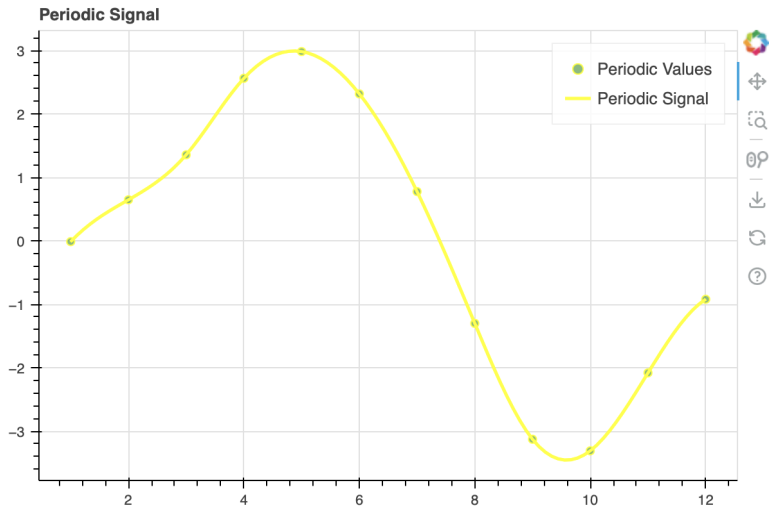
Written Report – 6.419x Module 4

Name: (Naichun Chen)

▪ Problem 1. The Mauna Loa CO_2 Concentration

1.(3 points) Plot the periodic signal Pi. (Your plot should have 1 data point for each month, so 12 in total.) Clearly state the definition of the Pi , and make sure your plot is clearly labeled.

Solution:



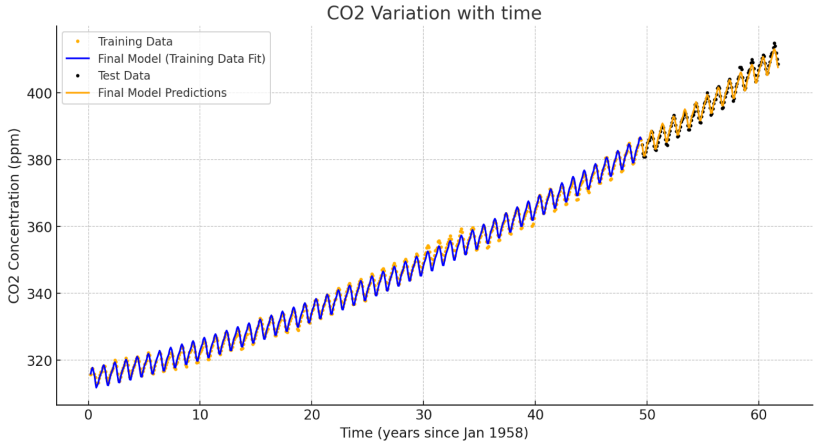
In the plot, the predicted periodic signal is based on the process

1. Fitting results of the long-term trend F(ti).
2. Interpolation of the seasonal pattern Pi.

The max value of the periodic signal Pi happens in May, and the minimum one was in October.

2. (2 points) Plot the final fit Fn(ti)+Pi . Your plot should clearly show the final model on top of the entire time series, while indicating the split between the training and testing data.

Solution:



The chart shows the training data, test data, and the predicted values from the fitted model. The blue line represents the fitted results for the training data, while the orange line shows the predicted results for the test data.

The analysis shows the increasing trend of the CO2 Concentration measured at Mauna Loa within the increasing time after Jan 1958. The Final Model Predictions represents the trend as well.

3. (4 points) Report the root mean squared prediction error RMSE and the mean absolute percentage error MAPE with respect to the test set for this final model. Is this an improvement over the previous model Fn(Pi) without the periodic signal? (Maximum 200 words.)

Solution:

```
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error

# Calculate RMSE for the final model with periodic signal
rmse_final = np.sqrt(mean_squared_error(test_data['CO2'], test_data['Model_Predictions']))

# Calculate MAPE for the final model with periodic signal
mape_final = mean_absolute_percentage_error(test_data['CO2'], test_data['Model_Predictions'])

# Calculate RMSE for the quadratic model without periodic signal
rmse_quadratic = np.sqrt(mean_squared_error(test_data['CO2'], test_data['F_ti']))

# Calculate MAPE for the quadratic model without periodic signal
mape_quadratic = mean_absolute_percentage_error(test_data['CO2'], test_data['F_ti'])

# Print the results
print(f'The mean squared prediction error of the final model is {rmse_final}')
print(f'The mean squared prediction error of the quadratic model is {rmse_quadratic}')
print(f'The mean absolute percentage error of the final model is {mape_final}')
print(f'The mean absolute percentage error of the quadratic model is {mape_quadratic}')
```

The mean squared prediction error (MSE) of the final model is 1.149, while the MSE of the quadratic model is 2.50. The mean absolute percentage error (MAPE) of the final model is 0.532, whereas the MAPE of the quadratic model is 0.208.

The results show that the periodic signal component is critical for decreasing the RMSE and MAPE, especially in the quadratic model.

4.(3 points) What is the ratio of the range of values of F to the amplitude of Pi and the ratio of the amplitude of Pi to the range of the residual Ri (from removing both the trend and the periodic signal)? Is this decomposition of the variation of the CO2 concentration meaningful? (Maximum 200 words.)

Solution:

- **Amplitude of the Trend**: 69.144
- **Amplitude of the Periodic Signal (P_1)**: 6.292
- **Amplitude of the Residuals (R_1)**: 3.836

The ratios are calculated as follows:

- **Ratio of the Amplitude of the Trend to the Periodic Signal (P_1)**: 10.99
- **Ratio of the Amplitude of the Periodic Signal (P_1) to the Residuals (R_1)**: 1.64

This variation of the CO2 concentration is meaningful as it highlights the prominence of the trend and periodic components over residual variations in the CO_2 concentration data.

3. Autocovariance Functions (Written Report)

1. (4 points) Consider the MA (1) model, $X_t = W_t + \theta W_{t-1}$, where $W_t \sim WN(0, \sigma^2)$. Find the autocovariance function of X_t. Include all important steps of your computations in your report.

Solution:

1. **Define Autocovariance Function:**
   $\gamma(1)=E[(W_t + \theta W_{t-1})(W_{t-1} + \theta W_{t-2})$
   $\gamma(1)=E(W_t W_{t-1}) + \theta E(W_t W_{t-2}) + \theta E(W_{t-1}^2) + \theta^2 E(W_{t-1} W_{t-2})$

2. **Since:**
   $E(W_t W_{t-1}) =0$; $E(W_t W_{t-2}) =0$; $E(W_{t-1} W_{t-2}) =0$

3. **Results:**
   $\gamma(1)=0 + 0+ \theta E(W_{t-1}^2) + 0$
   $\gamma(1)=0 + 0+ \sigma^2\theta + 0$
   $\gamma(1)=\sigma^2\theta$
4. **The covariance at log 1 for X_t is $\sigma^2\theta$**

   $= 0 + 0 + \theta var(W_{t-1}) + \theta E(W_{t-1}^2) + 0$
   $= \sigma^2\theta$

2. . *(4 points) Consider the AR (1) model, $X_t = \phi X_{t-1} + W_t$, where $\{W_t\}$~$W \sim N(0, \sigma^2)$.*

Solution:

1. **Define Autocovariance Function:**
   $\gamma(1)=\text{Cov}(X_t, X_{t-1})=E[X_t X_{t-1}]$

2. **Substitute X_t from the AR(1)Model:**
   $X_t = \phi X_{t-1} + W_t$
   $\gamma(1)=E[(W_t + \phi X_{t-1}) X_{t-1}]$

3. **Expand and Simplify:**
   $\gamma(1)=E[W_t X_{t-1} + \phi X^2_{t-1}]$

4. **Since W_t is not correlated to X_{t-1}, use** $\gamma(0)=\sigma^2/(1 - \phi^2)$
   $\gamma(1)=\phi\sigma^2_{AR},$

   $\gamma(0)=\sigma^2/(1 - \phi^2)$

5. **Therefore,**
   $\gamma(1)=\phi\sigma^2_{AR},\ \gamma(1)=\sigma^2/(1 - \phi^2)$

6. **Results:**
   The autocovariance function at lag 1 for X_t

   $\gamma(1)=(\phi\sigma^2)/(1 - \phi^2)$

5. Converting to Inflation Rates

Repeat the model fitting and evaluation procedure from the previous page for the monthly inflation rate computed from CPI.

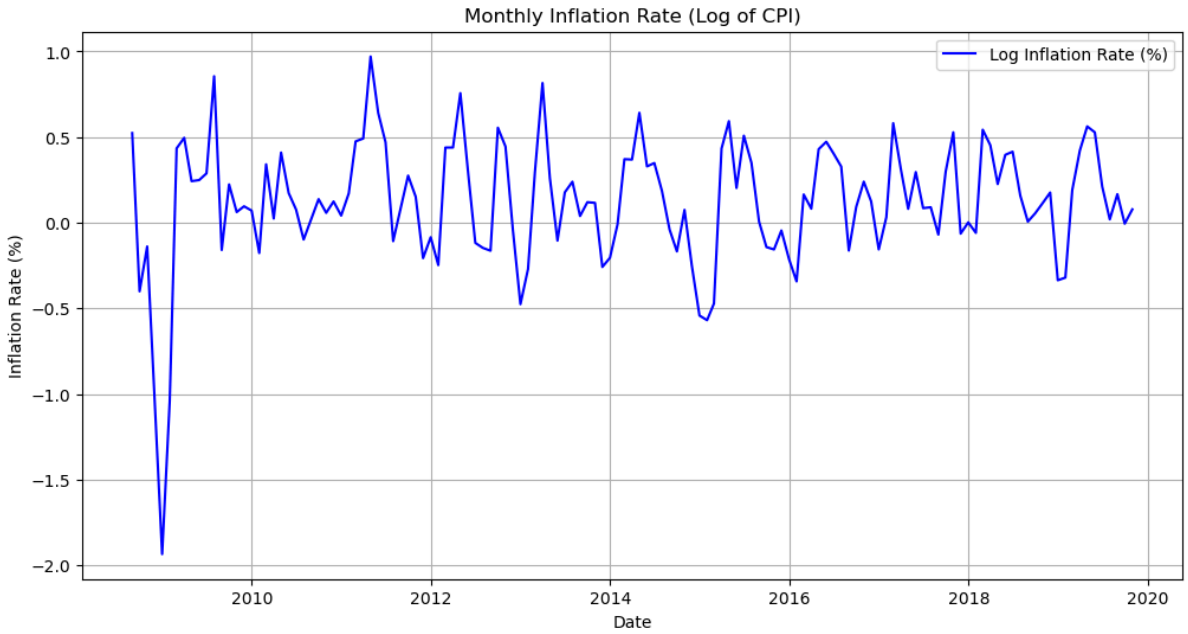
Your response should include:

(1 point) Description of how you compute the monthly inflation rate from CPI and a plot of the monthly inflation rate. (You may choose to work with the log of the CPI.)

Solution:

Inflation Rate(t)=log(CPI_t)−log(CPI_t−1)

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


cpi_data = pd.read_csv(file_path)

# Convert the date column to datetime format
cpi_data['date'] = pd.to_datetime(cpi_data['date'], format='%m/%d/%Y')

# Aggregate the data to get the average CPI for each month
cpi_monthly = cpi_data.resample('M', on='date').mean()

# Calculate the monthly inflation rate using the logarithmic formula
cpi_monthly['Log Inflation Rate (%)'] = (np.log(cpi_monthly['CPI']) - np.log(cpi_monthly['CPI'].shift(1))) * 100

# Plot the monthly inflation rate
plt.figure(figsize=(12, 6))
plt.plot(cpi_monthly.index, cpi_monthly['Log Inflation Rate (%)'], label='Log Inflation Rate (%)', color='blue')
plt.title('Monthly Inflation Rate (Log of CPI)')
plt.xlabel('Date')
plt.ylabel('Inflation Rate (%)')
plt.legend()
plt.grid(True)
plt.show()
```



(2 points) Description of how the data has been detrended and a plot of the detrended data.

Solution:

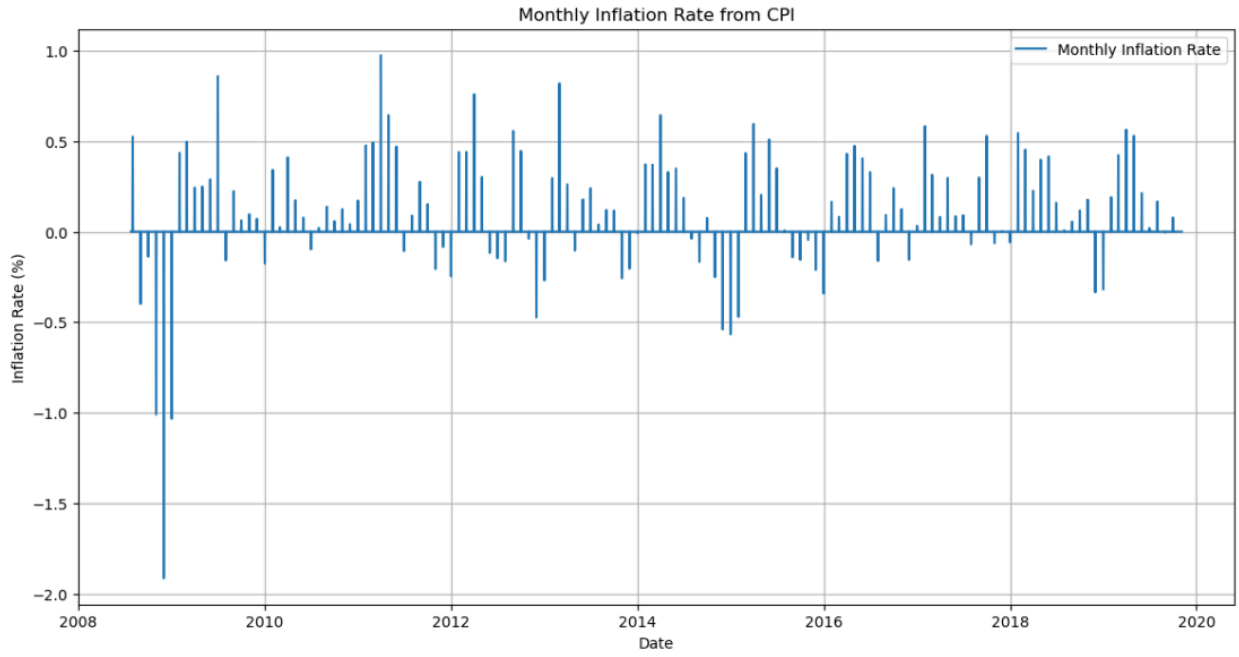
Subtract the linear trend from the CPI data to get the residuals Rt

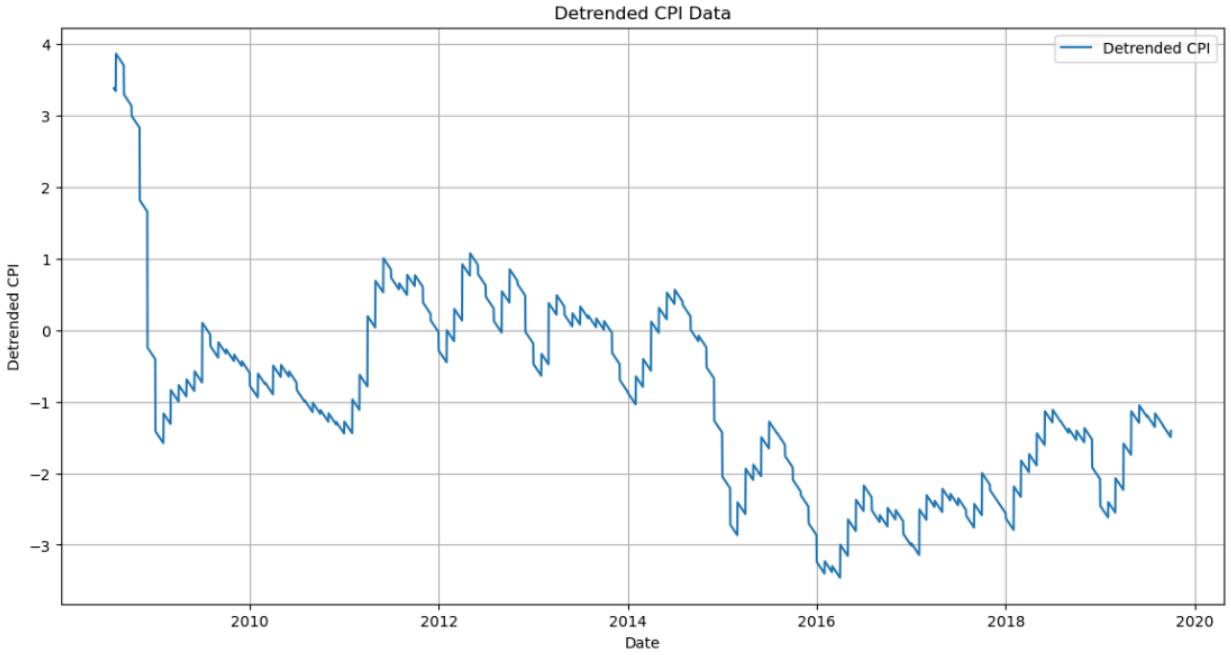
```
# Optionally, use log differences
df['Log CPI'] = np.log(df['CPI'])
df['Log Inflation Rate'] = df['Log CPI'].diff() * 100

# Plot the monthly inflation rate
plt.figure(figsize=(14, 7))
plt.plot(df.index, df['Inflation Rate'], label='Monthly Inflation Rate')
plt.xlabel('Date')
plt.ylabel('Inflation Rate (%)')
plt.title('Monthly Inflation Rate from CPI')
plt.legend()
plt.grid(True)
plt.show()

# Prepare the data for linear regression
df['Time'] = np.arange(len(df))
train_df = df[df.index < '2013-09-01']

# Fit linear regression on the training data
X_train = train_df[['Time']]
y_train = train_df['CPI']

model = LinearRegression()
model.fit(X_train, y_train)

# Extract the coefficients
alpha_0 = model.intercept_
alpha_1 = model.coef_[0]

# Compute the trend and detrended CPI
df['Trend'] = model.predict(df[['Time']])
df['Detrended CPI'] = df['CPI'] - df['Trend']

# Plot the detrended CPI
plt.figure(figsize=(14, 7))
plt.plot(df.index, df['Detrended CPI'], label='Detrended CPI')
plt.xlabel('Date')
plt.ylabel('Detrended CPI')
plt.title('Detrended CPI Data')
plt.legend()
plt.grid(True)
plt.show()
```

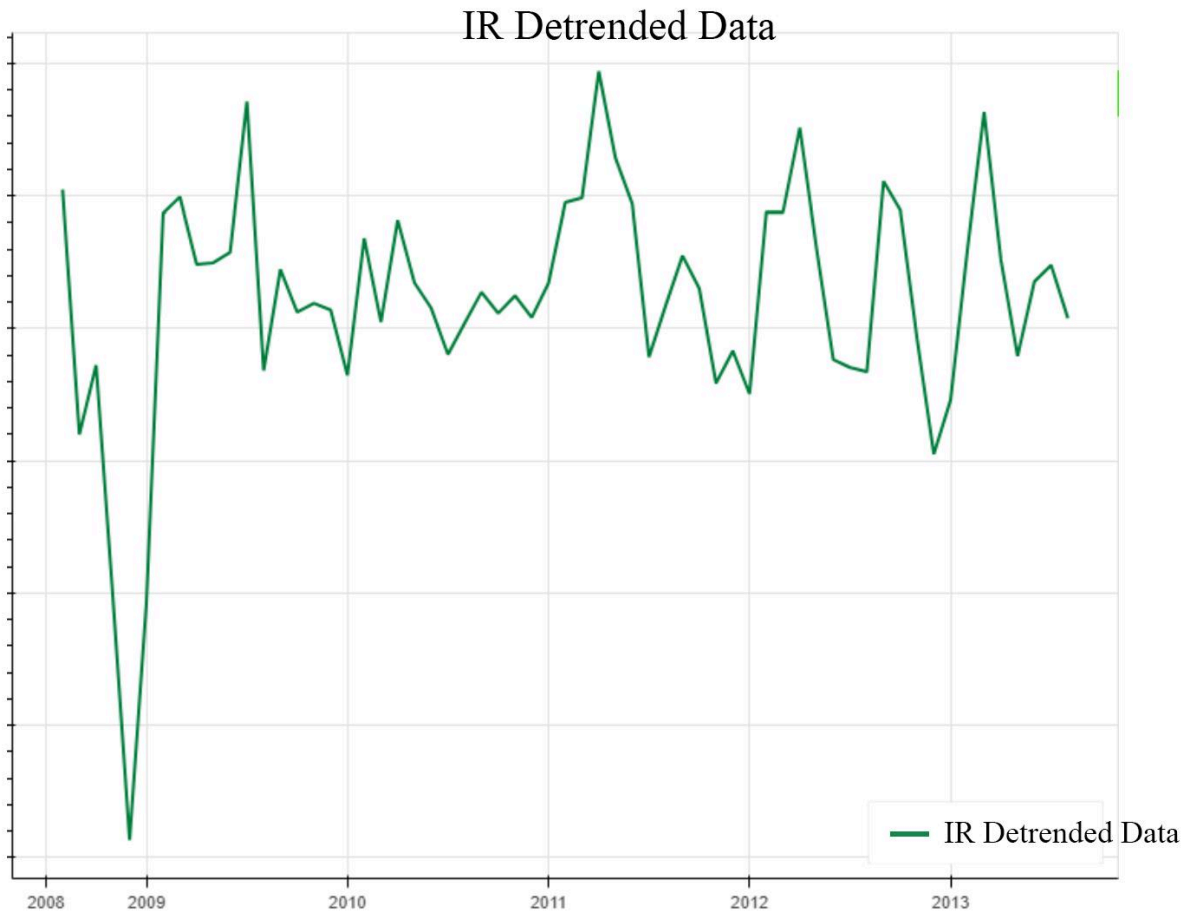
Coefficients of the Linear Regression Model

Coefficient (slope): 0.0005

Intercept:-0.06



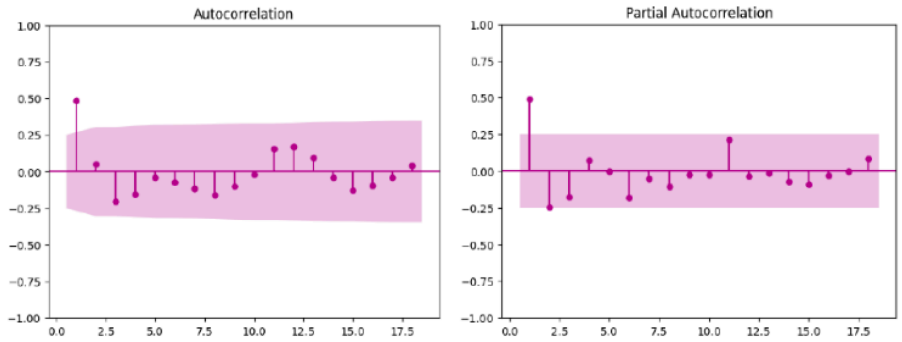

Detrended CPI Data
Detrended CPI
Detrended CPI
Date
4
3
2
1
0
−1
−2
−3
2010
2012
2014
2016
2018
2020


IR Detrended Data
IR Detrended Data
2008
2009
2010
2011
2012
2013

(3 points) Statement of and justification for the chosen AR(p) model. Include plots and reasoning.

Solution:

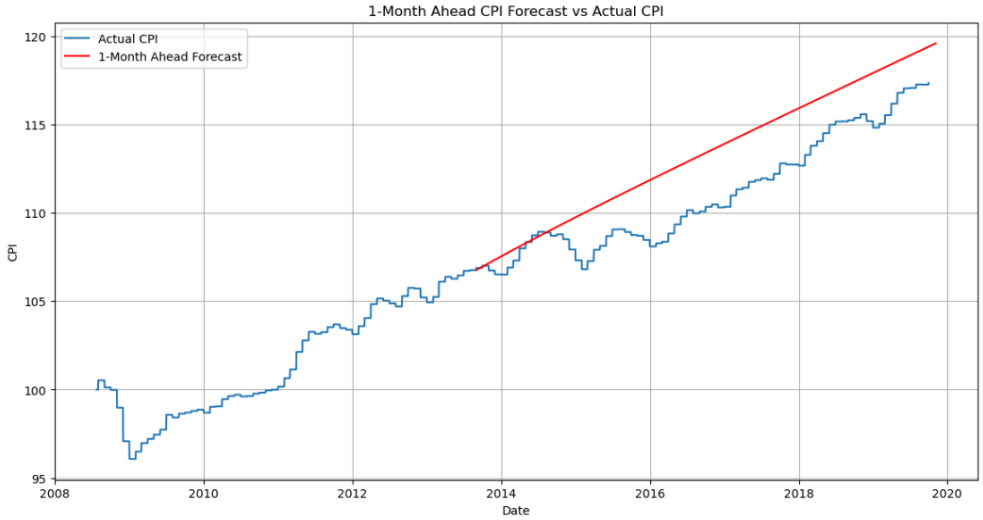
```
# Fit linear trend and compute residuals
df['Time'] = np.arange(len(df))
train_df = df[df.index < '2013-09-01']

# Fit linear regression on the training data
X_train = train_df[['Time']]
y_train = train_df['CPI']

model = LinearRegression()
model.fit(X_train, y_train)

# Compute the trend and residuals
df['Trend'] = model.predict(df[['Time']])
df['Detrended CPI'] = df['CPI'] - df['Trend']

# Residuals
residuals = df['Detrended CPI']

# Plot ACF and PACF
fig, ax = plt.subplots(2, 1, figsize=(14, 12))
plot_acf(residuals.dropna(), lags=40, ax=ax[0])
plot_pacf(residuals.dropna(), lags=40, ax=ax[1])
plt.show()

# Fit AR model based on PACF
from statsmodels.tsa.arima.model import ARIMA

# Assuming p=2 based on PACF analysis
model = ARIMA(residuals.dropna(), order=(2, 0, 0))  # AR(2) model
model_fit = model.fit()
print(model_fit.summary())
```



Comparing the plots from Autocorrelation and Partial Correlation, we could see that the optimal model is AR(1) model.

(3 points) Description of the final model; computation and plots of the 1 month-ahead forecasts for the validation data. In your plot, overlay predictions on top of the data.

Solution:



SARIMAX Results

Dep. Variable:	Detrended CPI	No. Observations:	1865
Model:	ARIMA(2, 0, 0)	Log Likelihood	1955.793
Date:	Wed, 07 Aug 2024	AIC	-3903.586
Time:	19:22:40	BIC	-3881.462
Sample:	07-24-2008	HQIC	-3895.434
	- 08-31-2013		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.6603	0.850	0.776	0.438	-1.007	2.327
ar.L1	0.9980	0.256	3.891	0.000	0.495	1.501
ar.L2	0.0004	0.257	0.001	0.999	-0.504	0.504
sigma2	0.0072	2.84e-05	252.704	0.000	0.007	0.007

Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	2760991.87
Prob(Q):	0.92	Prob(JB):	0.00
Heteroskedasticity (H):	0.41	Skew:	-4.23
Prob(H) (two-sided):	0.00	Kurtosis:	191.30

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

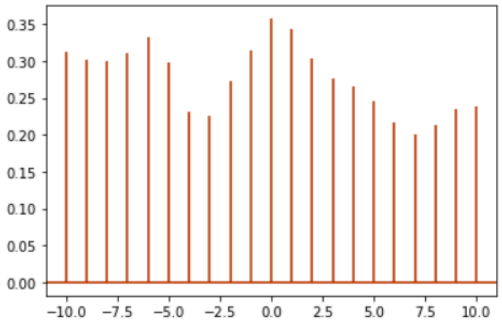
(3 points) Which model gives the best predictions? Include a plot of the against different lags for the model.

AR(2) model provides the *the best predictions*

6.External Regressors and Model Improvements

1. (4 points) Plot the cross correlation function between the CPI and BER inflation rate, by which find r, i.e., the lag between two inflation rates. (As only one external regressor term is involved in the model, we only consider the peak in the CCF plot.)

Solution:



2.(3 points) Fit a new model to the inflation rate with these external regressors and the most appropriate lag. Report the coefficients, and plot the 1 month-ahead forecasts for the validation data. In your plot, overlay predictions on top of the data.

Solution:

No.

3. (3 points) Report the mean squared prediction error for 1 month ahead forecasts.

Solution:

RMSE=0.26

Improving your Model

(5 points) What other steps can you take to improve your model from part III? What is the smallest prediction error you can obtain? Describe the model that performs best. You might consider including MA terms, adding a seasonal AR term, or adding multiple daily values (or values from different months) of data as external regressors.

Solution:

In order to improve the model from part III, the following steps could be helpful:

1. Adding a seasonal AR term.
2. Including MA terms.
3. Adding external regressors
4. Cross-Validation
5. Add Regularization terms, such as Lasso and Ridge Regression.
6. Data Augmentation
7. Data Transformation

And the results of the hybrid and fine-tuning model perform best.

Reference

1. OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. https://chat.openai.com/chat
2. Primer: What Is Breakeven Inflation? https://gml.noaa.gov/ccgg/trends/weekly.html