

```

-----
ValueError                                Traceback (most recent call last)
<ipython-input-7-e0447762367c> in <module>
----> 1 cv.income_statement('AAPL', period = 'quarter', ftype = 'full')

~\Anaconda3\lib\site-packages\pyfmpcloud\company_valuation.py in income_statement(ticker, period, ftype)
    74     response = urlopen(url)
    75     data = response.read().decode("utf-8")
--> 76     return safe_read_json(data)
    77
    78 def cash_flow_statement(ticker, period = 'annual', ftype = 'full'):

~\Anaconda3\lib\site-packages\pyfmpcloud\company_valuation.py in safe_read_json(data)
    345     raise Exception(data[20:-3])
    346     else:
--> 347     return pd.read_json(data)
    348

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    197     else:
    198         kwargs[new_arg_name] = new_arg_value
--> 199     return func(*args, **kwargs)
    200
    201     return cast(F, wrapper)

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    294     )
    295     warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 296     return func(*args, **kwargs)
    297
    298     return wrapper

~\Anaconda3\lib\site-packages\pandas\io\json\_json.py in read_json(path_or_buf, orient, typ, dtype, convert_axes, convert_dates, keep_default_dates, numpy, precise_float, date_unit, encoding, lines, chunksize, compression, nrows)
    591
    592     compression = infer_compression(path_or_buf, compression)
--> 593     filepath_or_buffer, _, compression, should_close = get_filepath_or_buffer(
    594         path_or_buf, encoding=encoding, compression=compression
    595     )

~\Anaconda3\lib\site-packages\pandas\io\common.py in get_filepath_or_buffer(filepath_or_buffer, encoding, compression, mode, storage_options)
    219

```

```

220         try:
--> 221             file_obj = fsspec.open(
222                 filepath_or_buffer, mode=mode or "rb", **(storage_options or {})
)
223             ).open()

```

```

~\Anaconda3\lib\site-packages\fsspec\core.py in open(urlpath, mode, compression, encoding, errors, protocol, newline, **kwargs)

```

```

427     ``OpenFile`` object.
428     """
--> 429     return open_files(
430         [urlpath],
431         mode,

```

```

~\Anaconda3\lib\site-packages\fsspec\core.py in open_files(urlpath, mode, compression, encoding, errors, name_function, num, protocol, newline, auto_mkdir, expand, **kwargs)

```

```

278     be used as a single context
279     """
--> 280     fs, fs_token, paths = get_fs_token_paths(
281         urlpath,
282         mode,

```

```

~\Anaconda3\lib\site-packages\fsspec\core.py in get_fs_token_paths(urlpath, mode, num, name_function, storage_options, protocol, expand)

```

```

598         "share the same protocol"
599     )
--> 600     cls = get_filesystem_class(protocol)
601     options = list(map(cls._get_kwargs_from_urls, urlpath))
602     paths = [cls._strip_protocol(u) for u in urlpath]

```

```

~\Anaconda3\lib\site-packages\fsspec\registry.py in get_filesystem_class(protocol)

```

```

189     if protocol not in registry:
190         if protocol not in known_implementations:
--> 191             raise ValueError("Protocol not known: %s" % protocol)
192         bit = known_implementations[protocol]
193         try:

```

```

ValueError: Protocol not known: [ {

```

```

    "date" : "2020-09-26",
    "symbol" : "AAPL",
    "fillingDate" : "2020-10-30",
    "acceptedDate" : "2020-10-29 18:06:25",
    "period" : "Q4",
    "revenue" : 64698000000,
    "costOfRevenue" : 40009000000,

```

"grossProfit" : 24689000000,
"grossProfitRatio" : 0.38160375900337,
"researchAndDevelopmentExpenses" : 4978000000,
"generalAndAdministrativeExpenses" : 4936000000,
"sellingAndMarketingExpenses" : 0.0,
"otherExpenses" : 126000000,
"operatingExpenses" : 9914000000,
"costAndExpenses" : 49923000000,
"interestExpense" : 0.0,
"depreciationAndAmortization" : 2702000000,
"ebitda" : 17477000000,
"ebitdaratio" : 0.270131997897926,
"operatingIncome" : 14775000000,
"operatingIncomeRatio" : 0.228368728554206,
"totalOtherIncomeExpensesNet" : 126000000,
"incomeBeforeTax" : 14901000000,
"incomeBeforeTaxRatio" : 0.230316238523602,
"incomeTaxExpense" : 2228000000,
"netIncome" : 12673000000,
"netIncomeRatio" : 0.195879316207611,
"eps" : 0.73,
"epsdiluted" : 0.73,
"weightedAverageShsOut" : 17352119000,
"weightedAverageShsOutDil" : 17528214000,
"link" : "https

In []: