

BIBLIOTECA DE PROGRAMAS

COCOA/TAG

COCOA* es el nombre dado a un paquete de programas diseñado para el tratamiento de textos y entre otras cosas calcula concordancias, recuento de palabras etc. a partir del texto suministrado por el usuario, en cualquier idioma, siempre que esté escrito en caracteres latinos.

Ha sido realizado por Godelieve LM Berry Rogghe del "Atlas Computer Laboratory" y T. D. Crawford de la Universidad de Cardiff.

Concordancias:

Cuando una palabra aparece más de una vez en el texto a tratar, cada ocurrencia puede ser listada en el orden en que aparece en el texto o por orden alfabético de acuerdo a su contexto a derecha o izquierda.

Las palabras claves, habitualmente serán listadas, siguiendo el orden alfabético invertido pudiendo limitarse las concordancias a aquellas palabras claves que tengan una determinada frecuencia o empiecen por determinados caracteres.

Además es posible listar unicamente cadenas de palabras que contengan 2 palabras consecutivas a una distancia menor que un numero de palabras prefijado por el usuario.

Recuentos de palabras:

Pueden pedirse los listados de las distintas palabras que aparecen en el texto de acuerdo a los criterios siguientes:

- Por orden alfabético
- Por orden alfabético invertido
- Por su orden de frecuencia de menor a mayor

TARJETAS DE CONTROL DEL PROGRAMA COCOA

Son aquellas tarjetas en las que especificaremos las opciones elegidas entre todo los posibles proporcionados por el programa COCOA, y hacen referencia al tipo de análisis que queremos que haga, como tipo de recuento elegido, sobre que palabras va realizar concordancias etc. también especificaremos información correspondiente a las características del texto a tratar como longitud en caracteres de cada letra, simbolos especiales utilizados para puntuación, etc.

(*) COCOA, Siglas correspondientes a "A word count and concordance generator"



Las reglas para la codificación de estas tarjetas vienen en el manual de usuarios del COCOA y está a disposición de los usuarios en la biblioteca del centro.

CODIFICACION DEL TEXTO EN TARJETAS

Cada tarjeta será considerada como una línea del texto sujeto a los siguientes convenios:

1) El signo + indica que la continuación del texto está al comienzo de una nueva tarjeta ignorando todo lo que hay a la derecha de ese signo + en la tarjeta donde aparece, pudiendo por tanto, utilizarse para introducir comentarios en el texto sin que estos tengan que procesarse.

Un espacio en blanco antes del signo + o antes del comienzo de la siguiente tarjeta indica que lo que comienza en esa tarjeta ha de considerarse como una nueva palabra, la ausencia de este espacio en blanco provoca la continuación de la palabra.

EJEMPLO: Si tuvieramos COLI+ y en la siguiente tarjeta aparece NA en las dos primeras columnas, considera el programa la palabra COLINA.

2) El signo / indica comienzo de línea en el texto es decir si en una tarjeta aparece un signo / lo que va a continuación se supone que es una nueva línea del texto, lo habitual es considerar que cada línea comienza en una nueva tarjeta.

3) Los simbolos < > se utilizan para hacer referencia al texto y no pueden utilizarse con ningún otro proposito.

Una referencia al texto consiste en:

- 1 un signo <
- 2 una letra
- 3 un espacio en blanco
- 4 la referencia
- 5 un signo >

EJEMPLO: A CALDERON ; P NMD

La letra que precede a la referencia puede ser cualquiera que convenga al usuario excepto la L que está reservada para hacer referencia al número de líneas.

4) Los simbolos (()) se utilizan para encerrar comentarios que el usuario no desea incluir en el recuento de palabras pero que a la hora de producir concordancias si aparecen en el contexto.

TARJETAS DEL SISTEMA NECESARIAS PARA UTILIZAR EL COCOA

```
// JOB
// EXEC COCOA
```

```

//COCOA.CONTROL DD *
tarjetas de control del programa COCOA
/*
//COCOA.TEXT DD *
tarjetas con el texto a analizar
/*
//

```

PROGRAMA TAG

Juntamente con el análisis que hace el programa COCOA, puede utilizarse el programa TAG, que utiliza como entradas la salida del programa COCOA y un diccionario dado por el usuario.

Este programa produce una clasificación de las palabras del texto analizado, en cada una de las categorías del diccionario que hayamos suministrado.

Codificación del diccionario

Un diccionario está formado por un conjunto de palabras agrupadas en categorías o palabras claves.

EJEMPLO: PASADO: AYER, FUI, ERA
 PRESENTE: HOY, SOY, AHORA, VAMOS

A la hora de su codificación se siguen las siguientes reglas:

Cada categoría o palabra clave, en el ejemplo, PASADO y PRESENTE, han de comenzar en una tarjeta empezando a codificarse en la primera columna el nombre de la categoría, a continuación separadas por blancos las palabras que contiene esa categoría, dejando 4 blancos entre la palabra clave y la primera palabra de la categoría que se codifica, si se necesitasen más de una tarjeta para codificar una categoría se continuará en tarjetas sucesivas dejando un blanco en la primera columna.

EJEMPLO: PASADO AYER FUI ERA
 PRESENTE HOY SOY AHORA VAMOS

TARJETAS DEL SISTEMA NECESARIAS PARA UTILIZAR EL TAG

```

// JOB
// EXEC COTAG
//COCOA.CONTROL DD *
tarjetas de control de programa COCOA
/*
//COCOA.TEXT DD *
tarjetas con el texto a analizar
//TAG.DICT DD *
Diccionario del TAG
/*
//

```

Este paquete se encuentra situado en el disco OSIRIS, información que habrá que especificar en la hoja de presentación del programa, tanto el COCOA como el COTAG necesitan 210 K de memoria para su ejecución.

EJEMPLO:

```
//      JOB
// EXEC COTAG
//COCOA. CONTROL DD *
72
0 <., : ; ¿? $ " >
1 <' - >
2*
3 <1ABCEDEFGHIJKLMNOPQRSTUVWXYZ >
4*
5*
6*
7*
W 1 24
P <T4> (<TEXT >) <ZZZZ > 100000
F <1,3> (A Z/1 1000)
/*
//COCOA.TEXT DD *
<TEXT >
YO ME LLAMO ELVIRA .....
.....
/*
//TAG. DICT DD *
PASADO AYER HICE FUI .....
ERAMOS .....
PRESENTE SOY SOMOS AHORA .....
.....
/*
//
```

BIBLIOTECA DE PROGRAMAS A.P.L.Descripción de los workspaces 1 UTIL, 10 PROBIT y 100 CIRCUITOS

Recientemente han sido añadidos a la biblioteca de programas APL, tres workspaces de los que a continuación damos una breve descripción.

Dichos workspaces son contribución de usuarios del sistema APL en nuestra instalación y han sido incluidos en bibliotecas públicas para ponerlos a disposición de todos nuestros usuarios APL.

WORKSPACE: 1 UTIL

Este espacio de trabajo, ha sido preparado para dar cabida a funciones de utilidad y contiene actualmente las funciones siguientes:

LISLEN 'nombre de función'

produce un listado "lento" de la función cuyo nombre aparece en el argumento. Puede ser de mucha utilidad cuando el terminal está desajustado y produce errores en los listados standard de funciones, sobre todo cuando éstas contienen numerosos caracteres compuestos.

NL NC EDILEN 'nombre de función'

Simula el procedimiento de edición [NL □ NC] con las mismas características de listado "lento".

En ambos casos, puede encontrarse una descripción más detallada en el propio workspace, bajo los nombres LISLEN△INFO y EDILEN△INFO respectivamente.

WORKSPACE: 10 PROBIT

En la biblioteca pública n.10 (estadística), se ha incluido el workspace PROBIT, que permite realizar el análisis probit que se describe en el artículo: "Un método de análisis para variables tipo quantal (Probit)", publicado en el Boletín del CCUC, n.36 (Junio 1980).

Aparte de la función principal PROBIT, contiene algunas funciones auxiliares cuya descripción, debida a su autor Juan Pablo López de la Fuente, incluimos a continuación:



1. FUNCION ADNA

Cálcula el valor de la función de distribución de una normal $N(0,1)$ para cualquier argumento que se le dé, se ha mejorado su precisión respecto a la versión que se utilizó para el artículo del Boletín, el error de esta función es ahora de $+0.0000007$.

2. FUNCION CUAD

Presenta un cuadro, con % MORTANDAD, PROBIT, LOG DOSIS y valor de las dosis. Esta función está formateada de modo que el resultado final es siempre un cuadro. Puede cambiarse el número de decimales en los formatos si el usuario desea otra precisión.

3. FUNCION DNA1

Calcula la función de densidad para una normal $N(0,1)$.

4. FUNCIONES ERROR, ERRORES

Las funciones ERROR y ERRORES calculan los errores para la pendiente, media, logaritmos y dosis aplicadas.

5. FUNCION EVAL

Tiene la función que estamos maximizando y lo único que hace es calcular el valor de dicha función.

6. FUNCIONES PLOT y VS

Dibujan la gráfica de una curva. Utilizan las funciones auxiliares:

ARGUMENT, PEPT, PRT, PLOT TRANS y WIDTH y todas las variables globales del ws.

7. FUNCION INTG

Calcula la función de densidad, función de distribución para una CHI^2 . Tiene las siguientes funciones auxiliares: ITG, FUNC, CORR, DD esta última es una función auxiliar de la CORR.

8. FUNCION INVD

Calcula, tomando valores de la probabilidad, los valores correspondientes en los ejes cartesianos, que corresponden a los valores del PROBIT. Es en definitiva la inversa de la $N(0,1)$. Usa como función auxiliar la CDCAL.

9. FUNCION NEWTON

Calcula el gradiente, hessiano y el nuevo punto, para el cual la función se va a evaluar, comparandola posteriormente con el valor que tenía anteriormente.

10. FUNCION NLINREG

Esta función calcula los coeficientes de una línea de regresión como aproximación para el cálculo de $ALFA = XINI_1$ y $BETA = XINI_2$

WORKSPACE: 100 CIRCUITOS

Este Ws ha sido preparado en este Centro por Raúl García Pérez (ETSI Telecomunicación, Madrid) y su interés viene respaldado por haber obtenido el Segundo Premio del Concurso Nacional "MINI-MICRO" del SIMO 79, organizado por la Fundación CITEMA.

Se trata de unos programas, en el campo de la inteligencia artificial, que generan circuitos cerrados de cualquier marca de maquetas automovilísticas o ferroviarias.

Quien esté interesado en una descripción completa la encontrará en el WS "CIRCUITOS", incluido en la biblioteca 100 de la instalación APL de este Centro. Si bien diremos aquí que opera sobre una cantidad predeterminada de pistas de cada tipo de los existentes en la marca, y que a partir de ellas genera los posibles circuitos que contengan un determinado tramo, prefijado también por el usuario.

Los circuitos así obtenidos pueden almacenarse en un índice, así como ser dibujados en cualquier momento o durante su proceso de creación. Para hacerlo se pueden utilizar el elemento de impresión normal o la bola 114 (PLOTTER) indistintamente.

Ejemplo de funcionamiento:

STROMBECKER

INDICE COMPLETO

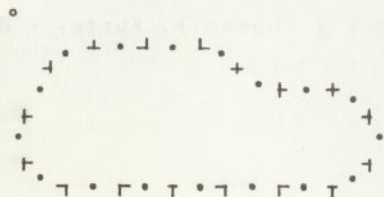
□ INDICE DE LA MARCA: STROMBECKER.

- 1 RCCCCRCCRRCCRC
- 2 RRRCRCRCRRRRRCCRCRCRCCCRCCCCRCRC
- 3 RCCCRCCRCRCCRRRCRCRCCCRCCRCRRCCRRRC
- 4 RRRRCCRRRRRCCRCR
- 5 RRRRRCCCRCCRRRC

PINTA 5

□ INDICE (CON DIBUJOS) DE LA MARCA: STROMBECKER.

5 RRRRRCCCRCCRRRC



□