¿Como conseguir un mejor descanso?

Análisis para la optimización del sueño

Autor: Isanagui Rojas Martínez

2023-04-30

1. Introducción

Como bien indica el título de este proyecto **¿Como conseguir un mejor descanso?**, este trabajo pretende realizar un análisis estadístico sobre métricas del sueño. Pero antes es oportuno realizar una aproximación a que es el sueño, que función tiene y algunos apuntes básicos para que nos permitirán tener un buen descanso.

El sueño es un estado fisiológico durante el cual nuestro cuerpo se relaja, se calma nuestro sistema nervioso, nuestra respiración se vuelve más lenta y profunda, nuestra actividad cerebral cambia, permitiendo de esta forma que nuestro cerebro y nuestro cuerpo rejuvenezcan.

El sueño cumple con varias funciones importantes en nuestro cuerpo: consolidación de la memoria, regulación de la temperatura corporal, reparación y regeneración de tejidos, regulación del metabolismo y el sistema inmunológico. También juega un papel crucial en el mantenimiento de nuestro bienestar emocional y mental.

Para tener un buen descanso con el sueño, se recomienda seguir estas prácticas:

1. Establecer un horario de sueño regular y tratar de mantenerlo incluso los fines de semana.
2. Crear un ambiente tranquilo y oscuro para dormir.
3. Evitar la cafeína, el alcohol y el tabaco antes de dormir.
4. Realizar actividades relajantes antes de acostarse, como leer o tomar un baño caliente.
5. Mantener una temperatura cómoda en la habitación para dormir.
6. Evitar las siestas largas durante el día.
7. Mantener una rutina de ejercicio regular, pero evitar el ejercicio intenso antes de dormir.
8. Limitar el uso de dispositivos electrónicos antes de dormir, ya que la luz azul emitida por estas pantallas dificulta la conciliación del sueño.

Siguiendo estas recomendaciones, podemos mejorar la calidad del sueño y asegurar que cuerpo y mente se recuperan adecuadamente durante el descanso nocturno.

El sujeto nos ha proporcionado información extra, externa a los datos recogidos que nos permiten interpretar algunas de las tendencias de los datos que disponemos:

- En el mes de junio el sujeto trabajaba en turnos nocturnos, hecho que le provocaba dormir pocas horas y de mala calidad. Hecho que creemos que afectará a la calidad del sueño.
- En el mes de septiembre el sujeto realizó una sustitución en horario nocturno durante la tercera semana del mes.
- En el mes de enero el sujeto estaba en una situación de estrés debido a su nuevo trabajo, en el cual hacía jornadas largas de más 8h muchos días. Además de los nervios de ser el primer mes que trabajaba 100% a comisión sin sueldo fijo, y con mucha presión por parte de la empresa. Debido a toda esta situación estaba valorando la opción de dejar el trabajo, y finalmente el 31 de enero fue su último día de trabajo.

2. Obtención de los datos

Los datos biométricos extraídos para analizar la calidad del sueño, provienen de un smartwatch modelo **Garmin Fenix 6X Pro**. El smartwatch recoge una gran cantidad de métricas para medir el sueño y determinar si se ha tenido un buen descanso. Para obtener los datos hay que realizar una solicitud a Garmin desde su página: https://www.garmin.com/es-ES/account/datamanagement/exportdata/. Ellos los compilan y posteriormente te los mandan en formato JSON.

Los datos proporcionados están divididos en tres archivos distintos, el primer archivo tiene **20 variables** y los otros dos archivos tienen **18 variables**, a excepción de las dos variables de más que tiene el primer archivo, comparten la misma estructura y formato. El lapso temporal de los datos que se dispone para el estudio van desde el *2022-06-15* hasta el *2023-03-29*, un total de **237 días**.

Las variables que recoge el dispositivo pertinentes para determinar la calidad del sueño se agrupan en 5 tipos:

1. Métricas relacionadas con el sueño (en segundos):

- deepSleepSeconds
- lightSleepSeconds
- remSleepSeconds
- awakeSleepSeconds

2. Métricas de biodatos (se recogen en una lista de variables spo2SleepSummary):

- averageSPO2
- averageHR
- lowestSPO2
- averageRespiration
- lowestRespiration
- highestRespiration
- awakeCount
- avgSleepStress

3. Métricas de puntuación sobre la calidad del sueño de 0 a 100 (se recogen en una lista de variables sleepScores). Garmin valora la puntuación de estas variables de la siguiente forma:

```
# Crear un data.frame con los nombres y valores de los grupos
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.2.2
```

```
grupos.df <- data.frame(
  Grupos = c("Pobre", "Justo", "Bueno", "Excelente"),
  Valores = c("0-60", "60-79", "79-89", "89-100")
)
kable(grupos.df, row.names = FALSE, comment = FALSE, align = c("l", "r"))
```

Grupos	Valores
Pobre	0-60
Justo	60-79

Grupos	Valores
Bueno	79-89
Excelente	89-100

- overallScore
- qualityScore
- durationScore
- recoveryScore
- deepScore
- remScore
- lightScore
- awakeiningsScore
- awakeTimeScore
- combinatedAwakeScore
- resfultnedsScore
- interruptionsScore
- restlessMomentCount

4. Variables temporales que marcan la fecha de inicio y fin del sueño.
5. Variables factoriales en formato texto que evalúan el sueño de forma positiva o negativa.

Este es la estructura de los datos que proporciona Garmin, pero para poder trabajar con estos datos hemos tenido que unificar los datos en un solo conjunto. El proceso de unificación de los datos ha constado de dos fases:

1. Eliminación de las dos variables que solo poseía el primer conjunto de datos *sleep1*.
2. Extracción de las variables que estaban agrupadas en listas para poder usar la función *cbind()*. Tras realizar todo este proceso pasamos de tener 18 variables iniciales a 30 variables.

Con los datos ya agrupados en *sleep123* se ha llevado a cabo el siguiente proceso de transformación del conjunto de datos:

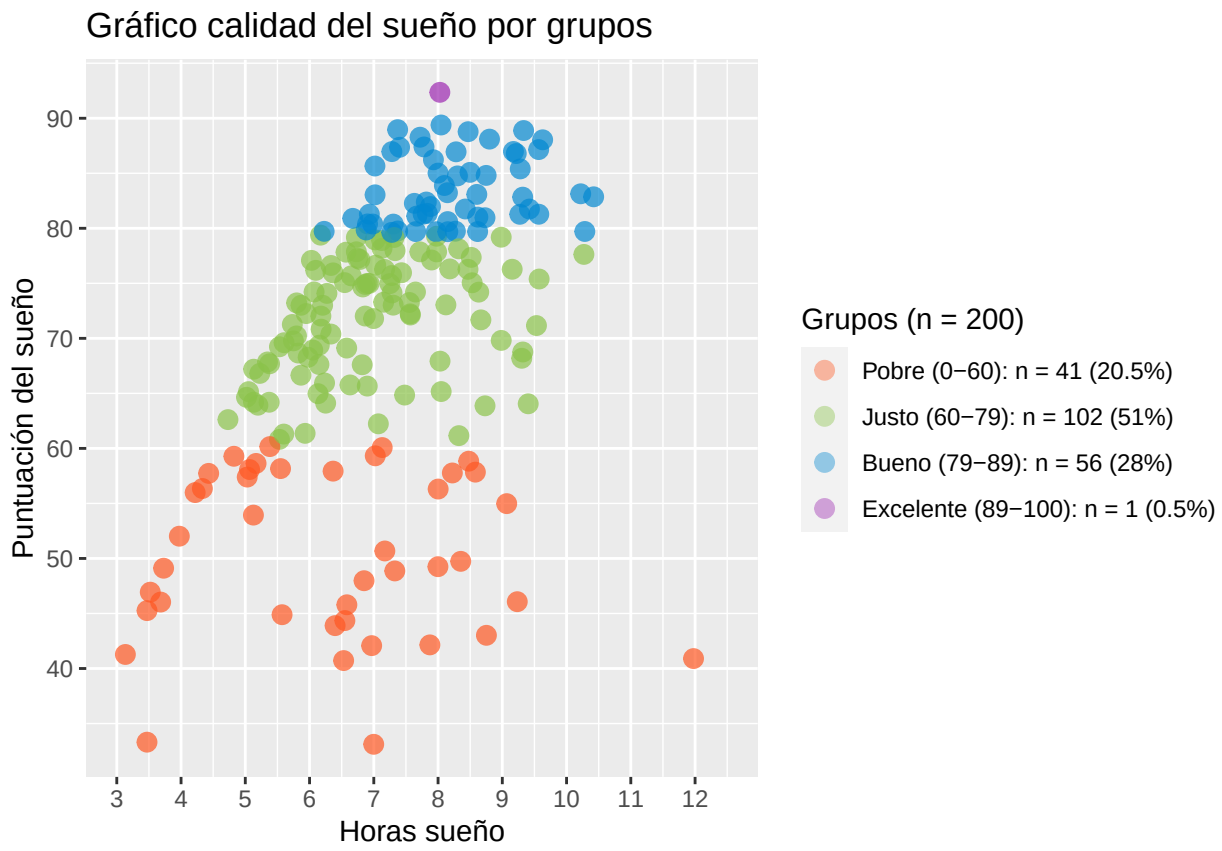
- Eliminación de variables que no aportan información útil para el análisis.
- Transformación de las variables que miden el tiempo de sueño en segundos a minutos.
- Creación de dos variables nuevas, resultado de la suma de las variables que miden los distintos tipos de sueño, expresadas en minutos y horas (para facilitar la comprensión de los datos).
- Recodificación de las variables al formato adecuado: las temporales como date, las factoriales como factor y por último las integer se han transformado con un bucle a numeric para evitar problemas durante el proceso de análisis estadístico.
- Eliminación de casos no validos y valores NA para evitar sesgos en el estudio.

Éste proceso de transformación y depuración de los datos como resultado ha dejado el conjunto de datos de la siguiente forma: **200 días** de datos recogidos y **30 variables**.

3. Análisis exploratorio

Comenzaremos realizando un análisis de dispersión de las mediciones agrupándolas según la clasificación de Garmin:

```r
#creacion grupos de sueño

grupos <- cut(sleep.no.na$overallScore,
              breaks = c(0, 60, 79, 89, 100),
              labels = c("Pobre", "Justo", "Bueno", "Excelente"))

# Cálculo de frecuencia y porcentaje de cada grupo
grupos_count <- table(grupos)
total_casos <- sum(grupos_count)

# Modificación de la variable grupo_label para incluir el rango de cada etiqueta
grupo_range <- c("0-60", "60-79", "79-89", "89-100")
grupo_label <- paste0(names(grupos_count), " (", grupo_range, "): n = ", grupos_count, " (",
                      round(100*grupos_count/total_casos,2), "%)")

# Creación del gráfico
ggplot(sleep.no.na, aes(x = totalsleepH, y = overallScore)) +
  geom_point(aes(color = grupos), alpha = 0.7, size = 3, position = "jitter") +
  ylab("Puntuación del sueño") +
  xlab("Horas sueño") +
  ggtitle("Gráfico calidad del sueño por grupos") +
  scale_x_continuous(limits = c(3,12.5), breaks = seq(3,12.5,1)) +
  scale_color_manual(values = c("#FF5722", "#8BC34A", "#0288D1", "#9C27B0"),
                     labels = grupo_label) +
  labs(color = "Calidad del sueño") +
  guides(color = guide_legend(title = paste0("Grupos (n = ", total_casos, ")"),
                              ncol = 1,
                              override.aes = list(size = 3, alpha = 0.4)))
```




```
pj<-sum(grupos_count[1:2])
be<-sum(grupos_count[3:4])

pj1<- (pj/nrow(sleep.no.na))*100
be1<- (be/nrow(sleep.no.na))*100
```

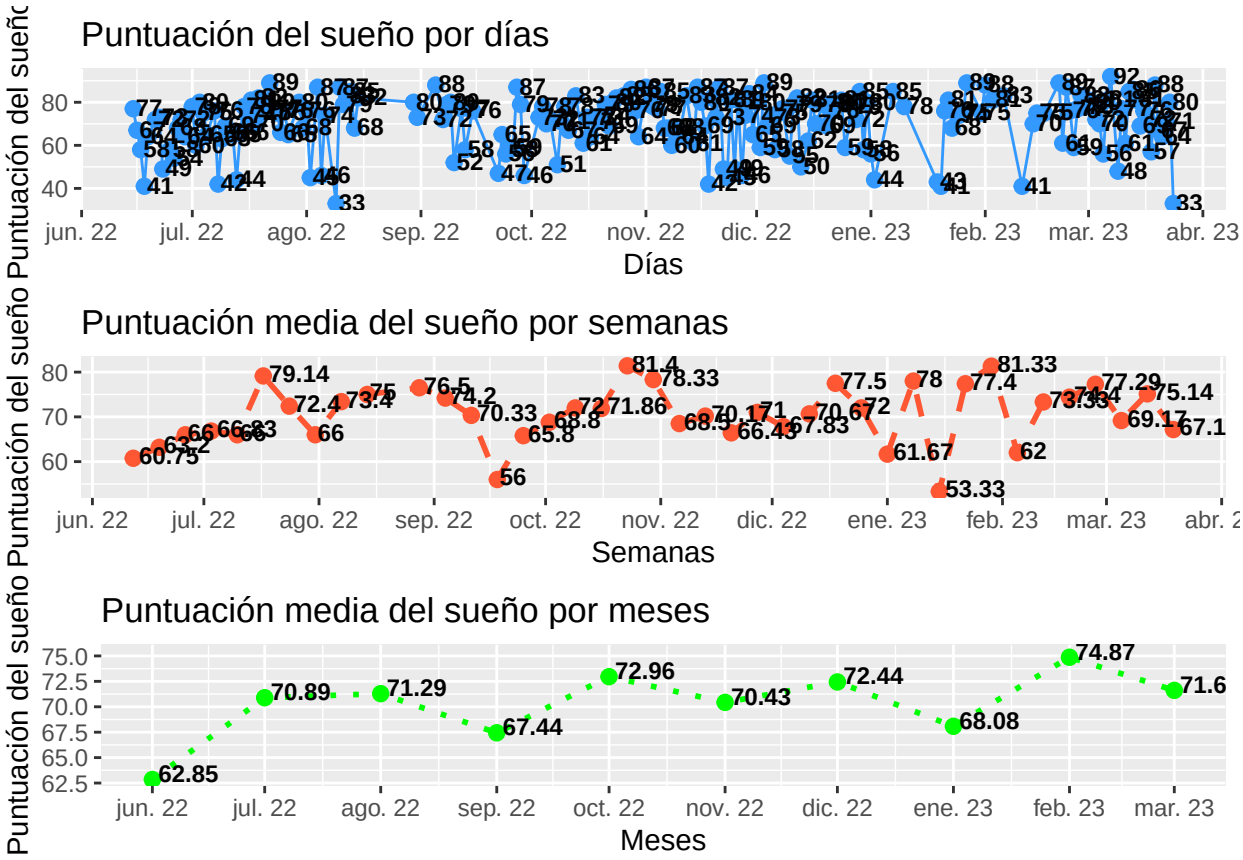
Podemos observar cierta correlación en la distribución de los datos, no se observa agrupaciones en los datos la suma de los dos primeros grupos son **143** mediciones que suponen el **71.5%** de los casos. Los dos grupos superiores suman **57** mediciones que suponen el **28.5%** de los casos.

Para poder observar el comportamiento de los datos a escala temporal, se ha generado un *dashboard* que nos permitirá ver con mayor detalle la dispersión de la calidad del sueño en el tiempo de las mediciones tomadas, estos tres gráfico nos permiten observar el comportamiento de los datos en distintas escalas temporales en la puntuación de **overallScore**:

1. El primer gráfico muestra la puntuación de forma diaria.
2. El segundo gráfico muestra la media de la puntuación del sueño por semanas.
3. El tercer gráfico muestra la media de la puntuación del sueño por meses.

```
#Linea temporal del sueño por dia

sleep_dia <- sleep.no.na %>%
  group_by(Dia = floor_date(calendarDate, unit = "day")) %>%
  summarize(media_de_datos = overallScore)
```

```r
#Linea temporal del sueño por semana
sleep_semana <- sleep.no.na %>%
  group_by(Semana = floor_date(calendarDate, unit = "week")) %>%
  summarize(media_de_datos = round(mean(overallScore),2))

#Linea temporal del sueño por meses

sleep_mensual <- sleep.no.na %>%
  group_by(Mes = floor_date(calendarDate, unit = "month")) %>%
  summarize(media_de_datos = round(mean(overallScore),2))

# Crear los tres gráficos separados

daily_plot <- ggplot(sleep_dia, aes(x = Dia, y = media_de_datos )) +
  ggtitle ("Puntuación del sueño por días") +
  geom_line(aes(color = "Puntuación diaria"), linetype = "solid") +
  geom_point(color="#3399FF", size=2.4) +
  geom_text(data = sleep_dia, aes(label = media_de_datos), hjust = -0.1, size = 3.3, nudge_y = 0.6, fon
  scale_x_date(date_breaks ="1 month", date_labels = "%b %y") +
  scale_color_manual(values = c("#3399FF"), labels = "Puntuación diaria") +
  scale_linetype_manual(values = c("solid"), labels = "Puntuación diaria") +
  ylab ("Puntuación del sueño")+
  xlab ("Días")+
  theme(legend.position = "none")  # ocultar la leyenda

weekly_plot <- ggplot(sleep_semana, aes(x = Semana, y = media_de_datos )) +
  ggtitle ("Puntuación media del sueño por semanas") +
  geom_line(aes(color = "Puntuación semanal"), size=1, linetype = "dashed") +
  geom_point(color="#FF5733", size=2.4) +
  geom_text(data = sleep_semana, aes(label = media_de_datos), hjust = -0.1, size = 3.3, nudge_y = 0.6,
  scale_x_date(date_breaks ="1 month", date_labels = "%b %y") +
  scale_color_manual(values = c("#FF5733"), labels = "Puntuación semanal") +
  scale_linetype_manual(values = c("dashed"), labels = "Puntuación semanal") +
  ylab ("Puntuación del sueño")+
  xlab ("Semanas")+
  theme(legend.position = "none")  # ocultar la leyenda

monthly_plot <- ggplot(sleep_mensual, aes(x = Mes, y = media_de_datos )) +
  ggtitle ("Puntuación media del sueño por meses") +
  geom_line(aes(color = "Puntuación mensual"), size=1, linetype = "dotted") +
  geom_point(color="green", size=2.4) +
  geom_text(data = sleep_mensual, aes(label = media_de_datos), hjust = -0.1, size = 3.3, nudge_y = 0.6,
  scale_x_date(date_breaks ="1 month", date_labels = "%b %y") +
  scale_color_manual(values = c("green"), labels = "Puntuación mensual") +
  scale_linetype_manual(values = c("dotted"), labels = "Puntuación mensual") +
  ylab ("Puntuación del sueño")+
  xlab ("Meses")+
  theme(legend.position = "none")  # ocultar la leyenda

# Unir los tres gráficos en un panel

grid.arrange(daily_plot, weekly_plot, monthly_plot, ncol=1)
```



A través de esta visualización podemos ver el comportamiento de *overallScore* desde *2022-06-15* hasta *2023-03-24* en tres diferentes escalas temporales, diaria, media semanal y media mensual. Si miramos el gráfico mensual vemos que desde el inicio de las mediciones hasta el final de las mismas ha habido un incremento en la calidad del sueño. Y si observamos los semanales y diarios nos ayudan a ver los altibajos en la puntuación del sueño con más detalle para entender los datos mensuales.

Si observamos la media de junio del 2022 el mes con menor puntuación, en los gráficos de las mediciones semanales podemos observar que también tiene medias bajas y si observamos los datos diarios podemos ver que oscilan mucho las puntuaciones pero con pocas puntuaciones altas. Si hacemos las mismas observaciones con los restos de meses a excepción de septiembre del 2022 y enero del 2023 que se parecen más al primer mes comentado, el resto tienen puntuaciones más altas como tendencia, hecho que hace subir las observaciones tanto en las medias semanales como mensuales.

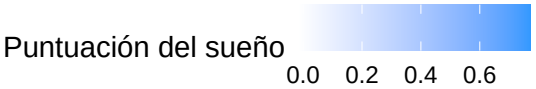
Las puntuaciones más bajas en el gráfico diario son los días que menos horas se ha dormido por norma general, por lo que creemos que la cantidad total de tiempo dormido es una variable que influye en la puntuación de *overallScore*. Para poder ver con mayor detalle los días de los que se dispone información sobre el sueño, hemos generado una serie temporal de gráficos de calor con la puntuación de *overallScore* escalada de 0 a 1.

```
generar_calendario <- function(año, mes, sleep.no.na) {
  datos_mes <- sleep.no.na %>%
    filter(year(calendarDate) == año, month(calendarDate) == mes) %>%
    complete(calendarDate = seq(as.Date(paste0(año, "-", mes, "-01")), as.Date(paste0(año, "-", mes, "-
    mutate(overallScore = ifelse(is.na(overallScore), -1, overallScore)) %>%
    dplyr::select(calendarDate, overallScore) %>%
    mutate(overallScore = (overallScore/100)) %>%
    pull(overallScore)
```

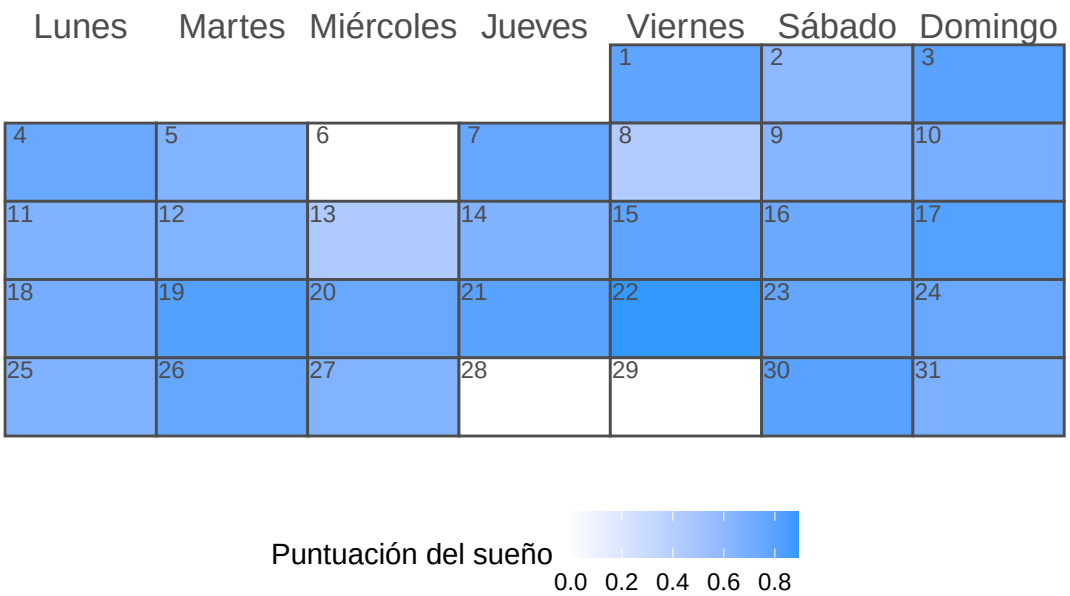
```
  dias <- rep(min(datos_mes)-0.05, days_in_month(as.Date(paste0(año, "-", mes, "-01"))))
  dias[!is.na(datos_mes)] <- datos_mes[!is.na(datos_mes)]

  calendR(year = año,
          month = mes,
          special.days = dias,
          gradient = TRUE,
          low.col = "white",
          special.col = "#3399FF",
          legend.pos = "bottom",
          legend.title = "Puntuación del sueño",
          start = c("M"))

}

#Extrae las fechas únicas de la columna de fecha
unique_dates <- unique(sleep.no.na$calendarDate)

# Convierte las fechas a formato POSIXlt para poder acceder a los componentes de año y mes
date_components <- as.POSIXlt(unique_dates)

# Define los rangos de años y meses para los que deseas generar gráficos
years <- unique(year(sleep.no.na$calendarDate))
months <- unique(month(sleep.no.na$calendarDate))

for (year in years) {
  months <- unique(month(sleep.no.na$calendarDate[year(sleep.no.na$calendarDate) == year]))
  for (month in months) {
    p <- generar_calendario(year, month, sleep.no.na)
    plot(p)
  }
}
```

JUNIO 2022

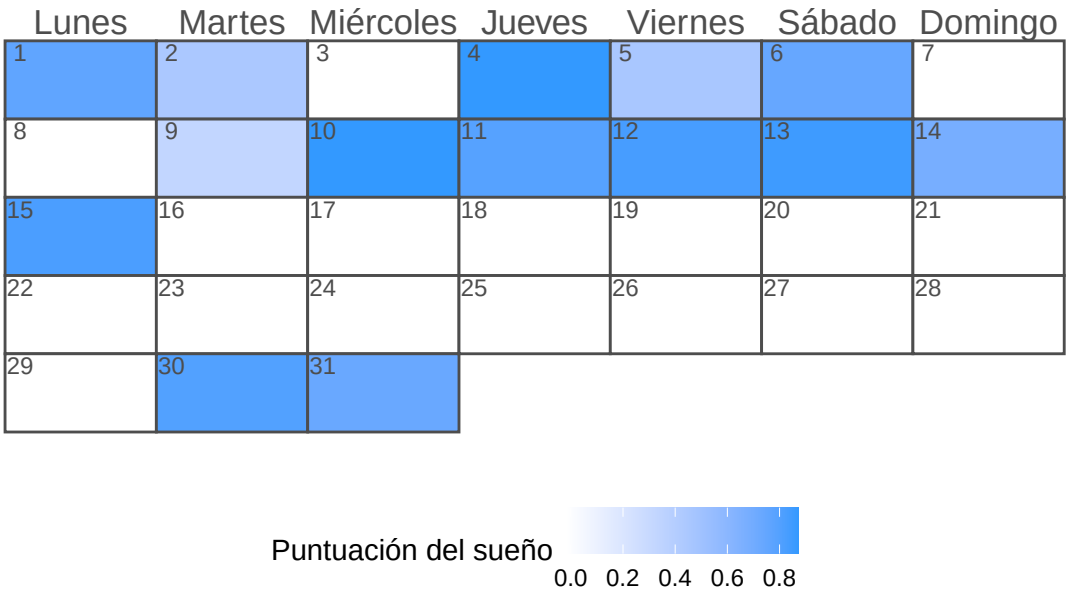
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			



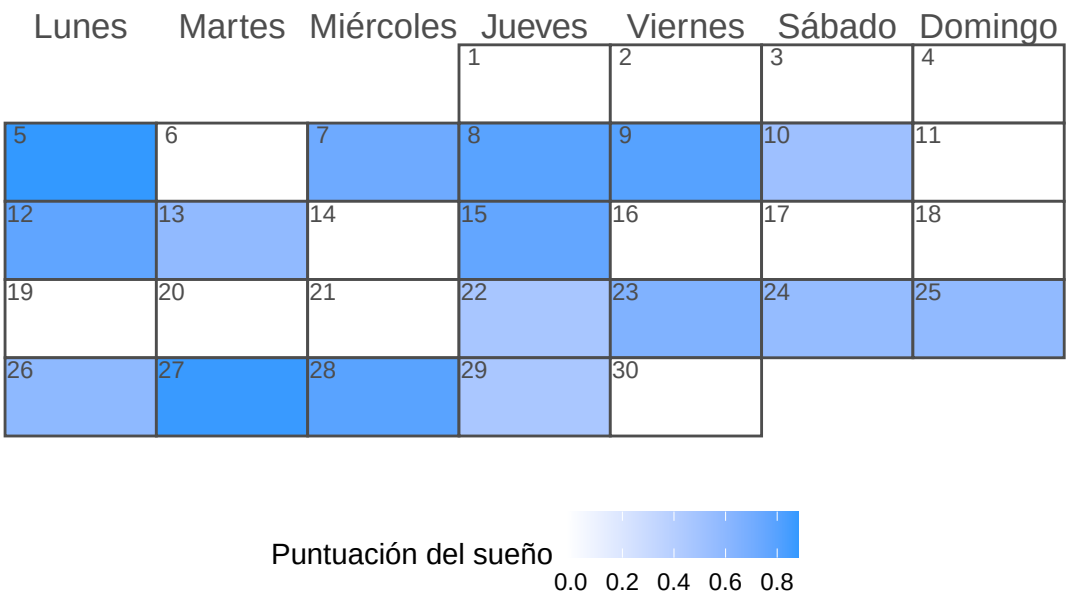
JULIO 2022


Lunes
Martes
Miércoles
Jueves
Viernes
Sábado
Domingo
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
Puntuación del sueño
0.0
0.2
0.4
0.6
0.8

AGOSTO 2022

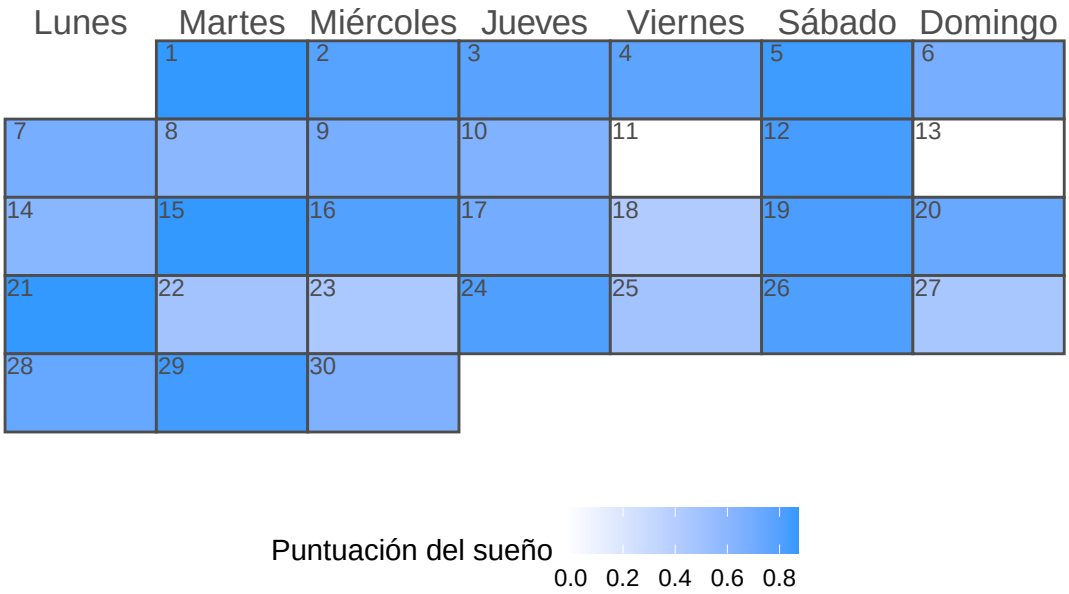

Lunes
Martes
Miércoles
Jueves
Viernes
Sábado
Domingo
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
Puntuación del sueño
0.0
0.2
0.4
0.6
0.8

SEPTIEMBRE 2022

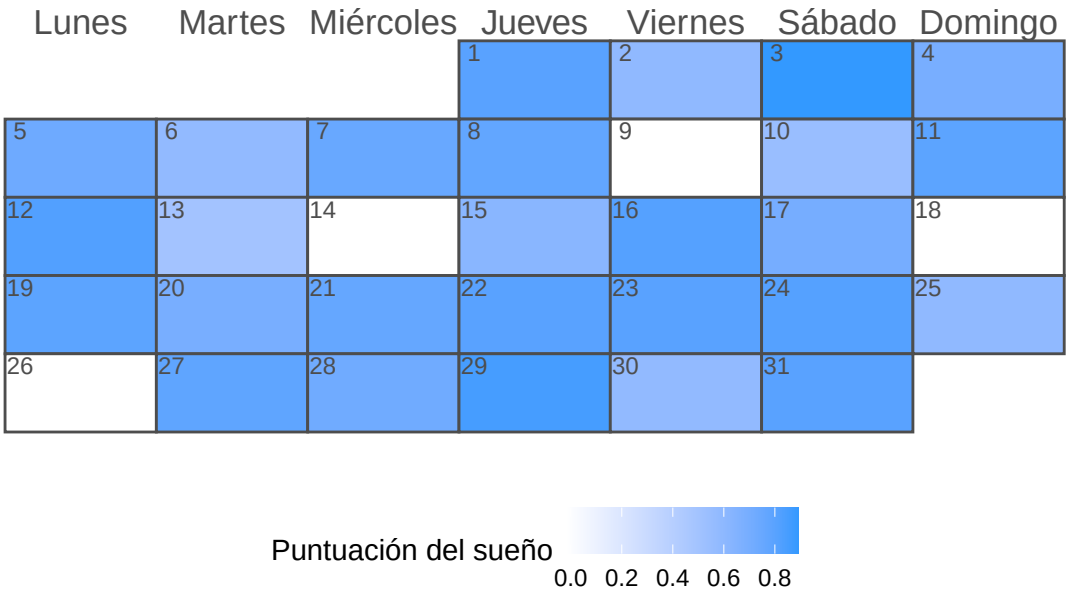


OCTUBRE 2022

NOVIEMBRE 2022



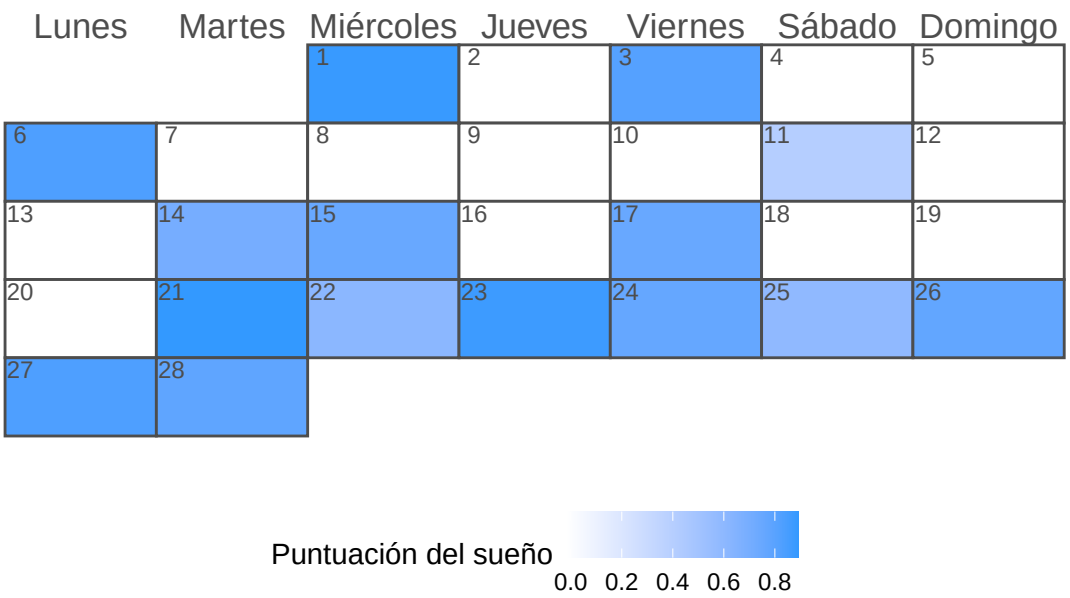
DICIEMBRE 2022

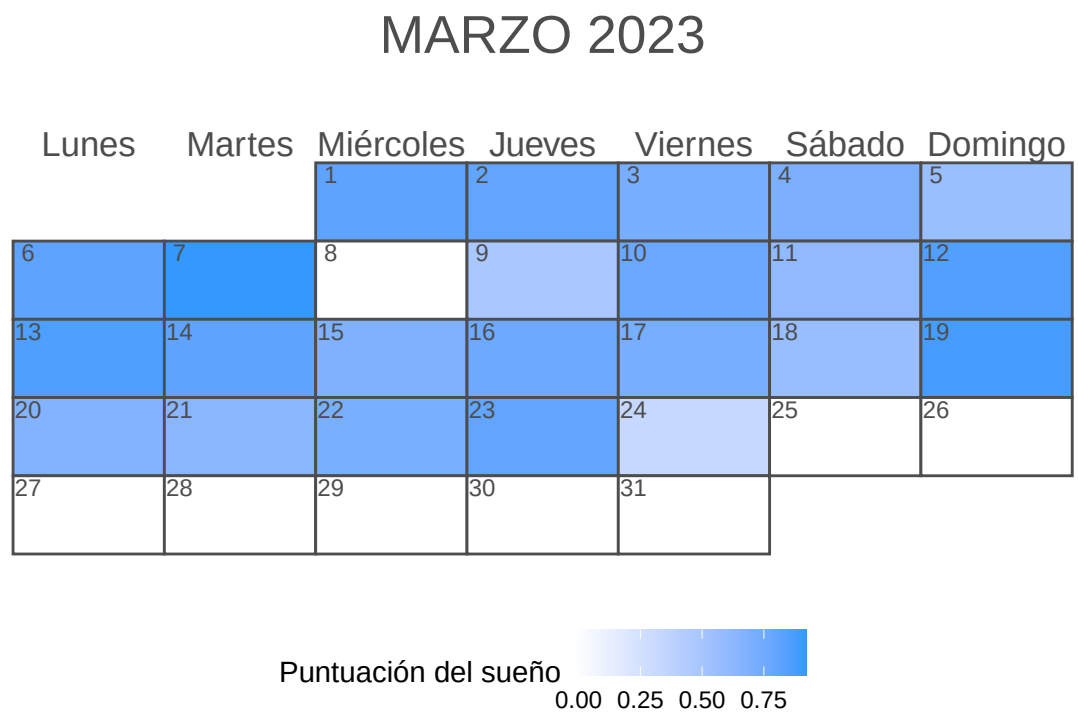


ENERO 2023


Lunes
Martes
Miércoles
Jueves
Viernes
Sábado
Domingo
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
Puntuación del sueño
0.0
0.2
0.4
0.6
0.8

FEBRERO 2023





Queremos observar si el tiempo total de sueño *totalsleepH* influye en la puntuación obtenida en *overallScore*. Para valorar si existe relación entre estas dos variables vamos realizar una prueba de correlación entre ellas:

4. Análisis estadístico

4.1 Regresión lineal simple

- Prueba de correlación:

```
cor(sleep.no.na$totalsleepH, sleep.no.na$overallScore)
```

```
## [1] 0.4438687
```

Los resultados de la prueba de correlación son positivos, muestran que existe una relación positiva y moderada, confirmando de esta forma que es correcto realizar un modelo de análisis de regresión lineal.

- Modelo estadístico de regresión lineal:

```
modelo <- lm(overallScore ~ totalsleepH, data = sleep.no.na)

resultados1 <- summary(modelo)$coefficients

# Ajustar el número de decimales para Estimate, Std. Error y t value
```

Table 2: Coeficientes del modelo de regresión lineal simple

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	44.12	3.898	11.32	< 2.22e-16
totalsleepH	3.707	0.532	6.97	4.6043e-11

Table 3: R^2 y R^2 ajustado

R-squared	0.197
Adjusted R-squared	0.193

```
resultados1[, 1:3] <- round(resultados1[, 1:3],3)

# Convertir los valores de Pr(>|t|) en formato cientifico

resultados1[, 4] <- format.pval(resultados1[, 4], scientific = TRUE)

knitr::kable(resultados1, caption = "Coeficientes del modelo de regresión lineal simple", table.envir= "
  kableExtra::column_spec(1, width = "2.5cm") %>%
  kableExtra::column_spec(2, width = "2.5cm") %>%
  kableExtra::column_spec(3, width = "2.5cm") %>%
  kableExtra::column_spec(4, width = "2.5cm") %>%
  kableExtra::column_spec(5, width = "2.5cm")

#R cuadrado y R cuadrado ajustado

resultados2 <- rbind(c("R-squared", round(summary(modelo)$r.squared, 3)))
resultados2 <- rbind(resultados2,c("Adjusted R-squared", round(summary(modelo)$adj.r.squared, 3)))

# Imprime la tabla de resultados

knitr::kable(resultados2, caption = "$R^2$ y $R^2$ ajustado", table.envir= "table") %>%
  kableExtra::column_spec(1, width = "2.5cm") %>%
  kableExtra::column_spec(2, width = "2.5cm")
```

La ecuación de la recta de la regresión es: $overallScore = 44.12 + 3.707(totalsleepH)$. De aquí podemos extraemos que si una persona no duerme nada su puntuación de **overallScore** será $Intercept = 44.12$. El coeficiente de **totalsleepH** es $\beta = 3.707$, muestra la existencia de una pendiente positiva y por cada hora adicional de sueño del sujeto, se espera un aumento de $\Delta y = 3.707$ en la puntuación de **overallScore**.

Este modelo de análisis es significativo debido a que el $p-valor = 4.6043e-11$ es inferior a $p-valor < 0.05$, demostrando que es muy poco probable que esta relación causal sea fruto del azar.

```
modelo1t <- lm(overallScore ~ totalsleepH, data = sleep.no.na)

resultados1t <- summary(modelo1t)$coefficients

# Ajustar el número de decimales para Estimate, Std. Error y t value

resultados1t[, 1:3] <- round(resultados1t[, 1:3],3)
```

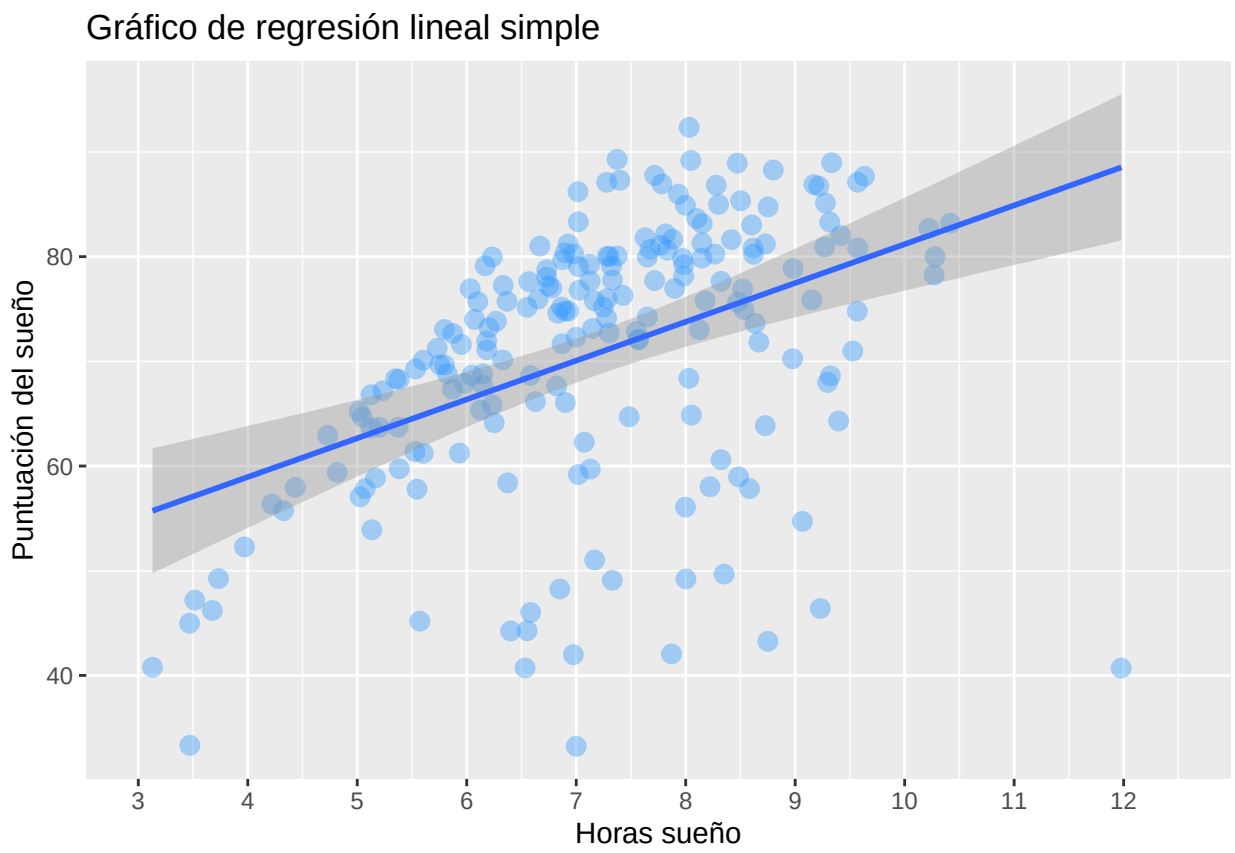
```r
# Convertir los valores de Pr(>|t|) en formato cientifico

resultados1t[, 4] <- format.pval(resultados1t[, 4], scientific = TRUE)

#R cuadrado y R cuadrado ajustado

resultados2t <- rbind(round(summary(modelo1t)$r.squared, 3))
resultados2t <- rbind(resultados2t,c(round(summary(modelo1t)$adj.r.squared, 3)))
```

El valor de $R^2_{ajustado} = 0.193$ nos indica que este modelo de regresión lineal simple explica el **19.3%** de la variabilidad en **overallScore**.

El valor del error estándar residual indica la cantidad de variabilidad no explicada en el modelo, y se puede utilizar para evaluar la precisión de las predicciones del modelo. En este caso, el valor del error estándar residual es $t-value = 11.32$, lo que significa que las predicciones del modelo pueden tener un error aproximado de 11.32 puntos.

```r
ggplot(sleep.no.na, aes(x = totalsleepH, y = overallScore)) +
  geom_point(color="#3399FF", alpha=0.4, size=3, position = "jitter") +
  ylab("Puntuación del sueño") +
  xlab("Horas sueño") +
  ggtitle("Gráfico de regresión lineal simple") +
  scale_x_continuous(limits = c(3,12.5), breaks = seq(3,12.5,1)) +
  geom_smooth(method = "lm", se = TRUE, level = 0.99)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



El gráfico de regresión lineal muestra la relación entre la variable **independiente x=totalsleepH** y la variable **dependiente y=overallScore**. Como hemos observado antes con la ecuación de la recta entre las 5 y 10 horas de sueño se concentran la mayor parte de las mediciones en la calidad del sueño del sujeto, cumpliendo que a mayor cantidad de horas dormidas **x=totalsleepH** mayor puntuación en **y=overallScore** lo que indica que a más horas de sueño, mayor puntuación en la calidad del sueño.

Hemos podido observar que este modelo solo es capaz de explicar el **19.3%** de la variabilidad en **overallScore**. En el gráfico de regresión lineal podemos observar que hay casos que no cumplen con la ecuación de la recta a más horas de sueño mayor puntuación de **overallScore**. Por ello y para poder mejorar éste análisis inicial usaremos todos los datos y variables de las que disponemos para realizar un análisis de PCA (Principal Component Analisys).

4.2 Análisis PCA Para poder realizar un análisis de PCA se ha tenido que transformar el conjunto de datos nuevamente para poder cumplir con los requisitos estádisticos que requiere hacer el análisis:

1. Eliminación de variables no numéricas.
2. Transformación de las variables temporales a través de varios procesos:

   2.1 Selección solo de la hora de inició y fin de la medición del sueño, eliminando la fecha.

   2.2 Transformación del formato de hora de hh:mm a hh(las horas que no tenían una medición a en punto, se las ha redondeando al valor hh más cercano).

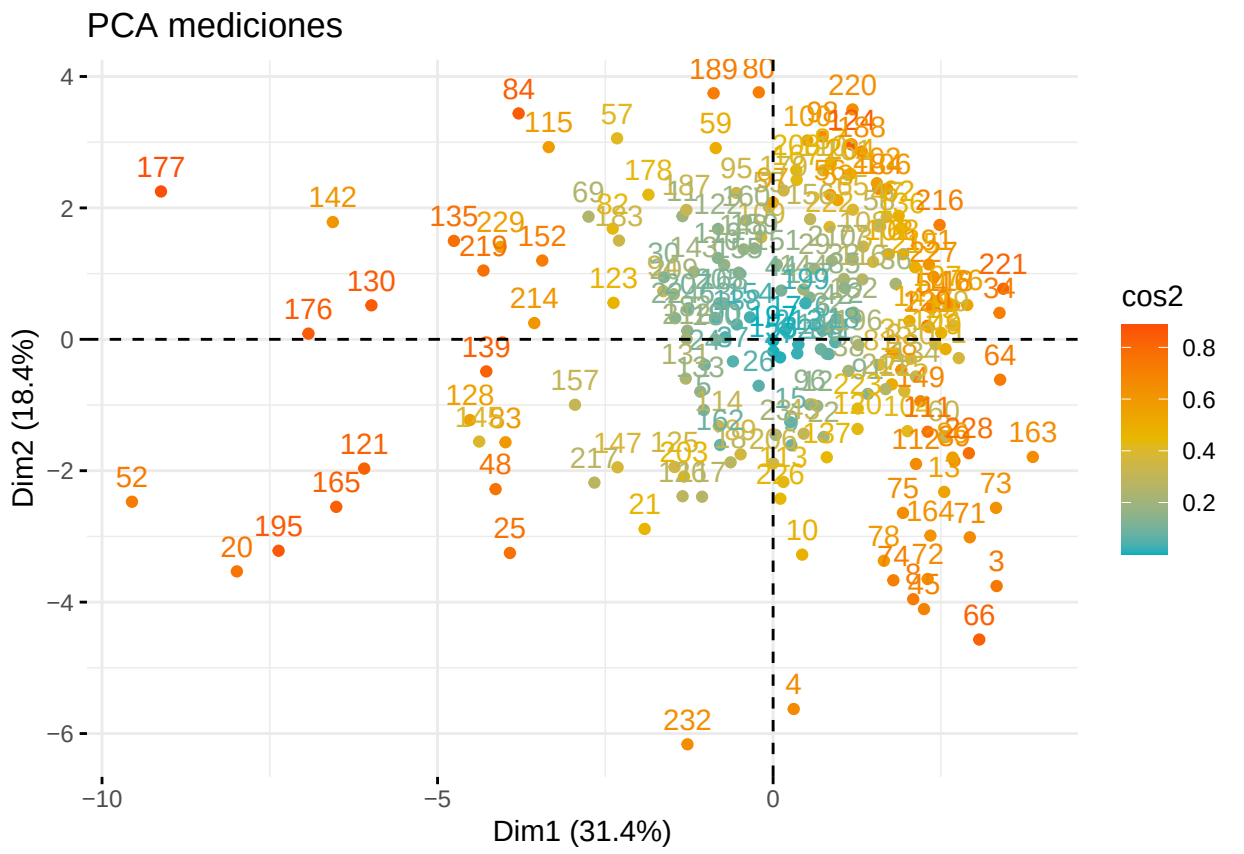
   2.3 Recodificación de esta nueva variable a valor numérico.

   2.4 Transformación de los valores de inicio de medición temporal redondeados: aquellos que están por debajo de las 23h se les ha restado -24 para diferenciarlos de las horas después de las 00h.
3. Eliminación de las variables de medición del tiempo de sueño en minutos.
4. Eliminación de cuatro variables de puntuación sobre las mediciones de **Garmin Fenix 6x Pro**.

Después de esta nueva fase de depuración, el conjunto de datos tiene **19 variables** y **200 mediciones**.

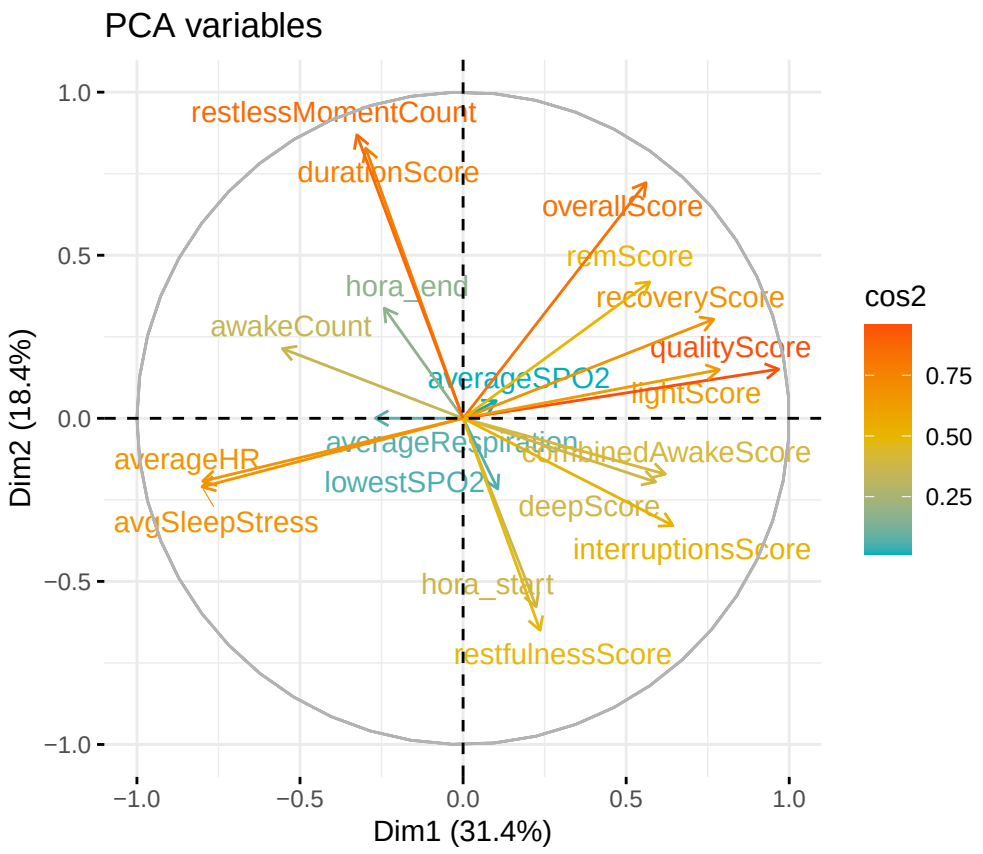
4.2.1 Análisis exploratorio PCA Podemos graficar el objeto creado a través del PCA con el nombre *pca* para entender mejor nuestros datos. Ambos gráficos tienen un degradado basado en el Cos^2 para medir la calidad de la representación de las observaciones en el plano de los componentes principales, tanto para individuos (primer gráfico) como para variables (segundo gráfico):

```
fviz_pca_ind(pca,
             col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             title= "PCA mediciones")
```



El gráfico de mediciones nos permite observar la existencia de agrupaciones en nuestro conjunto de datos. En caso de existir una clara agrupación de los datos nos permitiría realizar un análisis de clustering. Pero como podemos observar en nuestros datos no se aprecia ninguna agrupación en los datos.

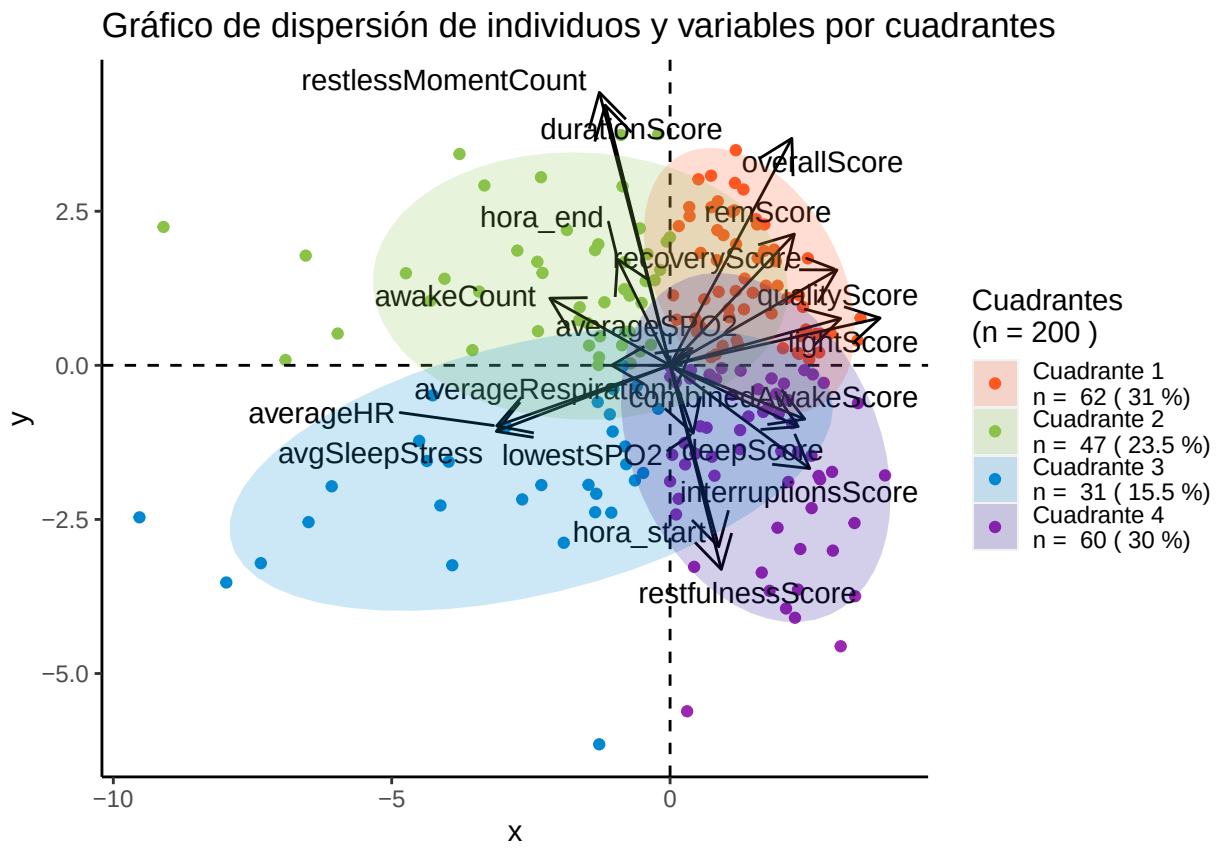
```
fviz_pca_var(pca, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE,
             title= "PCA variables")
```



Ambos gráficos se interpretan de la misma manera: cuanto más bajo sea el valor del Cos^2, menor será la capacidad de representación de esas mediciones o variables en el conjunto del PCA. Por el contrario, cuanto mayor sea el valor del Cos^2, mayor será la capacidad de representación de esas mediciones o variables.

Si combinamos ambos gráficos en uno solo y eliminamos el degradado basado en el Cos^2, podemos obtener información adicional interesante:

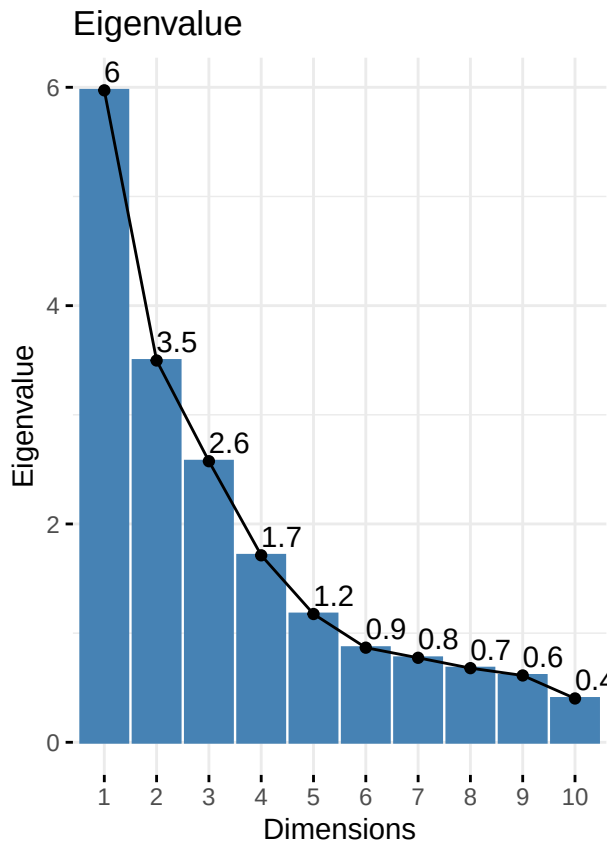
```
# Crear el gráfico de dispersión con las flechas que representan a las variables
  ggplot(datos, aes(x = x, y = y)) +
  geom_point(aes(color = Cuadrantes)) +
  geom_segment(data = flechas, aes(x = x0, y = y0, xend = x1, yend = y1), arrow = arrow(length = unit(0
  ggrepel::geom_text_repel(data = flechas, aes(x = x1, y = y1, label = variable)) +
  ggtitle("Gráfico de dispersión de individuos y variables por cuadrantes")+
  theme(panel.background = element_rect(fill = "white"),
        axis.line = element_line(colour = "black")) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  stat_ellipse(aes(fill = Cuadrantes), geom = "polygon", alpha = 0.2)+
    scale_color_manual(values = c("#FF5722", "#8BC34A", "#0288D1", "#9C27B0")) +
    scale_fill_manual(values = c("#FF5722", "#8BC34A", "#0288D1", "#260F99"))+
    labs(color = paste("Cuadrantes\n(n =", total_casos, ")"))+
    guides(fill = guide_legend(title = paste("Cuadrantes\n(n =", total_casos, ")")))
```

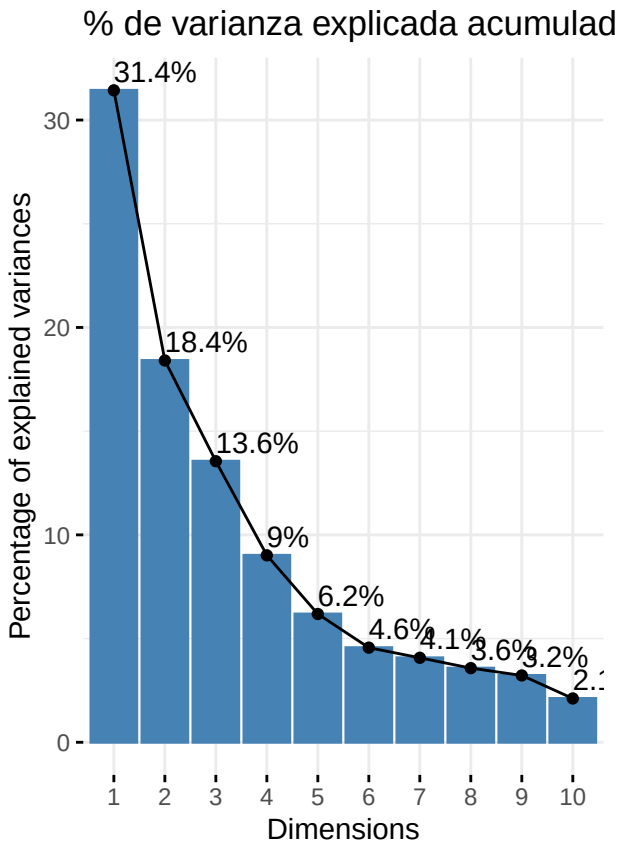


En este gráfico, debemos observar la distribución de los datos por cuadrantes en relación al eje (0,0). Hay cuatro cuadrantes, cada uno de ellos coloreado y resaltado con una elipse que agrupa los datos en su interior. Las mediciones agrupadas en el primer cuadrante están influenciadas por las variables que se encuentran dentro de ese mismo cuadrante. La longitud del vector que representa a cada variable nos muestra su importancia en relación a las mediciones existentes en el mismo cuadrante.

4.2.2 Selección de PC Una de las principales ventajas del análisis PCA es su capacidad para reducir la dimensionalidad de un conjunto de datos al reducir el número de variables a través de la identificación de Componentes Principales (PC). Existen varios métodos para seleccionar los PC más importantes, pero en este trabajo solo consideraremos dos de ellos. El primer método consiste en seleccionar aquellos PC cuyo valor propio (eigenvalue) sea mayor que 1. Un valor propio inferior a 1 indica que la variable original aporta más información que el PC correspondiente. El segundo método consiste en seleccionar aquellos PC que expliquen un porcentaje acumulado de la varianza igual o superior al 80%.

```
p1<-fviz_eig(pca, addlabels = TRUE, choice = "eigenvalue", main = "Eigenvalue")
p2<-fviz_eig(pca, addlabels = TRUE, choice = "variance", main = "% de varianza explicada acumulada")

# Crear el panel con los dos gráficos
grid.arrange(p1, p2, ncol = 2)
```






```
# Contar el número de filas donde el valor de "Eigenvalue" es mayor que 1
num_components <- sum(eig.var$Eigenvalue > 1)

num_componentst <- sum(eig.var$Eigenvalue)

# Mostrar el resultado
num_components
```

```
## [1] 5
```

```
# componentes totales del grafico

n_comp.eig <- length(p1$data$eig)

n_comp.var <- length(p2$data$eig)

# Encontrar el índice del primer componente mayor al 80% de la varianza acumulada
index <- which(eig.var$`% Acumulado` >=80)[1]

# Mostrar el resultado
eig.var[index,]
```

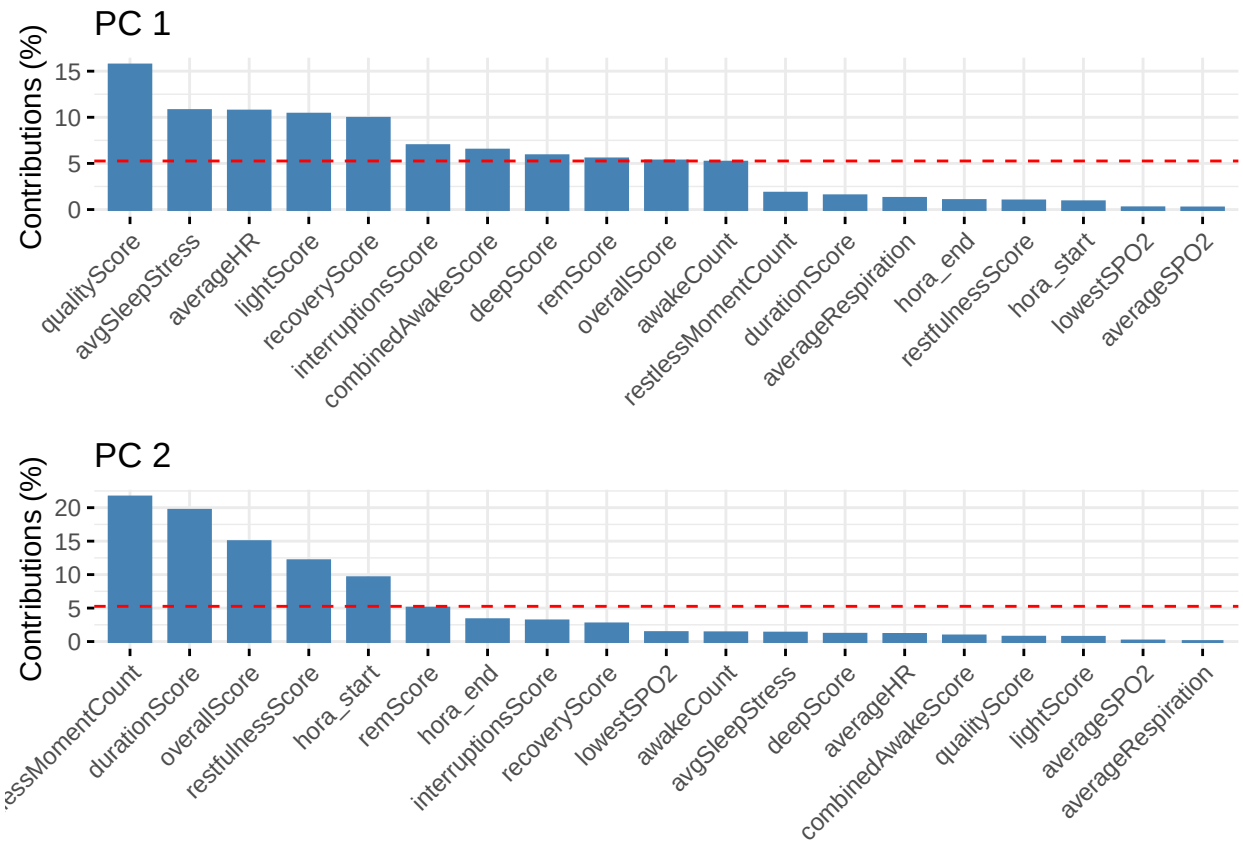
```
##        Eigenvalue % Acumulado
## comp 6  0.8668503    83.15728
```

```r
# Acceder al valor de "% Acumulado" del componente correspondiente
acumulado <- eig.var[index, "% Acumulado"]

# Mostrar el resultado
acumulado<-round(acumulado,2)
```

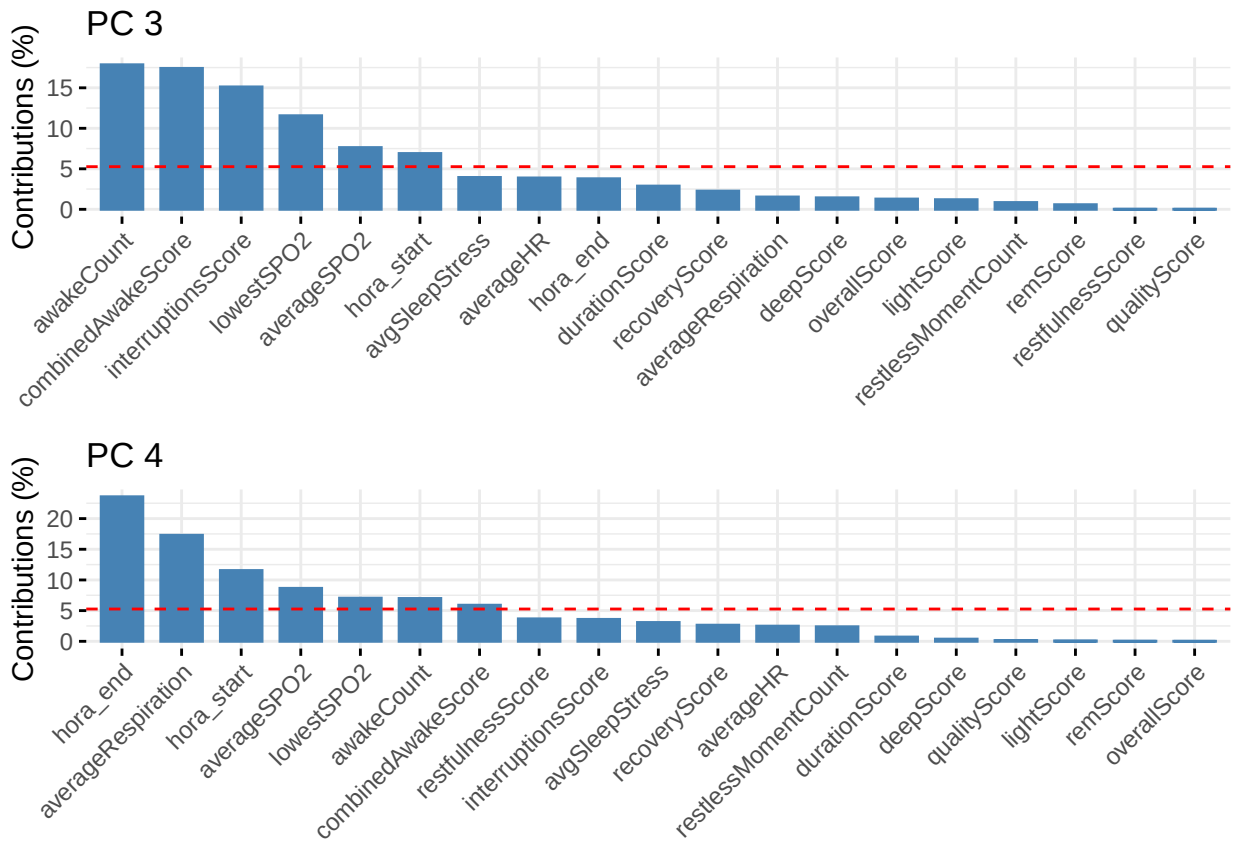
En este *dashboard* de Scree plots, podemos observar los primeros **10 PC** en términos del porcentaje de varianza acumulada y del valor propio (eigenvalue).Estos gráficos nos ayudan a determinar cuántos PC debemos seleccionar. Utilizando el método de selección basado en el criterio de *eigenvalue* > 1, deberíamos seleccionar los primeros **5 PC**. Utilizando el segundo método, basado en el porcentaje de varianza acumulada explicada, deberíamos seleccionar los primeros **6 PC**, que suman un total de **83.16% varianza acumulada explicada**. Para nuestro estudio, utilizaremos el segundo método y seleccionaremos **6 PC**.

A continuación y una vez ya seleccionados los **6 PC**, vamos a ver que variables del conjunto de datos para entender que variables agrupa cada PC:

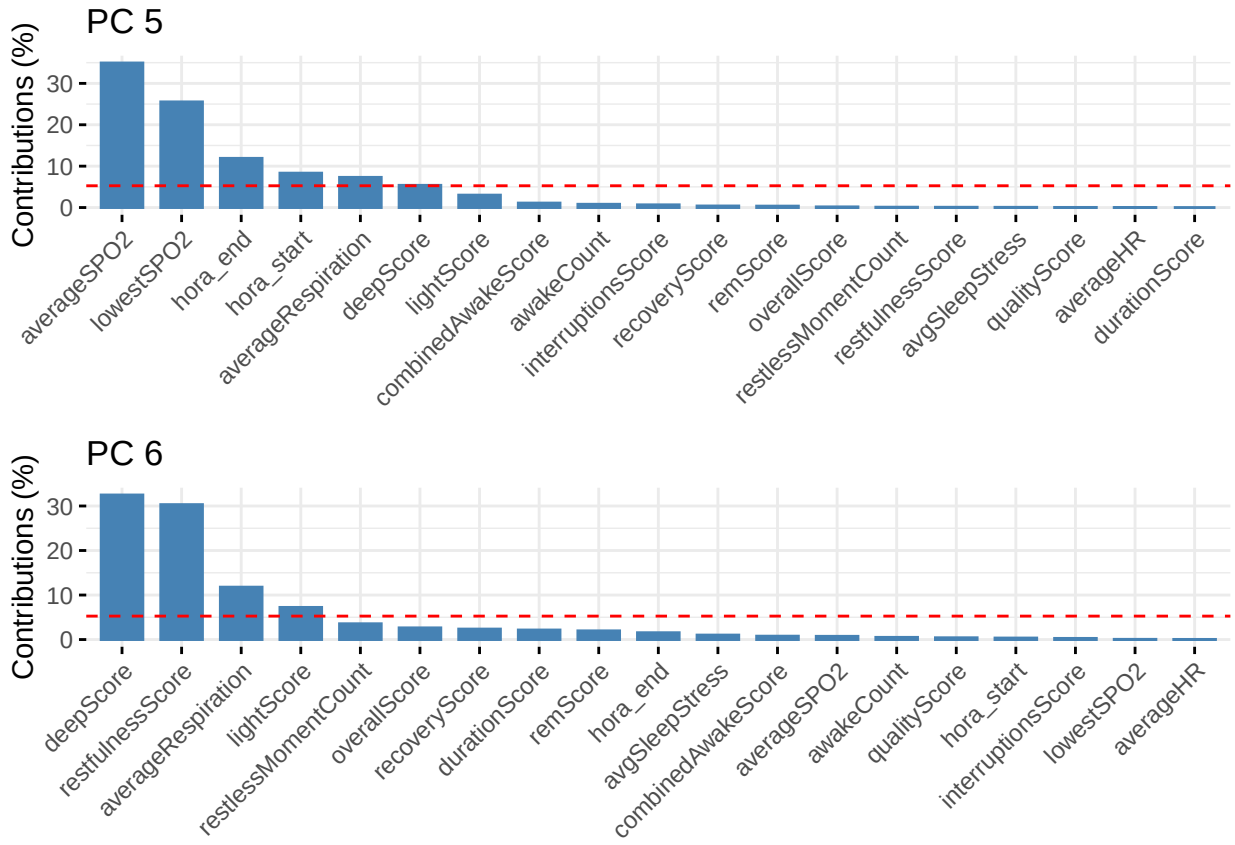
```r
#Graficos de contribucion de variables a PC
v1<-fviz_contrib(pca, axes = 1, choice = "var", title ="PC 1")
v2<-fviz_contrib(pca, axes = 2, choice = "var", title ="PC 2")
v3<-fviz_contrib(pca, axes = 3, choice = "var", title ="PC 3")
v4<-fviz_contrib(pca, axes = 4, choice = "var", title ="PC 4")
v5<-fviz_contrib(pca, axes = 5, choice = "var", title ="PC 5")
v6<-fviz_contrib(pca, axes = 6, choice = "var", title ="PC 6")

#Union de los graficos
grid.arrange(v1,v2,ncol = 1)
```



```
grid.arrange(v3,v4,ncol=1)
```




```
grid.arrange(v5,v6,ncol = 1)
```



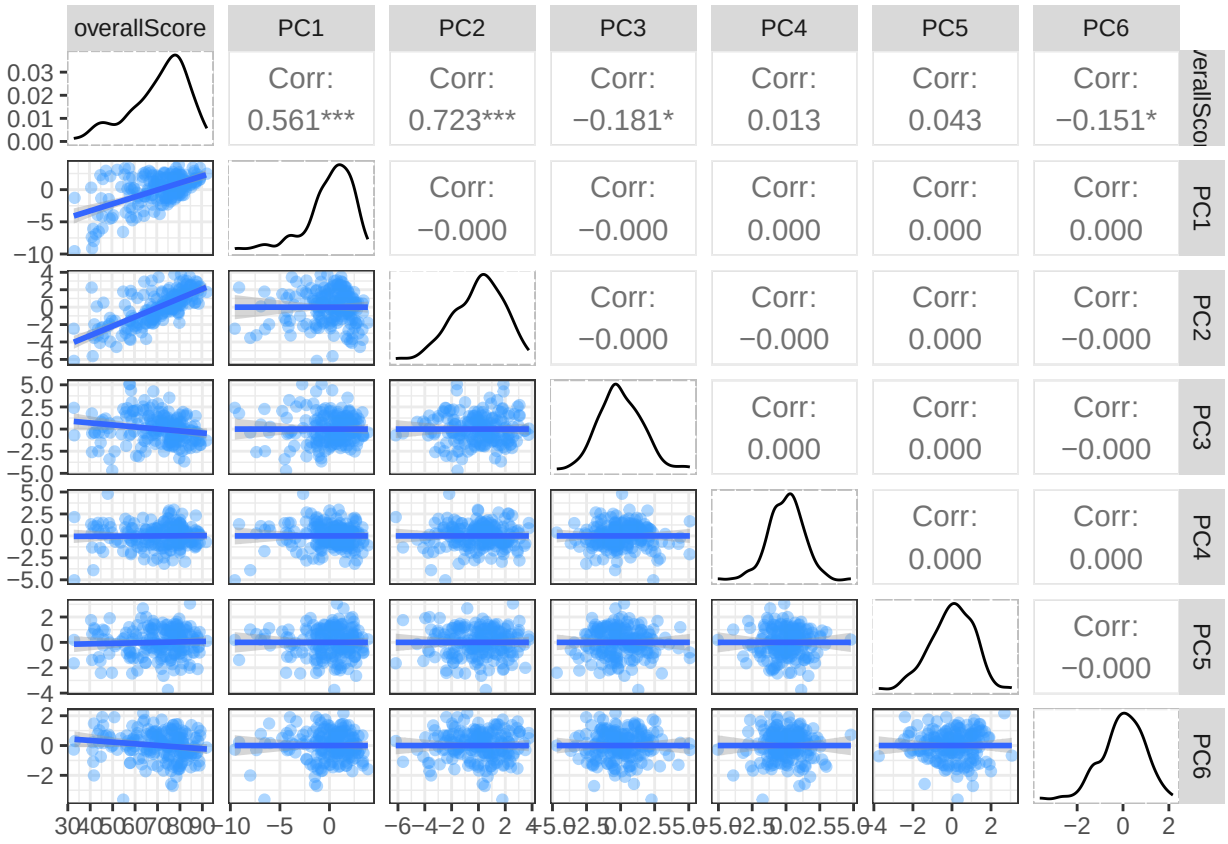
Están ordenados de mayor varianza explicada a menor varianza explicada como podemos observar en los Scree plot anteriores.

Para finalizar el apartado de análisis hemos creado una matriz de correlaciones y regresión lineal simple para todos los PC seleccionados.

```
#Matriz de correlacion multiple y regresion lineal multiple

p_scatter <- ggpairs(data_all, columns = 1:7, lower = list(continuous = function(data, mapping, ...) {
  ggplot(data = data, mapping = mapping) +
    geom_point(alpha = 0.4, color = "#3399FF", position = "jitter") +
    geom_smooth(method = "lm", se = TRUE, level = 0.99) +
    theme_bw()
}))

p_scatter <- p_scatter +
  theme(panel.background = element_rect(fill = "white", colour = "gray"))

p_scatter
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Crear el objeto "modelo" con los coeficientes del modelo
modelo56 <- resumen_modelopca$coefficients

# Redondear los coeficientes
modelo56[, 1:3] <- round(modelo56[, 1:3], 3)

# Convertir los valores de Pr(>|t|) en formato científico
modelo56[, 4] <- format.pval(modelo56[, 4], scientific = TRUE)

# Imprimir la tabla de coeficientes
```

Table 4: Coeficientes del modelo de regresión lineal simple

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	70.7	0.296	238.898	< 2.22e-16
pca_XDim.1	2.903	0.121	23.976	< 2.22e-16
pca_XDim.2	4.888	0.158	30.889	< 2.22e-16
pca_XDim.3	-1.427	0.184	-7.737	5.5439e-13
pca_XDim.4	0.124	0.226	0.547	5.8522e-01
pca_XDim.5	0.504	0.273	1.848	6.6122e-02
pca_XDim.6	-2.049	0.318	-6.448	8.9019e-10

Table 5: R^2 y R^2 ajustado

R-squared	0.894
Adjusted R-squared	0.891

```
knitr::kable(modelo56, caption = "Coeficientes del modelo de regresión lineal simple", table.envir= "tab
  kableExtra::column_spec(1, width = "2.5cm") %>%
  kableExtra::column_spec(2, width = "2.5cm") %>%
  kableExtra::column_spec(3, width = "2.5cm") %>%
  kableExtra::column_spec(4, width = "2.5cm") %>%
  kableExtra::column_spec(5, width = "2.5cm")
```

```
# Seleccionar los valores de R cuadrado y R cuadrado ajustado y redondear
rcuadrado <- round(resumen_modelopca$r.squared, 3)
rcuadrado_ajustado <- round(resumen_modelopca$adj.r.squared, 3)

# Crear la tabla de resultados de R cuadrado y R cuadrado ajustado
resultados <- rbind(c("R-squared", rcuadrado))
resultados <- rbind(resultados, c("Adjusted R-squared", rcuadrado_ajustado))

# Imprimir la tabla de resultados
knitr::kable(resultados, caption = "$R^2$ y $R^2$ ajustado", table.envir= "table") %>%
  kableExtra::column_spec(1, width = "2.5cm") %>%
  kableExtra::column_spec(2, width = "2.5cm")
```

El valor de $R^2_{ajustado} = 0.891$ nos indica que este modelo es capaz de explicar **89.1%** de la variabilidad en **overallScore**. En el gráfico matriz podemos observar todas las correlaciones de **overallScore** en la primera fila con todos los PC y en la primera columna podemos ver todos los gráficos de dispersión.

5. Conclusiones

Como hemos podido observar en el modelo de regresión lineal simple inicial la relación solo es capaz de explicar el **19.3%** de la varioanza explicada. Con el análisis PCA exploratorio en caso de haber observado la existencia de grupos en los datos se podría haber realizado un análisis de clusterings. Además nuestro PCA ha demostrado tener una baja redundancia, indicando que hay poca correlación entre las variables originales y que cada variable proporciona información única.

Hemos continuado el PCA para reducir la dimensionalidad del conjunto de datos, seleccionado el número de PC con el método de **varianza acumulada explicada**, eligiendo de esta forma **6 PC**. Y de nuevo hemos

vuelto a calcular un modelo de regresión lineal simple con todos los PC consiguiendo explicar **89.1%** de la variabilidad en **overallScore**

Debido a la falta de un marco teórico sólido sobre el sueño y sus implicaciones para poder modelar e interpretar adecuadamente los datos. Por esta razón, se ha llevado a cabo una aproximación inductiva con el objetivo de identificar las variables que tienen mayor influencia en la variable dependiente **overallScore**. Se creyó que la cantidad de tiempo dormido sería uno de los factores más determinantes y pese a que el primer modelo era capaz de explicar **19.3%** de los datos, este resultado nos mostraba que hay otras variables que afectan a la calidad del sueño.

Con el PCA se buscaba repetir el mismo proceso pero reduciendo la dimensionalidad para conseguir un mayor nivel de explicación **89.1%**.

6. (In)Conclusiones

Debido a su trabajo, el sujeto no ha podido seguir las recomendaciones para lograr un buen descanso mencionadas en la introducción. Para mejorar los resultados del estudio, sería interesante que el sujeto siguiera estas recomendaciones durante 1 o 2 meses. Luego, podríamos analizar los nuevos datos y compararlos con los resultados anteriores. De esta manera, podríamos realizar un *training-test* con datos que cumplen con las premisas necesarias para tener un buen descanso y otros que no necesariamente las cumplen. Así, podríamos analizar con mayor detalle los efectos de las distintas variables en la calidad del sueño.