Client Report (Placeholder)

Parham Pishrobat, Sarah Mosri, Johnson Chen

2024-02-25

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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(lubridate)
library(rlang)
```

```
##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

1. Introduction

- Background of the study.
- Objective(s) of the study.
- Statistical questions to answer.

The study conducted by Jue, Press, and Loewenstein investigates the effectiveness of various strategies to encourage consumers to choose zero-calorie beverages over sugary alternatives. The research focuses on the impact of visual calorie content presentations and financial incentives. By examining the efficacy of price discounts and visual messages that detail the caloric content and physical activity required to offset these calories, the study aims to identify effective methods to shift consumer preferences towards healthier beverage choices.

2. Data Description and Summaries

- Data collection method.

- Study design.
- Sample size.
- Variables measured.
- Missing data.

```
beverage_sales <- read.csv("../rawdata/june1data.csv")
head(beverage_sales)
```

```
##   Count DofW Site Intervention ZeroCal Sugary Juice100 Ojuice Sports Total
## 1     1    4 chop       preint     734    556      176    112     67  5112
## 2     2    5 chop       preint     651    601      165    121     64  5118
## 3     3    6 chop       preint     194    217       64     47     24  1451
## 4     4    7 chop       preint     178    193       53     52     19  1204
## 5     5    1 chop       preint     652    563      147    100     64  4626
## 6     6    2 chop       preint     789    496      165     87     53  5050
```

```
dim(beverage_sales)
```

```
## [1] 631  10
```

3. Exploratory Analysis

- Suggesting tables and figures for data summarization.
- Initial findings.

```
#' Summarize Data
#'
#' Provides a summary for numeric and categorical variables in the dataset.
#' @param df Data frame to summarize.
#' @return A list containing summaries for numeric and categorical variables.
summarize_data <- function(df) {
  numerical_summary   <- df %>% select_if(is.numeric) %>% summary()
  categorical_summary <- df %>% select_if(is.character) %>% summary()
  list(numerical = numerical_summary, categorical = categorical_summary)
}
summarize_data(beverage_sales)$numerical
```

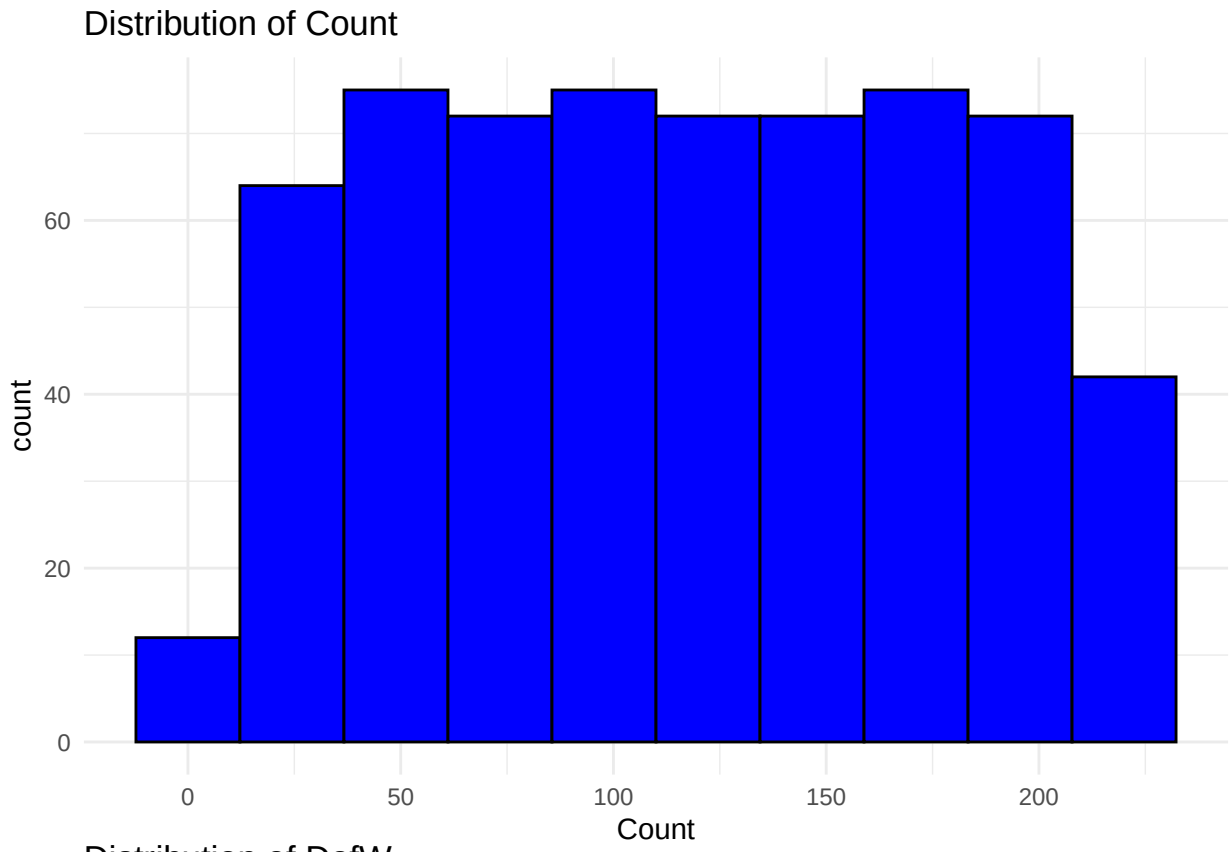
```
##      Count            DofW          ZeroCal           Sugary
##  Min.   :  1.0   Min.   :1.000   Min.   :   9.0   Min.   : 15.0
##  1st Qu.: 64.0   1st Qu.:2.000   1st Qu.: 131.5   1st Qu.:139.0
##  Median :116.0   Median :4.000   Median : 207.5   Median :188.5
##  Mean   :116.2   Mean   :4.022   Mean   : 315.8   Mean   :262.3
##  3rd Qu.:169.0   3rd Qu.:6.000   3rd Qu.: 553.5   3rd Qu.:413.2
##  Max.   :221.0   Max.   :7.000   Max.   :1000.0   Max.   :782.0
##                                  NA's   :9        NA's   :9
##     Juice100         Ojuice          Sports          Total
##  Min.   :  0.0   Min.   : 24.00   Min.   :  0.00   Min.   : 200
##  1st Qu.: 45.0   1st Qu.: 51.00   1st Qu.: 13.00   1st Qu.:1309
##  Median : 67.0   Median :100.00   Median : 19.50   Median :2179
##  Mean   :100.2   Mean   : 87.91   Mean   : 33.05   Mean   :2463
##  3rd Qu.:175.0   3rd Qu.:114.00   3rd Qu.: 61.00   3rd Qu.:2483
##  Max.   :305.0   Max.   :152.00   Max.   :162.00   Max.   :6229
##  NA's   :210     NA's   :410      NA's   :217      NA's   :10
```

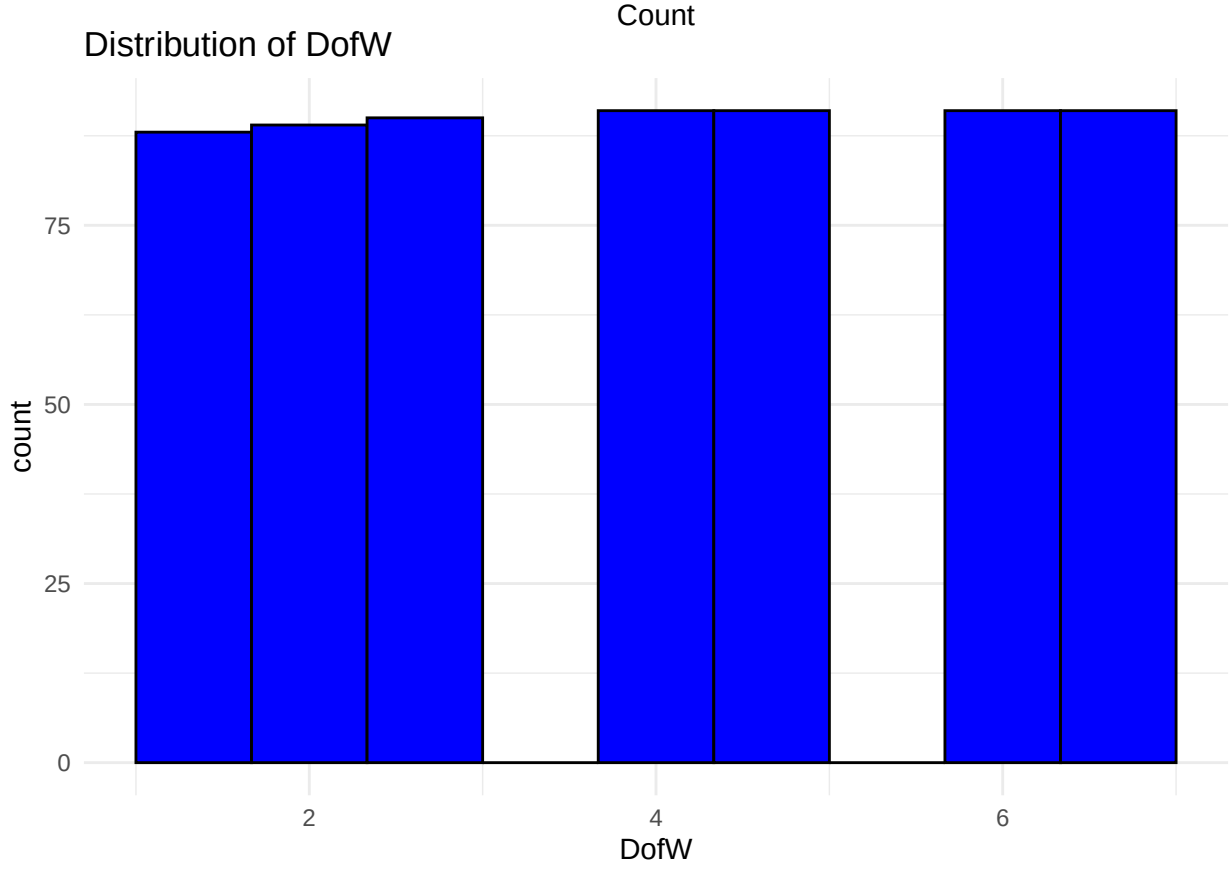
```r
summarize_data(beverage_sales)$categorical
```

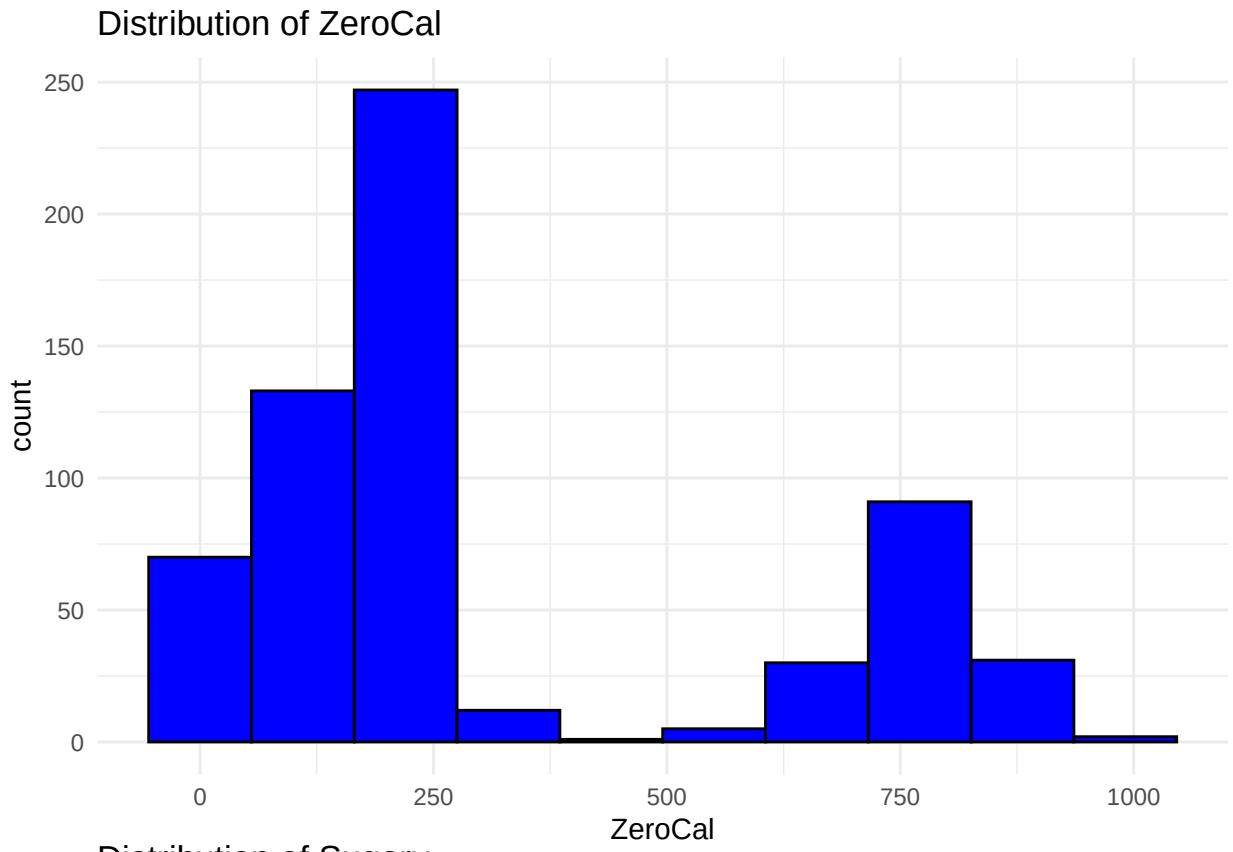
```
##      Site           Intervention
##  Length:631         Length:631
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

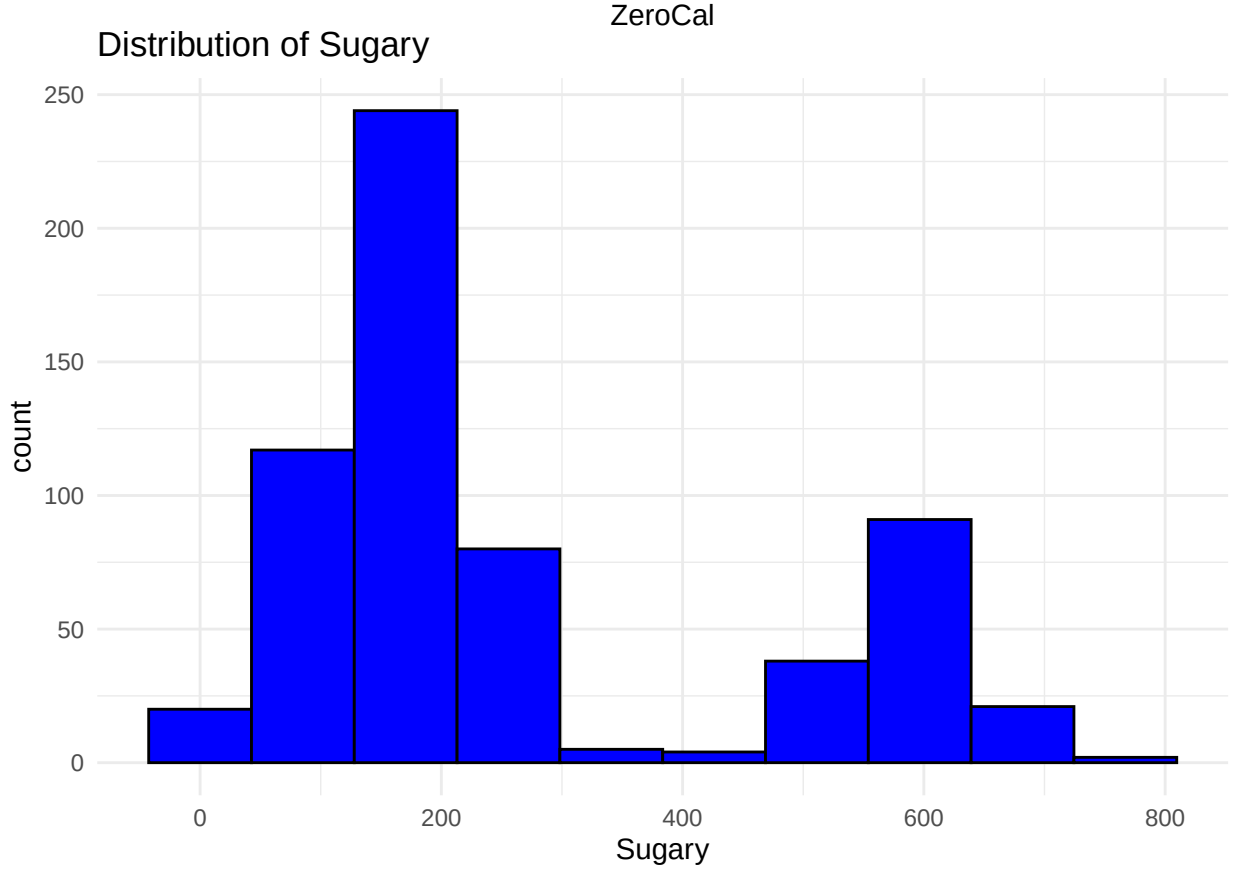
```r
#' Plot Histograms for Numeric Variables
#'
#' Plots histograms for all numeric variables in the dataset to explore distributions.
#' @param df Data frame containing numeric variables.
plot_histograms <- function(df) {
  numeric_vars <- select_if(df, is.numeric) %>% names()

  for (var in numeric_vars) {
    print(ggplot(df, aes_string(x = var)) +
      geom_histogram(bins = 10, fill = "blue", color = "black", na.rm = T) +
      theme_minimal() +
      labs(title = paste("Distribution of", var)))
  }
}
plot_histograms(beverage_sales)
```

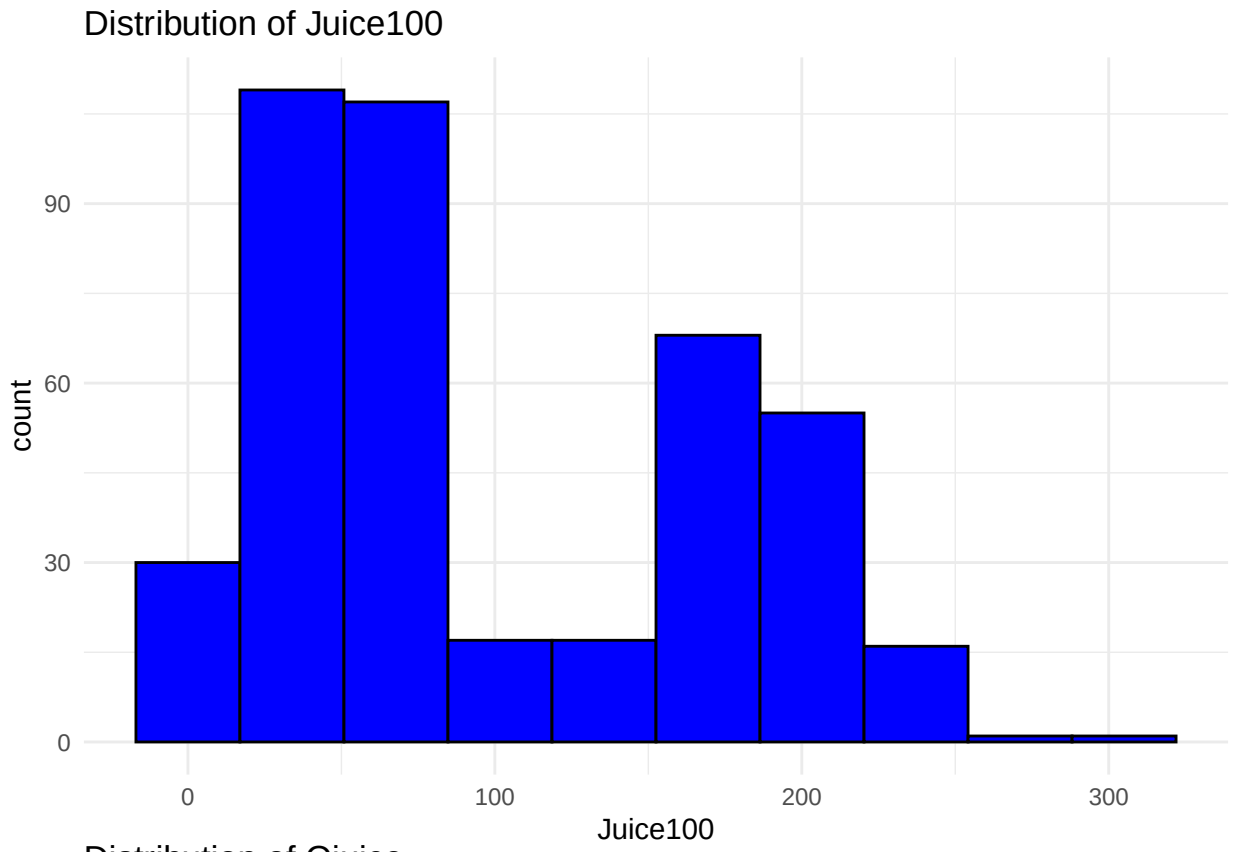
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

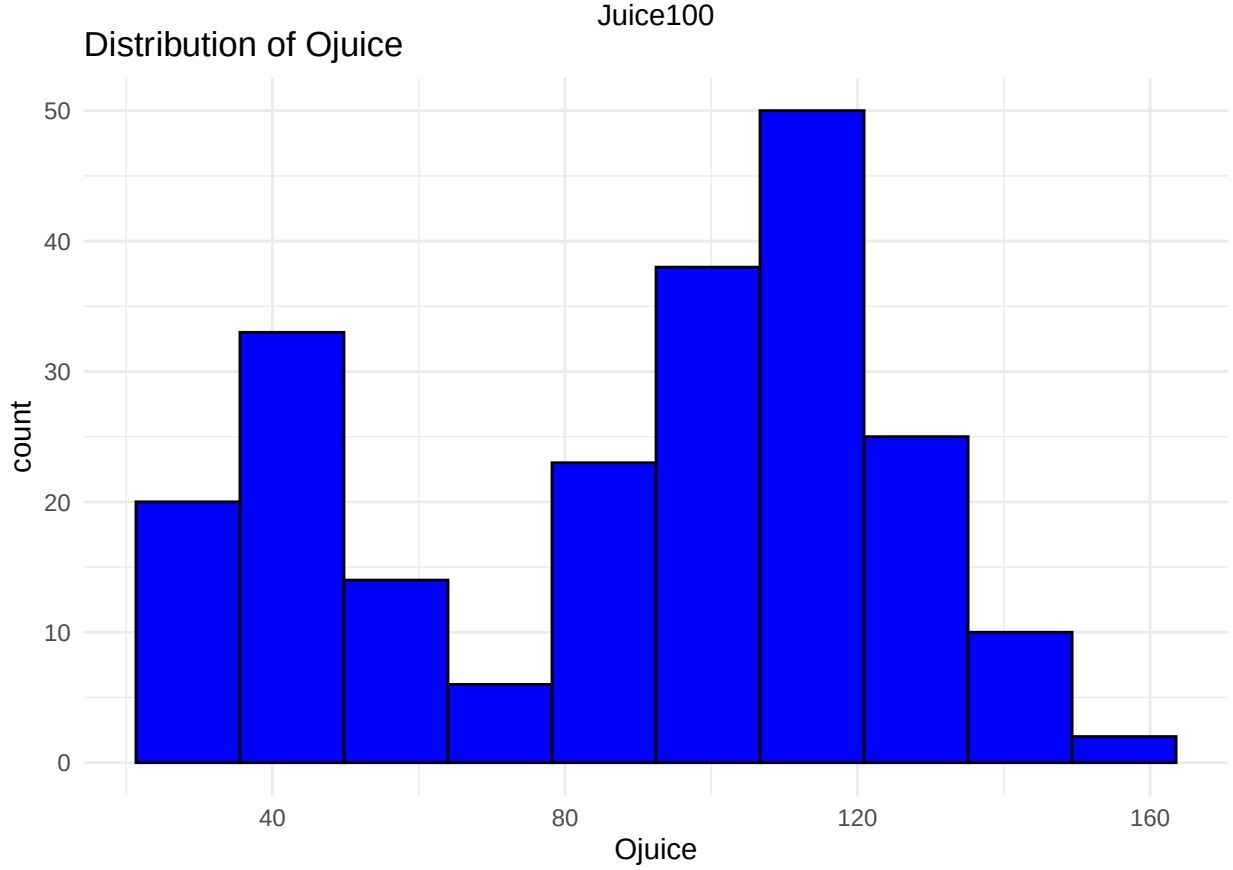

Distribution of Count
count
60
40
20
0
0
50
100
150
200
Count

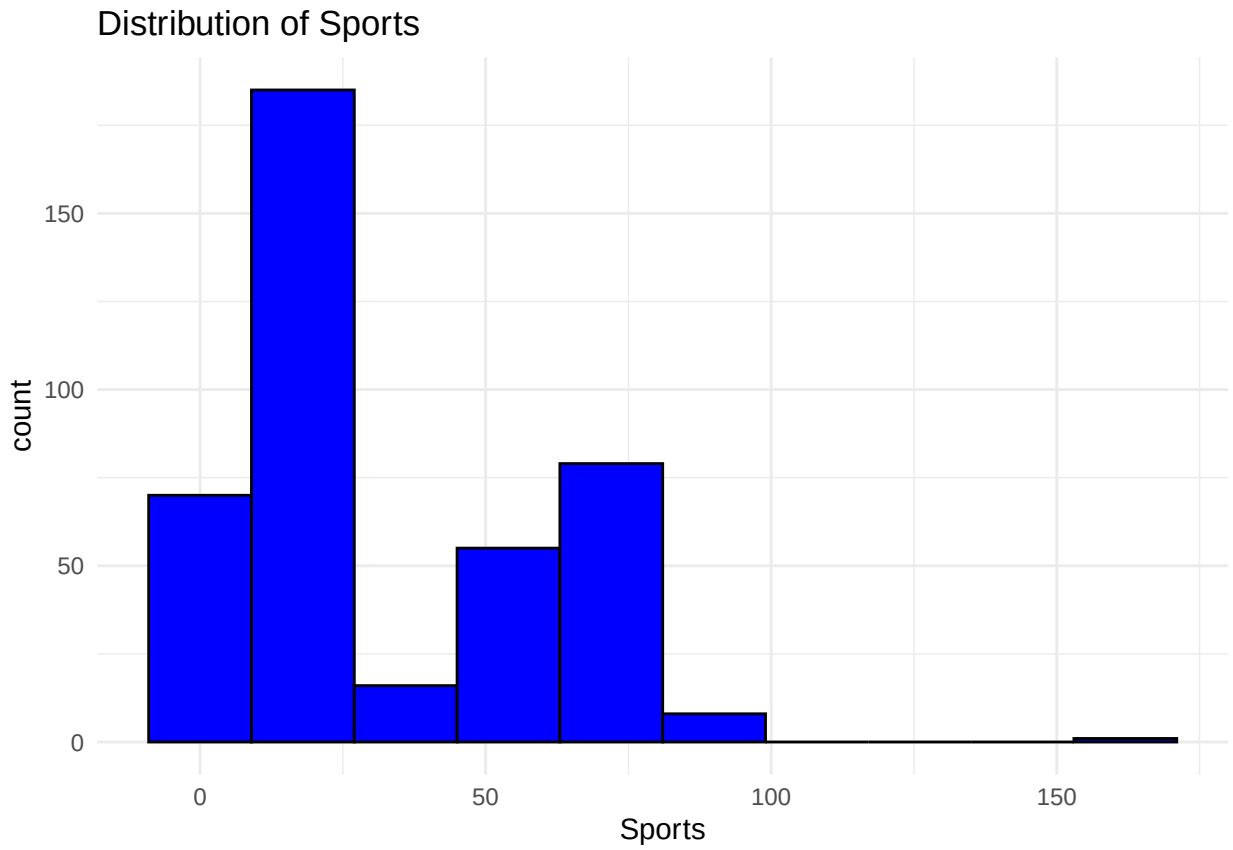

Distribution of DofW
count
75
50
25
0
2
4
6
DofW

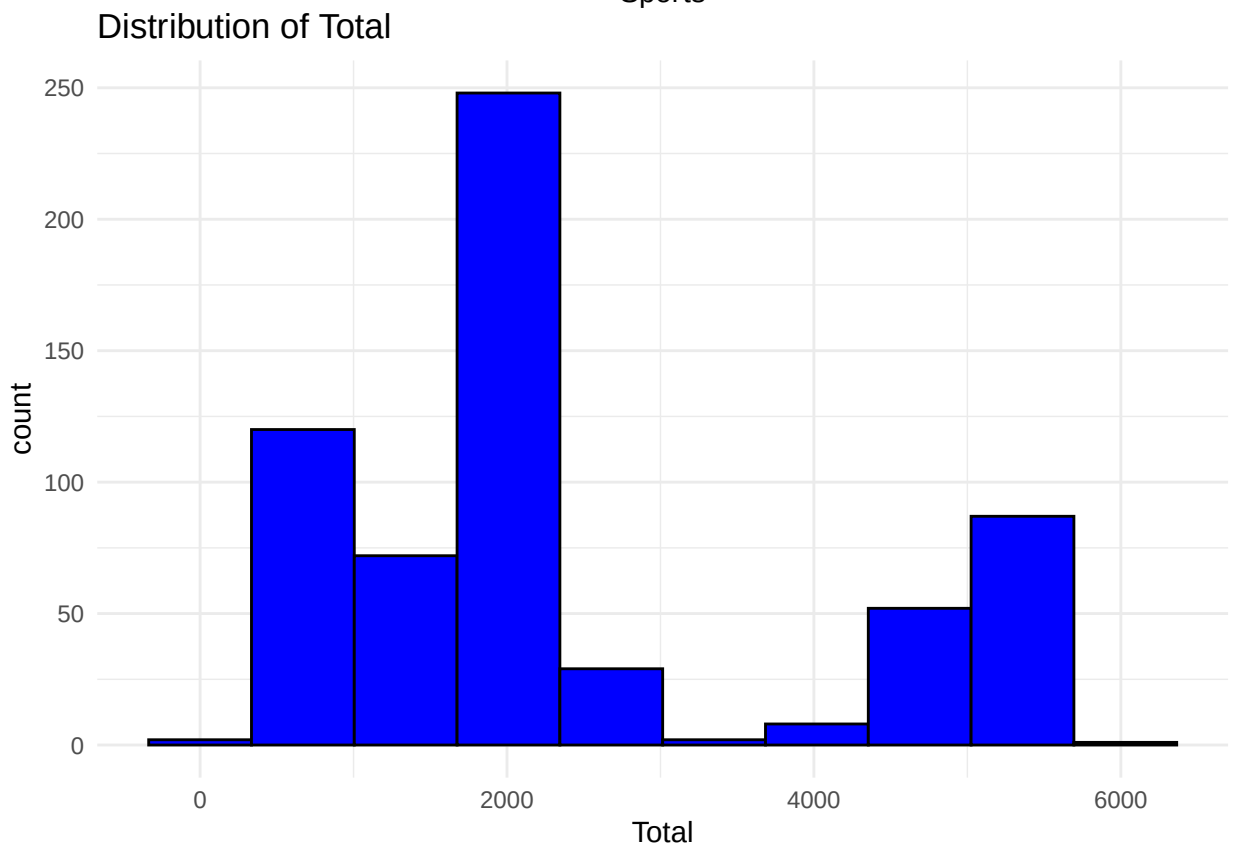

Distribution of ZeroCal
250
200
150
100
50
0
count
0
250
500
750
1000
ZeroCal


Distribution of Sugary
250
200
150
100
50
0
count
0
200
400
600
800
Sugary

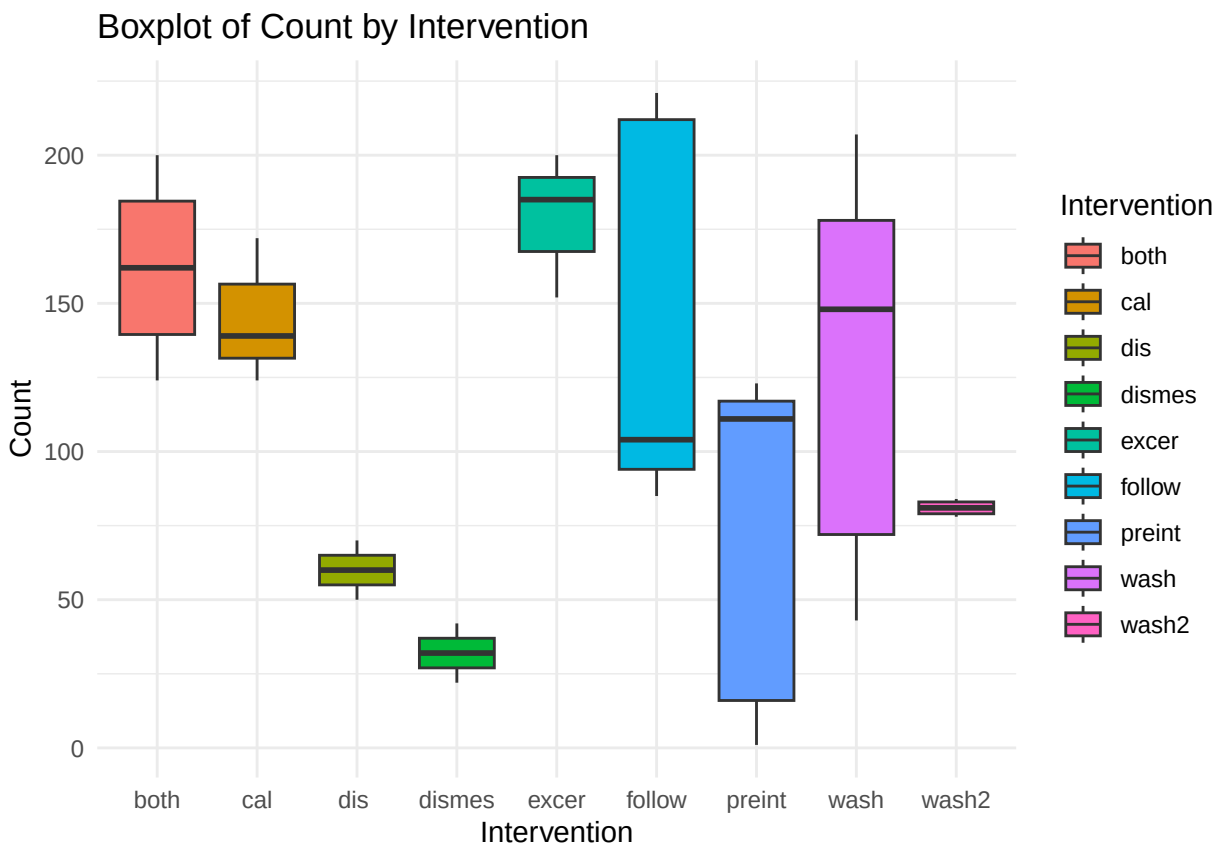

Distribution of Juice100
count
90
60
30
0
0
100
200
300
Juice100

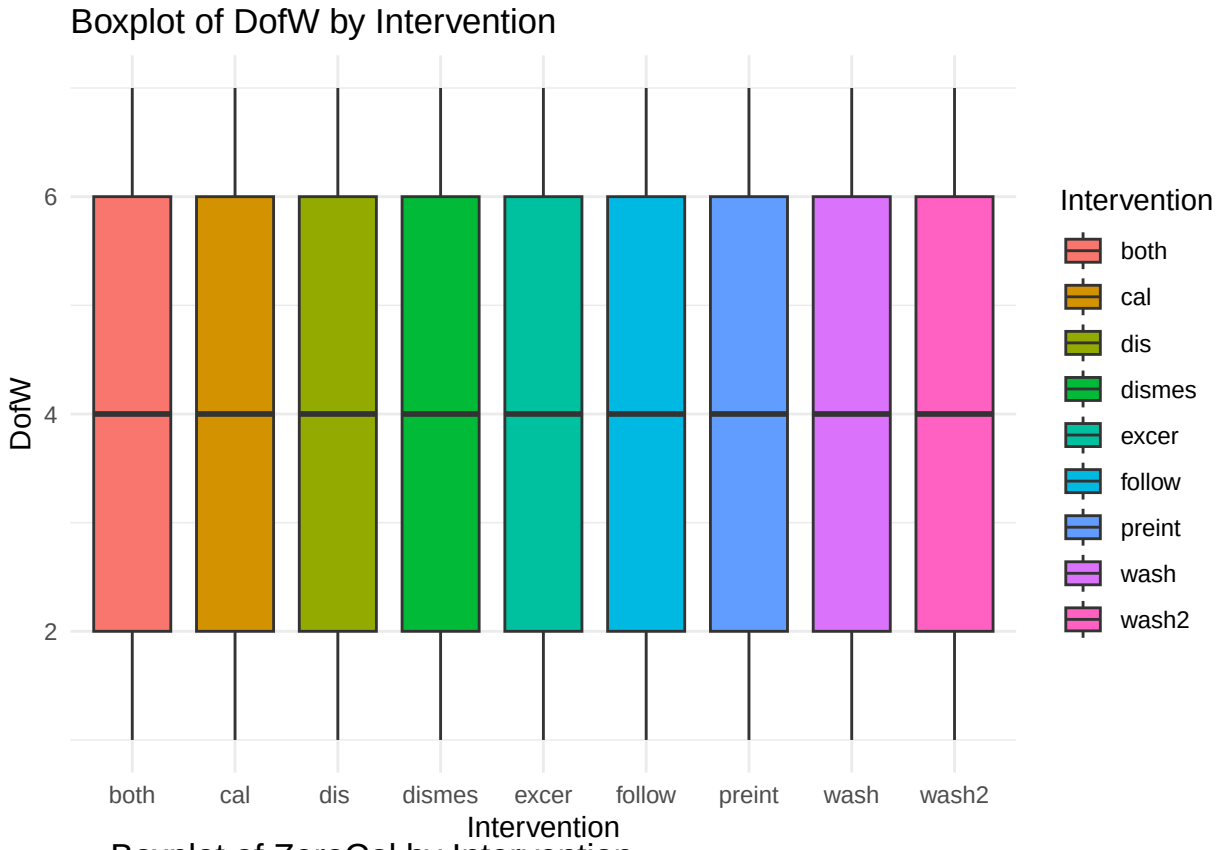

Distribution of Ojuice
count
50
40
30
20
10
0
40
80
120
160
Ojuice

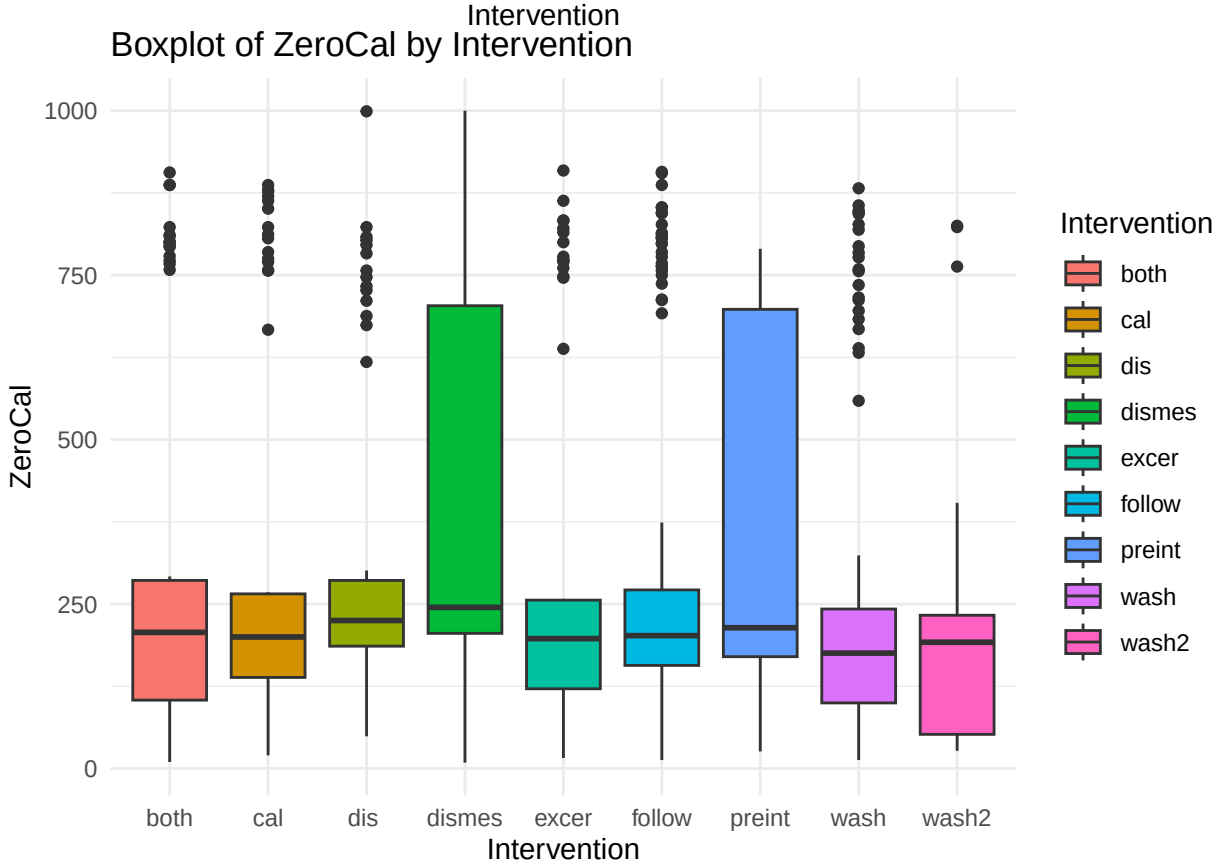

Distribution of Sports
count
150
100
50
0
0
50
100
150
Sports

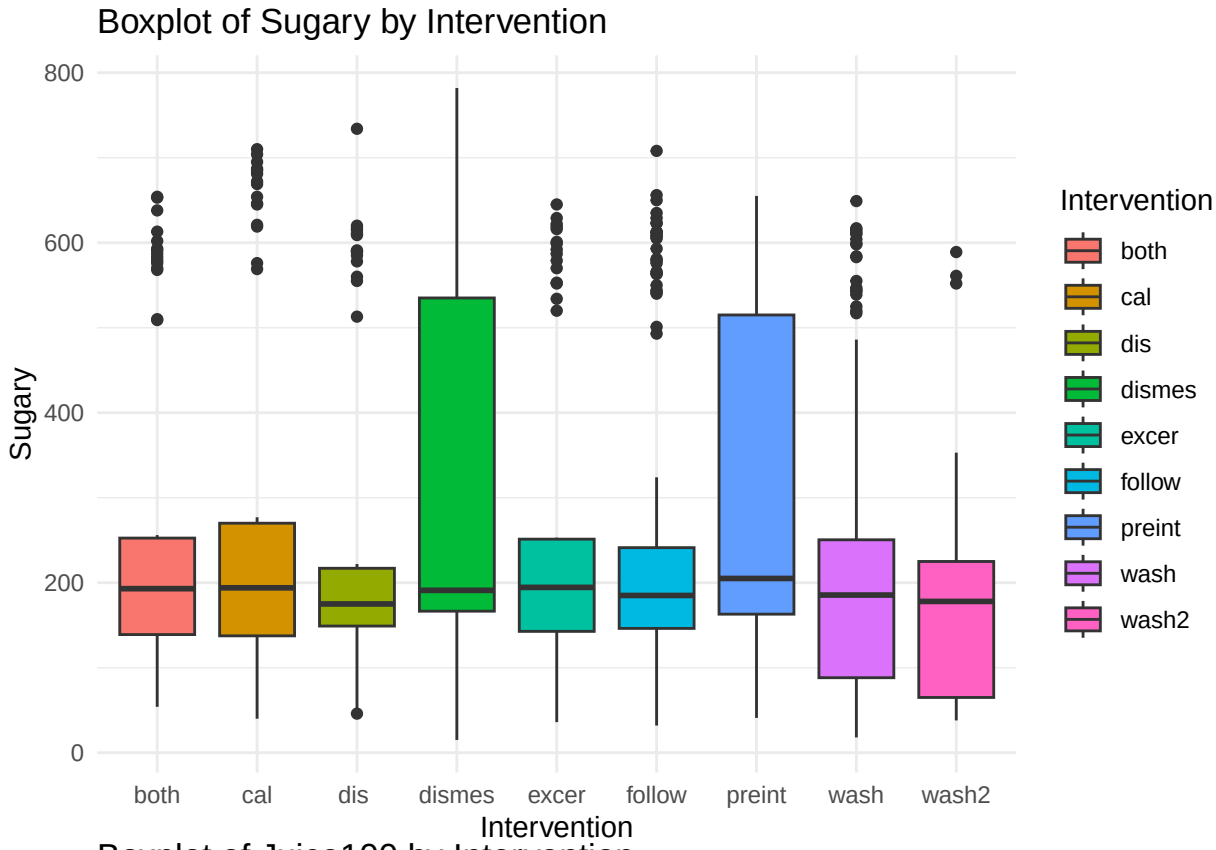

Distribution of Total
count
250
200
150
100
50
0
0
2000
4000
6000
Total

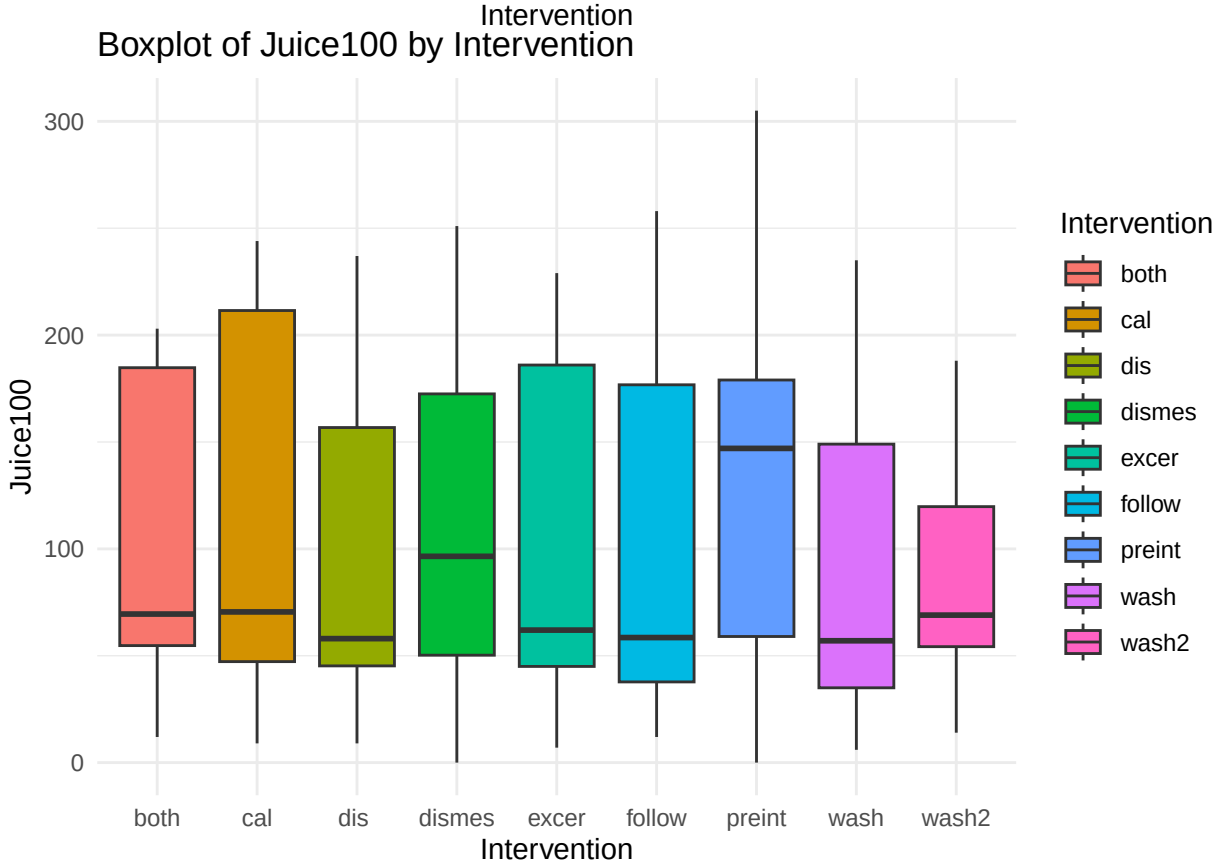
```r
#' Boxplots for Numeric Variables by Category
#'
#' Generates boxplots for numeric variables by a specified categorical variable.
#' @param df Data frame containing the data.
#' @param category Name of the categorical variable.
boxplots_by_category <- function(df, category) {
  numeric_vars <- select_if(df, is.numeric) %>% names()

  for (var in numeric_vars) {
    print(ggplot(df, aes_string(x = category, y = var, fill = category)) +
      geom_boxplot(na.rm = T) +
      theme_minimal() +
      labs(title = paste("Boxplot of", var, "by", category)))
  }
}
boxplots_by_category(beverage_sales, "Intervention")
```

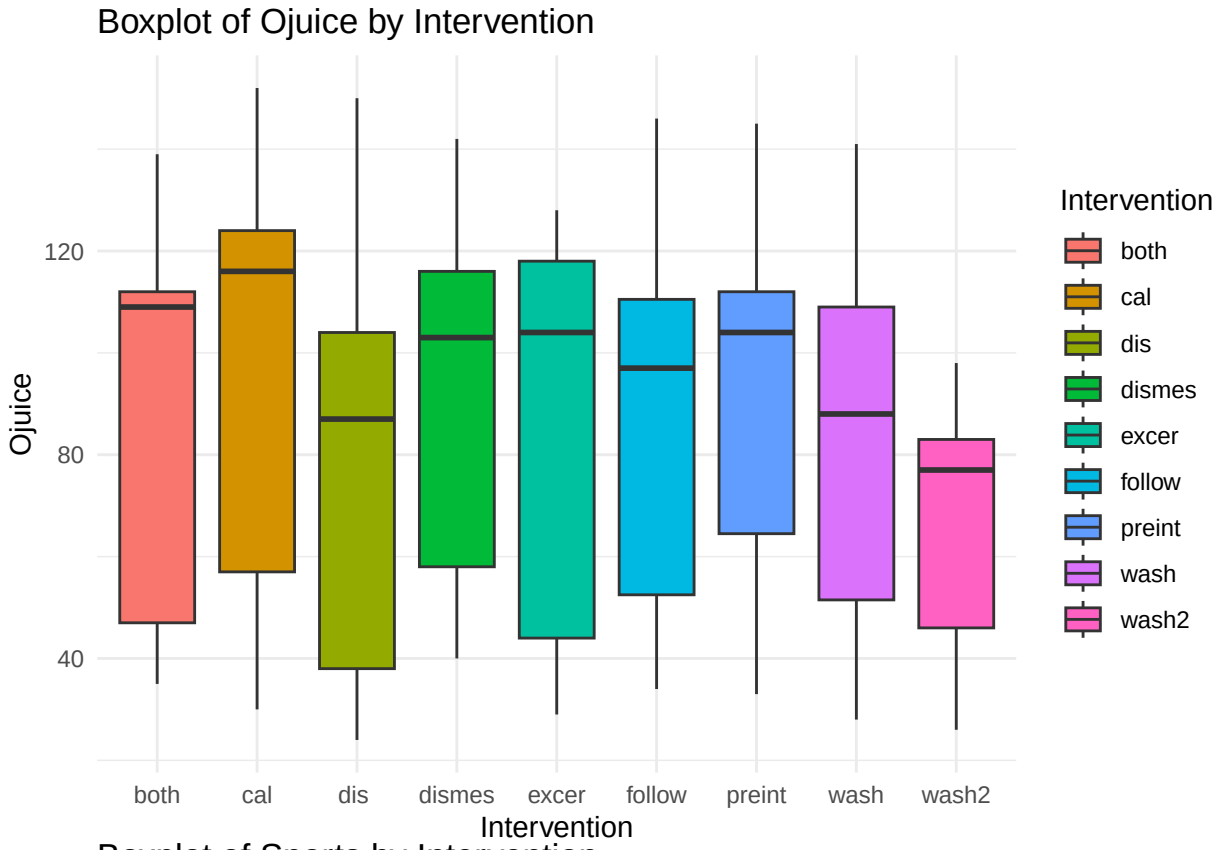


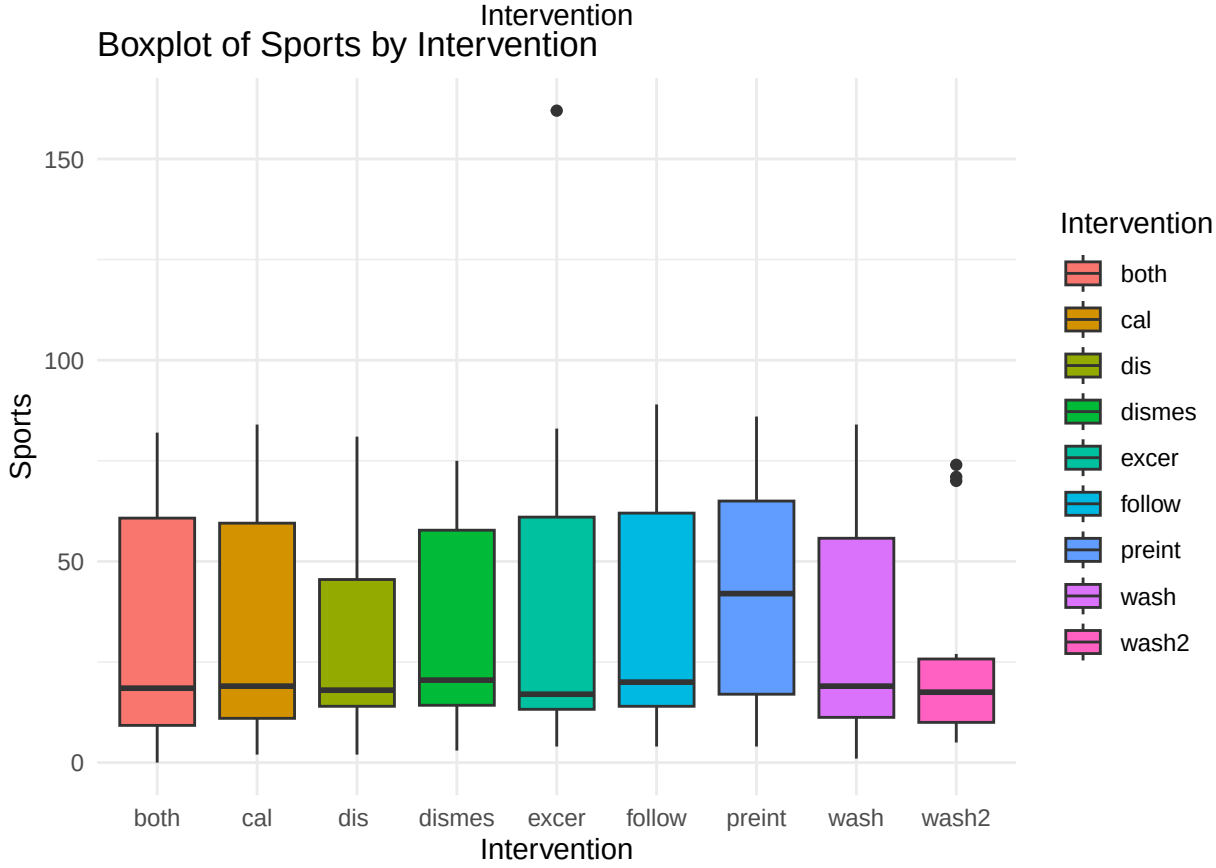

Boxplot of DofW by Intervention
DofW
6
4
2
both
cal
dis
dismes
excer
follow
preint
wash
wash2
Intervention
Intervention
both
cal
dis
dismes
excer
follow
preint
wash
wash2

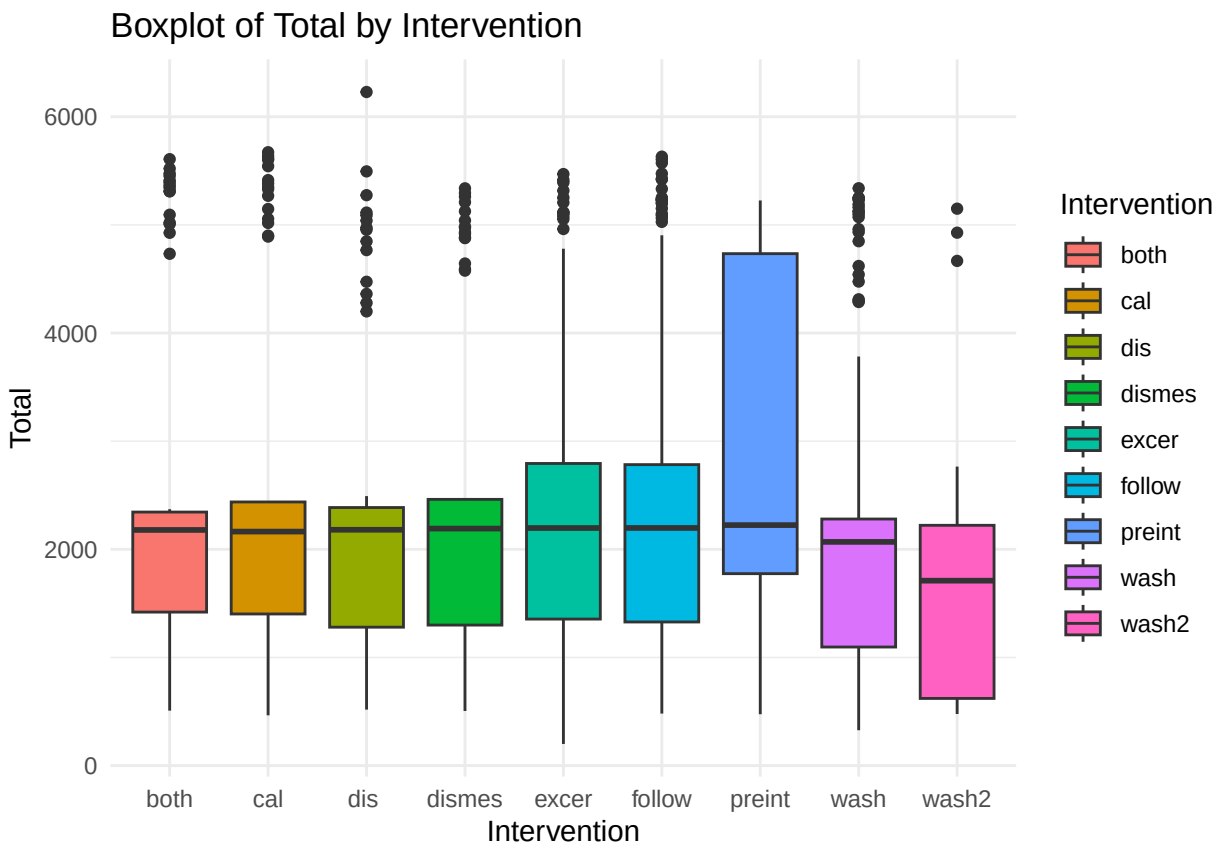

Boxplot of ZeroCal by Intervention
ZeroCal
1000
750
500
250
0
both
cal
dis
dismes
excer
follow
preint
wash
wash2
Intervention
Intervention
both
cal
dis
dismes
excer
follow
preint
wash
wash2


Boxplot of Sugary by Intervention
Sugary
800
600
400
200
0
both
cal
dis
dismes
excer
follow
preint
wash
wash2
Intervention
Intervention
both
cal
dis
dismes
excer
follow
preint
wash
wash2

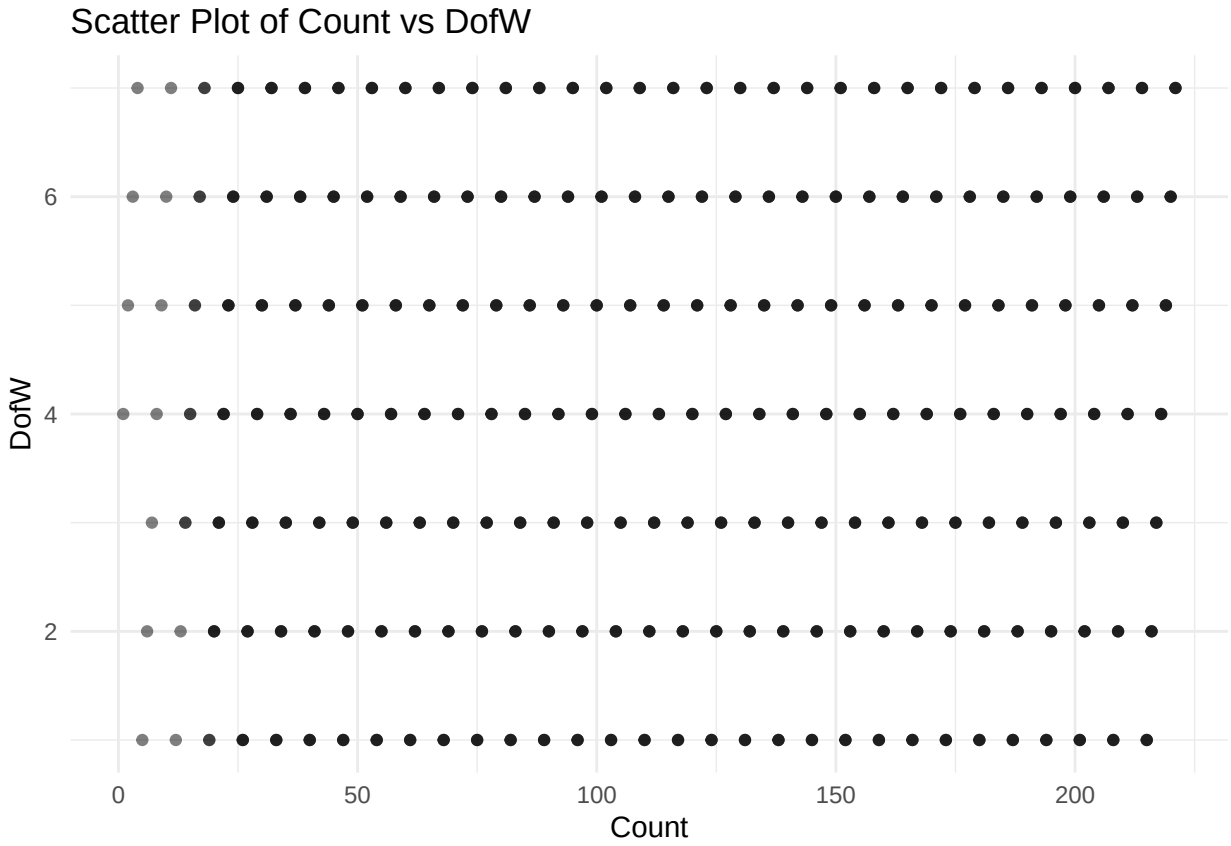

Boxplot of Juice100 by Intervention
Juice100
300
200
100
0
both
cal
dis
dismes
excer
follow
preint
wash
wash2
Intervention
Intervention
both
cal
dis
dismes
excer
follow
preint
wash
wash2


Boxplot of Ojuice by Intervention
Ojuice
120
80
40
both
cal
dis
dismes
excer
follow
preint
wash
wash2
Intervention
Intervention
both
cal
dis
dismes
excer
follow
preint
wash
wash2

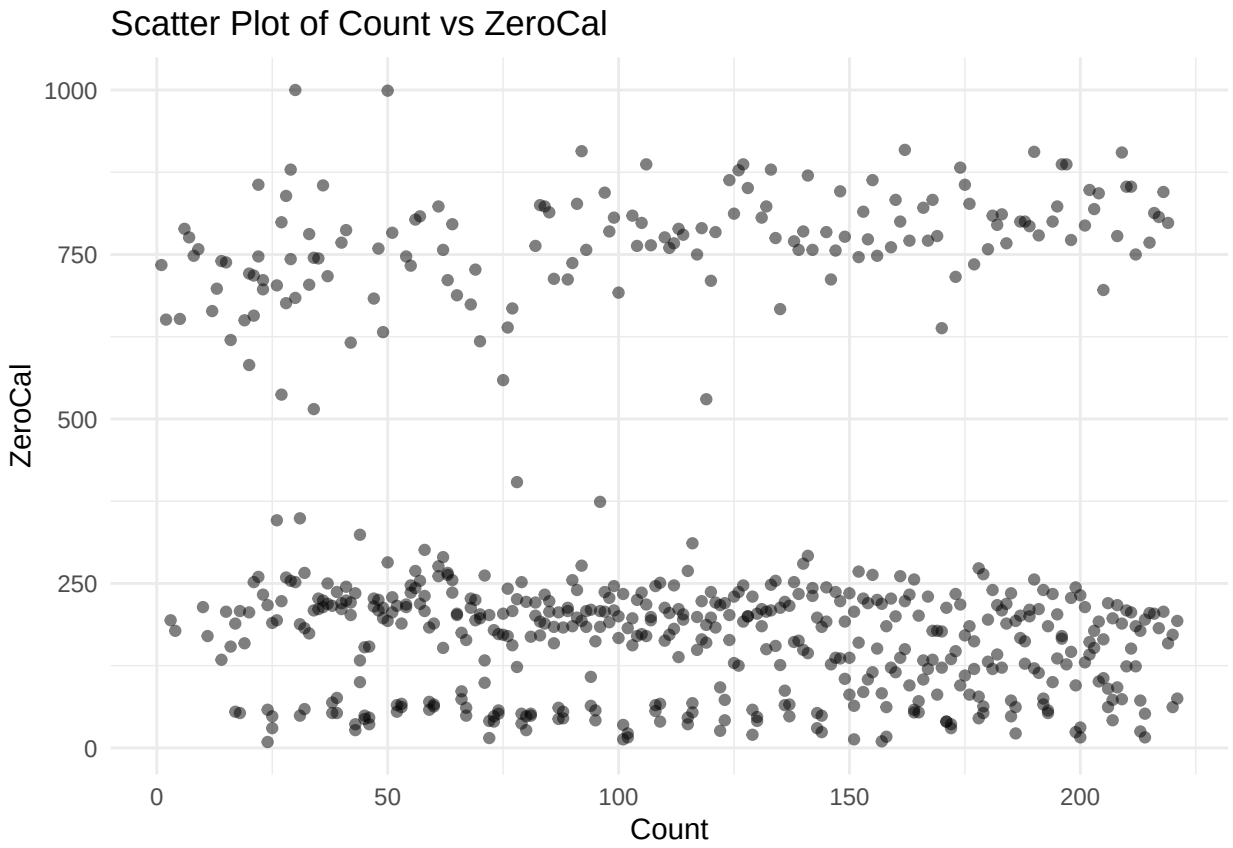

Boxplot of Sports by Intervention
Sports
150
100
50
0
both
cal
dis
dismes
excer
follow
preint
wash
wash2
Intervention
Intervention
both
cal
dis
dismes
excer
follow
preint
wash
wash2



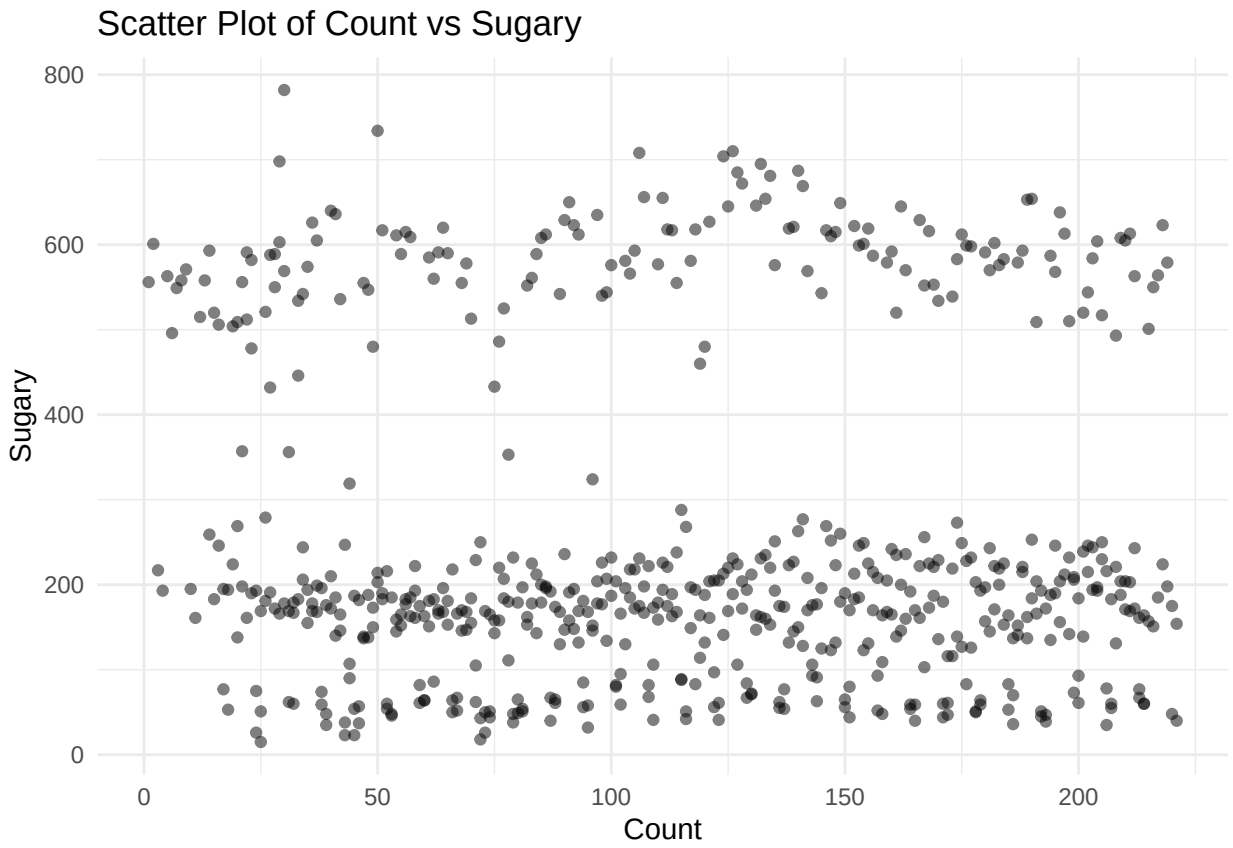

```r
#' Scatter Plots for Numeric Variables
#'
#' Generates scatter plots for pairs of numeric variables.
#' @param df Data frame containing the data.
scatter_plots <- function(df) {
  numeric_vars <- select_if(df, is.numeric) %>% names()

  if (length(numeric_vars) > 1) {
    for (i in 1:(length(numeric_vars) - 1)) {
      for (j in (i + 1):length(numeric_vars)) {
        print(ggplot(df, aes_string(x = numeric_vars[i], y = numeric_vars[j])) +
          geom_point(alpha = 0.5) +
          theme_minimal() +
          labs(title = paste("Scatter Plot of", numeric_vars[i], "vs", numeric_vars[j])))
      }
    }
  }
}
scatter_plots(beverage_sales)
```



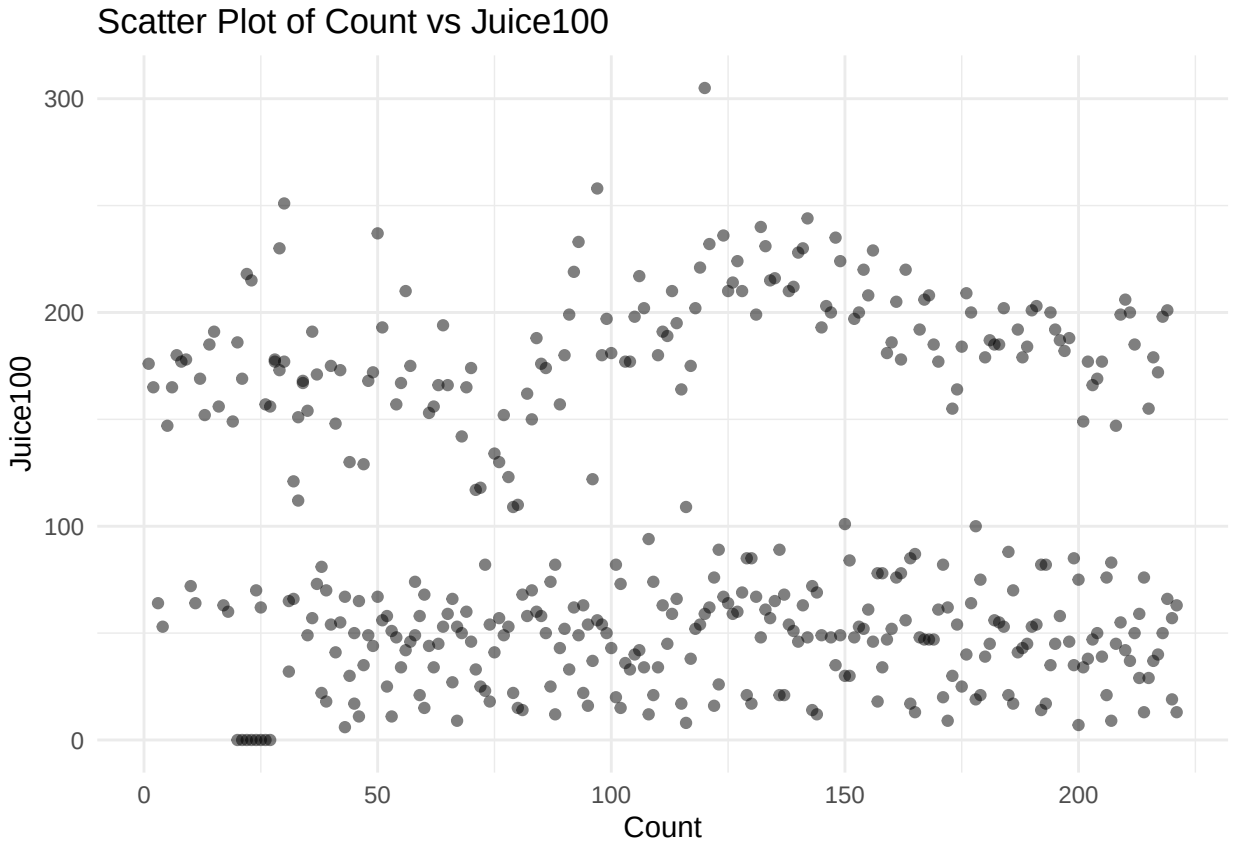

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```



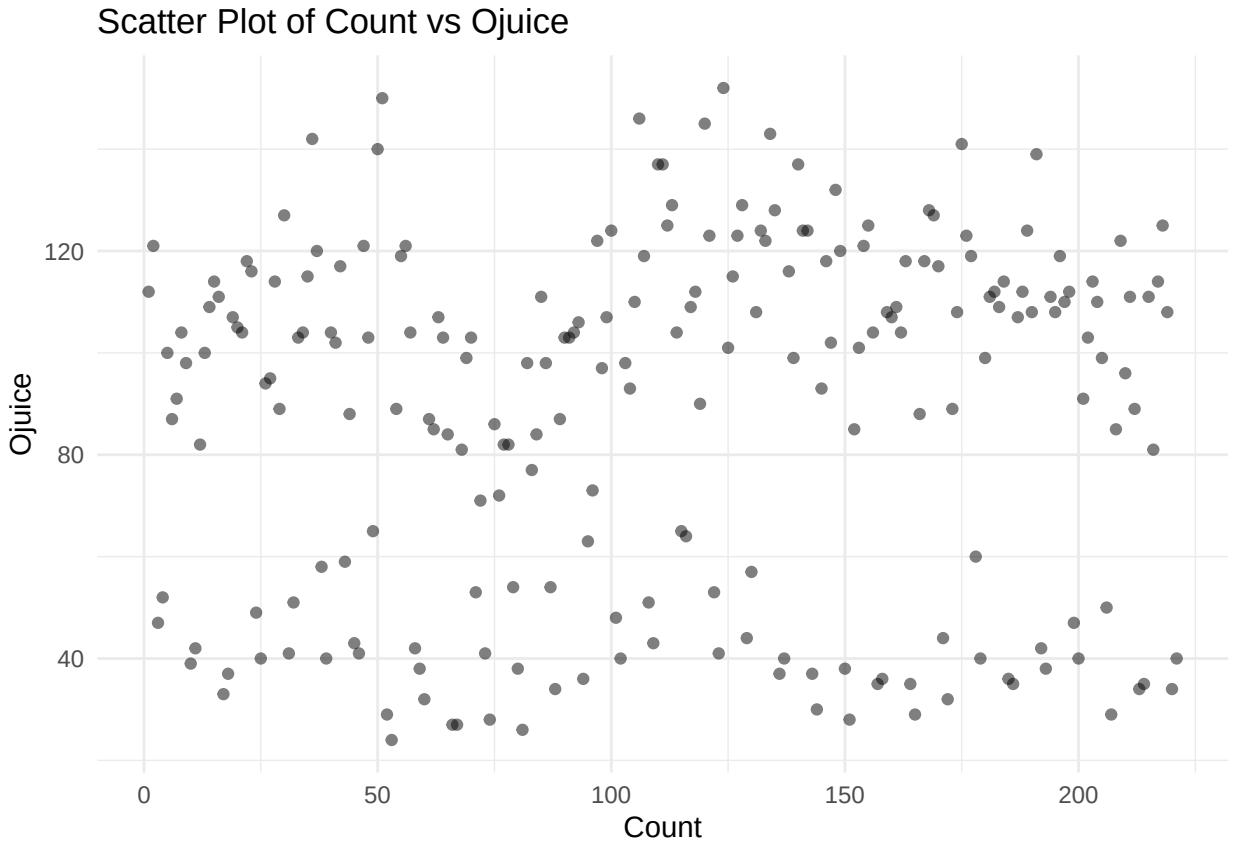

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```



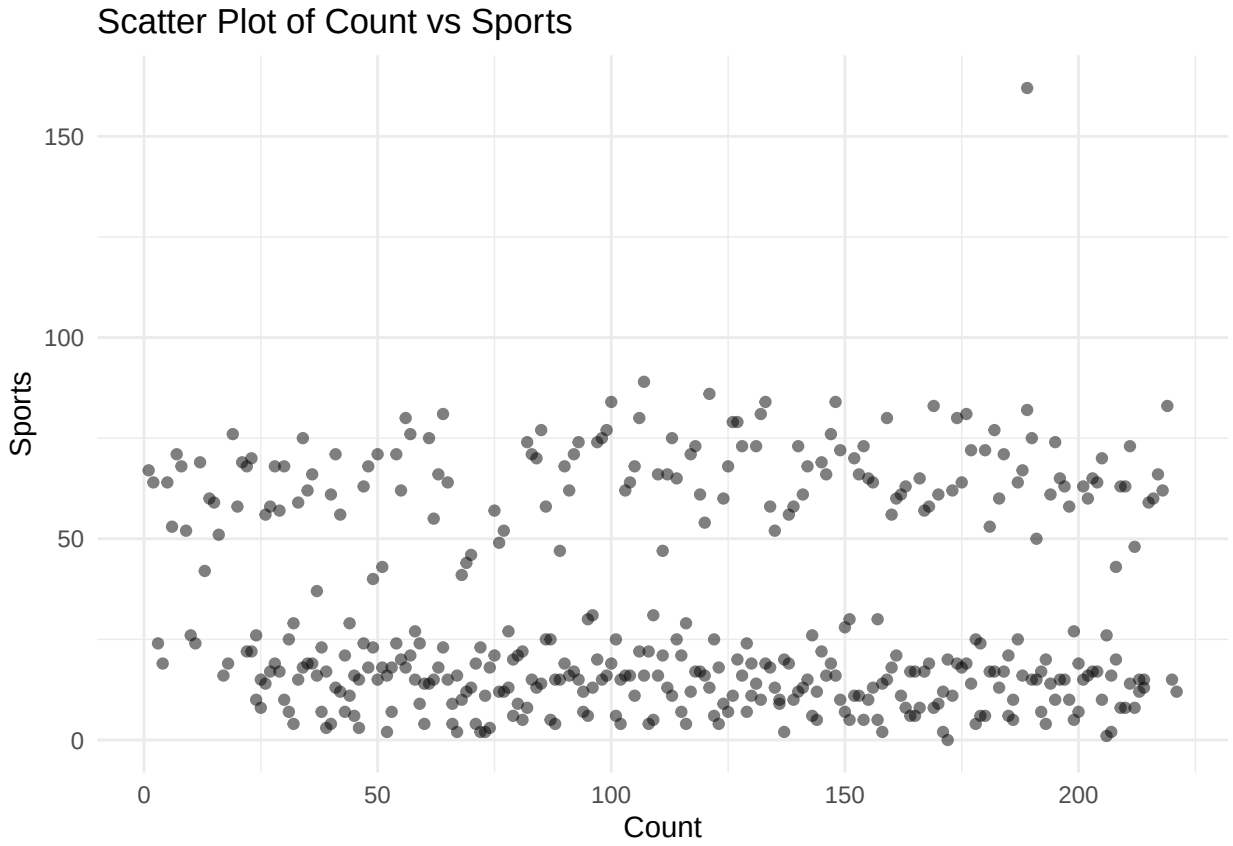

```
## Warning: Removed 210 rows containing missing values (‘geom_point()‘).
```




```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```

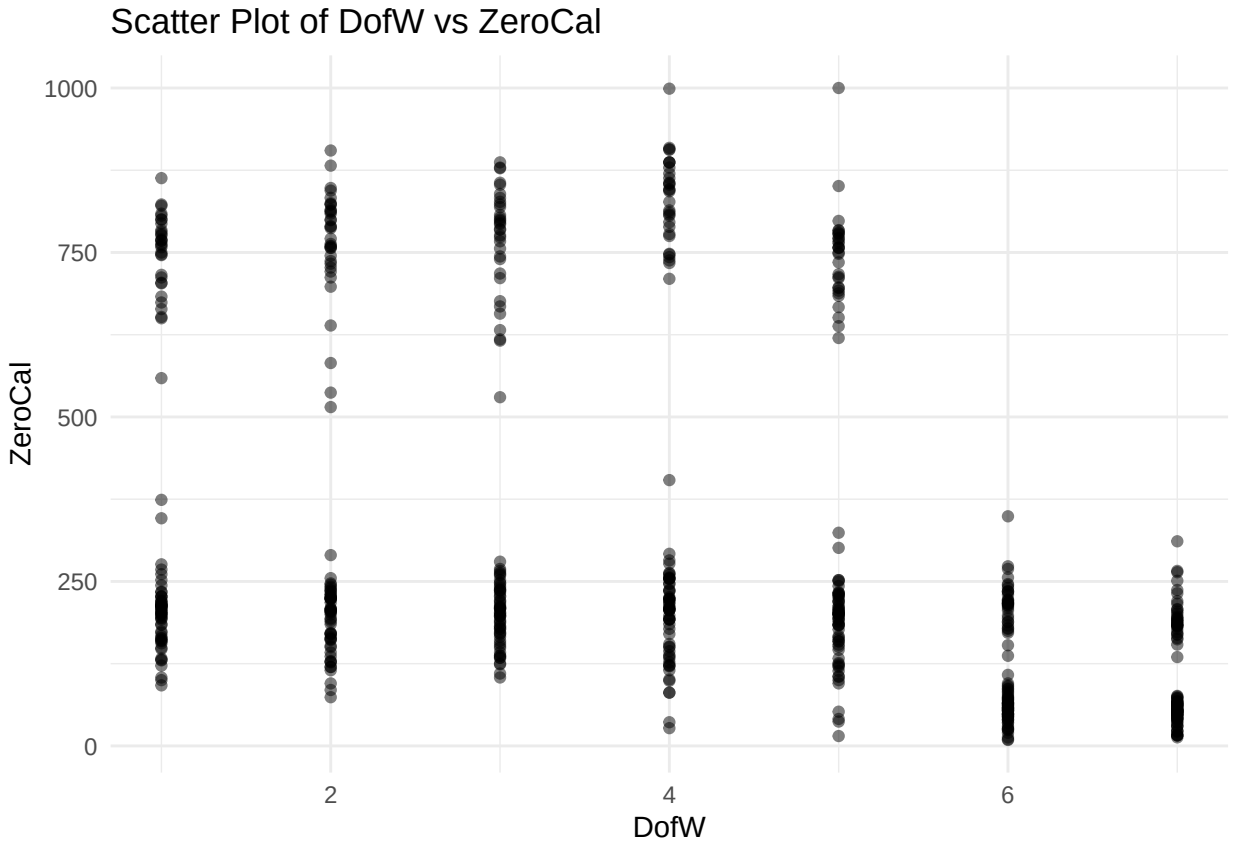



```
## Warning: Removed 217 rows containing missing values (‘geom_point()‘).
```

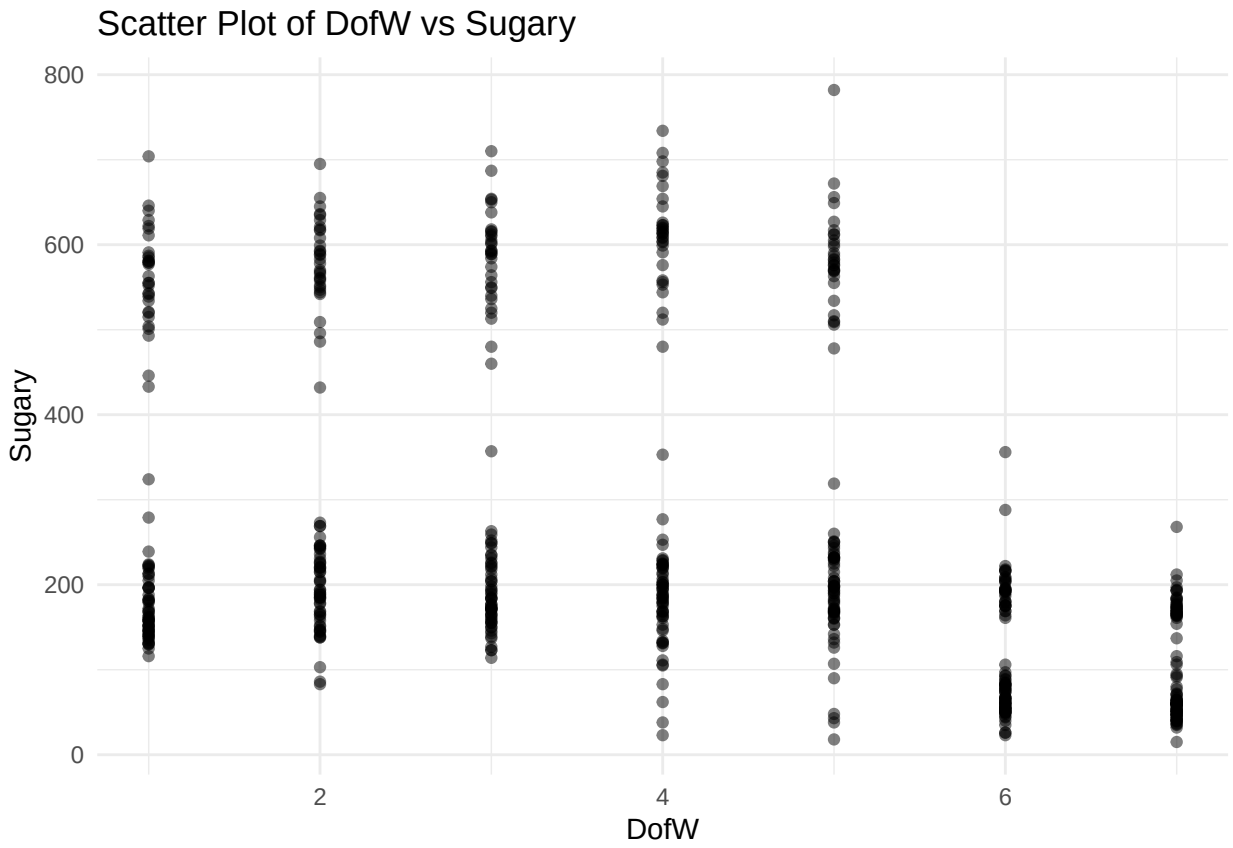



```
## Warning: Removed 10 rows containing missing values (‘geom_point()‘).
```

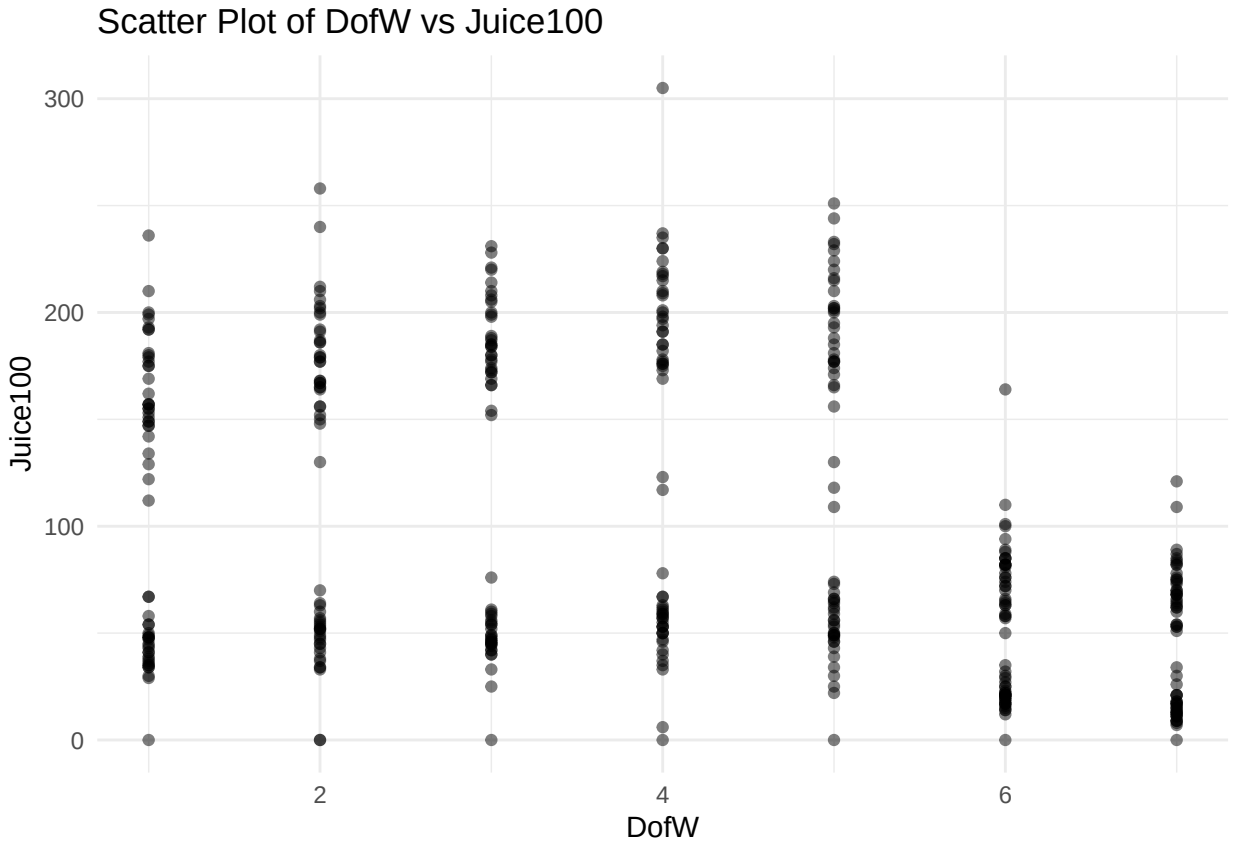

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```



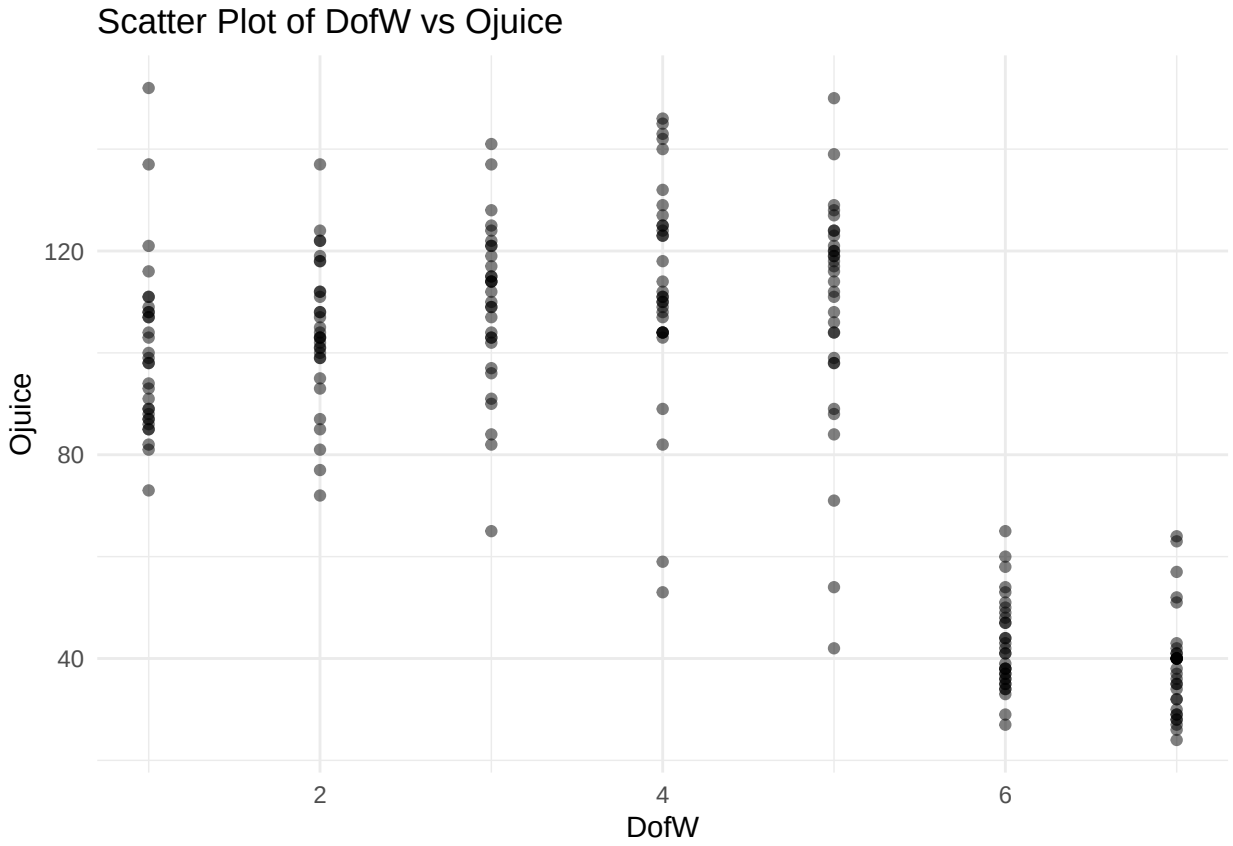

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```



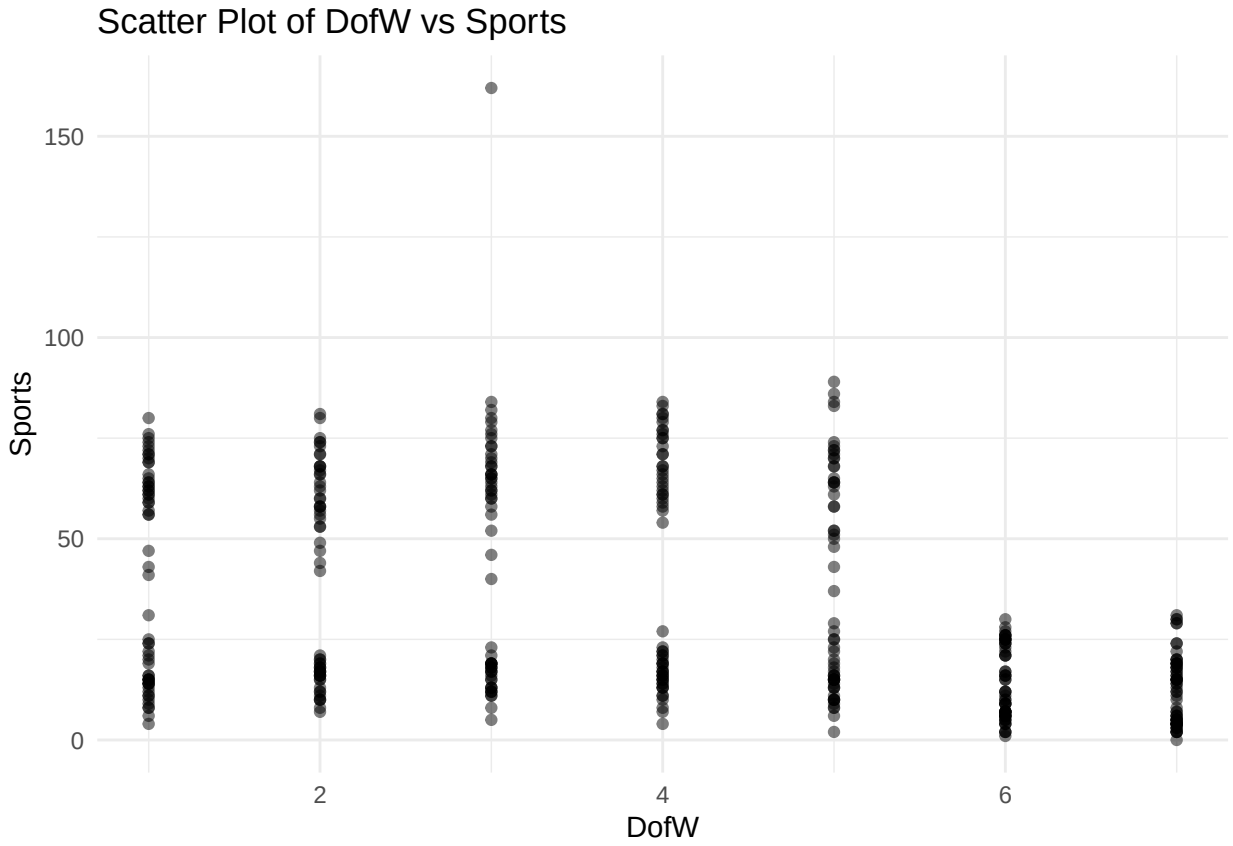

```
## Warning: Removed 210 rows containing missing values (‘geom_point()‘).
```



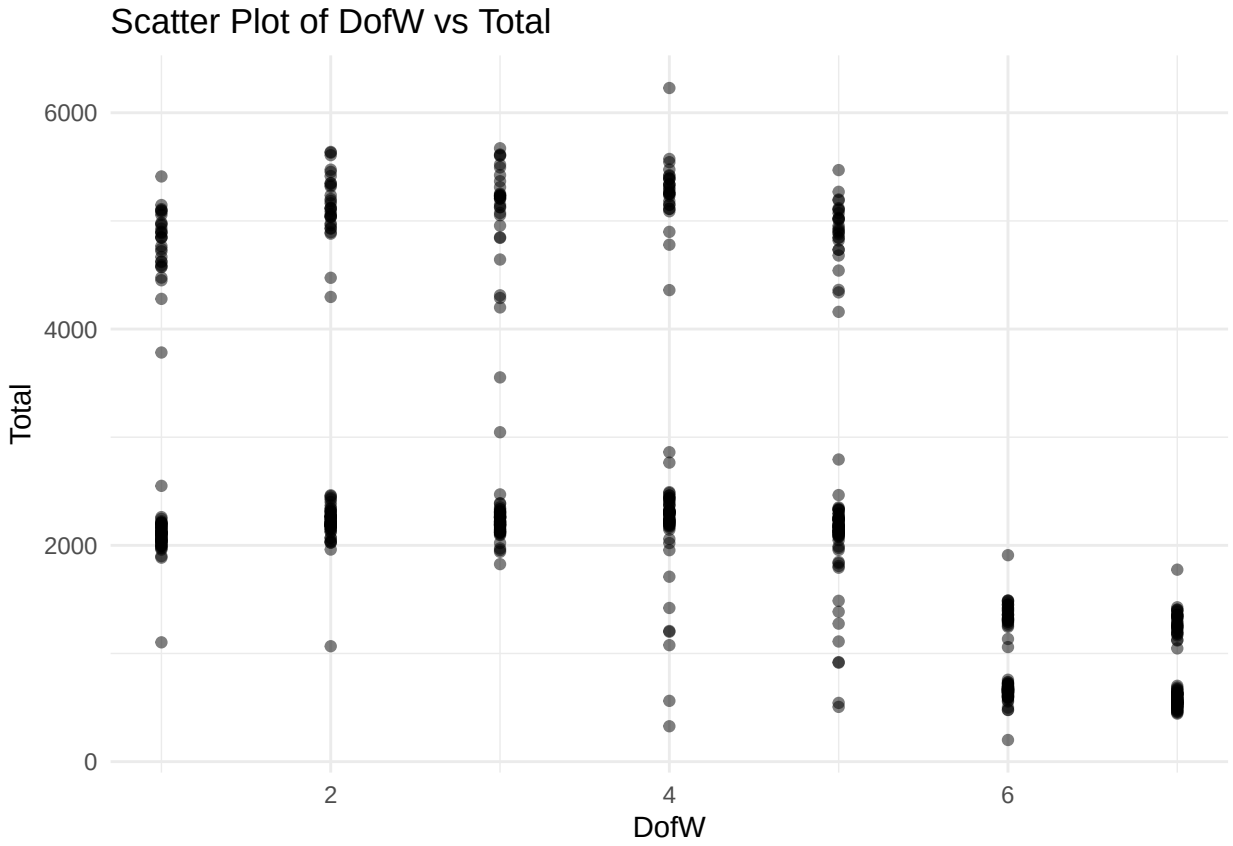

```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```



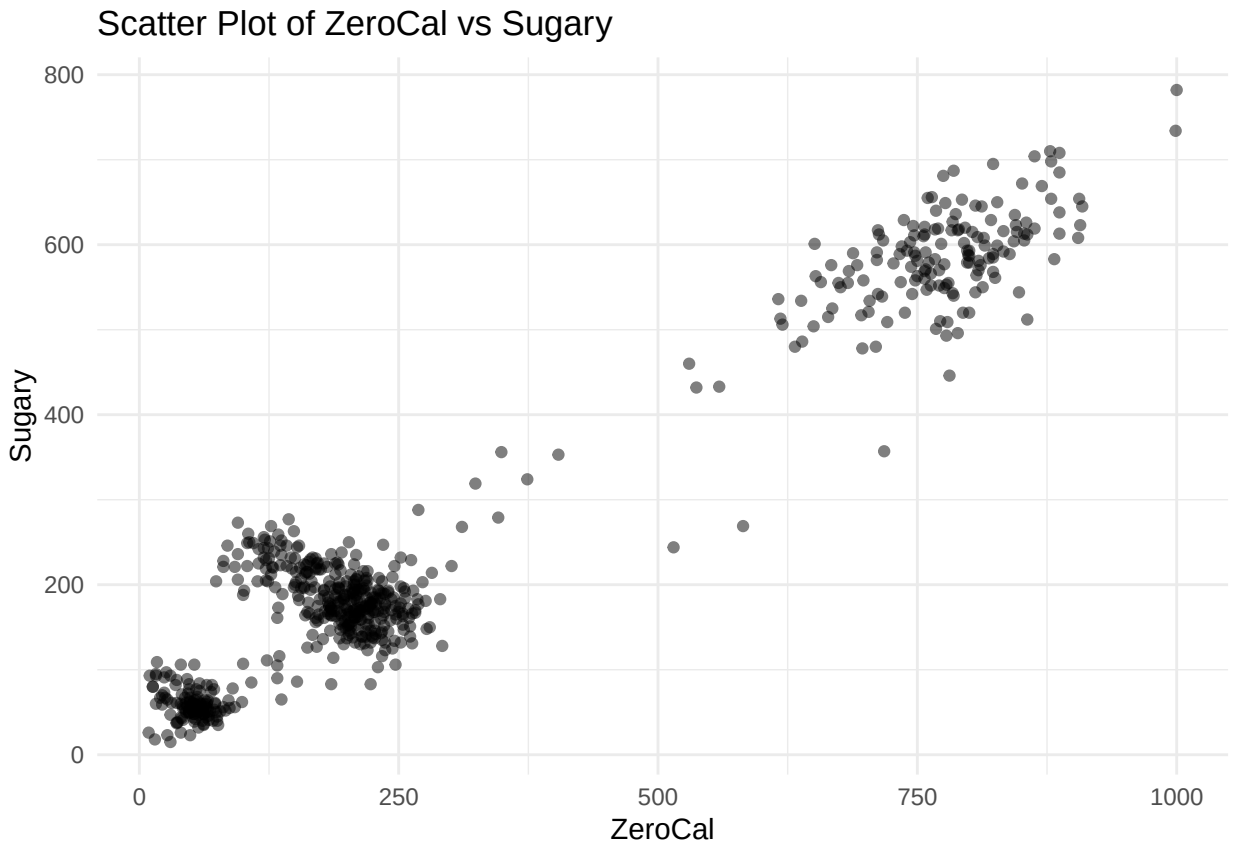

```
## Warning: Removed 217 rows containing missing values (‘geom_point()‘).
```




```
## Warning: Removed 10 rows containing missing values (‘geom_point()‘).
```

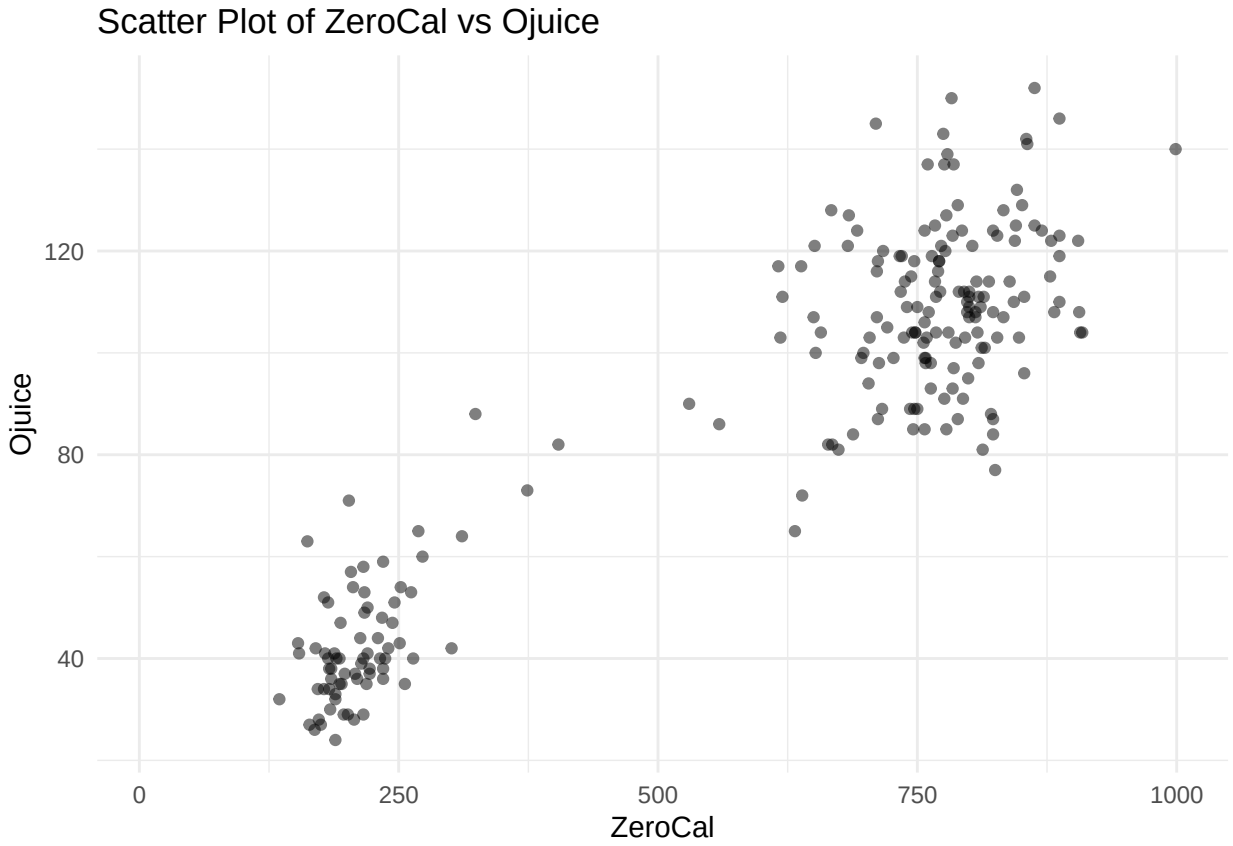



```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```

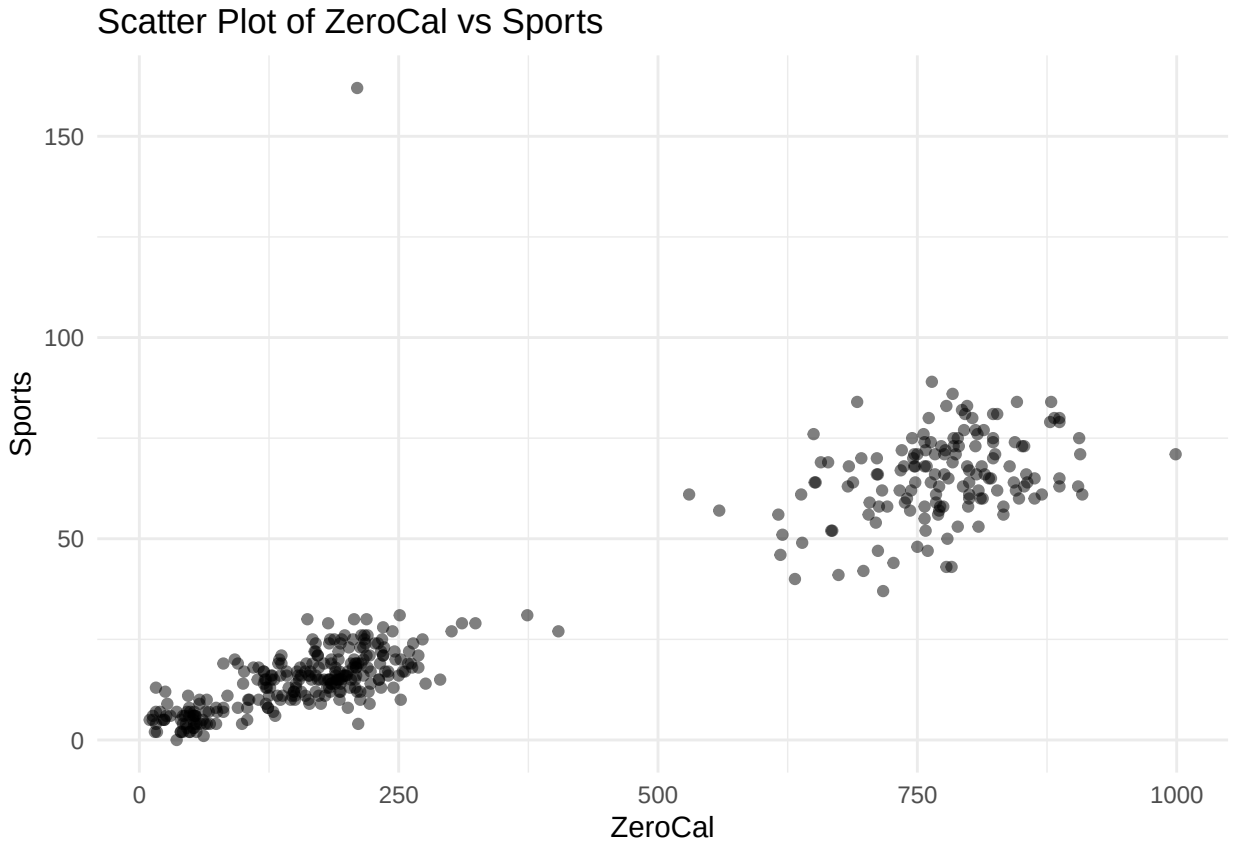



```
## Warning: Removed 210 rows containing missing values (‘geom_point()‘).
```

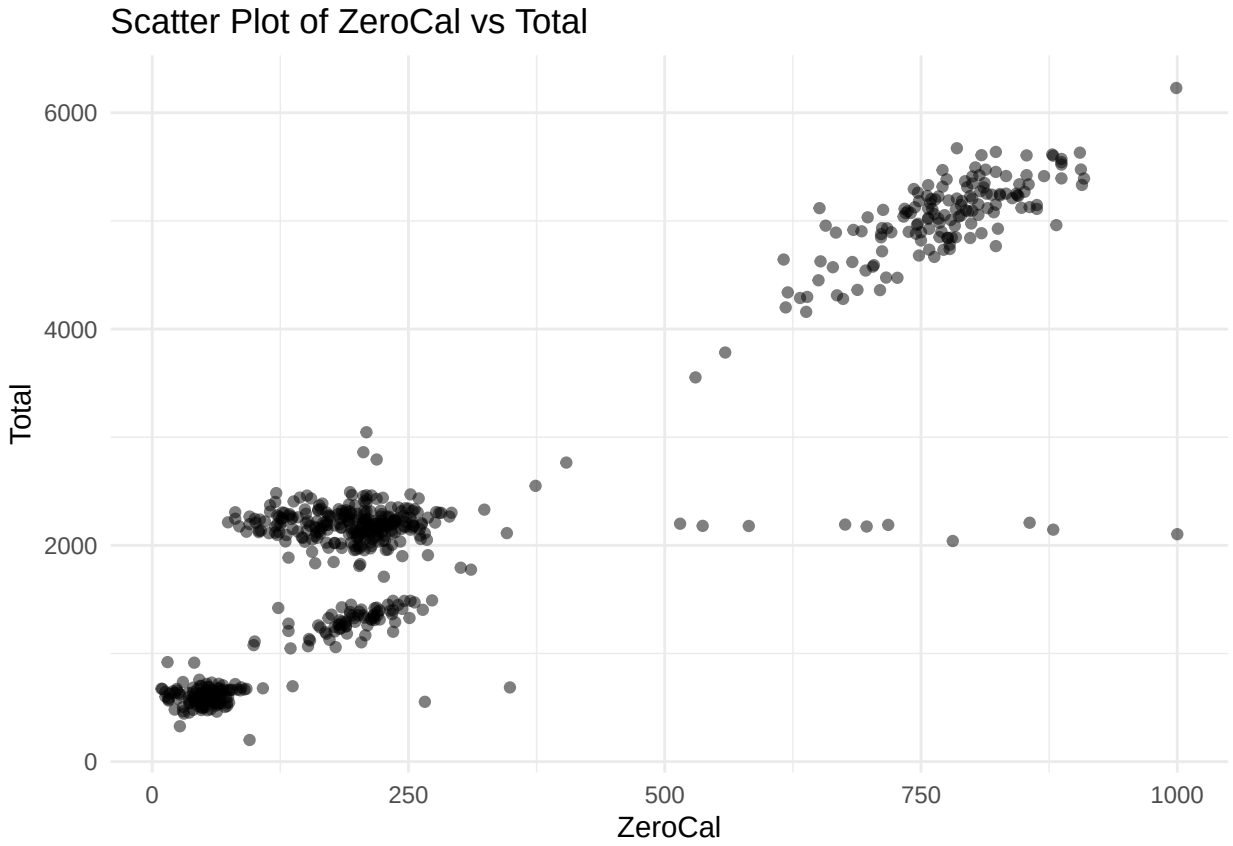

```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```



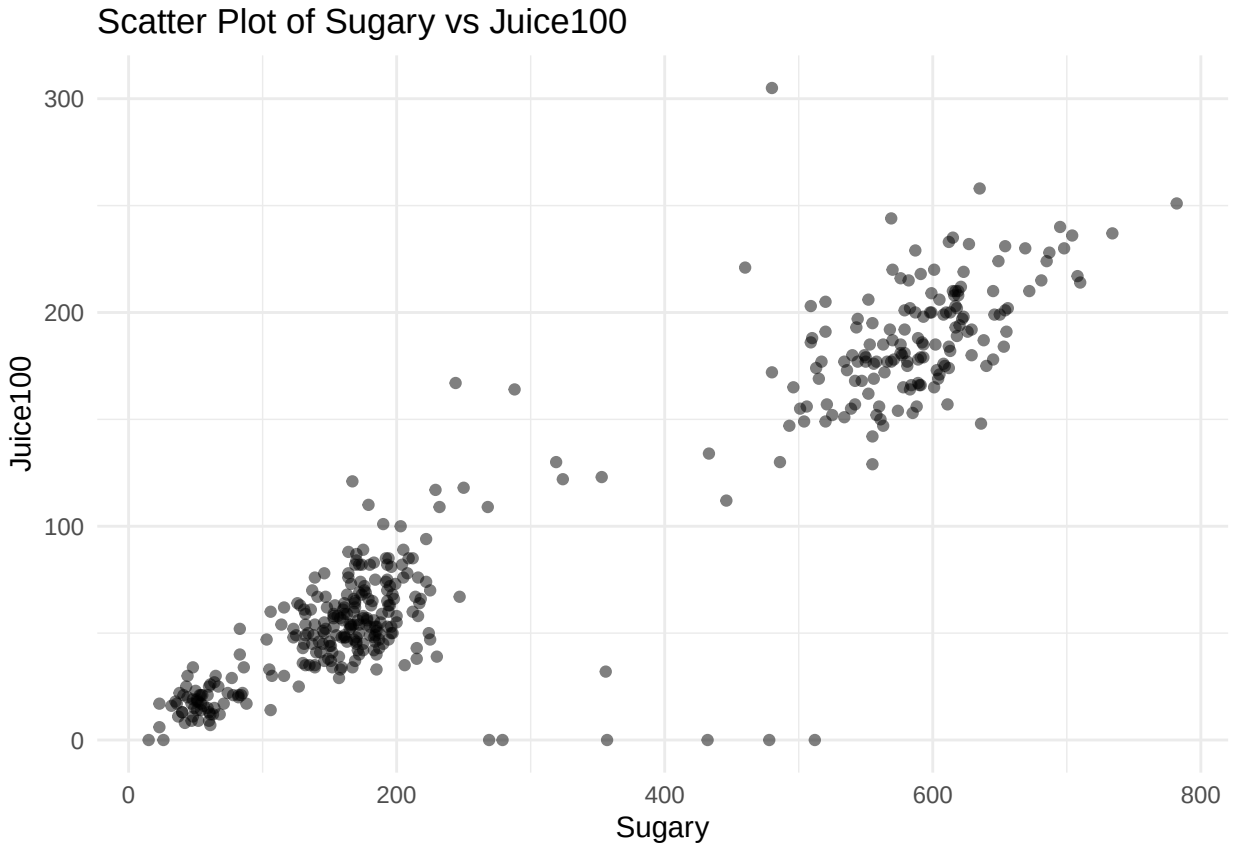

```
## Warning: Removed 217 rows containing missing values (‘geom_point()‘).
```



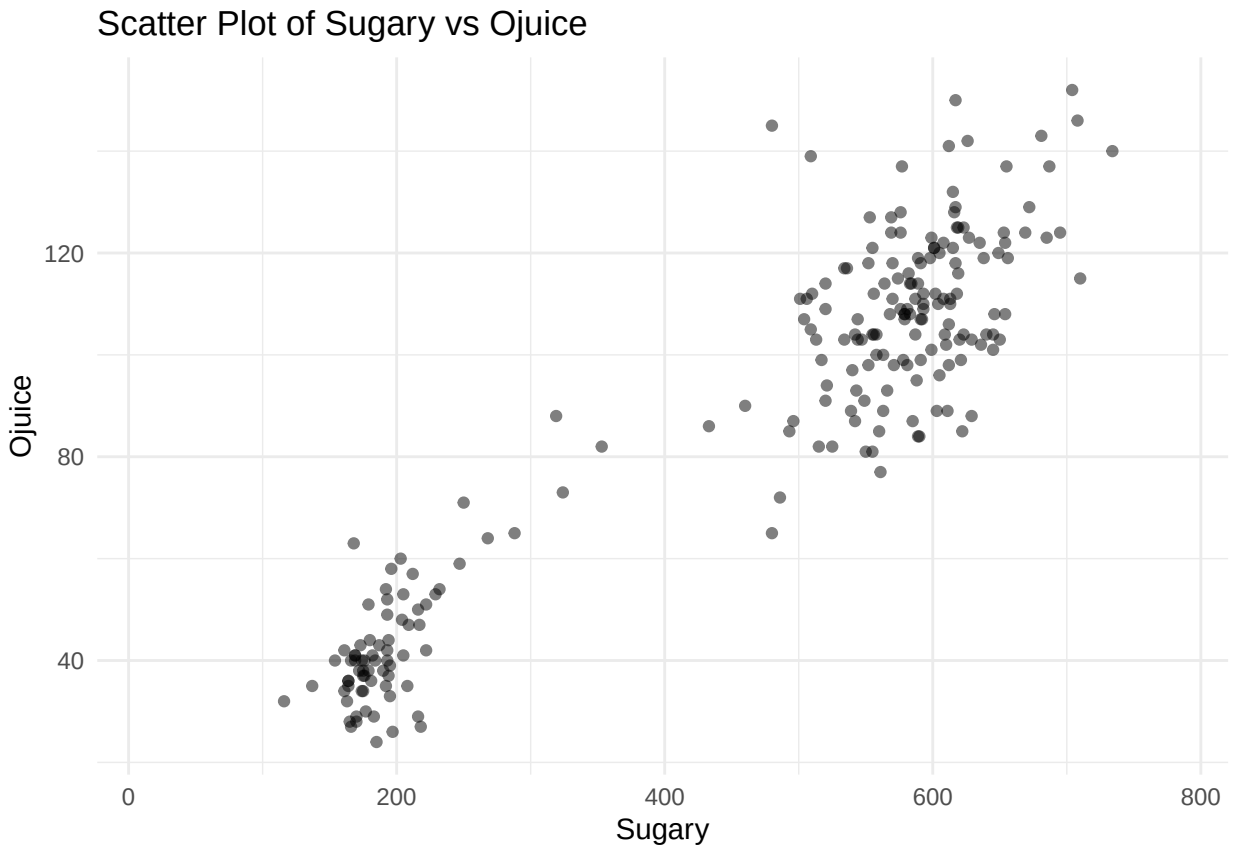

```
## Warning: Removed 10 rows containing missing values (‘geom_point()‘).
```



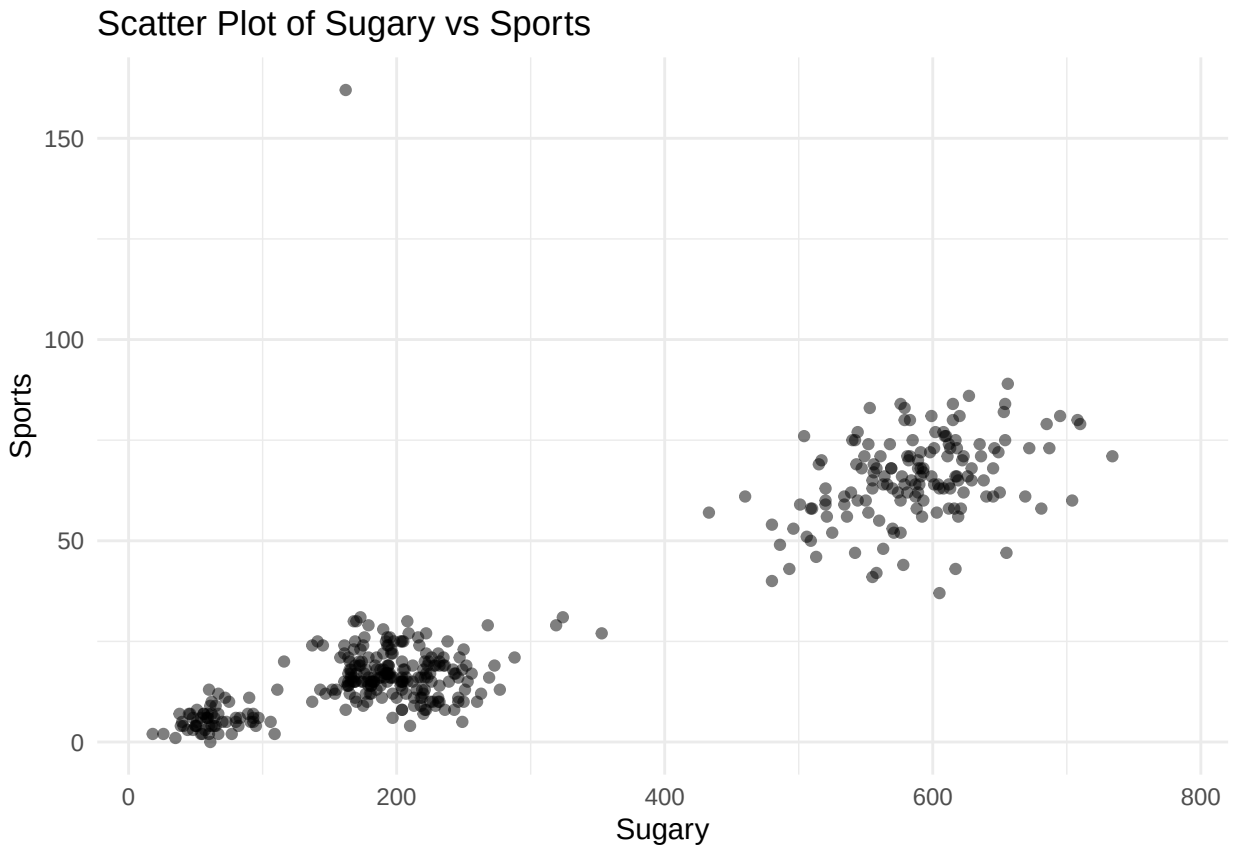

```
## Warning: Removed 210 rows containing missing values (‘geom_point()‘).
```



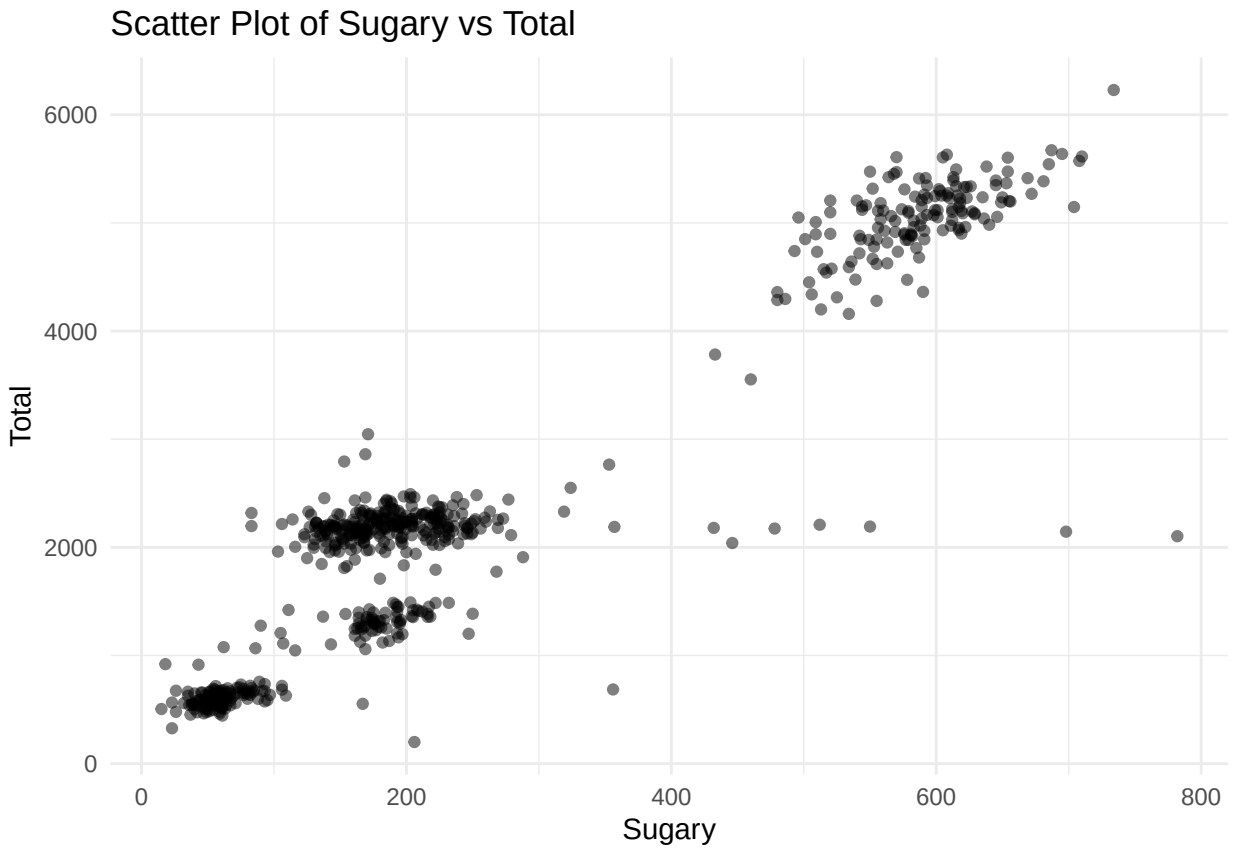

```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```



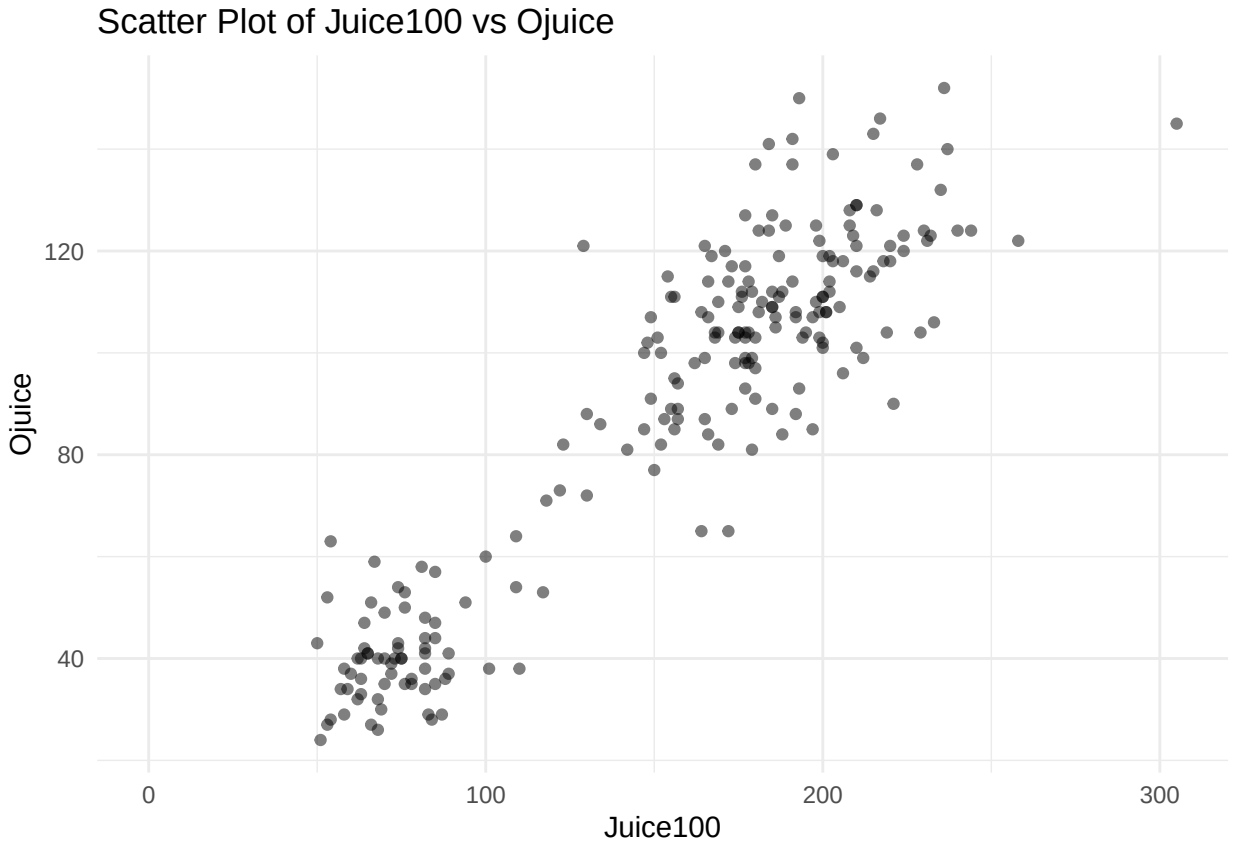

```
## Warning: Removed 217 rows containing missing values (‘geom_point()‘).
```



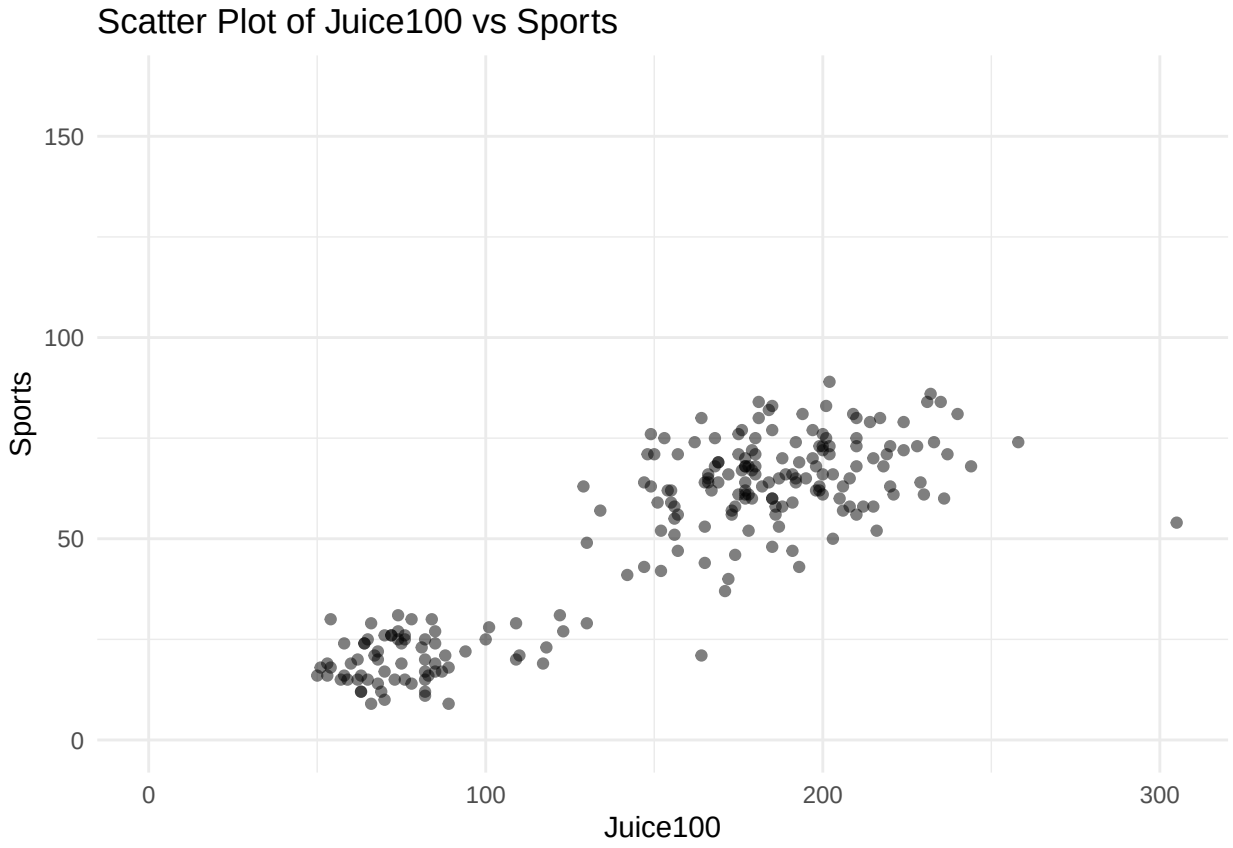

```
## Warning: Removed 10 rows containing missing values (‘geom_point()‘).
```



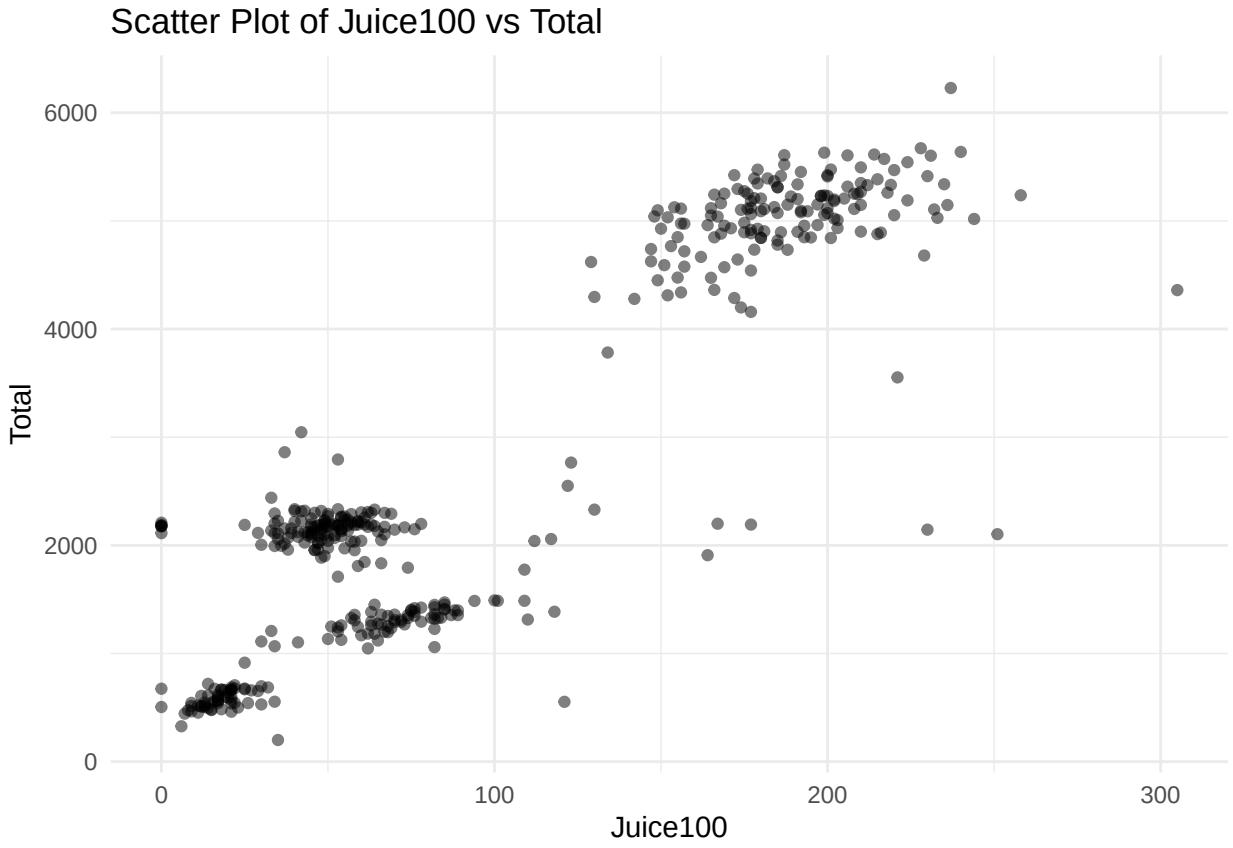

```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```



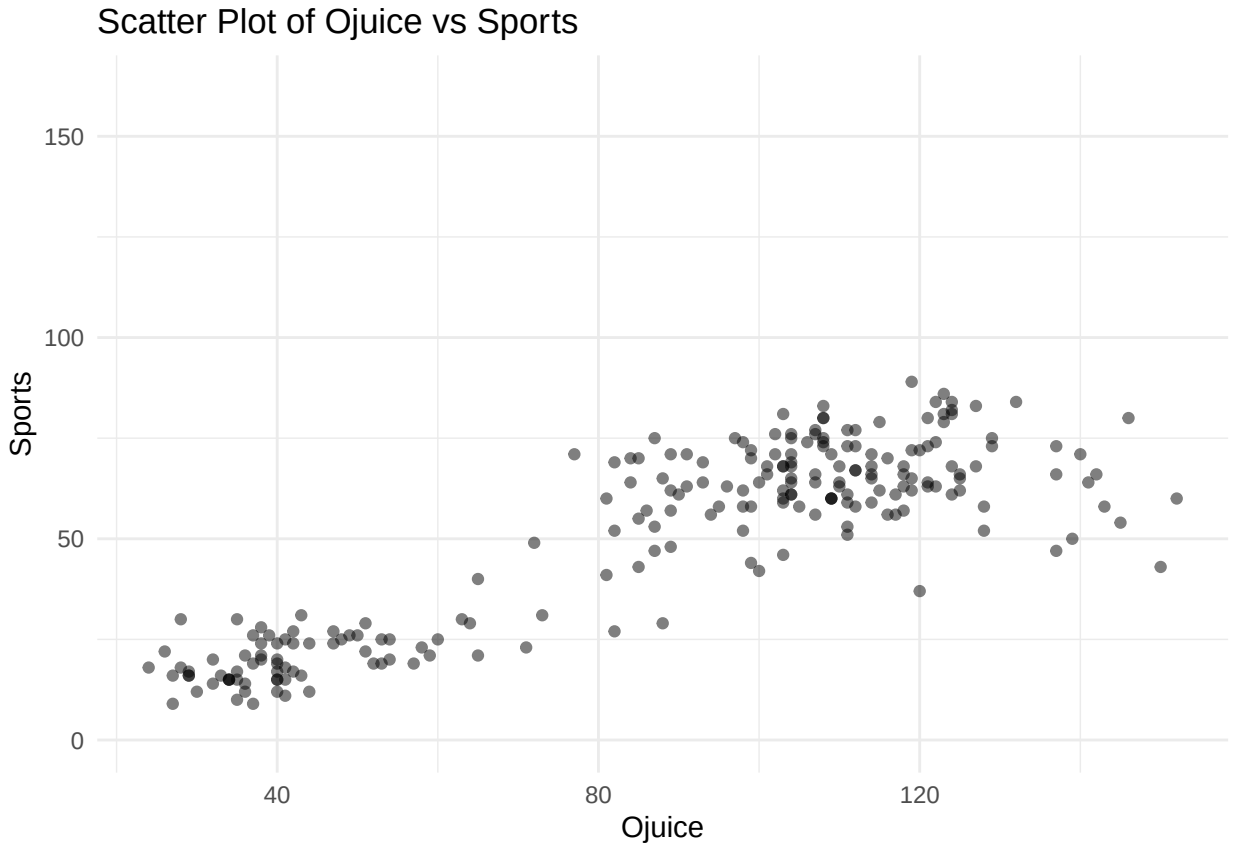

```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```



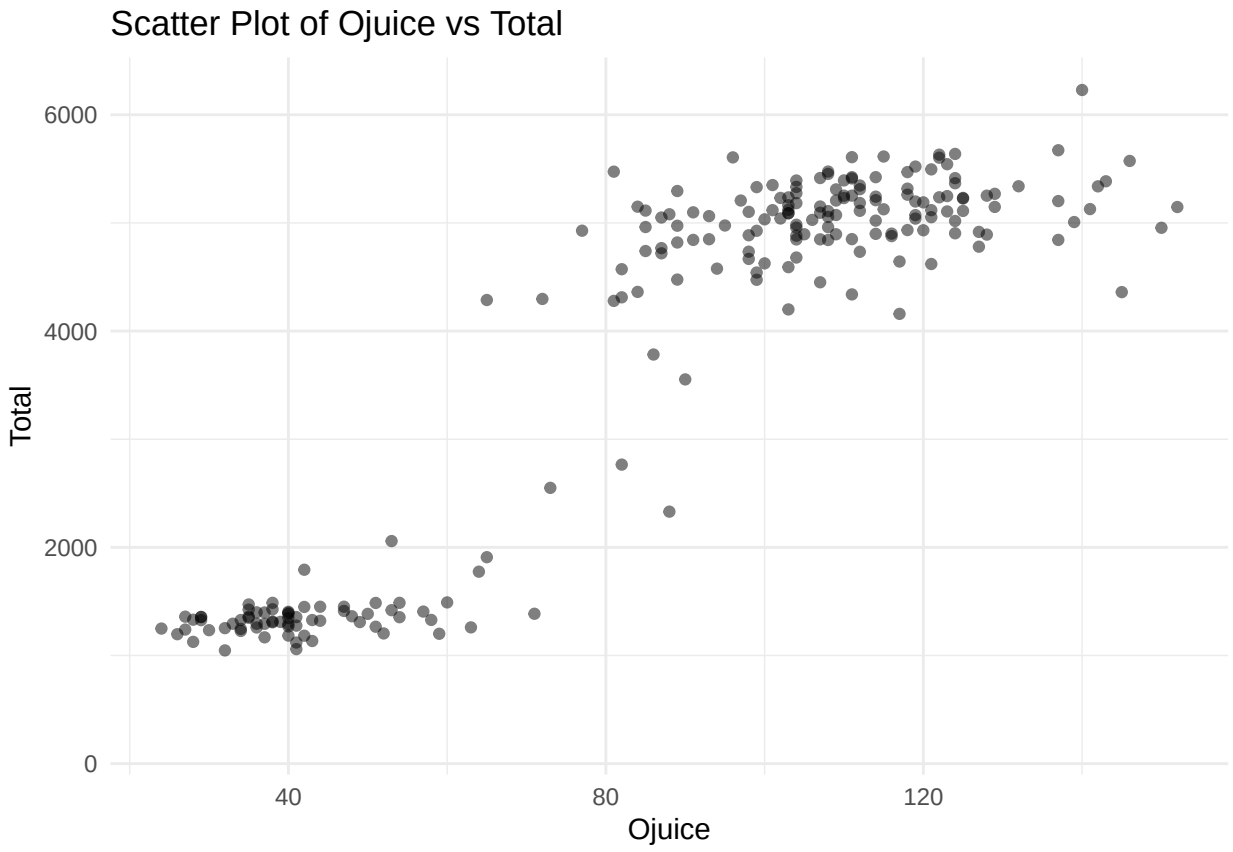

```
## Warning: Removed 210 rows containing missing values (‘geom_point()‘).
```



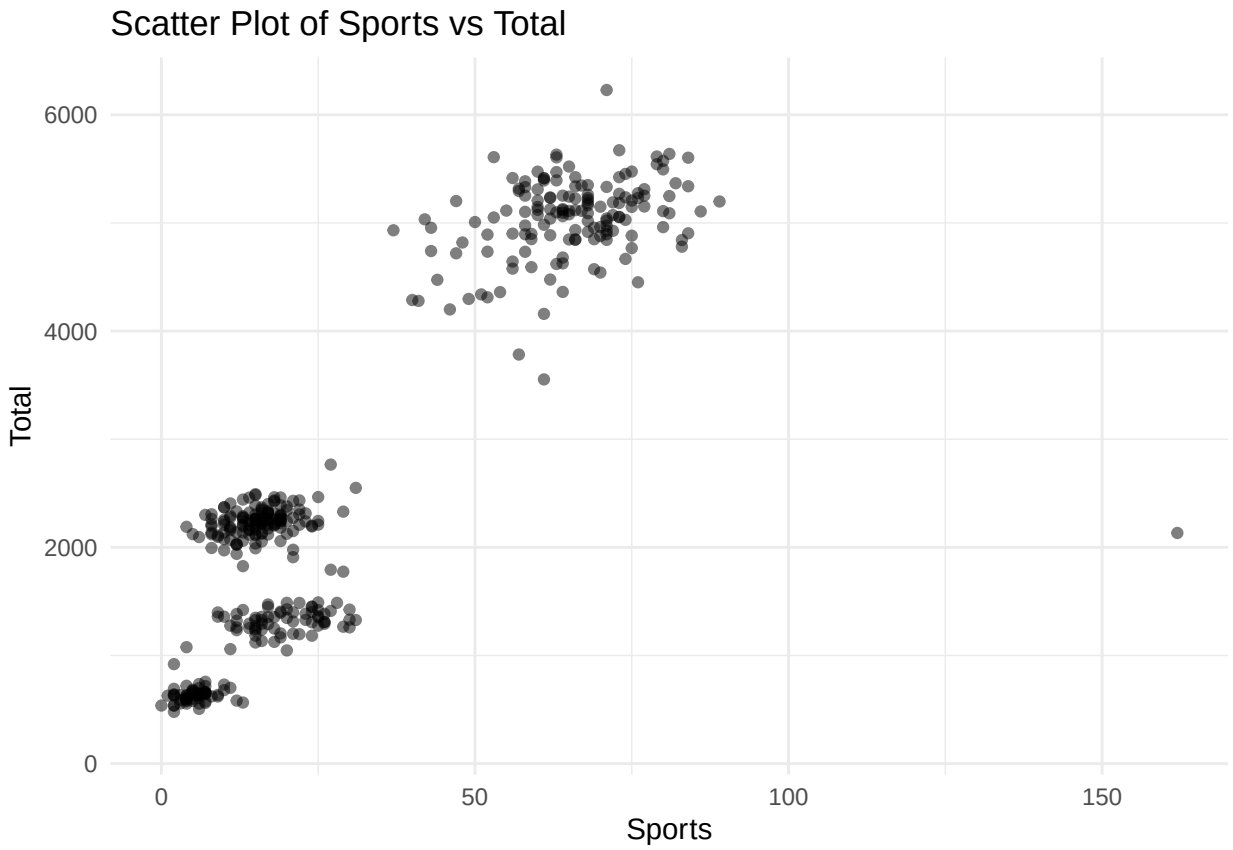

```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```




```
## Warning: Removed 410 rows containing missing values (‘geom_point()‘).
```




```
## Warning: Removed 218 rows containing missing values (‘geom_point()‘).
```




```
#' Summarize Sales by Dynamic Category
#'
#' Summarizes total sales by a specified category (e.g., Site, Intervention).
#' @param df Data frame containing the sales data.
#' @param category The name of the column to group by, as a string.
#' @return A data frame with the total sales summarized by the specified category.
summarise_sales <- function(df, category) {
  stat <- df %>%
    group_by(.data[[category]]) %>%
    summarise(Total_Sales = sum(Total, na.rm = TRUE), .groups = 'drop')
  return(stat)
}
summarise_sales(beverage_sales, "Site")
```

```
## # A tibble: 3 x 2
##   Site  Total_Sales
##   <chr>       <dbl>
## 1 HF         346461
## 2 NS         331309.
## 3 chop       851824
```

```
summarise_sales(beverage_sales, "Intervention")
```

```
## # A tibble: 9 x 2
##   Intervention Total_Sales
```

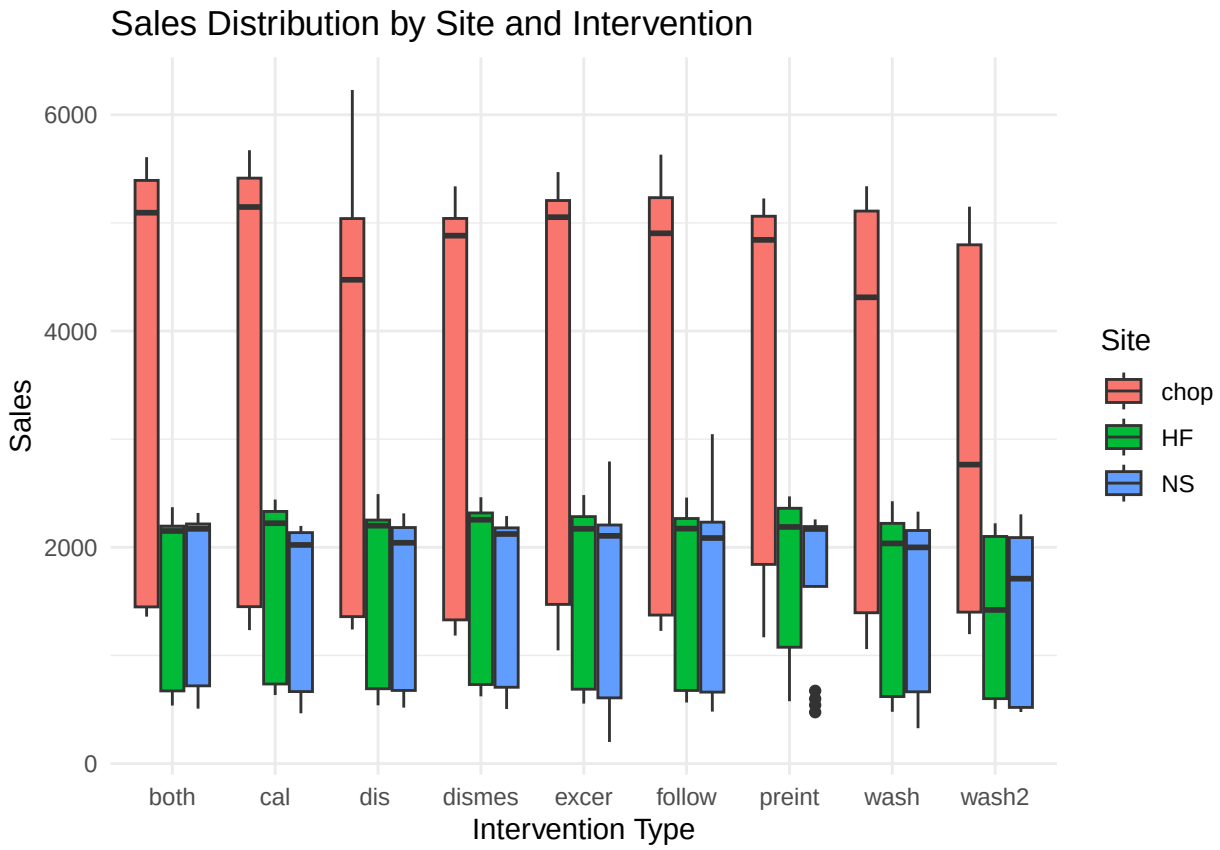
```
##   <chr>              <dbl>
## 1 "both"           161185
## 2 "cal"            160773
## 3 "dis "           149783.
## 4 "dismes"         156711
## 5 "excer"          153247.
## 6 "follow"         274933.
## 7 "preint"         203973
## 8 "wash"           228219
## 9 "wash2"           40770
```

```r
#' Handle Missing Data
#'
#' Provides an overview and simple strategies for handling missing data.
#' @param df Data frame with potential missing values.
#' @return Data frame with missing values handled.
handle_missing_data <- function(df) {
  # Overview of missing data
  missing_overview <- sapply(df, function(x) sum(is.na(x)))

  # Strategies for imputation or removal?

  return(list(MissingOverview = missing_overview, CleanedData = df))
}
handle_missing_data(beverage_sales)$MissingOverview
```

```
##        Count         DofW         Site Intervention      ZeroCal       Sugary
##            0            0            0            0            9            9
##     Juice100       Ojuice       Sports        Total
##          210          410          217           10
```

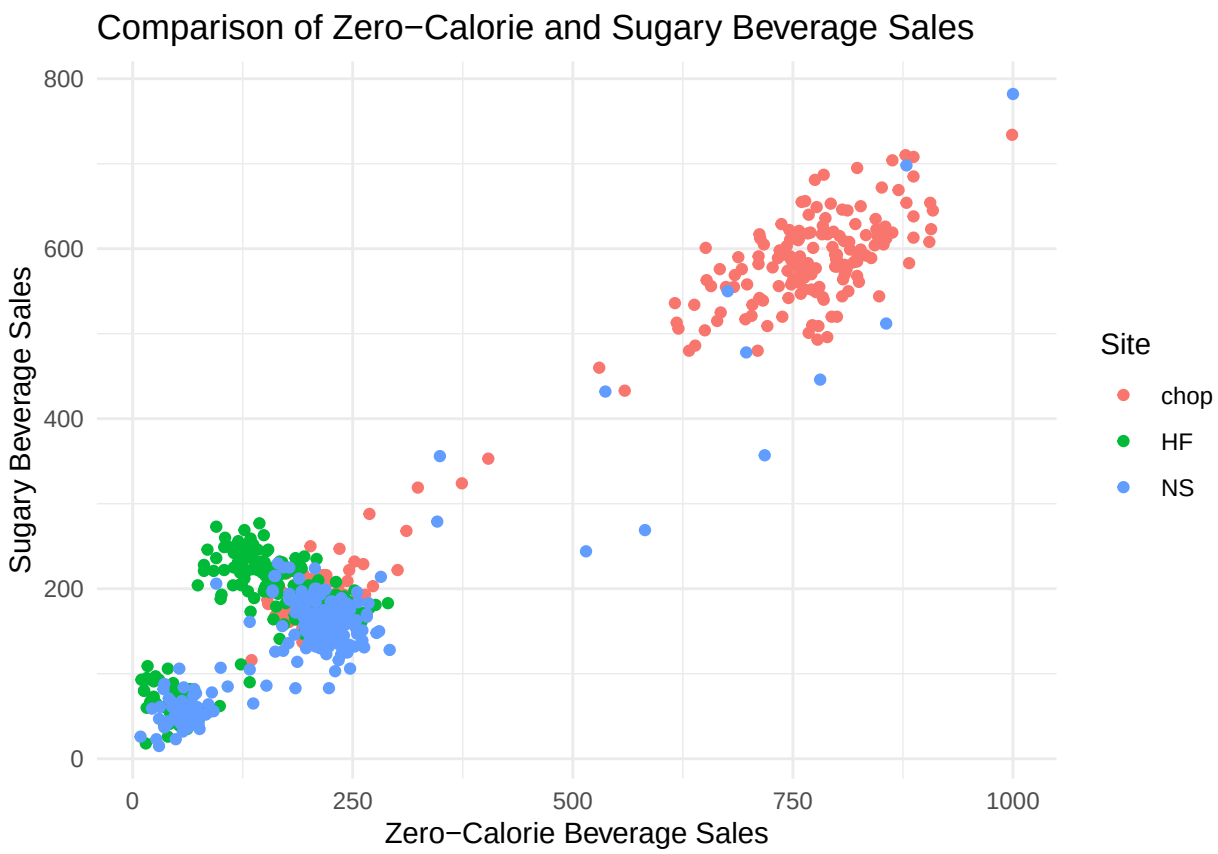
```r
#' Plot Boxplot for Sales Distribution by Site and Intervention
#'
#' Creates a boxplot to visualize the distribution of sales by site and intervention.
#' @param df Data frame containing the sales data.
#' @param x_var Name of the variable to be used on the x-axis, typically interventions.
#' @param y_var Name of the variable to be used on the y-axis, typically total sales.
#' @param fill_var Name of the variable to be used for fill color, typically site.
#' @param title Plot title.
#' @param x_lab Label for the x-axis.
#' @param y_lab Label for the y-axis.
#' @return A ggplot object representing the boxplot.
plot_sales_distribution <- function(df, x_var, y_var, fill_var, title = "Sales Distribution by Site and
                                    x_lab = "Intervention Type", y_lab = "Sales") {
  ggplot(df, aes_string(x = x_var, y = y_var, fill = fill_var)) +
    geom_boxplot() +
    theme_minimal() +
    labs(title = title, x = x_lab, y = y_lab)
}
plot_sales_distribution(beverage_sales, "Intervention", "Total", "Site")
```

```
## Warning: Removed 10 rows containing non-finite values (`stat_boxplot()`).
```




```
#' Plot Scatter for Zero-Calorie vs Sugary Beverage Sales
#'
#' Creates a scatter plot to visualize the relationship between zero-calorie and sugary beverage sales,
#' @param df Data frame containing the beverage sales data.
#' @param x_var Name of the variable representing zero-calorie beverage sales.
#' @param y_var Name of the variable representing sugary beverage sales.
#' @param color_var Name of the variable to be used for point colors, typically site.
#' @param title Plot title.
#' @param x_lab Label for the x-axis.
#' @param y_lab Label for the y-axis.
#' @return A ggplot object representing the scatter plot.
plot_beverage_sales_comparison <- function(df, x_var, y_var, color_var, title = "Comparison of Zero-Cal
                                           x_lab = "Zero-Calorie Beverage Sales", y_lab = "Sugary Bever
  ggplot(df, aes_string(x = x_var, y = y_var, color = color_var)) +
    geom_point() +
    theme_minimal() +
    labs(title = title, x = x_lab, y = y_lab)
}
plot_beverage_sales_comparison(beverage_sales, "ZeroCal", "Sugary", "Site")
```

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```




```
#' Execute EDA Functions
#'
#' Calls the defined functions to perform EDA on the dataset.
execute_eda <- function(df) {
  summary <- summarize_data(df)
  print(summary)

  plot_histograms(df)
  boxplots_by_category(df, "Site") # Replace "Site" with the appropriate categorical variable if differ
  scatter_plots(df)
  missing_data <- handle_missing_data(df)

  print(missing_data$MissingOverview)
}
# execute_eda(beverage_sales)
```

4. Formal Analysis

- Suggested statistical models and methods.
- Interpretation of results.

5. Conclusions

- Recommendations to the clients.

6. References

- Properly formatted citations.

7. Statistical Appendix

- Mathematical formulas.
- Additional tables/figures.