HoughPost Example

Panxi Chen, Sophia Luo

2022-12-12

Hough Transform Demonstration using "HoughPost" Package

```
# Check if required packages are installed. If not, then install them.
if (!require('devtools')) {
  install.packages("devtools")
}
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

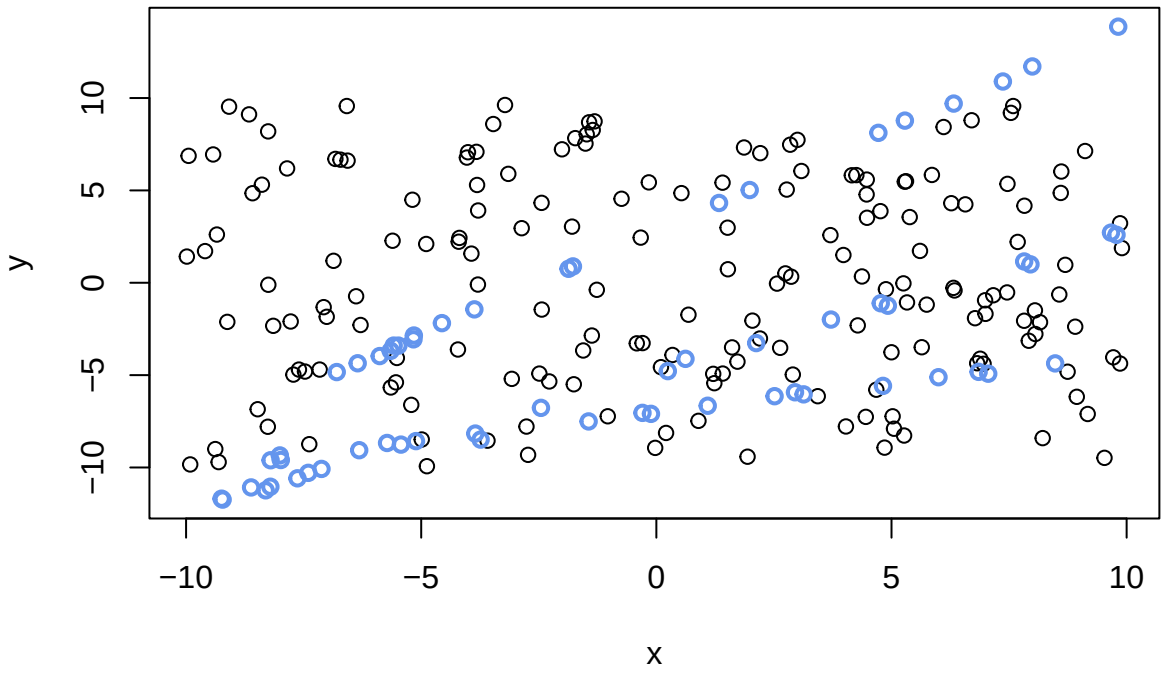
```
library(devtools)

# Install/update HoughPost package from GitHub if needed
install_github("ChenPanxi/Statistical_Computing/HoughPost")
```

```
## Skipping install of 'HoughPost' from a github remote, the SHA1 (cdc88a22) has not changed since last
##   Use `force = TRUE` to force installation
```

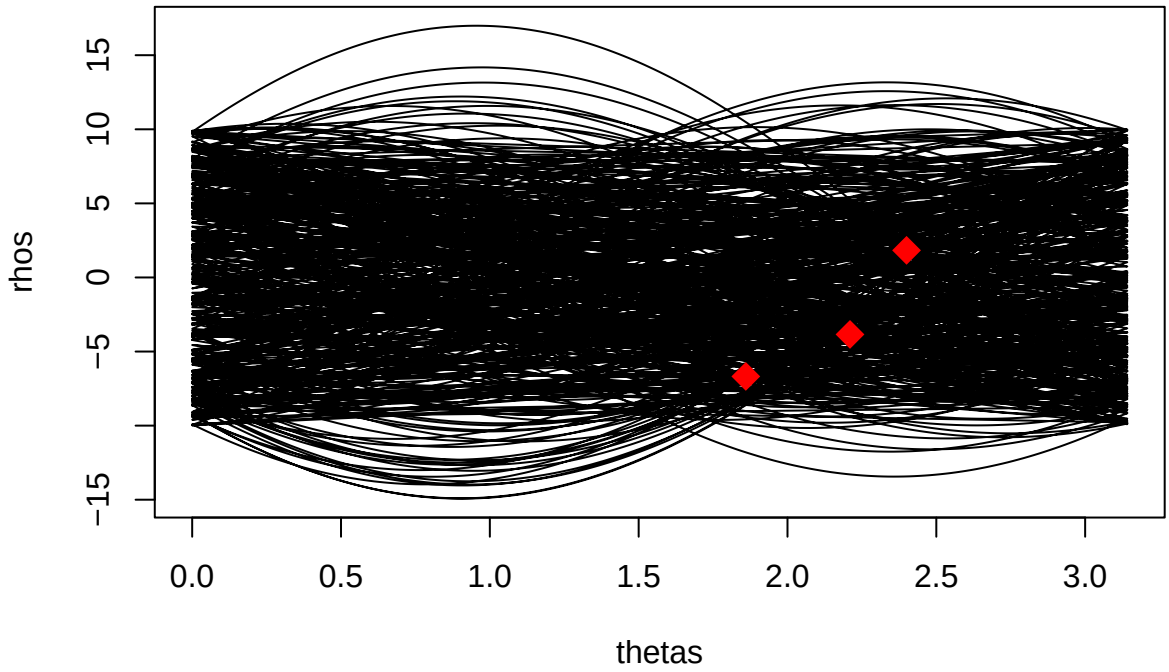
```
library(HoughPost)
```

Example 1.1: Line detection (3 lines)

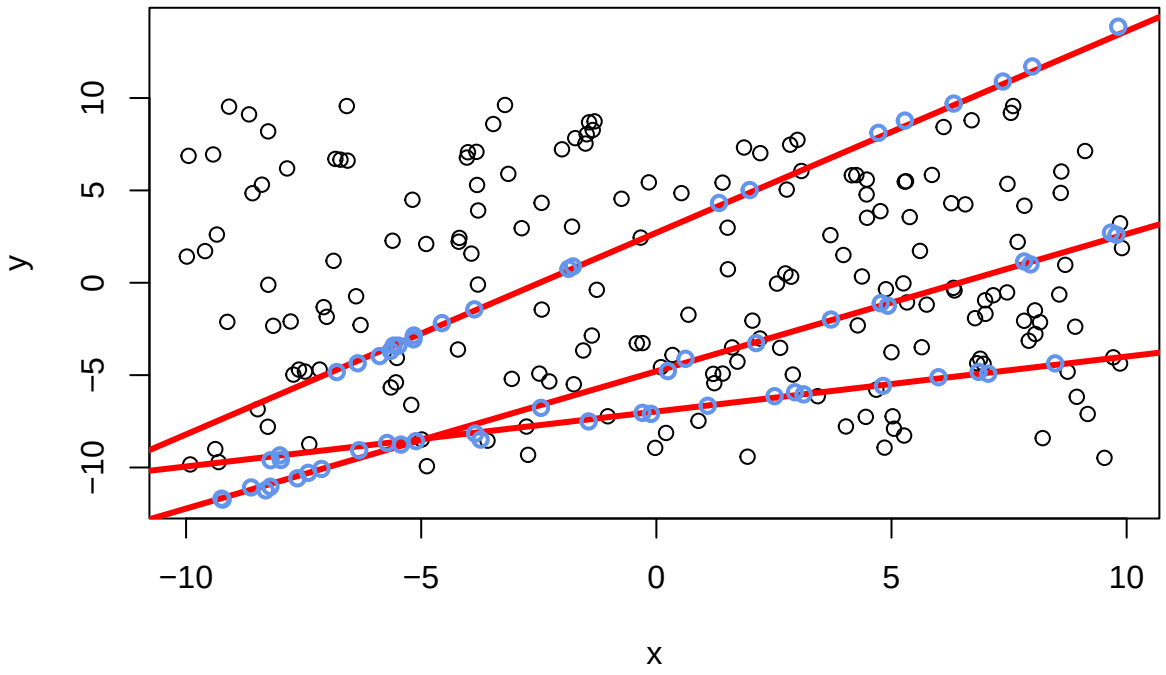
```
# Generate points along lines (line shown in blue, color can be changed)
set.seed(666)
coords_line = simulator(nshape=3, npoint=20, nnoise=180, shape_type='line',
                        line_noise=TRUE, noise=TRUE)
```




```
# Project points into Hough (parameter) space
hs_line = hough_space(coords_line$coord, thres=13, shape_type='line')
```

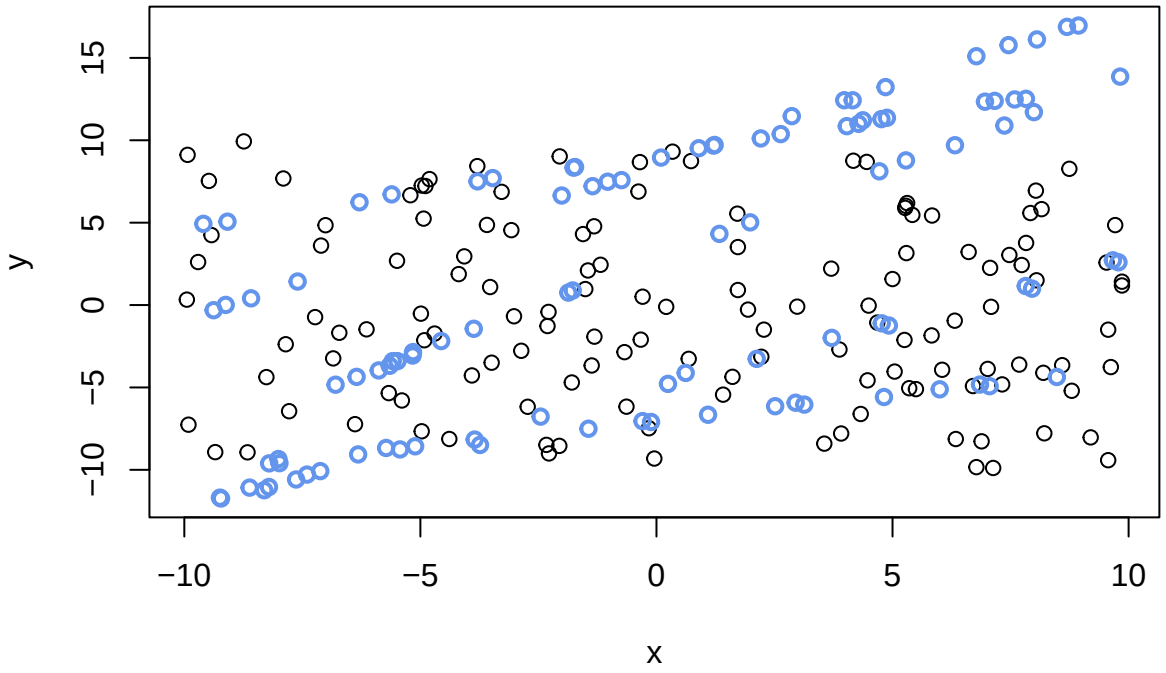



```
# Line detection
ht_line = hough_trans(coords_line$coord, hs_line$coords, shape_type='line')

# Show detected lines compared to true lines
ht_line = hough_trans(coords_line$coord, hs_line$coords,
                      coords_line$coord_shape, shape_type='line')
```

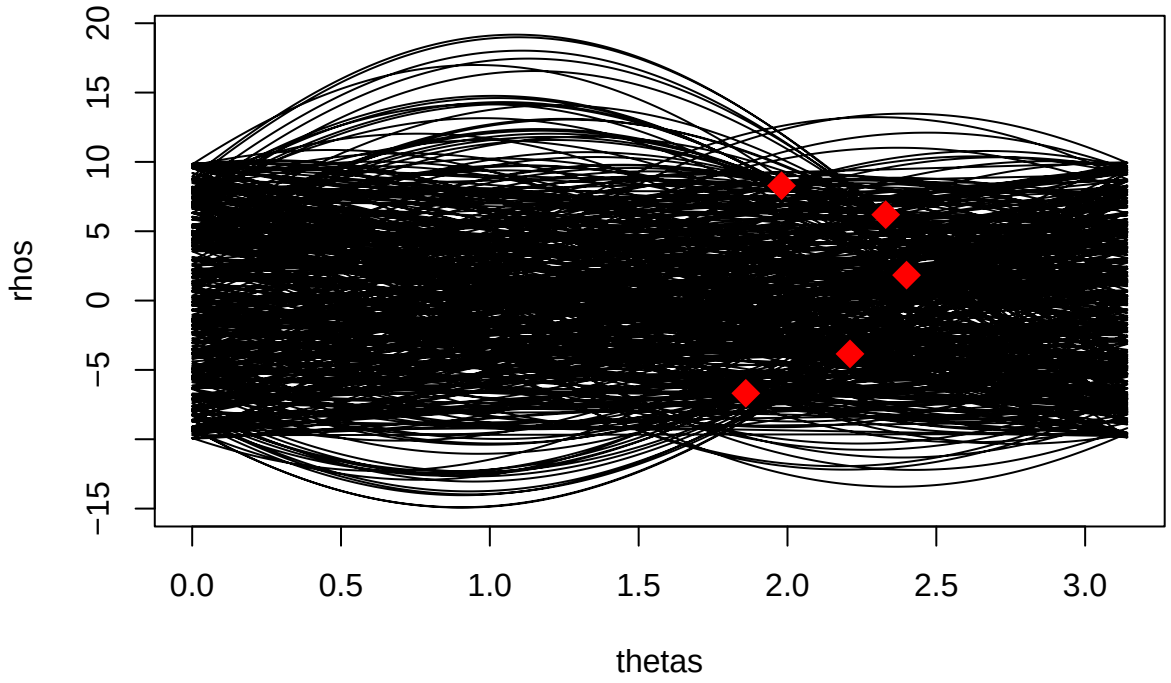


Example 1.2: Line detection (5 lines)

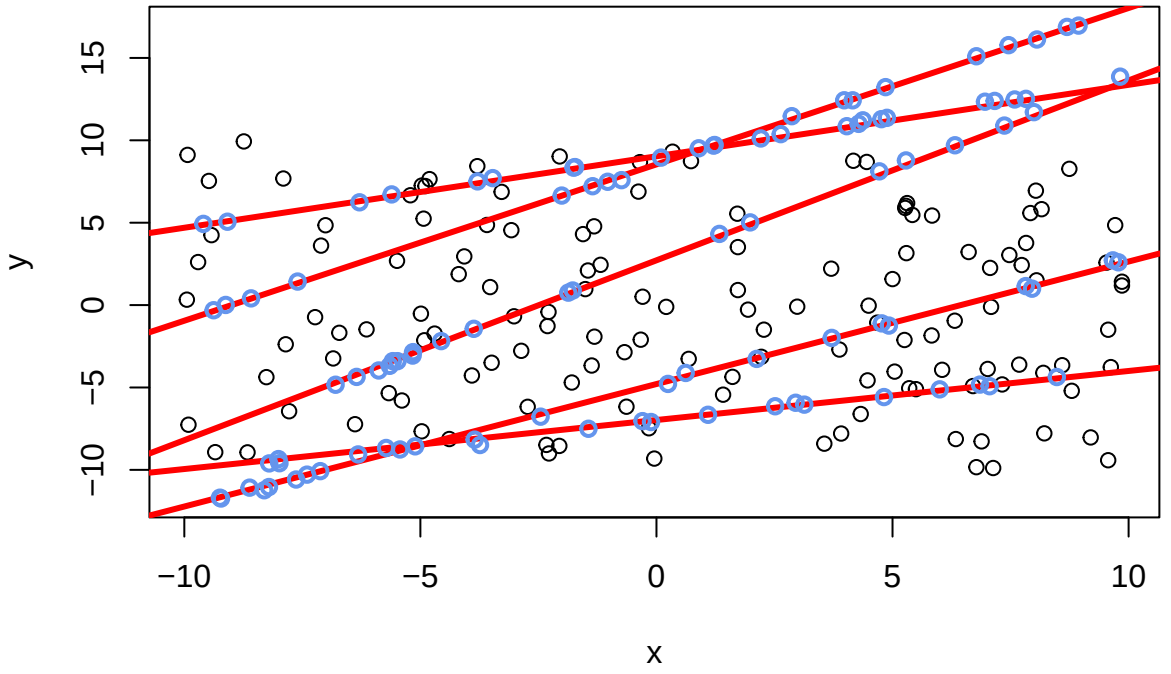
```
# Generate points along lines (line shown in blue, color can be changed)
set.seed(666)
coords_5_line = simulator(nshape=5, npoint=20, nnoise=135, shape_type='line',
                          line_noise=TRUE, noise=TRUE)
```




```
# Project points into Hough (parameter) space
hs_5_line = hough_space(coords_5_line$coord, thres=12, shape_type='line')
```



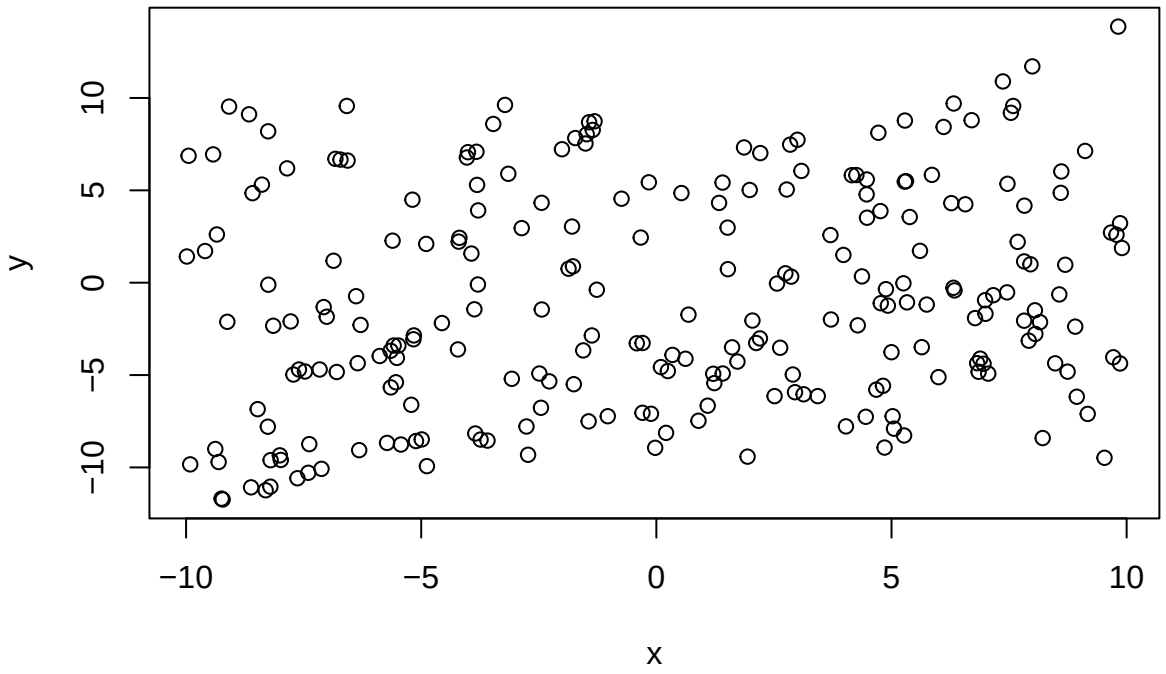

```
# Line detection
ht_5_line = hough_trans(coords_5_line$coord, hs_5_line$coords,
                        shape_type='line')

# Show detected lines compared to true lines
ht_line = hough_trans(coords_5_line$coord, hs_5_line$coords,
                      coords_5_line$coord_shape, shape_type='line')
```



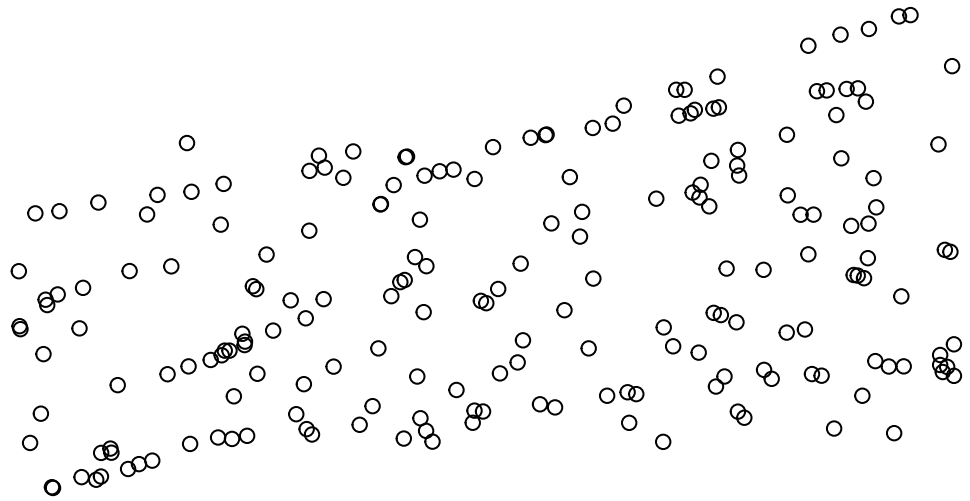
Create plots that can be exported as an images for use as input to Python implementations

```
# Example 1.1
set.seed(666)
```

```
coords_line = simulator(nshape=3, npoint=20, nnoise=180, shape_type='line',
                        show_simulation = FALSE, line_noise=TRUE, noise=TRUE)
```

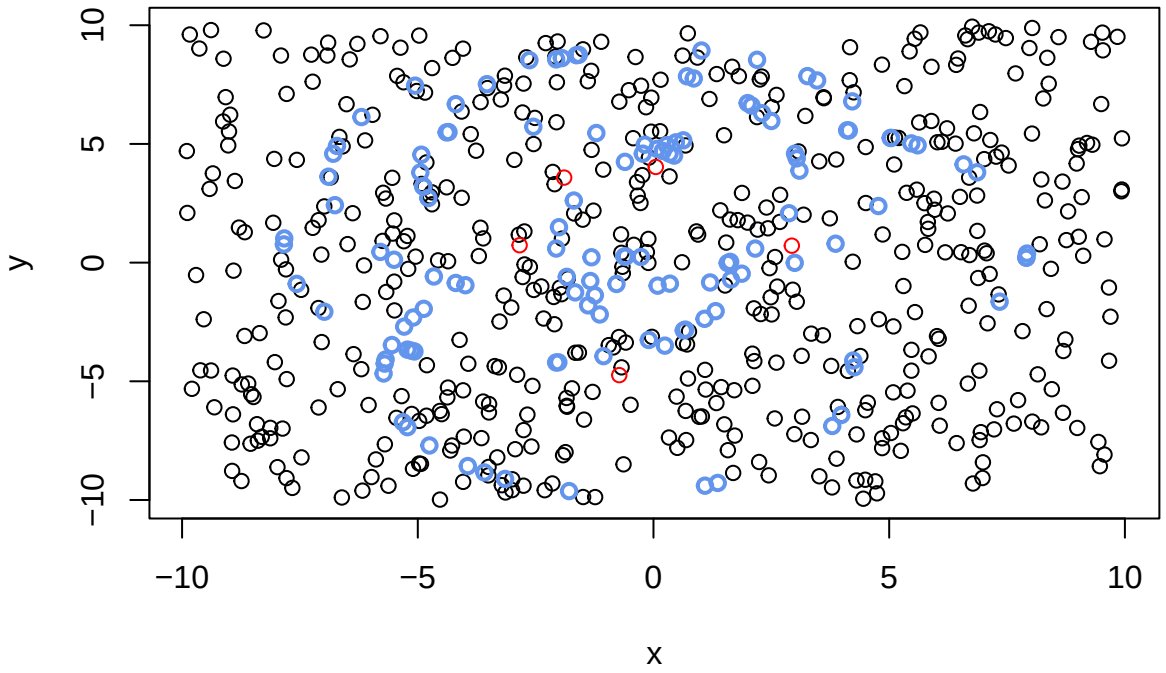



```
# Example 1.2
set.seed(666)
ht_line_output = simulator(nshape=5, npoint=20, nnoise=100, shape_type='line',
                           show_simulation=FALSE, noaxis=TRUE)
```

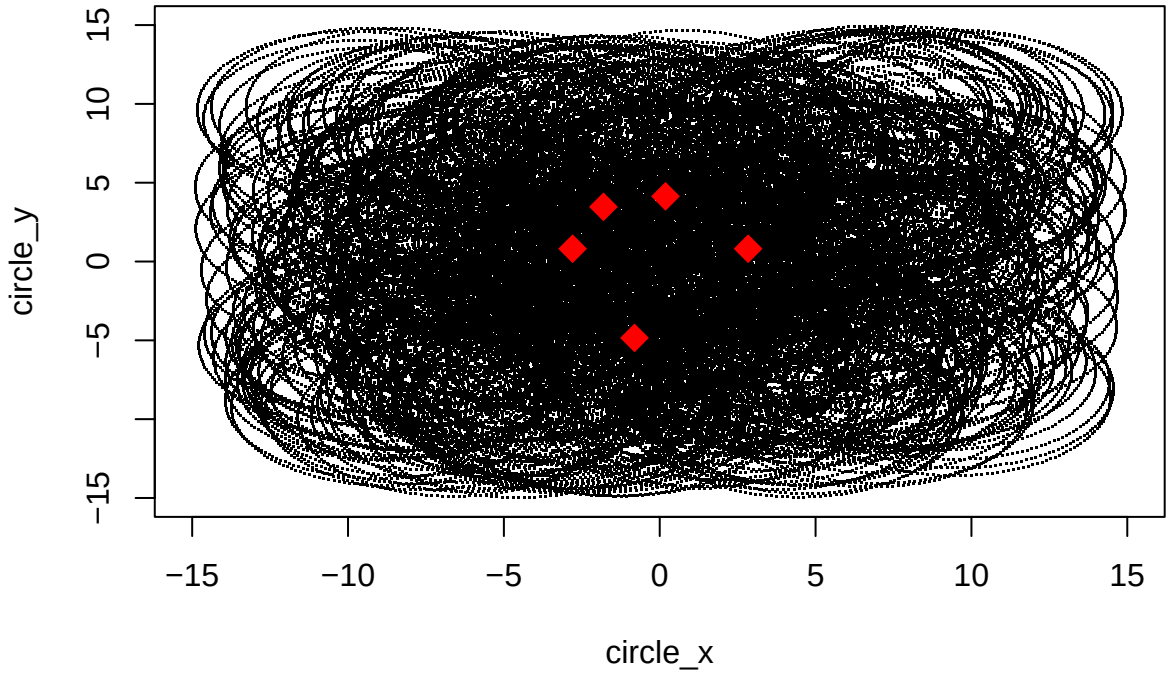


Example 2: Circle Detection (Assuming Known Radius)

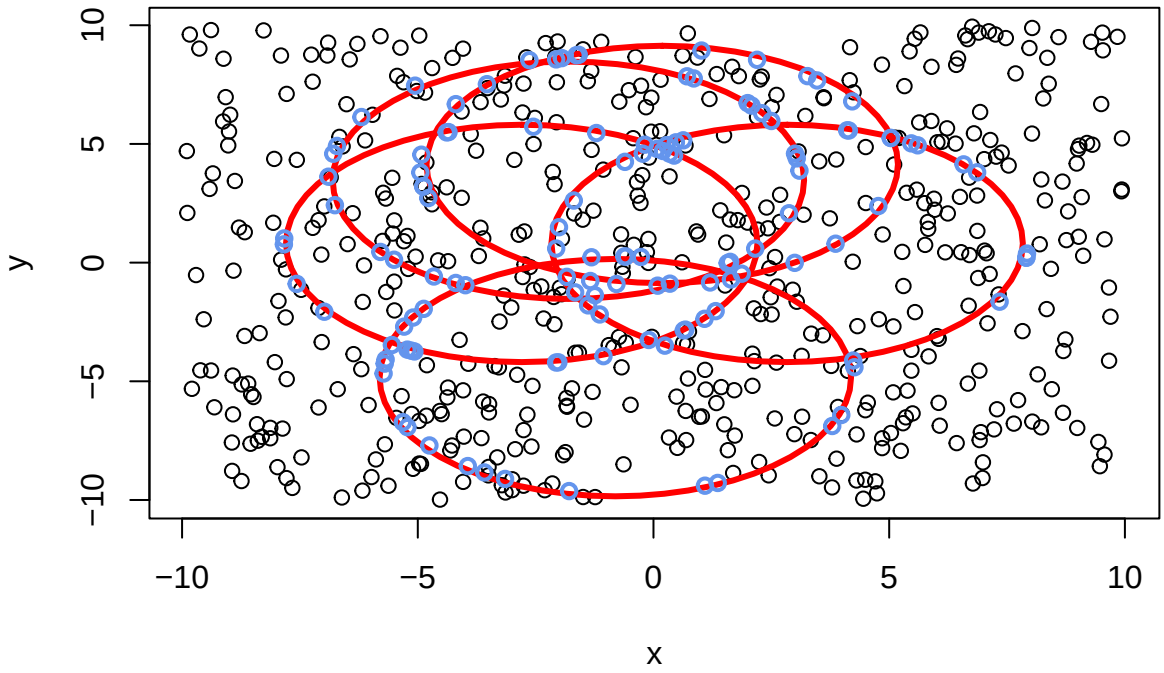
```
# Simulate points at fixed distance (blue) from random centers (shown in red)
set.seed(666)
circle_noise = simulator(nshape=5, npoint=25, nnoise=500, radius=5,
                         shape_type='circle', line_noise=TRUE, noise=TRUE)
points(circle_noise$center, col='red')
```




```
# Hough space projection
hs_circle = hough_space(circle_noise$coord, steps=90, thres=70,
                        shape_type='circle')
```




```
# Detect circles
ht_circle = hough_trans(circle_noise$coord, hs_circle$coords,
                        shape_type='circle')

# Compare detected circles to points on true circles
ht_circle = hough_trans(circle_noise$coord, hs_circle$coords,
                        circle_noise$coord_shape, shape_type='circle')
```



Create plot that can be used as image for input to Python implementations

```
# Example 2
set.seed(666)
circle_noise_output = simulator(nshape=5, npoint=25, nnoise=500, radius=5,
                                shape_type='circle', show_simulation=FALSE, noaxis=TRUE)
```