Video Games Sales Prediction

Minh K. Chau

2023-05-04

Introduction

About the Dataset

I like video games. In this project, I will use a data set obtained from **Kaggle**, a data science community, to generate an algorithm to predict sales of video games in North America (NA), Europe (EU), Japan (JP), and other countries. According to Kaggle, this dataset was generated by a scrape of **vgchartz.com**.

The dataset includes the following variables:

- **Rank -** ranking of overall sales
- **Name** - The games name
- **Platform** - Platform of the games release (i.e. PC,PS4, etc.)
- **Year** - Year of the game's release
- **Genre** - Genre of the game
- **Publisher** - Publisher of the game
- **NA_Sales** - Sales in North America (in millions)
- **EU_Sales** - Sales in Europe (in millions)
- **JP_Sales** - Sales in Japan (in millions)
- **Other_Sales** - Sales in the rest of the world (in millions)
- **Global_Sales** - Total worldwide sales.

The data set contains 16,598 rows in-total.

Packages, Read-in Files, and Settings

For this project, the following packages were used:

- **Tidyverse -** for data wrangling, transformation, and plotting
- **Caret -** to develop a Machine Learning algorithm
- **ggpubr -** to combine multiple plots into one

- **Randomforest** - generate an algorithm using Random Forest approach

Seed will be set at 5 for consistency in results and write-up. Numerical outputs will be rounded up to 3 digits after the decimal place.

The dataset was renamed as **Orig (original).**

```
#---------------Library & Read-in Files----------------
library(tidyverse)
library(caret)
library(ggpubr)
library(randomForest)

setwd("C:/Users/minhk/Documents/GitHub/ML - Video games sales")
orig <- read.csv("vgsales.csv")

options(digits = 3)
set.seed(5)
```

Methods

Data Exploration & Transformation

Year

```
range(orig$Year)
```

```
## [1] "1980" "N/A"
```

```
length(unique(orig$Year))
```

```
## [1] 40
```

The "Year" variable has a range between 1980 - NA, with about 40 unique values. This indicates a presence of an unknown values. Data transformation is conducted, transforming years in numerical values into categorical ones, using the following rule:

Table 1: Table 1. Rule for transforming years into Years.ranged

From	To	Category
1980	1984	1980s
1985	1989	1985s
1990	1994	1990s
1995	1999	1995s
2000	2004	2000s
2005	2009	2005s
2010	2014	2010s
2015	2020	2015s

From	To	Category
	Missing or unknown	unknown

```
# Transform individual Years to Yr-Released
orig <- orig %>%
  mutate(Year = as.numeric(Year),
         yr.released = ifelse(Year < 1985, "1980s",
                        ifelse(Year >=1985 & Year <1990,"1985s",
                        ifelse(Year >=1990 & Year <1995,"1990s",
                        ifelse(Year >=1995 & Year <2000,"1995s",
                        ifelse(Year >=2000 & Year <2005,"2000s",
                        ifelse(Year >=2005 & Year <2010,"2005s",
                        ifelse(Year >=2010 & Year <2015,"2010s",
                        ifelse(Year >=2015 & Year <=2020,"2015s","N/A")))))))))

orig <- orig %>%
  mutate(yr.released = ifelse(is.na(yr.released) == TRUE,"unknown",yr.released))

orig %>%
  group_by(yr.released) %>%
  count()
```

```
## # A tibble: 9 x 2
## # Groups:   yr.released [9]
##   yr.released     n
##   <chr>       <int>
## 1 1980s         122
## 2 1985s          83
## 3 1990s         281
## 4 1995s        1488
## 5 2000s        3198
## 6 2005s        6010
## 7 2010s        4183
## 8 2015s         962
## 9 unknown       271
```

The above tibble showed the number of rows by different groups of Years released.

Publishers

I first explored about all different Publishers, and their global sales in total.

```
length(unique(orig$Publisher))
```

```
## [1] 579
```

There are about 579 unique Publishers.

```
orig %>%
  select(Global_Sales,
         Publisher) %>%
  group_by(Publisher) %>%
  summarise(Total.sales = sum(Global_Sales)) %>%
  arrange(desc(Total.sales)) %>%
  head(50) %>%
  data.frame()
```

```
##                                Publisher Total.sales
## 1                               Nintendo      1786.6
## 2                         Electronic Arts      1110.3
## 3                              Activision       727.5
## 4              Sony Computer Entertainment       607.5
## 5                                 Ubisoft       474.7
## 6                     Take-Two Interactive       399.5
## 7                                      THQ       340.8
## 8             Konami Digital Entertainment       283.6
## 9                                    Sega       273.0
## 10                    Namco Bandai Games       254.1
## 11                 Microsoft Game Studios       245.8
## 12                                 Capcom       200.9
## 13                                  Atari       157.2
## 14 Warner Bros. Interactive Entertainment       153.9
## 15                               Square Enix       145.2
## 16             Disney Interactive Studios       120.0
## 17                        Eidos Interactive        99.0
## 18                                LucasArts        87.3
## 19                     Bethesda Softworks        82.1
## 20                           Midway Games        69.8
## 21                  Acclaim Entertainment        64.1
## 22                          Vivendi Games        58.2
## 23                             SquareSoft        57.6
## 24                              505 Games        55.9
## 25                             Tecmo Koei        53.5
## 26                            Codemasters        47.9
## 27                     Virgin Interactive        43.9
## 28                                Unknown        34.7
## 29                          Enix Corporation        33.7
## 30                               Deep Silver        25.7
## 31                            GT Interactive        25.2
## 32                               D3Publisher        24.1
## 33     Sony Computer Entertainment Europe        23.9
## 34                               Hudson Soft        23.0
## 35                                MTV Games        21.0
## 36                  Universal Interactive        17.8
## 37                                Banpresto        17.4
## 38                      Rising Star Games        17.3
## 39                               Infogrames        16.3
## 40                  Majesco Entertainment        16.1
## 41                     Hasbro Interactive        15.2
## 42                                    N/A        14.9
## 43                   Nippon Ichi Software        14.3
```

```
## 44                          989 Studios        13.3
## 45                Zoo Digital Publishing        12.9
## 46                                Atlus        12.7
## 47                              Level 5        12.2
## 48                   Empire Interactive        11.3
## 49                  ASCII Entertainment        10.9
## 50                                  3DO        10.1
```

Per the table above, there are some publishers that sold a lot of video games while others were not as popular. For this variable, I grouped publishers into categories based on their total global sales. The rules are as followed:

Table 2: Publishers & their Rank

Publishers	Ranking
Nintendo, Electronic Arts, Activision, Sony Computer Entertainment, Ubisoft	top-5
Take-Two Interactive, THQ, Konami Digital Entertainment, Sega, Namco Bandai Games	top-10
Microsoft Game Studios, Capcom, Atari, Warner Bros. Interactive Entertainment, Squared Enix	top-15
Disney Interactive Studios, Elidos Interactive, LucasArts, Bethesda Softworks, Midway Games	top-20
Acclaim Entertainment, Vivendi Games, SquareSoft, 505 Games, Tecmo Koei	top-25
Codemasters, Virgin Interactive, Enix Corporation, Deep Silver, GT Interactive	top-30
all other publishers	other

```
orig <- orig %>%
  mutate(publisher.rank = ifelse(Publisher == "Nintendo","top-5",
                          ifelse(Publisher == "Electronic Arts","top-5",
                          ifelse(Publisher == "Activision","top-5",
                          ifelse(Publisher == "Sony Computer Entertainment","top-5",
                          ifelse(Publisher == "Ubisoft","top-5",
                          ifelse(Publisher == "Take-Two Interactive","top-10",
                          ifelse(Publisher == "THQ","top-10",
                          ifelse(Publisher == "Konami Digital Entertainment","top-10",
                          ifelse(Publisher == "Sega","top-10",
                          ifelse(Publisher == "Namco Bandai Games","top-10",
                          ifelse(Publisher == "Microsoft Game Studios","top-15",
                          ifelse(Publisher == "Capcom","top-15",
                          ifelse(Publisher == "Atari","top-15",
                          ifelse(Publisher == "Warner Bros. Interactive Entertainment","top-15",
                          ifelse(Publisher == "Square Enix","top-15",
                          ifelse(Publisher == "Disney Interactive Studios","top-20",
                          ifelse(Publisher == "Eidos Interactive","top-20",
                          ifelse(Publisher == "LucasArts","top-20",
                          ifelse(Publisher == "Bethesda Softworks","top-20",
                          ifelse(Publisher == "Midway Games","top-20",
                          ifelse(Publisher == "Acclaim Entertainment","top-25",
                          ifelse(Publisher == "Vivendi Games","top-25",
                          ifelse(Publisher == "SquareSoft","top-25",
                          ifelse(Publisher == "505 Games","top-25",
                          ifelse(Publisher == "Tecmo Koei","top-25",
                          ifelse(Publisher == "Codemasters","top-30",
```

```
                          ifelse(Publisher == "Virgin Interactive","top-30",
                          ifelse(Publisher == "Enix Corporation","top-30",
                          ifelse(Publisher == "Deep Silver","top-30",
                          ifelse(Publisher == "GT Interactive","top-30","other")
                          ))))))))))))))))))))))))))))))

orig %>%
  group_by(publisher.rank) %>%
  summarise(Total.sales = sum(Global_Sales),
            n = n(),
            mean.sales = mean(Global_Sales)) %>%
  arrange(desc(mean.sales))
```

```
## # A tibble: 7 x 4
##   publisher.rank Total.sales     n mean.sales
##   <chr>                <dbl> <int>      <dbl>
## 1 top-5               4707.  4633      1.02
## 2 top-15               903.  1398      0.646
## 3 top-20               458.   775      0.591
## 4 top-10              1551.  3531      0.439
## 5 top-30               176.   411      0.429
## 6 top-25               289.   930      0.311
## 7 other                836.  4920      0.170
```

Per the table above, publishers were ranked according to global sales and their mean sales.

Platforms

```
length(unique(orig$Platform))
```

```
## [1] 31
```

There are 31 different platforms. I proceeded to compare them by their Global Sales.

```
orig %>%
  group_by(Platform) %>%
  summarise(n = n(),
            mean = mean(Global_Sales),
            sum = sum(Global_Sales))%>%
  arrange(desc(sum)) %>%
  print(n = 32)
```

```
## # A tibble: 31 x 4
##    Platform     n   mean     sum
##    <chr>    <int>  <dbl>   <dbl>
##  1 PS2       2161 0.581  1256.
##  2 X360      1265 0.775   980.
##  3 PS3       1329 0.721   958.
##  4 Wii       1325 0.699   927.
```

```
##  5 DS        2163 0.380   822.
##  6 PS        1196 0.611   731.
##  7 GBA        822 0.387   318.
##  8 PSP       1213 0.244   296.
##  9 PS4        336 0.828   278.
## 10 PC         960 0.270   259.
## 11 XB         824 0.313   258.
## 12 GB          98 2.61    255.
## 13 NES         98 2.56    251.
## 14 3DS        509 0.486   247.
## 15 N64        319 0.686   219.
## 16 SNES       239 0.837   200.
## 17 GC         556 0.359   199.
## 18 XOne       213 0.662   141.
## 19 2600       133 0.730    97.1
## 20 WiiU       143 0.572    81.9
## 21 PSV        413 0.150    61.9
## 22 SAT        173 0.194    33.6
## 23 GEN         27 1.05     28.4
## 24 DC          52 0.307    16.0
## 25 SCD          6 0.312     1.87
## 26 NG          12 0.12      1.44
## 27 WS           6 0.237     1.42
## 28 TG16         2 0.08      0.16
## 29 3DO          3 0.0333    0.1
## 30 GG           1 0.04      0.04
## 31 PCFX         1 0.03      0.03
```

With the sum of global sales less than 2 millions for SCD, NG, WS, TG16, 3DO, GG, PCFX, I decided that they are not popular platforms. I will rename those platforms into "Not-Popular."

```
# the last platforms are very low => rename them to other
orig <- orig %>%
  mutate(Platform = ifelse(Platform == "SCD","Not-Popular",
                    ifelse(Platform == "NG","Not-Popular",
                    ifelse(Platform == "WS","Not-Popular",
                    ifelse(Platform == "TG16","Not-Popular",
                    ifelse(Platform == "3DO","Not-Popular",
                    ifelse(Platform == "GG","Not-Popular",
                    ifelse(Platform == "PCFX","Not-Popular",Platform)))))))

orig %>%
  group_by(Platform) %>%
  summarise(n = n(),
            mean = mean(Global_Sales),
            sum = sum(Global_Sales))%>%
  arrange(desc(sum)) %>%
  print(n = 32)
```

```
## # A tibble: 25 x 4
##    Platform        n  mean     sum
##    <chr>       <int> <dbl>   <dbl>
##  1 PS2          2161 0.581 1256.
```

```
##  2 X360         1265 0.775  980.
##  3 PS3          1329 0.721  958.
##  4 Wii          1325 0.699  927.
##  5 DS           2163 0.380  822.
##  6 PS           1196 0.611  731.
##  7 GBA           822 0.387  318.
##  8 PSP          1213 0.244  296.
##  9 PS4           336 0.828  278.
## 10 PC            960 0.270  259.
## 11 XB            824 0.313  258.
## 12 GB             98 2.61   255.
## 13 NES            98 2.56   251.
## 14 3DS           509 0.486  247.
## 15 N64           319 0.686  219.
## 16 SNES          239 0.837  200.
## 17 GC            556 0.359  199.
## 18 XOne          213 0.662  141.
## 19 2600          133 0.730   97.1
## 20 WiiU          143 0.572   81.9
## 21 PSV           413 0.150   61.9
## 22 SAT           173 0.194   33.6
## 23 GEN            27 1.05    28.4
## 24 DC             52 0.307   16.0
## 25 Not-Popular    31 0.163    5.06
```

Genres

```
unique(orig$Genre)
```

```
##  [1] "Sports"       "Platform"     "Racing"       "Role-Playing" "Puzzle"
##  [6] "Misc"         "Shooter"      "Simulation"   "Action"       "Fighting"
## [11] "Adventure"    "Strategy"
```

```
sum(is.na(orig$Genre))
```

```
## [1] 0
```

There is no missing data for gaming genres. There are 12 genres in total.

```
orig %>%
  group_by(Genre) %>%
  summarise(n = n(),
            mean = mean(Global_Sales),
            sum = sum(Global_Sales))%>%
  arrange(desc(sum)) %>%
  print(n = 12)
```

```
## # A tibble: 12 x 4
##    Genre            n  mean   sum
##    <chr>        <int> <dbl> <dbl>
```

```
##  1 Action        3316 0.528 1751.
##  2 Sports        2346 0.567 1331.
##  3 Shooter       1310 0.792 1037.
##  4 Role-Playing  1488 0.623  927.
##  5 Platform       886 0.938  831.
##  6 Misc          1739 0.466  810.
##  7 Racing        1249 0.586  732.
##  8 Fighting       848 0.529  449.
##  9 Simulation     867 0.452  392.
## 10 Puzzle         582 0.421  245.
## 11 Adventure     1286 0.186  239.
## 12 Strategy       681 0.257  175.
```

The tibble above indicates that some genres are more likely to have more sales while there are some that are not.

Data Visualization

Individual Plots

The following code chunk makes individual barplots of each of the following sales by ***platforms, released year, genre, publishers, top-10 publishers***

- Global
- North America
- Europe
- Japan
- Other Countries

```
#----Global sales
# by Platforms
glo.plat <- orig %>%
  select(Global_Sales,
         Platform) %>%
  group_by(Platform) %>%
  summarise(sum = sum(Global_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Platform,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
```

```
  xlab("Global")

# by Released Year
glo.yr <- orig %>%
  select(Global_Sales,
         yr.released) %>%
  group_by(yr.released) %>%
  summarise(sum = sum(Global_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = yr.released,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(yr.released,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Global")

# by Genre
glo.genres <- orig %>%
  select(Global_Sales,
         Genre) %>%
  group_by(Genre) %>%
  summarise(sum = sum(Global_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Genre,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Genre,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Global")

# by Publishers
glo.pub <- orig %>%
  select(Global_Sales,
         publisher.rank) %>%
  group_by(publisher.rank) %>%
  summarise(sum = sum(Global_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = publisher.rank,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
```

```
                          y= reorder(publisher.rank,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Global")

# by top 10 publishers
glo.pus <- orig %>%
  select(Global_Sales,
         Publisher) %>%
  group_by(Publisher) %>%
  summarise(sum = sum(Global_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Publisher,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Global")

#----North America sales
# NA sales by Platforms
na.plat <- orig %>%
  select(NA_Sales,
         Platform) %>%
  group_by(Platform) %>%
  summarise(sum = sum(NA_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Platform, sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("North America")
```

```r
# NA Sales by Released Year
na.yr <- orig %>%
  select(NA_Sales,
         yr.released) %>%
  group_by(yr.released) %>%
  summarise(sum = sum(NA_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = yr.released,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(yr.released,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("North America")

# NA sales by Genre
na.genres <- orig %>%
  select(NA_Sales,
         Genre) %>%
  group_by(Genre) %>%
  summarise(sum = sum(NA_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Genre,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Genre,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("North America")

# by Publishers
na.pub <- orig %>%
  select(NA_Sales,
         publisher.rank) %>%
  group_by(publisher.rank) %>%
  summarise(sum = sum(NA_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = publisher.rank,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(publisher.rank,sum)),
           stat = "identity",
```

```
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("North America")

# by top 10 publishers
na.pus <- orig %>%
  select(NA_Sales,
         Publisher) %>%
  group_by(Publisher) %>%
  summarise(sum = sum(NA_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Publisher,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("North America")

#----European sales
# Europe sales by Platforms
eu.plat <- orig %>%
  select(EU_Sales,
         Platform) %>%
  group_by(Platform) %>%
  summarise(sum = sum(EU_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Platform, sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Europe")

# EU Sales by Released Year
```

```r
eu.yr <- orig %>%
  select(EU_Sales,
         yr.released) %>%
  group_by(yr.released) %>%
  summarise(sum = sum(EU_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = yr.released,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(yr.released,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Europe")

# EU sales by Genre
eu.genres <- orig %>%
  select(EU_Sales,
         Genre) %>%
  group_by(Genre) %>%
  summarise(sum = sum(EU_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Genre,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Genre,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Europe")

# by Publishers
eu.pub <- orig %>%
  select(EU_Sales,
         publisher.rank) %>%
  group_by(publisher.rank) %>%
  summarise(sum = sum(EU_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = publisher.rank,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(publisher.rank,sum)),
           stat = "identity",
           width = 0.5,
```

```
          position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Europe")

# by top 10 publishers
eu.pus <- orig %>%
  select(EU_Sales,
         Publisher) %>%
  group_by(Publisher) %>%
  summarise(sum = sum(EU_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Publisher,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Europe")

#----Japanese Sales
# Japan sales by Platforms
jp.plat <- orig %>%
  select(JP_Sales,
         Platform) %>%
  group_by(Platform) %>%
  summarise(sum = sum(JP_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Platform, sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Japan")

# JP Sales by Released Year
jp.yr <- orig %>%
```

```
  select(JP_Sales,
         yr.released) %>%
  group_by(yr.released) %>%
  summarise(sum = sum(JP_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = yr.released,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(yr.released,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Japan")

# JP sales by Genre
jp.genres <- orig %>%
  select(JP_Sales,
         Genre) %>%
  group_by(Genre) %>%
  summarise(sum = sum(JP_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Genre,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Genre,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Japan")

# by Publishers
jp.pub <- orig %>%
  select(JP_Sales,
         publisher.rank) %>%
  group_by(publisher.rank) %>%
  summarise(sum = sum(JP_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = publisher.rank,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(publisher.rank,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
```

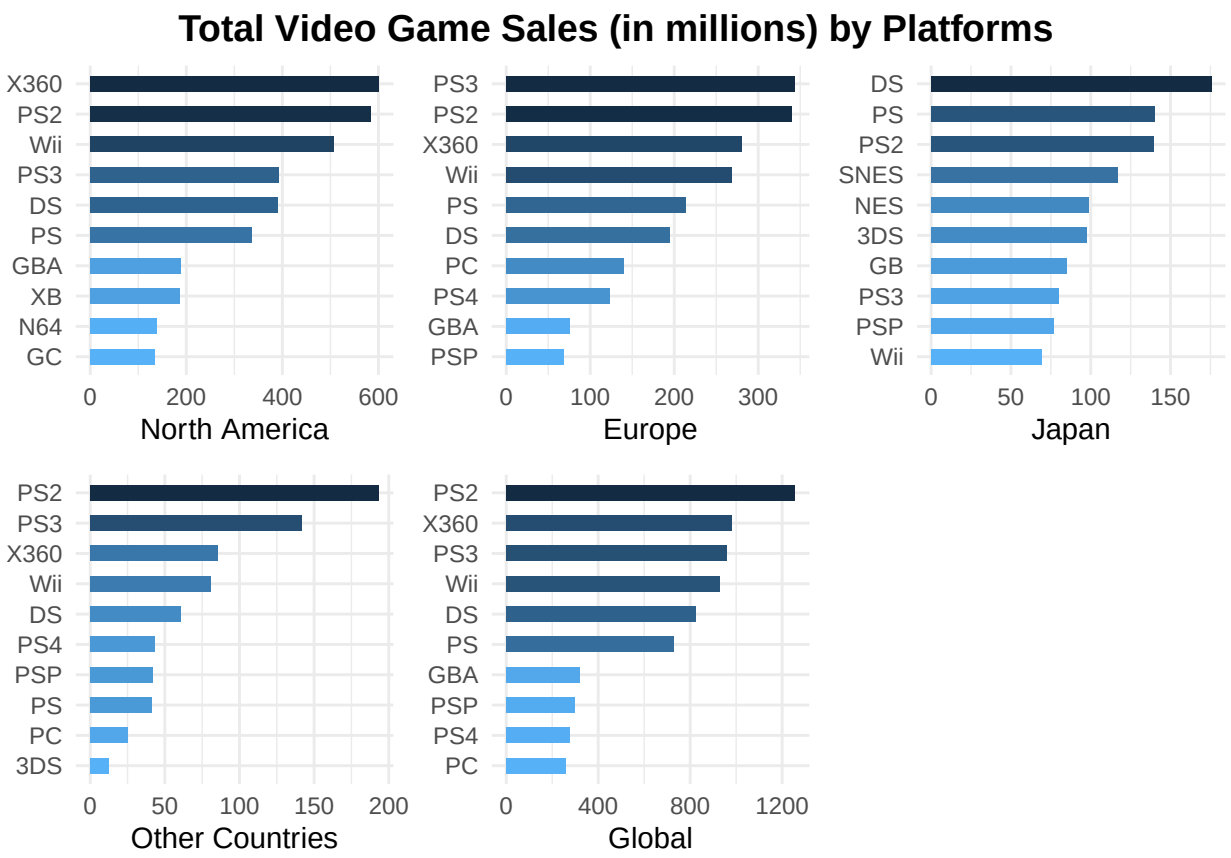
```
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Japan")

# by top 10 publishers
jp.pus <- orig %>%
  select(JP_Sales,
         Publisher) %>%
  group_by(Publisher) %>%
  summarise(sum = sum(JP_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Publisher,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Japan")

#----Other Country Sales
# Other countries sales by Platforms
other.plat <- orig %>%
  select(Other_Sales,
         Platform) %>%
  group_by(Platform) %>%
  summarise(sum = sum(Other_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Platform, sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Other Countries")

#---
# other Sales by Released Year
other.yr <- orig %>%
```

```
  select(Other_Sales,
         yr.released) %>%
  group_by(yr.released) %>%
  summarise(sum = sum(Other_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = yr.released,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(yr.released,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Other Countries")

# other sales by Genre
other.genres <- orig %>%
  select(Other_Sales,
         Genre) %>%
  group_by(Genre) %>%
  summarise(sum = sum(Other_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Genre,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Genre,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Other Countries")

# by Publishers
other.pub <- orig %>%
  select(Other_Sales,
         publisher.rank) %>%
  group_by(publisher.rank) %>%
  summarise(sum = sum(Other_Sales)) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = publisher.rank,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(publisher.rank,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
```

```
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Other Countries")

# by top 10 publishers
other.pus <- orig %>%
  select(Other_Sales,
         Publisher) %>%
  group_by(Publisher) %>%
  summarise(sum = sum(Other_Sales)) %>%
  arrange(desc(sum)) %>%
  head(10) %>%
  data.frame() %>%
  ggplot(aes(x = sum,
             y = Platform,
             fill = -sum)) +
  geom_bar(mapping = aes(x= sum,
                         y= reorder(Publisher,sum)),
           stat = "identity",
           width = 0.5,
           position = "dodge") +
  theme_minimal() +
  theme(axis.title.y = element_blank(),
        legend.position = "none") +
  xlab("Other Countries")
```

Comparing Sales by Platforms

```
ggarrange(na.plat,
          eu.plat,
          jp.plat,
          other.plat,
          glo.plat,
          legend.grob = NULL,
          legend = NULL) %>%
  annotate_figure(top = text_grob("Total Video Game Sales (in millions) by Platforms",
                                  color = "black",
                                  face = "bold",
                                  size = 14))
```



The plot above combine plots from different individual countries and compare them by Platforms. By sorting them in a descending order, I concluded that each country have different preference in gaming platforms. For example:

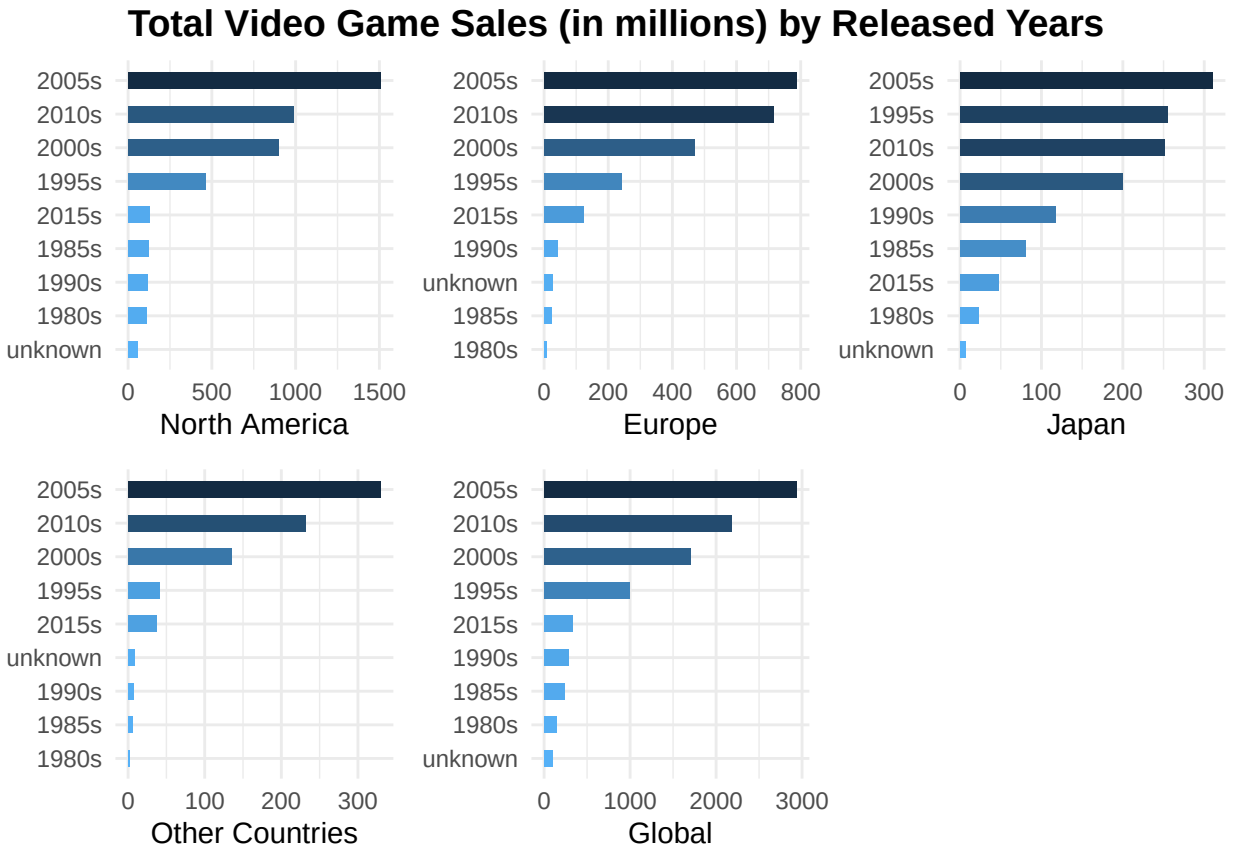
- NA: X360, PS2, Wii
- EU: PS3, PS2, X360
- Japan: DS, PS, Ps2
- Other countries: PS2, PS3, X360

But also, looking at the different scales, I can conclude that some countries buy more games than other. For example, NA buys more games than any other countries, followed by Europe, other countries, and then Japan. This is attributed to different population size or it can be different demands as well.

Comparing Sales by Released Years

```
# -----Released Years -----
ggarrange(na.yr,
          eu.yr,
          jp.yr,
          other.yr,
          glo.yr,
          legend = NULL) %>%
```

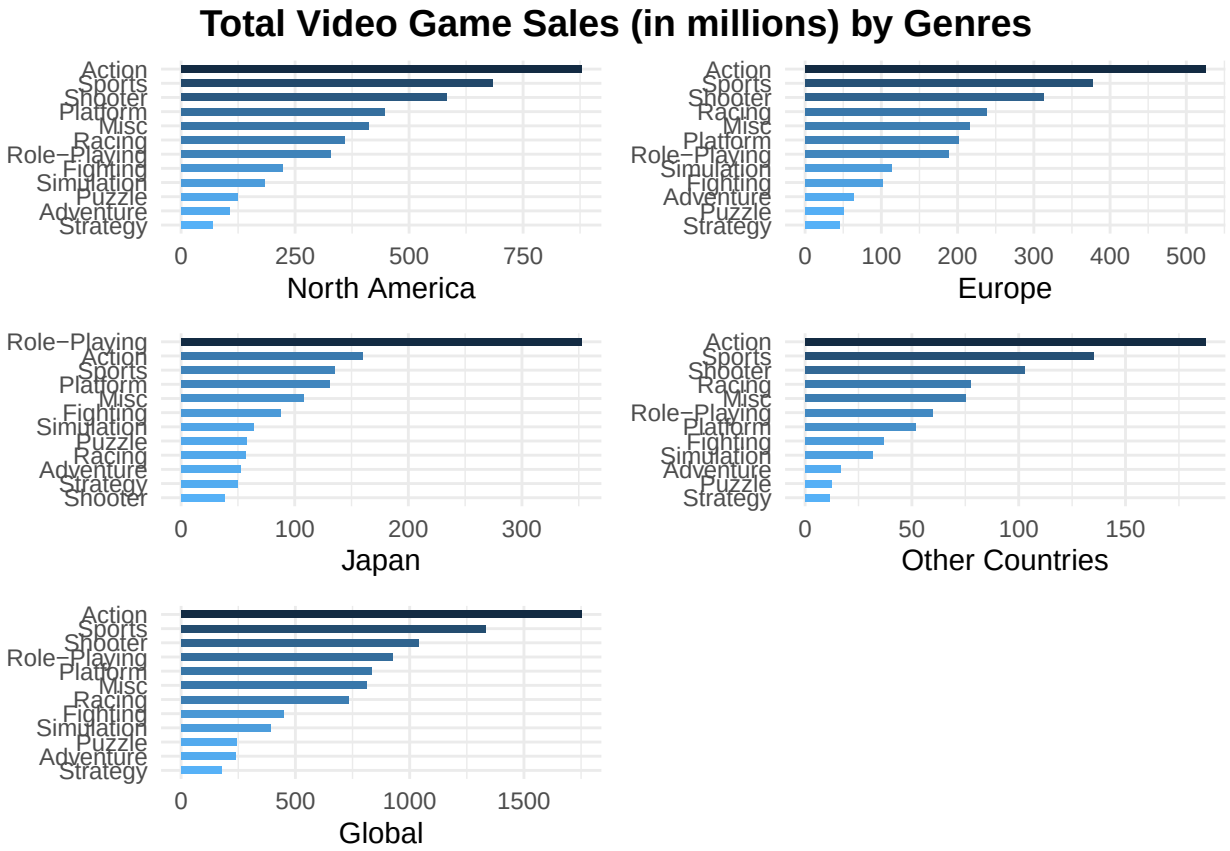
```
  annotate_figure(top = text_grob("Total Video Game Sales (in millions) by Released Years",
                                  color = "black",
                                  face = "bold",
                                  size = 14))
```



Per the plot above, regardless of countries, video games released in the 2005s, 2010s, and 2000s were bought more than those released in any other years.

Comparing Sales by Video Games Genres

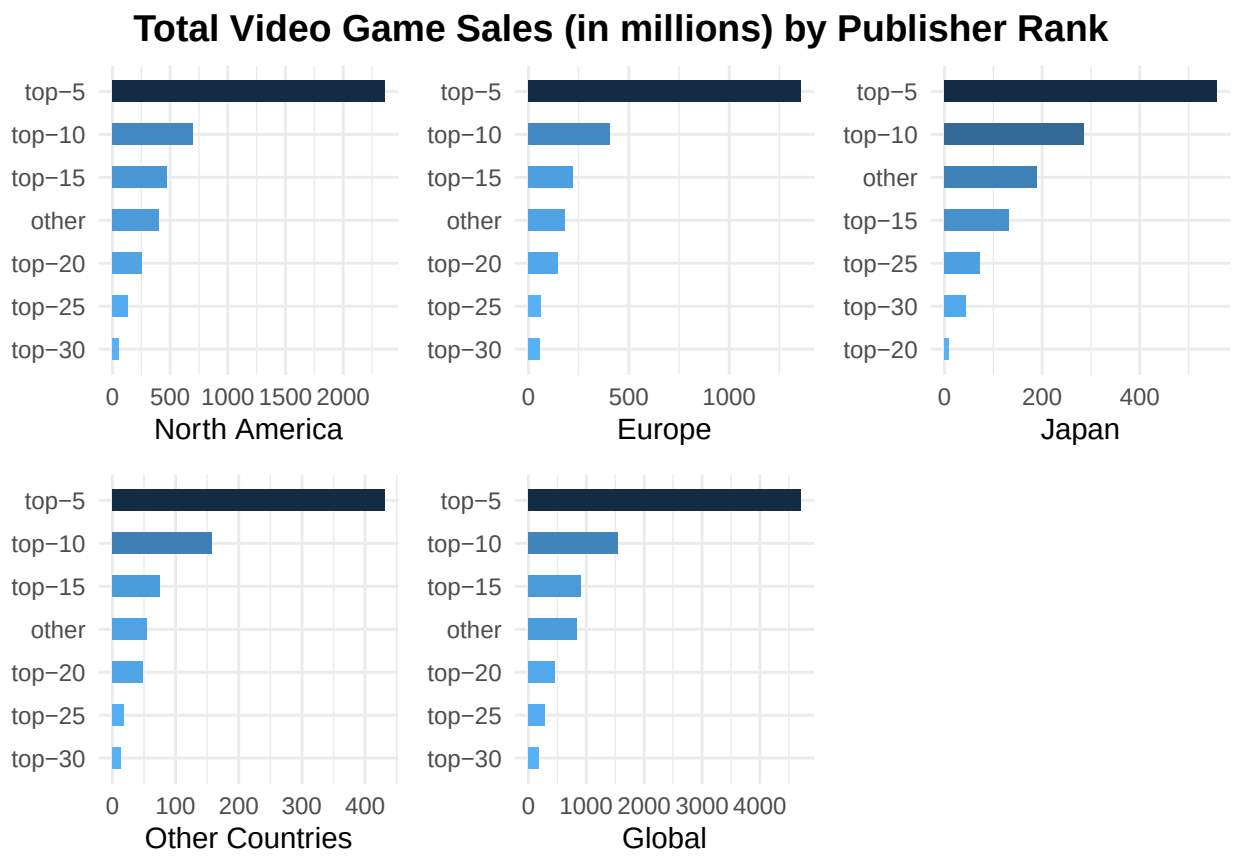
```
# --------------Genres ------------------
ggarrange(na.genres,
          eu.genres,
          jp.genres,
          other.genres,
          glo.genres,
          legend = NULL,
          nrow = 3,
          ncol = 2) %>%
  annotate_figure(top = text_grob("Total Video Game Sales (in millions) by Genres",
                                  color = "black",
                                  face = "bold",
                                  size = 14))
```



Per the figure above, there is a consistent pattern in Genres popularity. Action, Sport, and Shooter remained popular among all countries except Japan. Japan's number one genre is Role-Playing Games.

Comparing Sales by Publisher Rank

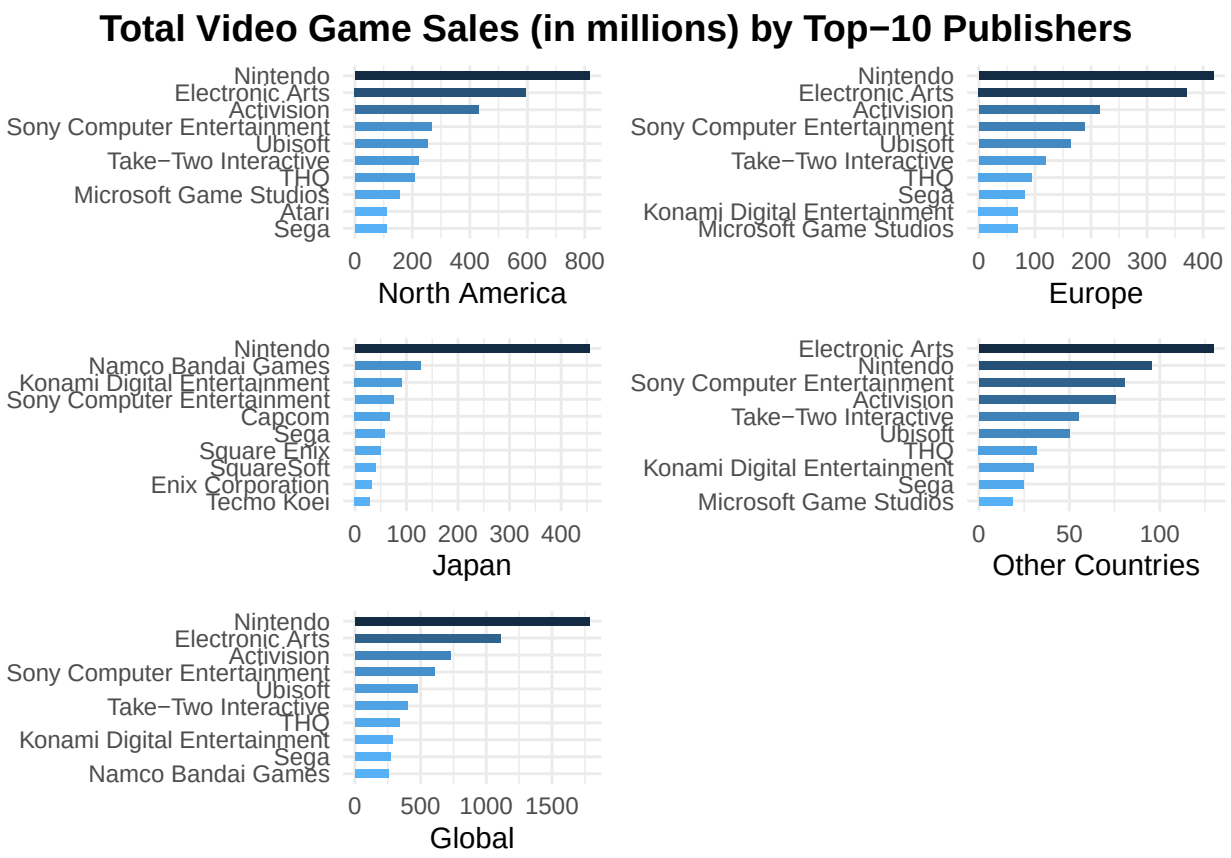
```
ggarrange(na.pub,
          eu.pub,
          jp.pub,
          other.pub,
          glo.pub,
          legend = NULL) %>%
  annotate_figure(top = text_grob("Total Video Game Sales (in millions) by Publisher Rank",
                                  color = "black",
                                  face = "bold",
                                  size = 14))
```



In regard to Publisher ranks, the pattern remain consistent across countries.

Comparing Sales by Top-10 Publishers

```
ggarrange(na.pus,
          eu.pus,
          jp.pus,
          other.pus,
          glo.pus,
          legend = NULL,
          ncol = 2,
          nrow = 3) %>%
  annotate_figure(top = text_grob("Total Video Game Sales (in millions) by Top-10 Publishers",
                                  color = "black",
                                  face = "bold",
                                  size = 14))
```



There is a difference in preference between countries Although Nintendo remains popular among all countries, Japan preferred their country's publisher more (e.g., Namco Bandai Games, Konami Digital Entertainment) while other countries prefers Electronic Arts.

Preparing Train & Test Set

To develop the algorithm, I will split the Original data set into two parts:

Train set: contains 70% of data from the original.

Test set: contains 30% of data from the original

Both the train & Test set are splitted by Global Sales. They only have the following variables:

- Predictors: Released Year, Publisher Rank, Genre, and Platform
- Outcome: NA sales, JP Sales, EU Sales, Other Countries Sales

I will develop different algorithm for each sales because global sales are just a sum of all sales. I came to this conclusion because due to the graph above, we can see a different patterns in sales from each countries using different predictors.

```
# Train index
train.index <- createDataPartition(orig$Global_Sales,
                                   times = 1,
                                   p = .7,
                                   list = FALSE)
```

```
# Train set
train.set <- orig %>%
  slice(train.index) %>%
  select(!c(Rank,
            Name,
            Year,
            Publisher)) %>%
  mutate(Platform = factor(Platform),
         Genre = factor(Genre),
         yr.released = factor(yr.released),
         publisher.rank = factor(publisher.rank))

# test set
test.set <- orig %>%
  slice(-train.index) %>%
  select(!c(Rank,
            Name,
            Year,
            Publisher)) %>%
  mutate(Platform = factor(Platform),
         Genre = factor(Genre),
         yr.released = factor(yr.released),
         publisher.rank = factor(publisher.rank))
```

It is necessary to check the level (of predictors) between the Train set and the test set because if the two sets have different levels in their predictors, the algorithm would not run.

Platforms

```
# predictors level Checking between train vs. test
levels(factor(train.set$Platform))
```

```
##  [1] "2600"        "3DS"         "DC"          "DS"          "GB"
##  [6] "GBA"         "GC"          "GEN"         "N64"         "NES"
## [11] "Not-Popular" "PC"          "PS"          "PS2"         "PS3"
## [16] "PS4"         "PSP"         "PSV"         "SAT"         "SNES"
## [21] "Wii"         "WiiU"        "X360"        "XB"          "XOne"
```

```
levels(factor(test.set$Platform))         # have same levels
```

```
##  [1] "2600"        "3DS"         "DC"          "DS"          "GB"
##  [6] "GBA"         "GC"          "GEN"         "N64"         "NES"
## [11] "Not-Popular" "PC"          "PS"          "PS2"         "PS3"
## [16] "PS4"         "PSP"         "PSV"         "SAT"         "SNES"
## [21] "Wii"         "WiiU"        "X360"        "XB"          "XOne"
```

Genres

```
levels(factor(train.set$Genre))
```

```
##  [1] "Action"       "Adventure"    "Fighting"     "Misc"         "Platform"
##  [6] "Puzzle"       "Racing"       "Role-Playing" "Shooter"      "Simulation"
## [11] "Sports"       "Strategy"
```

```
levels(factor(test.set$Genre))    # same levels
```

```
##  [1] "Action"       "Adventure"    "Fighting"     "Misc"         "Platform"
##  [6] "Puzzle"       "Racing"       "Role-Playing" "Shooter"      "Simulation"
## [11] "Sports"       "Strategy"
```

Released Year

```
levels(factor(train.set$yr.released))
```

```
## [1] "1980s"   "1985s"   "1990s"   "1995s"   "2000s"   "2005s"   "2010s"
## [8] "2015s"   "unknown"
```

```
levels(factor(test.set$yr.released))
```

```
## [1] "1980s"   "1985s"   "1990s"   "1995s"   "2000s"   "2005s"   "2010s"
## [8] "2015s"   "unknown"
```

Publisher Rank

```
levels(factor(train.set$publisher.rank))
```

```
## [1] "other"  "top-10" "top-15" "top-20" "top-25" "top-30" "top-5"
```

```
levels(factor(test.set$publisher.rank))
```

```
## [1] "other"  "top-10" "top-15" "top-20" "top-25" "top-30" "top-5"
```

All predictors between the two sets have the same levels. Proceed to developing algorithms.

The Algorithms

Two algorithm will be developed using different methods: **Generalized Linear Model (GLM) & Random Forest (RF).** The purpose of the algorithm is predicting scores (i.e., predicting a continuous value), so these two methods are suitable.

For Random Forest, the algorithm will perform calculation of about 50 trees (i.e., to avoid long computation time and it is recommended that trees are stable about 48 - 162), and the tree with the best outcome will be used as the algorithm. I will then compare the ***Root Mean Squared Error (RMSE)*** between the two algorithm and pick the one with the lowest RMSE.

This same procedure will applied for all four Sales: NA, EU, JP, and Other Countries.

North America

```
# GLM model
NA.fit_glm <- train(data = train.set,
                    NA_Sales~Platform + Genre + yr.released + publisher.rank,
                    method = "glm")

NA.y_hat_glm <- predict(NA.fit_glm,test.set)

# Random Forest Model
NA.train_forest <- train(data = train.set,
                         NA_Sales~Platform + Genre +
                       yr.released + publisher.rank,
                    method = "rf",
                    tuneGrid = data.frame(mtry = seq(1:7)),
                    ntree = 50)

NA.fit_forest <- randomForest(NA_Sales~Platform + Genre +
                                yr.released + publisher.rank,
                       data = train.set,
                       minNode = NA.train_forest$bestTune$mtry)
NA.y_hat_forest <- predict(NA.fit_forest,test.set)

# RMSE
data.frame(GLM = mean((NA.y_hat_glm - test.set$NA_Sales)^2),
           Random.Forest = mean((NA.y_hat_forest - test.set$NA_Sales)^2))
```

```
##     GLM Random.Forest
## 1 0.393          0.38
```

Europe

```
# GLM model
EU.fit_glm <- train(data = train.set,
                    EU_Sales~Platform + Genre + yr.released + publisher.rank,
                    method = "glm")

EU.y_hat_glm <- predict(EU.fit_glm,test.set)

# Random Forest Model
EU.train_forest <- train(data = train.set,
                         EU_Sales~Platform + Genre +
                           yr.released + publisher.rank,
                         method = "rf",
                         tuneGrid = data.frame(mtry = seq(1:7)),
                         ntree = 50)

EU.fit_forest <- randomForest(EU_Sales~Platform + Genre +
                                yr.released + publisher.rank,
                              data = train.set,
                              minNode = EU.train_forest$bestTune$mtry)
```

```
EU.y_hat_forest <- predict(EU.fit_forest,test.set)

# RMSE
data.frame(GLM = mean((EU.y_hat_glm - test.set$EU_Sales)^2),
           Random.Forest = mean((EU.y_hat_forest - test.set$EU_Sales)^2))
```

```
##     GLM Random.Forest
## 1 0.147         0.145
```

Japan

```
# GLM model
JP.fit_glm <- train(data = train.set,
                    JP_Sales~Platform + Genre + yr.released + publisher.rank,
                    method = "glm")

JP.y_hat_glm <- predict(JP.fit_glm,test.set)

# Random Forest Model
JP.train_forest <- train(data = train.set,
                         JP_Sales~Platform + Genre +
                           yr.released + publisher.rank,
                         method = "rf",
                         tuneGrid = data.frame(mtry = seq(1:7)),
                         ntree = 50)

JP.fit_forest <- randomForest(JP_Sales~Platform + Genre +
                                yr.released + publisher.rank,
                              data = train.set,
                              minNode = JP.train_forest$bestTune$mtry)

JP.y_hat_forest <- predict(JP.fit_forest,test.set)

# RMSE
data.frame(GLM = mean((JP.y_hat_glm - test.set$JP_Sales)^2),
           Random.Forest = mean((JP.y_hat_forest - test.set$JP_Sales)^2))
```

```
##      GLM Random.Forest
## 1 0.0676        0.0632
```

Other Countries

```
# GLM model
Other.fit_glm <- train(data = train.set,
                    Other_Sales~Platform + Genre + yr.released + publisher.rank,
                    method = "glm")

Other.y_hat_glm <- predict(Other.fit_glm,test.set)
```

```
# Random Forest Model
Other.train_forest <- train(data = train.set,
                            Other_Sales ~ Platform + Genre +
                              yr.released + publisher.rank,
                            method = "rf",
                            tuneGrid = data.frame(mtry = seq(1:7)),
                            ntree = 50)

Other.fit_forest <- randomForest(Other_Sales~Platform + Genre +
                                   yr.released + publisher.rank,
                                   data = train.set,
                                   minNode = Other.train_forest$bestTune$mtry)

Other.y_hat_forest <- predict(Other.fit_forest,
                                test.set)

# RMSE
data.frame(GLM = mean((Other.y_hat_glm - test.set$Other_Sales)^2),
           Random.Forest = mean((Other.y_hat_forest - test.set$Other_Sales)^2))
```

```
##      GLM Random.Forest
## 1 0.0164         0.016
```

Results.

Using GLM and Random Forest methods, two algorithms were established to predict sales by video games genres, publisher ranks, platforms, and released years for North America, Europe, Japan, and Other Countries. Comparing between the GLM and Random Forest method, Random Forest yields a lower RMSE in overall. Hence, **I will choose algorithm generated by Random Forest method.**

RMSE table for Random Forest Method

```
# RMSE Table: Random Forest Method
data.frame(North.America = mean((NA.y_hat_forest - test.set$NA_Sales)^2),
           Europe = mean((EU.y_hat_forest - test.set$EU_Sales)^2),
           Japan = mean((JP.y_hat_forest - test.set$JP_Sales)^2),
           Other.Countries = mean((Other.y_hat_forest - test.set$Other_Sales)^2))
```

```
##   North.America Europe  Japan Other.Countries
## 1          0.38  0.145 0.0632           0.016
```

RMSE table for GLM method

```
# RMSE Table: GLM Method
data.frame(North.America = mean((NA.y_hat_glm - test.set$NA_Sales)^2),
           Europe = mean((EU.y_hat_glm - test.set$EU_Sales)^2),
           Japan = mean((JP.y_hat_glm - test.set$JP_Sales)^2),
           Other.Countries = mean((Other.y_hat_glm - test.set$Other_Sales)^2))
```

```
##   North.America Europe  Japan Other.Countries
## 1         0.393  0.147 0.0676          0.0164
```

Conclusion

In this project, I utilize Machine Learning concepts of GLM and Random Forest to generate algorithms to predict sales by the following predictors: Genres, Publisher Rank, Platforms, and Released Year in North America, Europe, Japan, and Other Countries. The RMSE I acquired with my algorithms are lower than .15 in all countries, except America.

Some limitations with my algorithms lie in the assumptions I made about the dataset. With this method, I did not factor in population differences between the countries. It is possible that Japan has lower sales in video games comparing to other countries can be due to massive difference in population size. Another potential factor that can influence video games sales would be access to the market. Some consoles were established in Japan first while others may first be populated in Western countries.

As such, it is highly recommended that future algorithm developers take into account these geographic and demographic differences into the dataset so that a more thorough approach can be achieved.

References

(2016). Video Game Sales Dataset (Version 2). Kaggle. Retrieved May 1st, 2023 from https://www.kaggle.com/datasets/gregorut/videogamesales

Irizarry, R. (n.d.). *Introduction to data science*. rafalab. Retrieved April 29, 2023, from https://rafalab.github.io/dsbook/