IRDBデータによる日本の機関リポジトリの特徴の分析

小野 亘[†1,a)]

概要：本稿は、Python のデータ解析のためのライブラリである PANDAS を用い、NII の IRDB データから、日本の機関リポジトリの特徴の分析を行ったものである。日本の機関リポジトリは、紀要に大きな特徴があることをデータの上から明らかにし、特徴的な機関の抽出を行った。

キーワード：IRDB, 機関リポジトリ, 紀要

Analysis of the characteristics of Japanese institutional repositories using IRDB data

Wataru, ONO[†1,a)]

Abstract: This paper is an analysis of the characteristics of Japanese institutional repositories from NII's IRDB data using PANDAS, a library for data analysis in Python. The data revealed that Japanese institutional repositories have significant characteristics in their departmental bulletin paper (Kiyou), and I extracted the characteristic institutions.

Keywords: IRDB, Intitutional repositories, Departmental Bulletin Paper

1. はじめに

本稿では、国立情報学（NII）の IRDB から統計データを取得し、日本の機関リポジトリに登録されたコンテンツの種別 (資源タイプ）の分析を行い、その側面から、日本の機関リポジトリの特徴を明らかにするものである。

2. 方法

Python のデータ解析のためのライブラリである PANDAS を用い、NII の IRDB データから、日本の機関リポジトリの特徴の分析を行ったものである。資源タイプは 47 項目あるが、大項目 12 に紀要を加えた、13 項目について検討した。今回は、Python 3.10.7、Pandas 1.5.2 を使用した。分析に使用した Jupyter Notebook ファイルは以下 に置いた。

†1 現在，東京大学教養学部（駒場図書館）
Presently with Komaba Library, University of Tokyo
a) ono.wataru@mail.u-tokyo.ac.jp
https://irdb.nii.ac.jp/
https://github.com/wonox/irdbscraping/blob/main/irdb.ipynb

3. 結論

日本の機関リポジトリでは、リポジトリの中で、紀要の占める割合が 9 割を超える機関が 7 割を超えていることが分かった。、主成分分析を用いて特徴的な機関の抽出を行った結果、第 1 主成分には Article と Book が、第 2 主成分には Kiyou の影響が大きいことが分かり、第 1 主成分が高い東京工業大学が Article と Book の割合が高いこと、第 2 主成分が高い北海道大学や早稲田大学が Kiyou や Lecture の割合が高いこと、外れ値を抽出してみると、' 東京経済大学', ' 東北大学', ' 大阪大学', ' 大阪市立大学', ' 岡山大学', ' 新潟大学', ' 九州大学', ' 神戸大学', ' 慶應義塾大学', ' 金沢大学', ' 海洋研究開発機構', ' 宇宙航空研究開発機構', ' 一橋大学', ' 広島大学' が特徴的な機関であることが分かった。

4. データの読み込み、前処理

4.1 準備

NII の IRDB の「コンテンツ統計（全体)」から、統計データを取得し、

```
https://irdb.nii.ac.jp/statistics/all
```

から、「統計ファイル（全機関の機関別統計）」をクリックし、ダウンロードする。今年度（2023 年度）であれば、2023.csv というファイルが取得できるので、それを Pandas の DataFrame に読み込む

上記で作った df2205i を全件数と、本文ありとに分け、差分（メタデータのみ）を取り出し、列名を振りなおす（コードの詳細は省略）。

```
df2205_all = df2205i.iloc
    [:,17:64]
df2205_honbun = df2205i.iloc
    [:,75:]
df2205_sabun = df2205_all -
    df2205_honbun
```

Program 1

hoge

4.2 資源タイプの集約

機関リポジトリには、コンテンツの種類を表す 47 個の資源タイプのいずれかを付すことになっているが、[2] 以下の大項目 13 個にまとめる。

(1) departmental bulletin paper: 紀要
(2) Article: 論文
(3) Book: 図書
(4) Cartographic Material: 地図
(5) Conference object: 会議録
(6) Dataset: データセット
(7) Image: イメージ
(8) Lecture: 講演
(9) Patent: 特許
(10) Report: 報告書
(11) Sound: 音声
(12) Thesis: 学位論文
(13) Multiple: その他

JPCOAR スキーマガイドラインの資源タイプ語彙別表[*1] で示されている大項目は 12 項目だが、日本の機関リポジトリは、紀要に一つの特徴があることが経験上推測されるため、Article の内数である、departmental bulletin paper（紀要）を別項目とし、全部で 13 項目とした。

ここで作成したデータフレームを

```
pd.set_option('display.
    float_format', lambda x: '
    %.1f' % x)
# 小数点二位で切り捨て
df2205_all_d.describe()
```

[*1] https://schema.irdb.nii.ac.jp/ja/resource_type_vocabulary

にて確認すると、表 1 のとおりとなる。

5. 構成比

全体の構成は図 1 のとおりとなる。一見して、kiyou が多くを占めていることが分かる。

図 2 のとおり、全体（n=792）を、リポジトリごとのコンテンツ数の比率の高い順に並べてみると、kiyou が多くを占め、かつ kiyou が各リポジトリの中でほとんどを占めている機関が多いことが、よりはっきりする。

実際には、資源タイプの内、Kiyou が 90

6. 主成分分析

前記、前処理の時点で、13 次元のデータであり、それを可視化することは不可能である。そこで、主成分分析を用いて情報をなるべく失うことなく 2 次元へと次元圧縮をし、データの可視化をおこなってみる。主成分分析（principal component analysis）とは、相関のある多数の変数から、相関のない少数で全体のばらつきを最もよく表す、主成分と呼ばれる変数を合成する多変量解析の一手法で、データの次元を削減するために用いられる。

6.1 主成分分析の前処理

主成分分析の前処理として、標準偏差が小さい列を削除しておく。標準偏差が小さい、ということは、全体のバラツキが小さいということ、つまり、測定値の分布が平均値の周りに集まっているということを表している。

```
std0 = (df2205_all_d.std(axis=0)
    > 1)
df2205_all_dstd = df2205_all_d.
    loc[:, std0]
```

上記で、標準偏差が 1 より小さい列を削除することで、Cartographic Material、Patent、Report、Sound、Multiple の 5 列が削除されて、8 列（8 項目）になる。

ここで作成したデータフレームを

```
df2205_all_dstd.describe()
```

にて確認すると、表 2 のとおりとなる。

25%は第一四分位数、75%は第三四分位数を表しており、kiyou（紀要）を除けば、Book と Thesis の 2 つに 75%は第三四分位数に数字があるのみであることが分かる。四分位数とは、

“データを小さい方から並び替え、データの個数（サンプルサイズ）で 4 等分した時の区切り点を四分位数と言う。それぞれ 25 パーセンタイル（第一四分位数）、50 パーセンタイル（中央値）、75 パーセンタイル（第三四分位数）とよばれる。”

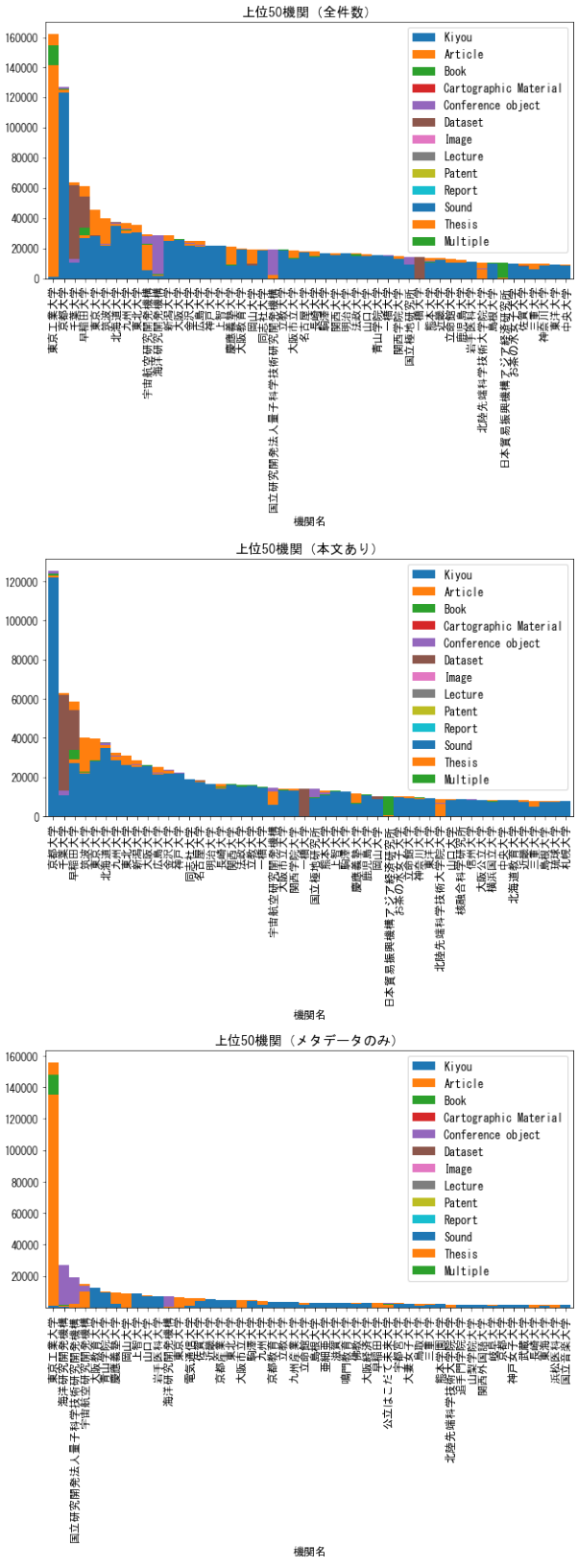

図 1　コンテンツ数上位 50 機関で構成比率の積み上げグラフ

Fig. 1　Stacked graph of composition ratios for the top 50 institutions in terms of number of contents

*2

であり、紀要の場合全体の 3/4 の位置で 129 件あるが、

*2　https://bellcurve.jp/statistics/glossary/1919.html

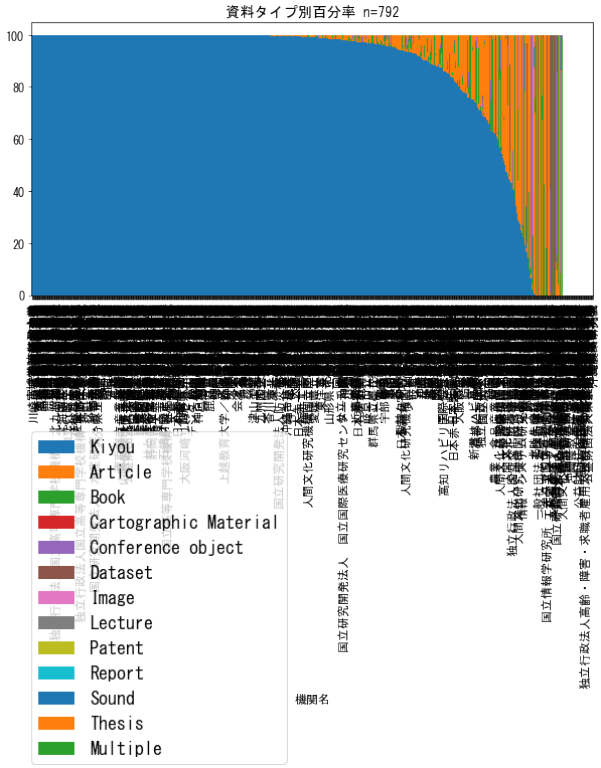

図 2　資料タイプ別百分率

Fig. 2　Stacked graph of composition ratios for the top 50 institutions in terms of number of contents

図書は 1/4 の位置で 2 件、学位論文は 1/4 の位置で 20 件しかないことが分かる。紀要を除けば、他の資源タイプはゼロ件のリポジトリが圧倒的に多いことが推測できる。

この df2205_all_dstd で箱ひげ図を描いてみると、図 3 のようになる。

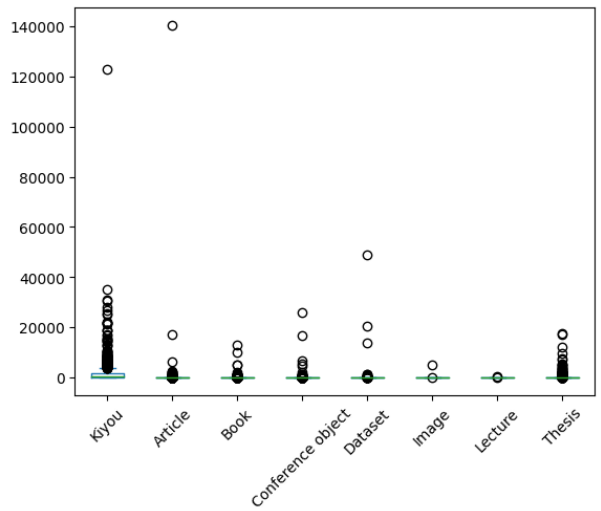

図 3　外れ値を除いた箱ひげ図

Fig. 3　Single column figure with caption explicitly broken by \\.

箱ひげ図は、データのばらつきをわかりやすく表現するための統計図であり、一番上の線が最大値、一番下の線が

最小値、中央の線が中央値などを示し、丸は外れ値であり値に含まれない。しかし、図 3 では、外れ値ばかりで、このままではばらつきなどが分からない。そこで、それぞれ大きく外れているデータを以下で確認する。

```
df2205_all_dstd[df2205_all_dstd['Kiyou']>40000]
df2205_all_dstd[df2205_all_dstd['Article']>40000]
df2205_all_dstd[df2205_all_dstd['Dataset']>40000]
```

結果は表 3 のとおりとなった。

'Kiyou' は京都大学が、'Article' は東京工業大学が、'Dataset' は千葉大学が、それぞれ大きく突出していることが分かる。ここではデータの全体の感じをつかむため、この 3 機関を外し、さらにデータの多い'Kiyou' を外して箱ひげ図を再度作成する。

図 4 のとおり、これでもあまり意味のある図にはならなかった。

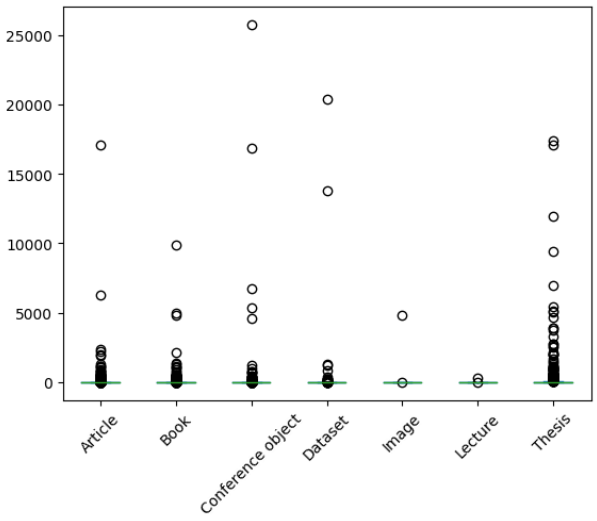

図 4 外れ値を除いた箱ひげ図
Fig. 4 Single column figure with caption explicitly broken by \\.

6.2 特徴量の確認

相関行列や散布図を用いて特徴量の分布などを確認する。

Kiyou と Article はあまり関係がなく、Kiyou と関係があるのは Thesis であることが分かる。これは、日本の機関リポジトリの一般的な印象とも合致する。

6.3 特徴量の標準化

IRDB のデータから機関リポジトリの特徴を見るためには、量的な特徴を見るのが分かりやすいが、機関リポジトリ

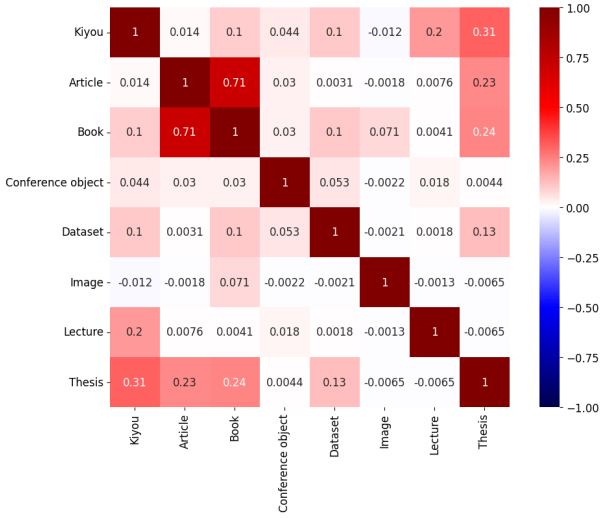

図 5 相関行列のヒートマップ (相関係数の値あり)
Fig. 5 Single column figure with caption explicitly broken by \\.

ごとのコンテンツの量に差がありすぎるため、量以外の特徴がつかみにくい。ここでは、分析の前処理として、特徴量を標準化する。標準化はスケーリング (Feature Scaling) の一種で、特徴量間のスケールを変換することである。特徴量間で異なるスケールを揃えるため、資源タイプごとに、元のデータの平均を 0、標準偏差が 1 となるように変換する。

```
# 変数（特徴量）の標準化
df2205_all_std = df2205_all_dstd.apply(lambda x: (x-x.mean())/x.std(), axis=0)
# 結果を少数以下2桁で丸めて表示
df2205_all_std.describe().round(2)
```

結果は 4 のとおり、平均が 0、標準偏差が 1 となるように変換されていることが分かる。

6.4 主成分分析

以下で、主成分分析を実行する。

```
from sklearn.decomposition import PCA
# 主成分分析の実行
pca = PCA()
pca.fit(df2205_all_std)

# データを主成分に変換
pca_row = pca.transform(df2205_all_std)
```

次に、寄与率を求め、累積寄与率のグラフを書きます。寄与率は、データの全情報の中で、各要素のもつ情報が占める割合を表し、値が大きいほど相対的に説明力が高い主成分であることを示す。
累積寄与率は、寄与率を大きい順に順次足したもので、主成分が全体の中でどれだけの割合を占めるかを示す。

```
# 寄与率を求める
pca_col = ["PC{}".format(x + 1) for x in range(len(df2205_all_std.columns))]
df_con_ratio = pd.DataFrame([pca.explained_variance_ratio_], columns = pca_col)
print(df_con_ratio)

# 累積寄与率を図示する
cum_con_ratio = np.hstack([0, pca.explained_variance_ratio_]).cumsum()
plt.plot(cum_con_ratio, 'D-')
plt.xticks(range(9))
plt.yticks(np.arange(0,1.05,0.05))
plt.grid()
plt.show()
```

以下のグラフによれば、第 1 主成分の寄与率は，0.23951 である一般的には、累積寄与率が 80 %以上になる主成分数を採用して分析結果に用いることが多いと言われているが、第 2 主成分までだと約 40

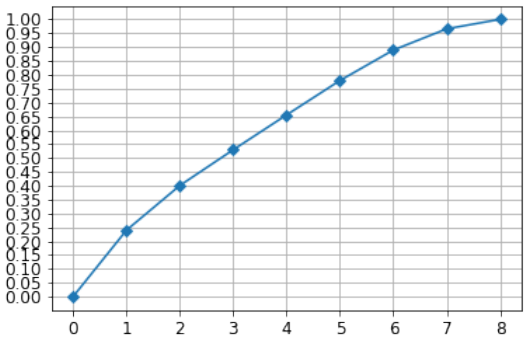

図 6 累積寄与率
Fig. 6 cumulative contribution ratio

第 1 主成分と第 2 主成分でプロットしてみる。

```
# 第1主成分と第2主成分でプロットする
import matplotlib.pyplot as plt
plt.rcParams["font.family"] = "Meiryo" # "MS Gothic"
plt.figure(figsize=(12, 12))
plt.scatter(pca_row[:, 0], pca_row[:, 1], alpha=0.8)  # c=list(df2205_all_std.iloc[:, 0]))
plt.grid()
plt.xlabel("PC1")
plt.ylabel("PC2")

for k, v in pca_tokuten_0.iterrows():
    plt.annotate(k,xy=(v[0],v[1]),size=8)
plt.show()
```

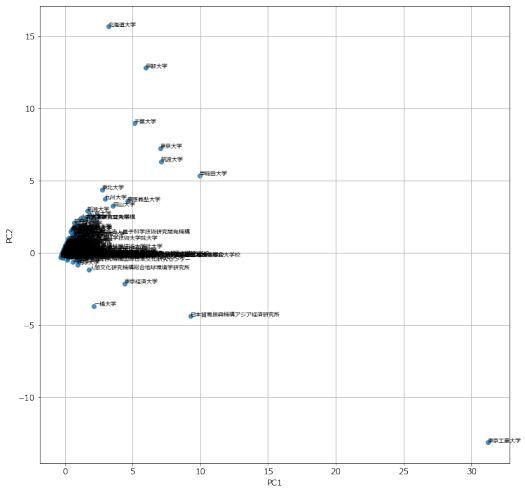

図 7 第 1 主成分と第 2 主成分
Fig. 7 first and second principal component

グラフのとおり、第 1 主成分では東京工業大学が、第 2 主成分では北海道大学や早稲田大学などが突出しており、原点付近に集中しているため、文字が重なり真っ黒になってしまった。

次に、主成分に対する各変数の影響度合いを見るため、主成分負荷量を求める。これにより，各主成分が何を意味しているかが分かりやすくなる。

```
# 主成分負荷量を求める
df_pca = pd.DataFrame(pca_row, columns = pca_col)
```

```
    df_pca_vec = pd.DataFrame(pca.
        components_, columns=
        df2205_all_std.columns,
        index=["PC{}".format(x + 1)
            for x in range(len(df_pca
            .columns))])
    print(df_pca_vec)

    # 主成分負荷量を図示する
    plt.figure(figsize=(6, 6))
    for x, y, name in zip(pca.
        components_[0], pca.
        components_[1],
        df2205_all_std.columns[0:]):
        plt.text(x, y, name)
    plt.scatter(pca.components_[0],
        pca.components_[1])
    plt.grid()
    plt.xlabel("PC1")
    plt.ylabel("PC2")
    plt.show()
```

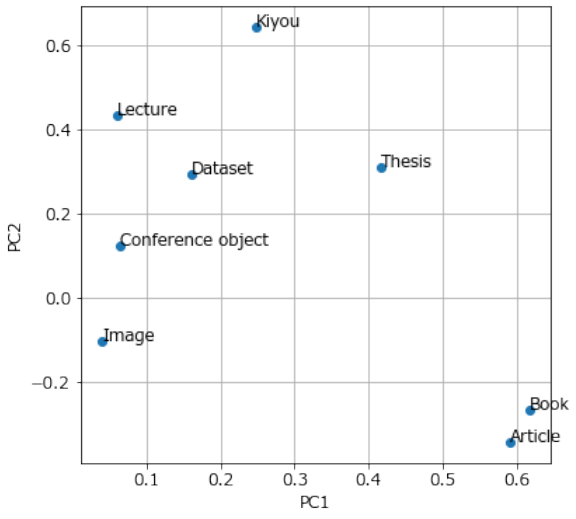

図 8　主成分負荷量

Fig. 8　principal component loading

図 8 を見ると、第 1 主成分には Article と Book が、第 2 主成分には Kiyou の影響が大きいことが分かり、図 7 で、第 1 主成分が高い東京工業大学が Article と Book の割合が高いこと、第 2 主成分が高い北海道大学や早稲田大学が Kiyou や Lecture の割合が高いこと、また、図 5 の印象とも一致する。

図 7 では、いわゆる外れ値が大きいので、外れ値を抜いて分析してみる。

```
    # 外れ値（例外的に、値が大きな機関）を削除する
    # nd配列pca_rowをDataFrameにする
    pca_row_df = pd.DataFrame(
        pca_row, index= list(
        df2205_all_std.index),
        columns=["PC{}".format(x +
        1)
            for x in range(len(
                df2205_all_std.columns
                ))])
    outlier = list(pca_row_df[
        pca_row_df[['PC1','PC2']] >
        5].dropna(how='all').index)
    pca_row_df_o = pca_row_df.drop(
        outlier)
    print(outlier)
```

これにより、' 早稲田大学', ' 筑波大学', ' 東京工業大学', ' 東京大学', ' 京都大学', ' 日本貿易振興機構アジア経済研究所', ' 北海道大学', ' 千葉大学' 8 機関が抽出され、日本の機関リポジトリとしては特徴的な機関であることが分かる。さきほど同じように、第 1 主成分と第 2 主成分でプロットしてみる。

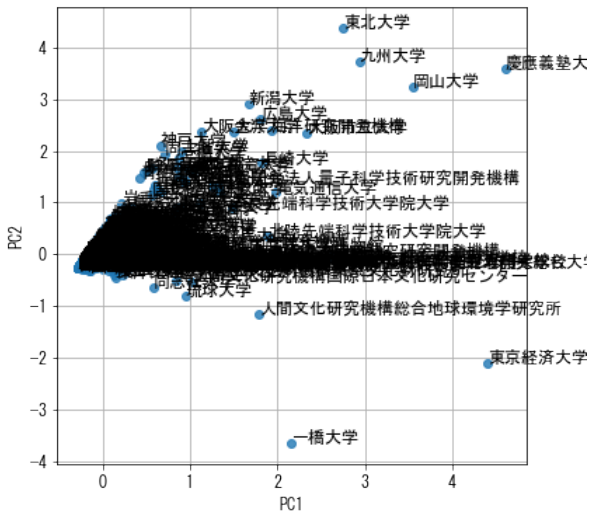

図 9　第 1 主成分と第 2 主成分（外れ値を削除）

Fig. 9　first and second principal component:outlier

さらに同じように外れ値を抽出してみると、' 東京経済大学', ' 東北大学', ' 大阪大学', ' 大阪市立大学', ' 岡山大学', ' 新潟大学', ' 九州大学', ' 神戸大学', ' 慶應義塾大学', ' 金沢

大学', ' 海洋研究開発機構', ' 宇宙航空研究開発機構', ' 一橋大学', ' 広島大学' が特徴的な機関であることが分かった。

7. おわりに

Python の使用には、[1] などを参考にした。

参考文献

[1] 寺田学, 辻真吾, 鈴木たかのり, 福島真太朗: Python によるあたらしいデータ分析の教科書翔泳社 (2022).

[2] 大園 隼彦, 片岡 朋子, 高橋 菜奈子, 田口 忠祐, 林 豊, 南山 泰之: “JPCOAR スキーマの策定：日本の学術成果の円滑な国際的流通を目指して.”情報管理, vol. 60, no. 10, 2017, pp. 719–29 urljhttps://doi.org/10.1241/johokanri.60.719.（参照 18 Mar. 2023.）

表 1 df2205_all_d.describe

Table 1 df2205_all_d.describe

	Kiyou	Article	Book	Cartographic Material	Conference object	Dataset	Image	Lecture	Patent	Report	Sound	Thesis	Multiple
count	792.0	792.0	792.0	792.0	792.0	792.0	792.0	792.0	792.0	792.0	792.0	792.0	792.0
mean	2011.5	249.6	78.3	0.0	93.8	112.7	6.1	0.4	0.0	0.0	0.0	217.2	0.0
std	5952.6	5051.3	643.6	0.4	1151.1	1948.2	170.3	10.4	0.0	0.0	0.2	1201.0	0.0
min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	128.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	482.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	1517.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19.0	0.0
max	123440.0	140940.0	12907.0	10.0	25743.0	48965.0	4792.0	294.0	0.0	0.0	7.0	17369.0	0.0

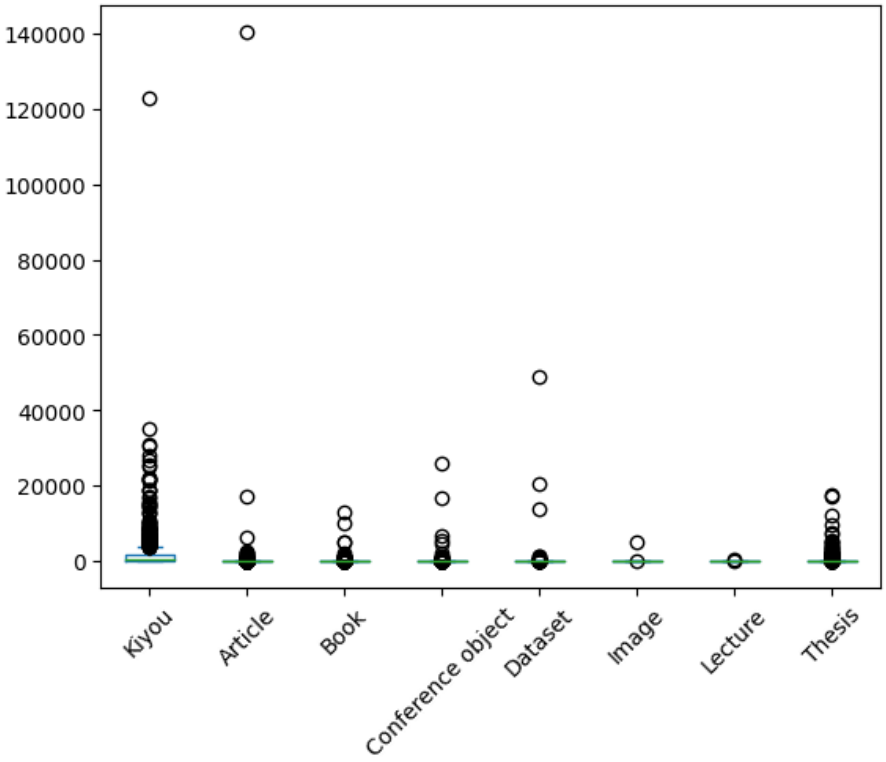

図 10 df2205_all_dstd の箱ひげ図

表 2 df2205_all_dstd.describe

Table 2 df2205_all_dstd.describe

	1Kiyou	2Article	2Book	5Conference object	5Dataset	6Image	7Lecture	8Thesis
count	789.000000	789.000000	789.000000	789.000000	789.000000	789.000000	789.000000	789.000000
mean	1993.239544	249.602028	78.128010	93.576679	112.820025	6.076046	0.372624	218.742712
std	5919.979304	5044.485846	644.070755	1152.428752	1949.766679	170.599642	10.431092	1200.556001
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	129.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	487.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	1506.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	20.000000
max	122935.000000	140481.000000	12883.000000	25743.000000	48965.000000	4792.000000	293.000000	17369.000000

表 3 df2205_all_dstd.describe

Table 3 df2205_all_dstd.describe

	1Kiyou	2Article	2Book	5Conference object	5Dataset	6Image	7Lecture	8Thesis
京都大学'Kiyou'	122935	1460	940	1147	254	0	0	0
東京工業大学'Article'	828	140481	12883	0	11	0	0	7572
千葉大学'Dataset'	10557	91	134	2065	48965	0	0	1970

表 4 df2205_all_dstd.describe

Table 4 df2205_all_dstd.describe

	1Kiyou	2Article	2Book	5Conference object	5Dataset	6Image	7Lecture	8Thesis
count	789.00	789.00	789.00	789.00	789.00	789.00	789.00	789.00
mean	-0.00	-0.00	0.00	0.00	0.00	0.00	-0.00	0.00
std	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
min	-0.34	-0.05	-0.12	-0.08	-0.06	-0.04	-0.04	-0.18
25%	-0.31	-0.05	-0.12	-0.08	-0.06	-0.04	-0.04	-0.18
50%	-0.25	-0.05	-0.12	-0.08	-0.06	-0.04	-0.04	-0.18
75%	-0.08	-0.05	-0.12	-0.08	-0.06	-0.04	-0.04	-0.17
max	20.43	27.80	19.88	22.26	25.06	28.05	28.05	14.29