Final Project

04/10/2024

```r
library(readxl)
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```r
library(knitr)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```r
data <- read_excel("C:/Users/Suma2/Downloads/Applied stats_projrct/new jwrsey.xlsx")


#data <- read_excel("C:/Users/Owner/Desktop/Applied Statistics/Project/TaxRevenue-NJ.xlsx")
```

```r
head(data)
```

```
## # A tibble: 6 x 16
##   State  Year Quarter StateRevenue AvgTaxRate TaxRateRank AvgTaxRateOnWages
##   <chr> <dbl>   <dbl>        <dbl>      <dbl>       <dbl>             <dbl>
## 1 NJ     1997       1      1434997       1.12          12              2.42
## 2 NJ     1997       2      1281554       1.17          11              2.55
## 3 NJ     1997       3      1366904       1.16          11              2.54
## 4 NJ     1997       4      1422519       1.15          10              2.54
## 5 NJ     1998       1      1425682       1.05          12              2.32
## 6 NJ     1998       2      1374767       0.98          13              2.18
## # i 9 more variables: AvgTaxRateOnWagesRank <dbl>, MinTaxWage <dbl>,
## #   TrustFund <dbl>, TFPerWages <dbl>, TFWagesRank <dbl>, Interest <dbl>,
## #   HighCostMultiple <dbl>, AvgHCM <dbl>, AvgHCMRank <dbl>
```

Description of Variables:

- **State:** Represents the state where the data is recorded.
- **Year:** Indicates the calendar year for the data.
- **Quarter:** Represents the quarter (1, 2, 3, or 4) of the year in which the data is recorded.
- **StateRevenue:** Reflects the state revenue over the past 12 months.
- **AvgTaxRate:** Represents the average tax rate over the past 12 months, expressed as a percentage.
- **TaxRateRank:** Indicates the rank of the average tax rate over the past 12 months among other states.
- **AvgTaxRateOnWages:** Reflects the average tax rate on taxable wages over the past 12 months, expressed as a percentage.
- **AvgTaxRateOnWagesRank:** Indicates the rank of the average tax rate on taxable wages over the past 12 months among other states.
- **MinTaxWage:** Represents the taxable wage base, which is the maximum amount of earnings subject to a particular tax.
- **TrustFund:** Reflects the balance in the trust fund.
- **TFPerWages:** Indicates the trust fund balance as a percentage of total wages.
- **TFWagesRank:** Indicates the rank of the trust fund balance among other states based on total wages.
- **Interest:** Represents the interest earned on the trust fund.
- **HighCostMultiple:** Reflects the high cost multiple.
- **AvgHCM:** Represents the average high cost multiple ACHM.
- **AvgHCMRank:** Indicates the rank of the average high cost multiple ACHM among other states.

```
any(is.na(data))
```

```
## [1] TRUE
```

```
final_data <- na.omit(data)
```

```
final_data <- final_data[, -1]
```

Exploratory Data Analysis:

```
names(final_data)
```

```
##  [1] "Year"                  "Quarter"               "StateRevenue"
##  [4] "AvgTaxRate"            "TaxRateRank"           "AvgTaxRateOnWages"
##  [7] "AvgTaxRateOnWagesRank" "MinTaxWage"            "TrustFund"
## [10] "TFPerWages"            "TFWagesRank"           "Interest"
## [13] "HighCostMultiple"      "AvgHCM"                "AvgHCMRank"
```

```r
kable(summary(final_data))
```

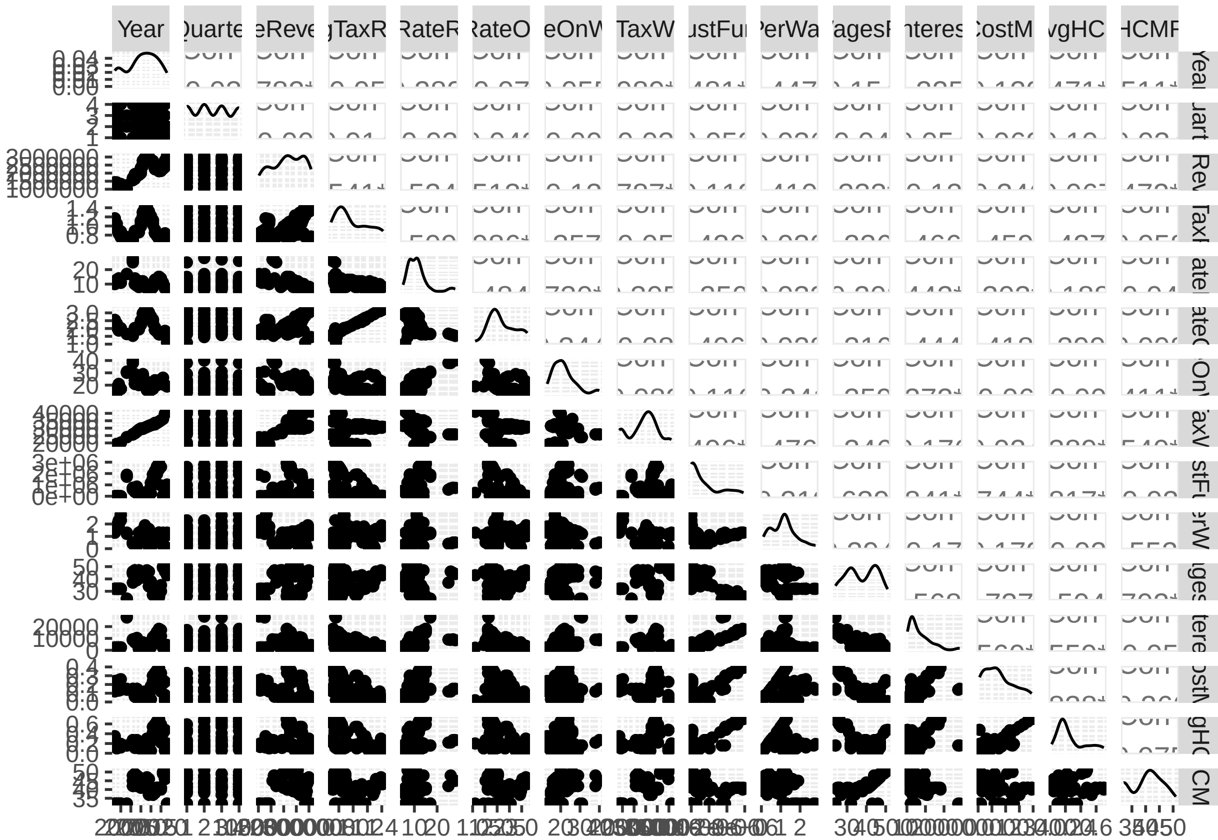
Year	Quarter	StateRevenue	AvgTaxRate	TaxRateRank	ARTaxRate	AvgTaxRateOnWages	MaxTaxWage	TaxWageRank	RTFPerWFW	WagesIntRank	InterestRank	HighCostMultiple	AvgHCM	AvgHCMRank
Min. :1997	Min. :1.000	Min. :1019738	Min. :0.6900	Min. : 5.00	Min. :1.100	Min. :13.00	Min. :18600	Min. : 1237	Min. :0.100	Min. :24.00	Min. : 303	Min. :0.0100	Min. :0.0200	Min. :32.00
1st Qu.:2006	1st Qu.:1.750	1st Qu.:1632217	1st Qu.:0.8075	1st Qu.: 8.00	1st Qu.:1.845	1st Qu.:17.00	1st Qu.:25800	1st Qu.: 32521	1st Qu.:0.575	1st Qu.:32.00	1st Qu.: 2355	1st Qu.:0.0600	1st Qu.:0.1400	1st Qu.:37.00
Median :2012	Median :2.000	Median :2232901	Median :0.8900	Median :10.50	Median :2.025	Median :20.50	Median :30000	Median : 284938	Median :1.150	Median :37.50	Median : 4199	Median :0.1400	Median :0.1900	Median :41.00
Mean :2011	Mean :2.476	Mean :2149466	Mean :0.9794	Mean :11.04	Mean :2.206	Mean :21.36	Mean :29148	Mean : 701143	Mean :1.112	Mean :37.42	Mean : 6518	Mean :0.1507	Mean :0.2632	Mean :40.45
3rd Qu.:2017	3rd Qu.:3.000	3rd Qu.:2753066	3rd Qu.:1.1600	3rd Qu.:12.00	3rd Qu.:2.558	3rd Qu.:24.00	3rd Qu.:32600	3rd Qu.: 996269	3rd Qu.:1.325	3rd Qu.:45.00	3rd Qu.: 9064	3rd Qu.:0.2325	3rd Qu.:0.3275	3rd Qu.:45.00
Max. :2023	Max. :4.000	Max. :3086141	Max. :1.4200	Max. :28.00	Max. :3.230	Max. :40.00	Max. :41100	Max. :2931281	Max. :2.800	Max. :51.00	Max. :28444	Max. :0.3900	Max. :0.7000	Max. :51.00

```r
dim(final_data)
```

```
## [1] 84 15
```

- The dimension of the dataset is 85*15

```r
ggpairs(final_data)
```



```r
idx<-which(final_data$TaxRateRank > 20)
idx
```

```
## [1] 19 20 21 22
```

```r
final_data$Year[idx]
```

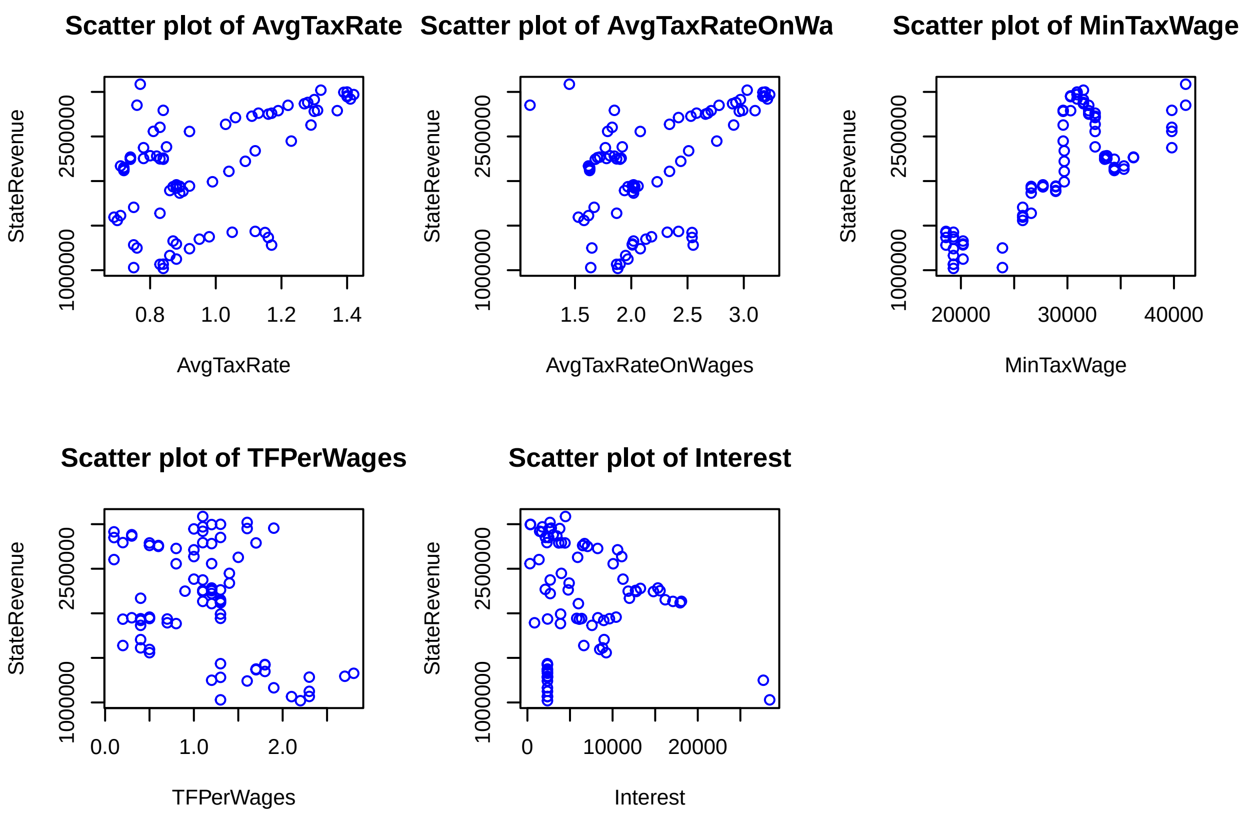
```
## [1] 2006 2006 2006 2006
```

```r
skewness_df <- data.frame(Skewness = sapply(final_data, skewness))
skewness_df
```

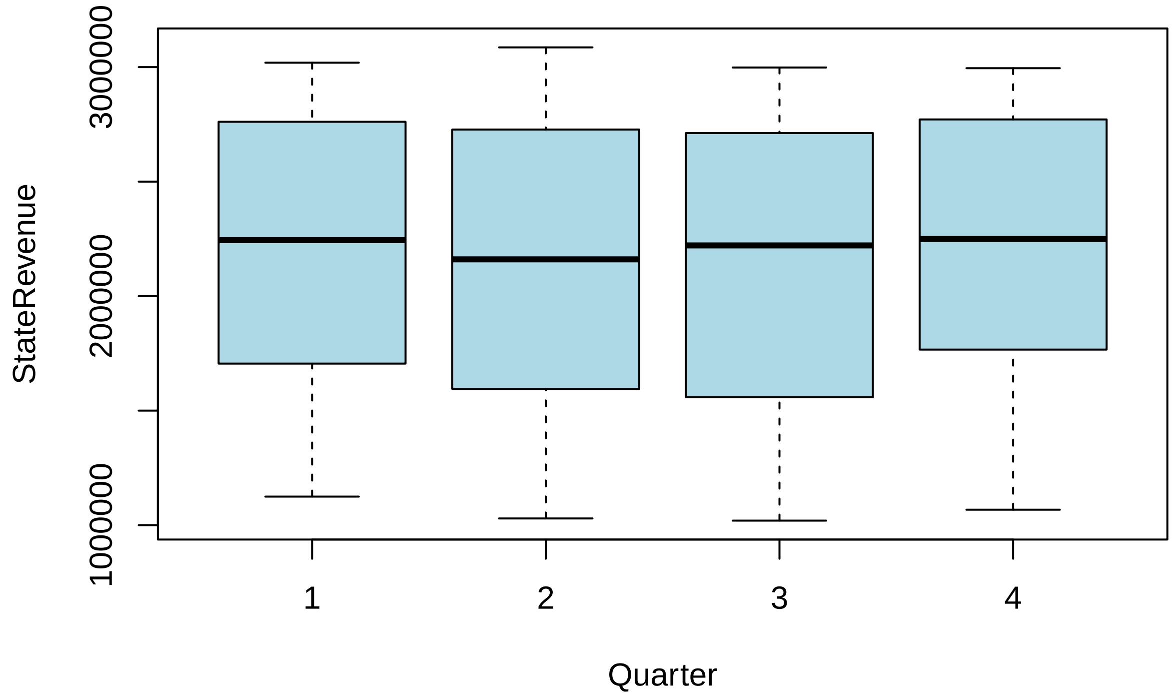
```
##                          Skewness
## Year                  -0.32471706
## Quarter                0.03321954
## StateRevenue          -0.25330234
## AvgTaxRate             0.60440272
## TaxRateRank            2.02511992
## AvgTaxRateOnWages      0.50782532
## AvgTaxRateOnWagesRank  1.30941179
## MinTaxWage            -0.26671923
## TrustFund              1.15730964
## TFPerWages             0.42454353
## TFWagesRank           -0.08625748
```

```
## Interest                   1.61777259
## HighCostMultiple           0.59931801
## AvgHCM                     1.01674822
## AvgHCMRank                -0.23525012
```

```r
feature_names <- c("AvgTaxRate", "AvgTaxRateOnWages", "MinTaxWage", "TFPerWages", "Interest")

par(mfrow = c(2, 3))

for (feature in feature_names) {
  # Create scatter plot with frequencies on y-axis
  plot(final_data[["StateRevenue"]]~final_data[[feature]],
       main = paste("Scatter plot of", feature),
       xlab = feature, ylab = "StateRevenue", col = "blue")
}
#boxplot(StateRevenue~Quarter,data = final_data)
par(mfrow = c(1, 1))
```

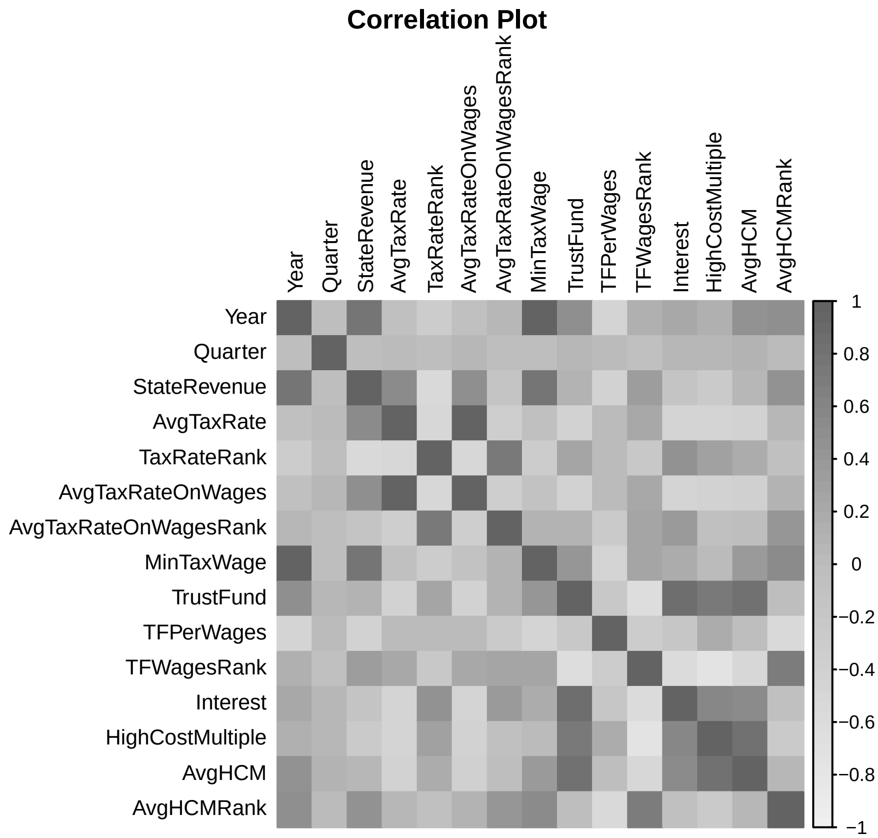



```r
boxplot(StateRevenue~Quarter,data = final_data, col = "lightblue", border = "black")
```

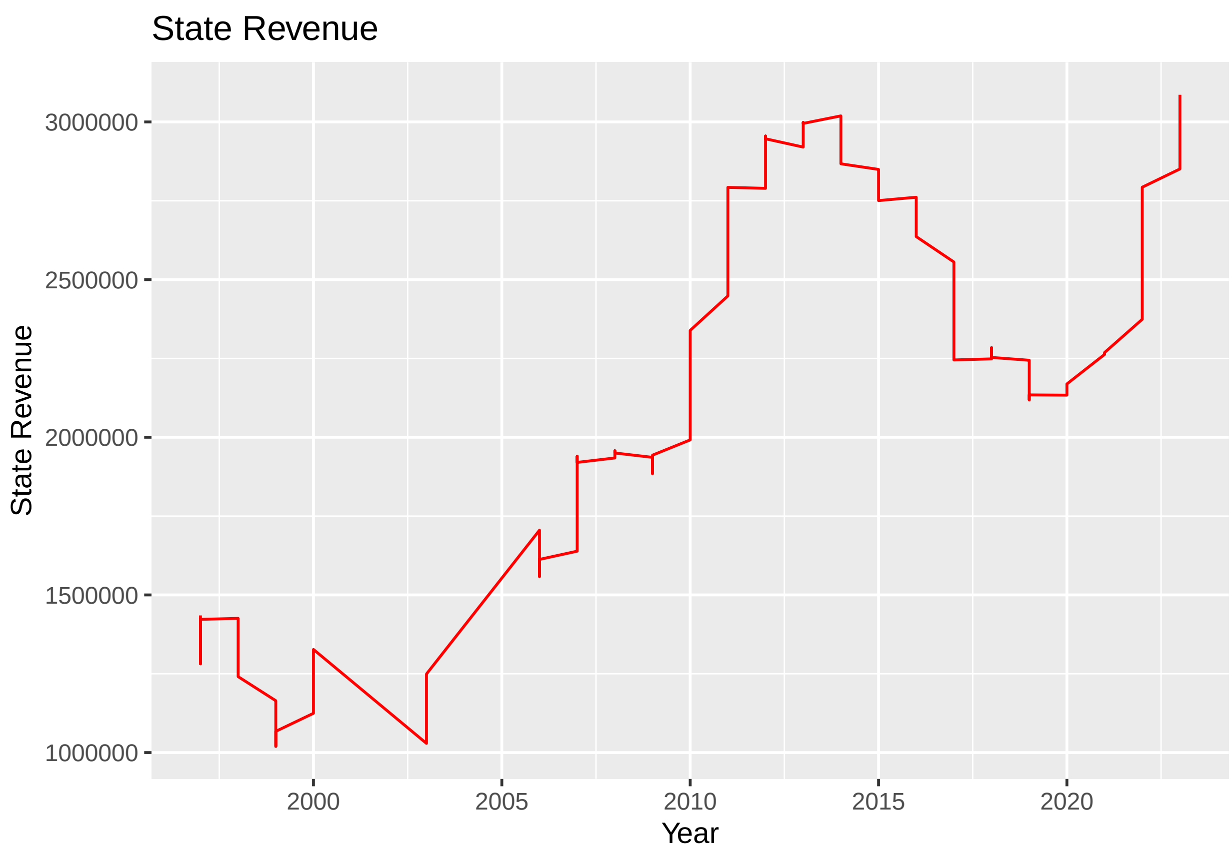



```
# Reset the layout to default
```

```
par(mfrow = c(1, 1), cex = 0.7)

cor_matrix <- cor(final_data)
my_colors <- colorRampPalette(c("#f0f0f0", "#bdbdbd", "#636363"))(50)
corrplot(cor_matrix, method = "color", col = my_colors, tl.col = "black", title = "Correlation Plot", ma
```




```r
ggplot(final_data, aes(x = Year)) +
  geom_line(aes(y = StateRevenue), color = "red", linetype = "solid") +
  ggtitle("State Revenue") +
  ylab("State Revenue") +
  xlab("Year")
```




```
set.seed(123)

# Define the proportion of the data you want in the training set
train_proportion <- 0.7

# Generate random indices for the training set
train_indices <- sample(1:nrow(final_data), round(train_proportion * nrow(final_data)))

# Create the training set
train_data <- final_data[train_indices, ]

# Create the testing set excluding the training set
test_data <- final_data[-train_indices, ]
```

REGRESSION:

```
LinearRegresssion_model1 <- lm(StateRevenue ~ AvgTaxRate + TaxRateRank + AvgTaxRateOnWages + AvgTaxRate

summary(LinearRegresssion_model1)
```

```
##
## Call:
## lm(formula = StateRevenue ~ AvgTaxRate + TaxRateRank + AvgTaxRateOnWages +
```

```
##     AvgTaxRateOnWagesRank + MinTaxWage + TrustFund + TFPerWages +
##     TFWagesRank + Interest + HighCostMultiple + AvgHCM + AvgHCMRank,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -228063  -51296  -12095   60951  157057
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.600e+06  2.070e+05  -7.729 7.44e-10 ***
## AvgTaxRate             8.342e+05  4.254e+05   1.961 0.055933 .
## TaxRateRank            2.058e+03  6.716e+03   0.306 0.760721
## AvgTaxRateOnWages      4.267e+05  1.826e+05   2.337 0.023830 *
## AvgTaxRateOnWagesRank  1.756e+02  5.458e+03   0.032 0.974467
## MinTaxWage             6.970e+01  5.127e+00  13.594  < 2e-16 ***
## TrustFund              2.478e-01  6.076e-02   4.078 0.000178 ***
## TFPerWages            -5.213e+04  3.077e+04  -1.694 0.097024 .
## TFWagesRank            2.882e+04  7.133e+03   4.041 0.000200 ***
## Interest              -8.721e+00  5.943e+00  -1.467 0.149105
## HighCostMultiple      -2.090e+05  3.326e+05  -0.628 0.532805
## AvgHCM                 1.868e+05  2.499e+05   0.747 0.458661
## AvgHCMRank            -2.967e+04  7.379e+03  -4.021 0.000214 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88340 on 46 degrees of freedom
## Multiple R-squared:  0.9836, Adjusted R-squared:  0.9793
## F-statistic: 229.2 on 12 and 46 DF,  p-value: < 2.2e-16
```

The model effectively predicts state revenue using tax rates, trust fund size, and cost metrics, with tax rates and trust fund size being particularly impactful. However, there are predictions that are significantly off the mark, and a few variables do not appear to contribute much to the predictability of the model. Policymakers and analysts can use these insights to understand the factors affecting state revenue and consider revising tax rates or trust fund policies to influence revenue outcomes.

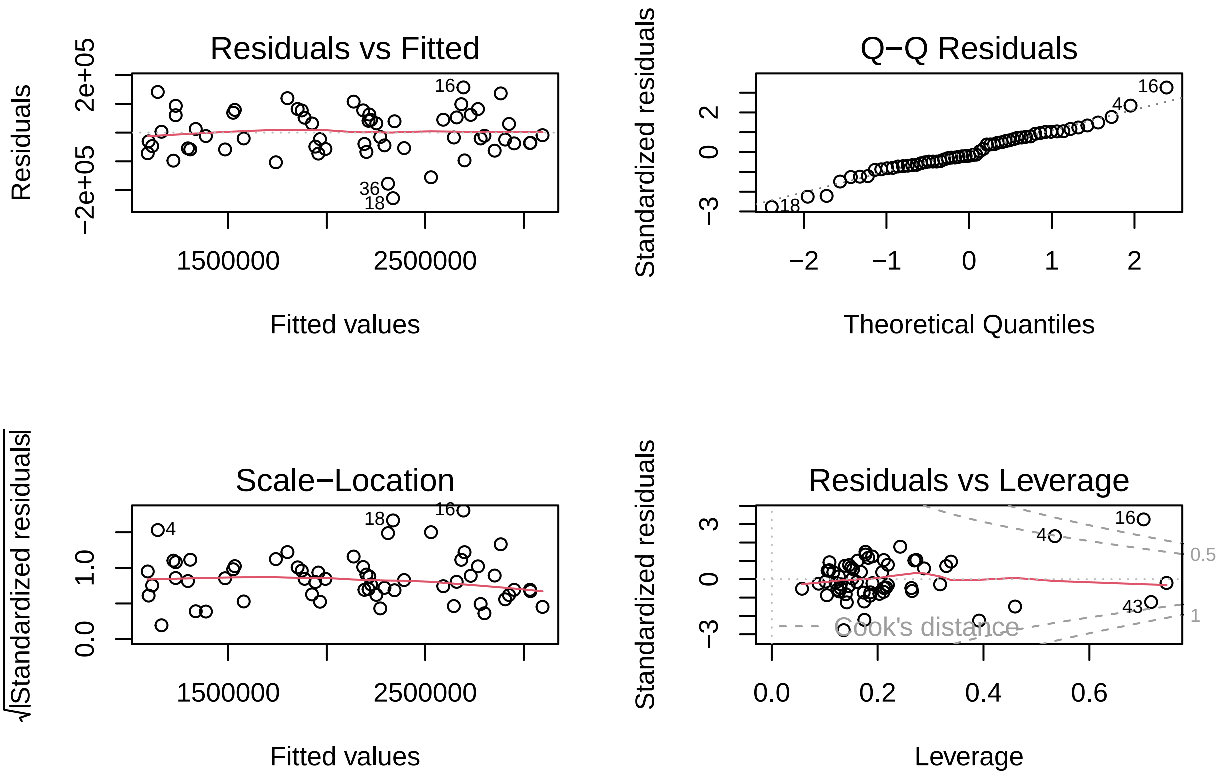
```r
LinearRegresssion_model2<- lm(StateRevenue ~ Year + Quarter+ AvgTaxRate + TaxRateRank + AvgTaxRateOnWag

summary(LinearRegresssion_model2)
```

```
##
## Call:
## lm(formula = StateRevenue ~ Year + Quarter + AvgTaxRate + TaxRateRank +
##     AvgTaxRateOnWages + AvgTaxRateOnWagesRank + MinTaxWage +
##     TrustFund + TFPerWages + TFWagesRank + Interest + HighCostMultiple +
##     AvgHCM + AvgHCMRank, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -185285  -45557  -13986   54175  163972
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
```

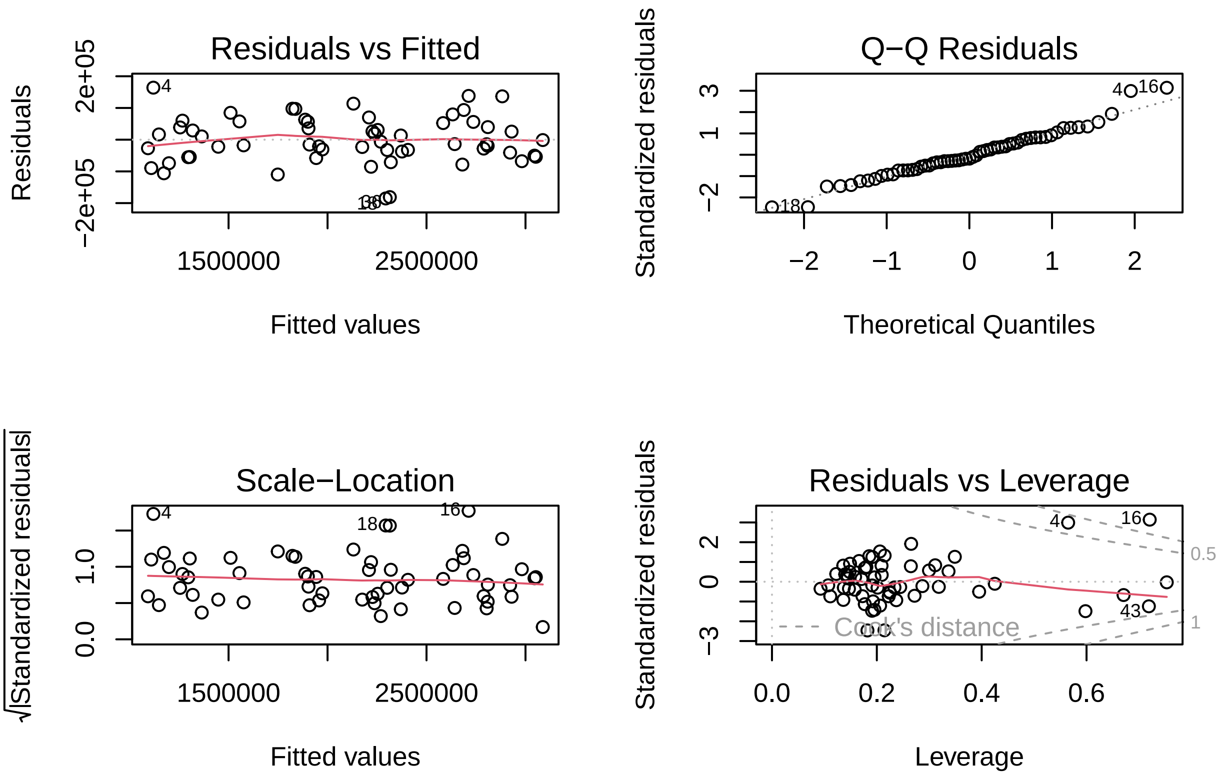
```
## (Intercept)           -1.001e+08  3.900e+07  -2.568  0.01370 *
## Year                   4.994e+04  1.978e+04   2.525  0.01525 *
## Quarter                1.467e+04  1.102e+04   1.332  0.18985
## AvgTaxRate             1.141e+06  4.160e+05   2.742  0.00879 **
## TaxRateRank            2.054e+03  6.389e+03   0.321  0.74939
## AvgTaxRateOnWages      2.560e+05  1.825e+05   1.402  0.16783
## AvgTaxRateOnWagesRank -1.061e+03  5.204e+03  -0.204  0.83934
## MinTaxWage             7.838e+00  2.533e+01   0.309  0.75848
## TrustFund              2.358e-01  5.812e-02   4.058  0.00020 ***
## TFPerWages            -5.833e+04  2.931e+04  -1.990  0.05286 .
## TFWagesRank            3.081e+04  6.798e+03   4.533 4.44e-05 ***
## Interest              -6.000e+00  5.821e+00  -1.031  0.30823
## HighCostMultiple      -4.346e+05  3.496e+05  -1.243  0.22039
## AvgHCM                 1.065e+05  2.515e+05   0.423  0.67403
## AvgHCMRank            -3.043e+04  7.111e+03  -4.280 9.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83170 on 44 degrees of freedom
## Multiple R-squared:  0.9861, Adjusted R-squared:  0.9816
## F-statistic: 222.2 on 14 and 44 DF,  p-value: < 2.2e-16
```

This model reinforces the findings of Model 1 regarding the impact of tax rates and the trust fund on state revenue while showing an additional annual increase in revenue. It provides a more detailed view by considering the year-over-year changes and potential seasonal effects within the year, although the latter did not show a significant effect. The inclusion of time variables has slightly improved the model's explanatory power, making it a potentially more accurate tool for forecasting New Jersey's state revenue. However, some predictors that were not significant could be examined further or potentially dropped in future modeling efforts.

```r
par(mfrow = c(2, 2))
plot(LinearRegresssion_model1)
```




```r
par(mfrow = c(2, 2))
plot(LinearRegresssion_model2)
```

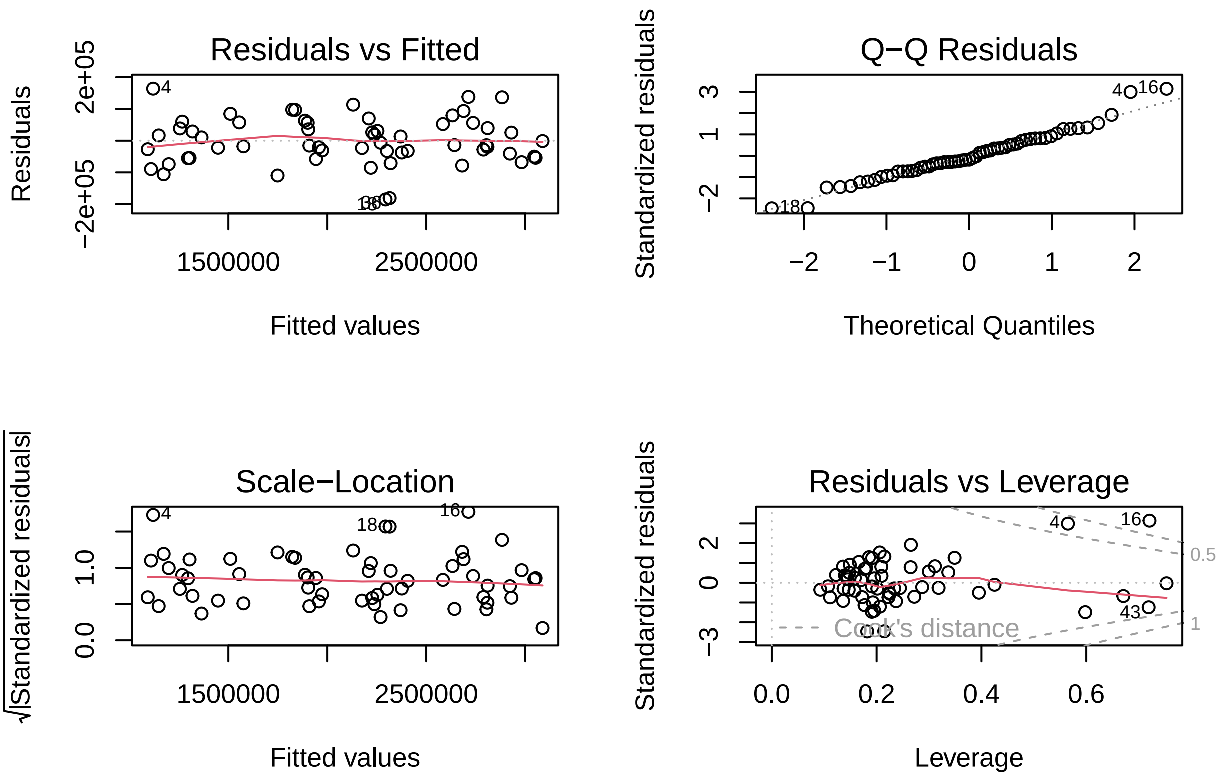



```
#Backward Selection
modback<-step(lm(final_data$StateRevenue ~ ., data = final_data),  direction = "backward", trace = 0)
summary(modback)
```

```
##
## Call:
## lm(formula = final_data$StateRevenue ~ Year + Quarter + AvgTaxRate +
##     AvgTaxRateOnWages + AvgTaxRateOnWagesRank + TrustFund + TFPerWages +
##     TFWagesRank + Interest + HighCostMultiple + AvgHCMRank, data = final_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -200237  -53718   -3163   55660  213126
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.176e+08  4.917e+06 -23.927  < 2e-16 ***
## Year                    5.884e+04  2.487e+03  23.660  < 2e-16 ***
## Quarter                 1.482e+04  9.160e+03   1.618 0.110012
## AvgTaxRate              7.944e+05  3.735e+05   2.127 0.036869 *
## AvgTaxRateOnWages       3.522e+05  1.623e+05   2.169 0.033359 *
## AvgTaxRateOnWagesRank  4.836e+03  2.633e+03   1.837 0.070337 .
## TrustFund               2.679e-01  4.637e-02   5.777 1.80e-07 ***
## TFPerWages             -5.356e+04  2.285e+04  -2.343 0.021869 *
## TFWagesRank             2.529e+04  5.863e+03   4.314 5.02e-05 ***
## Interest               -1.622e+01  4.535e+00  -3.577 0.000626 ***
```

```
## HighCostMultiple      -5.841e+05  2.099e+05  -2.783 0.006881 **
## AvgHCMRank            -2.673e+04  5.940e+03  -4.500 2.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91080 on 72 degrees of freedom
## Multiple R-squared:  0.9807, Adjusted R-squared:  0.9777
## F-statistic: 332.1 on 11 and 72 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2, 2))
plot(LinearRegresssion_model2)
```

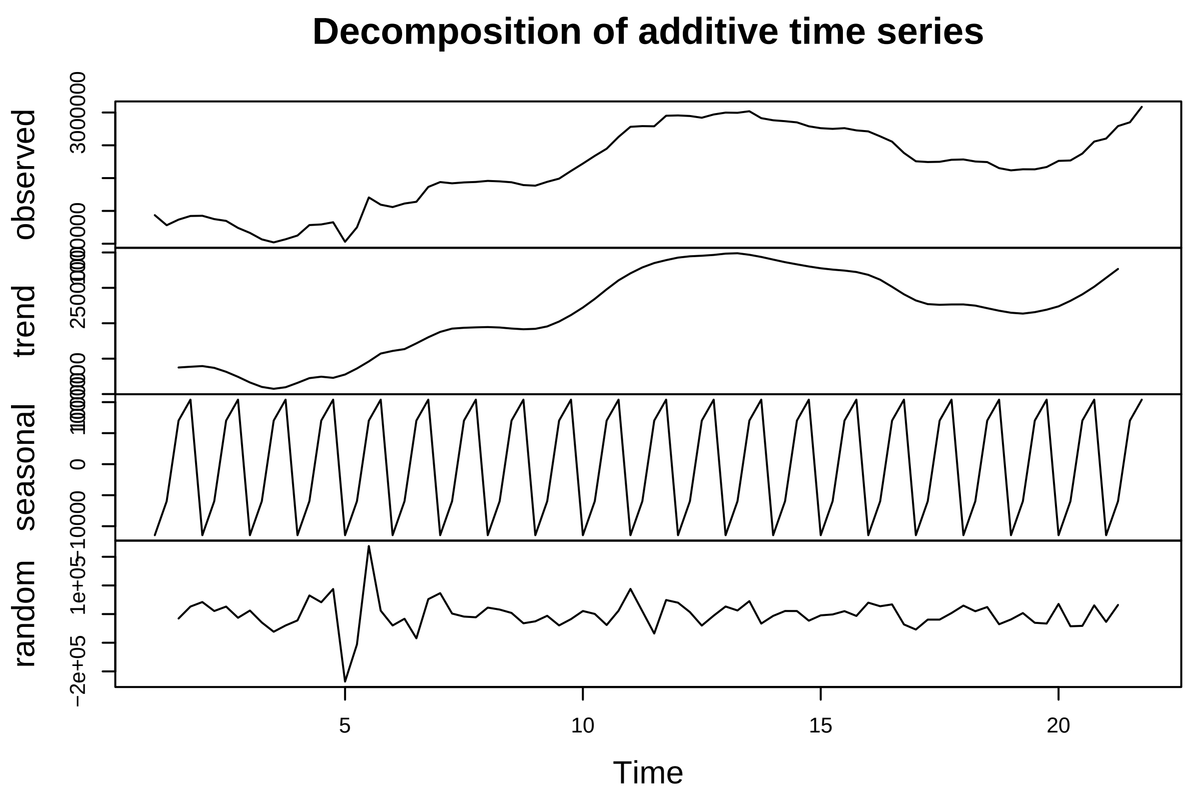


This backward selection model has done an excellent job at identifying the most relevant variables for predicting New Jersey's state revenue. It has highlighted the importance of tax rates, the size of the trust fund, trust fund rankings, and cost metrics as key drivers of revenue. Despite some predictors having counterintuitive relationships with revenue, such as TFPerWages, the model overall provides a very strong and statistically robust tool for understanding and forecasting state revenue. It will be especially useful for year-over-year revenue predictions due to the significance of the Year variable.

```r
#ARIMA Model
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
y=ts(final_data$StateRevenue,frequency=4)
plot(decompose(y))
```

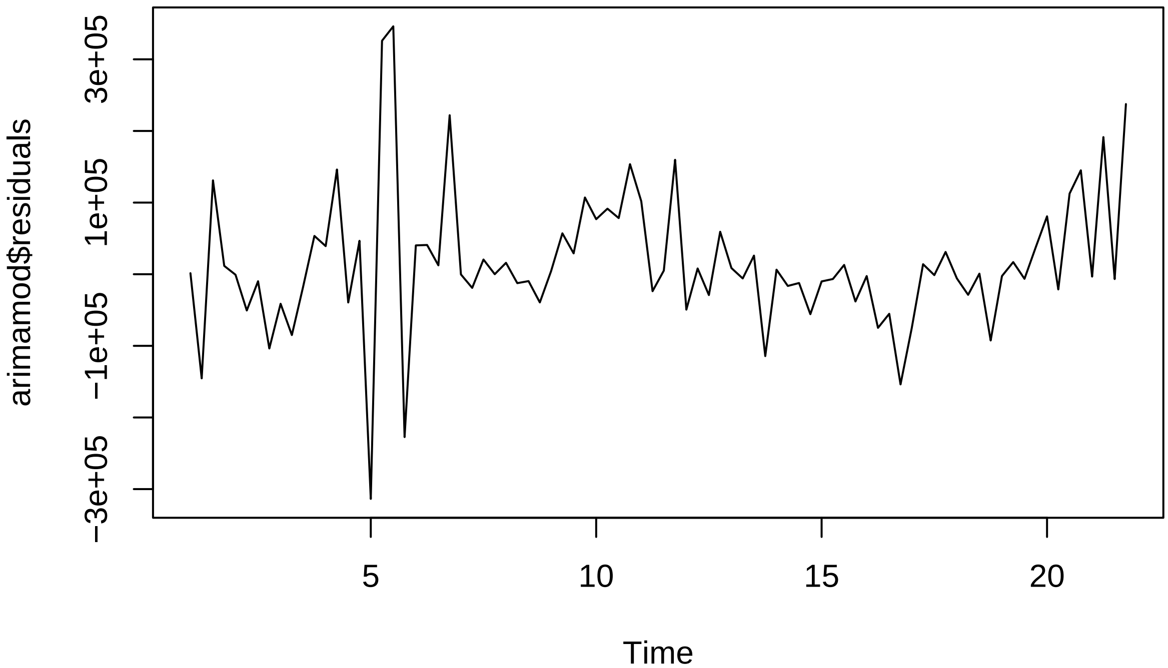



```r
arimamod<-auto.arima(ts(final_data$StateRevenue,frequency=4))
arimamod
```

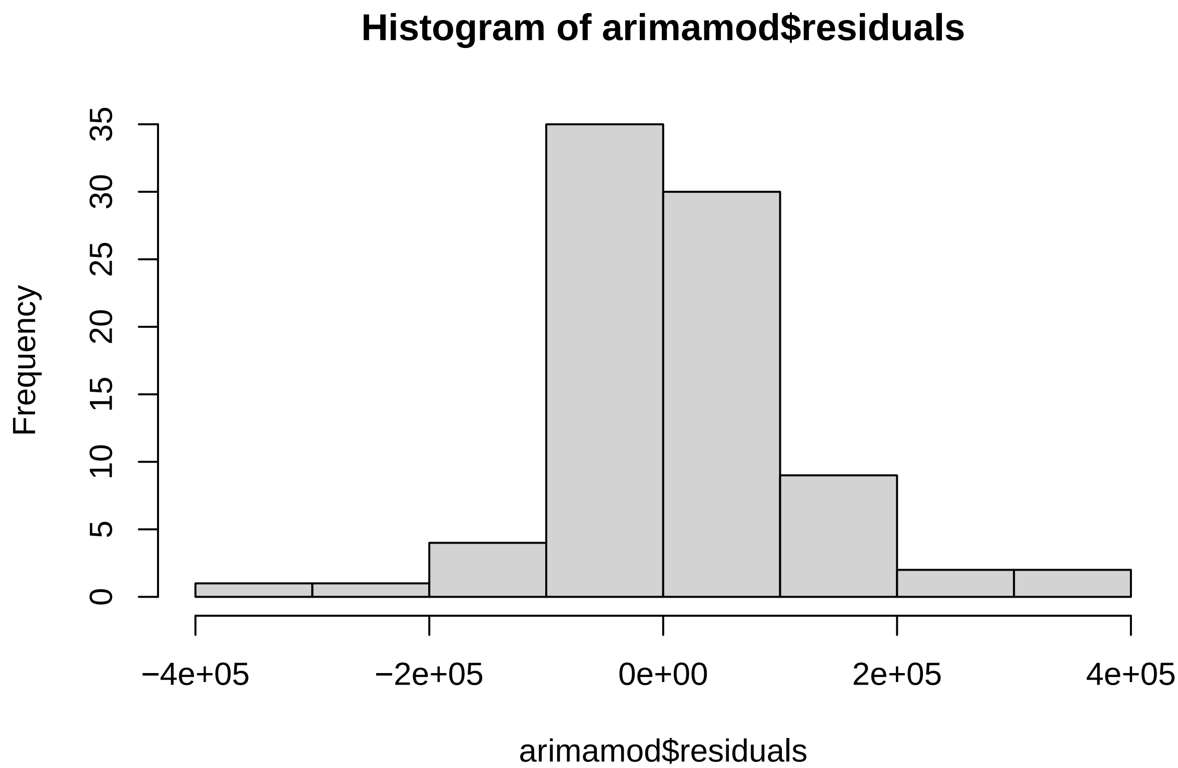
```
## Series: ts(final_data$StateRevenue, frequency = 4)
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.3373
## s.e.  0.1258
##
## sigma^2 = 1.01e+10:  log likelihood = -1073.33
## AIC=2150.65   AICc=2150.8   BIC=2155.49
```

The ARIMA model is a relatively simple time series model that is useful for forecasting state revenue based on its past changes and the relationship of past errors to the current prediction. The moderate ma1 value suggests that errors from one period can moderately predict errors in the next, which can be useful for short-term forecasting. The model's goodness of fit is reasonable, but whether this model is the best for predicting New Jersey's state revenue would depend on a comparison with other models' AIC and BIC values, as well as their practical forecasting performance.

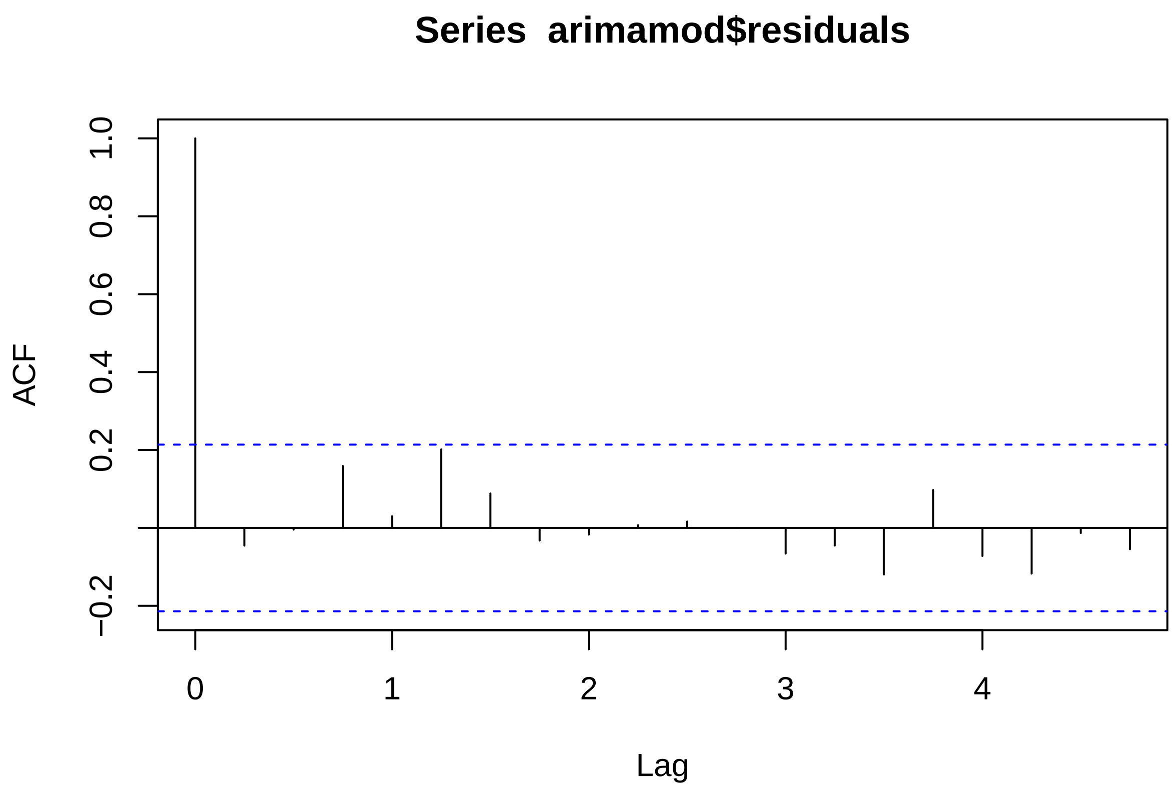
```r
#par(mfrow = c(2, 2))
plot(arimamod$residuals)
```



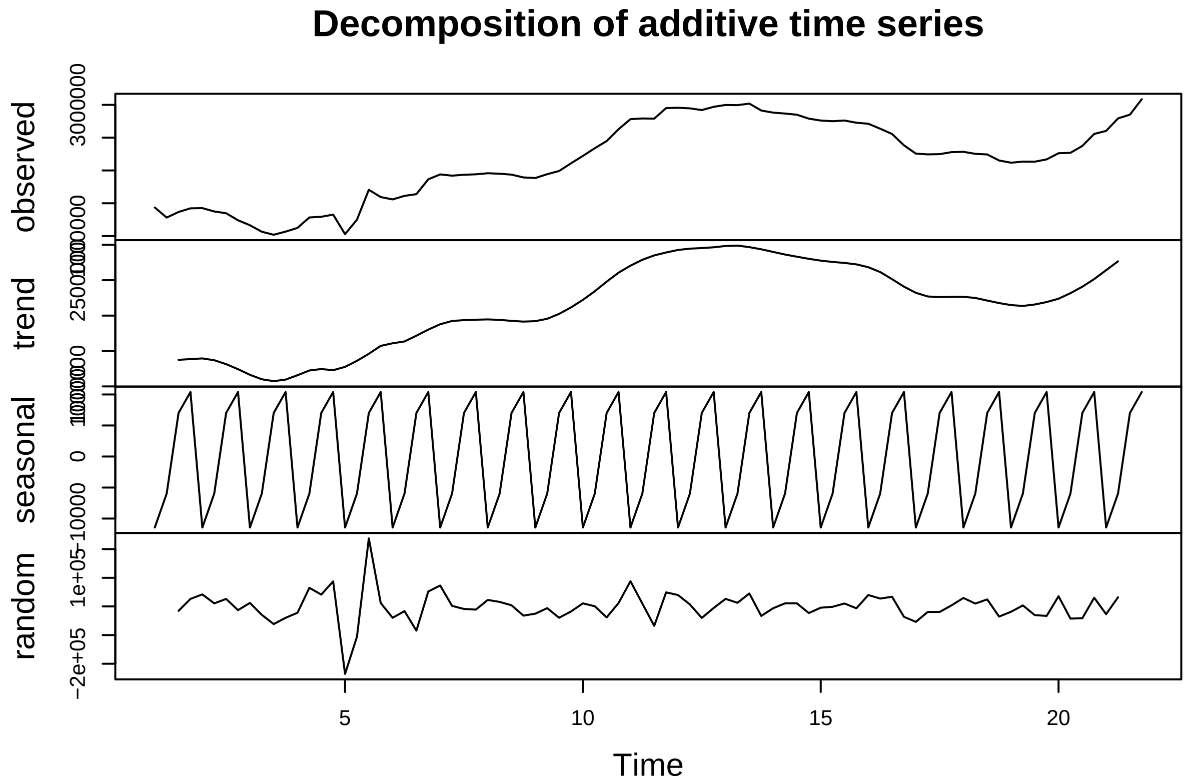

```r
hist(arimamod$residuals)
```




```
acf(arimamod$residuals)
```




```
library(forecast)
X<-as.matrix(final_data[,c("AvgTaxRate", "TrustFund"  , "TFWagesRank" , "Interest" , "HighCostMultiple"
y=ts(final_data$StateRevenue,frequency=4)
plot(decompose(y))
```

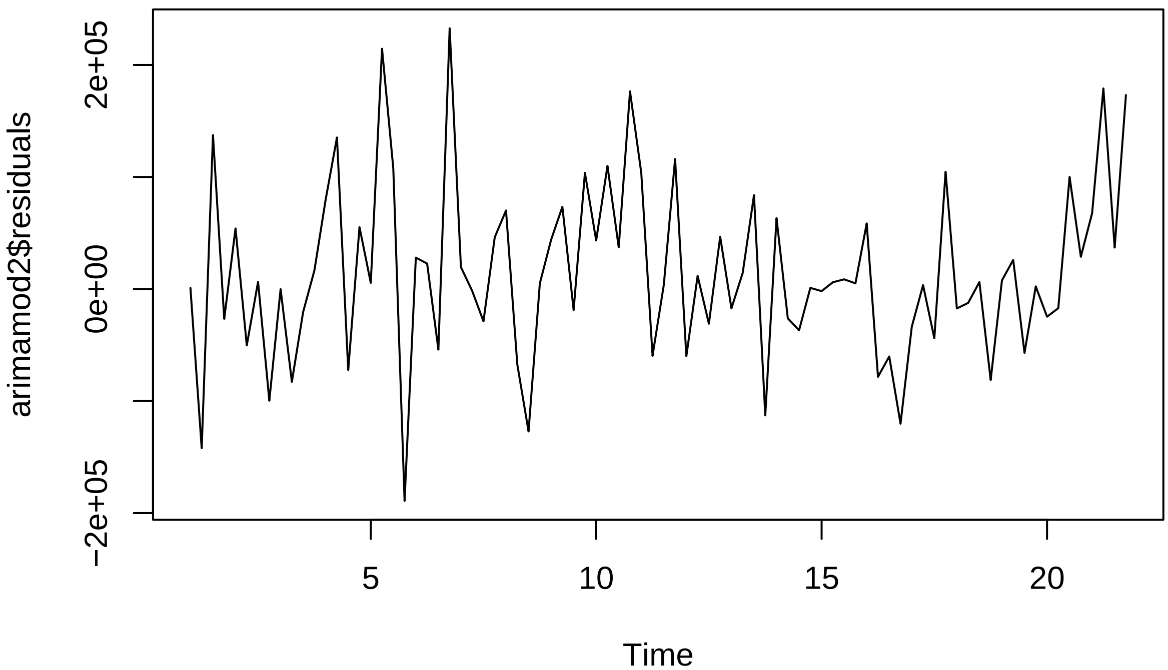



```r
arimamod2<-Arima(y,order = c(0,1,1),xreg=X)
arimamod2
```

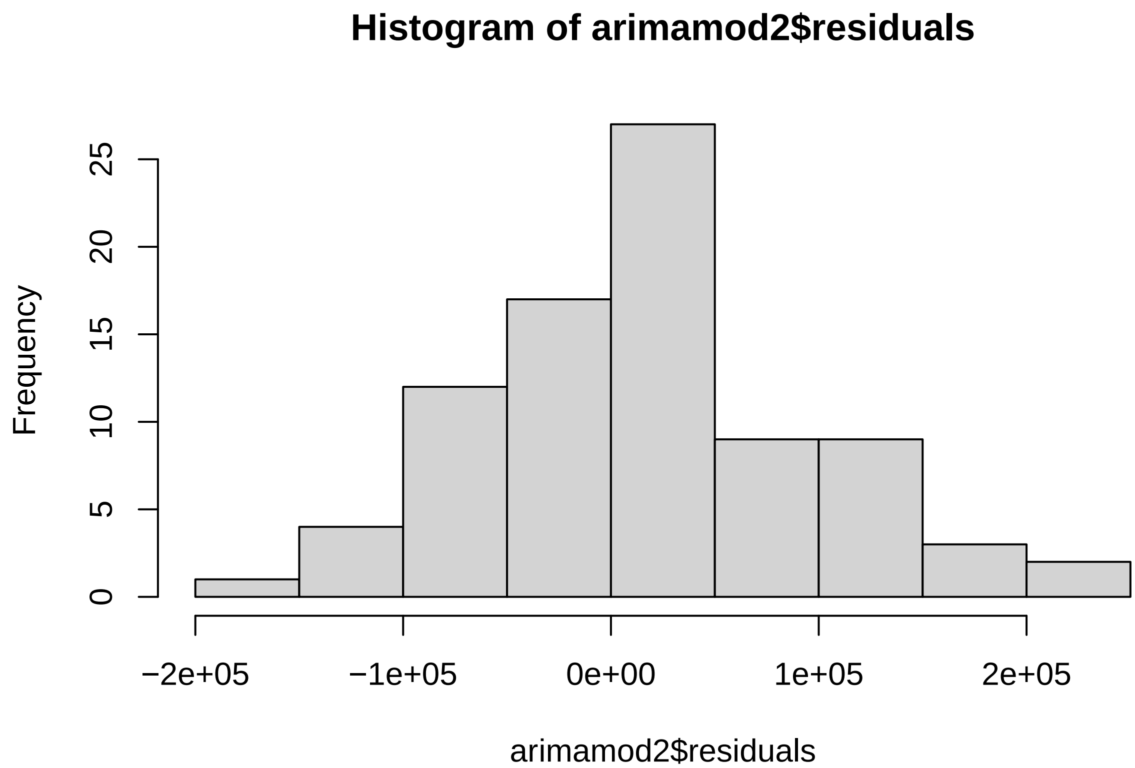
```
## Series: y
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##          ma1  AvgTaxRate  TrustFund  TFWagesRank  Interest  HighCostMultiple
##       0.3465    320693.3     0.0725     7361.567  -13.7676         -317857.0
## s.e.  0.1420    262053.7     0.0431     3386.705    3.7471          163103.9
##
## sigma^2 = 6.947e+09:  log likelihood = -1055.18
## AIC=2124.35   AICc=2125.85   BIC=2141.28
```

This ARIMA model with regression indicates that certain economic factors are significantly associated with the state revenue and that past errors can help predict future ones. It suggests that tax policy, financial health (as measured by the trust fund), and cost control are important for state revenue. The negative coefficients on interest and high costs indicate areas where spending reductions could potentially increase revenue. Overall, while the model is statistically significant and provides useful insights into factors affecting state revenue, there is still unexplained variability that could be explored with additional data or different modeling techniques.

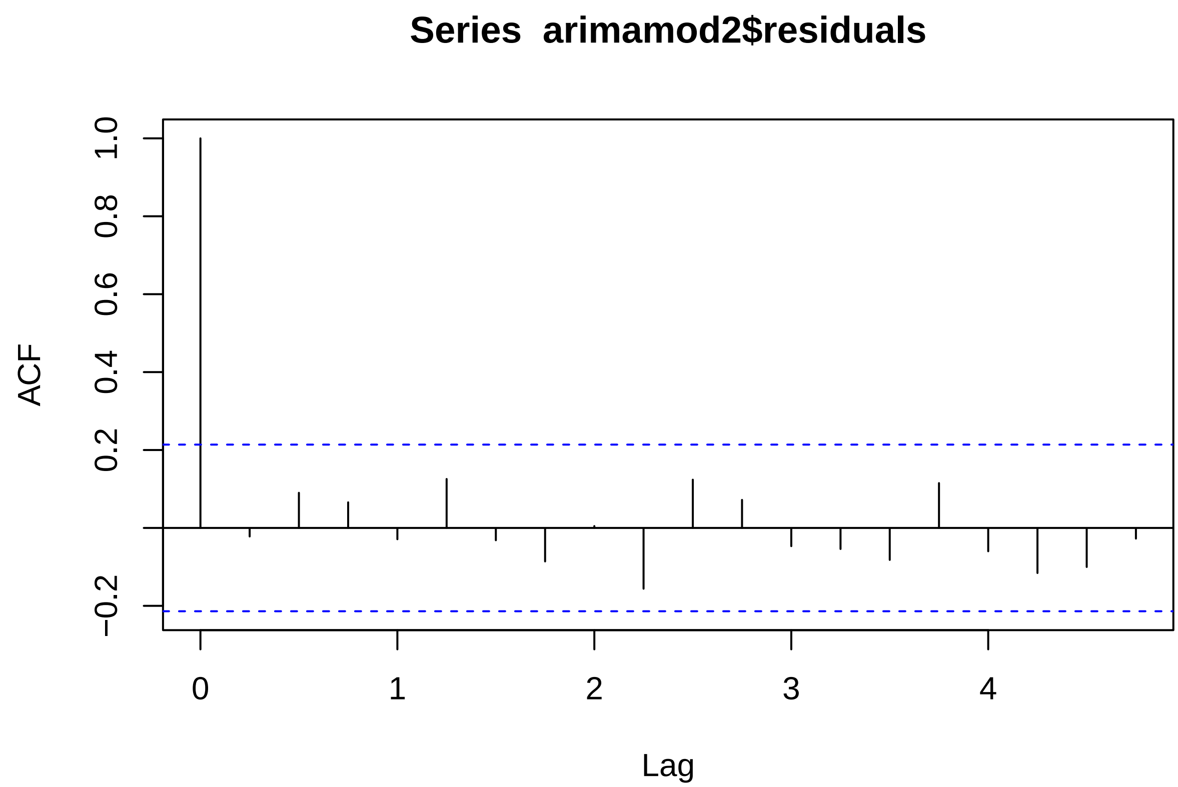
```r
#par(mfrow = c(2, 2))
plot(arimamod2$residuals)
```




```r
hist(arimamod2$residuals)
```




```
acf(arimamod2$residuals)
```




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

```
## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3
```

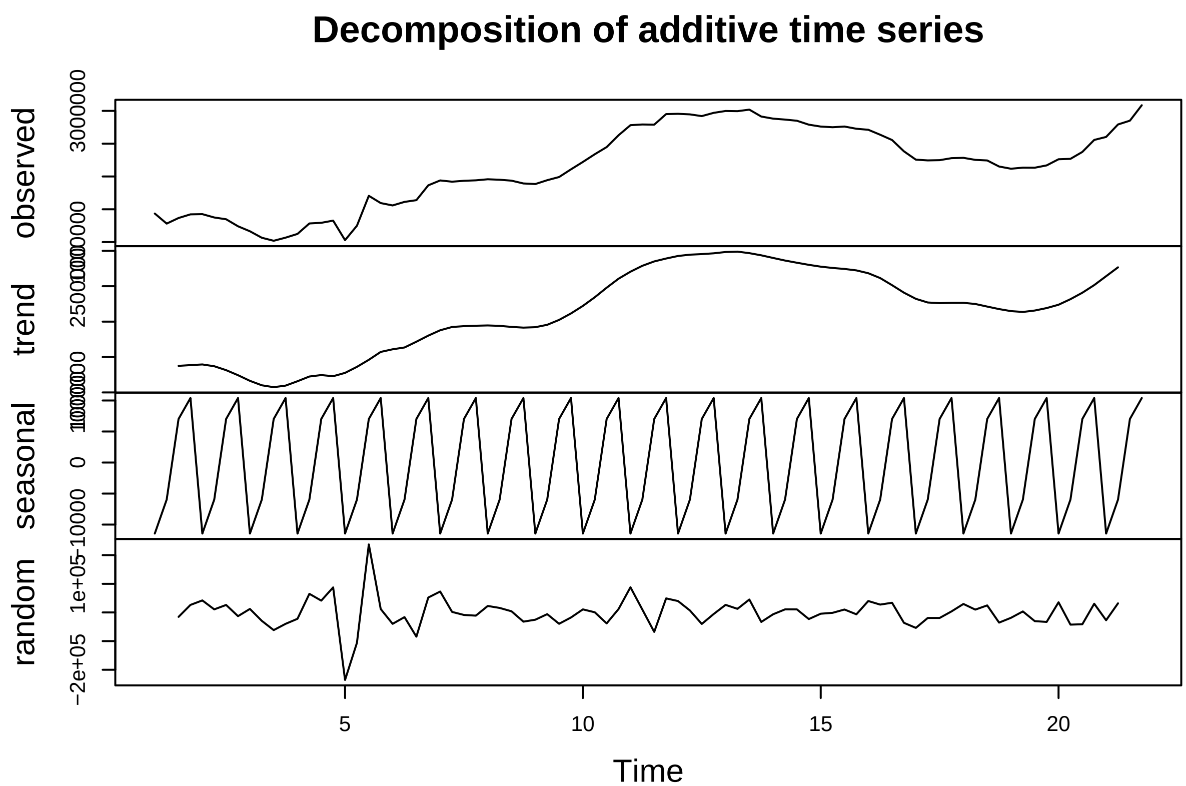
```r
vif_model <- vif(lm(StateRevenue ~ AvgTaxRate + Year + Quarter + TrustFund + TFWagesRank + Interest + Hi
print(vif_model)
```

```
##       AvgTaxRate             Year          Quarter        TrustFund
##         1.564269         3.296936         1.015812        13.164158
##      TFWagesRank         Interest HighCostMultiple
##         4.156988         5.027555         3.868768
```

```r
library(forecast)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice
```

```r
# Prepare exogenous variables matrix
X <- as.matrix(final_data[, c("AvgTaxRate", "Year", "Quarter", "Interest", "HighCostMultiple")])

# Center and scale predictors
X_scaled <- scale(X)

# Create time series object
y <- ts(final_data$StateRevenue, frequency = 4)
plot(decompose(y))
```

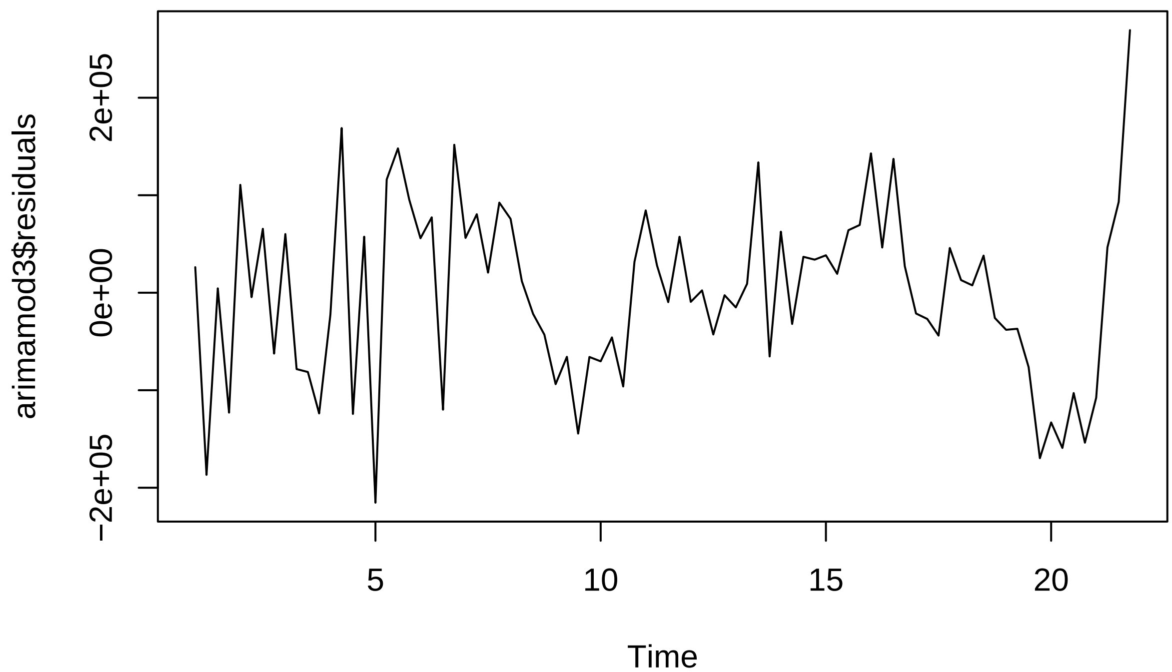



```r
# Fit ARIMA model
arimamod3 <- Arima(y, order = c(0, 0, 1), xreg = X_scaled)
arimamod3
```

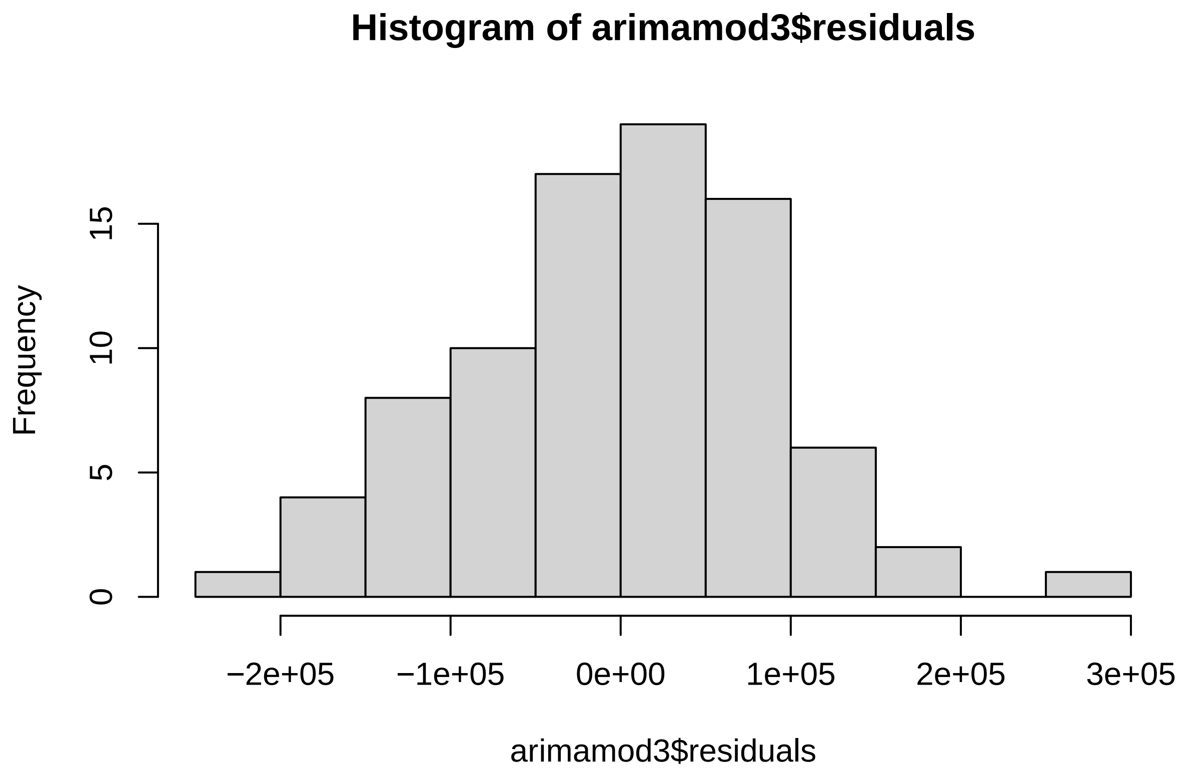
```
## Series: y
## Regression with ARIMA(0,0,1) errors
##
## Coefficients:
##          ma1   intercept  AvgTaxRate      Year    Quarter   Interest
##       0.5752  2152189.82   311732.90  510201.8  25208.403  -24465.26
## s.e.  0.0782    15447.07    17711.73   15728.5   7782.489   16921.17
##       HighCostMultiple
##              -63062.31
## s.e.          15655.09
##
```

```
## sigma^2 = 8.881e+09:  log likelihood = -1077.84
## AIC=2171.68   AICc=2173.6   BIC=2191.12
```

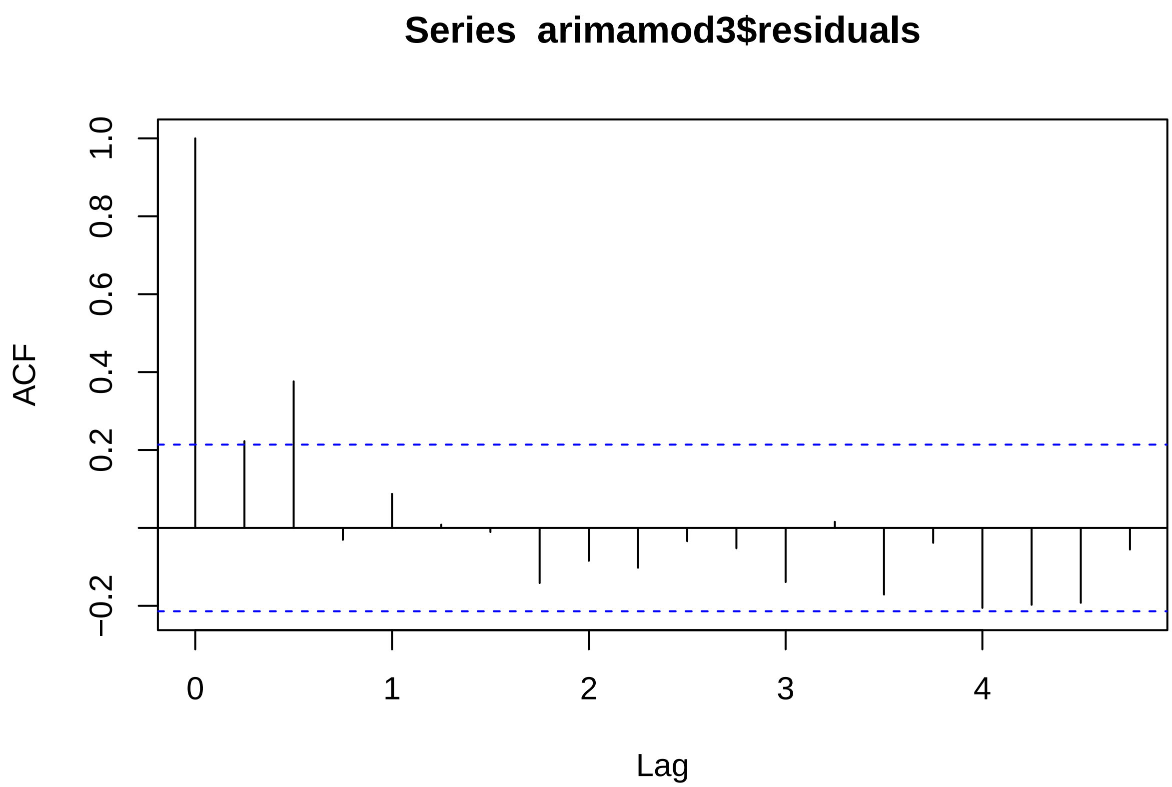
```r
#par(mfrow = c(2, 2))
plot(arimamod3$residuals)
```




```r
hist(arimamod3$residuals)
```




```
acf(arimamod3$residuals)
```



This model is a sophisticated tool that combines regression analysis with time series error modeling to predict state revenue. It concludes that tax rates, yearly trends, and economic costs are significant determinants of state revenue, with higher tax rates and yearly growth associated with increased revenue, while higher costs diminish it. The presence of autocorrelated errors suggests that while the model can predict revenue changes based on these variables, there's still a pattern in the forecast errors that could potentially be exploited to improve the model. This model would be particularly useful for making short-term predictions about state revenue when considering tax policy and cost management strategies.

```
library(readxl)

# Read the data from the Excel file
data <- read_excel("C:/Users/Suma2/Downloads/Applied stats_projrct/new jwrsey.xlsx")

# Set the seed for reproducibility
set.seed(123)

# Shuffle the data
data <- data[sample(nrow(data)), ]

# Splitting the data
test_data <- data[1:10, ]  # Test data with 10 records
train_data <- data[11:nrow(data), ]  # Training data with the rest

# You can now fit models on your train_data and evaluate them on test_data.

# Assuming you have fitted models and calculated AIC for each, here's how you could compare them:
# (Note: Replace model_aic_1, model_aic_2, etc., with actual AIC values from your models)
```

```r
model_aic_1 <- AIC(LinearRegresssion_model1)  # Replace model1 with your actual model
model_aic_2 <- AIC(LinearRegresssion_model2)
model_aic_3 <- AIC(modback)
model_aic_4 <- AIC(arimamod)
model_aic_5 <- AIC(arimamod2)
model_aic_6 <- AIC(arimamod3)

# Collecting all AICs for comparison
aic_values1 <- c(model_aic_1, model_aic_2,model_aic_3)
aic_values2 <- c(model_aic_4, model_aic_5, model_aic_6)


# Finding the model with the minimum AIC
min_aic <- min(aic_values1)
min_aic1 <- min(aic_values2)
best_model_index <- which(aic_values1 == min_aic)
best_model_index1 <- which(aic_values2 == min_aic1)

# Print the best model index

print(paste("AIC of Linear Regression model 1 is ", model_aic_1))
```

```
## [1] "AIC of Linear Regression model 1 is  1524.64653112886"
```

```r
print(paste("AIC of Linear Regression model 2 is ", model_aic_2))
```

```
## [1] "AIC of Linear Regression model 2 is  1518.90146233524"
```

```r
print(paste("AIC of Linear Regression model 3 is ", model_aic_3))
```

```
## [1] "AIC of Linear Regression model 3 is  2169.90169741169"
```

```r
print(paste("AIC of Arima model 1 is ", model_aic_4))
```

```
## [1] "AIC of Arima model 1 is  2150.65465187486"
```

```r
print(paste("AIC of Arima model 2 is ", model_aic_5))
```

```
## [1] "AIC of Arima model 2 is  2124.35169934136"
```

```r
print(paste("AIC of Arima model 3 is ", model_aic_6))
```

```
## [1] "AIC of Arima model 3 is  2171.67729325139"
```

```r
print(paste("The best Linear model is ", best_model_index, "with an AIC of", min_aic))
```

```
## [1] "The best Linear model is  2 with an AIC of 1518.90146233524"
```

```r
print(paste("The best Arima  Model is Model ", best_model_index1, "with an AIC of", min_aic1))
```

```
## [1] "The best Arima  Model is Model  2 with an AIC of 2124.35169934136"
```

So,The linear Model2 is the best among the two linear models.It provides a more detailed view by considering the year-over-year changes and potential seasonal effects within the year, although the latter did not show a significant effect. The inclusion of time variables has slightly improved the model's explanatory power, making it a potentially more accurate tool for forecasting New Jersey's state revenue

Model 1 adds economic indicators to the mix, combining regression with ARIMA errors. It identifies several significant predictors of state revenue, such as average tax rate and trust fund size, while also accounting for patterns in the model's errors.So,basic Arima model is the best with less AIC values