In [ ]:

```
###necessary libaries###
import numpy as np
import pandas as pd
import seaborn as sns
import glob
import os
from datetime import datetime
import matplotlib.pyplot as plt
import re

# file where csv files lies
path = r'C:\Users\victo\Master_Thesis\merging_data\porsche\hourly\merged_files'
all_files = glob.glob(os.path.join(path, "*.csv"))

# read files to pandas frame
list_of_files = []

for filename in all_files:
    list_of_files.append(pd.read_csv(filename,
                                     sep=',',
                                     )
                         )

# Concatenate all content of files into one DataFrames
concatenate_dataframe = pd.concat(list_of_files,
                                  ignore_index=True,
                                  axis=0,
                                  )

#print(concatenate_dataframe)

#calculating correlation price vs semantics
new_df_price = concatenate_dataframe[['return_one_hot_encoded',
                                      'flair_sentiment_header_score',
                                      'flair_sentiment_content_score',
                                      'compound_vader_header',
                                      'compound_vader_articel_content',
                                      'polarity_textblob_sentiment_header',
                                      'polarity_textblob_sentiment_content']]

new_df_price[['return_one_hot_encoded',
              'flair_sentiment_header_score',
              'flair_sentiment_content_score',
              'compound_vader_header',
              'compound_vader_articel_content',
              'polarity_textblob_sentiment_header',
              'polarity_textblob_sentiment_content']] = new_df_price[['return_one_h
ot_encoded',
                                                                      'flair_sentim
ent_header_score',
                                                                      'flair_sentim
ent_content_score',
                                                                      'compound_vad
er_header',
                                                                      'compound_vad
er_articel_content',
                                                                      'polarity_tex
tblob_sentiment_header',
                                                                      'polarity_tex
tblob_sentiment_content']].fillna(0)

print(new_df_price)
```

```
corr_price = new_df_price.corr()
corr_price.fillna(0)
print(corr_price)
corr_price.to_excel(r'C:\Users\victo\Master_Thesis\correlation\porsche\hourly\correlation\porsche_correlation_price_with_semantics.xlsx')

#calculating correlation volume vs semantics
new_df_volume = concatenate_dataframe[['volume_one_hot_encoded',
                                       'flair_sentiment_header_score',
                                       'flair_sentiment_content_score',
                                       'compound_vader_header',
                                       'compound_vader_articel_content',
                                       'polarity_textblob_sentiment_header',
                                       'polarity_textblob_sentiment_content']]

new_df_volume[['volume_one_hot_encoded',
               'flair_sentiment_header_score',
               'flair_sentiment_content_score',
               'compound_vader_header',
               'compound_vader_articel_content',
               'polarity_textblob_sentiment_header',
               'polarity_textblob_sentiment_content']] = new_df_volume[['volume_one_hot_encoded',
                                                                        'flair_sentiment_header_score',
                                                                        'flair_sentiment_content_score',
                                                                        'compound_vader_header',
                                                                        'compound_vader_articel_content',
                                                                        'polarity_textblob_sentiment_header',
                                                                        'polarity_textblob_sentiment_content']].fillna(0)

print(new_df_volume)
corr_volume = new_df_volume.corr()
corr_volume.fillna(0)
print(corr_volume)
corr_volume.to_excel(r'C:\Users\victo\Master_Thesis\correlation\porsche\hourly\correlation\porsche_correlation_volume_with_semantics.xlsx')
```