Template RMarkdown File

Robert J. Dellinger

April 06, 2025

Contents

Introduction

This document outlines the methodology for data cleaning, exploration, and visualization. It is structured to ensure transparency and reproducibility of all analyses.

Methodology

Briefly describe the methods used in the project, including data sources, cleaning steps, and techniques applied to handle missing or inconsistent data.

Math Equations with LaTeX

Here is an inline equation: $E = mc^2$.

And here is a displayed equation:

$$f(x) = \int_{-\infty}^{\infty} e^{-x^2}\, dx$$

Both inline and block math can be rendered seamlessly with LaTeX.

You can also create multiline equations with alignment:

$$a = b + c$$

$$d = e + f$$

Loading Data

The first step in any data analysis is to load the data. This section outlines the process of importing the data into R, including any necessary transformations or adjustments to ensure compatibility with the analysis.

```
# Example of loading data from a CSV file raw_data <-
# read_csv(here('Data', 'Raw', 'data_file.csv')) raw_data <-
# read_excel(here('Data', 'Raw', 'data_file.xlsx'))
```

Cleaning Data

The data cleaning process involves several steps to ensure the data is in a suitable format for analysis. This includes handling missing values, correcting data types, and removing duplicates.

```
# Example of cleaning data - Removing duplicates - Handling missing
# values - Converting data types and cleaning white space - Renaming
# columns, etc.


# cleaned_data <- raw_data %>% clean_names() %>% mutate(column_name =
# as_factor(column_name)) %>% mutate(date_column =
# as.Date(date_column, format = '%Y-%m-%d')) %>% # convert to date
# mutate(numeric_column = as.numeric(numeric_column)) %>% # convert
# to numeric mutate(accross(everything(), ~str_squish(.))) %>% #
# clean whitespace drop_na()
```

Data Exploration

Data exploration is a crucial step in understanding the dataset and identifying patterns or anomalies. This section includes summary statistics, visualizations, and any other relevant analyses to gain insights into the data.

```
# Explore the cleaned data using basic summaries:
# glimpse(cleaned_data) summary(cleaned_data) str(cleaned_data)
```

Data Visualization

Data visualization is an essential part of data analysis, allowing for the communication of findings in a clear and effective manner. This section includes various plots and charts to

illustrate key insights from the data.

```
# Example of creating a summary table summary_table <- cleaned_data
# %>% group_by(group_var) %>% summarise(mean_value = mean(value_var,
# na.rm = TRUE)) %>% ungroup() %>% kable() %>%
# kable_styling(full_width = F, position = 'left')


# save_kable(summary_table, file = here(output_path_tables,
# 'summary_table.html'), bootstrap_options = c('striped', 'hover',
# 'condensed'), full_width = F, position = 'left')
```

```
# Example of visualization plot
data("diamonds")


# First Plot (p1) - Scatter plot with smoothing line
p1 <- ggplot(
  subset(diamonds, carat >= 2.2),
  aes(x = table, y = price, colour = cut)
) +
  geom_point(alpha = 0.7) +  # Points with transparency
  geom_smooth(method = "loess", alpha = 0.05, linewidth = 1, span = 1, formula = y ~ x)
  scale_color_npg() +  # Apply NPG color scale
  theme_bw() +  # White background theme
  theme(
    plot.title = element_text(hjust = 0.5),  # Center the title
    legend.position = "top",  # Position the legend at the top
    text = element_text(size = 10)  # Set text size
  )
```

```r
# Second Plot (p2) - Histogram of depth for specific carat ranges
p2 <- ggplot(
  subset(diamonds, carat > 2.2 & depth > 55 & depth < 70),
  aes(x = depth, fill = cut)
) +
  geom_histogram(colour = "black", binwidth = 1, position = "dodge") +  # Dodged histog
  scale_fill_npg() +  # Apply NPG fill color scale
  theme_bw() +  # White background theme
  theme(
    plot.title = element_text(hjust = 0.5),  # Center the title
    legend.position = "top",  # Position the legend at the top
    text = element_text(size = 10)  # Set text size
  )

p1_npg <- p1 + scale_color_npg()
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

```r
p2_npg <- p2 + scale_fill_npg()
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```r
grid.arrange(p1_npg, p2_npg, ncol = 2)

# Example of saving a plot
# ggsave(filename = here(output_path_images, "plot_name.png"), plot = last_plot(), wid
```

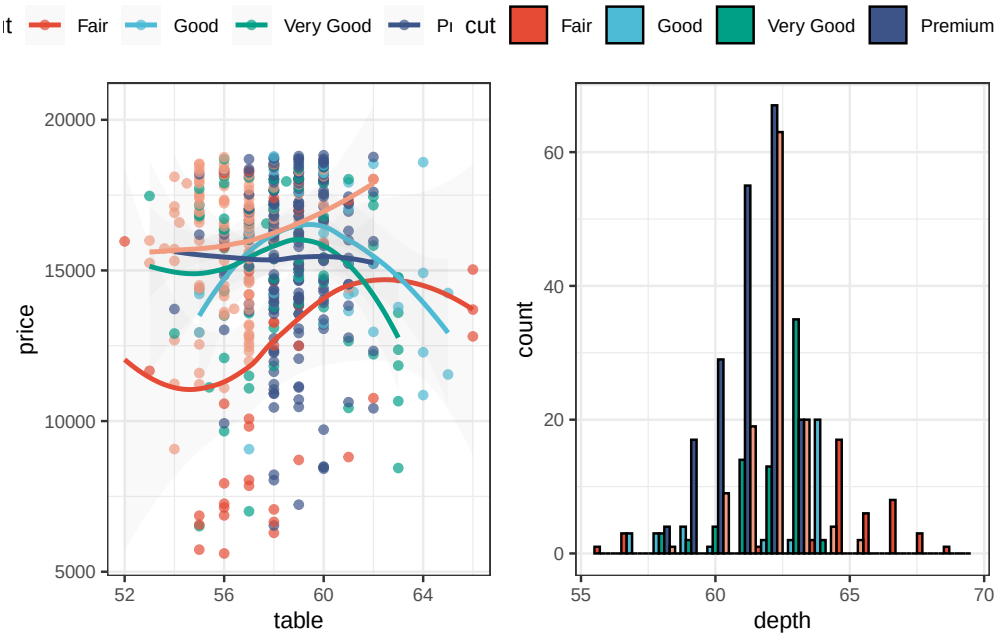

Figure 1: Example of a scatter plot

Literature Cited

Citing the packages and data used in the analysis is important for reproducibility and transparency. The following code generates a bibliography of all loaded packages. Items can be cited directly within the documentation using the syntax `@key` where key is the citation key in the first line of the entry, e.g., R Core Team (2024), Wickham et al. (2024), Wickham (2023), Müller (2020). To put citations in parentheses, use `[@key]` instead.

Müller, K. (2020). *Here: A simpler way to find your files*. https://here.r-lib.org/

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Wickham, H. (2023). *Tidyverse: Easily install and load the tidyverse*. https://tidyverse.tidyverse.org

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H., Dunnington, D., & van den Brand, T. (2024). *ggplot2: Create elegant data visualisations using the grammar of graphics*. https://ggplot2.tidyverse.org