

# 金丸 洋平

店舗販売員

## 自己紹介

現在、小売りの販売員として日々商品を販売しています。

売り場のレイアウト変更や商品の陳列が得意です。とてもキレイで気持ちのいい売り場に仕上がっていると思います。

売上前年比は直近3年の平均で115%達成。

この先、新しい仕事にチャレンジするために自己学習を続けています。

## スキル

プログラミング(Python,R,その他)  
統計学基礎知識  
データ集計・探索的データ分析

## 趣味

景観を眺めること  
音楽を聴くこと  
データ、プログラミングの勉強

## 連絡先

携帯：080-3001-4457  
Email：YouNat.Kana@outlook.com  
skyrocket\_xvii@icloud.com

## 参考書籍

### 統計学・数学関連

統計学入門 はじめての統計学 ゼロから学ぶ統計解析 基礎統計学Ⅰ  
統計学基礎(2級対応) 統計学(1級対応) 現代数理統計学の基礎  
ゼロから学ぶ微積分 ゼロから学ぶ線形代数 最適化数学 多変量解析がわかる  
多変量解析のはなし バイオサイエンスの統計学 医療を志す人のための基礎数学  
やさしく学ぶ機械学習を理解するための数学のきほん  
人工知能プログラミングのための数学 その問題、数理モデルが解決します

### 機械学習・AI関連

入門Python3 Pythonではじめる機械学習  
Pythonではじめるデータラングリング ゼロから作るDeep Learning  
Pythonクロウリング&スクレイピング Pythonによるデータ分析入門  
Pythonデータサイエンスハンドブック あたらしい機械学習の教科書  
機械学習のエッセンス データを読み解くアルゴリズムの技法 Kaggleで勝つ  
Scikit-learnによる実践機械学習 RとPythonで学ぶデータサイエンス&機械学習  
Python機械学習プログラミング Python実践100本ノックシリーズ4冊

### R関連その他

Rによるやさしい統計学 Rプログラミング入門 Rではじめるデータサイエンス入門  
データ解析のための統計モデリング入門 Rで学ぶデータサイエンス(一般化線形モデル)  
Rによる多変量解析入門 効果検証入門 SAS入門 実用SAS生物統計ハンドブック  
Rによるテキストマイニング入門 実践Rによるテキストマイニング

### データ分析関連

やってみようテキストマイニング 社会調査のための計量テキスト分析  
調査系論文の読み方 戦略的データサイエンス入門 データ分析のための数理モデル入門  
分析者のためのデータ解釈学入門 マーケティング・リサーチ入門  
社会調査の考え方(上)

## 参考サイト

我楽多頓陳館(<http://www.snap-tck.com/room04/c01/stat/stat.html>)  
米国データサイエンティストのブログ(<https://datawokagaku.com/>)  
米国データサイエンティストがやさしく教えるデータサイエンスのためのPython講座(udemy)  
キカガク流脱ブラックボックスコース(<https://www.kikagaku.ai/>)  
SIGNATE Quest(<https://quest.signate.jp/quests>) 9課題/全16課題 実施済み  
DjangoBrothers([https://djangobrothers.com/tutorials/blog\\_app/](https://djangobrothers.com/tutorials/blog_app/))  
⇒DjangoBrosBook購入 実施済み  
SQL攻略(<http://sql.main.jp/>)  
TECHPROjin(<https://tech.pjin.jp/blog/2016/12/05/sql練習問題-一覧まとめ/>)  
⇒全75問実施済み

# 学習まとめ

---

## その1. 社会やビジネスにおけるさまざまなデータと指標を知る。

- ・量的データ：間隔尺度、比例尺度
  - ・質的データ：名義尺度、順序尺度
  - ・相関：相関、自己相関、偏相関、疑似相関など
  - ・データ標準化：MinMax、Z得点、中心化(平均値を引く)など
  - ・交絡と中間因子
  - ・多重共線性：VIF
  - ・データのバイアス：セレクションバイアス、選択バイアスなど
  - ・データのバイアスを取り除く：RCT、OVB(脱落バイアス)、傾向スコア、DID、回帰不連続など
- 

## その2. データ前処理を実施する。

- ・サンプルサイズの確認(分析する上で十分なサイズであること)
  - ・表記ゆれの補正
  - ・欠損値、外れ値の確認と処置(グラフによる可視化)
  - ・質的データの処置(ダミー変数化)
  - ・各データの単位を揃える(無名単位化)
  - ・必要であれば、新しいデータを作成する(データ+データ、データ×データ、(データ×データ)/総データ数など)
- 

## その3. データを正しく解釈して、目的に適う分析手法を選択する。

- ・回帰と分類  
線形・非線形回帰モデル、決定木モデル、勾配ブースティングモデル、ロジスティック回帰モデル、クラスタリング、ディープラーニングなど
  - ・多変量解析  
線形・非線形重回帰、パス解析、因子分析、共分散構造分析、一般化線形モデル(対数線形モデル、ロジスティック回帰)、クラスタリング、コレスポンデンス分析など
- 

## その4. 分析結果を評価する。

- ・各分析モデルについての評価方法  
予測の正解率、分類の正解率(適合率、再現率、調和平均)、可視化(ROC曲線)、偏回帰係数の検定、因子負荷の検定、適合度の検定(カイ二乗値)、モデルの当てはまりの良さ(AIC、BIC)、構成概念の信頼性と妥当性の評価など
- 

## その5. 分析結果を考察する。

オッズ比(ロジット関数)による効果指標、より重要な変数の特定、因果関係の特定 ⇒ 新たなKPI策定、経営リソースの再配分など

↓↓↓↓

"初手として、その1～その3を実務レベルで修得する。"

"リサーチデザインについて学習する。"

# 実践 1．頻出単語の抽出

データの種類：質的データ(非構造化データ)  
データの対象：東野 圭吾著作「白鳥とコウモリ」の読書レビュー文(全539件)  
具体的方法：

- ① Pythonを用いたWebスクレイピングにより読書レビュー文を取得。CSV形式でローカルPC環境に保存。
- ② CSVファイルの文字コードをUTF8からSJISに変換。
- ③ Rを用いてCSVファイルを読み込み。以下の手順はすべてRを用いて実施。
- ④ RMeCabライブラリのdocDF関数によりCSVファイルの形態素解析を実行。
- ⑤ 品詞を「名詞、動詞、形容詞」に限定する。
- ⑥ 名詞、動詞、形容詞それぞれの品詞細分類の内容を確認して、さらに絞り込む。
- ⑦ 各品詞ごとの頻出単語上位10個を表示。

出力表示：

TERM	POS1	POS2	book_reviews_csv_sjis.csv	TERM	POS1	POS2	book_reviews_csv_sjis.csv	TERM	POS1	POS2	book_reviews_csv_sjis.csv
事件	名詞	一般	377	する	動詞	自立	1009	ない	形容詞	自立	187
東野	名詞	固有名	320	読む	動詞	自立	411	面白い	形容詞	自立	127
被害	名詞	一般	251	いる	動詞	非自立	348	良い	形容詞	自立	50
作品	名詞	一般	245	ある	動詞	自立	324	切ない	形容詞	自立	49
圭吾	名詞	固有名	234	なる	動詞	自立	320	すごい	形容詞	自立	43
家族	名詞	一般	203	思う	動詞	自立	294	深い	形容詞	自立	36
加害	名詞	一般	202	しまう	動詞	非自立	183	分厚い	形容詞	自立	34
罪	名詞	一般	175	いく	動詞	非自立	164	いい	形容詞	自立	33
人	名詞	一般	144	できる	動詞	自立	97	長い	形容詞	自立	32
犯人	名詞	一般	144	くる	動詞	非自立	94	多い	形容詞	自立	30

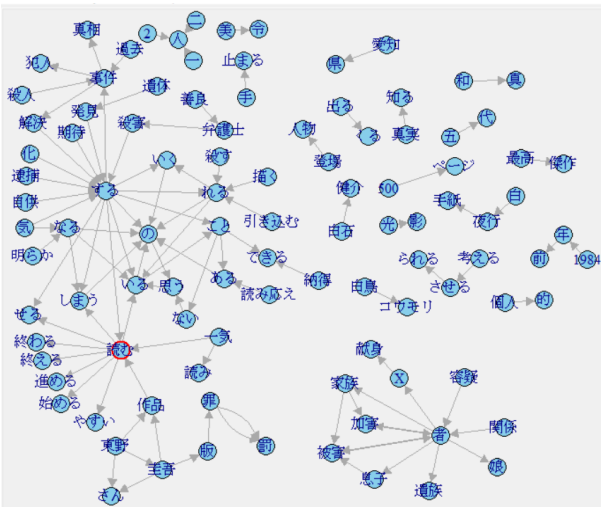
頻出単語であるため、ユニークな単語はないものの、「事件、被害、犯人、東野、圭吾、分厚い、長い」から長編ミステリー小説であることが示唆される。  
また、「面白い、良い」が頻出上位であることから、この作品の評価は高いと思う一方、「ない」が形容詞の頻出最上位であるため、もしかすると「面白くない」のかもしれない。

# 実践 2．バイグラムとネットワークグラフの作成

実践 1 と同様のデータに対して、単語同士の繋がりをグラフにする。  
具体的方法：

- 実践 1 の①～⑤までの手順は同様。
- ① RMeCabライブラリのdocDF関数の引数に「nDF = 1, N = 2」を与える。  
N = 2 ⇒ バイグラムの指定。  
nDF = 1 ⇒ バイグラムとして抽出した2つの単語を別々のカラムに出力する。
- ② 出現頻度が16以上のものに絞る。
- ③ ネットワークグラフ作成用のigraphライブラリのgraph\_from\_data\_frame関数により、バイグラムをネットワークオブジェクトに変換。
- ④ tkplot()によりネットワークグラフを描画。

出力表示：



ネットワークグラフでは、単語同士の繋がりがよくわかる。  
動詞の頻出単語で最も多かった「する」は、「逮捕、解決、自供、発見、殺害」などの単語と強い繋がりがあり、レビュー者は本作品の内容の要約を書く際に「する」を多用したことがわかる。  
また、「する」の次に多かった「読む」は、「終わる、進める、やすい、一气、作品」などの単語と強く繋がっており、本作品の感想を書く際に「読む」を多用したものと思われる。  
その他、名詞に関しては、本作品の登場人物の名前や著者の名前と作品のタイトル、著者の他作品のタイトルなどが出力されており、動詞の場合と同様に、本作品のあらすじと感想を記述する際に多用される単語であることがわかる。  
名詞の場合は、頻出単語のみでもある程度レビューの内容を推察可能だが、動詞(とくに「する」)の場合は、バイグラムの方が明確にレビューの内容を推し量ることができる。

具体的方法：

実践1の①～③までの手順は同様。

- ① 単語感情極性対応表をダウンロードして読み込み。各単語の感情極性値：-1.0～1.0 (離散値)
- ② 重複行を出力。同じ綴りと読みの単語については、それらの感情極性値の平均値を使用する。
- ③ 読み込んだレビュー文を句点ごとの一文に分割する。
- ④ `tidyverse` ライブラリの `tidyble` 関数により、データフレームに変換。それぞれの一文に対して ID 付与。
- ⑤ `RMeCab` ライブラリの `RMeCabC` 関数により、一文ごとに形態素解析を実行。
- ⑥ 形態素(単語)ごとに単語感情極性対応表の感情極性値を参照し、一文(ID)ごとにその合計を求める。
- ⑦  $X: ID$  (一文)、 $Y: EM$  (感情極性値) として、グラフを描画。

### (1)形態素解析した各単語の感情極性値(EM)のサマリー

EM	
Min. : -68.408	⇒平均値が-3.750 とマイナスになっており、最小値に関しては-68.408と極端に小さな値になっている。
1st Qu.: -5.249	極端な最小値により平均値がマイナスの値に引っ張られたかもしれない。
Median : -2.911	
Mean : -3.750	
3rd Qu.: -1.239	
Max. : 2.984	

```
em_reviews %>% filter(EM == min(EM)) %>% left_join(sent_book_reviews) %>% select(S) %>% pull()
```

[illegible]

マイナス値が大きくてもブックログにおけるレビュー全体の評価平均は4.22点と非常に高い評価を得ている作品である。小説のような文芸作品は、感情極性値とレビューアーの評価との間に正の相関があったり、負の相関があったりするのかもしれない。

実用書のレビュー文との比較をしてみるとその違いが明確になるかもしれないので、今後の課題とする。

(4)感情極性値が1より大きい文章を抽出  
em\_reviews %>% filter(EM > 1) %>% left\_join(sent\_book\_reviews) %>% select(S) %>% pull()

[1] "東野圭吾の真実探偵。矢々に展開させてもらった。"  
[2] "法廷の下の正義。良心の下正義。愛情の下正義。"  
[3] "ワザワザ設定のオオジリディ世界のワザワザディディで楽しめた。"  
[4] "でも、凄くよくできた感動的な話なんです。"  
[5] "ワザワザ展開らしかった。展開された。"  
[6] "早く映像化が望みたい！2022年のベスト10を更新しました！おすすめです。"  
[7] "面白いら、面白いらだけで。展開面白がるのみ。"  
[8] "でも、一番面白かったかと云えば？です。"  
[9] "それが白鳥で驚がコウモリワザワザ探偵が面白いので期待して読んでたが、期待通りの作品だった。"  
[10] "ワザワザ矢々のヒットワザワザ2021年3月読者ワザワザのワザワザ登場人物がみんな動機で真実品。"

(5)感情極性値が-35より小さい文章を抽出  
em\_reviews %>% filter(EM < -35) %>% left\_join(sent\_book\_reviews) %>% select(S) %>% pull()

[1] "ワザワザ東野圭吾さんの作品を読むのがこの白鳥とコウモリではじめてでした。ページ数も多かったのですがどんどん読み進めてしまいました。最初の30ページほどで犯人が逮捕された。。これから400ページどうするのか？と驚いてましたがどちらかと言うとこっちが本編って感じました裁判の結果だけを考えている刑事。弁護士と事件に違和感を覚えるあの真実を探っていく被害者。加害者の息子。娘という構図かな？あと刑事さんが出てきますね群の中としては悪とされたり正義とされたりするものもその犯人の感じの方や考え方でその構図が逆転したりっていうのもあるのかなと考えました。。結局何が問題で何が正しかったんだらうと最後まで考えさせられる内容でしたラスト辺りの読みのがれは泣きました。ワザワザ作者自ら「今後の目標はこの作品を超えることです」との意気込み。"

[2] "東野ドームの登場の図が出てくるから。犯罪は、犯人の根本犯罪からなのか。事件は、誰か犯罪人からか。犯人として捕まる。根本犯罪。事件解決の違和感に迫るプロセスが面白い根本の事件の図は、j00から400ページに渡るその記述の見た目が変わっていく様を読みながら確認するのだ事件は、実情も無いと思う強い意志をもつ女性として描かれているが、どこか無情なような。やはりラプラスの魔女の。犯罪内情を思い出すそんな危機感をほらんだ行動を起こす犯人に会おうと思って違和感をぶつけようと思った。犯人入を。無情にしようと思っていない事にも。違和感を感じるあまり自分の周囲には居ないような人物なんだろうと。名字のイメージ根本一喝き。真加害者(真加一画。グレー)白石一白被害者。あと、それで白鳥とコウモリ。か。一画通。読んだ作品にも。似た部分がある。ササザと雲の犯罪。と。その犯罪な図解一画解の事件に惹きつけようということ。犯人になるのを謝らない。自己犠牲の精神一画被害者の献身ラストには。本当の犯罪者の心理と。その犯罪を。忘れずに描いた。また。犯人犯行という行為が。どれほどあった行為であるかは。ストーリー中で説明されなければならない場合は。自分が。日帰り遠出旅行の道迷いの傍らに。その犯人と初めて手を繋いだまま帰ったことを。思い出しませんでしたそんなお話でした。"

[3] "ワザワザ登場人物全ての行動。裏付けられた心機。それを物にできる歴史に対して。伏線を描きながら解いていく物語の展開に解読されないわけがない事実だけを言々と連綿と犯人事件は。実は犯罪に類と登場人物の感情(感情的に見れば読まれること)とその展開から成り立っている少なからず読みしこの矛盾する感情に共通できるから。外では間違いないような明確を下し。時には驚愕することがあるのだから本来同じ視点。感情を持つはずがない被害者と加害者を決して交わって読みことのない白鳥とコウモリに描かれているのか真実を追求した結果。過去から現在につながる罪と罰の図解にたどり着いてしまった犯罪と事件は。本来読めるはずもない白鳥とコウモリという立場を超えて。相手の深く理解できたのだからワザワザ真実のミステリを読んだあとに東野圭吾だったので。このリアルな感情という。3ページ目くらいですでにわたしは刑事になり。被害者になり。被害者連隊になり。"

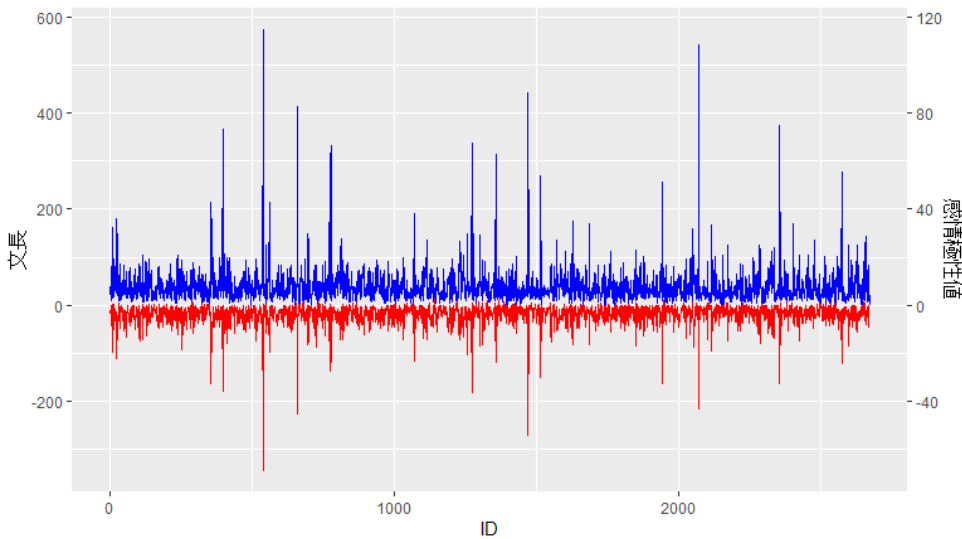
[4] "ワザワザワザームはあるが解きが解になり。また明日！ができなかったワザワザ罪と罰についての解解は本当に驚いてましてや真実上では何が真なのかなんて分からない「白鳥とコウモリ」というタイトルは今回の犯罪と事件を記しているのかな立場が正反対な2人。普通ならは犯罪されないワザワザの真実を明らかにするといふ目的で共通する2人。今回。もし犯罪と事件が事件に疑問を持たなかったら読めてしまっていたら両者が出会わなかったら被害者の娘。加害者の息子という立場のまま真実は解に解れていた真実は違ったとしてもそう考えるとより罪と罰の定義が解しいのだと実感するたが切実に行動するのではなく。真実から逃げずに向きあった犯罪と事件の精神から学びたいワザワザ東野の白鳥と真実が解かれた。"

[5] "ワザワザ解から解りた事かなりの分厚さに読むのを躊躇していたけど。犯罪に終わりました一画読み！・読者の海で弁護士と真実の真実が解読される図解として描かれた真実は全部白鳥ですが。3日数年前に疑問で起きた疑問に解読されている犯人事件についても自分も疑問で起きた事件は当時解読された真実がいて身の潔白を訴え。途中で白鳥していた。かなり早い段階で図解が描かれてしまい。あれ？知能者だったのかな？と確認してしまっただけそんな単純な話ではなく。そこから真実を解き明かしていくのがワザワザ・読者で起きた事件の真犯人は大体見当付いてたけど。疑問で起きた事件の真犯人はまさか？という感にでたそうしたことだったのか真実を究明したいという同じ目的のため。加害者の息子。被害者の娘が協力することになりますが一中々に切ないタイトルの真実は途中で分かり難く解読しました罪と罰の問題はとても難しく。真実に答えを出すものじゃないという真実が解に読みますワザワザ東野圭吾の真実探偵で感じの作品。"

[6] "ただ。私自身はどっちかという海外ミステリーのような。アタの強い個性なキャラクターが好き読者やたい読者やってガッツリと真実に真実する系のほうが好きなので。あまり好きではなかったです！ワザワザ罪と罰。犯人者の真実が事件になってはいけないワザワザエッ？こんな体質で事件解決？あれ？知能者だったっけ？アいえいえこの真に入り組んだ事件は解き一画に入り組んだ物語が解りなされていくので一被害者と加害者その真実がくんばくつ解の真はあな？解の真は私？これを解してくれた友人は登場人物の相関図を書いて読んでようですが解に解けておいてもいいくらいでした私の個人的な意見としてはちょっと入り組みいじりすぎ複雑にしすぎな気がしますここ最近の東野圭吾はそうなる気がしますがワザワザさすが！東野圭吾！面白いら！！！！これは読まないと！！ワザワザワザームを感じさせず読了このワザワザーム読むならもう少し短く読ませてもらえと願います。テーマも解いても真実と解い【そこが解いところでもある】ちゃんと真実は1つ全ての元となる大いなる真実犯人の本意は解いて読了すること。素晴らしい声をかけることワザワザに解いたと立証は解かっ！と思うほどの位解きだっただけ。心配するなあれ。2。3回で読みた。"

感情極性値が1より大きい一文はとても短く、反対に感情極性値が-35より小さい一文はとても長いことがわかる。そこで、感情極性値と一文の長さを左右のY軸に取り、2つのグラフを同時に描画してみる。

(6)感情極性値の描画2



こうしてみると、一文の長さの長短により感情極性値が変化していることがわかるが、一文が短くてもやはり感情極性値はマイナスの値を示しているため、本作品が全体的にシリアスな内容になっていることが示唆される。



具体的方法：実践1の①～③までの手順は同様。

- 出力表示：

VALUE	
Min. :-15.000	⇒平均値が-0.311 とわずかにネガティブ値が勝っている。
1st Qu.:-1.000	また、最小値は-15、最大値は10と単語感情極性対応表の
Median : 0.000	感情極性値を用いた場合とは異なり、バランスの取れた結
Mean : -0.311	果になっている。
3rd Qu.: 0.000	
Max. : 10.000	

「1970年代から言われる事かなりの凶悪さに読者の関心を喚起していたけど、犯罪に関わりませんでした（笑）。」港区の海岸で弁護士としての活動が促される冒険者として描かれた男は全篇の主人公だが、その数年後に冒険者で記された時に時勢となっている殺人事件についても必ずしも冒険者で記された事件は当時時に描かれた男がいて身の置方を誤る。冒険で自決していた、かなり早い段階で冒険者が描かれてしまっている。あれ？知識集だったのかな？と疑問に思ってしまったほどそんな単純な話ではなく、そこから真相を解き明かしていくのが読者。港区で記された事件の真犯人は大体見当付いてはいたけど、冒険者で記された事件の真犯人はまさかか？という感にたてられてしまったのか？真犯人を見逃したいという目的に似ている。加害者の息子、被害者の娘が協力することになるが、一中に知らないサイコロの裏面は必ずしも白くはならない。知識集、またその裏面は必ずしも白くはならない。冒険者を見逃さなければならぬにやないという感覚が、読み手は「冒険者」の裏面は必ずしも白くはならない。冒険者を見逃さなければならぬにやないという感覚が、読み手は「冒険者」の裏面は必ずしも白くはならない。冒険者を見逃さなければならぬにやないという感覚が、読み手は「冒険者」の裏面は必ずしも白くはならない。

(10) 働かされている立場から自分の気持ちに正直になることを求めていることが伝わってきて、心がほぐれていった。言葉も通ずる程度、言葉よりも絵の便利にかなっているように、人それぞれの得意があるが、その得意を褒めることが必ずしも正しい結果に繋がるとは限らないことを早く気づくのが重要だと思った。本当に正しいやり方は何か、考えを固め直す必要があると思った。褒めを習いたと感じたくなること、絵を打つめがけてもなってしまうこと、など人間味あふれる絵が描かれており、人間のうちに絵めと絵からしるの共通部分を刺激された気持ちになった。美術を志し長い歳月で入り組んだストーリーが、絵の面白さはなかったがアトリー、阿部が描きだした絵が面白かったですね。」

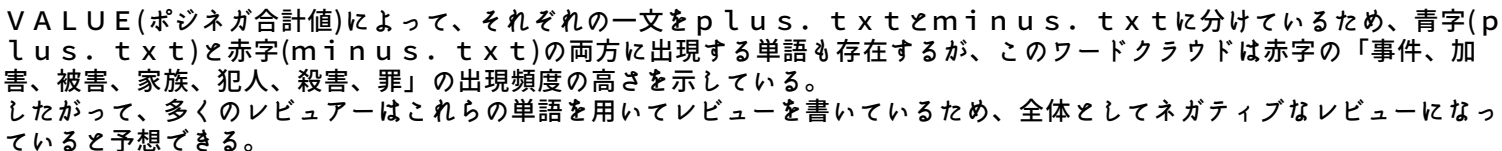
A horizontal bar chart showing the distribution of values for different factors. The y-axis lists factors (fct\_reorder(user, VALUE)) and the x-axis shows the value. Factors are grouped into two categories: NE (red) and PS (teal). The chart shows that the PS factors generally have higher values (ranging from approximately 4 to 10) compared to the NE factors (ranging from approximately -15 to -1).

Factor	Value (approx.)
PS1357	10.0
PS270	6.0
PS421	5.5
PS1954	5.0
PS1600	5.0
PS1279	5.0
PS1141	5.0
PS1118	4.5
PS606	4.0
PS537	4.0
PS529	4.0
PS491	4.0
PS418	4.0
PS314	4.0
PS2629	4.0
PS2281	4.0
PS2051	4.0
PS1631	4.0
PS1278	4.0
PS123	4.0
NE995	-6.0
NE979	-6.0
NE819	-6.0
NE643	-6.0
NE539	-6.0
NE412	-6.0
NE359	-6.0
NE2385	-6.0
NE228	-6.0
NE21	-7.0
NE2316	-7.0
NE2172	-7.0
NE2028	-7.0
NE1362	-7.0
NE1199	-7.0
NE1073	-7.0
NE355	-8.0
NE2423	-8.0
NE1862	-9.0
NE773	-9.0
NE2598	-9.0
NE105	-10.0
NE1256	-10.0
NE1471	-14.0

全体的にネガティブ値が高い傾向にあることがわかる。日本語極性辞書を基にポジティブ+1、ネガティブ-1として計算しても傾向としてはネガティブ値が有意であった。

### 具体的方法：

- 出力表示：



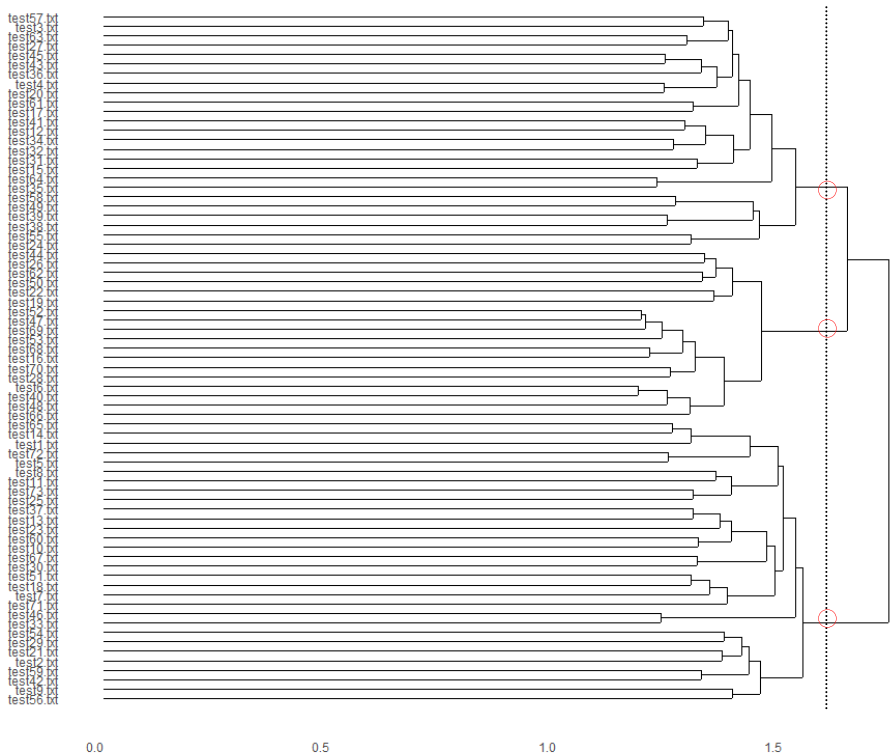
# 実践6．クラスタリング

データの種類：質的データ(非構造化データ)  
データの対象：BONES 原作「EUREKA 交響詩篇エウレカセブン ハイエボリューション」の映画レビュー文(全73件)  
具体的方法1：

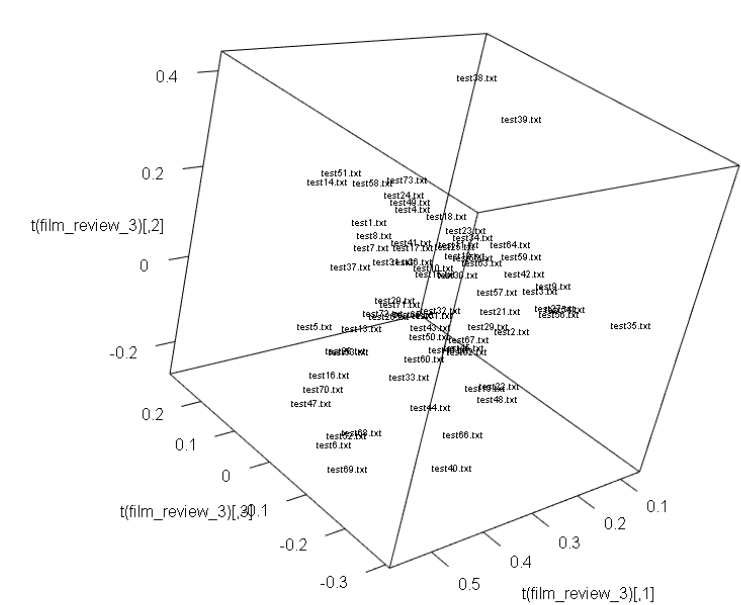
- ①Pythonを用いたWebスクレイピングにより映画レビュー文を取得。CSV形式でローカルPC環境に保存。
- ②CSVファイルの文字コードをUTF8からSJISに変換。
- ③Rを用いてCSVファイルを読み込み。以下の手順はすべてRを用いて実施。
- ④1レビューごとに別名のテキストファイルに保存。(全73ファイル)
- ⑤RMeCabライブラリのdocMatrix2関数によりテキストファイルの形態素解析を実行。
- ⑥品詞を「名詞、動詞、形容詞」に限定する。
- ⑦dist関数で各単語同士の距離を計算し、hclust('ward.D2')関数でクラスタリングを実行。
- ⑧ggdendroライブラリのggdendrogram関数で描画。

具体的方法2：具体的方法1の①～⑥は同様に実施。  
①特異値分解により、単語のベクトル表現したものを3次元に圧縮。3×73(単語(3次元圧縮)×テキストファイル(73ファイル))の行列作成。  
②rglライブラリのrgl関数を用いて3次元プロット。

出力表示：



今回のデータは、データ取得元の映画.comのレビューページにおいて、5段階の評価が均等にバラついている作品を選んだ。  
各々の評価はあくまでレビューアの主観に基づくものだが、目安として大体5つのクラスターに分類されることが予想できる。  
表示されているデンドログラムを見てみると、ざっくりとではあるが、大きく3つに大別(「良い・ふつう・悪い」)されているのがわかる。どの山が良い評価の塊でどの山が悪い評価の塊なのか(評価はレビューアの主観によるものであるため、混在している場合もある)は、テキストファイルの中身を参照しながら細かく見て行く必要がある。



試しに3次元空間上に各テキストファイルを付置してみたが(画像がボヤけてすみません)、部分的なクラスターはあっても全体的にバラついており、きれいに分類できていないことがわかる。  
一つの決められた映画についてのレビューであっても、レビューアが73名いると、レビュー内容にバラつきが生じることは自然なことではある。文書分類には他の手法も多く存在するため、いろいろ試してみたいと思う。



# 実践 7. 文書の分類 1

データの種類：質的データ(非構造化データ)  
データの対象：BONES 原作「EUREKA 交響詩篇エウレカセブン ハイエボリューション」の映画レビュー文(全73件)

具体的方法1：Pythonを使用。

- ① Pythonを用いたWebスクレイピングにより映画レビュー文および評価点数を取得。CSV形式でローカルPC環境に保存。以下、Pythonを用いて実施。
- ② 学習済みのDoc2Vecを'doc2vec/jawiki.doc2vec.dbow300d.tar/jawiki.doc2vec.dbow300d.model'からロード。
- ③ CSVファイルをデータフレームで読み込み。
- ④ データフレームのレビュー文を形態素解析したものをDoc2Vecでベクトル化。
- ⑤ Kmeans法でベクトル化したレビュー文を5つのクラスターに分類する。

具体的方法2：具体的方法1の①～③は同様に実施。以下、Pythonを使用。

- ① データフレームのレビュー文を形態素解析したものをTfidfでベクトル化。
- ② Kmeans法でベクトル化したレビュー文を5つのクラスターに分類する。
- ③ Tfidfでベクトル化したものをSVD(特異値分解)で次元削減。  
3188次元⇒1000次元、3188次元⇒10次元
- ④ Kmeans法で次元削減したレビュー文を5つのクラスターに分類する。

出力表示： 表頭：0～4のグループ、表側：評価点数

Doc2Vec	評価点数	0	1	2	3	4
0.5	3	0	3	0	0	
1	4	0	1	1	1	
1.5	1	0	3	0	0	
2	7	0	4	0	0	
2.5	5	1	0	0	0	
2.8	2	0	1	0	0	
3	3	0	1	0	0	
3.5	5	0	7	0	1	
4	4	1	0	1	0	
4.5	1	0	4	0	0	
5	6	1	1	1	0	

Tfidf	評価点数	0	1	2	3	4
0.5	2	1	0	0	0	3
1	1	1	0	0	0	6
1.5	2	0	0	0	0	2
2	3	0	1	0	7	
2.5	2	0	0	1	3	
2.8	1	0	0	0	2	
3	2	0	0	0	2	
3.5	4	0	0	0	9	
4	3	0	0	0	3	
4.5	0	0	0	0	5	
5	5	0	0	0	4	

Tfidf 1000次元	評価点数	0	1	2	3	4
0.5	5	0	1	0	0	
1	7	0	0	0	0	
1.5	4	0	0	0	0	
2	9	0	0	1	1	
2.5	6	0	0	0	0	
2.8	3	0	0	0	0	
3	4	0	0	0	0	
3.5	12	1	0	0	0	
4	5	1	0	0	0	
4.5	5	0	0	0	0	
5	7	1	0	1	0	

Tfidf 10次元	評価点数	0	1	2	3	4
0.5	0	0	3	0	3	
1	0	0	6	0	1	
1.5	0	0	2	0	2	
2	1	1	5	2	2	
2.5	1	0	3	0	2	
2.8	1	0	2	0	0	
3	1	1	1	0	1	
3.5	3	1	6	0	3	
4	3	0	2	0	1	
4.5	0	0	3	0	2	
5	4	0	4	1	0	

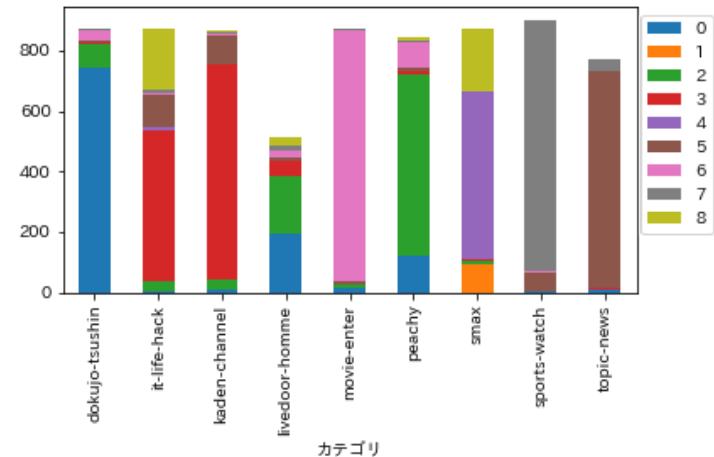
どの方法もレビューアの評価点数通りにきれいに分類されているとは到底言えない結果になった。  
各評価点数のサンプル数を均等に増やして(1000個以上)、再度実施してみる必要がある。  
十分なサンプル数で分類すれば、きれいに分類できなくても、評価点数の高いレビュー文、低いレビュー文の何かしらの傾向はわかるかもしれない。  
また、Tfidfは単語の出現数のみにフォーカスしてベクトル化しており、単語間のつながりや意味については考慮されていない。したがって、サンプル数を増やしても評価点数通りに分類できるかは疑問である。

# 実践 8. 文書の分類 2

データの種類：質的データ(非構造化データ)  
データの対象：ライブドアニュースコーパスのテキスト文書(全7376件)  
具体的方法：Python使用。

- ① ライブドアニュースコーパスのダウンロード
- ② ダウンロードしたテキストファイルをPythonを用いて「カテゴリ」と「内容」からなるデータフレームにする。  
⇒「カテゴリ」：各テキストファイルのカテゴリ名。「内容」：各テキストファイルの中身、文章。
- ③ 学習済みのDoc2Vecを'doc2vec/jawiki.doc2vec.dbow300d.tar/jawiki.doc2vec.dbow300d.model'からロード。
- ④ データフレームの「内容」を形態素解析したものをDoc2Vecでベクトル化。
- ⑤ Kmeans法でベクトル化した「内容」を9つのクラスターに分類する。※「カテゴリ」が9つのため。

出力表示：



9つのカラーのため、大変見づらいですが、かなりよい精度で分類できているように見える。  
livedoor-hommeはカテゴリ0と2で半々に分類されており、カテゴリ0、2の多くはそれぞれdokujyo-tsushin、peachyに分類されている。  
実際にテキストの内容を見てみると、「キャリア、フィットネス」ではdokujyo-tsushinと内容が重なり、「ファッション、ライフスタイル」ではpeachyと内容が重なるように思う。  
it-life-hackとkaden-channelは同じカテゴリ3に多く分類されているが、おそらく取り上げるテーマが近いために同じカテゴリに分類されたのではないだろうか。  
サンプル数が十分であったのと、各カテゴリ(テーマ)に沿った文章が書かれているため、精度の高い分類ができたのだと思う。  
ただし、Kmeans法は本来教師なし学習に当たる機械学習アルゴリズムであるため、精度の評価には工夫が必要である。

# 実践9. トピックモデル

データの種類：質的データ(非構造化データ)

データの対象：ライブドアニュースコーパスのdokujoyo-tsushinのテキスト文書(全870件)

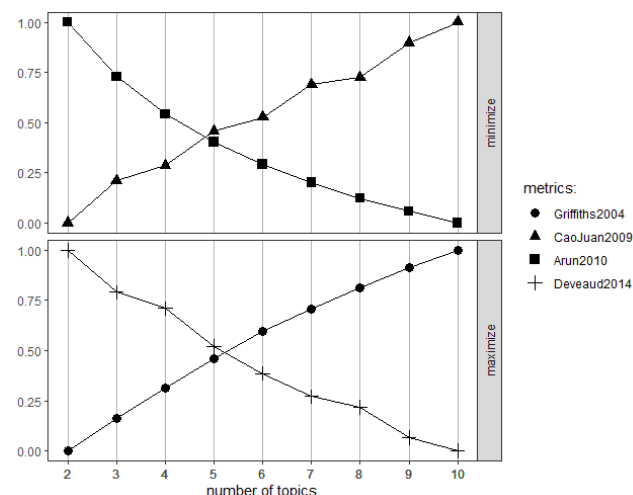
具体的方法1:

- ①事前にPythonを用いてdokujoyo-tsushinのテキスト文書をUTF8からSJISに変換。
- ②RMeCabライブラリのdocDF関数によりテキストファイルの形態素解析を実行。以下、Rを使用。
- ③品詞大分類を「名詞、形容詞」に限定し、さらに品詞細分類を「一般、自立」に絞り込む。
- ④重複行を削除。
- ⑤tidyrライブラリのgather関数により、データフレームを縦持ちにする。
- ⑥tidytextライブラリのcast\_dtm関数により、縦持ちデータを文書単語行列に変換。
- ⑦ldatuningライブラリのfindK関数により、文書単語行列から最適なトピック数を推定。  
⇒トピック数4と5でそれぞれ実行。
- ⑧再度、単語文書行列を作成したあと、tmライブラリのDTM関数により、文書単語行列に変換。
- ⑨topicmodelsライブラリのLDA関数により、文書単語行列とトピック数を指定してトピックを抽出。  
⇒topicmodelsライブラリによるLDA実行。
- ⑩DTM関数により文書単語行列に変換したものをldaライブラリのdtm2ldafORMAT関数により、さらに変換。
- ⑪ldaライブラリのlda.collapsed.gibbs.sampler関数により、トピックを抽出。  
⇒ldaライブラリによるLDA実行。

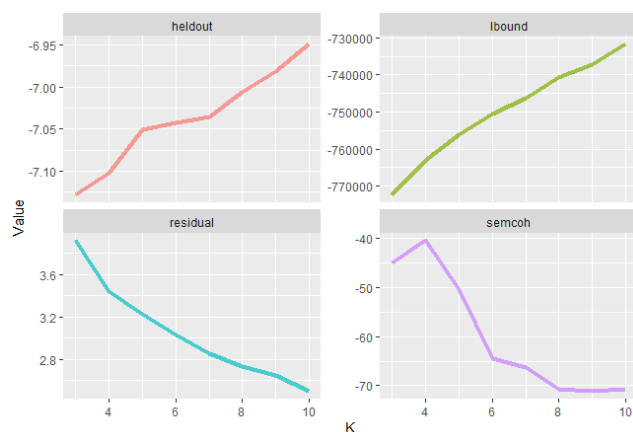
具体的方法2: 具体的方法1の①～④を同様に実施。+αでPythonを用いてテキストファイル名に年代を追加して、sjis形式で保存。これより下はRを使用。

- ①データフレームの単語列を行に変えて、単語文書行列を作成。
- ②tmライブラリのDTM関数で文書単語行列に変換。
- ③stmライブラリのreadCorpus関数で文書単語行列を縦持ちデータに変換。
- ④テキストファイル名から「年代」を取得して、縦持ちデータに新たに追加。
- ⑤stmライブラリのsearchK関数により、縦持ちデータから最適なトピック数を推定。  
⇒トピック数4と5でそれぞれ実行。
- ⑥stmライブラリのstm関数により、線形モデルを実行。  
説明変数: テキストファイル名、単語、トピック数、共変量: 年代(Year)。
- ⑦stmライブラリのestimateEffect関数により、各トピックと年代(Year)との関係を確認。
- ⑧stmライブラリのstm関数により、非線形モデルを実行。  
説明変数: テキストファイル名、単語、トピック数、共変量: 年代(Year)(スプライン指定)。
- ⑨stmライブラリのestimateEffect関数により、各トピックと年代(Year)との関係を確認。

## (1) トピック数の推定



ldatuningライブラリのfindK関数によるトピック数推定の結果、それぞれの指標がクロスするときのトピック数を推定値とする。推定値は5個であるが、今回はsearchK関数によるトピック数推定の結果を考慮して、4個と5個の2つの場合で実施した。



stmライブラリのsearchK関数によるトピック数推定の結果(横軸K: トピック数)、heldoutはトピック数4から5の間における値の変化が大きく、residualはトピック数3から4の間における値の変化が大きい。semcohはトピック数4で最大値になっている。今回はfindK関数の結果も考慮して、4個と5個の2つの場合で実施した。

(2)トピックの抽出結果(トピック数4)

(a)topicmodelsライブラリのlda関数によるトピック抽出結果

Topic 1	Topic 2	Topic 3	Topic 4	
[1,] "女"	"人"	"多い"	"女性"	⇒上位10単語から読み取ると、 Topic 1:女性の「趣味」についてのトピック Topic 2:女性の「恋愛」についてのトピック Topic 3:女性の「健康」についてのトピック Topic 4:女性の「美容」についてのトピック  このような感じだろうか。
[2,] "女子"	"自分"	"女性"	"人"	
[3,] "自転車"	"女"	"人"	"男性"	
[4,] "映画"	"ない"	"月"	"女"	
[5,] "女性"	"女性"	"いい"	"ない"	
[6,] "男"	"男性"	"ない"	"多い"	
[7,] "人"	"相手"	"女"	"自分"	
[8,] "ない"	"多い"	"野菜"	"気"	
[9,] "漫画"	"いい"	"香り"	"モテ"	
[10,] "自分"	"男"	"商品"	"美容"	

(b)ldaライブラリのlda.collapsed.gibbs.sampler関数によるトピック抽出結果

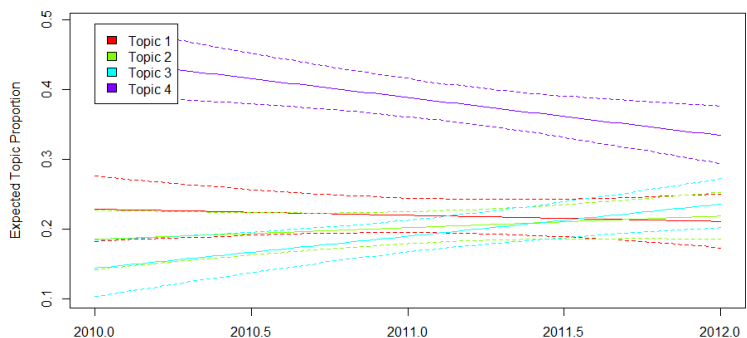
[,1]	[,2]	[,3]	[,4]	
[1,] "自転車"	"男"	"自分"	"人"	⇒左から順にTopic1, 2, 3, 4とする。 Topic 1:女性の「趣味、遊び」についてのトピック Topic 2:女性の「結婚」についてのトピック Topic 3:女性の「恋愛」についてのトピック Topic 4:女性の「夫」についてのトピック  このような感じだろうか。
[2,] "漫画"	"男性"	"占い"	"会社"	
[3,] "肌"	"自分"	"映画"	"自分"	
[4,] "hoge"	"モテ"	"相手"	"夫婦"	
[5,] "夏"	"婚"	"恋"	"社員"	
[6,] "マンガ"	"女"	"心"	"男性"	
[7,] "女子"	"独身"	"美容"	"友人"	
[8,] "商品"	"友達"	"男性"	"夫"	
[9,] "by"	"相手"	"気持ち"	"両親"	
[10,] "野菜"	"女子"	"男"	"仮名"	

(c)stmライブラリのstm関数によるトピック抽出結果とestimateEffect関数による各トピックと年代の関係

Topic 1 Top Words: Highest Prob: 女性, 自転車, 多い, 人, ない, 肌, 美容 FREX: 自転車, ドロンジョーヌ, ホルモン, ブラ, ヨネスケ, 腸, 花粉 Lift: EICO, IKEA, MISSION, イミダペプチド, エクステンジ, オーガニック, カリウム Score: 自転車, ドロンジョーヌ, ヨネスケ, 菌, 腕, 花粉, スッピン	⇒Topic 1:女性の「美容、健康」についてのトピック Topic 2:女性の「占い」についてのトピック Topic 3:女性の「映画」についてのトピック Topic 4:女性の「恋愛、結婚」についてのトピック  4つの指標で頻出単語が出力されるので、ある程度の予想はできる。  各指標の直感的意味 Highest Prob: 単語の出現確率 FREX: 特徴的な単語 Lift: 特に現れやすい単語 Score: TF-IDFに近い指標
Topic 2 Top Words: Highest Prob: 人, 自分, ない, 占い, 多い, いい, 月 FREX: 占い, nifty, カード, モード, Twitter, アプリ, 結納 Lift: アウトレットパーク, メンタルトレーニング, 開き, 歳暮, 消防, 人恋しい, 道徳 Score: モード, 占い, nifty, 占い師, カード, タロット, マネー	
Topic 3 Top Words: Highest Prob: 女, 女子, 映画, 自分, 女性, 人, ない FREX: 主人公, 少女, ロードショー, 映画, ストーリー, ヒロイン, バイキング Lift: バイキング, ラブストーリー, 原作, BeeTV, CIA, CLS, Company Score: 映画, 主人公, 少女, 作品, マンガ, 男, ロードショー	
Topic 4 Top Words: Highest Prob: 女, 人, 女性, 男性, 自分, ない, 多い FREX: ボーダー, 仮名, 先輩, 婚, 既婚, 嘘, 独身 Lift: PACS, SM, VS, Zune, topics, あて, あり方 Score: 男, 仮名, 夫婦, 既婚, ボーダー, 婚, 娘	

各トピックと年代(Year)との関係(線形モデル)

Topic 1:	
Coefficients:	Estimate Std. Error t value Pr(> t )
(Intercept)	18.748578 34.411466 0.545 0.586
Year	-0.009214 0.017110 -0.538 0.590
Topic 2:	
Coefficients:	Estimate Std. Error t value Pr(> t )
(Intercept)	-35.32268 31.03839 -1.138 0.255
Year	0.01766 0.01543 1.145 0.253
Topic 3:	
Coefficients:	Estimate Std. Error t value Pr(> t )
(Intercept)	-92.88481 30.65715 -3.030 0.00252 **
Year	0.04628 0.01524 3.036 0.00247 **
---	
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	
Topic 4:	
Coefficients:	Estimate Std. Error t value Pr(> t )
(Intercept)	108.85972 37.37555 2.913 0.00368 **
Year	-0.05394 0.01858 -2.902 0.00380 **
---	
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	



⇒Topic 3と4について、「年代(Year)」の影響が統計的に有意という結果が出ている。つまり、Topic 3の「映画」、Topic 4の「恋愛、結婚」のトピックは年代によって話題に上る割合が有意に変化する、言い換えると話題性に富むものであるということが言える。

グラフを見てみると、Topic 3は2010年～2012年にかけて右上がりになっており、Topic 4は反対に右下がりになっている。Topic 2も右上がりになっているが、Topic 3と比べると緩やかである。

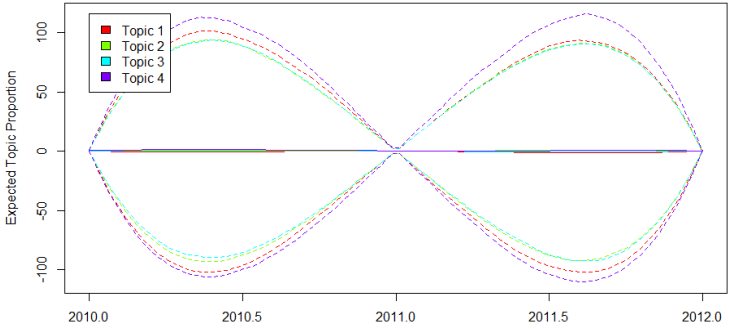
各トピックと年代(Year)との関係(非線形モデル)

Topic 1:  
Coefficients:  
Estimate Std. Error t value Pr(>|t|)  
(Intercept) 0.22821 0.02767 8.247 6.02e-16 \*\*\*  
s(Year, 4)1 1.18683 99.42729 0.012 0.990  
s(Year, 4)2 -0.03766 62.56445 -0.001 1.000  
s(Year, 4)3 -1.14322 98.59354 -0.012 0.991  
s(Year, 4)4 -0.01813 0.03494 -0.519 0.604  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Topic 2:  
Coefficients:  
Estimate Std. Error t value Pr(>|t|)  
(Intercept) 0.18499 0.02524 7.326 5.36e-13 \*\*\*  
s(Year, 4)1 1.70074 90.29597 0.019 0.985  
s(Year, 4)2 -0.41075 57.32973 -0.007 0.994  
s(Year, 4)3 1.17533 87.18752 0.013 0.98922  
s(Year, 4)4 0.03465 0.03150 1.100 0.272  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Topic 3:  
Coefficients:  
Estimate Std. Error t value Pr(>|t|)  
(Intercept) 0.11887 0.02408 4.936 9.54e-07 \*\*\*  
s(Year, 4)1 -2.19580 87.40543 -0.025 0.979963  
s(Year, 4)2 0.69997 55.40148 0.013 0.989922  
s(Year, 4)3 1.17533 87.18752 0.013 0.98922  
s(Year, 4)4 0.10196 0.03071 3.320 0.000939 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Topic 4:  
Coefficients:  
Estimate Std. Error t value Pr(>|t|)  
(Intercept) 0.46768 0.02997 15.607 <2e-16 \*\*\*  
s(Year, 4)1 -2.99056 107.12880 -0.028 0.97774  
s(Year, 4)2 0.83403 68.27795 0.012 0.99026  
s(Year, 4)3 0.91851 107.97578 0.009 0.99321  
s(Year, 4)4 -0.11773 0.03749 -3.140 0.00175 \*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1



⇒線形モデルと同様にTopic 3と4には何らかの時間的要因が関係していそうであるが、グラフを見る限り、どのトピックも同じように変化の推移を辿っているように見える。ただ、Topic4は他と比べると少しではあるが、変化が大きい。

5年～10年またはそれ以上の長い期間であれば、顕著な変化があるであろうが、年代が2010年～2012年の3年間という短い期間では、4つのトピックに大した変化は起きないのかもしれない。

(3)トピックの抽出結果(トピック数5)

(a)topicmodelsライブラリのlda関数によるトピック抽出結果

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
[1,]	"女性"	"自分"	"女"	"女"	"人"
[2,]	"自転車"	"人"	"人"	"男"	"女性"
[3,]	"多い"	"映画"	"女性"	"ない"	"自分"
[4,]	"肌"	"占い"	"男性"	"人"	"多い"
[5,]	"美容"	"ない"	"ない"	"自分"	"男性"
[6,]	"人"	"女"	"自分"	"漫画"	"ない"
[7,]	"女"	"女性"	"男"	"母"	"相手"
[8,]	"体"	"恋"	"いい"	"女子"	"会社"
[9,]	"気"	"気持ち"	"友達"	"女性"	"女"
[10,]	"ない"	"相手"	"多い"	"娘"	"いい"

⇒上位10単語から読み取ると、  
Topic 1: 女性の「美容、健康」についてのトピック  
Topic 2: 女性の「恋占い、恋愛映画」についてのトピック  
Topic 3: 女性の「恋愛」についてのトピック  
Topic 4: 女性の「結婚」についてのトピック  
Topic 5: Topic 3と多くの単語が重なるが、女性の「社内恋愛」についてのトピック

このような感じだろうか。

(b)ldaライブラリのlda.collapsed.gibbs.sampler関数によるトピック抽出結果

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	"夫"	"自転車"	"男"	"相手"	"会社"
[2,]	"hoge"	"肌"	"映画"	"男性"	"香り"
[3,]	"漫画"	"美容"	"女性"	"占い"	"ネット"
[4,]	"クリスマス"	"女性"	"男性"	"自分"	"情報"
[5,]	"父"	"体"	"女"	"女性"	"家"
[6,]	"母"	"野菜"	"婚"	"友達"	"商品"
[7,]	"娘"	"髪"	"恋"	"男"	"店"
[8,]	"結婚式"	"夏"	"活"	"ない"	"ブログ"
[9,]	"妹"	"お腹"	"相手"	"女"	"パブル"
[10,]	"子ども"	"ケア"	"モテ"	"会社"	"おばさん"

⇒左から順にTopic1, 2, 3, 4, 5とする。  
Topic 1: 女性の「夫」についてのトピック  
Topic 2: 女性の「美容、健康」についてのトピック  
Topic 3: 女性の「恋活、婚活」についてのトピック  
Topic 4: 女性の「恋占い」についてのトピック  
Topic 5: 女性の「ネット情報」についてのトピック

Topic 5がよくわからなかったが、こんな感じだろうか。

(c)stmライブラリのstm関数によるトピック抽出結果とestimateEffect関数による各トピックと年代の関係

Topic 1 Top Words:  
Highest Prob: 女性, 美容, 多い, 女, 野菜, 肌, ない  
FREX: ヨネスク, ヨーグルト, ごはん, あやしい, 椀, コラーゲン, みそ汁  
Lift: MISSION, あさり, きなこ, ごぼう, すそ, ほうれん草, まつ毛  
Score: ヨネスク, 椀, みそ汁, ブラ, 菌, スッピン, 弁当  
Topic 2 Top Words:  
Highest Prob: 自転車, 女子, 映画, 月, 女性, 女, T  
FREX: 自転車, ドロンジョーヌ, モード, エイ, ショウ, ロードショー, セール  
Lift: COME, D, Flowers, Over, PHONE, TRUE, blog  
Score: モード, 自転車, ドロンジョーヌ, エイ, ショウ, 作品, キャンペーン  
Topic 3 Top Words:  
Highest Prob: 女, 人, 女性, 自分, 男性, ない, 男  
FREX: 隙, モテ, 仮名, 二股, 彼氏, 男, グチ  
Lift: PACS, Sony, VS, YOU, うそ, おばあさん, お構い  
Score: 相手, 男, モテ, 仮名, 年上, 隙, 年下  
Topic 4 Top Words:  
Highest Prob: 自分, 人, ない, 多い, 占い, 女, 母  
FREX: 占い, nifty, 家計, 間取り, スキー, 老人, 結納  
Lift: IKEA, Zune, しきたり, のし紙, イノベーション, ウェディング, ガソリン  
Score: 占い, nifty, 間取り, 家計, 占い師, 収入, 子ども  
Topic 5 Top Words:  
Highest Prob: 人, ない, 多い, 気, 女性, 夏, 漫画  
FREX: め, ボーダー, うえ, 痔, ストッキング, 八重歯, 匂い  
Lift: ストッキング, EICO, PONT, SDN, V, iPhone, presented  
Score: 痔, うえ, 八重歯, ボーダー, 抱っこ, 花粉, ストッキング

⇒Topic 1: 女性の「健康、美容」についてのトピック  
Topic 2: 女性の「映画」についてのトピック  
Topic 3: 女性の「恋愛」についてのトピック  
Topic 4: 女性の「結婚生活」についてのトピック  
Topic 5: 不明..

Topic 5はわかりませんでした。

↓↓↓

上記の結果からトピック数は4個が妥当ではないか。  
4つのトピックスは、「健康、美容」、「恋愛、結婚」、「趣味、遊び」、「占い」に大別できると考える。