PWS Cup 2022

COVID-19 重篤化患者のプライバシーを守り切れるか？

ルール説明

PWS Cup ワーキンググループ　グループ長 野島 良

version 1.1

- 2022/09/22 更新情報
- ルール説明1.0→ルール説明1.1
- 評価プログラムの公開および評価指標の記載を追記（Ｐ33～39）

- １．匿名化評価指標は、有用性評価２と、オッズ評価２（重篤化リスク評価）の４つになります。スコアはその平均値を表示しています。
- There are four anonymization evaluation indicators: usefulness rating 2 and odds rating. Scores are their averages.
- ２．評価の年齢は年代でまとめてあります。
- The age of the usefulness evaluation is summarized by age.
- ３．有用性評価相関行列については、本戦ではリターン値を変更します。
- For the usefulness evaluation correlation matrix, the return value will be changed in the Final Round.
- return 1 - (dfOD.corr()-dfAD.corr()).abs().sum().sum()/(dfOD.shape[1]*dfOD.shape[1]*2)

- ※評価の年齢は有用性評価のみ年代でまとめてありますと報告しましたが、すべてに共通にしてあります。

PWS Cup 2022
匿名ヘルスケアデータコンテスト

- COVID-19 重篤化患者のプライバシーを守り切れるか？
  - コロナ重篤化患者の、性別、年齢、人種、学歴、病歴などからなるデータ
    - [*]において、NHANESをベースとした合成データが作成
    - 本合成データを参考
  - 誰のレコードかわからない様に匿名化しても、コロナで重篤化するリスクを算出できることを目指す

[*]B. Seligman, M. Ferranna, D.E. Bloom, Social determinants of mortality from COVID-19: A simulation study using NHANES, PLOS Medicine 18(12): e1003888

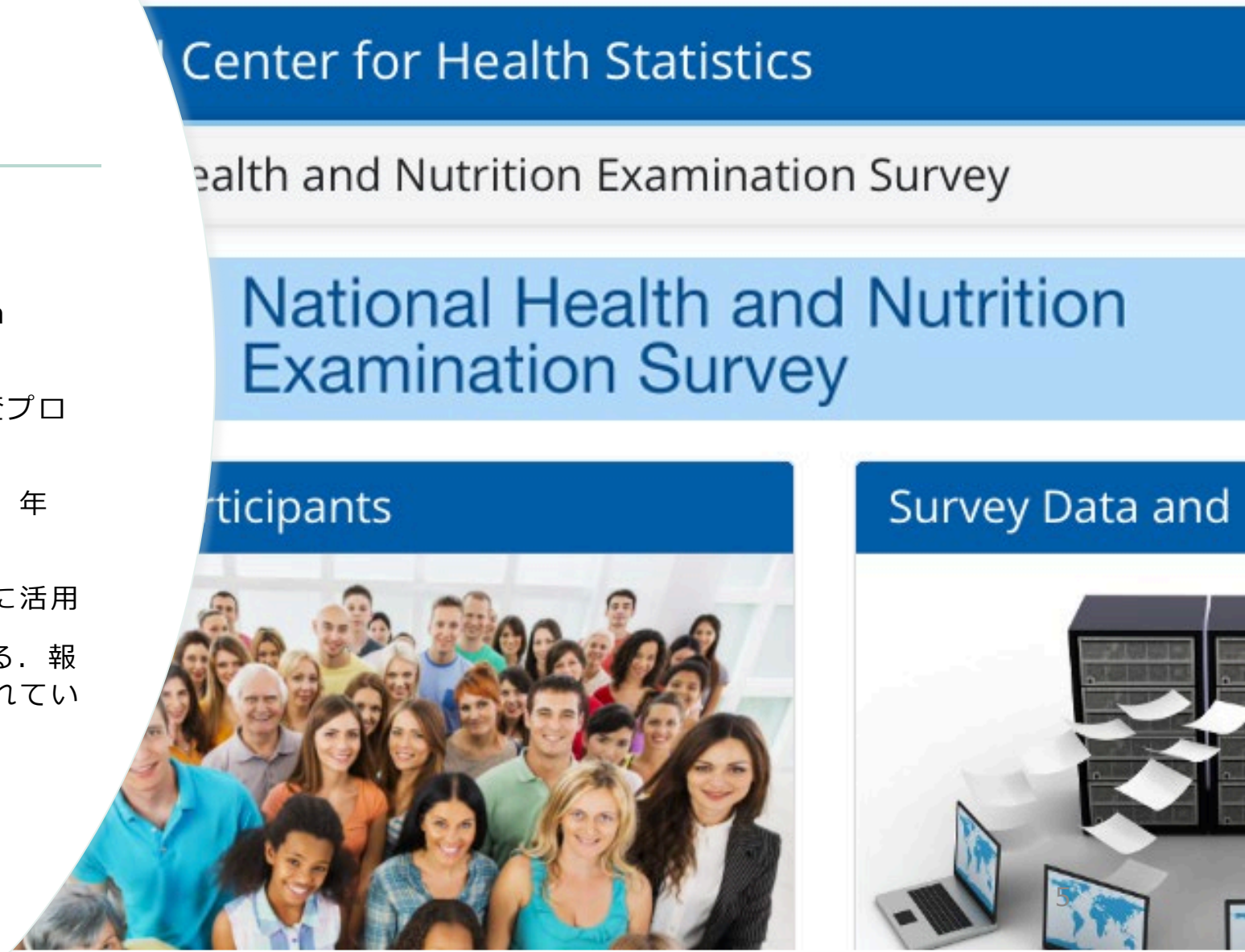
ストーリー

- 登場人物
  - 加工者：コロナ重篤化患者からなるデータを匿名化する
  - 攻撃者：匿名化されたデータから、知人が重篤化したかどうかを特定する
  - 活用者：匿名化されたデータから、重篤化リスクを算出する
  - 審判（事務局）：どの加工者が正しく安全に加工しているか判定

データ

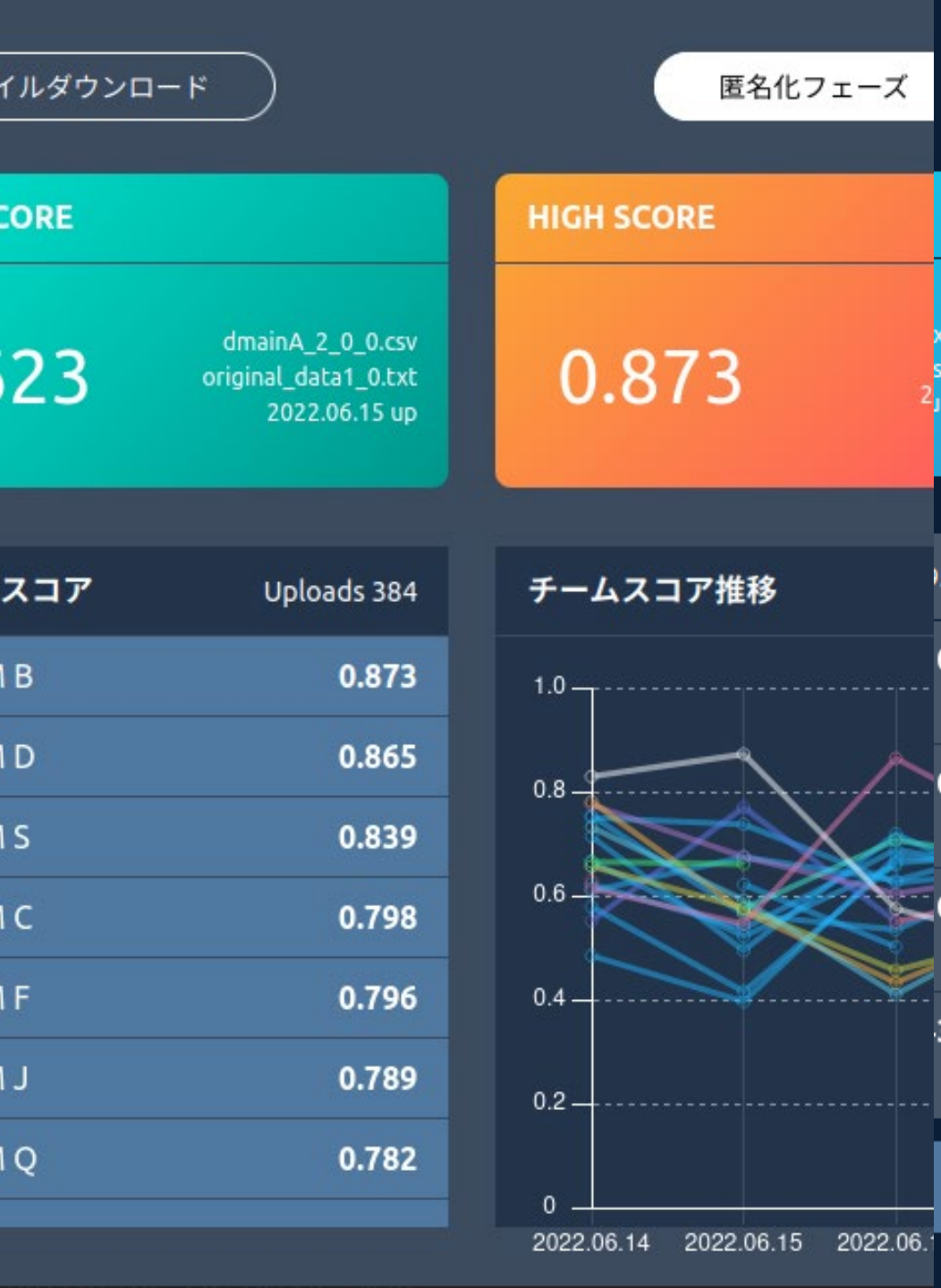
NHANES 概要

- National Health and Nutrition Examination Survey
- CDC (米国疾病対策センター)の国民健康栄養調査プログラム
- 1960年代から行われている調査．全米15箇所で，年5,000人を調査している．
- 疫学研究，健全な公共健康政策やサービスの施策に活用
- 被験者世帯は，NCHS所長からのレターを受け取る．報酬と診断結果を得る．プライバシーは法律で守られている(privacy is protected by public laws)



スケジュール

5月	6月	7月	8月	9月	10月
• データセット整備，ルール案 • ポスター，ウェブ	• 有用性評価， • 安全性評価	• トライアル • 参加者募集開始 2022/7/22-9/14	• 予備戦 匿名化 2022/8/18-8/30	• 予備戦　攻撃 2022/9/2-9/13 • 本戦　匿名化 2022/9/16-10/3	• 本戦　攻撃 2022/10/7-10/18 • リハーサル • 2022/10/26 CSSポスターセッション（終日）



コンテスト用システム！

- 今年度は、コンテスト用のシステムを準備
- みんなでオンラインで対戦しよう

参加方法とアクセス方法

- 参加方法（2022/7/22～9/14）

  PWSCUP2022ホームページ
  https://www.iwsec.org/pws/2022/cup22.html
  PWSCUP2022参加規程
  https://www.iwsec.org/pws/2022/entry.html
  をよくお読みになり、エントリーフォームから、お申し込みください。

- アクセス方法

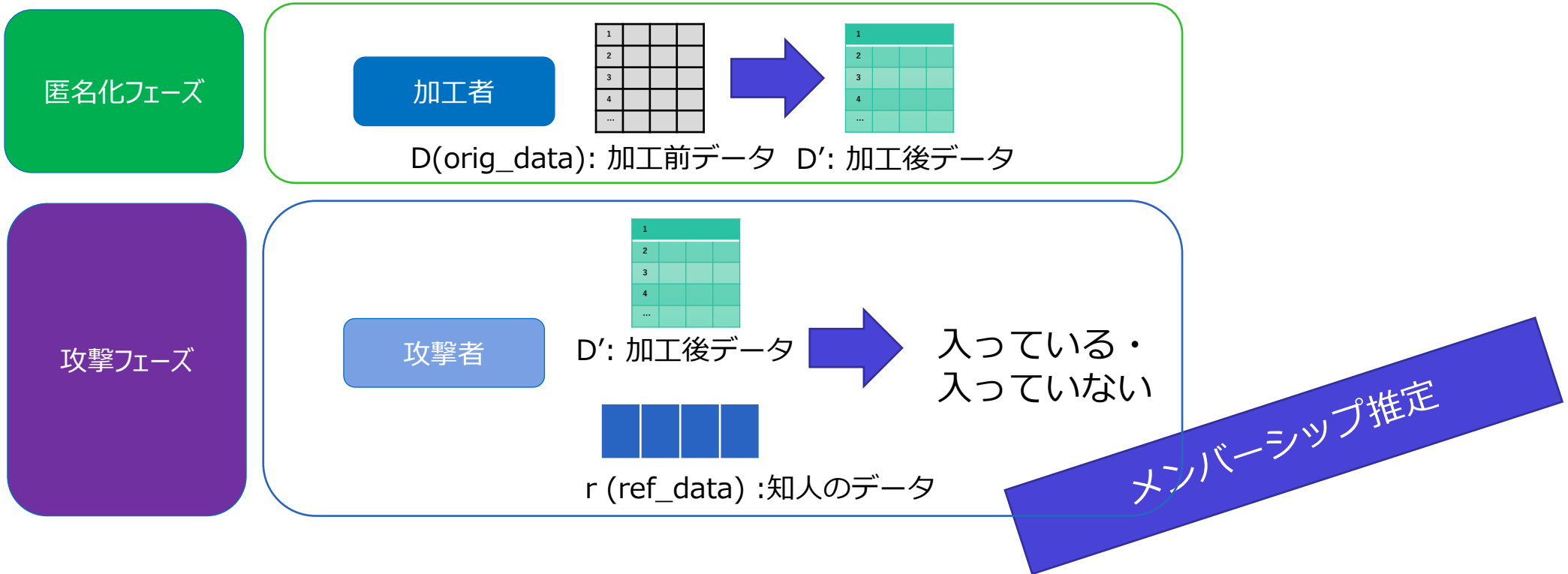
  PWSCUP2022事務局から参加登録完了メールが届きます。

  その後、大会システムのURLとIDとパスワード、利用規約をメールでうけとり、
  大会のシステムのマイページにアクセス可能となります。

本年度は締め切りました。

コンテスト概要

- 加工者：重篤化した患者のデータ（D）を匿名化（D ’）する
- 攻撃者：知人が重篤化したかどうかを知りたい




スミレちゃん
元ファイルダウンロード
匿名化フェーズ
攻撃フェーズ
PWSCUP
SCORE
623
dmainA_2_0_0.csv
original_data1_0.txt
2022.06.15 up
TEAM SCORE
0.623
dmainA_2_0_0.csv
original_data1_0.txt
2022.06.15 up
HIGH SCOR
0.87
47:
ードリスト
Uploads 4
_2_0_0.csv
0.623
l_data1_0.txt
6.15 up
_2_1_0.csv
0.546
l_data1_0.txt
6.18 up
_2_2_0.csv
0.503
l_data1_0.txt
ファイル管理
全チームスコア
Uploads 384
1 TEAM B 0.873
2 TEAM D 0.865
3 TEAM S 0.839
4 TEAM C 0.798
5 TEAM F 0.796
6 TEAM J 0.789
7 TEAM Q 0.782
チームスコア
スコアを
1.0
0.8
0.6
0.4
0.2
0
2022.06.14
2022.06.15
2022.06.16
2022.06.17
2022.06.18
2022.06.19
2022.06.20
匿名化フェーズ
1．元データをダウンロード
2．事務局が提供する加工手法を使い、
データを加工
3．結果をアップロード
4．システム上で自動採点
5．複数の結果の中から1つを選び提出
10

匿名化フェーズ詳細（加工方法）

- 課題となるデータは６種類用意されており、マイページからダウンロードできます。
- ダウンロードしたデータは、事務局が提供するアルゴリズム（Python3）プログラムを使って各自の環境で加工し、加工手法のログとともにマイページにアップロードし、数秒後、採点結果を確認できます。
- データ加工した結果は複数アップロード可能（条件付き）ですが、最終結果としてその中から1つ選んで提出してください。

  注）加工に利用するアルゴリズムはPWSCUPシステムのダウンロードサイトから入手が可能です

  　　提出した匿名化データは攻撃フェーズでほかのチームからの攻撃対象データとして使われます。

匿名化に使うデータ（加工前データ）

次のようなデータを匿名化してもらいます

コロナ重篤化した人のデータセット (300名)

病歴

AGE	GENDER	RACE	INCOME	EDUCATION	VETERAN	NOH	HTN	DM	IHD	CKD	COPD	CA
66	2	4	3	2	0	1	1	0	0	0	0	0
80	2	4	4	3	0	2	1	0	0	0	0	0
76	1	4	7	4	1	1	0	0	0	0	0	0
50	1	6	15	5	0	3	1	0	0	0	0	0
77	2	3	77	3	0	1	0	0	0	0	0	0

年齢(Race)　　：20 ～ 80
性別(Gender)　：1 男性、 2 女性
人種(Race)　：1 ヒスパニック系ではない白人
2 メキシコ系アメリカ人
3 その他のヒスパニック
4ヒスパニック系ではない黒人
6 ヒスパニック系ではないアジア人
7 その他
収入(Income): 1 ～15 大きいほど収入大
77、99不明

教育(Education): 1 ～ 5 大きいほど高学歴
5が大学卒業以上
7、 9 が不明
軍歴（Veteran): 経験者は1
経験者でなければ9
所帯数（NOH）：最大 7
各病歴(症状があった場合１ なければ０)
HTN (高血圧）、DM （ 糖尿病）、IHD (心血管疾患)、CKD(慢性腎臓病)、COPD(慢性閉塞性肺疾患)、CA （ガン）

付録：データセット生成の概要

- 本カップでは[*]を参考にデータを生成しています。[*] では:
  - まず、 NHANES2017-2018から各人（各レコード）が、年齢、人種、収入、教育レベル、軍歴、各種病歴からなるデータセットを抽出
  - 続けて、ガウスコピュラに基づきコロナ重篤化か否かを見積り、その擬似データを上記レコードに付与。つまり、各人（各レコード）が、年齢、人種、収入、教育レベル、軍歴、各種病歴、コロナ重篤化 となる一部合成データとなるデータセット D0 を生成
- 本カップで用いるデータセット
  - 加工前データ D：D0の内、コロナに重篤化した個人をランダムに300名選び加工前データ D とします。
  - 攻撃用データ R：D0のうち、コロナに重篤化しなかった個人をランダムに25名選び、さらにDから25名をランダムに選び合算したデータ R

本カップにおいては上記操作、つまりD0を加工前データ毎に作っています。また、コロナに重篤化する個人の選び方は、原論文よりも高い確率としています。

[*] B. Seligman, M. Ferranna, D.E. Bloom, Social determinants of mortality from COVID-19: A simulation study using NHANES, PLOS Medicine 18(12): e1003888

top2.py

列col でしきい値 theta より大きい行を出力する．列は 1_5 の様に複数与えても良い．

```
for i in range(len(cols)):
        ex &= (df.loc[:, int(cols[i])] < int(thetas[i]))
```

コマンド: python top2.py 加工前.csv 加工後.csv col theta

bottom2.py

列col でしきい値 theta より小さい行を出力する．列は 1_5 の様にベクトルで与えても良い．

```
for i in range(len(cols)):
        ex &= (df.loc[:, int(cols[i])] > int(thetas[i]))
```

コマンド: python bottom2.py 加工前.csv 加工後.csv col theta

kanony.py

列columnsを準識別子とみなしてk-匿名化する．削除する行を出力

```
def kanony(df, qi=[1, 2], k=1):
        return df.groupby(qi).filter(lambda x: x[0].count() >= k)
```

コマンド: python kanony.py 加工前.csv 加工後.csv k col

exclude.py

入力input.csvから排除行番号exclude を除いて出力する.

```
exclude_rows = list([int(i) for i in sys.argv[3].split("_")])
df = df.drop(index=df.index[exclude_rows])
```

コマンド

exclude.py 元ファイル名　4 結果ファイル名

shuffle.py

行と列をランダムサンプリングする

```
df2 =  df.sample(frac=1, random_state=int(sys.argv[3]))
df2.to_csv(sys.argv[2], header=False, index=False)
```

コマンド

suffle.py 元ファイル名　4 結果ファイル名

rr.py

列col において, 確率1 - prob で他の値と置き換える. ただし,

random は乱択アルゴリズムで利用する乱数とする.

```
def rr(x, q):
    uniq = x.value_counts().index.values
    y = [i if random.random() < q else random.choice(uniq) for i in x]
    return(y)
def rrdf(df, q, target):
    df2 = df.copy()
    for i in target:
        df2.iloc[:, i] = rr(df.iloc[:, i], q)
    return df2
```

コマンド: python rr:py 加工前:csv 加工後:csv prob col random

dp.py→lap.py

名前変更

差分プライバシーに基づいて行colsにεのラプラスノイズを付加する

```
def lap(x, eps):
  x = (x + np.random.default_rng().laplace(0, 1/eps,
x.shape[0]))
  return  ((x * 2 + 1) // 2).astype(int)
```

コマンド: python dp.py 加工前.csv 加工後.csv col epsilon

四捨五入

top2_round.py
new

```
for i in range(len(cols)):
    df.iloc[df.iloc[:,cols[i]] >= chop[i], cols[i]] = chop[i]
df.to_csv(out, index = None, header = None)
```

コマンド: python top2_round.py 加工前:csv 加工後:csv col
theta

31	23	41	80	61

61以上を削除するのではなく、60にする

31	23	41	60	60

bottom2_round.py
new

```
for i in range(len(cols)):
    df.iloc[df.iloc[:,cols[i]] <= chop[i], cols[i]] = chop[i]
df.to_csv(out, index = None, header = None)
```

python bottom2_round.py 加工前.csv 加工後.csv col theta

31	23	41	80	61

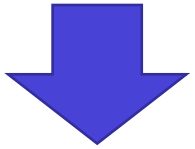
30未満を削除するのではなく、30にする

31	30	41	80	61

age_layer.py　new

年齢について…

31	23	41	80	67



30	23	40	80	60

python age_layer.py 加工前.csv 加工後.csv

nn.py　new

```
for i in range(len(cols)):
    df.iloc[rows[i],cols[i]] = 99
df.to_csv(out, index = None, header = None)
```

好きな場所を99（不明の意味）に置き換えることができる。

python nn.py 加工前.csv 加工後.csv 1_2 3_4 (1行3列目と2行4列目を99)

average.py new

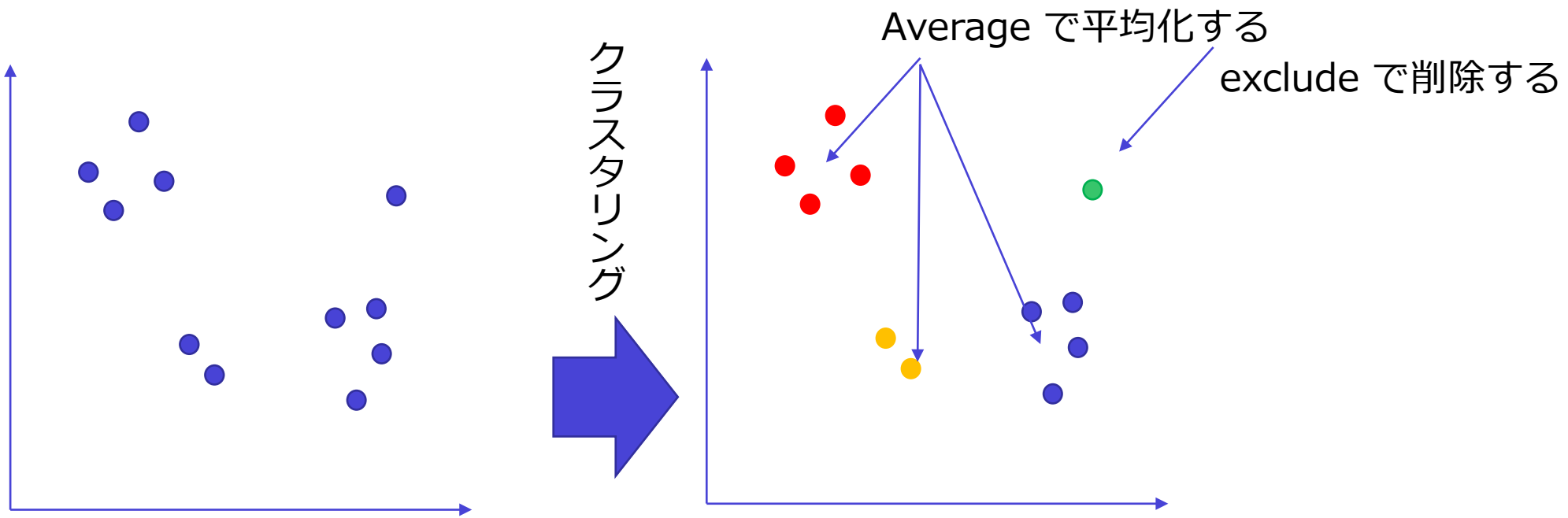
```
avg = df.iloc[rows,:].mean().astype(int)
for i in range(len(rows)):
    df.iloc[rows[i]] = avg
df.to_csv(out, index = None, header = None)
```

指定した行の平均を取り、それらの行にその値を割り当てる

python nn.py 加工前.csv 加工後.csv 1_2_3_4

1, 2, 3, 4行目の平均を取り、それらを平均と置き換える

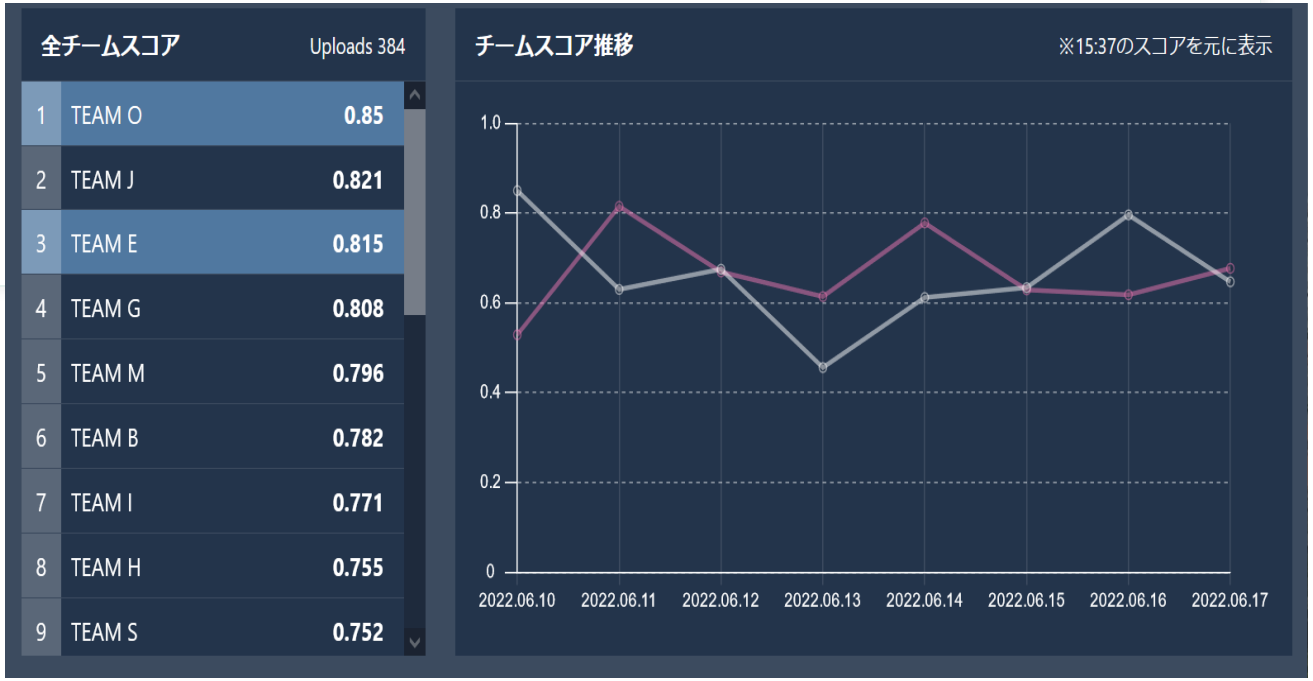
average で意図していること



クラスタリングして、平均に置き換える

匿名化フェーズ詳細
提出物

加工データアップロード時に
１．加工データ
２．加工に用いたアルゴリズムのログデータ
を提出します。
提出ファイルはアップロード時にフォーマットチェックが行われ、正しいデータがシステムに登録されます。



ログデータの例（正式なものは大会開始時に公開）：

```
python ../Anon/kanony2.py ../Data/orig_data1.csv anon_data1_k.csv 2 1_2
python ../Anon/rr.py anon_data1_k.csv anon_data2_rr.csv 0.2  1_2 31
python ../Anon/dp2.py anon_data2_rr.csv  anon_data2_rr.csv  0  0.1 31
python ../Anon/top2.py anon_data2_rr.csv  anon_data2_rr.csv  0 80
python ../Anon/bottom2.py anon_data2_rr.csv  anon_data2_rr.csv  4  1
python ../Anon/shuffle.py anon_data2_rr.csv  anon_data2_rr.csv  4
python ../Anon/exclude.py anon_data2_rr.csv anon_data2_rr.csv 4
```

例：これを提出する加工データとしてください。


スミレちゃん
元ファイルダウンロード
匿名化フェーズ
攻撃フェーズ
PWSCUP
UR SCORE
0.426
amainA_3_46.txt
K dmainK_2_2_2.csv
2022.06.16 up
TEAM FILE
dmainA_2_0_0.csv
0.268
[73]
dmainA_2_1_0.csv
0.278
[67]
dmainA_2_2_
0.31
[19]
ップロードリスト
Uploads 95
ainA_3_7.txt
0
dmainC_2_0_0.csv
2022.06.14 up
ainA_3_8.txt
0
dmainC_2_1_4.csv
2022.06.21 up
ainA_3_26.txt
0
dmainG_2_3_3.csv
2022.06.14 up
ainA_3_22.txt
0.243
dmainF_2_3_0.csv
ファイル管理
匿名化ファイルランク
Uploads 99
1 O dmainO_2_3_2.csv 0.378 [91]
2 G dmainG_2_2_0.csv 0.367 [91]
3 D dmainD_2_1_0.csv 0.363 [91]
4 B dmainB_2_3_0.csv 0.356 [91]
5 E dmainE_2_2_4.csv 0.356 [91]
6 L dmainL_2_1_2.csv 0.343 [91]
7 E dmainE_2_0_3.csv 0.336 [91]
8 O dmainO_2_2_2.csv 0.335 [91]
チームスコ
1.0
0.8
0.6
0.4
0.2
0
2022.06.14
攻撃フェーズ
①他のチームが作成した加工データ
②事務局が用意した知人のデータ
を使って、
知人が「加工後のデータ」に含まれているかどうか
を推定します。
• 知人のリストは2X名から構成されています。
• X名がデータに含まれており、X名が含まれていない構成となっています。
• 的中させた人数をZ としたときに，下記を得点とします
min ((2X − Z)/X, 1)
二倍に
• 攻撃回数は平等性を期すため制限があります。
1．攻撃対象データを選び、攻撃対象データと知人データを
ダウンロード
2．推定データを作成
3．結果をアップロード
4．システム上で自動採点

攻撃フェーズ詳細
提出物

- 攻撃者は、他の参加者が加工した加工後データD’毎 に下記のような知人リスト R を得ます。
- Rの半分はD’ の加工前データDに含まれており, 残りの半分はDに含まれていません。

AGE	GENDER	RACE	INCOME	EDUCATION	VETERAN	NOH	HTN	DM	IHD	CKD	COPD	CA
66	2	4	3	2	0	1	1	0	0	0	0	0
80	2	4	4	3	0	2	1	0	0	0	0	0
76	1	4	7	4	1	1	0	0	0	0	0	0
50	1	6	15	5	0	3	1	0	0	0	0	0
77	2	3	77	3	0	1	0	0	0	0	0	0

知人リストR

提出ファイル

1
0
1
1
1
1

参加者（攻撃者）は、Rの各行について、Dに含まれていると判断すれば１、含まれていないと判断すれば0となるファイルを提出してください

総合評価

- **予備戦・本戦**
  - 有用性評価: 3 つの評価手法の平均。それぞれ最低 0 、最大 1
  - 安全性評価：最大 1 、最低 0 （破られていないものの割合）
  - 評価：有用性評価と安全性評価の平均
- **総合評価**
  - 予備戦 1 対 本戦 9 の比で評価

攻撃部門

- **攻撃部門：**他チームの加工データに対して、どれだけメンバーシップ推定を成功させたか
  - 評価合算方法: 予備戦 1 対 本戦 9 の比で評価

Bチームの他チームに対する攻撃結果

A	B	C	D	E	F	G	H	I	J	平均
0.2		0.3	0.2	0.1	1	0.1	0.2	0.1	0.2	0.26

攻撃していないところ

評価点

二刀流部門

- **二刀流部門:** 総合部門と攻撃部門の結果の両方で一流だったチーム
  - 総合評価部門と攻撃部門の最終順位の平均で順位付け

有用性評価: 重篤化リスク評価

- ロジスティック回帰分析：
  - COVID ~ AGE + GENDER + RACE + INCOME + EDUCATION + VETERAN + NOH + HTN + DM + IHD + CKD + COPD + CA
- 加工データ、加工前データにおいてそれぞれロジスティック回帰分析を行う
- それぞれの偏回帰係数（の指数乗）の差分の合計(OR比)
- ただし、加工データには重篤化患者しか含まれないため、重篤化患者ではない個人のデータがプログラムに含まれている。

有用性評価: 重篤化リスク評価

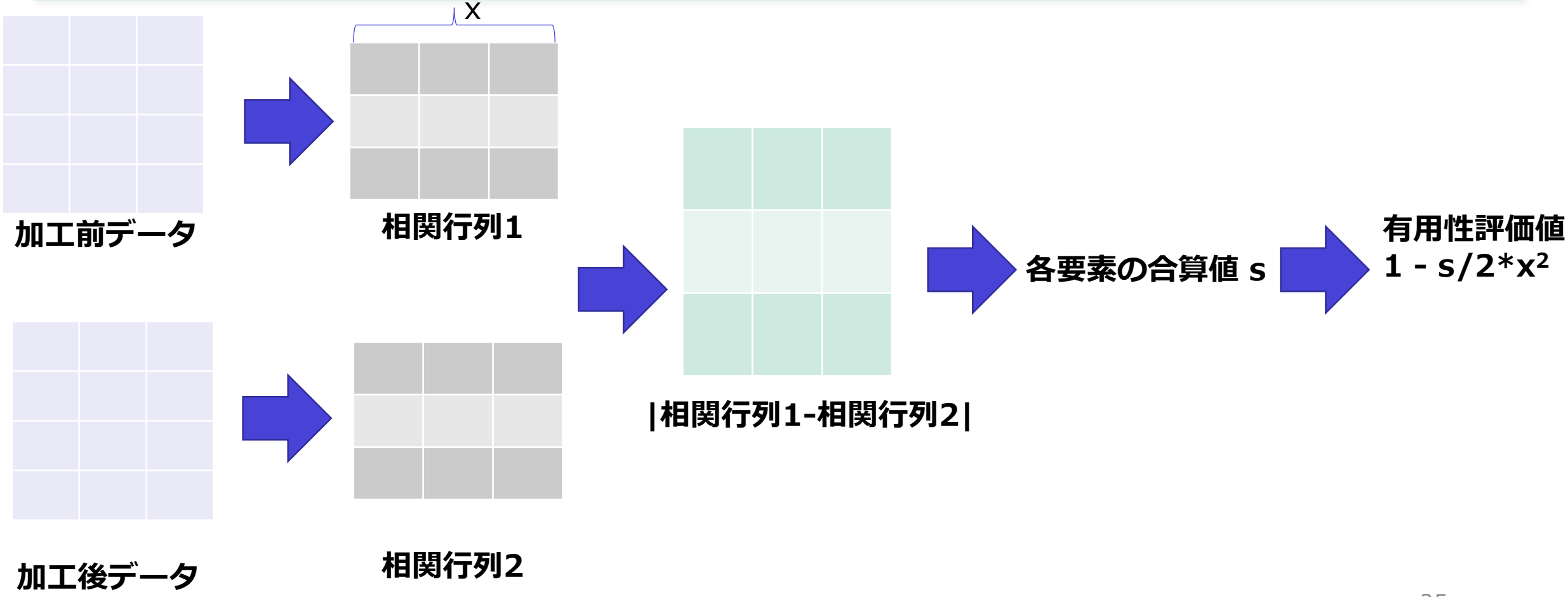
```
def odds(df):
    model = smf.glm(formula='COVID ~
AGE+GENDER+RACE+INCOME+EDUCATION+VETERAN+NOH+HTN+DM+IHD+CKD+COPD+CA', data=df, family=
sm.families.Binomial() )
    res = model.fit()
    df2 = pd.DataFrame(res.params, columns=['Coef'])
    df2['OR'] = np.exp(res.params)
    df2['pvalue'] = res.pvalues
    return(df2)

def oddsDiff(df_anon, df_orig):
    df_anon.columns = ['AGE','GENDER','RACE','INCOME','EDUCATION','VETERAN','NOH','HTN','DM','IHD','CKD','COPD','CA']
    df_orig.columns = ['AGE','GENDER','RACE','INCOME','EDUCATION','VETERAN','NOH','HTN','DM','IHD','CKD','COPD','CA']
    df_anon['COVID'] = 1
    df_orig['COVID'] = 1
    df_utility = pd.read_csv(os.path.join(os.path.dirname(__file__),"utility.csv")
    df_orig = pd.concat([df_orig, df_utility])
    df_anon =pd.concat([df_anon, df_utility])
    df_anon = odds(df_anon)['OR']
    df_orig = odds(df_orig)['OR']
    return([max(1-((df_anon - df_orig).abs().max())/20, 0), max(1-((df_anon - df_orig).abs().mean())/20, 0)])
```

配布済

Score C

Score D

有用性評価: 相関行列の差



有用性評価: 相関行列の差

```
def corrDiff(dfOD, dfAD):
    #return 1 - (dfOD.corr()-
dfAD.corr()).abs().sum().sum()/(dfOD.shape[0]*dfOD.shape[1]*2)
```

```
    return 1 - (dfOD.corr()-
dfAD.corr()).abs().sum().sum()/(dfOD.shape[1]*dfOD.shape[1]*2)
```

Score B

2022/09/22修正

有用性評価: 集計数の差

		加工前データ	加工後データ	差分
年齢	20代	23	31	8
	30代	10	11	1
	…	21	23	10
	80以上	12	32	20
	不明（99）	12	31	19
人種	1 ヒスパニック系ではない白人	32	23	9
	…			
	4ヒスパニック系ではない黒人	31	23	8
	6 ヒスパニック系ではないアジア人	31	33	2
	7 その他	22	1	21
	99 不明	12	3	9
…				
CA	0	44	22	22
	1	11	13	2
	99	1	2	3737373737

合計値A　　合計値B

有用性評価値 1 – B/A

有用性評価: 集計数の差

```
def agDiff(dfOR, dfAD):
    retOR = 0
    retOR_RD =0
    for i  in [0 , 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12]:
        dfOR1=pd.DataFrame(dfOR.loc[:,i].value_counts().sort_index())
        dfRD1=pd.DataFrame(dfAD.loc[:,i].value_counts().sort_index())
        df_concat = pd.concat([dfOR1, dfRD1], axis=1).fillna(0)

        retOR2= abs(df_concat.iloc[:,0]).sum()

        retOR_RD2 = abs(df_concat.iloc[:,0]-df_concat.iloc[:,1]).sum()

        retOR=retOR+retOR2 #元データの集計数

        retOR_RD=retOR_RD+retOR_RD2 #元データと匿名後の集計数の差分

    return 1-retOR_RD/retOR
```

Score A

年代共通処理

```
dfOD[0]=dfOD[0].apply(lambda x:math.floor(x/10)*10)
dfAD[0]=dfAD[0].apply(lambda x:math.floor(x/10)*10)
```

お願い

- チームの代表者はCSS2022に参加登録を行い，最終日にプレゼンテーションをお願いします.
- ルール・システムなど、まだ検討中で変更するかもしれないことをご了承ください.

（留意事項） NHANESは倫理承認されており，CDCの趣旨に沿った分析には，追加の承認は不要であることをご承知おきください