ggbeeswarm package usage example (version 0.7.0)

Erik Clarke [aut, cre], Scott Sherrill-Mix [aut]

Abstract

This is a collection of examples of usage for the **ggbeeswarm** package.

Keywords: visualization, display, one dimensional, grouped, groups, violin, scatter, points, quasirandom, beeswarm, van der Corput, beeswarm, ggplot, ggplot2.

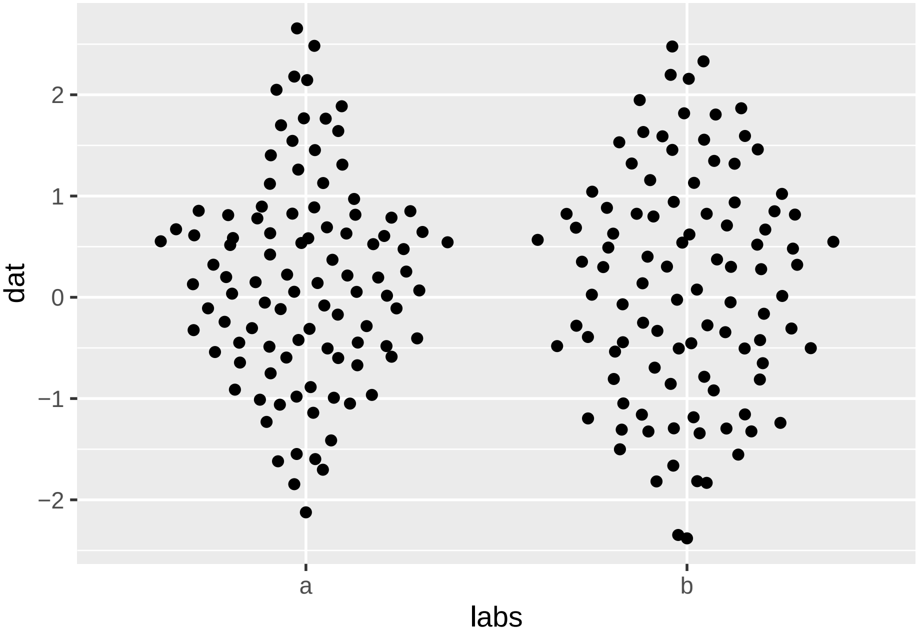
1. The basics

This is the simplest example of using `geom_quasirandom` to generate violin scatter plots:

```
>   library(ggbeeswarm)
>   set.seed(12345)
>   n<-100
>   dat<-rnorm(n*2)
>   labs<-rep(c('a','b'),n)
>   ggplot(mapping=aes(labs, dat)) + geom_quasirandom()
```

```
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0
```

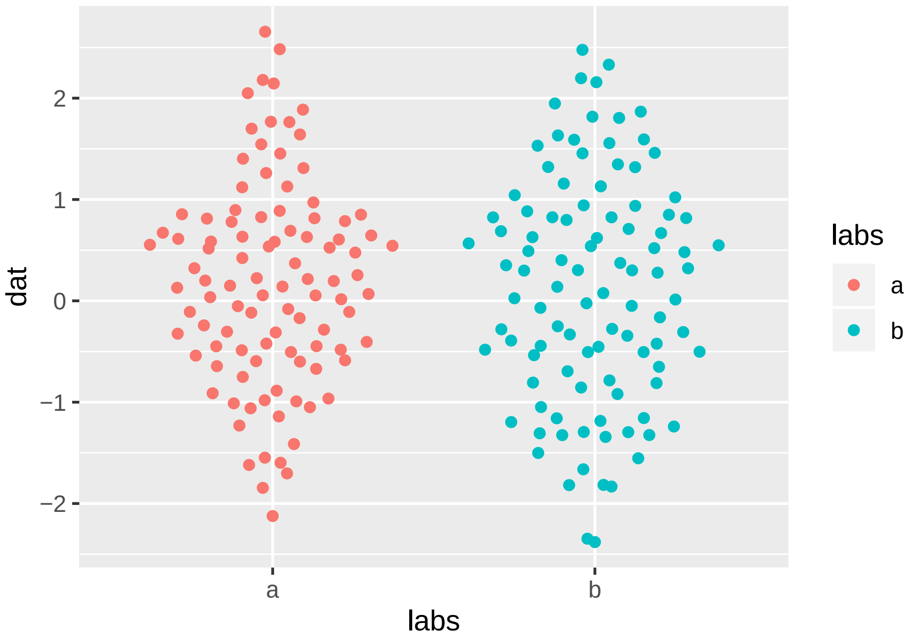
```
   x          y PANEL group
1  1  0.5855288     1     1
2  2  0.7094660     1     2
3  1 -0.1093033     1     1
4  2 -0.4534972     1     2
5  1  0.6058875     1     1
6  2 -1.8179560     1     2
7  1  0.6300986     1     1
8  2 -0.2761841     1     2
9  1 -0.2841597     1     1
10 2 -0.9193220     1     2
   x          y PANEL group xmin xmax       ymax
1  1  0.5855288     1     1    1    1  0.5855288
2  1 -0.1093033     1     1    1    1 -0.1093033
3  1  0.6058875     1     1    1    1  0.6058875
4  1  0.6300986     1     1    1    1  0.6300986
5  1 -0.2841597     1     1    1    1 -0.2841597
6  1 -0.1162478     1     1    1    1 -0.1162478
7  1  0.3706279     1     1    1    1  0.3706279
8  1 -0.7505320     1     1    1    1 -0.7505320
9  1 -0.8863575     1     1    1    1 -0.8863575
10 1  1.1207127     1     1    1    1  1.1207127
```



Normal `ggplot` options can be used:

```
>   ggplot(mapping=aes(labs, dat)) + geom_quasirandom(aes(color=labs))
```

```
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   colour x          y PANEL group
1       a 1  0.5855288     1     1
2       b 2  0.7094660     1     2
3       a 1 -0.1093033     1     1
4       b 2 -0.4534972     1     2
5       a 1  0.6058875     1     1
6       b 2 -1.8179560     1     2
7       a 1  0.6300986     1     1
8       b 2 -0.2761841     1     2
9       a 1 -0.2841597     1     1
10      b 2 -0.9193220     1     2
   colour x          y PANEL group xmin xmax       ymax
1       a 1  0.5855288     1     1    1    1  0.5855288
2       a 1 -0.1093033     1     1    1    1 -0.1093033
3       a 1  0.6058875     1     1    1    1  0.6058875
4       a 1  0.6300986     1     1    1    1  0.6300986
5       a 1 -0.2841597     1     1    1    1 -0.2841597
6       a 1 -0.1162478     1     1    1    1 -0.1162478
7       a 1  0.3706279     1     1    1    1  0.3706279
8       a 1 -0.7505320     1     1    1    1 -0.7505320
9       a 1 -0.8863575     1     1    1    1 -0.8863575
10      a 1  1.1207127     1     1    1    1  1.1207127
```

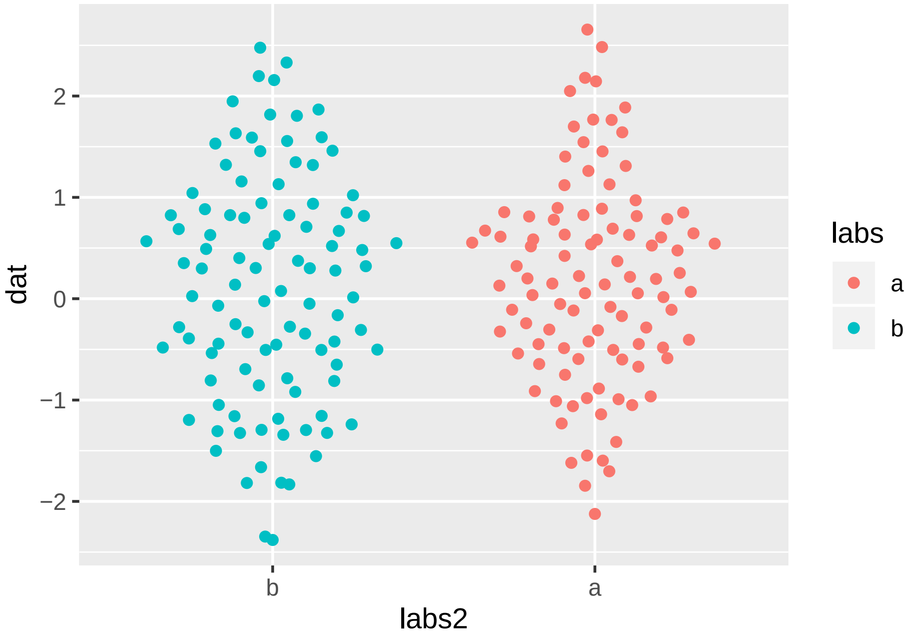


Factors can be used to generate custom group orderings:

```
>   labs2<-factor(labs,levels=c('b','a'))
>   ggplot(mapping=aes(labs2, dat)) + geom_quasirandom(aes(color=labs))
```

```
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   colour x          y PANEL group
```

```
1       a 2  0.5855288     1     1
2       b 1  0.7094660     1     2
3       a 2 -0.1093033     1     1
4       b 1 -0.4534972     1     2
5       a 2  0.6058875     1     1
6       b 1 -1.8179560     1     2
7       a 2  0.6300986     1     1
8       b 1 -0.2761841     1     2
9       a 2 -0.2841597     1     1
10      b 1 -0.9193220     1     2
   colour x          y PANEL group xmin xmax       ymax
1       b 1  0.7094660     1     2    1    1  0.7094660
2       b 1 -0.4534972     1     2    1    1 -0.4534972
3       b 1 -1.8179560     1     2    1    1 -1.8179560
4       b 1 -0.2761841     1     2    1    1 -0.2761841
5       b 1 -0.9193220     1     2    1    1 -0.9193220
6       b 1  1.8173120     1     2    1    1  1.8173120
7       b 1  0.5202165     1     2    1    1  0.5202165
8       b 1  0.8168998     1     2    1    1  0.8168998
9       b 1 -0.3315776     1     2    1    1 -0.3315776
10      b 1  0.2987237     1     2    1    1  0.2987237
```

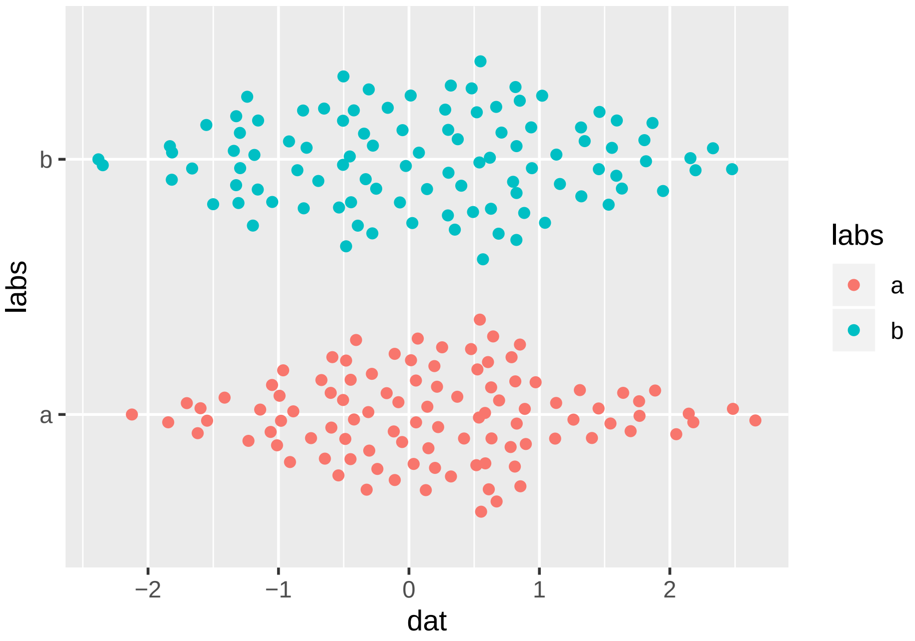


The axes can also be switched with a categorical y-axis using the argument `groupOnX=FALSE`:

```
>   ggplot(mapping=aes(dat,labs)) + geom_quasirandom(aes(color=labs),groupOnX=FALSE)
```

```
[1] 0
$width
```

```
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] FALSE

$dodge.width
[1] 0

   colour          x y PANEL group
1       a  0.5855288 1     1     1
2       b  0.7094660 2     1     2
3       a -0.1093033 1     1     1
4       b -0.4534972 2     1     2
5       a  0.6058875 1     1     1
6       b -1.8179560 2     1     2
7       a  0.6300986 1     1     1
8       b -0.2761841 2     1     2
9       a -0.2841597 1     1     1
10      b -0.9193220 2     1     2
   colour          x y PANEL group
1       a  0.5855288 1     1     1
2       a -0.1093033 1     1     1
3       a  0.6058875 1     1     1
4       a  0.6300986 1     1     1
5       a -0.2841597 1     1     1
6       a -0.1162478 1     1     1
7       a  0.3706279 1     1     1
8       a -0.7505320 1     1     1
9       a -0.8863575 1     1     1
10      a  1.1207127 1     1     1
```

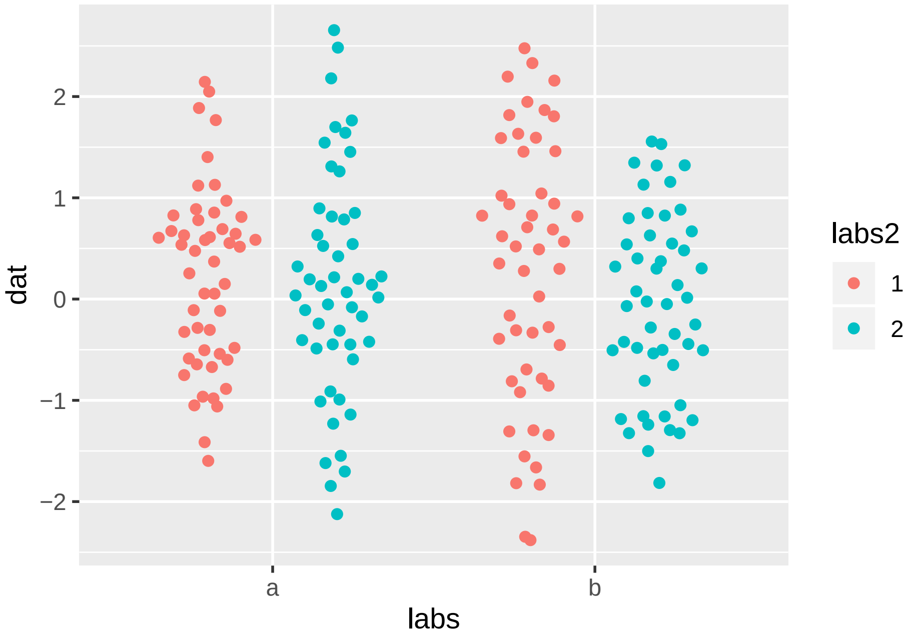


And dodging can be used to compare within groups:

```
>   labs2<-factor(rep(1:2,each=n))
>   ggplot(mapping=aes(labs,dat,color=labs2)) + geom_quasirandom(dodge.width=.8)
```

```
[1] 0.8
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0.8

   x          y colour PANEL group
```

```
1  1  0.5855288      1     1     1
2  2  0.7094660      1     1     3
3  1 -0.1093033      1     1     1
4  2 -0.4534972      1     1     3
5  1  0.6058875      1     1     1
6  2 -1.8179560      1     1     3
7  1  0.6300986      1     1     1
8  2 -0.2761841      1     1     3
9  1 -0.2841597      1     1     1
10 2 -0.9193220      1     1     3
     x           y colour PANEL group xmin xmax        ymax
1  1.2  0.22392541      2     1     2    1  1.4  0.22392541
2  1.2  0.42241853      2     1     2    1  1.4  0.42241853
3  1.2  0.14108431      2     1     2    1  1.4  0.14108431
4  1.2 -0.31160608      2     1     2    1  1.4 -0.31160608
5  1.2 -0.44803329      2     1     2    1  1.4 -0.44803329
6  1.2 -1.23017225      2     1     2    1  1.4 -1.23017225
7  1.2  1.26124227      2     1     2    1  1.4  1.26124227
8  1.2 -0.08075376      2     1     2    1  1.4 -0.08075376
9  1.2 -0.05215359      2     1     2    1  1.4 -0.05215359
10 1.2  2.18000240      2     1     2    1  1.4  2.18000240
```

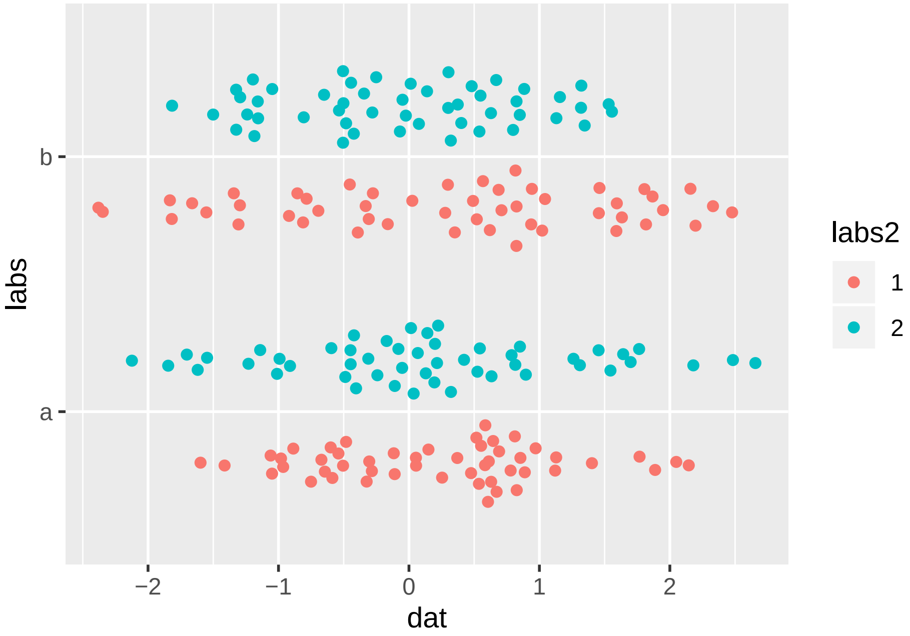


Or on the y-axis:

```
>   labs2<-factor(rep(1:2,each=n))
>   ggplot(mapping=aes(dat,labs,color=labs2)) + geom_quasirandom(dodge.width=.8,groupOnX=F
```

```
[1] 0.8
$width
```

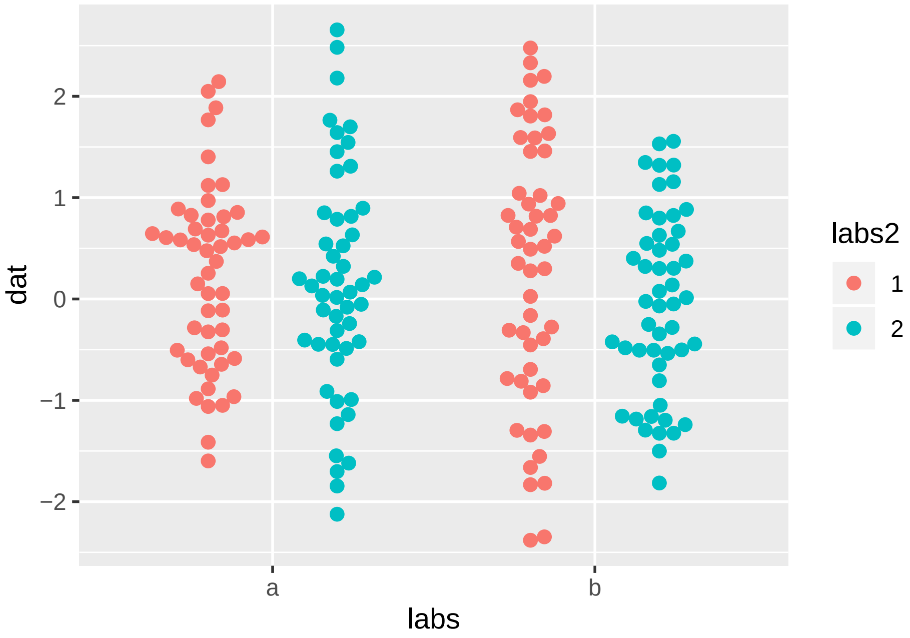
```
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] FALSE

$dodge.width
[1] 0.8

            x y colour PANEL group
1   0.5855288 1      1     1     1
2   0.7094660 2      1     1     3
3  -0.1093033 1      1     1     1
4  -0.4534972 2      1     1     3
5   0.6058875 1      1     1     1
6  -1.8179560 2      1     1     3
7   0.6300986 1      1     1     1
8  -0.2761841 2      1     1     3
9  -0.2841597 1      1     1     1
10 -0.9193220 2      1     1     3
             x   y colour PANEL group
1   0.22392541 1.2      2     1     2
2   0.42241853 1.2      2     1     2
3   0.14108431 1.2      2     1     2
4  -0.31160608 1.2      2     1     2
5  -0.44803329 1.2      2     1     2
6  -1.23017225 1.2      2     1     2
7   1.26124227 1.2      2     1     2
8  -0.08075376 1.2      2     1     2
9  -0.05215359 1.2      2     1     2
10  2.18000240 1.2      2     1     2
```



And with `geom_beeswarm`:

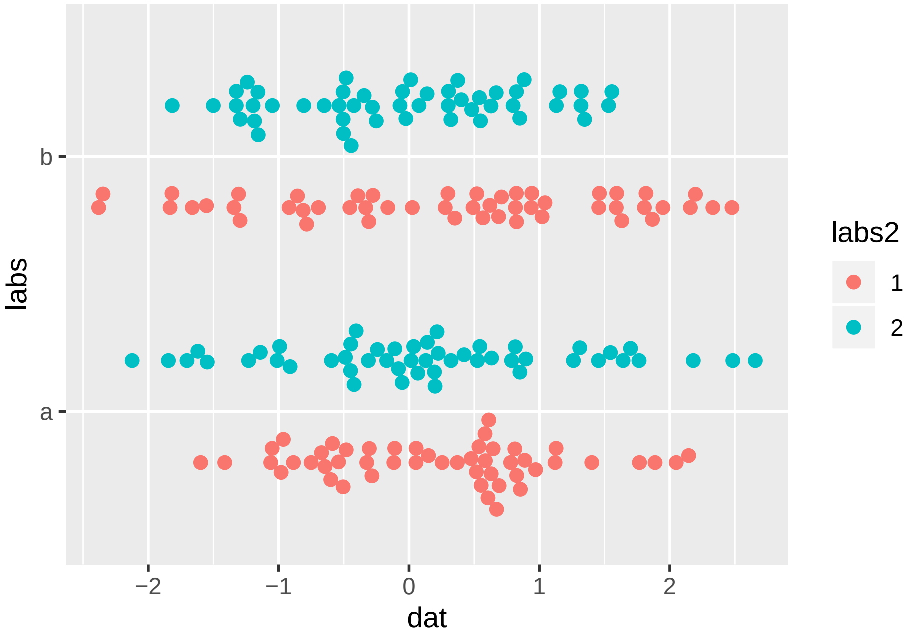
```
>   ggplot(mapping=aes(labs,dat,color=labs2)) +
+     geom_beeswarm(dodge.width=.8,cex=2)

[1] 0.8
$groupOnX
[1] TRUE

$dodge.width
[1] 0.8

   x          y colour PANEL group
1  1  0.5855288      1     1     1
2  2  0.7094660      1     1     3
3  1 -0.1093033      1     1     1
4  2 -0.4534972      1     1     3
5  1  0.6058875      1     1     1
6  2 -1.8179560      1     1     3
7  1  0.6300986      1     1     1
8  2 -0.2761841      1     1     3
9  1 -0.2841597      1     1     1
10 2 -0.9193220      1     1     3
     x           y colour PANEL group xmin xmax        ymax
1  1.2  0.22392541      2     1     2    1  1.4  0.22392541
2  1.2  0.42241853      2     1     2    1  1.4  0.42241853
3  1.2  0.14108431      2     1     2    1  1.4  0.14108431
4  1.2 -0.31160608      2     1     2    1  1.4 -0.31160608
```

```
5  1.2 -0.44803329      2     1     2    1  1.4 -0.44803329
6  1.2 -1.23017225      2     1     2    1  1.4 -1.23017225
7  1.2  1.26124227      2     1     2    1  1.4  1.26124227
8  1.2 -0.08075376      2     1     2    1  1.4 -0.08075376
9  1.2 -0.05215359      2     1     2    1  1.4 -0.05215359
10 1.2  2.18000240      2     1     2    1  1.4  2.18000240
```



```
>   ggplot(mapping=aes(dat,labs,color=labs2)) +
+     geom_beeswarm(dodge.width=.8,cex=2,groupOnX=FALSE)

[1] 0.8
$groupOnX
[1] FALSE

$dodge.width
[1] 0.8

             x y colour PANEL group
1   0.5855288 1      1     1     1
2   0.7094660 2      1     1     3
3  -0.1093033 1      1     1     1
4  -0.4534972 2      1     1     3
5   0.6058875 1      1     1     1
6  -1.8179560 2      1     1     3
7   0.6300986 1      1     1     1
8  -0.2761841 2      1     1     3
9  -0.2841597 1      1     1     1
```

```
10 -0.9193220 2      1     1     3
             x   y colour PANEL group
1   0.22392541 1.2      2     1     2
2   0.42241853 1.2      2     1     2
3   0.14108431 1.2      2     1     2
4  -0.31160608 1.2      2     1     2
5  -0.44803329 1.2      2     1     2
6  -1.23017225 1.2      2     1     2
7   1.26124227 1.2      2     1     2
8  -0.08075376 1.2      2     1     2
9  -0.05215359 1.2      2     1     2
10  2.18000240 1.2      2     1     2
```

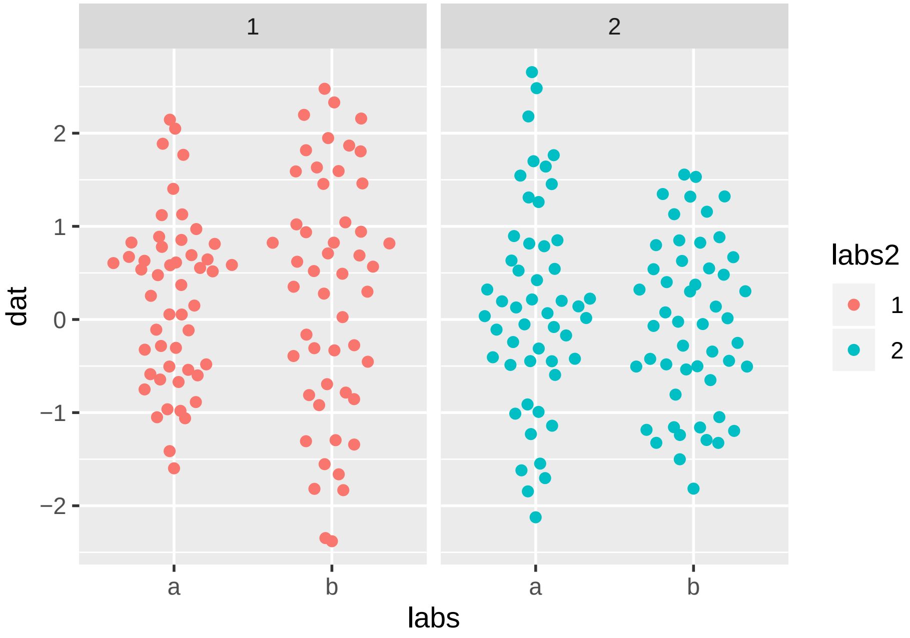


Both `geom_beeswarm` and `geom_quasirandom` also work with facets:

```
>   df<-data.frame(labs,dat,labs2)
>   ggplot(df,aes(labs,dat,color=labs2)) +
+     geom_quasirandom() +
+     facet_grid(.~labs2)

[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
```

```
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x          y colour PANEL group
1  1  0.5855288      1     1     1
2  2  0.7094660      1     1     3
3  1 -0.1093033      1     1     1
4  2 -0.4534972      1     1     3
5  1  0.6058875      1     1     1
6  2 -1.8179560      1     1     3
7  1  0.6300986      1     1     1
8  2 -0.2761841      1     1     3
9  1 -0.2841597      1     1     1
10 2 -0.9193220      1     1     3
   x          y colour PANEL group xmin xmax       ymax
1  1  0.5855288      1     1     1    1    1  0.5855288
2  1 -0.1093033      1     1     1    1    1 -0.1093033
3  1  0.6058875      1     1     1    1    1  0.6058875
4  1  0.6300986      1     1     1    1    1  0.6300986
5  1 -0.2841597      1     1     1    1    1 -0.2841597
6  1 -0.1162478      1     1     1    1    1 -0.1162478
7  1  0.3706279      1     1     1    1    1  0.3706279
8  1 -0.7505320      1     1     1    1    1 -0.7505320
9  1 -0.8863575      1     1     1    1    1 -0.8863575
10 1  1.1207127      1     1     1    1    1  1.1207127
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL
```

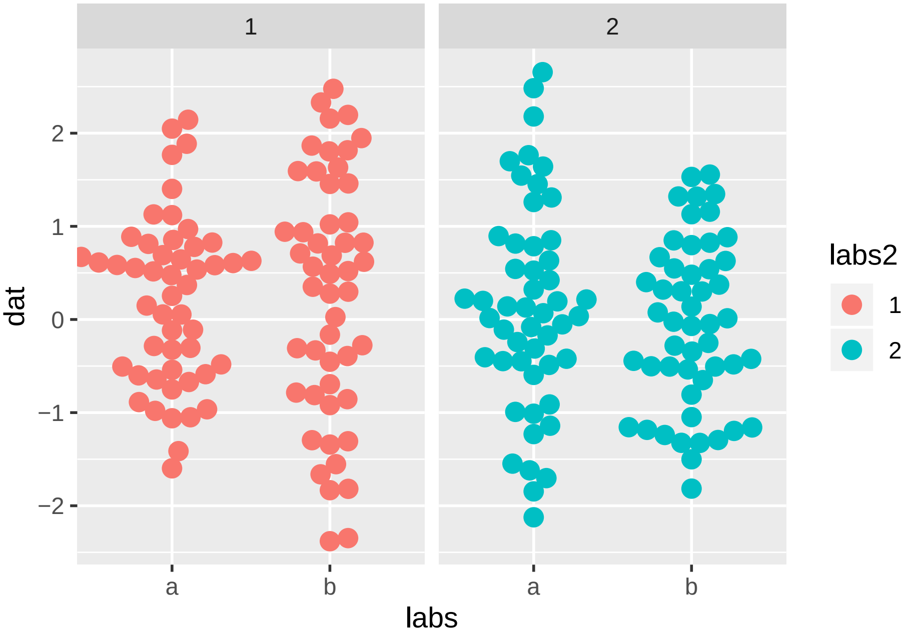
```
$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x          y colour PANEL group
1  1  0.2239254      2     2     2
2  2 -1.1562233      2     2     4
3  1  0.4224185      2     2     2
4  2 -1.3247553      2     2     4
5  1  0.1410843      2     2     2
6  2 -0.5360480      2     2     4
7  1 -0.3116061      2     2     2
8  2  1.5561096      2     2     4
9  1 -0.4480333      2     2     2
10 2  0.3211235      2     2     4
   x           y colour PANEL group xmin xmax        ymax
1  1  0.22392541      2     2     2    1    1  0.22392541
2  1  0.42241853      2     2     2    1    1  0.42241853
3  1  0.14108431      2     2     2    1    1  0.14108431
4  1 -0.31160608      2     2     2    1    1 -0.31160608
5  1 -0.44803329      2     2     2    1    1 -0.44803329
6  1 -1.23017225      2     2     2    1    1 -1.23017225
7  1  1.26124227      2     2     2    1    1  1.26124227
8  1 -0.08075376      2     2     2    1    1 -0.08075376
9  1 -0.05215359      2     2     2    1    1 -0.05215359
10 1  2.18000240      2     2     2    1    1  2.18000240
```



```
>   ggplot(df,aes(labs,dat,color=labs2)) +
+     geom_beeswarm(cex=3) +
+     facet_grid(.~labs2)
```

```
[1] 0
$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x          y colour PANEL group
1  1  0.5855288      1     1     1
2  2  0.7094660      1     1     3
3  1 -0.1093033      1     1     1
4  2 -0.4534972      1     1     3
5  1  0.6058875      1     1     1
6  2 -1.8179560      1     1     3
7  1  0.6300986      1     1     1
8  2 -0.2761841      1     1     3
9  1 -0.2841597      1     1     1
10 2 -0.9193220      1     1     3
   x          y colour PANEL group xmin xmax       ymax
1  1  0.5855288      1     1     1    1    1  0.5855288
2  1 -0.1093033      1     1     1    1    1 -0.1093033
3  1  0.6058875      1     1     1    1    1  0.6058875
4  1  0.6300986      1     1     1    1    1  0.6300986
```

```
5  1 -0.2841597      1     1     1    1    1 -0.2841597
6  1 -0.1162478      1     1     1    1    1 -0.1162478
7  1  0.3706279      1     1     1    1    1  0.3706279
8  1 -0.7505320      1     1     1    1    1 -0.7505320
9  1 -0.8863575      1     1     1    1    1 -0.8863575
10 1  1.1207127      1     1     1    1    1  1.1207127
$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x          y colour PANEL group
1  1  0.2239254      2     2     2
2  2 -1.1562233      2     2     4
3  1  0.4224185      2     2     2
4  2 -1.3247553      2     2     4
5  1  0.1410843      2     2     2
6  2 -0.5360480      2     2     4
7  1 -0.3116061      2     2     2
8  2  1.5561096      2     2     4
9  1 -0.4480333      2     2     2
10 2  0.3211235      2     2     4
   x           y colour PANEL group xmin xmax        ymax
1  1  0.22392541      2     2     2    1    1  0.22392541
2  1  0.42241853      2     2     2    1    1  0.42241853
3  1  0.14108431      2     2     2    1    1  0.14108431
4  1 -0.31160608      2     2     2    1    1 -0.31160608
5  1 -0.44803329      2     2     2    1    1 -0.44803329
6  1 -1.23017225      2     2     2    1    1 -1.23017225
7  1  1.26124227      2     2     2    1    1  1.26124227
8  1 -0.08075376      2     2     2    1    1 -0.08075376
9  1 -0.05215359      2     2     2    1    1 -0.05215359
10 1  2.18000240      2     2     2    1    1  2.18000240
```



2. Options

There are several ways to plot grouped one-dimensional data combining points and density estimation including:

pseudorandom The kernel density is estimated then points are distributed uniform randomly within the density estimate for a given bin. Selection of an appropriate number of bins does not greatly affect appearance but coincidental clumpiness is common.

alternating within bins The kernel density is estimated then points are distributed within the density estimate for a given bin evenly spaced with extreme values alternating from right to left e.g. max, 3rd max, ..., 4th max, 2nd max. If maximums are placed on the outside then these plots often form consecutive "smiley" patterns. If minimums are placed on the outside then "frowny" patterns are generated. Selection of the number of bins can have large effects on appearance important.

tukey An algorithm described by Tukey and Tukey in "Strips displaying empirical distributions: I. textured dot strips" using constrained permutations of offsets to distrbute the data.

quasirandom The kernel density is estimated then points are distributed quasirandomly using the von der Corput sequence within the density estimate for a given bin. Selection of an appropriate number of bins does not greatly affect appearance and position does not depend on plotting parameters.

beeswarm The package **beeswarm** provides methods for generating a "beeswarm" plot where points are distibuted so that no points overlap. Kernel density is not calculated although

> the resulting plot does provide an approximate density estimate. Selection of an appropriate number of bins affects appearance and plot and point sizes must be known in advance.

The first four options are included within `geom_quasirandom` using the `method=` argument and beeswarm plots are generated with `geom_beeswarm`:

```
>   library(gridExtra)
>   dat <- list(
+     'Normal'=rnorm(50),
+     'Dense normal'= rnorm(500),
+     'Bimodal'=c(rnorm(100), rnorm(100,5)),
+     'Trimodal'=c(rnorm(100), rnorm(100,5),rnorm(100,-3))
+   )
>   labs<-rep(names(dat),sapply(dat,length))
>   labs<-factor(labs,levels=unique(labs))
>   dat<-unlist(dat)
>   p1<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(alpha=.2) +
+     ggtitle('quasirandom') + labs(x='') +
+     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
>   p2<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='pseudorandom',alpha=.2) +
+     ggtitle('pseudorandom') + labs(x='') +
+     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
>   p3<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='smiley',alpha=.2) +
+     ggtitle('smiley') + labs(x='') +
+     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
>   p4<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='frowney',alpha=.2) +
+     ggtitle('frowney') + labs(x='') +
+     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
>   p5<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='tukey',alpha=.2) +
+     ggtitle('tukey') + labs(x='') +
+     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
>   p6<-ggplot(mapping=aes(labs, dat)) +
+     geom_beeswarm(alpha=.2,size=.75) +
+     ggtitle('geom_beeswarm') + labs(x='') +
+     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
>   grid.arrange(p1, p2, p3, p4, p5, p6, ncol=3)
```

```
[1] 0
$width
NULL
```

```
$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE
```

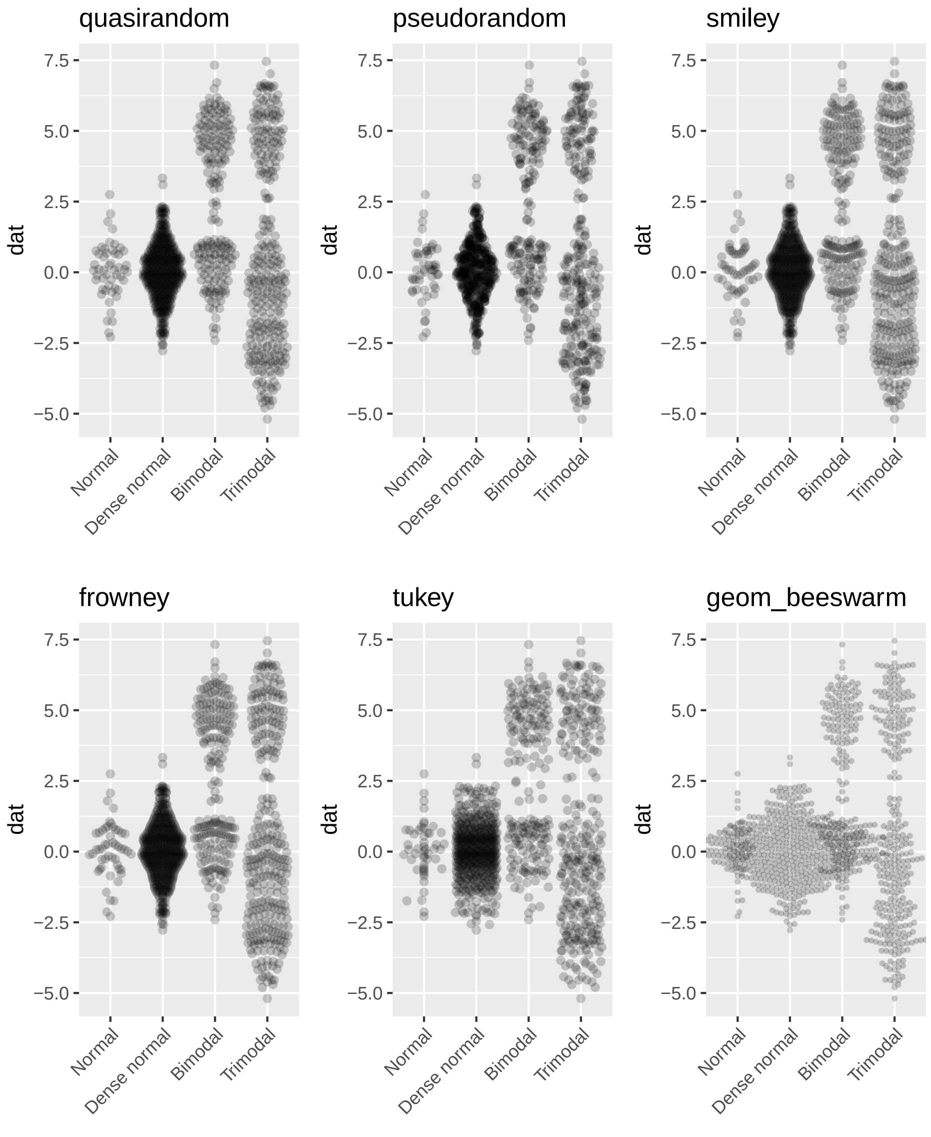
```
$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "pseudorandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5
```

```
$nbins
NULL

$method
[1] "smiley"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL
```

```
$method
[1] "frowney"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "tukey"
```

```
$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
```

```
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
```



`quasirandom` calls `vipor::offsetX` which calls `stats::density` to compute kernel density estimates. The tightness of the fit can be adjusted with the `bandwidth` option and the width of the offset with `width`. `nbins` to adjust the number of bins used in the kernel density is also provided but this can usually be left at its default when using quasirandom offsets but is useful for non-quasirandom methods:

```
>   library(gridExtra)
>   p1<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(bandwidth=2,alpha=.2) +
+     ggtitle('bandwidth=2') + labs(x='')
>   p2<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(bandwidth=.1,alpha=.2) +
+     ggtitle('bandwidth=.1') + labs(x='')
>   p3<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(width=.1,alpha=.2) +
+     ggtitle('width=.1') + labs(x='')
>   p4<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(nbins=100,alpha=.2) +
+     ggtitle('nbins=100') + labs(x='')
>   grid.arrange(p1, p2, p3, p4, ncol=1)
```

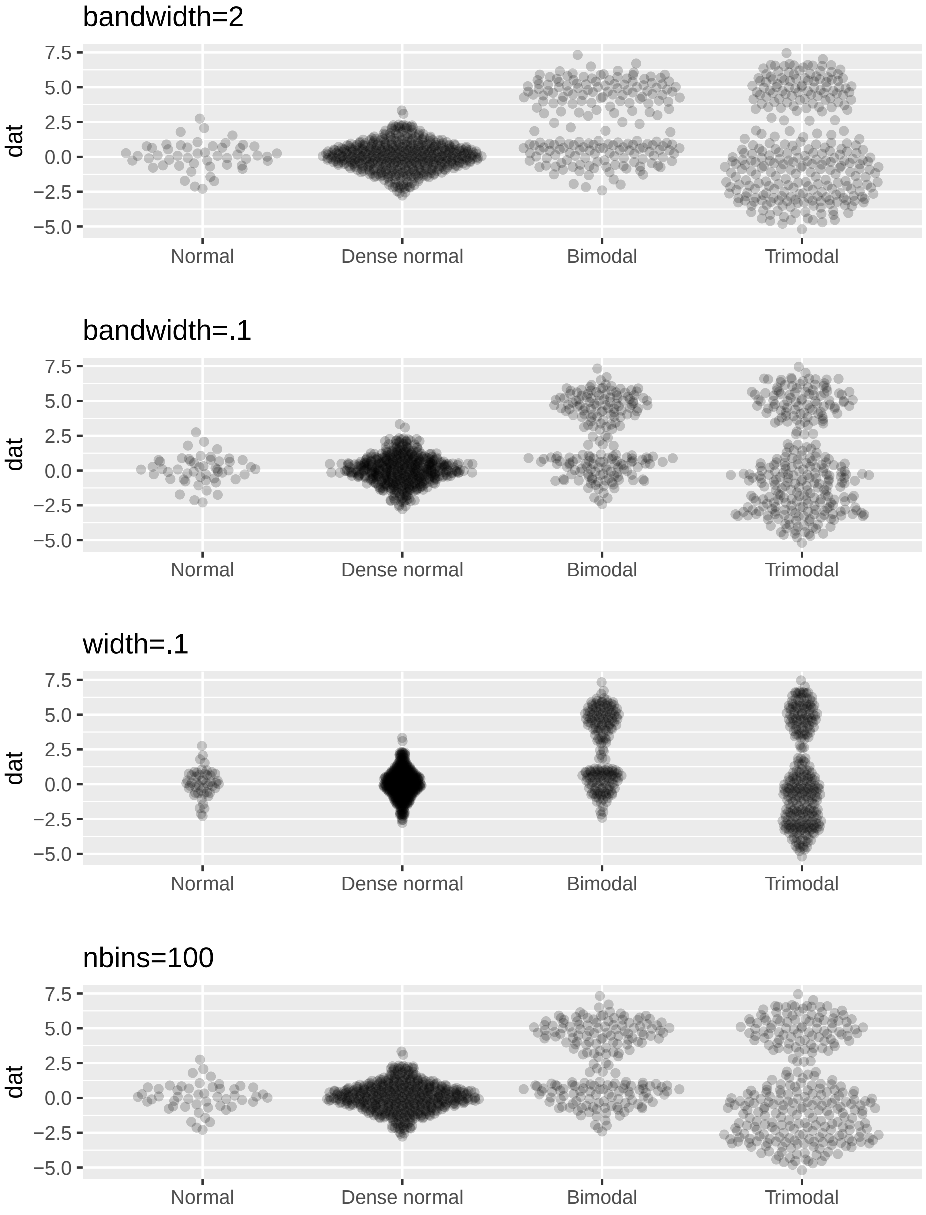
```
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 2

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
```

```
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.1

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
```

```
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
[1] 0.1

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
```

```
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
[1] 100

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
```

```
5  1  0.83134882      1      1     1     1  0.83134882
6  1  0.10521182      1      1     1     1  0.10521182
7  1 -1.74171309      1      1     1     1 -1.74171309
8  1  0.64524699      1      1     1     1  0.64524699
9  1  0.09710422      1      1     1     1  0.09710422
10 1 -0.07673369      1      1     1     1 -0.07673369
```



The `frowney` or `smiley` methods are sensitive to the number of bins so the argument `nbins` is more useful/necessary with them:

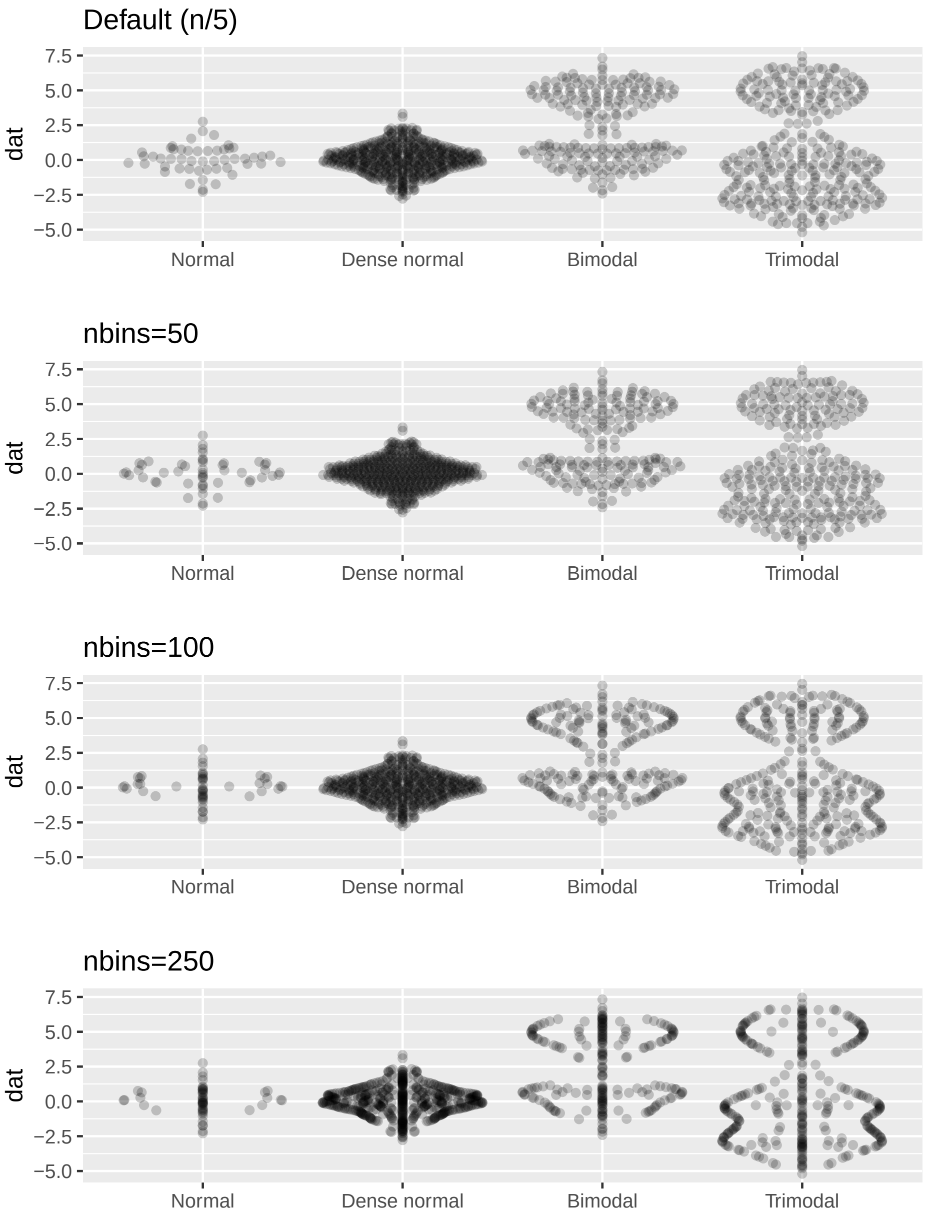
```
>   p1<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='smiley',alpha=.2) +
+     ggtitle('Default (n/5)') + labs(x='')
>   p2<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='smiley',nbins=50,alpha=.2) +
+     ggtitle('nbins=50') + labs(x='')
>   p3<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='smiley',nbins=100,alpha=.2) +
+     ggtitle('nbins=100') + labs(x='')
>   p4<-ggplot(mapping=aes(labs, dat)) +
+     geom_quasirandom(method='smiley',nbins=250,alpha=.2) +
+     ggtitle('nbins=250') + labs(x='')
>   grid.arrange(p1, p2, p3, p4, ncol=1)
```

```
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "smiley"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
```

```
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
[1] 50

$method
[1] "smiley"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
```

```
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
[1] 100

$method
[1] "smiley"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
```

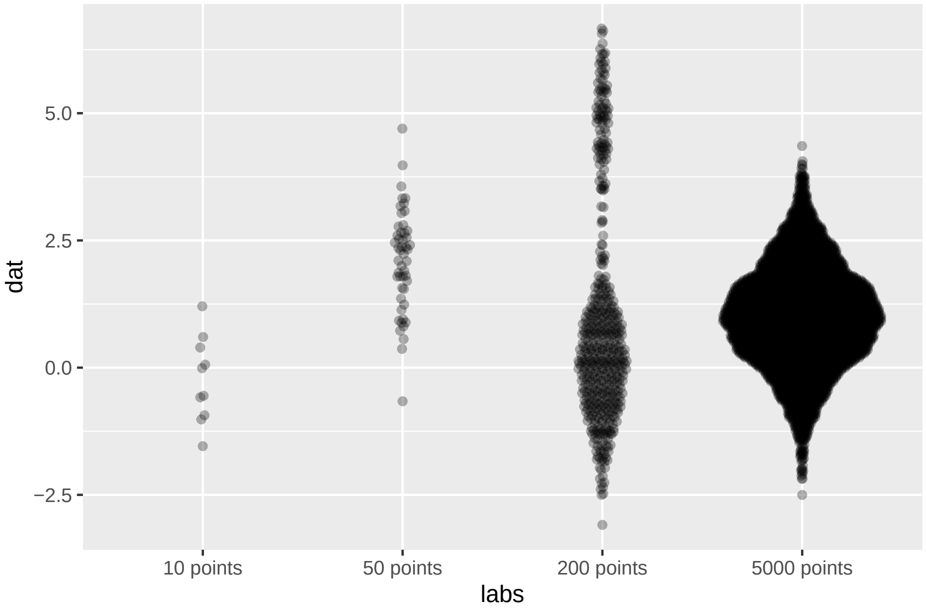
```
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
[1] 250

$method
[1] "smiley"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x           y PANEL group
1  1 -1.43614571     1     1
2  1 -0.62925965     1     1
3  1  0.24352177     1     1
4  1  1.05836223     1     1
5  1  0.83134882     1     1
6  1  0.10521182     1     1
7  1 -1.74171309     1     1
8  1  0.64524699     1     1
9  1  0.09710422     1     1
10 1 -0.07673369     1     1
   x           y PANEL group xmin xmax        ymax
1  1 -1.43614571     1     1    1    1 -1.43614571
2  1 -0.62925965     1     1    1    1 -0.62925965
3  1  0.24352177     1     1    1    1  0.24352177
4  1  1.05836223     1     1    1    1  1.05836223
5  1  0.83134882     1     1    1    1  0.83134882
6  1  0.10521182     1     1    1    1  0.10521182
7  1 -1.74171309     1     1    1    1 -1.74171309
8  1  0.64524699     1     1    1    1  0.64524699
9  1  0.09710422     1     1    1    1  0.09710422
10 1 -0.07673369     1     1    1    1 -0.07673369
```



The `varwidth` argument scales the width of a group by the square root of the number of observations in that group (as in the function `boxplot`):

```
>   dat <- list(
+     '10 points'=rnorm(10),
+     '50 points'=rnorm(50,2),
+     '200 points'=c(rnorm(400), rnorm(100,5)),
+     '5000 points'= rnorm(5000,1)
+   )
>   labs<-rep(names(dat),sapply(dat,length))
>   labs<-factor(labs,levels=unique(labs))
>  dat<-unlist(dat)
>  ggplot(mapping=aes(labs, dat)) + geom_quasirandom(alpha=.3,varwidth=TRUE)
```

```
[1] 0
$width
NULL

$varwidth
[1] TRUE

$bandwidth
[1] 0.5

$nbins
NULL

$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x            y PANEL group
1  1 -1.017831865     1     1
2  1  0.602702761     1     1
3  1 -0.935592691     1     1
4  1  1.205344164     1     1
5  1 -0.582575513     1     1
6  1  0.058723259     1     1
7  1 -0.009744727     1     1
8  1  0.395972632     1     1
9  1 -1.542892653     1     1
10 1 -0.552577137     1     1
   x            y PANEL group xmin xmax         ymax
1  1 -1.017831865     1     1    1    1 -1.017831865
2  1  0.602702761     1     1    1    1  0.602702761
3  1 -0.935592691     1     1    1    1 -0.935592691
4  1  1.205344164     1     1    1    1  1.205344164
5  1 -0.582575513     1     1    1    1 -0.582575513
6  1  0.058723259     1     1    1    1  0.058723259
7  1 -0.009744727     1     1    1    1 -0.009744727
8  1  0.395972632     1     1    1    1  0.395972632
9  1 -1.542892653     1     1    1    1 -1.542892653
10 1 -0.552577137     1     1    1    1 -0.552577137
```



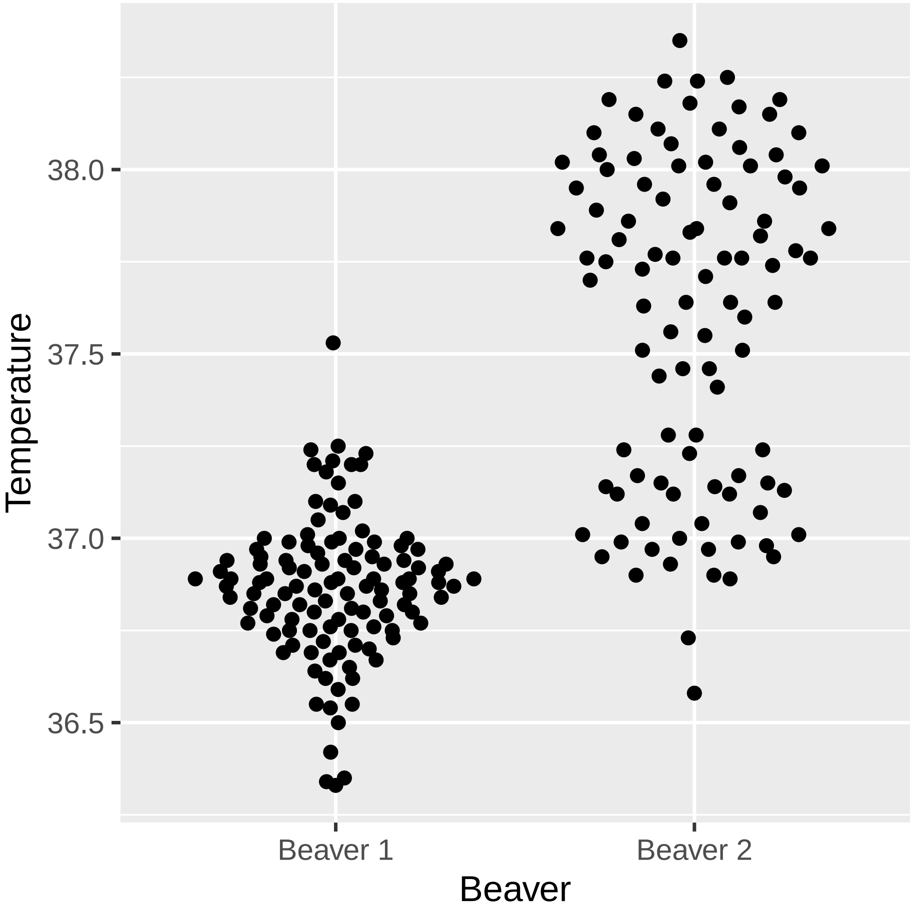
3. Real data

An example using the `beaver1` and `beaver2` data from the **datasets** package:

```
> beaver<-data.frame(
+   'Temperature'=c(beaver1$temp,beaver2$temp),
+   'Beaver'=rep(
+     c('Beaver 1','Beaver 2'),
+     c(nrow(beaver1),nrow(beaver2))
+   )
+ )
> ggplot(beaver,mapping=aes(Beaver, Temperature)) + geom_quasirandom()
```

```
[1] 0
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL

$method
```

```
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0

   x     y PANEL group
1  1 36.33     1     1
2  1 36.34     1     1
3  1 36.35     1     1
4  1 36.42     1     1
5  1 36.55     1     1
6  1 36.69     1     1
7  1 36.71     1     1
8  1 36.75     1     1
9  1 36.81     1     1
10 1 36.88     1     1
   x     y PANEL group xmin xmax  ymax
1  1 36.33     1     1    1    1 36.33
2  1 36.34     1     1    1    1 36.34
3  1 36.35     1     1    1    1 36.35
4  1 36.42     1     1    1    1 36.42
5  1 36.55     1     1    1    1 36.55
6  1 36.69     1     1    1    1 36.69
7  1 36.71     1     1    1    1 36.71
8  1 36.75     1     1    1    1 36.75
9  1 36.81     1     1    1    1 36.81
10 1 36.88     1     1    1    1 36.88
```

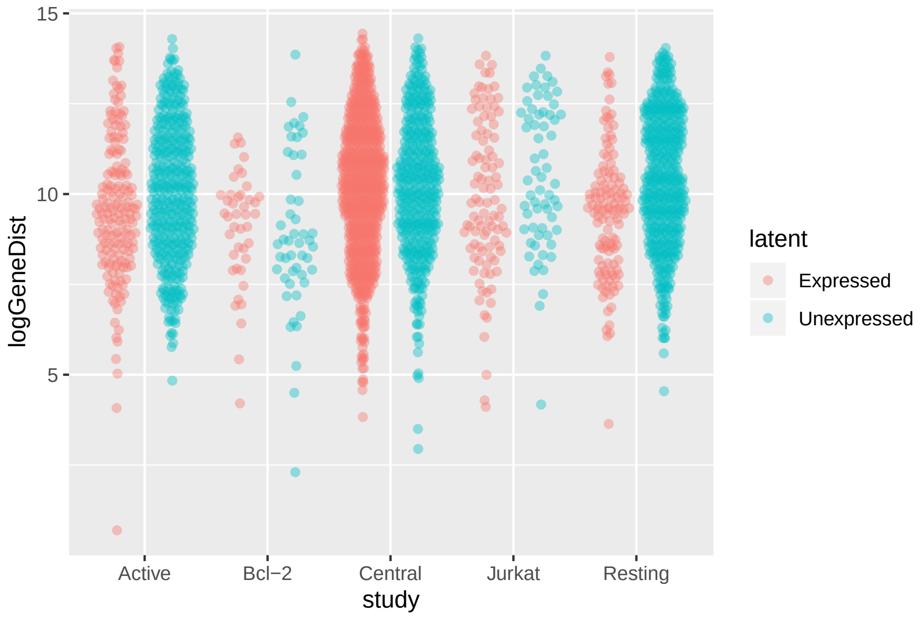


An example using the `integrations` data from the **vipor** package and the argument `dodge.width`:

```
>   library(vipor)
>   ints<-integrations[integrations$nearestGene>0,]
>   ints$logGeneDist<-log(ints$nearestGene)
>   ggplot(ints,mapping=aes(study, logGeneDist,color=latent)) +
+   geom_quasirandom(dodge.width=.9,alpha=.4)
```

```
[1] 0.9
$width
NULL

$varwidth
[1] FALSE

$bandwidth
[1] 0.5

$nbins
NULL
```

```
$method
[1] "quasirandom"

$groupOnX
[1] TRUE

$dodge.width
[1] 0.9

   x         y      colour PANEL group
1  3  8.669227 Unexpressed     1     6
2  1  7.370860 Unexpressed     1     2
3  5  7.143618   Expressed     1     9
4  5  8.974238 Unexpressed     1    10
5  3  6.920672 Unexpressed     1     6
6  3  6.385194 Unexpressed     1     6
7  3  8.035603 Unexpressed     1     6
8  3 10.528142 Unexpressed     1     6
9  1  8.108924 Unexpressed     1     2
10 3  9.641603   Expressed     1     5
       x         y      colour PANEL group xmin xmax
1  1.225  7.370860 Unexpressed     1     2    1 1.45
2  1.225  8.108924 Unexpressed     1     2    1 1.45
3  1.225  7.804659 Unexpressed     1     2    1 1.45
4  1.225 10.831746 Unexpressed     1     2    1 1.45
5  1.225  9.172639 Unexpressed     1     2    1 1.45
6  1.225 10.191032 Unexpressed     1     2    1 1.45
7  1.225 13.013653 Unexpressed     1     2    1 1.45
8  1.225 12.160547 Unexpressed     1     2    1 1.45
9  1.225 10.291196 Unexpressed     1     2    1 1.45
10 1.225 10.672855 Unexpressed     1     2    1 1.45
        ymax
1   7.370860
2   8.108924
3   7.804659
4  10.831746
5   9.172639
6  10.191032
7  13.013653
8  12.160547
9  10.291196
10 10.672855
```



Affiliation:

Github: http://github.com/eclarke/ggbeeswarm
Cran: https://cran.r-project.org/package=ggbeeswarm