macrovariables

June 2, 2024

[ ]:

```
import pandas as pd
import numpy as np
from functools import reduce
```

[ ]:

```
#### USD to EUR Exchange Rate ####
usd_eur_exchange = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪usd_eur_exchange.csv', header=0, names=['date', 'usd_eur_exchange'])
# Convert the 'Date' column to datetime
usd_eur_exchange['date'] = pd.to_datetime(usd_eur_exchange['date'])
usd_eur_exchange = usd_eur_exchange.sort_values(by='date')
# remove missing values
usd_eur_exchange = usd_eur_exchange.replace('.', np.nan).dropna()
# Filter the data to start from 01/01/2004
usd_eur_exchange = usd_eur_exchange[usd_eur_exchange['date'] >= '2005-01-01']
usd_eur_exchange = usd_eur_exchange[usd_eur_exchange['date'] <= '2024-01-01']
#length
print(len(usd_eur_exchange['usd_eur_exchange']))
```

[ ]:

```
#### Brent ####
brent = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/brent_filtered.
 ↪csv', header=0, names=['date', 'brent'])
# Convert the 'Date' column to datetime
brent['date'] = pd.to_datetime(brent['date'])
# Interpolate missing values
brent['brent'] = brent['brent'].interpolate(method='linear')
# remove missing values
brent = brent.replace('.', np.nan)
#remove dates that are missing from exchange rate
brent = brent[brent['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].isin(brent['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['brent'] = np.nan
# Append the missing dates to the new timeseries
brent = pd.concat([brent, missing_dates_timeseries]).sort_values(by='date').
 ↪reset_index(drop=True)
# replace '.' to 'nan'
```

```python
brent = brent.replace('.', np.nan)
# Make values float type
brent['brent'] = brent['brent'].replace('.', np.nan).astype(float)
# sort by date
brent = brent.sort_values(by='date')
# Interpolate missing values
brent['brent'] = brent['brent'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
brent = brent[brent['date'] >= '2005-01-01']
brent = brent[brent['date'] <= '2024-01-01']
#length
print(len(brent['brent']))
```

[ ]:

```python
#### EU CPI ####
eu_cpi = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/eu_cpi.csv',␣
 ↪delimiter=';', header=0, names=['date', 'eu_cpi'])
# Convert the 'Date' column to datetime
eu_cpi['date'] = pd.to_datetime(eu_cpi['date'])
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].isin(eu_cpi['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eu_cpi'] = np.nan
# Append the missing dates to the new timeseries
eu_cpi = pd.concat([eu_cpi, missing_dates_timeseries]).sort_values(by='date').
 ↪reset_index(drop=True)
# replace '.' to 'nan'
eu_cpi = eu_cpi.replace('.', np.nan)
# Make values float type
eu_cpi['eu_cpi'] = eu_cpi['eu_cpi'].replace('.', np.nan).astype(float)
# sort by date
eu_cpi = eu_cpi.sort_values(by='date')
# Fill NaN values using forward fill method
eu_cpi['eu_cpi'] = eu_cpi['eu_cpi'].fillna(method='ffill')
# Filter the data to start from 01/01/2004
eu_cpi = eu_cpi[eu_cpi['date'] >= '2005-01-01']
eu_cpi = eu_cpi[eu_cpi['date'] <= '2024-01-01']
#length
print(len(eu_cpi['eu_cpi']))
```

[ ]:

```python
#### EU MRO Rate ####
eu_mro_rate = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/eu_mro_rate.
 ↪csv', delimiter=';', header=0, names=['date', 'eu_mro_rate'])
# Convert the 'Date' column to datetime
eu_mro_rate['date'] = pd.to_datetime(eu_mro_rate['date'])
# remove missing values
eu_mro_rate = eu_mro_rate.replace('.', np.nan)
#remove dates that are missing from exchange rate
```

```
eu_mro_rate = eu_mro_rate[eu_mro_rate['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(eu_mro_rate['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eu_mro_rate'] = np.nan
# Append the missing dates to the new timeseries
eu_mro_rate = pd.concat([eu_mro_rate, missing_dates_timeseries]).
 ↪sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
eu_mro_rate = eu_mro_rate.replace('.', np.nan)
# Make values float type
eu_mro_rate['eu_mro_rate'] = eu_mro_rate['eu_mro_rate'].replace('.', np.nan).
 ↪astype(float)
# sort by date
eu_mro_rate = eu_mro_rate.sort_values(by='date')
# Interpolate missing values
eu_mro_rate['eu_mro_rate'].interpolate(method='linear', inplace=True)
# Filter the data to start from 01/01/2004
eu_mro_rate = eu_mro_rate[eu_mro_rate['date'] >= '2005-01-01']
eu_mro_rate = eu_mro_rate[eu_mro_rate['date'] <= '2024-01-01']
#length
print(len(eu_mro_rate['eu_mro_rate']))
```

[ ]:

```
#### EU Unemployment Rate ####
eu_unemployment_rate = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪eu_unemployment_rate.csv', delimiter=';', header=0, names=['date',␣
 ↪'eu_unemployment_rate'])
# Convert the 'Date' column to datetime
eu_unemployment_rate['date'] = pd.to_datetime(eu_unemployment_rate['date'])
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(eu_unemployment_rate['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eu_unemployment_rate'] = np.nan
# Append the missing dates to the new timeseries
eu_unemployment_rate = pd.concat([eu_unemployment_rate,␣
 ↪missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
eu_unemployment_rate = eu_unemployment_rate.replace('.', np.nan)
# Make values float type
eu_unemployment_rate['eu_unemployment_rate'] =␣
 ↪eu_unemployment_rate['eu_unemployment_rate'].replace('.', np.nan).
 ↪astype(float)
# Fill NaN values using forward fill method
```

```
eu_unemployment_rate['eu_unemployment_rate'] =␣
 ↪eu_unemployment_rate['eu_unemployment_rate'].fillna(method='ffill')
# sort by date
eu_unemployment_rate = eu_unemployment_rate.sort_values(by='date')
# Filter the data to start from 01/01/2004
eu_unemployment_rate = eu_unemployment_rate[eu_unemployment_rate['date'] >=␣
 ↪'2005-01-01']
eu_unemployment_rate = eu_unemployment_rate[eu_unemployment_rate['date'] <=␣
 ↪'2024-01-01']
#length
print(len(eu_unemployment_rate['eu_unemployment_rate']))
```

[ ]:

```
#### EU Yield 3M ####
eu_yield_3m = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/eu_yield_3m.
 ↪csv', delimiter=';', header=0, names=['date', 'eu_yield_3m'])
# Convert the 'Date' column to datetime
eu_yield_3m['date'] = pd.to_datetime(eu_yield_3m['date'])
# remove missing values
eu_yield_3m = eu_yield_3m.replace('.', np.nan)
#remove dates that are missing from exchange rate
eu_yield_3m = eu_yield_3m[eu_yield_3m['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(eu_yield_3m['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eu_yield_3m'] = np.nan
# Append the missing dates to the new timeseries
eu_yield_3m = pd.concat([eu_yield_3m, missing_dates_timeseries]).
 ↪sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
eu_yield_3m = eu_yield_3m.replace('.', np.nan)
# Make values float type
eu_yield_3m['eu_yield_3m'] = eu_yield_3m['eu_yield_3m'].replace('.', np.nan).
 ↪astype(float)
# sort by date
eu_yield_3m = eu_yield_3m.sort_values(by='date')
# Interpolate missing values
eu_yield_3m['eu_yield_3m'] = eu_yield_3m['eu_yield_3m'].
 ↪interpolate(method='linear')
# Filter the data to start from 01/01/2004
eu_yield_3m = eu_yield_3m[eu_yield_3m['date'] >= '2005-01-01']
eu_yield_3m = eu_yield_3m[eu_yield_3m['date'] <= '2024-01-01']
#length
print(len(eu_yield_3m['eu_yield_3m']))
```

[ ]:

```
#### EU Yield 10Y ####
eu_yield_10y = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪eu_yield_10y.csv', delimiter=';', header=0, names=['date', 'eu_yield_10y'])
# Convert the 'Date' column to datetime
eu_yield_10y['date'] = pd.to_datetime(eu_yield_10y['date'])
# remove missing values
eu_yield_10y = eu_yield_10y.replace('.', np.nan)
#remove dates that are missing from exchange rate
eu_yield_10y = eu_yield_10y[eu_yield_10y['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(eu_yield_10y['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eu_yield_10y'] = np.nan
# Append the missing dates to the new timeseries
eu_yield_10y = pd.concat([eu_yield_10y, missing_dates_timeseries]).
 ↪sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
eu_yield_10y = eu_yield_10y.replace('.', np.nan)
# Make values float type
eu_yield_10y['eu_yield_10y'] = eu_yield_10y['eu_yield_10y'].replace('.', np.
 ↪nan).astype(float)
# sort by date
eu_yield_10y = eu_yield_10y.sort_values(by='date')
# Interpolate missing values
eu_yield_10y['eu_yield_10y'] = eu_yield_10y['eu_yield_10y'].
 ↪interpolate(method='linear')
# Filter the data to start from 01/01/2004
eu_yield_10y = eu_yield_10y[eu_yield_10y['date'] >= '2005-01-01']
eu_yield_10y = eu_yield_10y[eu_yield_10y['date'] <= '2024-01-01']
#length
print(len(eu_yield_10y['eu_yield_10y']))
```

[ ]:

```
#### EU Yield 30Y ####
eu_yield_30y = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪eu_yield_30y.csv', delimiter=';', header=0, names=['date', 'eu_yield_30y'])
# Convert the 'Date' column to datetime
eu_yield_30y['date'] = pd.to_datetime(eu_yield_30y['date'])
# remove missing values
eu_yield_30y = eu_yield_30y.replace('.', np.nan)
#remove dates that are missing from exchange rate
eu_yield_30y = eu_yield_30y[eu_yield_30y['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(eu_yield_30y['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eu_yield_30y'] = np.nan
```

```
# Append the missing dates to the new timeseries
eu_yield_30y = pd.concat([eu_yield_30y, missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
eu_yield_30y = eu_yield_30y.replace('.', np.nan)
# Make values float type
eu_yield_30y['eu_yield_30y'] = eu_yield_30y['eu_yield_30y'].replace('.', np.nan).astype(float)
# sort by date
eu_yield_30y = eu_yield_30y.sort_values(by='date')
# Interpolate missing values
eu_yield_30y['eu_yield_30y'] = eu_yield_30y['eu_yield_30y'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
eu_yield_30y = eu_yield_30y[eu_yield_30y['date'] >= '2005-01-01']
eu_yield_30y = eu_yield_30y[eu_yield_30y['date'] <= '2024-01-01']
#length
print(len(eu_yield_30y['eu_yield_30y']))
```

[ ]:

```
#### EUR Stoxx VIX ####
eur_stoxx_vix = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/eur_stoxx_vix.csv', delimiter=';', header=0, names=['date','ticker', 'eur_stoxx_vix'])
# Remove column for 'ticker'
eur_stoxx_vix.drop(columns=['ticker'], inplace=True)
# Convert the 'Date' column to datetime
eur_stoxx_vix['date'] = pd.to_datetime(eur_stoxx_vix['date'], dayfirst= True )
# remove missing values
eur_stoxx_vix = eur_stoxx_vix.replace('.', np.nan)
#remove dates that are missing from exchange rate
eur_stoxx_vix = eur_stoxx_vix[eur_stoxx_vix['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].isin(eur_stoxx_vix['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eur_stoxx_vix'] = np.nan
# Append the missing dates to the new timeseries
eur_stoxx_vix = pd.concat([eur_stoxx_vix, missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
eur_stoxx_vix = eur_stoxx_vix.replace('.', np.nan)
# Make values float type
eur_stoxx_vix['eur_stoxx_vix'] = eur_stoxx_vix['eur_stoxx_vix'].replace('.', np.nan).astype(float)
# sort by date
```

```
eur_stoxx_vix = eur_stoxx_vix.sort_values(by='date')
# Interpolate missing values
eur_stoxx_vix['eur_stoxx_vix'] = eur_stoxx_vix['eur_stoxx_vix'].
 ↪interpolate(method='linear')
# Filter the data to start from 01/01/2004
eur_stoxx_vix = eur_stoxx_vix[eur_stoxx_vix['date'] >= '2005-01-01']
eur_stoxx_vix = eur_stoxx_vix[eur_stoxx_vix['date'] <= '2024-01-01']
#length
print(len(eur_stoxx_vix['eur_stoxx_vix']))
```

```
[ ]: #### EUR Stoxx ####
eur_stoxx = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/eur_stoxx.
 ↪csv', header=0, names=['date', 'eur_stoxx'])
# Convert the 'Date' column to datetime
eur_stoxx['date'] = pd.to_datetime(eur_stoxx['date'].apply(lambda x: x[:10]),␣
 ↪dayfirst= True)
# Converting 'value' column to numeric value
eur_stoxx['eur_stoxx'] = eur_stoxx['eur_stoxx'].apply(lambda x: float(x.
 ↪replace(',', '.')))
# remove missing values
eur_stoxx = eur_stoxx.replace('.', np.nan)
#remove dates that are missing from exchange rate
eur_stoxx = eur_stoxx[eur_stoxx['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(eur_stoxx['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['eur_stoxx'] = np.nan
# Append the missing dates to the new timeseries
eur_stoxx = pd.concat([eur_stoxx, missing_dates_timeseries]).
 ↪sort_values(by='date').reset_index(drop=True)
# Make values float type
eur_stoxx['eur_stoxx'] = eur_stoxx['eur_stoxx'].replace('.', np.nan).
 ↪astype(float)
# replace '.' to 'nan'
eur_stoxx = eur_stoxx.replace('.', np.nan)
# sort by date
eur_stoxx = eur_stoxx.sort_values(by='date')
# Interpolate missing values
eur_stoxx['eur_stoxx'] = eur_stoxx['eur_stoxx'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
eur_stoxx = eur_stoxx[eur_stoxx['date'] >= '2005-01-01']
eur_stoxx = eur_stoxx[eur_stoxx['date'] <= '2024-01-01']
#length
print(len(eur_stoxx['eur_stoxx']))
```

[ ]:

```
#### US CPI ####
us_cpi = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/us_cpi.csv',␣
  ↪delimiter=';', header=0, names=['date', 'us_cpi', '1', '2', '3'])
# Remove columns for '1', '2', '3'
us_cpi.drop(columns=['1'], inplace=True)
us_cpi.drop(columns=['2'], inplace=True)
us_cpi.drop(columns=['3'], inplace=True)
# Convert the 'Date' column to datetime
us_cpi['date'] = pd.to_datetime(us_cpi['date'])
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].isin(us_cpi['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_cpi'] = np.nan
# Append the missing dates to the new timeseries
us_cpi = pd.concat([us_cpi, missing_dates_timeseries]).sort_values(by='date').
  ↪reset_index(drop=True)
# replace '.' to 'nan'
us_cpi = us_cpi.replace('.', np.nan)
# Make values float type
us_cpi['us_cpi'] = us_cpi['us_cpi'].replace('.', np.nan).astype(float)
# Fill NaN values using forward fill method
us_cpi['us_cpi'] = us_cpi['us_cpi'].fillna(method='ffill')
# sort by date
us_cpi = us_cpi.sort_values(by='date')
# Filter the data to start from 01/01/2004
us_cpi = us_cpi[us_cpi['date'] >= '2005-01-01']
us_cpi = us_cpi[us_cpi['date'] <= '2024-01-01']
#length
print(len(us_cpi['us_cpi']))
```

[ ]:

```
#### US Federal Fund Rate ####
us_federal_fund_rate = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
  ↪us_federal_funds_rate_filtered.csv', header=0, names=['date',␣
  ↪'us_federal_fund_rate'])
# Convert the 'Date' column to datetime
us_federal_fund_rate['date'] = pd.to_datetime(us_federal_fund_rate['date'])
# remove missing values
us_federal_fund_rate = us_federal_fund_rate.replace('.', np.nan)
#remove dates that are missing from exchange rate
us_federal_fund_rate = us_federal_fund_rate[us_federal_fund_rate['date'].
  ↪isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
  ↪isin(us_federal_fund_rate['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_federal_fund_rate'] = np.nan
# Append the missing dates to the new timeseries
```

```
us_federal_fund_rate = pd.concat([us_federal_fund_rate,␣
 ↪missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_federal_fund_rate = us_federal_fund_rate.replace('.', np.nan)
# Make values float type
us_federal_fund_rate['us_federal_fund_rate'] =␣
 ↪us_federal_fund_rate['us_federal_fund_rate'].replace('.', np.nan).
 ↪astype(float)
# sort by date
us_federal_fund_rate = us_federal_fund_rate.sort_values(by='date')
# Interpolate missing values
us_federal_fund_rate['us_federal_fund_rate'] =␣
 ↪us_federal_fund_rate['us_federal_fund_rate'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
us_federal_fund_rate = us_federal_fund_rate[us_federal_fund_rate['date'] >=␣
 ↪'2005-01-01']
us_federal_fund_rate = us_federal_fund_rate[us_federal_fund_rate['date'] <=␣
 ↪'2024-01-01']
#length
print(len(us_federal_fund_rate['us_federal_fund_rate']))
```

[ ]:

```
#### US S&P 500 ####
us_sp500 = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪us_sp500_filtered.csv', header=0, names=['date','1', '2','3', 'us_sp500',␣
 ↪'4'])
# Remove columns for '1', '2', '3' and '4'
us_sp500.drop(columns=['1'], inplace=True)
us_sp500.drop(columns=['2'], inplace=True)
us_sp500.drop(columns=['3'], inplace=True)
us_sp500.drop(columns=['4'], inplace=True)
# Convert the 'Date' column to datetime
us_sp500['date'] = pd.to_datetime(us_sp500['date'])
# remove missing values
us_sp500 = us_sp500.replace('.', np.nan)
#remove dates that are missing from exchange rate
us_sp500 = us_sp500[us_sp500['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(us_sp500['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_sp500'] = np.nan
# Append the missing dates to the new timeseries
us_sp500 = pd.concat([us_sp500, missing_dates_timeseries]).
 ↪sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_sp500 = us_sp500.replace('.', np.nan)
```

```
# Make values float type
us_sp500['us_sp500'] = us_sp500['us_sp500'].replace('.', np.nan).astype(float)
# sort by date
us_sp500 = us_sp500.sort_values(by='date')
# Interpolate missing values
us_sp500['us_sp500'] = us_sp500['us_sp500'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
us_sp500 = us_sp500[us_sp500['date'] >= '2005-01-01']
us_sp500 = us_sp500[us_sp500['date'] <= '2024-01-01']
#length
print(len(us_sp500['us_sp500']))
```

[ ]:

```
#### US S&P 500 VIX ####
us_sp500_vix = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
  ↪us_sp_500_vix.csv', header=0, names=['date','1', '2','3', 'us_sp500_vix'],␣
  ↪na_values='.')
# Remove columns for '1', '2', '3' and '4'
us_sp500_vix.drop(columns=['1'], inplace=True)
us_sp500_vix.drop(columns=['2'], inplace=True)
us_sp500_vix.drop(columns=['3'], inplace=True)
# Convert the 'Date' column to datetime
us_sp500_vix['date'] = pd.to_datetime(us_sp500_vix['date'])
# remove missing values
us_sp500_vix = us_sp500_vix.replace('.', np.nan)
#remove dates that are missing from exchange rate
us_sp500_vix = us_sp500_vix[us_sp500_vix['date'].isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
  ↪isin(us_sp500_vix['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_sp500_vix'] = np.nan
# Append the missing dates to the new timeseries
us_sp500_vix = pd.concat([us_sp500_vix, missing_dates_timeseries]).
  ↪sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_sp500_vix = us_sp500_vix.replace('.', np.nan)
# Make values float type
us_sp500_vix['us_sp500_vix'] = us_sp500_vix['us_sp500_vix'].replace('.', np.
  ↪nan).astype(float)
# sort by date
us_sp500_vix = us_sp500_vix.sort_values(by='date')
# Interpolate missing values
us_sp500_vix['us_sp500_vix'] = us_sp500_vix['us_sp500_vix'].
  ↪interpolate(method='linear')
# Filter the data to start from 01/01/2004
us_sp500_vix = us_sp500_vix[us_sp500_vix['date'] >= '2005-01-01']
us_sp500_vix = us_sp500_vix[us_sp500_vix['date'] <= '2024-01-01']
```

```
#length
print(len(us_sp500_vix['us_sp500_vix']))
```

[ ]:

```
#### US Treasury Yield 3M ####
us_treasury_yield_3m = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪us_treasury_yield_3m_filtered.csv', header=0, names=['date',␣
 ↪'us_treasury_yield_3m'])
# Convert the 'Date' column to datetime
us_treasury_yield_3m['date'] = pd.to_datetime(us_treasury_yield_3m['date'])
# remove missing values
us_treasury_yield_3m = us_treasury_yield_3m.replace('.', np.nan)
#remove dates that are missing from exchange rate
us_treasury_yield_3m = us_treasury_yield_3m[us_treasury_yield_3m['date'].
 ↪isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(us_treasury_yield_3m['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_treasury_yield_3m'] = np.nan
# Append the missing dates to the new timeseries
us_treasury_yield_3m = pd.concat([us_treasury_yield_3m,␣
 ↪missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_treasury_yield_3m = us_treasury_yield_3m.replace('.', np.nan)
# Make values float type
us_treasury_yield_3m['us_treasury_yield_3m'] =␣
 ↪us_treasury_yield_3m['us_treasury_yield_3m'].replace('.', np.nan).
 ↪astype(float)
# sort by date
us_treasury_yield_3m = us_treasury_yield_3m.sort_values(by='date')
# Interpolate missing values
us_treasury_yield_3m['us_treasury_yield_3m'] =␣
 ↪us_treasury_yield_3m['us_treasury_yield_3m'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
us_treasury_yield_3m = us_treasury_yield_3m[us_treasury_yield_3m['date'] >=␣
 ↪'2005-01-01']
us_treasury_yield_3m = us_treasury_yield_3m[us_treasury_yield_3m['date'] <=␣
 ↪'2024-01-01']
#length
print(len(us_treasury_yield_3m['us_treasury_yield_3m']))
```

[ ]:

```
#### US Treasury Yield 10Y ####
us_treasury_yield_10y = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪us_treasury_yield_10y_filtered.csv', header=0, names=['date',␣
 ↪'us_treasury_yield_10y'])
# Convert the 'Date' column to datetime
```

```
us_treasury_yield_10y['date'] = pd.to_datetime(us_treasury_yield_10y['date'])
# remove missing values
us_treasury_yield_10y = us_treasury_yield_10y.replace('.', np.nan)
#remove dates that are missing from exchange rate
us_treasury_yield_10y = us_treasury_yield_10y[us_treasury_yield_10y['date'].
  ↪isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
  ↪isin(us_treasury_yield_10y['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_treasury_yield_10y'] = np.nan
# Append the missing dates to the new timeseries
us_treasury_yield_10y = pd.concat([us_treasury_yield_10y,␣
  ↪missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_treasury_yield_10y = us_treasury_yield_10y.replace('.', np.nan)
# Make values float type
us_treasury_yield_10y['us_treasury_yield_10y'] =␣
  ↪us_treasury_yield_10y['us_treasury_yield_10y'].replace('.', np.nan).
  ↪astype(float)
# sort by date
us_treasury_yield_10y = us_treasury_yield_10y.sort_values(by='date')
# Interpolate missing values
us_treasury_yield_10y['us_treasury_yield_10y'] =␣
  ↪us_treasury_yield_10y['us_treasury_yield_10y'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
us_treasury_yield_10y = us_treasury_yield_10y[us_treasury_yield_10y['date'] >=␣
  ↪'2005-01-01']
us_treasury_yield_10y = us_treasury_yield_10y[us_treasury_yield_10y['date'] <=␣
  ↪'2024-01-01']
#length
print(len(us_treasury_yield_10y['us_treasury_yield_10y']))
```

[ ]:

```
#### US Treasury Yield 30Y ####
us_treasury_yield_30y = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
  ↪us_treasury_yield_30y_filtered.csv', header=0, names=['date',␣
  ↪'us_treasury_yield_30y'])
# Convert the 'Date' column to datetime
us_treasury_yield_30y['date'] = pd.to_datetime(us_treasury_yield_30y['date'])
# remove missing values
us_treasury_yield_30y = us_treasury_yield_30y.replace('.', np.nan)
#remove dates that are missing from exchange rate
us_treasury_yield_30y = us_treasury_yield_30y[us_treasury_yield_30y['date'].
  ↪isin(usd_eur_exchange['date'])]
#insert dates from exchange rate that are missing in timeseries
```

```
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(us_treasury_yield_30y['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_treasury_yield_30y'] = np.nan
# Append the missing dates to the new timeseries
us_treasury_yield_30y = pd.concat([us_treasury_yield_30y,␣
 ↪missing_dates_timeseries]).sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_treasury_yield_30y = us_treasury_yield_30y.replace('.', np.nan)
# Make values float type
us_treasury_yield_30y['us_treasury_yield_30y'] =␣
 ↪us_treasury_yield_30y['us_treasury_yield_30y'].replace('.', np.nan).
 ↪astype(float)
# sort by date
us_treasury_yield_30y = us_treasury_yield_30y.sort_values(by='date')
# Interpolate missing values
us_treasury_yield_30y['us_treasury_yield_30y'] =␣
 ↪us_treasury_yield_30y['us_treasury_yield_30y'].interpolate(method='linear')
# Filter the data to start from 01/01/2004
us_treasury_yield_30y = us_treasury_yield_30y[us_treasury_yield_30y['date'] >=␣
 ↪'2005-01-01']
us_treasury_yield_30y = us_treasury_yield_30y[us_treasury_yield_30y['date'] <=␣
 ↪'2024-01-01']
#length
print(len(us_treasury_yield_30y['us_treasury_yield_30y']))
```

[ ]:

```
#### US Unemployment ####
us_unemployment = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/
 ↪us_unemployment_rate.csv', delimiter=';', header=0, names=['date',␣
 ↪'us_unemployment'])
# Convert the 'Date' column to datetime
us_unemployment['date'] = pd.to_datetime(us_unemployment['date'])
#insert dates from exchange rate that are missing in timeseries
missing_dates = usd_eur_exchange[~usd_eur_exchange['date'].
 ↪isin(us_unemployment['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
missing_dates_timeseries['us_unemployment'] = np.nan
# Append the missing dates to the new timeseries
us_unemployment = pd.concat([us_unemployment, missing_dates_timeseries]).
 ↪sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
us_unemployment = us_unemployment.replace('.', np.nan)
# Make values float type
us_unemployment['us_unemployment'] = us_unemployment['us_unemployment'].
 ↪replace('.', np.nan).astype(float)
# Fill NaN values using forward fill method
```

```
us_unemployment['us_unemployment'] = us_unemployment['us_unemployment'].
 ↪fillna(method='ffill')
# sort by date
us_unemployment = us_unemployment.sort_values(by='date')
# Filter the data to start from 01/01/2004
us_unemployment = us_unemployment[us_unemployment['date'] >= '2005-01-01']
us_unemployment = us_unemployment[us_unemployment['date'] <= '2024-01-01']
#length
print(len(us_unemployment['us_unemployment']))
```

[ ]:

```
# List of dataframes
dfs = [usd_eur_exchange, brent, eu_cpi, eu_mro_rate, eu_unemployment_rate,␣
 ↪eu_yield_3m, eu_yield_10y, eu_yield_30y, eur_stoxx_vix, eur_stoxx, us_cpi,␣
 ↪us_federal_fund_rate, us_sp500, us_sp500_vix, us_treasury_yield_3m,␣
 ↪us_treasury_yield_10y, us_treasury_yield_30y, us_unemployment]
# Merge all dataframes on the 'date' column in one command
macrovariables = reduce(lambda left, right: pd.merge(left, right, on='date',␣
 ↪how='outer'), dfs)

macrovariables.to_csv('/Users/nadjos/Desktop/Uni/speciale/macrovariables.csv',␣
 ↪index=False)
```