Video Game Sales

Justin Byun

1/6/2022

Introduction

Video games were an entertainment source that were first created in the 1950's and 1960's. However, it was only in the 1970's during which the first consumer-ready video game hardware was ready for release, the console known as the "Magnovox Odyssey" and the first video games, Pong and Computer Space. Since then, video games have exploded in popularity, bringing in a total worldwide revenue of 175.8 billion dollars in 2021 when considering PC, console, and mobile games together.

Objective

The dataset we will be looking at in this project will contain data on video games with sales specifically greater than 100,000 copies. Using this dataset, we will answer the following questions below:

1. Which video game genres sell the most, across all platforms and regions? Do certain regions like different genres more or less?
2. In each region, which publishers have the greatest sales? Which publishers have the greatest sales worldwide?
3. During which years did video games have the greatest sales? (Provide a visualization to graphically represent the change in sales over the years)
4. For each decade present in the data set, what platforms had the most video game sales? What was the best-selling game of each decade?

Loading in packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
```

Loading in the data

```
vgSalesData <- read_csv('vgsales.csv')
```

```
## Rows: 16598 Columns: 11

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Platform, Year, Genre, Publisher
## dbl (6): Rank, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales

##
## 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.
```

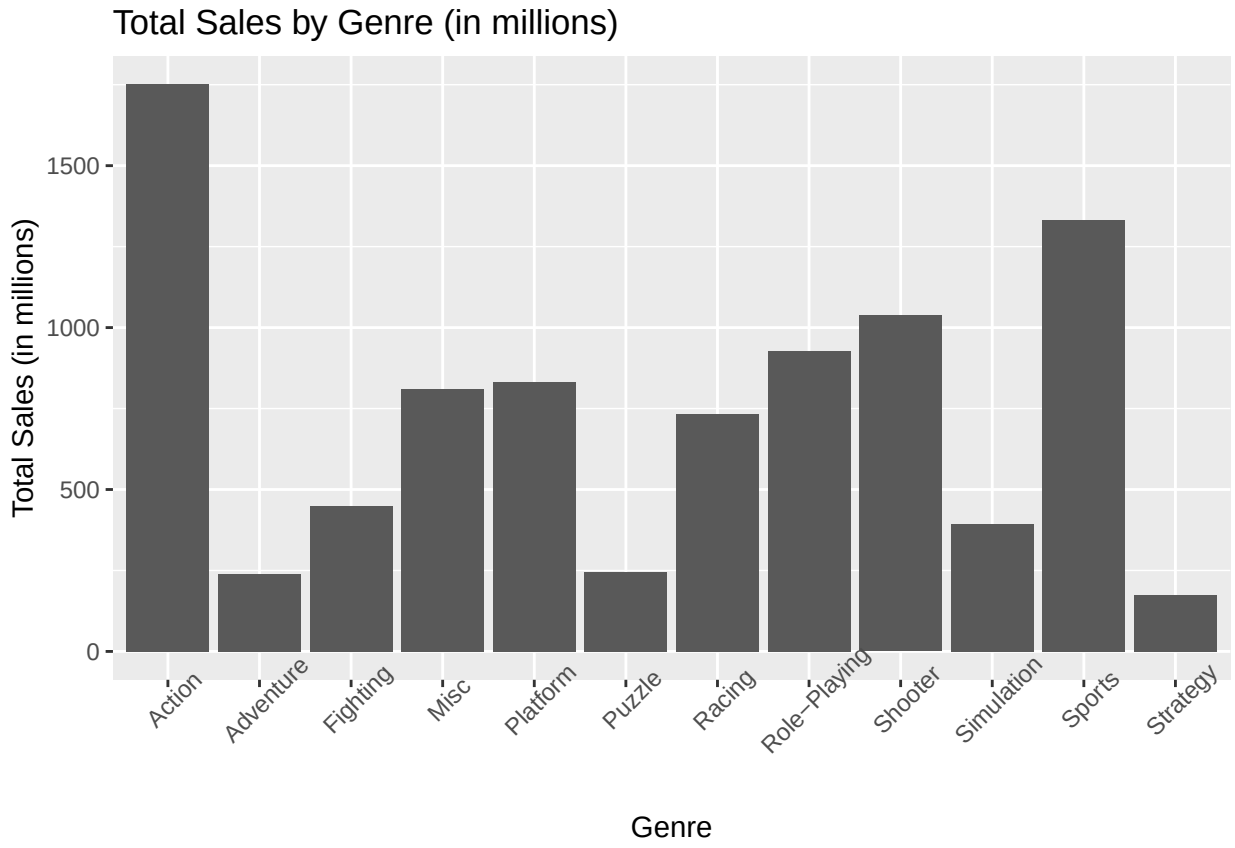
Taking a look into dimensions of dataset and data types of variables

```
glimpse(vgSalesData)
```

```
## Rows: 16,598
## Columns: 11
## $ Rank         <dbl> 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~
```

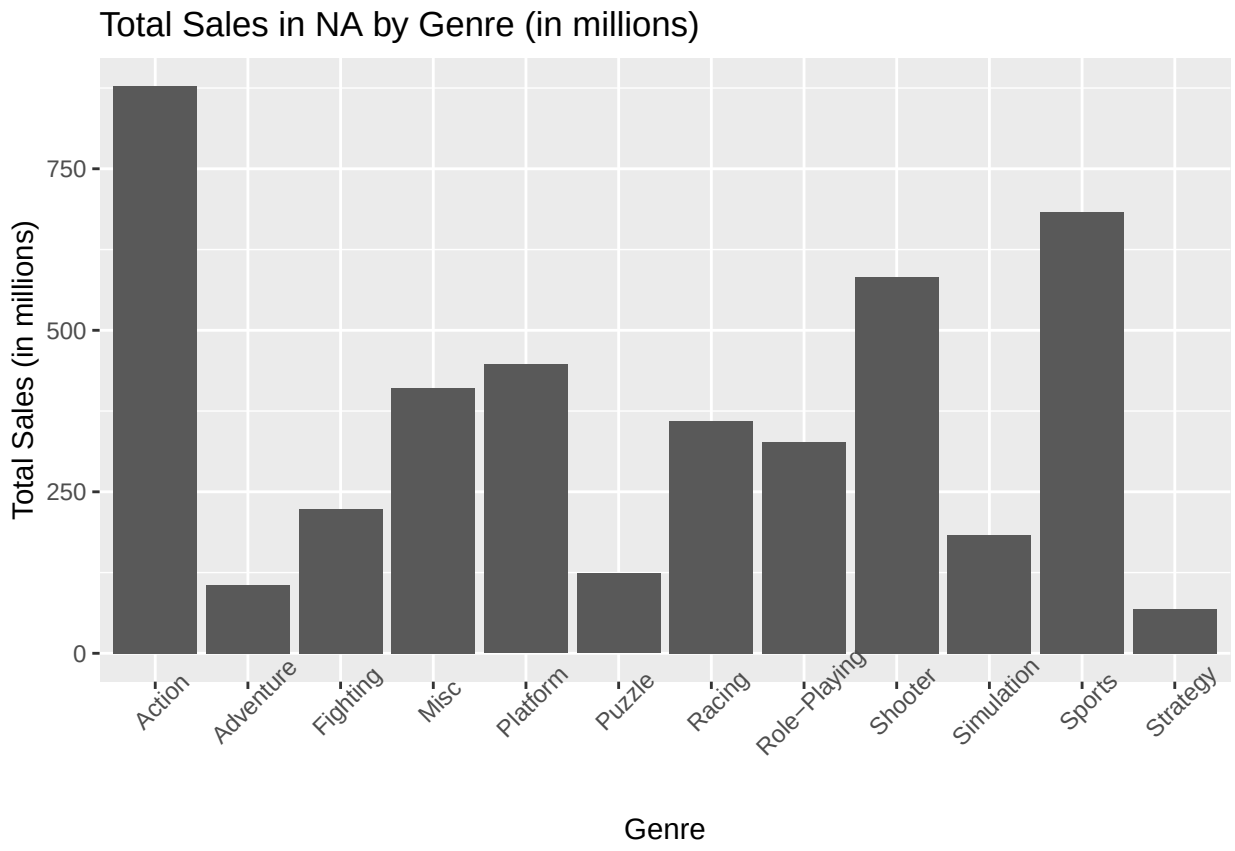
```
# Grouping the data entries based on the "Genre" variable and finding global sales
# for each genre
salesbyGenre <- vgSalesData %>%
  group_by(Genre) %>%
  summarize(
    totalSales = sum(Global_Sales)
  )
```

```r
# Creating a bar chart to visualize global sales for each genre
salesbyGenre %>%
  ggplot(aes(x = Genre, y = totalSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = 'Genre',
    y = 'Total Sales (in millions)',
    title = 'Total Sales by Genre (in millions)'
  ) +
  theme(
    axis.text.x = element_text(angle = 45)
  )
```

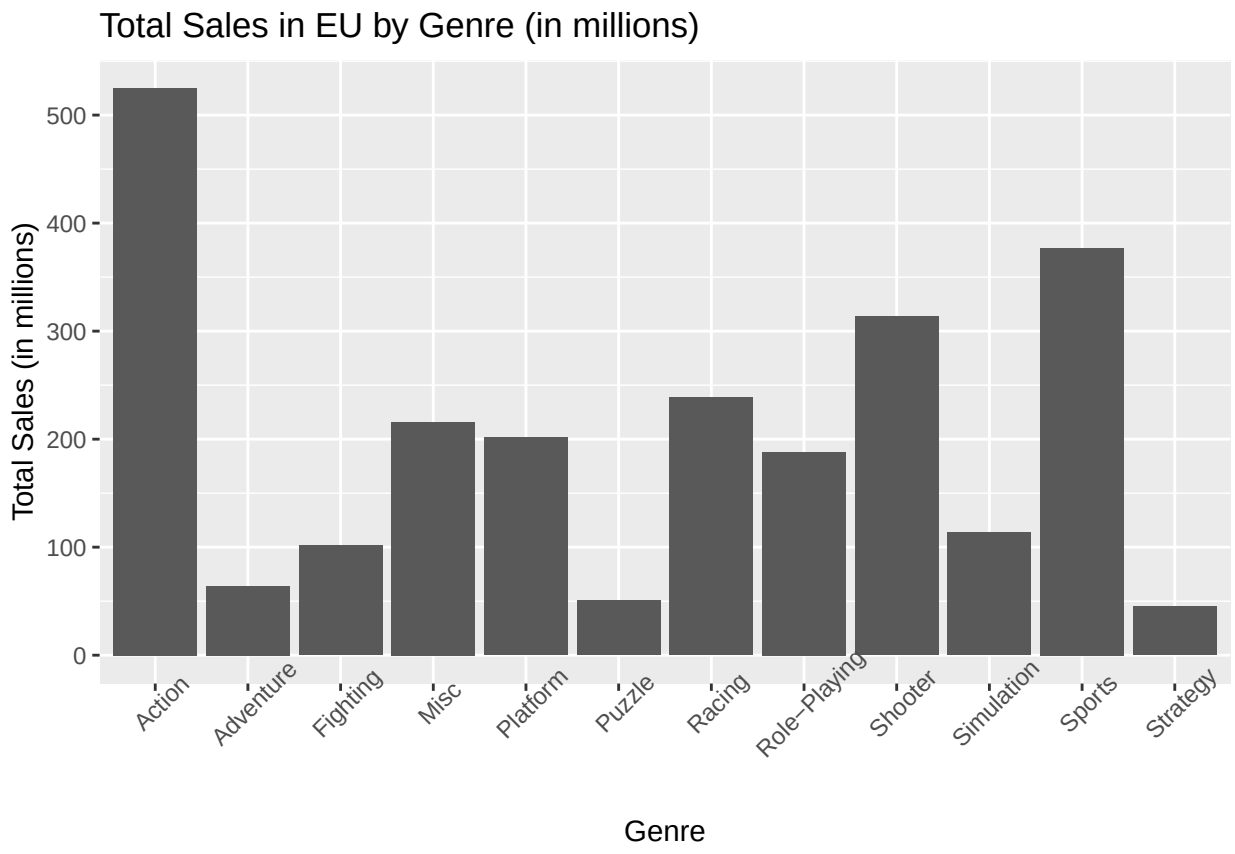



```r
# Grouping the dataset by genre, and then focusing on NA, EU, and JP sales separately
naSalesbyGenre <- vgSalesData %>%
  group_by(Genre) %>%
  summarize(naSales = sum(NA_Sales))

euSalesbyGenre <- vgSalesData %>%
  group_by(Genre) %>%
  summarize(euSales = sum(EU_Sales))

jpSalesbyGenre <- vgSalesData %>%
  group_by(Genre) %>%
  summarize(jpSales = sum(JP_Sales))
```

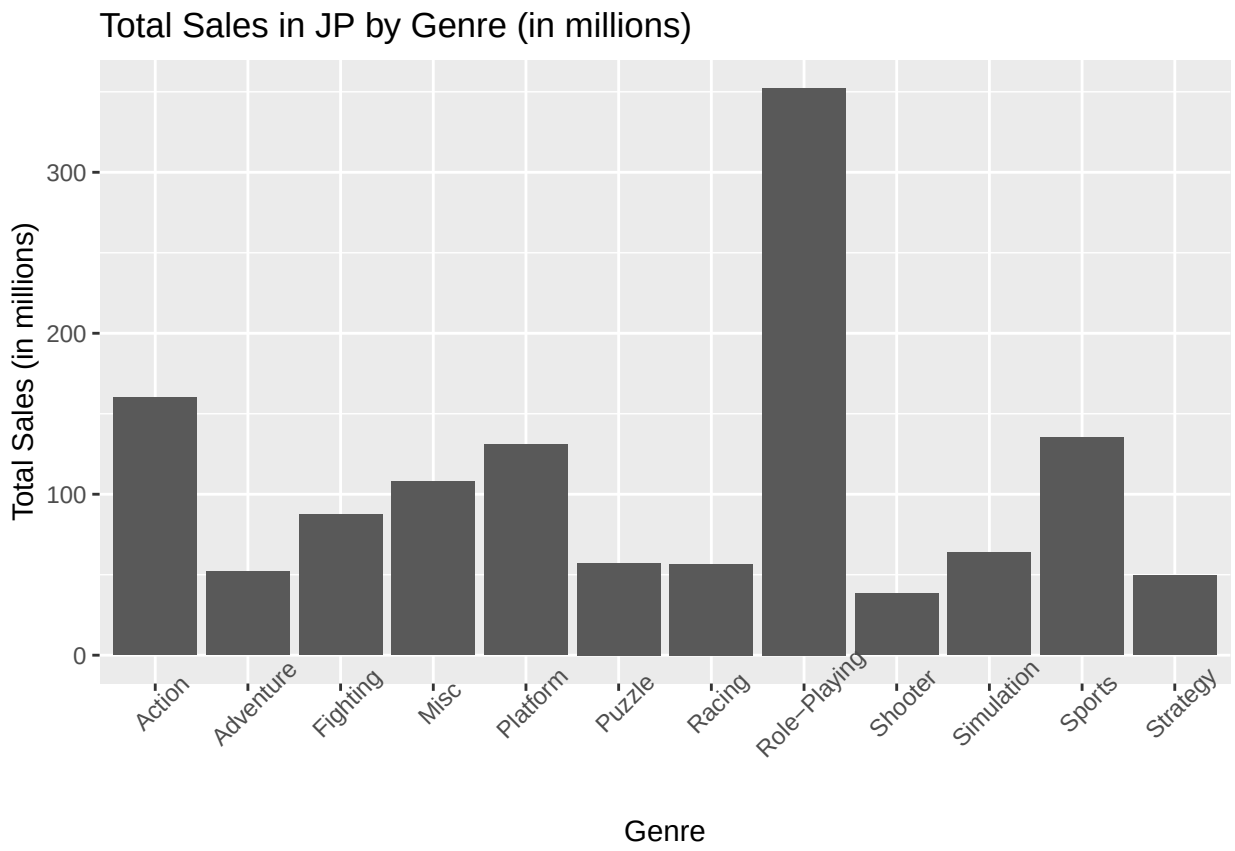
```r
# Creating a bar chart for NA, EU, and JP to visualize the total sales of each
# genre in those regions
naSalesbyGenre %>%
  ggplot(aes(x = Genre, y = naSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = 'Genre',
    y = 'Total Sales (in millions)',
    title = 'Total Sales in NA by Genre (in millions)'
  ) +
  theme(
    axis.text.x = element_text(angle = 45)
  )
```




```r
euSalesbyGenre %>%
  ggplot(aes(x = Genre, y = euSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = 'Genre',
    y = 'Total Sales (in millions)',
    title = 'Total Sales in EU by Genre (in millions)'
  ) +
  theme(
    axis.text.x = element_text(angle = 45)
  )
```




```
jpSalesbyGenre %>%
  ggplot(aes(x = Genre, y = jpSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = 'Genre',
    y = 'Total Sales (in millions)',
    title = 'Total Sales in JP by Genre (in millions)'
  ) +
  theme(
    axis.text.x = element_text(angle = 45)
  )
```




```
# Finding total video game sales by publisher worldwide and in specific regions
salesByPublisher <- vgSalesData %>%
  group_by(Publisher) %>%
  summarize(globalPublisherSales = sum(Global_Sales),
            naPublisherSales = sum(NA_Sales),
            euPublisherSales = sum(EU_Sales),
            jpPublisherSales = sum(JP_Sales),
            otherPublisherSales = sum(Other_Sales))

# Finding publishers with greatest worldwide sales by arrange dataframe in
# descending order of globalPublisherSales
salesByPublisher %>%
  arrange(desc(globalPublisherSales)) %>%
  select(Publisher, globalPublisherSales)
```

```
## # A tibble: 579 x 2
##    Publisher                     globalPublisherSales
##    <chr>                                        <dbl>
##  1 Nintendo                                     1787.
##  2 Electronic Arts                              1110.
##  3 Activision                                    727.
##  4 Sony Computer Entertainment                   608.
##  5 Ubisoft                                       475.
##  6 Take-Two Interactive                          400.
##  7 THQ                                           341.
##  8 Konami Digital Entertainment                  284.
```

```
##  9 Sega                                        273.
## 10 Namco Bandai Games                          254.
## # ... with 569 more rows
```

```
# Finding publishers with greatest NA sales by arranging dataframe in descending
# order of naPublisherSales
salesByPublisher %>%
  arrange(desc(naPublisherSales)) %>%
  select(Publisher, naPublisherSales)
```

```
## # A tibble: 579 x 2
##    Publisher                    naPublisherSales
##    <chr>                                   <dbl>
##  1 Nintendo                                 817.
##  2 Electronic Arts                          595.
##  3 Activision                               430.
##  4 Sony Computer Entertainment              265.
##  5 Ubisoft                                  253.
##  6 Take-Two Interactive                     220.
##  7 THQ                                      209.
##  8 Microsoft Game Studios                   155.
##  9 Atari                                    110.
## 10 Sega                                     109.
## # ... with 569 more rows
```

```
# Finding publishers with greatest EU sales by arranging dataframe in descending
# order of euPublisherSales
salesByPublisher %>%
  arrange(desc(euPublisherSales)) %>%
  select(Publisher, euPublisherSales)
```

```
## # A tibble: 579 x 2
##    Publisher                     euPublisherSales
##    <chr>                                    <dbl>
##  1 Nintendo                                 419.
##  2 Electronic Arts                          371.
##  3 Activision                               216.
##  4 Sony Computer Entertainment              188.
##  5 Ubisoft                                  163.
##  6 Take-Two Interactive                     118.
##  7 THQ                                       94.7
##  8 Sega                                      82
##  9 Konami Digital Entertainment              69.7
## 10 Microsoft Game Studios                    68.6
## # ... with 569 more rows
```

```
# Finding publishers with greatest JP sales by arranging dataframe in descending
# order of jpPublisherSales
salesByPublisher %>%
  arrange(desc(jpPublisherSales)) %>%
  select(Publisher, jpPublisherSales)
```

```
## # A tibble: 579 x 2
##    Publisher                    jpPublisherSales
##    <chr>                                   <dbl>
##  1 Nintendo                                455.
##  2 Namco Bandai Games                      127.
##  3 Konami Digital Entertainment             91.3
##  4 Sony Computer Entertainment              74.1
##  5 Capcom                                   68.1
##  6 Sega                                     57.0
##  7 Square Enix                              49.9
##  8 SquareSoft                               40.1
##  9 Enix Corporation                         32.4
## 10 Tecmo Koei                               29.2
## # ... with 569 more rows
```

```
# Finding publishers with greatest Other sales by arranging dataframe in descending
# order of otherPublisherSales
salesByPublisher %>%
  arrange(desc(otherPublisherSales)) %>%
  select(Publisher, otherPublisherSales)
```

```
## # A tibble: 579 x 2
##    Publisher                    otherPublisherSales
##    <chr>                                      <dbl>
##  1 Electronic Arts                            130.
##  2 Nintendo                                    95.3
##  3 Sony Computer Entertainment                 80.4
##  4 Activision                                  75.3
##  5 Take-Two Interactive                        55.2
##  6 Ubisoft                                     50.3
##  7 THQ                                         32.1
##  8 Konami Digital Entertainment                30.3
##  9 Sega                                        24.5
## 10 Microsoft Game Studios                      18.6
## # ... with 569 more rows
```

```
# Converting Year column into integer type
vgSalesData$Year <- as.integer(vgSalesData$Year)
```

```
## Warning: NAs introduced by coercion
```

```
glimpse(vgSalesData)
```

```
## Rows: 16,598
## Columns: 11
## $ Rank         <dbl> 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         <int> 2006, 1985, 2008, 2009, 1996, 1989, 2006, 2006, 2009, 198~
## $ 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~
```

```
# Re-arranging the data set to observe which year the oldest video games
# in the data set were created
arrange(vgSalesData, Year)
```

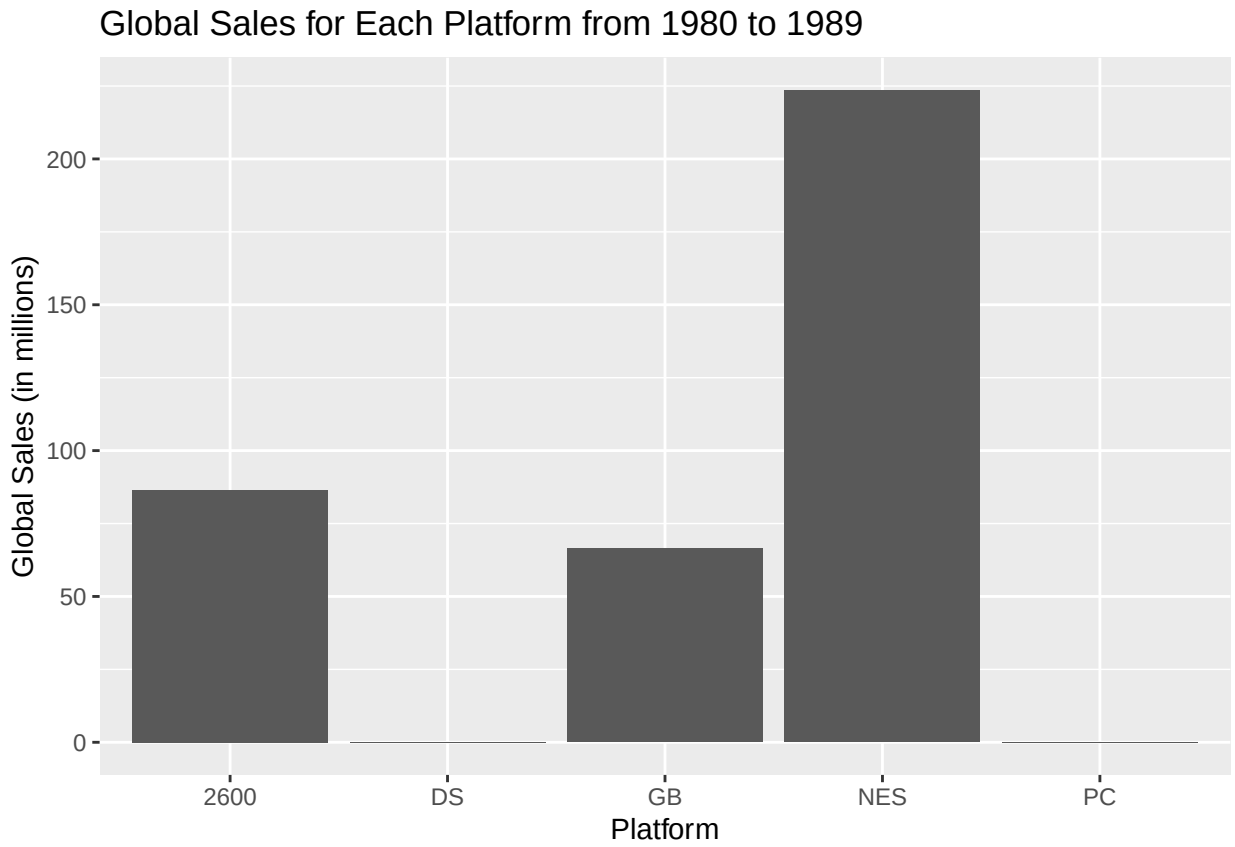
```
## # A tibble: 16,598 x 11
##     Rank Name        Platform  Year Genre   Publisher NA_Sales EU_Sales JP_Sales
##    <dbl> <chr>       <chr>    <int> <chr>   <chr>        <dbl>    <dbl>    <dbl>
##  1   259 Asteroids   2600      1980 Shooter Atari         4        0.26        0
##  2   545 Missile Co~ 2600      1980 Shooter Atari         2.56     0.17        0
##  3  1768 Kaboom!     2600      1980 Misc    Activisi~     1.07     0.07        0
##  4  1971 Defender    2600      1980 Misc    Atari         0.99     0.05        0
##  5  2671 Boxing      2600      1980 Fighti~ Activisi~     0.72     0.04        0
##  6  4027 Ice Hockey  2600      1980 Sports  Activisi~     0.46     0.03        0
##  7  5368 Freeway     2600      1980 Action  Activisi~     0.32     0.02        0
##  8  6319 Bridge      2600      1980 Misc    Activisi~     0.25     0.02        0
##  9  6898 Checkers    2600      1980 Misc    Atari         0.22     0.01        0
## 10   240 Pitfall!    2600      1981 Platfo~ Activisi~     4.21     0.24        0
## # ... with 16,588 more rows, and 2 more variables: Other_Sales <dbl>,
## #   Global_Sales <dbl>
```

```
# Finding out when the latest games in the data set were made
max(vgSalesData$Year, na.rm = TRUE)
```

```
## [1] 2020
```

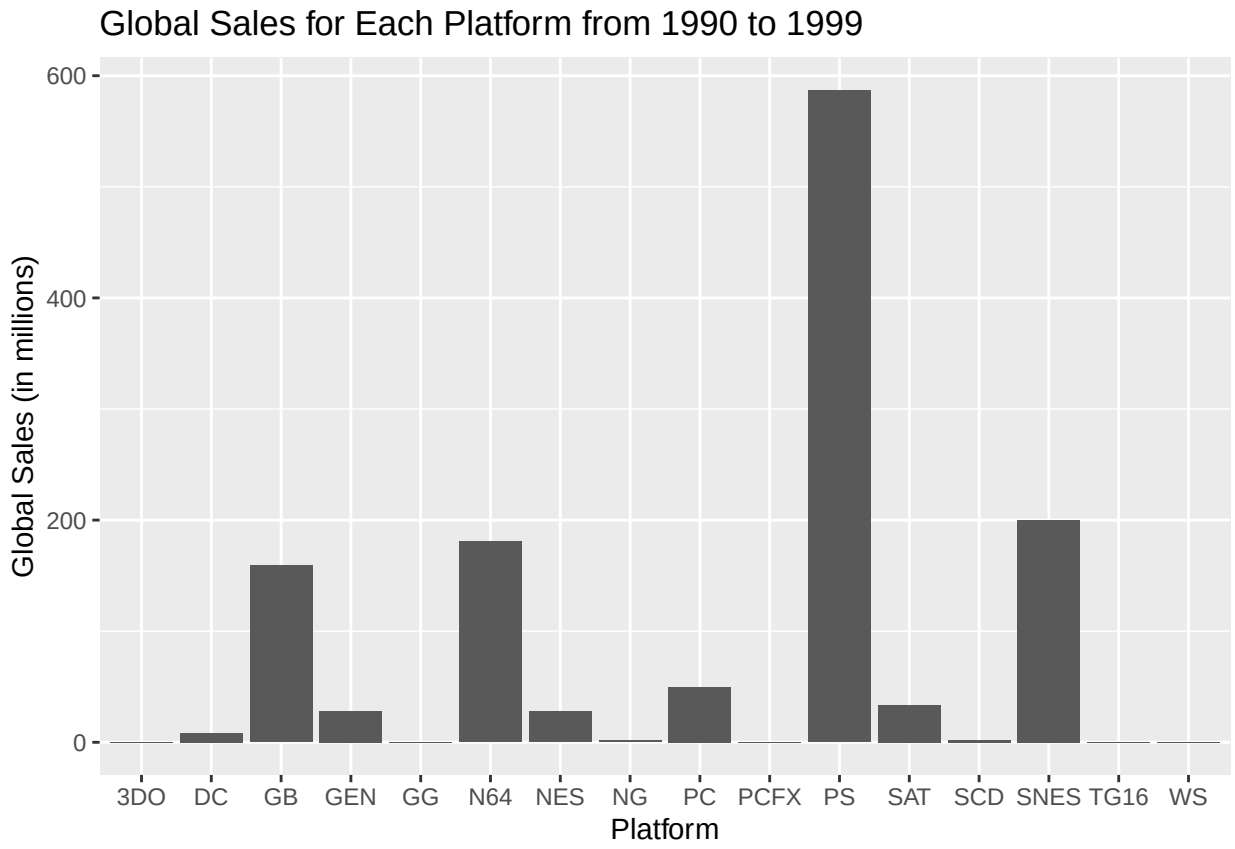
```
# Filtering for data entries that have the years 1980 to 1989
firstDecade <- vgSalesData %>%
  filter(Year >= 1980 & Year <= 1989)

# Grouping the firstDecade data entries by platform, and summarizing the sum of
# each platform's global sales
firstDecadePlatformSales <- firstDecade %>%
  group_by(Platform) %>%
  summarize(
    platformSales = sum(Global_Sales)
  )
```

```
# Creating bar chart to visualize the amount of sales each platform from 1980 to
# 1989 had worldwide
firstDecadePlatformSales %>%
  ggplot(aes(x = Platform, y = platformSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = "Platform",
    y = "Global Sales (in millions)",
    title = "Global Sales for Each Platform from 1980 to 1989"
  )
```



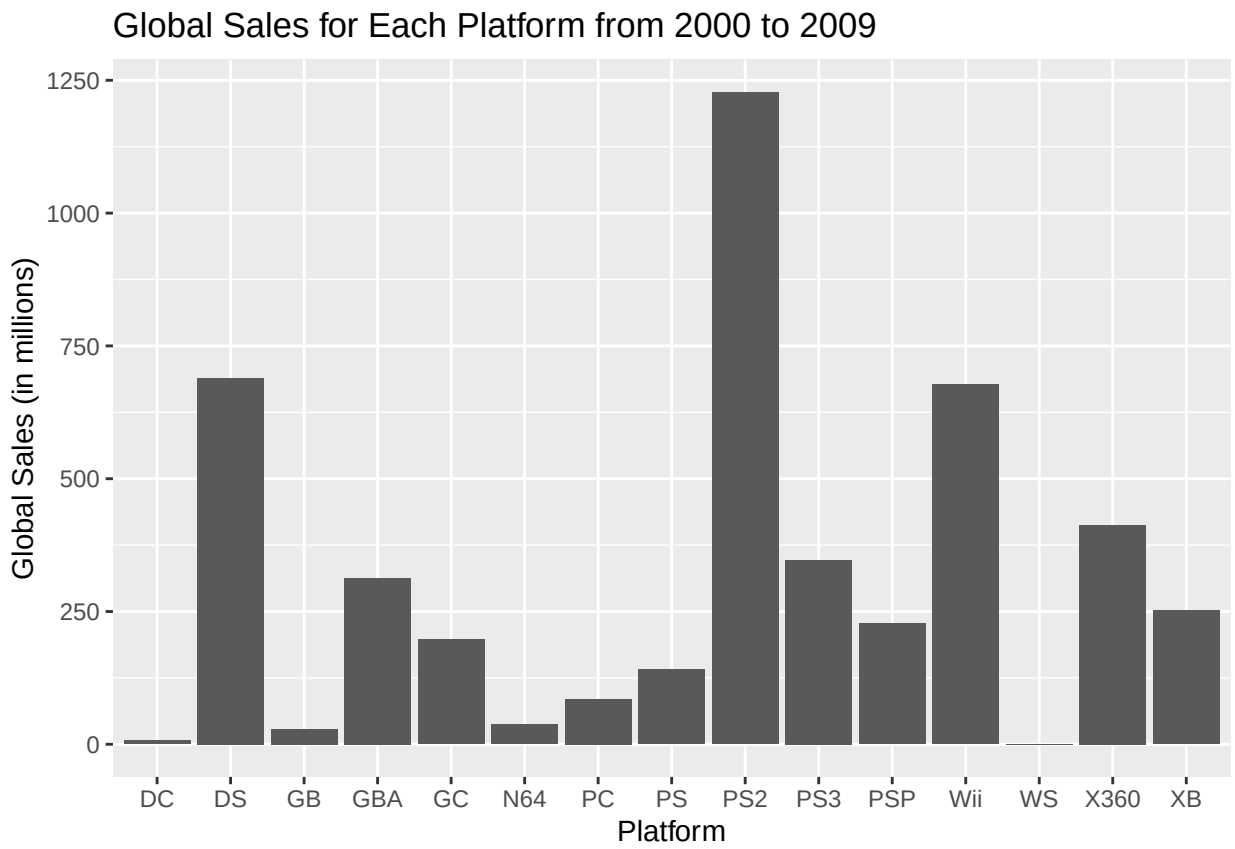

```
# Filtering for data entries that have the years 1990 to 1999
secondDecade <- vgSalesData %>%
  filter(Year >= 1990 & Year <= 1999)

# Grouping the secondDecade data entries by platform, and summarizing the sum of
# each platform's global sales
secondDecadePlatformSales <- secondDecade %>%
  group_by(Platform) %>%
  summarize(
    platformSales = sum(Global_Sales)
  )
```

```
# Creating bar chart to visualize the amount of sales each platform from 1990 to
# 1999 had worldwide
secondDecadePlatformSales %>%
  ggplot(aes(x = Platform, y = platformSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = "Platform",
    y = "Global Sales (in millions)",
    title = "Global Sales for Each Platform from 1990 to 1999"
  )
```



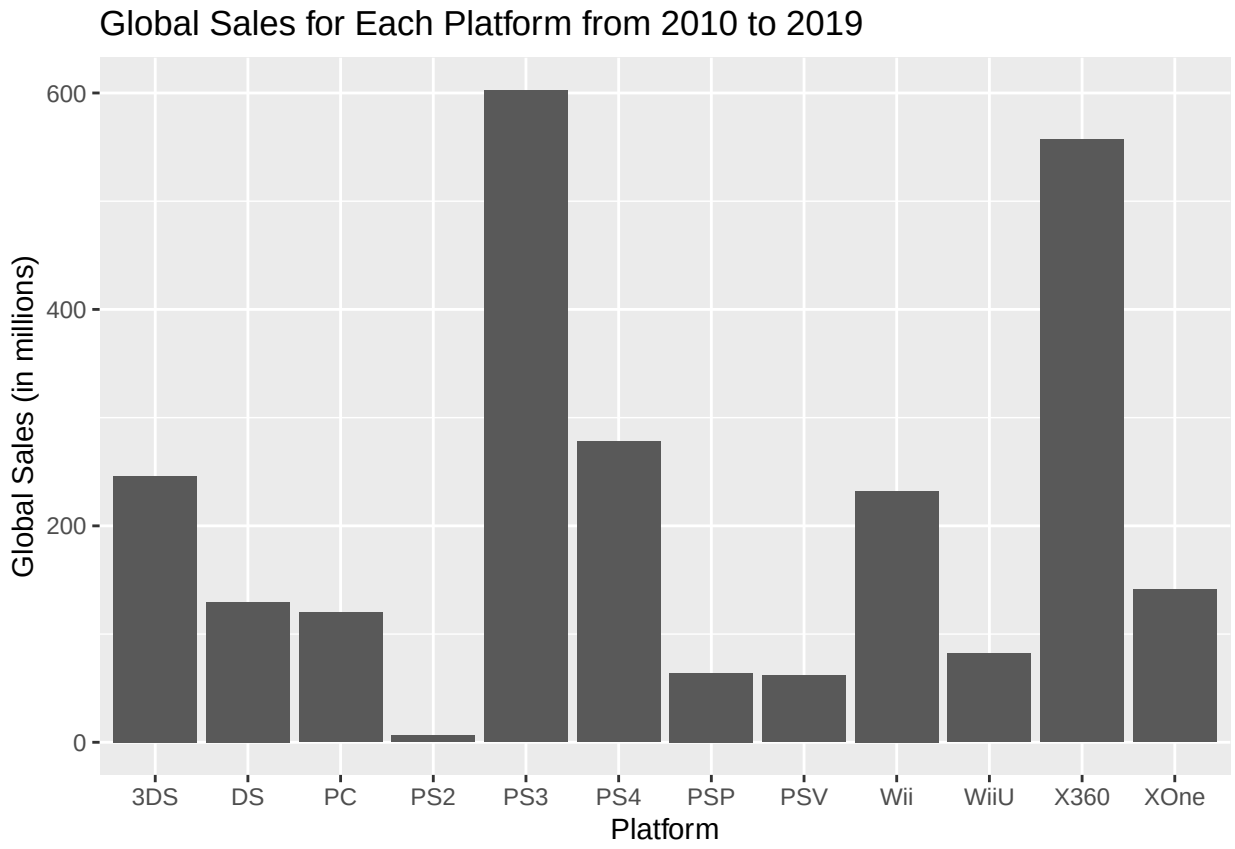

```
# Filtering for data entries that have the years 2000 to 2009
thirdDecade <- vgSalesData %>%
  filter(Year >= 2000 & Year <= 2009)

# Grouping the thirdDecade data entries by platform, and summarizing the sum of
# each platform's global sales
thirdDecadePlatformSales <- thirdDecade %>%
  group_by(Platform) %>%
  summarize(
    platformSales = sum(Global_Sales)
  )
```

```
# Creating bar chart to visualize the amount of sales each platform from 2000 to
# 2009 had worldwide
thirdDecadePlatformSales %>%
  ggplot(aes(x = Platform, y = platformSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = "Platform",
    y = "Global Sales (in millions)",
    title = "Global Sales for Each Platform from 2000 to 2009"
  )
```




```
# Filtering for data entries that have the years 2010 to 2019
lastDecade <- vgSalesData %>%
  filter(Year >= 2010 & Year <= 2019)

# Grouping the lastDecade data entries by platform, and summarizing the sum of
# each platform's global sales
lastDecadePlatformSales <- lastDecade %>%
  group_by(Platform) %>%
  summarize(
    platformSales = sum(Global_Sales)
  )
```

```
# Creating bar chart to visualize the amount of sales each platform from 2010 to
# 2019 had worldwide
lastDecadePlatformSales %>%
  ggplot(aes(x = Platform, y = platformSales)) +
  geom_bar(stat = 'identity') +
  labs(
    x = "Platform",
    y = "Global Sales (in millions)",
    title = "Global Sales for Each Platform from 2010 to 2019"
  )
```

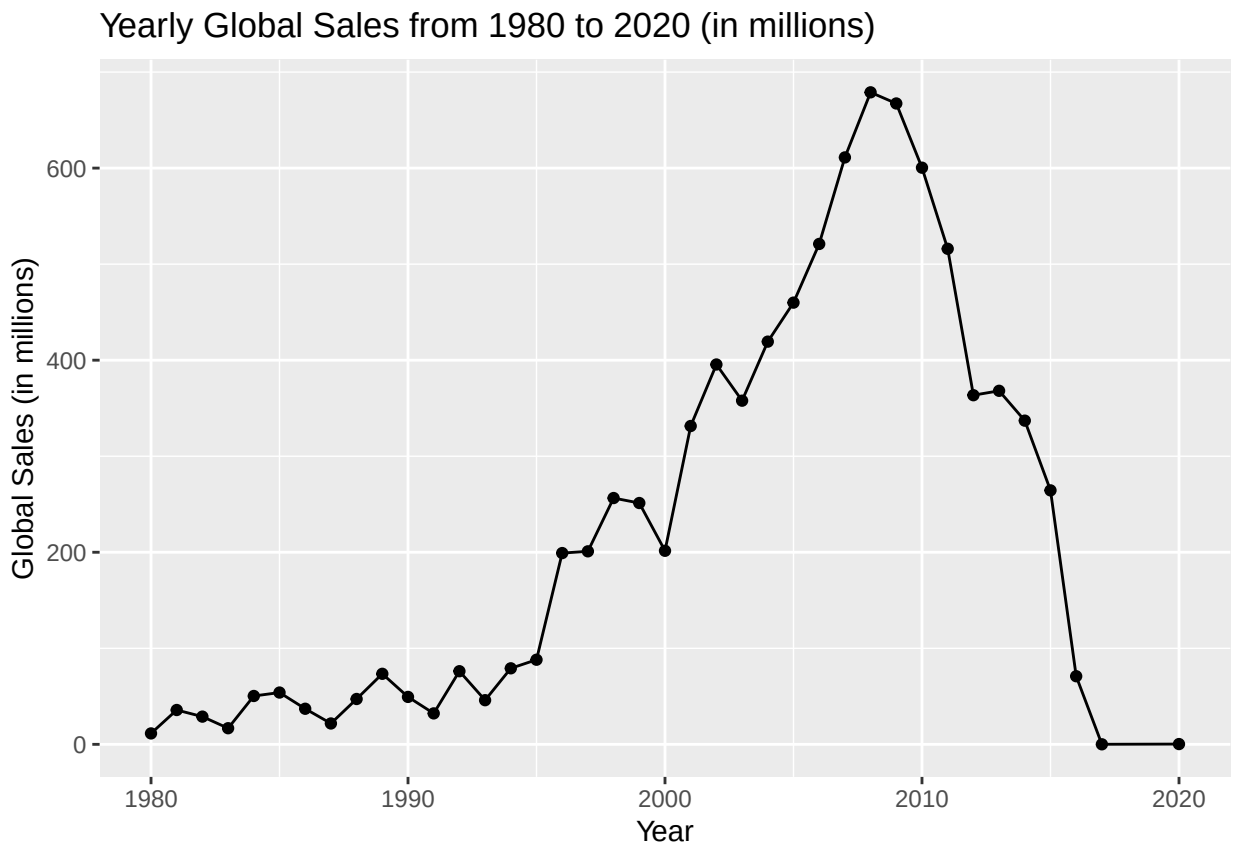



```
# Grouping data entries in original data set by year, and then finding global sales
# for each year
globalSalesByYear <- vgSalesData %>%
  group_by(Year) %>%
  summarize(
    yearlyGlobalSales = sum(Global_Sales)
  )
```

```
# Creating line graph to represent yearly global sales from 1980 to 2020
globalSalesByYear %>%
  ggplot(aes(x = Year, y = yearlyGlobalSales)) +
  geom_line() +
  geom_point() +
  labs(
    y = 'Global Sales (in millions)',
    title = 'Yearly Global Sales from 1980 to 2020 (in millions)'
  )
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```


Yearly Global Sales from 1980 to 2020 (in millions)
Global Sales (in millions)
600
400
200
0
1980
1990
2000
2010
2020
Year