Title: “Regression” Authors: Gaurang Goel (GXG190015), Anurag Diwate (AXD190004) Date: 02/17/2023

#Linear Regression in General Terms Linear regression is a statistical approach for establishing a link between one or more independent variables and a dependent variable. By minimizing the squared difference between the actual and anticipated values, it attempts to find the line that best describes the data points. The equation for the line is y = mx + b, where y is the dependent variable, x is the independent variable, and m and b are the slope and y-intercept of the line, respectively.

#Strengths and Weaknesses of Linear Regression Linear regression offers various advantages, including its simplicity and readability. It allows one to make predictions based on the premise that the variables have a linear relationship. It does, however, have several weaknesses, such as being susceptible to outliers, having multicollinearity, and having non-linear connections between variables. Furthermore, it presupposes that the model’s mistakes are normally distributed with a constant variance, which may not be the case in real-world datasets.

#Data Source https://www.kaggle.com/datasets/gregorut/videogamesales (https://www.kaggle.com/datasets/gregorut/videogamesales)

```
#importing dataset
mydata <- read.csv("e:/vgsales2.csv", na.strings = c("", "NA", "N/A"))
mydata <- na.omit(mydata)

#attaching data
attach(mydata)

#checking validity
names(mydata)
```

```
##  [1] "Rank"         "Name"         "Platform"     "Year"         "Genre"
##  [6] "Publisher"    "NA_Sales"     "EU_Sales"     "JP_Sales"     "Other_Sales"
## [11] "Global_Sales"
```

```
class(Name)
```

```
## [1] "character"
```

```
#Part A: Dividing the data into 80/20 train/test
#
set.seed(123)
#
train_idx <- sample(nrow(mydata), 0.8 * nrow(mydata))
train_data <- mydata[train_idx, ]
train_data$Year <- as.numeric(train_data$Year)
test_data <- mydata[-train_idx, ]
#print("success")

#Part B: Data Exploration
#names() method returns the names of the headers in the dataset
names(train_data)
```

```
##  [1] "Rank"         "Name"         "Platform"     "Year"         "Genre"
##  [6] "Publisher"    "NA_Sales"     "EU_Sales"     "JP_Sales"     "Other_Sales"
## [11] "Global_Sales"
```

```
#nrow() method returns the number of rows in the dataset
nrow(train_data)
```

```
## [1] 13032
```

```
#nrow() method returns the number of columns in the dataset
ncol(train_data)
```

```
## [1] 11
```

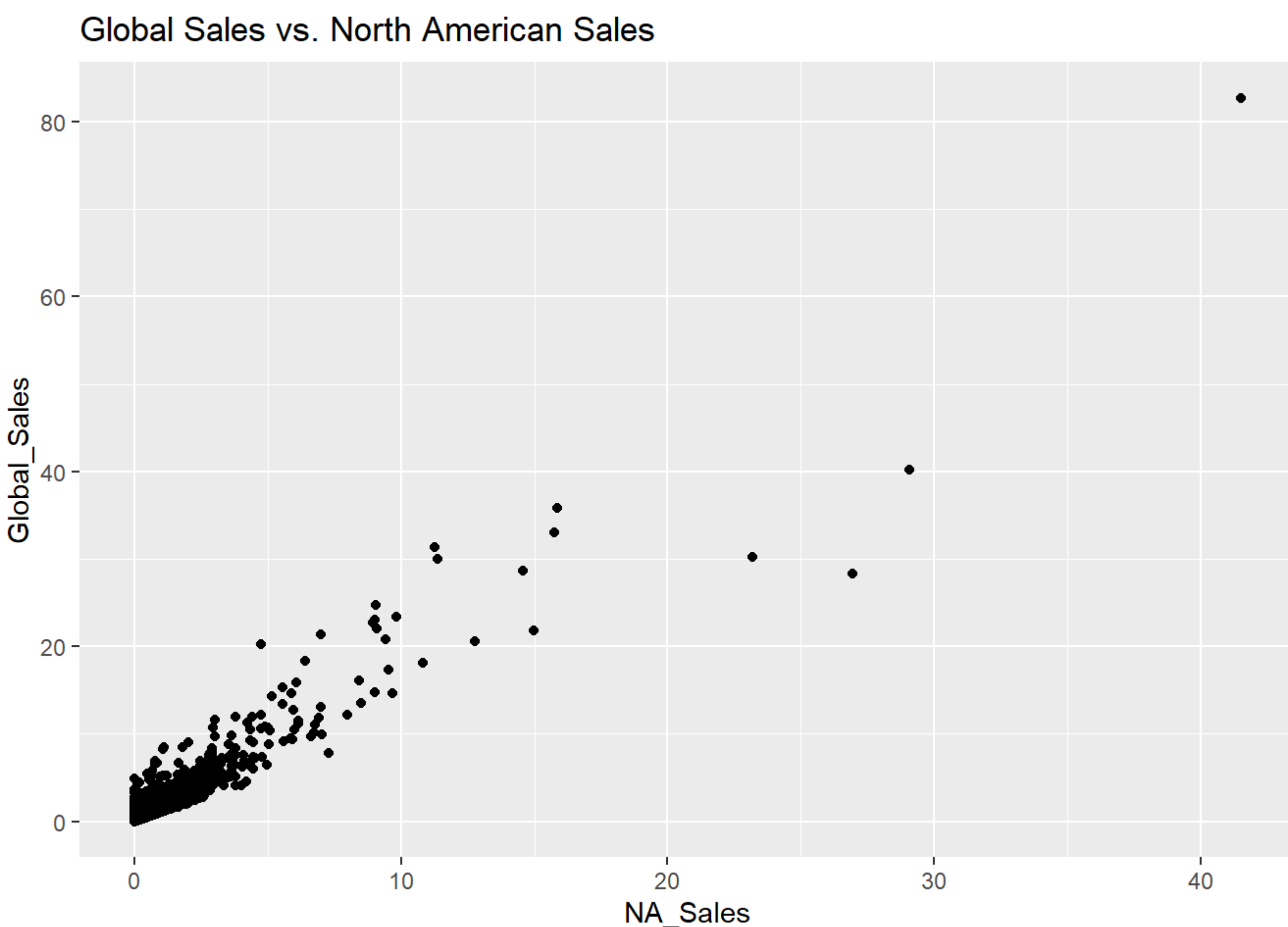
```
#The summary() method provides an overview of each variable's distribution in the dataset.
summary(train_data)
```

```
##       Rank           Name             Platform             Year     
##  Min.   :    1   Length:13032       Length:13032       Min.   :1980  
##  1st Qu.: 4122   Class :character   Class :character   1st Qu.:2003  
##  Median : 8274   Mode  :character   Mode  :character   Median :2007  
##  Mean   : 8279                                         Mean   :2006  
##  3rd Qu.:12423                                         3rd Qu.:2010  
##  Max.   :16600                                         Max.   :2020  
##     Genre            Publisher            NA_Sales          EU_Sales      
##  Length:13032       Length:13032       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   : 0.2665   Mean   : 0.1474  
##                                        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   : 0.08029   Mean   : 0.04929   Mean   : 0.5438  
##  3rd Qu.: 0.04000   3rd Qu.: 0.04000   3rd Qu.: 0.4800  
##  Max.   :10.22000   Max.   :10.57000   Max.   :82.7400  
```

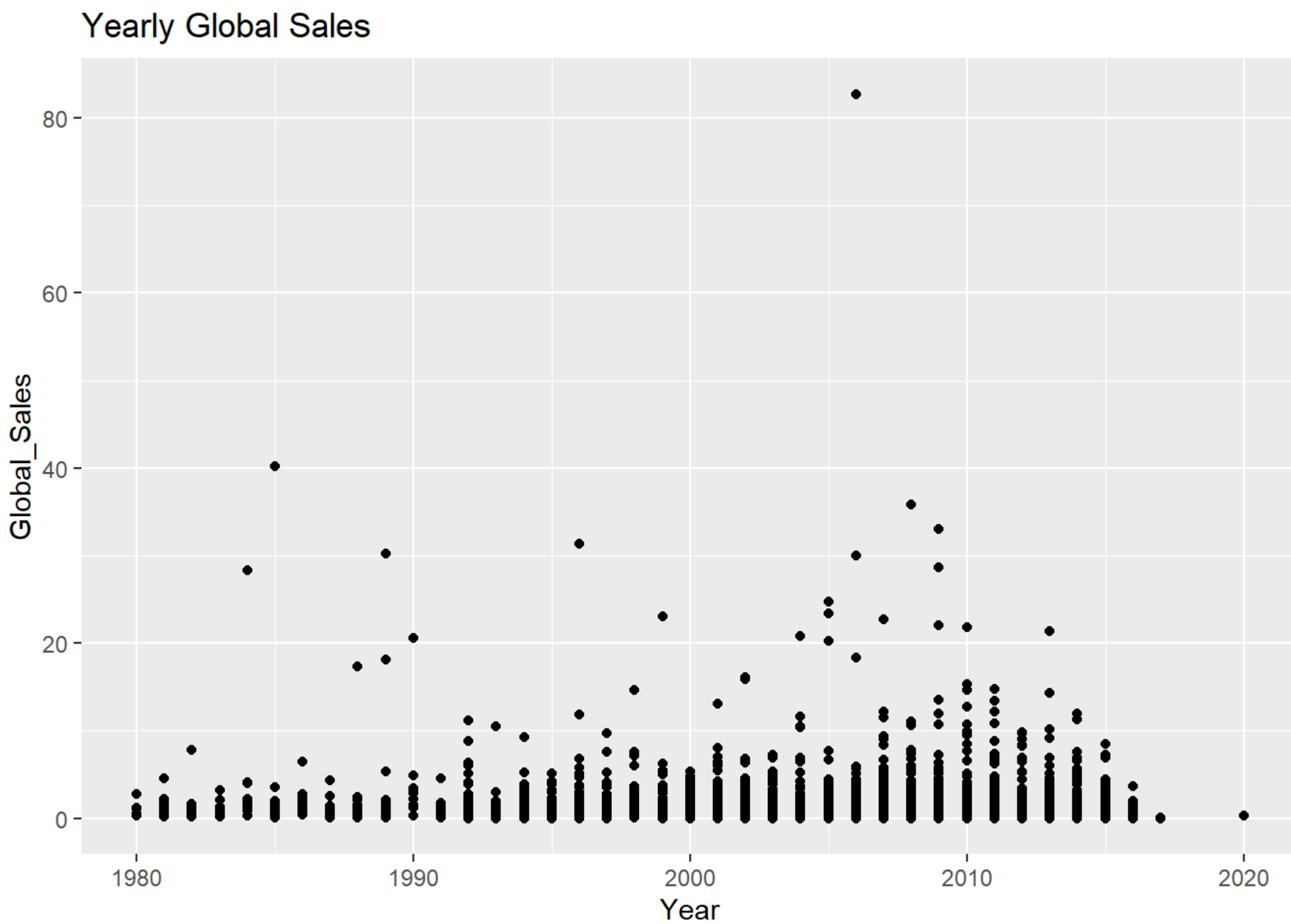
```
#The str() method displays the dataset's structure, including variable data types.
str(train_data)
```

```
## 'data.frame':    13032 obs. of  11 variables:
##  $ Rank        : int  2500 2549 10609 8878 12709 3029 1864 9504 3422 13783 ...
##  $ Name        : chr  "I Spy: Fun House" "Mega Man" "Van Helsing" "Candace Kane's Candy Facto
ry" ...
##  $ Platform    : chr  "DS" "NES" "GBA" "Wii" ...
##  $ Year        : num  2007 1987 2004 2008 2011 ...
##  $ Genre       : chr  "Puzzle" "Platform" "Action" "Action" ...
##  $ Publisher   : chr  "Scholastic Inc." "Capcom" "Activision" "Destineer" ...
##  $ NA_Sales    : num  0.77 0.45 0.07 0.14 0.05 0.16 0.92 0.11 0.13 0 ...
##  $ EU_Sales    : num  0 0.08 0.03 0 0 0.36 0.09 0 0.22 0 ...
##  $ JP_Sales    : num  0 0.27 0 0 0 0 0 0 0.01 0.04 ...
##  $ Other_Sales : num  0.06 0.01 0 0.01 0.01 0.14 0.09 0.02 0.23 0 ...
##  $ Global_Sales: num  0.83 0.81 0.1 0.15 0.06 0.67 1.1 0.13 0.59 0.04 ...
##  - attr(*, "na.action")= 'omit' Named int [1:307] 180 378 432 471 608 625 650 653 712 783 ...
##   ..- attr(*, "names")= chr [1:307] "180" "378" "432" "471" ...
```

```
#Part C: Graphs
#The first graph depicts the link between worldwide and North American sales, while the second d
epicts the evolution of global sales over time.
library(ggplot2)
ggplot(train_data, aes(x = NA_Sales, y = Global_Sales)) +
  geom_point() +
  ggtitle("Global Sales vs. North American Sales")
```



```
ggplot(train_data, aes(x = Year, y = Global_Sales)) +
  geom_point() +
  ggtitle("Yearly Global Sales")
```

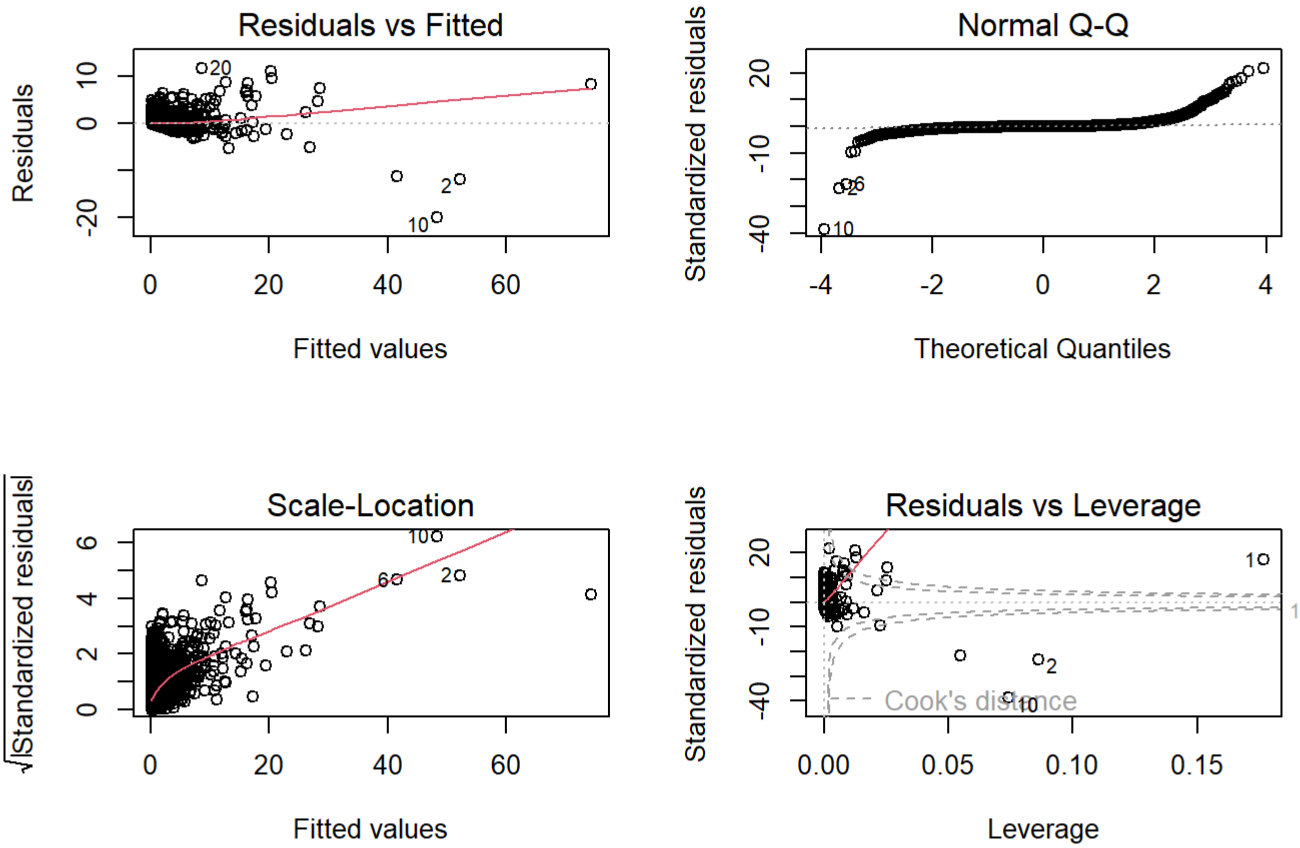



```
#Part D: Linear Regression
#Simple linear regression model using NA_Sales as the predictor variable and Global_Sales as the response variable
model <- lm(Global_Sales ~ NA_Sales, data = train_data)
summary(model)
```

```
##
## Call:
## lm(formula = Global_Sales ~ NA_Sales, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0254  -0.1095  -0.0554   0.0184  11.6400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.066140   0.004935    13.4   <2e-16 ***
## NA_Sales    1.792398   0.005486   326.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5381 on 13030 degrees of freedom
## Multiple R-squared:  0.8912, Adjusted R-squared:  0.8912
## F-statistic: 1.068e+05 on 1 and 13030 DF,  p-value: < 2.2e-16
```

```
#Explanation: The intercept and slope coefficients indicate the regression line's intercept and
 slope, respectively. The Std. Error column displays the standard errors of the coefficients, wh
ilst the t value and Pr(>|t|) columns display the coefficients' t-value and p-value. The Multipl
e R-squared value measures how well the model fits the data, whereas the Residual standard error
measures how variable the response variable is around the regression line.

#Part E: Residual Plot
par(mfrow = c(2, 2))
plot(model)
```



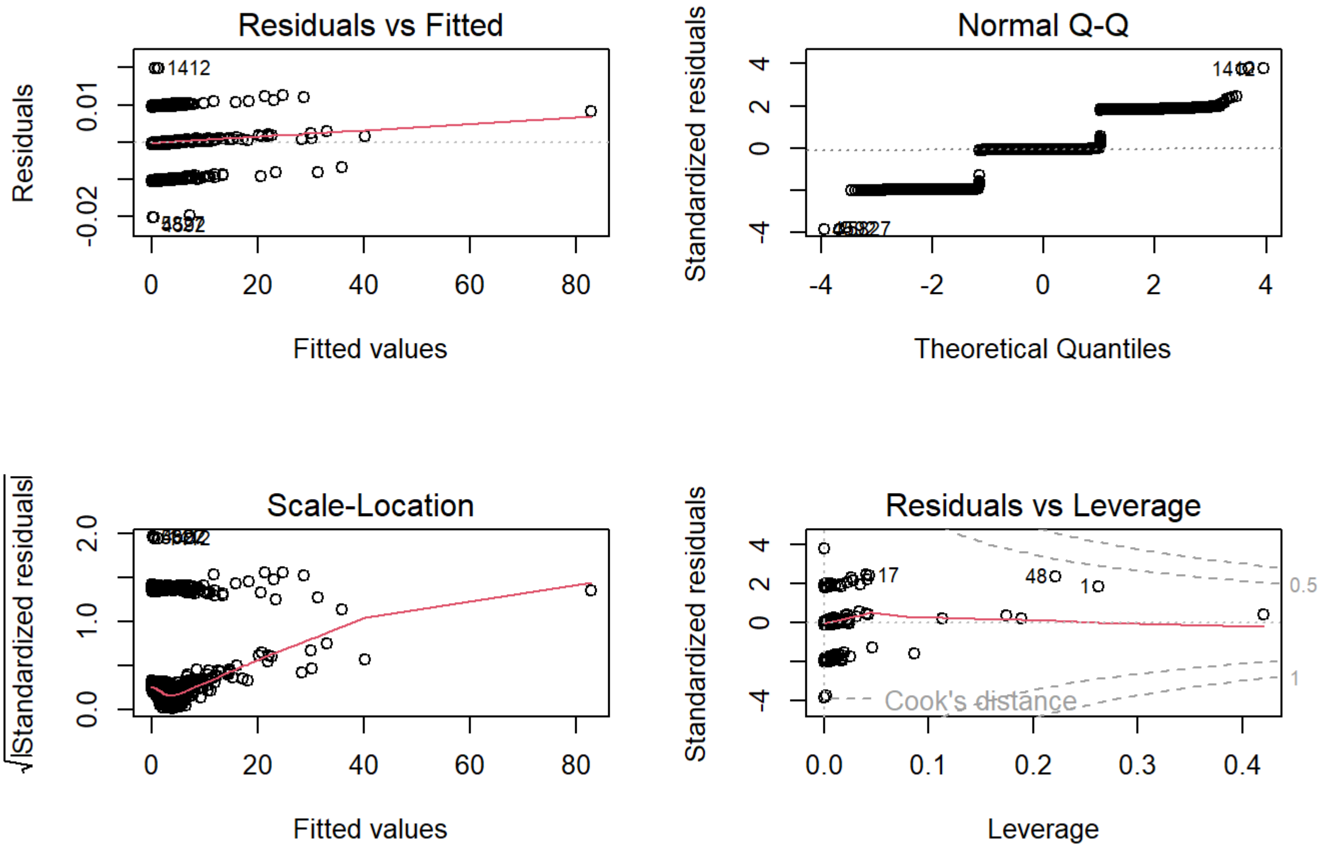

```
#Explanation: The plot() function generates a four-panel plot of the residuals that includes a h
istogram, a regular Q-Q plot, a scale-location plot, and a residuals-vs-leverage plot. The distr
ibution of the residuals appears to be generally normal, based on the residual plots, with the e
xception of some tiny deviations in the tails of the histogram and the Q-Q plot. The scale-locat
ion plot indicates that the residuals are equally distributed over the range of the predictor va
riable, indicating that the variance of the residuals is constant. The residuals-vs-leverage plo
t reveals that there are no significant data items that have a major influence on the regression
line.

#Part F: Multiple Regression Model
#In this case, Global sales is the responsive variable, and all the others are predictor variabl
es.
model2 <- lm(Global_Sales ~ NA_Sales + EU_Sales + JP_Sales + Other_Sales + Year, data = train_da
ta)
summary(model2)
```

```
##
## Call:
## lm(formula = Global_Sales ~ NA_Sales + EU_Sales + JP_Sales +
##     Other_Sales + Year, data = train_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0202976 -0.0003811 -0.0003288 -0.0002689  0.0198397
##
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept)  1.664e-02  1.634e-02     1.019    0.308
## NA_Sales     1.000e+00  8.892e-05 11246.002   <2e-16 ***
## EU_Sales     9.998e-01  1.607e-04  6220.885   <2e-16 ***
## JP_Sales     1.000e+00  1.673e-04  5977.740   <2e-16 ***
## Other_Sales  9.999e-01  3.309e-04  3021.492   <2e-16 ***
## Year        -8.117e-06  8.142e-06    -0.997    0.319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005253 on 13026 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.513e+08 on 5 and 13026 DF,  p-value: < 2.2e-16
```

```
#The coefficients and their interpretation are comparable to that of the simple linear regressio
n model. The adjusted R-squared value is greater, indicating that the multiple linear regression
model explains more of the response variable variance than the basic linear regression model.

#residual plot
par(mfrow = c(2, 2))
plot(model2)
```

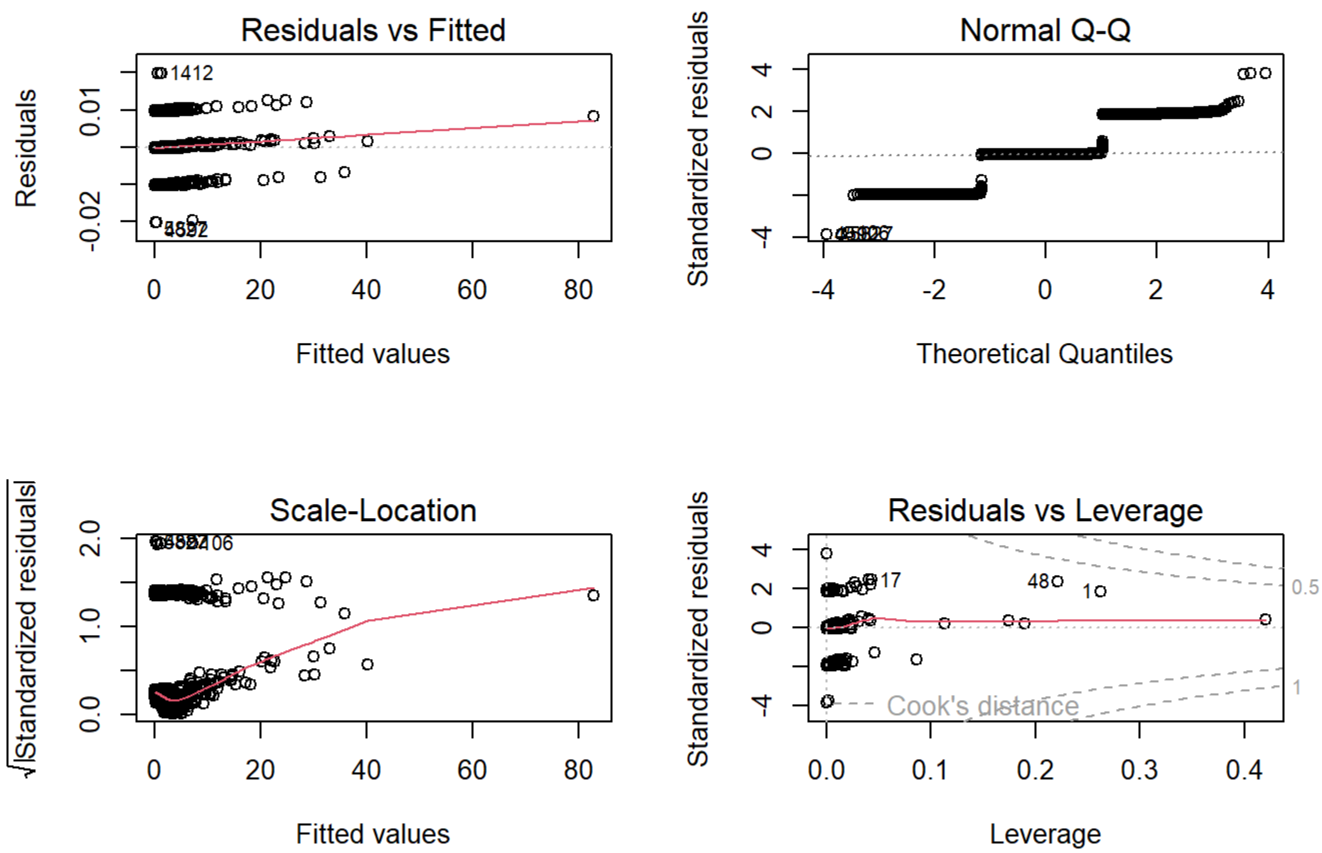



```
#Part G: Linear Regression model using a different combination of predictors
#using NA_Sales, EU_Sales, JP_Sales, Other_Sales, Year, and a quadratic term of Year as the pred
ictor variables and Global_Sales as the response variable.
train_data$Year2 <- (train_data$Year)^2
test_data$Year2 <- (test_data$Year)^2
model3 <- lm(Global_Sales ~ NA_Sales + EU_Sales + JP_Sales + Other_Sales + Year + Year2, data =
 train_data)
summary(model3)
```

```
##
## Call:
## lm(formula = Global_Sales ~ NA_Sales + EU_Sales + JP_Sales +
##     Other_Sales + Year + Year2, data = train_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0203145 -0.0003985 -0.0003371 -0.0002337  0.0198505
##
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -2.443e+00  3.300e+00    -0.740    0.459
## NA_Sales     1.000e+00  8.911e-05 11221.912   <2e-16 ***
## EU_Sales     9.998e-01  1.610e-04  6211.083   <2e-16 ***
## JP_Sales     1.000e+00  1.683e-04  5943.090   <2e-16 ***
## Other_Sales  9.998e-01  3.311e-04  3020.116   <2e-16 ***
## Year         2.447e-03  3.294e-03     0.743    0.458
## Year2       -6.126e-07  8.219e-07    -0.745    0.456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005254 on 13025 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.094e+08 on 6 and 13025 DF,  p-value: < 2.2e-16
```

```
#The results and their interpretation are identical to those of the multiple linear regression m
odel, with the exception of an additional coefficient for the quadratic component of Year. The a
djusted R-squared value is somewhat higher than that of the multiple linear regression model, im
plying that the third linear regression model explains slightly more variance in the response va
riable.

#residual plot
par(mfrow = c(2, 2))
plot(model3)
```


Residuals vs Fitted
Residuals
Fitted values
1412
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
17
48
1
0.5
1
Cook's distance

```
#Part H: Results
#The adjusted R-squared values of all three linear regression models are high, indicating that t
hey all explain a considerable part of the variation in the Global Sales variable. Nonetheless,
 the adjusted R-squared values for the multiple linear regression model and the third linear reg
ression model are both greater than those for the basic linear regression model.
#We think that the third linear regression model is better because it has the greatest adjusted
 R-squared value of the three models, implying that it accounts for the most variance in the res
ponse variable.

#Part I: Evaluation
Year <- as.numeric(Year)
# Simple Linear Regression Model
pred1 <- predict(model, newdata = test_data)
cor1 <- cor(test_data$Global_Sales, pred1)
mse1 <- mean((test_data$Global_Sales - pred1)^2)

# Multiple Linear Regression Model
test_data$Year <- as.numeric(test_data$Year)
pred2 <- predict(model2, newdata = test_data)
cor2 <- cor(test_data$Global_Sales, pred2)
mse2 <- mean((test_data$Global_Sales - pred2)^2)

# Third Linear Regression Model
pred3 <- predict(model3, newdata = test_data)
cor3 <- cor(test_data$Global_Sales, pred3)
mse3 <- mean((test_data$Global_Sales - pred3)^2)

# Output
cat("Simple Linear Regression Model:\n")
```

```
## Simple Linear Regression Model:
```

```
cat("Correlation:", cor1, "\n")
```

```
## Correlation: 0.9230345
```

```
cat("MSE:", mse1, "\n\n")
```

```
## MSE: 0.2423783
```

```
cat("Multiple Linear Regression Model:\n")
```

```
## Multiple Linear Regression Model:
```

```
cat("Correlation:", cor2, "\n")
```

```
## Correlation: 0.999992
```

```
cat("MSE:", mse2, "\n\n")
```

```
## MSE: 2.607319e-05
```

```
cat("Third Linear Regression Model:\n")
```

```
## Third Linear Regression Model:
```

```
cat("Correlation:", cor3, "\n")
```

```
## Correlation: 0.9999921
```

```
cat("MSE:", mse3, "\n\n")
```

```
## MSE: 2.606705e-05
```

```
#We investigated Kaggle's Video Game Sales dataset, which provides data on video game sales in v
arious areas. We explored and cleaned the data, divided it into training and test sets, and deve
loped three linear regression models to predict the Global Sales variable.
#The first linear regression model we created was a basic one with a single predictor, the NA Sa
les variable. The second linear regression model was a multiple linear regression model that inc
luded predictors such as NA Sales, EU Sales, JP Sales, and Other Sales. A polynomial regression
 model with interaction terms between predictor variables was the third linear regression model.
#We used correlation and mean squared error (MSE) measures to assess the performance of our mode
ls on the test data. The third linear regression model performed the best of the three, with the
highest correlation and lowest MSE values.
#Overall, our findings suggest that video game sales in different areas are highly connected, an
d that combining diverse sales data might assist to better estimate worldwide sales. Game creato
rs and publishers may use our best performing model to produce more accurate sales estimates and
guide business choices.
```