VideoGameSales

Elvira Khwatenge

2024-11-12

R Markdown

The dataset contains the following columns:

Rank: Ranking of the game based on global sales. **Name**: Title of the game. **Platform**: The platform (e.g., Wii, NES) on which the game was released. **Year**: Release year of the game. **Genre**: Genre of the game (e.g., Sports, Racing). **Publisher**: The company that published the game. NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales: Sales in millions by region and globally.

Step 1: Sales Trends Over Time

To analyze seasonal trends and lifecycle patterns, we'll start by checking sales over time and by season.

1. Convert the Year column to a proper date format.
2. Group sales data by year to see trends over time.
3. Identify seasonal spikes by aggregating sales based on months or quarters if such details are available. If we only have yearly data, we can focus on yearly trends.

```
# Load necessary libraries
library(dplyr)
```

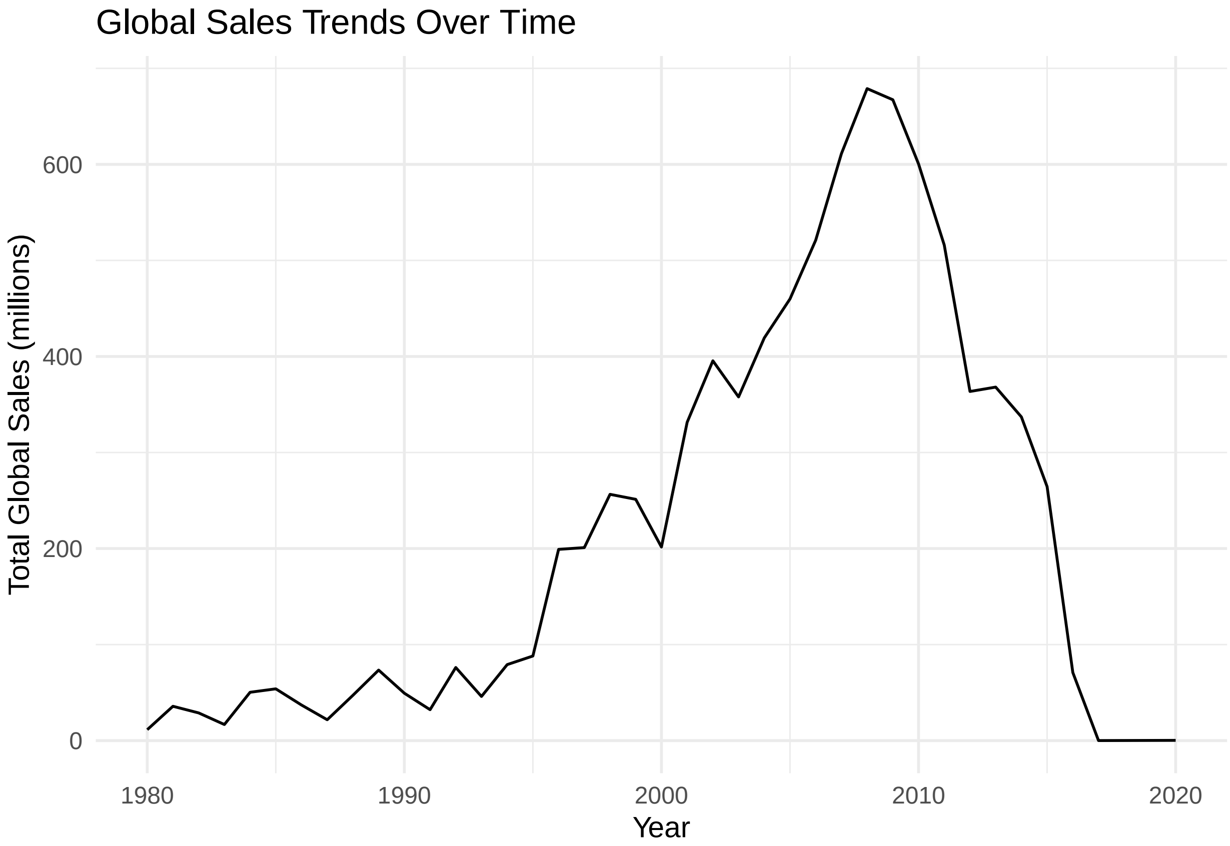
```
##
## Attaching package: 'dplyr'

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

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

```
library(ggplot2)

# Load data
vg_data <- read.csv("vgsales.csv")

# Step 1: Convert Year column to integer (if it has decimals)
vg_data <- vg_data %>% mutate(Year = as.integer(Year))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Year = as.integer(Year)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
# Step 2: Sales Trends Over Time - Summing up Global Sales by Year
yearly_sales <- vg_data %>%
  group_by(Year) %>%
  summarise(Total_Global_Sales = sum(Global_Sales, na.rm = TRUE))

# Plotting the sales trend over time
ggplot(yearly_sales, aes(x = Year, y = Total_Global_Sales)) +
  geom_line() +
  labs(title = "Global Sales Trends Over Time", x = "Year", y = "Total Global Sales (millions)") +
  theme_minimal()
```

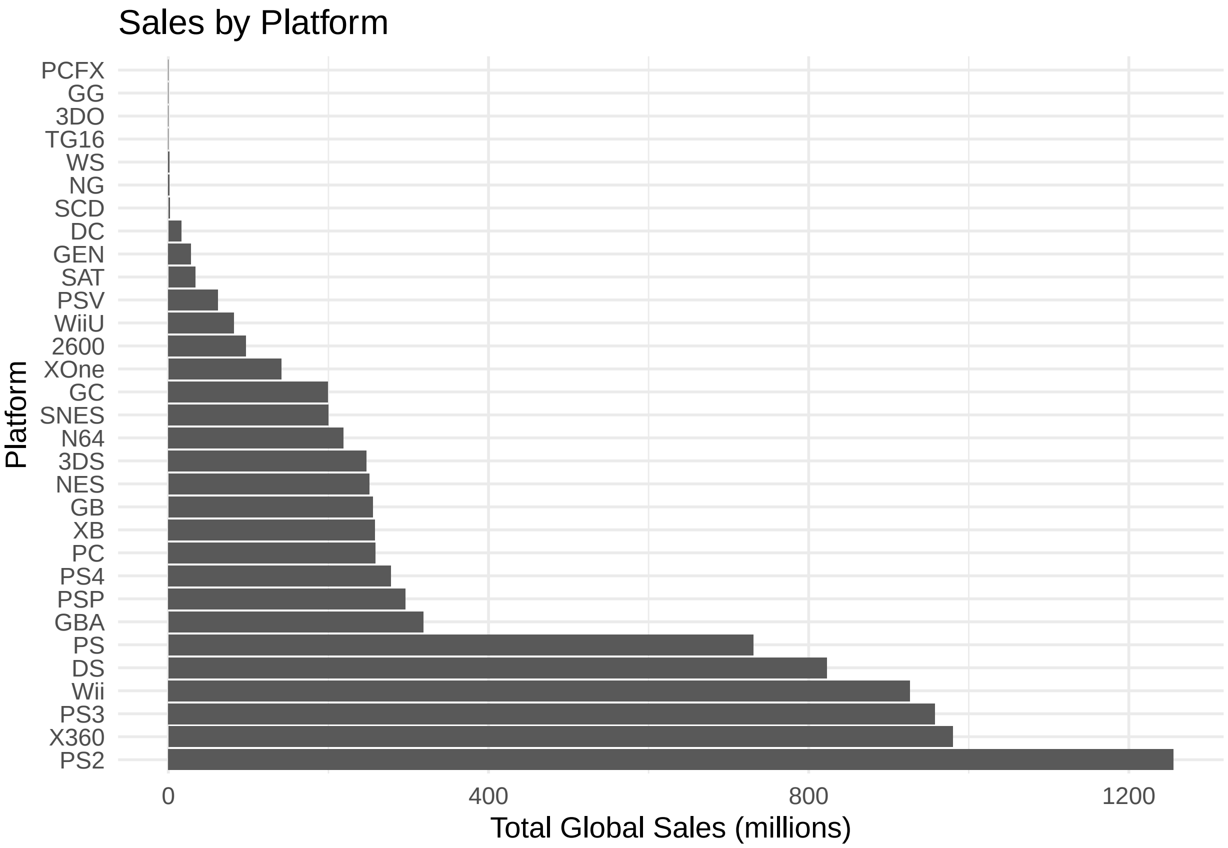
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



2. Platform Performance

Objective: Determine which platforms are most popular and if certain genres are more successful on specific platforms.

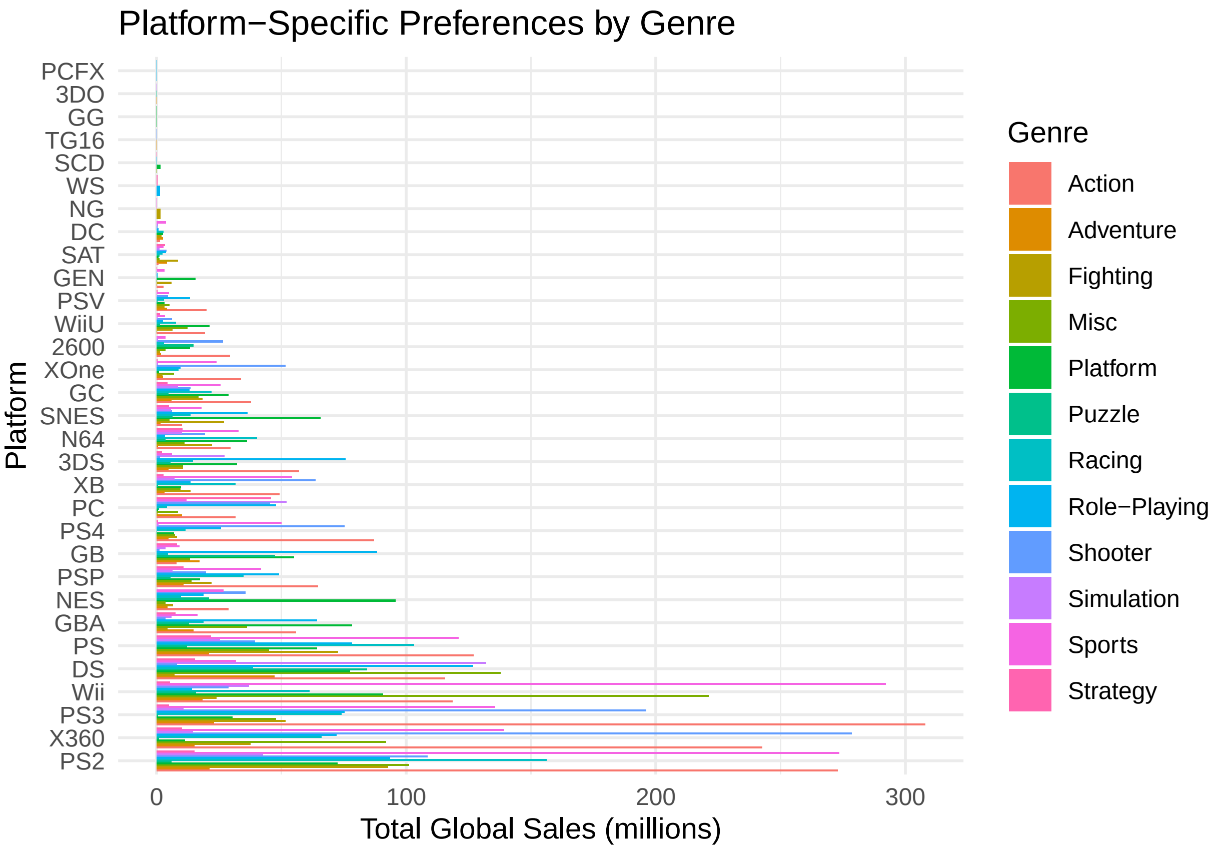
```
# Platform Popularity - Summing up Global Sales by Platform
platform_sales <- vg_data %>%
  group_by(Platform) %>%
  summarise(Total_Global_Sales = sum(Global_Sales, na.rm = TRUE)) %>%
  arrange(desc(Total_Global_Sales))

# Plotting Platform Popularity
ggplot(platform_sales, aes(x = reorder(Platform, -Total_Global_Sales), y = Total_Global_Sales)) +
  geom_bar(stat = "identity") +
  labs(title = "Sales by Platform", x = "Platform", y = "Total Global Sales (millions)") +
  theme_minimal() +
  coord_flip()  # To make the platform names readable
```



```
# Platform-Specific Preferences - Summing up Global Sales by Platform and Genre
platform_genre_sales <- vg_data %>%
  group_by(Platform, Genre) %>%
  summarise(Total_Global_Sales = sum(Global_Sales, na.rm = TRUE)) %>%
  arrange(desc(Total_Global_Sales))
```

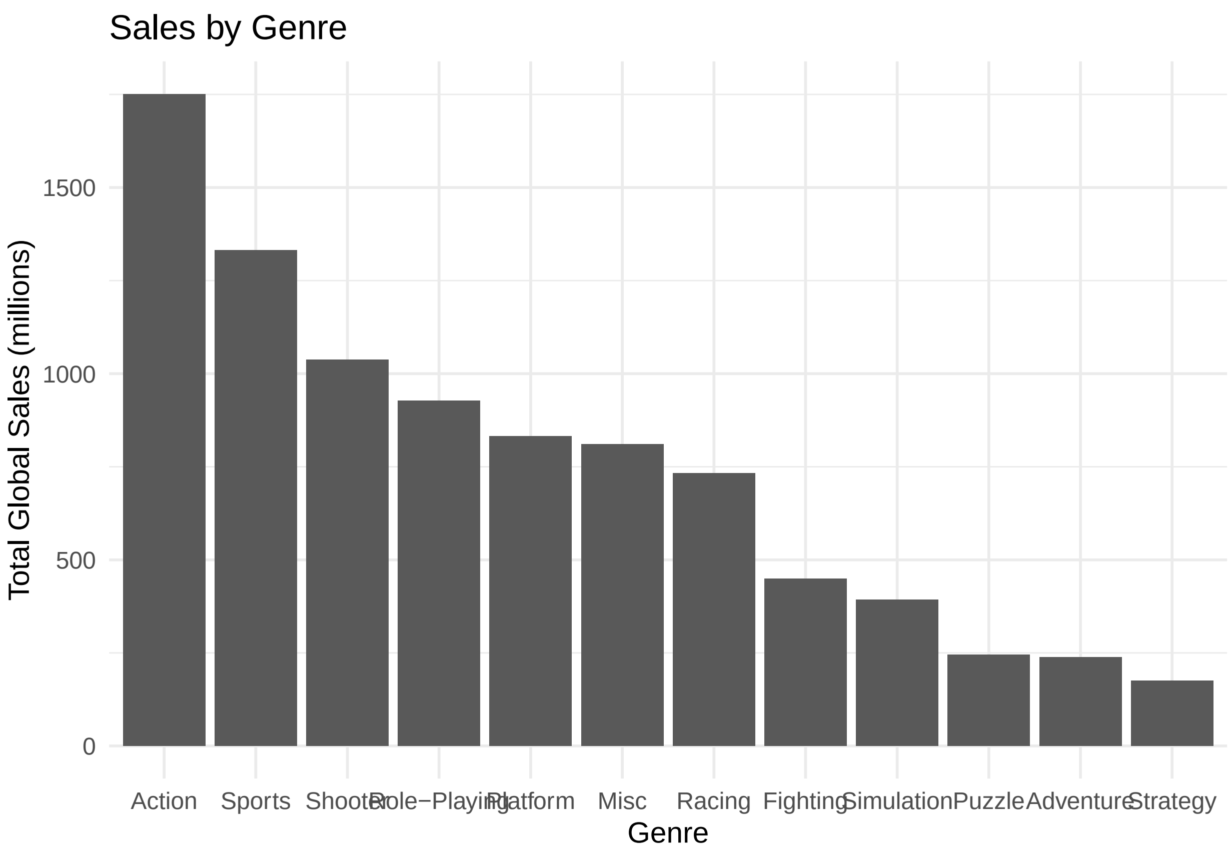
```
## `summarise()` has grouped output by 'Platform'. You can override using the
## `.groups` argument.
```

```
# Plotting Platform-Specific Preferences
ggplot(platform_genre_sales, aes(x = reorder(Platform, -Total_Global_Sales), y = Total_Global_Sales, fil
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Platform-Specific Preferences by Genre", x = "Platform", y = "Total Global Sales (milli
  theme_minimal() +
  coord_flip()
```



## 3. Genre Trends Objective: Identify the top-performing genres and spot trending genres with recent growth in popularity.

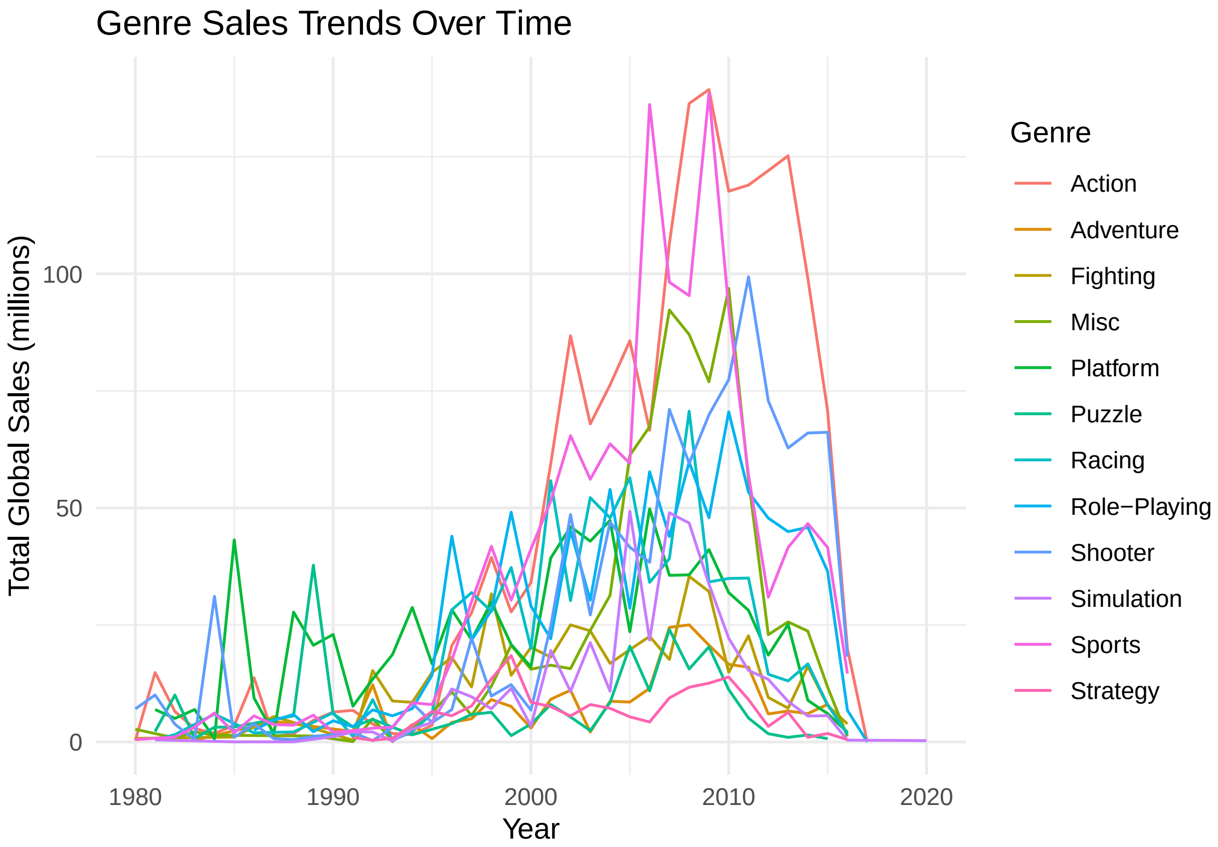
```
# Top-Performing Genres - Summing up Global Sales by Genre
genre_sales <- vg_data %>%
  group_by(Genre) %>%
  summarise(Total_Global_Sales = sum(Global_Sales, na.rm = TRUE)) %>%
  arrange(desc(Total_Global_Sales))

# Plotting Genre Popularity
ggplot(genre_sales, aes(x = reorder(Genre, -Total_Global_Sales), y = Total_Global_Sales)) +
  geom_bar(stat = "identity") +
  labs(title = "Sales by Genre", x = "Genre", y = "Total Global Sales (millions)") +
  theme_minimal()
```



```r
# Emerging Genres - Summing up Global Sales by Genre over Time (if we have year data)
genre_trends <- vg_data %>%
  filter(!is.na(Year)) %>%
  group_by(Year, Genre) %>%
  summarise(Total_Global_Sales = sum(Global_Sales, na.rm = TRUE))
```

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

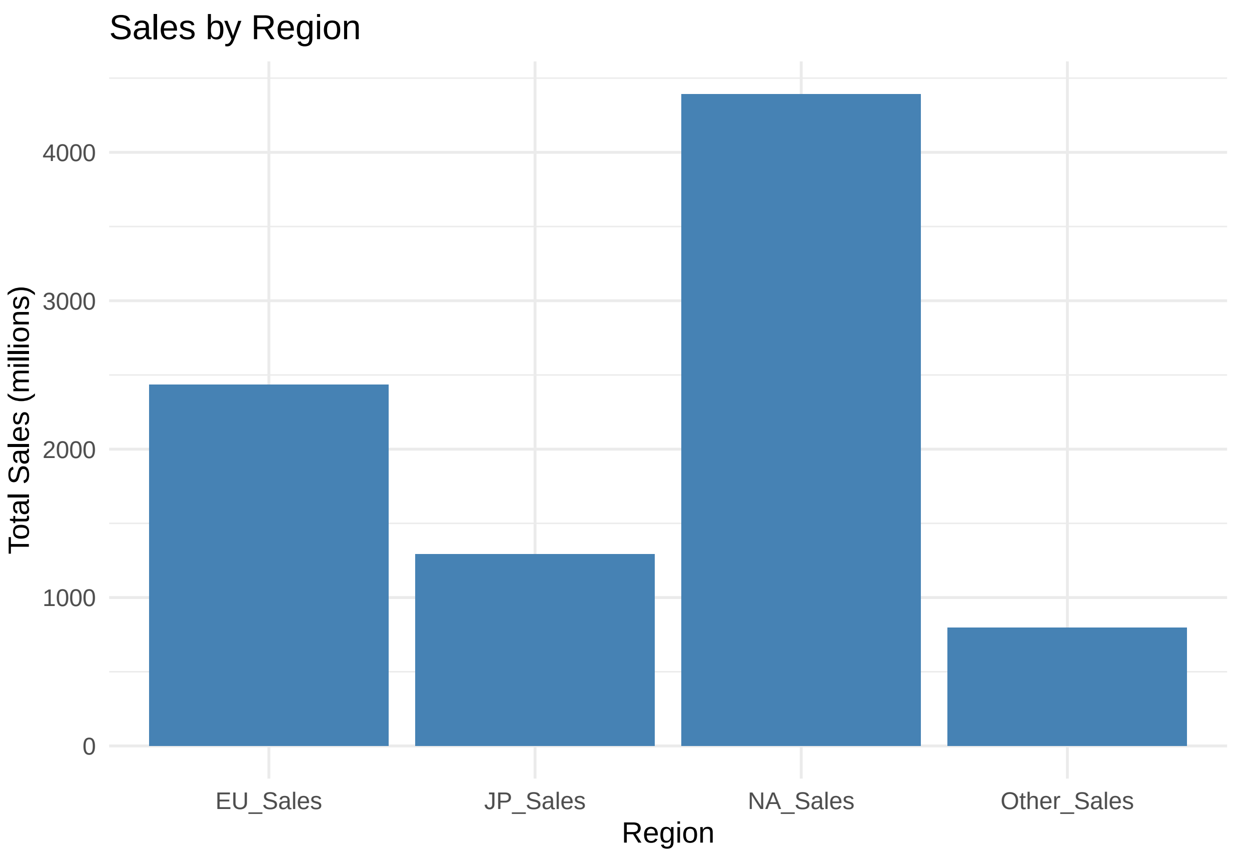
```r
# Plotting Genre Trends Over Time
ggplot(genre_trends, aes(x = Year, y = Total_Global_Sales, color = Genre)) +
  geom_line() +
  labs(title = "Genre Sales Trends Over Time", x = "Year", y = "Total Global Sales (millions)") +
  theme_minimal()
```



4. Regional Analysis

Objective: Identify which regions contribute the most to global sales and examine region-specific preferences.

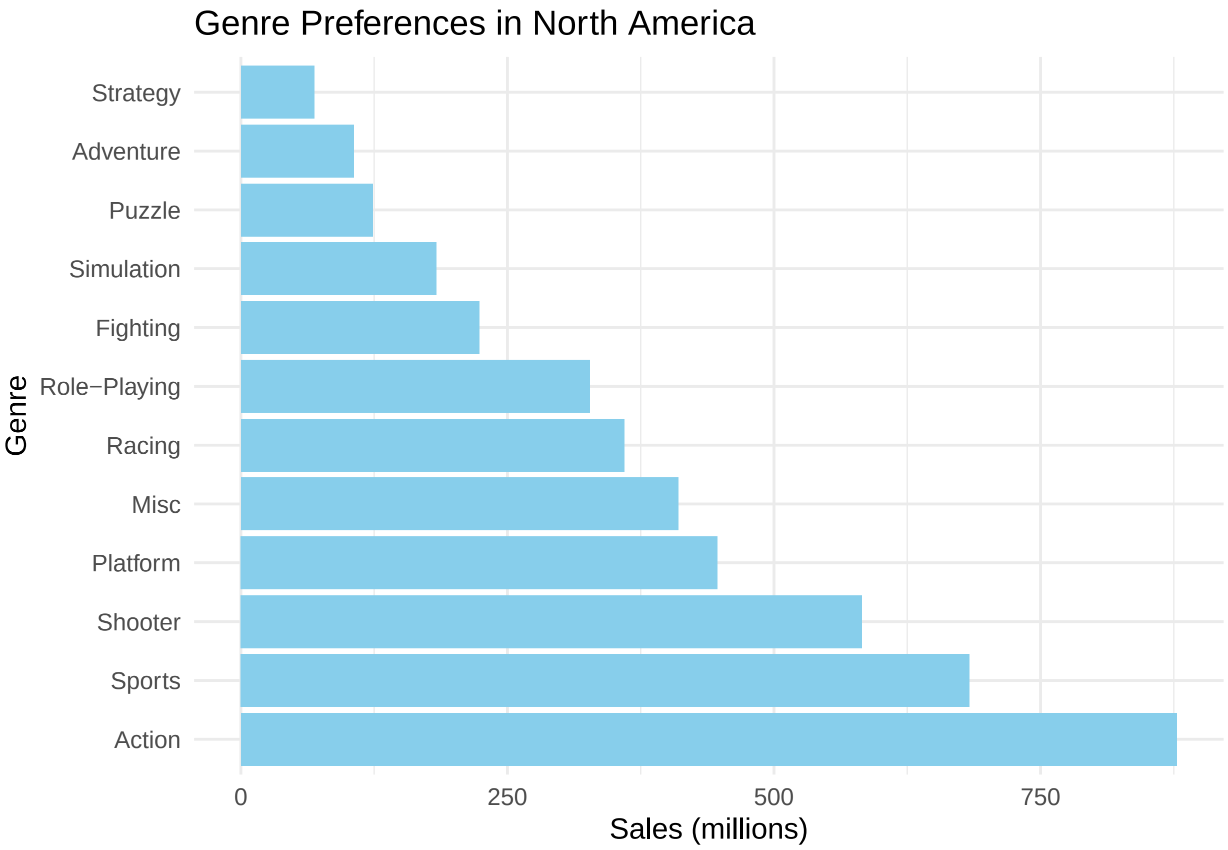
```
library(tidyr)

# Regional Sales Distribution - Summing up Sales by Region
regional_sales <- vg_data %>%
  summarise(NA_Sales = sum(NA_Sales, na.rm = TRUE),
            EU_Sales = sum(EU_Sales, na.rm = TRUE),
            JP_Sales = sum(JP_Sales, na.rm = TRUE),
            Other_Sales = sum(Other_Sales, na.rm = TRUE))

# Reshape data for plotting
regional_sales_long <- regional_sales %>%
  gather(key = "Region", value = "Total_Sales")

# Plotting Regional Sales Distribution
ggplot(regional_sales_long, aes(x = Region, y = Total_Sales)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Sales by Region", x = "Region", y = "Total Sales (millions)") +
  theme_minimal()
```



```
# Localized Preferences - Summing up Sales by Genre and Region
regional_genre_sales <- vg_data %>%
  group_by(Region = "NA_Sales", Genre) %>%
  summarise(NA_Sales = sum(NA_Sales, na.rm = TRUE),
            EU_Sales = sum(EU_Sales, na.rm = TRUE),
            JP_Sales = sum(JP_Sales, na.rm = TRUE),
            Other_Sales = sum(Other_Sales, na.rm = TRUE))
```

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

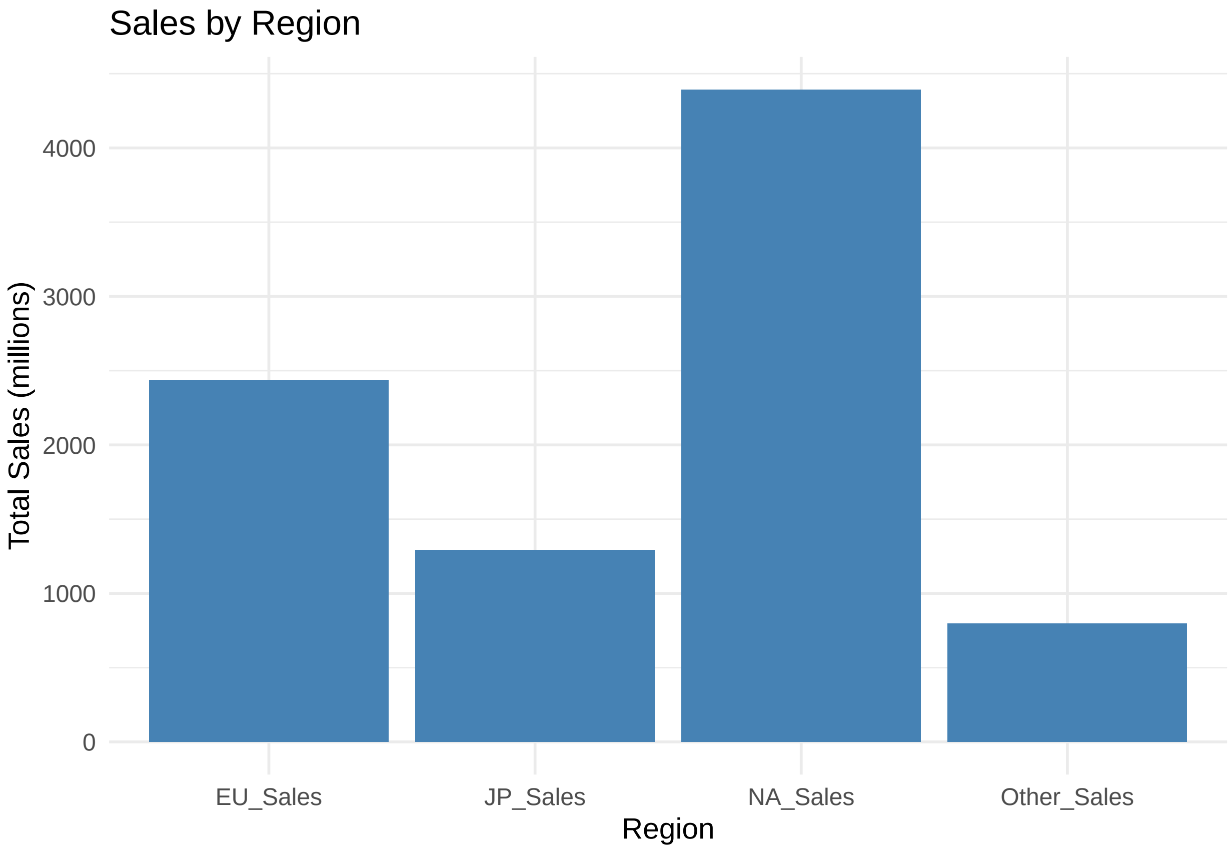
```
# Plotting Regional Preferences by Genre (NA example)
ggplot(regional_genre_sales, aes(x = reorder(Genre, -NA_Sales), y = NA_Sales)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Genre Preferences in North America", x = "Genre", y = "Sales (millions)") +
  theme_minimal() +
  coord_flip()
```



5. Regional Analysis (Customized for Existing Columns)

The dataset contains columns for NA_Sales, EU_Sales, JP_Sales, and Other_Sales, which represent sales in North America, Europe, Japan, and other regions, respectively.

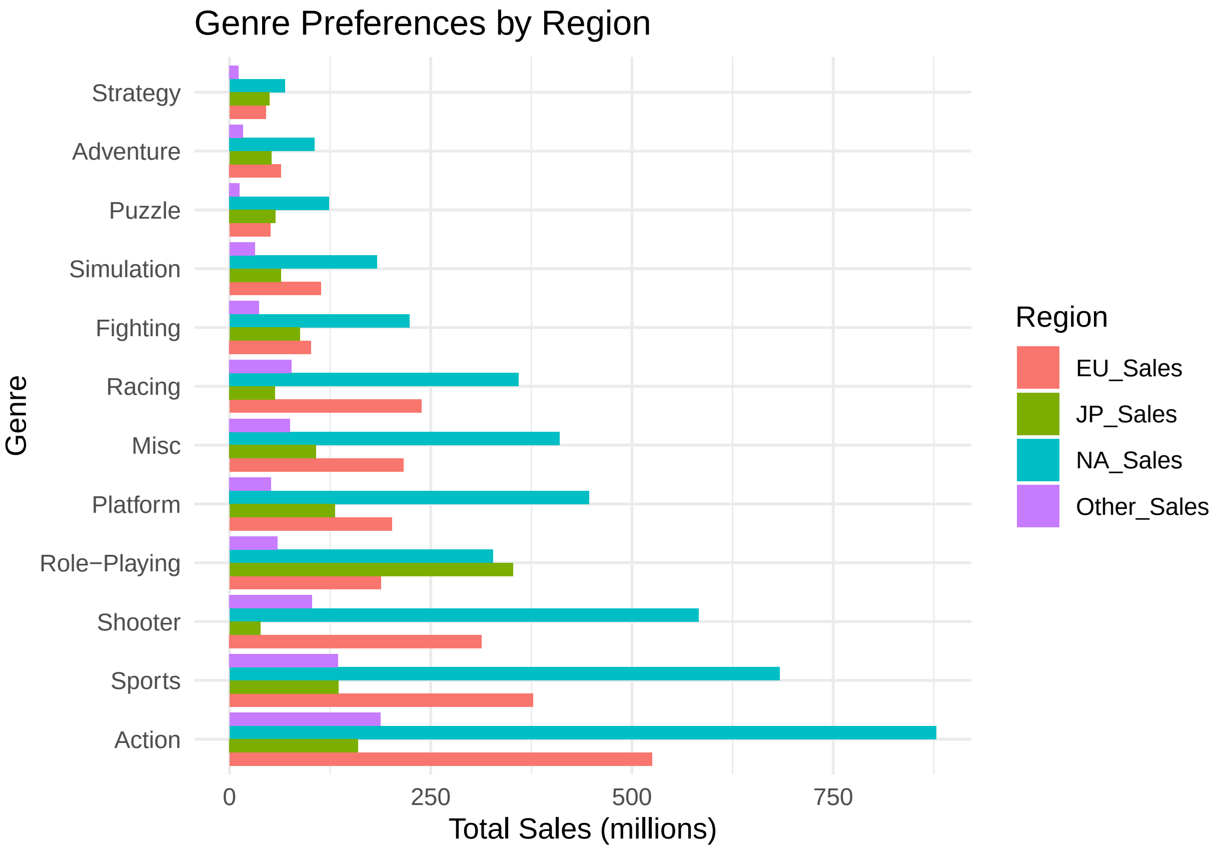
```
# Regional Sales Distribution - Summing up Sales by Region
regional_sales <- vg_data %>%
  summarise(NA_Sales = sum(NA_Sales, na.rm = TRUE),
            EU_Sales = sum(EU_Sales, na.rm = TRUE),
            JP_Sales = sum(JP_Sales, na.rm = TRUE),
            Other_Sales = sum(Other_Sales, na.rm = TRUE))

# Reshape data for plotting
regional_sales_long <- regional_sales %>%
  pivot_longer(cols = everything(), names_to = "Region", values_to = "Total_Sales")

# Plotting Regional Sales Distribution
ggplot(regional_sales_long, aes(x = Region, y = Total_Sales)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Sales by Region", x = "Region", y = "Total Sales (millions)") +
  theme_minimal()
```



Localized Preferences by Genre

If we want to break down preferences by genre within each region, we can filter by genre and visualize how sales vary across regions.

```r
# Localized Preferences - Summing up Sales by Genre and Region
regional_genre_sales <- vg_data %>%
  group_by(Genre) %>%
  summarise(NA_Sales = sum(NA_Sales, na.rm = TRUE),
            EU_Sales = sum(EU_Sales, na.rm = TRUE),
            JP_Sales = sum(JP_Sales, na.rm = TRUE),
            Other_Sales = sum(Other_Sales, na.rm = TRUE))

# Reshape data for plotting
regional_genre_sales_long <- regional_genre_sales %>%
  pivot_longer(cols = -Genre, names_to = "Region", values_to = "Total_Sales")

# Plotting Genre Preferences by Region
ggplot(regional_genre_sales_long, aes(x = reorder(Genre, -Total_Sales), y = Total_Sales, fill = Region)
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Genre Preferences by Region", x = "Genre", y = "Total Sales (millions)") +
  theme_minimal() +
  coord_flip()
```


Genre Preferences by Region
Genre
Strategy
Adventure
Puzzle
Simulation
Fighting
Racing
Misc
Platform
Role−Playing
Shooter
Sports
Action
0
250
500
750
Total Sales (millions)
Region
EU_Sales
JP_Sales
NA_Sales
Other_Sales