

# Predict Game Sales Project

*Skyler Bowthorpe*

*11/24/2019*

##Introduction This report is a capstone project for the Data Science: Capstone course on the edx platform. This dataset has information about videogames including things like genre, publisher, and sales. The goal is to predict the number of sales a game will get based on the basic info provided. In a business context this is very useful information as developing games it time consuming and expensive.

First this script sets up the necessary libraries and imports the dataset. I will include the CSV file in the GitHub repository.

## Data visualization

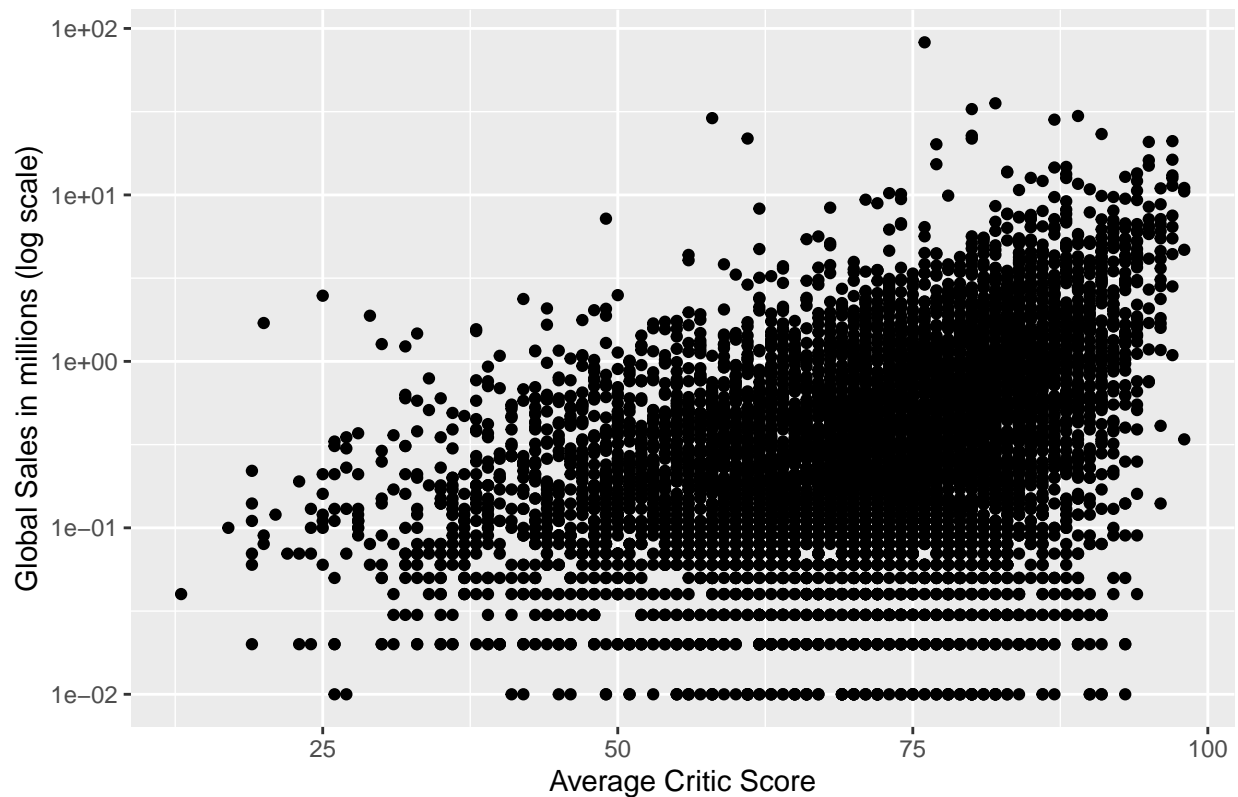
Next I will start exploring and visualizing the dataset. I want to clarify the variables that may help predict sales so they can be included in the prediction model.

```
#do critic scores effect sales?  
#My hypothesis is that higher critic scores may influence consumers to buy certain games more.  
cor(Global_Sales,Critic_Score)
```

```
## [1] 0.2369535
```

```
#This is a lower correlation than I was expecting  
#Plot the relationship to further understand the relationship.  
ggplot()+  
  geom_point(aes(Critic_Score,Global_Sales))+  
  scale_y_continuous(trans = "log10")+  
  xlab("Average Critic Score")+  
  ylab("Global Sales in millions (log scale)")+  
  ggtitle("The correlation between critic scores and global sales")
```

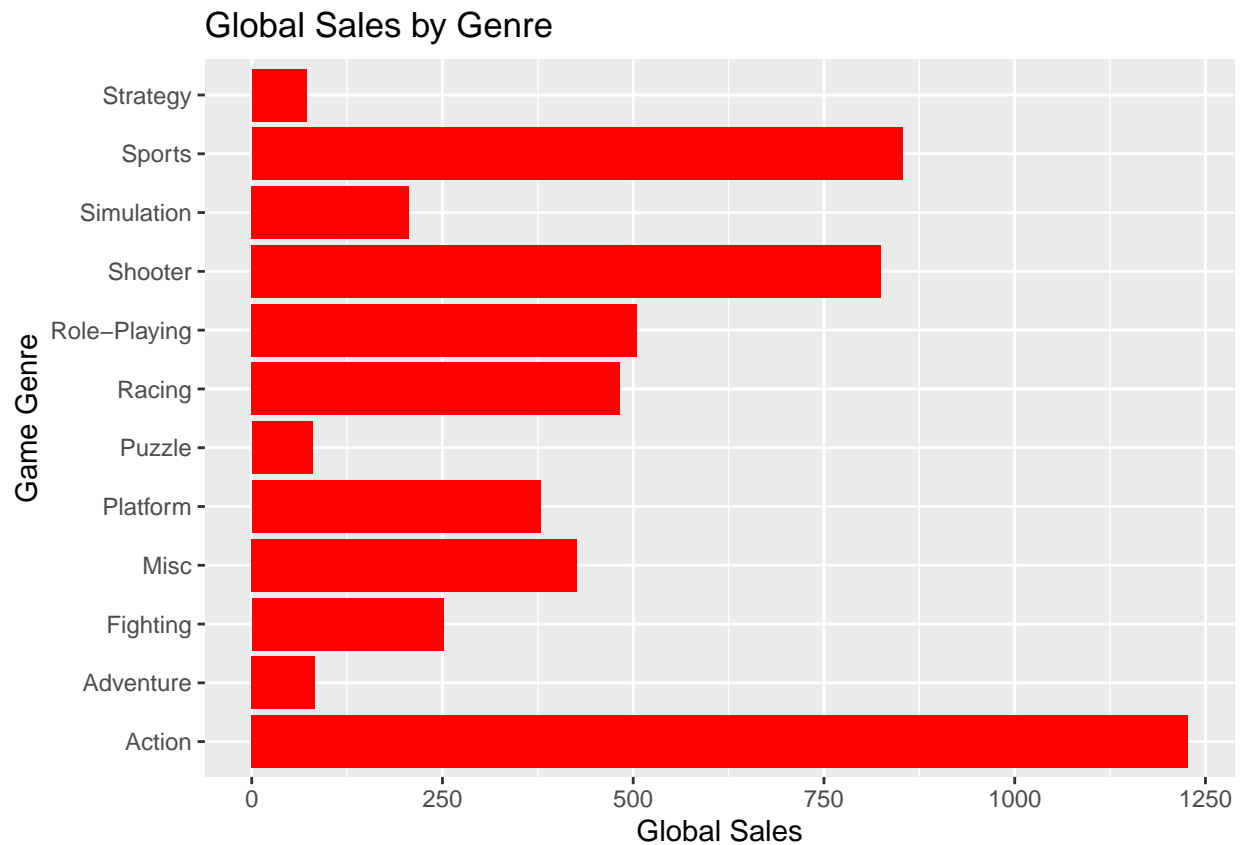
The correlation between critic scores and global sales



```
dat <- dat[!(names(dat) %in% c("Critic_Score", "Critic_Count",
                              "User_Score", "User_Count",
                              "Developer", "Year_of_Release"))]

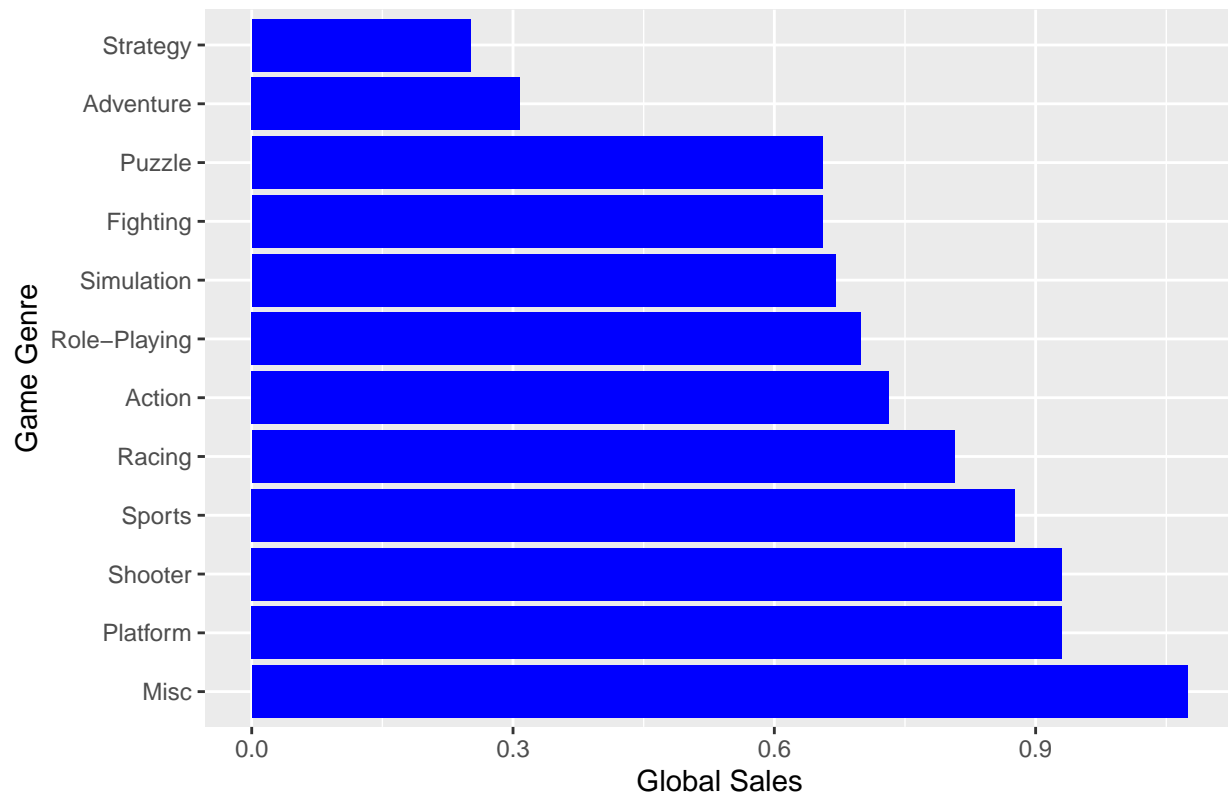
# Games seem to appeal to people differently.
# This is a good start let's explore other variables that may help predict global sales

#What are the most popular Genres? Do games in those genres have high sales individually?
ggplot(dat, aes(Genre,Global_Sales)) +
  geom_bar(fill="red", stat="identity")+
  coord_flip()+
  xlab("Game Genre")+
  ylab("Global Sales")+
  ggtitle("Global Sales by Genre")
```



```
# The data shows that action games have the highest
# collective sales but also a very high number of action titles.
average_rev_by_genre <- aggregate(Global_Sales~Genre,dat,mean)
arrange_by_rev2 <- arrange(average_rev_by_genre,desc(Global_Sales))
arrange_by_rev2$Genre=factor(arrange_by_rev2$Genre,levels=arrange_by_rev2$Genre)
ggplot(arrange_by_rev2,aes(Genre,Global_Sales))+
  geom_bar(fill="blue", stat="identity")+
  coord_flip()+
  xlab("Game Genre")+
  ylab("Global Sales")+
  ggtitle("Average Game Sales by Game Genre")
```

Average Game Sales by Game Genre



*# While the Adventure Genre had the most sales; Miscellaneous, Platforming, and Shooter games had the higher per-game performance.  
# What are some of the games that are in the Miscellaneous category?*

```
head(dat %>% filter(Genre == "Misc"), n = 10)
```

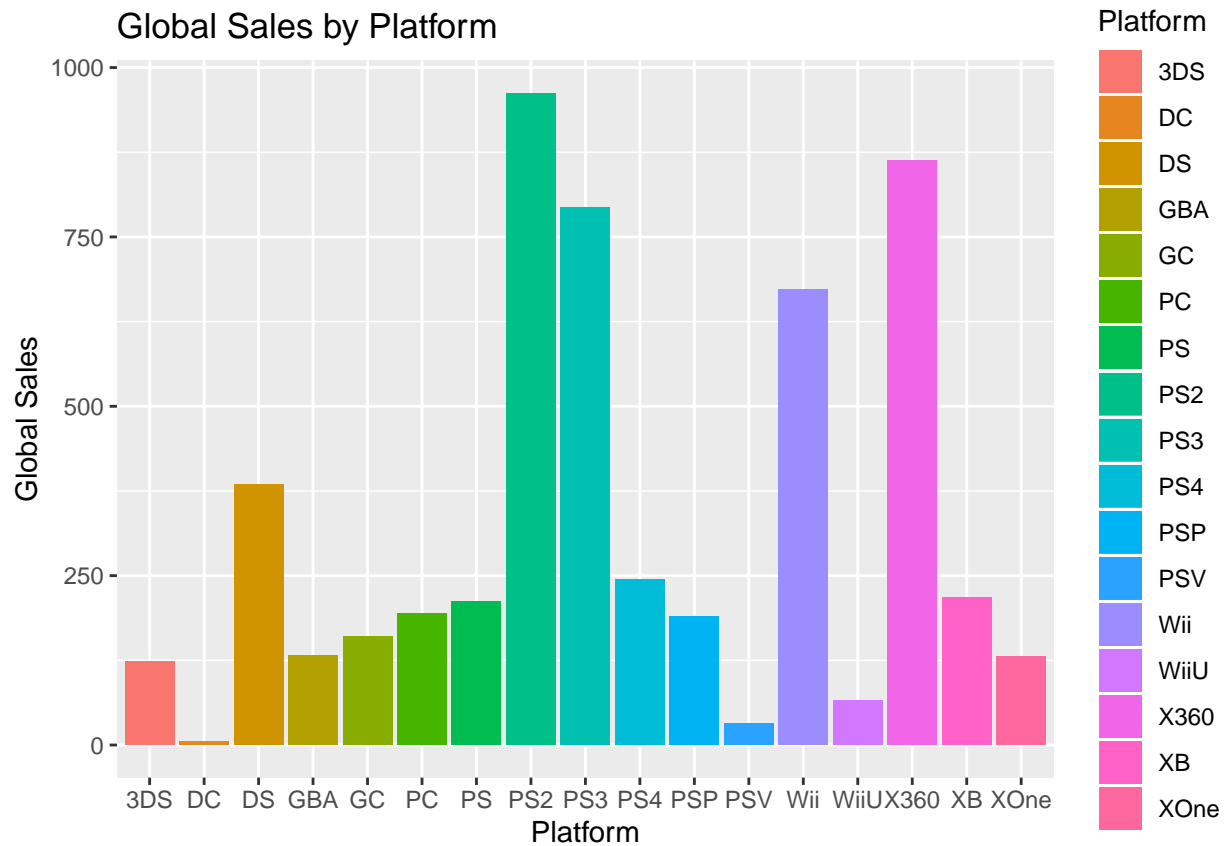
```
##           Name Platform Genre
## 1           Wii Play      Wii  Misc
## 2   Kinect Adventures!  X360  Misc
## 3 Brain Age: Train Your Brain in Minutes a Day      DS  Misc
## 4           Just Dance 3      Wii  Misc
## 5           Just Dance 2      Wii  Misc
## 6       Mario Party DS      DS  Misc
## 7           Wii Party      Wii  Misc
## 8       Mario Party 8      Wii  Misc
## 9           Just Dance      Wii  Misc
## 10          Just Dance 4      Wii  Misc
##           Publisher Global_Sales Rating
## 1           Nintendo      28.92      E
## 2 Microsoft Game Studios      21.81      E
## 3           Nintendo      20.15      E
## 4           Ubisoft      10.12     E10+
## 5           Ubisoft       9.44     E10+
## 6           Nintendo       8.91      E
## 7           Nintendo       8.38      E
## 8           Nintendo       8.27      E
```

```
## 9          Ubisoft      7.20  E10+
## 10         Ubisoft      6.76  E10+
```

*#Misc includes party, music and learning type games.*

*#charts comparing platfoms*

```
ggplot(dat, aes(Platform,Global_Sales,fill=Platform))+
  geom_bar(stat="identity")+
  xlab("Platform")+
  ylab("Global Sales")+
  ggtitle("Global Sales by Platform")
```



*# For more clarity I will categorize the platforms into major companies*

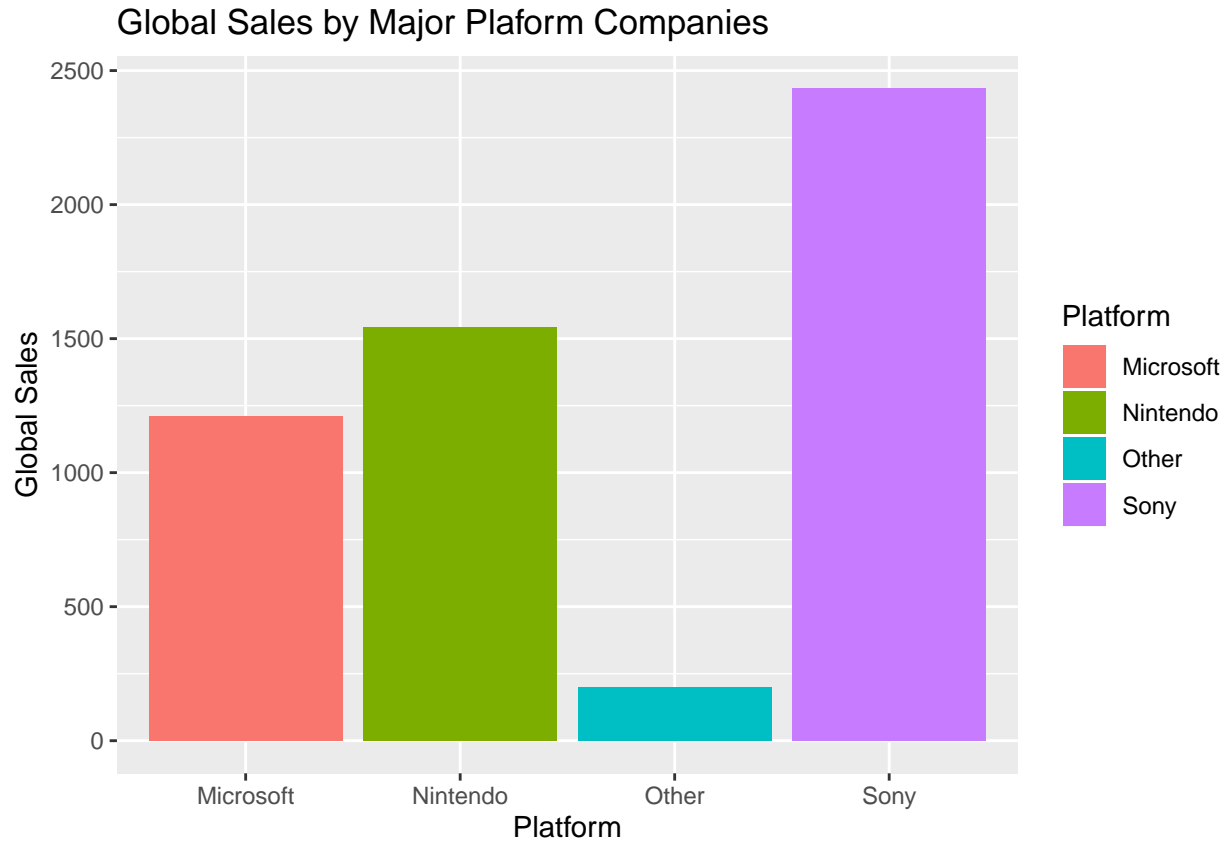
*# New generatoins of game systems replace the old every few years.*

```
dat$Platform <- as.character(dat$Platform)
dat$Platform[dat$Platform %in% c("PS","PS2","PS3","PS4","PSP","PSV")] <- "Sony"
dat$Platform[dat$Platform %in% c("XB","XOne","X360")] <- "Microsoft"
dat$Platform[dat$Platform %in% c("Wii","NES","GB","DS",
                                "SNES","GBA","3DS","N64",
                                "GC","WiiU")] <- "Nintendo"
dat$Platform[!(dat$Platform %in% c("Nintendo","Sony","Microsoft"))] <- "Other"
dat$Platform <- as.factor(dat$Platform)
```

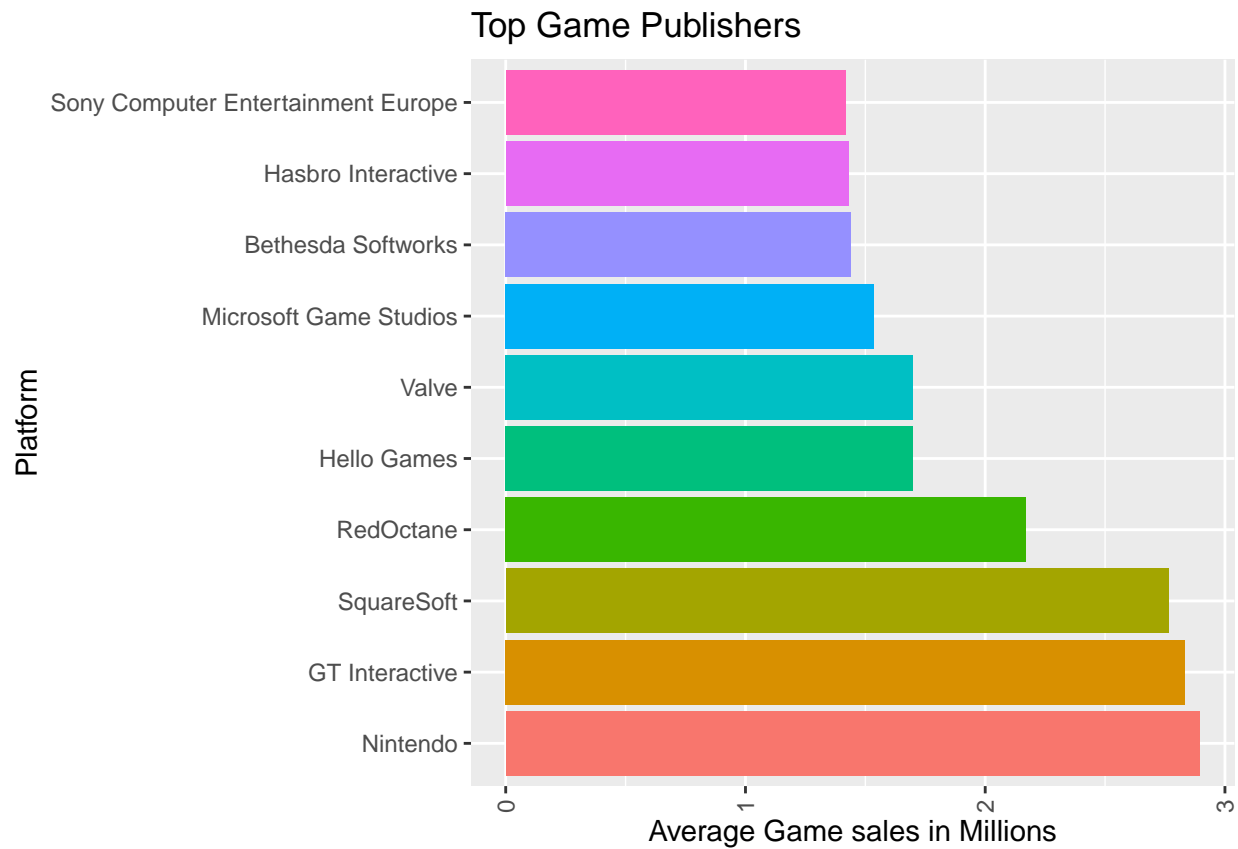
*#Much cleaner chart comparing platform companies.*

```
ggplot(dat, aes(Platform,Global_Sales,fill=Platform))+
```

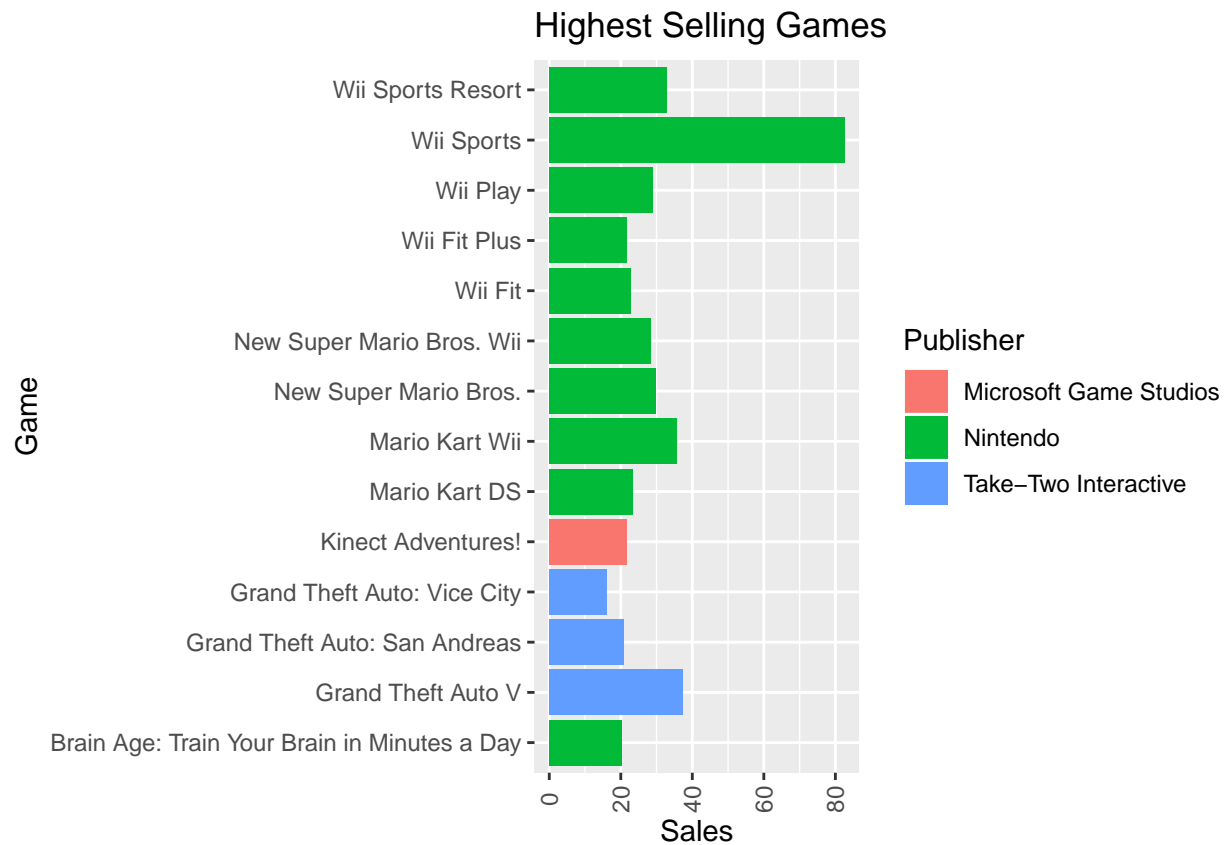
```
geom_bar(stat="identity")+
xlab("Platform")+
ylab("Global Sales")+
ggtitle("Global Sales by Major Platform Companies")
```



```
#There are too many publishers to plot all of them. I will show just the top ten.
top_publishers <- aggregate(Global_Sales~Publisher,dat,mean) #average of game sales per publisher
arrange_by_rev3 <- arrange(top_publishers,desc(Global_Sales))
arrange_by_rev3$Publisher = factor(arrange_by_rev3$Publisher, levels = arrange_by_rev3$Publisher)
ggplot(head(arrange_by_rev3,10),aes(Publisher,Global_Sales,fill=Publisher))+
  geom_bar(stat="identity")+
  coord_flip()+
  labs(x="Platform",y="Average Game sales in Millions")##Flipping the axis to it is more readable
  theme(axis.text.x=element_text(angle=90,vjust=0.5),legend.position="none")+
  ggtitle("Top Game Publishers")
```

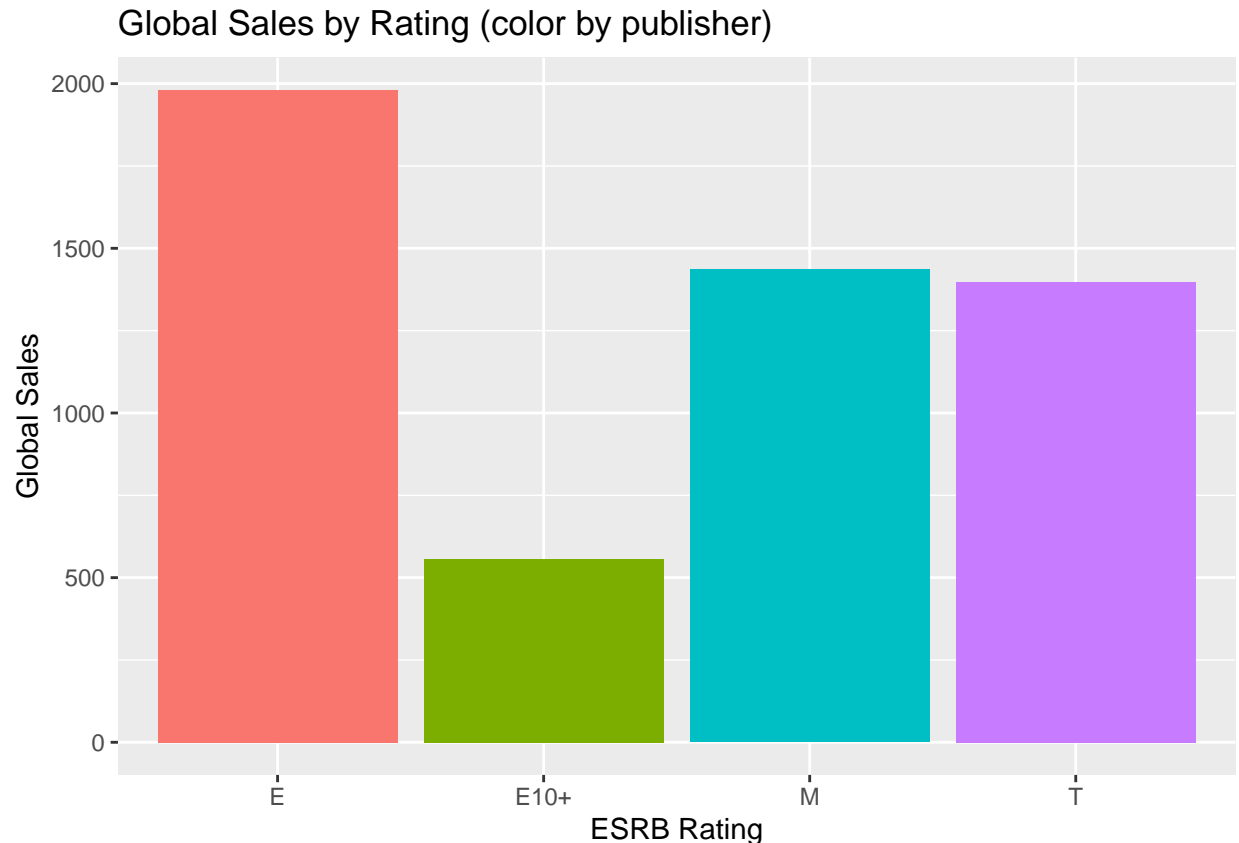


```
#Publishers vs highest sales games.
top_games_by_publisher = dat %>% select(Name,Global_Sales,Publisher) %>% arrange(desc(Global_Sales))
ggplot(head(top_games_by_publisher,15),aes(Name,Global_Sales,fill=Publisher))+
  geom_bar(stat="identity")+
  coord_flip()+
  labs(x="Platform",y="Average Game sales in Millions")+
  theme(axis.text.x = element_text(angle=90,vjust=0.5))+
  ggtitle("Highest Selling Games")+
  labs(x="Game",y="Sales")
```



```
#What effect the parent safety ratings have on sales?
ggplot(subset(dat, Rating %in% c("E", "E10+", "T", "M")),
  aes(Rating, Global_Sales, fill=Rating)) +
  geom_bar(stat="identity") +
  theme(legend.position="none") +
  xlab("ESRB Rating") +
  ylab("Global Sales") +
  ggtitle("Global Sales by Rating (color by publisher)")
```





*#That's enough visualization. Lets build the model*

## Prediction model

This model will have limited accuracy because I am trying to predict a continuous variable. To help with this, I have rounded the sales to the closest million.

I ran into issues while building, I did not have a powerful enough machine to use my initial approach to building the model which is surprising because I chose that method because it is less hardware intensive than other methods. I will include a comment of the approach I would have used if my first approach was possible.

#Methodology 1. Split dataset into 80% training set and 20% training 2. Create a naïve model that is just the average of all game sales for a basis of comparison 3. Add a Genre effect that predicts higher sales if a game is in a popular genre (I learned this does not inform the prediction very well) 4. Learn I don't have enough ram to keep adding effects to the model (This project is about learning not about getting everything right) 5. Pivot and training and tuning a general linear model to predict sales 6. Test gml with the test set to measure accuracy

```
#Create training and test partitions
index <- createDataPartition(y=dat$Global_Sales, p=0.8, list=FALSE)
train <- dat[index,]
test <- dat[-index,]
validate <- test
# Create a na?ve set for comparison
RMSE <- function(true_sales, predicted_sales){
```

```

sqrt(mean((true_sales - predicted_sales)^2))}

mu_hat <- mean(train$Global_Sales)
naive_rmse <- RMSE(test$Global_Sales, mu_hat)
predictions <- rep(2.5, nrow(test))
rmse_results <- data_frame(method="Just the average",RMSE=naive_rmse)

#Rating effect
mu <- mean(train$Global_Sales)

genre_avgs <- train %>%
  group_by(Genre) %>%
  mutate(genre_effect = mean(Global_Sales - mu))

predicted_sales <- mu + test %>%
  left_join(genre_avgs, by="Genre") %>%
  .$genre_effect
model_1_rmse <- RMSE(predicted_sales, test$Global_Sales)
rmse_results <- bind_rows(rmse_results,
  data_frame(method="ESRB Rating Effect Model",
    RMSE = model_1_rmse ))
rmse_results %>% knitr::kable()

```

method	RMSE
Just the average	1.673764
ESRB Rating Effect Model	1.673851

*#This is how I would have continued to test and tune the model but I received this error:  
 #Error: Evaluation error: cannot allocate vector of size 3.4 Gb. and it crashed my computer.  
 #I do not know of a less memory intensive way of continuing this method...*

```

#Add platform effect to the model
#platform_avgs <- train %>%
#group_by(Platform) %>%
#mutate(platform_effect = mean(Global_Sales - mu))

#predicted_sales <- test %>%
  #full_join(genre_avgs,test, by = "Genre") %>%
  #full_join(platform_avgs,test, by = "Genre") %>%
  #mutate(pred = mu + genre_effect + platform_effect) %>%
  #.$pred
#model_2_rmse <- RMSE(predicted_sales, test$Global_Sales)
#rmse_results <- bind_rows(rmse_results,
  #data_frame(method="Rating + Platform Effects Model",
    #RMSE = model_2_rmse ))
#rmse_results %>% knitr::kable()

#New Method, hopefully my computer will handle it.
lambda <- seq(2,-2,length=100)
alpha <- seq(0,1,length=10) # I got help on a forum for this bit.
trControl <- trainControl(method = "CV",number= 10,repeats = 5)

```

```

grid = expand.grid(.alpha=alpha,.lambda=lambda)
train_model <- train(Global_Sales ~ Platform + Genre + Rating,
                     data=train,
                     method = 'glmnet',
                     tuneGrid=grid,
                     trControl=trControl,
                     standardize=TRUE,
                     maxit= 10000)
train_model$bestTune

##      alpha      lambda
## 44         0 0.5462277

final <- train_model$finalModel

#Make predictions with tuned model
predicted_global_sales <- predict(train_model,test,s=final$lambda.min)
#using min lamda found in train_model
check <- data.frame(Game=validate$Name, Actual=validate$Global_Sales)
prediction <- round(predicted_global_sales,2)

#check the output of GLM model
check <- check[1:length(prediction),]
check$Predicted <- abs(prediction)
check$diff <- abs(check$Predicted - check$Actual)
RMSE_glm <- sqrt(mean(check$diff^2))
# using a different method of arriving at RMSE but the answer is the same
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="GLM model",
                                      RMSE = RMSE_glm ))
rmse_results %>% knitr::kable()

```

method	RMSE
Just the average	1.673764
ESRB Rating Effect Model	1.673851
GLM model	1.649622

## ##Conclusion

This data suggests that you can directionally predict global sales of a videogame but the medium appeals to people differently and it is difficult to do with any certainty. Thank you for the great opportunity to learn about data science and machine learning I have enjoyed these courses. This report was a great way to apply the lessons from those courses.