Looking under the hood: Investigating the blockchain with R

Bernhard Pfaff
bernhard_pfaff@fra.invesco.com

Invesco Asset Management GmbH
Frankfurt am Main

DSF-R, WU Vienna, 13 and 14 September 2018
Prologue

Who knows, who this person is?

Prologue
A reminiscence to Vienna & the Austrian School

Friedrich August von Hayek
(* 8 May 1899 in Vienna; † 23 March 1992 in Freiburg im Breisgau).

Overview

- BTC: Charts & Statistics
- BTC: Blockchain Primer
- R package rbtc
- Block Analysis
- Summary
  - item Appendix
BTC: Charts & Statistics

Fade away of a frenzy?

BTC price progression
Sample from 2017−09−15 to 2018−06−30
Source: https://coinmarketcap.com
USD per BTC
2017−10−01 2018−01−01 2018−04−01 2018−07−01
5000 10000 15000 20000
Nerve bundles should better stay away . . .
BTC: Charts & Statistics

Returns and Volatility

BTC continuous returns

Sample from 2017-09-15 to 2018-06-30
Source: https://coinmarketcap.com

BTC squared continuous returns

Sample from 2017-09-15 to 2018-06-30
Source: https://coinmarketcap.com
BTC: Charts & Statistics

Daily Returns: Descriptive Statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>-22.512</td>
</tr>
<tr>
<td>Maximum</td>
<td>18.458</td>
</tr>
<tr>
<td>Mean</td>
<td>-0.196</td>
</tr>
<tr>
<td>Median</td>
<td>-0.373</td>
</tr>
<tr>
<td>StdDev</td>
<td>5.341</td>
</tr>
<tr>
<td>Skewness</td>
<td>-0.067</td>
</tr>
<tr>
<td>Kurtosis (excess)</td>
<td>1.644</td>
</tr>
</tbody>
</table>
BTC: Blockchain Primer

Key Items

- A distributed (P2P) ledger (double-entry bookkeeping) system.
- Bitcoin is an open source project, MIT license.
- Network started in 2009 (*genesis block*).
- Three net environments: mainnet, testnet, regtest.
- Three broad transaction types: P2PKH (P2PK), P2SH & SegWit (P2WPKH).
- Transactions are validated, collected in blocks and are confirmed by solving a cryptographic puzzle (Proof-of-Work).
- First transaction in a block is termed *coinbase*: the miner’s revenue (transaction fees plus mining reward).
- Common bitcoin denomination (units):
  1. 1 cBTC = 0.01 BTC (a hundreth),
  2. 1 mBTC = 0.001 BTC (a thousandth),
  3. 1 (μ)uBTC = 0.000001 BTC (one millionth),
  4. 1 Satoshi = 1 sat = 0.00000001 BTC (one hundred millionth).
Transactions in Detail

- Double-entry bookkeeping: Sum of input values (debits) equals sum of output values (credits) plus a transaction fee.
- Transactions are chained: only outputs from a previous transaction (UTXOs) can be used as inputs in a new transaction (discrete and indivisible units of value).
- Content of transaction output: a) amount of bitcoin, b) length script in bytes and c) locking script itself (witness script / scriptPubKey).
- Content of transaction input: a) transaction hash, b) index number of the UTXO in the former transaction, c) length script in bytes, d) unlocking script (scriptSig) and e) sequence number (locktime).
- Transactions are broadcasted as serialized byte-streams.
BTC: Blockchain Primer

Transactions in progress

BTC: Blockchain Primer
Blocks in Detail

- Content of a block: a) size of block (bytes), b) block header, c) count of transactions and d) transactions in the block.
- Content of a block header: a) version of the software protocol, b) previous block hash, c) the Merkle Root, d) timestamp, e) difficulty target and f) the nonce.
- Approximately every 10 minutes a block is mined.
- Every 2,016 blocks the difficulty target is adjusted to meet this mining time goal.
BTC: Blockchain Primer
Blocks: Content and Chain

R package rbtc

Structure

- Purely written in R and utilizes S4-classes and methods.
- Dependencies to the packages http (see Wickham, 2017) and rjson (see Couture-Beil, 2014) for conducting calls to and receiving responses from the RPC-JSON API.
- Dependencies to openssl (see Ooms, 2018) and gmp (see Lucas et al., 2017) for creating BTC addresses.
- Requirement(s): bitcoin-core/bitcoind; configuration file (see https://en.bitcoin.it/wiki/Running_Bitcoin).
- Hosted on GitHub: https://github.com/bpfaff/rbtc/
R package rbtc

Key Functions

- `startbtc()` and `stopbtc()`:
  bitcoind can also be started/stopped from the shell, supports TOR.

- `conrpc()`:
  creating a connection object, used in http posts.

- `getblockhash()`:
  hash of a block for a provided height.

- `getblock()`:
  block’s content (three levels of verbosity).

- `getrawtransaction()` & `decoderawtransaction()`:
  retrieving & decoding of a transaction (two levels of verbosity).

- `decodescript()`:
  decoding of a P2SH (UTXO address starts with a ’3’).
Key S4-Classes

- **CONRPC:**
  object for establishing calls to JSON-RPC.

- **ANSRPC:**
  objects returned by calls to JSON-RPC.

- **ECPARAM and ECPPOINT:**
  elliptic curve computations.

- **BTCADR:**
  object containing private & public keys, WIF, public hash and BTC address – should not be used on mainnet, only for exemplary purposes.
R package rbtc

Utility Functions

- Functions for coercing integer to date times and *vice versa*: `int2date()`, `date2int()`, `intMinDay()`, `intMaxDay()`, `intRangeDay()`, `intRangePeriod()`.

- Functions for aggregating/retrieving information contained in blocks/transactions: `blockstats()`, `txstats()`, `txfee()`, `txids()`, `txinids()`, `utxoage()`, `utxovalue()`, `timeofblock()`, `blockattime()`.

- Functions/methods & operators related to cryptographic algorithms: `ecpoint()`, `doubleUp()`, `+`, `*`, `AND`, `leftmostBit()`, `createPrivateKey()`, `PrivKey2Wif()`, `Wif2PrivKey()`, `PrivKey2PubKey()`, `PubKey2PubHash()`, `PubHash2BtcAdr()`, `createBtcAdr()`, `concatHex()`, `hash256()`, `hash160()`, `base58CheckEncode()`, `base58CheckDecode()`, `decodeHex()`.
Sample period from 09/15/2017 until 06/30/2018.

This corresponds to the block heights from 485,295 until 529,950; a total of 44,6656 blocks.

Empty blocks and coinbase transactions are excluded from the analysis (see https://news.bitcoin.com).

For saving/retrieval of intermediate results the Rpackages **RSQLite** (see Müller et al., 2018) and **DBI** (see R Special Interest Group on Databases (R-SIG-DB) et al., 2018) are utilized.

Time series aggregation with R package **timeSeries** (see Würtz et al., 2017, in memoriam Diethelm!).
Block Analysis
Gearing up . . .

> ## Code to show on slide: retrieving blockstats (DBI)
Block Analysis

When the crowd kicked in . . .
Block Analysis

UTXO: Cameron & Tyler, was that you?
Chasing the Coins: BTC’s Black Friday was on 22nd December 2017

- Biggest price swing intra-day ever since.
- Analysing the vintage of the traded UTXOs in the blocks on that day.
- Can trading patterns/behaviour be detected?
- A data science field experiment for behavioral finance?
Chasing the Coins: BTCs Black Friday 2017/12/22

```r
> sblock <- blockattime("2017-12-22 00:00:00")[2, 1]
> eblock <- blockattime("2017-12-22 23:59:59")[1, 1]
> bseq <- sblock:eblock
> n <- length(bseq)
> ## Creating SQLite-table
> coinage <- data.frame("Height" = integer(),
+ "Time" = integer(),
+ "AgeMin" = numeric(),
+ "AgeMax" = numeric(),
+ "AgeMean" = numeric(),
+ "AgeMedian" = numeric()
+ )
> dbCreateTable(conlite,
+ name = "coinage",
+ fields = coinage)
> ## Looping through the blocks
> for (i in 1:n){
+ h <- slot(getblockhash(con, bseq[i]),
+ "result")
+ b <- slot(getblock(con, h),
+ "result")
+ btime <- b["time"]
+ txsinblock <- unlist(b["tx"])[-1] # ex coinbase
+ k <- length(txsinblock)
+ if (k > 0){ # mind the empty blocks
+   vage <- c()
+   for (j in 1:k){
```
```
inputage <- uxtxage(con, txsinblock[j])
vage <- c(vage, inputage[["AgeInput"]])
}
ans <- data.frame("Height" = bseq[i],
                  "Time" = btime,
                  "AgeMin" = min(vage),
                  "AgeMax" = max(vage),
                  "AgeMean" = mean(vage),
                  "AgeMedian" = stats::median(vage),
                  )
dbAppendTable(conlite,
              name = "coinage",
              value = ans)
}
Have a look at this transaction:

```r
> txid <- ""
> (tx <- slot(getrawTransaction(con, ),
+    "result"))
```

Something suspicious with utxo address 17EGi2sxaBtfceqANacKkeHVvoKXgAbnjc?

Let's check:

1. Apply base 58 check decoding
2. Drop last four bytes, *i.e.*, take first 20 bytes
3. Convert from raw to character format . . .
> y <- "17EGi2sxAbtfceqANacKkeHVvoKXgAbnjc"
> (r1 <- base58CheckDecode(y)[1:20])
[1] 44 53 46 2d 52 20 56 69 65 6e 66 61 3a 20 47 72 65 61 74 21
> (r2 <- rawToChar(r1))
[1] "DSF-R Vienna: Great!"

## ... and in the opposite direction
> x <- "DSF-R Vienna: Great!"  # must be 20 characters long
> (s <- paste(charToRaw(x), collapse = ""))
[1] "4453462d522056696e6613a20477265617421"
> (se <- c(decodeHex("00"), decodeHex(s)))
[1] 00 44 53 46 2d 52 20 56 69 65 6e 66 61 3a 20 47 72 65 61 74 21
> (cs <- hash256(se)[1:4])
[1] 6b dc 1b bb
> (pkh <- c(se, cs))
[1] 00 44 53 46 2d 52 20 56 69 65 6e 66 61 3a 20 47 72 65 61 74 21 6b dc 1b bb
> (y <- base58CheckEncode(pkh))
[1] "17EGi2sxAbtfceqANacKkeHVvoKXgAbnjc"

Wahrschau! The amount send to this address is lost, unless you find the corresponding private key, good luck!
See Ken Shiriff’s blog for more examples.
Summary

- A tool for investigating the blockchain and deriving descriptive statistics.
- The package can be utilized for data science research in finance (viewed as a field experiment: e.g., market micro structure analysis and/or behavioral finance).
- Complementary to the R packages rbitcoin (see Gorecki, 2014) and coindeskr (see AbdulMajedRaja, 2018), i.e. linking summary statistics derived from the block chain with BTC price history.
- Package is hosted on CRAN, R-Forge & GitHub — your choice.

Thank You!
**Appendix**

**Glossary**

- **BIP**: Bitcoin improvement proposal.
- **ECDSA**: Elliptic Curve Digital Signature Algorithm.
- **P2P**: Peer-to-Peer Network Architecture.
- **P2PK & P2PKH**: Pay-to-Public-Key(-Hash).
- **P2SH**: Pay-to-Script-Hash.
- **P2WPKH**: Pay-to-Witness-Public-Key-Hash.
- **satoshi**: smallest bitcoin unit.
- **TOR**: The Onion Router.
- **TXID**: Transaction identifier.
- **UTXO**: unspent transaction outputs (vout field JSON output from RPC-API).
- **WIF**: Wallet Import Format.
Appendix

Session Information

> sessionInfo()
R version 3.5.1 (2018-07-02)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.1 LTS

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
[1] LC_CTYPE=de_DE.UTF-8 LC_NUMERIC=C
[3] LC_TIME=de_DE.UTF-8 LC_COLLATE=de_DE.UTF-8
[5] LC_MONETARY=de_DE.UTF-8 LC_MESSAGES=de_DE.UTF-8
[7] LC_PAPER=de_DE.UTF-8 LC_NAME=C

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] timeSeries_3042.102 timeDate_3043.102 DBI_1.0.0 [4] rbtc_0.1-4

loaded via a namespace (and not attached):
[1] httr_1.3.1 compiler_3.5.1 rjson_0.2.20 R6_2.2.2 gmp_0.5-13.2 [6] openssl_1.0.2


Appendix (contd.)

Bibliography


