Homework 4

Han Nguyen - TXN200004

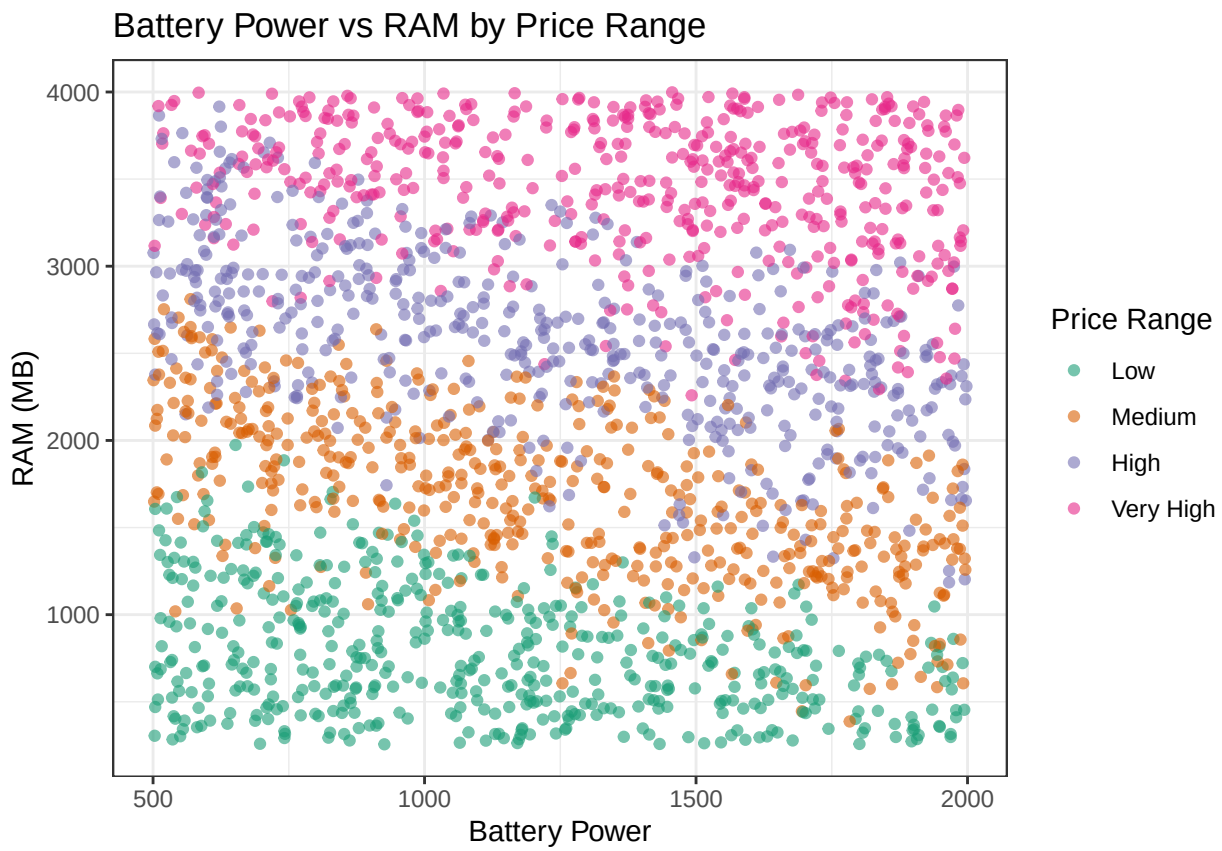
10/07/2025

Problem 1

```
# Load ggplot2
library(ggplot2)

# Read the data
mobile_data <- read.csv("train.csv")

# Convert price_range to factor with labels
mobile_data$price_range <- factor(mobile_data$price_range,
                                  levels = c(0, 1, 2, 3),
                                  labels = c("Low", "Medium", "High", "Very High"))
```

(a) Scatter plot: Battery Power vs RAM with colors by Price Range

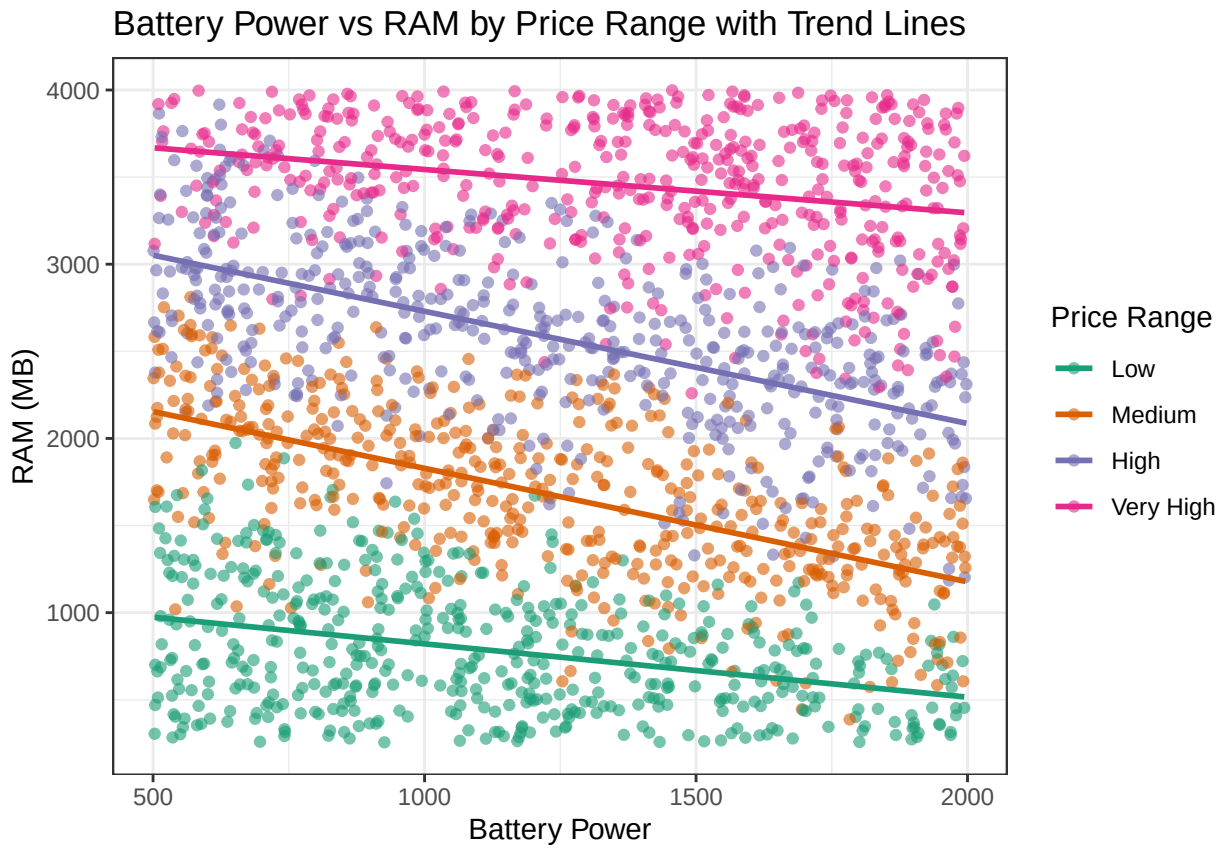
```
ggplot(mobile_data, aes(x = battery_power, y = ram, color = price_range)) +
  geom_point(alpha = 0.6) +
  labs(title = "Battery Power vs RAM by Price Range",
       x = "Battery Power",
       y = "RAM (MB)",
       color = "Price Range") +
  theme_bw() +
  scale_color_brewer(palette = "Dark2")
```



(b) Scatter plot with trend lines for each price range

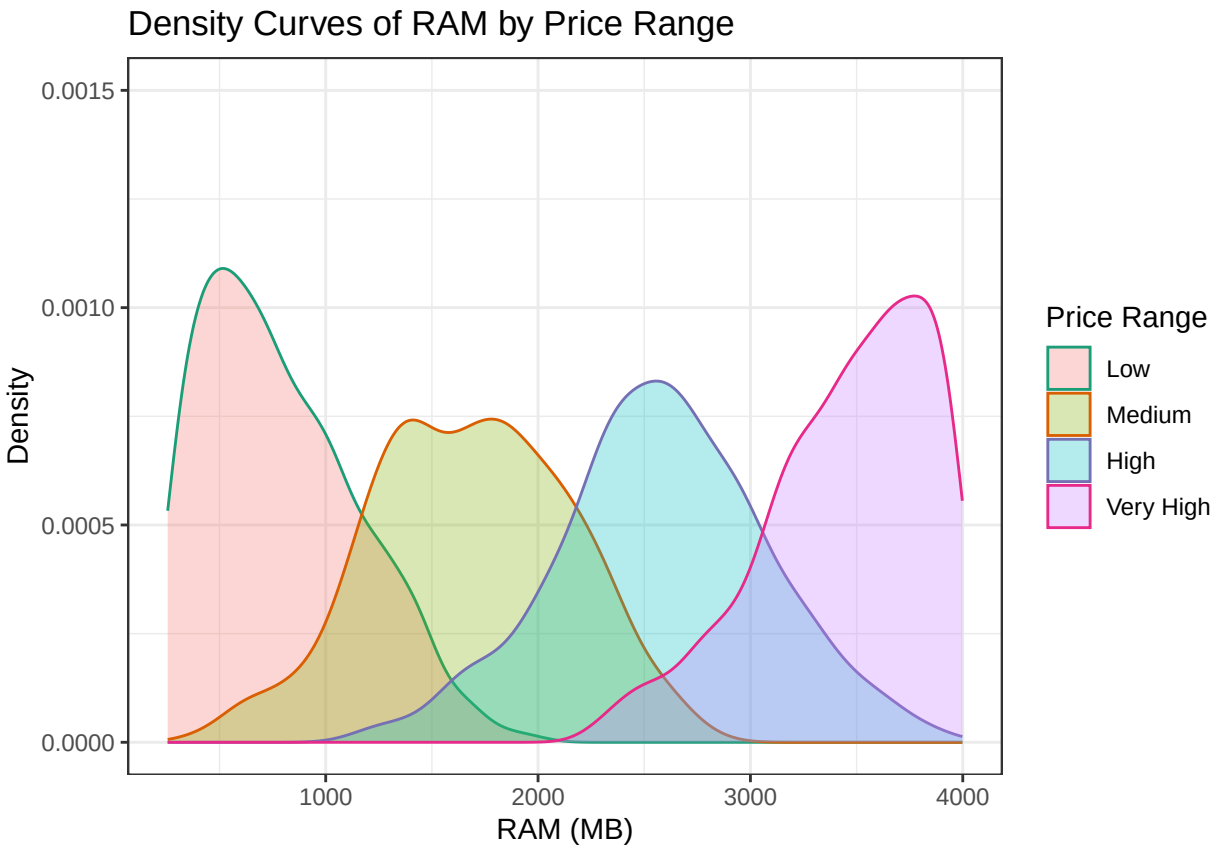
```
ggplot(mobile_data, aes(x = battery_power, y = ram, color = price_range)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Battery Power vs RAM by Price Range with Trend Lines",
       x = "Battery Power",
       y = "RAM (MB)",
       color = "Price Range") +
  theme_bw() +
  scale_color_brewer(palette = "Dark2")
```

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



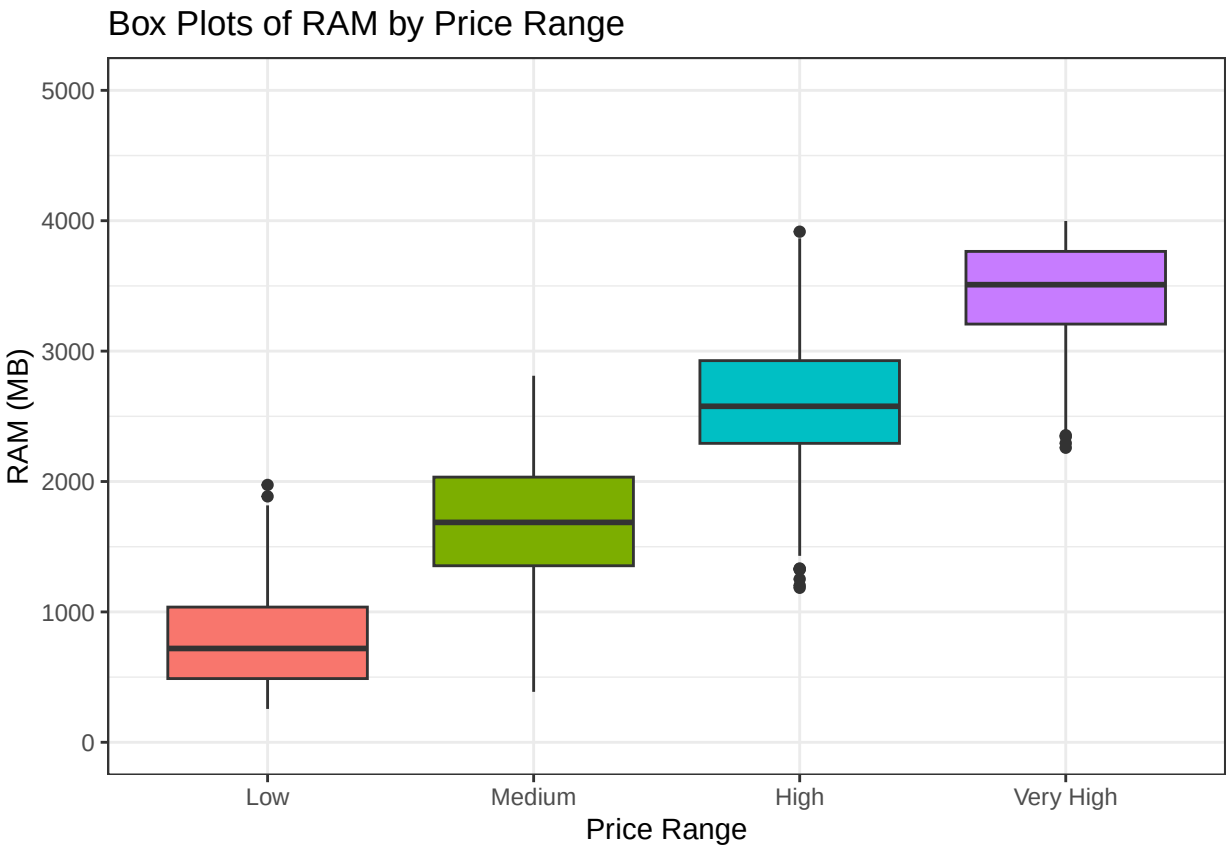
(c) Density curves of RAM for 4 price ranges

```
ggplot(mobile_data, aes(x = ram, fill = price_range, color = price_range)) +
  geom_density(alpha = 0.3) +
  labs(title = "Density Curves of RAM by Price Range",
       x = "RAM (MB)",
       y = "Density",
       fill = "Price Range",
       color = "Price Range") +
  theme_bw() +
  ylim(0, 0.0015) +
  scale_color_brewer(palette = "Dark2")
```



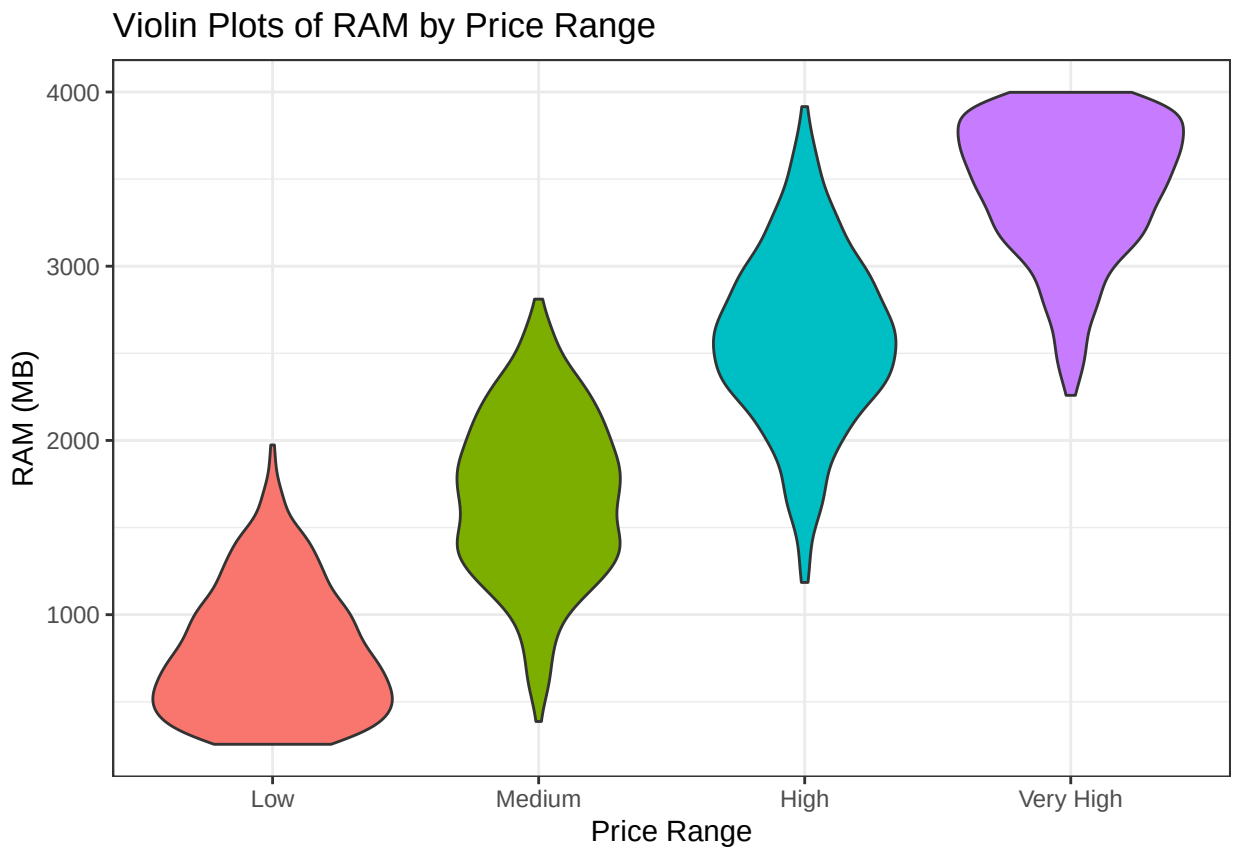
(d) Box plots of RAM for 4 price ranges

```
ggplot(mobile_data, aes(x = price_range, y = ram, fill = price_range)) +
  geom_boxplot() +
  labs(title = "Box Plots of RAM by Price Range",
       x = "Price Range",
       y = "RAM (MB)") +
  theme_bw() +
  theme(legend.position = "none") +
  ylim(0, 5000) +
  scale_color_brewer(palette = "Dark2")
```



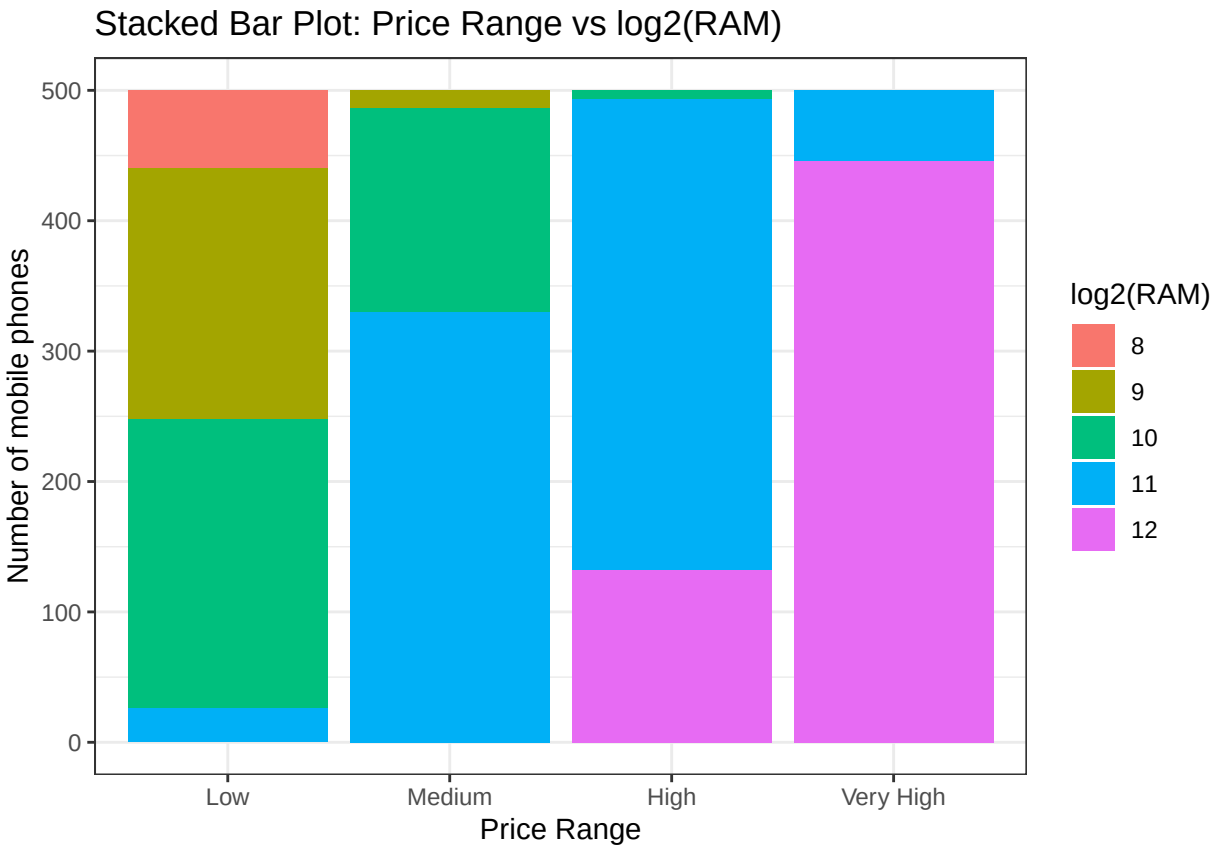
(e) Violin plot of RAM for 4 price ranges

```
ggplot(mobile_data, aes(x = price_range, y = ram, fill = price_range)) +
  geom_violin() +
  labs(title = "Violin Plots of RAM by Price Range",
       x = "Price Range",
       y = "RAM (MB)") +
  theme_bw() +
  theme(legend.position = "none") +
  scale_color_brewer(palette = "Dark2")
```



(f) Stacked bar plot: Price Range vs log2(RAM)

```
# Create log2(ram) variable
mobile_data$log2_ram <- factor(round(log2(mobile_data$ram)))

ggplot(mobile_data, aes(x = price_range, fill = log2_ram)) +
  geom_bar(position = "stack") +
  labs(title = "Stacked Bar Plot: Price Range vs log2(RAM)",
       x = "Price Range",
       y = "Number of mobile phones",
       fill = "log2(RAM)") +
  theme_bw()
```


Stacked Bar Plot: Price Range vs log2(RAM)
Number of mobile phones
500
400
300
200
100
0
Low
Medium
High
Very High
Price Range
log2(RAM)
8
9
10
11
12

Problem 2

```
# Load necessary packages
library(ggplot2)
library(UsingR)
```

```
## Loading required package: MASS
```

```
## Loading required package: HistData
```

```
## Loading required package: Hmisc
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# Load the UScereal dataset
data("UScereal")
```

(a) Replace manufacturer abbreviations with full names

```
levels(UScereal$mfr) <- c("General Mills", "Kelloggs", "Nabisco",
                          "Post", "Quaker Oats", "Ralston Purina")
```

(b) Convert shelf to factor with proper labels

```
UScereal$shelf <- factor(UScereal$shelf,
                         levels = c(1, 2, 3),
                         labels = c("Lower", "Middle", "Upper"))
```

(c) Create Product variable from row names

```
UScereal$product <- rownames(UScereal)
```

Check the structure:

```
str(UScereal)
```

```
## 'data.frame':    65 obs. of  12 variables:
##  $ mfr      : Factor w/ 6 levels "General Mills",..: 3 2 2 1 2 1 6 4 5 1 ...
##  $ calories : num  212 212 100 147 110 ...
##  $ protein  : num  12.12 12.12 8 2.67 2 ...
##  $ fat      : num  3.03 3.03 0 2.67 0 ...
##  $ sodium   : num  394 788 280 240 125 ...
##  $ fibre    : num  30.3 27.3 28 2 1 ...
##  $ carbo    : num  15.2 21.2 16 14 11 ...
##  $ sugars   : num  18.2 15.2 0 13.3 14 ...
##  $ shelf    : Factor w/ 3 levels "Lower","Middle",..: 3 3 3 1 2 3 1 3 2 1 ...
##  $ potassium: num  848.5 969.7 660 93.3 30 ...
##  $ vitamins : Factor w/ 3 levels "100%","enriched",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ product  : chr  "100% Bran" "All-Bran" "All-Bran with Extra Fiber" "Apple Cinnamon Cheerios" ...
```

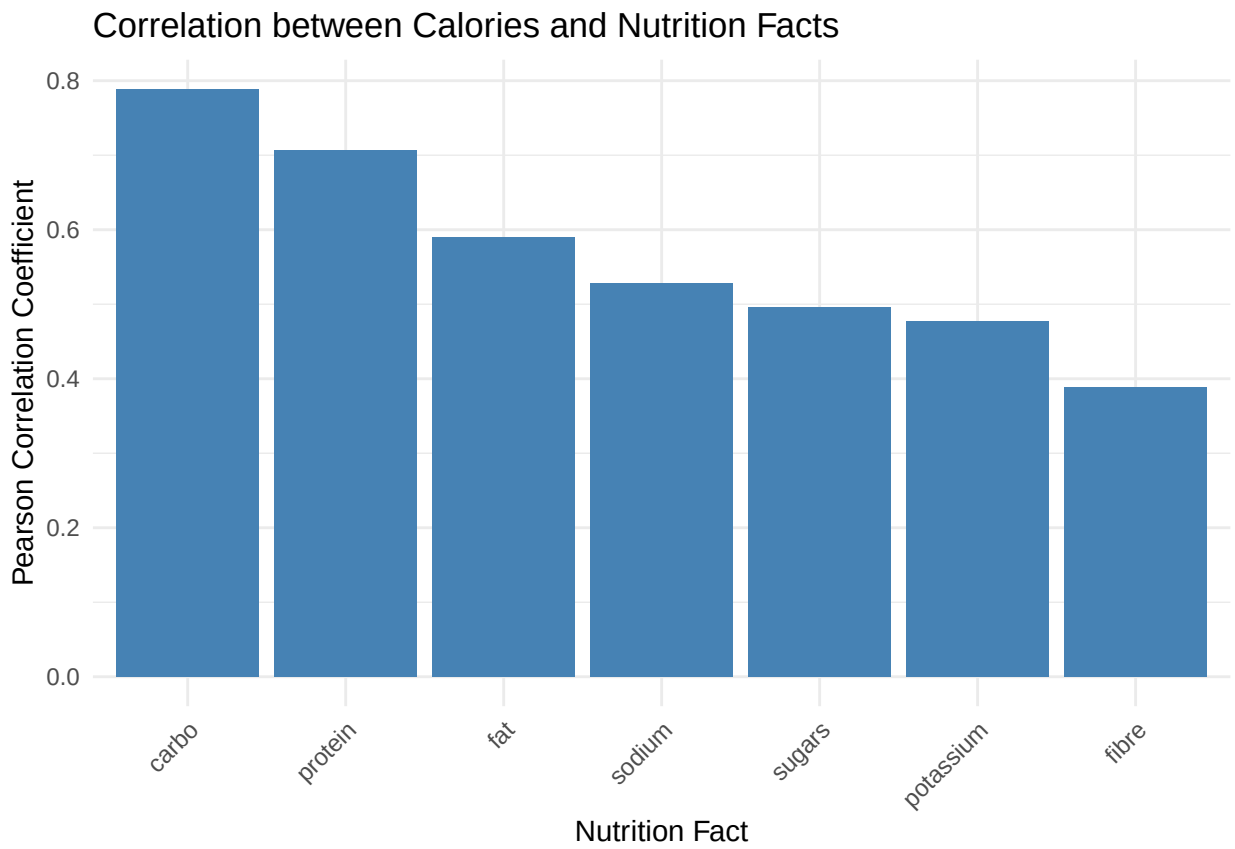
(d) Pearson correlation between calories and nutrition facts

```r
# Calculate correlations
nutrition_vars <- c("protein", "fat", "sodium", "fibre", "carbo",
                    "sugars", "potassium")
correlations <- sapply(nutrition_vars, function(var) {
  cor(UScereal$calories, UScereal[[var]])
})

# Display correlations
print(round(correlations, 4))
```

```
##   protein       fat    sodium     fibre     carbo    sugars potassium
##    0.7060    0.5902    0.5287    0.3882    0.7887    0.4953    0.4766
```

(e) Bar plot of correlations in decreasing order

```r
# Create data frame for plotting
cor_df <- data.frame(
  nutrition = names(correlations),
  correlation = correlations
)
cor_df <- cor_df[order(-cor_df$correlation), ]
cor_df$nutrition <- factor(cor_df$nutrition, levels = cor_df$nutrition)

ggplot(cor_df, aes(x = nutrition, y = correlation)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Correlation between Calories and Nutrition Facts",
       x = "Nutrition Fact",
       y = "Pearson Correlation Coefficient") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



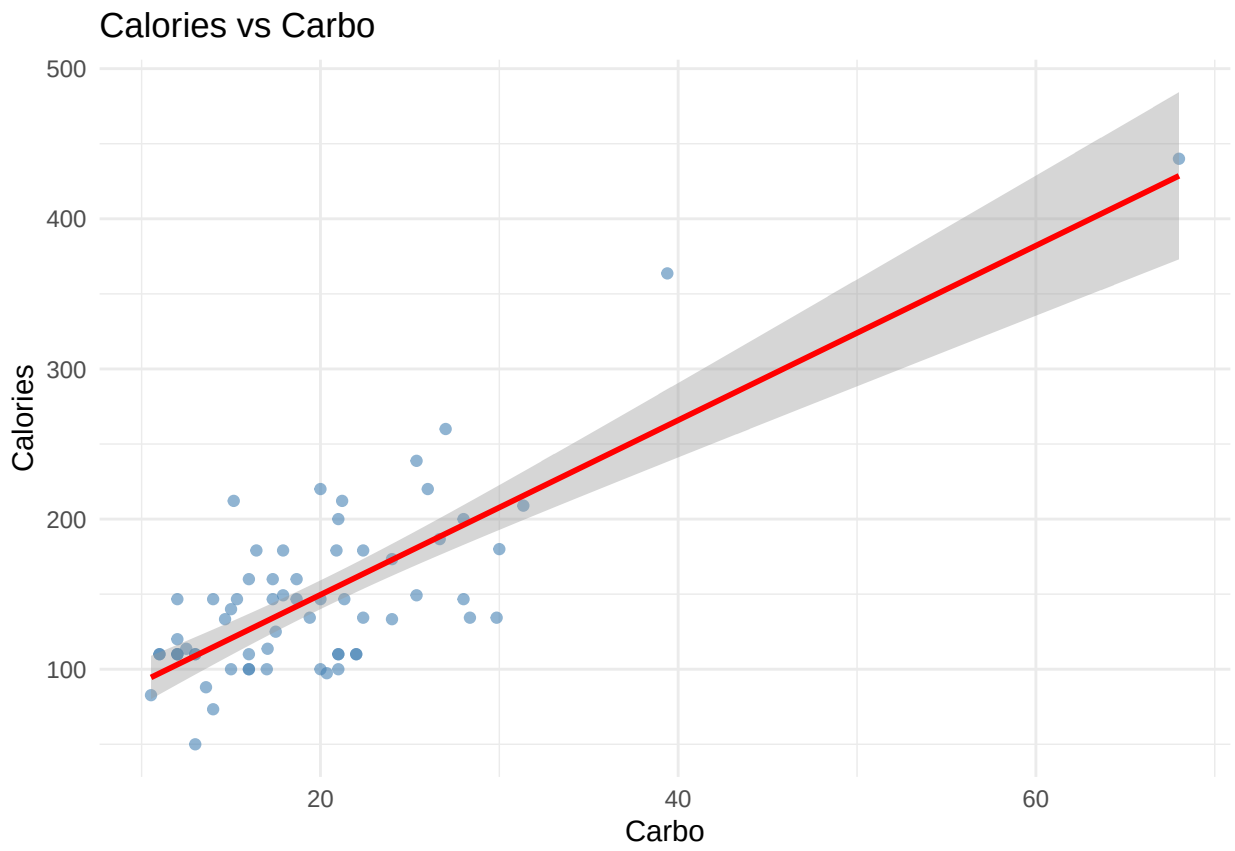
The nutrition fact with the highest correlation to calories is **carbo** with a correlation of 0.7887.

(f) Scatter plot with trend line for highest correlation

```
# Find the nutrition fact with highest correlation
highest_cor_var <- names(which.max(correlations))

ggplot(UScereal, aes_string(x = highest_cor_var, y = "calories")) +
  geom_point(color = "steelblue", alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(title = paste("Calories vs", tools::toTitleCase(highest_cor_var)),
       x = tools::toTitleCase(highest_cor_var),
       y = "Calories") +
  theme_minimal()
```

```
## 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.
```

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




```
# Get the linear model for interpretation
lm_model <- lm(calories ~ fat, data = UScereal)
intercept <- round(coef(lm_model)[1], 2)
slope <- round(coef(lm_model)[2], 2)
```

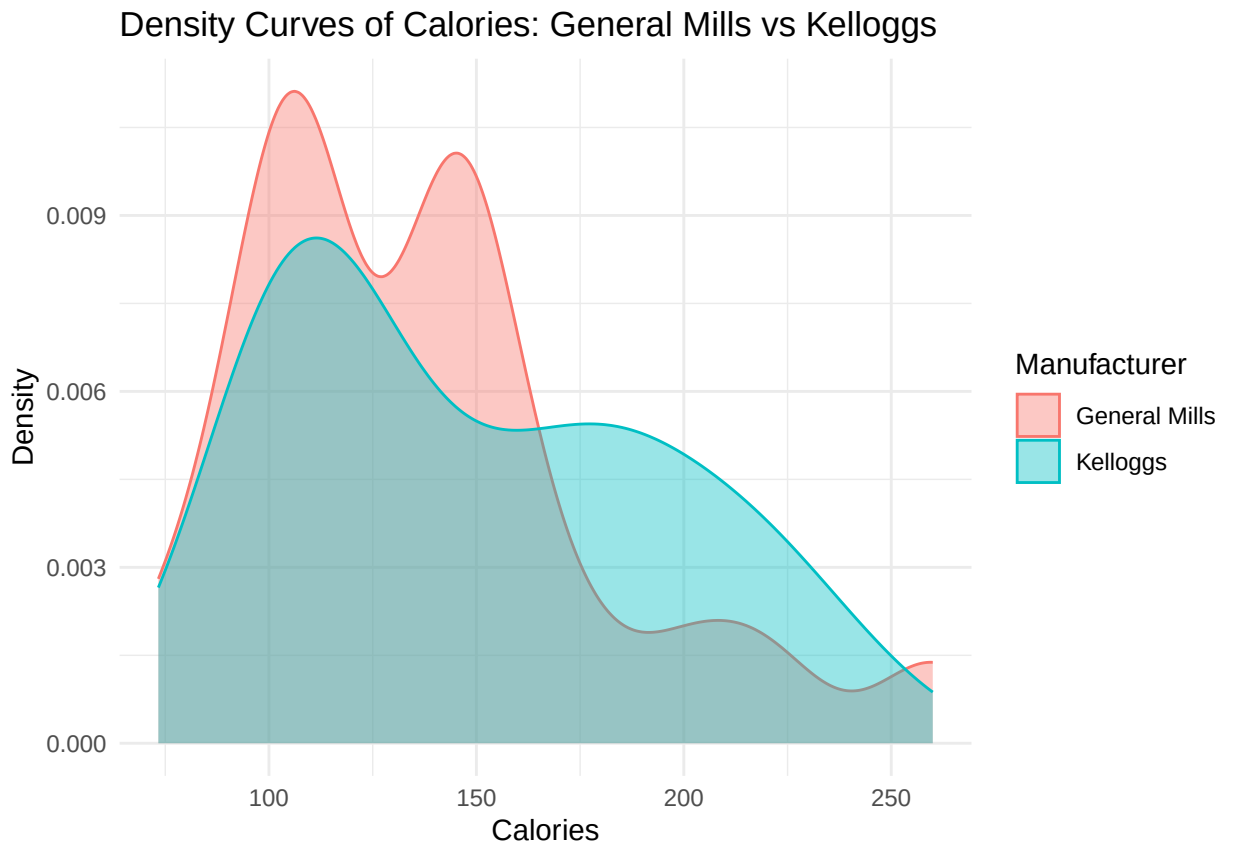
Interpretation:

- **Intercept** (117.6): When fat content is 0 grams, the predicted calorie content is approximately 117.6 calories.
- **Slope** (22.36): For each additional gram of fat, the calorie content increases by approximately 22.36 calories on average. This makes sense because fat contains about 9 calories per gram.

(g) Density curves comparing General Mills and Kelloggs

```
# Subset data for General Mills and Kelloggs
gm_kelloggs <- subset(UScereal, mfr %in% c("General Mills", "Kelloggs"))

ggplot(gm_kelloggs, aes(x = calories, fill = mfr, color = mfr)) +
  geom_density(alpha = 0.4) +
  labs(title = "Density Curves of Calories: General Mills vs Kelloggs",
       x = "Calories",
       y = "Density",
       fill = "Manufacturer",
```

```
      color = "Manufacturer") +
  theme_minimal()
```

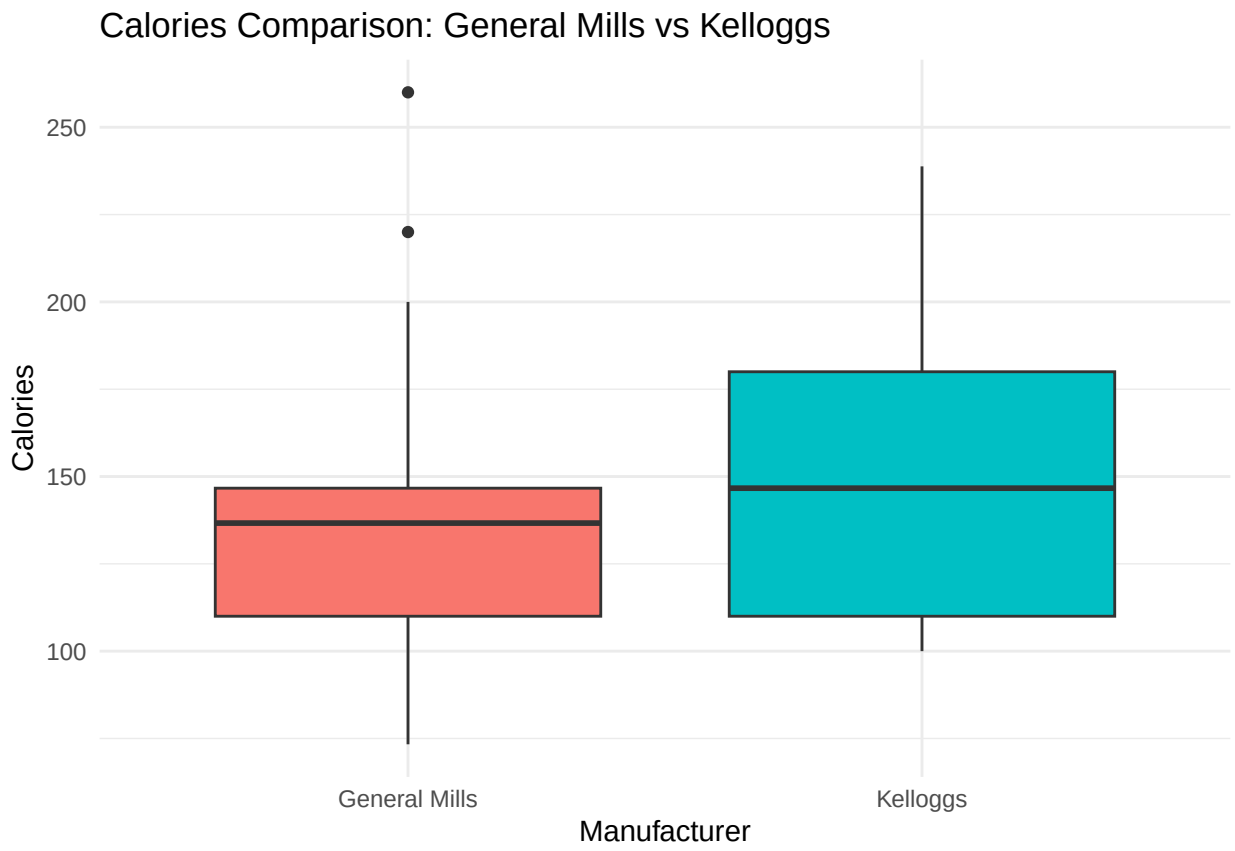


Shape descriptions:

- **General Mills:** The distribution is roughly unimodal with a peak around 110-120 calories. It shows a slight right skew with some cereals extending toward higher calorie values.
- **Kelloggs:** The distribution is also unimodal but appears more spread out with a peak around 100-110 calories. It has a flatter shape compared to General Mills, indicating more variability in calorie content.

(h) Box plot comparing calories between manufacturers

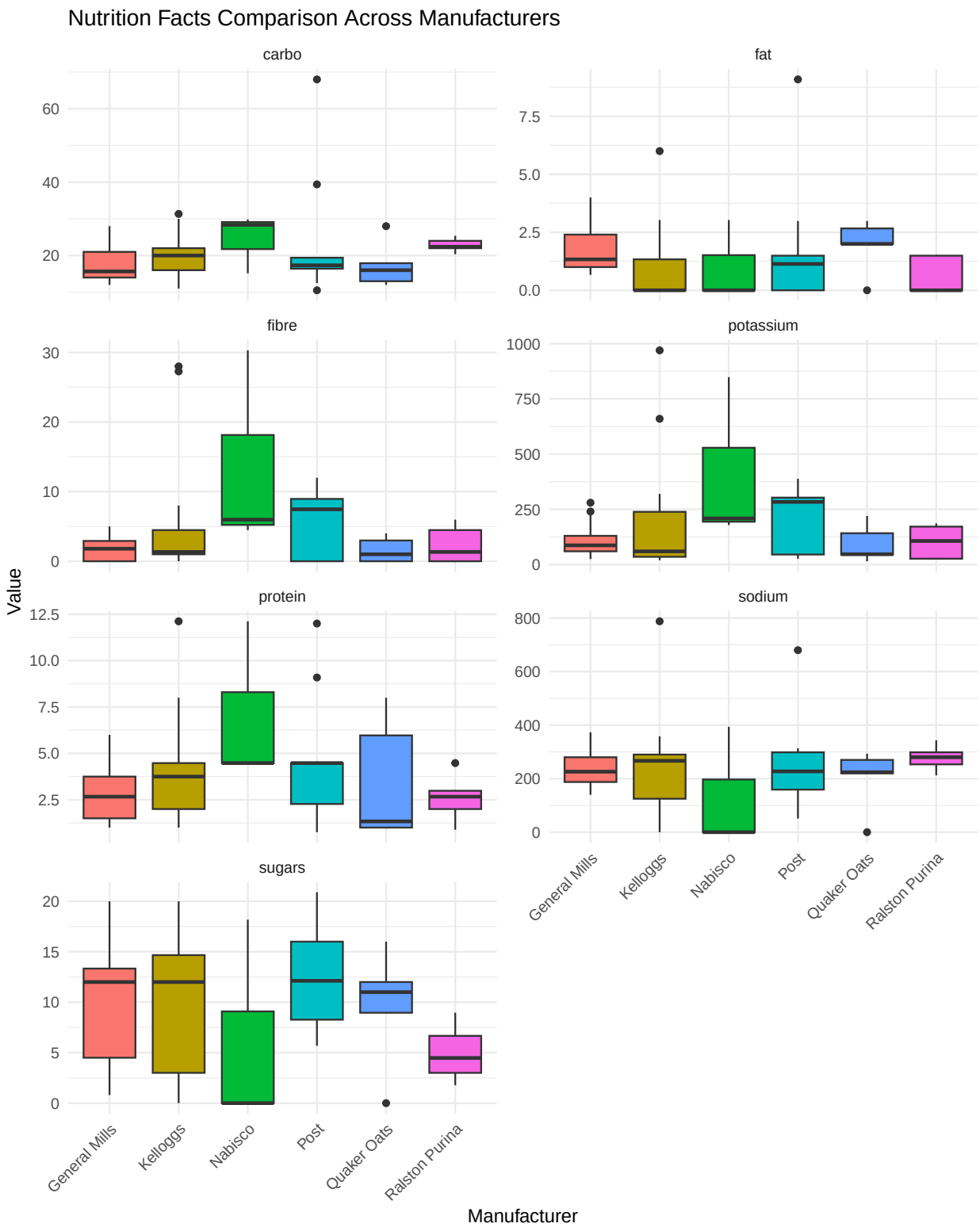
```
ggplot(gm_kelloggs, aes(x = mfr, y = calories, fill = mfr)) +
  geom_boxplot() +
  labs(title = "Calories Comparison: General Mills vs Kelloggs",
       x = "Manufacturer",
       y = "Calories") +
  theme_minimal() +
  theme(legend.position = "none")
```



The box plots show that General Mills and Kelloggs have similar median calorie values (around 110 calories). However, General Mills has a slightly more compact distribution with fewer outliers, while Kelloggs shows more variability. The overlap in the interquartile ranges suggests that calories are not significantly different between these two manufacturers.

(i) Side-by-side box plots for seven nutrition facts

```
# Reshape data manually without reshape2
nutrition_long <- data.frame(
  mfr = rep(UScereal$mfr, times = length(nutrition_vars)),
  Nutrition = rep(nutrition_vars, each = nrow(UScereal)),
  Value = c(UScereal$protein, UScereal$fat, UScereal$sodium,
            UScereal$fibre, UScereal$carbo, UScereal$sugars,
            UScereal$potassium)
)

ggplot(nutrition_long, aes(x = mfr, y = Value, fill = mfr)) +
  geom_boxplot() +
  facet_wrap(~ Nutrition, scales = "free_y", ncol = 2) +
  labs(title = "Nutrition Facts Comparison Across Manufacturers",
       x = "Manufacturer",
       y = "Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```



Discussion:

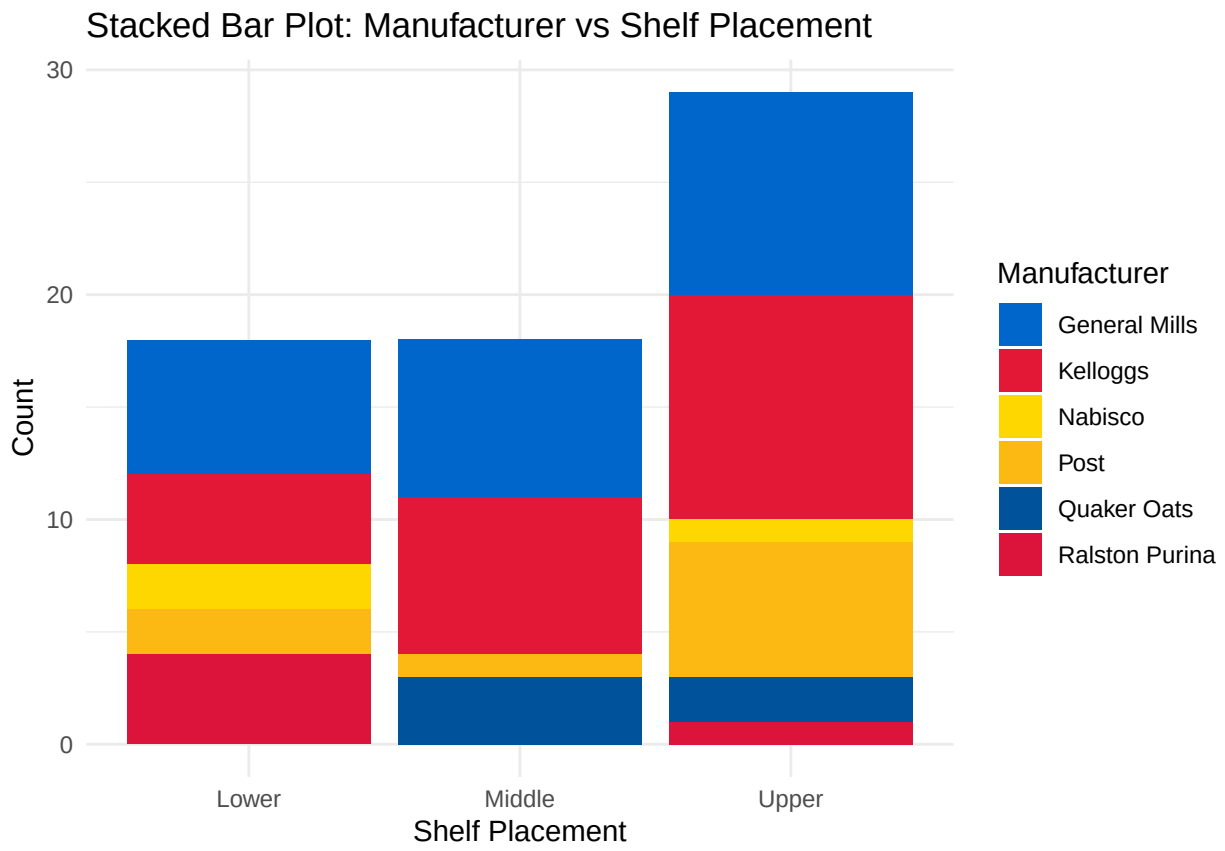
Based on the box plots, **Nabisco** and **Quaker Oats** appear to aim for healthier diets. They show:

- Higher fiber content (important for digestive health)
- Lower sodium levels (better for heart health)
- Moderate protein levels
- Lower sugar content compared to some other manufacturers

Kelloggs and Post tend to have higher sugar content, while Ralston Purina shows more variability across different nutrition metrics.

(j) Stacked bar plot: Manufacturer vs Shelf placement

```
# Create stacked bar plot with custom colors inspired by brand logos
# General Mills (blue), Kelloggs (red), Nabisco (blue/yellow),
# Post (yellow), Quaker Oats (blue/red), Ralston Purina (red/blue)
brand_colors <- c("General Mills" = "#0066CC",
                  "Kelloggs" = "#E31837",
                  "Nabisco" = "#FFD700",
                  "Post" = "#FDB913",
                  "Quaker Oats" = "#00539B",
                  "Ralston Purina" = "#DC143C")

ggplot(UScereal, aes(x = shelf, fill = mfr)) +
  geom_bar(position = "stack") +
  scale_fill_manual(values = brand_colors) +
  labs(title = "Stacked Bar Plot: Manufacturer vs Shelf Placement",
       x = "Shelf Placement",
       y = "Count",
       fill = "Manufacturer") +
  theme_minimal()
```



The stacked bar plot shows the relationship between manufacturers and shelf placement. The middle shelf has the most products, followed by upper and lower shelves. General Mills and Kelloggs, being the dominant manufacturers, have products across all shelf levels, with strong presence on the middle and upper shelves where products are most visible to shoppers.