MITx 6.419x Data Analysis
Statistical Modeling and Computation in Applications

Analysis 4
Time Series Analysis

Jonathan Chang (JonathanChang6d41)

May 16, 2021

Contents

Problem 2
Mauna Loa CO_2 concentration

Part (1)

Given that months from January of 1985 is $i = 0, 1, 2, \ldots$, we assume equidistant periods that fit neatly into 12 per year (adjusted for leap years), such that each period is indexed sometime in the middle of their respective months. That time in years is denoted $t = \frac{0}{12}, \frac{1}{12}, \frac{2}{12}, \ldots, \frac{i}{12}, \ldots$. We define the CO_2 concentration (in ppm) per each month i as C_i, which is composed of a regressed quadratic trend line, estimated seasonal variations and residuals, $C_i = F_2(t_i) + P_i + R_i$.

Precisely, a trend line is fitted over C_i as $F_2(t_i) = \alpha_0 + \alpha_1 t + \alpha_2 t^2$. Then the differences $C_i - F_2(t_i)$ are sorted into the 12 months across all the years and the mean de-trended concentrations (i.e. $[\overline{C_i - F_2(t_i)}_{\text{Jan}}, \overline{C_i - F_2(t_i)}_{\text{Feb}}, \ldots, \overline{C_i - F_2(t_i)}_{\text{Dec}}]$) are mapped to each month i as P_i. The following graph shows such a seasonal component for any particular year, with a cubic spline fitted through the 12 points.

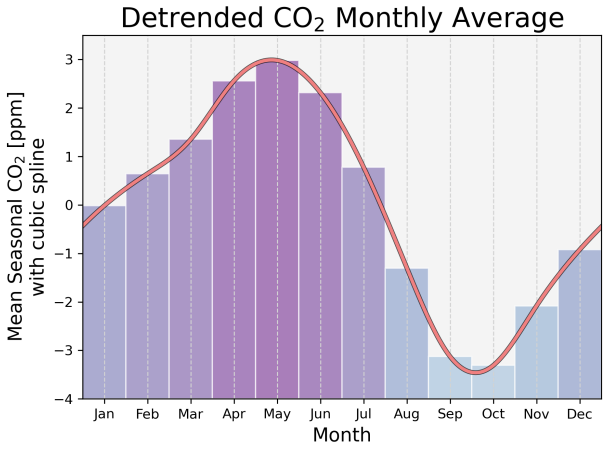

Figure 2.1: Estimated period curve is the de-trended CO_2 concentration averaged by month, and fitted with a cubic spline.

Part (2)

The final model is the trend plus seasonality component that hopefully minimizes any residuals $C_i - F_2(t_i) - P_i = R_i$. The model is trained over a training set consisting of consecutive time series, and predicts a test set of times after it. The split is shown in the plot. Since the residuals are so small, making it hard to see the prediction plotted on top of the data, the difference between C_i and $F_2(t_i) + P_i$ is plotted instead. Blue segments represent when the model predicts higher than the data, whereas red segments represent when it crosses under the data. A plot of the de-trended residuals is included to clarify that difference.

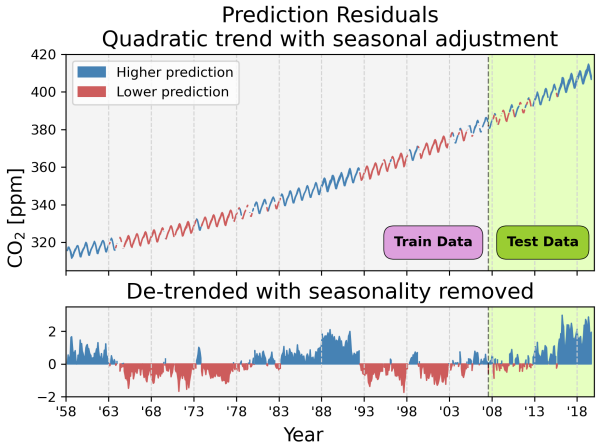

Figure 2.2: Residuals between final fit and CO_2 concentration is plotted over the entire time series.

Notice that the range in CO_2 concentration in the residuals is diminutive compared to the range of the overall fit, proving that the trend and seasonality are meaningful, since their contribution exceed the residual noise. It's also not

surprising that residual magnitude increases toward the end of the test data, showing that the goodness of the trend fit is not indefinite. They diverge after some time.

Part (3)

Test set RMSE and MAPE is greatly improved with seasonality as part of the model, with over 54% lower RMSE and 60% lower MAPE compared to using just the trend.

	Trend $[F_2(t_i)]$	**Final Fit** $[F_2(t_i) + P_i]$
RMSE	2.501	**1.149**
MAPE	0.53%	**0.21%**

Figure 2.3: RMSE and MAPE on the test set. The final fit has less error.

This is sensible, since the range of P_i is $[-3.3, 3.0]$, whereas the range of R_i is $[-1.78, 2.98]$. The smaller residual range is indicative of smaller error, and that the improvement isn't just noise. The final fit more than halves the fit with just the trend, proving that seasonality is a significant component.

Part (4)

The trend F is quadratic and therefore approaches ∞ with time, but within the years in the data set, it's roughly 95.47. In comparison, the range of P_i is about 6.30, and the range of R_i is about 4.72.

	Min	Max	**Range**
$F_2(t_i)$	314.27	409.74	**95.47**
P_i	-3.31	2.99	**6.30**
R_i	-1.74	2.98	**4.72**

Figure 2.4: Range of the trend, period, and residuals.

The ratio of the range of $F_2(t_i)$ to the amplitude of P_i is 15.15. The ratio of the amplitude of P_i to the range of R_i is 1.33. In general, the trend is meaningful if its range is much wider than the amplitude of the seasonal component, so that it doesn't get lost in noise. And the fact that the seasonal amplitude is greater than residuals show a clear periodic effect.

Unfortunately, the trend and period doesn't quite account for all the autocorrelation. We could see that the residuals are clearly still correlated.

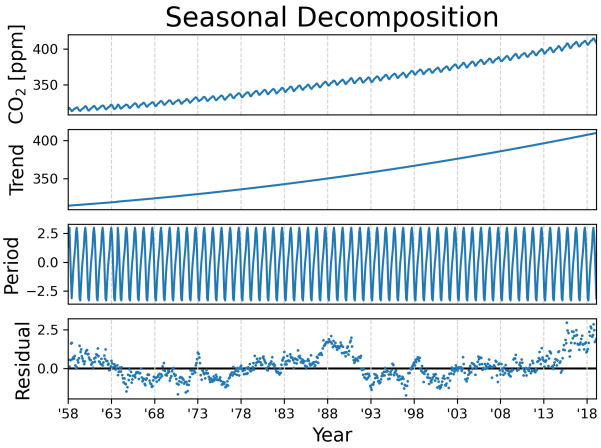

Figure 2.5: Residuals are still correlated.

To be sure, the autocorrelation function (ACF) shows neither an autoregressive nor moving average. From the slow decay, it doesn't seem residuals are stationary.

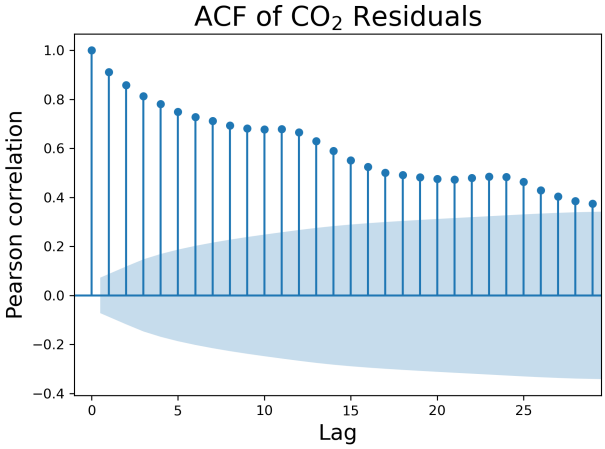

Figure 2.6: Autocorrelations decay to zero very slowly, indicating non-stationary data.

However, it remains true that the range of residuals is less than trend and period.

Problem 3
Autocovariance Functions

Autocovariance is defined as $\mathbf{E}[(X_s - \mu_s)(X_t - \mu_t)]$. We'll also assume $W \sim \mathcal{N}(0, \sigma^2)$.

Part (1)

Moving Average MA(1)

Given the model $X_t = \theta W_{t-1} + W_t$, we want to find the autocovariance function $\gamma_X(h)$. First, we note that $\mu_t = 0$.

$$\begin{aligned}
\mu_t &= \mathbf{E}[X_t] \\
&= \mathbf{E}[\theta W_{t-1} + W_t] \\
&= \mathbf{E}[\theta W_{t-1}] + \mathbf{E}[W_t] && W_{t-1} \text{ and } W_t \text{ are independent} \\
&= \theta \mathbf{E}[W_{t-1}] + \mathbf{E}[W_t] \\
&= \theta \cdot 0 + 0 && \text{since } \mu_t = 0 \\
&= 0
\end{aligned}$$

This allows us to simplify the autocovariance.

$$\begin{aligned}
\gamma_X(h) &= \mathbf{E}[(X_t - \mu_t)(X_{t-h} - \mu_{t-h})] \\
&= \mathbf{E}[X_t X_{t-h}] && \text{since } \mu_t = 0 \\
&= \mathbf{E}[(\theta W_{t-1} + W_t)(\theta W_{t-h-1} + W_{t-h})] && \text{expanding the terms} \\
&= \mathbf{E}[\theta^2 W_{t-1} W_{t-h-1} + \theta W_t W_{t-h-1} \\
&\qquad + \theta W_{t-1} W_{t-h} + W_t W_{t-h}]
\end{aligned}$$

Since for any $s \neq t$, W_s and W_t are independent, and therefore $\mathbf{E}[W_s W_t] = \mathbf{E}[W_s]\mathbf{E}[W_t] = 0$. Since the model is of order 1, $\gamma_X(s-t) = 0$ if $|s-t| > 1$. We'll show this later. Suffice to say, there's a limited range of $h \in [0, 1]$ so that $\gamma_X(h) \neq 0$. Plugging in the result above, we get

$$\begin{aligned}
\gamma_X(0) &= \mathbf{E}[\theta^2 W_{t-1} W_{t-1} + \theta W_t W_{t-1} \\
&\qquad + \theta W_t W_{t-1} + W_t W_t] \\
&= \theta^2 \mathbf{E}[W_{t-1}^2] + \mathbf{E}[W_t^2] \\
&= \theta^2 \sigma^2 + \sigma^2 && \text{since } \mu_t = 0 \\
&= (\theta^2 + 1)\sigma^2 \\
\gamma_X(1) &= \mathbf{E}[\theta^2 W_{t-1} W_{t-2} + \theta W_t W_{t-2} \\
&\qquad + \theta W_{t-1} W_{t-1} + W_t W_{t-1}] \\
&= \theta \mathbf{E}[W_{t-1}^2] \\
&= \theta \sigma^2 && \text{since } \mu_t = 0
\end{aligned}$$

$$\begin{aligned}
\gamma_X(2) &= \mathbf{E}[\theta^2 W_{t-1}W_{t-3} + \theta W_t W_{t-3} \\
&\qquad + \theta W_{t-1}W_{t-2} + W_t W_{t-2}] \\
&= 0 && s \neq t \text{ for every term}
\end{aligned}$$

In fact, we could see that for any $n \geq 2$,

$$\begin{aligned}
\gamma_X(n) &= \mathbf{E}[\theta^2 W_{t-1}W_{t-1-n} + \theta W_t W_{t-(n+1)} \\
&\qquad + \theta W_{t-1}W_{t-1-(n-1)} + W_t W_{t-n}] && \text{note: } n-1 \geq 1 \\
&= 0 && s \neq t \text{ for every term}
\end{aligned}$$

Part (2)

Autoregression AR(1)

Given the model $X_t = \phi X_t + W_{t-1}$, we want to find the autocovariance function $\gamma_X(h)$. Suppose that $|\phi| < 1$. Again, we find that $\mu_t = 0$.

$$\begin{aligned}
\mu_t &= \mathbf{E}[X_t] \\
&= \mathbf{E}[\phi X_{t-1} + W_t] \\
&= \mathbf{E}[\phi X_{t-1}] + \mathbf{E}[W_t] && W_t \text{ and } X_{t-1} \text{ are independent} \\
&= \phi \mathbf{E}[X_{t-1}] \\
&= \phi^j \mathbf{E}[X_{t-j}] && \text{infinite regress}
\end{aligned}$$

But if we assume that $X_0 = 0$, then it solves that problem. The next steps are similar.

$$\begin{aligned}
\gamma_X(h) &= \mathbf{E}[(X_t - \mu_t)(X_{t-h} - \mu_{t-h})] \\
&= \mathbf{E}[X_t X_{t-h}] && \text{since } \mu_t = 0 \\
&= \mathbf{E}[(\phi X_{t-1} + W_t)(\phi X_{t-h-1} + W_{t-h})] && \text{expanding the terms} \\
&= \mathbf{E}[\phi^2 X_{t-1}X_{t-h-1} + \phi W_t X_{t-h-1} \\
&\qquad + \phi X_{t-1}W_{t-h} + W_t W_{t-h}] \\
&= \mathbf{E}[\phi^2 X_{t-1}X_{t-h-1} + \phi X_{t-1}W_{t-h} \\
&\qquad + W_t W_{t-h}] && W_t \text{ cannot depend a past } X_{t-h-1}
\end{aligned}$$

X_t relies on every past term by proportion ϕ, so it could be expressed as a geometric sum. We see this by writing out a few terms.

$$\begin{aligned}
X_0 &= 0 \\
X_1 &= W_1 \\
X_2 &= \phi W_1 + W_2 \\
X_3 &= \phi(\phi W_1 + W_2) + W_3 \\
X_t &= \sum_{j=0}^{t} \phi^{t-j} W_j
\end{aligned}$$

And since W_s and W_t are independent for $s \neq t$, and X_t is a weighted sum of W_i we should find that the autocovariance is a sum of products of corresponding weights.

$$\begin{aligned}
\gamma_X(h) &= \mathbf{E}[(X_t - \mu_t)(X_{t-h} - \mu_{t-h})] \\
&= \mathbf{E}[X_t X_{t-h}] && \text{since } \mu_t = 0 \\
&= \mathbf{E}[(\sum_{i=1}^{t} \phi^{t-i} W_j)(\sum_{j=1}^{t-h} \phi^{t-h-j} W_j)] && \text{expanding the terms} \\
&= \mathbf{E}[(\sum_{i=1}^{t-h} \phi^{t-i} W_j)(\sum_{j=1}^{t-h} \phi^{t-h-j} W_j)] && W_s \perp W_t \text{ if } s > t \\
&= \mathbf{E}[\sum_{j=1}^{t-h} \phi^{t-j} \phi^{t-h-j} W_j^2] && \text{eliminating independent terms} \\
&= \sum_{j=1}^{t-h} \phi^{2t-2j-h} \mathbf{E}[W_j^2] && \text{due to linearity} \\
&= \sum_{j=1}^{t-h} \phi^{2t-2j-h} \sigma^2 && \text{since } \mu_t = 0 \\
&= \sigma^2 \sum_{j=1}^{t-h} \phi^{2t-2j-h}
\end{aligned}$$

Note that if $|\phi| < 1$, then AR(1) is stationary in the steady state ($t \to \infty$). We could simplify the sum using the geometric series.

$$\begin{aligned}
\sigma^2 \sum_{j=1}^{t-h} \phi^{2t-2j-h} &= \sigma^2 \phi^{-h} \left(\sum_{j=0}^{t-h} \phi^{2(t-j)} - \phi^{2t} \right) && \text{start sum at 0} \\
&= \sigma^2 \phi^{-h} \left(\sum_{k=h}^{t} \phi^{2k} - \phi^{2t} \right) && \text{let } k = t - j \\
&= \sigma^2 \phi^{-h} \left(\left[\sum_{k=0}^{t} \phi^{2k} - \sum_{k=0}^{h-1} \phi^{2k} \right] - \phi^{2t} \right) \\
&= \sigma^2 \phi^{-h} \left(\left[\frac{1}{1-\phi^2} - \frac{1-\phi^{2h}}{1-\phi^2} \right] - \phi^{2t} \right) && \text{geometric series}
\end{aligned}$$

Let us take a moment to prove this last step.

$$\begin{aligned}
\sum_{k=0}^{t} \phi^{2k} &= 1 + \phi^2 + \phi^4 + \cdots + \phi^{2t} \\
(1-\phi^2) \sum_{k=0}^{t} \phi^{2k} &= (1-\phi^2)(1 + \phi^2 + \phi^4 + \cdots + \phi^{2t}) \\
&= 1 - \phi^{2(t+1)} && \text{telescoping series}
\end{aligned}$$

$$\sum_{k=0}^{t} \phi^{2k} = \frac{1-\phi^{2(t+1)}}{1-\phi^2} \qquad \text{if } t \text{ is finite}$$
$$= \frac{1}{1-\phi^2} \qquad \text{if } t \to \infty$$

Using this result, we plug it back into the prior equation.

$$\sigma^2 \sum_{j=1}^{t-h} \phi^{2t-2j-h} = \sigma^2 \phi^{-h} \left(\left[\sum_{k=0}^{t} \phi^{2k} - \sum_{k=0}^{h-1} \phi^{2k} \right] - \phi^{2t} \right) \quad \text{note that there is an infinite and finite sum}$$
$$= \sigma^2 \phi^{-h} \left(\left[\frac{1}{1-\phi^2} - \frac{1-\phi^{2h}}{1-\phi^2} \right] - \phi^{2t} \right) \quad \text{the sum to } h-1 \text{ creates the } \phi^{2((h-1)+1)} \text{ term}$$
$$= \sigma^2 \phi^{-h} \left(\frac{\phi^{2h}}{1-\phi^2} - \phi^{2t} \right)$$
$$= \phi^h \frac{\sigma^2}{1-\phi^2} \qquad \phi^t \to 0 \text{ if } |\phi| < 1 \text{ and } t \to \infty$$

Since $\mu = 0$ as $t \to \infty$ and σ^2 does not depend on t, the proof also concludes that AR(1) is stationary for $|\phi| < 1$ and $t \to \infty$.

As a sanity check, we could try the formula we've derived on some low t and h so that it's easy to recognize its correctness. For example, note that the covariance $\text{Cov}(\alpha_1 W_1 + \alpha_2 W_2, \beta_1 W_1) = \alpha_1 \beta_1$, since $W_i \perp W_j$ (is independent to) for every $i \neq j$. In other words, we just multiply the coefficients to the white noise corresponding to the same time.

t	h	Covariance	$\gamma_X(h)$	
			Formula	Result
1	0	$\text{Cov}(W_1, W_1)$	$\sigma^2 \sum_{j=1}^{1} \phi^{2-2j}$	σ^2
2	0	$\text{Cov}(\phi W_1 + W_2, \phi W_1 + W_2)$	$\sigma^2 \sum_{j=1}^{2} \phi^{4-2j}$	$(\phi^2+1)\sigma^2$
2	1	$\text{Cov}(W_1, \phi W_1 + W_2)$	$\sigma^2 \sum_{j=1}^{1} \phi^{4-2j-1}$	$\phi\sigma^2$
3	1	$\text{Cov}(\phi W_1 + W_2, \phi^2 W_1 + \phi W_2 + W_3)$	$\sigma^2 \sum_{j=1}^{2} \phi^{6-2j-1}$	$(\phi^3+\phi)\sigma^2$
3	2	$\text{Cov}(W_1, \phi^2 W_1 + \phi W_2 + W_3)$	$\sigma^2 \sum_{j=1}^{1} \phi^{6-2j-2}$	$\phi^2\sigma^2$

Figure 3.1: Sanity check demonstrating AR(1) formula works.

Problem 5
Converting to Inflation Rates

Part (1)

Conversion to monthly inflation involves aggregating CPI to a monthly time series, and then finding the relative growth to obtain inflation. First, I obtained the mean CPI per each month, and then applied the formula

$$\text{Inflation}_t = \frac{\text{CPI}_t - \text{CPI}_{t-1}}{\text{CPI}_{t-1}}$$

This could be succinctly achieved in code along with each other metric in our analysis.

```
def inflation(df, column_name):
  curr = df[column_name]
  prev = df[column_name].shift()
  return (curr - prev) / prev

def convert_date(df, column_name):
  df.loc[:, 'year'] = df[column_name].map(
    lambda x: x[:4])
  df.loc[:, 'month'] = df[column_name].map(
    lambda x: x[5:7])
  df.loc[:, 'day'] = df[column_name].map(
    lambda x: x[-2:])
  df.drop(column_name, axis=1, inplace=True)

df = pd.read_csv('data/PriceStats_CPI.csv')
tyield = pd.read_csv('data/T10YIE.csv')

# Separate date into year, month, day
convert_date(df, 'date')
convert_date(tyield, 'DATE')

# Aggregate inflation columns
df.loc[:, 'PSDaily'] = inflation(
  df, 'PriceStats')
df = df.groupby(['year', 'month']).agg({
  'CPI': ['mean'],
  'PriceStats': ['mean', 'last'],
  'PSDaily': ['mean']})
new_columns = [
  'CPI', 'PSMean', 'PSLast', 'PSDaily']
df.columns = new_columns

# Aggregate BER inflation columns
tyield.loc[:, 'T10YIE'] =
  tyield['T10YIE'] / 100
tyield = tyield.groupby(
  ['year', 'month']).agg(
  {'T10YIE': ['mean']})
tyield.columns = ['BER']

# Inner join dataframes
df = pd.concat([df, tyield],
  axis=1, join='inner')

# Convert index back into year and month
# columns
df.reset_index(inplace=True)

# Format inflation columns
for column in new_columns[:-1]:
df.loc[:, column] = inflation(df, column)
df.loc[:, 'PSDaily'] *= 30
df.reset_index(inplace=True)
df.loc[:, 'BER'] =
  (df['BER'] + 1) ** (1 / 12) - 1

# Find training split
N_train = df.index[(df['month'] == '08')
  & (df['year'] == '2013')][0]
```

Listing 1: Processing data frame to get inflations.

As before, the first step to building the model is to find the trend. Macroeconomic theory should inform us that both high inflation and deflation are undesirable, and therefore the government employs various means of monetary and fiscal policy to avoid those outcomes, both of which would be disastrous. It should be rare to expect hyperinflation or deflation to go unchecked, so a trend with non-zero slope would be unreasonable

outside of the short term.

Furthermore, the beginning of our training set coincides with a real estate bubble crash and deflationary credit crunch at the end of 2008, and does not at all reflect the trend at normal times. Excluding this period, H_0 could not be rejected that there is no visible trend. The existence of any such trend would be lost in the noise. However, the model should include a constant term, a y-intercept, which is the expected mean inflation after the crash, $\mu = 0.001923$.

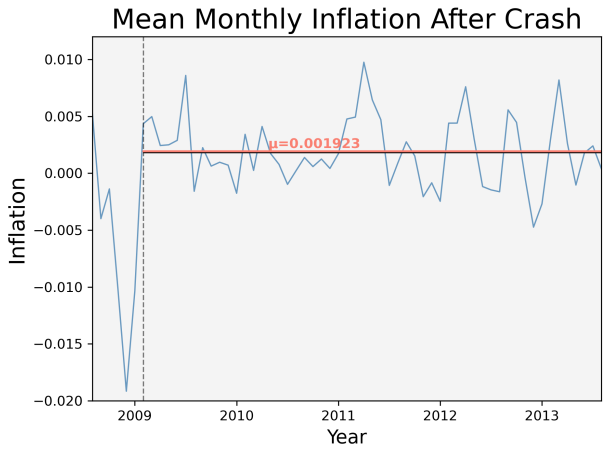

Figure 5.1: Mean value of the training set after the economic crash.

Given that there is no trend but a constant, the de-trended signal is $\text{CPI}_t - \mu,\ \forall t$.

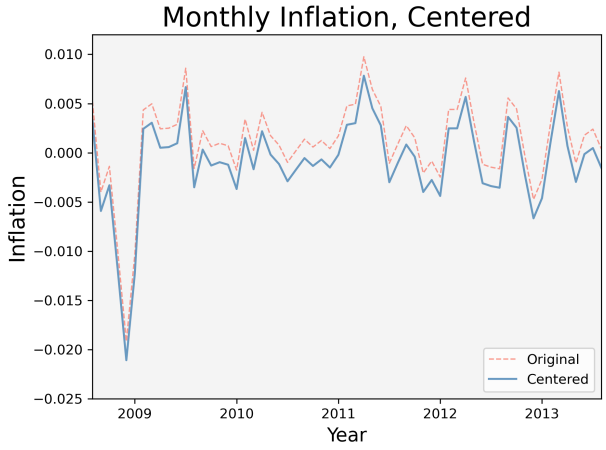

Figure 5.2: The effect of removing the training set mean after the economic crash from inflation.

The autocorrelation function (ACF) should give us a hint as to what kind of signal remains. For example, white noise should be uncorrelated for all $s \neq t$, and moving averages should be uncorrelated outside of some period, such that $\text{Cov}(X_t, X_{t-(p+i)}) = 0,\ \forall i \geq 0$. If the ACF falls off exponentially, but never quite reaches 0, then it may indicate an auto regressive function.

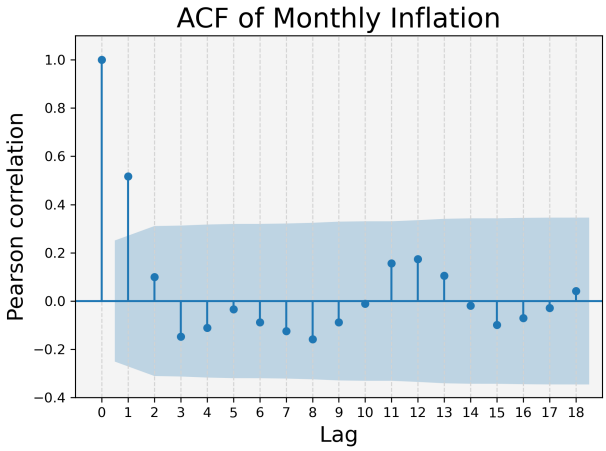

Figure 5.3: Autocorrelation function shows exponentially decreasing autocorrelation, indicating a possible auto regression (AR).

Since autocorrelation extends infinitely in the ACF for AR functions, it is unhelpful in finding the *order* of the signal. Partial autocorrelation function (PACF) show the autocorrelation between two terms that are not captured by the terms in between.

Formally,

$$\alpha_X(h) = \text{Corr}(X_t - \hat{X}_t, X_{t-h} - \hat{X}_{t-h})$$

where $\hat{X}_t$ is the linear regression of X_t on $X_{t-h+1}, X_{t-h+2}, \ldots, X_{t-1}$, and likewise $\hat{X}_{t-h}$ is the linear regression of X_{t-h} on $X_{t-h+1}, X_{t-h+2}, \ldots, X_{t-1}$.

Intuitively, we want to find autocorrelations that are statistically significant, so that H_0 could be rejected. The plot shows such significant partial autocorrelations outside of the blue box.

The PACF plot shows a lag of 1 that is statistically significant, so the signal is AR(1). The updated model is then $\text{AR}(1) + \mu$.

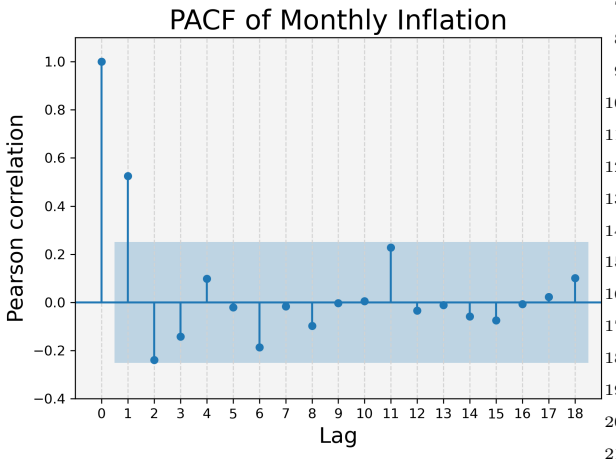

Figure 5.4: Partial autocorrelation function shows an order of 1, so the signal is AR(1).

Expanded, the model could be described as

$$\begin{aligned}\hat{X}_t &= \mu + \phi X_{t-1} \\ &= 0.001923 + 0.535114 X_{t-1}\end{aligned}$$

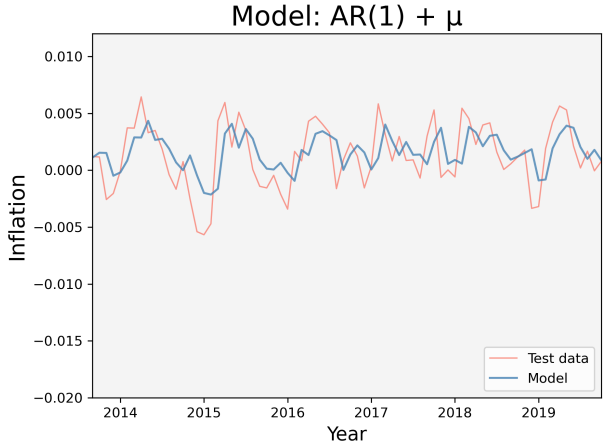

Figure 5.5: The model is a function of the prior term and a constant. Predictions are compared with the validation set.

```
# Skip deflationary period for mean
after_crash = 7
mean = np.mean(
  df.loc[after_crash:N_train, 'CPI'])
args = {'lags': 1, 'trend': 'n',
  'old_names': False}
params = AutoReg(
  df.loc[1:N_train, 'CPI'] - mean,
  **args).fit().params

preds = AutoReg(
  df.loc[:last_index, 'CPI'] - mean,
  **args).predict(params)

y = df.loc[N_train + 1:last_index, 'CPI']

# Add mean to model and get validation
preds += mean
preds = preds.loc[N_train + 1:last_index]

# Calculate RMSE
error = mean_squared_error(y, preds,
  squared=False)
```

Listing 2: Predict using model and calculate RMSE.

The resultant RMSE is **0.002396**.

Part (2)

Some preliminary exploration has shown that the final model turns out not to have the best performance. However, we should keep in mind that we should make the best decisions given the information available in the training set, and that determining a model after a number of trials increases family-wise false positive rate.

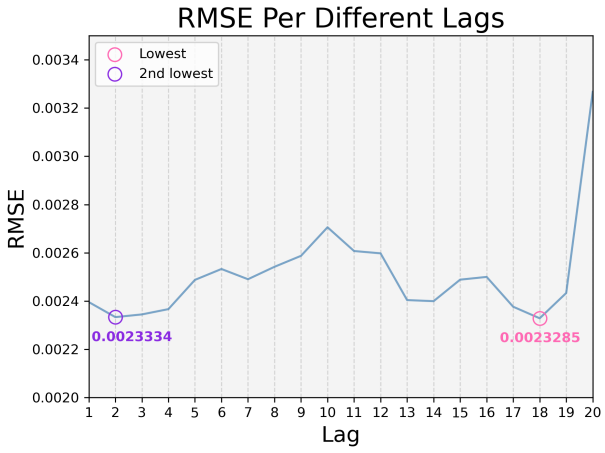

Figure 5.6: Lags that generate best validation RMSE are 18, then 2.

For example, the second lag from PACF is barely insignificant; therefore, I've determined that an AR(2) model generates better validation RMSE. A lower mean $\mu^* \approx 0.0013$ improves RMSE as well. But making arbitrary decisions on trial and error basis overfits the validation set, and given that we have no other holdout test set, this should be avoided at all costs.

Therefore, the model remains what it is.

Part (3)

From our data, we've explored three different series that could be used to construct inflation: consumer price index (CPI), PriceStats, and break-even rate (BER). Over the validation date range, we compare these against the predictions using CPI from the model previously described.

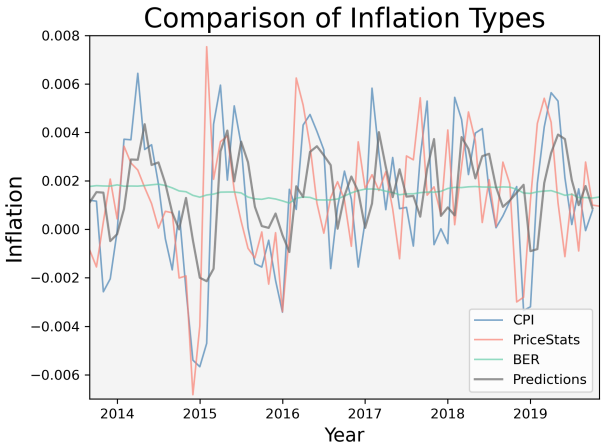

Figure 5.7: Different metrics of inflation compared to predictions.

The predictions lag CPI and PriceStats data by a period, and doesn't swing as wide, highlighting some of the flaws of using this method.

Problem 6
External Regressors and Model Improvements

Part (1)

Exogenous regressors are external time series that might be useful to predict the endogenous variable (i.e. monthly CPI inflation). The model would be updated as

$$\begin{aligned}\text{CPI}_t = \mu &+ \phi_{\text{CPI}}\text{CPI}_{t-1} \\ &+ \phi_{\text{PriceStats}}\text{PriceStats}_{t-\text{lag}_{\text{PriceStats}}} \\ &+ \phi_{\text{BER}}\text{BER}_{t-\text{lag}_{\text{BER}}}\end{aligned}$$

$\text{lag}_{\text{PriceStats}}$ and lag_{BER}, then, should be the most correlated lags between monthly CPI and the respective exogenous variables.

We plot, using `pyplot.xcorr()`, the cross-correlation between de-trended monthly CPI inflation, and monthly PriceStats and BER inflation.

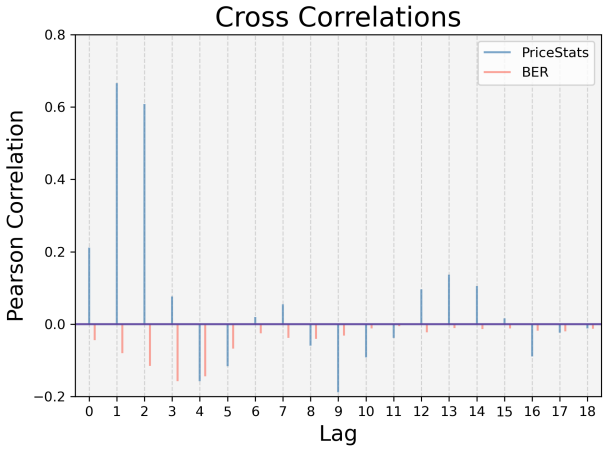

Figure 6.1: Cross-correlation with PriceStats and BER. Note that $\text{lag}_{\text{PriceStats}} = 1$ and $\text{lag}_{\text{BER}} = 3$ are optimal.

Part (2)

The updated model is then

$$\begin{aligned}\text{CPI}_t = \mu &+ \phi_{\text{CPI}}\text{CPI}_{t-1} \\ &+ \phi_{\text{PriceStats}}\text{PriceStats}_{t-1} \\ &+ \phi_{\text{BER}}\text{BER}_{t-3}\end{aligned}$$

Two classes in the `statsmodels` library will be used to fit this model to compare, `AutoReg`, which we've used before, and `SARIMAX` (the latter stands for Seasonal Auto-Regression Integrative Moving Average eXtended). Although, the more advanced features of SARIMAX will be left alone for now. The two give different parameters and results (we'll let them optimize for μ as well).

	`AutoReg`	`SARIMAX`
μ^*	0.000198	-0.001608
ϕ_{CPI}	0.361012	-0.076094
$\phi_{\text{PriceStats}}$	0.589776	0.727741
ϕ_{BER}	-0.155394	0.971676

Figure 6.2: Model parameters with exogenous variables by two different methods.

These are implemented as follows.

```
for lag in range(1, 5):
  df.loc[:, 'PSLast' + str(lag)] =
    df['PSLast'].shift(lag)
  df.loc[:, 'BER' + str(lag)] =
    df['BER'].shift(lag)

exogs = ['PSLast1', 'BER3']
```

Listing 3: Create lagged exogenous variables.

```
args = lambda end: {
  'endog': df.loc[3:end, 'CPI'] - mean,
  'exog': df.loc[3:end, exogs],
  'lags': 1, 'trend': 'n',
  'old_names': False }

train_model = AutoReg(**args(N_train))
params = train_model.fit().params
test_model = AutoReg(**ar_args(last_index))
preds = test_model.predict(params,
  start=N_train - 2, end=last_index - 3)
preds += mean
```

Listing 4: Predict using `AutoReg`.

```
args = lambda start, end: {
  'endog': df.loc[start:end, 'CPI'] - mean,
  'exog': df.loc[start:end, exogs] }

train_model = SARIMAX(**args(3, N_train),
  order=(lag, 0, 0), trend=trend)
model_fit = train_model.fit(
  disp=False, full_output=False)
preds = model_fit.extend(**args(N_train + 1,
  last_index)).fittedvalues
preds += mean
```

Listing 5: Predict using `SARIMAX`.

Part (3)

The error is obtained from the two models, both constrained to the originally calculated $\mu = 0.001923$, and using the new μ^* calculated by the respective methods.

	`AutoReg`	`SARIMAX`
MSE with μ	2.897e-6	3.391e-6
RMSE with μ	0.001702	0.001842
MSE with μ^*	2.889e-6	3.584e-6
RMSE with μ^*	0.001700	0.001893

Figure 6.3: MSE and RMSE with different means and methods.

The best error comes from the predictions reported by `AutoReg` and μ^* (using `trend='c'`). The RMSE is **0.001702**, which is a big improvement over the RMSE without exogenous variables, 0.002396. Interestingly, the original μ gave better results using `SARIMAX`, with SARIMAX being worse in general, despite their similar underlying algorithms. According to `statsmodels` documentation, `AutoReg` uses conditional MLE with ordinary least squares (OLS), and `SARIMAX` uses MLE with Kalman filter, a recursive least squares algorithm.

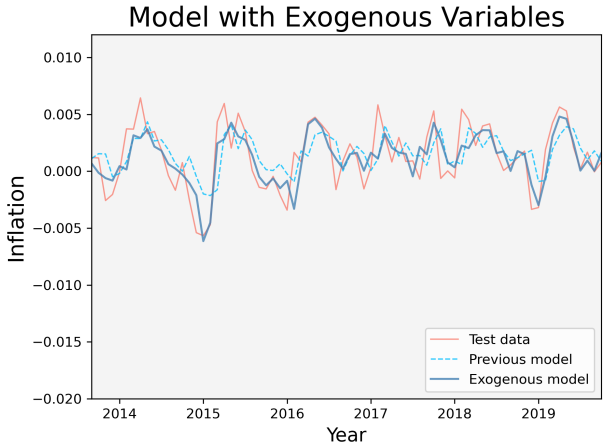

Figure 6.4: Exogenous variables improve the error.

Part (4)

We want to investigate the options within the SARIMAX model. The first thing to note from the prior best model is that BER is biased as a small, positive number. This might not be the most helpful. For example, if a high BER is highly correlated with high CPI-based inflation, then this would produce a positive coefficient. But this would contradict a low BER even if it is correlated with a negative inflation, since the BER is still positive. Therefore, to maximize correlation, we want to center each variable as best as possible, by taking the mean over the training set.

To confirm, we examine the cross-correlation plot again to see that the correlations have in fact improved. In particular, BER is no longer negatively correlated.

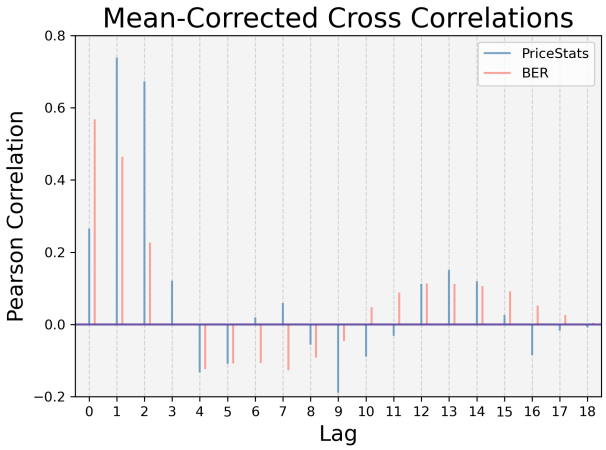

Figure 6.5: Centering exogenous variables improve their cross-correlation.

Throughout the analysis, we've used the last value of each month for PriceStats, and the mean of each month for BER. But it's possible that different values in the month are helpful. We explore the first, mean and last of the month.

```
exogs = ['PSFirstShift1',
  'PSMeanShift1', 'PSMeanShift2',
  'PSLastShift1', 'PSLastShift2',
  'BERShift1', 'BERLastShift1']
```

Listing 6: To avoid too many parameters, only variables with Corr$(\cdot) > 0.4$ make it into the list.

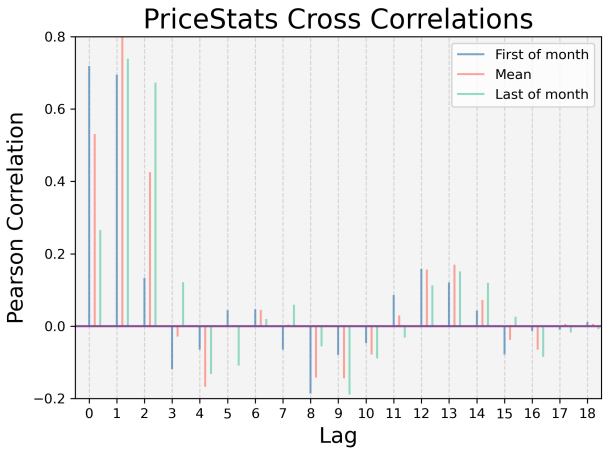

Figure 6.6: Correlations with first value of the month for lag 2 isn't high enough.

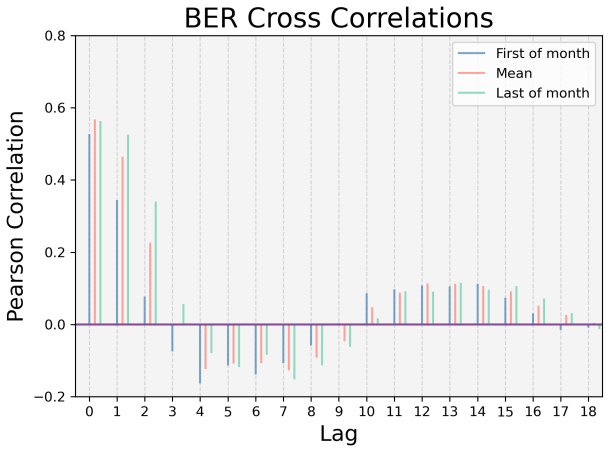

Figure 6.7: Only the mean and last values of the month for lag 1 suffices.

Now with centered exogenous variables and the final variables list, we train a new model with the same parameters as before. Before, the second lag barely missed significance, and we've already confirmed that AR(2) performed better than AR(1). Although repeatedly looking at the validation set is bad practice since it introduces bias, I'll break the rule here once since we're trying to find the best model.

With the same code as before[2] (except `lag=2`), RMSE is now **0.001569**, a small improvement over 0.001702 before.

Now it's time to explore whether seasonality exists in the residuals.

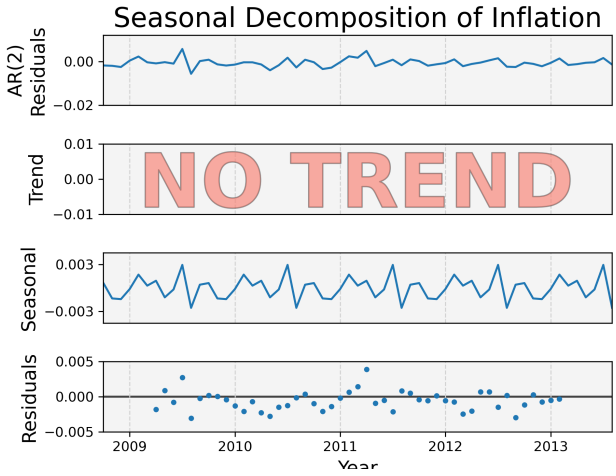

Figure 6.8: Seasonality and correlations in residuals still exist.

There is still seasonality even after accounting for the auto-regression. `statsmodels`'s `seasonal_decompose` method decomposes a signal into trend, seasonality and residuals by running a moving average as the trend. Since we are not fitting the models with a moving average, I have moved the trend component to the residuals. We note that there is still a clear seasonal component, and that the residuals isn't white noise, as there are still some correlations left in the pattern.

We could also look at the ACF plot at seasonal intervals to be sure.

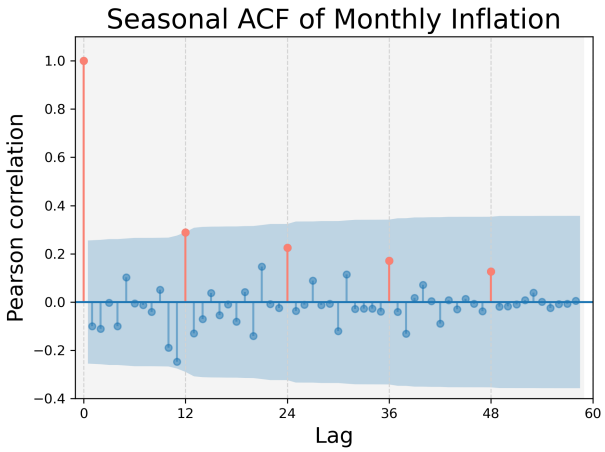

Figure 6.9: Seasonal amplitude doesn't rise above noise floor as in the Mauna Loa CO_2 concentration data set, but there's still good reason to believe it's there.

Although there is 5 months in the training set, there isn't enough observations for the fifth month in lag. In previous exercises, we subtracted a monthly mean to account for seasonality. Using the ACF, note that using an AR model at those intervals is analogous to fitting a weighted mean, with weights optimal to the training set. Since SARIMAX has this functionality, we will use this instead.

After fitting just the seasonal component on top of the AR(2), we end up with an RMSE of **0.001034**, which is another significant improvement from 0.001569 without the seasonality. However, as we've noted before, there are still correlations in the residuals. This could be demonstrated again by observing a PACF plot on the residuals after seasonality.

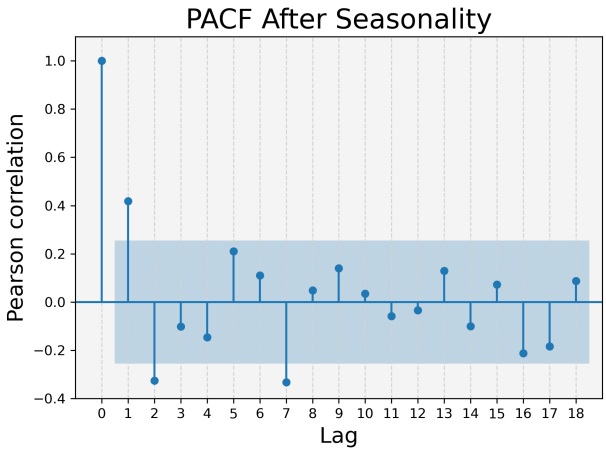

Figure 6.10: An AR(2) in the residuals still.

It turns out, I just needed to fit an AR(2) at the same time as seasonality.

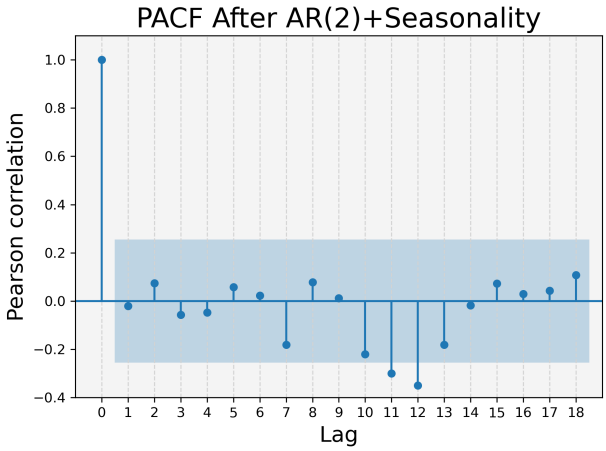

Figure 6.11: No more correlations left in residuals of predictions by an AR(2) + seasonality model.

	RMSE
De-trended model	0.002396
Exogenous variables	0.001702
Centered, multiple lags	0.001569
With seasonality	0.001034
Final model	0.000980

Figure 6.12: Our journey in RMSE.

The final model has an RMSE of **0.000980**, and the following parameters.

	Parameters
PSFirstShift1	0.150098
PSMeanShift1	0.571603
PSLastShift1	0.132754
PSMeanShift2	-0.098212
PSLastShift2	-0.022132
BERShift1	-0.918590
BERLastShift1	1.899677
ar.L1	0.196362
ar.L2	0.249418
ar.S.L12	0.527911
ar.S.L24	0.185787
ar.S.L36	0.141968
ar.S.L48	0.097526
μ	0.001923
σ^2	0.000002

Figure 6.13: Final model parameters.

```
mean = 0.001923 # Mean we found at beginning.
train_model = SARIMAX(
  df.loc[shift:N_train, 'CPI'] - mean,
  exog=(df.loc[shift:N_train, exogs]
    - ext_means[exogs]),
  cov_type='robust', maxiter=250,
  order=(2, 0, 0),
  seasonal_order=(4, 0, 0, 12), trend=trend)
model_fit = train_model.fit(
  disp=False, full_output=False)

y = df.loc[N_train + 1:last_index, 'CPI']

test_model = train_model.extend(
  df.loc[N_train+1:last_index, 'CPI'] - mean,
  exog=(df.loc[N_train+1:last_index, exogs]
    - ext_means[exogs])
preds = test_model.fittedvalues + mean
error = mean_squared_error(y, preds,
  squared=False)
```

Listing 7: Final model in code.

For fun, I performed a grid search on hyperparameters in the perimeter of the final model. The best model is the same, but with AR(4), with an RMSE of 0.000972. But that's cheating.

Here's a plot of final model performance (not the one from the grid search!) over the model with exogenous variables from the previous section.

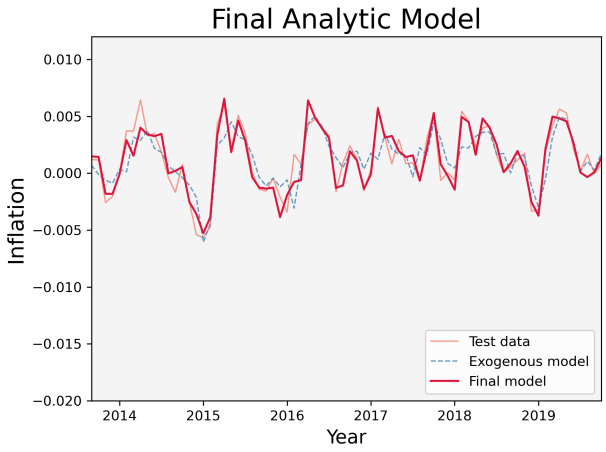

Figure 6.14: Now that's a pretty good fit!.

List of Figures

List of Listings