final report

Chu,Li,Hsu

2025-01-09

Table of contents

```r
library(showtext)
```

```
Loading required package: sysfonts

Loading required package: showtextdb
```

```r
showtext_auto()  # 啟用 showtext
font_add("Microsoft JhengHei UI", "C:/Windows/Fonts/msjh.ttc")  # 添加你使用的字體
```

```r
library(Hmisc)
library(skimr)
library(DataExplorer)
library(ggplot2)
library(dplyr)
library(corrplot)
library(GGally)
library(plotly)
library(gridExtra)
library(knitr)
library(car)
#setwd("C:/Users/anya3/Downloads")
#setwd("C:\\Users\\user\\Downloads\\統諮期末\\統諮期末\\統諮期末 1226")
setwd("C:/Users/User/OneDrive/桌面/統諮期末")
data <- read.csv("Sleep_health_and_lifestyle_dataset.csv")
```

1. Conduct necessary data preprocessing

敘述性統計/missing values 診斷

```r
# Check structure of the dataset
dim(data)
```

[1] 374 13

```r
names(data)
```

[1] "Person.ID" "Gender"
[3] "Age" "Occupation"
[5] "Sleep.Duration" "Quality.of.Sleep"
[7] "Physical.Activity.Level" "Stress.Level"
[9] "BMI.Category" "Blood.Pressure"

[11] "Heart.Rate" "Daily.Steps"
[13] "Sleep.Disorder"

```r
data$Occupation <- as.factor(data$Occupation)
latex(describe(data), file="")
```

data
13 Variables 374 Observations

Person.ID

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	374	1	187.5	125	19.65	38.30	94.25	187.50	280.75	336.70	355.35

```
lowest :   1   2   3   4   5, highest: 370 371 372 373 374
```

Gender

n	missing	distinct
374	0	2

```
Value      Female   Male
Frequency     185    189
Proportion  0.495  0.505
```

Age

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	31	0.997	42.18	9.933	29.65	31.00	35.25	43.00	50.00	54.00	58.00

```
lowest : 27 28 29 30 31, highest: 55 56 57 58 59
```

Occupation

n	missing	distinct
374	0	11

```
lowest : Accountant           Doctor               Engineer             Lawyer               Manager
highest: Sales Representative Salesperson          Scientist            Software Engineer    Teacher
```

Sleep.Duration

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	27	0.997	7.132	0.9153	6.0	6.1	6.4	7.2	7.8	8.2	8.4

```
lowest : 5.8 5.9 6   6.1 6.2, highest: 8.1 8.2 8.3 8.4 8.5
```

Quality.of.Sleep

n	missing	distinct	Info	Mean	Gmd
374	0	6	0.938	7.313	1.329

```
Value          4     5     6     7     8     9
Frequency      5     7   105    77   109    71
Proportion 0.013 0.019 0.281 0.206 0.291 0.190

For the frequency table, variable is rounded to the nearest 0
```

Physical.Activity.Level

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	16	0.97	59.17	23.69	30	30	45	60	75	90	90

```
Value         30    32    35    40    42    45    47    50    55    60    65    70    75    80
Frequency     68     2     4     6     2    68     1     4     6    70     2     3    67     2
Proportion 0.182 0.005 0.011 0.016 0.005 0.182 0.003 0.011 0.016 0.187 0.005 0.008 0.179 0.005

Value         85    90
Frequency      2    67
Proportion 0.005 0.179

For the frequency table, variable is rounded to the nearest 0
```

Stress.Level

n	missing	distinct	Info	Mean	Gmd
374	0	6	0.97	5.385	2.017

```
Value          3     4     5     6     7     8
Frequency     71    70    67    46    50    70
Proportion 0.190 0.187 0.179 0.123 0.134 0.187

For the frequency table, variable is rounded to the nearest 0
```

BMI.Category

n	missing	distinct
374	0	4

```
Value             Normal Normal Weight         Obese    Overweight
Frequency            195            21            10           148
Proportion         0.521         0.056         0.027         0.396
```

Blood.Pressure

n	missing	distinct
374	0	25

```
lowest : 115/75 115/78 117/76 118/75 118/76, highest: 135/90 139/91 140/90 140/95 142/92
```

Heart.Rate

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	19	0.963	70.17	4.353	65	65	68	70	72	75	78

```
Value         65    67    68    69    70    72    73    74    75    76    77    78    80    81
Frequency     67     2    94     2    76    69     2     2    36     2     2     5     3     2
Proportion 0.179 0.005 0.251 0.005 0.203 0.184 0.005 0.005 0.096 0.005 0.005 0.013 0.008 0.005

Value         82    83    84    85    86
Frequency      1     2     2     3     2
Proportion 0.003 0.005 0.005 0.008 0.005

For the frequency table, variable is rounded to the nearest 0
```

Daily.Steps

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	20	0.962	6817	1801	4930	5000	5600	7000	8000	8000	10000

```
Value       3000  3300  3500  3700  4000  4100  4200  4800  5000  5200  5500  5600  6000  6200
Frequency      3     2     3     2     3     2     2     2    68     2     4     2    68     1
Proportion 0.008 0.005 0.008 0.005 0.008 0.005 0.005 0.005 0.182 0.005 0.011 0.005 0.182 0.003

Value       6800  7000  7300  7500  8000 10000
Frequency      3    66     2     2   101    36
Proportion 0.008 0.176 0.005 0.005 0.270 0.096

For the frequency table, variable is rounded to the nearest 0
```

Sleep.Disorder

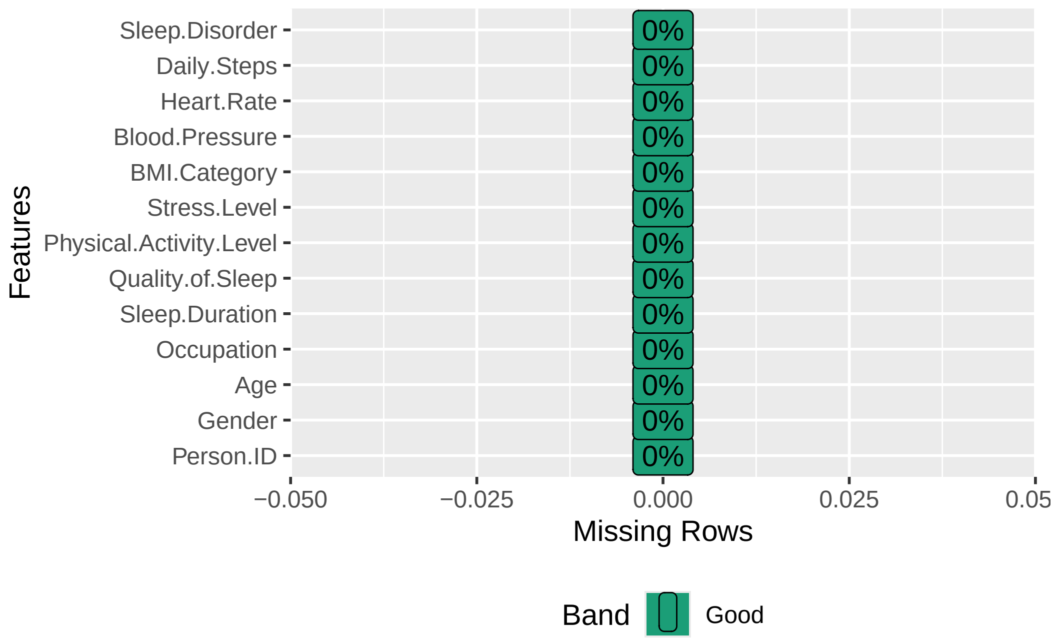
n	missing	distinct
374	0	3

```
Value         Insomnia        None Sleep Apnea
Frequency           77         219          78
Proportion       0.206       0.586       0.209
```

```r
sum(is.na(data))
```

[1] 0

```r
plot_missing(data)
```



此筆資料集共有 374 筆資料，13 個變數且無缺失值

變數解釋表

```
summary_table <- data %>%
  summarise(
    Variable = c(
      "Person ID",
      "Gender",
      "Age",
      "Occupation",
      "Sleep Duration",
      "Quality of Sleep",
      "Physical Activity Level",
      "Stress Level",
      "BMI Category",
      "Blood Pressure",
      "Heart Rate",
      "Daily Steps",
      "Sleep Disorder"
    ),
    Description = c(
      "編號",
      "性別",
      "年齡",
      "職業",
      "每日睡眠時長（小時）",
      "主觀認定之睡眠品質",
      "身體活動量",
      "主觀認定之壓力程度",
```

```r
      "BMI 類別",
      " 血壓",
      " 脈搏",
      " 每日步數",
      " 睡眠疾病"
    ),
    remark=c(
      "1-374",
      "Male/Female",
      "27-59 歲",
      "11 種",
      "5.8-8.5",
      "4-9,(scale: 1-10)",
      "30-90",
      "3-8,(scale: 1-10)",
      "Normal/Normal Weight/Obese/Overweight",
      "Systolic 收縮壓/Diastolic 舒張壓",
      "65-86",
      "3000-10000",
      "None/Insomnia 失眠/Apnea 睡眠呼吸暫停"
    )
  )
kable(summary_table, format = "markdown", digits = 2, caption = " 變數解釋")
```

Table 1: 變數解釋

Variable	Description	remark
Person ID	編號	1-374
Gender	性別	Male/Female
Age	年齡	27-59 歲
Occupation	職業	11 種
Sleep Duration	每日睡眠時長 (小時)	5.8-8.5
Quality of Sleep	主觀認定之睡眠品質	4-9,(scale: 1-10)
Physical Activity Level	身體活動量	30-90
Stress Level	主觀認定之壓力程度	3-8,(scale: 1-10)
BMI Category	BMI 類別	Normal/Normal Weight/Obese/Overweight
Blood Pressure	血壓	Systolic 收縮壓/Diastolic 舒張壓
Heart Rate	脈搏	65-86
Daily Steps	每日步數	3000-10000
Sleep Disorder	睡眠疾病	None/Insomnia 失眠/Apnea 睡眠呼吸暫停

資料前處理 - 變數處理 (刪除、分類)

```r
# 刪除 Person ID
data <- data %>% dplyr::select(-`Person.ID`)
```

```r
# 刪除血壓中的舒張壓
data <- data %>%
  tidyr::separate(col = `Blood.Pressure`,
                  into = c("Blood.Pressure", "BloodPressure_Lower"),
                  sep = "/",
                  convert = TRUE) # convert=TRUE 會自動轉換為數值型別
data <- data %>% dplyr::select(-`BloodPressure_Lower`)

# 分類 physical activity level
data$Physical.Activity.Level<-ifelse(data$Physical.Activity.Level<=45,"<=45",
                              ifelse(data$Physical.Activity.Level<=60,"45~60",
                              ifelse(data$Physical.Activity.Level<=75,"60~75",
                              "75~90")))
# 分類 daily steps
data$Daily.Steps <- ifelse(data$Daily.Steps<=5000,"<=5000",
                    ifelse(data$Daily.Steps<=6000,"5001~7500","7500up"))


# 將睡眠疾病->0,1
data$Sleep.Disorder <- ifelse(data$Sleep.Disorder=="None",0,1)

# 分類 BMI
data$BMI.Category <- ifelse(data$BMI.Category == "Normal Weight","Normal",
                            data$BMI.Category)
data$BMI.Category <- ifelse(data$BMI.Category == "Obese","Overweight",
                            data$BMI.Category)

# 分類 quality of sleep
data$Quality.of.Sleep <- ifelse(data$Quality.of.Sleep==4 |
                                data$Quality.of.Sleep==5,"4-5",
                                data$Quality.of.Sleep)

# 分類 occupation
data$Occupation <- ifelse(data$Occupation=="Manager" |
                          data$Occupation=="Sales Representative" ,
                          "Salesperson",data$Occupation)
data$Occupation <- ifelse(data$Occupation=="Software Engineer" ,
                          "Engineer",data$Occupation)
```

Encoding Categorical Variables

```r
data$Gender <- as.factor(data$Gender)
data$Occupation <- as.factor(data$Occupation)
data$Quality.of.Sleep <- as.factor(data$Quality.of.Sleep)
data$Stress.Level <- as.factor(data$Stress.Level)
data$BMI.Category <- as.factor(data$BMI.Category)
data$Sleep.Disorder <- as.factor(data$Sleep.Disorder)
```

```r
data$Physical.Activity.Level <- as.factor(data$Physical.Activity.Level)
data$Daily.Steps <- as.factor(data$Daily.Steps)
```

處理後的資料

```r
# Check structure of the dataset
latex(describe(data), file="")
```

data

12 Variables 374 Observations

Gender

n	missing	distinct
374	0	2

Value	Female	Male
Frequency	185	189
Proportion	0.495	0.505

Age

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	31	0.997	42.18	9.933	29.65	31.00	35.25	43.00	50.00	54.00	58.00

lowest : 27 28 29 30 31, highest: 55 56 57 58 59

Occupation

n	missing	distinct
374	0	10

Value	1	10	11	2	3	4	6
Frequency	37	4	40	71	63	47	73
Proportion	0.099	0.011	0.107	0.190	0.168	0.126	0.195

Value	8	9	Salesperson
Frequency	32	4	3
Proportion	0.086	0.011	0.008

Sleep.Duration

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	27	0.997	7.132	0.9153	6.0	6.1	6.4	7.2	7.8	8.2	8.4

lowest : 5.8 5.9 6   6.1 6.2, highest: 8.1 8.2 8.3 8.4 8.5

Quality.of.Sleep

n	missing	distinct
374	0	5

Value	4-5	6	7	8	9
Frequency	12	105	77	109	71
Proportion	0.032	0.281	0.206	0.291	0.190

Physical.Activity.Level

n	missing	distinct
374	0	4

Value	<=45	45~60	60~75	75~90
Frequency	150	81	72	71
Proportion	0.401	0.217	0.193	0.190

Stress.Level

n	missing	distinct
374	0	6

Value	3	4	5	6	7	8
Frequency	71	70	67	46	50	70
Proportion	0.190	0.187	0.179	0.123	0.134	0.187

BMI.Category

n	missing	distinct
374	0	2

```
Value          Normal Overweight
Frequency         216        158
Proportion      0.578      0.422
```

Blood.Pressure

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	18	0.965	128.6	8.74	115	118	125	130	135	140	140

```
Value        115   117   118   119   120   121   122   125   126   128   129   130   131   132
Frequency     34     2     3     2    45     1     1    69     2     5     2   101     2     3
Proportion 0.091 0.005 0.008 0.005 0.120 0.003 0.003 0.184 0.005 0.013 0.005 0.270 0.005 0.008

Value        135   139   140   142
Frequency     29     2    69     2
Proportion 0.078 0.005 0.184 0.005

For the frequency table, variable is rounded to the nearest 0
```

Heart.Rate

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
374	0	19	0.963	70.17	4.353	65	65	68	70	72	75	78

```
Value         65    67    68    69    70    72    73    74    75    76    77    78    80    81
Frequency     67     2    94     2    76    69     2     2    36     2     2     5     3     2
Proportion 0.179 0.005 0.251 0.005 0.203 0.184 0.005 0.005 0.096 0.005 0.005 0.013 0.008 0.005

Value         82    83    84    85    86
Frequency      1     2     2     3     2
Proportion 0.003 0.005 0.005 0.008 0.005

For the frequency table, variable is rounded to the nearest 0
```

Daily.Steps

n	missing	distinct
374	0	3

```
Value         <=5000 5001~7500    7500up
Frequency         87        76       211
Proportion     0.233     0.203     0.564
```

Sleep.Disorder

n	missing	distinct
374	0	2

```
Value          0     1
Frequency    219   155
Proportion 0.586 0.414
```

共 11 個自變數 (分別有 7 個類別變數以及 4 個連續變數)

用來預測一個應變數-是否有睡眠疾病 (類別變數)

Table one

```r
library(tableone)

# 定義變數
categorical_vars <- c('Gender','Occupation','Quality.of.Sleep',
                      'Physical.Activity.Level','Stress.Level',
                      'BMI.Category','Daily.Steps')
continuous_vars <- c('Blood.Pressure','Age','Sleep.Duration','Heart.Rate')

# 分組
```

```r
group_var <- "Sleep.Disorder"

# 建立 Table One
table_one <- CreateTableOne(vars = c(categorical_vars, continuous_vars),
                            strata = group_var,
                            data = data,
                            factorVars = categorical_vars,
                            addOverall = TRUE)

# Table One
print(table_one,showAllLevels = TRUE)
```

```
                            Stratified by Sleep.Disorder
                             level        Overall        0
  n                                          374            219
  Gender (%)                 Female          185 (49.5)      82 (37.4)
                             Male            189 (50.5)     137 (62.6)
  Occupation (%)             1                37 ( 9.9)      30 (13.7)
                             10                4 ( 1.1)       3 ( 1.4)
                             11               40 (10.7)       9 ( 4.1)
                             2                71 (19.0)      64 (29.2)
                             3                63 (16.8)      57 (26.0)
                             4                47 (12.6)      42 (19.2)
                             6                73 (19.5)       9 ( 4.1)
                             8                32 ( 8.6)       2 ( 0.9)
                             9                 4 ( 1.1)       2 ( 0.9)
                             Salesperson       3 ( 0.8)       1 ( 0.5)
  Quality.of.Sleep (%)       4-5              12 ( 3.2)       0 ( 0.0)
                             6               105 (28.1)      40 (18.3)
                             7                77 (20.6)      40 (18.3)
                             8               109 (29.1)     101 (46.1)
                             9                71 (19.0)      38 (17.4)
  Physical.Activity.Level (%) <=45           150 (40.1)      70 (32.0)
                             45~60            81 (21.7)      75 (34.2)
                             60~75            72 (19.3)      39 (17.8)
                             75~90            71 (19.0)      35 (16.0)
  Stress.Level (%)           3                71 (19.0)      40 (18.3)
                             4                70 (18.7)      43 (19.6)
                             5                67 (17.9)      57 (26.0)
                             6                46 (12.3)      43 (19.6)
                             7                50 (13.4)       3 ( 1.4)
                             8                70 (18.7)      33 (15.1)
  BMI.Category (%)           Normal          216 (57.8)     200 (91.3)
                             Overweight      158 (42.2)      19 ( 8.7)
  Daily.Steps (%)            <=5000           87 (23.3)      63 (28.8)
                             5001~7500        76 (20.3)      13 ( 5.9)
                             7500up          211 (56.4)     143 (65.3)
  Blood.Pressure (mean (SD))              128.55 (7.75)  124.05 (5.73)
```

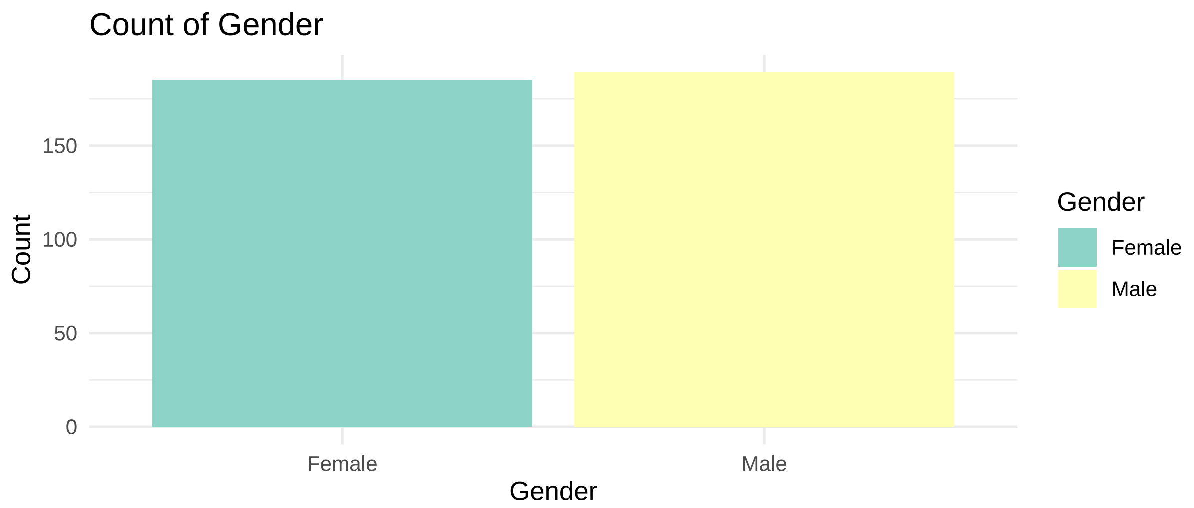
```
Age (mean (SD))                          42.18 (8.67)   39.04 (7.83)
Sleep.Duration (mean (SD))                7.13 (0.80)    7.36 (0.73)
Heart.Rate (mean (SD))                   70.17 (4.14)   69.02 (2.66)
                           Stratified by Sleep.Disorder
                            1              p      test
n                              155
Gender (%)                     103 (66.5)  <0.001
                                52 (33.5)
Occupation (%)                   7 ( 4.5)  <0.001
                                 1 ( 0.6)
                                31 (20.0)
                                 7 ( 4.5)
                                 6 ( 3.9)
                                 5 ( 3.2)
                                64 (41.3)
                                30 (19.4)
                                 2 ( 1.3)
                                 2 ( 1.3)
Quality.of.Sleep (%)            12 ( 7.7)  <0.001
                                65 (41.9)
                                37 (23.9)
                                 8 ( 5.2)
                                33 (21.3)
Physical.Activity.Level (%)     80 (51.6)  <0.001
                                 6 ( 3.9)
                                33 (21.3)
                                36 (23.2)
Stress.Level (%)                31 (20.0)  <0.001
                                27 (17.4)
                                10 ( 6.5)
                                 3 ( 1.9)
                                47 (30.3)
                                37 (23.9)
BMI.Category (%)                16 (10.3)  <0.001
                               139 (89.7)
Daily.Steps (%)                 24 (15.5)  <0.001
                                63 (40.6)
                                68 (43.9)
Blood.Pressure (mean (SD))  134.92 (5.40)  <0.001
Age (mean (SD))              46.63 (7.84)  <0.001
Sleep.Duration (mean (SD))    6.81 (0.77)  <0.001
Heart.Rate (mean (SD))       71.79 (5.19)  <0.001
```

2. EDA

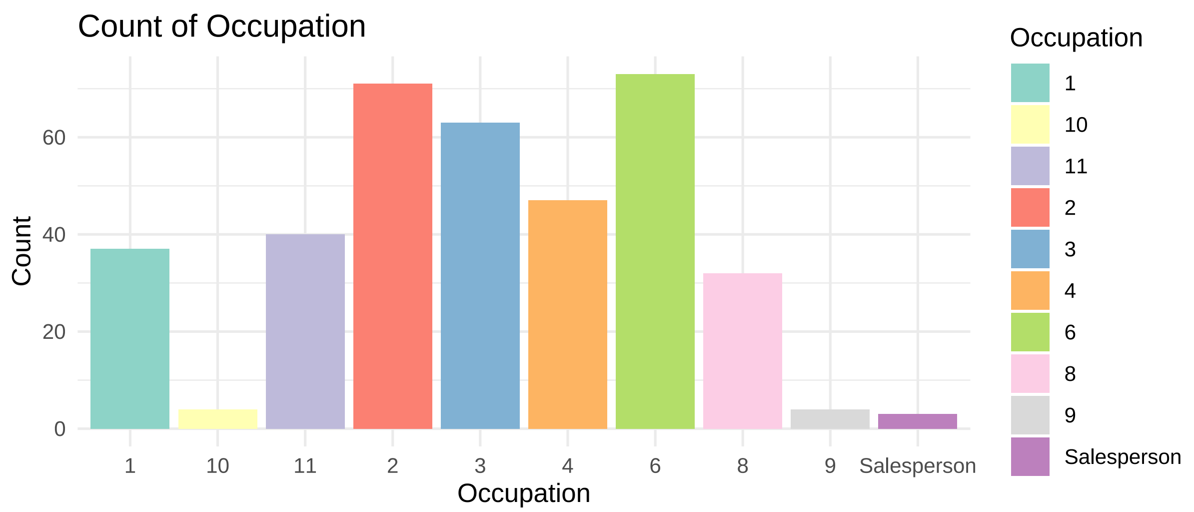
Distribution of the data

i.categorical variable

```
ggplot(data, aes(x = Gender, fill = Gender)) +
  geom_bar() +
  labs(title = "Count of Gender", x = "Gender", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```

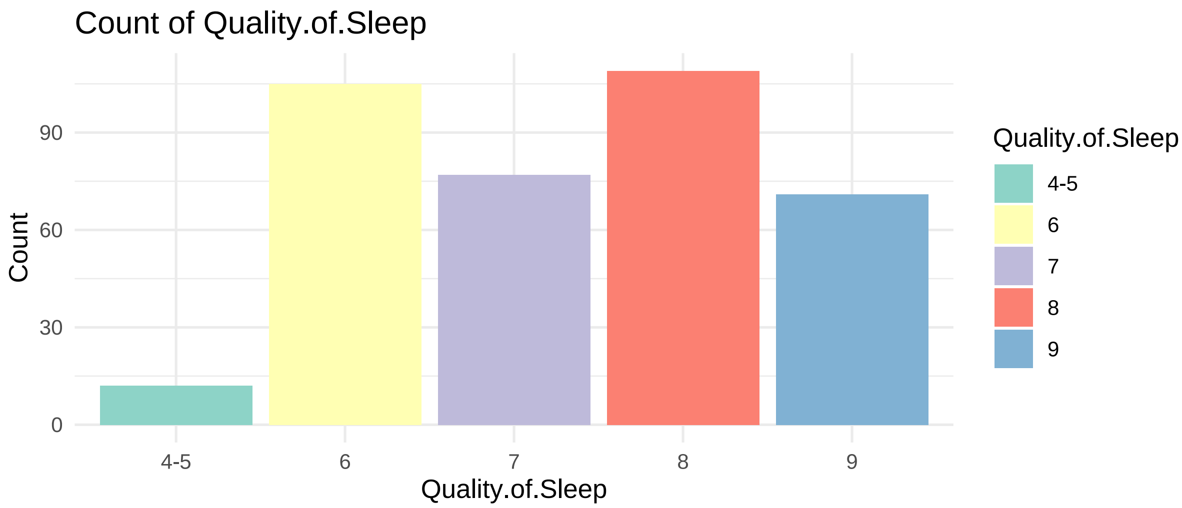



```
ggplot(data, aes(x = Occupation, fill = Occupation)) +
  geom_bar() +
  labs(title = "Count of Occupation", x = "Occupation", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```

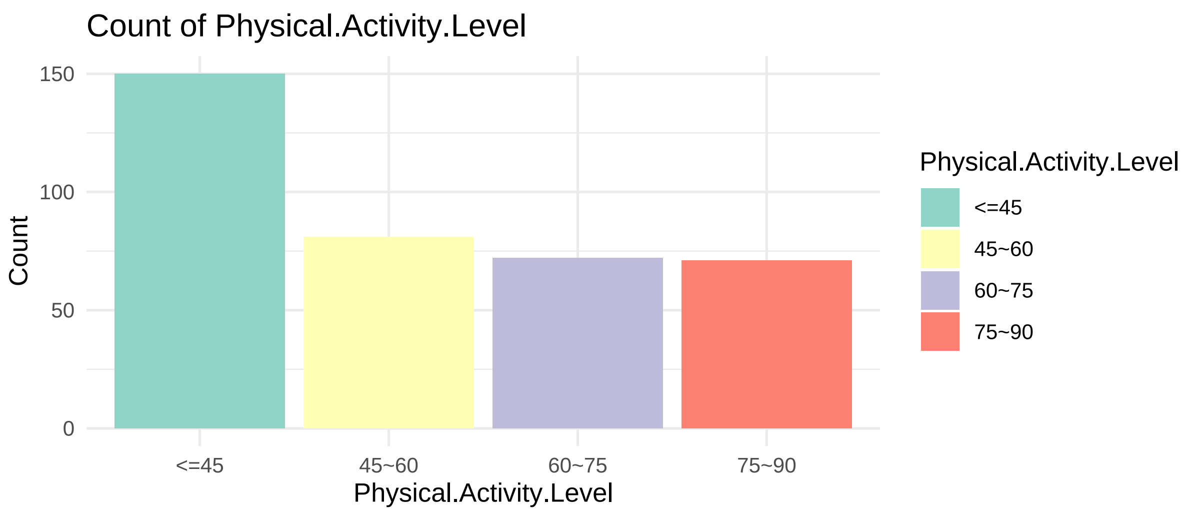



```
ggplot(data, aes(x = Quality.of.Sleep, fill = Quality.of.Sleep)) +
  geom_bar() +
  labs(title = "Count of Quality.of.Sleep", x = "Quality.of.Sleep", y = "Count") +
```

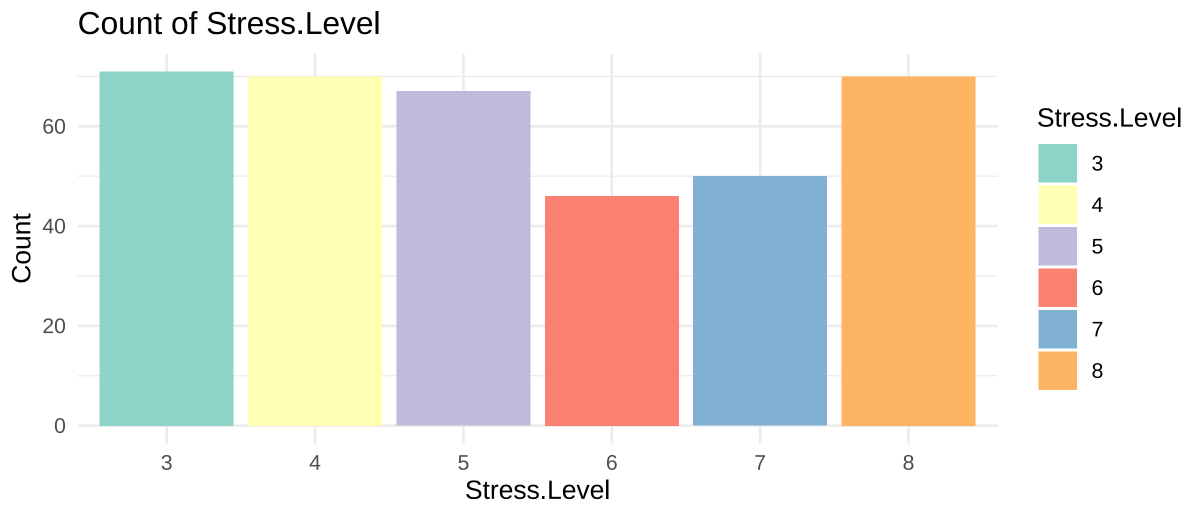
```r
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```



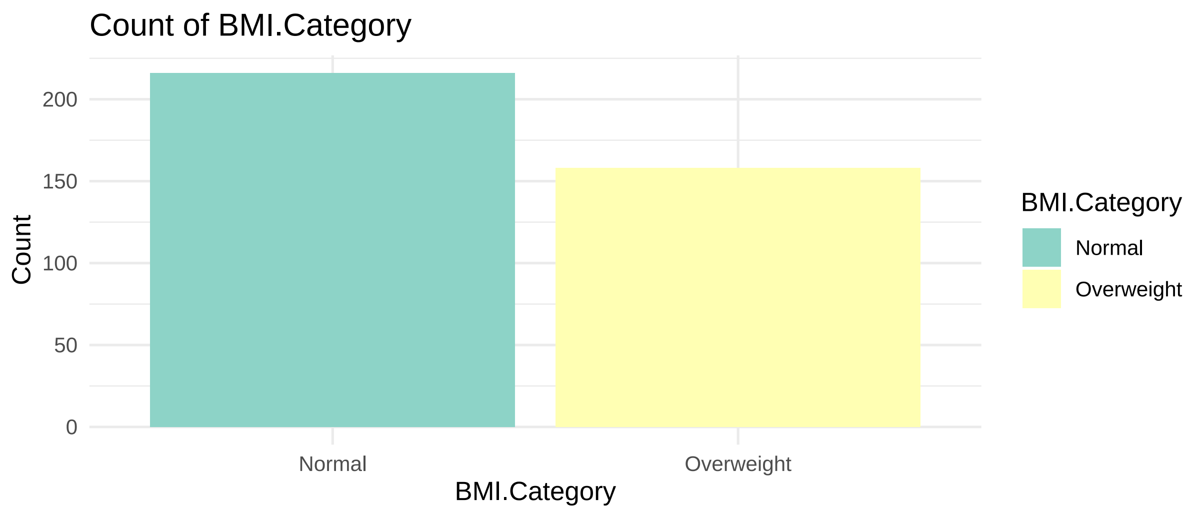

```r
ggplot(data,
  aes(x = Physical.Activity.Level, fill = Physical.Activity.Level)) +
  geom_bar() +
  labs(title = "Count of Physical.Activity.Level",
  x = "Physical.Activity.Level", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```



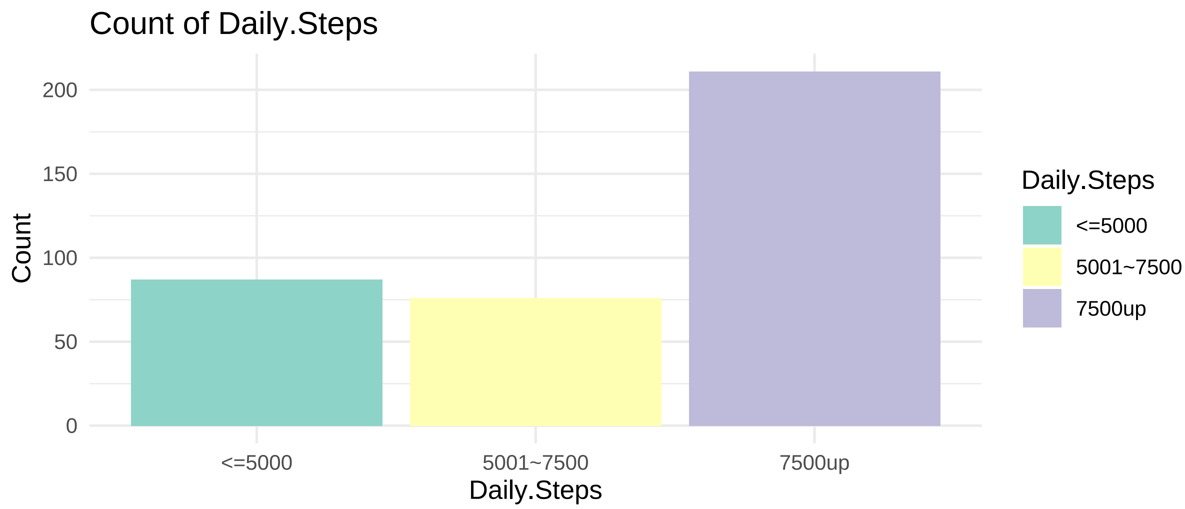

```r
ggplot(data, aes(x = Stress.Level, fill = Stress.Level)) +
  geom_bar() +
  labs(title = "Count of Stress.Level", x = "Stress.Level", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```




```r
ggplot(data, aes(x = BMI.Category, fill = BMI.Category)) +
  geom_bar() +
  labs(title = "Count of BMI.Category", x = "BMI.Category", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```

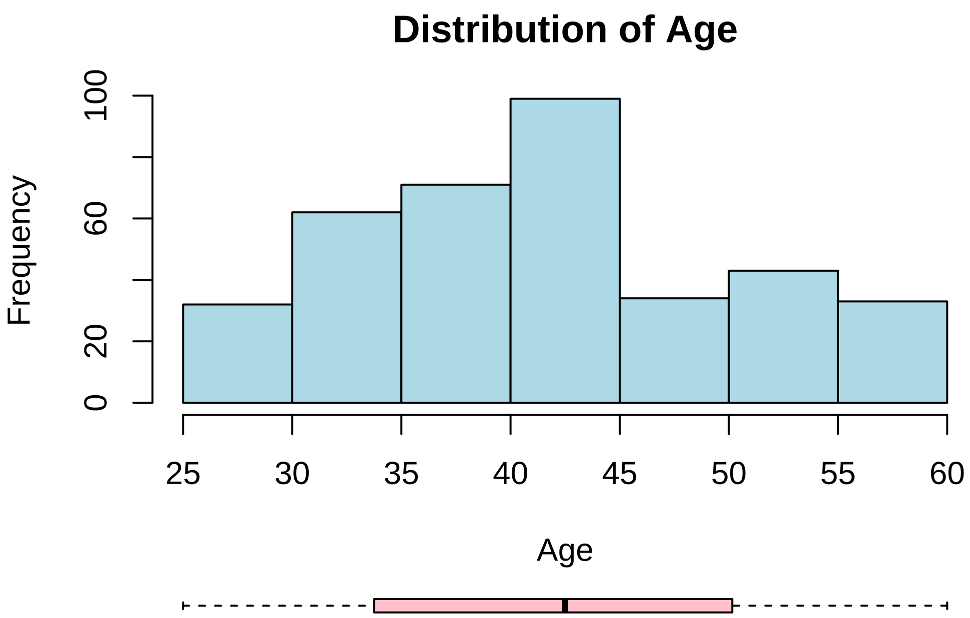



```r
ggplot(data, aes(x = Daily.Steps, fill = Daily.Steps)) +
  geom_bar() +
  labs(title = "Count of Daily.Steps", x = "Daily.Steps", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```



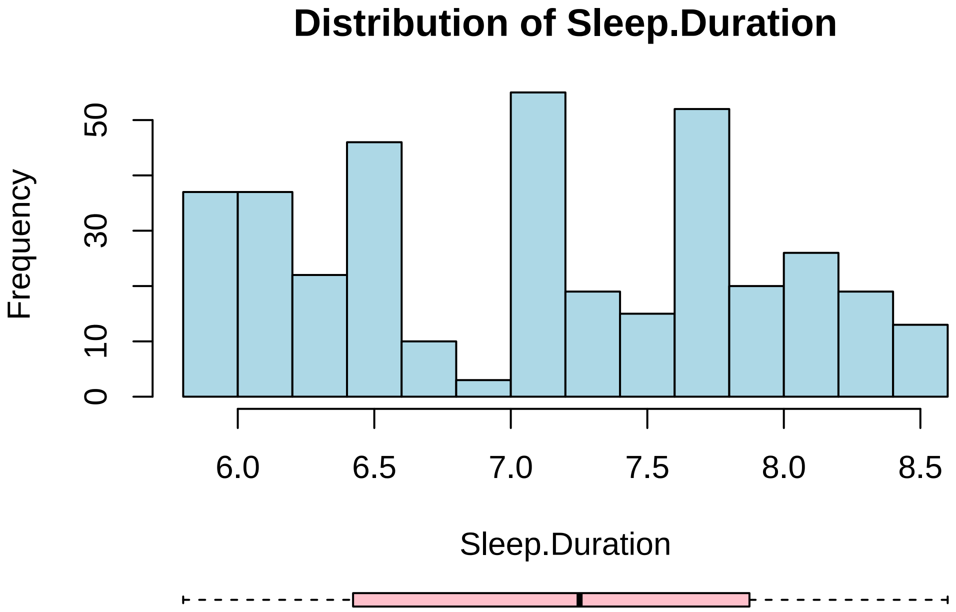
ii.continuous variable

```
graphics::layout(mat = matrix(c(1,2),2, byrow = FALSE),  height = c(8,1))
par(mar=c(4, 4, 3, 2))
hist(data$Age, main = 'Distribution of Age',
     xlab="Age",col="lightblue")
par(mar=c(0.5, 4, 0.5, 2))
boxplot(data$Age, xaxt = "n", horizontal=TRUE,
        col="pink", border="black", frame = FALSE)
```

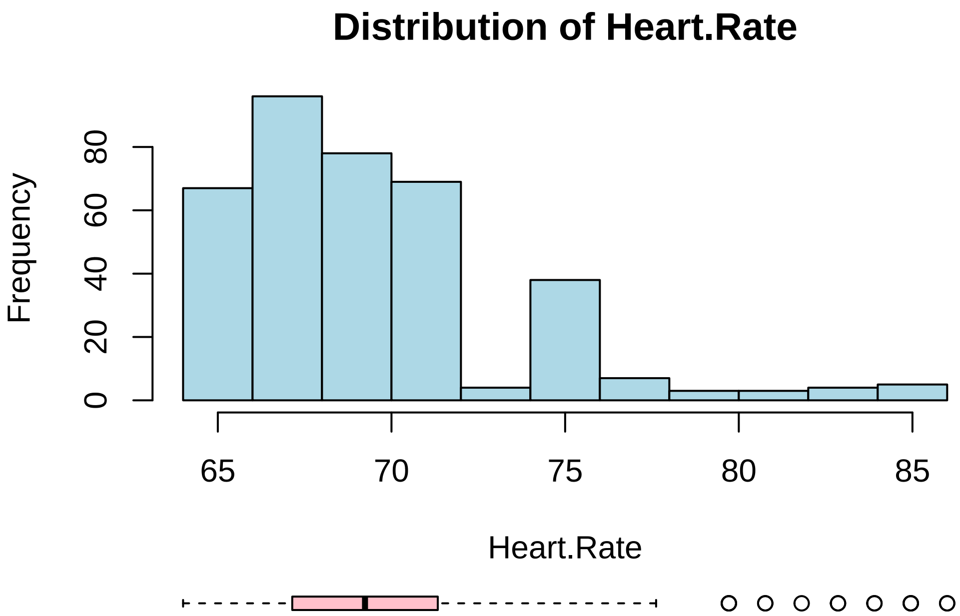



```
par(mar=c(4, 4, 3, 2))
hist(data$Sleep.Duration, main = 'Distribution of Sleep.Duration',
     xlab="Sleep.Duration",col="lightblue")
par(mar=c(0.5, 4, 0.5, 2))
boxplot(data$Sleep.Duration, xaxt = "n", horizontal=TRUE,
```

```r
        col="pink", border="black", frame = FALSE)
```

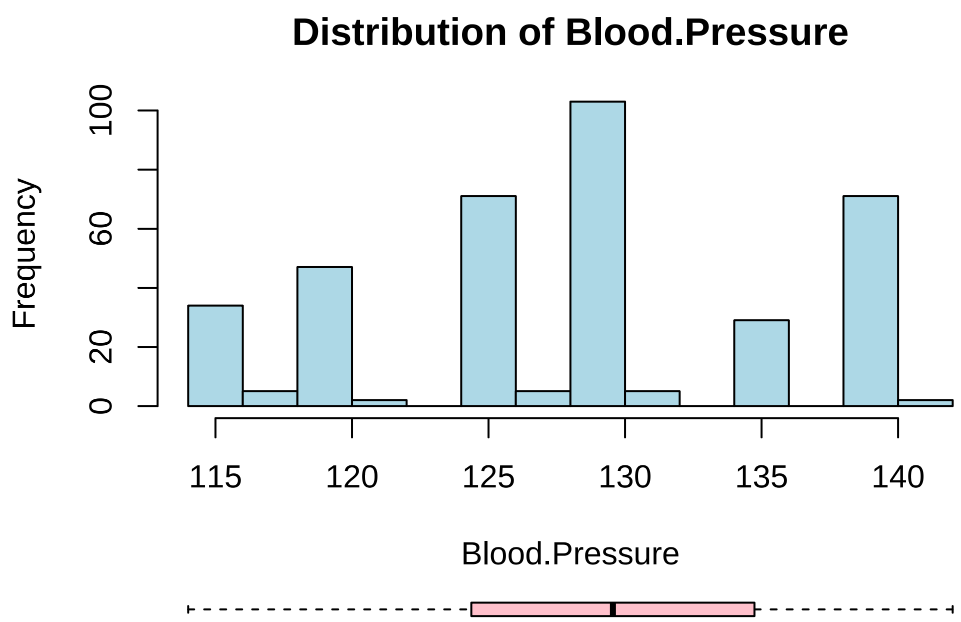



```r
par(mar=c(4, 4, 3, 2))
hist(data$Heart.Rate, main = 'Distribution of Heart.Rate',
     xlab="Heart.Rate",col="lightblue")
par(mar=c(0.5, 4, 0.5, 2))
boxplot(data$Heart.Rate, xaxt = "n", horizontal=TRUE,
        col="pink", border="black", frame = FALSE)
```



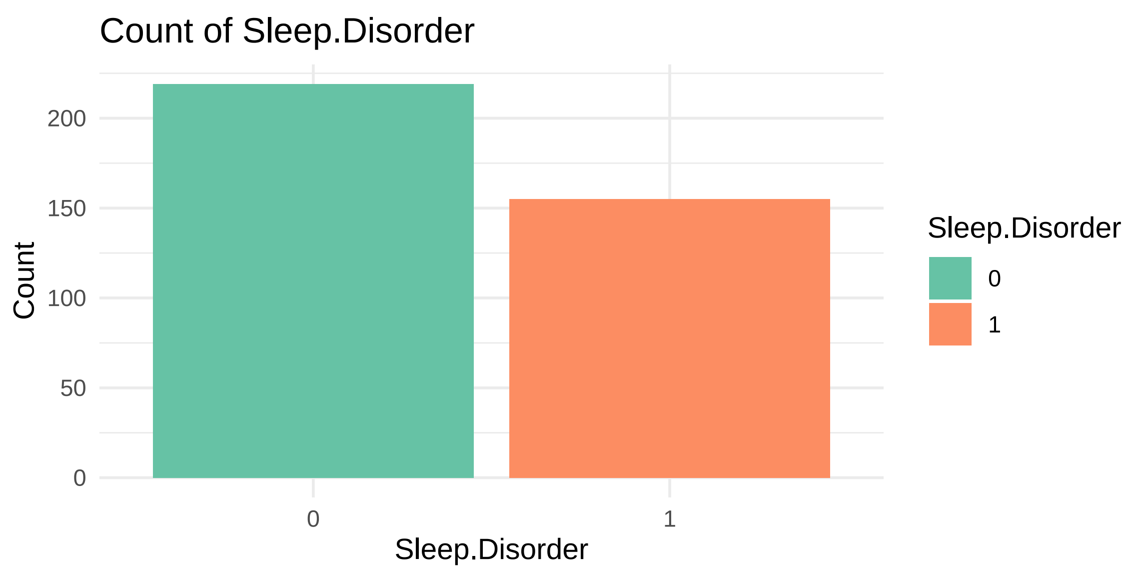

```r
par(mar=c(4, 4, 3, 2))
hist(data$Blood.Pressure, main = 'Distribution of Blood.Pressure',
     xlab="Blood.Pressure",col="lightblue")
```

```r
par(mar=c(0.5, 4, 0.5, 2))
boxplot(data$Blood.Pressure, xaxt = "n", horizontal=TRUE,
        col="pink", border="black", frame = FALSE)
```



iii.Sleep Disorder

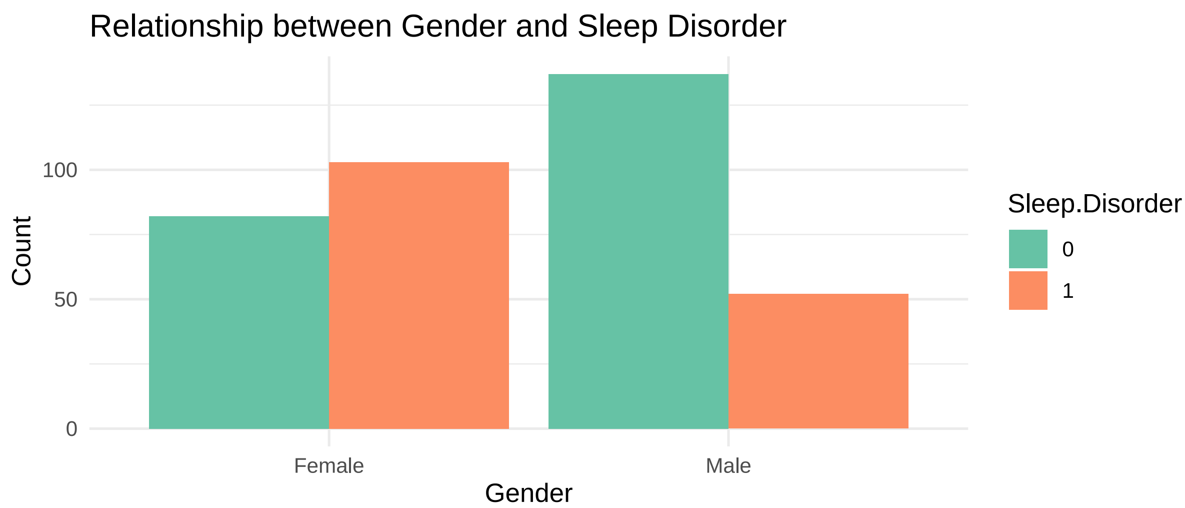
```r
ggplot(data, aes(x = Sleep.Disorder, fill = Sleep.Disorder)) +
  geom_bar() +
  labs(title = "Count of Sleep.Disorder", x = "Sleep.Disorder", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



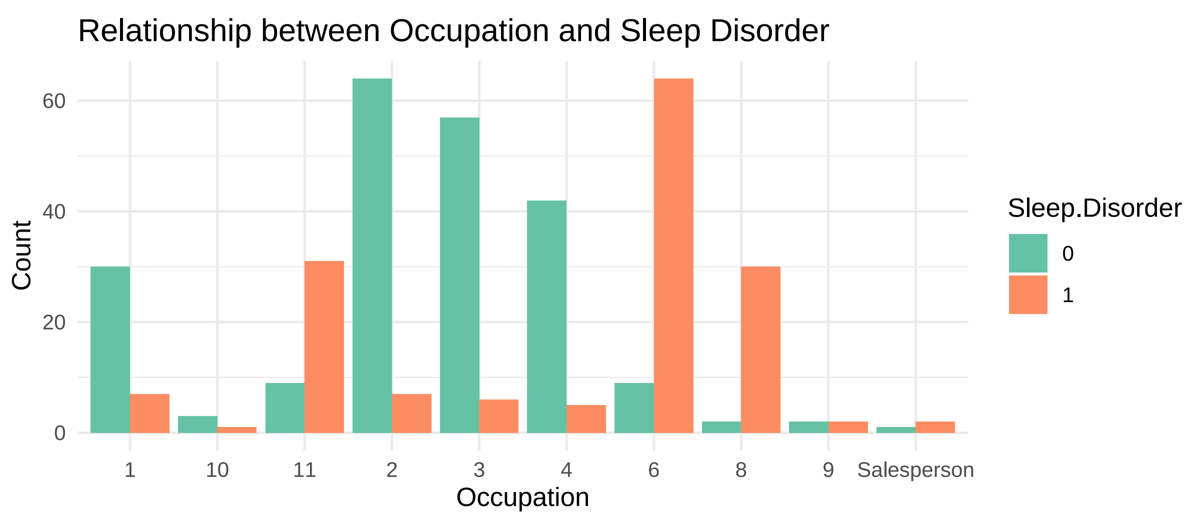
Correlation between data(variables & sleep disorder)

i.categorical variable

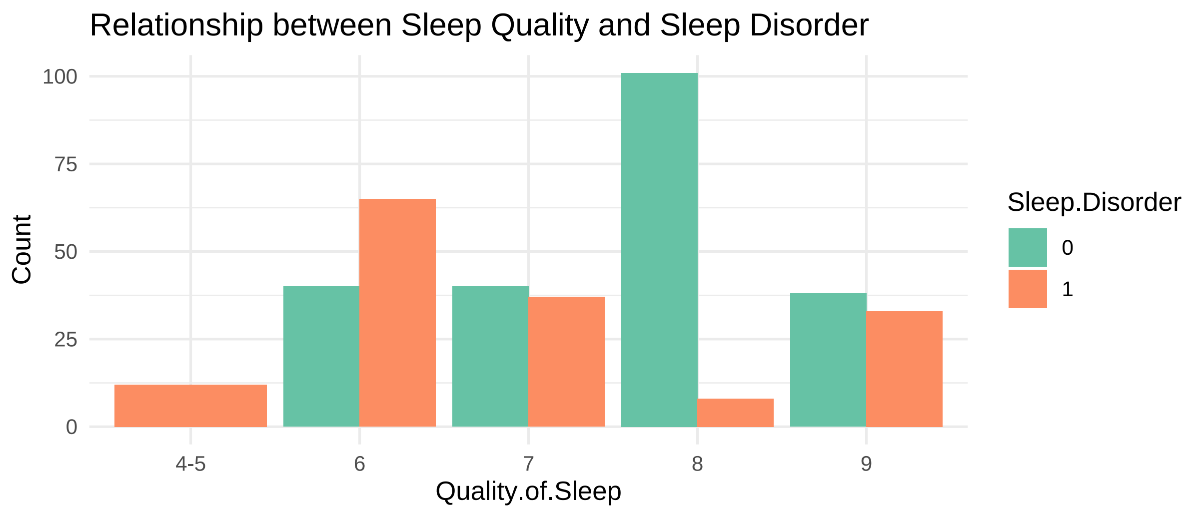
```
ggplot(data, aes(x = Gender, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between Gender and Sleep Disorder",
       x = "Gender",
       y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```



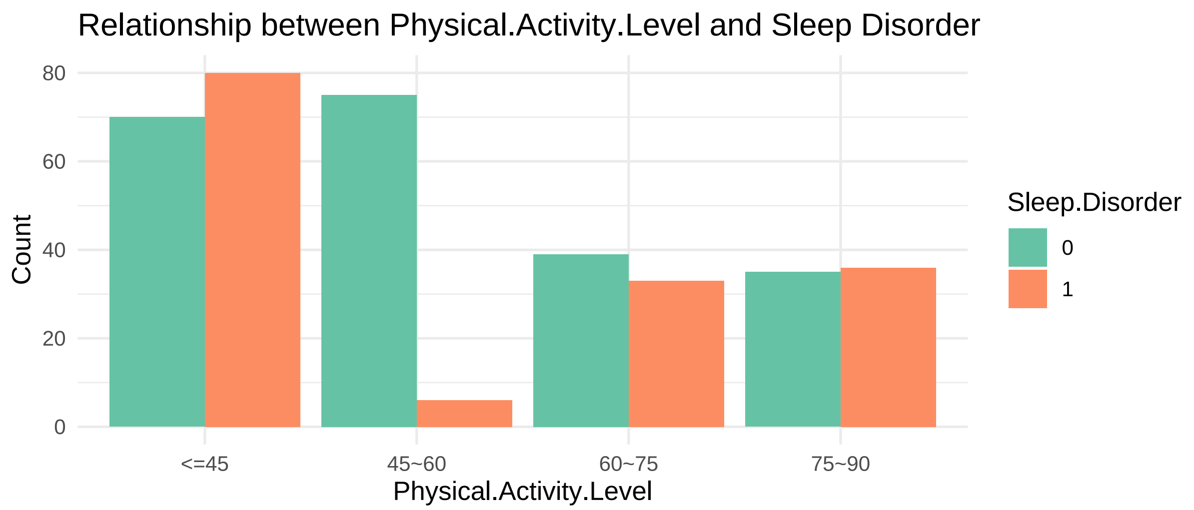

```
ggplot(data, aes(x = Occupation, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between Occupation and Sleep Disorder",
       x = "Occupation",
       y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```



```r
ggplot(data, aes(x = Quality.of.Sleep, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between Sleep Quality and Sleep Disorder",
       x = "Quality.of.Sleep",
       y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```

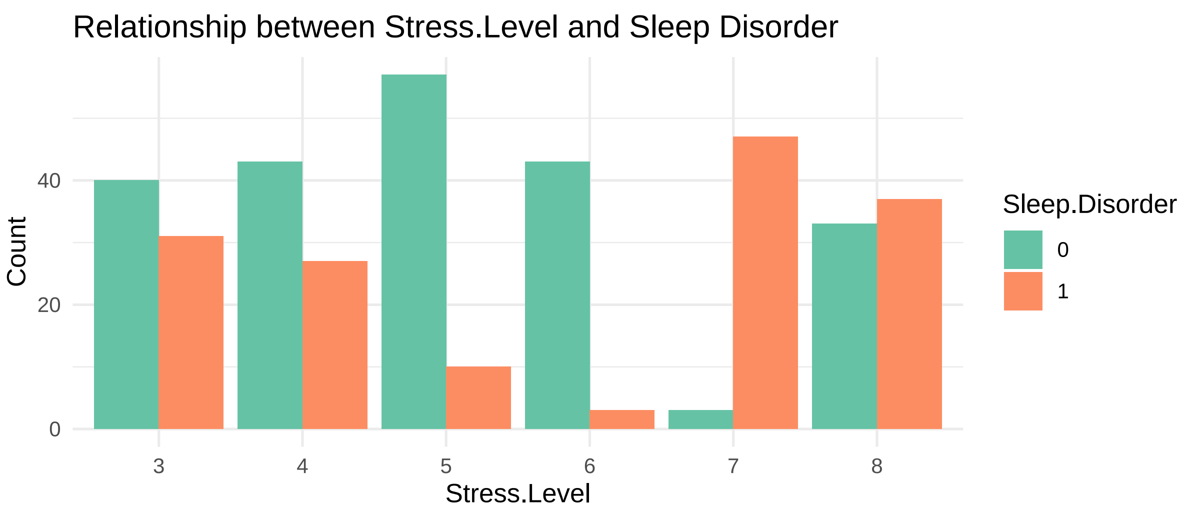



```r
ggplot(data, aes(x = Physical.Activity.Level, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between Physical.Activity.Level and Sleep Disorder",
       x = "Physical.Activity.Level",
       y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```

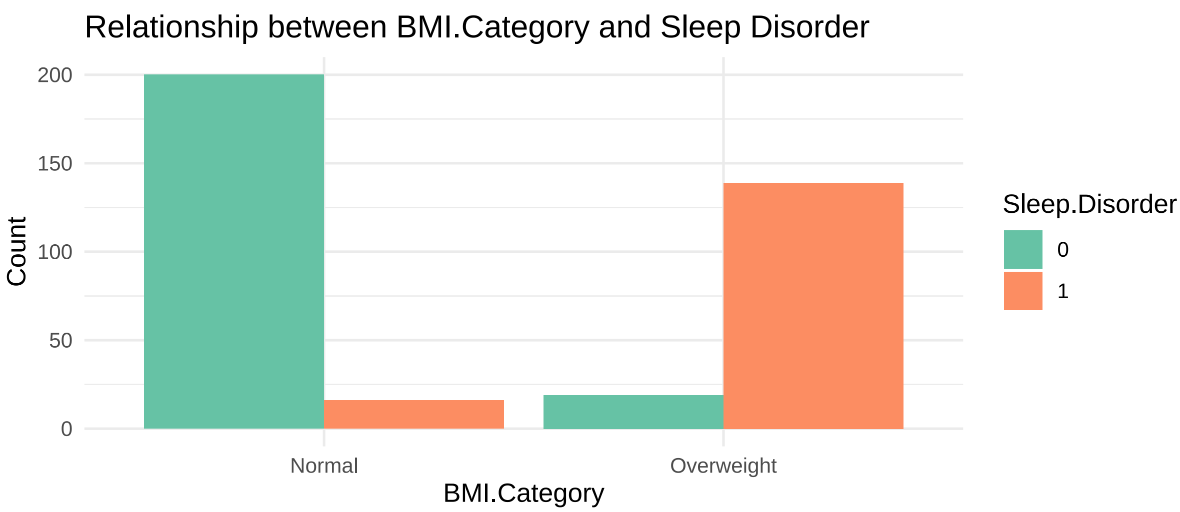



```r
ggplot(data, aes(x = Stress.Level, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between Stress.Level and Sleep Disorder",
       x = "Stress.Level",
```

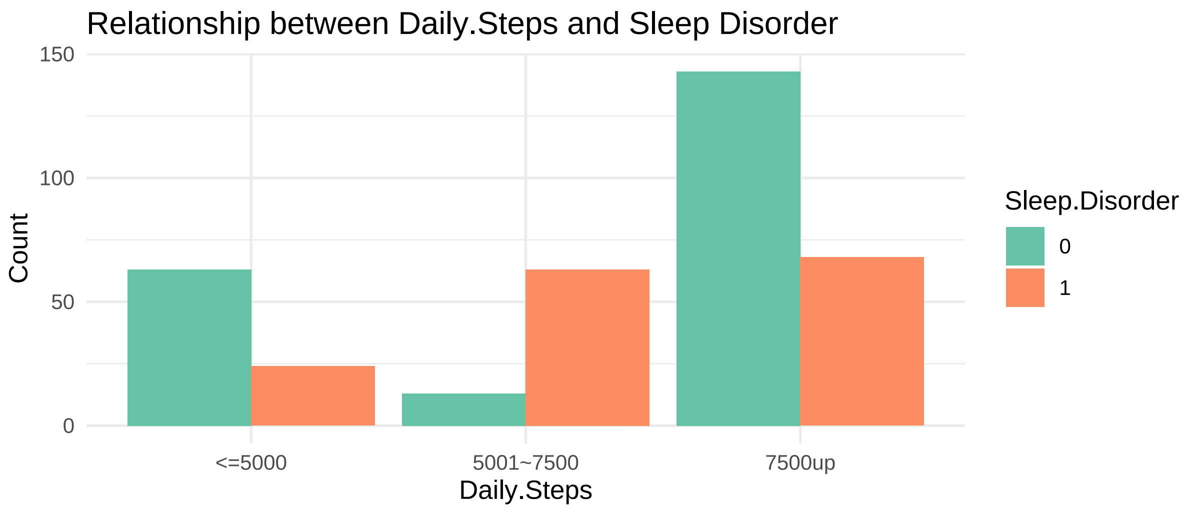
```
      y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```




```
ggplot(data, aes(x = BMI.Category, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between BMI.Category and Sleep Disorder",
      x = "BMI.Category",
      y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```




```
ggplot(data, aes(x = Daily.Steps, fill = Sleep.Disorder)) +
  geom_bar(position = "dodge") +
  labs(title = "Relationship between Daily.Steps and Sleep Disorder",
      x = "Daily.Steps",
      y = "Count") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```



1. 性別: 調查資料中，女生中有睡眠疾病的比例較高；男性中無睡眠疾病的比例較高

2. 職業: 無睡眠疾病比例較高的有會計師、醫師、工程師以及律師；有睡眠疾病比例較高的有護士、商人以及老師

3. 睡眠品質: 可大致上看出睡眠品質越高，有睡眠疾病的比例越低

4. 身體活動量: 無法觀察出明顯趨勢

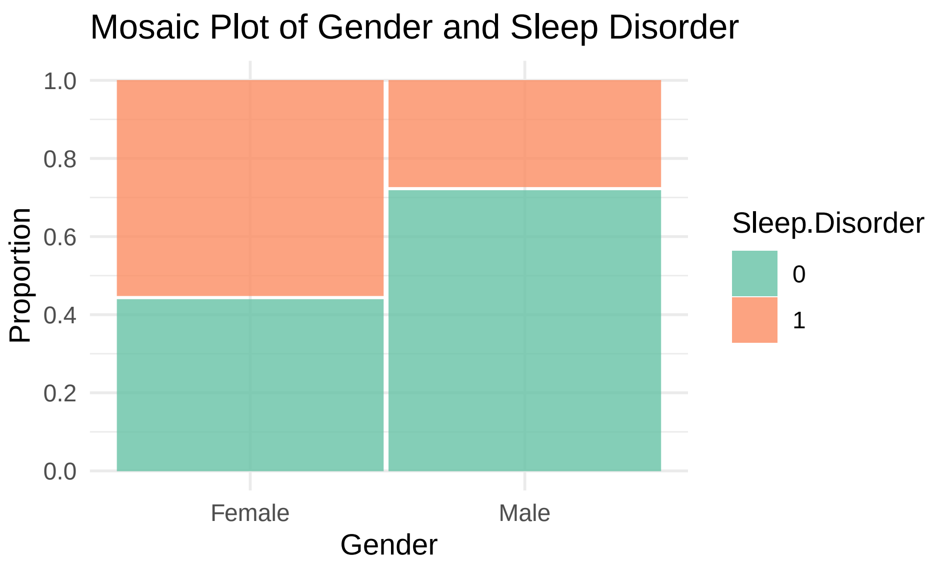
5. 壓力指數: 可大致上看出壓力指數高，有睡眠疾病的比例也高但睡眠疾病比例最低的是壓力指數適中的人

6.BMI 指數:BMI 正常的人大多無睡眠疾病，而過重的人大多有睡眠疾病

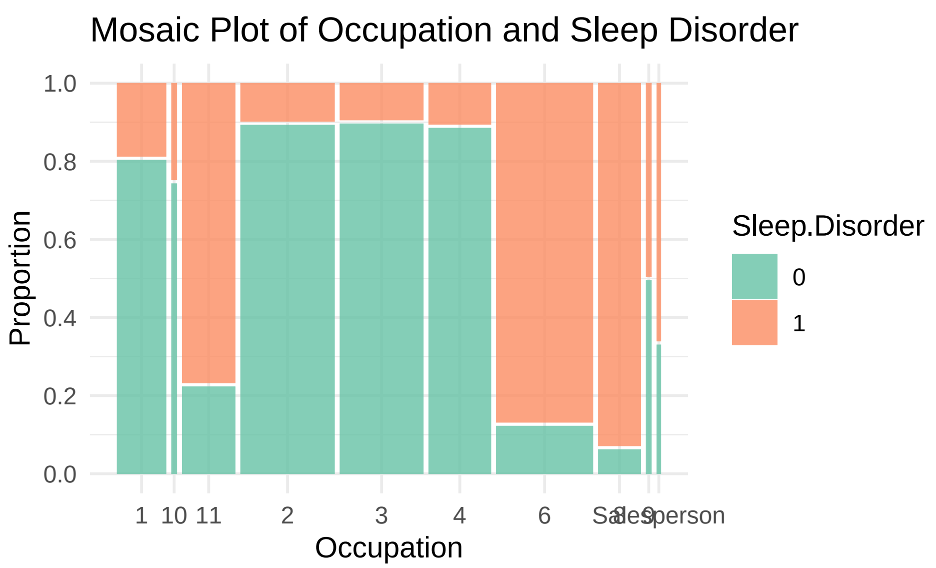
7. 每日步數: 每日走大於 7500 步的人擁有睡眠疾病的比例遠低於無睡眠疾病

馬賽克圖-可以清楚看出比例

```
library(ggmosaic)
# 繪製馬賽克圖
# Gender 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(Gender), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of Gender and Sleep Disorder",
       x = "Gender",
       y = "Proportion") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                          breaks = seq(0, 1, 0.2))
```

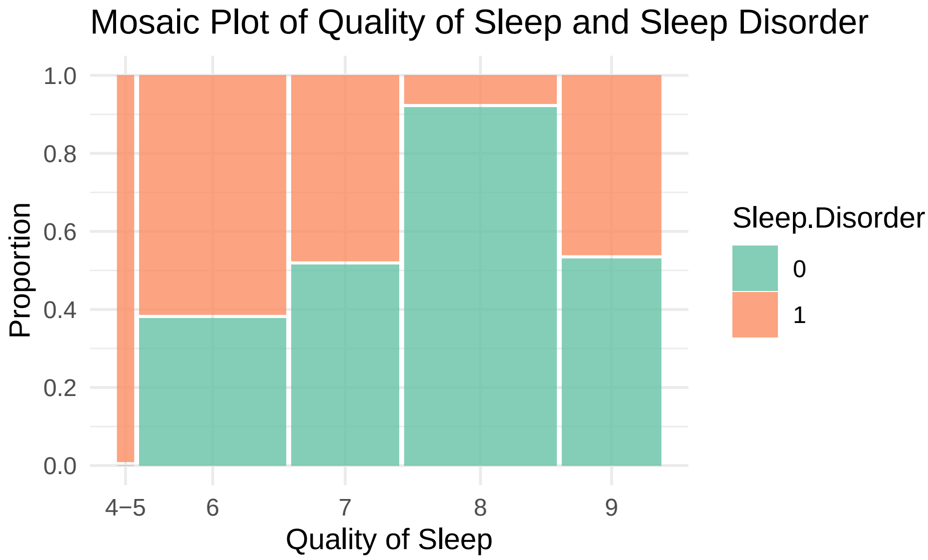



```
# Occupation 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(Occupation), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of Occupation and Sleep Disorder",
       x = "Occupation",
       y = "Proportion") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                          breaks = seq(0, 1, 0.2))
```

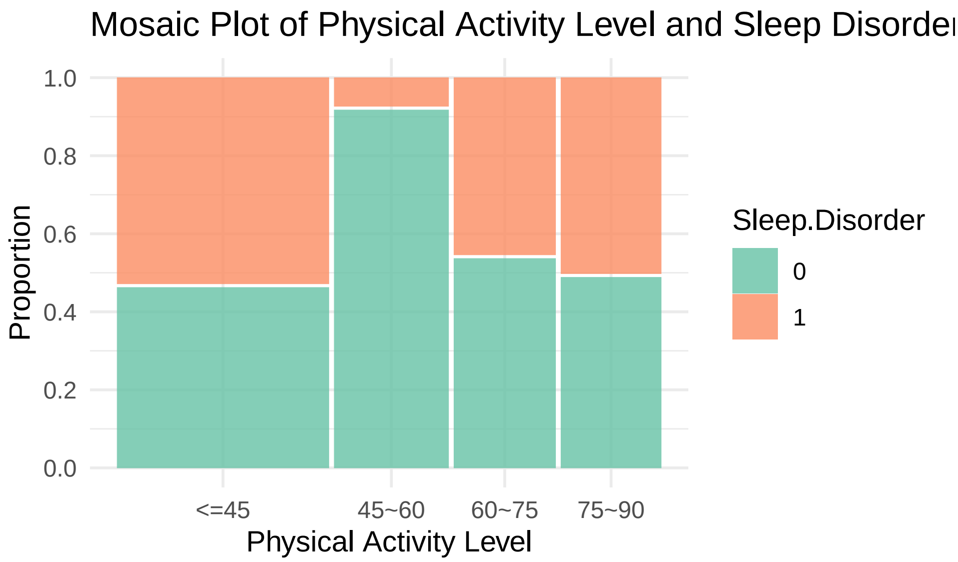



```
# Quality.of.Sleep 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(Quality.of.Sleep), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of Quality of Sleep and Sleep Disorder",
       x = "Quality of Sleep",
       y = "Proportion") +
  theme_minimal() +
```

```r
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                          breaks = seq(0, 1, 0.2))
```



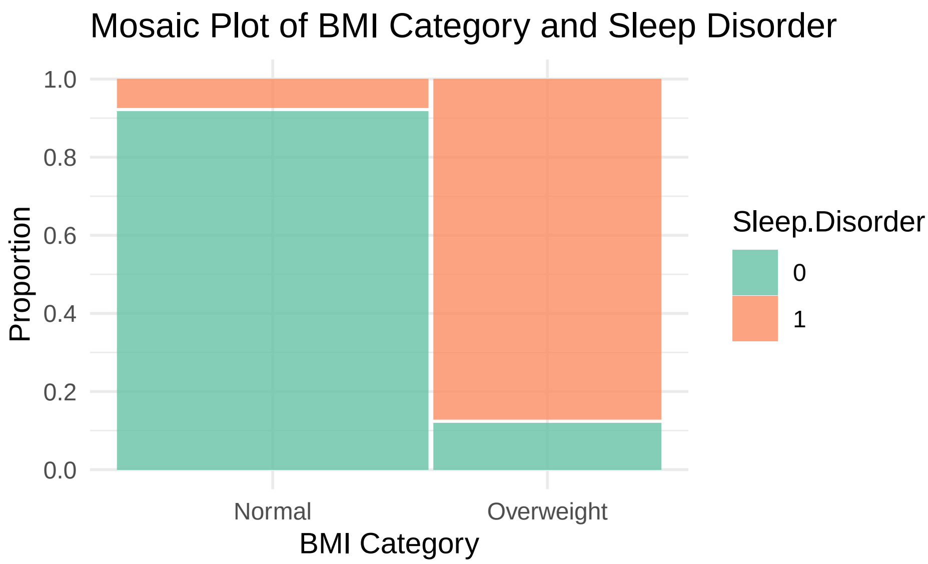

```r
# Physical.Activity.Level 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(Physical.Activity.Level), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of Physical Activity Level and Sleep Disorder",
       x = "Physical Activity Level",
       y = "Proportion") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                          breaks = seq(0, 1, 0.2))
```



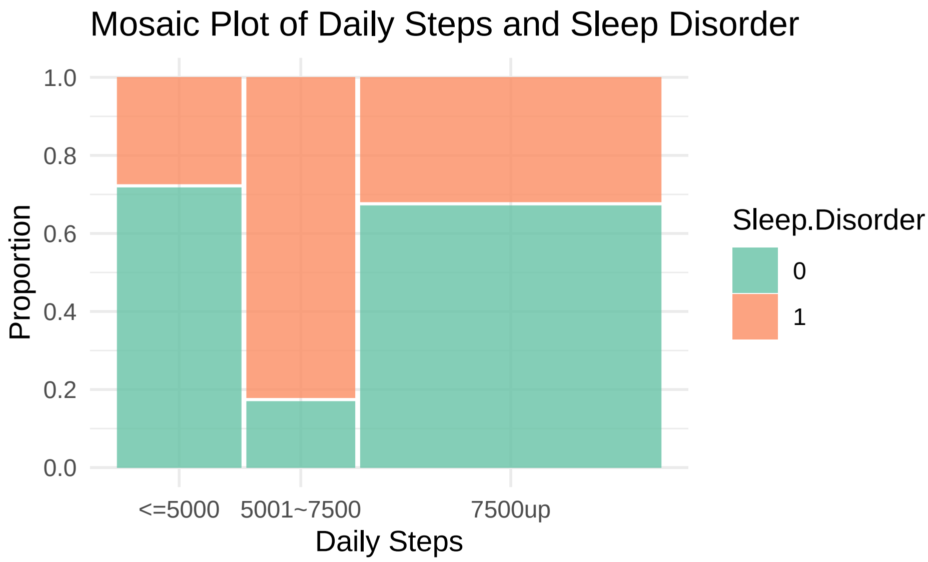

```r
# Stress.Level 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(Stress.Level), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of Stress Level and Sleep Disorder",
```

```
      x = "Stress Level",
      y = "Proportion") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                            breaks = seq(0, 1, 0.2))
```


```
# BMI.Category 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(BMI.Category), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of BMI Category and Sleep Disorder",
      x = "BMI Category",
      y = "Proportion") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                            breaks = seq(0, 1, 0.2))
```

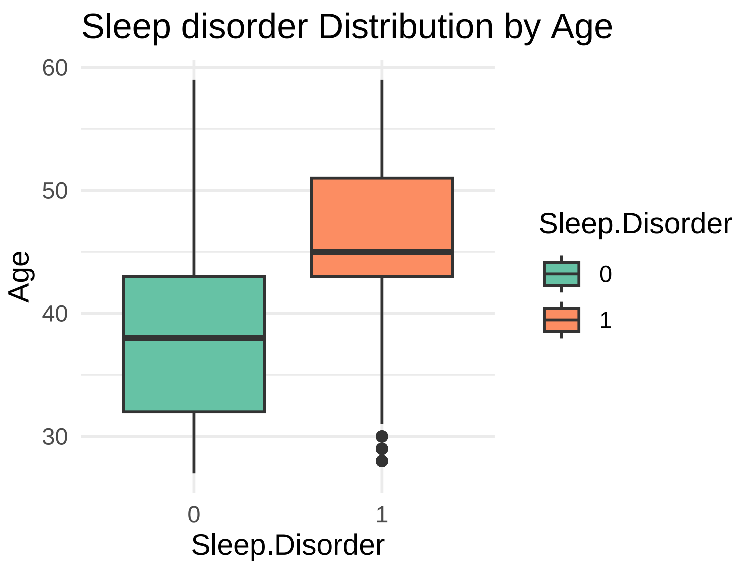


```
# Daily.Steps 和 Sleep.Disorder
ggplot(data) +
  geom_mosaic(aes(x = product(Daily.Steps), fill = Sleep.Disorder)) +
  labs(title = "Mosaic Plot of Daily Steps and Sleep Disorder",
       x = "Daily Steps",
       y = "Proportion") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")+ scale_y_continuous(limits = c(0, 1),
                                                         breaks = seq(0, 1, 0.2))
```

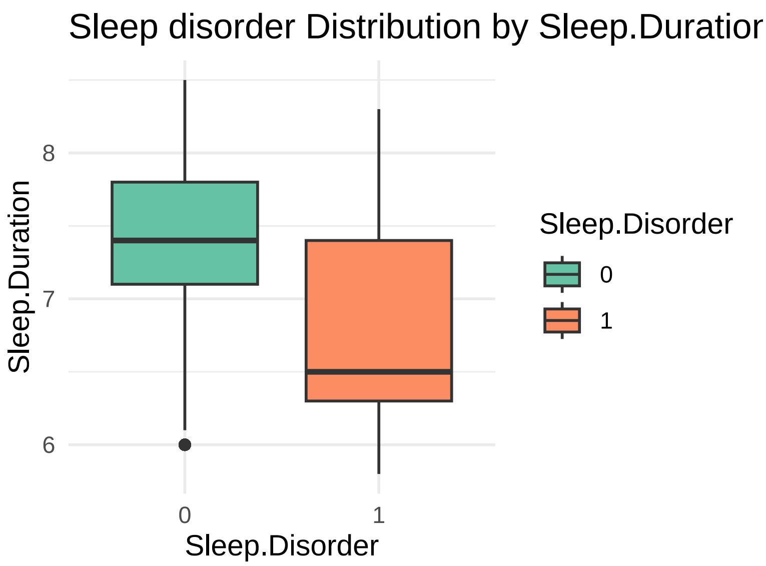


ii.continuous variable

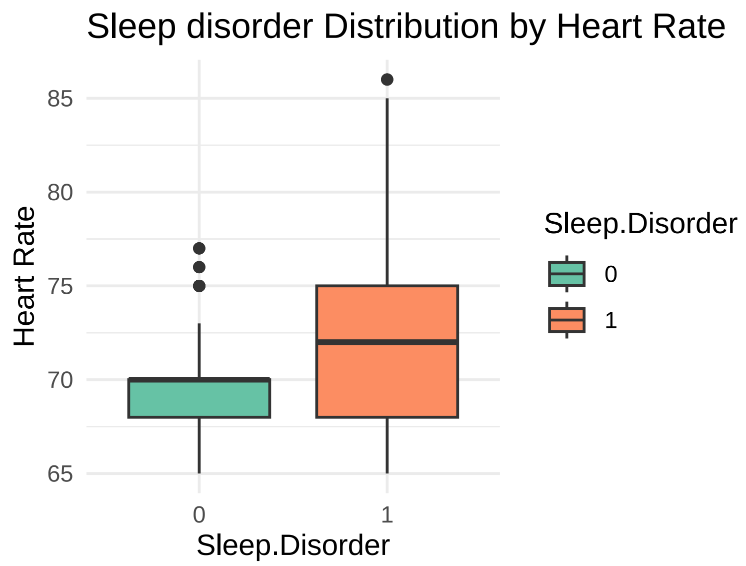
```
ggplot(data, aes(x = Sleep.Disorder, y = Age, fill = Sleep.Disorder)) +
  geom_boxplot() +
  labs(title = "Sleep disorder Distribution by Age",
       x = "Sleep.Disorder", y = "Age") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



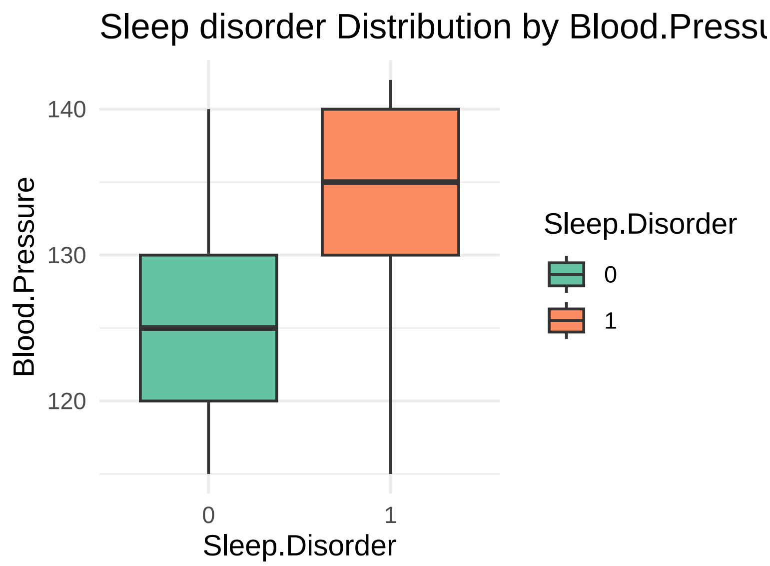

```r
ggplot(data, aes(x = Sleep.Disorder, y = Sleep.Duration, fill = Sleep.Disorder)) +
  geom_boxplot() +
  labs(title = "Sleep disorder Distribution by Sleep.Duration",
       x = "Sleep.Disorder", y = "Sleep.Duration") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```




```r
ggplot(data, aes(x = Sleep.Disorder, y = Heart.Rate, fill = Sleep.Disorder)) +
  geom_boxplot() +
  labs(title = "Sleep disorder Distribution by Heart Rate",
       x = "Sleep.Disorder", y = "Heart Rate") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



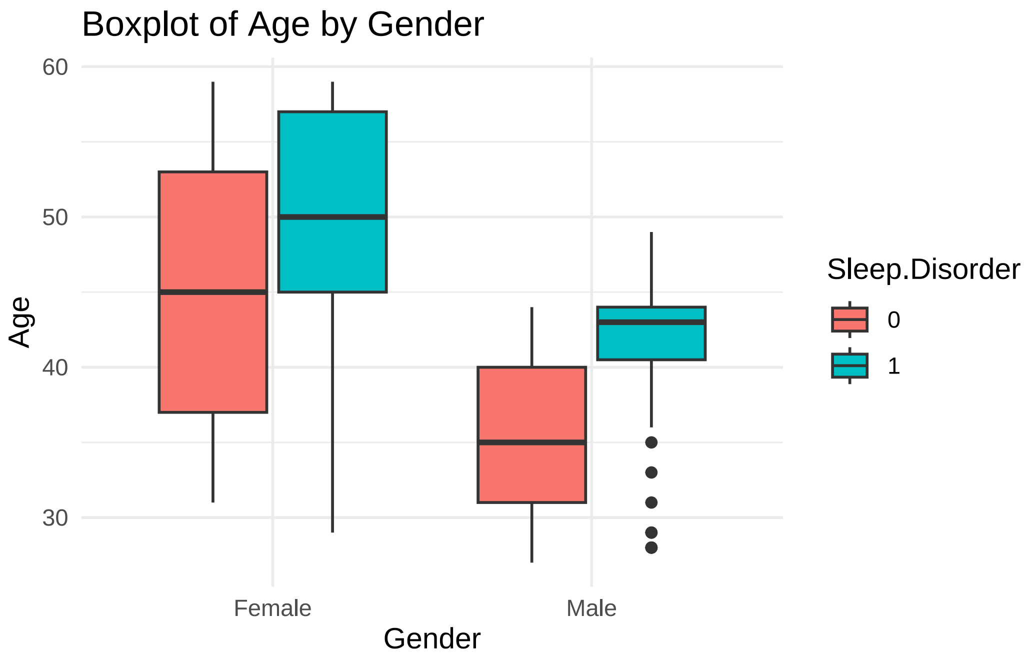

```r
ggplot(data, aes(x = Sleep.Disorder, y = Blood.Pressure, fill = Sleep.Disorder)) +
  geom_boxplot() +
  labs(title = "Sleep disorder Distribution by Blood.Pressure",
       x = "Sleep.Disorder", y = "Blood.Pressure") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



1. 年齡: 有睡眠疾病的平均年齡高於無睡眠疾病
2. 睡眠時長: 有睡眠疾病的睡眠時長低於無睡眠疾病
3. 心率: 有睡眠疾病的人心率平均高於無睡眠疾病
4. 血壓: 有睡眠疾病的人血壓平均高於無睡眠疾病

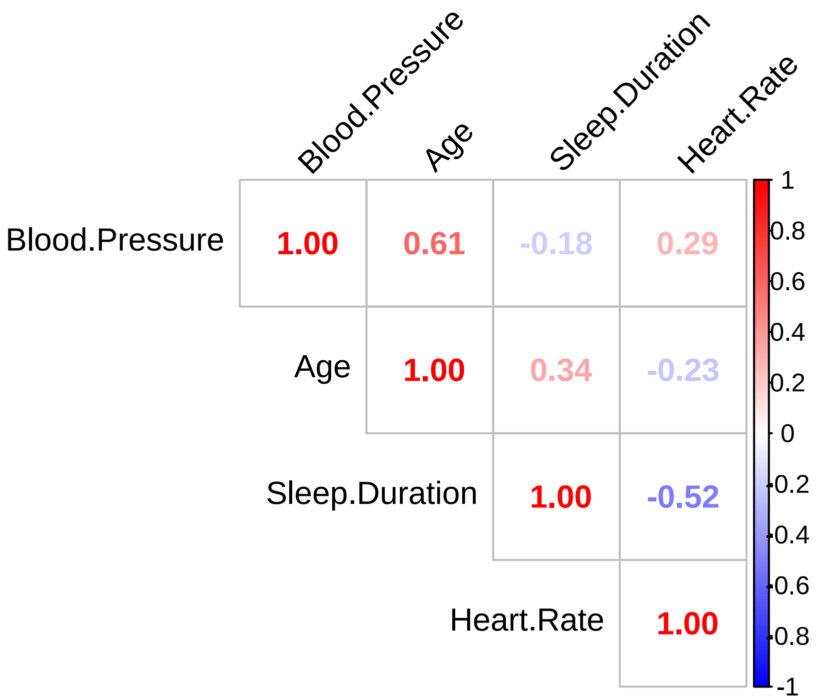
兩變數對 sleep disorder 關係圖

```r
ggplot(data, aes(x = Gender, y = Age, fill = Sleep.Disorder)) +
  geom_boxplot() +
  labs(title = "Boxplot of Age by Gender", x = "Gender", y = "Age") +
  theme_minimal()
```



連續型自變數之間的關係

```r
#heatmap
numeric_vars <- data %>% dplyr::select(Blood.Pressure,Age, Sleep.Duration,Heart.Rate)
cor_matrix <- cor(numeric_vars)
corrplot(cor_matrix, method = "number", type = "upper",
         tl.col = "black", tl.srt = 45,
         col = colorRampPalette(c("blue", "white", "red"))(200))
```



可以發現，變數間呈現負相關的組合: Blood.Pressure &sleep duration、Heart.Rate & sleep duration 其中 Heart.Rate & sleep duration 相關係數達到-0.5

變數間呈現正相關的組合: Blood.Pressure & Age 相關係數達到 0.6，相關性很高

類別型自變數之間的關係（計算 Cramér's V ）

選擇此統計指標的原因在於:

使用卡方檢定，其缺點在於無法衡量關聯性的強度。

而 Cramer's V 優點在於: 可以衡量關聯性的強度，並提供更直觀的解釋

然而由於此資料為小樣本，某些組合在列聯表中會出現樣本數為 0 的格子，直接使用 Cramer's V 計算可能會導致結果不準確，甚至無法計算。因此使用 Bootstrap 方法來計算 Cramer's V 做修正。

(Bootstrap 的方法，透過重複從原始資料中抽取樣本，建立多個模擬資料集，並計算每個資料集的 Cramer's V 係數。

最後，可以透過計算這些 Cramer's V 係數的平均值和標準誤差，得到更穩健的 Cramer's V 估計值及其信賴區間。)

類別自變數間皆顯著而高度相關可能的組合有 (Cramér's V 大於 0.5):

1.Gender 跟 Occupation、stress level 有關 (由高到低排序)

2.BMI.Category 跟 Occupation、Daily.Steps、Quality of Sleep、Stress.Level

3.Physical.Activity.Level 又跟 Daily.Steps、Stress.level、Occupation 有關 (由高到低排序)

4.Quality.of.Sleep 跟 Stress.level、BMI.Category、Occupation、Physical.Activity.Level、Daily.Steps 有關 (由高到低排序)

5. 其中，值得注意的是:

Occupation 幾乎與所有類別變數的組合皆高度相關

(與 Gender、Quality.of.Sleep、Physical.Activity.Level、Stress.Level、BMI.Category

、Daily.Steps、Sleep.Disorder 等變數組合)

-> 可能反映了職業對生活習慣、健康指標和心理壓力的潛在影響。

另外，直接與 Sleep.Disorder(目標變數) 具高度相關的變數有以下幾組，可能對於預測結果會有幫助，由 Cramer's V 高到低依序排序: BMI.Category、Occupation、Stress.Level、Quality.of.Sleep

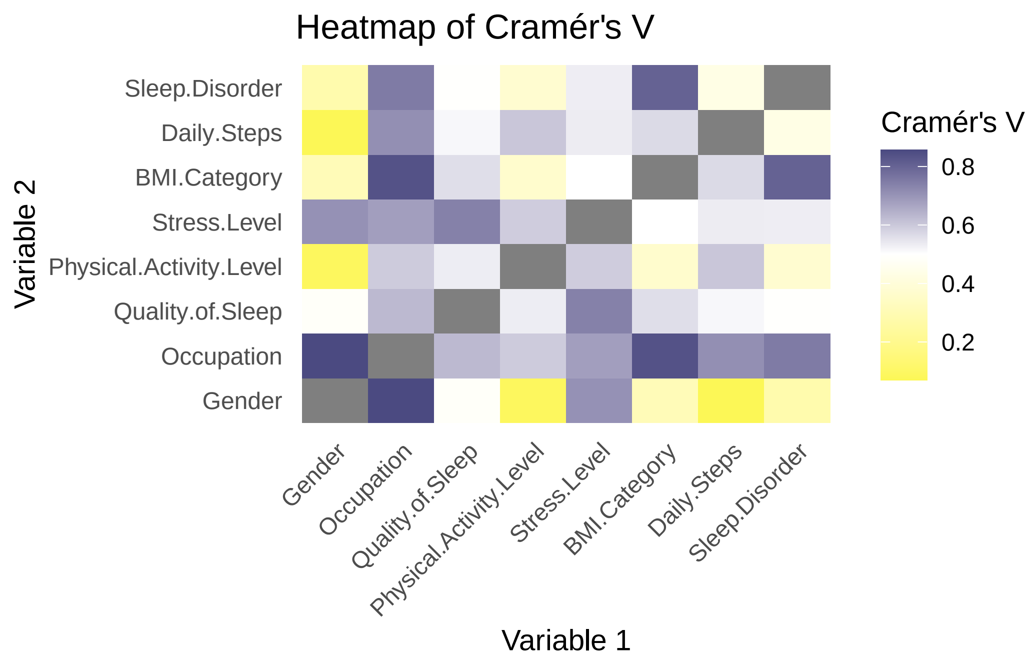
Heatmap of Cramer's V

```r
library(ggplot2)
library(reshape2)
library(knitr)

# 定義計算 Cramér's V 的函數
cramers_v <- function(table) {
  chi_sq <- chisq.test(table)
  n <- sum(table)  # 總樣本數
  min_dim <- min(nrow(table), ncol(table)) - 1  # 最小維度
  v <- sqrt(chi_sq$statistic / (n * min_dim))
  return(v)
}

# 進行 bootstrap 重抽樣計算 Cramér's V
bootstrap_cramers_v <- function(data, var1, var2, n_bootstrap = 1000) {
  v_values <- numeric(n_bootstrap)

  for (i in 1:n_bootstrap) {
    # 進行 bootstrap 重抽樣
    bootstrap_sample <- data[sample(nrow(data), replace = TRUE), ]
    tbl <- table(bootstrap_sample[[var1]], bootstrap_sample[[var2]])

    # 檢查列聯表是否包含 NA 或空格
    if (all(dim(tbl) > 1)) {
      v_values[i] <- cramers_v(tbl)
    } else {
      v_values[i] <- NA  # 如果列聯表的某個維度為 1，設為 NA
    }
  }

  # 計算均值和 95% 置信區間，忽略 NA
  mean_v <- mean(v_values, na.rm = TRUE)
  ci_lower <- quantile(v_values, 0.025, na.rm = TRUE)
```

```r
  ci_upper <- quantile(v_values, 0.975, na.rm = TRUE)
  return(list(mean = mean_v, ci_lower = ci_lower, ci_upper = ci_upper))
}

# 取得所有變數名稱
all_vars <- names(data)

# 確定類別變數
categorical_vars <- all_vars[sapply(data, is.factor)]

# 計算每對變數的 Cramér's V 並存儲結果
cramers_v_matrix <- matrix(NA, nrow = length(categorical_vars),
                           ncol = length(categorical_vars))
rownames(cramers_v_matrix) <- categorical_vars
colnames(cramers_v_matrix) <- categorical_vars

results <- data.frame(
  Variable1 = character(),
  Variable2 = character(),
  Cramers_V_Mean = numeric(),
  Cramers_V_Lower_CI = numeric(),
  Cramers_V_Upper_CI = numeric(),
  stringsAsFactors = FALSE
)

for (i in 1:(length(categorical_vars) - 1)) {
  for (j in (i + 1):length(categorical_vars)) {
    var1 <- categorical_vars[i]
    var2 <- categorical_vars[j]

    # 計算 bootstrap Cramér's V 和信賴區間
    cramers_v_result <- bootstrap_cramers_v(data, var1, var2)

    # 存儲結果
    results <- rbind(results, data.frame(
      Variable1 = var1,
      Variable2 = var2,
      Cramers_V_Mean = cramers_v_result$mean,
      Cramers_V_Lower_CI = cramers_v_result$ci_lower,
      Cramers_V_Upper_CI = cramers_v_result$ci_upper
    ))

    # 更新 Cramér's V 矩陣
    cramers_v_matrix[var1, var2] <- cramers_v_result$mean
    cramers_v_matrix[var2, var1] <- cramers_v_result$mean # Cramér's V 是對稱的
  }
}
```

```r
# 轉換為長格式數據框以便於 ggplot
cramers_v_df <- melt(cramers_v_matrix)
```

```r
# 繪製 heatmap 使用顏色漸變（低值：淺粉紅，中間值：白色，高值：深藍）
heatmap_plot <- ggplot(cramers_v_df, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low="#FCF756", high = "#222A68",
                       midpoint = 0.5, name = "Cramér's V") +  # 添加中間顏色
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1,size = 9),
        axis.text.y = element_text(angle = 0, hjust = 1,size = 9),
        plot.title = element_text(size = 13),
        panel.grid = element_blank()) +
  labs(title = "Heatmap of Cramér's V ", x = "Variable 1", y = "Variable 2")

# 顯示熱力圖
print(heatmap_plot)
```




```r
# 5. 使用 knitr 輸出結果
kable(results, caption = "Categorical Variables Correlation
      Results with Bootstrap Cramér's V and Confidence Intervals")
```

Table 2: Categorical Variables Correlation Results with Bootstrap Cramér's V and Confidence Intervals

	Variable1	Variable2	Cramers_V_Mean	Cramers_V_Lower_CI	Cramers_V_Upper_CI
2.5%	Gender	Occupation	0.8555444	0.8140123	0.8910011
2.5%1	Gender	Quality.of.Sleep	0.4845276	0.4197725	0.5472702
2.5%2	Gender	Physical.Activity.Level	0.0865044	0.0262399	0.1600976
2.5%3	Gender	Stress.Level	0.7059606	0.6464357	0.7610154

	Variable1	Variable2	Cramers_V_Mean	Cramers_V_Lower_CI	Cramers_V_Upper_CI
2.5%4	Gender	BMI.Category	0.3078853	0.2153752	0.4010777
2.5%5	Gender	Daily.Steps	0.0699555	0.0132612	0.1513761
2.5%6	Gender	Sleep.Disorder	0.2807166	0.1843609	0.3719341
2.5%7	Occupation	Quality.of.Sleep	0.6286894	0.5772549	0.6839716
2.5%8	Occupation	Physical.Activity.Level	0.5950051	0.5563616	0.6339064
2.5%9	Occupation	Stress.Level	0.6802944	0.6414050	0.7160202
2.5%10	Occupation	BMI.Category	0.8378161	0.7857550	0.8896379
2.5%11	Occupation	Daily.Steps	0.7100905	0.6600186	0.7582372
2.5%12	Occupation	Sleep.Disorder	0.7498998	0.6842837	0.8135619
2.5%13	Quality.of.Sleep	Physical.Activity.Level	0.5332824	0.4935642	0.5696461
2.5%14	Quality.of.Sleep	Stress.Level	0.7387936	0.7108007	0.7672962
2.5%15	Quality.of.Sleep	BMI.Category	0.5606279	0.5031105	0.6174705
2.5%16	Quality.of.Sleep	Daily.Steps	0.5142479	0.4585751	0.5665889
2.5%17	Quality.of.Sleep	Sleep.Disorder	0.4953816	0.4233791	0.5651868
2.5%18	Physical.Activity.Level	Stress.Level	0.5924724	0.5528431	0.6311184
2.5%19	Physical.Activity.Level	BMI.Category	0.3658878	0.2894847	0.4417051
2.5%20	Physical.Activity.Level	Daily.Steps	0.6042249	0.5703590	0.6363956
2.5%21	Physical.Activity.Level	Sleep.Disorder	0.3734040	0.3025775	0.4476397
2.5%22	Stress.Level	BMI.Category	0.4992792	0.4250994	0.5701039
2.5%23	Stress.Level	Daily.Steps	0.5345695	0.4798841	0.5888210
2.5%24	Stress.Level	Sleep.Disorder	0.5329867	0.4627971	0.5986801
2.5%25	BMI.Category	Daily.Steps	0.5677209	0.4952619	0.6320995
2.5%26	BMI.Category	Sleep.Disorder	0.8023936	0.7364084	0.8606203
2.5%27	Daily.Steps	Sleep.Disorder	0.4301340	0.3449287	0.5137367

連續 v.s. 類別變數

類別 vs. 連續:

使用 Kruskal-Wallis 檢定，皆為顯著 (p-value<0.05)

其中，值得注意的是，可以發現有幾個變數組合之 p-value 值極小，分別為:

1.Sleep.Duration/Quality.of.Sleep 2.Sleep.Duration/Stress.Level

3.Quality.of.Sleep/Heart.Rate 4.Stress.Level/Heart.Rate

```r
# 獲取所有變數名稱
all_vars <- names(data)

# 確定類別與連續變數
categorical_vars <- all_vars[sapply(data, is.factor)]
continuous_vars <- all_vars[sapply(data, is.numeric)]

# 初始化結果數據框
results <- data.frame(
  Variable1 = character(),
  Variable2 = character(),
  Correlation_Type = character(),
```

```
  #P_Value = numeric(),
  P_Value = character(), # 添加科學記號顯示的欄位
  stringsAsFactors = FALSE
)

# 計算相關性
for (i in 1:(length(all_vars) - 1)) {
  for (j in (i + 1):length(all_vars)) {
    var1 <- all_vars[i]
    var2 <- all_vars[j]

    # 連續對類別 (Kruskal-Wallis 檢定)
    if ((var1 %in% categorical_vars && var2 %in% continuous_vars) ||
        (var1 %in% continuous_vars && var2 %in% categorical_vars)) {
      cat_var <- ifelse(var1 %in% categorical_vars, var1, var2)
      cont_var <- ifelse(var1 %in% continuous_vars, var1, var2)
      kw_test <- kruskal.test(data[[cont_var]] ~ data[[cat_var]])
      p_value_sci <- formatC(kw_test$p.value, format = "e", digits = 2)# 換為科學記號格式
      results <- rbind(results, data.frame(
        Variable1 = var1,
        Variable2 = var2,
        Correlation_Type = "Kruskal-Wallis",
       # P_Value = kw_test$p.value,
        P_Value = p_value_sci # 加入科學記號欄位
      ))
    }
  }
}
# 查看結果
library(knitr)
kable(results, caption = "Correlation Test Results")
```

Table 3: Correlation Test Results

Variable1	Variable2	Correlation_Type	P_Value
Gender	Age	Kruskal-Wallis	8.33e-30
Gender	Sleep.Duration	Kruskal-Wallis	1.44e-02
Gender	Blood.Pressure	Kruskal-Wallis	3.55e-05
Gender	Heart.Rate	Kruskal-Wallis	3.56e-09
Age	Occupation	Kruskal-Wallis	4.04e-40
Age	Quality.of.Sleep	Kruskal-Wallis	3.79e-37
Age	Physical.Activity.Level	Kruskal-Wallis	1.79e-09
Age	Stress.Level	Kruskal-Wallis	2.04e-37
Age	BMI.Category	Kruskal-Wallis	3.27e-24
Age	Daily.Steps	Kruskal-Wallis	4.25e-03
Age	Sleep.Disorder	Kruskal-Wallis	2.99e-18
Occupation	Sleep.Duration	Kruskal-Wallis	8.53e-23

Variable1	Variable2	Correlation_Type	P_Value
Occupation	Blood.Pressure	Kruskal-Wallis	8.30e-48
Occupation	Heart.Rate	Kruskal-Wallis	2.25e-28
Sleep.Duration	Quality.of.Sleep	Kruskal-Wallis	3.73e-66
Sleep.Duration	Physical.Activity.Level	Kruskal-Wallis	3.47e-19
Sleep.Duration	Stress.Level	Kruskal-Wallis	1.76e-67
Sleep.Duration	BMI.Category	Kruskal-Wallis	6.18e-11
Sleep.Duration	Daily.Steps	Kruskal-Wallis	4.13e-11
Sleep.Duration	Sleep.Disorder	Kruskal-Wallis	3.63e-09
Quality.of.Sleep	Blood.Pressure	Kruskal-Wallis	3.30e-10
Quality.of.Sleep	Heart.Rate	Kruskal-Wallis	2.40e-46
Physical.Activity.Level	Blood.Pressure	Kruskal-Wallis	2.21e-17
Physical.Activity.Level	Heart.Rate	Kruskal-Wallis	2.88e-09
Stress.Level	Blood.Pressure	Kruskal-Wallis	2.31e-16
Stress.Level	Heart.Rate	Kruskal-Wallis	2.80e-55
BMI.Category	Blood.Pressure	Kruskal-Wallis	2.10e-48
BMI.Category	Heart.Rate	Kruskal-Wallis	6.39e-09
Blood.Pressure	Daily.Steps	Kruskal-Wallis	3.91e-05
Blood.Pressure	Sleep.Disorder	Kruskal-Wallis	2.59e-43
Heart.Rate	Daily.Steps	Kruskal-Wallis	7.17e-01
Heart.Rate	Sleep.Disorder	Kruskal-Wallis	7.89e-08

一些類別變數交互作用的圖

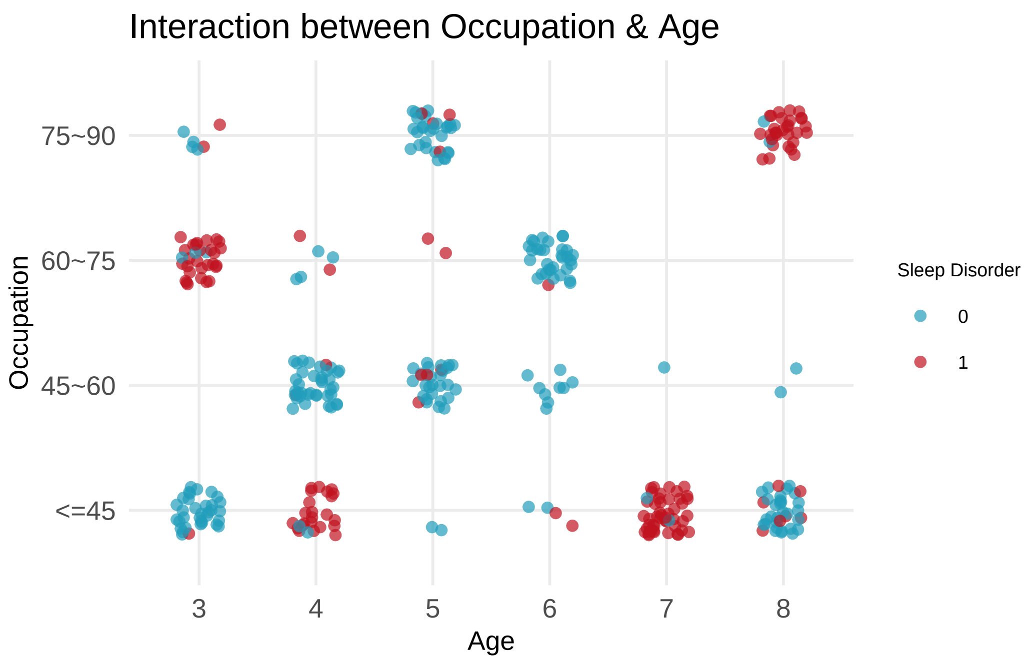
透過交互作用圖可以對變數之間的交互作用有更好的判斷與解讀

職業對變數的交互作用圖放在這

1. 年齡和職業

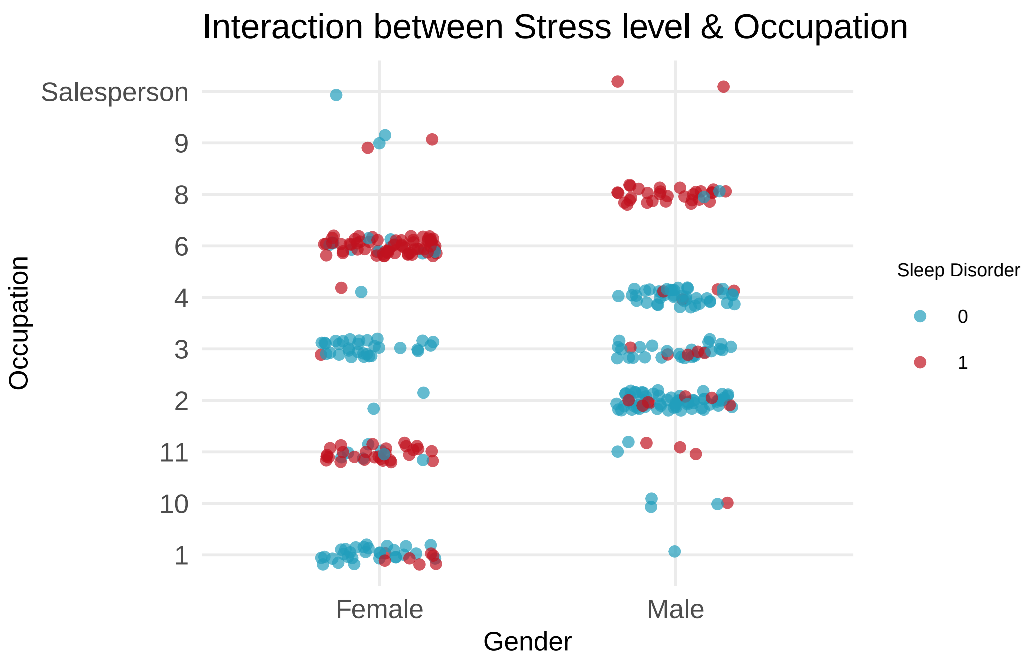
```
ggplot(data, aes(x = Stress.Level, y = Physical.Activity.Level,
                 color = Sleep.Disorder)) +
  geom_point(alpha = 0.7,
             position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc") # 根據 Sleep.Disorder 的值指定顏色
  ) +
  labs(
    title = "Interaction between Occupation & Age",
    x = "Age",
    y = "Occupation",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
```

```
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



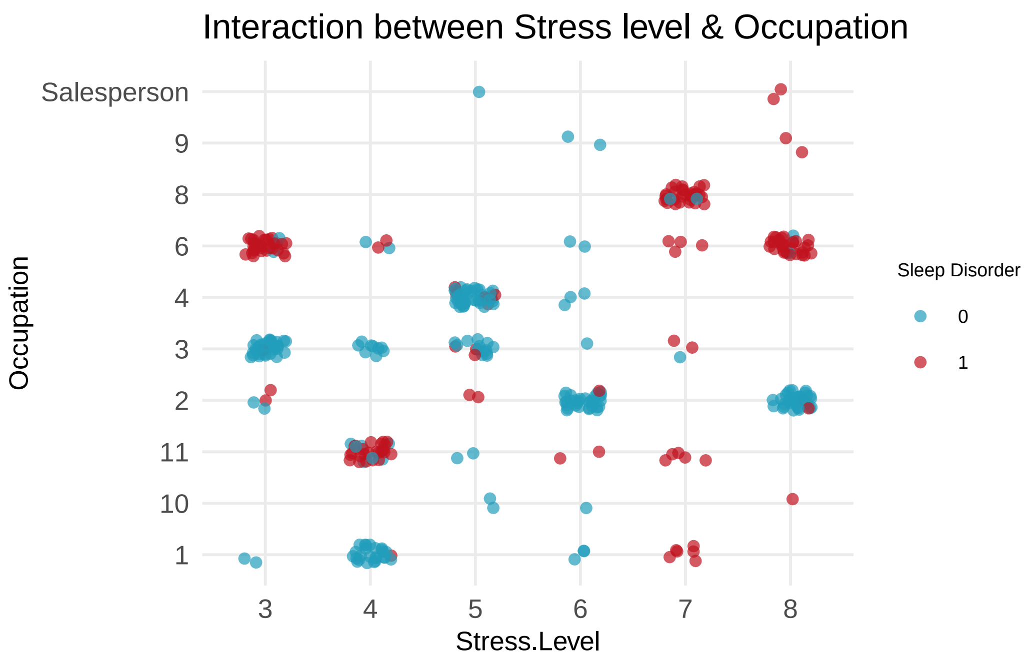
2. 性別和職業

```
ggplot(data, aes(x = Gender, y = Occupation, color = Sleep.Disorder)) +
  geom_point(alpha = 0.7, position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc") # 根據 Sleep.Disorder 的值指定顏色
  ) +
  labs(
    title = "Interaction between Stress level & Occupation",
    x = "Gender",
    y = "Occupation",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



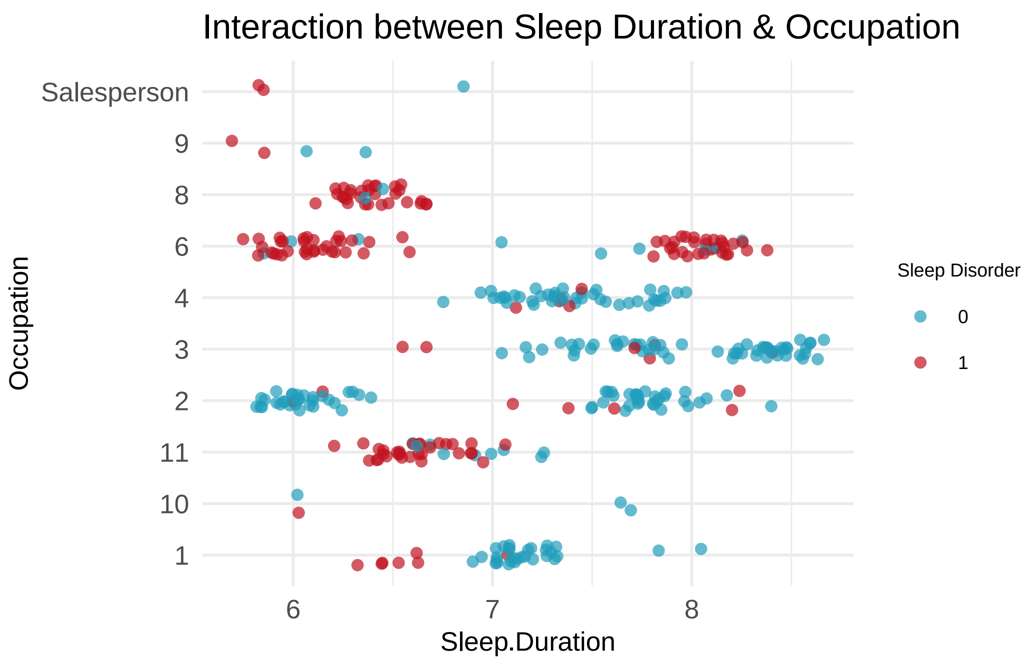
3. 壓力和職業

```
ggplot(data, aes(x = Stress.Level, y = Occupation, color = Sleep.Disorder)) +
  geom_point(alpha = 0.7, position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc") # 根據 Sleep.Disorder 的值指定顏色
  ) +
  labs(
    title = "Interaction between Stress level & Occupation",
    x = "Stress.Level",
    y = "Occupation",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



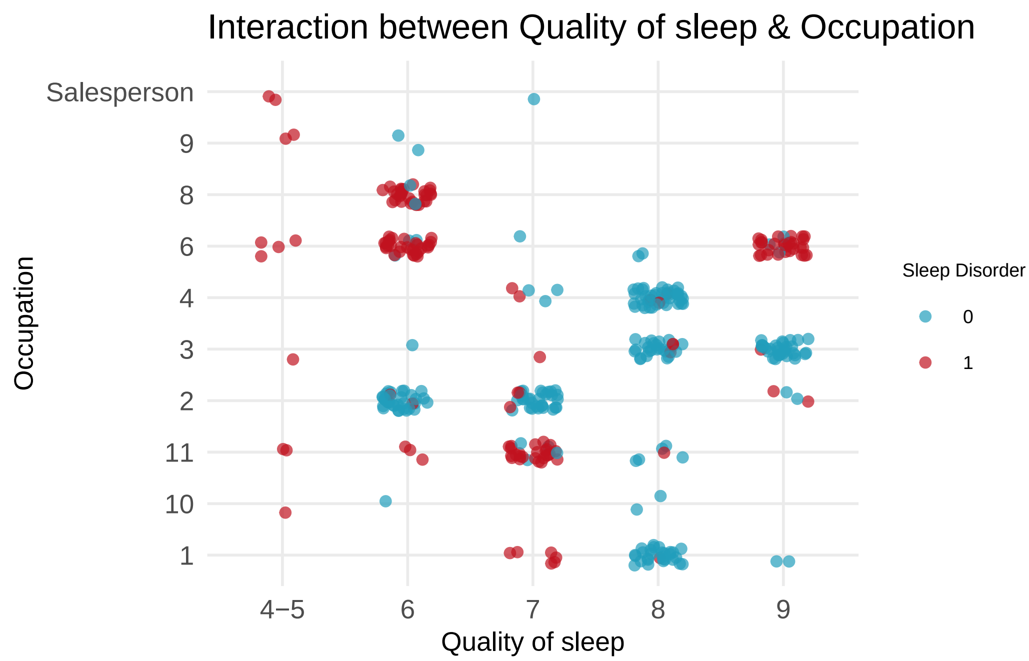
4. 睡眠時長和職業

```
ggplot(data, aes(x = Sleep.Duration, y = Occupation, color = Sleep.Disorder)) +
  geom_point(alpha = 0.7, position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc") # 根據 Sleep.Disorder 的值指定顏色
  ) +
  labs(
    title = "Interaction between Sleep Duration & Occupation",
    x = "Sleep.Duration",
    y = "Occupation",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



5. 睡眠品質和職業

```
ggplot(data, aes(x = Quality.of.Sleep, y = Occupation, color = Sleep.Disorder)) +
  geom_point(alpha = 0.7, position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc") # 根據 Sleep.Disorder 的值指定顏色
  ) +
  labs(
    title = "Interaction between Quality of sleep & Occupation",
    x = "Quality of sleep",
    y = "Occupation",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



結論:

1. 發現 Blood.Pressure、BMI.Category
無論是哪一種職業，對睡眠疾病皆可以看到明顯的分群

2. 幾乎所有護士、顯著比例的銷售人員和教師患有睡眠
疾病；醫生、會計師、工程師、律師則大部分皆無患睡眠疾病

3. 護士大多屬於女性，且年齡大多介於50-60歲、有較
高的BMI、血壓得到睡眠疾病，但在壓力水準下，卻有
極端分群，分別在壓力低和壓力高的群體有大部分的人
有睡眠疾病，同理在睡眠品質和睡眠時長下也有相似的
狀況

4. 而大部分的銷售人員年齡大多介於40-50歲，在患有睡
眠疾病下，同時具有較高的BMI、血壓、巨大壓力以及
睡眠時長短又品質較低的現象

5. 大部分的教師年齡大多介於40-50歲，在患有睡眠疾病
下，同時具有較高的BMI、血壓、睡眠時長短的現象

6. 患有睡眠疾病的人，貌似有較高的血壓、BMI、較年輕
、睡眠時長較短；沒有患病的人與之相反，這樣的情形
也顯示在職業上

其他感興趣想了解的變數交互作用圖

1.Sleep.Duration & Quality.of.Sleep

2.Sleep.Duration & Stress.Level

3.Physical.Activity.Level & (BMI、Quality.of.Sleep、Sleep Duration)

1.Sleep.Duration & Quality.of.Sleep

觀察 boxplot 第一張圖，整體趨勢可以大致看到隨著睡眠時長增加，睡眠品質呈現上升的趨勢。大部分人的睡眠品質較高時，睡眠時長在 7~8 小時之間。

普遍研究也認為，適當的睡眠時長與較高的睡眠品質相關。

而觀察散佈圖，看睡眠疾病 (紅色: 有睡眠疾病) 與睡眠時長的關係，可以發現過短或過長的睡眠時長與睡眠疾病之間可能也有密切的關聯，

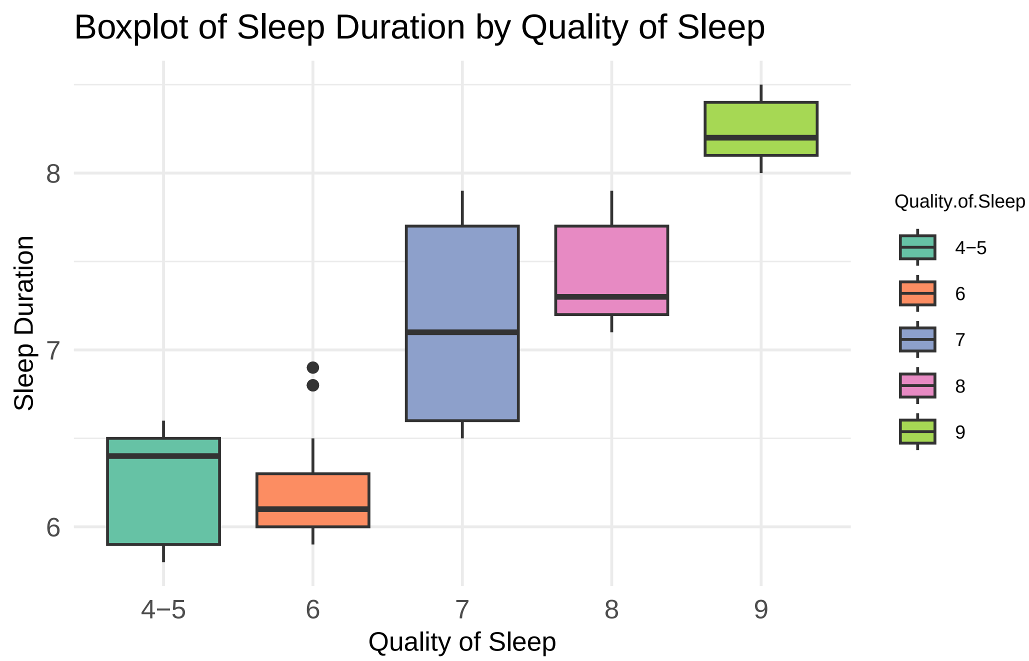
這裡可以從 Kruskal-Wallis 檢定的結果顯著 p-value:3.63E-09 證實，睡眠時長的變化可能會影響患睡眠疾病的風險。

綜合來看，睡眠品質、睡眠時長跟睡眠疾病有一定的相關。

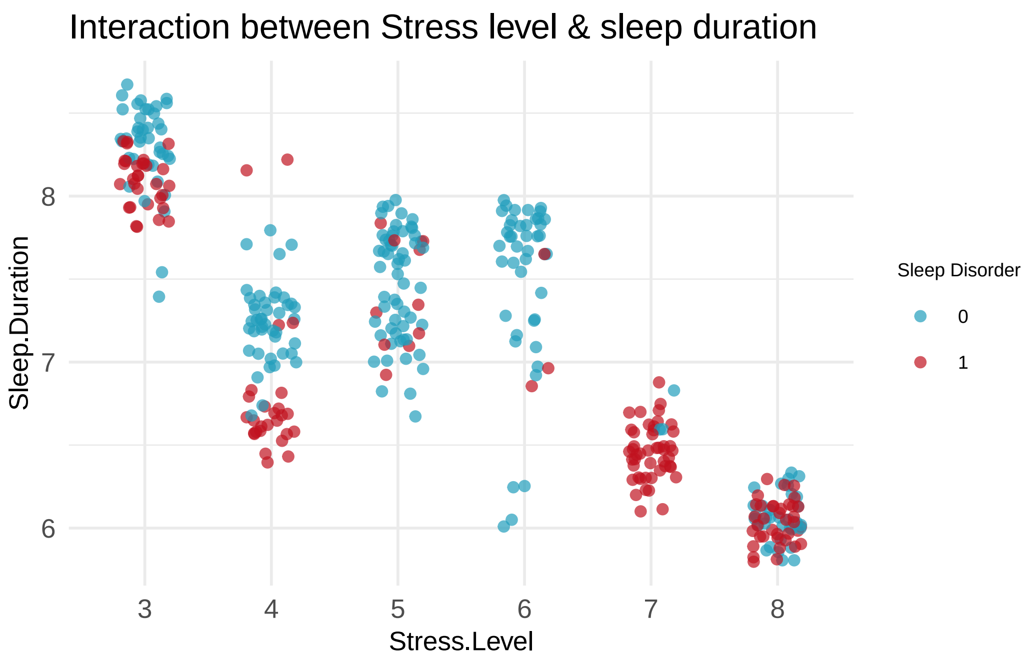
睡眠品質為 8 或 9 時，無睡眠疾病的群體（綠色）有稍長的睡眠時長；而睡眠疾病的群體在睡眠品質為 4-5、睡眠時長短 (6) 附近最多；

而雖然睡眠品質為 6-7 的範圍中，異常值較多，顯示此範圍內的睡眠時長變異性較大，但無睡眠疾病的群體似乎睡眠時長也較為稍長。

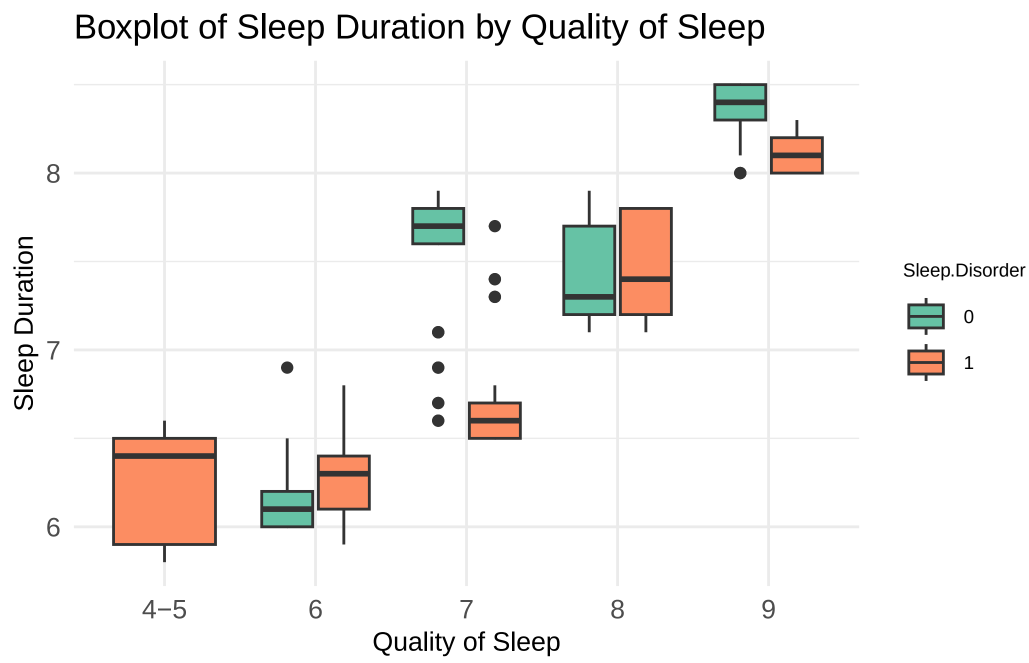
```
cat_var <- "Quality of Sleep"
cont_var <- "Sleep Duration"

ggplot(data, aes(x=Quality.of.Sleep, y=Sleep.Duration, fill=Quality.of.Sleep)) +
  geom_boxplot() +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = paste("Boxplot of", cont_var, "by", cat_var),
    x = cat_var,
    y = cont_var
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```




```
ggplot(data, aes(x = Stress.Level, y = Sleep.Duration, color = Sleep.Disorder)) +
  geom_point(alpha = 0.7, position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc")
  ) +
  labs(
    title = "Interaction between Stress level & sleep duration",
    x = "Stress.Level",
    y = "Sleep.Duration",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```




```
ggplot(data, aes(x=Quality.of.Sleep, y=Sleep.Duration, fill=Sleep.Disorder)) +
  geom_boxplot() +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = paste("Boxplot of", cont_var, "by", cat_var),
    x = cat_var,
    y = cont_var
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```

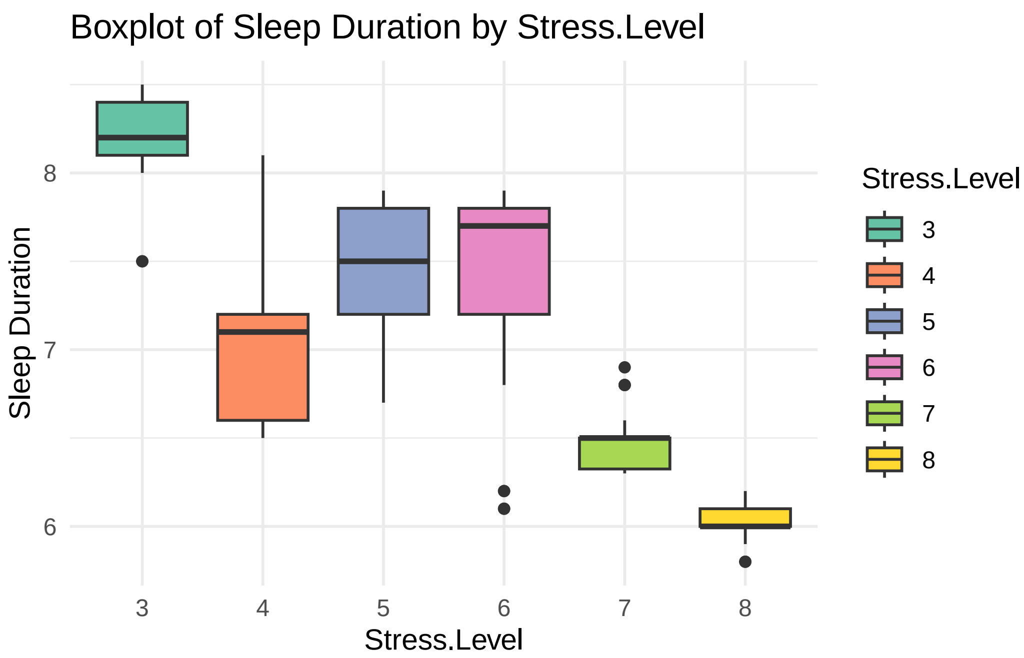


2. stress Level 和 Sleep Duration 的關聯

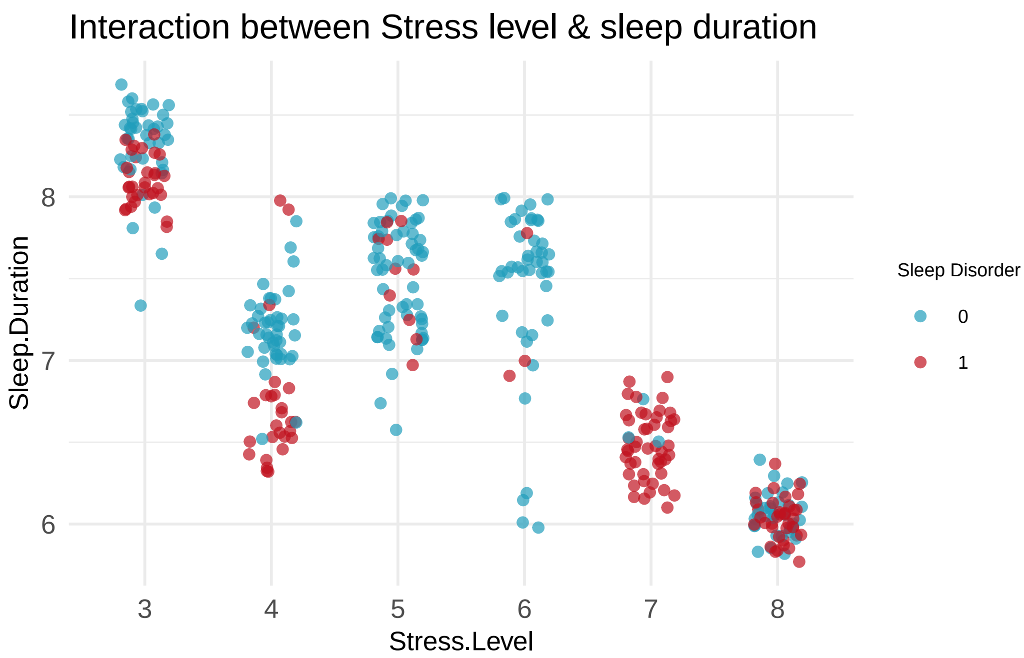
從 boxplot 圖可以觀察到，隨著壓力等級增加，睡眠時長呈現下降趨勢，例如壓力等級為 7 或 8 時，睡眠時長的中位數明顯減少。而當壓力等級較低（例如 3 或 4）時，睡眠時長分布集中且範圍較窄。

高壓力水平常與較短的睡眠時間相關。壓力會增加皮質醇的分泌，這可能干擾睡眠，導致失眠或睡眠質量差。長期高壓力也可能導致睡眠障礙，這反過來會進一步增加壓力，形成惡性循環。

```r
cat_var <- "Stress.Level"
cont_var <- "Sleep Duration"

ggplot(data, aes(x = `Stress.Level`, y = `Sleep.Duration`, fill = `Stress.Level`)) +
  geom_boxplot() +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = paste("Boxplot of", cont_var, "by", cat_var),
    x = cat_var,
    y = cont_var
  ) +
  theme_minimal()
```




```r
ggplot(data, aes(x = Stress.Level, y = Sleep.Duration, color = Sleep.Disorder)) +
  geom_point(alpha = 0.7, position = position_jitter(width = 0.2, height = 0.2)) +
  scale_color_manual(
    values = c("1" = "#c1121f", "0" = "#219ebc") # 根據 Sleep.Disorder 的值指定顏色
  ) +
  labs(
    title = "Interaction between Stress level & sleep duration",
    x = "Stress.Level",
    y = "Sleep.Duration",
    color = "Sleep Disorder"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



3.Physical.Activity.Level & (BMI、Quality.of.Sleep、Sleep Duration)

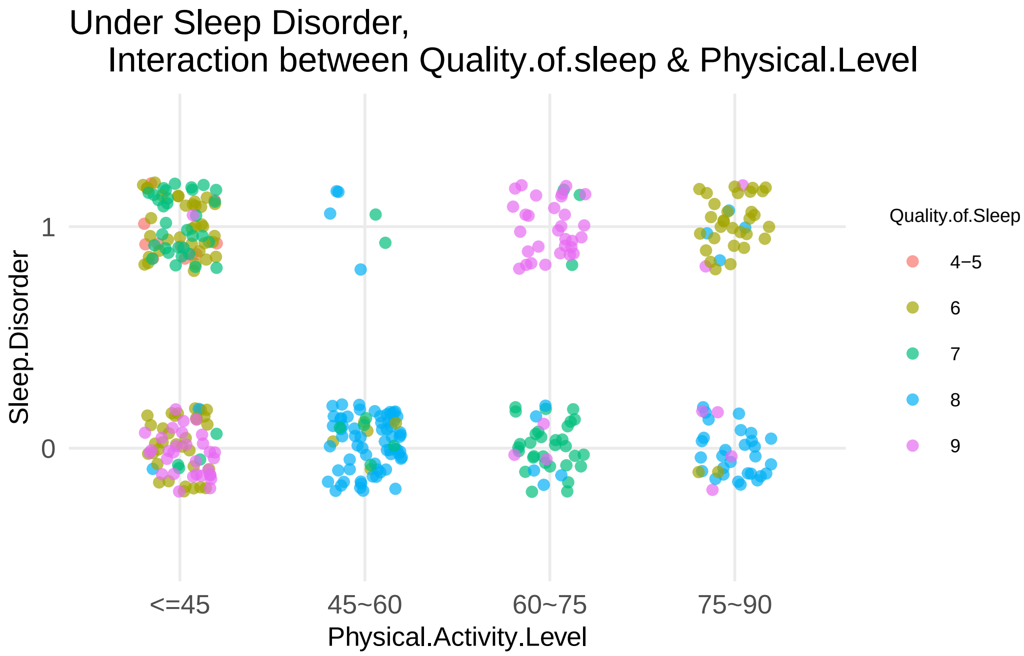
1. 身體活動水平與睡眠品質疾病關聯

從圖中可以觀察:

高身體活動水平 (60-90) 與較高的睡眠品質相關，特別在無睡眠障礙者中明顯；低身體活動水平 (<=45) 則與較低的睡眠品質相關，尤其對有睡眠障礙者影響顯著。

```
ggplot(data, aes(x = Physical.Activity.Level, y = Sleep.Disorder,
                 color = Quality.of.Sleep)) +
  geom_point(alpha = 0.7,
             position = position_jitter(width = 0.2, height = 0.2)) +
  #scale_color_manual(
   # values = c("Overweight" = "#c1121f", "Normal" = "#219ebc")) +
  labs(
    title = "Under Sleep Disorder,
    Interaction between Quality.of.sleep & Physical.Level",
    x = "Physical.Activity.Level",
    y = "Sleep.Disorder",
    color = "Quality.of.Sleep"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
```

```
)
```



2. 身體活動水平與睡眠時長疾病關聯

從圖中可以觀察:

無睡眠障礙者通常分佈在較長的睡眠時間範圍，尤其在高身體活動水平時；而有睡眠障礙者則集中於較短的睡眠時間，特別是在低身體活動水平下。

說明:

無睡眠障礙者通常分佈在較長的睡眠時間範圍，尤其在高身體活動水平 (60-90) 時。

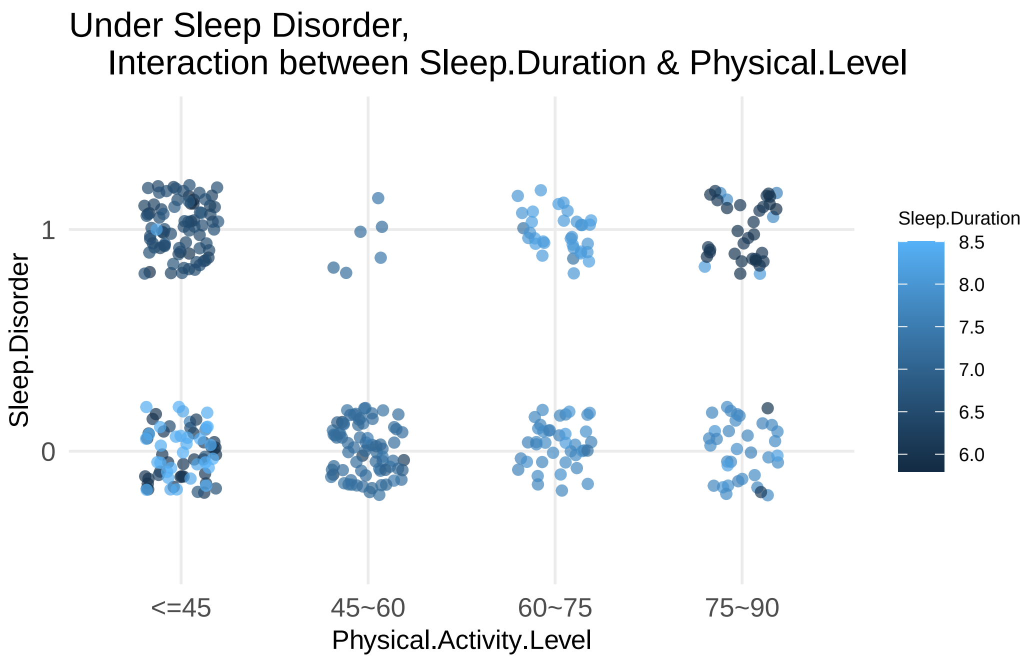
有睡眠障礙者則在較短的睡眠時間範圍內集中，尤其在低身體活動水平 (<=45) 時

高身體活動水平 (60-90) 通常與較長的睡眠時間相關，無論是否有睡眠障礙。

低身體活動水平 (<=45) 則與較短的睡眠時間相關，尤其是在有睡眠障礙的情況下。

```
ggplot(data, aes(x = Physical.Activity.Level, y = Sleep.Disorder,
                 color = Sleep.Duration)) +
  geom_point(alpha = 0.7,
             position = position_jitter(width = 0.2, height = 0.2)) +
  #scale_color_manual(
   # values = c("Overweight" = "#c1121f", "Normal" = "#219ebc")) +
  labs(
    title="Under Sleep Disorder,
    Interaction between Sleep.Duration & Physical.Level",
    x = "Physical.Activity.Level",
    y = "Sleep.Disorder",
    color = "Sleep.Duration"
  ) +
  theme_minimal()+
  theme(
```

```
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



總結解讀

身體活動水平對於睡眠時長和睡眠品質都有正向影響，尤其是在無睡眠障礙的情況下。高身體活動水平有助於延長睡眠時間和提高睡眠品質。

睡眠障礙者則在睡眠時間和品質都有顯著降低，即使有較高的身體活動水平，這種負面影響仍然存在。

綜合來看，增加身體活動水平可能是改善睡眠問題的一種有效策略，特別是在無睡眠障礙的情況下。

3. 身體活動水平與 BMI 疾病關聯

從圖中可以觀察：

過重的人在低身體活動水平下更容易出現睡眠障礙，而正常體重和適度身體活動水平的人群則較少出現睡眠障礙。

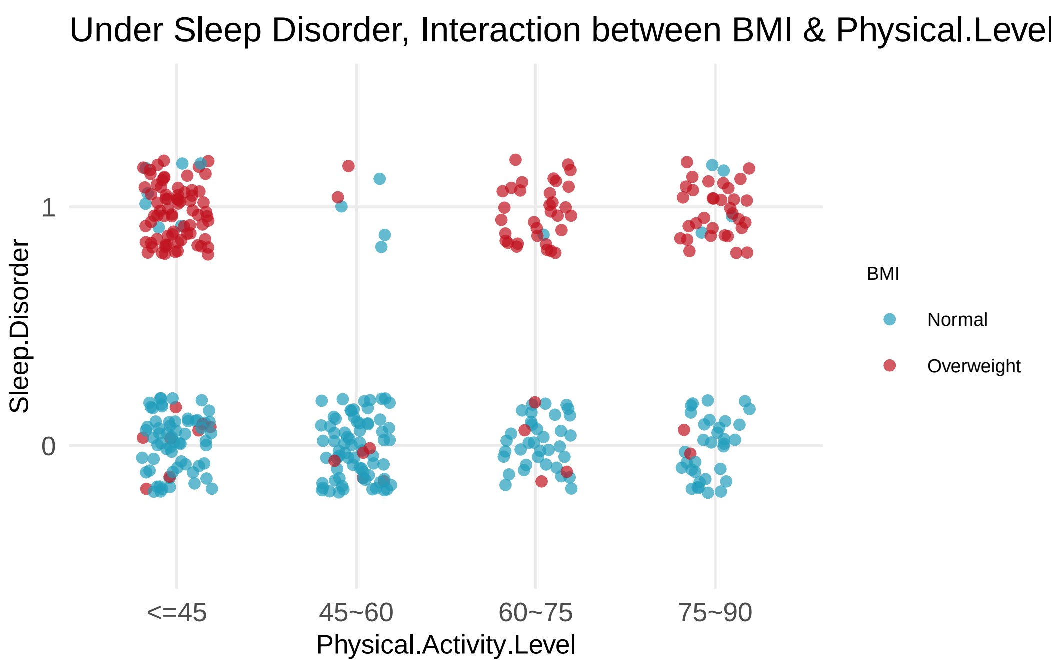
說明:

在 Sleep Disorder = Yes 的情況下，過重（紅色）樣本的數量似乎多於正常（藍色）樣本，特別是在較低的身體活動水平（<=45）

在 Sleep Disorder = No 的情況下，正常體重樣本的數量似乎較多，尤其是在較高的身體活動水平（46-75）

```
ggplot(data, aes(x = Physical.Activity.Level, y = Sleep.Disorder,
                 color = BMI.Category)) +
  geom_point(alpha = 0.7,
             position = position_jitter(width = 0.2, height = 0.2)) +
```

```
  scale_color_manual(
    values = c("Overweight" = "#c1121f", "Normal" = "#219ebc") # 根據 Sleep.Disorder 的
  ) +
  labs(
    title = "Under Sleep Disorder, Interaction between BMI & Physical.Level",
    x = "Physical.Activity.Level",
    y = "Sleep.Disorder",
    color = "BMI"
  ) +
  theme_minimal()+
  theme(
    plot.title = element_text(size = 13),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 7),
    legend.text = element_text(size = 7)
  )
```



3. Construct a predictive model for sleep disorder

由於我們想要找出跟睡眠疾病有關的可能因素，並兼顧模型的預測性能以及穩定性，這裡我們使用三種模型進行比較與評估，分別是 logistic regression、randomforest 以及 xgboost，以下是建置模型的流程:

1. 在各自模型中選取最佳的變數組合 (根據 Accuracy、Kappa、Specificity、Sensitivity、AUC 等指標綜合評估)

2. 對模型進行調參，使用 Grid Search(指定一組候選參數的範圍，穩定地嘗試所有可能的組合，並選擇最佳結果)

3. 由於我們的資料集屬於小樣本，最後透過 cross-validation 盡量減少過度擬合的影響

最後，在這三種模型之間做比較 (根據 Accuracy、Kappa、Specificity、Sensitivity、AUC 等指標綜合評估)，進而評估哪一種模型最好。

```
library(caret)          # For data partitioning and confusion matrix
library(ROCR)           # For ROC curve and AUC
library(pROC)
library(randomForest)
library(xgboost)
library(Matrix)
library(pscl)
library(glmnet)
library(MASS)
library(tidyr)
```

```
set.seed(014)
train_index <- createDataPartition(data$Sleep.Disorder, p = 0.8, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]
```

logistic regression

由於我們想要找出跟睡眠疾病有關的關鍵因素，並兼顧模型的預測性能以及穩定性，因此流程如下:

1. 使用四種方式 (所有變數/stepwise/Elastic Net/自選) 進行變數篩選

2. 再透過交叉驗證，確保所選模型在不同的數據子集上表現一致

3. 進一步評估模型的穩定性和泛化能力，並依據 Accuracy、Kappa、Specificity、Sensitivity、AUC 等指標，綜合考量後，挑選最終模型。

最終，我們選擇羅吉斯迴歸中的自選當作代表。

自選模型在各個指標表現都優於其他變數選擇的模型，並且具有以下優點:

1. 係數估計的 std.Error 都來的比其他還小 (0~1 左右) 且大部分顯著

2. 變數選擇較其他模型少 (4)，模型簡潔也具有較高解釋力 (AIC)

3. 共線性低 (GVIF^(1/(2*Df))) 皆在 5 以下，且都在 1~2 附近

在自選變數中，我們基於 EDA 分析、Background Knowledge 選的變數，基於多組變數組合嘗試後，最終選取 Blood.Pressure + BMI.Category + Stress.Level + Physical.Activity.Level，這組變數組合在解釋性和預測上達到最好的平衡。

以下是篩選的想法:

根據 EDA 分析->

優先選擇跟目標變數最有相關的變數:BMI、血壓、職業、睡眠品質、壓力

避免共線性問題，導致 std.Error 過大，估計不準確:

其中由於職業、睡眠品質跟多個變數具有蠻高的相關性，因此不放入

Background Knowledge->

Physical Activity Level: 基於運動對睡眠的益處，以及其可控性和公共衛生意義，將其納入模型

logistic regression(全放/共線性非常高)

Age + Gender + Occupation + Sleep.Duration + Quality.of.Sleep + Physical.Activity.Level + Stress.Level + BMI.Category + Blood.Pressure + Heart.Rate + Daily.Steps

```r
model <- glm(Sleep.Disorder ~ Age + Gender + Occupation + Sleep.Duration +
             Quality.of.Sleep + Physical.Activity.Level + Stress.Level +
             BMI.Category + Blood.Pressure + Heart.Rate + Daily.Steps,
             data = train_data, family = binomial())
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(model)
```

```

Call:
glm(formula = Sleep.Disorder ~ Age + Gender + Occupation + Sleep.Duration +
    Quality.of.Sleep + Physical.Activity.Level + Stress.Level +
    BMI.Category + Blood.Pressure + Heart.Rate + Daily.Steps,
    family = binomial(), data = train_data)

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -1.100e+03  1.283e+05  -0.009    0.993
Age                          -1.310e-01  3.814e-01  -0.343    0.731
GenderMale                    1.309e+01  1.303e+04   0.001    0.999
Occupation10                  3.562e+01  1.720e+04   0.002    0.998
Occupation11                  7.738e+01  8.363e+03   0.009    0.993
Occupation2                   1.119e+02  2.215e+04   0.005    0.996
Occupation3                   6.278e+01  1.331e+04   0.005    0.996
Occupation4                   6.107e+01  1.331e+04   0.005    0.996
Occupation6                   1.024e+02  1.330e+04   0.008    0.994
Occupation8                   9.838e+01  1.759e+04   0.006    0.996
Occupation9                   1.771e+02  1.150e+05   0.002    0.999
OccupationSalesperson         3.761e+01  4.176e+04   0.001    0.999
Sleep.Duration               -5.798e+00  3.573e+00  -1.623    0.105
Quality.of.Sleep6             3.997e+01  1.891e+04   0.002    0.998
Quality.of.Sleep7             1.721e+02  2.982e+04   0.006    0.995
Quality.of.Sleep8             1.891e+02  2.660e+04   0.007    0.994
Quality.of.Sleep9             2.733e+02  3.997e+04   0.007    0.995
Physical.Activity.Level45~60 -6.467e+01  6.629e+03  -0.010    0.992
Physical.Activity.Level60~75 -1.402e+02  1.628e+04  -0.009    0.993
Physical.Activity.Level75~90 -7.438e+01  7.535e+03  -0.010    0.992
Stress.Level4                 1.098e+02  1.810e+04   0.006    0.995
Stress.Level5                 6.445e+01  1.568e+04   0.004    0.997
Stress.Level6                 1.266e+02  1.881e+04   0.007    0.995
Stress.Level7                 1.779e+02  2.421e+04   0.007    0.994
Stress.Level8                 1.104e+02  2.373e+04   0.005    0.996
BMI.CategoryOverweight        7.441e+00  1.633e+04   0.000    1.000
```

```
Blood.Pressure              3.219e+00  7.001e+02   0.005    0.996
Heart.Rate                  6.374e+00  7.062e+02   0.009    0.993
Daily.Steps5001~7500       -6.837e+01  1.046e+04  -0.007    0.995
Daily.Steps7500up           2.996e+01  3.888e+03   0.008    0.994

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 406.83  on 299  degrees of freedom
Residual deviance: 121.64  on 270  degrees of freedom
AIC: 181.64

Number of Fisher Scoring iterations: 20
```

```r
predicted_probabilities <- predict(model, newdata = test_data, type = "response")
predicted_classes <- ifelse(predicted_probabilities > 0.4, 1, 0)

# Confusion Matrix
confusion_matrix <- confusionMatrix(as.factor(predicted_classes),
                                    test_data$Sleep.Disorder)
acc_all <- confusion_matrix$overall[1]
sen_all <- confusion_matrix$byClass[1]
spe_all <- confusion_matrix$byClass[2]
print(confusion_matrix)
```

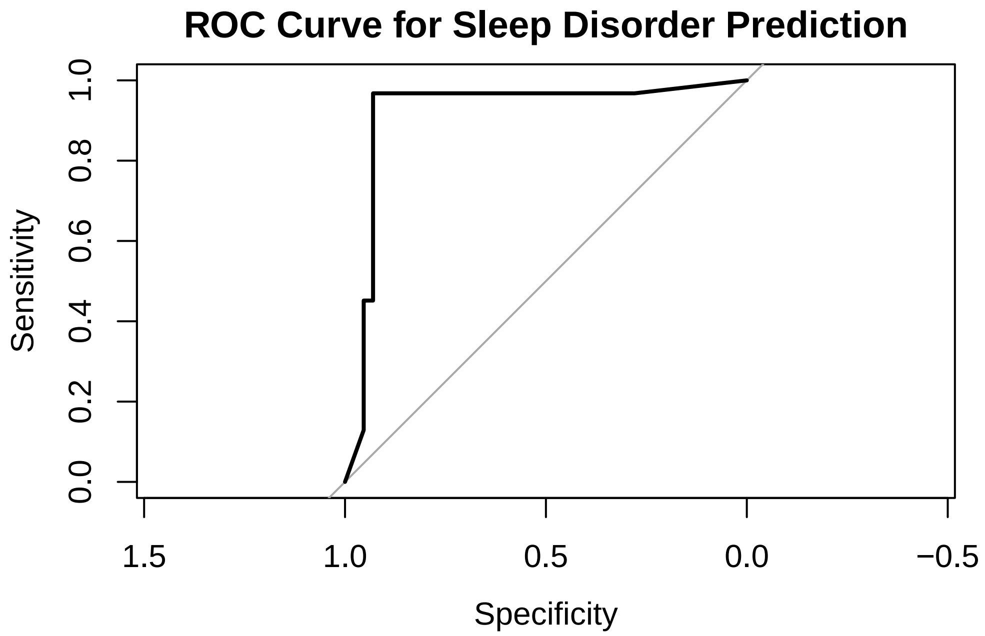
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 40  1
         1  3 30

               Accuracy : 0.9459
                 95% CI : (0.8673, 0.9851)
    No Information Rate : 0.5811
    P-Value [Acc > NIR] : 1.204e-12

                  Kappa : 0.89

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9302
            Specificity : 0.9677
         Pos Pred Value : 0.9756
         Neg Pred Value : 0.9091
             Prevalence : 0.5811
         Detection Rate : 0.5405
   Detection Prevalence : 0.5541
      Balanced Accuracy : 0.9490
```

```
       'Positive' Class : 0
```

```r
# ROC
roc_curve1 <- roc(test_data$Sleep.Disorder, predicted_probabilities)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```r
plot(roc_curve1, main = "ROC Curve for Sleep Disorder Prediction")
```




```r
auc_all <- auc(roc_curve1)
print(paste("AUC:", auc_all))
```

```
[1] "AUC: 0.918229557389347"
```

```r
vif(model)
```

```
                                GVIF Df GVIF^(1/(2*Df))
Age                     1.823150e+02  1        13.50241
Gender                  6.909196e+08  1     26285.34902
Occupation              2.271234e+34  9        81.03660
Sleep.Duration          1.391057e+02  1        11.79431
Quality.of.Sleep        9.016783e+26  4      2340.89239
Physical.Activity.Level 8.979900e+22  3      6691.83541
Stress.Level            2.290843e+41  5     13677.27346
BMI.Category            1.087336e+09  1     32974.77364
Blood.Pressure          4.193124e+08  1     20477.11790
Heart.Rate              6.129962e+07  1      7829.40721
Daily.Steps             4.060233e+15  2      7982.47814
```

logistic regression(stepwise 挑變數/共線性高)

Sleep.Duration + Quality.of.Sleep + Physical.Activity.Level + Stress.Level + BMI.Category + Daily.Steps

```r
library(MASS)
model <- glm(Sleep.Disorder ~ Age + Gender + Occupation + Sleep.Duration +
             Quality.of.Sleep + Physical.Activity.Level + Stress.Level +
             BMI.Category + Blood.Pressure +
             Heart.Rate + Daily.Steps,
             data = train_data, family = binomial())

logistic_model_step <- stepAIC(model, direction = "both")
```

```
Start:  AIC=181.64
Sleep.Disorder ~ Age + Gender + Occupation + Sleep.Duration +
    Quality.of.Sleep + Physical.Activity.Level + Stress.Level +
    BMI.Category + Blood.Pressure + Heart.Rate + Daily.Steps

                          Df Deviance    AIC
- Occupation               9   129.15 171.15
- Quality.of.Sleep         4   121.64 173.64
- Stress.Level             5   129.29 179.29
- BMI.Category             1   121.64 179.64
- Gender                   1   121.64 179.64
- Age                      1   121.76 179.76
- Heart.Rate               1   121.76 179.76
- Physical.Activity.Level  3   126.17 180.17
<none>                         121.64 181.64
- Sleep.Duration           1   124.40 182.40
- Blood.Pressure           1   124.97 182.97
- Daily.Steps              2   128.16 184.16

Step:  AIC=171.15
Sleep.Disorder ~ Age + Gender + Sleep.Duration + Quality.of.Sleep +
    Physical.Activity.Level + Stress.Level + BMI.Category + Blood.Pressure +
    Heart.Rate + Daily.Steps

                          Df Deviance    AIC
- BMI.Category             1   129.21 169.21
- Gender                   1   129.44 169.44
- Age                      1   129.47 169.47
- Heart.Rate               1   130.02 170.02
- Quality.of.Sleep         4   136.62 170.62
<none>                         129.15 171.15
- Sleep.Duration           1   131.46 171.46
- Physical.Activity.Level  3   138.85 174.85
- Daily.Steps              2   137.07 175.07
- Stress.Level             5   147.24 179.24
```

```
- Blood.Pressure           1   139.36 179.36
+ Occupation               9   121.64 181.64

Step:  AIC=169.21
Sleep.Disorder ~ Age + Gender + Sleep.Duration + Quality.of.Sleep +
    Physical.Activity.Level + Stress.Level + Blood.Pressure +
    Heart.Rate + Daily.Steps

                          Df Deviance    AIC
- Age                      1   129.48 167.48
- Gender                   1   129.55 167.55
<none>                         129.21 169.21
- Quality.of.Sleep         4   137.25 169.25
- Sleep.Duration           1   131.47 169.47
- Heart.Rate               1   131.53 169.53
+ BMI.Category             1   129.15 171.15
- Physical.Activity.Level  3   138.85 172.85
- Daily.Steps              2   137.07 173.07
- Stress.Level             5   147.33 177.33
- Blood.Pressure           1   141.17 179.17
+ Occupation               9   121.64 179.64

Step:  AIC=167.48
Sleep.Disorder ~ Gender + Sleep.Duration + Quality.of.Sleep +
    Physical.Activity.Level + Stress.Level + Blood.Pressure +
    Heart.Rate + Daily.Steps

                          Df Deviance    AIC
- Gender                   1   129.62 165.62
- Quality.of.Sleep         4   137.30 167.30
<none>                         129.48 167.48
- Sleep.Duration           1   131.57 167.57
- Heart.Rate               1   132.83 168.83
+ Age                      1   129.21 169.21
+ BMI.Category             1   129.47 169.47
- Physical.Activity.Level  3   138.85 170.85
- Daily.Steps              2   138.01 172.01
+ Occupation               9   121.76 177.76
- Stress.Level             5   149.85 177.85
- Blood.Pressure           1   147.90 183.90

Step:  AIC=165.62
Sleep.Disorder ~ Sleep.Duration + Quality.of.Sleep + Physical.Activity.Level +
    Stress.Level + Blood.Pressure + Heart.Rate + Daily.Steps

                          Df Deviance    AIC
<none>                         129.62 165.62
- Sleep.Duration           1   131.65 165.65
```

```
- Quality.of.Sleep         4   138.76 166.76
- Heart.Rate               1   132.94 166.94
+ Gender                   1   129.48 167.48
+ Age                      1   129.55 167.55
+ BMI.Category             1   129.56 167.56
- Physical.Activity.Level  3   139.29 169.29
- Daily.Steps              2   139.48 171.48
+ Occupation               9   121.76 175.76
- Stress.Level             5   152.93 178.93
- Blood.Pressure           1   158.11 192.11
```

```r
summary(logistic_model_step)
```

```

Call:
glm(formula = Sleep.Disorder ~ Sleep.Duration + Quality.of.Sleep +
    Physical.Activity.Level + Stress.Level + Blood.Pressure +
    Heart.Rate + Daily.Steps, family = binomial(), data = train_data)

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -74.8668  5923.1827  -0.013   0.9899
Sleep.Duration                 -4.1580     3.0284  -1.373   0.1697
Quality.of.Sleep6             -21.2659  2207.2329  -0.010   0.9923
Quality.of.Sleep7               7.9679  3273.0208   0.002   0.9981
Quality.of.Sleep8               6.1934  3273.0221   0.002   0.9985
Quality.of.Sleep9              29.9454  5923.0317   0.005   0.9960
Physical.Activity.Level45~60   -7.8359     4.0813  -1.920   0.0549 .
Physical.Activity.Level60~75   -8.3723     4.3743  -1.914   0.0556 .
Physical.Activity.Level75~90   -6.1017     4.2267  -1.444   0.1488
Stress.Level4                  24.3259  4936.5596   0.005   0.9961
Stress.Level5                  19.7103  4936.5582   0.004   0.9968
Stress.Level6                  23.1702  4936.5594   0.005   0.9963
Stress.Level7                  51.6351  5496.4066   0.009   0.9925
Stress.Level8                  38.5977  5496.4005   0.007   0.9944
Blood.Pressure                  0.4234     0.1652   2.563   0.0104 *
Heart.Rate                      0.3688     0.2368   1.557   0.1194
Daily.Steps5001~7500           -7.8605     6.0778  -1.293   0.1959
Daily.Steps7500up               4.5210     2.0229   2.235   0.0254 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 406.83  on 299  degrees of freedom
Residual deviance: 129.62  on 282  degrees of freedom
AIC: 165.62

Number of Fisher Scoring iterations: 18
```

```r
vif(logistic_model_step)
```

```
                                 GVIF Df GVIF^(1/(2*Df))
Sleep.Duration          9.975028e+01  1        9.987506
Quality.of.Sleep        9.811062e+15  4       99.761852
Physical.Activity.Level 2.274635e+03  3        3.626480
Stress.Level            8.769556e+17  5       62.272710
Blood.Pressure          2.339624e+01  1        4.836966
Heart.Rate              8.166992e+00  1        2.857795
Daily.Steps             5.861185e+02  2        4.920354
```

```r
pseudo_r2 <- pR2(logistic_model_step)
```

```
fitting null model for pseudo-r2
```

```r
print(pseudo_r2)
```

```
         llh      llhNull           G2     McFadden         r2ML         r2CU
 -64.8100005 -203.4146451  277.2092892    0.6813897    0.6030841    0.8124143
```

```r
predicted_probs <- predict(logistic_model_step,newdata=test_data,type = "response")
predicted_classes <- ifelse(predicted_probs > 0.4, 1, 0)
conf_matrix <- confusionMatrix(as.factor(predicted_classes),
                               as.factor(test_data$Sleep.Disorder))
print(conf_matrix)
```

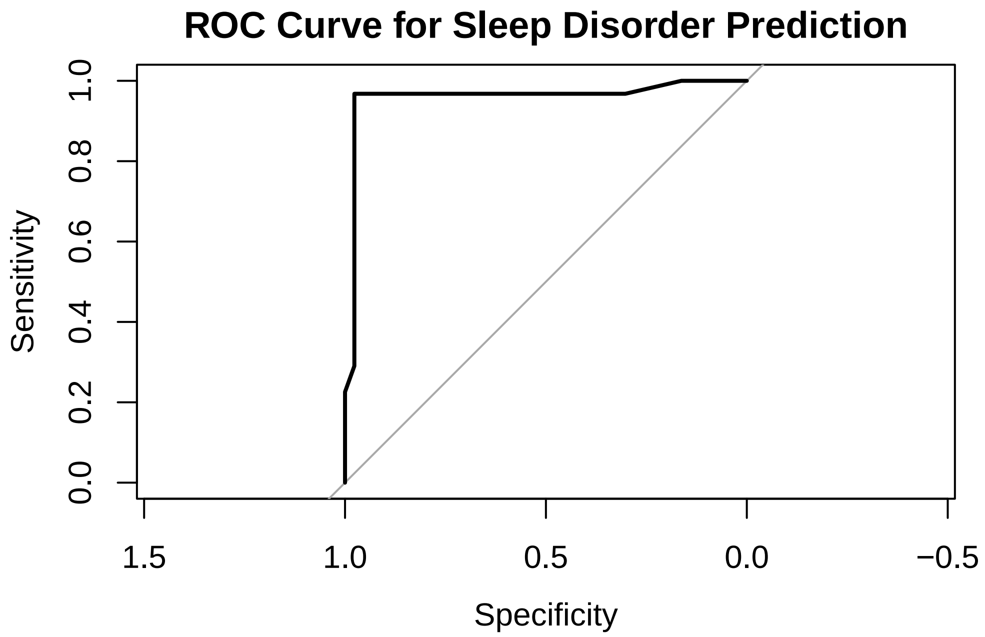
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 42  1
         1  1 30

               Accuracy : 0.973
                 95% CI : (0.9058, 0.9967)
    No Information Rate : 0.5811
    P-Value [Acc > NIR] : 5.216e-15

                  Kappa : 0.9445

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9767
            Specificity : 0.9677
         Pos Pred Value : 0.9767
         Neg Pred Value : 0.9677
             Prevalence : 0.5811
         Detection Rate : 0.5676
   Detection Prevalence : 0.5811
      Balanced Accuracy : 0.9722
```

```
        'Positive' Class : 0
```

```r
acc_step <- conf_matrix$overall[1]
sen_step <- conf_matrix$byClass[1]
spe_step <- conf_matrix$byClass[2]
# ROC
roc_curve2 <- roc(test_data$Sleep.Disorder, predicted_probs)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```r
plot(roc_curve2, main = "ROC Curve for Sleep Disorder Prediction")
```




```r
auc_step <- auc(roc_curve2)
```

logistic regression(Elastic net/共線性高)

Occupation + Sleep.Duration + Quality.of.Sleep + Physical.Activity.Level + Stress.Level + BMI.Category + Blood.Pressure + Heart.Rate + Gender

```r
library(glmnet)

# 訓練 Elastic Net 模型
variablenames <- names(data)[-c(12)]
formula.x <- formula(paste("~", paste(variablenames, collapse=" + ")))
X <- model.matrix(formula.x, data)
y <- data$Sleep.Disorder

## Using cross validation folds to select lambda.
```

```
cv <- cv.glmnet(x=X, y=y, family = "binomial",  alpha = 0.5)
coefs <- coef(cv, s=cv$lambda.1se)
best_lambda <- cv$lambda.min
print(best_lambda)
```

```
[1] 0.005236659
```

```
fre.variables <- names(coefs[which(coefs[,1]!=0),1])
fre.variables
```

```
 [1] "(Intercept)"                "GenderMale"
 [3] "Occupation11"               "Occupation4"
 [5] "Occupation6"                "Sleep.Duration"
 [7] "Quality.of.Sleep8"          "Physical.Activity.Level45~60"
 [9] "Stress.Level6"              "Stress.Level7"
[11] "BMI.CategoryOverweight"     "Blood.Pressure"
[13] "Heart.Rate"
```

```
logistic_model_select <- glm(Sleep.Disorder ~ Blood.Pressure  + Stress.Level +
                              Sleep.Duration+ Occupation +Heart.Rate +
                              Physical.Activity.Level + BMI.Category +
                              Quality.of.Sleep + Gender,
                              data = train_data, family = binomial())
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logistic_model_select)
```

```
Call:
glm(formula = Sleep.Disorder ~ Blood.Pressure + Stress.Level +
    Sleep.Duration + Occupation + Heart.Rate + Physical.Activity.Level +
    BMI.Category + Quality.of.Sleep + Gender, family = binomial(),
    data = train_data)

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -4.696e+01  8.784e+03  -0.005   0.9957
Blood.Pressure               2.655e-01  1.254e-01   2.117   0.0342 *
Stress.Level4                1.642e+01  5.025e+03   0.003   0.9974
Stress.Level5                1.863e+01  5.025e+03   0.004   0.9970
Stress.Level6                1.805e+01  5.025e+03   0.004   0.9971
Stress.Level7                5.093e+01  6.277e+03   0.008   0.9935
Stress.Level8                3.406e+01  7.263e+03   0.005   0.9963
Sleep.Duration              -2.066e+00  2.307e+00  -0.895   0.3707
Occupation10                 3.346e+00  7.666e+03   0.000   0.9997
Occupation11                 1.827e+01  1.982e+03   0.009   0.9926
Occupation2                  2.246e+01  1.982e+03   0.011   0.9910
Occupation3                  1.829e+01  1.982e+03   0.009   0.9926
Occupation4                  1.697e+01  1.982e+03   0.009   0.9932
```

```
Occupation6                  1.628e+01  1.982e+03   0.008   0.9934
Occupation8                  8.004e+00  4.158e+03   0.002   0.9985
Occupation9                  6.811e+00  1.737e+05   0.000   1.0000
OccupationSalesperson        2.816e-02  3.230e+03   0.000   1.0000
Heart.Rate                   1.049e-01  1.591e-01   0.659   0.5098
Physical.Activity.Level45~60 -1.279e+00  2.508e+00  -0.510   0.6099
Physical.Activity.Level60~75  8.004e-01  4.363e+00   0.183   0.8544
Physical.Activity.Level75~90 -7.319e-01  2.850e+00  -0.257   0.7974
BMI.CategoryOverweight        1.810e+00  3.274e+00   0.553   0.5805
Quality.of.Sleep6            -3.460e+01  3.550e+03  -0.010   0.9922
Quality.of.Sleep7            -1.573e+01  6.334e+03  -0.002   0.9980
Quality.of.Sleep8            -1.096e+01  6.334e+03  -0.002   0.9986
Quality.of.Sleep9             2.477e+00  8.085e+03   0.000   0.9998
GenderMale                   -5.472e+00  3.219e+00  -1.700   0.0891 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 406.83  on 299  degrees of freedom
Residual deviance: 129.83  on 273  degrees of freedom
AIC: 183.83

Number of Fisher Scoring iterations: 18
```

```r
vif(logistic_model_select)
```

```
                                GVIF Df GVIF^(1/(2*Df))
Blood.Pressure          1.325550e+01  1        3.640810
Stress.Level            6.780779e+24  5      304.178111
Sleep.Duration          6.128154e+01  1        7.828253
Occupation              1.191197e+12  9        4.686925
Heart.Rate              4.558447e+00  1        2.135052
Physical.Activity.Level 1.784192e+03  3        3.482627
BMI.Category            4.651564e+01  1        6.820238
Quality.of.Sleep        3.592374e+17  4      156.466979
Gender                  4.456013e+01  1        6.675337
```

```r
pseudo_r2 <- pR2(logistic_model_select)
```

```
fitting null model for pseudo-r2
```

```r
print(pseudo_r2)
```

```
         llh      llhNull           G2     McFadden         r2ML         r2CU
 -64.9141029 -203.4146451  277.0010844    0.6808779    0.6028086    0.8120431
```

```r
predicted_probs <- predict(logistic_model_select,newdata=test_data,type="response")
predicted_classes <- ifelse(predicted_probs > 0.4, 1, 0)
library(caret)
conf_matrix <- confusionMatrix(as.factor(predicted_classes),
```

```
                                as.factor(test_data$Sleep.Disorder))
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 41  2
         1  2 29

               Accuracy : 0.9459
                 95% CI : (0.8673, 0.9851)
    No Information Rate : 0.5811
    P-Value [Acc > NIR] : 1.204e-12

                  Kappa : 0.889

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9535
            Specificity : 0.9355
         Pos Pred Value : 0.9535
         Neg Pred Value : 0.9355
             Prevalence : 0.5811
         Detection Rate : 0.5541
   Detection Prevalence : 0.5811
      Balanced Accuracy : 0.9445

       'Positive' Class : 0
```

```
roc_elastic <- roc(test_data$Sleep.Disorder, predicted_classes)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
acc_ela <- conf_matrix$overall[1]
sen_ela <- conf_matrix$byClass[1]
spe_ela <- conf_matrix$byClass[2]
auc_ela <-auc(roc_elastic)
```

logistic regression(手選變數 by 變數間相關係數/scatter plot/共線性解決)

變數選取: Blood.Pressure + BMI.Category + Stress.Level + Physical.Activity.Level

```
logistic_model_original <- glm(Sleep.Disorder ~ Blood.Pressure + BMI.Category  +
                                Stress.Level + Physical.Activity.Level,
                                data = train_data, family = binomial())
summary(logistic_model_original)
```

```
Call:
glm(formula = Sleep.Disorder ~ Blood.Pressure + BMI.Category +
    Stress.Level + Physical.Activity.Level, family = binomial(),
    data = train_data)

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -38.67259    9.60224  -4.027 5.64e-05 ***
Blood.Pressure                 0.28458    0.07541   3.774 0.000161 ***
BMI.CategoryOverweight         0.89536    0.88360   1.013 0.310911
Stress.Level4                  2.14511    1.02459   2.094 0.036294 *
Stress.Level5                  0.26160    0.95272   0.275 0.783637
Stress.Level6                  0.72261    1.15843   0.624 0.532769
Stress.Level7                  3.80540    1.11274   3.420 0.000627 ***
Stress.Level8                  0.99930    0.86760   1.152 0.249405
Physical.Activity.Level45~60  -0.91096    0.81485  -1.118 0.263587
Physical.Activity.Level60~75  -0.05316    0.85957  -0.062 0.950690
Physical.Activity.Level75~90  -0.40708    0.81081  -0.502 0.615624
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 406.83  on 299  degrees of freedom
Residual deviance: 147.55  on 289  degrees of freedom
AIC: 169.55

Number of Fisher Scoring iterations: 6
```

```
library(car)
vif(logistic_model_original)
```

```
                            GVIF Df GVIF^(1/(2*Df))
Blood.Pressure          4.112830  1        2.028011
BMI.Category            3.880791  1        1.969972
Stress.Level            6.411097  5        1.204185
Physical.Activity.Level 4.718280  3        1.295082
```

```
library(pscl)
pseudo_r2 <- pR2(logistic_model_original)
```

```
fitting null model for pseudo-r2
```

```
print(pseudo_r2)
```

```
         llh      llhNull           G2     McFadden         r2ML         r2CU
 -73.7735540 -203.4146451  259.2821824    0.6373243    0.5786426    0.7794892
```

```
predicted_probs <- predict(logistic_model_original,newdata=test_data,type="response")
predicted_classes <- ifelse(predicted_probs > 0.4, 1, 0)
```

```r
library(caret)
conf_matrix <- confusionMatrix(as.factor(predicted_classes),
                               as.factor(test_data$Sleep.Disorder))
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 42  1
         1  1 30

               Accuracy : 0.973
                 95% CI : (0.9058, 0.9967)
    No Information Rate : 0.5811
    P-Value [Acc > NIR] : 5.216e-15

                  Kappa : 0.9445

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9767
            Specificity : 0.9677
         Pos Pred Value : 0.9767
         Neg Pred Value : 0.9677
             Prevalence : 0.5811
         Detection Rate : 0.5676
   Detection Prevalence : 0.5811
      Balanced Accuracy : 0.9722

       'Positive' Class : 0
```

```r
roc_manual <- roc(test_data$Sleep.Disorder, predicted_classes)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```r
acc_self <- conf_matrix$overall[1]
sen_self <- conf_matrix$byClass[1]
spe_self <- conf_matrix$byClass[2]
auc_self <- auc(roc_manual)
logistic_model_steps <- glm(Sleep.Disorder ~ Blood.Pressure + BMI.Category,
                            data = train_data, family = binomial())
anova(logistic_model_steps, logistic_model_original, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: Sleep.Disorder ~ Blood.Pressure + BMI.Category
```

```
Model 2: Sleep.Disorder ~ Blood.Pressure + BMI.Category + Stress.Level +
    Physical.Activity.Level
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       297     180.18
2       289     147.55  8   32.634 7.167e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

logistic comparison(無 cross validation)

```r
results <- data.frame(
  Method = c("All Variables","Stepwise","Elastic Net","Manual Selection"),
  Accuracy = c(acc_all,acc_step,acc_ela,acc_self),
  Sensitivity = c(sen_all,sen_step,sen_ela,sen_self),
  Specificity = c(spe_all,spe_step,spe_ela,spe_self),
  AUC = c(auc_all,auc_step,auc_ela,auc_self)
)
print(results)
```

```
            Method  Accuracy Sensitivity Specificity       AUC
1    All Variables 0.9459459   0.9302326   0.9677419 0.9182296
2         Stepwise 0.9729730   0.9767442   0.9677419 0.9587397
3      Elastic Net 0.9459459   0.9534884   0.9354839 0.9444861
4 Manual Selection 0.9729730   0.9767442   0.9677419 0.9722431
```

logistic + cross validation + comparison

```r
# 自定義評估函數
levels(data$Sleep.Disorder) <- c("No", "Yes")

custom_summary <- function(data, lev = NULL, model = NULL) {
  cm <- confusionMatrix(as.factor(data$pred), as.factor(data$obs))
  roc_curve <- roc(response = data$obs, predictor = data$Yes, levels = rev(lev))
  auc_value <- auc(roc_curve)
  # 返回所需的指標
  out <- c(
    Accuracy = cm$overall["Accuracy"],
    Kappa = cm$overall["Kappa"],
    Sensitivity = cm$byClass["Sensitivity"],
    Specificity = cm$byClass["Specificity"],
    AUC = auc_value
  )
  return(out)
}
train_control <- trainControl(
  method = "cv",            # Cross-validation
  number = 5,              # 5-fold cross-validation
  classProbs = TRUE,        # 計算概率
```

```r
  summaryFunction = custom_summary, # 自定義評估函數
)
set.seed(014)
# 所有變數模型
model_all <- train(Sleep.Disorder ~ Age + Gender + Occupation +
                   Sleep.Duration + Quality.of.Sleep + Physical.Activity.Level +
                   Stress.Level + BMI.Category + Blood.Pressure +
                   Heart.Rate + Daily.Steps,
                   data = data, method = "glm", family = "binomial",
                   trControl = train_control)

# Stepwise 變數選擇模型
model_step <- train(Sleep.Disorder ~ Sleep.Duration + Quality.of.Sleep +
                    Physical.Activity.Level + Stress.Level +
                      BMI.Category + Daily.Steps,
                    data = data, method = "glm", family = "binomial",
                    trControl = train_control,)

# Elastic Net 模型
model_ela <- train(Sleep.Disorder ~ Blood.Pressure + Stress.Level +
                   Sleep.Duration + Occupation + Heart.Rate +
                   Physical.Activity.Level + BMI.Category +
                   Quality.of.Sleep + Gender,
                   data = data, method = "glm", family = "binomial",
                   trControl = train_control)

# 手選變數模型
model_self <- train(Sleep.Disorder ~  BMI.Category  + Blood.Pressure +
                    Physical.Activity.Level+ Stress.Level   ,
                    data = data, method = "glm", family = "binomial",
                    trControl = train_control)


# 各模型比較
summary(model_all) #std 大
```

```
Call:
NULL

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -3.002e+01  9.941e+03  -0.003   0.9976
Age                            -4.243e-01  3.608e-01  -1.176   0.2396
GenderMale                     -1.859e+01  1.240e+03  -0.015   0.9880
Occupation10                   -1.329e+01  4.030e+03  -0.003   0.9974
Occupation11                    1.295e+00  1.563e+00   0.828   0.4076
Occupation2                     1.965e+01  1.240e+03   0.016   0.9874
```

```
Occupation3                         -4.193e+00  1.420e+01  -0.295    0.7678
Occupation4                         -6.813e+00  1.422e+01  -0.479    0.6318
Occupation6                         -2.163e+00  1.720e+01  -0.126    0.8999
Occupation8                          6.424e+01  4.491e+03   0.014    0.9886
Occupation9                          6.903e+01  3.355e+07   0.000    1.0000
OccupationSalesperson               -8.765e+00  2.515e+04   0.000    0.9997
Sleep.Duration                      -6.668e+00  3.442e+00  -1.937    0.0528 .
Quality.of.Sleep6                   -2.031e+01  4.069e+03  -0.005    0.9960
Quality.of.Sleep7                    6.053e+01  6.345e+03   0.010    0.9924
Quality.of.Sleep8                    5.802e+01  6.345e+03   0.009    0.9927
Quality.of.Sleep9                    6.551e+01  9.940e+03   0.007    0.9947
`Physical.Activity.Level45~60` -2.849e+01  1.240e+03  -0.023    0.9817
`Physical.Activity.Level60~75` -3.577e+01  1.241e+03  -0.029    0.9770
`Physical.Activity.Level75~90` -2.308e+01  1.240e+03  -0.019    0.9852
Stress.Level4                        2.412e+01  7.945e+03   0.003    0.9976
Stress.Level5                        4.051e+01  8.041e+03   0.005    0.9960
Stress.Level6                        2.651e+01  7.945e+03   0.003    0.9973
Stress.Level7                        6.799e+01  9.806e+03   0.007    0.9945
Stress.Level8                        6.149e+01  9.072e+03   0.007    0.9946
BMI.CategoryOverweight               2.892e+01  1.240e+03   0.023    0.9814
Blood.Pressure                       7.228e-01  5.945e-01   1.216    0.2240
Heart.Rate                          -7.187e-01  9.532e-01  -0.754    0.4508
`Daily.Steps5001~7500`              -7.050e+01  2.481e+03  -0.028    0.9773
Daily.Steps7500up                    2.225e+00  3.224e+00   0.690    0.4901
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 507.47  on 373  degrees of freedom
Residual deviance: 142.73  on 344  degrees of freedom
AIC: 202.73

Number of Fisher Scoring iterations: 19
```

```
summary(model_step)#std 大
```

```
Call:
NULL

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                       6.750   5733.002   0.001   0.9991
Sleep.Duration                   -4.811      2.251  -2.137   0.0326 *
Quality.of.Sleep6               -20.809   1631.342  -0.013   0.9898
Quality.of.Sleep7                 9.795   2609.699   0.004   0.9970
Quality.of.Sleep8                 8.945   2609.700   0.003   0.9973
Quality.of.Sleep9                30.591   5732.985   0.005   0.9957
```

```
`Physical.Activity.Level45~60`   -4.596      2.461  -1.868   0.0618 .
`Physical.Activity.Level60~75`   -4.175      2.723  -1.533   0.1253
`Physical.Activity.Level75~90`   -1.490      1.896  -0.786   0.4319
Stress.Level4                    19.366   5104.564   0.004   0.9970
Stress.Level5                    19.497   5104.564   0.004   0.9970
Stress.Level6                    19.988   5104.564   0.004   0.9969
Stress.Level7                    49.740   5495.986   0.009   0.9928
Stress.Level8                    39.887   5495.983   0.007   0.9942
BMI.CategoryOverweight            6.483      1.543   4.203 2.63e-05 ***
`Daily.Steps5001~7500`           -8.730      4.194  -2.081   0.0374 *
Daily.Steps7500up                 1.290      1.317   0.979   0.3275
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 507.47  on 373  degrees of freedom
Residual deviance: 163.83  on 357  degrees of freedom
AIC: 197.83

Number of Fisher Scoring iterations: 18
```

```r
summary(model_self)#std 小
```

```
Call:
NULL

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -34.66527    8.09551  -4.282 1.85e-05 ***
BMI.CategoryOverweight           1.42232    0.77627   1.832 0.066915 .
Blood.Pressure                   0.25147    0.06333   3.971 7.17e-05 ***
`Physical.Activity.Level45~60`  -0.32021    0.67800  -0.472 0.636725
`Physical.Activity.Level60~75`   0.28872    0.78506   0.368 0.713042
`Physical.Activity.Level75~90`  -0.01012    0.72286  -0.014 0.988831
Stress.Level4                    1.78595    0.89246   2.001 0.045377 *
Stress.Level5                    0.00124    0.88456   0.001 0.998881
Stress.Level6                    0.05486    1.05766   0.052 0.958631
Stress.Level7                    3.78209    1.03519   3.654 0.000259 ***
Stress.Level8                    0.92343    0.84899   1.088 0.276738
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 507.47  on 373  degrees of freedom
Residual deviance: 172.05  on 363  degrees of freedom
AIC: 194.05
```

```
Number of Fisher Scoring iterations: 6
```

```
comparison <- data.frame(
  Model = c("all", "stepwise", "elastic net", "manual"),
  Accuracy = c(model_all$results[[2]],model_step$results[[2]],
               mean(model_ela$results[[2]]),model_self$results[[2]]),
  Kappa = c(model_all$results[[3]],model_step$results[[3]],
               mean(model_ela$results[[3]]),model_self$results[[3]]),
  Sensitivity = c(model_all$results[[4]],model_step$results[[4]],
               mean(model_ela$results[[4]]),model_self$results[[4]]),
  Specificity = c(model_all$results[[5]],model_step$results[[5]],
               mean(model_ela$results[[5]]),model_self$results[[5]]),
  AUC = c(model_all$results[[6]],model_step$results[[6]],
               mean(model_ela$results[[6]]),model_self$results[[6]])
)
print(comparison)
```

```
        Model  Accuracy     Kappa Sensitivity Specificity       AUC
1         all 0.9359640 0.8674801   0.9500000   0.9161290 0.9323109
2    stepwise 0.9224865 0.8404613   0.9316068   0.9096774 0.9223829
3 elastic net 0.9116396 0.8185340   0.9220930   0.8967742 0.9008286
4      manual 0.9411171 0.8784967   0.9543340   0.9225806 0.9337039
```

```
comparison_long <- pivot_longer(comparison, cols = -Model, names_to = "Metric",
                                values_to = "Value")
ggplot(comparison_long, aes(x = Metric, y = Value, fill = Model)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(
    title = "Logistic Model Performance Comparison",
    x = "Performance Metric",
    y = "Value"
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_cartesian(ylim = c(0.8, 0.97))
```

最終模型

```r
model_self$resample
```

```
  Accuracy.Accuracy Kappa.Kappa Sensitivity.Sensitivity Specificity.Specificity
1         0.9466667   0.8878924               1.0000000               0.8709677
2         0.9189189   0.8319455               0.9534884               0.8709677
3         0.9200000   0.8381295               0.8863636               0.9677419
4         0.9333333   0.8618785               0.9545455               0.9032258
5         0.9866667   0.9726377               0.9772727               1.0000000
        AUC Resample
1 0.9072581    Fold1
2 0.9148537    Fold2
3 0.9384164    Fold3
4 0.9226540    Fold4
5 0.9853372    Fold5
```

```r
print(model_self)
```

```
Generalized Linear Model

374 samples
  4 predictor
  2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 299, 300, 299, 299, 299
Resampling results:
```

```
Accuracy.Accuracy  Kappa.Kappa  Sensitivity.Sensitivity
0.9411171          0.8784967    0.954334
Specificity.Specificity  AUC
0.9225806                0.9337039
```

random forest

最終變數組合選取:Sleep.Duration +Stress.Level + BMI.Category + Blood.Pressure + Occupation

著重於健康、職業與睡眠

選擇此組預測變數，基於 Randomforest 中的 MDA 為主要參考，以 EDA 分析結果為輔。

發現其變數組合不僅符合睡眠疾病預測的目標，且符合先前 EDA 的分析結果

Sleep.Duration: 睡眠時長過長或過短都可能與睡眠障礙有關。

Stress.Level: 高壓力水平常與較短的睡眠時間和較差的睡眠質量相關，可能導致睡眠障礙。

BMI.Category: 過重或肥胖容易導致睡眠呼吸中止等問題。

Blood.Pressure: 高血壓可能與睡眠呼吸中止等睡眠障礙有關。

Occupation: 某些職業可能面臨較大的工作壓力或需要輪班工作，進而影響睡眠品質。

其中 BMI.Category, Occupation, Stress.Level 和 Quality.of.Sleep 等變數都與 Sleep.Disorder 具有高度相關性，而 Blood.Pressure 在 SHAP 圖中顯示為重要的預測變數

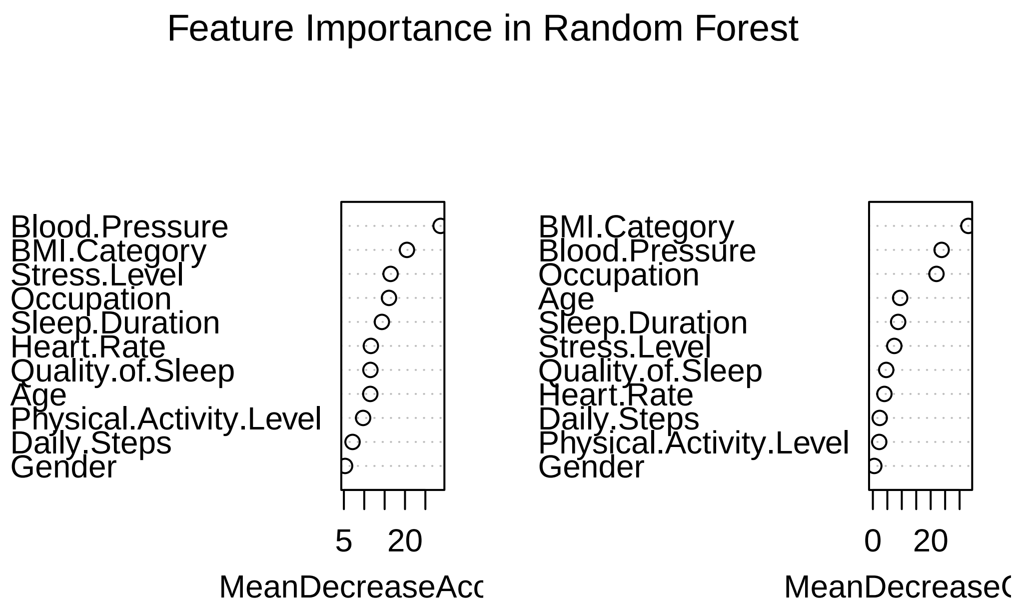
另外，從交互作用分析圖，睡眠時長和睡眠品質的關係、壓力等級和睡眠時長的關係、以及身體活動水平與 BMI 和睡眠障礙的關係，也支持這些變數作為預測變數的合理性。

rf 自選

```r
set.seed(014)
rf_model <- randomForest::randomForest(Sleep.Disorder ~ . ,
                         data = train_data,
                         ntree = 500,  # Number of trees in the forest
                         mtry = 3,  # Number of predictors considered for each split
                         importance = TRUE)  # To calculate variable importance
print(rf_model)
```

```
Call:
 randomForest(formula = Sleep.Disorder ~ ., data = train_data,      ntree = 500, mtry
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 6.33%
Confusion matrix:
    0   1 class.error
0 168   8  0.04545455
1  11 113  0.08870968
```

```r
# Plot variable importance
var_imp <- importance(rf_model)
varImpPlot(rf_model, main = "Feature Importance in Random Forest")
```




```r
rf_model <- randomForest(Sleep.Disorder ~ Sleep.Duration + Stress.Level +
                         BMI.Category + Blood.Pressure + Occupation ,
                         data = train_data,
                  ntree = 500,  # Number of trees in the forest
                  mtry = 3,     # Number of predictors considered for each split
                  importance = TRUE)  # To calculate variable importance
print(rf_model)
```

```

Call:
 randomForest(formula = Sleep.Disorder ~ Sleep.Duration + Stress.Level +      BMI.Cat
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 6.33%
Confusion matrix:
    0   1 class.error
0 168   8  0.04545455
1  11 113  0.08870968
```

```r
predicted_classes <- predict(rf_model, newdata = test_data)
predicted_probabilities <- predict(rf_model, newdata = test_data,
                                   type = "prob")[, 2]

#  Model Evaluation
# Confusion Matrix to assess performance
```

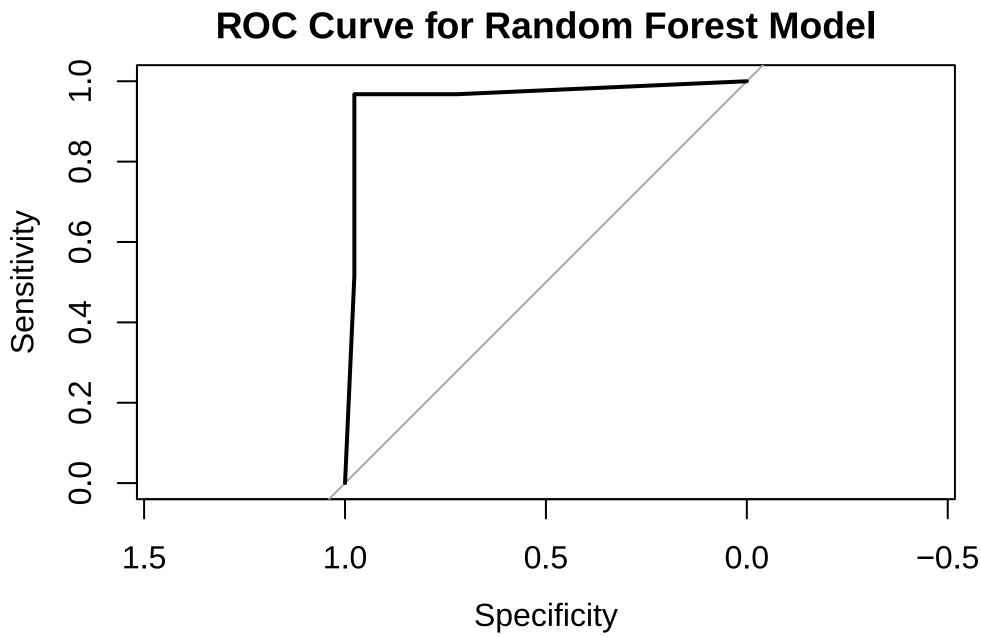
```r
confusion_matrix <- confusionMatrix(predicted_classes,
                                    as.factor(test_data$Sleep.Disorder))
print(confusion_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 41  1
         1  2 30

               Accuracy : 0.9595
                 95% CI : (0.8861, 0.9916)
    No Information Rate : 0.5811
    P-Value [Acc > NIR] : 9.21e-14

                  Kappa : 0.9171

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9535
            Specificity : 0.9677
         Pos Pred Value : 0.9762
         Neg Pred Value : 0.9375
             Prevalence : 0.5811
         Detection Rate : 0.5541
   Detection Prevalence : 0.5676
      Balanced Accuracy : 0.9606

       'Positive' Class : 0
```

```r
# ROC Curve and AUC
roc_curve <- roc(test_data$Sleep.Disorder, predicted_probabilities)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

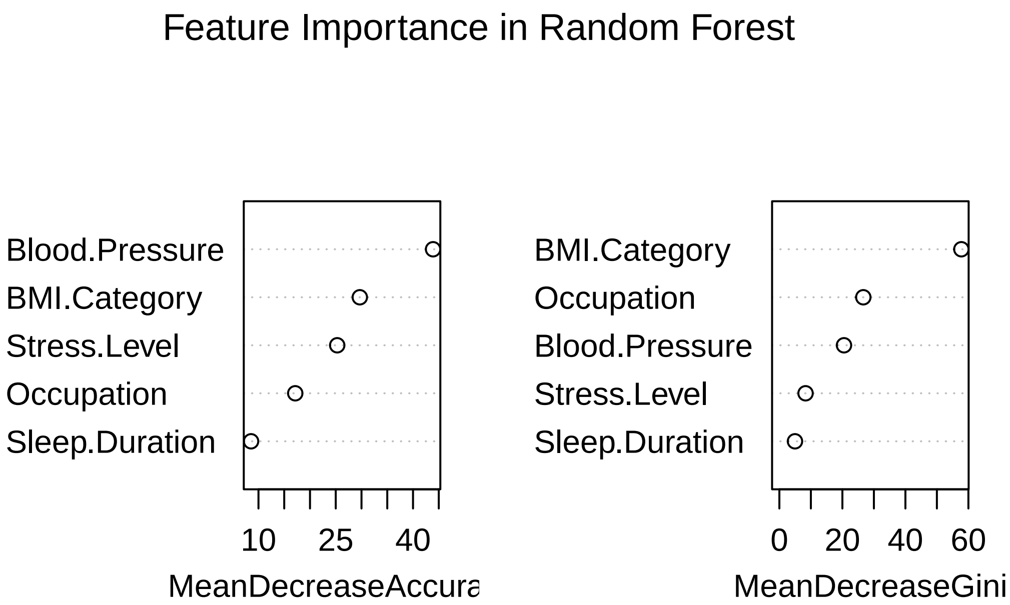
```r
plot(roc_curve, main = "ROC Curve for Random Forest Model")
```




```r
auc_value <- auc(roc_curve)
print(paste("AUC:", auc_value))
```

```
[1] "AUC: 0.962865716429107"
```

```r
# Plot variable importance
var_imp <- importance(rf_model)
varImpPlot(rf_model, main = "Feature Importance in Random Forest")
```

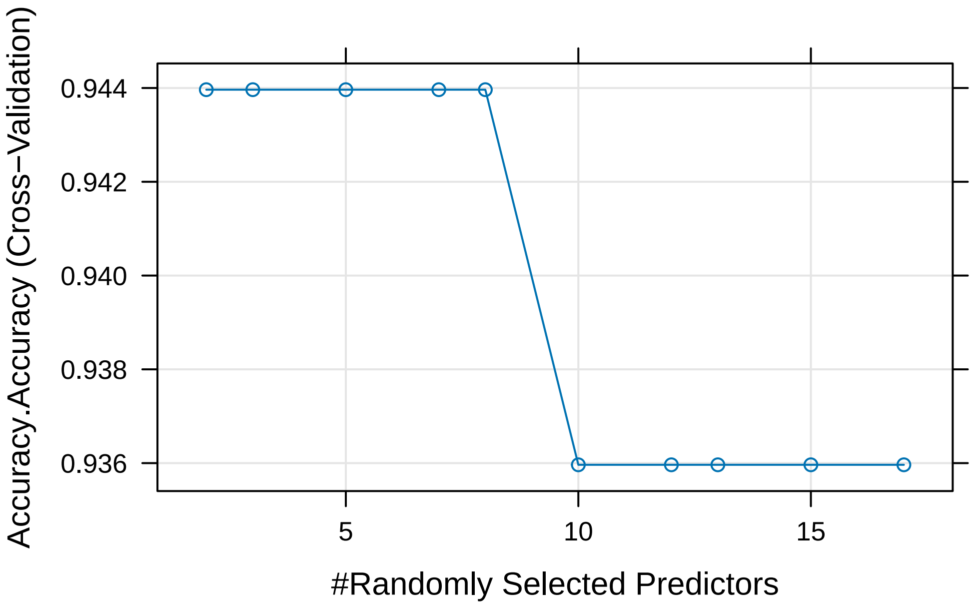


randomforest + cross validation

```
set.seed(014)
rf_model <- train(
  Sleep.Disorder ~ Sleep.Duration +Stress.Level + BMI.Category +
                   Blood.Pressure + Occupation ,
  data = data,
  method = "rf",            # 隨機森林
  trControl = train_control,
  tuneLength = 10            # 搜索最佳參數的範圍
)
rf_model$results
```

```
   mtry Accuracy.Accuracy Kappa.Kappa Sensitivity.Sensitivity
1     2          0.943964   0.8838919               0.9590909
2     3          0.943964   0.8838919               0.9590909
3     5          0.943964   0.8838919               0.9590909
4     7          0.943964   0.8838919               0.9590909
5     8          0.943964   0.8838919               0.9590909
6    10          0.935964   0.8670013               0.9545455
7    12          0.935964   0.8670013               0.9545455
8    13          0.935964   0.8670013               0.9545455
9    15          0.935964   0.8670013               0.9545455
10   17          0.935964   0.8670013               0.9545455
   Specificity.Specificity       AUC Accuracy.AccuracySD Kappa.KappaSD
1                0.9225806 0.9421145          0.02557226    0.05396334
2                0.9225806 0.9333885          0.02557226    0.05396334
3                0.9225806 0.9309708          0.02557226    0.05396334
4                0.9225806 0.9311174          0.02557226    0.05396334
5                0.9225806 0.9330270          0.02557226    0.05396334
6                0.9096774 0.9347848          0.03313101    0.07014640
7                0.9096774 0.9366995          0.03313101    0.07014640
8                0.9096774 0.9353765          0.03313101    0.07014640
9                0.9096774 0.9378691          0.03313101    0.07014640
10               0.9096774 0.9372127          0.03313101    0.07014640
   Sensitivity.SensitivitySD Specificity.SpecificitySD      AUCSD
1                 0.04065578                0.07426364 0.03463237
2                 0.04065578                0.07426364 0.03857839
3                 0.04065578                0.07426364 0.04314695
4                 0.04065578                0.07426364 0.04205897
5                 0.04065578                0.07426364 0.04015355
6                 0.03593497                0.08349793 0.04111762
7                 0.03593497                0.08349793 0.04142630
8                 0.03593497                0.08349793 0.04099335
9                 0.03593497                0.08349793 0.04158433
10                0.03593497                0.08349793 0.04196721
```

```
plot(rf_model)              # 繪製調參過程
```




```
rf_model$bestTune
```

```
  mtry
1    2
```

```
rf_model$results[1,]
```

```
  mtry Accuracy.Accuracy Kappa.Kappa Sensitivity.Sensitivity
1    2          0.943964   0.8838919               0.9590909
  Specificity.Specificity       AUC Accuracy.AccuracySD Kappa.KappaSD
1               0.9225806 0.9421145          0.02557226    0.05396334
  Sensitivity.SensitivitySD Specificity.SpecificitySD      AUCSD
1                0.04065578                0.07426364 0.03463237
```

```
rf_model$resample
```

```
  Accuracy.Accuracy Kappa.Kappa Sensitivity.Sensitivity Specificity.Specificity
1         0.9333333   0.8618785               0.9545455               0.9032258
2         0.9333333   0.8631886               0.9318182               0.9354839
3         0.9200000   0.8301887               1.0000000               0.8064516
4         0.9466667   0.8920863               0.9090909               1.0000000
5         0.9864865   0.9721176               1.0000000               0.9677419
        AUC Resample
1 0.9358504    Fold1
2 0.9530792    Fold3
3 0.8951613    Fold4
4 0.9354839    Fold5
5 0.9909977    Fold2
```

xgboost

最終變數組合: Sleep.Duration + Age + BMI.Category + Blood.Pressure + Quality.of.Sleep

顯示出 xgboost 選的變數組合著重在多面向睡眠健康評估指標，包括生理和生活型態等多個面向。

xgboost 模型的變數選擇是透過特徵重要性 (Feature Importance) 和 SHAP 圖分析來決定最終的變數組合，挑選對模型預測能力貢獻度較高的變數。

並且其變數也與 EDA 分析的結果是一致的，結論如下:

1.Sleep.Duration:

過短或過長的睡眠時長都可能增加睡眠障礙風險。

2.Age:

不同年齡層的睡眠障礙比例有所差異

3.BMI.Category:

EDA 分析顯示 BMI.Category 與 Sleep Disorder 存在顯著關聯，過重或肥胖者更容易出現睡眠障礙。

4.Blood.Pressure: 高血壓者更容易出現睡眠障礙。

5.Quality.of.Sleep: 睡眠品質差的人更容易出現睡眠障礙。

除此之外，也考慮了共線性問題，透過將 xgboost 模型選取的變數 (Blood.Pressure, Age, BMI.Category, Quality.of.Sleep, 以及 Sleep.Duration) 放入邏輯迴歸模型中，計算 GVIF 值來判斷共線性-> 顯示沒有共線性問題

一、特徵重要性:

1.xgboost 自己的

透過三種指標來衡量：

Gain: 指該變數在模型中提升預測能力的程度。

Cover: 指該變數在模型中涵蓋的樣本比例，高代表變數具有較高的區分能力。

Frequency: 指該變數在模型中被使用的次數。

```
data_dummy <- model.matrix(Sleep.Disorder ~ ., data = data)[,-1] # Remove intercept
levels(data$Sleep.Disorder) <- c(0,1
                                )
labels<-as.numeric(as.character(data$Sleep.Disorder))
# Split the data into training and testing sets
set.seed(014)
train_index <- createDataPartition(labels, p = 0.8, list = FALSE)
X_train <- data_dummy[train_index, ]
X_test <- data_dummy[-train_index, ]
y_train <- labels[train_index]
y_test <- labels[-train_index]
dtrain <- xgb.DMatrix(data = X_train, label = y_train)
dtest <- xgb.DMatrix(data = X_test, label = y_test)

# Set hyperparameters for the XGBoost model
param_list <- list(
  objective = "binary:logistic", # For binary classification
  eval_metric = "auc",           # We want to maximize AUC
```

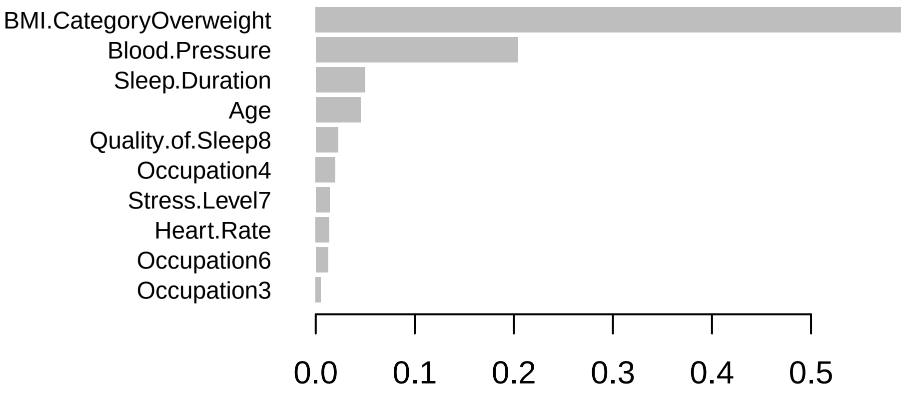
```r
  eta = 0.1,                        # Learning rate
  max_depth = 6,                    # Depth of the trees
  subsample = 0.8,                  # Row sampling ratio
  colsample_bytree = 0.8,
  verbose = 1,                      # 訓練日誌詳細程度
  watchlist = list(train = dtrain, test = dtest),
  early_stopping_rounds = 10# Feature sampling ratio
)

# Train the XGBoost model
set.seed(014)
xgb_model <- xgboost(
  data = dtrain,
  params = param_list,            # Use params to specify objective
  nrounds = 100                # Print training log
#  watchlist = list(train = dtrain, test = dtest),
 # early_stopping_rounds = 10  # Stop early if performance doesn't improve
)
```

```
[03:01:16] WARNING: src/learner.cc:767:
Parameters: { "early_stopping_rounds", "verbose", "watchlist" } are not used.

[1] train-auc:0.936038
[2] train-auc:0.947711
[3] train-auc:0.947550
[4] train-auc:0.947550
[5] train-auc:0.954066
[6] train-auc:0.964957
[7] train-auc:0.964957
[8] train-auc:0.965947
[9] train-auc:0.966177
[10]    train-auc:0.966960
[11]    train-auc:0.966499
[12]    train-auc:0.968203
[13]    train-auc:0.971910
[14]    train-auc:0.972417
[15]    train-auc:0.971450
[16]    train-auc:0.971450
[17]    train-auc:0.972463
[18]    train-auc:0.972693
[19]    train-auc:0.972279
[20]    train-auc:0.972670
[21]    train-auc:0.976515
[22]    train-auc:0.976745
[23]    train-auc:0.976400
[24]    train-auc:0.976906
[25]    train-auc:0.978495
[26]    train-auc:0.978817
```

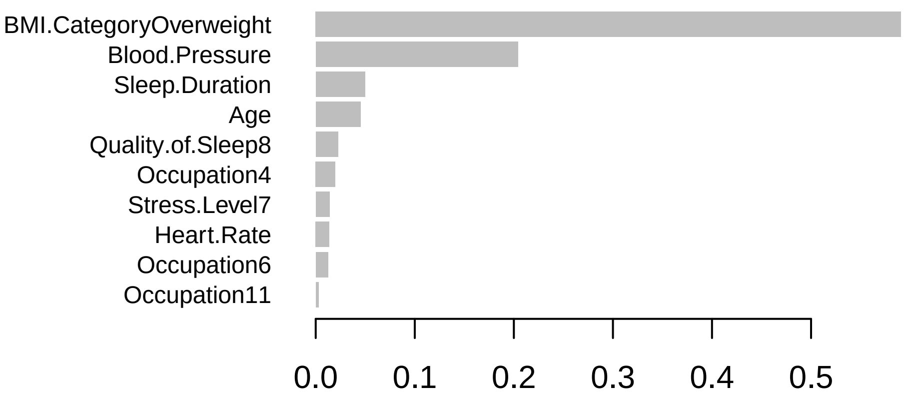
```
[27]	train-auc:0.979554
[28]	train-auc:0.981212
[29]	train-auc:0.981488
[30]	train-auc:0.981350
[31]	train-auc:0.981074
[32]	train-auc:0.982041
[33]	train-auc:0.981995
[34]	train-auc:0.981534
[35]	train-auc:0.982225
[36]	train-auc:0.982686
[37]	train-auc:0.982732
[38]	train-auc:0.982824
[39]	train-auc:0.982916
[40]	train-auc:0.983192
[41]	train-auc:0.983054
[42]	train-auc:0.983837
[43]	train-auc:0.983468
[44]	train-auc:0.983883
[45]	train-auc:0.983607
[46]	train-auc:0.983376
[47]	train-auc:0.984850
[48]	train-auc:0.985034
[49]	train-auc:0.985264
[50]	train-auc:0.985495
[51]	train-auc:0.985587
[52]	train-auc:0.986577
[53]	train-auc:0.986393
[54]	train-auc:0.986162
[55]	train-auc:0.986669
[56]	train-auc:0.987360
[57]	train-auc:0.987314
[58]	train-auc:0.987083
[59]	train-auc:0.987636
[60]	train-auc:0.988419
[61]	train-auc:0.988281
[62]	train-auc:0.987728
[63]	train-auc:0.988004
[64]	train-auc:0.987728
[65]	train-auc:0.987728
[66]	train-auc:0.987912
[67]	train-auc:0.988465
[68]	train-auc:0.988511
[69]	train-auc:0.988465
[70]	train-auc:0.989017
[71]	train-auc:0.989017
[72]	train-auc:0.989109
[73]	train-auc:0.988649
[74]	train-auc:0.989017
```

```
[75]	train-auc:0.989155
[76]	train-auc:0.989155
[77]	train-auc:0.988695
[78]	train-auc:0.988557
[79]	train-auc:0.988234
[80]	train-auc:0.988557
[81]	train-auc:0.988234
[82]	train-auc:0.988327
[83]	train-auc:0.988327
[84]	train-auc:0.988327
[85]	train-auc:0.988234
[86]	train-auc:0.988281
[87]	train-auc:0.988741
[88]	train-auc:0.988649
[89]	train-auc:0.988649
[90]	train-auc:0.988925
[91]	train-auc:0.989109
[92]	train-auc:0.988925
[93]	train-auc:0.989109
[94]	train-auc:0.988879
[95]	train-auc:0.988879
[96]	train-auc:0.988557
[97]	train-auc:0.989017
[98]	train-auc:0.989294
[99]	train-auc:0.989340
[100]	train-auc:0.990076
```

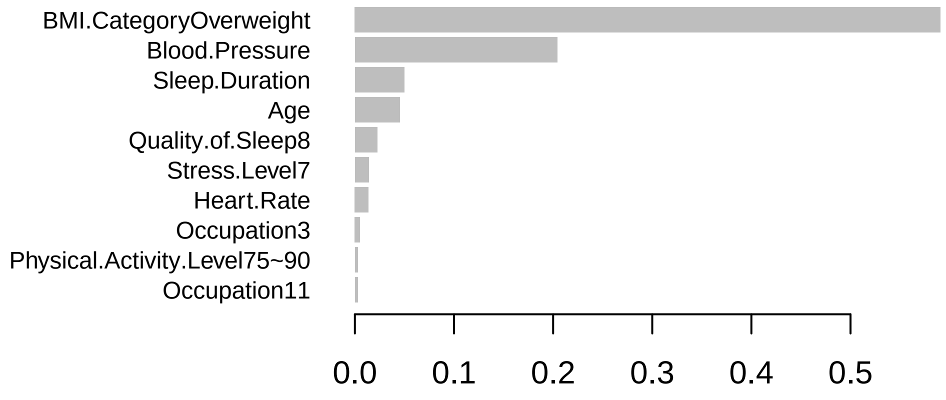
```r
importance_matrix <- xgb.importance(model = xgb_model)
# Plot feature importance
# 依據 Gain 排序繪製
importance_matrix_gain <- importance_matrix[order(-importance_matrix$Gain),][1:10,]
xgb.plot.importance(importance_matrix_gain)
```




```
# 依據 Cover 排序繪製
importance_matrix_cover <- importance_matrix[order(-importance_matrix$Cover),][1:10,]
xgb.plot.importance(importance_matrix_cover)
```




```
# 依據 Frequency 排序繪製
importance_matrix_frequency <-
  importance_matrix[order(-importance_matrix$Frequency), ][1:10, ]
xgb.plot.importance(importance_matrix_frequency)
```



2.SHAP 圖

SHAP 圖可以視覺化每個變數對個別樣本預測結果的貢獻程度，並觀察到每個變數在不同樣本上的影響方向和強度，進而更精準地選擇變數。

(1) 變數重要性：較高的 SHAP 值表示變數對模型預測的影響更大。

Blood.Pressure、BMI 是對 xgboost 模型預測 Sleep Disorder 強兩個最重要的變數。

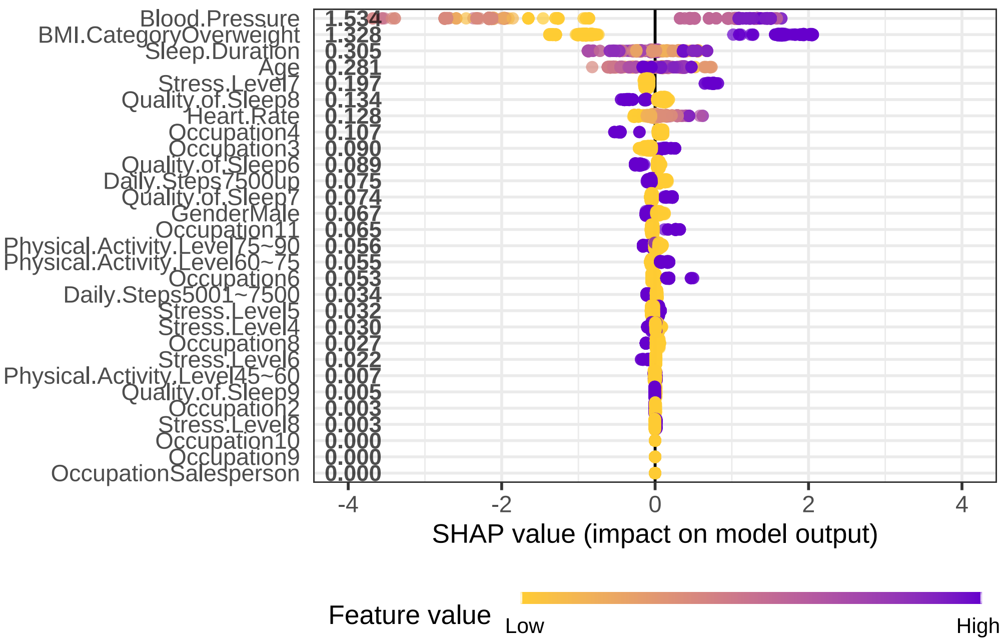
(2) 變數影響方向：SHAP 值可以顯示每個變數對預測結果是正向影響還是負向影響。正的 SHAP 值表示變數會增加預測 Sleep Disorder 的機率，而負的 SHAP 值表示變數會降低預測 Sleep Disorder 的機率。

較高的 Blood.Pressure 值 (紫色) 通常與較高的 SHAP 值相關聯，表示 Blood.Pressure 對 Sleep Disorder 的預測有正向影響。

BMI.CategoryOverweight 的 SHAP 值大部分是正值，這表示當該特徵為 “Overweight” 時，會增加模型的預測值

Age 的 SHAP 值也大多為正，顯示年齡對預測值有正面的影響，年齡越大 (紫色)，對模型的影響越大。

```
set.seed(012)
library(shapviz)
suppressPackageStartupMessages({
library("SHAPforxgboost"); library("ggplot2"); library("xgboost")
library("data.table"); library("here")
})
shap_xgboost<-shap.prep(xgb_model=xgb_model,X_train=X_train)
shap.plot.summary(shap_xgboost)
```



從特徵重要性挑變數組合

```
data_dummy <- model.matrix(Sleep.Disorder ~ Sleep.Duration +Age + BMI.Category +
                            Blood.Pressure+Quality.of.Sleep  , data = data)[, -1]
                                                           # Remove intercept
levels(data$Sleep.Disorder) <- c(0, 1)

# Split the data into training and testing sets
set.seed(014) # For reproducibility
train_index <- createDataPartition(labels, p = 0.8, list = FALSE)
X_train <- data_dummy[train_index, ]
X_test <- data_dummy[-train_index, ]
y_train <- labels[train_index]
y_test <- labels[-train_index]
dtrain <- xgb.DMatrix(data = X_train, label = y_train)
dtest <- xgb.DMatrix(data = X_test, label = y_test)

# Set hyperparameters for the XGBoost model
param_list <- list(
  objective = "binary:logistic", # For binary classification
  eval_metric = "auc",           # We want to maximize AUC
  eta = 0.1,                     # Learning rate
  max_depth = 6,                 # Depth of the trees
  subsample = 0.8,               # Row sampling ratio
  colsample_bytree = 0.8,
  verbose = 1,                   # 訓練日誌詳細程度
  watchlist = list(train = dtrain, test = dtest),
  early_stopping_rounds = 10# Feature sampling ratio
)
```

```r
# Train the XGBoost model
set.seed(014)
xgb_model <- xgboost(
  data = dtrain,
  params = param_list,          # Use params to specify objective
  nrounds = 100              # Print training log
)
```

```
[03:05:17] WARNING: src/learner.cc:767:
Parameters: { "early_stopping_rounds", "verbose", "watchlist" } are not used.

[1] train-auc:0.936038
[2] train-auc:0.958947
[3] train-auc:0.956944
[4] train-auc:0.955010
[5] train-auc:0.956599
[6] train-auc:0.961871
[7] train-auc:0.960997
[8] train-auc:0.959753
[9] train-auc:0.961871
[10]    train-auc:0.961941
[11]    train-auc:0.962585
[12]    train-auc:0.970045
[13]    train-auc:0.969631
[14]    train-auc:0.969400
[15]    train-auc:0.969400
[16]    train-auc:0.973107
[17]    train-auc:0.973292
[18]    train-auc:0.973522
[19]    train-auc:0.977989
[20]    train-auc:0.978910
[21]    train-auc:0.977022
[22]    train-auc:0.977643
[23]    train-auc:0.979117
[24]    train-auc:0.979071
[25]    train-auc:0.979669
[26]    train-auc:0.979808
[27]    train-auc:0.979071
[28]    train-auc:0.979946
[29]    train-auc:0.980429
[30]    train-auc:0.980752
[31]    train-auc:0.980890
[32]    train-auc:0.980475
[33]    train-auc:0.981212
[34]    train-auc:0.981672
[35]    train-auc:0.982317
[36]    train-auc:0.982732
[37]    train-auc:0.982962
```

```
[38]	train-auc:0.983054
[39]	train-auc:0.983238
[40]	train-auc:0.983376
[41]	train-auc:0.983745
[42]	train-auc:0.983791
[43]	train-auc:0.984205
[44]	train-auc:0.984528
[45]	train-auc:0.985057
[46]	train-auc:0.985057
[47]	train-auc:0.985425
[48]	train-auc:0.985195
[49]	train-auc:0.985333
[50]	train-auc:0.985379
[51]	train-auc:0.985241
[52]	train-auc:0.985379
[53]	train-auc:0.985425
[54]	train-auc:0.986393
[55]	train-auc:0.986254
[56]	train-auc:0.986024
[57]	train-auc:0.986162
[58]	train-auc:0.986070
[59]	train-auc:0.986807
[60]	train-auc:0.986162
[61]	train-auc:0.986393
[62]	train-auc:0.986531
[63]	train-auc:0.987175
[64]	train-auc:0.987820
[65]	train-auc:0.988004
[66]	train-auc:0.988096
[67]	train-auc:0.988004
[68]	train-auc:0.987912
[69]	train-auc:0.988050
[70]	train-auc:0.988004
[71]	train-auc:0.988004
[72]	train-auc:0.987820
[73]	train-auc:0.987544
[74]	train-auc:0.987544
[75]	train-auc:0.987820
[76]	train-auc:0.987866
[77]	train-auc:0.987958
[78]	train-auc:0.988142
[79]	train-auc:0.987866
[80]	train-auc:0.988142
[81]	train-auc:0.988234
[82]	train-auc:0.988234
[83]	train-auc:0.988188
[84]	train-auc:0.988465
[85]	train-auc:0.988465
```

```
[86]    train-auc:0.988465
[87]    train-auc:0.988465
[88]    train-auc:0.988373
[89]    train-auc:0.988511
[90]    train-auc:0.988695
[91]    train-auc:0.988511
[92]    train-auc:0.988787
[93]    train-auc:0.988833
[94]    train-auc:0.988971
[95]    train-auc:0.988971
[96]    train-auc:0.988971
[97]    train-auc:0.989063
[98]    train-auc:0.989155
[99]    train-auc:0.989155
[100]   train-auc:0.989432
```

```r
# Predict probabilities on the test set
pred_probs <- predict(xgb_model, newdata = dtest)
# Convert probabilities to binary predictions (threshold = 0.5)
predictions <- ifelse(pred_probs > 0.5, 1, 0)
# Confusion matrix
confusion_matrix <- confusionMatrix(as.factor(predictions), as.factor(y_test))
print(confusion_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 38  4
         1  3 29

               Accuracy : 0.9054
                 95% CI : (0.8148, 0.9611)
    No Information Rate : 0.5541
    P-Value [Acc > NIR] : 4.745e-11

                  Kappa : 0.808

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9268
            Specificity : 0.8788
         Pos Pred Value : 0.9048
         Neg Pred Value : 0.9062
             Prevalence : 0.5541
         Detection Rate : 0.5135
   Detection Prevalence : 0.5676
      Balanced Accuracy : 0.9028
```

```
       'Positive' Class : 0
```

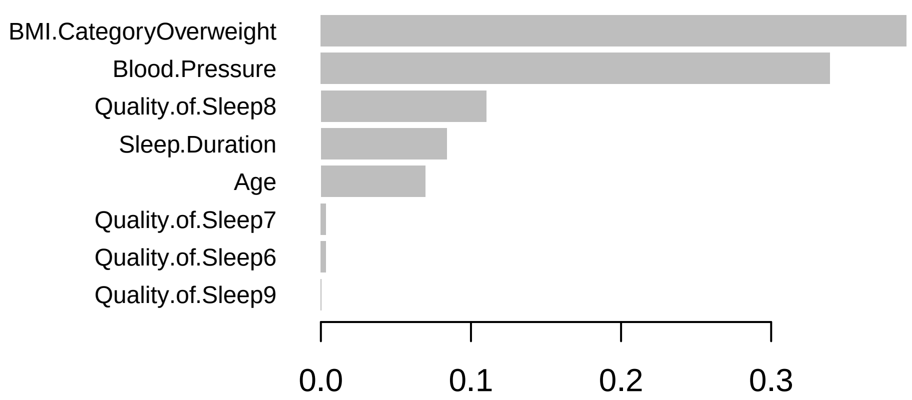
```
# Calculate AUC
auc <- roc(y_test, pred_probs)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
print(auc$auc)
```

```
Area under the curve: 0.8751
```

```
importance_matrix <- xgb.importance(model = xgb_model)
# Plot feature importance
xgb.plot.importance(importance_matrix)
```



xgboost + cross validation

```
set.seed(014)
levels(data$Sleep.Disorder) <- c("No", "Yes")
tuneGrid <- expand.grid(
  nrounds = c(50, 100, 150),
  max_depth = c(3, 6, 9),
  eta = c(0.01, 0.1, 0.3),
  gamma = c(0, 1),
  colsample_bytree = c(0.6, 0.8, 1),
  min_child_weight = c(1, 3),
  subsample = c(0.6, 0.8)
)
```

```r
xgb_model <- train(
  Sleep.Disorder ~ Sleep.Duration + Age + BMI.Category +
    Blood.Pressure + Quality.of.Sleep,
  data = data,
  method = "xgbTree",
  trControl = train_control,
  tuneGrid = tuneGrid
)
```

```r
# 查看模型結果
summary(xgb_model)
```

```
              Length Class              Mode
handle            1  xgb.Booster.handle externalptr
raw           54092  -none-             raw
niter             1  -none-             numeric
call              5  -none-             call
params            8  -none-             list
callbacks         1  -none-             list
feature_names     8  -none-             character
nfeatures         1  -none-             numeric
xNames            8  -none-             character
problemType       1  -none-             character
tuneValue         7  data.frame         list
obsLevels         2  -none-             character
param             0  -none-             list
```

```r
xgb_model$bestTune
```

```
    nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
217      50         3 0.1     0              0.6                1       0.6
```

```r
#Accuracy was used to select the optimal model using the
# largest value.
#The final values used for the model were nrounds =
# 50, max_depth = 6, eta = 0.3, gamma = 0, colsample_bytree
# = 0.6, min_child_weight = 1 and subsample = 0.6.
xgb_model$results[121,][12]
```

```
          AUC
505 0.9331873
```

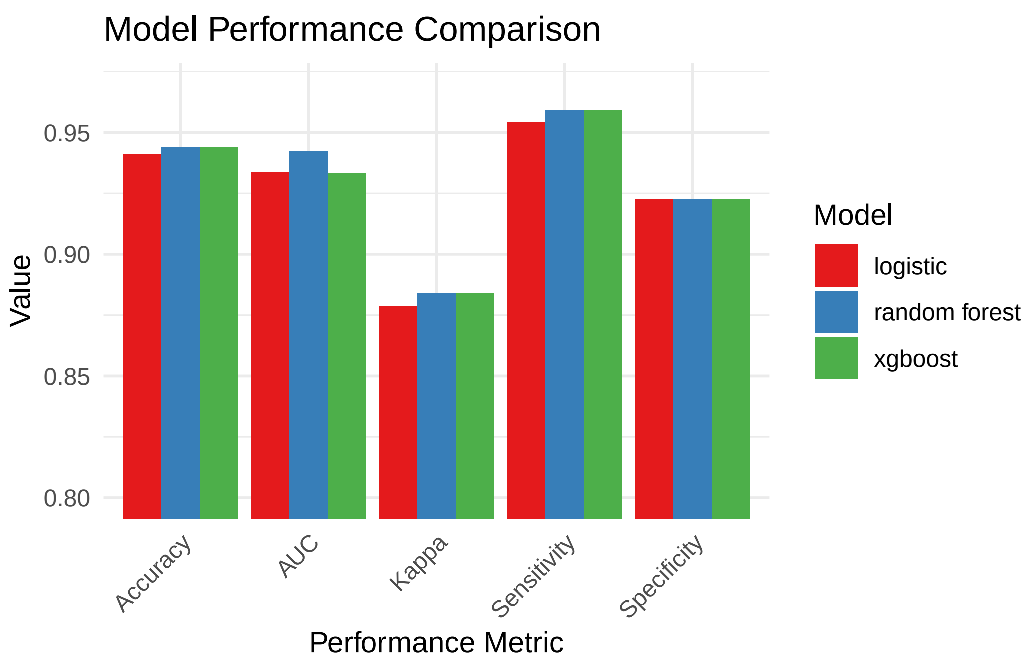
comparison three model

```r
comparison <- data.frame(
  Model = c("logistic", "random forest", "xgboost"),
  Accuracy = c(mean(model_self$results[[2]]),rf_model$results[1,][[2]],
             xgb_model$results[121,][[8]]),
  Kappa = c(mean(model_self$results[[3]]),rf_model$results[1,][[3]],
```

```
              xgb_model$results[121,][[9]]),
  Sensitivity = c(mean(model_self$results[[4]]),rf_model$results[1,][[4]],
                    xgb_model$results[121,][[10]]),
  Specificity = c(mean(model_self$results[[5]]),rf_model$results[1,][[5]],
                    xgb_model$results[121,][[11]]),
  AUC = c(mean(model_self$results[[6]]),rf_model$results[1,][[6]],
            xgb_model$results[121,][[12]])
)
print(comparison)
```

```
          Model  Accuracy     Kappa Sensitivity Specificity       AUC
1      logistic 0.9411171 0.8784967   0.9543340   0.9225806 0.9337039
2 random forest 0.9439640 0.8838919   0.9590909   0.9225806 0.9421145
3       xgboost 0.9439640 0.8838919   0.9590909   0.9225806 0.9331873
```

```
library(tidyr)
comparison_long <- pivot_longer(comparison, cols = -Model,
                                names_to = "Metric", values_to = "Value")
ggplot(comparison_long, aes(x = Metric, y = Value, fill = Model)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(
    title = "Model Performance Comparison",
    x = "Performance Metric",
    y = "Value"
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_cartesian(ylim = c(0.8, 0.97))
```



總結

1. 在職業中，以護士、銷售人員、老師有較高比例有睡眠疾病
2. 睡眠品質高、睡眠時長較長、壓力程度適中、BMI 正常、血壓正常、有運動習慣、較年輕的人明顯有較低比例有睡眠疾病
3. Logistic regression 變數組合著重於健康和生活運動習慣
   Randomforest 著重於健康、職業與睡眠 XGBoost 更全面反映可能的風險因子 (年齡、睡眠品質)
4. 此筆資料樣本數少，因此綜合模型結果、時間效率等考量下，我們認為使用傳統統計方法 (羅吉斯迴歸) 就能有不錯的成果。