Introducción a solaR

Oscar Perpiñán Lamigueiro

27 de Septiembre de 2010

1. Introducción

El paquete solaR incluye un conjunto de funciones destinadas al cálculo de la radiación solar incidente en sistemas fotovoltaicos y a la simulación del funcionamiento de diferentes aplicaciones de esta tecnología energética. En la versión actual de este paquete se incluyen funciones que permiten realizar todas las etapas de cálculo desde la radiación global en el plano horizontal hasta la productividad final de sistemas fotovoltaicos de conexión a red y de bombeo. También se incluye una herramienta de análisis estadístico del funcionamiento de plantas fotovoltaicas compuestas por varias unidades.

Esta versión incluye numerosos cambios respecto a la versión anterior. Aunque se ha intentado preservar el uso externo de las funciones, es innegable que los cambios son visibles. Debe destacarse:

1. La mayor parte del paquete está basado en clases y métodos S4.
2. Todas las series temporales están construidas con la clase zoo [8].
3. La mayor parte de las funciones y sus argumentos han sido renombrados con palabras inglesas para facilitar su comprensión internacional.

2. Geometría solar

Las ecuaciones que determinan el movimiento aparente solar están incluidas en las funciones fSolD y fSolI. La primera de ellas realiza el cálculo de los parámetros que pueden considerarse constantes en un día (por ejemplo, la declinación) y la segunda incluye el conjunto de valores que evolucionan a lo largo del día (por ejemplo, los ángulos cenital y azimutal). Estas funciones también proporcionan el valor de la radiación solar disponible fuera de la atmósfera: fSolD entrega valores de irradiación mientras que fSolI ofrece la evolución de la irradiancia.

Es posible realizar los cálculos para un día concreto:

```
> BTd = fBTd(mode = "serie")
> lat = 37.2
> SolD <- fSolD(lat, BTd[100])
> SolI <- fSolI(SolD, sample = "hour", keep.night = FALSE)
> head(SolI)

                        w aman cosThzS     AlS     AzS    Bo0      rd
2010-04-10 06:00:00 -1.5708    1 0.07927 0.07935 -1.6758  107.8 0.01130
2010-04-10 07:00:00 -1.3090    1 0.28365 0.28760 -1.5179  385.8 0.04044
2010-04-10 08:00:00 -1.0472    1 0.47410 0.49394 -1.3472  644.9 0.06759
2010-04-10 09:00:00 -0.7854    1 0.63764 0.69143 -1.1433  867.3 0.09091
2010-04-10 10:00:00 -0.5236    1 0.76313 0.86814 -0.8742 1038.0 0.10880
2010-04-10 11:00:00 -0.2618    1 0.84202 1.00101 -0.4957 1145.3 0.12005
                       rg
2010-04-10 06:00:00 0.007935
2010-04-10 07:00:00 0.032395
2010-04-10 08:00:00 0.060379
2010-04-10 09:00:00 0.088405
2010-04-10 10:00:00 0.112414
2010-04-10 11:00:00 0.128619
```

o para un conjunto de días:

```
> SolD <- fSolD(lat, BTd[c(10, 50, 100)])
> print(SolD)

             decl    eo     ws Bo0d       EoT
2010-01-10 -0.3847 1.033 -1.258 4497 -0.035464
2010-02-19 -0.2082 1.022 -1.410 6327 -0.059933
2010-04-10  0.1315 0.995 -1.671 9541 -0.004637
attr(,"lat")
[1] 37.2
```

Mediante la función `fBTd` es posible crear bases temporales con diferente estructura. Así, para realizar el cálculo en los denominados días promedio empleamos el siguiente código:

```
> lat = 37.2
> SolD <- fSolD(lat, BTd = fBTd(mode = "prom"))
> SolI <- fSolI(SolD, sample = "10 min", keep.night = FALSE)
```

y podemos obtener la figura 1.
También podemos extender los cálculos a todos los días de un año:

```
> BTd = fBTd(mode = "serie")
> solD <- fSolD(lat, BTd)
> summary(solD)

     Index                 decl                  eo              ws
 Min.   :2010-01-01   Min.   :-4.09e-01   Min.   :0.967   Min.   :-1.91
 1st Qu.:2010-04-02   1st Qu.:-2.89e-01   1st Qu.:0.977   1st Qu.:-1.80
 Median :2010-07-02   Median : 2.63e-16   Median :1.000   Median :-1.57
 Mean   :2010-07-02   Mean   : 9.31e-18   Mean   :1.000   Mean   :-1.57
 3rd Qu.:2010-10-01   3rd Qu.: 2.89e-01   3rd Qu.:1.023   3rd Qu.:-1.34
 Max.   :2010-12-31   Max.   : 4.09e-01   Max.   :1.033   Max.   :-1.24
      B0d             EoT
 Min.   : 4235   Min.   :-6.18e-02
 1st Qu.: 5472   1st Qu.:-2.59e-02
 Median : 8302   Median :-2.48e-03
 Mean   : 8116   Mean   : 1.24e-05
 3rd Qu.:10742   3rd Qu.: 2.16e-02
 Max.   :11607   Max.   : 7.09e-02
```

para obtener la figura 2.
Estas dos funciones han sido agrupadas en una nueva función llamada `calcSol`. Esta función construye un objeto de clase `Sol` que contiene en sus slots los objetos `zoo` creados por `fSolD` y `fSolI`. Esta clase cuenta con métodos para acceder a esta información (por ejemplo, `as.zooD, as.zooI`) y para visualizarla.

3. Radiación solar

Es de uso común disponer de valores de radiación global en el plano horizontal, ya sea en forma de promedios mensuales o de una serie más o menos completa de valores diarios a lo largo de uno o varios años. Para estudiar el funcionamiento de un sistema fotovoltaico es necesario transformar esta información a valores de irradiancia global, difusa y directa en el plano horizontal para transformarlos posteriormente a valores en el plano de la superficie del generador.

3.1. Irradiación e irradiancia en el plano horizontal

La extracción de las componentes de la irradiación difusa y directa se realiza con la función `fCompD`. Esta función necesita los resultados obtenidos con `fSolD`, la radiación global en el plano horizontal (en $^{Wh}/_{m^2}$) y la correlacción entre la fracción de difusa y el índice de claridad. En la versión actual de `solaR`, esta función incorpora las correlacciones de Collares Pereira y Rabl [**?**], y la de Page [4]. Además, permite al usuario elaborar su propia correlacción y entregarla a través del argumento `f`. Repitamos de nuevo los cálculos para un día concreto:

```
> BTd = fBTd(mode = "serie")
> SolD <- fSolD(lat, BTd[100])
> SolI <- fSolI(SolD, sample = "hour")
> G0d = zoo(5000, index(SolD))
> fCompD(SolD, G0d, corr = "Page")

               Fd    Ktd  G0d  D0d  B0d
2010-04-10 0.4078 0.5241 5000 2039 2961

> fCompD(SolD, G0d, corr = "CPR")

               Fd    Ktd  G0d  D0d  B0d
2010-04-10 0.5582 0.5241 5000 2791 2209
```

y para los días promedio:

```
> mon = month.abb
> p <- xyplot(AlS * 180/pi ~ AzS * 180/pi, groups = month, data = SolI,
+     type = "l", col = "black", xlab = expression(psi[s]), ylab = expression(gamma[s]))
> plab = p + glayer(panel.text(0, y[x == 0], mon[group.value],
+     pos = 4, cex = 0.8))
> print(plab)
```

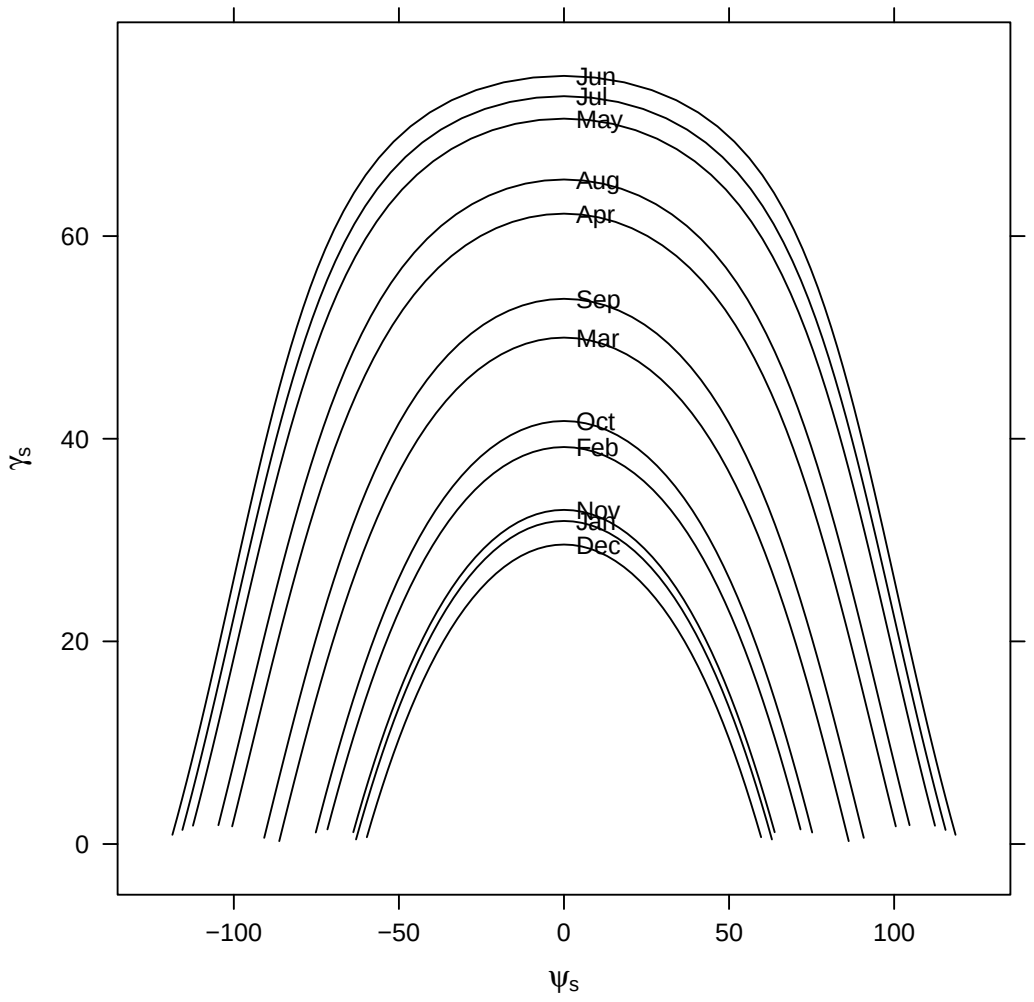

Figura 1: Azimut y altura solar en los doce días promedio

```
> p <- xyplot(solD$decl)
> print(p)
```

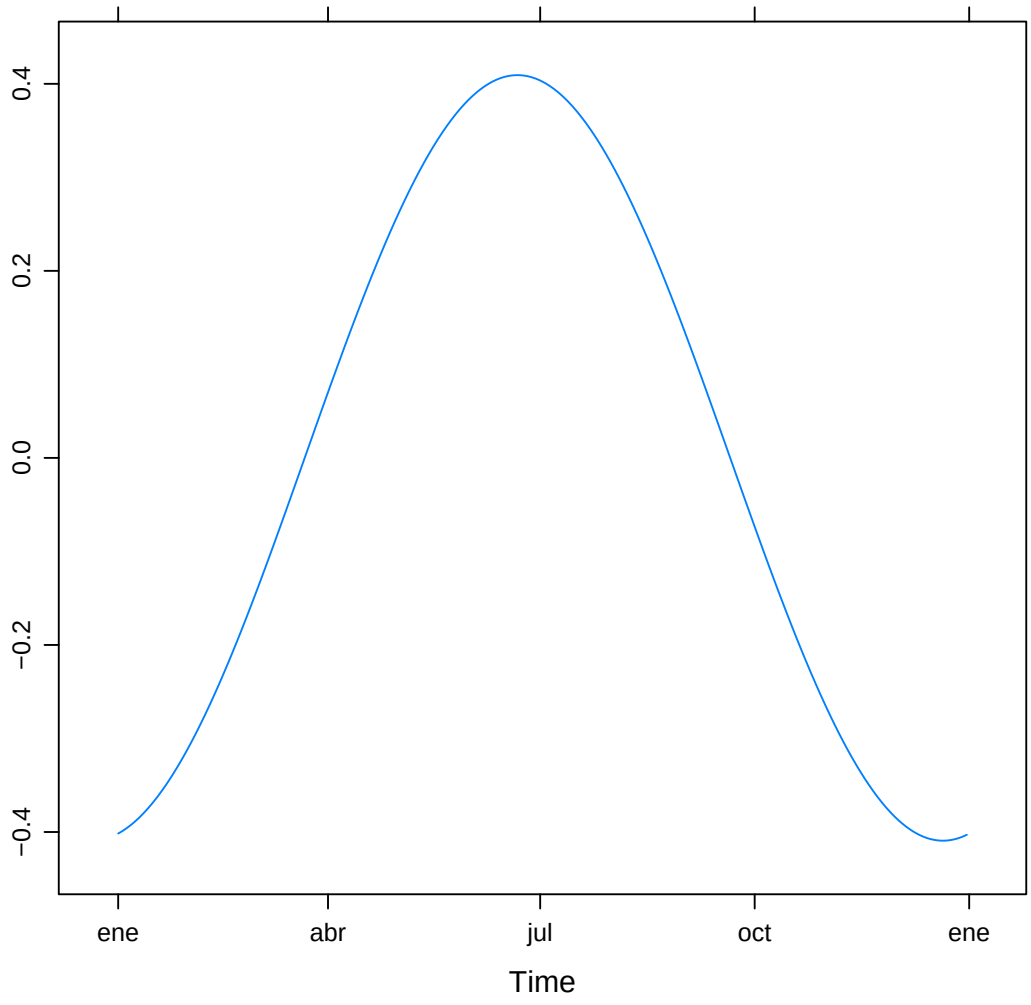

Figura 2: Declinación a lo largo del año

```
> lat = 37.2
> G0dm = c(2.766, 3.491, 4.494, 5.912, 6.989, 7.742, 7.919, 7.027,
+     5.369, 3.562, 2.814, 2.179) * 1000
> Rad = readG0dm(G0dm, lat)
> solD <- fSolD(lat, fBTd(mode = "prom"))
> fCompD(solD, Rad, corr = "Page")

               Fd    Ktd  G0d    D0d  B0d
2010-01-17 0.3354 0.5882 2766  927.6 1838
2010-02-14 0.3452 0.5794 3491 1205.2 2286
2010-03-15 0.3573 0.5687 4494 1605.9 2888
2010-04-15 0.3195 0.6022 5912 1888.9 4023
2010-05-15 0.2871 0.6309 6989 2006.5 4982
2010-06-10 0.2437 0.6693 7742 1886.8 5855
2010-07-18 0.2070 0.7018 7919 1639.0 6280
2010-08-18 0.2209 0.6894 7027 1552.4 5475
2010-09-18 0.2804 0.6368 5369 1505.6 3863
2010-10-19 0.3728 0.5550 3562 1328.1 2234
2010-11-18 0.3475 0.5775 2814  977.8 1836
2010-12-13 0.4233 0.5104 2179  922.3 1257
```

Definamos ahora una función que entrega el resultado de la correlacción de Page. Evidentemente, obtenemos el mismo resultado que con `corr='Page'`.

```
> fKTd = function(x) {
+     (0.99 * (x <= 0.17)) + (x > 0.17) * (1.188 - 2.272 * x +
+         9.473 * x^2 - 21.856 * x^3 + 14.648 * x^4)
+ }
> fCompD(SolD, G0d, corr = "user", f = fKTd)

               Fd    Ktd  G0d  D0d  B0d
2010-04-10 0.5582 0.5241 5000 2791 2209
```

La construcción del perfil diario de irradiancias se realiza con la función `fCompI`. Esta función necesita la información proporcionada por `fCompD` y por `fSolI`. Por ejemplo, para los días promedio obtenemos:

```
> lat = 37.2
> sol <- calcSol(lat, fBTd(mode = "prom"), sample = "hour", keep.night = FALSE)
> G0dm = c(2.766, 3.491, 4.494, 5.912, 6.989, 7.742, 7.919, 7.027,
+     5.369, 3.562, 2.814, 2.179) * 1000
> Ta = c(10, 14.1, 15.6, 17.2, 19.3, 21.2, 28.4, 29.9, 24.3, 18.2,
+     17.2, 15.2)
> BD <- readG0dm(G0dm = G0dm, Ta = Ta, lat = 37.2)
> compD <- fCompD(sol, BD, corr = "Page")
> compI <- fCompI(compD, sol)
> summary(compI)

     Index                          kt                G0
 Min.   :2010-01-17 08:00:00   Min.   :0.401   Min.   :  6.19
 1st Qu.:2010-04-15 10:15:00   1st Qu.:0.507   1st Qu.:187.39
 Median :2010-06-10 18:30:00   Median :0.587   Median :419.26
 Mean   :2010-06-29 18:25:21   Mean   :0.581   Mean   :424.39
 3rd Qu.:2010-09-18 11:45:00   3rd Qu.:0.646   3rd Qu.:624.50
 Max.   :2010-12-13 16:00:00   Max.   :0.765   Max.   :972.56
       D0               B0
 Min.   :  2.59   Min.   :  3.59
 1st Qu.: 78.42   1st Qu.:116.48
 Median :130.24   Median :265.97
 Mean   :122.86   Mean   :301.53
 3rd Qu.:170.43   3rd Qu.:453.84
 Max.   :230.50   Max.   :787.71
```

resultado que recogemos en la figura 3.

3.1.1. Obtención de medidas meteorológicas

La versión actual de este paquete incorpora una función llamada `readMAPA` capaz de acceder a la información disponible en la página www.mapa.es/siar. En esta página se almacenan las medidas diarias de estaciones agroclimáticas repartidas en la mayor parte de la superficie de España. Esta función necesita el código de la estación y su provincia, y las fechas de inicio y final. Los códigos de las estaciones y provincias disponibles están almacenados en `RedEstaciones`. Por ejemplo, la provincia de Madrid cuenta con las siguientes estaciones:

```
> data(RedEstaciones)
> Madrid <- subset(RedEstaciones, NomProv == "Madrid")
> print(Madrid)

     Provincia Estacion NomProv                    NomEst
P209        28        1  Madrid Center:_Finca_experimental
P210        28        2  Madrid                   Arganda
P211        28        3  Madrid                  Aranjuez
P212        28        4  Madrid       Fuentiduena_de_Tajo
P213        28        5  Madrid     San_Martin_de_la_Vega
P214        28        6  Madrid                  Chinchon
P215        28      102  Madrid           Villa_del_Prado
```

```
> p <- xyplot(G0 + B0 + D0 ~ w | month, data = compI, type = "l",
+     auto.key = list(space = "right"))
> print(p)
```

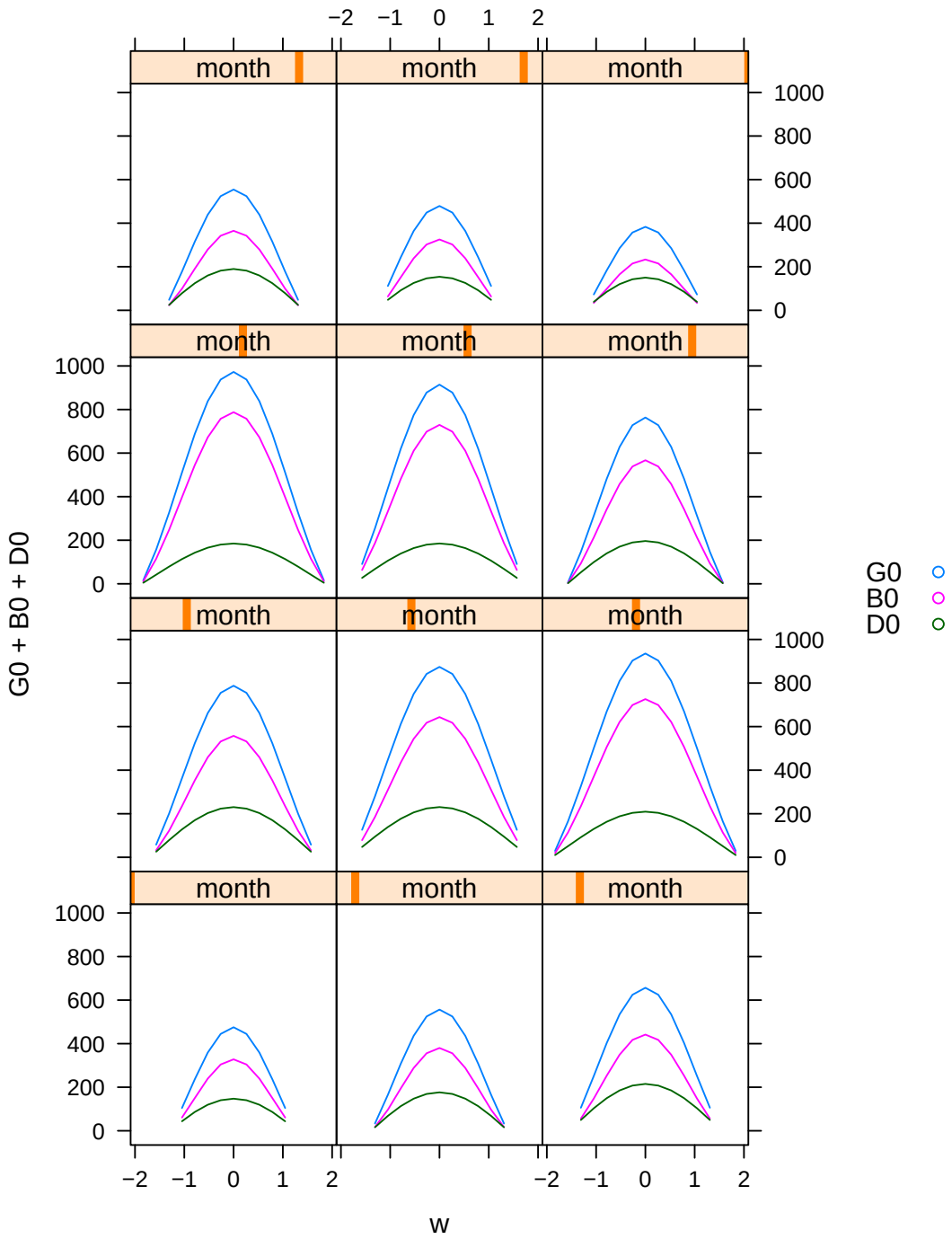

Figura 3: Irradiancia global, difusa y directa en los días promedio

En esta versión de solaR, la función readMAPA crea un objeto de clase Meteo. Para acceder a todos los datos obtenidos se utiliza la función getData. Si sólo se necesita la serie de irradiación, se puede obtener con getG0.

Obtengamos la información del año 2009 disponible en la estación de Aranjuez. Es importante resaltar que las medidas de radiación disponibles tienen unidades de $^{MJ}/_{m^2}$, pero la función readMAPA realiza la conversión a $^{Wh}/_{m^2}$:

```
> Aranjuez <- readMAPA(28, 3, "01/01/2009", "31/12/2009")

Downloading data from www.mapa.es/siar...

> print(Aranjuez)

Object of class  Meteo

Source of meteorological information: mapa-Est: 3 Prov: 28
Latitude of source:  0 degrees

Meteorological Data:
     Index              TempMedia        TempMax      HorMinTempMax
 Min.   :2009-01-01   Min.   :-5.31   Min.   :-2.36   Min.   :   0
 1st Qu.:2009-04-02   1st Qu.: 8.85   1st Qu.:14.92   1st Qu.:1350
 Median :2009-07-02   Median :14.32   Median :23.72   Median :1440
 Mean   :2009-07-02   Mean   :15.33   Mean   :23.35   Mean   :1432
 3rd Qu.:2009-10-01   3rd Qu.:23.67   3rd Qu.:32.61   3rd Qu.:1520
 Max.   :2009-12-31   Max.   :30.68   Max.   :40.76   Max.   :2220

    TempMin       HorMinTempMin   HumedadMedia     HumedadMax     HorMinHumMax
 Min.   :-11.30   Min.   :   0   Min.   : 22.2   Min.   : 49.1   Min.   :   0
 1st Qu.:  2.07   1st Qu.: 440   1st Qu.: 42.4   1st Qu.: 79.1   1st Qu.: 420
 Median :  7.40   Median : 530   Median : 60.3   Median : 92.1   Median : 530
 Mean   :  7.48   Mean   : 711   Mean   : 59.8   Mean   : 96.7   Mean   : 679
 3rd Qu.: 13.26   3rd Qu.: 630   3rd Qu.: 74.7   3rd Qu.: 97.1   3rd Qu.: 640
 Max.   : 21.36   Max.   :2350   Max.   :100.0   Max.   :650.0   Max.   :2350

   HumedadMin      HorMinHumMin    VelViento         DirViento
 Min.   :   0.0   Min.   :   0   Min.   :  0.272   Min.   :  1.12
 1st Qu.:  14.3   1st Qu.:1400   1st Qu.:  0.754   1st Qu.: 43.89
 Median :  26.4   Median :1510   Median :  1.062   Median :108.90
 Mean   :  64.3   Mean   :1414   Mean   :  4.916   Mean   :144.07
 3rd Qu.:  47.8   3rd Qu.:1600   3rd Qu.:  1.778   3rd Qu.:239.80
 Max.   :1640.0   Max.   :2310   Max.   :359.600   Max.   :357.70

  VelVientoMax    DirVientoVelMax  HorMinVelMax  Precipitacion       EtPMon
 Min.   :  1.57   Min.   :   0    Min.   :   0   Min.   : 0.00   Min.   :0.00
 1st Qu.:  4.22   1st Qu.: 193    1st Qu.:1217   1st Qu.: 0.00   1st Qu.:1.38
 Median :  5.82   Median : 250    Median :1358   Median : 0.00   Median :2.88
 Mean   : 10.28   Mean   : 244    Mean   :1330   Mean   : 1.19   Mean   :3.41
 3rd Qu.:  7.66   3rd Qu.: 270    3rd Qu.:1523   3rd Qu.: 0.20   3rd Qu.:5.38
 Max.   :338.20   Max.   :1834    Max.   :2356   Max.   :24.83   Max.   :8.56
                                                                 NA's   :8.00
       G
 Min.   :  77
 1st Qu.:2639
 Median :5147
 Mean   :4845
 3rd Qu.:7169
 Max.   :8753
 NA's   :   8
```

Se pueden representar los datos con un método específico de xyplot:

Esta base de datos incorpora información sobre los valores máximos y mínimos de temperatura. Con esta información la función fTemp genera un perfil intradiario de la temperatura ambiente [2]. Representamos la evolución intradiaria de esta temperatura "sintética" en el mes de marzo en la figura 5.

```
> lat = 41
> sol = calcSol(lat, BTd = indexD(Aranjuez), sample = "hour")
> Temp <- fTemp(sol, Aranjuez)
```

Existen otras dos funciones que también acceden a datos radiación y temperatura y los organizan en un objeto Meteo. Para leer la información de un fichero local o un data.frame se emplean las funciones readBD y df2Meteo. Para construir un objeto Meteo con 12 medias mensuales de radiación (y, opcionalmente, de temperatura ambiente) se utiliza la función readG0dm.

3.1.2. La función calcG0

Todo el proceso de cálculo en el plano horizontal, incluyendo la obtención de medidas con readMAPA y la generación de series de temperatura, está integrado dentro de la función calcG0. Por ejemplo, con las siguientes líneas obtenemos, a partir de las medidas de la estación meteorológica de Aranjuez, las componentes de irradiación e irradiancia en el plano horizontal.

```
> p = xyplot(G ~ TempMedia | month, data = Aranjuez, type = c("p",
+     "r"))
> print(p)
```

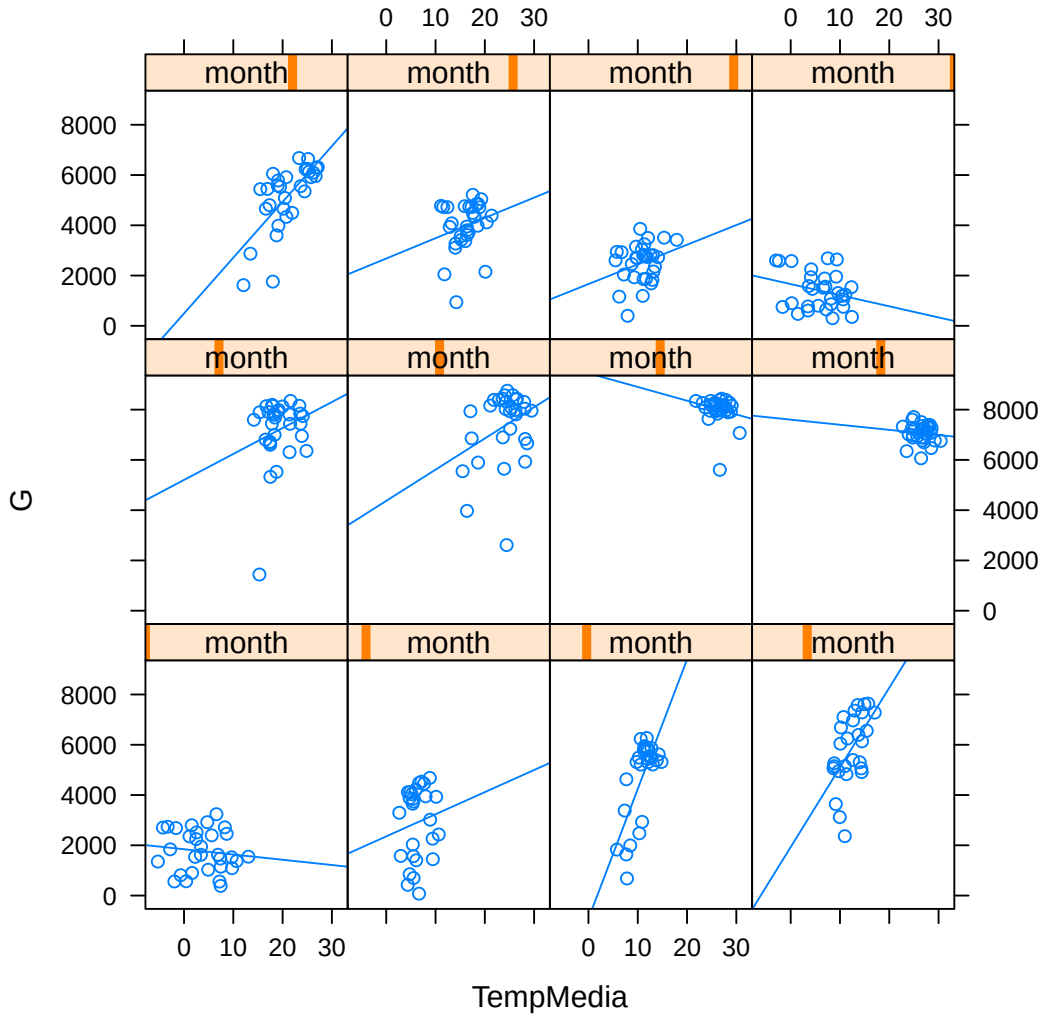

Figura 4: Relación entre irradiación y temperatura media en la estación de Aranjuez.

```
> wTemp = window(Temp, start = as.POSIXct("2009-03-01"), end = as.POSIXct("2009-03-31"))
> p = xyplot(wTemp, col = "black", ylab = "T") + layer_(panel.xblocks(x,
+     DoY, col = c("lightgray", "white")))
> print(p)
```

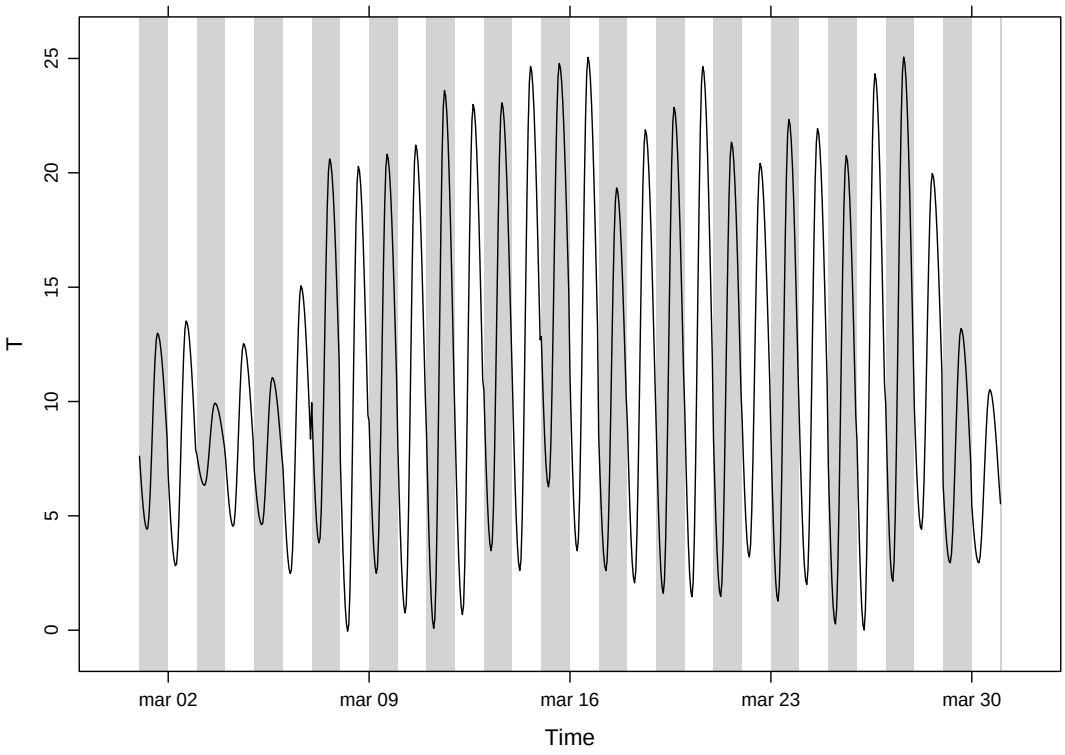

Figura 5: Evolución de la temperatura ambiente en el mes de Marzo.

```
> g0 <- calcG0(lat = 37.2, modeRad = "mapa", mapa = list(prov = 28,
+     est = 3, start = "01/01/2009", end = "31/12/2009"))

Downloading data from www.mapa.es/siar...

> print(g0)

Object of class  G0

Source of meteorological information: mapa-Est: 3 Prov: 28

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
           G0d    D0d    B0d
ene 2009 1.764 1.1461 0.6176
feb 2009 2.916 1.2915 1.6244
mar 2009 4.725 1.5877 3.1371
abr 2009 5.819 2.2890 3.5303
may 2009 7.198 2.2475 4.9510
jun 2009 7.354 2.4525 4.9013
jul 2009 8.002 2.0457 5.9566
ago 2009 7.061 1.9446 5.1160
sep 2009 5.168 1.8897 3.2782
oct 2009 3.993 1.4684 2.5244
nov 2009 2.510 1.3175 1.1920
dic 2009 1.397 0.9773 0.4192

Yearly values:
      G0d   D0d  B0d
2009 1730 614.7 1115
```

3.2. Irradiación y irradiancia en la superficie del generador

Estas componentes de irradiancia en el plano horizontal son transformadas al plano del generador mediante la función `fInclin`. Pero antes se debe realizar el cálculo de la geometría de la superficie del generador, tarea realizada por la función `fTheta`. Esta función necesita el resultado de `calcSol`, y también la información sobre el comportamiento de la superficie (estática, seguimiento a doble eje o con un eje horizontal N-S). En la versión actual de este paquete, esta función permite el cálculo del movimiento mediante *backtracking* sólo para seguidores de eje horizontal [5]. En versiones posteriores se incorporará esta funcionalidad a los seguidores de doble eje.

A partir de los resultados de `fTheta`, la función `fInclin` proporciona las componentes de irradiancia global, difusa y directa en el plano del generador, distinguiendo entre irradiancia incidente e irradiancia efectiva (aquella que incorpora las pérdidas por suciedad e incidencia no perpendicular) [3].

En esta versión, estas dos funciones quedan integradas dentro de la función `calcGef`. Esta función construye un objeto `Gef` que contiene objetos `Meteo`, `Sol` y `G0`. Cuenta con métodos propios para transformar su contenido en objetos `zoo` y `data.frame`. También cuenta con métodos de visualización con `xyplot`, tanto con fórmulas como con el objeto completo como único argumento.

Utilicemos los cálculos previos realizados con `calcG0` para calcular la irradiación e irradiancia en un superficie fija orientada al Sur:

```
> p = xyplot(g0)
> print(p)
```

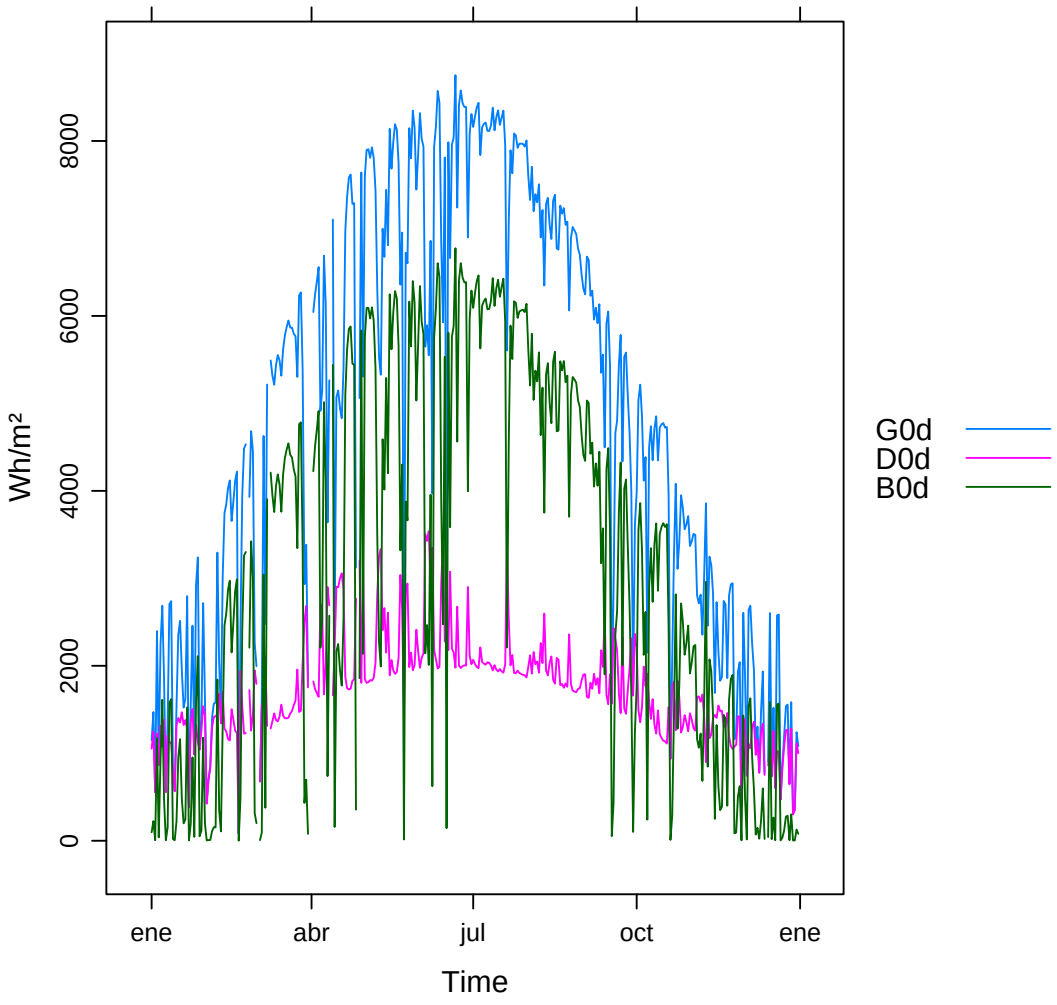

Figura 6: Componentes de irradiación en el plano horizontal obtenidas con la función `calcG0`.

```
> gef <- calcGef(lat = 37.2, modeRad = "prev", prev = g0, beta = 30)
> print(gef)

Object of class  Gef

Source of meteorological information: mapa-Est: 3 Prov: 28

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
            Bod  Gefd   Defd   Befd
ene 2009  8.720 2.202 1.1248 1.0600
feb 2009  9.801 3.590 1.2721 2.2907
mar 2009 10.289 4.933 1.4229 3.4693
abr 2009 10.428 5.194 1.8782 3.2654
may 2009 10.225 6.480 1.9645 4.4461
jun 2009 10.025 6.284 2.0870 4.1265
jul 2009 10.080 6.989 1.7606 5.1518
ago 2009 10.281 6.799 1.7726 4.9591
sep 2009 10.270 5.585 1.8129 3.7228
oct 2009  9.894 5.096 1.5445 3.5135
nov 2009  8.977 3.357 1.3684 1.9641
dic 2009  8.484 1.657 0.9117 0.7328

Yearly values:
      Bod Gefd  Defd Befd
2009 3573 1772 575.6 1180
-----------------
Mode of tracking:  fixed
    Inclination:  30
    Orientation:  0
```

Como ejemplo, en la figura 7 mostramos la relación entre la radiación efectiva y la radiación incidente frente al coseno del ángulo de incidencia para este sistema estático.

En las siguientes líneas calculamos el movimiento de un seguidor de eje horizontal N-S con *backtracking* cuyo máximo ángulo de inclinación es 60° y representamos la evolución del ángulo de inclinación a lo largo del día en la figura 8.

```
> G0dm = c(2766, 3491, 4494, 5912, 6989, 7742, 7919, 7027, 5369,
+     3562, 2814, 2179)
> Ta = c(10, 14.1, 15.6, 17.2, 19.3, 21.2, 28.4, 29.9, 24.3, 18.2,
+     17.2, 15.2)
> prom = list(G0dm = G0dm, Ta = Ta)
> structHoriz = list(L = 4.83)
> distHoriz = data.frame(Lew = structHoriz$L * 4, H = 0)
> gefBT = calcGef(lat = 37.2, prom = prom, sample = "10 min", modeTrk = "horiz",
+     modeShd = "bt", betaLim = 60, distances = distHoriz, struct = structHoriz)
```

4. Productividad de un Sistema Fotovoltaico de Conexión a Red

A partir de los cálculos de radiación descritos la función `fProd` simula el funcionamiento de un Sistema Fotovoltaico de Conexión a Red (SFCR) tomando en consideración ciertos parámetros de su configuración (características del módulo y del inversor, configuración eléctrica del módulo y pérdidas en el sistema). Por ejemplo, las siguientes instrucciones calculan los parámetros principales del sistema (configurado con las opciones por defecto de `fProd`) bajo determinadas condiciones de radiación y temperatura ambiente.

```
> inclin = data.frame(Gef = c(200, 400, 600, 800, 1000), Ta = 25)
> fProd(inclin)

   Gef Ta    Tc   Voc   Isc  Vmpp   Impp   Vdc    Idc   Pac   Pdc   EffI
1  200 25 31.75 673.3 10.34 533.1  9.586 533.1  9.586  4212  4737 0.9164
2  400 25 38.50 655.4 20.68 516.3 19.090 516.3 19.090  8275  9137 0.9334
3  600 25 45.25 637.5 31.02 499.6 28.506 499.6 28.506 11972 13202 0.9346
4  800 25 52.00 619.7 41.36 483.0 37.824 483.0 37.824 15323 16936 0.9325
5 1000 25 58.75 601.8 51.70 466.5 47.037 466.5 47.037 18342 20342 0.9293
```

En primer lugar `fProd` calcula el punto MPP del generador (`Vmpp` e `Impp`) en las condiciones de irradiancia y temperatura marcadas en `Inclin`. A continuación, comprueba que este punto se encuentre dentro de la ventana de búsqueda del MPP del inversor (determinada por los valores `inverter$Vmin` e `inverter$Vmax`). En caso de que el punto MPP calculado se encuentre fuera de estos márgenes, la función asigna el valor límite de la ventana y calcula el valor de corriente correspondiente. En esta situación, la función activa un aviso. En todo caso, la tensión y corriente de entrada al inversor son `Vdc` e `Idc`:

```
> p <- xyplot(Gef/G ~ cosTheta | month, data = gef, type = "smooth")
> print(p)
```

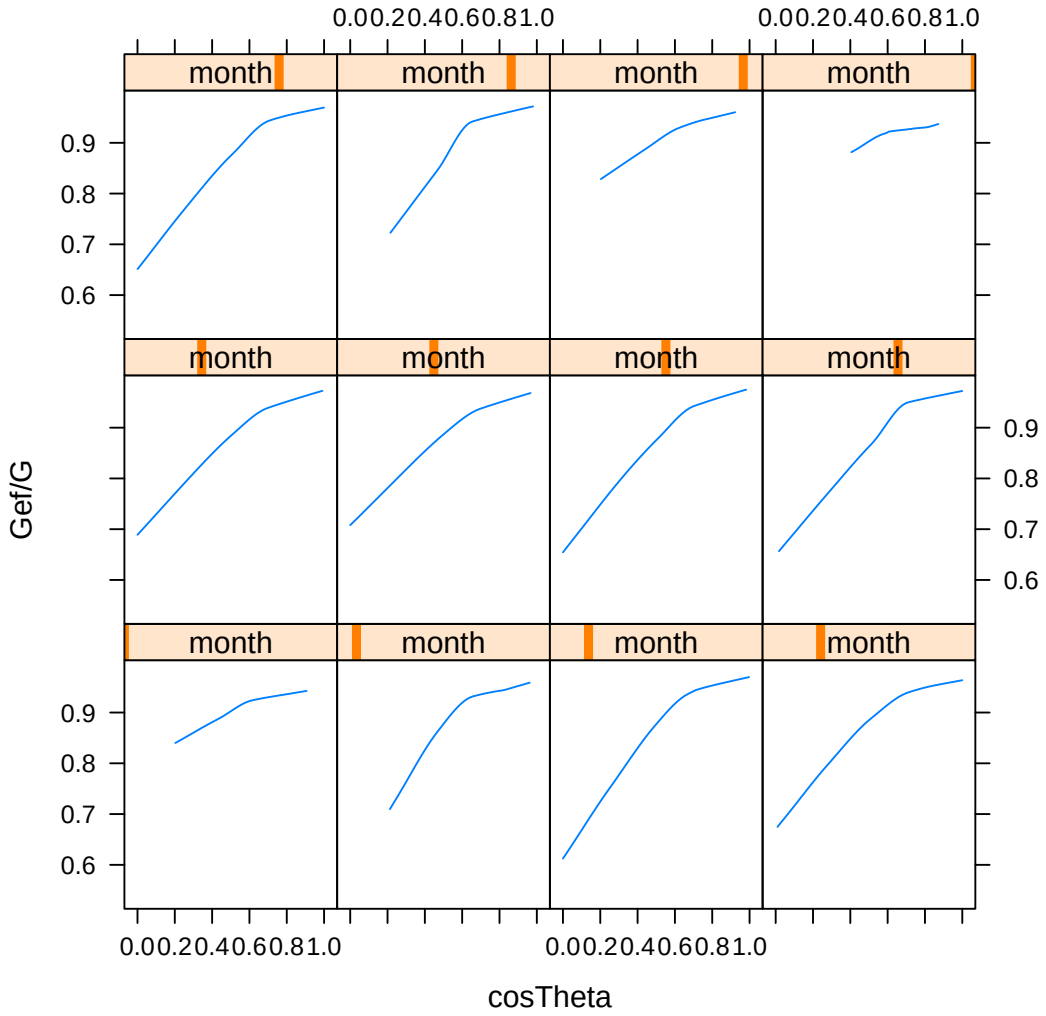

Figura 7: Relación entre la irradiancia efectiva y la incidente frente al coseno del ángulo de incidencia para un sistema estático.

```
> p <- xyplot(r2d(Beta) ~ r2d(w), data = gefBT, type = "l", xlab = expression(omega(degrees)),
+     ylab = expression(beta(degrees)))
> print(p)
```

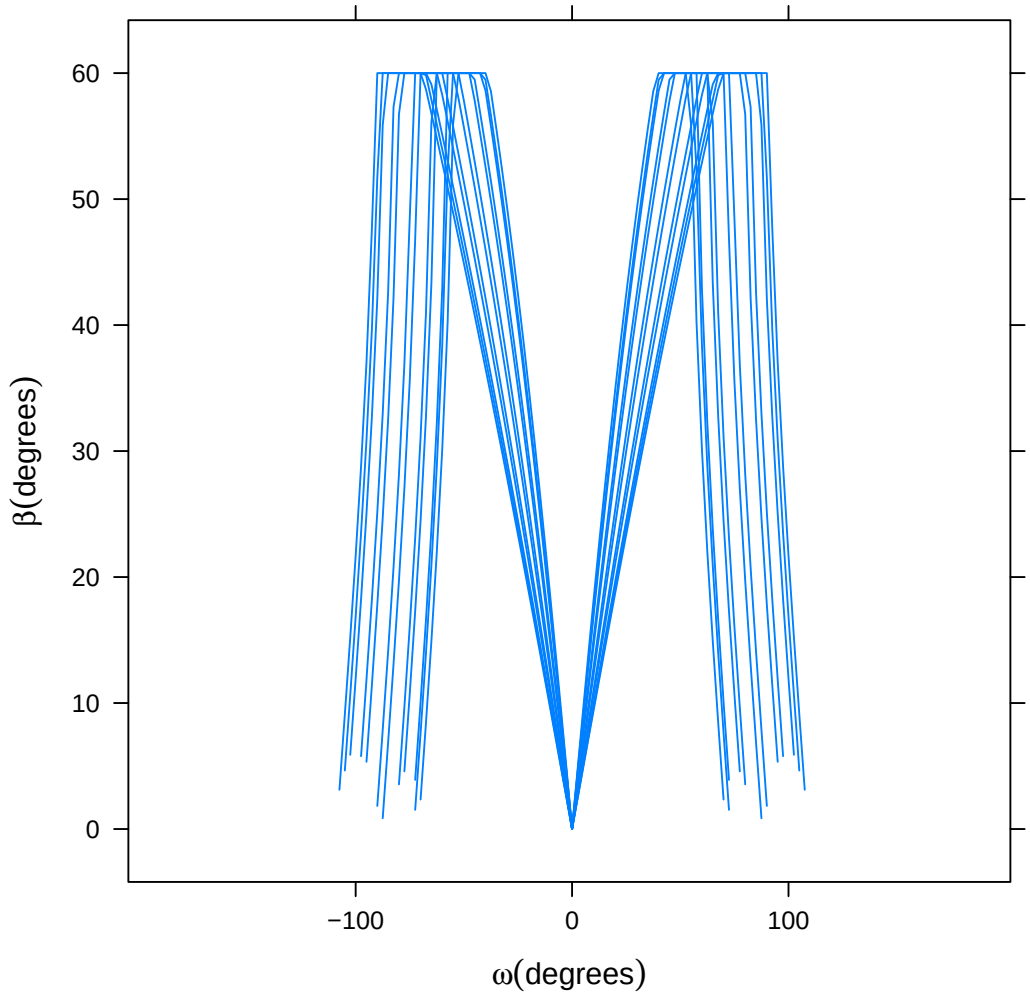

Figura 8: Evolución de la inclinación de un seguidor de eje horizontal con *backtracking* y limitación de ángulo.

```
> inclin = data.frame(Gef = 800, Ta = 30)
> gen1 = list(Nms = 10, Nmp = 11)
> prod = fProd(inclin, generator = gen1)
> print(prod)

  Gef Ta Tc   Voc   Isc  Vmpp  Impp Vdc   Idc   Pac   Pdc   EffI
1 800 30 57 505.3 41.36 392.3 37.68 420 33.83 11943 13169 0.9346
```

En este caso concreto, las pérdidas debidas a la limitación en tensión del inversor son:

```
> with(prod, Vdc * Idc/(Vmpp * Impp))

[1] 0.961
```

La función `prodGCPV` (en la anterior versión su nombre era `prodSFCR`) integra todos los cálculos de radiación (en el plano horizontal y en el plano inclinado) y la simulación del SFCR haciendo uso de todas las funciones anteriormente descritas.

La función `prodGCPV` construye un objeto S4 de clase `ProdGCPV`. Incluye las clases `Meteo`, `Sol`, `G0` y `Gef`. Al igual que las anteriores, cuenta con métodos para ser transformado en un `data.frame` o en un `zoo`, y para ser representado con `xyplot` como objeto completo o mediante fórmulas.

La siguiente secuencia de instrucciones calcula la productividad del mismo SFCR funcionando como estático, doble eje y con eje horizontal. Para el módulo, generador, inversor y pérdidas del sistema se utilizan los valores por defecto que incorpora la función `prodGCPV`.

```
> lat = 37.2
> G0dm = c(2766, 3491, 4494, 5912, 6989, 7742, 7919, 7027, 5369,
+     3562, 2814, 2179)
> Ta = c(10, 14.1, 15.6, 17.2, 19.3, 21.2, 28.4, 29.9, 24.3, 18.2,
+     17.2, 15.2)
> prom = list(G0dm = G0dm, Ta = Ta)
> ProdFixed <- prodGCPV(lat = lat, prom = prom, keep.night = FALSE)
> Prod2x <- prodGCPV(lat = lat, prom = prom, modeTrk = "two", keep.night = FALSE)
> ProdHoriz <- prodGCPV(lat = lat, prom = prom, modeTrk = "horiz",
+     keep.night = FALSE)
```

La comparativa de funcionamiento se muestra en la figura 9.

4.1. Influencia de las sombras

Uno de los factores a tener en cuenta en el funcionamiento de los sistemas fotovoltaicos es el impacto de las sombras en los generadores [6]. En este paquete se incluyen cinco funciones que calculan las sombras mutuas entre generadores pertenecientes a una misma planta. Estas funciones son `fSombra2X`, `fSombraHoriz`, `fSombraEst`, `fSombra6` y `fSombra`. Las tres primeras calculan las sombras en plantas de seguimiento a doble eje, eje horizontal y sistemas estáticos, respectivamente. La función `fSombra6` calcula las sombras en grupos de 6 seguidores de doble eje, permitiendo el cálculo de la sombra promedio de la planta. Finalmente, la función `fSombra` permite el uso de cualquiera de ellas de forma sencilla a través de su parámetro `modoTrk`. Por ejemplo, la siguiente secuencia calcula el sombreado en un seguidor de doble eje rodeado de otros cinco. Las dimensiones de la estructura del seguidor y la configuración en filas y columnas de la planta vienen recogidas en la lista `struct`, mientras que las distancias entre seguidores están definidas en el `data.frame` `distances`.

La figura 10 muestra la incidencia de sombreado a lo largo del día (eje X) y del año (eje Y). Es evidente que el sombreado es de importancia en las primeras y últimas horas del día, y en algunas horas cercanas al mediodía en los meses de invierno.

En este caso, dado que el `data.frame` `distances` sólo contiene una fila, la función`fSombra6` construye una red simétrica de seguidores, situando en (0,0,0) el seguidor en estudio. Esta misma red se podría haber construido con:

```
> distances = data.frame(Lew = c(-40, 0, 40, -40, 40), Lns = c(30,
+     30, 30, 0, 0), H = 0)
> ShdFactor2 <- fSombra6(Angles, distances, struct, prom = FALSE)
> identical(coredata(ShdFactor), coredata(ShdFactor2))

[1] TRUE
```

Además de este caso trivial, el `data.frame` `distances` puede definir una cuadrícula irregular alrededor del seguidor problema. Dado que este seguidor está en (0,0,0) `distances` debe tener 5 filas.

Cuando `prom=TRUE`, a partir de la distribución de seguidores en la planta (`Nrow` y `Ncol`), se obtiene una media ponderada del sombreado en el conjunto completo.

El uso de estas funciones está integrado en la función `prodGCPV`, tal y como muestran los siguientes ejemplos:

```
> struct2x = list(W = 23.11, L = 9.8, Nrow = 2, Ncol = 8)
> dist2x = data.frame(Lew = 40, Lns = 30, H = 0)
```

```
> ComparePac <- CBIND(two = as.zooI(Prod2x)$Pac, horiz = as.zooI(ProdHoriz)$Pac,
+     fixed = as.zooI(ProdFixed)$Pac)
> AngSol = as.zooI(as(ProdFixed, "Sol"))
> ComparePac = CBIND(AngSol, ComparePac)
> mon = month(index(ComparePac))
> p = xyplot(two + horiz + fixed ~ AzS | mon, data = ComparePac,
+     type = "l", auto.key = list(space = "right", lines = TRUE,
+         points = FALSE), ylab = "Pac")
> print(p)
```

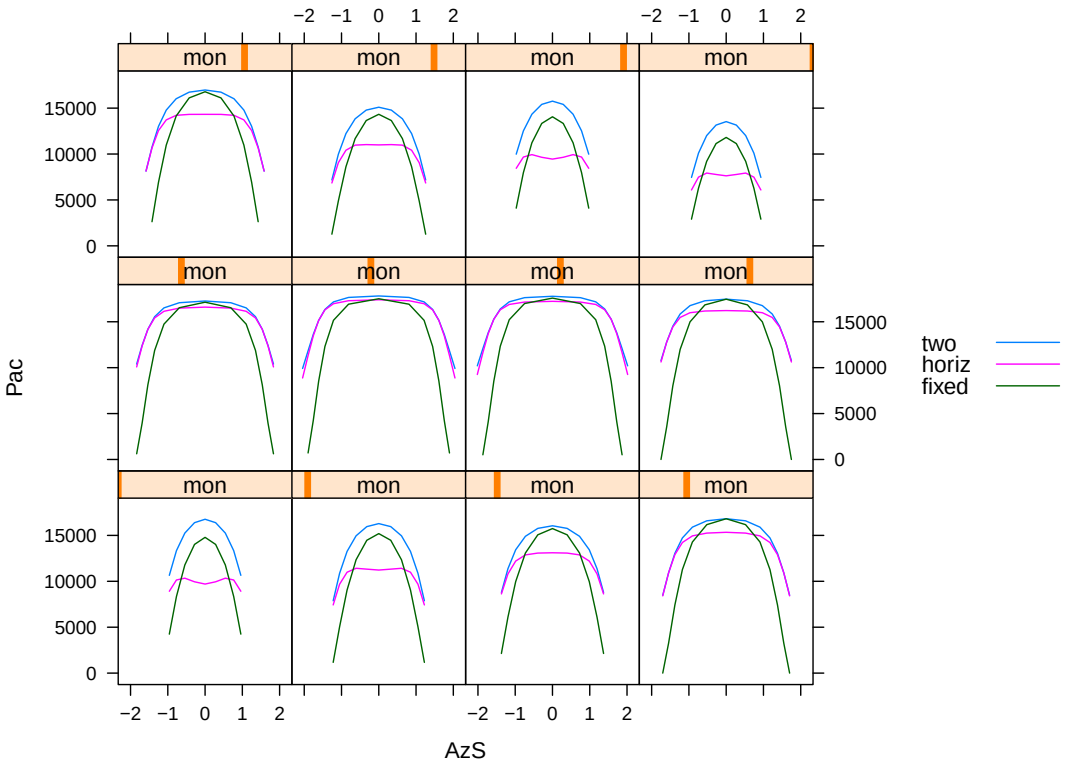

Figura 9: Comparativa de funcionamiento entre sistemas de seguimiento

```
> YlOrBr = brewer.pal(n = 5, "YlOrBr")
> p <- levelplot(FS ~ w * day, data = Angles, col.regions = colorRampPalette(YlOrBr,
+     space = "Lab"))
> print(p)
```

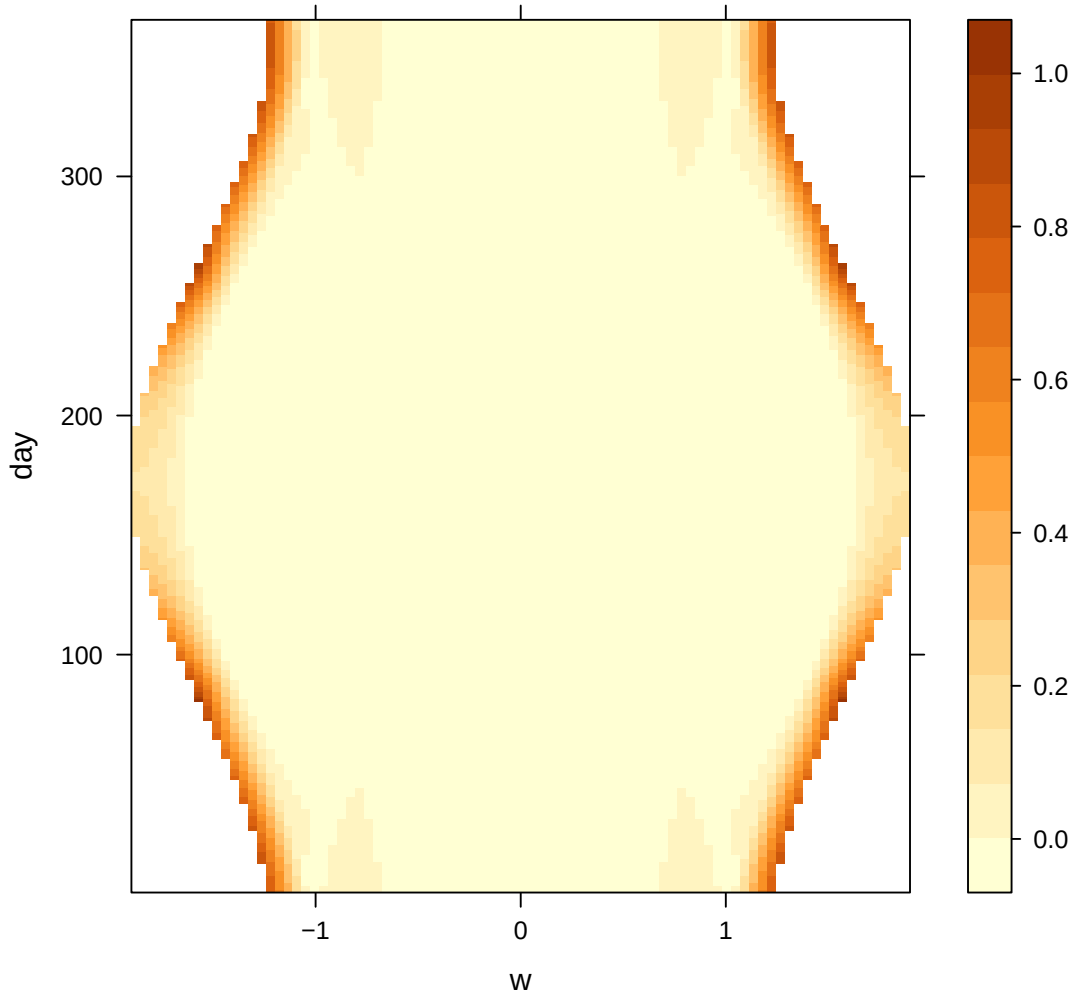

Figura 10: Sombreado en una planta de seguimiento a doble eje.

```
> prod2xShd <- prodGCPV(lat = lat, prom = prom, modeTrk = "two",
+     modeShd = "area", struct = struct2x, distances = dist2x)
> print(prod2xShd)

Object of class  ProdGCPV

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
            Eac   Edc    Yf
ene 2010 123.70 136.7 4.675
feb 2010 132.75 146.9 5.017
mar 2010 141.32 156.1 5.341
abr 2010 168.73 186.6 6.377
may 2010 187.22 207.0 7.076
jun 2010 217.19 240.2 8.209
jul 2010 216.97 240.0 8.200
ago 2010 187.18 207.0 7.074
sep 2010 158.68 175.8 5.997
oct 2010 124.47 137.6 4.704
nov 2010 117.40 129.7 4.437
dic 2010  93.45 103.2 3.532

Yearly values:
       Eac   Edc   Yf
2010 56881 62895 2150
-----------------
Mode of tracking:  two
    Inclination limit: 90
-----------------
Generator:
    Modules in series:  12
    Modules in parallel:  11
    Nominal power (kWp):  26.5
```

```
> structHoriz = list(L = 4.83)
> distHoriz = data.frame(Lew = structHoriz$L * 4, H = 0)
> prodHorizShd <- prodGCPV(lat = lat, prom = prom, sample = "10 min",
+     modeTrk = "horiz", modeShd = "area", betaLim = 60, distances = distHoriz,
+     struct = structHoriz)
> print(prodHorizShd)

Object of class  ProdGCPV

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
            Eac    Edc    Yf
ene 2010  84.51  93.37 3.194
feb 2010 101.34 111.95 3.830
mar 2010 124.15 137.13 4.692
abr 2010 158.60 175.33 5.994
may 2010 182.95 202.36 6.914
jun 2010 200.53 221.96 7.579
jul 2010 198.71 219.91 7.510
ago 2010 178.00 196.86 6.727
sep 2010 143.51 158.54 5.424
oct 2010  99.07 109.49 3.744
nov 2010  82.11  90.79 3.103
dic 2010  62.17  68.94 2.350

Yearly values:
       Eac   Edc   Yf
2010 49196 54403 1859
-----------------
Mode of tracking:  horiz
    Inclination limit: 60
-----------------
Generator:
    Modules in series:  12
    Modules in parallel:  11
    Nominal power (kWp):  26.5
```

```
> prodHorizBT <- prodGCPV(lat = lat, prom = prom, sample = "10 min",
+     modeTrk = "horiz", modeShd = "bt", betaLim = 60, distances = distHoriz,
+     struct = structHoriz)
> print(prodHorizBT)

Object of class  ProdGCPV

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
           Eac    Edc    Yf
ene 2010  83.85  92.66 3.169
feb 2010 100.64 111.13 3.804
mar 2010 123.66 136.61 4.674
abr 2010 157.99 174.67 5.971
may 2010 182.22 201.58 6.887
jun 2010 199.60 220.88 7.544
jul 2010 197.73 218.86 7.473
ago 2010 177.14 195.94 6.695
sep 2010 142.87 157.86 5.400
oct 2010  98.54 108.85 3.724
nov 2010  81.42  89.98 3.077
dic 2010  61.77  68.51 2.335

Yearly values:
       Eac   Edc   Yf
2010 48946 54126 1850
-----------------
Mode of tracking:  horiz
    Inclination limit: 60
-----------------
Generator:
    Modules in series:  12
    Modules in parallel:  11
    Nominal power (kWp):  26.5
```

4.2. Ubicación de seguidores en una planta

El efecto de las sombras descrito en el apartado anterior exige separar los elementos que componen una planta fotovoltaica. Por una parte, una mayor separación disminuye las pérdidas por sombreado mutuo y aumenta la productividad del sistema. Pero por otra aumentan los costes relacionados con el área ocupada por unidad de potencia y aumentan los costes relacionados con los elementos de unión entre estructuras (cableado, canalizaciones, zanjas). Por tanto, la separación óptima entre elementos (seguidores o estructuras estáticas) es aquella que conduce al mínimo valor del coste de la energía producida por el sistema.

Este paquete incluye una función llamada `optimShd` (en la versión anterior era `optimSombra`) diseñada para calcular el efecto del sombreado mutuo en un conjunto de posibles separaciones entre elementos [6]. El consiguiente conjunto de resultados combina la productividad (`Yf`) y la ocupación de terreno (`ROT`) para cada una de las posibles separaciones. El diseñador deberá tomar la decisión oportuna a partir de este conjunto de resultados efectuando las traducciones económicas que considere necesarias.

Supongamos un sistema de seguimiento de eje horizontal sin *backtracking* con un seguidor de 4,83 m de altura. Nos interesa estudiar las separaciones comprendidas entre 2 y 5 veces esta dimensión. Además, analizaremos el funcionamiento limitando el ángulo de inclinación del seguidor:

```
> structHoriz = list(L = 4.83)
> distHoriz = list(Lew = structHoriz$L * c(2, 5))
> Shd12Horiz <- optimShd(lat = lat, prom = prom, modeTrk = "horiz",
+     betaLim = 60, distances = distHoriz, res = 2, struct = structHoriz,
+     modeShd = "area", prog = FALSE)
> print(Shd12Horiz)

Object of class  Shade

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Dimensions of structure:
$L
[1] 4.83

Shade calculation mode:
[1] "area"
Productivity without shadows:
Object of class  ProdGCPV

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
            Eac    Edc    Yf
ene 2010  87.48  96.48 3.306
feb 2010 110.99 122.45 4.195
mar 2010 127.37 140.51 4.814
abr 2010 166.91 184.33 6.308
may 2010 184.90 204.33 6.988
jun 2010 213.39 235.96 8.065
jul 2010 214.37 237.02 8.102
ago 2010 185.35 204.80 7.005
sep 2010 158.57 175.05 5.993
oct 2010 106.20 117.19 4.014
nov 2010  84.08  92.75 3.178
dic 2010  65.45  72.42 2.474

Yearly values:
       Eac   Edc   Yf
2010 51901 57326 1962
-----------------
Mode of tracking:  horiz
    Inclination limit: 60
-----------------
Generator:
    Modules in series:  12
    Modules in parallel:  11
    Nominal power (kWp):  26.5

Summary of results:
      Lew              H           FS              GCR             Yf
 Min.   : 9.66   Min.   :0   Min.   :0.0397   Min.   :2.00   Min.   :1714
 1st Qu.:13.16   1st Qu.:0   1st Qu.:0.0495   1st Qu.:2.72   1st Qu.:1793
 Median :16.66   Median :0   Median :0.0642   Median :3.45   Median :1836
 Mean   :16.66   Mean   :0   Mean   :0.0712   Mean   :3.45   Mean   :1822
 3rd Qu.:20.16   3rd Qu.:0   3rd Qu.:0.0859   3rd Qu.:4.17   3rd Qu.:1865
 Max.   :23.66   Max.   :0   Max.   :0.1261   Max.   :4.90   Max.   :1884
```

La función `optimShd` crea un objeto de clase `Shade` (en la anterior versión era `sombra`). Para esta clase `solaR` incorpora un método S4 de `plot` que permite representar gráficamente los resultados (figura 11).

Realicemos los cálculos para un sistema estático (figura 12):

```
> plot(Shd12Horiz)
```

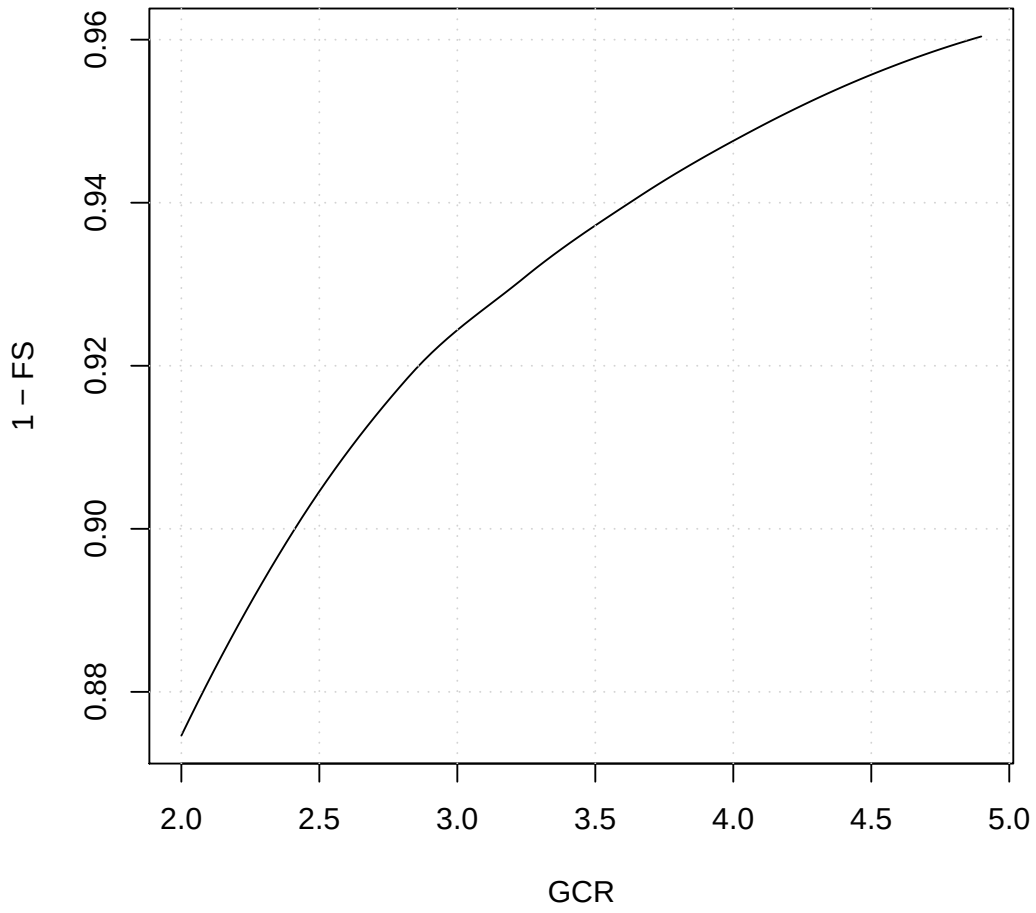

Figura 11: Sombreado mutuo en una planta de seguimiento de eje horizontal sin backtracking

```
> structFixed = list(L = 5)
> distFixed = list(D = structFixed$L * c(1, 3))
> Shd12Fixed <- optimShd(lat = lat, prom = prom, modeTrk = "fixed",
+     distances = distFixed, res = 1, struct = structFixed, modeShd = "area",
+     prog = FALSE)
> print(Shd12Fixed)

Object of class  Shade

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Dimensions of structure:
$L
[1] 5

Shade calculation mode:
[1] "area"
Productivity without shadows:
Object of class  ProdGCPV

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
            Eac    Edc    Yf
ene 2010  91.56 101.24 3.460
feb 2010  99.34 110.18 3.755
mar 2010 108.42 120.10 4.098
abr 2010 121.63 134.64 4.597
may 2010 128.84 143.01 4.869
jun 2010 133.28 147.92 5.037
jul 2010 132.94 147.58 5.024
ago 2010 128.31 142.04 4.849
sep 2010 118.32 131.04 4.472
oct 2010  94.52 104.83 3.572
nov 2010  87.33  96.56 3.301
dic 2010  70.84  78.47 2.677

Yearly values:
       Eac   Edc   Yf
2010 40017 44345 1512
-----------------
Mode of tracking:  fixed
    Inclination:  27.2
    Orientation:  0
-----------------
Generator:
    Modules in series:  12
    Modules in parallel:  11
    Nominal power (kWp):  26.5

Summary of results:
       D               H           FS                GCR            Yf
 Min.   : 5.0   Min.   :0   Min.   :0.000113   Min.   :1.0   Min.   :1319
 1st Qu.: 7.5   1st Qu.:0   1st Qu.:0.000749   1st Qu.:1.5   1st Qu.:1477
 Median :10.0   Median :0   Median :0.002811   Median :2.0   Median :1508
 Mean   :10.0   Mean   :0   Mean   :0.023061   Mean   :2.0   Mean   :1478
 3rd Qu.:12.5   3rd Qu.:0   3rd Qu.:0.023409   3rd Qu.:2.5   3rd Qu.:1511
 Max.   :15.0   Max.   :0   Max.   :0.127777   Max.   :3.0   Max.   :1512
```

Por último, supongamos que queremos ubicar un seguidor de 23,11 m de ancho y 9,8 m de alto en una planta. Nos interesa configurar esta planta en una red de 2 filas y 8 columnas.

```
> struct2x = list(W = 23.11, L = 9.8, Nrow = 2, Ncol = 8)
```

Probaremos las separaciones comprendidas entre 30 m y 50 m para la dirección E-O y entre 20 m y 50 m para la dirección N-S.

```
> dist2x = list(Lew = c(30, 50), Lns = c(20, 50))
```

Obtenemos los resultados con una resolución de 5 m con la siguiente instrucción:

```
> plot(Shd12Fixed)
```

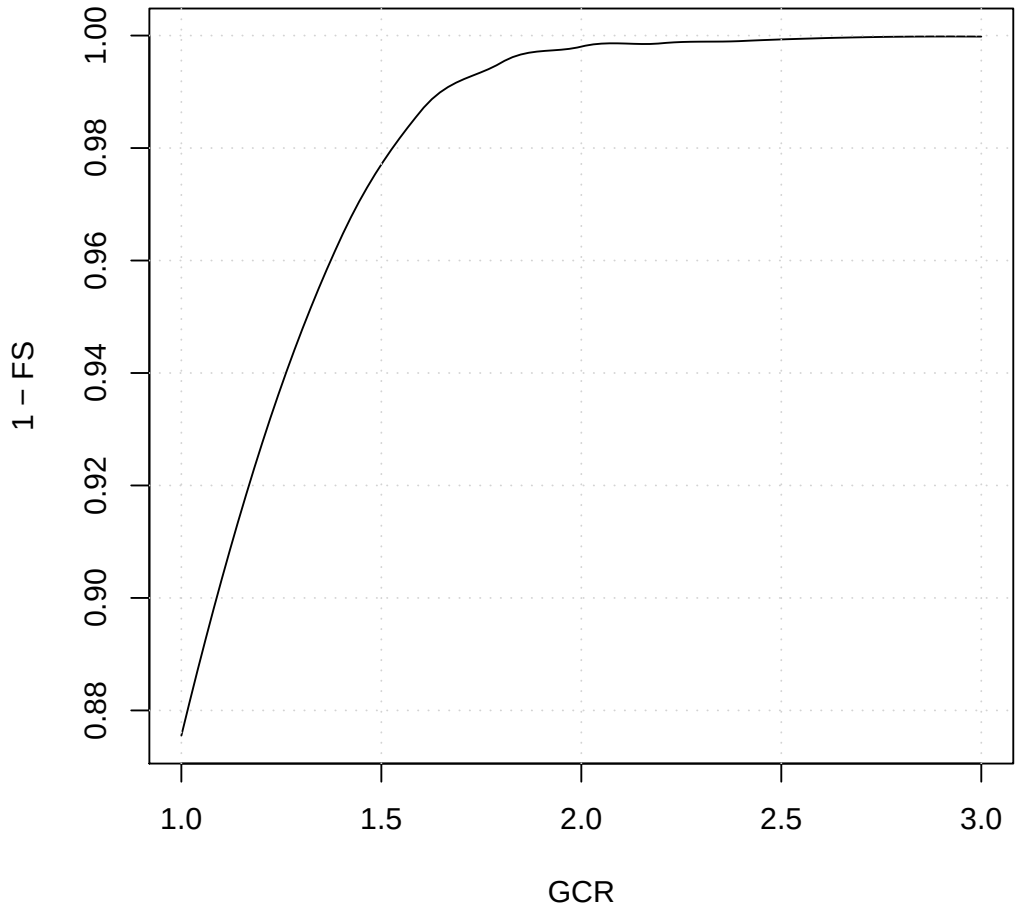

Figura 12: Sombreado mutuo en una planta con estructuras estáticas

```
> ShdM2x <- optimShd(lat = lat, prom = prom, modeTrk = "two", modeShd = c("area",
+     "prom"), distances = dist2x, struct = struct2x, res = 5,
+     prog = FALSE)
> print(ShdM2x)

Object of class  Shade

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Dimensions of structure:
$W
[1] 23.11

$L
[1] 9.8

$Nrow
[1] 2

$Ncol
[1] 8

Shade calculation mode:
[1] "area" "prom"
Productivity without shadows:
Object of class  ProdGCPV

Source of meteorological information: prom-

Latitude of source:  37.2 degrees
Latitude for calculations:  37.2 degrees

Monthly averages:
            Eac   Edc    Yf
ene 2010 128.01 141.4 4.838
feb 2010 142.11 157.0 5.371
mar 2010 144.55 159.6 5.463
abr 2010 176.14 194.6 6.657
may 2010 189.53 209.5 7.163
jun 2010 221.17 244.6 8.359
jul 2010 222.88 246.4 8.423
ago 2010 193.42 213.8 7.310
sep 2010 175.88 194.4 6.647
oct 2010 131.09 144.7 4.955
nov 2010 120.25 132.8 4.545
dic 2010  98.94 109.2 3.740

Yearly values:
       Eac   Edc   Yf
2010 59144 65355 2235
-----------------
Mode of tracking:  two
    Inclination limit: 90
-----------------
Generator:
    Modules in series:  12
    Modules in parallel:  11
    Nominal power (kWp):  26.5

Summary of results:
      Lew          Lns           H           FS              GCR
 Min.   :30   Min.   :20   Min.   :0   Min.   :0.0147   Min.   : 2.65
 1st Qu.:35   1st Qu.:25   1st Qu.:0   1st Qu.:0.0215   1st Qu.: 4.53
 Median :40   Median :35   Median :0   Median :0.0336   Median : 5.96
 Mean   :40   Mean   :35   Mean   :0   Mean   :0.0359   Mean   : 6.18
 3rd Qu.:45   3rd Qu.:45   3rd Qu.:0   3rd Qu.:0.0457   3rd Qu.: 7.73
 Max.   :50   Max.   :50   Max.   :0   Max.   :0.0913   Max.   :11.04
       Yf
 Min.   :2031
 1st Qu.:2133
 Median :2160
 Mean   :2155
 3rd Qu.:2187
 Max.   :2203
```

5. Funcionamiento de sistemas fotovoltaicos de bombeo

5.1. Modelado de bombas centrífugas

El primer paso a dar para analizar el funcionamiento de un sistema fotovoltaico de bombeo es caracterizar la bomba centrífuga en unas condiciones de altura manométrica constante (suposición de funcionamiento en estos sistemas) [1]. Empleamos la función `fPump` (antes `fBomba`) para analizar el funcionamiento de una bomba SP8A44

```
> plot(ShdM2x)
```

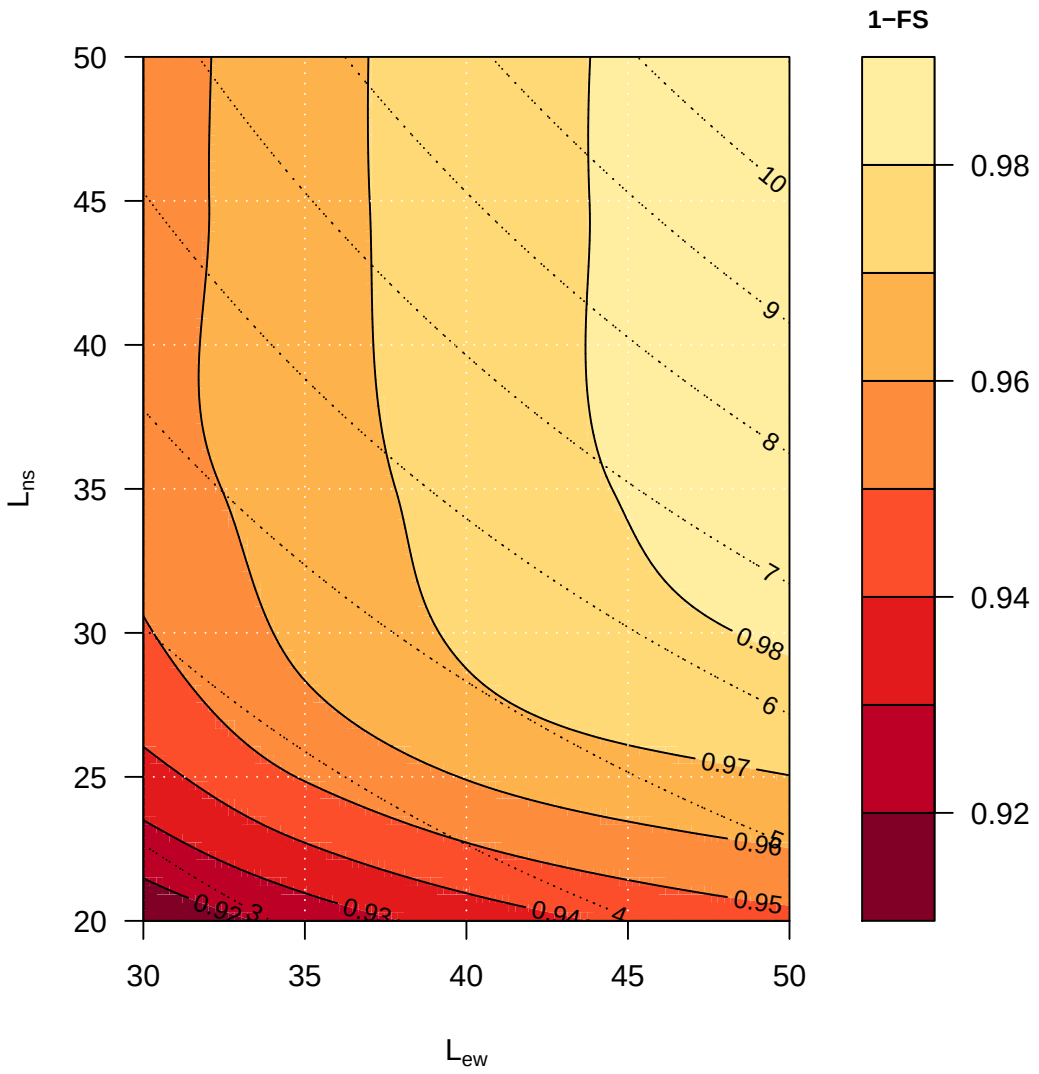

Figura 13: Sombreado mutuo en una planta de seguimiento a doble eje

(http://net.grundfos.com/Appl/WebCAPS/InitCtrl?mode=1) trabajando a una altura manométrica de $H = 40\,\text{m}$. La información sobre esta bomba la extraemos de `pumpCoef` (antes `CoefBomba`):

```
> data(pumpCoef)
> CoefSP8A44 <- subset(pumpCoef, Qn == 8 & stages == 44)
> fSP8A44 <- fPump(pump = CoefSP8A44, H = 40)
```

El resultado de `fPump` es un conjunto de funciones que relacionan, en un rango de funcionamiento determinado por la bomba, la potencia mecánica e hidráulica, el caudal y la frecuencia con la potencia eléctrica de entrada. Con estas funciones es posible calcular estos parámetros de funcionamiento para cualquier valor de potencia eléctrica que esté comprendido dentro del rango de funcionamiento (figuras 14 y 15):

```
> SP8A44 = with(fSP8A44, {
+     Pac = seq(lim[1], lim[2], by = 100)
+     Pb = fPb(Pac)
+     etam = Pb/Pac
+     Ph = fPh(Pac)
+     etab = Ph/Pb
+     f = fFreq(Pac)
+     Q = fQ(Pac)
+     result = data.frame(Q, Pac, Pb, Ph, etam, etab, f)
+ })
> SP8A44$etamb = with(SP8A44, etab * etam)
```

5.2. Nomogramas de sistemas fotovoltaicos de bombeo

En las licitaciones públicas de esta aplicación es de uso común la norma internacional IEC 61725. Este texto define el perfil de irradiancia a partir de la duración del día, la irradiación diaria y el valor máximo de irradiancia. A partir de este perfil se puede calcular el funcionamiento de un sistema fotovoltaico de bombeo para diferentes alturas manométricas y un conjunto de potencias de generador fotovoltaico. Al representar las combinaciones posibles en un gráfico de doble entrada se obtiene una herramienta que ayuda a la selección de la mejor combinación de bomba y generador en unas condiciones de radiación y altura [1]. Este tipo de gráficos se obtienen con la función `NmgPVPS`. Por ejemplo, generemos un nomograma (figura 16) para la bomba SP8A44 trabajando en un rango de alturas desde 50 a 80 metros, alimentada por diferentes potencias de generador fotovoltaico. La extraña forma de la curva correspondiente a una altura de 50 metros indica que el funcionamiento de esta bomba a esta altura no es adecuada.

Debe señalarse que los resultados obtenidos con este método son diferentes a los que resultan de procedimientos basados en otro tipo de perfiles de irradiancia, como los explicados anteriormente.

5.3. Productividad de sistemas fotovoltaicos de bombeo

Otro enfoque diferente consiste en simular el funcionamiento del sistema empleando el mismo itinerario de cálculo que hemos recorrido con los sistemas de conexión a red. Si allí empleamos la función `prodSFCR` ahora nos serviremos de la función `prodPVPS`. Los parámetros de entrada de esta función son muy parecidos a aquella, salvo las pertinentes diferencias en la definición del inversor. En esta función no están habilitadas las opciones de cálculo de sombra. Nuevamente con la bomba SP8A44, calculemos el caudal que proporcionará este sistema con un generador de 5500 Wp contra una altura manométrica de 50 metros descargando datos de www.mapa.es/siar.

```
> lab = c(expression(eta[motor]), expression(eta[pump]), expression(eta[mp]))
> p <- xyplot(etam + etab + etamb ~ Pac, data = SP8A44, type = "l",
+     ylab = "Eficiencia")
> print(p + glayer(panel.text(x[1], y[1], lab[group.number], pos = 3)))
```

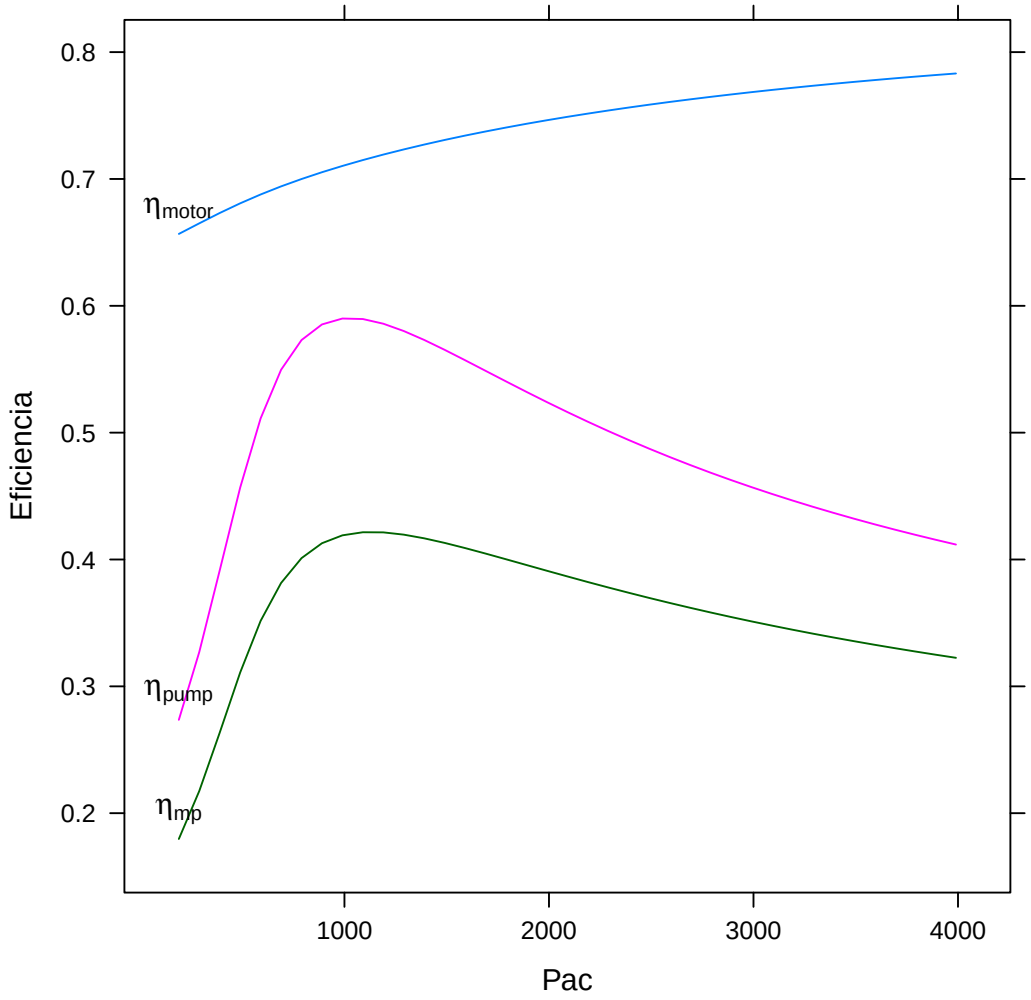

Figura 14: Eficiencia de motor, bomba y motobomba para diferentes valores de potencia eléctrica de una bomba SP8A44 trabajando a una altura manométrica de $H = 40\,\mathrm{m}$

```
> lab = c(expression(P[pump]), expression(P[hyd]))
> p <- xyplot(Pb + Ph ~ Pac, data = SP8A44, type = "l", ylab = "Potencia (W)",
+     xlab = "Potencia AC (W)")
> print(p + glayer(panel.text(x[length(x)], y[length(x)], lab[group.number],
+     pos = 3)))
```

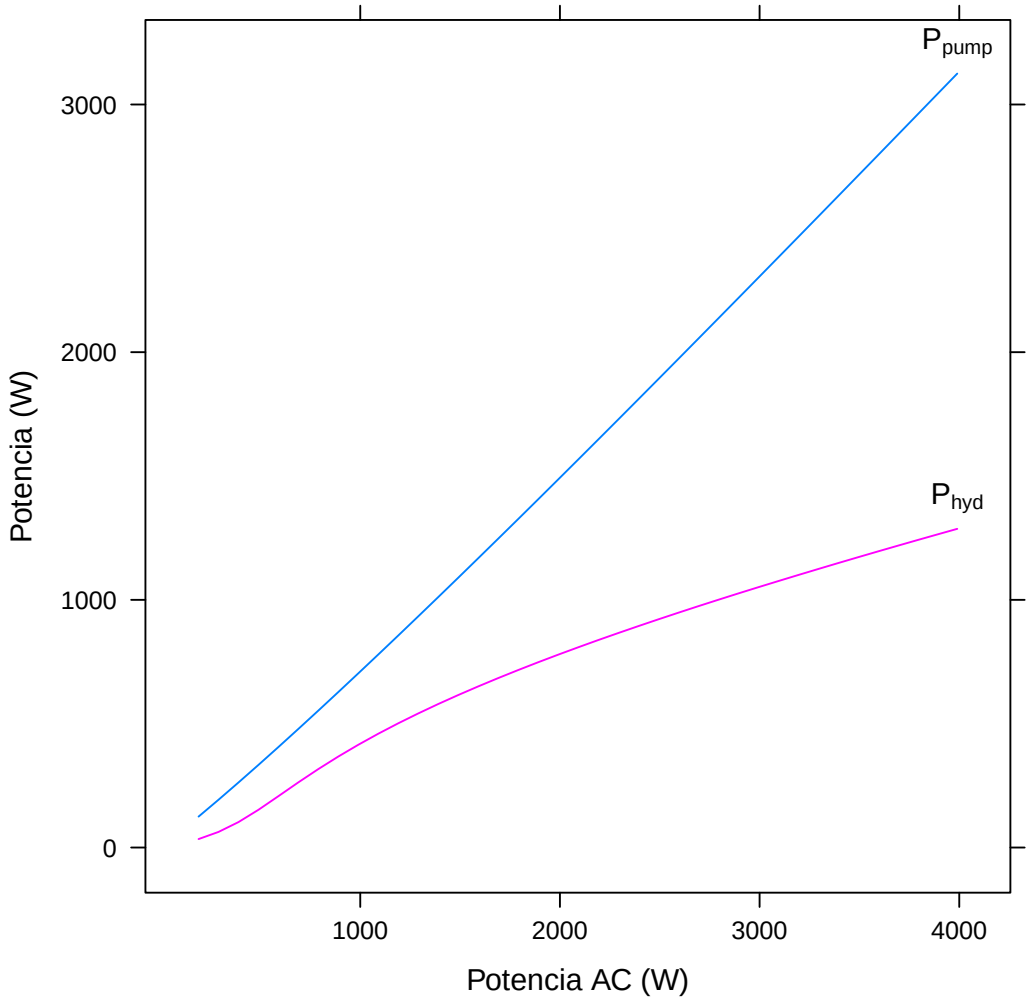

Figura 15: Potencia mecánica e hidráulica frente a la potencia eléctrica de una bomba SP8A44 trabajando a una altura manométrica de $H = 40\,\mathrm{m}$

```
> Pg = seq(3000, 5500, by = 500)
> H = seq(50, 80, by = 5)
> NmgSP8A44 <- NmgPVPS(pump = CoefSP8A44, Pg = Pg, H = H, Gd = 6000,
+     title = "Elección de bombas", theme = custom.theme())
> print(NmgSP8A44$plot)
```

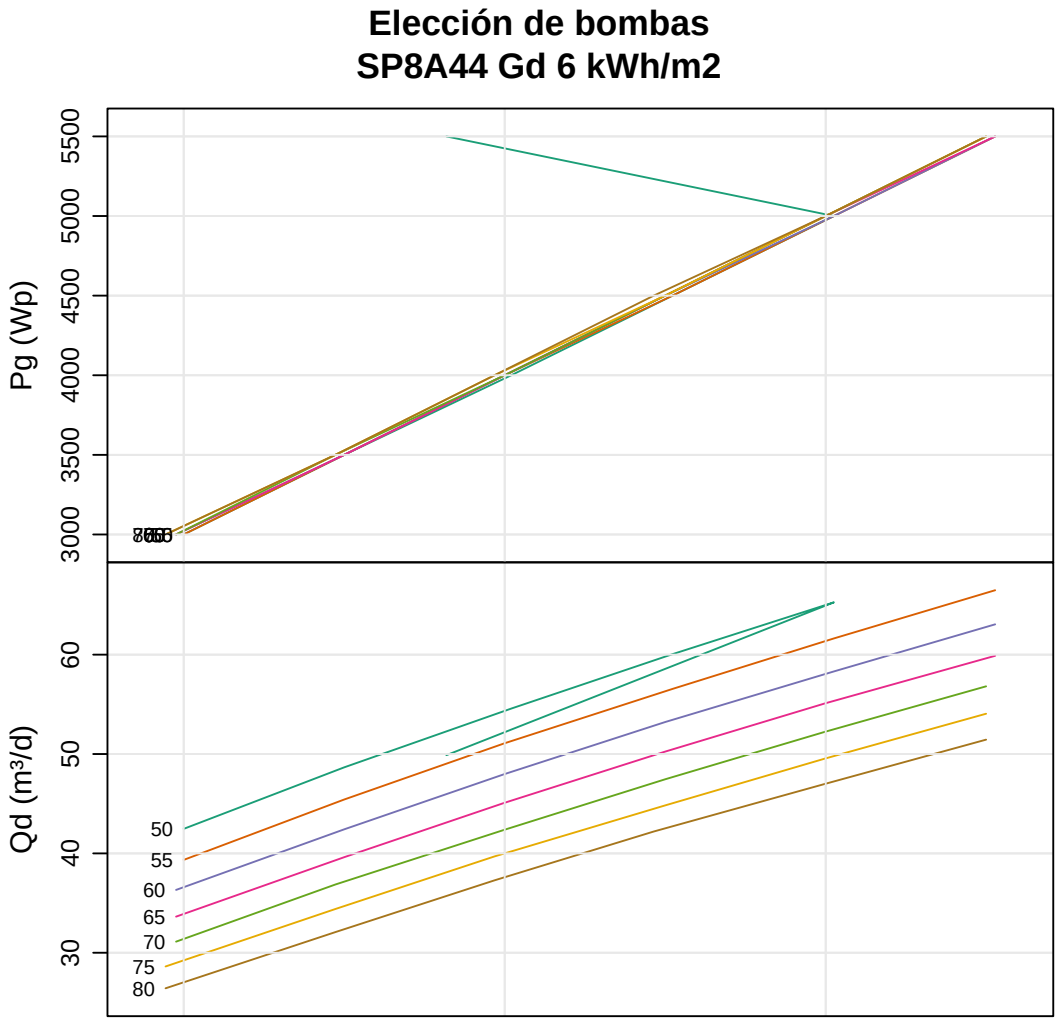

Figura 16: Nomograma para la bomba SP8A44 trabajando en un rango de alturas desde 50 a 80 metros, alimentada por diferentes potencias de generador fotovoltaico

```
> prodSP8A44 <- prodPVPS(lat = 41, modeRad = "mapa", mapa = list(prov = 28,
+     est = 3, start = "01/01/2009", end = "31/12/2009"), pump = CoefSP8A44,
+     Pg = 5500, H = 50)

Downloading data from www.mapa.es/siar...

> print(prodSP8A44)

Object of class  ProdPVPS

Source of meteorological information: mapa-Est: 3 Prov: 28

Latitude of source:  41 degrees
Latitude for calculations:  41 degrees

Monthly averages:
            Eac    Qd    Yf
ene 2009 12.212 35.09 2.220
feb 2009 17.926 50.08 3.259
mar 2009 22.717 62.88 4.130
abr 2009 23.470 66.52 4.267
may 2009 27.700 78.55 5.036
jun 2009 26.231 75.39 4.769
jul 2009 28.641 81.38 5.207
ago 2009 28.180 79.82 5.124
sep 2009 24.587 69.68 4.470
oct 2009 23.877 67.31 4.341
nov 2009 17.578 50.85 3.196
dic 2009  9.146 25.89 1.663

Yearly values:
      Eac    Qd   Yf
2009 7985 22634 1452
-----------------
Mode of tracking:  fixed
    Inclination:  31
    Orientation:  0
-----------------
Pump:
    Qn:  8
    Stages:  44
Height (m):  50
Generator (Wp):  5500
```

La relación entre el caudal y la irradiancia efectiva queda recogido en la figura 17.

Supongamos que deseamos extraer más caudal con esta bomba y ampliamos el generador fotovoltaico hasta una potencia de 7000 Wp. Podemos comprobar que, dado el rango de funcionamiento de la bomba, la decisión no es muy acertada. La productividad (`Yf`) y el caudal normalizado a la potencia (`Qn`) han descendido respecto de la otra configuración. Observando la gráfica correspondiente (figura 18) comprobamos que en los meses centrales del año, en los momentos de alta irradiancia la bomba alcanza su caudal y frecuencia máximas, y el variador de frecuencia ordena el paro del sistema.

```
> prodSP8A44Lim <- prodPVPS(lat, modeRad = "prev", prev = prodSP8A44,
+     pump = CoefSP8A44, H = 50, Pg = 7000)
```

6. Análisis estadístico de Plantas Fotovoltaicas

En una planta fotovoltaica, los sistemas individuales que la componen son, teóricamente, idénticos, y su funcionamiento a lo largo del tiempo debiera ser aproximadamente el mismo. Sin embargo, debido a sus diferencias prácticas –tolerancia en potencia, pérdidas por dispersión, suciedad–, el funcionamiento individual de cada sistema se desvía del comportamiento promedio del conjunto. Cuando un sistema se comporta correctamente estas desviaciones están comprendidas en un rango determinado y no son signo de mal funcionamiento.

Tratando estas desviaciones como un proceso estocástico, se puede aplicar un análisis estadístico al funcionamiento del conjunto de sistemas para identificar un sistema defectuoso como aquel que se aparta significativamente del comportamiento promedio.

En [7] se expone un método para llevar a cabo este análisis y una herramienta gráfica que resume los resultados. Este gráfico, denominado "Target Diagram", muestra en un sólo gráfico la raiz del valor cuadrático medio, la media y la desviación estándar de la diferencia entre las plantas y una referencia dada (por defecto, la mediana).

Esta versión de `solaR` incluye dos funciones –`analyzeData` y `TargetDiagram`– para realizar este análisis. Las siguientes líneas muestran un ejemplo de aplicación basado en unos datos de productividad de una planta compuesta por 22 sistemas nominalmente iguales (estos datos son accesibles mediante `data(prodEx)`).

```
> data(prodEx)
> prodStat <- analyzeData(prodEx)
```

```
> p = xyplot(Q ~ Gef | month, data = prodSP8A44, cex = 0.5, type = c("p",
+     "smooth"), col.symbol = "gray", col.line = "black")
> print(p)
```

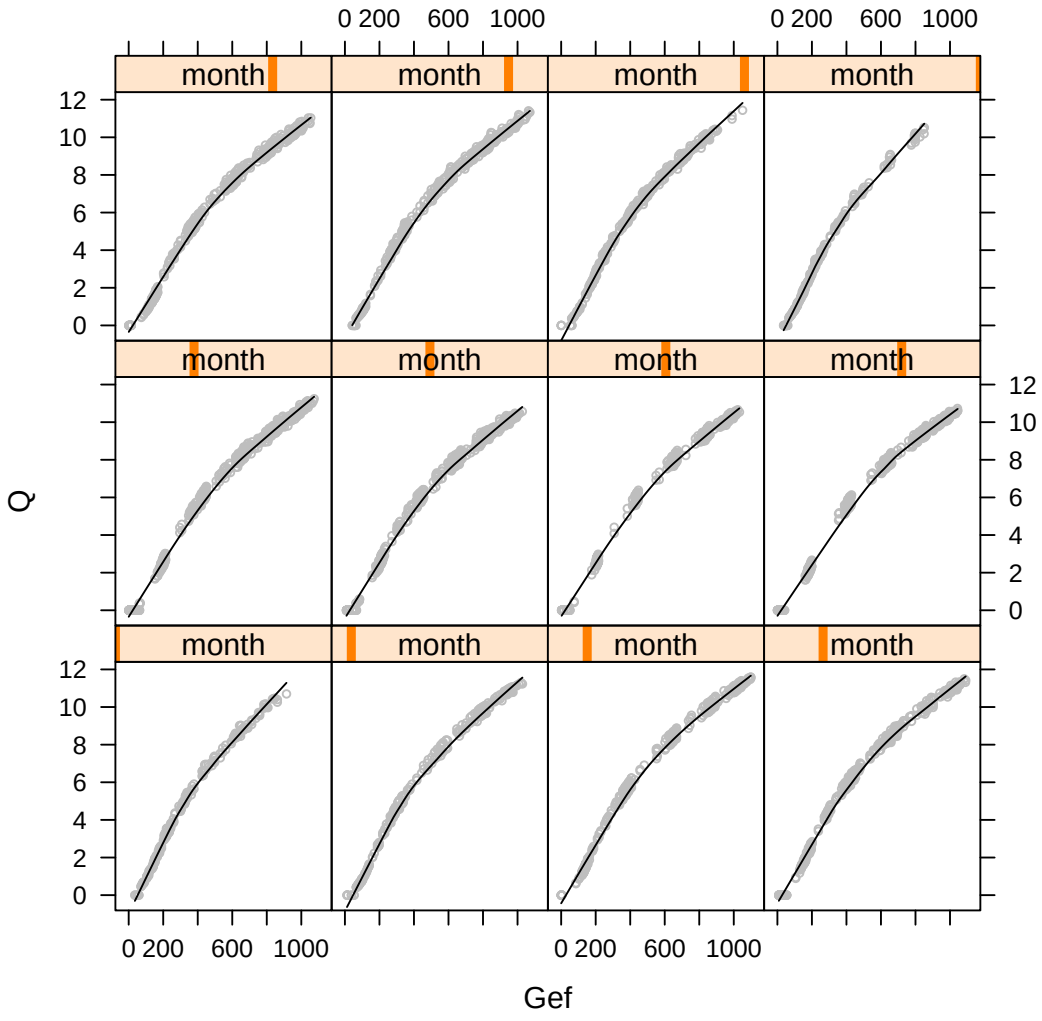

Figura 17: Caudal frente a irradiancia de un sistema basado en la bomba SP8A44 con un generador de 5500 Wp contra una altura manométrica de 50 metros

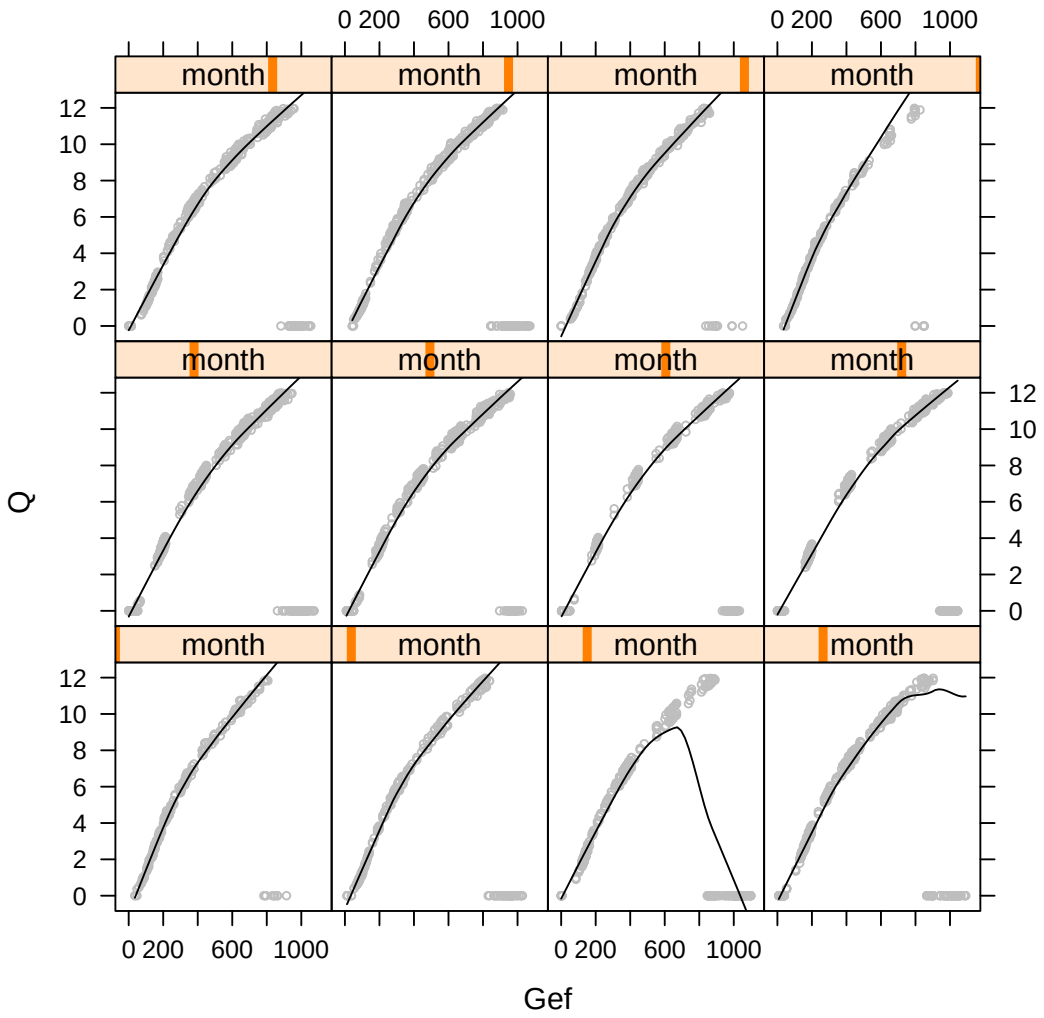

Figura 18: Caudal frente a irradiancia de un sistema basado en la bomba SP8A44 con un generador de 7000 Wp contra una altura manométrica de 50 metros

```
> p = xyplot(prodStat$stat)
> print(p)
```

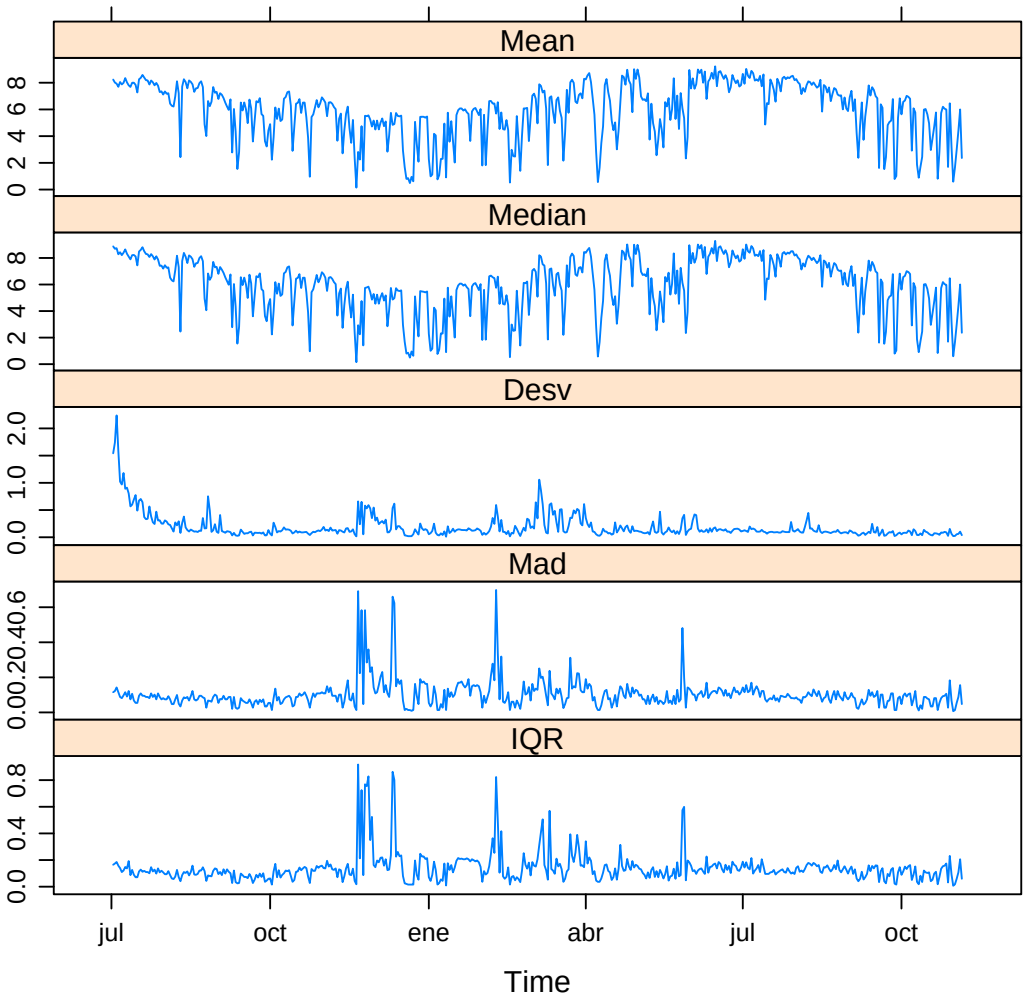

Figura 19: Análisis estadístico de un conjunto de 22 plantas.

```
> day = as.Date("2008-8-29")
> ndays = c(5, 10, 15, 20)
> palette = brewer.pal(n = length(ndays), name = "Set1")
> TDColor <- TargetDiagram(prodEx, end = day, ndays = ndays, color = palette)
> print(TDColor$plot)
```

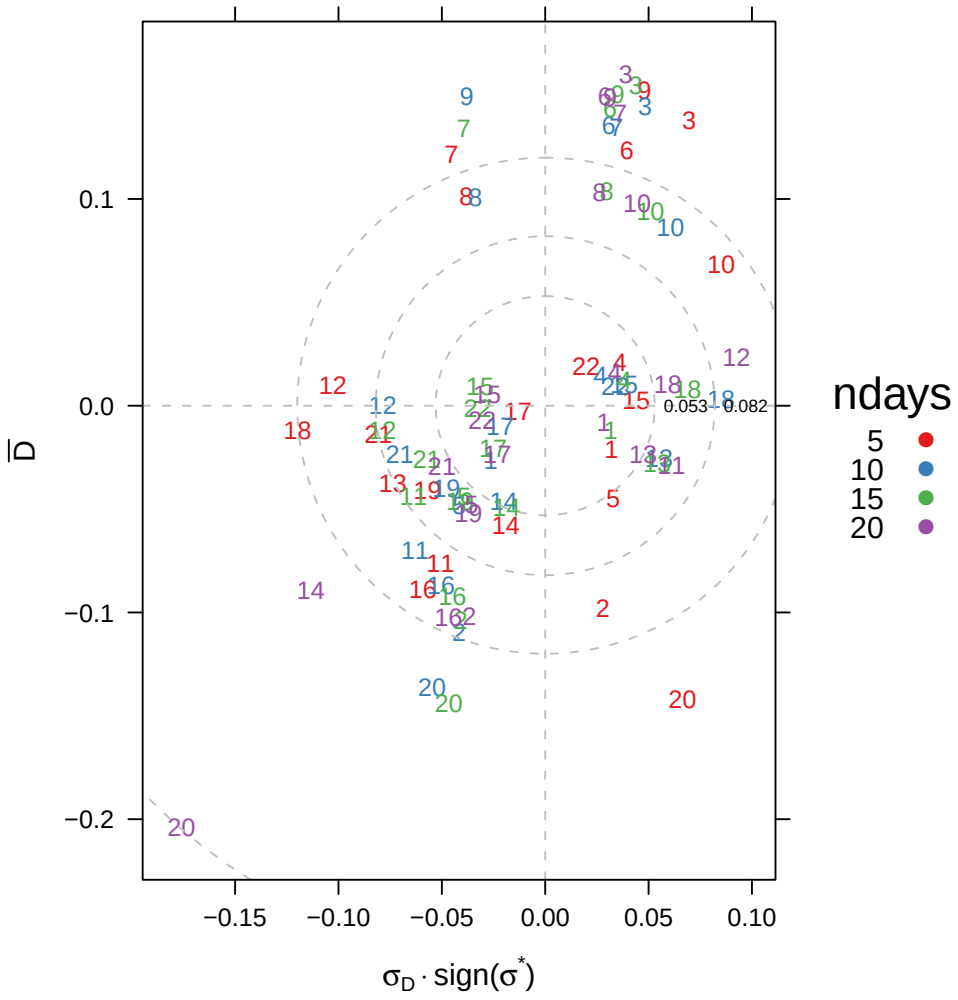

Figura 20: "Target Diagram" del análisis estadístico de un conjunto de 22 plantas durante determinados períodos temporales para detectar sistemas con mal funcionamiento.

Referencias

[1] M.~Alonso Abella, E.~Lorenzo, and F.~Chenlo. Pv water pumping systems based on standard frequency converters. *Progress in Photovoltaics: Research and Applications*, 11(3):179–191, 2003.

[2] Thomas~A. Huld, Marcel Súri, Ewan~D. Dunlop, and Fabio Micale. Estimating average daytime and daily temperature profiles within europe. *Environmental Modelling & Software*, 21(12):1650 – 1661, 2006.

[3] N.~Martin and J.~M. Ruíz. Calculation of the pv modules angular losses under field conditions by means of an analytical model. *Solar Energy Materials & Solar Cells*, 70:25–38, 2001.

[4] J.~K. Page. The calculation of monthly mean solar radiation for horizontal and inclined surfaces from sunshine records for latitudes 40n-40s. In *U.N. Conference on New Sources of Energy*, volume~4, pages 378–390, 1961.

[5] D.~Panico, P.~Garvison, H.~J. Wenger, and D.~Shugar. Backtracking: a novel strategy for tracking pv systems. In *IEEE Photovoltaic Specialists Conference*, pages 668–673, 1991.

[6] O.~Perpiñán. *Grandes Centrales Fotovoltaicas: producción, seguimiento y ciclo de vida*. PhD thesis, UNED, 2008.

[7] O.~Perpiñán. Statistical analysis of the performance and simulation of a two-axis tracking pv system. *Solar Energy*, 83(11):2074–2085, 2009.

[8] A.~Zeileis and G.~Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27, 2005.