FINAL PROJECT - AMAZON DATA Charlotte Rogers

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [2]:
```
df = pd.read_csv('Retail.OrderHistory.1.csv', dtype=object)
```

In [3]:
```
list(df)
```

Out[3]:
```
['Website',
 'Order ID',
 'Order Date',
 'Purchase Order Number',
 'Currency',
 'Unit Price',
 'Unit Price Tax',
 'Shipping Charge',
 'Total Discounts',
 'Total Owed',
 'Shipment Item Subtotal',
 'Shipment Item Subtotal Tax',
 'ASIN',
 'Product Condition',
 'Quantity',
 'Order Status',
 'Shipment Status',
 'Ship Date',
 'Shipping Option',
 'Carrier Name & Tracking Number',
 'Product Name',
 'Gift Message',
 'Gift Sender Name',
 'Gift Recipient Contact Details']
```

In [4]:
```
df.shape
```

Out[4]:
```
(215, 24)
```

In [5]:
```
df = df.fillna(0)
```

In [6]:
```
pd.set_option('display.max_columns', 27)
```

In [7]:
```
df
```

Out[7]:

	Website	Order ID	Order Date	Purchase Order Number	Currency	Unit Price	Unit Price Tax	Shipping Charge	Total Discounts	Total Owed	Shipment Item Subtotal	Shipment Item Subtotal Tax	A
0	Amazon.com	113-6878423-2831408	2023-04-25T13:40:44.835z	Not Applicable	USD	14.12	0	6.99	0	21.11	Not Available	Not Available	B07LGVZ
1	Amazon.com	113-6732882-0977830	2023-04-23T14:27:15.0z	Not Applicable	USD	68.43	6.07	0	0	74.5	68.43	6.07	B007NX6
2	Amazon.com	114-3612302-9099413	2023-04-20T00:29:05.0z	Not Applicable	USD	12.71	1.13	0	0	13.84	34.69	3.08	B07Y2V1I
3	Amazon.com	114-3612302-9099413	2023-04-20T00:29:05.0z	Not Applicable	USD	21.98	1.95	0	0	23.93	34.69	3.08	B07L87G
4	Amazon.com	111-6057495-0973048	2023-03-27T19:14:05.0z	Not Applicable	USD	11.99	1.06	6.52	0	19.57	11.99	1.06	B07LCBT
...	...	...	...	...	...	...	...	...	...	...	...	...	
210	Amazon.com	112-6953446-6330615	2019-12-15T05:59:38.0z	Not Applicable	USD	5.98	0	0	0	5.98	5.98	0	B07GWJR
211	Amazon.com	112-4902046-7943445	2019-12-05T15:31:40.0z	Not Applicable	USD	5.49	0.51	6.54	0	12.54	5.49	0.51	312367
212	Amazon.com	112-2404022-6093033	2019-12-05T15:30:19.0z	Not Applicable	USD	14.32	1.04	6.42	0	21.78	14.32	1.04	B01CQUX
213	Amazon.com	112-9574863-4399409	2019-12-02T22:45:03.0z	Not Applicable	USD	10.69	0	5.99	0	16.68	10.69	0	B07FQCJ
214	Amazon.com	112-9598378-5432221	2019-11-24T13:38:11.0z	Not Applicable	USD	7.68	0.71	9.81	0	18.2	7.68	0.71	B000NY1

215 rows × 24 columns

In [8]:
```
df['Order Date'] = pd.to_datetime(df['Order Date']).dt.date
```

In [9]:
```
df
```

Out[9]:

	Website	Order ID	Order Date	Purchase Order Number	Currency	Unit Price	Unit Price Tax	Shipping Charge	Total Discounts	Total Owed	Shipment Item Subtotal	Shipment Item Subtotal Tax	ASIN	Pro Cond
0	Amazon.com	113-6878423-2831408	2023-04-25	Not Applicable	USD	14.12	0	6.99	0	21.11	Not Available	Not Available	B07LGVZ42S	
1	Amazon.com	113-6732882-0977830	2023-04-23	Not Applicable	USD	68.43	6.07	0	0	74.5	68.43	6.07	B007NX6LMI	
2	Amazon.com	114-3612302-9099413	2023-04-20	Not Applicable	USD	12.71	1.13	0	0	13.84	34.69	3.08	B07Y2V1HXV	
3	Amazon.com	114-3612302-9099413	2023-04-20	Not Applicable	USD	21.98	1.95	0	0	23.93	34.69	3.08	B07L87GPPT	
4	Amazon.com	111-6057495-0973048	2023-03-27	Not Applicable	USD	11.99	1.06	6.52	0	19.57	11.99	1.06	B07LCBTXZN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210	Amazon.com	112-6953446-6330615	2019-12-15	Not Applicable	USD	5.98	0	0	0	5.98	5.98	0	B07GWJRPTS	
211	Amazon.com	112-4902046-7943445	2019-12-05	Not Applicable	USD	5.49	0.51	6.54	0	12.54	5.49	0.51	312367546	
212	Amazon.com	112-2404022-6093033	2019-12-05	Not Applicable	USD	14.32	1.04	6.42	0	21.78	14.32	1.04	B01CQUX96S	
213	Amazon.com	112-9574863-4399409	2019-12-02	Not Applicable	USD	10.69	0	5.99	0	16.68	10.69	0	B07FQCJJ6K	
214	Amazon.com	112-9598378-5432221	2019-11-24	Not Applicable	USD	7.68	0.71	9.81	0	18.2	7.68	0.71	B000NY17J6	

215 rows × 24 columns

In [10]:
```
list(df)
```

Out[10]:
```
['Website',
 'Order ID',
 'Order Date',
 'Purchase Order Number',
 'Currency',
 'Unit Price',
 'Unit Price Tax',
 'Shipping Charge',
 'Total Discounts',
 'Total Owed',
 'Shipment Item Subtotal',
 'Shipment Item Subtotal Tax',
 'ASIN',
 'Product Condition',
 'Quantity',
 'Order Status',
 'Shipment Status',
 'Ship Date',
 'Shipping Option',
 'Carrier Name & Tracking Number',
 'Product Name',
 'Gift Message',
 'Gift Sender Name',
 'Gift Recipient Contact Details']
```

In [11]:
```
df.drop(['Payment Instrument Type','Order Status','Shipment Status','ASIN','Product Condition','Carrier Name &
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27296\3715074763.py in <module>
----> 1 df.drop(['Payment Instrument Type','Order Status','Shipment Status','ASIN','Product Condition','Carrier Name & Tracking Number','Gift Message','Gift Sender Name'],axis=1,inplace=True)

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    309                     stacklevel=stacklevel,
    310                 )
--> 311             return func(*args, **kwargs)
    312
    313         return wrapper

~\anaconda3\lib\site-packages\pandas\core\frame.py in drop(self, labels, axis, index, columns, level, inplace, errors)
   4955                 weight  1.0     0.8
   4956         """
-> 4957         return super().drop(
   4958             labels=labels,
   4959             axis=axis,

~\anaconda3\lib\site-packages\pandas\core\generic.py in drop(self, labels, axis, index, columns, level, inplace, errors)
   4265         for axis, labels in axes.items():
   4266             if labels is not None:
-> 4267                 obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4268
   4269         if inplace:

~\anaconda3\lib\site-packages\pandas\core\generic.py in _drop_axis(self, labels, axis, level, errors, consolidate, only_slice)
   4309                 new_axis = axis.drop(labels, level=level, errors=errors)
   4310             else:
-> 4311                 new_axis = axis.drop(labels, errors=errors)
   4312             indexer = axis.get_indexer(new_axis)
   4313

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in drop(self, labels, errors)
   6659         if mask.any():
   6660             if errors != "ignore":
-> 6661                 raise KeyError(f"{list(labels[mask])} not found in axis")
   6662             indexer = indexer[~mask]
   6663         return self.delete(indexer)

KeyError: "['Payment Instrument Type'] not found in axis"
```

In [12]:
```
df.shape
```

Out[12]:
```
(215, 24)
```

In [13]:
```
df['Total Owed'].sum()
```

Out[13]:
```
'21.1174.513.8423.9319.57129.9512.9522.811.9517.932.795.995.991.595.4906.795.4911.994.996.394.4927.7934.7220.2210.4115.6454.3110.8422.8526.1225.13013.0422.84215.687.2825.3525.6619.5947.8413.0417.066.5428.048.7310.378.7320.7510.3716.3810.9126.5115.2814.1916.3627.227.320.4422.0912.1318.5213.4915.8215.2716.3623.9320.3529.398.0711.9721.7617.9435.923.5912.2216.19014.6722.8456.612.9713.0814.1836.4530.026.427.516.06258.0115.3624.7816.614.25.3121.7330.5318.9625.875011.7519.8810.15018.5622.9310.9112.9812.016.8510.7116.0814.0611.5422.57.993.494.495.991.773.295.995.293.2919.9117.1327.8619.6420.0636.8432.1631.096.4213.9316.0827.0633.222016.2839.1324.1221.7317.4126.8110.9132.7614.3614.1927.5619.6526.1643.6947.9721.8424.0218.5413.0825.521.8233.2224.663.210015.999.979.6421.713.9213.8724.558.0430.619.1421.8216.6731.2315.2217.9718.92016.9917.9922.996.9943.698.739.823517.947.5712.5834.2816.072020.0764.0413.933.23.213.219.999.9925.9522.857.99006.9920.95.9812.5421.7816.6818.2'
```

In [14]:
```
df['Total Owed'] = pd.to_numeric(df['Total Owed'])
```

In [15]:
```
df['Total Owed'].sum()
```

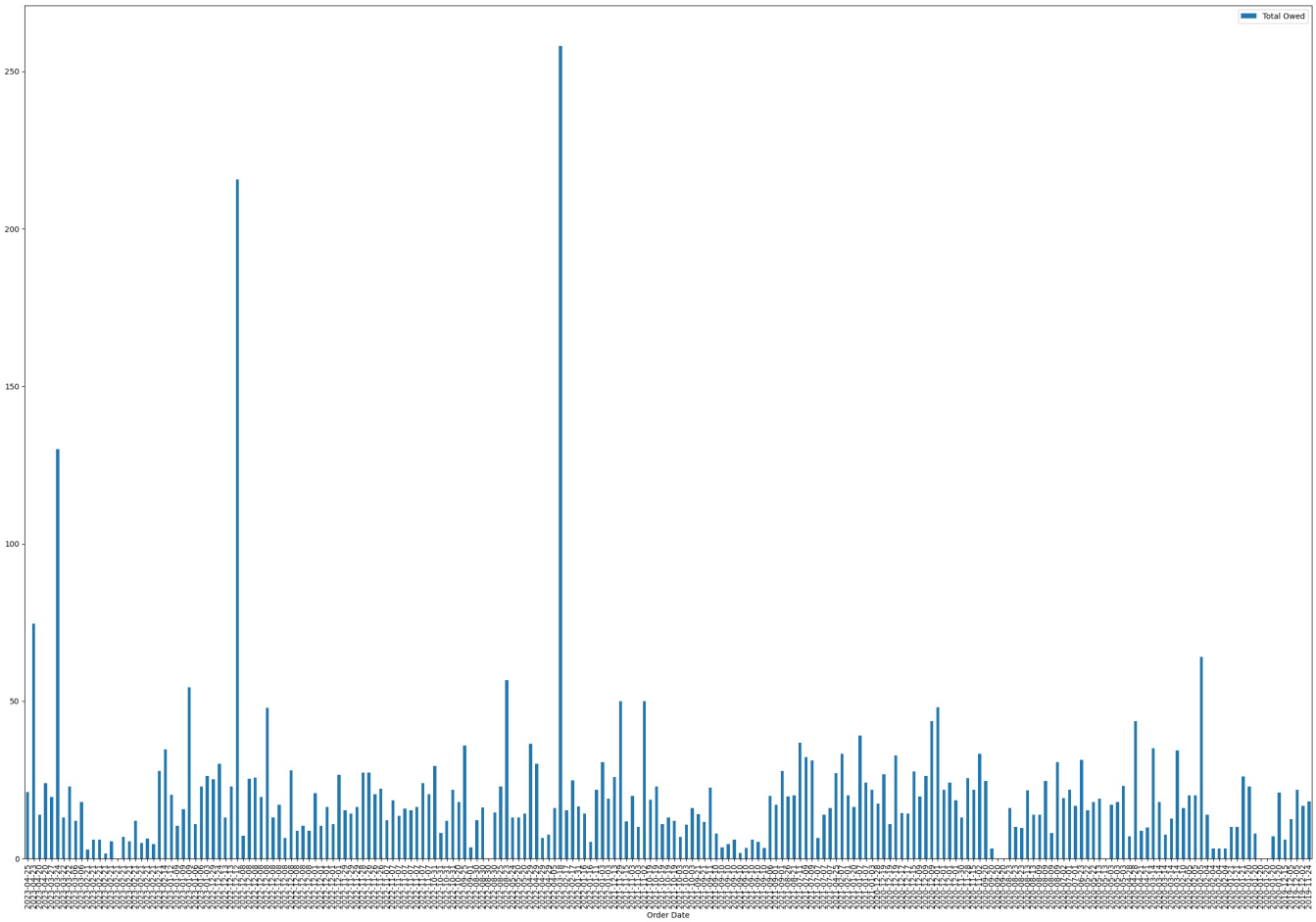
Out[15]:
```
4421.01
```

In [16]:
```
df['Total Owed'].mean()
```

Out[16]:
```
20.5628372093023
```

In [17]:
```
df.plot.bar(x='Order Date', y='Total Owed', rot=90, figsize=(30,20))
```

Out[17]:
```
<AxesSubplot:xlabel='Order Date'>
```

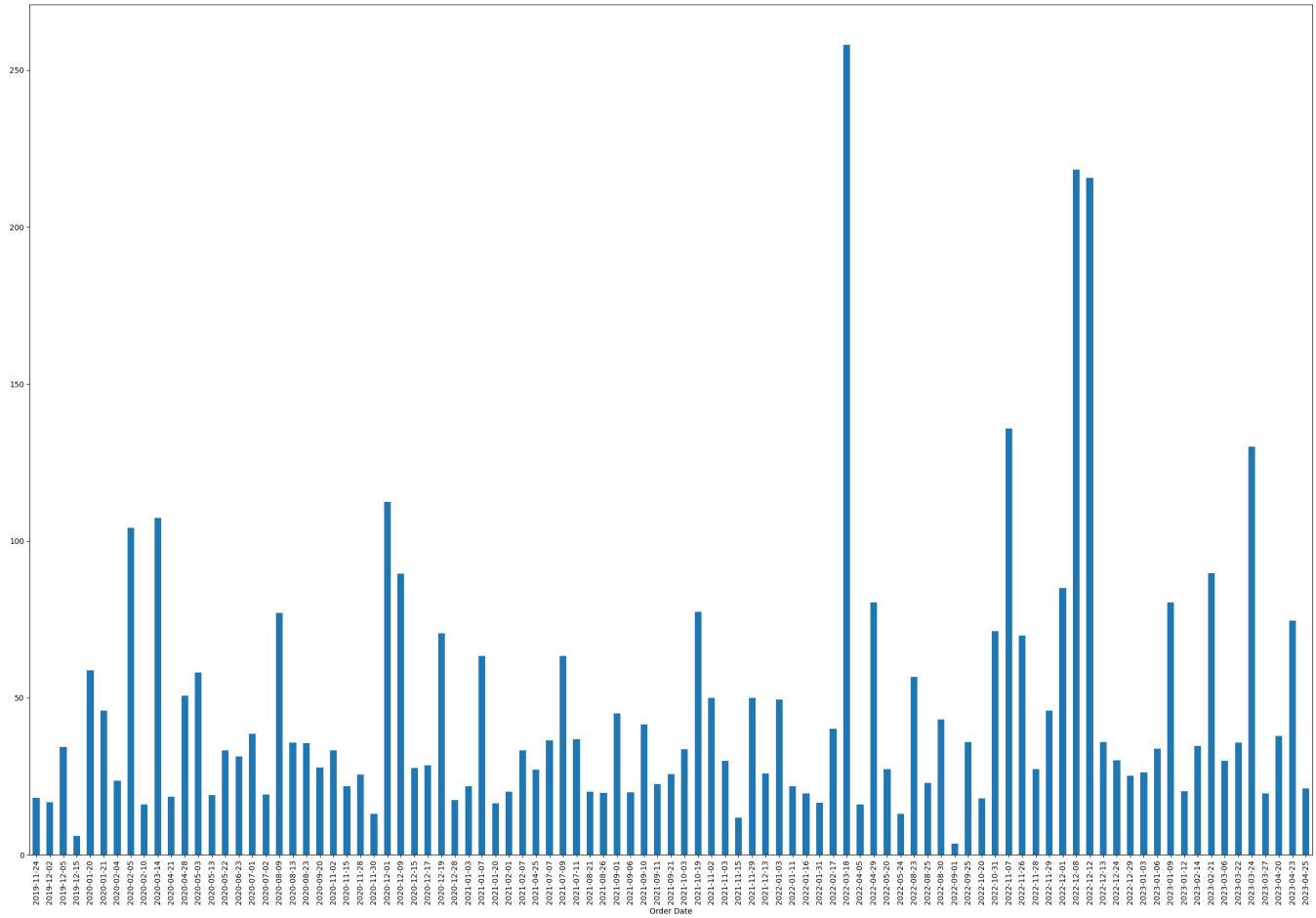


In [18]:
```
same_day_orders = df.groupby('Order Date').sum()["Total Owed"]
same_day_orders.head()
```

Out[18]:
```
Order Date
2019-11-24    18.20
2019-12-02    16.68
2019-12-05    34.32
2019-12-15     5.98
2020-01-20    58.73
Name: Total Owed, dtype: float64
```

In [19]:
```
same_day_orders.plot.bar(figsize=(30,20))
```

Out[19]:
```
<AxesSubplot:xlabel='Order Date'>
```

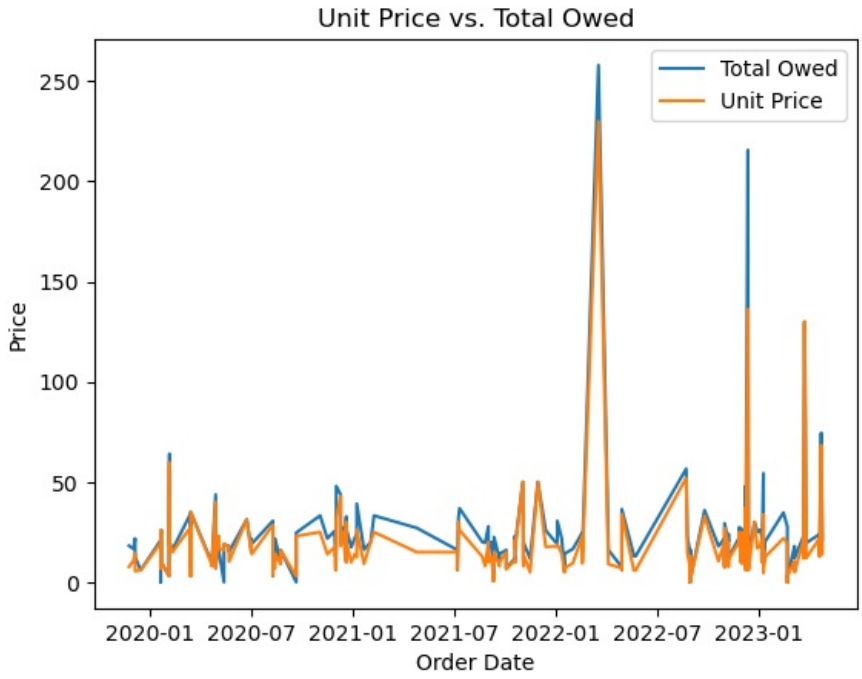


In [20]:
```
df['Unit Price'] = pd.to_numeric(df['Unit Price'])
df['Unit Price'].sum()
```

Out[20]:
```
3617.11
```

In [21]:
```
plt.plot(df['Order Date'],df['Total Owed'])
plt.plot(df['Order Date'],df['Unit Price'])
plt.title("Unit Price vs. Total Owed")
plt.xlabel("Order Date")
plt.ylabel("Price")
plt.legend(["Total Owed", "Unit Price"])
plt.show
```

Out[21]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [22]:
```
pip install bokeh
```

```
Requirement already satisfied: bokeh in c:\users\charo\anaconda3\lib\site-packages (2.4.3)
Requirement already satisfied: typing-extensions>=3.10.0 in c:\users\charo\anaconda3\lib\site-packages (from bo
keh) (4.3.0)
Requirement already satisfied: pillow>=7.1.0 in c:\users\charo\anaconda3\lib\site-packages (from bokeh) (9.2.0)
Requirement already satisfied: tornado>=5.1 in c:\users\charo\anaconda3\lib\site-packages (from bokeh) (6.1)
Requirement already satisfied: Jinja2>=2.9 in c:\users\charo\anaconda3\lib\site-packages (from bokeh) (2.11.3)
Requirement already satisfied: packaging>=16.8 in c:\users\charo\anaconda3\lib\site-packages (from bokeh) (21.3
)
Requirement already satisfied: PyYAML>=3.10 in c:\users\charo\anaconda3\lib\site-packages (from bokeh) (6.0)
Requirement already satisfied: numpy>=1.11.3 in c:\users\charo\anaconda3\lib\site-packages (from bokeh) (1.21.5
)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\charo\anaconda3\lib\site-packages (from Jinja2>=2.9
->bokeh) (2.0.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\charo\anaconda3\lib\site-packages (from pac
kaging>=16.8->bokeh) (3.0.9)
Note: you may need to restart the kernel to use updated packages.
```

In [23]:
```
from bokeh.plotting import figure, show
```

In [24]:
```
new_plot = figure(title="Amazon Spending Data", x_axis_label='Order Dates', y_axis_label='Amount Spent')
new_plot.line(df['Order Date'], df['Total Owed'], legend_label="Orders", line_width=5)
```

Out[24]: **GlyphRenderer**(id = '1039', …)

In [25]:
```
show(new_plot)
```

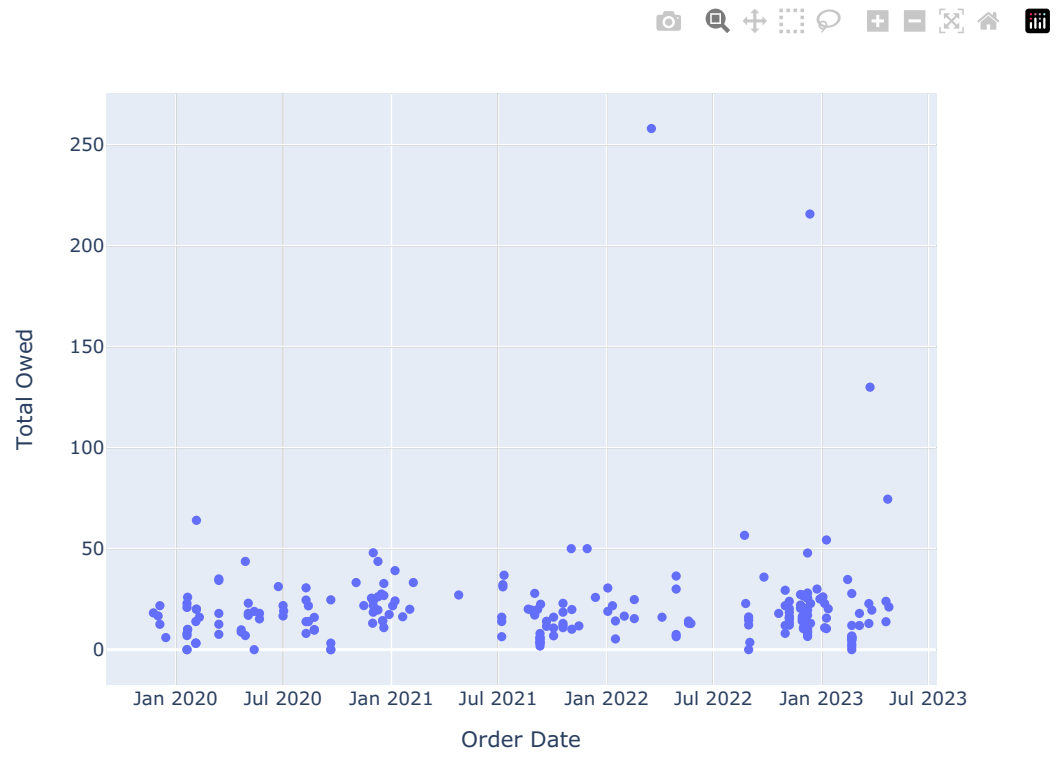
In [26]:
```
new_plot = figure(title="Amazon Spending Data", x_axis_label='Order Dates', y_axis_label='Amount Spent')
new_plot.line(df['Order Date'], df['Total Owed'], legend_label="Total Paid", color='black', line_width=5)
new_plot.line(df['Order Date'], df['Unit Price'], legend_label="Item Price", color='red', line_width=2)
new_plot.line(df['Order Date'], df['Unit Price Tax'], legend_label="Tax", color='pink', line_width=2)
show(new_plot)
```

In [27]:
```
pip install plotly==5.14.1
```

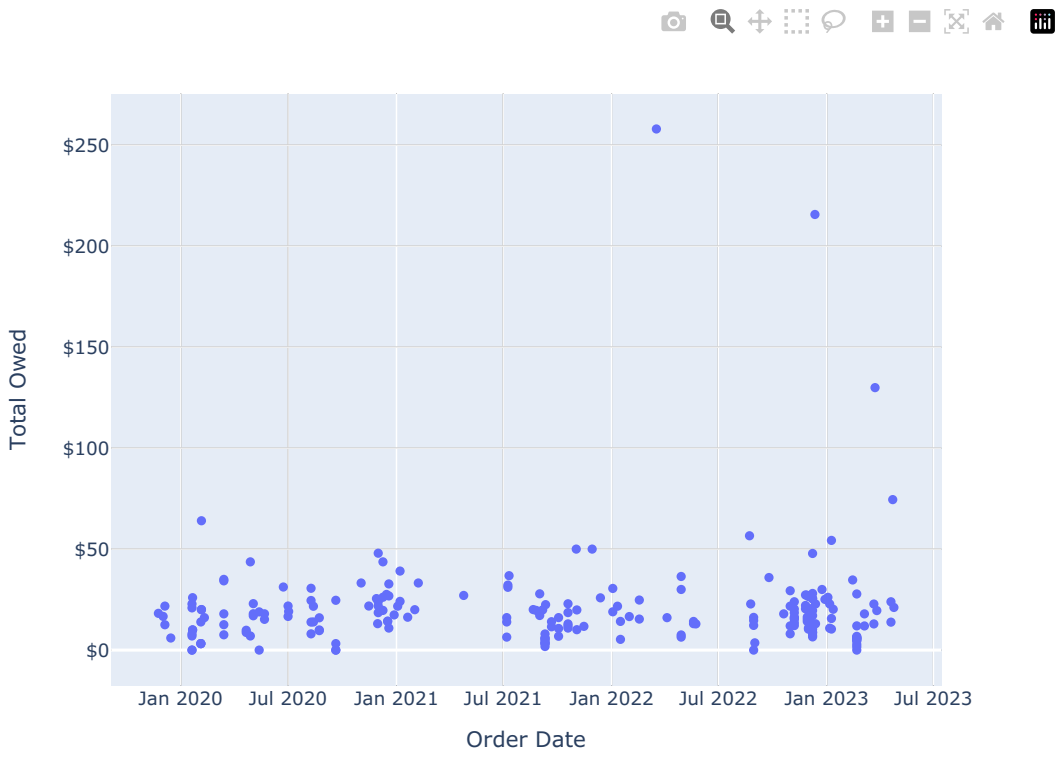
```
Requirement already satisfied: plotly==5.14.1 in c:\users\charo\anaconda3\lib\site-packages (5.14.1)
Requirement already satisfied: packaging in c:\users\charo\anaconda3\lib\site-packages (from plotly==5.14.1) (2
1.3)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\charo\anaconda3\lib\site-packages (from plotly==5.14
.1) (8.0.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\charo\anaconda3\lib\site-packages (from pac
kaging->plotly==5.14.1) (3.0.9)
Note: you may need to restart the kernel to use updated packages.
```

In [28]:
```
import plotly.express as px
```

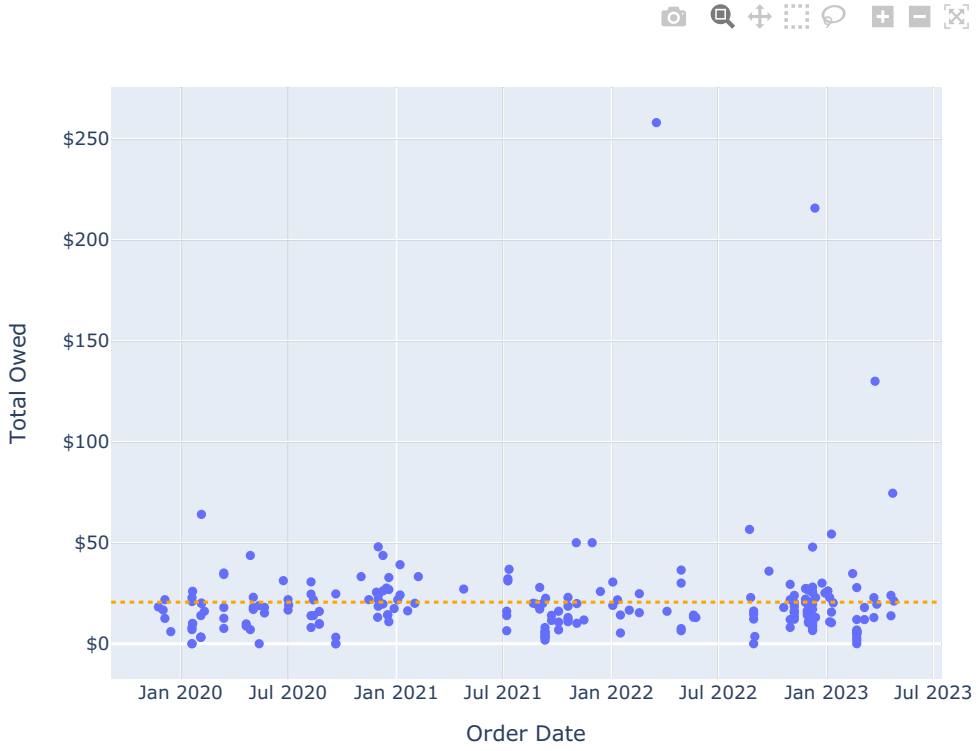
In [29]:
```
scattered = px.scatter(df, 'Order Date', 'Total Owed')
scattered.show()
```



In [30]:
```
scattered.update_yaxes(tickprefix="$", showgrid=True)
```



In [31]:
```
scattered.add_shape(type="line", line_color="orange", line_width=2, opacity=1, line_dash="dot", x0=0, x1=1, xre
```


$250
$200
$150
$100
$50
$0
Total Owed
Jan 2020
Jul 2020
Jan 2021
Jul 2021
Jan 2022
Jul 2022
Jan 2023
Jul 2023
Order Date

In [ ]: