R Markdown Output

Last run on: 2020-12-16 09:57:13

2020-12-16 09:57:13

Overview

This document has code embedded throughout. In the next section we will create a visualization using the already loaded dataset `eth_data`:

```
datatable(eth_data)
```

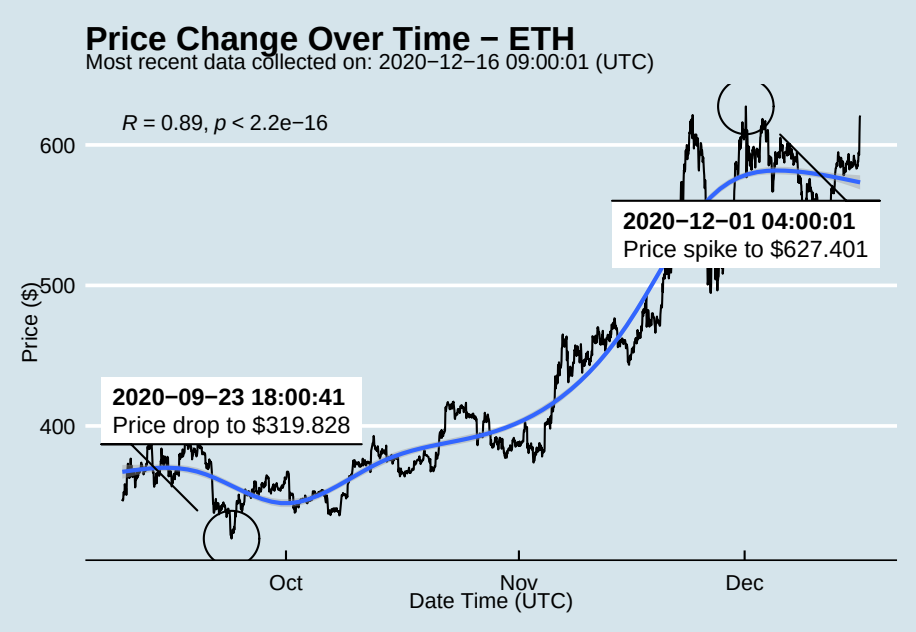
Show 10 entries Search:

	pair	symbol	ask_1_price	date_time_utc
1	ETHUSD	ETH	620.942	2020-12-16T16:00:01Z
2	ETHUSD	ETH	619.786	2020-12-16T15:00:01Z
3	ETHUSD	ETH	608.544	2020-12-16T14:00:01Z
4	ETHUSD	ETH	592.865	2020-12-16T13:00:01Z
5	ETHUSD	ETH	593.411	2020-12-16T12:00:01Z
6	ETHUSD	ETH	594.547	2020-12-16T11:00:01Z
7	ETHUSD	ETH	588.045	2020-12-16T10:00:01Z
8	ETHUSD	ETH	588.269	2020-12-16T09:00:01Z
9	ETHUSD	ETH	584.837	2020-12-16T08:00:01Z
10	ETHUSD	ETH	585.739	2020-12-16T07:00:02Z

Showing 1 to 10 of 2,207 entries Previous 1 2 3 4 5 … 221 Next

The table above uses the **DT** package (Xie, Cheng, and Tan 2020). The **bookdown** package (Xie 2016) is packed with functionality, including in-text citations that automatically generate the references as we have done here.

Price Chart - Ethereum



Python Code Example

```
import pandas as pd
# Create the Python object from R
df = r.cryptodata
# Show the new Python dataframe
df
```

```
##         pair symbol  ask_1_price       date_time_utc
## 0     ETHUSD    ETH      620.942 2020-12-16 16:00:01
## 1     BTCUSD    BTC    20658.910 2020-12-16 16:00:00
## 2     BTCUSD    BTC    20660.760 2020-12-16 15:00:01
## 3     ETHUSD    ETH      619.786 2020-12-16 15:00:01
## 4     BTCUSD    BTC    20327.440 2020-12-16 14:00:01
## ...      ...    ...          ...                 ...
## 5285  BTCUSD    BTC    11972.900 2020-08-10 06:03:50
## 5286  BTCUSD    BTC    11985.890 2020-08-10 05:03:48
## 5287  BTCUSD    BTC    11997.470 2020-08-10 04:32:55
## 5288  BTCUSD    BTC    10686.880                 NaT
## 5289  ETHUSD    ETH      357.844                 NaT
##
## [5290 rows x 4 columns]
```

One more Python example

The code below creates a new column `price_percentile` that specifies if the price for the row was in the upper or lower 50th percentile of prices (BTC should be upper and ETH lower):

```
import numpy as np
# Create a new column based on the ask_1_price value:
df['price_percentile'] = np.where(df['ask_1_price'] >
                                  np.percentile(df['ask_1_price'], 50),
```

```
                            'upper 50th percentile of prices',
                            'lower 50th percentile of prices')
# Show modified dataframe:
df[['symbol', 'ask_1_price', 'price_percentile']]
```

```
##      symbol  ask_1_price                price_percentile
## 0       ETH      620.942  lower 50th percentile of prices
## 1       BTC    20658.910  upper 50th percentile of prices
## 2       BTC    20660.760  upper 50th percentile of prices
## 3       ETH      619.786  lower 50th percentile of prices
## 4       BTC    20327.440  upper 50th percentile of prices
## ...     ...          ...                              ...
## 5285    BTC    11972.900  upper 50th percentile of prices
## 5286    BTC    11985.890  upper 50th percentile of prices
## 5287    BTC    11997.470  upper 50th percentile of prices
## 5288    BTC    10686.880  upper 50th percentile of prices
## 5289    ETH      357.844  lower 50th percentile of prices
##
## [5290 rows x 3 columns]
```

bookdown Examples

- See this example for a more complex bookdown document which updates automatically every 12 hours using the same tools as this example.
- Supervised Machine Learning for Text Analysis in R
- JavaScript for R

Find more examples published through the bookdown website: https://bookdown.org/home/archive/

Xie, Yihui. 2016. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman; Hall/CRC. https://github.com/rstudio/bookdown.

Xie, Yihui, Joe Cheng, and Xianying Tan. 2020. *DT: A Wrapper of the JavaScript Library 'DataTables'*. https://CRAN.R-project.org/package=DT.