Stock Volatility Forecasting Report

This is a report on analyzing and forecasting the stock price volatility based on data using `ARIMA` **Recurrent Neural Networks (RNNs)**, **Convolutional Neural Network (C** **Generative Adversarial Network (GAN)**. The global structure of the report is:

Part I. Data analysis and cleaning

- Basic manipulation and analysis
- Data cleaning
- Data preparation

Part II. Statistical analysis

- Statistical analysis of the volatility data
- Time series analysis with `ARIMA`

Part III. Deep learning models

- Basic model: single-step, single-feature forecasting with LSTM
- Generalized model: multi-step, multi-feature forcasting with LSTM
- Advanced model: Generative Adversarial Network (GAN) with RNN and CNN.

Part IV. Conclusions and Next steps

- Conclusions
- Next steps

References:

- [My own deep learning project]
- [Using the latest advancements in deep learning to predict stock price movements]

Introduction

1. The Notebook

Follow the notebook, we can recreate all the results, notice the followings

- Upload the `stockdata3.csv` file to the root folder on google colab.
- To navigate better, use the table of contents bottom on the upper-left sidebar.
- **For clarity, all code cells are hiden, double click on the cell to get the**
- Change the parameters as indicated in the comments to create more custom outputs.
- All source code can also be found in the project file folder

2. The stock prices dataset

This report uses a Stock Prices Dataset provided by SIG.

Before analyzing the data with codes, we have the following observations.

- This dataset contains $\mathbf{6}$ different stocks $\mathbf{a, b, c, d, e}$ and $\mathbf{f}$.
- Data were collected every $\mathbf{1}$ minute, beginning from $\mathbf{Day\ 1, 9{:}30\ am\ to\ Day\ 362, 1}$
- In total, we have $\mathbf{98353\ rows\ (minutes)}$ and $\mathbf{8\ columns\ (day\ number, time}$

Part I. Data analysis and cleaning

Basic manipulation

Code and examples

```
basic.py
```

read the file and show the head

	day	timestr	a	b	c	d	e	f
0	1	09:30:00	325.450	13.795	94.500	49.985	49.93	17.025
1	1	09:31:00	325.245	13.890	94.515	49.990	49.96	17.025
2	1	09:32:00	325.580	13.905	94.565	49.995	49.96	17.025
3	1	09:33:00	325.470	13.955	94.645	50.065	49.92	17.025
4	1	09:34:00	325.295	13.975	94.580	50.030	49.90	17.025

days per month, minutes per day and find special day(s)

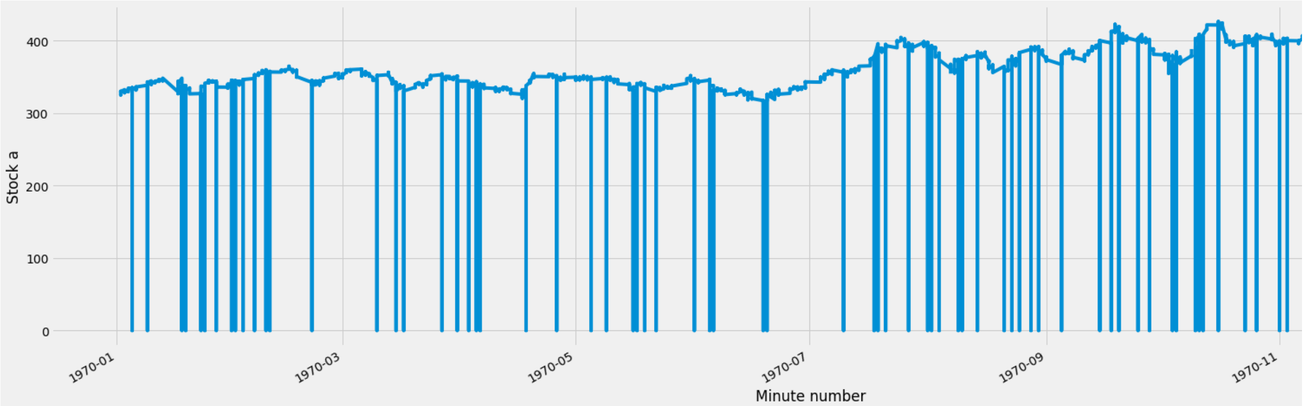
```
Mean number of days per month: 21.0
Mean number of sample points per day: 390.2857142857143
Day 327 has 211 minutes data!
       day   timestr        a      b       c       d      e      f
88963  327  12:56:00  364.570  5.115  43.413  43.935  40.24  9.765
88964  327  12:57:00  364.700  5.115  43.405  43.875  40.24  9.765
88965  327  12:58:00  364.545  5.215  43.405  43.865  40.27  9.765
88966  327  12:59:00  364.380  5.125  43.325  43.855  40.33  9.765
88967  327  13:00:00  363.580  5.225  43.333  43.835  40.31  9.765
```

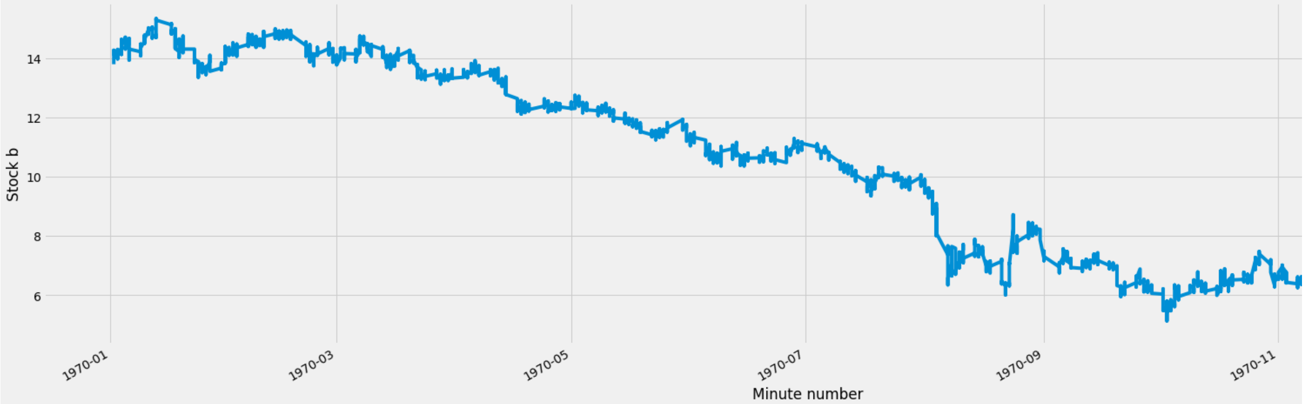
Basic checks: find null values and fill, set index, etc.

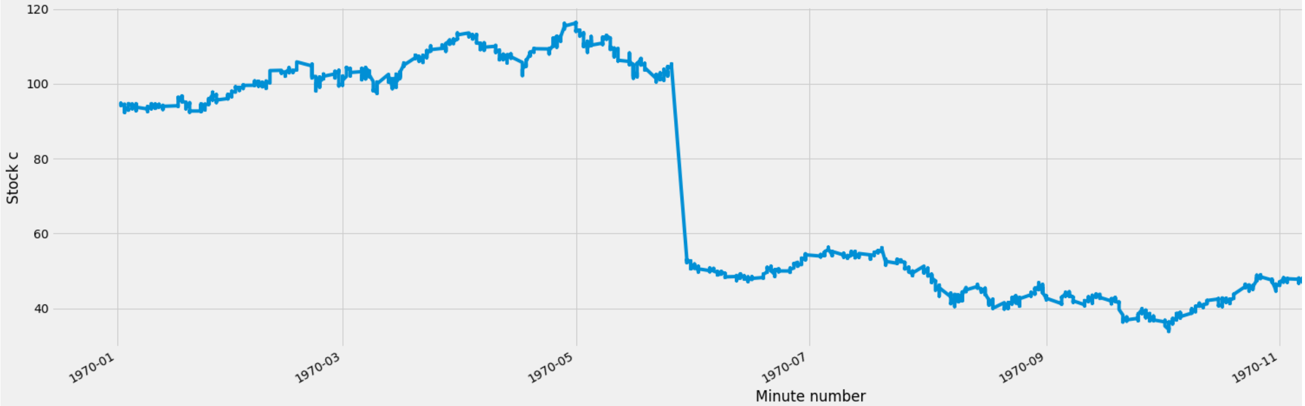
```
Null values summary:

timestr       0
a            71
b             0
c            31
d            18
e             0
f          1371
dtype: int64
```

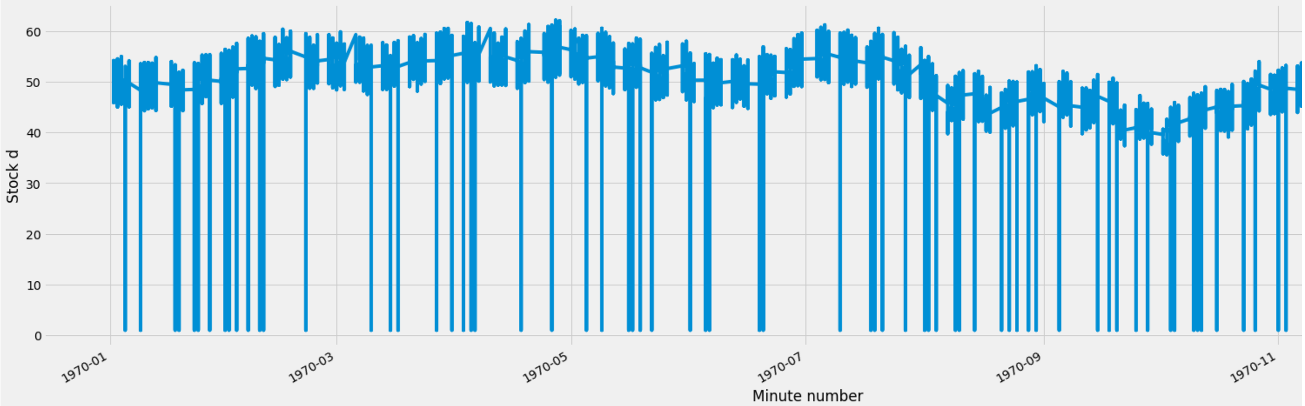
day	timestr	a	b	c	d	e	f
1970-01-02	09:30:00	325.450	13.795	94.500	49.985	49.93	17.025
1970-01-02	09:31:00	325.245	13.890	94.515	49.990	49.96	17.025

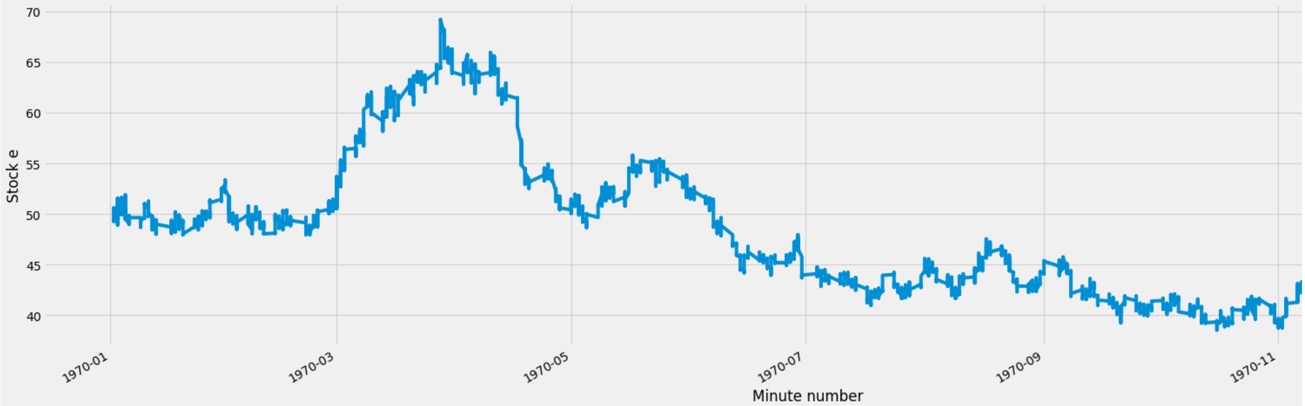
Example: plot the stock price columns


Stock a
400
300
200
100
0
1970-01
1970-03
1970-05
1970-07
1970-09
1970-11
Minute number


Stock b
14
12
10
8
6
1970-01
1970-03
1970-05
1970-07
1970-09
1970-11
Minute number


Stock c
120
100
80
60
40
1970-01
1970-03
1970-05
1970-07
1970-09
1970-11
Minute number


Stock d
60
50
40
30
20
10
0
1970-01
1970-03
1970-05
1970-07
1970-09
1970-11
Minute number


Stock e
70
65
60
55
50
45
40
1970-01
1970-03
1970-05
1970-07
1970-09
1970-11
Minute number

Data Analysis

Basic aspects:

- Most days contain data for the full 391 minutes from `9:30 am` to `4:00 pm`
- Day 327 doesn't have the full number ofminutes, the data stops at `1:00 pm` that day, it only
- Every month contains 21 days.

As a high level overview, some distinguishable features appear when we plot the data:

- Prices of a $\textbf{drop abnormally}$ at some random places.
- Prices of b seem to have relatively stable volatility in the first 7 months and last 3 months a in the 8-th and 9-th month.
- Prices of c $\textbf{drop drastically}$ in the 6-th month, besides that, the price of c is quite stabl
- Prices of d $\textbf{drop abnormally}$ at some random places.
- Prices of d have a $\textbf{large day-to-day volatility}$ compared to the longer-term volatilit
- Prices of f are very $\textbf{illiquid}$, that is, the prices don't move much on a minute-to-minute ba
- Peices of b and e have no null entries, no random drops.

check for special day(s)

```
Day 1970-11-24 00:00:00 has 211 minutes data!
               timestr        a      b       c       d      e      f
day
1970-11-24  12:56:00  364.570  5.115  43.413  43.935  40.24  9.765
1970-11-24  12:57:00  364.700  5.115  43.405  43.875  40.24  9.765
1970-11-24  12:58:00  364.545  5.215  43.405  43.865  40.27  9.765
1970-11-24  12:59:00  364.380  5.125  43.325  43.855  40.33  9.765
1970-11-24  13:00:00  363.580  5.225  43.333  43.835  40.31  9.765
```

check for random drops in prices of stock a and stock d

```
Stock prices of a have 93 random drops
Stock prices of d have 93 random drops
Stock prices of a and d drop at the same time in 93 places
```

	a	d
day		
1970-01-05	0.0	1.0

Data analysis

- The times of the price drops in stock a and stock d seem to be random and appear on 93 di
- The price drops of stock a and d happend $\mathbf{at\ the\ exactly\ same\ places}$.
- Abnormal values of a are all 0.0's, abnormal values of b are all 1.0's.

We conclude that $\mathbf{these\ drops\ are\ recording\ errors}$, the default value of missing a price 1.0.

Data cleaning

Data cleaning: set abnormal values to NaN

We set those abnormal values in the prices of stock a and stock d as $\mathbf{NaN}$, which makes it com functionality of pandas.

set abnormal values to NaN

Data cleaning: add missing data on day 327

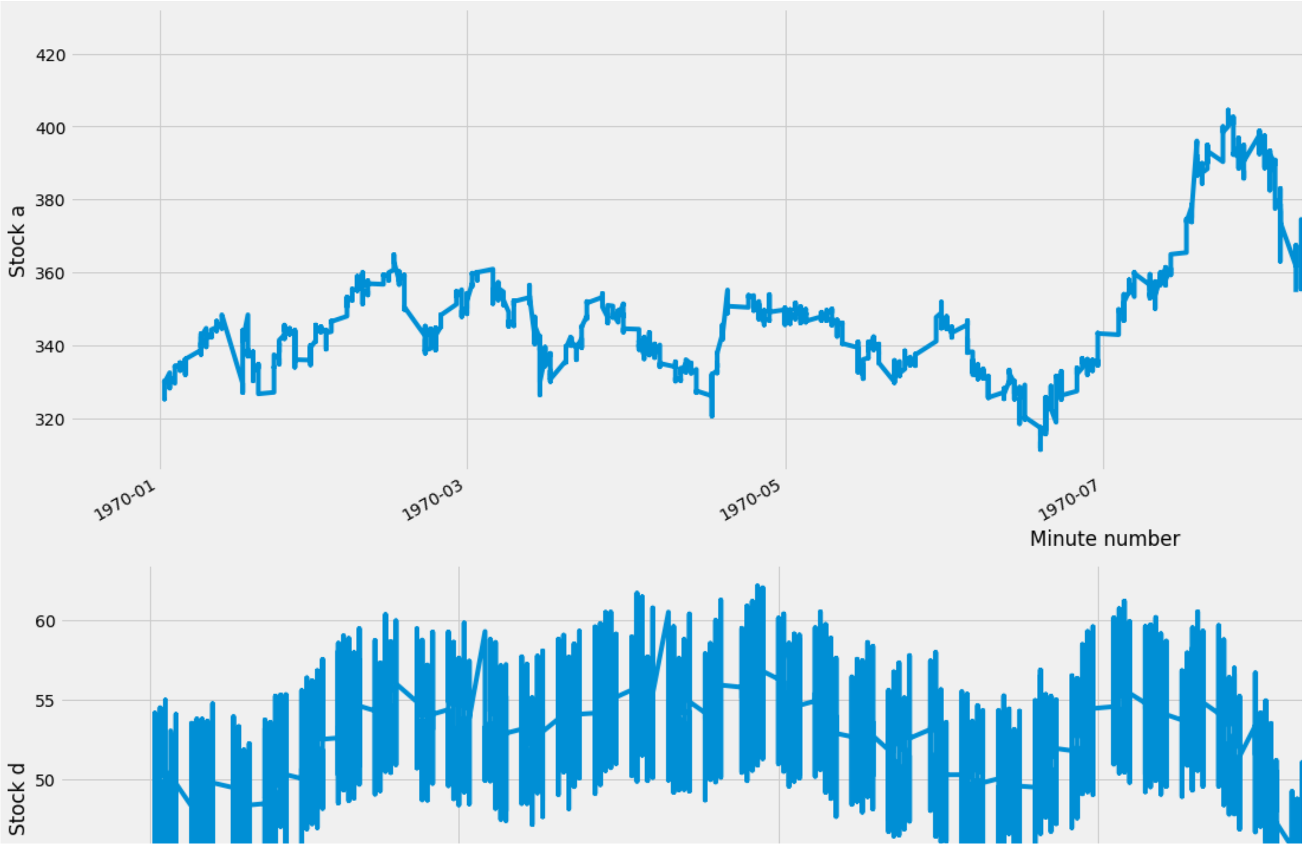
Compute volatility

Make sure we already have what we want

- no random drops in prices of stock a and stock d
- day 327 has full 391 rows

cleaned data

```
Day 327 now has 391 minutes of data
```



Data preparation: from prices to daily volatility of

We first define the quantity(volatility measured in annualized percent return) that we're interested the following factors affect our code realization

- Quantities of interest
- Model assumptions
- Frequency of sampling

Quantities of interest: Daily volatility of annualized monthly percent return $\sigma_{d,r}$

Definition: the **Annualized monthly percent return** $A_m(t)$ at time t is given by

$$A_m(t) = (\frac{\text{Price at time a month later than } t}{\text{Price at time } t})^{\frac{1 \text{ yea}}{1 \text{ mon}}}$$
$$= (\frac{p(t+t_m)}{p(t)})^{12} - 1$$

where

- $p(t)$ denote the price at time t, we use $\mathbf{minute}$ as the unit, the same as in our `stockdata3`
- t_m denote a month in unit of the dataset, for our `stockdata3.csv` dataset, $t_m = 391 \text{ mins}$

Definition: the **Daily volatility of annualized monthly percent return** $\sigma_{d,m}(t)$ at time t is defined tc $\{A_m(x)|t \leq x \leq t + t_d\}$, where

- t_d denote the number of minutes in a day, for us $t_d = 391$. To be more precise

$$\sigma_{d,m}(t) = \sqrt{\left[\frac{1}{t_d}\sum_{j=0}^{t_d}\left(A_m(t+j) - \frac{1}{t_d+1}\sum_{i=0}^{t_d} A_m\right.\right.}$$

Model assumptions

- We're interested in the statistics on percent returns after holding an asset for a month (but v
- Volatility is a constant in a day.

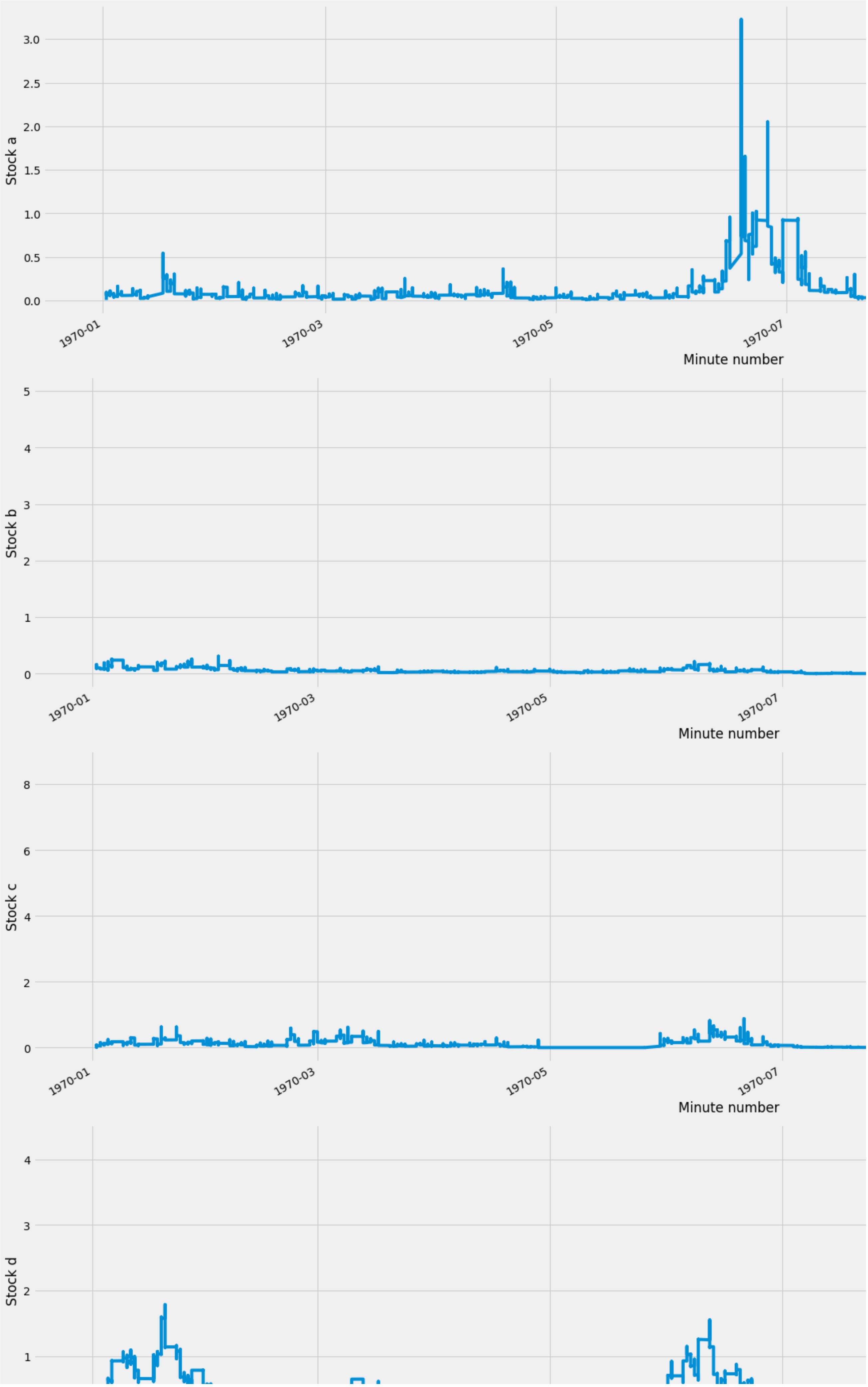
The first assumption tells us how to chose window size to compute the annualized percent return deviation functionality in `pandas` to compute our volatility of annualized monthly percent return. T we need

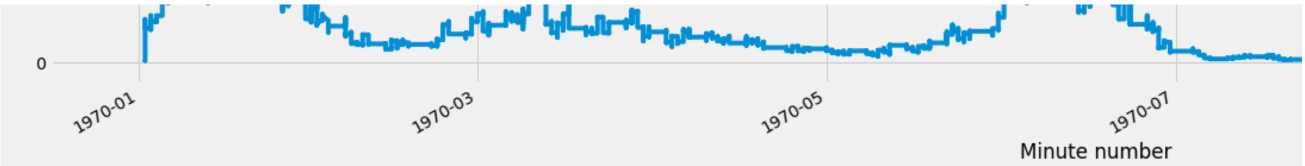
- window size: `t_m = t_d*21=8211`
- window moving step: `1`, that is (every minute has a annualized percent return)

The second assumption let us estimate the volatility by computing sample deviation of the percer

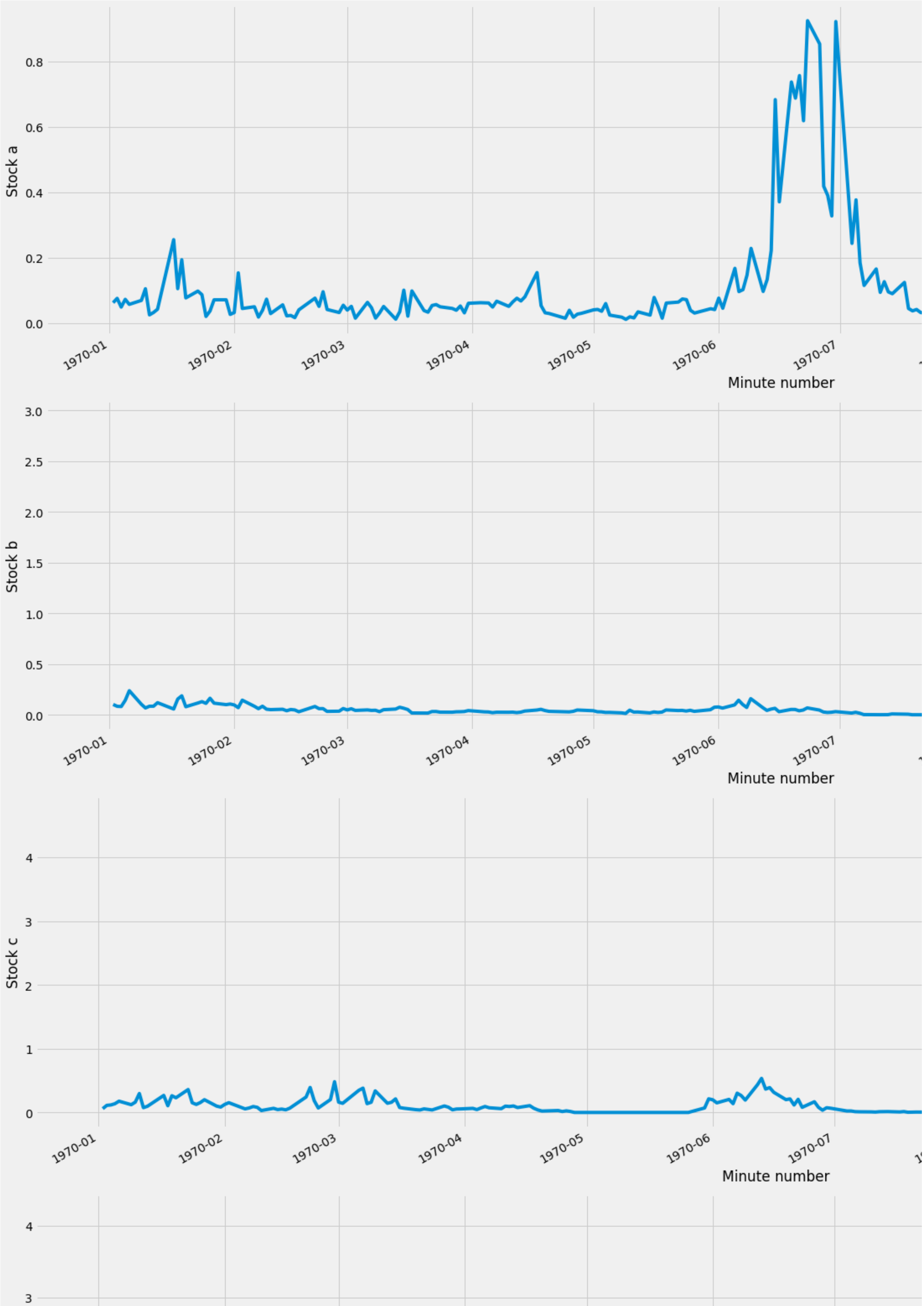
- sample size: `t_d=391`, that is we use a consecutive sequence of 391 data points to comput
- sample frenquency: `t=1`, that is we'll consider the daily movement of the volatility.

Volatility in minutes


Stock a
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1970-01
1970-03
1970-05
1970-07
Minute number
Stock b
5
4
3
2
1
0
1970-01
1970-03
1970-05
1970-07
Minute number
Stock c
8
6
4
2
0
1970-01
1970-03
1970-05
1970-07
Minute number
Stock d
4
3
2
1



Daily volatility



Final volatility dataset

```
a     0
b     0
c     0
```

Part II. Statistical analysis and models

Correlation analysis

Though it seems that there's no obvious correlation among the 6 stocks, and some of them even of the report, we compute several different correlations (**Naive correlation, Pearson correlation, local Pearson correlation, instan** and related statistics in order to

- Test the validity of our observations (i.e. no two stocks are apprantly correlated).
- Chose source and target stocks for later machine learning(especially deep learning) models

By doing so, we can get more understanding about the 'quality' and 'inner relations' of the data. If the stock that we want to predict (e.g. "f"), then there is no need for us to use it in the training of o one stock has higher-than-random correlations to another stock, then it's good to use one of them this case, **to determine which stock volatility leads, and which stock volatil Dynamic time wrapping**.

Code and Examples

correlation.py

```
Requirement already satisfied: dtw in /usr/local/lib/python3.6/dist-packages (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from
```

```
#@title Example: Naive correlation.
new_sigma.corr()
```

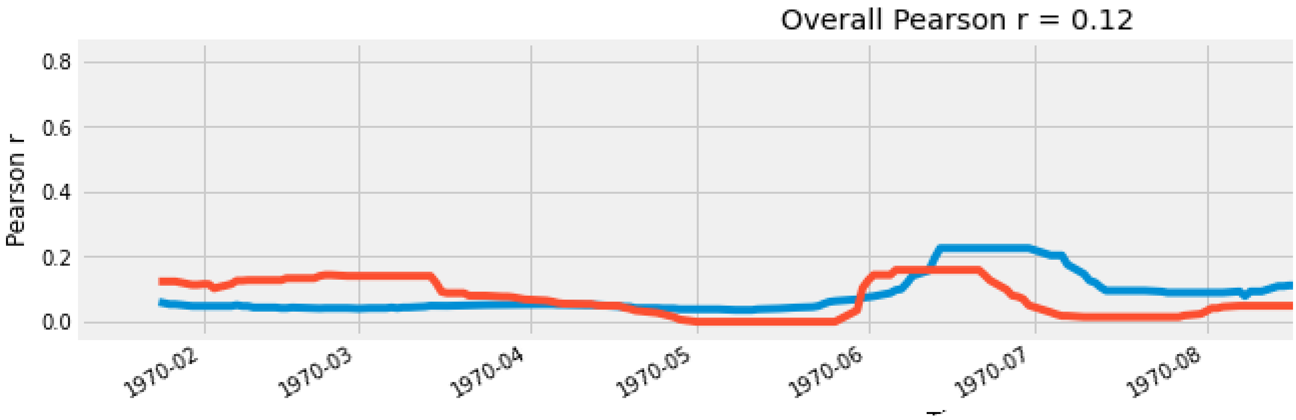
Example: Naiv

	a	b	c	e	f
a	1.000000	0.178992	0.124927	-0.139937	0.033105
b	0.178992	1.000000	0.864241	-0.064492	0.505138
c	0.124927	0.864241	1.000000	-0.031285	0.609643
e	-0.139937	-0.064492	-0.031285	1.000000	-0.069070
f	0.033105	0.505138	0.609643	-0.069070	1.000000

```
#@title Example: Pearson correlation
pearson(new_sigma, "a", "c")
```

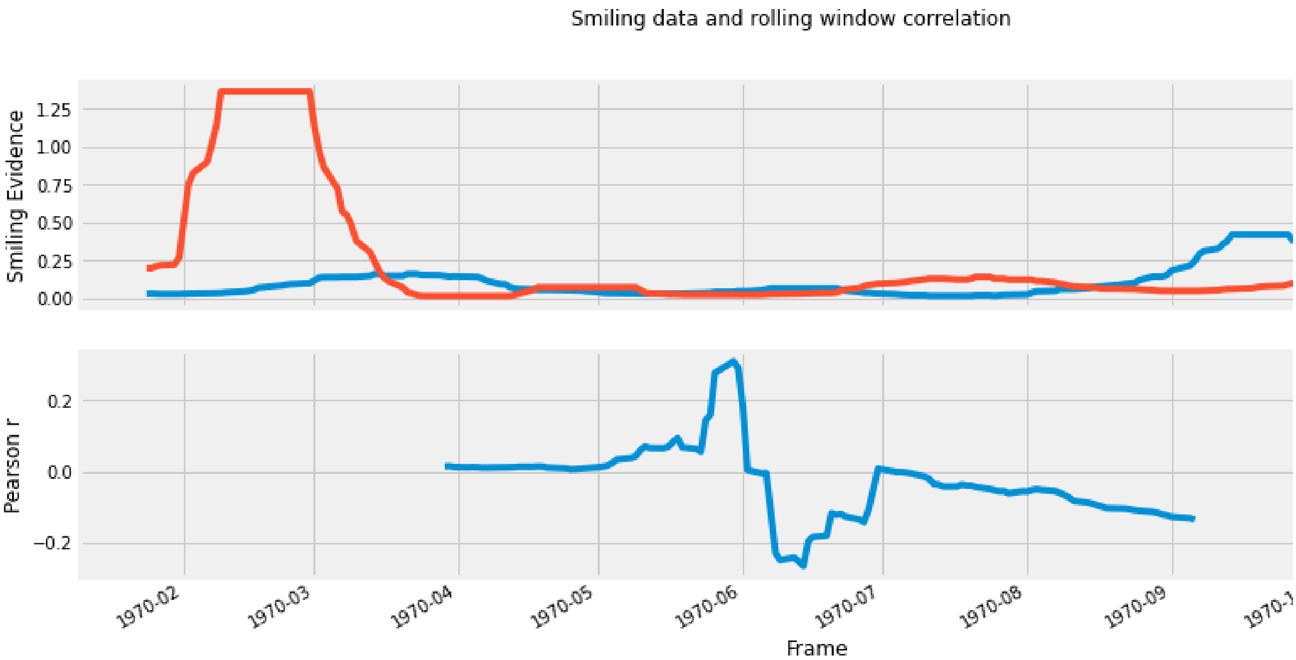
Example: Pear

```
Pandas computed Pearson r: 0.12492677413179742
Scipy computed Pearson r: 0.12492677413179744 and p-value: 0.05797907566773995
```



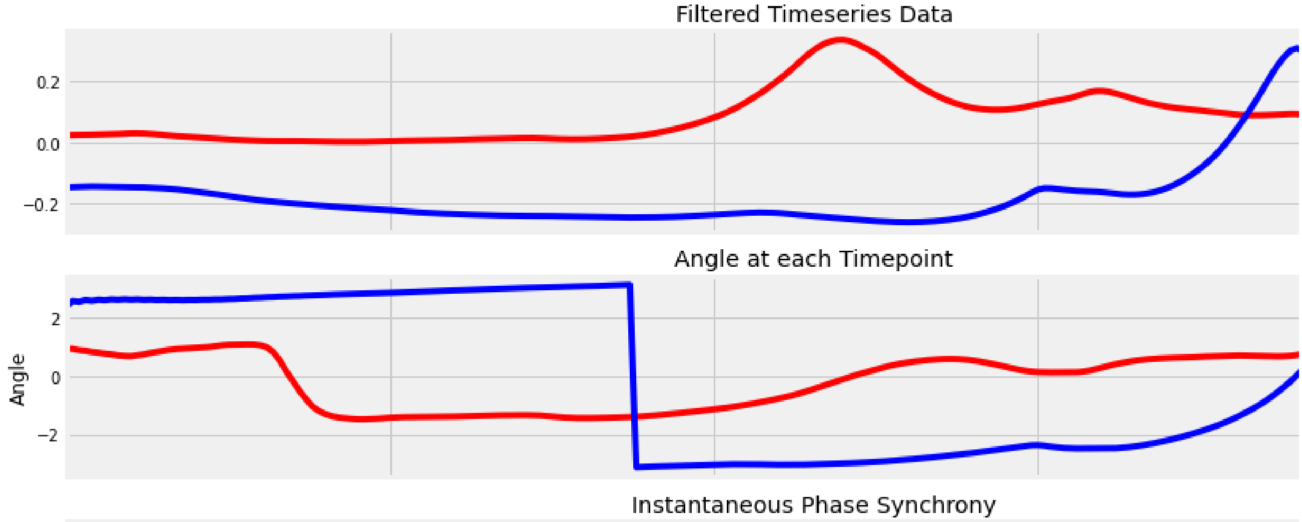

```
#@title Example: local Pearson correlation
local_pearson(new_sigma, "f", "e")
```

Example: loca



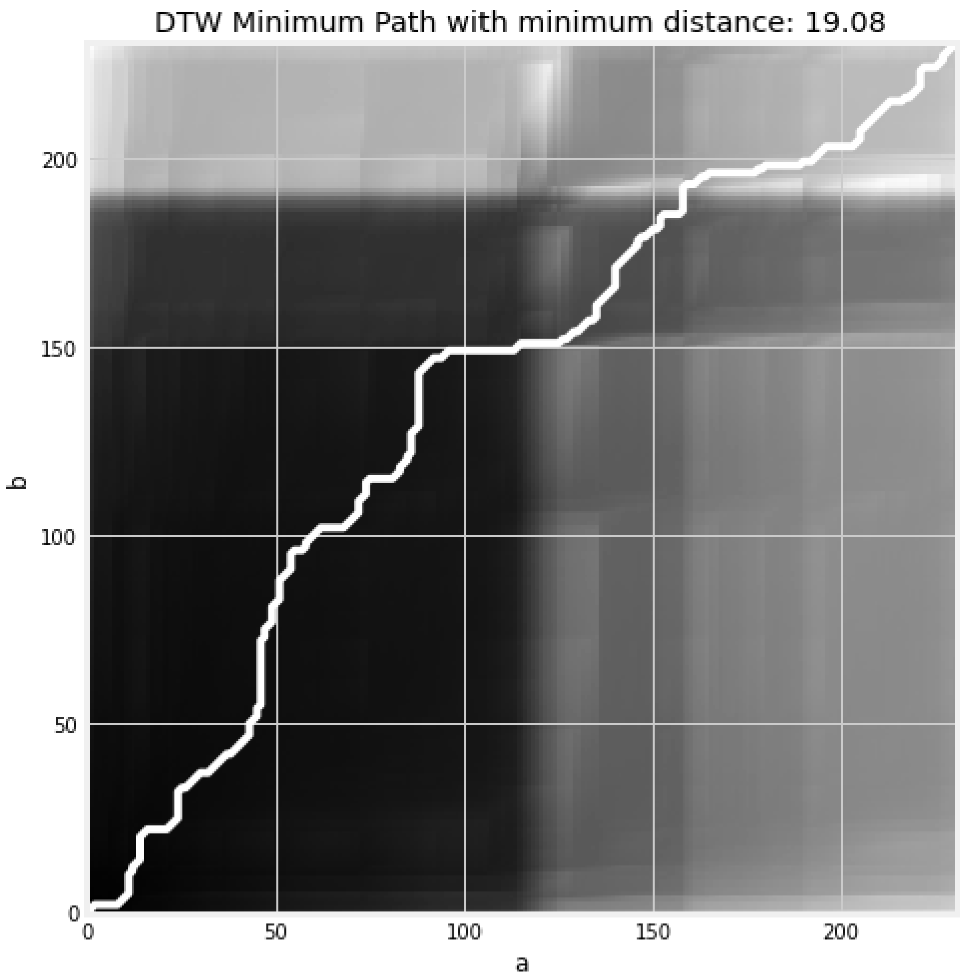

```
#@title Example: instantaneous phase synchronization
instant_phase_sync(new_sigma, "a", "b")
```

Example: insta




```
#@title Example: dynamic time wraping
dynamic_time_warping(new_sigma, "a", "b")
```

Example: dyna



Data analysis

Inspecting the correlations from different angles, we find

- **Stock b and stock c have the highest correlation, 0.864241,** among all th
- **Stock b and stock f, Stock c and stock f** also have high positive correlation$.
- All other pairs have relatively low correlations.

- $\textbf{Stock } b \textbf{ slightly leads stock } a.$

We conclude that

- $\textbf{The assumption that no two stocks have apparent correlation is wro}$
- It's reasonable to $\textbf{use stock } a \textbf{ and stock } f \textbf{ as targets and the other stocks as source for f}$.

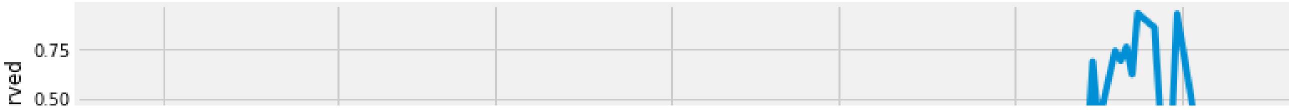
Statistical models for predicting volatility

As we'll see below, some remarkable patterns (e.g. seasonality pattern) naturally appear in our da

- We visualize our data using $\textbf{time-series decomposition}$ that allows us to decompos trend, seasonality, and noise.
- We $\textbf{train an ARIMA (Autoregressive Integrated Moving Average) m}$ Volatility values. To get optimal output, we first
- Use $\textbf{grid search}$(in a range) to get the optimal parameters for the ARIMA mode.
- Fit arima models to predict next month's volatility

time_series.py

```
plot_component(new_sigma, "a")
```




```
arima_parameters(new_sigma, "a")
```

```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:-90.39976752180513
-90.39976752180513
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:-104.6220404675714
-104.6220404675714
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:-37.93079196365547
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:-121.41039301464231
-121.41039301464231
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:-113.84340411735667
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:-131.54236196798252
-131.54236196798252
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:-57.495873873140084
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:-114.54466043367323
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:-208.70966183847128
-208.70966183847128
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:-197.24167911716506
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:-103.96001828398299
ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:-202.16848163063582
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:-203.58604837770298
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:-216.8758521946111
-216.8758521946111
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:-135.47926852752335
```

```
arima_parameters(new_sigma, "b")
```

```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:170.07716662303517
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:171.4894723986554
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:295.99244410313185
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:167.97755357713928
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:171.0217094393121
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:171.14819938319445
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:240.35004513125935
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:173.81980273179033
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:0.021926806985391067
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:10.470838674848721
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:133.4761652527779
ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:21.584047009046785
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:8.487974476915923
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:11.552996030856292
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:89.06706505631814
ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:28.263363670679496
ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:-91.04951798935619
-91.04951798935619
ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:-74.00992181183118
ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:77.02467580342876
ARIMA(0, 1, 0)x(0, 1, 1, 12)12 - AIC:-42.32582201252652
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:-75.08794012733885
ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:-72.01184551233767
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:25.369008281037566
ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:-36.281514256806034
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:-87.70209710986671
ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:-70.73889751151115
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:79.47382549922017
ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:-39.08762494963351
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:-73.14269283939211
ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:-68.7397493337976
```

```
arima_parameters(new_sigma, "f")
```

```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:247.99757729220502
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:232.03747689037294
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:319.7137248769461
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:227.70141678387574
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:228.67377045275097
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:229.9117970732568
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:282.7315535157633
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:232.855001615836
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:107.84989948113763
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:101.16457643410824
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:193.51659680308134
ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:111.53183486696223
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:98.19783457495264
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:101.34031670247452
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:166.7131597091078
ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:115.64499685926572
ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:-20.169147903329332
-20.169147903329332
ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:-10.723609000199815
ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:104.64454664081094
ARIMA(0, 1, 0)x(0, 1, 1, 12)12 - AIC:18.327876840857144
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:-11.822215289801377
ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:-8.88126757326286
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:77.56528210875692
ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:25.453247989244147
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:-20.429899618764082
-20.429899618764082
ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:-9.336208418240872
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:107.16134958549422
ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:18.977575026484587
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:-11.5562380965073
ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:-7.478656499438642
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:78.00783988815468
ARIMA(0, 1, 1)x(1, 1, 1, 12)12 - AIC:26.154840605737697
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:-37.43898748560808
-37.43898748560808
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:-27.784465671689375
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:84.96802822594874
ARIMA(1, 0, 0)x(0, 1, 1, 12)12 - AIC:0.2992347111687792
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:-27.98019714618877
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:-26.066206938084804
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:59.73826753861643
ARIMA(1, 0, 0)x(1, 1, 1, 12)12 - AIC:7.547042262332624
```

imports

model fit

```
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:61.757484602404
```

Model parameters

```
[0.11561704 0.73469703]
[0.13240487 0.81177913]
[0.17833905 0.80715809]
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:78.44992161664612
```

Model predictions

```
[0.06660579 0.07960862 0.08916176 0.09618043 0.10133702 0.10512555
 0.10790897 0.10995395 0.11145638 0.11256022 0.1133712  0.11396703
 0.11440478 0.1147264  0.11496269 0.11513629 0.11526384 0.11535754
 0.11542639 0.11547697 0.11551413 0.11554143]
[0.28566923 0.25682168 0.23340384 0.21439373 0.19896171 0.18643432
 0.17626485 0.16800949 0.16130796 0.15586779 0.15145158 0.14786659
 0.14495637 0.14259392 0.14067612 0.1391193  0.1378555  0.13682958
 0.13599676 0.13532069 0.13477187 0.13432635]
[0.3384968  0.30761167 0.28268249 0.26256071 0.24631924 0.23320981
 0.22262843 0.21408758 0.20719377 0.20162937 0.19713802 0.19351279
 0.19058666 0.18822481 0.18631842 0.18477966 0.18353764 0.18253514
 0.18172596 0.18107282 0.18054563 0.18012011]
```

Data Analysis

- Components plot show the obvious seasonality, for example, for stock a, we can find an alte alternate between high values and low values in a period of roughly $3-4$ months.
- The optimal ARIMA parameters for "a" are `(1, 0, 1)x(0, 0, 1, 12)`
- As we forecast further out into the future, we becomes less confident in our values. This is r by our model, which grow larger as we move further out into the future.
- We'll do more carefully analysis of the predictions for the deep learning models.

Part III. Deep learning models

Basic model: single-step, single-feature forecasting with L

$\textbf{Recurrent Neural Networks (RNNs)}$ are good fits for time-series analysis because R designed to capture patterns developing through time.

However, vanilla RNNs have a major disadvantage---the vanishing gradient problem---"the changes so small, making the network unable to converge to a optimal solution.

$\textbf{LSTM (Long-Short Term Memory)}$ is a variation of vanilla RNNS,it overcomes the va problem by clipping gradients if they exceed some constant bounds.

In this section, we will

- Process the data to fit the LSTM model
- $\textbf{Build and train the LSTM model for single-step, single-feature pred}$ volatility with only today's values of the other 5 stocks).

imports

Data preparation

Build and train the LSTM model

Make sure data forms are correct

```
(161, 1, 5)
(161, 1)
(69, 1, 5)
(69, 1)
```

LSTM with SGD, RMSprop, Adam optimizers, epochs = 100

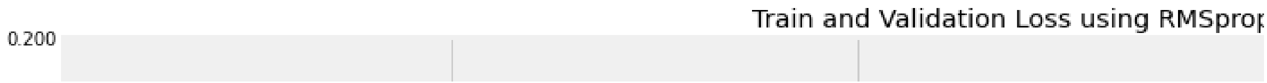
```
Train on 161 samples, validate on 69 samples
Epoch 1/100
161/161 [==============================] - 1s 3ms/step - loss: 0.0346 - accuracy: 0.6
Epoch 2/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0262 - accuracy: 0.6
Epoch 3/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0216 - accuracy: 0.6
Epoch 4/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0183 - accuracy: 0.6
Epoch 5/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0160 - accuracy: 0.6
Epoch 6/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0150 - accuracy: 0.6
Epoch 7/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0143 - accuracy: 0.6
Epoch 8/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0138 - accuracy: 0.6
Epoch 9/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 10/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 11/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0139 - accuracy: 0.6
Epoch 12/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 13/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0128 - accuracy: 0.6
Epoch 14/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0137 - accuracy: 0.6
Epoch 15/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0137 - accuracy: 0.6
Epoch 16/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 17/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 18/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0130 - accuracy: 0.6
Epoch 19/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 20/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 21/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0138 - accuracy: 0.6
Epoch 22/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 23/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0138 - accuracy: 0.6
Epoch 24/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 25/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 26/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0140 - accuracy: 0.6
Epoch 27/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 28/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0137 - accuracy: 0.6
Epoch 29/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0137 - accuracy: 0.6
Epoch 30/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
```

```
Epoch 31/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 32/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 33/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 34/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 35/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 36/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 37/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 38/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 39/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 40/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 41/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0131 - accuracy: 0.6
Epoch 42/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 43/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0133 - accuracy: 0.6
Epoch 44/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0137 - accuracy: 0.6
Epoch 45/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0131 - accuracy: 0.6
Epoch 46/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 47/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 48/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 49/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 50/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 51/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 52/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 53/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 54/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0133 - accuracy: 0.6
Epoch 55/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0136 - accuracy: 0.6
Epoch 56/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0130 - accuracy: 0.6
Epoch 57/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 58/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0133 - accuracy: 0.6
Epoch 59/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 60/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0131 - accuracy: 0.6
Epoch 61/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0131 - accuracy: 0.6
```

```
Epoch 62/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0134 - accuracy: 0.6
Epoch 63/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 64/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 65/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 66/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0133 - accuracy: 0.6
Epoch 67/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0131 - accuracy: 0.6
Epoch 68/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0132 - accuracy: 0.6
Epoch 69/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 70/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0131 - accuracy: 0.6
Epoch 71/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 72/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 73/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0138 - accuracy: 0.6
Epoch 74/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0129 - accuracy: 0.6
Epoch 75/100
161/161 [==============================] - 0s 2ms/step - loss: 0.0135 - accuracy: 0.6
Epoch 76/100
```

Plot result

Plot result



Plot predictions

Plot predictions

```
Train Score: 0.11 RMSE
Test Score: 0.11 RMSE
```

Data Analysis

We only trained the model for 100 epochs, feel free to modify it to any number as long as we have results we find during the experiments

- LSTM with Adam or RMSprop optimizers work better than the SGD optimizer in this project.
- Each model fits the training dataset very well.
- **The prediction captures the range and characteristics of the real dat**
- Most importantly, the model predict the large rises or drops in the volatility before they happ investments.

Generalized model: multi-step, multi-feature forecasting

We build a multi-step, multi-feature LSTM model in this section. That means we can use several-d features in the future.

For example, we can use last 12-day's data of a, b, c, f to predict next thr section, we

- Process the data to fit the requirements of all possible multi-step, multi-feature prediction ta
- We modify the LSTM model accordingly.
- Plot the 3-day prediction for a and f with last 12-day's data of a, b, c and f.

Data preparation

Make the data forms are all correct

```
(151, 12, 4)
(151, 6)
(66, 12, 4)
(151, 6)
```

Scaling, vectorize and de_vectorize

model training

Train the model. Change the optimizer parameter to use other optimizers, e.g.

```
Train on 151 samples, validate on 66 samples
Epoch 1/200
151/151 [==============================] - 1s 7ms/step - loss: 0.0190 - accuracy: 0.2
Epoch 2/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0156 - accuracy: 0.2
Epoch 3/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0144 - accuracy: 0.2
Epoch 4/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0140 - accuracy: 0.2
Epoch 5/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0132 - accuracy: 0.2
Epoch 6/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0123 - accuracy: 0.2
Epoch 7/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0120 - accuracy: 0.2
Epoch 8/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0118 - accuracy: 0.2
Epoch 9/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0112 - accuracy: 0.2
Epoch 10/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0111 - accuracy: 0.2
Epoch 11/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0110 - accuracy: 0.2
Epoch 12/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0106 - accuracy: 0.2
Epoch 13/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0106 - accuracy: 0.2
Epoch 14/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0102 - accuracy: 0.2
Epoch 15/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0105 - accuracy: 0.2
Epoch 16/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0101 - accuracy: 0.3
Epoch 17/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0094 - accuracy: 0.2
Epoch 18/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0095 - accuracy: 0.2
Epoch 19/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0096 - accuracy: 0.2
Epoch 20/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0093 - accuracy: 0.2
Epoch 21/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0093 - accuracy: 0.2
Epoch 22/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0091 - accuracy: 0.2
Epoch 23/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0093 - accuracy: 0.2
Epoch 24/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0090 - accuracy: 0.2
Epoch 25/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0089 - accuracy: 0.2
Epoch 26/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0089 - accuracy: 0.2
Epoch 27/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0090 - accuracy: 0.2
Epoch 28/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0090 - accuracy: 0.2
Epoch 29/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0088 - accuracy: 0.2
Epoch 30/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0087 - accuracy: 0.2
```

```
Epoch 31/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0088 - accuracy: 0.2
Epoch 32/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0085 - accuracy: 0.3
Epoch 33/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0084 - accuracy: 0.2
Epoch 34/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0084 - accuracy: 0.2
Epoch 35/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0083 - accuracy: 0.2
Epoch 36/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0083 - accuracy: 0.2
Epoch 37/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0085 - accuracy: 0.2
Epoch 38/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0084 - accuracy: 0.3
Epoch 39/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0084 - accuracy: 0.2
Epoch 40/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0082 - accuracy: 0.2
Epoch 41/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0082 - accuracy: 0.3
Epoch 42/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0081 - accuracy: 0.3
Epoch 43/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0081 - accuracy: 0.2
Epoch 44/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0080 - accuracy: 0.2
Epoch 45/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0082 - accuracy: 0.2
Epoch 46/200
151/151 [==============================] - 1s 6ms/step - loss: 0.0080 - accuracy: 0.3
Epoch 47/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0078 - accuracy: 0.3
Epoch 48/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0080 - accuracy: 0.2
Epoch 49/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0078 - accuracy: 0.2
Epoch 50/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0078 - accuracy: 0.3
Epoch 51/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0077 - accuracy: 0.3
Epoch 52/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0078 - accuracy: 0.2
Epoch 53/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0073 - accuracy: 0.3
Epoch 54/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0075 - accuracy: 0.3
Epoch 55/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0074 - accuracy: 0.3
Epoch 56/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0076 - accuracy: 0.2
Epoch 57/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0073 - accuracy: 0.3
Epoch 58/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0074 - accuracy: 0.3
Epoch 59/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0075 - accuracy: 0.2
Epoch 60/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.2
Epoch 61/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0075 - accuracy: 0.3
```

```
151/151 [==============================] - 1s 5ms/step - loss: 0.0075 - accuracy: 0.
Epoch 62/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0072 - accuracy: 0.2
Epoch 63/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0072 - accuracy: 0.3
Epoch 64/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0073 - accuracy: 0.2
Epoch 65/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0072 - accuracy: 0.2
Epoch 66/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0072 - accuracy: 0.2
Epoch 67/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.2
Epoch 68/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0073 - accuracy: 0.3
Epoch 69/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.3
Epoch 70/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.3
Epoch 71/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0072 - accuracy: 0.2
Epoch 72/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.3
Epoch 73/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0070 - accuracy: 0.3
Epoch 74/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.3
Epoch 75/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0070 - accuracy: 0.2
Epoch 76/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0066 - accuracy: 0.3
Epoch 77/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.3
Epoch 78/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 79/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0071 - accuracy: 0.3
Epoch 80/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0066 - accuracy: 0.2
Epoch 81/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0069 - accuracy: 0.2
Epoch 82/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0066 - accuracy: 0.2
Epoch 83/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 84/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0067 - accuracy: 0.4
Epoch 85/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0066 - accuracy: 0.3
Epoch 86/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.2
Epoch 87/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0064 - accuracy: 0.3
Epoch 88/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0066 - accuracy: 0.3
Epoch 89/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 90/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0064 - accuracy: 0.3
Epoch 91/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 92/200
```

```
151/151 [==============================] - 1s 5ms/step - loss: 0.0066 - accuracy: 0.3
Epoch 93/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0064 - accuracy: 0.3
Epoch 94/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 95/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 96/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0064 - accuracy: 0.3
Epoch 97/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.3
Epoch 98/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0062 - accuracy: 0.3
Epoch 99/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0060 - accuracy: 0.3
Epoch 100/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0063 - accuracy: 0.3
Epoch 101/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0060 - accuracy: 0.3
Epoch 102/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0063 - accuracy: 0.3
Epoch 103/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0060 - accuracy: 0.3
Epoch 104/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0060 - accuracy: 0.3
Epoch 105/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0060 - accuracy: 0.3
Epoch 106/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0061 - accuracy: 0.3
Epoch 107/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0058 - accuracy: 0.3
Epoch 108/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0058 - accuracy: 0.3
Epoch 109/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0058 - accuracy: 0.3
Epoch 110/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0056 - accuracy: 0.3
Epoch 111/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0058 - accuracy: 0.3
Epoch 112/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0056 - accuracy: 0.3
Epoch 113/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0057 - accuracy: 0.3
Epoch 114/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0057 - accuracy: 0.3
Epoch 115/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0055 - accuracy: 0.3
Epoch 116/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0055 - accuracy: 0.4
Epoch 117/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0054 - accuracy: 0.3
Epoch 118/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0057 - accuracy: 0.3
Epoch 119/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0054 - accuracy: 0.3
Epoch 120/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0055 - accuracy: 0.3
Epoch 121/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0052 - accuracy: 0.3
Epoch 122/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0054 - accuracy: 0.3
Epoch 123/200
```

```
Epoch 123/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0049 - accuracy: 0.3
Epoch 124/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0050 - accuracy: 0.3
Epoch 125/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0051 - accuracy: 0.3
Epoch 126/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0051 - accuracy: 0.3
Epoch 127/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0051 - accuracy: 0.3
Epoch 128/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0049 - accuracy: 0.3
Epoch 129/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0050 - accuracy: 0.3
Epoch 130/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0048 - accuracy: 0.3
Epoch 131/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0049 - accuracy: 0.4
Epoch 132/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0049 - accuracy: 0.3
Epoch 133/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0047 - accuracy: 0.3
Epoch 134/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0049 - accuracy: 0.3
Epoch 135/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0047 - accuracy: 0.4
Epoch 136/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0045 - accuracy: 0.4
Epoch 137/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0046 - accuracy: 0.3
Epoch 138/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0046 - accuracy: 0.3
Epoch 139/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0048 - accuracy: 0.4
Epoch 140/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0042 - accuracy: 0.3
Epoch 141/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0044 - accuracy: 0.4
Epoch 142/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0044 - accuracy: 0.4
Epoch 143/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0042 - accuracy: 0.4
Epoch 144/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0042 - accuracy: 0.4
Epoch 145/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0044 - accuracy: 0.4
Epoch 146/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0043 - accuracy: 0.4
Epoch 147/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0042 - accuracy: 0.3
Epoch 148/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0040 - accuracy: 0.4
Epoch 149/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0042 - accuracy: 0.4
Epoch 150/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0039 - accuracy: 0.4
Epoch 151/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0041 - accuracy: 0.4
Epoch 152/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0039 - accuracy: 0.4
Epoch 153/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0038 - accuracy: 0.4
```

```
Epoch 154/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0040 - accuracy: 0.4
Epoch 155/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0038 - accuracy: 0.3
Epoch 156/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0036 - accuracy: 0.4
Epoch 157/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0038 - accuracy: 0.4
Epoch 158/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0036 - accuracy: 0.4
Epoch 159/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0036 - accuracy: 0.4
Epoch 160/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0038 - accuracy: 0.4
Epoch 161/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0034 - accuracy: 0.4
Epoch 162/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0036 - accuracy: 0.4
Epoch 163/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0035 - accuracy: 0.5
Epoch 164/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0033 - accuracy: 0.4
Epoch 165/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0036 - accuracy: 0.4
Epoch 166/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0035 - accuracy: 0.4
Epoch 167/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0033 - accuracy: 0.4
Epoch 168/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0033 - accuracy: 0.4
Epoch 169/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0031 - accuracy: 0.4
Epoch 170/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0034 - accuracy: 0.4
Epoch 171/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0030 - accuracy: 0.4
Epoch 172/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0031 - accuracy: 0.4
Epoch 173/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0030 - accuracy: 0.4
Epoch 174/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0029 - accuracy: 0.4
Epoch 175/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0029 - accuracy: 0.4
Epoch 176/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0030 - accuracy: 0.4
Epoch 177/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0029 - accuracy: 0.5
Epoch 178/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0030 - accuracy: 0.4
Epoch 179/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0028 - accuracy: 0.4
Epoch 180/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0026 - accuracy: 0.4
Epoch 181/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0028 - accuracy: 0.3
Epoch 182/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0028 - accuracy: 0.4
Epoch 183/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0026 - accuracy: 0.4
Epoch 184/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0026 - accuracy: 0.4
```

```
Epoch 185/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0028 - accuracy: 0.4
Epoch 186/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0028 - accuracy: 0.4
Epoch 187/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0028 - accuracy: 0.4
Epoch 188/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0027 - accuracy: 0.4
Epoch 189/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0026 - accuracy: 0.4
Epoch 190/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0025 - accuracy: 0.4
Epoch 191/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0025 - accuracy: 0.5
Epoch 192/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0025 - accuracy: 0.4
Epoch 193/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0026 - accuracy: 0.4
Epoch 194/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0024 - accuracy: 0.4
Epoch 195/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0024 - accuracy: 0.5
Epoch 196/200
151/151 [==============================] - 1s 5ms/step - loss: 0.0025 - accuracy: 0.4
```

Make predictions with the trained model

Plot Multi-step, Multi-feature predictions.

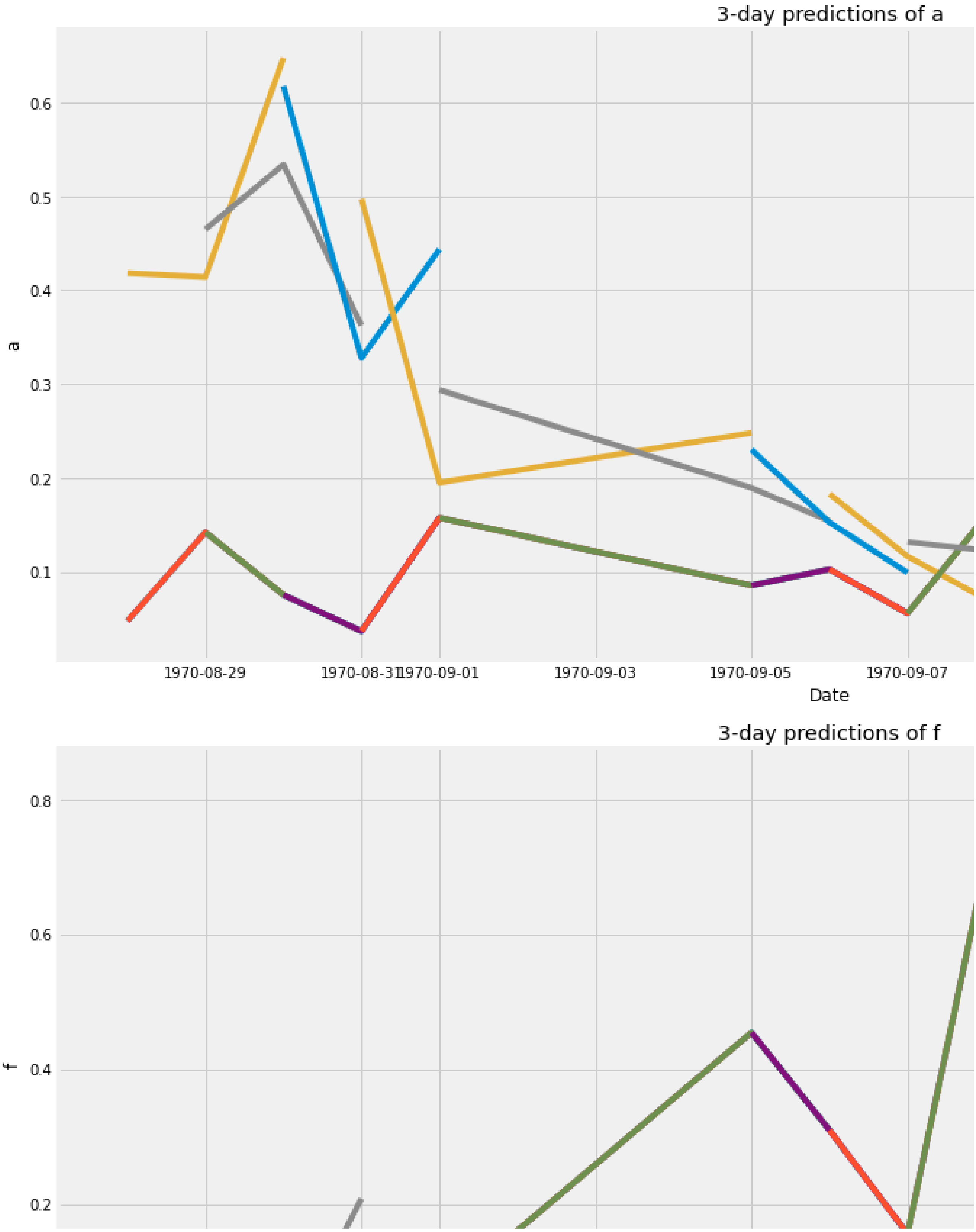
```
new_sigma.index.name="day"
new_sigma[165:][["a", "f"]].index.name
```

```
'day'
```

To read to graph below

- Each short line segment is a 3-day prediction: start, middle, end point of the line segment m
  data respectively.
- X axies are the data.
- The long line is the real data.

Plot predictions



Data Analysis

Though the dataset is not big enough, we still successfully capture several features in the predicti

- **Model predictions shows similar trend as the real data**, e.g. from the prec more or less in the most correct range and goes in the same direction as the real data.
- **The model captures the range of the real data very precisely**.
- **All 3-day predictions are liquid**, which means the modell successfully captures tl

Advanced model: Generative Adversarial Network (GAN

Generative Adversarial Networks (GAN) have been a successful model in genera
The idea that GANs can to used to predict time-series data is new and e
in learning characteristics of data, our model is based on the **assumptions**.

- Values of a **feature has certain patterns** and behavior (characteristics).
- **The future values of a feature should follow more or less the same pa** operating in a totally different way, or the economy drastically changes).

Our **goal** is that

- Generate future data that has similar (surely not exactly the same) distribution as the histori

In our model, we use

- **LSTM as a time-series generator**.
- **1-dimensional CNN as a discriminator**.

imports

Data preparation

Make sure all data forms are as what we want

```
(151, 12, 4)
(151, 6)
(66, 12, 4)
(66, 6)
```

Model architecture: LSTM generator

It's a 1-layer LSTM model.

- 50 hidden layers of LSTM cells
- 1 dense layer with 6 (2*3) dimensional output, since we have 2 features and 3 months to pre

Create generator

```
Model: "sequential_9"
```

Model architecture: CNN discriminator

The structure of the discriminator is given by

- Reshape layer. Each row in y_train is acturally 1-dimensional (6,), which is different from (6,1
- 1-dimensional Convolutional layer with 32, 3×1 filters to capture the characteristics of 3-mc
- LeakyReLU layer
- Dropout layer. Random reconfigurate 10% of the weights to zero to prevent overfitting.
- 1-dimensional Convolutional layer with 64, 3×1 filters to capture more characteristics of the
- Batchnormalization layer. To normalize the data.
- 1 Dense layer with 50 hidden nets.
- Dropout layer.
- 1 Dense layer with 1 net.

Create discriminator

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
reshape_1 (Reshape)          (None, 6, 1)              0         
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 4, 32)             128       
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 4, 32)             0         
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 32)             0         
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 2, 64)             6208      
_________________________________________________________________
batch_normalization_1 (Batch (None, 2, 64)             256       
_________________________________________________________________
dense_10 (Dense)             (None, 2, 50)             3250      
_________________________________________________________________
dropout_2 (Dropout)          (None, 2, 50)             0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 100)               0         
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 101       
=================================================================
Total params: 9,943
Trainable params: 9,815
Non-trainable params: 128
_________________________________________________________________
```

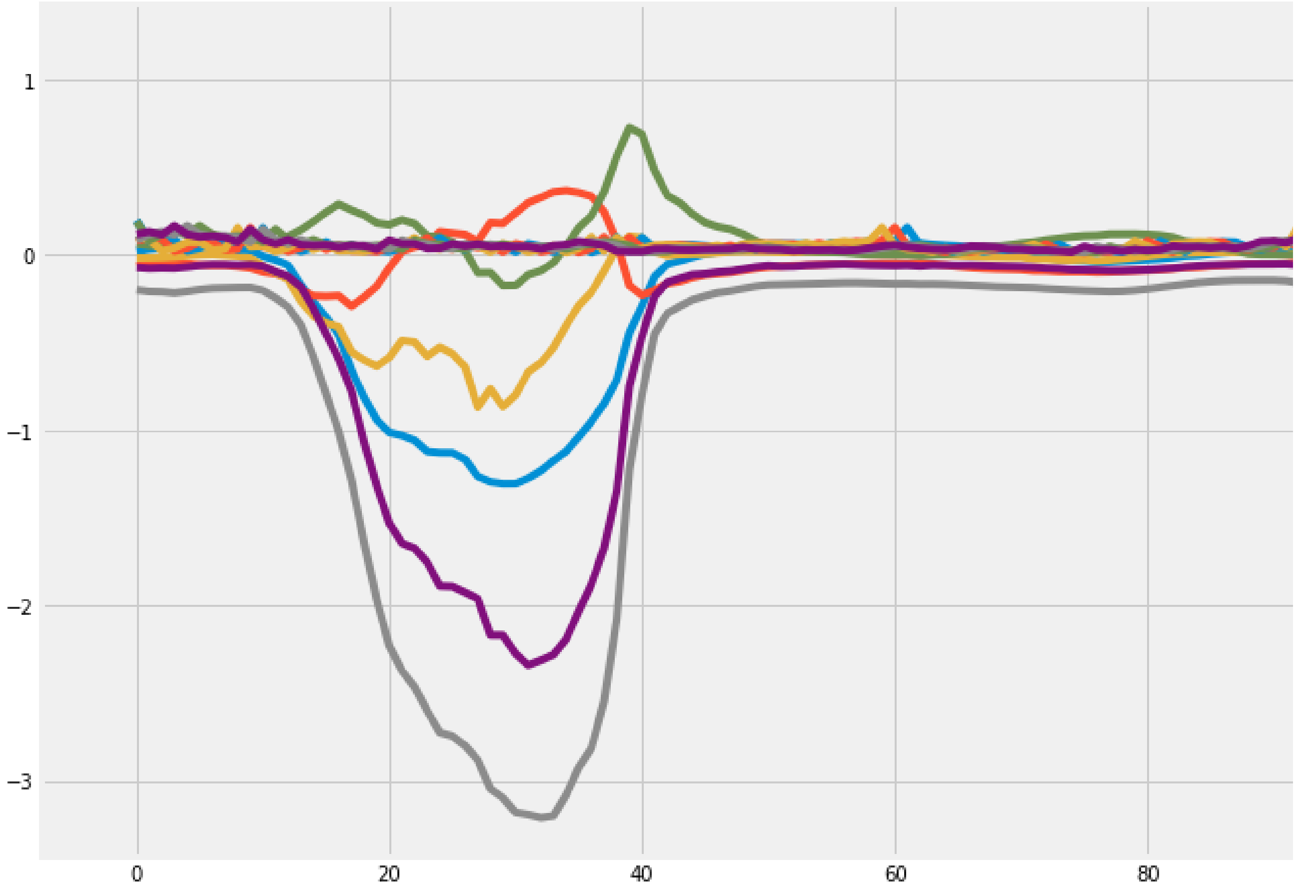
Create a GAN model with LSTM as the generator and CNN as the discriminatc

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         (None, 12, 4)             0         
_________________________________________________________________
sequential_9 (Sequential)    (None, 6)                 11306     
_________________________________________________________________
sequential_10 (Sequential)   (None, 1)                 9943      
=================================================================
Total params: 21,249
```

Training function for the entangled GAN model

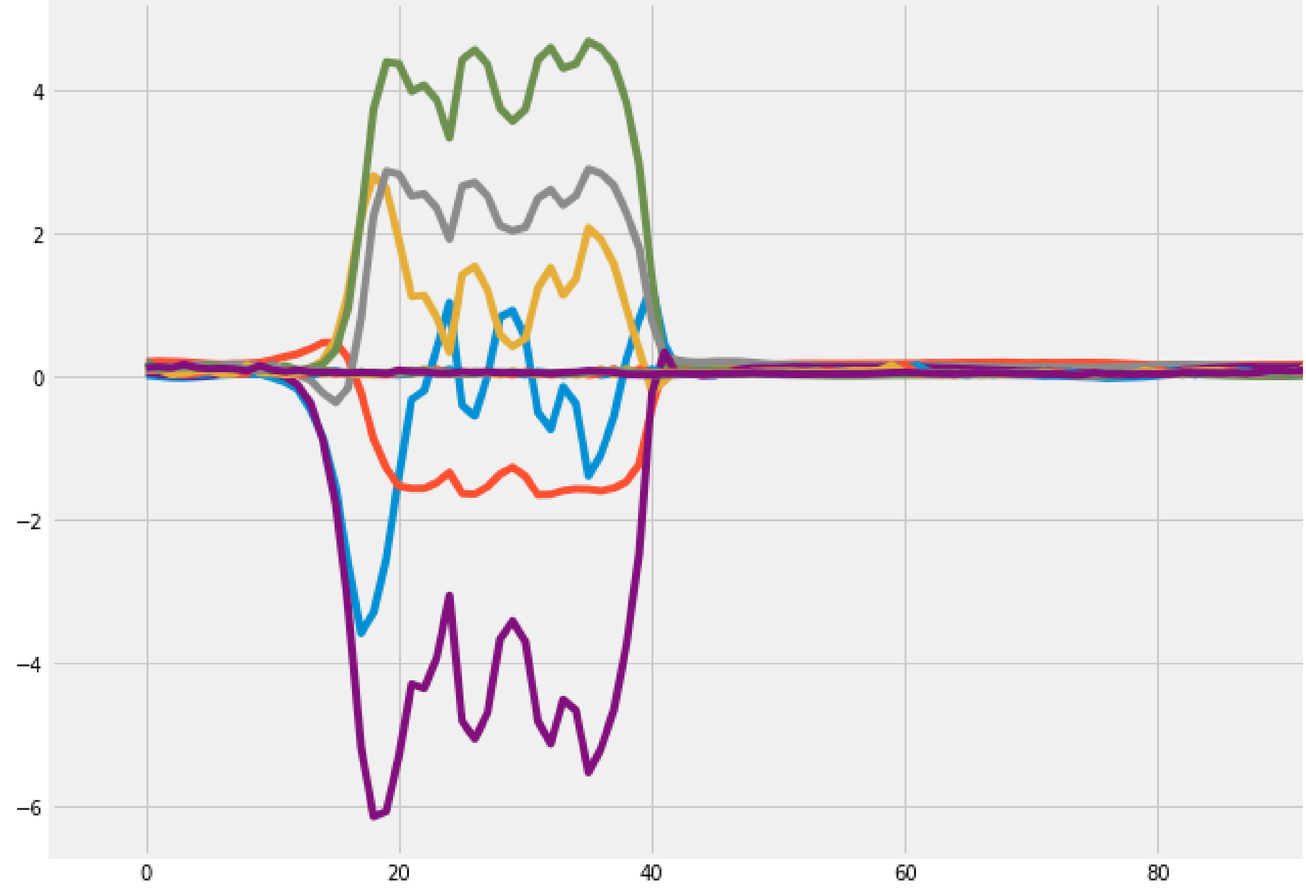
```
training(x_train, y_train, x_test, y_test, epochs=100, random_size=128)
```

```
  0%|          | 0/128 [00:00<?, ?it/s]Epoch 1
100%|██████████| 128/128 [00:05<00:00, 23.11it/s]
```



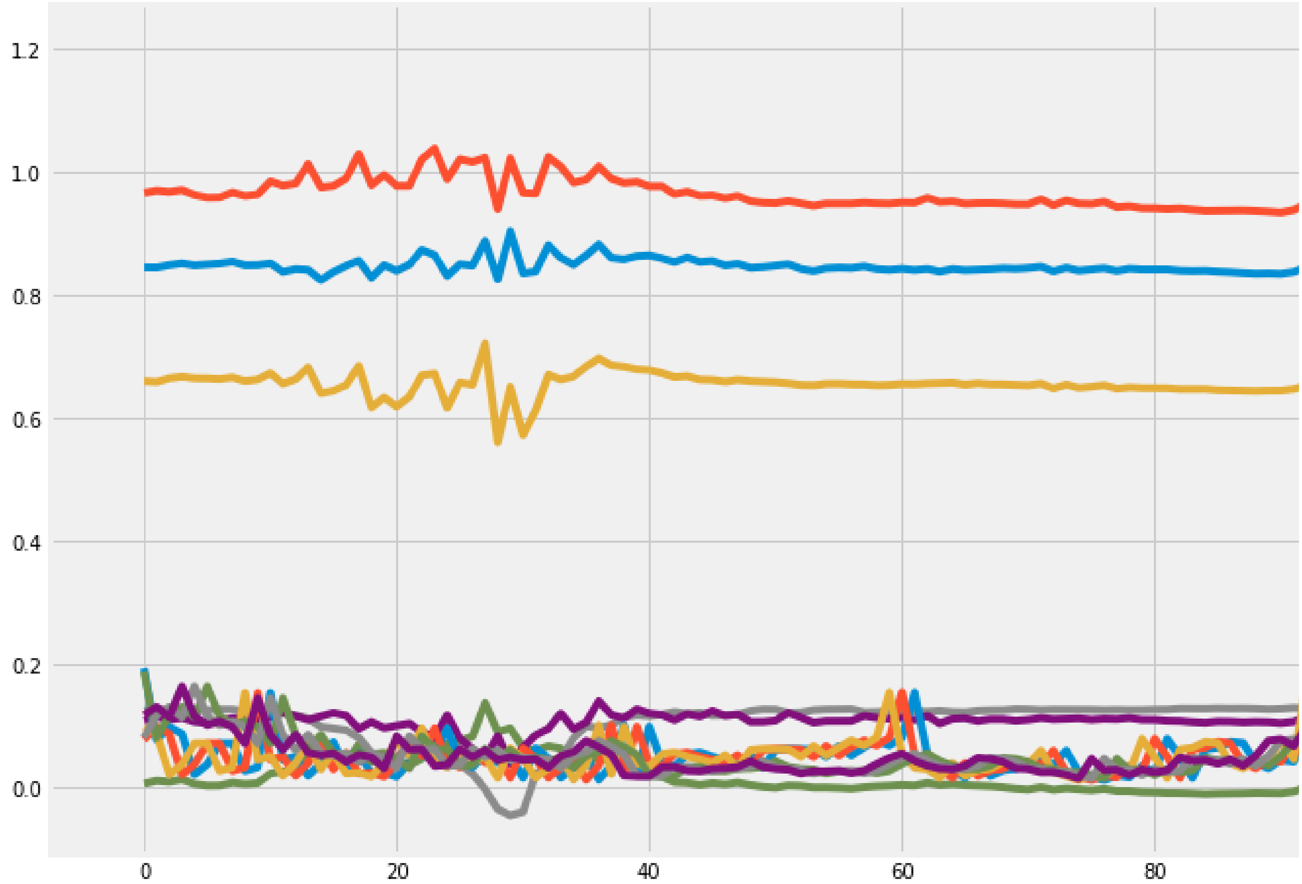

```
  3%|▎         | 4/128 [00:00<00:03, 37.64it/s]Epoch 2
100%|██████████| 128/128 [00:03<00:00, 37.76it/s]
  4%|▍         | 5/128 [00:00<00:03, 40.91it/s]Epoch 3
100%|██████████| 128/128 [00:03<00:00, 41.67it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.84it/s]Epoch 4
100%|██████████| 128/128 [00:03<00:00, 41.45it/s]
  3%|▎         | 4/128 [00:00<00:03, 32.14it/s]Epoch 5
100%|██████████| 128/128 [00:03<00:00, 39.96it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.94it/s]Epoch 6
100%|██████████| 128/128 [00:03<00:00, 40.29it/s]
  3%|▎         | 4/128 [00:00<00:03, 39.50it/s]Epoch 7
100%|██████████| 128/128 [00:03<00:00, 40.92it/s]
  4%|▍         | 5/128 [00:00<00:02, 43.49it/s]Epoch 8
100%|██████████| 128/128 [00:03<00:00, 40.92it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.87it/s]Epoch 9
100%|██████████| 128/128 [00:03<00:00, 40.65it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.18it/s]Epoch 10
100%|██████████| 128/128 [00:03<00:00, 41.37it/s]
  4%|▍         | 5/128 [00:00<00:03, 39.72it/s]Epoch 11
100%|██████████| 128/128 [00:03<00:00, 37.44it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.11it/s]Epoch 12
100%|██████████| 128/128 [00:03<00:00, 38.12it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.90it/s]Epoch 13
100%|██████████| 128/128 [00:03<00:00, 41.85it/s]
  4%|▍         | 5/128 [00:00<00:02, 44.00it/s]Epoch 14
100%|██████████| 128/128 [00:03<00:00, 41.29it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.53it/s]Epoch 15
100%|██████████| 128/128 [00:03<00:00, 39.90it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.51it/s]Epoch 16
100%|██████████| 128/128 [00:03<00:00, 42.45it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.41it/s]Epoch 17
100%|██████████| 128/128 [00:03<00:00, 39.65it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.56it/s]Epoch 18
```

```
  4%|▌         | 5/128 [00:00<00:02, 42.56it/s]Epoch 18
100%|██████████| 128/128 [00:03<00:00, 41.50it/s]
  4%|▌         | 5/128 [00:00<00:02, 42.77it/s]Epoch 19
100%|██████████| 128/128 [00:03<00:00, 40.81it/s]
  4%|▌         | 5/128 [00:00<00:02, 41.28it/s]Epoch 20
100%|██████████| 128/128 [00:03<00:00, 41.24it/s]
```



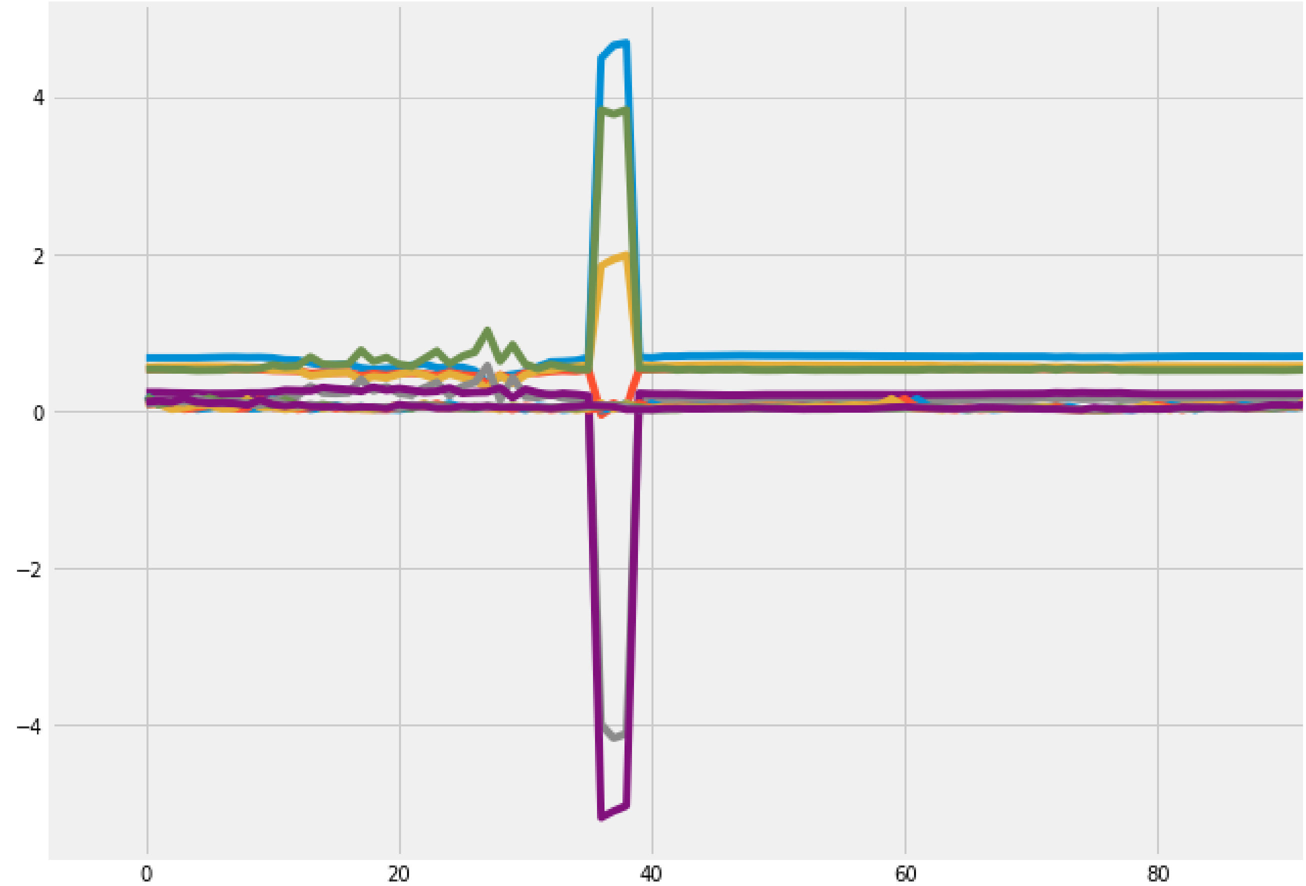

```
  3%|▎         | 4/128 [00:00<00:03, 33.53it/s]Epoch 21
100%|██████████| 128/128 [00:03<00:00, 39.88it/s]
  4%|▌         | 5/128 [00:00<00:02, 43.50it/s]Epoch 22
100%|██████████| 128/128 [00:02<00:00, 42.70it/s]
  4%|▌         | 5/128 [00:00<00:02, 42.98it/s]Epoch 23
100%|██████████| 128/128 [00:03<00:00, 42.25it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.68it/s]Epoch 24
100%|██████████| 128/128 [00:03<00:00, 40.34it/s]
  4%|▌         | 5/128 [00:00<00:02, 43.45it/s]Epoch 25
100%|██████████| 128/128 [00:03<00:00, 41.41it/s]
  4%|▌         | 5/128 [00:00<00:02, 44.05it/s]Epoch 26
100%|██████████| 128/128 [00:03<00:00, 41.71it/s]
  4%|▌         | 5/128 [00:00<00:02, 44.32it/s]Epoch 27
100%|██████████| 128/128 [00:03<00:00, 40.61it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.03it/s]Epoch 28
100%|██████████| 128/128 [00:03<00:00, 39.91it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.03it/s]Epoch 29
100%|██████████| 128/128 [00:03<00:00, 39.73it/s]
  4%|▌         | 5/128 [00:00<00:03, 40.64it/s]Epoch 30
100%|██████████| 128/128 [00:03<00:00, 37.71it/s]
  3%|▎         | 4/128 [00:00<00:03, 33.80it/s]Epoch 31
100%|██████████| 128/128 [00:03<00:00, 35.07it/s]
  3%|▎         | 4/128 [00:00<00:03, 34.70it/s]Epoch 32
100%|██████████| 128/128 [00:03<00:00, 38.11it/s]
  4%|▌         | 5/128 [00:00<00:02, 41.00it/s]Epoch 33
100%|██████████| 128/128 [00:03<00:00, 39.58it/s]
  4%|▌         | 5/128 [00:00<00:02, 42.68it/s]Epoch 34
100%|██████████| 128/128 [00:03<00:00, 41.53it/s]
  4%|▌         | 5/128 [00:00<00:02, 42.22it/s]Epoch 35
100%|██████████| 128/128 [00:03<00:00, 39.74it/s]
```

```
100%|██████████| 128/128 [00:03<00:00, 39.74it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.35it/s]Epoch 36
100%|██████████| 128/128 [00:03<00:00, 39.22it/s]
  3%|▎         | 4/128 [00:00<00:03, 39.66it/s]Epoch 37
100%|██████████| 128/128 [00:03<00:00, 38.23it/s]
  4%|▍         | 5/128 [00:00<00:03, 40.71it/s]Epoch 38
100%|██████████| 128/128 [00:03<00:00, 40.52it/s]
  3%|▎         | 4/128 [00:00<00:03, 33.66it/s]Epoch 39
100%|██████████| 128/128 [00:03<00:00, 40.66it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.31it/s]Epoch 40
100%|██████████| 128/128 [00:03<00:00, 39.02it/s]
```




```
  2%|▏         | 3/128 [00:00<00:04, 26.65it/s]Epoch 41
100%|██████████| 128/128 [00:03<00:00, 39.33it/s]
  3%|▎         | 4/128 [00:00<00:03, 37.57it/s]Epoch 42
100%|██████████| 128/128 [00:03<00:00, 39.86it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.49it/s]Epoch 43
100%|██████████| 128/128 [00:03<00:00, 40.71it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.21it/s]Epoch 44
100%|██████████| 128/128 [00:03<00:00, 40.78it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.02it/s]Epoch 45
100%|██████████| 128/128 [00:03<00:00, 39.94it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.19it/s]Epoch 46
100%|██████████| 128/128 [00:03<00:00, 39.95it/s]
  4%|▍         | 5/128 [00:00<00:03, 40.44it/s]Epoch 47
100%|██████████| 128/128 [00:03<00:00, 39.89it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.02it/s]Epoch 48
100%|██████████| 128/128 [00:03<00:00, 39.67it/s]
  4%|▍         | 5/128 [00:00<00:02, 43.02it/s]Epoch 49
100%|██████████| 128/128 [00:03<00:00, 42.01it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.23it/s]Epoch 50
100%|██████████| 128/128 [00:03<00:00, 40.16it/s]
  3%|▎         | 4/128 [00:00<00:03, 39.63it/s]Epoch 51
100%|██████████| 128/128 [00:03<00:00, 38.74it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.41it/s]Epoch 52
100%|██████████| 128/128 [00:03<00:00, 40.42it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.86it/s]Epoch 53
```

```
  3%|▍         | 4/128 [00:00<00:03, 36.86it/s]Epoch 53
100%|██████████| 128/128 [00:03<00:00, 41.04it/s]
  3%|▎         | 4/128 [00:00<00:03, 38.98it/s]Epoch 54
100%|██████████| 128/128 [00:03<00:00, 41.70it/s]
  3%|▎         | 4/128 [00:00<00:03, 39.79it/s]Epoch 55
100%|██████████| 128/128 [00:03<00:00, 41.92it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.28it/s]Epoch 56
100%|██████████| 128/128 [00:03<00:00, 39.36it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.31it/s]Epoch 57
100%|██████████| 128/128 [00:03<00:00, 41.88it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.88it/s]Epoch 58
100%|██████████| 128/128 [00:02<00:00, 42.78it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.72it/s]Epoch 59
100%|██████████| 128/128 [00:03<00:00, 38.82it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.77it/s]Epoch 60
100%|██████████| 128/128 [00:03<00:00, 41.20it/s]
```




```
  3%|▎         | 4/128 [00:00<00:03, 39.08it/s]Epoch 61
100%|██████████| 128/128 [00:03<00:00, 40.54it/s]
  3%|▎         | 4/128 [00:00<00:03, 37.99it/s]Epoch 62
100%|██████████| 128/128 [00:03<00:00, 41.31it/s]
  4%|▍         | 5/128 [00:00<00:03, 40.69it/s]Epoch 63
100%|██████████| 128/128 [00:03<00:00, 40.69it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.81it/s]Epoch 64
100%|██████████| 128/128 [00:03<00:00, 40.68it/s]
  4%|▍         | 5/128 [00:00<00:02, 42.98it/s]Epoch 65
100%|██████████| 128/128 [00:03<00:00, 39.06it/s]
  3%|▎         | 4/128 [00:00<00:03, 36.15it/s]Epoch 66
100%|██████████| 128/128 [00:03<00:00, 40.39it/s]
  4%|▍         | 5/128 [00:00<00:02, 41.26it/s]Epoch 67
100%|██████████| 128/128 [00:03<00:00, 41.84it/s]
  4%|▍         | 5/128 [00:00<00:02, 43.44it/s]Epoch 68
100%|██████████| 128/128 [00:03<00:00, 40.21it/s]
  4%|▍         | 5/128 [00:00<00:03, 40.99it/s]Epoch 69
100%|██████████| 128/128 [00:03<00:00, 42.35it/s]
  4%|▍         | 5/128 [00:00<00:03, 39.97it/s]Epoch 70
100%|██████████| 128/128 [00:03<00:00, 40.61it/s]
```