MTH783P Final Project:

Kothari Divya, Student ID:240485889, ah24177@qmul.ac.uk

Last compiled on: 07 May, 2025

Abstract

This project forecasts and compares the last 7 quarters of U.S. consumption changes using economic indicators. The methods used were ARIMA, ARIMAX and Linear regression.The best-performing model was a manually tuned ARIMAX(3,2,1)(1,0,1)[4], using Income, Savings, and Unemployment as predictors (AIC = 141.6)

This report analyzes quarterly percentage changes in U.S. economic indicators from 1970 Q1 to 2014 Q4 to forecast consumption changes for the last 7 quarters (2015 Q1–2016 Q3).The dataset from the fpp2 package includes five key variables: Consumption, Income, Production, Savings, and Unemployment.

Q1: Split the data :

The uschange data was split into a training set from 1970 Q1 up to and including 2014 Q4 and a test set from 2015 Q1 to 2016 Q3 using the window function.

Q2: Exploring the dataset:

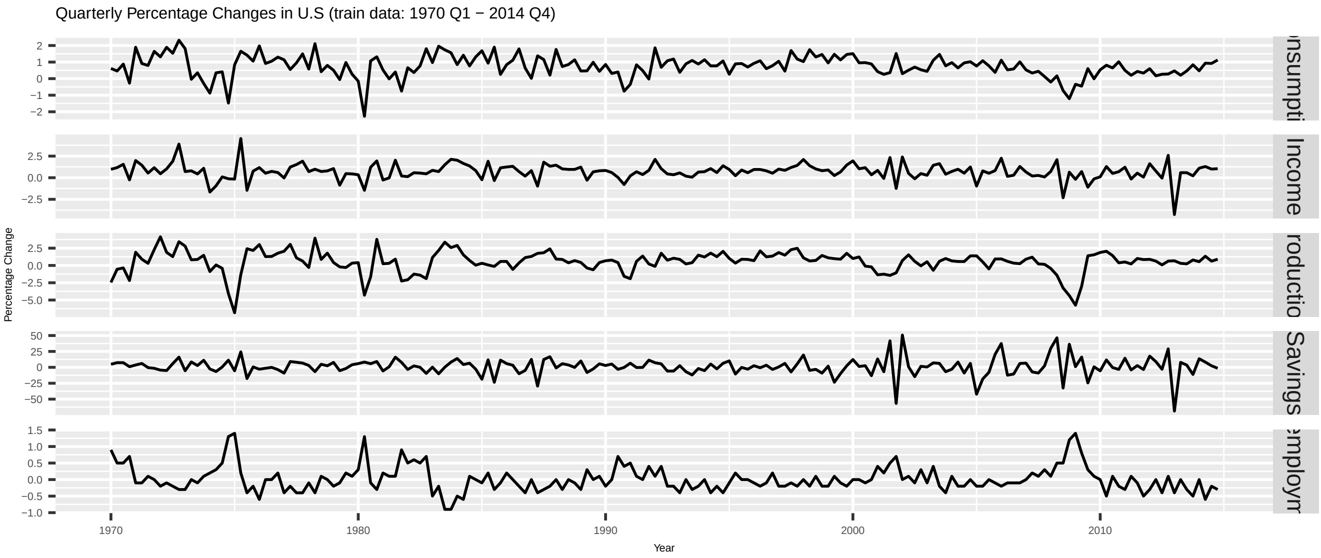
Variables include: Consumption, Income, Production, Savings, Unemployment. All variables represent percentage changes from the previous quarter, indicating already differenced data. There are 0 missing values.

Summary statistics of the training set:

	Min	Median	Mean	Max
Consumption	-2.27	0.79	0.75	2.32
Income	-4.27	0.72	0.72	4.54
Production	-6.85	0.70	0.54	4.15
Savings	-68.79	1.13	1.22	50.76
Unemployment	-0.90	0.00	0.01	1.40

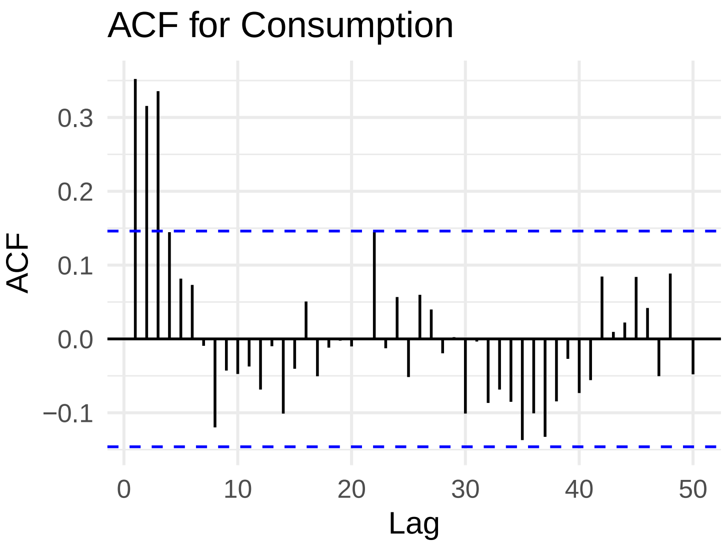
Observations : Consumption and Income have similar means and medians, indicating symmetry. Production is moderately variable, showing some fluctuations, but no significant trends. Unemployment shows minimal variation around 0, indicating stability. It also behaves in the opposite direction compared to the other variables. Savings exhibits high volatility, especially after 2000, with large fluctuations and outliers.

Visualize the data to understand the volatility and trends



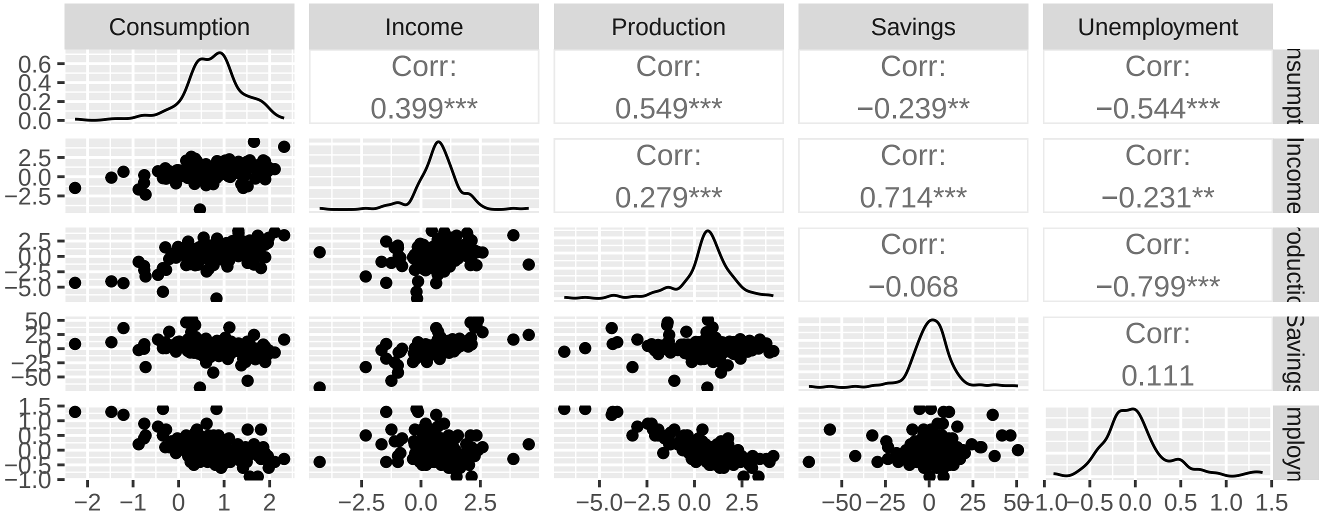
Key observations from the plot:No clear trend or seasonality, suggesting stationarity. Consumption, Income, and Production are centered around 0 with periods of volatility during certain years(1975, 1980, 2008,etc). Savings show a lot of volatility, especially after 2000, with large fluctuations. Unemployment has narrow fluctuations with some spikes during recessions. Consumption, income and production respond similar to each other and hint at correlation between them.

e} \end{figure}



Key observation from ACF plot: Consumption shows some autocorrelation (especially up to lag 3), suggesting mild non-stationarity.
Further, the Augmented Dickey-Fuller (ADF) test was applied to check for stationarity. All variables are stationary, as the p-value is less than 0.01. This confirms that despite the ACF suggesting some autocorrelation, the dataset is stationary and does not have a significant trend.

Pair plot to explore relationship between variables:



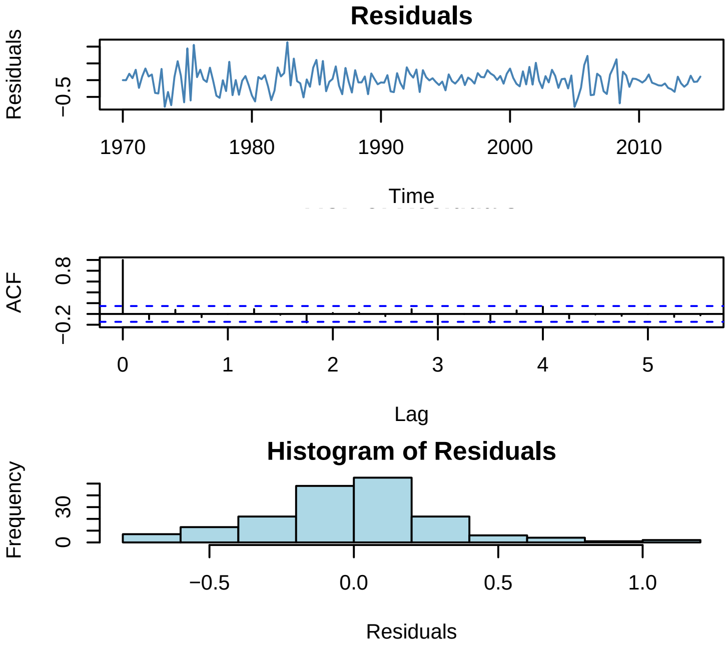
Key insights from the correlation analysis: Consumption is positively correlated with both Production(0.55) and Income(0.39).Consumption has a weak negative correlation with Savings(-0.24).Consumption has a moderate negative correlation with Unemployment(-0.54).Production and Unemployment have a strong negative correlation (-0.799).Savings and Income have a positive correlation (0.714),suggesting that higher income is associated with higher savings.

Q3: Model selection:

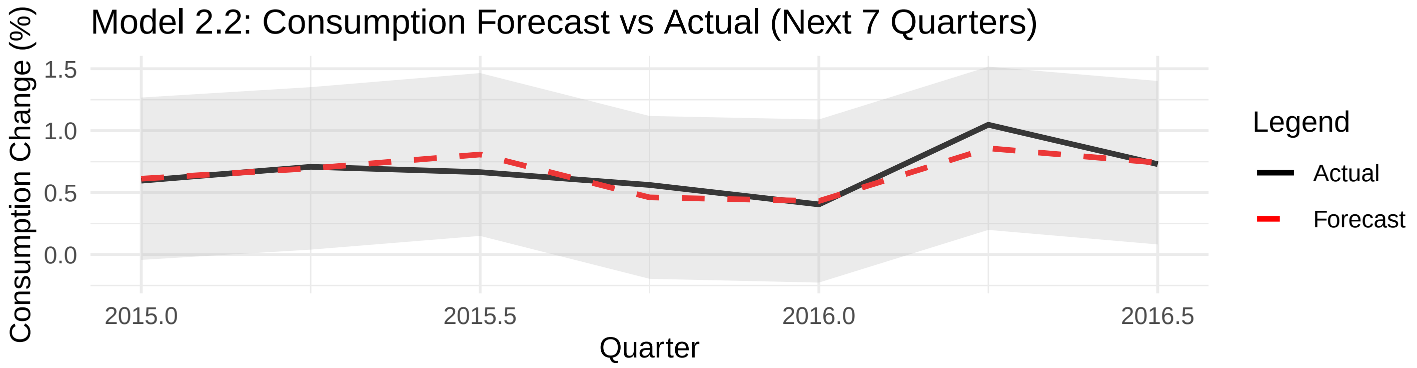
Manual tuning ARIMAX(3,2,1)(1,0,1)[4] seems like the best fit: This manually tuned model combines a regression structure with an ARIMA model excluding the 'Production' variable since its effect(previously tested on auto ARIMAX) was small (coefficient = 0.0326) and statistically weak (p-value = 0.106), while Income had a stronger impact (coefficient = 0.6963, $p < 0.001$). Removing it improved model accuracy (AIC dropped from 148.57 to 141.6). The model has Lowest AIC(141.6) and BIC (173.42),Lowest RMSE (0.322) and MAE (0.240) on training data among all models. Most reasonable MAPE (77.9) compared to other models. Includes Income, Savings and Unemployment as predictors, which all show significant relationships with Consumption.Residual Analysis:Residuals are mostly white noise (though with minor autocorrelation at some lags).Ljung-Box p=0.019 shows mild autocorrelation. Residual distribution is approximately normal and seems unbiased.ACF1 value (-0.096) shows minimal autocorrelation.
Metrics:

```
## AIC=141.6 | BIC=173.4 | RMSE=0.332 | Res.SD=0.322 | Ljung-Box(p)=0.193
```

Model Comparison:
Tested the dataset on three models:
1. **Auto ARIMA (3,0,0)(2,0,0)**[4]: No external predictors, higher AIC/BIC (333.74),high MAPE(177.01) and higher forecast errors.
2. **Auto ARIMA with all Regressors (Model 2.1–2.2)**: Good fit but low p-value (0.0009) rejects the null hypothesis and shows some residual autocorrelation in residuals at lags up to 8
3. **Simple Linear Regression**: Ignores time structure, shows some residual autocorrelation, and has higher errors than time series models.



Forecast and Comparison for the next 7 quarters:



```
##                         ME       RMSE        MAE       MPE     MAPE      MASE
## Training set -0.005044404 0.32161031 0.24011665 6.3450835 77.90340 0.3673577
## Test set      0.014996586 0.09842901 0.07154606 0.7504596 10.04255 0.1094593
##                     ACF1 Theil's U
## Training set -0.09615351        NA
## Test set     -0.43382748 0.3223499
```

```
##   Quarter Forecast Actual  Error
## 1 2015 Q1    0.612  0.596  0.016
## 2 2015 Q2    0.695  0.708 -0.013
## 3 2015 Q3    0.808  0.665  0.143
## 4 2015 Q4    0.461  0.562 -0.101
## 5 2016 Q1    0.432  0.405  0.027
## 6 2016 Q2    0.858  1.048 -0.190
## 7 2016 Q3    0.742  0.730  0.012
```

MODEL ASSUMPTIONS & VALIDATION: Linearity: Partial plots show Income (linear), Unemployment (inverse). Stationarity: 2nd order differencing, no trends in ACF/PACF. ADF test proved. Residuals analysis:Constant variance, approx normal.
LIMITATIONS: Residual autocorr.(p=0.019), possible overfitting, weak Savings (-0.044), no shock absorption, minimal seasonality.Cannot account for unforeseen events

STRENGTHS: AIC=141.6, BIC=173.42. ARIMAX combines predictors and time structure. High accuracy (RMSE=0.098, ME=-0.005).Residual Mean Error (ME = -0.005) is near zero, suggesting an unbiased model.
IMPROVEMENTS: Try dynamic regression/VAR, add more variables/lags.

[END of the REPORT]

R code

```r
if (!require("fpp2"))install.packages("fpp2")
if (!require("ggplot2"))install.packages("ggplot2")
if (!require("forecast"))install.packages("forecast")
if (!require("corrplot"))install.packages("corrplot")
if (!require("tidyverse"))install.packages("tidyverse")
if (!require("tseries"))install.packages("tseries")
if (!require("tseries"))install.packages("gridExtra")
library(fpp2)
library(ggplot2)
library(forecast)
library(corrplot)
library(tidyverse)
library(tseries)
library(corrplot)
library(gridExtra)
```

```r
# Q1: Split the data into training and test sets
# Checking where the dataset ends to ensure correct splitting
end(uschange)
```

```
## [1] 2016    3
```

```r
# Training set (1970 Q1 - 2014 Q4)
train <- window(uschange, end=c(2014, 4))
# Test set (2015 Q1 onwards)
test <- window(uschange, start=c(2015, 1))
```

```r
# Q2: Exploring the dataset
# View the first few rows of the training dataset
head(train)
```

```
##         Consumption     Income Production   Savings Unemployment
## 1970 Q1   0.6159862  0.9722610 -2.4527003 4.8103115          0.9
## 1970 Q2   0.4603757  1.1690847 -0.5515251 7.2879923          0.5
## 1970 Q3   0.8767914  1.5532705 -0.3587079 7.2890131          0.5
## 1970 Q4  -0.2742451 -0.2552724 -2.1854549 0.9852296          0.7
## 1971 Q1   1.8973708  1.9871536  1.9097341 3.6577706         -0.1
## 1971 Q2   0.9119929  1.4473342  0.9015358 6.0513418         -0.1
```

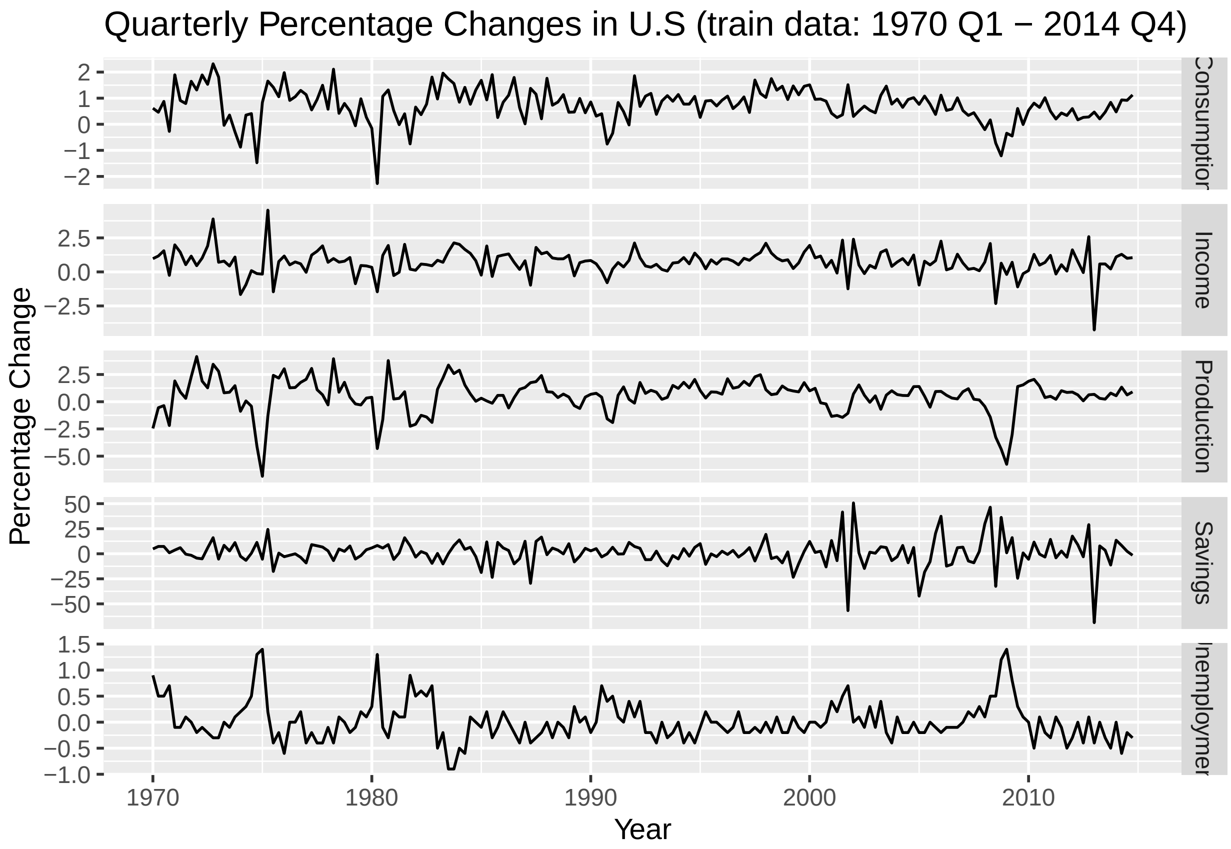
```r
# Check for missing values in the training set
sum(is.null(train))  # No missing values, returns 0
```

```
## [1] 0
```

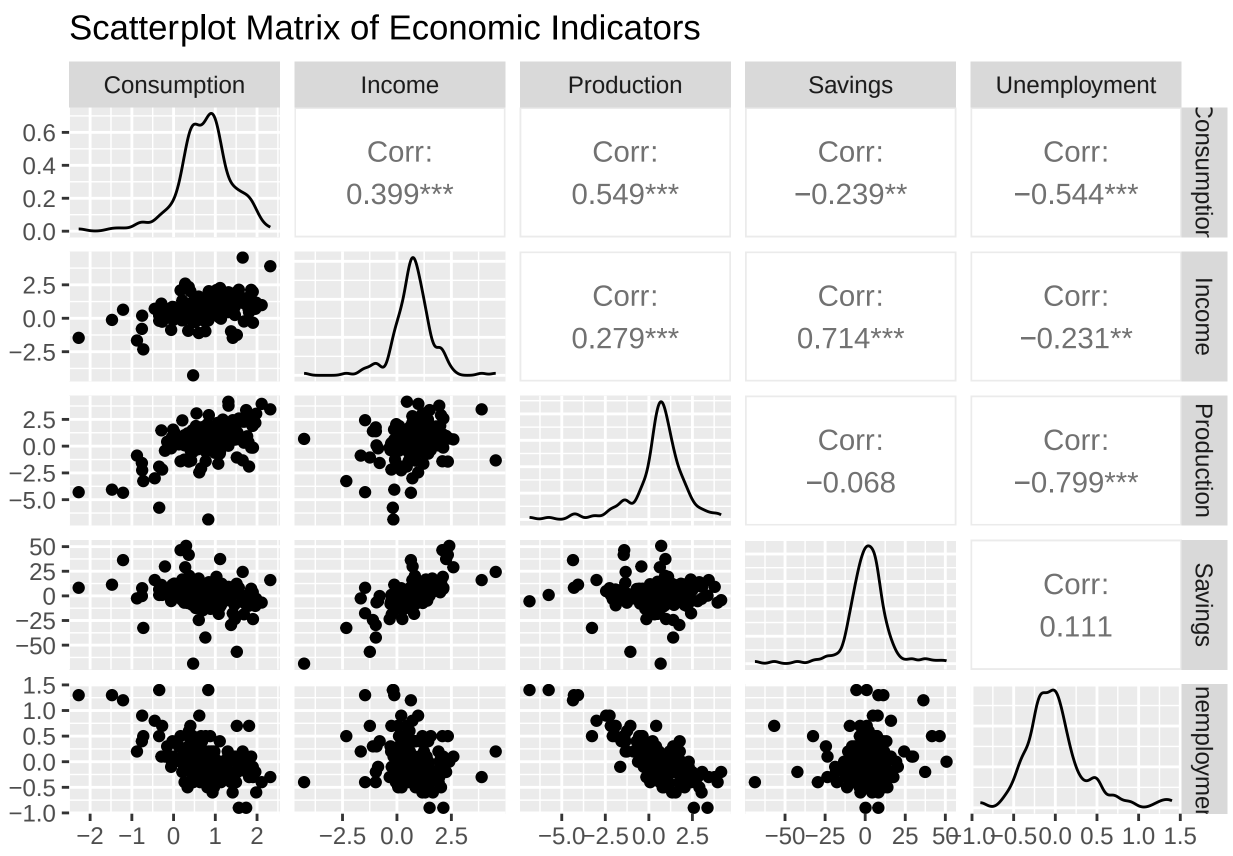
```r
# Summary statistics of the training set
summary(train)
```

```
##   Consumption          Income          Production          Savings
##  Min.   :-2.2741   Min.   :-4.2652   Min.   :-6.8510   Min.   :-68.788
##  1st Qu.: 0.4159   1st Qu.: 0.2833   1st Qu.: 0.1429   1st Qu.: -4.820
##  Median : 0.7888   Median : 0.7232   Median : 0.6979   Median :  1.133
##  Mean   : 0.7493   Mean   : 0.7185   Mean   : 0.5377   Mean   :  1.215
##  3rd Qu.: 1.1083   3rd Qu.: 1.1727   3rd Qu.: 1.3420   3rd Qu.:  7.065
##  Max.   : 2.3183   Max.   : 4.5365   Max.   : 4.1496   Max.   : 50.758
##   Unemployment
##  Min.   :-0.90000
##  1st Qu.:-0.20000
##  Median : 0.00000
##  Mean   : 0.01167
##  3rd Qu.: 0.10000
##  Max.   : 1.40000
```

```
# Visualize the data to understand the volatility and trends
autoplot(train, facets = TRUE) +
  ggtitle("Quarterly Percentage Changes in U.S (train data: 1970 Q1 - 2014 Q4)") +
  xlab("Year") + ylab("Percentage Change")
```



```
# Scatterplot matrix to explore relationship between variables
library(GGally)
ggpairs(as.data.frame(train),title = "Scatterplot Matrix of Economic Indicators")
```

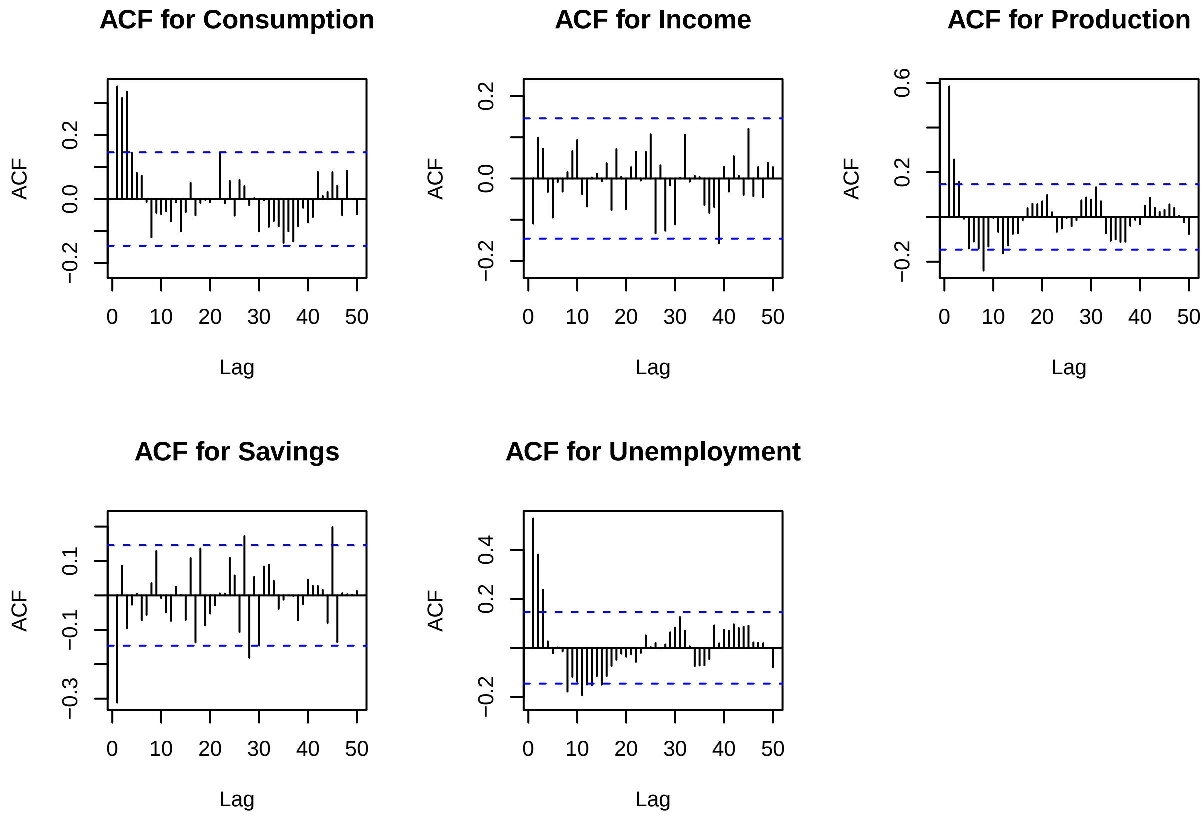


```r
# Plot all variables in the training set

# Calculate correlation between Consumption and other variables
cor_values <- cor(train[, 1], train[, -1])
cor_values  # Display the correlation values
```

```
##          Income Production    Savings Unemployment
## [1,] 0.3989102  0.5487901 -0.2393123   -0.5439411
```

```r
# Plot Autocorrelation Function (ACF) to check for trend or seasonality
par(mfrow = c(2, 3))
for (i in 1:ncol(train)) {
  Acf(train[, i], lag.max = 50, main = paste("ACF for", colnames(train)[i]))
  #  ACF for each variable
}
par(mfrow = c(1, 1))
```



```r
# Apply Augmented Dickey-Fuller (ADF) test to check for stationarity
library(tseries)
for (i in 1:ncol(train)) {
  print(paste("ADF test for column:", colnames(train)[i]))
  adf_test <- adf.test(train[, i])
  print(adf_test)
}
```

```
## [1] "ADF test for column: Consumption"
```

```
## Warning in adf.test(train[, i]): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train[, i]
## Dickey-Fuller = -4.4219, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
## [1] "ADF test for column: Income"
```

```
## Warning in adf.test(train[, i]): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train[, i]
## Dickey-Fuller = -6.0046, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
## [1] "ADF test for column: Production"
## Warning in adf.test(train[, i]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  train[, i]
## Dickey-Fuller = -5.1571, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
## [1] "ADF test for column: Savings"

## Warning in adf.test(train[, i]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  train[, i]
## Dickey-Fuller = -6.7917, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
## [1] "ADF test for column: Unemployment"

## Warning in adf.test(train[, i]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  train[, i]
## Dickey-Fuller = -4.3548, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
##Q3.
#Extract Consumption variable from train and test data
c_train <- train[, "Consumption"]
c_test <- test[, "Consumption"]
```

```
# Model 1: ARIMA Models
# Model 1.1: ARIMA(3,0,0): since lag at 3
m1_fit <- Arima(c_train, order = c(3, 0, 0))
print(summary(m1_fit))
```

```
## Series: c_train
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3    mean
##       0.2291  0.1610  0.2034  0.7515
## s.e.  0.0727  0.0736  0.0724  0.1072
##
## sigma^2 = 0.361:  log likelihood = -161.87
## AIC=333.74   AICc=334.09   BIC=349.71
##
## Training set error measures:
```

```
##                       ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.001211929 0.5941473 0.4384788 49.66529 177.0143 0.6708347
##                    ACF1
## Training set 0.01552794
```

```r
# Model 1.2: Auto ARIMA
m1_auto <- auto.arima(c_train)
summary(m1_auto)
```

```
## Series: c_train
## ARIMA(3,0,0)(2,0,0)[4] with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3     sar1     sar2    mean
##       0.2271  0.1777  0.2200  -0.0334  -0.1803  0.7522
## s.e.  0.0737  0.0736  0.0724   0.0772   0.0744  0.0946
##
## sigma^2 = 0.353:  log likelihood = -158.94
## AIC=331.88   AICc=332.53   BIC=354.23
##
## Training set error measures:
##                         ME      RMSE       MAE      MPE     MAPE     MASE
## Training set 0.0002895183 0.5841338 0.4391314 65.53638 188.5312 0.671833
##                     ACF1
## Training set 0.009685967
```

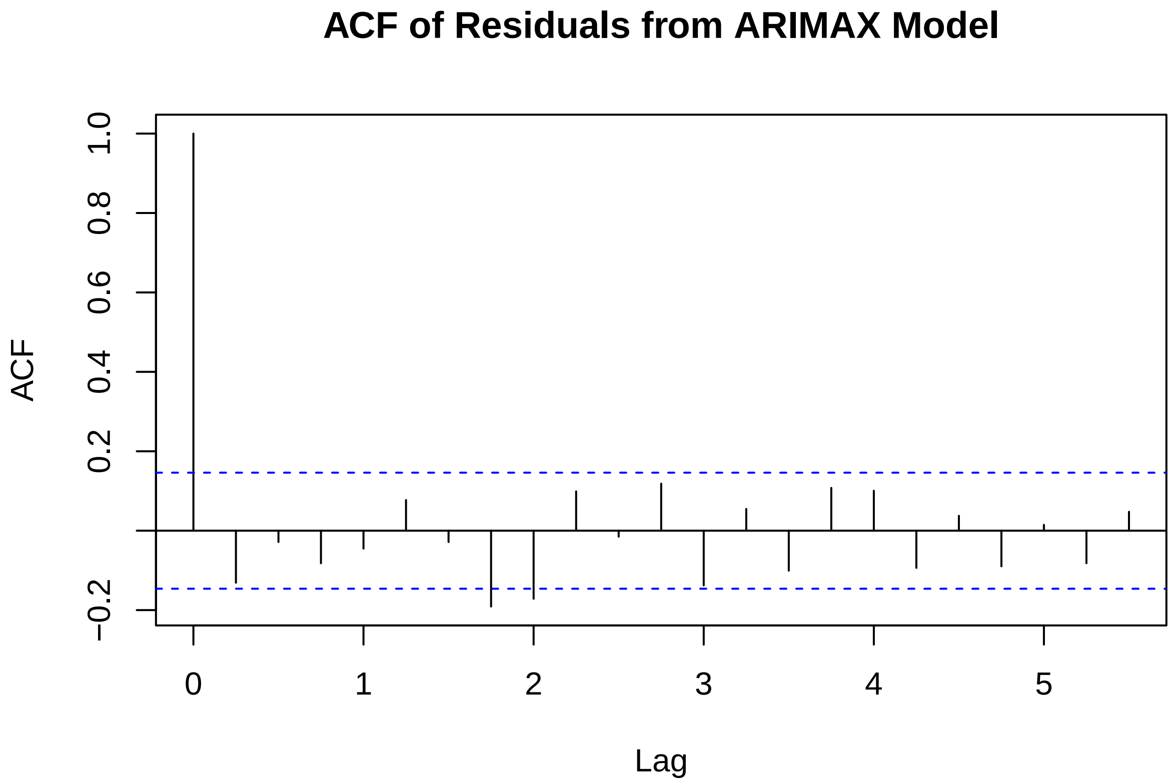
```r
#model shows moderate training set errors but the MAPE(188.53) is unusually high

# Model 2: ARIMA with External Predictors
# Model 2.1: Include all variables
fit_regarima <- auto.arima(c_train,xreg =
                             train[, c("Income", "Production", "Savings",
                                                  "Unemployment")])
summary(fit_regarima)
```

```
## Series: c_train
## Regression with ARIMA(3,1,0)(1,0,0)[4] errors
##
## Coefficients:
##           ar1      ar2      ar3     sar1    drift  Income  Production  Savings
##       -0.8047  -0.5422  -0.5286  -0.4607  -0.0008  0.6963      0.0326  -0.0449
## s.e.   0.0671   0.0796   0.0808   0.0793   0.0062  0.0452      0.0270   0.0031
##       Unemployment
##            -0.2667
## s.e.        0.1140
##
## sigma^2 = 0.1255:  log likelihood = -64.28
## AIC=148.57   AICc=149.88   BIC=180.44
##
## Training set error measures:
##                          ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.0004764331 0.3442272 0.2633871 6.178704 92.59988 0.4029595
##                     ACF1
## Training set -0.1310828
```

```
residuals_regarima <- residuals(fit_regarima)

# Plot the ACF of residuals
acf(residuals_regarima, main = "ACF of Residuals from ARIMAX Model")
```

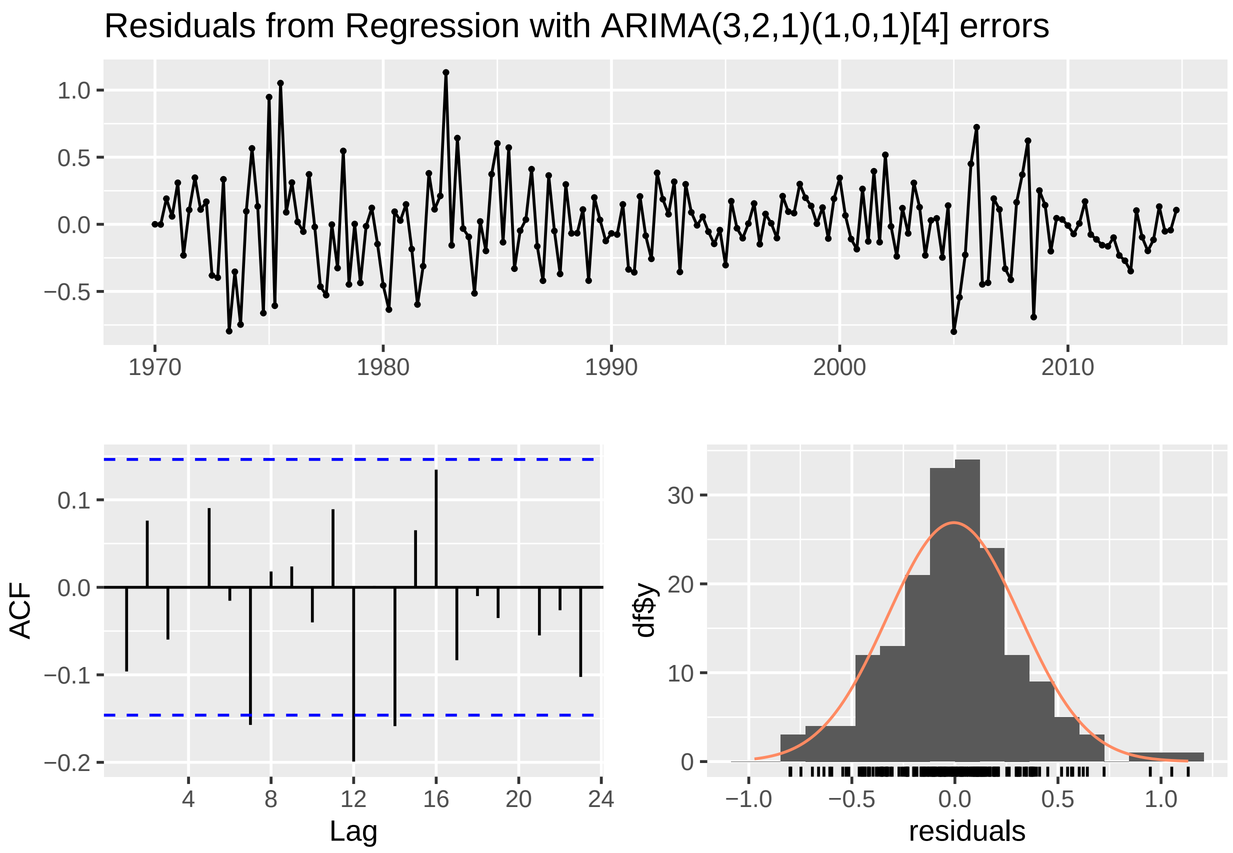


```
#Model 2.2: manual tuning ARIMAX
#the ACF/PACF of auto ARIMAX showed autocorrelation up to lag 3 and seasonality every 4 quarter
manual_regarima<- Arima(c_train,order = c(3, 2, 1),
                        seasonal = c(1, 0, 1),
                        xreg = train[, c("Income", "Savings", "Unemployment")])
summary(manual_regarima)
```

```
## Series: c_train
## Regression with ARIMA(3,2,1)(1,0,1)[4] errors
##
## Coefficients:
##           ar1      ar2      ar3      ma1     sar1     sma1  Income  Savings
##       -0.9712  -0.9226  -0.9483  -1.0000  -0.0660  -0.9605  0.7054  -0.0441
## s.e.   0.0307   0.0515   0.0377   0.0313   0.0815   0.0702  0.0421   0.0029
##       Unemployment
##            -0.3706
## s.e.        0.0746
##
## sigma^2 = 0.1102:  log likelihood = -60.8
## AIC=141.6   AICc=142.92   BIC=173.42
##
```

```
## Training set error measures:
##                        ME      RMSE       MAE      MPE    MAPE      MASE
## Training set -0.005044404 0.3216103 0.2401166 6.345083 77.9034 0.3673577
##                     ACF1
## Training set -0.09615351
```

```
#Residual Analysis:
checkresiduals(manual_regarima)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,2,1)(1,0,1)[4] errors
## Q* = 9.8494, df = 3, p-value = 0.01989
##
## Model df: 6.   Total lags used: 9
```

```
# Model 3:  Linear Models
# Model 3.1: Multiple regression with all predictors
fit_lm <- tslm(Consumption ~ Income + Production +
                 Savings + Unemployment, data = train)
summary(fit_lm)
```

```
##
## Call:
## tslm(formula = Consumption ~ Income + Production + Savings +
##     Unemployment, data = train)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max 
## -0.88250 -0.18191 -0.03121  0.15109  1.20338 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   0.270341   0.038761   6.975 6.05e-11 ***
## Income        0.713487   0.043019  16.586  < 2e-16 ***
## Production    0.044181   0.027212   1.624    0.106    
## Savings      -0.045189   0.002834 -15.945  < 2e-16 ***
## Unemployment -0.212744   0.110569  -1.924    0.056 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3347 on 175 degrees of freedom
## Multiple R-squared:  0.7538, Adjusted R-squared:  0.7482 
## F-statistic: 133.9 on 4 and 175 DF,  p-value: < 2.2e-16
```

```r
# Calculate AIC and BIC for the linear model
aic_lm <- AIC(fit_lm)
bic_lm <- BIC(fit_lm)
print(paste("AIC: ", aic_lm))
```

```
## [1] "AIC:  123.682785513849"
```

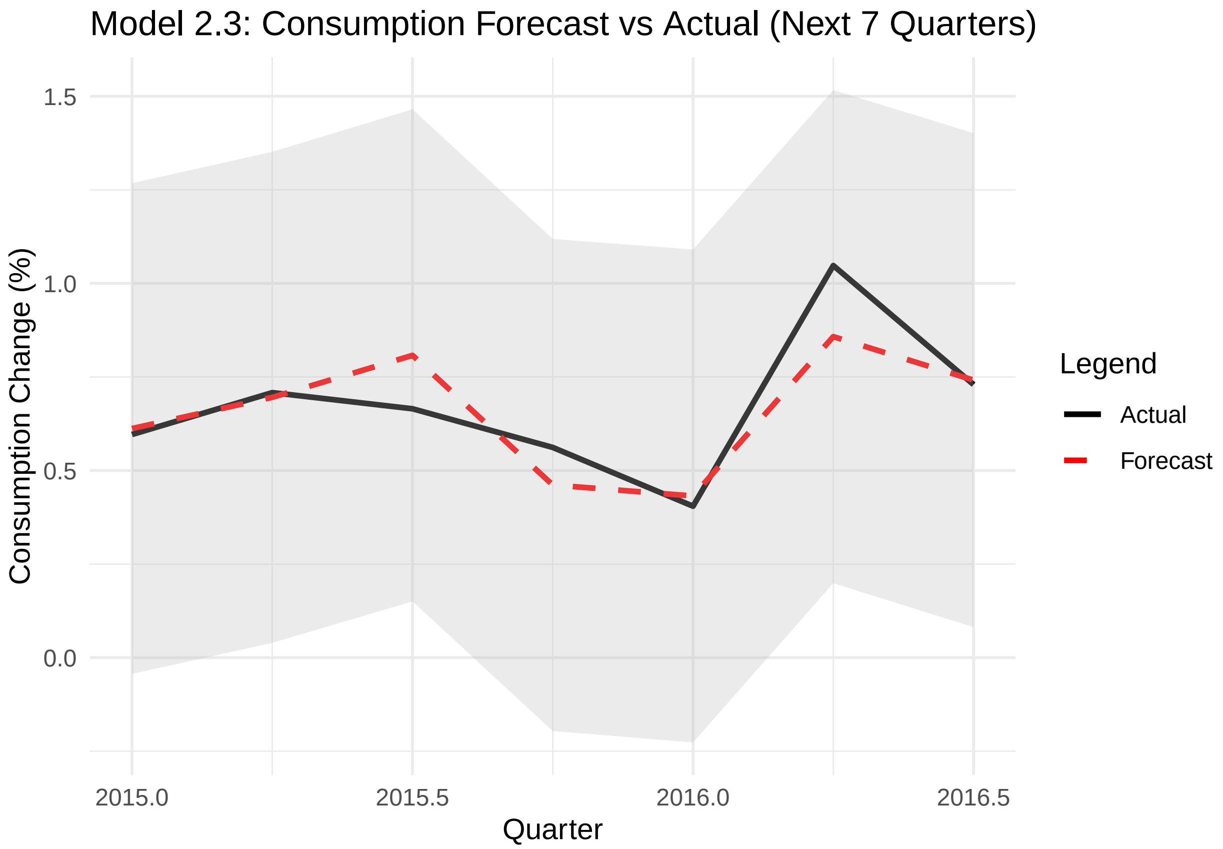
```r
print(paste("BIC: ", bic_lm))
```

```
## [1] "BIC:  142.84052661919"
```

```r
# Model Selection:
#Model 2.2 : Manual tuning ARIMAX seems like the best fit
#Lowest AIC(141.6) and BIC (173.42) among all models.
#Lowest RMSE (0.322) and MAE (0.240) on training data.
#Most reasonable MAPE (77.9) compared to other models.
#Includes Income, Savings and Unemployment as predictors, which all show significant relationsh
#Residual Analysis:Residuals are mostly white noise (though with minor autocorrelation at some
```

```r
# Forecast the next 7 quarters:
# Best model(ARIMA with regressors, manual tuning) forecast
best_forecast <- forecast(manual_regarima,
                          xreg = test[, c("Income", "Savings", "Unemployment")],h = 7)
#plot for the comparison of actual vs forecast values
# Create a dataframe for plotting
forecast_df <- data.frame(
  Quarter = time(c_test),
  Actual = as.numeric(c_test),
  Forecast = as.numeric(best_forecast$mean),
  Lower_95 = as.numeric(best_forecast$lower[, "95%"]),
  Upper_95 = as.numeric(best_forecast$upper[, "95%"])
)
# Plot
ggplot(forecast_df, aes(x = Quarter)) +geom_line(aes(y = Actual, color = "Actual"),
                              linewidth = 1) +geom_line(aes(y = Forecast, color = "Forecast"), li
  geom_ribbon(aes(ymin = Lower_95, ymax = Upper_95), fill = "gray", alpha = 0.3) +
  labs(title = "Model 2.3: Consumption Forecast vs Actual (Next 7 Quarters)",
       x = "Quarter",y = "Consumption Change (%)",color = "Legend" ) +scale_color_manual(values =
  theme_minimal()
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



```
# Accuracy metrics
accuracy(best_forecast, c_test)
```

```
##                         ME       RMSE        MAE       MPE     MAPE      MASE
## Training set -0.005044404 0.32161031 0.24011665 6.3450835 77.90340 0.3673577
## Test set      0.014996586 0.09842901 0.07154606 0.7504596 10.04255 0.1094593
##                     ACF1 Theil's U
## Training set -0.09615351        NA
## Test set     -0.43382748 0.3223499
```

```
# selected model shows strong predictive performance,
#with low test RMSE (0.098) and MAPE (10%), indicating high accuracy.


# Compare forecasted and actual values:
library(zoo)
forecast_vals <- round(as.numeric(best_forecast$mean), 3)
actual_vals <- round(as.numeric(c_test), 3)
quarters <- as.yearqtr(time(c_test))  # Use readable quarter format
```

```r
# Combine into a data frame
comparison_df <- data.frame(
  Quarter = quarters,
  Forecast = forecast_vals,
  Actual = actual_vals,
  Error = round(forecast_vals - actual_vals, 3)
)
# View comparison
print(comparison_df)
```

```
##   Quarter Forecast Actual  Error
## 1 2015 Q1    0.612  0.596  0.016
## 2 2015 Q2    0.695  0.708 -0.013
## 3 2015 Q3    0.808  0.665  0.143
## 4 2015 Q4    0.461  0.562 -0.101
## 5 2016 Q1    0.432  0.405  0.027
## 6 2016 Q2    0.858  1.048 -0.190
## 7 2016 Q3    0.742  0.730  0.012
```

```r
#Conclusion:
#The comparison shows the forecasted and actual values for each quarter, along with the   error
```