analysis

May 15, 2021

1 Analysis of stock prices in different time periods

NOTE: `base date point` will be set separatly for each period.

Example: if we want to get daily prices within a week then each Monday will be set as `base date point`

[1]:
```
import sys

sys.path.append('..')

from analysis import Column
from common import plot, YahooRange

from loguru import logger
import numpy as np
import pandas as pd
from seaborn import lineplot, barplot, scatterplot, boxplot
from matplotlib import pyplot

FILENAME = "sp500/sp500.csv"
LIMIT = None

logger.remove()
logger.add(sys.stdout, level="INFO")

pass
```

1.1 Monthly stock price fluctuations within a year

[2]:
```
from analysis import get_best_month

df = get_best_month(FILENAME, YahooRange.YEARS_10, limit=LIMIT)
df
```

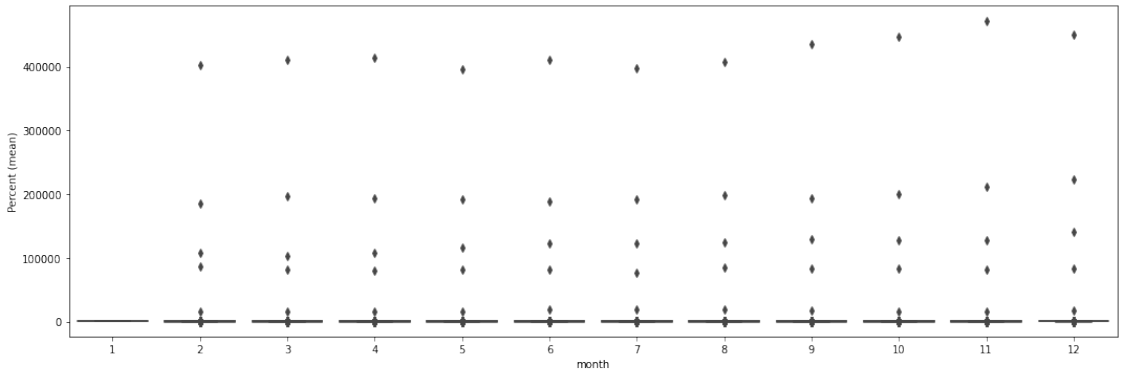
[2]:

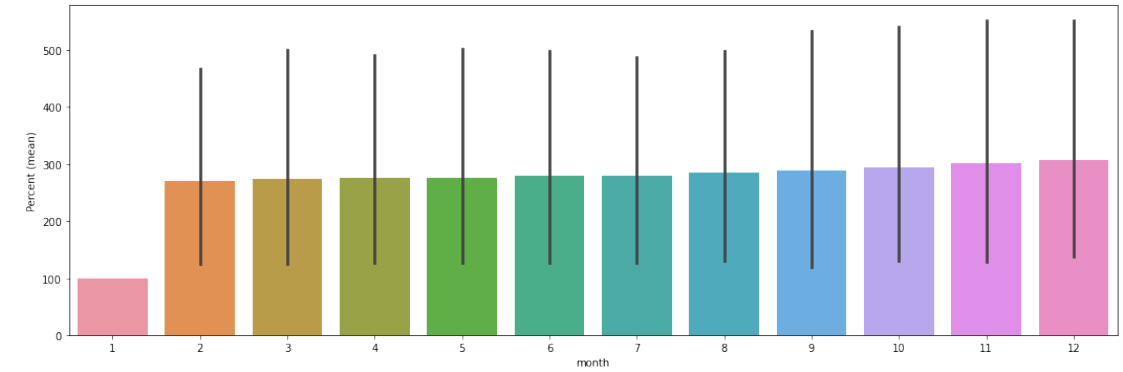
	year	month	Symbol	Percent (mean)
0	2011	1	MHK	100.0
1	2011	2	MHK	97.123051

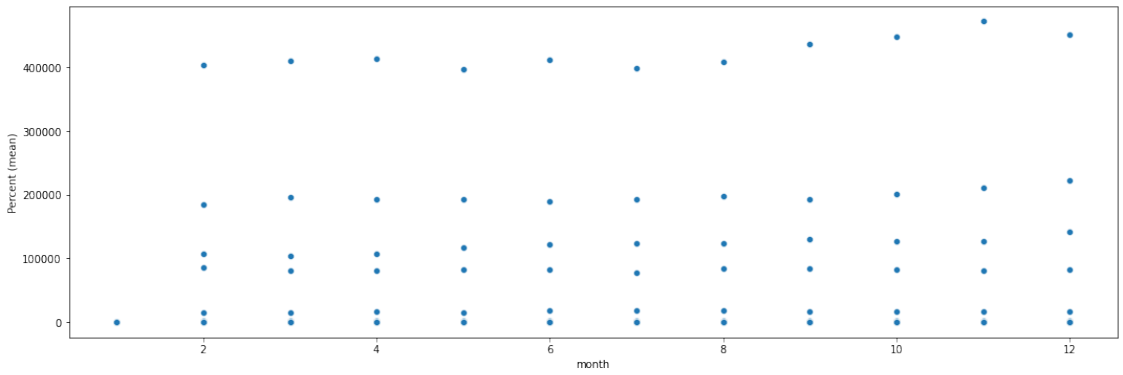
```
2      2011      3    MHK      101.039859
3      2011      4    MHK      107.036391
4      2011      5    MHK      104.592717
…       …        …     …              …
57451  2020      8   EXPD      107.288216
57452  2020      9   EXPD      112.872036
57453  2020     10   EXPD      115.899262
57454  2020     11   EXPD      114.080386
57455  2020     12   EXPD      113.622486

[57456 rows x 4 columns]
```

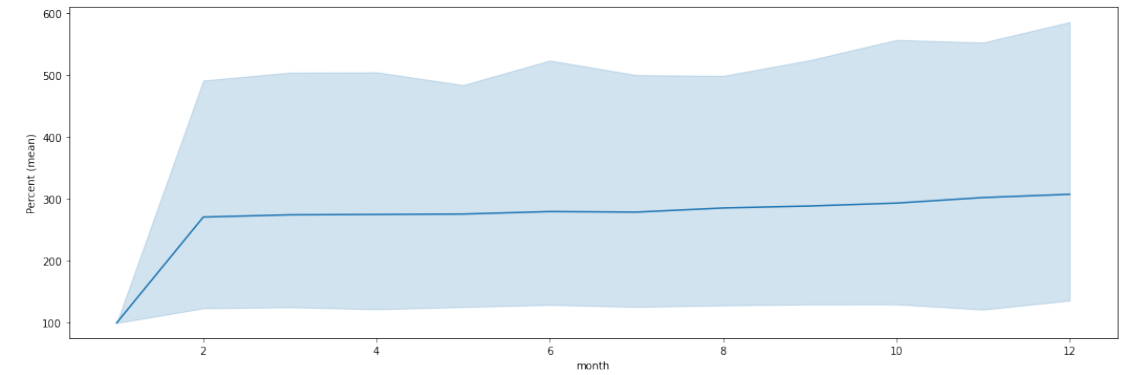
[3]:
```
plot(x=Column.MONTH, y=Column.PERCENT, data=df)
```

```
       Percent (mean)
month
1               100.0
2          270.803607
3          274.391687
4           274.97437
5          275.651566
```


Percent (mean)
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
11
12
month


Percent (mean)
500
400
300
200
100
0
1
2
3
4
5
6
7
8
9
10
11
12
month


Percent (mean)
400000
300000
200000
100000
0
2
4
6
8
10
12
month


Percent (mean)
600
500
400
300
200
100
2
4
6
8
10
12
month

1.2 Weekly stock price fluctuations within a year

[4]:
```
from analysis import get_best_week

df = get_best_week(FILENAME, YahooRange.YEARS_10, limit=LIMIT)

df
```

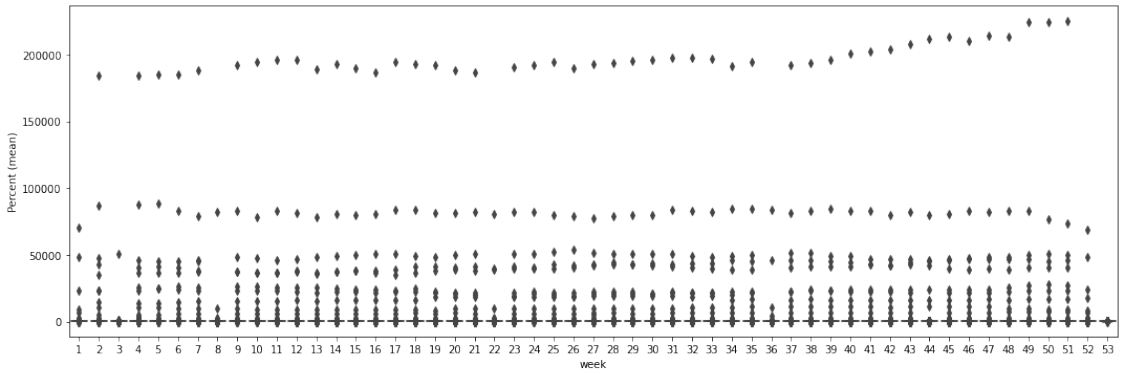
[4]:

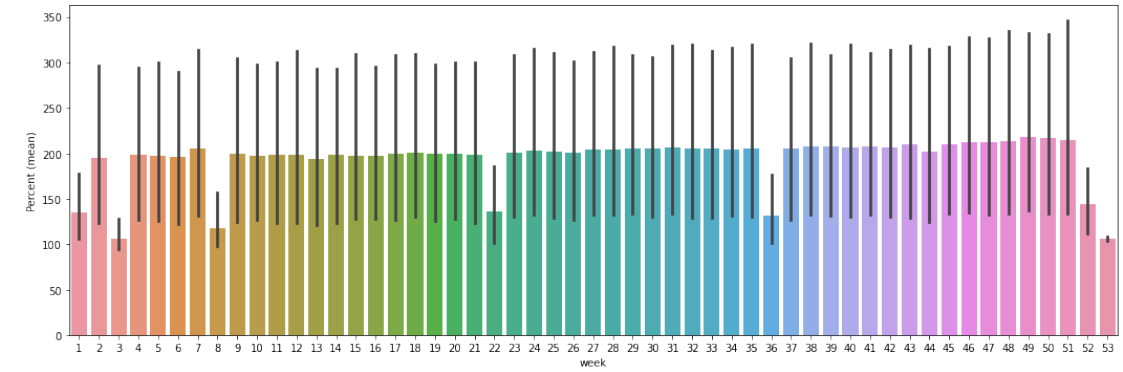
	year	week	Symbol	Percent (mean)
0	2015	2	PAYC	100.0
1	2015	3	PAYC	95.761078
2	2015	4	PAYC	90.558764
3	2015	5	PAYC	104.894021
4	2015	6	PAYC	100.462424
…	…	…	…	…
249504	2020	49	NRG	85.631884
249505	2020	50	NRG	90.508655
249506	2020	51	NRG	88.148924
249507	2020	52	NRG	87.30991
249508	2020	53	NRG	92.842162

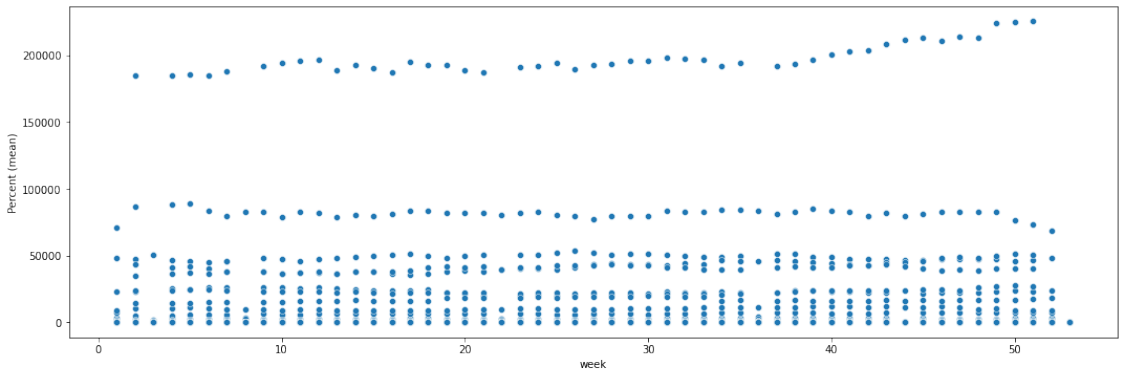
[249509 rows x 4 columns]

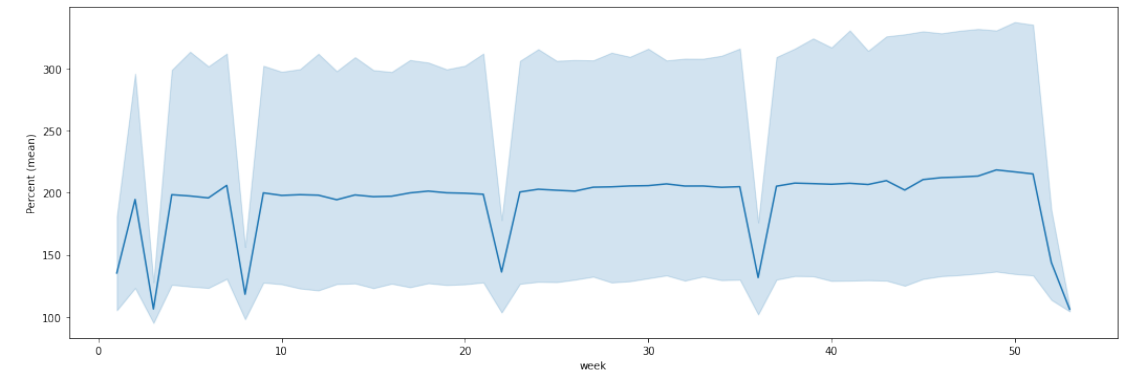
[5]:
```
plot(x=Column.WEEK, y=Column.PERCENT, data=df)
```

week	Percent (mean)
1	135.431788
2	194.639916
3	106.422592
4	198.492734
5	197.438821


Percent (mean)
week


Percent (mean)
week


Percent (mean)
week


Percent (mean)
week

1.3 Daily stock price fluctuations within a month

[6]:
```
from analysis import Column,get_best_month_day

df = get_best_month_day(FILENAME, YahooRange.YEARS_10, limit=LIMIT)

df
```

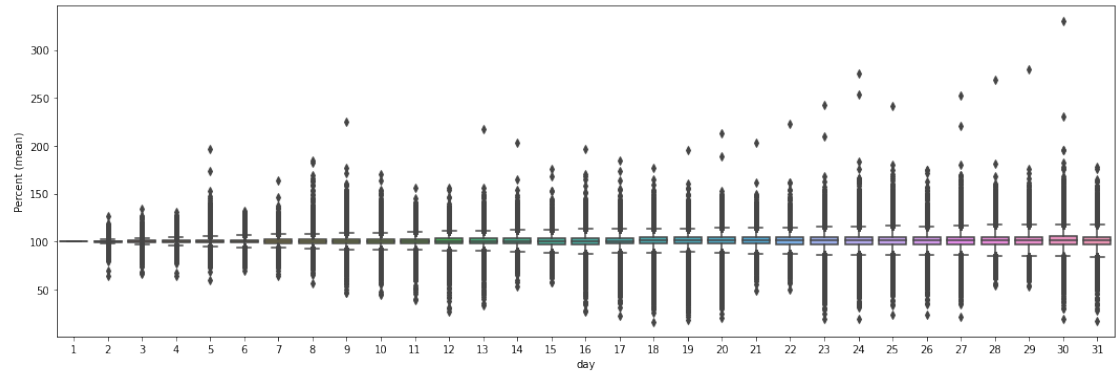
[6]:

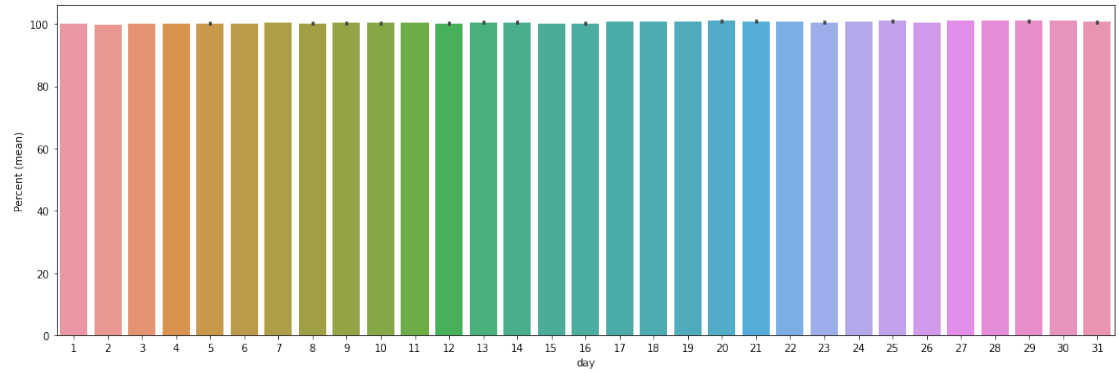
	year	month	day	Symbol	Percent (mean)
0	2011	12	1	EA	100.0
1	2011	12	2	EA	102.07792
2	2011	12	5	EA	101.298698
3	2011	12	6	EA	98.917749
4	2011	12	7	EA	99.65368
…	…	…	…	…	…
1209969	2020	11	23	NUE	109.771652
1209970	2020	11	24	NUE	111.890555
1209971	2020	11	25	NUE	114.420897
1209972	2020	11	27	NUE	113.063152
1209973	2020	11	30	NUE	112.528287

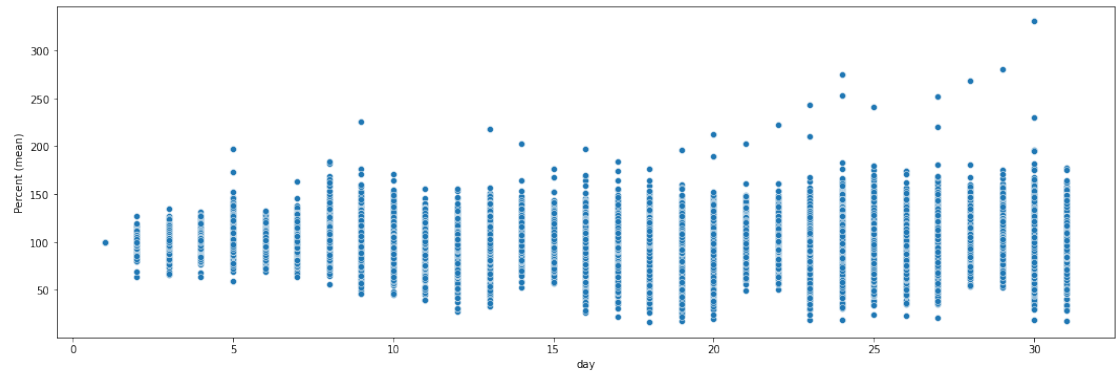
[1209974 rows x 5 columns]

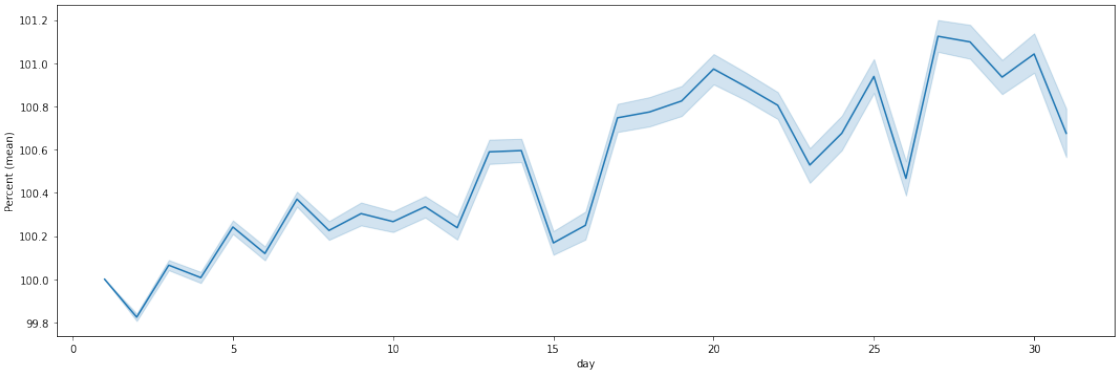
[7]:
```
plot(x=Column.DAY, y=Column.PERCENT, data=df)
```

day	Percent (mean)
1	100.0
2	99.824726
3	100.065176
4	100.007597
5	100.241899


Percent (mean)
day


Percent (mean)
day


Percent (mean)
day


Percent (mean)
day

1.4 Daily stock price fluctuations within a week

[8]:
```
from analysis import get_best_weekday


df = get_best_weekday(FILENAME, YahooRange.YEARS_10, limit=LIMIT)

df
```

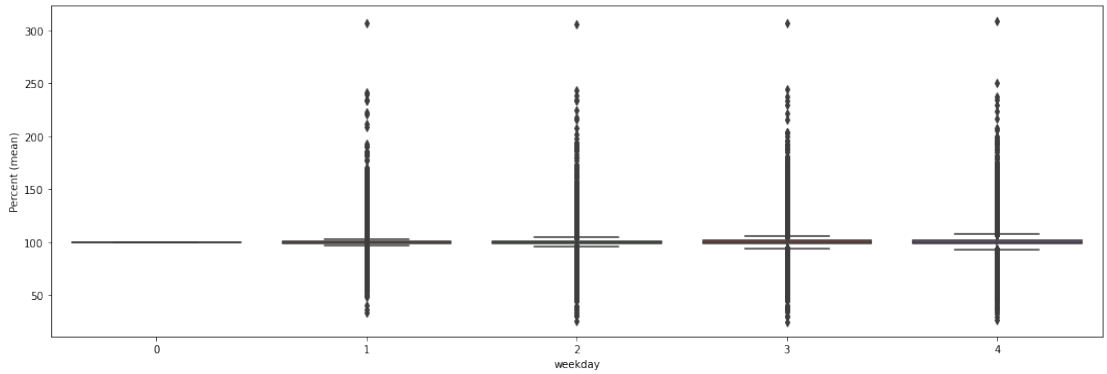
[8]:

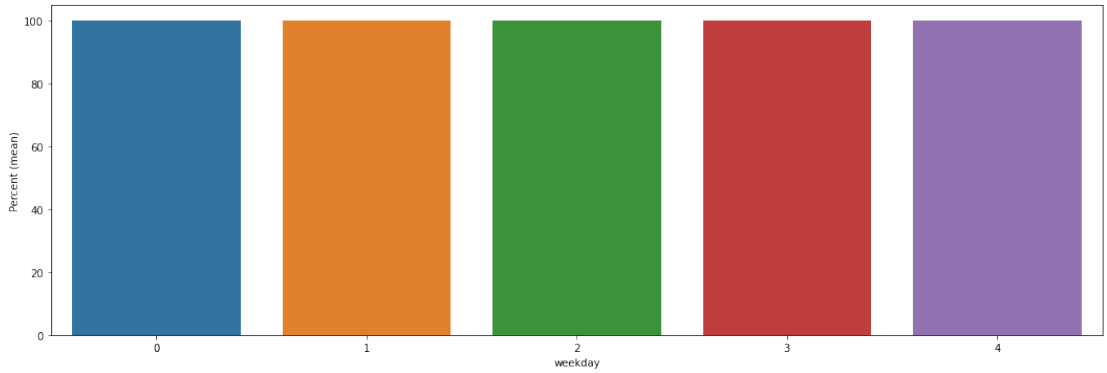
	year	week	weekday	Symbol	Percent (mean)
0	2011	52	1	EA	100.0
1	2011	52	2	EA	101.05719
2	2011	52	3	EA	98.942819
3	2011	52	4	EA	100.19222
4	2011	1	0	EA	100.0
…	…	…	…	…	…
1208795	2020	51	4	NUE	96.704447
1208796	2020	53	0	NUE	100.0
1208797	2020	53	1	NUE	99.258556
1208798	2020	53	2	NUE	98.498104
1208799	2020	53	3	NUE	100.665404

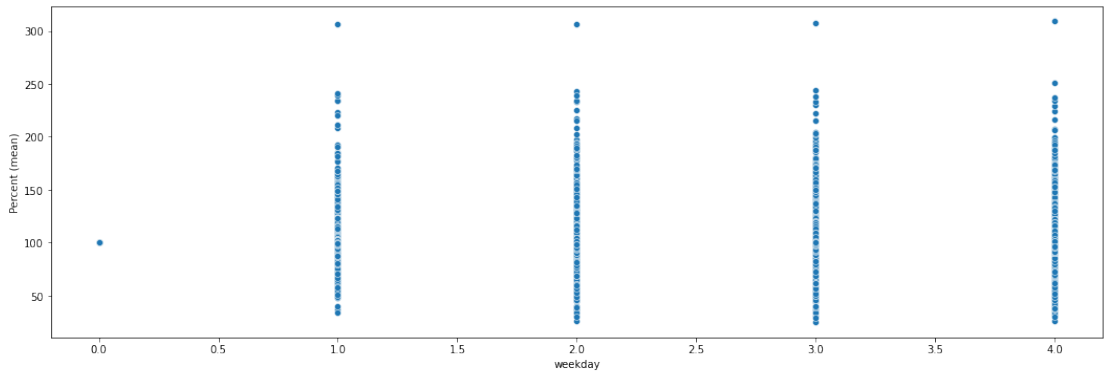
[1208800 rows x 5 columns]

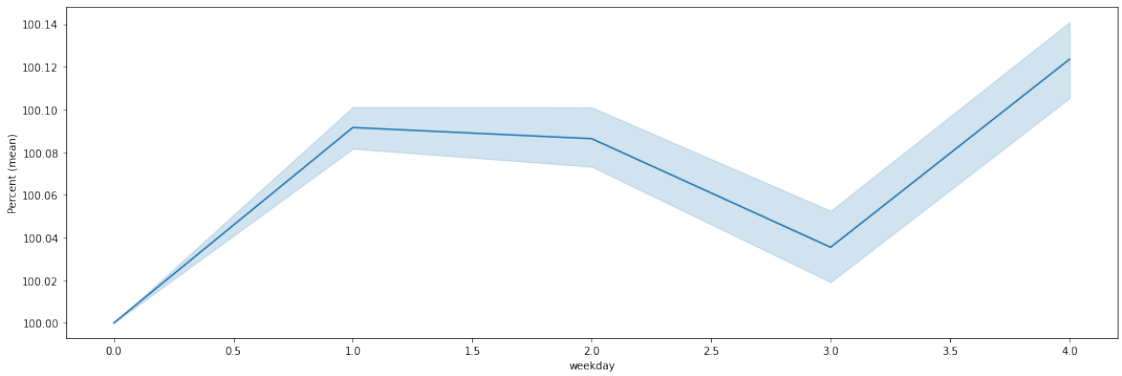
[9]:
```
plot(x=Column.WEEKDAY, y=Column.PERCENT, data=df)
```

weekday	Percent (mean)
0	100.0
1	100.091575
2	100.086351
3	100.035453
4	100.123565


Percent (mean)
300
250
200
150
100
50
0
1
2
3
4
weekday


Percent (mean)
100
80
60
40
20
0
0
1
2
3
4
weekday


Percent (mean)
300
250
200
150
100
50
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
weekday


Percent (mean)
100.14
100.12
100.10
100.08
100.06
100.04
100.02
100.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
weekday

1.5 Hourly stock price fluctuations with a day

[10]:
```
from analysis import get_best_hour

df = get_best_hour(FILENAME, YahooRange.YEARS_2, limit=LIMIT)

df
```

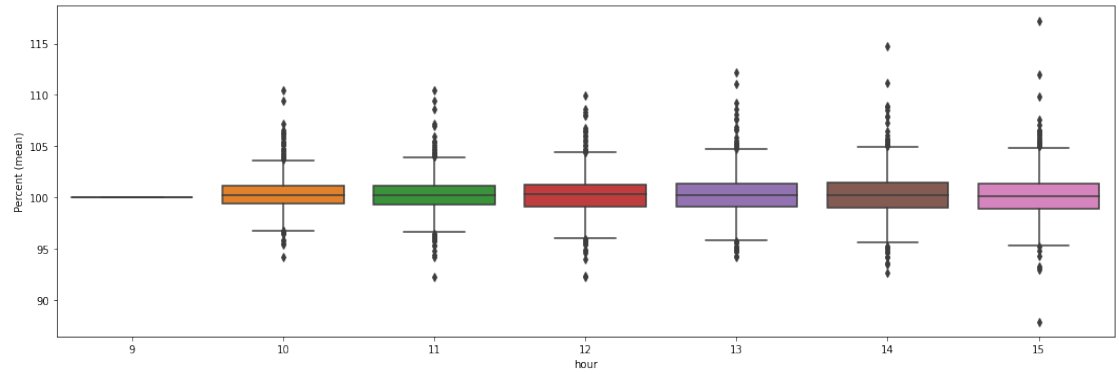
[10]:

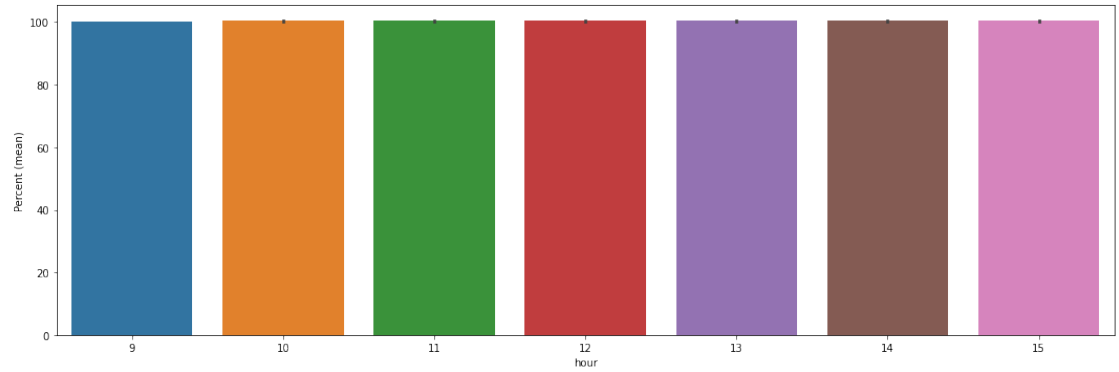
	year	week	day	hour	Symbol	Percent (mean)
0	2020	14	3	9	OTIS	100.0
1	2020	14	3	10	OTIS	99.065634
2	2020	14	3	11	OTIS	104.238834
3	2020	14	3	12	OTIS	106.494983
4	2020	14	3	13	OTIS	111.109845
…	…	…	…	…	…	…
5363	2020	53	31	11	CARR	99.308883
5364	2020	53	31	12	CARR	99.016483
5365	2020	53	31	13	CARR	99.33546
5366	2020	53	31	14	CARR	99.33546
5367	2020	53	31	15	CARR	100.119623

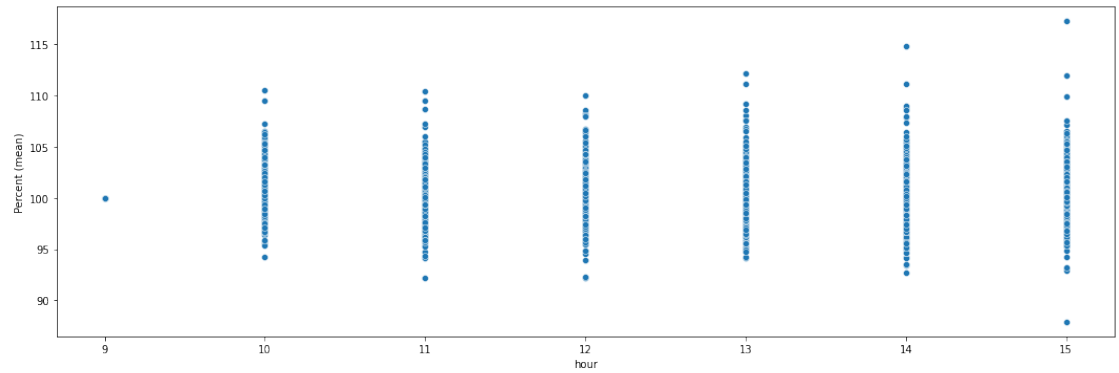
[5368 rows x 6 columns]

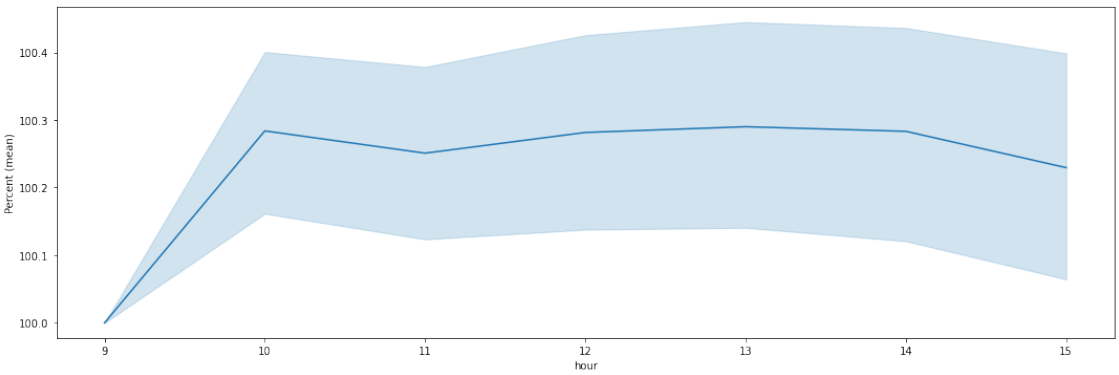
[11]:
```
plot(x=Column.HOUR, y=Column.PERCENT, data=df)
```

hour	Percent (mean)
9	100.0
10	100.28387
11	100.250948
12	100.281567
13	100.290068


Percent (mean)
115
110
105
100
95
90
9
10
11
12
13
14
15
hour


Percent (mean)
100
80
60
40
20
0
9
10
11
12
13
14
15
hour


Percent (mean)
115
110
105
100
95
90
9
10
11
12
13
14
15
hour


Percent (mean)
100.4
100.3
100.2
100.1
100.0
9
10
11
12
13
14
15
hour

1.6 Hourly and quarterly stock price fluctuations within a day

[12]:
```
from analysis import get_best_time

df = get_best_time(FILENAME, YahooRange.DAYS_58, limit=LIMIT)

df
```

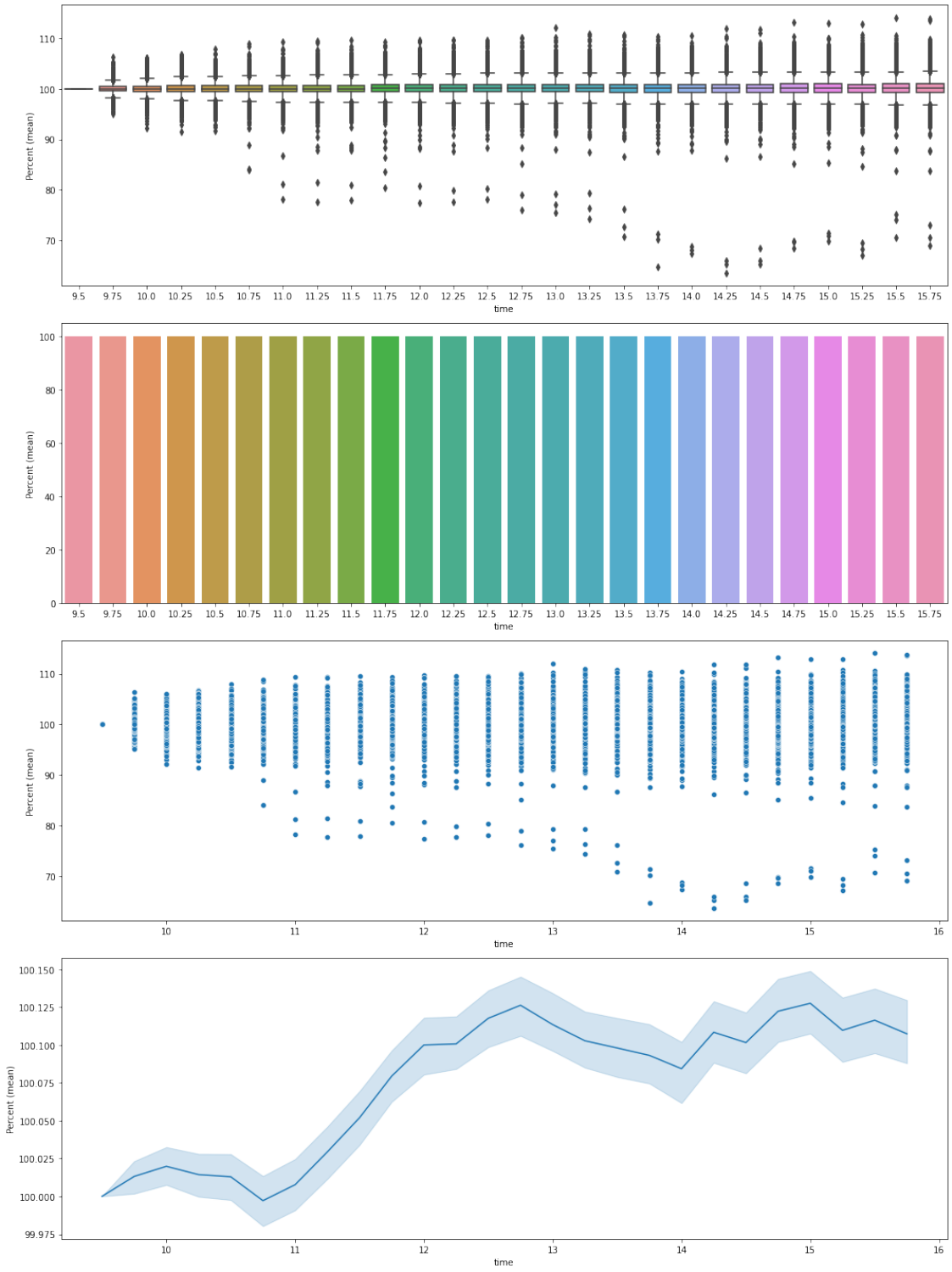
[12]:

	year	week	day	hour	minute	time	Symbol	Percent (mean)
0	2021	11	17	9	30	9.5	GPN	100.0
1	2021	11	17	9	45	9.75	GPN	100.050962
2	2021	11	17	10	0	10.0	GPN	99.902694
3	2021	11	17	10	15	10.25	GPN	99.888794
4	2021	11	17	10	30	10.5	GPN	99.990731
…	…	…	…	…	…	…	…	…
511660	2021	19	11	14	45	14.75	RHI	99.488606
511661	2021	19	11	15	0	15.0	RHI	99.577546
511662	2021	19	11	15	15	15.25	RHI	99.466375
511663	2021	19	11	15	30	15.5	RHI	99.822128
511664	2021	19	11	15	45	15.75	RHI	99.399674

[511665 rows x 8 columns]

[13]:
```
plot(x=Column.TIME, y=Column.PERCENT, data=df)
```

time	Percent (mean)
9.50	100.0
9.75	100.013161
10.00	100.019854
10.25	100.014358
10.50	100.012946


Percent (mean)
time
9.5 9.75 10.0 10.25 10.5 10.75 11.0 11.25 11.5 11.75 12.0 12.25 12.5 12.75 13.0 13.25 13.5 13.75 14.0 14.25 14.5 14.75 15.0 15.25 15.5 15.75
10 11 12 13 14 15 16
100.150 100.125 100.100 100.075 100.050 100.025 100.000 99.975

1.7 Quarterly stock price fluctuations within an hour

[14]:
```
from analysis import get_best_quarter

df = get_best_quarter(FILENAME, YahooRange.DAYS_58, limit=LIMIT)

df
```

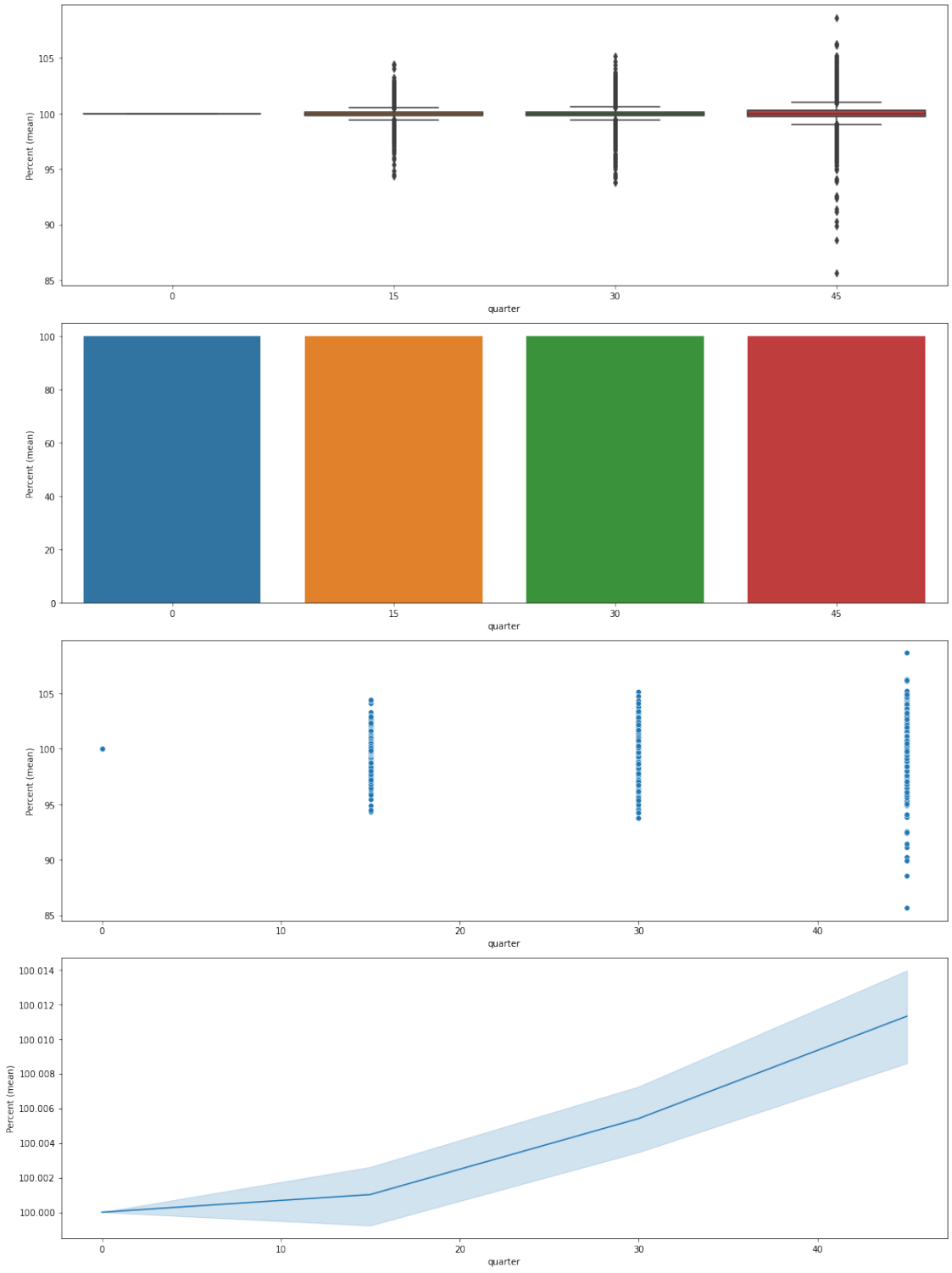
[14]:

	year	week	day	hour	minute	quarter	Symbol	Percent (mean)
0	2021	11	17	9	30	30	GPN	100.0
1	2021	11	17	9	45	45	GPN	100.050962
2	2021	11	17	10	0	0	GPN	100.0
3	2021	11	17	10	15	15	GPN	99.986087
4	2021	11	17	10	30	30	GPN	100.088123
…	…	…	…	…	…	…	…	…
511266	2021	19	11	14	45	45	RHI	99.92184
511267	2021	19	11	15	0	0	RHI	100.0
511268	2021	19	11	15	15	15	RHI	99.888357
511269	2021	19	11	15	30	30	RHI	100.245619
511270	2021	19	11	15	45	45	RHI	99.821373

[511271 rows x 8 columns]

[15]:
```
plot(x=Column.QUARTER, y=Column.PERCENT, data=df)
```

quarter	Percent (mean)
0	100.0
15	100.001017
30	100.005404
45	100.011317


Percent (mean)
quarter
0
15
30
45
10
20
40
85
90
95
100
105
0
20
40
60
80
100
100.000
100.002
100.004
100.006
100.008
100.010
100.012
100.014

[ ]: