practical 1

Narges Yarahmadi Gharaei

2023-05-16

installing packages and importing them

```
library(tidyverse)
library(datasauRus)
library(knitr)
```

dataset

```
datasaurus_dozen
```

```
## # A tibble: 1,846 x 3
##    dataset     x     y
##    <chr>   <dbl> <dbl>
##  1 dino     55.4  97.2
##  2 dino     51.5  96.0
##  3 dino     46.2  94.5
##  4 dino     42.8  91.4
##  5 dino     40.8  88.3
##  6 dino     38.7  84.9
##  7 dino     35.6  79.9
##  8 dino     33.1  77.6
##  9 dino     29.0  74.5
## 10 dino     26.2  71.4
## # i 1,836 more rows
```

check dataset's dimension

```
print( nrow(datasaurus_dozen))
```

```
## [1] 1846
```

```
print( ncol(datasaurus_dozen))
```

```
## [1] 3
```

```
print( dim(datasaurus_dozen))
```

```
## [1] 1846    3
```

count each dataset : every dataset inside datasaurs_dozen has 142 points

```
datasaurus_dozen %>% count(dataset)
```

```
## # A tibble: 13 x 2
##    dataset        n
##    <chr>      <int>
##  1 away         142
##  2 bullseye     142
##  3 circle       142
##  4 dino         142
##  5 dots         142
##  6 h_lines      142
##  7 high_lines   142
##  8 slant_down   142
##  9 slant_up     142
## 10 star         142
## 11 v_lines      142
## 12 wide_lines   142
## 13 x_shape      142
```

filtering data (here dino) and put in dino_data

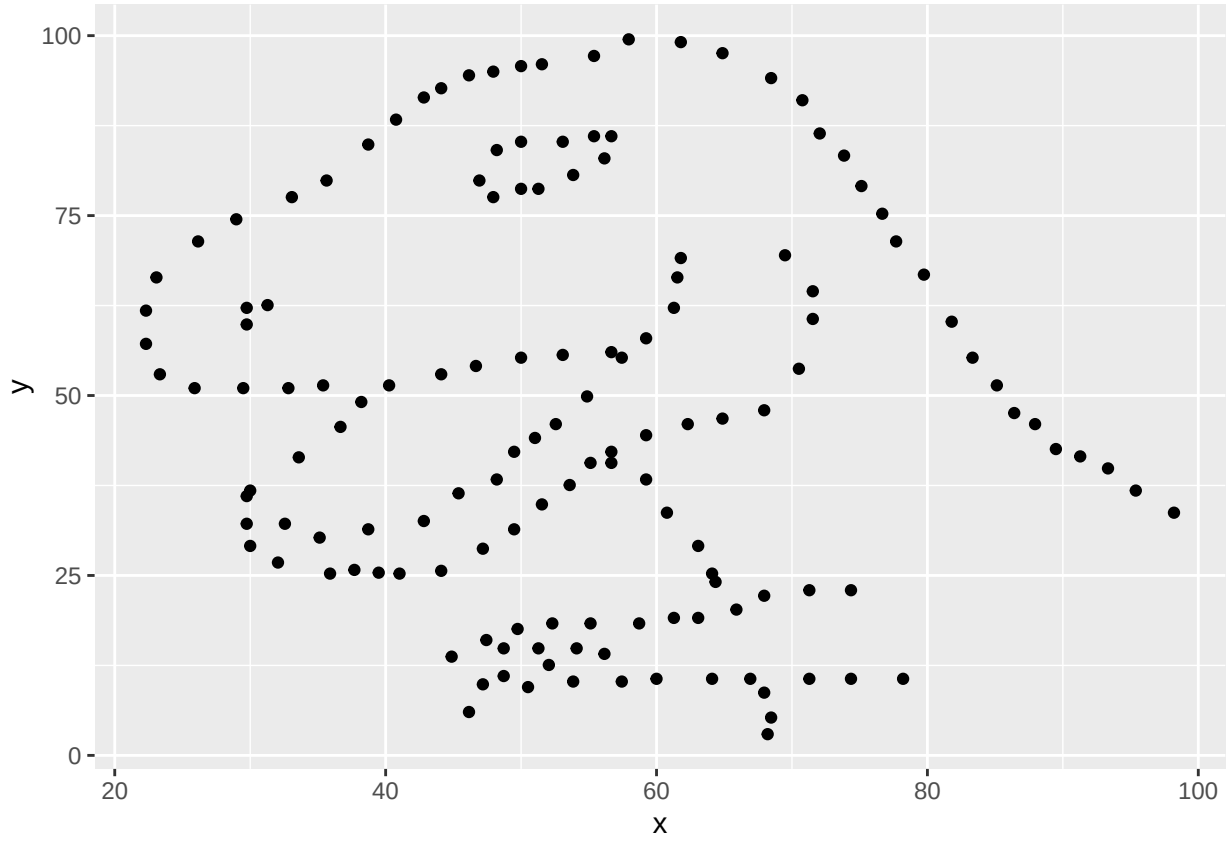
```
dino_data <- datasaurus_dozen %>%
  filter(dataset == "dino")
```

```
dino_data
```

```
## # A tibble: 142 x 3
##    dataset     x     y
##    <chr>   <dbl> <dbl>
##  1 dino     55.4  97.2
##  2 dino     51.5  96.0
##  3 dino     46.2  94.5
##  4 dino     42.8  91.4
##  5 dino     40.8  88.3
##  6 dino     38.7  84.9
##  7 dino     35.6  79.9
##  8 dino     33.1  77.6
##  9 dino     29.0  74.5
## 10 dino     26.2  71.4
## # i 132 more rows
```

plot dino data

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



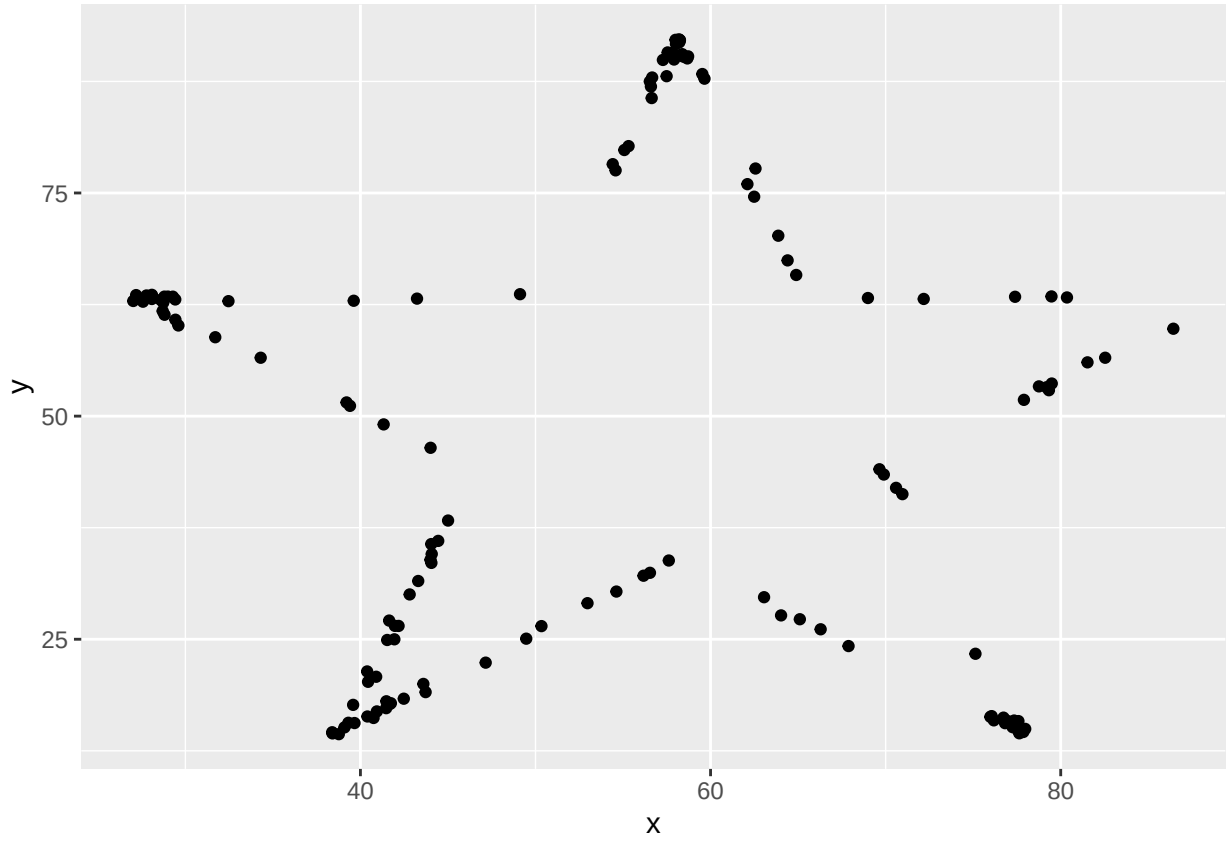
check the corelation of dino dataset

```
dino_data_r = dino_data %>% summarize(dino_data_r = cor(x, y))
print(dino_data_r)
```

```
## # A tibble: 1 x 1
##   dino_data_r
##         <dbl>
## 1     -0.0645
```

question 3 : do the same steps for star dataset

```
star_dataset <- datasaurus_dozen %>%  filter(dataset == "star")
ggplot(data = star_dataset, mapping = aes(x = x, y = y)) + geom_point()
```

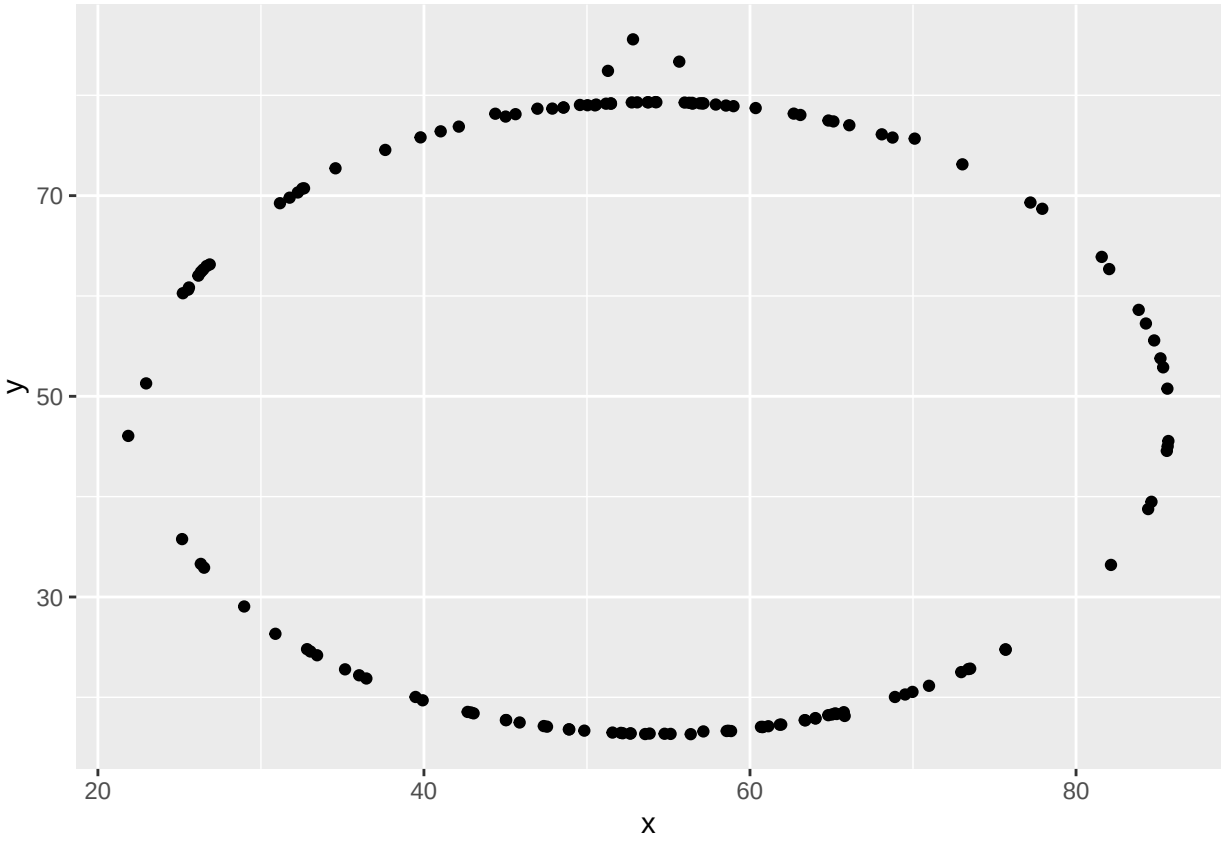



```
star_dataset_r = star_dataset %>% summarize(star_dataset_r = cor(x, y))
print(star_dataset_r)
```

```
## # A tibble: 1 x 1
##   star_dataset_r
##            <dbl>
## 1        -0.0630
```

question 4 : do the same steps for star dataset

```
circle_dataset <- datasaurus_dozen %>%  filter(dataset == "circle")
ggplot(data = circle_dataset, mapping = aes(x = x, y = y)) + geom_point()
```



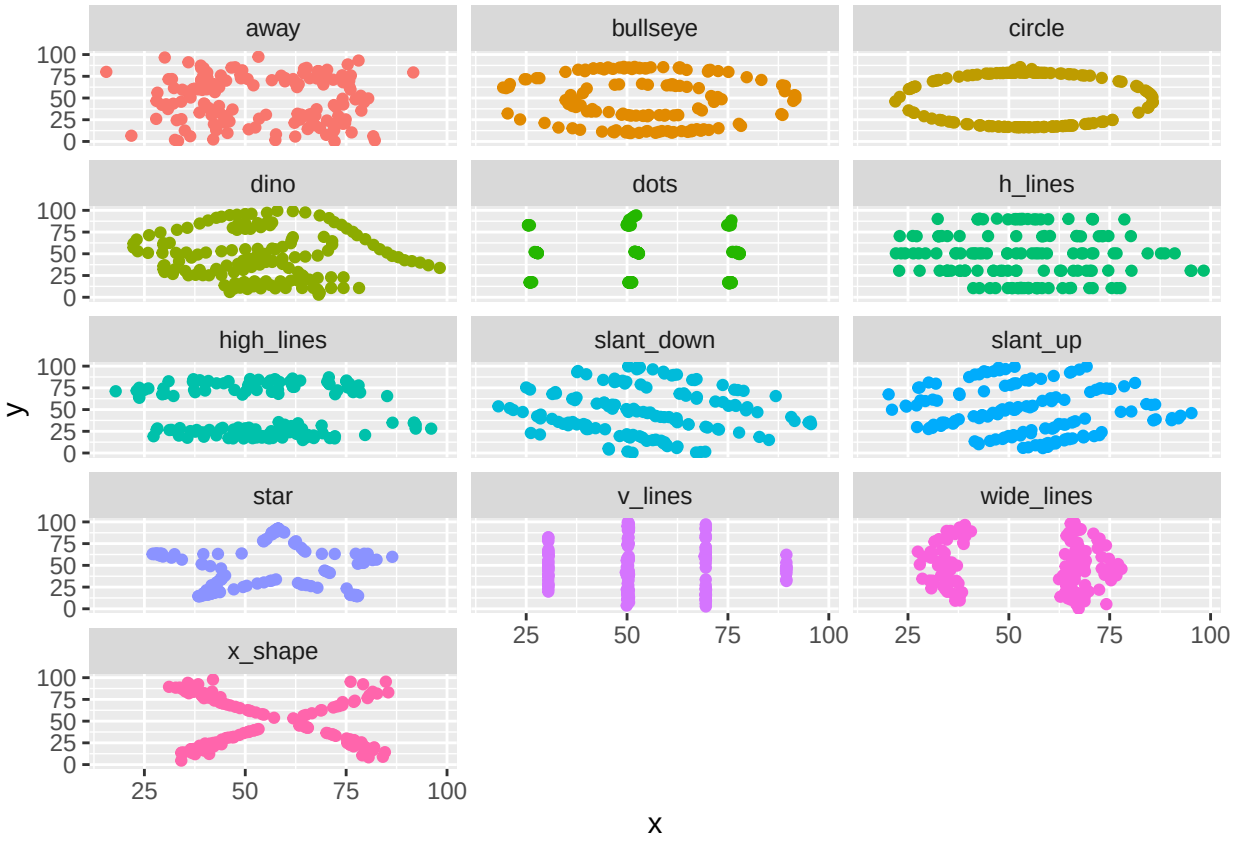

```
circle_dataset_r  = circle_dataset %>% summarize(circle_dataset_r = cor(x, y))
print(circle_dataset_r)
```

```
## # A tibble: 1 x 1
##   circle_dataset_r
##              <dbl>
## 1          -0.0683
```

As the correlation coefficient (r) exhibits a nearly identical and considerably low negative value, along with the datasets having almost identical mean values, it becomes challenging to ascertain the most suitable linear fit for Y as a function of X.

question 5:

```
ggplot(datasaurus_dozen, aes(x = x, y = y, color = dataset))+
  geom_point()+
  facet_wrap(~ dataset, ncol = 3) +
  theme(legend.position = "none")
```




```
datasaurus_dozen %>%
group_by(dataset) %>%
summarize(r = cor(x, y))
```

```
## # A tibble: 13 x 2
##    dataset          r
##    <chr>        <dbl>
##  1 away       -0.0641
##  2 bullseye   -0.0686
##  3 circle     -0.0683
##  4 dino       -0.0645
##  5 dots       -0.0603
##  6 h_lines    -0.0617
##  7 high_lines -0.0685
##  8 slant_down -0.0690
##  9 slant_up   -0.0686
## 10 star       -0.0630
## 11 v_lines    -0.0694
## 12 wide_lines -0.0666
## 13 x_shape    -0.0656
```