Visualizations on Results Data

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Read the data

In [3]:
```
df = pd.read_csv('/Users/haiqiong/Desktop/project03/results.csv')
df
```

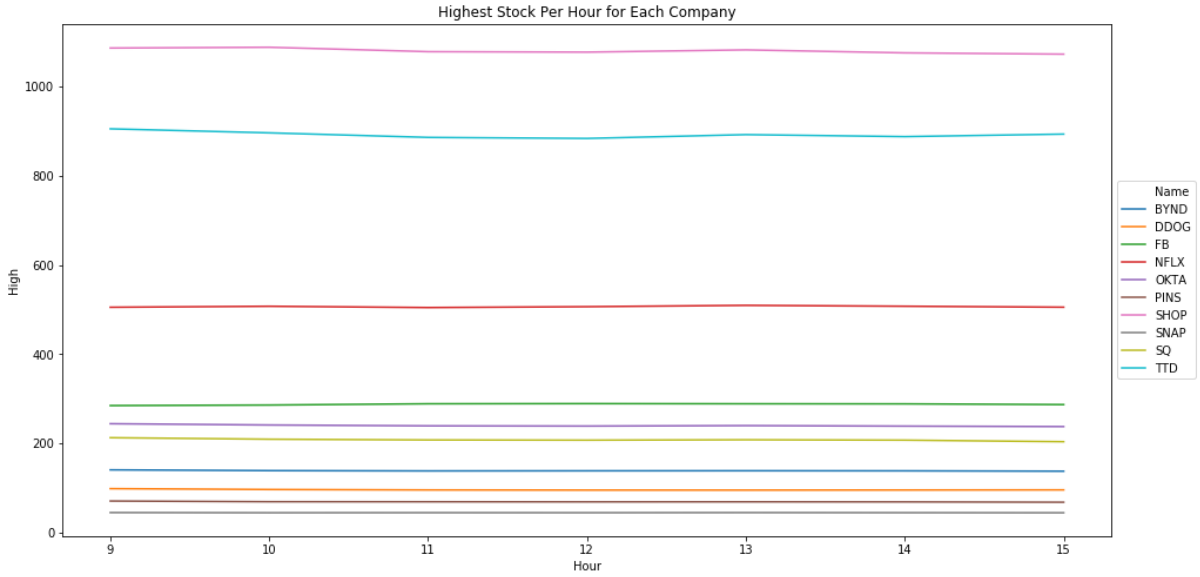
Out[3]:

	Name	High	Hour	Time
0	BYND	140.91	9	12/01/2020 09:40:00
1	BYND	139.15	10	12/01/2020 10:00:00
2	BYND	138.50	11	12/01/2020 11:25:00
3	BYND	138.72	12	12/01/2020 12:30:00
4	BYND	138.88	13	12/01/2020 13:40:00
...	...	...	...	...
65	TTD	885.78	11	12/01/2020 11:10:00
66	TTD	883.44	12	12/01/2020 12:00:00
67	TTD	892.00	13	12/01/2020 13:50:00
68	TTD	887.53	14	12/01/2020 14:25:00
69	TTD	893.16	15	12/01/2020 15:55:00

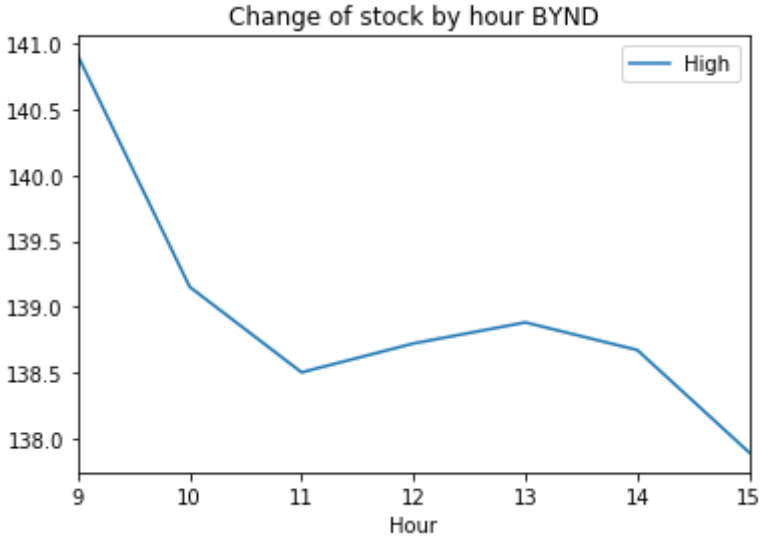
70 rows × 4 columns

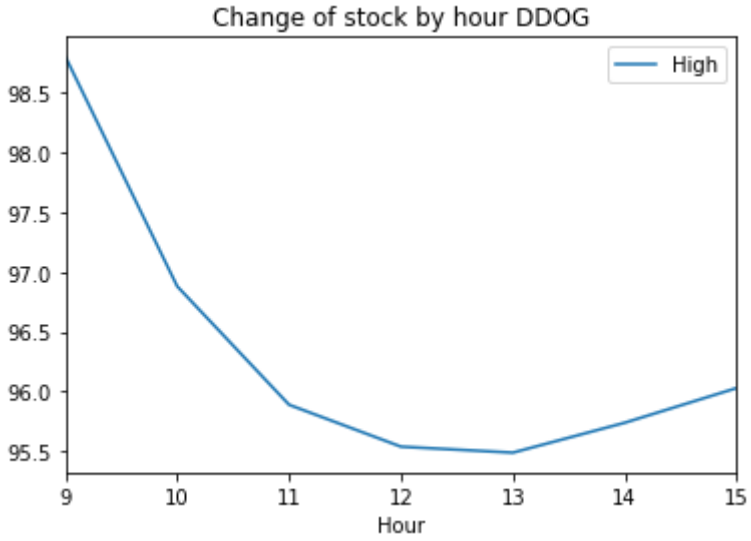
Visualizations

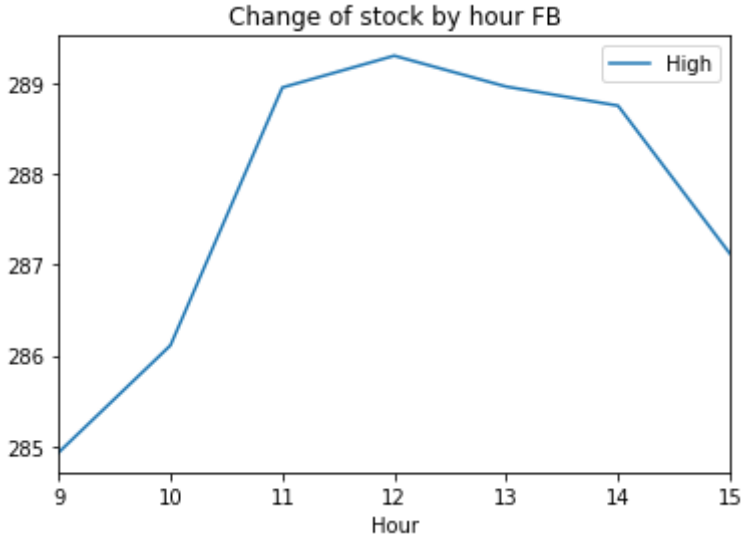
In [4]:
```
## 1. Line plot for all stocks
## With same x-axis and same y-axis

plt.figure(figsize=(16,8))
plt.title('Highest Stock Per Hour for Each Company')
sns.lineplot(x='Hour',y='High',hue='Name',data=df)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5));
```

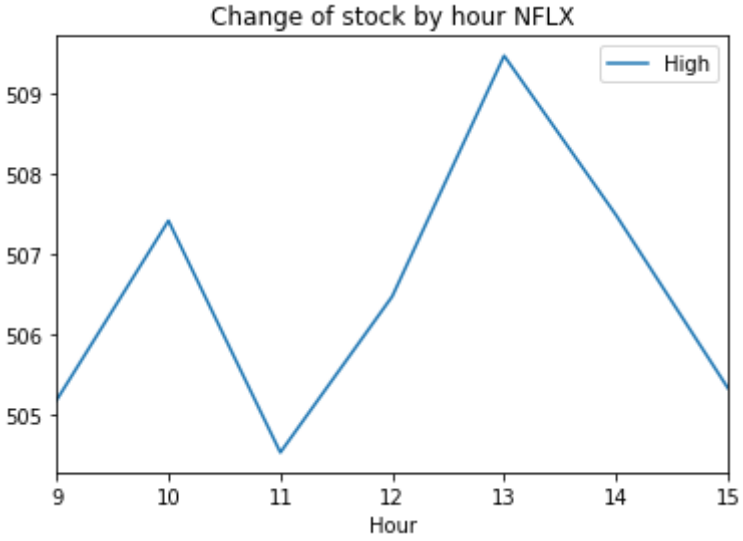


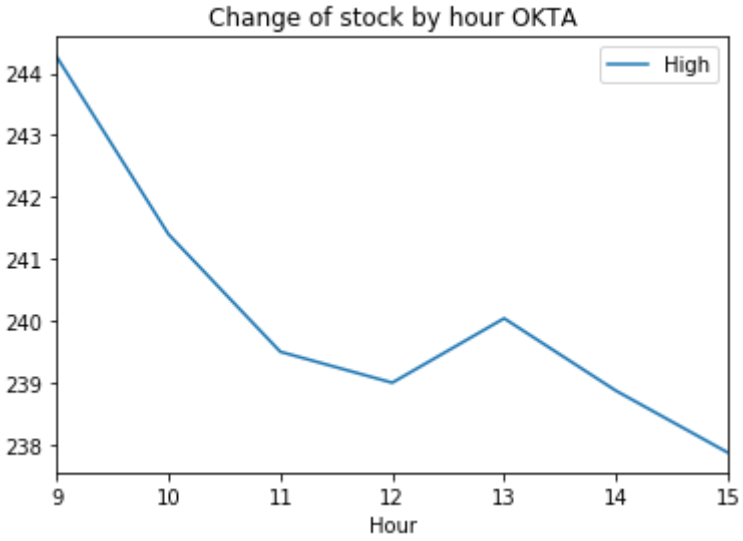
In [5]:
```
## 2. Line plot for all stocks by hour on 2020/12/01
## With same x-axis and different y-axis

df = df[['Name','High','Hour']]

for i in df['Name'].unique():
    plot = df[df['Name'] == i]
    plot.set_index('Hour').plot(kind='line')
    plt.title(f'Change of stock by hour {i}')
    plt.show()
```

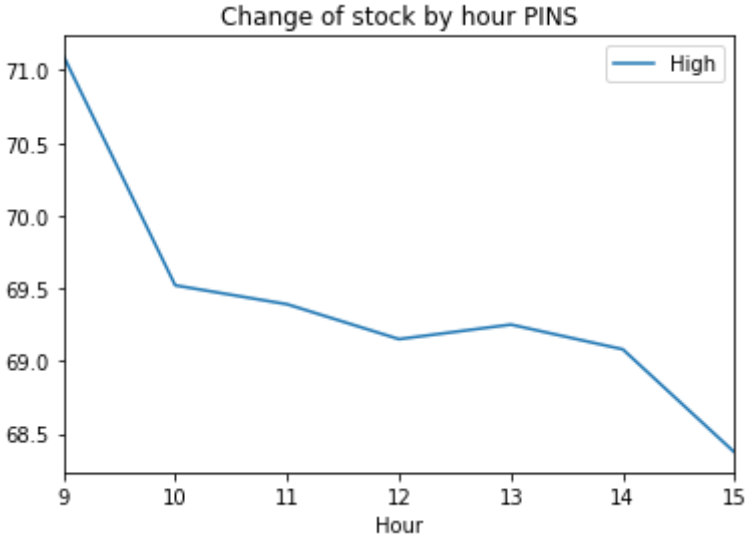

Change of stock by hour BYND
High
141.0
140.5
140.0
139.5
139.0
138.5
138.0
9
10
11
12
13
14
15
Hour

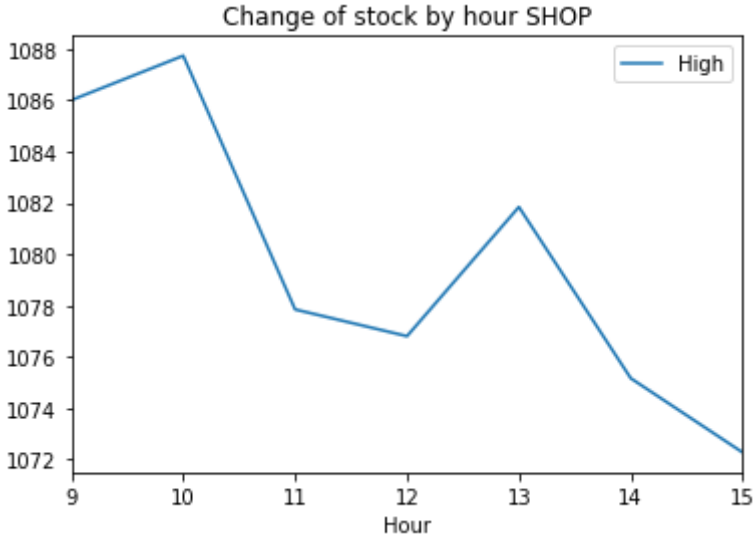

Change of stock by hour DDOG
High
98.5
98.0
97.5
97.0
96.5
96.0
95.5
9
10
11
12
13
14
15
Hour

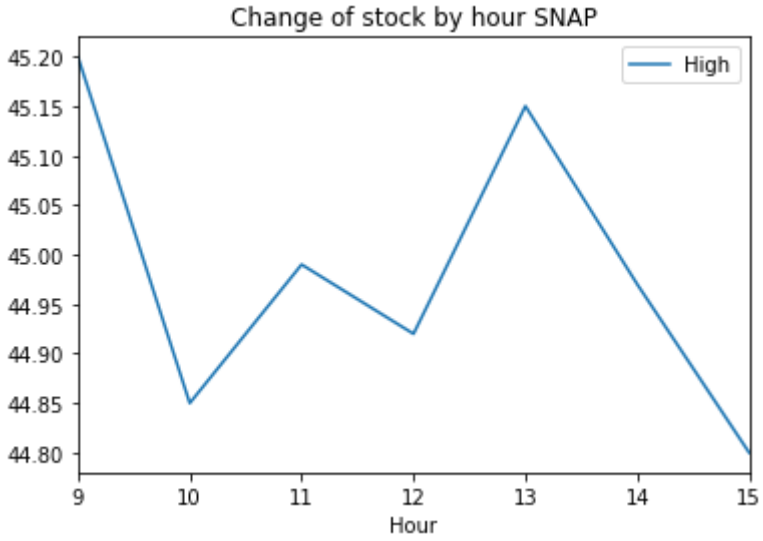

Change of stock by hour FB
High
289
288
287
286
285
9
10
11
12
13
14
15
Hour

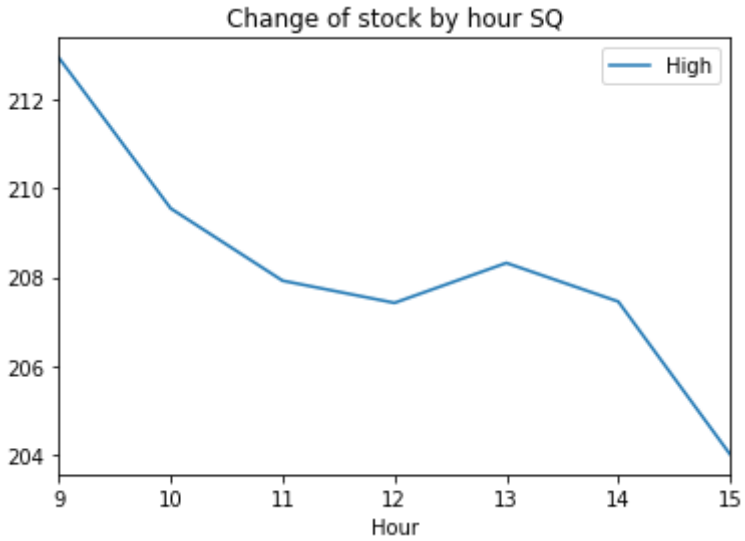

Change of stock by hour NFLX
High
509
508
507
506
505
9
10
11
12
13
14
15
Hour

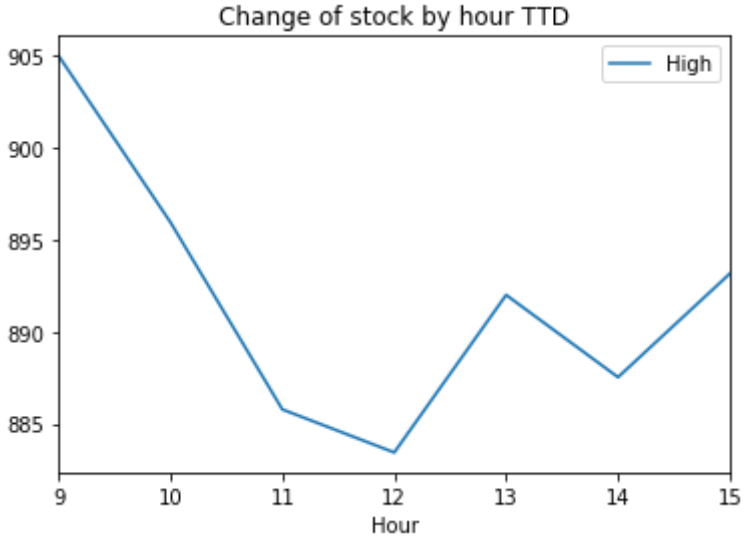

Change of stock by hour OKTA
High
244
243
242
241
240
239
238
9
10
11
12
13
14
15
Hour


Change of stock by hour PINS
High
71.0
70.5
70.0
69.5
69.0
68.5
9
10
11
12
13
14
15
Hour


Change of stock by hour SHOP
High
1088
1086
1084
1082
1080
1078
1076
1074
1072
9
10
11
12
13
14
15
Hour


Change of stock by hour SNAP
High
45.20
45.15
45.10
45.05
45.00
44.95
44.90
44.85
44.80
9
10
11
12
13
14
15
Hour


Change of stock by hour SQ
High
212
210
208
206
204
9
10
11
12
13
14
15
Hour



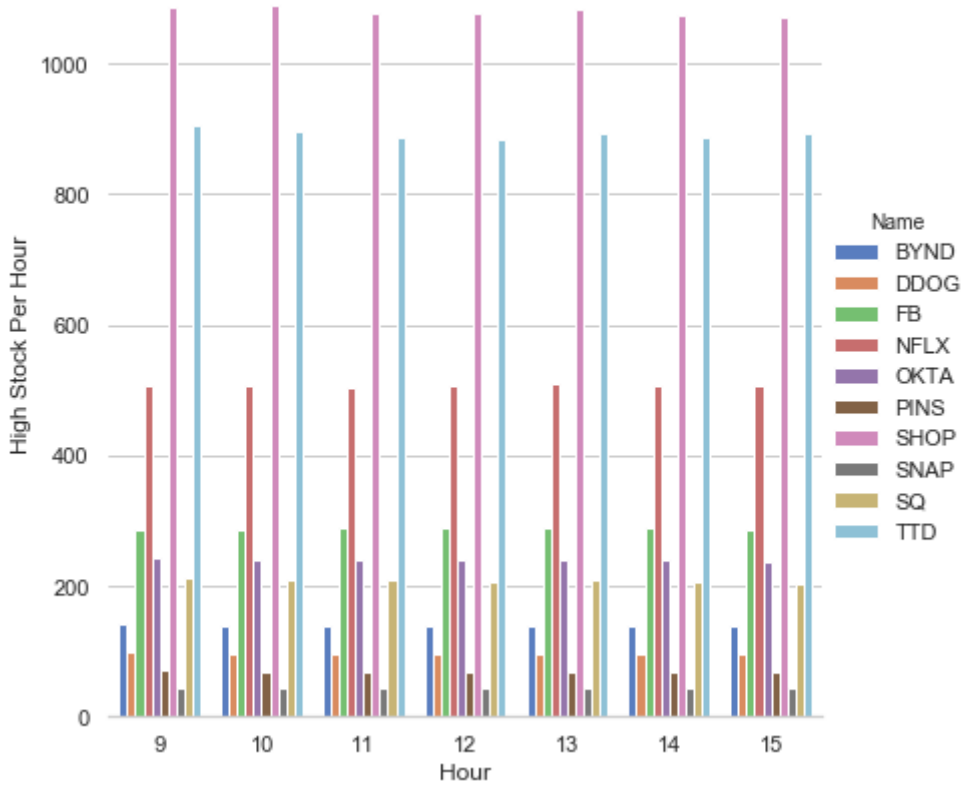
In [7]:
```
## Pivoting the table to group all the data for a bar c

df_pivot = df.pivot_table('High', ['Name'], 'Hour')
df_pivot
```

Out[7]:

Hour	9	10	11	12	13	14	15
Name							
BYND	140.91	139.15	138.50	138.72	138.88	138.67	137.89
DDOG	98.80	96.88	95.89	95.54	95.49	95.74	96.03
FB	284.93	286.11	288.95	289.30	288.96	288.75	287.12
NFLX	505.19	507.42	504.54	506.48	509.47	507.49	505.34
OKTA	244.27	241.40	239.50	239.00	240.04	238.87	237.87
PINS	71.10	69.52	69.39	69.15	69.25	69.08	68.37
SHOP	1086.00	1087.73	1077.85	1076.81	1081.84	1075.17	1072.28
SNAP	45.20	44.85	44.99	44.92	45.15	44.97	44.80
SQ	212.95	209.55	207.92	207.42	208.32	207.45	204.00
TTD	905.00	895.95	885.78	883.44	892.00	887.53	893.16

In [8]:
```
## 4. Grouped bar plot

sns.set(style="whitegrid")

# Draw a nested barplot to show survival for class and sex
g = sns.catplot(x="Hour", y="High", hue="Name", data=df,
                height=6, kind="bar", palette="muted")
g.despine(left=True)
g.set_ylabels("High Stock Per Hour")
```

Out[8]: <seaborn.axisgrid.FacetGrid at 0x1a2067ebd0>



In [ ]:
```
```