In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
from tqdm import tqdm
from sklearn.metrics import mean_squared_error, r2_score
```

In [ ]:
```
# hide warning messages
import warnings
warnings.filterwarnings("ignore")
```

In [ ]:
```
# better plots
sns.set(rc={'figure.figsize':(12,8)});
```

In [ ]:
```
directory = os.path.dirname(os.getcwd())
directory
```

Out[ ]:
```
'd:\\github\\AssignmentEconometricsIV'
```

In [ ]:
```
# read the data
input_path = f'{directory}\\data\\stacionarized_cpi.csv'
df = pd.read_csv(input_path)
df['date'] = pd.to_datetime(df['date'])
df = df.set_index('date')
```

In [ ]:
```
df.head()
```

Out[ ]:

| date | RPI | W875RX1 | DPCERA3M086SBEA | RETAILx | INDPRO | IPFPNSS | IPFINAL | IPCONGD |
|---|---|---|---|---|---|---|---|---|
| 1959-03-01 | 0.643011 | 0.735934 | 0.941009 | 0.832120 | 1.430253 | 0.603609 | 0.489927 | 0.000000 |
| 1959-04-01 | 0.649412 | 0.704864 | -0.363947 | 0.061571 | 2.107741 | 1.433803 | 1.454234 | 1.565338 |
| 1959-05-01 | 0.576311 | 0.661646 | 1.200535 | 0.780340 | 1.495024 | 0.826920 | 0.958304 | 0.476849 |
| 1959-06-01 | 0.310244 | 0.297379 | 0.370829 | 0.906434 | 0.114438 | 0.703445 | 0.712642 | -0.476849 |
| 1959-07-01 | -0.058921 | -0.076384 | -0.342687 | -0.033018 | -2.423797 | 0.116693 | 0.824692 | 1.305596 |

5 rows × 104 columns

In [ ]:
```
# set the amount of lags
lags = 2
```

In [ ]:
```
# create lagged variables
for col in df.columns:
    for i in range(lags):
        name_col = col + f"(-{i+1})"
        df[name_col] = df[col].shift(i+1)
```

In [ ]:
```
# inflation rate ahead
df['CPIAUCSL(+1)'] = df['CPIAUCSL'].shift(-1)
```

In [ ]:
```
# drop nan rows
df = df.dropna()
```

In [ ]:
```
# plot Inflation rate of US
plt.plot(df['CPIAUCSL'], color='red', label='Inflation rate', linewidth = 0.5)

plt.xlabel(f'Time ($t$)')
plt.ylabel('Inflation Rate US (CPIAUCSL)')
plt.legend()
plt.show()
```


In [ ]:
```
import glmnet_python.glmnet_python
from cvglmnet import cvglmnet
from cvglmnetPredict import cvglmnetPredict
from cvglmnetPlot import cvglmnetPlot
from cvglmnetCoef import cvglmnetCoef

from functions.linear_models import Ridge, LASSO
```

In [ ]:
```
# set variables
y = df["CPIAUCSL(+1)"]
```

```
X = df.drop("CPIAUCSL(+1)", axis=1)
```

In [ ]:
```
df
```

Out[ ]:

| date | RPI | W875RX1 | DPCERA3M086SBEA | RETAILx | INDPRO | IPFPNSS | IPFINAL | IPCONG |
|---|---|---|---|---|---|---|---|---|
| 1959-05-01 | 0.576311 | 0.661646 | 1.200535 | 0.780340 | 1.495024 | 0.826920 | 0.958304 | 0.4768 |
| 1959-06-01 | 0.310244 | 0.297379 | 0.370829 | 0.906434 | 0.114438 | 0.703445 | 0.712642 | -0.4768 |
| 1959-07-01 | -0.058921 | -0.076384 | -0.342687 | -0.033018 | -2.423797 | 0.116693 | 0.824692 | 1.3055 |
| 1959-08-01 | -0.563656 | -0.574751 | 0.600333 | 0.636421 | -3.446532 | -0.702622 | -0.234750 | 0.1178 |
| 1959-09-01 | 0.072136 | 0.000000 | 1.001845 | -1.315700 | -0.120914 | -0.470898 | -0.353804 | -0.3537 |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| 2021-06-01 | -0.268168 | 0.246839 | 0.588917 | 0.848804 | 0.546727 | 0.002528 | 0.114008 | -0.2029 |
| 2021-07-01 | 0.803329 | 0.376098 | -0.313442 | -1.637075 | 0.767226 | 1.461057 | 1.668025 | 0.9707 |
| 2021-08-01 | -0.041013 | -0.094940 | 0.727968 | 1.152697 | -0.136496 | -0.232743 | -0.485054 | -0.4072 |
| 2021-09-01 | -1.333562 | 0.169571 | 0.264718 | 0.738929 | -1.018960 | -0.235388 | -0.539534 | -0.5550 |
| 2021-10-01 | -0.226827 | 0.002789 | 0.740685 | 1.769110 | 1.664607 | 1.015773 | 1.119566 | 1.0356 |

750 rows × 313 columns

In [ ]:
```
# add forecast columns
df["CPIAUCSL_estimated_Ridge"] = np.nan
df["CPIAUCSL_estimated_LASSO"] = np.nan
```

In [ ]:
```
# set some parameters
rolling_window = 490 - lags
T = len(y) - rolling_window
```

In [ ]:
```
# list of forecast squared errors
errors_Ridge = []
errors_LASSO = []

for t in tqdm(range(T), desc='Processing for time'):
    # predict date
    date = y[[rolling_window+t]].index

    # estimation sets
    X_train = X[t:(rolling_window+t)]
```

```
    y_train = y[t:(rolling_window+t)]

    # forecast sets
    X_test = X.iloc[[rolling_window+t]]
    y_test = y[rolling_window+t]

    # estimations
    y_pred_Ridge, error_pred_Rigde = Ridge(X_train, y_train, X_test, y_test)
    y_pred_LASSO, error_pred_LASSO = LASSO(X_train, y_train, X_test, y_test)

    # fill forecast columns
    df["CPIAUCSL_estimated_Ridge"][date] = y_pred_Ridge
    df["CPIAUCSL_estimated_LASSO"][date] = y_pred_LASSO

    # append forecast squared errors
    errors_Ridge.append(error_pred_Rigde)
    errors_LASSO.append(error_pred_LASSO)
```

```
Processing for time:   0%|          | 0/262 [00:00<?, ?it/s]Processing for time: 10
0%|██████████| 262/262 [58:23<00:00, 13.37s/it]
```

In [ ]:
```
df[["CPIAUCSL", "CPIAUCSL_estimated_Ridge", "CPIAUCSL_estimated_LASSO"]][rolling_win
```

Out[ ]:

| date | CPIAUCSL | CPIAUCSL_estimated_Ridge | CPIAUCSL_estimated_LASSO |
|---|---|---|---|
| **2000-01-01** | 0.295334 | 0.095432 | 0.161590 |
| **2000-02-01** | 0.411765 | 0.347451 | 0.410884 |
| **2000-03-01** | 0.584795 | 0.393511 | 0.467927 |
| **2000-04-01** | -0.058514 | 0.016565 | 0.052578 |
| **2000-05-01** | 0.175234 | 0.221143 | 0.177611 |
| **...** | ... | ... | ... |
| **2021-06-01** | 0.896742 | 0.271468 | 0.484639 |
| **2021-07-01** | 0.471599 | 0.089117 | 0.225026 |
| **2021-08-01** | 0.273614 | 0.157016 | 0.208878 |
| **2021-09-01** | 0.410742 | 0.237001 | 0.323774 |
| **2021-10-01** | 0.934505 | 0.375331 | 0.512426 |

262 rows × 3 columns

In [ ]:
```
# prediction dates
prediction_dates = list(X.iloc[(rolling_window):].index)
```

In [ ]:
```
# treat the error list
errors_Ridge_ = [item[0] for item in errors_Ridge]
errors_LASSO_ = [item[0] for item in errors_LASSO]

# compute cumulated mse
cum_errors_Ridge = list(np.cumsum(errors_Ridge_))
cum_errors_LASSO = list(np.cumsum(errors_LASSO_))
```

In [ ]:
```
errors = {'Cumulated_MSE_Ridge': cum_errors_Ridge, 'Cumulated_MSE_LASSO': cum_errors
mse = pd.DataFrame(errors, index=prediction_dates)
```

In [ ]:
```
mse
```

Out[ ]:

| | Cumulated_MSE_Ridge | Cumulated_MSE_LASSO |
|---|---|---|
| **2000-01-01** | 0.100067 | 0.062587 |
| **2000-02-01** | 0.156399 | 0.092832 |
| **2000-03-01** | 0.360726 | 0.369972 |
| **2000-04-01** | 0.385901 | 0.385016 |
| **2000-05-01** | 0.515197 | 0.547513 |
| **...** | ... | ... |
| **2021-06-01** | 22.719491 | 17.702431 |
| **2021-07-01** | 22.753531 | 17.704792 |
| **2021-08-01** | 22.817907 | 17.745541 |
| **2021-09-01** | 23.304419 | 18.118534 |
| **2021-10-01** | 23.462633 | 18.186481 |

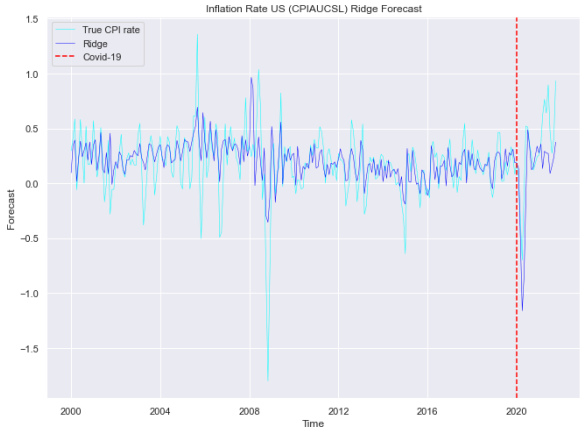
262 rows × 2 columns

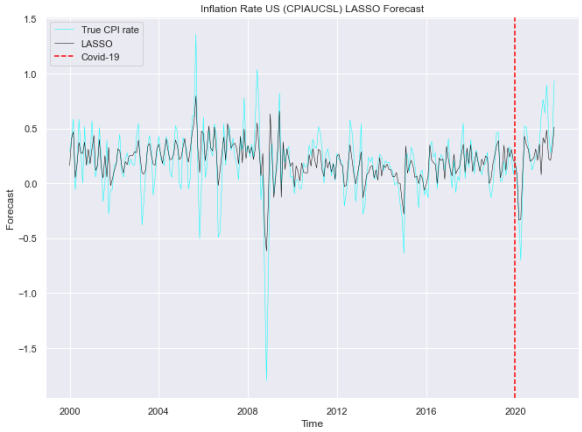
In [ ]:
```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(24, 8))

# True CPI
axes[0].plot(df['CPIAUCSL'][rolling_window:], color='cyan', label='True CPI rate', l
# Ridge forecasts
axes[0].plot(df["CPIAUCSL_estimated_Ridge"], color='blue', label='Ridge', linewidth=
# adding a vertical line at 2020, January
axes[0].axvline(x=mse.index[240], color='red', linestyle='--', label='Covid-19')
# labels
axes[0].set_xlabel('Time')
axes[0].set_ylabel('Forecast')
axes[0].set_title('Inflation Rate US (CPIAUCSL) Ridge Forecast')
axes[0].legend()

# True CPI
axes[1].plot(df['CPIAUCSL'][rolling_window:], color='cyan', label='True CPI rate', l
# LASSO forecasts
axes[1].plot(df["CPIAUCSL_estimated_LASSO"], color='black', label='LASSO', linewidth
# adding a vertical line at 2020, January
axes[1].axvline(x=mse.index[240], color='red', linestyle='--', label='Covid-19')
# labels
axes[1].set_xlabel('Time')
axes[1].set_ylabel('Forecast')
axes[1].set_title('Inflation Rate US (CPIAUCSL) LASSO Forecast')
axes[1].legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x1fd0b4e8910>
```




In [ ]:

```
plt.plot(df['CPIAUCSL'][rolling_window:], color='cyan', label='True CPI rate', linew

plt.plot(df["CPIAUCSL_estimated_Ridge"], color='blue', label='Ridge', linewidth = 0.

plt.plot(df["CPIAUCSL_estimated_LASSO"], color='black', label='LASSO', linewidth = 0

# adding a vertical line at 2020, January
plt.axvline(x=mse.index[240], color='red', linestyle='--', label='Covid-19')

plt.xlabel('Time')
plt.ylabel('Inflation Rate US (CPIAUCSL) Forecasts')
plt.legend()
plt.show()
```


In [ ]:

```
# plot of cumulated MSE
plt.plot(mse["Cumulated_MSE_Ridge"], color='blue', label="Ridge", linewidth = 0.5)

# plot of cumulated MSE
plt.plot(mse["Cumulated_MSE_LASSO"], color='black', label="LASSO", linewidth = 0.5)

# adding a vertical line at 2020, January
plt.axvline(x=mse.index[240], color='red', linestyle='--', label='Covid-19')
```

```
plt.xlabel('Time')
plt.ylabel('Cumulated MSE')
plt.title('Inflation Rate US (CPIAUCSL) Cumulated MSE')
plt.legend()
plt.show()
```