

Apéndice: códigos

Nota. Las secciones precedidas por la letra *A* contendrán los códigos de los programas de MATLAB. Por el contrario, las secciones precedidas por la letra *B* introducirán los códigos de los programas de RStudio.

A.1 Programas ejecutados durante la competición

Función auxiliar *calculoapuestas*

```
function [tabla]=calculoapuestas(probabilidadactiva,
probabilidadcomp, jugadoract, jugadorcomp, partidoact, resultado)
%Los valores de entrada que necesita son:
%probabilidadactiva: vector de longitud 6 con las 6 probabilidades
%de el jugador activo o jugador 1

%probabilidadcomp: vector de longitud 6 con las 6 probabilidades
%de el %jugador competidor o jugador 2

%jugadoract: nombre del jugador activo o jugador 1 introducido
%entre comillas simples: 'nombre'

%jugadorcomp: nombre del jugador competidor o jugador 2 introducido
%entre comillas simples: 'nombre'

%partidoact: nombre del partido en el que realizaron la prediccion
%tanto jugador 1 como jugador 2 tambien entre comillas
%simples: 'Equipo1-Equipo2'

%resultado: un numero que puede ser 1 en caso de victoria local,
%2 en caso de empate y 3 en caso de derrota local
if resultado=="0"
    table=[];
    return
else
A=vertcat(probabilidadactiva,probabilidadcomp);
%Lo primero que hago es crear una matriz de dos filas con los dos
%vectores de probabilidades
tablag=string(zeros(1,7));
%tablag es un vector fila que contendra los datos solicitados
```

```

%en caso de que los intervalos de victoria local de
%jugador 1 y jugador 2 sean disjuntos.
%En caso de que no sean disjuntos, se quedaria como una
%fila de 0's y lo eliminare al final de la funcion
tablae=string(zeros(1,7));
%tablae es un vector fila que contendra los datos solicitados
%en caso de que los intervalos de empate de jugador 1 y jugador 2
%sean disjuntos.
%En caso de que no sean disjuntos, se quedaria como una
%fila de 0's y lo eliminare al final de la funcion
tablap=string(zeros(1,7));
%tablap es un vector fila que contendra los datos solicitados
%en caso de que los intervalos de derrota local de
%jugador 1 y jugador 2 sean disjuntos.
%En caso de que no sean disjuntos, se quedaria como una
%fila de 0's y lo eliminare al final de la funcion
if A(1,1)>A(2,2)
%Voy comprobando si el limite inferior del intervalo
%de victoria local de jugador 1 es mayor que el
%limite superior del intervalo de victoria
%local de jugador 2
    fprintf('El jugador %s y el jugador %s apostaran entre si
    en la victoria del equipo local en el
    partido %s \n',jugadoract,jugadorcomp,partidoact)
    tablag(1)=convertCharsToStrings(jugadoract);
%El primer valor de tablag sera el Username del jugador 1
    tablag(2)=convertCharsToStrings(jugadorcomp);
%El segundo valor de tablag sera el Username del jugador 2
    tablag(3)="Victorialocal";
%El tercer valor de tablag sera el resultado por el que apuestan
    if resultado=="1"
        tablag(4)=convertCharsToStrings(jugadoract);
%El cuarto valor de tablag sera el ganador de la apuesta
%(en este caso jugador activo o jugador 1 pues asigno mayor
%probabilidad de victoria y resultado 1 equivale a victoria local)
        tablag(5)=(A(1,1)-A(2,2))*((A(1,1)+A(2,2))/2);
%El quinto valor de tablag seran los puntos de
%jugador 1 tras la apuesta
        tablag(6)=- (A(1,1)-A(2,2))*((A(1,1)+A(2,2))/2);
%El sexto valor de tablag seran los puntos de
%jugador 2 tras la apuesta
    else
        tablag(4)=convertCharsToStrings(jugadorcomp);
%En caso de que el resultado de partido no sea 1, entonces estamos
%en el caso de empate o derrota local y ganaria jugador 2
        tablag(5)=- (A(1,1)-A(2,2))*((A(1,1)+A(2,2))/2);
        tablag(6)=(A(1,1)-A(2,2))*((A(1,1)+A(2,2))/2);
    end
    tablag(7)=convertCharsToStrings(partidoact);
%El septimo valor de tablag sera el nombre del partido
%en el que se aposto

elseif A(1,2)<A(2,1)
%Voy comprobando si el limite inferior del intervalo de victoria

```

```

%local de jugador 2 es mayor que el limite superior del intervalo
%de victoria local de jugador 1
    fprintf('El jugador %s y el jugador %s apostaran entre si
en la victoria del equipo local en el
partido %s \n', jugadoract, jugadorcomp, partidoact)
    tablag(1)=convertCharsToStrings(jugadoract);
    tablag(2)=convertCharsToStrings(jugadorcomp);
    tablag(3)="Victorialocal";
    if resultado=="1"
        tablag(4)=convertCharsToStrings(jugadorcomp);
%Ganaria en este caso jugador 2 pues asigno mayor
%probabilidad de victoria local
        tablag(5)=- (A(2,1)-A(1,2)) * ((A(2,1)+A(1,2))/2);
        tablag(6)=(A(2,1)-A(1,2)) * ((A(2,1)+A(1,2))/2);
    else
        tablag(4)=convertCharsToStrings(jugadoract);
%Ganaria jugador 1 pues el resultado fue empate o derrota local
%y jugador 2 pues asigno mayor probabilidad de victoria local
        tablag(5)=(A(2,1)-A(1,2)) * ((A(2,1)+A(1,2))/2);
        tablag(6)=- (A(2,1)-A(1,2)) * ((A(2,1)+A(1,2))/2);
    end
    tablag(7)=convertCharsToStrings(partidoact);

end

    if A(1,3)>A(2,4)
%Voy comprobando si el limite inferior del intervalo de empate
%de jugador 1 es mayor que el limite superior del
%intervalo de empate de jugador 2
        fprintf('El jugador %s y el jugador %s apostaran entre
si por el empate en el
partido %s \n', jugadoract, jugadorcomp, partidoact)
        tablae(1)=convertCharsToStrings(jugadoract);
        tablae(2)=convertCharsToStrings(jugadorcomp);
        tablae(3)="Empate";
        if resultado=="2"
            tablae(4)=convertCharsToStrings(jugadoract);
%Gana jugador 1 pues el resultado del partido fue 2 (empate)
%y jugador 1 asigno mayor probabilidad a empate que jugador 2
            tablae(5)=(A(1,3)-A(2,4)) * ((A(1,3)+A(2,4))/2);
            tablae(6)=- (A(1,3)-A(2,4)) * ((A(1,3)+A(2,4))/2);
        else
            tablae(4)=convertCharsToStrings(jugadorcomp);
%Gana jugador 2 pues salio victoria local o derrota local y
%jugador 1 habia asignado mayor probabilidad a empate
            tablae(5)=- (A(1,3)-A(2,4)) * ((A(1,3)+A(2,4))/2);
            tablae(6)=(A(1,3)-A(2,4)) * ((A(1,3)+A(2,4))/2);
        end
        tablae(7)=convertCharsToStrings(partidoact);
    elseif A(1,4)<A(2,3)
%Voy comprobando si el limite inferior del intervalo de
%empate de jugador 2 es mayor que el limite superior
%del intervalo de empate de jugador 1
        fprintf('El jugador %s y el jugador %s apostaran

```

```

entre si por el empate en el
partido %s \n', jugadoract, jugadorcomp, partidoact)
    tablae(1)=convertCharsToStrings(jugadoract);
    tablae(2)=convertCharsToStrings(jugadorcomp);
    tablae(3)="Empate";
    if resultado=="2"
        tablae(4)=convertCharsToStrings(jugadorcomp);
%Gana jugador 2 pues salio empate y jugador 2 era el que
%habia asignado mayor probabilidad a empate
        tablae(5)=- (A(2,3)-A(1,4)) * ((A(2,3)+A(1,4))/2);
        tablae(6)= (A(2,3)-A(1,4)) * ((A(2,3)+A(1,4))/2);
    else
        tablae(4)=convertCharsToStrings(jugadoract);
%Gana jugador 1 pues salio victoria local o derrota local
%y jugador 2 era el que mas probabilidad habia
%asignado para empate
        tablae(5)= (A(2,3)-A(1,4)) * ((A(2,3)+A(1,4))/2);
        tablae(6)= - (A(2,3)-A(1,4)) * ((A(2,3)+A(1,4))/2);
    end
    tablae(7)=convertCharsToStrings(partidoact);
end
if A(1,5)>A(2,6)
%Voy comprobando si el limite inferior del intervalo de
%derrota local de jugador 1 es mayor que el limite superior
%del intervalo de derrota local de jugador 2
    fprintf('El jugador %s y el jugador %s apostaran entre
si en la derrota del equipo local en el
partido %s\n', jugadoract, jugadorcomp, partidoact)
    tablap(1)=convertCharsToStrings(jugadoract);
    tablap(2)=convertCharsToStrings(jugadorcomp);
    tablap(3)="Derrotalocal";
    if resultado=="3"
        tablap(4)=convertCharsToStrings(jugadoract);
%Gana jugador 1 pues salio derrota local y jugador 1 fue
%el que mas probabilidad asigno a derrota local
        tablap(5)= (A(1,5)-A(2,6)) * ((A(1,5)+A(2,6))/2);
        tablap(6)= - (A(1,5)-A(2,6)) * ((A(1,5)+A(2,6))/2);
    else
        tablap(4)=convertCharsToStrings(jugadorcomp);
%Gana jugador 2 pues salio victoria local o empate y
%jugador 1 fue el que mas probabilidad asigno a derrota local
        tablap(5)= - (A(1,5)-A(2,6)) * ((A(1,5)+A(2,6))/2);
        tablap(6)= (A(1,5)-A(2,6)) * ((A(1,5)+A(2,6))/2);
    end
    tablap(7)=convertCharsToStrings(partidoact);
elseif A(1,6)<A(2,5)
%Voy comprobando si el limite inferior del intervalo de
%derrota local de jugador 2 es mayor que el limite superior
%del intervalo de derrota local de jugador 1
    fprintf('El jugador %s y el jugador %s apostaran entre
si en la derrota del equipo local en el
partido %s\n', jugadoract, jugadorcomp, partidoact)
    tablap(1)=convertCharsToStrings(jugadoract);
    tablap(2)=convertCharsToStrings(jugadorcomp);

```

```

        tablap(3)="Derrotalocal";
        if resultado=="3"
            tablap(4)=convertCharsToStrings(jugadorcomp);
            %Gana jugador 2 pues salio derrota local y jugador 2 fue
            %el que mas probabilidad asigno a derrota local
            tablap(5)=-(A(2,5)-A(1,6))*((A(2,5)+A(1,6))/2);
            tablap(6)=(A(2,5)-A(1,6))*((A(2,5)+A(1,6))/2);
        else
            tablap(4)=convertCharsToStrings(jugadoract);
            %Gana jugador 1 pues salio victoria local o empate y jugador 2
            %fue el que mas probabilidad asigno a derrota local
            tablap(5)=(A(2,5)-A(1,6))*((A(2,5)+A(1,6))/2);
            tablap(6)=- (A(2,5)-A(1,6))*((A(2,5)+A(1,6))/2);
        end
        tablap(7)=convertCharsToStrings(partidoact);
    end
    if tablag(3)=="0"
        %Si el valor correspondiente a "Victorialocal" es "0",
        %significa que la matriz no pudo entrar en el primer if
        %(linea 28 hasta 60) y por tanto no eran disjuntos los
        %intervalos de prob de victoria local. Por ello, elimino
        %tablag que es una fila de "0"s
        tablag=[];
    end
    if tablae(3)=="0"
        %Si el valor correspondiente a "Empate" es "0", significa
        %que la matriz no pudo entrar en el segundo if
        %(linea 62 hasta 92) y por tanto no eran disjuntos
        %los intervalos de prob de empate. Es por ello que elimino
        %tablae, pues es una fila de "0"s
        tablae=[];
    end
    if tablap(3)=="0"
        %Si el valor correspondiente a "Derrotalocal" es "0", significa
        %que la matriz no pudo entrar en el tercer if
        %(linea 93 hasta 124) y por tanto no eran disjuntos
        %los intervalos de prob de derrota. Es por ello que elimino
        %tablap, pues es una fila de "0"s
        tablap=[];
    end
    tabla=[tablag;tablae;tablap];
end
end

```

Programa de actualización diaria de clasificación *MotorDefinitivo*

```

clear all

%Primero asignamos y modificamos B
B=readtable('MotorWCC.xlsm','sheet','Partidos',
    'Range','A:A'); %Lee todos los partidos del

```

```

%mundial en formato tabla
B=table2array(B); %Lo pasamos a tipo cell
B=convertCharsToStrings(B); %Lo convertimos en un
%vector de strings
%Despues asignamos y modificamos D
D=readtable('MotorWCC.xlsm','sheet','Partidos',
'Range','B:B'); %Lee los resultados de los partidos
%del mundial en formato tabla
D=table2array(D); %Lo convertimos en un %vector tipo numerico

%Asignamos y modificamos F
F=readtable('MotorWCC.xlsm','sheet','Partidos',
'Range','C:C'); %Lee si los partidos seleccionados
%ya se escogieron en una actualizacion
%en formato tabla
F=table2array(F); %Lo convertimos en un vector tipo numerico

%Asignamos y modificamos G
G=readtable('MotorWCC.xlsm','sheet','Partidos',
'Range','D:D'); %Lee los resultados de los partidos
%del mundial en formato tabla
G=table2array(G);

%Construimos y modificamos E
E=[B,D,F,G]; %Uno cada partido con el resultado
%asociado a ese partido y la columna que seria
%de D pasa a ser tipo string al unificarlo
%con B (por ser B string)
E(find(E(:,2)=="0" | E(:,3)=="1"),:)=[]; %Eliminamos
%aquellos partidos que no se hayan disputado
%aun (los que tienen resultado 0)
npartidos=size(E,1); %Guardamos en una variable
%el numero de partidos disputados

%Construimos y modificamos C
C=readtable('MotorWCC.xlsm','sheet','Table1','Range','H:H'); %Lee
%todos los usuarios registrados en la web en formato tabla
C=table2array(C); %Lo convertimos en un vector tipo cell
C(1)=[]; %Eliminamos el primer elemento de C,
%pues era el encabezado
C=convertCharsToStrings(C); %Lo convertimos en un vector
%de tipo string
C(end)=[]; %Eliminamos a un usuario que se registro dos veces
nombres=size(C,1); %Guardamos en una variable el
%numero de usuarios registrados

%Construimos y modificamos X
X1=readtable('MotorWCC.xlsm','sheet','Probabilidad1',
'Range','C:C');
%Lee la hora en la que se envio cada respuesta en formato tabla
leerX=strcat("M1:M",num2str(height(X1))); %Creamos un string con
%los rangos en los que escribira y leera en Excel (primero se
%escribe y luego se lee para obtener
%el formato deseado de Datetime una vez pasado a matriz)

```

```

xlswrite('MotorWCC.xlsm',table2array(X1),
    'Probabilidad',leerX); %Le decimos que escriba
%la tabla en el rango predefinido
X=readtable('MotorWCC.xlsm','sheet','Probabilidad',
    'Range',leerX); %Y ahora que lo lea, esta vez
% con el formato que nos interesa
X=table2array(X); %Lo transformamos a matriz y
%obtenemos el tipo Datatime
X(find((isnat(X))==1))=[]; %Eliminamos aquellos elementos (si es
%que los hay) que no sean una fecha y hora

%Construimos la matriz A
leerA=strcat("P1:", "U", num2str(length(X)+1)); %Creamos un
%string con los rangos en los que leera en Excel
%la matriz con las probabilidades
A=table2array(readtable('MotorWCC.xlsm','sheet','Probabilidad',
    'Range',leerA));
%Leemos la tabla con las probabilidades y la
%transformamos directamente en matriz numerica
nencuestas=size(A,1); %Asignamos una variable para el numero total
%de encuestas

%Construimos la matriz a
leera=strcat("O1:V", num2str(length(X)+1)); %Creamos un
%string con los rangos en los que leera en Excel los
%usuarios y los partidos en los que apostaron
%esos usuarios
a=readtable('MotorWCC.xlsm','sheet','Probabilidad','Range',leera);
%Leemos una tabla mayor en formato tabla, que
%filtraremos a continuacion:
a(:,2:7)=[]; %Pulimos a para quedarnos solo con
%los partidos y los usuarios que participaron en
%las encuestas
a=table2array(a); %La transformamos en una matriz de caracteres

%Eliminamos las encuestas de los usuarios no registrados o los que
%introdujeron mal los datos:

usunoreg=find(a(:,2)=="SINUSUARIO"); %Buscamos la posicion
%de los usuarios no registrados
a(usunoreg,:)=[]; %Eliminamos al usuario y los
%partidos en los que aposto
A(usunoreg,:)=[]; %Eliminamos las probabilidades
%que habia asignado
X(usunoreg)=[]; %Eliminamos la hora de entrega
%de sus encuestas

%Eliminar las apuestas de partidos que aun no se jugaron:

table=[]; %Inicializamos una tabla vacia
areult=[]; %Inicializamos una tabla que contendra
%aquellas filas de a cuyos partidos ya
%tengan resultado

```

```

Aresult=[]; %Inicializamos una tabla que contendra
%aquellas filas de A cuyos partidos ya
%tengan resultado
Xresult=[]; %Inicializamos una tabla que contendra
%aquellos elementos de X cuyos partidos ya
%tengan resultado
for i=1:npartidos
    U=find(a(:,1)==E(i,1)); %U sera un vector
    %con los indices donde a tenga un partido
    %con resultado
    aresult=[aresult;a(U,:)]; %Completamos aresult
    %con los usuarios y partidos con resultado
    %que apostaron esos usuarios
    Aresult=[Aresult;A(U,:)]; %Completamos Aresult
    %con los vectores de probabilidad que cada
    %usuario asigna a un partido con resultado
    Xresult=[Xresult;X(U,:)]; %Completamos Xresult
    %con la fecha y hora de entrega de una apuesta
    %para un partido con resultado
    U=[];
end

%Eliminar las respuestas de un participante para
%un mismo partido y quedarse con la ultima:

j=1; %Inicializamos un contador
coinwhile=[]; %Inicializamos un vector que nos
%sera util en el while siguiente
aulti=[]; %Inicializamos una tabla que contendra
%aquellas filas de aresult sin repetir participante
%y partido. Es decir, se quedaria con la ultima
%entrega de un participante para ese partido
Aulti=[]; %Inicializamos una tabla que contendra
%aquellas filas de Aresult sin repetir participante
%y partido. Es decir, se quedaria con el vector de
%probabilidades de la ultima entrega de un participante
%para ese partido
Xulti=[]; %Inicializamos un vector que contendra
%aquellos valores de Xresult sin repetir participante
%y partido. Es decir, se quedaria con la ultima
%fecha de entrega de un participante para ese
%partido

while (j<=size(aresult,1)) %Mientras el contador sea
%menor que el numero de enuestas de
%partidos CON resultado
    if (isempty(find(coinwhile==j))==1) %Si el numero
    %del contador NO esta en coinwhile

        nombre=convertCharsToStrings(char(aresult(j,2))); %Asignamos
        %nombre del participante/usuario
        partido=convertCharsToStrings(char(aresult(j,1))); %Asignamos
        %partido por el que apuesta

```



```

L=max(Xresult(find(areult(:,2)==nombre &
    areult(:,1)==partido)));
%Miramos si hay alguna encuesta en donde
%coincidan partido y usuario (al menos hay uno siempre)

M=find(Xresult==L); %Buscamos el que tenga la
%fecha mas reciente y nos quedamos con ese

if length(M)==1

aulti=[aulti;areult(M,:)]; %Rellenamos la tabla
%de participantes y partidos y ya no repite
%participante y partido
Aulti=[Aulti;Aresult(M,:)]; %Rellenamos la tabla
%de probabilidades y en esta tabla estara
%la ultima encuesta enviada para cada
%participante y partido
Xulti=[Xulti;Xresult(M,:)]; %Rellenamos la tabla de fechas
%de entrega de encuestas sin repetir participante y partido

coinwhile=[coinwhile,
    (find(areult(:,2)==nombre &
    areult(:,1)==partido))'];
%Actualizamos el vector con las posiciones
%donde ya hemos mirado que coinciden participante
%y partido

else
    horavariosnombres=areult(M,2)==nombre;
    Mayuda=find(horavariosnombres==1);
    aulti=[aulti;areult(M(Mayuda),:)] ; %Rellenamos la tabla de
    %participantes y partido y ya no repite
    %participante y partido
    Aulti=[Aulti;Aresult(M(Mayuda),:)] ; %Rellenamos la tabla de
    %probabilidades y en esta tabla estara
    %la ultima encuesta enviada para
    %cada participante y partido
    Xulti=[Xulti;Xresult(M(Mayuda),:)] ; %Rellenamos
    %la tabla de fechas de entrega de encuestas
    %sin repetir participante y partido

    coinwhile=[coinwhile,
        (find(areult(:,2)==nombre &
        areult(:,1)==partido))'];
    %Actualizamos el vector con las posiciones
    %donde ya hemos mirado que
    %coinciden participante y partido
end
end
j=j+1;%Actualizamos contador
end

```

```

%Realizamos la competicion entre los participantes:
tpartidos=E(:,1);
npartidostot=length(tpartidos); %Guardamos en una
%variable el numero de partidos disputados
minjorn=min(str2double(E(:,4)));
maxjorn=max(str2double(E(:,4)));
leerpuntuacion=strcat("H1:", "H", num2str(nnombres+1)); %Creamos
%un string con los rangos en los que leera en Excel
%la puntuacion acumulada
puntuacion=table2array(readtable('MotorWCC.xlsm', 'sheet',
'Clasificacion', 'Range', leerpuntuacion));

for i=minjorn:maxjorn
    E(find(E(:,4)~=num2str(i)),:)=[]; %Eliminamos
    %aquellos partidos que no sean de la misma
    %jornada que el parametro que recorre el for
    %Eliminar las apuestas de partidos que aun no se jugaron:
    table=[]; %Inicializamos una tabla vacia
    apart=[]; %Inicializamos una tabla que contendra aquellas
    %filas de a cuyos partidos ya tengan resultado
    Apart=[]; %Inicializamos una tabla que contendra aquellas
    %filas de A cuyos partidos ya tengan resultado
    npartidos=length(E(:,1));
    for h=1:npartidos
        U=find(aulti(:,1)==E(h,1)); %U sera un vector con los
        %indices donde a tenga un partido con
        %resultado
        apart=[apart;aulti(U,:)]; %Completamos aresult con
        %los usuarios y partidos con resultado
        % que apostaron esos usuarios
        Apart=[Apart;Aulti(U,:)]; %Completamos Aresult con
        %los vectores de probabilidad que cada usuario
        %asigna a un partido con resultado
        U=[];
    end

    tnombres=apart(:,2); %Nos quedamos con la lista
    %definitiva de usuarios participantes en
    %las encuestas de los partidos
    for k=1:nnombres
        jugadoract=C(k); %Fijamos el primer jugador en la
        %lista de registro
        U=find(tnombres==jugadoract); %Lo buscamos en
        %la lista de usuarios participantes en las encuestas
        if length(U)>0
            for h=1:length(U)
                partidoact=convertCharsToStrings(char(apart(U(h),1)));
                %Vamos asignando como partido activo
                %los partidos en los que apuesta el
                %jugador activo
                probabilidadactiva=Apart(U(h),:); %Asignamos
                %como probabilidad activa la fila
                %correspondiente a las probabilidades
            end
        end
    end
end

```

```

%del partido activo
V=find( apart(:,1)==partidoact); %Buscamos en
%el resto de apuestas las asignaciones
%para partido activo
    if length(V)>0
        for p=1:length(V)
            if (jugadoract~=tnombres(V(p)) &&
                k<find(C==char(tnombres(V(p))))))
%Comprobamos que jugador activo no apueste
%contra si mismo y que una vez hayan apostado
%jugador A contra jugador B, no sea necesario
%volvera comprobar si jugador B apuesta contra jugador A
                jugadorcomp=char(tnombres(V(p)));
%Asignamos al jugador competidor, que apostara
%contra jugador activo
                probabilidadcomp=Apart(V(p),:);
%Le asignamos su probabilidad para el partido activo
                table=[table;
                    calculoapuestas(probabilidadactiva,
                    probabilidadcomp,
                    jugadoract,jugadorcomp,
                    partidoact,E(E==partidoact,2))];
%Rellenamos la tabla con ayuda de la funcion calculoapuestas
            end
        end
    end
end
end
end

if size(table,1)==0

    else
        for w=1:length(C)
            Q=find(table(:,1)==C(w));
%Fijado un jugador de la lista de registro,
%miramos cuantas veces y en que posiciones
%aparece como jugador 1 en la tabla
            if length(Q)>0
                if i<=13
                    puntuacion(w)=puntuacion(w)+
                        sum(str2double(table(Q,5)));
%Si aparece alguna vez como jugador 1, le sumamos
%la cantidad que tiene asignada tras conocer
%el resultado del partido
                    elseif i>13 && i<=22
                        puntuacion(w)=puntuacion(w)+
                            2*sum(str2double(table(Q,5)));
                    else
                        puntuacion(w)=puntuacion(w)+
                            3*sum(str2double(table(Q,5)));
                    end
                end
            end
        end
        R=find(table(:,2)==C(w));

```

```

%De nuevo, y con el mismo jugador fijado, miramos
%cuantas veces y en que posiciones aparece
%como jugador 2 en la tabla
    if length(R)>0
        if i<=13
            puntuacion(w)=puntuacion(w)+
                sum(str2double(table(R,6)));
%Si aparece alguna vez como jugador 2, le sumamos
%la cantidad que tiene asignada tras conocer
%el resultado del partido
            elseif i>13 && i<=22
                puntuacion(w)=puntuacion(w)+
                    2*sum(str2double(table(R,6)));
            else
                puntuacion(w)=puntuacion(w)+
                    3*sum(str2double(table(R,6)));
            end
        end
    end
end

%Finalmente calculamos y exportamos la clasificacion:
if i==maxjorn
ordenpart=[];
puntuacion=round(puntuacion.*10000000)./10000000;
y=sort(puntuacion,'descend');
y=round(y.*10000000)./10000000;
contador=1;
    while contador<=nombres
        k=find(puntuacion==y(contador));
        if length(k)==1
            ordenpart=[ordenpart;C(k)];
            contador=contador+1;
        else
            ordenpart=[ordenpart;C(k)];
            contador=contador+length(k);
        end
    end
end
E=[B,D,F,G];
end
clasificacion=[ordenpart,y]

ultimoparteval=sum(F==1);
escribirparteval=strcat("C",num2str(ultimoparteval+2),":C",
num2str(ultimoparteval+1+npartidos));
xlswrite('MotorWCC.xlsm',ones(npartidostot,1),
'Partidos',escribirparteval);

tablavieja=readtable('MotorWCC.xlsm','sheet','Hoja2',
'Range','C:C');
tablavieja=table2array(tablavieja);
escribirtable=strcat("B",num2str(length(tablavieja)+4));

```

```

xlswrite('MotorWCC.xlsm',puntuacion,'Clasificacion','H2');
xlswrite('MotorWCC.xlsm',table,'Hoja2',escribirtable);
xlswrite('MotorWCC.xlsm',ordenpart,'Clasificacion','B3');
xlswrite('MotorWCC.xlsm',y,'Clasificacion','C3');

```

A.2 Programas para el análisis de las asignaciones de los participantes

```

clear all
B=readtable('MotorWCC_RStudio.xlsm','sheet','Partidos',
    'Range','A:A');
B=table2array(B);
B=convertCharsToStrings(B);
D=readtable('MotorWCC_RStudio.xlsm','sheet','Partidos',
    'Range','B:B');
D=table2array(D);
F=readtable('MotorWCC_RStudio.xlsm','sheet','Partidos',
    'Range','C:C');
F=table2array(F);
C=readtable('MotorWCC_RStudio.xlsm','sheet','Table1',
    'Range','H:H');
C=table2array(C);
C(1)=[];
C=convertCharsToStrings(C);
C(end)=[];
nombres=size(C,1);
X1=readtable('MotorWCC_RStudio.xlsm','sheet','Probabilidad1',
    'Range','C:C');
leerX=strcat("M1:M",num2str(height(X1)));
xlswrite('MotorWCC_RStudio.xlsm',table2array(X1),
    'Probabilidad',leerX);
X=readtable('MotorWCC_RStudio.xlsm','sheet','Probabilidad',
    'Range',leerX);
X=table2array(X);
X(find((isnat(X))==1))=[];
leerA=strcat("P1:", "U", num2str(length(X)+1));
A=readtable('MotorWCC_RStudio.xlsm','sheet','Probabilidad',
    'Range',leerA);
A=table2array(A);
nencuestas=size(A,1);
leera=strcat("O1:V", num2str(length(X)+1));
a=readtable('MotorWCC_RStudio.xlsm','sheet','Probabilidad',
    'Range',leera);
a(:,2:7)=[];
a=table2array(a);
usunoreg=find(a(:,2)=="SINUSUARIO");
a(usunoreg,:)=[];
A(usunoreg,:)=[];
X(usunoreg)=[];
j=1;
coinwhile=[];

```

```

aulti=[];
Aulti=[];
Xulti=[];
while (j<=size(a,1))
    if(isempty(find(coinwhile==j))==1)
        nombre=convertCharsToStrings(char(a(j,2)));
        partido=convertCharsToStrings(char(a(j,1)));

        L=max(X(find(a(:,2)==nombre & a(:,1)==partido)));
        M=find(X==L);
        if length(M)==1
            aulti=[aulti;a(M,:)];
            Aulti=[Aulti;A(M,:)];
            Xulti=[Xulti;X(M,:)];
            coinwhile=[coinwhile,
                (find(a(:,2)==nombre &
                    a(:,1)==partido))'];
        else
            horavariosnombres=a(M,2)==nombre;
            Mayuda=find(horavariosnombres==1);
            aulti=[aulti;a(M(Mayuda),:)]];
            Aulti=[Aulti;A(M(Mayuda),:)]];
            Xulti=[Xulti;X(M(Mayuda),:)]];
            coinwhile=[coinwhile,
                (find(a(:,2)==nombre &
                    a(:,1)==partido))'];
        end
    end
    j=j+1;
end
E=[B,D,F];
puntuacion=zeros(50,1);
puestospart=zeros(50,23);
maxpuntuacion1=[];
maxpuntuacion2=[];
maxpuntuacion3=[];
for i=1:23 %Hubo 23 dias de competicion
    E(find(E(:,3)~=num2str(i)),:)=[];
    npartidos=size(E,1); %Guardamos en una variable
    %disputados el numero de partidos
    table=[];
    aresult=[];
    Aresult=[];
    for h=1:npartidos
        U=find(aulti(:,1)==E(h,1));
        aresult=[aresult;aulti(U,:)];
        Aresult=[Aresult;Aulti(U,:)];
        U=[];
    end
    tnombres=areult(:,2);
    for k=1:nnombres
        jugadoract=C(k);
        U=find(tnombres==jugadoract);
        if length(U)>0

```

```

for h=1:length(U)
partidoact=convertCharsToStrings(
char(aresult(U(h),1)));
probabilidadactiva=Aresult(U(h),:);
V=find(aresult(:,1)==partidoact);
    if length(V)>0
        for p=1:length(V)
            if (jugadoract~=tnombres(V(p)) &&
                k<find(C==char(tnombres(V(p))))))
                jugadorcomp=char(tnombres(V(p)));
                probabilidadcomp=Aresult(V(p),:);
                table=[table;
                    calculoapuestas(probabilidadactiva,
                    probabilidadcomp, jugadoract,
                    jugadorcomp,partidoact,
                    E(E==partidoact,2))];
            end
        end
    end
end
end
end
end
if size(table,1)==0

else
for w=1:length(C)
Q=find(table(:,1)==C(w));
if length(Q)>0
    if i<=13
        puntuacion(w)=puntuacion(w)+
            sum(str2double(table(Q,5)));
        maxpuntuacion1=[maxpuntuacion1;
            C(w),sum(str2double(table(Q,5))),i];
    elseif i>13 && i<=22
        puntuacion(w)=puntuacion(w)+
            2*sum(str2double(table(Q,5)));
        maxpuntuacion2=[maxpuntuacion2;
            C(w),2*sum(str2double(table(Q,5))),i];
    else
        puntuacion(w)=puntuacion(w)+
            3*sum(str2double(table(Q,5)));
        maxpuntuacion3=[maxpuntuacion3;
            C(w),3*sum(str2double(table(Q,5))),i];
    end
end
R=find(table(:,2)==C(w));
if length(R)>0
    if i<=13
        puntuacion(w)=puntuacion(w)+
            sum(str2double(table(R,6)));
        maxpuntuacion1=[maxpuntuacion1;
            C(w),sum(str2double(table(R,6))),i];
    elseif i>13 && i<=22
        puntuacion(w)=puntuacion(w)+

```

```

        2*sum(str2double(table(R,6)));
maxpuntuacion2=[maxpuntuacion2;
    C(w),2*sum(str2double(table(R,6))),i];
else
puntuacion(w)=puntuacion(w)+
    3*sum(str2double(table(R,6)));
maxpuntuacion3=[maxpuntuacion3;
    C(w),3*sum(str2double(table(R,6))),i];
end
end
end
ordenpart=[];
puntuacion=round(puntuacion.*10000000)./10000000;
y=sort(puntuacion,'descend');
y=round(y.*10000000)./10000000;
contador=1;
while contador<=nombres
k=find(puntuacion==y(contador));
if length(k)==1
ordenpart=[ordenpart;C(k)];
puestospart(k,i)=contador;
contador=contador+1;
else
ordenpart=[ordenpart;C(k)];
puestospart(k,i)=contador:(contador+length(k)-1);
contador=contador+length(k);
end
end

E=[B,D,F];
end

```

Evolución de las asignaciones durante la competición

```

asignaciones=[];
for i=1:length(B)
    asignaciones=[asignaciones;length(find(aulti(:,1)==B(i)))];
    %Vector con las asignaciones para cada partido
end
B(find(asignaciones==max(asignaciones))) %Partido con
%mas asignaciones
B(find(asignaciones==min(asignaciones))) %Partidos con
%menos asignaciones
plot(1:64,asignaciones)
xlim([1,64])
ylim([0,50])
xlabel('Partidos')
ylabel('Asignaciones')
title('Evolucion del numero de asignaciones por partido')

```


Análisis de la máxima y mínima ganancia por jornada durante la competición

```
maxpuntuacion111=[];
for i=1:13
    maxpuntuacion11=maxpuntuacion1(find(maxpuntuacion1(:,3)==
        num2str(i)),:);
    for j=1:(size(maxpuntuacion11,1)-1)
        if maxpuntuacion11(j,1)==maxpuntuacion11(j+1,1)
            maxpuntuacion111=[maxpuntuacion111;
                maxpuntuacion11(j,1),
                str2double(maxpuntuacion11(j,2))+
                str2double(maxpuntuacion11(j+1,2)),i];
        elseif j==1
            maxpuntuacion111=[maxpuntuacion111;
                maxpuntuacion11(j,1),
                maxpuntuacion11(j,2),i];
        elseif j>1 &&
            maxpuntuacion11(j,1)~=maxpuntuacion11(j+1,1) &&
            maxpuntuacion11(j,1)~=maxpuntuacion11(j-1,1)
            maxpuntuacion111=[maxpuntuacion111;
                maxpuntuacion11(j,1),
                maxpuntuacion11(j,2),i];
        end
    end
end
maxpuntuacion222=[];
for i=14:22
    maxpuntuacion22=maxpuntuacion2(find(maxpuntuacion2(:,3)==
        num2str(i)),:);
    for j=1:(size(maxpuntuacion22,1)-1)
        if maxpuntuacion22(j,1)==maxpuntuacion22(j+1,1)
            maxpuntuacion222=[maxpuntuacion222;
                maxpuntuacion22(j,1),
                str2double(maxpuntuacion22(j,2))+
                str2double(maxpuntuacion22(j+1,2)),i];
        elseif j==1
            maxpuntuacion222=[maxpuntuacion222;
                maxpuntuacion22(j,1),
                maxpuntuacion22(j,2),i];
        elseif j>1 &&
            maxpuntuacion22(j,1)~=maxpuntuacion22(j+1,1) &&
            maxpuntuacion22(j,1)~=maxpuntuacion22(j-1,1)
            maxpuntuacion222=[maxpuntuacion222;
                maxpuntuacion22(j,1),
                maxpuntuacion22(j,2),i];
        end
    end
end
maxpuntuacion333=[];
maxpuntuacion33=maxpuntuacion3(find(maxpuntuacion3(:,3)==
    num2str(23)),:);
```

```

for j=1:(size(maxpuntuacion33,1)-1)
    if maxpuntuacion33(j,1)==maxpuntuacion33(j+1,1)
        maxpuntuacion333=[maxpuntuacion333;
            maxpuntuacion33(j,1),
            str2double(maxpuntuacion33(j,2))+
            str2double(maxpuntuacion33(j+1,2)),23];
    elseif j==1
        maxpuntuacion333=[maxpuntuacion333;
            maxpuntuacion33(j,1),
            maxpuntuacion33(j,2),23];
    elseif j>1 &&
        maxpuntuacion33(j,1)~=maxpuntuacion33(j+1,1) &&
        maxpuntuacion33(j,1)~=maxpuntuacion33(j-1,1)
        maxpuntuacion333= [maxpuntuacion333;
            maxpuntuacion33(j,1),
            maxpuntuacion33(j,2),23];
    end
end
maxpuntuacionfinal=[];
minpuntuacionfinal=[];
for k=1:13
    longjorn1=find(maxpuntuacion111(:,3)==num2str(k));
    maxj1=max(str2double(maxpuntuacion111(longjorn1,2)));
    minj1=min(str2double(maxpuntuacion111(longjorn1,2)));
    posmaxj1=find(maxpuntuacion111(:,2)==num2str(maxj1));
    posminj1=find(maxpuntuacion111(:,2)==num2str(minj1));
    maxpuntuacionfinal=[maxpuntuacionfinal;maxj1,
        maxpuntuacion111(posmaxj1(1),1)];
    minpuntuacionfinal=[minpuntuacionfinal;minj1,
        maxpuntuacion111(posminj1(1),1)];
end
for k=14:22
    longjorn2=find(maxpuntuacion222(:,3)==num2str(k));
    maxj2=max(str2double(maxpuntuacion222(longjorn2,2)));
    minj2=min(str2double(maxpuntuacion222(longjorn2,2)));
    posmaxj2=find(maxpuntuacion222(:,2)==num2str(maxj2));
    posminj2=find(maxpuntuacion222(:,2)==num2str(minj2));
    maxpuntuacionfinal=[maxpuntuacionfinal;maxj2,
        maxpuntuacion222(posmaxj2(1),1)];
    minpuntuacionfinal=[minpuntuacionfinal;minj2,
        maxpuntuacion222(posminj2(1),1)];
end

longjorn3=find(maxpuntuacion333(:,3)==num2str(23));
maxj3=max(str2double(maxpuntuacion333(longjorn3,2)));
minj3=min(str2double(maxpuntuacion333(longjorn3,2)));
posmaxj3=find(maxpuntuacion333(:,2)==num2str(maxj3));
posminj3=find(maxpuntuacion333(:,2)==num2str(minj3));
maxpuntuacionfinal=[maxpuntuacionfinal;maxj3,
    maxpuntuacion333(posmaxj3(1),1)];
minpuntuacionfinal=[minpuntuacionfinal;minj3,
    maxpuntuacion333(posminj3(1),1)];
plot(1:23,str2double(maxpuntuacionfinal(:,1)))
xlim([1 23])

```

```

ylim([0 100])
title('Evolucion de la mayor ganancia de puntos por jornada
a lo largo de la competicion')
xlabel('Jornadas')
ylabel('Puntuacion')
plot(1:23,-str2double(minpuntuacionfinal(:,1)), "red")
xlim([1 23])
ylim([0 100])
title('Evolucion de la mayor perdida de puntos por jornada
a lo largo de la competicion')
xlabel('Jornadas')
ylabel('Puntuacion')

```

Evolución de la posición de los participantes a lo largo de la competición

```

vectorjornada=1:23;
vectorposIonescuTulceaVence=puestospart(19,:);
vectorposManuteBol=puestospart(15,:);
vectorposKolmogorovSmirnov=puestospart(14,:);
vectorposDiegoReyer=puestospart(39,:);
vectorposMazepin=puestospart(12,:);
vectorposMiquelbs=puestospart(16,:);
vectorposmaraloncas=puestospart(43,:);
vectorposFingelmo=puestospart(6,:);
vectorposSantosD=puestospart(35,:);
%Comparacion posicion 3 primeros
plot(vectorjornada,-vectorposManuteBol)
title('Evolucion de los participantes')
xlabel('Jornadas')
ylabel('Posicion')
xlim([1 23])
ylim([-50 -1])
yticks([-50 -45 -40 -35 -30 -25 -20 -15 -10 -5 -1])
yticklabels({'50', '45', '40', '35', '30', '25', '20', '15',
'10', '5', '1'})
hold on
plot(vectorjornada,-vectorposDiegoReyer)
plot(vectorjornada,-vectorposKolmogorovSmirnov)
legend('Manute Bol', 'DiegoReyer', 'Kolmogorov Smirnov',
'Location', 'Best')
%Comparacion posicion 3 ultimos
hold off
plot(vectorjornada,-vectorposIonescuTulceaVence)
title('Evolucion de los participantes')
xlabel('Jornadas')
ylabel('Posicion')
xlim([1 23])
ylim([-50 -1])
yticks([-50 -45 -40 -35 -30 -25 -20 -15 -10 -5 -1])
yticklabels({'50', '45', '40', '35', '30', '25', '20', '15',
'10', '5', '1'})

```

```

hold on
plot(vectorjornada,-vectorposMazepin)
plot(vectorjornada,-vectorposMiquelbs)
legend('Ionescu-TulceaVence', 'Mazepin', 'Miquel\_bs',
      'Location', 'Best')
%Comparacion posicion 3 intermedios
hold off
plot(vectorjornada,-vectorposmaralencas)
title('Evolucion de los participantes')
xlabel('Jornadas')
ylabel('Posicion')
xlim([1 23])
ylim([-50 -1])
yticks([-50 -45 -40 -35 -30 -25 -20 -15 -10 -5 -1])
ticklabels({'50', '45', '40', '35', '30', '25', '20', '15',
           '10', '5', '1'})
hold on
plot(vectorjornada,-vectorposFingelmo)
plot(vectorjornada,-vectorposSantosD)
legend('maralencas', 'Fingelmo', 'SantosD', 'Location', 'Best')

```

Análisis de la precisión y el margen medio de imprecisión en las asignaciones

a) Análisis de la *precisión* en las asignaciones

```

E=[B,F];
contadorprecisas=0;
for i=1:size(Aulti,1)
    if Aulti(i,1)==Aulti(i,2) && Aulti(i,3)==Aulti(i,4) &&
        Aulti(i,5)==Aulti(i,6)
        contadorprecisas=contadorprecisas+1;
    end
end
%Si quisiesemos mirar la proporcion con el numero total
%de apuestas activadas hacemos:
contadorprecisas/(size(Aulti,1))
%Si quisiesemos mirar la proporcion de asignaciones
%precisas POR JORNADA
matrizprecisionjornadas=zeros(23,1);
asignporjorn=[];
for g=1:23
E(find(E(:,2)~=num2str(g)),:)=[];
npartidos=size(E,1);
contadorprecisasjornada=0;
numpartidos=E(:,1);
aprecis=[];
Aprecis=[];
for q=1:npartidos
    partidoact=E(q,1);
    pospartido=find(aulti(:,1)==partidoact);
    aprecis=[aprecis;aulti(pospartido,:)];
end
end

```

```

    Aprecis=[Aprecis;Aulti(pospartido,:)];
end
asignporjorn=[asignporjorn;size(aprecis,1)];
tpartidos=aprecis(:,1);
if isempty(tpartidos)==0
    for i=1:length(tpartidos)
        if Aprecis(i,1)==Aprecis(i,2) &&
            Aprecis(i,3)==Aprecis(i,4) &&
            Aprecis(i,5)==Aprecis(i,6)
            contadorprecisasjornada=contadorprecisasjornada+1;
        end
    end
end

end
fprintf('La jornada %s ha tenido una proporcion
de asignaciones precisas de %.3f \n',num2str(g),
contadorprecisasjornada/(asignporjorn(g)))
matrizprecisionjornadas(g)=contadorprecisasjornada
/(asignporjorn(g));
E=[B,F];
end
plot(1:23,matrizprecisionjornadas)
xlim([1 23])
ylim([0 1])
title('Evolucion de la precision a lo largo de la competicion')
xlabel('Jornadas')
ylabel('Proporcion asignaciones precisas')
%Si quisiesemos comparar la precision de las asignaciones
%POR PARTICIPANTE
matrizprecisionparticipantes=zeros(nnombres,1);
for g=1:nnombres
    participante=C(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    aprecis2=aulti(posparticipante,:);
    Aprecis2=Aulti(posparticipante,:);
    contadorprecisasparticipante=0;
    tnombres=aprecis2(:,2);
    if isempty(tnombres)==0
        for i=1:length(tnombres)
            if Aprecis2(i,1)==Aprecis2(i,2) &&
                Aprecis2(i,3)==Aprecis2(i,4) &&
                Aprecis2(i,5)==Aprecis2(i,6)
                contadorprecisasparticipante=
                    contadorprecisasparticipante+1;
            end
        end
    end
    fprintf('El participante %s ha tenido una proporcion
de asignaciones precisas de %.3f \n',participante,
contadorprecisasparticipante/(length(tnombres)))
    matrizprecisionparticipantes(g)=contadorprecisasparticipante/
(length(tnombres));
end
end

```

```

V=find(C=="Manuel.martin" | C=="YSale" | C=="choru"
      | C=="Grislydiamond" | C=="Sgs");
matrizprecisionparticipantes(V)=[];
H=C;
H(V)=[];
H(find(matrizprecisionparticipantes==0));
bar(matrizprecisionparticipantes)
title('Proporcion asignaciones precisas por participante')

```

b) Análisis del *margen medio de imprecisión* en las asignaciones

```

matrizmargenimprecision=[];
for g=1:23
E(find(E(:,2)~=num2str(g)),:)=[]; npartidos=size(E,1);
numpartidos=E(:,1);
aprecis3=[];
Aprecis3=[];
for q=1:length(numpartidos)
partidoact=E(q,1);
pospartido=find(aulti(:,1)==partidoact);
aprecis3=[aprecis3;aulti(pospartido,:)];
Aprecis3=[Aprecis3;Aulti(pospartido,:)];
end
tpartidos=aprecis3(:,1);
if isempty(tpartidos)==0
for i=1:length(tpartidos)
if Aprecis3(i,1)~=Aprecis3(i,2)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,2)-Aprecis3(i,1),g]];
if Aprecis3(i,3)~=Aprecis3(i,4)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,4)-Aprecis3(i,3),g]];
if Aprecis3(i,5)~=Aprecis3(i,6)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,6)-Aprecis3(i,5),g]];
end
elseif Aprecis3(i,5)~=Aprecis3(i,6)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,6)-Aprecis3(i,5),g]];
end
elseif Aprecis3(i,3)~=Aprecis3(i,4)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,4)-Aprecis3(i,3),g]];
if Aprecis3(i,5)~=Aprecis3(i,6)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,6)-Aprecis3(i,5),g]];
end
elseif Aprecis3(i,5)~=Aprecis3(i,6)
matrizmargenimprecision=[matrizmargenimprecision;
[Aprecis3(i,6)-Aprecis3(i,5),g]];
end
end
end
end
end

```

```

E=[B,F];
end
%Comparacion de la imprecision entre los usuarios
%imprecisos por jornadas
mediaimprecjorn=zeros(23,1);
for i=1:23
    matrizmargenimprecision2=matrizmargenimprecision(find(
        matrizmargenimprecision(:,2)==i),:);
    mediaimprecjorn(i)=mean(matrizmargenimprecision2(:,1));
end
plot(1:23,mediaimprecjorn)
title('Evolucion del margen medio de imprecision por jornada')
xlabel('Jornadas')
ylabel('Margen medio imprecision')
xlim([1 23])
ylim([0 0.3])
%Comparacion de la imprecision entre todos los usuarios (precisos
%e imprecisos) por jornada
mediaimprecjorntotal=zeros(23,1);
for i=1:23
    matrizmargenimprecision2=matrizmargenimprecision(find(
        matrizmargenimprecision(:,2)==i),:);
    mediaimprecjorntotal(i)=sum(matrizmargenimprecision2(:,1))/
        (3*assignporjorn(i));
end
plot(1:23,mediaimprecjorntotal,"red")
title('Evolucion conjunta del margen medio total
de imprecision total por jornada')
xlabel('Jornadas')
ylabel('Margen medio total imprecision')
xlim([1 23])
ylim([0 0.2])

```

Análisis de la *consistencia tipo 1* en las asignaciones

a) Análisis de *evitar la pérdida segura* en las asignaciones

```

E=B;
Q=readtable('MotorWCC.RStudio.xlsm','sheet','Table1',
    'Range','F:F'); %Lee
%la formacion de todos los usuarios registrados en la web
%en formato tabla
Q=table2array(Q); %Lo convertimos en un vector tipo cell
Q(1)=[]; %Eliminamos el primer elemento de Q, pues
%era el encabezado
Q(50)=[];
Q=convertCharsToStrings(Q);
%Proporcion de asignaciones que evitan la perdida segura
contadorevitarperdidasegura=0;
for g=1:size(Aulti,1)
    if (Aulti(g,1)+Aulti(g,3)+Aulti(g,5)<=1) &&
        (round((Aulti(g,2)+Aulti(g,4)+Aulti(g,6))*1000000)/1000000>=1)

```

```

        contadorevitarperdidasegura=contadorevitarperdidasegura+1;
    end
end
fprintf('La proporcion de las asignaciones en las que se evita la
perdida segura es de %.3f \n',
    contadorevitarperdidasegura/size(Aulti,1))
%Test chi-cuadrado proporcion asignaciones que
%evitan la perdida segura segun formacion
posgrupo1=find(Q=="Matematicas" |
    Q=="Matematicas y Fisica" | Q=="Fisica");
posgrupo2=find(Q=="Ingenieria");
posgrupo3=find(Q=="ESO" |
    Q=="Bachillerato" | Q=="Grado Superior");
posgrupo4=find(Q=="Economia" | Q=="Derecho" | Q=="ADE");
posgrupo5=find(Q=="Biologia" | Q=="Biotecnologia" |
    Q=="Enfermeria" | Q=="Quimica");
grupo1=C(posgrupo1);
grupo2=C(posgrupo2);
grupo3=C(posgrupo3);
grupo4=C(posgrupo4);
grupo5=C(posgrupo5);
asigngrupo1=0;
asigngrupo2=0;
asigngrupo3=0;
asigngrupo4=0;
asigngrupo5=0;
contadorevitarperdidaseguragrupo1=0;
contadorevitarperdidaseguragrupo2=0;
contadorevitarperdidaseguragrupo3=0;
contadorevitarperdidaseguragrupo4=0;
contadorevitarperdidaseguragrupo5=0;
%Grupo 1
for i=1:length(grupo1)
    partact=grupo1(i);
    K=find(aulti(:,2)==partact);
    asigngrupo1=asigngrupo1+length(K);
    for j=1:length(K)
        if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
            (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),6))*
                1000000)/1000000)>=1)
            contadorevitarperdidaseguragrupo1=
                contadorevitarperdidaseguragrupo1+1;
        end
    end
end
end
%Grupo 2
for i=1:length(grupo2)
    partact=grupo2(i);
    K=find(aulti(:,2)==partact);
    asigngrupo2=asigngrupo2+length(K);
    for j=1:length(K)
        if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
            (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),6))*
                1000000)/1000000)>=1)

```



```

        contadorevitarperdidaseguragrupo2=
            contadorevitarperdidaseguragrupo2+1;
    end
end
end
%Grupo 3
for i=1:length(grupo3)
partact=grupo3(i);
K=find(aulti(:,2)==partact);
asigngrupo3=asigngrupo3+length(K);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000)>=1)
        contadorevitarperdidaseguragrupo3=
            contadorevitarperdidaseguragrupo3+1;
    end
end
end
end
%Grupo 4
for i=1:length(grupo4)
partact=grupo4(i);
K=find(aulti(:,2)==partact);
asigngrupo4=asigngrupo4+length(K);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000)>=1)
        contadorevitarperdidaseguragrupo4=
            contadorevitarperdidaseguragrupo4+1;
    end
end
end
end
%Grupo 5
for i=1:length(grupo5)
partact=grupo5(i);
K=find(aulti(:,2)==partact);
asigngrupo5=asigngrupo5+length(K);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000)>=1)
        contadorevitarperdidaseguragrupo5=
            contadorevitarperdidaseguragrupo5+1;
    end
end
end
end
promedioevitperdsegrupol=
    contadorevitarperdidaseguragrupo1/asigngrupo1;
promedioevitperdsegrupol2=
    contadorevitarperdidaseguragrupo2/asigngrupo2;
promedioevitperdsegrupol3=
    contadorevitarperdidaseguragrupo3/asigngrupo3;
promedioevitperdsegrupol4=

```

```

    contadorevitarperdidaseguragrupo4/assigngrupo4;
promedioevitperdsegrupo5=
    contadorevitarperdidaseguragrupo5/assigngrupo5;
c=bar([promedioevitperdsegrupo1,promedioevitperdsegrupo2,
promedioevitperdsegrupo3,promedioevitperdsegrupo4,
promedioevitperdsegrupo5],.40,"red")
ylim([0,1])
set(gca,'xticklabel',["Grupo 1","Grupo 2","Grupo 3",
    "Grupo 4","Grupo 5"])
title('Proporcion asignaciones en las que se
    evita la perdida segura')
c.FaceColor=[0.4940 0.1840 0.5560];
%Test de proporciones de asignaciones que evitan
%la perdida segura entre participantes con formacion en
%Matematicas / Doble Grado Matematicas y Fisica
%y los que tienen otro tipo de formacion.
leerpuntuacion=strcat("H1:", "H", num2str(nnombres+1));
puntuacion=table2array(readtable('MotorWCC_RStudio.xlsm','sheet',
'Clasificacion','Range',leerpuntuacion));
particactivos=C;
posA=find(Q=="Matematicas" | Q=="Matematicas y Fisica");
equipoA=particactivos(posA);
posB=setdiff(1:50,posA);
equipoB=particactivos(posB);
vectorinactivos=find(puntuacion==0);
posinactivosA=intersect(posA,vectorinactivos);
posinactivosB=intersect(posB,vectorinactivos);
eliminarA=C(posinactivosA);
eliminarB=C(posinactivosB);
vectposeliminarA=zeros(length(posinactivosA),1);
for i=1:length(posinactivosA)
    vectposeliminarA(i)=find(equipoA==eliminarA(i));
end
vectposeliminarB=zeros(length(posinactivosB),1);
for i=1:length(posinactivosB)
    vectposeliminarB(i)=find(equipoB==eliminarB(i));
end
equipoA(vectposeliminarA)=[];
equipoB(vectposeliminarB)=[];
contadorevitarperdidaseguraequipoA=0;
numparteequipoA=0;
for g=1:length(equipoA)
    participante=equipoA(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    numparteequipoA=numparteequipoA+length(posparticipante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=areult(:,2);
    for i=1:length(tnombres)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorevitarperdidaseguraequipoA=

```

```

        contadorevitarperdidaseguraequipoA+1;
    end
end
end
contadorevitarperdidaseguraequipoB=0;
numparteequipoB=0;
for g=1:length(equipoB)
    participante=equipoB(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    numparteequipoB=numparteequipoB+length(posparticipante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=aresult(:,2);
    for i=1:length(tnombres)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorevitarperdidaseguraequipoB=
                contadorevitarperdidaseguraequipoB+1;
        end
    end
end
end
promedioA=contadorevitarperdidaseguraequipoA/numparteequipoA;
promedioB=contadorevitarperdidaseguraequipoB/numparteequipoB;
c=bar([promedioA;promedioB],.40,"red")
ylim([0,1])
set(gca,'xticklabel',"Matematicas o Doble Grado",
    "Otra formacion")
title('Proporcion asignaciones en las que se
    evita la perdida segura')
c.FaceColor=[0 0.4470 0.7410];
%Test de proporciones de asignaciones que evitan la perdida segura
%entre participantes distinto genero.
W=readtable('MotorWCC_RStudio.xlsm','sheet','Table1',
    'Range','G:G'); %Lee el genero de todos los
%usuarios registrados en la web en formato tabla
W=table2array(W); %Lo convertimos en un vector tipo cell
W(1)=[]; %Eliminamos el primer elemento de Q,
%pues era el encabezado
W(50)=[];
W=convertCharsToStrings(W); %Lo convertimos en un
%vector de tipo string
particactivos=C;
posC=find(W=="Hombre / Male");
equipoC=particactivos(posC);
posD=setdiff(1:50,posC);
equipoD=particactivos(posD);
posinactivosC=intersect(posC,vectorinactivos);
posinactivosD=intersect(posD,vectorinactivos);
eliminarC=C(posinactivosC);
eliminarD=C(posinactivosD);
vectposeliminarC=zeros(length(posinactivosC),1);
for i=1:length(posinactivosC)

```

```

        vectposeeliminarC(i)=find(equipoC==eliminarC(i));
end
vectposeeliminarD=zeros(length(posinactivosD),1);
for i=1:length(posinactivosD)
    vectposeeliminarD(i)=find(equipoD==eliminarD(i));
end
equipoC(vectposeeliminarC)=[];
equipoD(vectposeeliminarD)=[];
contadorevitarperdidaseguraequipoC=0;
numparteequipoC=0;
for g=1:length(equipoC)
    participante=equipoC(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    numparteequipoC=numparteequipoC+length(posparticipante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=areult(:,2);
    for i=1:length(tnombres)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorevitarperdidaseguraequipoC=
                contadorevitarperdidaseguraequipoC+1;
        end
    end
end
contadorevitarperdidaseguraequipoD=0;
numparteequipoD=0;
for g=1:length(equipoD)
    participante=equipoD(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    numparteequipoD=numparteequipoD+length(posparticipante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=areult(:,2);
    for i=1:length(tnombres)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorevitarperdidaseguraequipoD=
                contadorevitarperdidaseguraequipoD+1;
        end
    end
end
end
promedioC=contadorevitarperdidaseguraequipoC/numparteequipoC;
promedioD=contadorevitarperdidaseguraequipoD/numparteequipoD;
c=bar([promedioC;promedioD],.40,"blue")
ylim([0,1])
set(gca,'xticklabel',["Hombre","Mujer"])
title('Proporcion asignaciones en las que se
    evita la perdida segura')
c.FaceColor=[0.8500 0.3250 0.0980];

```

```

%Test de proporciones de asignaciones que evitan la perdida segura
%entre participantes con conocimientos previos
%en probabilidades imprecisas y los que no parten
%con dichas nociones
posE=find(C=="Walley4Ever" | C=="Arthur Van Camp" |
    C=="Kalmanhorof" | C=="Keano" | C=="YSale" |
    C=="MOJI" | C=="NachoM");
equipoE=C(posE);
posF=setdiff(1:50,posE);
equipoF=C(posF);
posinactivosE=intersect(posE,vectorinactivos);
posinactivosF=intersect(posF,vectorinactivos);
eliminarE=C(posinactivosE);
eliminarF=C(posinactivosF);
vectposeliminarE=zeros(length(posinactivosE),1);
for i=1:length(posinactivosE)
    vectposeliminarE(i)=find(equipoE==eliminarE(i));
end
vectposeliminarF=zeros(length(posinactivosF),1);
for i=1:length(posinactivosF)
    vectposeliminarF(i)=find(equipoF==eliminarF(i));
end
equipoE(vectposeliminarE)=[];
equipoF(vectposeliminarF)=[];
contadorevitarperdidaseguraequipoE=0;
numparteequipoE=0;
for g=1:length(equipoE)
    participante=equipoE(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    numparteequipoE=numparteequipoE+length(posparticipante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=areult(:,2);
    for i=1:length(tnombres)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorevitarperdidaseguraequipoE=
                contadorevitarperdidaseguraequipoE+1;
        end
    end
end
end
contadorevitarperdidaseguraequipoF=0;
numparteequipoF=0;
for g=1:length(equipoF)
    participante=equipoF(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    numparteequipoF=numparteequipoF+length(posparticipante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=areult(:,2);
    for i=1:length(tnombres)

```

```

        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorevitarperdidaseguraequipoF=
                contadorevitarperdidaseguraequipoF+1;
        end
    end
end
end
promedioE=contadorevitarperdidaseguraequipoE/numpartequipoE;
promedioF=contadorevitarperdidaseguraequipoF/numpartequipoF;
c=bar([promedioE;promedioF],.40,"green")
ylim([0,1])
set(gca,'xticklabel',["Expertos imprecisas",
    "No expertos en imprecisas"])
title('Proporcion asignaciones en las que se
    evita la perdida segura')
c.FaceColor=[0.9290 0.6940 0.1250];

```

b) Análisis de la *coherencia* en las asignaciones

```

contadorcoherencia=0;
for g=1:size(Aulti,1)
    if (Aulti(g,1)+Aulti(g,3)+Aulti(g,6)<=1) &&
        (Aulti(g,1)+Aulti(g,4)+Aulti(g,5)<=1) &&
        (Aulti(g,2)+Aulti(g,3)+Aulti(g,5)<=1) &&
        (round((Aulti(g,2)+Aulti(g,4)+Aulti(g,5))*
            1000000)/1000000>=1) &&
        (round((Aulti(g,2)+Aulti(g,3)+Aulti(g,6))*
            1000000)/1000000>=1) &&
        (round((Aulti(g,1)+Aulti(g,4)+Aulti(g,6))*
            1000000)/1000000>=1)
        contadorcoherencia=contadorcoherencia+1;
    end
end
fprintf('La proporcion de las asignaciones coherentes es
de %.4f \n',contadorcoherencia/size(Aulti,1))
%Test chi-cuadrado proporcion asignaciones coherentes
%segun formacion
contadorcoherenciagrupo1=0;
contadorcoherenciagrupo2=0;
contadorcoherenciagrupo3=0;
contadorcoherenciagrupo4=0;
contadorcoherenciagrupo5=0;
%Grupo 1
for i=1:length(grupo1)
    partact=grupo1(i);
    K=find(aulti(:,2)==partact);
    for j=1:length(K)
        if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),6)<=1) &&
            (Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),5)<=1) &&
            (Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
            (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),5))*
                1000000)/1000000>=1) &&

```

```

        (round((Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),6))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000>=1)
        contadorcoherenciagrupo1=contadorcoherenciagrupo1+1;
    end
end
end
%Grupo 2
for i=1:length(grupo2)
partact=grupo2(i);
K=find(aulti(:,2)==partact);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),6)<=1) &&
        (Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),5)<=1) &&
        (Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),5))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),6))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000>=1)
        contadorcoherenciagrupo2=contadorcoherenciagrupo2+1;
    end
end
end
end
%Grupo 3
for i=1:length(grupo3)
partact=grupo3(i);
K=find(aulti(:,2)==partact);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),6)<=1) &&
        (Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),5)<=1) &&
        (Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),5))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),6))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000>=1)
        contadorcoherenciagrupo3=contadorcoherenciagrupo3+1;
    end
end
end
end
%Grupo 4
for i=1:length(grupo4)
partact=grupo4(i);
K=find(aulti(:,2)==partact);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),6)<=1) &&
        (Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),5)<=1) &&
        (Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),5))*
            1000000)/1000000>=1) &&

```

```

        (round((Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),6))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000>=1)
        contadorcoherenciagrupo4=contadorcoherenciagrupo4+1;
    end
end
end
%Grupo 5
for i=1:length(grupo5)
partact=grupo5(i);
K=find(aulti(:,2)==partact);
for j=1:length(K)
    if (Aulti(K(j),1)+Aulti(K(j),3)+Aulti(K(j),6)<=1) &&
        (Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),5)<=1) &&
        (Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),5)<=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),4)+Aulti(K(j),5))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),2)+Aulti(K(j),3)+Aulti(K(j),6))*
            1000000)/1000000>=1) &&
        (round((Aulti(K(j),1)+Aulti(K(j),4)+Aulti(K(j),6))*
            1000000)/1000000>=1)
        contadorcoherenciagrupo5=contadorcoherenciagrupo5+1;
    end
end
end
promediocoherenciagrupo1=contadorcoherenciagrupo1/asigngrupo1;
promediocoherenciagrupo2=contadorcoherenciagrupo2/asigngrupo2;
promediocoherenciagrupo3=contadorcoherenciagrupo3/asigngrupo3;
promediocoherenciagrupo4=contadorcoherenciagrupo4/asigngrupo4;
promediocoherenciagrupo5=contadorcoherenciagrupo5/asigngrupo5;
c=bar([promediocoherenciagrupo1,promediocoherenciagrupo2,
    promediocoherenciagrupo3,promediocoherenciagrupo4,
    promediocoherenciagrupo5],.40,"red")
ylim([0,1])
set(gca,'xticklabel',["Grupo 1","Grupo 2","Grupo 3",
    "Grupo 4","Grupo 5"])
title('Proporcion de asignaciones coherentes')
c.FaceColor=[0.4940 0.1840 0.5560];
%Test de proporciones de asignaciones coherentes
%entre participantes con formacion en Matematicas /
%Doble Grado Matematicas y Fisica y los que tienen
%otro tipo de formacion.
contadorcoherenciaequipoA=0;
for g=1:length(equipoA)
participante=equipoA(g);
tnombres=aulti(:,2);
posparticipante=find(tnombres==participante);
aresult=aulti(posparticipante,:);
Aresult=Aulti(posparticipante,:);
tnombres=aresult(:,2);
for i=1:length(tnombres)
    if (Aresult(i,1)+Aresult(i,3)+Aresult(i,6)<=1) &&
        (Aresult(i,1)+Aresult(i,4)+Aresult(i,5)<=1) &&

```



```

        (Aresult(i,2)+Aresult(i,3)+Aresult(i,5)<=1) &&
        (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,5))*
            1000000)/1000000>=1) &&
        (round((Aresult(i,2)+Aresult(i,3)+Aresult(i,6))*
            1000000)/1000000>=1) &&
        (round((Aresult(i,1)+Aresult(i,4)+Aresult(i,6))*
            1000000)/1000000>=1)
        contadorcoherenciaequipoA=contadorcoherenciaequipoA+1;
    end
end
end
contadorcoherenciaequipoB=0;
for g=1:length(equipoB)
    participante=equipoB(g);
    tnombrs=aulti(:,2);
    posparticipante=find(tnombrs==participante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombrs=areult(:,2);
    for i=1:length(tnombrs)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,6)<=1) &&
            (Aresult(i,1)+Aresult(i,4)+Aresult(i,5)<=1) &&
            (Aresult(i,2)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,5))*
                1000000)/1000000>=1) &&
            (round((Aresult(i,2)+Aresult(i,3)+Aresult(i,6))*
                1000000)/1000000>=1) &&
            (round((Aresult(i,1)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorcoherenciaequipoB=contadorcoherenciaequipoB+1;
        end
    end
end
end
promediocohA=contadorcoherenciaequipoA/numparteequipoA;
promediocohB=contadorcoherenciaequipoB/numparteequipoB;
c=bar([promediocohA;promediocohB],.40,"red")
ylim([0,1])
set(gca,'xticklabel',"Matematicas o Doble Grado",
    "Otra formacion")
title('Proporcion de asignaciones coherentes')
c.FaceColor=[0 0.4470 0.7410];
%Test de proporciones de asignaciones coherentes
%entre participantes de distinto genero.
contadorcoherenciaequipoC=0;
for g=1:length(equipoC)
    participante=equipoC(g);
    tnombrs=aulti(:,2);
    posparticipante=find(tnombrs==participante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombrs=areult(:,2);
    for i=1:length(tnombrs)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,6)<=1) &&
            (Aresult(i,1)+Aresult(i,4)+Aresult(i,5)<=1) &&

```

```

        (Aresult(i,2)+Aresult(i,3)+Aresult(i,5)<=1) &&
        (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,5))*
            1000000)/1000000>=1) &&
        (round((Aresult(i,2)+Aresult(i,3)+Aresult(i,6))*
            1000000)/1000000>=1) &&
        (round((Aresult(i,1)+Aresult(i,4)+Aresult(i,6))*
            1000000)/1000000>=1)
        contadorcoherenciaequipoC=contadorcoherenciaequipoC+1;
    end
end
end
contadorcoherenciaequipoD=0;
for g=1:length(equipoD)
    participante=equipoD(g);
    tnombrs=aulti(:,2);
    posparticipante=find(tnombrs==participante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombrs=areult(:,2);
    for i=1:length(tnombrs)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,6)<=1) &&
            (Aresult(i,1)+Aresult(i,4)+Aresult(i,5)<=1) &&
            (Aresult(i,2)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,5))*
                1000000)/1000000>=1) &&
            (round((Aresult(i,2)+Aresult(i,3)+Aresult(i,6))*
                1000000)/1000000>=1) &&
            (round((Aresult(i,1)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorcoherenciaequipoD=contadorcoherenciaequipoD+1;
        end
    end
end
end
promediocohC=contadorcoherenciaequipoC/numparteequipoC;
promediocohD=contadorcoherenciaequipoD/numparteequipoD;
c=bar([promediocohC;promediocohD],.40,"blue")
ylim([0,1])
set(gca,'xticklabel',["Hombre","Mujer"])
title('Proporcion de asignaciones coherentes')
c.FaceColor=[0.8500 0.3250 0.0980];
%Test de proporciones de asignaciones coherentes entre
%participantes con conocimientos previos en probabilidades
%imprecisas y los que no parten con dichas nociones
contadorcoherenciaequipoE=0;
for g=1:length(equipoE)
    participante=equipoE(g);
    tnombrs=aulti(:,2);
    posparticipante=find(tnombrs==participante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombrs=areult(:,2);
    for i=1:length(tnombrs)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,6)<=1) &&
            (Aresult(i,1)+Aresult(i,4)+Aresult(i,5)<=1) &&

```

```

        (Aresult(i,2)+Aresult(i,3)+Aresult(i,5)<=1) &&
        (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,5))*
            1000000)/1000000>=1) &&
        (round((Aresult(i,2)+Aresult(i,3)+Aresult(i,6))*
            1000000)/1000000>=1) &&
        (round((Aresult(i,1)+Aresult(i,4)+Aresult(i,6))*
            1000000)/1000000>=1)
        contadorcoherenciaequipoE=contadorcoherenciaequipoE+1;
    end
end
end
contadorcoherenciaequipoF=0;
for g=1:length(equipoF)
    participante=equipoF(g);
    tnombres=aulti(:,2);
    posparticipante=find(tnombres==participante);
    aresult=aulti(posparticipante,:);
    Aresult=Aulti(posparticipante,:);
    tnombres=aresult(:,2);
    for i=1:length(tnombres)
        if (Aresult(i,1)+Aresult(i,3)+Aresult(i,6)<=1) &&
            (Aresult(i,1)+Aresult(i,4)+Aresult(i,5)<=1) &&
            (Aresult(i,2)+Aresult(i,3)+Aresult(i,5)<=1) &&
            (round((Aresult(i,2)+Aresult(i,4)+Aresult(i,5))*
                1000000)/1000000>=1) &&
            (round((Aresult(i,2)+Aresult(i,3)+Aresult(i,6))*
                1000000)/1000000>=1) &&
            (round((Aresult(i,1)+Aresult(i,4)+Aresult(i,6))*
                1000000)/1000000>=1)
            contadorcoherenciaequipoF=contadorcoherenciaequipoF+1;
        end
    end
end
end
promediocohE=contadorcoherenciaequipoE/numparteequipoE;
promediocohF=contadorcoherenciaequipoF/numparteequipoF;
c=bar([promediocohE;promediocohF],.40,"green")
ylim([0,1])
set(gca,'xticklabel',["Expertos imprecisas",
    "No expertos imprecisas"])
title('Proporcion de asignaciones coherentes')
c.FaceColor=[0.9290 0.6940 0.1250];

```

Análisis de la *consistencia tipo 2* en las asignaciones

Función auxiliar *funcionpermut*

```

function[contador]=funcionpermut(a1,b1,c1,a2,b2,c2,F1,F2,f)
contador=0;
matrizperm=perms([a1,b1,c1]);
aa=[a2,b2,c2];
nn=numel(aa);
ii = mod(bsxfun(@plus, 1:nn, (0:nn-1).')-1, nn) + 1;
matrizpermcicl=aa(ii);

```

```

for bb=1:size(matrizperm,1)
    for cc=1:size(matrizpermcicl,1)
        if f(matrizperm(bb,1),matrizperm(bb,2),
            matrizperm(bb,3),matrizpermcicl(cc,1),
            matrizpermcicl(cc,2),matrizpermcicl(cc,3),
            F1,F2)<=
                (2*F1*matrizperm(bb,3)*matrizpermcicl(cc,3))
            contador=contador+1;
        end
    end
end
end
end

```

Análisis de la *coherencia tipo 2* en las asignaciones

```

D=readtable('MotorWCC_RStudio.xlsm','sheet','Partidos',
    'Range','D:D');
D=table2array(D);
E=[B,D];
asignjorn=zeros(64,1);
matrizcontadorcoherencia=zeros(nnombres,64);
syms x y z p q t u r
f=@(x,y,z,p,q,t,u,r) max([u*(q^2+t^2-p^2),r*(y^2+z^2-x^2)])+
    max([u*(p^2+t^2-q^2),r*(x^2+z^2-y^2)]);
F=[];
contadortipo1=0;
contadortipo2=0;
contadortipo3=0;
contadortipo4=0;
contadortipo5=0;
for i=1:64 %Hubo 64 partidos
    E(find(E(:,2)~=num2str(i)),:)=[];
    %Se eliminan las apuestas de partidos que aun no se jugaron:
    aresult=[];
    Aresult=[];
    Xresult=[];
    U=find(aulti(:,1)==E(1,1));
    aresult=[aresult;aulti(U,:)];
    Aresult=[Aresult;Aulti(U,:)];
    Xresult=[Xresult;Xulti(U,:)];
    asignjorn(i)=size(aresult,1);
    tnombres=aresult(:,2);
    matrizriesgo=[];
    for j=1:length(tnombres)
        matrizriesgo=[matrizriesgo;
            (1-Aresult(j,2))^2+(Aresult(j,3))^2+(Aresult(j,5))^2,
            (1-Aresult(j,4))^2+(Aresult(j,1))^2+(Aresult(j,5))^2,
            (1-Aresult(j,6))^2+(Aresult(j,1))^2+(Aresult(j,3))^2];
    end

    for k=1:nnombres
        jugadoract=C(k);
        U=find(tnombres==jugadoract);
    end
end

```

```

if length(U)>0
    if (Aresult(U,1)==Aresult(U,2)) &&
        (Aresult(U,3)==Aresult(U,4)) &&
        (Aresult(U,5)==Aresult(U,6)) &&
        (round((Aresult(U,1)+Aresult(U,3)+Aresult(U,5))*
            1000000)/1000000==1)
            contadortipol=contadortipol+1;
    elseif (Aresult(U,1)==Aresult(U,2) &&
        Aresult(U,3)==Aresult(U,4)) &&
        (Aresult(U,5)==Aresult(U,6)) &&
        (round((Aresult(U,1)+Aresult(U,3)+Aresult(U,5))*
            1000000)/1000000~=1)
        matrizcontadorcoherencia(k,i)=
            matrizcontadorcoherencia(k,i)+1;
        contadortipo2=contadortipo2+1;
    elseif ((Aresult(U,1)==Aresult(U,2)) &&
        (Aresult(U,3)==Aresult(U,4)) &&
        (Aresult(U,5)~=Aresult(U,6))) ||
        ((Aresult(U,1)==Aresult(U,2)) &&
        (Aresult(U,3)~=Aresult(U,4)) &&
        (Aresult(U,5)==Aresult(U,6))) ||
        ((Aresult(U,1)~=Aresult(U,2)) &&
        (Aresult(U,3)==Aresult(U,4)) &&
        (Aresult(U,5)==Aresult(U,6)))
        matrizcontadorcoherencia(k,i)=
            matrizcontadorcoherencia(k,i)+1;
        contadortipo3=contadortipo3+1;
    else
        contadortipo4=contadortipo4+1;
        riesgojugl=matrizriesgo(U,:);
        S1=[Aresult(U,1),Aresult(U,3),Aresult(U,6);
            Aresult(U,1),Aresult(U,4),Aresult(U,5);
            Aresult(U,2),Aresult(U,3),Aresult(U,5)];
        ladoa1=sqrt((S1(1,1)-S1(2,1))^2+(S1(1,2)-S1(2,2))^2+
            (S1(1,3)-S1(2,3))^2);
        ladob1=sqrt((S1(1,1)-S1(3,1))^2+(S1(1,2)-S1(3,2))^2+
            (S1(1,3)-S1(3,3))^2);
        ladoc1=sqrt((S1(2,1)-S1(3,1))^2+(S1(2,2)-S1(3,2))^2+
            (S1(2,3)-S1(3,3))^2);
        semiper1=(ladoa1+ladob1+ladoc1)/2;
        F1=sqrt(semiper1*(semiper1-ladoa1)*
            (semiper1-ladob1)*(semiper1-ladoc1));
        for h=1:length(tnombres)
            if h~=U
                riesgojug2=matrizriesgo(h,:);
                S2=[Aresult(h,1),Aresult(h,3),Aresult(h,6);
                    Aresult(h,1),Aresult(h,4),Aresult(h,5);
                    Aresult(h,2),Aresult(h,3),Aresult(h,5)];
                ladoa2=sqrt((S2(1,1)-S2(2,1))^2+
                    (S2(1,2)-S2(2,2))^2+(S2(1,3)-S2(2,3))^2);
                ladob2=sqrt((S2(1,1)-S2(3,1))^2+
                    (S2(1,2)-S2(3,2))^2+(S2(1,3)-S2(3,3))^2);
                ladoc2=sqrt((S2(2,1)-S2(3,1))^2+
                    (S2(2,2)-S2(3,2))^2+(S2(2,3)-S2(3,3))^2);

```

```

semiper2=(ladoa2+ladob2+ladoc2)/2;
F2=sqrt (semiper2*(semiper2-ladoa2)*
(semiper2-ladob2)*(semiper2-ladoc2));
contadortriangulo=funcionpermut (ladoa1,ladob1,
ladoc1,ladoa2,ladob2,ladoc2,F1,F2,f);
if (((riesgojug2(1)<=riesgojug1(1)) &&
(riesgojug2(2)<=riesgojug1(2)) &&
(riesgojug2(3)<riesgojug1(3))) ||
((riesgojug2(1)<=riesgojug1(1)) &&
(riesgojug2(2)<riesgojug1(2)) &&
(riesgojug2(3)<=riesgojug1(3))) ||
((riesgojug2(1)<riesgojug1(1)) &&
(riesgojug2(2)<=riesgojug1(2)) &&
(riesgojug2(3)<=riesgojug1(3))))&&
(contadortriangulo>0))
matrizcontadorcoherencia(k,i)=
matrizcontadorcoherencia(k,i)+1;
contadortipo5=contadortipo5+1;
end
end
end
end
end
E=[B,D];
end
%Eliminamos los participantes inactivos
matrizcontadorcoherencia2=matrizcontadorcoherencia;
matrizcontadorcoherencia2(V,:)=[];
Y=H;
%Media de asignaciones incoherentes por participante
for i=1:length(Y)
participativo=Y(i);
apuestasparticiactiv=length(find(aulti(:,2)==participativo));
mediacoherenciaporpart(i)=1-sum(matrizcontadorcoherencia2(i,:)^=0)
/apuestasparticiactiv;
end
bar(mediacoherenciaporpart)
%Media de asignaciones incoherentes por jornada
mediaincoherenciaporjorn=1-mean(matrizcontadorcoherencia2~=0);
plot ([1:64],mediaincoherenciaporjorn)
xlim([1 64])

%La estimacion total del numero de apuestas incoherentes
%en el sentido de que haya otra prediccion de otro
%participante con menor riesgo es:
fprintf('%%.4f es la proporcion de las asignaciones coherentes en
el sentido de que no haya otra prediccion de otro participante
con menor riesgo \n',1-sum(sum(matrizcontadorcoherencia2~=0)
/size(aulti,1))
%Test chi-cuadrado proporcion asignaciones
%coherentes tipo 2 segun formacion
grupol=intersect (C(posgrupol),Y);

```

```

grupo2=intersect (C (posgrupo2), Y);
grupo3=intersect (C (posgrupo3), Y);
grupo4=intersect (C (posgrupo4), Y);
grupo5=intersect (C (posgrupo5), Y);
numpartgrupo1=0;
numpartgrupo2=0;
numpartgrupo3=0;
numpartgrupo4=0;
numpartgrupo5=0;
tnombres2=aulti (:, 2);
%Grupo 1
for g=1:length (grupo1)
participante=grupo1 (g);
posparticipante=find (tnombres2==participante);
numpartgrupo1=numpartgrupo1+length (posparticipante);
end
matrizincoherenciagrupo1=matrizcontadorcoherencia (posgrupo1, :);
promediocoherenciagrupo1=1-sum (sum (matrizincoherenciagrupo1~=0))
/numpartgrupo1;
%Grupo 2
for g=1:length (grupo2)
participante=grupo2 (g);
posparticipante=find (tnombres2==participante);
numpartgrupo2=numpartgrupo2+length (posparticipante);
end
matrizincoherenciagrupo2=matrizcontadorcoherencia (posgrupo2, :);
promediocoherenciagrupo2=1-sum (sum (matrizincoherenciagrupo2~=0))
/numpartgrupo2;
%Grupo 3
for g=1:length (grupo3)
participante=grupo3 (g);
posparticipante=find (tnombres2==participante);
numpartgrupo3=numpartgrupo3+length (posparticipante);
end
matrizincoherenciagrupo3=matrizcontadorcoherencia (posgrupo3, :);
promediocoherenciagrupo3=1-sum (sum (matrizincoherenciagrupo3~=0))
/numpartgrupo3;
%Grupo 4
for g=1:length (grupo4)
participante=grupo4 (g);
posparticipante=find (tnombres2==participante);
numpartgrupo4=numpartgrupo4+length (posparticipante);
end
matrizincoherenciagrupo4=matrizcontadorcoherencia (posgrupo4, :);
promediocoherenciagrupo4=1-sum (sum (matrizincoherenciagrupo4~=0))
/numpartgrupo4;
%Grupo 5
for g=1:length (grupo5)
participante=grupo5 (g);
posparticipante=find (tnombres2==participante);
numpartgrupo5=numpartgrupo5+length (posparticipante);
end
matrizincoherenciagrupo5=matrizcontadorcoherencia (posgrupo5, :);
promediocoherenciagrupo5=1-sum (sum (matrizincoherenciagrupo5~=0))

```

```

    /numpartgrupo5;
c=bar([promediocoherenciagrupo1,promediocoherenciagrupo2,
    promediocoherenciagrupo3,promediocoherenciagrupo4,
    promediocoherenciagrupo5],.40,"red")
set(gca,'xticklabel',["Grupo 1","Grupo 2","Grupo 3",
    "Grupo 4","Grupo 5"])
ylim([0,1])
title('Proporcion de asignaciones coherentes tipo 2')
c.FaceColor=[0.4940 0.1840 0.5560];
%Test de proporciones de asignaciones coherentes
%tipo 2 entre participantes con formacion en Matematicas /
%Doble Grado Matematicas y Fisica y los que tienen
%otro tipo de formacion.
matrizincoherenciaequipoA=matrizcontadorcoherencia(posA,:);
matrizincoherenciaequipoB=matrizcontadorcoherencia(posB,:);
promedioincoherenciaequipoA=sum(sum(matrizincoherenciaequipoA~=0))
    /numpartequipoA;
promedioincoherenciaequipoB=sum(sum(matrizincoherenciaequipoB~=0))
    /numpartequipoB;
c=bar([1-promedioincoherenciaequipoA;
    1-promedioincoherenciaequipoB],.40,"red")
set(gca,'xticklabel',["Matematicas o Doble Grado",
    "Otra formacion"])
ylim([0,1])
title('Proporcion de asignaciones coherentes tipo 2')
c.FaceColor=[0 0.4470 0.7410];
%Test de proporciones de asignaciones coherentes
%tipo 2 entre participantes con distinto genero.
matrizincoherenciaequipoC=matrizcontadorcoherencia(posC,:);
matrizincoherenciaequipoD=matrizcontadorcoherencia(posD,:);
promedioincoherenciaequipoC=sum(sum(matrizincoherenciaequipoC~=0))
    /numpartequipoC;
promedioincoherenciaequipoD=sum(sum(matrizincoherenciaequipoD~=0))
    /numpartequipoD;
c=bar([1-promedioincoherenciaequipoC;
    1-promedioincoherenciaequipoD],.40,"red")
set(gca,'xticklabel',["Hombre","Mujer"])
ylim([0,1])
title('Proporcion de asignaciones coherentes tipo 2')
c.FaceColor=[0.8500 0.3250 0.0980];
%Test de proporciones de asignaciones coherentes tipo 2 entre
%participantes con conocimientos previos en probabilidades
%imprecisas y los que no parten con dichas nociones
matrizincoherenciaequipoE=matrizcontadorcoherencia(posE,:);
matrizincoherenciaequipoF=matrizcontadorcoherencia(posF,:);
promedioincoherenciaequipoE=sum(sum(matrizincoherenciaequipoE~=0))
    /numpartequipoE;
promedioincoherenciaequipoF=sum(sum(matrizincoherenciaequipoF~=0))
    /numpartequipoF;
c=bar([1-promedioincoherenciaequipoE;
    1-promedioincoherenciaequipoF],.40,"red")
set(gca,'xticklabel',["Expertos imprecisas",
    "No expertos imprecisas"])
ylim([0,1])

```



```

title('Proporcion de asignaciones coherentes tipo 2')
c.FaceColor=[0.9290 0.6940 0.1250];

```

A.3 Programas para el análisis de las asignaciones de las estrategias ficticias y evolución en la posición a lo largo de la competición

```

clear all
B=readtable('MotorWCC.Estrategias.xlsm','sheet','Partidos',
    'Range','A:A');
B=table2array(B);
B=convertCharsToStrings(B);
D=readtable('MotorWCC.Estrategias.xlsm','sheet','Partidos',
    'Range','B:B');
D=table2array(D);
F=readtable('MotorWCC.Estrategias.xlsm','sheet','Partidos',
    'Range','C:C');
F=table2array(F);
C=readtable('MotorWCC.Estrategias.xlsm','sheet','Table1',
    'Range','H:H');
C=table2array(C);
C(1)=[];
C=convertCharsToStrings(C);
C(end-8)=[];
nnombres=size(C,1);
X1=readtable('MotorWCC.Estrategias.xlsm','sheet','Probabilidad1',
    'Range','C:C');
leerX=strcat("M1:M",num2str(height(X1)));
xlswrite('MotorWCC.Estrategias.xlsm',table2array(X1),
    'Probabilidad',leerX);
X=readtable('MotorWCC.Estrategias.xlsm','sheet','Probabilidad',
    'Range',leerX);
X=table2array(X);
X(find((isnan(X))==1))=[];leerA=strcat("P1:", "U",
    num2str(length(X)+1));
A=table2array(readtable('MotorWCC.Estrategias.xlsm','sheet',
    'Probabilidad', 'Range',leerA));
nencuestas=size(A,1);
leera=strcat("O1:V",num2str(length(X)+1));
a=readtable('MotorWCC.Estrategias.xlsm','sheet','Probabilidad',
    'Range',leera);
a(:,2:7)=[];
a=table2array(a);usunoreg=find(a(:,2)=="SINUSUARIO");
a(usunoreg,:)=[];
A(usunoreg,:)=[];
X(usunoreg)=[];
E=[B,D,F];
puntuacion=zeros(nnombres,1);
puestospart=zeros(nnombres,23);
    j=1;
    coinwhile=[];

```

```

aulti=[];
Aulti=[];
Xulti=[];
while (j<=size(a,1))
    if isempty(find(coinwhile==j))==1
        nombre=convertCharsToStrings(char(a(j,2)));
        partido=convertCharsToStrings(char(a(j,1)));
        L=max(X(find(a(:,2)==nombre & a(:,1)==partido)));
        M=find(X==L);
        if length(M)==1
            aulti=[aulti;a(M,:)];
            Aulti=[Aulti;A(M,:)];
            Xulti=[Xulti;X(M,:)];
            coinwhile=[coinwhile,(find(a(:,2)==nombre &
                a(:,1)==partido))'];
        else
            horavariosnombres=a(M,2)==nombre;
            Mayuda=find(horavariosnombres==1);
            aulti=[aulti;a(M(Mayuda),:)]];
            Aulti=[Aulti;A(M(Mayuda),:)]];
            Xulti=[Xulti;X(M(Mayuda),:)]];
            coinwhile=[coinwhile,(find(a(:,2)==nombre &
                a(:,1)==partido))'];
        end
    end
    j=j+1;
end
for i=1:23
    E(find(E(:,3)~=num2str(i)),:)=[];
    npartidos=size(E,1);
    table=[];
    aresult=[];
    Aresult=[];
    for h=1:npartidos
        U=find(aulti(:,1)==E(h,1));
        aresult=[aresult;aulti(U,:)];
        Aresult=[Aresult;Aulti(U,:)];
        U=[];
    end
    tnombres=areult(:,2);
    for k=1:nnombres
        jugadoract=C(k);
        U=find(tnombres==jugadoract);
        if length(U)>0
            for h=1:length(U)
                partidoact=convertCharsToStrings(char(areult(U(h),1)));
                probabilidadactiva=Aresult(U(h),:);
                V=find(areult(:,1)==partidoact);
                if length(V)>0
                    for p=1:length(V)
                        if (jugadoract~=tnombres(V(p)) &&
                            k<find(C==char(tnombres(V(p))))))
                            jugadorcomp=char(tnombres(V(p)));
                            probabilidadcomp=Aresult(V(p),:);
                        end
                    end
                end
            end
        end
    end
end

```

```

table=[table;
        calculoapuestas(
            probabilidadactiva,
            probabilidadcomp, jugadoract,
            jugadorcomp, partidoact,
            E(E==partidoact,2) )];
    end
end
end
end
end
if size(table,1)==0
else
    for w=1:length(C)
        Q=find(table(:,1)==C(w));
        if length(Q)>0
            if i<=13
                puntuacion(w)=puntuacion(w)+
                    sum(str2double(table(Q,5)));
            elseif i>13 && i<=22
                puntuacion(w)=puntuacion(w)+
                    2*sum(str2double(table(Q,5)));
            else
                puntuacion(w)=puntuacion(w)+
                    3*sum(str2double(table(Q,5)));
            end
        end
        R=find(table(:,2)==C(w));
        if length(R)>0
            if i<=13
                puntuacion(w)=puntuacion(w)+
                    sum(str2double(table(R,6)));
            elseif i>13 && i<=22
                puntuacion(w)=puntuacion(w)+
                    2*sum(str2double(table(R,6)));
            else
                puntuacion(w)=puntuacion(w)+
                    3*sum(str2double(table(R,6)));
            end
        end
    end
end
puntuacion=round(puntuacion.*10000000)./10000000;
y=sort(puntuacion, 'descend');
y=round(y.*10000000)./10000000;
contador=1;
while contador<=nnombres
    k=find(puntuacion==y(contador));
    if length(k)==1
        ordenpart=[ordenpart;C(k)];
        puestospart(k,i)=contador;
        contador=contador+1;
    else

```

```

ordenpart=[ordenpart;C(k)];
puestospart(k,i)=contador:(contador+length(k)-1);
contador=contador+length(k);
end
end
E=[B,D,F];
end
clasificacion=[ordenpart,y];
vectorjornada=1:23;
vectorposEstrategia1=puestospart(find(C=="Estrategia1"),:);
%Estrategia1: asignar (1,0,0) a lo mas probable
vectorposEstrategia2=puestospart(find(C=="Estrategia2"),:);
%Estrategia2: asignar (1,0,0) a lo menos probable
vectorposEstrategia3=puestospart(find(C=="Estrategia3"),:);
%Estrategia3: estimar (p1,p2,p3) segun las cuotas de las
%casas de apuestas
vectorposEstrategia4=puestospart(find(C=="Estrategia4"),:);
%Estrategia4: asignar 1 a lo mas probable y no apostar en
%las otras dos asignaciones
vectorposEstrategia5=puestospart(find(C=="Estrategia5"),:);
%Estrategia5: asignar (0,0,0) en todos los partidos
vectorposEstrategia6=puestospart(find(C=="Estrategia6"),:);
%Estrategia6: Modelo de contaminacion
vectorposEstrategia7=puestospart(find(C=="Estrategia7"),:);
%Estrategia7: Modelo Pari-mutuel
vectorposEstrategia8=puestospart(find(C=="Estrategia8"),:);
%Estrategia8: Variacion Total
plot(vectorjornada,-vectorposEstrategia1)
title('Evolucion de los participantes')
xlabel('Jornadas')
ylabel('Posicion')
xlim([1 23])
ylim([-58 -1])
yticks([-58 -55 -50 -45 -40 -35 -30 -25 -20 -15 -10 -5 -1])
yticklabels({'58','55','50','45','40','35','30','25','20',
'15','10','5','1'})
hold on
plot(vectorjornada,-vectorposEstrategia2)
plot(vectorjornada,-vectorposEstrategia4)
plot(vectorjornada,-vectorposEstrategia5)
legend('Estrategia1','Estrategia2','Estrategia4',
'Estrategia5','Location','northoutside')

hold off

plot(vectorjornada,-vectorposEstrategia3)
title('Evolucion de los participantes')
xlabel('Jornadas')
ylabel('Posicion')
xlim([1 23])
ylim([-58 -1])
yticks([-58 -55 -50 -45 -40 -35 -30 -25 -20 -15 -10 -5 -1])
yticklabels({'58','55','50','45','40','35','30','25','20',
'15','10','5','1'})

```

```

hold on
plot(vectorjornada,-vectorposEstrategia6)
plot(vectorjornada,-vectorposEstrategia7)
plot(vectorjornada,-vectorposEstrategia8)
legend('Estrategia3','Estrategia6','Estrategia7',
'Estrategia8','Location','northoutside')
%Este caso seria para el parametro de dispersion
%epsilon=0.1, para el caso epsilon=0.0001 habria
%que modificar el parametro desde la base de
%datos del Excel y volver a ejecutar el codigo
%desde el comienzo de la seccion A.3 del Apendice.
PosEstr=zeros(8,1);
for i=1:8
PosEstr(i)=find(clasificacion(:,1)==strcat("Estrategia",
num2str(i)));
end
xlswrite('MotorWCC_Estrategias.xlsm',PosEstr,
'Clasificacion','O3');
xlswrite('MotorWCC_Estrategias.xlsm',y(PosEstr),
'Clasificacion','P3');
xlswrite('MotorWCC_Estrategias.xlsm',puntuacion,
'Clasificacion','L2');
xlswrite('MotorWCC_Estrategias.xlsm',ordenpart,
'Clasificacion','E3');
xlswrite('MotorWCC_Estrategias.xlsm',y,
'Clasificacion','F3');

```

B.1 Programas para la representación de los diagramas de sectores

```

#Instalamos paquete readxl
install.packages("readxl")

#Cargamos paquete readxl
library(readxl)

#Buscar ruta archivo Excel
file.choose()

#Copiar ruta de la consola
ruta_excel<-"C:\\Users\\Usuario\\Desktop\\
World Cup Competition\\World-Cup-Competition\\
MotorWCC_RStudio.xlsm"

#Introducimos los datos
tabla<-read_excel(ruta_excel, sheet="Table1",
range="D2:H52")

#Para agrupar por genero

```

```

vectgenero<-c(sum(tabla$Genero=="Hombre / Male"),
             sum(tabla$Genero=="Mujer / Female"))
             /length(tabla$Genero)
etiqueta<-paste(vectgenero*100,"%",sep = " ")
pie(vectgenero,labels=etiqueta,col=c("red","blue"),
    main="Comparativa entre el genero de
        los participantes")
legend("topright",c("Genero masculino",
                    "Genero femenino"), cex=1.2,fill=c("red","blue"))

#Para agrupar por edad
vectedad<-c(sum(tabla$Edad<25),sum(tabla$Edad>=25 &
    tabla$Edad<45),sum(tabla$Edad>=45))
    /length(tabla$Edad)
etiqueta<-paste(vectedad*100,"%",sep = " ")
pie(vectedad,labels=etiqueta,col=c("red","blue",
    "yellow"), main="Comparativa entre la edad
    de los participantes")
legend("topright",c("Menor de 25 a~nos",
                    "Entre 25 y 44 a~nos", "Mayor de 44 a~nos"),
    cex=1,fill=c("red","blue","yellow"))

#Para agrupar por formacion
vectestudios<-c(sum(tabla$Estudios=="Matematicas")+
    sum(tabla$Estudios=="Matematicas y Fisica")+
    sum(tabla$Estudios=="Fisica"),
    sum(tabla$Estudios=="Ingenieria"),
    sum(tabla$Estudios=="ESO")+
    sum(tabla$Estudios=="Bachillerato")+
    sum(tabla$Estudios=="Grado Superior"),
    sum(tabla$Estudios=="Economia")+
    sum(tabla$Estudios=="Derecho")+
    sum(tabla$Estudios=="ADE"),
    sum(tabla$Estudios=="Biologia")+
    sum(tabla$Estudios=="Biotecnologia")+
    sum(tabla$Estudios=="Enfermeria")+
    sum(tabla$Estudios=="Quimica"))
    /length(tabla$Estudios)
etiqueta<-paste(vectestudios*100,"%",sep = " ")
pie(vectestudios,labels=etiqueta,col=c("red","blue",
    "yellow","green","purple"), main="Comparativa
    entre la formacion de los participantes")
legend("topleft",c("Matematicas y/o Fisica",
                    "Ingenieria",
                    "ESO, Bachillerato o Grado Superior",
                    "Ciencias Sociales y Juridicas",
                    "Ciencias Biosanitarias"),
    cex=0.63,fill=c("red","blue","yellow",
                    "green","purple"))

```

```

#Para agrupar por conocimientos previos en imprecisas
vectconocimprecis<-c(
  sum(tabla$`Nombre usuario`=="Walley4Ever")+
  sum(tabla$`Nombre usuario`=="Arthur Van Camp")+
  sum(tabla$`Nombre usuario`=="Kalmanhorof")+
  sum(tabla$`Nombre usuario`=="Keano")+
  sum(tabla$`Nombre usuario`=="YSale")+
  sum(tabla$`Nombre usuario`=="MOJI")+
  sum(tabla$`Nombre usuario`=="NachoM"),
  50-(sum(tabla$`Nombre usuario`=="Walley4Ever")+
  sum(tabla$`Nombre usuario`=="Arthur Van Camp")+
  sum(tabla$`Nombre usuario`=="Kalmanhorof")+
  sum(tabla$`Nombre usuario`=="Keano")+
  sum(tabla$`Nombre usuario`=="ESO")+
  sum(tabla$`Nombre usuario`=="YSale")+
  sum(tabla$`Nombre usuario`=="MOJI")+
  sum(tabla$`Nombre usuario`=="NachoM")))
/length(tabla$`Nombre usuario`)
etiqueta<-paste(vectconocimprecis*100,"%",
  sep = " ")
pie(vectconocimprecis,labels=etiqueta,
  col=c("red","blue"),
  main="Comparativa entre expertos e inexpertos
  en probabilidades imprecisas")
legend("topleft",c("Expertos en imprecisas",
  "Inexpertos en imprecisas"),cex=0.63,
  fill=c("red","blue"))

```

B.2 Análisis de la existencia de diferencias significativas por grupos en cuanto a la consistencia

```
#Evitar perdida segura
```

```

#Para ver si existen diferencias significativas segun la
#formacion de los participantes (entre los formados en
#Matematicas/Doble Grado y los que tienen otra
#formacion) a la hora de evitar la perdida segura
#en las asignaciones.
numpartequipoA<-946
numpartequipoB<-1191
contadorevitarperdidaseguraequipoA<-761
contadorevitarperdidaseguraequipoB<-917
matrizevitarperdidasegurafomacion<-matrix(c(
  contadorevitarperdidaseguraequipoA,

```

```

    numparte equipoA - contadorevitarperdidaseguraequipoA ,
    contadorevitarperdidaseguraequipoB ,
    numparte equipoB - contadorevitarperdidaseguraequipoB),
    ncol=2,byrow = T)
prop.test(matrizevitarperdidaseguraformacion ,
    alternative="greater",conf.level = 0.95,
    correct = FALSE)
#Sale pvalor=0.02687, se rechaza la hipotesis nula
#y podemos concluir que existen diferencias
#significativas en la formacion en Matematicas
#y/o Fisica de los participantes a la hora de
#evitar la perdida segura en las asignaciones.

#Para ver si existen diferencias significativas
#entre el genero de los participantes (entre hombres
#y mujeres en este caso) a la hora de evitar
#la perdida segura en las asignaciones.
numparte equipoC<-1767
numparte equipoD<-370
contadorevitarperdidaseguraequipoC<-1450
contadorevitarperdidaseguraequipoD<-228
matrizevitarperdidaseguragenero<-matrix(c(
    contadorevitarperdidaseguraequipoC ,
    numparte equipoC - contadorevitarperdidaseguraequipoC ,
    contadorevitarperdidaseguraequipoD ,
    numparte equipoD - contadorevitarperdidaseguraequipoD),
    ncol=2,byrow = T)
prop.test(matrizevitarperdidaseguragenero ,
    alternative="greater",conf.level = 0.95,
    correct = FALSE)
#Sale pvalor=2.2e-16, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en el genero de los participantes
#a la hora de evitar la perdida segura en
#las asignaciones.

#Para ver si existen diferencias significativas
#entre los conocimientos previos en probabilidades
#imprecisas de los participantes a la hora de
#evitar la perdida segura en las asignaciones.
numparte equipoE<-291
numparte equipoF<-1846
contadorevitarperdidaseguraequipoE<-290
contadorevitarperdidaseguraequipoF<-1388
matrizevitarperdidaseguraconocimprecisas<-matrix(c(
    contadorevitarperdidaseguraequipoE ,
    numparte equipoE - contadorevitarperdidaseguraequipoE ,
    contadorevitarperdidaseguraequipoF ,
    numparte equipoF - contadorevitarperdidaseguraequipoF),

```



```

    ncol=2,byrow = T)
prop.test(matrizevitarperdidaseguraconocimprecisas,
  alternative="greater",conf.level = 0.95)
#Sale pvalor=2.2e-16, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en los conocimientos previos en
#probabilidades imprecisas de los
#participantes a la hora de evitar la perdida
#segura en las asignaciones.

#Coherencia

#Para ver si existen diferencias significativas entre la
#formacion de los participantes (entre los formados en
#Matematicas/Doble Grado y los que tienen otra formacion)
#a la hora de ser coherentes en las asignaciones.
contadorcoherenciaequipoA<-530
contadorcoherenciaequipoB<-598
matrizcoherenciaformacion<-matrix(c(
  contadorcoherenciaequipoA,
  numparteequipoA-contadorcoherenciaequipoA,
  contadorcoherenciaequipoB,
  numparteequipoB-contadorcoherenciaequipoB),
  ncol=2,byrow = T)
prop.test(matrizcoherenciaformacion,
  alternative="greater",conf.level = 0.95,
  correct = FALSE)
#Sale pvalor=0.003739, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en la formacion en Matematicas
#y/o Fisica de los participantes a la hora de ser
#coherentes en las asignaciones.

#Para ver si existen diferencias significativas entre el
#genero de los participantes (entre hombres y mujeres
#en este caso) a la hora de ser coherentes
#en las asignaciones
contadorcoherenciaequipoC<-995
contadorcoherenciaequipoD<-133
matrizcoherenciagenero<-matrix(c(
  contadorcoherenciaequipoC,
  numparteequipoC-contadorcoherenciaequipoC,
  contadorcoherenciaequipoD,
  numparteequipoD-contadorcoherenciaequipoD),
  ncol=2,byrow = T)
prop.test(matrizcoherenciagenero,
  alternative="greater",conf.level = 0.95,

```

```

    correct = FALSE)
#Sale pvalor=4.843e-13, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en el genero de los
#participantes a la hora de ser coherentes en
#las asignaciones.

#Para ver si existen diferencias significativas
# entre los conocimientos previos en probabilidades
#imprecisas de los participantes a la hora de ser
#coherentes en las asignaciones.
contadorcoherenciaequipoE<-251
contadorcoherenciaequipoF<-877
matrizcoherenciaconocimprecisas<-matrix(c(
    contadorcoherenciaequipoE,
    numparteequipoE-contadorcoherenciaequipoE,
    contadorcoherenciaequipoF,
    numparteequipoF-contadorcoherenciaequipoF),
    ncol=2,byrow = T)
prop.test(matrizcoherenciaconocimprecisas,
    alternative="greater",conf.level = 0.95,
    correct = FALSE)
#Sale pvalor< 2.2e-16, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en los conocimientos previos
#en probabilidades imprecisas de los participantes
#a la hora de ser coherentes en las asignaciones.

#Scoring rules

#Para ver si existen diferencias significativas entre la
#formacion de los participantes (entre los formados en
#Matematicas/Doble Grado y los que tienen otra formacion)
#a la hora de ser incoherente en el sentido de que haya
#otra prediccion de otro participante con menor riesgo.
contadorscoringequipoA<-873
contadorscoringequipoB<-1057
matrizscoringformacion<-matrix(c(
    contadorscoringequipoA,
    numparteequipoA-contadorscoringequipoA,
    contadorscoringequipoB,
    numparteequipoB-contadorscoringequipoB),
    ncol=2,byrow = T)
prop.test(matrizscoringformacion,
    alternative="greater",conf.level = 0.95,
    correct = FALSE)
#Sale pvalor=0.003037, se rechaza la hipotesis

```

```

#nula y podemos concluir que existen diferencias
#significativas en la formacion en Matematicas
#y/o Fisica de los participantes a la hora de ser
#incoherente en el sentido de que haya otra
#prediccion de otro participante con menor riesgo.

#Para ver si existen diferencias significativas
#entre el genero de los participantes a la hora
#de ser incoherente en el sentido de que haya
#otra prediccion de otro participante con
#menor riesgo.
contadorscoringequipoC<-1597
contadorscoringequipoD<-333
matrizscoringgenero<-matrix(c(
  contadorscoringequipoC,
  numpartequipoC-contadorscoringequipoC,
  contadorscoringequipoD,
  numpartequipoD-contadorscoringequipoD),
  ncol=2,byrow = T)
prop.test(matrizscoringgenero,
  alternative="greater",conf.level = 0.95,
  correct = FALSE)
#Sale pvalor=0.4113, no se rechaza la hipotesis
#nula y no podemos concluir que existan diferencias
#significativas en el genero de los participantes
#a la hora de ser incoherente en el sentido de que
#haya otra prediccion de otro participante con
#menor riesgo.

#Para ver si existen diferencias significativas
#entre los conocimientos previos en probabilidades
#imprecisas de los participantes a la hora de ser
#incoherente en el sentido de que haya otra prediccion
#de otro participante con menor riesgo.
contadorscoringequipoE<-286
contadorscoringequipoF<-1644
matrizscoringconocimprecisas<-matrix(c(
  contadorscoringequipoE,
  numpartequipoE-contadorscoringequipoE,
  contadorscoringequipoF,
  numpartequipoF-contadorscoringequipoF),
  ncol=2,byrow = T)
prop.test(matrizscoringconocimprecisas,
  alternative="greater",conf.level = 0.95,
  correct = FALSE)
#Sale pvalor=3.814e-07, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en los conocimientos previos
#en probabilidades imprecisas de los participantes

```

```

#a la hora de ser incoherente en el sentido de que
#haya otra prediccion de otro participante
#con menor riesgo.

#TEST CHI CUADRADO

#Evitar la perdida segura

matrizevitarperdidaseguraformacion<-matrix(c(761,
      946-761,542, 706-542,133,181-133,85,102-85,157,
      202-157) ,nrow=5,byrow=T)
chisq.test(matrizevitarperdidaseguraformacion)
#Sale pvalor=0.1091, no se rechaza la hipotesis nula y no
#podemos concluir que existan diferencias significativas
#en la formacion de los participantes a la hora de
#evitar la perdida segura en las asignaciones.

#Coherencia

matrizcoherenciagruposformacion<-matrix(c(530,
      946-530,375, 706-375,71,181-71,62,102-62,
      90,202-90),nrow = 5, byrow = T)
chisq.test(matrizcoherenciagruposformacion)
#Sale pvalor=4.034e-05, se rechaza la hipotesis
#nula y podemos concluir que existen diferencias
#significativas en la formacion de los participantes
#a la hora de ser coherentes en las asignaciones.

#Coherencia tipo 2 segun scoring rules

matrizcoherencia2gruposformacion<-matrix(c(946-73,73,
      706-88,88,181-10,10,102-15,15,202-21,21),nrow = 5,
      byrow = T)
chisq.test(matrizcoherencia2gruposformacion)
#Sale pvalor=0.001881, se rechaza la hipotesis nula y
#podemos concluir que existen diferencias significativas
#en la formacion de los participantes a la hora
#de ser incoherente en el sentido de que haya
#otra prediccion de otro participante con menor riesgo.

```