Practical 0

Alexandra Del Favero-Campbell

2024-05-09

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

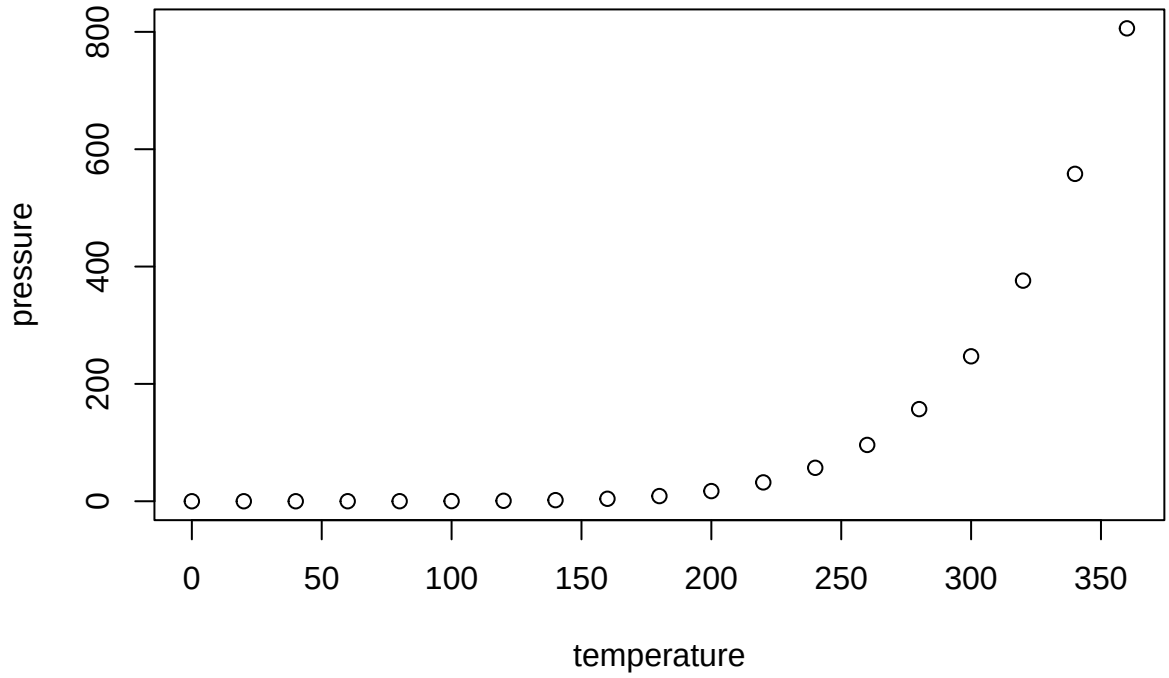
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
```

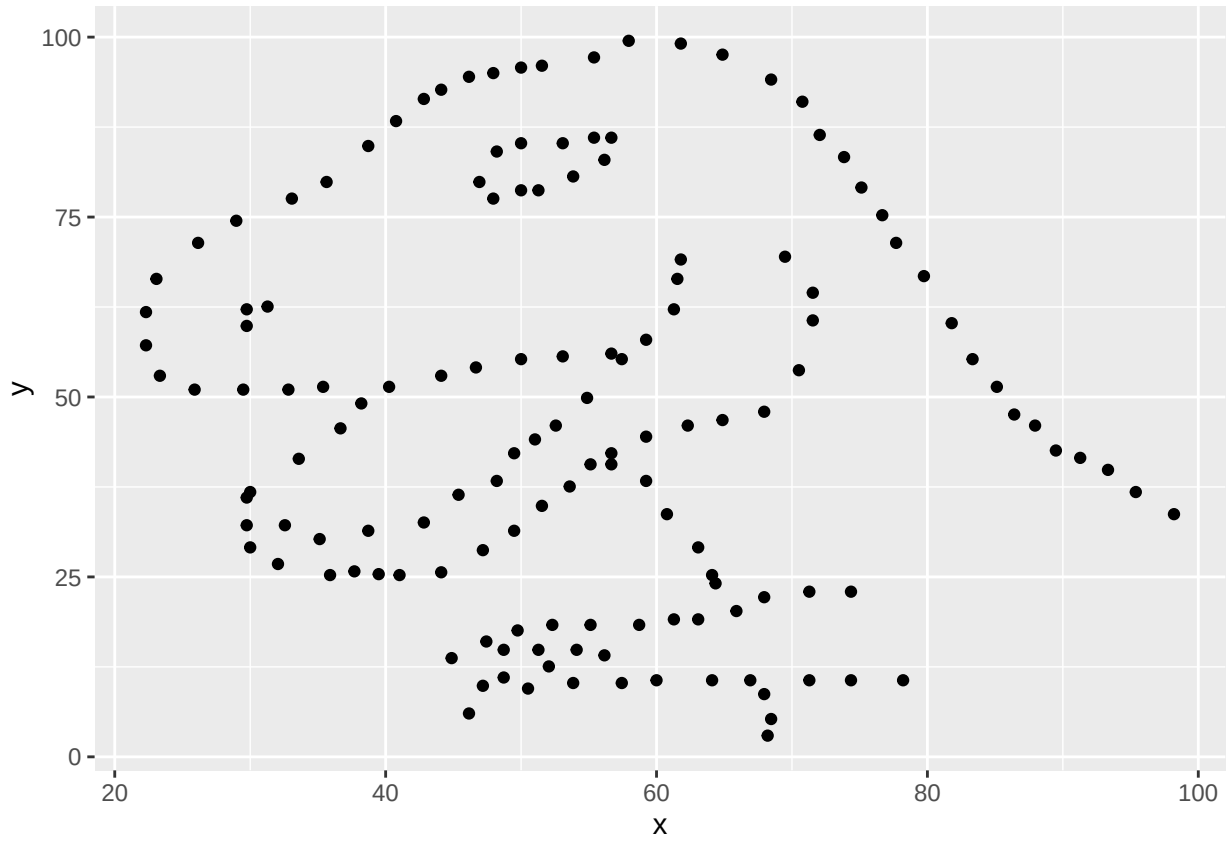
```
## v ggplot2   3.5.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(datasauRus)
```

```
## Warning: package 'datasauRus' was built under R version 4.3.3
```

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

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



Calculate the summary statistic (i.e., correlation coefficient) between x and y.

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

```
## # A tibble: 1 x 1
```

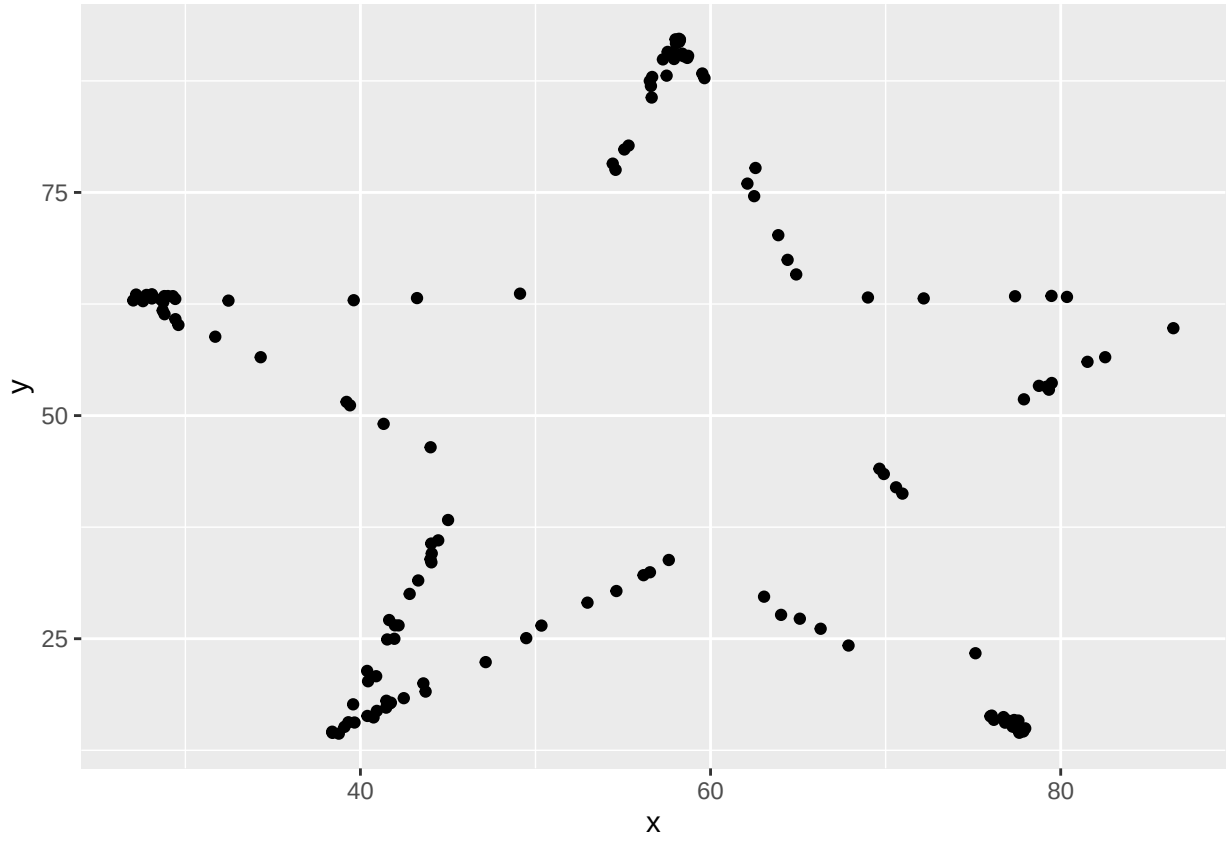
```
##         r
##     <dbl>
## 1 -0.0645
```

3. Plot y vs. x for the "star" dataset. Then, calculate the correlation coefficient between x and y for this dataset. How does this value compare to the r in "dino"?

```
star_data<-datasaurus_dozen %>%
  filter(dataset=="star")
```

Plot x and y of star dataset

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



Calculate correlation coefficient of x and y from star dataset.

```
star_data %>%
  summarize(r=cor(x,y))
```

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

Comparing both correlation coefficients, we see that both datasets have a quite low negative correlation coefficient between their respective x and y variables. However, the dino dataset has a bit of a minutely higher correlation compared to the star dataset.

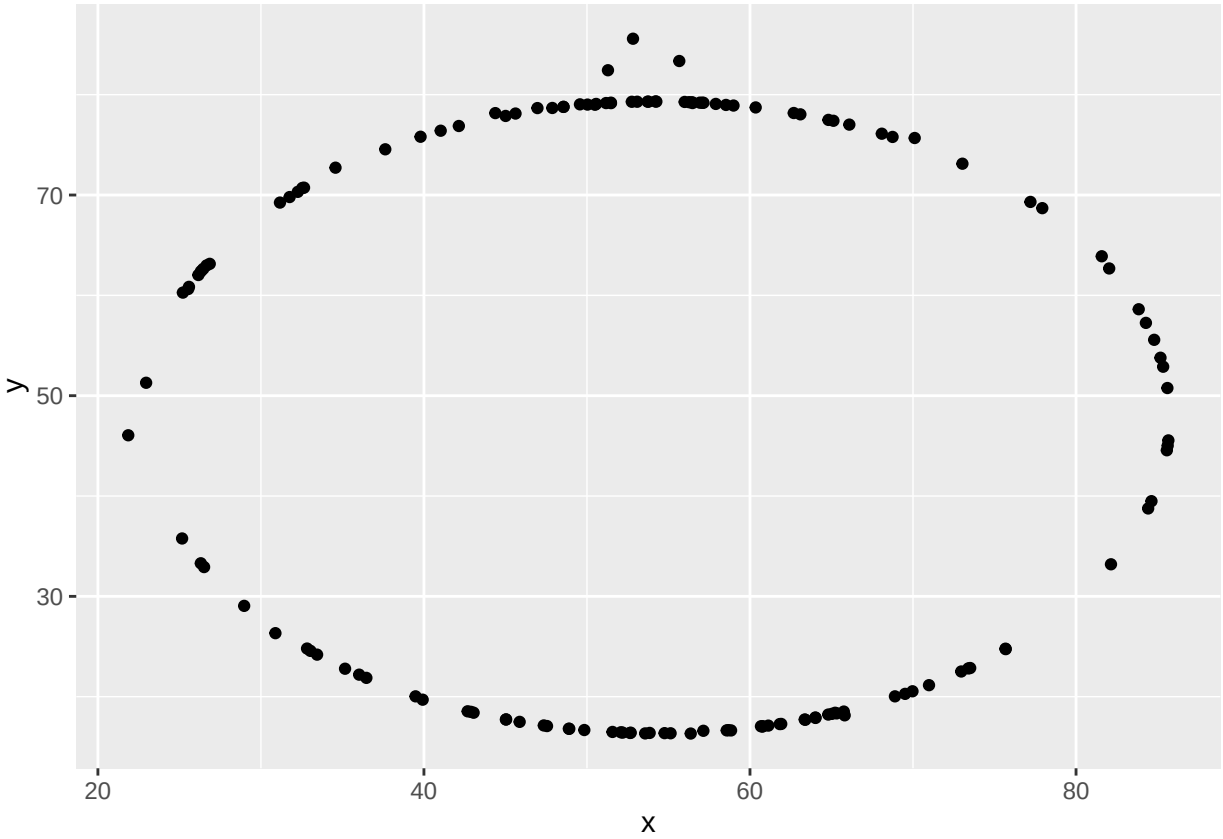
4. Plot y vs. x for the "circle" dataset. Then, calculate the correlation coefficient between x and y for this dataset. How does this value compare to the r of dino?

A. Filter out our data so that we only are looking at the "circle" dataset.

```
circle_data <- datasaurus_dozen %>%
  filter(dataset == "circle")
```

B. Plot visualization of x and y of "circle" dataset.

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



C. Calculate correlation coefficient (r) between x and y and compare it to the dino dataset.

```
circle_data %>%
  summarize(r=cor(x,y))
```

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

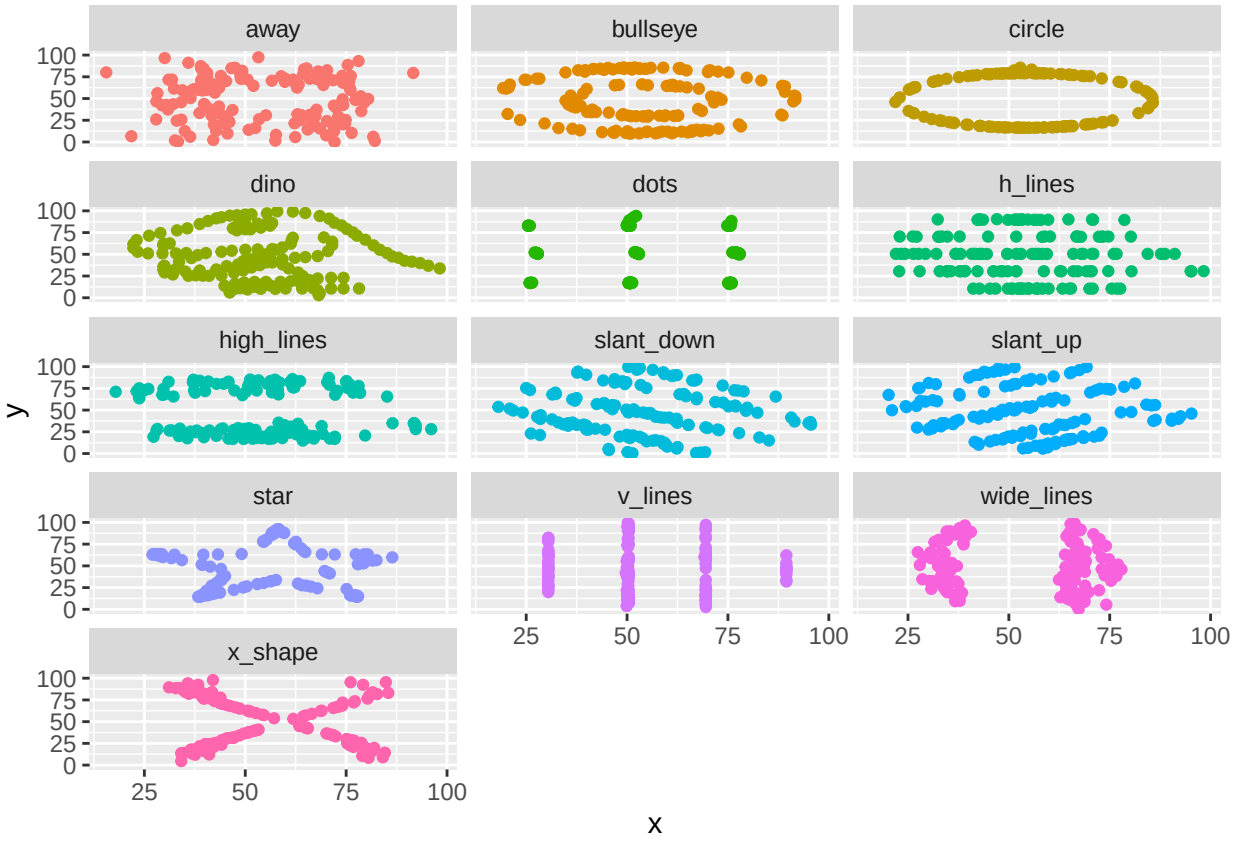
```
dino_data %>%
  summarize(r=cor(x,y))
```

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

Both of these correlation coefficients are quite low and both show a negative relationship. However, the circle dataset has a slightly higher correlation between x and y than the dino dataset does.

5. Plot all datasets at once. In order to do this we will make use of facetting.

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



Now we will also use the group_by function to generate all of the correlation coefficients for these plots.

```
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
```