Introduction

Introduction

Context

The data in this study reports video games sales in EU, NA, Japan, and other places. Inlcuded also are the names of the games, the platform, publisher, year, and the rank in sales.

Other analyses completed

Gap of this analysis

The Analysis Question

The question/objectives in this analysis is

To explore the data and find various patterns and trends

Determine which Platforms and Genres are successful in Japan

Predict Japanese Sales

The priamry benefit of predicting Japanese sales is a company can determine what genres and platforms to offer in Japan.

KPIs

The prediction of Japanese sales is the primary KPI. This metric will be assessed using the following tools.

Summary statistics

Mean absolute error

Correlation

Solution Overview

Gradient boosting was employed to predict Japanese video game sales

The OLS regression model indicated which genre and platforms were significant

The boosted regression model indicated which variables where of importance

Importing Packages and Data

Packages

Data/ Scrubbing/exploring

```
library(readr)
vgsales <- 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.
```

```
summary(vgsales)
```

```
##       Rank           Name              Platform             Year
##  Min.   :    1   Length:16598       Length:16598       Length:16598
##  1st Qu.: 4151   Class :character   Class :character   Class :character
```

```
##  Median : 8300   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 8301
##  3rd Qu.:12450
##  Max.   :16600
##     Genre             Publisher           NA_Sales          EU_Sales
##  Length:16598       Length:16598       Min.   : 0.0000   Min.   : 0.0000
##  Class :character   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median : 0.0800   Median : 0.0200
##                                        Mean   : 0.2647   Mean   : 0.1467
##                                        3rd Qu.: 0.2400   3rd Qu.: 0.1100
##                                        Max.   :41.4900   Max.   :29.0200
##     JP_Sales          Other_Sales        Global_Sales
##  Min.   : 0.00000   Min.   : 0.00000   Min.   : 0.0100
##  1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.0600
##  Median : 0.00000   Median : 0.01000   Median : 0.1700
##  Mean   : 0.07778   Mean   : 0.04806   Mean   : 0.5374
##  3rd Qu.: 0.04000   3rd Qu.: 0.04000   3rd Qu.: 0.4700
##  Max.   :10.22000   Max.   :10.57000   Max.   :82.7400
```

Continuous Variables

Below is a correlation matrix

```
cor(vgsales[,7:11]) %>%corrplot(method = 'number')
```

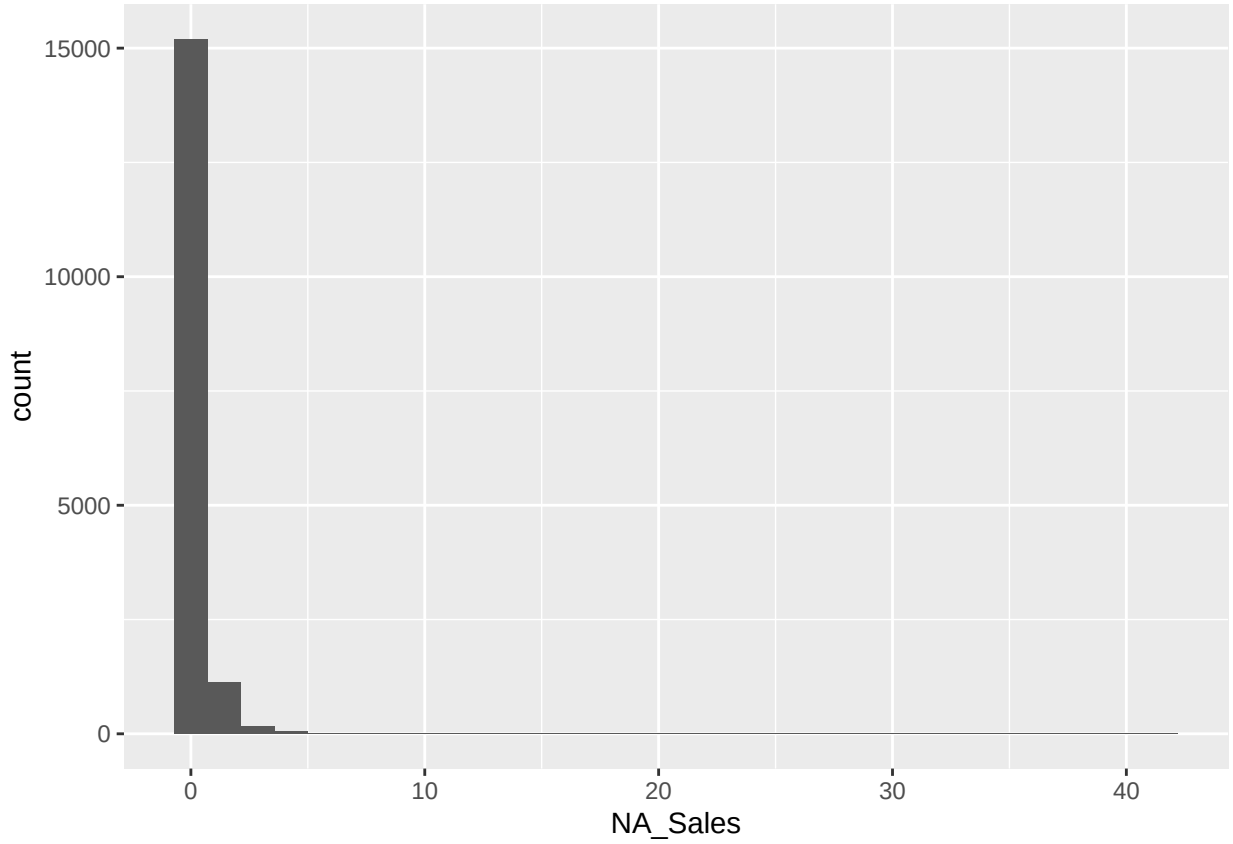
It appears that there is a strong correlation between NA and EU sales and between EU and other sales. Global sales is a sum so it makes sense that it's correlation is so high. However, Japanese sales are different from the other sales as its correlation with global sales is low compared to the other variables.

Purpose

We already know what the descriptive states are but now lets look at some histograms

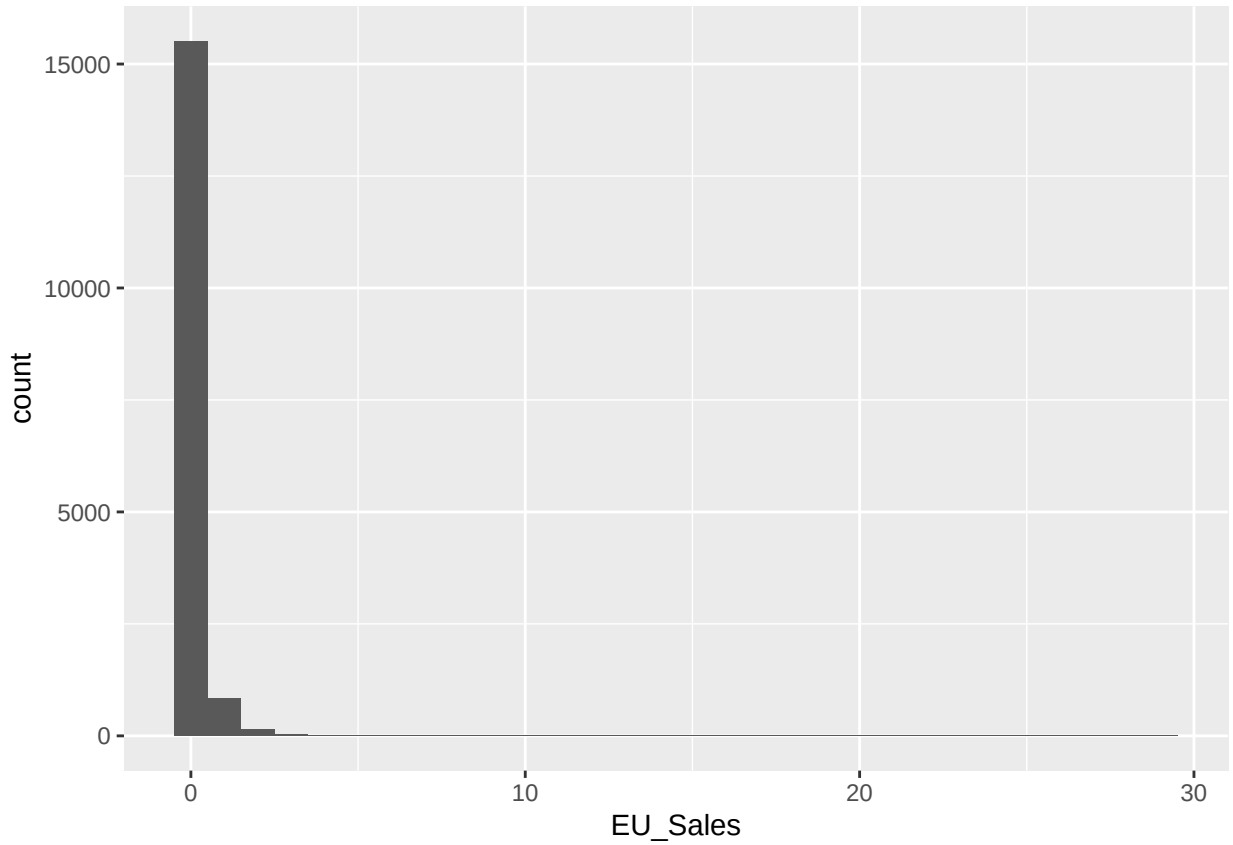
```
ggplot(vgsales,aes(NA_Sales))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



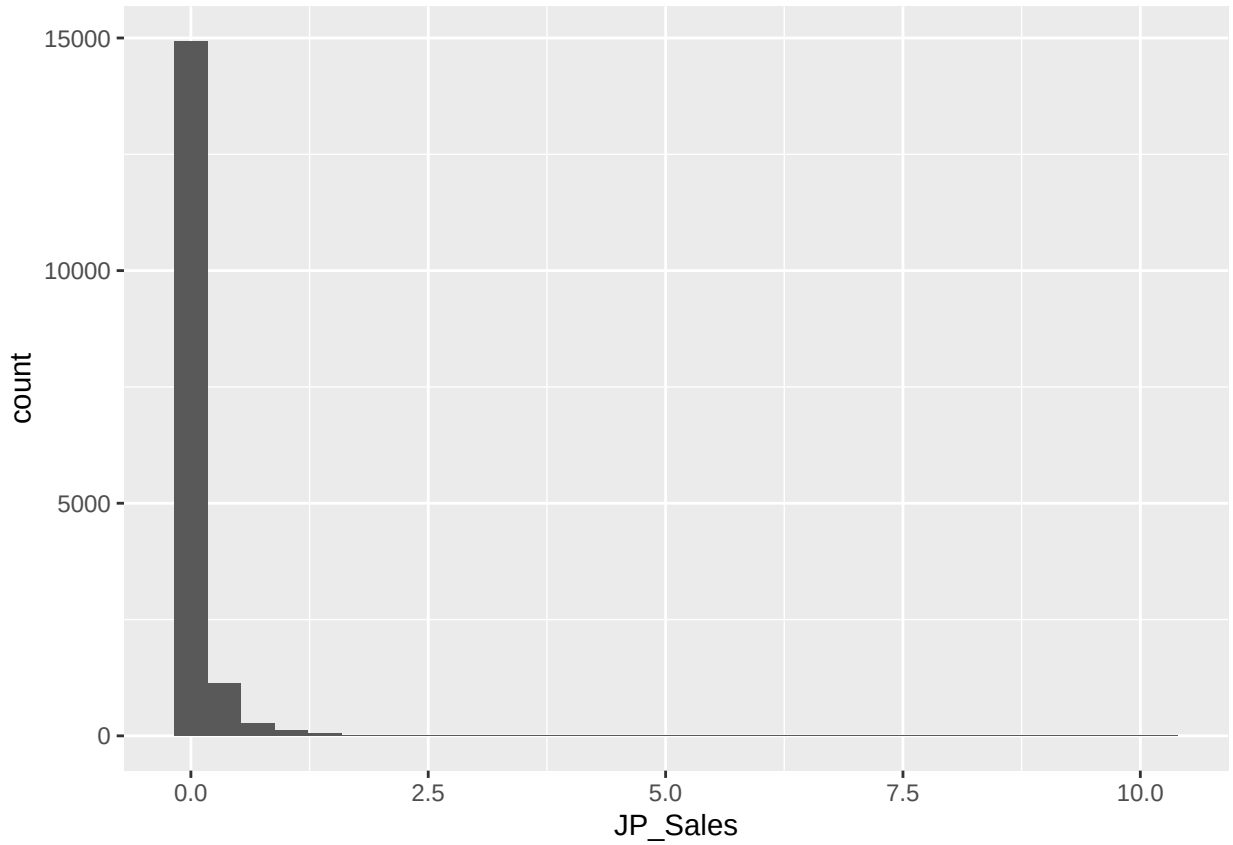

```
ggplot(vgsales,aes(EU_Sales))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



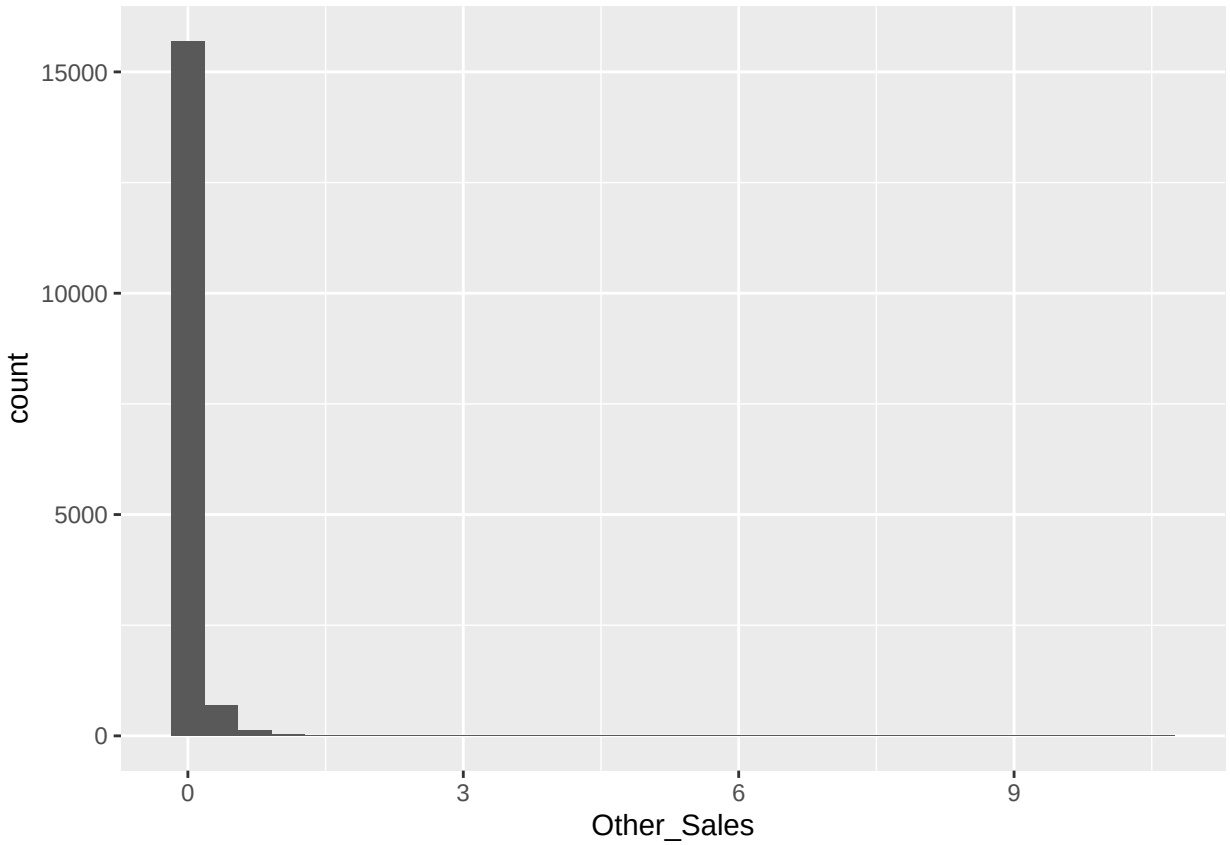

```
ggplot(vgsales,aes(JP_Sales))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(vgsales,aes(Other_Sales))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Observation

none of the continuous variables follow a normal distribution. This means we will need to do a log transformation when developing the model

Categorical Variables

A look at the summary doesn't indicate any major problems with the data. We will double check for duplicates and missing values below.

```
print(table(duplicated(vgsales)))
```

```
##
## FALSE
## 16598
```

```
print(table(is.na(vgsales)))
```

```
##
##  FALSE
## 182578
```

Both outputs indicate there are no duplicates or missing values. Well now look at the categorical variables a little more closely.

Purpose

Check categorical variables

```
length(unique(vgsales$Name))
```

```
## [1] 11493
```

```
table(vgsales$Platform)
```

```
##
## 2600  3DO  3DS   DC   DS   GB  GBA   GC  GEN   GG  N64  NES   NG   PC PCFX   PS
##  133    3  509   52 2163   98  822  556   27    1  319   98   12  960    1 1196
##  PS2  PS3  PS4  PSP  PSV  SAT  SCD SNES TG16  Wii WiiU   WS X360   XB XOne
## 2161 1329  336 1213  413  173    6  239    2 1325  143    6 1265  824  213
```

```
table(vgsales$Year)
```

```
##
## 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995
##    9   46   36   17   14   14   21   16   15   17   16   41   43   60  121  219
## 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
##  263  289  379  338  349  482  829  775  763  941 1008 1202 1428 1431 1259 1139
## 2012 2013 2014 2015 2016 2017 2020  N/A
##  657  546  582  614  344    3    1  271
```

There are a lot of names in the name category and none of them are that common.

In addition, there are a lot of different platforms.

Looking by year we can see that there are different amounts of data for each year. What this means is that data from the 1990's and 2000's is going to heavily influence the model because more data comes from that time. This will make it hard to determine if different years were more profitable than others.

Lastly, the year column does have missing values but they were coded in a way that the function I used could detect them. Since the value is small I can probably just leave these values out when year is a criteria.

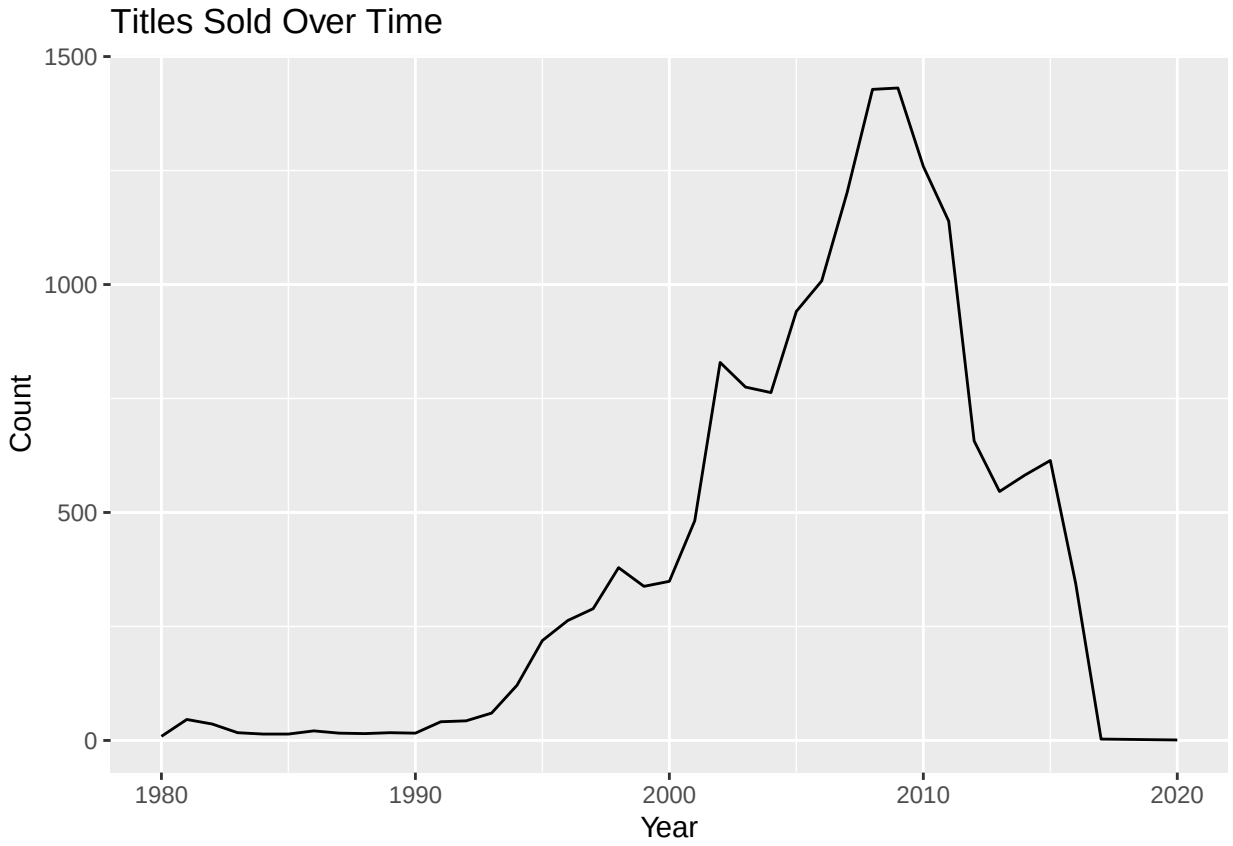
Categorical Visualizations

Purpose

Our first visualization will look at the count of sales over time.

```
vgsales %>% dplyr::transmute(Year=as.numeric(Year)) %>%filter(Year != 'N/A') %>%
        group_by(Year) %>%dplyr::count(Year) %>%
        ggplot(aes(Year,n,group=1))+geom_line()+ggtitle('Titles Sold Over Time')+labs(y="Count")
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```



Observation

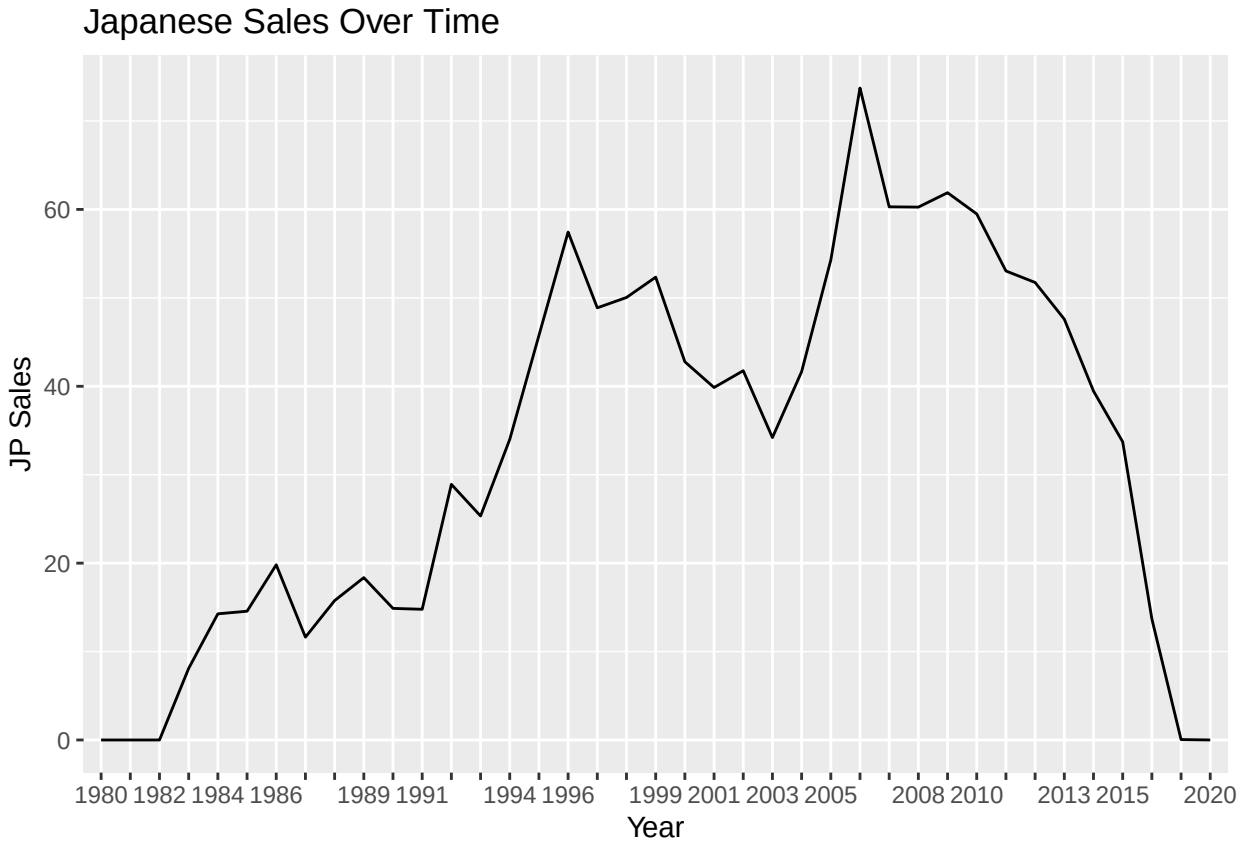
From this first chart we can see the units sold peak in the 2008 and 2009 and began to decline steep after 2011. This makes sense because most of the data comes from the 1990's and 2000's

Visual 2

Purpose

This second chart shows Japanese sales over time and rough follows the pattern of the previous plot.

```
vgsales %>% dplyr::transmute(Year,JP_Sales) %>% dplyr::filter(Year != 'N/A') %>%
        dplyr::group_by(Year) %>% dplyr::summarize(all_sales=sum(JP_Sales)) %>%
       ggplot(aes(Year,all_sales,group=1))+geom_line()+ggtitle('Japanese Sales Over Time') +
        scale_x_discrete(guide = guide_axis(check.overlap = TRUE))+labs(y='JP Sales')
```



Observation

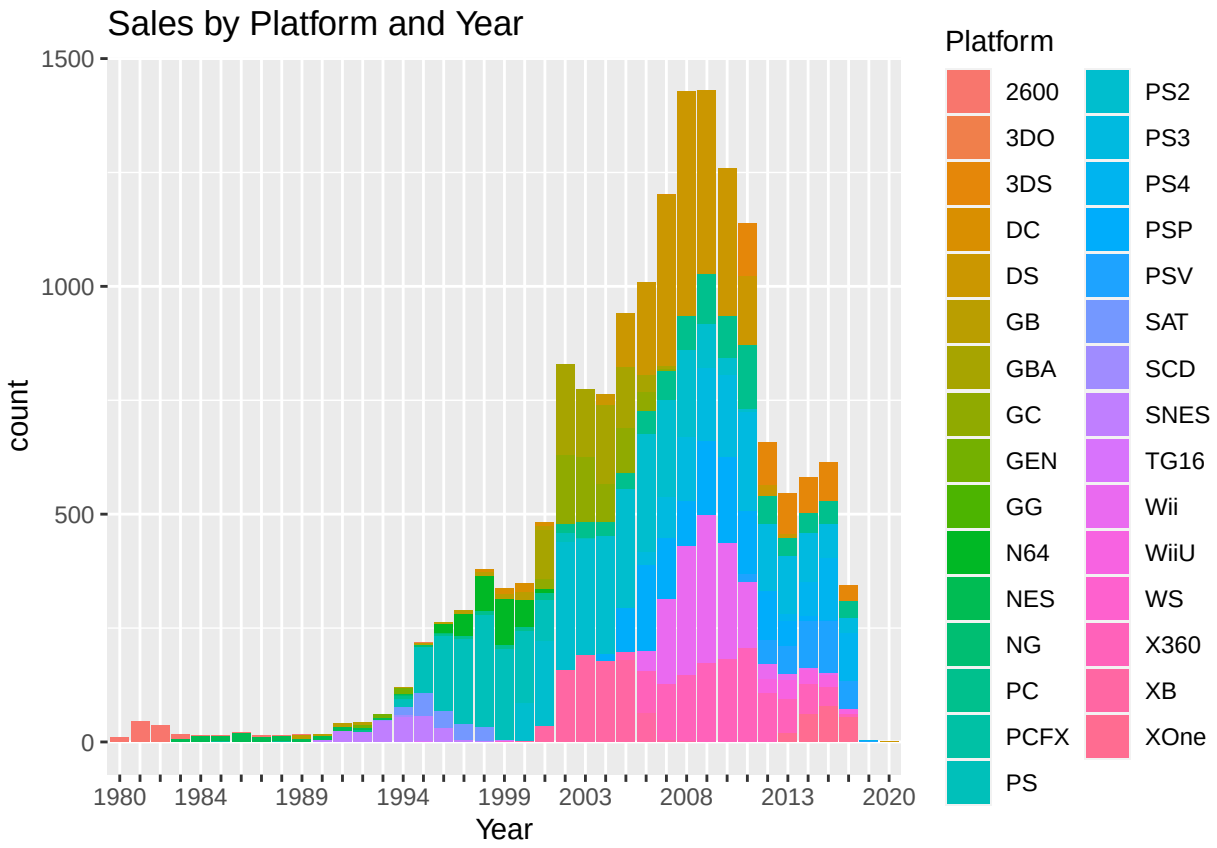
From this first chart we can see the sales peak in the 2008 and 2009 and began to decline steep after 2011. Again this is because of when most of our data came from

Visual 3 & 4

Purpose

This chart shows Units sold over time while considering the various platforms that were available while plot 4 shows units sold by platform.

```
vgsales %>% filter(Year !='N/A') %>%
        ggplot(aes(x = factor(Year), fill = factor(Platform))) +
        geom_bar() +
        scale_x_discrete(guide = guide_axis(check.overlap = TRUE))+ggtitle('Sales by Platform and Year'
```



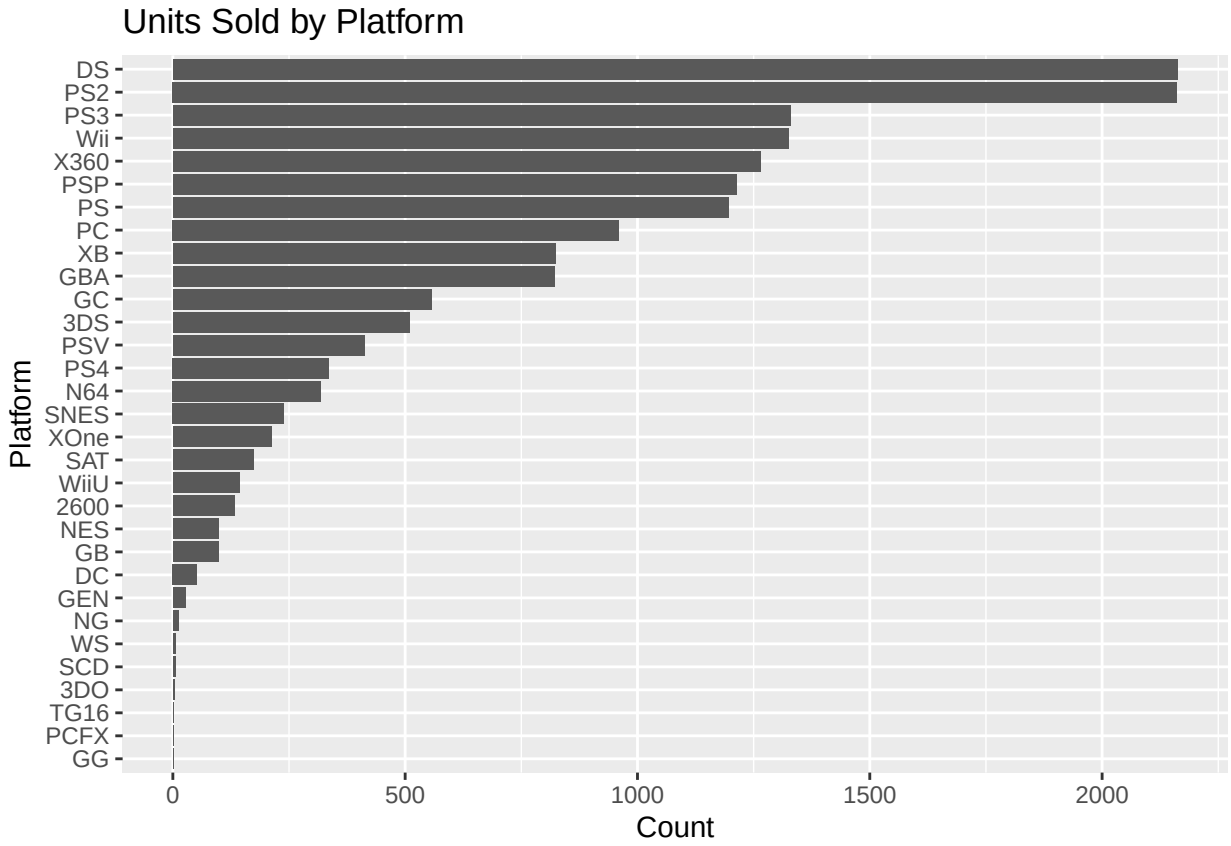
Observation>

Nintendo generally dominants the platform business followed by PlayStation

Purpose

Determine how many units were sold by platform

```
vgsales %>% dplyr::select(Platform) %>% dplyr::group_by(Platform)%>% dplyr::count(Platform) %>%
ggplot(aes( x= reorder(Platform, +n), y = n))+geom_bar(stat='identity')+coord_flip()+ggtitle('Units Sol
```



Observation

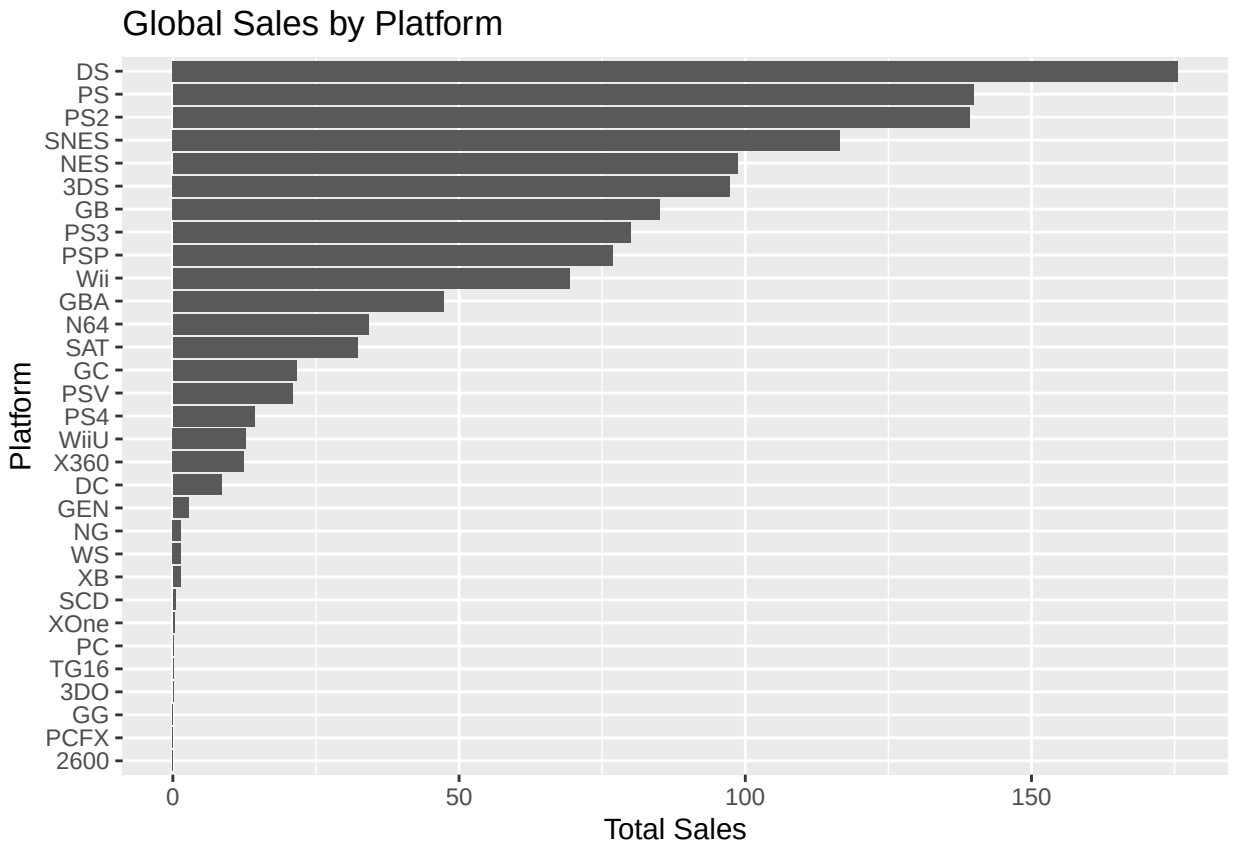
The Nintendo DS and PS2 are the top platforms sold in terms of units all-time in this dataset. In addition, Nintnendoo and Sony hold the top four spots. Sony controls 4 of the top 6 spots indicating that Sony is the top Platform maker.

Visual 5

Purpose

This chart shows Japanese sales by Platform

```
vgsales %>% dplyr::select(Platform, JP_Sales) %>%dplyr::group_by(Platform) %>%
        dplyr::summarize(total_sales_platform=sum(JP_Sales)) %>%
        ggplot(aes(reorder(Platform,+total_sales_platform),total_sales_platform)) + geom_bar(stat='iden
```



Observation

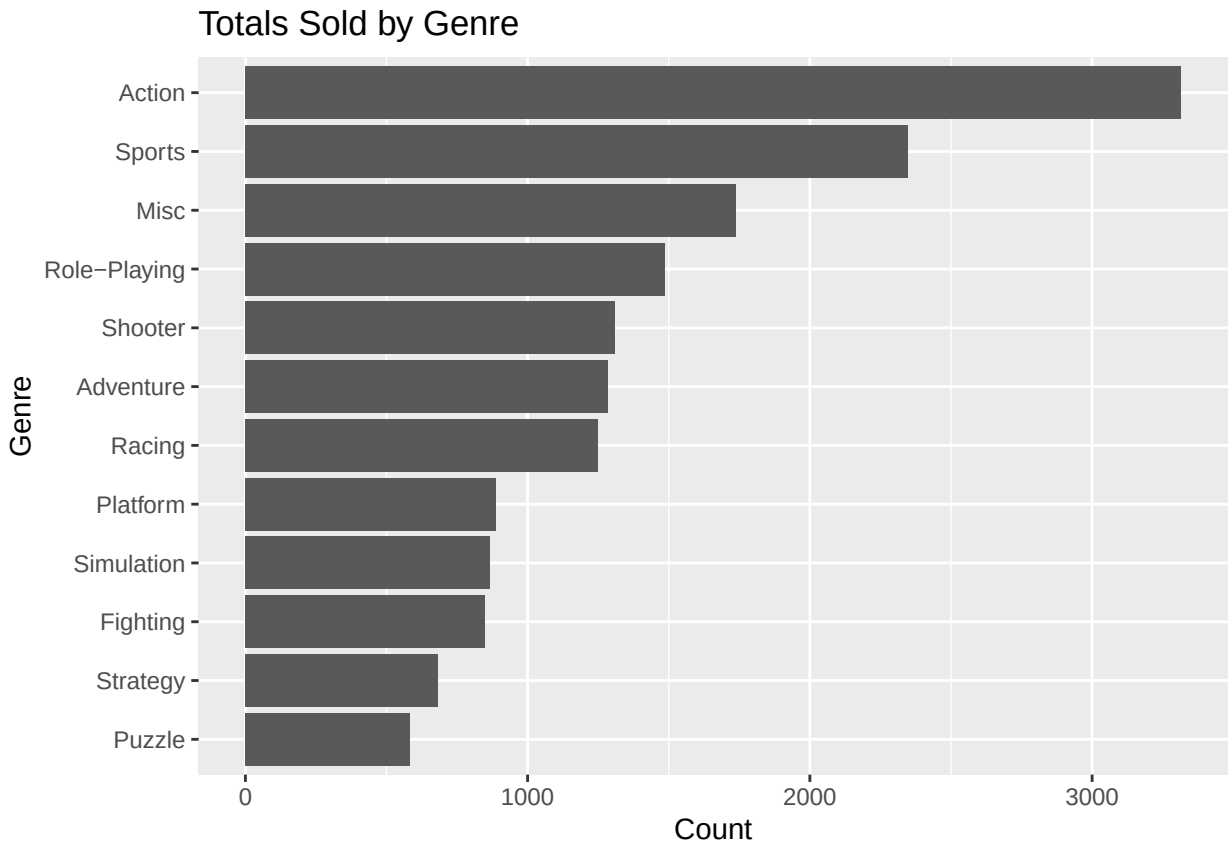
The Nintendo DS has sold the most units but the PS2 has generate the most money followed by the x360. The falloff in sales for the older platforms is probably a combination of the older units nomrally have a lower price compared to the price of platforms today. In addition, the gaming indsutry has grown tremendously over the years.

Visual 5

Purpose

This chart shows Units sold while considering the various genres.

```
vgsales %>% dplyr::select(Genre) %>% dplyr::group_by(Genre)%>% dplyr::count(Genre) %>%
ggplot(aes( x= reorder(Genre, +n), y = n))+geom_bar(stat='identity')+coord_flip()+ggtitle("Totals Sold
```



Observation

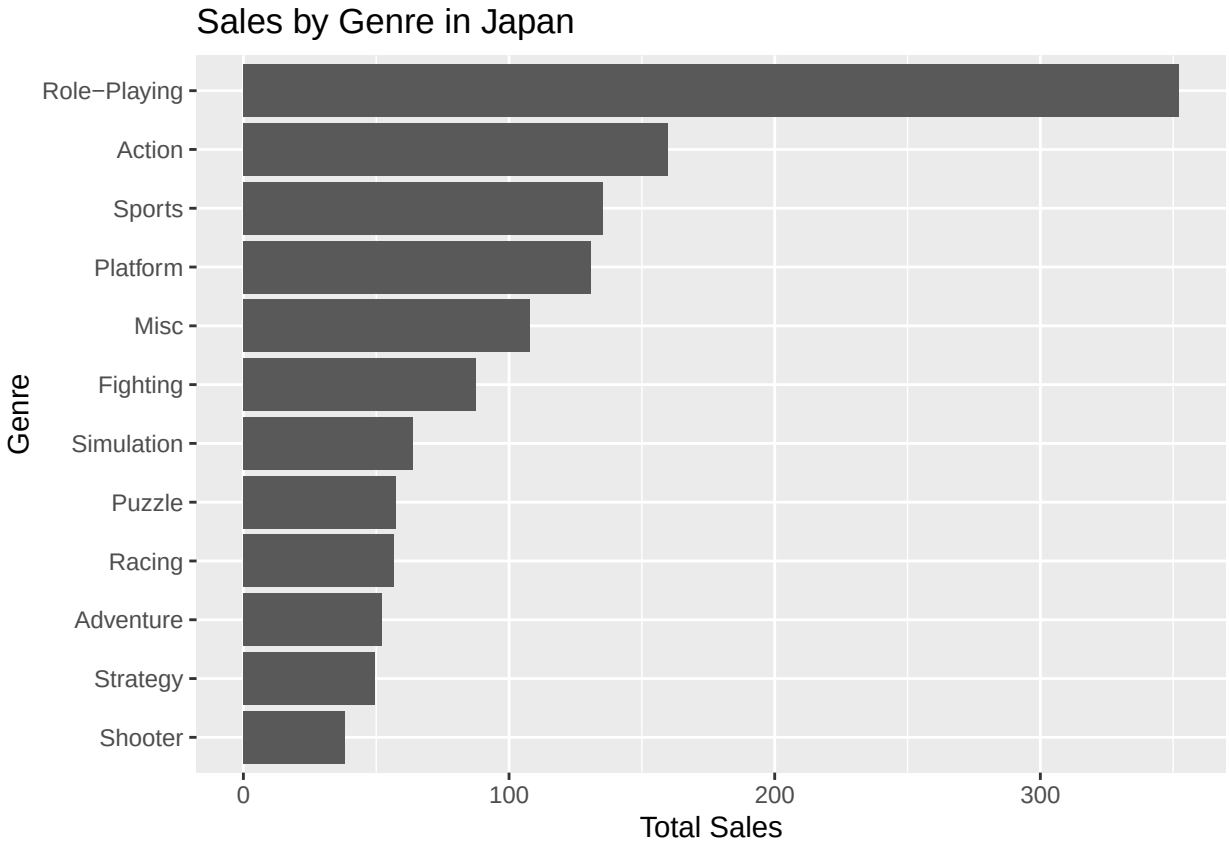
Action games are the most popular in terms of units sold. This might be signficant when predict sales for Japan. Puzzle games don't seem to sale well in comparision to other genres that are available.

Visual 6

Purpose

This chart shows Sales while considering the various genres.

```
vgsales %>% dplyr::select(Genre, JP_Sales) %>%dplyr::group_by(Genre) %>%
        dplyr::summarize(total_sales_genre=sum(JP_Sales)) %>%
        ggplot(aes(reorder(Genre,+total_sales_genre),total_sales_genre)) + geom_bar(stat='identity')+co
```



Observation

Japanese love role-playing games by far. From there the pattern is similiar to the previous plot

Visual 7

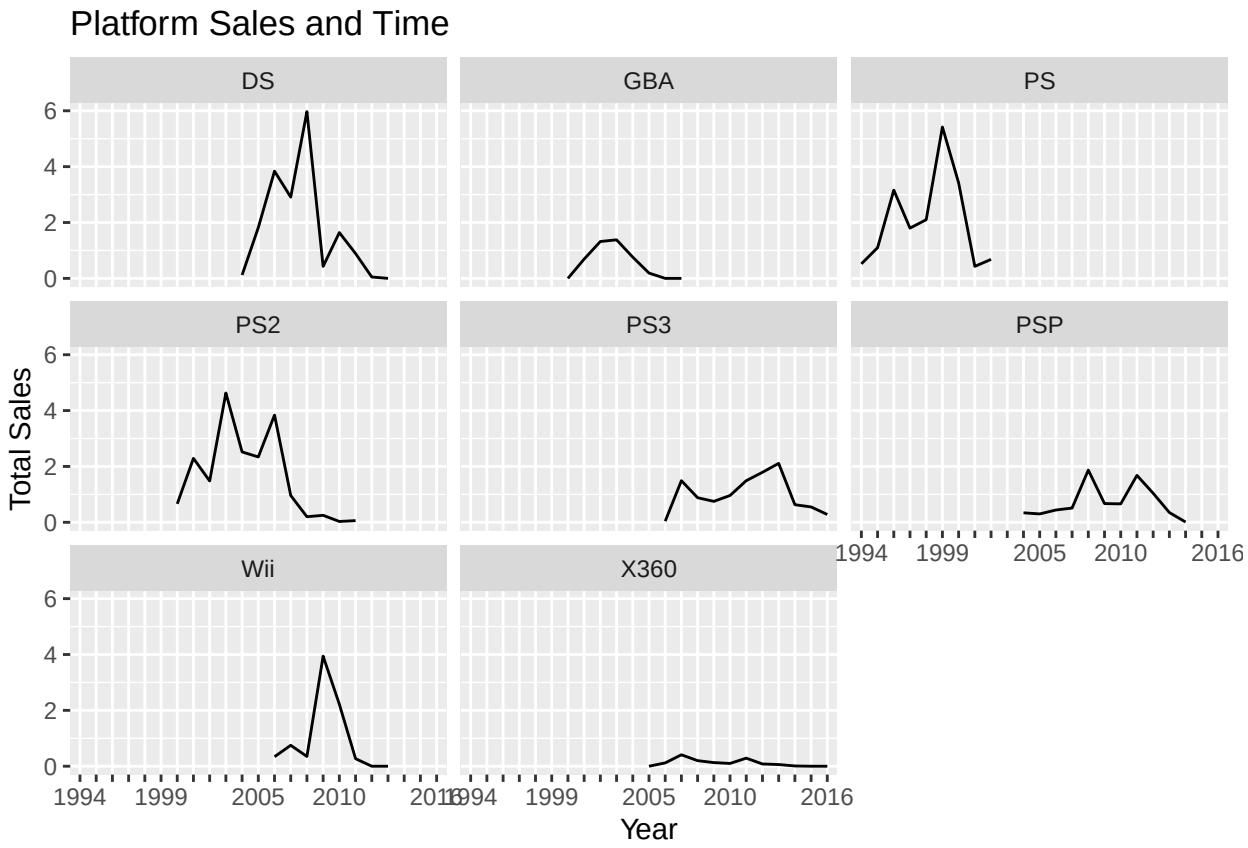
Purpose

This chart shows Sales while considering the various select Platforms over time

```
vgsales %>% dplyr::select(Year,Platform,JP_Sales)%>%
        dplyr::filter(Platform==c('DS','PS2','PS3','Wii','X360','PS','GBA','PSP') & Year!=('N/A'))%>%
        dplyr::group_by(Year,Platform) %>%   dplyr::summarize(total_sales=sum(JP_Sales)) %>%ggplot(aes(
        scale_x_discrete(guide = guide_axis(check.overlap = TRUE))+ggtitle("Platform Sales and Time")+la
```

```
## Warning in Platform == c("DS", "PS2", "PS3", "Wii", "X360", "PS", "GBA", :
## longer object length is not a multiple of shorter object length
```

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



Observation

The nintendo platforms seem to be wildly popular with a steep decline. Sony and Microsoft systems usually have a slightly more staying power. Again, all time data is inlfuence by the amount of data available per year which was scarce ac times

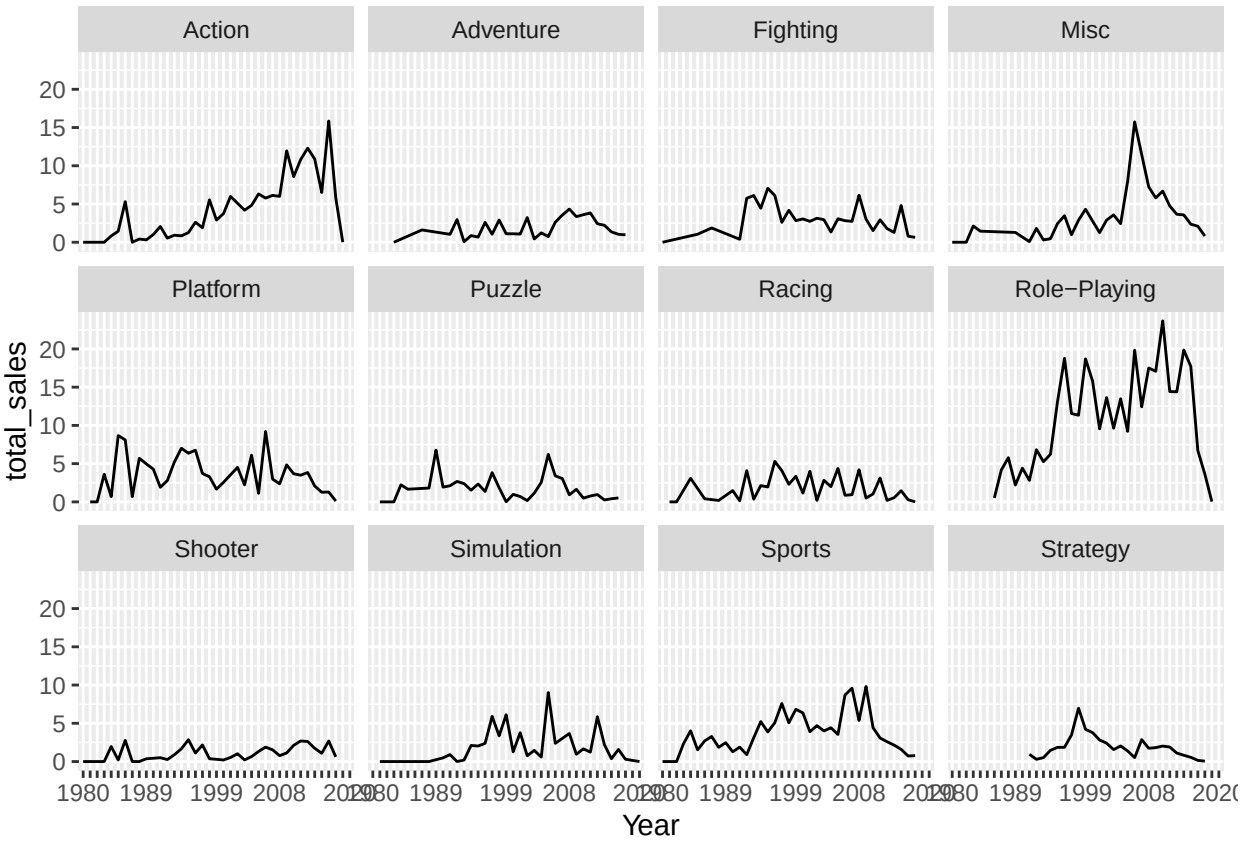
Visual 8

Purpose

This chart shows Sales while considering the various genres over time

```
vgsales %>% dplyr::select(Year, JP_Sales,Genre) %>% filter(Year!='N/A') %>%
        dplyr::group_by(Year,Genre) %>%
        dplyr::summarize(total_sales=sum(JP_Sales))%>%
        ggplot(aes(Year,total_sales,group=1))+geom_line()+facet_wrap(~Genre)+
        scale_x_discrete(guide = guide_axis(check.overlap = TRUE))
```

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



Observation

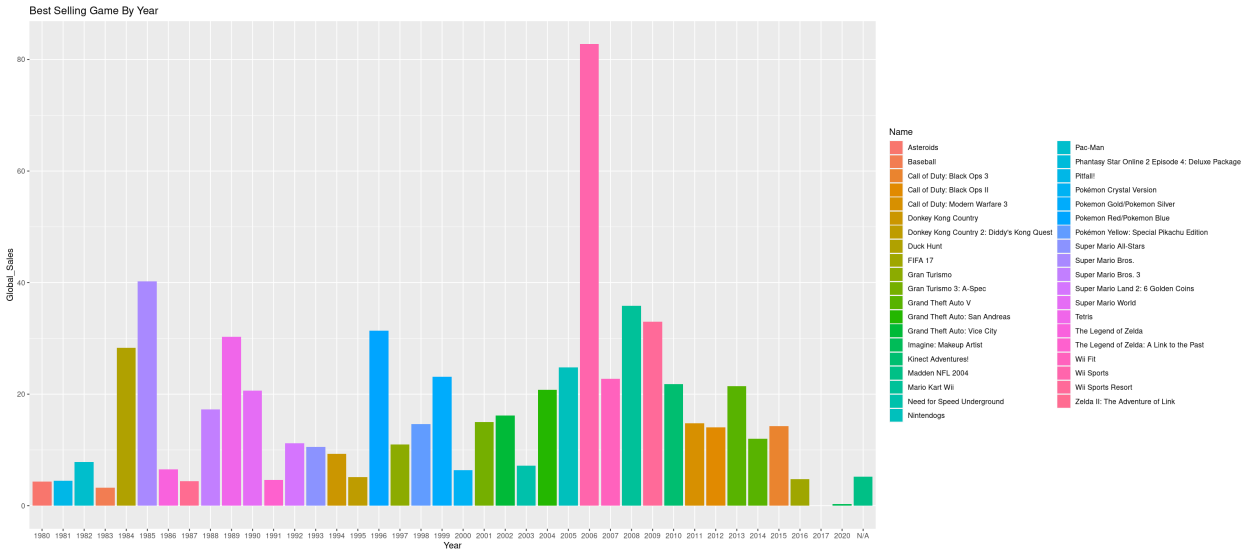
Role-Playing games were the best performers over the years. Platforms, strategy, adventures, and fighting have the most consistent performance.

Visual 9

Purpose

This chart shows the best selling game in terms of revenue by year.

```
knitr::include_graphics("good.png")
```



Observation

Wii sports was the best selling game in term of revenue by far

Visual 9

Purpose

This chart shows the average sales by genre.

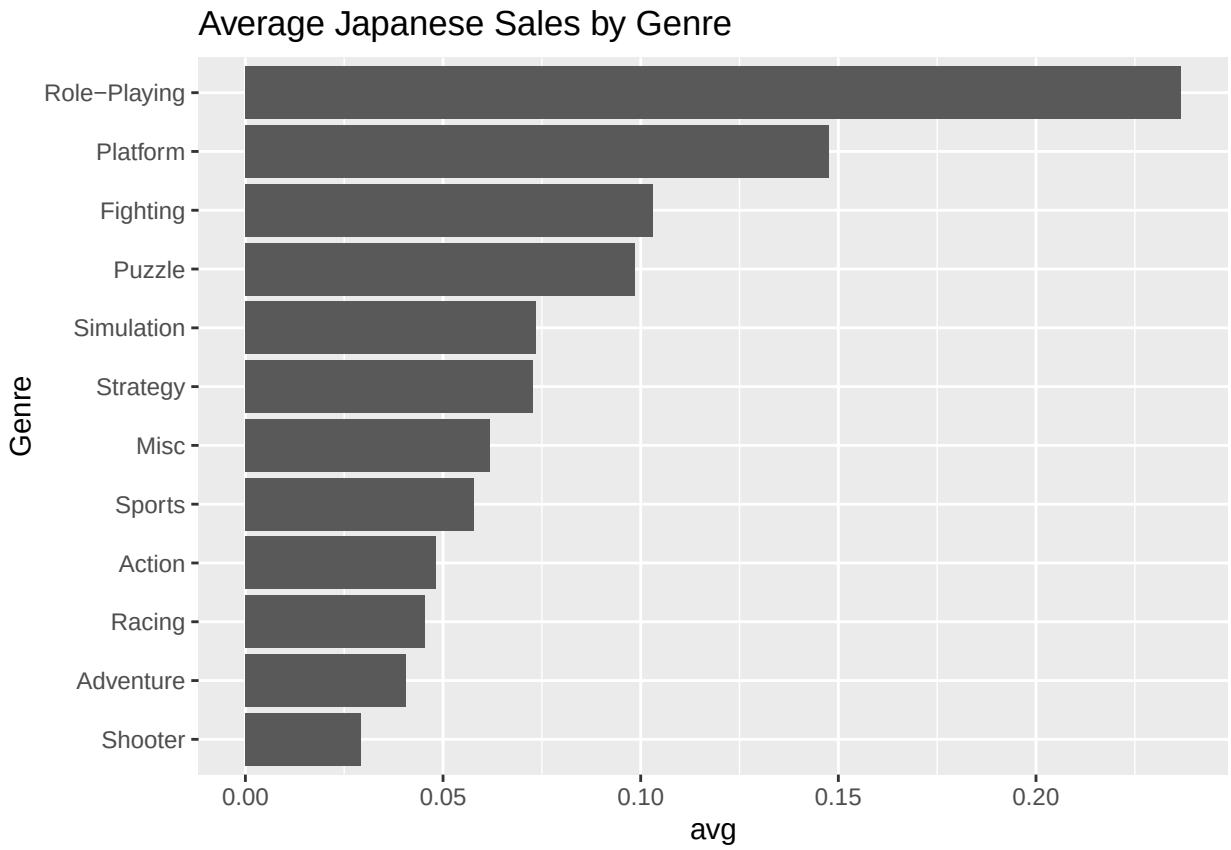
```
genre_totals<-vgsales%>%dplyr::select(Genre,JP_Sales) %>%dplyr::count(Genre)

sales<-vgsales%>%dplyr::select(Genre,JP_Sales) %>%dplyr::group_by(Genre)%>%
        dplyr::summarize(total_sales=sum(JP_Sales))
sales$total_sales/genre_totals$n
```

```
##  [1] 0.04823583 0.04048989 0.10300708 0.06196665 0.14759594 0.09847079
##  [7] 0.04538831 0.23676747 0.02922137 0.07347174 0.05770247 0.07262849
```

```
genre_totals$total_sales<-sales$total_sales
sum(genre_totals$total_sales)
```

```
## [1] 1291.02
```

```
genre_totals%>% dplyr::mutate(avg=(total_sales/n)) %>%
        ggplot(aes(reorder(Genre,+avg),avg))+geom_bar(stat='identity')+ggtitle('Average Japanese Sales
```



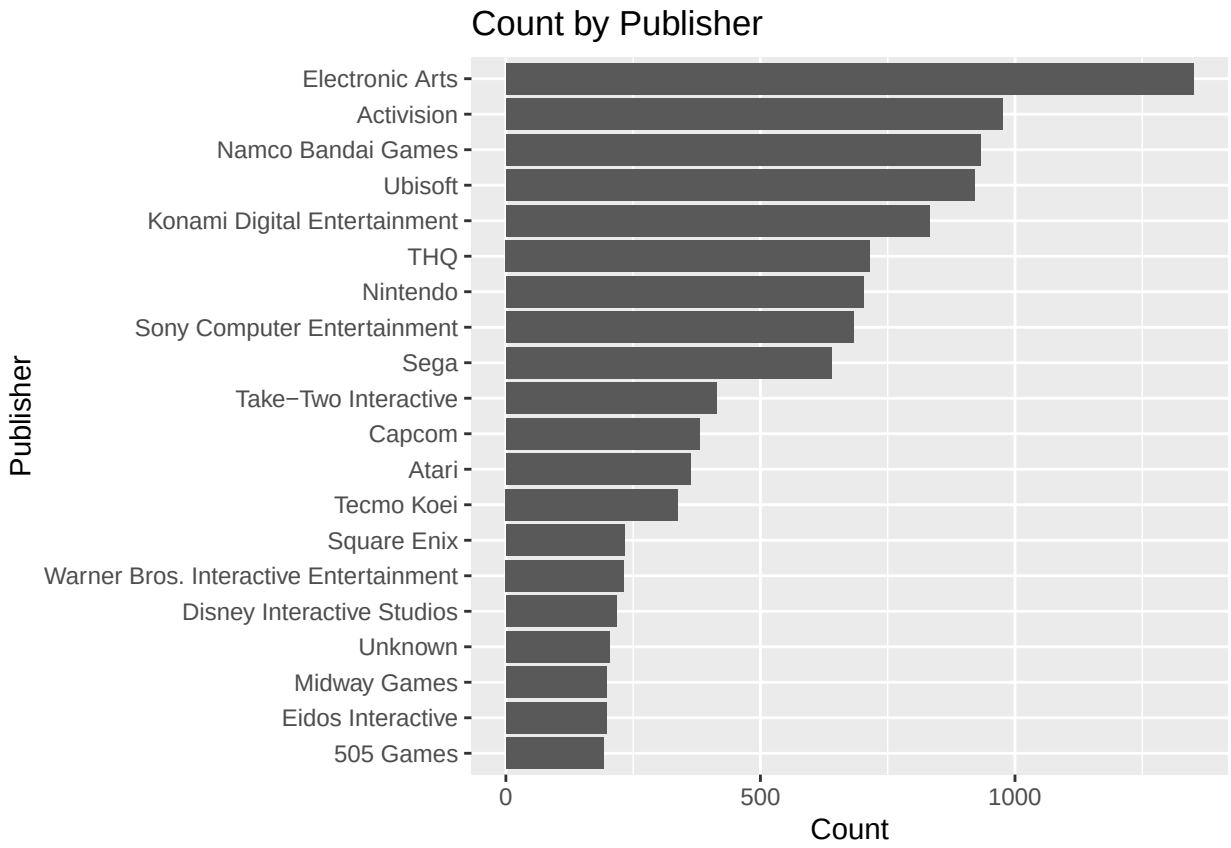
Observation

A role-playing game was on average generate much more revenue than other genres. Despite action and sports have a large number of units sold the actual average sales generate was low.

Data Prep

We now moving to model development. There are too many Publishers to make a understandable model. Therefore, we will take the top twenty publishers for are model. Below is the code and output for this experience

```
publisher_select<-vgsales %>%dplyr:: select(Publisher) %>%dplyr::group_by(Publisher)%>%dplyr::count(Publ
        dplyr::ungroup() %>% dplyr::transmute(Publisher,n,new=n/sum(n))
pub_names<-(publisher_select$Publisher[1:20])
pub_names_plot<-(publisher_select[1:20,])

ggplot(pub_names_plot,aes(reorder(Publisher,+n),n))+geom_bar(stat='identity')+coord_flip()+ggtitle('Cou
```



If we take the top 20 publishers we still have 65% of the data as shown below

```
sum(publisher_select$new[1:20])
```

```
## [1] 0.6458609
```

Dealing with the Platforms

The platforms also need to be simplified for interpretation. For example all the PS have been recoded as Sony and the same with the Nintendo platforms. When it was not clear what manufacturer a platform belong to it was recoded as other

```
vgsales_clean<-vgsales %>% mutate(simple_platform=recode(Platform, `Wii`="Nintendo",`NES`="Nintendo",`GB
`GBA`="Nintendo",`3DS`="Nintendo",`PS4`="Sony",`N64`="Nintendo",`PS`="Sony",`XB`="Microsoft",`2600`="Oth
`PSP`="Sony",`XOne`="Microsoft",`GC`="Nintendo",`WiiU`="Nintendo",`GEN`="Other",`DC`="Sega",`PSV`="Sony
`SAT`="Sega",`SCD`="Other",`WS`="Other",`NG`="Other",`TG16`="Other",`3DO`="Other",`GG`="Sega",`PCFX`="O
```

Log Transformation

We also need to do a log transformation of the continuous variables. In addition, because log cannot handle zeros we have to add a constant of 1 to each value so we do not get an error. We will reverse this when it is time for interpretation.

```
vgsales_clean<-vgsales_clean%>%filter(Publisher %in% pub_names)
vgsales_clean$log_JP_Sales<-log(vgsales_clean$JP_Sales+1)
vgsales_clean$log_NA_Sales<-log(vgsales_clean$NA_Sales+1)
vgsales_clean$log_EU_Sales<-log(vgsales_clean$EU_Sales+1)
vgsales_clean$log_Other_Sales<-log(vgsales_clean$Other_Sales+1)
```

Model Development

Below we divide are dataset into train and test sets

```
set.seed(123)
ind<-sample(2,nrow(vgsales_clean),replace=T,prob = c(0.7,0.3))
train<-vgsales_clean[ind==1,]
test<-vgsales_clean[ind==2,]
```

I like to always begin with a regression model just to get an insight into the data before completing a more complex analysis.

```
summary(lm(log_JP_Sales~log_NA_Sales+log_EU_Sales+log_Other_Sales+Genre+Publisher+simple_platform,train)
```

```
##
## Call:
## lm(formula = log_JP_Sales ~ log_NA_Sales + log_EU_Sales + log_Other_Sales +
##     Genre + Publisher + simple_platform, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44798 -0.05844 -0.00517  0.03241  1.60392
##
## Coefficients:
##                                                Estimate Std. Error t value
## (Intercept)                                   -0.050553   0.012915  -3.914
## log_NA_Sales                                   0.082773   0.008387   9.869
## log_EU_Sales                                   0.131540   0.011930  11.026
## log_Other_Sales                                0.155640   0.023119   6.732
## GenreAdventure                                -0.001277   0.007930  -0.161
## GenreFighting                                  0.020035   0.007674   2.611
## GenreMisc                                     -0.004544   0.005961  -0.762
## GenrePlatform                                  0.026064   0.007257   3.591
## GenrePuzzle                                    0.018440   0.010409   1.772
## GenreRacing                                    0.010091   0.007005   1.441
## GenreRole-Playing                              0.087637   0.006657  13.166
## GenreShooter                                  -0.013358   0.006479  -2.062
## GenreSimulation                                0.025464   0.007743   3.289
## GenreSports                                    0.018151   0.005325   3.409
## GenreStrategy                                  0.028312   0.008656   3.271
## PublisherActivision                           -0.027348   0.013019  -2.101
## PublisherAtari                                 0.008779   0.014710   0.597
## PublisherCapcom                                0.102759   0.014621   7.028
## PublisherDisney Interactive Studios           -0.026606   0.016039  -1.659
## PublisherEidos Interactive                     0.003197   0.016469   0.194
## PublisherElectronic Arts                      -0.036983   0.012818  -2.885
## PublisherKonami Digital Entertainment          0.079258   0.013130   6.037
## PublisherMidway Games                         -0.010736   0.016633  -0.645
## PublisherNamco Bandai Games                    0.105711   0.013077   8.084
## PublisherNintendo                              0.275148   0.013645  20.165
## PublisherSega                                  0.036321   0.013771   2.637
## PublisherSony Computer Entertainment           0.036677   0.013647   2.688
## PublisherSquare Enix                           0.105207   0.016487   6.381
## PublisherTake-Two Interactive                 -0.024906   0.014396  -1.730
## PublisherTecmo Koei                            0.070909   0.014834   4.780
## PublisherTHQ                                  -0.020208   0.013359  -1.513
## PublisherUbisoft                              -0.015140   0.012971  -1.167
```

```
## PublisherUnknown                                  0.019520   0.016429   1.188
## PublisherWarner Bros. Interactive Entertainment -0.025186   0.016133  -1.561
## simple_platformNintendo                           0.033590   0.005053   6.648
## simple_platformOther                              0.009512   0.015751   0.604
## simple_platformPC                                 0.009334   0.008152   1.145
## simple_platformSega                               0.154317   0.014558  10.600
## simple_platformSony                               0.026094   0.005015   5.203
##                                                  Pr(>|t|)
## (Intercept)                                      9.14e-05 ***
## log_NA_Sales                                      < 2e-16 ***
## log_EU_Sales                                      < 2e-16 ***
## log_Other_Sales                                  1.80e-11 ***
## GenreAdventure                                   0.872124
## GenreFighting                                    0.009051 **
## GenreMisc                                        0.445949
## GenrePlatform                                    0.000331 ***
## GenrePuzzle                                      0.076499 .
## GenreRacing                                      0.149762
## GenreRole-Playing                                 < 2e-16 ***
## GenreShooter                                     0.039261 *
## GenreSimulation                                  0.001011 **
## GenreSports                                      0.000656 ***
## GenreStrategy                                    0.001078 **
## PublisherActivision                              0.035715 *
## PublisherAtari                                   0.550691
## PublisherCapcom                                  2.28e-12 ***
## PublisherDisney Interactive Studios              0.097197 .
## PublisherEidos Interactive                       0.846065
## PublisherElectronic Arts                         0.003923 **
## PublisherKonami Digital Entertainment            1.65e-09 ***
## PublisherMidway Games                            0.518633
## PublisherNamco Bandai Games                      7.26e-16 ***
## PublisherNintendo                                 < 2e-16 ***
## PublisherSega                                    0.008370 **
## PublisherSony Computer Entertainment             0.007213 **
## PublisherSquare Enix                             1.86e-10 ***
## PublisherTake-Two Interactive                    0.083667 .
## PublisherTecmo Koei                              1.78e-06 ***
## PublisherTHQ                                     0.130400
## PublisherUbisoft                                 0.243159
## PublisherUnknown                                 0.234819
## PublisherWarner Bros. Interactive Entertainment 0.118537
## simple_platformNintendo                          3.18e-11 ***
## simple_platformOther                             0.545922
## simple_platformPC                                0.252273
## simple_platformSega                               < 2e-16 ***
## simple_platformSony                              2.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1337 on 7519 degrees of freedom
## Multiple R-squared:  0.436,  Adjusted R-squared:  0.4332
## F-statistic:   153 on 38 and 7519 DF,  p-value: < 2.2e-16
```

The results indicate that there are many significant differences within the Publisher categorical variable and also among the platforms. Role-playing is signficant among the genres along with other groups. The continuous variables also show a significant relationship Japanese sales. The overall variance explained is 43% which might be considered low in a finance context.

Parameter Tuning

Below we set up the parameter tuning for our boosted random forest

```
#grid development
grid<-expand.grid(.n.trees=seq(100,500,by=100),.interaction.depth=seq(1,4,by=1),.shrinkage=c(.001,.01,.
                  .n.minobsinnode=10)
control<-trainControl(method = "cv", repeats=10)
```

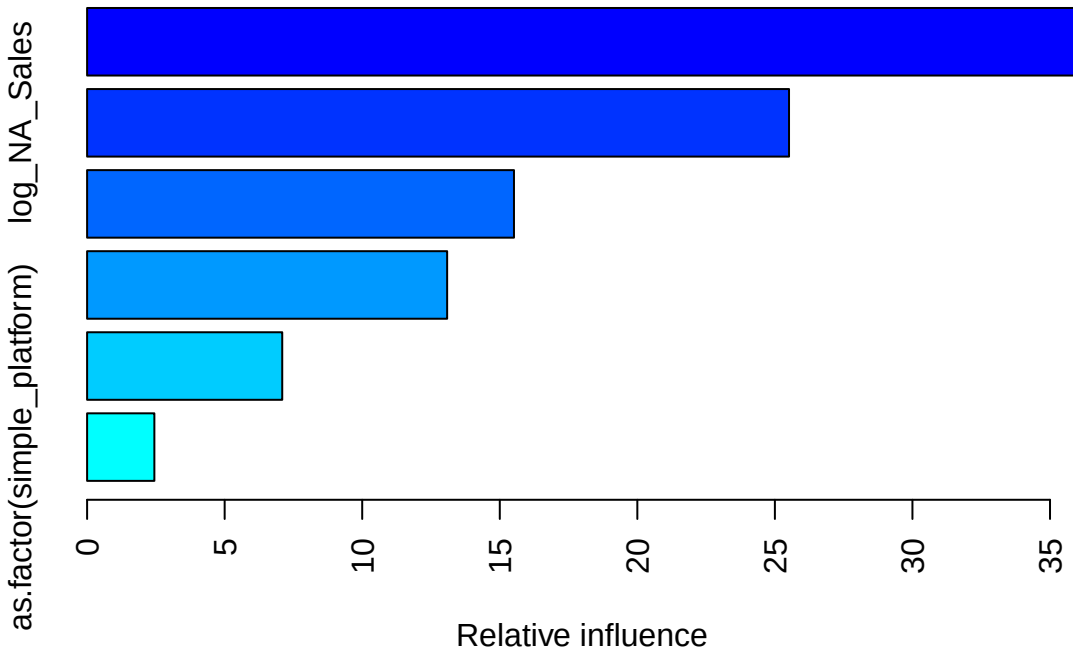
```
## Warning: `repeats` has no meaning for this resampling method.
```

Model Training

Now we train our model using the settings from the parameter tuning

Based on the feedback from the tuning we can use these values to for our model. Followed by an output that shows us the importance of easch variable.

```
gbm.sales<-gbm(log_JP_Sales~log_NA_Sales+log_EU_Sales+log_Other_Sales+as.factor(Publisher)+as.factor(Ge
               shrinkage = .1,distribution = 'gaussian')
 summary(     gbm.sales, cBars = 40,method = relative.influence, las = 3)
```




```
##                                                 var   rel.inf
## as.factor(Publisher)             as.factor(Publisher) 36.346704
## log_NA_Sales                             log_NA_Sales 25.513171
## log_EU_Sales                             log_EU_Sales 15.515523
## as.factor(Genre)                     as.factor(Genre) 13.086854
## log_Other_Sales                       log_Other_Sales  7.092988
## as.factor(simple_platform) as.factor(simple_platform)  2.444761
```

It appears that the publisher is the most important factor in determining sales in Japan having almost half of the influence. North American and European Sales also have high importance. Other sales and platofrm

do not seem to be that important.

Model Evaluation

Below, we are going to use or test data in order to evaluate the model

```
gbm.test<-predict(gbm.sales,newdata = test,n.trees = 500,interaction.depth = 4,
                  shrinkage = .1,distribution = 'gaussian')
```

Residuals

First we take a look at the residuals which is useful for comparing models.

```
gbm.resid<-gbm.test-test$log_JP_Sales
mean(gbm.resid^2)
```

```
## [1] 0.01431338
```

We want this number to be as low as possible as it indicates less error in the model.

Comparing Summary Statistics

Another tool for examining models is to compare the descriptive statistics of the predicted model with the test data.

```
exp(summary(gbm.test-1))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3088  0.3687  0.3730  0.3933  0.4012  3.1516
```

```
exp(summary(test$log_JP_Sales-1))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3679  0.3679  0.3679  0.3940  0.3863  4.1276
```

The values are similar at the mean but really begin to struggle at higher values. This indicates that the model is good at predict typical or average sales but struggle with blockbuster or highly successful games.

Mean Absolute Error

```
exp(mean(abs((test$log_JP_Sales-gbm.test))-1))
```

```
## [1] 0.3881731
```

What this tells us is that the avaerage difference between the actual and predicted values is 0.3880155. In order for this to make sense we have to convert this into millions of dollars which becomes $388,015. We need to compare this value to the mean absolute error to the mean of our model

```
exp(mean(abs(mean(train$log_JP_Sales)-test$log_JP_Sales)-1))
```

```
## [1] 0.4055417
```

The value about is 0.4055417 and when we multiplied it by one million we get $405,541.7 which is close to the calculated MAE. We want there to be a large difference between these two numbers. If not it means are model is not much better than the mean at predicting future values.

Correlation, Plots, and Confidence Interval

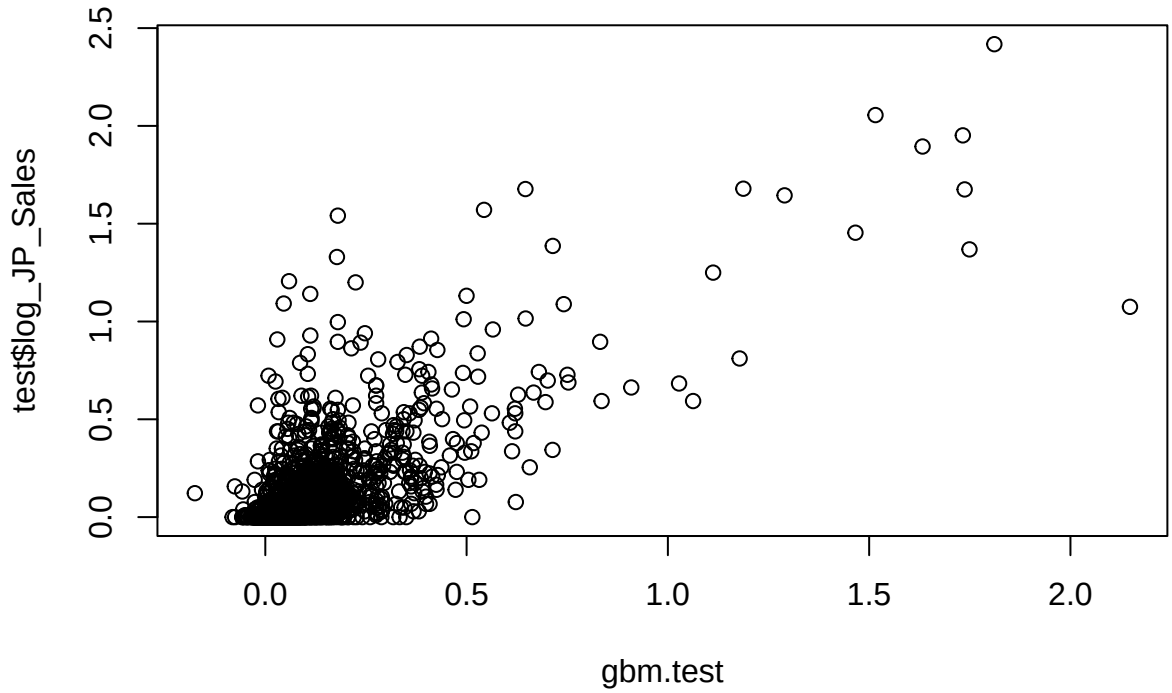
We will now look at the correlation between the predicted and actual values

```
cor(gbm.test,test$log_JP_Sales)
```

```
## [1] 0.7588277
```

The correlation is high but we need to confirm this with a visual.

```
plot(gbm.test,test$log_JP_Sales)
```



There is a trend but the problem is all the values that are lumped together near the bottom. This indicates that most games dont make a lot of money but there are a lot of exceptions. In other words, something else explains Japanese sales in addition to what is in this model.

Interpreting/Conclusions

The Data

The publisher is the most important variable in determining Japanese sales

There are a lot of games that do not sale that well and a handful of games that are all-time greats

Role-Playing games are a big hit in Japan along with games by Electronic Arts

The Code

The EDA provides insights in global sales as well as sales in Japan

This analysis provides insight into which platforms, genres, and publishers to focus on boosting sales in Japan

The model predicts sales of video games in Japan

Limitations

The majority of the data came from the 1990's and 2000's making it hard to predict Japanese sales in the 2020's

Individual games were all unique which means it was not reasonable to include this variable in the model

The model does not do good job estimating lower selling games