Video Game Sales Analysis

Savas turkoglu

5/18/2020

1- Introduction

- This data science project created for Harvard Data Science Certification program at Edx by Savas Turkoglu

We will analysis and visualizate dataset that about video game sales arround the world over years

The video game industry is growing so fast that some believe it will reach over $300 billion by 2025. With billions of dollars in profit and over 2.5 billion gamers around the world, we can expect video game platforms to continue developing in 2020. Besides the consistent and impressive growth of the industry, it is interesting to note that there has been a shift in revenue sources in the gaming space lately. The gaming industry used to make most of its money by selling games but today its revenue is coming from a different perspective.

source: Forbes https://www.forbes.com/sites/ilkerkoksal/2019/11/08/video-gaming-industry--its-revenue-shift/#8569649663e5

a- data set

I get data from kaggle https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings We have more than 16000 row data about video game industry that contain can give us an idea about sales such as -> platform,genre, publisher, rating, Name, sales by region (NA_sale, EU_Sales...) etc.. names ->“Name” “Platform” “Year_of_Release” “Genre” “Publisher” “NA_Sales”“EU_Sales” “JP_Sales” “Other_Sales” “Global_Sales” "Rating but there are many missing data in the dataset we have to deal with this missing data every columns , even ve can drop some columns if thre are a lot of gaps wee’ll visualie this data for get some idea about this industry. wee’ww try estimate sale performance

load libraryes Loading packages for data exploration, visualization, preprocessing,

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")

if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(plotly)) install.packages("plotly", repos = "http://cran.us.r-project.org")

if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(kernlab)) install.packages("kernlab", repos = "http://cran.us.r-project.org")
```

```
#ml
library(randomForest)
library(kernlab)
library(caret)
```

```r
#data exploration
library(dplyr)
library(tidyverse)
#plot
library(data.table)
library(ggplot2)
library(knitr)
```

```r
knitr::opts_chunk$set(
  echo = TRUE,
  message = FALSE,
  warning = FALSE
)
```

1- a Data overview

data from kaggle https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings Context Motivated by Gregory Smith's web scrape of VGChartz Video Games Sales, this data set simply extends the number of variables with another web scrape from Metacritic. Unfortunately, there are missing observations as Metacritic only covers a subset of the platforms. Also, a game may not have all the observations of the additional variables discussed below. Complete cases are ~ 6,900

Load data from external source

```r
url<- 'https://likyapix.com/game-data.csv'

data <- read.csv(url)
```

take a look data

```r
head(data)
```

```
##                       Name Platform Year_of_Release        Genre Publisher
## 1               Wii Sports      Wii            2006       Sports  Nintendo
## 2        Super Mario Bros.      NES            1985     Platform  Nintendo
## 3           Mario Kart Wii      Wii            2008       Racing  Nintendo
## 4        Wii Sports Resort      Wii            2009       Sports  Nintendo
## 5 Pokemon Red/Pokemon Blue       GB            1996 Role-Playing  Nintendo
## 6                  Tetris       GB            1989       Puzzle  Nintendo
##   NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales Critic_Score Critic_Count
## 1    41.36    28.96     3.77        8.45        82.53           76           51
## 2    29.08     3.58     6.81        0.77        40.24           NA           NA
## 3    15.68    12.76     3.79        3.29        35.52           82           73
## 4    15.61    10.93     3.28        2.95        32.77           80           73
## 5    11.27     8.89    10.22        1.00        31.37           NA           NA
## 6    23.20     2.26     4.22        0.58        30.26           NA           NA
##   User_Score User_Count Developer Rating
## 1          8        322  Nintendo      E
## 2                    NA
## 3        8.3        709  Nintendo      E
```

```
## 4          8        192  Nintendo      E
## 5                    NA
## 6                    NA
```

```
#names
names(data)
```

```
##  [1] "Name"            "Platform"        "Year_of_Release" "Genre"
##  [5] "Publisher"       "NA_Sales"        "EU_Sales"        "JP_Sales"
##  [9] "Other_Sales"     "Global_Sales"    "Critic_Score"    "Critic_Count"
## [13] "User_Score"      "User_Count"      "Developer"       "Rating"
```

```
#summary
 summary(data)
```

```
##                           Name          Platform    Year_of_Release
##  Need for Speed: Most Wanted:   12   PS2    :2161   2008   :1427
##  FIFA 14                    :    9   DS     :2152   2009   :1426
##  LEGO Marvel Super Heroes   :    9   PS3    :1331   2010   :1255
##  Madden NFL 07              :    9   Wii    :1320   2007   :1197
##  Ratatouille                :    9   X360   :1262   2011   :1136
##  Angry Birds Star Wars      :    8   PSP    :1209   2006   :1006
##  (Other)                    :16663   (Other):7284   (Other):9272
##           Genre                             Publisher        NA_Sales
##  Action      :3370   Electronic Arts             : 1356   Min.   : 0.0000
##  Sports      :2348   Activision                  :  985   1st Qu.: 0.0000
##  Misc        :1750   Namco Bandai Games          :  939   Median : 0.0800
##  Role-Playing:1500   Ubisoft                     :  933   Mean   : 0.2633
##  Shooter     :1323   Konami Digital Entertainment:  834   3rd Qu.: 0.2400
##  Adventure   :1303   THQ                         :  715   Max.   :41.3600
##  (Other)     :5125   (Other)                     :10957
##     EU_Sales         JP_Sales        Other_Sales        Global_Sales
##  Min.   : 0.000   Min.   : 0.0000   Min.   : 0.00000   Min.   : 0.0100
##  1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.00000   1st Qu.: 0.0600
##  Median : 0.020   Median : 0.0000   Median : 0.01000   Median : 0.1700
##  Mean   : 0.145   Mean   : 0.0776   Mean   : 0.04733   Mean   : 0.5335
##  3rd Qu.: 0.110   3rd Qu.: 0.0400   3rd Qu.: 0.03000   3rd Qu.: 0.4700
##  Max.   :28.960   Max.   :10.2200   Max.   :10.57000   Max.   :82.5300
##
##   Critic_Score    Critic_Count      User_Score     User_Count
##  Min.   :13.00   Min.   :  3.00          :6704   Min.   :    4.0
##  1st Qu.:60.00   1st Qu.: 12.00   tbd    :2425   1st Qu.:   10.0
##  Median :71.00   Median : 21.00   7.8    : 324   Median :   24.0
##  Mean   :68.97   Mean   : 26.36   8      : 290   Mean   :  162.2
##  3rd Qu.:79.00   3rd Qu.: 36.00   8.2    : 282   3rd Qu.:   81.0
##  Max.   :98.00   Max.   :113.00   8.3    : 254   Max.   :10665.0
##  NA's   :8582    NA's   :8582     (Other):6440   NA's   :9129
##      Developer        Rating
##           :6623          :6769
##  Ubisoft  : 204   E      :3991
##  EA Sports: 172   T      :2961
##  EA Canada: 167   M      :1563
##  Konami   : 162   E10+   :1420
```

```
##  Capcom   : 139   EC     :   8
##  (Other)  :9252   (Other):   7
```

```
# chack duplicarion
duplicated(data) %>%sum()
```

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

```
#dimensions
dim(data)
```

```
## [1] 16719    16
```

Data content

Alongside the fields: Name, Platform, YearofRelease, Genre, Publisher, NASales, EUSales, JPSales, Other-Sales, Global_Sales, Rating - The ESRB ratings Acknowledgements

check missing mavlue

```
sapply(data, function(x) sum(is.na(x)))
```

```
##            Name        Platform Year_of_Release           Genre       Publisher
##               0               0               0               0               0
##        NA_Sales        EU_Sales        JP_Sales     Other_Sales    Global_Sales
##               0               0               0               0               0
##    Critic_Score    Critic_Count      User_Score      User_Count       Developer
##            8582            8582               0            9129               0
##          Rating
##               0
```

check empty values

```
sapply(data, function(x) sum(x==''))
```

```
##            Name        Platform Year_of_Release           Genre       Publisher
##               2               0               0               2               0
##        NA_Sales        EU_Sales        JP_Sales     Other_Sales    Global_Sales
##               0               0               0               0               0
##    Critic_Score    Critic_Count      User_Score      User_Count       Developer
##              NA              NA            6704              NA            6623
##          Rating
##            6769
```

As we can see there are many missing data in columns such as Critical_count, Critical_Score, User_Score, User_Count ,Developer columns and this missing datas more than half of dataset and will not give us an idea about dataset. Therefore we will ignore tihs columns during analysis an visualization bu we handle this columns on prediction.

```
data <- data %>% filter( as.numeric(Year_of_Release) < 2019)
```

2 Analysis and predictin

2-a Analysis and visualize data

Platform

first we'll look at platform column There are several popular video game plarform such as Nintendo, PS, XOne Among video game lovers and there is hard competiton between these companies let's look up

unique(data$Platform) there are more than 20 differen game platform in the dataset

```
platform_ <- data %>% group_by(Platform) %>%
  summarize(count       = n(),
            Global_sales = sum(Global_Sales),
            NA_Sales     = sum(NA_Sales),
            EU_Sales     = sum(EU_Sales),
            JP_Sales     = sum(JP_Sales),
            )
platform_ %>% arrange(desc(Global_sales))%>% knitr::kable()
```

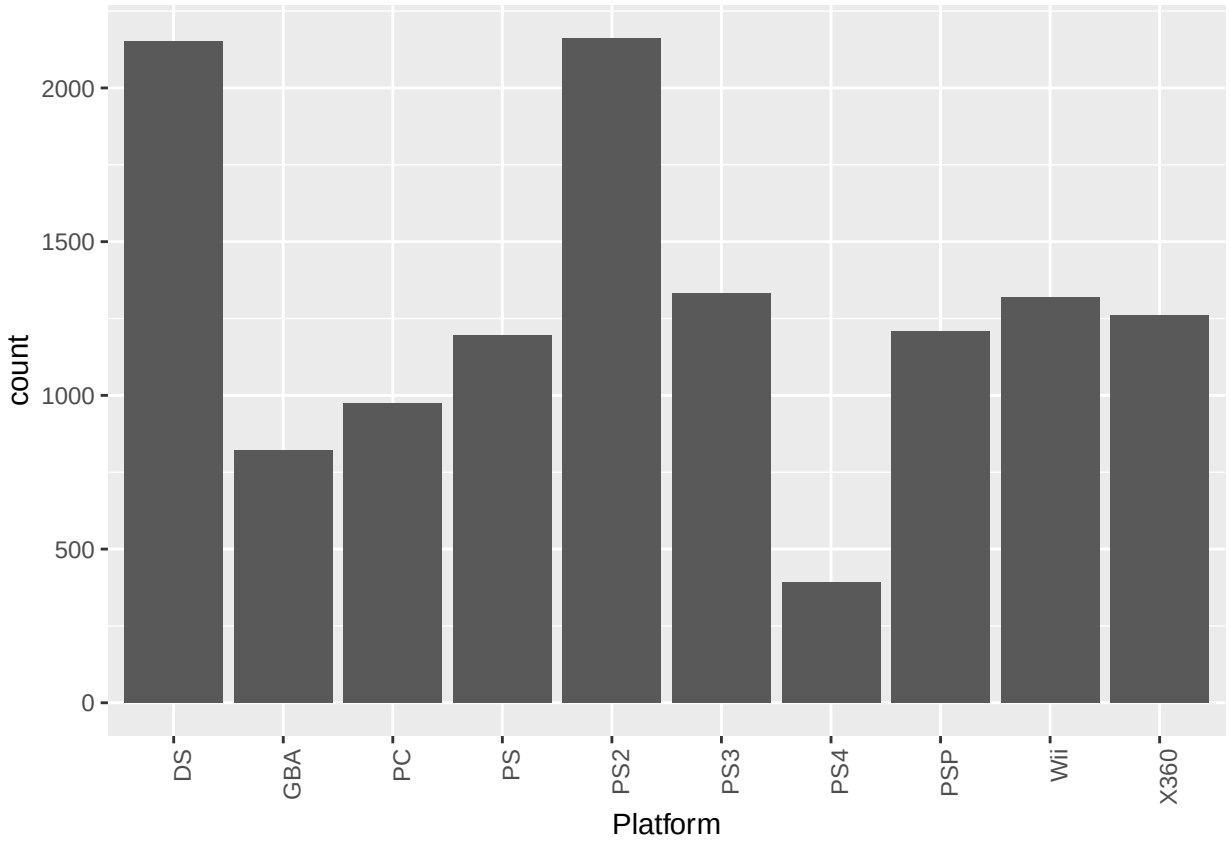
Platform	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
PS2	2161	1255.64	583.84	339.29	139.20
X360	1262	971.63	602.47	270.76	12.43
PS3	1331	939.43	393.49	330.29	80.19
Wii	1320	908.13	496.90	262.21	69.33
DS	2152	807.10	382.67	188.89	175.57
PS	1197	730.68	336.52	213.61	139.82
GBA	822	318.50	187.54	75.25	47.33
PS4	393	314.23	108.74	141.09	16.00
PSP	1209	294.30	109.17	66.68	76.78
PC	974	260.30	94.53	142.44	0.17
3DS	520	259.09	83.49	61.48	100.67
XB	824	258.26	186.69	60.95	1.38
GB	98	255.45	114.32	47.82	85.12
NES	98	251.07	125.94	21.15	98.65
N64	319	218.88	139.02	41.06	34.22
SNES	239	200.05	61.23	19.04	116.55
GC	556	199.36	133.46	38.71	21.58
XOne	247	159.44	93.12	51.59	0.34
2600	133	97.08	90.60	5.47	0.00
WiiU	147	82.16	38.10	25.13	13.01
PSV	432	54.12	12.58	13.12	21.93
SAT	173	33.59	0.72	0.54	32.26
GEN	29	30.78	21.05	6.05	2.70
DC	52	15.97	5.43	1.69	8.56
SCD	6	1.87	1.00	0.36	0.45
NG	12	1.44	0.00	0.00	1.44
WS	6	1.42	0.00	0.00	1.42
TG16	2	0.16	0.00	0.00	0.16

Platform	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
3DO	3	0.10	0.00	0.00	0.10
GG	1	0.04	0.00	0.00	0.04
PCFX	1	0.03	0.00	0.00	0.03

as expected PlayStation Series at the top, X360 and Nintendo following

top 10 platform

```
platform_ %>% arrange(desc(Global_sales)) %>%
  head(10) %>%
  ggplot(aes(x=Platform, y=count)) + geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



look up all platform at table

```
platform_ %>% arrange(desc(Global_sales))%>% knitr::kable()
```

Platform	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
PS2	2161	1255.64	583.84	339.29	139.20
X360	1262	971.63	602.47	270.76	12.43
PS3	1331	939.43	393.49	330.29	80.19
Wii	1320	908.13	496.90	262.21	69.33
DS	2152	807.10	382.67	188.89	175.57

Platform	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
PS	1197	730.68	336.52	213.61	139.82
GBA	822	318.50	187.54	75.25	47.33
PS4	393	314.23	108.74	141.09	16.00
PSP	1209	294.30	109.17	66.68	76.78
PC	974	260.30	94.53	142.44	0.17
3DS	520	259.09	83.49	61.48	100.67
XB	824	258.26	186.69	60.95	1.38
GB	98	255.45	114.32	47.82	85.12
NES	98	251.07	125.94	21.15	98.65
N64	319	218.88	139.02	41.06	34.22
SNES	239	200.05	61.23	19.04	116.55
GC	556	199.36	133.46	38.71	21.58
XOne	247	159.44	93.12	51.59	0.34
2600	133	97.08	90.60	5.47	0.00
WiiU	147	82.16	38.10	25.13	13.01
PSV	432	54.12	12.58	13.12	21.93
SAT	173	33.59	0.72	0.54	32.26
GEN	29	30.78	21.05	6.05	2.70
DC	52	15.97	5.43	1.69	8.56
SCD	6	1.87	1.00	0.36	0.45
NG	12	1.44	0.00	0.00	1.44
WS	6	1.42	0.00	0.00	1.42
TG16	2	0.16	0.00	0.00	0.16
3DO	3	0.10	0.00	0.00	0.10
GG	1	0.04	0.00	0.00	0.04
PCFX	1	0.03	0.00	0.00	0.03

Sales

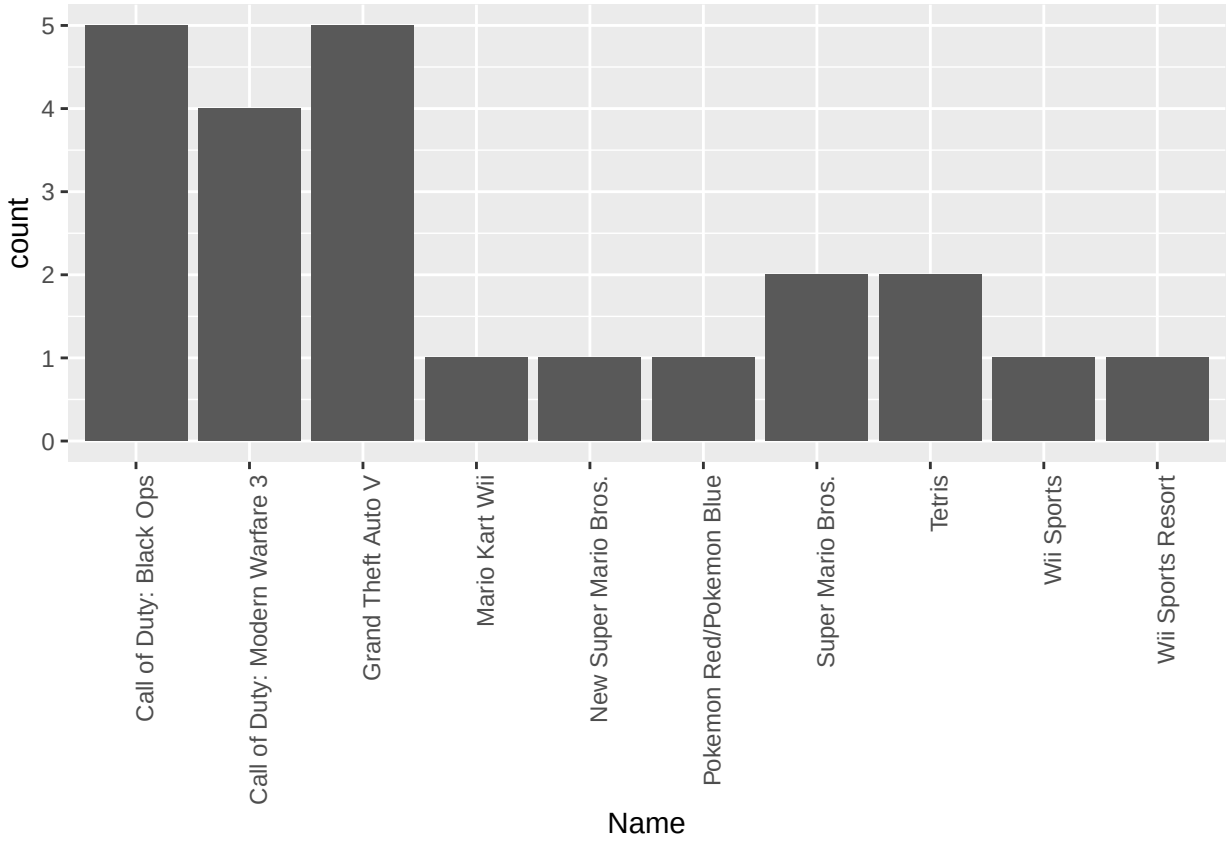
```
game_ <- data %>% group_by(Name) %>%
             summarize(count       = n(),
            Global_sales = sum(Global_Sales),
            NA_Sales     = sum(NA_Sales),
            EU_Sales    = sum(EU_Sales),
            JP_Sales    = sum(JP_Sales),
  )
```

```
### ------  global sale by game
 game_global_ <- game_ %>% arrange(desc(Global_sales)) %>% head(10)

 # over view global sales
 game_global_ %>%  knitr::kable()
```

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Wii Sports	1	82.53	41.36	28.96	3.77
Grand Theft Auto V	5	56.57	23.84	23.42	1.42
Super Mario Bros.	2	45.31	32.48	4.88	6.96
Tetris	2	35.84	26.17	2.95	6.03
Mario Kart Wii	1	35.52	15.68	12.76	3.79

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Wii Sports Resort	1	32.77	15.61	10.93	3.28
Pokemon Red/Pokemon Blue	1	31.37	11.27	8.89	10.22
Call of Duty: Black Ops	5	30.82	17.57	9.35	0.59
Call of Duty: Modern Warfare 3	4	30.59	15.54	11.15	0.62
New Super Mario Bros.	1	29.80	11.28	9.14	6.50

```
# top 10 global sale
game_global_ %>% head(10) %>% ggplot(aes(x=Name, y=count)) + geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

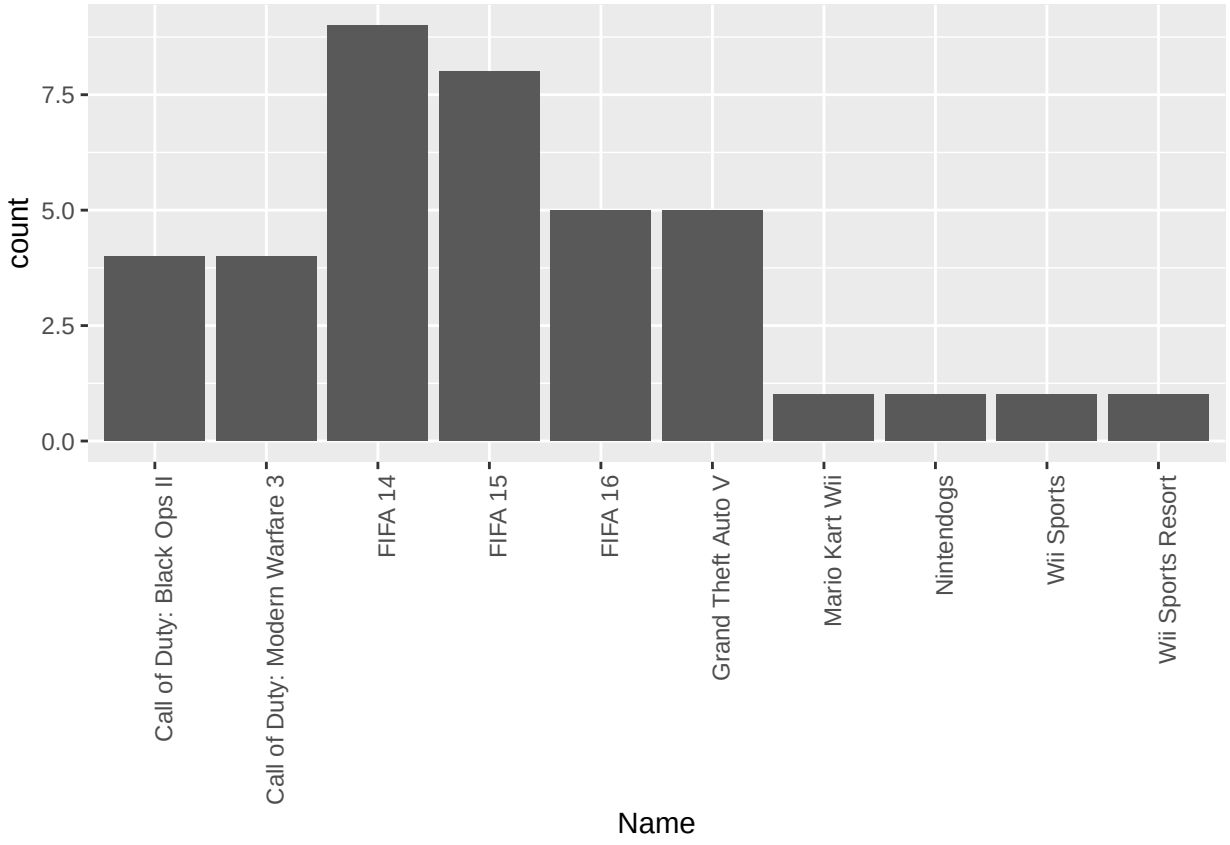



```
### ------  EU sale by game
game_eu_ <- game_ %>% arrange(desc(EU_Sales)) %>% head(10)

# over view global sales
game_eu_ %>%  knitr::kable()
```

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Wii Sports	1	82.53	41.36	28.96	3.77
Grand Theft Auto V	5	56.57	23.84	23.42	1.42
Mario Kart Wii	1	35.52	15.68	12.76	3.79
FIFA 15	8	17.34	3.09	12.02	0.14
Call of Duty: Modern Warfare 3	4	30.59	15.54	11.15	0.62

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
FIFA 16	5	16.30	3.05	11.09	0.11
FIFA 14	9	16.48	2.81	10.96	0.20
Nintendogs	1	24.67	9.05	10.95	1.93
Wii Sports Resort	1	32.77	15.61	10.93	3.28
Call of Duty: Black Ops II	4	29.40	14.08	10.84	0.72

```
# top 10 global sale
game_eu_ %>% head(10) %>% ggplot(aes(x=Name, y=count)) + geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

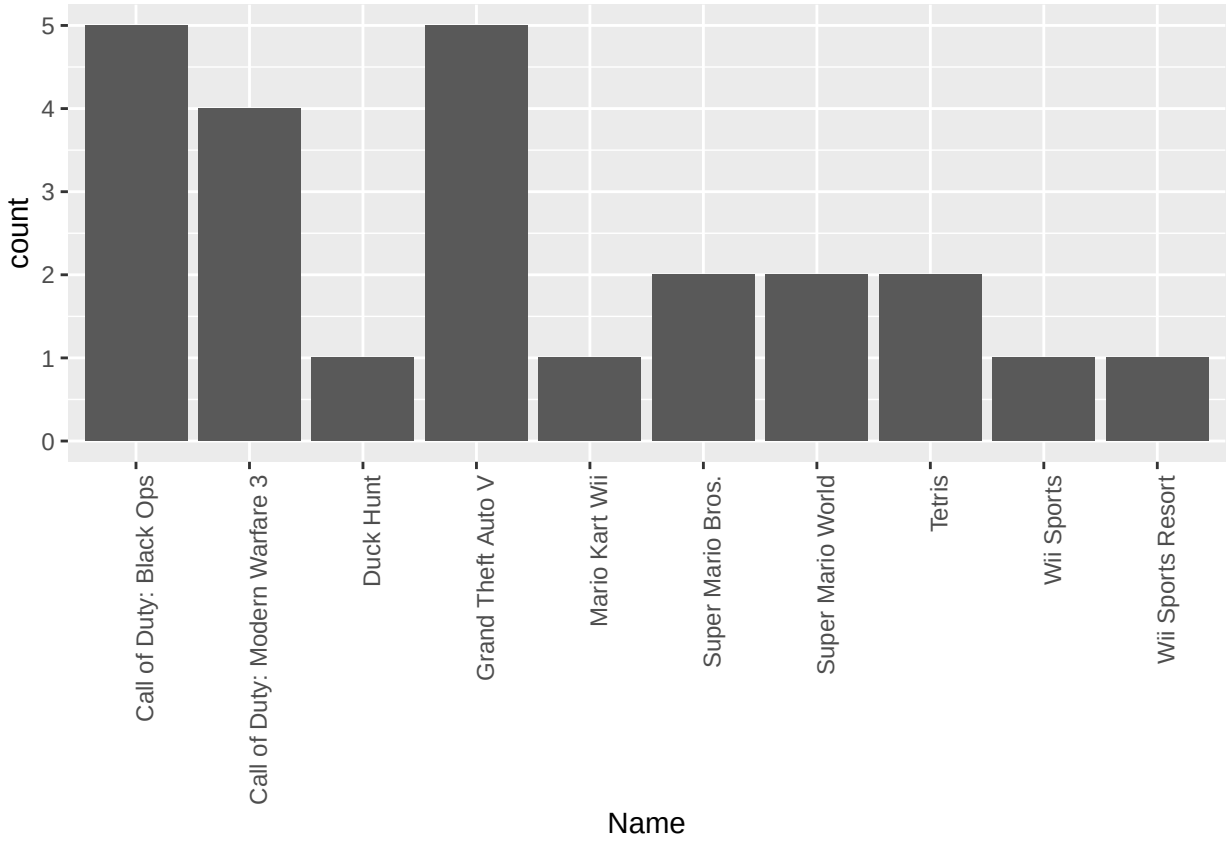



```
### ------  NA sale by game
game_na_ <- game_ %>% arrange(desc(NA_Sales)) %>% head(10)

# over view global sales
game_na_ %>%  knitr::kable()
```

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Wii Sports	1	82.53	41.36	28.96	3.77
Super Mario Bros.	2	45.31	32.48	4.88	6.96
Duck Hunt	1	28.31	26.93	0.63	0.28
Tetris	2	35.84	26.17	2.95	6.03
Grand Theft Auto V	5	56.57	23.84	23.42	1.42

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Call of Duty: Black Ops	5	30.82	17.57	9.35	0.59
Super Mario World	2	26.07	15.99	4.86	4.49
Mario Kart Wii	1	35.52	15.68	12.76	3.79
Wii Sports Resort	1	32.77	15.61	10.93	3.28
Call of Duty: Modern Warfare 3	4	30.59	15.54	11.15	0.62

```
# top 10 global sale
game_na_ %>% head(10) %>% ggplot(aes(x=Name, y=count)) + geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

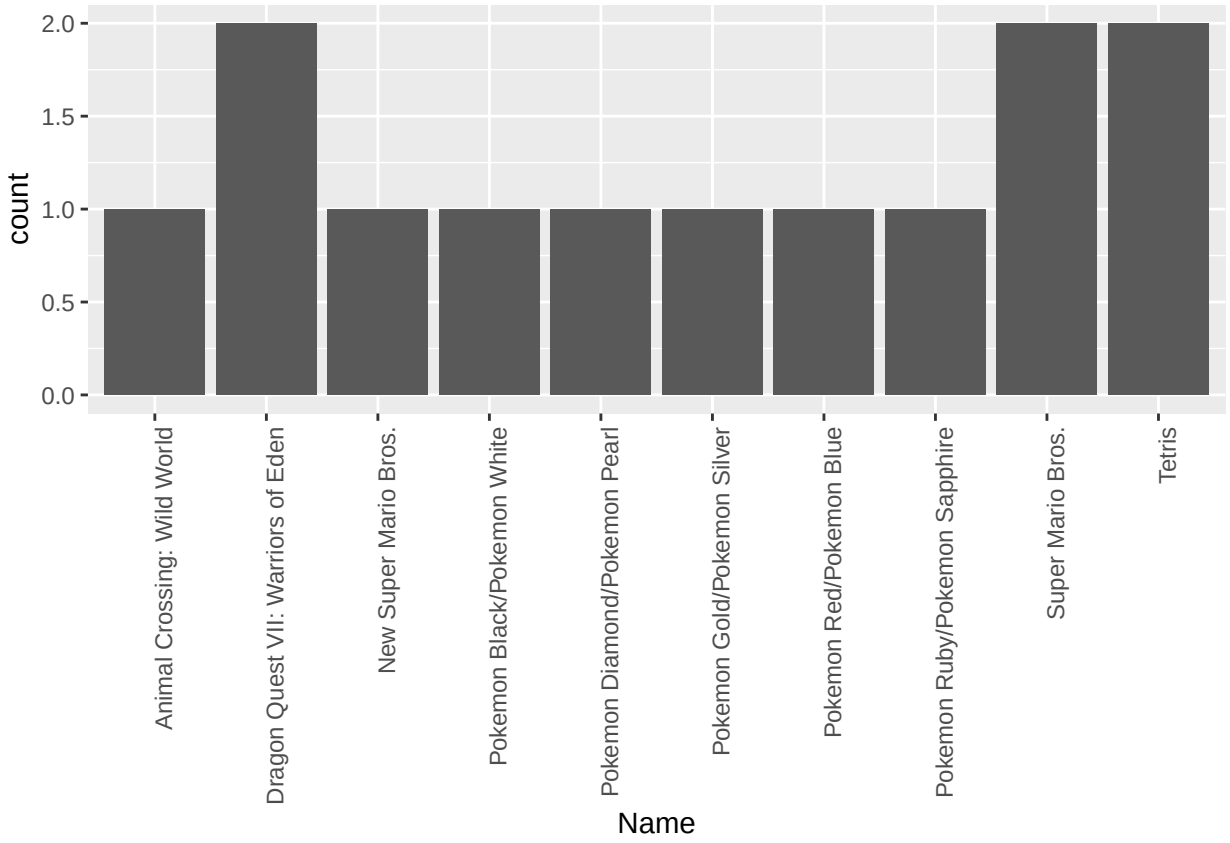



```
### ------  JP sale by game
game_jp_ <- game_ %>% arrange(desc(JP_Sales)) %>% head(10)

# over view global sales
game_jp_ %>%  knitr::kable()
```

Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Pokemon Red/Pokemon Blue	1	31.37	11.27	8.89	10.22
Pokemon Gold/Pokemon Silver	1	23.10	9.00	6.18	7.20
Super Mario Bros.	2	45.31	32.48	4.88	6.96
New Super Mario Bros.	1	29.80	11.28	9.14	6.50
Pokemon Diamond/Pokemon Pearl	1	18.25	6.38	4.46	6.04

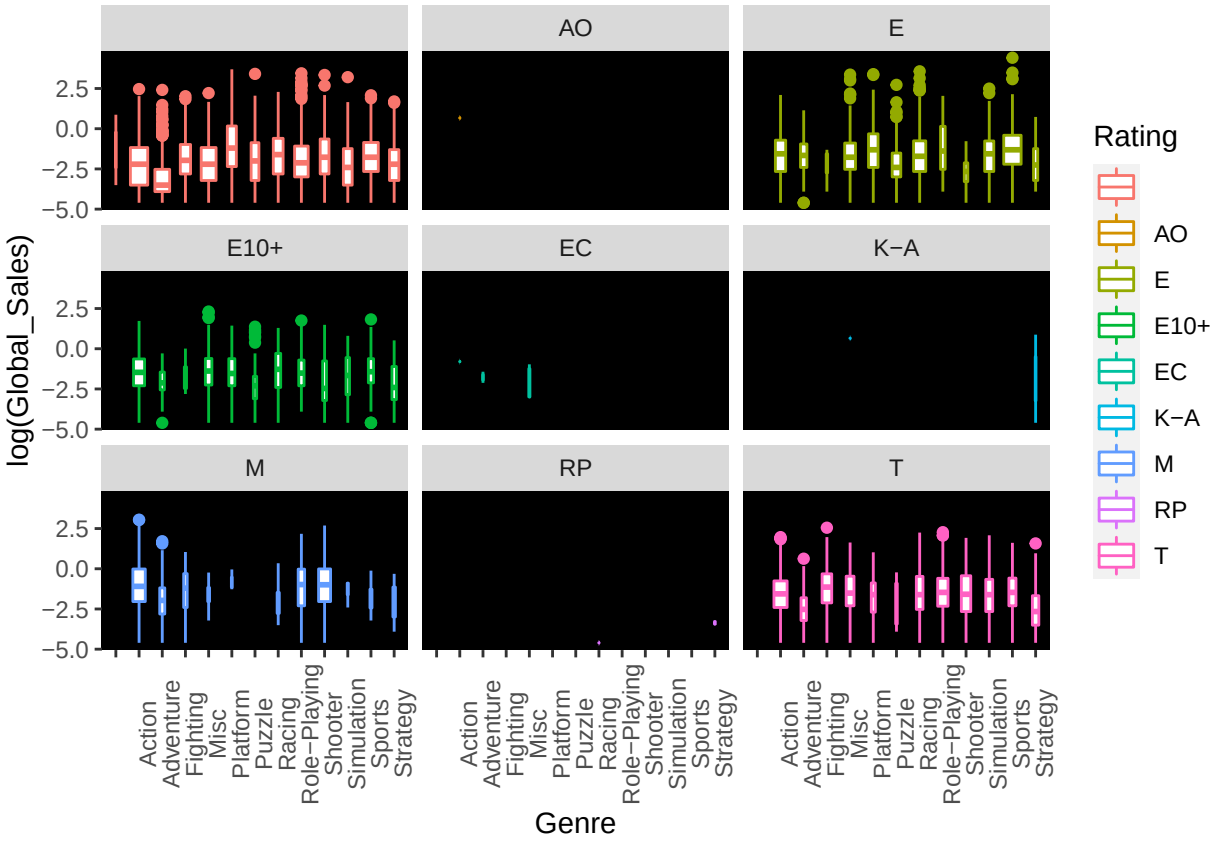
Name	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Tetris	2	35.84	26.17	2.95	6.03
Pokemon Black/Pokemon White	1	15.14	5.51	3.17	5.65
Dragon Quest VII: Warriors of Eden	2	5.93	0.26	0.23	5.40
Pokemon Ruby/Pokemon Sapphire	1	15.85	6.06	3.90	5.38
Animal Crossing: Wild World	1	12.13	2.50	3.45	5.33

```
# top 10 global sale
game_jp_ %>% head(10) %>% ggplot(aes(x=Name, y=count)) + geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Distribution of Global Sales across Genres and Rating

```
data %>%
  ggplot(aes(x=Genre,y=log(Global_Sales),col=Rating))+
  geom_boxplot(varwidth=TRUE)+facet_wrap(~Rating)+
  theme(axis.text.x=element_text(angle=90),panel.background = element_rect(fill="black"), panel.grid.ma
  panel.grid.minor=element_blank())
```



Genres

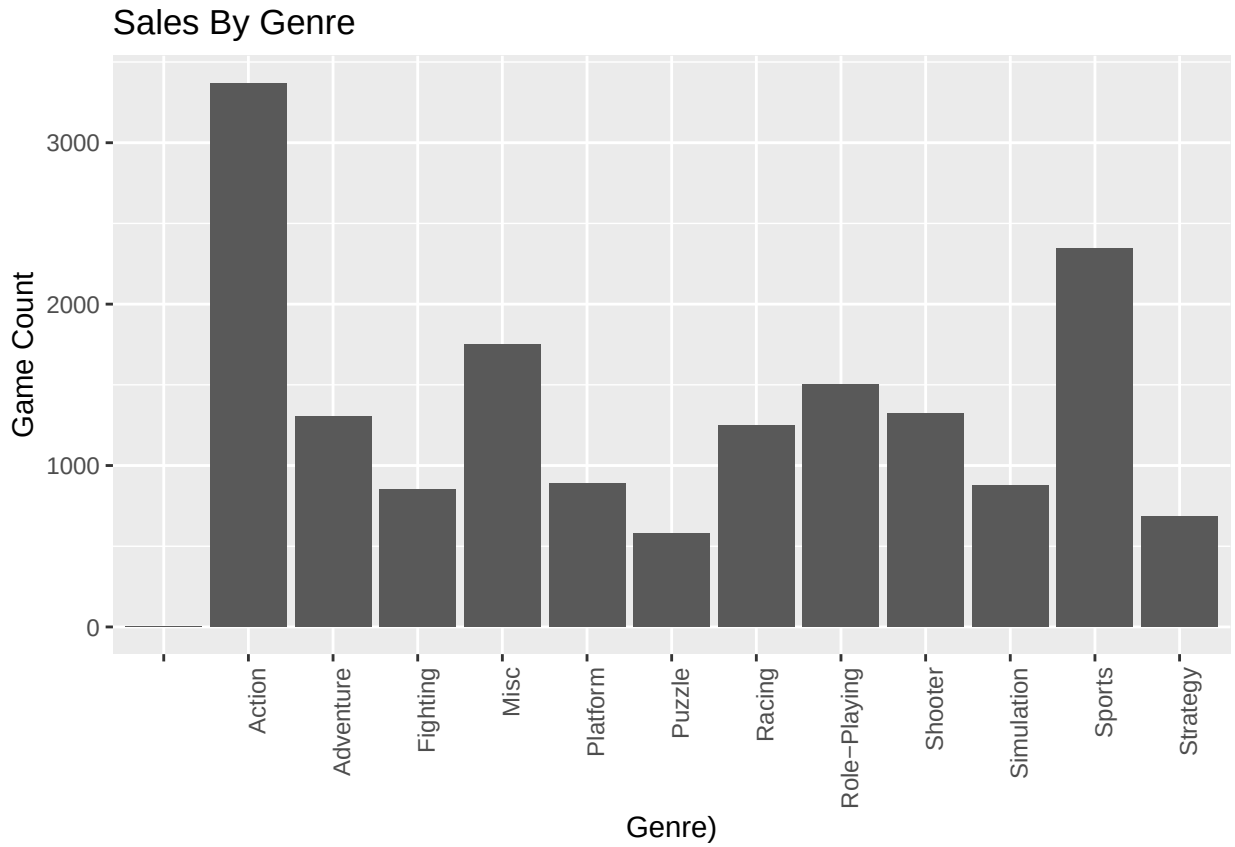
```
genre_ <- data %>% group_by(Genre) %>%
  summarize(count       = n(),
            Global_sales = sum(Global_Sales),
            NA_Sales     = sum(NA_Sales),
            EU_Sales     = sum(EU_Sales),
            JP_Sales     = sum(JP_Sales),
  )

genre_ %>% head()
```

```
## # A tibble: 6 x 6
##   Genre       count Global_sales NA_Sales EU_Sales JP_Sales
##   <fct>       <int>        <dbl>    <dbl>    <dbl>    <dbl>
## 1 ""              2         2.42     1.78     0.53     0.03
## 2 "Action"     3370      1745.     879.     519.     161.
## 3 "Adventure"  1303       238.     105.      63.5     52.3
## 4 "Fighting"    849       447.     223.     100.      87.5
## 5 "Misc"       1750       803.     407.     213.     108.
## 6 "Platform"    888       828.     446.     200.     131.
```

```
genre_  %>% arrange(desc(Global_sales)) %>% head(10) %>%  knitr::kable()
```

Genre	count	Global_sales	NA_Sales	EU_Sales	JP_Sales
Action	3370	1745.27	879.01	519.13	161.44
Sports	2348	1332.00	684.43	376.79	135.54
Shooter	1323	1052.94	592.24	317.34	38.76
Role-Playing	1500	934.40	330.81	188.71	355.46
Platform	888	828.08	445.50	200.35	130.83
Misc	1750	803.18	407.27	212.74	108.11
Racing	1249	728.90	359.35	236.51	56.71
Fighting	849	447.48	223.36	100.33	87.48
Simulation	874	390.42	182.19	113.52	63.80
Puzzle	580	243.02	122.87	50.01	57.31

```
genre_ %>% ggplot(aes(x=Genre, y=count)) + geom_bar(stat = "identity") +
xlab("Genre)") + ylab("Game Count") + ggtitle('Sales By Genre') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Publisher

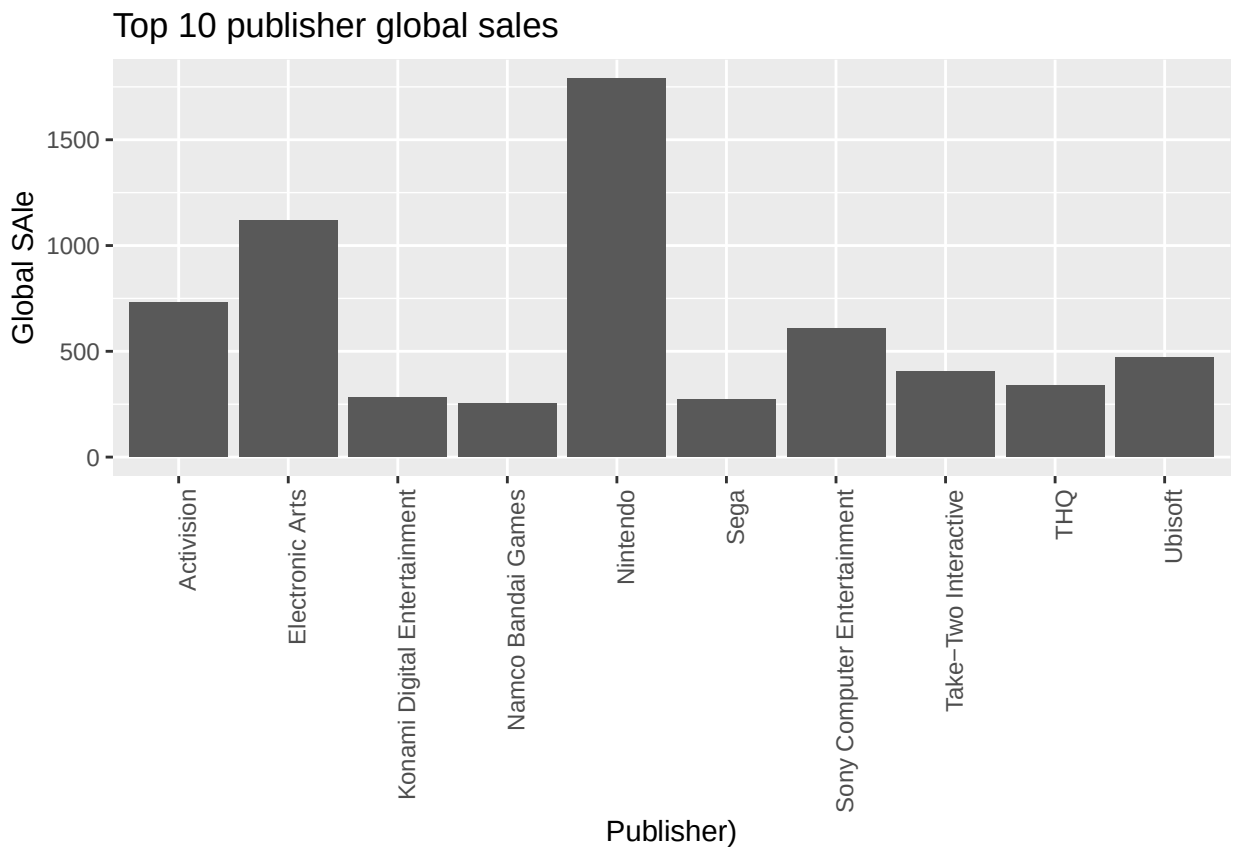
```
publisher_ <- data %>% group_by(Publisher)%>% filter(!is.na(Publisher)) %>%
  summarize(count       = n(),
            Global_sales = sum(Global_Sales),
            NA_Sales     = sum(NA_Sales),
            EU_Sales    = sum(EU_Sales),
            JP_Sales    = sum(JP_Sales),
  )
```

top 10 publisher globaly

```
publisher__ <- publisher_ %>% arrange(desc(Global_sales))  %>% head(10)

publisher__ %>% head()
```

```
## # A tibble: 6 x 6
##   Publisher                    count Global_sales NA_Sales EU_Sales JP_Sales
##   <fct>                        <int>        <dbl>    <dbl>    <dbl>    <dbl>
## 1 Nintendo                       706        1789.     817.     419.   458.
## 2 Electronic Arts               1356        1117.     600.     374.    14.4
## 3 Activision                     985         731.     433.     216.     6.71
## 4 Sony Computer Entertainment    687         606.     266.     187.    74.2
## 5 Ubisoft                        933         472.     253.     162.     7.52
## 6 Take-Two Interactive           422         404.     223.     119.     5.93
```

```
publisher__  %>% ggplot(aes(x=as.character(Publisher), y=Global_sales)) + geom_bar(stat = "identity") +
  xlab("Publisher)") + ylab("Global SAle") + ggtitle('Top 10 publisher global sales') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

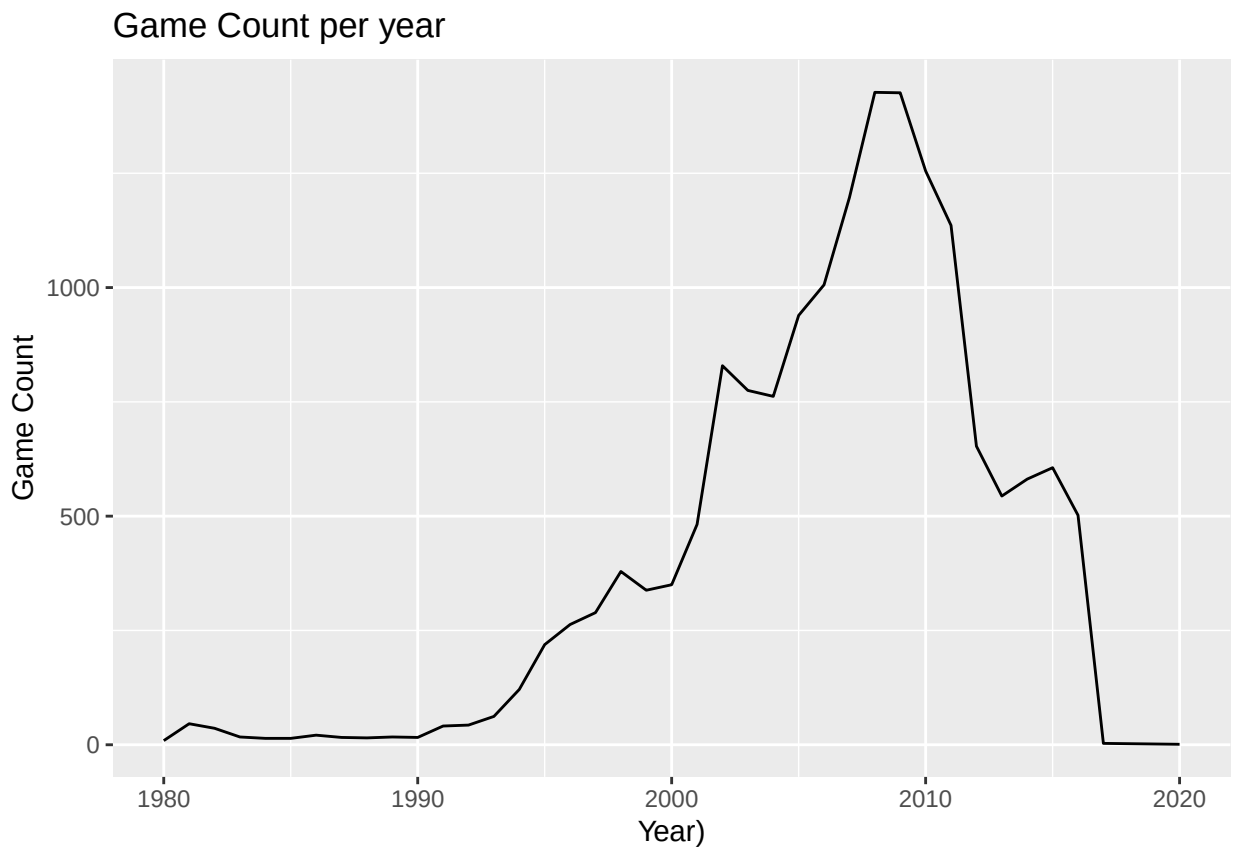


Release year

let'look at the change game industry over year

game sale over year

```
year_ <- data %>% group_by(Year_of_Release) %>%
  summarize( count       = n(),
             Global_sales = sum(Global_Sales),
             NA_Sales     = sum(NA_Sales),
             EU_Sales     = sum(EU_Sales),
             JP_Sales     = sum(JP_Sales),
  )

year_  %>% ggplot(aes(x=as.numeric(as.character(Year_of_Release)), y=count)) + geom_path() +
  xlab("Year)") + ylab("Game Count") + ggtitle('Game Count per year')
```

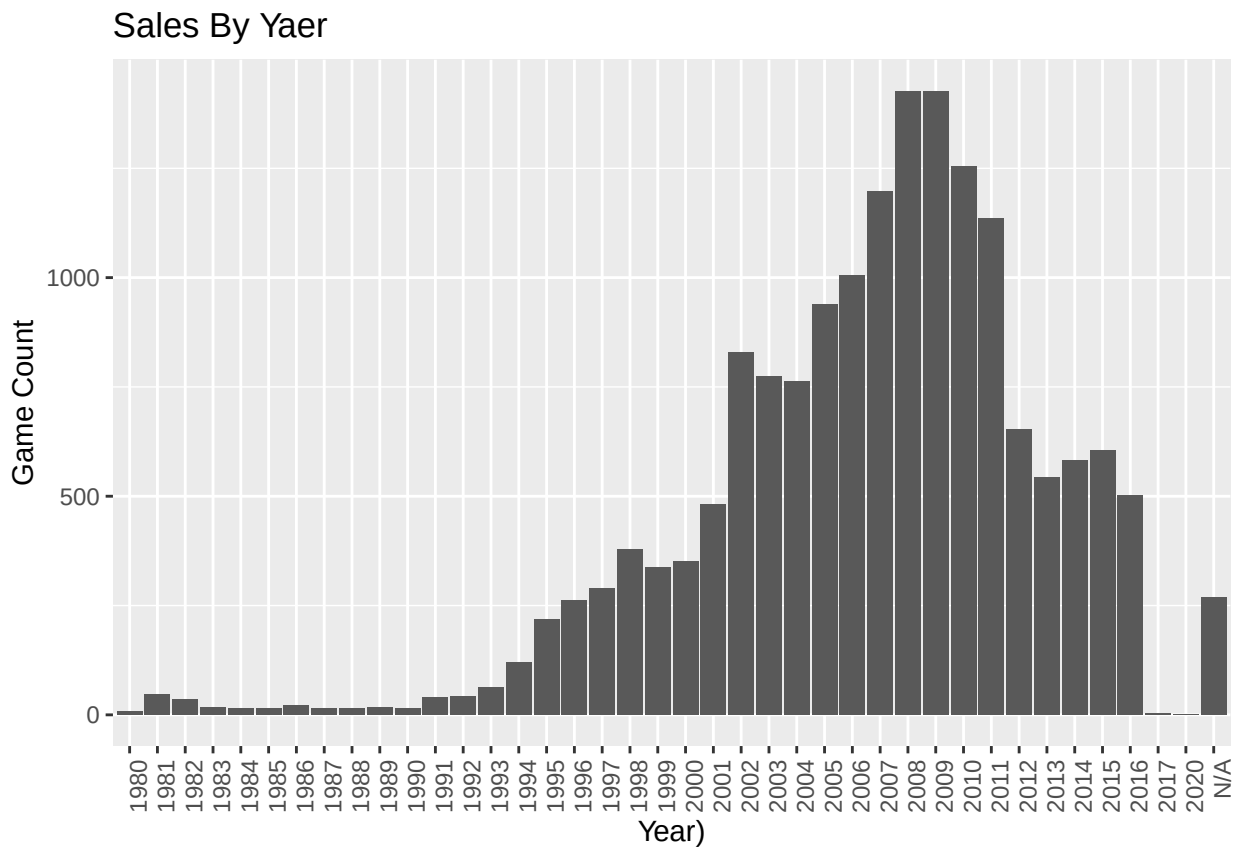


publisher change overyear

```
year_publisher_ <- data %>% group_by(Year_of_Release, Publisher) %>% group_by(Year_of_Release) %>%
  summarize( count       = n() )

year_publisher_  %>% head()
```

```
## # A tibble: 6 x 2
##   Year_of_Release count
##   <fct>           <int>
## 1 1980                9
## 2 1981               46
## 3 1982               36
## 4 1983               17
## 5 1984               14
## 6 1985               14
```

```
year_publisher_   %>% ggplot(aes(x=Year_of_Release, y=count)) + geom_bar(stat = "identity") +
  xlab("Year)") + ylab("Game Count") + ggtitle('Sales By Yaer') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

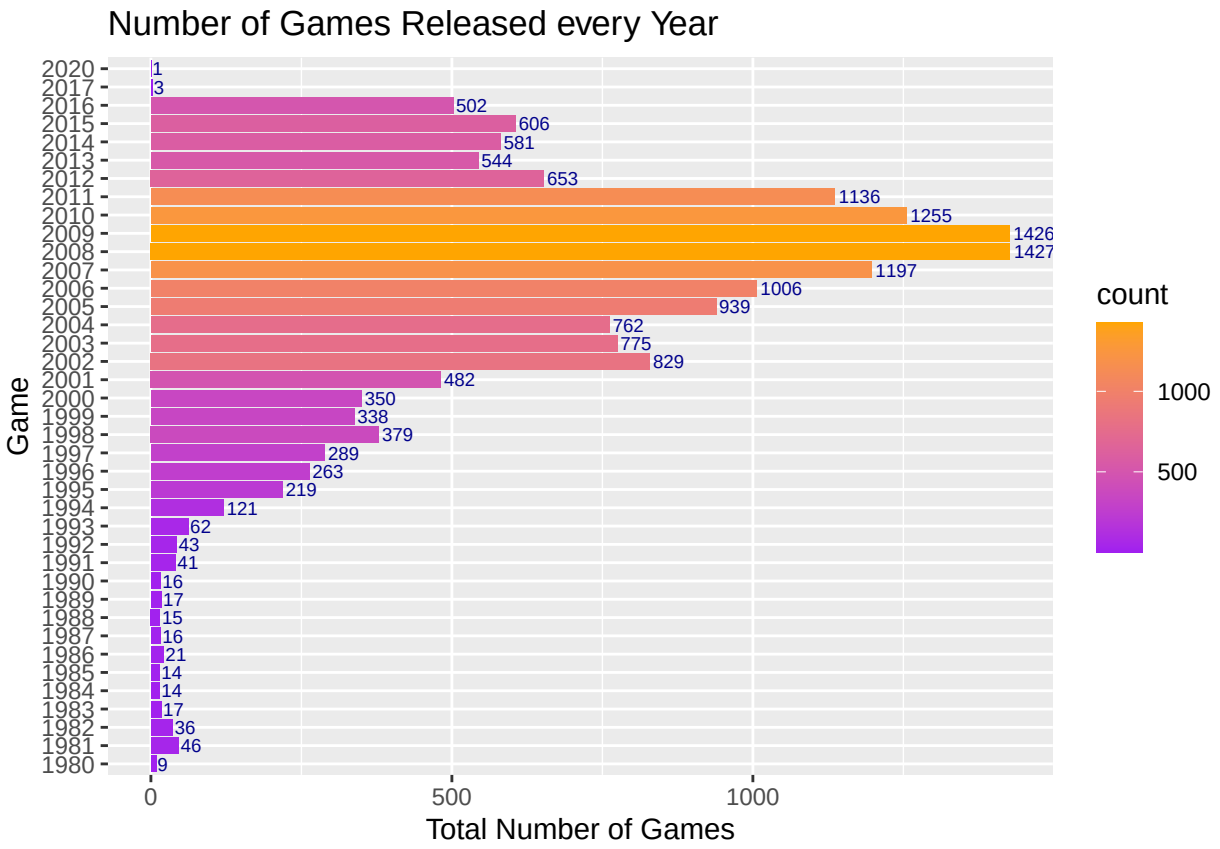


sales over year

```
year_sale_ <- data %>% group_by(Year_of_Release) %>%

  summarize( count       = n(),
             Global_sales = sum(Global_Sales),
             NA_Sales     = sum(NA_Sales),
             EU_Sales     = sum(EU_Sales),
             JP_Sales     = sum(JP_Sales),
  )
```

what is the total number of games released every year?

```
ggplot(data[data$Year_of_Release!="N/A",], aes(x=Year_of_Release, fill=..count..)) +
  geom_bar()+
  scale_color_gradient(low="purple", high="orange")+
  scale_fill_gradient(low="purple", high="orange")+
  labs(title="Number of Games Released every Year", x= "Game",
       y= "Total Number of Games")+
  geom_text(stat='count',aes(label=..count..), hjust=-0.1,color="darkblue", size=2.5)+
  coord_flip()
```



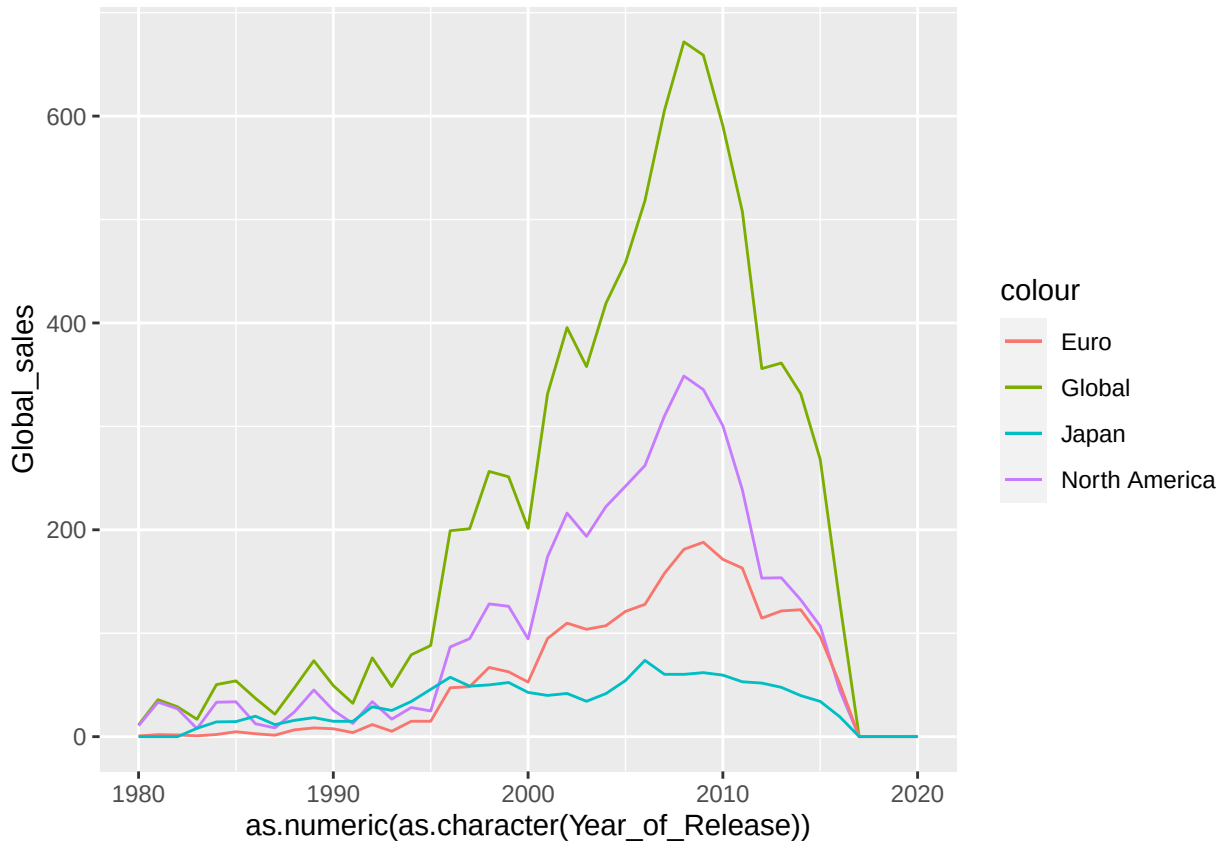
Sales over year by By Region

```
year_sale_ <- data  %>% group_by(Year_of_Release) %>%

  summarize( count       = n(),
             Global_sales = sum(Global_Sales),
             NA_Sales     = sum(NA_Sales),
             EU_Sales     = sum(EU_Sales),
             JP_Sales     = sum(JP_Sales),
  )

year_sale_
```

```
## # A tibble: 40 x 6
##    Year_of_Release count Global_sales NA_Sales EU_Sales JP_Sales
##    <fct>           <int>        <dbl>    <dbl>    <dbl>    <dbl>
##  1 1980                9         11.4    10.6      0.67      0
##  2 1981               46         35.8    33.4      1.96      0
##  3 1982               36         28.9    26.9      1.65      0
##  4 1983               17         16.8     7.76     0.8       8.1
##  5 1984               14         50.4    33.3      2.1      14.3
##  6 1985               14         53.9    33.7      4.74     14.6
##  7 1986               21         37.1    12.5      2.84     19.8
##  8 1987               16         21.7     8.46     1.41     11.6
```

```
##  9 1988               15         47.2    23.9      6.59     15.8
## 10 1989               17         73.4    45.2      8.44     18.4
## # ... with 30 more rows
```

```
ggplot(year_sale_, aes(as.numeric(as.character(Year_of_Release)))) +
  geom_line(aes(y = Global_sales, colour = "Global")) +
  geom_line(aes(y = NA_Sales, colour = "North America")) +
  geom_line(aes(y = EU_Sales, colour = "Euro")) +
  geom_line(aes(y = JP_Sales, colour = "Japan"))
```



2- b Modelling and prediction

wee will try predict the sales of future games based on the pattern that can be learned from this data set. At the moment, the analysis focuses only on the sales in EURO Sales.

prepare data afor modellig

remove missing val in Genre & Name

```
data <- data %>% filter(!is.na(Genre)&!is.na(Name))
```

Train-test spilit

```
indexes<-createDataPartition(y=data$Global_Sales,p=0.7,list=FALSE)
train_set<-data[indexes,]
test_set<-data[-indexes,]

nrow(train_set)
```

```
## [1] 11705
```

```
nrow(test_set)
```

```
## [1] 5014
```

lineer regression

```
ln_fit <- train(EU_Sales~Year_of_Release+Genre+Critic_Score+
                  Critic_Count+User_Score+User_Count+Platform,
                data=train_set,
                 na.action = na.omit,
                method="lm")

test_p_b <- predict(ln_fit,test_set)
ln_rmse<- RMSE(test_p_b, test_set$EU_Sales) #

ln_rmse
```

```
## [1] 0.6090377
```

```
rmse_results <- data_frame(method = "Lineer regression", RMSE =  ln_rmse)
```

Sport vector machine

```
control <- trainControl(method='none', )

set.seed(666)
sv_fit <- train(EU_Sales~Year_of_Release+Genre+Critic_Score+
                  Critic_Count+User_Score+User_Count+Platform,
                data=train_set,
                na.action = na.omit,
                trControl=control,
                method="svmLinear" )


test_p_svm<- predict(sv_fit,test_set)

sv_rmse<- RMSE(test_p_svm, test_set$EU_Sales) #

sv_rmse
```

```
## [1] 0.5697345
```

```
rmse_results <- bind_rows(rmse_results, data_frame(method = "Support vectore meachine", RMSE =  sv_rmse))
```

random forest

```
#10 folds repeat 3 times
control <- trainControl(method='none')


set.seed(666)
#Number randomely variable selected is mtry
mtry <- sqrt(ncol(train_set))
tunegrid <- expand.grid(.mtry=mtry)
rf_fit <- train(EU_Sales~Year_of_Release+Genre+Critic_Score+
                      Critic_Count+User_Score+User_Count+Platform,
                    data=train_set,
                    method='rf',
                    na.action = na.omit,

                    tuneGrid=tunegrid,
                    trControl=control)

rf_prd<- predict(rf_fit,test_set)

rf_rmse<- RMSE(rf_prd, test_set$EU_Sales)


rf_rmse
```

```
## [1] 0.5633096
```

```
rmse_results <- bind_rows(rmse_results, data_frame(method = "Random  forest", RMSE =  rf_rmse))
```

check results

```
rmse_results %>% knitr::kable()
```

method	RMSE
Lineer regression	0.6090377
Support vectore meachine	0.5697345
Random forest	0.5633096

3 Results

```
rmse_results %>% knitr::kable()
```

method	RMSE
Lineer regression	0.6090377
Support vectore meachine	0.5697345
Random forest	0.5633096

Random forest models with the best performance. performance are actually very close to each other. lineer regression has a slightly worse performance compared to the two models. support vector machne (linear) has the worst performance.

4 Conclusion

Data Science and machine learning methods can help to industy that improving fast with many publisher even though most people spend their most time on mobile phone, social meida and other modern thinks. We can see how improving over years in this simple work. in this work we tried analysis and visualization this sample data and create machine learning models that predict game sales in Euro we can see random forest model work fine in this data set, but we can get different result if we work larger dataset that has less missing data

My project Github repository is **in this link**