Comment le temps de travail et le revenu influencent-ils le sommeil, et quelles implications pour la santé et la productivité ?

Anaïs AUGÉ & Piero PELOSI

LDD3-EM | Projet Bi-Disciplinaire | Code R complet

1. Chargement des données et transformation des variables

```r
df <- read.csv("Z_data.csv", sep = ",", header = TRUE)
```

```r
# Prétraitement et transformation des variables pour une analyse approfondie
df <- df %>%
  mutate(
    # Convertit le temps de travail hebdomadaire en heures par jour.
    # Remplace les zéros par NA pour éviter de fausser les moyennes.
    h_travail_jour = ifelse(work == 0, NA, work / 7 / 60),

    # Convertit le temps de sommeil hebdomadaire en heures par jour.
    h_sommeil_jour = sleep / 7 / 60,

    # Renomme 'educ' en 'education' pour plus de clarté.
    education = educ,

    # Renomme 'exper' en 'experience' pour plus de clarté.
    experience = exper,
    uc = 1,                                      # Unité de consommation (UC) base pour chaque adulte.
    uc = ifelse(married == 1, uc + 0.5, uc),     # UC + 0.5 si marié, représentant le conjoint.
    uc = ifelse(yngkid == 1, uc + 0.3, uc),      # UC + 0.3 pour chaque enfant de moins de 3 ans.

    # Calcule le revenu ajusté par unité de consommation.
    revenu_ajuste_uc = earnings / uc,

    # Catégorise le revenu ajusté en trois groupes: Bas, Moyen, Élevé.
    cat_revenu_ajuste_uc = cut(revenu_ajuste_uc,
                               breaks=quantile(revenu_ajuste_uc,
                                               probs=c(0, 0.45, 0.8, 1),
                                               na.rm = TRUE),
                               include.lowest=TRUE,
                               labels=c("Bas", "Moyen", "Élevé")),

    # Applique le logarithme au nombre d'heures de travail par jour.
    log_h_travail_jour = log(h_travail_jour+1),

    # Applique le logarithme au nombre d'heures de sommeil par jour.
    log_h_sommeil_jour = log(h_sommeil_jour),

    # Calcule les heures passées dans d'autres activités par jour.
    h_autres_activites_jour = rest / 7 / 60
  )

# Supprime les lignes contenant des valeurs manquantes pour nettoyer les données.
df <- na.omit(df)
```

2. Statistiques descriptives

→ Quelques moyennes utiles reprises du travail de Biddle et Hamermesh

```
# Moyenne heure de sommeil des individus intérogés
mean(df$h_sommeil_jour)
```

```
## [1] 7.738123
```

```
# Moyenne heure de sommeil en fonction du sexe : Femme v.s. Homme
df %>%
  group_by(sex) %>%
  summarise(moyenne_temps_sommeil = mean(h_sommeil_jour, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   sex    moyenne_temps_sommeil
##   <chr>                  <dbl>
## 1 Female                  7.85
## 2 Male                    7.65
```

```
# Moyenne heure de sommeil par groupe : Femme v.s. Homme
# En fonction du statut marital et de la présence d'un jeune enfant
df %>%
  mutate(sexe = sex,
         marie = married,
         jeune_enfant = yngkid) %>%
  group_by(sexe, marie, jeune_enfant) %>%
  summarise(moyenne_temps_sommeil = mean(h_sommeil_jour, na.rm = TRUE), .groups = "drop")
```

```
## # A tibble: 7 x 4
##   sexe   marie jeune_enfant moyenne_temps_sommeil
##   <chr>  <int>        <int>                 <dbl>
## 1 Female     0            0                  7.60
## 2 Female     0            1                  8.84
## 3 Female     1            0                  7.94
## 4 Female     1            1                  7.69
## 5 Male       0            0                  7.57
## 6 Male       1            0                  7.61
## 7 Male       1            1                  7.86
```

→ Vérification d'une subdivision cohérente de la population

```
# Tableau de comptage pour la catégorie de revenu
df %>%
  group_by(cat_revenu_ajuste_uc) %>%
  summarise(Count = n(),
            .groups = 'drop')
```

```
## # A tibble: 3 x 2
##   cat_revenu_ajuste_uc Count
##   <fct>                <int>
## 1 Bas                    244
## 2 Moyen                  185
## 3 Élevé                   84
```

→ Un premier modèle général

```
# Modélisation de la première régression linéaire
reg_SD1 <- lm(h_sommeil_jour ~ h_travail_jour +
                revenu_ajuste_uc +
                age +
                education,
              data = df)
```

```
# Affichage des coefficients de la régression 1, arrondis à trois décimales
# Intercept, pentes pour le travail journalier, le revenu ajusté, l'âge et le nv d'éducation
round(summary(reg_SD1)$coefficients, 3)
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.651      0.324  26.719    0.000
## h_travail_jour     -0.153      0.022  -7.100    0.000
## revenu_ajuste_uc    0.000      0.000  -1.348    0.178
## age                 0.003      0.004   0.835    0.404
## education          -0.012      0.018  -0.684    0.494
```

```
# Affichage du R² ajusté de la première régression linéaire
summary(reg_SD1)$adj.r.squared
```

```
## [1] 0.09887269
```

→ Un deuxième modèle d'intéraction, plus précis

```
# Modélisation de la régression linéaire 2
reg_SD2 <- lm(h_sommeil_jour ~ h_travail_jour +
                h_travail_jour:cat_revenu_ajuste_uc,
              data = df)
```

```
# Affichage des coefficients de la régression, arrondis à trois décimales
round(summary(reg_SD2)$coefficients, 3)
```

```
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                 8.555      0.120  71.292    0.000
## h_travail_jour                             -0.135      0.023  -5.867    0.000
## h_travail_jour:cat_revenu_ajuste_ucMoyen   -0.026      0.017  -1.574    0.116
## h_travail_jour:cat_revenu_ajuste_ucÉlevé   -0.049      0.020  -2.405    0.017
```

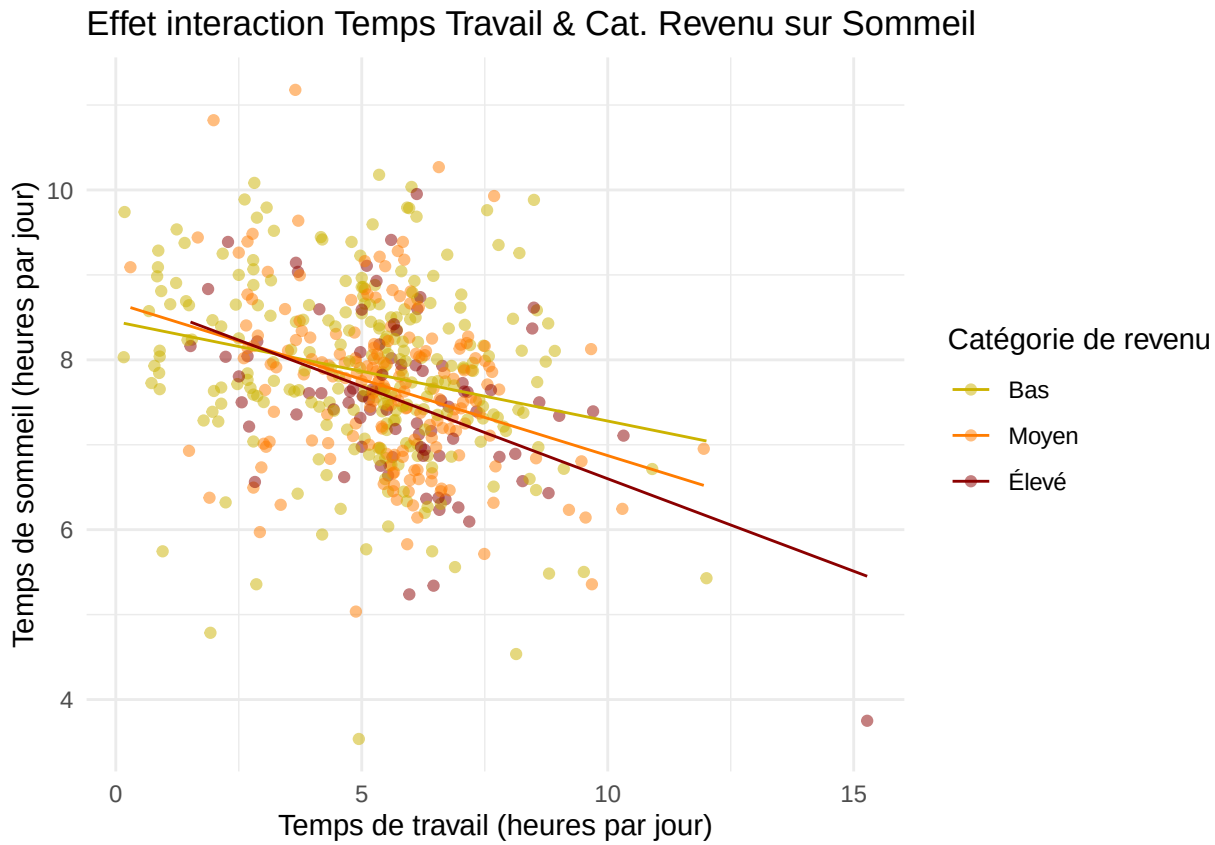
```
# Affichage du R² ajusté de la régression 2
summary(reg_SD2)$adj.r.squared
```

```
## [1] 0.1044983
```

→ Visualisation via lignes de tendance

```
# Création d'un graphique pour visualiser l'interaction entre
# le temps de travail et les catégories de revenu sur le sommeil
ggplot(df, aes(x = h_travail_jour, y = h_sommeil_jour, color = cat_revenu_ajuste_uc)) +
  geom_point(cex = 1.5, alpha = 0.5) +                # Points avec transparence et taille ajustées
  scale_color_manual(values = c("Bas" = "#CCB300",    # Couleurs personnalisées pour cat de revenu
                                "Moyen" = "#FF7C00",
                                "Élevé" = "#8B0000"),
                     name = "Catégorie de revenu") +
  labs(
    title = "Effet interaction Temps Travail & Cat. Revenu sur Sommeil",
    x = "Temps de travail (heures par jour)",                                    # Légende axe x
    y = "Temps de sommeil (heures par jour)") +                                  # Légende axe y
  theme_minimal() +
  geom_line(data = subset(df, !is.na(cat_revenu_ajuste_uc)),  # Lignes tendance par cat de revenu
            aes(group = cat_revenu_ajuste_uc),
            stat = "smooth", method = "lm", se = FALSE) +     # Lignes de tendance basées sur la
                                                              # régression linéaire sans IC
  theme(legend.position = "right")                            # Position de la légende
```

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



3. Le modèle économétrique

→ Mise en place de 6 modèles à tester

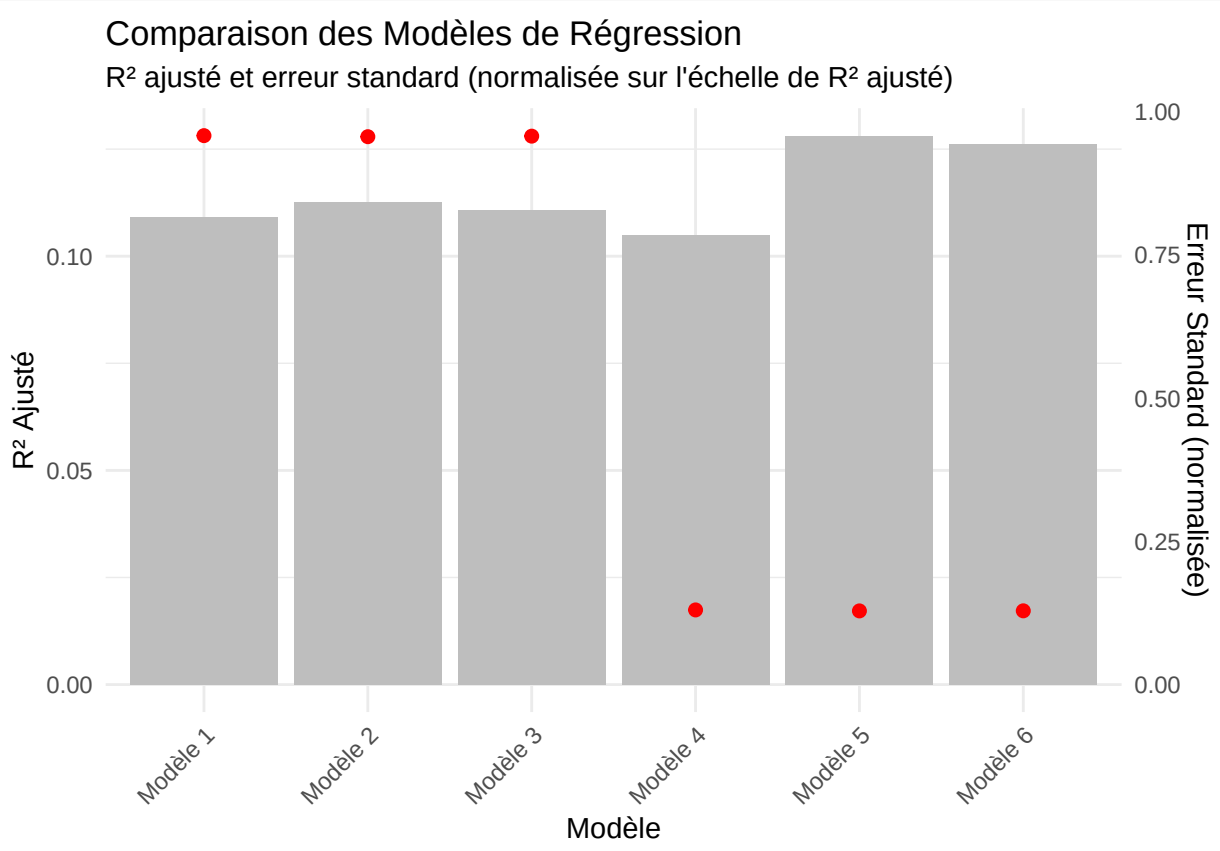
```r
# Modèle Quadratique avec Interaction
reg1 <- lm(h_sommeil_jour ~ h_travail_jour +
                            I(h_travail_jour^2) +
                            h_travail_jour:cat_revenu_ajuste_uc, data = df)
# Modèle Cubique avec Interaction
reg2 <- lm(h_sommeil_jour ~ h_travail_jour +
                            I(h_travail_jour^2) +
                            I(h_travail_jour^3) +
                            h_travail_jour:cat_revenu_ajuste_uc, data = df)
# Modèle Puissance Quatrième avec Interaction
reg3 <- lm(h_sommeil_jour ~ h_travail_jour +
                            I(h_travail_jour^2) +
                            I(h_travail_jour^3) +
                            I(h_travail_jour^4) +
                            h_travail_jour:cat_revenu_ajuste_uc, data = df)
# Modèle Log-NV avec Interaction
reg4 <- lm(log_h_sommeil_jour ~ h_travail_jour +
                                h_travail_jour:cat_revenu_ajuste_uc, data = df)
# Modèle Log-NV avec terme Cubique et Interaction
reg5 <- lm(log_h_sommeil_jour ~ h_travail_jour +
                                I(h_travail_jour^2) +
                                I(h_travail_jour^3) +
                                h_travail_jour:cat_revenu_ajuste_uc, data = df)
# Modèle Log-NV avec terme Cubique, Interaction et Autres Variables
reg6 <- lm(log_h_sommeil_jour ~ h_travail_jour +
                                I(h_travail_jour^2) +
                                I(h_travail_jour^3) +
                                h_travail_jour:cat_revenu_ajuste_uc +
                                age + educ + h_autres_activites_jour, data = df)
```

```r
# Calcul du R² ajusté et des erreurs standard pour chaque modèle
r2_lm1 = summary(reg1)$adj.r.squared ; se_lm1 <- summary(reg1)$sigma
r2_lm2 = summary(reg2)$adj.r.squared ; se_lm2 <- summary(reg2)$sigma
r2_lm3 = summary(reg3)$adj.r.squared ; se_lm3 <- summary(reg3)$sigma
r2_lm4 = summary(reg4)$adj.r.squared ; se_lm4 <- summary(reg4)$sigma
r2_lm5 = summary(reg5)$adj.r.squared ; se_lm5 <- summary(reg5)$sigma
r2_lm6 = summary(reg6)$adj.r.squared ; se_lm6 <- summary(reg6)$sigma

# Création d'un dataframe pour stocker des statistiques de différents modèles de régression
model_data <- data.frame(
  # Crée une colonne 'Model' avec des facteurs pour chaque modèle.
  # Les facteurs sont ordonnés selon l'ordre spécifié.
  Model = factor(c("Modèle 1", "Modèle 2", "Modèle 3", "Modèle 4", "Modèle 5", "Modèle 6"),
                 levels = c("Modèle 1", "Modèle 2", "Modèle 3", "Modèle 4", "Modèle 5", "Modèle 6")),

  # Crée une colonne 'Adjusted_R2' qui contient les valeurs du R² ajusté pour chaque modèle.
  Adjusted_R2 = c(r2_lm1, r2_lm2, r2_lm3, r2_lm4, r2_lm5, r2_lm6),
  # Crée une colonne 'Standard_Error' avec l'erreur standard des estimateurs pour chaque modèle.
  Standard_Error = c(se_lm1, se_lm2, se_lm3, se_lm4, se_lm5, se_lm6)
)
```

```r
# Normalisation de l'erreur standard pour correspondre à l'échelle de R² ajusté
model_data$Normalized_SE <-
  model_data$Standard_Error / max(model_data$Standard_Error) * max(model_data$Adjusted_R2)
```

→ Visualisation via histogramme

```r
# Création d'un graphique pour comparer les performances de différents modèles de régression
ggplot(model_data, aes(x = Model)) +
  # Ajoute des barres pour le R² ajusté de chaque modèle
  geom_col(aes(y = Adjusted_R2), fill = "grey") +
  # Ajoute des points pour l'erreur standard normalisée
  geom_point(aes(y = Normalized_SE), color = "red", size = 2) +
  # Configure l'échelle de l'axe Y pour afficher le R² ajusté
  # Et une seconde échelle pour l'erreur standard normalisée
  scale_y_continuous(
    name = "R² Ajusté",
    sec.axis = sec_axis(~ . / max(model_data$Adjusted_R2) * max(model_data$Standard_Error),
                        name = "Erreur Standard (normalisée)")) +
  # Ajoute des étiquettes pour le graphique
  labs(title = "Comparaison des Modèles de Régression",
       subtitle = "R² ajusté et erreur standard (normalisée sur l'échelle de R² ajusté)",
       x = "Modèle") +
  theme_minimal() +
  # Ajuste le texte de l'axe des abscisses pour une meilleure lisibilité
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")
```



→ Comparaison des 3 modèles retenus (4,5,6) via ANOVA

```r
# Comparaison des modèles de régression linéaire par ANOVA
anova(reg4, reg5, reg6)
```

```
## Analysis of Variance Table
##
## Model 1: log_h_sommeil_jour ~ h_travail_jour + h_travail_jour:cat_revenu_ajuste_uc
## Model 2: log_h_sommeil_jour ~ h_travail_jour + I(h_travail_jour^2) + I(h_travail_jour^3) +
##     h_travail_jour:cat_revenu_ajuste_uc
## Model 3: log_h_sommeil_jour ~ h_travail_jour + I(h_travail_jour^2) + I(h_travail_jour^3) +
##     h_travail_jour:cat_revenu_ajuste_uc + age + educ + h_autres_activites_jour
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    509 8.6351
## 2    507 8.3772  2  0.257930 7.7868 0.0004671 ***
## 3    504 8.3473  3  0.029868 0.6011 0.6145066
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

→ Focus sur le modèle 5

```r
# Affichage des coefficients de la régression, arrondis à trois décimales
round(summary(reg5)$coefficients, 3)
```

```
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              2.155      0.036  59.765    0.000
## h_travail_jour                          -0.040      0.020  -1.964    0.050
## I(h_travail_jour^2)                      0.006      0.003   1.809    0.071
## I(h_travail_jour^3)                      0.000      0.000  -2.563    0.011
## h_travail_jour:cat_revenu_ajuste_ucMoyen -0.003      0.002  -1.345    0.179
## h_travail_jour:cat_revenu_ajuste_ucÉlevé -0.005      0.003  -1.760    0.079
```

```r
# Affichage du R² ajusté de la régression
summary(reg5)$adj.r.squared
```

```
## [1] 0.1281479
```

```r
# Affichage de la F-statistique de la régression
summary(reg5)$fstatistic
```

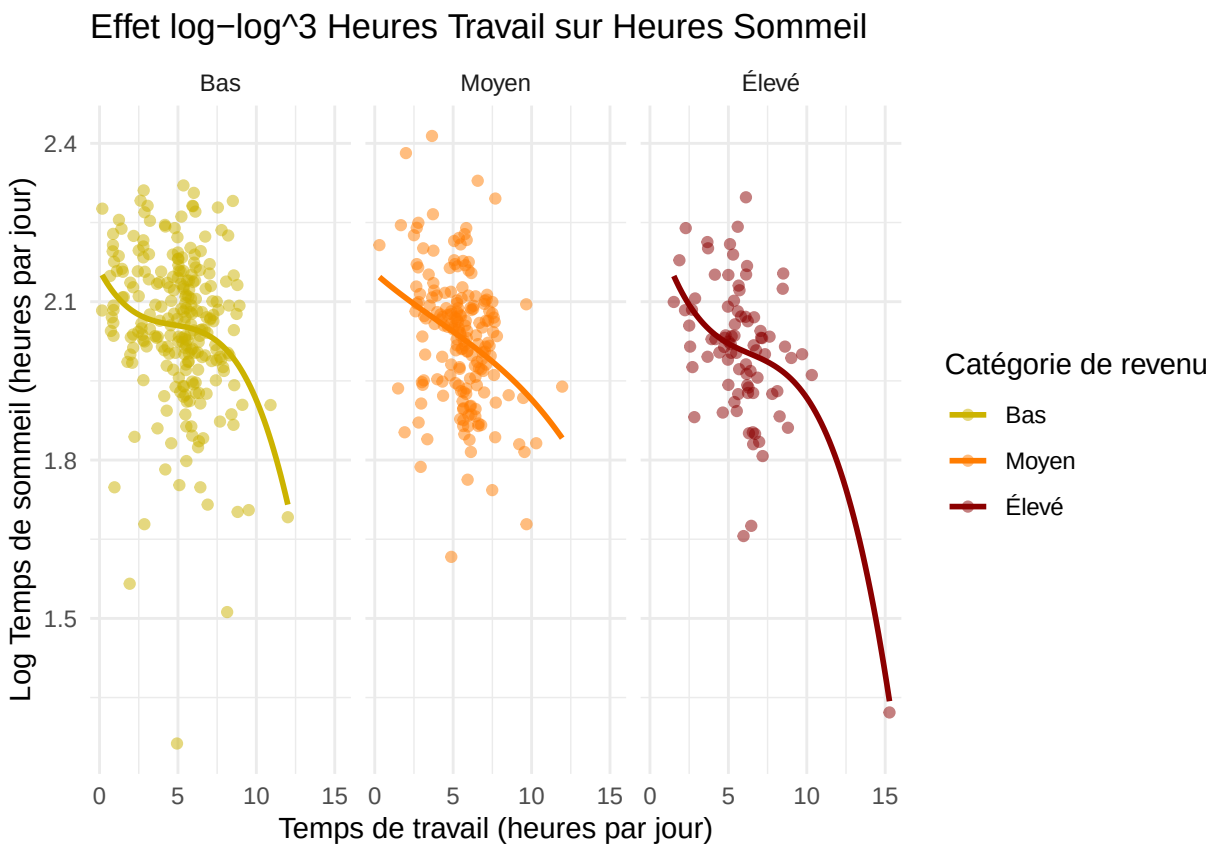
```
##     value     numdf     dendf
##  16.05111   5.00000 507.00000
```

```r
pf(summary(reg5)$fstatistic[1], summary(reg5)$fstatistic[2], summary(reg5)$fstatistic[3], lower.tail = I
```

```
##        value
## 1.063926e-14
```

→ Visualisation modèle 5

```r
# Construction du graphique avec ggplot2
ggplot(df, aes(x = h_travail_jour, y = log_h_sommeil_jour, color = cat_revenu_ajuste_uc)) +
  # Ajout de points pour représenter les données individuelles
  # Avec une taille et une transparence ajustées
  geom_point(cex = 1.5, alpha = 0.5) +
  # Personnalisation des couleurs des points selon la catégorie de revenu
  scale_color_manual(values = c("Bas" = "#CCB300",
                                "Moyen" = "#FF7C00",
                                "Élevé" = "#8B0000"),
                     name = "Catégorie de revenu") +
  # Ajout d'une ligne de tendance polynomiale de degré 3 pour chaque catégorie de revenu
  # Sans l'intervalle de confiance
  geom_smooth(method = "lm", formula = y ~ poly(x, 3), se = FALSE) +
  # Séparation du graphique en facettes, une pour chaque catégorie de revenu
  facet_wrap(~ cat_revenu_ajuste_uc) +
  # Ajout de titres et étiquettes aux axes pour une meilleure compréhension
  labs(
    title = "Effet log-log^3 Heures Travail sur Heures Sommeil",
    x = "Temps de travail (heures par jour)",
    y = "Log Temps de sommeil (heures par jour)") +

  # Application d'un thème minimaliste pour une présentation épurée
  theme_minimal()
```



4. Statistiques explicatives

→ Calcul de la moyenne de temps libre par catégorie de revenu

```
# Mise à jour du dataframe avec une nouvelle colonne pour les heures d'autres activités par jour
df <- df %>%
  mutate(
    # Convertit les minutes de repos hebdomadaires en heures par jour.
    # Remplace les valeurs de 0 par NA pour éviter les biais dans les moyennes.
    h_autres_activites_jour = ifelse(rest == 0, NA, rest / 7 / 60)
  )
# Supprime toutes les lignes contenant des valeurs NA dans la dataframe pour nettoyer les données
df <- na.omit(df)

# Groupe les données par catégorie de revenu ajusté.
# Calcule la moyenne des heures passées dans d'autres activités.
df %>%
  group_by(cat_revenu_ajuste_uc) %>%
  summarize(
    # Calcule la moyenne des heures d'autres activités par jour, en ignorant les NA
    Moyenne_heures_activites = mean(h_autres_activites_jour, na.rm = TRUE),
    # Supprime le regroupement après la synthèse pour éviter la création de sous-groupes
    .groups = 'drop'
  )
```

```
## # A tibble: 3 x 2
##   cat_revenu_ajuste_uc Moyenne_heures_activites
##   <fct>                                  <dbl>
## 1 Bas                                    0.319
## 2 Moyen                                  0.214
## 3 Élevé                                  0.425
```

5. Statistiques explicatives

→ Données peu utilisables pour l'évaluation des impacts sur la santé

```
# Moyenne heure de sommeil en fonction de la santé
df %>%
  group_by(gdhealth) %>%
  summarise(moyenne_temps_sommeil = mean(h_sommeil_jour, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   gdhealth moyenne_temps_sommeil
##      <int>                 <dbl>
## 1        0                  7.81
## 2        1                  7.78
```