While Coin Metrics **Network Data Pro** offers users the ability to analyze aggregated on-chain metrics for a variety of cryptoassets, some power users may require additional customization and granularity. Coin Metrics **ATLAS search engine** equips users with the on-chain analytical toolset they need to workshop their own custom metrics.

Resources

This notebook demonstrates basic functionality offered by the Coin Metrics Python API Client, ATLAS, and Network Data Pro.

Coin Metrics offers a vast assortment of data for hundreds of cryptoassets. The Python API Client allows for easy access to this data using Python without needing to create your own wrappers using `requests` and other such libraries.

To understand the data that Coin Metrics offers, feel free to peruse the resources below.

- The Coin Metrics API v4 (https://docs.coinmetrics.io/api/v4) website contains the full set of endpoints and data offered by Coin Metrics.
- The Coin Metrics Knowledge Base (https://docs.coinmetrics.io/info) gives detailed, conceptual explanations of the data that Coin Metrics offers.
- The API Spec (https://coinmetrics.github.io/api-client-python/site/api_client.html) contains a full list of functions.

Notebook Setup

In [1]:

```
from os import environ
import pandas as pd
import logging
from datetime import date, datetime, timedelta
from coinmetrics.api_client import CoinMetricsClient
import json
import logging
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from IPython.display import Markdown as md
```

In [2]:

```
sns.set_theme()
sns.set(rc={'figure.figsize':(12,8)})
sns.set_style("whitegrid",{'axes.grid' : False,'grid.linestyle': '--', 'grid.color': 'black','axes.edgecolor': 'white','font.fam
ily': ['Lato']})
```

In [3]:

```
logging.basicConfig(
    format='%(asctime)s %(levelname)-8s %(message)s',
    level=logging.INFO,
    datefmt='%Y-%m-%d %H:%M:%S'
)
```

In [4]:

```
# We recommend privately storing your API key in your local environment.
try:
    api_key = environ["CM_API_KEY"]
    logging.info("Using API key found in environment")
except KeyError:
    api_key = ""
    logging.info("API key not found. Using community client")

client = CoinMetricsClient(api_key)
```

```
2022-11-09 17:57:25 INFO     Using API key found in environment
```

Block-by-Block Metrics

With **Network Data Pro**, block-by-block metrics are offered for both BTC and ETH. With **ATLAS**, however, we have the ability to retrieve block-by-block metrics for any asset in ATLAS coverage.

In this example, we quantify the total amount of USDC and USDT transferred on a block-by-block basis.

In [5]:

```
# Selecting the past X hours of transactions
X = 3
end = datetime.now()
start = end - timedelta(hours=X)
```

In [6]:

```
# Retrieving data for both USDC and USDT (ETH chain)
asset_list = ['usdc','usdt_eth']
```

In [7]:

```
for asset in asset_list:
    print('Fetching ' + asset.upper() + ' transactions...')

    # ATLAS 'Get List of Transactions' function
    vars()[asset + '_tx'] = client.get_list_of_transactions(
            asset=asset,
            start_time=start,
            end_time=end
    ).to_dataframe().drop_duplicates(subset=['transaction_hash'],keep='last')
```

```
Fetching USDC transactions...
Fetching USDT_ETH transactions...
```

USDC Transactions

In [8]:

```
usdc_tx.tail()
```

Out[8]:

	block_hash	height	transaction_hash	consensus_time	min_chain_sequenc
18716	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	ae4d0fc07c89e577792d4e6ae1114cad2a97f30f0222ea...	2022-11-09 17:57:23+00:00	1
18717	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	b5ee494b6930e975396e56786cf84db69b795558145b6a...	2022-11-09 17:57:23+00:00	1
18718	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	5a47a8ed382fb1d8715a2a1d16e29ea8b4b1b83b217be6...	2022-11-09 17:57:23+00:00	1
18719	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	2a12aec30483681c25c71a1ec322c66abce4745cbac94e...	2022-11-09 17:57:23+00:00	1
18720	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	446af0abd38fa55c986b1c1c3bdb77b744634235c31420...	2022-11-09 17:57:23+00:00	1

In [9]:

```python
# Transform balance updates into block-by-block transfer metrics with one line of code
usdc_tx_bbb = pd.DataFrame(usdc_tx.groupby('height')['amount'].sum())
usdc_tx_bbb
```

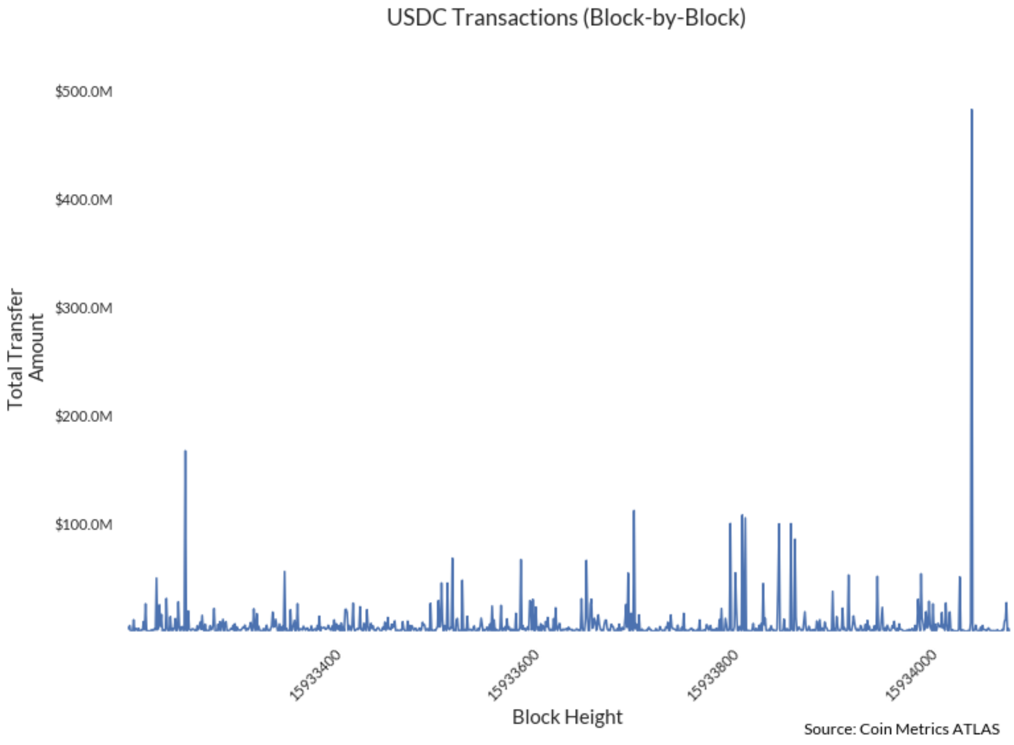
Out[9]:

	amount
height	
15933208	507091.67629
15933209	2977364.85398
15933210	3398410.67384
15933211	2377371.735658
15933212	5444830.727578
...	...
15934095	26574543.81572
15934096	163425.841317
15934097	2857591.812299
15934098	222612.771777
15934099	366710.426768

889 rows × 1 columns

In [10]:

```python
ax = sns.lineplot(data=usdc_tx_bbb,y=usdc_tx_bbb['amount'],x=usdc_tx_bbb.index)
ax.set_xlabel("Block Height", fontsize = 15)
ax.set_ylabel("Total Transfer \nAmount", fontsize = 15)
plt.setp(ax.get_xticklabels(), rotation=45)
ax.xaxis.set_ticks(plt.gca().get_xticks())
plt.gca().set_xticklabels(['{:.0f}'.format(x) for x in plt.gca().get_xticks()])
ax.yaxis.set_ticks(plt.gca().get_yticks())
plt.gca().set_yticklabels(['${:,.1f}M'.format(y/1000000) for y in plt.gca().get_yticks()])
plt.ylim([usdc_tx_bbb['amount'].min(), usdc_tx_bbb['amount'].max()*1.1])
plt.xlim([usdc_tx_bbb.index[0], usdc_tx_bbb.index[-1]])
plt.annotate('Source: Coin Metrics ATLAS',xy=(1, -0.195), xycoords='axes fraction',color='black',xytext=(-8, 6), textcoords='offset pixels',horizontalalignment='right',verticalalignment='bottom')
ax.set_title('\nUSDC Transactions (Block-by-Block) \n', fontsize = 17);
```


USDC Transactions (Block-by-Block)

Source: Coin Metrics ATLAS

In [11]:

```
largest_tx = usdc_tx.loc[usdc_tx['amount'].idxmax()]
```

In [12]:

```
largest_tx
```

Out[12]:

```
block_hash                   9e583063d52db0cb9cef46242df9aeffa7032e1ad256af...
height                                                                15934060
transaction_hash             1a0d804233cf62eb0afdbaef568742948afd150bf9a8cd...
consensus_time                                       2022-11-09 17:49:23+00:00
min_chain_sequence_number                                            105405514
max_chain_sequence_number                                            105405521
n_balance_updates                                                            8
amount                                                         480448349.16217
Name: 17981, dtype: object
```

In [13]:

```
md('<br><font size="3.5">Transaction info can also be viewed in the **ATLAS** graphical user interface:<br><br>**Largest Transac
tion:** <br>https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=' + str(largest_tx.transaction_hash) )
```

Out[13]:

Transaction info can also be viewed in the **ATLAS** graphical user interface:

Largest Transaction:
https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=1a0d804233cf62eb0afdbaef568742948afd150bf9a8cd71bd947057cc77b827 (https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=1a0d804233cf62eb0afdbaef568742948afd150bf9a8cd71bd947057cc77b827)

USDT Transactions

In [14]:

```
usdt_eth_tx.tail()
```

Out[14]:

	block_hash	height	transaction_hash	consensus_time	min_chain_seque
24109	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	b5ee494b6930e975396e56786cf84db69b795558145b6a...	2022-11-09 17:57:23+00:00	
24110	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	c904577fc84387acaeab539a22ecf66f4dbc4173ad434a...	2022-11-09 17:57:23+00:00	
24111	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	d886e0717003663e1936ce586eccaa2d8dfc958921584f...	2022-11-09 17:57:23+00:00	
24112	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	bb2ff57ca27ebc44430571ef31918a4aa3ff02d2e6d565...	2022-11-09 17:57:23+00:00	
24113	dbdcf86a908afda724a6b45e2830b87e694489c14b944e...	15934099	ea83d72f9a42b4750f50092b5c9a533cd1e19713794b4b...	2022-11-09 17:57:23+00:00	

In [15]:

```
# Transform balance updates into block-by-block transfer metrics with one line of code
usdt_tx_bbb = pd.DataFrame(usdt_eth_tx.groupby('height')['amount'].sum())
usdt_tx_bbb
```

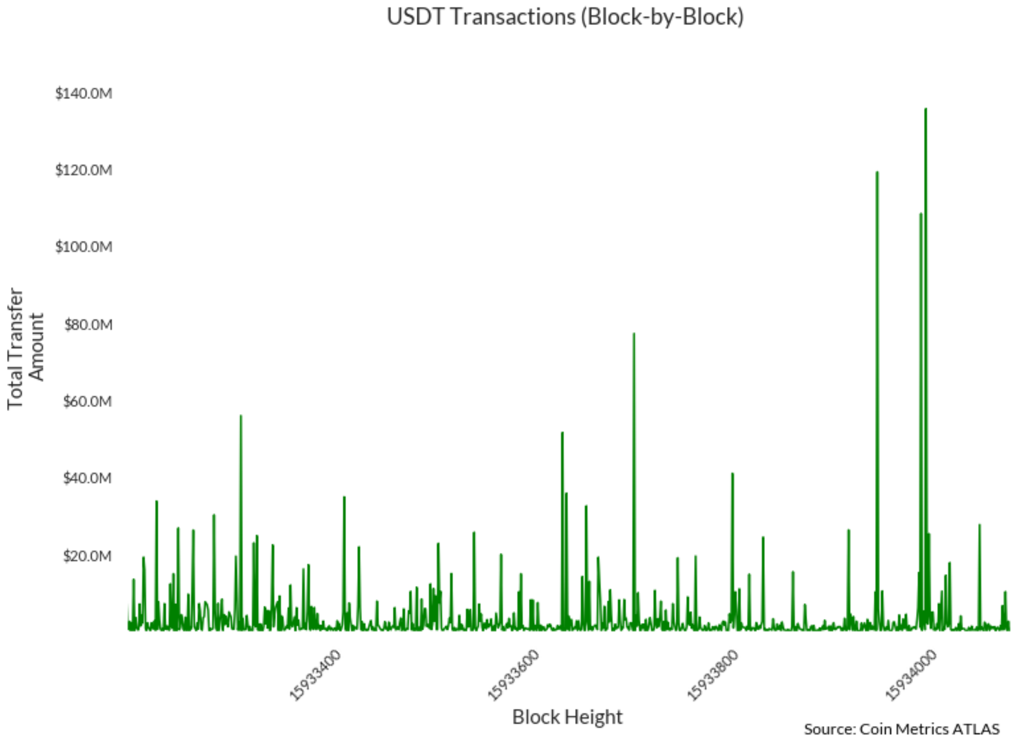
Out[15]:

	amount
height	
15933208	695343.825351
15933209	7222687.638798
15933210	884299.636952
15933211	529013.72989
15933212	2571280.855578
...	...
15934095	475017.511305
15934096	225307.633622
15934097	2600788.558732
15934098	171146.29567
15934099	507970.184634

888 rows × 1 columns

In [16]:

```
ax = sns.lineplot(data=usdt_tx_bbb,y=usdt_tx_bbb['amount'],x=usdt_tx_bbb.index,color='green')
ax.set_xlabel("Block Height", fontsize = 15)
ax.set_ylabel("Total Transfer \nAmount", fontsize = 15)
plt.setp(ax.get_xticklabels(), rotation=45)
ax.xaxis.set_ticks(plt.gca().get_xticks())
plt.gca().set_xticklabels(['{:.0f}'.format(x) for x in plt.gca().get_xticks()])
ax.yaxis.set_ticks(plt.gca().get_yticks())
plt.gca().set_yticklabels(['${:,.1f}M'.format(y/1000000) for y in plt.gca().get_yticks()])
plt.ylim([usdt_tx_bbb['amount'].min(), usdt_tx_bbb['amount'].max()*1.1])
plt.xlim([usdt_tx_bbb.index[0], usdt_tx_bbb.index[-1]])
plt.annotate('Source: Coin Metrics ATLAS',xy=(1, -0.195), xycoords='axes fraction',color='black',xytext=(-8, 6), textcoords='offset pixel
s',horizontalalignment='right',verticalalignment='bottom')
ax.set_title('\nUSDT Transactions (Block-by-Block) \n', fontsize = 17);
```



Source: Coin Metrics ATLAS

In [17]:

```
largest_usdt_tx = usdt_eth_tx.loc[usdt_eth_tx['amount'].idxmax()]
largest_usdt_tx
```

Out[17]:

```
block_hash                    1f2721a9399ff06ff260eee8c962df8a47c0fd71d58eb5...
height                                                                 15934014
transaction_hash              98e373fce1869a94d71f33c33ed34e44b77bf80dd3540a...
consensus_time                                        2022-11-09 17:39:59+00:00
min_chain_sequence_number                                             325971102
max_chain_sequence_number                                             325971111
n_balance_updates                                                            10
amount                                                      112533805.116516
Name: 21866, dtype: object
```

In [18]:

```
md('<br><font size="3.5">Transaction info can also be viewed in the **ATLAS** graphical user interface:<br><br>**Largest Transaction:** <b
r>https://atlas.coinmetrics.io/transaction-details?asset=usdt_eth&tx_hash=' + str(largest_usdt_tx.transaction_hash) )
```

Out[18]:

Transaction info can also be viewed in the **ATLAS** graphical user interface:

Largest Transaction:
https://atlas.coinmetrics.io/transaction-details?asset=usdt_eth&tx_hash=98e373fce1869a94d71f33c33ed34e44b77bf80dd3540a67bbd496cbcd9af238 (https://atlas.coinmetrics.io/transaction-details?asset=usdt_eth&tx_hash=98e373fce1869a94d71f33c33ed34e44b77bf80dd3540a67bbd496cbcd9af238)

Cross-Asset Metrics

With **Network Data Pro**, users have the ability to retrieve aggregated daily transfers for specific assets such as USDC and USDT. With **ATLAS**, however, we have the ability to create more customized, granular metrics.

In this example, we quantify the total size of **transactions where USDT and USDC are transferred simultaneously**, on a block-by-block basis.

In [19]:

```
usdc_tx['USDC Amount'] = usdc_tx['amount']
usdt_eth_tx['USDT Amount'] = usdt_eth_tx['amount']

merged = pd.merge(usdc_tx[['height','USDC Amount','transaction_hash']], usdt_eth_tx[['USDT Amount','transaction_hash']], on ='transaction_
hash')
merged['Total Amount'] = merged['USDT Amount'] + merged['USDC Amount']
```

In [20]:

```
merged.tail()
```

Out[20]:

	height	USDC Amount	transaction_hash	USDT Amount	Total Amount
1348	15934094	14151.683758	a6e06896d2d7e966d11ac5760a59346697cc19a1f0729d...	14174.0	28325.683758
1349	15934094	10058128.952608	dd006c72c79b908dbaf060099feaee5e9c663a7c13636b...	10070851.280146	20128980.232754
1350	15934095	49921.223617	e82974d7ccec5d6df9d7c96dd35ac93b65bc97b485e4fe...	50000.0	99921.223617
1351	15934098	5745.546814	c80b2ff45df42c6629c07e34d807d51d9f8728f43b7ff0...	2876.733815	8622.280629
1352	15934099	43930.621711	b5ee494b6930e975396e56786cf84db69b795558145b6a...	300000.0	343930.621711

In [21]:

```
last_tx = merged.iloc[-1]
txhash = last_tx.transaction_hash
```

In [22]:

```
md('<br><font size="3.5">Transaction info can also be viewed in the **ATLAS** graphical user interface:<br><br>**USDC:** <br>https://atla
s.coinmetrics.io/transaction-details?asset=usdc&tx_hash=' + str(txhash) + '<br>**USDT:** <br>https://atlas.coinmetrics.io/transaction-deta
ils?asset=usdt_eth&tx_hash=' + str(txhash))
```

Out[22]:

Transaction info can also be viewed in the **ATLAS** graphical user interface:

USDC:
https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=b5ee494b6930e975396e56786cf84db69b795558145b6ab1fae422e696fb068e (https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=b5ee494b6930e975396e56786cf84db69b795558145b6ab1fae422e696fb068e)
USDT:
https://atlas.coinmetrics.io/transaction-details?asset=usdt_eth&tx_hash=b5ee494b6930e975396e56786cf84db69b795558145b6ab1fae422e696fb068e (https://atlas.coinmetrics.io/transaction-details?asset=usdt_eth&tx_hash=b5ee494b6930e975396e56786cf84db69b795558145b6ab1fae422e696fb068e)

In [23]:

```
# Determine total amount of USDT + USDC transferred on a block-by-block basis
both_tx_bbb = pd.DataFrame(merged.groupby('height')['Total Amount'].sum())
both_tx_bbb
```

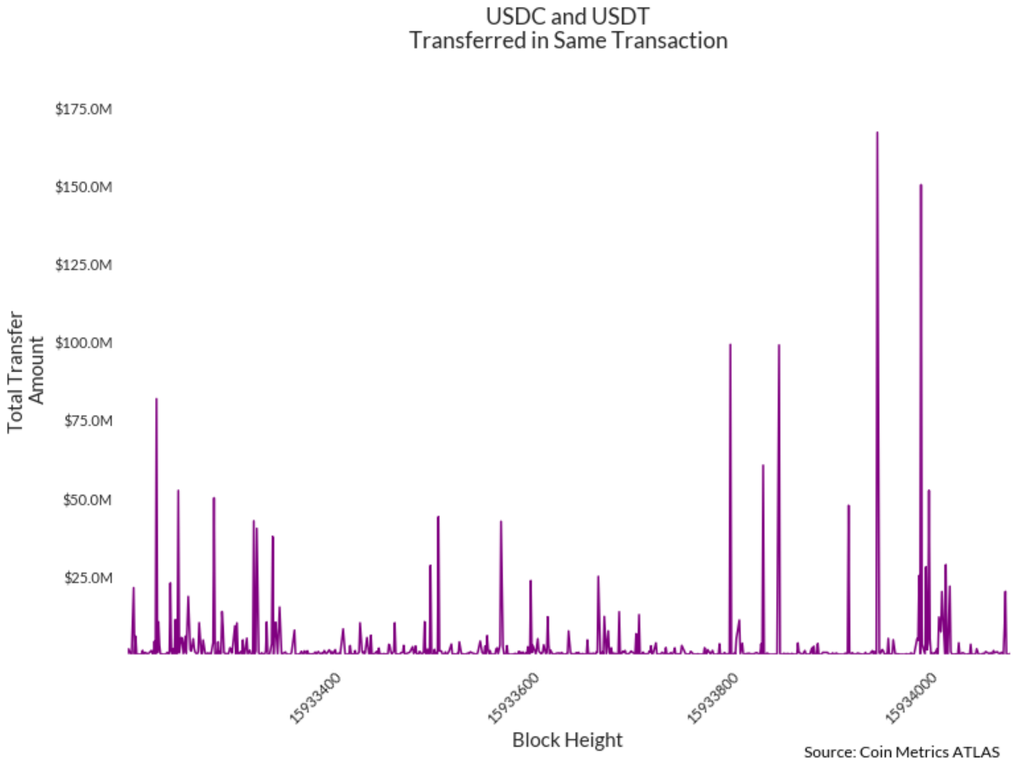
Out[23]:

	Total Amount
height	
15933208	9682.837705
15933209	270988.723569
15933210	285645.514178
15933211	1764724.188115
15933213	9227.973294
...	...
15934093	8552972.364746
15934094	20157305.916512
15934095	99921.223617
15934098	8622.280629
15934099	343930.621711

675 rows × 1 columns

In [24]:

```
ax = sns.lineplot(data=both_tx_bbb,y=both_tx_bbb['Total Amount'],x=both_tx_bbb.index,color='purple')
ax.set_xlabel("Block Height", fontsize = 15)
ax.set_ylabel("Total Transfer \nAmount", fontsize = 15)
plt.setp(ax.get_xticklabels(), rotation=45)
ax.xaxis.set_ticks(plt.gca().get_xticks())
plt.gca().set_xticklabels(['{:.0f}'.format(x) for x in plt.gca().get_xticks()])
ax.yaxis.set_ticks(plt.gca().get_yticks())
plt.gca().set_yticklabels(['${:,.1f}M'.format(y/1000000) for y in plt.gca().get_yticks()])
plt.ylim([both_tx_bbb['Total Amount'].min(), both_tx_bbb['Total Amount'].max()*1.1])
plt.xlim([both_tx_bbb.index[0], both_tx_bbb.index[-1]])
plt.annotate('Source: Coin Metrics ATLAS',xy=(1, -0.195), xycoords='axes fraction',color='black',xytext=(-8, 6), textcoords='offset pixel
s',horizontalalignment='right',verticalalignment='bottom')
ax.set_title('\nUSDC and USDT\nTransferred in Same Transaction\n', fontsize = 17);
```



Entity-Based Metrics

With **ATLAS,** we can derived our own metrics based on externally-sourced tagged addresses.

In this example, we leverage our catalog of Uniswap liquidity pool contract addresses to **estimate USDC inflows and DEX Supply** for major trading pairs.

Retrieving DEX Markets

In [25]:

```
defi_mkts = client.catalog_full_markets(
    exchange="uniswap_v3_eth",
).to_dataframe()
```

In [26]:

```
# Enter an asset ticker to see available 'base' and 'quote' markets
asset = 'usdc'
defi_mkts = defi_mkts.loc[(defi_mkts['base']==asset) | (defi_mkts['quote']==asset)]
```

In [27]:

```
defi_mkts.tail()
```

Out[27]:

	market	min_time	max_time	exchange	type	base	quote	pool_config_id	contract_address	fee	
552	uniswap_v3_eth-agg-usdc-weth-spot	2021-05-04 00:00:00+00:00	2022-11-09 22:29:00+00:00	uniswap_v3_eth	spot	usdc	weth	-1	<NA>	<NA>	a0b86991c6218b36c1d19d4a2
553	uniswap_v3_eth-agg-usdc-wluna-spot	2022-05-16 00:00:00+00:00	2022-11-09 22:29:00+00:00	uniswap_v3_eth	spot	usdc	wluna	-1	<NA>	<NA>	a0b86991c6218b36c1d19d4a2
554	uniswap_v3_eth-agg-usdc-zrx-spot	2022-01-22 00:00:00+00:00	2022-07-21 23:58:00+00:00	uniswap_v3_eth	spot	usdc	zrx	-1	<NA>	<NA>	a0b86991c6218b36c1d19d4a2
569	uniswap_v3_eth-agg-wbtc-usdc-spot	2021-05-05 00:00:00+00:00	2022-11-09 22:29:00+00:00	uniswap_v3_eth	spot	wbtc	usdc	-1	<NA>	<NA>	2260fac5e5542a773aa44fbc
592	uniswap_v3_eth-agg-yfi-usdc-spot	2022-07-29 00:00:00+00:00	2022-11-09 22:29:00+00:00	uniswap_v3_eth	spot	yfi	usdc	-1	<NA>	<NA>	0bc529c00c6401aef6d220be

In [28]:

```
defi_list = defi_mkts.contract_address.dropna().tolist()
defi_list[0:5]
```

Out[28]:

```
['5e35c4eba72470ee1177dcb14dddf4d9e6d915f4',
 '5777d92f208679db4b9778590fa3cab3ac9e2168',
 'df50fbde8180c8785842c8e316ebe06f542d3443',
 '486263aa56d1b49d78dea765754164b880c99954',
 '0f5353bf7acc77b38bcbb437a19a9e509787515b']
```

In [29]:

```
len(defi_list)
```

Out[29]:

```
90
```

In [30]:

```
day_end = datetime.now()
day_start = day_end - timedelta(days=1)
```

In [31]:

```
defi_balance = client.get_list_of_balance_updates(
    asset=asset,
    accounts=defi_list,
    start_time=day_start,
    end_time=day_end
).to_dataframe()
```

In [32]:

```
defi_balance
```

Out[32]:

	block_hash	height	consensus_time	chain_sequence_number	account
0	55521cdcdd8c335fef32b303085bb6aef8071cbb520496...	15926966	2022-11-08 17:59:47+00:00	105032428	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640
1	55521cdcdd8c335fef32b303085bb6aef8071cbb520496...	15926966	2022-11-08 17:59:47+00:00	105032430	8ad599c3a0ff1de082011efddc58f1908eb6e6d8
2	55521cdcdd8c335fef32b303085bb6aef8071cbb520496...	15926966	2022-11-08 17:59:47+00:00	105032432	7bea39867e4169dbe237d55c8242a8f2fcdcc387
3	55521cdcdd8c335fef32b303085bb6aef8071cbb520496...	15926966	2022-11-08 17:59:47+00:00	105032434	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640
4	55521cdcdd8c335fef32b303085bb6aef8071cbb520496...	15926966	2022-11-08 17:59:47+00:00	105032437	5777d92f208679db4b9778590fa3cab3ac9e2168
...	...	...	...	...	...
40537	497f9962579856b27049447238a1ee0c2888d73df206b2...	15934110	2022-11-09 17:59:35+00:00	105407913	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640
40538	497f9962579856b27049447238a1ee0c2888d73df206b2...	15934110	2022-11-09 17:59:35+00:00	105407914	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640
40539	497f9962579856b27049447238a1ee0c2888d73df206b2...	15934110	2022-11-09 17:59:35+00:00	105407916	99ac8ca7087fa4a2a1fb6357269965a2014abc35
40540	497f9962579856b27049447238a1ee0c2888d73df206b2...	15934110	2022-11-09 17:59:35+00:00	105407918	9a772018fbd77fcd2d25657e5c547baff3fd7d16
40541	497f9962579856b27049447238a1ee0c2888d73df206b2...	15934110	2022-11-09 17:59:35+00:00	105407928	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640

40542 rows × 16 columns

Retrieve DEX Inflows

In [33]:

```
defi_inflow = defi_balance[defi_balance.change > 0]
```

In [34]:

```
largest_inflow = defi_balance.loc[defi_balance['change'].idxmax()]
largest_inflow
```

Out[34]:

```
block_hash                    9e583063d52db0cb9cef46242df9aeffa7032e1ad256af...
height                                                                 15934060
consensus_time                                        2022-11-09 17:49:23+00:00
chain_sequence_number                                                 105405517
account                                5777d92f208679db4b9778590fa3cab3ac9e2168
account_creation_height                                                13605124
change                                                         240223927.341704
previous_balance                                                 2020251.911267
new_balance                                                    242244179.252971
transaction_sequence_number                                                   1
previous_n_debits                                                         75434
previous_n_credits                                                        65566
transaction_hash              1a0d804233cf62eb0afdbaef568742948afd150bf9a8cd...
previous_debit_height                                                  15934060
previous_credit_height                                                 15934047
previous_chain_sequence_number                                        105405514
Name: 40302, dtype: object
```

In [35]:

```
largest_inflow.transaction_hash
md('<br><font size="3.5">Transaction info can also be viewed in the **ATLAS** graphical user interface:<br><br>**Largest Transaction:** <br>https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=' + str(largest_inflow.transaction_hash) )
```

Out[35]:

Transaction info can also be viewed in the **ATLAS** graphical user interface:

Largest Transaction:
https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=1a0d804233cf62eb0afdbaef568742948afd150bf9a8cd71bd947057cc77b827 (https://atlas.coinmetrics.io/transaction-details?asset=usdc&tx_hash=1a0d804233cf62eb0afdbaef568742948afd150bf9a8cd71bd947057cc77b827)

In [36]:

```
defi_inflow = pd.DataFrame(defi_inflow[['consensus_time','account','change','height']])
defi_inflow_sum = pd.DataFrame(defi_inflow.groupby('height')['change'].sum())
```

In [37]:

```
defi_inflow_sum
```

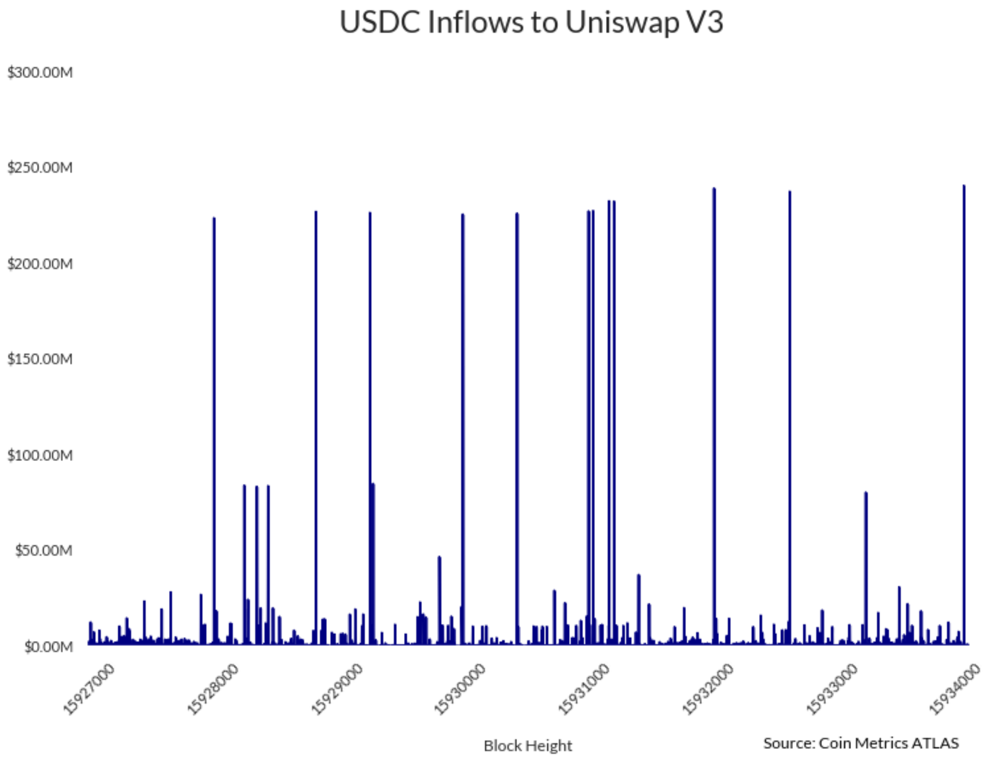
Out[37]:

	change
height	
15926966	128511.592256
15926967	14424.095205
15926968	447641.231732
15926969	339046.867277
15926971	1815628.758073
...	...
15934104	200.0
15934106	8606.714564
15934108	332.796979
15934109	1460.633591
15934110	1843.843307

6177 rows × 1 columns

In [38]:

```
ax = defi_inflow_sum['change'].plot.area(color='navy')
plt.title('\n ' + asset.upper() + ' Inflows to Uniswap V3\n',fontdict={'fontsize':23})
ax.set_xlabel("\nBlock Height")
ax.yaxis.set_ticks(plt.gca().get_yticks())
plt.setp(ax.get_xticklabels(), rotation=45)
ax.xaxis.set_ticks(plt.gca().get_xticks())
plt.gca().set_xticklabels(['{:.0f}'.format(x) for x in plt.gca().get_xticks()])
plt.xlim([defi_inflow_sum.index[0], defi_inflow_sum.index[-1]])
plt.annotate('Source: Coin Metrics ATLAS',xy=(1, -0.195), xycoords='axes fraction',color='black',xytext=(-8, 6), textcoords='offset pixels',horizontalalignment='right',verticalalignment='bottom')
plt.gca().set_yticklabels(['${:,.2f}M'.format(x/1000000) for x in plt.gca().get_yticks()]);
```



Retrieve DEX Supply

In [39]:

```
defi_new_bal = pd.DataFrame(defi_balance[['consensus_time','account','new_balance','height']])
defi_new_bal = defi_new_bal.sort_values(by='height')
defi_new_bal = pd.DataFrame(defi_new_bal.drop_duplicates(subset=['account','height'],keep='last'))
```

In [40]:

```
defi_new_bal
```

Out[40]:

	consensus_time	account	new_balance	height
9	2022-11-08 17:59:47+00:00	3416cf6c708da44db2624d63ea0aaef7113527c6	81518726.576821	15926966
6	2022-11-08 17:59:47+00:00	d0fc8ba7e267f2bc56044a7715a489d851dc6d78	144131.10651	15926966
4	2022-11-08 17:59:47+00:00	5777d92f208679db4b9778590fa3cab3ac9e2168	227903882.623814	15926966
3	2022-11-08 17:59:47+00:00	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640	58166015.345805	15926966
2	2022-11-08 17:59:47+00:00	7bea39867e4169dbe237d55c8242a8f2fcdcc387	7932321.388701	15926966
...	...	...	...	...
40532	2022-11-09 17:59:23+00:00	8ad599c3a0ff1de082011efddc58f1908eb6e6d8	39111776.898415	15934109
40533	2022-11-09 17:59:23+00:00	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640	33585779.401115	15934109
40540	2022-11-09 17:59:35+00:00	9a772018fbd77fcd2d25657e5c547baff3fd7d16	226276.859299	15934110
40539	2022-11-09 17:59:35+00:00	99ac8ca7087fa4a2a1fb6357269965a2014abc35	5643957.619623	15934110
40541	2022-11-09 17:59:35+00:00	88e6a0c2ddd26feeb64f039a2c41296fcb3f5640	33202226.936513	15934110

20197 rows × 4 columns

In [41]:

```
bal_updates_pivot = defi_new_bal.pivot(index="height",columns="account",values="new_balance")
bal_updates_pivot = bal_updates_pivot.ffill()
bal_updates_pivot = bal_updates_pivot.bfill()
```

In [42]:

```
bal_updates_w_zeros = bal_updates_pivot.fillna(0)
```

In [43]:

```
column_list = list(bal_updates_w_zeros)
bal_updates_w_zeros["sum"] =bal_updates_w_zeros[column_list].sum(axis=1)
```

In [44]:

```
bal_updates_w_zeros
```

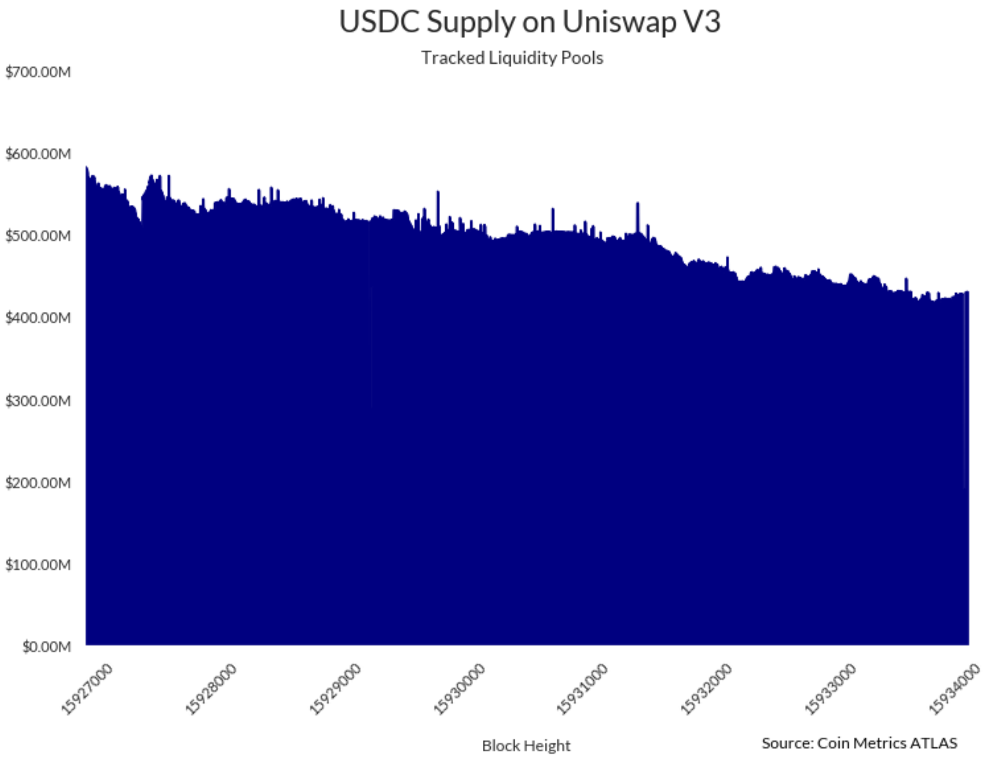
Out[44]:

account	00cef0386ed94d738c8f8a74e8bfd0376926d24c	07a6e955ba4345bae83ac2a6faa771fddd8a2011	3416cf6c708da44db2624d63ea0aaef7113527c6	3d61c8c42c
height				
15926966	3204279.066504	837546.094086	81518726.576821	
15926967	3204279.066504	837546.094086	81511399.850216	
15926968	3204279.066504	837546.094086	81511399.850216	
15926969	3204279.066504	837546.094086	81511399.850216	
15926970	3204279.066504	837546.094086	81509399.560422	
...	...	...	...	
15934106	2795493.163861	154552.063181	13936884.143085	
15934107	2795493.163861	154552.063181	13935011.810859	
15934108	2795493.163861	154552.063181	13935011.810859	
15934109	2795493.163861	154552.063181	13935011.810859	
15934110	2795493.163861	154552.063181	13935011.810859	

6837 rows × 37 columns

In [45]:

```
ax = bal_updates_w_zeros['sum'].plot.area(color='navy')
plt.title('\n ' + asset.upper() + ' Supply on Uniswap V3\n',fontdict={'fontsize':23})
plt.suptitle('\n\n\nTracked Liquidity Pools\n',size=13.5)
ax.set_xlabel("\nBlock Height")
ax.yaxis.set_ticks(plt.gca().get_yticks())
plt.setp(ax.get_xticklabels(), rotation=45)
ax.xaxis.set_ticks(plt.gca().get_xticks())
plt.gca().set_xticklabels(['{:.0f}'.format(x) for x in plt.gca().get_xticks()])
plt.xlim([bal_updates_w_zeros.index[0], bal_updates_w_zeros.index[-1]])
plt.annotate('Source: Coin Metrics ATLAS',xy=(1, -0.195), xycoords='axes fraction',color='black',xytext=(-8, 6), textcoords='offset pixel
s',horizontalalignment='right',verticalalignment='bottom')
plt.gca().set_yticklabels(['${:,.2f}M'.format(x/1000000) for x in plt.gca().get_yticks()]);
```


USDC Supply on Uniswap V3
Tracked Liquidity Pools
$700.00M
$600.00M
$500.00M
$400.00M
$300.00M
$200.00M
$100.00M
$0.00M
Block Height
Source: Coin Metrics ATLAS