Untitled1

November 19, 2024

[1]:
```
!pip install yfinance
#!pip install pandas
#!pip install requests
!pip install bs4
#!pip install plotly
```

```
Collecting yfinance
  Downloading yfinance-0.2.49-py2.py3-none-any.whl.metadata (13 kB)
Collecting pandas>=1.3.0 (from yfinance)
  Downloading
pandas-2.2.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(89 kB)
                           89.9/89.9 kB
10.4 MB/s eta 0:00:00
Collecting numpy>=1.16.5 (from yfinance)
  Downloading
numpy-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(62 kB)
                           62.0/62.0 kB
8.1 MB/s eta 0:00:00
Requirement already satisfied: requests>=2.31 in
/opt/conda/lib/python3.11/site-packages (from yfinance) (2.31.0)
Collecting multitasking>=0.0.7 (from yfinance)
  Downloading multitasking-0.0.11-py3-none-any.whl.metadata (5.5 kB)
Collecting lxml>=4.9.1 (from yfinance)
  Downloading lxml-5.3.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (3.8 kB)
Requirement already satisfied: platformdirs>=2.0.0 in
/opt/conda/lib/python3.11/site-packages (from yfinance) (4.2.1)
Requirement already satisfied: pytz>=2022.5 in /opt/conda/lib/python3.11/site-
packages (from yfinance) (2024.1)
Collecting frozendict>=2.3.4 (from yfinance)
  Downloading frozendict-2.4.6-py311-none-any.whl.metadata (23 kB)
Collecting peewee>=3.16.2 (from yfinance)
  Downloading peewee-3.17.8.tar.gz (948 kB)
                          948.2/948.2 kB
81.1 MB/s eta 0:00:00
  Installing build dependencies … done
  Getting requirements to build wheel … done
```

```
  Preparing metadata (pyproject.toml) … done
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/opt/conda/lib/python3.11/site-packages (from yfinance) (4.12.3)
Collecting html5lib>=1.1 (from yfinance)
  Downloading html5lib-1.1-py2.py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.11/site-
packages (from beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: six>=1.9 in /opt/conda/lib/python3.11/site-
packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /opt/conda/lib/python3.11/site-
packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/lib/python3.11/site-packages (from pandas>=1.3.0->yfinance) (2.9.0)
Collecting tzdata>=2022.7 (from pandas>=1.3.0->yfinance)
  Downloading tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.11/site-packages (from requests>=2.31->yfinance) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.11/site-
packages (from requests>=2.31->yfinance) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/opt/conda/lib/python3.11/site-packages (from requests>=2.31->yfinance) (2.2.1)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.11/site-packages (from requests>=2.31->yfinance)
(2024.6.2)
Downloading yfinance-0.2.49-py2.py3-none-any.whl (101 kB)
                         101.1/101.1 kB
9.2 MB/s eta 0:00:00
Downloading frozendict-2.4.6-py311-none-any.whl (16 kB)
Downloading html5lib-1.1-py2.py3-none-any.whl (112 kB)
                         112.2/112.2 kB
12.9 MB/s eta 0:00:00
Downloading lxml-5.3.0-cp311-cp311-manylinux_2_28_x86_64.whl (5.0 MB)
                         5.0/5.0 MB
125.3 MB/s eta 0:00:0000:01
Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Downloading
numpy-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.3 MB)
                         16.3/16.3 MB
115.5 MB/s eta 0:00:0000:010:01
Downloading
pandas-2.2.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.1
MB)
                         13.1/13.1 MB
109.3 MB/s eta 0:00:0000:010:01
Downloading tzdata-2024.2-py2.py3-none-any.whl (346 kB)
                         346.6/346.6 kB
32.9 MB/s eta 0:00:00
Building wheels for collected packages: peewee
```

```
  Building wheel for peewee (pyproject.toml) … done
  Created wheel for peewee: filename=peewee-3.17.8-py3-none-any.whl
size=138965
sha256=a9001eccf7b496ad3763a17515b1e55972850fce59ebc7dfa8273615e9dde878
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/ff/6c/15/506e25bc390de
450a7fa53c155cd9b0fbd13ad3e84a9abc183
Successfully built peewee
Installing collected packages: peewee, multitasking, tzdata, numpy, lxml,
html5lib, frozendict, pandas, yfinance
Successfully installed frozendict-2.4.6 html5lib-1.1 lxml-5.3.0
multitasking-0.0.11 numpy-2.1.3 pandas-2.2.3 peewee-3.17.8 tzdata-2024.2
yfinance-0.2.49
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Requirement already satisfied: beautifulsoup4 in /opt/conda/lib/python3.11/site-
packages (from bs4) (4.12.3)
Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.11/site-
packages (from beautifulsoup4->bs4) (2.5)
Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
```

[2]:
```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

[3]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True,␣
 ↪subplot_titles=("Historical Share Price", "Historical Revenue"),␣
 ↪vertical_spacing = .3)
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date,␣
 ↪infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Share␣
 ↪Price"), row=1, col=1)
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date,␣
 ↪infer_datetime_format=True), y=revenue_data.Revenue.astype("float"),␣
 ↪name="Revenue"), row=2, col=1)
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeslider_visible=True)
```

```
    fig.show()
```

[4]:
```
tesla = yf.Ticker('TSLA')
```

[5]:
```
tesla_data = tesla.history(period="max")
```

[6]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head(5)
```

[6]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0

[7]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data = requests.get(url).text
```

[8]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<html>
 <head>
 </head>
 <body>
  <div style="margin: 50px auto; width: 50%; border: 1px solid #dfdfdf; padding:
20px 50px 30px 50px; font-family:helvetica;">
   <h1>
    We do not allow automated access to our servers.
   </h1>
   <h2>
    <p>
     Automated access to our data is prohibited by our data provider.
    </p>
    <p>
     If you are a user attempting to access the site via a browser, please
follow this process to regain access:
    </p>
    <ul>
```

```
     <li>
      Go to
      <a href="https://whatismyipaddress.com/" rel="noopener noreferrer"
target="_blank">
       whatismyipaddress
      </a>
      and obtain your IPv4 address
     </li>
     <li>
      Email us your IPv4 address at
      <a class="__cf_email__" data-
cfemail="4821262e270825292b3a273c3a2d262c3b66262d3c" href="/cdn-cgi/l/email-
protection">
       [email protected]
      </a>
     </li>
     <li>
      We will add you to our whitelist within 24 hours
     </li>
    </ul>
   </h2>
  </div>
  <script data-cfasync="false">
   !function(){"use strict";function e(e){try{if("undefined"==typeof
console)return;"error"in
console?console.error(e):console.log(e)}catch(e){}}function t(e,t){var
r=e.substr(t,2);return parseInt(r,16)}function r(r,n){for(var
c="",o=t(r,n),a=n+2;a<r.length;a+=2){var l=t(r,a)^o;c+=String.fromCharCode(l)}tr
y{c=decodeURIComponent(escape(c))}catch(t){e(t)}return function(e){return
i.innerHTML='<a href="'+e.replace(/"/g,"&quot;")+'"></a>',i.childNodes[0].getAtt
ribute("href")||""}(c)}function n(t){try{(function(t){for(var
n=t.querySelectorAll("a"),o=0;o<n.length;o++)try{var a=n[o],i=a.href.indexOf(c);
i>-
1&&(a.href="mailto:"+r(a.href,i+c.length))}catch(t){e(t)}})(t),function(t){for(v
ar n=t.querySelectorAll(o),c=0;c<n.length;c++)try{var
i=n[c],l=i.parentNode,u=i.getAttribute(a);if(u){var f=r(u,0),d=document.createTe
xtNode(f);l.replaceChild(d,i)}}catch(t){e(t)}}(t),function(t){for(var r=t.queryS
electorAll("template"),c=0;c<r.length;c++)try{n(r[c].content)}catch(t){e(t)}}(t)
}catch(t){e(t)}}var c="/cdn-cgi/l/email-protection#",o=".__cf_email__",a="data-
cfemail",i=document.createElement("div");n(document),function(){var e=document.c
urrentScript||document.scripts[document.scripts.length-
1];e.parentNode.removeChild(e)}()}();
  </script>
  <script>
   (function(){function c(){var
b=a.contentDocument||a.contentWindow.document;if(b){var d=b.createElement('scrip
t');d.innerHTML="window.__CF$cv$params={r:'8e506d3e4ed45aff',t:'MTczMjAyMTY4Mi4w
MDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/cdn-
```

```
cgi/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head'
)[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(docum
ent.body){var a=document.createElement('iframe');a.height=1;a.width=1;a.style.po
sition='absolute';a.style.top=0;a.style.left=0;a.style.border='none';a.style.vis
ibility='hidden';document.body.appendChild(a);if('loading'!==document.readyState
)c();else if(window.addEventListener)document.addEventListener('DOMContentLoaded
',c);else{var e=document.onreadystatechange||function(){};document.onreadystatec
hange=function(b){e(b);'loading'!==document.readyState&&(document.onreadystatech
ange=e,c())}}})();
  </script>
 </body>
</html>
```

[9]:
```
tesla_revenue = pd.DataFrame(columns =["Date","Revenue"])

for table in soup.find_all('table'):
    if table.find('th').getText().startswith("Tesla Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            if len(col) != 2: continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

            tesla_revenue = tesla_revenue.append({"Date":Date, "Revenue":
 ↪Revenue}, ignore_index=True)
```

[20]:
```
tesla_revenue.dropna(axis=0, how='all', subset=['Revenue']) #drop NaN values
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""] #drop empty␣
 ↪string values
```

[11]:
```
tesla_revenue.tail(5)
```

[11]:
```
Empty DataFrame
Columns: [Date, Revenue]
Index: []
```

[12]:
```
gme = yf.Ticker('GME')
```

[13]:
```
gme_data = gme.history(period = "max")
```

[14]:
```
gme_data.reset_index(inplace=True)
gme_data.head(5)
```

[14]:

	Date	Open	High	Low	Close	Volume
0	2002-02-13 00:00:00-05:00	1.620129	1.693350	1.603296	1.691667	76216000
1	2002-02-14 00:00:00-05:00	1.712707	1.716074	1.670626	1.683250	11021600

```
2 2002-02-15 00:00:00-05:00  1.683250  1.687458  1.658002  1.674834   8389600
3 2002-02-19 00:00:00-05:00  1.666418  1.666418  1.578047  1.607504   7410400
4 2002-02-20 00:00:00-05:00  1.615920  1.662210  1.603296  1.662210   6892800

   Dividends  Stock Splits
0        0.0           0.0
1        0.0           0.0
2        0.0           0.0
3        0.0           0.0
4        0.0           0.0
```

[22]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url).text
```

[23]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<html>
 <head>
 </head>
 <body>
  <div style="margin: 50px auto; width: 50%; border: 1px solid #dfdfdf; padding:
20px 50px 30px 50px; font-family:helvetica;">
   <h1>
    We do not allow automated access to our servers.
   </h1>
   <h2>
    <p>
     Automated access to our data is prohibited by our data provider.
    </p>
    <p>
     If you are a user attempting to access the site via a browser, please
follow this process to regain access:
    </p>
    <ul>
     <li>
      Go to
      <a href="https://whatismyipaddress.com/" rel="noopener noreferrer"
target="_blank">
       whatismyipaddress
      </a>
      and obtain your IPv4 address
     </li>
     <li>
      Email us your IPv4 address at
      <a class="__cf_email__" data-
cfemail="4821262e270825292b3a273c3a2d262c3b66262d3c" href="/cdn-cgi/l/email-
```

```
protection">
       [email protected]
      </a>
     </li>
     <li>
      We will add you to our whitelist within 24 hours
     </li>
    </ul>
   </h2>
  </div>
  <script data-cfasync="false">
   !function(){"use strict";function e(e){try{if("undefined"==typeof
console)return;"error"in
console?console.error(e):console.log(e)}catch(e){}}function t(e,t){var
r=e.substr(t,2);return parseInt(r,16)}function r(r,n){for(var
c="",o=t(r,n),a=n+2;a<r.length;a+=2){var l=t(r,a)^o;c+=String.fromCharCode(l)}tr
y{c=decodeURIComponent(escape(c))}catch(t){e(t)}return function(e){return
i.innerHTML='<a href="'+e.replace(/"/g,"&quot;")+'"></a>',i.childNodes[0].getAtt
ribute("href")||""}(c)}function n(t){try{(function(t){for(var
n=t.querySelectorAll("a"),o=0;o<n.length;o++)try{var a=n[o],i=a.href.indexOf(c);
i>-
1&&(a.href="mailto:"+r(a.href,i+c.length))}catch(t){e(t)}})(t),function(t){for(v
ar n=t.querySelectorAll(o),c=0;c<n.length;c++)try{var
i=n[c],l=i.parentNode,u=i.getAttribute(a);if(u){var f=r(u,0),d=document.createTe
xtNode(f);l.replaceChild(d,i)}}catch(t){e(t)}}(t),function(t){for(var r=t.queryS
electorAll("template"),c=0;c<r.length;c++)try{n(r[c].content)}catch(t){e(t)}}(t)
}catch(t){e(t)}}var c="/cdn-cgi/l/email-protection#",o=".__cf_email__",a="data-
cfemail",i=document.createElement("div");n(document),function(){var e=document.c
urrentScript||document.scripts[document.scripts.length-
1];e.parentNode.removeChild(e)}()}();
  </script>
  <script>
   (function(){function c(){var
b=a.contentDocument||a.contentWindow.document;if(b){var d=b.createElement('scrip
t');d.innerHTML="window.__CF$cv$params={r:'8e4fdf2c8a923998',t:'MTczMjAxNTg2My4w
MDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/cdn-
cgi/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head'
)[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(docum
ent.body){var a=document.createElement('iframe');a.height=1;a.width=1;a.style.po
sition='absolute';a.style.top=0;a.style.left=0;a.style.border='none';a.style.vis
ibility='hidden';document.body.appendChild(a);if('loading'!==document.readyState
)c();else if(window.addEventListener)document.addEventListener('DOMContentLoaded
',c);else{var e=document.onreadystatechange||function(){};document.onreadystatec
hange=function(b){e(b);'loading'!==document.readyState&&(document.onreadystatech
ange=e,c())}}}})();
  </script>
 </body>
</html>
```

[16]:
```
gme_revenue = pd.DataFrame(columns = ["Date","Revenue"])

for table in soup.find_all('table'):
    if table.find('th').getText().startswith("GameStop Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            if len(col) != 2:
                continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

            gme_revenue = gme_revenue.append({"Date":Date, "Revenue":Revenue},␣
 ↪ignore_index=True)
```

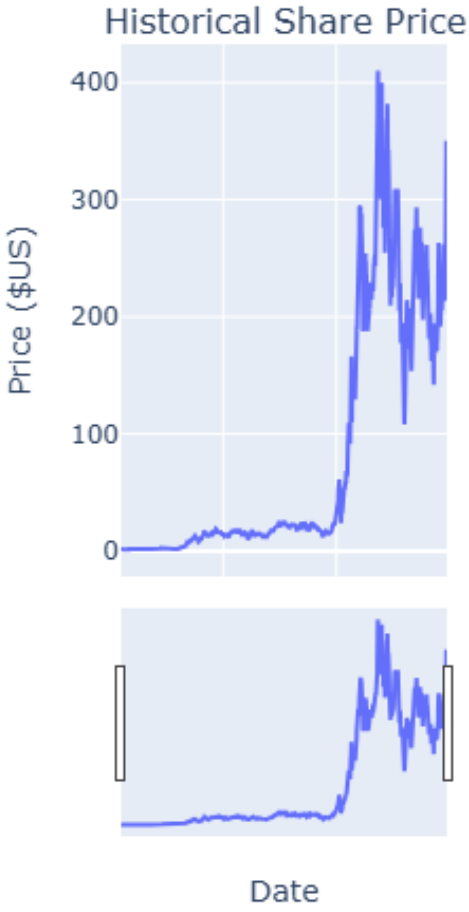
[17]:
```
gme_revenue.tail(5)
```

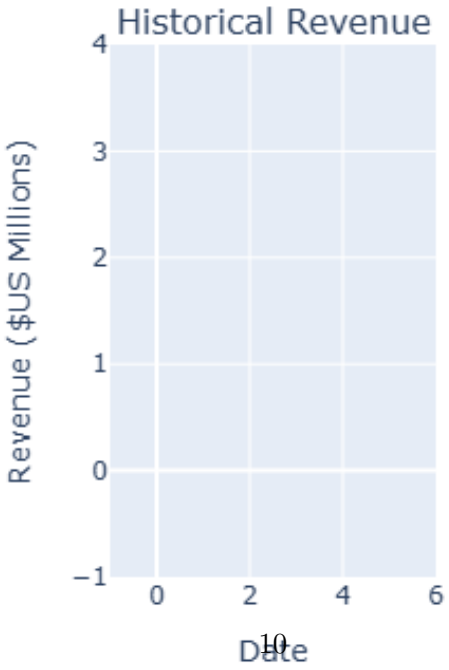
[17]:
```
Empty DataFrame
Columns: [Date, Revenue]
Index: []
```

[18]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla')
```

```
/tmp/ipykernel_160/3121625665.py:3: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.

/tmp/ipykernel_160/3121625665.py:4: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.
```

Tesla

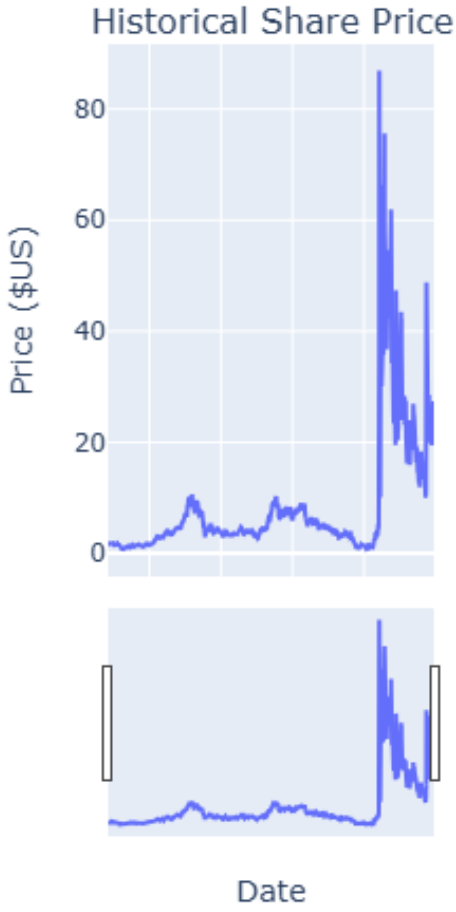

Historical Share Price
Price ($US)
400
300
200
100
0
Date

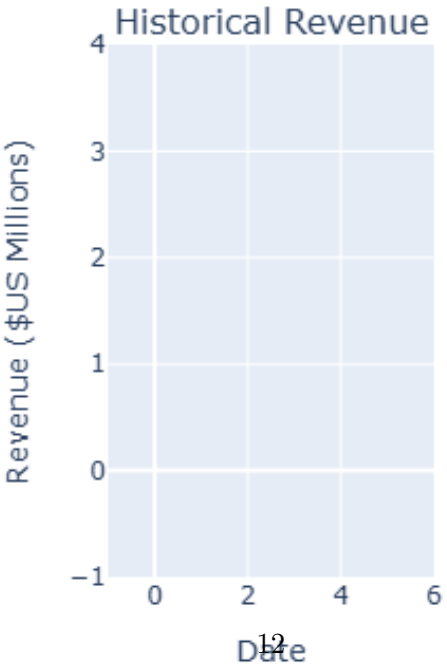

Historical Revenue
Revenue ($US Millions)
4
3
2
1
0
−1
0
2
4
6
Date

[19]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

```
/tmp/ipykernel_160/3121625665.py:3: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.

/tmp/ipykernel_160/3121625665.py:4: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.
```

GameStop


Historical Share Price
Price ($US)
80
60
40
20
0
Date


Historical Revenue
Revenue ($US Millions)
4
3
2
1
0
−1
0
2
4
6
Date

[ ]:

[ ]: