In [1]:

```
import pandas as pd
from datetime import datetime
import time
```

In [2]:

```
# Normalization/standardization of column names
def renamingColumns(df, suffix):

    df.columns = [col.lower() for col in df.columns]
    df.columns = [col_name+'_'+suffix for col_name in df.columns]
    df.columns = [c.replace(' ', '_') for c in df.columns]
    df.rename(columns = {'date_'+suffix:'date'}, inplace = True)

    df.head

    return df
```

In [3]:

```
# Convert timestamp to date format
# Useful for making the timestamp humanely friendly

def timestampToDate(timestamp):

    return datetime.fromtimestamp(timestamp).date()
```

In [4]:

```
# Importing the data

dfOriginal = pd.read_csv('Datasets/rvnOriginal.csv')
df = dfOriginal.copy()
df.tail(10)
```

Out[4]:

	Unnamed: 0	height	size	hash	ti
1774750	182	10	262	000000574a871aca8b41b39e03cad1f00bf5e0fdcf477b...	15150159
1774751	183	9	262	000000488c4ed0a6df591c62e8a2600fc1e25690ea8ca4...	15150159
1774752	184	8	262	000000e2e9415eeeabe677f6f2f1df0e776ee5c5322f1e...	15150159
1774753	185	7	262	0000003a6463eaf8a71dbde333e61b713ed56a37eb8cac...	15150159
1774754	186	6	262	0000006d7a8490b5c4a7e41bb5549f905c75b59f902db8...	15150158
1774755	187	5	262	000000a123ae742de194b6db9f3768e2a1dcd43470912d...	15150158
1774756	188	4	262	0000002a5f3b73c4366e4d2614e62a166ed935fd8f85d2...	15150158
1774757	189	3	262	0000003a02e3de84a46ced09886c4809ebdeca4532e002...	15150158
1774758	190	2	262	000000e2aa6490a97dd26301516ce1ec3fcc5a9cea3bb3...	15150157
1774759	191	1	262	00000058bcc33dea08b53691edb9e49a9eb8bac36a0db1...	15150157

In [5]:

```
# Removing garbage
df = df.drop('Unnamed: 0', 1)
df = df.drop('poolInfo', 1)
df = df.drop('isMainChain', 1)

df = df.drop('size', 1)
df = df.drop('hash', 1)
df = df.drop('txlength', 1)
df = df.drop('minedBy', 1)

df.tail(10)
```

Out[5]:

	height	time
1774750	10	1515015971
1774751	9	1515015970
1774752	8	1515015936
1774753	7	1515015905
1774754	6	1515015847
1774755	5	1515015840
1774756	4	1515015833
1774757	3	1515015816
1774758	2	1515015759
1774759	1	1515015723

In [6]:

```
# Export dataset for RVN halving prediction
df.to_csv('Datasets/rvnHalvingPrediction.csv')
```

In [7]:

```
# Creating the dataframe for the S2F model with the flow field , stores the amount of
#    blocks generated in one day * 5000 (amount of mined coins per block)
# Important: based on days

dfS2F = df.copy()

# Handling the dates
dfS2F['date']  = dfS2F['time']
dfS2F['date']  = dfS2F['date'].apply(timestampToDate)

dfS2F['flow'] = 0
dfS2F = dfS2F.groupby('date').count() * 5000

dfS2F = dfS2F.drop('time', 1)

# Renaming columns
renamingColumns(dfS2F, 'RVN')

dfS2F
```

Out[7]:

date	height_RVN	flow_RVN
2018-01-03	5945000	5945000
2018-01-04	10945000	10945000
2018-01-05	11500000	11500000
2018-01-06	9265000	9265000
2018-01-07	13170000	13170000
...	...	...
2021-05-27	7330000	7330000
2021-05-28	7020000	7020000
2021-05-29	7130000	7130000
2021-05-30	7250000	7250000
2021-05-31	7060000	7060000

1245 rows × 2 columns

In [8]:

```
# Adding the stock variable

dfS2F['stock_RVN'] = dfS2F['flow_RVN'].cumsum()
dfS2F
```

Out[8]:

date	height_RVN	flow_RVN	stock_RVN
2018-01-03	5945000	5945000	5945000
2018-01-04	10945000	10945000	16890000
2018-01-05	11500000	11500000	28390000
2018-01-06	9265000	9265000	37655000
2018-01-07	13170000	13170000	50825000
...	...	...	...
2021-05-27	7330000	7330000	8845340000
2021-05-28	7020000	7020000	8852360000
2021-05-29	7130000	7130000	8859490000
2021-05-30	7250000	7250000	8866740000
2021-05-31	7060000	7060000	8873800000

1245 rows × 3 columns

In [9]:

```
# Importing daily price RVN-USD
# source: https://au.finance.yahoo.com/quote/RVN-USD/history?period1=1520640000&period2=162

dfPriceRVN = pd.read_csv('Datasets/RVN-USD.csv')
dfPriceRVN
```

Out[9]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-03-10	0.026499	0.028772	0.026063	0.028618	0.028618	171820.0
1	2018-03-11	0.028520	0.033503	0.026241	0.031883	0.031883	279104.0
2	2018-03-12	0.031496	0.034305	0.028595	0.030258	0.030258	218114.0
3	2018-03-13	0.029902	0.030913	0.025711	0.027902	0.027902	167669.0
4	2018-03-14	0.027723	0.028685	0.023747	0.024386	0.024386	131838.0
...	...	...	...	...	...	...	...
1174	2021-05-27	0.101504	0.103803	0.091664	0.094475	0.094475	66895929.0
1175	2021-05-28	0.094473	0.096013	0.077685	0.081287	0.081287	63652398.0
1176	2021-05-29	0.081168	0.084461	0.070398	0.076148	0.076148	46482352.0
1177	2021-05-30	0.076206	0.083790	0.070914	0.078637	0.078637	38637338.0
1178	2021-05-31	0.078666	0.084311	0.074492	0.084311	0.084311	42069239.0

1179 rows × 7 columns

In [10]:

```
# Renaming columns

renamingColumns(dfPriceRVN, 'RVN')
```

Out[10]:

	date	open_RVN	high_RVN	low_RVN	close_RVN	adj_close_RVN	volume_RVN
0	2018-03-10	0.026499	0.028772	0.026063	0.028618	0.028618	171820.0
1	2018-03-11	0.028520	0.033503	0.026241	0.031883	0.031883	279104.0
2	2018-03-12	0.031496	0.034305	0.028595	0.030258	0.030258	218114.0
3	2018-03-13	0.029902	0.030913	0.025711	0.027902	0.027902	167669.0
4	2018-03-14	0.027723	0.028685	0.023747	0.024386	0.024386	131838.0
...	...	...	...	...	...	...	...
1174	2021-05-27	0.101504	0.103803	0.091664	0.094475	0.094475	66895929.0
1175	2021-05-28	0.094473	0.096013	0.077685	0.081287	0.081287	63652398.0
1176	2021-05-29	0.081168	0.084461	0.070398	0.076148	0.076148	46482352.0
1177	2021-05-30	0.076206	0.083790	0.070914	0.078637	0.078637	38637338.0
1178	2021-05-31	0.078666	0.084311	0.074492	0.084311	0.084311	42069239.0

1179 rows × 7 columns

In [11]:

```
# Importing daily price BTC-USD
# source: https://au.finance.yahoo.com/quote/RVN-USD/history?period1=1520640000&period2=162

dfPriceBTC = pd.read_csv('Datasets/BTC-USD.csv')
dfPriceBTC
```

Out[11]:

	Date	Open	High	Low	Close	Adj Close	Volur
0	2018-03-10	9350.589844	9531.320313	8828.469727	8866.000000	8866.000000	5.386320e+
1	2018-03-11	8852.780273	9711.889648	8607.120117	9578.629883	9578.629883	6.296370e+
2	2018-03-12	9602.929688	9937.500000	8956.429688	9205.120117	9205.120117	6.457400e+
3	2018-03-13	9173.040039	9470.379883	8958.190430	9194.849609	9194.849609	5.991140e+
4	2018-03-14	9214.650391	9355.849609	8068.589844	8269.809570	8269.809570	6.438230e+
...	...	...	...	...	...	...	
1174	2021-05-27	39316.890625	40379.617188	37247.902344	38436.968750	38436.968750	4.321097e+
1175	2021-05-28	38507.082031	38856.968750	34779.039063	35697.605469	35697.605469	5.520019e+
1176	2021-05-29	35684.156250	37234.500000	33693.929688	34616.066406	34616.066406	4.523101e+
1177	2021-05-30	34607.406250	36400.667969	33520.738281	35678.128906	35678.128906	3.164608e+
1178	2021-05-31	35658.593750	37468.250000	34241.945313	37332.855469	37332.855469	3.900985e+

1179 rows × 7 columns

In [12]:

```
# Renaming columns

renamingColumns(dfPriceBTC, 'BTC')
```

Out[12]:

	date	open_BTC	high_BTC	low_BTC	close_BTC	adj_close_BTC	volume_I
0	2018-03-10	9350.589844	9531.320313	8828.469727	8866.000000	8866.000000	5.386320e
1	2018-03-11	8852.780273	9711.889648	8607.120117	9578.629883	9578.629883	6.296370e
2	2018-03-12	9602.929688	9937.500000	8956.429688	9205.120117	9205.120117	6.457400e
3	2018-03-13	9173.040039	9470.379883	8958.190430	9194.849609	9194.849609	5.991140e
4	2018-03-14	9214.650391	9355.849609	8068.589844	8269.809570	8269.809570	6.438230e
...	...	...	...	...	...	...	
1174	2021-05-27	39316.890625	40379.617188	37247.902344	38436.968750	38436.968750	4.321097e
1175	2021-05-28	38507.082031	38856.968750	34779.039063	35697.605469	35697.605469	5.520019e
1176	2021-05-29	35684.156250	37234.500000	33693.929688	34616.066406	34616.066406	4.523101e
1177	2021-05-30	34607.406250	36400.667969	33520.738281	35678.128906	35678.128906	3.164608e
1178	2021-05-31	35658.593750	37468.250000	34241.945313	37332.855469	37332.855469	3.900985e

1179 rows × 7 columns

In [13]:

```
# Adjusting the Date field for merge
dfS2F.reset_index(level=0, inplace=True)
dfS2F.date = dfS2F.date.astype(str)

dfPriceRVN.date = dfPriceRVN.date.astype(str)
```

In [14]:

```
# Merge
dfS2F = dfS2F.merge(dfPriceRVN, left_on='date', right_on='date', how='left')
dfS2F = dfS2F.merge(dfPriceBTC, left_on='date', right_on='date', how='left')
dfS2F.fillna(0)
```

Out[14]:

	date	height_RVN	flow_RVN	stock_RVN	open_RVN	high_RVN	low_RVN	close_RVN
0	2018-01-03	5945000	5945000	5945000	0.000000	0.000000	0.000000	0.000000
1	2018-01-04	10945000	10945000	16890000	0.000000	0.000000	0.000000	0.000000
2	2018-01-05	11500000	11500000	28390000	0.000000	0.000000	0.000000	0.000000
3	2018-01-06	9265000	9265000	37655000	0.000000	0.000000	0.000000	0.000000
4	2018-01-07	13170000	13170000	50825000	0.000000	0.000000	0.000000	0.000000
...	...	...	...	...	...	...	...	...
1240	2021-05-27	7330000	7330000	8845340000	0.101504	0.103803	0.091664	0.094475
1241	2021-05-28	7020000	7020000	8852360000	0.094473	0.096013	0.077685	0.081287
1242	2021-05-29	7130000	7130000	8859490000	0.081168	0.084461	0.070398	0.076148
1243	2021-05-30	7250000	7250000	8866740000	0.076206	0.083790	0.070914	0.078637
1244	2021-05-31	7060000	7060000	8873800000	0.078666	0.084311	0.074492	0.084311

1245 rows × 16 columns

In [15]:

```
# Export the data
dfS2F.to_csv('Datasets/rvnS2F.csv')
```