Lab5.R

2447218

2024-11-21

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.4.2
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.4.2
```

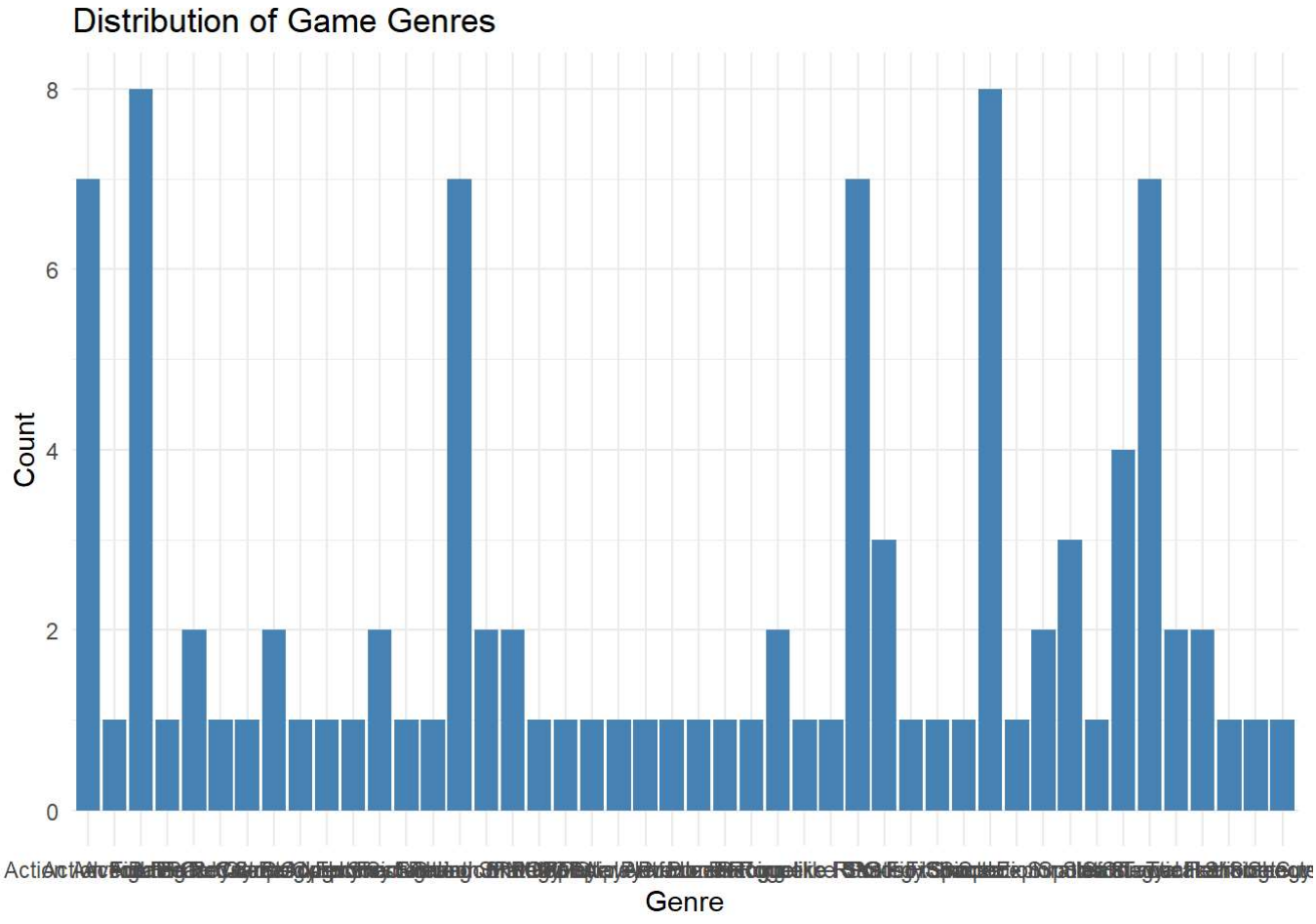
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
games_data <- read_csv("D:/MCA/Assignments/Rprogramming/Lab5/datset.csv")
```

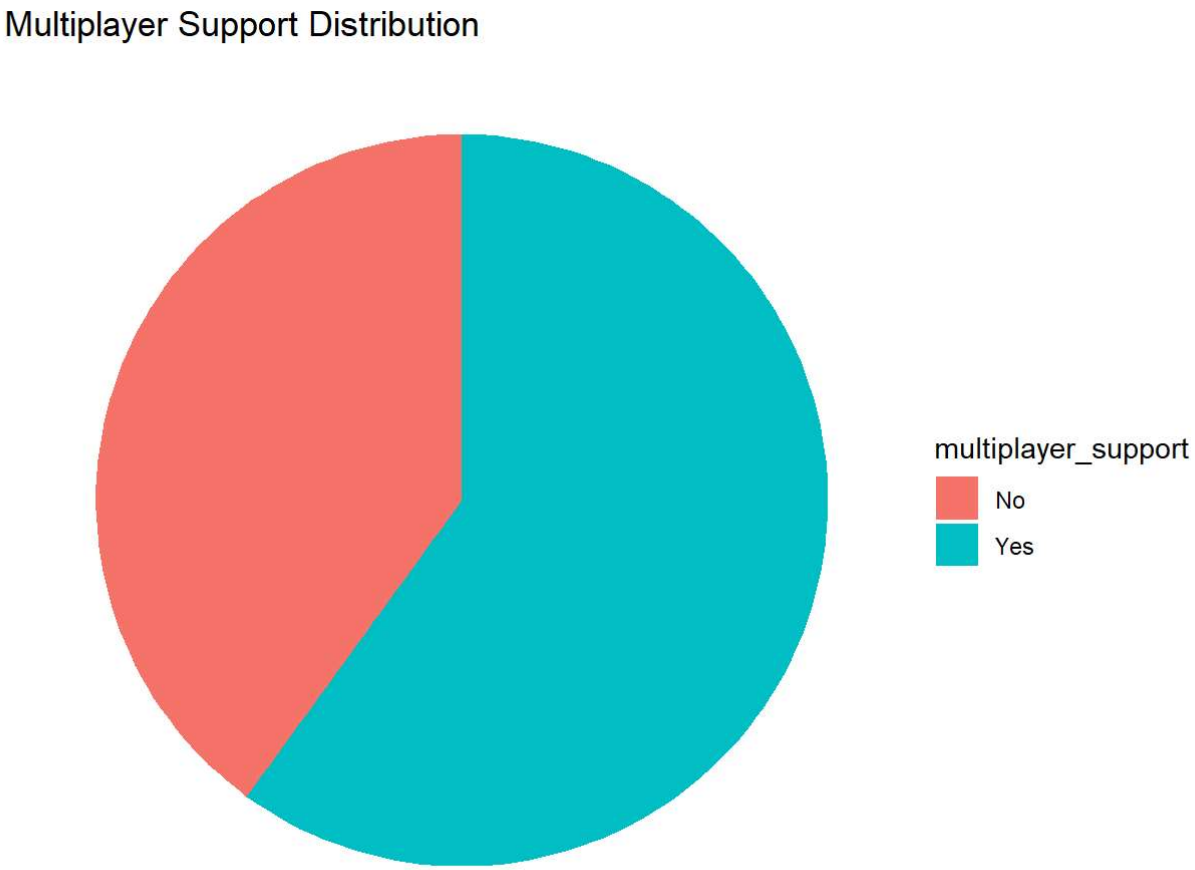
```
## Rows: 100 Columns: 14
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): title, genre, multiplayer_support, complexity_level, publisher, pla...
## dbl (7): game_id, price, user_rating, release_year, developer_rating, playti...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

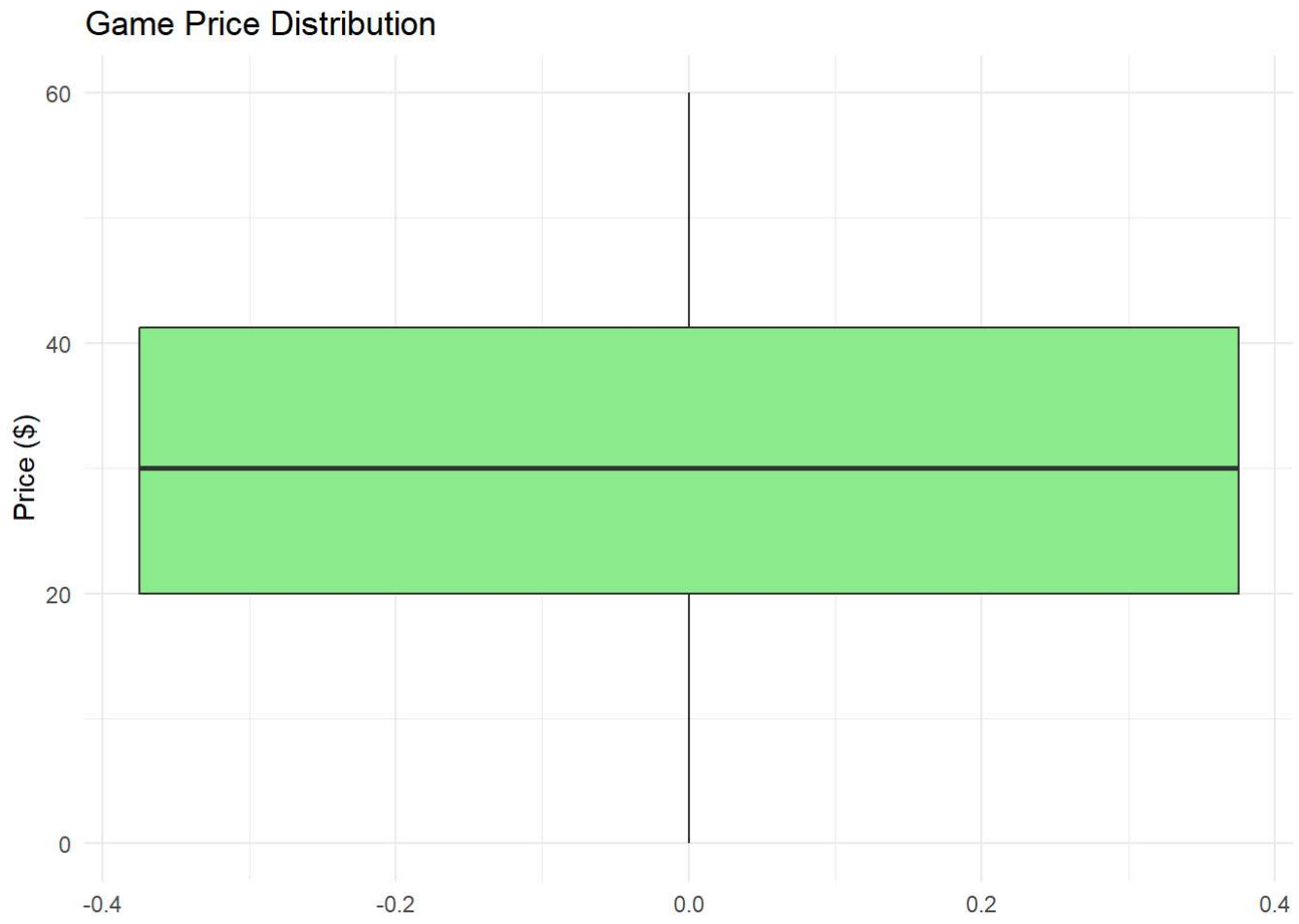
```
# 1. Single Variable Plots
# Bar Plot (Genre)
plt_bar_genre <- ggplot(games_data, aes(x = genre)) +
  geom_bar(fill = "steelblue") +
  theme_minimal() +
  labs(title = "Distribution of Game Genres", x = "Genre", y = "Count")
print(plt_bar_genre)
```



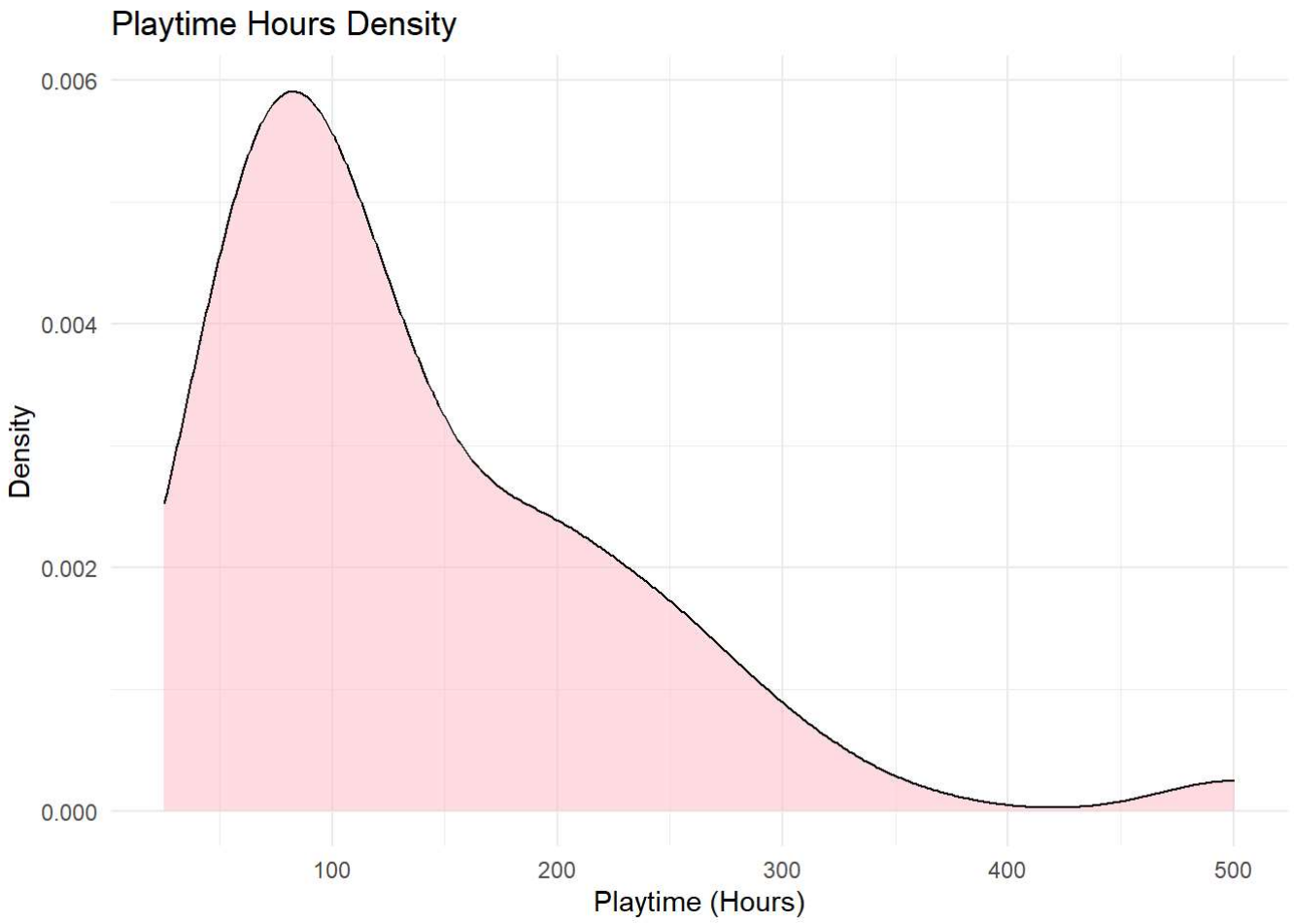

```
# Pie Chart (Multiplayer Support)
plt_pie_multiplayer <- ggplot(games_data, aes(x = "", fill = multiplayer_support)) +
  geom_bar(width = 1) +
  coord_polar("y") +
  theme_void() +
  labs(title = "Multiplayer Support Distribution")
print(plt_pie_multiplayer)
```



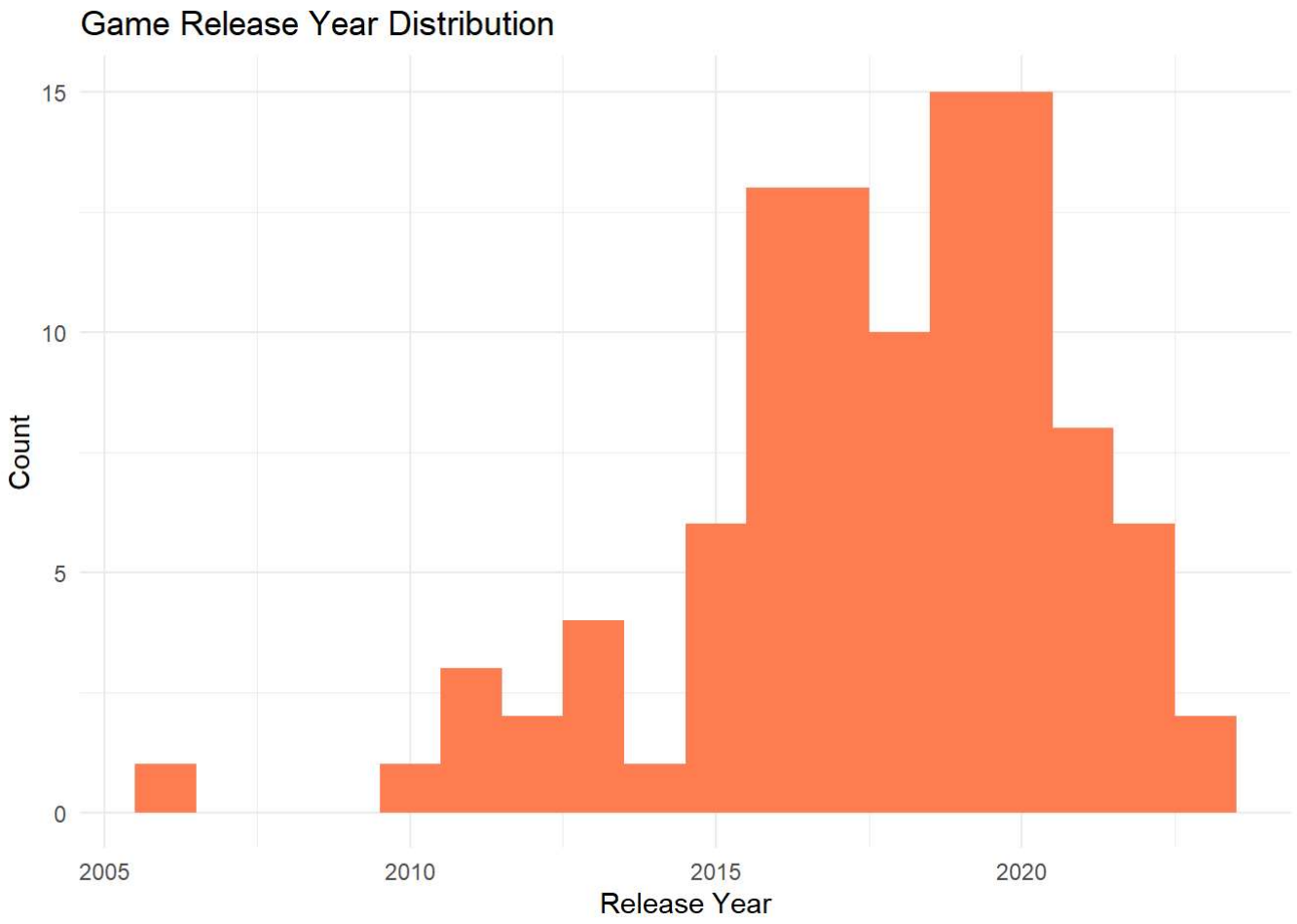

```
# Box Plot (Price)
plt_box_price <- ggplot(games_data, aes(y = price)) +
  geom_boxplot(fill = "lightgreen") +
  theme_minimal() +
  labs(title = "Game Price Distribution", y = "Price ($)")
print(plt_box_price)
```



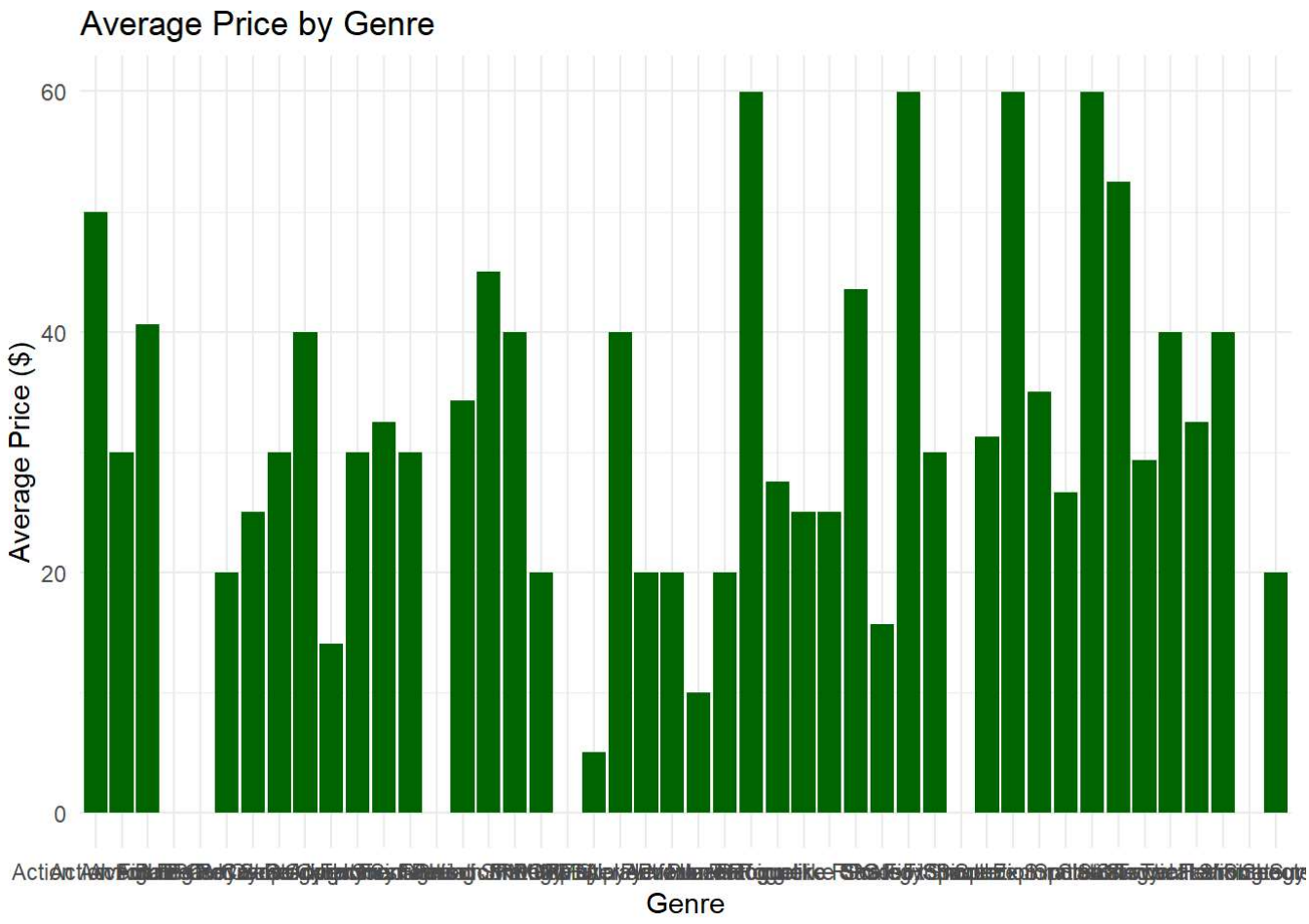

```
# Density Plot (Playtime)
plt_density_playtime <- ggplot(games_data, aes(x = playtime_hours)) +
  geom_density(fill = "pink", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Playtime Hours Density", x = "Playtime (Hours)", y = "Density")
print(plt_density_playtime)
```



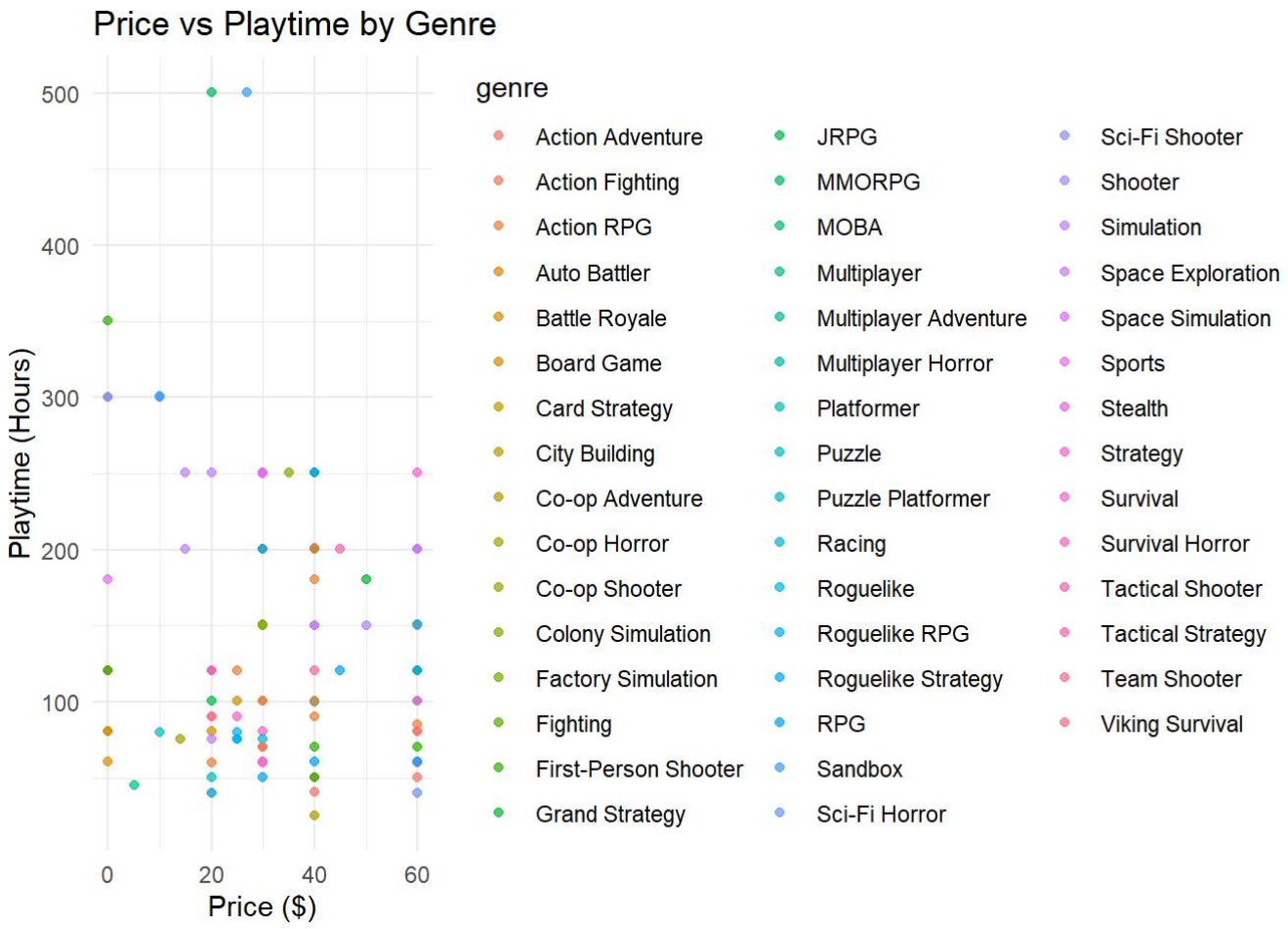

```
# Histogram (Release Year)
plt_hist_release <- ggplot(games_data, aes(x = release_year)) +
  geom_histogram(binwidth = 1, fill = "coral") +
  theme_minimal() +
  labs(title = "Game Release Year Distribution", x = "Release Year", y = "Count")
print(plt_hist_release)
```



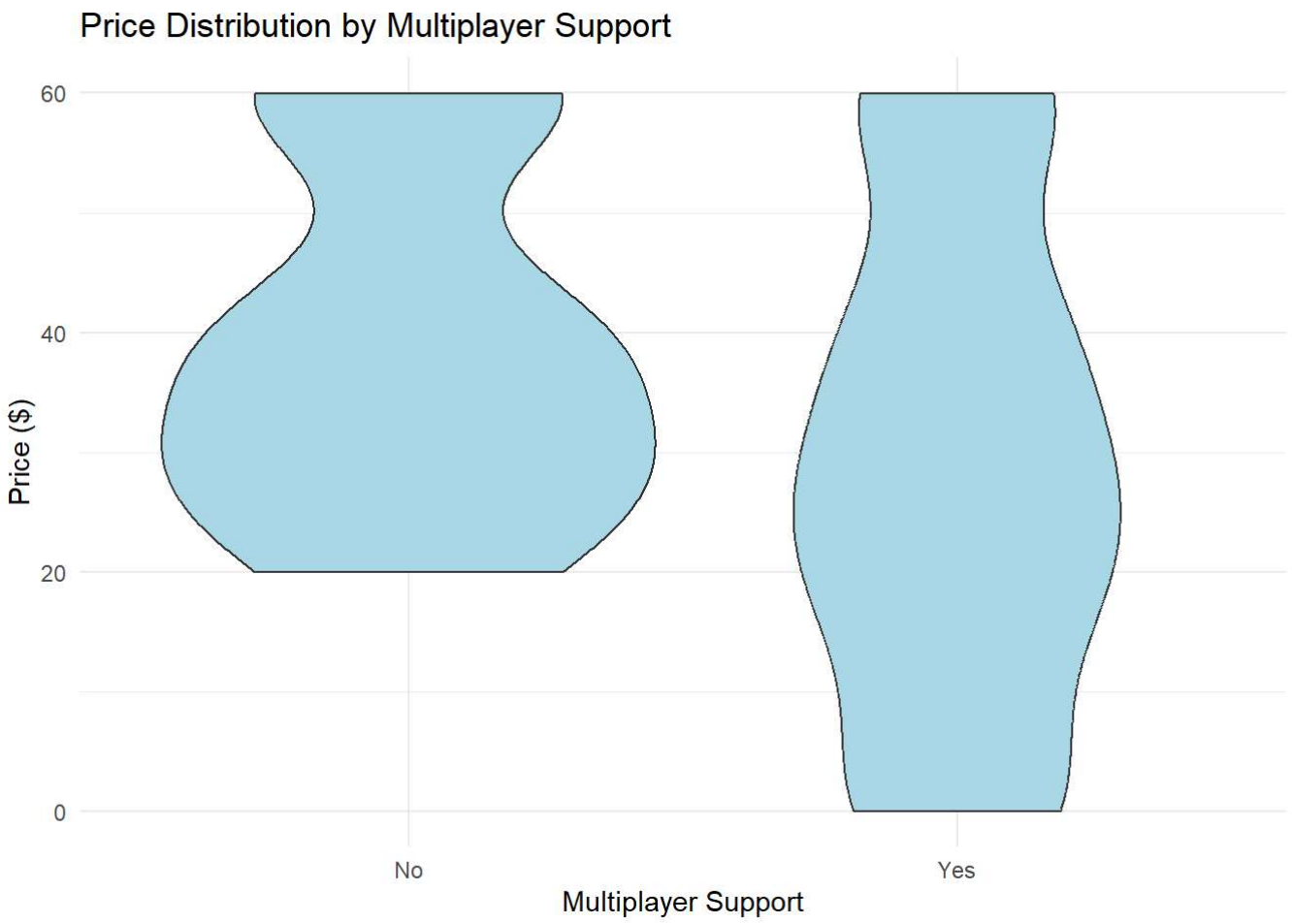

```
# 2. Two Variable Plots
# Bar Plot (Average Price by Genre)
plt_bar_price_genre <- games_data %>%
  group_by(genre) %>%
  summarise(avg_price = mean(price)) %>%
  ggplot(aes(x = genre, y = avg_price)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  theme_minimal() +
  labs(title = "Average Price by Genre", x = "Genre", y = "Average Price ($)")
print(plt_bar_price_genre)
```



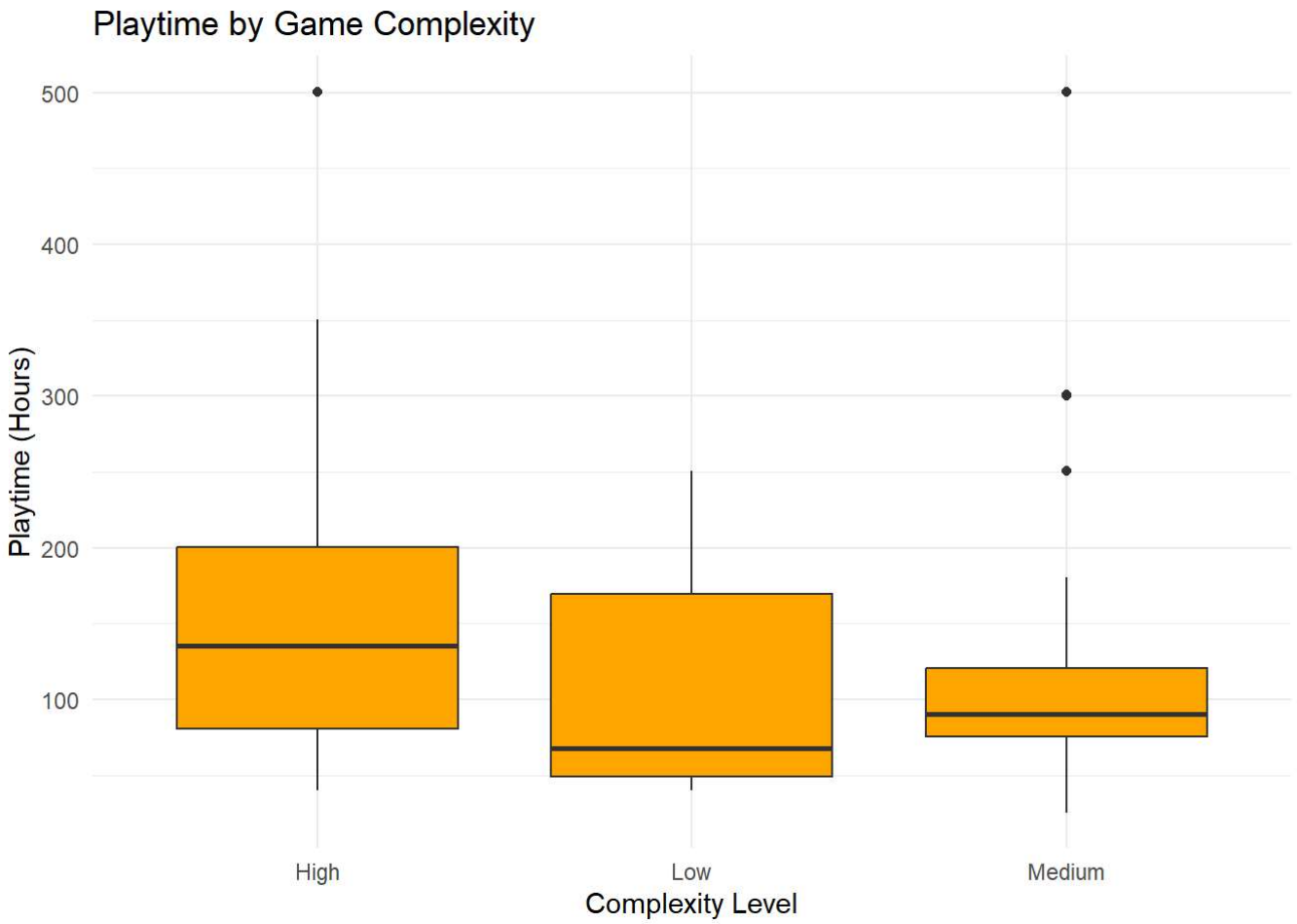

```
# Scatter Plot (Price vs Playtime)
plt_scatter_price_playtime <- ggplot(games_data, aes(x = price, y = playtime_hours, color = genr
e)) +
  geom_point(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Price vs Playtime by Genre", x = "Price ($)", y = "Playtime (Hours)")
print(plt_scatter_price_playtime)
```




```
# Violin Plot (Price by Multiplayer Support)
plt_violin_price_multiplayer <- ggplot(games_data, aes(x = multiplayer_support, y = price)) +
  geom_violin(fill = "lightblue") +
  theme_minimal() +
  labs(title = "Price Distribution by Multiplayer Support", x = "Multiplayer Support", y = "Pric
e ($)")
print(plt_violin_price_multiplayer)
```



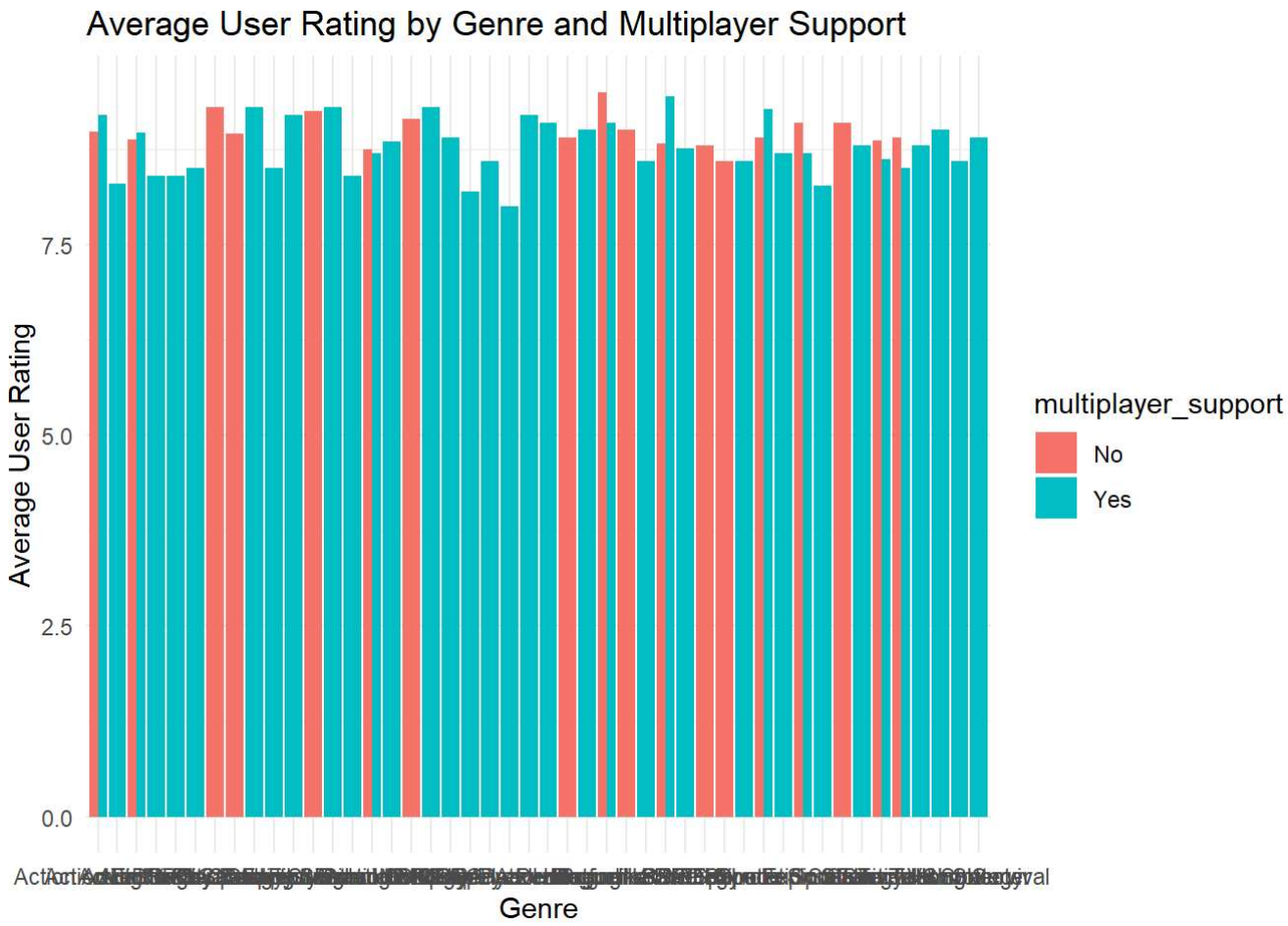

```
# Box Plot (Playtime by Complexity)
plt_box_playtime_complexity <- ggplot(games_data, aes(x = complexity_level, y = playtime_hours))
+
  geom_boxplot(fill = "orange") +
  theme_minimal() +
  labs(title = "Playtime by Game Complexity", x = "Complexity Level", y = "Playtime (Hours)")
print(plt_box_playtime_complexity)
```



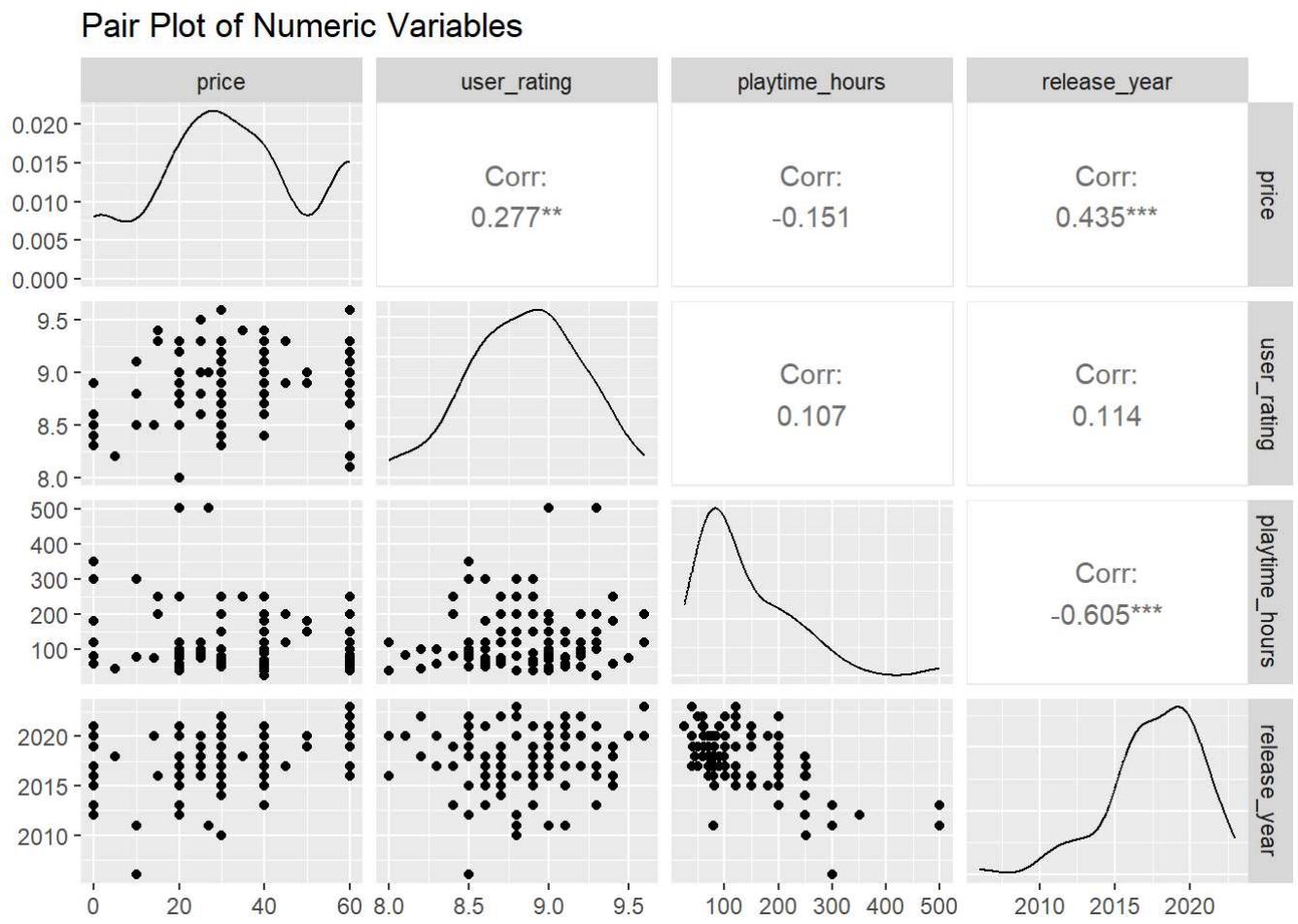

```
# 3. Multi-variable Plots
# Bar Plot (Average User Rating by Genre and Multiplayer Support)
plt_bar_multi <- games_data %>%
  group_by(genre, multiplayer_support) %>%
  summarise(avg_rating = mean(user_rating)) %>%
  ggplot(aes(x = genre, y = avg_rating, fill = multiplayer_support)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Average User Rating by Genre and Multiplayer Support",
       x = "Genre", y = "Average User Rating")
```

```
## `summarise()` has grouped output by 'genre'. You can override using the
## `.groups` argument.
```

```
print(plt_bar_multi)
```




```
# Pair Plot
plt_pair <- ggpairs(games_data[, c("price", "user_rating", "playtime_hours", "release_year")],
                    title = "Pair Plot of Numeric Variables")
print(plt_pair)
```


Pair Plot of Numeric Variables
price
user_rating
playtime_hours
release_year
Corr:
0.277**
Corr:
-0.151
Corr:
0.435***
Corr:
0.107
Corr:
0.114
Corr:
-0.605***
0.020
0.015
0.010
0.005
0.000
9.5
9.0
8.5
8.0
500
400
300
200
100
2020
2015
2010
0
20
40
60
8.0
8.5
9.0
9.5
100
200
300
400
500
2010
2015
2020

```
# Metadata and Inference Analysis Comment
#
# Metadata Overview:
# - Dataset: Video Game Dataset
# - Variables:
#   - Categorical: Genre, Multiplayer Support, Complexity Level
#   - Numerical: Price, Playtime Hours, User Rating, Release Year
#
# Key Insights:
# 1. Genre Distribution: Reveals market composition and game type prevalence
# 2. Multiplayer Support: Shows proportion of multiplayer vs single-player games
# 3. Price Distribution: Identifies pricing strategies across different game types
# 4. Playtime Analysis: Understand player engagement and game duration patterns
#
# Analytical Inferences:
# - Market Composition: Understand genre trends and market dynamics
# - Player Behavior: Analyze engagement, complexity, and value perception
# - Pricing Strategies: Compare pricing across genres and game types
#
# Recommendations:
# - For Developers: Focus on high-engagement genres and features
# - For Analysts: Track market trends and player preferences
#
# tokens while maintaining its helpfulness, quality, completeness, and accuracy.
# It focuses on addressing the specific query or task at hand, avoiding tangential
# information unless helpful for understanding or completing the request.
```