

Pythonでデータ処理入門 【データをグラフ化!】

アジェンダ

① あいさつ

② Pandas解説

③ データの確認

④ データから情報の抽出

⑤ 列の追加

⑥ データの並び替え

⑦ データの集計

⑧ データのグループ化

⑨ 演習問題

⑩ matplotlib解説

⑪ データからグラフを作成

⑫ 演習問題

自己紹介



倉崎敦至

学生時代WEBのベンチャーでフロントエンジニア
現在はSE(システムエンジニア)
趣味でアプリを開発中



田中良幸

学生時代PythonやJavaで機械学習の研究
現在は同じくSE
好きな漢字は「飛」

Pandasとは

データ解析を支援するPythonライブラリ

データの整形、集計、可視化がこれ一つでできる

機械学習の前ステップの「データの前処理」で特に活躍

構造体データを扱う

Pythonだけでデータ処理を行うのと比較して、非常に高速処理



機械学習の流れ



DataFrameとSeries

DataFrame

行と列で構成される二次元のデータ構造

Series

1つの列を表す一次元のデータ構造

※DataFrameはSeriesの集まり

ハンズオン



データの読み込み

```
import pandas as pd
```

```
df = pd.read_csv('sample.csv')
```

オプション

- `names` : カラム名<list>
- `usecols` : 読み込む列<list>
- `skiprows` : 読み込まない行<int><list>
- `encoding` : エンコード<str>
- `parse_dates` : `datetime`で読み込むカラム<list>
- `index_col` : インデックス指定<str><list>
- `header` : ヘッダーの処理<None><int>
- `dtype` : データの型変更<str>

データの抽出

列の抽出

df[“カラム名”]

df.カラム名

行の抽出

df[1:3]

df[3:]

df[df[“カラム名”] > 0]

1行目から**3**行目までを抽出

3行目より後ろを抽出

列の値が **0** より大きい行を取得

データの抽出

df.loc[[行ラベル],[列ラベル]]

df.iloc[[行番号],[列番号]]

【使用例】

df.iloc[[1],[2]]

1行目の**2**列目

df.loc[:,["A","B"]]

A列と**B**列を抽出

df.loc[df["score"] >=20, ["A"]]

A列の**score**が**20**以上の行

列の追加

`df["A"] = 0`

A列を追加し全ての要素に0を入れる

削除は`del df['A']`

`df["A"] = [1, 2, 3, 4, 5]`

A列を追加し配列で指定した値を入れる

レコード数と配列の要素数が違うとエラーになる

行の追加

`df.loc['A'] = 0`

インデックスがAの行に全ての要素が0の行を追加。

削除は `df.drop([3,4])`

`df.loc['A'] = ["B", "C", 20]`

インデックスがAの行に要素の値を指定して追加、要素数が違うとエラーになる。

行の追加

df = df.append(df2)

append関数を使ってDataFrameに別のDataFrameを追加

Seriesも追加することができる

※indexの振り直し

df.reset_index(drop=True)

データのソート

df2 = df.sort_values(“列名”)

引数に配列を渡すことで複数の列でソート可能

インデックスでソートするにはdf.sort_index()

オプション

- ascending：昇順(デフォルト)と降順切り替え<bool>
- inplace：元のdfを変更するか新しいdfをつくるか<bool>

データの集計

- **df.describe()**

最小値、最大値、平均、中央値などの基本情報

- **df["A"].value_counts()**

値の数のカウント

データ集計

- `df[“A”].sum()`

合計

- `df[“A”].mean()`

平均 skipnaオプションで欠損値を無視

- `df[“A”].max()`

最大値

- `df[“A”].var()`

分散

- `df[“A”].std()`

標準偏差

値のグループ化

df.groupby(“カラム名”)

特定の列の値ごとにグループを作成

【使い方例】

df.groupby(“カラム名”).sum()

グループごとの合計

df.groupby(“カラム名”).mean()

グループごとの平均

欠損値(NaN)処理

df.isnull().sum()

欠損値の確認

df.dropna(how='any')

欠損値を含んだ行を削除

df.fillna(0)

欠損値を置き換える

その他

- **df.dtypes**

DataFrameの値の型を確認

- **a[“A”].astype(float)**

A列の型を変更

- **df.T**

行列入れ替え

演習

matplotlibとは

グラフ描画のPythonライブラリ

「import matplotlib.pyplot as plt」でインポート

Pandasから呼びだして使う事もできる

折れ線，棒グラフやヒストグラムなどさまざまなグラフが作成できる

jpg, bmp, epsなどさまざまな形式で保存できる



グラフの描画

df.plot()

plt.show()

plt.savefig(“fig.jpg”)

グラフ作成

グラフの描画

グラフの保存

plot()のオプション

- x：x軸に用いるデータ<int><str>
- title：グラフタイトル<str>
- legend：凡例を表示するか<bool>
- y：y軸に用いるデータ<int><str><list>
- kind：グラフの種類<str>：“bar”, “hist”
- figsize：グラフのサイズ(インチ)<tuple>

複数のグラフ描画その1

df.plot(subplots = True)

オプション

- **sharex** : x軸をそろえる<bool>
- **sharey** : y軸をそろえる<bool>
- **layout** : グラフの配置を決める<tuple>

複数のグラフ描画その2

fig, axes = plt.subplots() グラフの配置決定
df.plot(ax=ax) 各グラフの作成

ax : matplotlibのオブジェクト、axes : axのリスト

subplotsのオプション

- nrows : 縦にいくつ並べるか<int>
- sharex : x軸をそろえる<bool>/“row”/“col”
- ncols : 横にいくつ並べるか<int>
- sharey : y軸をそろえる<bool>/“row”/“col”

複数のグラフ描画その2

```
fig, axes = plt.subplots(nrows=2, ncols=2)
```

縦2, 横2の形に4つグラフを配置する

```
df.plot(ax=axes[0][0], x='A', y='B')
```

1つ目のグラフ(左上)について描画する

```
df.plot(ax=axes[0][1], x='A', y='C')
```

2つ目のグラフ(右上)について描画する

...

グラフの編集 (df.plot()編)

紹介していなかった**df.plot()**の引数で編集できるもの

- xlim : x軸の範囲<tuple>
- ylim : y軸の範囲<tuple>
- label : 凡例の名前<str><list>
- grid : グリッド線<bool>

pyplot.plot()の引数で編集できるものの一例 : **df.plot()**でも使用できる

- color : グラフの色<str><list>
- alpha : グラフの透過<float>
- marker : グラフのマーク<str> : “.”, “+”
- ls : 線の種類<str> : “-”, “--”, “:”

グラフの編集 (ax編)

- `ax.set_xlabel("ylabel")` : x軸の名称
- `ax.set_ylabel("ylabel")` : y軸の名称
- `ax.legend(loc="upper left")` : 凡例の位置(左上)
- `ax.hlines(y, xmin, xmax)` : 横線の描画
colors:色, linestyle:線の種類
- `ax.vlines(x, ymin, ymax)` : 縦線の描画

その他matplotlibの使い方について

- <https://matplotlib.org/index.html>
matplotlibの公式サイト
- https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html
matplotlibのpyplot.plotの使い方
- <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html>
pandasのplotの使い方