# Error Debugging Example

I thought this would be a good example of how to debug a not so obvious error. Note that at the time, this line wasn’t in the script:

sp\_500\_symbols\_list = [x.replace(".", "-") for x in sp\_500\_raw\_symbols\_list]

This error was in reference to tickers\_price\_history\_df2. However, since we were pulling in data for the S&P 500, we were expecting yahooquery to return a dictionary of dataframes. We had isolated the dataframe with this line:

tickers\_price\_history\_df2 = tickers\_price\_history\_df\_or\_dict[f"{ticker}"]

Traceback (most recent call last):

File "<stdin>", line 59, in <module>

AttributeError: 'dict' object has no attribute 'reset\_index'

So what could be the problem?

When printing close\_df1 it was able to pull in all stock data for all the tickers in sp\_500\_symbols\_list up to BEN.

>>> print(close\_df1)

A\_close AA\_close AAL\_close ... BDX\_close BEAM\_close BEN\_close

2015-06-24 40.189999 28.211220 42.240002 ... 141.699997 NaN 50.619999

2015-06-25 40.049999 27.802710 42.240002 ... 141.910004 NaN 50.310001

2015-06-26 40.020000 28.091070 41.439999 ... 142.039993 NaN 50.480000

2015-06-29 38.740002 27.274050 39.750000 ... 141.779999 NaN 48.889999

2015-06-30 38.580002 26.793449 39.939999 ... 141.649994 NaN 49.029999

... ... ... ... ... ... ... ...

2020-06-18 88.209999 11.650000 16.490000 ... 236.229996 26.600000 22.559999

2020-06-19 88.730003 11.480000 16.000000 ... 237.589996 27.299999 21.950001

2020-06-22 88.529999 11.830000 14.920000 ... 235.190002 28.660000 21.690001

2020-06-23 89.279999 12.200000 14.000000 ... 237.830002 26.360001 21.549999

2020-06-24 86.589996 11.175000 13.191800 ... 231.279999 26.244101 21.160000

The symbol after BEN in sp\_500\_symbols\_list is BF.B. This stock exists, but Yahoo Finance uses BF-B as the ticker symbol. Since we had been using BF.B, yahooquery was returning an empty dictionary. Hence the AttributeError: ‘dict’ object has no attribute ‘reset\_index’.

>>> print(tickers\_price\_history\_df\_or\_dict)

[1260 rows x 7 columns], 'BF.B': {'meta': {'currency': None, 'symbol': 'BF.B', 'exchangeName': 'YHD', 'instrumentType': 'MUTUALFUND', 'firstTradeDate': None, 'regularMarketTime': 1561759658, 'gmtoffset': -14400, 'timezone': 'EDT', 'exchangeTimezoneName': 'America/New\_York', 'priceHint': 2, 'currentTradingPeriod': {'pre': {'timezone': 'EDT', 'start': 1592985600, 'end': 1593005400, 'gmtoffset': -14400}, 'regular': {'timezone': 'EDT', 'start': 1593005400, 'end': 1593028800, 'gmtoffset': -14400}, 'post': {'timezone': 'EDT', 'start': 1593028800, 'end': 1593043200, 'gmtoffset': -14400}}, 'dataGranularity': '1d', 'range': '5y', 'validRanges': ['1mo', '3mo', '6mo', 'ytd', '1y', '2y', '5y', '10y', 'max']}, 'indicators': {'quote': [{}], 'adjclose': [{}]}}

The simple solution is to replace any periods with dashes in sp\_500\_symbols\_list using the following list comprehension:

sp\_500\_symbols\_list = [x.replace(".", "-") for x in sp\_500\_raw\_symbols\_list]