

# Stataを使ったデータ加工の実践 ～院内がん登録とEFファイルから 診療の質にせまる

平成25年9月20日

国立がん研究センターがん対策情報センター  
がん政策科学研究部  
東 尚弘

# 最近の診療の質評価

- 標準診療実施の検証が課題
- 電子データの整備は進行
- 個別患者の特殊性に対応が必要
  - 今後集計を元に、個々の症例の解析が重要
  - 誤分類をいかに低減するかが鍵

# データ源1： 院内がん登録

- 診療された全がんの基本情報をDB化  
→ 診断～治療初期の患者の病状
- 国指定の397がん診療連携拠点病院で必須  
→ 全がんの3分の2をカバー
- 綿密なコードルール＋研修修了者が入力  
→ データの質が期待できる

## データ源2:

DPC (Diagnosis-Procedure Combination)

E/Fファイル

- 医療機関からの診療報酬の入院分について、1日包括料金システムを検証する目的
- 全国で最も標準化された診療データ
- 診療行為がコード化されて列挙
- 薬も診療行為も1レコード
- 入院に続いて外来もデータ化

# DPC・E/Fファイルの構造

1レコード1診療行為(コード)

患者ID、診療行為、日付がある

患者ID	診療行為(コード)	日付
1	血液検査A	20111002
1	手術B	20111002
1	注射薬C	20111002
2	手術D	20111004
2	放射線治療	20111010
3	内視鏡検査	20111012
3	血液検査	20111012
4	レントゲン検査	20111011
4	処方薬X	20111025
4	内視鏡検査	20111021

# さて、これらを使って医療を検討

例えば・・

1. 麻薬開始時の下剤処方率
2. 乳房温存術後7月以内の放射線療法施行率

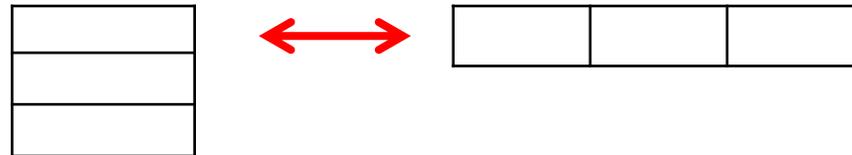
他

3. 200人分のファイルをID毎に分割して、ID自体をファイル名として保存

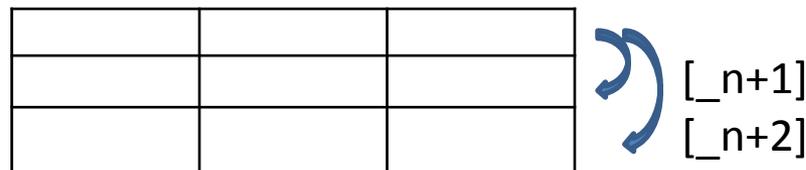
Stataをつかってどのように算定するか

# Stataで検討がしやすい理由

- ① reshapeコマンドにより、  
縦長のデータ  $\leftrightarrow$  横長データ  
の加工がしやすい



- ② `x[_n+○]`で、前後レコード(行)が参照可能



# 1.麻薬処方開始時の下剤処方

- 麻薬の多い副作用:便秘
- 麻薬を初めて使用するときには下剤を処方
- 普通に考えると

reshape wide → 変数名変換

ID	診療行為	開始日
1	麻薬	5/7
1	下剤	5/7
2	麻薬	6/6
3	麻薬	9/19
4	下剤	2/4
5	麻薬	2/10
6	麻薬	10/10
6	下剤	10/25



ID	麻薬開始日	下剤処方日	標準実施
1	5/7	5/7	O
2	6/6	-	X
3	9/19	-	X
4		2/4	-
5	2/10	-	X
6	10/10	10/25	X

ID: 患者番号

# しかし

- 下剤は特に高齢者でよく処方  
麻薬開始前から定期服用していることも  
→
- 麻薬の開始日以前に処方があればOK？
- 中止されていたりするかも？  
→
- 麻薬開始前31日前～開始日に処方でOK

# 麻薬開始前31日前～開始日に処方

- 麻薬開始日が基準
- その前何回かの下剤処方日を検討

ID	麻薬処方日
1	5/7
1	6/10
1	6/17
2	9/19
2	10/4
2	2/10
3	10/10
3	10/25

sort ID 麻薬処方日  
by ID: keep if \_n==1



ID	麻薬開始日
1	5/7
2	9/19
3	10/10

ID	下剤処方日
1	4/10
1	5/10
1	6/10
2	9/19
2	10/6
2	11/3
3	10/18

麻薬ファイルにflgの上日付変数をまとめ、  
appendで結合、

# 2つのデータをAppendで結合

ID	日付	麻薬flg
1	4/10	
1	5/7	1
1	5/10	
1	6/10	
2	9/19	
2	9/19	1
2	10/6	
2	11/3	
3	10/10	1
3	10/18	



O



O



x

```
gsort ID 日付 -麻薬flg
```

```
gen pass=0
```

```
replace pass=1 if /*
```

```
*/ 麻薬flg==1 /*
```

```
*/ & ID==ID[_n-1] /*
```

```
*/ & 日付<=日付[_n-1]+31
```

後、

```
keep if 麻薬flg==1
```

```
sum pass
```

# 2.乳房温存術後の放射線療法

普通は単純に考えると

reshape wide

ID	診療行為	日付
1	手術	5/7
1	放射線	6/10
2	手術	6/6
3	手術	9/19
4	放射線	2/4
5	手術	2/10
6	手術	10/10
6	放射線	10/25



ID	手術日	放射線日	標準実施
1	5/7	6/10	O
2	6/6	-	X
3	9/19	-	X
4		2/4	-
5	2/10	-	X
6	10/10	10/31	O

ID: 患者番号

# しかし・・・

- 課題

- 手術がいくつもある可能性

- 温存術後に切除術があったら除外
- 放射線療法が先にあったら、反対側

- (7ヶ月以内という期限)・・・今回は省略

# 実際には

ID	診療行為	日付
1	放射線	5/7
1	温存手術	6/1
1	放射線	7/11
2	切除手術	2/10
2	温存手術	10/10
2	放射線	11/25
3	温存手術	6/6
3	切除手術	9/19
3	放射線	11/4
4	温存手術	10/10
4	温存手術	11/25
4	放射線	12/25

## 必要なルール

- 温存手術よりも前の切除術や放射線は無視  
∴ おそらく反対側への処置 (1,2)
- 温存のあとに切除術があったら除外  
∴ 追加切除 (3)
- 温存術が2回あったら後をとる  
∴ 反対側なのか、追加かは不明 (4)

## プログラム手順

- ①温存術とりだし、各患者最後の手術のみ
- ②切除術とりだし、1回目、2回目とwideする
- ③IDで結合、切除1回目が温存より前ならすてて、2回目の日付を1回目に移す。・・・ファイルA
- ④放射線をとりだし、Aの下につける
- ⑤日付でソート
- ⑥温存のあとの同一IDのレコードに放射線があればO、無ければ×

# プログラム(①～②)

(わかりやすさのため、変数のみ日本語を使っています)

```
preserve // ①
  keep if 診療=="温存"
  sort ID 日付
  by ID: keep if _n==_N
  save temp, replace
```

①温存術とりだし、各患者最後の手術のみ

```
restore
keep if 診療=="切除" // ②
sort ID 日付
by ID: gen a=_n
reshape wide 日付, i(ID) j(a)
```

②切除術とりだし、1回目、2回目とwideする  
(注、切除術は両側で2回までしかない)

# プログラム(③)

```
merge 1:1 ID using temp, /*
*/ keep(match using)
replace 日付1=日付2 if 日付1<日付
replace 日付2=. if 日付1==日付2
keep if 日付1==. | 日付1<日付
gen 対象=1
save ファイルA, replace
```

IDで結合、

切除1回目が温存より  
前ならすてて、2回目の  
日付を1回目に移す。  
分母を表す印をつける  
・・・ファイルA

# プログラム (④～⑥)

```
use 元のデータ, clear //④
keep if 診療行為="放射線"
append using "ファイルA"
sort 日付 //⑤
gen PASS=0
replace PASS=1 if 対象==1 & /*
*/ ID[ID[_n+1] //⑥
keep if 対象==1
```

④放射線をとりだし、Aの下につける

⑤日付でソート

⑥温存のあとの同一IDのレコードに放射線があれば0、無ければ×

これでPASS 1/0を検討

# 3. 他の使い方

- ファイルをID毎に分割、ID.dtaとして保存

ID	診療行為	日付
13	放射線	5/7
13	温存手術	6/1
13	放射線	7/11
27	切除手術	2/10
27	温存手術	10/10
27	放射線	11/25
32	温存手術	6/6
32	切除手術	9/19
32	放射線	11/4
41	温存手術	10/10
41	温存手術	11/25
41	放射線	12/25

13.dta

ID	診療行為	日付
13	放射線	5/7
13	温存手術	6/1

27.dta

ID	診療行為	日付
27	切除手術	2/10
27	温存手術	10/10
27	放射線	11/25

32.dta

ID	診療行為	日付
32	温存手術	6/6
32	切除手術	9/19
32	放射線	11/4

41.dta

ID	診療行為	日付
41	温存手術	10/10
41	温存手術	11/25
41	放射線	12/25

4ファイルぐらいなら簡単。200あったら？

# 繰り返し処理で自動化

```
egen a=group(ID)          /*番号をつける*/
forvalues i=1/200 {       /*200人*/
  preserve                /*退避*/
  keep if a==`i'          /*対象をのこす*/
  local file=ID           /*IDをローカルマクロ*/
  save "`file'.dta", replace /*ファイルに保存*/
  restore
}
```

# まとめ

- Stataは統計だけでなくデータ処理に利用可能
  - 複雑な臨床状況を検討するに最適
- 繰り返し処理などで作業の効率化も