Agricultural Dataset of an Argentinian Corn Field

Yield monitor data for a corn field in Argentina with variable nitrogen.

Jack Nguyen, Nikita Kiselov

2025-12-18

1 Introduction

This dataset covers the experimental study of corn yield and nitrogen fertilizer treatment with field characteristics from the Las Rosas farm in Rio Cuarto, Cordoba, Argentina. In it, data was collected using yield monitor from strip trials during the harvests in years 1999 and 2001 to determine the potential of spatial econometric analysis in estimating the site-specific crop response functions when it comes to applying nitrogen treatment. There are 3443 operations and 8 variables, including the year, latitude, longitude, yield in quintals/ha, amount of nitrogen in kg/ha, topographic factor, brightness value, rep factor, and nitrogen as a factor. Of these variables, both the topographic, rep, and nitrogen factors, are qualitative while the rest are quantitative.

2 Data Statistics

Table 1: Summary Statistics of Yield by Nitrogen Fertilizer Level

Level	Mean	Median	SD	IQR	Range
N0	64.97	61.95	20.94	36.00	12.66 - 108.84
N1	68.62	65.19	19.20	29.75	27.44 - 110.54
N2	69.65	67.06	19.31	28.11	31.79 - 112.85
N3	70.34	66.83	19.23	27.72	19.41 - 110.12
N4	72.56	69.17	19.15	27.53	32.05 - 117.9
N5	72.83	70.35	20.14	29.79	31.79 - 117.19

2.1 Advanced Tables with kableExtra

For even more formatting options, use the `{kableExtra}` package:

Table 2: Advanced table formatting with kableExtra

X Category	Count	Mean Y	SD Y	Min Y	Max Y
Low	26	19.94	4.46	10.76	26.75
Medium	62	24.11	4.32	12.96	33.44
High	12	30.43	6.15	20.77	43.23

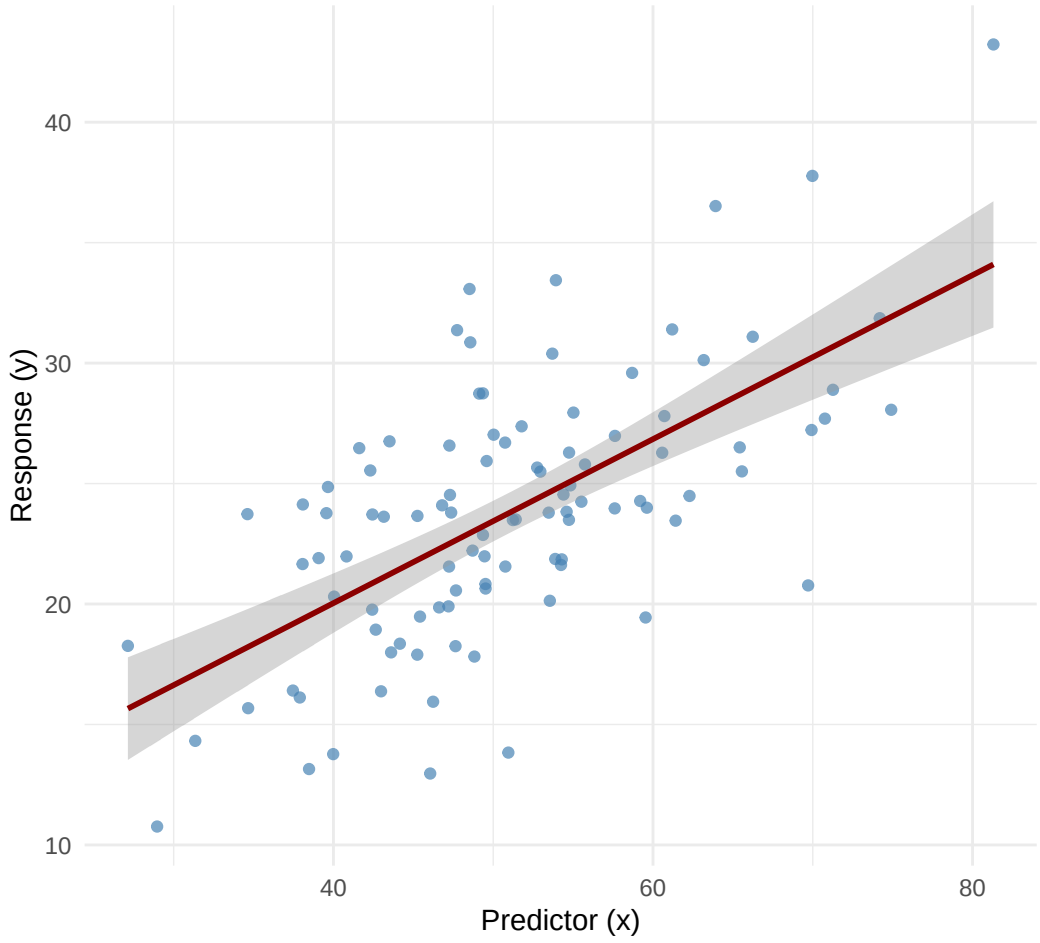
Notice how Table 2 includes footnotes and styling. The `html-table-processing: none` option prevents Quarto from re-processing tables created with `{kableExtra}`.

3 Creating Figures with R

Figures generated from R code should include proper labels, captions, and alt text for accessibility.

3.1 Basic Plot with Caption

Figure 1: Relationship between predictor x and response y



As shown in Figure 1, there appears to be a positive linear association between the predictor and the response.

3.2 Plot with Alt Text from Code

You can also specify alt text within the `labs()` function and extract it automatically:

The distribution in Figure 2 shows that x values are approximately normally distributed.

3.3 Multiple Plots

4 Including External Images

To include external images (not generated by R), you can use markdown syntax:

```
![Caption for your image](path/to/image.png){#fig-external}
```

Or use knitr:

Figure 2: Distribution of predictor variable x

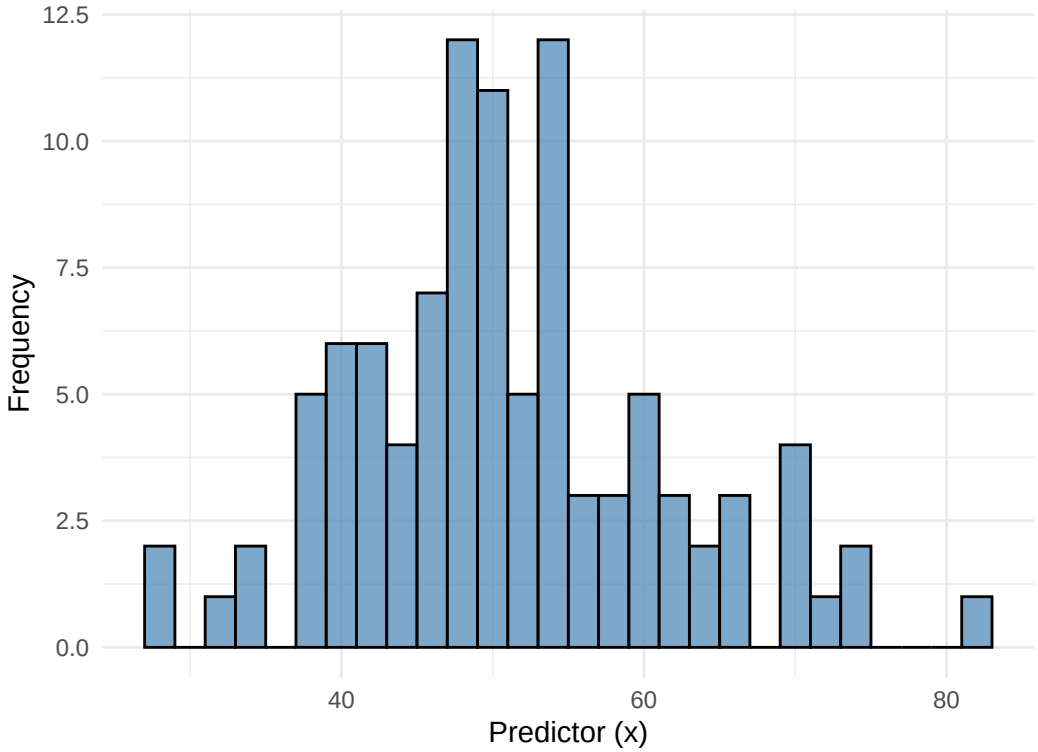
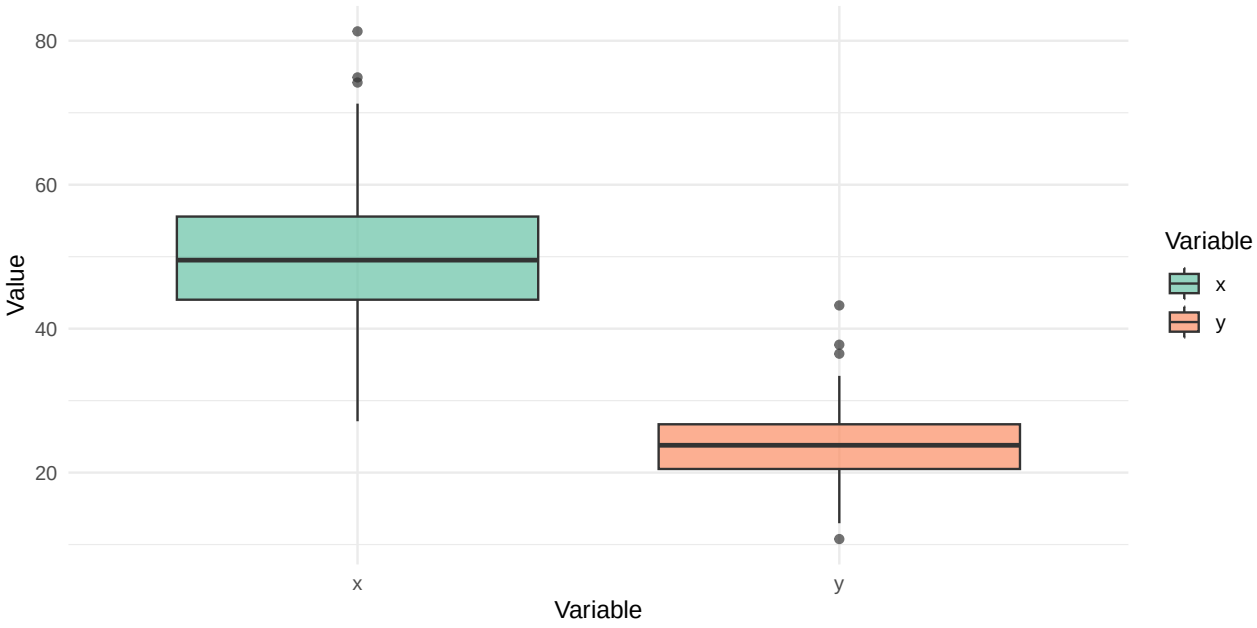


Figure 3: Comparison of distributions



```r
knitr::include_graphics("path/to/image.png")
```

5 Cross-Referencing

Cross-referencing allows you to refer to tables, figures, and sections by name rather than hard-coding numbers.

5.1 Referencing Tables

To reference a table, use `@tbl-label`. For example:

- See Table 1 for summary statistics
- Compare results in Table 2

5.2 Referencing Figures

To reference a figure, use `@fig-label`. For example:

- The scatterplot in Figure 1 shows a clear trend
- Distribution details are in Figure 2

5.3 Referencing Sections

You can also reference sections by adding labels to headings. For example, the heading:

```
# Data Analysis {#sec-analysis}
```

Can be referenced as: See **?@sec-analysis** for details.

6 Key Points to Remember

1. **Table Labels**: Must start with `tbl-` (e.g., `#| label: tbl-results`)
2. **Figure Labels**: Must start with `fig-` (e.g., `#| label: fig-plot1`)
3. **Section Labels**: Use `{#sec-name}` after heading text
4. **Alt Text**: Always include `fig-alt` for accessibility
5. **Captions**: Use `fig-cap` for figures and `tbl-cap` for tables
6. **Code Appendix**: Automatically collects all your code at the end

7 Formatting Tips

- Use **bold** text with `**text**` or `__text__`
- Use *italic* text with `*text*` or `_text_`
- Create lists with `-` or numbered lists with `1.`, `2.`, etc.
- Add inline math with `$equation$` (e.g., $\mu = 50$, $\sigma^2 = 100$)
- Add display math with `$$equation$$`

8 Conclusion

This template provides a foundation for creating professional statistical reports using Quarto. Key features include:

- Professional tables using both markdown and `kable()`
- Figures with proper captions and alt text for accessibility
- Cross-referencing capabilities for tables, figures, and sections

- Automatic code appendix generation

When you start your own project, replace this content with your analysis while maintaining the structure demonstrated here.

9 Code Appendix

The code below shows all the R code used to generate the tables and figures in this document. The code is displayed here for transparency and reproducibility but is hidden in the main body of the report.

```
library(tidyverse)
library(knitr)
library(kableExtra)
library(agridat)
data(lasrosas.corn)
dat <- lasrosas.corn
sample_data <- tibble(
  id = 1:100,
  x = rnorm(100, mean = 50, sd = 10),
  y = 3 + 0.4 * x + rnorm(100, sd = 5)
)

dat %>% group_by(nf) %>%
  summarise(
    Mean = mean(yield, na.rm = TRUE),
    Median = median(yield, na.rm = TRUE),
    SD = sd(yield, na.rm = TRUE),
    IQR = IQR(yield, na.rm = TRUE),
    Range = paste(min(yield, na.rm = TRUE), "-", max(yield, na.rm = TRUE)),
    .groups = "drop"
  ) %>% kable(
      digits = 2,
      col.names = c("Level", "Mean", "Median", "SD", "IQR", "Range"))
sample_data %>%
  group_by(cut(x, breaks = 3, labels = c("Low", "Medium", "High"))) %>%
  summarize(
    Count = n(),
    Mean_Y = mean(y),
    SD_Y = sd(y),
    Min_Y = min(y),
    Max_Y = max(y),
    .groups = "drop"
  ) %>%
  kable(
    digits = 2,
    col.names = c("X Category", "Count", "Mean Y", "SD Y", "Min Y", "Max Y"),
    align = "lccccc"
  ) %>%
  kable_classic() %>%
  add_footnote(
    label = c("Data grouped by tertiles of x values."),
    notation = "none",
    threeparttable = TRUE
  )
ggplot(sample_data, aes(x = x, y = y)) +
  geom_point(alpha = 0.7, color = "steelblue") +
  geom_smooth(method = "lm", se = TRUE, color = "darkred") +
  theme_minimal() +
  labs(
```

```
    x = "Predictor (x)",
    y = "Response (y)"
  )
ggplot(sample_data, aes(x = x)) +
  geom_histogram(
    binwidth = 2,
    fill = "steelblue",
    color = "black",
    alpha = 0.7
  ) +
  theme_minimal() +
  labs(
    x = "Predictor (x)",
    y = "Frequency",
    alt = "Histogram showing the distribution of x values, which appears approximately normal and cente
  )
# Reshape data for plotting
plot_data <- sample_data %>%
  select(x, y) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value")

ggplot(plot_data, aes(x = variable, y = value, fill = variable)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2") +
  labs(
    x = "Variable",
    y = "Value",
    fill = "Variable"
  )
```