# Sleep Health Statistical Study

Rebeca Suárez Ojeda

## 1 Preparación del conjunto de datos

### 1.1 Cargar el archivo de datos

```
library(readxl)

df <- read_excel("sleephealth.xlsx")
head(df)
```

```
## # A tibble: 6 x 13
##   Person.ID Gender Age   Occupation           Sleep.Duration Quality.of.Sleep
##       <dbl> <chr>  <chr> <chr>                <chr>                     <dbl>
## 1         1 Male   27    Software Engineer    6.1                           6
## 2         2 Male   28    Doctor               6.2                           6
## 3         3 Male   28    Doctor               6.2                           6
## 4         4 Male   28    Sales Representative 5.9                           4
## 5         5 Male   28    Sales Representative 5.9                           4
## 6         6 Male   28    Software Engineer    5.9                           4
## # i 7 more variables: Physical.Activity.Level <dbl>, Stress.Level <dbl>,
## #   BMI.Category <chr>, Blood.Pressure <chr>, Heart.Rate <dbl>,
## #   Daily.Steps <dbl>, Sleep.Disorder <chr>
```

### 1.2 Nombres de las columnas

```
colnames(df)[colnames(df) == "Person.ID"] <- "ID"
colnames(df)[colnames(df) == "Sleep.Duration"] <- "SH"
colnames(df)[colnames(df) == "Quality.of.Sleep"] <- "SQ"
colnames(df)[colnames(df) == "Physical.Activity.Level"] <- "PHY"
colnames(df)[colnames(df) == "Stress.Level"] <- "Stress"
colnames(df)[colnames(df) == "BMI.Category"] <- "BMI"
colnames(df)[colnames(df) == "Blood.Pressure"] <- "BP"
colnames(df)[colnames(df) == "Heart.Rate"] <- "HR"
colnames(df)[colnames(df) == "Daily.Steps"] <- "Steps"
colnames(df)[colnames(df) == "Sleep.Disorder"] <- "SD"


colnames(df)
```

```
##  [1] "ID"         "Gender"     "Age"        "Occupation" "SH"
##  [6] "SQ"         "PHY"        "Stress"     "BMI"        "BP"
## [11] "HR"         "Steps"      "SD"
```

## 2 Normalización de formatos en variables categóricas

Antes de proceder con la limpieza, se revisaron los valores únicos de las variables categóricas:

```
list(
  Gender = unique(df$Gender),
  Occupation = unique(df$Occupation),
  BMI = unique(df$BMI),
  SleepDisorder = unique(df$`SD`)
)
```

```
## $Gender
## [1] "Male"   "Female"
##
## $Occupation
##  [1] "Software Engineer"    "Doctor"               "Sales Representative"
##  [4] "Teacher"              "Nurse"                "Engineer"
##  [7] "Accountant"           "Scientist"            "Lawyer"
## [10] "Salesperson"          "engineer"             "Manager"
##
## $BMI
## [1] "Overweight"    "Normal"        "Obese"         "Normal Weight"
##
## $SleepDisorder
## [1] "None"        "Sleep Apnea" "Insomnia"
```

- En `Occupation`, se detectaron inconsistencias como `"engineer"` (minúscula), y categorías redundantes como `"Salesperson"` y `"Sales Representative"`, que hacen referencia al mismo rol. Se unificaron bajo `"Sales"`.
- En `BMI`, se encontraron las categorías `"Normal"` y `"Normal Weight"`, que también se unificaron como `"Normal"`.
- Las variables `Gender` y `SleepDisorder` no presentaron inconsistencias.

Tras esta revisión, se aplicaron transformaciones para estandarizar el formato: eliminación de espacios (`str_trim()`), capitalización (`str_to_title()`), y conversión a factor.

A continuación se muestra el código utilizado:

```
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(stringr)

# --------------------------------
# Normalizar Occupation
# --------------------------------
df$Occupation <- df$Occupation %>%
  str_trim() %>%
  str_to_lower() %>%
  str_to_title()

df$Occupation[df$Occupation %in% c("Sales Representative", "Salesperson")] <- "Sales"
df$Occupation <- as.factor(df$Occupation)

# --------------------------------
# Normalizar BMI
# --------------------------------
df$BMI <- df$BMI %>%
  str_trim() %>%
  str_to_lower() %>%
  str_to_title()

df$BMI[df$BMI %in% c("Normal", "Normal Weight")] <- "Normal"
df$BMI <- as.factor(df$BMI)

# --------------------------------
# Normalizar Sleep Disorder (SD)
# --------------------------------
df$SD <- df$SD %>%
  str_trim() %>%
  str_to_lower() %>%
  str_to_title() %>%
  as.factor()

df$Gender <- as.factor(df$Gender)
df$SD <- as.factor(df$SD)


list(
  Gender = levels(df$Gender),
  Occupation = levels(df$Occupation),
  BMI = levels(df$BMI),
  SleepDisorder = levels(df$SD)
)
```

```
## $Gender
## [1] "Female" "Male"
##
## $Occupation
##  [1] "Accountant"        "Doctor"            "Engineer"
##  [4] "Lawyer"            "Manager"           "Nurse"
##  [7] "Sales"             "Scientist"         "Software Engineer"
## [10] "Teacher"
##
```

```
## $BMI
## [1] "Normal"     "Obese"      "Overweight"
##
## $SleepDisorder
## [1] "Insomnia"    "None"        "Sleep Apnea"
```

# 3 Inconsistencias en variables cuantitativas

```
str(df)
```

```
## tibble [374 x 13] (S3: tbl_df/tbl/data.frame)
##  $ ID        : num [1:374] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender    : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Age       : chr [1:374] "27" "28" "28" "28" ...
##  $ Occupation: Factor w/ 10 levels "Accountant","Doctor",..: 9 2 2 7 7 9 10 2 2 2 ...
##  $ SH        : chr [1:374] "6.1" "6.2" "6.2" "5.9" ...
##  $ SQ        : num [1:374] 6 6 6 4 4 4 6 7 7 7 ...
##  $ PHY       : num [1:374] 42 60 60 30 30 30 40 75 75 75 ...
##  $ Stress    : num [1:374] 6 8 8 8 8 8 7 6 6 6 ...
##  $ BMI       : Factor w/ 3 levels "Normal","Obese",..: 3 1 1 2 2 2 2 1 1 1 ...
##  $ BP        : chr [1:374] "126/83" "125/80" "125/80" "140/90" ...
##  $ HR        : num [1:374] 77 75 75 85 85 85 82 70 70 70 ...
##  $ Steps     : num [1:374] 4200 10000 10000 3000 3000 3000 3500 8000 8000 8000 ...
##  $ SD        : Factor w/ 3 levels "Insomnia","None",..: 2 2 2 3 3 1 1 2 2 2 ...
```

Se detectó que las columnas `Age` y `SH` estaban tipadas como texto (`character`) en lugar de numéricas. Se corrigió este error mediante su conversión a tipo `numeric`.

Además, la variable `BP`, que contenía la presión arterial como texto en formato `"sistólica/diastólica"`, fue separada en dos columnas independientes de tipo numérico: `BPsyst` y `BPdias`.

Las demás variables numéricas (`SQ`, `PHY`, `Stress`, `HR`, `Steps`) ya estaban correctamente tipadas.

Los valores no numéricos o vacíos en estas columnas han sido transformados automáticamente a `NA` como resultado de las conversiones.

```
library(dplyr)
library(stringr)

# Convertir columnas mal tipadas a numéricas
df <- df %>%
  mutate(
    Age = as.numeric(Age),
    SH = as.numeric(SH)
  )
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘SH = as.numeric(SH)‘.
## Caused by warning:
## ! NAs introducidos por coerción
```

```r
# Separar presión arterial en dos columnas numéricas
df <- df %>%
  mutate(
    BPsyst = as.numeric(str_extract(BP, "^\\d+")),
    BPdias = as.numeric(str_extract(BP, "(?<=/)\\d+"))
  )

# Eliminar la columna BP original
df <- df %>%
  select(-BP)

str(df)
```

```
## tibble [374 x 14] (S3: tbl_df/tbl/data.frame)
##  $ ID        : num [1:374] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender    : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Age       : num [1:374] 27 28 28 28 28 28 29 29 29 29 ...
##  $ Occupation: Factor w/ 10 levels "Accountant","Doctor",..: 9 2 2 7 7 9 10 2 2 2 ...
##  $ SH        : num [1:374] 6.1 6.2 6.2 5.9 5.9 5.9 6.3 7.8 7.8 7.8 ...
##  $ SQ        : num [1:374] 6 6 6 4 4 4 6 7 7 7 ...
##  $ PHY       : num [1:374] 42 60 60 30 30 30 40 75 75 75 ...
##  $ Stress    : num [1:374] 6 8 8 8 8 8 7 6 6 6 ...
##  $ BMI       : Factor w/ 3 levels "Normal","Obese",..: 3 1 1 2 2 2 2 1 1 1 ...
##  $ HR        : num [1:374] 77 75 75 85 85 85 82 70 70 70 ...
##  $ Steps     : num [1:374] 4200 10000 10000 3000 3000 3000 3500 8000 8000 8000 ...
##  $ SD        : Factor w/ 3 levels "Insomnia","None",..: 2 2 2 3 3 1 1 2 2 2 ...
##  $ BPsyst    : num [1:374] 126 125 125 140 140 140 140 120 120 120 ...
##  $ BPdias    : num [1:374] 83 80 80 90 90 90 90 80 80 80 ...
```

```r
summary(select(df, Age, SH, SQ, PHY, Stress, HR, Steps, BPsyst, BPdias))
```

```
##       Age              SH              SQ             PHY       
##  Min.   : 27.00   Min.   :5.800   Min.   :4.000   Min.   :30.00  
##  1st Qu.: 36.00   1st Qu.:6.400   1st Qu.:6.000   1st Qu.:45.00  
##  Median : 43.00   Median :7.200   Median :7.000   Median :60.00  
##  Mean   : 57.84   Mean   :7.129   Mean   :7.313   Mean   :59.17  
##  3rd Qu.: 50.00   3rd Qu.:7.800   3rd Qu.:8.000   3rd Qu.:75.00  
##  Max.   :999.00   Max.   :8.500   Max.   :9.000   Max.   :90.00  
##  NA's   :3        NA's   :2                                      
##      Stress            HR            Steps           BPsyst      
##  Min.   :3.000   Min.   :65.00   Min.   : 3000   Min.   :115.0  
##  1st Qu.:4.000   1st Qu.:68.00   1st Qu.: 5600   1st Qu.:125.0  
##  Median :5.000   Median :70.00   Median : 7000   Median :130.0  
##  Mean   :5.385   Mean   :70.17   Mean   : 6817   Mean   :128.6  
##  3rd Qu.:7.000   3rd Qu.:72.00   3rd Qu.: 8000   3rd Qu.:135.0  
##  Max.   :8.000   Max.   :86.00   Max.   :10000   Max.   :142.0  
##                                                                 
##      BPdias      
##  Min.   :75.00  
##  1st Qu.:80.00  
##  Median :85.00  
##  Mean   :84.65  
##  3rd Qu.:90.00
```

```
##  Max.   :95.00
##
```

# 4. Valores erróneos o atípicos

## 4.1 Valores erróneos

Se revisaron los valores de las variables numéricas y se identificaron datos erróneos según límites fisiológicos o lógicos. Por ejemplo, se detectó un valor de edad igual a 999, lo cual es inviable, y se sustituyó por `NA`. También se aplicaron las siguientes reglas para asegurar que en posibles futuros datos se eliminen estos valores erroneos:

- Edad debe estar entre 0 y 120 años.
- Horas de sueño entre 0 y 24.
- Calidad del sueño y nivel de estrés en el rango [1, 10].
- Presión arterial y frecuencia cardíaca dentro de rangos fisiológicos.

Los valores que no cumplieron estos criterios fueron transformados en `NA`.

```r
df <- df %>%
  mutate(
    Age = ifelse(Age < 0 | Age > 120, NA, Age),
    SH = ifelse(SH < 0 | SH > 24, NA, SH),
    SQ = ifelse(SQ < 1 | SQ > 10, NA, SQ),
    Stress = ifelse(Stress < 1 | Stress > 10, NA, Stress),
    PHY = ifelse(PHY < 0 | PHY > 1440, NA, PHY),
    HR = ifelse(HR < 30 | HR > 200, NA, HR),
    Steps = ifelse(Steps < 0, NA, Steps),
    BPsyst = ifelse(BPsyst < 50 | BPsyst > 250, NA, BPsyst),
    BPdias = ifelse(BPdias < 30 | BPdias > 150, NA, BPdias)
  )
```

## 4.2 Valores atípicos

```r
summary(select(df, Age, SH, SQ, PHY, Stress, HR, Steps, BPsyst, BPdias))
```

```
##       Age              SH              SQ             PHY
##  Min.   :27.00   Min.   :5.800   Min.   :4.000   Min.   :30.00
##  1st Qu.:36.00   1st Qu.:6.400   1st Qu.:6.000   1st Qu.:45.00
##  Median :43.00   Median :7.200   Median :7.000   Median :60.00
##  Mean   :42.37   Mean   :7.129   Mean   :7.313   Mean   :59.17
##  3rd Qu.:50.00   3rd Qu.:7.800   3rd Qu.:8.000   3rd Qu.:75.00
##  Max.   :73.00   Max.   :8.500   Max.   :9.000   Max.   :90.00
##  NA's   :9       NA's   :2
##      Stress            HR            Steps           BPsyst
##  Min.   :3.000   Min.   :65.00   Min.   : 3000   Min.   :115.0
##  1st Qu.:4.000   1st Qu.:68.00   1st Qu.: 5600   1st Qu.:125.0
##  Median :5.000   Median :70.00   Median : 7000   Median :130.0
##  Mean   :5.385   Mean   :70.17   Mean   : 6817   Mean   :128.6
```

```
##  3rd Qu.:7.000   3rd Qu.:72.00   3rd Qu.: 8000   3rd Qu.:135.0
##  Max.   :8.000   Max.   :86.00   Max.   :10000   Max.   :142.0
##
##      BPdias
##  Min.   :75.00
##  1st Qu.:80.00
##  Median :85.00
##  Mean   :84.65
##  3rd Qu.:90.00
##  Max.   :95.00
##
```

Se analizaron los valores atípicos mediante el método del rango intercuartílico (IQR) aplicado a las variables cuantitativas. Tras observar los valores mínimos, máximos y los cuartiles, se concluyó que todos los valores se encuentran dentro de rangos razonables, tanto a nivel estadístico como fisiológico.

En consecuencia, no fue necesario eliminar ningún valor como atípico.

# 5. Imputación

```
colSums(is.na(df[, c("Age", "SH", "SQ", "Stress", "PHY", "HR", "Steps", "BPsyst", "BPdias")]))
```

```
##    Age     SH     SQ Stress    PHY     HR  Steps BPsyst BPdias
##      9      2      0      0      0      0      0      0      0
```

Se detectaron valores faltantes únicamente en las variables `Age` (9 casos) y `SH` (2 casos).
Para completar estos valores, se utilizó el método de imputación por k vecinos más cercanos (k = 3), mediante la función `kNN()` del paquete `VIM`.

```
library(VIM)
```

```
## Cargando paquete requerido: colorspace

## Cargando paquete requerido: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Adjuntando el paquete: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
# Imputamos solo Age y SH con k = 3
df_imputed <- kNN(df, variable = c("Age", "SH"), k = 3, imp_var = TRUE)

# Mostrar las filas donde se imputó Age o SH
df_imputed[which(df_imputed$Age_imp | df_imputed$SH_imp),
           c("ID", "Age", "Age_imp", "SH", "SH_imp")]
```

```
##      ID Age Age_imp  SH SH_imp
## 25   25  30   FALSE 7.7   TRUE
## 42   42  31    TRUE 7.7  FALSE
## 52   52  37    TRUE 7.5  FALSE
## 54   54  32   FALSE 7.7   TRUE
## 76   76  33    TRUE 6.0  FALSE
## 87   87  35    TRUE 7.2  FALSE
## 154 154  39    TRUE 7.2  FALSE
## 180 180  42    TRUE 7.8  FALSE
## 245 245  44    TRUE 6.3  FALSE
## 305 305  50    TRUE 6.1  FALSE
## 334 334  54    TRUE 8.4  FALSE
```

```r
colSums(is.na(df_imputed[, c("Age", "SH", "SQ", "Stress", "PHY", "HR", "Steps", "BPsyst", "BPdias")]))
```

```
##    Age     SH     SQ Stress    PHY     HR  Steps BPsyst BPdias
##      0      0      0      0      0      0      0      0      0
```

# 6. Correlaciones

## 6.1 Matriz de correlaciones

Se calculó la matriz de correlaciones de Pearson entre las variables numéricas. A continuación se muestran los resultados redondeados:

```r
numeric_vars <- df_imputed[, c("Age", "SH", "SQ", "PHY", "Stress", "HR", "Steps", "BPsyst", "BPdias")]

# Calculamos la matriz de correlaciones de Pearson
cor_matrix <- cor(numeric_vars)

round(cor_matrix, 2)
```

```
##          Age    SH    SQ   PHY Stress    HR Steps BPsyst BPdias
## Age     1.00  0.34  0.48  0.17  -0.43 -0.23  0.05   0.57   0.56
## SH      0.34  1.00  0.88  0.21  -0.81 -0.52 -0.04  -0.18  -0.17
## SQ      0.48  0.88  1.00  0.19  -0.90 -0.66  0.02  -0.12  -0.11
## PHY     0.17  0.21  0.19  1.00  -0.03  0.14  0.77   0.27   0.38
## Stress -0.43 -0.81 -0.90 -0.03   1.00  0.67  0.19   0.10   0.09
## HR     -0.23 -0.52 -0.66  0.14   0.67  1.00 -0.03   0.29   0.27
## Steps   0.05 -0.04  0.02  0.77   0.19 -0.03  1.00   0.10   0.24
## BPsyst  0.57 -0.18 -0.12  0.27   0.10  0.29  0.10   1.00   0.97
## BPdias  0.56 -0.17 -0.11  0.38   0.09  0.27  0.24   0.97   1.00
```

Interpretación de resultados:

- Se observa una fuerte correlación positiva entre SH (Sleep Hours) y SQ (Sleep Quality), lo cual indica que más horas de sueño se relacionan con una mejor calidad.
- Stress muestra una alta correlación negativa con SQ, lo que concuerda con la intuición: más estrés implica peor calidad del sueño.

- Las variables BPsyst y BPdias están fuertemente correlacionadas (r = 0.97), como es habitual en medidas de presión arterial.
- También hay una correlación moderada entre Age y la presión arterial.

## 6.2 Cálculo de correlaciones

Se implementó manualmente la fórmula de la correlación de Pearson entre dos variables numéricas:

$$r = \cos(\alpha) = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Figure 1: Fórmula de correlación de Pearson

Esta fórmula calcula la covarianza entre las dos variables dividida por el producto de sus desviaciones estándar. A partir de ella se construyó la siguiente función en R:

```
cor_pearson <- function(x, y) {
  x <- as.numeric(x)
  y <- as.numeric(y)

  x_mean <- mean(x)
  y_mean <- mean(y)

  numerator <- sum((x - x_mean) * (y - y_mean))

  denominator <- sqrt(sum((x - x_mean)^2) * sum((y - y_mean)^2))

  r <- numerator / denominator
  return(r)
}
```

Se comparó el resultado de la función con el resultado nativo de R:

```
manual <- cor_pearson(df_imputed$SH, df_imputed$SQ)
builtin <- cor(df_imputed$SH, df_imputed$SQ)

manual
```

```
## [1] 0.8832504
```

```
builtin
```

```
## [1] 0.8832504
```

Ambos resultados coinciden, validando la implementación.

# 7. Análisis descriptivo y visual

## 7.1 Tabla resumen de tendencia central y variabilidad de HR según SD (Sleep Disorder)

Se calculó la media y la desviación estándar de las variables `SH`, `SQ`, `HR` y `Stress`, agrupadas según el tipo de trastorno del sueño (`SD`). Esto permite observar diferencias entre personas con y sin trastornos como insomnio o apnea del sueño.

```
library(dplyr)
library(kableExtra)
```

```
##
## Adjuntando el paquete: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
tabla_resumen <- df_imputed %>%
  group_by(SD) %>%
  summarise(
    SH_media = mean(SH, na.rm = TRUE),
    SH_sd = sd(SH, na.rm = TRUE),

    SQ_media = mean(SQ, na.rm = TRUE),
    SQ_sd = sd(SQ, na.rm = TRUE),

    HR_media = mean(HR, na.rm = TRUE),
    HR_sd = sd(HR, na.rm = TRUE),

    Stress_media = mean(Stress, na.rm = TRUE),
    Stress_sd = sd(Stress, na.rm = TRUE)
  )

tabla_resumen %>%
  kable(digits = 2, caption = "Resumen estadístico por tipo de trastorno del sueño") %>%
  kable_styling(full_width = FALSE)
```

Table 1: Resumen estadístico por tipo de trastorno del sueño

| SD | SH_media | SH_sd | SQ_media | SQ_sd | HR_media | HR_sd | Stress_media | Stress_sd |
|---|---|---|---|---|---|---|---|---|
| Insomnia | 6.59 | 0.39 | 6.53 | 0.80 | 70.47 | 4.95 | 5.87 | 1.46 |
| None | 7.36 | 0.73 | 7.63 | 0.98 | 69.02 | 2.66 | 5.11 | 1.59 |
| Sleep Apnea | 7.03 | 0.97 | 7.21 | 1.65 | 73.09 | 5.12 | 5.67 | 2.33 |

La tabla muestra cómo varían las medidas de sueño, frecuencia cardíaca y estrés según el tipo de trastorno del sueño (`SD`):

- Las personas con insomnio presentan menos horas de sueño, peor calidad, mayor frecuencia cardíaca y más estrés.

- En el caso de sleep apnea, aunque las horas de sueño son similares a las personas sin trastorno, la calidad del sueño y la frecuencia cardíaca se ven afectadas.
- Las personas sin trastorno obtienen los mejores resultados en todas las métricas, como era esperable.

Estos resultados reflejan el impacto negativo que los trastornos del sueño tienen sobre la salud general y el bienestar.

## 7.2 Gráfico de medias según HR y Occupation

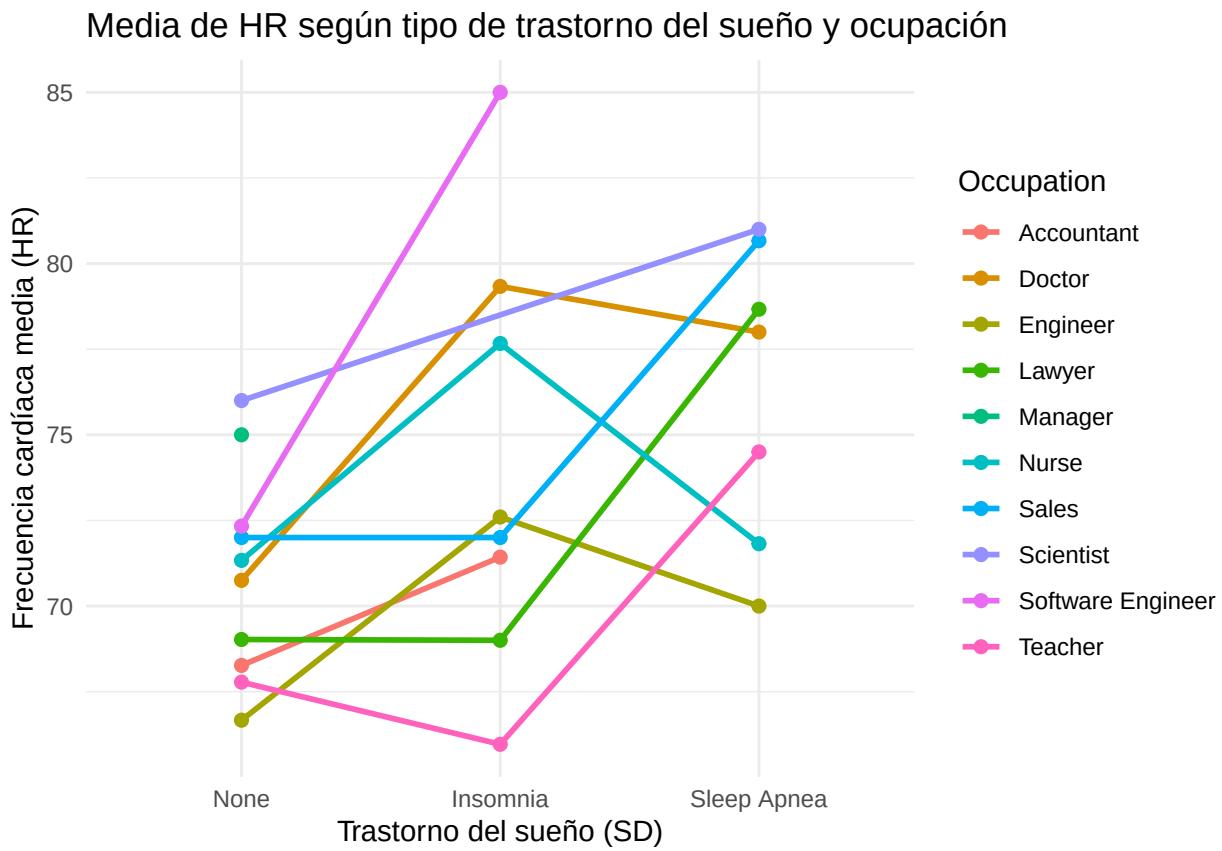
```
library(ggplot2)

df_plot <- df_imputed %>%
  group_by(SD, Occupation) %>%
  summarise(HR_media = mean(HR, na.rm = TRUE), .groups = "drop")

df_plot$SD <- factor(df_plot$SD, levels = c("None", "Insomnia", "Sleep Apnea"))

ggplot(df_plot, aes(x = SD, y = HR_media, group = Occupation, color = Occupation)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Media de HR según tipo de trastorno del sueño y ocupación",
    x = "Trastorno del sueño (SD)",
    y = "Frecuencia cardíaca media (HR)"
  ) +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




Se observa cómo la frecuencia cardíaca media varía en función del tipo de trastorno del sueño (`SD`) y según la ocupación. Algunas profesiones como `Software Engineer` presentan picos de HR elevados ante casos de `Insomnia`, lo que podría estar relacionado con el estrés o la carga mental del trabajo.

Este gráfico permite visualizar de forma clara la relación entre ocupación, calidad del sueño y frecuencia cardíaca, lo que puede ser útil para análisis posteriores de bienestar y salud laboral.

```r
write.csv(df_imputed, "sleephealth_processed.csv", row.names = FALSE)
```