exercise_set_1

2025-02-13

the following exercises are a test in disguise.

can you think of any improvements to the following code?

go through the exercises and answer them while fixing issues and improving on code workflow

make a Rmarkdown (or Quarto) version of this document with your responses

render the document in PDF and HTML formats

```
rm(list=ls())

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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(ggplot2)
```

PROBLEM 1

```
anscombe_quartet = readRDS("anscombe quartet.rds")

str(anscombe_quartet)
```

```
## tibble [44 x 3] (S3: tbl_df/tbl/data.frame)
##  $ dataset: chr [1:44] "dataset_1" "dataset_1" "dataset_1" "dataset_1" ...
##  $ x      : num [1:44] 10 8 13 9 11 14 6 4 12 7 ...
##  $ y      : num [1:44] 8.04 6.95 7.58 8.81 8.33 ...
```

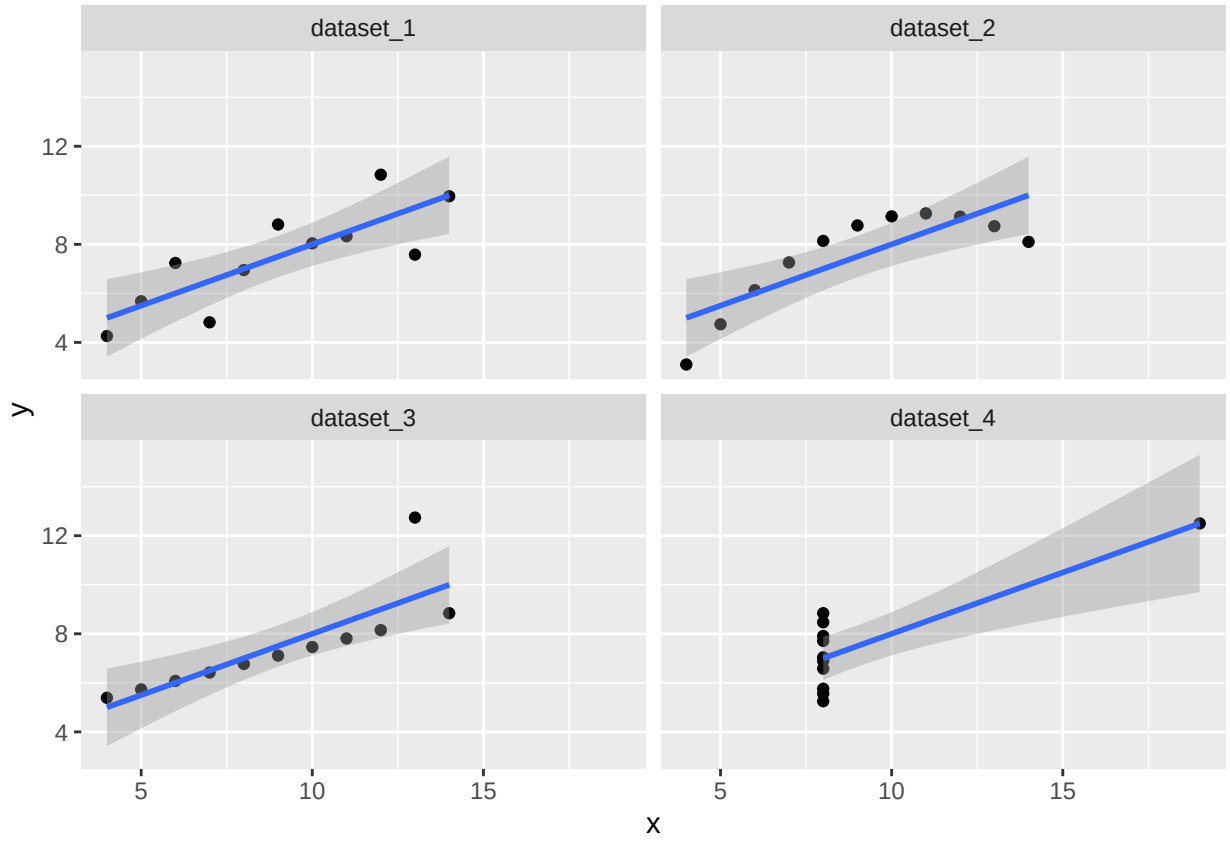
let's check some summary statistics:

```
anscombe_quartet %>%
  group_by(dataset) %>%
  summarise(
    mean_x    = mean(x),
    mean_y    = mean(y),
    min_x     = min(x),
    min_y     = min(y),
    max_x     = max(x),
    max_y     = max(y),
    crrltn    = cor(x, y)
  )
```

```
## # A tibble: 4 x 8
##   dataset   mean_x mean_y min_x min_y max_x max_y crrltn
##   <chr>      <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 dataset_1      9   7.50     4  4.26    14 10.8   0.816
## 2 dataset_2      9   7.50     4  3.1     14  9.26  0.816
## 3 dataset_3      9   7.5      4  5.39    14 12.7   0.816
## 4 dataset_4      9   7.50     8  5.25    19 12.5   0.817
```

let's plot the data with ggplot:

```
ggplot(anscombe_quartet, aes(x=x,y=y)) +
  geom_point() +
  geom_smooth(method = "lm",formula = "y ~ x") +
  facet_wrap(~dataset)
```




```
ggsave('plot_1.jpg', width = 5, height = 5, units = "in", dpi = 300)
```

Dataset 1: A typical linear relationship with moderate variability.

Dataset 2: Also linear but with slightly more spread.

Dataset 3: Mostly linear, but one outlier affects the trend.

Dataset 4: A vertical cluster of points except for one, artificially maintaining the regression line.

Linear regression appears appropriate for Dataset 1 and Dataset 2, where the relationship between x and y is consistently linear with moderate spread. For Dataset 3, an outlier significantly influences the regression line, making linear regression misleading. For Dataset 4, the data is mostly vertical except for one influential point, making the regression unreliable.

While summary statistics (e.g., mean, variance, correlation) might suggest similar relationships across all datasets, the scatter plots reveal crucial differences. This highlights the importance of visualising data rather than relying solely on numerical summaries before applying regression models.

PROBLEM 2

```
datasaurus_dozen = readRDS("datasaurus_dozen.rds")

str(datasaurus_dozen)
```

```
## tibble [1,846 x 3] (S3: tbl_df/tbl/data.frame)
##  $ dataset: chr [1:1846] "dino" "dino" "dino" "dino" ...
##  $ x      : num [1:1846] 55.4 51.5 46.2 42.8 40.8 ...
##  $ y      : num [1:1846] 97.2 96 94.5 91.4 88.3 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   dataset = col_character(),
##   ..   x = col_double(),
##   ..   y = col_double()
##   .. )
```

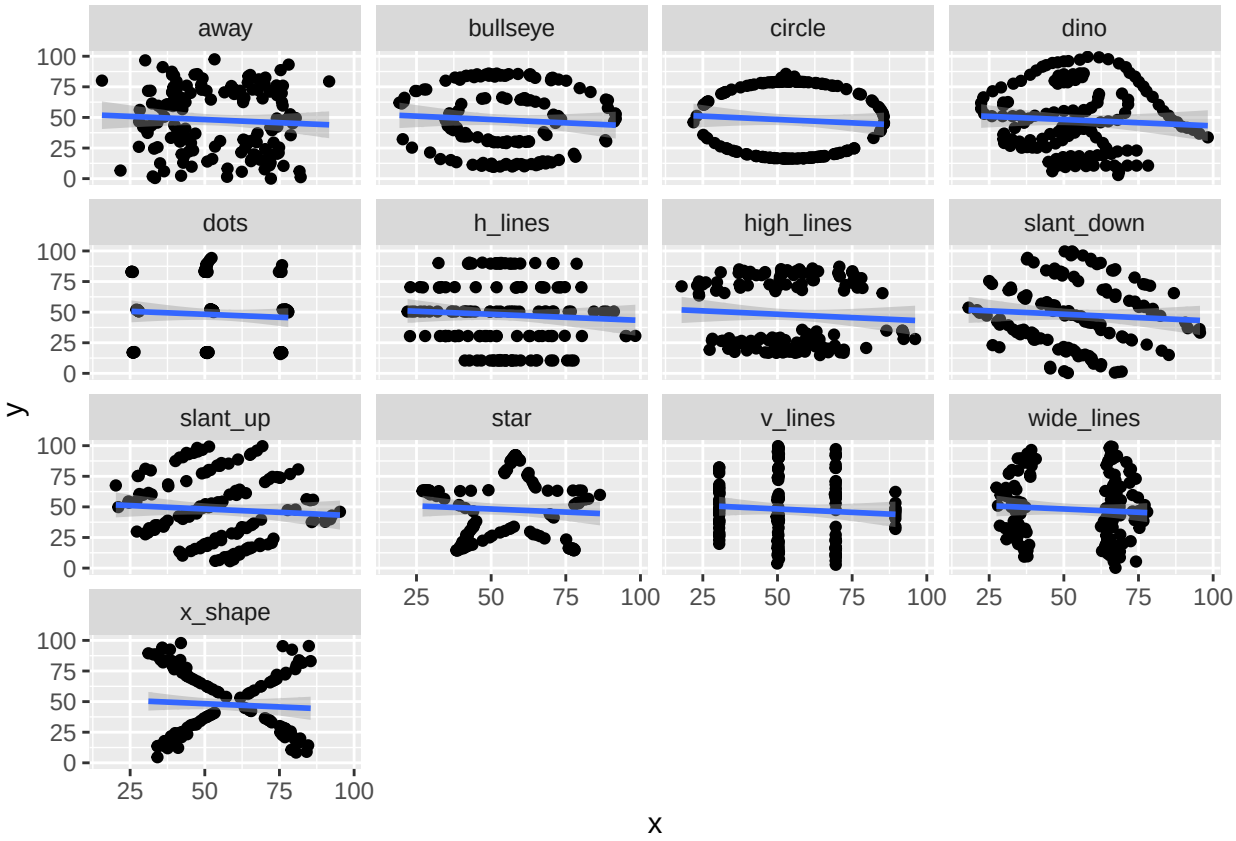
let's check some summary statistics:

```
datasaurus_dozen %>%
  group_by(dataset) %>%
  summarise(
    mean_x    = mean(x),
    mean_y    = mean(y),
    min_x     = min(x),
    min_y     = min(y),
    max_x     = max(x),
    max_y     = max(y),
    crrltn    = cor(x, y)
  )
```

```
## # A tibble: 13 x 8
##    dataset    mean_x mean_y min_x   min_y max_x max_y  crrltn
##    <chr>       <dbl>  <dbl> <dbl>   <dbl> <dbl> <dbl>   <dbl>
##  1 away         54.3   47.8  15.6  0.0151  91.6  97.5 -0.0641
##  2 bullseye     54.3   47.8  19.3  9.69    91.7  85.9 -0.0686
##  3 circle       54.3   47.8  21.9 16.3     85.7  85.6 -0.0683
##  4 dino         54.3   47.8  22.3  2.95    98.2  99.5 -0.0645
##  5 dots         54.3   47.8  25.4 15.8     78.0  94.2 -0.0603
##  6 h_lines      54.3   47.8  22.0 10.5     98.3  90.5 -0.0617
##  7 high_lines   54.3   47.8  17.9 14.9     96.1  87.2 -0.0685
##  8 slant_down   54.3   47.8  18.1  0.304   95.6  99.6 -0.0690
##  9 slant_up     54.3   47.8  20.2  5.65    95.3  99.6 -0.0686
## 10 star         54.3   47.8  27.0 14.4     86.4  92.2 -0.0630
## 11 v_lines      54.3   47.8  30.4  2.73    89.5  99.7 -0.0694
## 12 wide_lines   54.3   47.8  27.4  0.217   77.9  99.3 -0.0666
## 13 x_shape      54.3   47.8  31.1  4.58    85.4  97.8 -0.0656
```

let's plot the data with ggplot:

```
ggplot(datasaurus_dozen, aes(x=x,y=y)) +
  geom_point() +
  geom_smooth(method = "lm",formula = "y ~ x") +
  facet_wrap(~dataset)
```




```
ggsave('plot_2.jpg', width = 5, height = 5, units = "in", dpi = 300)
```