

犯罪種別分析の流れ

2024-07-22

目次

犯罪種別分析（大まかな方向性）	1
データ	1
分析の方向性	1
論文	2
学ぶべきこと	3
大まかなスケジュール	3
メモ/調べること	3
学んだことまとめ	3
分析	4

犯罪種別分析（大まかな方向性）

データ

eStat を眺めていたら犯罪には 6 つの種類があることがわかった。種類は 6 つあり、「凶悪犯」、「粗暴犯」、「窃盗犯」、「知能犯」、「風俗犯」、「その他の刑法犯」に分かれる。これは刑法上の犯罪の分類¹である。刑法以外の犯罪を「特別法」というのだが、具体的にどのような犯罪か分かりづらいため、今回は省いて刑法の 6 つのみで分析する。

eStat の K 安全から認知件数を 6 つ取得した。

分析の方向性

大まかなモチベーションとしては、どの都道府県はどのような犯罪傾向を持っているか・どの都道府県に類似傾向があるかなどを分析して、どのようにその犯罪に対応すべきかを考えることである。

まずは種別でクラスタ分析を行いたい。たとえば、A 県と B 県は粗暴犯と窃盗犯が多いクラスタ 1 に属するように、県ごとの傾向を大まかに分類したい。それぞれのクラスタに必要な対応は異なるはずで、どのような傾向があるかを示すのは有効である。

クラスタ分析を行った後の方向性については決まっていないが、クラスタ分析を行った後に有効な統計手法があるかを調べていきたい。

¹警察白書

論文

少し調べてみた感じだと、地域ごとの犯罪類型に関する論文は見つからなかったが、同様の分析をしていてもおかしくないなので調べていきたい。

調べる論文系統

- (1) 地域ごとの犯罪傾向・類型
- (2) 犯罪に限らず都道府県レベルの類型・クラスタ分析手法を用いた分析
- (3) 犯罪に関する知識になりそうな論文
- (4) 計量経済学的手法の論文←どんな方法があるかを知るため

犯罪と社会経済情勢に関するシンプルな計量分析

犯罪と3つの説明変数からパネルデータを用いて、重回帰分析をしている。非説明変数同士に多重共線性が考えられるため、パネルデータの固定効果モデルを用いている。他にもモデルはある。

シンプルでわかりやすく面白かった。

単回帰分析・重回帰分析・パネルデータ分析(モデルの選定も!)を重ねて、各分析によってそれぞれの変数の説明が変わることがわかった。変わった理由についても考察されている。

Beckerの話は出てきていない。説明変数は3つで、一般論からの説明変数の選定だと思う。

失業率と犯罪発生率の関係: 時系列および都道府県別パネル分析

先行研究の部分が犯罪に関しての研究の流れを掴みやすい。他の犯罪に関する論文に関してもBeckerという人が登場したので、おそらく犯罪における経済学の重要人物だと考えられる。

(海外)

「実際、いくつかの研究は、失業率よりも賃金の方が犯罪率の変動の説明力が高いことを示している。」

「脱落変数バイアスを取り除くためには、できるだけ多くの説明変数を加える必要があるが、それにも限界がある。そこで、近年用いられている手法は、地域別のパネルデータを用いるというものである。」

「「逆の因果関係」「操作変数を探し出して、操作変数法を用いて推定する」

「多くの犯罪の種類において犯罪発生率、都市化、離婚率、警察力、若年失業率、居住開始率との間に共和分関係が認められることを示している。」

「殺人と強盗については、1人あたり消費が、窃盗については失業率が、長期的な発生率をより説明することを示している。」

(日本)

「鉱工業・通信・水道光熱・建設業従事者比率と犯罪発生率が負の関係にあること、所得不平等度のジニ係数と少年検挙率が正の相関をもつこと」

「Tsushima (1996) は、1986 年から 1988 年の間の 3 年間の都道府県別データを平均して、犯罪の種類別に、失業率、所得不平等度、貧困率等が犯罪発生率に与える影響を推定している。」

「朴(1993a、b、1994、1998) は、1954 年から 1988 年にかけての時系列データを用いて、犯罪種類別の犯罪発生率を様々な要因で回帰分析している」

この論文はパネルデータを用いて、失業率と犯罪が関係あるかを見てる。また種別の言及もしてあるため、この論文はしっかり見たいかも！

少年犯罪と労働市場：時系列および都道府県別パネル分析+

前置きは上の論文と似ていた。

これもやりたいことと似ている気がするのでちゃんと読みたい。

学ぶべきこと

- ・ 計量経済学的手法の本を読む
- ・ 犯罪に関する知識
- ・ 論文の書き方

大まかなスケジュール

- ・ ずっと 犯罪知識・計量経済学的知識
- ・ 7月21日~7月24日 論文めぐり
- ・ 7月24日 作り切る・教授に見てもらう
- ・ 8月1日 応募する
- ・ 8月30日 死ぬ気でやる

メモ/調べること

- ・ ☑ 計量経済学におけるパネルデータ分析のメリットとデメリットとは
- ・ Becker とは
- ・ F 値とは
- ・ 失業率と犯罪発生率の関係：時系列および都道府県別パネル分析を理解して読む
- ・ Tsushima (1996)を読む
- ・ 朴(1993a、b、1994、1998)を読む

学んだことまとめ

(5) 犯罪の重要説明変数

犯罪の発生率は失業率と密接しているようです。犯罪の発生率に関する論文を調べてみると、多くは題名から失業率と犯罪の発生率の関係性について言及するものが多いです。

(6) 計量経済学のパネルデータ分析

1 時点のデータ（横断的）を使ったり、時系列的なデータ（縦断的）を使うことによる問題を解決できる手法（横断的縦断的）らしい。どっかの教科書にはめっちゃいい手法のように書かれてた。

有名なモデルは固定効果モデルや変量効果モデル。適切なモデルを選択する方法として、Hausman 検定がある。

統計データ分析コンペティションの受賞論文を眺めていたら基本的にはこのパネルデータ分析を用いているものが多い。おそらく手に入るデータセットがパネルデータで、パネルデータ分析をすることが可能だから？

なんかわかりやすそう

ただパネルデータ分析をすることで得られた結果が何なのかがよくわかっていない。

分析

1975 年からのデータを使いやすい形に変換した。

```
import numpy as np
import pandas as pd

all_data = pd.read_csv("./data/all.csv")

all_data.head()

all_data = all_data.drop('/項目', axis=1)

all_data = all_data.rename(columns={
    '調査年': 'year',
    '地域': 'prefecture',
    'K4201_刑法犯認知件数【件】': 'keiho_number',
    'K420101_凶悪犯認知件数【件】': 'kyoaku_number',
    'K420102_粗暴犯認知件数【件】': 'sobou_number',
    'K420103_窃盗犯認知件数【件】': 'settou_number',
    'K420104_知能犯認知件数【件】': 'tinou_number',
    'K420105_風俗犯認知件数【件】': 'huzoku_number'
})

# year 列から年度を抽出
all_data['year'] = all_data['year'].str.extract(r'(\d+)').astype(int)

# 列名を変更する関数
def rename_columns(df, year):
    return df.rename(columns={
        'keiho_number': f'{year}_keiho_number',
        'kyoaku_number': f'{year}_kyoaku_number',
```



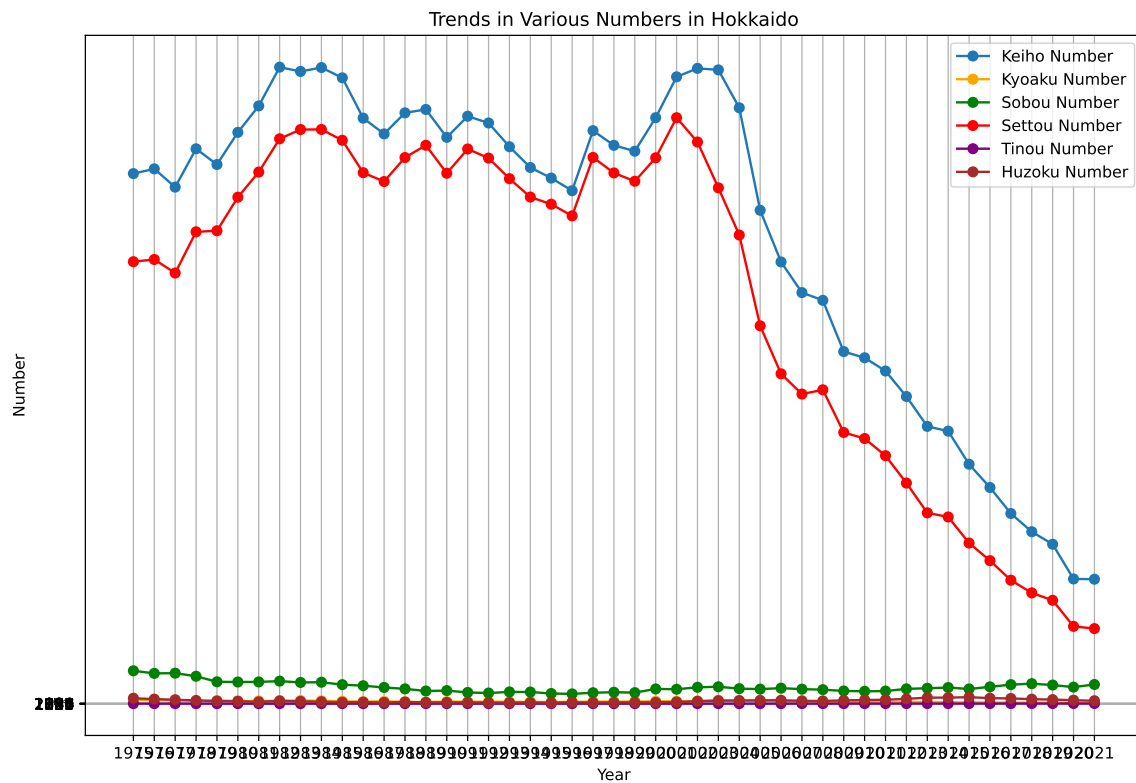
```

# 各指標のプロット
plt.plot(hokkaido_data['year'], hokkaido_data['keiho_number'], marker='o',
label='Keiho Number')
plt.plot(hokkaido_data['year'], hokkaido_data['kyoaku_number'], marker='o',
label='Kyoaku Number', color='orange')
plt.plot(hokkaido_data['year'], hokkaido_data['sobou_number'], marker='o',
label='Sobou Number', color='green')
plt.plot(hokkaido_data['year'], hokkaido_data['settou_number'], marker='o',
label='Settou Number', color='red')
plt.plot(hokkaido_data['year'], hokkaido_data['tinou_number'], marker='o',
label='Tinou Number', color='purple')
plt.plot(hokkaido_data['year'], hokkaido_data['huzoku_number'], marker='o',
label='Huzoku Number', color='brown')

# グラフのタイトルとラベルの設定
plt.title('Trends in Various Numbers in Hokkaido')
plt.xlabel('Year')
plt.ylabel('Number')
plt.grid(True)
plt.legend()
plt.xticks(hokkaido_data['year']) # 年度ごとにX軸のラベルを表示

# グラフの表示
plt.show()

```



さらっとみていたところ知能犯はある時期からできたっぽい。

年度で比べると明らかに低下しているが、都道府県ごとの比較をしたい。

```
import pandas as pd
import matplotlib.pyplot as plt

# 2021年の北海道のデータを抽出
hokkaido_2021 = all_data[(all_data['prefecture'] == '北海道') & (all_data['year'] == 2021)]

# レーダーチャートのデータ準備
labels = ['Keiho Number', 'Kyoaku Number', 'Sobou Number', 'Settou Number', 'Tinou Number', 'Huzoku Number']
values = hokkaido_2021[['keiho_number', 'kyoaku_number', 'sobou_number', 'settou_number', 'tinou_number', 'huzoku_number']].values.flatten().tolist()
values += values[:1] # レーダーチャートを閉じるために最初の値を末尾に追加

# 角度の計算
angles = [n / float(len(labels)) * 2 * np.pi for n in range(len(labels))]
angles += angles[:1]

# レーダーチャートのプロット
plt.figure(figsize=(8, 8))
```

```

ax = plt.subplot(111, polar=True)

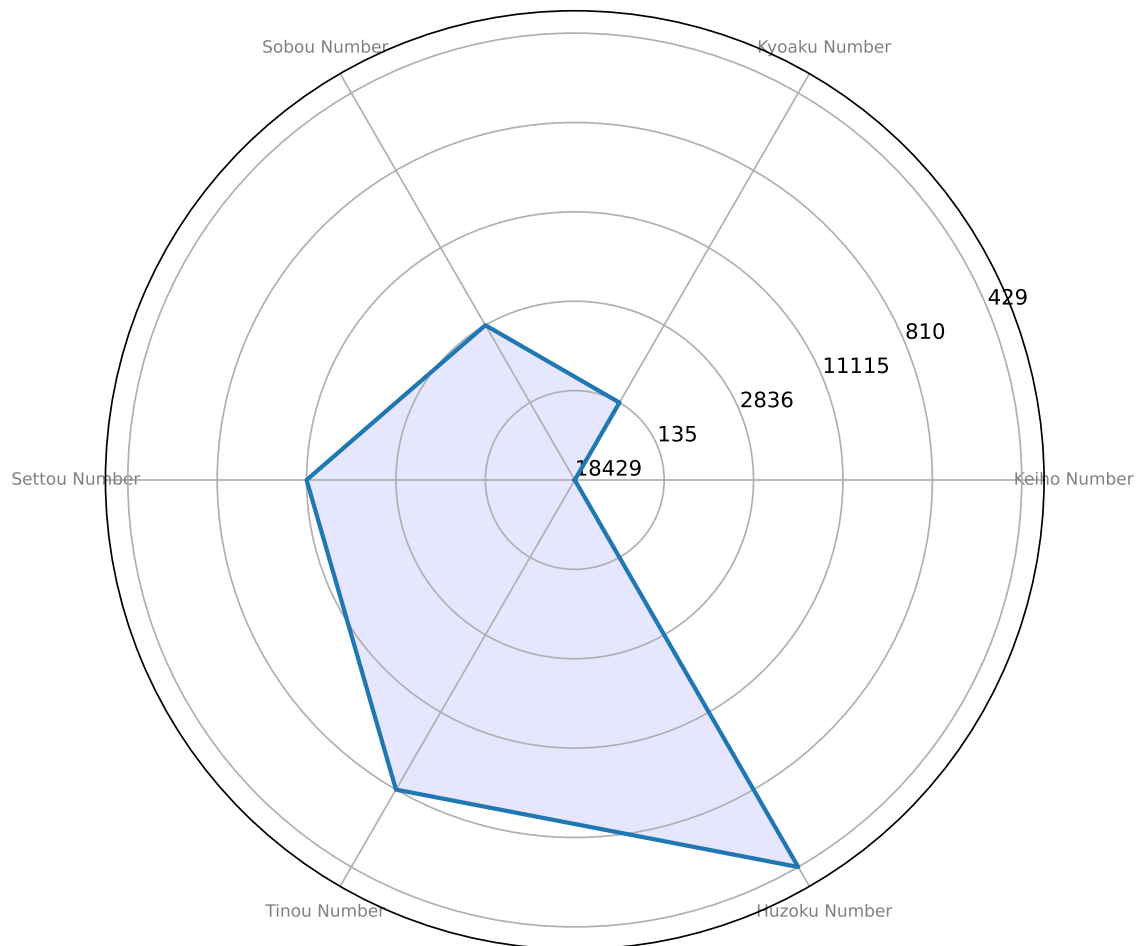
plt.xticks(angles[:-1], labels, color='grey', size=8)
ax.plot(angles, values, linewidth=2, linestyle='solid')
ax.fill(angles, values, 'b', alpha=0.1)

# グラフのタイトル設定
plt.title('2021 Hokkaido Crime Statistics', size=20, color='blue', y=1.1)

# グラフの表示
plt.show()

```

2021 Hokkaido Crime Statistics




```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
import japanize_matplotlib

df = all_data.copy()

# 2021年のデータのみをフィルタリング
df_2021 = df[df['year'] == 2021]

# 必要な列を標準化
scaler = StandardScaler()
df_2021[['keiho_number', 'kyoaku_number', 'sobou_number',
'settou_number', 'tinou_number', 'huzoku_number']] =
scaler.fit_transform(df_2021[['keiho_number', 'kyoaku_number', 'sobou_number',
'settou_number', 'tinou_number', 'huzoku_number']])

# 対象の都道府県を抽出
target_prefectures = ['北海道', '沖縄県', '福岡県', '東京都', '大阪府']
target_data = df_2021[df_2021['prefecture'].isin(target_prefectures)]

# レーダーチャートのデータ準備
labels = ['Keiho Number', 'Kyoaku Number', 'Sobou Number', 'Settou Number',
'Tinou Number', 'Huzoku Number']
angles = [n / float(len(labels)) * 2 * np.pi for n in range(len(labels))]
angles += angles[:1]

plt.figure(figsize=(7, 7))

for prefecture in target_prefectures:
    data = target_data[target_data['prefecture'] == prefecture]
    values = data[['keiho_number', 'kyoaku_number', 'sobou_number',
'settou_number', 'tinou_number', 'huzoku_number']].values.flatten().tolist()
    values += values[:1] # レーダーチャートを閉じるために最初の値を末尾に追加
    ax = plt.subplot(111, polar=True)
    ax.plot(angles, values, linewidth=2, linestyle='solid', label=prefecture)
    ax.fill(angles, values, alpha=0.1)

plt.xticks(angles[:-1], labels, color='grey', size=8)
plt.title('2021 Crime Statistics (Standardized)', size=20, color='blue', y=1.1)
plt.legend(loc='upper right', bbox_to_anchor=(1.1, 1.1))

# グラフの表示
plt.show()

```

```
/var/folders/34/77s51jk90j9by6zm4wx73wkw0000gn/T/  
ipykernel_50751/2468611948.py:14: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

2021 Crime Statistics (Standardized)

