

# INF206

Programming

プログラミング

Week 10 | November 30, 2022



## Today:

- Working with data in Python
- Self learning: Python objects and libraries
- One-on-one with me to discuss your project

# Data in Python

まずはデータを探す 🔍

- formatはcsvかxlsがいいでしょう

統計データを探す

統計データの活用

統計データの高度利用

統計関連情報

リンク集

●統計データを探す (政府統計の調査結果を探します)

その他の絞込

利用ガイド

すべて

政府統計一覧の中から探します

分野

17の統計分野から探します

組織

統計を作成した府省等から探します

キーワード検索: 例: 国勢調査

検索

●統計データを活用する

グラフ

主要指標をグラフで表示  
(統計ダッシュボード)

時系列表

主要指標を時系列表で表示  
(統計ダッシュボード)

地図

地図上に統計データを表示

地域

都道府県、市区町村の  
主要データを表示

●統計データの高度利用

マイクロデータの利用  
公的統計のマイクロデータの利用案内

開発者向け

API、LODで統計データを取得

●統計関連情報

統計分類・調査計画等

<https://www.e-stat.go.jp/>

Programming | プログラミング

→ 一覧からデータを探す

→ データ活用事例を見る

→ データで見る東京

→ データをリクエストする

→ データを活用された方へ

# TOKYO OPEN DATA

東京都 オープンデータカタログサイト

公開中のデータを検索する

データを探す

オープンデータは様々な用途で活用することができます。

- 都や区市町村に関する様々な情報を調べる
- 都政に関するデータを確認する
- アプリやサービス開発に活用する

都知事杯オープンデータ・ハッカソン

東京都オープンデータ・ラウンドテーブル

東京都コロナウイルス関連オープンデータ

# 【2022最新】データ分析に必要なオープンデータソース70選！

2022年10月07日

---

質のいいデータソースはデータ分析とデータ活用に大きく役割を果たしています。ビッグデータの収集は大変な仕事だと思っている方は多いかもしれませんが、実はそうでもありません。本記事では政府機関、医療・健康、金融・経済、ビジネス・企業、ジャーナリズム・メディア、不動産・観光、文化・教育など、カテゴリ別に誰でも利用できる70選のオープンデータソースを整理いたしました。ぜひデータ分析にお役に立ててください。

<https://www.octoparse.jp/blog/70-amazing-free-data-sources-you-should-know/#>



1. データをダウンロード・作成
2. Excelでクリーンアップ
3. CSV (UTF8)としてセーブ
4. Google Colab Driveにアップロード
5. Pythonで導入

# Example



警視庁

[安全な暮らし](#)

[交通安全](#)

[相談・お悩み](#)

[手続き](#)

[事件・事故](#)

[警視庁について](#)

[トップページ](#) → [警視庁について](#) → [発生状況・統計](#) → [犯罪発生状況](#) → 犯罪発生情報（年計）

## 犯罪発生情報（年計）

更新日：2022年6月23日

### 年計 2021年（令和3年）

- [令和3年 ひったくり（CSVファイル：6KB）](#)
- [令和3年 車上ねらい（CSVファイル：190KB）](#)
- [令和3年 部品ねらい（CSVファイル：165KB）](#)
- [令和3年 自動販売機ねらい（CSVファイル：15KB）](#)
- [令和3年 自動車盗（CSVファイル：24KB）](#)
- [令和3年 オートバイ盗（CSVファイル：105KB）](#)
- [令和3年 自転車盗（CSVファイル：2,462KB）](#)

### 犯罪発生状況

[東京の犯罪（令和3年版）](#)

[区市町村の町丁別、罪種別  
及び手口別認知件数](#)

[犯罪発生情報（年計）](#)

このページを見ている人はこ  
んなページも見ています

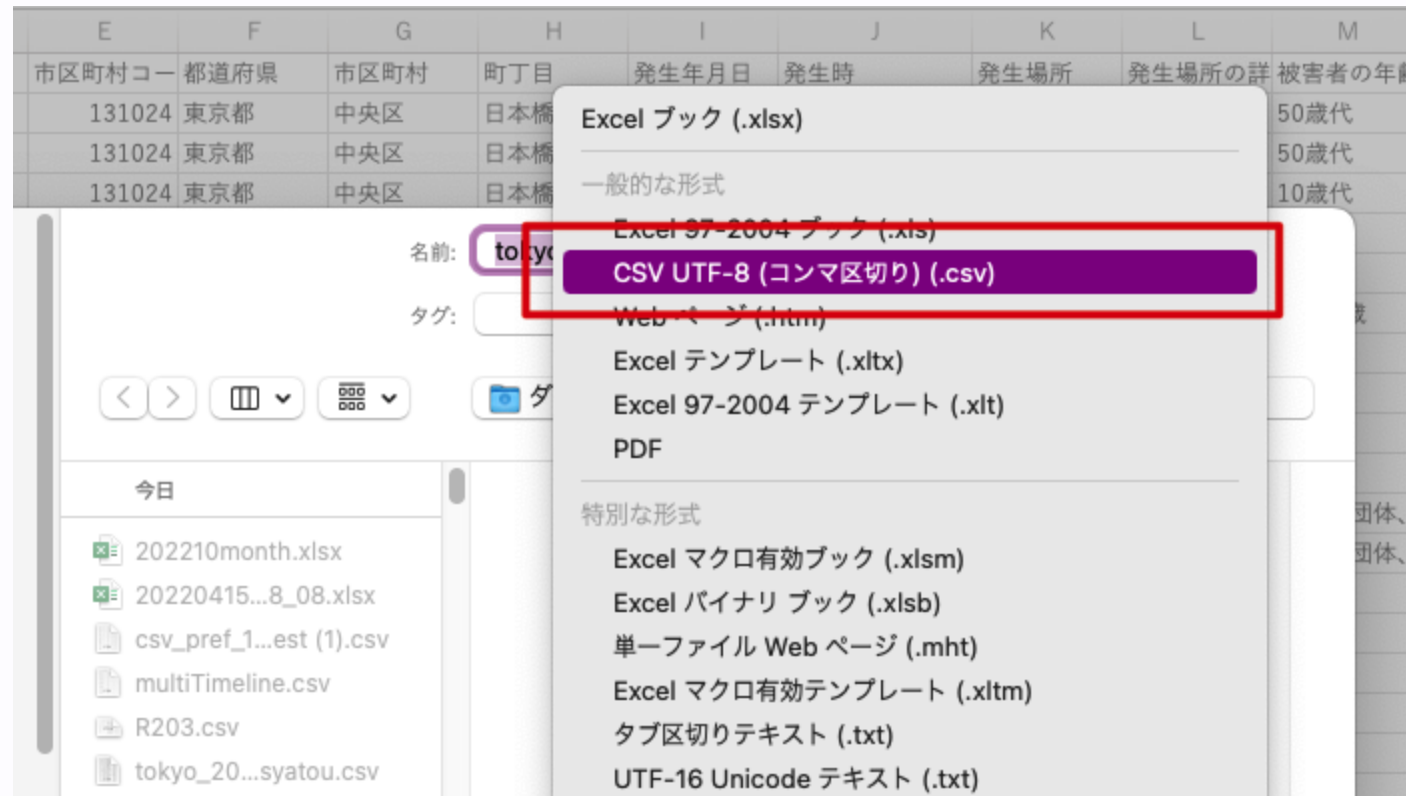
[東京の犯罪（令和3年版）](#)

[https://www.keishicho.metro.tokyo.lg.jp/about\\_mpd/jokyo\\_tokei/jokyo/hanzaihasseijyo](https://www.keishicho.metro.tokyo.lg.jp/about_mpd/jokyo_tokei/jokyo/hanzaihasseijyo)

## Excel check

- カッコ、特別文字（・とか）を取り除く

# UTF-8で保存



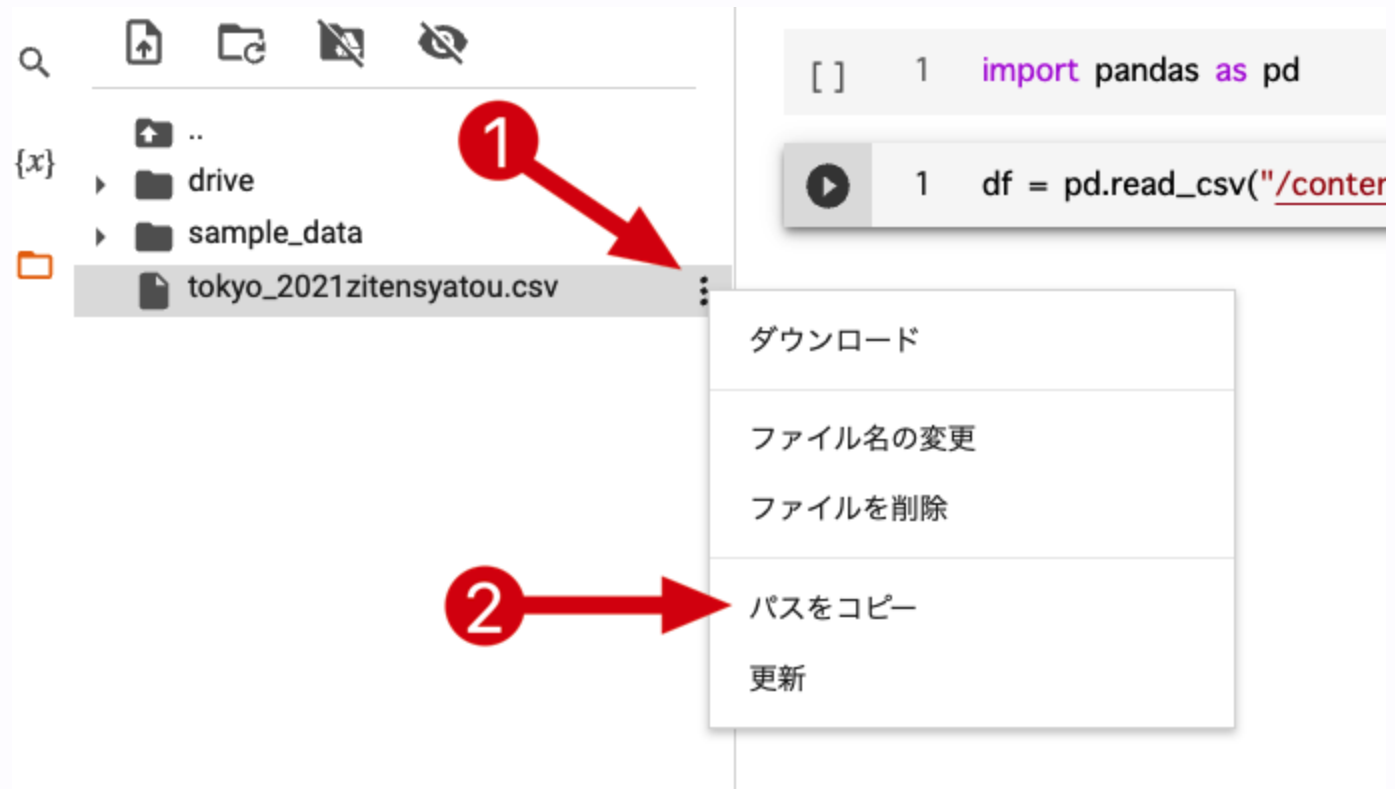
# Google Colabにアップロード



ではデータをPythonでインポート  
pandasを導入

```
import pandas as pd
```

## データのpathをコピー



```
df = pd.read_csv("/content/tokyo_2021zitenasyatou.csv")
```

pathをダブルクォーテーションの中に入れる



# データを分析

```
df.info()
```

✓  
0  
秒

```
[31] 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 64 entries, 0 to 63  
Data columns (total 14 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0  罪名         64 non-null    object  
1  手口         64 non-null    object  
2  管轄警察署   64 non-null    object  
3  管轄交番・駐在所 61 non-null    object  
4  市区町村コード 63 non-null    float64  
5  都道府県     64 non-null    object  
6  市区町村     64 non-null    object  
7  町丁目       64 non-null    object  
8  発生年月日   64 non-null    object  
9  発生時       64 non-null    int64  
10 発生場所     64 non-null    object  
11 被害者の性別  64 non-null    object  
12 被害者の年齢  64 non-null    object  
13 現金被害の有無 64 non-null    object  
dtypes: float64(1), int64(1), object(12)  
memory usage: 7.1+ KB
```

```
df.describe(include="all")
```

```
1 df.describe(include="all")
```



	罪名	手口	管轄 警察 署	管轄 交 番・ 駐在 所	市区町村コード	都道 府県	市区 町村	町丁 目	発生年月日	発生
count	64	64	64	61	63.000000	64	64	64	64	64.000000
unique	1	1	39	54	NaN	1	27	61	57	NaN
top	窃盗	ひっ たく り	新宿	歌舞 伎町 交番	NaN	東京 都	足立 区	歌舞 伎町 1丁 目	2021/12/12	NaN
freq	64	64	10	4	NaN	64	10	2	2	NaN
mean	NaN	NaN	NaN	NaN	131309.936508	NaN	NaN	NaN	NaN	12.64062

## plotly.expressでデータを可視化

```
import plotly.express as px
```

データって色んな表現がある。  
例えば...

```
df = df.sort_values(by="被害者の年齢")
fig = px.bar(df, y="被害者の年齢", color="被害者の性別", orientation="h")
fig.show()
```

