スケジュール

day 2

- 講義 ― 自然言語処理の最新動向
- 講義 ― テキストマイニングの手順
- 講義&演習 ― データ理解

day 3

- 講義&演習 ― 演習環境の準備
- 講義&演習 ― テキスト解析 (1)
- 講義&演習 ― テキスト解析 (2)

day 4

- 講義&演習 ― テキスト分析 (1)
- 演習 ― テキスト分析 (実践編)

day 5

- 演習 ― テキスト分析 (実践編)

演習環境の準備

(再掲) 無償で利用できる機械学習環境

- 近年、機械学習の教育・研究を目的とした研究用ツールが相次いで登場

Colaboratory

https://colab.research.google.com

amazon SageMaker Studio Lab

https://studiolab.sagemaker.aws/

演習で使用
↓

	Clab(無償版)	Studio Lab
GPU	T4(16GB)	T4(16GB)
最長実行時間	12時間	CPU:12時間 GPU:4時間
メモリ	12GB	15GB
ディスク	CPU:100GB GPU:78GB	15GB (**永続化**)
ターミナル	×	○
ランタイムの保存と再開	×	○
費用	無償	無償
その他	Googleアカウントが必要	AWSアカウントは不要 (クレカ不要)

(再掲) SageMaker Studio Lab のアカウント作成

- https://studiolab.sagemaker.aws/ にログインして、アカウントを作成してください

アカウント作成手順

1. アカウント作成フォームからアカウントの申し込みを行う
   注意: **リファラルコード**をアカウント作成フォームに忘れずに入力ください (**受講者限定**です)
2. 「Account request confirmed ...」のメールを受信し、メール内のリンクからアカウントを作成する
   → リクエストの受付はすぐにメールが届きます
3. 「Verify your email ...」のメールを受信し、メール内のリンクからメールアドレスを認証する
   → リファラルコードを利用している場合は2~3分以内に結果が届きます
4. 「Your account is ready ...」のメールを受信する
   → これで「Sign in」できます

※ リファラルコードの有効期間: 2024/6/27 ~ 2024/7/12

演習環境の準備

- https://studiolab.sagemaker.aws/ にログインして、Runtimeを起動してください

1. ログインする

① 画面右上の「**Sign in**」ボタンを押す

② Eメールアドレス/ユーザー名、パスワードを入力する

③「**Sign in**」を押してプロジェクトのページを開く

2. Runtime を起動する

④「**My Project**」の「**Select compute type**」から「**CPU**」を選択する

⑤「**Start runtime**」を押す

⑥ 起動時に多要素認証を求められた場合、使用可能なデバイスで認証する

演習環境の準備

- Jupyter Lab を起動して、教材を開いてください

3. Jupyter Lab を起動する

① ランタイムが開始したら「**Open project**」を押す

4. 教材を開く

② 「**Git**」 ボタンを押す

③ 「**Clone a Repository**」を押す

④ 「**Git repository URL ...** 」に「https://github.com/haradatm/lecture-gssm2024」を入力する

⑤ (任意)「**Project directory to ...** 」に保存先のパスを入力する

例)「sagemaker-studiolab-notebooks」

演習環境の準備

- Jupyter Lab を起動して、教材を開いてください (続き)

4. 教材を開く (続き)

⑥ 画面右下の「**Clone**」を押す

⑦ ポップアップした画面で「**OK**」を押す

この後、処理完了まで**7分程度**待ちます

⑧「**done**」が表示される

⑨ 続いて「**プロンプト**」が表示されれば完了

拡大図

注意:
⑦で「OK」を押さなかった場合、**⑧や⑨の表示がされない場合**は後述の「**トラブルシューティング**」から再開してください

(参考) トラブルシューティング

- P.8 の⑦で「OK」を押さなかった場合や、⑧や⑨の表示がされない場合

① 画面左の **File Browser** から「**environment.yml**」を選択する

② 右クリックメニューから「**Build Conda Environment**」を選択する → P.8 ⑦ へ

(参考) トラブルシューティング

- (いちからやり直すために) プロジェクトを完全削除する **注意: この操作は慎重に行ってください**

① 画面上の + ボタンで **Launcher** を開く

② 画面上の **Launcher** から **Terminal** を開く

③ **Terminal** で右のコマンドを実行する

④ **Stop runtime** で Runtime を停止 → P.7 へ

```
# ホームディレクトリへ移動
cd
# 仮想環境を削除
conda remove -n gssm2024 --all     # Proceed ([y]/n)? Y
# プロジェクトを削除
rm -fr sagemaker-studiolab-notebooks/lecture-gssm2024
```

(再掲) Jupyter Lab の教材を最新化するには

- 下記の手順で、教材を最新化(pull)することができます

- **教材を最新化**する

① 「**Git**」 ボタンを押す

② 「**Refresh**」 ボタンを押す

③ もし、競合がある場合(Changedが0でない場合)、**対象のファイルを手動で退避**した後、「**Discard changes**」 ボタンを押して変更を破棄する

④ 「**Pull latest changes**」 を押す

⑤ 画面の右下に「Successfully published」が表示されること確認する

テキスト解析 (1)

(再掲) テキストマイニングの手順

- データをよく知る
  - データ件数や構成比を集計 → データを理解する
    - 旅行目的別の人気エリアは?
    - 同伴者別の人気エリアは?
    - 数値評価による人気エリアの差異は?
- テーマを設定する
  - 解決すべき課題を決める → 分析目的を明確にする
    - 数値評価が低い原因は?
    - 高評価の施設に学ぶ改善点は?
- テキスト分析に取り組む
  - これら課題を解決するために、テキスト分析を実施

テキスト分析の手順

①自然言語処理によりテキストを数値化する → ②統計解析や可視化を行う → ③結果を読み解きながら解析のための辞書を編纂する → 分析のサイクルを回していく(①へ)

代表的なテキスト解析器

● **速度**重視では **MeCab**、**精度**と**出力情報の豊富さ**重視では **JUMAN++** がお勧め

出典: https://taku910.github.io/mecab/ をもとに加筆修正

形態素解析器		ChaSen	MeCab	JUMAN	JUMAN++
	コスト推定	HMM	CRF	人手	RNNLM
	探索方法	接続コスト最小法 (ビタビアルゴリズム)			
係り受け解析		~~Cabocha~~	CaboCha	KNP	

JUMAN++ 深層学習を使った手法で、自然な言葉の繋がりを考慮した

単語分割+品詞タグ付け精度 (F1)

処理速度 (文/秒)

学習・評価データ

京都大学テキストコーパス (NEWS),
京都大学ウェブ文書リードコーパス (WEB)

RNN言語モデルの学習

Webコーパス 1000万文

出所:
https://drive.google.com/file/d/1DVnrsWw4skRgC8jU6_RkeofOQEHFwctc/view?usp=sharing

モダンなテキスト解析器 **Ginza**

- Megagon Labs と国立国語研究所の共同研究結果として公開された OSS の日本語自然言語処理ライブラリ
  - 「著作権表示」と「MIT ライセンスの全文」を記載する、という2条件のみで商用利用が可能
- **spaCy** (機械学習を組み込んだ自然言語処理ライブラリ) 上で動作するので、**係り受け解析**や**固有表現抽出**などの機能も利用可能
  - 形態素解析には **Sudachi** (徳島人工知能NLP研究所)を利用、辞書は半年に約1回の頻度で更新されている (らしい)
  - 20億文以上のWebテキストで事前学習した **Transformers** モデルも利用可能

ただし、**高い利便性や機能**の一方で**処理が遅い** (形態素解析でMeCabの10倍ぐらい)

演習 ― Jupyter ノートブックの使い方

- day-3-1.ipynb を開いてください

  ① 画面左の **File Browser** から ① **notebooks** をフォルダを開く

  ② さらに **day-3-1.ipynb** ノートブックを開く

演習 ― Jupyter ノートブックの使い方

- カーネル **gssm2024:Python** を選択してください **!重要!**

  ① ページ上部の **Kernel** メニューから「**Change Kernel ...** 」を選ぶ

  ② ポップアップ画面から「**gssm2024:Python**」を選択し、「**Select**」を押す

  ③ 右上隅にカーネル名「**gssm2024:Python**」が表示されていることを確認する

演習 ― Jupyter ノートブックの使い方

● Python コードを実行する

- Jupyter ノートブックでは、Python のプログラムを実行することができます

  (別のパッケージをインストールすることで、R 言語や C++ も実行できます)

- ノートブック上で、Pythonコードを実行するには、**セルをクリックして選択**し、キーボードで **Shift-Enter** を入力するか、ドキュメントの上部にあるノートブックツールバーの**再生ボタンをクリック**します

練習：左図の赤枠にあるセルをクリックし、セル内の計算を実行する

演習 ― テキスト解析 (1)

- 形態素解析器 **MeCab** と、係り受け解析器 **CaboCha** をインストールする

「2.1 **MeCab** のインストール」

① セルをクリックして選択し、再生ボタンを押す

この後、処理完了まで約3分程度待ちます

②「**Successfully ...** 」を確認する

「2.2 **CaboCha** のインストール」

③ セルをクリックして選択し、再生ボタンを押す

この後、処理完了まで約5分程度待ちます

④「**Successfully ...** 」を確認する

「2.3 **Kernel** のリスタート」

⑤ メニューバーにある **Kernel** メニューをクリックし、プルダウンメニューから [**Restart Kernel ...**] を選択する

演習 ー テキスト解析 (1)

- day-3-2.ipynb を開いてください

  ① 画面左の **File Browser** から ① **notebooks** をフォルダを開く (既に開いている場合はスキップ)

  ② 次に **day-3-2.ipynb** ノートブックを開く

演習 ― テキスト解析 (1)

- カーネル **gssm2024:Python** を選択してください **!重要!**

  ① ページ上部の **Kernel** メニューから「**Change Kernel ...** 」を選ぶ

  ② ポップアップ画面から「**gssm2024:Python**」を選択し、「**Select**」を押す

  ③ 右上隅にカーネル名「**gssm2024:Python**」が表示されていることを確認する

演習 ― テキスト解析 (1)

- 形態素解析を行う (コマンドライン実行と同じ形式)

3.1 MeCab を使う

(1) そのまま出力してみる

①

```
import MeCab

tagger = MeCab.Tagger("-r ../tools/usr/local/etc/mecabrc")
print(tagger.parse("今日はいい天気です"))
```

```
今日	名詞,副詞可能,*,*,*,*,今日,キョウ,キョー
は	助詞,係助詞,*,*,*,*,は,ハ,ワ
いい	形容詞,自立,*,*,形容詞・イイ,基本形,いい,イイ,イイ
天気	名詞,一般,*,*,*,*,天気,テンキ,テンキ
です	助動詞,*,*,*,特殊・デス,基本形,です,デス,デス
EOS
```

① セルをクリックして選択し、再生ボタンを押す

- この方法では、コマンドライン実行した場合と同じ形式で出力されます
- ただし、テキスト解析では、**テキストを数値化し、統計処理を行う**必要があります
- そこで、**統計処理で扱いやすい DataFrame 型**(テーブル形式)に格納します →次ページ

演習 ― テキスト解析 (1)

- 形態素解析を行う (DataFrame 型に格納する)

②

```python
import pandas as pd

node = tagger.parseToNode("今日はいい天気です")
features = []
while node:
    features.append(node.feature.split(','))
    node = node.next

columns = [
    "品詞", "品詞細分類1", "品詞細分類2", "品詞細分類3", "活用型", "活用形", "基本形",
    "読み", "発音",
]
pd.DataFrame(features, columns=columns)
```

[2]:

	品詞	品詞細分類1	品詞細分類2	品詞細分類3	活用型	活用形	基本形	読み	発音
0	BOS/EOS	*	*	*	*	*	*	*	*
1	名詞	副詞可能	*	*	*	*	今日	キョウ	キョー
2	助詞	係助詞	*	*	*	*	は	ハ	ワ
3	形容詞	自立	*	*	形容詞・イイ	基本形	いい	イイ	イイ
4	名詞	一般	*	*	*	*	天気	テンキ	テンキ
5	助動詞	*	*	*	特殊・デス	基本形	です	デス	デス
6	BOS/EOS	*	*	*	*	*	*	*	*

② セルをクリックして選択し、再生ボタンを押す

- この方法では、形態素解析器の出力を**統計処理で扱いやすい DataFrame 型** (テーブル形式) に格納しています

練習：入力文「今日はいい天気です」の内容を変更して、形態素解析(②)を行った結果を確認してください

演習 ― テキスト解析 (1)

● 係り受け解析を行う (コマンドライン実行と同じ形式)

4.1 CaboCha を使う

(1) そのまま出力してみる

①
```
import CaboCha

cp = CaboCha.Parser("-r ../tools/usr/local/etc/cabocharc")
tree = cp.parse("今日はいい天気です")
print(tree.toString(CaboCha.FORMAT_LATTICE))
```

```
* 0 2D 0/1 -1.041733
今日	名詞,副詞可能,*,*,*,*,今日,キョウ,キョー
は	助詞,係助詞,*,*,*,*,は,ハ,ワ
* 1 2D 0/0 -1.041733
いい	形容詞,自立,*,*,形容詞・イイ,基本形,いい,イイ,イイ
* 2 -1D 0/1 0.000000
天気	名詞,一般,*,*,*,*,天気,テンキ,テンキ
です	助動詞,*,*,*,特殊・デス,基本形,です,デス,デス
EOS
```

① セルをクリックして選択し、再生ボタンを押す

- この方法では、コマンドライン実行した場合と同じ形式で出力されます
- ただし、**係り元**や**係り先**の関係を把握するには、この出力形式でも、表形式でも直感的ではありません
- そこで、**係り受け関係を確認し易いツリー形式**で出力します →次ページ

演習 ― テキスト解析 (1)

- 係り受け解析を行う (係り受けペアを抽出する)

②

```python
# 構文木(tree)からチャンクを取り出す
def get_chunks(tree):
    chunks = {}
    key = 0
    for i in range(tree.size()):
        tok = tree.token(i)
        if tok.chunk:
            chunks[key] = tok.chunk
            key += 1
    return chunks


# チャンク(chunk)から表層形を取り出す
def get_surface(chunk):
    surfa
    beg =
    end =
    for
        t
        s
    retu
```

③

```python
tree = cp.parse("今日はいい天気です")
chunks = get_chunks(tree)

for from_chunk in chunks.values():
    if from_chunk.link < 0:
        continue
    to_chunk = chunks[from_chunk.link]

    from_surface = get_surface(from_chunk)
    to_surface = get_surface(to_chunk)

    print(from_surface, '->', to_surface)
```

```
今日は -> 天気です
いい -> 天気です
```

← 繰り返し呼ばれる処理などをまとめて関数として定義したもの

② セルをクリックして選択し、再生ボタンを押す (③より前に一度実行しておく)

③ セルをクリックして選択し、再生ボタンを押す

- この方法では、係り受け解析器の出力を**係り元と係り先の関係**を持つ単語のペアを抽出しています

練習: 入力文「今日はいい天気です」の内容を変更して、係り受け解析(③のみ)を行った結果を確認してください

テキスト解析 (2)

(再掲) テキストマイニングの手順

- データをよく知る
  - データ件数や構成比を集計 → データを理解する
    - 旅行目的別の人気エリアは?
    - 同伴者別の人気エリアは?
    - 数値評価による人気エリアの差異は?
- テーマを設定する
  - 解決すべき課題を決める → 分析目的を明確にする
    - 数値評価が低い原因は?
    - 高評価の施設に学ぶ改善点は?
- テキスト分析に取り組む
  - これら課題を解決するために、テキスト分析を実施

テキスト分析の手順

①自然言語処理によりテキストを数値化する → ②統計解析や可視化を行う → ③結果を読み解きながら解析のための辞書を編纂する → 分析のサイクルを回していく(①へ)

(参考) KH Coder とは

2023年12月 **無償公開** が **終了** しました

- 社会調査データを分析する目的で開発されたフリー(~~ﾕｰｻﾞｰ~~)のツール
  - 高機能かつ~~商用可能~~でフリー
  - Rを用いた多変量解析と可視化
  - 実装されている分析手法
    - 階層的クラスター分析
    - 多次元尺度構成法(MDS)
    - 対応分析
    - 共起ネットワーク
    - 自己組織化マップ
    - 文書のクラスター分析
    - トピックモデル (LDA)

論文検索サービスも提供 → http://khcoder.net/bib.html

KH Coderを用いた研究事例のリスト ◀ 6135件

※2023/6/16 現在

→1646→2042→2695→3741件→4554件→昨年5355件→6135件)

(参考) 2023年12以降のライセンスや料金

2023年12月 **無償公開** が **終了** しました

	Starting	Base	Master	Pro
通常ライセンス 販売価格(税込)	無料	59,950	196,900	396,000
アカデミック・ライセンス 販売価格(税込)	無料	24,750	69,850	–
インストール可能台数 ※ライセンス保有者が管理するPCに限る	1台	2台	2台	2台

	Starting	Base	Master	Pro
無料で分析を始められる 一部機能制限ありの公開テスト(beta)版 ・ 分析対象はデータファイルの最初の100件目まで ・ 強制抽出語と無視する語の指定はそれぞれ1語のみ ・ 分析対象の品詞タイプを選択できない	○	–	–	–
機能制限なしの正式版 外部プラグイン「文錦@シリーズ」購入による機能追加も可能	–	○	○	○
アップデートのサブスクリプション (1年間有効) KH Coderのバージョンアップに追従できる便利なバッチを提供	–	○	○	○
インストール／セットアップのサポート KH Coderの導入段階のトラブルシューティング	–	–	○	○
セミナー受講券 (1年半以内有効) 「計量テキスト分析公式セミナー」初級編・ステップアップ編に各1回参加可	–	–	1名	1名
使い方のQ&A対応 使い方に関する困り事やご質問にメールでサポート	–	–	–	6回/年

出典: https://www.screen.co.jp/as/solution/khcoder

(参考) 2023年12以降のライセンスや料金

2023年12月 **無償公開** が **終了** しました

	Starting	Base	Master	Pro
通常ライセンス 販売価格(税込) ※下段は、バージョンアップに追従できるアップデートのサブスクリプション費用 (1年間)	無料	59,950 (43,780)	196,900 (43,780)	396,000 (43,780)
アカデミック・ライセンス 販売価格(税込) ※下段は、バージョンアップに追従できるアップデートのサブスクリプション費用 (1年間)	無料	24,750 (18,480)	69,850 (18,480)	–
インストール可能台数 ※ライセンス保有者が管理するPCに限る	1台	2台	2台	2台

	Starting	Base	Master	Pro
無料で分析を始められる 一部機能制限ありの公開テスト(beta)版 ・分析対象はデータファイルの最初の100件目まで ・強制抽出語と無視する語の指定はそれぞれ1語のみ ・分析対象の品詞タイプを選択できない	○	–	–	–
機能制限なしの正式版 外部プラグイン「文錦@シリーズ」購入による機能追加も可能	–	○	○	○
アップデートのサブスクリプション (1年間有効) KH Coderのバージョンアップに追従できる便利なバッチを提供	–	○	○	○
インストール／セットアップのサポート KH Coderの導入段階のトラブルシューティング	–	–	○	○
セミナー受講券 (1年半以内有効) 「計量テキスト分析公式セミナー」初級編・ステップアップ編に各1回参加可	–	–	1名	1名
使い方のQ&A対応 使い方に関する困り事やご質問にメールでサポート	–	–	–	6回/年

出典: https://www.screen.co.jp/as/solution/khcoder

(参考) KH Coder — 分析手法 (1)

2023年12月 **無償公開** が **終了** しました

共起ネットワーク

抽出語またはコードを用いて、出現パターンの似通ったものを線で結んだ図、すなわち共起関係を線（edge）で表したネットワークを描く機能です。

共起の程度が非常に強いものだけを線で結んだ図

やや弱い共起関係も描画に含め、自動的にグループ分け（色分け）

出現数が多い語ほど大きく、また共起の程度が強いほど太い線で描画

多次元尺度構成法（MDS）

同じく抽出語またはコードを用いての、多次元尺度構成法です。

2次元の解

New! クラスタリングと色分け

3次元の解

対応分析

同じく抽出語またはコードを用いての、対応分析です。

同時布置図

New! バブルプロット

複数の外部変数を用いた多重対応分析

分析手法	説明
共起ネットワーク	• 同時に出現した**単語**同士を**ネットワークで結んで図示**したもの • 同時に出現したかといった共起の有無を集計し、ネットワークを作成 • 関係の強さ Jaccard 係数で評価し、媒介性やグラフクラスタリングを使ってサブグラフも検出できる
多次元尺度構成法 (MDS)	• 出現パターンの似た**単語**同士を**近くに置くよう図示**したもの • 出現パターンとは、ある単語がどの文書に出現したかといった関係を単語ベクトルとして表現したもの • 似ている(＝距離が近い)の計算は Jaccard、ユークリッド、コサイン距離のいずれかで求める
対応分析 (コレスポンデンス分析)	• 出現パターンの似た**単語や外部変数**を**近くに置くよう図示**したもの • 単語と単語または外部変数が**同時に出現した頻度をクロス集計**し、相関が最大になるような2軸でプロット • PCA が元の情報をそのまま可視化するのに対して、対応分析は似ているものを近くに表示する • 外部変数も同時にプロットできる

(参考) KH Coder — 分析手法 (2)

2023年12月 **無償公開** が **終了** しました

トピックモデル (LDA)

文書ごとにトピックの出現割合を表示したり、各トピックに高い確率で出現する語を表示できます。

探索方法の選択

トピックに出現する単語

文書ごとのトピック比率

階層的クラスター分析

抽出語の階層的クラスター分析を行い、デンドログラムを表示します。抽出語だけでなくコーディング結果（コード）についても、同じように分析を行えます。

New! デンドログラム

抽出語は出現数や品詞で選択

コードはチェックボックスで直接選択

文書のクラスター分析

文書の分類を行うクラスター分析です。

クラスター分析の結果画面

併合水準のプロット。クラスター数5付近から併合水準が急上昇。10でも少し上がっているので、この場合クラスター数は11が良いか。

文書のデンドログラム。左の棒グラフは各文書の長さをあらわす。なお、文書数が500を超える場合、デンドログラムは表示不可。

分析手法	説明
トピックモデル (LDA)	• 文書が複数のトピックを持つと仮定、文書ごとにトピックの出現割合、各トピックに高確率で出現する語を表示 • R の topicmodels パッケージに含まれる LDA 関数(ギブスサンプリング)を利用 (乱数のシードは固定) • **トピックモデル**は**教師なし学習**のため、**コーディングルール**で単語を集約するよりも客観性が高い
階層的クラスター分析	• 出現パターンの似た**単語**同士を**グルーピング(クラスタリング)**して、樹形図にしたもの • 出現パターンは、ある単語がどの文書に出現したかといった関係を単語ベクトルとして表現したもの • 似ている(=距離が近い)の計算は Jaccard、ユークリッド、コサイン距離のいずれかで求める
文書のクラスター分析	• 似た**文書**同士を**グルーピング(クラスタリング)**して、樹形図にしたもの • 各文書は,文書中に出現する単語の有無でベクトル化した文書ベクトルで表現 • 似ている(=距離が近い)の計算は Jaccard、ユークリッド、コサイン距離のいずれかで求める • いわゆる Ward法, 群平均法, 最遠隣法で階層クラスタを作成する

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● プロジェクトの作成

注1: 次回 KH Coderを起動した時は「新規」ではなく「開く」を選択します
注2: ②のファイル選択後,ここに「テキスト」等の選択項目が表示されるまで数分がかかります

①メニューから「プロジェクト」「新規」を選択 (注1)

② 「参照」をクリックして「rakuten-1000-2022-2023.xlsx」を開く

③ 「コメント」を選択 (注2)

④ 「MeCab」を選択

⑤説明「楽天トラベル」等を入力

⑥ 「OK」をクリック

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 前処理(形態素解析)の実行

注1: EXCELファイルを読み込んで分析する場合,あらかじめ「---cell---」が入力されています
注2: メニューから「前処理」「複合語の検出」を選ぶと,複合語候補の一覧を出力できます

①メニューから「前処理」「語の取捨選択」を選ぶ

②複合語など強制的に抽出したい語あればここに登録する (例:**湯畑**)

③削除語(ストープワード)など抽出したくない語あればここに登録する

④メニューから「前処理」「前処理の実行」を選ぶ

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 頻出語を確認する

①メニューから「ツール」「抽出語」「抽出語リスト」
→右下「EXCEL出力」ボタンを選択

② 「頻出150語」を選択

③ 「OK」をクリック

	A	B	C	D	E	F	G	H
1	抽出語	出現回数		抽出語	出現回数		抽出語	出現回数
2	部屋	6689		子供	661		プラン	389
3	良い	4302		過ごす	657		見える	388
4	思う	3976		家族	648		機会	387
5	利用	3481		予約	636		設備	387
6	ホテル	2831		過ごせる	626		旅館	386
7	風呂	2702		駐車	613		置く	384
8	宿泊	2649		素晴らしい	612		きれい	377
9	食事	2447		月	611		歩く	368
10	美味しい	2249		バス	610		湯	359
11	朝食	2172		丁寧	610		施設	345
12	満足	1785		アメニティ	609		無料	345
13	スタッフ	1712		清潔	556		新しい	340
14	温泉	1705		入れる	544		楽しい	335
15	対応	1603		使う	536		掃除	335
16	立地	1374		初めて	523		気持ち	328
17	行く	1334		無い	521		雰囲気	328
18	広い	1314		人	520		女性	323
19	綺麗	1193		バイキング	515		シャワー	321
20	宿	1171		嬉しい	515		建物	316
21	大変	1157		ベッド	514		高い	316
22	少し	1156		他	504		問題	316
23	残念	1155		親切	503		全体	314
24	最高	1118		種類	502		大きい	313

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 共起ネットワークの作成(1)

①メニューから「ツール」「抽出語」「共起ネットワーク」を選ぶ

② 「集計単位」として「H5」を選んで 「OK」をクリック

③品詞を選択する,例えば
名詞, 名詞B, 名詞C, 未知語,
形容詞, 形容詞B, 形容動詞, タグ

⑤ 「カラー」の箇所を「サブグラフ検出(modularity)」に変更

④ 「調整」をクリックして「強い共起関係ほど…」をチェック

(参考) KH Coder の品詞体系

2023年12月 **無償公開** が **終了** しました

表 A.1 KH Coder の品詞体系

KH Coder 内の品詞名	茶筌の出力における品詞名
名詞	名詞–一般（漢字を含む 2 文字以上の語）
名詞 B	名詞–一般（平仮名のみの語）
名詞 C	名詞–一般（漢字 1 文字の語）
サ変名詞	名詞–サ変接続
形容動詞	名詞–形容動詞語幹
固有名詞	名詞–固有名詞–一般
組織名	名詞–固有名詞–組織
人名	名詞–固有名詞–人名
地名	名詞–固有名詞–地域
ナイ形容	名詞–ナイ形容詞語幹
副詞可能	名詞–副詞可能
未知語	未知語
感動詞	感動詞またはフィラー
タグ	タグ
動詞	動詞–自立（漢字を含む語）
動詞 B	動詞–自立（平仮名のみの語）
形容詞	形容詞（漢字を含む語）
形容詞 B	形容詞（平仮名のみの語）
副詞	副詞（漢字を含む語）
副詞 B	副詞（平仮名のみの語）
否定助動詞	助動詞「ない」「まい」「ぬ」「ん」
形容詞（非自立）	形容詞－非自立（「がたい」「つらい」「にくい」等）
その他	上記以外のもの

出典: KH Coder 3 リファレンス・マニュアル

注: どの品詞を選択すべきかは、分析対象のデータや分析目的により異なります。

分析結果を確認しながら、適宜、適切な品詞選択を検討することが重要です。

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 前後文脈を確認する

①メニューから「ツール」「抽出語」「KWICコンコーダンス」を選ぶ

②検索したい語 (例:「風呂」) を入力して Enter キー

ダブルクリックで,さらに広い範囲の文脈を表示

③「集計」をクリックするとコロケーション統計(右)を開く

N	抽出語	品詞	合計	左合計	右合計	左5	左4	左3	左2	左1	右1	右2	右3	右4	右5	スコア
1	広い	形容詞	190	41	149	8	5	13	13	2	1	104	20	18	6	81.0
2	良い	形容詞	222	77	145	40	9	14	13	1	6	48	43	21	27	77.4
3	最高	名詞	105	13	92	5	3	3	2			44	12	22	8	42.8
4	部屋	名詞	439										22	20	18	172.7
5	トイレ	名詞	117										1	4	4	60.2
6	露天風呂	名詞	87										9	13	13	30.6
7	風呂	名詞	130										31	14	20	35.6

「右1」は右側の1つ目(=直後)に出現していた回数

「広い」は「風呂」の2語後に 104 回出現

④表示する語の品詞を選択 (例: 形容詞,形容詞B,形容動詞)

⑤「右合計」でソート

注: 共起ネットワーク上で「風呂」をクリックすると①②と同じ操作となります (V3以降)

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 外部変数を利用する

①メニューから「ツール」「外部変数と見出し」を開く

④「特徴語」「一覧(Excel形式)」を選択

	A	B	C	D	E	F	G	H	I	J	K
1											
2	01_登別			02_草津			03_箱根			04_道後	
3	風呂	.115		湯畑	.327		美味しい	.136		温泉	.109
4	温泉	.107		温泉	.136		露天風呂	.134		立地	.082
5	美味しい	.094		風呂	.126		風呂	.116		最高	.066
6	良い	.093		宿	.120		部屋	.109		広い	.063
7	バイキング	.090		美味しい	.102		良い	.106		浴場	.059
8	残念	.078		良い	.100		温泉	.102		よい	.058
9	ない	.077		部屋	.096		宿	.097		フロント	.057
10	夕食	.076		最高	.090		スタッフ	.096		大変	.057
11	種類	.075		夕食	.085		夕食	.095		夕食	.055
12	露天風呂	.074		ない	.074		ない	.083		便利	.055
13	05_湯布院			06_札幌			07_名古屋			08_東京	
14	宿	.180		ホテル	.092		ホテル	.086		駅	.102
15	美味しい	.144		立地	.077		便利	.072		ホテル	.086
16	露天風呂	.135		便利	.077		駅	.070		便利	.078
17	風呂	.127		綺麗	.071		綺麗	.069		立地	.077
18	温泉	.124		浴場	.070		フロント	.066		近い	.071
19	最高	.114		フロント	.065		立地	.065		綺麗	.064
20	スタッフ	.110		広い	.063		近い	.059		快適	.063
21	家族	.104		快適	.056		アメニティ	.056		コンビニ	.059
22	部屋	.099		駅	.056		快適	.055		フロント	.055
23	良い	.097		ベッド	.055		コンビニ	.051		アメニティ	.052
24	09_大阪			10_福岡							
25	ホテル	.108		ホテル	.090						
26	駅	.096		便利	.087						
27	便利	.080		立地	.082						
28	立地	.074		駅	.0						
29	綺麗	.072		フロント							
30	フロント	.067		綺麗							
31	快適	.064		トイレ							
32	広い	.064		コンビ							
33	近い	.064		よい							
34	アメニティ	.054		快適							

各エリアの特徴語を10件ずつ
一覧 (数値は Jaccard係数)

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 共起ネットワークの作成(2)

①メニューから「ツール」「抽出語」「共起ネットワーク」を選ぶ

② 「集計単位」として「H5」を選んで 「OK」をクリック

③品詞を選択する,例えば
名詞, 名詞B, 名詞C, 未知語,
形容詞, 形容詞B, 形容動詞, タグ

④ 「共起パターンの変化を探る(相関)」にチェックを入れて数値変数に「総合」を選択

赤なら高評価で多く共起し
青なら低評価で多く共起している

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 共起ネットワークの作成(3)

①メニューから「ツール」「抽出語」「共起ネットワーク」を選ぶ

② 「集計単位」として「H5」を選んで 「OK」をクリック

④「語-外部変数・見出し」を選択し,「カテゴリー」を選ぶ

③品詞を選択する,例えば
名詞, 名詞B, 名詞C, 未知語,
形容詞, 形容詞B, 形容動詞, タグ

⑤ 「調整」をクリックして「強い共起関係ほど･･･」をチェック

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 対応分析による探索(1)

①メニューから「ツール」「抽出語」「対応分析」を選ぶ

②「H5」になっていることを確認

③「抽出語×外部変数」「**カテゴリー**」「エリア」を選択

④「OK」をクリック

原点(0.0) から見て離れている語ほど特徴的

原点(0,0) 付近にはあまり特徴のない語が集まるる

原点(0,0)

⑤「調整」をクリックして「バブルプロット」をチェック

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 対応分析による探索(2)

①メニューから「ツール」「抽出語」「対応分析」を選ぶ

②「H5」になっていることを確認

③「抽出語×外部変数」「**カテゴリー**」「**総合**」を選択

④「OK」をクリック

左右に**カテゴリー**、上下に**総合評価**が分かれて布置

「温泉」はレジャーに、「立地」はビジネスに多く出現したと見られる

原点(0,0)

⑤「調整」をクリックして「バブルプロット」をチェック

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● コーディングルール (語ではなくコンセプトを数える方法)

①メニューから「ツール」「文書」「文書検索」を選ぶ

② 「参照」をクリックして「coding-rule.txt」を開く

③ダブルクリック

④「H5」を選択

ダブルクリックで,さらに広い範囲の文脈を表示可能

coding-rule.txt の中身

*ポジ
良い or 美味しい or 広い or 多い or 素晴らしい or 嬉しい or 気持ちよい or 楽しい or 近い or 大きい or 気持ち良い or 温かい or 早い or 優しい or 新しい or 暖かい or 快い or 明るい or 美しい or 可愛い

*ネガ
古い or 無い or 高い or 悪い or 小さい or 狭い or 少ない or 寒い or 遅い or 熱い or 欲しい or 暑い or 冷たい or 遠い or 臭い or 暗い

*風呂1-2
<>風呂-->1 | <>風呂-->2

*風呂4-5
<>風呂-->4 | <>風呂-->5

外部変数

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● 対応分析による探索(3)

①メニューから「ツール」「コーディング」「対応分析」を選ぶ

②「文」になっていることを確認

③見たい「コード」を選択

④「抽出語×外部変数」「カテゴリー」「エリア」を選択

⑤「OK」をクリック

原点(0,0)

上下(寄与率が低い)に**ポジネガ**、左右に**風呂**と**エリア**が分かれて布置

「風呂」の低評価はビジネスに多く出現したと見られる

⑥「調整」をクリックして「バブルプロット」をチェック

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● クロス集計

①メニューから「ツール」「コーディング」「クロス集計」を選ぶ

注: プロット左側のラベルは表示されません

②「参照」をクリックして「coding-rule.txt」を開く

⑤「集計」をクリック

④「エリア」を選択

③「H5」(=セル)を選択

「**」なら誤差の範囲を超える統計的な差がある

⑥「すべて」をクリック

(参考) KH Coder の使い方

2023年12月 **無償公開** が **終了** しました

● トピックモデルによる分析

①メニューから「ツール」「文書」「トピックモデル」「トピックの推定」を選ぶ

レジャーとビジネスで注目している観点(=トピック)が異なる

KH Coder の解析・分析手法

KH Coder の特徴

- 「単語と単語」、「外部変数と単語」の**関係に注目した分析**が得意
  - 特徴的な単語を見つける
    - 特定の文書に特徴的な単語を見つける → TF・IDF
      → その文書に特に頻出するが、他の文書ではそれほどではない

- **特徴的な関係を見つける**
  - 関係性のある単語と単語と見つける → **共起ネットワーク(Jaccard係数)**
    例) 単語:「風呂」と 単語:「広い」に関係がありそう
  - 関係性の強い単語と外部変数を見つける → **対応分析(カイ2乗値)**
    例) 外部変数:「レジャー」と 単語:「風呂」に関係がありそう

KH Coder で使われるデータ表

- 「文書-抽出語」表
  - 【行】ある文中に出現する単語の数を要素とする文ベクトル
  - 【列】全文中に出現する単語の数を要素とする単語ベクトル

「文書-抽出語」頻度表

h5	id	部屋	ホテル	風呂	スタッ	温泉	立地	最高	フロン	浴場	露天風	夕食	感じ	子供	家族	バス	アメニ	バイキ	ベット	種類	トイレ	綺麗	大変	残念	便利	快適	丁寧	清潔	親切	良い	美味し	広い	多い	近い	素晴ら	無い	嬉しい	古い	ない	よい
1	1	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	2	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
3	3	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
4	4	1	0	1	0	0	0	1	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	0	0	0	0	0	0
5	5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0
6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	0	1	0	0	0	0	0	0	1	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	0	0	0	0
8	8	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
9	9	1	0	0	1	1	0	0	0	0	0	0	[illegible]	0	0	0	0	0	0	0	0	0	2	1	[illegible]	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	10	1	0	0	0	0	0	0	0	0	0	0	[illegible]	0	0	0	0	0	0	0	0	0	0	0	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

「抽出語-抽出語」共起頻度表 (共起ネットワーク)

	部屋	良い	ホテル	風呂	美味し	ない	スタッ	温泉	よい	立地	広い	綺麗だ	宿	大変だ	残念だ	最高
部屋	4568	1961	1091	1273	1068	1012	783	706	676	691	955	817	461	512	635	526
良い	1961	3701	892	1026	896	692	712	638	339	701	558	494	406	439	468	356
ホテル	1091	892	2041	391	354	506	411	235	267	356	291	339	72	207	283	198
風呂	1273	1026	391	2143	598	503	376	359	333	317	414	280	334	277	325	294
美味しい	1068	896	354	598	1962	379	432	429	229	215	299	255	300	308	249	270
ない	1012	692	506	503	379	1629	337	296	269	289	260	185	202	193	341	159
スタッフ	783	712	411	376	432	337	1411	275	216	201	182	184	206	214	203	175
温泉	706	638	235	359	429	296	275	1298	210	176	221	120	293	197	182	246
よい	676	339	267	333	229	269	216	210	1205	238	167	135	124	124	158	113
立地	691	701	356	317	215	289	201	176	238	1328	167	186	120	139	157	243
広い	955	558	291	414	299	260	182	221	167	167	1186	209	122	153	152	128
綺麗だ	817	494	339	280	255	185	184	120	135	186	209	1132	76	120	119	117

「外部変数-抽出語」クロス集計表 (対応分析)

	部屋	良い	ホテル	風呂	美味し	ない	スタッ	温泉	よい	立地	広い	綺麗だ	宿	大変だ	残念だ	最高
A_レジャー	2398	2046	757	1535	1430	880	888	1188	631	518	631	459	769	652	609	697
B_ビジネス	2170	1655	1284	608	532	749	523	110	574	810	555	673	57	355	436	294
01_登別	447	409	194	323	255	187	148	222	114	38	125	88	76	120	148	124
02_草津	488	434	181	352	274	180	154	275	117	155	114	109	195	136	122	165
03_箱根	548	436	134	326	355	202	212	212	133	57	140	93	161	137	150	82
04_道後	416	349	191	181	174	130	135	225	137	176	129	87	59	108	102	123
05_湯布院	499	418	57	353	372	181	239	254	130	92	123	82	278	151	87	203
06_札幌	452	346	255	121	129	151	114	38	103	166	129	142	10	87	92	74
07_名古屋	434	310	241	116	97	144	102	18	133	141	84	138	11	58	89	42
08_東京	441	338	240	106	99	131	99	12	104	166	98	128	14	69	82	56
09_大阪	431	317	297	135	88	162	93	20	121	161	132	144	9	62	82	62
10_福岡	412	344	251	130	119	161	115	22	113	176	112	121	13	79	91	60

Jaccard 係数

- Jaccard 係数は、共起の強さを測る尺度 (KH Coderで標準的に使用)
  - どちらの語も含まない文書を無視 →言語のようなスパースデータ分析に向いている

$$\text{Jaccard 係数} = \frac{c}{a+b+c}$$

抽出語「風呂」の場合:

c = 323 (“共起”列の値)

a = 2143 (“全体”列の値) − 323 = 1820

b = (323 / 0.323) − 323 = 677

*1 括弧内はデータ全体に対する割合(前提確率)　*2 括弧内は「10_登別」を付与したデータに対する割合(条件付き確率)

カイ2乗値

- カイ2乗値は「無関係でない」度合いを測る尺度 → カテゴリと変数間の関連性を測定

$$\text{カイ2乗値} = \frac{\left(\text{観測度数}_{ij} - \text{期待度数}_{ij}\right)^2}{\text{期待度数}_{ij}}$$

「**観測度数**」: カテゴリと変数に従ってクロス集計された度数
「**期待度数**」: 変数が互いに独立している場合に期待される度数
「**観測度数 − 期待度数**」: 実際の度数と独立と期待される度数の差

- カイ2乗値も大きい → カテゴリと変数間の関係が**期待より強い** を示す

クロス集計表 (観測度数)

	A	B	C	D	E	合計
地質学	3	19	39	14	10	85
生物化学	1	2	13	1	12	29
科学	6	25	49	21	29	130
動物学	3	15	41	35	26	120
物理学	10	22	47	9	26	114
工学	3	11	25	15	34	88
微生物学	1	6	14	5	11	37
植物学	0	12	34	17	23	86
統計学	2	5	11	4	7	29
数学	2	11	37	8	20	78
合計	31	128	310	129	198	796

期待度数

	A	B	C	D	E	合計
地質学	3.310	13.668	33.103	13.775	21.143	85.000
生物化学	1.129	4.663	11.294	4.700	7.214	29.000
科学	5.063	20.905	50.628	21.068	32.337	130.000
動物学	4.673	19.296	46.734	19.447	29.849	120.000
物理学	4.440	18.332	44.397	18.475	28.357	114.000
工学	3.427	14.151	34.271	14.261	21.889	88.000
微生物学	1.441	5.950	14.410	5.996	9.204	37.000
植物学	3.349	13.829	33.492	13.937	21.392	86.000
統計学	1.129	4.663	11.294	4.700	7.214	29.000
数学	3.038	12.543	30.377	12.641	19.402	78.000
合計	31.000	128.000	310.000	129.000	198.000	796.000

観測度数-期待度数

	A	B	C	D	E	合計
地質学	-0.310	5.332	5.897	0.225	-11.143	0.000
生物化学	-0.129	-2.663	1.706	-3.700	4.786	0.000
科学	0.937	4.095	-1.628	-0.068	-3.337	0.000
動物学	-1.673	-4.296	-5.734	15.553	-3.849	0.000
物理学	5.560	3.668	2.603	-9.475	-2.357	0.000
工学	-0.427	-3.151	-9.271	0.739	12.111	0.000
微生物学	-0.441	0.050	-0.410	-0.996	1.796	0.000
植物学	-3.349	-1.829	0.508	3.063	1.608	0.000
統計学	0.871	0.337	-0.294	-0.700	-0.214	0.000
数学	-1.038	-1.543	6.623	-4.641	0.598	0.000
合計	0.000	0.000	0.000	0.000	0.000	0.000

カイ2乗値

	A	B	C	D	E	合計
地質学	0.029	2.080	1.050	0.004	5.873	9.036
生物化学	0.015	1.521	0.258	2.913	3.176	7.882
科学	0.173	0.802	0.052	0.000	0.344	1.373
動物学	0.599	0.957	0.703	12.438	0.496	15.194
物理学	6.964	0.734	0.153	4.859	0.196	12.906
工学	0.053	0.702	2.508	0.038	6.700	10.001
微生物学	0.135	0.000	0.012	0.166	0.351	0.663
植物学	3.349	0.242	0.008	0.673	0.121	4.393
統計学	0.671	0.024	0.008	0.104	0.006	0.814
数学	0.354	0.190	1.444	1.704	0.018	3.710
合計	12.343	7.252	6.196	22.899	17.282	65.972

その他の尺度

- **出現パターンが似てる** を測る ＝ ユークリッド距離、コサイン距離
  - 1つひとつの文が長く、各文中での語の出現回数の大小が重要なケースに向く
    (語が1回出現したか、10回出現したかを区別したい)

ユークリッド距離	コサイン距離
サイズ(出現回数の大小)の差までも見る場合向き	傾きが似ているかどうかだけを見る場合向き
$d(\boldsymbol{x},\mathbf{y}) = \sqrt{\sum (x_i - y_i)^2}$	$d(\boldsymbol{x},\mathbf{y}) = \dfrac{\sum x_i y_i}{\sqrt{\sum {x_i}^2 \sum {y_i}^2}}$

※ $\mathbf{x}, \mathbf{y}$ はそれぞれの単語ベクトル (単語の出現パターン)

共起ネットワーク

コード: day-4.ipynb

- 共起頻度の高いエッジを残す

出現回数の高い語と共起するだけで上位に来てしまう

- Jaccard 係数が上位のエッジを残す

対応分析 (コレスポンデンス分析)

コード: day-4.ipynb

● 対応分析は、カイ2乗値を利用して、関係の強さを強調して表現できる

出現回数の高い語を強調 (原点から遠い位置に布置)

対応分析

カイ2乗値

共起頻度

特異値分解

- 第2軸の選び方
  →特徴的な軸
- 強調するもの
  →関係の強さ

- 第2軸の選び方
  →特徴的な軸
- 強調するもの
  →出現回数

直交性ではなく、特徴的な2軸を選択する

PCA

- 第2軸の選び方
  →第1軸と直行
- 強調するもの
  →関係の強さ

- 第2軸の選び方
  →第1軸と直行
- 強調するもの
  →出現回数

出現頻度より、関係の強さを強調

特異値分解

- 特異値分解　$X = USV^T$

- S の特異値が小さいものを削る

演習 ― テキスト解析 (2)

- Jupyter Lab を起動して、教材を最新化(pull)してください

- **教材を最新化**する

① 「**Git**」 ボタンを押す

② 「**Refresh**」 ボタンを押す

③ もし、競合がある場合(Changedが0でない場合)、**対象のファイルを手動で退避**した後、「**Discard changes**」 ボタンを押して変更を破棄する

④ 「**Pull latest changes**」 を押す

⑤ 画面の右下に「Successfully published」が表示されること確認する

演習 ― テキスト解析 (2)

- day-3-3.ipynb を開いてください

  ① 画面左の **File Browser** から ① **notebooks** をフォルダを開く (既に開いている場合はスキップ)

  ② 次に **day-3-3.ipynb** ノートブックを開く

演習 ― テキスト解析 (2)

- カーネル **gssm2024:Python** を選択してください **!重要!**

  ① ページ上部の **Kernel** メニューから「**Change Kernel ...** 」を選ぶ

  ② ポップアップ画面から「**gssm2024:Python**」を選択し、「**Select**」を押す

  ③ 右上隅にカーネル名「**gssm2024:Python**」が表示されていることを確認する

演習 ― テキスト解析 (2)

- KH Coder のテキスト解析&分析 (**day-3-3.ipynb**)

演習:

① ページ上部の Run メニューから「Run All Cells」を選ぶ

この後、Step-by-step で解説します

day 3 – レポート課題

- 以下を PDF ファイルで提出 してください
  - ノートブック **day-3-3.ipynb** の末尾にある「**【演習】2021~2022 データセット**」に従って、別のデータセット (rakuten-1000-**2021-2022**.xlsx.zip) で作図した 「共起ネットワーク図」と「対応分析プロット」のキャプチャ

※ 何らかの事情で上記のキャプチャを提出できない場合、本日の講義の感想を文章で記述してください

レポート形式	提出先	期限
PDF	manaba	次回～18:20