

# large\_data\_analysis1.ipynb の実行方法の手引き

2023.2.22

mdqu 日本語 (ja)

2023-0225

- 参加者
- バッジ
- コンピテンシ
- 評価
- 一般
- 参考にしたオンラインリソース
- 学生の状況をモニターする
- Zoom講義中に問いかけて反応を見る
- 自動採点の小テストで学んでもらう&理解度チェック
- 受講者にメッセージを送る

# Jupyter Notebook 利用事例ワークショップ

ダッシュボード / マイコース / 2023-0225

編集モードの開始

## アナウンスメント

学生から秘匿

熊本大学 教授システム学研究センター (RCiS) 喜多敏博

以下の各画像は、クリックすると (別タブ・別ウィンドウ

学生から秘匿

large\_data\_analysis1.ipynb

事前準備の画面です。

まず、このファイルをダウンロードします。

## 参考にしたオンラインリソース

- Excelの最大行数と最大列数について (バージョンによる違い) (Office Hack)
- Excel最大行 (100万行) の壁を軽々超える方法 (@IT)
- Mozilla Location Service] Downloads
- Ichnaea Data import and export

ドライブ

ドライブで検索

新規

マイドライブ

Colab Notebooks

パソコン

共有アイテム

最近使用したアイテム

スター付き

ゴミ箱

保存容量

15 GB 中 3.16 GB を使用

保存容量を購入

マイドライブ > Colab Notebooks

ファイル

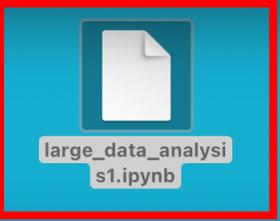
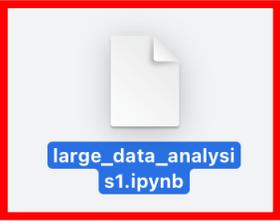
large\_data\_analys1.ipynb

ファイルをドロップして次のフォルダにアップロード

Colab Notebooks

ダウンロードした ipynb のファイルを以下の場所にアップロードします。

マイドライブ > Colab Notebooks



drive.google.com/drive/folders/17jZBNeOO\_e6OxAw5ObuAlwm\_bkPTuuRz

ドライブ

ドライブで検索

マイドライブ > Colab Notebooks

アップロード完了

Colab Notebooks

large\_data\_analysis1.ipynb

1 個のアップロード完了

large\_data\_analysis1.ipynb

保存容量

15 GB 中 3.16 GB を使用

保存容量を購入

large\_data\_analysis1.ipynb

- 2023-0225
- 参加者
- バッジ
- コンピテンシ
- 評価
- 一般
- 参考にしたオンラインリソース
- 学生の状況をモニターする
- Zoom講義中に問いかけて反応を見る
- 自動採点の小テストで

## 参考にしたオンラインリソース

- Excelの最大行数と最大列数について (バージョンによる違い) (Office Hack)
- Excel最大行 (100万行) の壁を軽々超える方法 (@IT)
- Mozilla Location Service] Downloads**
- Ichnaea Data import and export
- Jupyter Notebook (岡崎 直観)
- 図解! Jupyter Notebookを徹底解説! (インストール・使い方・起動・終了方法) (AI-interのPython3入門)
- Plotly.pyによるデータ可視化のすすめ (フューチャー技術ブログ)
- 【Python】スライス操作についてまとめ (Qlita, @tanuk1647)
- matplotlib で散布図 (Scatter plot) を描く (Python でデータサイエンス)
- 【Python】 相関関係のヒートマップを作る方法 (HTOMblog)
- 【folium】 地図のMarkerの色や形・アイコン・タイルを変える

次に、CSVファイル(実際は、CSV.gzファイル)をダウンロードします。

large\_data\_analysi  
s1.ipynb

To the extent possible under law, [Mozilla Corporation](#) has waived all copyright and related or neighboring rights to *MLS Cell Network Export Data*. This work is published from the United States.

## Cell Networks

We export both a daily snapshot of the entire cell network collection as well as hourly differential updates. File names contain either the word *diff* or *full* to specify their type. The full cell export is limited to those cell networks whose position has been confirmed in the last 12 months.

## Full Cell Exports

- [MLS-full-cell-export-2023-02-19T000000.csv.gz](#) (225361kB)
- [MLS-full-cell-export-2023-02-18T000000.csv.gz](#) (225177kB)
- [MLS-full-cell-export-2023-02-17T000000.csv.gz](#) (224862kB)

CSV.gz ファイルをダウンロードします。  
ここでは2023-02-19バージョンのファイル。

## Differential Cell Exports

- [MLS-diff-cell-export-2023-02-19T230000.csv.gz](#) (1598kB)
- [MLS-diff-cell-export-2023-02-19T220000.csv.gz](#) (1822kB)
- [MLS-diff-cell-export-2023-02-19T210000.csv.gz](#) (1612kB)
- [MLS-diff-cell-export-2023-02-18T110000.csv.gz](#) (1612kB)
- [MLS-diff-cell-export-2023-02-18T100000.csv.gz](#) (1518kB)
- [MLS-diff-cell-export-2023-02-18T090000.csv.gz](#) (1518kB)

https://d2koia3g127518.cloudfront.net/export/MLS-full-cell-export-2023-02-19T000000.csv.gz





drive.google.com/drive/my-drive

ドライブ

ドライブで検索

新規

マイドライブ

Colab Notebooks

パソコン

共有アイテム

最近使用したアイテム

スター付き

ゴミ箱

保存容量

15 GB 中 3.37 GB を使用

保存容量を購入

アップロード完了

1 個のアップロード完了

MLS-full-cell-export-2023-02-19T000000...

large\_data\_analysis1.ipynb

MLS-full-cell-export-2...0.csv.gz

ここから実行結果の画面です。

三角印▶をクリックして実行  
します。

```
# Google Colab で実行している場合、ユーザの Google Drive をマウント(ファイルが使えるように)する。
import os
import sys
moduleList = sys.modules
ENV_COLAB = False
if 'google.colab' in moduleList:
    ENV_COLAB = True
    # print("google_colab")
if ENV_COLAB:
    from google.colab import drive
    drive.mount('/content/drive')
    gdpath = "/content/drive/MyDrive/"
else:
    gdpath = ""
```

Mounted at /content/drive

```
[3] # Google Drive のマイドライブ直下にファイルをアップロードしてください。
# gz形式で圧縮されたファイルをアップロードした場合は、以下のようなコマンドで解凍できます。
# ! gunzip /content/drive/MyDrive/MLS-full-cell-export-2023-01-31T000000.csv.gz
! gunzip /content/drive/MyDrive/MLS-full-cell-export-2023-02-19T000000.csv.gz
```

1分47秒 完了時間: 9:05



```
# Google Colab で実行している場合、ユーザの Google Drive をマウント(ファイルが使えるように)する。  
import os  
import sys  
moduleList = sys.modules  
ENV_COLAB = False  
if 'google.colab' in moduleList:  
    ENV_COLAB = True  
    # print("google_colab")  
if ENV_COLAB:  
    from google.colab import drive, files  
    drive.mount('/content/drive')  
    gdrive_path = "/content/drive/My Drive/"  
else:  
    gdrive_path = ""
```

このノートブックに **Google ドライブ** のファイルへのアクセスを許可しますか？

このノートブックは Google ドライブ ファイルへのアクセスをリクエストしています。Google ドライブへのアクセスを許可すると、ノートブックで実行されたコードに対し、Google ドライブ内のファイルの変更を許可することになります。このアクセスを許可する前に、ノートブック コードをご確認ください。

スキップ **Google ドライブに接続**



ログイン - Google アカウント

accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?access\_type=offlin...

Google にログイン



## アカウントの選択

「Google Drive for desktop」に移動



 別のアカウントを使用

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を Google Drive for desktop と共有します。このアプリを使用する前に、Google Drive for desktop の [プライバシー ポリシー](#) と [利用規約](#) をご確認ください。

Downloads | マイドライブ - Google ド | large\_data\_analysis1.ipynb | gzip --- gzip ファイルのセ

EHDKi0#scrollTo=kRpf\_3Pb-uxs

コメント 共有 設定 K

RAM ディスク

↑ ↓ ↻ 設定 印刷 削除

ファイルが使えるようにする。

```
実行 (28 秒) Cell > mount() > _mount() > blocking_request() > read_reply_from_input()
```



ログイン - Google アカウント

accounts.google.com/signin/oauth/consent?authuser=0&part=AJi8hANWeP0xX6wv8n5Yw6XqPQqLjP...

Google にログイン



## Google Drive for desktop が Google アカウントへのアクセスをリクエストしています



Google Drive for desktop に以下を許可します:

-  Google ドライブのすべてのファイルの表示、編集、作成、削除 
-  Google フォトの写真、動画、アルバムの表示 
-  モバイル クライアントの設定とウェブテストの取得 
-  プロフィールや連絡先などの Google ユーザー情報の表示 
-  Google ドライブ内のファイルのアクティビティ履歴の表示 

マイドライブ - Google ドライブ

large\_data\_analysis1.ipynb

gzip --- gzip ファイルのセ

コメント 共有 設定 

RAM ディスク

↑ ↓ ↻ 🗨 ⚙ 📄 🗑 ⋮

```
king_request() > read_reply_from_input()
```

large\_data\_analysis1.ipynb

MLS-full-cell-export-2...0.csv.gz



Chrome ファイル 編集 表示 履歴 ブックマーク プロファイル タブ ウィンドウ ヘルプ

ログイン - Google アカウント

accounts.google.com/signin/oauth/consent?authuser=0&part=AJi8hANWeP0xX6wv8n5Yw6XqPQqLjP...

- モバイルクライアントの設定とウェブテストの取得 ⓘ
- プロフィールや連絡先などの Google ユーザー情報の表示 ⓘ
- Google ドライブ内のファイルのアクティビティ履歴の表示 ⓘ
- Google ドライブのドキュメントの表示、編集、作成、削除 ⓘ

**Google Drive for desktop** を信頼できることを確認

お客様の機密情報をこのサイトやアプリと共有することがあります。アクセス権の確認、削除は、[Google アカウント](#) でいつでも行えます。

Google で [データ共有を安全に行う方法](#) についての説明をご覧ください。

Google Drive for desktop の [プライバシーポリシー](#) と [利用規約](#) をご覧ください。

キャンセル **許可**

日本語 ヘルプ プライバシー 規約

マイドライブ - Google ドライブ x large\_data\_analysis1.ipynb x gzip --- gzip ファイルのセ x +

コメント 共有 設定 K

[設定] でブラウザの通知を有効にすると、コードセルの実行が完了したときにアラートを受け取ることができます

OK スキップ

```
locking_request() > read_reply_from_input()
```

large\_data\_analysis1.ipynb MLS-full-cell-export-2...0.csv.gz

large\_data\_analysis1.ipynb

```

ENV_COLAB = True
[5] # print("google_colab")
if ENV_COLAB:
    from google.colab import drive
    drive.mount('/content/drive')
    gopath = "/content/drive/MyDrive/"
else:
    gopath = ""
  
```

Mounted at /content/drive

```

# Google Drive のマイドライブ直下にファイルをアップロードしてください。
# gz形式で圧縮されたファイルをアップロードした場合は、以下のようなコマンドで解凍できます。
# ! gunzip /content/drive/MyDrive/MLS-full-cell-export-2023-01-31T000000.csv.gz
! gunzip /content/drive/MyDrive/MLS-full-cell-export-2023-02-19T000000.csv.gz
  
```

```

# マイドライブ直下のcsvファイルを読み込んで、DataFrame形式にする(数十秒~1分間かかる場合も)
import pandas as pd
df = pd.read_csv(gopath + 'MLS-full-cell-export-2023-01-31T000000.csv')
df

[ ] stat_val1 = df.describe(include='all')
  
```

16秒 完了時間: 9:07

この例では、2023-02-19の日付の csv.gz ファイルを解凍しています。

適宜、ご自分がダウンロードしたファイル名に書き換えて実行します。

# Google Drive のマイドライブ直下にファイルをアップロードしてください。  
 # gz形式で圧縮されたファイルをアップロードした場合は、以下のようなコマンドで解凍できます。  
 # ! gunzip /content/drive/MyDrive/MLS-full-cell-export-2023-01-31T000000.csv.gz  
 ! gunzip /content/drive/MyDrive/MLS-full-cell-export-2023-02-19T000000.csv.gz



drive.google.com/drive/my-drive

ドライブ

ドライブで検索

マイドライブ

新規

マイドライブ

Colab Notebooks

パソコン

共有アイテム

最近使用したアイテム

スター付き

ゴミ箱

保存容量

15 GB 中 3.37 GB を使用

保存容量を購入

名前	オーナー	最終更新	ファイルサイズ
Colab Notebooks	自分	2022/09/27 自分	-
MLS-full-cell-export-2023-02-19T000000.csv	自分	8:45 自分	704.3 MB

CSVファイルに解凍されます。

large\_data\_analysis1.ipynb

MLS-full-cell-export-2...0.csv.gz

ここからは、順次、実行した結果の画面になります。

```
# マイドライブ直下のcsvファイルを読み込んで、DataFrame形式にする (数十秒~1分間かかる場合も)
import pandas as pd
df = pd.read_csv(gdpath + 'MLS-full-cell-export-2023-02-19T000000.csv')
df
```

	radio	mcc	net	area	cell	unit	lon	lat	range	samples	changeable	created	updated	averageSignal
0	GSM	202	0	25	12852	NaN	23.800216	37.994988	0	1	1	1657960739	1657960739	NaN
1	GSM	202	0	27	18263	NaN	23.791401	38.054497	57	10	1	1659027120	1659027313	NaN
2	GSM	202	0	27	53076	NaN	23.771893	38.020783	251	22	1	1673005956	1673006345	NaN
3	GSM	202	0	38	5942	NaN	23.730197	37.967260	5	2	1	1664896047	1664896069	NaN
4	GSM	202	0	38	27851	NaN	23.740069	37.957128	13	3	1	1664893942	1664894123	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9004717	LTE	748	10	29240	1314310	455.0	-122.398675	47.585079	0	20	1	1580020567	1648509556	NaN
9004718	LTE	748	10	29400	1298444	NaN	-56.142933	-34.866093	2	9	1	1668718930	1676672461	NaN
9004719	LTE	748	10	29400	5129220	NaN	-56.142931	-34.866092	3	12	1	1651361767	1675304555	NaN
9004720	LTE	748	10	29730	683523	NaN	-57.851620	-34.470681	0	2	1	1638065422	1652998809	NaN
9004721	LTE	748	10	29730	1294595	NaN	-53.454918	-33.690997	0	1	1	1672764366	1672764366	NaN

9004722 rows x 14 columns

```
[ ] stat_val1 = df.describe(include='all')
stat_val1
```

✓ 17 秒 完了時間: 9:11

```
stat_val1 = df.describe(include='all')
stat_val1
```

	radio	mcc	net	area	cell	unit	lon	lat	range	samples	changeable
<b>count</b>	9004722	9.004722e+06	9.004722e+06	9.004722e+06	9.004722e+06	2.445510e+06	9.004722e+06	9.004722e+06	9.004722e+06	9.004722e+06	9004722.0
<b>unique</b>	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
<b>top</b>	GSM	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
<b>freq</b>	3509248	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
<b>mean</b>	NaN	4.080228e+02	7.253158e+01	1.898268e+04	4.151380e+07	2.280386e+02	-1.213643e+01	2.522748e+01	4.796682e+03	1.403937e+03	1.0
<b>std</b>	NaN	1.770020e+02	1.501781e+02	1.690214e+04	6.316237e+07	1.437231e+02	7.688108e+01	2.556888e+01	1.090369e+04	1.993935e+04	0.0
<b>min</b>	NaN	2.020000e+02	0.000000e+00	1.000000e+00	1.000000e+00	0.000000e+00	-1.753443e+02	-8.215183e+01	0.000000e+00	1.000000e+00	1.0
<b>25%</b>	NaN	2.600000e+02	2.000000e+00	4.003000e+03	2.946200e+04	1.020000e+02	-8.358330e+01	1.084153e+01	1.660000e+02	6.000000e+00	1.0
<b>50%</b>	NaN	3.340000e+02	1.000000e+01	1.407000e+04	8.970770e+06	2.220000e+02	-4.693037e-01	3.289076e+01	9.610000e+02	3.500000e+01	1.0
<b>75%</b>	NaN	5.200000e+02	4.000000e+01	3.080000e+04	5.709570e+07	3.460000e+02	3.437945e+01	4.577043e+01	4.030000e+03	2.470000e+02	1.0
<b>max</b>	NaN	7.500000e+02	9.990000e+02	6.553300e+04	2.684355e+08	5.110000e+02	1.785150e+02	8.287881e+01	1.802190e+05	1.748846e+07	1.0

```
# 値を取り出す
stat_val1.loc['mean']['mcc']
```

6秒 完了時間: 9:15

```
[9] # 値を取り出す
stat_val1.loc['mean']['mcc']
```

408.02278382386487

```
[10] # 最頻値
df_mode = df.mode()
df_mode
```

	radio	mcc	net	area	cell	unit	lon	lat	range	samples	changeable	created	updated	averageSignal
0	GSM	334	1	101	65535	0.0	39.99083	43.403799	0	1	1	1676353618	1676630758	NaN

```
[11] # 値を取り出す
df_mode.loc[0]['lon']
```

39.99083

```
[12] # 中央値
df_median = df.median()
df_median
```

<ipython-input-12-5066631b7aaa>:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

```
df_median = df.median()
mcc      3.340000e+02
net      1.000000e+01
area     1.407000e+04
cell     8.970770e+06
unit     2.220000e+02
lon     -4.693037e-01
lat      3.289076e+01
range    9.610000e+02
samples  3.500000e+01
changeable 1.000000e+00
created  1.653812e+09
updated  1.669774e+09
averageSignal NaN
dtype: float64
```

```
[13] # 値を取り出す
df_median['lon']
```

-0.4693037

各列同士がどれくらい関係があるか、相関行列を計算してみます。

```
# 相関行列を計算する
corr_matrix = df.corr()
corr_matrix
```

	mcc	net	area	cell	unit	lon	lat	range	samples	changeable	created	updated	averageSignal
mcc	1.000000	-0.163985	-0.113574	-0.057634	-0.033202	-0.024885	-0.840090	0.017364	0.018980	NaN	0.089940	0.153629	NaN
net	-0.163985	1.000000	0.148572	0.165785	-0.009199	-0.367975	0.170321	-0.015955	-0.016951	NaN	0.073363	-0.038294	NaN
area	-0.113574	0.148572	1.000000	0.161958	-0.010102	-0.026101	0.048701	-0.016897	-0.023962	NaN	0.069568	-0.107968	NaN
cell	-0.057634	0.165785	0.161958	1.000000	-0.012525	-0.004761	0.028700	-0.072359	-0.028481	NaN	0.086308	-0.136407	NaN
unit	-0.033202	-0.009199	-0.010102	-0.012525	1.000000	-0.025226	0.025431	-0.015891	-0.005256	NaN	0.009263	-0.019226	NaN
lon	-0.024885	-0.367975	-0.026101	-0.004761	-0.025226	1.000000	0.087587	-0.038818	0.000239	NaN	-0.118712	-0.173809	NaN
lat	-0.840090	0.170321	0.048701	0.028700	0.025431	0.087587	1.000000	-0.032327	-0.012895	NaN	-0.107195	-0.178366	NaN
range	0.017364	-0.015955	-0.016897	-0.072359	-0.015891	-0.038818	-0.032327	1.000000	0.067035	NaN	-0.259826	0.099469	NaN
samples	0.018980	-0.016951	-0.023962	-0.028481	-0.005256	0.000239	-0.012895	0.067035	1.000000	NaN	-0.043391	0.055697	NaN
changeable	NaN	NaN	NaN	NaN									
created	0.089940	0.073363	0.069568	0.086308	0.009263	-0.118712	-0.107195	-0.259826	-0.043391	NaN	1.000000	0.015905	NaN
updated	0.153629	-0.038294	-0.107968	-0.136407	-0.019226	-0.173809	-0.178366	0.099469	0.055697	NaN	0.015905	1.000000	NaN
averageSignal	NaN	NaN	NaN	NaN									

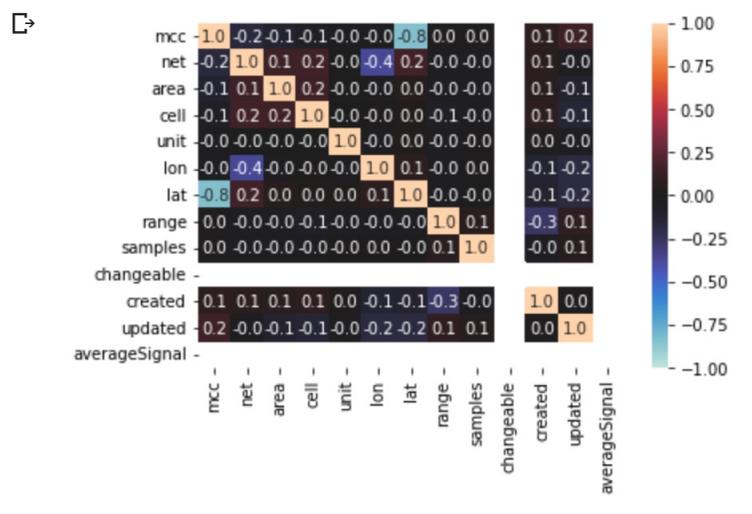
6秒 完了時間: 9:25

```
# 相関行列のヒートマップ表示
df.corr().style.background_gradient(cmap = "bwr", vmin = -1, vmax = 1)
```

	mcc	net	area	cell	unit	lon	lat	range	samples	changeable	created	updated	averageSignal
mcc	1.000000	-0.163985	-0.113574	-0.057634	-0.033202	-0.024885	-0.840090	0.017364	0.018980	nan	0.089940	0.153629	nan
net	-0.163985	1.000000	0.148572	0.165785	-0.009199	-0.367975	0.170321	-0.015955	-0.016951	nan	0.073363	-0.038294	nan
area	-0.113574	0.148572	1.000000	0.161958	-0.010102	-0.026101	0.048701	-0.016897	-0.023962	nan	0.069568	-0.107968	nan
cell	-0.057634	0.165785	0.161958	1.000000	-0.012525	-0.004761	0.028700	-0.072359	-0.028481	nan	0.086308	-0.136407	nan
unit	-0.033202	-0.009199	-0.010102	-0.012525	1.000000	-0.025226	0.025431	-0.015891	-0.005256	nan	0.009263	-0.019226	nan
lon	-0.024885	-0.367975	-0.026101	-0.004761	-0.025226	1.000000	0.087587	-0.038818	0.000239	nan	-0.118712	-0.173809	nan
lat	-0.840090	0.170321	0.048701	0.028700	0.025431	0.087587	1.000000	-0.032327	-0.012895	nan	-0.107195	-0.178366	nan
range	0.017364	-0.015955	-0.016897	-0.072359	-0.015891	-0.038818	-0.032327	1.000000	0.067035	nan	-0.259826	0.099469	nan
samples	0.018980	-0.016951	-0.023962	-0.028481	-0.005256	0.000239	-0.012895	0.067035	1.000000	nan	-0.043391	0.055697	nan
changeable	nan	nan	nan	nan									
created	0.089940	0.073363	0.069568	0.086308	0.009263	-0.118712	-0.107195	-0.259826	-0.043391	nan	1.000000	0.015905	nan
updated	0.153629	-0.038294	-0.107968	-0.136407	-0.019226	-0.173809	-0.178366	0.099469	0.055697	nan	0.015905	1.000000	nan
averageSignal	nan	nan	nan	nan									

```
[ ] # 相関行列のヒートマップ表示: seaborn を使う場合
import seaborn
import matplotlib.pyplot as plt
```

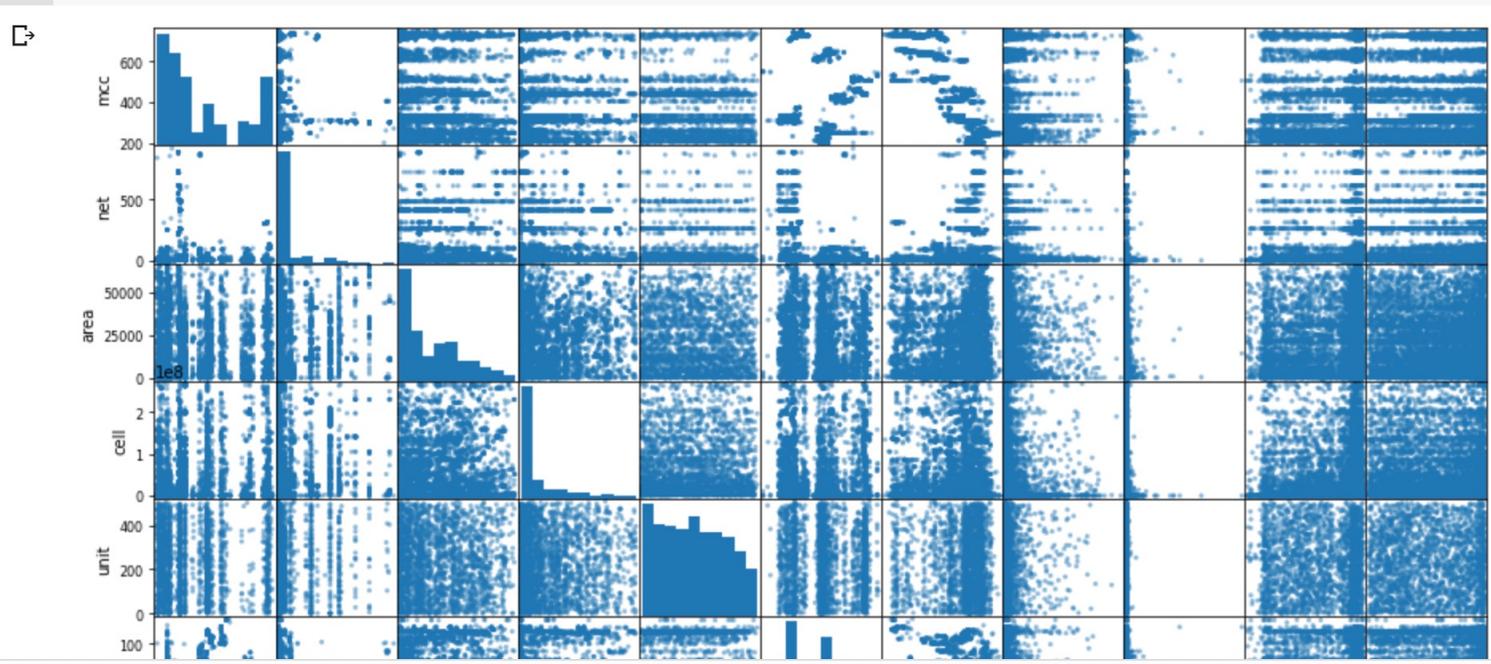
```
# 相関行列のヒートマップ表示: seaborn を使う場合
import seaborn
import matplotlib.pyplot as plt
seaborn.heatmap(corr_matrix,
                 vmin=-1.0, vmax=1.0, center=0,
                 annot=True, fmt='.1f',
                 xticklabels=corr_matrix.columns.values,
                 yticklabels=corr_matrix.columns.values
                 )
plt.show()
```



```
[ ] df2 = df.copy() # df2 = df と書くと複製にならな(参照渡しになってしまう)
# いくつかの列を削除する
```

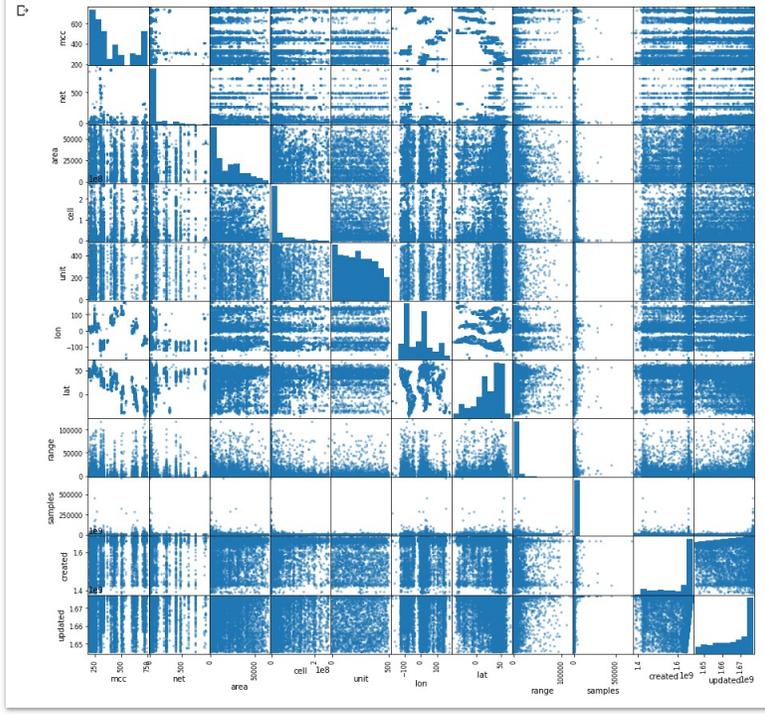
1秒 完了時間: 9:26

```
df2 = df.copy() # df2 = df と書くと複製にならな(参照渡しになってしまう)
# いくつかの列を削除する
df2.drop('changeable', axis=1, inplace=True)
df2.drop('averageSignal', axis=1, inplace=True)
# df2 = df.loc[:,['mcc','cell','lon','lat']]
df_sub = df2[::1000] # 1/1000 で間引いたスライス
pd.plotting.scatter_matrix(df_sub, figsize=(15, 15))
plt.show()
```



✓ 13秒 完了時間: 9:27

```
df2 = df.copy() # df2 = df と書くことで複製になる(参照渡しにならない)
# いくつかの列を削除する
df2.drop("changeable", axis=1, inplace=True)
df2.drop("averageSignal", axis=1, inplace=True)
# df2 = df.loc[:,['mcc','cell','lon','lat']]
df_sub = df2[::1000] # 1/1000 で間引いたスライス
pd.plotting.scatter_matrix(df_sub, figsize=(15, 15))
plt.show()
```



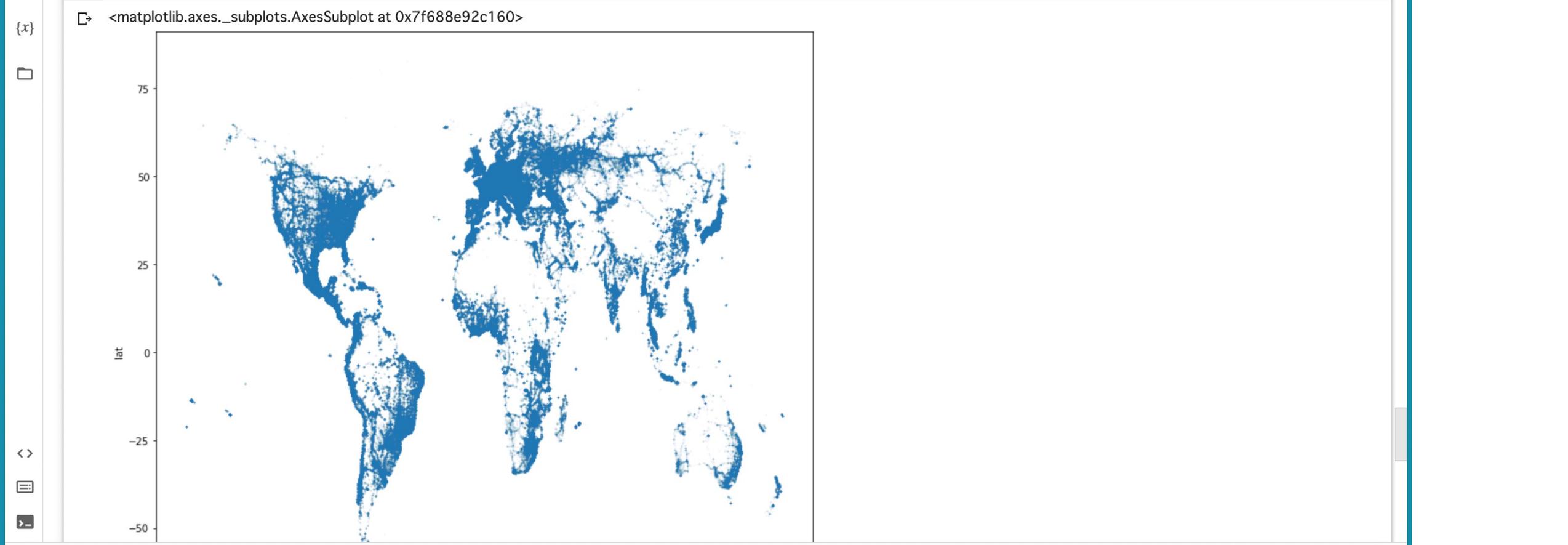
```
[ ] df3 = df.loc[:,['lon','lat']]
df3.plot.scatter(x=0, y=1, s=1, alpha=0.02, figsize=(12, 12))

[ ] df4 = df.loc[:,['mcc','lat']]
df4.plot.scatter(x=0, y=1, s=3, alpha=0.5, figsize=(12, 12))

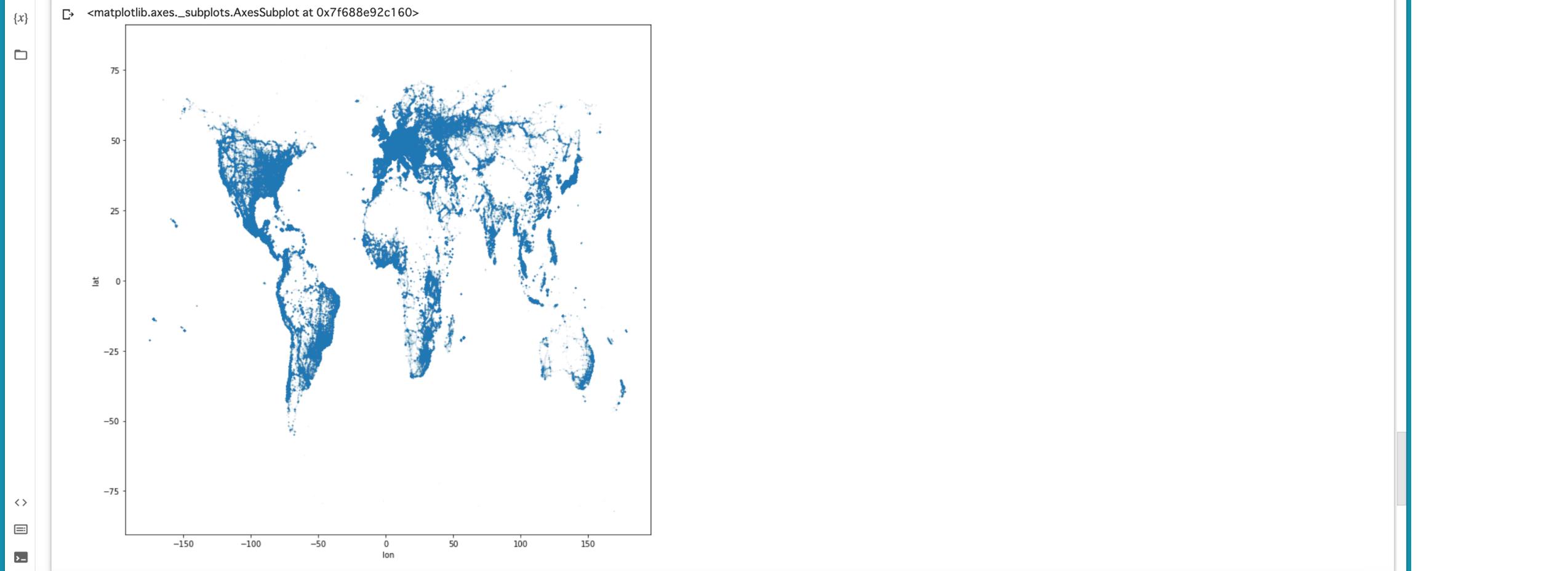
[ ] import folium
map = folium.Map(location=[df['lat'].mean(), df['lon'].mean()], zoom_start=3)

df_sub2 = df[::10000] # 1/10000 に間引く
# データフレームの各行に対して地図上のマーカーを追加する
```

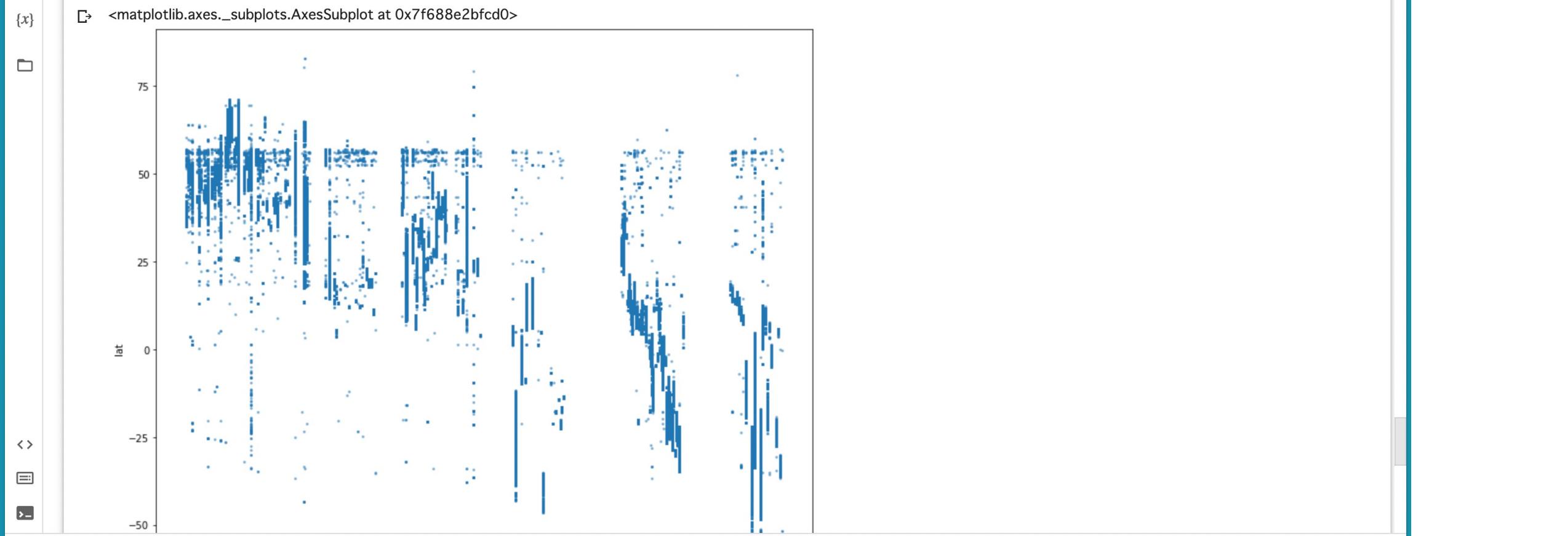
```
df3 = df.loc[:,['lon','lat']]  
df3.plot.scatter(x=0, y=1, s=1, alpha=0.02, figsize=(12, 12))
```



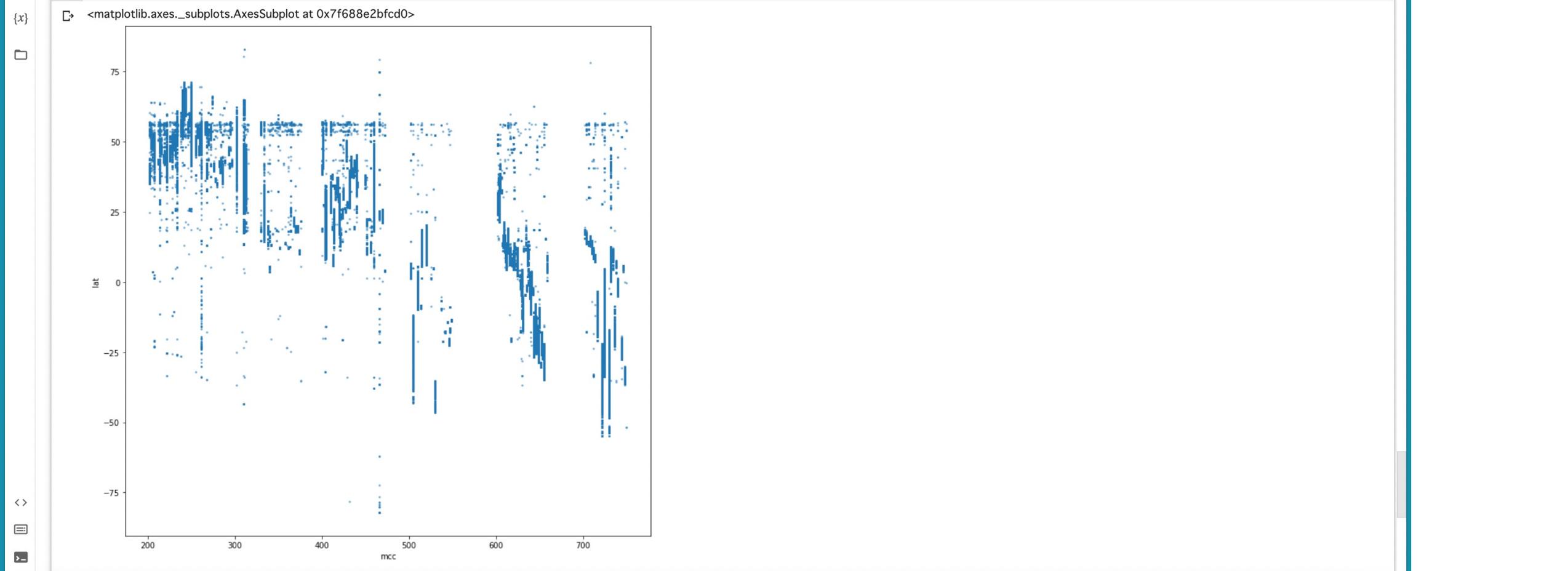
```
df3 = df.loc[:,['lon','lat']]  
df3.plot.scatter(x=0, y=1, s=1, alpha=0.02, figsize=(12, 12))
```



```
df4 = df.loc[:,['mcc','lat']]  
df4.plot.scatter(x=0, y=1, s=3, alpha=0.5, figsize=(12, 12))
```



```
df4 = df.loc[:,['mcc','lat']]  
df4.plot.scatter(x=0, y=1, s=3, alpha=0.5, figsize=(12, 12))
```





```
+ コード + テキスト
import folium
map = folium.Map(location=[df['lat'].mean(), df['lon'].mean()], zoom_start=3)

df_sub2 = df[:10000] # 1/10000 に間引く
# データフレームの各行に対して地図上のマーカーを追加する
for index, row in df_sub2.iterrows():
    folium.Marker([row['lat'], row['lon']], popup = str(row['mcc'])+' '+str(row['area']).add_to(map)

map
```

