Final Course Project

Final Project in R Markdown

Kurtik Appadoo and Kyle Zaslaw

March 2024

Contents

1 Introduction

Forecasting key economic indicators such as Real GDP Growth, CPI (Consumer Price Index), Real Non-Residential Fixed Investment, and Average Weekly Earnings is pivotal for comprehending and guiding the economic trajectory of a country. These forecasts serve as the backbone for economic policy-making, investment decisions, and provide insights into the overall health and direction of the economy. Forecasting these economic indicators is critical for several intertwined reasons. Understanding CPI in conjunction with Real GDP Growth helps central banks and policymakers balance growth with inflation control, crucial for maintaining purchasing power and economic stability. Forecasting Real Non-Residential Fixed Investment sheds light on business sentiment and future productivity, as these investments are directly tied to capacity expansion and technological advancements, driving long-term GDP growth. Average Weekly Earnings offer insights into consumer spending potential, inflationary pressures, and the health of the labor market, which in turn influences Real GDP Growth through consumer spending.

To holistically forecast these indicators, we'll employ a few forecasting models we've studied in class as well as contribute the effect of various predictors in helping the accuracy of our model and thus forecast. Given the time series nature of these indicators, ARIMA models, augmented with seasonal adjustments, will capture underlying trends, cycles, and seasonal effects, crucial for accurate short-term and medium-term forecasts. To avoid using just one model, we'll also establish and compare Time series Linear models to encapsulate the underlying and existing linear relationships in existing economic models. As we tend to observe in most economic literature, ARIMA models or just models that make use of lagged variables tend to yield more accurate results, however some TSLM models have proven to provide useful insights, Thus we'll set up and compare both.

Accurate forecasting and analysis could unveil; How inflation dynamics (CPI) interact with GDP growth, the implications of fixed investment on future productivity and GDP, and the feedback loop between earnings growth, consumer spending, and overall economic activity. Evaluate the effectiveness of monetary and fiscal policies on stabilizing inflation, stimulating investment, and enhancing workers' earnings without overheating the economy. Identifying emerging trends and potential risks in inflation, investment, and labor markets, guiding policymakers and investors in proactive decision-making. Utilizing forecasts to fine-tune monetary policy, ensuring sustainable growth while keeping inflation in check, thus preserving the economy's purchasing power. Fiscal incentives or support measures could be designed to encourage non-residential fixed investment in sectors identified as growth drivers, enhancing long-term economic prospects. Understanding the trajectory of average weekly earnings alongside labor market trends can inform minimum wage adjustments, tax policies, and social welfare programs to support equitable growth.

In blending these forecasts, we gain a comprehensive view of the economy, allowing for nuanced policy interventions that support sustainable growth, manage inflation, encourage productive investment, and ensure the well-being of the workforce. This integrated approach enhances our ability to navigate economic complexities, leverage opportunities for growth, and mitigate potential risks.

2 Literature Review

2.0.1 Real GDP growth quarterly

Consumer Expenditure Indicates consumer confidence and disposable income, driving demand and GDP growth. Lagged variable: Identifies trends in consumer behavior over time, affecting future economic activity.

Business Investment Reflects business confidence and is crucial for productivity and expanding production capacity. Lagged variable: Gauges the delayed impact of investments on economic growth, considering the time to fruition.

Government Spending/Investment Directly boosts GDP through public services and infrastructure, stimulating economic activity. Lagged variable: Helps understand the rollout and impact of fiscal policies on future GDP growth.

Interest Rates Influence borrowing costs, spending, investment, and thereby economic growth. Affect exchange rates and export competitiveness. Lagged variable: Accounts for the time lag in monetary policy's effect on the economy, guiding future GDP predictions.

Lagged data for these predictors is crucial in forecasting real GDP growth, as it captures the time-delayed effects of economic activities and policies on the overall economy, enhancing forecast accuracy.

2.0.2 Average Weekly Earnings monthly

Employment Rate Reflects labor demand; higher employment rates can lead to wage increases due to competition for labor. Lagged Data: Wage adjustments may follow changes in employment rates as employers react to shifting labor market conditions over time.

Inflation Rate (CPI) Workers seek wages that keep pace with the cost of living, influencing wage negotiations. Lagged Data: Past inflation trends help predict wage adjustments as both employers and employees consider inflation in their wage determinations.

Labor Productivity Higher productivity can lead to wage increases as businesses generate more revenue per employee. Lagged Data: Productivity gains often translate into wage growth after businesses assess the durability of these improvements.

Industry Growth Rates Fast-growing industries may offer higher wages due to increased demand for skilled labor. Lagged Data: Wages adjust following industry growth, reflecting a period of evaluation and financial planning by businesses.

Understanding these predictors and incorporating lagged data allows for a more accurate forecast of average weekly earnings, accounting for the time it takes for economic and policy changes to fully impact wage levels.

2.0.3 CPI monthly (index)

Producer Price Index (PPI): The PPI measures the average changes in selling prices received by domestic producers for their output and is often seen as a leading indicator for CPI. Increases in PPI can indicate rising costs for producers that may be passed on to consumers, leading to higher CPI.

Lagged Data Usage: A short lag, often one to two months, is useful since changes in producer prices quickly translate to consumer prices, especially in fast-moving consumer goods.

Wage Growth: Rising wages can increase disposable income, leading to higher demand for goods and services, which can push prices up. Wage growth is a component of cost-push inflation, where the cost of production increases, leading to higher prices.

Lagged Data Usage: A lag of a few months is often considered, as changes in wage policies or labor market conditions may take time to manifest in consumer prices.

Oil Prices: Oil prices directly affect the cost of transportation and production of goods. High oil prices can lead to increased production costs for goods and services, contributing to higher CPI. Lagged Data Usage: The effect of oil price changes on CPI can be immediate or delayed, depending on the extent to which businesses absorb the increased costs before passing them on to consumers, usually considering a lag of one to three months.

Exchange Rates: The strength of a country's currency can influence inflation through import prices. A weaker currency makes imports more expensive, contributing to higher consumer prices, whereas a stronger currency can have the opposite effect. Lagged Data Usage: Exchange rate movements may take several months to influence CPI as changes in import costs are gradually passed on to consumer prices.

Economic Activity Indicators (e.g., GDP Growth, Unemployment Rate): High levels of economic activity and low unemployment can lead to increased demand for goods and services, potentially causing prices to rise if supply does not keep pace with demand.

Lagged Data Usage: These indicators may have varying lags, with GDP data often considered on a quarterly basis and unemployment rates on a monthly basis. The impact on CPI may be observed over several months as changes in economic activity affect consumer spending patterns.

2.0.4 Real non-Residential Fixed Investments growth quarterly

Interest Rates: Interest rates are a critical factor in investment decisions. Lower interest rates reduce the cost of borrowing, making it cheaper for businesses to finance new projects and investments, whereas higher rates do the opposite. Lagged Data Usage: The effect of interest rate changes on investment can have a lag, as firms adjust their investment plans and financing arrangements. A lag of a few months to a year can be considered.

Corporate Profits: Higher corporate profits can signal strong business conditions, providing firms with the internal funds needed for investment. It indicates the financial health and potential for reinvestment into productive assets. Lagged Data Usage: Investments driven by corporate profits might be observed with a lag, as decisions on allocating profits into new investments are made over time, typically considering a lag of one to two quarters.

Technological Changes and Innovations: Technological advancements can create new investment opportunities by improving the efficiency or reducing the cost of production. Businesses need to invest in new technology to remain competitive and capitalize on these advancements. Lagged Data Usage: The adoption of new technologies can have a varied lag, as firms need time to assess, plan, and implement technological investments, often over months to several years depending on the sector and technology complexity.

Depending on the number of authors, use this citation style:

- One author: Tarassow (2019) says . . .
- Two authors: Smeekes and Etienne (2018) say
- Karanasos et. all (2021) say . . .

3 Data and Time Series Characteristics

3.1 Data

3.1.1 Dependent variable: Real GDP Growth

- FEDFUNDS (Federal Funds Rate): This series is quarterly, obtained from FRED, covering the period from July 1954 to February 2024. The Federal Funds Rate is typically expressed in percentage per annum, indicating the interest rate at which depository institutions trade federal funds (balances held at Federal Reserve Banks) with each other overnight.

- Government Spending: This series is quarterly, obtained from FRED, covering the period from January 1947 to October 2023. Government spending data are usually expressed in currency terms, often in billions of USD, representing the total government expenditures within a given period.

- Investment: This series is quarterly, obtained from FRED, covering the period from January 1947 to October 2023. Investment data typically refer to gross private domestic investment and are expressed in currency terms, often in billions of USD, indicating the amount of investment in business, residential, and inventory investments within the economy.

- Real GDP (Gross Domestic Product): This series is quarterly, obtained from FRED, covering the period from January 2002 to October 2023. Real GDP is usually expressed in billions of USD (adjusted for inflation), measuring the total economic output of a country, accounting for changes in price levels or purchasing power.

- Personal Consumption Expenditures (PCE): This series is quarterly, obtained from FRED, covering the period from January 1959 to January 2024. PCE data are typically expressed in billions of USD (nominal or real), reflecting the total value of goods and services consumed by households.
- Consumer Sentiment: This series is quarterly, obtained from FRED, covering the period from November 1952 to January 2024. Consumer Sentiment is measured as an index, which reflects the overall health of the economy as perceived by consumers, based on their attitudes towards current economic conditions and future expectations.

3.1.2 Dependent variable: Average Weekly Income

- Average Weekly Earnings: This series is quarterly, obtained from FRED, covering the period from March 2006 to February 2024. The data is likely in USD, reflecting average weekly earnings across the economy.
- CPI (Consumer Price Index): The CPI series is quarterly, sourced from FRED, covering an extensive period from January 1913 to February 2024. It is measured as an index, reflecting changes in the price level of a basket of consumer goods and services.
- Labor Productivity: This series is quarterly, obtained from FRED, covering the period from April 1947 to October 2023. The data is presented as an index, indicating the efficiency of labor input in the production process.
- Unemployment Rate: The Unemployment Rate series is quarterly, obtained from FRED, covering the period from January 1948 to February 2024. The data is in percentage terms, representing the fraction of the labor force that is not currently employed but actively seeking employment.

3.1.3 Dependent variable: CPI

- CPI Data: This series represents the Consumer Price Index (CPI) for All Urban Consumers (CPI-U), which measures the average change over time in the prices paid by urban consumers for a market basket of consumer goods and services. The dataset shows monthly data points. The data is typically obtained from the Bureau of Labor Statistics (BLS). The provided dataset starts from January 1913. However, without viewing the entire dataset, the end date is not specified here. Generally, CPI data is updated monthly and can run up to the most recent full month or the previous month, depending on the dataset provided.
- PPI Data: The Producer Price Index (PPI) series measures the average change over time in the selling prices received by domestic producers for their output. The data is crucial for understanding inflation at the producer level before it impacts consumers. Typically sourced from the Bureau of Labor Statistics (BLS). Specific dates are not mentioned here, but PPI data is usually updated monthly, similar to the CPI.
- Wage Growth: This dataset likely contains information on the average change in wage and salary disbursements over time, which can indicate economic health and consumer purchasing power. Data on wage growth can be collected from various sources, including the BLS or the Bureau of Economic Analysis (BEA), depending on the specific measure of wage growth used. As with the other datasets, this typically would cover a similar range, often updated on a monthly or quarterly basis.
- Unemployment Rate (UNRATENSA): The unemployment rate data measures the percentage of the total labor force that is unemployed but actively seeking employment and willing to work. The dataset provided seems to focus on non-seasonally adjusted figures. Commonly sourced from the Bureau of Labor Statistics (BLS). This dataset is usually updated monthly and covers a range that could be similar to the other datasets mentioned.

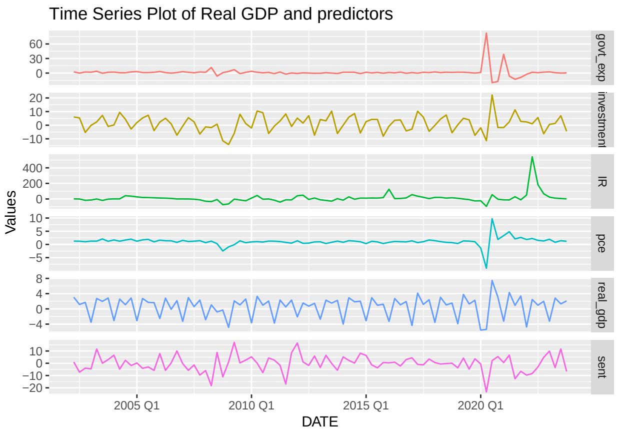
- Oil Prices: This dataset represents the changes in oil prices over time, which can significantly impact global economies and inflation rates. Oil prices are a critical component in production costs and transportation. Oil price data can be obtained from various sources, including the U.S. Energy Information Administration (EIA) or global oil market tracking organizations. Oil prices are typically tracked daily, but datasets might aggregate this information on a monthly basis to align with economic indicators.

3.1.4 Dependent variable: Real Nonresidential Fixed Investment

- Interest Rates: BOGZ1FL072052006Q: This dataset likely represents a specific economic or financial series, given the code-like name. Based on the data structure, it appears to be a quarterly series with numerical values associated with each date. Without direct reference, the source is not explicitly mentioned, but codes similar to this often originate from federal databases, possibly the Board of Governors of the Federal Reserve System (U.S. Federal Reserve) given the "BOG" prefix. Coverage Period: From July 1954 to at least July 1955 (as per the provided sample), indicating a historical series that could extend to recent years. The exact end date is not provided here, and the series is updated quarterly.
- Corporate Profits - Q0973BUSQ027NNBR: This dataset also features a code-like title, suggesting it's from a specific economic database. It contains numerical values recorded quarterly, likely representing an economic indicator. The "NNBR" in the name suggests a potential connection to the National Bureau of Economic Research (NBER) or related datasets. Starting from January 1946, this historical dataset provides quarterly data, with the sample showing data up to January 1947. The full dataset's range and update frequency would need further specifics.
- Technology Inventories - UITITI: Appears to be a monthly dataset with numerical values, potentially related to unemployment, labor, or another economic measure based on the naming convention. The specific source is not indicated by the name alone, but it may be associated with governmental or economic research databases, possibly labor or economic statistics. The dataset covers from January 1992, providing monthly updates. The sample shown extends to May 1992, without the end date of the full dataset specified.
- Real Nonresidential Fixed Investment (ND000336Q): This quarterly dataset measures real nonresidential fixed investment, capturing business investments in physical assets like buildings (excluding residential), machinery, and equipment, adjusted for inflation. While the source is not explicitly mentioned, data of this nature is often obtained from the Bureau of Economic Analysis (BEA) or similar economic statistical agencies. Starting from January 2007, with the provided sample extending to January 2008. This type of data is typically updated quarterly, reflecting changes in business investment behaviors over time.

3.2 Time Series Characteristics

Time series plots:



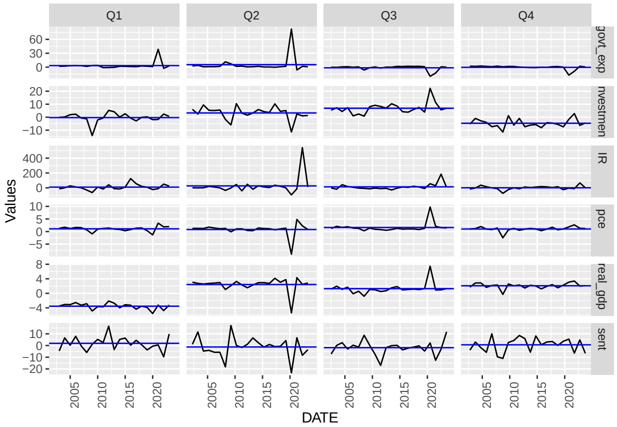
The time series of Real Gdp and its predictors are shown and after transforming the series to show growthfrom previous lag, we can observe that the provided series are all mostly stationary. We only observe a volatile change for all the series around the time of Covid.

KPSS test results for stationary:

```
## # A tibble: 6 x 3
##   Series      kpss_stat kpss_pvalue
##   <chr>           <dbl>       <dbl>
## 1 IR             0.367       0.0912
## 2 govt_exp       0.0645      0.1
## 3 investment     0.0841      0.1
## 4 pce            0.154       0.1
## 5 real_gdp       0.108       0.1
## 6 sent           0.0801      0.1
```

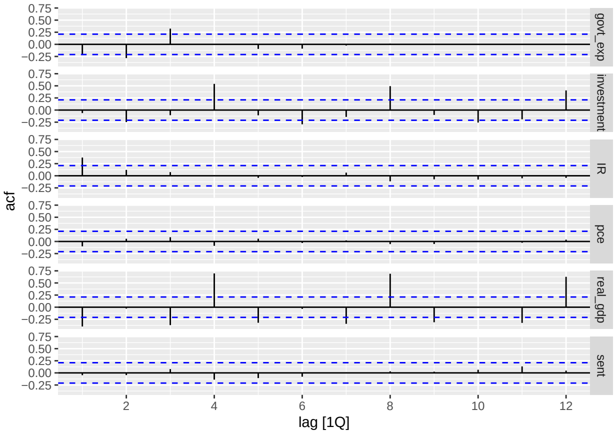
The KPSS test results indicate that the series is stationary, with a p-value of 0.1 for all variables, besides Interest Rate, which was 0.09. This suggests that no differencing might be required to achieve stationarity, which is essential for accurate forecasting

Seasonality plots:

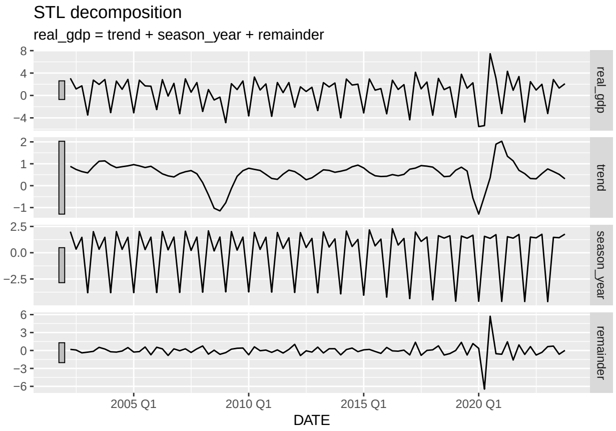


The subseries plot highlights clear seasonal patterns within the data, for government expenditure and real gdp. suggesting that incorporating seasonal components into our forecasting model could improve accuracy." However, for the other variables, it doesn't seem like there is a seasonal factor in the series.

Autocorrelation properties from correlograms:



The ACF plot shows, mostly useful for real gdp, that the current value is highly correlated with a lagged 4, 8, and 12 reading, as it would make sense from our understanding that the series is seasonal, so the data from the same quarter provides high correlation.



The STL decomposition separates the time series into trend, seasonal, and remainder components. The trend component shows a rather stationary line, except at the 2008 and covid mark from significant world events. Meanwhile the seasonal component reveals somewhat of s a seasonal trend. The remainder component, ideally resembling random noise,does for the most part besides again at the covid part.

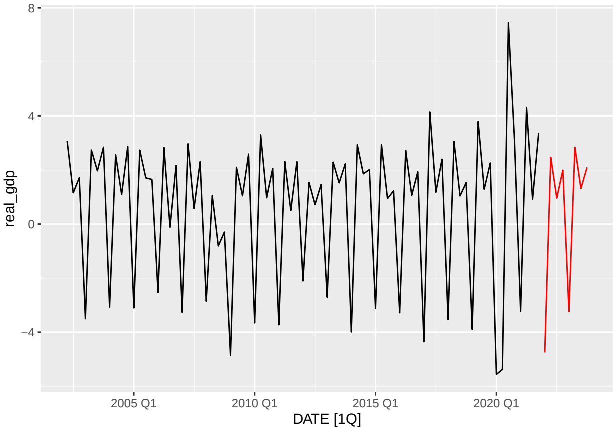
Correlations of consumption with the lags of income and sentiment

```
## # A tibble: 13 x 6
##          lag FED_rate      PCE investment govt_exp     sent
##     <cf_lag>    <dbl>    <dbl>      <dbl>    <dbl>    <dbl>
##  1        0Q  0.117    0.408       0.327  -0.362    0.0479
##  2       -1Q  0.0685  -0.0113      0.460   0.306   -0.0966
##  3       -2Q -0.00176 -0.190      -0.460   0.144    0.159
##  4       -3Q -0.0723   0.139      -0.0591 -0.163    0.0827
##  5       -4Q  0.0302  -0.0714      0.0460 -0.00334 -0.109
##  6       -5Q  0.0976   0.0850      0.472   0.0681   0.00990
##  7       -6Q  0.0111  -0.211      -0.479   0.173    0.0780
##  8      -7Q  0.0325   0.143      -0.0616 -0.225    0.0184
##  9      -8Q -0.0459  -0.00519     0.0983 -0.0306  -0.137
## 10      -9Q  0.0481   0.0478      0.471   0.0596  -0.0769
## 11     -10Q -0.00324 -0.138      -0.415   0.106    0.115
## 12     -11Q  0.00174  0.153      -0.109  -0.166    0.0912
## 13     -12Q -0.0554  -0.0196      0.0313  0.0532  -0.132
```

The CCF between real GDP and the predictors are shown. FedFUNDS interest rate had very weak correlation coefficients so we decided to drop it from being used in the models, however the other variables did show some good correlation at some lags, as we can see. Notably, from investment showing decent correlation at lag 1,4 and 9.

4 Application of Forecasting Methods

Specify training and test sets: I specified the 2021-2023 as test set and the rest as the training set



We specified the test set from 2021 onwards so that we could capture the covid event, and the training set as anything before.

Dummy Variables We include dummy variables for the year of 2008 and the first quarter of 2020 in order to capture the financial crisis of 2008 as well as the residual errors from the start of covid. Both were done in a way to capture the residual errors from those periods.

4.1 Exploring TSLM-D models

In this group I have tried these models:

```
#### Exploring TSLM-D models

fit_lm <- train |>
  model(
    lm_1 = TSLM(real_gdp ~ season()  + lag(govt_exp,6) + lag(investment,5) +
                  dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4 +
                  dummy_2020Q1 + dummy_2020Q2),
    lm_2 = TSLM(real_gdp ~ season()  + lag(govt_exp,6) + lag(investment,5) +
                  lag(pce,7) + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 +
                  dummy_2008Q4  + dummy_2020Q1 + dummy_2020Q2),
    lm_3 = TSLM(real_gdp ~ season()  + lag(govt_exp,7) + lag(investment,5) +
                  lag(govt_exp,6) + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3
                + dummy_2008Q4  + dummy_2020Q1 + dummy_2020Q2),
    lm_4 = TSLM(real_gdp ~ season()  + lag(govt_exp,6) + lag(investment,5) +
                  lag(pce, 7) + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 +
                  dummy_2008Q4  + dummy_2020Q1 + dummy_2020Q2),
    lm_5 = TSLM(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1) +
                  lag(pce, 2) + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 +
                  dummy_2008Q4  + dummy_2020Q1 + dummy_2020Q2),
    lm_6 = TSLM(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1) +
                  dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4  +
                  dummy_2020Q1 + dummy_2020Q2),
    lm_7 = TSLM(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1) +
                  lag(pce,3) + lag(sent, 2) + dummy_2008Q1 + dummy_2008Q2 +
                  dummy_2008Q3 + dummy_2008Q4  + dummy_2020Q1 + dummy_2020Q2)
  )
```

```
## # A tibble: 7 x 2
##   .model  AICc
##   <chr>  <dbl>
## 1 lm_6   -40.1
## 2 lm_7   -40.1
## 3 lm_5   -35.2
## 4 lm_1    19.9
## 5 lm_2    23.5
## 6 lm_4    23.5
## 7 lm_3    23.6
```

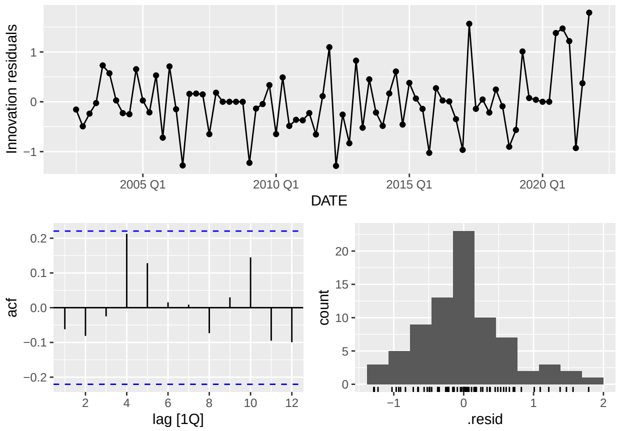
Based on the AICs of all the TSLM models, it seems the TSLM_6 performed best.

Mathematically, the chosen model is the following:

$$y_t = \beta_0 + \beta_{1,1} g_{t-4} + \beta_{1,1} i_{t-2} + \varepsilon_t$$

where ε_t is assumed to be white noise (i.e. mean zero and serially uncorrelated) and the regression also includes 3 seasonal dummies that are not shown for brevity aswell as dummy variables for the 2008 year and the first two quarters of 2020. Note that we ended up chosing y_{t-1} and s_{t-1}, which are exactly the lags where the positive correlation of these two variables with real gdp is highest as seen in the correlation table in the previous section.

Do residuals diagnostics and try to improve



The residuals seem to have an average centered at zero which is a good sign for our model along with the ACF that indicates good significance of the model. The residuals show that the model has pretty good fit with the training data but also that residuals seem to be kept in check, partially with the introduction of dummy variables taking out the events of covid and the financial crisis of 2008.

4.2 Exploring TSLM-D-ARIMA models

In this group, I will take the model chosen in the TSLM-D group as my base model and see whether fitting an ARIMA model for the error term is going to make any difference.

In this group I have tried the following specifications:

```r
fit_lm_arima <- train |>
  model(
    lm_arima_1 = ARIMA(real_gdp ~ season()  + lag(govt_exp,7) + lag(investment,5)
                       + lag(govt_exp,6)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2),
    lm_arima_2 = ARIMA(real_gdp ~ 0 + season() + pdq(1,0,2) + PDQ(2,0,1) +
                         lag(govt_exp,7) +
                         lag(investment, 5) + lag(govt_exp, 6)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2),
    lm_arima_3 = ARIMA(real_gdp ~ 0 + season() + pdq(1,0,3) + PDQ(2,0,0) +
                         lag(govt_exp,7) +
                         lag(investment, 5) + lag(govt_exp, 6)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2),
    lm_arima_4 = ARIMA(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1)
                       + lag(pce, 2)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2),
    lm_arima_5 = ARIMA(real_gdp ~ season() + 0 + lag(govt_exp,1) + lag(investment,1)
                       + pdq(0:1,0,0:2) + PDQ(0:2,0,0:2)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2)
  )
```

It seems all specifications for η_t lead to the same ARIMA model: `ARIMA(0,0,1)`

Mathematically, `ARIMA(0,0,1)` means `ARIMA()` function suggests the best model for η_t is the following

$$\eta_t = \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

where ε_t is a white noise process. Consequently, the full model becomes

$$y_t = \beta_0 + \beta_{1,1} g_{t-1} + \beta_{1,1} i_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

```
## # A tibble: 5 x 2
##   .model      AICc
##   <chr>      <dbl>
## 1 lm_arima_4  180.
## 2 lm_arima_5  204.
## 3 lm_arima_1  227.
## 4 lm_arima_2  245.
## 5 lm_arima_3  254.
```

Based on the AICs of all the TSLM-D-ARIMA models, it seems the LM_ARIMA_4 performed best. Let's pick `lm_arima_4` from this group.

4.3 ARIMA models

In this group I have tried the following specifications:

```
fit_arima <- train |> model(
  auto = ARIMA(real_gdp ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
               + dummy_2020Q1 + dummy_2020Q2),
  auto_s = ARIMA(real_gdp ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                 + dummy_2020Q1 + dummy_2020Q2, stepwise = FALSE),
  arima_p = ARIMA(real_gdp ~ season() + 0 + pdq(1,0,1) + PDQ(0,0,0)
                  + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4 +
                    dummy_2020Q1 + dummy_2020Q2),
  arima002001 = ARIMA(real_gdp ~ season() + 0 + pdq(1,0,2) + PDQ(1,0,1)
                      + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                      + dummy_2020Q1 + dummy_2020Q2),
  arima202202 = ARIMA(real_gdp ~ season() + 0 + pdq(1,0,0) + PDQ(1,0,1)
                      + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                      + dummy_2020Q1 + dummy_2020Q2),
  arima111 = ARIMA(real_gdp ~ season() + 0 + pdq(1,1,1)
                   + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4 +
                     dummy_2020Q1 + dummy_2020Q2),
  arima010 = ARIMA(real_gdp ~ season() + 0 + pdq(0,1,0)
                   + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4 +
                     dummy_2020Q1 + dummy_2020Q2),
  arima_seasonal = ARIMA(real_gdp ~ season() + 0 + PDQ(1,0,1)
                         + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                         + dummy_2020Q1 + dummy_2020Q2),
  arima_complex = ARIMA(real_gdp ~ season() + 0 + pdq(2,0,2) + PDQ(2,0,2)
                        + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                        + dummy_2020Q1 + dummy_2020Q2),
  arima_stepwise_off = ARIMA(real_gdp ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3
                             + dummy_2008Q4   + dummy_2020Q1 + dummy_2020Q2, stepwise = FALSE, approxima
  arima_non_seasonal = ARIMA(real_gdp ~ season() + 0 + pdq(0,0,2)
                             + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                             + dummy_2020Q1 + dummy_2020Q2),
  arima_seasonal_adjust = ARIMA(real_gdp ~ season() + 0 + pdq(2,0,2) + PDQ(1,0,1)
                                + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                                + dummy_2020Q1 + dummy_2020Q2)
  )
```

Tabulating the proposed specifications

Using the Akaike information criterion (AICs) we pick the winning among ARIMA

```
## # A tibble: 11 x 2
##    .model                AICc
##    <chr>                <dbl>
##  1 auto                  229.
##  2 auto_s                229.
##  3 arima_stepwise_off    229.
##  4 arima111              234.
##  5 arima_p               239.
##  6 arima_seasonal_adjust 244.
##  7 arima002001           248.
##  8 arima_complex         252.
##  9 arima010              256.
## 10 arima202202           257.
## 11 arima_non_seasonal    267.
```

Based on the AICs of all the ARIMA models, it seems the auto arima model performed best.

4.4 Comparing three models on the test set

```
fit_train <- train |>
  model(
    lm_1       = TSLM(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1)
                      + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                      + dummy_2020Q1 + dummy_2020Q2),
    lm_arima_1 = ARIMA(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1)
                       + lag(pce, 2)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2),
    auto       = ARIMA(real_gdp ~ season() +
                         dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4
                       + dummy_2020Q1 + dummy_2020Q2)
  )
```

Note that I'm manually specifying ARIMA models for `lm_arima_1` and `auto` as chosen in the previous section. Since we are again estimating on the same train set, but this would be important if we were to estimate one of these ARIMA models on the full sample. Because without these restrictions, ARIMA() function will probably choose another model, which might not be what we want.

When the three models, one from each group, are compared on the test set with respect to the RMSE measure, `lm_6` turns out to be the winner

```
## # A tibble: 3 x 10
##   .model     .type      ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr>      <chr>   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 lm_1       Test  -0.0321 0.190 0.154 -1.73   9.00   NaN   NaN -0.265
## 2 auto       Test  -0.320  0.537 0.363 -8.56  15.6    NaN   NaN  0.270
## 3 lm_arima_1 Test  -0.167  0.556 0.415 -0.321 17.1    NaN   NaN -0.293
```

The AICc values help compare the relative quality of the models while adjusting for model complexity. Lower AICc values indicate a better fit to the data, given the number of parameters. This comparison can guide the selection of the most appropriate model for forecasting real GDP, balancing fit and parsimony. By comparing all the best models from each respective model type, we see that the TSLM model performs better with the lowest RMSE value.

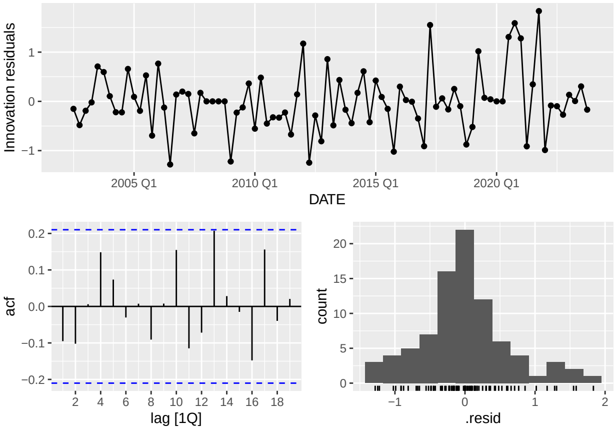
4.5 Obtaining out-of-sample forecasts

Next we estimate the winner model using the full sample

```
fit_full <- data_ts |>
  model(
    lm_1 = ARIMA(real_gdp ~ season()  + lag(govt_exp,1) + lag(investment,1) +
                   pdq(0,0,0) + PDQ(0,0,0)
                 + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4 +
                   dummy_2020Q1 + dummy_2020Q2)
  )
```

Due to a bug in the TSLM function, we use the ARIMA model with all p,d, q, P, D, and Q settings set to 0.

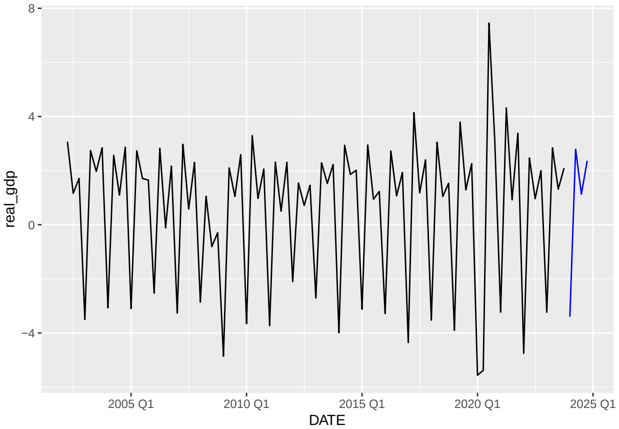
Residual diagnostics and see if you can improve model further:



How do you generate future values for predictors? I used the ARIMA() in order to generate future values for our predictors based on the historical data of those predictors and the automatic algorithm from ARIMA() that will forecast those future values.

Forecasting next 4 quarters:

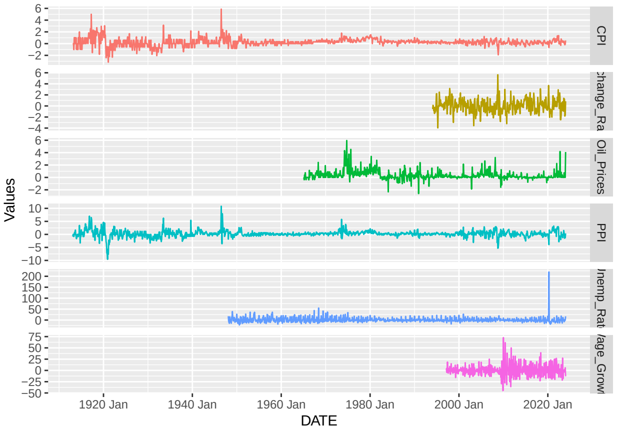
```
## # A fable: 4 x 12 [1Q]
## # Key:     .model [1]
##   .model    DATE      real_gdp .mean govt_exp investment dummy_2008Q1
##    <chr>    <qtr>        <dist> <dbl>    <dbl>      <dbl>        <dbl>
## 1 lm_1   2024 Q1 N(-3.4, 0.43) -3.40     2.31     -0.237            0
## 2 lm_1   2024 Q2  N(2.8, 0.43)  2.79     1.61      2.93             0
## 3 lm_1   2024 Q3  N(1.1, 0.43)  1.13     1.30      7.26             0
## 4 lm_1   2024 Q4  N(2.4, 0.43)  2.37     1.93     -4.58             0
## # i 5 more variables: dummy_2008Q2 <dbl>, dummy_2008Q3 <dbl>,
## #   dummy_2008Q4 <dbl>, dummy_2020Q1 <dbl>, dummy_2020Q2 <dbl>
```



#zaz stuff

4.6 Time Series Characteristics

Time series plots:



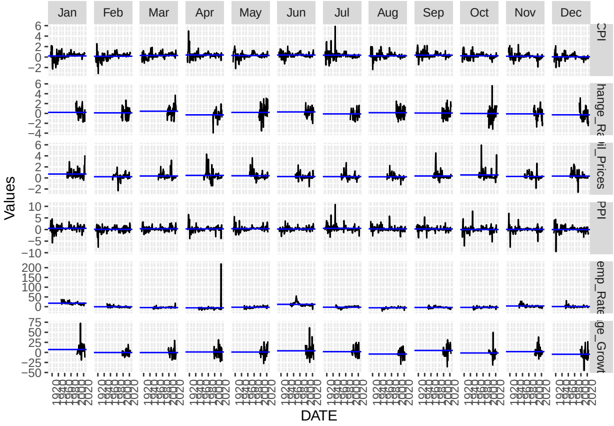
The time series plot of CPI and its predictors reveals no clear trend in the data over time, iwhile showing to be mostly stationary. While some vairables show periods of stability, others exhibit significant fluctuations, which could be considered outliers or unusual observations. These fluctuations may relate to specific economic events or changes, underlining the importance of context when interpreting the data. The overall volatility of the series is around the time of COVID

KPSS test results for stationary:

```
## # A tibble: 6 x 3
##   Series         kpss_stat kpss_pvalue
##   <chr>              <dbl>       <dbl>
## 1 CPI               0.351       0.0981
## 2 Exchange_Rates    0.130       0.1
## 3 Oil_Prices        1.31        0.01
## 4 PPI               0.107       0.1
## 5 Unemp_Rate        0.0415      0.1
## 6 Wage_Growth       0.241       0.1
```

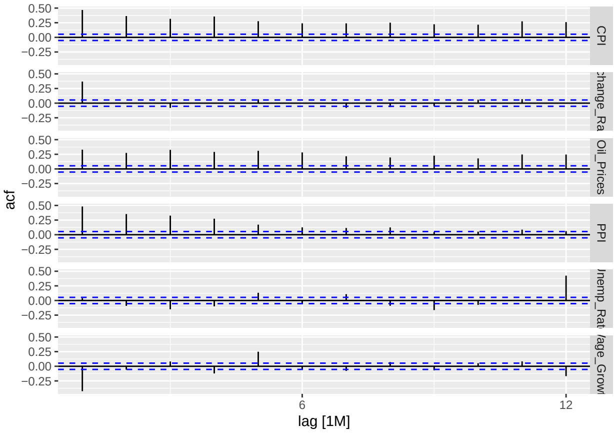
The KPSS test results indicate that the series is stationary, with a p-value around or greater than 0.1 for all variables, besides oil prices, which was .01. This suggests that no differencing might be required to achieve stationarity, which is essential for accurate forecasting.

Seasonality plots:

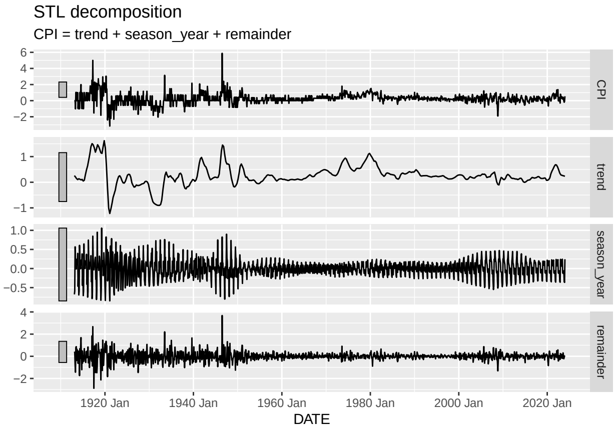


The subseries plot highlights that the data is non-seasonal for all variables.

Autocorrelation properties from correlograms:



The ACF plot shown, proves to be mostly useful for CPI. The current value is highly correlated with a lagged 4 reading. This makes sense, given the series is non-seasonal, so the lags are not correlated in a specific way.



The STL decomposition separates the time series into trend, seasonal, and remainder components. The trend component shows a rather stationary line for all years after 1950, except for subtle bumps in the covid era. While, the seasonal component reveals some seasonality that varies in magnitude. The remainder component shows that the series is stationary, similar to white noise.

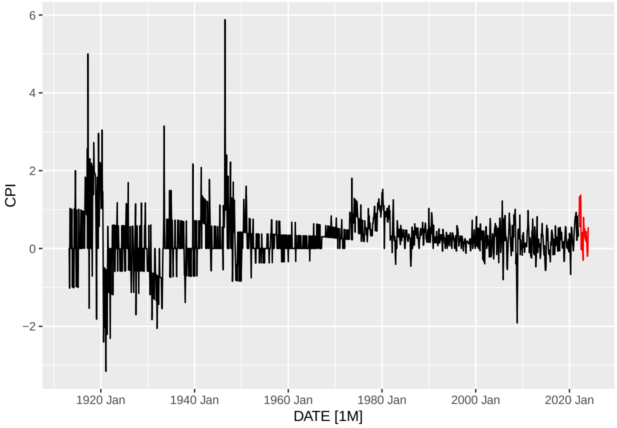
Correlations of CPI with the lags of Wage Growth, Oil Prices, PPI, Exchange Rates, and Unemployment Rates

```
## # A tibble: 13 x 6
##         lag Wage_Growth Oil_Prices   PPI Exchange_Rates Unemp_Rate
##    <cf_lag>       <dbl>      <dbl> <dbl>          <dbl>      <dbl>
##  1       0M     0.0286       0.291 0.656       -0.00966   -0.0857
##  2      -1M     0.0316       0.286 0.450       -0.171     -0.0160
##  3      -2M     0.0317       0.294 0.331       -0.146      0.0366
##  4      -3M     0.00322      0.271 0.309       -0.0466     0.0281
##  5      -4M     0.0441       0.256 0.314       -0.0326     0.0201
##  6      -5M    -0.00714      0.238 0.280       -0.0556    -0.0351
##  7      -6M     0.00563      0.191 0.188       -0.0986    -0.0730
##  8      -7M    -0.0470       0.145 0.194       -0.0537    -0.0234
##  9      -8M     0.0656       0.151 0.224        0.00432    0.0348
## 10      -9M     0.0608       0.133 0.207       -0.0172     0.00180
## 11     -10M    -0.00853      0.167 0.196        0.0237    -0.0597
## 12     -11M    -0.0159       0.125 0.237        0.0284    -0.113
## 13     -12M     0.0597       0.164 0.206        0.0612     0.0321
```

The CCF between CPI and the predictors are shown. Wage growth and exchange rates had very weak correlation coefficients so we decided to drop it from being used in the models, however the other variables did show some good correlation at some lags, as we can see. Specifically, from Oil Prices and PPI showing great correlation at lag 1 and 2.

5 Application of Forecasting Methods

Specify training and test sets: I specified the 2021-2023 as test set and the rest as the training set



We specified the test set from 2021 onwards so that we could capture the covid event, and the training set as anything before.

Dummy Variables We include dummy variables for the year of 2008 to capture the financial crisis of 2008. This was done in a way to capture the residual errors from that period.

5.1 Exploring TSLM-D models

In this group I have tried these models:

```
#### Exploring TSLM-D models

fit_lm <- train |>
  model(
    lm_1 = TSLM(CPI ~ season() + lag(Oil_Prices,2) + lag(PPI,1) + dummy_2008Q1 + dummy_2008Q2 + dummy_20
    lm_2 = TSLM(CPI ~ season() + lag(Oil_Prices,2) + lag(PPI,1) + lag(Oil_Prices,3) + lag(PPI, 2)+ dummy
    lm_3 = TSLM(CPI ~ season() + lag(Oil_Prices, 1) + lag(PPI, 3)+ dummy_2008Q1 + dummy_2008Q2 + dummy_
    lm_4 = TSLM(CPI ~ season() + lag(Oil_Prices,1) + lag(Oil_Prices, 2) + lag(PPI, 1) + lag(PPI, 2)+ du
  )
```

```
## # A tibble: 4 x 2
##   .model   AICc
##   <chr>   <dbl>
## 1 lm_4   -1675.
## 2 lm_2   -1663.
## 3 lm_1   -1645.
## 4 lm_3   -1538.
```

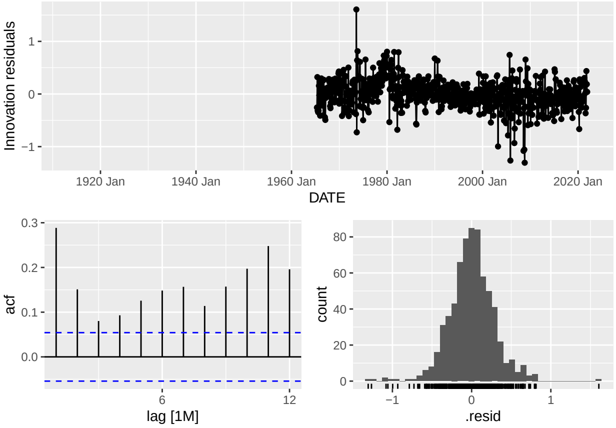
Based on the AICs of all the TSLM models, it seems that TSLM4 performed the best, though TSLM2 had a lower RMSE, so we came to the conclusion to use TSLM2.

Mathematically, the chosen model is the following:

$$y_t = \beta_0 + \beta_{1,1} g_{t-4} + \beta_{1,1} i_{t-2} + \varepsilon_t$$

where ε_t is assumed to be white noise (i.e. mean zero and serially uncorrelated) and the regression also includes 3 seasonal dummies that are not shown for brevity aswell as dummy variables for the 2008 year and the first two quarters of 2020. Note that we ended up chosing y_{t-1} and s_{t-1}, which are exactly the lags where the positive correlation of these two variables with real gdp is highest as seen in the correlation table in the previous section. KURTIK ** DO THIS FOR CPI

Do residuals diagnostics and try to improve



The residuals seem to have an average centered at zero which is a good sign for our model along with the ACF that indicates good significance of the model. The residuals show that the model has pretty good fit with the training data but also that residuals seem to be kept in check, partially with the introduction of dummy variables taking out the financial crisis of 2008.

5.2 Exploring TSLM-D-ARIMA models

In this group, I will take the model chosen in the TSLM-D group as my base model and see whether fitting an ARIMA model for the error term is going to make any difference.

In this group I have tried the following specifications:

```
fit_lm_arima <- train |>
  model(
    lm_arima_1 = ARIMA(CPI ~ season() + lag(Oil_Prices,2) + lag(PPI,1)+ dummy_2008Q1 + dummy_2008Q2 + du
    lm_arima_2 = ARIMA(CPI ~ season() + lag(Oil_Prices,2) + lag(PPI,1)
                       + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0) + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3
  )
```

Mathematically, `ARIMA(0,0,1)` means `ARIMA()` function suggests the best model for η_t is the following

$$\eta_t = \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

where ε_t is a white noise process. Consequently, the full model becomes

$$y_t = \beta_0 + \beta_{1,1} g_{t-1} + \beta_{1,1} i_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

KURTIK DO THIS **

```
## # A tibble: 2 x 2
##   .model      AICc
##   <chr>      <dbl>
## 1 lm_arima_2  162.
## 2 lm_arima_1  184.
```

Based on the AICs of all the TSLM-D-ARIMA models, it seems the LM_ARIMA_2 performed best, due to the AICc being lower.

5.3 ARIMA models

In this group I have tried the following specifications:

```
fit_arima <- train |> model(
  auto = ARIMA(CPI ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4),
  auto_s = ARIMA(CPI ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4, stepwise = FALSE),
  arima_p = ARIMA(CPI ~ season() + 0 + pdq(0,0,0) + PDQ(0:3,1,0:3) + dummy_2008Q1 + dummy_2008Q2 + dumm
  arima002001 = ARIMA(CPI ~ season() + 0 + pdq(0,0,2) + PDQ(0,0,1) + dummy_2008Q1 + dummy_2008Q2 + dumm
  arima202202 = ARIMA(CPI ~ season() + 0 + pdq(2,0,0) + PDQ(0,0,1) + dummy_2008Q1 + dummy_2008Q2 + dumm
```

Tabulating the proposed specifications

Using the Akaike information criterion (AICs) we pick the winning among ARIMA

```
## # A tibble: 4 x 2
##   .model       AICc
##   <chr>       <dbl>
## 1 auto        2118.
## 2 auto_s      2118.
## 3 arima202202 2188.
## 4 arima002001 2250.
```

Based on the AICs of all the -ARIMA models, it seems the auto arima model performed best.

5.4 Comparing three models on the test set

```
fit_train <- train |>
  model(
    lm_1       = TSLM(CPI ~ season() + 0 + lag(Oil_Prices,2) + lag(PPI,1) + lag(Oil_Prices,3) + lag(PPI
    lm_arima_2 = ARIMA(CPI ~ season() + 0 + lag(Oil_Prices,2) + lag(PPI,1)
                       + pdq(0:2, 0, 0:2) + PDQ(0:1, 0, 0)+ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3
    auto       = ARIMA(CPI ~ season() + 0 + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4)
  )
```

Note that I'm manually specifying ARIMA models for `lm_arima_1` and `auto` as chosen in the previous section. Since we are again estimating on the same train set, but this would be important if we were to estimate one of these ARIMA models on the full sample. Because without these restrictions, ARIMA() function will probably choose another model, which might not be what we want.

When the three models, one from each group, are compared on the test set with respect to the RMSE measure, `lm_6` turns out to be the winner

```
## # A tibble: 3 x 10
##   .model     .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>      <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 lm_1       Test  0.0623 0.381 0.293  344.  359.   NaN   NaN -0.166
## 2 lm_arima_2 Test  0.141  0.383 0.286  216.  227.   NaN   NaN  0.125
## 3 auto       Test  0.143  0.427 0.336  260.  274.   NaN   NaN  0.306
```

The AICc values help compare the relative quality of the models while adjusting for model complexity. Lower AICc values indicate a better fit to the data, given the number of parameters. This comparison can guide the selection of the most appropriate model for forecasting CPI, balancing fit and parsimony. By comparing all the best models from each respective model type, we see that the TSLM model performs better with the lowest RMSE value.

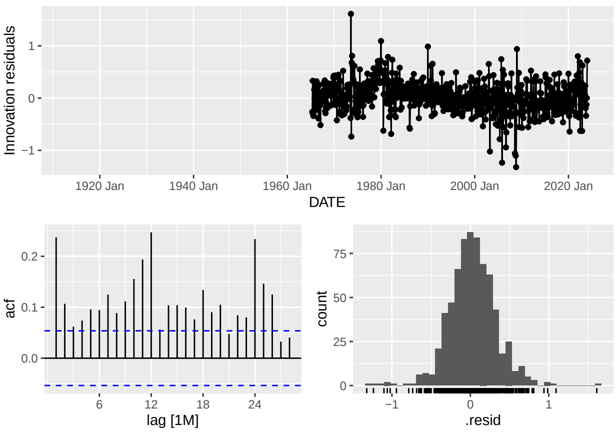
5.5 Obtaining out-of-sample forecasts

Next we estimate the winner model using the full sample

```
fit_full <- data_ts2 |>
  model(
    lm_1 = ARIMA(CPI ~ season() + 0 + lag(Oil_Prices,2) + lag(PPI,1) + lag(Oil_Prices,3) + lag(PPI, 2)
                 + pdq(0, 0, 0) + PDQ(0, 0, 0)+ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008
```

Due to a bug in the TSLM function, we use the ARIMA model with all p,d, q, P, D, and Q settings set to 0.

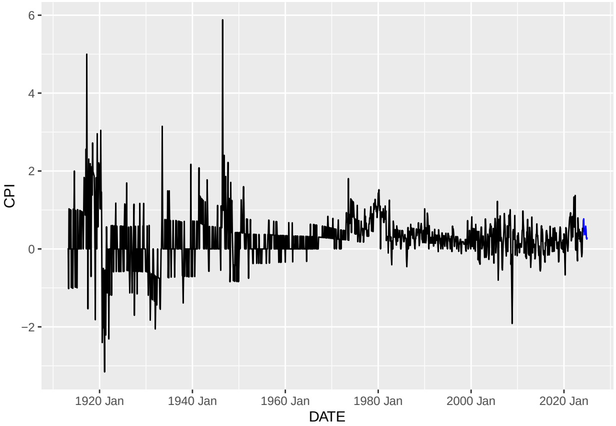
Residual diagnostics and see if you can improve model further:



How do you generate future values for predictors? I used the ARIMA() in order to generate future values for our predictors based on the historical data of those predictors and the automatic algorithm from ARIMA() that will forecast those future values.

Forecasting next 12 months:

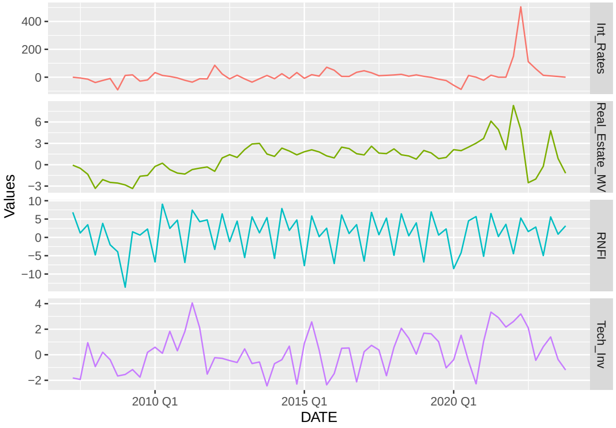
```
## # A fable: 12 x 12 [1M]
## # Key:     .model [1]
##    .model     DATE             CPI .mean Oil_Prices     PPI dummy_2008Q1
##    <chr>     <mth>          <dist> <dbl>      <dbl>   <dbl>        <dbl>
##  1 lm_1   2024 Feb N(0.34, 0.049) 0.342      1.04  -0.145            0
##  2 lm_1   2024 Mar N(0.73, 0.049) 0.731      0.807 -0.103            0
##  3 lm_1   2024 Apr N(0.77, 0.049) 0.773      1.17   0.0347           0
##  4 lm_1   2024 May N(0.45, 0.049) 0.453      1.15   0.174            0
##  5 lm_1   2024 Jun N(0.48, 0.049) 0.480      1.45   0.116            0
##  6 lm_1   2024 Jul N(0.36, 0.049) 0.364      1.54   0.138            0
##  7 lm_1   2024 Aug N(0.48, 0.049) 0.480      1.20   0.156            0
##  8 lm_1   2024 Sep N(0.58, 0.049) 0.585      1.23   0.171            0
##  9 lm_1   2024 Oct  N(0.5, 0.049) 0.496      1.30   0.183            0
## 10 lm_1   2024 Nov N(0.28, 0.049) 0.277      1.31   0.192            0
## 11 lm_1   2024 Dec N(0.25, 0.049) 0.251      1.34   0.200            0
## 12 lm_1   2025 Jan N(0.29, 0.049) 0.294      1.31   0.206            0
## # i 5 more variables: dummy_2008Q2 <dbl>, dummy_2008Q3 <dbl>,
## #   dummy_2008Q4 <dbl>, dummy_2020Q1 <dbl>, dummy_2020Q2 <dbl>
```



#RNFI

5.6 Time Series Characteristics

Time series plots:



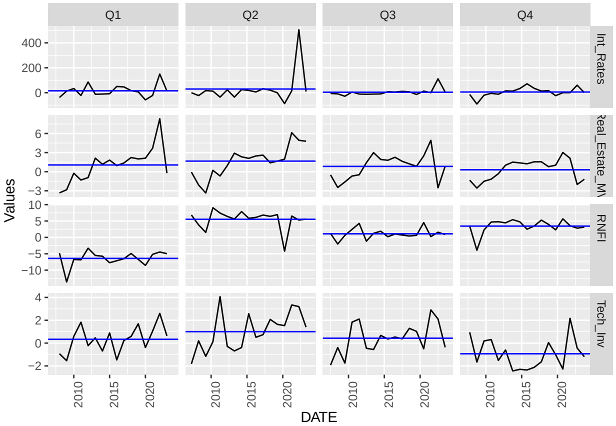
The time series plot of Real Nonresidential Fixed Investment and its predictors shows that the series is mostly stationary, except for the time period of COVID. To further analyze this, a KPSS needs to be ran to identify if the variables are stationary or not.

KPSS test results for stationary:

```
## # A tibble: 4 x 3
##   Series          kpss_stat kpss_pvalue
##   <chr>               <dbl>       <dbl>
## 1 Int_Rates           0.329         0.1
## 2 RNFI              0.0877      0.1
## 3 Real_Estate_MV    0.968       0.01
## 4 Tech_Inv          0.424       0.0669
```

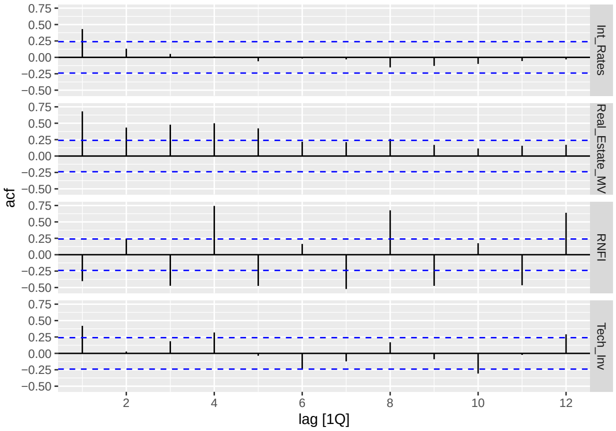
The KPSS test results indicate that the series is stationary, with a p-value around or greater than 0.1 for all variables, besides Real Estate Values, which was .01. This suggests that no differencing might be required to achieve stationarity, which is essential for accurate forecasting.

Seasonality plots:

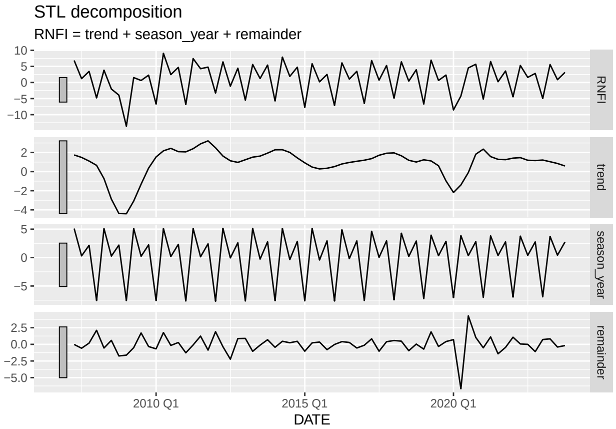


The subseries plot highlights clear seasonal patterns within the data, for RNFI and Tech_Inv, though not for Interest Rates and Real estate values. These patterns suggest that incorporating seasonal components into our forecasting model could improve accuracy for the two variables that had seasonality.

Autocorrelation properties from correlograms:



The ACF plot shown, proves to be mostly useful for RNFI. The current value is highly correlated with a lagged 4, 8, and 12 reading. This makes sense, given the series seems to be seasonal, so the data from the same quarter provides high correlation.



The STL decomposition separates the time series into trend, seasonal, and remainder components. The trend component shows no clear trend throughout the years. While, the seasonal component reveals a ton of seasonality. The remainder component shows that the series is stationary, except for the period of COVID, similar to white noise.

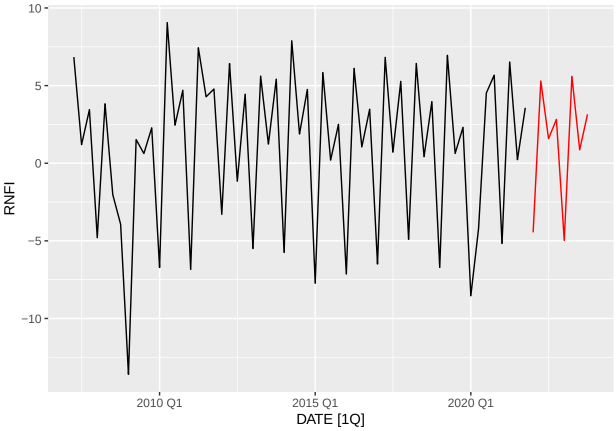
Correlations of RNFI with the lags of Interest Rates, Tech Inventories, and Real Estate Market Value

```
## # A tibble: 13 x 4
##         lag Int_Rates Tech_Inv Real_Estate_MV
##    <cf_lag>     <dbl>    <dbl>          <dbl>
##  1       0Q   0.120    0.0473         0.137
##  2      -1Q   0.0833   0.395          0.185
##  3      -2Q   0.0679  -0.117          0.110
##  4      -3Q  -0.0743  -0.0895         0.0131
##  5      -4Q   0.0582  -0.00381        0.0671
##  6      -5Q   0.0553   0.269          0.106
##  7      -6Q   0.0654  -0.0874        -0.00865
##  8      -7Q   0.0109  -0.0712        -0.0428
##  9      -8Q  -0.0380  -0.0318         0.00233
## 10      -9Q  -0.00800  0.276         -0.0527
## 11     -10Q   0.00209 -0.103         -0.0888
## 12     -11Q  -0.0235  -0.107         -0.101
## 13     -12Q  -0.0737  -0.0279        -0.0319
```

The CCF between RNFI and the predictors are shown. Interest Rates had very weak correlation coefficients so we decided to drop it from being used in the models, however the other variables did show some good correlation at some lags, as we can see. Specifically, from Tech Inventory and Real Estate Values showing great correlation at lag 1 and 5.

6 Application of Forecasting Methods

Specify training and test sets: I specified the 2021-2023 as test set and the rest as the training set



We specified the test set from 2021 onwards so that we could capture the covid event, and the training set as anything before.

Dummy Variables We include dummy variables for the year of 2008 to capture the financial crisis of 2008 and the first quarter of 2020 to capture the residual errors from the start of COVID. This was done in a way to capture the residual errors from those periods.

6.1 Exploring TSLM-D models

In this group I have tried these models:

```
#### Exploring TSLM-D models

fit_lm <- train |>
  model(
    lm_1 = TSLM(RNFI ~ season() + lag(Tech_Inv,1) + lag(Real_Estate_MV,1) + dummy_2008Q1 + dummy_2008Q2
    lm_2 = TSLM(RNFI ~ season() + lag(Tech_Inv,5) + lag(Real_Estate_MV,2) + dummy_2008Q1 + dummy_2008Q2
    lm_3 = TSLM(RNFI ~ season() + lag(Tech_Inv,1) + lag(Tech_Inv,5) + lag(Real_Estate_MV,1) + lag(Real_
    lm_4 = TSLM(RNFI ~ season() + lag(Tech_Inv,1) + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_
    lm_5 = TSLM(RNFI ~ season() + lag(Tech_Inv, 1) + lag(Tech_Inv,5) + lag(Tech_Inv,9)+ lag(Real_Estate_
```

```
## # A tibble: 5 x 2
##   .model  AICc
##   <chr>  <dbl>
## 1 lm_5    48.8
## 2 lm_2    79.6
## 3 lm_3    84.9
## 4 lm_4    84.9
## 5 lm_1    85.9
```

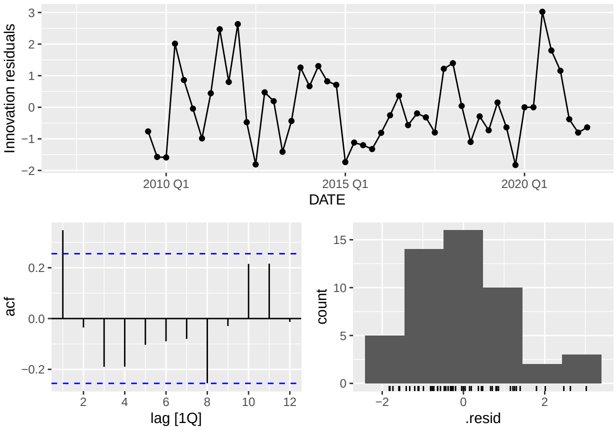
Based on the AICs of all the TSLM models, it seems that TSLM5 performed the best

Mathematically, the chosen model is the following:

$$y_t = \beta_0 + \beta_{1,1} g_{t-4} + \beta_{1,1} i_{t-2} + \varepsilon_t$$

where ε_t is assumed to be white noise (i.e. mean zero and serially uncorrelated) and the regression also includes 3 seasonal dummies that are not shown for brevity aswell as dummy variables for the 2008 year and the first two quarters of 2020. Note that we ended up chosing y_{t-1} and s_{t-1}, which are exactly the lags where the positive correlation of these two variables with real gdp is highest as seen in the correlation table in the previous section. KURTIK ** DO THIS FOR RNFI

Do residuals diagnostics and try to improve



The residual analysis reveals the average is centered around zero, though there was a lot of randomness that was shown. Overall, the residuals show that the model has a good fit with the training data, which could be due to the introduction of dummy variables during 2008 and 2020.

6.2 Exploring TSLM-D-ARIMA models

In this group, I will take the model chosen in the TSLM-D group as my base model and see whether fitting an ARIMA model for the error term is going to make any difference.

In this group I have tried the following specifications:

```
fit_lm_arima <- train |>
  model(
    lm_arima_1 = ARIMA(RNFI ~ season() + 0 + lag(Tech_Inv,1) + lag(Real_Estate_MV,1) + dummy_2008Q1 + du
    lm_arima_2 = ARIMA(RNFI ~ season() + 0 + lag(Tech_Inv,5) + lag(Real_Estate_MV,2) + dummy_2008Q1 + du
    lm_arima_3 = ARIMA(RNFI ~ season() + 0 + lag(Tech_Inv,5) + lag(Real_Estate_MV,2)
                       + pdq(1, 0, 3) + PDQ(2, 0, 0)+ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy
    lm_arima_4 = ARIMA(RNFI ~ season() + 0 + lag(Tech_Inv,1) + lag(Real_Estate_MV,1)
                       + pdq(1, 0, 2) + PDQ(2, 0, 1)+ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy
```

Mathematically, `ARIMA(0,0,1)` means `ARIMA()` function suggests the best model for η_t is the following

$$\eta_t = \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

where ε_t is a white noise process. Consequently, the full model becomes

$$y_t = \beta_0 + \beta_{1,1} g_{t-1} + \beta_{1,1} i_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

KURTIK DO THIS **

```
## # A tibble: 4 x 2
##   .model      AICc
##   <chr>      <dbl>
## 1 lm_arima_2  220.
## 2 lm_arima_3  235.
## 3 lm_arima_1  247.
## 4 lm_arima_4  257.
```

Based on the AICs of all the TSLM-D-ARIMA models, it seems the LM_ARIMA_2 performed best, due to the AICc being lower.

6.3 ARIMA models

In this group I have tried the following specifications:

```
fit_arima <- train |> model(
  auto = ARIMA(RNFI ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4   + dummy_2020Q1 + dumm
  auto_s = ARIMA(RNFI ~ dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4   + dummy_2020Q1 + du
  arima_p = ARIMA(RNFI ~ season() + 0 + pdq(1,0,1) + PDQ(0,0,0)
                  + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4   + dummy_2020Q1 + dummy_2
  arima002001 = ARIMA(RNFI ~ season() + 0 + pdq(1,0,2) + PDQ(1,0,1)
                      + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4   + dummy_2020Q1 + du
  arima202202 = ARIMA(RNFI ~ season() + 0 + pdq(1,0,0) + PDQ(1,0,1)
                      + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4   + dummy_2020Q1 + du
```

Tabulating the proposed specifications

Using the Akaike information criterion (AICs) we pick the winning among ARIMA

```
## # A tibble: 5 x 2
##   .model       AICc
##   <chr>       <dbl>
## 1 auto         237.
## 2 auto_s       237.
## 3 arima202202  238.
## 4 arima_p      246.
## 5 arima002001  249.
```

Based on the AICs of all the -ARIMA models, it seems the auto arima model performed best.

6.4 Comparing three models on the test set

```
fit_train <- train |>
  model(
    lm_1       = TSLM(RNFI ~ season() + 0 + lag(Tech_Inv,1) + lag(Real_Estate_MV,1) + lag(Tech_Inv, 5) +
                      + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4 + dummy_2020Q1 + dumm
    lm_arima_1 = ARIMA(RNFI ~ season() + 0 + lag(Tech_Inv,1) + lag(Real_Estate_MV,1) + lag(Tech_Inv, 5)
                       + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4  + dummy_2020Q1 + du
    auto       = ARIMA(RNFI ~ season() + 0 +
                       dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4  + dummy_2020Q1 + dumm
  )
```

Note that I'm manually specifying ARIMA models for `lm_arima_1` and `auto` as chosen in the previous section. Since we are again estimating on the same train set, but this would be important if we were to estimate one of these ARIMA models on the full sample. Because without these restrictions, ARIMA() function will probably choose another model, which might not be what we want.

***IS THIS THE SAME? ˆ and below

When the three models, one from each group, are compared on the test set with respect to the RMSE measure, `lm_6` turns out to be the winner

```
## # A tibble: 3 x 10
##   .model     .type      ME  RMSE   MAE    MPE  MAPE  MASE RMSSE    ACF1
##   <chr>      <chr>   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 lm_arima_1 Test    0.888  1.37  1.01   16.7  44.0   NaN   NaN -0.761
## 2 auto       Test   -0.431  1.40  1.29  -42.1  45.2   NaN   NaN -0.342
## 3 lm_1       Test   -1.91   2.04  1.91 -105.  105.    NaN   NaN -0.0135
```

The AICc values help compare the relative quality of the models while adjusting for model complexity. Lower AICc values indicate a better fit to the data, given the number of parameters. This comparison can guide the selection of the most appropriate model for forecasting RNFI, balancing fit and parsimony. By comparing all the best models from each respective model type, we see that the TSLM-D-ARIMA model performs better with the lowest RMSE value.

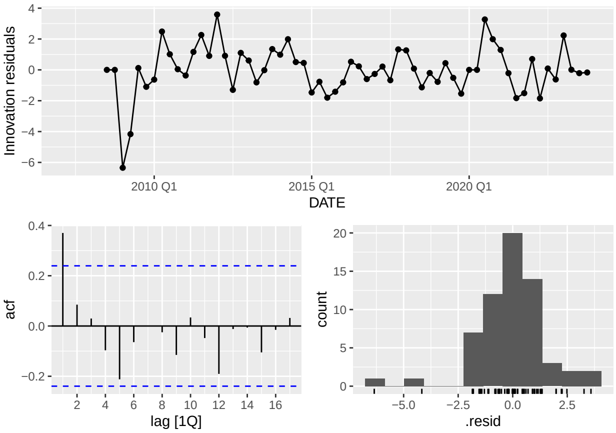
6.5 Obtaining out-of-sample forecasts

Next we estimate the winner model using the full sample

```
fit_full <- data_ts3 |>
  model(
    lm_arima_1 = ARIMA(RNFI ~ season()  + lag(Tech_Inv,1) + lag(Real_Estate_MV,1) + lag(Tech_Inv, 5) +
                   pdq(0,0,0) + PDQ(0,0,0)
                 + dummy_2008Q1 + dummy_2008Q2 + dummy_2008Q3 + dummy_2008Q4   + dummy_2020Q1 + dummy_2
  )
```

Due to a bug in the TSLM function, we use the ARIMA model with all p,d, q, P, D, and Q settings set to 0.

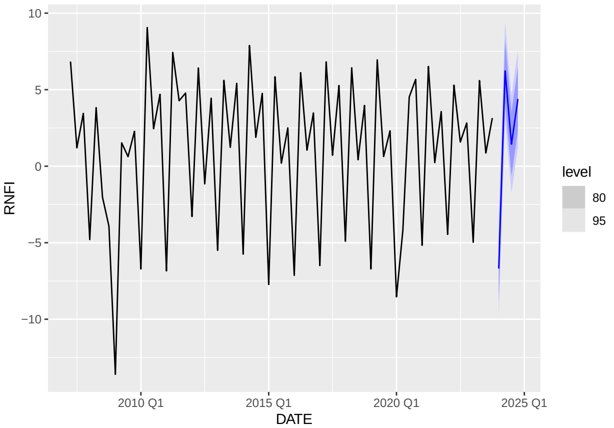
Residual diagnostics and see if you can improve model further:



How do you generate future values for predictors? I used the ARIMA() in order to generate future values for our predictors based on the historical data of those predictors and the automatic algorithm from ARIMA() that will forecast those future values. ******

Forecasting next 4 quarters:

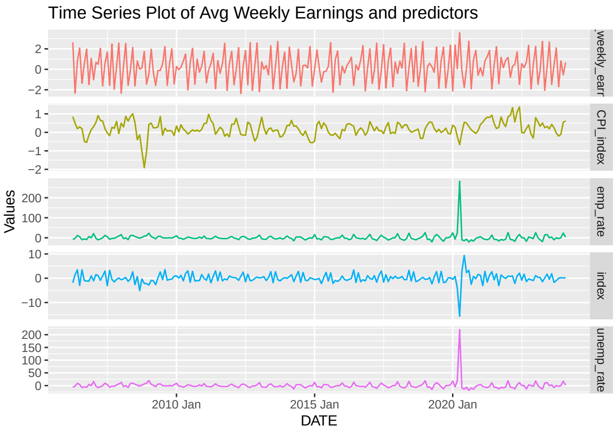
```
## # A fable: 4 x 12 [1Q]
## # Key:     .model [1]
##   .model        DATE          RNFI .mean Real_Estate_MV Tech_Inv dummy_2008Q1
##   <chr>        <qtr>        <dist> <dbl>          <dbl>    <dbl>        <dbl>
## 1 lm_arima_1 2024 Q1 N(-6.7, 2.6) -6.69          1.47    0.729            0
## 2 lm_arima_1 2024 Q2  N(6.2, 2.6)  6.22          0.976   2.41             0
## 3 lm_arima_1 2024 Q3  N(1.4, 2.6)  1.44          1.15    1.50             0
## 4 lm_arima_1 2024 Q4  N(4.4, 2.6)  4.39          1.09   -0.0125           0
## # i 5 more variables: dummy_2008Q2 <dbl>, dummy_2008Q3 <dbl>,
## #   dummy_2008Q4 <dbl>, dummy_2020Q1 <dbl>, dummy_2020Q2 <dbl>
```



7 Average Weekly Earning

7.1 Time Series Characteristics

Time series plots:



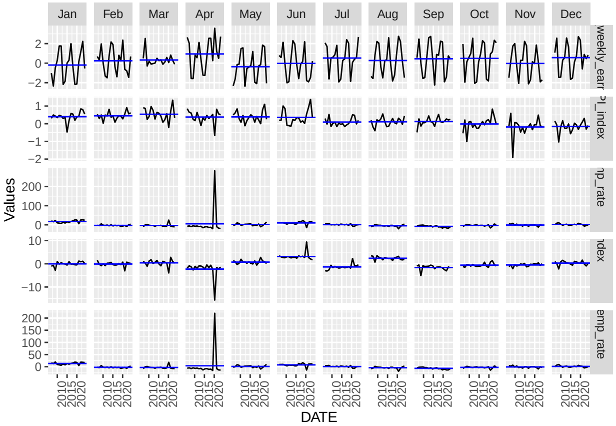
The time series of average weekly earnings and its predictors are shown and after transforming the series to show growth from previous lag, we can observe that the provided series are all mostly stationary. We only observe a volatile change for all the series around the time of Covid.

KPSS test results for stationary:

```
## # A tibble: 5 x 3
##   Series              kpss_stat kpss_pvalue
##   <chr>                   <dbl>       <dbl>
## 1 CPI_index              0.329          0.1
## 2 avg_weekly_earnings    0.168          0.1
## 3 emp_rate               0.0589         0.1
## 4 index                  0.0471         0.1
## 5 unemp_rate             0.0504         0.1
```

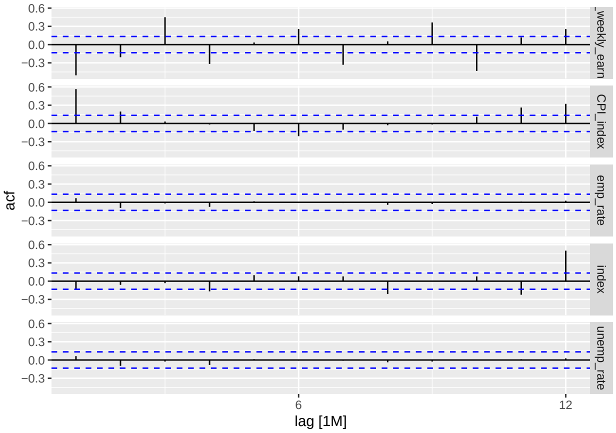
The KPSS test results indicate that the series is stationary, with a test statistic of 0.1 for all variables. This suggests that no differencing might be required to achieve stationarity, which is essential for accurate forecasting.

Seasonality plots:

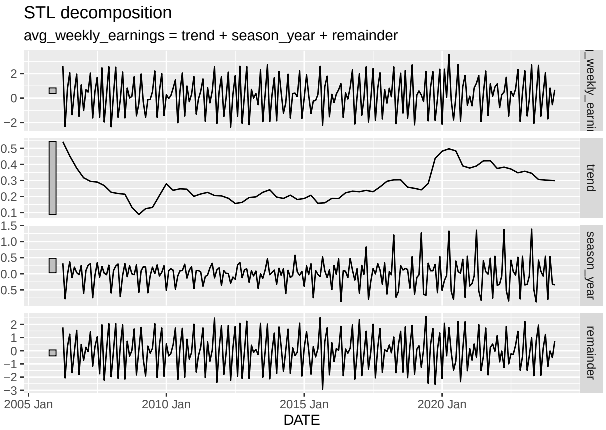


The subseries plot highlights clear seasonal patterns within the data, for cpi and average weekly earnings. suggesting that incorporating seasonal components into our forecasting model could improve accuracy." However, for the other variables, it doesn't seem like there is a seasonal factor in the series.

Autocorrelation properties from correlograms:



The ACF plot shows, mostly useful for average weekly earnings, that the current value is highly correlated with a lagged 3,6 and 10 reading, as it would make sense from our understanding that the series is seasonal.



The STL decomposition separates the time series into trend, seasonal, and remainder components. The trend component shows a rather stationary line, except at the 2008 and covid mark from significant world events. Meanwhile the seasonal component reveals somewhat of a seasonal trend, that increases in magnitude in most recent years. The remainder component, ideally resembling random noise,does for the most part.

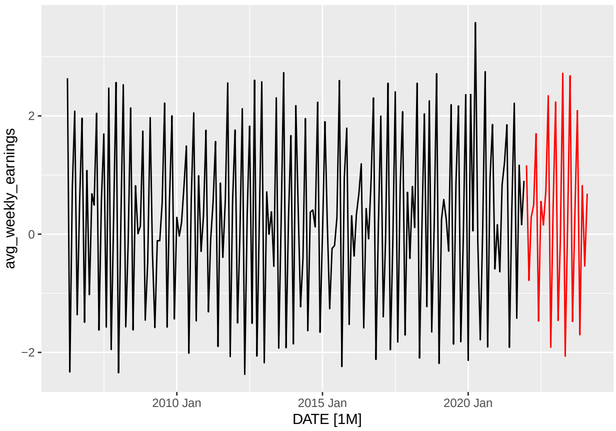
Correlations of consumption with the lags of income and sentiment

```
## # A tibble: 25 x 5
##          lag      CPI Unemp_rate Emp_rate Ind_index
##     <cf_lag>    <dbl>      <dbl>    <dbl>     <dbl>
##  1        0M -0.0117     0.0783   0.0780   -0.0786
##  2       -1M -0.0110     0.00485  0.00501   0.0290
##  3       -2M  0.0633    -0.0841  -0.0836    0.0513
##  4       -3M -0.00159    0.0217   0.0250    0.0229
##  5       -4M -0.00758    0.0899   0.0888    0.0430
##  6       -5M  0.0694    -0.118   -0.118    -0.0445
##  7       -6M  0.0667     0.0254   0.0262    0.0206
##  8       -7M  0.0237     0.0421   0.0411   -0.0524
##  9       -8M -0.0198    -0.0124  -0.0118   -0.00568
## 10       -9M -0.0259     0.0220   0.0236    0.0725
## # i 15 more rows
```

The CCF between average weekly earnings and the predictors are shown. Most predictors had weak correlation across the board, indicating that perhaps an arima model using lagged instances of the dependent variable might perform best. The best correlations came from a newly introduced predictor for industry index.

8 Application of Forecasting Methods

Specify training and test sets: I specified the 2021-2023 as test set and the rest as the training set



We specified the test set from 2021 onwards so that we could capture the covid event, and the training set as anything before.

8.1 Exploring TSLM-D models

In this group I have tried these models:

```
#### Exploring TSLM-D models

fit_lm <- train |>
  model(
    lm_1 = TSLM(avg_weekly_earnings ~ season() + lag(CPI_index,5) + lag(emp_rate,4) + lag(unemp_rate,4)
    lm_2 = TSLM(avg_weekly_earnings ~ season() + lag(CPI_index,5) + lag(CPI_index,6) + lag(unemp_rate, 
    lm_3 = TSLM(avg_weekly_earnings ~ season() + lag(CPI_index,5)),
    lm_4 = TSLM(avg_weekly_earnings ~ season() + lag(index, 9) + lag(CPI_index,5) + lag(CPI_index,2)),
    lm_5 = TSLM(avg_weekly_earnings ~ season() + lag(unemp_rate, 4)),
    lm_6 = TSLM(avg_weekly_earnings ~ season() + lag(index, 9) + lag(CPI_index,5))
  )
```

```
## # A tibble: 6 x 2
##   .model  AICc
##   <chr>  <dbl>
## 1 lm_6    160.
## 2 lm_3    162.
## 3 lm_1    163.
## 4 lm_4    163.
## 5 lm_5    164.
## 6 lm_2    165.
```

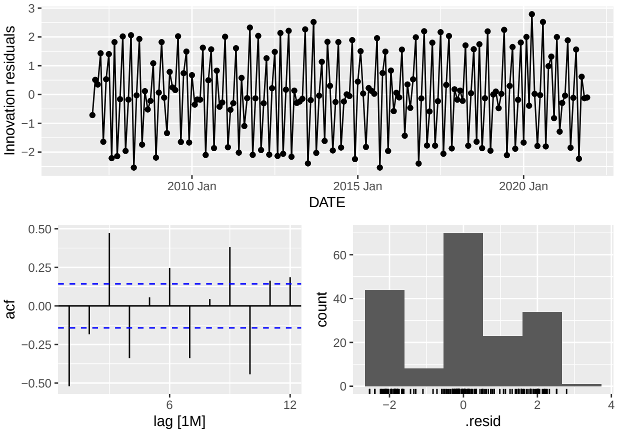
Based on the AICs of all the TSLM models, it seems the TSLM6 performed best, using only a lagged instance of CPI and industry index.

Mathematically, the chosen model is the following:

$$y_t = \beta_0 + \beta_{1,1} i_{t-9} + \beta_{1,1} c_{t-5} + \varepsilon_t$$

where ε_t is assumed to be white noise (i.e. mean zero and serially uncorrelated) and the regression also includes 3 seasonal dummies that are not shown for brevity .

Do residuals diagnostics and try to improve



Our residual analysis suggests that our residuals are situated at mean zero so that is a good sign. However we then notice a huge autocorrelation in the ACF graph which isn't very good when justifying our forecast.

8.2 Exploring TSLM-D-ARIMA models

In this group, I will take the model chosen in the TSLM-D group as my base model and see whether fitting an ARIMA model for the error term is going to make any difference.

In this group I have tried the following specifications:

```
fit_lm_arima <- train |>
  model(
    lm_arima_1 = ARIMA(avg_weekly_earnings ~ season() + 0 + lag(CPI_index,5) + lag(emp_rate,4) + lag(un
                          pdq(0:3,0,0:2) + PDQ(0:1,0,0:1)),
    lm_arima_2 = ARIMA(avg_weekly_earnings ~ season() + 0 + lag(CPI_index,5) + lag(emp_rate,4) + lag(un
                         stepwise = FALSE),
    lm_arima_3 = ARIMA(avg_weekly_earnings ~ season() + 0 + lag(CPI_index,5) + lag(emp_rate,4) + lag(un
                          pdq(2:3,0,0:1) + PDQ(0:2,0,0:1)),
    lm_arima_4 = ARIMA(avg_weekly_earnings ~ season() + lag(index, 9) + lag(CPI_index,5))
  )
```

```
## # A tibble: 4 x 2
##   .model      AICc
##   <chr>      <dbl>
## 1 lm_arima_4  523.
## 2 lm_arima_2  531.
## 3 lm_arima_3  538.
## 4 lm_arima_1  542.
```

Based on the AICs of all the TSLM-D-ARIMA models, it seems the LM_ARIMA_2 performed best.

8.3 ARIMA models

In this group I have tried the following specifications:

```r
fit_arima <- train |> model(
  auto = ARIMA(avg_weekly_earnings ~ season() + 0 ),
  auto_s = ARIMA(avg_weekly_earnings ~ season() + 0 , stepwise = FALSE),
  arima_p = ARIMA(avg_weekly_earnings ~ season() + 0 + pdq(0,0,0) + PDQ(0:3,0,0:2)),
  arima1 = ARIMA(avg_weekly_earnings ~ season() + 0 + pdq(0:3,0,0:1) + PDQ(0:1,0,0:1)),
  arima2 = ARIMA(avg_weekly_earnings ~ season() + 0 + pdq(0:2,0,0:2) + PDQ(0:1,0,0:1)),
  arima3 = ARIMA(avg_weekly_earnings ~ season() + 0  +
                     pdq(2:3,0,0:1) + PDQ(0:2,0,0:1))
  )
```

Tabulating the proposed specifications

Using the Akaike information criterion (AICs) we pick the winning among ARIMA

```
## # A tibble: 6 x 2
##   .model   AICc
##   <chr>   <dbl>
## 1 auto_s   542.
## 2 auto     549.
## 3 arima2   549.
## 4 arima1   551.
## 5 arima3   551.
## 6 arima_p  683.
```

Based on the AICs of all the -ARIMA models, it seems the auto_s arima model performed best.

8.4 Comparing three models on the test set

```r
fit_train <- train |>
  model(
    lm_1     =  TSLM(avg_weekly_earnings ~ season() + lag(index, 9) + lag(CPI_index,5)),
    lm_arima_1 = ARIMA(avg_weekly_earnings ~ season() + lag(index, 9) + lag(CPI_index,5)),
    auto_s = ARIMA(avg_weekly_earnings ~ season() + 0 , stepwise = FALSE)
  )
```

When the three models, one from each group, are compared on the test set with respect to the RMSE measure, `lm_6` turns out to be the winner

```
## # A tibble: 3 x 10
##   .model     .type      ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>      <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 auto_s     Test   0.0843  1.29  1.06  62.1  92.2   NaN   NaN -0.534
## 2 lm_arima_1 Test  -0.0244  1.39  1.14  60.3 112.    NaN   NaN -0.492
## 3 lm_1       Test  -0.0365  1.63  1.32  98.6 104.    NaN   NaN -0.486
```

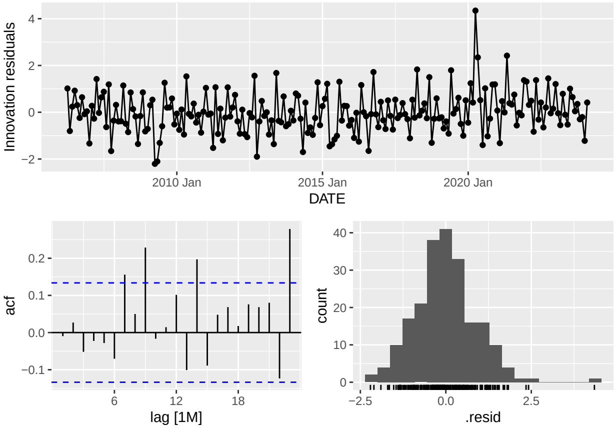
The AICc values help compare the relative quality of the models while adjusting for model complexity. Lower AICc values indicate a better fit to the data, given the number of parameters. This comparison can guide the selection of the most appropriate model for forecasting average weekly earnings, balancing fit and parsimony. By comparing all the best models from each respective model type, we see that the auto_s model performs better with the lowest RMSE value.

8.5 Obtaining out-of-sample forecasts

Next we estimate the winner model using the full sample

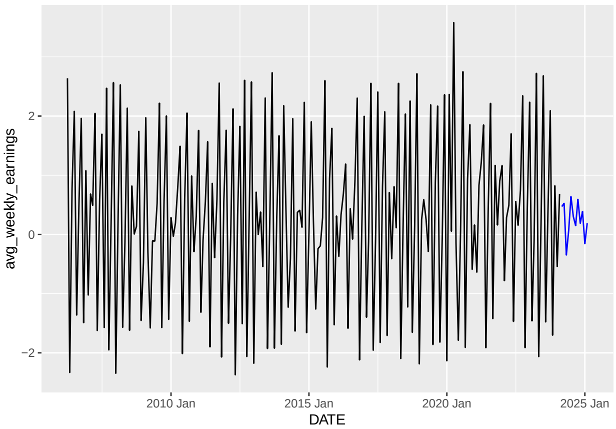
```
fit_full <- data_ts4 |>
  model(
    auto_s = ARIMA(avg_weekly_earnings ~ season() + 0 , stepwise = FALSE)
  )
```

Due to a bug in the TSLM function, we use the ARIMA model with all p,d, q, P, D, and Q settings set to 0. Residual diagnostics and see if you can improve model further:



Forecasting next 12 months for average weely earnings:

```
## # A fable: 12 x 4 [1M]
## # Key:     .model [1]
##    .model     DATE avg_weekly_earnings   .mean
##    <chr>     <mth>              <dist>   <dbl>
##  1 auto_s 2024 Mar       N(0.47, 0.86)  0.471
##  2 auto_s 2024 Apr        N(0.53, 1.5)  0.525
##  3 auto_s 2024 May       N(-0.35, 1.5) -0.347
##  4 auto_s 2024 Jun       N(0.069, 1.8)  0.0691
##  5 auto_s 2024 Jul        N(0.64, 1.9)  0.640
##  6 auto_s 2024 Aug         N(0.3, 1.9)  0.303
##  7 auto_s 2024 Sep        N(0.15, 1.9)  0.148
##  8 auto_s 2024 Oct          N(0.59, 2)  0.593
##  9 auto_s 2024 Nov        N(0.19, 2.1)  0.189
## 10 auto_s 2024 Dec        N(0.39, 2.1)  0.388
## 11 auto_s 2025 Jan       N(-0.16, 2.2) -0.157
## 12 auto_s 2025 Feb        N(0.19, 2.2)  0.188
```



The final graph visualizes forecasts for future periods based on the selected model, using historical data on the dependent variable, we were able to forecast for the next 12 months as observed on our graph.

9 Conclusion

10 Limitations and Conclusion

10.1 Limitations

When forecasting Real GDP, we've come to face some limitations in our methodology and execution methods. These limitations arose from either the choice of series in terms of variable, historical data,as well as how they correlate with the dependent variable, in this case Real GDP. This was the case for CPI as well. One of the challenges was the choice of models. For most, we use the AIC criteria as means to compare similar structured models and pick the winning one in that category. However, when analyzing the accuracy of those models, we came to notice that it wasn't always the case that the winning model had the lowest RMSE value. We use AIC and RMSE as a means to compare and pick models, but when their results seem to oppose, we would tend to pick the model based on AIC, due to its relative superiority. Perhaps this wasn't the best choice in hindsight, so that would definitely be something to look into for future forecasting projects. Another issue we came across in forecasting our models was dealing with major world events that affected the series in an unpredictable manner. Events like the 2008 financial crisis or COVID, gave our series very sudden and unpredictable spikes at some time intervals which made our training set not entirely predictable. To deal with those complications, we included dummy variables to capture the events in 2008 and early 2020. However, just by viewing the time series, it was hard to get a full grasp on all major events that would have affected the series and therefore a lot was not captured in dummy variables and just left in the series. Additionally, adding too many dummy variables would've limited the data we were working with so that was another complication. When working with CPI and RNFI, we realized that some of the predictors were not useful indicators for forecasting. Specifically, when we checked for correlation for CPI predictors, the lag values were not greater than 0.1 for Wage Growth, Unemployment Rates, and Exchange Rates. Because of these results, we decided it was best to stick with PPI and Oil Prices as predictors, which had great correlation among different lags. When checking for correlation between RNFI and its predictors, we came across the same problem, and decided it was best to exclude Interest Rates in the future models. Overall, the decision of leaving out these predictors was made in order to successfully forecast the dependent variables in the best way possible, avoiding fluctuations in data, and decreasing the complexity of the models, leading to a lower AICc.

10.2 Conclusion

Forecasting key economic indicators—Real GDP, CPI, Average Weekly Earnings, and Real Nonresidential Fixed Investment—is crucial for shaping economic policies and investment decisions. This complex task hinges on analyzing vast amounts of data and understanding the interplay between various economic forces. We employed a range of methodologies, including time series analysis and econometric modeling to predict future economic conditions. In the end, forecasting remains challenging due to the unpredictable nature of economic shocks, such as geopolitical events or pandemics. Economic forecasting is indispensable for navigating the future economic landscape. It requires a balance of technical expertise and an understanding of global dynamics, with the aim of supporting informed decision-making and fostering a stable economic environment.