

Bitcoin Notebook Hello World

November 15, 2015

This project connects [Rusty Russell's bitcoin-iterate](#) to the Jupyter iPython Notebook. The `bi()` Python function is a small glue layer between `bitcoin-iterate` and the notebook software.

Using this tool you can quickly and easily make `bitcoin-iterate` queries, process the results somehow, and then graph them in a way that is reproduceable and literate.

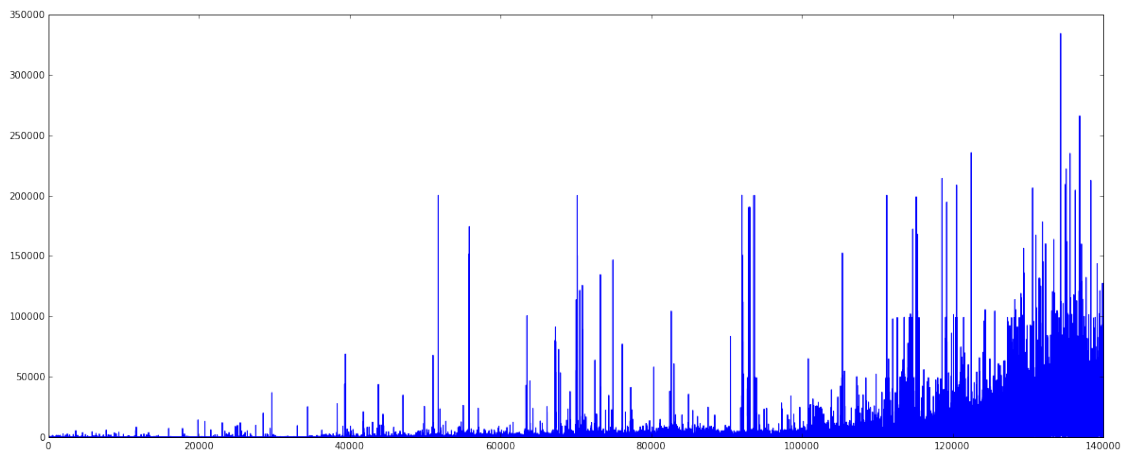
The source code is available at [bitcoin-notebook on GitHub](#).

```
In [86]: from bi import bi
import matplotlib.pyplot as plt
%matplotlib inline
# import bi; reload(bi)
```

Basic graph of Block length over time (up to block 140000).

```
In [78]: result = bi("--block %b1 --end=140000")

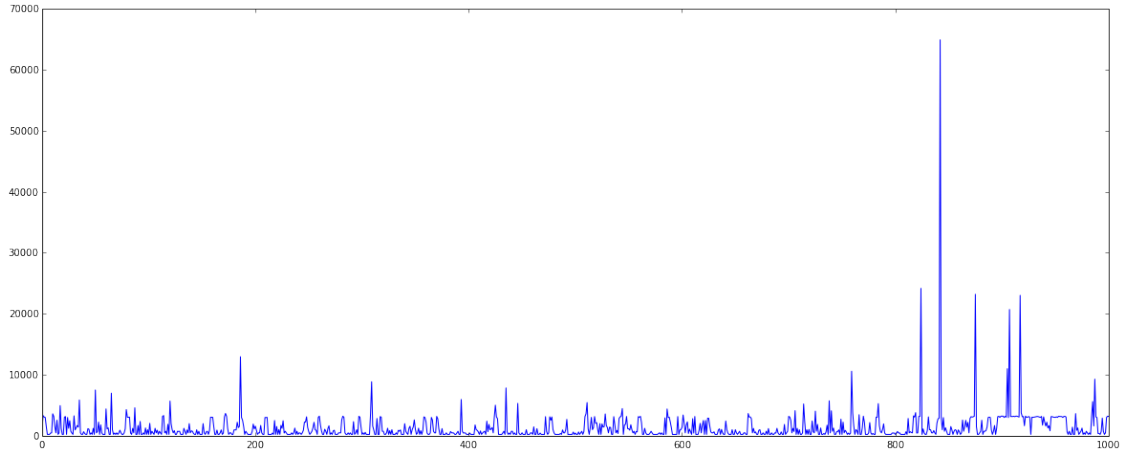
plt.figure(figsize=(20,8))
plt.plot(result["results"]["%b1"], label="%b1")
plt.show()
```



Example of a smaller chunk of block lengths.

```
In [79]: result = bi("--block %b1 --start=100000 --end=101000")

plt.figure(figsize=(20,8))
plt.plot(result["results"]["%b1"], label="%b1")
plt.show()
```



Example query showing datetime conversion to Python native.

```
In [80]: bi("--block %bv,%bs,%bt,%bn --start=140000 --end=140001")["results"]

Out[80]: {'%bn': [1789739685, 3704812052],
          '%bs': [datetime.datetime(2011, 8, 8, 1, 49, 38),
                  datetime.datetime(2011, 8, 8, 2, 1, 37)],
          '%bt': [436789733, 436789733],
          '%bv': [1, 1]}
```

Output amount.

```
In [81]: query = bi("--output %oa --start=140000 --end=140001")["results"]
          # first five values from each bucket
          [{r: query[r][:5]} for r in query]

Out[81]: [{'%oa': [5015835346, 47981000000, 11825000000, 42548000000, 260144584612]}]
```

Transaction nLockTime and fee paid.

```
In [82]: query = bi("--tx %tt,%tF --start=140000 --end=140001")["results"]
          # first five values from each bucket
          [{r: query[r][:5]} for r in query]

Out[82]: [{'%tF': [-5015835346, 0, 1000000, 0, 0]}, {'%tt': [0, 0, 0, 0, 0]}]
```

0.1 Some of the original bitcoin-iterate examples

Block hash and transaction size for some transactions in the main chain.

```
In [83]: query = bi("--tx=%bh,%t1 --start=140000 --end=140001")["results"]
          # first five values from each bucket
          [{r: query[r][:5]} for r in query]

Out[83]: [{'%bh': [u'db97f2fe217299bd8205663dc526ecf0de6600a81747cf286e08000000000000',
                    u'db97f2fe217299bd8205663dc526ecf0de6600a81747cf286e08000000000000',
                    u'db97f2fe217299bd8205663dc526ecf0de6600a81747cf286e08000000000000',
                    u'db97f2fe217299bd8205663dc526ecf0de6600a81747cf286e08000000000000',
                    u'db97f2fe217299bd8205663dc526ecf0de6600a81747cf286e08000000000000'],
          '%t1': [135, 258, 1123, 3462, 258]}]
```

Output script sizes (block number, transaction number, output number, output script length).

```
In [84]: query = bi("--output=%bN,%tN,%oN,%o1 --start=140000 --end=140001")["results"]
# first five values from each bucket
[{'r': query[r][:5]} for r in query]
```

```
Out[84]: [{'bN': [140000, 140000, 140000, 140000, 140000]},
{'tN': [0, 1, 1, 2, 3]},
{'oN': [0, 0, 1, 0, 0]},
{'o1': [67, 25, 25, 25, 25]}]
```

Five largest blocks, by height and blockhash.

```
In [85]: bi("--block=%b1,%bN,%bh | sort -nr | head -n5")["results"]
```

```
Out[85]: {'bN': [128827, 131795, 127045, 136995, 98504],
'bh': ['u'7297375319170694731115145927ffa109cfe2fa09661be9611d0000000000000',
u'7839335585705819301155468bbc248811f89a90f3d47420980d0000000000000',
u'74586167633485278438783ee2b634ea1d539666b9ed74192125000000000000',
u'5042380824809401107599e32d0c8781aa6b2092fcb8976ade01000000000000',
u'72903606314217367967233c81da5565cb2ffe35d28c18026141000000000000'],
'b1': [18164, 30615, 4370, 16167, 2103]}
```

Enjoy!