Visual Data Analysis of Fraudulent Transactions

Your CFO has also requested detailed trends data on specific card holders. Use the starter notebook to query your database and generate visualizations that supply the requested information as follows, then add your visualizations and observations to your markdown report.

In [1]:
```
# Initial imports
import pandas as pd
import calendar
import plotly.express as px
import hvplot.pandas
from sqlalchemy import create_engine
```

In [2]:
```
# Create a connection to the database
engine = create_engine("postgresql://postgres:postgres@localhost:5432/fraud_detection")
```

Data Analysis Question 1

The two most important customers of the firm may have been hacked. Verify if there are any fraudulent transactions in their history. For privacy reasons, you only know that their cardholder IDs are 2 and 18.

- Using hvPlot, create a line plot representing the time series of transactions over the course of the year for each cardholder separately.
- Next, to better compare their patterns, create a single line plot that containins both card holders' trend data.
- What difference do you observe between the consumption patterns? Does the difference suggest a fraudulent transaction? Explain your rationale in the markdown report.

In [3]:
```
# loading data for card holder 2 and 18 from the database
query = """
        SELECT ch.id AS cardholder, t.date AS hour, t.amount
        FROM transaction AS t
        JOIN credit_card AS cc ON cc.card = t.card
        JOIN card_holder AS ch ON ch.id = cc.id_card_holder
        WHERE ch.id in (2, 18)
        ORDER BY hour
        """
df_question1 = pd.read_sql(query, engine)

df_question1.head()
```

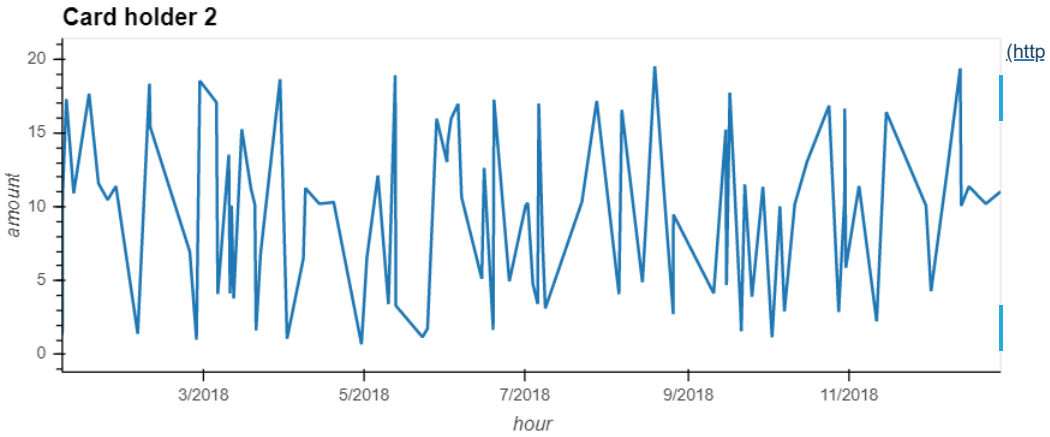
Out[3]:

	cardholder	hour	amount
0	18	2018-01-01 23:15:10	2.95
1	18	2018-01-05 07:19:27	1.36
2	2	2018-01-06 02:16:41	1.33
3	2	2018-01-06 05:13:20	10.82
4	18	2018-01-07 01:10:54	175.00

In [4]:
```
# Plot for cardholder 2
data_2 = df_question1[df_question1["cardholder"] == 2]
data_2_plot = data_2.hvplot.line("hour", "amount", label="Card holder 2", dynamic=False)

data_2_plot
```

Out[4]:

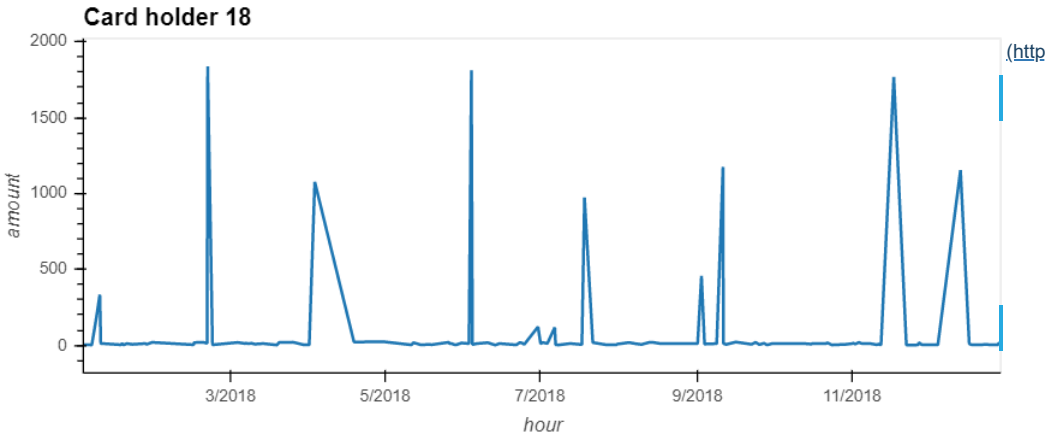


(http

In [5]:
```
# Plot for cardholder 18
data_18 = df_question1[df_question1["cardholder"] == 18]
data_18_plot = data_18.hvplot.line("hour", "amount", label="Card holder 18")

data_18_plot
```

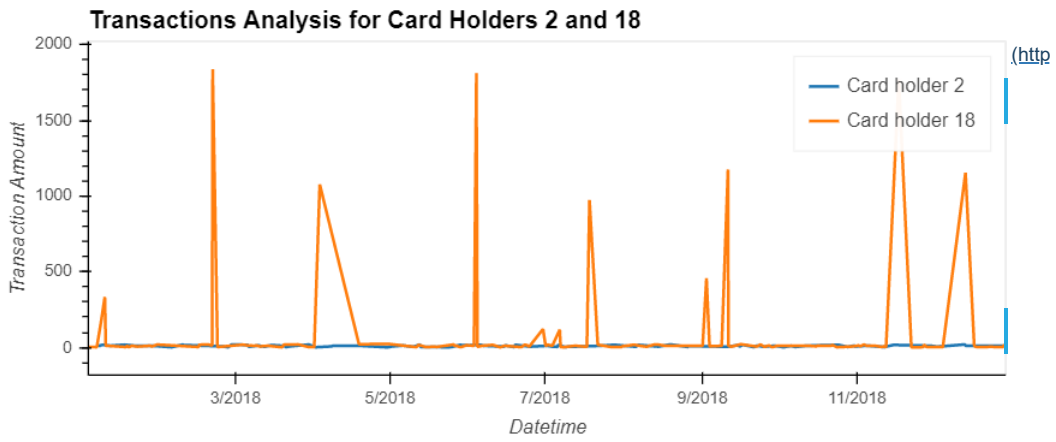
Out[5]:



(http

In [6]:
```
# Combined plot for card holders 2 and 18
plot_1 = data_2_plot * data_18_plot
plot_1.opts(
    title="Transactions Analysis for Card Holders 2 and 18",
    xlabel="Datetime",
    ylabel="Transaction Amount",
)
```

Out[6]:



(http

Sample Conclusions for Question 1

After visualy analyzed the plots and the spending patterns, it can be concluded that there may be some fraudulent transactions in the card holder 18 records since there are some anomalous amounts throughout the year that break the typical spending pattern that can be seen on card holder 2.

Data Analysis Question 2

The CEO of the biggest customer of the firm suspects that someone has used her corporate credit card without authorization in the first quarter of 2018 to pay quite expensive restaurant bills. Again, for privacy reasons, you know only that the cardholder ID in question is 25.

- Using Plotly Express, create a box plot, representing the expenditure data from January 2018 to June 2018 for cardholder ID 25.
- Are there any outliers for cardholder ID 25? How many outliers are there per month?
- Do you notice any anomalies? Describe your observations and conclusions in your markdown report.

In [7]:
```
# loading data of daily transactions from jan to jun 2018 for card holder 25
query = """
        SELECT date_part('month', t.date) AS month, date_part('day', t.date) as day, t.amount
        FROM transaction AS t
        JOIN credit_card AS cc ON cc.card = t.card
        JOIN card_holder AS ch ON ch.id = cc.id_card_holder
        WHERE ch.id = 25 AND date_part('month', t.date) <= 6
        ORDER BY month, day
        """
df_question2 = pd.read_sql(query, engine)

df_question2.head()
```

Out[7]:

	month	day	amount
0	1.0	2.0	1.46
1	1.0	5.0	10.74
2	1.0	7.0	2.93
3	1.0	10.0	1.39
4	1.0	14.0	17.84

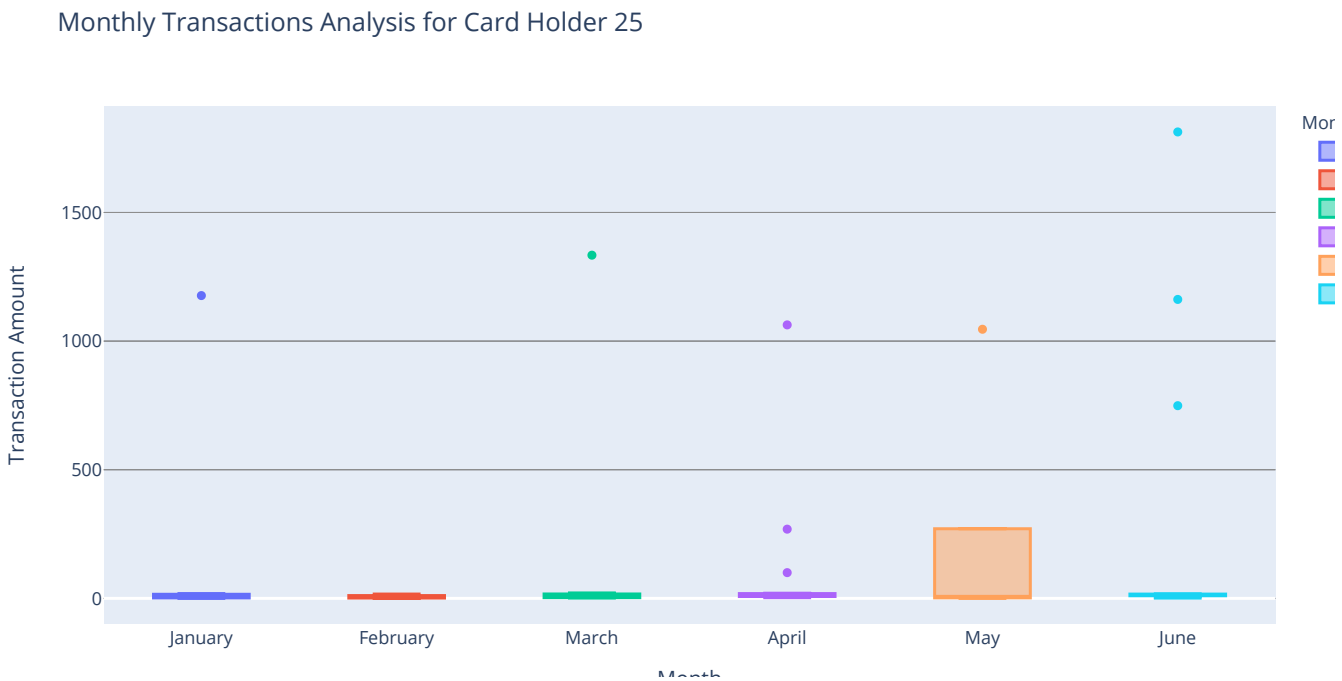
In [8]:
```
# Loop to change the numeric month to month names
for i in range(df_question2.shape[0]):
    df_question2.iloc[i, 0] = calendar.month_name[int(df_question2.iloc[i, 0])]

df_question2.head()
```

Out[8]:

	month	day	amount
0	January	2.0	1.46
1	January	5.0	10.74
2	January	7.0	2.93
3	January	10.0	1.39
4	January	14.0	17.84

In [9]:
```
# Creating the six box plots using plotly express
px.box(
    df_question2,
    x="month",
    y="amount",
    title="Monthly Transactions Analysis for Card Holder 25",
    labels={"month": "Month", "amount": "Transaction Amount"},
    color="month",
    boxmode="overlay",
)
```



Sample Conclusions for Question 2

It can be concluded that card holder 25 has been hacked along all the first semester of 2018, exept for february where there are not anomalous transactions.