exercise one

```
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 becor
```

```
anscombe_quartet = readRDS("exercise-set_1-20250213/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 ...
```

What does the function str() do?

It gives you an overview of the data's structure, like the variables and the nature of the variables.

```
anscombe_quartet %>%
  group_by(dataset) %>%
  summarise(
    mean_x    = mean(x),
```

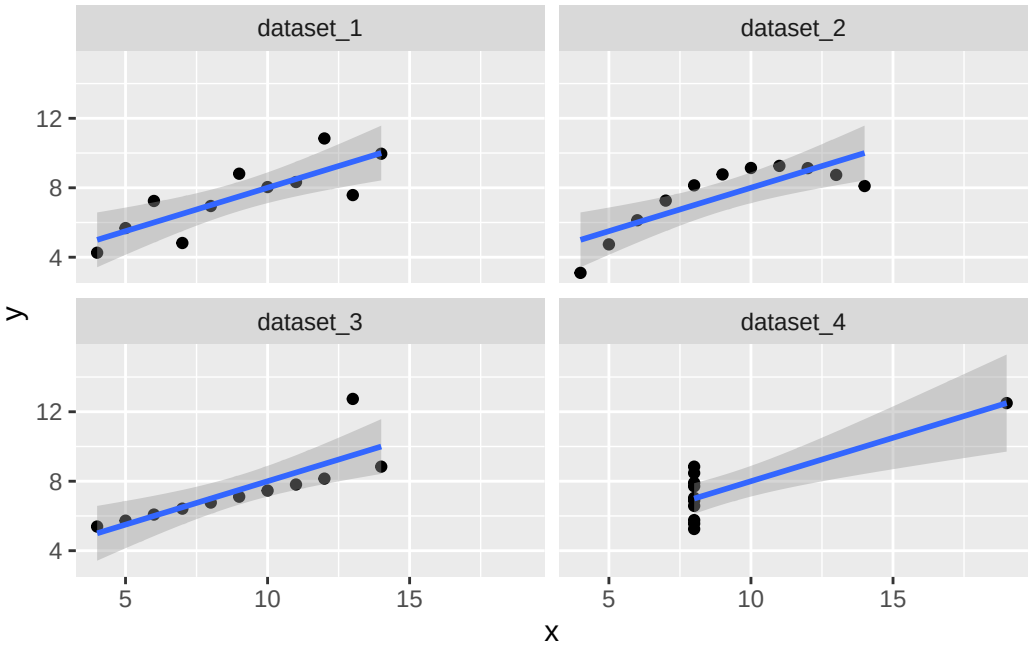
```
    mean_y    = mean(y),
    min_x     = min(x),
    min_y     = min(y),
    max_x     = max(x),
    max_y     = max(y),
    crrltn    = cor(x, y)
  )
```

```
# 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
```

What do the summary statistics tell us about the different datasets?

They seem virtually identical, with the min and max digits being very similar and means being exactly the same.

```
library(ggplot2)
anscombe_quartet <- ggplot(anscombe_quartet, aes(x=x,y=y)) +
  geom_point() +
  geom_smooth(method = "lm",formula = "y ~ x") +
  facet_wrap(~dataset)
print(anscombe_quartet)
```




```
ggsave("anscombe_plot.png", plot = anscombe_quartet, width = 5, height = 5, units = "in", dp
```

What do the plots tell us about the different datasets?

Dataset 1 follows a rough linear trend, no extreme outliers

Dataset 2 is curved, does not follow line of best fit

Dataset 3 roughly linear as well, one extreme outlier

Dataset 4 shows a cluster around one x digit with one extreme outlier

Describe the relationship between x and y in the different datasets.

1 As x increases, y increases in a linear fashion - positive relationship

2 Y increases faster when X increases

3 Nearly linear, may be a few outliers that pull the line up or down (positive)

4 None, but outliers create positive slope on fitted line

Would linear regression be an appropriate statistical model to analyse the x-y relationship in each dataset?

1 yes, simple straight line

2 no, curved pattern, not linear

3 maybe?

4 No, linear regression would not be representative of cluster pattern

What conclusions can you draw for the plots and summary statistics?

Summary statistics can be misleading and mask important aspects of a data set. Pairing them with plots is essential in order to assess any nuances in the data.

Problem 2

```
datasaurus_dozen = readRDS("exercise-set_1-20250213/datasaurus_dozen.rds")
library(tidyverse)
```

```
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()
  .. )
```

```
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
```

Print descriptive statistics and make a nicely formatted table

```
datasaurus_summary <- datasaurus_dozen %>%
  group_by(dataset) %>%
  summarise_if(is.double, list(mean = mean, sd = sd))
print(datasaurus_summary)
```

```
# A tibble: 13 x 5
   dataset    x_mean y_mean  x_sd  y_sd
   <chr>       <dbl>  <dbl> <dbl> <dbl>
 1 away         54.3   47.8  16.8  26.9
 2 bullseye     54.3   47.8  16.8  26.9
 3 circle       54.3   47.8  16.8  26.9
 4 dino         54.3   47.8  16.8  26.9
 5 dots         54.3   47.8  16.8  26.9
 6 h_lines      54.3   47.8  16.8  26.9
 7 high_lines   54.3   47.8  16.8  26.9
 8 slant_down   54.3   47.8  16.8  26.9
 9 slant_up     54.3   47.8  16.8  26.9
10 star         54.3   47.8  16.8  26.9
11 v_lines      54.3   47.8  16.8  26.9
12 wide_lines   54.3   47.8  16.8  26.9
13 x_shape      54.3   47.8  16.8  26.9
```

Calculate the correlations for x and y

```
r1 <- lm(x ~ y, datasaurus_dozen)
summary(r1)
```

```
Call:
lm(formula = x ~ y, data = datasaurus_dozen)

Residuals:
    Min      1Q  Median      3Q     Max
-37.380 -13.256  -1.538  13.000  43.359

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.23159    0.79341  70.873  < 2e-16 ***
y           -0.04110    0.01446  -2.841  0.00454 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.68 on 1844 degrees of freedom
Multiple R-squared:  0.004358,  Adjusted R-squared:  0.003819
F-statistic: 8.072 on 1 and 1844 DF,  p-value: 0.004544
```

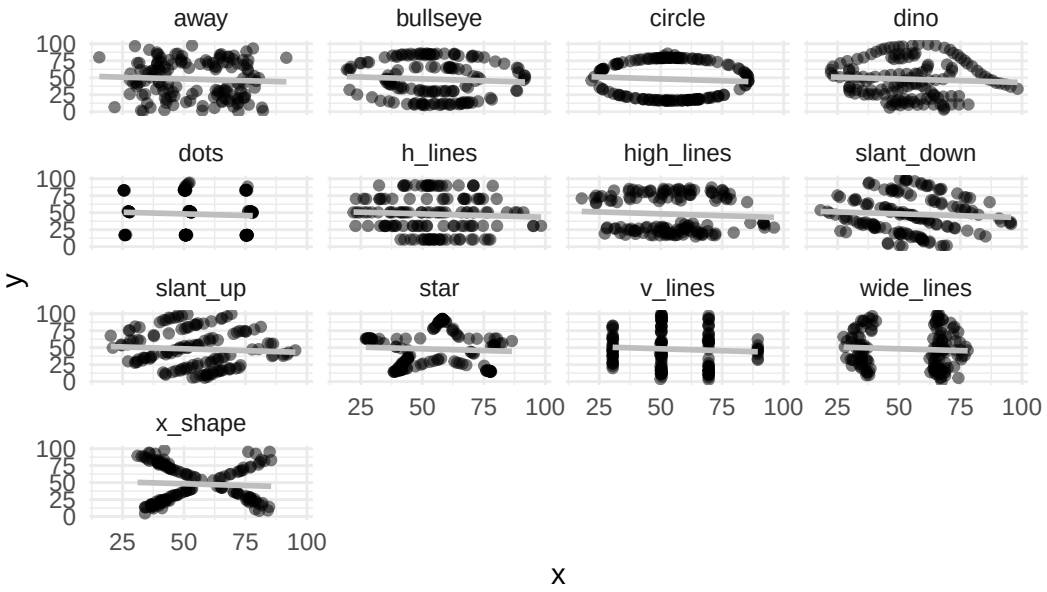
Plot their relationships including the line of best fit.

```r
library(ggplot2)
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```r
datasaurus_plot <- ggplot(datasaurus_dozen, aes(x = x, y = y)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "gray") +
  facet_wrap(~ dataset) +
  labs(
    title = "Datasaurus Dozen: Relationship between x and y",
    x = "x",
    y = "y"
  ) +
  theme_minimal()
print(datasaurus_plot)
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
ggsave("datasaurus_plot.png", plot = datasaurus_plot, width = 5, height = 5, units = "in", dp
```

```
`geom_smooth()` using formula = 'y ~ x'
```