

ggbeeswarm package usage example (version 0.5.0)

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

Abstract

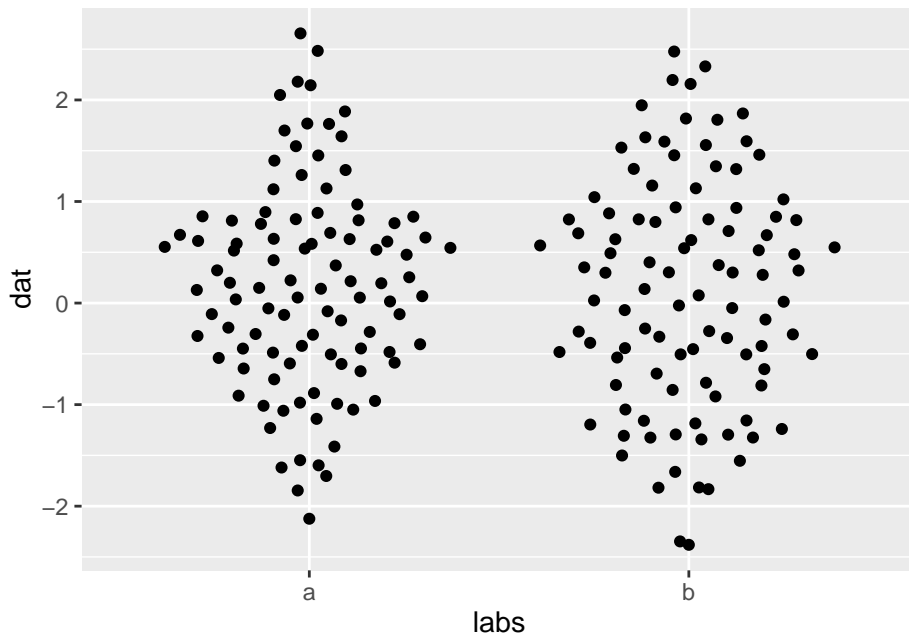
This is a collection of examples of usage for the **ggbeeswarm** package.

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

1. The basics

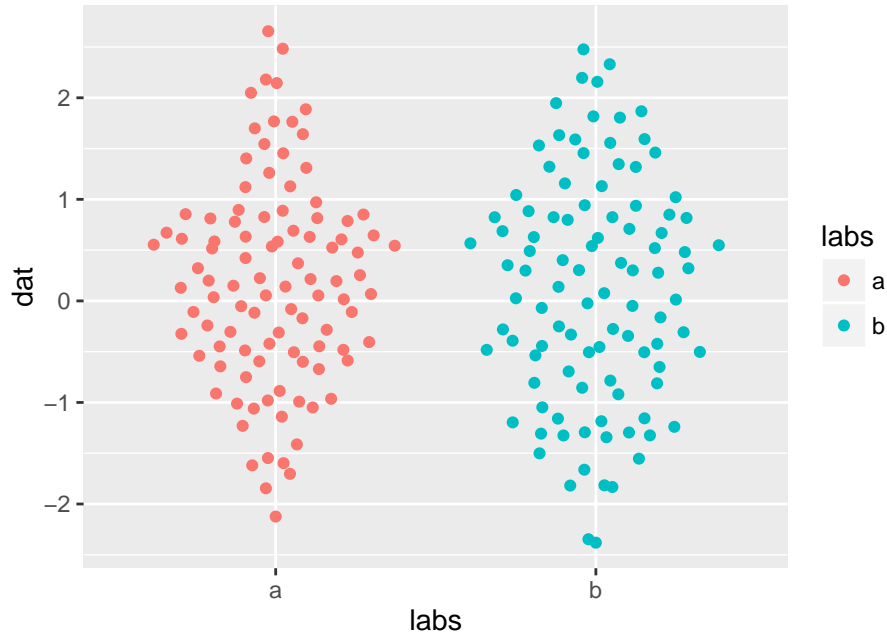
This is the simplest example of using `geom_quasirandom` to generate violin scatter plots:

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



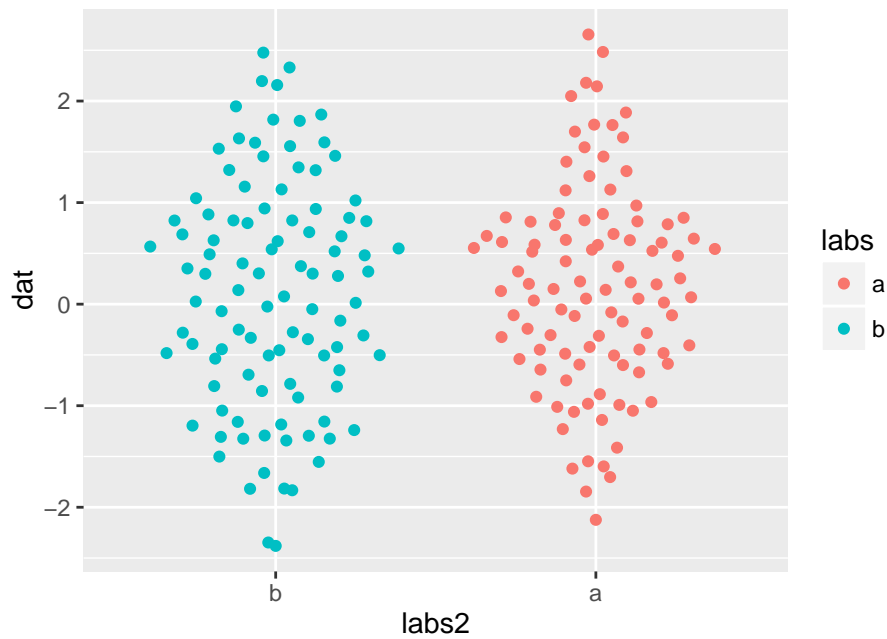
Normal **ggplot** options can be used:

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



Factors can be used to generate custom group orderings:

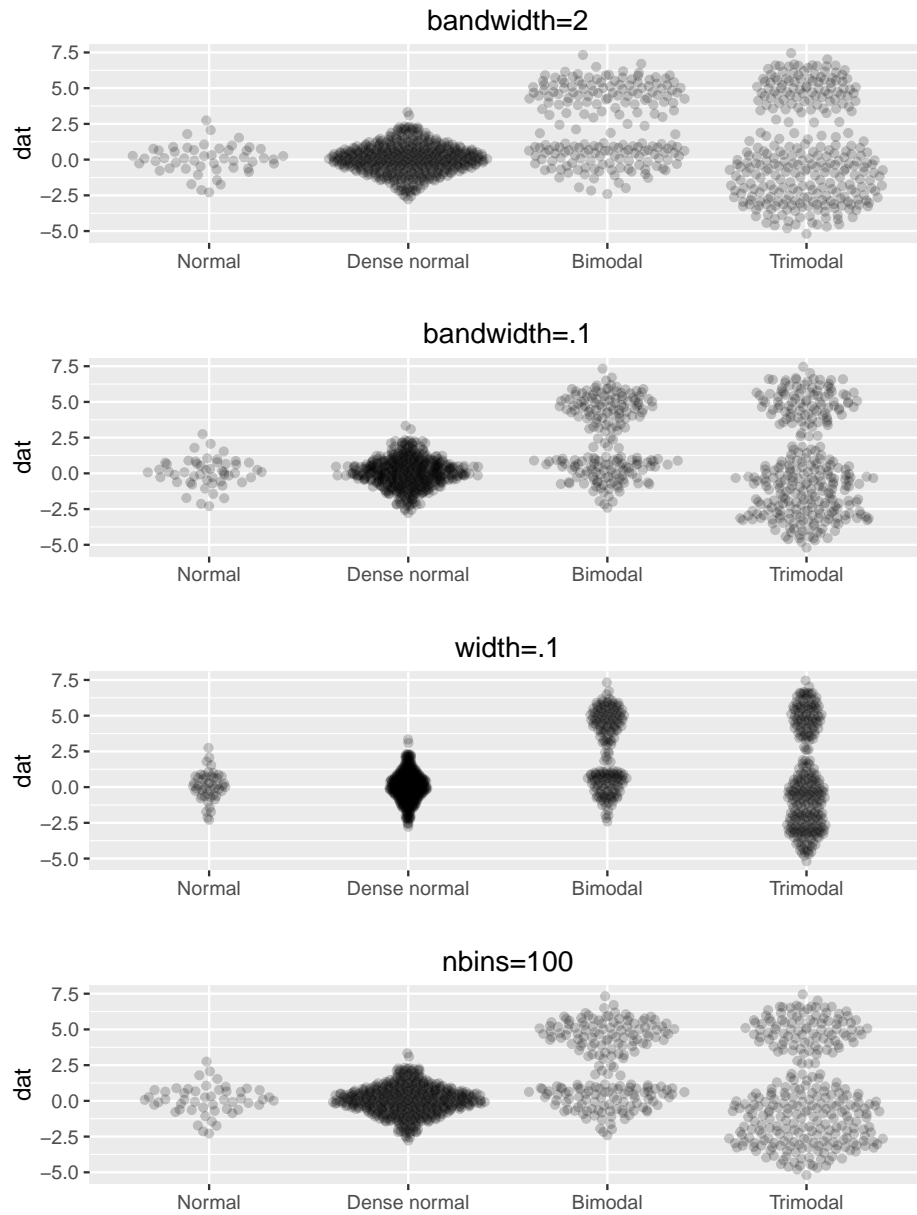
```
> labs2<-factor(labs,levels=c('b','a'))
> ggplot(mapping=aes(labs2, dat)) + geom_quasirandom(aes(color=labs))
```



2. Options

`quasirandom` calls `vipor::offsetX` which calls `stats::density` to compute kernel density estimates. The tightness of the fit can be adjusted with the `adjust` 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:

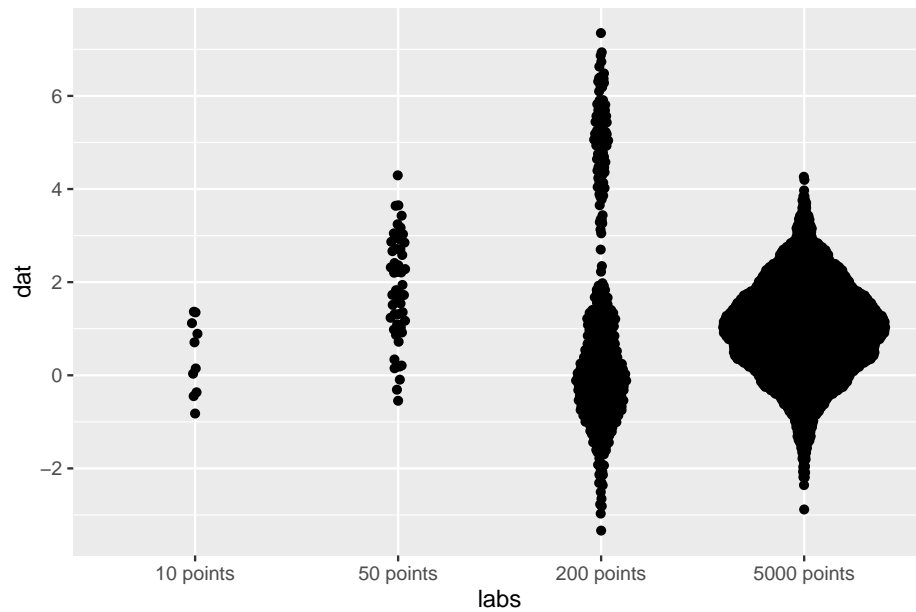
```
> library(gridExtra)
> dat <- list(
+   'Normal'=rnorm(50),
+   'Dense normal'= rnorm(500),
+   'Bimodal'=c(rnorm(100), rnorm(100,5)),
+   'Trimodal'=c(rnorm(100), rnorm(100,5),rnorm(100,-3))
+ )
> labs<-rep(names(dat),sapply(dat,length))
> labs<-factor(labs,levels=unique(labs))
> dat<-unlist(dat)
> p1<-ggplot(mapping=aes(labs, dat)) +
+   geom_quasirandom(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)
```



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))
```

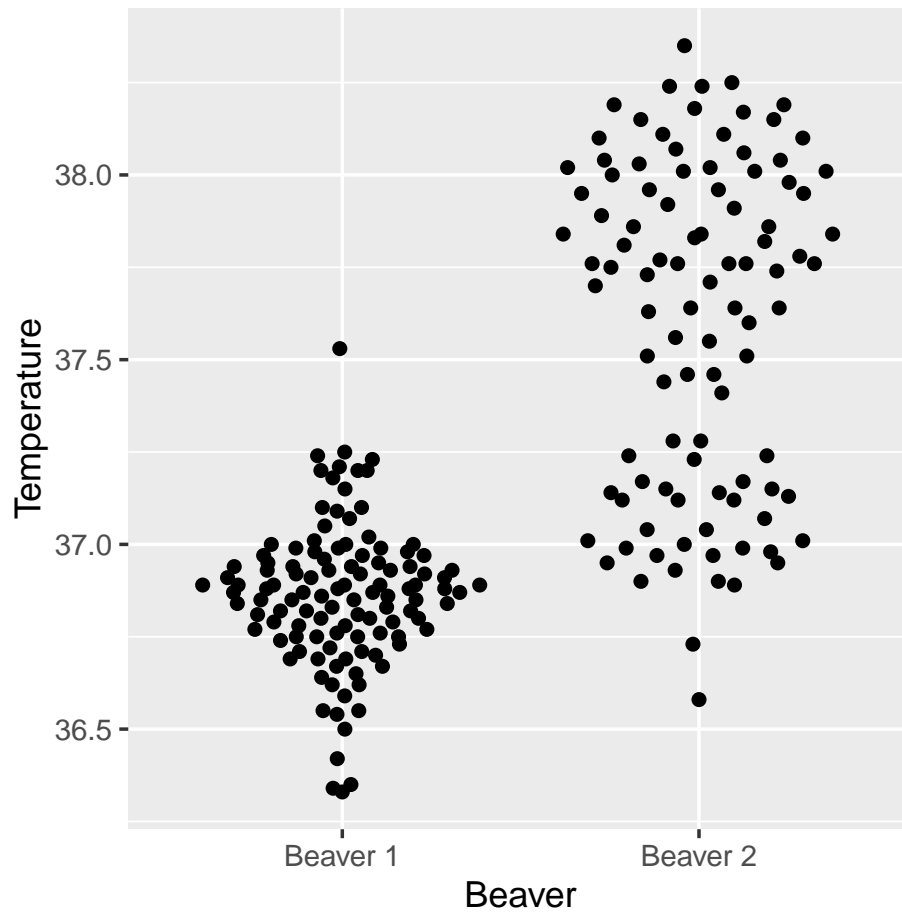
```
> dat<-unlist(dat)
> ggplot(mapping=aes(labs, dat)) + geom_quasirandom(varwidth=TRUE)
```



3. Real data

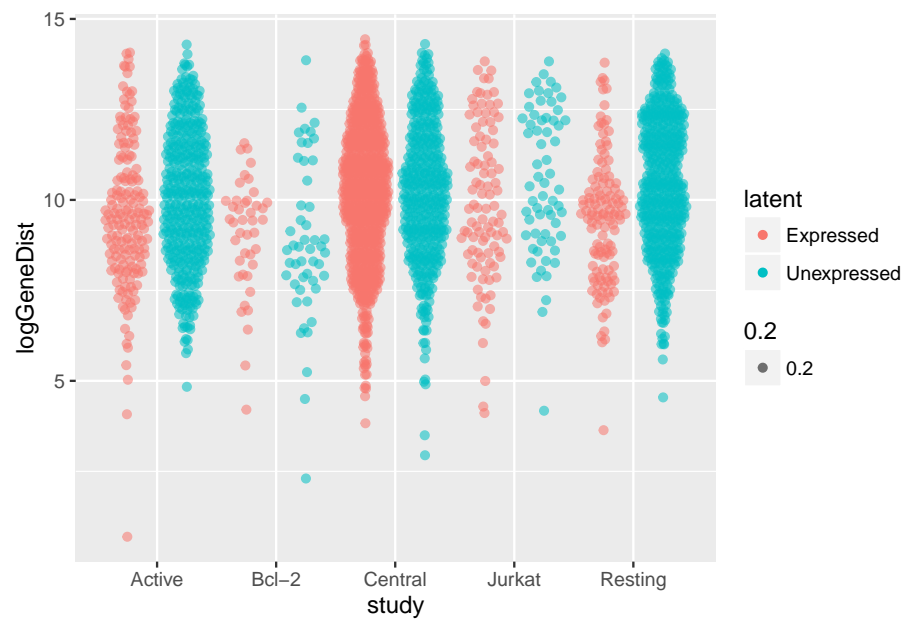
An example using the `beaver1` and `beaver2` data from the **datasets** package:

```
> 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()
```



An example using the `integrations` data from the **vipor** package:

```
> library(vipor)
> ints<-integrations[integrations$nearestGene>0,]
> ints$logGeneDist<-log(ints$nearestGene)
> ggplot(ints,mapping=aes(study, logGeneDist,color=latent,alpha=.2)) +
+   geom_quasirandom(dodge.width=1)
```



Affiliation:

Github: <http://github.com/eclarke/ggbeeswarm>

Cran: <https://cran.r-project.org/package=ggbeeswarm>