A Simple Taylor Rule to Understand How the Federal Reserve Sets Interest Rates

Eshara Abeysekera, Jason Stewart, Louis Wu, Peter Liu

The **Taylor Rule** is a guideline suggesting how a central bank (such as the U.S. Federal Reserve) might adjust short-term interest rates.

Think of the economy as a large room with a **thermostat**, and **interest rates** are the thermostat's *temperature dial.* When the "room" (the economy) gets too hot, you lower the thermostat setting; when it's too cold, you raise it. The **Taylor Rule** helps decide how to adjust that dial, based on two key factors:

1. The gap between **actual inflation** and **target inflation**
2. The gap between **actual output** and **potential output**

Mathematically, in its original form, the **Taylor Rule** can be written as:

$$i_t = r^* + \pi_t + 0.5\big(\pi_t - \pi^*\big) + 0.5\big(\text{Output Gap}_t\big),$$

where:

- i_t = **nominal interest rate** (our thermostat dial).
- r^* = **long-run real interest rate** (think of this as a baseline setting).
- π_t = **current inflation**.
- π^* = **target (desired) inflation**.

Here's how this formula fits our **"thermostat" analogy**:

1. **Inflation Gap** $(\pi_t - \pi^*)$:
   - If actual inflation π_t is higher than the target π^*, it's like the room is *too hot.* The formula says to **increase** i_t (the interest rate), which "cools" the economy by making borrowing more expensive and slowing spending.
2. **Output Gap**:
   - If **actual output** is above **potential output**, it's like the room is overheated. The rule again prescribes increasing i_t so the economy doesn't overheat further.
   - If **actual output** is below **potential**, it's like the room is *too cold.* Lower i_t (the thermostat) to warm up the economy—cheaper borrowing can stimulate growth.

Task 1: Data Retrieval and Initial EDA (Quarterly Frequency)

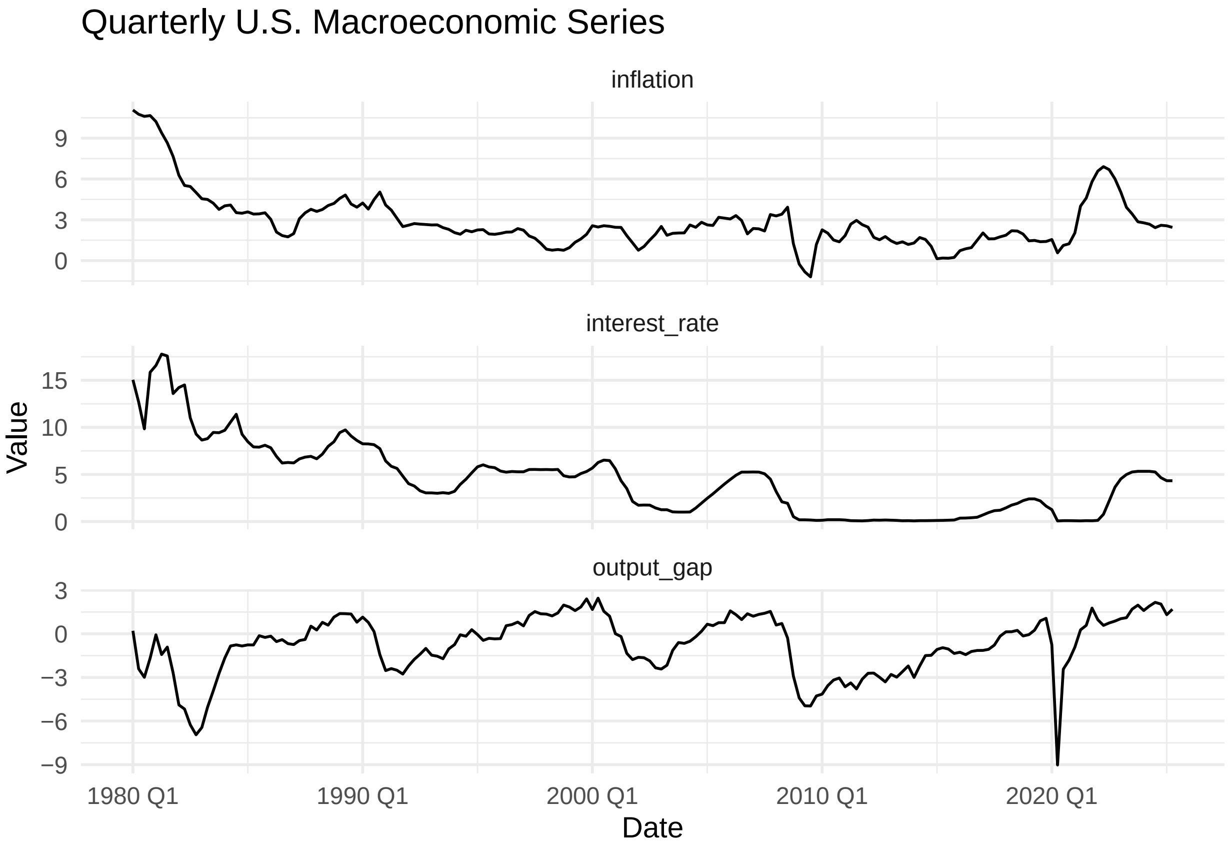
Obtain Data (Quarterly) (2 Points)

Construct Variables (2 Points)

```r
fredr_set_key(Sys.getenv("FRED_API_KEY"))

df <- purrr::map_dfr(
  c("FEDFUNDS", "PCEPI", "GDPC1", "GDPPOT"),
  ~fredr(
    series_id = .x,
    observation_start = as.Date("1979-01-01"),
    frequency = "q"
  )
) %>%
  dplyr::select(date, series_id, value) %>%
  pivot_wider(names_from = series_id, values_from = value) %>%
  mutate(
    date = yearquarter(date),
    interest_rate = FEDFUNDS,
    inflation = 100 * (PCEPI / lag(PCEPI, n = 4) - 1),
    output_gap = 100 * (GDPC1 - GDPPOT) / GDPPOT
  ) %>%
  dplyr::select(c(date, interest_rate, inflation, output_gap)) %>%
  drop_na() %>%
  as_tsibble(index = date)
```

Initial Plots and Summaries (6 Points)

```r
df %>%
  pivot_longer(
    cols = c(interest_rate, inflation, output_gap),
    names_to = "series",
    values_to = "value"
  ) %>%
  ggplot(aes(x = date, y = value)) +
  geom_line() +
  facet_wrap(~series, scales = "free_y", ncol = 1) +
  labs(
    title = "Quarterly U.S. Macroeconomic Series",
    x = "Date",
    y = "Value"
  ) +
  theme_minimal()
```



```r
kable(
  data.frame(
    series = c("interest_rate", "inflation", "output_gap"),
    min = c(min(df$interest_rate), min(df$inflation), min(df$output_gap)),
    max = c(max(df$interest_rate), max(df$inflation), max(df$output_gap)),
    mean = c(mean(df$interest_rate), mean(df$inflation), mean(df$output_gap)),
    var = c(var(df$interest_rate), var(df$inflation), var(df$output_gap))
  )
)
```

series	min	max	mean	var
interest_rate	0.060000	17.780000	4.4232418	15.339583
inflation	-1.195952	11.069696	2.8523573	4.204819
output_gap	-9.024282	2.451811	-0.7547185	4.044786

Based on the data, we see high inflation periods around the early 1980s and in the early 2020s. The 2020 COVID-19 pandemic was a notable event economically, where a decline in interest rates were immediately followed by a spike within a year.

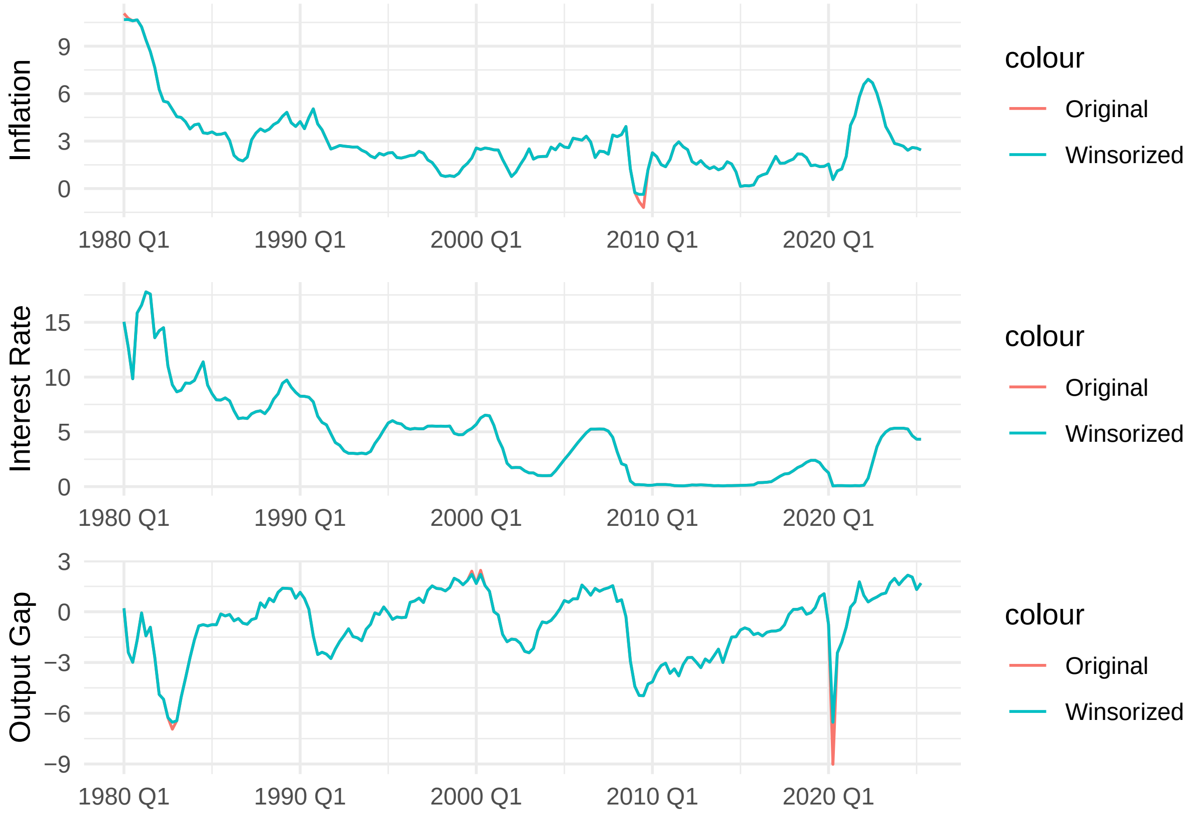
Inflation rates started above 10% in the 1980s and fell to low single digits by the 2000s. The sudden drop around 2008 is most likely associated with the collapse of the US housing market and the recession that followed. This is perhaps relevant for the notably low and stable interest rates from 2008 to 2017, which may have been in response to the economic crisis.

Task 2: Preprocessing (12 Points)

2.1: Frequency Alignment (Resampling) (2 Points)

2.2: Outlier Detection & Treatment (5 Points)

```r
winsorize <- function(x, lower_quantile = 0.01, upper_quantile = 0.99) {
  lower <- quantile(x, lower_quantile, na.rm = TRUE)
  upper <- quantile(x, upper_quantile, na.rm = TRUE)
  x_wins <- pmin(pmax(x, lower), upper)
  return(x_wins)
}

df <- df %>%
  mutate(
    interest_rate.w = winsorize(interest_rate),
    inflation.w = winsorize(inflation),
    output_gap.w = winsorize(output_gap)
  )

winsor_1 <- df %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = inflation, color = "Original")) +
  geom_line(aes(y = inflation.w, color = "Winsorized")) +
  labs(x = NULL, title = NULL, y = "Inflation")

winsor_2 <- df %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = interest_rate, color = "Original")) +
  geom_line(aes(y = interest_rate, color = "Winsorized")) +
  labs(x = NULL, title = NULL, y = "Interest Rate")

winsor_3 <- df %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = output_gap, color = "Original")) +
  geom_line(aes(y = output_gap.w, color = "Winsorized")) +
  labs(x = NULL, title = NULL, y = "Output Gap")

winsor_1 / winsor_2 / winsor_3
```

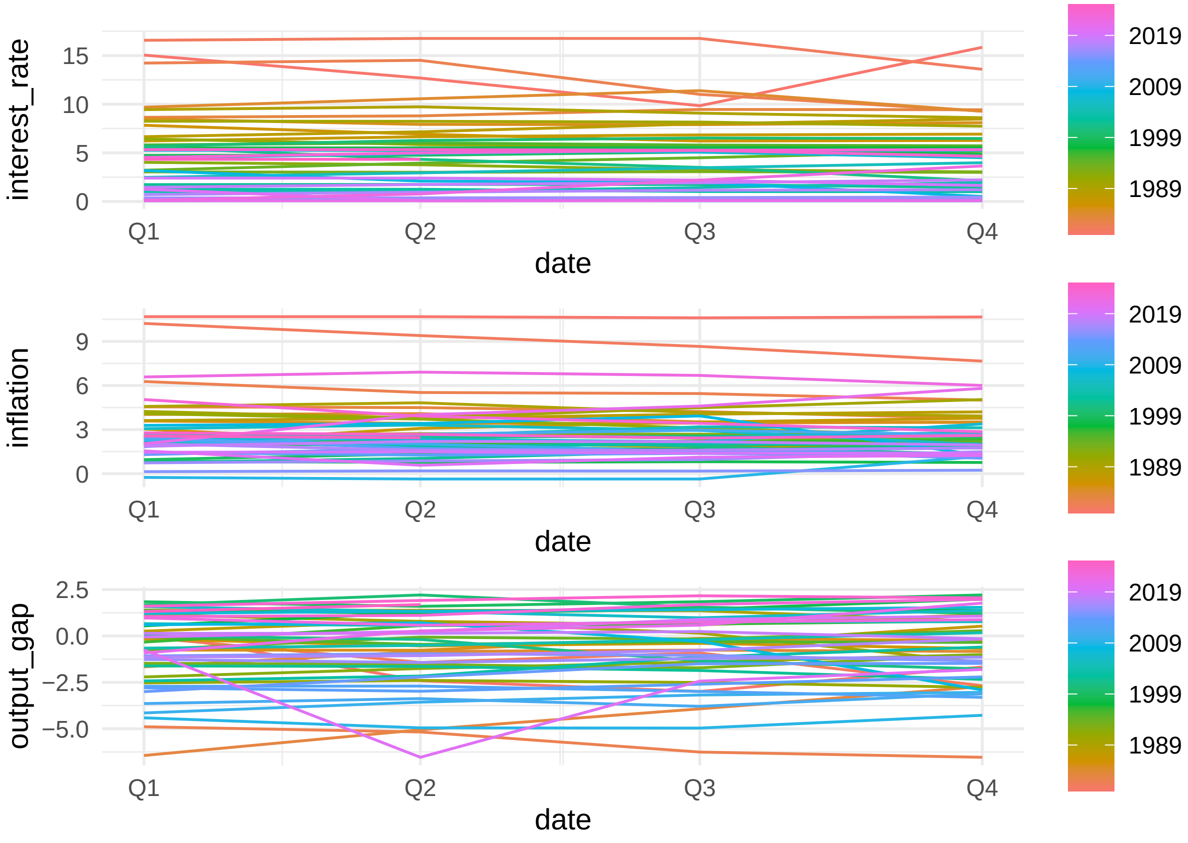


```r
df <- df %>%
  mutate(
    interest_rate = interest_rate.w,
    inflation = inflation.w,
    output_gap = output_gap.w
  ) %>%
  dplyr::select(-c(interest_rate.w, inflation.w, output_gap.w))
```

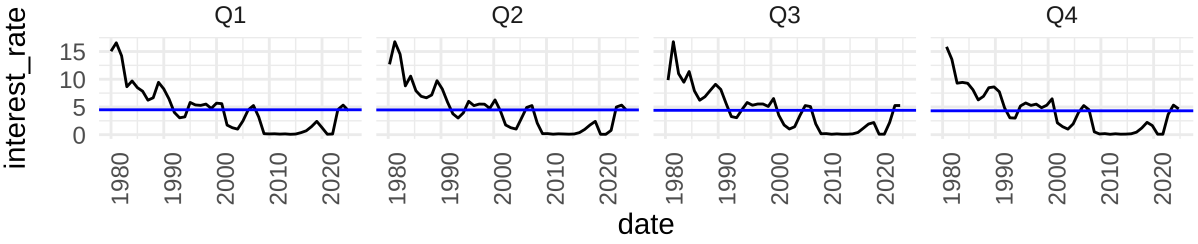
From the plot above, we see observe multiple visible areas of outliers, with the most notable ones occurring in 2008 for `inflation` and around 2020 for `output_gap`. We used percentile-based Winsorizing to replace the outliers with the cap values of 1st and 99th percentile values (i.e. if a value was higher than the 99th percentile, we replaced it with the 99th percentile value, and if it was lower than the 1st percentile, we replaced it with the 1st percentile value). We chose this method to avoid removing extreme data points. These outliers represent real shocks caused by significant events, and eliminating them would make the model less reflective of actual behavior. Instead, Winsorizing allows us to retain the true minimum and maximum values while smoothing extreme fluctuations so they don't overly distort the model.

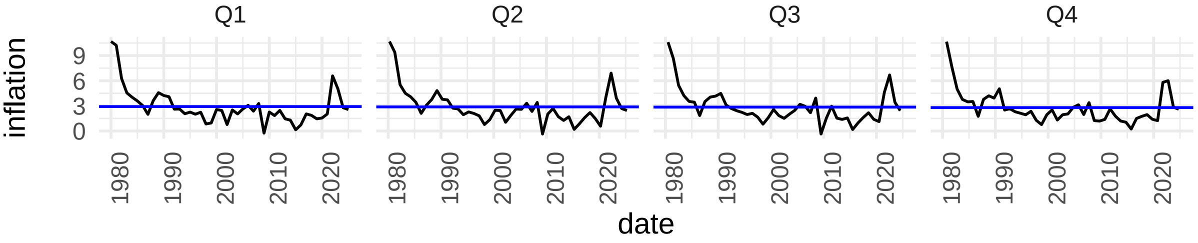
2.3: Seasonal Adjustment (5 Points)

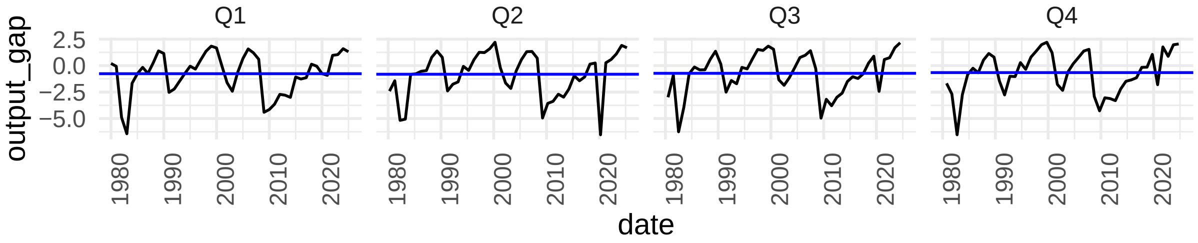
```
season_1 <- df %>%
  gg_season(interest_rate)
season_2 <- df %>%
  gg_season(inflation)
season_3 <- df %>%
  gg_season(output_gap)

subseries_1 <- df %>%
  gg_subseries(interest_rate)
subseries_2 <- df %>%
  gg_subseries(inflation)
subseries_3 <- df %>%
  gg_subseries(output_gap)

season_1 / season_2 / season_3
```



```
subseries_1 / subseries_2 / subseries_3
```







```r
kable(
  data.frame(
    series = c("interest_rate", "inflation", "output_gap"),
    QS_pval = c(
      qs(df$interest_rate, freq = 4, diff = FALSE)$Pval,
      qs(df$inflation, freq = 4, diff = FALSE)$Pval,
      qs(df$output_gap, freq = 4, diff = FALSE)$Pval
    ),
    QS_diff_pval = c(
      qs(df$interest_rate, freq = 4)$Pval,
      qs(df$inflation, freq = 4)$Pval,
      qs(df$output_gap, freq = 4)$Pval
    )
  )
)
```

series	QS_pval	QS_diff_pval
interest_rate	0	1
inflation	0	1
output_gap	0	1

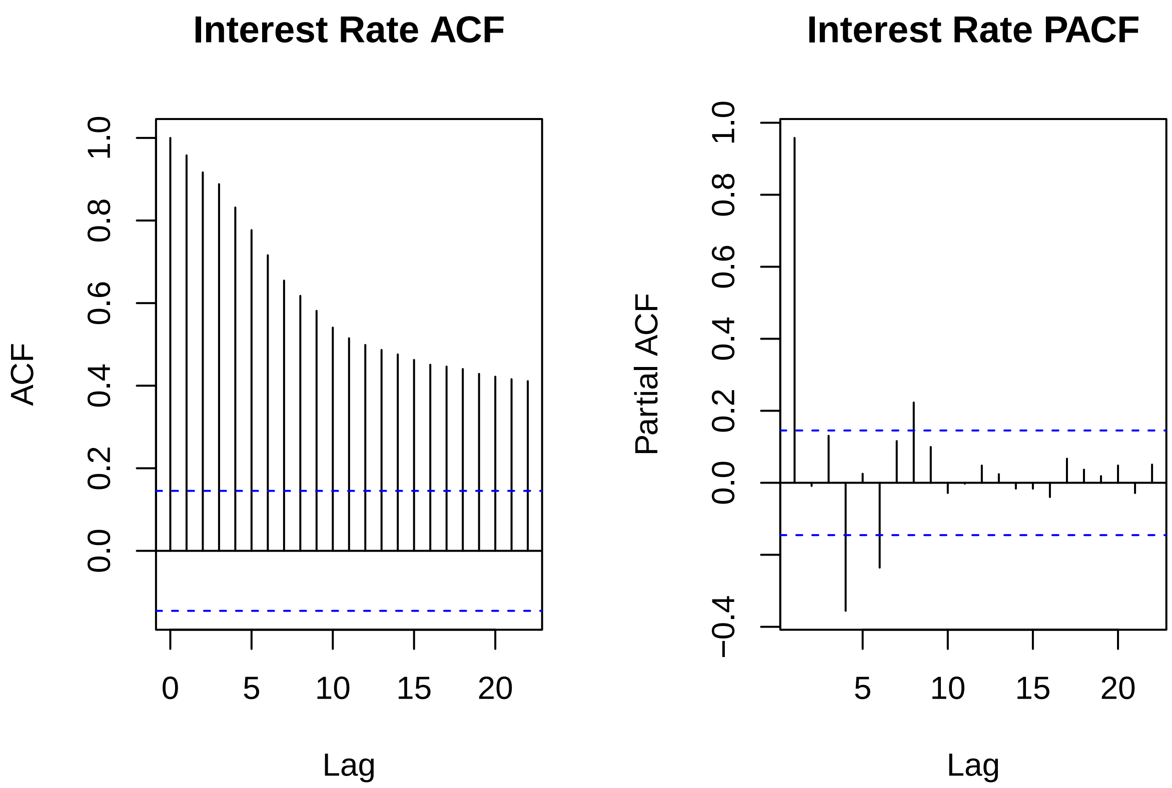
All three measures (inflation, interest rate, and output gap) show no signs of seasonality. The values remain relatively constant within a given year in the seasonality plots and the subseries plots are relatively similar across quarters. Additionally, the QS tests fail to reject the null hypothesis of no seasonality. Therefore, no seasonality adjustment is needed for the data.

Task 3: Stationarity Testing & Potential Structural Breaks (13 Points)

3.1: Stationarity Testing (8 Points)

Stationary Test for Interest Rate: The ACF and PACF plots indicate non-stationary behavior due to slow decay and multiple significant lags. The Unit Root Augmented Dickey-Fuller trend test yields a test statistic below the 5 percent critical value, suggesting that the interest rate series is trend non-stationary. However, after first order differencing, the ADF test yields a test statistic beyond the 5 percent critical value, suggesting that the interest rate series is difference stationary.

```
par(mfrow = c(1, 2))
acf(df$interest_rate, main = "Interest Rate ACF")
pacf(df$interest_rate, main = "Interest Rate PACF")
```



```
print('Interest Rate Augmented Dickey-Fuller Test (Trend)')
```

```
## [1] "Interest Rate Augmented Dickey-Fuller Test (Trend)"
```

```
urdf_trend_interest_rate <- summary(
  ur.df(
    df$interest_rate,
    type = "trend",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```
urdf_trend_interest_rate@cval['tau3',]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```
print(
  paste(
    'Test Statistic:',
    urdf_trend_interest_rate@teststat[,'tau3']
  )
)
```

```
## [1] "Test Statistic: -2.3572970482327"
```

```
print('Interest Rate Augmented Dickey-Fuller Test (Difference)')
```

```
## [1] "Interest Rate Augmented Dickey-Fuller Test (Difference)"
```

```
urdf_diff_interest_rate <- summary(
  ur.df(
    diff(df$interest_rate),
    type = "none",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```
urdf_diff_interest_rate@cval['tau1',]
```

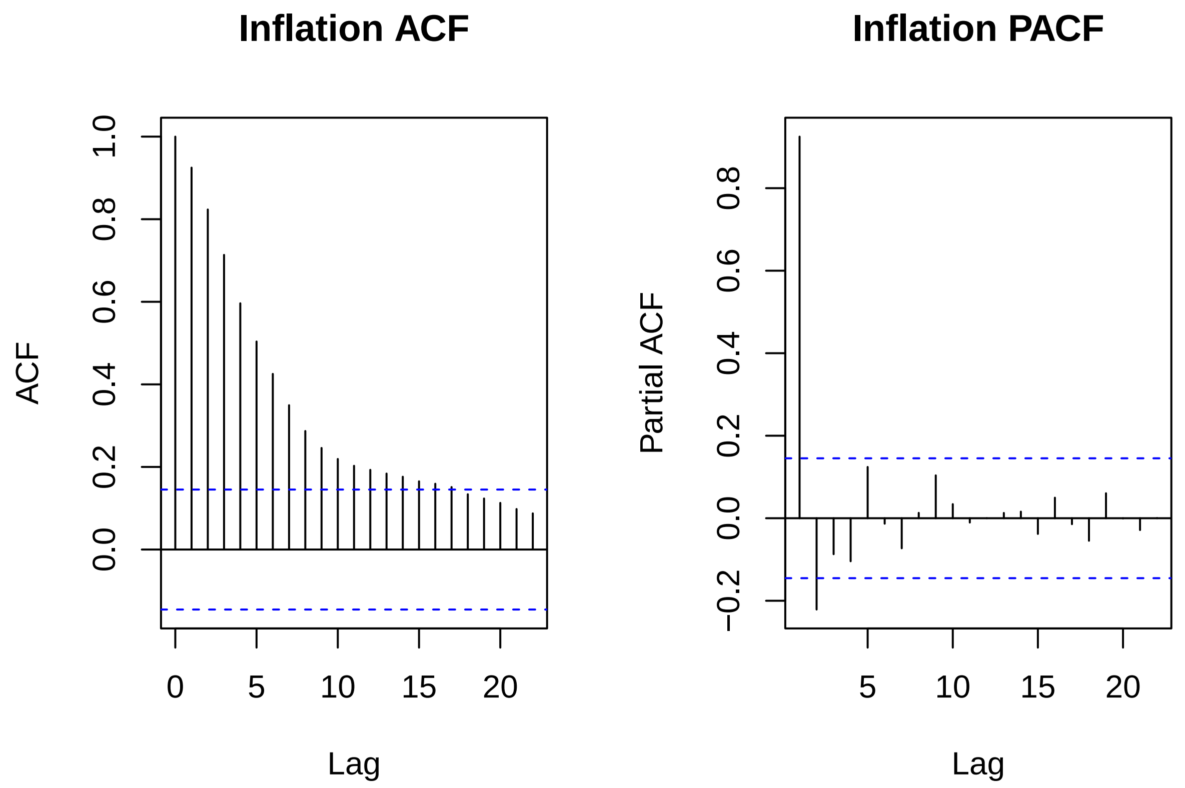
```
##  1pct  5pct 10pct
## -2.58 -1.95 -1.62
```

```
print(
  paste(
    'Test Statistic:',
    urdf_diff_interest_rate@teststat[,'tau1']
  )
)
```

```
## [1] "Test Statistic: -9.96519022583542"
```

Stationary Test for Inflation: The ACF and PACF plots indicate stationary behavior due to rapid decay and no significant lags. The Unit Root Augmented Dickey-Fuller test yields a test statistic below the 5 percent critical value, suggesting that the interest rate series is strictly non-stationary.

```
par(mfrow = c(1, 2))
acf(df$inflation, main = "Inflation ACF")
pacf(df$inflation, main = "Inflation PACF")
```



```
print('Inflation Augmented Dickey-Fuller Test (Trend)')
```

```
## [1] "Inflation Augmented Dickey-Fuller Test (Trend)"
```

```
urdf_trend_inflation <- summary(
  ur.df(
    df$inflation,
    type = "trend",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```r
urdf_trend_inflation@cval['tau3',]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```r
print(
  paste(
    'Test Statistic:',
    urdf_trend_inflation@teststat[,'tau3']
  )
)
```

```
## [1] "Test Statistic: -4.28416167071156"
```

```r
print('Inflation Augmented Dickey-Fuller Test (Drift)')
```

```
## [1] "Inflation Augmented Dickey-Fuller Test (Drift)"
```

```r
urdf_drift_inflation <- summary(
  ur.df(
    df$inflation,
    type = "drift",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```r
urdf_drift_inflation@cval['tau2',]
```

```
##  1pct  5pct 10pct
## -3.46 -2.88 -2.57
```

```r
print(
  paste(
    'Test Statistic:',
    urdf_drift_inflation@teststat[,'tau2']
  )
)
```

```
## [1] "Test Statistic: -4.43177901032921"
```

```r
print('Inflation Augmented Dickey-Fuller Test (None)')
```

```
## [1] "Inflation Augmented Dickey-Fuller Test (None)"
```

```r
urdf_none_inflation <- summary(
  ur.df(
    df$inflation,
    type = "none",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```r
urdf_none_inflation@cval['tau1',]
```

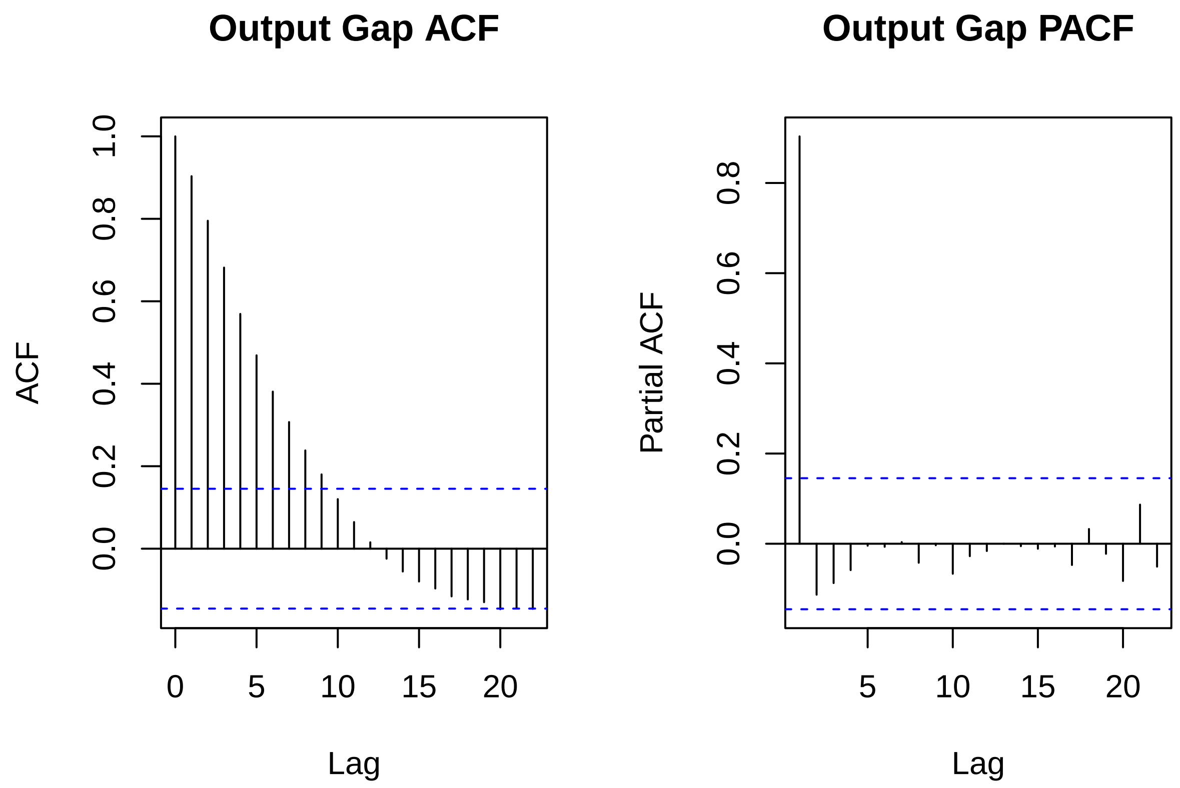
```
##  1pct  5pct 10pct 
## -2.58 -1.95 -1.62
```

```r
print(
  paste(
    'Test Statistic:',
    urdf_none_inflation@teststat[,'tau1']
  )
)
```

```
## [1] "Test Statistic: -3.12024992463496"
```

Stationary Test for Output Gap: The ACF and PACF plots indicate non-stationary behavior due to oscillating decay. The Unit Root Augmented Dickey-Fuller trend test yields a test statistic below the 5 percent critical value, suggesting that the interest rate series is trend non-stationary. However, after first order differencing, the ADF test yields a test statistic beyond the 5 percent critical value, suggesting that the output gap series is difference stationary.

```r
par(mfrow = c(1, 2))
acf(df$output_gap, main = "Output Gap ACF")
pacf(df$output_gap, main = "Output Gap PACF")
```



```r
print('Output Gap Augmented Dickey-Fuller Test (Trend)')
```

```
## [1] "Output Gap Augmented Dickey-Fuller Test (Trend)"
```

```r
urdf_trend_output_gap <- summary(
  ur.df(
    df$output_gap,
    type = "trend",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```r
urdf_trend_output_gap@cval['tau3',]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```r
print(
  paste(
    'Test Statistic:',
    urdf_trend_output_gap@teststat[,'tau3']
  )
)
```

```
## [1] "Test Statistic: -3.19441743122096"
```

```
print('Output Gap Augmented Dickey-Fuller Test (Difference)')
```

```
## [1] "Output Gap Augmented Dickey-Fuller Test (Difference)"
```

```
urdf_diff_output_gap <- summary(
  ur.df(
    diff(df$output_gap),
    type = "none",
    selectlags = "AIC"
  )
)
print('Critical Values')
```

```
## [1] "Critical Values"
```

```
urdf_diff_output_gap@cval['tau1',]
```

```
##  1pct  5pct 10pct
## -2.58 -1.95 -1.62
```

```
print(
  paste(
    'Test Statistic:',
    urdf_diff_output_gap@teststat[,'tau1']
  )
)
```

```
## [1] "Test Statistic: -9.0725432170296"
```

3.2: Stationarity with structural break (5 Points)

Stationary Test with Breaks for Interest Rate: The Zivot-Andrews test yields a test statistic below the 5 percent critical value, suggesting that the interest rate series is trend non-stationary. However, after first order differencing, the ZA test yields a test statistic beyond the 5 percent critical value, suggesting that the interest rate series is difference stationary.

```
print('Interest Rate Zivot-Andrews Test')
```

```
## [1] "Interest Rate Zivot-Andrews Test"
```

```
urza_both_interest_rate <- summary(
  ur.za(
    df$interest_rate,
    model = "both"
  )
)
print(
  paste(
    '5 percent Critical Value:',
    urza_both_interest_rate@cval[2]
  )
)
```

```
## [1] "5 percent Critical Value: -5.08"
```

```
print(
  paste(
    'Test Statistic:',
    urza_both_interest_rate@teststat
  )
)
```

```
## [1] "Test Statistic: -3.67170937273428"
```

```
print('Interest Rate Zivot-Andrews Test (Difference)')
```

```
## [1] "Interest Rate Zivot-Andrews Test (Difference)"
```

```
urza_diff_interest_rate <- summary(
  ur.za(
    diff(df$interest_rate),
    model = "both"
  )
)
print(
  paste(
    '5 percent Critical Value:',
    urza_diff_interest_rate@cval[2]
  )
)
```

```
## [1] "5 percent Critical Value: -5.08"
```

```
print(
  paste(
    'Test Statistic:',
    urza_diff_interest_rate@teststat
  )
)
```

```
## [1] "Test Statistic: -12.6153890285503"
```

Stationary Test with Breaks for Inflation: The Zivot-Andrews test yields a test statistic below the 5 percent critical value, suggesting that the inflation series is trend non-stationary. However, after first order differencing, the ZA test yields a test statistic beyond the 5 percent critical value, suggesting that the inflation series is difference stationary.

```
print('Inflation Zivot-Andrews Test')
```

```
## [1] "Inflation Zivot-Andrews Test"
```

```
urza_both_inflation <- summary(
  ur.za(
    df$inflation,
    model = "both"
  )
)
print(
  paste(
    '5 percent Critical Value:',
    urza_both_inflation@cval[2]
  )
)
```

```
## [1] "5 percent Critical Value: -5.08"
```

```
print(
  paste(
    'Test Statistic:',
    urza_both_inflation@teststat
  )
)
```

```
## [1] "Test Statistic: -4.7749940881385"
```

```
print('Inflation Zivot-Andrews Test (Difference)')
```

```
## [1] "Inflation Zivot-Andrews Test (Difference)"
```

```r
urza_diff_inflation <- summary(
  ur.za(
    diff(df$inflation),
    model = "both"
  )
)
print(
  paste(
    '5 percent Critical Value:',
    urza_diff_inflation@cval[2]
  )
)
```

```
## [1] "5 percent Critical Value: -5.08"
```

```r
print(
  paste(
    'Test Statistic:',
    urza_diff_inflation@teststat
  )
)
```

```
## [1] "Test Statistic: -9.54087824235928"
```

Stationary Test with Breaks for Output Gap: The Zivot-Andrews test yields a test statistic below the 5 percent critical value, suggesting that the output gap series is trend non-stationary. However, after first order differencing, the ZA test yields a test statistic beyond the 5 percent critical value, suggesting that the output gap series is difference stationary.

```r
print('Output Gap Zivot-Andrews Test')
```

```
## [1] "Output Gap Zivot-Andrews Test"
```

```r
urza_both_output_gap <- summary(
  ur.za(
    df$output_gap,
    model = "both"
  )
)
print(
  paste(
    '5 percent Critical Value:',
    urza_both_output_gap@cval[2]
  )
)
```

```
## [1] "5 percent Critical Value: -5.08"
```

```r
print(
  paste(
    'Test Statistic:',
    urza_both_output_gap@teststat
  )
)
```

```
## [1] "Test Statistic: -4.81581059629692"
```

```r
print('Output Gap Zivot-Andrews Test (Difference)')
```

```
## [1] "Output Gap Zivot-Andrews Test (Difference)"
```

```r
urza_diff_output_gap <- summary(
  ur.za(
    diff(df$output_gap),
    model = "both"
  )
)
print(
  paste(
    '5 percent Critical Value:',
    urza_diff_output_gap@cval[2]
  )
)
```

```
## [1] "5 percent Critical Value: -5.08"
```

```r
print(
  paste(
    'Test Statistic:',
    urza_diff_output_gap@teststat
  )
)
```

```
## [1] "Test Statistic: -13.0788558834627"
```

Task 4: Estimating the Basic Taylor Rule (OLS) (10 Points)

```
test_start <- yearquarter(as.Date("2023-01-01"))

df <- df %>%
  mutate(inflation_gap = inflation - 2)

df_train <- df %>%
  filter(date < test_start)
df_test <- df %>%
  filter(date >= test_start)

taylor_model <- lm(interest_rate ~ inflation_gap + output_gap, data = df_train)
summary(taylor_model)
```

```
##
## Call:
## lm(formula = interest_rate ~ inflation_gap + output_gap, data = df_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7427 -1.7712  0.0094  2.0260  7.3171
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.4033     0.2503  13.597   <2e-16 ***
## inflation_gap   1.3675     0.1029  13.294   <2e-16 ***
## output_gap      0.1987     0.1115   1.782   0.0766 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.794 on 169 degrees of freedom
## Multiple R-squared:  0.5159, Adjusted R-squared:  0.5102
## F-statistic: 90.07 on 2 and 169 DF,  p-value: < 2.2e-16
```

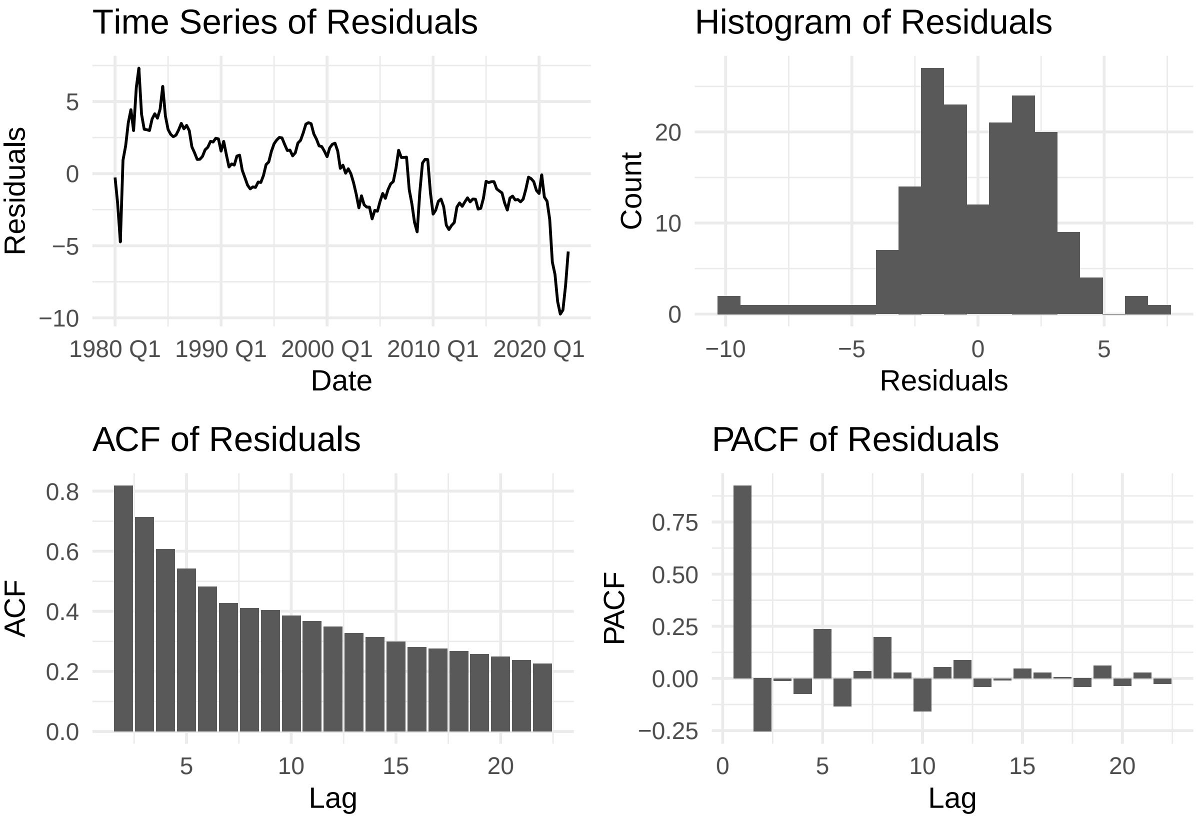
```
alpha1 <- coef(taylor_model)["inflation_gap"]
```

Our estimate for α_1 is 1.3674597, which is well about 4 times the theoretical value of 1 and is highly-significant. This suggests that actual policy is not aligned with Taylor Rule theory. Specifically, the Federal Reserve is quite aggressive in setting interest rates in response to bring inflation. This result is probably strongly influenced by the early 1980's period, when Fed Chairn Paul Volker was responding to potential hyperinflation experienced in the late 1970's (known as the "Volker Period") and acting aggressively. Interestingly, as of the time of writing J Powell, the current fed chair has just announced a reduction in interest rates despite risks on both sides of the equation (high-inflation and unemployment).

2. **Model Diagnostics**

```
df_train <- df_train %>%
  mutate(
    predictions_taylor = fitted(taylor_model),
    residuals_taylor = residuals(taylor_model)
```

```r
  )

p1 <- df_train %>%
  ggplot(aes(x = date, y = residuals_taylor)) +
  geom_line() +
  labs(title = "Time Series of Residuals", x = "Date", y = "Residuals") +
  theme_minimal()

p2 <- df_train %>%
  ggplot(aes(x = residuals_taylor)) +
  geom_histogram(bins = 20) +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Count") +
  theme_minimal()

acf_obj <- acf(df_train$residuals_taylor, plot = FALSE)
acf_df <- data.frame(lag = acf_obj$lag[-1], acf = acf_obj$acf[-1])
p3 <- ggplot(acf_df[-1, ], aes(x = lag, y = acf)) +   # omit lag 0
  geom_bar(stat = "identity") +
  labs(title = "ACF of Residuals", x = "Lag", y = "ACF") +
  theme_minimal()

pacf_obj <- pacf(df_train$residuals_taylor, plot = FALSE)
pacf_df <- data.frame(lag = pacf_obj$lag, pacf = pacf_obj$acf)
p4 <- ggplot(pacf_df, aes(x = lag, y = pacf)) +
  geom_bar(stat = "identity") +
  labs(title = "PACF of Residuals", x = "Lag", y = "PACF") +
  theme_minimal()

(p1 | p2) / (p3 | p4)
```



```r
dwtest(taylor_model)
```

```
##
##  Durbin-Watson test
##
## data:  taylor_model
## DW = 0.12806, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

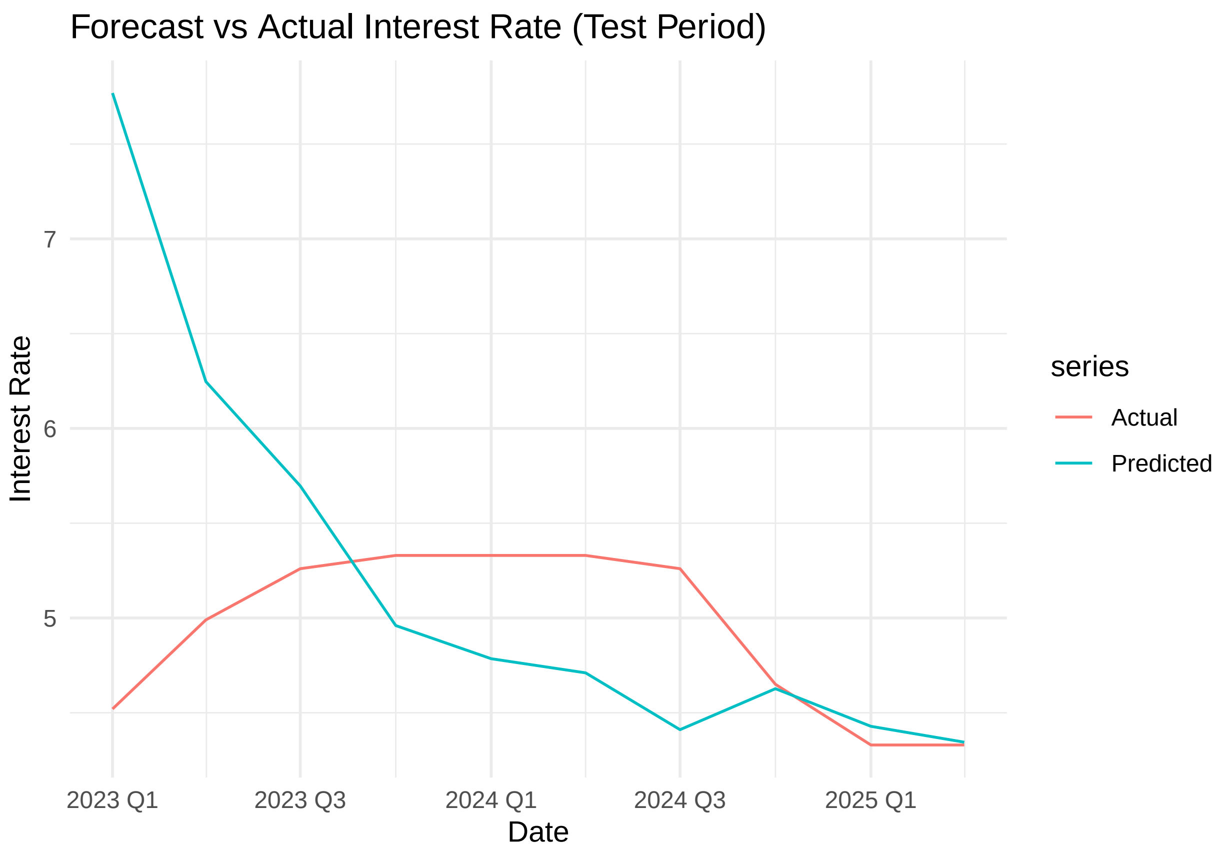
```r
taylor_train_rmse <- sqrt(mean(df_train$residuals_taylor ^ 2))
```

Our residual plot shows clear downward trend over time from 1980 to 2020, indicating that it does not exhibit white noise behavior. The histogram of residual appears to be bimodal, again indicating this is not white noise. The ACF suggests a strong AR component in our residuals, and the PACF has significant lags at 1, 2 and 3. In the context of the Taylor rule, this suggests that policy may consist of some inertia from previous periods, as opposed to being purely responsive to current economic indicators.

Our Durbin-Watson test confirms the presence of autocorrelation in our model. This OLS assumption is violated: The standard errors of coefficients may be underestimated, making t-tests unreliable. Possible remedies include adding the lagged dependent variable (e.g., interest rate smoothing in the Taylor rule) or performing ARIMA regression. On average, using our model would predict an interest rate that is has a RMSE of 2.7691966 from the actual interest rate over the period of our training data.

3. **Forecast Performance**

```r
df_test <- df_test %>%
  mutate(
    predictions_taylor = predict(taylor_model, newdata = df_test),
    residuals_taylor = interest_rate - predictions_taylor
  )

df_test %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = interest_rate, color = "Actual")) +
  geom_line(aes(y = predictions_taylor, color = "Predicted")) +
  labs(
    title = "Forecast vs Actual Interest Rate (Test Period)",
    x = "Date",
    y = "Interest Rate",
    color = "series"
  )
```



```r
taylor_test_rmse <- sqrt(mean(df_test$residuals_taylor ^ 2))
```

Our test data predicted interest rates has a RMSE of 1.1779778 from actual interest rates. Our forecast bands do include the actual interests rates over this time. However if we working for a large financial services company, we would not have a model that would enable us to reasonably manage our expectations of interest rates or trade on a yield curve.

Task 5: Cointegration and Error Correction Model (ECM) (15 Points)

5.1: Cointegration Test (5 Points)

```
eg_test <- ur.df(resid(taylor_model), type = "none", selectlags = "AIC")
summary(eg_test)
```

```
## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression none 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.9451 -0.4002 -0.0025  0.4382  5.8980 
## 
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)   
## z.lag.1    -0.07370    0.02751  -2.680  0.00811 **
## z.diff.lag  0.22061    0.07730   2.854  0.00486 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.9581 on 168 degrees of freedom
## Multiple R-squared:  0.06838,    Adjusted R-squared:  0.05729 
## F-statistic: 6.165 on 2 and 168 DF,  p-value: 0.002607
## 
## 
## Value of test-statistic is: -2.6795 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

Because the p-value is less than 0.05, we reject the null hypothesis. There is sufficient evidence to suggest that the residuals are stationary, and thus conclude there is a long-run equilibrium relationship.

5.2: Estimating the ECM via OLS (5 Points)

```
longrun_model <- lm(
  interest_rate ~ inflation_gap + output_gap,
  data = df_train
)

df_train <- df_train %>%
  mutate(
    lag_resid_longrun = lag(resid(longrun_model)),
    diff_interest_rate = c(NA, diff(interest_rate)),
    diff_inflation_gap = c(NA, diff(inflation_gap)),
    diff_output_gap = c(NA, diff(output_gap))
  )

df_test <- df_test %>%
  mutate(
    lag_resid_longrun = lag(predict(longrun_model, newdata = df_test)),
    diff_interest_rate = c(NA, diff(interest_rate)),
    diff_inflation_gap = c(NA, diff(inflation_gap)),
    diff_output_gap = c(NA, diff(output_gap))
  )

ECM <- lm(
  diff_interest_rate ~ diff_inflation_gap + diff_output_gap + lag_resid_longrun,
  data = df_train,
  na.action = na.exclude
)
summary(ECM)
```

```
##
## Call:
## lm(formula = diff_interest_rate ~ diff_inflation_gap + diff_output_gap +
##     lag_resid_longrun, data = df_train, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7522 -0.3080  0.0162  0.3147  5.1941
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.06497    0.05562  -1.168    0.244
## diff_inflation_gap  0.01595    0.11321   0.141    0.888
## diff_output_gap     0.36334    0.06917   5.253 4.52e-07 ***
## lag_resid_longrun  -0.08480    0.02038  -4.160 5.07e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.7262 on 167 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.2333, Adjusted R-squared:  0.2195
## F-statistic: 16.94 on 3 and 167 DF,  p-value: 1.183e-09
```

5.3: Compare ECM Forecasts (5 Points)

```r
df_train <- df_train %>%
  mutate(
    predictions_ecm_diff = fitted(ECM),
    predictions_ecm = cumsum(replace_na(predictions_ecm_diff, 0)) + interest_rate[1],
    residuals_ecm = interest_rate - predictions_ecm
  )

df_test <- df_test %>%
  mutate(
    predictions_ecm_diff = predict(ECM, newdata = df_test),
    predictions_ecm = cumsum(replace_na(predictions_ecm_diff, 0)) + interest_rate[1],
    residuals_ecm = interest_rate - predictions_ecm
  )

ecm_train_rmse = sqrt(mean(df_train$residuals_ecm ^ 2, na.rm = TRUE))
ecm_test_rmse = sqrt(mean(df_test$residuals_ecm ^ 2, na.rm = TRUE))

kable(
  tibble(
    Model = c("Taylor Rule (OLS)", "ECM"),
    RMSE_Train = c(taylor_train_rmse, ecm_train_rmse),
    RMSE_Test = c(taylor_test_rmse, ecm_test_rmse)
  )
)
```

Model	RMSE_Train	RMSE_Test
Taylor Rule (OLS)	2.769197	1.177978
ECM	4.892315	2.999939

Task 6: ARIMA Modeling of Taylor Rule Errors (10 Points)

```r
residual_arima_fit <- auto.arima(df_train$residuals_taylor, ic = "aic")
summary(residual_arima_fit)
```

```
## Series: df_train$residuals_taylor
## ARIMA(4,1,3)
##
## Coefficients:
##          ar1     ar2     ar3      ar4     ma1      ma2      ma3
##       0.0686  0.5525  0.3051  -0.3907  0.2247  -0.7892  -0.2623
## s.e.  0.1729  0.1076  0.1352   0.0990  0.1730   0.0937   0.1653
##
## sigma^2 = 0.7756:  log likelihood = -218.42
## AIC=452.84   AICc=453.73   BIC=477.98
##
## Training set error measures:
##                       ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.07170408 0.8599476 0.5969107 50.39576 131.0558 0.9070651
##                     ACF1
## Training set 0.01199196
```

```r
residual_arima_forecast <- forecast(residual_arima_fit, h = nrow(df_test))

df_train <- df_train %>%
  mutate(
    predictions_arima = predictions_taylor + residual_arima_fit$fitted,
    residuals_arima = interest_rate - predictions_arima
  )

df_test <- df_test %>%
  mutate(
    predictions_arima = predictions_taylor + residual_arima_forecast$mean,
    residuals_arima = interest_rate - predictions_arima
  )

arima_train_rmse <- sqrt(mean(df_train$residuals_arima ^ 2))
arima_test_rmse <- sqrt(mean(df_test$residuals_arima ^ 2))

kable(
  tibble(
    Model = c("Taylor Rule (OLS)", "ECM", "ARIMA"),
    RMSE_Train = c(taylor_train_rmse, ecm_train_rmse, arima_train_rmse),
    RMSE_Test = c(taylor_test_rmse, ecm_test_rmse, arima_test_rmse)
  )
)
```

Model	RMSE_Train	RMSE_Test
Taylor Rule (OLS)	2.7691966	1.177978
ECM	4.8923145	2.999939
ARIMA	0.8599476	3.331859

Task 7: VAR Modeling (15 Points)

7.1: Estimate a VAR Model (10 Points)

```r
var_data <- df_train %>%
  as_tibble() %>%
  dplyr::select(diff_interest_rate, diff_inflation_gap, diff_output_gap) %>%
  drop_na()

VARselect_result <- VARselect(var_data, lag.max = 10, type = "const")
selected_lag <- VARselect_result$selection[["AIC(n)"]]
var_model <- VAR(var_data, p = selected_lag, type = "const")

summary(var_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: diff_interest_rate, diff_inflation_gap, diff_output_gap
## Deterministic variables: const
## Sample size: 167
## Log Likelihood: -406.556
## Roots of the characteristic polynomial:
## 0.797 0.797 0.7941 0.7941 0.7481 0.7481 0.6037 0.6037 0.4568 0.4568 0.4215 0.4215
## Call:
## VAR(y = var_data, p = selected_lag, type = "const")
##
##
## Estimation results for equation diff_interest_rate:
## ===================================================
## diff_interest_rate = diff_interest_rate.l1 + diff_inflation_gap.l1 + diff_output_gap.l1 + diff_intere
##
##                       Estimate Std. Error t value Pr(>|t|)
## diff_interest_rate.l1  0.376236   0.082065   4.585 9.35e-06 ***
## diff_inflation_gap.l1  0.005883   0.094079   0.063 0.950219
## diff_output_gap.l1     0.045388   0.059347   0.765 0.445563
## diff_interest_rate.l2 -0.112568   0.063878  -1.762 0.080013 .
## diff_inflation_gap.l2  0.065848   0.099485   0.662 0.509030
## diff_output_gap.l2     0.090978   0.059067   1.540 0.125551
## diff_interest_rate.l3  0.078822   0.059741   1.319 0.188992
## diff_inflation_gap.l3  0.128759   0.100165   1.285 0.200559
## diff_output_gap.l3     0.002758   0.059492   0.046 0.963081
## diff_interest_rate.l4 -0.211816   0.059106  -3.584 0.000454 ***
## diff_inflation_gap.l4  0.065628   0.094364   0.695 0.487801
## diff_output_gap.l4     0.046403   0.057479   0.807 0.420741
## const                 -0.062177   0.043819  -1.419 0.157933
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 0.5482 on 154 degrees of freedom
## Multiple R-Squared: 0.2802,  Adjusted R-squared: 0.2241
## F-statistic: 4.995 on 12 and 154 DF,  p-value: 5.682e-07
##
##
## Estimation results for equation diff_inflation_gap:
## ===================================================
## diff_inflation_gap = diff_interest_rate.l1 + diff_inflation_gap.l1 + diff_output_gap.l1 + diff_inter
##
##                        Estimate Std. Error t value Pr(>|t|)
## diff_interest_rate.l1  0.1096241  0.0685891   1.598  0.11203
## diff_inflation_gap.l1  0.3700643  0.0786302   4.706 5.58e-06 ***
## diff_output_gap.l1     0.0116416  0.0496016   0.235  0.81475
## diff_interest_rate.l2 -0.0292490  0.0533886  -0.548  0.58459
## diff_inflation_gap.l2 -0.1173180  0.0831487  -1.411  0.16028
## diff_output_gap.l2     0.0737383  0.0493673   1.494  0.13731
## diff_interest_rate.l3 -0.0071183  0.0499308  -0.143  0.88682
## diff_inflation_gap.l3  0.2311771  0.0837170   2.761  0.00646 **
## diff_output_gap.l3    -0.0093832  0.0497230  -0.189  0.85057
## diff_interest_rate.l4  0.0004715  0.0494003   0.010  0.99240
## diff_inflation_gap.l4 -0.4335747  0.0788682  -5.497 1.57e-07 ***
## diff_output_gap.l4    -0.0326716  0.0480406  -0.680  0.49747
## const                 -0.0204020  0.0366234  -0.557  0.57828
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 0.4582 on 154 degrees of freedom
## Multiple R-Squared: 0.3145,  Adjusted R-squared: 0.2611
## F-statistic: 5.888 on 12 and 154 DF,  p-value: 2.289e-08
##
##
## Estimation results for equation diff_output_gap:
## ================================================
## diff_output_gap = diff_interest_rate.l1 + diff_inflation_gap.l1 + diff_output_gap.l1 + diff_interest
##
##                        Estimate Std. Error t value Pr(>|t|)
## diff_interest_rate.l1  0.264568   0.122803   2.154   0.0328 *
## diff_inflation_gap.l1 -0.094006   0.140781  -0.668   0.5053
## diff_output_gap.l1     0.049758   0.088807   0.560   0.5761
## diff_interest_rate.l2 -0.141280   0.095588  -1.478   0.1414
## diff_inflation_gap.l2  0.210139   0.148870   1.412   0.1601
## diff_output_gap.l2     0.024997   0.088388   0.283   0.7777
## diff_interest_rate.l3  0.069596   0.089397   0.779   0.4375
## diff_inflation_gap.l3  0.054334   0.149888   0.362   0.7175
## diff_output_gap.l3    -0.044595   0.089025  -0.501   0.6171
## diff_interest_rate.l4 -0.139637   0.088447  -1.579   0.1164
## diff_inflation_gap.l4 -0.203189   0.141207  -1.439   0.1522
## diff_output_gap.l4    -0.020910   0.086013  -0.243   0.8082
## const                  0.009988   0.065571   0.152   0.8791
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 0.8204 on 154 degrees of freedom
## Multiple R-Squared: 0.08641, Adjusted R-squared: 0.01522
## F-statistic: 1.214 on 12 and 154 DF,  p-value: 0.2782
##
##
##
## Covariance matrix of residuals:
##                    diff_interest_rate diff_inflation_gap diff_output_gap
## diff_interest_rate            0.30054            0.04875          0.1349
## diff_inflation_gap            0.04875            0.20994          0.1353
## diff_output_gap               0.13493            0.13528          0.6730
##
## Correlation matrix of residuals:
##                    diff_interest_rate diff_inflation_gap diff_output_gap
## diff_interest_rate             1.0000             0.1941          0.3000
## diff_inflation_gap             0.1941             1.0000          0.3599
## diff_output_gap                0.3000             0.3599          1.0000
```

```r
df_train <- df_train %>%
  mutate(
    predictions_var_diff = c(
      rep(NA, selected_lag + 1),
      fitted(var_model)[, "diff_interest_rate"]
    ),
    predictions_var = cumsum(replace_na(predictions_var_diff, 0)) + interest_rate[1],
    residuals_var = interest_rate - predictions_var
  )

df_test <- df_test %>%
  mutate(
    predictions_var_diff = predict(
      var_model,
      n.ahead = nrow(df_test)
    )$fcst$diff_interest_rate[, "fcst"],
    predictions_var = cumsum(replace_na(predictions_var_diff, 0)) + interest_rate[1],
    residuals_var = interest_rate - predictions_var
  )

var_train_rmse = sqrt(mean(df_train$residuals_var ^ 2, na.rm = TRUE))
var_test_rmse = sqrt(mean(df_test$residuals_var ^ 2, na.rm = TRUE))

kable(
  tibble(
    Model = c("Taylor Rule (OLS)", "ECM", "ARIMA", "VAR"),
    RMSE_Train = c(taylor_train_rmse, ecm_train_rmse, arima_train_rmse, var_train_rmse),
    RMSE_Test = c(taylor_test_rmse, ecm_test_rmse, arima_test_rmse, var_test_rmse)
  )
)
```

Model	RMSE_Train	RMSE_Test
Taylor Rule (OLS)	2.7691966	1.177978
ECM	4.8923145	2.999939
ARIMA	0.8599476	3.331859
VAR	2.1474112	1.564714

```
causality(var_model, cause = "diff_inflation_gap")$Granger
```

```
##
##  Granger causality H0: diff_inflation_gap do not Granger-cause
##  diff_interest_rate diff_output_gap
##
## data:  VAR object var_model
## F-Test = 1.2467, df1 = 8, df2 = 462, p-value = 0.2699
```

```
causality(var_model, cause = "diff_output_gap")$Granger
```

```
##
##  Granger causality H0: diff_output_gap do not Granger-cause
##  diff_interest_rate diff_inflation_gap
##
## data:  VAR object var_model
## F-Test = 0.73077, df1 = 8, df2 = 462, p-value = 0.6644
```

As noted earlier, all variables need to be differenced to be stationary. Following standard practice, we selected the VAR order using the AIC criterion.

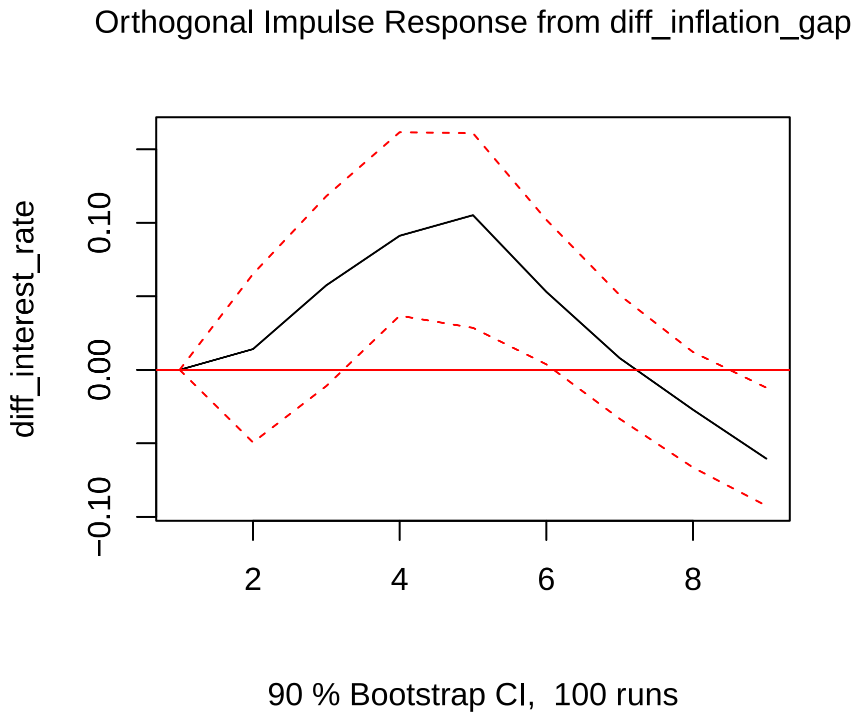
For a VAR(4) model, the first 4 observations in the training set are used to create lags and are not included in the model estimation. To properly align the fitted values with the training data, these observations are accounted for by padding. RMSE is then calculated for `interest_rate` on both the training and test sets. This provides a measure of the in-sample and out-of-sample forecast accuracy for each series.

Taylor Rule (OLS) and VAR models generalize better to the test data. The ECM performs the weakest across both sets. This pattern suggests that ARIMA may be overfitting the training data. Overall, the VAR model demonstrates the best balance between fit and generalization.

Both Granger causality tests fail to reject the null hypothesis. This means that differences in inflation gap do not Granger cause differences in interest rate and that differences in output gap do not Granger cause differences in interest rate.

Task 7.2: Impulse Response Function (IRF) (5 Points)

```
irf_result <- irf(
  var_model,
  impulse = "diff_inflation_gap",
  response = "diff_interest_rate",
  n.ahead = 8,
  boot = TRUE,
  ci = 0.90
)

plot(irf_result)
```



The impulse response plot indicates that a standard deviation shock to the differenced inflation gap leads to a 0.1% increase in the differenced interest rate after 5 quarters and a 0.06% decrease in the differenced interest rate after 8 quarters.