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 hvplot.pandas
from sqlalchemy import create_engine
```

In [2]:
```
# Create a connection to the database
engine = create_engine("postgresql://postgres:momo250400_@localhost:5432/Suspect_db")
```

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.

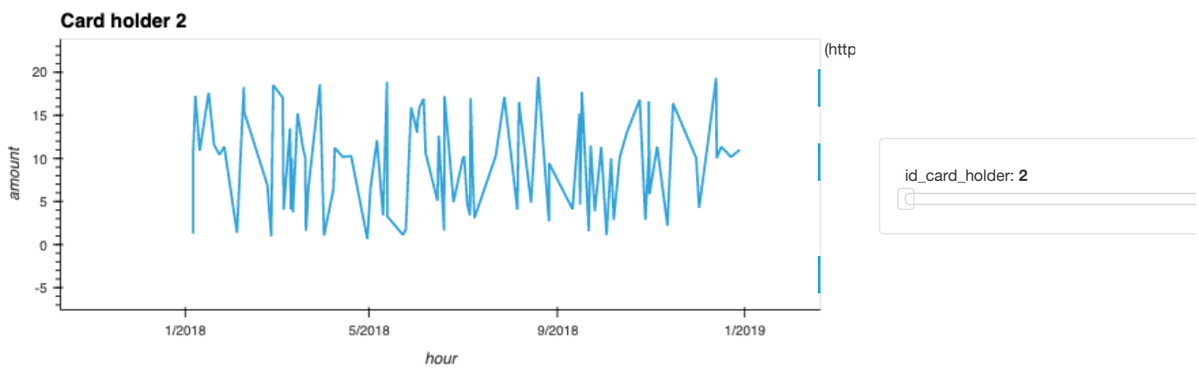
In [3]:
```
# loading data for card holder 2 and 18 from the database
# Write the query
query = """
SELECT a.id_card_holder, a.card_holder_name, b.card, c.amount, c.date AS hour
FROM card_holder AS a, credit_card AS b, transaction AS c
WHERE (a.id_card_holder = 2 OR a.id_card_holder= 18) AND
a.id_card_holder = b.card_holder_id AND
b.card = c.card
GROUP BY a.id_card_holder, b.card, c.date, c.amount
ORDER BY c.date ASC;
"""
# Create a DataFrame from the query result. HINT: Use pd.read_sql(query, engine)
transaction_df = pd.read_sql(query, engine)
#Display
display(transaction_df.head(3),transaction_df.tail(3))
```

	id_card_holder	card_holder_name	card	amount	hour
0	18	Malik Carlson	4498002758300	2.95	2018-01-01 23:15:10
1	18	Malik Carlson	344119623920892	1.36	2018-01-05 07:19:27
2	2	Shane Shaffer	4866761290278198714	1.33	2018-01-06 02:16:41

	id_card_holder	card_holder_name	card	amount	hour
229	18	Malik Carlson	344119623920892	12.88	2018-12-28 09:00:45
230	2	Shane Shaffer	675911140852	11.03	2018-12-28 15:30:55
231	18	Malik Carlson	4498002758300	12.25	2018-12-29 08:11:55

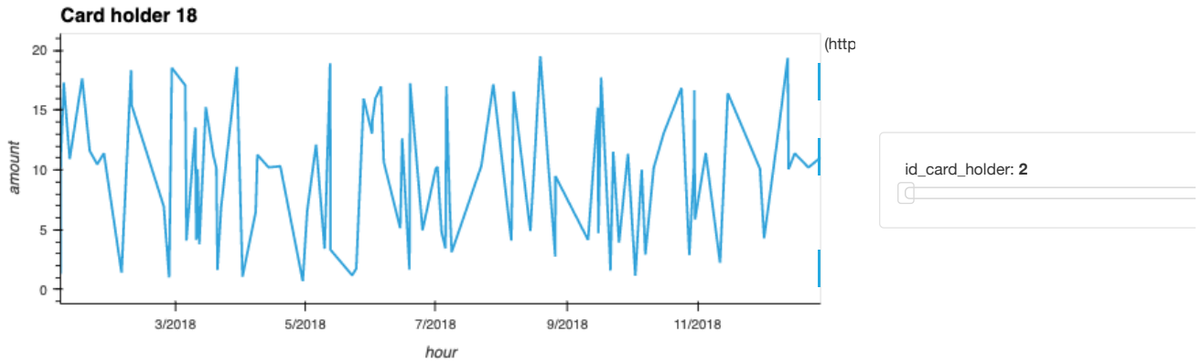
In [4]:
```
# Plot for cardholder 2
lineplot = transaction_df.hvplot.line(
    x = 'hour',
    y = 'amount',
    groupby = 'id_card_holder',
    xlabel = "hour",
    ylabel = "amount",
    title = "Card holder 2",
    size = (1000, 500)
).opts(yformatter = '%.0f')
lineplot
```

Out[4]:



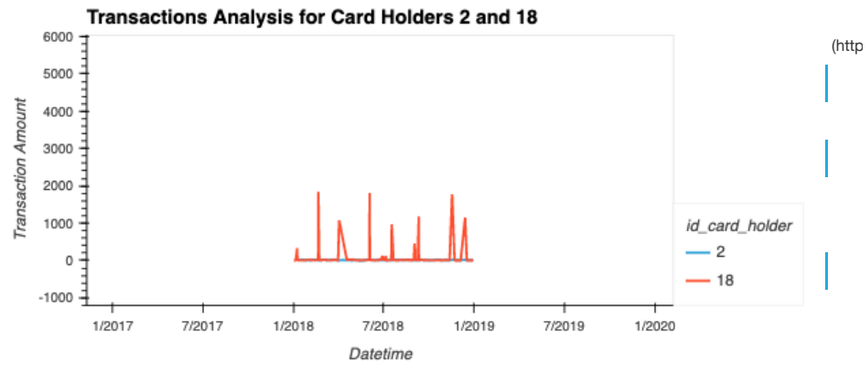
In [5]:
```
# Plot for cardholder 18
lineplot2 = transaction_df.hvplot.line(
    x = 'hour',
    y = 'amount',
    groupby = 'id_card_holder',
    xlabel = "hour",
    ylabel = "amount",
    title = "Card holder 18",
    size = (1000, 500)
).opts(yformatter = '%.0f')
lineplot2
```

Out[5]:



In [6]:
```
# Combined plot for card holders 2 and 18
lineplot3 = transaction_df.hvplot.line(
    x = 'hour',
    y = 'amount',
    by = 'id_card_holder',
    xlabel = "Datetime",
    ylabel = "Transaction Amount",
    title = "Transactions Analysis for Card Holders 2 and 18",
    size = (1000, 500)
).opts(yformatter = '%.0f')
lineplot3
```

Out[6]:



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 hvPlot, 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
# Write the query
query2 = """
SELECT EXTRACT(MONTH FROM c.date) AS month, EXTRACT(DAY FROM c.date) AS day, c.amount
FROM card_holder AS a, credit_card AS b, transaction AS c
WHERE (a.id_card_holder = 25) AND
a.id_card_holder = b.card_holder_id AND
b.card = c.card AND
CAST(c.date AS DATE) < '2018-07-01'
GROUP BY a.id_card_holder, c.date, c.amount
ORDER BY c.date ASC
        """
# Create a DataFrame from the query result. HINT: Use pd.read_sql(query, engine)
transaction_df2 = pd.read_sql(query2, engine)
#View
display(transaction_df2.head(3),transaction_df2.tail(3))
```

	month	day	amount
0	1.0	2.0	1.46
1	1.0	5.0	10.74
2	1.0	7.0	2.93

	month	day	amount
65	6.0	25.0	11.53
66	6.0	27.0	5.24
67	6.0	30.0	2.27

In [8]:
```
# loop to change the numeric month to month names
transaction_df2["month"] = transaction_df2["month"].astype('int')
transaction_df2["month"] = transaction_df2["month"].apply(lambda x: calendar.month_name[x])
```

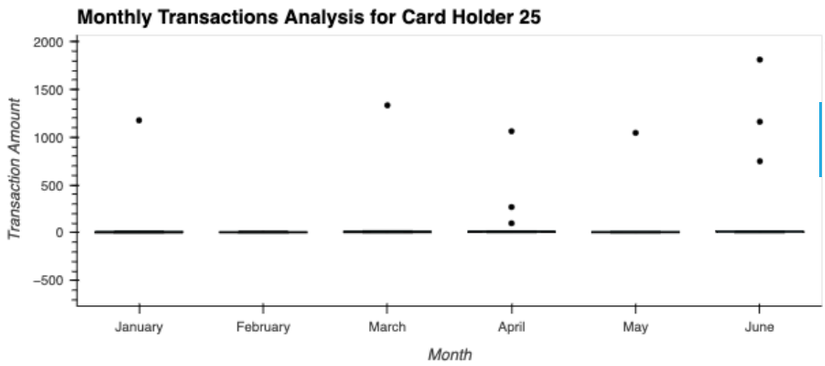
In [9]:
```
display(transaction_df2.head(3),transaction_df2.tail(3))
```

	month	day	amount
0	January	2.0	1.46
1	January	5.0	10.74
2	January	7.0	2.93

	month	day	amount
65	June	25.0	11.53
66	June	27.0	5.24
67	June	30.0	2.27

In [10]:
```
# Creating the six box plots using hvPlot
transaction_df2.hvplot.box(
    'amount',
    by = 'month',
    xlabel = "Month",
    ylabel = "Transaction Amount",
    color = "amount",
    title = "Monthly Transactions Analysis for Card Holder 25",
    legend = "top_right"
)
```

Out[10]:



(http

In [ ]: