



University of St.Gallen

School of Management, Economics, Law, Social Sciences and International Affairs

# The effect of Twitter activity on Bitcoin price

## *Documentation*

Software Engineering for Economists  
(7,610,1.00)

Dimitrios Koumnakes - 10-613-370

Severin Kranz - 13-606-355

Joël Sonderegger - 11-495-488

Alen Stepic - 11-475-258

Chi Xu - XX-XXX-XXX

Fall Term 2017

Supervisor

Prof. Dr. Philipp ZAHN

Department of Economics

December 27, 2017

## Abstract

(Insert text)

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Goal of the paper . . . . .	1
1.2	Methodology . . . . .	1
1.3	Scope . . . . .	1
<b>2</b>	<b>Data Collection</b>	<b>3</b>
2.1	Tweets Data . . . . .	3
2.1.1	Python Script . . . . .	3
2.1.2	Hardware Setup . . . . .	5
2.2	Bitcoin Price Data . . . . .	5
2.2.1	Execution . . . . .	5
2.2.2	Output . . . . .	5
2.2.3	API: Bitcoinaverage.com . . . . .	6
<b>3</b>	<b>Data Aggregation</b>	<b>7</b>
<b>4</b>	<b>Data Analysis</b>	<b>7</b>

# List of Figures

1	Twitter Keys and Access Tokens . . . . .	4
---	--	---

# List of Abbreviation

<b>API</b>	Application Programming Interface
<b>CKEY</b>	Consumer Key
<b>CSECRET</b>	Consumer Secret
<b>ATOKEN</b>	Access Token
<b>ASECRET</b>	Access Token Secret

# 1 Introduction

In the academic environment accountability and reproducibility is important. However, the publishing process of papers and journals seem to be outdated. New ways of data collection and data processing exist by using computational economics. The usage of algorithms can increase effectiveness and efficiency. Hence, much larger data sets can be processed. However this creates also new problems regarding to traceability and reproducibility. Often cited academic papers exist, where the initial computation is not reproducible. Furthermore, some academic papers even contain computational errors. Replicating data or existing results do not provide any new knowledge at all. Nevertheless, the ability to reproduce increases trustworthiness and indicates the quality of the conducted work. This explains why reproduction is of great relevance.

## 1.1 Goal of the paper

The goal of this documentation is the provision of a description. This description should enable the reader to reproduce the results discussed in the separate paper. Thus, it contains an explanation how the input data have been gathered, stored, aggregated and analysed. In other words, the input data, the model core, the model parameters and the applied math program are explained.

## 1.2 Methodology

This documentation consists out of four chapters. The first chapter contains a short introduction and provides the reader with an overview about the topic. Furthermore it points out the relevance of documentation. The second chapter discusses the input data. This includes the process of gathering and storing twitter tweets as well as the gathering of the bitcoin price data. The third chapter discusses how the data is aggregated by pointing out the core model and its parameters. Finally the fourth chapter discusses how the analysis has been conducted.

## 1.3 Scope

The scope of the documentation is the provision of an overview about the different steps which have been conducted to obtain the results in the paper. It does not contain any discussions about the results of the separate paper. It is not a deep description of the code as the code itself as the code is documented separately. Nevertheless, important lines of code are discussed.

## 2 Data Collection

Here, we provide a detailed description of how the data for the sequential analysis is gathered and stored. This includes two subsections the (1) tweets data and the (2) bitcoin price data.

### 2.1 Tweets Data

Whit the python script real-time twitter data are streamed and stored. This happens with help of a raspberry pi.

#### 2.1.1 Python Script

Twitter offers different Application Programming Interfaces (API) for collecting data. However, the time frame for gathering data on a free base is limited to 7 days. On the other hand, python offers different twitter libraries. Such as the open-source package tweepy. This package has been used for streaming the twitter data as it simplify the script.

#### Installing Tweepy

Tweepy can be installed very simple by running following commands in the command prompt.

---

```
pip install tweepy
```

---

If the previous downloaded python installation package does not contain the tweepy library, the tweepy library has to be downloaded. The package can be downloaded for free from the following link:

---

<https://pypi.python.org/pypi/tweepy>

---

#### Twitter Authentication

To access the twitter data, twitter requests an identification of the user. The identification is assured by different keys and access tokens. Those are the (1) consumer key (ckey), (2) consumer secret (csecret), (3) access token (atoken) and (4) access token secret (asecret).

To get the mentioned key and tokens a twitter account is needed. Once, an twitter account exist an application has to be created. This has to be conducted by login with the twitter account credentials under the following link:

---

<https://apps.twitter.com/>

---

After the creation of an application the keys and tokens can be extracted. Figure 1 illustrates

how to retrieve the keys and tokens.

The screenshot shows the 'Keys and Access Tokens' page for an application named 'hsgsoftengineering'. The page has a navigation bar with 'Details', 'Settings', 'Keys and Access Tokens' (highlighted with a red box), and 'Permissions'. A 'Test OAuth' button is in the top right. The main section is titled 'Application Settings' with a warning: 'Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.' Below this, a red box highlights the 'Consumer Key (API Key)' and 'Consumer Secret (API Secret)' fields. The 'Access Level' is 'Read and write (modify app permissions)', the 'Owner' is 'HSGSoftEngineer', and the 'Owner ID' is '933214447291604992'. An 'Application Actions' section contains 'Regenerate Consumer Key and Secret' and 'Change App Permissions' buttons. A horizontal scrollbar is visible. The next section is 'Your Access Token' with a warning: 'This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.' A red box highlights the 'Access Token' and 'Access Token Secret' fields. The 'Access Level' is 'Read and write', the 'Owner' is 'HSGSoftEngineer', and the 'Owner ID' is '933214447291604992'.

**hsgsoftengineering** Test OAuth

[Details](#) [Settings](#) **Keys and Access Tokens** [Permissions](#)

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)

Consumer Secret (API Secret)

Access Level: Read and write ([modify app permissions](#))

Owner: HSGSoftEngineer

Owner ID: 933214447291604992

**Application Actions**  
Regenerate Consumer Key and Secret Change App Permissions

---

### Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token

Access Token Secret

Access Level: Read and write

Owner: HSGSoftEngineer

Owner ID: 933214447291604992

Figure 1: Twitter Keys and Access Tokens

## Twitter Streaming API

By running the python script `collectTwitterData.py` real-time twitter data is pushed in a JSON format. Tweets are pushed just in case the tweet contains the defined key word `bitcoin`.

```
$ python collectTwitterData.py
```

From the JSON format the following parameters are decoded

- created at: Timestamp of the created tweet
- text: text of the tweet

The time timestamps is UTC time.

### 2.1.2 Hardware Setup

(Severin's Part)

## 2.2 Bitcoin Price Data

We wrote a Python script which collects Bitcoin price data as there was no preexisting data set that satisfied our needs. The Bitcoin price is best expressed by the Bitcoin Price Index. The Bitcoin price index (BPI) is an index of the exchange rate between the Bitcoin (BTC) and the US dollar (USD) (**kristoufek2015main**). The objective of the script was to gather hourly Bitcoin Price Index data for at least the time period in which we gather the tweets data. We found the an API by bitcoinaverage.com which sufficed our needs. An API description follows later.

### 2.2.1 Execution

**Twitter** By executing the python script `collectCryptocurrencyData.py` hourly data for the Bitcoin Price Index is retrieved.

---

```
$ python collectCryptocurrencyData.py
```

---

### 2.2.2 Output

After successfully running the python script `CollectCryptocurrencyData.py` the file `bpi.csv` is generated in the folder `/data`. It is important to note that every execution of the script overwrites any existing `bpi.csv` file.

The file `bpi.csv` contains historical Bitcoin Price Index data for one month on an hourly basis. Each data point consists of the following parameters:

- time: Timestamp on an hourly basis in UTC time
- average: Average price (in USD)
- high: Highest price (in USD)
- low: Lowest price (in USD)
- open: Opening price (in USD)



### 2.2.3 API: Bitcoinaverage.com

Bitcoinaverage.com offers a free API that provides real-time and historical price data for a range of crypto-currencies including Bitcoin. The following requests delivers data for an per hour monthly sliding window.

#### Request

The request to get the data for an per hour monthly sliding window looks as follows. This request require authentication that requires registration and the generation of an API key. The registration and generation of an API key is freely available on bitcoinaverage.com. The collectCryptocurrencyData.py already contains the necessary keys. This means that you need no register or generate keys to execute the script collectCryptocurrencyData.py.

---

<https://apiv2.bitcoinaverage.com/indices/global/history/BTCUSD?period=monthly&format=json>

---

**Response** An excerpt of an example response looks like the following:

---

```
[
  {
    "high": 8271.04,
    "average": 8247.83,
    "open": 8242.39,
    "low": 8217.72,
    "time": "2017-11-22 15:00:00"
  },
  {
    "high": 8246.82,
    "average": 8203.19,
    "open": 8203.81,
    "low": 8157.25,
    "time": "2017-11-22 14:00:00"
  },
  {
    "high": 8267.27,
    "average": 8238.62,
    "open": 8248.77,
    "low": 8198.54,
    "time": "2017-11-22 13:00:00"
  }
]
```

---

### **3 Data Aggregation**

After the two data sets (tweets and Bitcoin Price Index) are gathered

### **4 Data Analysis**

(Dimitri's Part)