

```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(alpha = .3,
            boxplot.args.pos = list(
              position = ggpp::position_dodgenudge(x = .1), width = 0.05
            )) +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange"))
```



Lets flip them, while keeping the boxplots dodging.

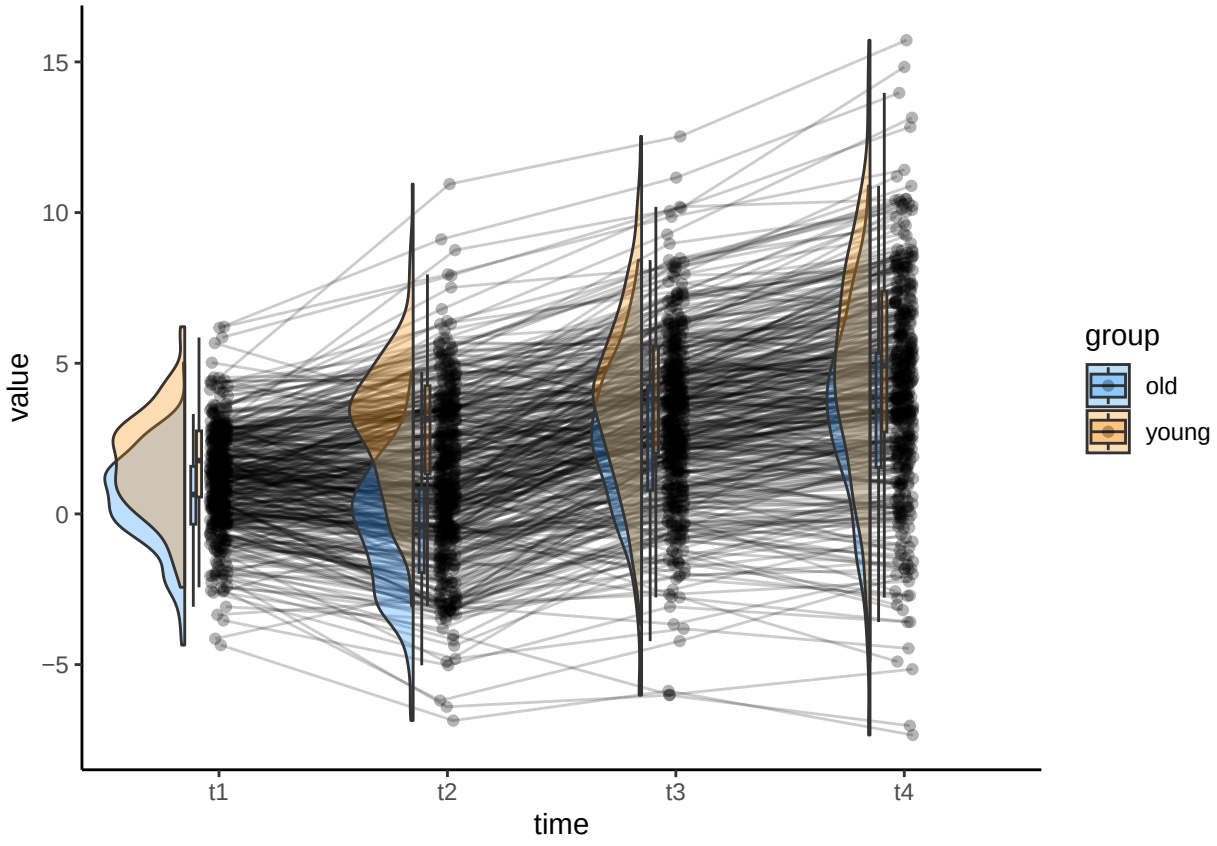
```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(alpha = .3, rain.side = 'l',
            boxplot.args.pos = list(
              position = ggpp::position_dodgenudge(x = .1), width = 0.05
            )) +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange"))
```



As you can see the `rain.side` arg negates the x position argument you give `geom_rain`.

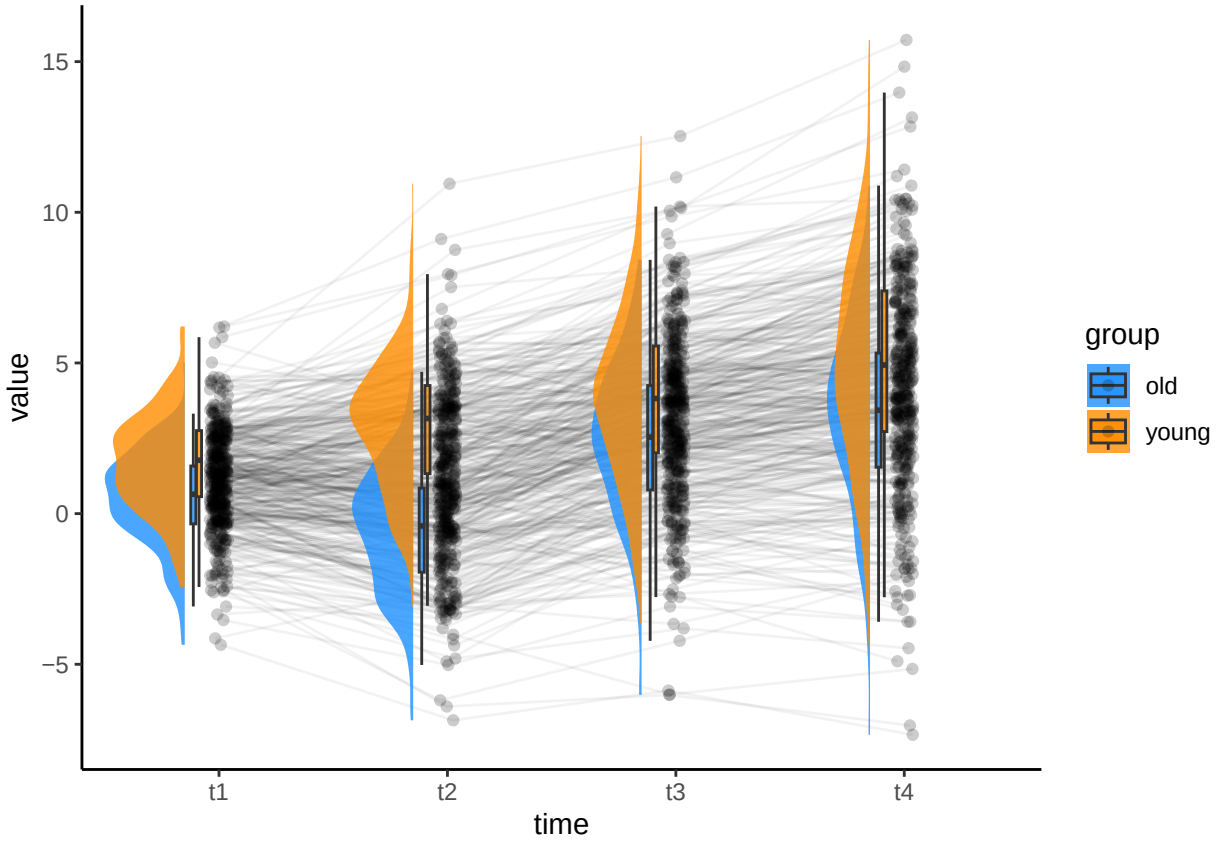
Lets connect the individuals across time! We can do this with `id.long.var`

```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(id.long.var = 'id',
            alpha = .3, rain.side = 'l',
            boxplot.args.pos = list(
              position = ggpp::position_dodgenudge(x = .1), width = 0.05
            )) +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange"))
```



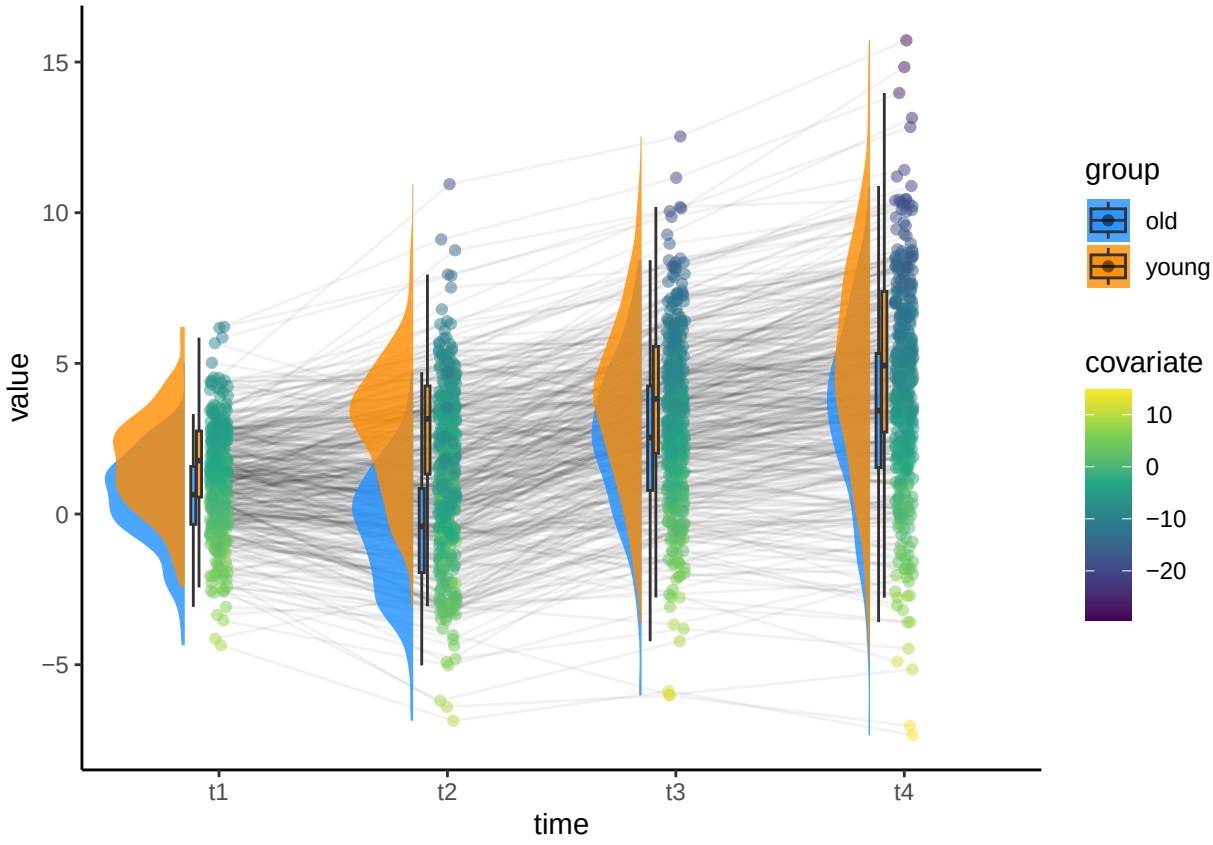
We can make some aesthetic changes to make the violins and box plot's more visible. We will do this with `boxplot.args`, `violin.args`, `line.args` and `point.args`.

```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(id.long.var = 'id', rain.side = 'l',
            boxplot.args = list(alpha = .8, outlier.shape = NA),
            violin.args = list(alpha = .8, color = NA),
            point.args = list(alpha = .2),
            line.args = list(alpha = .05),
            boxplot.args.pos = list(
              position = ggpp::position_dodgenudge(x = .1), width = 0.05
            )) +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange"))
```



Let's color the dots by a covariate and use the viridis color palette!

```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(cov = 'covariate',
            id.long.var = 'id', rain.side = 'l',
            boxplot.args = list(alpha = .8, outlier.shape = NA),
            violin.args = list(alpha = .8, color = NA),
            point.args = list(alpha = .5),
            line.args = list(alpha = .05),
            boxplot.args.pos = list(
              position = ggpp::position_dodgenudge(x = .1), width = 0.05
            )) +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange")) +
  scale_color_viridis_c()
```

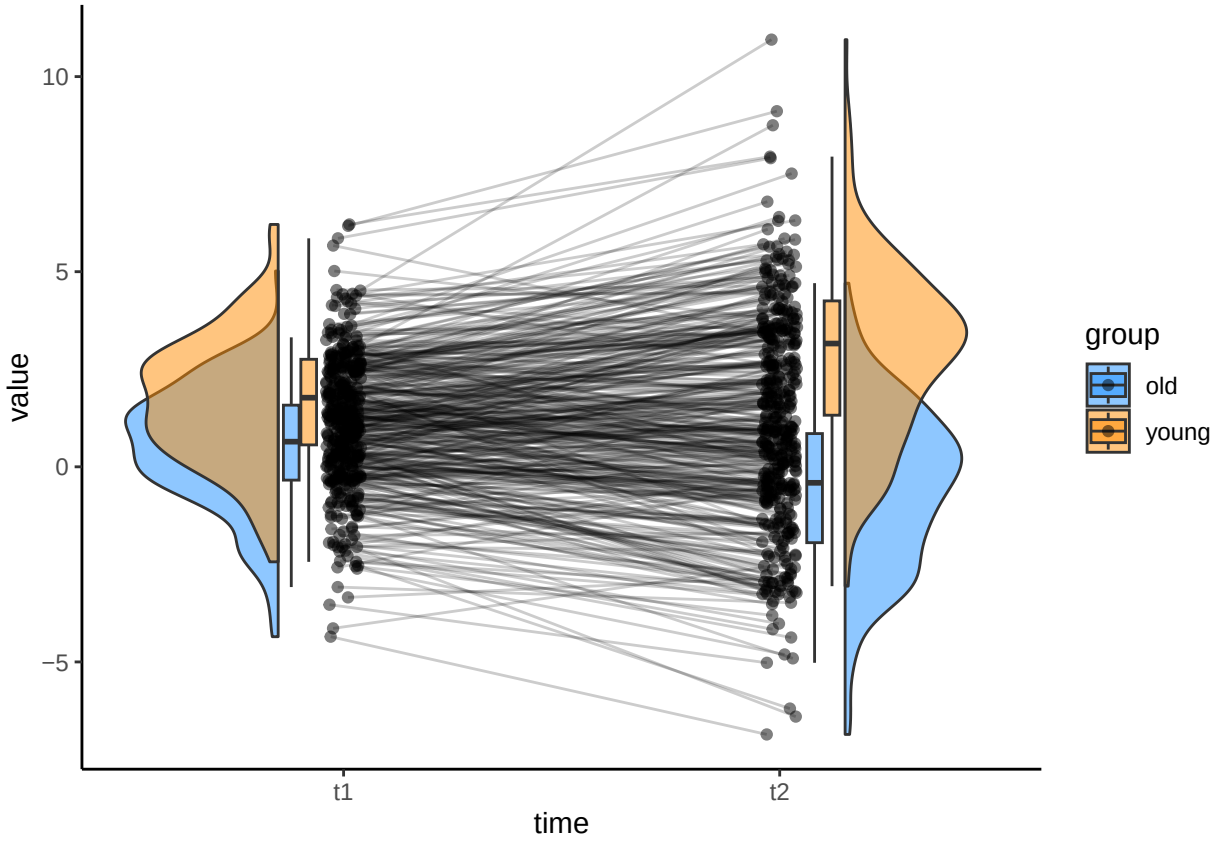


Inference with flanking

Let's take a step back and look if the groups are significantly different in the first two time points!

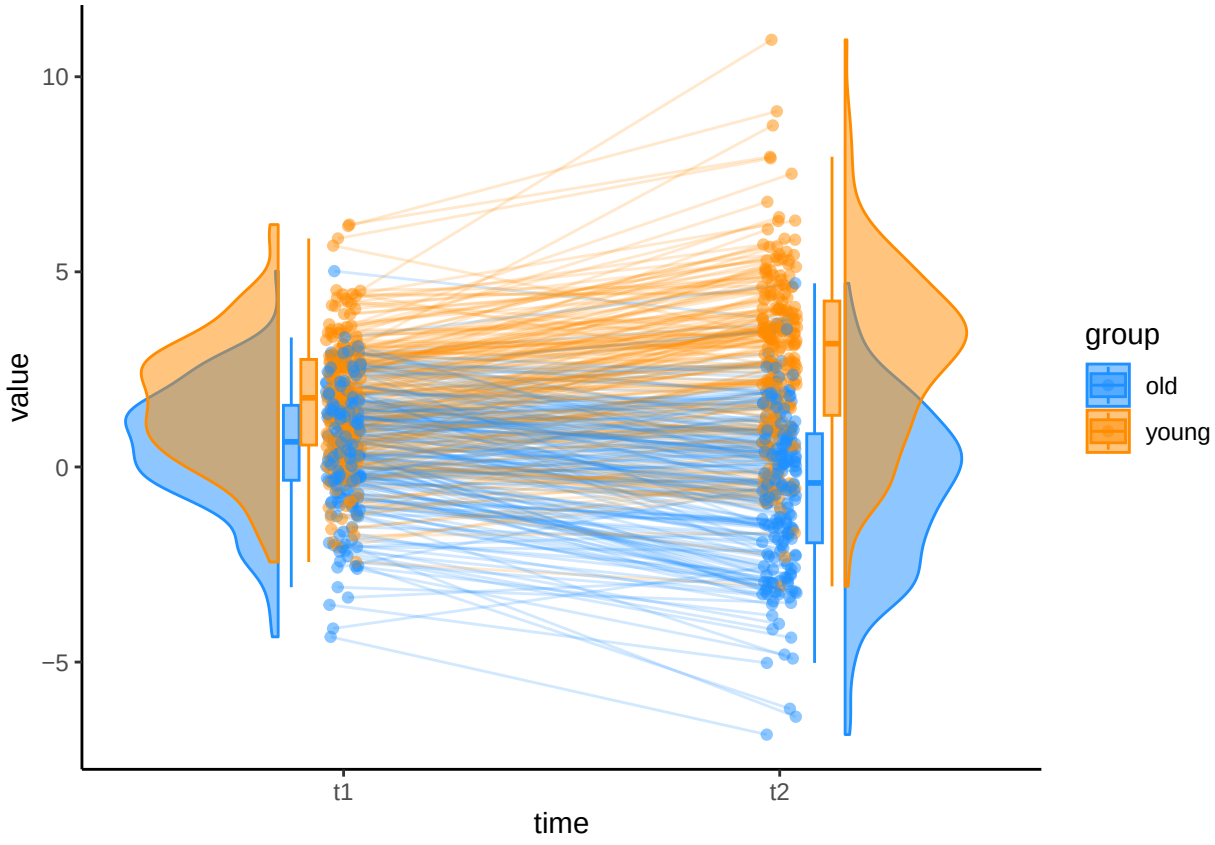
```
aging_subset <- aging[aging$time %in% c('t1', 't2'),]

ggplot(aging_subset, aes(time, value, fill = group)) +
  geom_rain(alpha = .5, rain.side = 'f', id.long.var = 'id') +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange"))
```

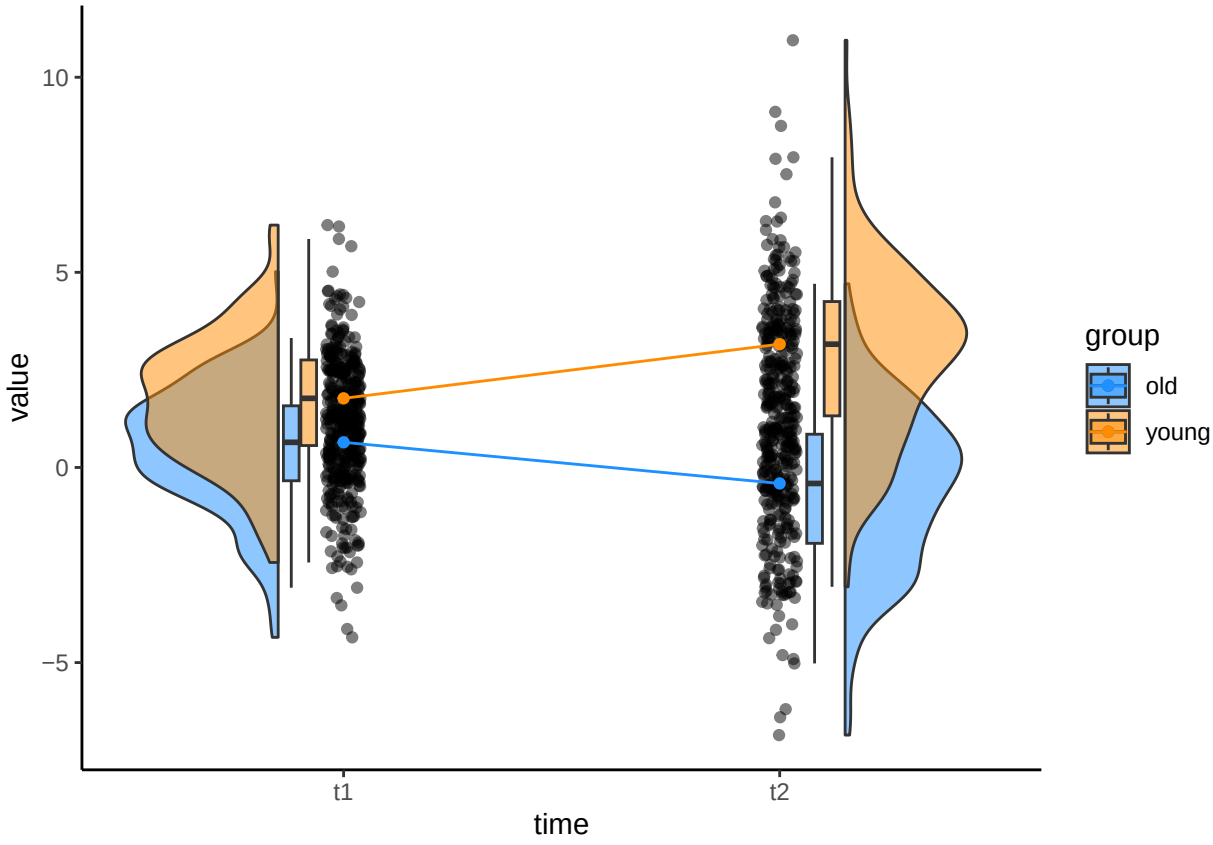


We can color the lines & dots as well!

```
ggplot(aging_subset, aes(time, value, fill = group, color = group)) +
  geom_rain(alpha = .5, rain.side = 'f', id.long.var = 'id') +
  theme_classic() +
  scale_fill_manual(values=c("dodgerblue", "darkorange")) +
  scale_color_manual(values=c("dodgerblue", "darkorange"))
```

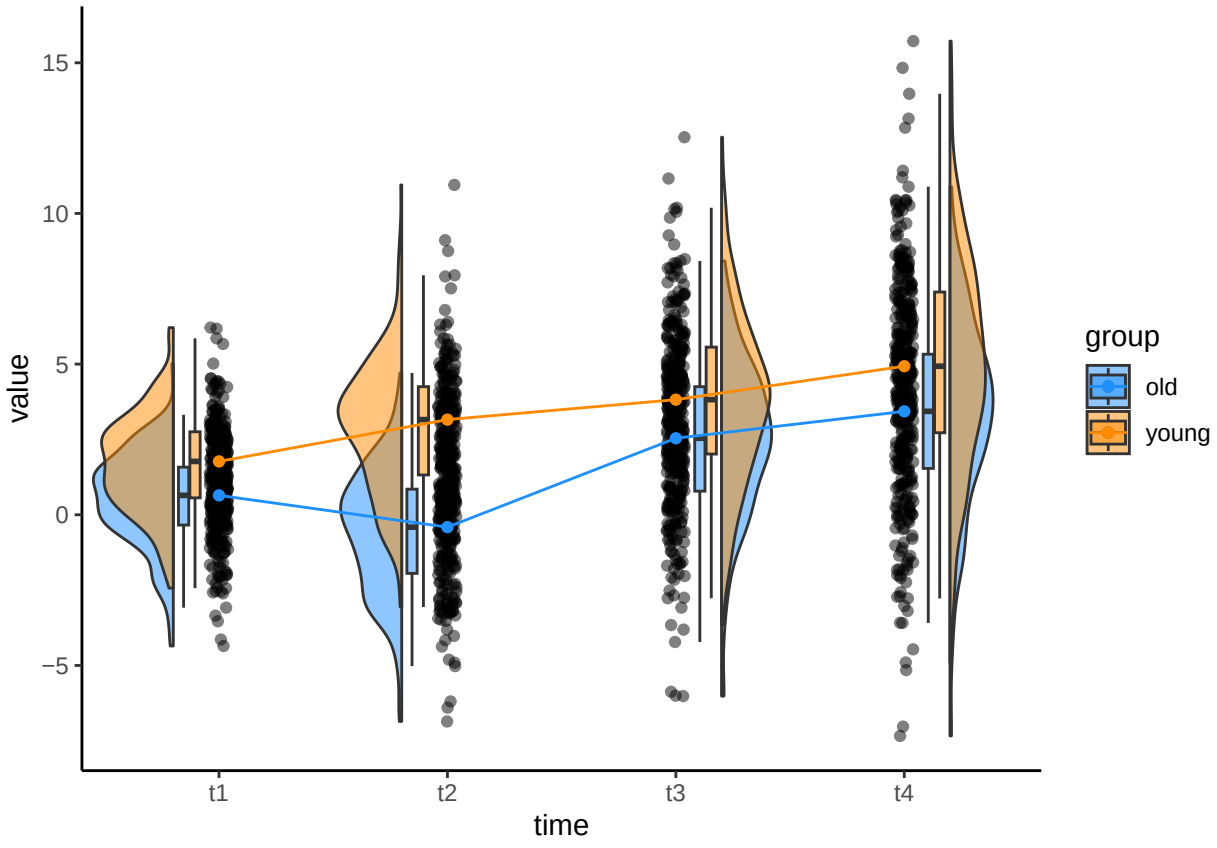


```
ggplot(aging_subset, aes(time, value, fill = group)) +
  geom_rain(alpha = .5, rain.side = 'f') +
  theme_classic() +
  stat_summary(fun = median, geom = "line", aes(group = group, color = group)) +
    stat_summary(fun = median, geom = "point",
                 aes(group = group, color = group)) +
    scale_fill_manual(values=c("dodgerblue", "darkorange")) +
    scale_colour_manual(values = c("dodgerblue", "darkorange"))
```



Advanced flanking To do advanced flanking we need to supply positioning arguments for each element/group of the boxplot & violin.

```
ggplot(aging, aes(time, value, fill = group)) +
  geom_rain(alpha = .5, rain.side = 'f',
            boxplot.args.pos = list(width = .1,
             position = ggpp::position_dodgenudge(x = c(-.13, -.13, # t1 old, t1 young
                                                        -.13, -.13,
                                                         .13, .13, # t3 old, t3 young
                                                         .13, .13))),
            violin.args.pos = list(width = .7,
             position = position_nudge(x = c(rep(-.2, 256*2), rep(-.2, 256*2),# t1
                                             rep(-.2, 256*2), rep(-.2, 256*2), # t2
                                             rep(.2, 256*2), rep(.2, 256*2),# t3
                                             rep(.2, 256*2), rep(.2, 256*2))))) +# t4
  theme_classic() +
  stat_summary(fun = median, geom = "line", aes(group = group, color = group)) +
    stat_summary(fun = median, geom = "point",
                 aes(group = group, color = group)) +
    scale_fill_manual(values=c("dodgerblue", "darkorange")) +
    scale_colour_manual(values = c("dodgerblue", "darkorange"))
```



Let's create a raincloudplot with `likert` data [IN PROGRESS]