Visualization

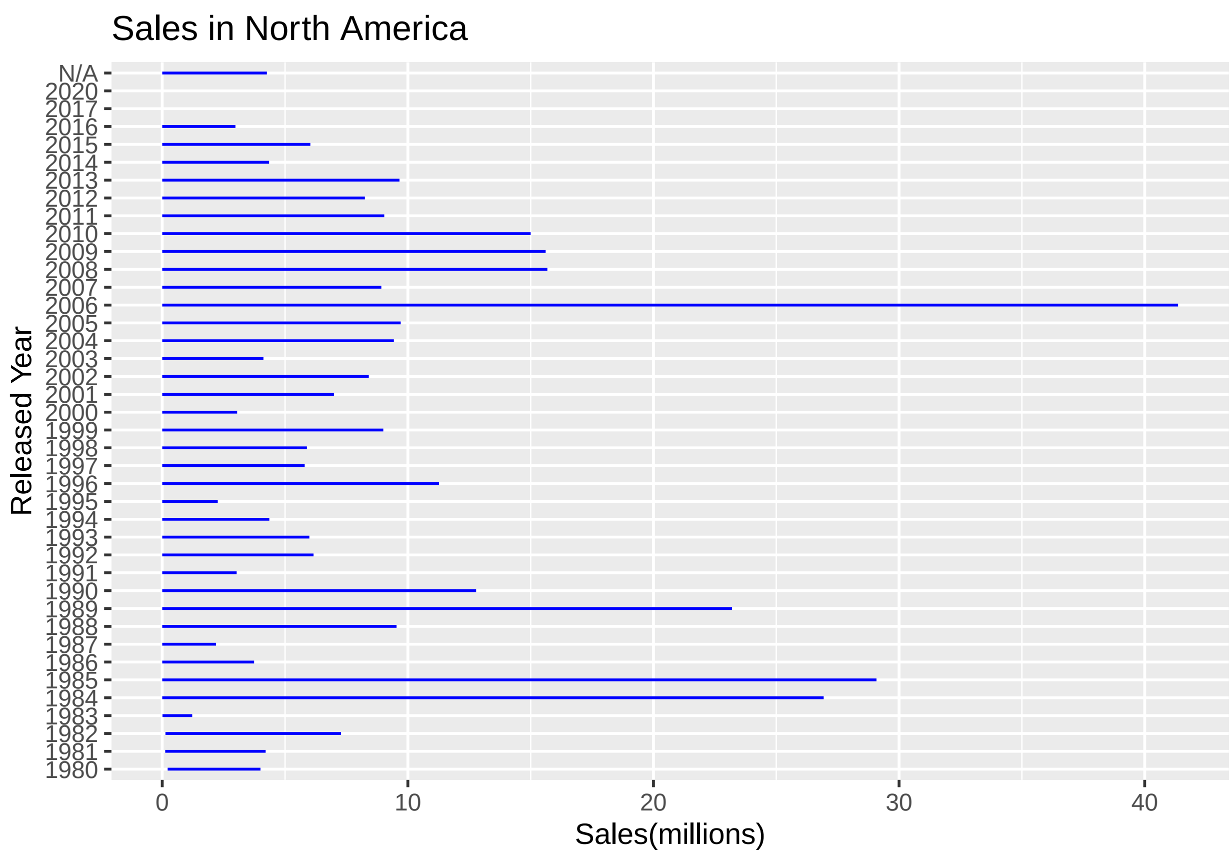
```r
library(readxl)
data0 <- read_excel("Video_Games_Sales_as_at_22_Dec_2016.xlsx")
#View(data0)
```

```r
#install.packages("ggplot2")
library(ggplot2)
```

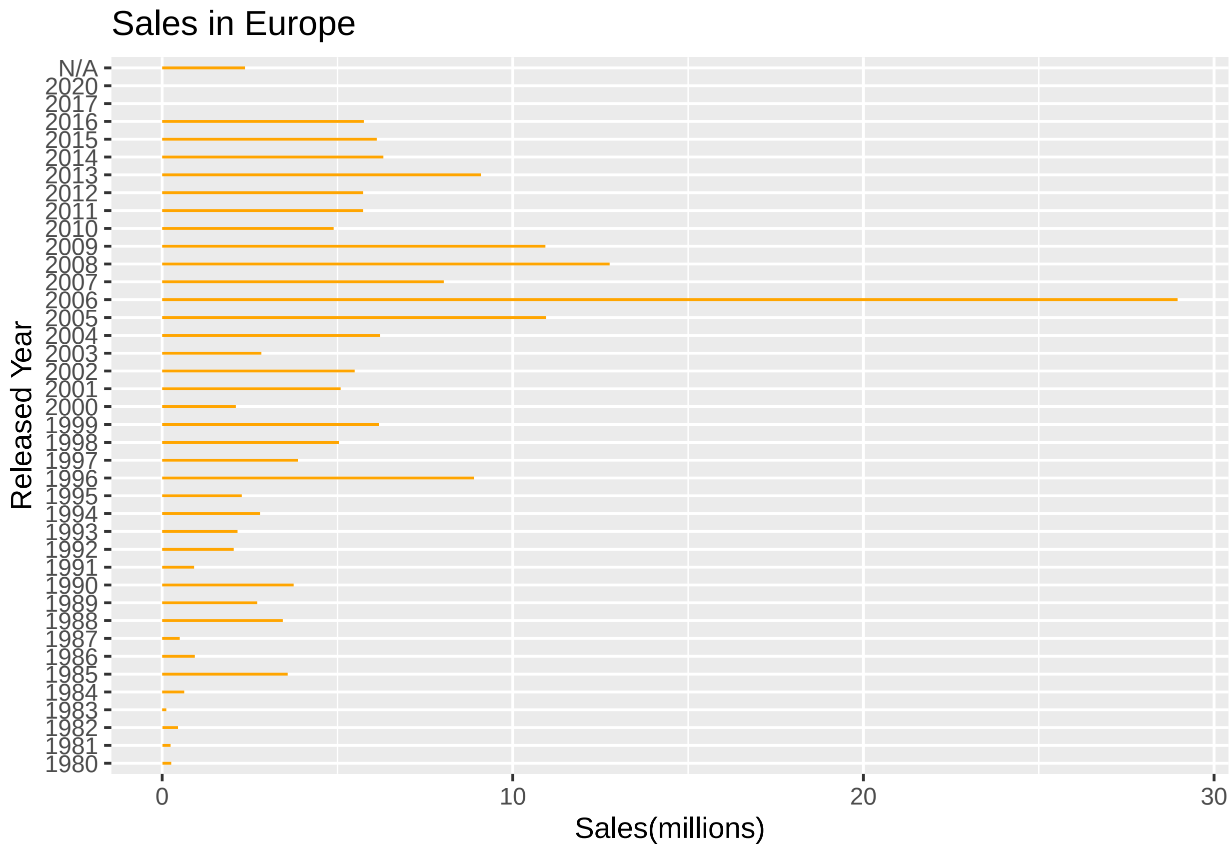
```r
str(data0)
```

```
## tibble [16,719 x 16] (S3: tbl_df/tbl/data.frame)
##  $ Name           : chr [1:16719] "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Reso
##  $ Platform       : chr [1:16719] "Wii" "NES" "Wii" "Wii" ...
##  $ Year_of_Release: chr [1:16719] "2006" "1985" "2008" "2009" ...
##  $ Genre          : chr [1:16719] "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher      : chr [1:16719] "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
##  $ NA_Sales       : num [1:16719] 41.4 29.1 15.7 15.6 11.3 ...
##  $ EU_Sales       : num [1:16719] 28.96 3.58 12.76 10.93 8.89 ...
##  $ JP_Sales       : num [1:16719] 3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales    : num [1:16719] 8.45 0.77 3.29 2.95 1 0.58 2.88 2.84 2.24 0.47 ...
##  $ Global_Sales   : num [1:16719] 82.5 40.2 35.5 32.8 31.4 ...
##  $ Critic_Score   : num [1:16719] 76 NA 82 80 NA NA 89 58 87 NA ...
##  $ Critic_Count   : num [1:16719] 51 NA 73 73 NA NA 65 41 80 NA ...
##  $ User_Score     : chr [1:16719] "8" NA "8.3000000000000007" "8" ...
##  $ User_Count     : num [1:16719] 322 NA 709 192 NA NA 431 129 594 NA ...
##  $ Developer      : chr [1:16719] "Nintendo" NA "Nintendo" "Nintendo" ...
##  $ Rating         : chr [1:16719] "E" NA "E" "E" ...
```

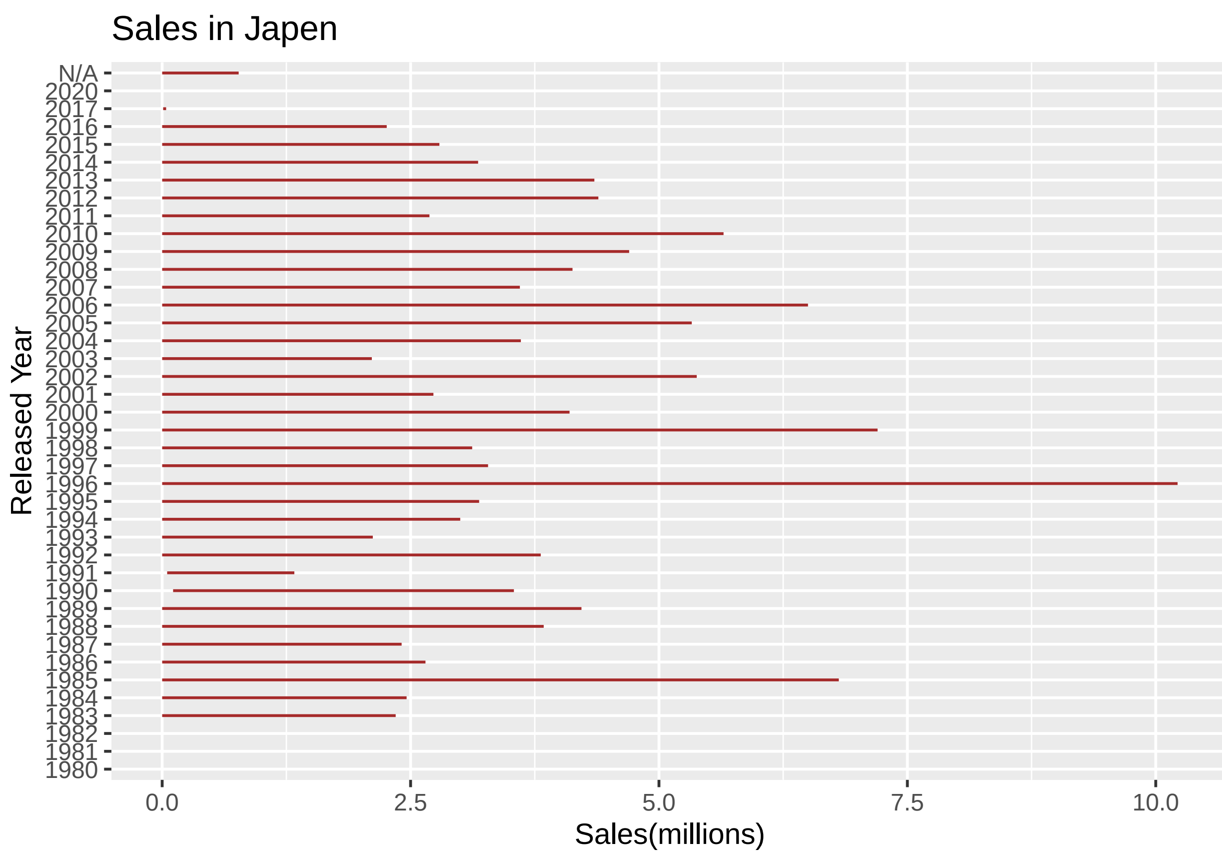
```r
ggplot(data0,aes(NA_Sales,Year_of_Release))+geom_line(color="blue")+labs(title="Sales in North America
```



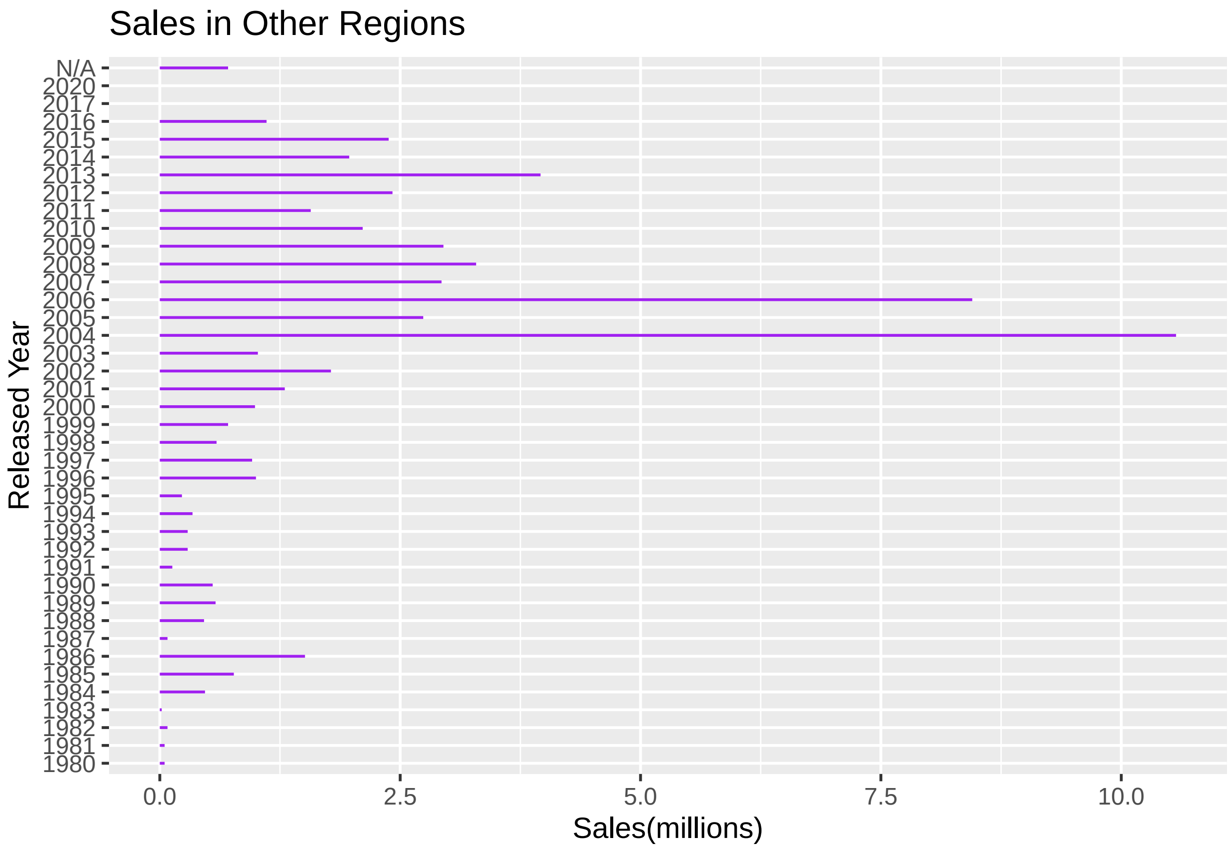

```
ggplot(data0,aes(EU_Sales,Year_of_Release))+geom_line(color="orange")+labs(title="Sales in Europe ",x="S
```



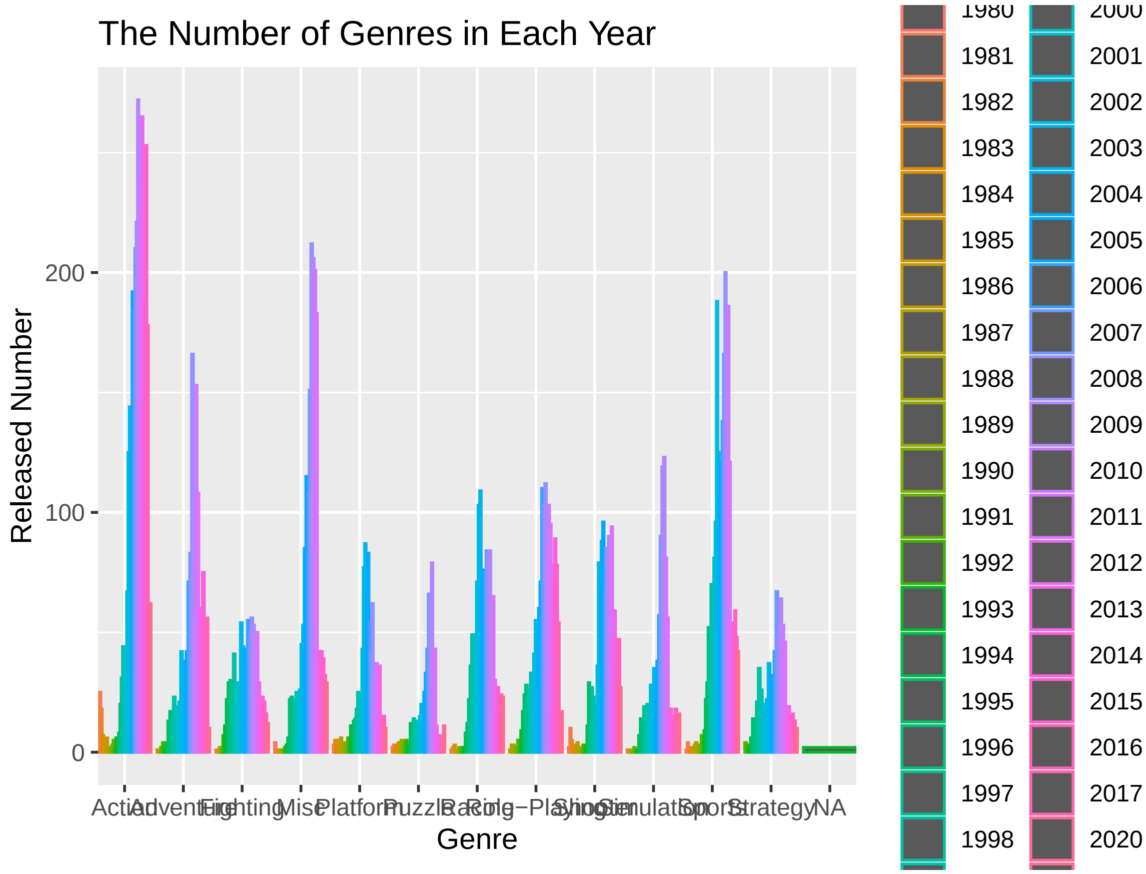

```
ggplot(data0,aes(JP_Sales,Year_of_Release))+geom_line(color="brown")+labs(title="Sales in Japen ",x="Sa
```




```
ggplot(data0,aes(Other_Sales,Year_of_Release))+geom_line(color="purple")+labs(title="Sales in Other Reg
```




```
ggplot(data0,aes(factor(Genre),color=factor(Year_of_Release)))+geom_bar(position="dodge")+scale_x_discre
```



‘

LOGISTIC REGRESSION

1) Data Cleaning

Removing Null value roes and converting non-numeric columns to numeric columns

`r vg <- read.csv("Video_Games_Sales.csv",header = TRUE,na.strings = c("", "N/A")) vg <- na.omit(vg) vg$User_Count<-as.numeric(as.character(vg$User_Count)) vg$User_Score<-as.numeric(as.character(vg$User_Score))`

Refining categorical variable

`r vg$Publisher2=0 vg$Publisher2[(vg$Publisher=="Nintendo")|(vg$Publisher=="Activision")|(vg$Publisher=="Sony Computer Entertainment")|(vg$Publisher=="Electronic Arts")|(vg$Publisher=="Take-Two Interactive")|(vg$Publisher=="Ubisoft")]=1 a <- vg[vg$Publisher2==1,] dt = sort(sample(nrow(a), nrow(a)*0.6)) train<-a[dt,] test<-a[-dt,]`

Relevance of the game based on launch year

`r a$year2[a$Year_of_Release<"2010"]=0 a$year2[(a$Year_of_Release=="2010")|(a$Year_of_Release>"2000")]=1`

Removing dependent variables: The Global_sales variable is basically the addition of all the other sales variables

`r vg1 <- a[,c(2,4,5,10,11,13,16,18)] vg2 <- a[,c(6,7,8,9,10,11,12,13,14)] vg3 <- a[,c(10,11,12,13,14)]`

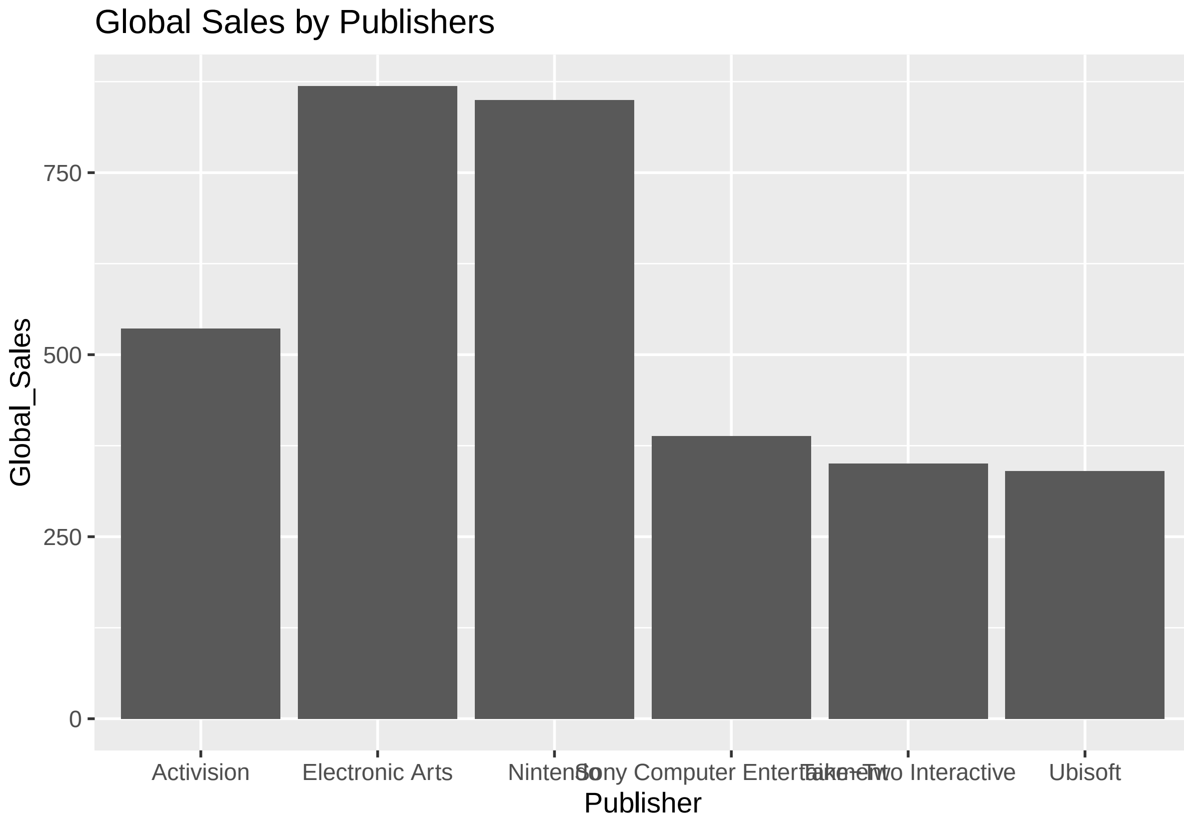
Creating Binary Variable “Hit”

“‘r vg2$Hit = 0vg$2Hit[vg2$Global_{S}ales >= mean(vg$2Global_Sales)]=1

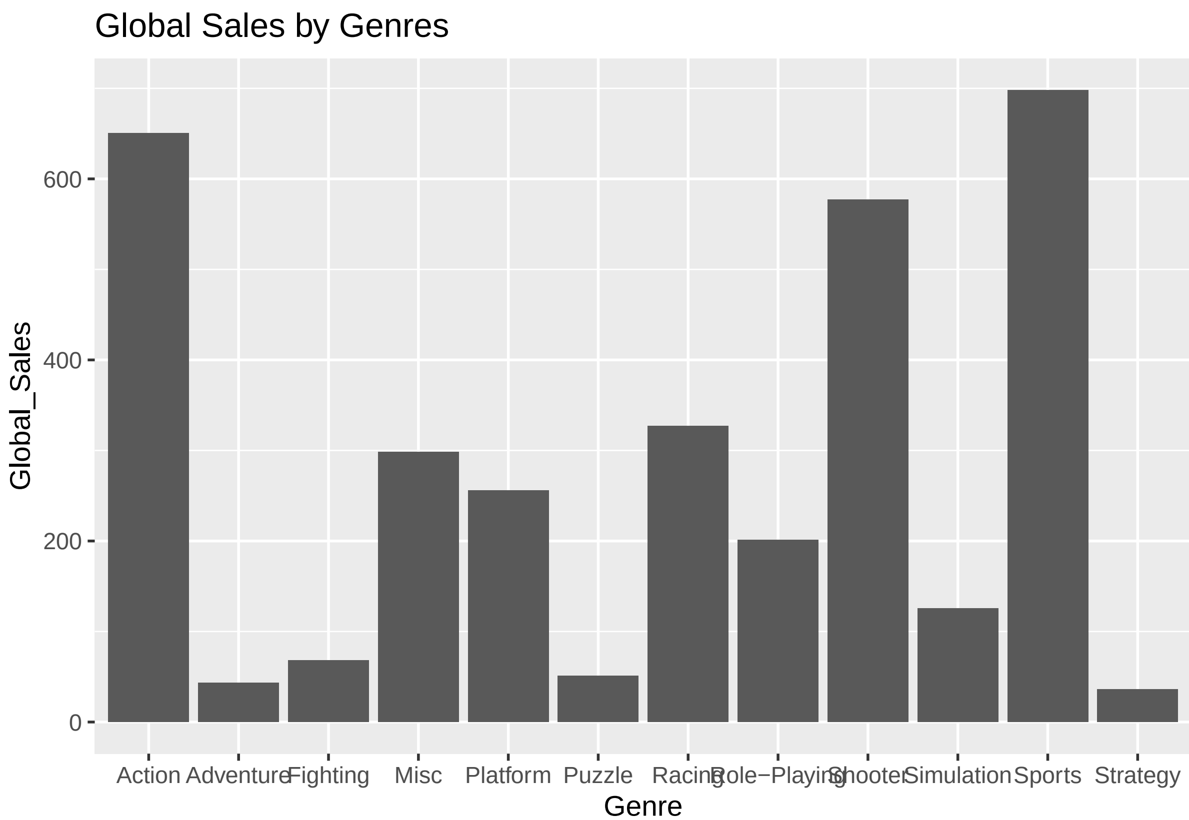
dt1 = sort(sample(nrow(vg2), nrow(vg2)*0.6)) train1 <- vg2[dt1,] test1 <- vg2[-dt1,] “‘

Data Visualization

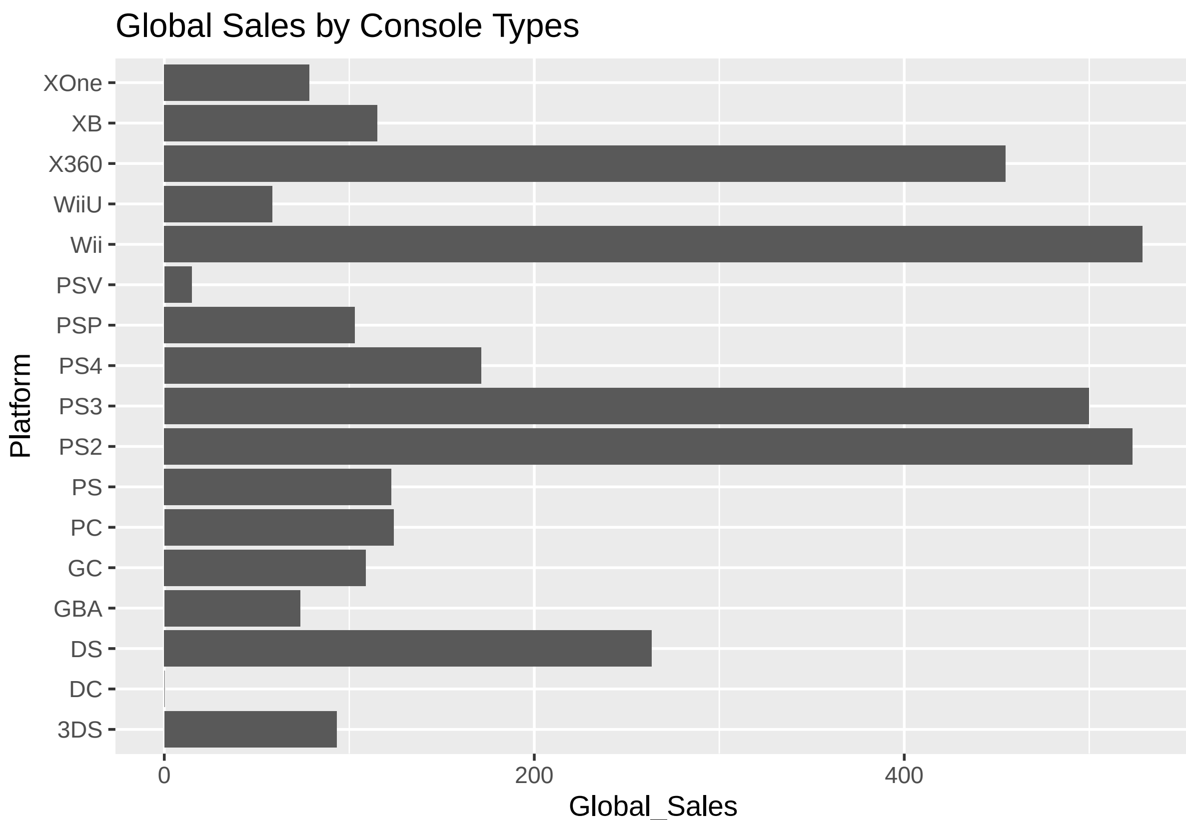
“‘r library(ggplot2, pos = .Machine$integer.max)a2 < -data.frame(Global_{S}ales = a$Global_Sales, Publisher=a$Publisher, Genre = a$Genre, Platform=a$Platform) ggplot(a2, aes(x=Publisher, y=Global_Sales)) + geom_bar(stat="identity") + ggtitle("Global Sales by Publishers") "'



r ggplot(a2, aes(x=Genre, y=Global_Sales)) + geom_bar(stat="identity") + ggtitle("Global Sales by Genres")

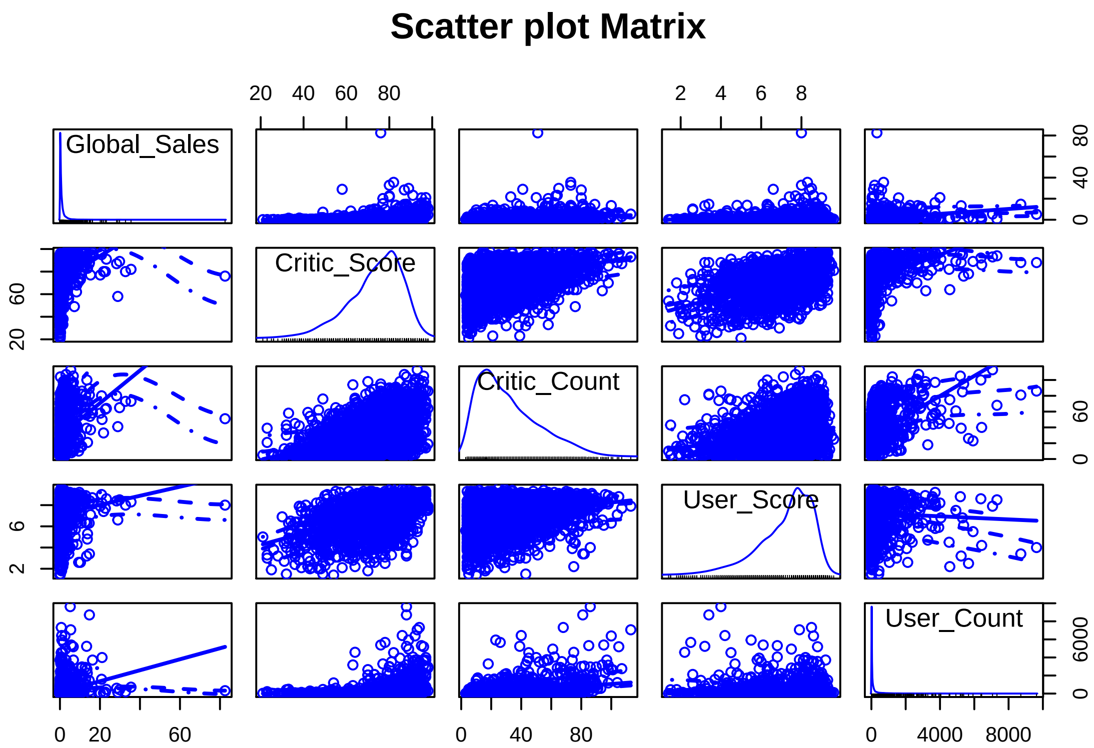


r ggplot(a2, aes(x=Global_Sales, y=Platform)) + geom_bar(stat="identity") + ggtitle("Global Sales by Console Types")



2)Data Exploration and Feature Selection

```
r library(car)
## Loading required package: carData
r scatterplotMatrix(vg3, main= "Scatter plot Matrix")
```




```
r cor(vg2)
##                 NA_Sales   EU_Sales   JP_Sales Other_Sales Global_Sales ## NA_Sales
1.00000000 0.85639181 0.54674654  0.73818901   0.96131479 ## EU_Sales     0.85639181
1.00000000 0.58414732  0.71018864   0.94514826 ## JP_Sales     0.54674654 0.58414732
1.00000000  0.41827437   0.66604007 ## Other_Sales  0.73818901 0.71018864 0.41827437
1.00000000   0.80289528 ## Global_Sales 0.96131479 0.94514826 0.66604007  0.80289528
1.00000000 ## Critic_Score 0.24330853 0.20976233 0.13528333  0.19266910   0.23651412
## Critic_Count 0.27069601 0.26112254 0.19093780  0.24071986   0.28225703 ##
User_Score   0.09324253 0.04723343 0.13263738  0.05206101   0.08698092 ## User_Count
0.24702228 0.29103659 0.07032574  0.24937455   0.26612860 ## Hit          0.48437911
0.45616532 0.31192053  0.40465005   0.49347719 ##              Critic_Score
Critic_Count  User_Score  User_Count       Hit ## NA_Sales        0.2433085
0.2706960  0.09324253  0.24702228 0.4843791 ## EU_Sales        0.2097623    0.2611225
0.04723343  0.29103659 0.4561653 ## JP_Sales         0.1352833    0.1909378  0.13263738
0.07032574 0.3119205 ## Other_Sales      0.1926691    0.2407199  0.05206101  0.24937455
0.4046501 ## Global_Sales     0.2365141    0.2822570  0.08698092  0.26612860 0.4934772
## Critic_Score     1.0000000    0.3817377  0.51858259  0.28328181 0.3474384 ##
Critic_Count     0.3817377    1.0000000  0.22440338  0.39606562 0.3572248 ## User_Score
0.5185826    0.2244034  1.00000000 -0.03386883 0.1452905 ## User_Count       0.2832818
0.3960656 -0.03386883  1.00000000 0.2673248 ## Hit             0.3474384    0.3572248
0.14529051  0.26732481 1.0000000
r cor(vg3)
```

```
##              Global_Sales Critic_Score Critic_Count  User_Score  User_Count ##
Global_Sales   1.00000000    0.2365141    0.2822570  0.08698092  0.26612860 ##
Critic_Score   0.23651412    1.0000000    0.3817377  0.51858259  0.28328181 ##
Critic_Count   0.28225703    0.3817377    1.0000000  0.22440338  0.39606562 ##
User_Score     0.08698092    0.5185826    0.2244034  1.00000000 -0.03386883 ##
User_Count     0.26612860    0.2832818    0.3960656 -0.03386883  1.00000000
```

The sale variables are all closely correlated to the Global_Sales variable.

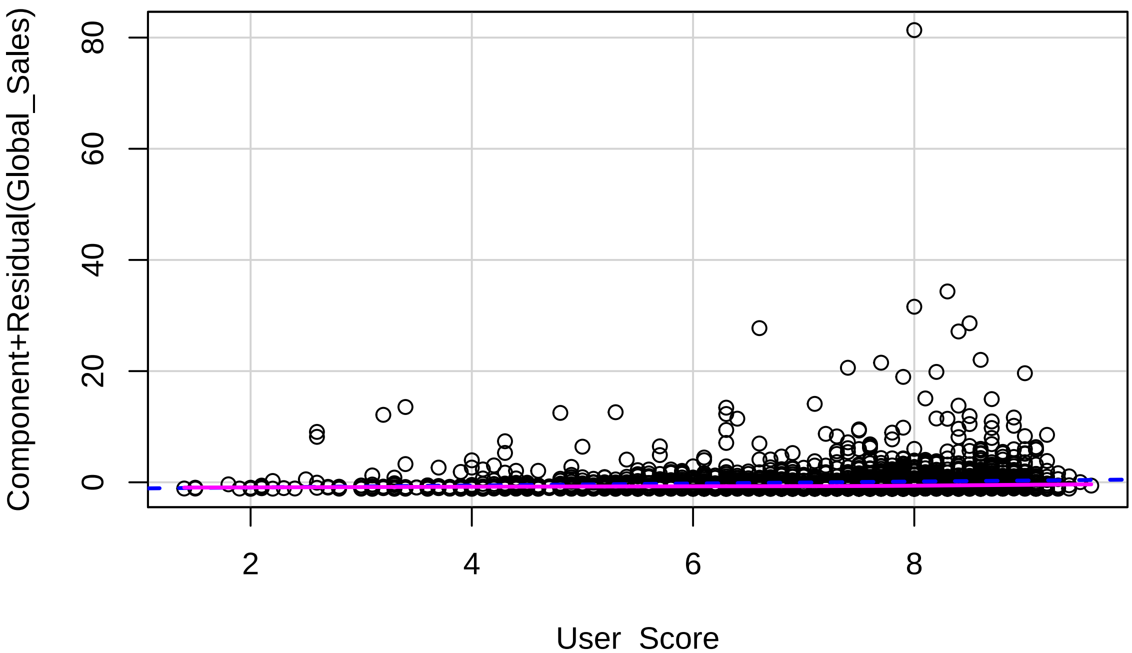
Performing Linear Regression

"'r lrm1 <- lm(Global_Sales ~ User_Score, data = vg1) lrm2 <- lm(Global_Sales ~ Critic_Score, data = vg1) lrm3 <- lm(Global_Sales ~ User_Score + Critic_Score, data = vg1)
summary(lrm1) "'

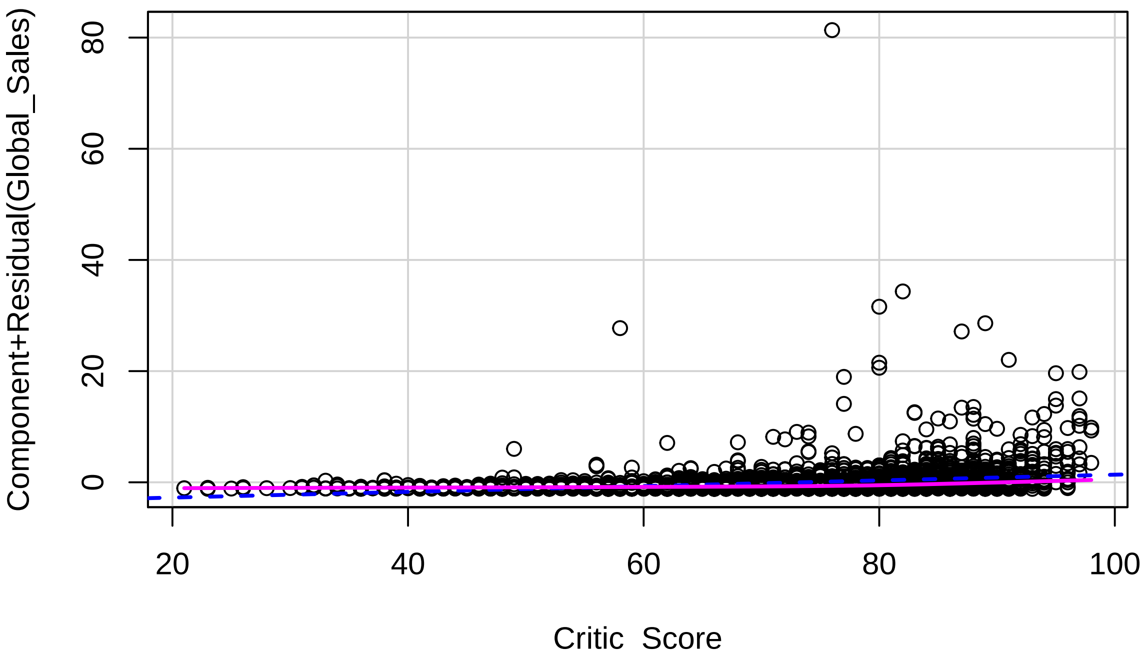
```
## ## Call: ## lm(formula = Global_Sales ~ User_Score, data = vg1) ## ## Residuals: ##
Min     1Q Median     3Q    Max ## -1.525 -0.977 -0.636 -0.027 81.207 ## ##
Coefficients: ##             Estimate Std. Error t value Pr(>|t|) ## (Intercept)
-0.09040    0.28058  -0.322    0.747 ## User_Score   0.17666    0.03817   4.628
3.85e-06 *** ## --- ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ## Residual standard error: 2.8 on 2810 degrees of freedom ## Multiple R-squared:
0.007566,   Adjusted R-squared:  0.007213 ## F-statistic: 21.42 on 1 and 2810 DF,
p-value: 3.853e-06
r summary(lrm2)
## ## Call: ## lm(formula = Global_Sales ~ Critic_Score, data = vg1) ## ## Residuals:
##    Min     1Q Median     3Q    Max ## -2.209 -0.985 -0.505  0.192 81.212 ## ##
Coefficients: ##               Estimate Std. Error t value Pr(>|t|) ## (Intercept)
-2.595806   0.297496  -8.726   <2e-16 *** ## Critic_Score  0.051503   0.003991  12.904
<2e-16 *** ## --- ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
## Residual standard error: 2.73 on 2810 degrees of freedom ## Multiple R-squared:
0.05594,    Adjusted R-squared:  0.0556 ## F-statistic: 166.5 on 1 and 2810 DF,
p-value: < 2.2e-16
r summary(lrm3)
## ## Call: ## lm(formula = Global_Sales ~ User_Score + Critic_Score, data = vg1) ##
## Residuals: ##    Min     1Q Median     3Q    Max ## -2.447 -0.978 -0.504  0.206
81.275 ## ## Coefficients: ##              Estimate Std. Error t value Pr(>|t|) ##
(Intercept)  -2.284828   0.327126  -6.985 3.55e-12 *** ## User_Score   -0.099100
0.043507  -2.278   0.0228 * ## Critic_Score  0.057013   0.004665  12.222  < 2e-16 ***
## --- ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ## ##
Residual standard error: 2.728 on 2809 degrees of freedom ## Multiple R-squared:
0.05768,    Adjusted R-squared:  0.05701 ## F-statistic: 85.97 on 2 and 2809 DF,
p-value: < 2.2e-16
```

CR PLots

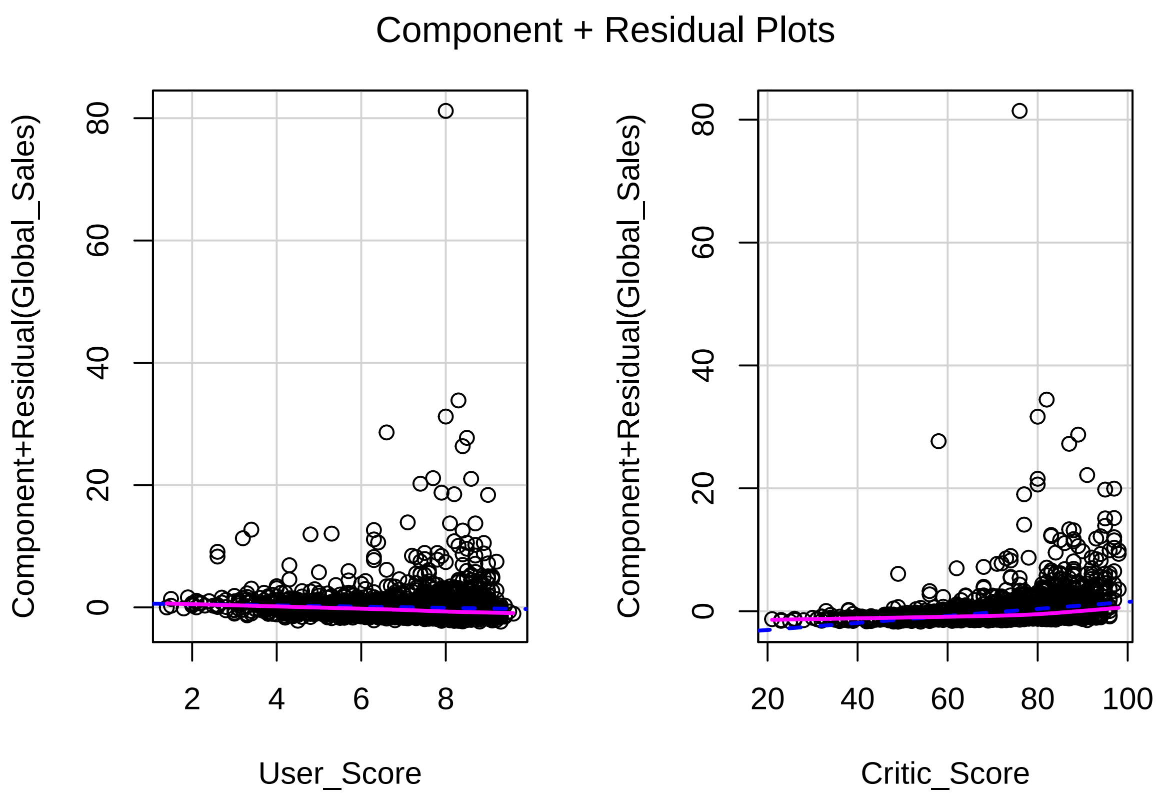
```
r library(car) crPlots(lrm1)
```



r crPlots(lrm2)

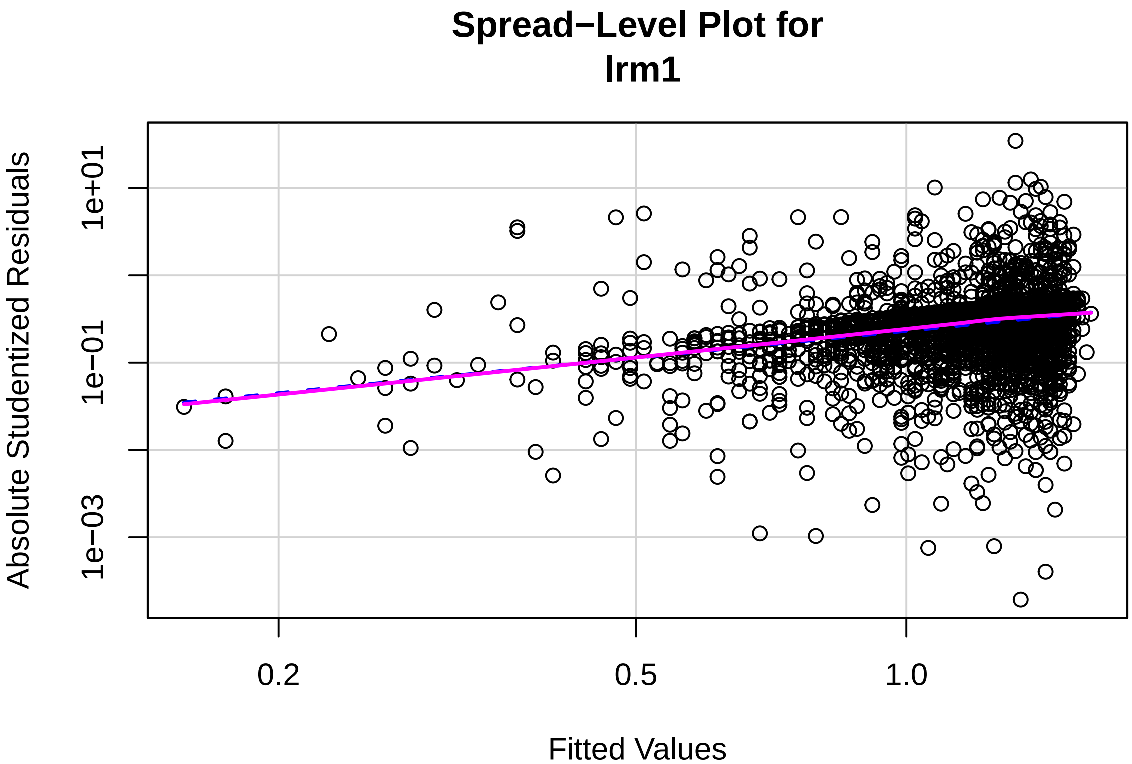


r crPlots(lrm3)

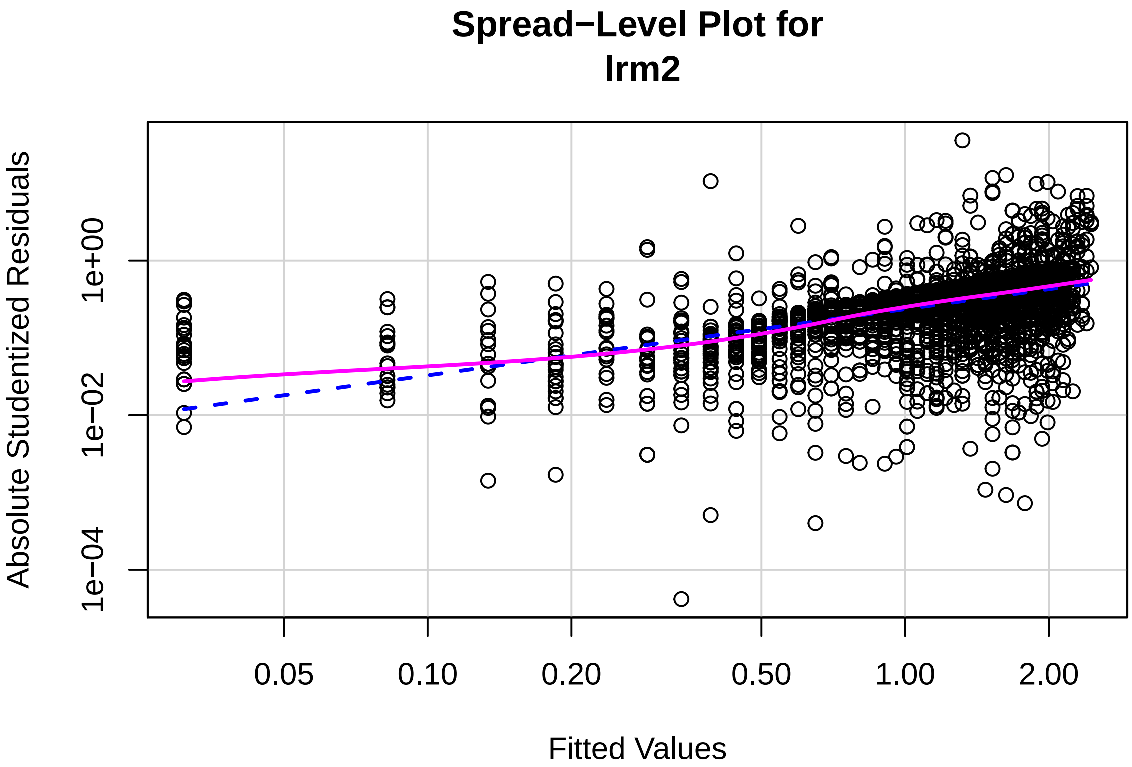


Spread Level Plot

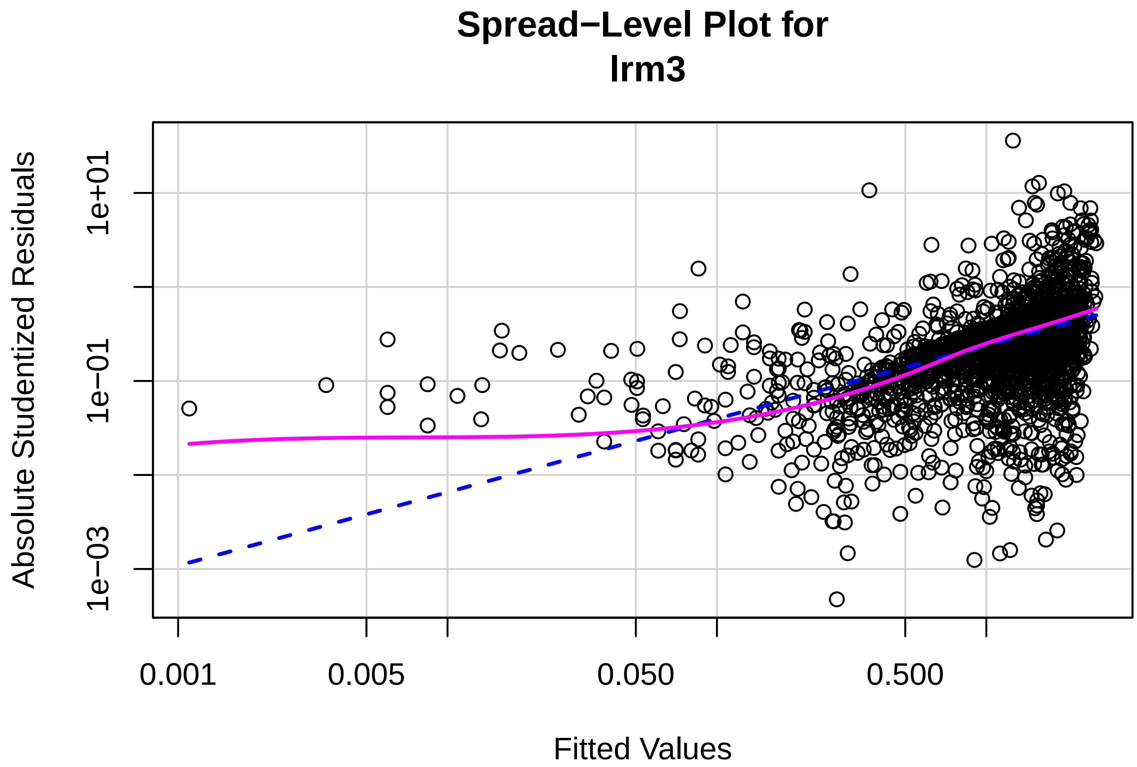
`r spreadLevelPlot(lrm1)`




```
## ## Suggested power transformation:  -0.02227186
r spreadLevelPlot(lrm2)
## Warning in spreadLevelPlot.lm(lrm2): ## 179 negative fitted values removed
```




```
## ## Suggested power transformation:  0.1432235
r spreadLevelPlot(lrm3)
## Warning in spreadLevelPlot.lm(lrm3): ## 168 negative fitted values removed
```

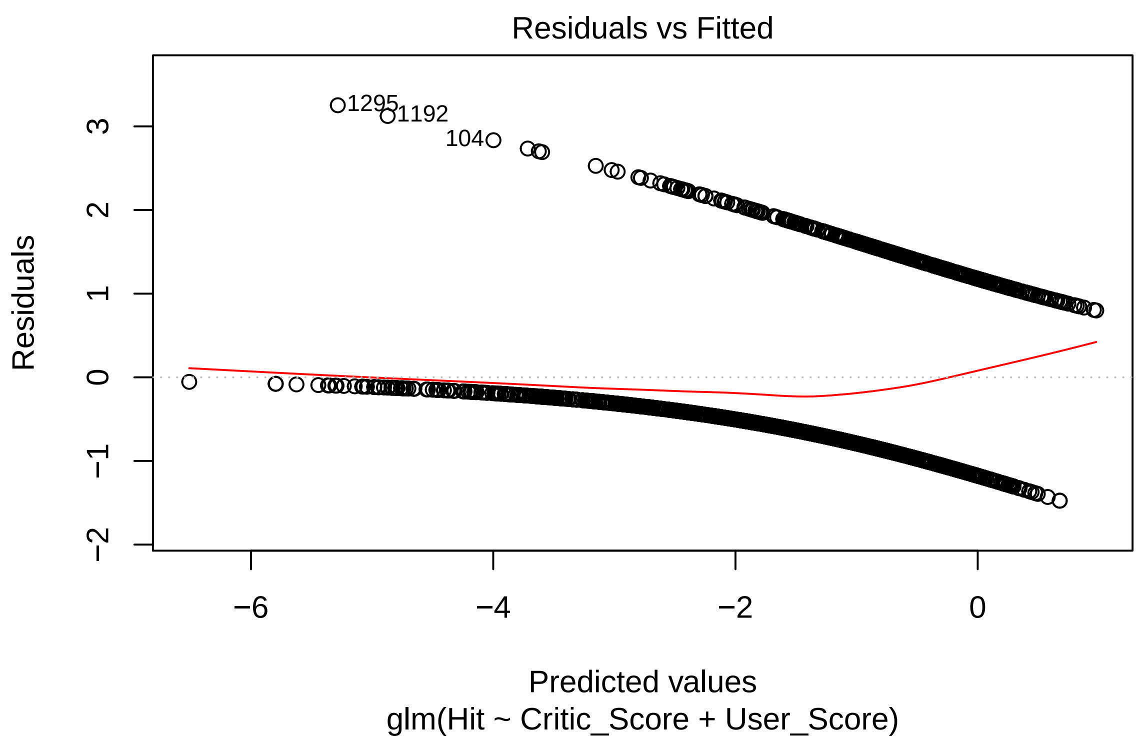


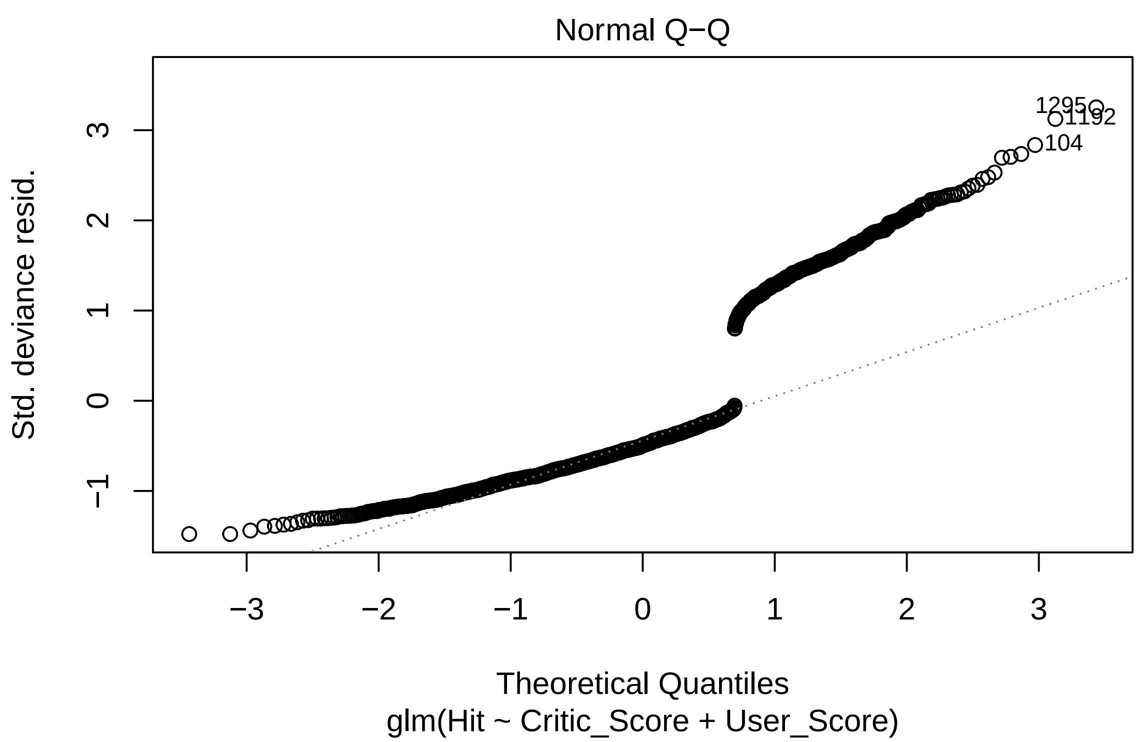

```
## ## Suggested power transformation:  0.2187696
r anova(lrm1, lrm2, lrm3)
## Analysis of Variance Table ## ## Model 1: Global_Sales ~ User_Score ## Model 2:
Global_Sales ~ Critic_Score ## Model 3: Global_Sales ~ User_Score + Critic_Score ##
Res.Df    RSS Df Sum of Sq      F  Pr(>F) ## 1    2810 22024 ## 2   2810 20950  0
1073.48 ## 3   2809 20912  1     38.62 5.1883 0.02281 * ## --- ## Signif. codes:  0
'***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
r AIC(lrm1, lrm2, lrm3)
##      df      AIC ## lrm1  3 13773.84 ## lrm2  3 13633.33 ## lrm3  4 13630.14
```

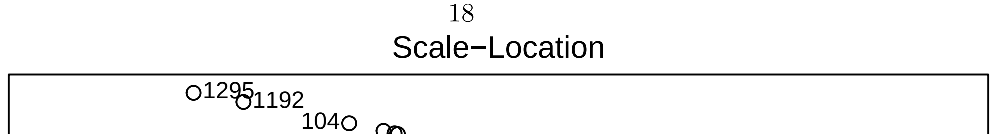
lrm3 is the better model.

Performing Classification using Logistic Regression

```
r logit1 <- glm(Hit ~ Critic_Score + User_Score, data = train1 , family = "binomial")
summary(logit1)
## ## Call: ## glm(formula = Hit ~ Critic_Score + User_Score, family = "binomial", ##
data = train1) ## ## Deviance Residuals: ##     Min       1Q   Median       3Q
Max ## -1.4752  -0.7706  -0.4932  -0.1096   3.2522 ## ## Coefficients: ##
Estimate Std. Error z value Pr(>|z|) ## (Intercept)  -8.116157   0.578709 -14.025
<2e-16 *** ## Critic_Score  0.100539   0.007633  13.172   <2e-16 *** ## User_Score
-0.101035   0.053714  -1.881     0.06 . ## --- ## Signif. codes:  0 '***' 0.001 '**'
0.01 '*' 0.05 '.' 0.1 ' ' 1 ## ## (Dispersion parameter for binomial family taken to
be 1) ## ##     Null deviance: 1871.1  on 1686  degrees of freedom ## Residual
deviance: 1601.7  on 1684  degrees of freedom ## AIC: 1607.7 ## ## Number of Fisher
Scoring iterations: 5
r plot(logit1)
```


Residuals vs Fitted
Residuals
3
2
1
0
−1
−2
1295
1192
104
−6
−4
−2
0
Predicted values
glm(Hit ~ Critic_Score + User_Score)


Normal Q−Q
Std. deviance resid.
3
2
1
0
−1
1295
1192
104
−3
−2
−1
0
1
2
3
Theoretical Quantiles
glm(Hit ~ Critic_Score + User_Score)


Scale−Location
1295
1192
104

Predicting on test set

```
r library("dplyr")
## ## Attaching package: 'dplyr'
## The following object is masked from 'package:car': ## ##     recode
## The following objects are masked from 'package:stats': ## ##     filter, lag
## The following objects are masked from 'package:base': ## ##     intersect, setdiff,
setequal, union
r pred <- predict(logit1, data = test1, type = "response") summary(pred)
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. ## 0.001486 0.101307 0.220493
0.243035 0.357186 0.726884
r #prob <- ifelse(test1$Global_Sales > mean(test1$Global_Sales), 1, 0 ) prob <-
ifelse(test1$Global_Sales > 1, 1, 0 ) table(prob ,test1$Hit)
## ## prob   0   1 ##    0 799   0 ##    1  57 269
```