# Market Data downloading

**HKEX_NO_ADJ: hkex_no_adjust**

Download data:
these codes are in *code_download_hkex_noadj* folder

1. step1: find max date in MarketDataUpdate

   *code: findingDayRangeInMarketData.py*

2. step2: download data from bloomberg

   *code: downloadNoAdj.py*

   only do part 1 can download data.

3. Step 3: change format of data

   *code: changeFormatHkex.py*

   this is because the data uploaded should be the format like that.

| date | value | name | adjust | feature | category |
|---|---|---|---|---|---|
| 1531094400 | 3795637 | 1 | no | volume | Equity |
| 1531180800 | 5213465 | 1 | no | volume | Equity |
| 1531267200 | 3915574 | 1 | no | volume | Equity |
| 1531353600 | 3752584 | 1 | no | volume | Equity |
| 1531440000 | 3224003 | 1 | no | volume | Equity |
| 1531699200 | 2975163 | 1 | no | volume | Equity |
| 1531785600 | 4710196 | 1 | no | volume | Equity |
| 1531872000 | 6448673 | 1 | no | volume | Equity |
| 1531958400 | 5624953 | 1 | no | volume | Equity |
| 1532044800 | 4253826 | 1 | no | volume | Equity |
| 1532304000 | 2796570 | 1 | no | volume | Equity |
| 1532390400 | 2496260 | 1 | no | volume | Equity |
| 1532476800 | 3031068 | 1 | no | volume | Equity |
| 1532563200 | 3132711 | 1 | no | volume | Equity |
| 1532649600 | 2522199 | 1 | no | volume | Equity |
| 1532908800 | 3427961 | 1 | no | volume | Equity |
| 1532995200 | 5616575 | 1 | no | volume | Equity |
| 1533081600 | 4944113 | 1 | no | volume | Equity |
| 1533168000 | 4922733 | 1 | no | volume | Equity |
| 1533254400 | 7092837 | 1 | no | volume | Equity |
| 1533513600 | 11569709 | 1 | no | volume | Equity |
| 1533600000 | 9417109 | 1 | no | volume | Equity |
| 1533686400 | 9893970 | 1 | no | volume | Equity |
| 1533772800 | 5882226 | 1 | no | volume | Equity |
| 1533859200 | 4075646 | 1 | no | volume | Equity |

But data downloaded from bloomberg is like

| date | ticker | PX_LAST | PX_OPEN | PX_HIGH | PX_LOW | PX_VOLUME | TURNOVER | SHORT_SELL_NUM_SHARES | SHORT_SELL_TURNOVER | CUR_MKT_CAP |
|---|---|---|---|---|---|---|---|---|---|---|
| 2018-07-06 | 1 HK Equity | 84.1 | 83.8 | 84.7 | 83 | 6120971 | 514085900 | 553000 | 46444720 | 324430.761 |
| 2018-07-09 | 1 HK Equity | 84.9 | 84.8 | 85.5 | 84.55 | 3795637 | 322324000 | 178000 | 15128180 | 327516.904 |
| 2018-07-10 | 1 HK Equity | 85.3 | 84.9 | 86 | 84.6 | 5213465 | 445408300 | 87000 | 7445400 | 329059.97 |
| 2018-07-11 | 1 HK Equity | 84.25 | 82.9 | 84.4 | 82.9 | 3915574 | 327822400 | 608000 | 51003680 | 325009.413 |
| 2018-07-12 | 1 HK Equity | 83.75 | 83.5 | 85 | 83.5 | 3752584 | 315043900 | 344500 | 28975580 | 323080.574 |
| 2018-07-13 | 1 HK Equity | 84.2 | 84 | 84.4 | 83.7 | 3224003 | 271203000 | 143000 | 12035850 | 324816.529 |
| 2018-07-16 | 1 HK Equity | 84.3 | 84 | 84.45 | 83.5 | 2975163 | 249850900 | 297000 | 24949720 | 325202.297 |
| 2018-07-17 | 1 HK Equity | 83.4 | 84 | 84.15 | 83.4 | 4710196 | 394455400 | 371500 | 31159450 | 321730.386 |
| 2018-07-18 | 1 HK Equity | 82.65 | 83.4 | 83.6 | 82.6 | 6448673 | 534954800 | 751500 | 62362350 | 318837.12 |
| 2018-07-19 | 1 HK Equity | 83.2 | 83.2 | 83.5 | 82.6 | 5624953 | 467270700 | 547500 | 45473920 | 320958.851 |
| 2018-07-20 | 1 HK Equity | 83.75 | 83.65 | 83.85 | 83.1 | 4253826 | 355441300 | 1020000 | 85221180 | 323080.574 |
| 2018-07-23 | 1 HK Equity | 83.45 | 83.75 | 84.2 | 83.05 | 2796570 | 233637300 | 321500 | 26883000 | 321923.270 |
| 2018-07-24 | 1 HK Equity | 83.85 | 83 | 84.2 | 83 | 2496260 | 209063400 | 431000 | 36137680 | 323466.342 |
| 2018-07-25 | 1 HK Equity | 84.2 | 84 | 84.45 | 83.95 | 3031068 | 255184900 | 347000 | 29207300 | 324816.529 |
| 2018-07-26 | 1 HK Equity | 84.3 | 85.8 | 85.8 | 84.1 | 3132711 | 265904600 | 604500 | 51436980 | 325202.297 |
| 2018-07-27 | 1 HK Equity | 84.6 | 85 | 85.25 | 84.25 | 2522199 | 213954500 | 323000 | 27366920 | 326359.601 |
| 2018-07-30 | 1 HK Equity | 85.35 | 85 | 85.5 | 84.6 | 3427961 | 291721000 | 414000 | 35223500 | 329252.8 |
| 2018-07-31 | 1 HK Equity | 85.3 | 85.35 | 85.8 | 84.7 | 5616575 | 478327200 | 494000 | 42186280 | 329059.97 |

4. step 4: upload csv file to *MongoDBcode: uploadHkexNOadjustToMongoDB.py*

I highly recommend that we download HKEX_NOADJ data through above 4 scripts, although you can only use *updatehkex-C.py*, this script. This is because we can only use

Carol's computer to download data and it takes us much time to change format of data so that other people cannot use Carol's computer.

5. Step 5: update data without overlap in MarketData

   *code: merge_hkex_without_overlap.py*

   this step is to make sure we do not update dupliacted data

**Checking data:**

- All scripts about checking are in *checking_hkex_noadj.py*
  1. check whether correctly downloaded

     we randomly retrieve a value in Bloomberg and check whether the data in MongoDB is the same

     *Function: def check_downloaded()*

  2. check whether download all the tickers

     *Function: def checkDownloaded()*

  3. check whether dtype is the same as the one in MongoDB

     *Function: def check_dtype()*

  4. find max day in MarketData and min day in MarketDataUpdate

     this step is to check max day in MarketData and min day in MarketDataUpdate. We should make sure there is no missing data.

     *Function: def findDayRange()*

  5. checking whether dataset has duplicated data

     *Function: def checking_duplicated():*

  6. if the dataset has duplicated data, drop duplicated data

     *Function: def dropDuplicated()*

- Three standard deviation principles checking data we downloaded

  The script below only check close price of no adjusted price
  *Code: roling_std.py*

**Ticker list**

1. updated list

   Actually the ticker list we use (universe 1025.csv) is not the finally updated ticker list. Therefore, when updating data, it may raise some errors.

   Final ticker list (2019. 8.9 ) contains ticker in MarketDataUpdate. It can be listed as follows:

   *File: MarketDataUpdate_list(2019.8.9).csv*

2. NO new data list

   When downloading data, some ticker have no new data and you would find that you download an empty file.

   *code: update_stock_empty_no_new-data.py*

3. **Abnormal start ticker list:**

**The list contains tickers that only have data after 2018 in Bloomberg. But in MongoDB MarketData they have previous time data. One reason is that ticker is replaced by a new company after previous one delisted, such as 2168 HK Equity, kaisa company**

| New data info | |
|---|---|
| **8096 HK Equity** | 2019.3.14—now |
| **4335 HK Equity** | no new data |
| **4336 HK Equity** | no new data |
| **6860 HK Equity** | 2018.7.11—now |
| **667 HK Equity** | 2019.6.11—now |
| **8017 HK Equity** | 2018.09.27 –now |
| **2168 HK Equity** | 2018.12.5 –now |
| **3868 HK Equity** | 2019.5.27 –now |
| **1832 HK Equity** | 2019.5.15 –now |
| **1775 HK Equity** | 2018.7.12 –now |
| **1743 HK Equity** | 2019.1.3 –now |
| **1025 HK Equity** | 2019.2.27 –now |

(now means until today，2019.8.9, they have data)

Old data info:

*Code: abnormal_start_list_info.py*

```
#667 HK Equity  2006-10-05-----------2011-12-08
#1025  2003-11-20----------2016-01-06
#1743    2007-07-06------------2008-06-16
#1775     2008-04-02---------2008-11-10
#1832    2007-07-13------------2010-05-24
#2168    this ticker is belongs to another company but it died and kasai own this ticker now, so the data start from 2018-12-05
#3868    2007-10-02-----------2017-11-29
#4335    2000-05-31----------, close price duanduanxuxu
#4336       close price duanduanxuxu
#6860     2007-09-04---------------2008-03-31
#8017       2000-08-16--------2016-10-18
#8096         2002-02-26----------2017-02-03
```

4. 358ticker_list

*File: 358ticker_list.csv*

the ticker list contains tickers have relatively large market capital
There are 4 companies delisted in the list

| Delisted ticker | |
|---|---|
| **805 HK Equity** | Up to 2018.1.30 |
| **13 HK Equity** | No data |
| **1880 HK Equity** | Up to 2017.1.27 |
| **1893 HK Equity** | Up to 2018.4.23 |

**HKEX_ADJ**

*Download data*

We do not download adjust price but calculate it.

$$\text{adjPrice} = noadjPrice * \frac{adjustmentFactorDaily}{oldA\text{djustmentFactor}}$$

$adjustmentFactorDaily$ : these data is in HKEX_AF_Daily, MarketDataUpdate, {feature: 'price_af_star'}
$adjustmentFactorDaily$ can be found in HKEX_AF_Daily, MarketDataUpdate
until now, we only use feature: price_af.

Calculating adjust price
*Code: calculate_adj_price.py*

**Checking data:**

1. Calculate error between adjprice we calculate and adj price download from Bloomberg
   *Code: finding_the_date_fix_enpowerment.py*
   This script is to find old adjustment factor and list wrong_cal_error is about this.
   After checking ticker in *358ticker_list.csv,* the result seems OK.
   *File: check_list_error.csv*
2. Checking price equal

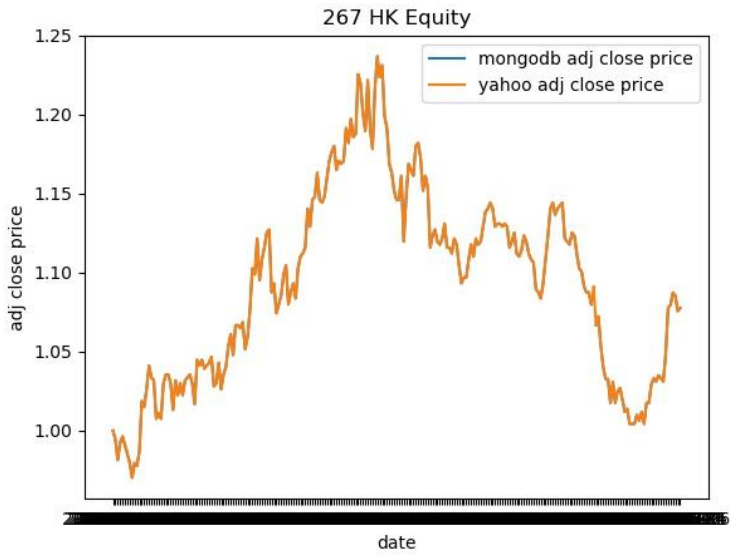
   If a stock does not have adjustment factor, it seems its adjust price is equal to unadjust price. This script is to check whether these two prices are equal.

   *Code: checking_whether_noadjustment_price_equal.py*
3. Checking the trend, compared with Yahoo Finance

   *Code: adj_price_plot_trend.py*

   If adj price is correctly calculated, the trend should be the same as data from Yahoo.

   Result can be seen as follows:



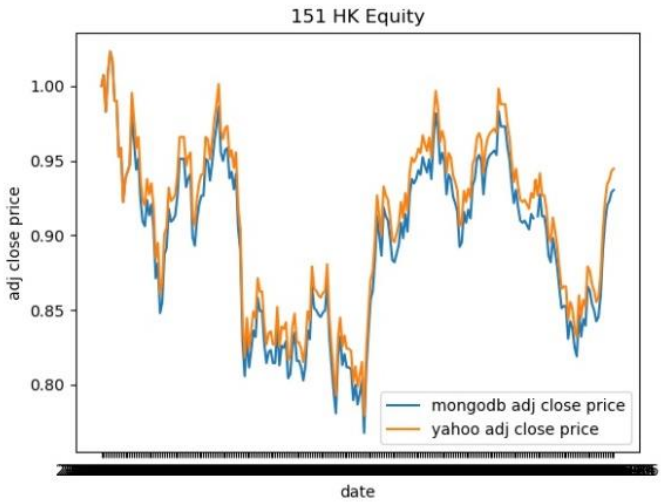

\

We can do this by randomly choose 30 samples from ticker_list. Then, manually download data from Yahoo Finance.

The way to download data from Yahoo can see this file.

*File:Download data from Yahoo Finance.docx*

during checking, we found that there is something wrong.




When checking 151 HK Equity, the trend between the price we calculated and the one in yahoo are not the same, so we should check data in bloomberg.

4. Checking the trend, compared with bloomberg

   *code: adj_compared_with_bbg.py*

   since there is monthly limited in bloomberg, we can only download the data need to be double checked.

   If the trend between ours and bloomberg’s are the same, we finish checking.

**HKEX_AF**

**Download Data**

*Code: update_hkex_adjustment_factors.py*
Also, I recommend download first and then upload.

**FX_Vol**

**Download Data**

1. Step 1: find min date in MarketData and max date in MarketDataUpdate
   *Code: findDayRangeFXVOL.py*
2. Step 2: download data in Bloomberg
   *Code: update_fx_vol_chuchu.py*

**FX**

We download FX data in Factset
**Factset usage**

*File: update_fx_about_macro.doc*
**Download Data**

1. Step 1：downolad csv file in Factset
   *Code: macro.py*
   Make sure excel has open Factset
   Make sure the path is modified in macro
   During updating data, keep *macro_for_refresh_1029.csv file* open
2. Step 2：change format
   *Code: changeFormat_FX.py*
3. Step 3: upload to MongoDB
   *Code: uploadCsvToMongoDB.py*

**Equity**

**Download Data:**
Download Equity data in Bloomberg
*Code: update_Equity.py*

**Checking data:**

All scripts about checking are in *checking_Equity.py*

- Checking whether correctly downloaded
  randomly retrieve a value in Bloomberg and check whether the data in MongoDB is the same
  *Function: def check_downloaded()*
- check whether dtype is the same as the one in MongoDB
  *Function: def check_dtype()*
- check if all the tickers have the same startDate to update from
  *Function: checkdayRange()*
- check if the downloaded data and updated data have same value on the startDate
  *Function: def check_same()*
- check whether there is duplicatd data, if there is, drop duplicated data
  *Function: def check_duplicated()*

**Equity_Vol**

Download Equity data in Bloomberg
*Code: update_equity_vol.py*