

CUENTAS DIFÍCILES

IMPLEMENTACIÓN DEL COMANDO **cuentacot**

George Vega Y.

Superintendencia de Pensiones

enero de 2012



CONTENIDOS

INTRODUCCIÓN

IMPLEMENTACIÓN

EJEMPLOS Y CASOS DE USO



CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad.

CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad. Es por eso que, al momento de realizar cuentas, es necesario recordar que:

CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad. Es por eso que, al momento de realizar cuentas, es necesario recordar que:

- ▶ Los datos observados no corresponden a periodos continuos perse.

CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad. Es por eso que, al momento de realizar cuentas, es necesario recordar que:

- ▶ Los datos observados no corresponden a periodos continuos perse.
- ▶ Si se restringen a ventanas de tiempo, el problema anterior se acentúa

CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad. Es por eso que, al momento de realizar cuentas, es necesario recordar que:

- ▶ Los datos observados no corresponden a periodos continuos perse.
- ▶ Si se restringen a ventanas de tiempo, el problema anterior se acentúa
- ▶ La solución de expandir los datos implica una disminución en el rendimiento y memoria disponible.

CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad. Es por eso que, al momento de realizar cuentas, es necesario recordar que:

- ▶ Los datos observados no corresponden a periodos continuos perse.
- ▶ Si se restringen a ventanas de tiempo, el problema anterior se acentúa
- ▶ La solución de expandir los datos implica una disminución en el rendimiento y memoria disponible. Y aún así sigue siendo complejo

CUENTAS COMPLEJAS

¿POR QUÉ CUENTAS COMPLEJAS?

Cada cuenta de cotizaciones tiene su grado de complejidad. Es por eso que, al momento de realizar cuentas, es necesario recordar que:

- ▶ Los datos observados no corresponden a periodos continuos perse.
- ▶ Si se restringen a ventanas de tiempo, el problema anterior se acentúa
- ▶ La solución de expandir los datos implica una disminución en el rendimiento y memoria disponible. Y aún así sigue siendo complejo
- ▶ Las multicotizaciones es otro problema, al igual que los cambios de empleador o tipo de contrato.

cuentacot

¿Qué es `cuentacot`?

Es un comando de Stata diseñado para abordar aquellas problemáticas.

¿Qué es `cuentacot`?

Es un comando de Stata diseñado para abordar aquellas problemáticas. Ideado inicialmente para el uso en la Base de Datos del Seguro de Censantía, las cuentas que el algoritmo realiza funcionan para cualquier base de datos compuestas por observaciones mensuales de individuos,

¿Qué es `cuentacot`?

Es un comando de Stata diseñado para abordar aquellas problemáticas. Ideado inicialmente para el uso en la Base de Datos del Seguro de Censantía, las cuentas que el algoritmo realiza funcionan para cualquier base de datos compuestas por observaciones mensuales de individuos, independientemente tengan que ver o con cotizaciones o con remuneraciones.

CONTENIDOS

INTRODUCCIÓN

IMPLEMENTACIÓN

EJEMPLOS Y CASOS DE USO

CARACTERÍSTICAS GENERALES

- ▶ Para uso de bases de datos de remuneraciones/cotizaciones.

CARACTERÍSTICAS GENERALES

- ▶ Para uso de bases de datos de remuneraciones/cotizaciones.
- ▶ Genera variables de tipo *dicotómicas* en base a algoritmo de cuenta de número de cotizaciones para cada momento del tiempo (observación).

CARACTERÍSTICAS GENERALES

- ▶ Para uso de bases de datos de remuneraciones/cotizaciones.
- ▶ Genera variables de tipo *dicotómicas* en base a algoritmo de cuenta de número de cotizaciones para cada momento del tiempo (observación).
- ▶ Las variables principales a especificar en el contador corresponden a **periodo** (campo numérico de fecha en formato YYYYMM), **persona** (variable id del individuo) y, opcionalmente, **empleador** (variable id del empleador).

CARACTERÍSTICAS GENERALES

- ▶ Para uso de bases de datos de remuneraciones/cotizaciones.
- ▶ Genera variables de tipo *dicotómicas* en base a algoritmo de cuenta de número de cotizaciones para cada momento del tiempo (observación).
- ▶ Las variables principales a especificar en el contador corresponden a **periodo** (campo numérico de fecha en formato YYYYMM), **persona** (variable id del individuo) y, opcionalmente, **empleador** (variable id del empleador).
- ▶ En base al campo **periodo**, el algoritmo determina qué observaciones son válidas para el cálculo en los resultados de **menncot** y **menndiscont** para cada momento del tiempo¹.

¹Este punto es importante pues el comando considera ventanas de tiempo entre observaciones de un mismo individuo; por lo que 3 observaciones seguidas no son consideradas como 3 remuneraciones/cotizaciones continuas necesariamente, depende del periodo en que se realiza cada una.

CARACTERÍSTICAS GENERALES

- ▶ Para uso de bases de datos de remuneraciones/cotizaciones.
- ▶ Genera variables de tipo *dicotómicas* en base a algoritmo de cuenta de número de cotizaciones para cada momento del tiempo (observación).
- ▶ Las variables principales a especificar en el contador corresponden a **periodo** (campo numérico de fecha en formato YYYYMM), **persona** (variable id del individuo) y, opcionalmente, **empleador** (variable id del empleador).
- ▶ En base al campo **periodo**, el algoritmo determina qué observaciones son válidas para el cálculo en los resultados de **menncot** y **menndiscont** para cada momento del tiempo¹.
- ▶ Si alguna de las variables a generar ya existe, por defecto el comando la reemplaza por la nueva.

¹Este punto es importante pues el comando considera ventanas de tiempo entre observaciones de un mismo individuo; por lo que 3 observaciones seguidas no son consideradas como 3 remuneraciones/cotizaciones continuas necesariamente, depende del periodo en que se realiza cada una.

CARACTERÍSTICAS GENERALES

- ▶ Para uso de bases de datos de remuneraciones/cotizaciones.
- ▶ Genera variables de tipo *dicotómicas* en base a algoritmo de cuenta de número de cotizaciones para cada momento del tiempo (observación).
- ▶ Las variables principales a especificar en el contador corresponden a **periodo** (campo numérico de fecha en formato YYYYMM), **persona** (variable id del individuo) y, opcionalmente, **empleador** (variable id del empleador).
- ▶ En base al campo **periodo**, el algoritmo determina qué observaciones son válidas para el cálculo en los resultados de **menncot** y **menndiscont** para cada momento del tiempo¹.
- ▶ Si alguna de las variables a generar ya existe, por defecto el comando la reemplaza por la nueva.
- ▶ El comando conserva el orden original de la base de datos antes de ser ejecutado.

¹Este punto es importante pues el comando considera ventanas de tiempo entre observaciones de un mismo individuo; por lo que 3 observaciones seguidas no son consideradas como 3 remuneraciones/cotizaciones continuas necesariamente, depende del periodo en que se realiza cada una.

SINTAXIS

El comando requiere de 2 variables al menos para poder funcionar.

SINTAXIS

El comando requiere de 2 variables al menos para poder funcionar. `periodo` (variable numérica en formato YYYYMM) y `persona` (id del individuo, típicamente el correl).

SINTAXIS

El comando requiere de 2 variables al menos para poder funcionar. `periodo` (variable numérica en formato YYYYMM) y `persona` (id del individuo, típicamente el correl).

`cuentacot` *periodo persona, dis //* Generaría variable `cuenta disc`

SINTAXIS

El comando requiere de 2 variables al menos para poder funcionar. *periodo* (variable numérica en formato YYYYMM) y *persona* (id del individuo, típicamente el correl).

cuentacot *periodo persona*
[*empleador*], [*keep cuentas parametros variables.adicionales*]

CUENTAS DISPONIBLES I

- discontinuas Genera variable de cuenta de cotizaciones discontinuas y la almacena como **cuenta_cot_dis**.
- continuas Genera variable dicotómica en base a cuenta de cotizaciones continuas, donde **cotiza_continuas'ncot'** es igual a 1 si **cuenta_cot_cont** \geq **ncot**.
- contempleador Genera variable dicotómica en base a cuenta de cotizaciones continuas con el mismo empleador, donde **cot_'ncot'_emp_cont** es igual a 1 si **cuenta_cot_'ncot'_emp_cont** \geq **ncot**.
- empleador Genera variable dicotómica en base a cuenta de cotizaciones discontinuas con el mismo empleador, donde **cot_'ncot'_emp_dis** es igual a 1 si **'cuenta_'ncot'_ult_cot_emp'** \geq **ncot**.
- menncont Genera variable dicotómica en base a cuenta de cotizaciones continuas en los últimos nper periodos, donde **cot_'ncot'_en_'nper'_cont** es igual a 1 si **cuenta_cot_'ncot'_en_'nper'_cont** \geq **ncot**
El resultado se puede leer como
"ncot cotizaciones continuas en los últimos nper periodos"

CUENTAS DISPONIBLES II

menndiscont Genera variable dicotómica en base a cuenta de cotizaciones discontinuas en los últimos `nper` periodos, donde `cot_‘ncot’_en_‘nper’_dis` es igual a 1 si `cuenta_cot_‘ncot’_en_‘nper’_dis` \geq `ncot`.
El resultado se puede leer como
“ncot cotizaciones discontinuas en los últimos nper periodos”

PARÁMETROS Y VARIABLES ADICIONALES

PARÁMETROS Y VARIABLES ADICIONALES

► Parámetros

- ncot Entero. Número de cotizaciones (por defecto 12) a considerar en el cálculo de **continuas**, **contemp**, **menncot** y **menndisc**.
- nper Entero. Número de periodos (por defecto 24) a considerar en el cálculo de **menncot** y **menndisc**.

PARÁMETROS Y VARIABLES ADICIONALES

► Parámetros

ncot Entero. Número de cotizaciones (por defecto 12) a considerar en el cálculo de **continuas**, **contemp**, **menncot** y **menndisc**.

nper Entero. Número de periodos (por defecto 24) a considerar en el cálculo de **menncot** y **menndisc**.

► Variables Adicionales

nsolic(*varname*) Variable que contiene el número de solicitud. Se utiliza para considerar solicitudes de beneficio de modo tal de llevar los contadores de cotizaciones continuas a 0. El comando identifica un periodo como solicitud cuando **nsolic** deja de ser *missing*

tipcon(*varname*) Variable que contiene el tipo de contrato. Se utiliza para considerar el tipo de contrato al momento de la cuenta de cotizaciones continuas (**continuas**). Realizando cuenta de cotizaciones sólo cuando **tipcon** es igual a 1, i.e. personas con contrato indefinido.

CUADRO 1: Variables y parámetros utilizados por cada cuenta

Cuentas	Variables			Parámetros	
	nsolic	tipcon	empleador	nper	ncot
<u>dis</u> continuas	X				
<u>contin</u> uas	X	X			X
<u>contem</u> pleador	X	X	X		X
<u>emple</u> ador	X	X	X		X
<u>menndis</u> cont	X			X	X
<u>mennc</u> ont	X			X	X

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:



ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)
- ▶ Individuo que deja de cotizar en 1 periodo [▶ ver](#)

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)
- ▶ Individuo que deja de cotizar en 1 periodo [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de empleador [▶ ver](#)

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)
- ▶ Individuo que deja de cotizar en 1 periodo [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de empleador [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de tipo de contrato [▶ ver](#)

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)
- ▶ Individuo que deja de cotizar en 1 periodo [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de empleador [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de tipo de contrato [▶ ver](#)
- ▶ Individuo con doble cotización en 1 periodo [▶ ver](#)

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)
- ▶ Individuo que deja de cotizar en 1 periodo [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de empleador [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de tipo de contrato [▶ ver](#)
- ▶ Individuo con doble cotización en 1 periodo [▶ ver](#)
- ▶ Individuo con solicitud de desempleo [▶ ver](#)

ESCENARIOS DE PRUEBA

Junto con mostrar el uso del comando, revisaremos su comportamiento en los siguientes escenarios:

- ▶ Individuo que cotiza siempre [▶ ver](#)
- ▶ Individuo que deja de cotizar en 1 periodo [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de empleador [▶ ver](#)
- ▶ Individuo que cotiza siempre pero cambia de tipo de contrato [▶ ver](#)
- ▶ Individuo con doble cotización en 1 periodo [▶ ver](#)
- ▶ Individuo con solicitud de desempleo [▶ ver](#)
- ▶ Individuo con grandes lagunas [▶ ver](#)

INDIVIDUO QUE COTIZA SIEMPRE

```
cuentacot periodo idper, nc(3) dis cont k
```


INDIVIDUO QUE DEJA DE COTIZAR EN 1 PERIODO

```
cuentacot periodo idper, nc(3) dis cont k
```


INDIVIDUO QUE COTIZA SIEMPRE PERO CAMBIA DE EMPLEADOR

```
cuentacot periodo idper idemp, nc(3) dis cont emp contemp k
```

INDIVIDUO QUE COTIZA SIEMPRE PERO CAMBIA DE EMPLEADOR

`cuentacot periodo idper idemp, nc(3) dis cont emp contemp k`

CUADRO 4: Cotizaciones de Camila

nombre	periodo	idper	idemp	tipcon	nsolic	marca	cuenta~ dis	cuenta~ cont	cuenta~ cont_emp	cuenta_3_ult_cot_emp
Camila	201002	133	154	1			1	1	1	1
Camila	201003	133	154	1			2	2	2	2
Camila	201004	133	154	1			3	3	3	3
Camila	201005	133	154	1			4	4	4	4
Camila	201006	133	154	1			5	5	5	5
Camila	201007	133	158	1		X	6	6	1	1
Camila	201008	133	158	1			7	7	2	2
Camila	201009	133	158	1			8	8	3	3
Camila	201010	133	158	1			9	9	4	4
Camila	201011	133	158	1			10	10	5	5
Camila	201012	133	158	1			11	11	6	6
Camila	201101	133	158	1			12	12	7	7

▶ volver

INDIVIDUO QUE COTIZA SIEMPRE PERO CAMBIA DE TIPO DE CONTRATO

```
cuentacot periodo idper idemp, nc(3) dis cont emp contemp k
```

INDIVIDUO QUE COTIZA SIEMPRE PERO CAMBIA DE TIPO DE CONTRATO

`cuentacot periodo idper idemp, nc(3) dis cont emp contemp k`

CUADRO 5: Cotizaciones de Gustavo

nombre	periodo	idper	idemp	tipcon	nsolic	marca	cuenta~ dis	cuenta~ cont	cuenta~ cont_emp	cuenta_3_ult_cot_emp
Gustavo	201008	189	158	1			1	1	1	1
Gustavo	201009	189	158	1			2	2	2	2
Gustavo	201010	189	158	1			3	3	3	3
Gustavo	201011	189	158	1			4	4	4	4
Gustavo	201012	189	158	1			5	5	5	5
Gustavo	201101	189	158	1			6	6	6	6
Gustavo	201102	189	158	0		X	7	1	1	1
Gustavo	201103	189	158	0			8	2	2	2
Gustavo	201104	189	158	0			9	3	3	3
Gustavo	201105	189	158	0			10	4	4	4
Gustavo	201106	189	158	0			11	5	5	5

▶ volver

INDIVIDUO CON DOBLE COTIZACIÓN EN 1 PERIODO

```
cuentacot periodo idper idemp, nc(3) dis cont emp contemp k
```

INDIVIDUO CON DOBLE COTIZACIÓN EN 1 PERIODO

`cuentacot periodo idper idemp, nc(3) dis cont emp contemp k`

CUADRO 6: Cotizaciones de Paola

nombre	periodo	idper	idemp	tipcon	nsolic	marca	cuenta~ dis	cuenta~ cont	cuenta~ cont_emp	cuenta_3.ult_cot.emp
Paola	201001	134	162	1			1	1	1	1
Paola	201002	134	162	1			2	2	2	2
Paola	201003	134	162	1			3	3	3	3
Paola	201004	134	162	1			4	4	4	4
Paola	201005	134	162	1			5	5	5	5
Paola	201006	134	162	1			6	6	6	6
Paola	201006	134	162	1		X	6	6	6	6
Paola	201007	134	162	1			7	7	7	7
Paola	201008	134	162	1			8	8	8	8
Paola	201009	134	162	1			9	9	9	9
Paola	201010	134	162	1			10	10	10	10

▶ volver

INDIVIDUO CON SOLICITUD DE DESEMPLEO

Primero sin solicitud

```
cuentacot periodo idper idemp, nc(3) np(6) dis cont menncont menndis k
```


INDIVIDUO CON SOLICITUD DE DESEMPLEO

Ahora con solicitud

```
cuentacot periodo idper idemp, nc(3) np(6) ns(nsolic) dis cont menncont menndis k
```


INDIVIDUO CON GRANDES LAGUNAS

```
cuentacot periodo idper idemp, nc(3) np(6) ns(nsolic) dis cont menncont menndis k
```

