

# technical\_indicators

June 2, 2024

```
[ ]: import pandas as pd
import ta
import numpy as np
import pandas as pd

[ ]: # load csv and name value variable
df = pd.read_csv('/Users/nadjos/Desktop/Uni/speciale/data/usd_eur_exchange.
↳csv', header=0, names=['date', 'usd_eur_exchange'])

df['date'] = pd.to_datetime(df['date'])
df = df.sort_values(by='date')
# remove missing values
df = df.replace('.', np.nan).dropna()
# Filter the data to start from 01/01/2004
df = df[df['date'] >= '2004-01-01']
df = df[df['date'] <= '2024-01-01']
df['usd_eur_exchange'] = pd.to_numeric(df['usd_eur_exchange'], errors='coerce')

[ ]: ##### making open, highs and lows price #####

# load csv and name value variable
usd_eur_finance = pd.read_csv('/Users/nadjos/Downloads/usd_eur_exchange_rate.
↳csv', header=0, names=['date', 'open', 'high', 'low', 'close', 'adj_close', 'v
↳olume' ])
# remove last 2 rows
usd_eur_finance = usd_eur_finance.iloc[:, :5]
usd_eur_finance['date'] = pd.to_datetime(usd_eur_finance['date'])
usd_eur_finance = usd_eur_finance.sort_values(by='date')
# remove missing values
usd_eur_finance = usd_eur_finance.replace('.', np.nan).dropna()
# Filter the data to start from 01/01/2004
usd_eur_finance = usd_eur_finance[usd_eur_finance['date'] >= '2004-01-01']
usd_eur_finance = usd_eur_finance[usd_eur_finance['date'] <= '2024-01-01']

[ ]: # Margin the to timeseries
combined_usd_eur = usd_eur_finance
# Sort
```

```
combined_usd_eur = combined_usd_eur.sort_values(by='date').
↳reset_index(drop=True)
```

```
[ ]: ##### Adjusting variable to usd_eur from FRED
combined_usd_eur = combined_usd_eur.fillna(method='ffill')
# remove missing values
combined_usd_eur = combined_usd_eur.replace('.', np.nan)
#remove dates that are missing from exchange rate
combined_usd_eur = combined_usd_eur[combined_usd_eur['date'].isin(df['date'])]
#insert dates from exchange rate that are missing in timeseries
missing_dates = df[~df['date'].isin(combined_usd_eur['date'])]
missing_dates_timeseries = missing_dates[['date']].copy()
# Append the missing dates to the new timeseries
combined_usd_eur = pd.concat([combined_usd_eur, missing_dates_timeseries]).
↳sort_values(by='date').reset_index(drop=True)
# replace '.' to 'nan'
combined_usd_eur = combined_usd_eur.replace('.', np.nan)
# sort by date
combined_usd_eur = combined_usd_eur.sort_values(by='date')
##### Adjusting variable to usd_eur from FRED
combined_usd_eur = combined_usd_eur.fillna(method='ffill')
```

```
[ ]: ##### Lagging indicators #####

# Simple Moving Averages, 10, 20.
df['SMA_10'] = ta.trend.sma_indicator(df['usd_eur_exchange'], window=10)
df['SMA_20'] = ta.trend.sma_indicator(df['usd_eur_exchange'], window=20)

# MACD, 12 and 26
df['MACD_12'] = ta.trend.macd(df['usd_eur_exchange'], window_slow=26,↳
↳window_fast=12)
```

```
[ ]: ##### leading indicators #####

# Calculate Parabolic SAR
psar_indicator = ta.trend.PSARIndicator(combined_usd_eur['high'],↳
↳combined_usd_eur['low'], df['usd_eur_exchange'])
df['Parabolic_SAR'] = psar_indicator.psar()

# Relative Strength Index
df['RSI'] = ta.momentum.rsi(df['usd_eur_exchange'], window=10)

# Rate of Change (ROC)
df['ROC'] = ta.momentum.roc(df['usd_eur_exchange'], window=2)

# Momentum indicator
```

```
momentum_indicator = ta.momentum.roc(df['usd_eur_exchange'], window=4) # ROC
↳stands for Rate of Change, similar to momentum
# Add to dataframe
df['momentum'] = momentum_indicator
```

```
[ ]: ##### volatility indicators #####

# Bollinger Bands
bb_indicator = ta.volatility.BollingerBands(df['usd_eur_exchange'], window=20,
↳window_dev=2)
df['BB_High'] = bb_indicator.bollinger_hband()
df['BB_Low'] = bb_indicator.bollinger_lband()
df['BB_Middle'] = bb_indicator.bollinger_mavg()

# Filter the data to start from 01/01/2005
df = df[df['date'] >= '2005-01-01']

technical_indicators = df

technical_indicators['usd_eur_exchange'] =
↳technical_indicators['usd_eur_exchange'].diff()
technical_indicators['usd_eur_exchange'] =
↳technical_indicators['usd_eur_exchange'].shift(-1) # Shift by -1 period
# cut off last row
technical_indicators = technical_indicators.iloc[:-1]
```

```
[ ]: technical_indicators.to_csv('/Users/nadjos/Desktop/Uni/speciale/
↳technical_indicators.csv', index=False)
```