Unemployment Forecasting

Sangeeta Kamite

2025-02-23

Load and Clean Data

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

```
# Load Unemployment Data
# Read the Unemployment Dataset (from local CSV)
df <- read.csv("unemployment.csv")

# Print column names to verify
colnames(df)
```

```
## [1] "observation_date" "UNRATE"
```

```
str(df)
```

```
## 'data.frame':    925 obs. of  2 variables:
##  $ observation_date: chr  "1948-01-01" "1948-02-01" "1948-03-01" "1948-04-01" ...
##  $ UNRATE          : num  3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
```

```
colnames(df)[1] <- "DATE"  # Rename first column if necessary
colnames(df)[2] <- "Unemployment_Rate"  # Rename for clarity


# Remove empty or missing values
df <- df %>% filter(DATE != "" & !is.na(DATE) & !is.na(Unemployment_Rate))

# Verify changes
str(df)
```

```
## 'data.frame':    925 obs. of  2 variables:
##  $ DATE            : chr  "1948-01-01" "1948-02-01" "1948-03-01" "1948-04-01" ...
##  $ Unemployment_Rate: num  3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
```

```r
# Convert DATE to Date Format
df$DATE <- as.Date(df$DATE, format="%Y-%m-%d")

# Rename Columns
colnames(df)[2] <- "Unemployment_Rate"

# Ensure the Unemployment Rate is Numeric
df$Unemployment_Rate <- as.numeric(df$Unemployment_Rate)
str(df)
```

```
## 'data.frame':    925 obs. of  2 variables:
##  $ DATE             : Date, format: "1948-01-01" "1948-02-01" ...
##  $ Unemployment_Rate: num  3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
```

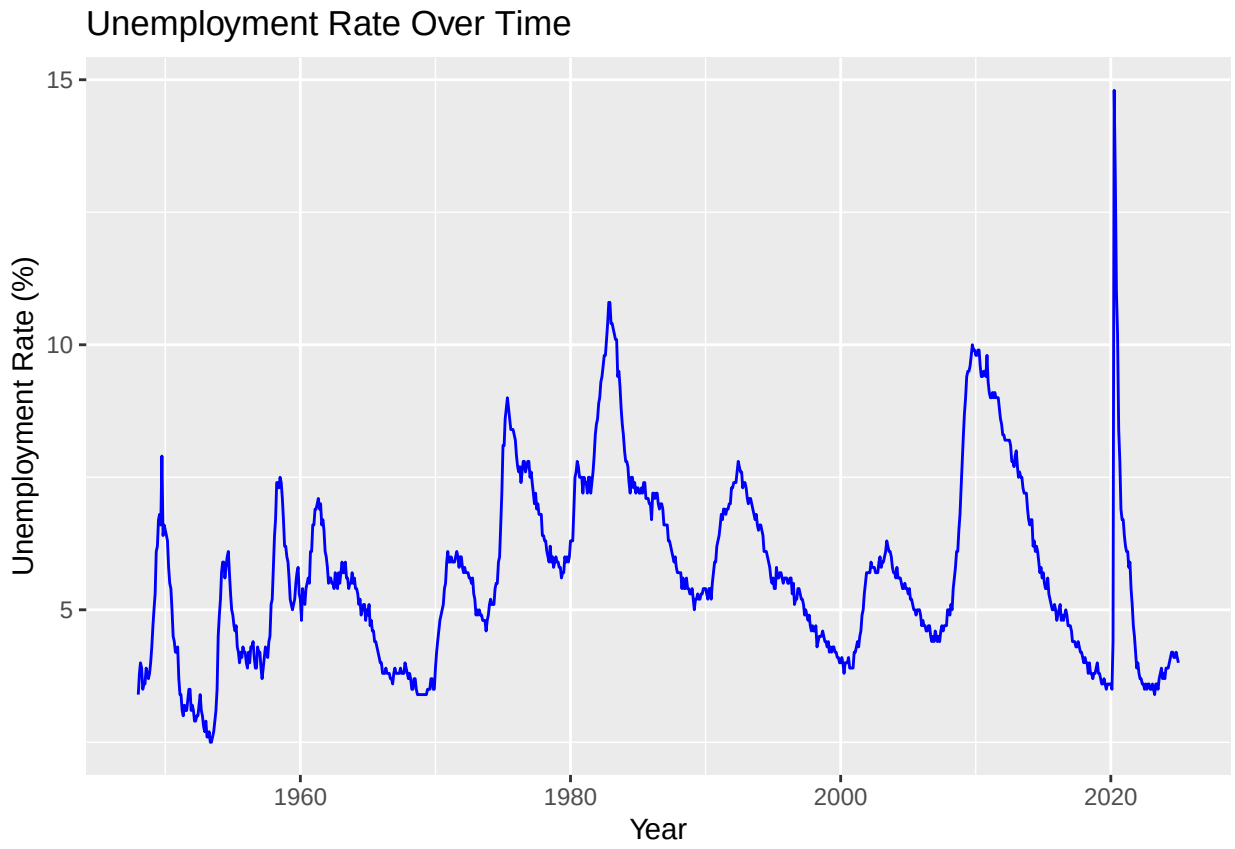
```r
# Remove any remaining NA values
df <- na.omit(df)

# Sort Data
df <- df %>% arrange(DATE)

# Check dataset structure
str(df)
```

```
## 'data.frame':    925 obs. of  2 variables:
##  $ DATE             : Date, format: "1948-01-01" "1948-02-01" ...
##  $ Unemployment_Rate: num  3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
```

Including Plots

You can also embed plots, for example:

```r
# Plot Unemployment Rate Over Time
ggplot(df, aes(x = DATE, y = Unemployment_Rate)) +
  geom_line(color = "blue") +
  ggtitle("Unemployment Rate Over Time") +
  xlab("Year") +
  ylab("Unemployment Rate (%)")
```




```
# Convert to Time Series Object
start_year <- year(min(df$DATE))
unemployment_ts <- ts(df$Unemployment_Rate, start = c(start_year, 1), frequency = 12)

# Verify time series
print(unemployment_ts)
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1948  3.4  3.8  4.0  3.9  3.5  3.6  3.6  3.9  3.8  3.7  3.8  4.0
## 1949  4.3  4.7  5.0  5.3  6.1  6.2  6.7  6.8  6.6  7.9  6.4  6.6
## 1950  6.5  6.4  6.3  5.8  5.5  5.4  5.0  4.5  4.4  4.2  4.2  4.3
## 1951  3.7  3.4  3.4  3.1  3.0  3.2  3.1  3.1  3.3  3.5  3.5  3.1
## 1952  3.2  3.1  2.9  2.9  3.0  3.0  3.2  3.4  3.1  3.0  2.8  2.7
## 1953  2.9  2.6  2.6  2.7  2.5  2.5  2.6  2.7  2.9  3.1  3.5  4.5
## 1954  4.9  5.2  5.7  5.9  5.9  5.6  5.8  6.0  6.1  5.7  5.3  5.0
## 1955  4.9  4.7  4.6  4.7  4.3  4.2  4.0  4.2  4.1  4.3  4.2  4.2
## 1956  4.0  3.9  4.2  4.0  4.3  4.3  4.4  4.1  3.9  3.9  4.3  4.2
## 1957  4.2  3.9  3.7  3.9  4.1  4.3  4.2  4.1  4.4  4.5  5.1  5.2
## 1958  5.8  6.4  6.7  7.4  7.4  7.3  7.5  7.4  7.1  6.7  6.2  6.2
## 1959  6.0  5.9  5.6  5.2  5.1  5.0  5.1  5.2  5.5  5.7  5.8  5.3
## 1960  5.2  4.8  5.4  5.2  5.1  5.4  5.5  5.6  5.5  6.1  6.1  6.6
## 1961  6.6  6.9  6.9  7.0  7.1  6.9  7.0  6.6  6.7  6.5  6.1  6.0
## 1962  5.8  5.5  5.6  5.6  5.5  5.5  5.4  5.7  5.6  5.4  5.7  5.5
## 1963  5.7  5.9  5.7  5.7  5.9  5.6  5.6  5.4  5.5  5.5  5.7  5.5
## 1964  5.6  5.4  5.4  5.3  5.1  5.2  4.9  5.0  5.1  5.1  4.8  5.0
## 1965  4.9  5.1  4.7  4.8  4.6  4.6  4.4  4.4  4.3  4.2  4.1  4.0
```

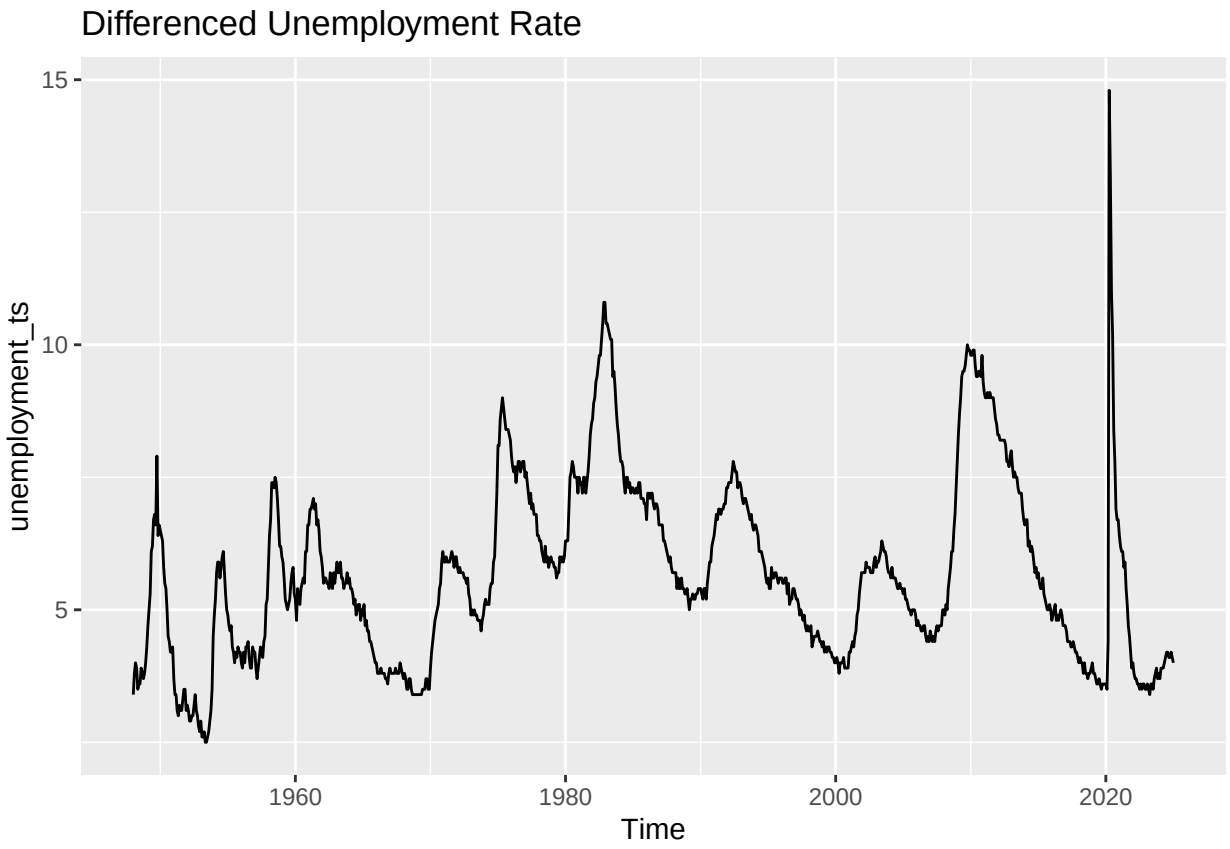
```
## 1966  4.0  3.8  3.8  3.8  3.9  3.8  3.8  3.8  3.7  3.7  3.6  3.8
## 1967  3.9  3.8  3.8  3.8  3.8  3.9  3.8  3.8  3.8  4.0  3.9  3.8
## 1968  3.7  3.8  3.7  3.5  3.5  3.7  3.7  3.5  3.4  3.4  3.4  3.4
## 1969  3.4  3.4  3.4  3.4  3.4  3.5  3.5  3.5  3.7  3.7  3.5  3.5
## 1970  3.9  4.2  4.4  4.6  4.8  4.9  5.0  5.1  5.4  5.5  5.9  6.1
## 1971  5.9  5.9  6.0  5.9  5.9  5.9  6.0  6.1  6.0  5.8  6.0  6.0
## 1972  5.8  5.7  5.8  5.7  5.7  5.7  5.6  5.6  5.5  5.6  5.3  5.2
## 1973  4.9  5.0  4.9  5.0  4.9  4.9  4.8  4.8  4.8  4.6  4.8  4.9
## 1974  5.1  5.2  5.1  5.1  5.1  5.4  5.5  5.5  5.9  6.0  6.6  7.2
## 1975  8.1  8.1  8.6  8.8  9.0  8.8  8.6  8.4  8.4  8.4  8.3  8.2
## 1976  7.9  7.7  7.6  7.7  7.4  7.6  7.8  7.8  7.6  7.7  7.8  7.8
## 1977  7.5  7.6  7.4  7.2  7.0  7.2  6.9  7.0  6.8  6.8  6.8  6.4
## 1978  6.4  6.3  6.3  6.1  6.0  5.9  6.2  5.9  6.0  5.8  5.9  6.0
## 1979  5.9  5.9  5.8  5.8  5.6  5.7  5.7  6.0  5.9  6.0  5.9  6.0
## 1980  6.3  6.3  6.3  6.9  7.5  7.6  7.8  7.7  7.5  7.5  7.5  7.2
## 1981  7.5  7.4  7.4  7.2  7.5  7.5  7.2  7.4  7.6  7.9  8.3  8.5
## 1982  8.6  8.9  9.0  9.3  9.4  9.6  9.8  9.8 10.1 10.4 10.8 10.8
## 1983 10.4 10.4 10.3 10.2 10.1 10.1  9.4  9.5  9.2  8.8  8.5  8.3
## 1984  8.0  7.8  7.8  7.7  7.4  7.2  7.5  7.5  7.3  7.4  7.2  7.3
## 1985  7.3  7.2  7.2  7.3  7.2  7.4  7.4  7.1  7.1  7.1  7.0  7.0
## 1986  6.7  7.2  7.2  7.1  7.2  7.2  7.0  6.9  7.0  7.0  6.9  6.6
## 1987  6.6  6.6  6.6  6.3  6.3  6.2  6.1  6.0  5.9  6.0  5.8  5.7
## 1988  5.7  5.7  5.7  5.4  5.6  5.4  5.4  5.6  5.4  5.4  5.3  5.3
## 1989  5.4  5.2  5.0  5.2  5.2  5.3  5.2  5.2  5.3  5.3  5.4  5.4
## 1990  5.4  5.3  5.2  5.4  5.4  5.2  5.5  5.7  5.9  5.9  6.2  6.3
## 1991  6.4  6.6  6.8  6.7  6.9  6.9  6.8  6.9  6.9  7.0  7.0  7.3
## 1992  7.3  7.4  7.4  7.4  7.6  7.8  7.7  7.6  7.6  7.3  7.4  7.4
## 1993  7.3  7.1  7.0  7.1  7.1  7.0  6.9  6.8  6.7  6.8  6.6  6.5
## 1994  6.6  6.6  6.5  6.4  6.1  6.1  6.1  6.0  5.9  5.8  5.6  5.5
## 1995  5.6  5.4  5.4  5.8  5.6  5.6  5.7  5.7  5.6  5.5  5.6  5.6
## 1996  5.6  5.5  5.5  5.6  5.6  5.3  5.5  5.1  5.2  5.2  5.4  5.4
## 1997  5.3  5.2  5.2  5.1  4.9  5.0  4.9  4.8  4.9  4.7  4.6  4.7
## 1998  4.6  4.6  4.7  4.3  4.4  4.5  4.5  4.5  4.6  4.5  4.4  4.4
## 1999  4.3  4.4  4.2  4.3  4.2  4.3  4.3  4.2  4.2  4.1  4.1  4.0
## 2000  4.0  4.1  4.0  3.8  4.0  4.0  4.0  4.1  3.9  3.9  3.9  3.9
## 2001  4.2  4.2  4.3  4.4  4.3  4.5  4.6  4.9  5.0  5.3  5.5  5.7
## 2002  5.7  5.7  5.7  5.9  5.8  5.8  5.8  5.7  5.7  5.7  5.9  6.0
## 2003  5.8  5.9  5.9  6.0  6.1  6.3  6.2  6.1  6.1  6.0  5.8  5.7
## 2004  5.7  5.6  5.8  5.6  5.6  5.6  5.5  5.4  5.4  5.5  5.4  5.4
## 2005  5.3  5.4  5.2  5.2  5.1  5.0  5.0  4.9  5.0  5.0  5.0  4.9
## 2006  4.7  4.8  4.7  4.7  4.6  4.6  4.7  4.7  4.5  4.4  4.5  4.4
## 2007  4.6  4.5  4.4  4.5  4.4  4.6  4.7  4.6  4.7  4.7  4.7  5.0
## 2008  5.0  4.9  5.1  5.0  5.4  5.6  5.8  6.1  6.1  6.5  6.8  7.3
## 2009  7.8  8.3  8.7  9.0  9.4  9.5  9.5  9.6  9.8 10.0  9.9  9.9
## 2010  9.8  9.8  9.9  9.9  9.6  9.4  9.4  9.5  9.5  9.4  9.8  9.3
## 2011  9.1  9.0  9.0  9.1  9.0  9.1  9.0  9.0  9.0  8.8  8.6  8.5
## 2012  8.3  8.3  8.2  8.2  8.2  8.2  8.2  8.1  7.8  7.8  7.7  7.9
## 2013  8.0  7.7  7.5  7.6  7.5  7.5  7.3  7.2  7.2  7.2  6.9  6.7
## 2014  6.6  6.7  6.7  6.2  6.3  6.1  6.2  6.1  5.9  5.7  5.8  5.6
## 2015  5.7  5.5  5.4  5.4  5.6  5.3  5.2  5.1  5.0  5.0  5.1  5.0
## 2016  4.8  4.9  5.0  5.1  4.8  4.9  4.8  4.9  5.0  4.9  4.7  4.7
## 2017  4.7  4.6  4.4  4.4  4.4  4.3  4.3  4.4  4.3  4.2  4.2  4.1
## 2018  4.0  4.1  4.0  4.0  3.8  4.0  3.8  3.8  3.7  3.8  3.8  3.9
## 2019  4.0  3.8  3.8  3.7  3.6  3.6  3.7  3.6  3.5  3.6  3.6  3.6
```

```
## 2020  3.6  3.5  4.4 14.8 13.2 11.0 10.2  8.4  7.8  6.9  6.7  6.7
## 2021  6.4  6.2  6.1  6.1  5.8  5.9  5.4  5.1  4.7  4.5  4.2  3.9
## 2022  4.0  3.8  3.7  3.7  3.6  3.6  3.5  3.6  3.5  3.6  3.6  3.5
## 2023  3.5  3.6  3.5  3.4  3.6  3.6  3.5  3.7  3.8  3.9  3.7  3.8
## 2024  3.7  3.9  3.9  3.9  4.0  4.1  4.2  4.2  4.1  4.1  4.2  4.1
## 2025  4.0
```

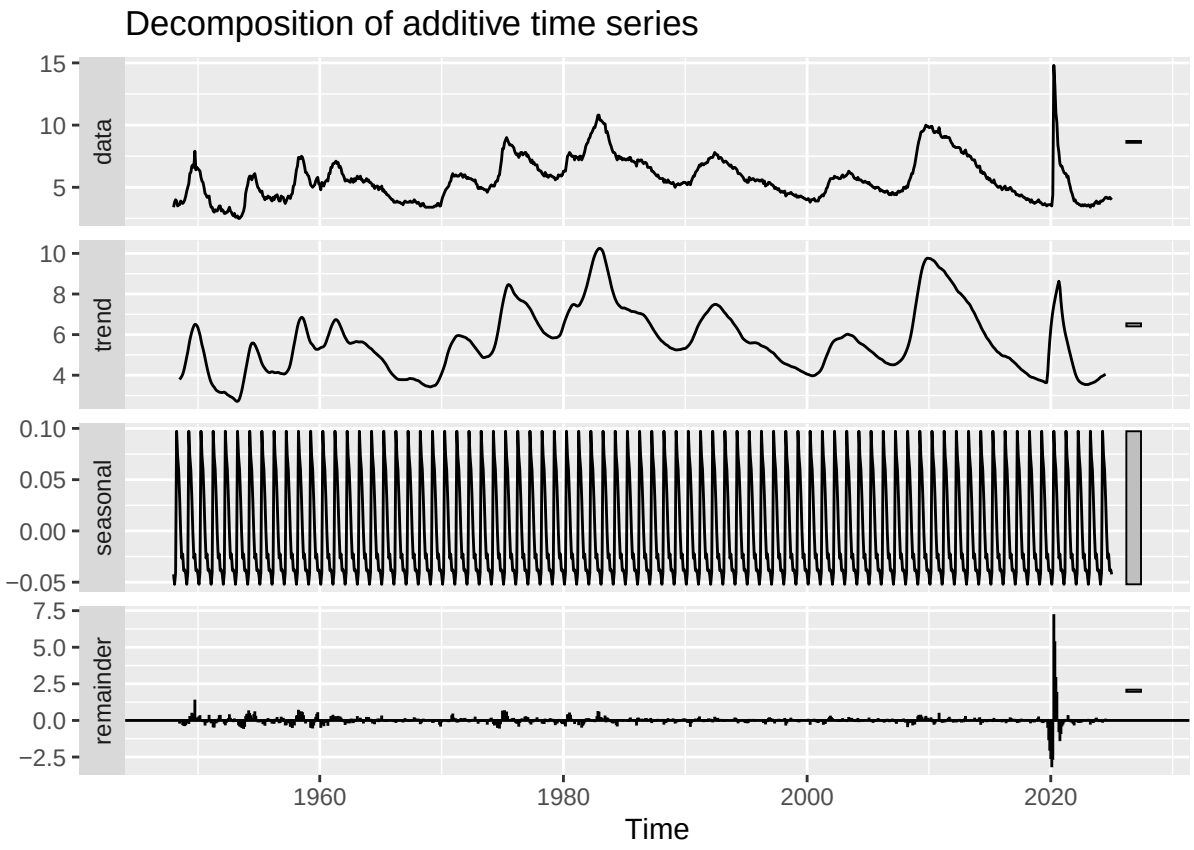
```r
# Perform Augmented Dickey-Fuller Test for Stationarity
adf_test <- adf.test(unemployment_ts)
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  unemployment_ts
## Dickey-Fuller = -3.9419, Lag order = 9, p-value = 0.0119
## alternative hypothesis: stationary
```

```r
# Apply Differencing if Necessary
if (adf_test$p.value > 0.05) {
  unemployment_ts <- diff(unemployment_ts, differences = 1)
}

# Plot Differenced Series
autoplot(unemployment_ts) + ggtitle("Differenced Unemployment Rate")
```



```r
# Decompose Time Series
unemployment_decomposed <- decompose(unemployment_ts)
autoplot(unemployment_decomposed)
```




```r
# Split Data into Training and Testing (80-20 Split)
train_size <- round(0.8 * length(unemployment_ts))
train_ts <- window(unemployment_ts, end = c(start_year + train_size / 12, train_size %% 12))
test_ts <- window(unemployment_ts, start = c(start_year + train_size / 12, train_size %% 12))
```

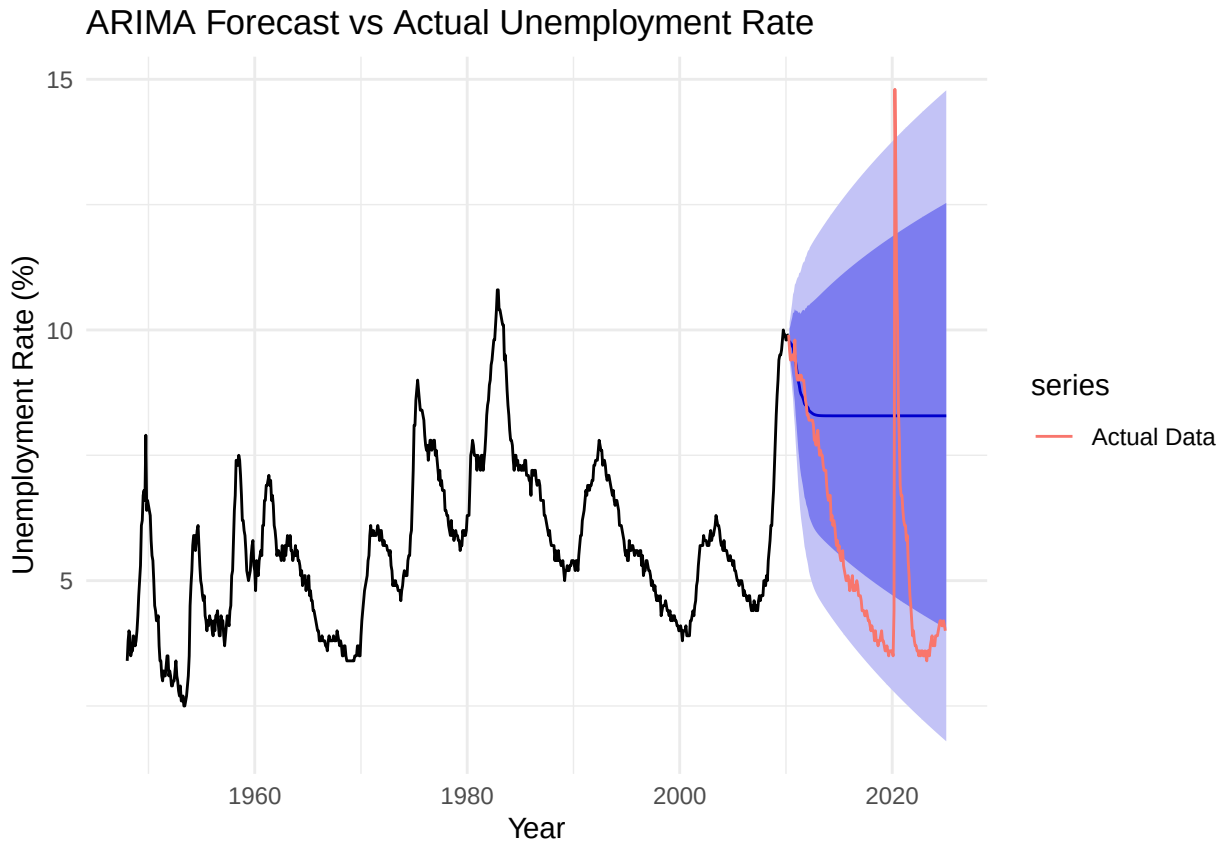
ARIMA Forecast

```r
# Fit ARIMA Model
arima_model <- auto.arima(train_ts)
summary(arima_model)
```

```
## Series: train_ts
## ARIMA(2,1,2)(0,0,2)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sma1     sma2
##       1.2297  -0.3621  -1.2331  0.5315  -0.249  -0.2243
## s.e.  0.2127   0.1942   0.1955  0.1450   0.037   0.0399
```

```
##
## sigma^2 = 0.03699:  log likelihood = 173.02
## AIC=-332.04   AICc=-331.89   BIC=-299.73
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.005333198 0.1914302 0.1411672 0.05547714 2.641887 0.1575183
##                     ACF1
## Training set -0.01239668
```

```r
# Forecast Using ARIMA
arima_forecast <- forecast(arima_model, h = length(test_ts))

# Plot ARIMA Forecast vs Actual
autoplot(arima_forecast) +
  autolayer(test_ts, series = "Actual Data") +
  ggtitle("ARIMA Forecast vs Actual Unemployment Rate") +
  xlab("Year") +
  ylab("Unemployment Rate (%)") +
  theme_minimal()
```




```r
# Evaluate ARIMA Model
arima_mae <- mean(abs(arima_forecast$mean - test_ts))
arima_rmse <- sqrt(mean((arima_forecast$mean - test_ts)^2))
cat("\nARIMA Model Performance:\n")
```

```
##
## ARIMA Model Performance:
```

```
cat("MAE:", arima_mae, "\nRMSE:", arima_rmse, "\n")
```

```
## MAE: 2.900051
## RMSE: 3.364702
```

XGBoost Machine Learning Model

```
# Create Lag Features for XGBoost
df$Year <- year(df$DATE)
df$Month <- month(df$DATE)
df$Lag_1 <- lag(df$Unemployment_Rate, 1)
df$Lag_12 <- lag(df$Unemployment_Rate, 12)
df <- na.omit(df)

# Train-Test Split (80-20)
train_size <- round(0.8 * nrow(df))
train_xgb <- df[1:train_size, c("Year", "Month", "Lag_1", "Lag_12")]
train_y <- df[1:train_size, "Unemployment_Rate"]
test_xgb <- df[(train_size + 1):nrow(df), c("Year", "Month", "Lag_1", "Lag_12")]
test_y <- df[(train_size + 1):nrow(df), "Unemployment_Rate"]

# Convert to XGBoost Matrix
train_matrix <- xgb.DMatrix(data = as.matrix(train_xgb), label = train_y)
test_matrix <- xgb.DMatrix(data = as.matrix(test_xgb), label = test_y)
```

ARIMA Forecast

```
# Train XGBoost Model
xgb_model <- xgboost(
  data = train_matrix,
  nrounds = 100,
  objective = "reg:squarederror",
  eta = 0.1
)
```

```
## [1]  train-rmse:4.865893
## [2]  train-rmse:4.386677
## [3]  train-rmse:3.955341
## [4]  train-rmse:3.566700
## [5]  train-rmse:3.216301
## [6]  train-rmse:2.901107
## [7]  train-rmse:2.617086
## [8]  train-rmse:2.361072
## [9]  train-rmse:2.130481
## [10] train-rmse:1.922794
## [11] train-rmse:1.735965
```

```
## [12] train-rmse:1.567892
## [13] train-rmse:1.416506
## [14] train-rmse:1.280460
## [15] train-rmse:1.157964
## [16] train-rmse:1.047925
## [17] train-rmse:0.949005
## [18] train-rmse:0.860161
## [19] train-rmse:0.780401
## [20] train-rmse:0.708828
## [21] train-rmse:0.644662
## [22] train-rmse:0.587058
## [23] train-rmse:0.535578
## [24] train-rmse:0.489534
## [25] train-rmse:0.448298
## [26] train-rmse:0.411536
## [27] train-rmse:0.378609
## [28] train-rmse:0.349312
## [29] train-rmse:0.323086
## [30] train-rmse:0.299767
## [31] train-rmse:0.278932
## [32] train-rmse:0.260296
## [33] train-rmse:0.243693
## [34] train-rmse:0.229371
## [35] train-rmse:0.216603
## [36] train-rmse:0.205428
## [37] train-rmse:0.195175
## [38] train-rmse:0.186242
## [39] train-rmse:0.178685
## [40] train-rmse:0.172062
## [41] train-rmse:0.166228
## [42] train-rmse:0.161238
## [43] train-rmse:0.156887
## [44] train-rmse:0.152732
## [45] train-rmse:0.149582
## [46] train-rmse:0.146526
## [47] train-rmse:0.143686
## [48] train-rmse:0.141251
## [49] train-rmse:0.139162
## [50] train-rmse:0.137289
## [51] train-rmse:0.135465
## [52] train-rmse:0.134050
## [53] train-rmse:0.132776
## [54] train-rmse:0.131533
## [55] train-rmse:0.130663
## [56] train-rmse:0.129912
## [57] train-rmse:0.129176
## [58] train-rmse:0.128139
## [59] train-rmse:0.127547
## [60] train-rmse:0.126824
## [61] train-rmse:0.125211
## [62] train-rmse:0.124307
## [63] train-rmse:0.123451
## [64] train-rmse:0.123094
## [65] train-rmse:0.122295
```

```
## [66] train-rmse:0.121936
## [67] train-rmse:0.121553
## [68] train-rmse:0.120886
## [69] train-rmse:0.120546
## [70] train-rmse:0.119869
## [71] train-rmse:0.119381
## [72] train-rmse:0.119151
## [73] train-rmse:0.118609
## [74] train-rmse:0.117696
## [75] train-rmse:0.117460
## [76] train-rmse:0.117028
## [77] train-rmse:0.116632
## [78] train-rmse:0.116192
## [79] train-rmse:0.116026
## [80] train-rmse:0.115179
## [81] train-rmse:0.115034
## [82] train-rmse:0.114474
## [83] train-rmse:0.114174
## [84] train-rmse:0.112905
## [85] train-rmse:0.112707
## [86] train-rmse:0.111897
## [87] train-rmse:0.110967
## [88] train-rmse:0.110167
## [89] train-rmse:0.109739
## [90] train-rmse:0.109621
## [91] train-rmse:0.108920
## [92] train-rmse:0.108779
## [93] train-rmse:0.108026
## [94] train-rmse:0.107703
## [95] train-rmse:0.106739
## [96] train-rmse:0.106080
## [97] train-rmse:0.105221
## [98] train-rmse:0.104742
## [99] train-rmse:0.103643
## [100]    train-rmse:0.102730
```

```
# Predict Using XGBoost
xgb_pred <- predict(xgb_model, test_matrix)

# Evaluate XGBoost Model
xgb_mae <- mean(abs(xgb_pred - test_y))
xgb_rmse <- sqrt(mean((xgb_pred - test_y)^2))
cat("\nXGBoost Model Performance:\n")
```

```
##
## XGBoost Model Performance:
```

```
cat("MAE:", xgb_mae, "\nRMSE:", xgb_rmse, "\n")
```

```
## MAE: 0.3250081
## RMSE: 0.8457338
```

```r
# Plot XGBoost Forecast vs Actual
ggplot() +
  geom_line(aes(x = df$DATE[(train_size + 1):nrow(df)], y = test_y, color = "Actual")) +
  geom_line(aes(x = df$DATE[(train_size + 1):nrow(df)], y = xgb_pred, color = "XGBoost Forecast")) +
  ggtitle("XGBoost Forecast vs Actual Unemployment Rate") +
  xlab("Year") +
  ylab("Unemployment Rate (%)") +
  theme_minimal()
```