STAT 8003 Project – US Core CPI

Background of the dataset

The Core CPI in the United States, with the full name of Consumer Price Index for All Urban Consumers: All Items Less Food and Energy in U.S. City Average (CPILFENS)[1], is a key US Economic Data which drove US Federal Reserve (Fed) Monetary Policy, and thus the short-term interest rate in the US and in the world. The Core US CPI, and more so the all-Items measures (with food and energy items), has exhibited great volatility since late 2021. The CPI was consistently higher than expected, which was also persistently above Fed's desired range, and has led to the quickest rate hikes cycle in recent years. Therefore, it is interesting to look at the data from a time series perspective and see could we add new insights to this important economic data.

Preview of the dataset

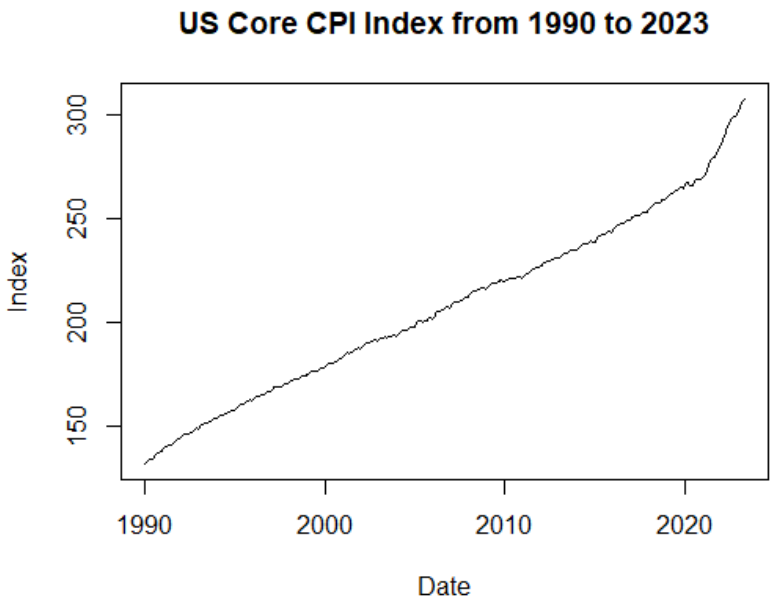

Figure 1: US Core CPI Index, Unadjusted (non-seasonally adjusted), 1990-2023

The Core CPI is an unadjusted (non-seasonally adjusted) data that measures all items excluding food and energy items for all urban consumers. In Figure 1, It exhibits a seemingly straight-line trend since 1990

[1] U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: All Items Less Food and Energy in U.S. City Average [CPILFENS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPILFENS, December 2, 2023.

until 2020, after which the slope apparently steepened. The series looks smooth but first glance but looks closely it has a lot of choppiness indicating that there could be seasonality in the data set.

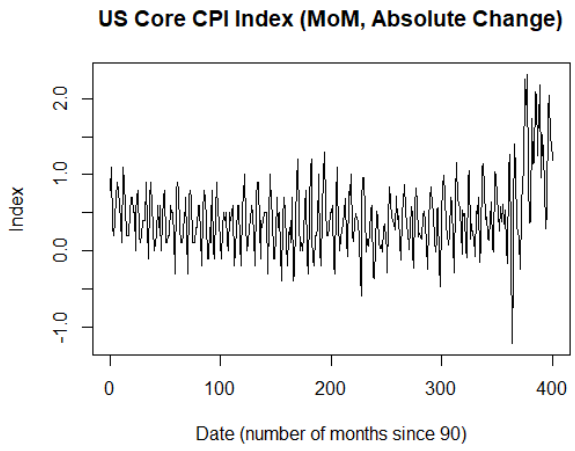

Figure 2: Absolute change in CPI, Month on Month (MoM)

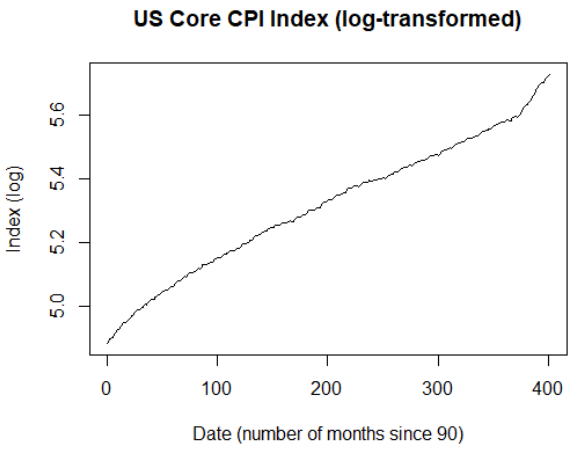

Figure 3: Log-transformed CPI Index

Figure 2. is the absolute change in index level, it may not be a very good view on the data series since in early years the index level is lower, so the percentage change is actually larger given similar absolute change. From Figure 3., which is the log-transformed Index, we can observe higher slopes in early years, while the much steeper slope after 2020 is still very visible.

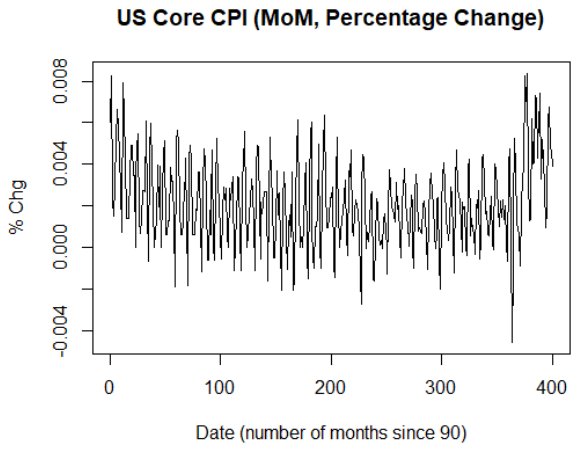

Figure 4: Percentage change in CPI, difference of log-Index, MoM

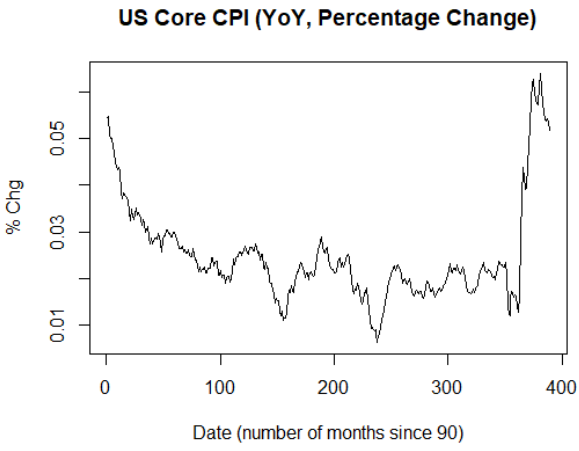

Figure 5: Percentage change in CPI, yearly difference (YoY)

Figure 4 is the percentage change in index level, created by differencing the log-Index, showing the trend that we observed above, higher percentage change in early years and after 2020, as well as an even higher level of choppiness, that may indicate the influence of seasonal factors. In the yearly percentage change as shown in Figure 5., the series is more stable and didn't exhibit any choppiness like in Figure 4.

Stationary through differencing

Considering the observation above, it is decided to take (monthly) differencing in the log-Index, and then take a yearly (12-monthly) differencing. Figure 6 shows the result of the operations, and we can see a still observable but much lower level of choppiness. The ACF and PACF plots of the monthly differencing, together with the monthly than yearly differencing, are shown below in Figure 7-10.

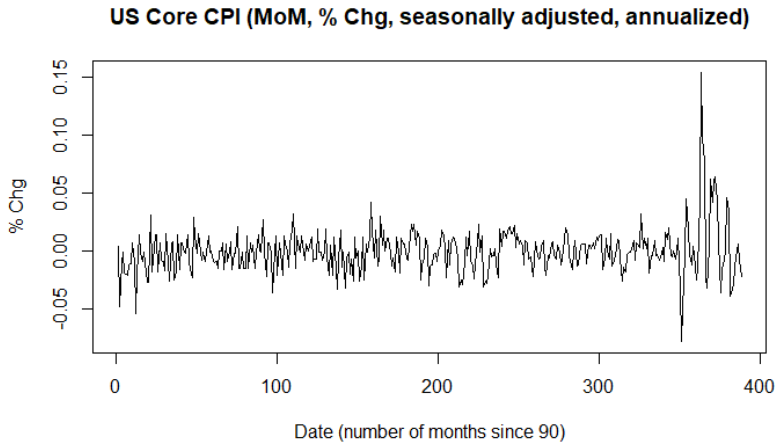

Figure 6: Monthly-then-yearly-differencing on log-Index, annualized (x12)

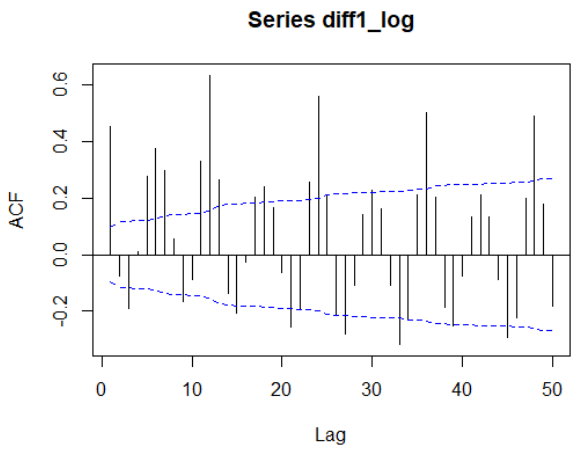

Figure 7: ACF of monthly differencing on log-Index

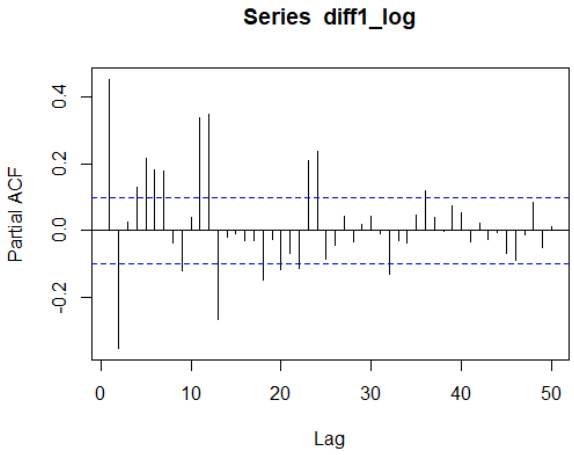

Figure 8: PACF of monthly differencing on log-Index

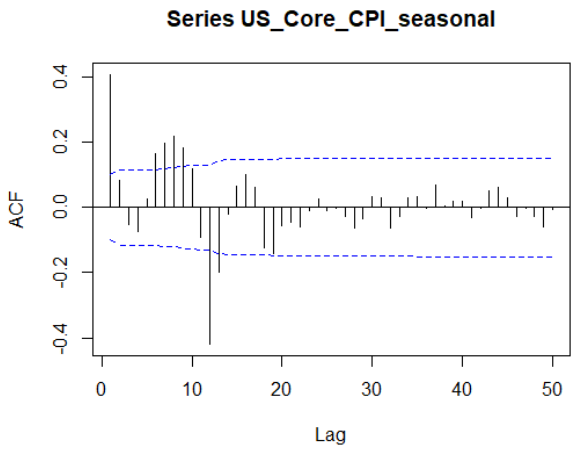

Figure 9: ACF of monthly-then-yearly-differencing

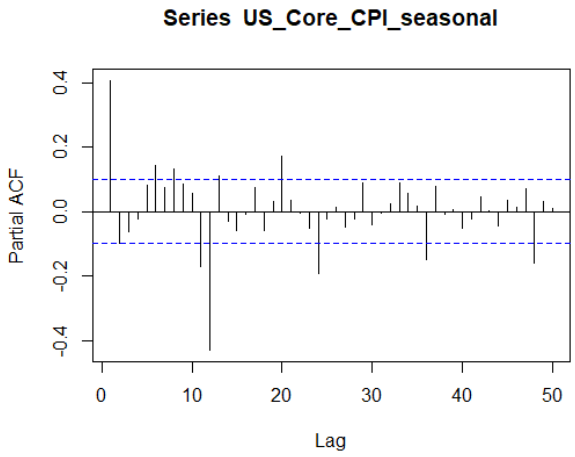

Figure 10: PACF of monthly-then-yearly-differencing

It is known that PACF of seasonal models are difficult to explain except for purely seasonal AR or MA models, therefore we will focus in the ACFs to infer any hints that would be the model. In the ACF of the monthly differencing (Figure 7.), it clearly indicates a seasonal trend that the ACF decays slowly but in a periodic way, ignoring the seasonality, it remains unclear whether the cutoff is at 1 or 3, or even longer lags. For the monthly-then-yearly-differencing log-Index, the ACF clearly cutoff after one seasonal lag. Hence, in a seasonal ARIMA model (p,d,q) x $(P,D,Q)_s$, we can start the test with d=1, D=1, s=12, as basic differencing and a yearly differencing are both needed. we would further assume q = 1 or 3, and Q = 1. p or P are unknown, but it is not likely that both are to be zero, as the ACFs and PACFs did not hint the dataset to be a pure MA model.

In the meantime, Augmented Dickey-Fuller Test is conducted on the monthly-then-yearly-differencing log-Index, the output is -4.6808 and p-value is 0.01, which means null hypothesis is rejected so the result data series should be stationary. This is somewhat contradictory to the “auto.arima” function on the above data series, which suggests an ARIMA (0,1,3) model for it.

Model selection and testing

First Model: ARIMA (0,1,1) x $(0,1,1)_{12}$

```
Call:
arima(x = log_cpi, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 1
2))

Coefficients:
          ma1     sma1
      0.4178  -0.6689
s.e.  0.0400   0.0488

sigma^2 estimated as 1.572e-06:  log likelihood = 2038.19,  aic = -4072.38
```

Figure 11: Result of the first model ARIMA (0,1,1) x $(0,1,1)_{12}$

(Full details of estimated parameters, log-likelihood, AIC, and Ljung-Box test results would in shown in Table 1 & 2 at the end of this section)

Parameter estimation and Ljung-Box test

The estimated parameters of the model are ma1 = 0.4178, while sma1 = -0.6689, with log-likelihood = 2038.19 while AIC = -4072.38. Ljung-Box test is then conducted at lag = 10,20,30,40, with all p-value significantly below 0.0a, we have strong evidence to reject the null hypothesis that the error terms have no correlation. The model is very likely to be insufficient to explain the variations in the timer series.

Over-parameterized method

From ARIMA (0,1,1) x $(0,1,1)_{12}$, d = 1 and D = 1 remains unchanged, while p, q, P, Q take turns to be increased by 1 in order to observe any change to the result. Parameter of ma1 changed significantly as p or q is increased by 1 (ARIMA (1,1,1) x $(0,1,1)_{12}$ or ARIMA (0,1,2) x $(0,1,1)_{12}$), while log-likelihood increased and AIC decreased markedly, so both changes are very likely to be helpful to reduce the correlations in the residuals, as confirmed by p-value of 0.3503 and 0.02556 respectively.

Increase of P and Q produced no significant change in above measurements, so these changes are deemed not useful. We then try to increase p by 1 to 2, despite apparent change in ar1 and ma1 parameters, log-likelihood decreased, and AIC increased, so this change is discarded as well.

The next step is to combine both changes in p and q and test ARIMA (1,1,2) x $(0,1,1)_{12}$ model, this model produces significant change to all AR and MA parameters again, while the sma1 parameter remain very consistent, and large improvements were seen at log-likelihood and AIC at 2060.49 and -4112.97 respectively.

Second Model: ARIMA (1,1,3) x $(0,1,1)_{12}$

```
Call:
arima(x = log_cpi, order = c(1, 1, 3), seasonal = list(order = c(0, 1, 1), period = 1
2))

Coefficients:
         ar1      ma1      ma2      ma3     sma1
      0.9904  -0.5421  -0.2433  -0.0869  -0.7859
s.e.  0.0139   0.0533   0.0585   0.0587   0.0374

sigma^2 estimated as 1.381e-06:  log likelihood = 2061.56,  aic = -4113.11
```

Figure 12: Result of the first model ARIMA (1,1,3) x $(0,1,1)_{12}$

Parameter estimation and Ljung-Box test

In view of the above section the p = 1 is likely to be a helpful change, and the earlier observations that q could be equals to 3 from "auto.arima" function and ACF plot, ARIMA (1,1,3) x $(0,1,1)_{12}$ is selected to be the second model

The estimated parameters of the model are ar1 = 0.9904, ma1 = -0.567, ma2 = -0.243, ma3 = -0.087, while sma1 = -0.786, with log-likelihood = 2061.56 while AIC = -4113.11. Ljung-Box test is then conducted at lag = 10,20,30,40, with all p-value significantly above (at 20 or above p-value are all higher than 0.75), we have

strong evidence **NOT** to reject the null hypothesis that the error terms have no correlation. The model is much better than the first one to explain the variations in the timer series.

Over-parameterized method

Holding d = 1 and D =1 unchanged, increasing p, q, P, Q, like above did not generate any improvement in log-likelihood and AIC despite large changes in parameters estimated. Therefore, based on AIC, the model ARIMA (1,1,3) x $(0,1,1)_{12}$ is selected to be the best model.

Table results

Model	p,d,q	P,D,Q	ar1	ar2	ma1	ma2	ma3	ma4	sar1	sma1	sma2
Model 1	0,1,1	0,1,1			0.418					-0.669	
	1,1,1	0,1,1	0.528		-0.032					-0.729	
	2,1,1	0,1,1	0.268	0.128	0.23					-0.728	
	0,1,2	0,1,1			0.48	0.21				-0.706	
	0,1,1	1,1,1			0.422				0.075	-0.719	
	0,1,1	0,1,2			0.422					-0.647	-0.046
	1,1,2	0,1,1	0.986		-0.567	-0.287				-0.785	
Model 2	**1,1,3**	**0,1,1**	**0.99**		**-0.542**	**-0.243**	**-0.087**			**-0.786**	
	2,1,3	0,1,1	0.293	0.687	0.141	-0.655	-0.25			-0.787	
	1,1,4	0,1,1	-0.94		1.451	0.734	0.334	0.084		-0.717	
	1,1,3	1,1,1	0.075		0.43	0.219	0.072		0.045	-0.745	
	1,1,3	0,1,2	0.056		0.448	0.229	0.076			-0.702	-0.03

Table 1: parameters of all the models tested

Model	p,d,q	P,D,Q	log-likelihood	AIC	p-value (Ljung-Box K=30)	K=10	K=20	K=40
Model 1	0,1,1	0,1,1	2038.19	-4072.38	4.16E-06	2.58E-09	6.22E-08	8.88E-05
	1,1,1	0,1,1	2049.15	-4092.31	0.3503			
	2,1,1	0,1,1	2049.14	-4090.28	0.342			
	0,1,2	0,1,1	2046.82	-4087.65	0.02556			
	0,1,1	1,1,1	2038.59	-4071.19	8.10E-06			
	0,1,1	0,1,2	2038.56	-4071.11	7.45E-06			
	1,1,2	0,1,1	2060.49	-4112.97	0.687			
Model 2	**1,1,3**	**0,1,1**	**2061.56**	**-4113.11**	**0.8363**	**0.3709**	**0.7586**	**0.9659**
	2,1,3	0,1,1	2061.48	-4110.97	0.7693			
	1,1,4	0,1,1	2048.55	-4085.09	0.1842			
	1,1,3	1,1,1	2048.72	-4085.45	0.2204			
	1,1,3	0,1,2	2048.7	-4085.4	0.2138			

Table 2: log-likelihood, AIC and test results of Ljung-Box tests

One interesting point of the selected model / final result is that ar1 = 0.9904 is very close to 1, which almost contradicts Dickey-Fuller test suggesting that the time series is stationary, but could explain why "auto.arima" function suggests that d = 1 after monthly and yearly differencing.

Another interesting point is the parameters of ARIMA (1,1,3) x $(0,1,1)_{12}$ and ARIMA (1,1,2) x $(0,1,1)_{12}$ are very similar, together with almost identical log-likelihood and AIC numbers, changing q from 2 to 3 adds little power to the model to explain the variation of the time series.

Diagnostics and Forecasting

Similar to what we observed in the preview / initial analysis of the dataset, the residuals of the models are confined in a tight range for nearly to whole length of the series, except the start and the end.

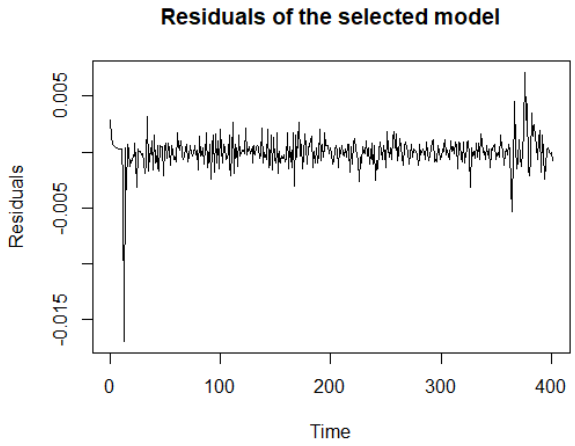

Figure 13: Residuals of ARIMA (1,1,3) x $(0,1,1)_{12}$ model

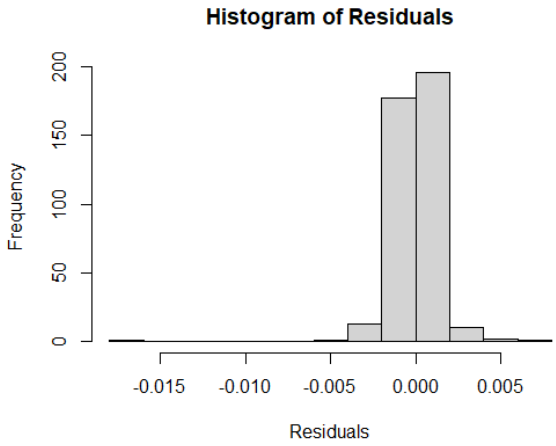

Figure 14: Histogram of ARIMA (1,1,3) x $(0,1,1)_{12}$ model

It would be interesting to see as well, without those two periods, whether the residuals would show normality. In term of the Q-Q plot as well as the Shapiro-Wilk test, null hypothesis of normality is rejected.

The forecasts as shown in Table 3, deviated from the actual values by some extents. The forecasts expected a continuation of trend which can be explained from the ar1 parameter that almost equal to 1.

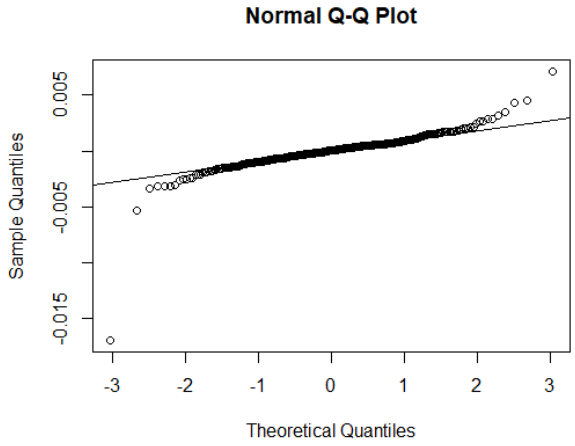

Figure 13: QQ plot of ARIMA (1,1,3) x $(0,1,1)_{12}$ model

Forecast	Actual	Forecast
1	308.910	309.717
2	309.402	310.913
3	310.103	312.256
4	310.817	313.463
5	311.380	314.891

Table 3: 5-period forecasts versus actual values

Conclusion

The inflation data showed it is indeed a difficult number to be forecasted. The inflation after 2020, showed a completely different trend with the time period some time after 1990 to 2020. If the economist reused their old model, that may help to explain why many economists did not foresee the spike in inflation.

One of the takeaways is that the fitted ARIMA, seems to be overfitted to the data and thus became weak in predictability. The continuation of falling inflation in starts of 90s, stable inflation in between, and the continuous rocketing inflation in 2021-2022 seems to encourage the overfitting in some way. Another takeaway is that forecast inflation is complex tasks that requires more than only time series techniques. A mixtures of timer series forecasting and regression of commodity prices and leading economic data, and possibly more advanced techniques maybe required to achieve a better result.

Appendix – R code and output

```
> library(tidyverse)
> library(forecast)
> library(tseries)
> library(tesseract)
> library(TSA)
>
> setwd("C:/Users/USER/Scripts/STAT8003/")
>
> ### Step 1: Load and inspect the data
> cpi_data <- read.csv("CPILFENS_90-23.csv")
> cpi_data_original <- cpi_data
> head(cpi_data)
      DATE CPILFENS
1 1/1/1990    132.0
2 2/1/1990    132.8
3 3/1/1990    133.9
4 4/1/1990    134.2
5 5/1/1990    134.4
6 6/1/1990    134.8
>
> #Remove last 5 data and re-format date
> cpi_data <- cpi_data[1:(nrow(cpi_data) - 5), ]
> cpi_data$DATE <- as.Date(cpi_data$DATE, format = "%m/%d/%Y")
>
> ### Step 2: Draw the time plot
> plot(cpi_data$DATE, cpi_data$CPILFENS, type = "l", xlab = "Date",
+      ylab = "Index", main = "US Core CPI Index from 1990 to 2023")
>
> ### Step 3: Check for stationarity and apply transformations if needed
> #first differencing
> diff1 <- diff(cpi_data$CPILFENS, 1)
> plot(diff1, type = "l", xlab = "Date (number of months since 90)",
+      ylab = "Index", main = "US Core CPI Index (MoM, Absolute Change)")
> #log transforming
> log_cpi <- log(cpi_data$CPILFENS)
> plot(log_cpi, type = "l", xlab = "Date (number of months since 90)",
+      ylab = "Index (log)", main = "US Core CPI Index (log-transformed)")
> #Monthly change
> diff1_log <- diff(log_cpi, 1)
> plot(diff1_log, type = "l", xlab = "Date (number of months since 90)",
+      ylab = "% Chg", main = "US Core CPI (MoM, Percentage Change)")
> #Yearly change
> diff12_log <- diff(log_cpi, 12)
> plot(diff12_log, type = "l", xlab = "Date (number of months since 90)",
+      ylab = "% Chg", main = "US Core CPI (YoY, Percentage Change)")
>
> ### Step 4: Specify the model based on ACF and PACF
> # acf and pacf graphs - monthly % change only
> acf_result_monthly <- acf(diff1_log, lag.max = 50, plot = TRUE, ci.type = "
ma")
> pacf_result_monthly <- pacf(diff1_log, lag.max = 50, plot = TRUE)
>
> # test monthly % change with yearly seasonal trend
> US_Core_CPI_seasonal <- diff(diff1_log, 12)
> plot(12*US_Core_CPI_seasonal, type = "l", xlab = "Date (number of months si
nce 90)",
+      ylab = "% Chg", main = "US Core CPI (MoM, % Chg, seasonally adjusted,
annualized)")
> adf.test(US_Core_CPI_seasonal)

        Augmented Dickey-Fuller Test
```

```
data:  US_Core_CPI_seasonal
Dickey-Fuller = -4.6808, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(US_Core_CPI_seasonal) : p-value smaller than printed p-value
>
> # acf and pacf graphs - monthly % change with seasonal (yearly) effect
> acf_result_seasonal <- acf(US_Core_CPI_seasonal, lag.max = 50, plot = TRUE,
 ci.type = "ma")
> pacf_result_seasonal <- pacf(US_Core_CPI_seasonal, lag.max = 50, plot = TRU
E)
>
> auto.arima(US_Core_CPI_seasonal)
Series: US_Core_CPI_seasonal
ARIMA(0,1,3)

Coefficients:
          ma1      ma2      ma3
      -0.5523  -0.3139  -0.1179
s.e.   0.0503   0.0586   0.0519

sigma^2 = 2.176e-06:  log likelihood = 1973.58
AIC=-3939.17   AICc=-3939.06   BIC=-3923.33
>
> ### Step 5: Specify, estimate, and check the model
> ## Model 1 ARIMA (0,1,1)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(0, 1, 1),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> # (a) Estimate parameters
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
         ma1     sma1
      0.4178  -0.6689
s.e.  0.0400   0.0488

sigma^2 estimated as 1.572e-06:  log likelihood = 2038.19,  aic = -4072.38

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
>
> # (b) Ljung-Box test
> Box.test(cpi_model$residuals, lag = 10, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 60.781, df = 10, p-value = 2.578e-09

> Box.test(cpi_model$residuals, lag = 20, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
```

```
X-squared = 72.835, df = 20, p-value = 6.233e-08

> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 77.739, df = 30, p-value = 4.155e-06

> Box.test(cpi_model$residuals, lag = 40, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 82.49, df = 40, p-value = 8.878e-05

>
> # (c) overparameterized method
> # (1,1,1)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(1, 1, 1),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
          ar1      ma1     sma1
      0.5282  -0.0315  -0.7287
s.e.  0.0937   0.1103   0.0448

sigma^2 estimated as 1.478e-06:  log likelihood = 2049.15,  aic = -4092.31

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 32.374, df = 30, p-value = 0.3503

> # (2,1,1)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(2, 1, 1),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
          ar1     ar2      ma1     sma1
      0.2683  0.1278  0.2304  -0.7281
s.e.     NaN     NaN     NaN   0.0447

sigma^2 estimated as 1.478e-06:  log likelihood = 2049.14,  aic = -4090.28

Training set error measures:
```

```
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning messages:
1: In sqrt(diag(x$var.coef)) : NaNs produced
2: In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 32.558, df = 30, p-value = 0.342

> # (0,1,2)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(0, 1, 2),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(0, 1, 2), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
         ma1     ma2     sma1
      0.4795  0.2100  -0.7059
s.e.  0.0495  0.0499   0.0466

sigma^2 estimated as 1.499e-06:  log likelihood = 2046.82,  aic = -4087.65

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 46.881, df = 30, p-value = 0.02556

> # (0,1,1)x(1,1,1)12
> cpi_model <- arima(log_cpi, order = c(0, 1, 1),
+                    seasonal = list(order = c(1, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(0, 1, 1), seasonal = list(order = c(1, 1, 1), pe
riod = 12))

Coefficients:
         ma1    sar1     sma1
      0.4219  0.0754  -0.7194
s.e.  0.0402  0.0829   0.0685

sigma^2 estimated as 1.568e-06:  log likelihood = 2038.59,  aic = -4071.19

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
```

```
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 75.68, df = 30, p-value = 8.098e-06

> # (0,1,1)x(0,1,2)12
> cpi_model <- arima(log_cpi, order = c(0, 1, 1),
+                    seasonal = list(order = c(0, 1, 2), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 2), pe
riod = 12))

Coefficients:
          ma1     sma1     sma2
      0.4215  -0.6466  -0.0464
s.e.  0.0402   0.0532   0.0540

sigma^2 estimated as 1.568e-06:  log likelihood = 2038.56,  aic = -4071.11

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 75.941, df = 30, p-value = 7.445e-06

>
> # Increase in p,q both improve log-likelihood and AIC significantly,
> # the next step is to try both of them
> # (1,1,2)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(1, 1, 2),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(1, 1, 2), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
          ar1      ma1      ma2     sma1
      0.9862  -0.5665  -0.2865  -0.7847
s.e.  0.0174   0.0506   0.0476   0.0375

sigma^2 estimated as 1.389e-06:  log likelihood = 2060.49,  aic = -4112.97

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test
```

```
data:  cpi_model$residuals
X-squared = 25.767, df = 30, p-value = 0.687

>
> # The model achieves a much better result, but as original non-seasonal ac
f,
> # together with auto.arima, suggest q maybe equals to 3,
> # the next model we are going to try is q = 3
>
> ## Model 2 ARIMA (1,1,3)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(1, 1, 3),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> # (a) Estimate parameters
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(1, 1, 3), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
         ar1      ma1      ma2      ma3     sma1
      0.9904  -0.5421  -0.2433  -0.0869  -0.7859
s.e.  0.0139   0.0533   0.0585   0.0587   0.0374

sigma^2 estimated as 1.381e-06:  log likelihood = 2061.56,  aic = -4113.11

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
>
> # (b) Ljung-Box test
> Box.test(cpi_model$residuals, lag = 10, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 10.83, df = 10, p-value = 0.3709

> Box.test(cpi_model$residuals, lag = 20, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 15.307, df = 20, p-value = 0.7586

> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 22.472, df = 30, p-value = 0.8363

> Box.test(cpi_model$residuals, lag = 40, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 25.316, df = 40, p-value = 0.9659

>
```

```
> # (c) overparameterized method
> # (2,1,3)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(2, 1, 3),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(2, 1, 3), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
         ar1     ar2     ma1     ma2      ma3     sma1
      0.2932  0.6872  0.1408  -0.655  -0.2497  -0.7872
s.e.     NaN     NaN     NaN     NaN      NaN   0.0376

sigma^2 estimated as 1.381e-06:  log likelihood = 2061.48,  aic = -4110.97

Training set error measures:
              ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning messages:
1: In sqrt(diag(x$var.coef)) : NaNs produced
2: In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 24.061, df = 30, p-value = 0.7693

> # (1,1,4)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(1, 1, 4),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(1, 1, 4), seasonal = list(order = c(0, 1, 1), pe
riod = 12))

Coefficients:
        ar1    ma1     ma2     ma3     ma4     sma1
      -0.95  1.451  0.7337  0.3338  0.0835  -0.7170
s.e.    NaN    NaN  0.0857  0.0888  0.0481   0.0449

sigma^2 estimated as 1.484e-06:  log likelihood = 2048.55,  aic = -4085.09

Training set error measures:
              ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning messages:
1: In sqrt(diag(x$var.coef)) : NaNs produced
2: In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 36.76, df = 30, p-value = 0.1842

> # (1,1,3)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(1, 1, 3),
+                    seasonal = list(order = c(1, 1, 1), period = 12))
```

```
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(1, 1, 3), seasonal = list(order = c(1, 1, 1), pe
riod = 12))

Coefficients:
         ar1     ma1     ma2     ma3    sar1     sma1
      0.0752  0.4293  0.2192  0.0715  0.0453  -0.7446
s.e.  3.9495  3.9414  2.0000  0.9163  0.0824   0.0626

sigma^2 estimated as 1.482e-06:  log likelihood = 2048.72,  aic = -4085.45

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 35.63, df = 30, p-value = 0.2204

> # (1,1,3)x(0,1,1)12
> cpi_model <- arima(log_cpi, order = c(1, 1, 3),
+                    seasonal = list(order = c(0, 1, 2), period = 12))
> print(summary(cpi_model))

Call:
arima(x = log_cpi, order = c(1, 1, 3), seasonal = list(order = c(0, 1, 2), pe
riod = 12))

Coefficients:
         ar1     ma1     ma2     ma3     sma1     sma2
      0.0561  0.4479  0.2288  0.0760  -0.7015  -0.0295
s.e.  1.2051  1.2015  0.6094  0.2803   0.0546   0.0533

sigma^2 estimated as 1.482e-06:  log likelihood = 2048.7,  aic = -4085.4

Training set error measures:
                ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
Warning message:
In trainingaccuracy(object, test, d, D) :
  test elements must be within sample
> Box.test(cpi_model$residuals, lag = 30, type = "Ljung-Box")

        Box-Ljung test

data:  cpi_model$residuals
X-squared = 35.826, df = 30, p-value = 0.2138

>
> # We have confirmed Model 2 ARIMA (1,1,3)x(0,1,1)12 is indeed the best mode
l
>
> # Step 6: Analysis of the best model
> best_cpi_model <- arima(log_cpi, order = c(1, 1, 3),
+                    seasonal = list(order = c(0, 1, 1), period = 12))
> residuals <- residuals(best_cpi_model)
>
```

```
> # Residual Plot
> plot(residuals, main = "Residuals of the selected model", ylab = "Residuals
")
>
> # Residual Histogram
> hist(residuals, main = "Histogram of Residuals", xlab = "Residuals")
>
> # QQ Plot
> qqnorm(residuals)
> qqline(residuals)
>
> # Shapiro-Wilk test for normality
> shapiro.test(residuals)

        Shapiro-Wilk normality test

data:  residuals
W = 0.75721, p-value < 2.2e-16

> # p-value of less than 2.2e-16, so normality is rejected
>
> # Step 7: Forecast the future values
> forecast_values <- forecast(log_cpi, h = 5, model = best_cpi_model)
> print(forecast_values)
    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
402       5.735660 5.734154 5.737165 5.733356 5.737963
403       5.739512 5.736861 5.742162 5.735458 5.743565
404       5.743822 5.740190 5.747454 5.738268 5.749377
405       5.747680 5.743183 5.752177 5.740803 5.754558
406       5.752227 5.746920 5.757535 5.744110 5.760345
> cpi_forecast <- exp(forecast_values$mean)
>
> # Compare with true values
> true_values <- cpi_data_original$CPILFENS[(nrow(cpi_data_original) - 4):nro
w(cpi_data_original)]
> comparison <- data.frame(Actual = true_values, Forecast = cpi_forecast)
> print(comparison)
   Actual Forecast
1 308.910 309.7172
2 309.402 310.9125
3 310.103 312.2557
4 310.817 313.4627
5 311.380 314.8913
```