In [0]:
```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *

from sklearn.tree import DecisionTreeClassifier # Decision Tree
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB
, CategoricalNB # Decision Tree
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA

from sklearn import metrics
from sklearn.preprocessing import StandardScaler

from sklearn.cluster import AgglomerativeClustering

from sklearn.model_selection import train_test_split # simple TT split
cv
from sklearn.model_selection import KFold # k-fold cv
from sklearn.model_selection import LeaveOneOut #LOO cv
from sklearn.model_selection import cross_val_score # cross validation
metrics
from sklearn.model_selection import cross_val_predict # cross validati
on metrics
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import mean_squared_error, r2_score

from sklearn.model_selection import GridSearchCV

from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture

from sklearn.metrics import silhouette_score

import scipy.cluster.hierarchy as sch
from matplotlib import pyplot as plt

%precision %.7g
%matplotlib inline
```

In [99]:
```
vg = pd.read_csv("https://raw.githubusercontent.com/mrNKit25/CSV/maste
r/vgsales.csv")
vg.head()
```

Out[99]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22

1. What type of games are more popular?

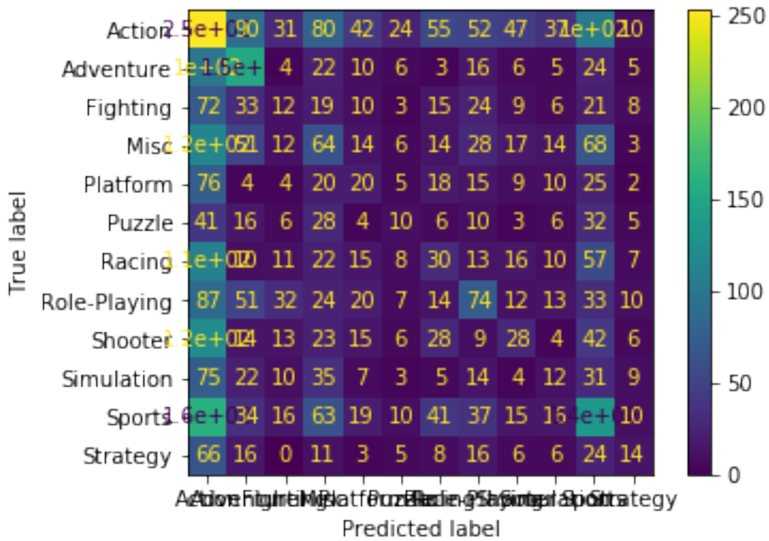
- The variables we are using will be genre, NA sales, EU sales, JP sales, Other sales, and Global sales. We are using standardizing for continuous values. We can use a predictive model using K-Fold decision tree.
- We chose this analysis plan because we are interested in knowing which year and game genre is the most popular. We can check this by taking a look at the sales of each country.
- Two Data visualizations:
  - Point graph that shows the year vs global sales and coloring of genre
  - Point graph that shows the year vs NA sales and coloring of the genre

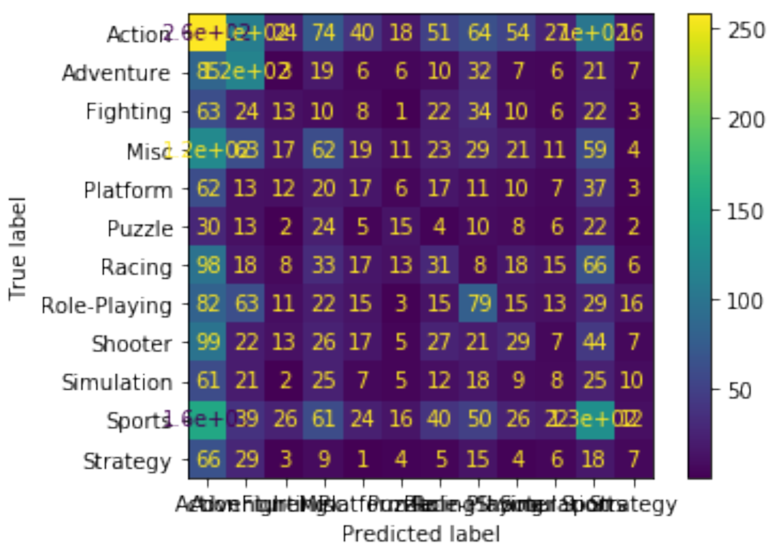
Note: we got rid of year because it was giving us NaN values.

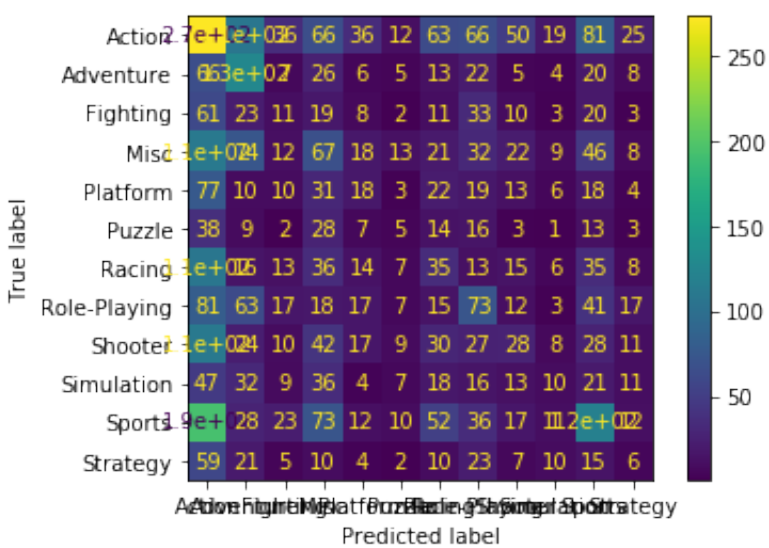
We believe that shooter games are more popular. Looking at the graphs, shooter games have been steadily increasing as the years go by. There appears to be a strong fan base for shooting games as many of the points are clustered together. We used predictive models by using K-Fold decision tree and had a 19% accuracy. We chose this method so we can check how each games are doing in sales. Looking at the confusion matrix, it shows that the data is large and may result in some inaccurate calculations.

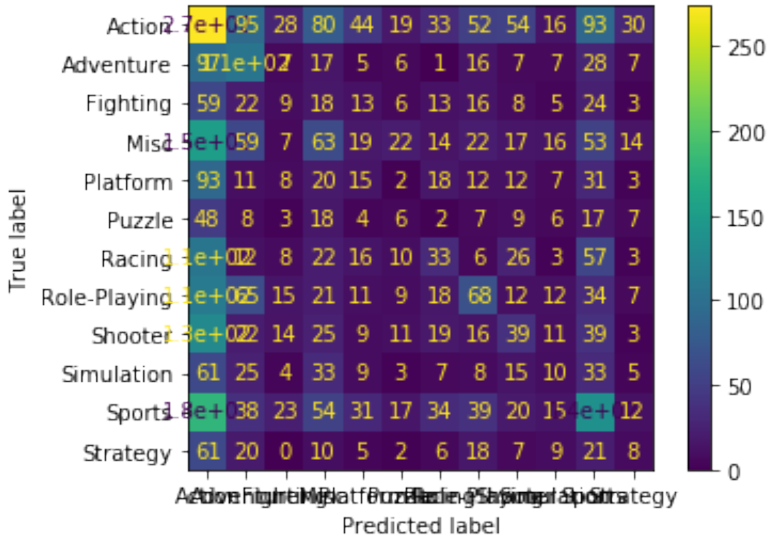
In [0]:
```
predictors = ["NA_Sales", "EU_Sales", "JP_Sales", "Global_Sales", "Oth
er_Sales"]
X = vg[predictors]
y = vg["Genre"]

kf = KFold(4, shuffle = True)

acc = []
depth = []
for train, test in kf.split(X):
    X_train = X.iloc[train,]
    X_test = X.iloc[test,]
    y_train = y[train]
    y_test = y[test]

    z = StandardScaler()
    X_train = z.fit_transform(X_train)
    X_test = z.transform(X_test)

    tree = DecisionTreeClassifier()
    tree.fit(X_train,y_train)

    acc.append(tree.score(X_test,y_test))
    depth.append(tree.get_depth())

    plot_confusion_matrix(tree,X_test,y_test)

print(acc)
print(np.mean(acc))
print(depth)
```

```
[0.1942168674698795, 0.18409638554216867, 0.18606893227283683, 0.185
58688840684504]
0.1874922684229325
[30, 34, 33, 33]
```

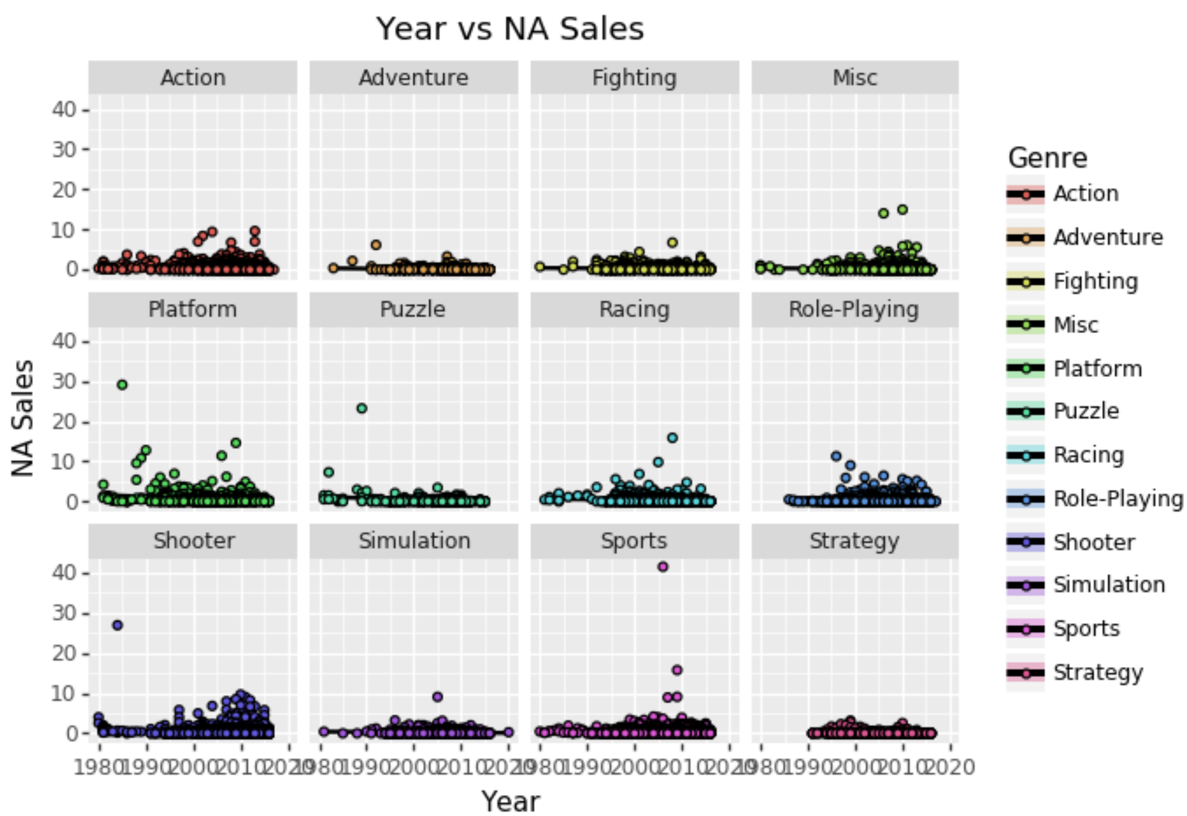

True label
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role-Playing
Shooter
Simulation
Sports
Strategy
Predicted label
250
200
150
100
50
0


True label
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role-Playing
Shooter
Simulation
Sports
Strategy
Predicted label
250
200
150
100
50


True label
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role-Playing
Shooter
Simulation
Sports
Strategy
Predicted label
250
200
150
100
50

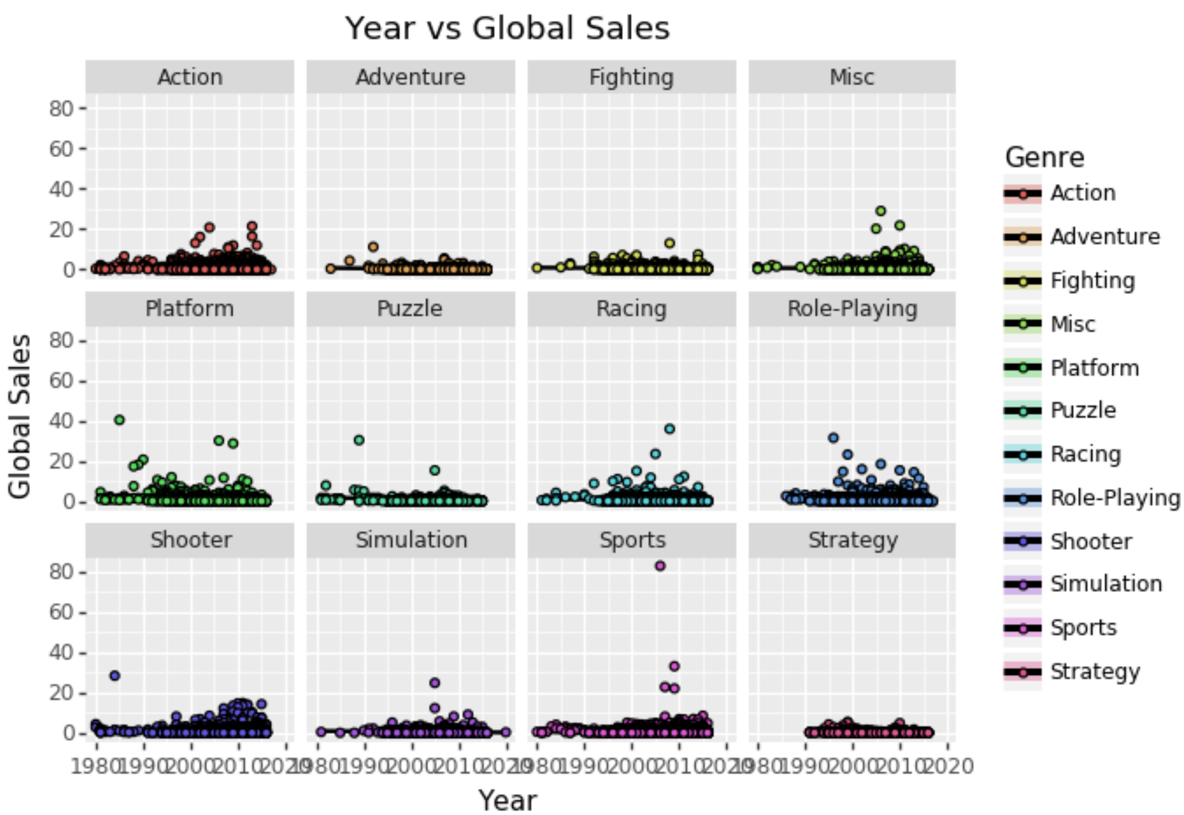


In [0]:
```
(ggplot(vg, aes(x = 'Year', y = 'NA_Sales', fill = 'Genre')) + stat_smooth(method='lm') + facet_wrap('Genre')
+ geom_point() + ggtitle('Year vs NA Sales') + labs(x = 'Year', y = 'NA Sales'))
```



Out[0]: <ggplot: (311967845)>

In [0]:
```
(ggplot(vg, aes(x = 'Year', y = 'Global_Sales', fill = 'Genre')) + sta
t_smooth(method='lm') + facet_wrap('Genre')
+ geom_point() + ggtitle('Year vs Global Sales') + labs(x = 'Year', y
= 'Global Sales'))
```



Out[0]: <ggplot: (311874753)>

2. What type of gaming system do more people use?

- The variables we are using will be Platform, NA Sales, EU Sales, JP Sales, Other sales, and Global Sales. We are using standardizing for continuous values. Prediction model using K-Fold validation.
- We chose this analysis plan because gaming systems are an essential aspect for the gaming community. There are many games that are exclusive only to that specific gaming system (ex. XBOX only has Halo). Therefore, we would like to see which gaming system most people use.
- Two data visualizations:
  - Boxplot graph that shows the global sales and platforms with color fill platform
  - Boxplot graph that shows NA sales and platforms with color fill platform

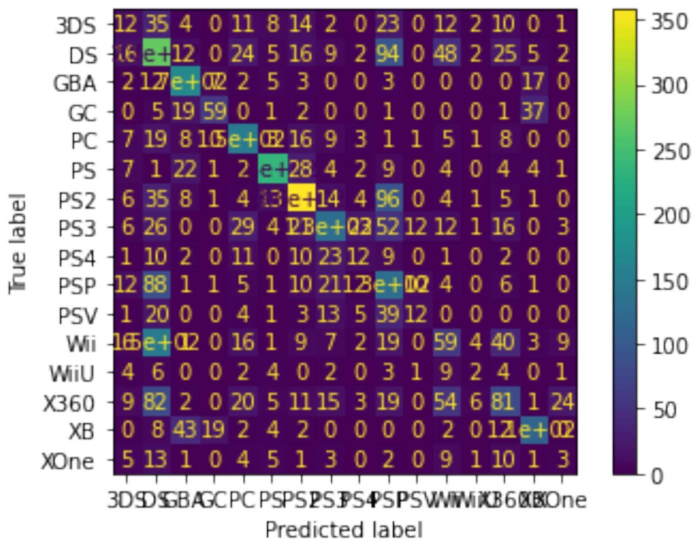
The gaming system that most people used is the Wii. We used a prediction model using K-Fold validation to help us determine this with a mean accuracy score of 46.62%. We can see a colored digaonal line from the top left to the bottom right that shows us that it was able to predict some of the labels correctly. The box plots shown also indicate that more people use Wii due to the higher amount of sales in both North America and Globally.

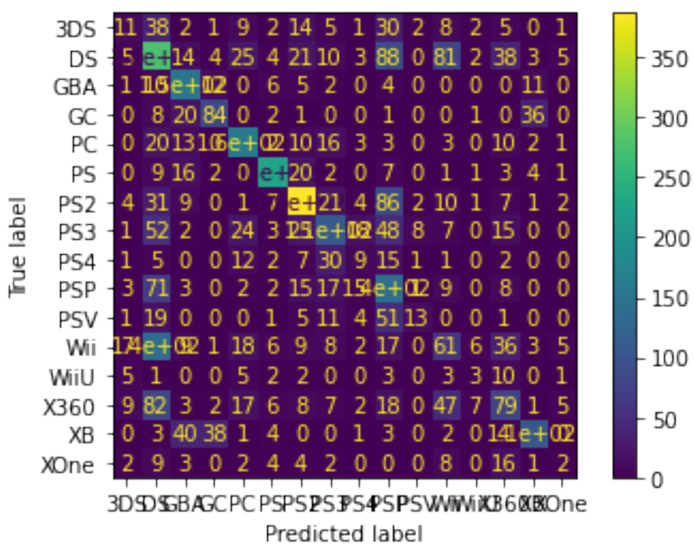
In [0]:
```
data = pd.read_csv("https://raw.githubusercontent.com/mrNKit25/CSV/master/vgsales.csv", index_col = "Platform")
data.drop(["GG", "PCFX", "TG16", "3DO", "WS", "SCD", "NG", "GEN", "DC", "NES", "GB", "2600", "SAT", "N64", "SNES"], inplace = True)
```

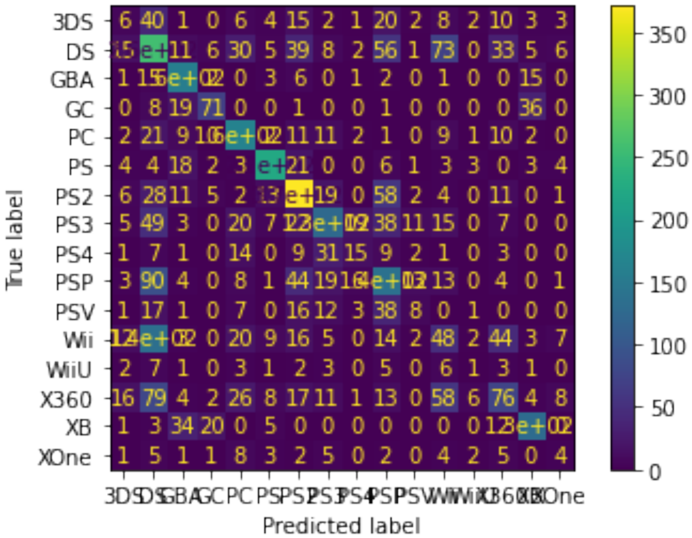
In [0]:
```
data = data.reset_index()
```

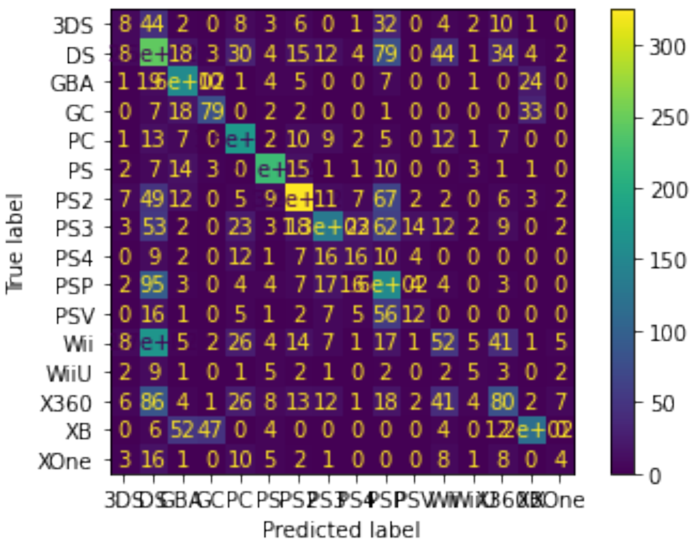
In [0]:
```
predictors = ["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales"]
X = data[predictors]
y = data["Platform"]

kf = KFold(4, shuffle = True)

acc = []
depth = []
for train, test in kf.split(X):
    X_train = X.iloc[train,]
    X_test = X.iloc[test,]
    y_train = y[train]
    y_test = y[test]

    z = StandardScaler()
    X_train = z.fit_transform(X_train)
    X_test = z.transform(X_test)

    tree = DecisionTreeClassifier()
    tree.fit(X_train,y_train)

    acc.append(tree.score(X_test,y_test))
    depth.append(tree.get_depth())

    plot_confusion_matrix(tree,X_test,y_test)

print(acc)
print(np.mean(acc))
print(depth)
```

```
[0.4656468758102152, 0.47083225304640913, 0.4659061446720249, 0.4627
949183303085]
0.4662950479647394
[27, 27, 29, 29]
```

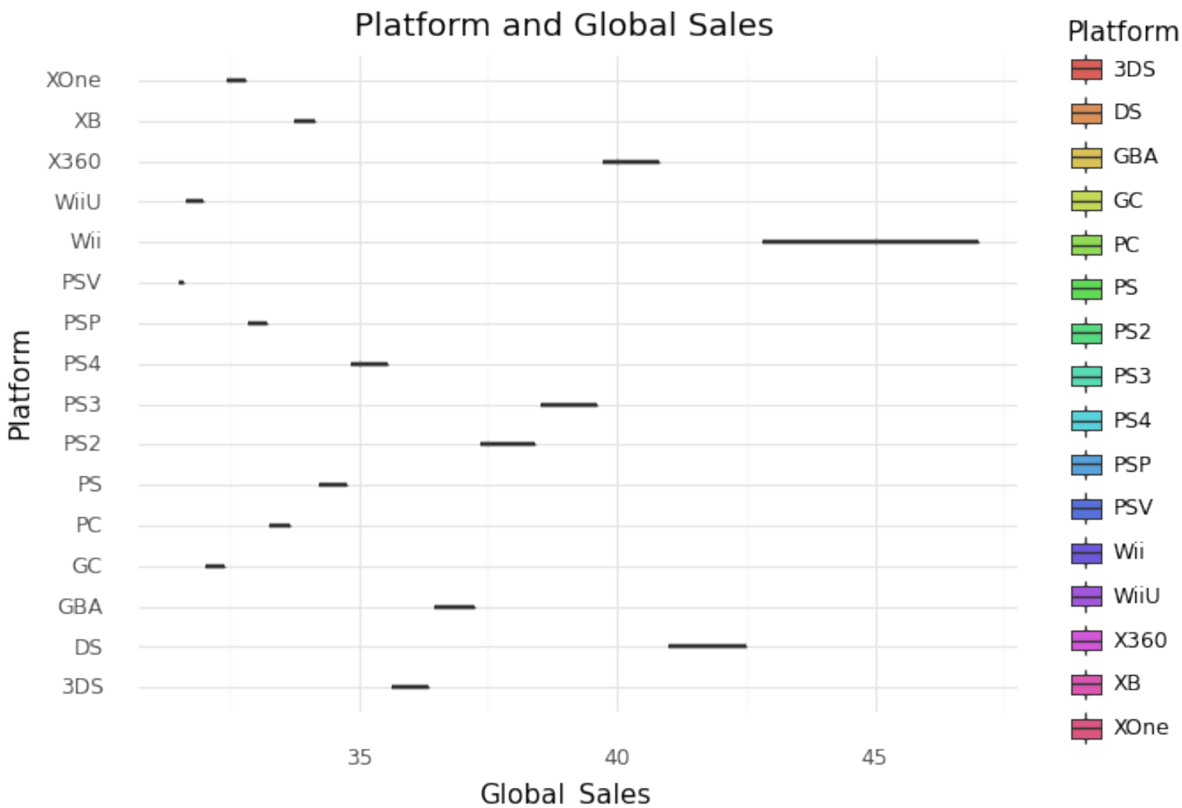





3DS
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
True label
Predicted label
350
300
250
200
150
100
50
0

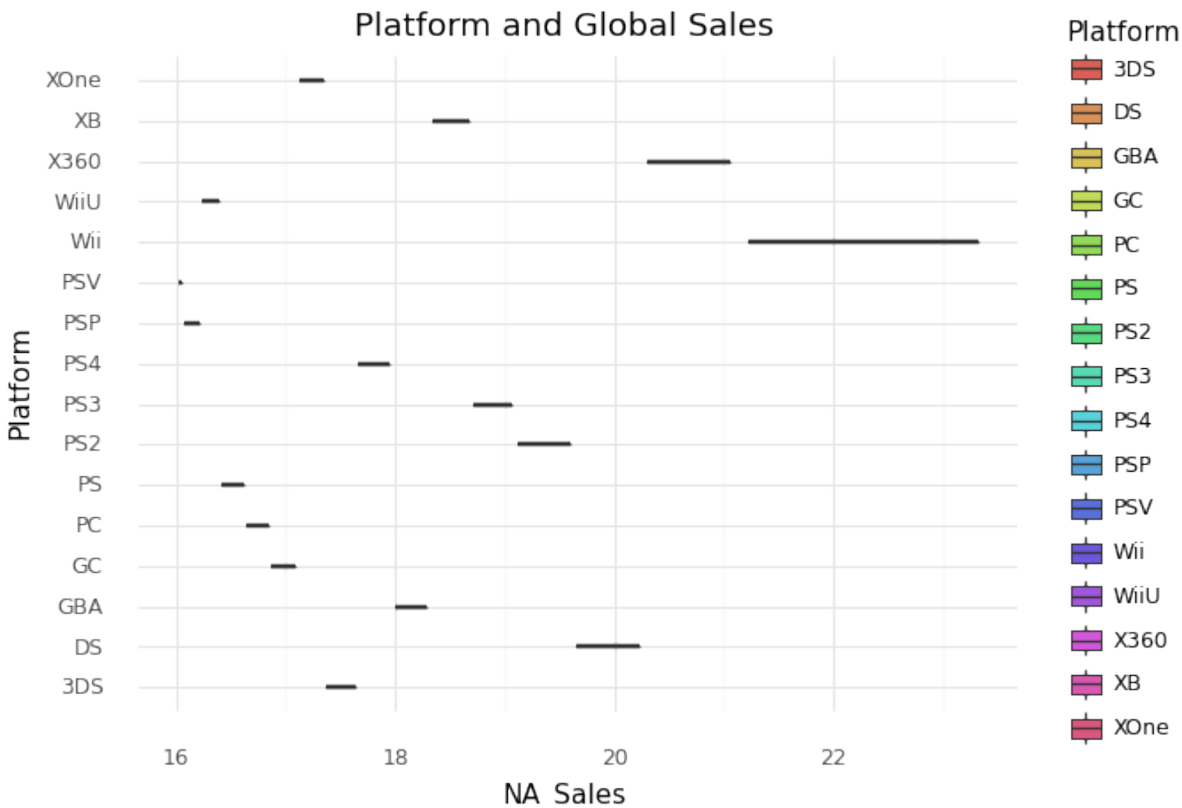

3DS
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
True label
Predicted label
300
250
200
150
100
50
0

In [0]:
```
(ggplot(data, aes(x = "Global_Sales", y = "Platform", fill = "Platform
"))+ geom_boxplot() + ggtitle('Platform and Global Sales')) + theme_mi
nimal()
```



Out[0]: <ggplot: (8791280197198)>

In [0]:
```
(ggplot(data, aes(x = "NA_Sales", y = "Platform", fill = "Platform"))+
geom_boxplot() + ggtitle('Platform and Global Sales')) + theme_minimal
()
```



Out[0]: <ggplot: (-9223363245578862606)>

3. Do people buy more video games over the years?

In [0]:
```
vg.dropna(how='any', inplace=True)
```

- The variables we are using will be Year, NA Sales, EU Sales, JP Sales, and Other Sales. We are using standardizing for continuous values. We used linear regression with train test split.
- We chose this analysis plan because gaming has become increasingly popular over the years. There are many free games offered as well, such as Fortnite and League of Legends. With the rise in e-sports, there may be more people that turn to gaming or online streaming as a career.
- Three data visualizations:
  - Graph that looks at Global Sales over the years
  - Graph that shows NA sales over the years (since we live in NA)
  - Bar graph that shows Global Sales with NA Sales as filled

We got rid of Global Sales because it takes into account of all the other sales.

Initially, we assumed more people bought video games over the years due to technology and manufacturing companies making products less expensive leading to more people buying video game consoles and more video games. Unfortunately, that was not the case. We constructed a linear regression model that shows a negative correlation between year and sales, except EU Sales. Looking at the graph, video games fluctuate throughout the years. With there being different games released each year, it is guaranteed that there will be a positive trend in video game sales in the long run.

In [0]:
```
X = vg[["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales"]]

X_train, X_test, y_train, y_test = train_test_split(vg[predictors], vg
["Year"], test_size=0.2)
```

In [0]:
```
zScore = StandardScaler()
zScore.fit(X_train)
Xz_train = zScore.transform(X_train)
Xz_test = zScore.transform(X_test)
LR = LinearRegression()
LR.fit(Xz_train, y_train)
```

Out[0]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, norma
lize=False)
```

In [0]:
```
pred = LR.predict(Xz_train)
pred[1:10]
```

Out[0]:
```
array([2006.62446102, 2005.50629378, 2005.93789181, 2007.29961933,
       2006.39335881, 2005.6803002 , 2006.81549014, 2006.66187205,
       2007.23309361])
```

In [0]:
```
mean_squared_error(y_train, pred)
```

Out[0]: 31.873179159882984

In [0]:
```
r2_score(y_train, pred)
```

Out[0]: 0.05198587392392473

In [0]:
```
coefficients = pd.DataFrame({"Coef": LR.coef_,"Name": predictors})
coefficients = coefficients.append({"Coef": LR.intercept_, "Name" : "I
ntercept" }, ignore_index = True)
coefficients
```

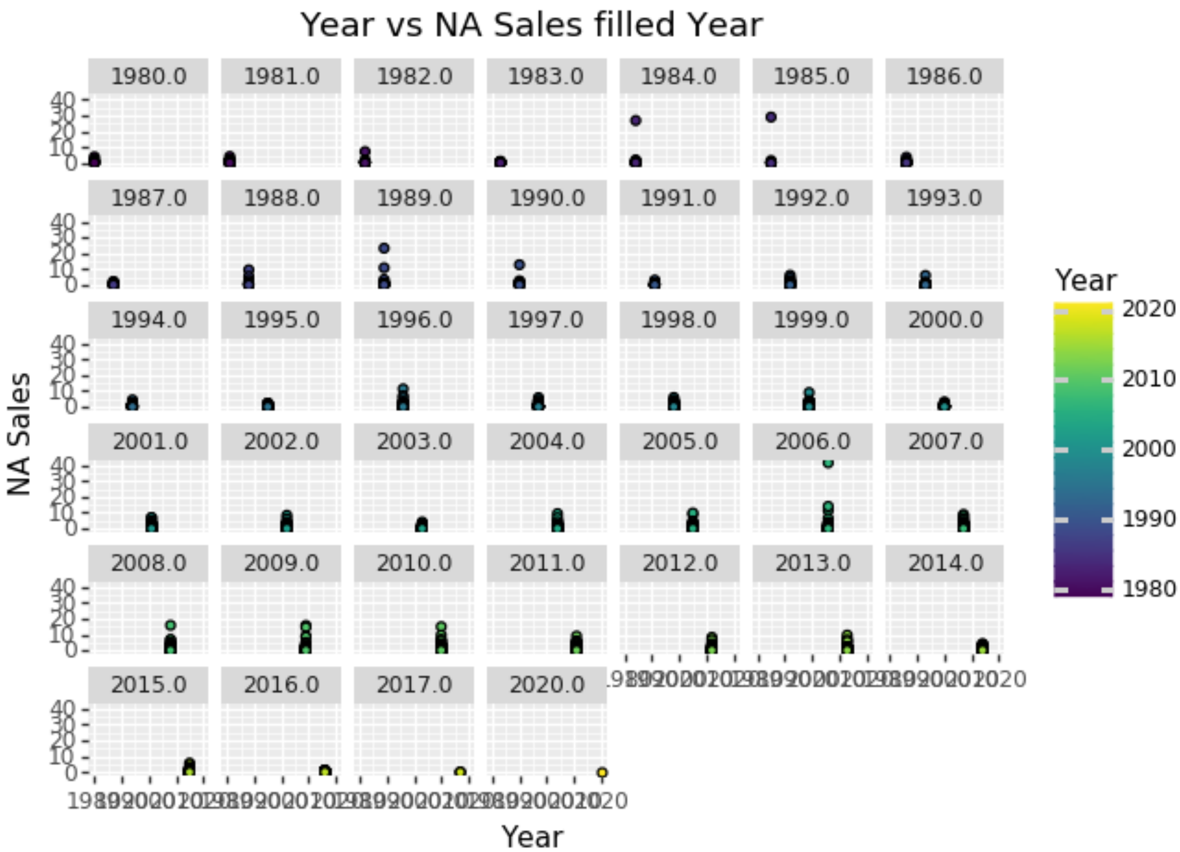
Out[0]:

	Coef	Name
0	-1.102747	NA_Sales
1	1.150336	EU_Sales
2	-0.936950	JP_Sales
3	-0.312116	Global_Sales
4	0.623735	Other_Sales
5	2006.434929	Intercept

In [0]:
```
(ggplot(vg, aes(x = 'Year', y = 'Global_Sales', fill = 'Year')) + stat
_smooth(method='lm') + facet_wrap('Year')
+ geom_point() + ggtitle('Year vs Global Sales filled Year') + labs(x
= 'Year', y = 'Global Sales'))
```

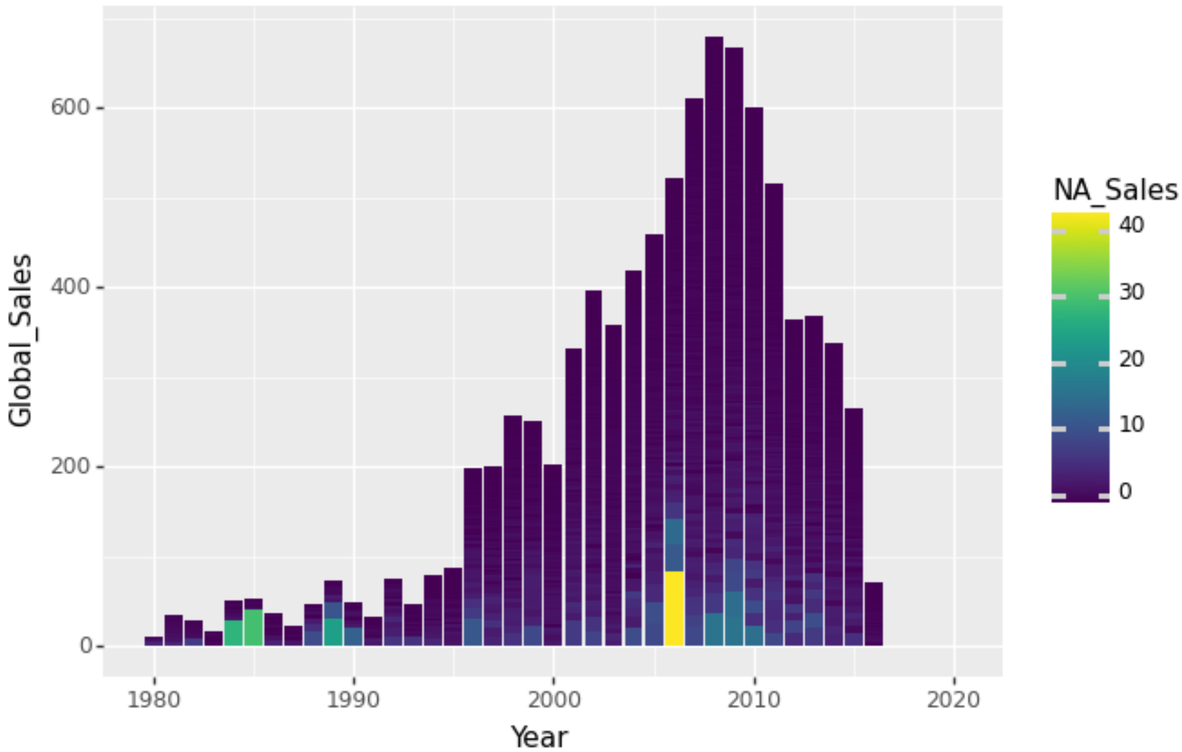
Out[0]: <ggplot: (309845377)>

In [0]:
```
(ggplot(vg, aes(x = 'Year', y = 'NA_Sales', fill = 'Year')) + stat_smooth(method='lm') + facet_wrap('Year')
+ geom_point() + ggtitle('Year vs NA Sales filled Year') + labs(x = 'Year', y = 'NA Sales'))
```



Out[0]: <ggplot: (312419165)>

In [0]:
```
(ggplot(vg, aes(x = "Year", y = "Global_Sales", fill = "NA_Sales")) +
geom_bar(stat = "identity"))
```



Out[0]: <ggplot: (8772984258003)>

4. Which company is the most successful in video game making?

- The variables we are using will be Publisher, NA Sales, EU Sales, JP Sales, Other Sales. We used linear regression and standardized. We used train test split as validation.

We are not using clustering model to answer this question.

- We chose this analysis plan because many different companies release games that span across different genres. For example, Nintendo makes games that can be played by anybody across any age group, with Animal Crossing and Super Mario being a few good examples.
- Three data visualizations:
  - Graph that looks at Global Sales over the years
  - Graph that shows NA sales over the years (since we live in NA)
  - Bar graph that shows Global Sales with NA Sales as filled

We got rid of Global Sales because it was taking into account for NA Sales, EU Sales, JP Sales, and Other Sales.

EA (Electronic Arts) is most successful in creating and producing video games. They have made over 1300 games, whereas their competitors have not come close. Therefore, we came to the conclusion that EA is the most successful video game producer in the industry. We decided not to go forward with using clustering models and decided to use linear regression instead. The mean squared error came out to be 20.41 with the r2 being 0.048. This is a very low score. However, although we saw EA as the top game producer, it didn't mean that their sales were too high above the rest.

In [0]:
```
vg["Publisher"].value_counts()
```

Out[0]:
```
Electronic Arts                 1351
Activision                       975
Namco Bandai Games               932
Ubisoft                          921
Konami Digital Entertainment     832
                                ...
Technos Japan Corporation          1
Imax                               1
DigiCube                           1
Rain Games                         1
PopTop Software                    1
Name: Publisher, Length: 578, dtype: int64
```

In [0]:
```
eaRows = [i == "Electronic Arts" for i in vg["Publisher"]]
```

In [0]:
```
df = vg.drop("Publisher", 1) #dropping publisher col
df['Publisher'] = eaRows #add eaRows true is Electronic Arts

df = df.set_index("Publisher")
df.drop(False, inplace= True)
df = df.reset_index()
df.head()
```

Out[0]:

	Publisher	Rank	Name	Platform	Year	Genre	NA_Sales	EU_Sales	JP_Sales	Ot
0	True	78	FIFA 16	PS4	2015.0	Sports	1.11	6.06	0.06	
1	True	83	FIFA Soccer 13	PS3	2012.0	Action	1.06	5.05	0.13	
2	True	84	The Sims 3	PC	2009.0	Simulation	0.98	6.42	0.00	
3	True	93	Star Wars Battlefront (2015)	PS4	2015.0	Shooter	2.93	3.29	0.22	
4	True	100	Battlefield 3	X360	2011.0	Shooter	4.46	2.13	0.06	

In [0]:
```
df.dropna(how='any', inplace=True)
```

In [0]:
```
X = df[["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales"]]

X_train, X_test, y_train, y_test = train_test_split(df[predictors], df
["Year"], test_size=0.2)
```

In [0]:
```
zScore = StandardScaler()
zScore.fit(X_train)
Xz_train = zScore.transform(X_train)
Xz_test = zScore.transform(X_test)
LR = LinearRegression()
LR.fit(Xz_train, y_train)
```

Out[0]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, norma
lize=False)
```

In [0]:
```
pred = LR.predict(Xz_train)
pred[1:10]
```

Out[0]:
```
array([2005.98925511, 2007.5364649 , 2006.4706799 , 2006.36239492,
       2006.07660858, 2006.72375447, 2005.03991977, 2006.62528538,
       2006.36067868])
```

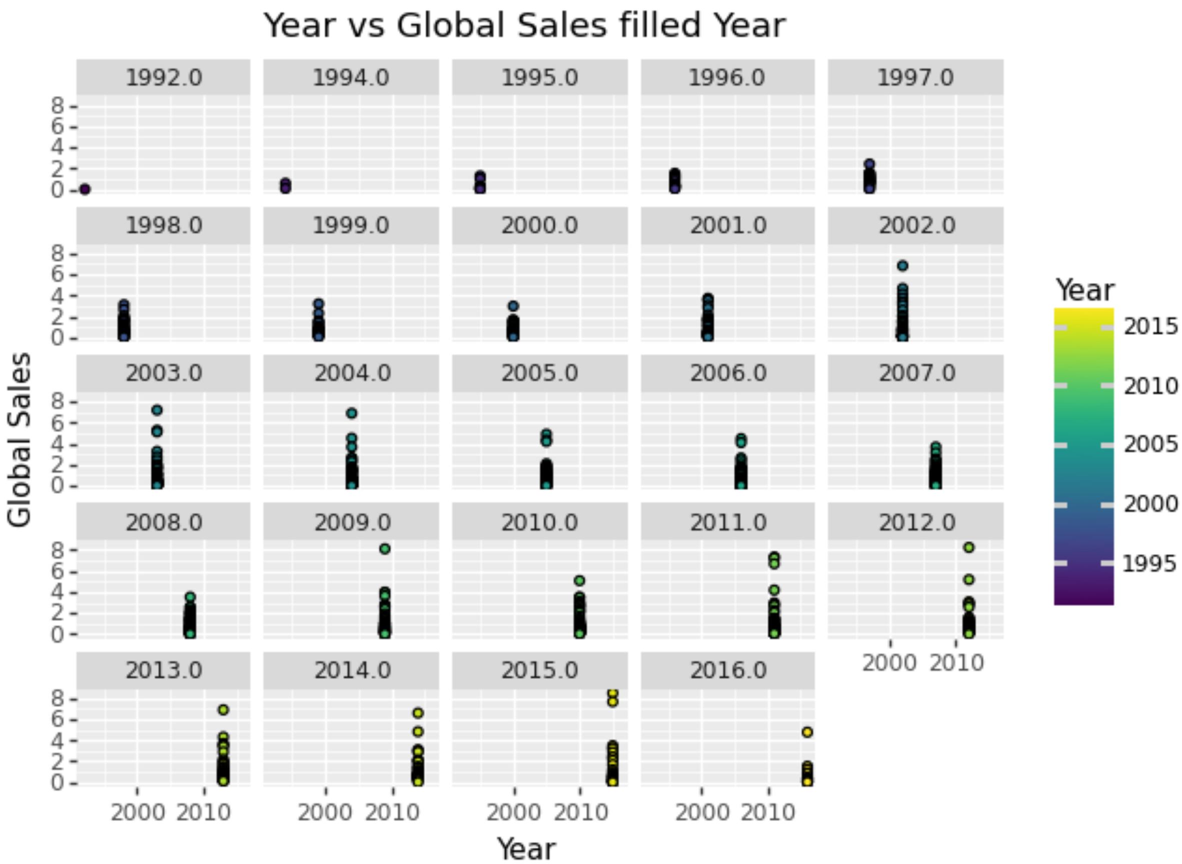
In [0]:
```
mean_squared_error(y_train, pred)
```
Out[0]: 20.40873

In [0]:
```
r2_score(y_train, pred)
```
Out[0]: 0.04776459

In [0]:
```
coefficients = pd.DataFrame({"Coef": LR.coef_,"Name": predictors})
coefficients = coefficients.append({"Coef": LR.intercept_, "Name" : "Intercept" }, ignore_index = True)
coefficients
```
Out[0]:

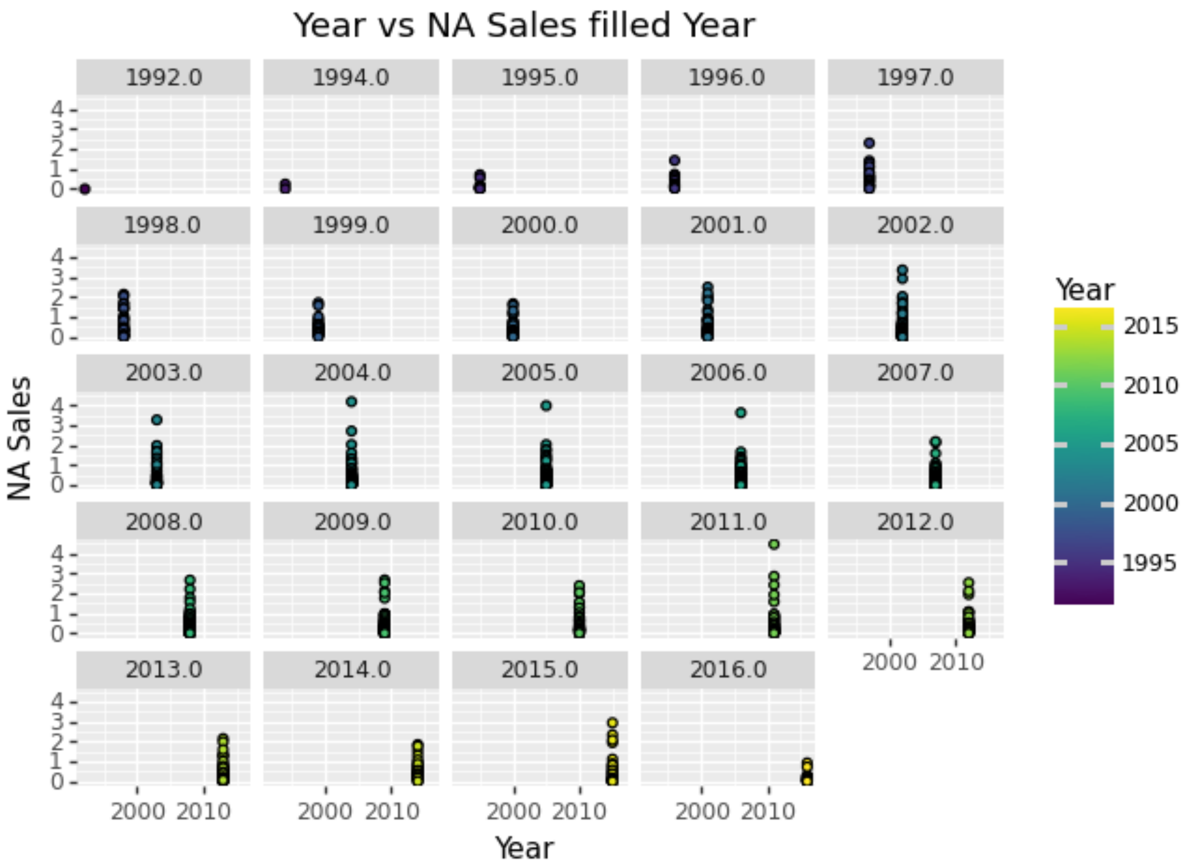
	Coef	Name
0	-0.634868	NA_Sales
1	0.154643	EU_Sales
2	-0.214533	JP_Sales
3	1.006006	Other_Sales
4	2006.427638	Intercept

In [0]:
```
(ggplot(df, aes(x = 'Year', y = 'Global_Sales', fill = 'Year')) + stat_smooth(method='lm') + facet_wrap('Year')
+ geom_point() + ggtitle('Year vs Global Sales filled Year') + labs(x = 'Year', y = 'Global Sales'))
```



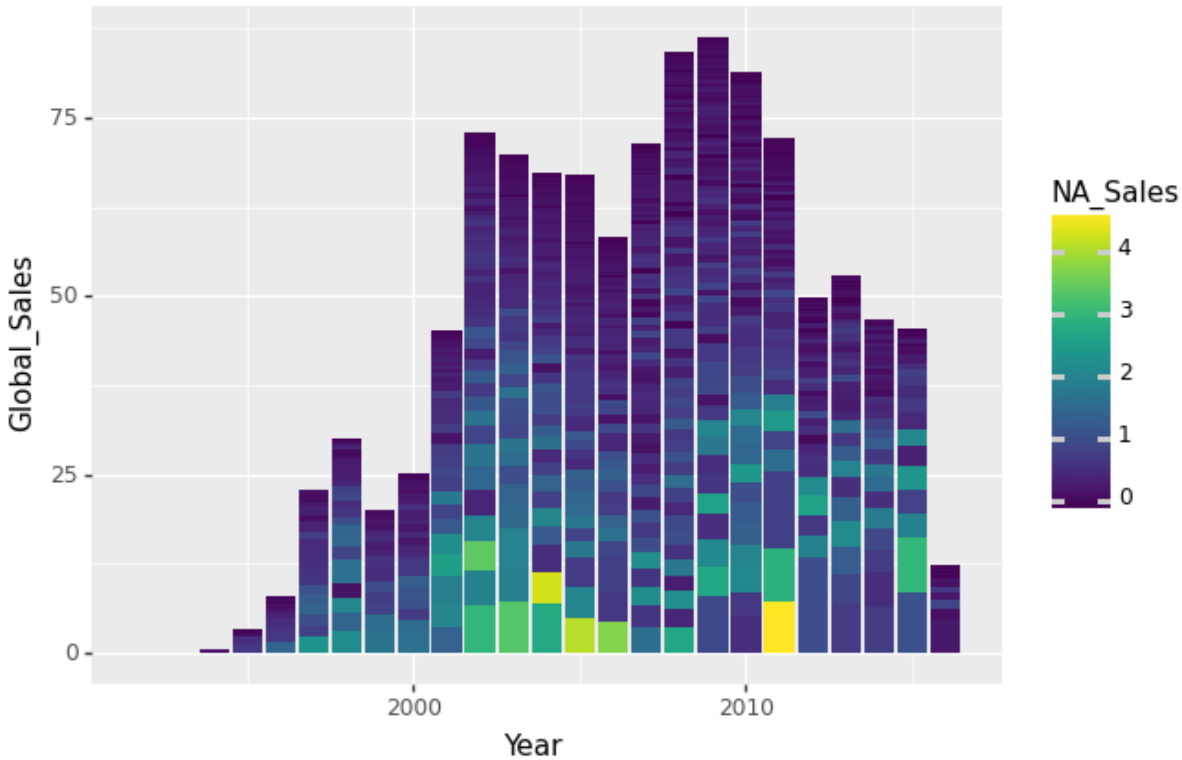
Out[0]: <ggplot: (-9223363306856101751)>

In [0]:
```
(ggplot(df, aes(x = 'Year', y = 'NA_Sales', fill = 'Year')) + stat_smooth(method='lm') + facet_wrap('Year')
+ geom_point() + ggtitle('Year vs NA Sales filled Year') + labs(x = 'Year', y = 'NA Sales'))
```



Out[0]: <ggplot: (-9223363306856643489)>

In [0]:
```
(ggplot(df, aes(x = "Year", y = "Global_Sales", fill = "NA_Sales")) +
geom_bar(stat = "identity"))
```



Out[0]: <ggplot: (8729998654314)>

5. Which country buys more video games?

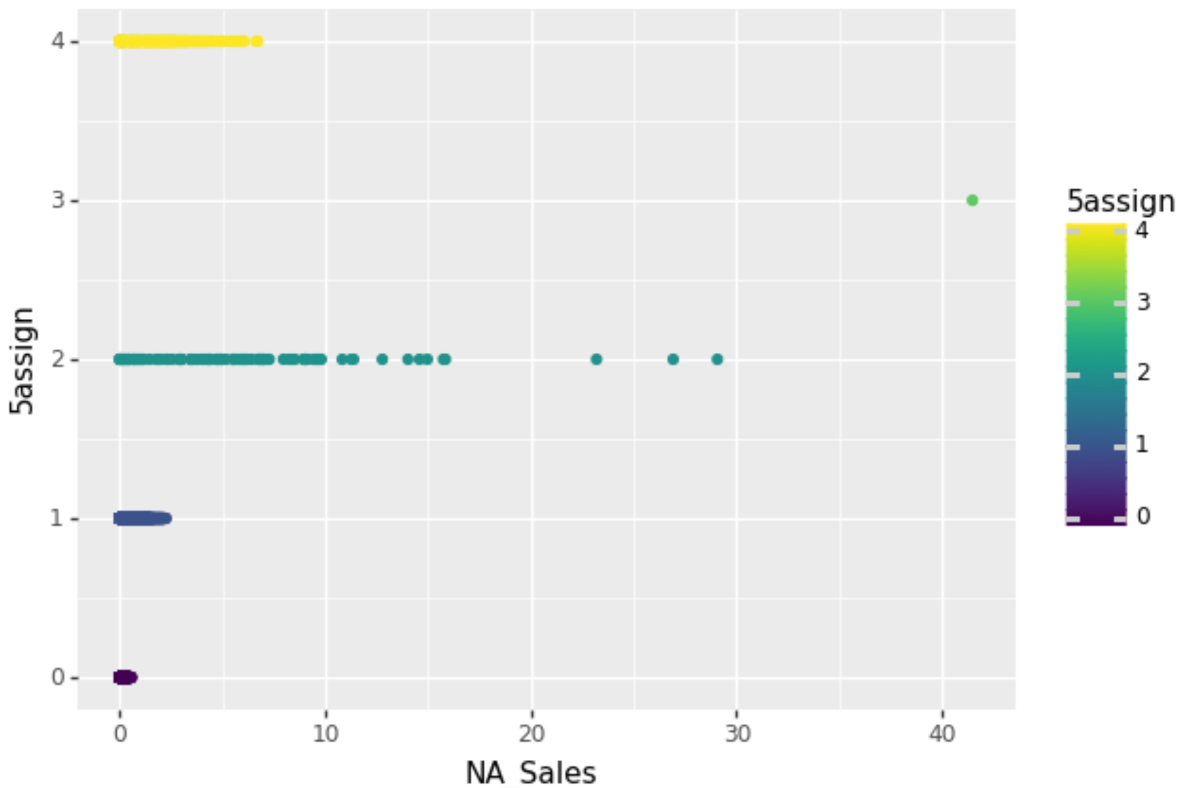
- The variables we are using will be NA Sales, EU Sales, JP Sales, Other Sales, Global Sales. We will use KFold model cluster.
- The 0 shows that there are unreleased games in that region
- We chose this analysis plan because the recent trend in online streaming and professional gaming has sparked an influx of new gamers. Some play for leisure whereas others try to make a living out of it. Korea and China have especially seen a large rate of growth in their video game usage. In Asian countries, gaming cafes are extremely popular places for people to hang out and play games. Therefore, we would like to see which countries have more sales in video games.
- Five data visualizations:
  - Graph cluster into a point for NA Sales
  - Graph cluster into a point for EU Sales
  - Bar Graph showing JP Sales
  - Bar Graph showing EU Sales
  - Bar Graph showing NA Sales

We used clustering model with KFold validation in order to determine this. North America ended up being the country with the highest amount of sales. The graph shows an abnormal amount of 0s, indicating a high amount of unreleased games in the region. North America, in particular, appears to have a greater amount of sales from the ranges of 1 to 5 compared to EU and JP. By using KFold model cluster, we have a 86.64% accuracy in our model. Taking a look at the clusters, the cluster with the lowest sales is the purple color. The turquoise color has a wide range of sales. The green cluster is the highest. The inbetween clusters are the yellow and blue.

In [0]:
```
feature = ["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales", "Global_
Sales"]
X = vg[feature]
Xdf = X

n_components = [2,3,4,5]
sil = []

for n in n_components:
    gmm = GaussianMixture(n_components = n)
    gmm.fit(X)

    clusters = gmm.predict(X)
    sil.append(silhouette_score(X, clusters))

    colName = str(n)+"assign"
    Xdf[colName] = clusters

print(sil)
```

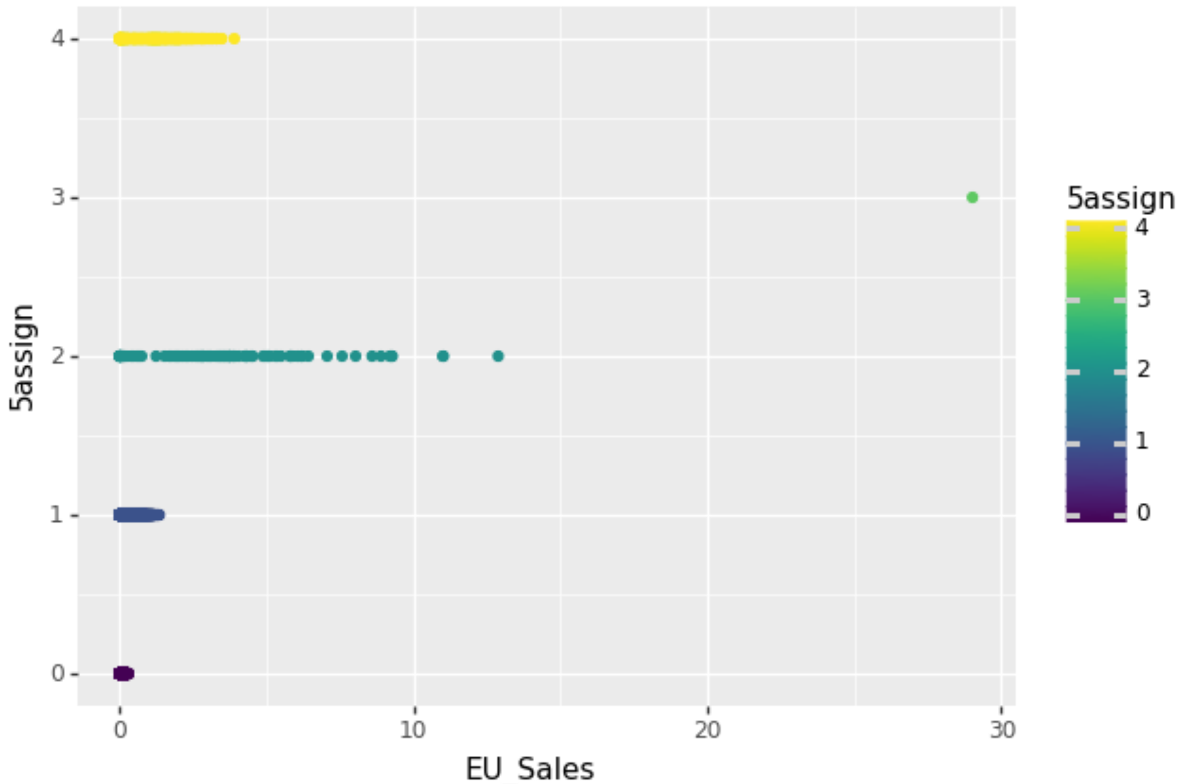
```
[0.5836392245578395, 0.7027288703319919, 0.8113007342720288, 0.86642
85468984777]
```

In [0]:
```
(ggplot(Xdf, aes(x = "NA_Sales", y = "5assign", color = "5assign")) +
geom_point())
```



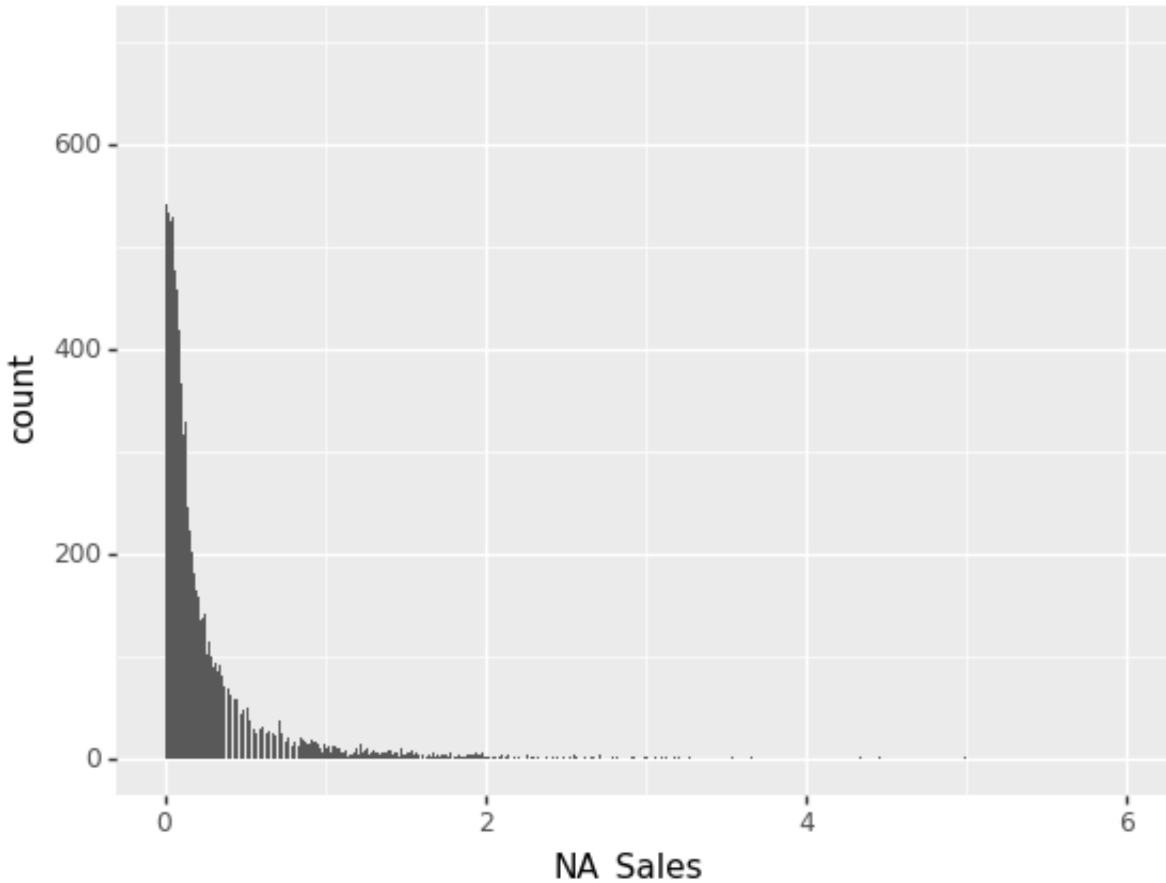
Out[0]: <ggplot: (322204201)>

In [0]:
```
(ggplot(Xdf, aes(x = "EU_Sales", y = "5assign", color = "5assign")) +
geom_point())
```



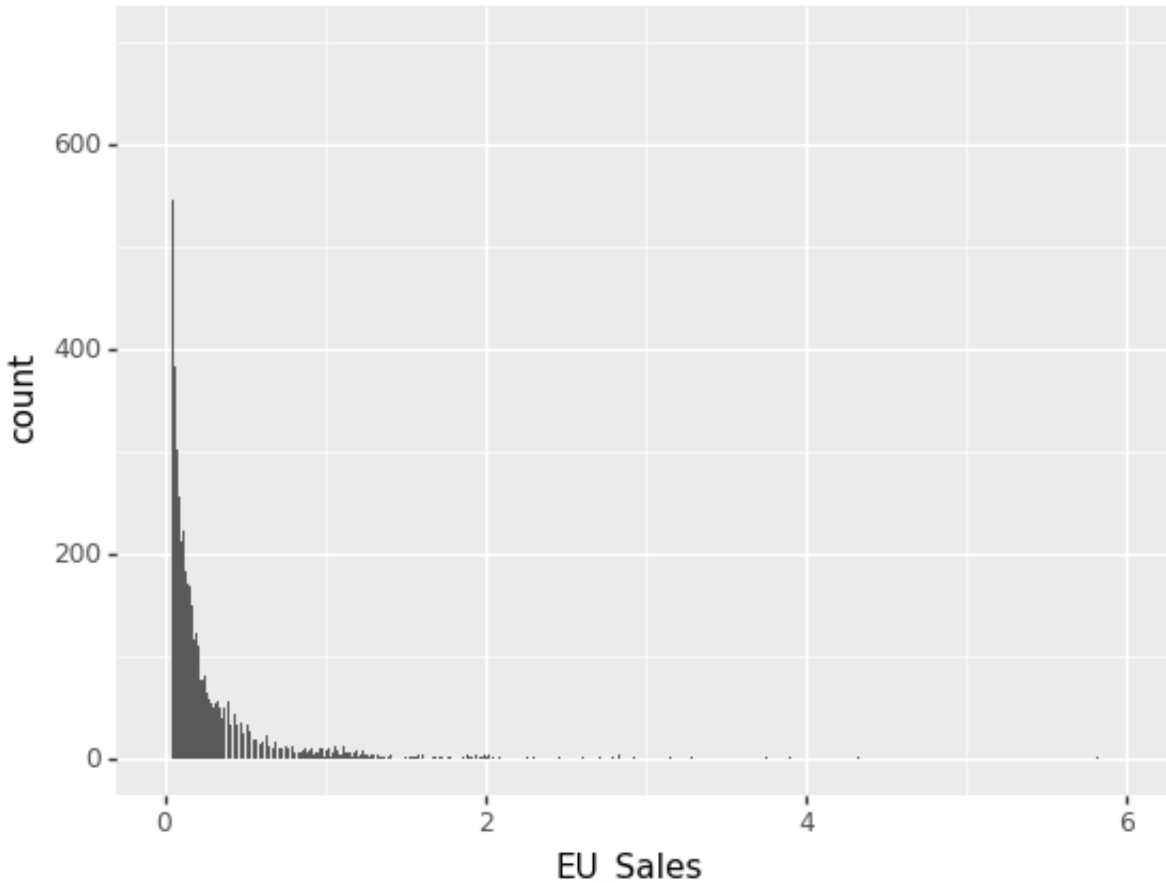
Out[0]: <ggplot: (324958029)>

In [0]:
```
(ggplot(Xdf, aes(x = 'NA_Sales')) + geom_bar() + xlim(0, 6) + ylim(0,
700))
```



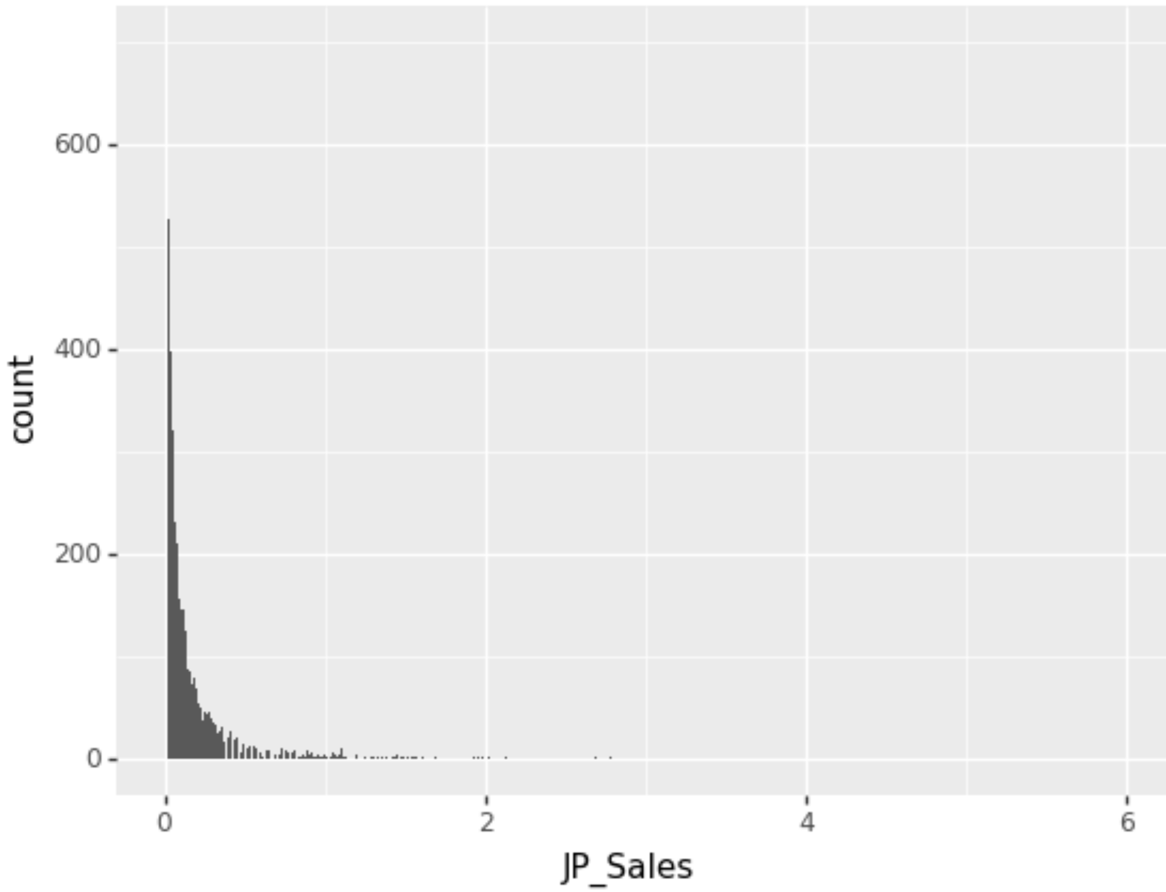
Out[0]: <ggplot: (321256469)>

In [0]:
```
(ggplot(Xdf, aes(x = "EU_Sales")) + geom_bar() + xlim(0, 6) + ylim(0,
700))
```



Out[0]: <ggplot: (319776921)>

In [0]:
```
(ggplot(Xdf, aes(x = "JP_Sales")) + geom_bar() + xlim(0, 6) + ylim(0,
700))
```



Out[0]: <ggplot: (321699045)>

6. Does Japan buy less video games compared to America?

- Our original question was asking which Asian countries, buys more video games. We modified the question because Other Sales could refer to ther non-Asian countries making our analysis inaccurate.
- The variables we are using will be JP Sales, NA Sales, EU Sales, and Other Sales. We will use Dimensionality reduction validation through PCA.
- We chose this analysis plan because different countries have different cultures. More Americans play video games for leisure whereas the Chinese environment is more competitive. There are more gaming accessories and equipment in Asia, so we would like to compare which countries buy/play more video games. Article about chinese being competitive video gamers: [https://venturebeat.com/2019/01/07/how-chinese-and-american-gamers-differ/ (https://venturebeat.com/2019/01/07/how-chinese-and-american-gamers-differ/)](https://venturebeat.com/2019/01/07/how-chinese-and-american-gamers-differ/)
- Four data visualizations:
  - Graph that shows Global Sales with JP sales filled
  - Graph that shows Global Sales with NA Sales
  - PCA of pc
  - PCA of pc with 0.95 intercept

Originally the question asked which Asian countries bought more video games. We then modified the question due to the data only providing Japan's sales. The information found in the other sales category did not hold much value as the other sales could be coming from non-Asian countries.

By using dimensionaltiy reduction validation through PCA, we were able to see which variables go together. The screen plot of the first point tells us that we know about 65% of our variables. The loading shows correlation to our sales variables. In addition, the linear regression models for JP Sales has an accuracy score of 0.98 and NA Sales has an accuracy score of 0.81. The bar graph of NA sales is from 0 to 40 whereas JP sales range is from 0 to 10. Looking at the graph, we can conclude that Japan buys less video games compared to America.

In [0]:
```
features = ["NA_Sales", "JP_Sales", "Other_Sales"]

z = StandardScaler()

vg = vg[features]

vg[features] = z.fit_transform(vg[features])
```

In [101]:
```
pca2 = PCA()
pca2.fit(vg)
print(pca2.explained_variance_ratio_)
```

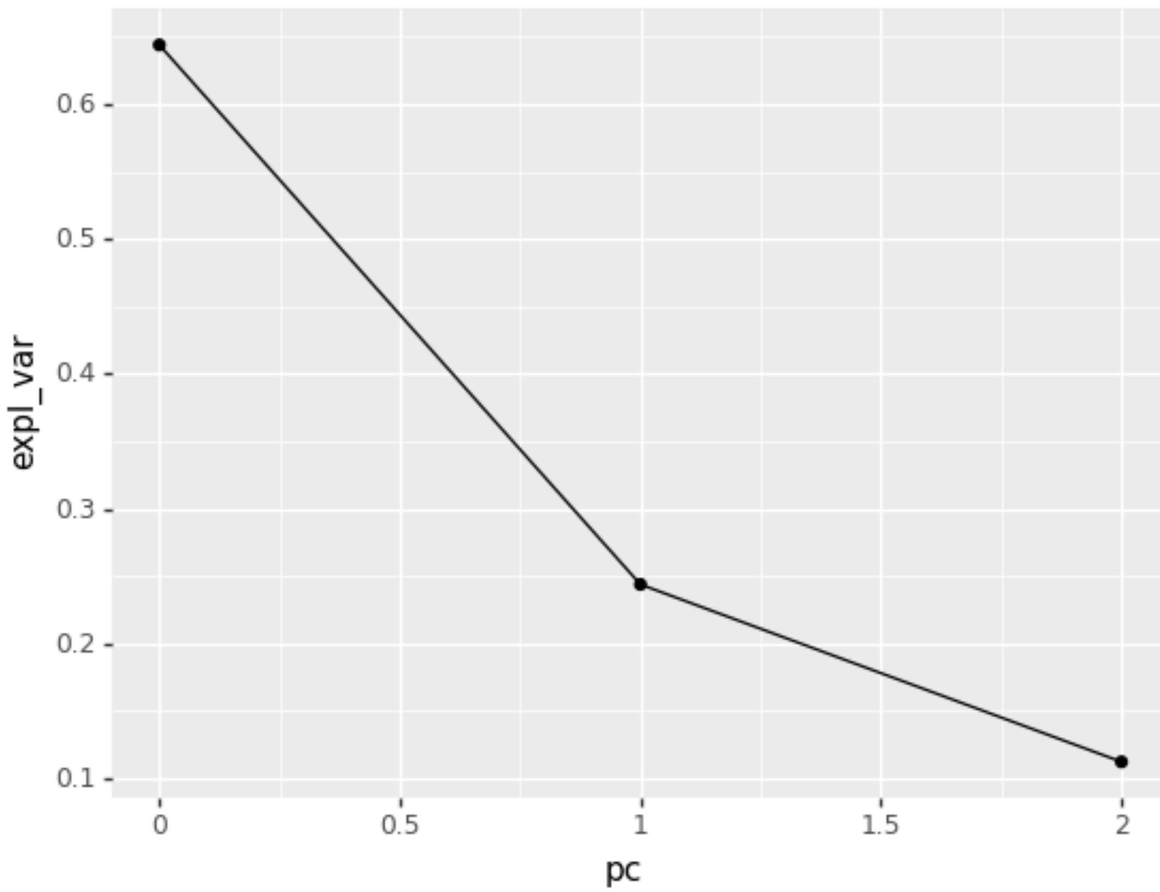
```
[0.64376763 0.24387195 0.11236043]
```

In [102]:
```
pcaDF2 = pd.DataFrame({"expl_var" : pca2.explained_variance_ratio_, "p
c": range(0,3), "cum_var": pca2.explained_variance_ratio_.cumsum()})
pcaDF2.head()
```

Out[102]:

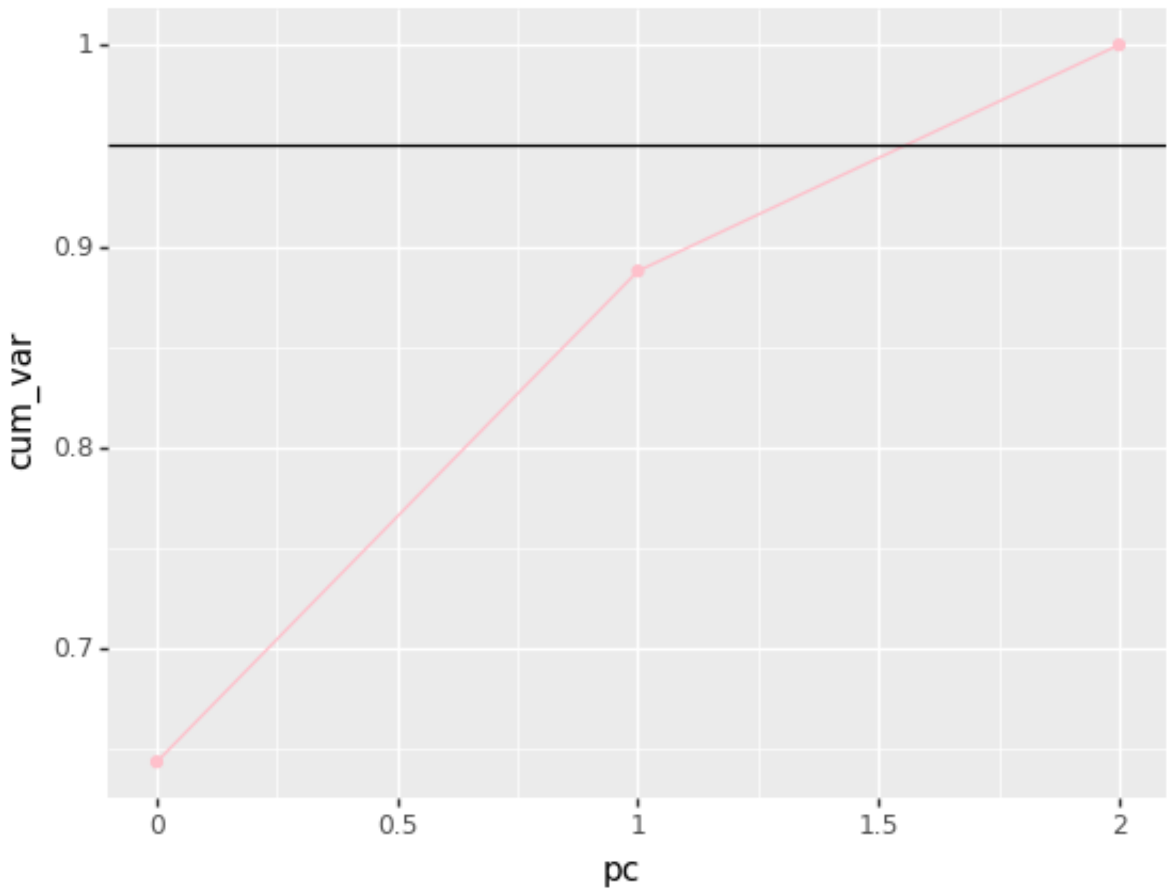
	expl_var	pc	cum_var
0	0.643768	0	0.643768
1	0.243872	1	0.887640
2	0.112360	2	1.000000

In [103]:
```
ggplot(pcaDF2, aes(x = "pc", y = "expl_var")) + geom_line() + geom_poi
nt()
```



Out[103]: <ggplot: (8730006257661)>

In [104]:
```
(ggplot(pcaDF2, aes(x = "pc", y = "cum_var")) + geom_line(color = "pin
k") +
 geom_point(color = "pink") + geom_hline(yintercept = 0.95))
```



Out[104]: <ggplot: (8730006421852)>

In [0]:
```
pcomps2 = pca2.transform(vg[features])
pcomps2 = pd.DataFrame(pcomps2[:,0:2])


pcomps2pink = pca2.transform(vg[features])
pcomps2pink = pd.DataFrame(pcomps2pink[:, 0:2])
```

In [108]:
```
#modeMod1 for pink
lr2 = LinearRegression()
lr2.fit(pcomps2pink, vg["JP_Sales"])
print("2 PCs for JP Sales:   ", lr2.score(pcomps2pink, vg["JP_Sales"])
)

#modeMod1
lr3 = LinearRegression()
lr3.fit(pcomps2, vg["JP_Sales"])
print("2 PCs for JP Sales:    ", lr3.score(pcomps2, vg["JP_Sales"]))

#modeMod1 for pink
lr4 = LinearRegression()
lr4.fit(pcomps2pink, vg["NA_Sales"])
print("2 PCs for NA Sales:   ", lr4.score(pcomps2pink, vg["NA_Sales"])
)

#modeMod1
lr4 = LinearRegression()
lr4.fit(pcomps2, vg["NA_Sales"])
print("2 PCs for NA Sales:    ", lr4.score(pcomps2, vg["NA_Sales"]))
```

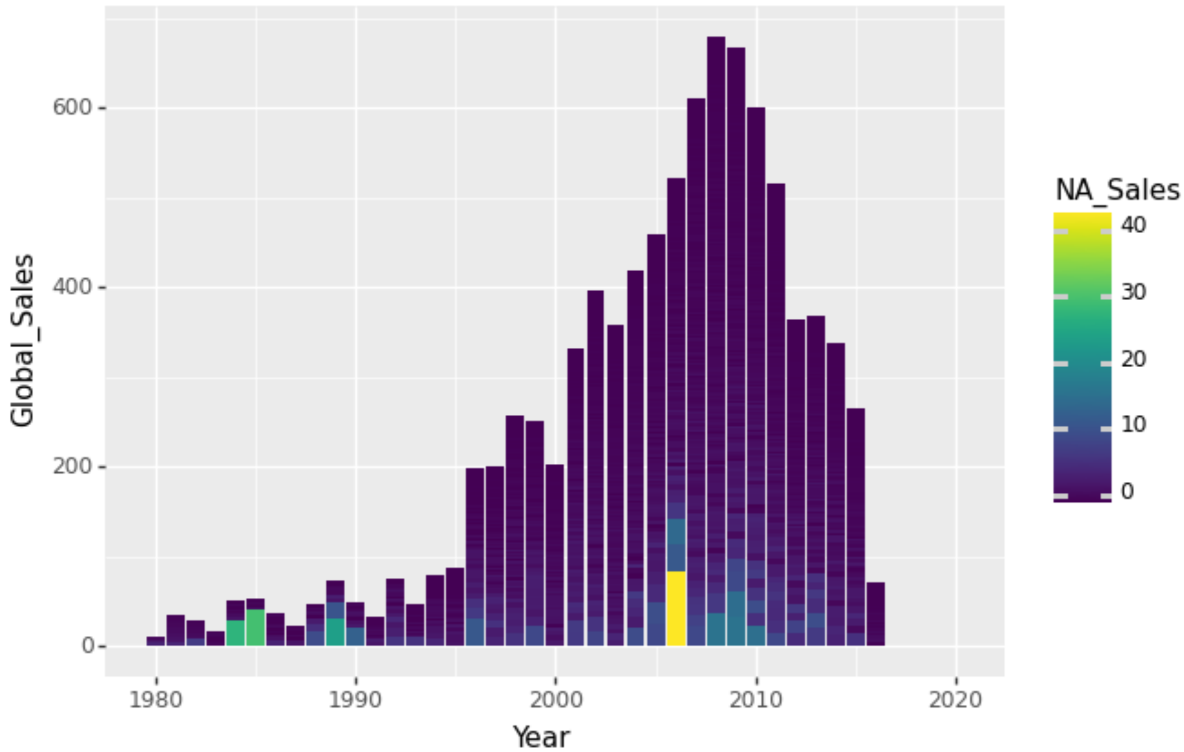
```
2 PCs for JP Sales:    0.980401457507077
2 PCs for JP Sales:     0.980401457507077
2 PCs for NA Sales:    0.8096123123592586
2 PCs for NA Sales:     0.8096123123592586
```

In [0]:
```
loadings = pd.DataFrame({"loading": pca2.components_.flatten(),
            "comp": np.repeat(range(0,3), 3 ,
        axis=0), "variable":np.tile(features,3) })

loadings.head(3)
```

Out[0]:

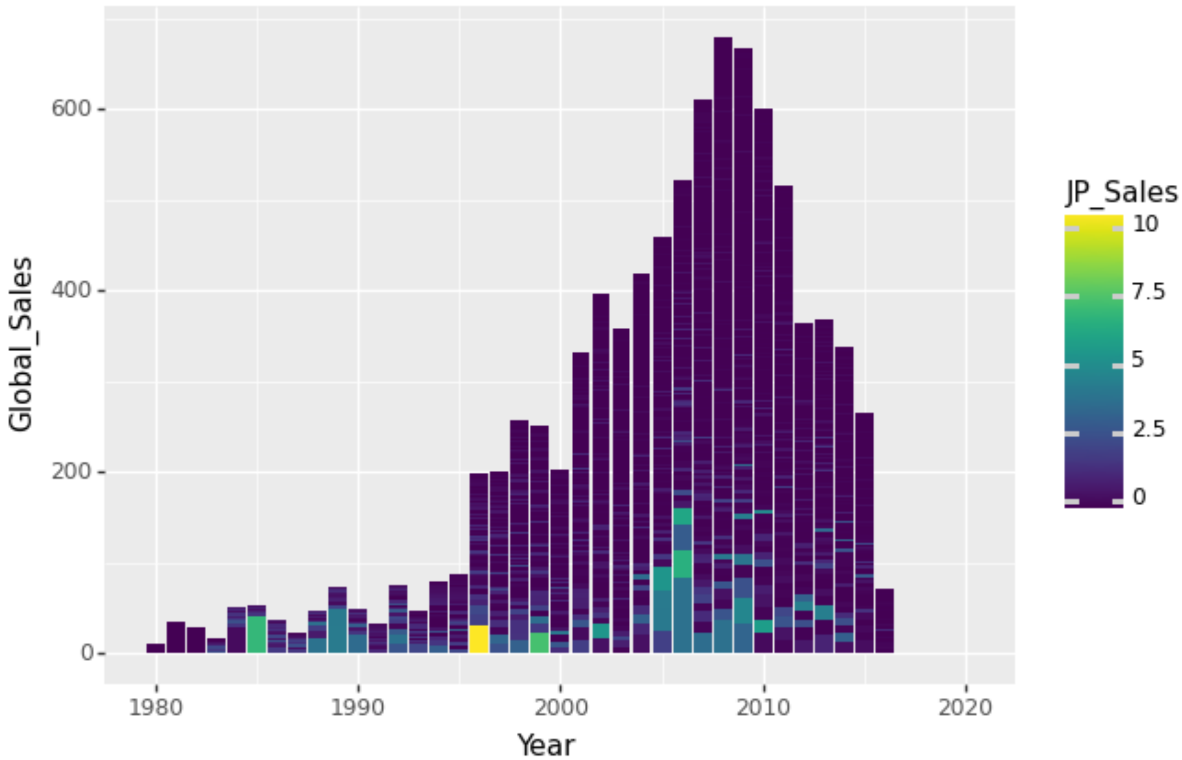
	loading	comp	variable
0	0.639890	0	NA_Sales
1	0.492780	0	JP_Sales
2	0.589668	0	Other_Sales

In [0]:
```
(ggplot(vg, aes(x = "Year", y = "Global_Sales", fill = "NA_Sales")) +
geom_bar(stat = "identity"))
```



Out[0]: <ggplot: (8772985918314)>

In [0]:
```
(ggplot(vg, aes(x = "Year", y = "Global_Sales", fill = "JP_Sales")) +
geom_bar(stat = "identity"))
```



Out[0]: <ggplot: (8772985986687)>