Improved Code: Exercise_Set 1

Your Name

2025-02-26

Problem 1: Exploring Anscombe's Quartet

Understanding the Dataset

Anscombe's Quartet demonstrates how summary statistics can be misleading and highlights the importance of data visualization.

```
# Load dataset
anscombe_quartet <- readRDS("anscombe quartet.rds")

# Inspect the dataset structure
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 `str()` do? The `str()` function displays the internal structure of an R object, providing an overview of its type, dimensions, and data format.

Summary Statistics

```
summary_table <- 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),
    correlation = cor(x, y)
  )

# Display summary statistics as a nicely formatted table
summary_table %>% kable(caption = "Summary Statistics of Anscombe's Quartet") %>% kable_styling()
```

Table 1: Summary Statistics of Anscombe's Quartet

dataset	mean_x	mean_y	min_x	min_y	max_x	max_y	correlation
dataset_1	9	7.500909	4	4.26	14	10.84	0.8164205
dataset_2	9	7.500909	4	3.10	14	9.26	0.8162365

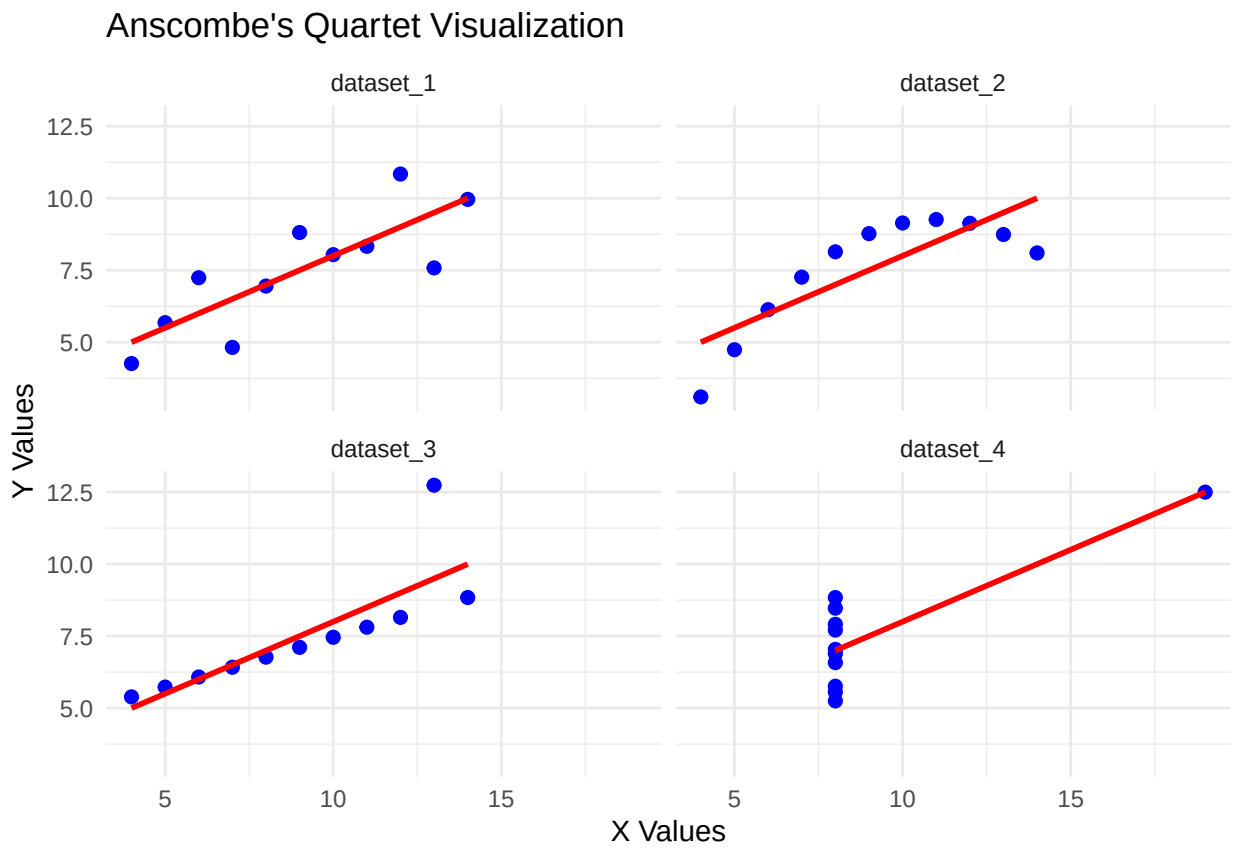
dataset_3	9	7.500000	4	5.39	14	12.74	0.8162867
dataset_4	9	7.500909	8	5.25	19	12.50	0.8165214

What do the summary statistics tell us? Despite having nearly identical means, variances, and correlations, the datasets are structurally very different. This emphasizes the importance of visualization.

Visualizing the Data

```
anscombe_plot <- ggplot(anscombe_quartet, aes(x = x, y = y)) +
  geom_point(color = "blue", size = 2) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "red") +
  facet_wrap(~dataset) +
  theme_minimal() +
  labs(title = "Anscombe's Quartet Visualization", x = "X Values", y = "Y Values")

print(anscombe_plot)
```



Saving the Plot

```
ggsave("anscombe_quartet.png", anscombe_plot, width = 5, height = 5, dpi = 300)
```

Interpretation of Plots: - Each dataset has the same summary statistics but vastly different distributions. - Some show non-linear trends, while others have outliers that influence correlations. - This reinforces why **visualizing data** is crucial before making conclusions.

Would linear regression be appropriate? - In some cases, yes (e.g., dataset 1), but others display non-linearity or outliers that violate regression assumptions.

Problem 2: Exploring the Datasaurus Dozen

Load and Inspect Data

```
datasaurus_dozen <- readRDS("datasaurus_dozen.rds")

# Structure of the dataset
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()
##   .. )
```

Summary Statistics for Datasaurus Dozen

```
datasaurus_summary <- 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),
    correlation = cor(x, y)
  )

# Display summary in a styled table
kable(datasaurus_summary, caption = "Summary Statistics of Datasaurus Dozen") %>% kable_styling()
```

Table 2: Summary Statistics of Datasaurus Dozen

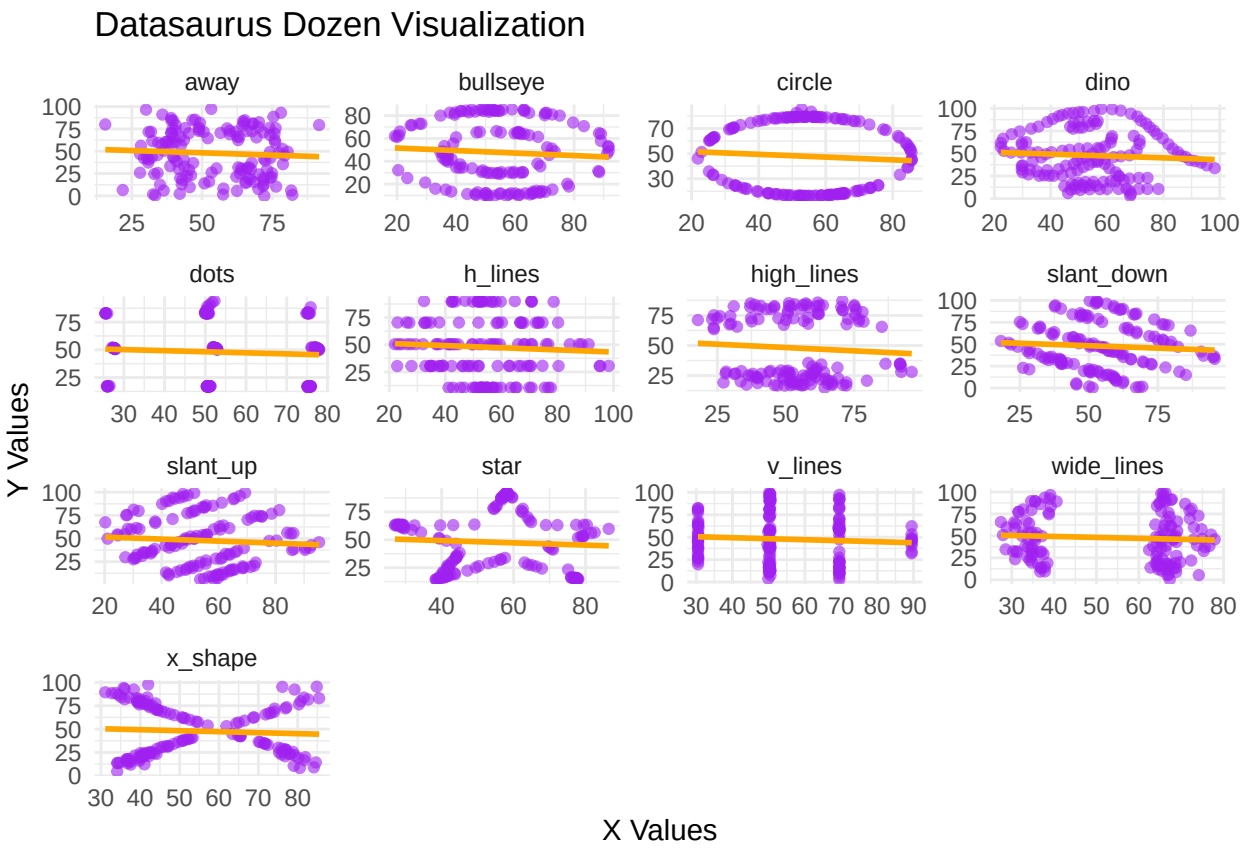
dataset	mean_x	mean_y	min_x	min_y	max_x	max_y	correlation
away	54.26610	47.83472	15.56075	0.0151193	91.63996	97.47577	-0.0641284
bullseye	54.26873	47.83082	19.28820	9.6915471	91.73554	85.87623	-0.0685864
circle	54.26732	47.83772	21.86358	16.3265464	85.66476	85.57813	-0.0683434
dino	54.26327	47.83225	22.30770	2.9487000	98.20510	99.48720	-0.0644719
dots	54.26030	47.83983	25.44353	15.7718920	77.95444	94.24933	-0.0603414
h_lines	54.26144	47.83025	22.00371	10.4639152	98.28812	90.45894	-0.0617148
high_lines	54.26881	47.83545	17.89350	14.9139625	96.08052	87.15221	-0.0685042
slant_down	54.26785	47.83590	18.10947	0.3038724	95.59342	99.64418	-0.0689797
slant_up	54.26588	47.83150	20.20978	5.6457775	95.26053	99.57959	-0.0686092

star	54.26734	47.83955	27.02460	14.3655905	86.43590	92.21499	-0.0629611
v_lines	54.26993	47.83699	30.44965	2.7347602	89.50485	99.69468	-0.0694456
wide_lines	54.26692	47.83160	27.43963	0.2170063	77.91587	99.28376	-0.0665752
x_shape	54.26015	47.83972	31.10687	4.5776614	85.44619	97.83761	-0.0655833

Visualizing the Different Datasets

```
datasaurus_plot <- ggplot(datasaurus_dozen, aes(x = x, y = y)) +
  geom_point(color = "purple", alpha = 0.6) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "orange") +
  facet_wrap(~dataset, scales = "free") +
  theme_minimal() +
  labs(title = "Datasaurus Dozen Visualization", x = "X Values", y = "Y Values")

print(datasaurus_plot)
```



Save the Plot

```
ggsave("datasaurus_dozen.png", datasaurus_plot, width = 6, height = 6, dpi = 300)
```

Observations: - Even though datasets have similar summary statistics, their visual patterns vary significantly. - Some datasets form shapes (e.g., a dinosaur!), while others show distinct trends. - Again, this highlights why **visualization matters** beyond summary statistics.

Conclusion

- Summary statistics alone can be misleading; always visualize your data.
- Linear regression isn't always the best choice—check for patterns, outliers, and non-linear relationships.
- Graphs provide insights that numbers alone cannot!

This document is rendered as **HTML** and **PDF** to showcase results in different formats.