日付と時間の新しい関数

What is the date?

```
. display %td today()
20apr2021
```

How many days until summer?

```
. display datediff(today(), td(20jun2021), "day")
61
```

How many days are in this month?

```
. display daysinmonth(td(20apr2021))
30
```

概要

Stata 17 では、日付と時間を正確に操作するための便利な関数が追加されました。大きく 3 つに分類されます。

1. **日時期間関数**：年齢や日時間の差分など、期間を計算する
2. **相対日付関数**：ある日付を基準にした次の誕生日のように、相対的な日付を計算する
3. **日時要素関数**：日時データから任意の要素を抽出する

✧ 日時関数の詳細は、PDF マニュアル[D] Data Management をご参照ください。Stata のメニューの「ヘルプ ＞ 英文 PDF マニュアル」から開きます。

✧ 本文中のコマンドをコピーし、Stata のコマンドウィンドウに貼り付けて実行できます。全ての操作のコマンドは、do ファイル date_time_functions.do にまとめられています。

例題 1：日付の期間

例題 1-1：年齢の計算

- 2 月 29 日生まれの人は、うるう年でない年はいつ誕生日を迎えるでしょうか？これは国や地域によって異なり、イギリスでは 3 月 1 日、台湾では 2 月 28 日です。
- 新しい関数 **age()** は、うるう年を処理するためのオプションの引数を使用して、年齢を整数で示します。
- 例えば次のコマンドを実行すると、結果は「17」と表示されます。うるう年である 2000 年 2 月 29 日生まれの人が、2018 年 2 月 28 日に何歳であるかを求めています。うるう年でない年の誕生日は 3 月 1 日と指定しています。※**td()** は Stata に日付を認識させる関数です。**日月年** の順に指定します。

```
display age(td(29feb2000), td(28feb2018), "01mar")
```

（結果）

```
17
```

- うるう年でない年の誕生日を 2 月 28 日と指定すると、結果は「18」と表示されます。

```
display age(td(29feb2000), td(28feb2018), "28feb")
```

（結果）

```
18
```

例題 1-2：一般的な日付の計算

- 関数 **datediff()** は、さまざまな時間単位（年、月、日）で、一般的な日付間の差分を計算します。値は最も近い整数に切り捨てられます。関数 **age()** は **datediff()** の特別なケースです。
- 例えば、2000 年 7 月 31 日に結婚し、結婚 10 周年の 2010 年 7 月 31 日までに何日経過したかを調べることができます。次のコマンドを実行すると、結果は「3652」と表示されます。

```
display datediff(td(31jul2000), td(31jul2010), "day")
```

（結果）

```
3652
```

例題 1-3：詳細な日付の計算

- 関数 **age_frac()** と **datediff_frac()** は年齢、および年、月、日単位での日付の差分を正確に計算します。
- 例えば、ある期間の正確な月数を計算する場合、Stata はその年が 365 日か 366 日か、その月が 28 日、29 日、30 日、31 日かを調べ、正確に差分を計算します。
- 次のコマンドを実行すると、2019 年 11 月 17 日から今日までの正確な月数が表示されます。

```
display datediff_frac(td(17nov2019), today(), "month")
```

（結果は実行した日によって変わります。2021 年 9 月 14 日に実行した場合は **21.903226** となります。）

- 上記の結果は、COVID-19 が最初に報告されてから本日までの正確な月数です。生存モデルでは正確な日付期間を計算することが重要です。**datediff_frac()** と **age()** は、互換性のある方法で差分を計算します。したがって、生存モデルで **datediff_frac()** の日時を使用し、モデルの予測子として **age()**の年齢を使用すると、一貫性が保たれます。

例題 2：時間の期間

- 地球の回転速度にはむらがあり、地球の自転によって決まる時刻と原子時計によって決まる時刻のずれを修正するために、「うるう秒」が実施されます。うるう秒による調整は 1972 年以降行われています。Stata では、うるう秒ありの場合となしの場合の両方を計算することができます。
- 関数 **Clockdiff()** と **clockdiff()** はそれぞれ、うるう秒ありの場合となしの場合について、指定した任意の時間単位で差分を計算します。値は最も近い整数に切り捨てられます。
- 関数 **Clockdiff_frac()** と **clockdiff_frac()** は、この値の切り捨てを行わず正確に計算します。例えば、天体が完全な軌道を 1 周するとき、その開始と終了のタイムスタンプ（うるう秒あり）を記録したとします。開始時刻は **begin** という変数、終了時刻は **end** という変数に記録されています。次のコマンドを実行すると、軌道周期の日数が計算されます。

```
generate double period = Clockdiff_frac(begin, end, "day")
```

（サンプルデータなし、コマンドの紹介のみ）

例題 3：相対日付

- Stata 内部では、日時データは数値に変換されて扱われます。これは 1960 年 1 月 1 日 0 時を基準とした相対的な数値です。これによって、ある日付を基準にした誕生日や、ある日付の月を基準にした日付などの計算を行うことができます。

例題 3-1：誕生日の計算

- 関数 **birthday()** は、指定された年の誕生日を求めます。例えば、2000 年 2 月 29 日生まれの人が、2018 年はいつ誕生日を迎えるかを表示するには、次のコマンドを実行します。

```
display birthday(td(29feb2000),2018)
```

（結果）

```
21244
```

- 結果の「21244」は Stata 形式の日付の数値です。次のコマンドのように日付の表示フォーマットを指定することも可能です。**%td** は「日月年」の形式で日付を表示します。
※数値のフォーマットの詳細は、PDF マニュアル[D] Data Management の format をご参照ください。
デフォルトでは、2 月 29 日生まれの人のうるう年でない年の誕生日は 3 月 1 日と設定されています。結果から、2018 年はうるう年ではなく、3 月 1 日に誕生日を迎えることがわかります。

```
display %td birthday(td(29feb2000),2018)
```

（結果）

```
01mar2018
```

- うるう年でない年の誕生日を 2 月 28 日とする場合は、次のように指定します。

```
display %td birthday(td(29feb2000),2018, "28feb")
```

（結果）

```
28feb2018
```

- 関数 **previousbirthday()** と **nextbirthday()** はそれぞれ、指定された日付を基準にした、前の誕生日と次の誕生日を求めます。

```
display %td previousbirthday(td(29feb2000),td(01apr2018))
```

（結果：2018 年 4 月 1 日を基準として、2000 年 2 月 29 日生まれの人の前の誕生日は 2018 年 3 月 1 日）

```
01mar2018
```

```
display %td nextbirthday(td(29feb2000),td(01apr2018))
```

（結果：2018 年 4 月 1 日を基準として、2000 年 2 月 29 日生まれの人の次の誕生日は 2019 年 3 月 1 日）

```
01mar2019
```

例題 3-2：月の日付の計算

- 関数 **daysinmonth()** 、**firstdayofmonth()**、および **lastdayofmonth()** は指定した月について、それぞれ、その月の日数、その月の最初の日、およびその月の最後の日を求めます。

```
display daysinmonth(td(14sep2021))
```

（結果：2021 年 9 月の日数は 30 日）

```
30
```

```
display %td firstdayofmonth(td(14sep2021))
```

（結果：2021 年 9 月の最初の日は 2021 年 9 月 1 日）

```
01sep2021
```

```
display %td lastdayofmonth(td(14sep2021))
```

（結果：2021 年 9 月の最後の日は 2021 年 9 月 30 日）

```
30sep2021
```

例題 3-3：うるう年の計算

- うるう年は約 4 年に一度あります。関数 **isleapyear()**、**previousleapyear()**、および **nextleapyear()**は指定された年について、それぞれ、その年がうるう年であるかどうか、その年の前のうるう年、その年の次のうるう年を求めます。

```
display isleapyear(2000)
```

（結果：うるう年であれば 1、そうでなければ 0。2000 年はうるう年。）

```
1
```

```
display previousleapyear(2000)
```

（結果：2000 年の前のうるう年は 1996 年）

```
1996
```

```
display nextleapyear(2000)
```

（結果：2000 年の次のうるう年は 2004 年）

```
2004
```

例題 4：日時要素

- 関数 **datepart()**、**Clockpart()**、および **clockpart()** は、指定された日時の任意の要素（年、月、日、時、分、秒、ミリ秒）を整数値で求めます。**Clockpart()** はうるう秒に対応しています。**clockpart()** は対応していません。

```
display datepart(td(14sep2021), "day")
```

（結果：2021 年 9 月 14 日の日の要素は 14）

```
14
```

```
display clockpart(tc(14sep2021 10:11:12), "minute")
```

（結果：2021 年 9 月 14 日 10 時 11 分 12 秒の分の要素は 11）

```
11
```

- 関数 **now()** は現在の時刻を、**today()** は今日の日付を返します。

```
display %tc now()
```

（結果は実行した日時によって変わります。2021 年 9 月 14 日 10 時 11 分 12 秒に実行した場合は **14sep2021 10:11:12** となります。）

```
display %td today()
```

（結果は実行した日によって変わります。2021 年 9 月 14 日に実行した場合は **14sep2021** となります。）

参考文献

- Pope Gregory XIII. 1582. *Inter gravissimas.*
- Sullivan, A. 1923. *The Pirates of Penzance or the Slave of the Duty*, libretto by W.S. Gilbert, G. Schimer.
- 国立天文台ウェブサイト　よくある質問　質問 4-3）「うるう秒」ってなに？ https://www.nao.ac.jp/faq/a0403.html