```
!pip install pathway bokeh --quiet
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.4/60.4 kB 2.3 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 149.4/149.4 kB 5.7 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 69.7/69.7 MB 9.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.6/77.6 kB 6.8 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 777.6/777.6 kB 47.8 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.2/139.2 kB 11.7 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 26.5/26.5 MB 75.1 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.5/45.5 kB 3.6 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.3/135.3 kB 10.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 244.6/244.6 kB 19.1 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 319.1/319.1 kB 23.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 985.8/985.8 kB 50.6 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 148.6/148.6 kB 12.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.8/139.8 kB 11.1 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 65.8/65.8 kB 6.1 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.7/55.7 kB 4.5 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 118.5/118.5 kB 11.0 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 196.2/196.2 kB 15.4 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 434.9/434.9 kB 31.3 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 83.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 87.0 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.3/13.3 MB 96.0 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 83.2/83.2 kB 7.4 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.2/2.2 MB 79.5 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 63.2 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
bigframes 2.8.0 requires google-cloud-bigquery[bqstorage,pandas]>=3.31.0, but you have google-cloud-bigquery 3.29.0 which is incompatible.
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
from datetime import datetime
import pathway as pw
import bokeh.plotting
import panel as pn
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files dataset.csv

- **dataset.csv**(text/csv) - 1595541 bytes, last modified: 7/1/2025 - 100% done

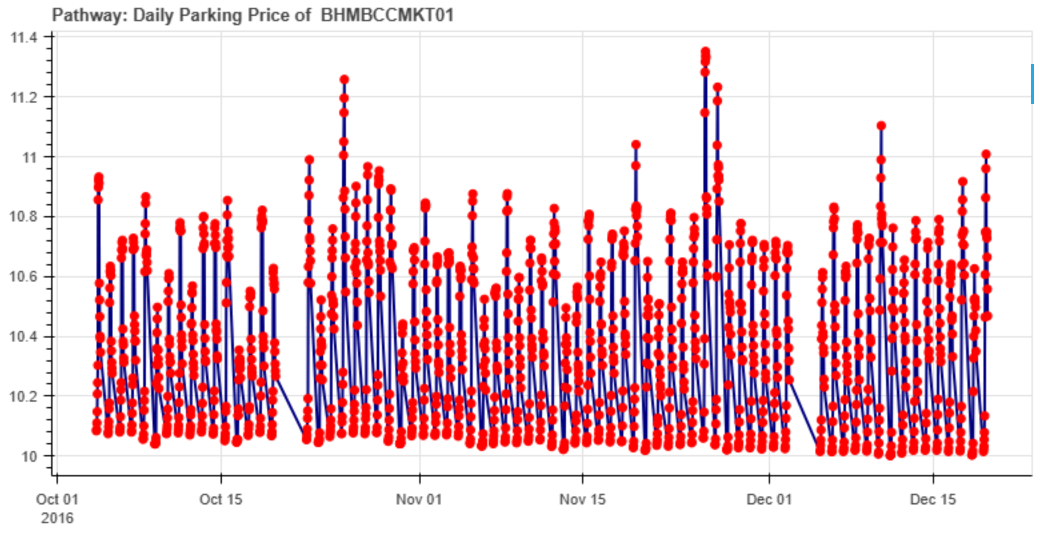
```
df=pd.read_csv('dataset.csv')
```

```
#making a list of parking lots to get plots one by one
lotss=df['SystemCodeNumber'].unique()
lots=lotss[0:14]
lots
```

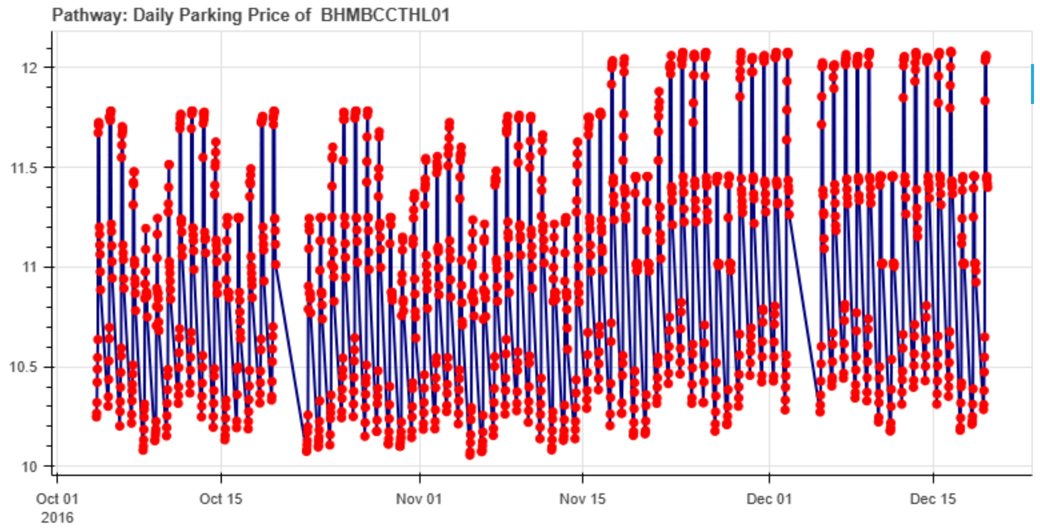
```
array(['BHMBCCMKT01', 'BHMBCCTHL01', 'BHMEURBRD01', 'BHMMBMMBX01',
       'BHMNCPHST01', 'BHMNCPNST01', 'Broad Street', 'Others-CCCPS105a',
       'Others-CCCPS119a', 'Others-CCCPS135a', 'Others-CCCPS202',
       'Others-CCCPS8', 'Others-CCCPS98', 'Shopping'], dtype=object)
```

```
plots=[]
for lot in lots:
  df['Timestamp'] = pd.to_datetime(df['LastUpdatedDate'] + ' ' + df['LastUpdatedTime'],  #combining date and time columns and converting it in datetime format
                                  format='%d-%m-%Y %H:%M:%S')
  df_lot=df[df['SystemCodeNumber']==lot]   #making dataframe for one lot at a time
  df_lot = df_lot.sort_values('Timestamp').reset_index(drop=True)  #sorting time

  df_lot[["Timestamp", "Occupancy", "Capacity","SystemCodeNumber"]].to_csv("parking_stream.csv", index=False)  #filtering out the columns that are being used for model 1
  class ParkingSchema(pw.Schema):
    Capacity:int
    Occupancy:int
    Timestamp:str
    SystemCodeNumber:str

  filename = f"parking_stream_{lot}.csv"
  df_lot[["Timestamp", "Occupancy", "Capacity", "SystemCodeNumber"]].to_csv(filename, index=False)
  data = pw.demo.replay_csv(filename, schema=ParkingSchema, input_rate=1000)  # Load the data as a simulated stream using Pathway's replay_csv function
  fmt = "%Y-%m-%d %H:%M:%S"
  data_with_time = data.with_columns(
      t = data.Timestamp.dt.strptime(fmt),   #containes full datetime
      day = data.Timestamp.dt.strptime(fmt).dt.strftime("%Y-%m-%dT00:00:00"),  #contains only day date
      hour = data.Timestamp.dt.strptime(fmt).dt.hour(),  #contains hour
      day_of_week = data.Timestamp.dt.strptime(fmt).dt.weekday(),   #assigns monday: 0,...,sunday: 6
      occupancy_rate = data.Occupancy / data.Capacity
  )
  def time_of_day_weight(hour):    #the reason for choosing these categories is explained in the report

    if 11 <= hour < 14:
        return 1.0    # Midday
    elif 14 <= hour < 17:
        return 0.7    # Evening
    else:
        return 0.4    # Morning

  def weekday_weight(day_of_week):   #the reason for choosing these categories is explained in the report

    if day_of_week < 5:
        return 1.0    # Weekday
    else:
        return 0.7    # Weekend

  def pricing_fn(occ_rate, tod_weight, wd_weight):
    return 10.0 + 2.0 * occ_rate * tod_weight * wd_weight

  import datetime

  data_with_price=(
     data_with_time.with_columns(
         tod_weight = pw.apply(time_of_day_weight, data_with_time.hour),
         wd_weight = pw.apply(weekday_weight, data_with_time.day_of_week)



        )
    .with_columns(
        price=pw.apply(
        pricing_fn,
        data_with_time.occupancy_rate,
        pw.this.tod_weight,
        pw.this.wd_weight
    )

    )


)

  plot_table = data_with_price.select(data_with_price.t, data_with_price.price,data_with_price.SystemCodeNumber)
  pn.extension()

#writing bokeh plot fn
  def price_plotter(source):

    # Create a Bokeh figure with datetime x-axis
    fig = bokeh.plotting.figure(
        height=400,
        width=800,
        title=f"Pathway: Daily Parking Price of  {lot}",
        x_axis_type="datetime",
    )
    # Plot a line graph showing how the price evolves over time
    fig.line("t", "price", source=source, line_width=2, color="navy")

    # Overlay red circles at each data point for better visibility
    fig.scatter("t", "price", source=source, size=6, color="red")

    return fig


# - 'sorting_col="t"' ensures the data is plotted in time order
  viz = plot_table.plot(price_plotter, sorting_col="t")
  plots.append(pn.Column(f"Lot: {lot}", viz.servable()))


dashboard = pn.Column(*plots)   ## Create a dashboard container that holds all individual plots stored in 'plots'
dashboard.servable()   # Make the dashboard servable
```

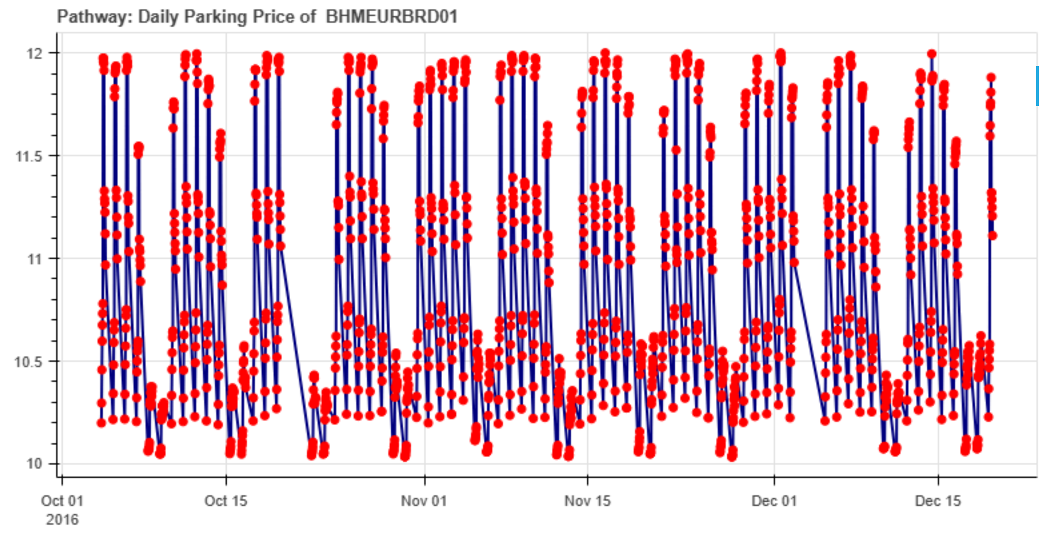
```
Lot: BHMBCCMKT01

Streaming mode
```



```
Lot: BHMBCCTHL01

Streaming mode
```

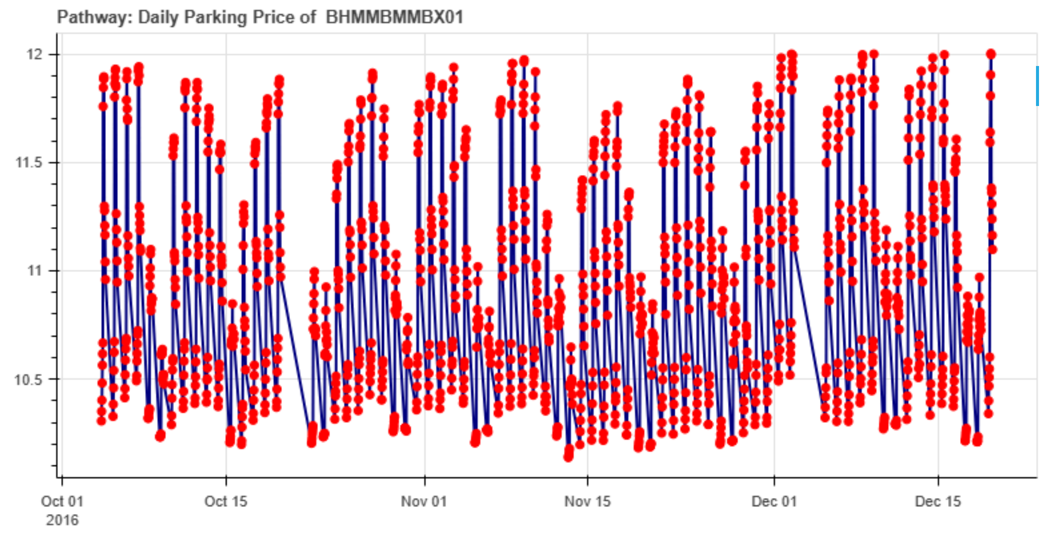


```
Lot: BHMEURBRD01

Streaming mode
```



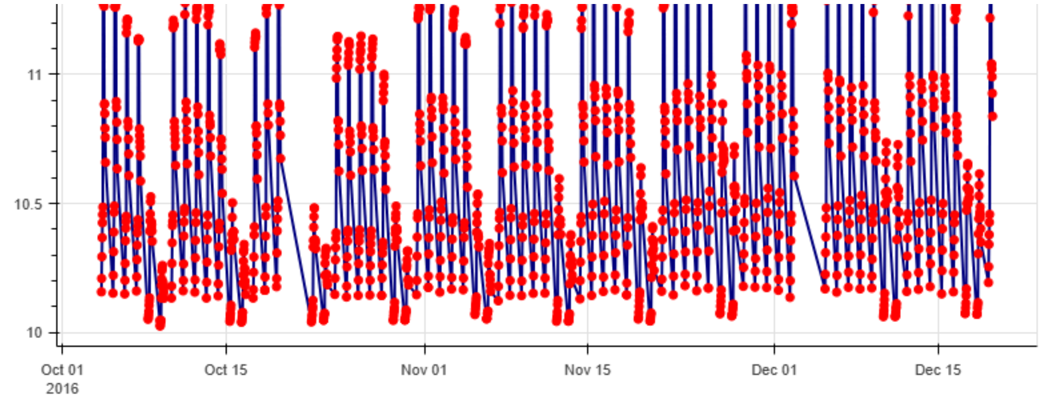
Lot: BHMMBMMBX01

Streaming mode



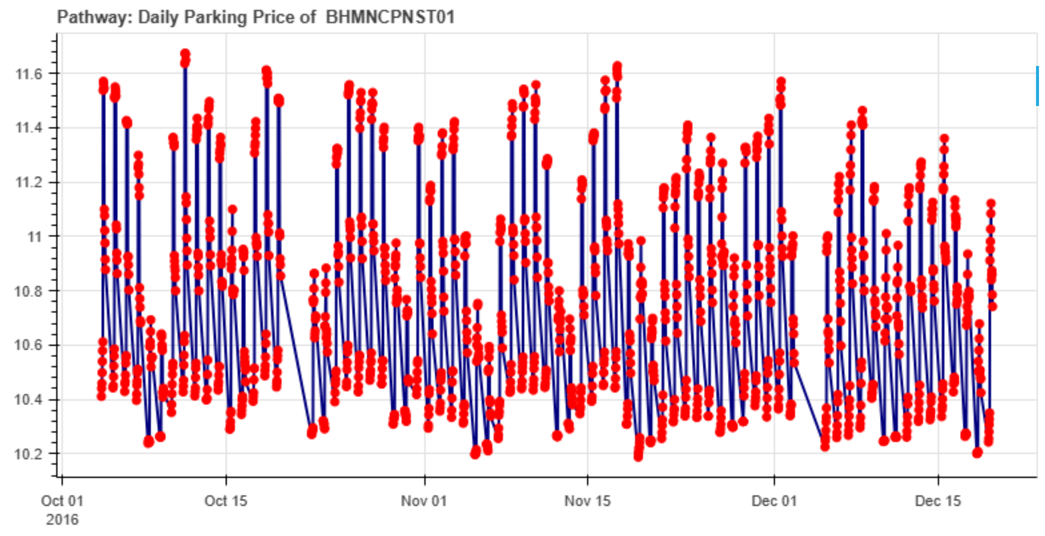
Lot: BHMNCPHST01

Streaming mode



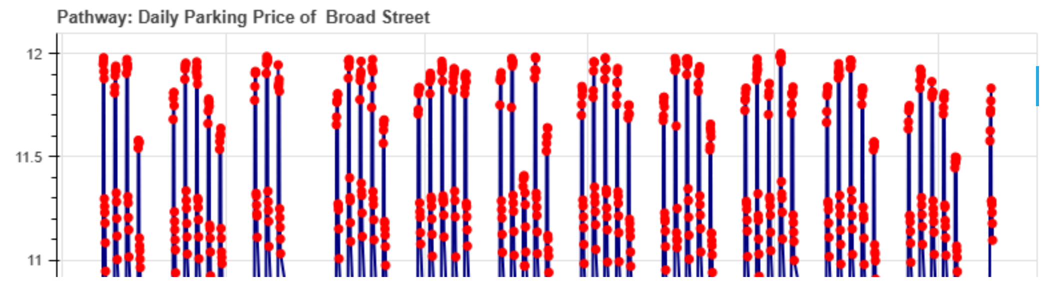
Lot: BHMNCPNST01

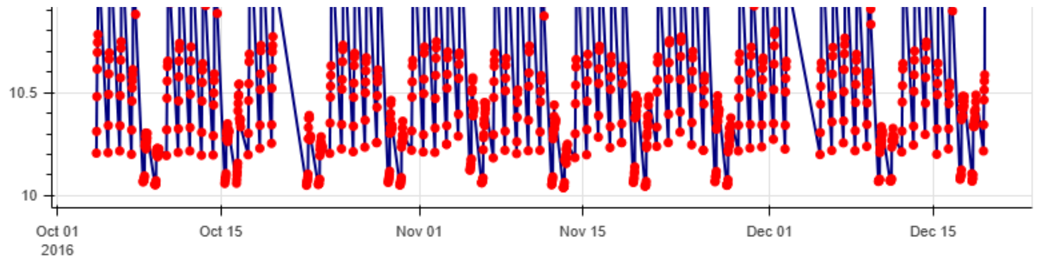
Streaming mode



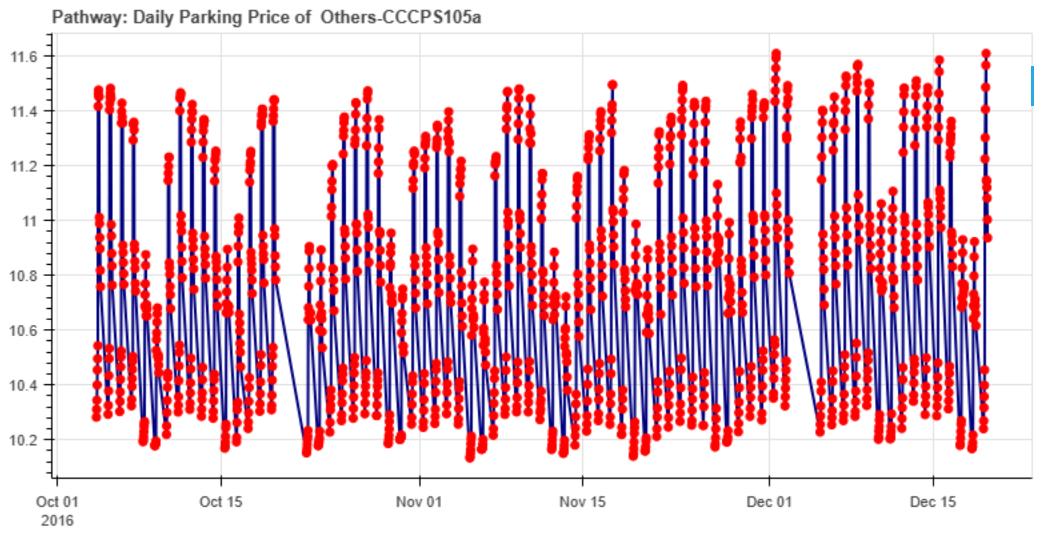
Lot: Broad Street

Streaming mode

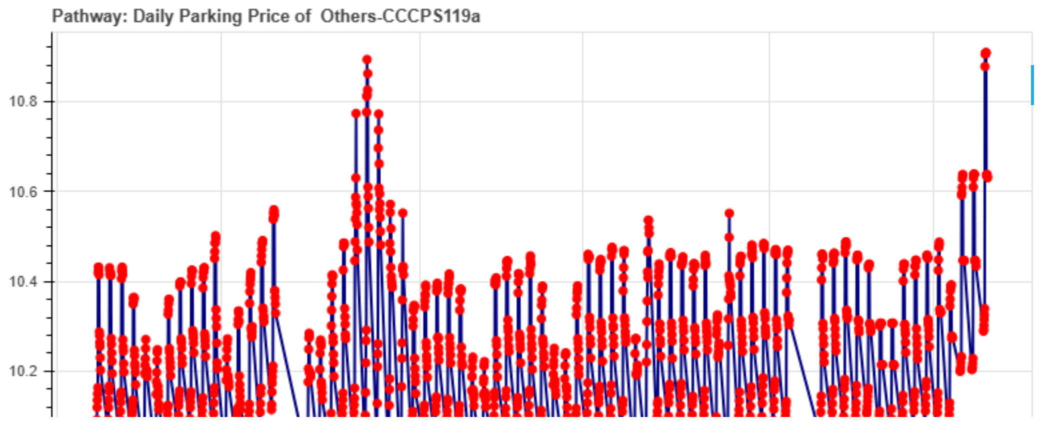


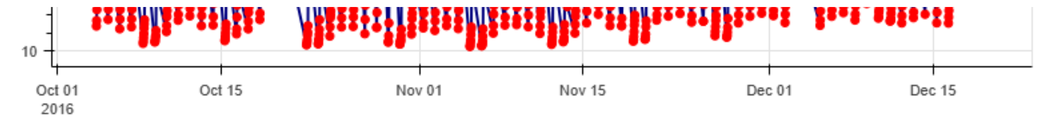


```
Lot: Others-CCCPS105a

Streaming mode
```



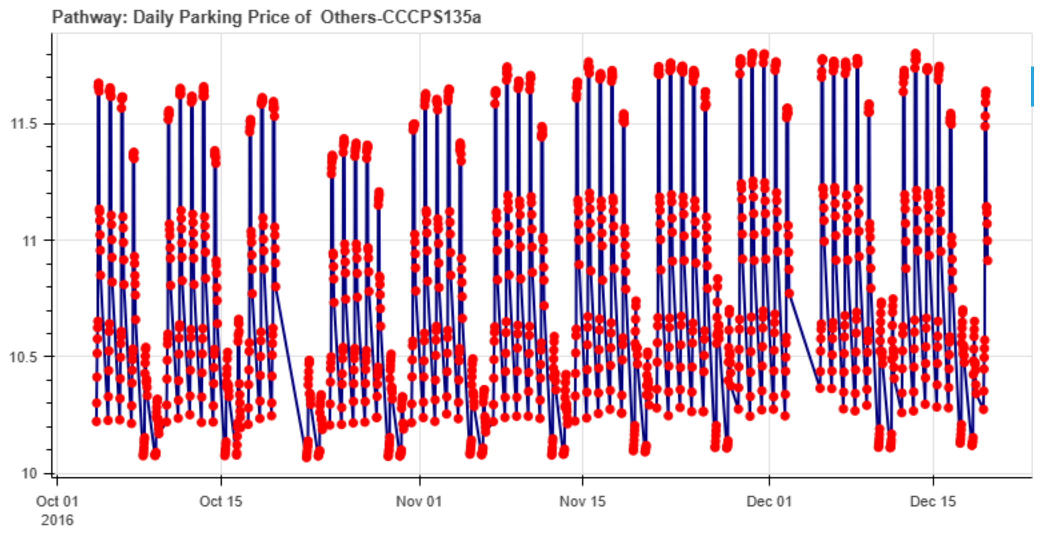
```
Lot: Others-CCCPS119a

Streaming mode
```





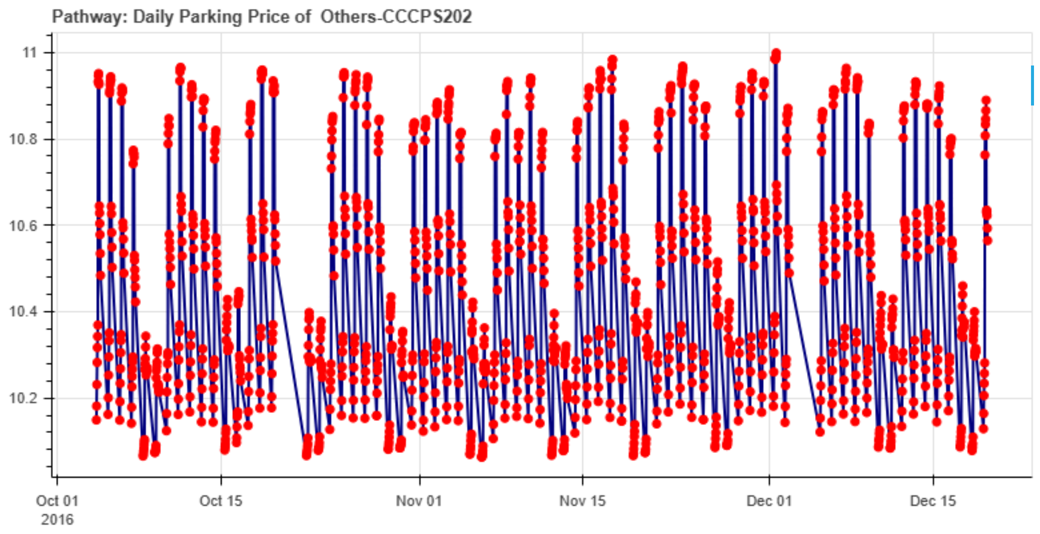
Lot: Others-CCCPS135a

Streaming mode



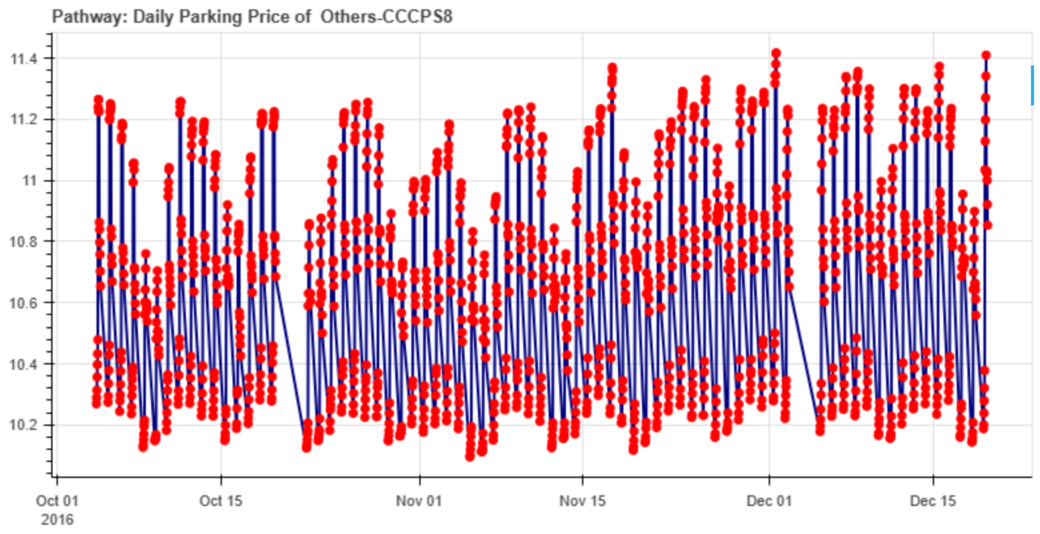
Lot: Others-CCCPS202

Streaming mode



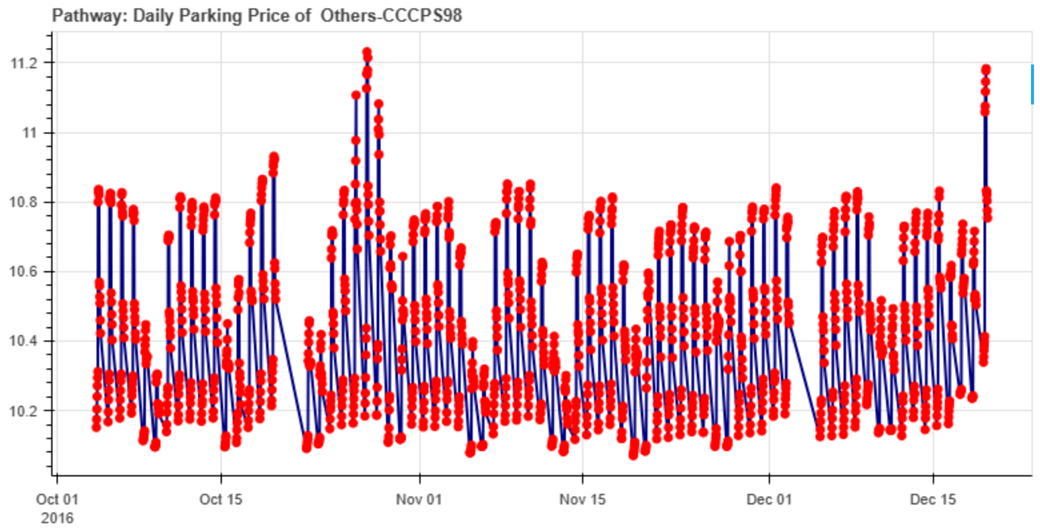
Lot: Others-CCCPS8

Streaming mode



Lot: Others-CCCPS98

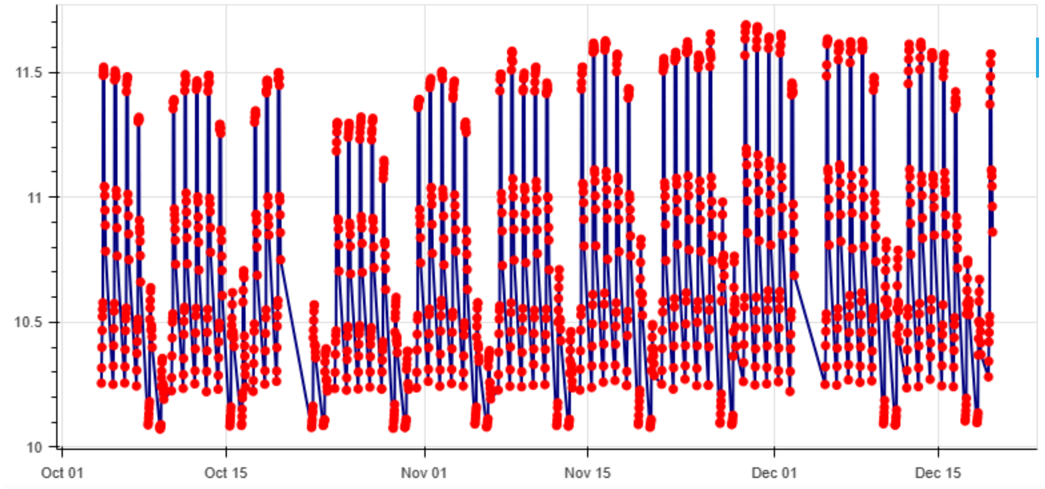
Streaming mode



Lot: Shopping

Streaming mode

Pathway: Daily Parking Price of Shopping



```
pw.run()
```

```
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
WARNING:pathway_engine.connectors.monitoring:PythonReader: Closing the data source
```

Start coding or generate with AI.