

Estadísticas Episodios

Este documento describe el detalle del cálculo y obtención de las variables que componen las estadísticas de cada período de día o noche de la tabla EPI de la aplicación `Actividorm` y la app nueva.

El desarrollo de la app nueva tiene en cuenta que hay muchos análisis anteriores (a la fecha), por esa razón se ha replicado el modelo de la tabla EPI como formula para procesar datos viejos usando las funciones nuevas.

Definiciones

Este vocabulario se usará durante todo el proceso y en los comentarios del código:

1. Un **id** corresponde un **sujeto**
2. Cada sujeto tendrá una serie de **episodios** de sueño o vigilia determinados por el estado actigráfico filtrado desde la detección del algoritmo de MiniMitter.
3. Estos episodios se encuentran agrupados en **periodos** de día o noche
4. Los períodos son numerados secuencialmente según orden de aparición **01, 02, 03**
5. El primer período corresponde *siempre* al **Día 01** el día que el sujeto se lleva el actígrafo.
6. Luego se sucede el primer par de períodos consecutivos **Noche 02** seguido de **Día 02**
7. Este par de períodos consecutivos puede durar más o menos de 24 horas

Consejos para el análisis

Como existen periodos inutilizables porque los sujetos se retiran el actígrafo, pensar en hacer análisis 24 horas no es siempre posible porque se requieren pares consecutivos, tipo **Noche 02 > Día 02**; solo de existir un par válido de este tipo (en un sujeto) se podrían hacer cálculos de tipo **causa y efecto** de la noche al día. Para trabajar el efecto del día hacia la noche será necesario contar con un par consecutivo de tipo **Día 02 > Noche 03**.

En algún momento incluiré esta opción de día completo

Se debe considerar que estas funciones no hacen cálculos por sujeto, **hacen cálculos sobre el período**, posteriormente se podría promediar para consolidar por sujeto, sin embargo, hacer esto es *unwise* ya que se pierde la información de los períodos al igual que el tamaño muestral, se recomienda que los datos sean analizados con modelos de regresión mixta para aprovechar al máximo la información. Medidas repetidas en SPSS puede no resultar porque el sistema no permitirá muestras desbalanceadas o con valores faltantes.

Nota mental:

En los settings del `Actividorm` y la app nueva se configura un parámetro para determinar la consolidación de un estado de sueño o vigilia, se llama **StateFilterDuration** (statedur en la app mia) y según he visto en algunos casos le asignan valores imposibles como 480 segundos por ejemplo.

El uso de este parámetro depende del tamaño del *epoch*, porque el actígrafo no es continuo en segundos, la medida más pequeña son 15 segundos, si el actígrafo está configurado al minuto este parámetro sólo puede ser utilizado en saltos de 60 segundos. Simplemente no es posible, así que el `Actividorm` debe aproximarlos al *epoch* del actígrafo. En mi app está configurado en minutos, por defecto es `statedur = 00:05`.

Consejos para combinar tablas

Los archivos de estadísticas por período se calculan por separado, es decir, hora de inicio estará en una tabla diferente que las duraciones. Dado que el identificador de sujeto es el **Id** y los períodos de ese sujeto son exclusivos de él, para combinar tablas se tendrá que crear una variable de **llave** que combine ambas variables. Así se crea un identificador único por fila.

Se pueden unir por ejemplo separando por un guión bajo.

```
"10524_Dia 01"  
"10524_Noche 02"  
"10524_Dia 02"  
"10524_Noche 03"
```

Para hacer este cálculo en **Excel** se puede usar la fórmula `concatenar` o su versión abreviada con el operador `&`, en el ejemplo el `id` es un valor de celda al igual que `periodo`.

```
=CONCATENAR(id, "_", periodo)  
=id & "_" & periodo
```

Aunque es un poco tedioso se puede usar la función `buscarv` para combinar las tablas, no es problema **mientras exista una llave** que combine id y periodo, es simple y sencillo.

En el caso de **Stata** se debe usar el comando `generate` y los nombres de las variables. Para combinar usar el comando `merge`.

```
generate key = id + "_" + periodo
```

Como **SPSS** es "especial" concatenar solo es posible cuando ambas variables son de tipo *cadena* y aun así no se le puede agregar un guión bajo a menos que cree una variable que solo contenga ese carácter. Considerando además que el sistema de combinación de archivos de SPSS no es del todo intuitivo y existe la posibilidad de perder registros (y por muchas otras razones) lo más recomendable es no usar SPSS para esta tarea. Si se va a usar SPSS para el análisis se recomienda que la combinación de las tablas se realice en otro programa y luego traer los datos.

Hora decimal

La mayor parte de las veces se registrará la hora decimal, por el simple motivo que las horas normales de tipo `15:44` no se pueden usar para hacer cálculos.

Las horas normales tienen base 6, van de 0 a 59 (o 1 a 60 si se prefiere). Entonces las horas decimales se calculan como:

```
hora.decimal = hora + (minutos/60)
```

De esta forma las `15:20` se transforma en `15.333`. Para interpretar este tipo de variable se toma la parte entera como la hora y la decimal se multiplica por 60 y se obtienen de vuelta los minutos.

Si por algún oscuro motivo se quiere pasar estas horas a Excel o trabajarlas en los formatos nativos de SPSS o Stata *que le vaya bien* porque explicarlo me tomaría mucho rato.

Hora continua

Las horas no son una variable circular aunque lo parezca, por esa razón en ocasiones hay que hacer un ajuste:

Existen algunas horas que atraviesan la media noche y ocurre que el reloj resetea al llegar a ese punto, razón por la cual no se pueden calcular estadísticas, por ejemplo el promedio entre 23:30 y 02:30 es 13:00... otsea....

Para solucionar esto el cálculo debe hacerse sumando 24 horas a la hora que pasa la media noche, es decir calcular el promedio entre 23:30 y 26:30 va a ser 25:00 y eso si es la 1 de la mañana. Por eso siempre hay que calcular los máximos y mínimos antes de usar variables de hora y chequear mediante un histograma.

La única variable que si o si puede tener este problema y tendrá una versión continua es la hora de inicio del período *Noche*, no obstante lo anterior, siempre se deben revisar las variables antes de trabajar.

Para trabajar con archivo antiguos

Hacer esto requiere programar un poco en R, solo es información útil para alguien que desee realizarlo. Las funciones debieran quedar cargadas en el *Global Environment* así que se podrían usar directamente. (cuando termine la librería)

1. Leer el EPI

Del Excel se debe pasar la tabla EPI a un CSV (por las fechas), básicamente es el EPI original. La variable **Nombre.Sujeto** debe contener el ID en numérico.

	Nombre.Sujeto	Periodo.del.registro		Hora	Estado	Actividad	Duracion..min.
1	10013	Noche 01	03-08-2016	1:41	S	1388	290,0
2	10013	Noche 01	03-08-2016	6:31	W	1333	29,0
3	10013	Noche 01	03-08-2016	7:00	S	137	35,0
4	10013	Día 01	03-08-2016	7:35	W	23239	85,0

5	10013	Dia 01	03-08-2016	9:00	S	1951	180,0
6	10013	Dia 01	03-08-2016	12:00	W	2324	17,0
Promedio.actividad.por.minuto Numero.Episodio Episodio.del.estado							
1		4,8		1		S001	
2		46,0		2		W001	
3		3,9		3		S002	
4		273,4		4		W001	
5		10,8		5		S001	
6		136,7		6		W002	

Ojo que el original tiene un pie de página que se debe sacar.

La función a usar es `function_read.epi()`, ya la documentaré como corresponde

2. Leer el ARQ

Misma cosa pero más sencillo se necesita que tenga esta forma en formato CSV.

La función a usar es `function_read.arq()` que también documentaré.

	Nombre.Sujeto	Periodo.del.registro	Jornada	Hora.Inicio
1	10013	Dia 01	Dia Completo(a)	03-08-2016 07:35
2	10013	Dia 02	Dia Completo(a)	04-08-2016 07:49
3	10013	Dia 03	Dia Completo(a)	05-08-2016 07:34
4	10013	Dia 04	Dia Completo(a)	06-08-2016 09:09
5	10207	Dia 01	Dia Completo(a)	03-06-2016 07:34
6	10207	Dia 04	Dia Completo(a)	06-06-2016 10:18

Indispensable que tenga el id en numérico, el período y le hora de inicio.

3. Combinar

Luego se combinan con **merge** mediante la variable **key** que será creada al pasar las funciones anteriores. Luego hay que sacar algunos episodios si fuera necesario, para ello usar la nota mental del punto 2 del pre procesado. El archivo final debe lucir así.

id	periodo	hora	estado	actividad	dur_min	mean_act_min	num_epi	epi_estado
10013	Dia 01	2016-08-03 07:35:00	W	23239	85	273.4	4	W001
10013	Dia 01	2016-08-03 09:00:00	S	1951	180	10.8	5	S001
10013	Dia 01	2016-08-03 12:00:00	W	2324	17	136.7	6	W002
10013	Dia 01	2016-08-03 12:17:00	S	2	20	0.1	7	S002
10013	Dia 01	2016-08-03 12:37:00	W	52688	213	247.4	8	W003
10013	Dia 01	2016-08-03 16:10:00	S	632	21	30.1	9	S003

Pre procesado

La tabla EPI del `Actividorm` al igual que la de la app nueva no se puede usar directamente para calcular las estadísticas de los períodos, debido a que se hace un pre-procesado que asegura la uniformidad de los registros para que luego no se produzcan errores. Las etapas de este proceso son las siguientes:

El pre-proceso se ejecuta sobre la tabla EPI ya depurada en el caso de tratar con archivos antiguos o directamente con la app nueva.

La función es: `function_validEvents()` que ya documentaré

Todo lo que se descarte queda en la tabla drop acompañado del id, periodo y motivo.

Luego de pasar la función se deben extraer las tablas del objeto lista que produce esta función. Además de remover la variable **actividad**. Se puede usar este código como guía:

```
# Pre procesar
epi <- function_validEvents(epi)
drop <- epi$drop
epi <- epi$datos

# Validar que se pueda analizar
epi <- select(epi, -actividad)
check.epidata(epi)
```

A continuación el detalle de lo que hace esta función.

Descartar el Dia 01

Como este día de registro no es completo porque no inicia desde el mínimo posible no se utiliza y en esta parte del procesado se descarta. Tanto la app nueva como `Actividorm` descartan este período para el cálculo de estadísticas.

Periodos repetidos

En la app nueva no van a existir periodos repetidos porque el análisis se realiza una sola vez, pero cuando se analizan tablas EPI del `Actividorm` puede ocurrir que se reprocese un archivo AWD y existan 2 archivos y al compilar todo pueden quedar episodios repetidos. En esta etapa se descarta lo repetido.

Nota mental:

En ciertas ocasiones puede ser que un sujeto tenga un falso inicio de sueño y por lo tanto noche, esto se detecta en el actograma, lo que ocurre es que se retrasa esa noche. En `Actividorm` esto se hace a mano y al procesar tablas EPI con este sistema obliga a eliminar los episodios previos al real inicio de la noche.

Lo primero es que al combinar la tabla EPI con la tabla ARQ para descartar los períodos con errores hay que asegurarse de traerse la hora del episodio en ARQ. Así en R creando una variable lógica (TRUE o FALSE) se pueden descartar fácilmente estos episodios con error.

```
epi <- arrange(epi, id, periodo, hora)
epi <- mutate(epi, edited = hora < hora.arq)
epi <- filter(epi, edited == FALSE)
```

Esto es usando la librería `dplyr` y lo que hace es ordenar por id, periodo y luego por hora, luego crea variable `edited` con verdadero si el episodio es menor que la hora del ARQ (la correcta) y luego nos quedamos con lo que sea mayor o igual a esa hora.

Crear periodos

Originalmente los periodos vienen con el día y el secuencial unido, pero en ocasiones se necesita calcular cosas solo para día o noche, por ello se separa, pero no se descarta la variable.

Las variables nuevas son:

- `dia.noc` con valores día o noche
- `seq.dia` con el correlativo

Mínimo de episodios

La para hacer cálculos y estadísticas se requiere un mínimo de episodios en un período, si una persona duerme toda la noche no se le puede calcular nada.

Originalmente se descartaban periodos con 1 episodio (no pueden haber 2) y solo se usaban 3 o más. En la última actualización se usan todo para que en caso que tenga solo un episodio se le pueda calcular horas de inicio y fin.

Se recomienda descartar este tipo de episodios cuando se hacen análisis porque tendrán estadísticas de 0% y 100% y esto sesga un montón los promedios. Obvio no olvidar reportar para incluir en los resultados.

El código para descartar periodos de 1 episodio quedó comentado en la línea 91 del script de la función. De momento sólo se registra la ocurrencia, pero no se descarta nada.

Primer episodio

Lo siguiente que se chequea, por precaución es que un el primer episodio de un período sea concordante.

- Si es Noche debe comenzar por Sueño
- Si es Dia debe comenzar por Vigilia

Antes generaba un error y descartaba el período ahora solo se guarda, por eso es bueno revisar esa tabla de cuando en vez.

Días de la semana

Otro aspecto que no tiene el `Actividorm` es la definición del día. Dado que existen diferentes usos para el día de la semana o tal vez se quisiera hacer alguna combinación rara, se registra el día de la semana a la cual pertenece el periodo o el transito en el caso de la noche.

Día del período

El día se define como el la fecha-hora del primer episodio de vigilia que ocurre igual o después de cierta hora (06:00 por ejemplo) y que tenga una duración determinada (mayor a 30 min por ejemplo).

Considerando esto el **día de la semana del período se extrae de la fecha del 1er episodio del día**. Por ejemplo el `25-04-2011` es Lunes, por lo tanto si hay un período día que inicie ese día será etiquetado como **Lunes**.

Noche del período

De la misma forma la noche se define cuando se encuentra un episodio de sueño posterior a cierta hora (*i.e.* 20:00) y que tenga una duración superior a un tiempo determinado (30 min por ejemplo).

El tiempo no es una variable circular y una vez que llega a las 23:59 técnicamente no pasa a 00:00 sino que a 24:00, es decir al día siguiente, por esta razón en un período de noche pueden ocurrir en 2 combinaciones.

- Comenzar en una fecha y terminar un día después.
- Comenzar y terminar en una misma fecha

Por ejemplo un período de noche puede iniciar el `26-04-2011 22:46` y terminar el `27-04-2011 07:23` o sea transita desde un martes a un miércoles.

Otra opción sería que comience el `27-04-2011 00:46` y terminar el `27-04-2011 07:23` o sea inicio y término en la misma fecha, aun así se etiquetará como de martes a miércoles.

Etiquetando de esta forma cada período se puede tener una idea muy clara para poder asignar cada período según se necesite.

Redondear duración

Finalmente decir que se usa una función de expansión de la data, se actualizó en algunos script pero en los más complicados como las estadísticas de duración se dejó aunque sea lento. Algún día lo veré...

Para asegurar el cálculo y aún cuando lo más probable es que la duración de un episodio sea un número entero (minutos) se redondea al valor más bajo, o sea 25.3 será 25.

Se hizo esto porque algún registro traía un decimal. Pero el 99.9% de las veces viene un número entero.

1. Hora de Inicio

Esta sección calcula la hora de inicio y variables asociadas del período de análisis. Los períodos de análisis pueden ser de noche o día porque tienen la misma estructura. Por ejemplo este extracto es la **Noche 02** del sujeto **10648** y contiene 3 episodios, 2 de sueño y 1 de vigilia.

id	fec.hora	estado	dur_min	mean_act	actividad	hora	dia	hora.abs	periodo
10648	2016-09-14 22:18:00	S	494	6.5	3231	22.300	mié-jue	22	Noche 02
10648	2016-09-15 06:32:00	W	42	67.3	2826	6.533	mié-jue	6	Noche 02
10648	2016-09-15 07:14:00	S	31	9.0	278	7.233	mié-jue	7	Noche 02

Variables de Inicio

A partir de un período como este se calculan las siguientes variables.

Variable	Descripción
id	Id del sujeto
period	Periodo de análisis
día_noc	Día o noche correspondiente al período
stage_ini	Estado de Sueño o Vigilia del episodio de inicio
hi	Hora de inicio del período (decimal)
hi_abs	Hora de inicio del período (Valor absoluto)
hi_m	Hora de la mitad del episodio de inicio
hf	Hora de fin del período (Decimal)
hf2	Hora de fin del período (Formato continuo)
wday	Día de la semana de la fecha (formato transito para la noche)
fecha	Fecha del episodio de inicio (para chequeos)
key	Para cruzar bases de datos (identifica id + periodo)

La hora de la mitad de la noche se calcula como se explica en el apartado de conteos.

Errores detectados

Este es un ejemplo, podría ser noche también.

Al revisar los datos en el proyecto de las leches en 6 meses (por ejemplo) se encontraron horas de inicio anómalas que el `Actividorm` determinó como inicio del día. El programa define la detección del día así:

El día se inicia al encontrar un período de vigilia luego de la hora indicada en `AwakeSearchTime` que dure al menos `AwakeDuration` segundos y que tenga menos de `AwakeMaximumSleep` segundos de sueño

Esto implica que no debieran existir días que inicien antes de las 06:00 (según configuración). Por ejemplo el `id = 10615` en el `Día 05` inicia a las 03:43, el extracto del archivo EPI es el siguiente:

id	fec.hora	estado	dur_min	dia.noc	seq.dia	hora
10615	2016-10-03 03:43:00	W	184	Dia	05	3.717
10615	2016-10-03 06:48:00	S	254	Dia	05	6.8
10615	2016-10-03 11:02:00	W	367	Dia	05	11.033
10615	2016-10-03 17:10:00	S	11	Dia	05	17.167
10615	2016-10-03 17:22:00	W	84	Dia	05	17.367
10615	2016-10-03 18:46:00	S	93	Dia	05	18.767
10615	2016-10-03 20:19:00	W	134	Dia	05	20.317

Según la configuración: La primera vigilia que ocurre después de las 06:00 que dura al menos 1800 segundos (30 min) y que tenga menos de 300 segundos de sueño (imagino que según estado actigráfico de MiniMitter) corresponde a la vigilia de las 11:02. Algo pasó entre medio que llevó al programa a hacer esto, lamentablemente no tengo como revisar el código.

Nota mental:

En la app nueva no se considera el parámetro de *AwakeMaximumSleep* (o de sueño), porque es redundante e induce error.

Al usar el parámetro *StateFilterDuration* para definir un episodio de sueño o vigilia consolidado se descarta de antemano aquellos epoch detectados como sueño por el algoritmo de MiniMitter. O sea, si en un episodio determinado como vigilia tiene dentro una secuencia `W W S S S W W` esos epoch de sueño no serán considerados como cambio de estado porque no son suficientes, esto se hace en `Actividorm` como en la app nueva, es decir, hay una regla para determinar cuando cambia un estado.

Por otra parte usar este parámetro es problemático porque correlaciona con la duración del episodio. Es decir, un episodio de vigilia (por ejemplo) tendrá más epoch de sueño según el algoritmo de MiniMitter mientras mayor sea la duración del episodio. La consecuencia puede ser (por ejemplo) **descartar un una vigilia que defina el inicio de día** porque supera el parámetro de cantidad de sueño MiniMitter.

Como la cantidad de sueño (para el ejemplo) MiniMitter depende de la duración del episodio puede ocurrir que si un sujeto estuvo despierto todo el día y de cuando en vez no se mueve y el algoritmo dice sueño y al sumar estos epoch habrá mucho sueño y si se supera este parámetro no habrá inicio de día. Pero estuvo despierto todo el día (según actograma) y se perdería en el recuento final y eso ocurre a veces con

`Actividorm`.

2. Conteo de episodios

Esta sección se ocupa de contar los episodios del período y calcular su porcentaje. Los períodos de análisis pueden ser de noche o día porque tienen la misma estructura. Por ejemplo este extracto es la **Noche 02** del sujeto **10648** y contiene 3 episodios, 2 de sueño y 1 de vigilia.

id	fec.hora	estado	dur_min	mean_act	actividad	hora	dia	hora.abs	periodo
10648	2016-09-14 22:18:00	S	494	6.5	3231	22.300	mié-jue	22	Noche 02
10648	2016-09-15 06:32:00	W	42	67.3	2826	6.533	mié-jue	6	Noche 02
10648	2016-09-15 07:14:00	S	31	9.0	278	7.233	mié-jue	7	Noche 02

Conteos

Lo siguiente son los conteos generales y sus porcentajes tienen la siguiente estructura y se usa para la misma para las mitades. Tenemos:

Variable	Descripción
nS	Número de episodios de Sueño
nW	Número de episodios de Vigilia (Wakeness)
nTot	Número Total de episodios
pS	Porcentaje de episodios de sueño sobre el total
pW	Porcentaje de episodios de Vigilia sobre el total

Existe una excepción: en caso que el sujeto sólo tenga un episodio en el período, en la noche por ejemplo: porque durmió de corrido todo el período. Igual se van a capturar sus estadísticas de conteos porque aunque son escasas las ocurrencias puede ser necesario describir esta situación. Pero **se recomienda filtrarlas antes de hacer análisis** para que no afecte los cálculos promediados ya que valores de 0 y 100 en el porcentaje va a subestimar y sobreestimar los promedios de forma dramática y no es recomendable si se quiere obtener estadísticas representativas de los períodos que sí tienen datos válidos.

Conteos en Mitades

Cuando se divide un período y se cuentan episodios en tales mitades hay que considerar algunas cosas, si bien el proceso de contar es el mismo se deben definir varias cosas primero:

Calcular la mitad del período

Lo primero es calcular la mitad del período, como se cuenta con la hora de inicio de cada episodio y su duración se puede calcular de esta forma. Se entiende como fecha-hora al constructo de tipo `30-06-2020 23:59:00`.

$\text{Fecha.Hora.Inicio} = \text{Fecha.Hora (Primer episodio)}$

$\text{Fecha.Hora.Final} = \text{Fecha.Hora (último episodio)} + \text{duración (último episodio)}$

$\text{Fecha.Hora.Mitad} = \text{Fecha.Hora.Inicio} + \text{minutos}((\text{Fecha.Hora.Final} - \text{Fecha.Hora.Inicio})/2)$

Según esa `Fecha.Hora.Mitad` se asignarán los diferentes episodios a cada mitad de la **noche o día** según corresponda.

Importante a considerar: Cuando se habla de mitades se asume que existen ocurrencias en ambas mitades, una persona no puede tener una sola mitad, siempre tendrá 2 y por esa razón un sujeto que duerme de corrido (o no duerme siestas) no será incluida en el cálculo de mitades de noche o día.

La razón es la siguiente.

- Si asigno ese único episodio a la 1ra mitad, la segunda quedará en blanco y no se puede tener una sola mitad.
- Si asigno uno en cada mitad tendría entonces 2 episodios, no uno.
- Además no se puede tener 2 episodios iguales seguidos, 2 episodios de sueño tienen que estar separados por una vigilia y viceversa.
- De lo anterior se desprende que la cantidad de episodios mínima para hacer cálculos es de 3 episodios en un período.

Un caso a considerar: Se podría pensar que alguien que duerme algo en la noche y se desvela podría tener 2 episodios en una noche, sin embargo el `actividorm` (y la app nueva) determina que el día comienza cuando encuentra la primera vigilia que dure **x** minutos y que sea posterior a **x** hora.

Entonces si se desveló y pasó la hora que determina el inicio del día, aun se debe localizar el episodio que inicia el día, por lo tanto el sistema comenzará el día en la siguiente vigilia que dure más de **x** minutos y por lo tanto deberá existir un sueño antes de ese episodio y por eso un período de 2 episodios no puede ocurrir. Lo mismo ocurre en la noche.

Cuando un sujeto se duerme y despierta antes de la hora de inicio del día y no duerme hasta la noche el `actividorm` se vuelve loco (noches que duran hasta las 7PM y cosas así.), en el sistema nuevo cuando pasa esto se fuerza que el inicio del día sea a la hora configurada (o marcada en la app) para asegurar la utilidad de ese día y dar cierre a esa noche.

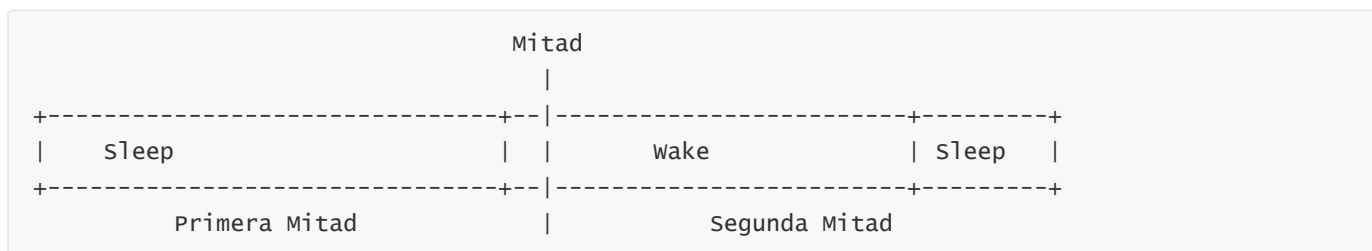
Gráficamente el inicio del día luce así:



No se puede adaptar el `actividorm` a lo que hace la app nueva porque el archivo de episodios ya está creado y en la app nueva esta distinción se hace antes de la creación del archivo de episodios. Ocurre que este archivo de episodios de sueño y vigilia es la clave para calcular las estadísticas generales, es el origen de todo (¬_¬).

Asignación episodio que cruza la mitad

Aunque puede ocurrir que un episodio comience justo en la `Fecha.Hora` de la mitad de la noche la mayoría de las veces existirá un episodio que cruza tal hora. Existen dos formas de asignar ese episodio a una mitad de período, por ejemplo en el siguiente diagrama:



Versión 1. Según hora

En esta variante los episodios se asignan a cada mitad **según la hora en que comienza el episodio**. Da igual que la mayor parte de un episodio que cruza la hora de la mitad del período se encuentre en la segunda parte.

Por ejemplo en el diagrama el episodio de vigilia a pesar de distribuirse principalmente en la segunda mitad será asignado a la primera mitad porque se inicia en esa parte.

- Entonces en este caso la primera mitad tendrá 2 episodios uno de sueño y otro de vigilia
- La segunda mitad sólo tendrá un episodio de sueño y ausencia de vigilia.

Como una persona no puede tener datos en una sola mitad **existe una excepción a esta regla**: Si ocurriera que todos los episodios se inician en la 1ra mitad significa que el último abarca toda la segunda mitad, razón por la cual será asignado a la segunda mitad.

La variable de versión 1 se indican con el sufijo `v1`

Versión 2. Según proporción

Esta variante asigna el episodio que cruza a la `Fecha.Hora` de la mitad del período según en qué mitad tiene mayor proporción. De esta forma se evita la excepción de la **Versión 1**.

Las variables de versión 2 se indican con el sufijo **v2**

Actividorm usa la versión 2 para signar el episodio que cruza la mitad

Cálculos

Los cálculos entonces son los mismos que para los conteos globales, solo que se aplican a los episodios de cada mitad.

Nota mental:

Mientras se reescribió el script de conteos encontré que si a la hora de inicio de un episodio se le suma la duración en minutos, el final no calza con el inicio del siguiente según los datos del **Actividorm**. (-_-)

Por ejemplo si tengo un episodio con hora **10:20** y duración **8** el episodio termina entonces a las **10:28:59** entonces el siguiente episodio tiene que comenzar a las **10:29**, pero a veces nup.

Este error del **Actividorm** provoca que no se puedan asignar correctamente los episodios a cada mitad, razón por la cual hubo que recalcular la duración de los episodios.

En la app nueva esto no ocurre, *"todo calza"* 

Variables de Conteos

Los códigos de las variables son las siguientes. Obvio que las divisiones son solo para orden mental.

Variables	Descripción	

id	Id del sujeto	
día.noc	Día o Noche d	
periodo	Período de análisis	
wday	Día de la semana de la fecha (formato transito para la noche)	

nS	N° de episodios de Sueño del período	
nW	N° de episodios de vigilia del período	
nTot	N° Total de episodios del período	
pS	Porcentaje de períodos de sueño sobre el total	
pW	Porcentaje de períodos de vigilia sobre el total	

nS_M1v1	N° de episodios de Sueño	Mitad 1, Version 1
nW_M1v1	N° de episodios de vigilia	Mitad 1, Version 1
nTot_M1v1	N° Total de episodios	Mitad 1, Version 1
pS_M1v1	Porcentaje de períodos de sueño	Mitad 1, Version 1
pW_M1v1	Porcentaje de períodos de vigilia	Mitad 1, Version 1

nS_M2v1	N° de episodios de Sueño	Mitad 2, Version 1
nW_M2v1	N° de episodios de vigilia	Mitad 2, Version 1

nTot_M2v1	N° Total de episodios	Mitad 2, Version 1
pS_M2v1	Porcentaje de períodos de sueño	Mitad 2, Version 1
pW_M2v1	Porcentaje de períodos de vigilia	Mitad 2, Version 1

nS_M1v2	N° de episodios de Sueño	Mitad 1, Version 2
nW_M1v2	N° de episodios de Vigilia	Mitad 1, Version 2
nTot_M1v2	N° Total de episodios	Mitad 1, Version 2
pS_M1v2	Porcentaje de períodos de sueño	Mitad 1, Version 2
pW_M1v2	Porcentaje de períodos de vigilia	Mitad 1, Version 2

nS_M2v2	N° de episodios de Sueño	Mitad 2, Version 2
nW_M2v2	N° de episodios de Vigilia	Mitad 2, Version 2
nTot_M2v2	N° Total de episodios	Mitad 2, Version 2
pS_M2v2	Porcentaje de períodos de sueño	Mitad 2, Version 2
pW_M2v2	Porcentaje de períodos de vigilia	Mitad 2, Version 2

key	Para cruzar bases de datos (identifica id + periodo)	

3. Duración de estados

Esta sección calcula la duración en minutos de los episodios de sueño y vigilia además del porcentaje de los episodios. Por ejemplo este extracto es la **Noche 02** del sujeto **10648** y contiene 3 episodios, 2 de sueño y 1 de vigilia.

id	fec.hora	estado	dur_min	mean_act	actividad	hora	dia	hora.abs	periodo
10648	2016-09-14 22:18:00	S	494	6.5	3231	22.300	mié-jue	22	Noche 02
10648	2016-09-15 06:32:00	W	42	67.3	2826	6.533	mié-jue	6	Noche 02
10648	2016-09-15 07:14:00	S	31	9.0	278	7.233	mié-jue	7	Noche 02

Para hacer el cálculo se usa la variable de duración del episodio `dur_min` (en minutos), en el caso del `Actividorm` que tiene errores en la asignación de esta variable pueden existir mínimas discrepancias si se hace al cálculo a mano (Ya lo expliqué).

Cabe destacar que la función de cálculo es un poco lenta debido a que usa una estrategia de expansión de la data porque fue creada durante el proceso de análisis de actividad. Se podría re escribir, pero funciona y no se modificó su estructura.

Duraciones generales

Este es un ejemplo de un bloque de duraciones globales.

Stime	Wtime	Ttime	Spct	Wpct
525	42	567	92.6	7.4

Las variables `Stime`, `Wtime` y el `Ttime` se encuentra en minutos. `Spct` y `Wpct` corresponden al porcentaje de la duración sobre el total. Es lo mismo para calcular duraciones en mitades y tercios.

Duraciones en Mitad y Tercios

Además de las duraciones totales se particionó la noche de cada sujeto en mitades y tercios. A estos segmentos se le aplicaron las mismas variables que en las duraciones generales.

En el caso de las duraciones no se asignan episodios a mitades o tercios sino que se calcula de forma precisa cuántos minutos hay en un tercio o mitad de noche.

Para hacer los cálculos de mitades y tercios se toma la duración de la noche o día y se particiona según el procedimiento explicado en los conteos. Luego con estas horas de corte se calcula la cantidad precisa de minutos existentes en cada partición.

Promedios

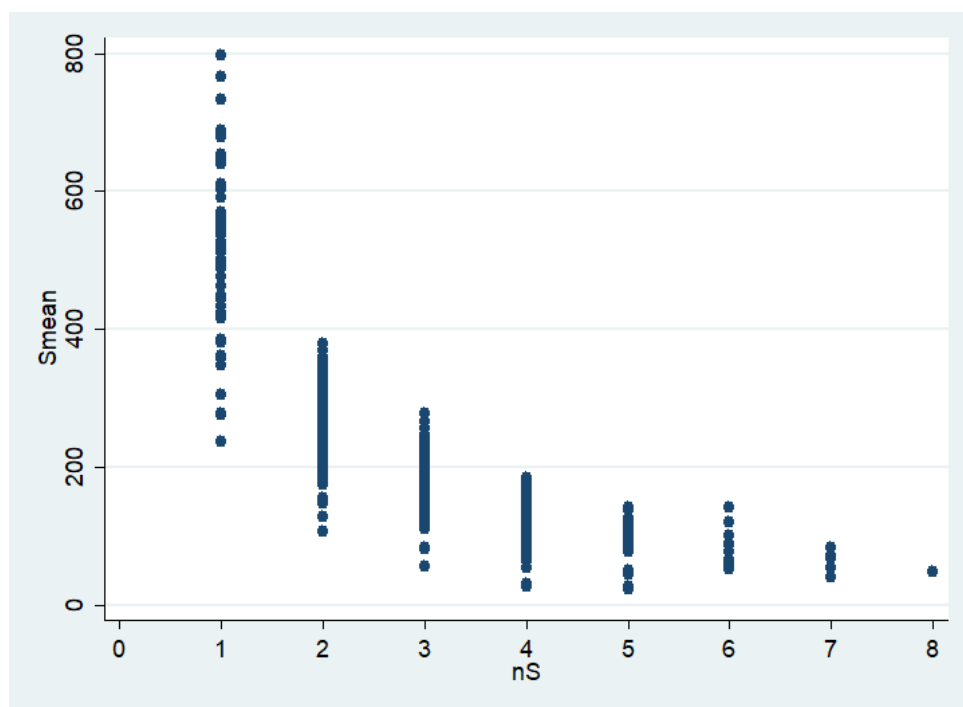
En esta sección se calculan los valores promedios de los episodios de sueño y vigilia para los episodios, no se hace el cálculo en mitades o tercios por la baja frecuencia de episodios que suele observarse.

A pesar que en términos de la estadística más básica posible calcular el promedio de una constante (un solo valor) es, por decir lo menos, absurdo... igual se hace, porque `Actividorm` lo hace `mean()`.

Dicho esto **se recomienda encarecidamente no considerar promedios basados en una observación**, o sea no usarlos para análisis. Es más, no recomiendo usar esta variable porque es muy inestable y no refleja en absoluto lo que pretende, por ejemplo:

Si un sujeto duerme toda la noche, unos 450 minutos, se despierta 10 minutos y luego duerme otros 10 (esto es bien común por cierto) el promedio calculado será $450+10/2 = 230$ minutos.

O sea se tiende a pensar que el sujeto durmió un par de episodios de 230 minutos, pero no es así.



Si graficamos N° de Sueños vs Promedio de sueños (en la noche) como se ve aquí (en wawas de 6 meses) vemos que la mayoría de los promedios se basan en 1 observación, quizás 2 o 3. Entonces con juicio estadístico diremos que esta variable no tiene valor analítico, es decir, lo que se saque de esta variable no significa nada. Dicho de otra forma, afirmar que el promedio de los episodios de sueño de un grupo u otro no solo será pretencioso, sino quizás equivocado.

Otra cosa sería calcular el promedio ponderado a la duración, pero eso será para una próxima versión.

Variables de duración

Los códigos de las variables son las siguientes. Obvio que las divisiones son solo para orden mental.

Variable	Descripción
id	Id del sujeto
dia.noc	Dia o noche del período
periodo	Período en análisis
wday	Dia de la semana de la fecha (formato transito para la noche)
nS	N° de episodios de sueño del período
nW	N° de episodios de vigilia del período
nTot	N° de episodios del período
Stime	Suma duración sueño período
Wtime	Suma duración vigilia períodos
Ttime	Suma total del período (duración período)
Spct	Porcentaje duración sueño sobre total
Wpct	Porcentaje duración vigilia sobre total

Smean	Promedio duración episodios de sueño	
Wmean	Promedio duración episodios de vigilia	
Tmean	Promedio duración episodios el período	

T1S	Duración sueño	Tercio 1
T1W	Duración vigilia	Tercio 1
T2S	Duración Sueño	Tercio 2
T2W	Duración Vigilia	Tercio 2
T3S	Duración Sueño	Tercio 3
T3W	Duración Vigilia	Tercio 3
T1tot	Duración del tercio 1	
T1Sp	Porcentaje Sueño	Tercio 1
T1Wp	Porcentaje Vigilia	Tercio 1
T2tot	Duración del tercio 2	
T2Sp	Porcentaje Sueño	Tercio 2
T2Wp	Porcentaje Vigilia	Tercio 2
T3tot	Duración del tercio 3	
T3Sp	Porcentaje Sueño	Tercio 3
T3Wp	Porcentaje Vigilia	Tercio 3

M1S	Duración sueño	Mitad 1
M1W	Duración vigilia	Mitad 1
M2S	Duración sueño	Mitad 2
M2W	Duración vigilia	Mitad 2
M1tot	Duración mitad 1	
M1Sp	Porcentaje sueño	Mitad 1
M1Wp	Porcentaje vigilia	Mitad 1
M2tot	Duración mitad 2	
M2Sp	Porcentaje sueño	Mitad 2
M2Wp	Porcentaje Vigilia	Mitad 2

key	Para cruzar bases de datos (identifica id + periodo)	

4. Duración máxima

Esta parte calcula o busca aquellos episodios que representan la máxima duración de un período. Por ejemplo este extracto es la **Noche 02** del sujeto **10648** y contiene 3 episodios, 2 de sueño y 1 de vigilia.

id	fec.hora	estado	dur_min	mean_act	actividad	hora	dia	hora.abs	periodo
10648	2016-09-14 22:18:00	S	494	6.5	3231	22.300	mié-jue	22	Noche 02
10648	2016-09-15 06:32:00	W	42	67.3	2826	6.533	mié-jue	6	Noche 02
10648	2016-09-15 07:14:00	S	31	9.0	278	7.233	mié-jue	7	Noche 02

Lo primero a tener en consideración es que un **máximo implica que si o si un período tiene al menos existen 2 episodios** para poder calcular.

No existe valor máximo para una constante, por ejemplo: El valor máximo de 10 no existe.

Esto limita el cálculo de esta estadística a aquellos periodos (u otro segmento) que posea al menos 2 episodios del mismo tipo, por ejemplo:

Secuencia Noche	Secuencia Día	Status estadística duración máxima
S	W	No permite calcular duración máxima
S - W - S	W - S - W	Permite el cálculo de sólo 1 máximo
S - W - S - W - S	W - S - W - S - W	A partir de cinco episodios se pueden calcular máximos a los dos estados

Esto opera para cualquier partición de la noche o el día, es decir, si se desea conocer cuál es el episodio de sueño más largo de la segunda mitad de la noche se requiere que tenga al menos 3 episodios con la secuencia **S - W - S**. Lo mismo para el día.

Cuando no se pueda calcular se dejará el valor correspondiente como valor faltante **NA**.

Máximos en mitades

Dada la certeza anterior para que se pueda calcular máximos en mitades de noche o día (es igual el cálculo) se requiere un mínimo de 3 episodios y por lo tanto **asignar los episodios a mitades**. Para esto se utilizará la estrategia de asignar los episodios según en qué mitad tengan la mayor parte de su duración, versión 2 en la sección conteos.

Variables de máximos

Variable	Descripción
id	Id del sujeto
dia.noc	Día o noche del período
periodo	Período en análisis
wday	Día de la semana de la fecha (formato transito para la noche)
durSmax	Duración sueño máximo del período
locSmax	Localización sueño máximo del período (Según mitad)
midSmax	Punto medio (hora decimal) del sueño más largo del período
durWmax	Duración vigilia máximo del período
locWmax	Localización vigilia máxima del período (Según mitad)
midWmax	Punto medio (hora decimal) de vigilia más larga del período

durSmax_M1	Duración sueño máximo	Mitad 1
midSmax_M1	Punto medio (decimal) del sueño máximo	Mitad 1
durWmax_M1	Duración vigilia máxima	Mitad 1
midWmax_M1	Punto medio (decimal) vigilia máxima	Mitad 1
durSmax_M2	Duración sueño máximo	Mitad 2
midSmax_M2	Punto medio (decimal) del sueño máximo	Mitad 2
durWmax_M2	Duración vigilia máxima	Mitad 2
midWmax_M2	Punto medio (decimal) vigilia máxima	Mitad 2

key Para cruzar bases de datos (identifica id + periodo)

Queda pendiente incluir máximos por mitades asignando episodios según hora de inicio.

5. Latencias

Para el cálculo de latencias es el mismo procedimiento de selección del período. Por ejemplo este extracto es la **Noche 02** del sujeto **10648** y contiene 3 episodios, 2 de sueño y 1 de vigilia.

id	fec.hora	estado	dur_min	mean_act	actividad	hora	dia	hora.abs	periodo
10648	2016-09-14 22:18:00	S	494	6.5	3231	22.300	mié-jue	22	Noche 02
10648	2016-09-15 06:32:00	W	42	67.3	2826	6.533	mié-jue	6	Noche 02
10648	2016-09-15 07:14:00	S	31	9.0	278	7.233	mié-jue	7	Noche 02

A diferencia del resto de los script de cálculo en esto de las latencias hay distinciones que hacer entre día y noche, por ejemplo la latencia a la primera siesta y a la primera vigilia es la misma variable dependiendo si el período es día o noche.

Latencias del inicio

Lo primero a registrar son las latencias que ocurren al inicio del período hasta el segundo despertar en el caso de la noche o segundo sueño en caso que el período sea día (es la misma variable).

Demás está decir que si el sujeto sólo tiene un estado porque duerme de corrido (por ejemplo) no se puede calcular nada, por lo cual solo se registrará valores faltantes. Las duraciones se calculan con la variable `dur_min`.

Se registra la hora del episodio siguiente como latencia para asegurar que exista un próximo episodio. Por ejemplo en la noche.

```

/  ----- \      \      \      \
|      S      | Dur.1ra.Lat |      |
|  ----- < Hr.1ra.Lat -- /      | Dur.2da.Lat |
|      W      |      |      | Dur.3ra.Lat
|  ----- < Hr.2da.Lat ----- /      |
Noche |      S      |      |
|  ----- < Hr.3ra.Lat ----- /
|      W
|  -----
|      S
\  -----

```

Para el día funciona igual pero en lugar de sueño es vigilia.

```

/ ----- \
|      w      | Dur.1ra.Lat |
| ----- < Hr.1ra.Lat -- /      | Dur.2da.Lat |
|      s      |              | Dur.3ra.Lat
| ----- < Hr.2da.Lat ----- /      |
Día |      w      |
| ----- < Hr.3ra.Lat ----- /
|      s
| -----
|      w
\ -----

```

Para registrar la hora y duración de la primera latencia se necesita que existan al menos 3 episodios y para el resto de variables al menos 5 episodios. Si no hay episodios suficiente se rellena con valores faltantes.

Latencia del final

También se registra la hora de inicio y duración del último episodio del período, porque en el caso del día corresponde a la latencia de dormir. Cuyo efecto se puede medir si se cuenta con el período consecutivo siguiente.

Variables Latencia

Variable	Descripción
id	Id del sujeto
dia.noc	Día o noche del período
periodo	Período en análisis
wday	Día de la semana de la fecha (formato transito para la noche)
n_epi	Número de episodios del período

lat_date	Fecha del primer episodio (para chequeos)
lat1_hora	Hora 1ra latencia (Hora inicio 2do episodio)
lat1_dur	Duración 1ra latencia (Duración 1er episodio)
lat2_hora	Hora 2da latencia (Hora inicio 3er episodio)
lat2_dur	Duración 2da latencia (Duración 1er + 2do episodio)
lat3_hora	Hora 3ra latencia (Hora inicio 4to episodio)
lat3_dur	Duración 3ra latencia (Duración 1er + 2do + 3er episodio)

durEpi2	Duración 2do episodio
durEpi3	Duración 3er episodio
latU_hora	Hora de inicio último episodio (latencia dormir si es día)
latU_dur	Duración último episodio

key	Para cruzar bases de datos (identifica id + periodo)
-----	--

Como puede adivinarse:

- El primer episodio es Vigilia si el período es día
- El primer episodio es Sueño si el período es noche

Y así sucesivamente 🔄

Por ejemplo: la variable **latency to second nap since end of first nap** sería la duración del 3er episodio cuando el período es día 😊

6. Día completo

Si una persona usara de forma continua el actígrafo desde que sale del laboratorio hasta que vuelve no se necesitarían ediciones manuales, app nueva y el **Actividorm** funcionaría perfecto.

Como no es el caso se deben borrar de forma rutinaria períodos completos, dejando lagunas en los registros que dificultan el cálculo de estadísticas de día completo. Para poder tener acceso a este tipo de análisis lo primero es buscar en todos los períodos de un sujeto cuales son consecutivos.

Noche a Día: Este tipo de registro de 24 horas comienza con la noche y en términos del registro secuencial de periodos necesita de la existencia de pares de períodos de tipo **Noche 03 & Día 03** .

Día a Noche: Este tipo de registro de 24 horas inicia en el día y termina al día siguiente (al término de la noche), en términos de las secuencias de **Actividorm** o la app nueva son del tipo **Día 03 & Noche 04** .

Entonces para cada sujeto hay que buscar en sus registros válidos cuantos de estos pares existen.

Períodos consecutivos

Cuando se encuentren pares de períodos que sirvan para hacer cálculos de día completo la unidad de análisis ahora es un par de períodos y en 2 versiones **Noche a Día** y **Día a Noche**.

Una vez identificadas las combinaciones que tiene el sujeto se tendrá esta estructura.

id	tipo	combi	filtro1	filtro2
10207	Dia a Noche	Dia 02 - Noche 03	Dia 02	Noche 03
10207	Dia a Noche	Dia 05 - Noche 06	Dia 05	Noche 06
10207	Dia a Noche	Dia 06 - Noche 07	Dia 06	Noche 07
10207	Dia a Noche	Dia 07 - Noche 08	Dia 07	Noche 08
10207	Noche a Dia	Noche 06 - Dia 06	Noche 06	Dia 06
10207	Noche a Dia	Noche 07 - Dia 07	Noche 07	Dia 07

Entonces con las variables `filtro1` y `filtro2` se filtran los episodios del par de períodos y se calculan algunas estadísticas.

Sujetos sin pares válidos

Existe la posibilidad que existan sujetos que no tienen pares de períodos consecutivos y en consecuencia no se les puede calcular nada.

Se registra esta ocurrencia en la tabla **drop** que contienen observaciones que se han descartado. Así se pueden realizar chequeos.

Variables día completo

Variable	Descripción

id	Id del sujeto
tipo	Tipo de par (Dia a noche o Noche a dia)
filtro1	Primer elemento del par
filtro2	Segundo elemento del par
combi	Combinación del par (con número identificador)
fecini	Fecha de inicio del primer par de períodos
dias	Días involucrados en el par

nS_24	N° de episodios de sueño
nW_24	N° de episodios de vigilia
nTot_24	N° Total de episodios
pS_24	Porcentaje episodios de sueño sobre total
pW_24	Porcentaje episodios de vigilia sobre el total

Sdur_24	Duración del sueño
Wdur_24	Duración de la vigilia
Tdur_24	Duración total del par de períodos

```

Spct_24      Porcentaje de duración de sueño
Wpct_24      Porcentaje de duración de vigilia
-----
inISMax_24   Hora inicio episodio sueño máximo en formato HH:MM (Para chequeos)
inISMax2_24  Hora inicio episodio sueño máximo (decimal)
midSMax_24   Hora mitad episodio sueño máximo (decimal)
perSmax_24   Período al cual pertenece el episodio de sueño máximo
dursMax_24   Duración del episodio de sueño máximo
-----
iniWMax_24   Hora inicio episodio vigilia máxima en formato HH:MM (Para chequeos)
iniWMax2_24  Hora inicio episodio vigilia máxima (decimal)
midWMax_24   Hora mitad episodio vigilia máxima (decimal)
perWmax_24   Período al cual pertenece el episodio de vigilia máxima
durWmax_24   Duración del episodio de vigilia máxima

```

Como es lógico esta sección no contiene la `key` para cruzar con otras bases de datos porque cada registro corresponde a estadísticas de 2 períodos.

7. Causa efecto

Este tipo de resultado no son cálculos de estadísticas propiamente tales, corresponde a los pares de períodos válidos que tiene cada sujetos, ya están disponibles en las estadísticas de día completo.

Se dejan aparte con el objeto de poder filtrar pares de variables de las otras tablas de datos y que pudieran tener un valor predictor en el ambiente de 24 horas. Tiene la siguiente estructura:

id	tipo	combi	filtro1	filtro2
10013	Día a Noche	Día 02 - Noche 03	Día 02	Noche 03
10013	Día a Noche	Día 03 - Noche 04	Día 03	Noche 04
10013	Día a Noche	Día 04 - Noche 05	Día 04	Noche 05
10013	Noche a Día	Noche 02 - Día 02	Noche 02	Día 02
10013	Noche a Día	Noche 03 - Día 03	Noche 03	Día 03
10013	Noche a Día	Noche 04 - Día 04	Noche 04	Día 04

Un ejemplo

Un par de variables que pudiera ser filtrado a partir de estos registros es (usando la primera línea) por ejemplo:

```

id = 10013
filtro1 = Día 02
filtro2 = Noche 03
Tabla conteos: [N° Episodios Sueño]
Tabla duración: [Duración sueño nocturno]

```

De esa forma se podrán hacer análisis más elaborado, pero como las combinaciones son demasiadas no se pueden calcular todas, pero se podría programar un `loop` o algo parecido para cada fila de esta tabla de pares.

El resultado será para cada sujeto un set de pares de variable, según tenga pares disponibles. No serán los mismo pares en todos los sujetos, aunque quizás se pueda filtrar a posterior las mayores ocurrencias de pares específicos.

La secuencia lógica para capturar las 2 variables de interés en general

1. Capturar la línea de pares de períodos (ejemplo: `Dia 02 - Noche 03`) de la tabla `CausaEfecto`
2. De esa línea guardar el valor de `filtro1`, `filtro2` y el `id`.
 - Crear una `key` con el `id` y el valor de `filtro1` del tipo `10013_Dia_02`
 - Usando esta `key` ir a la tabla `conteos` y capturar la variable `[N° Episodios Sueño]`
 - Crear otra `key` para el `id` y el valor de `filtro2` del tipo `10013_Noche_03`
 - Usando esta `key` ir a la tabla `duracion` y capturar la variable `[Duración sueño nocturno]`
3. Crear un `data.frame` con la información de la línea de la tabla `CausaEfecto` + los 2 registros capturados.
4. Apilar todas las líneas procesadas.
5. Et voila

Tendremos como resultado las variables de la tabla `CausaEfecto` + 2 variables. Con eso se puede hacer análisis.

Pero alguien tendrá que que programarlo.