Consumer Price Index and Inflation

A Time Series Analysis

Ben Zhang

2021/12/19

Contents

Introduction

I have just finished STAT 443 - Forecasting at the University of Waterloo. I was motivated to complete a full time series analysis using what I have learned in this course during the winter break. I learned forecasting methods with regression, Holt-Winters, and SARIMA models in this course. I will apply these powerful tools to predict a time series.

The Dataset - Consumer Price Index

The dataset was accessed from the Federal Reserve Bank of St.Louis (FRED). It is the monthly consumer price index for all urban consumers (all items less food and energy in U.S. city average) from January 1st, 1957 to Nov 1st, 2021 (n=779).

I was interested in the trend of inflation during the COVID pandemic for a while now. Ever since this spring, I noticed that almost all restaurants at the UW Plaza I frequent updated their prices for the new semster. My go-to orders became 2 to 3 dollars more expensive on average. The same went for my groceries. Every item subtly became a few percent more expensive. This was the first time in my life I really felt the impact of inflation at once around me. Before the pandemic, I would only find out about inflation everytime McDonalds make their McDouble few dimes more expensive.

Inflation is calcuated from CPI as :

$$\frac{CPI_{i+1} - CPI_i}{CPI_i} \times 100\%$$

With this in mind, I will formally analyze the inflation trend in the USA and make short-term predictions. I will be using the ggplots2 package for the first time to produce my plots. I will also use dplyr to manipulate my datasets.

Exploratory Analysis

Taking a peek at the data:

```
cpi <- read.csv("C:/Users/bben555/Desktop/CPILFENS.csv", stringsAsFactors = FALSE)
cpi = tibble(cpi)
cpilfens = cpi$CPILFENS

head(cpi)
```

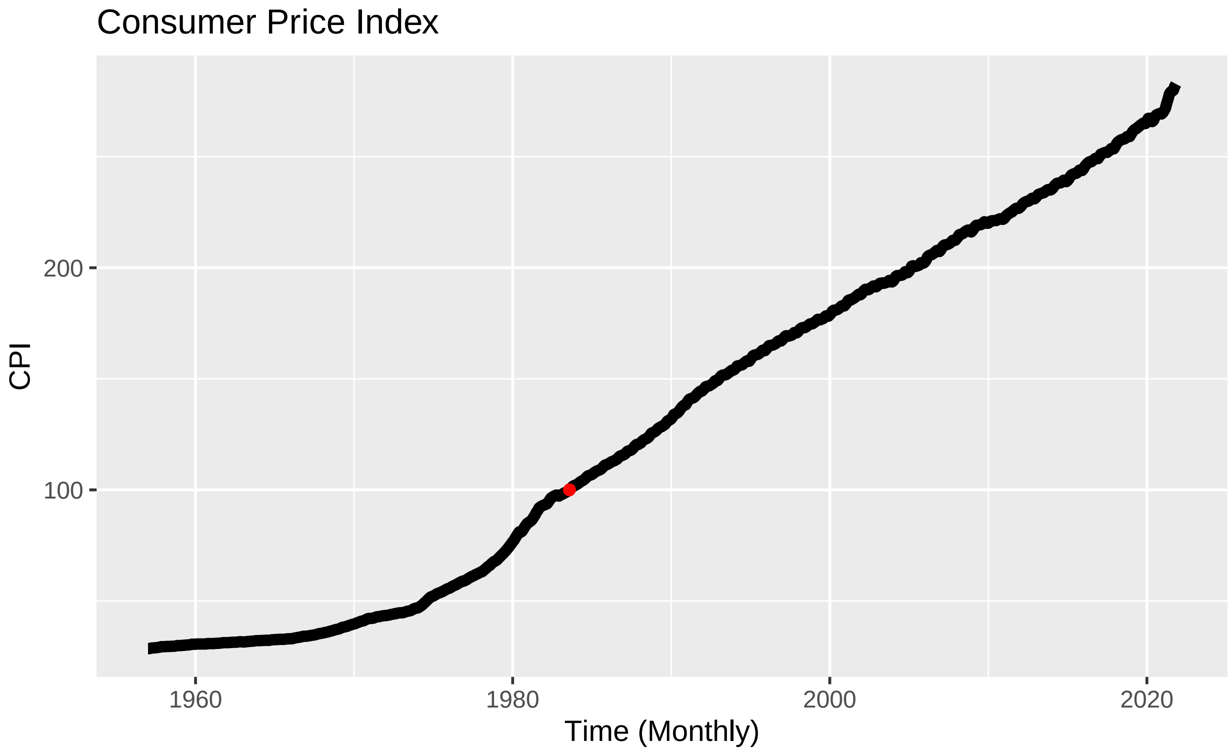
```
## # A tibble: 6 x 2
##   DATE       CPILFENS
##   <chr>         <dbl>
## 1 1957-01-01     28.5
## 2 1957-02-01     28.5
## 3 1957-03-01     28.7
## 4 1957-04-01     28.8
## 5 1957-05-01     28.8
## 6 1957-06-01     28.9
```

```
summary(cpi$CPILFENS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    28.5    45.0   128.5   131.4   201.2   282.8
```

A time-series plot of the CPI:

```
# plot the data

base.date = cpi %>%
    filter(CPILFENS == 100) %>%
    select(DATE)

cpi.gg = ggplot(cpi, aes(x = as.Date(DATE), y = CPILFENS)) +
    geom_line(lwd = 2) + labs(title = "Consumer Price Index",
    x = "Time (Monthly)", y = "CPI") + geom_point(aes(as.Date(base.date$DATE),
    100), col = "red", cex = 1.5)

plot(cpi.gg)
```



The red point, which indicates January 1983, is the base year where CPI = 100.

From the CPI plot we observe an ever increasing trend. Hence we can confirm that inflation has also steadly increased overtime. Three unique trends can be identified from this plot:

- The slope of CPI is the flattest between 1960 to 1970, indicating low inflation.
- The slope was drastically steeper during the 1970s.
- CPI in recent times is increasing the fastest in almost two decades.

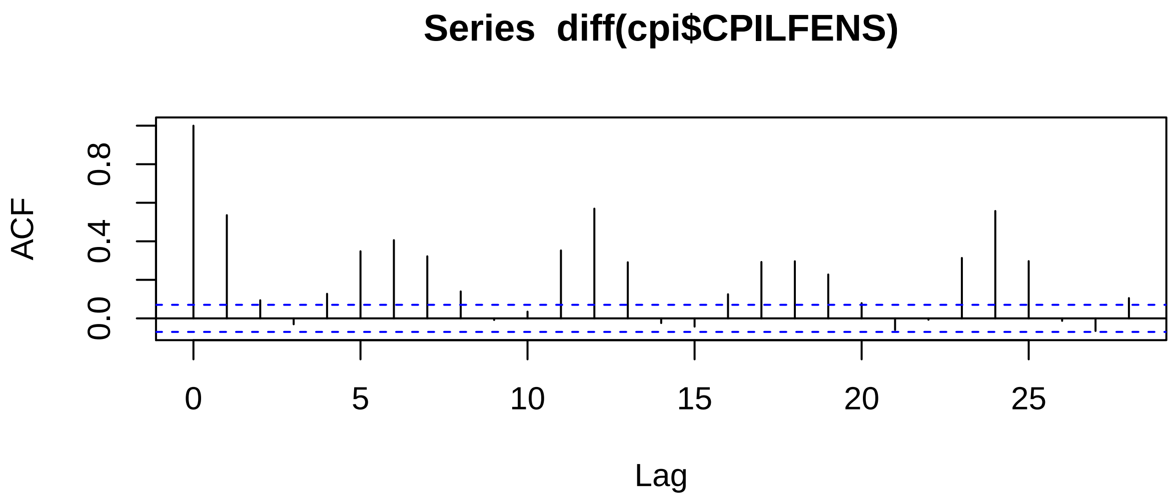
These trends are explained by the historical events post world war, artifically low interest rates in the 70s, and recovering from the pandemic that we are still living in.

Regression Models

First we will model the data with polynomial regression models. We will split the data into a training set (first 50 years) and a test set (last 13 years) to calculate the prediction power of each model.

We can also choose to include a seasonal effect if periodicity is present in the data. Doing so will further improve our prediction power.

```
# seasonal effect

acf(diff(cpi$CPILFENS))
```



By examining the once-differenced CPI data, it is clear that there is a seasonal trend with periodicity of 6 (semiannual). We will now try fitting models with degree 1 to 5 of time plus monthly categorical seasonal effects.

Best Fitting Regression Model

```
train = cpi %>%
    filter(DATE < "2007-01-01")
test = cpi %>%
    filter(DATE >= "2007-01-01")
nrow(train)
```

```
## [1] 600
```

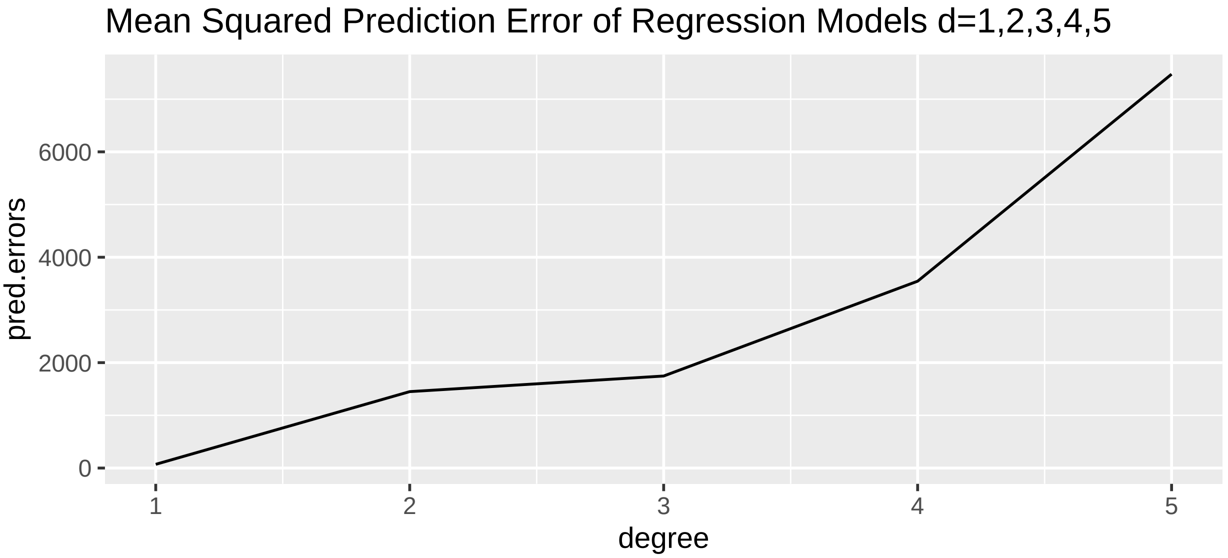
```
nrow(test)
```

```
## [1] 179
```

```
train.ts = ts(train$CPILFENS, start = c(1957, 1), frequency = 12)
test.ts = ts(test$CPILFENS, start = c(2007, 1), frequency = 12)

train.seasonal = as.factor(cycle(train.ts))
test.seasonal = as.factor(cycle(test.ts))

train.time = as.vector(time(train.ts))
test.time = as.vector(time(test.ts))

pred.errors = c()

for (i in 1:5) {

    reg = lm(cpi[1:600, ]$CPILFENS ~ poly(train.time, i) + train.seasonal)
    pred = predict(reg, newdata = data.frame(train.time = test.time,
        train.seasonal = test.seasonal))

    pred.errors[i] = mean((cpi[601:779, ]$CPILFENS - pred)^2)
}

pred.errors
```

```
## [1]   70.45069 1450.20335 1746.06933 3545.37022 7473.67123
```

```
pred.reg.gg = ggplot(data.frame(degree = c(1, 2, 3, 4, 5), pred.errors),
    aes(degree, pred.errors)) + geom_line()
pred.reg.gg + labs(title = "Mean Squared Prediction Error of Regression Models d=1,2,3,4,5")
```

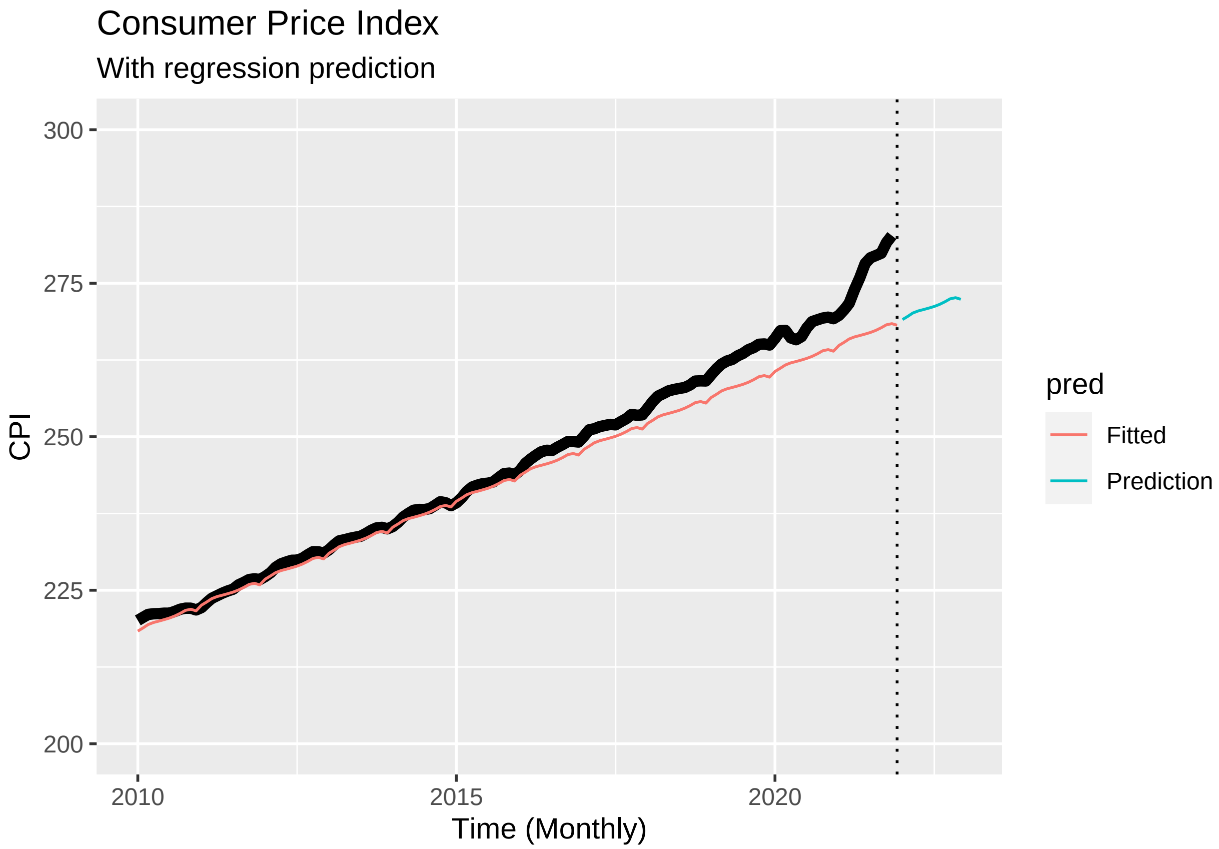


The best predicting regression model is a multiple linear regression model. The response variable is CPI and the two explanatory variables are time (i.e. 1980.50 for July 1st, 1980) and categorical variable month (Jan, Feb, . . .).

Prediction

```
cpi.ts = ts(cpi$CPILFENS, start = c(1957, 1), frequency = 12)
pred.ts = ts(1:13, start = c(2021, 12), frequency = 12)

fulltime = as.vector(time(cpi.ts))
seas = as.factor(cycle(cpi.ts))

bestregmodel = lm(cpi$CPILFENS ~ fulltime + seas)

regline = bestregmodel$fitted.values
reg.pred = predict(bestregmodel, newdata = data.frame(fulltime = as.vector(time(pred.ts)),
    seas = as.factor(cycle(pred.ts))), interval = "prediction")
```

```
reg.fitted = tibble(date = seq(as.Date("1957-01-01"), as.Date("2022-12-01"),
    by = "month"), reg.pred = c(regline, reg.pred[, 1]), pred = as.factor(c(rep("Fitted",
    length(regline) + 1), rep("Prediction", 12))))

cpi.gg + geom_line(data = reg.fitted, aes(x = date, y = reg.pred,
    col = pred)) + geom_vline(aes(xintercept = as.Date("2021-12-01")),
    lty = "dotted") + labs(subtitle = "With regression prediction") +
    scale_x_date(limits = c(as.Date("2010-01-01"), as.Date("2022-12-01"))) +
    ylim(c(200, 300))
```



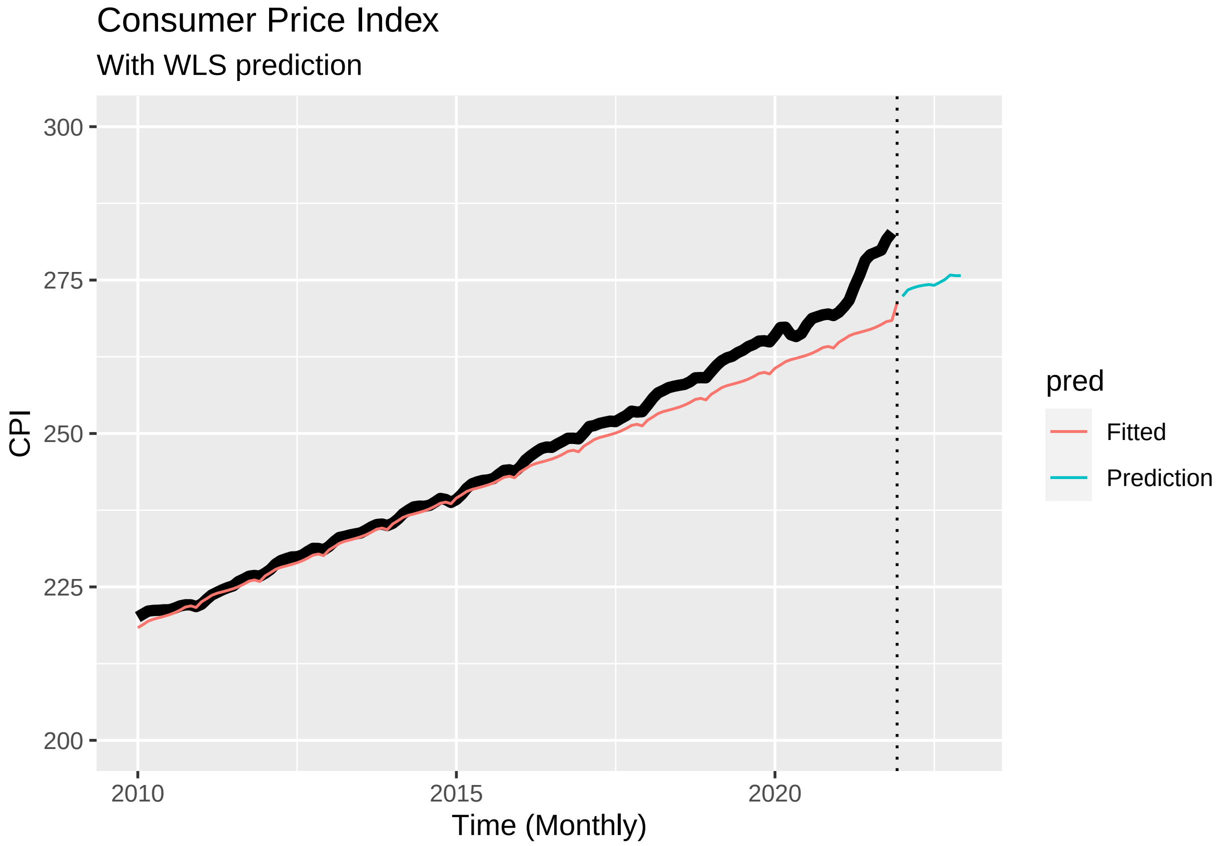
Here we plot the fitted regression line with 13 months of prediction. We see that regression systematically under-fitted the CPI during the pandemic.

Weighted Least Squares

Weighted least squares (WLS) can be used when there are outliers in the data. During the parameter estimation process, outliers will be given less weight to adjust for their influence on the estimates. We will fit a WLS model to see if it will provide a better fitted line.

```
reg.res = bestregmodel$residuals

weights = 1/lm(abs(reg.res) ~ bestregmodel$fitted.values)$fitted.values^2

wls = lm(cpi$CPILFENS ~ fulltime + seas, weights = weights)
```

```
wlsline = bestregmodel$fitted.values
wls.pred = predict(wls, newdata = data.frame(fulltime = as.vector(time(pred.ts)),
    seas = as.factor(cycle(pred.ts))), interval = "prediction")

wls.fitted = tibble(date = seq(as.Date("1957-01-01"), as.Date("2022-12-01"),
    by = "month"), wls.pred = c(wlsline, wls.pred[, 1]), pred = as.factor(c(rep("Fitted",
    length(wlsline) + 1), rep("Prediction", 12))))

cpi.gg + geom_line(data = wls.fitted, aes(x = date, y = wls.pred,
    col = pred)) + geom_vline(aes(xintercept = as.Date("2021-12-01")),
    lty = "dotted") + labs(subtitle = "With WLS prediction") +
    scale_x_date(limits = c(as.Date("2010-01-01"), as.Date("2022-12-01"))) +
    ylim(c(200, 300))
```



WLS produced a better fitted line and perhaps a more accurate prediction than MLS. However it still underfitted the CPI during the pandemic. Therefore regression is not a good method for this dataset as it cannot account for the change in the rate CPI increases in the underlying process. We could however, use only the data points since the pandemic has started to obtain a better fitting model. However, We will check Holt-Winters and Box-Jenkins models next.

Holt-Winters Model

Holt-Winters model can be used for exponential smoothing, double exponential smoothing, additive and multiplicative seasonal effects. Since seasonality exists in our data, an addictive seasonal effect or a multiplicative seasonal effect should be appropriate. We will compare the prediction power of a double exponential smoothing, additive and multiplicative seasonality models.

Best Fitting Model

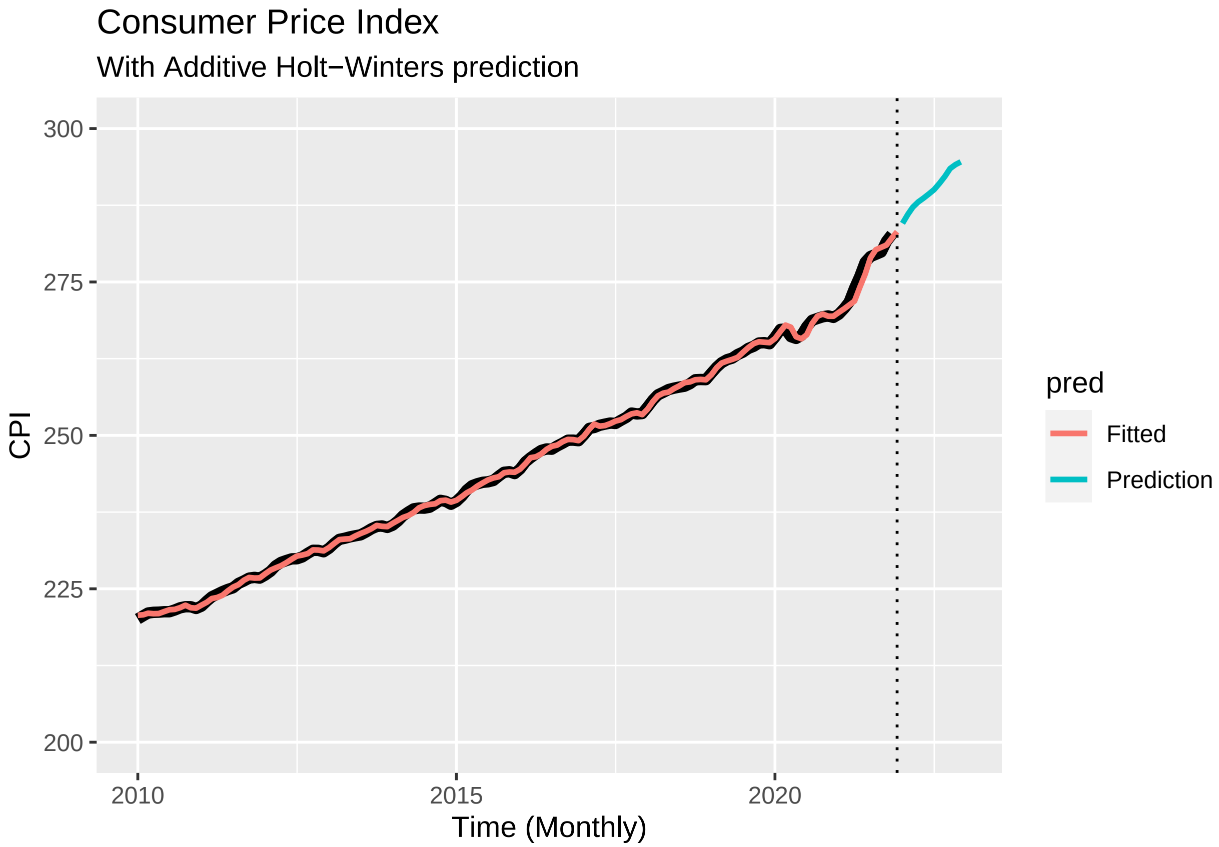
```
# double exponential smoothing, holtwinters models

add.hw = HoltWinters(train.ts)
mult.hw = HoltWinters(train.ts, seasonal = "multiplicative")
des.hw = HoltWinters(train.ts, gamma = FALSE)
data.frame(HWadd = mean((predict(add.hw, n.ahead = 179) - test.ts)^2),
    HWmult = mean((predict(mult.hw, n.ahead = 179) - test.ts)^2),
    DoubleExpSm = mean((predict(des.hw, n.ahead = 179) - test.ts)^2),
    row.names = "MSE")
```

```
##        HWadd   HWmult DoubleExpSm
## MSE 6.075782 10.80689    22.80579
```

The Holt-Winters Forecast with additive seasonality had the lowest mean squared prediction error. We will plot the fitted line with 13 months of predictions on top of the time series plot below.

Prediction

```
add.hw = HoltWinters(cpi.ts)

hw.fitted = tibble(date = seq(as.Date("1957-01-01"), as.Date("2022-12-01"),
    by = "month"), hw.pred = c(rep(NA, 12), add.hw$fitted[, 1],
    predict(add.hw, n.ahead = 13)), pred = reg.fitted$pred)

cpi.gg + geom_line(data = hw.fitted, aes(x = date, y = hw.pred,
    col = pred), lwd = 1) + geom_vline(aes(xintercept = as.Date("2021-12-01")),
    lty = "dotted") + labs(subtitle = "With Additive Holt-Winters prediction") +
    scale_x_date(limits = c(as.Date("2010-01-01"), as.Date("2022-12-01"))) +
    ylim(c(200, 300))
```



This is a much better fitting line than regression. HW model can adjust to the change in the rate CPI increases in the underlying process. We get the CPI prediction for the next 13 months.

```
predict(add.hw, n.ahead = 13)
```

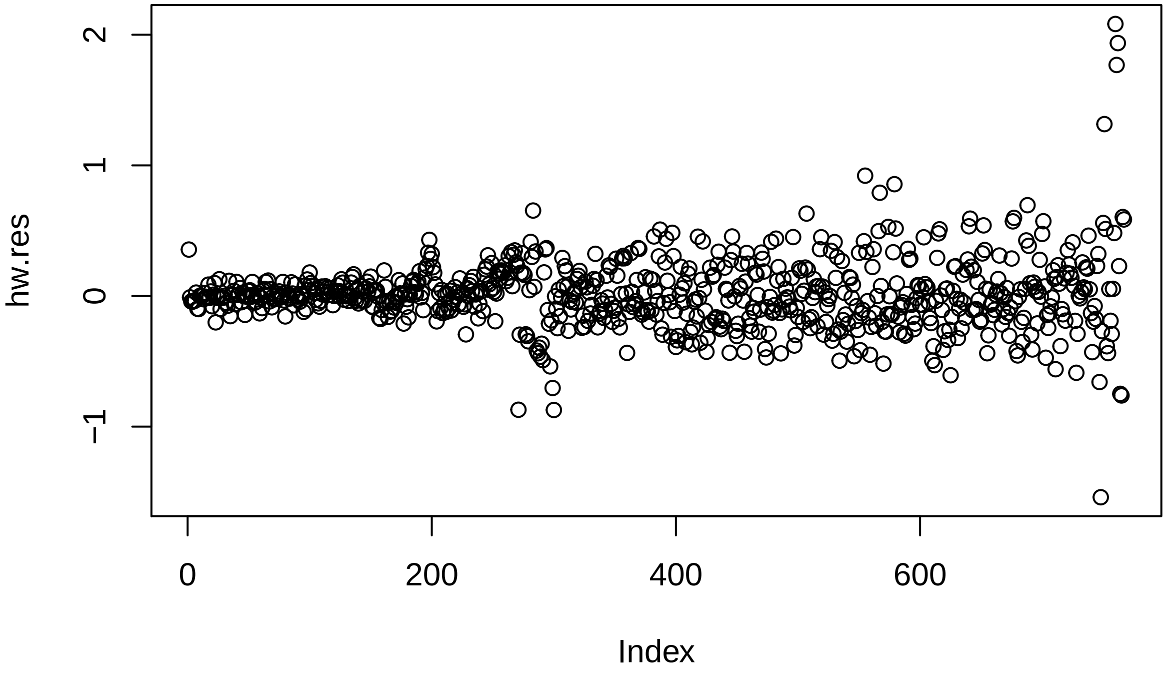
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2021
## 2022 284.5561 286.0369 287.1892 288.0378 288.6522 289.3422 290.0476 291.0784
##           Sep      Oct      Nov      Dec
## 2021                            283.1915
## 2022 292.2079 293.4887 294.1335 294.5710
```

Residuals Analysis

```
hw.fv = as.vector(add.hw$fitted[, 1])
hw.res = cpi[13:779, ]$CPILFENS - hw.fv

summary(hw.res)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.540429 -0.117552  0.002355  0.012298  0.125556  2.082269
```

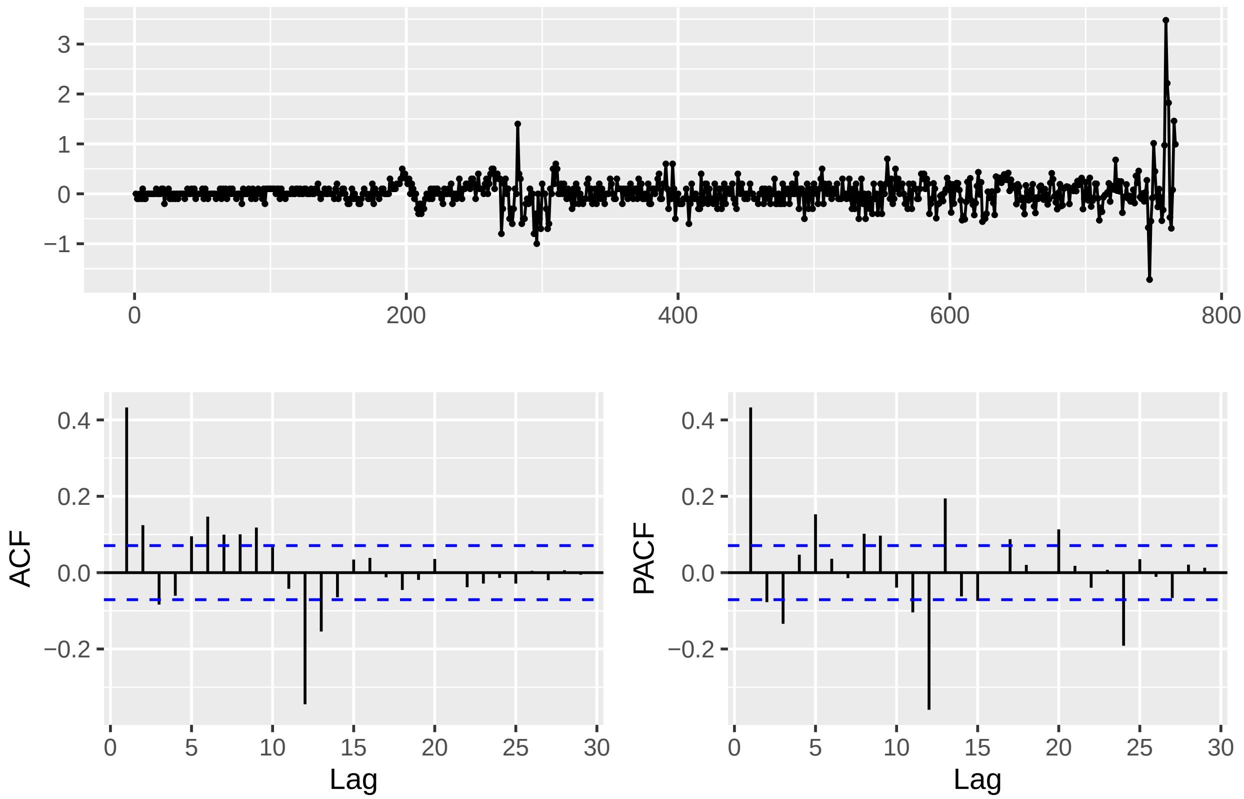
```
plot(hw.res)
```



The residual distributes differently throughout time. Residuals are not normal nor stationary. However they appears to be white noise for certain blocks of time. We cannot produce prediction intervals based on the normal distribution. We could again fit a Holt-Winters model with only recent data to obtain more uniform residuals. We could also attempt to stationarize the residuals and fit ARIMA models as well. In the next section, we will use the Box-Jenkins methodology to get predictions.

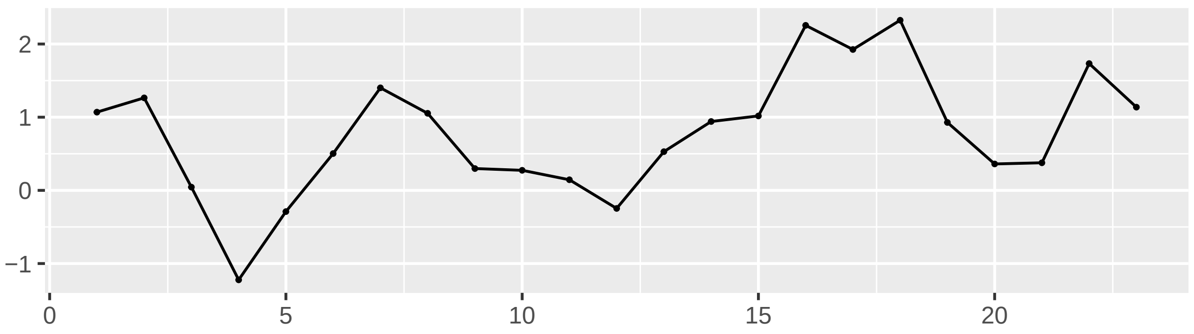
Box-Jenkins Model (SARIMA)

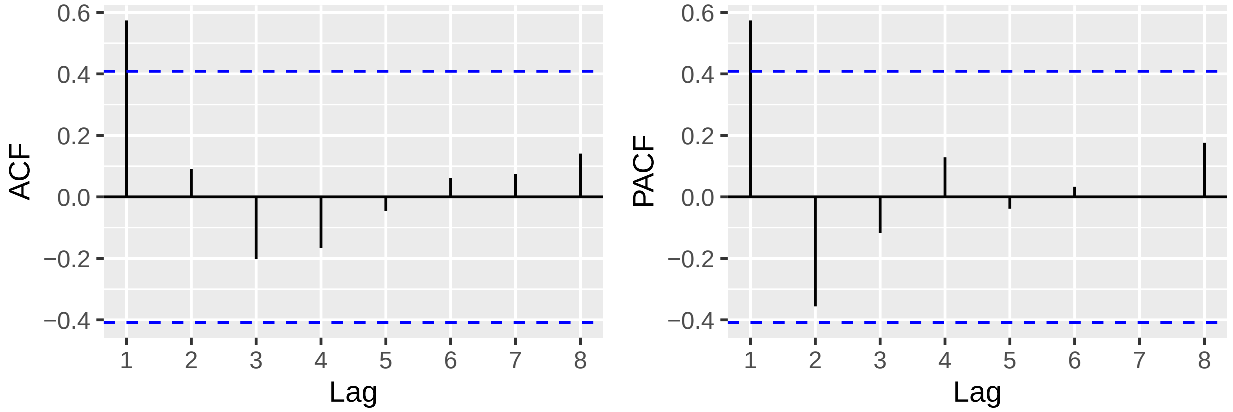
```
cpilfens %>%
    diff(lag = 12) %>%
    diff() %>%
    ggtsdisplay()
```



First we differenced the entire data until stationarity. $Y = (1 - B)(1 - B^1 2)X_t$ appears to be stationary. $(d = 1, D = 1, S = 12)$. However the variance is clearly different after the 700th observation. In this section we will try abandoning observations pre-pandemic and only use the most recent observations to fit a model that appropriately reflects the current COVID situation.

```
cpilfens[-(1:(779 - 24))] %>%
    diff() %>%
    ggtsdisplay()
```





Once differenced data is appropriate to make the sliced time series stationary. $Y_t = (1 - B)X_t$ is stationary. We will initially propose ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(2,1,2) based on the trends in the ACF and PACF plots.

Best Fitting Model

We will extract the corrected AIC from SARIMA models to compare the 5 proposed models.

```
data.frame(‘ARIMA(1,1,1)‘ = sarima(cpilfens[-(1:(779 - 24))],
    q = 1, d = 1, p = 1, details = FALSE)$AICc, ‘ARIMA(1,1,2)‘ = sarima(cpilfens[-(1:(779 -
    24))], q = 1, d = 1, p = 2, details = FALSE)$AICc, ‘ARIMA(2,1,2)‘ = sarima(cpilfens[-(1:(779 -
    24))], q = 2, d = 1, p = 2, details = FALSE)$AICc, row.names = "AICc")
```

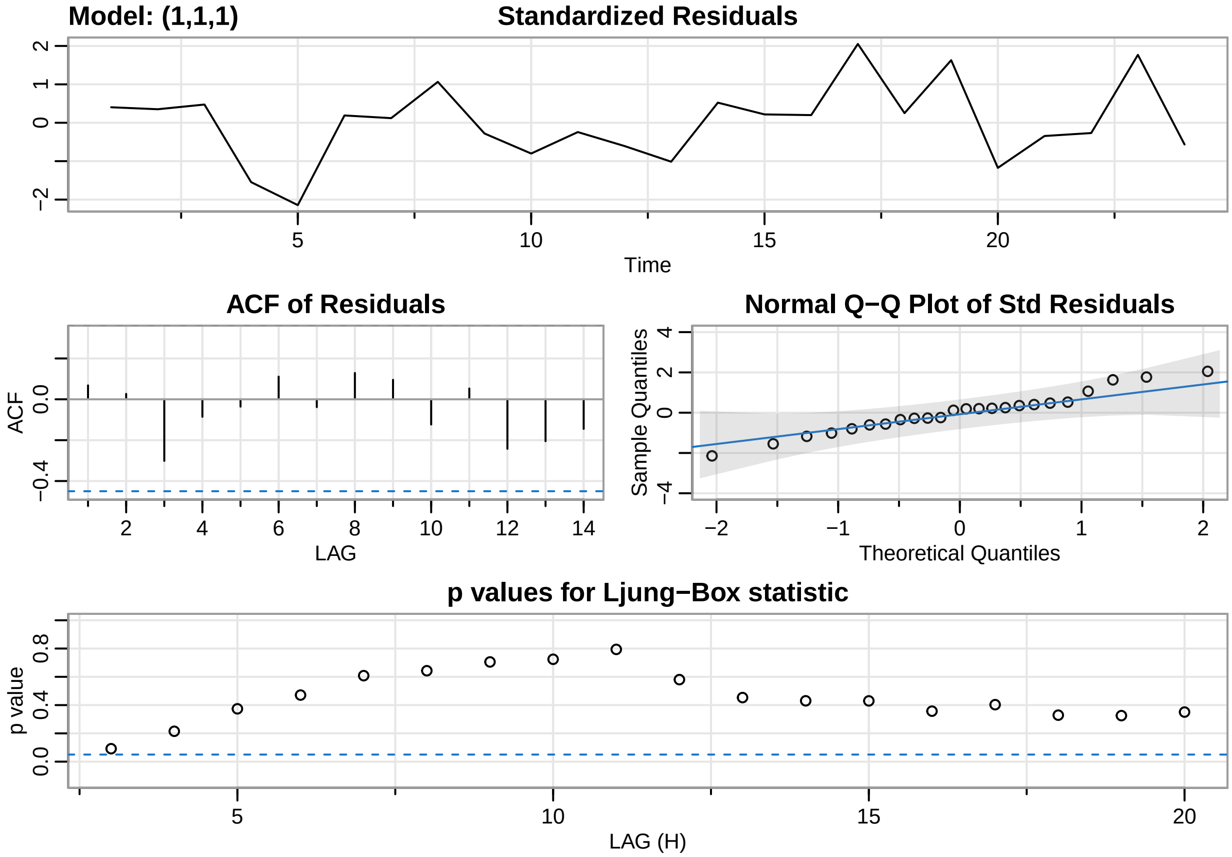
```
##      ARIMA.1.1.1. ARIMA.1.1.2. ARIMA.2.1.2.
## AICc     2.415687     2.480138     2.450197
```

Although the difference in AICc appears to be minimal, ARIMA(1,1,1) is the best fitting model.

Residuals Analysis & Prediction

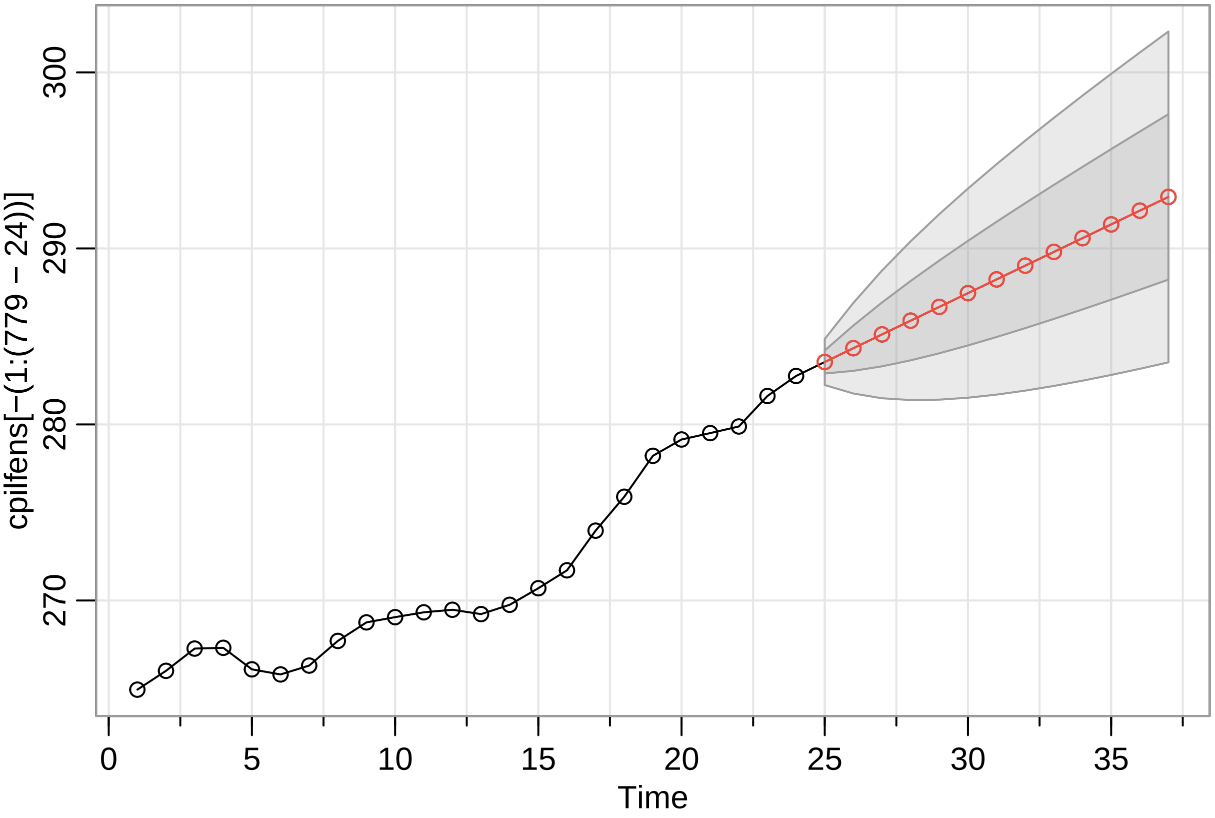
```
sarima(cpilfens[-(1:(779 - 24))], q = 1, d = 1, p = 1)$AICc
```

```
## initial  value -0.164673
## iter   2 value -0.291155
## iter   3 value -0.395255
## iter   4 value -0.399675
## iter   5 value -0.400112
## iter   6 value -0.400279
## iter   7 value -0.400292
## iter   8 value -0.400292
## iter   8 value -0.400292
## final  value -0.400292
## converged
## initial  value -0.411969
## iter   2 value -0.412152
## iter   3 value -0.412407
## iter   4 value -0.412453
## iter   5 value -0.412467
## iter   6 value -0.412468
## iter   7 value -0.412468
## iter   7 value -0.412468
## iter   7 value -0.412468
## final  value -0.412468
## converged
```



```
## [1] 2.415687
```

```
bj.pred = sarima.for(cpilfens[-(1:(779 - 24))], n.ahead = 13,
    q = 1, d = 1, p = 1)$pred
```



```
bj.pred
```

```
## Time Series:
## Start = 25
## End = 37
## Frequency = 1
##  [1] 283.5472 284.3327 285.1153 285.8969 286.6780 287.4591 288.2400 289.0210
##  [9] 289.8019 290.5829 291.3638 292.1448 292.9257
```

The residuals from the ARIMA(1,1,1) model is i.i.d normal. The light-grey 95% prediction interval is valid and we have ploted the predictions of monthly CPI for the next 13 months.

Conclusion and Discussion

The most appropriate model from this analysis is ARIMA(1,1,1). The expected inflation rates for the next 12 months are:

```
infrate = sapply(1:12, FUN = function(x) {
    (bj.pred[x + 1] - bj.pred[x])/bj.pred[x]
})
ts(infrate, start = c(2021, 12), frequency = 12)
```

```
##              Jan         Feb         Mar         Apr         May         Jun
## 2021
## 2022 0.002752498 0.002741237 0.002732375 0.002724424 0.002716835 0.002709405
##              Jul         Aug         Sep         Oct         Nov         Dec
## 2021                                                             0.002770182
## 2022 0.002702058 0.002694768 0.002687522 0.002680317 0.002673152
```

Comparing to the inflation rate of the past 12 months, 4.93%, the annual expected inflation rate next year is lower at 3.31%. 95% CI [0.51%, 6.07%]

This is higher than the average annual inflation in the past 10 years of 1.86%. Recovering from the pandemic has certainly boosted inflation and the prediction implies that the high inflation will continue albeit lower than the past 12 months. We can continue to expect noticable price adjustments in everyday purchases next year.

Reflecting on the analysis, the biggest challenge was the process/time series appeared to be systematically different throughout each time period. CPI followed a different distribution depending on the current political climate. Hence some of the early data was not necessary and will not aid model fitting because there is a violation of uniformity of nature. This is perhaps why the final SARIMA model with data points during the pandemic had the best fit and normal residuals. Therefore for our predictions to be accurate, we would have to assume that the pandemic situation will remain similar next year. However, that assumption might not be very realistic. Omicron variant is growing rampant in North America as of now and it is causing the nth wave. No one knows just how Omicron will impact the economy next year right now.