

R Markdown Output

Last run on: 2021-02-16 06:29:27

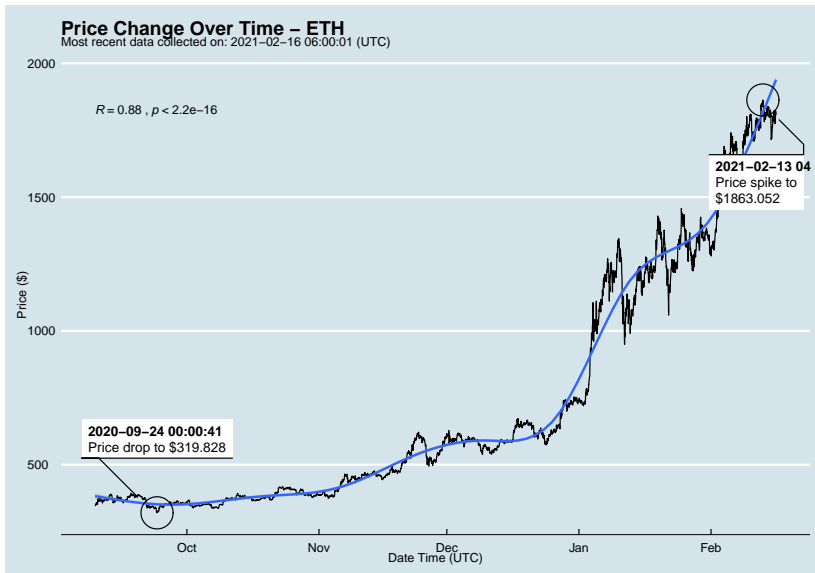
2021-02-16 06:29:27

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)
```

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      1808.255 2021-02-16 06:00:01
## 1      BTCUSD   BTC     49188.800 2021-02-16 06:00:00
## 2      ETHUSD   ETH      1812.365 2021-02-16 05:00:01
## 3      BTCUSD   BTC     49344.590 2021-02-16 05:00:00
## 4      BTCUSD   BTC     49612.760 2021-02-16 04:00:01
## ...      ...      ...           ...           ...
## 8241 BTCUSD   BTC     11972.900 2020-08-10 06:03:50
## 8242 BTCUSD   BTC     11985.890 2020-08-10 05:03:48
## 8243 BTCUSD   BTC     11997.470 2020-08-10 04:32:55
## 8244 BTCUSD   BTC     10686.880                NaT
## 8245 ETHUSD   ETH       357.844                NaT
```

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_p
                                  'upper 50th percentile of price
                                  'lower 50th percentile of price

# Show modified dataframe:
df[['symbol', 'ask_1_price', 'price_percentile']]
```

##	symbol	ask_1_price	price_percentile
## 0	ETH	1808.255	lower 50th percentile of price
## 1	BTC	49188.800	upper 50th percentile of price
## 2	ETH	1812.365	lower 50th percentile of price
## 3	BTC	49344.590	upper 50th percentile of price
## 4	BTC	49612.760	upper 50th percentile of price