

## R Markdown Output

Last run on: 2021-04-21 06:18:41

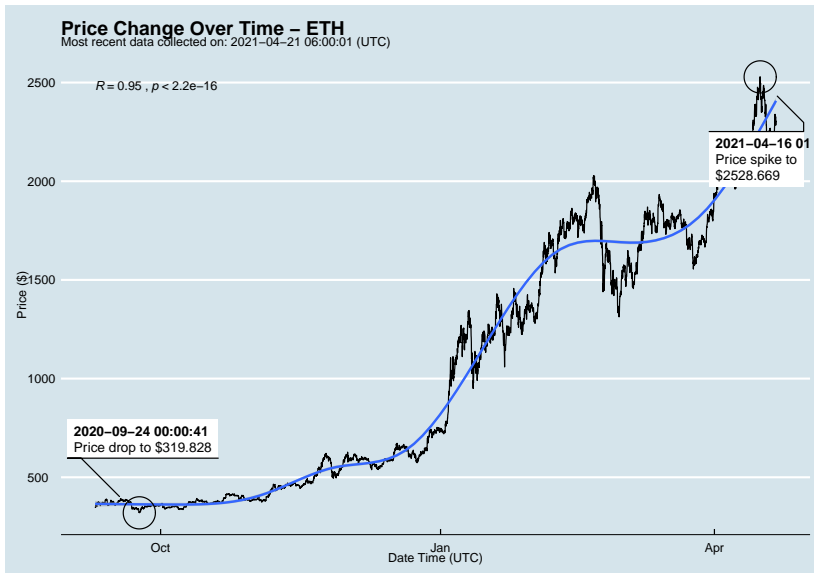
2021-04-21 06:18:41

# 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	2303.354	2021-04-21 06:00:01
## 1	BTCUSD	BTC	55386.850	2021-04-21 06:00:00
## 2	ETHUSD	ETH	2305.398	2021-04-21 05:00:01
## 3	BTCUSD	BTC	55385.080	2021-04-21 05:00:00
## 4	ETHUSD	ETH	2284.321	2021-04-21 04:00:01
## ...	...	...	...	...
## 11309	BTCUSD	BTC	11972.900	2020-08-10 06:03:50
## 11310	BTCUSD	BTC	11985.890	2020-08-10 05:03:48
## 11311	BTCUSD	BTC	11997.470	2020-08-10 04:32:55
## 11312	BTCUSD	BTC	10686.880	NaT
## 11313	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_price'],
                                                  50,
                                                  '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	0	ETH	2303.354	lower 50th percentile of price
## 1	1	BTC	55386.850	upper 50th percentile of price
## 2	2	ETH	2305.398	lower 50th percentile of price
## 3	3	BTC	55385.080	upper 50th percentile of price
## 4	4	ETH	2284.321	lower 50th percentile of price