CIS9760_Project3_Yaheng Wu(Phoebe)

Streaming Finance Data with AWS Lambda - Visualization Section

Big Idea: In this project, I've collected one day's worth of stock HIGH and LOW prices for each company listed below on December 1st, 2020, at a five minute interval with a Lambda function. This function is triggered every 5 minutes and places the collected data into a defined Kinesis Delivery Stream which was pointed to a S3 bucket as its destination. Then I configured AWS Glue to crawl this S3 bucket which allowed me to query the S3 files using AWS Athena to gain insight into the streamed data. The **results.csv** used in this Jupyter Notebook is the query output generated from Athena.

Below are the companies' stock tickers

- Facebook (FB)
- Shopify (SHOP)
- Beyond Meat (BYND)
- Netflix (NFLX)
- Pinterest (PINS)
- Square (SQ)
- The Trade Desk (TTD)
- Okta (OKTA)
- Snap (SNAP)
- Datadog (DDOG)

1. Import libraries

In [1]:
```
import numpy as np
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
import matplotlib.pyplot as plt
```

2. Read data

In [2]:
```
df = pd.read_csv("results.csv")
df
```

Out[2]:

	name	max_high	ts	hour
0	BYND	140.910004	2020-12-01 09:40:00-05:00	9
1	BYND	139.154999	2020-12-01 10:00:00-05:00	10
2	BYND	138.500000	2020-12-01 11:25:00-05:00	11
3	BYND	138.720001	2020-12-01 12:30:00-05:00	12
4	BYND	138.880005	2020-12-01 13:40:00-05:00	13
...	...	...	...	...
67	TTD	885.780029	2020-12-01 11:15:00-05:00	11
68	TTD	883.440002	2020-12-01 12:00:00-05:00	12
69	TTD	892.000000	2020-12-01 13:50:00-05:00	13
70	TTD	887.530029	2020-12-01 14:25:00-05:00	14
71	TTD	893.159973	2020-12-01 15:55:00-05:00	15

72 rows × 4 columns

3. Display unique values of the name column

In [3]:
```
df['name'].unique()
```

Out[3]:
```
array(['BYND', 'DDOG', 'FB', 'NFLX', 'OKTA', 'PINS', 'SHOP', 'SNAP', 'SQ',
       'TTD'], dtype=object)
```

4. Relabel stock tickers to company name

In [4]:
```
label_df = df.replace({'name' : { 'FB' : 'Facebook', 'SHOP' : 'Shopify', 'BYND' : 'Beyond Meat', 'NFLX' : 'Ne
tflix', \
                       'PINS': 'Pinterest', 'SQ': 'Square', 'TTD': 'The Trade Desk', \
                       'OKTA': 'Okta', 'SNAP': 'Snap', 'DDOG': 'Datadog' }})
```

5. Create seperate dataframe for each company

In [5]:
```
df_FB = label_df.loc[label_df['name'] == 'Facebook']
df_FB = df_FB.sort_values(by=['hour'])
df_NFLX = label_df.loc[label_df['name'] == 'Netflix']
df_NFLX = df_NFLX.sort_values(by=['hour'])
df_SHOP = label_df.loc[label_df['name'] == 'Shopify']
df_SHOP = df_SHOP.sort_values(by=['hour'])
df_BYND = label_df.loc[label_df['name'] == 'Beyond Meat']
df_BYND = df_BYND.sort_values(by=['hour'])
df_PINS = label_df.loc[label_df['name'] == 'Pinterest']
df_PINS = df_PINS.sort_values(by=['hour'])
df_SQ = label_df.loc[label_df['name'] == 'Square']
df_SQ = df_SQ.sort_values(by=['hour'])
df_TTD = label_df.loc[label_df['name'] == 'The Trade Desk']
df_TTD = df_TTD.sort_values(by=['hour'])
df_OKTA = label_df.loc[label_df['name'] == 'Okta']
df_OKTA = df_OKTA.sort_values(by=['hour'])
df_SNAP = label_df.loc[label_df['name'] == 'Snap']
df_SNAP = df_SNAP.sort_values(by=['hour'])
df_DDOG = label_df.loc[label_df['name'] == 'Datadog']
df_DDOG = df_DDOG.sort_values(by=['hour'])
```

6. Create a combined dataframe from multiple lists

In [6]:
```
df_hourly_high = pd.DataFrame(list(zip(df_FB['max_high'], df_NFLX['max_high'], df_SHOP['max_high'], \
                                       df_BYND['max_high'], df_PINS['max_high'], df_SQ['max_high'], \
                                       df_TTD['max_high'], df_OKTA['max_high'], df_SNAP['max_high'], \
                                       df_DDOG['max_high'])),\
                              columns=['Facebook','Netflix', 'Shopify', 'Beyond Meat', 'Pinterest', 'Square', \
                                       'The Trade Desk', 'Okta', 'Snap', 'Datadog'],
                              index = df_FB['hour'])
```

In [7]:
```
df_hourly_high
```

Out[7]:

hour	Facebook	Netflix	Shopify	Beyond Meat	Pinterest	Square	The Trade Desk	Okta	Snap	Datadog
9	284.929993	505.193115	1086.000000	140.910004	71.099998	212.949997	905.000000	244.270004	45.200001	98.800003
10	286.109985	507.420013	1087.729858	139.154999	69.519997	209.550003	895.950012	241.399994	44.849998	96.879997
11	288.950012	504.539886	1077.849854	138.500000	69.388702	207.923996	885.780029	239.500000	44.990002	95.889999
12	289.299988	506.480011	1076.810059	138.720001	69.150002	207.415100	885.780029	239.000000	44.919998	95.540001
13	288.959991	509.470001	1081.839966	138.880005	69.250000	208.320007	883.440002	240.039993	45.150002	95.489998
14	288.750000	507.494995	1075.170044	138.880005	69.080002	207.449997	892.000000	238.872894	44.974998	95.739998
15	287.119995	505.339996	1072.280029	138.669998	68.370003	204.000000	887.530029	237.865005	44.799999	96.029999

7. Visualize the Hourly “high” Stock Price for each company using plotly

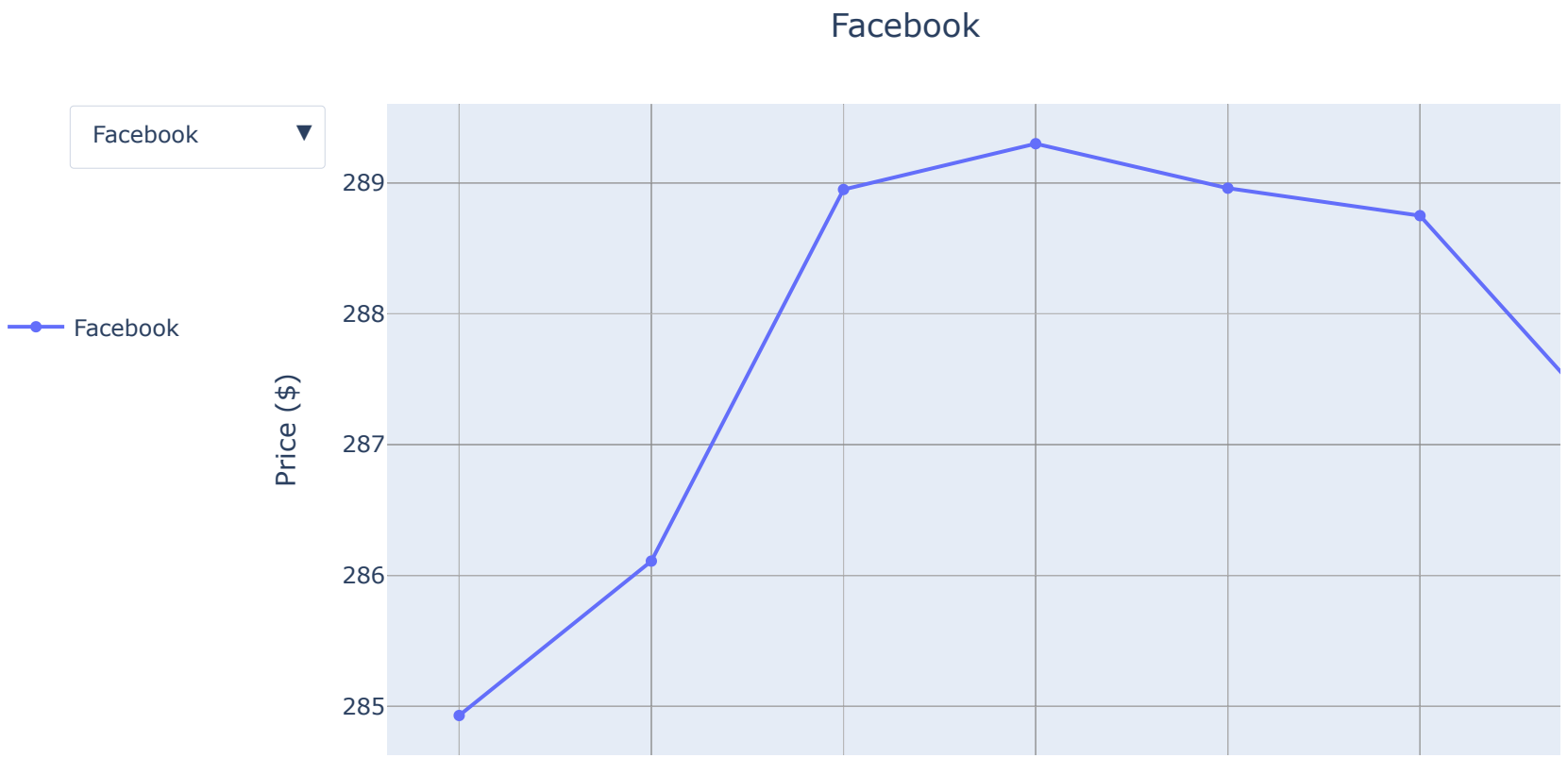
In [8]:
```
fig = go.Figure()

for column in df_hourly_high.columns.to_list():
    fig.add_trace(
        go.Scatter(
            x = df_hourly_high.index,
            y = df_hourly_high[column],
            name = column
        )
    )

fig.update_layout(
    title={
        'text': "Hourly “high” Stock Price on December 1st, 2020",
        'y':0.9,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    xaxis_title="Hour",
    yaxis_title="Price ($)",
    legend=dict(
        x=-0.3,
        y=0.7,
        traceorder='normal',
        font=dict(size=12)),
    updatemenus=[go.layout.Updatemenu(
        active=0,
        buttons=list(
            [dict(label = 'All',
                  method = 'update',
                  args = [{'visible': [True, True, True, True, True, True, True, True, True, True]},
                          {'title': 'All',
                           'showlegend':True}]),
             dict(label = 'Facebook',
                  method = 'update',
                  args = [{'visible': [True, False, False, False, False, False, False, False, False, False]},
                          {'title': 'Facebook',
                           'showlegend':True}]),
             dict(label = 'Netflix',
                  method = 'update',
                  args = [{'visible': [False, True, False, False, False, False, False, False, False, False]},
                          {'title': 'Netflix',
```

```
                    'showlegend':True}]),
             dict(label = 'Shopify',
                  method = 'update',
                  args = [{'visible': [False, False, True, False, False, False, False, False, False, False]},
                          {'title': 'Shopify',
                           'showlegend':True}]),
             dict(label = 'Beyond Meat',
                  method = 'update',
                  args = [{'visible': [False, False, False, True, False, False, False, False, False, False]},
                          {'title': 'Beyond Meat',
                           'showlegend':True}]),
             dict(label = 'Pinterest',
                  method = 'update',
                  args = [{'visible': [False, False, False, False, True, False, False, False, False, False]},
                          {'title': 'Pinterest',
                           'showlegend':True}]),
             dict(label = 'Square',
                  method = 'update',
                  args = [{'visible': [False, False, False, False, False, True, False, False, False, False]},
                          {'title': 'Square',
                           'showlegend':True}]),
             dict(label = 'The Trade Desk',
                  method = 'update',
                  args = [{'visible': [False, False, False, False, False, False, True, False, False, False]},
                          {'title': 'The Trade Desk',
                           'showlegend':True}]),
             dict(label = 'Okta',
                  method = 'update',
                  args = [{'visible': [False, False, False, False, False, False, False, True, False, False]},
                          {'title': 'Okta',
                           'showlegend':True}]),
             dict(label = 'Snap',
                  method = 'update',
                  args = [{'visible': [False, False, False, False, False, False, False, False, True, False]},
                          {'title': 'Snap',
                           'showlegend':True}]),
             dict(label = 'Datadog',
                  method = 'update',
                  args = [{'visible': [False, False, False, False, False, False, False, False, False, True]},
                          {'title': 'Datadog',
                           'showlegend':True}])
            ])
    )
```

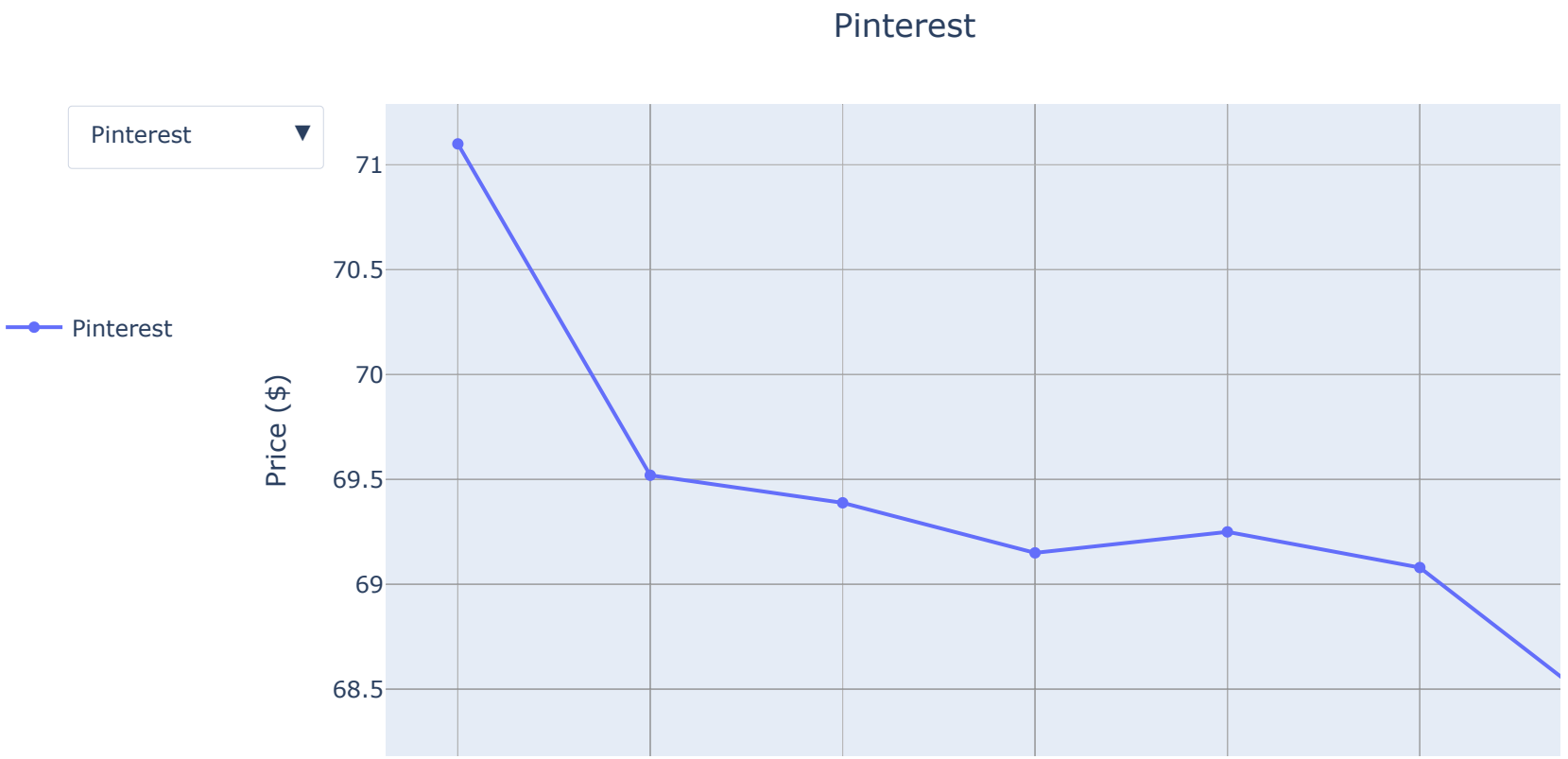
```
    ])

fig.show()
```


Hourly “high” Stock Price on December 1st, 2020
All
Facebook
Netflix
Shopify
Beyond Meat
Pinterest
Square
The Trade Desk
Okta
Snap
Datadog
Price ($)
1000
800
600
400
200
0

In [13]:
```
fig.show()
```



In [14]:
```
fig.show()
```



8. Prepare data for the second chart

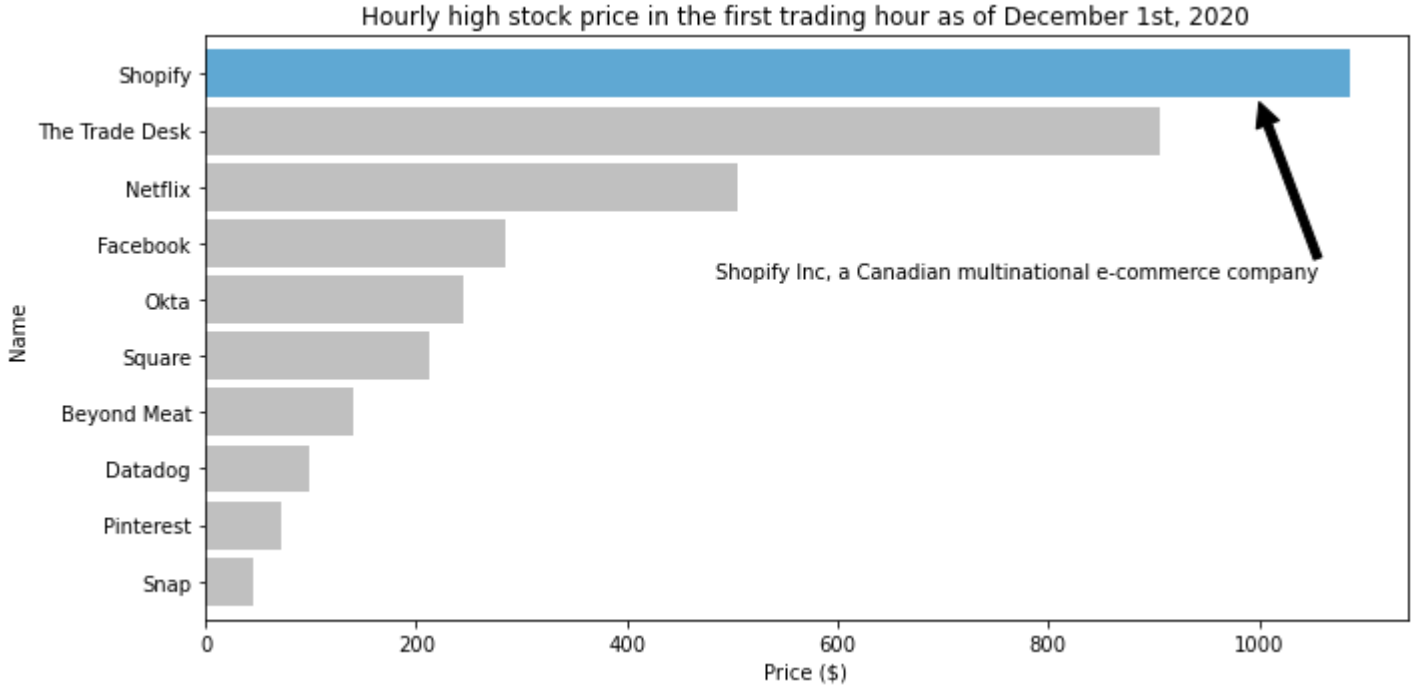
In [9]:
```
df_morning = label_df.loc[label_df.hour == 9]
df_morning = df_morning[['name', 'max_high']]
sorted_morning_df = df_morning.sort_values('max_high', ascending=False)
sorted_morning_df
```

Out[9]:

	name	max_high
43	Shopify	1086.000000
64	The Trade Desk	905.000000
22	Netflix	505.193115
15	Facebook	284.929993
29	Okta	244.270004
57	Square	212.949997
0	Beyond Meat	140.910004
8	Datadog	98.800003
36	Pinterest	71.099998
50	Snap	45.200001

9. Visualize the hourly high stock price in the first trading hour

In [10]:
```
ax = sorted_morning_df.plot(kind='barh', x='name', y='max_high',
            figsize=(10, 5), zorder=2, width=0.85, \
                 color=['#5fa8d3','silver','silver','silver','silver','silver','silver','silver','silver','si
lver'])

ax.invert_yaxis()
ax.set_xlabel("Price ($)")
ax.set_ylabel("Name")
ax.set_title("Hourly high stock price in the first trading hour as of December 1st, 2020")
ax.annotate('Shopify Inc, a Canadian multinational e-commerce company',
            xy=(1000, 0.5), xycoords='data',
            xytext=(30, -90), textcoords='offset points',
            arrowprops=dict(facecolor='black'),
            horizontalalignment='right', verticalalignment='bottom')
ax.get_legend().remove()

plt.tight_layout()
plt.show()
```



10. Prepare data for the third chart

In [11]:
```
df_from10 = label_df.loc[label_df.hour > 9]
df_from10['minute'] = df_from10['ts'].str[14:16].astype(int)
df_from10['Hourly High'] = np.where(df_from10['minute']>=30, 'Second Half Hour', 'First Half Hour')
df_from10
```

```
<ipython-input-11-62e5ff26b681>:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#r
eturning-a-view-versus-a-copy

<ipython-input-11-62e5ff26b681>:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#r
eturning-a-view-versus-a-copy
```

Out[11]:

	name	max_high	ts	hour	minute	Hourly High
1	Beyond Meat	139.154999	2020-12-01 10:00:00-05:00	10	0	First Half Hour
2	Beyond Meat	138.500000	2020-12-01 11:25:00-05:00	11	25	First Half Hour
3	Beyond Meat	138.720001	2020-12-01 12:30:00-05:00	12	30	Second Half Hour
4	Beyond Meat	138.880005	2020-12-01 13:40:00-05:00	13	40	Second Half Hour
5	Beyond Meat	138.880005	2020-12-01 13:45:00-05:00	13	45	Second Half Hour
...	...	...	...	...	...	...
67	The Trade Desk	885.780029	2020-12-01 11:15:00-05:00	11	15	First Half Hour
68	The Trade Desk	883.440002	2020-12-01 12:00:00-05:00	12	0	First Half Hour
69	The Trade Desk	892.000000	2020-12-01 13:50:00-05:00	13	50	Second Half Hour
70	The Trade Desk	887.530029	2020-12-01 14:25:00-05:00	14	25	First Half Hour
71	The Trade Desk	893.159973	2020-12-01 15:55:00-05:00	15	55	Second Half Hour

62 rows × 6 columns

11. Visualize which half an hour each hourly high stock price occurred

In [12]:
```
firstHalfCount = (df_from10['Hourly High']=='First Half Hour').sum()
secondHalfCount = (df_from10['Hourly High']=='Second Half Hour').sum()
labels = ["First Half Hour", "Second Half Hour"]
colors = ["#f7e3af", "#5fa8d3"]
explode = (0, 0.1)
plt.pie([firstHalfCount,secondHalfCount], labels = labels, \
        colors = colors, autopct="%.2f %%", pctdistance=0.8, explode=explode)
plt.title("62.9% of hourly high price occurred in the first half hour")
plt.show()
```