Packages

```
import warnings
warnings.filterwarnings('ignore')
```

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from plotnine import *
```

```
from sklearn.tree import DecisionTreeClassifier # Decision Tree
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB, CategoricalNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge, Lasso

from sklearn import metrics
from sklearn.preprocessing import StandardScaler #Z-score variables

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 validation metrics
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import plot_confusion_matrix

from sklearn.model_selection import GridSearchCV

from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression

from sklearn.cluster import AgglomerativeClustering

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

from sklearn.metrics import silhouette_score
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

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

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

Import Data

```
data = pd.read_csv("https://raw.githubusercontent.com/phamvlai/CPSC392_Project/master/
data.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27

Transform Data

```
data.isnull().sum()
```

```
Rank              0
Name              0
Platform          0
Year            271
Genre             0
Publisher        58
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
dtype: int64
```

```
data_clean = data.dropna()
```

```
data_clean.isnull().sum()
```

```
Rank           0
Name           0
Platform       0
Year           0
Genre          0
Publisher      0
```

GOAT

```
data_group = data_clean.groupby('Publisher')['Global_Sales'].sum().reset_index()
```

```
data_group.head()
```

	Publisher	Global_Sales
0	10TACLE Studios	0.11
1	1C Company	0.10
2	20th Century Fox Video Games	1.94
3	2D Boy	0.04
4	3DO	10.12

```
top_pub = data_group.nlargest(20, ['Global_Sales'])
```

```
pub = top_pub['Publisher']
```

```
data_top = data_clean[data_clean['Publisher'].isin(pub)]
```

```
data_top.shape
```

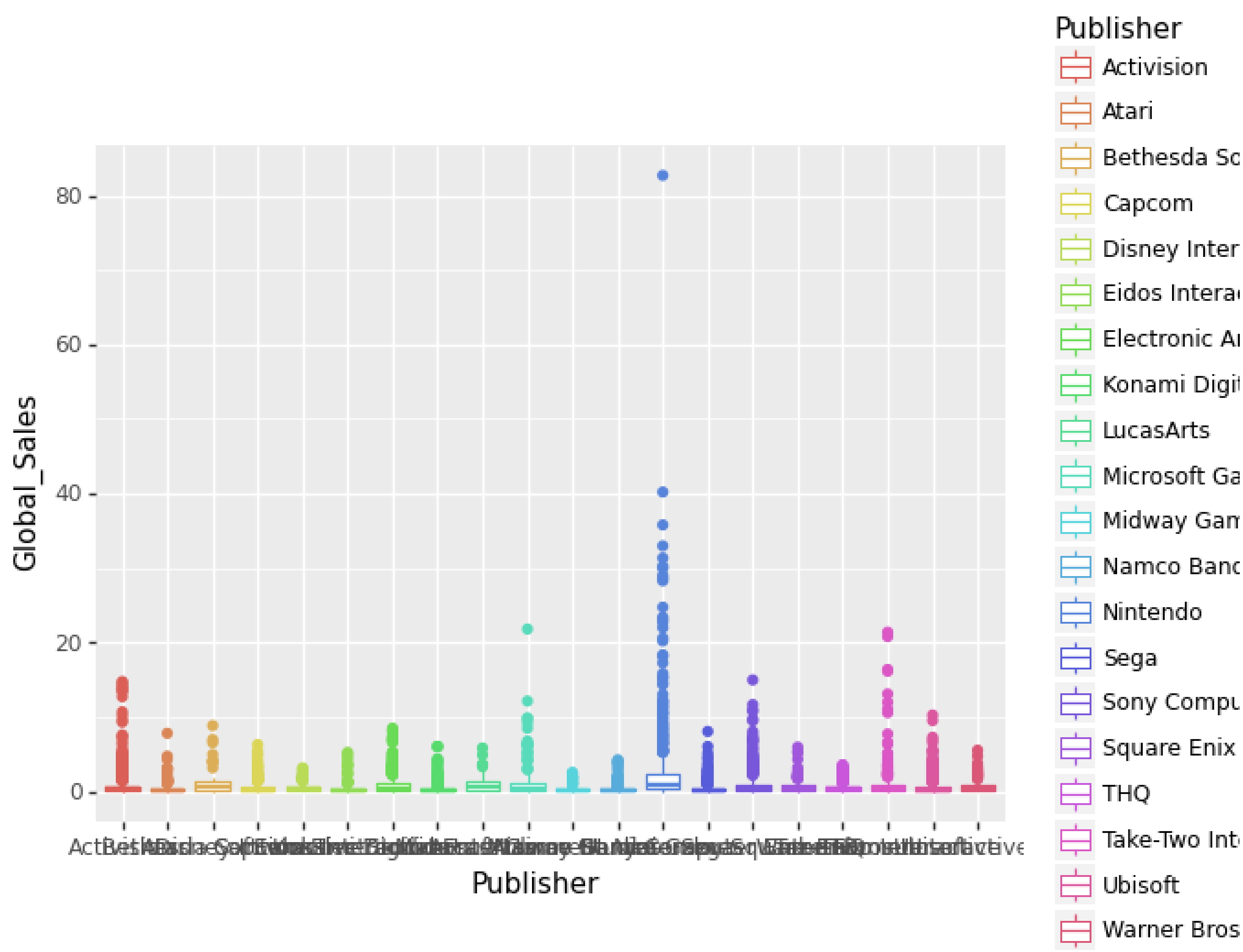
```
(10232, 11)
```

```
dups_pub = data_top.pivot_table(index=['Publisher'], aggfunc='size')
```

```
print(dups_pub)
```

```
Publisher
Activision                          966
Atari                               347
Bethesda Softworks                   69
Capcom                              376
Disney Interactive Studios          214
Eidos Interactive                   196
Electronic Arts                    1339
Konami Digital Entertainment        823
LucasArts                            89
```

```
(ggplot(data_top, aes("Publisher", "Global_Sales", color = "Publisher")) + geom_boxplo
```




```
<ggplot: (8753045160943)>
```

```
dummies = pd.get_dummies(data_top.Platform, drop_first = True)
data_dum = pd.concat([data_top,dummies], axis=1)
```

```
dummies2 = pd.get_dummies(data_top.Genre, drop_first = True)
data_dum = pd.concat([data_dum,dummies2], axis=1)
```

```
dummies3 = pd.get_dummies(data_top.Publisher, drop_first = True)
data_dum = pd.concat([data_dum,dummies3], axis=1)

data_final = data_dum.drop(['Rank','Name','Platform','Genre','Publisher','Other_Sales'

X = data_final.loc[:, data_final.columns != 'Global_Sales']
Y = data_final["Global_Sales"]

LR_Model = LinearRegression()

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
X_train.head()

zscore = StandardScaler()
zscore.fit(X_train)
Xz_train = zscore.transform(X_train)
Xz_test = zscore.transform(X_test)

LR_Model.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_pred = LR_Model.predict(X_test)
y_pred[1:10]
```

```
array([0.30149943, 0.11955938, 0.5660095 , 0.10170688, 0.46235791,
       0.37299965, 0.69213629, 0.52242723, 0.34644432])
```

```
true_vs_pred = pd.DataFrame({"predict": y_pred,"trueV": y_test})
true_vs_pred.head()
```

	predict	trueV
15903	-0.005508	0.02
10651	0.301499	0.10
10378	0.119559	0.11
10904	0.566009	0.09
13338	0.101707	0.05

```
mean_squared_error(y_test,y_pred)
```

```
0.4312542
```

```
r2_score(y_test,y_pred)
```

```
0.8735007
```

```
coefficients = pd.DataFrame({"Coef":LR_Model.coef_,
               "Name": X})
coefficients = coefficients.append({"Coef": LR_Model.intercept_,
                     "Name": "intercept"}, ignore_index = True)


coefficients
```

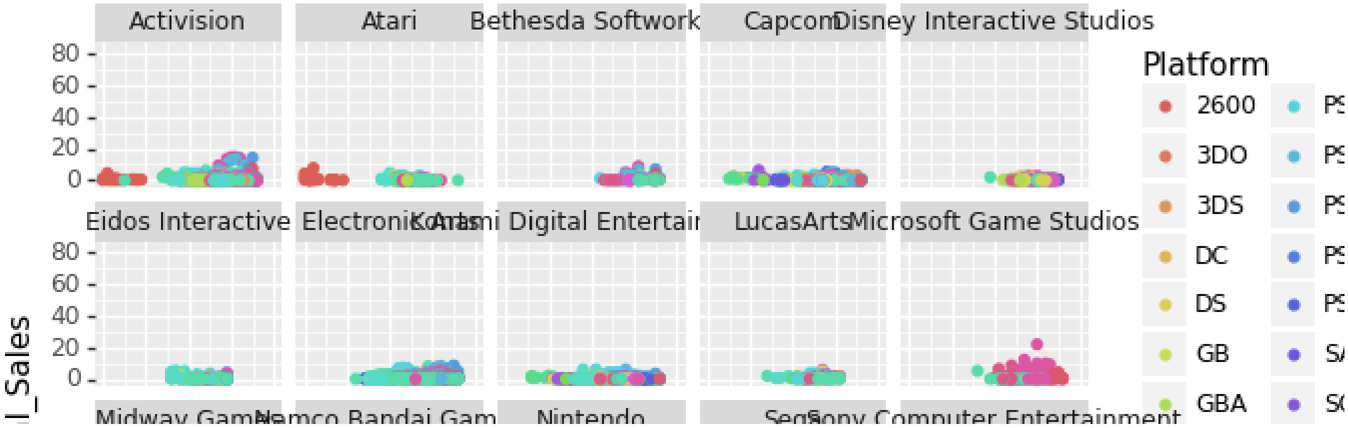
4	0.191887	(3, D, S)
5	0.204593	(D, C)
6	0.188928	(D, S)
7	-0.176990	(G, B)
8	0.190331	(G, B, A)
9	0.143645	(G, C)
10	0.139605	(G, E, N)
11	0.343885	(G, G)
12	0.058444	(N, 6, 4)
13	-0.068882	(N, E, S)
14	0.348028	(P, C)
15	0.296240	(P, S)
16	0.297075	(P, S, 2)
17	0.366757	(P, S, 3)
18	0.621911	(P, S, 4)
19	0.246308	(P, S, P)
20	0.211850	(P, S, V)
21	0.291576	(S, A, T)
22	0.257443	(S, C, D)
23	0.089700	(S, N, E, S)
24	0.330962	(W, S)
25	0.205820	(W, i, i)
26	0.192703	(W, i, i, U)
27	0.138603	(X, 3, 6, 0)
28	0.164216	(X, B)
29	0.140595	(X, O, n, e)
30	-0.009034	(A, d, v, e, n, t, u, r, e)
31	-0.037859	(F, i, g, h, t, i, n, g)
32	-0.006273	(M, i, s, c)
33	-0.035550	(P, u, z, z, l, e)
34	0.055128	(R, a, c, i, n, g)

```
best_var = coefficients.nlargest(20, ['Coef'])
```

```
best_var
```

	Coef	Name
1	1.689303	(N, A, _, S, a, l, e, s)
2	1.222156	(J, P, _, S, a, l, e, s)
18	0.621911	(P, S, 4)
17	0.366757	(P, S, 3)
14	0.348028	(P, C)
11	0.343885	(G, G)
24	0.330962	(W, S)
3	0.305403	(3, D, O)
16	0.297075	(P, S, 2)
15	0.296240	(P, S)
21	0.291576	(S, A, T)
22	0.257443	(S, C, D)
19	0.246308	(P, S, P)
20	0.211850	(P, S, V)
25	0.205820	(W, i, i)
5	0.204593	(D, C)
26	0.192703	(W, i, i, U)
4	0.191887	(3, D, S)
8	0.190331	(G, B, A)

```
(ggplot(data_top, aes("Year", "Global_Sales", color = "Platform")) + geom_point()) + i
```



The answer to my question of GOAT Publisher is Besthesda Softwork. Not only does my coefficients p
that Besthesda has relatively less games produced than the ones with higher global sales. As they pro
sales for each, as compared to Nintendo who produced more games overall, with few having extreme

Genres & Platform Connection

```
data_cluster = data_dum.drop(['Rank','Name','Platform','Year','Genre','Publisher','NA_
```

```
data_cluster.shape
```

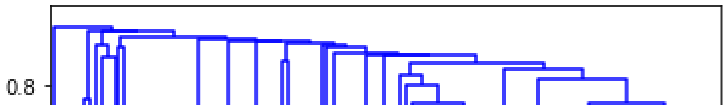
```
(10232, 57)
```

```
hac = AgglomerativeClustering(n_clusters = 4,
                              affinity = "cosine",
                             linkage = "average")
```

```
hac.fit(data_cluster)
```

```
AgglomerativeClustering(affinity='cosine', compute_full_tree='auto',
                        connectivity=None, distance_threshold=None,
                        linkage='average', memory=None, n_clusters=4)
```

```
dendro = sch.dendrogram(sch.linkage(data_cluster, metric = "cosine", method='average')
```




```
membership = hac.labels_
```

```
data_dum.shape
```

```
(10232, 68)
```

```
membership.shape
```

```
(10232,)
```

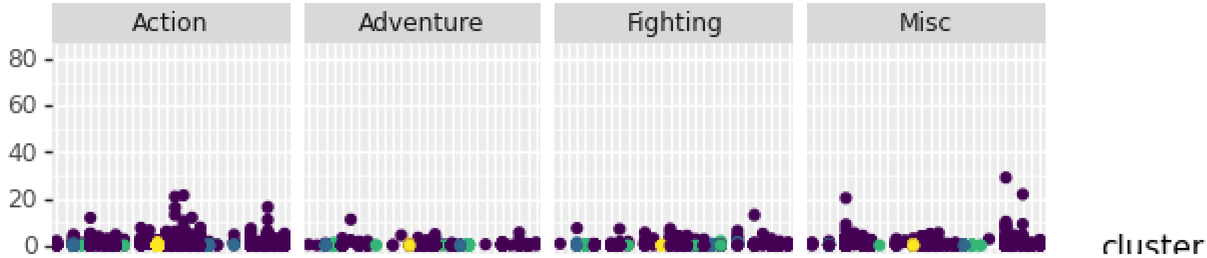
```
data_dum["cluster"] = membership
```

```
silhouette_score(X,membership)
```

```
-0.1390954
```

```
cols = []
count = 1
for column in data_dum.columns:
    if column == 'Platform':
        cols.append(f'Platform_{count}')
        count+=1
        continue
    cols.append(column)
data_dum.columns = cols
```

```
(ggplot(data_dum) + aes('Platform_1','Global_Sales',color = 'cluster')) + geom_point(a
```

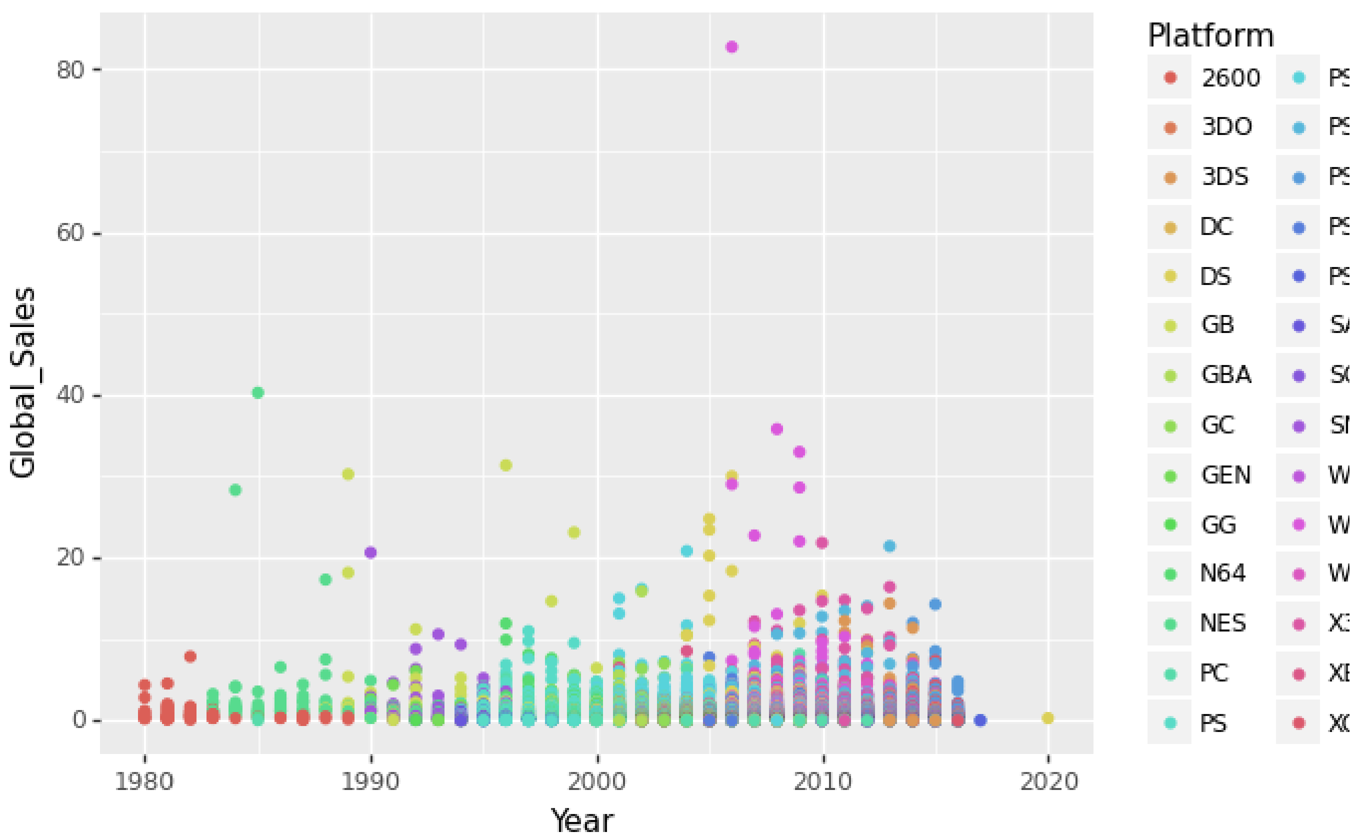


There is no connection between genres and platform that show results in global sales. While the dend
silhouette score indicates the objects were not well matched. In addition, the gg plot shows no distinc
a factor in global sales.

Platform Impact on Global Sales

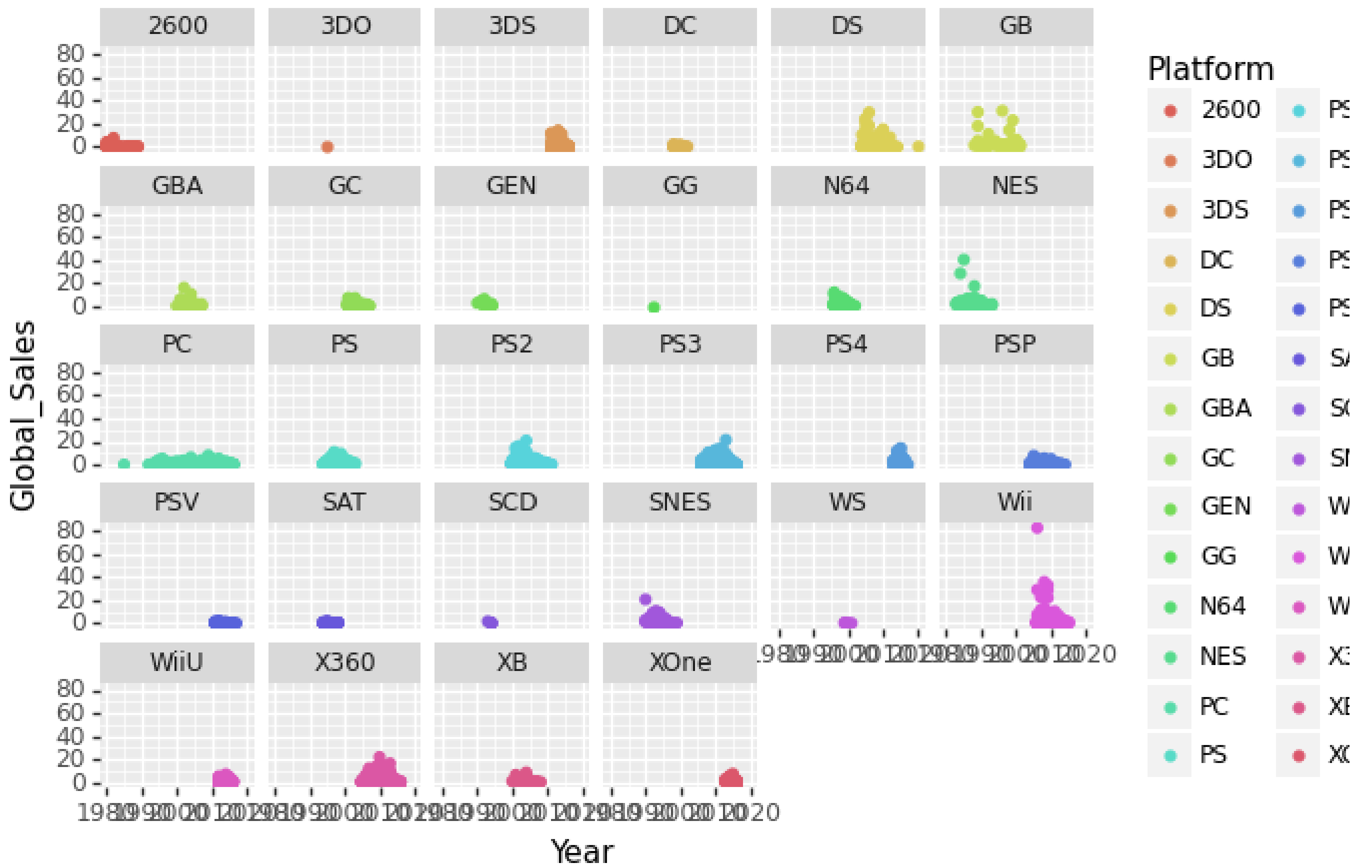
```
dummies_G = pd.get_dummies(data_top.Platform, drop_first = True)
data_Genre = pd.concat([data_top,dummies_G], axis=1)
```

```
(ggplot(data_top) + aes('Year','Global_Sales')) + geom_point(aes(color = "Platform"))
```




```
<ggplot: (-9223363283801938018)>
```

```
(ggplot(data_top) + aes('Year','Global_Sales')) + geom_point(aes(color = "Platform"))
```




```
<ggplot: (-9223363283810931803)>
```

```
data_Genre = data_Genre.drop(['Rank','Name','Platform','Year','Genre','Publisher','NA_

X = data_Genre.loc[:, data_Genre.columns != 'Global_Sales']
y = data_Genre["Global_Sales"]

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)



rr = Ridge()

rr.fit(X_train,y_train)

print("TRAIN: ", mean_absolute_error(y_train, rr.predict(X_train)))
print("TEST : ", mean_absolute_error(y_test, rr.predict(X_test)))
```

```
TRAIN:  0.7222204590520332
TEST :  0.7684381597653901
```

```
coefficients = pd.DataFrame({"Coef":rr.coef_,
                "Name":X})
coefficients = coefficients.append({"Coef": rr.intercept_, "Name": "intercept"}, ignoı

best_plat = coefficients.nlargest(10, ['Coef'])


best_plat
```

	Coef	Name
4	2.352606	(G, B)
10	1.580078	(N, E, S)
27	0.883200	intercept
20	0.469752	(S, N, E, S)
15	0.408161	(P, S, 4)
24	0.138095	(X, 3, 6, 0)
7	0.095311	(G, E, N)
9	0.070769	(N, 6, 4)
22	0.039427	(W, i, i)
14	[illegible]	[illegible]

The answer is that NES is the platform that makes the most impact on global sales. This was determi as well as a ridge regression. Additionally, the gg plot shows that while the Wii had the higher global s for a longer period, similar to GB, but better global sales.

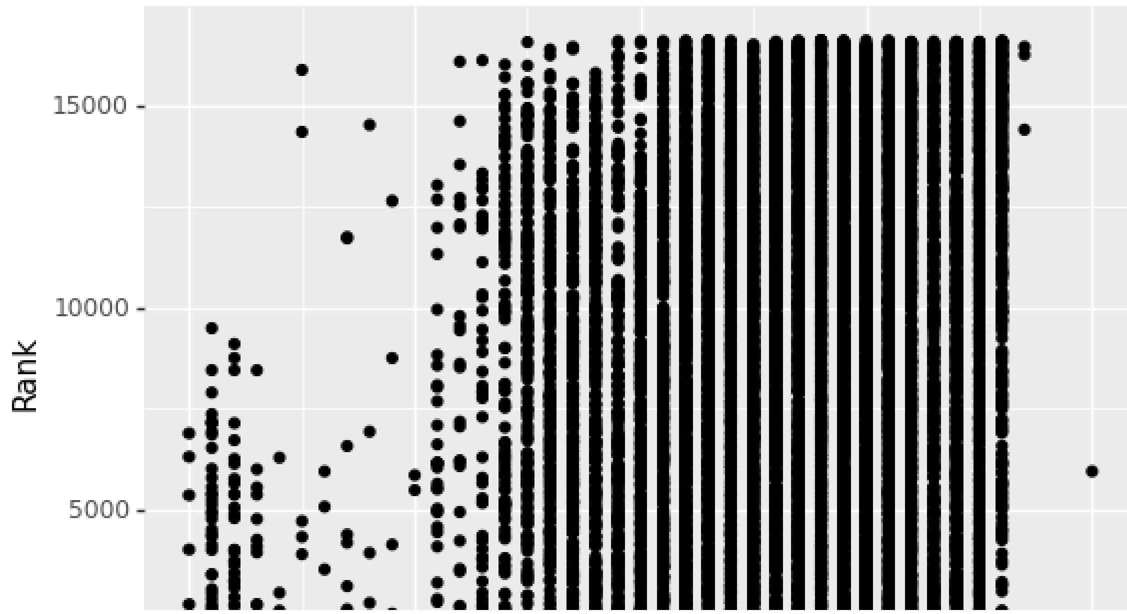
Riya Sagar

Is there a relationship between the year a game was released and the rank?

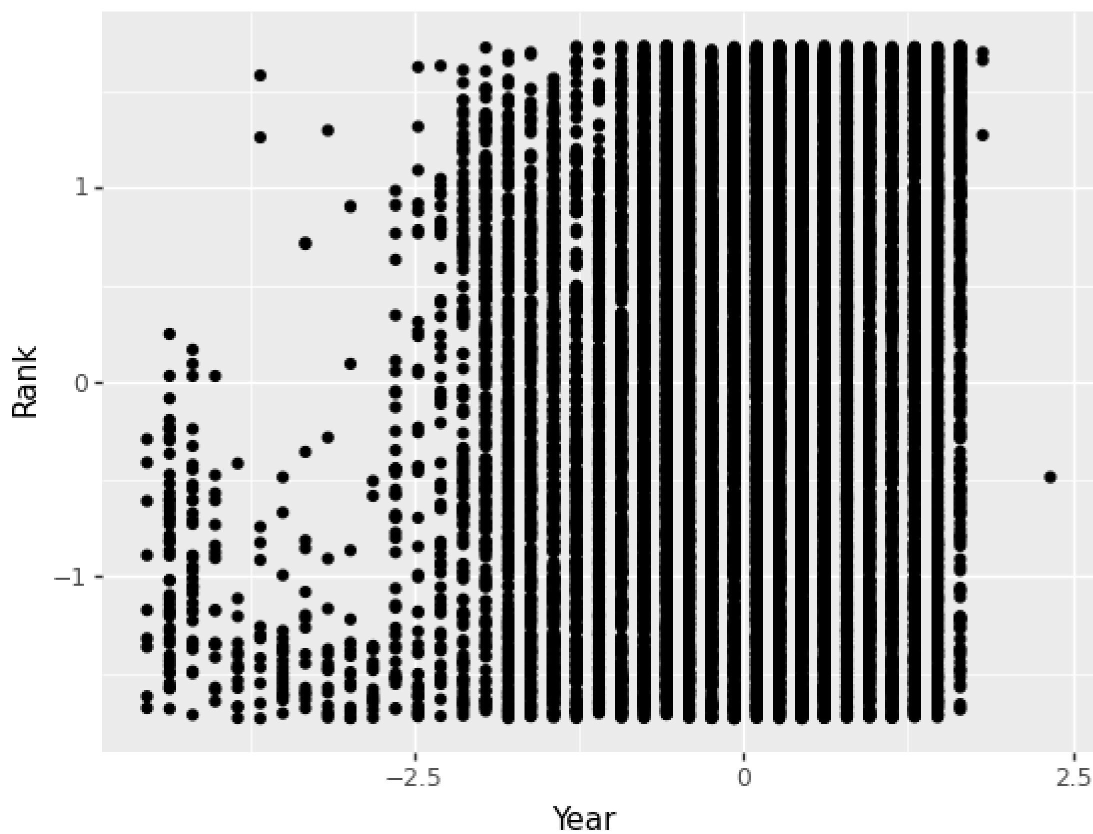
```
data.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27

```
(ggplot(data_clean, aes("Year", "Rank")) + geom_point())
```



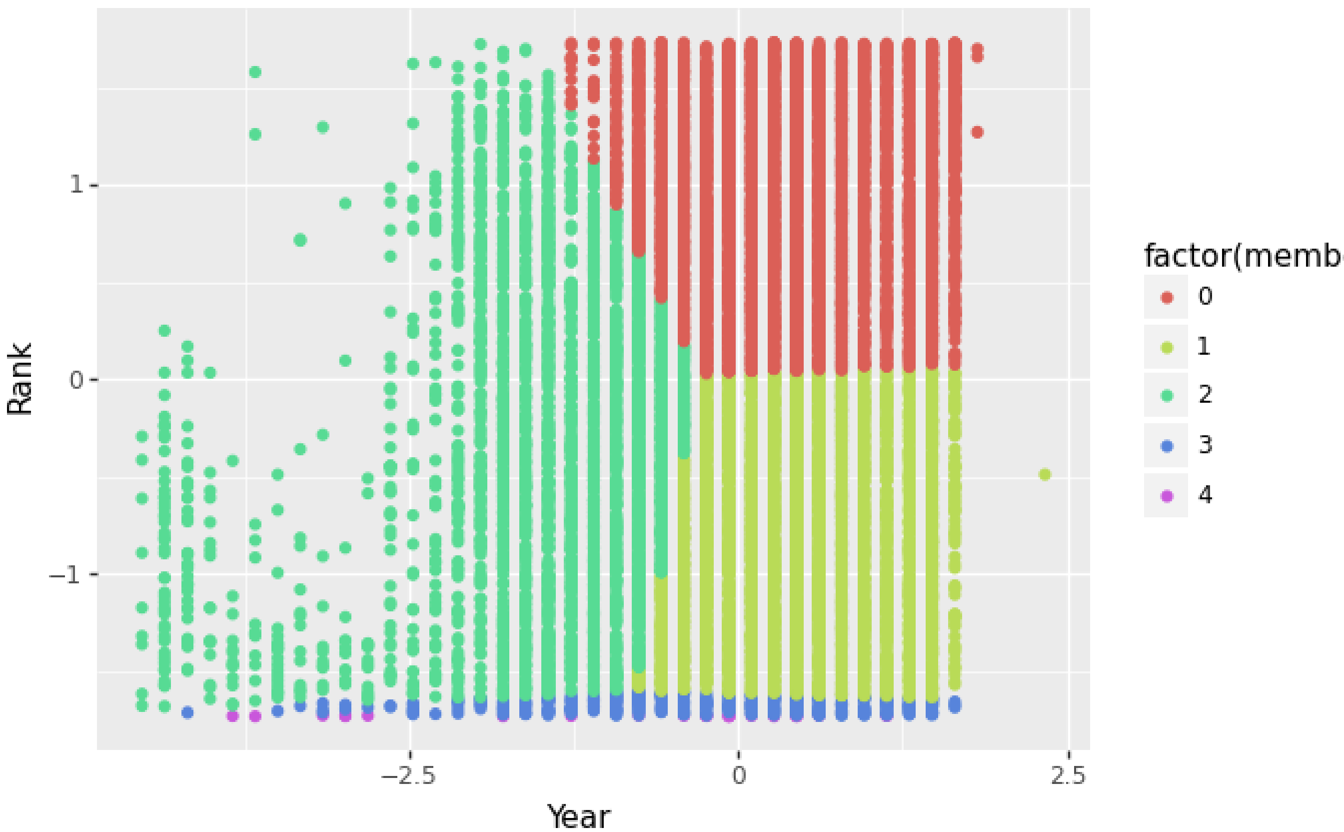

```
features = ["Year", "Rank","Global_Sales"]
X = data_clean[features]

z = StandardScaler()
X[["Year", "Rank","Global_Sales"]] = z.fit_transform(X)

(ggplot(X, aes("Year", "Rank")) + geom_point())
```




```
<ggplot: (8753042717616)>
```

```
KM = KMeans(n_clusters = 5)
KM.fit(X)
```

```
KM.fit(X)
membership = KM.predict(X)

X["cluster"] = membership

(ggplot(X, aes("Year", "Rank", color = "factor(membership)")) + geom_point())
```

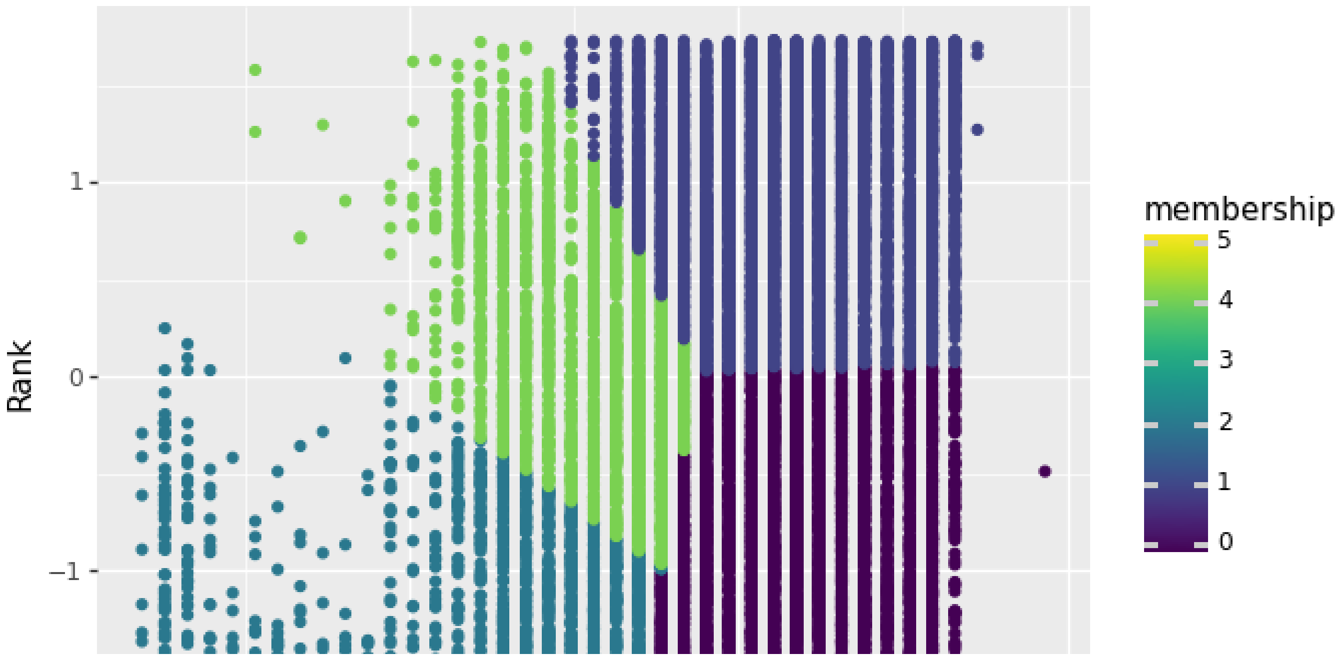



```
<ggplot: (-9223363283812263841)>
```

```
silhouette_score(X, membership)
```

```
0.4947733
```

```
KM = KMeans(n_clusters = 6)
KM.fit(X)
membership = KM.predict(X)

Xall = X
Xall["cluster"] = membership

(ggplot(X, aes("Year", "Rank", color = "membership")) + geom_point())
```




```
silhouette_score(X, membership)
```

```
0.5516621
```

The answer is that there is a relationship between the year a game was released and the rank it stand
clusters since it also has a higher silhouette score of .61, the green cluster is evenly spread out over t
that in the first ci=ouple of years, the ranking was very xporadic since not too many games were relea
very condensed distribution of the games since they are very frequently released now.

Does the genre of the game affect the sales in specific large countries?

```
genre_type = ('Action', 'Adventure', 'Fighting', 'Misc', 'Platform', 'Puzzle',
       'Racing', 'Role-Playing', 'Shooter', 'Simulation', 'Sports',
       'Strategy')

data["Genre_Type"] = "Genre_Type"
data.loc[(data["Genre"] == "Action"),"Genre_Type"] = 0
data.loc[(data["Genre"] == "Adventure"),"Genre_Type"] = 1
data.loc[(data["Genre"] == "Fighting"),"Genre_Type"] = 2
data.loc[(data["Genre"] == "Misc"),"Genre_Type"] = 3
data.loc[(data["Genre"] == "Platform"),"Genre_Type"] = 4
data.loc[(data["Genre"] == "Puzzle"),"Genre_Type"] = 5
data.loc[(data["Genre"] == "Racing"),"Genre_Type"] = 6
data.loc[(data["Genre"] == "Role-Playing"),"Genre_Type"] = 7
data.loc[(data["Genre"] == "Shooter"),"Genre_Type"] = 8
data.loc[(data["Genre"] == "Simulation"),"Genre_Type"] = 9
data.loc[(data["Genre"] == "Sports"),"Genre_Type"] = 10
data.loc[(data["Genre"] == "Strategy"),"Genre_Type"] = 11

data_gen = data
```

```
data_gen.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27

```
feats = ["Genre_Type","NA_Sales", "EU_Sales", "JP_Sales"]

X = data_gen[feats]


z = StandardScaler()
X[feats] = z.fit_transform(X)


EM = GaussianMixture(n_components = 5)

EM.fit(X)
```

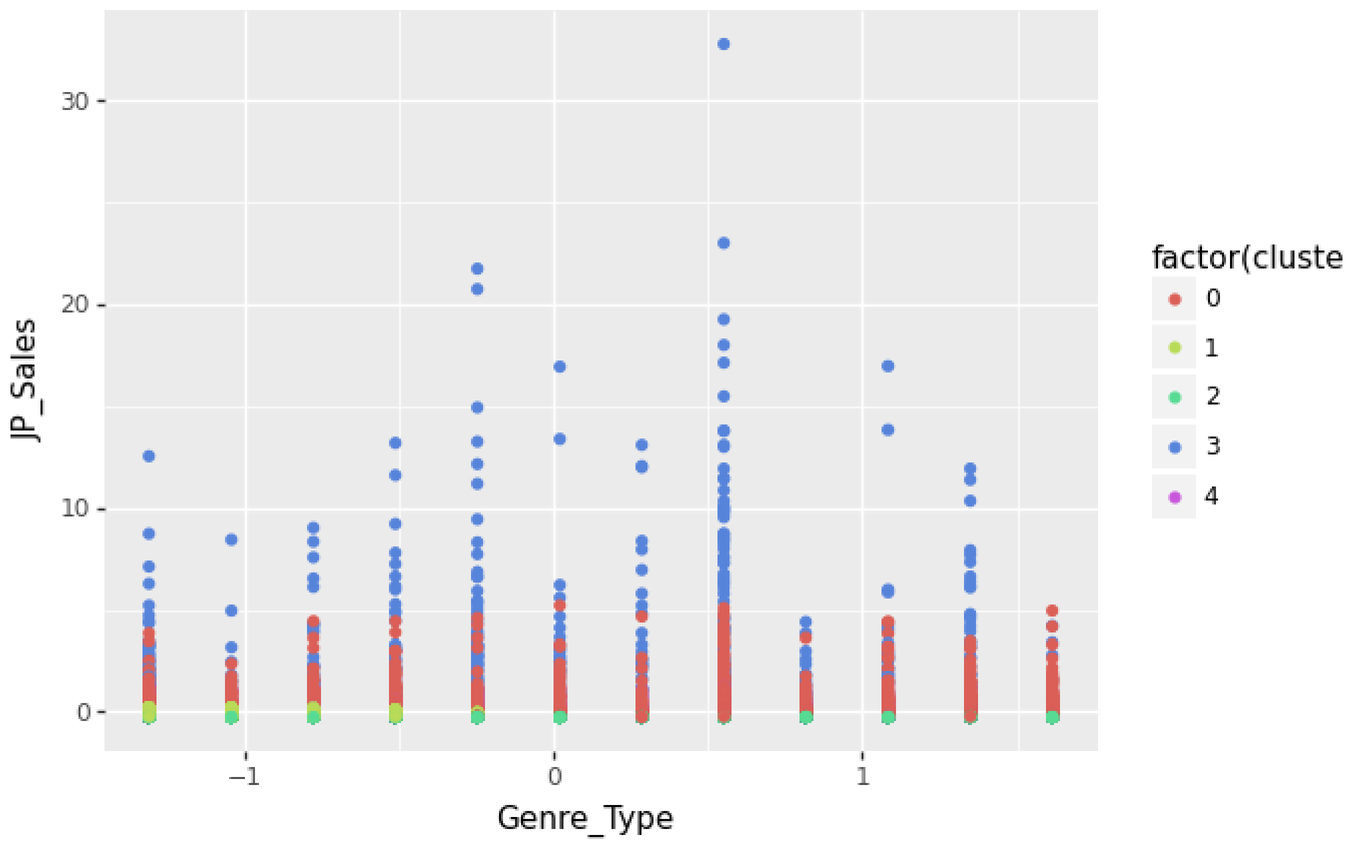
```
GaussianMixture(covariance_type='full', init_params='kmeans', max_iter=100,
                means_init=None, n_components=5, n_init=1, precisions_init=None,
                random_state=None, reg_covar=1e-06, tol=0.001, verbose=0,
                verbose_interval=10, warm_start=False, weights_init=None)
```

```
cluster = EM.predict(X)

silhouette_score(X, cluster)
```

```
-0.1403361
```

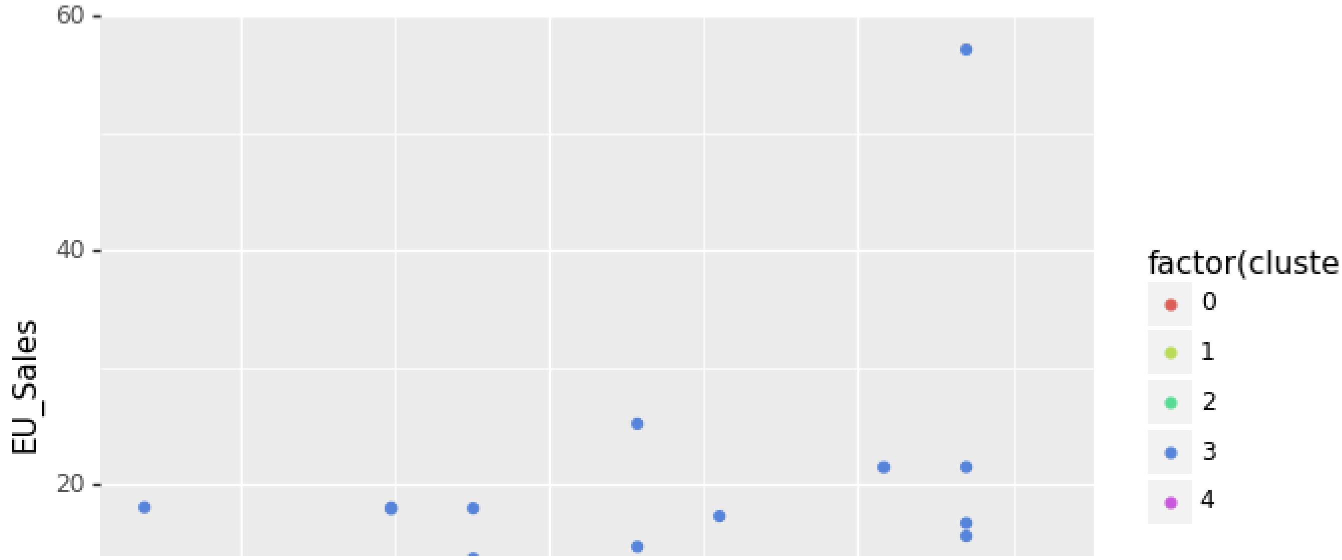
```
X["cluster"] = cluster

(ggplot(X, aes(x = "Genre_Type", y = "NA_Sales", color = "factor(cluster)")) + geom_po
```


```
(ggplot(X, aes(x = "Genre_Type", y = "JP_Sales", color = "factor(cluster)")) + geom_pc
```




```
<ggplot: (8753042333367)>
```

```
(ggplot(X, aes(x = "Genre_Type", y = "EU_Sales", color = "factor(cluster)")) + geom_pc
```



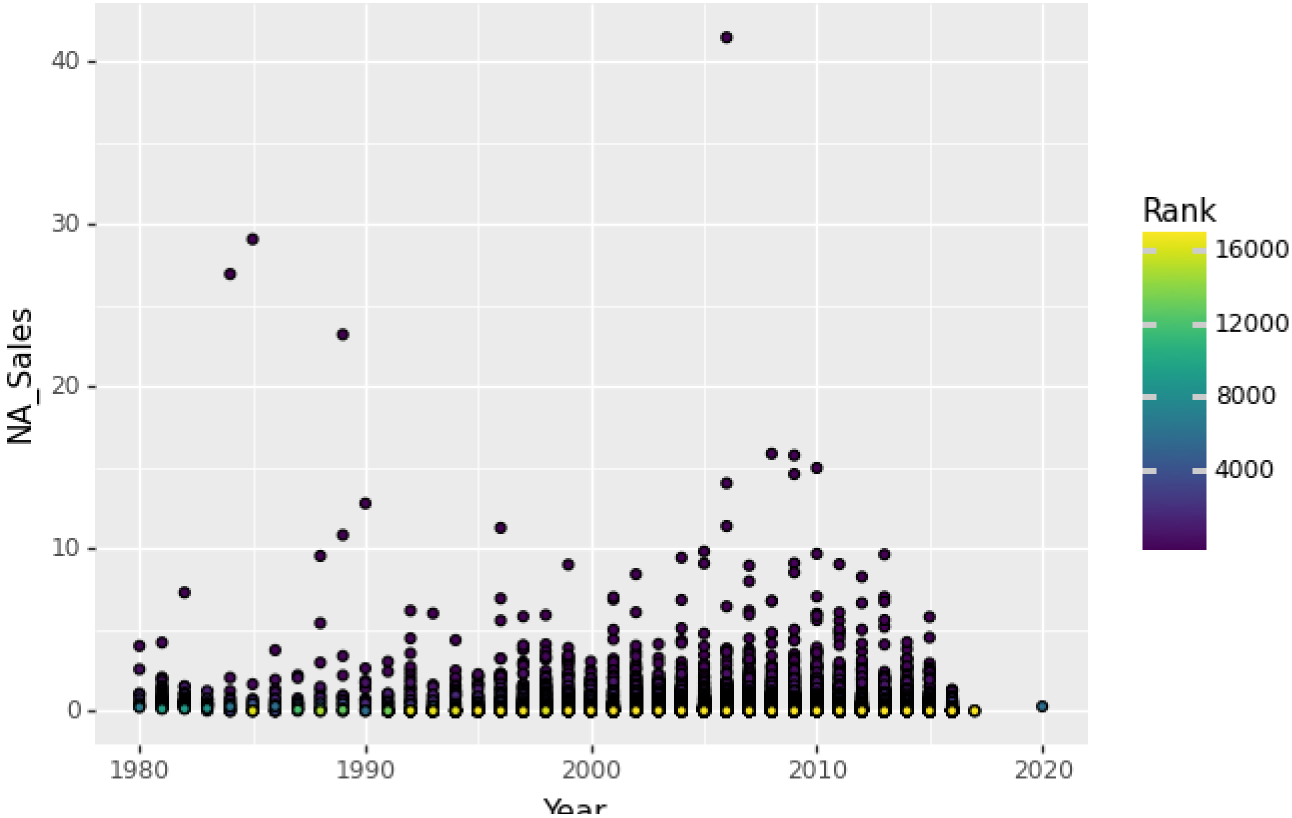
The answer is yes, the genre does affect sales in each of the major regions, but to varying degrees. W
see which genre is resulting in the highest number of sales, but it is not consistent. This helps us com
preferences per region and from a marketing perspective, we can see which genre of game will result

Which factor has the highest impact in determining rank, if any?

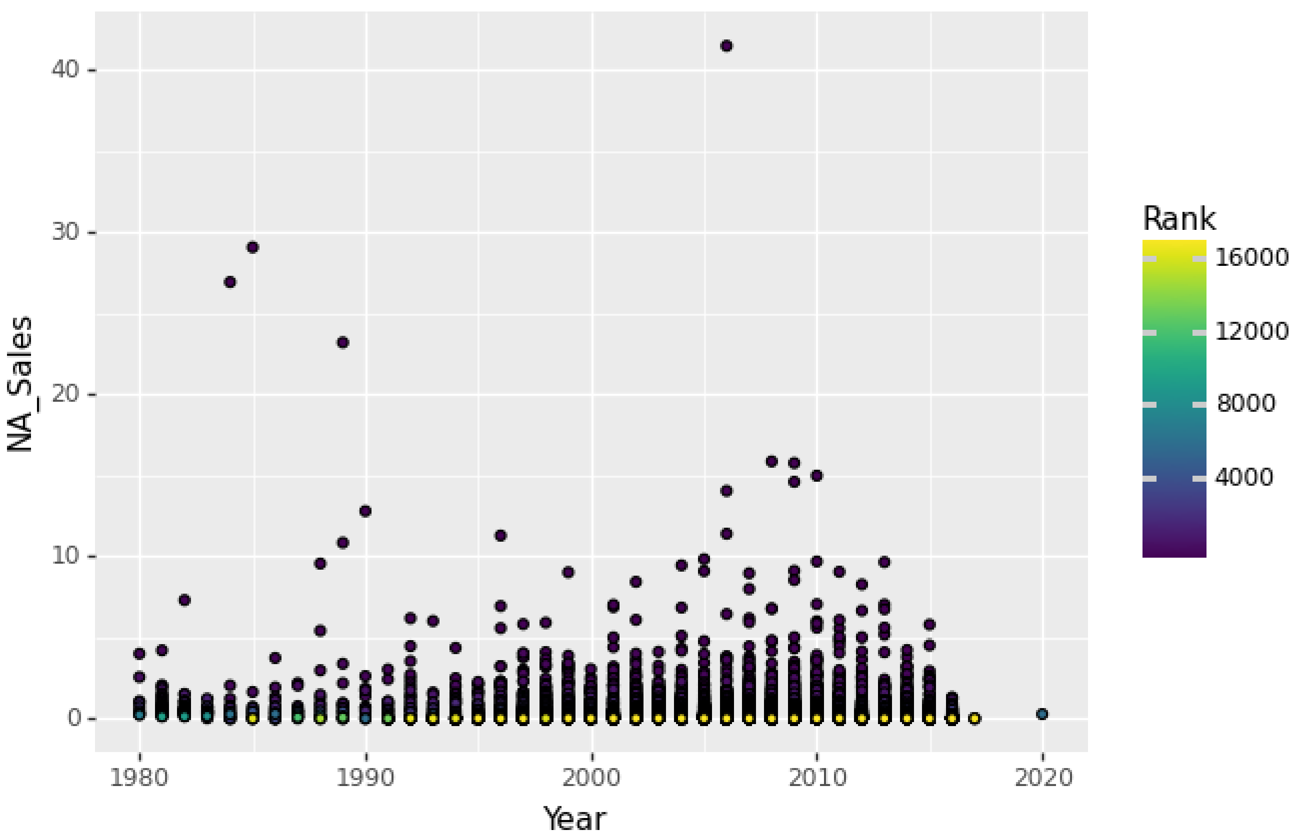
```
feat = ["Year","NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales"]
X = data_clean[feat]
y = data_clean["Rank"]

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)

z = StandardScaler()

X_train[feat] = z.fit_transform(X_train[feat])
X_test[feat] = z.transform(X_test[feat])

X_train.head()
```

	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales
7289	0.274004	-0.093486	-0.262996	-0.260501	-0.145045
12384	0.788507	-0.267044	-0.262996	-0.260501	-0.244405
14592	-1.784008	-0.304235	-0.262996	-0.260501	-0.244405
9824	-0.755002	-0.254647	-0.187315	-0.260501	-0.145045
4212	-0.240499	0.154455	-0.262996	-0.260501	0.053675

```
(ggplot(data_clean,aes(x = "Year", y = "NA_Sales", fill = "Rank")) + geom_point())
```

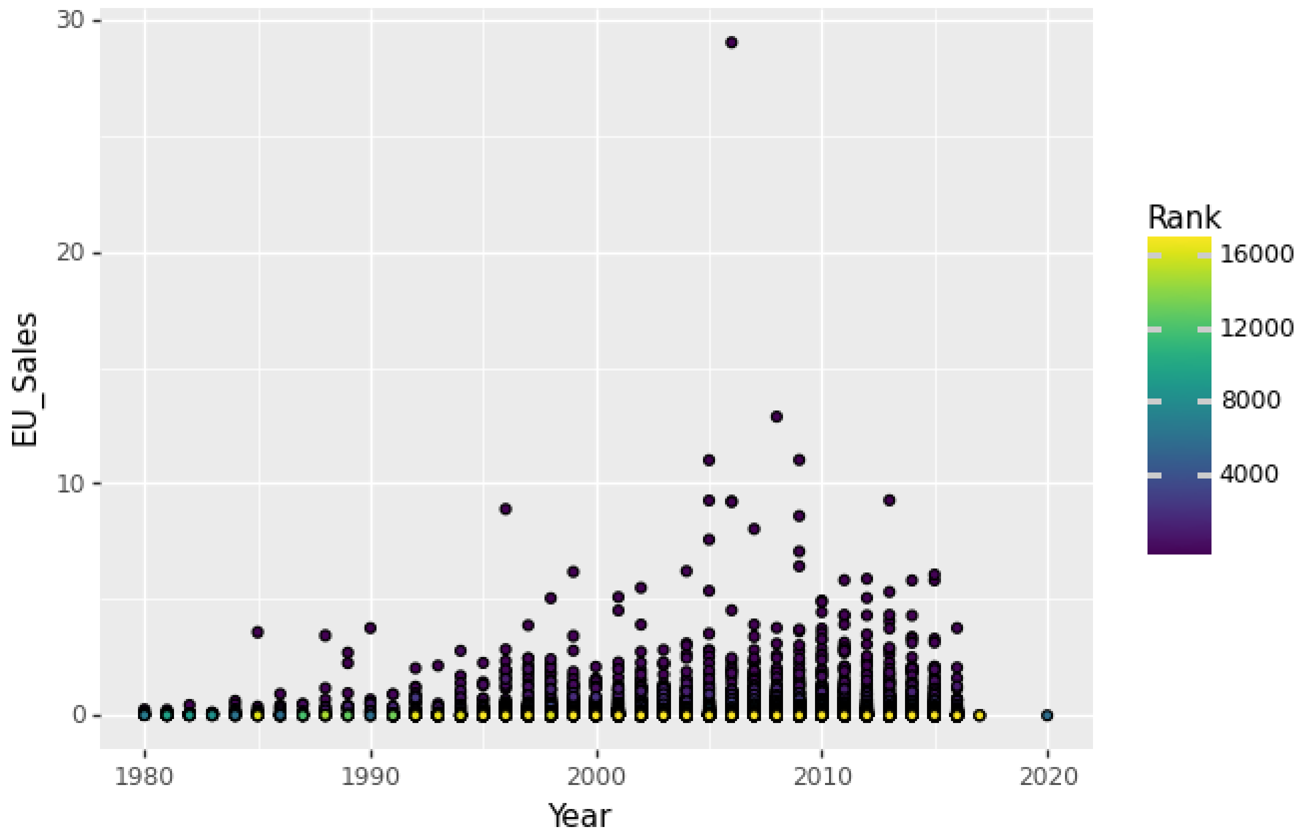



```
(ggplot(data_clean,aes(x = "Year", y = "NA_Sales", fill = "Rank")) + geom_point())
```



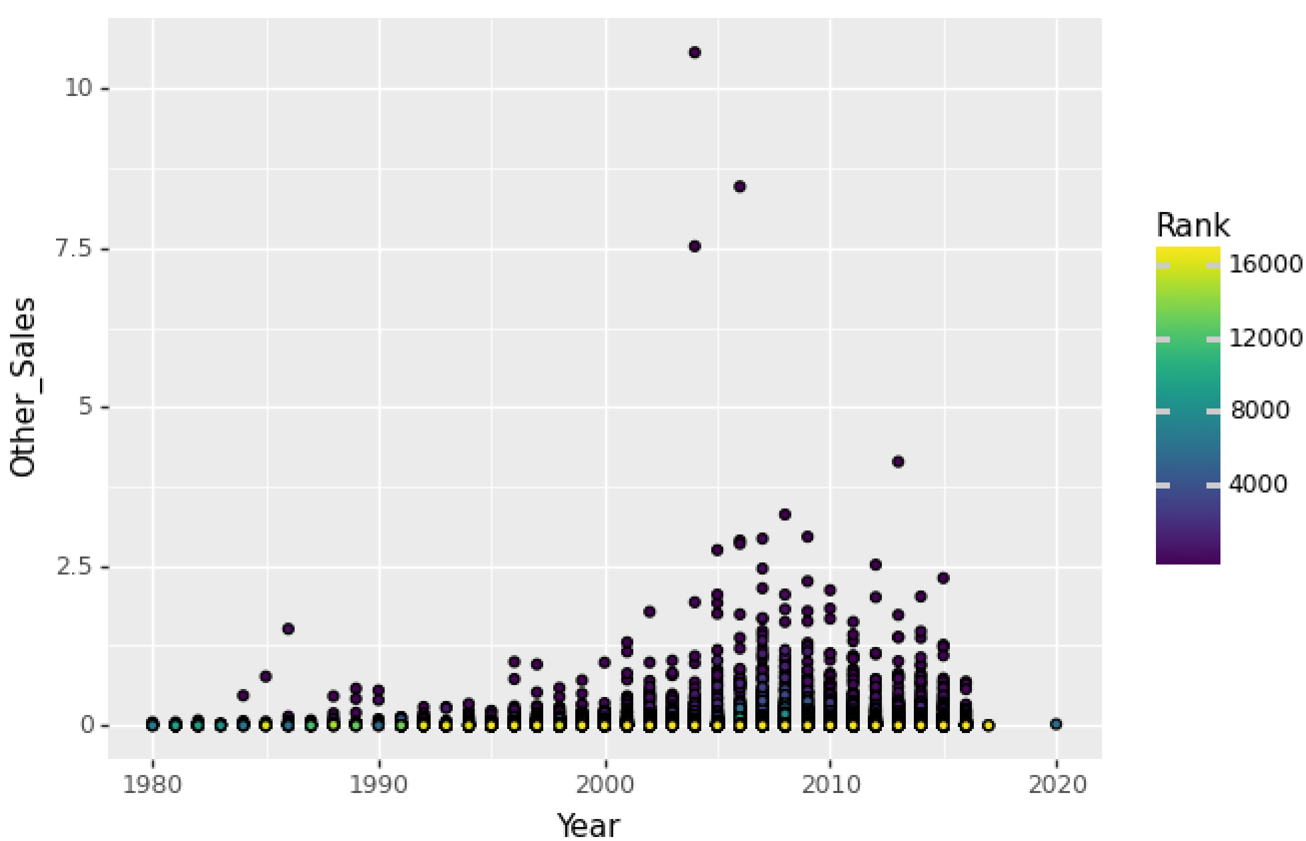
<ggplot: (8753041707412)>

```
(ggplot(data_clean,aes(x = "Year", y = "EU_Sales", fill = "Rank")) + geom_point())
```




```
<ggplot: (8753041929143)>
```

```
(ggplot(data_clean,aes(x = "Year", y = "Other_Sales", fill = "Rank")) + geom_point())
```




```
<ggplot: (-9223363283813139747)>
```

```
lsr = Lasso()

lsr.fit(X_train,y_train)

print("TRAIN: ", mean_absolute_error(y_train, lsr.predict(X_train)))
print("TEST : ", mean_absolute_error(y_test, lsr.predict(X_test)))
```

```
TRAIN:  3608.761953412216
TEST :  3615.394712259695
```

```
coefficients = pd.DataFrame({"Coef":lsr.coef_,
              "Name":X})
coefficients = coefficients.append({"Coef": lsr.intercept_, "Name": "intercept"}, ignc
coefficients
```

	Coef	Name
0	690.776298	(Y, e, a, r)
1	-1071.927344	(N, A, _, S, a, l, e, s)
2	-435.914889	(E, U, _, S, a, l, e, s)
3	-430.906884	(J, P, _, S, a, l, e, s)
4	-443.086793	(O, t, h, e, r, _, S, a, l, e, s)
5	8263.594153	intercept

When looking at the individual coefficients the factors that are affecting rank, for every one unit stand increase of 730 in rank. For every unit increase in NA_Sales, there is a decrease in rank by 761. For eve 624 decrease in rank. For every unit increase in JP_Sales, there is a decrease of 412 in rank. For every 564 decrease in rank. Using the coefficients and the ggplots, you can see that how sporadically spars major regions.

Alberto Ng

How does genre affect global sales, if it does?

```
data_clean.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85

```
data_clean.shape
```

```
(16291, 11)
```

```
cutoff = data_clean["Rank"].max()*0.2

data_clean['TopGS'] = 'zzz'
data_clean.loc[(data_clean['Rank'] > cutoff), 'TopGS'] = "0"
data_clean.loc[(data_clean['Rank'] <= cutoff), 'TopGS'] = "1"

data_LR = data_clean

data_LR.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27

```
dummy = pd.get_dummies(data_LR.Genre)

data_dum = pd.concat([data_LR, dummy], axis = 1)

data_LR = data_dum.drop(['Name','Platform','Genre','Publisher', 'NA_Sales', 'EU_Sales'

new_cutoff = data_clean["Rank"].max()*0.4

data_LR = data_LR[data_LR["Rank"] <= new_cutoff]
data_LR = data_LR.drop(['Rank'], axis=1)

dataLR = data_dum[data_dum["Rank"] <= new_cutoff]
dataLR = dataLR.loc[:,~dataLR.columns.duplicated()]

dataLR.head()
```

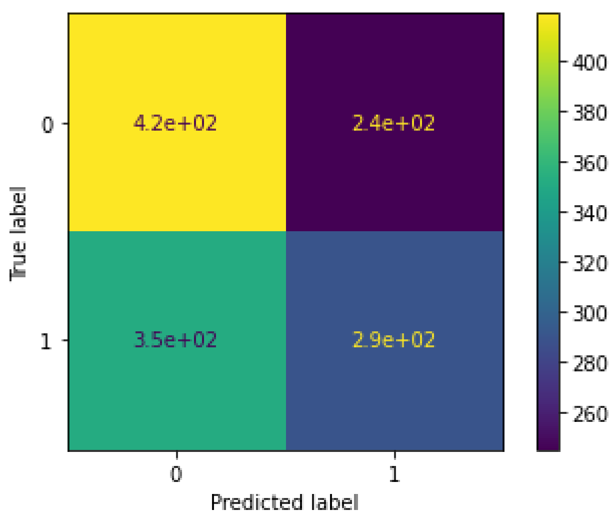
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sa
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	

```
X = data_LR.loc[:, data_LR.columns != 'TopGS']
y = data_LR["TopGS"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

LGMod = LogisticRegression()
LGMod = LGMod.fit(X_train, y_train)

GS_pred = LGMod.predict(X_test)
print(accuracy_score(y_test, GS_pred))
```

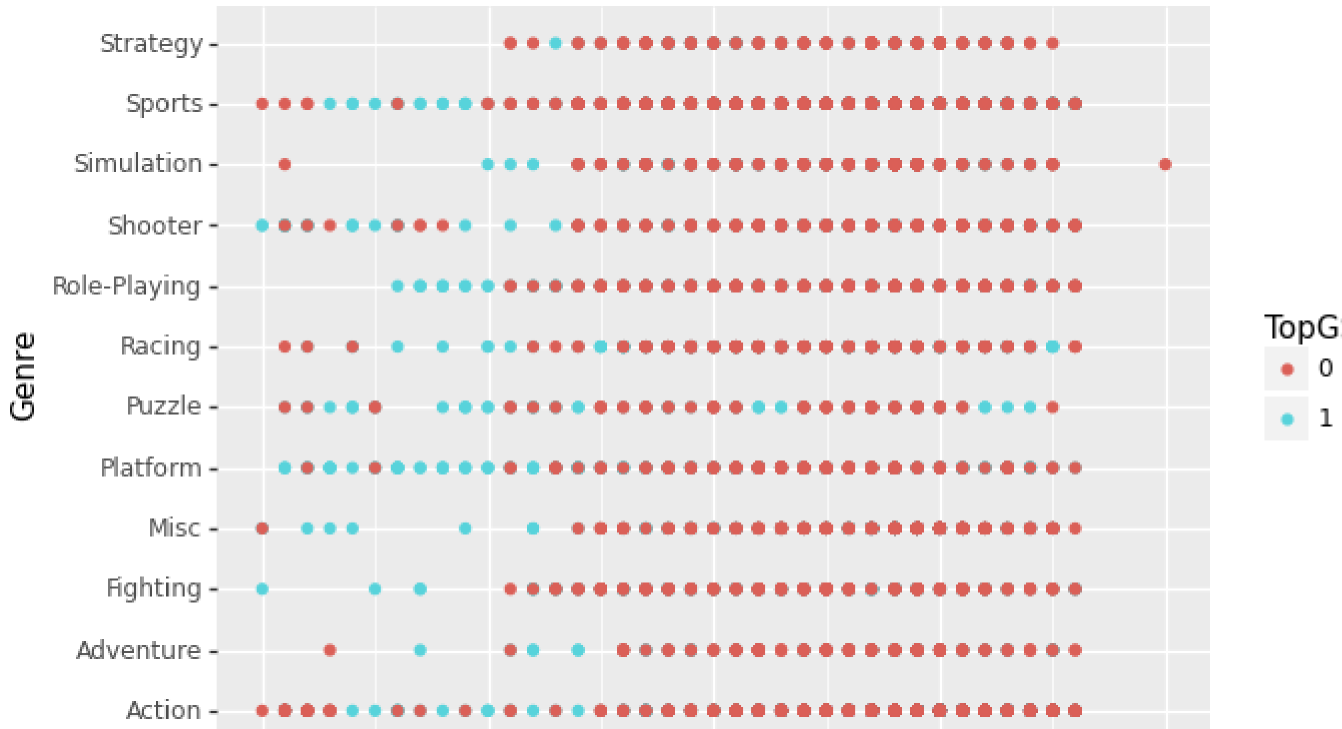
```
0.5428790199081164
```

```
plot_confusion_matrix(LGMod, X_test, y_test)
```

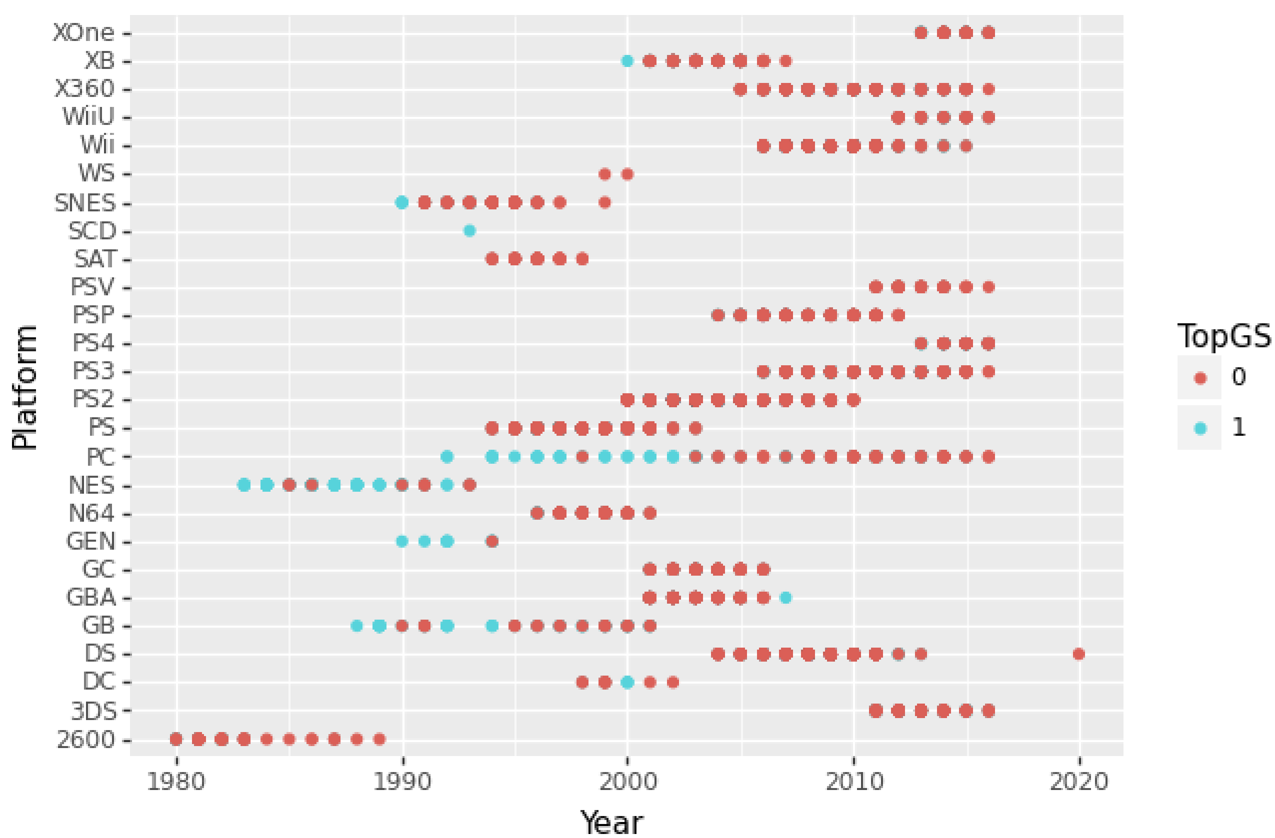
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f5f9f01a208>
```




```
(ggplot(dataLR, aes("Year", "Genre")) +
geom_point(aes(color = "TopGS")))
```




```
(ggplot(dataLR, aes("Year", "Platform")) +
geom_point(aes(color = "TopGS")))
```



<ggplot: (8753041618408)>

When looking at the ggplots and confusion matrix, you can see that there is not a correlation between
colored by TopGS, "0" represents the top 20-40% and the "1" represents the top 20%. The plot confusic

matrix of 0.54 shows us that it can not accurately predict the majority of the time.

Is there a relationship between Global_Sales and the platform?

```
cutoff = data_clean["Rank"].max()*0.2

data_clean['TopGS'] = 'zzz'
data_clean.loc[(data_clean['Rank'] > cutoff), 'TopGS'] = "0"
data_clean.loc[(data_clean['Rank'] <= cutoff), 'TopGS'] = "1"

data_LR = data_clean


dummy = pd.get_dummies(data_LR.Platform)

data_dum = pd.concat([data_LR, dummy], axis = 1)

data_LASSO = data_dum.drop(['Name','Platform','Genre','Publisher', 'NA_Sales', 'EU_Sal

new_cutoff = data_clean["Rank"].max()*0.4

data_LASSO = data_LASSO[data_LASSO["Rank"] <= new_cutoff]
data_LASSO = data_LASSO.drop(['Rank'], axis=1)

dataLASSO = data_dum[data_dum["Rank"] <= new_cutoff]

dataLASSO.head()
```

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

```
X = data_LASSO.loc[:, data_LASSO.columns != 'TopGS']
y = data_LASSO["TopGS"]


kf = KFold(n_splits = 15)
kf.split(X)

lasso = Lasso()
```

```
zscore = StandardScaler()

train_mae = []
test_mae = []


for train_indices, test_indices in kf.split(X):
    X_train = X.iloc[train_indices]
    X_test  = X.iloc[test_indices]
    y_train = y.iloc[train_indices]
    y_test  = y.iloc[test_indices]

    zscore.fit(X_train)
    X_test = zscore.transform(X_test)

    model = lasso.fit(X_train, y_train)
    train_mae.append(mean_absolute_error(y_train, lasso.predict(X_train)))
    test_mae.append(mean_absolute_error(y_test, lasso.predict(X_test)))

np.mean(train_mae)
```

```
0.4976171
```

```
np.mean(test_mae)
```

```
0.5333341
```

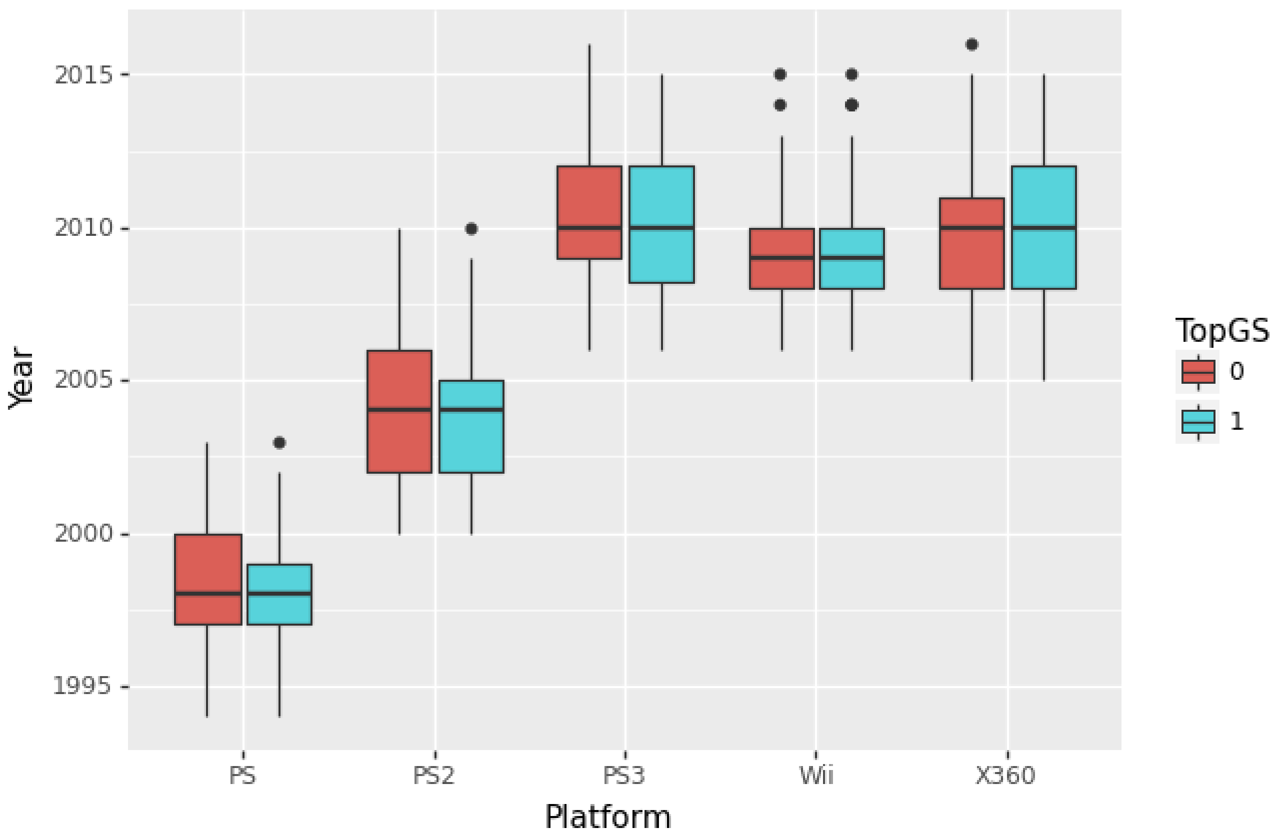
```
dataLASSO.Platform[:int(cutoff)].value_counts().nlargest(5)
```

```
PS2     493
PS3     421
X360    377
PS      321
Wii     267
Name: Platform, dtype: int64
```

```
dataLASSO = dataLASSO.loc[(dataLASSO.Platform == "PS2") | (dataLASSO.Platform == "PS3'
dataLASSO.head()
```

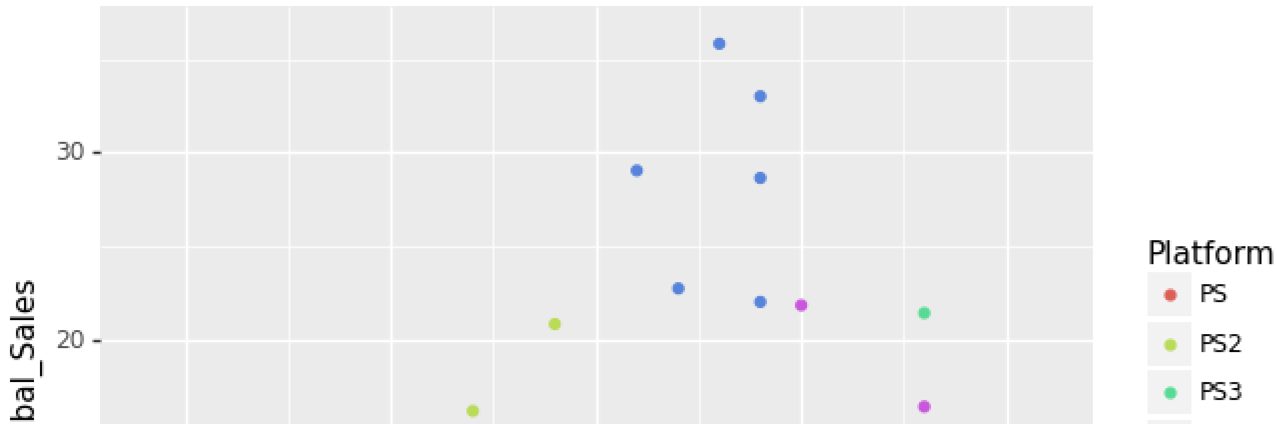
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77
2	3	Mario Kart	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79

```
(ggplot(dataLASSO, aes("Platform", "Year")) +
geom_boxplot(aes(fill = "TopGS")))
```




```
<ggplot: (8753041567061)>
```

```
(ggplot(dataLASSO, aes("Year", "Global_Sales")) +
geom_point(aes(color = "Platform")) +
ylim(0, 36))
```



While looking at the second ggplot, you can see that WII and X360 perform the best with global sales, mean of the mean absolute error from k-fold cv, the model wasn't overfitted by looking at the train anc platforms are also the newest out of all 5, which is why their global sales are a lot higher compared to

Do publishers affect global sales throughout the year?

```
data_EM = data_clean

#dummy = pd.get_dummies(data_EM.Publisher)

#data_dum = pd.concat([data_EM, dummy], axis = 1)

#data_EM = data_dum.drop(['Name','Platform','Genre'], axis=1)

data_EM = data_EM.drop(['Name','Platform','Genre'], axis=1)

new_cutoff = data_clean["Rank"].max()*0.4

data_EM = data_EM[data_EM["Rank"] <= new_cutoff]
data_EM = data_EM.drop(['Rank'], axis=1)


data_EM.Publisher[:int(cutoff)].value_counts().nlargest(5)
```

```
Electronic Arts                 553
Nintendo                        414
Activision                      265
Sony Computer Entertainment     244
Ubisoft                         204
Name: Publisher, dtype: int64
```

```
data_EM['publisher'] = 5
data_EM.loc[(data_EM['Publisher'] == "Electronic Arts"), 'publisher'] = 0
data_EM.loc[(data_EM['Publisher'] == "Nintendo"), 'publisher'] = 1
data_EM.loc[(data_EM['Publisher'] == "Activision"), 'publisher'] = 2
data_EM.loc[(data_EM['Publisher'] == "Sony Computer Entertainment"), 'publisher'] = 3
data_EM.loc[(data_EM['Publisher'] == "Ubisoft"), 'publisher'] = 4
```

```
data_EM = data_EM.drop(['Publisher'], axis=1)
data_EM.head()
```

	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	TopGS	publishe
0	2006.0	41.49	29.02	3.77	8.46	82.74	1	
1	1985.0	29.08	3.58	6.81	0.77	40.24	1	
2	2008.0	15.85	12.88	3.79	3.31	35.82	1	
3	2009.0	15.75	11.01	3.28	2.96	33.00	1	
4	1996.0	11.27	8.89	10.22	1.00	31.37	1	

```
X = data_EM.loc[:]
```

```
EM_Mod = GaussianMixture(n_components = 2)
```

```
EM_Mod.fit(X)
```

```
GaussianMixture(covariance_type='full', init_params='kmeans', max_iter=100,
                means_init=None, n_components=2, n_init=1, precisions_init=None,
                random_state=None, reg_covar=1e-06, tol=0.001, verbose=0,
                verbose_interval=10, warm_start=False, weights_init=None)
```

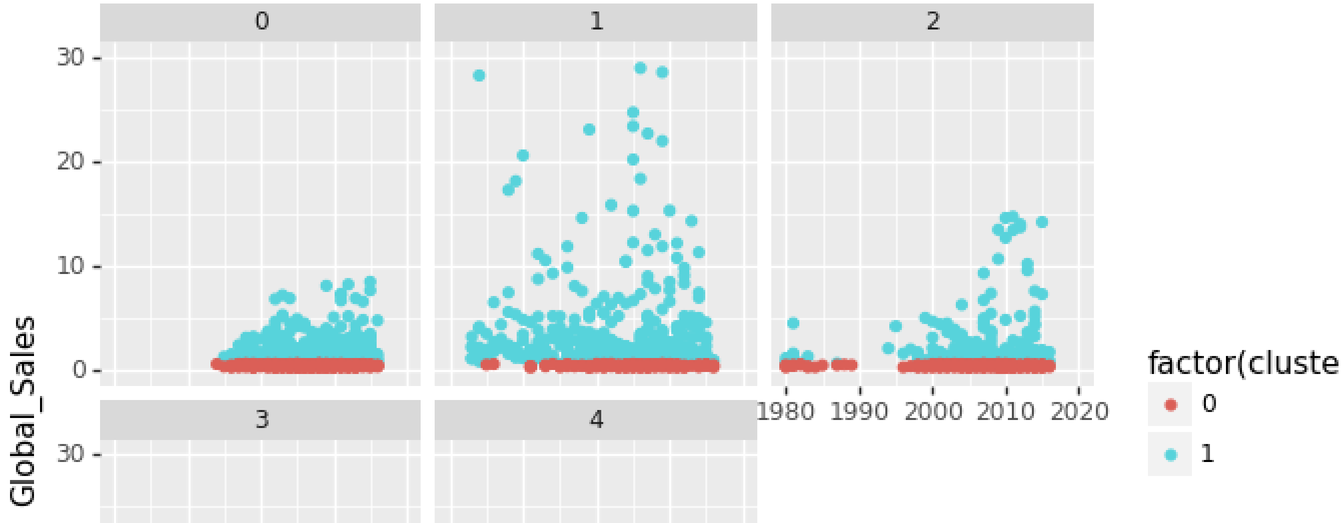
```
clusters = EM_Mod.predict(X)
silhouette_score(X, clusters)
```

```
0.04282681
```

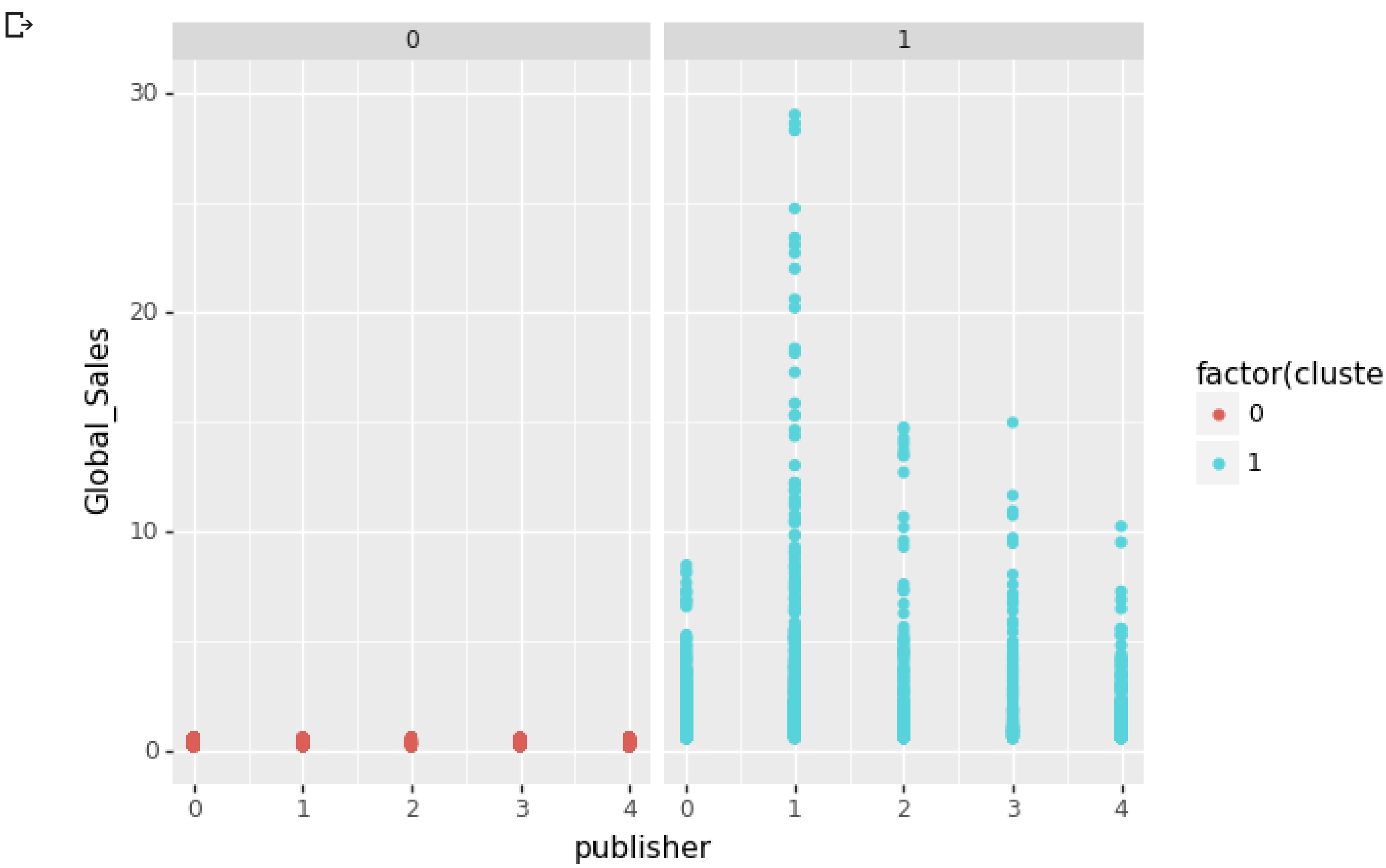
```
X["clusters"] = clusters
```

```
X = X.loc[(X.publisher == 0) | (X.publisher == 1) | (X.publisher == 2) | (X.publisher
```

```
(ggplot(X, aes("Year", "Global_Sales")) +
geom_point(aes(color = "factor(clusters)")) +
facet_wrap("publisher") +
ylim(0,30))
```




```
(ggplot(X, aes("publisher","Global_Sales")) +
geom_point(aes(color = "factor(clusters)")) +
facet_wrap("TopGS") +
ylim(0, 30))
```




```
<ggplot: (-9223363283811934541)>
```

When looking at the ggplots, you can see that publishers do affect global sales. For the first plot, publi
3(Sony) did the best and all dramatically affect global sales. Specifically for publisher 2, you can see tl
the course of year, the trend is the most interesting and obvious even though it isn't the highest perfor
ahowing us that this model has low cohesion and separation between clusters, but we are still able to