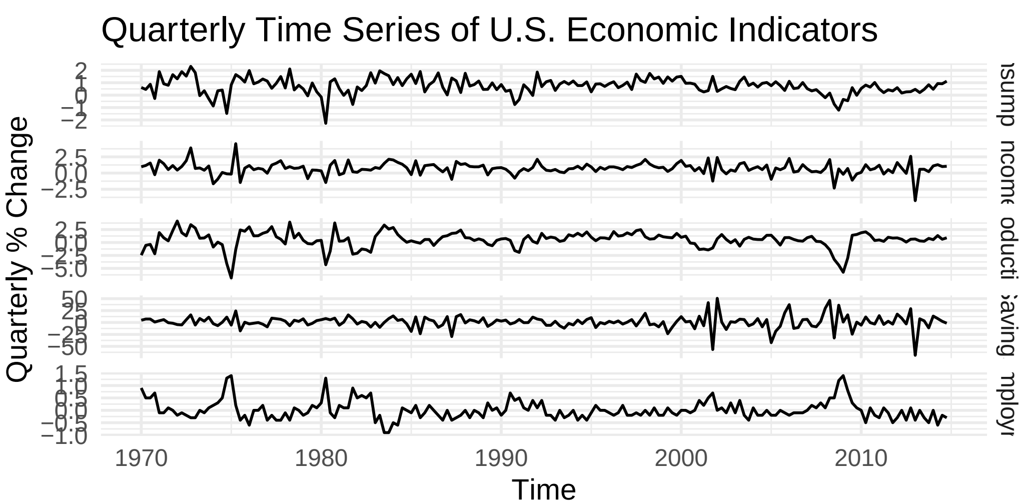
Abstract

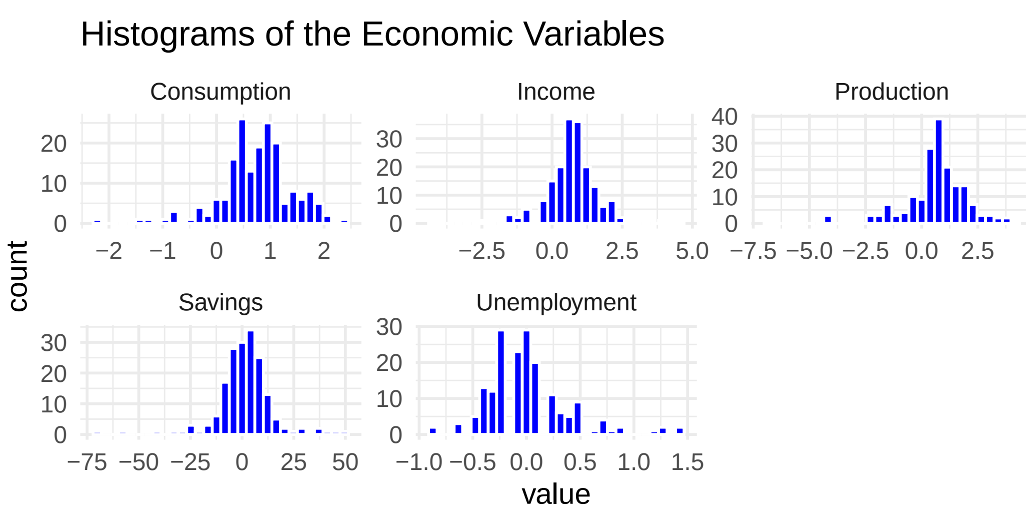
This report forecasts quarterly U.S. personal consumption expenditure for 2015 Q1 to 2016 Q3 using historical data from 1970 Q1 to 2014 Q4. The `uschange` dataset includes five economic indicators: Consumption, Income, Production, Savings, and Unemployment. I assessed their stationarity, distribution, patterns, and relationships to identify the most effective forecasting model.

Forecasting U.S. Consumption Using Economic Indicators

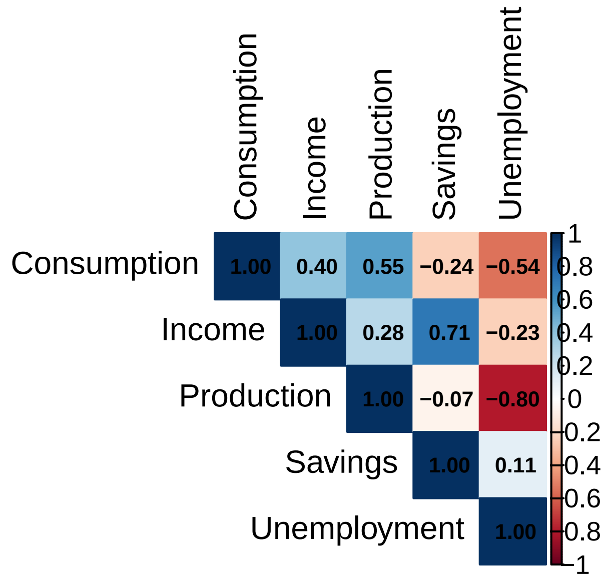
Data Exploration



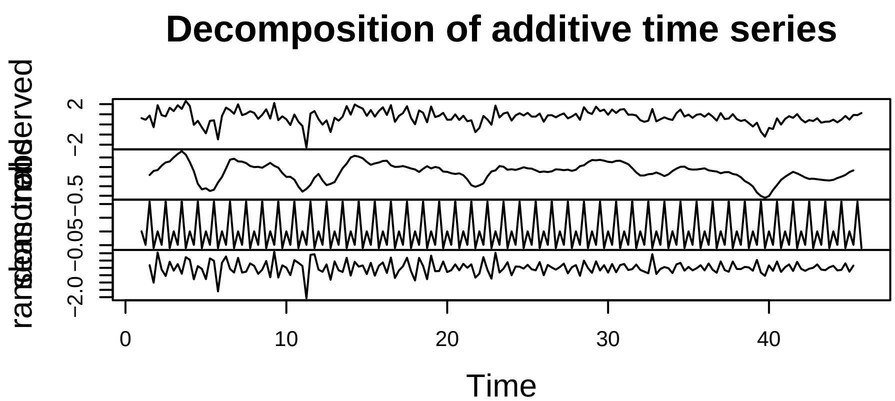
All five series show stable means over time. Consumption and Income show small fluctuations around zero, while Savings shows larger and sharper volatility, whereas Unemployment and Production show big peaks but were not as erratic as Savings.



All variables are roughly normally distributed, so no transformations were needed. Savings shows extreme values, indicating volatility. Consumption correlates most strongly with Production (r = 0.55) and Income (r = 0.40), and negatively with Unemployment (r = –0.54). Savings has a weak correlation (r = –0.24).



ADF and KPSS tests confirmed all variables are stationary, so no differencing was required.



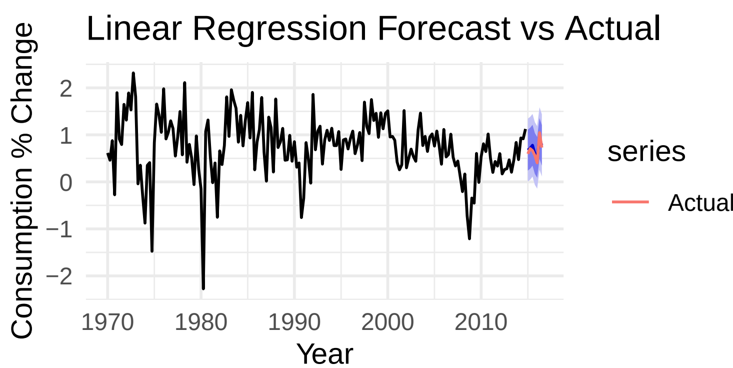
Decomposition of the Consumption series revealed a stable trend and minimal seasonality, indicating that seasonal effects are weak or absent. As a result, I ruled out models that rely on explicit seasonal components, such as Seasonal ARIMA, Holt-Winters Exponential Smoothing, and ETS models, which are more appropriate when strong seasonal patterns are present.

Model Selection and Forecasting

Autocorrelation analysis showed ACF spikes at lags 1–3 and PACF spikes at lags 1–2, suggesting an MA(3) and AR(1–2) structure. ARIMA(1,0,3) and (2,0,3) were tested, but auto.arima() selected ARIMA(3,0,0)(2,0,0)[4] with the lowest AIC (331.88). Residuals passed diagnostic checks (Ljung-Box p = 0.88), and the model achieved RMSE = 0.27 and Theil's U = 0.74.

A regression with ARIMA errors using all four predictors further improved fit, with auto.arima() selecting ARIMA(3,1,0)(1,0,0)[4] (AIC = 148.6, RMSE = 0.097, Theil's U = 0.34). Income had the strongest influence, and Savings,despite its volatility, contributed positively.

A simpler linear regression using the same predictors outperformed both, with RMSE = 0.087, MASE = 0.117, and Theil's U = 0.25. Residuals were normal and uncorrelated, making it the better model.



Conclusion

Three models were tested to forecast U.S. Consumption: ARIMA, regression with ARIMA errors, and linear regression. While ARIMA captured time-based patterns, the linear model had the best forecast accuracy and offered a strong mix of simplicity, interpretability, and a good performance. Predictors like Income and Production improved accuracy, with residuals meeting model assumptions. The linear regression model was chosen as the final model fit.

Strengths, Limitations and Potential Improvements

This project applied a thorough approach as it tested stationarity, distribution, autocorrelation, and variable relationships. Comparing ARIMA, regression with ARIMA errors, and linear regression showed that the linear model, despite not modelling residual structure, did the best due to strong, stationary predictors. While the regression-ARIMA model performed well, residuals showed slight autocorrelation. A short test period and the selection of seasonal terms by auto.arima() despite weak visual evidence suggest a possible limitation of overlooked seasonality. A potential improvement could be exploring lagged predictors.

[END of the REPORT]

R code

Recall that this section has no page limits, but you are encouraged to be parsimonious (privilege quality over quantity).

NOTE:

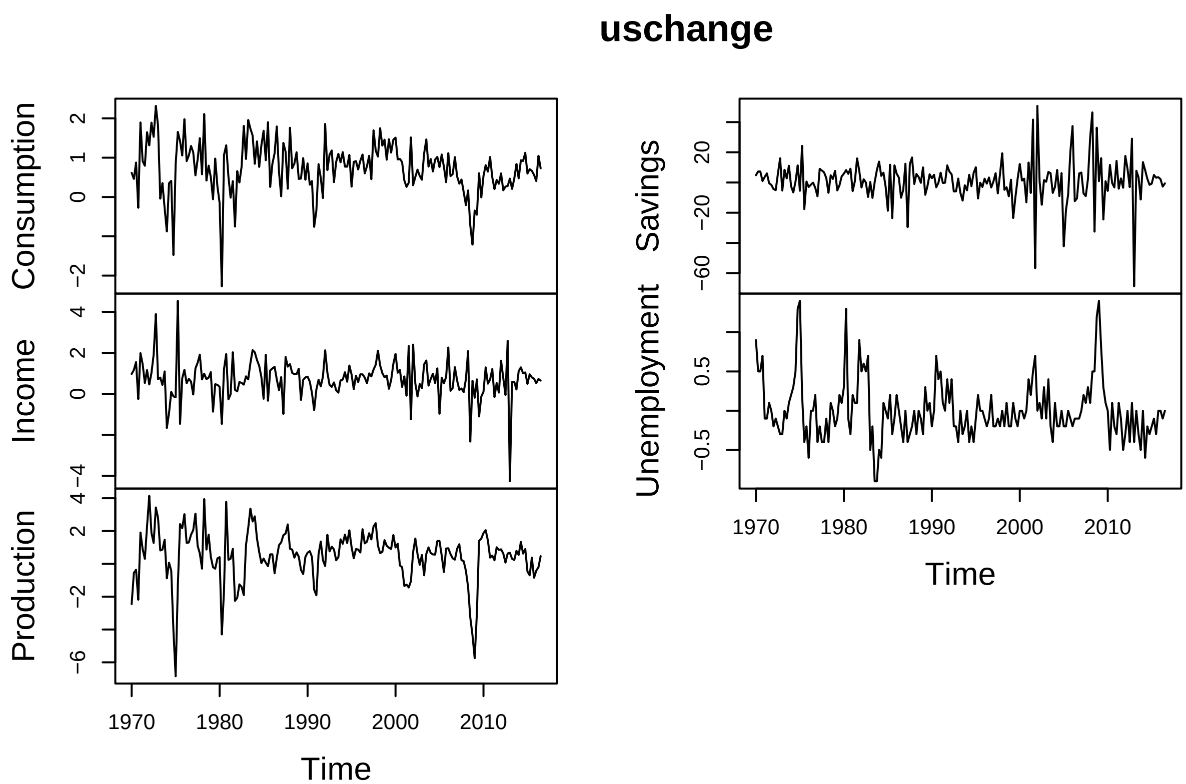
Some figures shown in the main report were generated from expanded versions of the code below. They were condensed in the report for space efficiency.

```
# LOAD REQUIRED PACKAGES
#libraries that are needed
library(forecast)
library(fpp2)
library(fma)
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.4.3
```

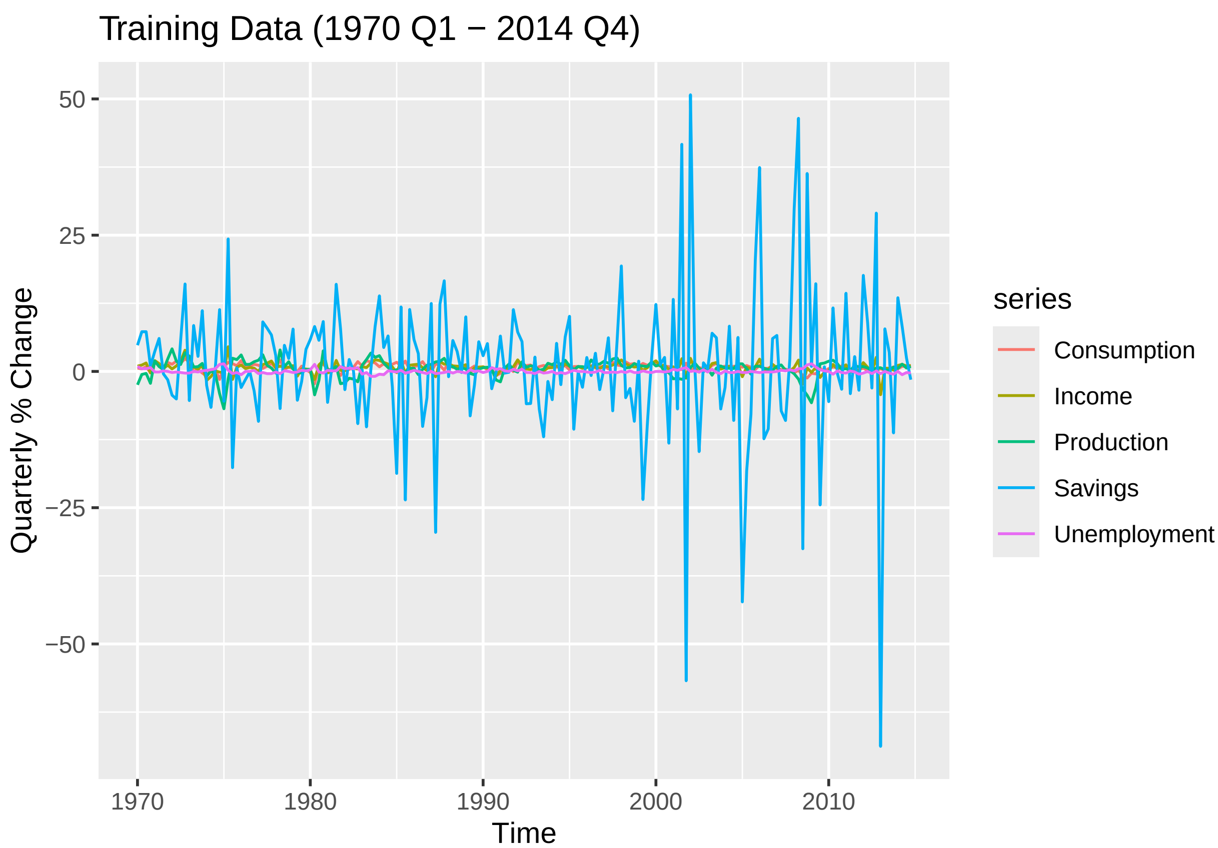
EXPLORING THE STRUCTURE OF THE DATASET

```
#shows the time series plots of all the variables together
plot(uschange)
```

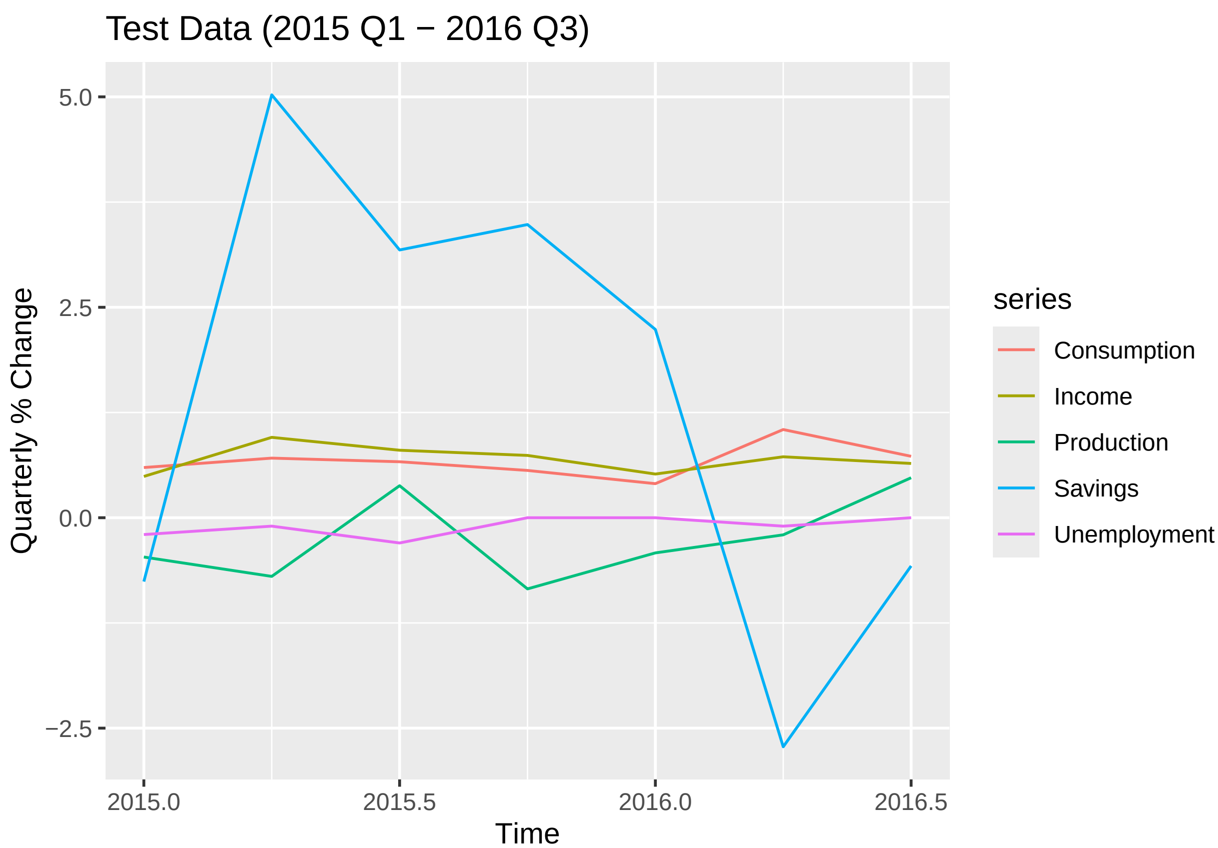


SPLITTING THE DATA INTO TRAINING AND TESTS

```
# Training set 1970 -2014, Q1 - Q4
training <- window(uschange, end = c(2014, 4))
# Test set 2015 Q1 - 2016 Q3
test <- window(uschange, start = c(2015, 1))

# Visualising the training and test splits
autoplot(training) +
  ggtitle("Training Data (1970 Q1 - 2014 Q4)") +
  ylab("Quarterly % Change") +
  xlab("Time")
```




```
autoplot(test) +
  ggtitle("Test Data (2015 Q1 - 2016 Q3)") +
  ylab("Quarterly % Change") +
  xlab("Time")
```



STATISTICS AND CORRELATION

```r
#min, max, mean, quartiles for each variable
summary(training)
```

```
##   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
```

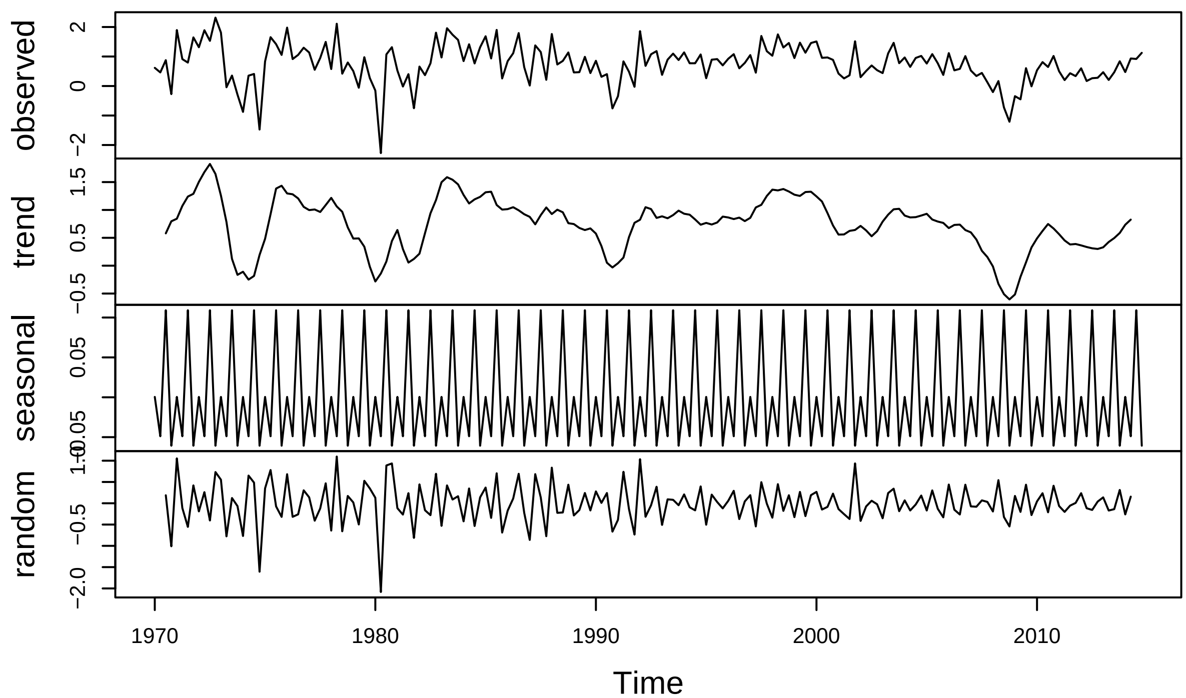
```r
#Identifies which variables are good predictors through a correlation matrix
cor(training)
```

```
##              Consumption     Income  Production     Savings Unemployment
## Consumption    1.0000000  0.3989102  0.54879015 -0.23931232   -0.5439411
## Income         0.3989102  1.0000000  0.27944720  0.71379623   -0.2312733
## Production     0.5487901  0.2794472  1.00000000 -0.06811485   -0.7989555
## Savings       -0.2393123  0.7137962 -0.06811485  1.00000000    0.1109808
## Unemployment  -0.5439411 -0.2312733 -0.79895549  0.11098078    1.0000000
```

Findings: The correlation matrix shows that Consumption has the strongest positive relationship with Production (0.55) and an okay relationship with Income (0.40). Unemployment is negatively correlated (–0.54)but strong, Savings shows only a weak negative correlation with Consumption (–0.24), which I can infer is due to its high volatility, and changing variance. Based on this, Production and Income are strong candidate predictors for modeling Consumption, and Unemployment and Savings may also be considered. *DECOMPOSE CONSUMPTION SERIES*

```r
#converts to a time series object
consumption_ts <- ts(training[,"Consumption"], frequency = 4, start = c(1970, 1))
# Decompose into trend seasonal and remainder
consumption_decomp <- decompose(consumption_ts)
#visualises it
plot(consumption_decomp)
```

Decomposition of additive time series



STATIONARY TEST - CONSUMPTION SERIES

```r
adf.test(consumption_ts)
```

```
## Warning in adf.test(consumption_ts): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  consumption_ts
## Dickey-Fuller = -4.4219, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(consumption_ts)
```

```
## Warning in kpss.test(consumption_ts): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  consumption_ts
## KPSS Level = 0.27652, Truncation lag parameter = 4, p-value = 0.1
```

```r
# checks remainder from decomposition
adf.test(na.omit(consumption_decomp$random))
```

```
## Warning in adf.test(na.omit(consumption_decomp$random)): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(consumption_decomp$random)
## Dickey-Fuller = -9.7551, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

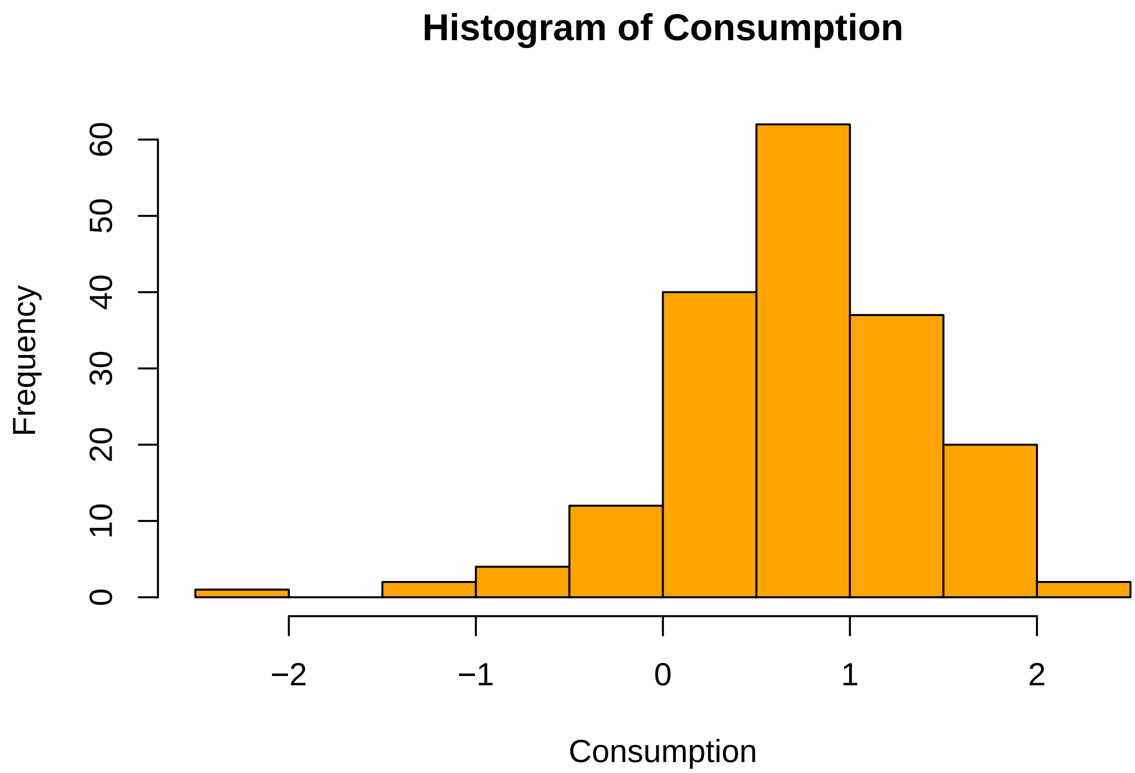
```r
kpss.test(na.omit(consumption_decomp$random))
```

```
## Warning in kpss.test(na.omit(consumption_decomp$random)): p-value greater than
## printed p-value
```

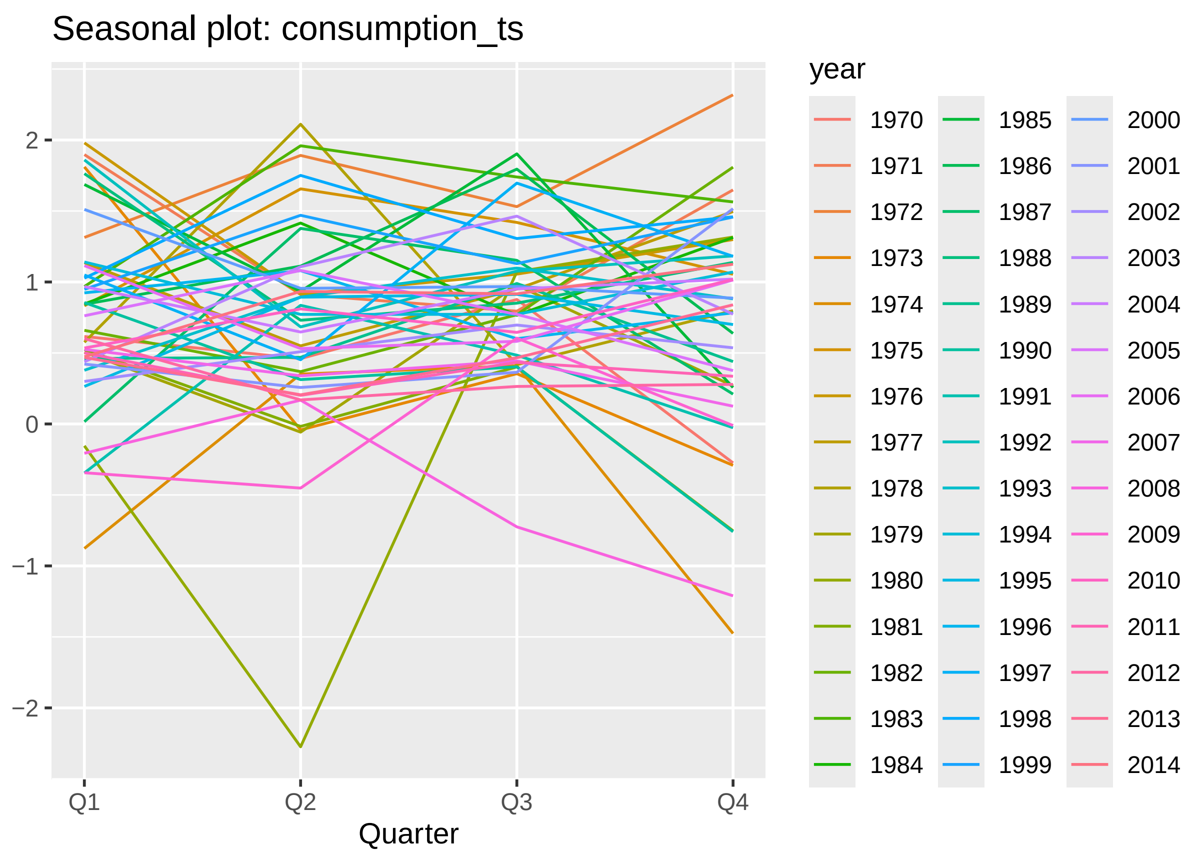
```
##
##  KPSS Test for Level Stationarity
##
## data:  na.omit(consumption_decomp$random)
## KPSS Level = 0.013127, Truncation lag parameter = 4, p-value = 0.1
```

DISTRIBUTION AND PATTERNS OF CONSUMPTION

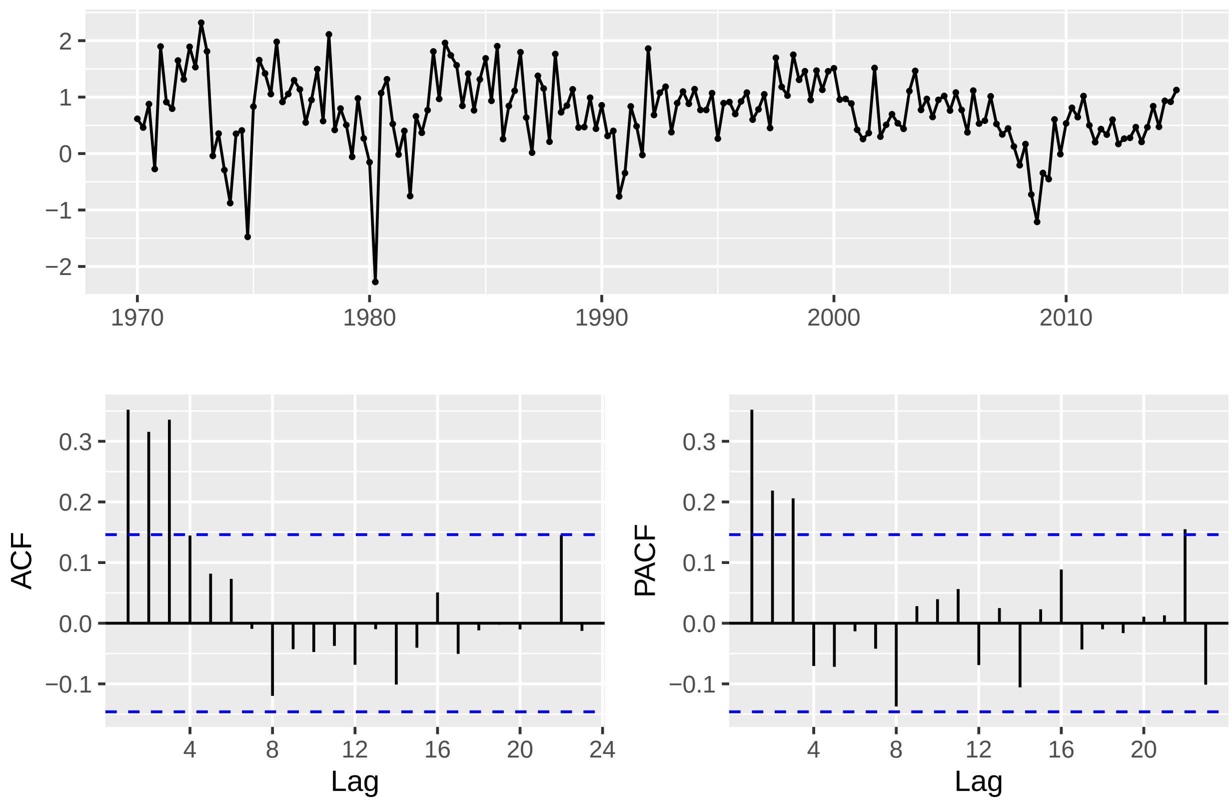
```r
#historgram to see distribution
hist(consumption_ts, col = "orange", main = "Histogram of Consumption", xlab = "Consumpt
```




```
#Seasonal patterns
ggseasonplot(consumption_ts)
```



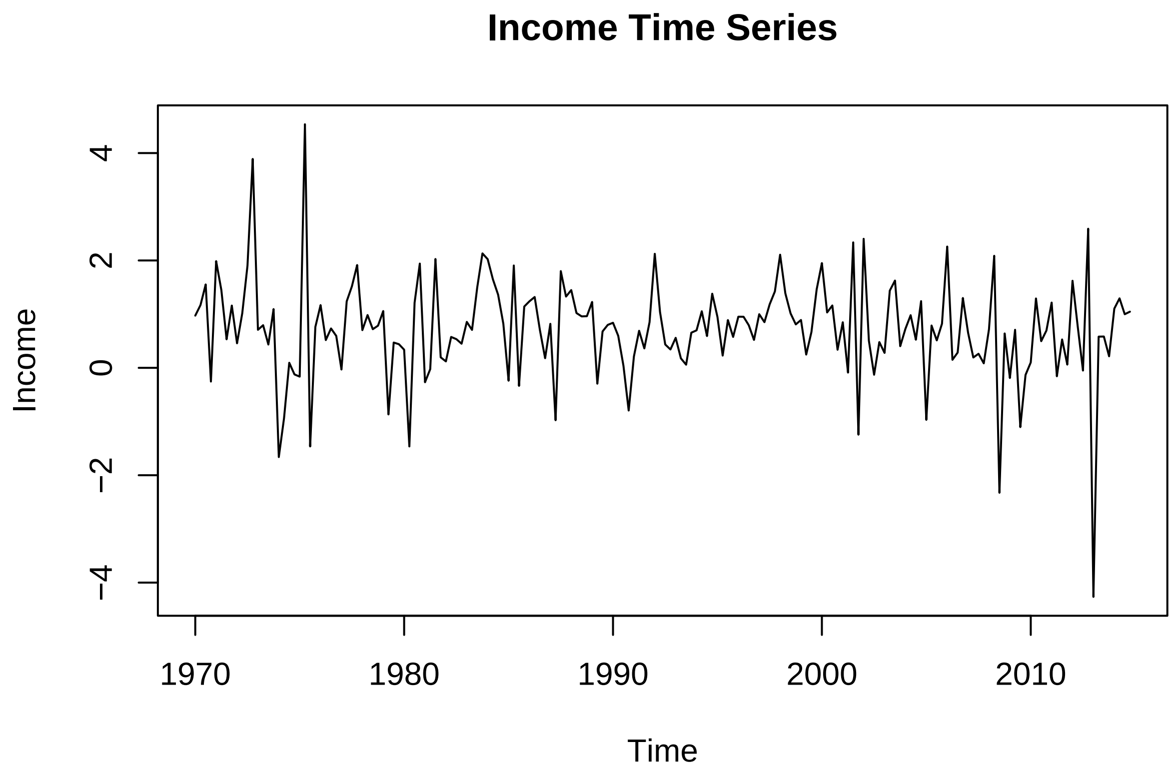

```
#ACF and PACF plots
ggtsdisplay(consumption_ts)
```



Findings: The result shows high variability across years, with no consistent or repeating pattern across quarters. This supports the conclusion from the decomposition that the series does not exhibit any strong seasonal effects. Therefore, seasonal ARIMA terms are unlikely to be needed for modelling.

The ACF plot of the Consumption series shows strong autocorrelation at lag 1 and weaker but still significant autocorrelation at lags 2 and 3. The sharp decay after lag 3 suggests a moving average component of order 3. These patterns indicate that an ARIMA(1,0,2) or ARIMA(1,0,3) model could be suitable for capturing the autocorrelation structure in the data. I have seen that consumption time series is stationary therefore d = 0 as we do not need to apply differencing to the series. *EDA FOR REST OF VARIABLES*

```
# Income
income_ts <- ts(training[,"Income"], start = c(1970, 1), frequency = 4)
plot(income_ts, main = "Income Time Series", ylab = "Income", xlab = "Time")
```



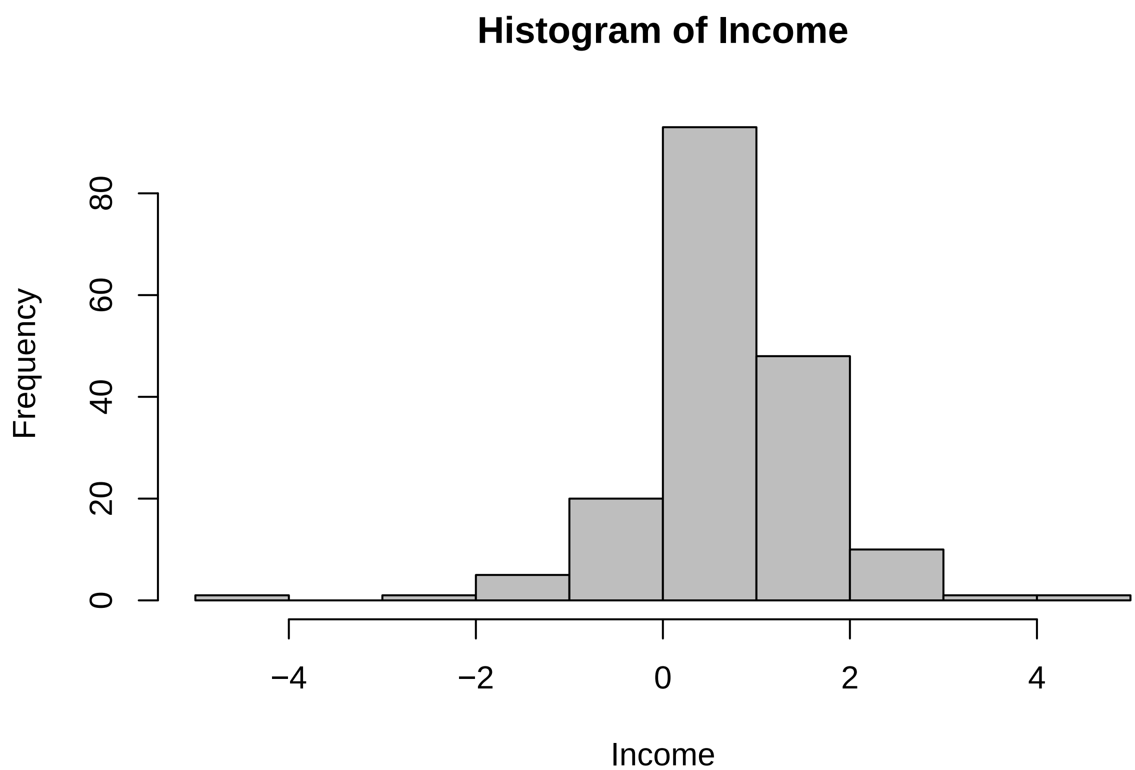

```r
#stationary tests
adf.test(income_ts)
```

```
## Warning in adf.test(income_ts): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  income_ts
## Dickey-Fuller = -6.0046, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

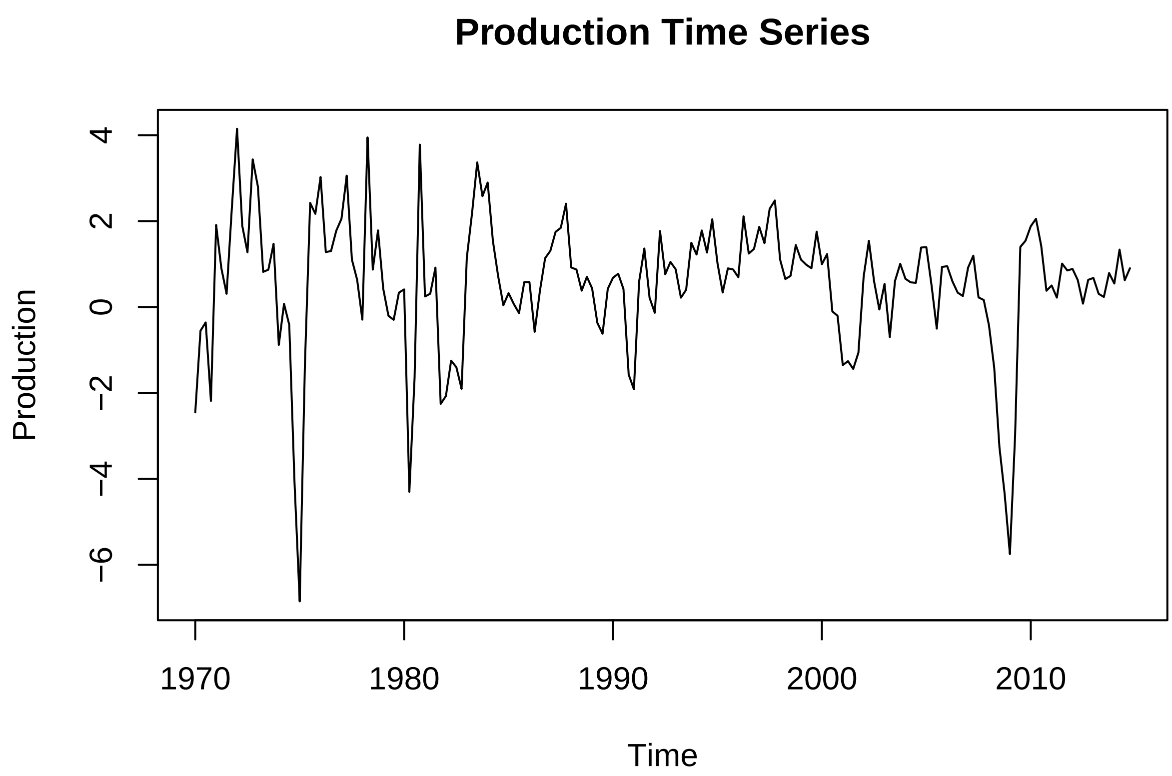
```r
kpss.test(income_ts)
```

```
## Warning in kpss.test(income_ts): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  income_ts
## KPSS Level = 0.23245, Truncation lag parameter = 4, p-value = 0.1
```

```r
#historgram to see distribution
hist(income_ts, col = "grey", main = "Histogram of Income", xlab = "Income")
```




```r
# Production
production_ts <- ts(training[,"Production"], start = c(1970, 1), frequency = 4)
plot(production_ts, main = "Production Time Series", ylab = "Production", xlab = "Time")
```



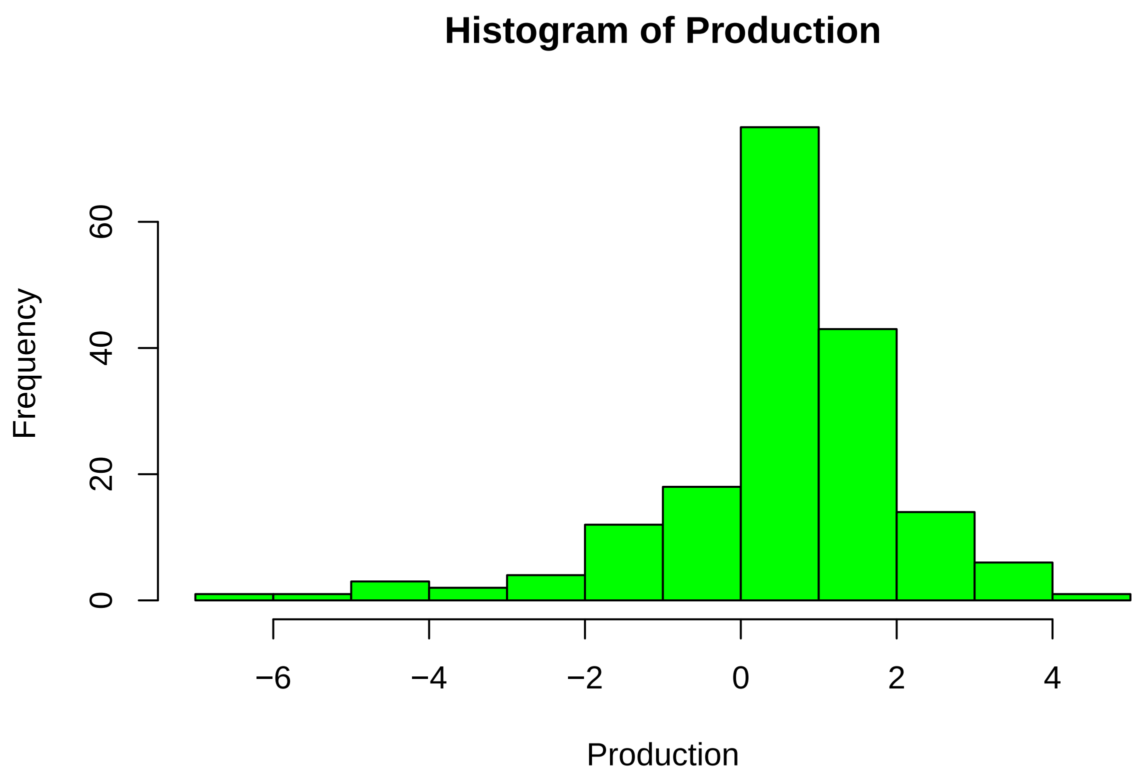

```r
#stationary tests
adf.test(production_ts)
```

```
## Warning in adf.test(production_ts): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  production_ts
## Dickey-Fuller = -5.1571, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

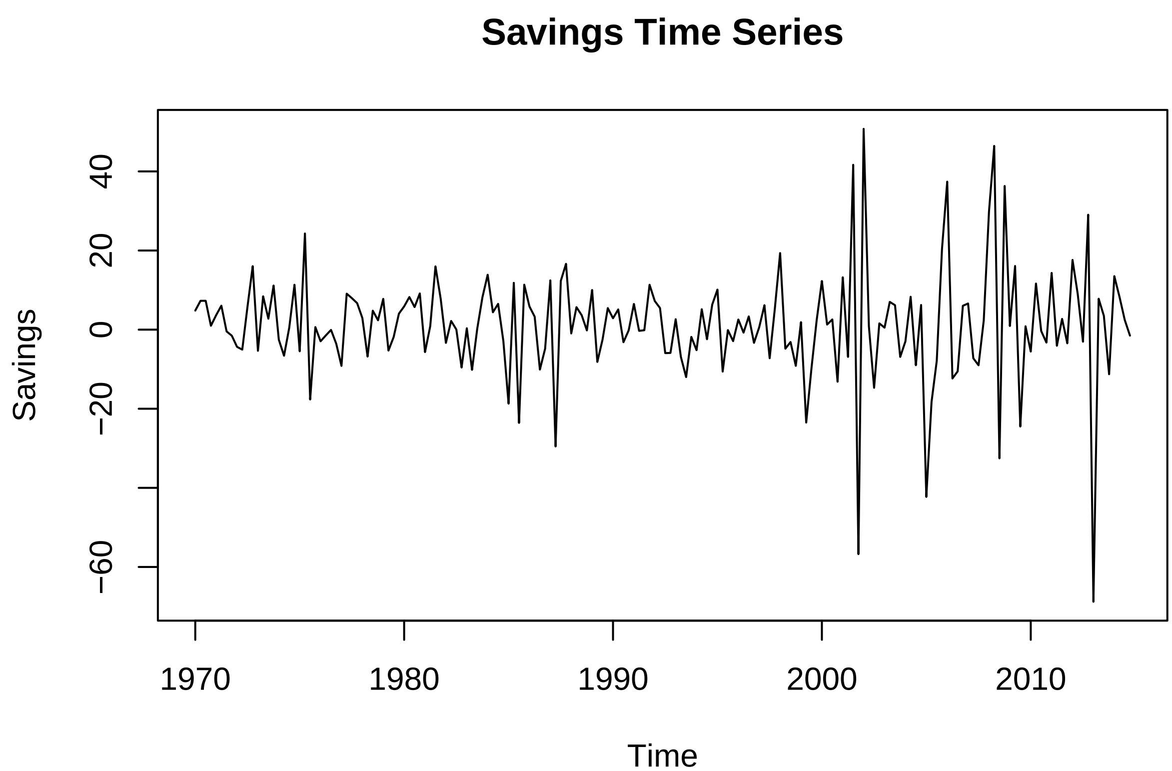
```r
kpss.test(production_ts)
```

```
## Warning in kpss.test(production_ts): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  production_ts
## KPSS Level = 0.07317, Truncation lag parameter = 4, p-value = 0.1
```

```r
#histogram to see distribution
hist(production_ts, col = "green", main = "Histogram of Production", xlab = "Production"
```




```r
# Savings
savings_ts <- ts(training[,"Savings"], start = c(1970, 1), frequency = 4)
plot(savings_ts, main = "Savings Time Series", ylab = "Savings", xlab = "Time")
```



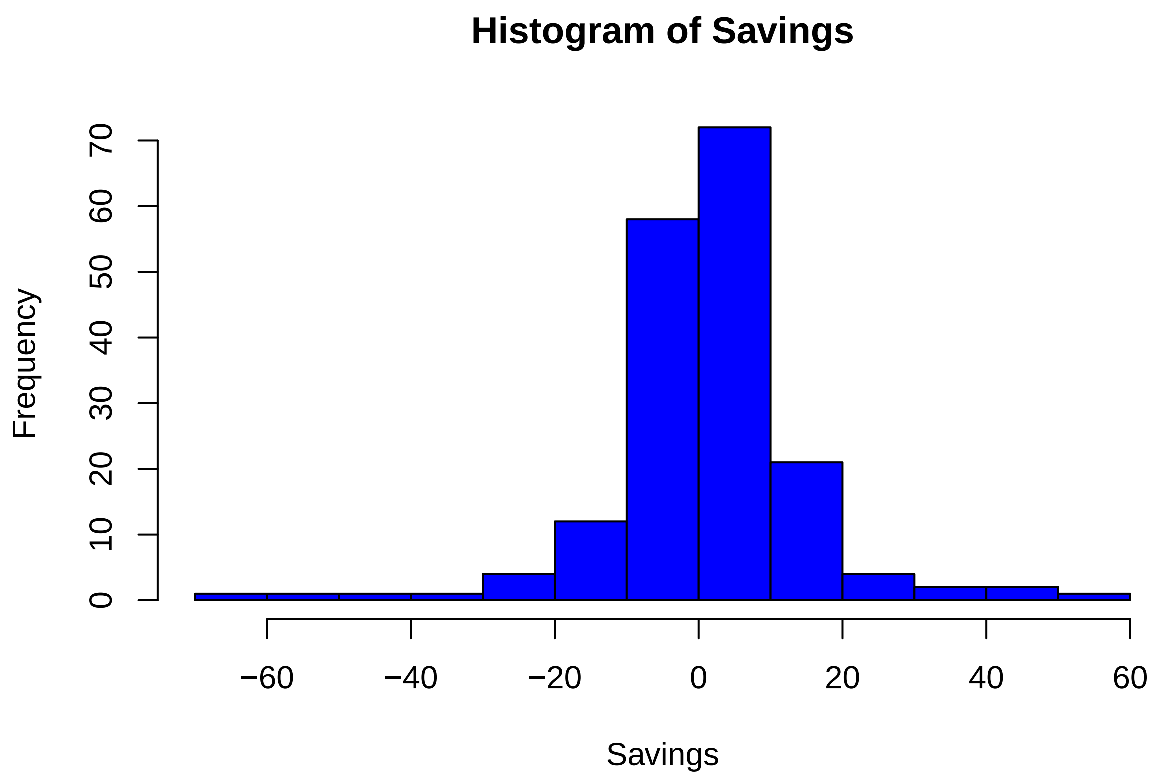

```r
#stationary tests
adf.test(savings_ts)
```

```
## Warning in adf.test(savings_ts): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  savings_ts
## Dickey-Fuller = -6.7917, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

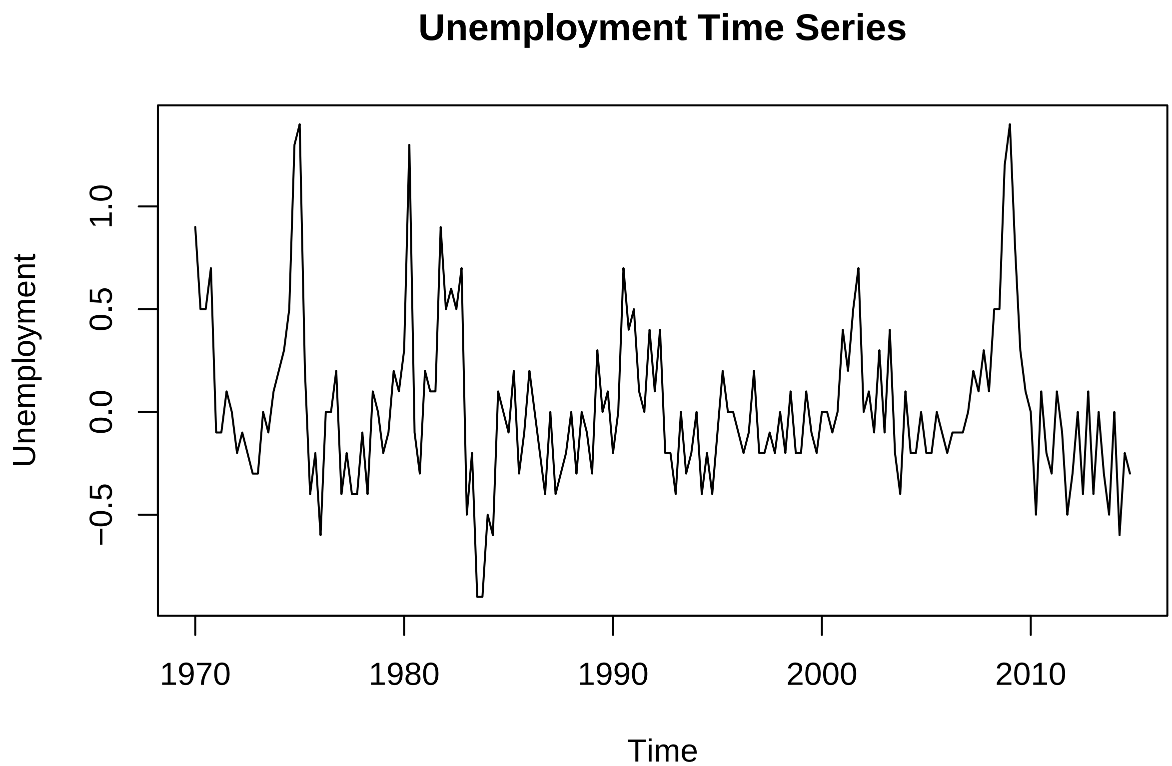
```r
kpss.test(savings_ts)
```

```
## Warning in kpss.test(savings_ts): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  savings_ts
## KPSS Level = 0.060873, Truncation lag parameter = 4, p-value = 0.1
```

```r
#histogram to see distribution
hist(savings_ts, col = "blue", main = "Histogram of Savings", xlab = "Savings")
```




```r
# Unemployment
unemployment_ts <- ts(training[,"Unemployment"], start = c(1970, 1), frequency = 4)
plot(unemployment_ts, main = "Unemployment Time Series", ylab = "Unemployment", xlab = "
```



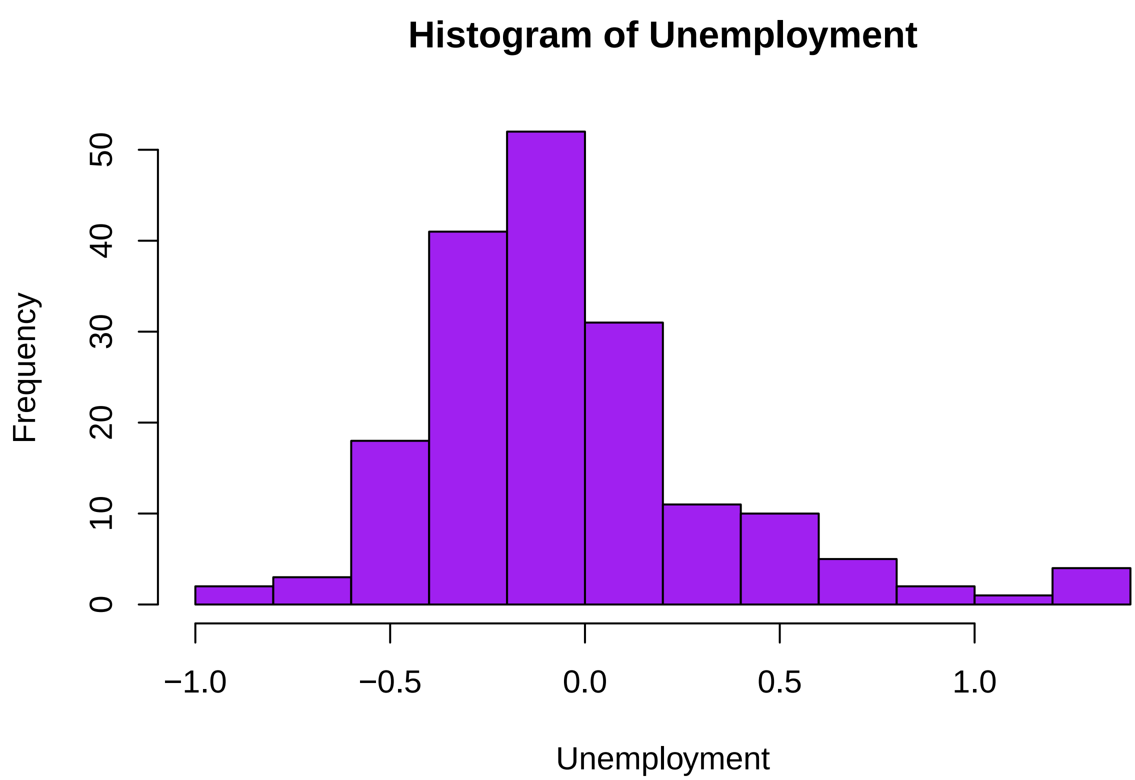

```r
#test for stationary
adf.test(unemployment_ts)
```

```
## Warning in adf.test(unemployment_ts): p-value smaller than printed p-value

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

```r
kpss.test(unemployment_ts)
```

```
## Warning in kpss.test(unemployment_ts): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  unemployment_ts
## KPSS Level = 0.095659, Truncation lag parameter = 4, p-value = 0.1
```

```r
#histogram
hist(unemployment_ts, col = "purple", main = "Histogram of Unemployment", xlab = "Unempl
```



Findings: Stationary was tested using the Augmented Dickey-Fuller (ADF) and KPSS tests, both confirming that all series were stationary at the 5% level. Histograms supported that most variables were evenly and well distributed around the centre. *Statistical Model 1 TEST: ARIMA on Consumption Only*

Rationale: An ARIMA model was selected to see the autocorrelation in the Consumption series, which appeared stationary and showed no strong seasonal patterns but showed some autocorrelation in its ACF and PACF plots.

```r
#Selecting ARIMA model, doing auto, and manual ones I have selected
arima_auto <- auto.arima(consumption_ts)
arima_manual <- Arima(consumption_ts, order = c(1, 0, 3))
arima_manual2 <- Arima(consumption_ts, order = c(2, 0, 3))

# Comparing AIC values
AIC(arima_auto); AIC(arima_manual); AIC(arima_manual2)
```

```
## [1] 331.8758
```

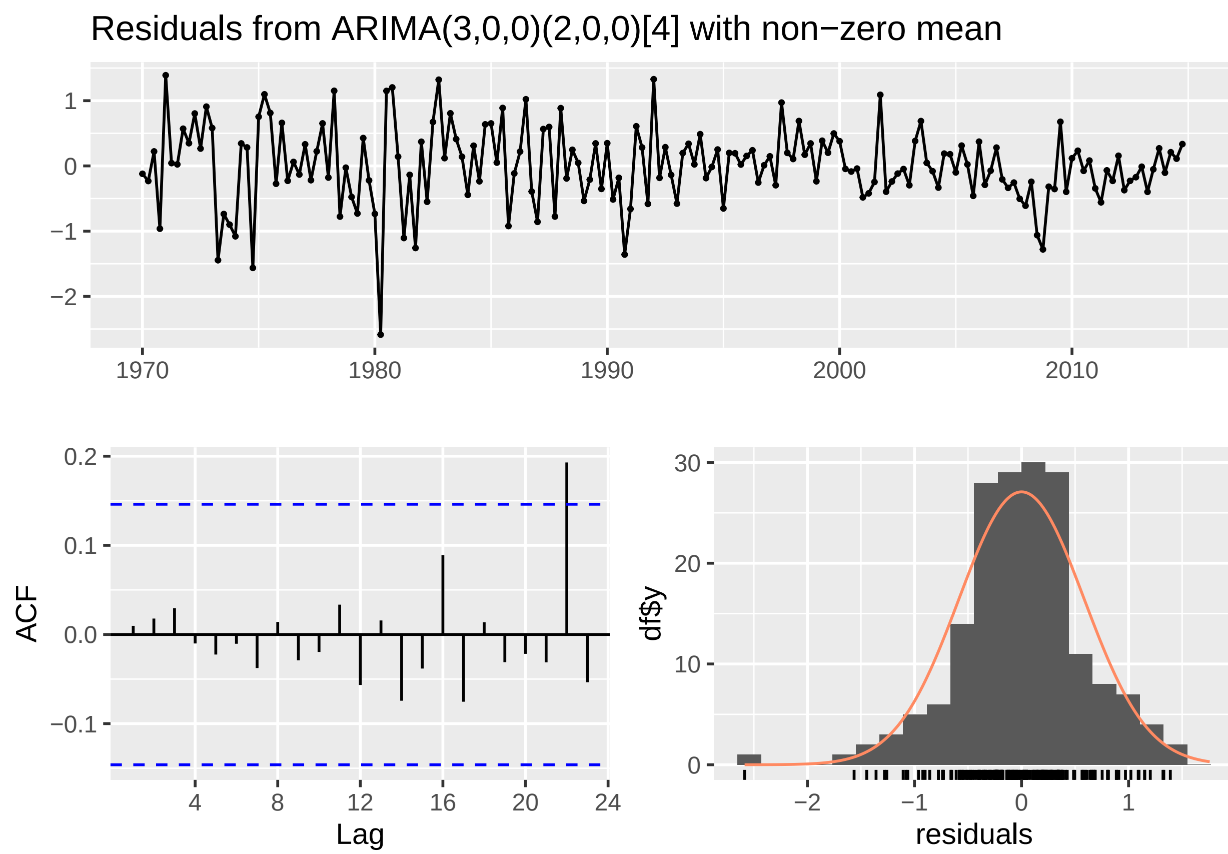
```
## [1] 335.0114
```

```
## [1] 336.5205
```

```r
# Checking the coefficients and fit statistics
summary(arima_auto)
```

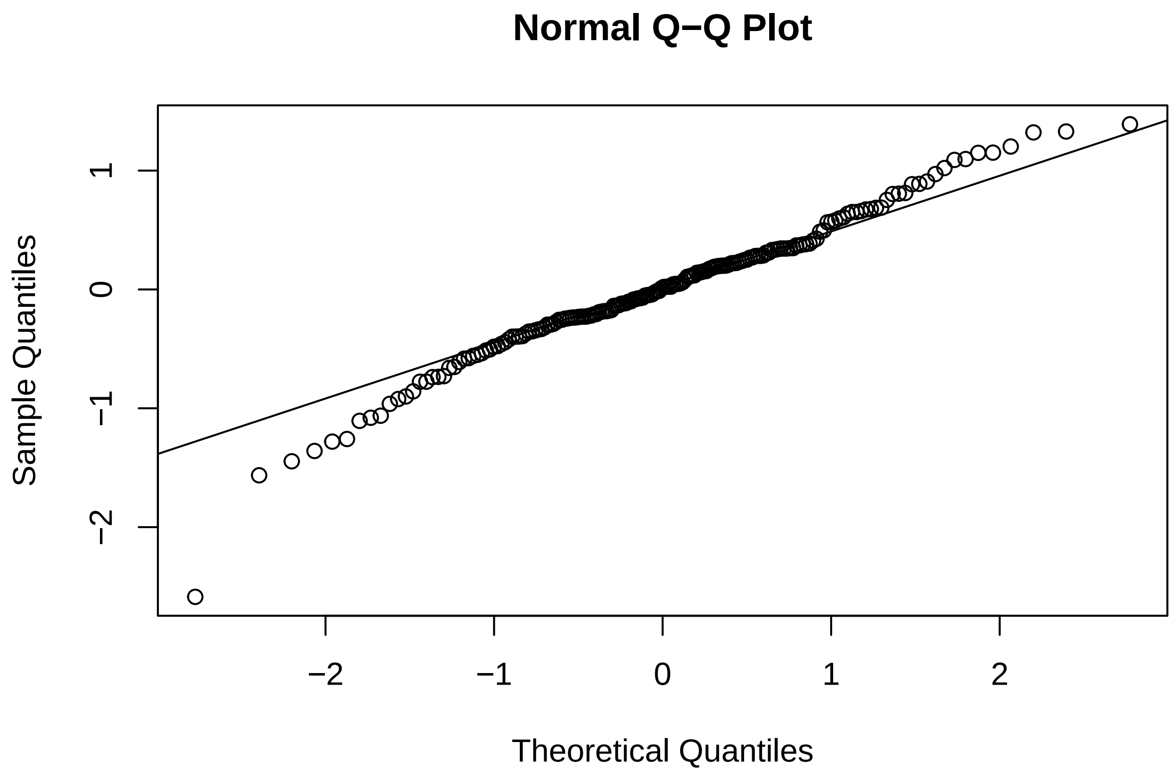
```
## Series: consumption_ts
## ARIMA(3,0,0)(2,0,0)[4] with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3     sar1     sar2    mean
##       0.2271  0.1777  0.2200  -0.0334  -0.1803  0.7522
## s.e.  0.0737  0.0736  0.0724   0.0772   0.0744  0.0946
##
## sigma^2 = 0.353:  log likelihood = -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
# Residual plots and Ljung-Box test
checkresiduals(arima_auto)
```

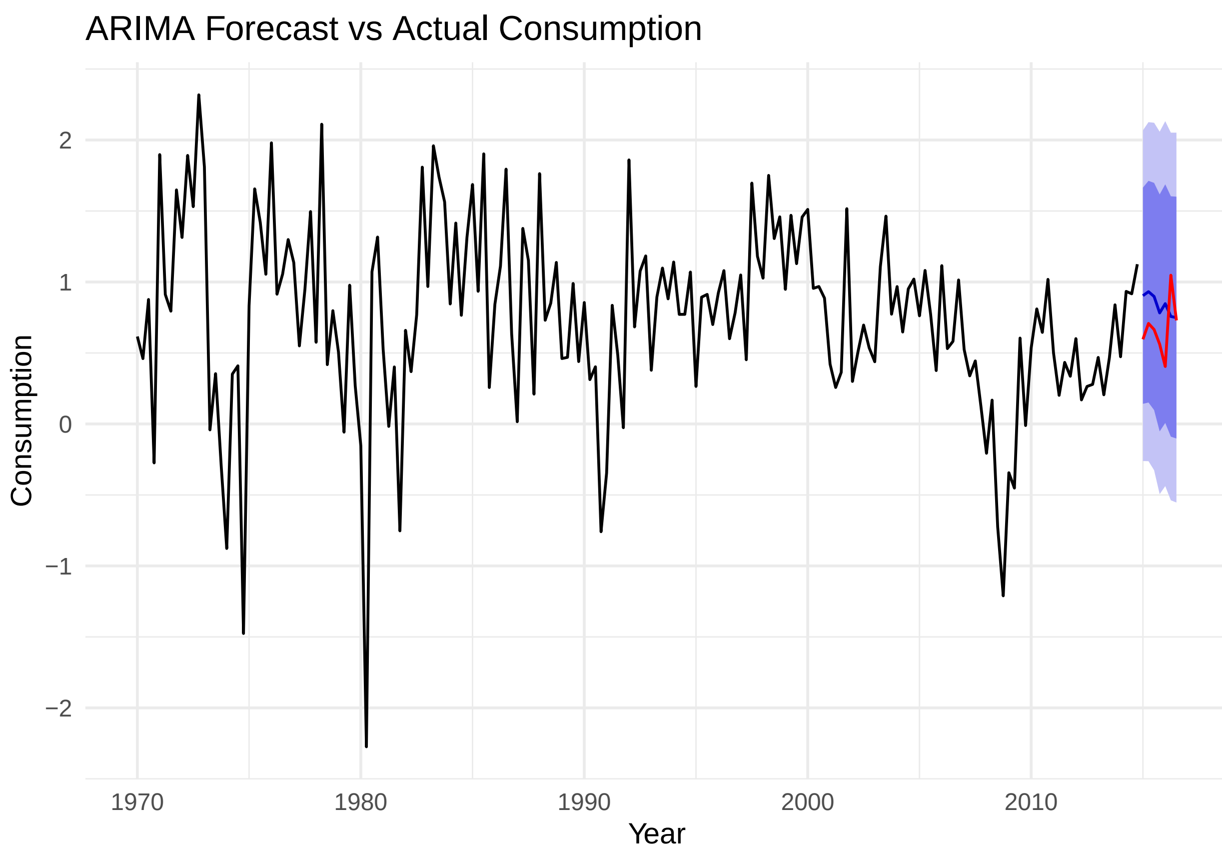



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0)(2,0,0)[4] with non-zero mean
## Q* = 0.67677, df = 3, p-value = 0.8787
##
## Model df: 5.   Total lags used: 8
```

```
# Q-Q plot of residuals
qqnorm(residuals(arima_auto))
qqline(residuals(arima_auto))
```




```r
# Forecasting the chosen ARIMA model (auto)
fcast_arima <- forecast(arima_auto, h = 7)
autoplot(fcast_arima) +
  autolayer(test[,"Consumption"], series = "Actual", color = "red") +
  ggtitle("ARIMA Forecast vs Actual Consumption") +
  ylab("Consumption") +
  xlab("Year") +
  theme_minimal()
```




```r
# The forecast accuracy metrics
accuracy(fcast_arima, test[,"Consumption"])
```

```
##                        ME      RMSE       MAE       MPE      MAPE      MASE
## Training set  0.0002895183 0.5841338 0.4391314  65.53638 188.53118 0.6718330
## Test set     -0.1651797648 0.2746917 0.2481962 -34.54071  42.46434 0.3797188
##                      ACF1 Theil's U
## Training set  0.009685967        NA
## Test set     -0.085512024 0.7365495
```

Findings: I selected ARIMA(1,0,3) and ARIMA(2,0,3) as manual candidate models, avoiding ARIMA(3,0,3) to reduce the risk of overfitting with too many high-order terms. These will be compared against a model selected by auto.arima()

The auto.arima() came out as the best model as its AIC was lower than the other 2 I tested. The function selected an ARIMA(3,0,0)(2,0,0)[4] model with a non-zero mean. Although visual inspection did not reveal clear seasonal patterns, the auto.arima() function selected a model with seasonal autoregressive terms.

Coefficients were all statistically significant and stable. The training residuals showed no autocorrelation, indicating that the model fits well. This model outperformed manual ARIMA alternatives based on AIC and is selected as model to compare to other types of statistical models. ## Statistical Model 2: Regression with ARIMA Errors *Rationale:* A regression model with ARIMA errors was implemented to evaluate whether including external predictors while also modeling autocorrellated residuals would improve the predictive performance.

```r
# Trying different combinations of predictors
fit_m2a <- auto.arima(consumption_ts, xreg = cbind(income_ts, production_ts))
fit_m2b <- auto.arima(consumption_ts, xreg = cbind(income_ts, production_ts, unemploymen
fit_m2c <- auto.arima(consumption_ts, xreg = cbind(income_ts, production_ts, unemploymen

# Comparing model fit
AIC(fit_m2a); BIC(fit_m2a)
```

```
## [1] 282.5582

## [1] 304.9088
```

```r
AIC(fit_m2b); BIC(fit_m2b)
```

```
## [1] 281.9414

## [1] 304.2531
```

```r
AIC(fit_m2c); BIC(fit_m2c)
```

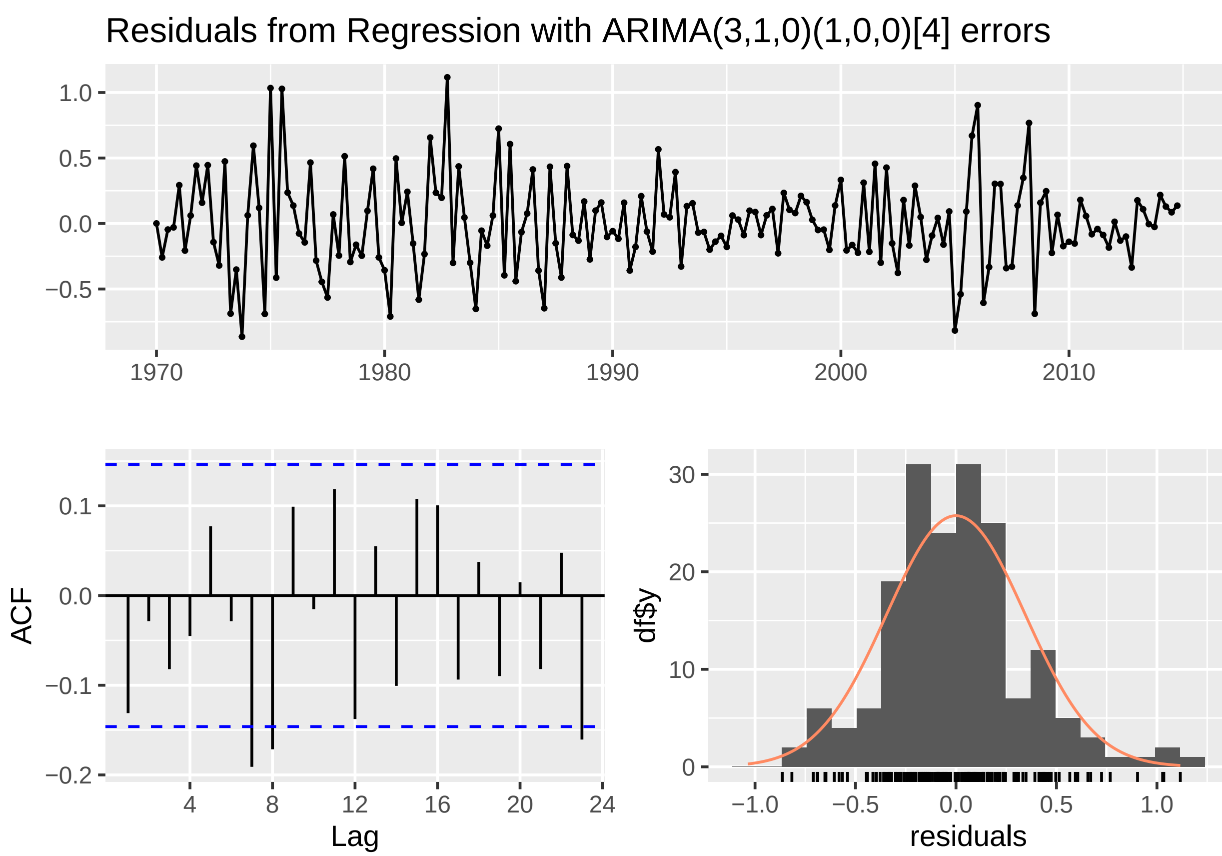
```
## [1] 148.5681

## [1] 180.442
```

```r
summary(fit_m2c)
```

```
## Series: consumption_ts
## Regression with ARIMA(3,1,0)(1,0,0)[4] errors
##
## Coefficients:
##           ar1      ar2      ar3     sar1    drift  income_ts  production_ts
##       -0.8047  -0.5422  -0.5286  -0.4607  -0.0008     0.6963         0.0326
## s.e.   0.0671   0.0796   0.0808   0.0793   0.0062     0.0452         0.0270
##       unemployment_ts  savings_ts
##               -0.2667     -0.0449
## s.e.           0.1140      0.0031
##
```

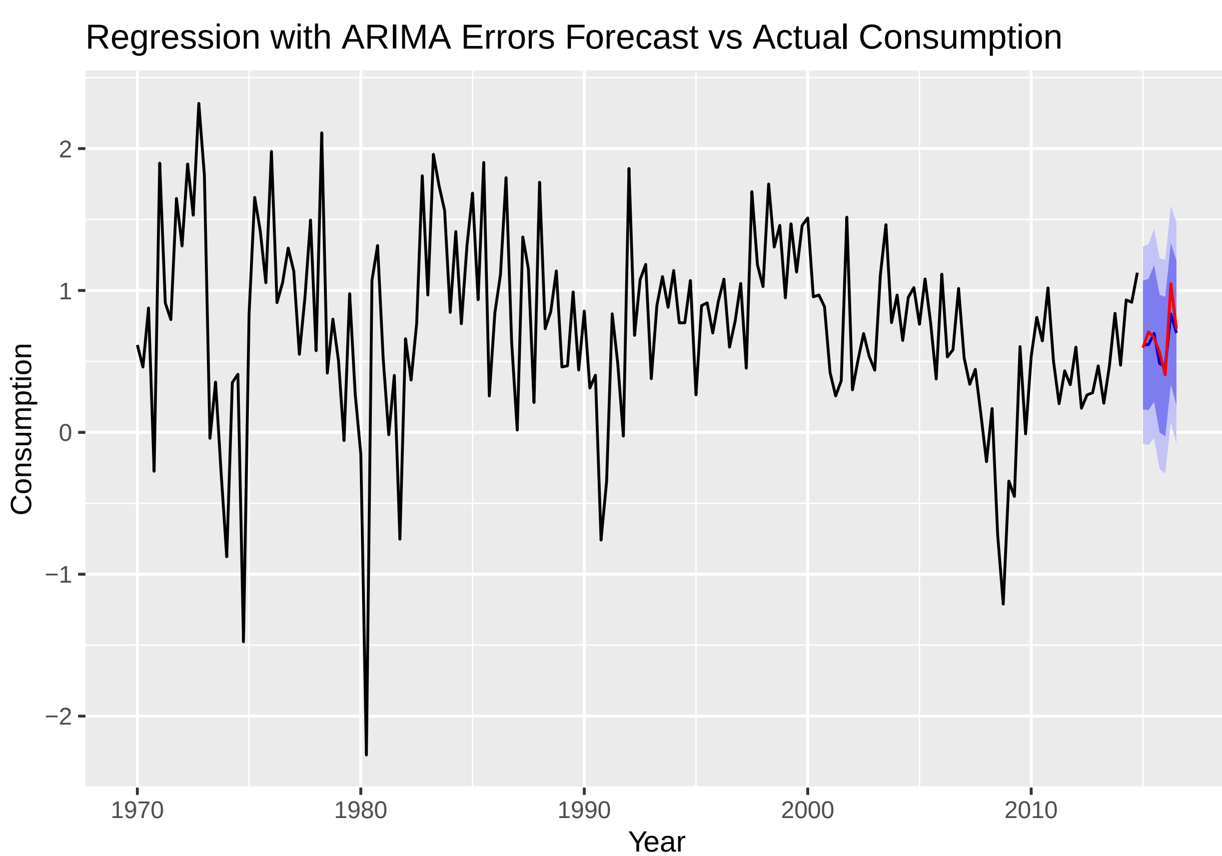
```
## sigma^2 = 0.1255:  log likelihood = -64.28
## AIC=148.57   AICc=149.88   BIC=180.44
##
## Training set error measures:
##                         ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.0004764331 0.3442272 0.2633871 6.178704 92.59988 0.4029595
##                    ACF1
## Training set -0.1310828
```

```
checkresiduals(fit_m2c)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,1,0)(1,0,0)[4] errors
## Q* = 18.69, df = 4, p-value = 0.0009042
##
## Model df: 4.   Total lags used: 8
```

```r
# Forecast from best regression ARIMA model picked based on AIC/BIC
future_xreg <- cbind(
  income_ts = test[,"Income"],
  production_ts = test[,"Production"],
  unemployment_ts = test[,"Unemployment"],
  savings_ts = test[,"Savings"]
)

forecast_m2c <- forecast(fit_m2c, xreg = future_xreg)

autoplot(forecast_m2c) +
  autolayer(test[,"Consumption"], series = "Actual", color = "red") +
  ggtitle("Regression with ARIMA Errors Forecast vs Actual Consumption") +
  ylab("Consumption") +
  xlab("Year")
```




```r
accuracy(forecast_m2c, test[,"Consumption"])
```

```
##                         ME       RMSE        MAE      MPE     MAPE      MASE
## Training set -0.0004764331 0.34422720 0.26338713 6.178704 92.59988 0.4029595
```

```
## Test set      0.0427189688 0.09658557 0.07448284 4.022171 10.52587 0.1139523
##                     ACF1 Theil's U
## Training set -0.1310828        NA
## Test set     -0.6131077 0.3446035
```

Findings: Model 2c (with all 4 predictors) has by far the lowest AIC and BIC, Unemployment and Savings do appear to contribute to forecasting Consumption. Especially adding Savings, despite its weak correlation, significantly improved model fit. The sharp AIC/BIC drop for model 2c justifies its inclusion.

The residuals are centered around zero with a constant variance. The histogram closely matches a normal distribution, suggesting residuals are okay. The ACF plot shows most autocorrolations fall within the confidence bands — but a few are close to the edge. Since p-value < 0.05 in the L box test, this indicates that some autocorrelation remains in the residuals meaning residuals may not be pure white noise. So despite good AIC/BIC performance the model has slightly concerning autocorrelation in residuals.

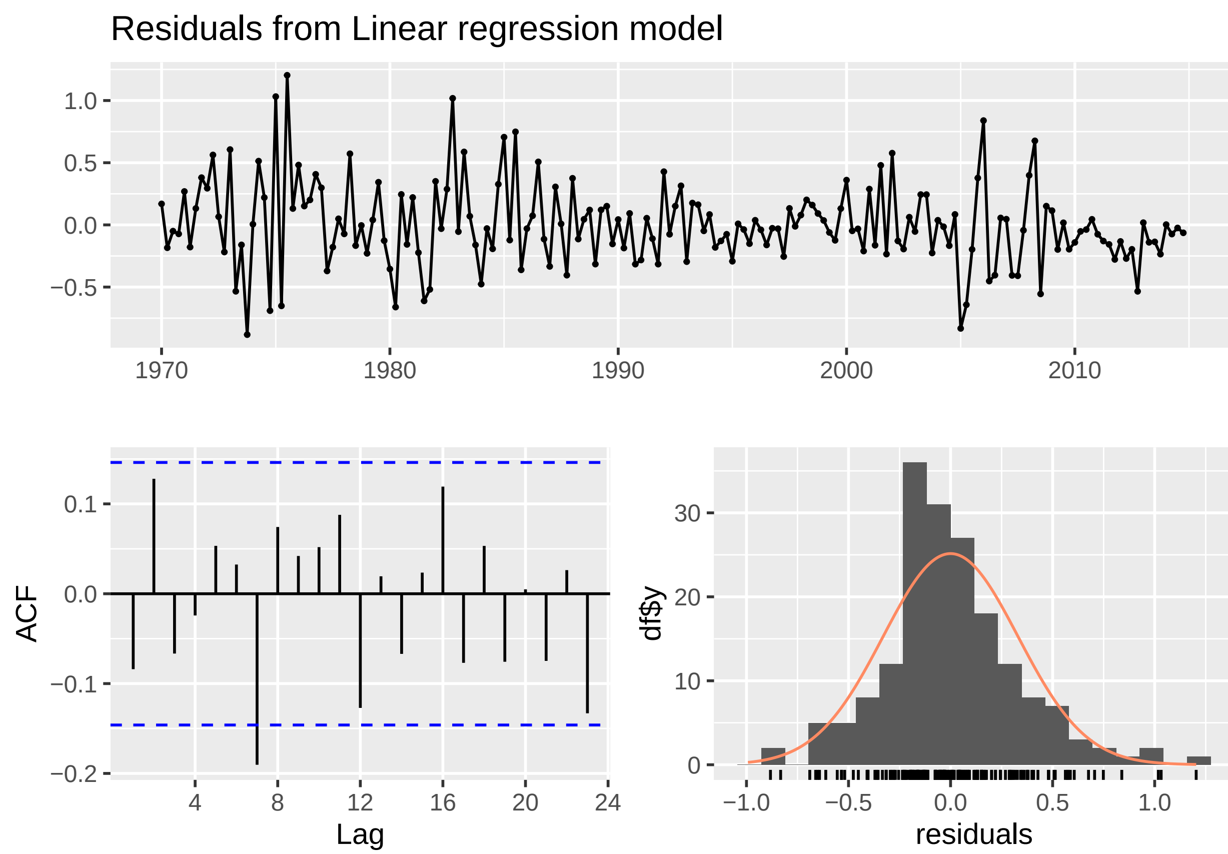
The regression model with ARIMA(3,1,0)(1,0,0)[4] errors shows a significantly better fit than the ARIMA-only model, with a much lower AIC (148.57 vs. 331.88) and lower training RMSE (0.34 vs. 0.58).

The forecast lines closely follow the actual red line with narrow confidence bands

Statistical Model 3: Linear Regression Model

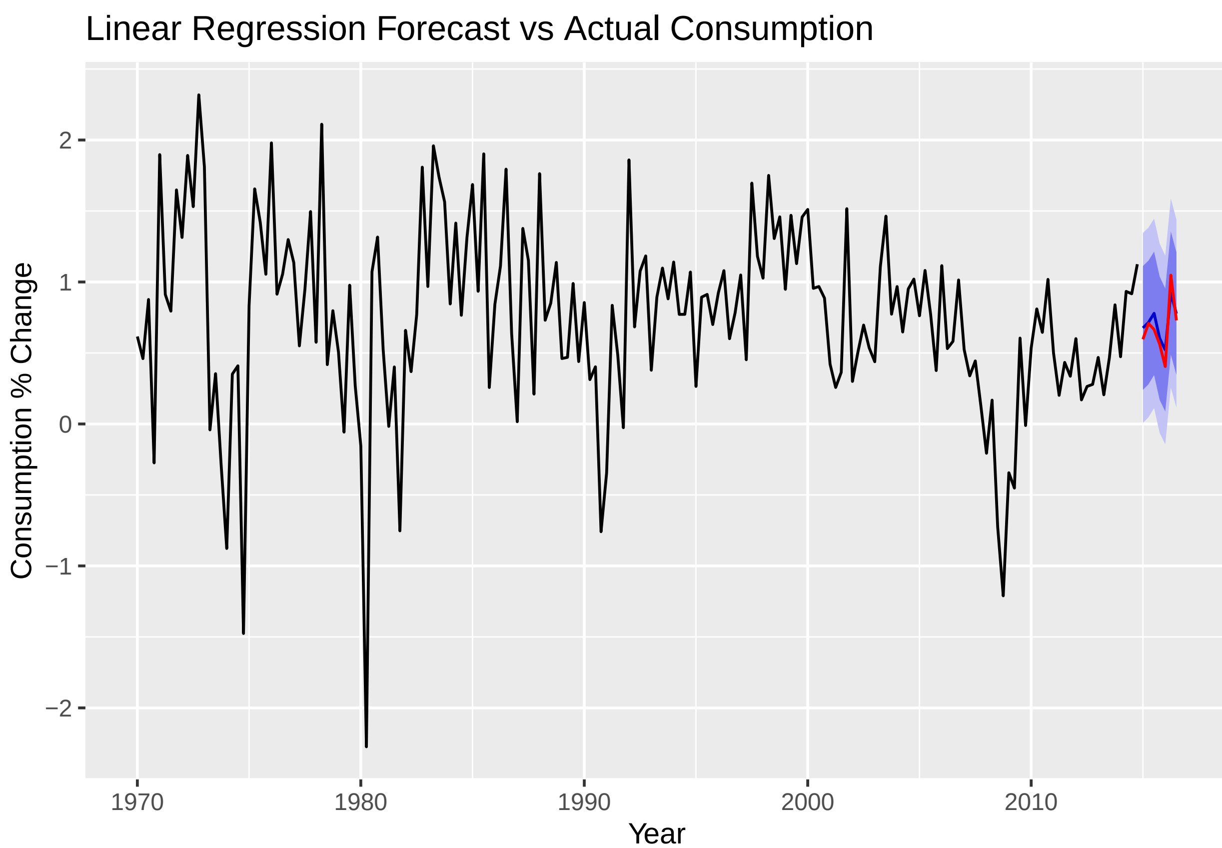
Rationale: This model was used to see if predictors like Income and Production, which showed strong correlations with Consumption, and adding uneployment and savings, as it was found these variables could improve the model, could help forecast it without needing to model patterns in the residuals

```
# Simple linear regression model
model_lm <- tslm(consumption_ts ~ income_ts + production_ts + unemployment_ts + savings_
checkresiduals(model_lm)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 14.404, df = 8, p-value = 0.07182
```

```r
# Forecasting using the test predictors
futurelm_xreg <- data.frame(
  income_ts = test[,"Income"],
  production_ts = test[,"Production"],
  unemployment_ts = test[,"Unemployment"],
  savings_ts = test[,"Savings"]
)

forecast_lm <- forecast(model_lm, newdata = futurelm_xreg)

autoplot(forecast_lm) +
  autolayer(test[,"Consumption"], series = "Actual", color = "red") +
  ggtitle("Linear Regression Forecast vs Actual Consumption") +
  ylab("Consumption % Change") +
  xlab("Year")
```




```r
accuracy(forecast_lm, test[,"Consumption"])
```

```
##                         ME       RMSE        MAE       MPE     MAPE      MASE
## Training set  1.831778e-17 0.32998484 0.24321096 18.456018 81.84864 0.3720917
## Test set     -4.024148e-02 0.08694945 0.07616324 -8.911877 12.34048 0.1165232
##                     ACF1 Theil's U
## Training set -0.08393455        NA
## Test set     -0.41841405 0.2495076
```

Findings: The residuals from the linear regression model appear to be fairly well-behaved. The residuals fluctuate around zero with constant variance. The linear regression model produced accurate forecasts, especially in the test period. The test set RMSE (0.087) and MASE (0.117) are both slightly better than those from the regression with ARIMA errors and significantly better than the ARIMA only model. The Theil's U statistic (0.25) also indicates strong accuracy. While the model does not account for autocorrelation, its strong test performance supports that simpler models can sometimes generalise better when predictors are strong.