# Content Page

# Abstract

In this report, the trends of 6 dark net markets (DNMs) are analyzed using the Kilos dataset. We analyzed their top vendors and possible reasons for their performances, such as user ratings.

The Kilos dataset is then augmented with the Agora dataset to categorize each vendor according to the type of items/services sold. With the categorization of vendors, we can identify which vendors are prevalent in each sub-category.

Lastly, the Kilos dataset is verified using 2 other datasets, the Silk Road 2 dataset and the Darknet Market Cocaine dataset. Despite being collected at different times, these datasets share a small number of vendors. The ratings of these common vendors are used to determine if there are any discrepancies with the Kilos dataset.

# Introduction

Online marketplaces hosted on Tor can provide a wide range of escrow services between buyers and sellers. This allows for buyers and sellers to transact using Bitcoin or other cryptocurrencies in exchange for services and products, such as drugs, weapons or hacking services.

As a result, multiple dark net markets (DNMs) had cropped up. The most famous was Silk Road 1, starting in 2011. Over the years, multiple other DNMs have emerged while older DNMs were taken down.

In this report, we aim to analyze the trends in the 6 major DNMs (Apollon, CannaHome, Empire, Samsara, Cryptonia, Cannazon) and their associated vendors.

The main datasets that we will be using are as follows:

1. Kilos dataset [1]. This dataset contains information such as site, vendor, timestamp, score and value_btc. It is collected from a search engine and contains information about the dark web vendors along with their item values as well as a review score. It contains data about the vendors of the 6 DNMs of interest. The dataset is dated from 2018 March to 2020 January.

2. Agora dataset [2]. This dataset contains similar marketplace data as well as categories, locations and other remarks. It contains data about the vendors of Agora (another DNM). The dataset is dated from 2014-2015.

All the code used in this analysis can be found in the accompanying Jupyter notebook file, *Dark Web Analysis.ipynb*.

# Problem Statement

In this report, we will be attempting to understand how vendors (of the 6 DNMs of interest) are able to have an impact in the market and the associated trends of the 6 DNMs. Some possible questions are what categories, vendors and sites dominate these DNMs and what factors are responsible in making a vendor successful (in terms of lifespan and earnings).

# Augmenting Kilos Dataset

| | site | vendor | timestamp | score | value_btc |
|---|---|---|---|---|---|
| 0 | Samsara | Doubleup | 1970-01-01 | 1.0 | 0.00022 |
| 1 | Samsara | SocialPharma | 1970-01-01 | 1.0 | 0.00492 |
| 2 | Samsara | REAL-monoko | 1970-01-01 | 1.0 | 0.00381 |
| 3 | Samsara | druggiebearsofficial | 1970-01-01 | 1.0 | 0.03159 |
| 4 | Samsara | DrunkDragon | 1970-01-01 | 1.0 | 0.00011 |

Table 1. Kilos dataset

The Kilos dataset has 5 features:

1. DNM site (Apollon, CannaHome, Cannazon, Cryptonia, Empire, & Samsara)
2. Vendor name
3. Timestamp (has been converted from unix epoch seconds)
4. Score (Review given by buyer. +1 is positive, -1 is negative, 0 is neutral)
5. Value of transaction in bitcoin

| | Vendor | Category | Item | ItemDescription | Price_BTC | Origin | Destination | Rating | Remarks |
|---|---|---|---|---|---|---|---|---|---|
| 0 | CheapPayTV | Services/Hacking | 12 Month HuluPlus gift Code | 12-Month HuluPlus Codes for $25. They are wort... | 0.050270 | Torland | NaN | 4.96/5 | NaN |
| 1 | CheapPayTV | Services/Hacking | Pay TV Sky UK Sky Germany HD TV and much mor... | Hi we offer a World Wide CCcam Service for En... | 0.152420 | Torland | NaN | 4.96/5 | NaN |
| 2 | KryptykOG | Services/Hacking | OFFICIAL Account Creator Extreme 4.2 | Tagged Submission Fix Bebo Submission Fix Adju... | 0.007000 | Torland | NaN | 4.93/5 | NaN |
| 3 | cyberzen | Services/Hacking | VPN > TOR > SOCK TUTORIAL | How to setup a VPN > TOR > SOCK super safe enc... | 0.019017 | NaN | NaN | 4.89/5 | NaN |
| 4 | businessdude | Services/Hacking | Facebook hacking guide | . This guide will teach you how to hack Faceb... | 0.062018 | Torland | NaN | 4.88/5 | NaN |

Table 2. Agora dataset

The Agora dataset has 9 features:

1. Vendor names
2. Category of item sold
3. Item sold
4. Item description
5. Value of item in bitcoin
6. Origin
7. Destination
8. Rating
9. Remarks

As both Kilos and Agora datasets have the *vendor* feature, we can add the *category* feature to the Kilos dataset from the Agora dataset, for vendors that are in both datasets.

Using the following code, we are able to to match the vendors in the Kilos dataset with the categories in the Agora dataset. This will enable us to augment the Kilos dataset. However, note that not all the vendors in the Agora dataset exist within the Kilos dataset.

```
columns = ['site', 'vendor', 'timestamp', 'score', 'value_btc']
kilos_df = pd.read_csv('Darkweb data scrape.csv', usecols=columns)

agora_df = pd.read_csv('Agora/Agora.csv',  encoding="latin1")
agora_df.columns = [x.replace(" ", "") for x in agora_df.columns]

# Get unique vendor-category counts from agora df
vendor_cat = agora_df.groupby(['Vendor','Category']).size().reset_index().rename(columns={0:'count'})

# Filter for categories that are valid ie have a '/' inside
valid_vendor_cats = vendor_cat[vendor_cat['Category'].str.contains("/")].copy()

# Change vendors to lowercase
valid_vendor_cats['Vendor'] = valid_vendor_cats['Vendor'].str.lower()
kilos_df['vendor'] = kilos_df['vendor'].str.lower()

# Do a left join
joined_df = kilos_df.merge(valid_vendor_cats, left_on='vendor', right_on='Vendor', how='left',
                          indicator = True)
joined_df = joined_df[joined_df['_merge'] == 'both']

# Extract only the interested columns
joined_df = joined_df[['site', 'vendor', 'timestamp', 'score', 'value_btc', 'Category']].copy()

# Clean the category column
joined_df['category'], joined_df['sub_category_one'], joined_df['sub_category_two'] = '', '', ''
for index, row in joined_df.iterrows():
    string_cat = row['Category']

    # Clean category. Obtain primary category and sub category one
    category = string_cat
    subcategory_one = ''
    if '/' in string_cat:
        category = string_cat.split('/')[0]
        subcategory_one = string_cat.split('/')[1]

    # Check for sub category two
    subcategory_two = ''
    if string_cat.count('/') > 1:
        subcategory_two = string_cat.split('/')[2]

    # Store the values
    joined_df.at[index, 'category'] = category
    joined_df.at[index, 'sub_category_one'] = subcategory_one
    joined_df.at[index, 'sub_category_two'] = subcategory_two

# Drop old Category column
joined_df = joined_df.drop(['Category'], axis=1)

# See what unique categories exist now
joined_df['category'].unique()

array(['Drugs', 'Drug paraphernalia', 'Services', 'Data', 'Forgeries',
```

```
       'Info', 'Tobacco', 'Counterfeits', 'Information', 'Weapons'],
      dtype=object)

# See how the dataset looks like now
joined_df.head()
```

| | site | vendor | timestamp | score | value_btc | Category | category | sub_category_one | sub_category_two |
|---|---|---|---|---|---|---|---|---|---|
| **36** | Samsara | instrument | 1970-01-01 | 1.0 | 0.00623 | Drugs/Ecstasy/MDA | Drugs | Ecstasy | MDA |
| **37** | Samsara | instrument | 1970-01-01 | 1.0 | 0.00623 | Drugs/Ecstasy/MDMA | Drugs | Ecstasy | MDMA |
| **38** | Samsara | instrument | 1970-01-01 | 1.0 | 0.00623 | Drugs/Ecstasy/Pills | Drugs | Ecstasy | Pills |
| **39** | Samsara | instrument | 1970-01-01 | 1.0 | 0.00623 | Drugs/Psychedelics/LSD | Drugs | Psychedelics | LSD |
| **40** | Samsara | instrument | 1970-01-01 | 1.0 | 0.00623 | Drugs/Stimulants/Meth | Drugs | Stimulants | Meth |

Table 3. Augmented Kilos dataset

After augmenting the dataset with the *category* feature and dropping rows that do not have a matching category, the total number of rows for the Kilos dataset had decreased from 235,862 to 56,953.

Despite the decrease in the number of rows, 56,953 is still a substantial number for us to do our subsequent categorical analysis on.

# Category Sales & Popularity

With the augmented *category* feature, we can analyze the Kilos dataset with respect to the categories of items sold. There are a total of 8 categories with the following percentages.

```
# Get percentages of each main category
kilos_df['category'].value_counts(normalize=True) * 100
```

| Category | Percentage (%) | | Category | Percentage (%) |
|---|---|---|---|---|
| Drugs | 87.663512 | | Forgeries | 0.986779 |
| Services | 3.897951 | | Counterfeits | 0.953418 |
| Information | 3.894439 | | Drug Paraphernalia | 0.790125 |
| Tobacco | 1.676821 | | Weapons | 0.136955 |

Table 4. Percentage distribution for each category

An overwhelming majority of the 6 DNMs are used for drug-related purchases. The next significant categories are the Information (3.89%) and Services sectors(such as hacking) (3.897%).

Interestingly, Forgeries, Counterfeits and Weapons market share are each below 1%. Perhaps this indicates that the sale of these services is not captured within these 6 DNMs.

## Top Vendors for Drugs

As Drugs represent an overwhelming majority of the dataset (87.66%), we will ascertain which vendors are the top sellers for Drugs.

```
# Get top vendors for drugs
drug_df = kilos_df[kilos_df['category'] == 'Drugs'].copy()
(drug_df['vendor'].value_counts(normalize=True) * 100).head(10)
```

| Vendor | Percentage |
|---|---|
| quicklick | 20.315661 |
| uknextday | 10.485309 |
| dreamweaver | 7.426843 |
| angelina | 6.749855 |
| grandwizardslair | 4.790995 |
| boomers | 4.374387 |
| instrument | 3.785527 |
| nextgeneration | 3.755483 |
| calicannabisclub | 3.074489 |
| vitaminstore | 2.571755 |

Table 5. Top vendors selling Drugs by Percentage

We can see that *quicklick* has a sizable portion of the Drugs market captured, at 20.3%. The next seller (*uknextday*) could only manage half of that market base, at 10.4%. The rest of the vendors only managed to capture small portions of the remaining market share.

## Top Vendors across other Categories

We can do a similar analysis on the remaining categories to determine which are the top vendors. There could be a possible correlation among the vendors.

```
# Get top 10 vendors for each category
top_vendors_overall = {}
for category in kilos_df.category.unique():
    sub_df = kilos_df[kilos_df['category'] == category]
    top_ten_vendors = sub_df['vendor'].value_counts(normalize = True).head(10)
    top_ten_vendors = top_ten_vendors.index.values
    top_vendors_overall[category] = top_ten_vendors
pd.DataFrame(dict([ (k, pd.Series(v)) for k,v in top_vendors_overall.items() ]))
```

| | Drugs | Drug paraphernalia | Services | Information | Forgeries | Tobacco | Counterfeits | Weapons |
|---|---|---|---|---|---|---|---|---|
| 0 | quicklick | joeybagadonuts | sexyhomer | dionysos | namedeclined | maling47 | sexyhomer | dionysos |
| 1 | uknextday | namedeclined | joeybagadonuts | namedeclined | lindalovelace | NaN | dionysos | NaN |
| 2 | dreamweaver | dionysos | namedeclined | sexyhomer | arctic | NaN | hackyboy | NaN |
| 3 | angelina | NaN | hackyboy | ukpharma | medsguru | NaN | moggymoo | NaN |
| 4 | grandwizardslair | NaN | biocanna | hackyboy | kingscan | NaN | NaN | NaN |
| 5 | boomers | NaN | weednation | colombiaconnection | threekings | NaN | NaN | NaN |
| 6 | instrument | NaN | shponglestarship | etimbuk | NaN | NaN | NaN | NaN |
| 7 | nextgeneration | NaN | etimbuk | goingpostal | NaN | NaN | NaN | NaN |
| 8 | calicannabisclub | NaN | medsguru | medsguru | NaN | NaN | NaN | NaN |
| 9 | vitaminstore | NaN | dutchmagic | NaN | NaN | NaN | NaN | NaN |

Table 6. Top 10 vendors for each category

We can see that for the categories Drug paraphernalia, Forgeries, Tobacco, Counterfeits and Weapons, there are barely 10 vendors. This could indicate that either these categories are dominated by these few vendors or the market for these categories are not captured in the Kilos dataset.

The only notable observation is that the vendor *sexyhomer* is involved in the selling of Services, Information and Counterfeits.

None of the sellers for Drugs are involved in the selling of other categories of items in the 6 DNMs.

# Sales over Time (Overall)

For this analysis, we will look at both the original Kilos dataset and the augmented Kilos dataset. To analyze with respect to time, we first remove all rows which have invalid dates. Rows that have invalid timestamps would have their time set to unix epoch 0 ie 1st January 1970.

```
# Extract rows which have a valid time
time_df = kilos_df[kilos_df['timestamp'] > '1970-01-01'].copy()
print('Length before time filtering:', len(kilos_df))
print('Length after time filtering', len(time_df))

Length before time filtering: 56953
Length after time filtering 54604
```

After filtering out these rows, we still have 54,604 rows, which is still substantial.

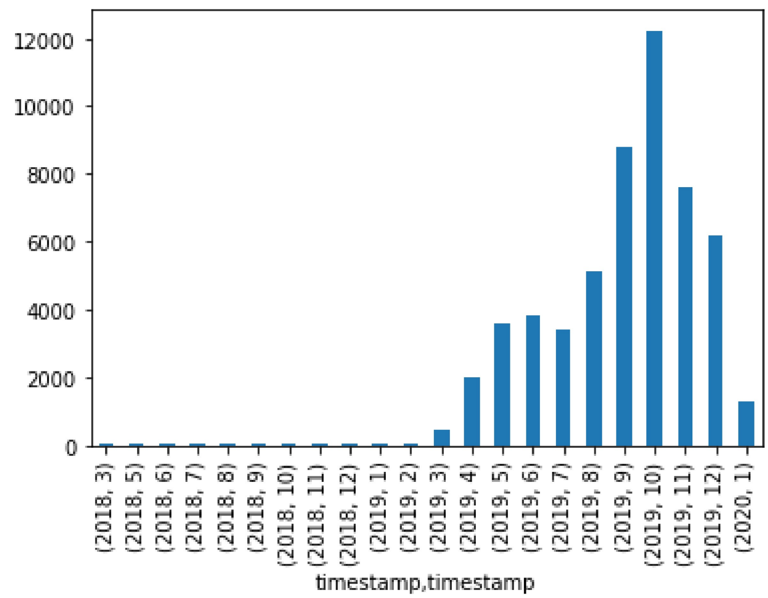


Figure 1. Time series plot for sales of Kilos dataset with matching vendor categories

From the plot, we can see that most of the traffic comes from May 2019 to December 2019.

When we consider the original dataset (which does not have matching vendor categories), the plot is also similar.

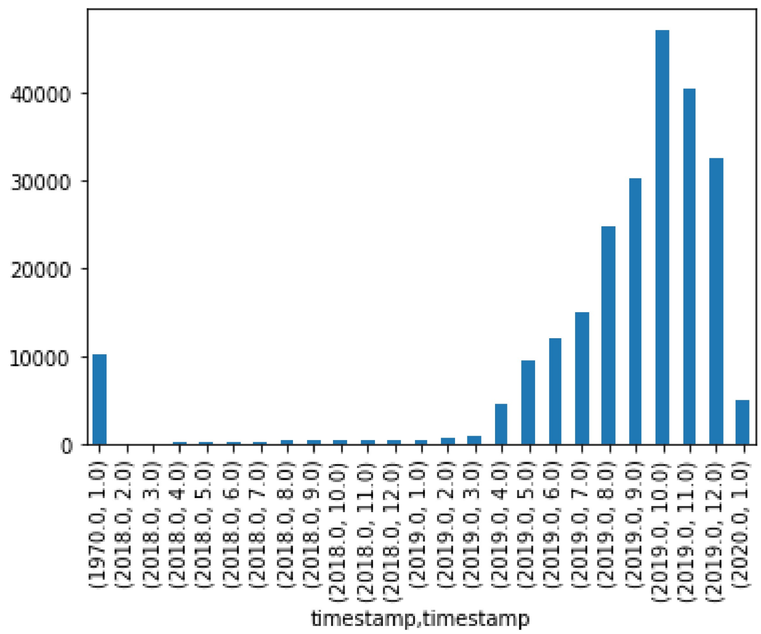


Figure 2. Time series plot of sales for original Kilos dataset

Most of the traffic lies between May 2019 to December 2019.

# Sales over Time (Categories)

Similar to the previous plot, we can further investigate the datetime plots for each of the 8 categories.

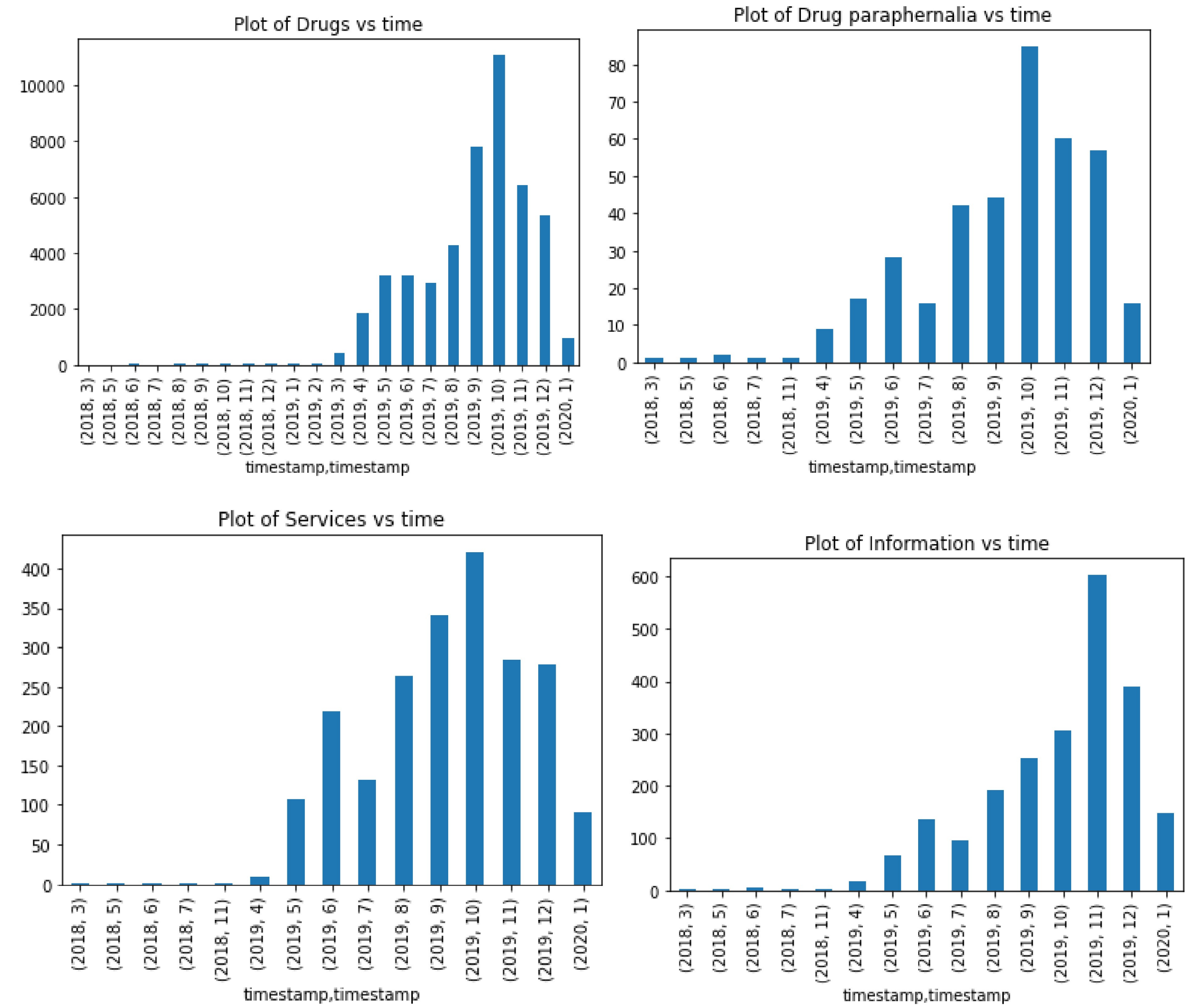


Figure 3. Time series plot of sales for categories: Drugs, Drug Paraphernalia, Services, Information

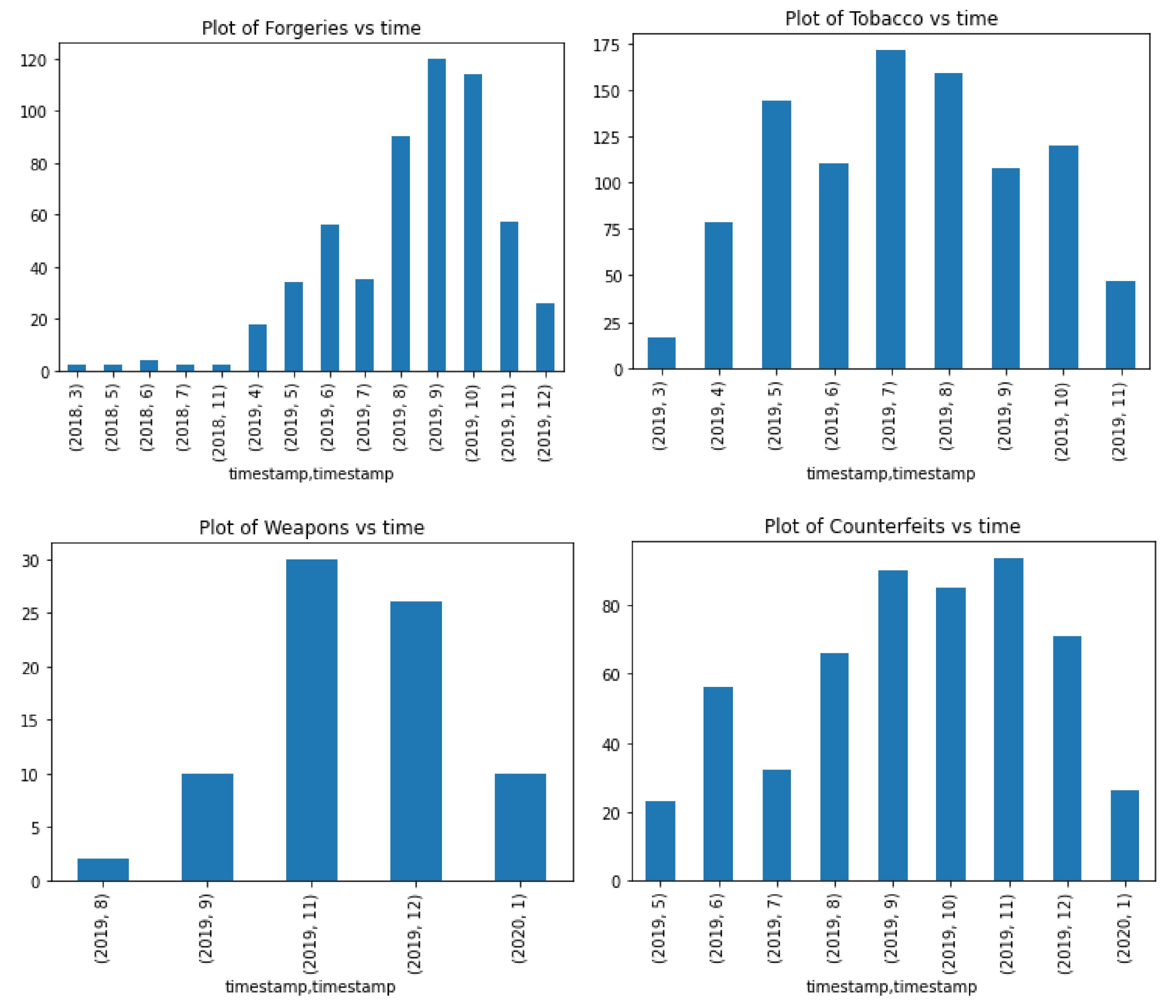


Figure 4. Time series plot of sales for categories: Forgeries, Tobacco, Weapons, Counterfeits

From the above graphs, all the categories appear to peak in the range of September 2019 to November 2019. The exception to this is Tobacco, which peaks in July 2019 instead.

This indicates that perhaps there is a peak in dark web transactions that occurs from September to November for each year.

# Sales over Time (Sites)

A time series plot of sales can be done for each of the 6 DNM sites.

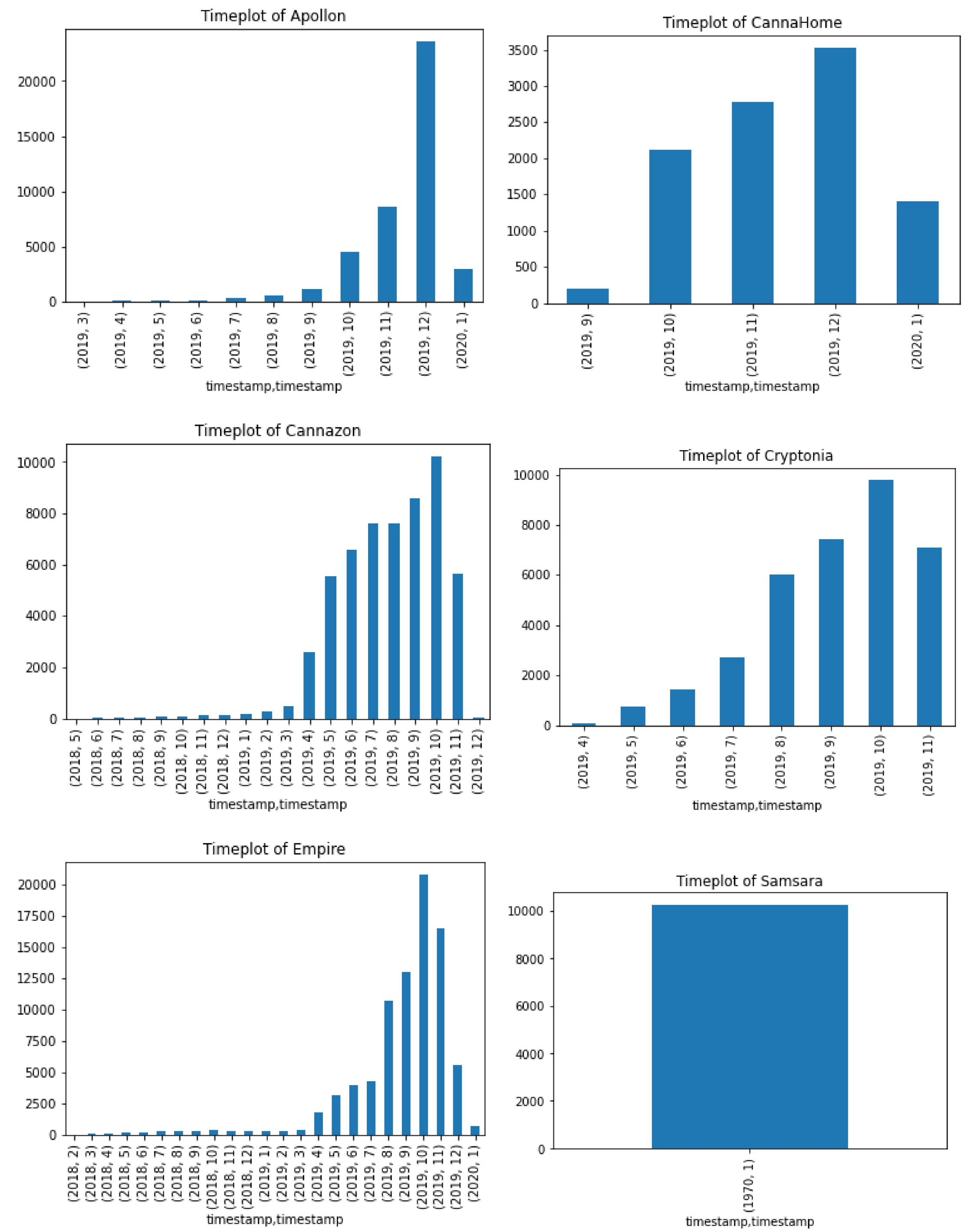


Figure 5. Time series plot of sales for each site

With the exception of Samsara (which has invalid timestamps), the other 5 sites are active largely around the 2019 August to 2019 December time period.

2 sites are of particular interest. CannaHome and Apollon both peaked in 2019 December while Empire, Cannazon and Cryptonia peaked in 2019 October.

# Popular Markets (Category)

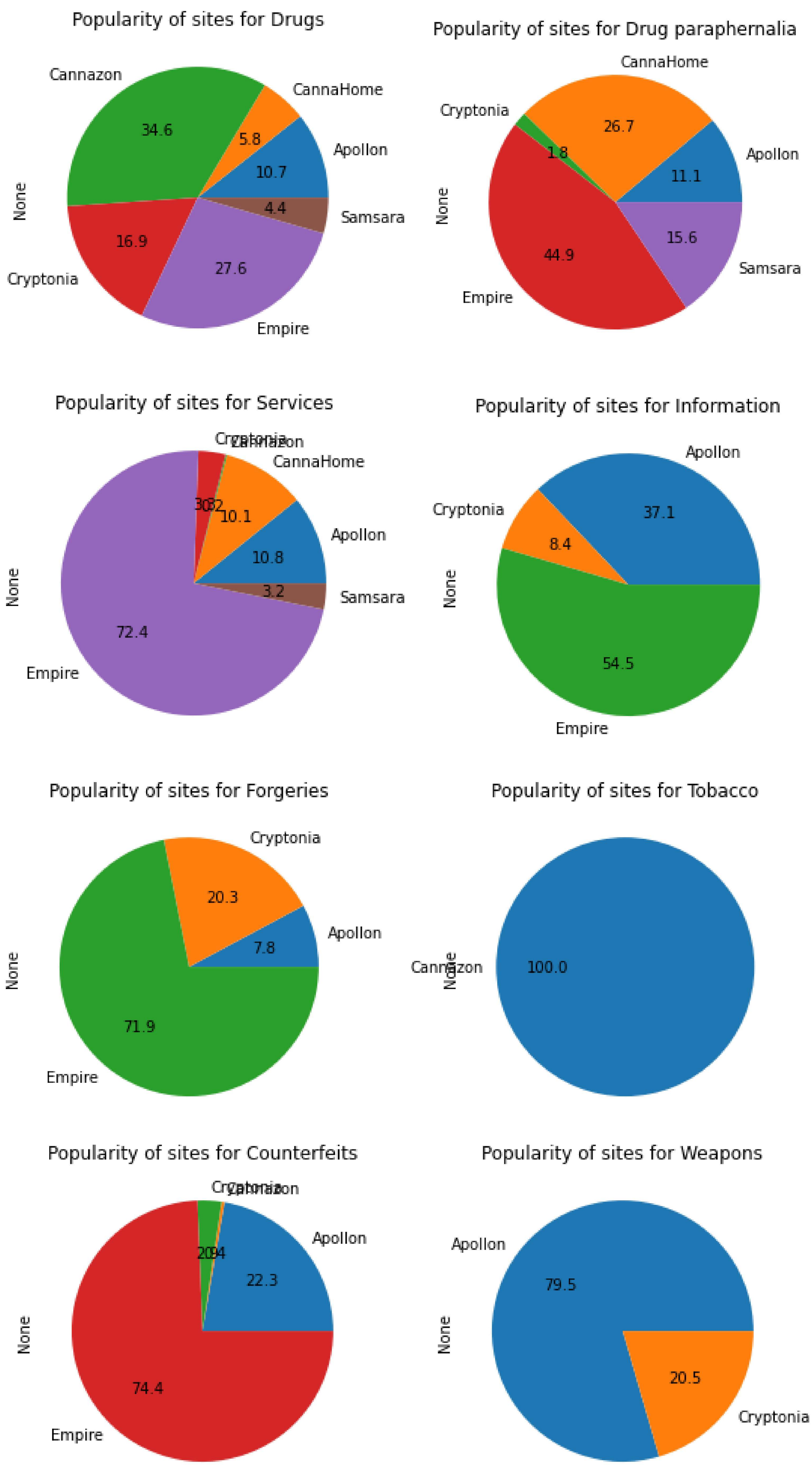

Figure 6. Pie charts of each category divided into site markets

Next, we can analyze which of the 6 DNMs dominate each category.

From the pie charts, Empire is the most popular site for Counterfeits, Forgeries, Information, Services and Drug Paraphernalia.

Cannazon is the most popular for Tobacco and Drugs whereas Apollon is the most popular for Weapons.

This information will be important for law enforcement. For instance, to monitor illegal arms sales, the Apollon market should be closely monitored.

# Vendor Lifetimes

We will analyze which vendors have the top 10 longest lifetimes in this dataset. As this analysis does not require the *category* column, we will be using the original Kilos dataset.

```
vendor_list = original_df['vendor'].unique()
vendor_timespans = []

for vendor in vendor_list:
    vendor_df = original_df[original_df['vendor'] == vendor]
    vendor_start = vendor_df['timestamp'].min()
    vendor_latest = vendor_df['timestamp'].max()
    timespan = vendor_latest - vendor_start
    vendor_timespans.append(timespan)

timespan_df = pd.DataFrame({
    'vendor': vendor_list,
    'timespan': vendor_timespans
})

timespan_df = timespan_df.sort_values(by=['timespan'], ascending = False)
timespan_df.head(10)
```

| | vendor | timespan |
|---|---|---|
| **1** | DrunkDragon | 675 days 09:18:00 |
| **0** | ofgrey | 670 days 00:20:00 |
| **3** | Goldoratt | 649 days 06:13:00 |
| **5** | MoneyMule | 637 days 13:06:00 |
| **4** | rvaska | 623 days 12:06:00 |
| **6** | HeinekenExpress | 616 days 13:32:00 |
| **9** | tvman | 605 days 04:26:00 |
| **2** | namedeclined | 603 days 07:43:00 |
| **10** | sixandeight | 600 days 09:00:00 |
| **11** | jimmy2018 | 597 days 00:15:00 |

Table 7. Top 10 vendors with the longest timespans

Intuitively, we would like to know how these vendors are the longest in the market. One possible reason is that they provide good customer service and thus, have high ratings.

```
average_ratings, median_ratings = [],[]
for vendor in vendors:
    vendor_df = original_df[original_df['vendor'] == vendor]
    average_rating = vendor_df['score'].mean()
    median_rating = vendor_df['score'].median()
    var_rating = vendor_df['score'].var()

    average_ratings.append(average_rating)
    median_ratings.append(median_rating)


vendor_rating_df = pd.DataFrame({
    'vendor': vendors,
    'average_rating': average_ratings,
    'median_rating': median_ratings
})
vendor_rating_df.head(10)
```

| | vendor | average_rating | median_rating |
|---|---|---|---|
| **1** | DrunkDragon | 0.520179 | 1.0 |
| **0** | ofgrey | 0.875000 | 1.0 |
| **3** | Goldoratt | 0.537755 | 1.0 |
| **5** | MoneyMule | 0.811554 | 1.0 |
| **4** | rvaska | 0.597345 | 1.0 |
| **6** | HeinekenExpress | 0.337818 | 1.0 |
| **9** | tvman | 0.758818 | 1.0 |
| **2** | namedeclined | 0.925743 | 1.0 |
| **10** | sixandeight | 0.829046 | 1.0 |
| **11** | jimmy2018 | 0.923913 | 1.0 |

Table 8. Longest lasting vendors by ratings

We can see that for the top 10 longest vendors, their median ratings is 1 while their average ratings are mostly positive (more than 0). Recall that +1 is for a positive review, -1 is for a negative review, 0 is for a neutral review.

To further investigate this theory, we can check if there is a statistical correlation between *ratings* and *vendor timespan.*

```
# See if there is a correlation between vendor timespan and ratings
timespan_df['average_rating'], timespan_df['median_rating'] = '', ''

for index, row in timespan_df.iterrows():
    current_vendor = row['vendor']

    # Extract average and median ratings
    vendor_df = original_df[original_df['vendor'] == current_vendor]
    average_rating = vendor_df['score'].mean()
    median_rating = vendor_df['score'].median()

    # Store values
    timespan_df.loc[index, 'average_rating'] = average_rating
    timespan_df.loc[index, 'median_rating'] = median_rating

timespan_df = timespan_df.astype({'average_rating': 'int32',
                                  'median_rating': 'int32'})

# Convert time delta to seconds
timespan_df['timespan'] = timespan_df['timespan'] / np.timedelta64(1, 's')

timespan_df.corr()
```

| | timespan | average_rating | median_rating |
|---|---|---|---|
| **timespan** | 1.000000 | -0.493012 | 0.034303 |
| **average_rating** | -0.493012 | 1.000000 | 0.166549 |
| **median_rating** | 0.034303 | 0.166549 | 1.000000 |

Table 9. Correlation table between ratings and timepan

From the correlation, average rating has a moderate impact (-0.49) on the timespan of vendors. However, the relation is opposite. The higher the average rating, the smaller the timespan (ie lifespan) of the vendor. This is a very interesting observation.

A possible explanation is that during the data collection for the Kilos dataset, review score ratings were generated using sentiment analysis. The author of the dataset stated that *"E.g. 'this shit is the bomb!' might be classified negatively despite context telling us that this is a positive review*". As this is the dark web, comments tend to be unfiltered. Hence, good vendors may have lots of reviews like the above which use vulgarities. These comments would then be classified as negative instead.

# Vendor Earnings

From the original Kilos dataset, we can determine which vendors earn the most income.

```
vendors = original_df['vendor'].unique()
vendor_earnings = []

for vendor in vendors:
    vendor_df = original_df[original_df['vendor'] == vendor]
    vendor_earning = vendor_df['value_btc'].sum()
    vendor_earnings.append(vendor_earning)

earnings_df = pd.DataFrame({
    'vendor': vendors,
    'total_btc_earnings': vendor_earnings
})
earnings_df = earnings_df.sort_values(by=['total_btc_earnings'], ascending = False)
earnings_df.head(10)
```

| | vendor | total_btc_earnings |
|---|---|---|
| **436** | CaliTerps | 2088.068601 |
| **365** | Insta | 384.991341 |
| **190** | GreenSupreme | 325.015451 |
| **434** | MexicanConnection | 303.519171 |
| **423** | cabackdoormedical | 172.616300 |
| **880** | NancyBotwin2019 | 156.940364 |
| **652** | iROCKice | 149.513500 |
| **331** | VanillaSurf | 131.068990 |
| **274** | tonystarkweed | 129.423152 |
| **605** | CTB | 123.932000 |

Table 10. Top 10 vendors by earnings

The vendor *CaliTerps* earns substantially more than the rest of the vendors.

```
# See what categories are these vendors in
top_vendors = earnings_df.head(10)['vendor']

# Read in dataset with categories
augmented_kilo = pd.read_csv('web_scrape_category.csv')
augmented_kilo['timestamp'] = pd.to_datetime(augmented_kilo['timestamp'])

# See if any of the top vendors are in the augmented kilo dataset
for vendor in top_vendors:
    if vendor in cat_df['vendor'].unique():
        print(vendor)

<Nothing printed>
```

Unfortunately, none of the vendors in this top 10 list have records in the augmented kilos dataset. Hence, we cannot determine which categories these vendors belong to.

However, we can analyze the relationship between the earnings of the vendors with their ratings.

```
# Analyze relationship between earnings and ratings
earnings_df['average_rating'], earnings_df['median_rating'] = -1, -1
for index, row in earnings_df.iterrows():
    vendor = row['vendor']

    # Get ratings corresponding to vendor
    vendor_ratings = original_df[original_df['vendor'] == vendor]

    # Store average rating
    earnings_df.loc[index, 'average_rating'] = vendor_ratings['score'].mean()
    earnings_df.loc[index, 'median_rating'] = vendor_ratings['score'].median()

earnings_df.head(10)
```

| | vendor | total_btc_earnings | average_rating | median_rating |
|---|---|---|---|---|
| **436** | CaliTerps | 2088.068601 | 0.986340 | 1.0 |
| **365** | Insta | 384.991341 | 0.997354 | 1.0 |
| **190** | GreenSupreme | 325.015451 | 0.993394 | 1.0 |
| **434** | MexicanConnection | 303.519171 | 1.000000 | 1.0 |
| **423** | cabackdoormedical | 172.616300 | 0.991424 | 1.0 |
| **880** | NancyBotwin2019 | 156.940364 | 1.000000 | 1.0 |
| **652** | iROCKice | 149.513500 | 1.000000 | 1.0 |
| **331** | VanillaSurf | 131.068990 | 0.984390 | 1.0 |
| **274** | tonystarkweed | 129.423152 | 0.995203 | 1.0 |
| **605** | CTB | 123.932000 | 1.000000 | 1.0 |

Table 11. Top vendors by earnings with ratings

```
earnings_df.corr()
```

| | total_btc_earnings | average_rating | median_rating |
|---|---|---|---|
| **total_btc_earnings** | 1.000000 | 0.024957 | 0.007493 |
| **average_rating** | 0.024957 | 1.000000 | 0.738488 |
| **median_rating** | 0.007493 | 0.738488 | 1.000000 |

Table 12. Correlation between earnings and ratings

From Table 11, the top 10 vendors have an almost perfect rating of 1.0 for their mean and median ratings. However, this relationship does not extend to the rest of the vendors as the correlation coefficients between *average_rating* and *median_rating* are a low 0.025 and 0.007 respectively.

To investigate this further, we limit our correlation analysis to the top 13 vendors in terms of earnings.

```
# Top 13 in terms of earnings
top_earnings = earnings_df.head(13)
top_earnings.corr()
```

| | total_btc_earnings | average_rating | median_rating |
|---|---|---|---|
| **total_btc_earnings** | 1.000000 | -0.526313 | NaN |
| **average_rating** | -0.526313 | 1.000000 | NaN |
| **median_rating** | NaN | NaN | NaN |

Table 13. Correlation between earnings and ratings for top 13 vendors

When we limit to the top 13 vendors, the correlation between average rating and total earnings is much more significant, with a magnitude of 0.526.

Hence, we can conclude that for the top earning vendors, their ratings are highly correlated with their earnings. Hence, ratings do matter for earning top bitcoin sales.

# Verifying with other datasets

There are 2 additional datasets that are similar to the Kilos dataset. These 2 datasets can be joined with the original Kilos dataset via the *vendor* column. However, the number of overlapping vendors is rather small. Hence, instead of generating new features, we can use these 2 datasets to verify the information in the Kilos dataset. This is as both of these 2 datasets contain a *rating* column.

The datasets are:

1. Darknet market cocaine listings [3]. This dataset is obtained from web-scraping techniques done in 2017.

| product_title | ships_from_to | grams | quality | btc_price | cost_per_gram | cost_per_gram_pure | escrow | product_link | ... | ships_to_SE |
|---|---|---|---|---|---|---|---|---|---|---|
| !!!!!INTRO OFFER!!!!! 1GR COCAINE 90% | NL → EU | 1.0 | 90.0 | 0.02577 | 0.02577 | 0.028633 | 1 | http://lchudifyeqm4ldjj.onion/viewProduct?offe... | ... | False |
| !!!!!INTRO OFFER!!!!! 2GR COCAINE 90% | NL → EU | 2.0 | 90.0 | 0.05150 | 0.02575 | 0.028611 | 1 | http://lchudifyeqm4ldjj.onion/viewProduct?offe... | ... | False |
| !!!INTRO!!! 0.5G COCAINE 89% - STRAIGHT FROM T... | NL → EU | 0.5 | 89.0 | 0.01649 | 0.03298 | 0.037056 | 1 | http://lchudifyeqm4ldjj.onion/viewProduct?offe... | ... | False |
| !1G! C O L O M B I A N C O C A I N E - 89% PURITY | FR → EU | 1.0 | 89.0 | 0.04120 | 0.04120 | 0.046292 | 1 | http://lchudifyeqm4ldjj.onion/viewProduct?offe... | ... | False |
| ** 1 Gram 87% Pure Uncut Colombian Cocaine ** | NL → WW | 1.0 | 87.0 | 0.03400 | 0.03400 | 0.039080 | 1 | http://lchudifyeqm4ldjj.onion/viewProduct?offe... | ... | False |

Table 14. Darknet market cocaine dataset

2. Silk road 2 listings [3]. This dataset is obtained by extracting data from the HTML files obtained by Gwern Branwen's large data trawl. The dataset is dated from December 2013 to November 2014.

| | Title | Sellerid | PriceUSD | PriceBTC | Rating | Reviews | Origin | Destination | Category | Subcategory | Market | Date |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Ray Ban Tech RB3460 001 Aviator/Flip Out/ Sung... | FoxyGirl | 61.161542 | 0.098232 | NaN | NaN | China | Worldwide | Apparel | None | SilkRoad2 | 2014-07-17 |
| 1 | Ray Ban RB3025 Aviator Classic Sunglasses Replica | FoxyGirl | 37.962594 | 0.060972 | NaN | NaN | China | Worldwide | Apparel | None | SilkRoad2 | 2014-07-17 |
| 2 | Rolex - Watch Box (AAA Grade Replica) | RepAAA | 100.429777 | 0.161301 | NaN | NaN | Hong Kong, (China) | Worldwide | Apparel | None | SilkRoad2 | 2014-07-17 |
| 3 | Ray Ban RB3016 - W0365 \| Clubmaster Sunglasses... | FoxyGirl | 45.343794 | 0.072827 | NaN | NaN | China | Worldwide | Apparel | None | SilkRoad2 | 2014-07-17 |
| 4 | Rolex - Submariner 2Tone YG/SS Black [Replica] | RepAAA | 149.635082 | 0.240330 | NaN | NaN | Hong Kong, (China) | Worldwide | Apparel | None | SilkRoad2 | 2014-07-17 |

Table 15. Silk road 2 dataset

# Darknet Market Cocaine Listings

This dataset has data for only 13 vendors from the original Kilos dataset. The 13 vendors are: 'onenation', 'AmsterdamNL', 'QuickLick', 'homegrow1919', 'Naturalminds', 'dutchcandyshop', 'CheechandChongAB', 'ukwhite', 'Foreigner', 'superdrugz', 'DutchSolution'.

We can use the darknet market dataset to verify that the ratings given to the vendors correlate with each other.

```
Data from Kilos dataset:

vendor
AmsterdamNL         0.949153
CheechandChongAB    0.904762
DutchSolution       1.000000
Foreigner           1.000000
Naturalminds        0.910138
QuickLick           0.911269
dutchcandyshop      0.886957
homegrow1919        1.000000
onenation           0.972093
superdrugz          1.000000
ukwhite             1.000000
Name: score, dtype: float64
```

Table 16. Average ratings from Kilos dataset

```
Data from Cocaine dataset:

vendor_name
AmsterdamNL         4.93
CheechandChongAB    4.95
DutchSolution       4.86
Foreigner           4.98
Naturalminds        4.97
QuickLick           4.87
dutchcandyshop      4.66
homegrow1919        4.96
onenation           4.84
superdrugz          4.94
ukwhite             4.88
Name: rating, dtype: float64
```

Table 17. Average ratings from Cocaine dataset

The Cocaine dataset ratings ranges from 0 to 5.

Firstly, we can observe that from the Kilos dataset, these vendors have a largely favourable review score (close to 1). This correlates with the Cocaine dataset as most of the scores are close to 5 (the maximum).

Next, we can observe that the vendor with the lowest score in the Kilos dataset in *dutchcandyshop* (0.8869). This correlates with the Cocaine dataset, where the same vendor has the lowest score of 4.66.

Hence, using the Cocaine dataset, we can verify that the information in the Kilos dataset is correct.

# Silk Road 2 Listings

This dataset has data for only 23 vendors from the original Kilos dataset. The 23 vendors are: 'instrument', 'namedeclined', 'Angelina', 'theanchor', 'BioCanna', 'GrandWizardsLair', 'CalisFinest', 'revenantchild', 'arctic', 'nzt48givesyouwings', 'HonestCocaine', 'MedIndia', 'MarleysMainMan', 'weednation', 'shroomdude', 'TripWithScience', 'WizardofOZs', 'ModernLove', 'ThreeKings', 'DutchMagic', 'calitreez', 'TheResistance', 'HollandOnline'

As the silk road 2 dataset is dated in 2014, the above vendors could be considered as one of the older and more established vendors. They were previously operating in Silk Road 2 in 2014 before moving on to the newer 6 sites in 2019.

Similar to the previous analysis, we can verify that the ratings of the overlapping vendors correlate with each other.

```
Data from Kilos dataset:

vendor
Angelina              0.743323
BioCanna              0.969512
CalisFinest           0.831395
DutchMagic            0.666667
GrandWizardsLair      0.979933
HollandOnline         1.000000
HonestCocaine         0.833333
MarleysMainMan        0.994152
MedIndia              0.957447
ModernLove            1.000000
TheResistance         0.977778
ThreeKings            1.000000
TripWithScience       0.968750
WizardofOZs           1.000000
arctic                0.888889
calitreez             1.000000
instrument            0.994709
namedeclined          0.925743
nzt48givesyouwings    1.000000
revenantchild         1.000000
shroomdude            1.000000
theanchor             0.800000
weednation            1.000000
Name: score, dtype: float64
```

Table 18. Average ratings from Kilos dataset

```
Data from Silk Road dataset:

Sellerid
Angelina              4.627424
BioCanna              4.609023
CalisFinest           4.503613
DutchMagic            4.656029
GrandWizardsLair           NaN
HollandOnline         4.826668
HonestCocaine         4.672421
MarleysMainMan        4.550000
MedIndia              4.050862
ModernLove            4.894915
TheResistance         4.835559
ThreeKings            4.611756
TripWithScience       4.850000
WizardofOZs           4.190909
arctic                4.750000
calitreez                  NaN
instrument            4.800581
namedeclined               NaN
nzt48givesyouwings    4.786532
revenantchild         4.738312
shroomdude            4.650000
theanchor             3.700000
weednation                 NaN
Name: Rating, dtype: float64
```

Table 19. Average ratings from Silk Road dataset

From the Kilos dataset, these vendors have a generally positive rating (close to 1). From the Silk Road dataset, these same vendors also have quite favourable reviews (close to 5).

Hence, using these 2 datasets, we can verify that the data in the Kilos dataset is correct.

# Discussion & Conclusion

In this report, analysis of the 6 DNMs (from the Kilos dataset) have revealed the following.

1. Overall sales (for each category and DNM site) peaked in the September 2019 to December 2019 time period.
2. The top 3 DNM sites that dominate the categories are:
    a. Empire - Counterfeits, Forgeries, Information, Services, Drug Paraphernalia
    b. Cannazon - Tobacco, Drugs
    c. Apollon - Weapons
3. Most of the top vendors specialized in their own category of items with the exception of *sexyhomer*. *sexyhomer* is involved in the Services, Information and Counterfeits categories.
4. Vendor ratings do have an impact on their lifespans.
5. Vendor ratings only have an impact on bitcoin earnings for the top earning vendors.
6. There are 23 vendors that have migrated from Silk Road 2 (in 2014) to the current 6 DNMs (in 2019).

With these insights, it is hoped law enforcement efforts can be focused on the appropriate DNMs according to their needs. For instance, if there is a need to target Drugs, the Cannazon site can be closely monitored. The top vendors for Drugs can also be focused on and new vendors can be identified, by analyzing their review ratings.

Furthermore, by understanding the overall trends of the DRMs, it can be ascertained when are the peak periods of traffic. Hence, law enforcement efforts can increase their surveillance during these peak periods.

Lastly, law enforcement agencies can increase surveillance on vendors that are of particular interest. These vendors could be the top performing vendors or those with longest lifespans or those that had migrated over from other DNMs, like Silk Road 2.

# References

[1] Kilos dataset. Released in 2020-01-13. Available at https://www.gwern.net/DNM-archives#kilos

[2] Agora dataset. Phillip Hames. (2017-12-06). Dark Net Marketplace Data (Agora 2014-2015). Available at https://www.kaggle.com/philipjames11/dark-net-marketplace-drug-data-agora-20142015

[3] Silk Road 2 Listings. Danie P. (2020-09-11). Available at https://www.kaggle.com/dpet922/silk-road-2-listings

[4] Darknet Market Cocaine Listings. David Skipper Everling. (2017-11-23). Available at https://www.kaggle.com/everling/cocaine-listings/metadata