TidyVerse CREATE Assignment

Jian Quan Chen

2023-03-27

Introduction

For this assignment, I will be creating a programming sample vignette to demonstrate the use of the `tidyr` package in the tidyverse package. I will be working with the "Video Game Sales" (https://www.kaggle.com/datasets/gregorut/videogamesales) dataset from Kaggle. The dataset was generated from a scrape of vgchartz.com and contains the sales of video games that sold greater than 100,000 copies from 1980 to 2020.

The `tidyr` package provides a set of functions that help tidy data, an important step in the data wrangling process. Ideally, in a tidy data set, each column should correspond to a single variable, each row should represent a single observation, and each cell should contains a single value.

In the "Video Game Sales" dataset, the sales (in millions) are presented in a wide format in which the sales of countries are split into multiple columns. In order to tidy this data, I will be using the `pivot_longer` function from the `tidyr` package to reshape these columns into one single column. Then, analyze the data to identify which region and genre had the most video game sales.

Code

Importing Library

```
library(tidyverse)
```

Importing the Dataset

```
video_games_df <- read.csv("https://raw.githubusercontent.com/LeJQC/MSDS/main/DATA%20607/TidyVerse%20CRE
glimpse(video_games_df)
```

```
## Rows: 16,598
## Columns: 11
## $ Rank         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
## $ Name         <chr> "Wii Sports", "Super Mario Bros.", "Mario Kart Wii", "Wii~
## $ Platform     <chr> "Wii", "NES", "Wii", "Wii", "GB", "GB", "DS", "Wii", "Wii~
## $ Year         <chr> "2006", "1985", "2008", "2009", "1996", "1989", "2006", "~
## $ Genre        <chr> "Sports", "Platform", "Racing", "Sports", "Role-Playing",~
## $ Publisher    <chr> "Nintendo", "Nintendo", "Nintendo", "Nintendo", "Nintendo~
## $ NA_Sales     <dbl> 41.49, 29.08, 15.85, 15.75, 11.27, 23.20, 11.38, 14.03, 1~
## $ EU_Sales     <dbl> 29.02, 3.58, 12.88, 11.01, 8.89, 2.26, 9.23, 9.20, 7.06, ~
## $ JP_Sales     <dbl> 3.77, 6.81, 3.79, 3.28, 10.22, 4.22, 6.50, 2.93, 4.70, 0.~
## $ Other_Sales  <dbl> 8.46, 0.77, 3.31, 2.96, 1.00, 0.58, 2.90, 2.85, 2.26, 0.4~
## $ Global_Sales <dbl> 82.74, 40.24, 35.82, 33.00, 31.37, 30.26, 30.01, 29.02, 2~
```

Reshaping Data Frame to Long Format

The sales from each region can be combined into one single column using `pivot_longer`. This function takes several arguments including:

- `data` : wide-format data frame to pivot
- `cols`: columns in the data frame that you want to pivot
- `names_to`: name of the column that is being created
- `values_to`: name of the column where the cell values are stored

There are more arguments to further manipulate the data frame but these are the most essential.

```
sales_df <- video_games_df %>%
  pivot_longer(
    cols = NA_Sales:Other_Sales,
    names_to = "Region",
    names_pattern = "(.*)_[A-Za-z]*",
    values_to = "Sales"
  )
glimpse(sales_df)
```

```
## Rows: 66,392
## Columns: 9
## $ Rank         <int> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, ~
## $ Name         <chr> "Wii Sports", "Wii Sports", "Wii Sports", "Wii Sports", "~
## $ Platform     <chr> "Wii", "Wii", "Wii", "Wii", "NES", "NES", "NES", "NES", "~
## $ Year         <chr> "2006", "2006", "2006", "2006", "1985", "1985", "1985", "~
## $ Genre        <chr> "Sports", "Sports", "Sports", "Sports", "Platform", "Plat~
## $ Publisher    <chr> "Nintendo", "Nintendo", "Nintendo", "Nintendo", "Nintendo~
## $ Global_Sales <dbl> 82.74, 82.74, 82.74, 82.74, 40.24, 40.24, 40.24, 40.24, 3~
## $ Region       <chr> "NA", "EU", "JP", "Other", "NA", "EU", "JP", "Other", "NA~
## $ Sales        <dbl> 41.49, 29.02, 3.77, 8.46, 29.08, 3.58, 6.81, 0.77, 15.85,~
```

Analyzing the Sales Data

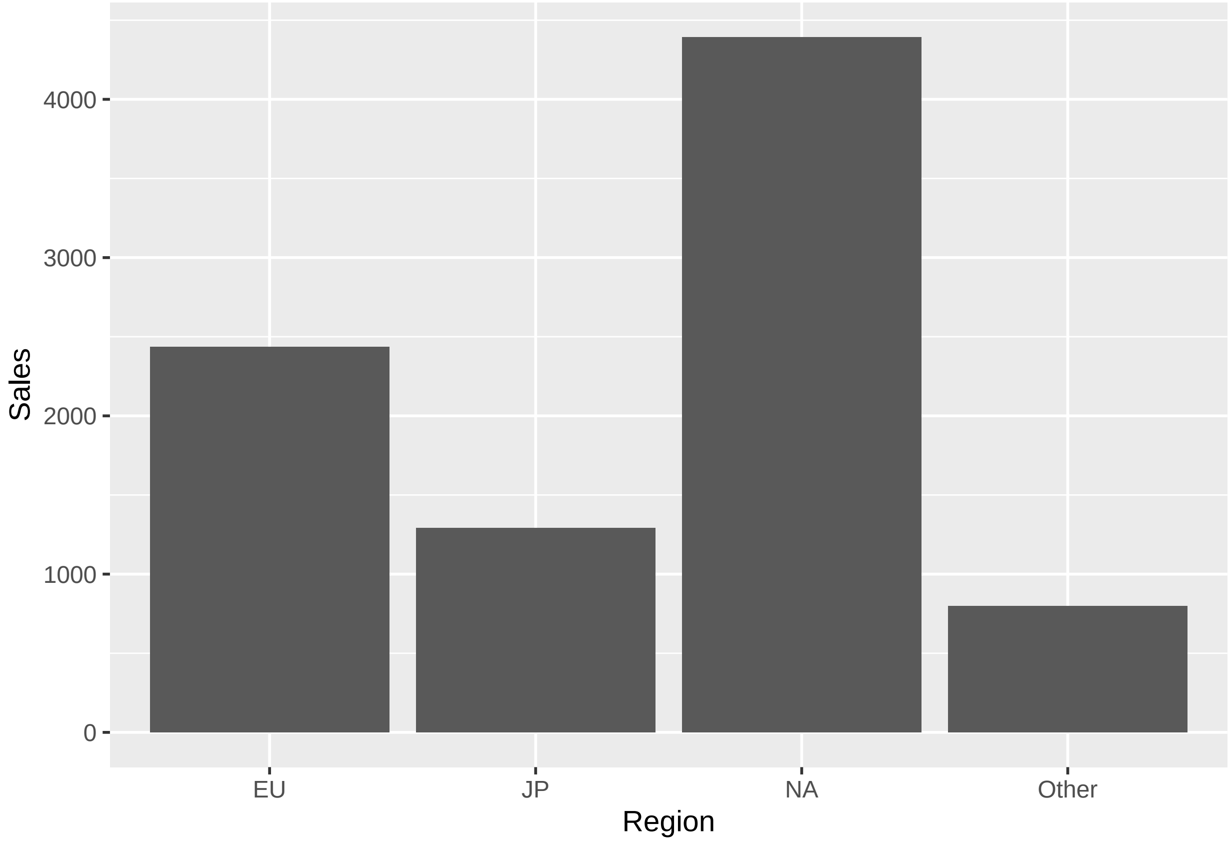
Sum of Sales by Region

```
sales_df %>%
  group_by(Region) %>%
  summarize(total_sales = sum(Sales)) %>%
  arrange(desc(total_sales))
```

```
## # A tibble: 4 x 2
##   Region total_sales
##   <chr>         <dbl>
## 1 NA            4393.
## 2 EU            2434.
## 3 JP            1291.
## 4 Other          798.
```

Plotting Sales by Region

```
sales_df %>%
  ggplot(aes(x=Region, y= Sales))+
  geom_bar(stat = "identity")
```



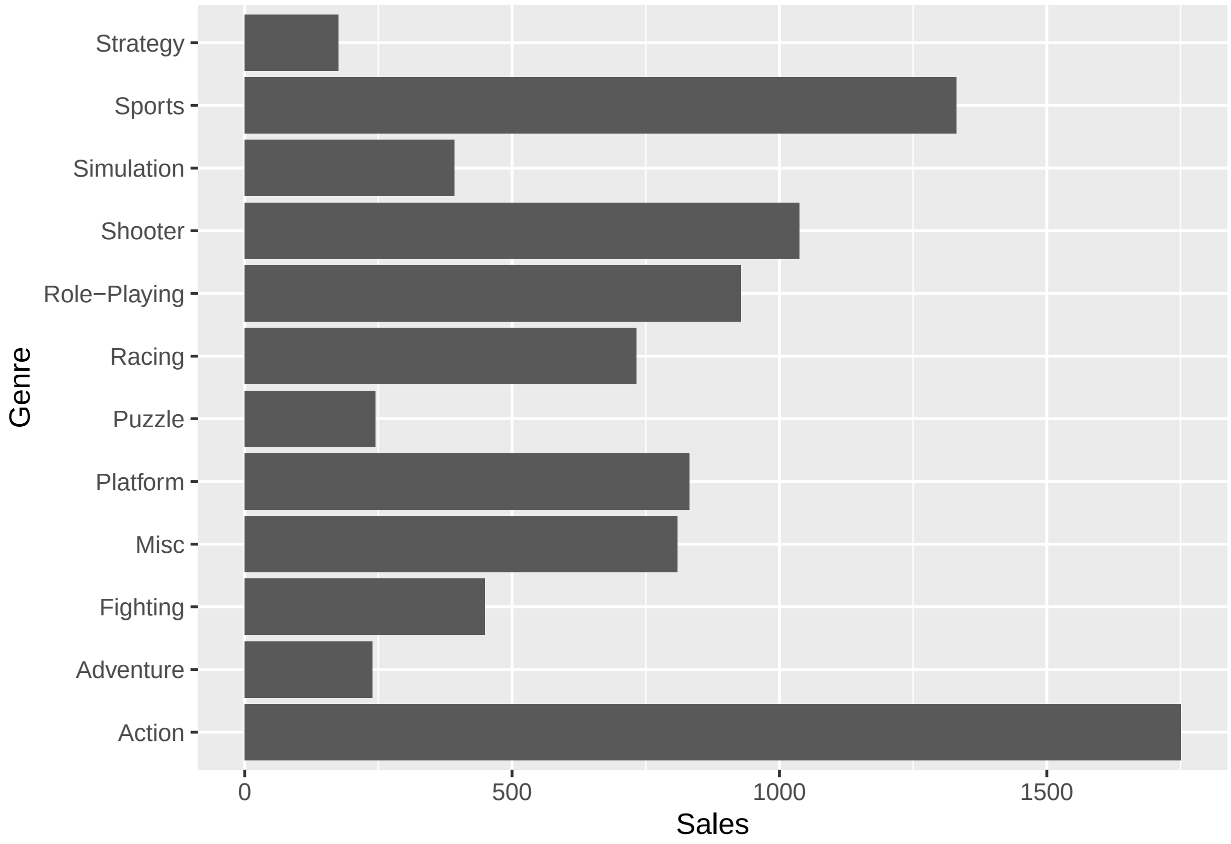
Most Popular Genres

```
sales_df %>%
  group_by(Genre) %>%
  summarise(count = n()) %>%
  mutate(percent = round(count/sum(count)*100)) %>%
  arrange(desc(count))
```

```
## # A tibble: 12 x 3
##    Genre        count percent
##    <chr>        <int>   <dbl>
##  1 Action       13264      20
##  2 Sports        9384      14
##  3 Misc          6956      10
##  4 Role-Playing  5952       9
##  5 Shooter       5240       8
##  6 Adventure     5144       8
##  7 Racing        4996       8
##  8 Platform      3544       5
##  9 Simulation    3468       5
## 10 Fighting      3392       5
## 11 Strategy      2724       4
## 12 Puzzle        2328       4
```

Sales by Genre

```
sales_df %>%
  group_by(Genre) %>%
  ggplot((aes(x = Genre,y = Sales)))+
  geom_bar(stat = "identity")+
  coord_flip()
```



Reshaping Data Frame back to Wide Format

Sometimes, the wide format of a dataset presents a better visualization of the data, which can make it easier to understand. For that, there is a `pivot_wider` function. This function is the inverse of `pivot_longer` and converts one column into multiple columns.

```
# Has the same amount of variables and observations as the starting data frame
sales_wide <- sales_df %>%
  pivot_wider(
    names_from = "Region",
    values_from = "Sales")
glimpse(sales_wide)
```

```
## Rows: 16,598
## Columns: 11
## $ Rank         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
## $ Name         <chr> "Wii Sports", "Super Mario Bros.", "Mario Kart Wii", "Wii~
## $ Platform     <chr> "Wii", "NES", "Wii", "Wii", "GB", "GB", "DS", "Wii", "Wii~
## $ Year         <chr> "2006", "1985", "2008", "2009", "1996", "1989", "2006", "~
## $ Genre        <chr> "Sports", "Platform", "Racing", "Sports", "Role-Playing",~
## $ Publisher    <chr> "Nintendo", "Nintendo", "Nintendo", "Nintendo", "Nintendo~
## $ Global_Sales <dbl> 82.74, 40.24, 35.82, 33.00, 31.37, 30.26, 30.01, 29.02, 2~
## $ ‘NA‘         <dbl> 41.49, 29.08, 15.85, 15.75, 11.27, 23.20, 11.38, 14.03, 1~
## $ EU           <dbl> 29.02, 3.58, 12.88, 11.01, 8.89, 2.26, 9.23, 9.20, 7.06, ~
## $ JP           <dbl> 3.77, 6.81, 3.79, 3.28, 10.22, 4.22, 6.50, 2.93, 4.70, 0.~
## $ Other        <dbl> 8.46, 0.77, 3.31, 2.96, 1.00, 0.58, 2.90, 2.85, 2.26, 0.4~
```

```
glimpse(video_games_df)
```

```
## Rows: 16,598
## Columns: 11
## $ Rank         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
## $ Name         <chr> "Wii Sports", "Super Mario Bros.", "Mario Kart Wii", "Wii~
## $ Platform     <chr> "Wii", "NES", "Wii", "Wii", "GB", "GB", "DS", "Wii", "Wii~
## $ Year         <chr> "2006", "1985", "2008", "2009", "1996", "1989", "2006", "~
## $ Genre        <chr> "Sports", "Platform", "Racing", "Sports", "Role-Playing",~
## $ Publisher    <chr> "Nintendo", "Nintendo", "Nintendo", "Nintendo", "Nintendo~
## $ NA_Sales     <dbl> 41.49, 29.08, 15.85, 15.75, 11.27, 23.20, 11.38, 14.03, 1~
## $ EU_Sales     <dbl> 29.02, 3.58, 12.88, 11.01, 8.89, 2.26, 9.23, 9.20, 7.06, ~
## $ JP_Sales     <dbl> 3.77, 6.81, 3.79, 3.28, 10.22, 4.22, 6.50, 2.93, 4.70, 0.~
## $ Other_Sales  <dbl> 8.46, 0.77, 3.31, 2.96, 1.00, 0.58, 2.90, 2.85, 2.26, 0.4~
## $ Global_Sales <dbl> 82.74, 40.24, 35.82, 33.00, 31.37, 30.26, 30.01, 29.02, 2~
```

Conclusion

By pivoting the sales columns to a long format, I was able to easily analyze which region and which genre had the most sales. North America has more video game sales compared to Europe, Japan, and other countries. As for genre, action and sports games were the most popular games sold.