```
In [1]: import numpy as np
        import pandas as pd
        from scipy import ndimage
        from scipy.cluster import hierarchy
        from scipy.spatial import distance_matrix
        from matplotlib import pyplot as plt
        from sklearn import manifold, datasets
        from sklearn.cluster import AgglomerativeClustering
        from sklearn.datasets.samples_generator import make_blobs
        %matplotlib inline
```

Throughout this project, I plan on analyzing and comparing attributes pulled from the Ethereum Blockchain dataset. To analyze, I will examine gas usage, accounts, hash rates, difficulty, blocks, payout rewards, ETH, uncles, balances, and smart contracts. Each of these attributes sheds light on the health of the Ethereum Blockchain network and some even correlate with each other, predicting increases in difficulty, for example, when the hash rate increases. Not only will I be examining the data for indications of relationships between the attributes listed, but also see the changes over the past year, taking note of any differences. I will extract the week on October 1, 2019 - October 10, 2019 and compare the data to the week of October 1, 2018 - October 10, 2018. I expect to see relationships between the attributes above and predictions that can be made by these relations, I am unsure of what to expect when it comes to date comparison, but I am hoping for interesting data.

This information would be useful for investors and miners of Ethereum. By understanding the attitude of the ethereum blockchain, the miners can know how many blocks they are likely to come upon when they are mining. We can predict gas usage, new accounts, hast rates and difficulty, payout rewards, ETH, uncles, balances, and smart contracts. Knowing the activity of these attributes allows investors to make safe investments, meaning they have confidence that their money was spent well. Now, as I had done in the background section, I will go into more detail of the format and span of each attribute and how we will use these numbers.

Gas Data Examination

The first attribute we will look at is gas usage. Gas monitors and compensates miners for their work running transactions and smart contracts. The more gas used is a good sign, miners are more incentivised to process transactions with higher gas usage, and the price per transaction has risen.

Data from October 1st, 2019 through October 10th, 2019

```
In [2]: gas_2019 = pd.read_csv("Gas_Data.csv")
        gas_2018 = pd.read_csv("Gas_Data_2018.csv")
```

In [3]:
```
gas_2019.head()
```

Out[3]:

	gas_used	gas_limit
0	8588999	8595658
1	4881815	8902273
2	9145852	9157295
3	9073389	9091159
4	9029541	9049830

In [4]:
```
gas_2018.head()
```

Out[4]:

	gas_used	gas_limit
0	1872245	7996107
1	7862605	7992222
2	7931656	7996144
3	7964810	7996144
4	5353874	7984452

The gas measurements are numeric values, usually whole numbers which correspond to the computation power needed for the given transaction. The limit is the limit each block is given until it 'expires'. The range of the gas usage and the gas limits is explored below as well as the percentage of the maximum gas used to the total gas used [same with the gas limit]

In [5]:

```
min_gas_2019 = min(gas_2019['gas_used'])
max_gas_2019 = max(gas_2019['gas_used'])
print("Gas usage in the week of 2019 ranges from", min_gas_2019, "to", max_gas
_2019)
total_gas_2019 = sum(gas_2019['gas_used'])
percent_gas_2019 = round((max_gas_2019/total_gas_2019)*100, 4)
print("The total amount of gas used during the week in 2019 is", total_gas_201
9, "making the maximum amount of gas used in one transaction", percent_gas_201
9, "% of the total gas used")
min_gas_2018 = min(gas_2018['gas_used'])
max_gas_2018 = max(gas_2018['gas_used'])
print("Gas usage in the week of 2018 ranges from", min_gas_2018, "to", max_gas
_2018)
total_gas_2018 = sum(gas_2018['gas_used'])
percent_gas_2018 = round((max_gas_2018/total_gas_2018)*100, 4)
print("The total amount of gas used during the week of 2018 is", total_gas_201
8, "making the maximum amount of gas used in one transaction", percent_gas_201
8, "% of the total gas used")
dif_18_19 = total_gas_2019 - total_gas_2018
print("The difference in gas used between the years is", dif_18_19)
```

```
Gas usage in the week of 2019 ranges from 0 to 10004473
The total amount of gas used during the week in 2019 is 113097277438 making t
he maximum amount of gas used in one transaction 0.0088 % of the total gas us
ed
Gas usage in the week of 2018 ranges from 0 to 8012355
The total amount of gas used during the week of 2018 is 98631599470 making th
e maximum amount of gas used in one transaction 0.0081 % of the total gas use
d
The difference in gas used between the years is 14465677968
```

In [6]:

```
min_gas_lim_2019 = min(gas_2019['gas_limit'])
max_gas_lim_2019 = max(gas_2019['gas_limit'])
print("Gas limit in the week in 2019 ranges from", min_gas_lim_2019, "to", max
_gas_lim_2019)
total_gas_lim_2019 = sum(gas_2019['gas_limit'])
percent_gas_lim_2019 = round((max_gas_lim_2019/total_gas_lim_2019)*100, 4)
print("The total gas limit during the week in 2019 is", total_gas_lim_2019, "m
aking the maximum gas limit for one block", percent_gas_lim_2019, "% of the to
tal gas limit")
min_gas_lim_2018 = min(gas_2018['gas_limit'])
max_gas_lim_2018 = max(gas_2018['gas_limit'])
print("Gas limit in the week in 2018 ranges from", min_gas_lim_2018, "to", max
_gas_lim_2018)
total_gas_lim_2018 = sum(gas_2018['gas_limit'])
percent_gas_lim_2018 = round((max_gas_lim_2018/total_gas_lim_2018)*100, 4)
print("The total gas limit during the week in 2018 is", total_gas_lim_2018, "m
aking the maximum gas limit for one block", percent_gas_lim_2018, "% of the to
tal gas limit")
dif_18_19_lim = total_gas_lim_2019 - total_gas_lim_2018
print("The difference in gas limits between the years is", dif_18_19_lim)
```

```
Gas limit in the week in 2019 ranges from 8305112 to 10009649
The total gas limit during the week in 2019 is 151650306106 making the maximu
m gas limit for one block 0.0066 % of the total gas limit
Gas limit in the week in 2018 ranges from 7976682 to 8019530
The total gas limit during the week in 2018 is 128004170890 making the maximu
m gas limit for one block 0.0063 % of the total gas limit
The difference in gas limits between the years is 23646135216
```

Account Data Examination

Our second attribute is the accounts on the blockchain. By looking at distinct and new accounts, we can see the growth of the blockchain and see if there are any accounts that are submitting more transactions than others. We will look at how many new accounts were created during the week in questions as well as the percentage that the top 10 accounts hold of the total transactions.

In [7]:

```
user_2019 = pd.read_csv("User_Data.csv")
```

In [8]:

```
user_2019.head()
```

Out[8]:

	from_address
0	0x2e5025e4b46136e9cf57df5c6b40ffba4847938a
1	0xcefc94f1c0a0be7ad47c7fd961197738fc233459
2	0x63faf9adbf8bff970517586d5737577b9afb8f5f
3	0x63faf9adbf8bff970517586d5737577b9afb8f5f
4	0x63faf9adbf8bff970517586d5737577b9afb8f5f

The account data is relayed in terms of addresses. Each user has a unique hashed address that allows us to track a users transation habits. We will look at the total number of transactions during the week in 2019 and see what percentage of those are from unique users.

In [9]:
```
unique_user_2019 = user_2019['from_address'].nunique()
total_user_2019 = len(user_2019)
percent_2019 = round((unique_user_2019/total_user_2019)*100, 4)
print("There were", total_user_2019, "transactions during the week in 2019", u
nique_user_2019, "of which were done from unique users, meaning", percent_2019
, "% of the transactions were from unique users")
```

```
There were 16000 transactions during the week in 2019 6650 of which were done
from unique users, meaning 41.5625 % of the transactions were from unique use
rs
```

Transaction Data Examination

In [10]:
```
transaction_2019 = pd.read_csv("Transaction_Data.csv")
```

In [11]:
```
transaction_2019.head()
```

Out[11]:

	transaction_index	receipt_status
0	91	1
1	100	1
2	120	1
3	118	1
4	134	1

The transaction data shows an index for each unique transaction and the status of that transaction. 1 represents a successful transaction while 2 represents a failed transaction. We will look at the total number of transactions and the number of unique transactions. Next, we will look at the number of failed transactions to get an idea of how many occur in a week.

In [12]:
```
total_transactions = len(transaction_2019)
unique_transaction_2019 = transaction_2019['transaction_index'].nunique()
percent_transaction_2019 = round((unique_transaction_2019/total_transactions)*
100, 4)
print("There are", total_transactions, "during the week in 2019 and", unique_t
ransaction_2019, "unique transactions")
```

```
There are 16000 during the week in 2019 and 367 unique transactions
```

```
In [13]: success_2019 = sum(transaction_2019['receipt_status'])
         failures_2019 = total_transactions - success_2019
         print("There were", failures_2019, "in the week during 2019.")
```

```
There were 99 in the week during 2019.
```

Difficulty Data Examination

The third attribute is the comparison between difficulty and hash rates. If there is a trend of decreasing hash rates and difficulty, it is more likely that miners will leave the market. So, if we see an increase in hash rates and difficulty, it most likely means that we will see an increase in accounts created during that week.

```
In [16]: diff_2019 = pd.read_csv("Difficulty_Data.csv")
         diff_2018 = pd.read_csv("Difficulty_Data_18.csv")
```

```
In [15]: diff_2019.head()
```

Out[15]:

	difficulty	total_difficulty
0	2450601433835267	12172532987467676832910
1	2440267061687949	12179794614002681712781
2	2456848949805071	12167955292109241573853
3	2457119537197136	12178751439057882152066
4	2478371631933243	12169185556456669797505

```
In [17]: diff_2018.head()
```

Out[17]:

	difficulty	total_difficulty
0	3381667710809097	7109776113537686399360
1	3292607650454272	7107190453748468983508
2	3250211802852484	7121752547893712254821
3	3255416338675367	7122139377011904239340
4	3204046394868426	7124222625837486276721

In [19]:

```
max_diff_19 = max(diff_2019['difficulty'])
max_diff_18 = max(diff_2018['difficulty'])
diff_max = max_diff_19 - max_diff_18
print("The max difficulty in 2019 was", max_diff_19, "while the max difficulty
in 2018 was", max_diff_18, "making the difference between these two measuremen
ts", diff_max)
```

```
The max difficulty in 2019 was 2514690137375735 while the max difficulty in 2
018 was 3405585906071665 making the difference between these two measurements
-890895768695930
```

This means that there was a transaction with a higher difficulty in 2018, though you can see the total difficulty has increased a lot as it is the difficulty of the overall block. A decrease in difficulty is a good sign, meaning the market is growing.

Block Data Examination

The fourth attribute we will examine is the blocks themselves. We want to know how many blocks there are that are being used in the week and if that number has increased or decreased over the year. We will also look at the most popular and least popular blocks during each week.

In [20]:

```
block_19 = pd.read_csv("Block_Data.csv")
block_18 = pd.read_csv("Block_Data_18.csv")
```

In [21]:

```
block_19.head()
```

Out[21]:

	number	total_difficulty	size	gas_used	transaction_count
0	8660603	12183511069872976529085	35421	9915469	166
1	8660801	12183988786670363324836	19460	9897019	95
2	8662510	12188056975244278429038	38684	9954402	96
3	8661027	12184531568844007075364	24910	9942923	135
4	8660588	12183474367721881228244	34603	9939968	172

In [22]:

```
block_18.head()
```

Out[22]:

	number	total_difficulty	size	gas_used	transaction_count
0	6444251	6991901960045717442428	17143	7922713	127
1	6447134	7001193818649018788849	23271	7984186	123
2	6444469	6992595956616344401430	28896	7983163	187
3	6443928	6990862801255004344193	32972	7981101	187
4	6445513	6995942741677095992250	27269	7982530	120

In [28]:

```
num_block_19 = max(block_19['number'])
num_block_18 = max(block_18['number'])
perc_inc = round((num_block_18/num_block_19)*100, 0)
print("The number of blocks in 2019 is", num_block_19, "while in 2018", num_bl
ock_18, ". A", perc_inc, "% Inc")
```

```
The number of blocks in 2019 is 8710739 while in 2018 6510315 . A 75.0 % Inc
```

Contracts Data Examination

The final attribute is smart contracts. These contracts is what sets Ethereum Blockchain apart from any other cryptocurrency.

In [30]:

```
contract_19 = pd.read_csv("Contracts_Data.csv")
contract_18 = pd.read_csv("Contracts_Data_18.csv")
```

In [31]:

```
contract_19.head()
```

Out[31]:

	address	is_erc20	is_erc721
0	0xfed153f22f1edd5b4003517403e65293323a137c	False	False
1	0xc54c3b6432eb662e7699a9866600e391f5598d52	False	False
2	0x2346c79c81185c83850a24ec96303062742a2855	False	False
3	0x4cb7538aa7dd0d30f91a1295c2084f2d04c0cc04	False	False
4	0x09c014bebc370c282f401f0a4e453a9ecc8a4c33	False	False

In [32]:

```
contract_18.head()
```

Out[32]:

	address	is_erc20	is_erc721
0	0xeea18d90ad58f50fdab41a61b0d5dc1dbd891af2	False	False
1	0x56510b7560babdc13c0bc175d5dc0ec09233b1de	False	False
2	0x88e4564360f2935286362e7b438b825c8e9da09d	False	False
3	0xeeb73f6e88f501ad7906e3a500197903634b4716	False	False
4	0x819f96004020e85e4642370e791b459675639af8	False	False

In [ ]:

In [ ]: