Lab 1

AUTHOR
Andre Gala-Garza

```
library(datasauRus)
```

Question 1

```
?datasaurus_dozen
```

```
starting httpd help server ... done
```

From the help file for the Datasaurus Dozen dataset, we find that there are 1846 rows and three columns (variables) in the dataset. The three variables are "dataset", which indicates where the data is coming from, "x" for the x-values, and "y" for the y-values, allowing the data to be plotted on a 2D Cartesian plane.

```
table(datasaurus_dozen$dataset)
```

```
      away   bullseye     circle       dino       dots    h_lines high_lines 
       142        142        142        142        142        142        142 
slant_down   slant_up       star    v_lines wide_lines    x_shape 
       142        142        142        142        142        142 
```

This table shows the frequency of data points per dataset in the Datasaurus Dozen. Interestingly, there are 142 points in each of 13 datasets for a total of 142 * 13 = 1846 rows, so the Datasaurus Dozen is actually a baker's dozen.

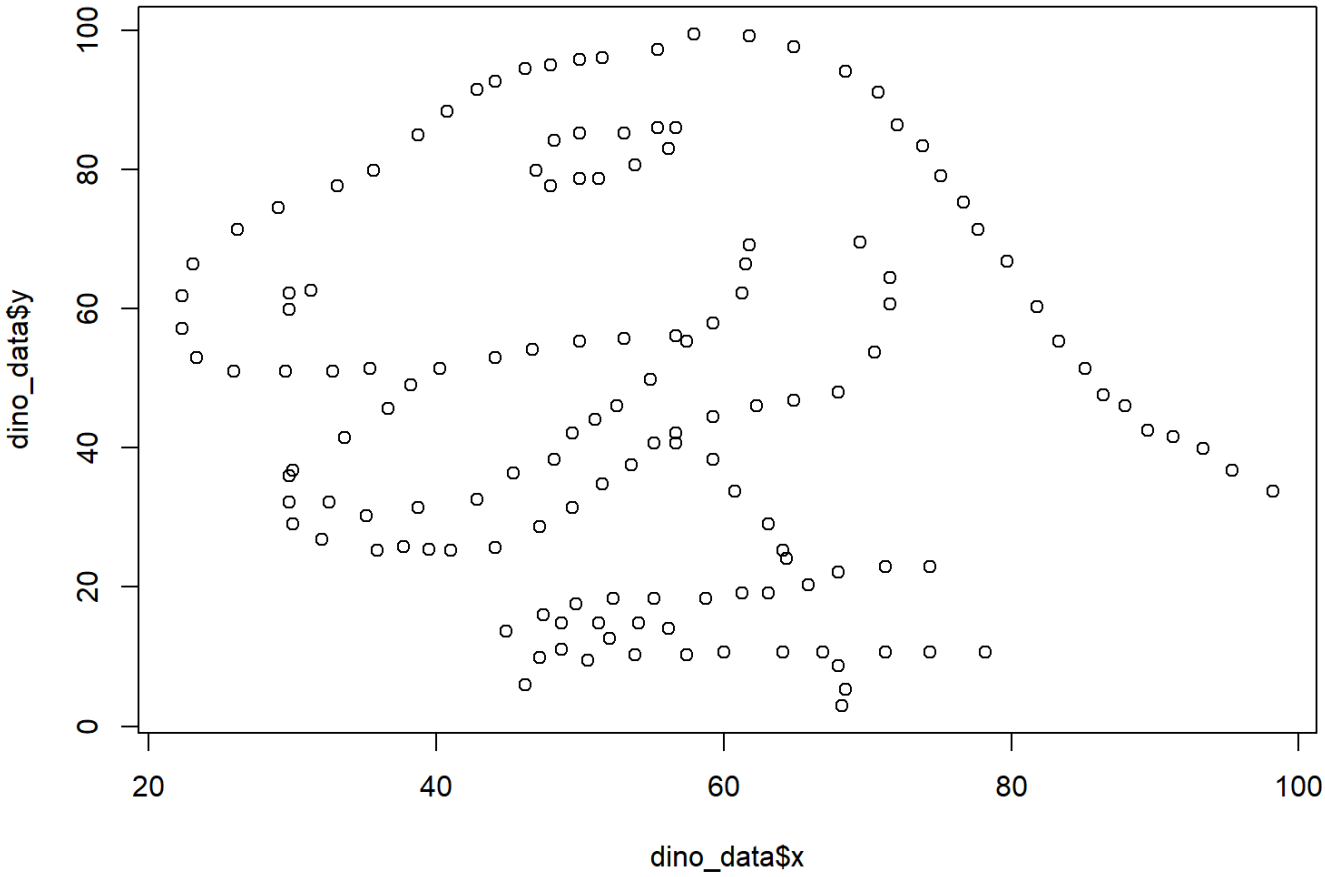
Question 2

We set dino_data as the subset of the Datasaurus Dozen where the "dataset" variable is equal to "dino":

```
dino_data <- datasaurus_dozen[datasaurus_dozen$dataset == 'dino', ]
```

We then plot the "x" variable against the "y" variable in this subset:

```
plot(dino_data$x, dino_data$y)
```




```
# ggplot(data = dino_data, mapping = aes(x = x, y = y)) +
#   geom_point()
```

We then calculate the correlation coefficient of the data:

```
cor(dino_data$x, dino_data$y)
```

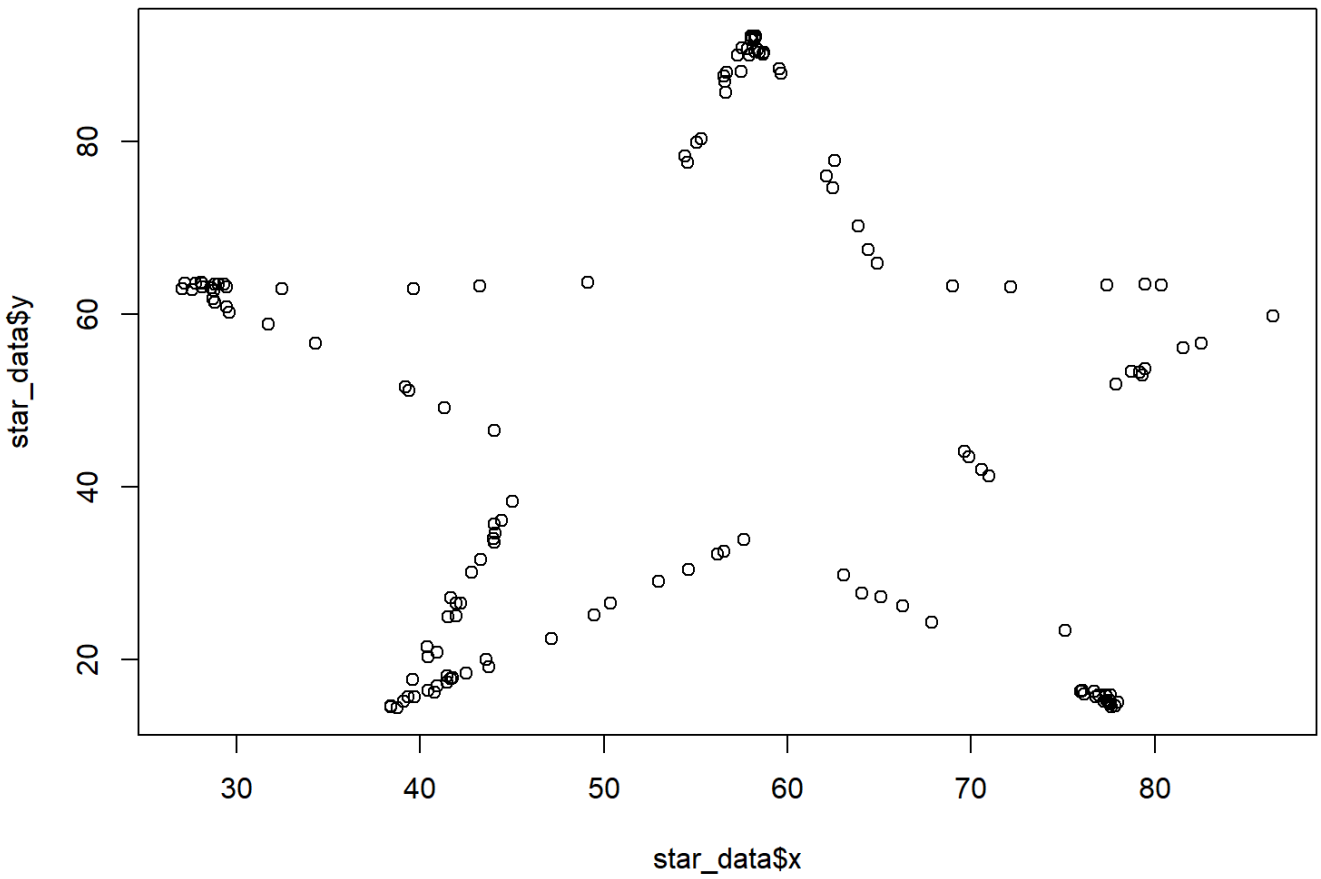
```
[1] -0.06447185
```

```
# dino_data |>
#   summarize(r = cor(x, y))
```

Question 3

We now repeat this procedure with the “star” dataset:

```
star_data <- datasaurus_dozen[datasaurus_dozen$dataset == 'star', ]
plot(star_data$x, star_data$y)
```




```
cor(star_data$x, star_data$y)
```

```
[1] -0.0629611
```

```
cor(dino_data$x, dino_data$y)
```

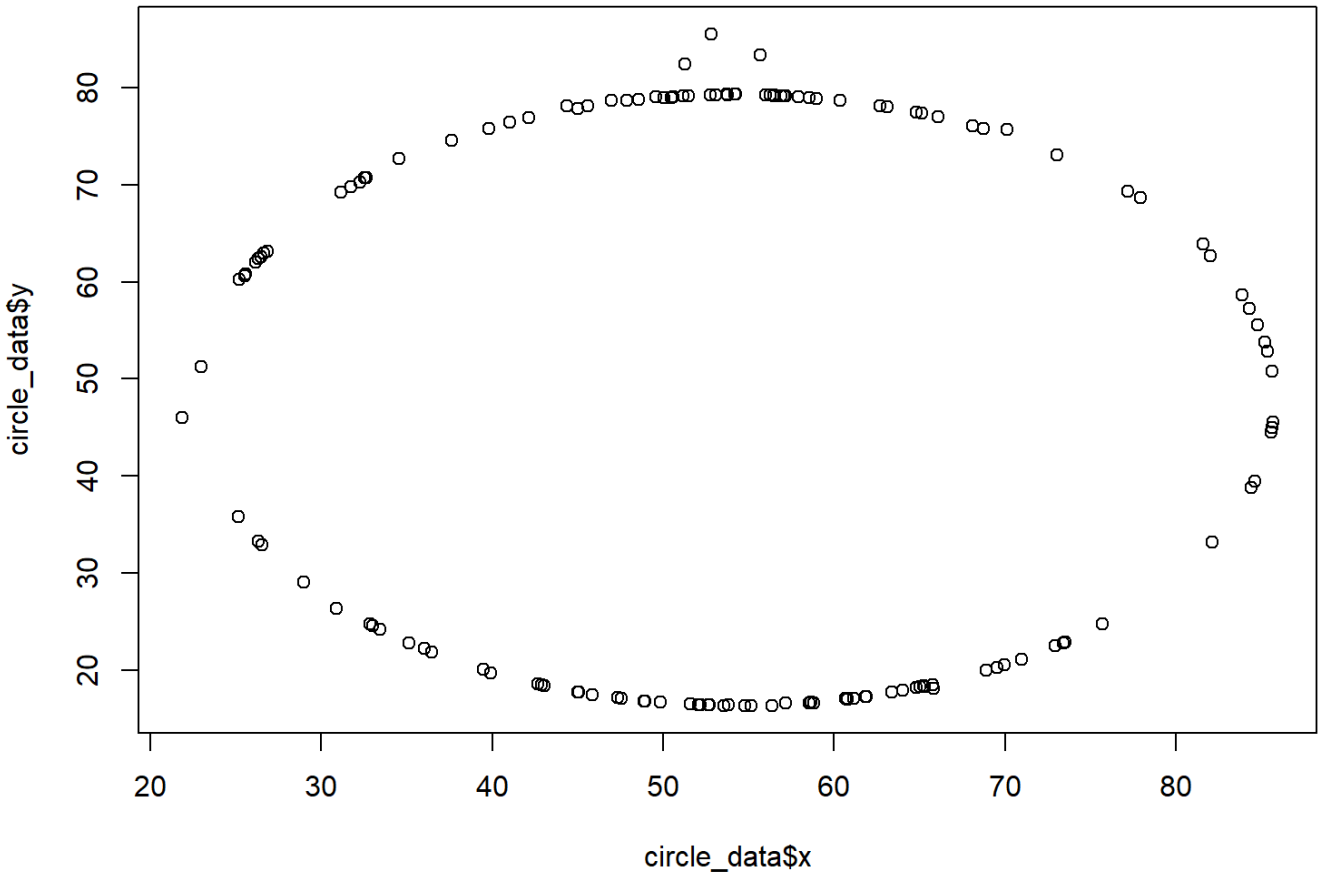
```
[1] -0.06447185
```

As verified by the Datasaurus Dozen help file, we find that the correlation coefficient for the "star" dataset is roughly equal to that of "dino", although not exactly equal; the "star" coefficient is roughly 0.002 units less negative.

Question 4

We now repeat this procedure with the "circle" dataset:

```
circle_data <- datasaurus_dozen[datasaurus_dozen$dataset == 'circle', ]
plot(circle_data$x, circle_data$y)
```




```
cor(circle_data$x, circle_data$y)
```

```
[1] -0.06834336
```

```
cor(dino_data$x, dino_data$y)
```

```
[1] -0.06447185
```

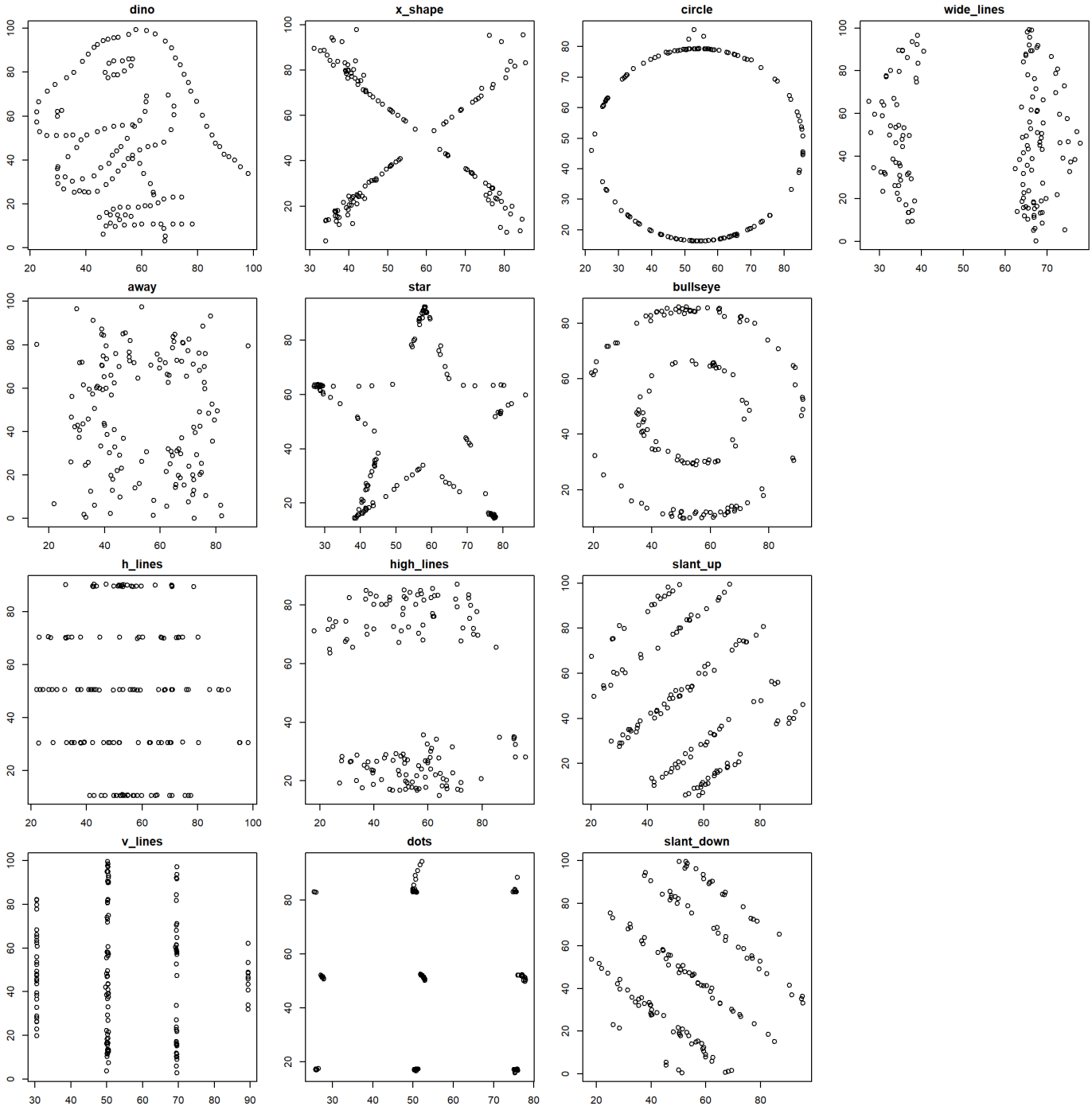
There is a slightly more significant difference between the Pearson's correlation coefficient for "dino" vs. "circle"; this time, the "circle" coefficient is roughly 0.004 units more negative.

Question 5

We now create a matrix to allow us to plot all 13 datasets at once:

```
layout(matrix(1:16, nrow=4, ncol=4))
par(mar = c(2,2,2,2))
for(name in unique(datasaurus_dozen$dataset)){
  subset <- datasaurus_dozen[datasaurus_dozen$dataset == name, ]
```

```
  plot(subset$x, subset$y, main = name)
}
layout(1)
```




```
par(mar = c(5,4,4,2) + 0.1)
```

Question 6

We now calculate the correlation coefficient between x and y for all 13 datasets:

```r
sapply(unique(datasaurus_dozen$dataset), function(name){
    subset <- datasaurus_dozen[datasaurus_dozen$dataset == name, ]
    return(cor(subset$x, subset$y))
})
```

```
      dino        away     h_lines     v_lines     x_shape        star 
-0.06447185 -0.06412835 -0.06171484 -0.06944557 -0.06558334 -0.06296110 
 high_lines        dots      circle    bullseye    slant_up  slant_down 
-0.06850422 -0.06034144 -0.06834336 -0.06858639 -0.06860921 -0.06897974 
 wide_lines 
-0.06657523 
```