

analysis

May 15, 2021

1 Analysis of stock prices in different time periods

NOTE: base date point will be set separately 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

pd.options.mode.chained_assignment = None

FILENAME = "dax/dax_mdax_sdax.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
```

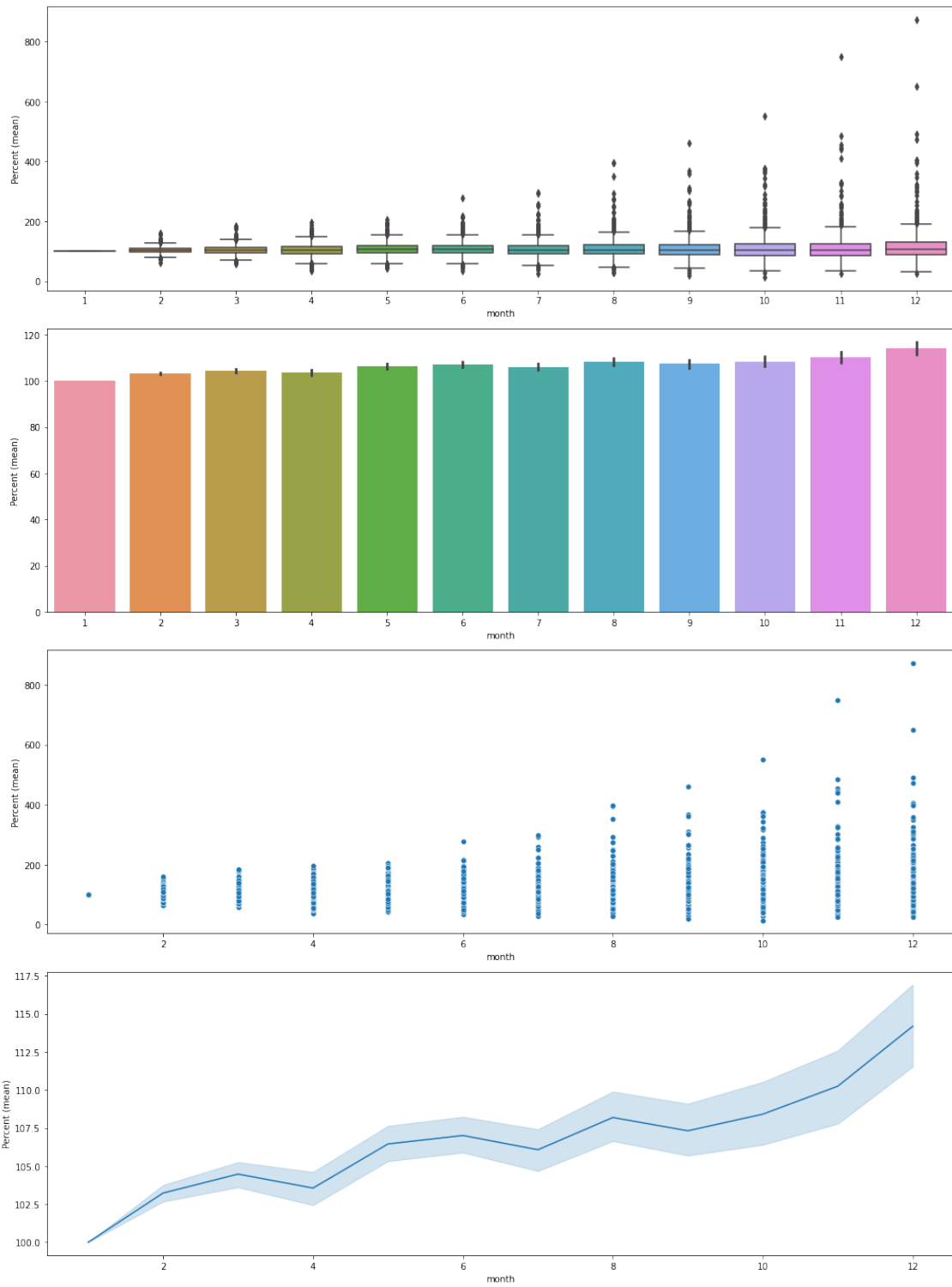
```
[2]:
```

	year	month	Symbol	Percent (mean)
0	2011	1	BVB.DE	100.0
1	2011	2	BVB.DE	114.45313
2	2011	3	BVB.DE	117.578127
3	2011	4	BVB.DE	107.812502
4	2011	5	BVB.DE	121.054689
...
15787	2020	8	PAT.DE	116.565956
15788	2020	9	PAT.DE	124.874111
15789	2020	10	PAT.DE	118.328294
15790	2020	11	PAT.DE	98.690835
15791	2020	12	PAT.DE	121.349444

[15792 rows x 4 columns]

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

	Percent (mean)
month	
1	100.0
2	103.235071
3	104.479022
4	103.561544
5	106.458575



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
```

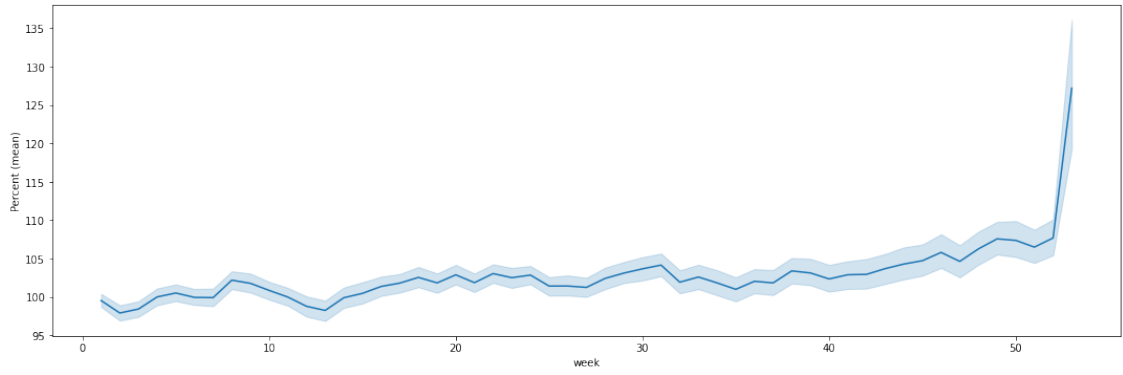
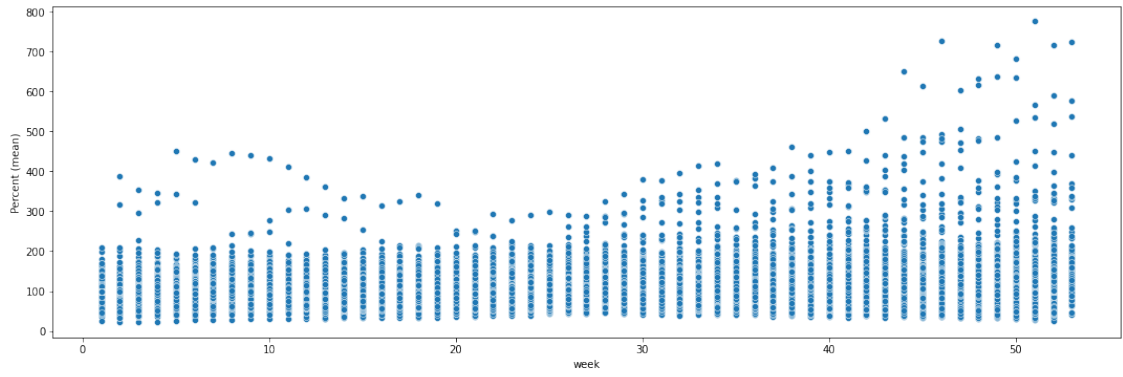
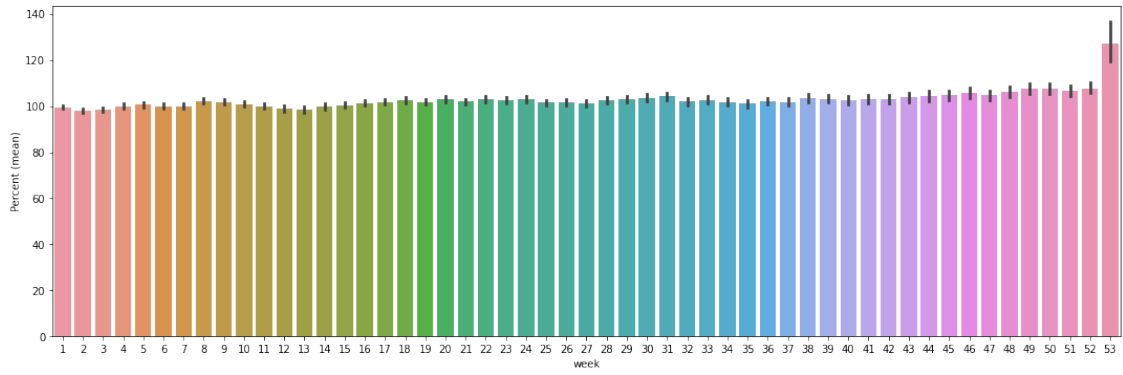
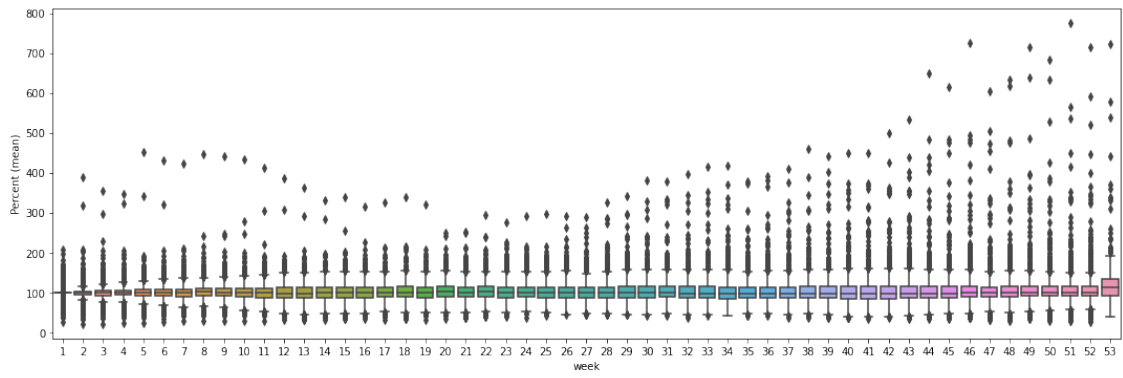
```
[4]:
```

	year	week	Symbol	Percent (mean)
0	2011	1	AFX.DE	100.0
1	2011	2	AFX.DE	99.441536
2	2011	3	AFX.DE	96.684123
3	2011	4	AFX.DE	97.033157
4	2011	5	AFX.DE	98.568936
...
67898	2020	49	S92.DE	140.763964
67899	2020	50	S92.DE	134.948682
67900	2020	51	S92.DE	145.524506
67901	2020	52	S92.DE	152.651073
67902	2020	53	S92.DE	160.205239

[67903 rows x 4 columns]

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

	Percent (mean)
week	
1	99.541714
2	97.919694
3	98.435284
4	100.015221
5	100.526573



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
```

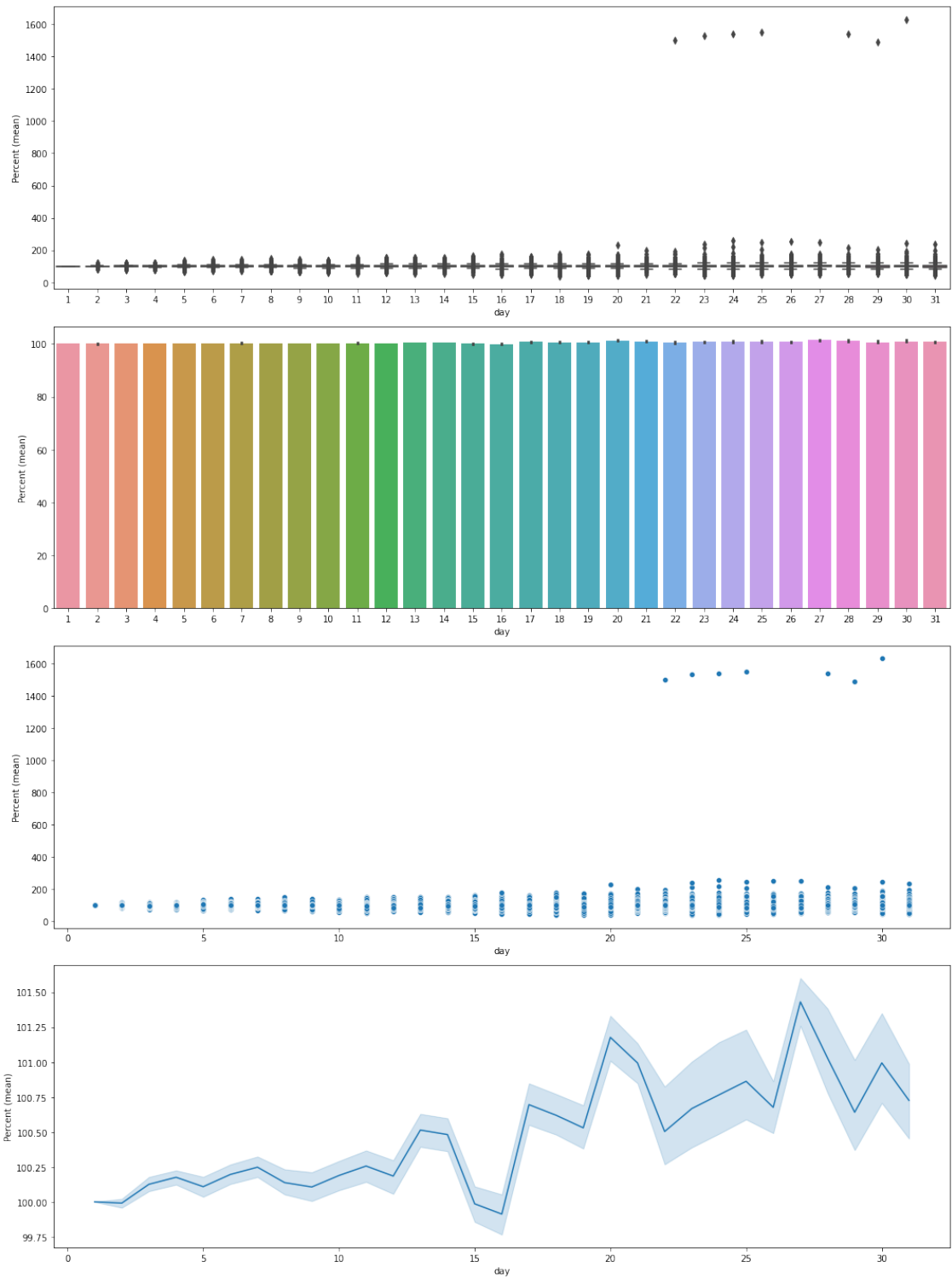
```
[6]:
```

	year	month	day	Symbol	Percent (mean)
0	2011	1	3	HYQ.DE	100.0
1	2011	1	4	HYQ.DE	100.308546
2	2011	1	5	HYQ.DE	102.989681
3	2011	1	6	HYQ.DE	102.989681
4	2011	1	7	HYQ.DE	108.628576
...
332648	2020	12	22	SBS.DE	103.436426
332649	2020	12	23	SBS.DE	103.951889
332650	2020	12	28	SBS.DE	104.123707
332651	2020	12	29	SBS.DE	104.467351
332652	2020	12	30	SBS.DE	104.123707

[332653 rows x 5 columns]

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

```
Percent (mean)
day
1          100.0
2      99.990816
3      100.125494
4      100.175699
5      100.108565
```



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
```

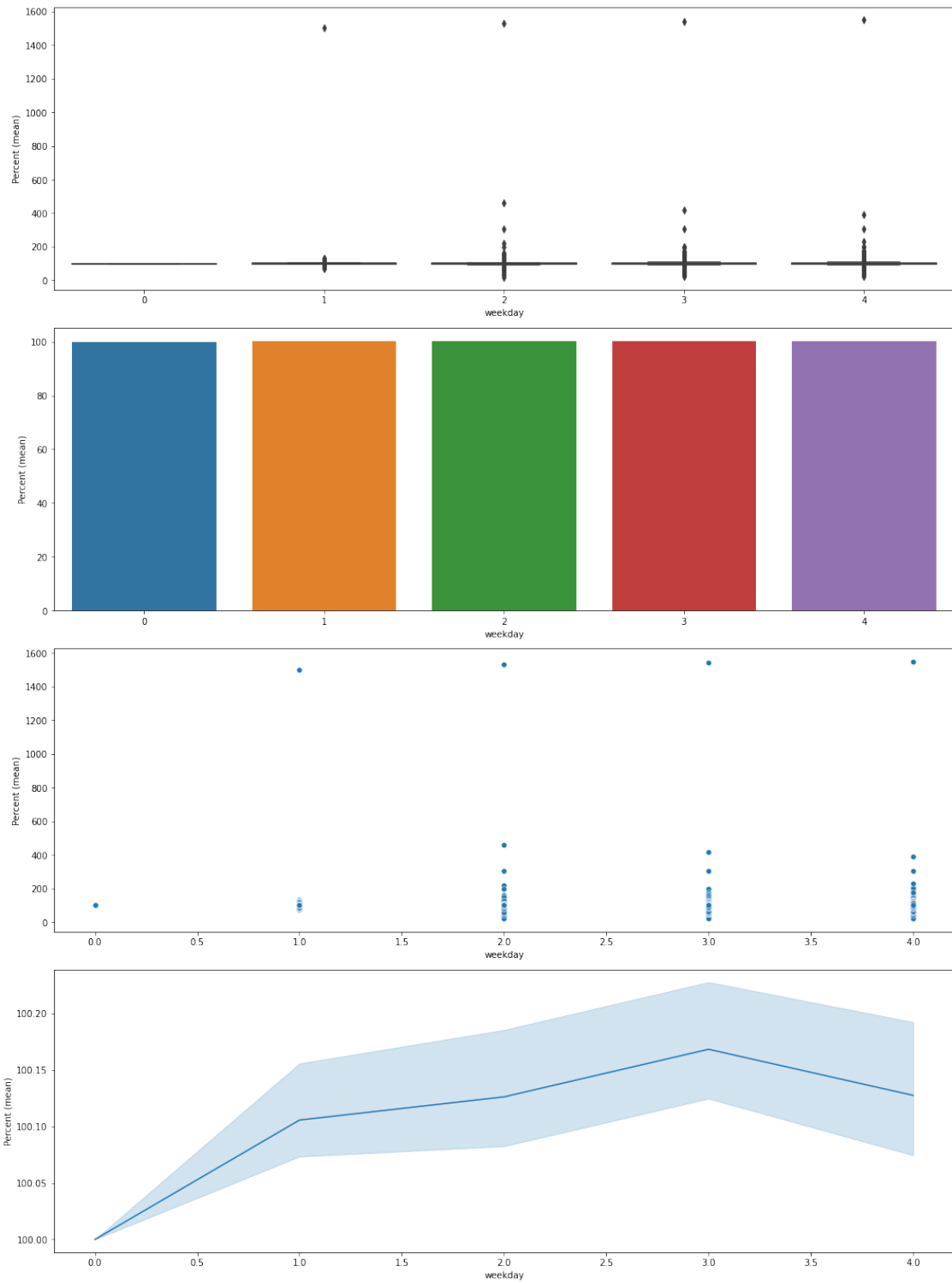
```
[8]:
```

	year	week	weekday	Symbol	Percent (mean)
0	2011	1	0	HYQ.DE	100.0
1	2011	1	1	HYQ.DE	100.308546
2	2011	1	2	HYQ.DE	102.989681
3	2011	1	3	HYQ.DE	102.989681
4	2011	1	4	HYQ.DE	108.628576
...
335762	2020	52	1	SBS.DE	102.555366
335763	2020	52	2	SBS.DE	103.066438
335764	2020	53	0	SBS.DE	100.0
335765	2020	53	1	SBS.DE	100.330034
335766	2020	53	2	SBS.DE	100.0

```
[335767 rows x 5 columns]
```

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

	Percent (mean)
weekday	
0	100.0
1	100.105653
2	100.126205
3	100.16834
4	100.127546



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
```

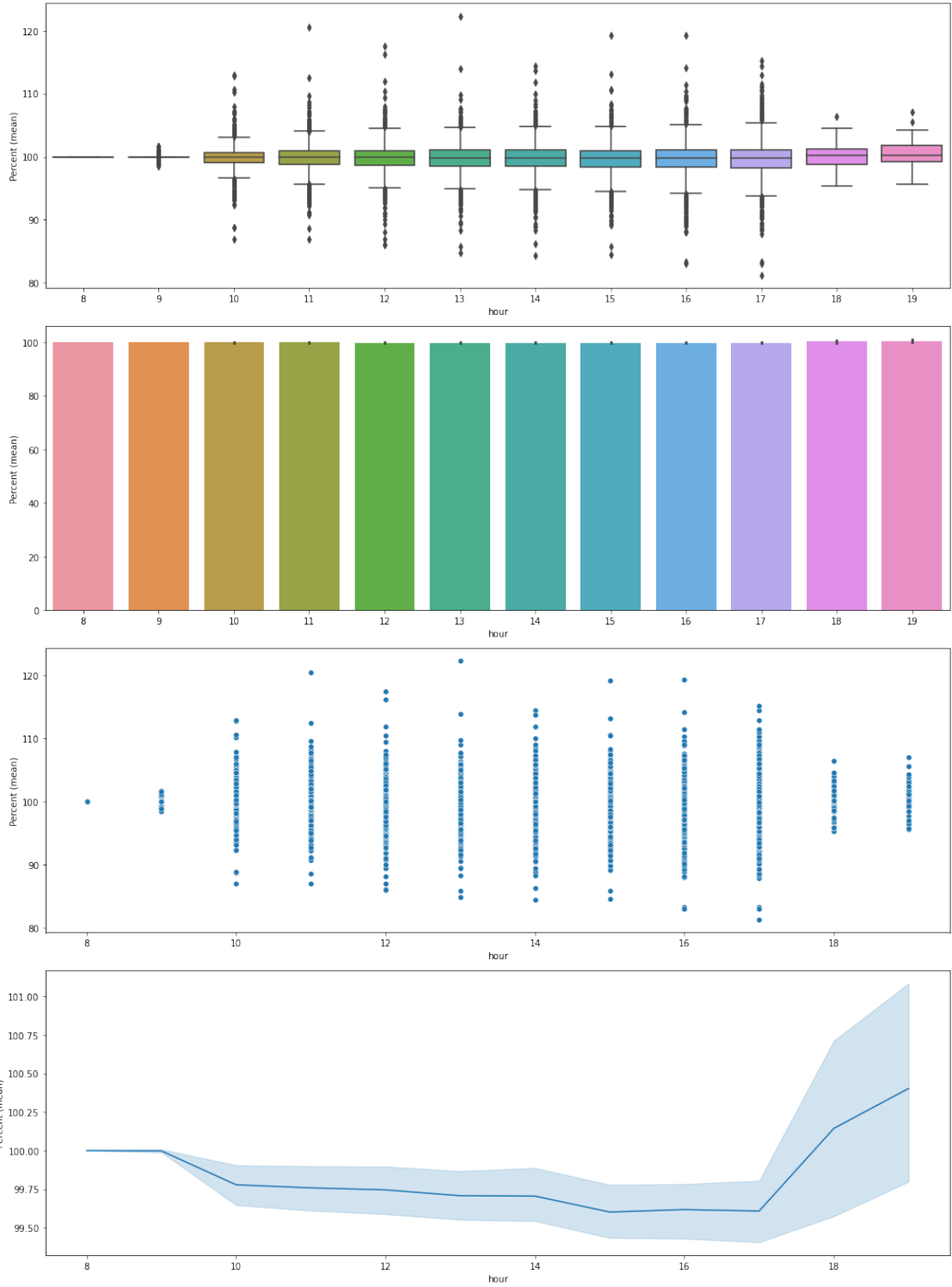
```
[10]:
```

	year	week	day	hour	Symbol	Percent (mean)
0	2020	40	28	9	ENR.F	100.0
1	2020	40	28	10	ENR.F	95.592915
2	2020	40	28	11	ENR.F	102.998636
3	2020	40	28	12	ENR.F	102.998636
4	2020	40	28	13	ENR.F	99.045884
...
9700	2020	53	28	13	8TRA.DE	100.306814
9701	2020	53	28	14	8TRA.DE	100.043832
9702	2020	53	28	15	8TRA.DE	100.131487
9703	2020	53	28	16	8TRA.DE	100.284894
9704	2020	53	28	17	8TRA.DE	99.846593

[9705 rows x 6 columns]

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

```
Percent (mean)
hour
8          100.0
9      99.997603
10     99.777622
11     99.757996
12     99.744947
```



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]:
```

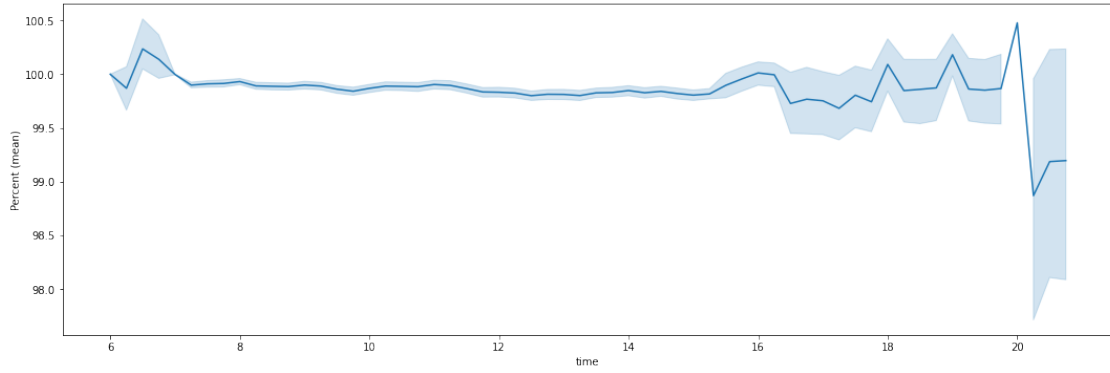
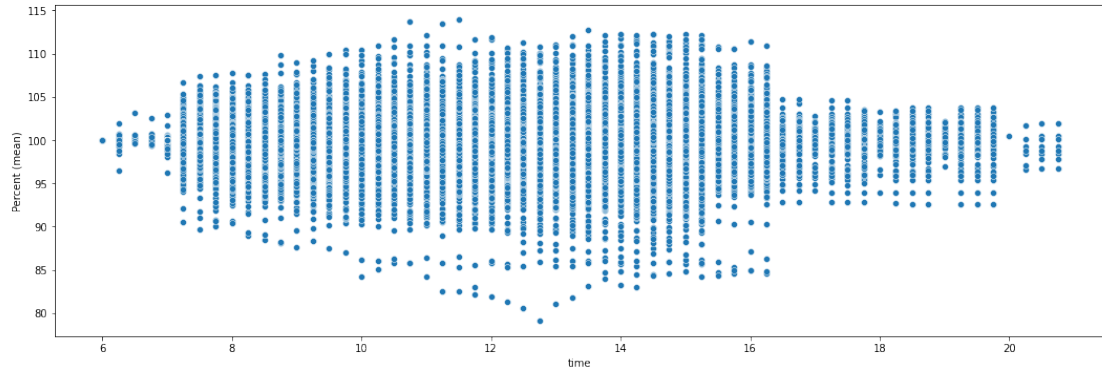
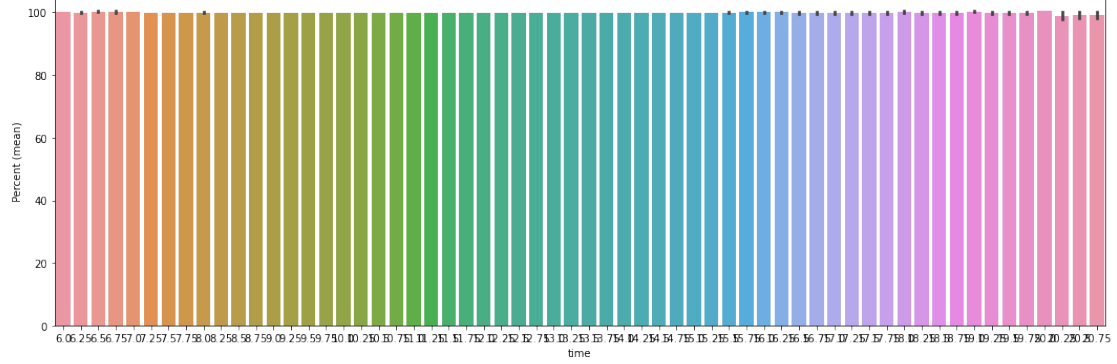
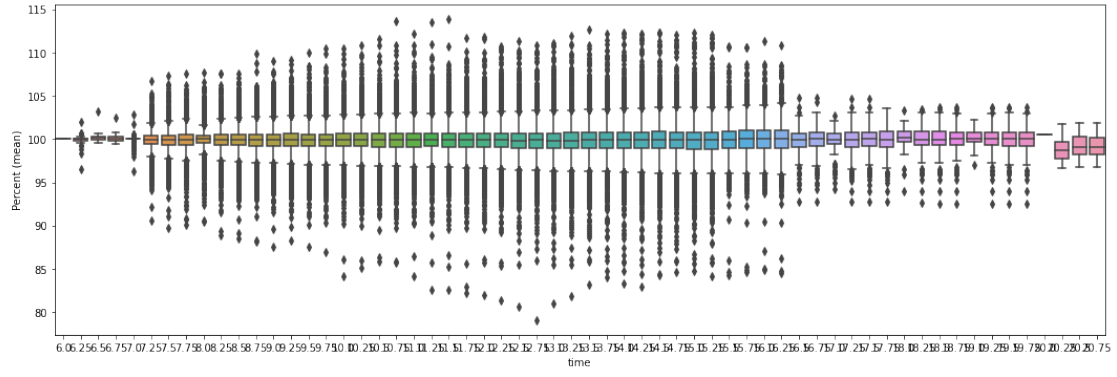
	year	week	day	hour	minute	time	Symbol	Percent (mean)
0	2021	11	17	8	0	8.0	SZU.DE	100.0
1	2021	11	17	8	15	8.25	SZU.DE	99.782295
2	2021	11	17	8	30	8.5	SZU.DE	99.782295
3	2021	11	17	8	45	8.75	SZU.DE	99.854866
4	2021	11	17	9	0	9.0	SZU.DE	99.637154
...
207268	2021	19	11	14	15	14.25	PBB.DE	96.702191
207269	2021	19	11	14	30	14.5	PBB.DE	96.480994
207270	2021	19	11	14	45	14.75	PBB.DE	96.420672
207271	2021	19	11	15	0	15.0	PBB.DE	96.682084
207272	2021	19	11	15	15	15.25	PBB.DE	96.420672

```
[207273 rows x 8 columns]
```

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

```
Percent (mean)
```

time	
6.00	100.0
6.25	99.869236
6.50	100.236121
6.75	100.138883
7.00	99.997762



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]:
```

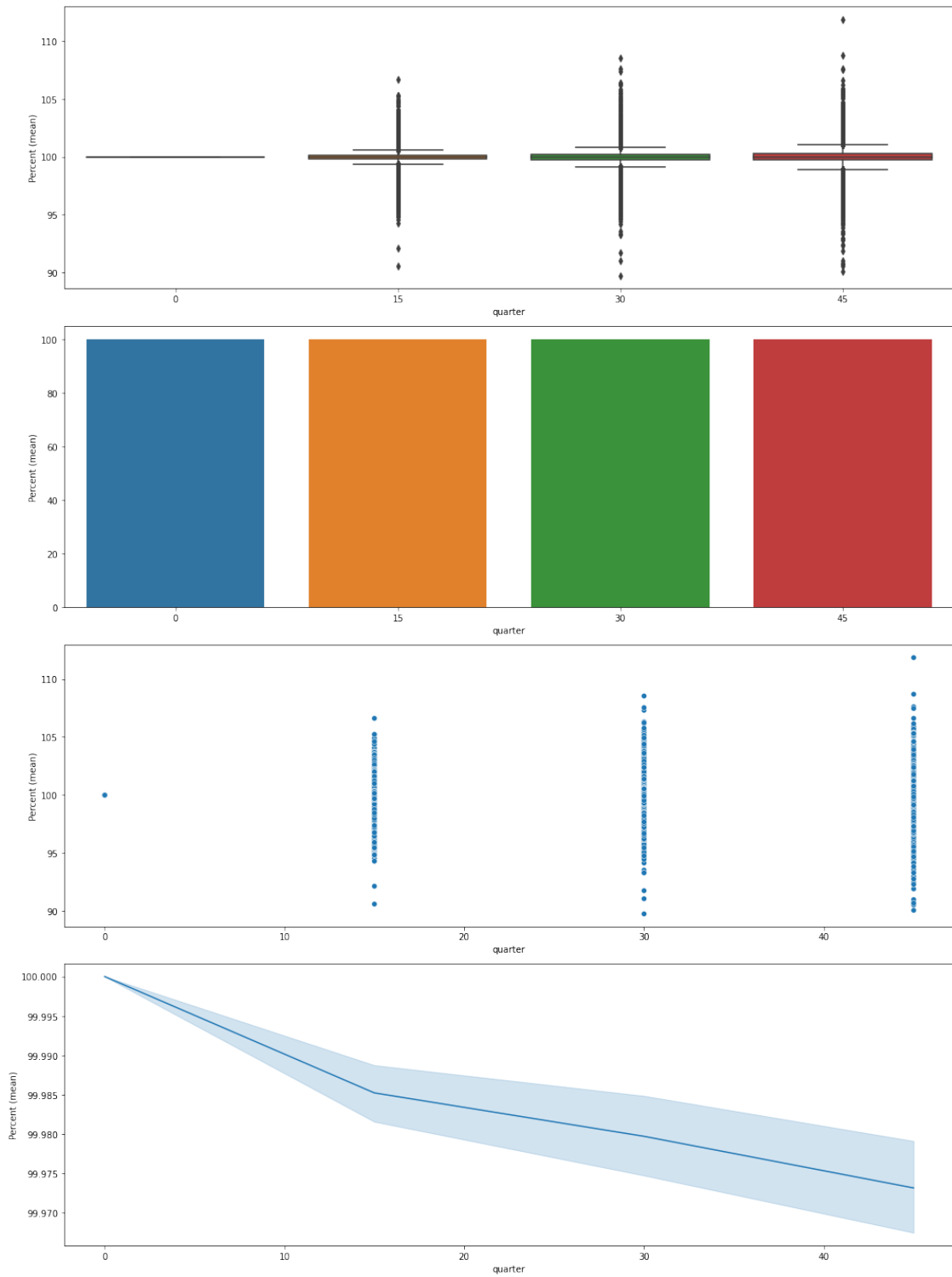
	year	week	day	hour	minute	quarter	Symbol	Percent (mean)
0	2021	11	17	8	0	0	SZU.DE	100.0
1	2021	11	17	8	15	15	SZU.DE	99.782295
2	2021	11	17	8	30	30	SZU.DE	99.782295
3	2021	11	17	8	45	45	SZU.DE	99.854866
4	2021	11	17	9	0	0	SZU.DE	100.0
...
206307	2021	19	11	15	15	15	PBB.DE	99.729617
206308	2021	19	11	7	0	0	PBB.DE	100.0
206309	2021	19	11	7	15	15	PBB.DE	98.451636
206310	2021	19	11	7	30	30	PBB.DE	98.230448
206311	2021	19	11	7	45	45	PBB.DE	98.049465

```
[206312 rows x 8 columns]
```

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

```
Percent (mean)
```

quarter	Percent (mean)
0	100.0
15	99.985235
30	99.979725
45	99.973171



[]: