# Chapitre 2 - Codes de réplication

## Table of contents



```r
# Dossier avec CSV
csv_dir <- "/Users/.../Chap2/SAGE_csv"

# Liste de tous les .csv (full.names = TRUE pour avoir le chemin complet)
csv_files <- list.files(csv_dir,
                        pattern    = "\\.csv$",
```

```
                        full.names = TRUE)

# Construire des noms d'objets (on enlève ".csv")
obj_names <- tools::file_path_sans_ext(basename(csv_files))

# Lire tous les fichiers dans une liste nommée
dfs_list <- setNames(
  lapply(csv_files, read.csv, stringsAsFactors = FALSE),
  obj_names
)

# Copier chacun des éléments de la liste dans l'environnement global
list2env(dfs_list, envir = .GlobalEnv)
```

# 1 Tendances transnationales des EFPV

```
EMP_TEMP_SEX_EDU_EC2_NB_A <-
 ↪  readRDS("/Users/.../Chap2/EMP_TEMP_SEX_EDU_EC2_NB_A.rds")
```

## 1.1 Panorama de la distribution brute des données

La section qui suit permettra de dresser un premier portrait de la variable "value" (effectifs d'emplois) et des principales variables catégorielles (vert/non-vert, sexe, niveau d'éducation, groupe de revenu). L'idée est à la fois de produire des statistiques sommaires et des graphes pour visualiser la forme des distributions.

```
library(dplyr)
library(ggplot2)

# Statistiques descriptives de "value"
EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  summarize(
    n        = n(),
    mean     = mean(value,  na.rm = TRUE),
    median   = median(value,na.rm = TRUE),
    sd       = sd(value,    na.rm = TRUE),
    min      = min(value,   na.rm = TRUE),
    q1       = quantile(value, 0.25, na.rm = TRUE),
```

```r
    q3       = quantile(value, 0.75, na.rm = TRUE),
    max      = max(value,   na.rm = TRUE)
  )

# Histogramme et densité de value
ggplot(EMP_TEMP_SEX_EDU_EC2_NB_A, aes(x = log10(value + 1))) +
  geom_histogram(binwidth = 0.1, fill = "steelblue", color = "white") +
  labs(
    x     = expression(log[10](value + 1)),
    y     = "Effectif",
    title = "Distribution (log10) de la taille des secteurs"
  ) +
  theme_minimal()

# Répartition de la variable binaire "EU_ISIC4_2dg"
EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  count(EU_ISIC4_2dg) %>%
  mutate(pct = n / sum(n) * 100)

ggplot(EMP_TEMP_SEX_EDU_EC2_NB_A, aes(x = factor(EU_ISIC4_2dg),
                                       fill = factor(EU_ISIC4_2dg))) +
  geom_bar() +
  scale_fill_manual(
    values = c("1" = "#d73027", "2" = "#1a9850", "3" = "#7570b3"),
    labels = c("1 : Non-vert", "2 : Vert", "3 : Total/X")
  ) +
  labs(
    x     = "Catégorie EU_ISIC4_2dg",
    y     = "Nombre d'observations",
    fill  = "",
    title = "Répartition des secteurs par potentiel de verdissement"
  ) +
  theme_minimal()

# Découpages par sexe, éducation et groupe de revenu
EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  count(sex) %>%
  mutate(pct = n / sum(n) * 100)

EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  count(edu) %>%
  mutate(pct = n / sum(n) * 100)
```

```r
EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  count(income_group) %>%
  mutate(pct = n / sum(n) * 100)

labels_list <- list(
  sex = c("1 = Total", "2 = Male", "3 = Female", "4 = Other"),
  edu = c(
    "1 = Less than basic", "2 = Basic", "3 = Intermediate",
    "4 = Advanced",        "5 = Total", "6 = Not classified"
  ),
  income_group = c(
    "1 = High income", "2 = Low income",
    "3 = Lower middle", "4 = Upper middle"
  )
)

for(var in c("sex", "edu", "income_group")) {
  p <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
    mutate(!!var := factor(.data[[var]])) %>%
    ggplot(aes_string(x = var, y = "log10(value + 1)")) +
    geom_boxplot(fill = "lightblue", outlier.size = 0.5) +
    scale_x_discrete(
      name   = var,
      labels = labels_list[[var]]
    ) +
    labs(
      y     = expression(log[10](value + 1)),
      title = paste("Boxplot (log10) de la taille des secteurs par", var)
    ) +
    theme_minimal() +
    theme(
      plot.title    = element_text(hjust = 0.5),
      axis.text.x   = element_text(angle = 45, hjust = 1)
    )
  print(p)
}

# Partie "Proportion d'emplois verts" - première approche
df2 <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  group_by(iso3c, year, sex, edu, income_group, isic_code_2dg) %>%
  summarise(
    total_jobs = sum(value, na.rm = TRUE),
    green_jobs = sum(value[EU_ISIC4_2dg == 2], na.rm = TRUE),
```

```r
    prop_green = green_jobs / total_jobs * 100,
    .groups    = "drop"
  )

# Statistiques sommaires de prop_green
df2 %>%
  summarize(
    mean = mean(prop_green, na.rm = TRUE),
    sd   = sd(prop_green,   na.rm = TRUE),
    min  = min(prop_green,  na.rm = TRUE),
    max  = max(prop_green,  na.rm = TRUE)
  )

# Histogramme de prop_green
ggplot(df2, aes(x = prop_green)) +
  geom_histogram(binwidth = 2, fill = "forestgreen", color = "white") +
  labs(
    x     = "% emplois verts",
    y     = "Effectif",
    title = "Distribution de la part d'emplois verts"
  ) +
  theme_minimal()
```

## 1.2 Tendances temporelles du verdissement par groupe de revenu

On va visualiser, année après année, l'évolution de la part moyenne d'EFPV (prop) dans chaque income_group.

```r
library(dplyr)
library(ggplot2)

# Construire df2
df2 <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  group_by(iso3c, year, sex, edu, income_group, isic_code_2dg) %>%
  summarise(
    total_jobs = sum(value, na.rm = TRUE),
    green_jobs = sum(value[EU_ISIC4_2dg == 2], na.rm = TRUE),
    prop_green = green_jobs / total_jobs * 100,
    .groups    = "drop"
  )
```

```r
# Construit df_trends avec IC95
df_trends <- df2 %>%
  group_by(income_group, year) %>%
  summarise(
    n         = n(),
    mean_prop = mean(prop_green, na.rm = TRUE),
    sd_prop   = sd(prop_green, na.rm = TRUE),
    .groups   = "drop"
  ) %>%
  mutate(
    se       = sd_prop / sqrt(n),
    ci_lower = mean_prop - qt(0.975, df = n - 1) * se,
    ci_upper = mean_prop + qt(0.975, df = n - 1) * se,

    income_group = factor(
      income_group,
      levels = c(1, 4, 3, 2),
      labels = c("Élevé",
                 "Moyenne supérieure",
                 "Moyenne inférieure",
                 "Plus faible")
    )
  )

# On trace avec ribbon pour l'IC
p_trends <- ggplot(df_trends,
                   aes(x = year,
                       y = mean_prop,
                       color = income_group,
                       fill  = income_group)) +
  geom_line(size = 1) +
  geom_ribbon(aes(ymin = ci_lower,
                  ymax = ci_upper),
              alpha = 0.2, color = NA) +
  scale_color_brewer(
    name    = "Groupe de revenu",
    palette = "Dark2"
  ) +
  scale_fill_brewer(
    name    = "Groupe de revenu",
    palette = "Dark2"
  ) +
  scale_y_continuous(
```

```
    name   = "% EFPV",
    limits = c(0, 100)
  ) +
  labs(
    title = "Évolution de la part moyenne des EFPV, par groupe de revenu
     ↪  (avec IC à 95 %)",
    x     = "Année"
  ) +
  theme_minimal(base_size = 14)

# Résultat
print(p_trends)
```

## 1.3 Tests statistiques robustes entre groupes de revenu

```
library(dplyr)
library(car)        # pour leveneTest
library(ggplot2)
library(knitr)

# Calcul de prop_green et prop_non_green par pays-année-groupe de revenu
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  group_by(iso3c, year, income_group) %>%
  summarise(
    total_jobs     = sum(value,                                na.rm = TRUE),
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0), na.rm = TRUE),
    prop_green     = 100 * green_jobs     / total_jobs,
    prop_non_green = 100 * non_green_jobs / total_jobs,
    .groups        = "drop"
  )

# Moyenne par pays (une seule valeur par iso3c)
df_country_means <- df_country_year %>%
  group_by(iso3c, income_group) %>%
  summarise(
    mean_prop_green     = mean(prop_green,     na.rm = TRUE),
    mean_prop_non_green = mean(prop_non_green, na.rm = TRUE),
    .groups             = "drop"
```

```
  )

# Fonction helper pour lancer les tests
run_tests <- function(var) {
  cat("=== Analyse pour", var, "===\n")
  mod <- lm(reformulate("factor(income_group)", response = var),
            data = df_country_means)
  sw <- shapiro.test(residuals(mod))
  cat("Shapiro-Wilk W =", round(sw$statistic, 3),
      " p =", format.pval(sw$p.value), "\n")
  lv <- leveneTest(as.formula(paste(var, "~ factor(income_group)")),
                   data = df_country_means)
  print(lv)
  if (sw$p.value > 0.05 && lv$`Pr(>F)`[1] > 0.05) {
    cat("→ Conditions ok : ANOVA classique + TukeyHSD\n")
    aov_res <- aov(as.formula(paste(var, "~ factor(income_group)")),
                   data = df_country_means)
    print(summary(aov_res))
    print(TukeyHSD(aov_res))
  } else {
    cat("→ Conditions non remplies : Welch ANOVA\n")
    print(oneway.test(as.formula(paste(var, "~ factor(income_group)")),
                      data      = df_country_means,
                      var.equal = FALSE))
  }
  cat("\n")
}

# Lance les tests
run_tests("mean_prop_green")
run_tests("mean_prop_non_green")

# Tableau récapitulatif par groupe de revenu
df_group_summary <- df_country_means %>%
  group_by(income_group) %>%
  summarise(
    n_countries        = n(),
    mean_green_avg     = mean(mean_prop_green,     na.rm = TRUE),
    sd_green           = sd(  mean_prop_green,     na.rm = TRUE),
    mean_non_green_avg = mean(mean_prop_non_green, na.rm = TRUE),
    sd_non_green       = sd(  mean_prop_non_green, na.rm = TRUE),
    median_green       = median(mean_prop_green,     na.rm = TRUE),
    q1_green           = quantile(mean_prop_green,   0.25, na.rm = TRUE),
```

```
    q3_green           = quantile(mean_prop_green,   0.75, na.rm = TRUE),
    median_non_green   = median(mean_prop_non_green, na.rm = TRUE),
    q1_non_green       = quantile(mean_prop_non_green, 0.25, na.rm = TRUE),
    q3_non_green       = quantile(mean_prop_non_green, 0.75, na.rm = TRUE),
    .groups            = "drop"
  ) %>%
  # transforme income_group en facteur ordonné décroissant
  mutate(
    income_group = factor(
      income_group,
      levels = c(1, 4, 3, 2),
      labels = c(
        "Revenu élevé",
        "Revenu moyen supérieur",
        "Revenu moyen inférieur",
        "Revenu faible"
      )
    )
  ) %>%
  arrange(income_group)

# Affichage du tableau avec kable
df_group_summary %>%
  rename(
    Groupe              = income_group,
    `Nb pays`           = n_countries,
    `Moy % verts`       = mean_green_avg,
    `SD % verts`        = sd_green,
    `Moy % non-verts`   = mean_non_green_avg,
    `SD % non-verts`    = sd_non_green,
    `Med % verts`       = median_green,
    `Med % non-verts`   = median_non_green,
    `Q1 % verts`        = q1_green,
    `Q1 % non-verts`    = q1_non_green,
    `Q3 % verts`        = q3_green,
    `Q3 % non-verts`    = q3_non_green
  ) %>%
  kable(digits = 2)
```

```
library(dplyr)
library(car)
library(ggplot2)
```

```
library(knitr)

# Calcul de prop_green et prop_non_green par pays-année-région
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%                              ## on se limite
   ↪  aux totaux agrégés (sex=1, edu=5)
  mutate(
    region = factor(region,
                    levels = 1:5,
                    labels = c("Africa", "Americas", "Asia", "Europe",
                     ↪  "Oceania"))
  ) %>%                                                       ## on renomme
 ↪  les modalités de region
  group_by(iso3c, year, region) %>%                          ## une ligne par
   ↪  pays, année et région
  summarise(
    total_jobs     = sum(value,                              na.rm = TRUE),
     ↪  ## total des emplois
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
     ↪  ## emplois « verts »
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0), na.rm = TRUE),
     ↪  ## emplois « non-verts »
    prop_green     = 100 * green_jobs     / total_jobs,                    ##
     ↪  % verts
    prop_non_green = 100 * non_green_jobs / total_jobs,                    ##
     ↪  % non-verts
    .groups        = "drop"
  )

# Moyenne par pays (une seule valeur par pays et par région)
df_country_means <- df_country_year %>%
  group_by(iso3c, region) %>%                                ## on regroupe
   ↪  par pays et région
  summarise(
    mean_prop_green     = mean(prop_green,     na.rm = TRUE), ## moyenne pays
     ↪  de % verts
    mean_prop_non_green = mean(prop_non_green, na.rm = TRUE), ## moyenne pays
     ↪  de % non-verts
    .groups             = "drop"
  )

# Fonction helper pour lancer les tests sur une variable donnée
```

```r
run_tests_region <- function(var) {
  cat("=== Analyse pour", var, "par region ===\n")

  # a) Ajustement d'un modèle linéaire sur region
  mod <- lm(
    reformulate("factor(region)", response = var),
    data = df_country_means
  )
  ## modélise la moyenne pays en fonction de la région

  # b) Shapiro-Wilk (normalité des résidus)
  sw <- shapiro.test(residuals(mod))
  cat("Shapiro-Wilk W =", round(sw$statistic, 3),
      " p =", format.pval(sw$p.value), "\n")
  ## vérifie la normalité des résidus

  # c) Levene (homogénéité des variances)
  lv <- leveneTest(
    as.formula(paste(var, "~ factor(region)")),
    data = df_country_means
  )
  print(lv)
  ## vérifie l'égalité des variances entre régions

  # d) Choix du test statistique
  if (sw$p.value > 0.05 && lv$`Pr(>F)`[1] > 0.05) {
    cat(" Conditions ok : ANOVA classique + TukeyHSD\n")
    aov_res <- aov(
      as.formula(paste(var, "~ factor(region)")),
      data = df_country_means
    )
    print(summary(aov_res))
    print(TukeyHSD(aov_res))
    ## si normalité et variances homogènes, ANOVA + Tukey
  } else {
    cat("→ Conditions non remplies : Welch ANOVA\n")
    print(oneway.test(
      as.formula(paste(var, "~ factor(region)")),
      data      = df_country_means,
      var.equal = FALSE
    ))
    ## sinon, test de Welch robuste
  }
```

```r
  cat("\n")
}

# Lancement des tests pour les deux indicateurs
run_tests_region("mean_prop_green")     ## tests pour % moyens d'emplois
↪  verts
run_tests_region("mean_prop_non_green") ## tests pour % moyens d'emplois
↪  non-verts

# Tableau récapitulatif par région
df_region_summary <- df_country_means %>%
  group_by(region) %>%                                       ## synthèse
  ↪  finale par région
  summarise(
    n_countries          = n(),
    mean_prop_green      = mean(mean_prop_green,     na.rm = TRUE),
    sd_prop_green        = sd(  mean_prop_green,     na.rm = TRUE),
    mean_prop_non_green  = mean(mean_prop_non_green, na.rm = TRUE),
    sd_prop_non_green    = sd(  mean_prop_non_green, na.rm = TRUE),
    median_prop_green    = median(mean_prop_green,     na.rm = TRUE),
    median_prop_non_green= median(mean_prop_non_green, na.rm = TRUE),
    q1_prop_green        = quantile(mean_prop_green,     0.25, na.rm = TRUE),
    q1_prop_non_green    = quantile(mean_prop_non_green, 0.25, na.rm = TRUE),
    q3_prop_green        = quantile(mean_prop_green,     0.75, na.rm = TRUE),
    q3_prop_non_green    = quantile(mean_prop_non_green, 0.75, na.rm = TRUE),
    .groups              = "drop"
  )

# Affichage du tableau
kable(
  df_region_summary,
  col.names = c(
    "Region", "Nb pays",
    "Moy % verts", "SD % verts",
    "Moy % non-verts", "SD % non-verts",
    "Med % verts", "Med % non-verts",
    "Q1 % verts", "Q1 % non-verts",
    "Q3 % verts", "Q3 % non-verts"
  ),
  digits = 2
)
```

```r
library(dplyr)
library(car)
library(ggplot2)
library(knitr)

# Calcul de prop_green et prop_non_green par pays-année-région
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  mutate(
    region = factor(region,
                    levels = 1:5,
                    labels = c("Africa", "Americas", "Asia", "Europe",
                     ↪  "Oceania"))
  ) %>%
  group_by(iso3c, year, region) %>%
  summarise(
    total_jobs     = sum(value, na.rm = TRUE),
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0), na.rm = TRUE),
    prop_green     = 100 * green_jobs     / total_jobs,
    prop_non_green = 100 * non_green_jobs / total_jobs,
    .groups        = "drop"
  )

# Moyenne par pays (une seule valeur par pays et région)
df_country_means <- df_country_year %>%
  group_by(iso3c, region) %>%
  summarise(
    mean_prop_green     = mean(prop_green,     na.rm = TRUE),
    mean_prop_non_green = mean(prop_non_green, na.rm = TRUE),
    .groups             = "drop"
  )

# Tests par région
run_tests_region <- function(var) {
  cat("=== Analyse pour", var, "par région ===\n")
  mod <- lm(reformulate("factor(region)", response = var),
            data = df_country_means)
  sw <- shapiro.test(residuals(mod))
  cat("Shapiro-Wilk W =", round(sw$statistic, 3),
      " | p-value =", format.pval(sw$p.value), "\n")
  lv <- leveneTest(as.formula(paste(var, "~ factor(region)")),
                   data = df_country_means)
```

```r
  print(lv)
  if (sw$p.value > 0.05 && lv$`Pr(>F)`[1] > 0.05) {
    cat("→ ANOVA classique + TukeyHSD\n")
    aov_res <- aov(as.formula(paste(var, "~ factor(region)")),
                   data = df_country_means)
    print(summary(aov_res))
    print(TukeyHSD(aov_res))
  } else {
    cat("→ Welch ANOVA (variances inégales)\n")
    print(oneway.test(as.formula(paste(var, "~ factor(region)")),
                      data      = df_country_means,
                      var.equal = FALSE))
  }
  cat("\n")
}

run_tests_region("mean_prop_green")
run_tests_region("mean_prop_non_green")

# Tableau récapitulatif par région
df_region_summary <- df_country_means %>%
  group_by(region) %>%
  summarise(
    n_countries       = n(),
    green_avg         = mean(mean_prop_green,     na.rm = TRUE),
    green_sd          = sd(  mean_prop_green,     na.rm = TRUE),
    green_med         = median(mean_prop_green,   na.rm = TRUE),
    green_q1          = quantile(mean_prop_green, 0.25, na.rm = TRUE),
    green_q3          = quantile(mean_prop_green, 0.75, na.rm = TRUE),
    non_green_avg     = mean(mean_prop_non_green,     na.rm = TRUE),
    non_green_sd      = sd(  mean_prop_non_green,     na.rm = TRUE),
    non_green_med     = median(mean_prop_non_green,   na.rm = TRUE),
    non_green_q1      = quantile(mean_prop_non_green, 0.25, na.rm = TRUE),
    non_green_q3      = quantile(mean_prop_non_green, 0.75, na.rm = TRUE),
    .groups           = "drop"
  )

# a) Renommer les colonnes
df_region_summary <- df_region_summary %>%
  dplyr::rename(
    Region               = region,
    `Nb pays`            = n_countries,
    `Moy % verts`        = green_avg,
```

```
    `SD % verts`         = green_sd,
    `Med % verts`        = green_med,
    `Q1 % verts`         = green_q1,
    `Q3 % verts`         = green_q3,
    `Moy % non-verts`    = non_green_avg,
    `SD % non-verts`     = non_green_sd,
    `Med % non-verts`    = non_green_med,
    `Q1 % non-verts`     = non_green_q1,
    `Q3 % non-verts`     = non_green_q3
  )

# b) Réordonner les colonnes
df_region_summary <- df_region_summary %>%
  dplyr::select(
    Region,
    `Nb pays`,
    `Moy % verts`, `SD % verts`, `Med % verts`, `Q1 % verts`, `Q3 % verts`,
    `Moy % non-verts`, `SD % non-verts`, `Med % non-verts`, `Q1 % non-verts`,
     ↪  `Q3 % non-verts`
  )

# Affichage
knitr::kable(df_region_summary, digits = 2,
             caption = "Statistiques descriptives des parts d'EFPV et EnFPV
              ↪  par région")
```

### 1.4 Visualisation fine : violin plots

```
library(dplyr)
library(ggplot2)

# On s'assure que df_country_year contient bien prop_green et la variable
 ↪  income_group
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  group_by(iso3c, year, income_group) %>%
  summarise(
    total_jobs = sum(value, na.rm = TRUE),
    green_jobs = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    prop_green = green_jobs / total_jobs * 100,
```

```r
    .groups     = "drop"
  )

# Transforme income_group en facteur avec labels lisibles et ordre désiré
df_country_year <- df_country_year %>%
  mutate(
    income_group = factor(income_group,
      levels = c(1, 4, 3, 2),
      labels = c("Revenus élevés",
                 "Moyen supérieur",
                 "Moyen inférieur",
                 "Faibles revenus")
    )
  )

# Violin + boxplot de prop_green par groupe de revenu
p <- ggplot(df_country_year, aes(x = income_group, y = prop_green, fill =
↪  income_group)) +
  geom_violin(trim = FALSE, alpha = 0.6) +
  geom_boxplot(width = 0.1, outlier.size = 0.5,
               position = position_nudge(x = 0.15)) +
  labs(
    title = "Répartition de la part d'EFPV par groupe de revenu",
    x     = "Groupe de revenu",
    y     = "% d'EFPV"
  ) +
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, 20)) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.x     = element_text(angle = 45, hjust = 1)
  )

# Résultats
print(p)

# Téléchargement graphique
ggsave(
  filename = "/Users/.../Chap2/violin_plot_green_jobs_by_income.png",
  plot     = p,
  width    = 8,
  height   = 6,
  dpi      = 300,
```

```r
  bg       = "white"
)
```

```r
library(dplyr)
library(ggplot2)

# Recalcule df_country_year en y incluant prop_non_green
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  group_by(iso3c, year, income_group) %>%
  summarise(
    total_jobs     = sum(value, na.rm = TRUE),
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0), na.rm = TRUE),
    prop_green     = 100 * green_jobs     / total_jobs,
    prop_non_green = 100 * non_green_jobs / total_jobs,
    .groups        = "drop"
  )

# Transforme income_group en facteur avec l'ordre souhaité
df_country_year <- df_country_year %>%
  mutate(
    income_group = factor(income_group,
      levels = c(1, 4, 3, 2),
      labels = c("Revenus élevés",
                 "Moyen supérieur",
                 "Moyen inférieur",
                 "Faibles revenus")
    )
  )

# Création du violon + boxplot pour prop_non_green
p_non_green <- ggplot(df_country_year, aes(x = income_group, y =
↪  prop_non_green, fill = income_group)) +
  geom_violin(trim = FALSE, alpha = 0.6) +
  geom_boxplot(width = 0.1, outlier.size = 0.5,
               position = position_nudge(x = 0.15)) +
  labs(
    title = "Répartition de la part d'EnFPV par groupe de revenus",
    x     = "Groupe de revenu",
    y     = "% EnFPV"
  ) +
```

```r
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, 20)) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.x     = element_text(angle = 45, hjust = 1)
  )

# Résultats
print(p_non_green)

# Téléchargement
ggsave(
  filename =
   ↪  "/Users/.../Chap2/Chap3_violin_plot_non_green_jobs_by_income.png",
  plot     = p_non_green,
  width    = 8,
  height   = 6,
  dpi      = 300,
  bg       = "white"
)
```

```r
library(dplyr)
library(ggplot2)

# Recalcul avec region
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  group_by(iso3c, year, income_group, region) %>%
  summarise(
    total_jobs = sum(value, na.rm = TRUE),
    green_jobs = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    prop_green = 100 * green_jobs / total_jobs,
    .groups    = "drop"
  )

# Renomme les modalités de région
df_country_year <- df_country_year %>%
  mutate(
    region = factor(region,
      levels = 1:5,
      labels = c("Afrique", "Amériques", "Asie", "Europe", "Oceanie")
    )
```

```r
  )

# Créer le graphique
p_region <- ggplot(
  df_country_year %>% filter(!is.na(region)),
  aes(x = region, y = prop_green, fill = region)
) +
  geom_violin(trim = FALSE, alpha = 0.6) +
  geom_boxplot(width = 0.1, outlier.size = 0.5,
               position = position_nudge(x = 0.15, y = 0)) +
  labs(
    title = "Répartition de la part d'EFPV par région",
    x     = "Région",
    y     = "% d'EFPV"
  ) +
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, 20)) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.x     = element_text(angle = 45, hjust = 1)
  )

# Résultat
p_region

# Téléchargement
ggsave("/Users/.../Chap2/violin_plot_green_jobs_by_region.png", plot =
 ↪  p_region,
       width = 8, height = 6, dpi = 300, bg = "white")
```

```r
library(dplyr)
library(ggplot2)

# Préparation des données avec region et calcul des proportions
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  mutate(
    region = factor(region, levels = 1:5,
                    labels = c("Afrique", "Ameriques", "Asie", "Europe",
                     ↪  "Oceanie"))
  ) %>%
  group_by(iso3c, year, income_group, region) %>%
```

```
  summarise(
    total_jobs     = sum(value, na.rm = TRUE),
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0), na.rm = TRUE),
    prop_green     = 100 * green_jobs     / total_jobs,
    prop_non_green = 100 * non_green_jobs / total_jobs,
    .groups        = "drop"
  )

# Création du graphique
p_non_green_region <- ggplot(df_country_year, aes(x = region, y =
↪ prop_non_green, fill = region)) +
  geom_violin(trim = FALSE, alpha = 0.6) +
  geom_boxplot(
    width = 0.1, outlier.size = 0.5,
    position = position_nudge(x = 0.15, y = 0)
  ) +
  labs(
    title = "Répartition de la part d'EnFPV par groupe de revenus",
    x     = "Région",
    y     = "% d'EnFPV"
  ) +
  scale_y_continuous(
    limits = c(0, 100),
    breaks = seq(0, 100, 20)
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.x     = element_text(angle = 45, hjust = 1)
  )

# Résultats
p_non_green_region

# Téléchargement
ggsave("/Users/.../Chap2/violin_plot_non_green_jobs_by_region.png", plot =
↪ p_non_green_region,
      width = 8, height = 6, dpi = 300, bg = "white")
```

## 1.5 Cartographie interactive

```r
# Le colonne iso_a3 de rnaturalearth ne contient pas les codes tels que "FRA"
↪ ou "XKX"
library(rnaturalearth)
world <- ne_countries(scale="medium", returnclass="sf") %>%
  # ici on prend adm0_a3 qui, lui, contient bien "FRA"
  dplyr::select(iso3c = adm0_a3, geometry)

# Vérifions à nouveau :
any(world$iso3c == "FRA")
#> TRUE
```

```r
library(sf)
library(dplyr)
library(gridExtra)

# Chargement de la carte via adm0_a3
world <- ne_countries(scale = "medium", returnclass = "sf") %>%
  dplyr::select(iso3c = adm0_a3, geometry)

# Vérifions si XKX est déjà présent
if (!"XKX" %in% world$iso3c) {
  # Crée un point vide au même CRS que 'world'
  empty_geom <- st_sfc(st_point(), crs = st_crs(world))
  # Ajoute une ligne avec géométrie vide
  world <- bind_rows(
    world,
    tibble(iso3c = "XKX", geometry = empty_geom)
  )
}

# a) Moyenne globale (toutes années) par pays
##    On utilise df_country_year qui contient bien prop_green et
↪ prop_non_green
df_world_global <- df_country_year %>%
  group_by(iso3c) %>%                                       ## un résumé
  ↪ par pays
  summarise(
    pct_green_global     = mean(prop_green,     na.rm = TRUE), ## moyenne %
    ↪ verts
```

```r
    pct_non_green_global = mean(prop_non_green, na.rm = TRUE), ## moyenne %
     ↪  non-verts
    .groups = "drop"
  )

# Calcul de l'année la mieux couverte dans df_country_year
best_year <- df_country_year %>%
  count(year) %>%
  slice_max(n, with_ties = FALSE) %>%
  pull(year)

# Affichage pour vérifier
message("Année la mieux couverte : ", best_year)

# b) À l’année la mieux couverte (best_year)
df_world_best <- df_country_year %>%
  filter(year == best_year) %>%
  dplyr::select(
    iso3c,
    pct_green_best     = prop_green,     ## récupère prop_green
    pct_non_green_best = prop_non_green  ## récupère prop_non_green
  )

# Fonction de création de carte
make_map <- function(data_sf, var, title) {
  ggplot(data_sf) +
    geom_sf(aes_string(fill = var), color = "grey60", size = 0.15) +
    scale_fill_gradient(
      low      = "#e5f5e0",
      high     = "#006d2c",
      name     = "% emplois",
      na.value = "grey90"
    ) +
    labs(title = title) +
    theme_minimal() +
    theme(
      panel.grid      = element_blank(),
      axis.text       = element_blank(),
      axis.ticks      = element_blank(),
      legend.position = "bottom"
    )
}
```

```r
# Assemblage des sf + données
# a) global verts
map_global_green <- world %>%
  left_join(df_world_global, by = "iso3c")

# b) global non-verts
map_global_non <- world %>%
  left_join(
    df_world_global %>% dplyr::select(iso3c, pct_non = pct_non_green_global),
    by = "iso3c"
  )

# c) best_year verts
map_best_green <- world %>%
  left_join(
    df_world_best %>% dplyr::select(iso3c, pct = pct_green_best),
    by = "iso3c"
  )

# d) best_year non-verts
map_best_non <- world %>%
  left_join(
    df_world_best %>% dplyr::select(iso3c, pct = pct_non_green_best),
    by = "iso3c"
  )

# Création des 4 graphiques
p1 <- make_map(map_global_green, "pct_green_global",     "% EFPV\n(Toutes les
↪  années)")
p2 <- make_map(map_global_non,    "pct_non",             "% EnFPV\n(Toutes
↪  les années)")
p3 <- make_map(map_best_green,    "pct", paste0("% EFPV\n(", best_year, ")"))
p4 <- make_map(map_best_non,      "pct", paste0("% EnFPV\n(", best_year,
↪  ")"))

# Affichage en 2×2
grid.arrange(p1, p2, p3, p4, ncol = 2)
# Affichage séquentiel dans la console
print(p1)
print(p2)
print(p3)
print(p4)
```

```r
# Téléchargement
ggsave("/Users/.../Chap2/map_JHGP_all_years.png", plot = p1, width = 6,
↪  height = 4, dpi = 300, bg = "white")
ggsave("/Users/.../Chap2/map_JnHGP_all_years.png", plot = p2, width = 6,
↪  height = 4, dpi = 300, bg = "white")
ggsave(paste0("/Users/.../Chap2/map_JHGP_", best_year, ".png"), plot = p3,
↪  width = 6, height = 4, dpi = 300, bg = "white")
ggsave(paste0("/Users/.../Chap2/map_JnHGP_", best_year, ".png"), plot = p4,
↪  width = 6, height = 4, dpi = 300, bg = "white")
```

### 1.6 Tests post-hoc et non-paramétriques

```r
library(dplyr)

# Calcul de prop_green et prop_non_green par pays-année-groupe de revenu
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  group_by(iso3c, year, income_group) %>%
  summarise(
    total_jobs     = sum(value, na.rm = TRUE),
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0), na.rm = TRUE),
    prop_green     = 100 * green_jobs     / total_jobs,
    prop_non_green = 100 * non_green_jobs / total_jobs,
    .groups        = "drop"
  )

# Moyenne par pays × groupe de revenu
df_country_means <- df_country_year %>%
  group_by(iso3c, income_group) %>%
  summarise(
    mean_prop_green     = mean(prop_green,     na.rm = TRUE),
    mean_prop_non_green = mean(prop_non_green, na.rm = TRUE),
    .groups             = "drop"
  )
```

```r
kw <- kruskal.test(mean_prop_green ~ income_group,
                   data = df_country_means)
print(kw)
```

```r
if (kw$p.value < 0.05) {
  library(dunn.test)
  dunn.test(
    df_country_means$mean_prop_green,
    g      = df_country_means$income_group,
    method = "holm"
  )
}
```

```r
# Chargement des packages nécessaires
library(dplyr)
library(dunn.test)

# On repart de df_country_year généré précédemment (qui contient region)
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  mutate(
    region = factor(region,
      levels = 1:5,
      labels = c("Africa","Americas","Asia","Europe","Oceania")
    )
  ) %>%
  group_by(iso3c, year, region) %>%
  summarise(
    total_jobs  = sum(value, na.rm = TRUE),
    green_jobs  = sum(if_else(EU_ISIC4_2dg == 2, value, 0), na.rm = TRUE),
    prop_green  = green_jobs / total_jobs * 100,
    .groups     = "drop"
  )

# Moyenne par pays × région
df_country_means_region <- df_country_year %>%
  group_by(iso3c, region) %>%
  summarise(
    mean_prop_green = mean(prop_green, na.rm = TRUE),
    .groups         = "drop"
  )

# Kruskal-Wallis global sur % verts par région
kw_region <- kruskal.test(
  mean_prop_green ~ region,
  data = df_country_means_region
```

```
)
print(kw_region)

# Test post-hoc de Dunn si p < 0.05
if (kw_region$p.value < 0.05) {
  library(dunn.test)
  dunn_region <- dunn.test(
    x      = df_country_means_region$mean_prop_green,
    g      = df_country_means_region$region,
    method = "holm",
    kw     = FALSE
  )
  print(dunn_region)
}
```

### 1.7 Analyse de clusters

Clustering hiérarchique ou k-means sur les vecteurs (%verts, %non_verts) pour regrouper les pays en "profils" de verdissement + comparer ces clusters aux classes de revenu et de région : est-ce qu'ils coïncident ? Ou apparaissent de nouveaux groupes (par exemple des pays "outliers" , ceux qui ont plus de vert que leur groupe de revenu, etc.)

```
EMP_TEMP_SEX_EDU_EC2_NB_A <-
↪  read.csv("/Users/.../Chap2/EMP_TEMP_SEX_EDU_EC2_NB_A.csv")
```

```
# Ménage
ls()                # liste les objets existants
rm(df_country_means)  # supprime l'ancien
rm(df_country_year)   # idem si besoin

# Refait df_country_year
df_country_year <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  mutate(
    region = factor(region,
      levels = 1:5,
      labels = c("Africa","Americas","Asia","Europe","Oceania")
    )
  ) %>%
  group_by(iso3c, year, income_group, region) %>%
  summarise(
```

```r
    total_jobs     = sum(value,                                    na.rm =
     ↪  TRUE),
    green_jobs     = sum(if_else(EU_ISIC4_2dg == 2, value, 0),    na.rm =
     ↪  TRUE),
    non_green_jobs = sum(if_else(EU_ISIC4_2dg == 1, value, 0),    na.rm =
     ↪  TRUE),
    prop_green     = 100 * green_jobs     / total_jobs,
    prop_non_green = 100 * non_green_jobs / total_jobs,
    .groups        = "drop"
  )

# Vérification
str(df_country_year)
# doit afficher iso3c, year, income_group, region, total_jobs, green_jobs,
 ↪  non_green_jobs, prop_green, prop_non_green

# 2) Refait df_country_means
df_country_means <- df_country_year %>%
  group_by(iso3c, income_group, region) %>%
  summarise(
    mean_prop_green     = mean(prop_green,     na.rm = TRUE),
    mean_prop_non_green = mean(prop_non_green, na.rm = TRUE),
    .groups             = "drop"
  )

# Vérification
str(df_country_means)
# doit afficher iso3c, income_group, region, mean_prop_green,
 ↪  mean_prop_non_green

# 3) Et enfin la sélection + standardisation
df_clust <- df_country_means %>%
  dplyr::select(iso3c, income_group, region,
                mean_prop_green, mean_prop_non_green) %>%
  mutate(
    z_green     = as.numeric(scale(mean_prop_green)),
    z_non_green = as.numeric(scale(mean_prop_non_green))
  )

# Contrôle final
str(df_clust)
# doit afficher iso3c, income_group, region, mean_prop_green,
 ↪  mean_prop_non_green, z_green, z_non_green
```

```r
library(dplyr)
library(cluster)
library(factoextra)
library(tibble)


# Matrice des indicateurs
mat <- df_country_means %>%
  dplyr::select(iso3c, region, income_group,
                mean_prop_green, mean_prop_non_green) %>%
  tibble::column_to_rownames("iso3c") %>%
  dplyr::select(mean_prop_green, mean_prop_non_green)

# Détermination du nombre optimal de clusters (méthode silhouette)
fviz_nbclust(mat, kmeans, method = "silhouette") +
  labs(subtitle = "Silhouette")

# Application de k-means (par exemple k = 4)
set.seed(42)
km4 <- kmeans(mat, centers = 4, nstart = 25)

# Ajout des labels de cluster à ton df
df_clusters <- df_country_means %>%
  mutate(cluster = factor(km4$cluster[iso3c]))

# Résultat
fviz_cluster(km4, data = mat,
             geom = "point",
             ellipse.type = "norm",
             palette = "jco",
             ggtheme = theme_minimal())
```

```r
# Avant le clustering, on renomme income_group
df_clust <- df_clust %>%
  mutate(
    income_group = factor(
      income_group,
      levels = 1:4,
      labels = c(
        "High income",
        "Low income",
```

```r
        "Lower middle income",
        "Upper middle income"
      )
    )
  )

# Choix du nombre optimal de clusters pour kmeans
set.seed(42)
p_silhouette <- fviz_nbclust(
  dplyr::select(df_clust, z_green, z_non_green),
  FUN = kmeans,
  method = "silhouette",
  k.max = 6
) +
  labs(title = "Choosing the optimal number of clusters", subtitle =
   ↪  "Silhouette for k-means")

# k-means à k = 4
set.seed(42)
km4 <- kmeans(
  dplyr::select(df_clust, z_green, z_non_green),
  centers = 4,
  nstart  = 50
)
df_clust <- df_clust %>% mutate(cluster_km = factor(km4$cluster))

# Hiérarchique (Ward)
diss_mat <- dist(dplyr::select(df_clust, z_green, z_non_green))
hc_ward <- hclust(diss_mat, method = "ward.D2")
df_clust$cluster_hc <- factor(cutree(hc_ward, k = 4))

# Graphique
p1 <- fviz_cluster(
  km4, data = dplyr::select(df_clust, z_green, z_non_green),
  geom = "point", ellipse.type = "convex", ggtheme = theme_minimal()
) + labs(title = "k-means (k=4)")

p2 <- fviz_dend(
  hc_ward, k = 4, rect = TRUE,
  k_colors = "jco", ggtheme = theme_minimal()
) + theme(legend.position = "none") +
  labs(title = "Hier. (Ward, k=4)")
```

```r
library(gridExtra)
grid.arrange(p1, p2, ncol = 2)

# Tables de contingence
# k-means × income_group
table(df_clust$cluster_km, df_clust$income_group) %>%
  prop.table(margin = 2) %>% round(2) %>%
  knitr::kable(caption = "k-means × Groupe de revenu")

# k-means × region
table(df_clust$cluster_km, df_clust$region) %>%
  prop.table(margin = 2) %>% round(2) %>%
  knitr::kable(caption = "k-means × Région")

# hiérarchique × income_group
table(df_clust$cluster_hc, df_clust$income_group) %>%
  prop.table(margin = 2) %>% round(2) %>%
  knitr::kable(caption = "Hier. × Groupe de revenu")

# hiérarchique × region
table(df_clust$cluster_hc, df_clust$region) %>%
  prop.table(margin = 2) %>% round(2) %>%
  knitr::kable(caption = "Hier. × Région")

# 6) Profils atypiques
df_clust %>%
  count(income_group, cluster_km) %>%
  group_by(income_group) %>%
  filter(n < max(n)) %>%
  arrange(income_group, cluster_km) %>%
  knitr::kable(
    caption = "Pays dans des clusters k-means minoritaires par groupe de
     ↪  revenu"
  )

# Téléchargement
ggsave(
  filename = "/Users/.../Chap2/silhouette_kmeans.png",
  plot = p_silhouette,
  width = 7,
  height = 5,
  dpi = 300,
  bg = "white"
```

```
)
```

1er graphique : montre que la largeur moyenne des silhouettes est maximale pour **k = 2** (  0.58), ce qui signifie que deux groupes tirent le meilleur compromis entre compacité et séparation. Au-delà, la qualité du découpage décroit (k = 3 à 6 voient leurs silhouettes moyennes baisser). Concrètement : **k = 2** : silhouette  0.58 ; **k = 3** : silhouette  0.52 ; **k = 4** : silhouette  0.46 ; **k = 5** : silhouette  0.46 ; **k = 6** : silhouette  0.44.

**k = 2** donne le meilleur score de silhouette, mais k = 4 permet de distinguer quatre profils de « verdissement » qui **ne coïncident pas strictement** avec les 4 catégories de revenus : Le cluster le plus « vert » (kmeans 4) contient surtout High income, mais aussi quelques Upper middle. Le cluster le plus « brun » (kmeans 1) mélange principalement Low et Lower middle. Le clustering hiérarchique aboutit à des découpages similaires (Europe très distinct, Amériques ensemble, etc.)  ; là aussi, on observe un léger **décalage** par rapport aux seules classes de revenus. Les « outliers » montrent des pays plus (ou moins) verts que leur groupe de revenu ne le suggérerait, suggérant des **profils nationaux atypiques** qui pourraient faire l'objet d'études de cas.

1. ***k-means (k=4) x groupe de revenu***

| Cluster k-means | High income | Low income | Lower middle income | Upper middle income |
|---|---|---|---|---|
| **1** | 0.00 | 0.47 | 0.12 | 0.00 |
| **2** | 0.36 | 0.05 | 0.30 | 0.54 |
| **3** | 0.00 | 0.47 | 0.53 | 0.34 |
| **4** | 0.64 | 0.00 | 0.05 | 0.11 |

Chaque cellule indique la proportion (entre 0 et 1) des pays d'un **groupe de revenu** donné (colonne) qui appartient à chaque **cluster k-means** (ligne). **Cluster 4** concentre clairement la majorité des **High income** (64 %), tandis que **Cluster 1** est quasi-exclusif des pays à revenus intermédiaires/inferieurs. **Cluster 2** associe majoritairement **Upper middle** (54 %) et **High** (36 %), **Cluster 3** rassemble surtout **Low** et **Lower middle**.

2. ***k-means (k=4) x région***

| Cluster k-means | Africa | Americas | Asia | Europe | Oceania |
|---|---|---|---|---|---|
| **1** | 0.30 | 0.00 | 0.03 | 0.00 | 0.06 |
| **2** | 0.22 | 0.62 | 0.31 | 0.38 | 0.31 |
| **3** | 0.45 | 0.29 | 0.44 | 0.06 | 0.31 |
| **4** | 0.03 | 0.10 | 0.22 | 0.56 | 0.31 |

**Cluster 2** regroupe surtout les **Amériques** (62 %) et un tiers de l'**Asie**, **Cluster 4** est majoritairement **Europe** (56 %), **Cluster 1** est plus africain et océanien, **Cluster 3** disperse davantage sur **Africa/Asia/Oceania**.

3. ***Clustering hiérarchique (Ward, k=4) x groupe de revenu***

| Cluster HC | High income | Low income | Lower middle income | Upper middle income |
|---|---|---|---|---|
| **1** | 0.04 | 0.47 | 0.58 | 0.34 |
| **2** | 0.14 | 0.05 | 0.23 | 0.43 |
| **3** | 0.82 | 0.00 | 0.12 | 0.23 |
| **4** | 0.00 | 0.47 | 0.07 | 0.00 |

**Cluster 3** (Ward) rassemble la quasi-totalité des **High income** (82 %), **Cluster 1** se concentre sur les **Lower middle** et **Low**, **Cluster 2** sur les **Upper middle**, **Cluster 4** est très minoritaire et mixte.

4. ***Clustering hiérarchique (Ward, k=4) x région***

| Cluster HC | Africa | Americas | Asia | Europe | Oceania |
|---|---|---|---|---|---|
| **1** | 0.50 | 0.29 | 0.44 | 0.06 | 0.38 |
| **2** | 0.17 | 0.48 | 0.22 | 0.19 | 0.19 |
| **3** | 0.07 | 0.24 | 0.31 | 0.75 | 0.38 |
| **4** | 0.25 | 0.00 | 0.03 | 0.00 | 0.06 |

**Cluster 3** (Ward) concentre l'**Europe** (75 %), **Cluster 2** rassemble les **Amériques**, **Cluster 1** touche surtout **Africa/Asia/Oceania**, **Cluster 4** est très marginal.

5. ***Pays "atypiques" dans k-means vs revenue***

| income_group | cluster_km | n |
|---|---|---|
| High income | 2 | 10 |
| Low income | 2 | 1 |
| Lower middle income | 1 | 5 |
| Lower middle income | 2 | 13 |
| Lower middle income | 4 | 2 |
| Upper middle income | 3 | 12 |
| Upper middle income | 4 | 4 |

Ce tableau liste les pays **dont le cluster k-means n'est pas le modal** de leur groupe de revenu. On voit notamment que 10 **High income** tombent dans le **Cluster 2** (plutôt "moyen-haut"), et 12 **Upper middle** dans le **Cluster 3** (plus "vert"), etc.

```r
library(dplyr)
library(knitr)

# Repérer le cluster majoritaire par groupe de revenu
major_clusters <- df_clust %>%
  dplyr::group_by(income_group, cluster_km) %>%
  dplyr::tally(name = "n") %>%
  dplyr::slice_max(n, with_ties = FALSE) %>%
  dplyr::ungroup() %>%
  dplyr::rename(major_cluster = cluster_km) %>%
  dplyr::select(income_group, major_cluster)

# 2) Construit le tableau des outliers
df_outliers <- df_clust %>%
  dplyr::left_join(major_clusters, by = "income_group") %>%
  dplyr::filter(cluster_km != major_cluster) %>%
  dplyr::select(
    Pays        = iso3c,
    `Groupe de revenu` = income_group,
    `Cluster k-means`  = cluster_km,
    `Cluster majoritaire` = major_cluster
  )

# 3) Résultat
knitr::kable(
  df_outliers %>%
    dplyr::arrange(`Groupe de revenu`, `Cluster k-means`),
  caption = "Pays atypiques vis-à-vis de leur groupe de revenu"
)
```

| ISO3C | Pays | Groupe de revenu | Cluster k-means | Cluster majoritaire |
|---|---|---|---|---|
| ARE | Arabie saoudite | High income | 2 | 4 |
| BRB | Barbade | High income | 2 | 4 |
| CHL | Chili | High income | 2 | 4 |
| CYP | Chypre | High income | 2 | 4 |
| GRC | Grèce | High income | 2 | 4 |
| GUY | Guyana | High income | 2 | 4 |

| ISO3C | Pays | Groupe de revenu | Cluster k-means | Cluster majoritaire |
|---|---|---|---|---|
| NRU | Nauru | High income | 2 | 4 |
| PAN | Panama | High income | 2 | 4 |
| PRT | Portugal | High income | 2 | 4 |
| URY | Uruguay | High income | 2 | 4 |
| GMB | Gambie | Low income | 2 | 1 |
| AFG | Afghanistan | Low income | 3 | 1 |
| BFA | Burkina Faso | Low income | 3 | 1 |
| GNB | Guinée-Bissau | Low income | 3 | 1 |
| LBR | Libéria | Low income | 3 | 1 |
| NER | Niger | Low income | 3 | 1 |
| RWA | Rwanda | Low income | 3 | 1 |
| SDN | Soudan | Low income | 3 | 1 |
| SLE | Sierra Leone | Low income | 3 | 1 |
| TGO | Togo | Low income | 3 | 1 |
| BEN | Bénin | Lower middle income | 1 | 3 |
| LAO | Laos | Lower middle income | 1 | 3 |
| VUT | Vanuatu | Lower middle income | 1 | 3 |
| ZMB | Zambie | Lower middle income | 1 | 3 |
| ZWE | Zimbabwe | Lower middle income | 1 | 3 |
| COG | Congo (Brazzaville) | Lower middle income | 2 | 3 |
| CPV | Cap-Vert | Lower middle income | 2 | 3 |
| EGY | Égypte | Lower middle income | 2 | 3 |
| JOR | Jordanie | Lower middle income | 2 | 3 |
| KIR | Kiribati | Lower middle income | 2 | 3 |
| LBN | Liban | Lower middle income | 2 | 3 |
| LSO | Lesotho | Lower middle income | 2 | 3 |
| NPL | Népal | Lower middle income | 2 | 3 |

| ISO3C | Pays | Groupe de revenu | Cluster k-means | Cluster majoritaire |
|---|---|---|---|---|
| PHL | Philippines | Lower middle income | 2 | 3 |
| PNG | Papouasie-Nouvelle-Guinée | Lower middle income | 2 | 3 |
| SWZ | Eswatini | Lower middle income | 2 | 3 |
| TJK | Tadjikistan | Lower middle income | 2 | 3 |
| TUN | Tunisie | Lower middle income | 2 | 3 |
| NIU | Niue | Lower middle income | 4 | 3 |
| PSE | Territoires palestiniens | Lower middle income | 4 | 3 |
| ALB | Albanie | Upper middle income | 3 | 2 |
| ECU | Équateur | Upper middle income | 3 | 2 |
| FJI | Fidji | Upper middle income | 3 | 2 |
| GEO | Géorgie | Upper middle income | 3 | 2 |
| GTM | Guatemala | Upper middle income | 3 | 2 |
| IDN | Indonésie | Upper middle income | 3 | 2 |
| MNG | Mongolie | Upper middle income | 3 | 2 |
| NAM | Namibie | Upper middle income | 3 | 2 |
| PER | Pérou | Upper middle income | 3 | 2 |
| SLV | Salvador | Upper middle income | 3 | 2 |
| THA | Thaïlande | Upper middle income | 3 | 2 |
| TON | Tonga | Upper middle income | 3 | 2 |
| BLR | Biélorussie | Upper middle income | 4 | 2 |

| ISO3C | Pays | Groupe de revenu | Cluster k-means | Cluster majoritaire |
|---|---|---|---|---|
| IRQ | Irak | Upper middle income | 4 | 2 |
| MDV | Maldives | Upper middle income | 4 | 2 |
| UKR | Ukraine | Upper middle income | 4 | 2 |

La colonne "cluster k-means" détaille ici où le pays se situe réellement, selon ses données vert/non-vert, et la colonne "cluster majoritaire" désigne où la plupart des pays de son groupe de revenu se situent. Les **pays listés** dans ce tableau sont ceux **pour lesquels ces deux clusters diffèrent**, autrement dit ceux dont le profil de verdissement **ne correspond pas** à celui majoritaire de leur catégorie de revenu.

6. Nuage de points coloré par cluster et formes selon income_group

```r
library(ggplot2)

# Vecteur de labels
income_levels <- c(
  `1` = "Revenu élevé",
  `2` = "Revenu faible",
  `3` = "Revenu moyen inférieur",
  `4` = "Revenu moyen supérieur"
)

# Création du graphique
p_scatter <- ggplot(df_clust, aes(x = z_green, y = z_non_green)) +
  geom_point(aes(color = cluster_km, shape = factor(income_group)), size = 3,
   ↪  alpha = 0.8) +
  scale_color_brewer(palette = "Set1", name = "Cluster k-means") +
  scale_shape_manual(
    name = "Groupe de revenu",
    values = c(16, 17, 15, 3), # cercles, triangles, carrés, croix
    labels = income_levels
  ) +
  labs(
    title    = "Profils de verdissement des pays par regroupement k-means
     ↪  (k=4)",
    subtitle = "Axes normalisés : z_green vs. z_non_green",
    x        = "z-score proportion d'EFPV",
```

```r
    y         = "z-score proportion d'EnFPV"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "right")

# Résultat
p_scatter

# Téléchargement
ggsave(
  filename = "/Users/.../Chap2/scatterplot_country_greening_clusters.png",
  plot = p_scatter,
  width = 8, height = 6, dpi = 300, bg = "white"
)
```

```r
readr::write_csv(df_c, "data/df_clust.csv")
# Chemins de sauvegarde
library(readr)

write_csv(df_country_means, "/Users/.../Chap2/df_country_means.csv")

write_csv(df_c, "/Users/.../Chap2/df_clust.csv")
```

# 2 Secteurs prédominants

## 2.1 Détail des secteurs

Tableau. Synthèse des codes ISIC 2 digit (source : https://unstats.un.org/unsd/classifications/ISIC/revision)

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| TOTAL | 1 | | covers aggregate activities when no more specific division applies; used for totals and unspecified categories |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| X | 2 | | denotes an indeterminate or mixed category not classifiable elsewhere |
| A01 | 3 | Crop and animal production, hunting and related service activities | Includes all primary production of crops and animals (including organic and genetically modified), plus incidental agricultural services, hunting and trapping. Excludes any further processing beyond preparing products for primary markets. |
| A02 | 4 | Forestry and logging | Covers plantation and natural forest management, roundwood production (for paper, timber industries) and gathering of non-wood forest products (firewood, charcoal, chips). |
| A03 | 5 | Fishing and aquaculture | Encompasses wild capture fisheries and aquaculture in marine, brackish or freshwater environments, including seed production (e.g. oysters). Excludes vessel construction/repair and recreational fishing; processing falls under manufacturing divisions. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| B05 | 6 | Mining of coal and lignite | Extraction of solid mineral fuels by underground or open-cast methods, including washing, grading and other steps to produce a marketable product. Excludes coking, briquette manufacture and mining support services. |
| B06 | 7 | Extraction of crude petroleum and natural gas | Covers exploration, drilling, extraction and initial treatment of crude oil, shale oil, oil sands and natural gas up to shipment point. Excludes refining and geophysical or well-testing services. |
| B07 | 8 | Mining of metal ores | Extraction of metallic ores via underground, open-cast or seabed mining, plus ore dressing and beneficiation (crushing, grinding, separation). Excludes smelting, roasting or blast furnace operations. |
| B08 | 9 | Other mining and quarrying | Extraction from quarries or mines of non-metallic minerals; dredging, rock crushing, sorting and basic preparation for industrial or construction use. Excludes further processing beyond primary mechanical treatment. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| B09 | 10 | Mining support service activities | Specialized services on contract for mining operations, including drilling, site preparation, well-foundation work, pumping, overburden removal, geological surveying and related support activities. |
| C10 | 11 | Manufacture of food products | Processing of agricultural, forestry and fishery products into foods for humans or animals, including intermediate products. Includes bakeries and slaughterhouses; excludes feed production from slaughter waste and conversion of food waste to raw materials. |
| C11 | 12 | Manufacture of beverages | Production of alcoholic and non-alcoholic beverages (beer, wine, spirits, soft drinks). Excludes fruit or vegetable juice production, milk-based drinks and coffee/tea manufacture. |
| C12 | 13 | Manufacture of tobacco products | Processing of tobacco into consumer products (cigarettes, cigars, smokeless tobacco) suitable for final consumption. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| C13 | 14 | Manufacture of textiles | Spinning, weaving, finishing of textile fibers and yarns, plus production of made-up textile articles (excluding apparel). Excludes fiber growing (division 1) and synthetic fiber manufacture (class 2030). |
| C14 | 15 | Manufacture of wearing apparel | All tailoring and garment production (ready-to-wear and bespoke) in all materials, including fur. No distinction by age or gender; includes accessories. |
| C15 | 16 | Manufacture of leather and related products | Tanning, dyeing and finishing of leather and fur, and fabrication into final products (footwear, bags, luggage), including imitation leather equivalents. |
| C16 | 17 | Manufacture of wood and of products of wood and cork… | Sawing, planing and assembling timber into products (lumber, plywood, containers), plus articles of cork, straw and weaving materials. Excludes furniture manufacture. |
| C17 | 18 | Manufacture of paper and paper products | Production of pulp, paper and converted paper products; often vertically integrated processes from pulp to finished goods. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| C18 | 19 | Printing and reproduction of recorded media | Printing of books, newspapers, forms and packaging; binding, plate-making, data imaging; reproduction of recorded media (CDs, tapes). Excludes publishing. |
| C19 | 20 | Manufacture of coke and refined petroleum products | Refining of crude petroleum and coal into fuels (gasoline, diesel, coke) and basic petrochemicals. Excludes secondary processing of petroleum byproducts in chemical industries. |
| C20 | 21 | Manufacture of chemicals and chemical products | Transformation of organic and inorganic raw materials via chemical processes into basic and intermediate chemicals and related products. |
| C21 | 22 | Manufacture of basic pharmaceutical products and pharmaceutical preparations | Production of medicinal chemical and botanical products, including active ingredients and formulations. |
| C22 | 23 | Manufacture of rubber and plastics products | Molding, extrusion and shaping of rubber and plastics into products (tyres, tubing, packaging). |
| C23 | 24 | Manufacture of other non-metallic mineral products | Manufacture of glass, ceramic, cement, lime, abrasives; includes shaped and finished stone products. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| C24 | 25 | Manufacture of basic metals | Smelting and refining of ferrous and non-ferrous metals from ore, scrap or pig iron; production of alloys. |
| C25 | 26 | Manufacture of fabricated metal products… | Production of metal parts, containers and structures (excluding machinery and equipment); includes weapons; excludes specialized repair and installation. |
| C26 | 27 | Manufacture of computer, electronic and optical products | Assembly of computers, peripherals, communications and optical instruments, including integrated circuits and miniaturization technologies. |
| C27 | 28 | Manufacture of electrical equipment | Production of generators, transformers, wiring, motors, lighting and electrical appliances. Excludes electronic devices. |
| C28 | 29 | Manufacture of machinery and equipment n.e.c. | Manufacture of machinery for material handling, packaging, spraying, and thermal or mechanical processing; general-purpose and special-purpose equipment. Excludes vehicles and electrical apparatus. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| C29 | 30 | Manufacture of motor vehicles, trailers and semi-trailers | Production of cars, trucks, trailers and related parts. Maintenance and repair classified separately. |
| C30 | 31 | Manufacture of other transport equipment | Shipbuilding, aircraft, railway stock and spacecraft manufacturing, plus component assembly. |
| C31 | 32 | Manufacture of furniture | Carpentry, molding and assembly of furniture from wood, metal, plastics or other materials. |
| C32 | 33 | Other manufacturing | Residual manufacturing of diverse products not covered elsewhere; processes and inputs vary widely. |
| C33 | 34 | Repair and installation of machinery and equipment | Specialized maintenance and repair of industrial machinery and installation services; excludes routine servicing and cleaning. |
| D35 | 35 | Electricity, gas, steam and air conditioning supply | Generation, transmission and distribution of electricity, gas supply, steam and air-conditioning services. |
| E36 | 36 | Water collection, treatment and supply | Collection, purification and distribution of water for domestic, industrial and agricultural use. |
| E37 | 37 | Sewerage | Operation of sewer systems and sewage treatment facilities for disposal and treatment of wastewater. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| E38 | 38 | Waste collection, treatment and disposal; materials recovery | Collection, recycling, treatment and disposal of municipal and industrial waste; includes material recovery operations. |
| E39 | 39 | Remediation and other waste management services | Cleanup of contaminated sites (soil, water, buildings) and specialized environmental services. |
| F41 | 40 | Construction of buildings | New construction, renovation, additions and alterations of residential, commercial and industrial buildings. |
| F42 | 41 | Civil engineering | Construction of infrastructure such as roads, bridges, dams, pipelines, tunnels and power lines. |
| F43 | 42 | Specialized construction activities | Trade-specific tasks in construction (plumbing, electrical, roofing, concreting, finishing) and equipment rental with operator. |
| G45 | 43 | Wholesale/retail trade & repair of motor vehicles & motorcycles | Sale of new/used vehicles, parts and accessories; repair, maintenance, washing and polishing; excludes fuel retail and vehicle rental. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| G46 | 44 | Wholesale trade, except of motor vehicles & motorcycles | Wholesale of goods on own account or commission basis, excluding motor vehicles; includes import/export and trade intermediation. |
| G47 | 45 | Retail trade, except of motor vehicles & motorcycles | Sale of new/used goods to the public through stores, stalls, mail order, excluding automotive products and on-site food services. |
| H49 | 46 | Land transport & transport via pipelines | Passenger and freight transport by road, rail and pipelines. |
| H50 | 47 | Water transport | Inland and maritime transport of passengers and cargo, including towing, ferries and excursion boats; excludes separate catering services. |
| H51 | 48 | Air transport | Air and space transport of passengers and freight; excludes airport operations and aircraft maintenance. |
| H52 | 49 | Warehousing & support activities for transportation | Storage and handling of goods, operation of transport infrastructure and cargo consolidation services. |
| H53 | 50 | Postal and courier activities | Mail and parcel pickup, sorting, transport and delivery services; includes express courier operations. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| I55 | 51 | Accommodation | Provision of short-term lodging (hotels, motels, hostels) and related services; excludes long-term real estate leasing. |
| I56 | 52 | Food and beverage service activities | Restaurants, cafes, bars and catering services providing meals and drinks for immediate consumption; excludes food manufacturing. |
| J58 | 53 | Publishing activities | Acquisition of content rights and publication of books, periodicals, directories, software in print, digital or audio formats. |
| J59 | 54 | Motion picture, video & television programme production, sound recording & music publishing | Film and TV production, editing, distribution, plus sound recording and music publishing services. |
| J60 | 55 | Programming and broadcasting activities | Creation, acquisition and transmission of radio, TV and data programmes over airwaves, cable or internet; excludes distribution of subscription packages. |
| J61 | 56 | Telecommunications | Transmission of voice, data, text, sound and video via wired, wireless or satellite networks; includes network operation and maintenance. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| J62 | 57 | Computer programming, consultancy & related activities | Software development, modification, testing, system integration, on-site management and IT consulting services. |
| J63 | 58 | Information service activities | Data processing, hosting, web portals, archives and other primary information supply services. |
| K64 | 59 | Financial service activities, except insurance & pension funding | Banking, fund management and financial intermediation excluding insurance and compulsory social security. |
| K65 | 60 | Insurance, reinsurance & pension funding, except compulsory social security | Underwriting and administration of insurance policies and pension plans; investment of premiums; excludes social security contributions. |
| K66 | 61 | Activities auxiliary to financial service & insurance activities | Brokerage, agency, fund administration and other support services related to banking and insurance. |
| L68 | 62 | Real estate activities | Buying, selling, leasing and management of real estate properties and related services. |
| M69 | 63 | Legal and accounting activities | Legal representation, notarial services, drafting of legal documents, accounting, auditing and bookkeeping. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| M70 | 64 | Activities of head offices; management consultancy activities | Strategic planning, organizational and financial consulting, and oversight of corporate groups. |
| M71 | 65 | Architectural & engineering activities; technical testing & analysis | Architectural design, engineering consulting, surveying, laboratory testing and quality assurance services. |
| M72 | 66 | Scientific research & development | Basic research, applied research and experimental development in natural sciences, engineering, social sciences and humanities; excludes market research. |
| M73 | 67 | Advertising and market research | Creation, placement and analysis of advertising campaigns, plus market and opinion research services. |
| M74 | 68 | Other professional, scientific & technical activities | Specialized professional services not elsewhere classified (e.g. translation, design, scientific consulting). |
| M75 | 69 | Veterinary activities | Clinical and field veterinary services for animals, including diagnostics, surgery and preventive care. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| N77 | 70 | Rental and leasing activities | Renting and leasing of equipment and tangible/non-financial intangible assets (vehicles, machinery, intellectual property); excludes financial leasing. |
| N78 | 71 | Employment activities | Recruitment, job placement, temporary staffing, executive search and related human resource services. |
| N79 | 72 | Travel agency, tour operator, reservation service & related activities | Sale and arrangement of travel, tours and accommodation, plus tourism promotion and guiding services. |
| N80 | 73 | Security and investigation activities | Guard and patrol services, detective work, cash-in-transit operations and alarm monitoring; excludes separate installation or retail of security equipment. |
| N81 | 74 | Services to buildings & landscape activities | Cleaning, maintenance and landscaping of buildings, grounds, specialized exterior cleaning and pest control. |
| N82 | 75 | Office administrative, office support & other business support activities | General office and back-office support services, document processing, call centers and administrative outsourcing. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| O84 | 76 | Public administration & defence; compulsory social security | Government functions, defense activities and administration of social security schemes. |
| P85 | 77 | Education | Primary, secondary and tertiary education, vocational training and adult education services. |
| Q86 | 78 | Human health activities | Hospital care, outpatient services, medical laboratories, dental practice and paramedical services. |
| Q87 | 79 | Residential care activities | Nursing homes, assisted living facilities and residential care combined with health or social services. |
| Q88 | 80 | Social work activities without accommodation | Community and welfare services (counseling, family support) provided outside residential settings. |
| R90 | 81 | Creative, arts & entertainment activities | Production and performance of arts, theatre, music, dance, literature and related cultural events. |
| R91 | 82 | Libraries, archives, museums & other cultural activities | Operation of libraries, museums, archives, botanical and zoological gardens, and heritage site preservation. |
| R92 | 83 | Gambling and betting activities | Casino gaming, lotteries, sports betting and operation of gambling facilities. |

| isic_orig | isic_fact | description | ISIC Rev4 ExplanatoryNoteInclusion |
|---|---|---|---|
| R93 | 84 | Sports activities & amusement & recreation activities | Operation of sports facilities, amusement parks, recreation centers and related entertainment services. |
| S94 | 85 | Activities of membership organizations | Activities of trade unions, professional associations, charitable organizations and other membership-based bodies. |
| S95 | 86 | Repair of computers & personal & household goods | Maintenance and repair of computers, electronics, home appliances, furniture and other household items. |
| S96 | 87 | Other personal service activities | Personal care services (laundry, beauty, funeral services), religious and civic ceremonies. |
| T97 | 88 | Activities of households as employers of domestic personnel | Private households employing domestic staff for childcare, cleaning, gardening and other personal services. |
| T98 | 89 | Undifferentiated goods- & services-producing activities of households for own use | Subsistence production of goods and services by households for their own consumption; no primary activity identifiable. |
| U99 | 90 | Activities of extraterritorial organizations & bodies | Activities of international organizations and diplomatic missions operating beyond national jurisdictions. |

Nous comparons, **par groupe de revenu** et **par statut vert / non-vert**, la part des secteurs

ISIC-2 qui pèsent le plus dans l'emploi.

```r
library(dplyr)
library(tibble)

# Construction de la table de correspondance ISIC 2-digit
rev_isic2 <- tribble(
  ~isic_code_2dg, ~isic_code_2dg_fact, ~isic_label,
  "TOTAL",         1L,                  "Aggregate total",
  "X",             2L,                  "Indeterminate / mixed",
  "A01",           3L,                  "Crop and animal production, hunting
   ↪  and related service activities",
  "A02",           4L,                  "Forestry and logging",
  "A03",           5L,                  "Fishing and aquaculture",
  "B05",           6L,                  "Mining of coal and lignite",
  "B06",           7L,                  "Extraction of crude petroleum and
   ↪  natural gas",
  "B07",           8L,                  "Mining of metal ores",
  "B08",           9L,                  "Other mining and quarrying",
  "B09",          10L,                  "Mining support service activities",
  "C10",          11L,                  "Manufacture of food products",
  "C11",          12L,                  "Manufacture of beverages",
  "C12",          13L,                  "Manufacture of tobacco products",
  "C13",          14L,                  "Manufacture of textiles",
  "C14",          15L,                  "Manufacture of wearing apparel",
  "C15",          16L,                  "Manufacture of leather and related
   ↪  products",
  "C16",          17L,                  "Manufacture of wood and of products
   ↪  of wood and cork",
  "C17",          18L,                  "Manufacture of paper and paper
   ↪  products",
  "C18",          19L,                  "Printing and reproduction of recorded
   ↪  media",
  "C19",          20L,                  "Manufacture of coke and refined
   ↪  petroleum products",
  "C20",          21L,                  "Manufacture of chemicals and chemical
   ↪  products",
  "C21",          22L,                  "Manufacture of basic pharmaceutical
   ↪  products and preparations",
  "C22",          23L,                  "Manufacture of rubber and plastics
   ↪  products",
  "C23",          24L,                  "Manufacture of other non-metallic
   ↪  mineral products",
```

```
"C24",          25L,                     "Manufacture of basic metals",
"C25",          26L,                     "Manufacture of fabricated metal
↪   products",
"C26",          27L,                     "Manufacture of computer, electronic
↪   and optical products",
"C27",          28L,                     "Manufacture of electrical equipment",
"C28",          29L,                     "Manufacture of machinery and
↪   equipment n.e.c.",
"C29",          30L,                     "Manufacture of motor vehicles,
↪   trailers and semi-trailers",
"C30",          31L,                     "Manufacture of other transport
↪   equipment",
"C31",          32L,                     "Manufacture of furniture",
"C32",          33L,                     "Other manufacturing",
"C33",          34L,                     "Repair and installation of machinery
↪   and equipment",
"D35",          35L,                     "Electricity, gas, steam and air
↪   conditioning supply",
"E36",          36L,                     "Water collection, treatment and
↪   supply",
"E37",          37L,                     "Sewerage",
"E38",          38L,                     "Waste collection, treatment and
↪   disposal; materials recovery",
"E39",          39L,                     "Remediation and other waste
↪   management services",
"F41",          40L,                     "Construction of buildings",
"F42",          41L,                     "Civil engineering",
"F43",          42L,                     "Specialized construction activities",
"G45",          43L,                     "Wholesale/retail trade & repair of
↪   motor vehicles & motorcycles",
"G46",          44L,                     "Wholesale trade, except of motor
↪   vehicles & motorcycles",
"G47",          45L,                     "Retail trade, except of motor
↪   vehicles & motorcycles",
"H49",          46L,                     "Land transport & transport via
↪   pipelines",
"H50",          47L,                     "Water transport",
"H51",          48L,                     "Air transport",
"H52",          49L,                     "Warehousing & support activities for
↪   transportation",
"H53",          50L,                     "Postal and courier activities",
"I55",          51L,                     "Accommodation",
```

```
"I56",         52L,                  "Food and beverage service
 ↪ activities",
"J58",         53L,                  "Publishing activities",
"J59",         54L,                  "Motion picture, video & television
 ↪ programme production, sound recording & music publishing",
"J60",         55L,                  "Programming and broadcasting
 ↪ activities",
"J61",         56L,                  "Telecommunications",
"J62",         57L,                  "Computer programming, consultancy &
 ↪ related activities",
"J63",         58L,                  "Information service activities",
"K64",         59L,                  "Financial service activities, except
 ↪ insurance & pension funding",
"K65",         60L,                  "Insurance, reinsurance & pension
 ↪ funding, except compulsory social security",
"K66",         61L,                  "Activities auxiliary to financial
 ↪ service & insurance activities",
"L68",         62L,                  "Real estate activities",
"M69",         63L,                  "Legal and accounting activities",
"M70",         64L,                  "Activities of head offices;
 ↪ management consultancy activities",
"M71",         65L,                  "Architectural & engineering
 ↪ activities; technical testing & analysis",
"M72",         66L,                  "Scientific research & development",
"M73",         67L,                  "Advertising and market research",
"M74",         68L,                  "Other professional, scientific &
 ↪ technical activities",
"M75",         69L,                  "Veterinary activities",
"N77",         70L,                  "Rental and leasing activities",
"N78",         71L,                  "Employment activities",
"N79",         72L,                  "Travel agency, tour operator,
 ↪ reservation service & related activities",
"N80",         73L,                  "Security and investigation
 ↪ activities",
"N81",         74L,                  "Services to buildings & landscape
 ↪ activities",
"N82",         75L,                  "Office administrative, office support
 ↪ & other business support activities",
"O84",         76L,                  "Public administration & defence;
 ↪ compulsory social security",
"P85",         77L,                  "Education",
"Q86",         78L,                  "Human health activities",
```

```
  "Q87",         79L,                   "Residential care activities",
  "Q88",         80L,                   "Social work activities without
   ↪ accommodation",
  "R90",         81L,                   "Creative, arts & entertainment
   ↪ activities",
  "R91",         82L,                   "Libraries, archives, museums & other
   ↪ cultural activities",
  "R92",         83L,                   "Gambling and betting activities",
  "R93",         84L,                   "Sports activities & amusement &
   ↪ recreation activities",
  "S94",         85L,                   "Activities of membership
   ↪ organizations",
  "S95",         86L,                   "Repair of computers & personal &
   ↪ household goods",
  "S96",         87L,                   "Other personal service activities",
  "T97",         88L,                   "Activities of households as employers
   ↪ of domestic personnel",
  "T98",         89L,                   "Undifferentiated goods- &
   ↪ services-producing activities of households for own use",
  "U99",         90L,                   "Activities of extraterritorial
   ↪ organizations & bodies"
)

# 2) Vérification
rev_isic2
```

### 2.2 Top 5 secteurs verts vs non-verts par groupe de revenus

```
library(dplyr)
library(knitr)

# Totaux par groupe et statut (vert/non-vert)
totaux <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5) %>%
  mutate(statut = if_else(EU_ISIC4_2dg == 2, "Vert", "Non-vert")) %>%
  group_by(income_group, statut) %>%
  summarise(grand_total = sum(value, na.rm = TRUE), .groups = "drop")

# Effectifs par secteur (on exclut TOTAL et X via isic_code_2dg)
secteurs <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
```

```r
  filter(sex == 1, edu == 5) %>%
  mutate(statut = if_else(EU_ISIC4_2dg == 2, "Vert", "Non-vert")) %>%
  filter(! isic_code_2dg %in% c("TOTAL", "X")) %>%
  group_by(income_group, statut, isic_code_2dg, isic_code_2dg_fact) %>%
  summarise(sect_total = sum(value, na.rm = TRUE), .groups = "drop")

# Calcul de la part et top 5 par groupe & statut
top_secteurs <- secteurs %>%
  left_join(totaux, by = c("income_group", "statut")) %>%
  mutate(prop = 100 * sect_total / grand_total) %>%
  arrange(income_group, statut, desc(prop)) %>%
  group_by(income_group, statut) %>%
  slice_head(n = 5) %>%
  ungroup() %>%
  mutate(
    Groupe = dplyr::recode(
      as.character(income_group),
      `1` = "High income",
      `2` = "Low income",
      `3` = "Lower middle income",
      `4` = "Upper middle income"
    )
  ) %>%
  dplyr::select(
    Groupe,
    Statut          = statut,
    ISIC_orig       = isic_code_2dg,
    ISIC_fact       = isic_code_2dg_fact,
    `Total groupe` = grand_total,
    `Total secteur`= sect_total,
    `% secteur`     = prop
  )

# Résultat
knitr::kable(
  top_secteurs,
  digits   = c(0,0,0,0,0,0,0,1),
  caption = "Top 5 secteurs « FPV » et « nFPV » par groupe de revenu (sans
  ↪ TOTAL ni X)"
)
```

Tableau. Top 5 secteurs "FPV" et "nFPV" par groupe de revenu

| Groupe | Statut | ISIC_-orig | ISIC_-fact | Intitulé secteur | Total groupe | Total secteur | % secteur |
|---|---|---|---|---|---|---|---|
| High income | Non-vert | I56 | 52 | Food and beverage service activities | 6 708 719 | 260 645 | 3.9 |
| High income | Non-vert | G45 | 43 | Wholesale and retail trade & repair of motor vehicles & motor-cycles | 6 708 719 | 108 392 | 1.6 |
| High income | Non-vert | K64 | 59 | Financial service activities, except insurance & pension funding | 6 708 719 | 102 828 | 1.5 |
| High income | Non-vert | M69 | 63 | Legal and ac-counting activities | 6 708 719 | 93 264 | 1.4 |
| High income | Non-vert | N81 | 74 | Services to buildings & land-scape activities | 6 708 719 | 93 034 | 1.4 |
| High income | Vert | G47 | 45 | Retail trade, except of motor vehicles & motor-cycles | 3 417 915 | 437 933 | 12.8 |
| High income | Vert | P85 | 77 | Education | 3 417 915 | 437 858 | 12.8 |

| Groupe | Statut | ISIC_-orig | ISIC_-fact | Intitulé secteur | Total groupe | Total secteur | % secteur |
|---|---|---|---|---|---|---|---|
| High income | Vert | Q86 | 78 | Human health activities | 3 417 915 | 413 939 | 12.1 |
| High income | Vert | O84 | 76 | Public administration & defence; compulsory social security | 3 417 915 | 244 881 | 7.2 |
| High income | Vert | F43 | 42 | Specialized construction activities | 3 417 915 | 164 735 | 4.8 |
| Low income | Non-vert | A01 | 3 | Crop and animal production, hunting and related service activities | 700 084 | 237 957 | 34.0 |
| Low income | Non-vert | T97 | 88 | Activities of households as employers of domestic personnel | 700 084 | 7 364 | 1.1 |
| Low income | Non-vert | S96 | 87 | Other personal service activities | 700 084 | 6 885 | 1.0 |
| Low income | Non-vert | I56 | 52 | Food and beverage service activities | 700 084 | 4 151 | 0.6 |

| Groupe | Statut | ISIC_-orig | ISIC_-fact | Intitulé secteur | Total groupe | Total secteur | % secteur |
|---|---|---|---|---|---|---|---|
| Low income | Non-vert | T98 | 89 | Undifferentiated goods- & services-producing activities of households | 710 084 | 4 070 | 0.6 |
| Low income | Vert | G47 | 45 | Retail trade, except of motor vehicles & motorcycles | 90 861 | 31 055 | 34.2 |
| Low income | Vert | P85 | 77 | Education | 90 861 | 9 685 | 10.7 |
| Low income | Vert | F41 | 40 | Construction of buildings | 90 861 | 8 351 | 9.2 |
| Low income | Vert | H49 | 46 | Land transport & transport via pipelines | 90 861 | 7 269 | 8.0 |
| Low income | Vert | H52 | 49 | Warehousing & support activities for transportation | 90 861 | 5 695 | 6.3 |
| Lower middle income | Non-vert | A01 | 3 | Crop and animal production, hunting and related service activities | 10 045 924 | 2 474 932 | 24.6 |

| Groupe | Statut | ISIC_-orig | ISIC_-fact | Intitulé secteur | Total groupe | Total secteur | % secteur |
|---|---|---|---|---|---|---|---|
| Lower middle income | Non-vert | C14 | 15 | Manufacture of wearing apparel | 10 045 924 | 171 359 | 1.7 |
| Lower middle income | Non-vert | I56 | 52 | Food and beverage service activities | 10 045 924 | 141 503 | 1.4 |
| Lower middle income | Non-vert | C10 | 11 | Manufacture of food products | 10 045 924 | 98 655 | 1.0 |
| Lower middle income | Non-vert | S96 | 87 | Other personal service activities | 10 045 924 | 88 249 | 0.9 |
| Lower middle income | Vert | G47 | 45 | Retail trade, except of motor vehicles & motor-cycles | 2 564 210 | 631 777 | 24.6 |
| Lower middle income | Vert | F41 | 40 | Construction of buildings | 2 564 210 | 454 277 | 17.7 |
| Lower middle income | Vert | H49 | 46 | Land transport & transport via pipelines | 2 564 210 | 270 134 | 10.5 |
| Lower middle income | Vert | P85 | 77 | Education | 2 564 210 | 235 270 | 9.2 |

| Groupe | Statut | ISIC_-orig | ISIC_-fact | Intitulé secteur | Total groupe | Total secteur | % secteur |
|---|---|---|---|---|---|---|---|
| Lower middle income | Vert | O84 | 76 | Public administration & defence; compulsory social security | 2 564 210 | 147 114 | 5.7 |
| Upper middle income | Non-vert | A01 | 3 | Crop and animal production, hunting and related service activities | 7 630 294 | 899 829 | 11.8 |
| Upper middle income | Non-vert | I56 | 52 | Food and beverage service activities | 7 630 294 | 238 798 | 3.1 |
| Upper middle income | Non-vert | T97 | 88 | Activities of households as employers of domestic personnel | 7 630 294 | 168 982 | 2.2 |
| Upper middle income | Non-vert | C10 | 11 | Manufacture of food products | 7 630 294 | 130 873 | 1.7 |

| Groupe | Statut | ISIC_-orig | ISIC_-fact | Intitulé secteur | Total groupe | Total secteur | % secteur |
|---|---|---|---|---|---|---|---|
| Upper middle income | Non-vert | G45 | 43 | Wholesale and retail trade & repair of motor vehicles & motorcycles | 7 630 294 | 124 608 | 1.6 |
| Upper middle income | Vert | G47 | 45 | Retail trade, except of motor vehicles & motorcycles | 2 700 269 | 630 838 | 23.4 |
| Upper middle income | Vert | P85 | 77 | Education | 2 700 269 | 267 439 | 9.9 |
| Upper middle income | Vert | O84 | 76 | Public administration & defence; compulsory social security | 2 700 269 | 246 776 | 9.1 |
| Upper middle income | Vert | F41 | 40 | Construction of buildings | 2 700 269 | 224 505 | 8.3 |
| Upper middle income | Vert | H49 | 46 | Land transport & transport via pipelines | 2 700 269 | 198 184 | 7.3 |

La colonne « **% secteur** » correspond à la part (en pourcentage) que représente chaque secteur **parmi l'ensemble des emplois de ce même groupe de revenu et de ce même**

**statut (Vert vs Non-vert)**. Concrètement, pour un couple « groupe de revenu + statut » donné (par exemple « High income – Vert »), on calcule d'abord le **total des emplois** (toutes branches confondues) : c'est la colonne « Total groupe ». Puis, pour chaque secteur ISIC, on calcule le **nombre d'emplois** qui lui revient : c'est la colonne « Total secteur ». Enfin, on prend **100 × (Total secteur) / (Total groupe)**, ce qui donne la proportion (%) de l'emploi de ce secteur parmi tous les emplois de ce groupe/status.

## 2.3 Calcule d'un indice de spécialisation (LQ/RCA)

L'**indice de localisation quotien (LQ)** ou « revealed comparative advantage » (RCA) mesure pour chaque pays et chaque secteur si ce pays est « sur-spécialisé » dans ce secteur, relativement au monde :

$$\mathrm{LQ}_{c,s} = \frac{\frac{E_{c,s}}{E_{c,\bullet}}}{\frac{E_{\bullet,s}}{E_{\bullet,\bullet}}}$$

- (E_{c,s}) : nombre d'emplois du pays (c) dans le secteur (s)
- (E_{c, • } = *{s'} E*{c,s'}) : effectif total de tous les secteurs dans le pays (c)
- (E_{ • ,s} = *{c'} E*{c',s}) : effectif total mondial dans le secteur (s)
- (E_{ • , • } = *{c',s'} E*{c',s'}) : effectif total mondial sur tous les secteurs

  **Note**
  Le symbole "( • )" dans un indice signifie que l'on somme sur toutes les valeurs de cet indice (tous les secteurs ou tous les pays).
  - (E_{c, • }) somme sur tous les secteurs de (c).
  - (E_{ • ,s}) somme sur tous les pays pour le secteur (s).
  - (E_{ • , • }) somme sur tous les pays et tous les secteurs.

Un LQ > 1 signifie "spécialisation", un LQ < 1 "sous-spécialisation" :

```
if ("package:plyr" %in% search()) {
  detach("package:plyr", unload = TRUE)
}
library(dplyr)
library(tidyr)
library(knitr)
```

```r
# Préparation de la base et calcul des LQ
df <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(sex == 1, edu == 5, isic_code_2dg_code > 2) %>%
  mutate(statut = if_else(EU_ISIC4_2dg == 2, "Vert", "Non-vert"))

Es <- df %>%
  group_by(iso3c, isic_code_2dg, statut) %>%
  summarise(Ecs = sum(value, na.rm = TRUE), .groups = "drop")

Ec_ <- Es %>%
  group_by(iso3c, statut) %>%
  summarise(Ec_ = sum(Ecs), .groups = "drop")

E_s <- Es %>%
  group_by(isic_code_2dg, statut) %>%
  summarise(E_s = sum(Ecs), .groups = "drop")

E__ <- Es %>%
  group_by(statut) %>%
  summarise(E__ = sum(Ecs), .groups = "drop")

LQ <- Es %>%
  left_join(Ec_, by = c("iso3c", "statut")) %>%
  left_join(E_s, by = c("isic_code_2dg", "statut")) %>%
  left_join(E__, by = "statut") %>%
  mutate(LQ = (Ecs / Ec_) / (E_s / E__)) %>%
  dplyr::select(iso3c, isic_code_2dg, statut, LQ)

# Ajout groupe de revenu
labels_inc <- c(
  "1" = "High income",
  "2" = "Low income",
  "3" = "Lower middle income",
  "4" = "Upper middle income"
)

countries <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  dplyr::distinct(iso3c, income_group)

LQ <- LQ %>%
  dplyr::left_join(countries, by = "iso3c") %>%
  dplyr::mutate(
    income_group = as.character(income_group),
```

```
    # on force bien la version dplyr de recode()
    groupe       = dplyr::recode(income_group, !!!labels_inc)
  ) %>%
  dplyr::select(iso3c, groupe, isic_code_2dg, statut, LQ)


# Agrégat LQ par secteur au sein de chaque (groupe, statut)
LQ_grp <- LQ %>%
  filter(!is.na(groupe)) %>%
  group_by(groupe, statut, isic_code_2dg) %>%
  summarise(
    LQ_moyenne = mean(LQ, na.rm = TRUE),
    .groups     = "drop"
  )

# Top 10 secteurs les plus « spécialisés » (LQ_moyenne décroissante)
top_specialises <- LQ_grp %>%
  group_by(groupe, statut) %>%
  arrange(desc(LQ_moyenne)) %>%
  slice_head(n = 10) %>%
  mutate(rang = row_number()) %>%
  ungroup()

# Top 10 secteurs les moins « spécialisés » (LQ_moyenne croissante, > 0)
bottom_specialises <- LQ_grp %>%
  filter(LQ_moyenne > 0) %>%
  group_by(groupe, statut) %>%
  arrange(LQ_moyenne) %>%
  slice_head(n = 10) %>%
  mutate(rang = row_number()) %>%
  ungroup()

# Fusion finale pour affichage
tableau <- top_specialises %>%
  dplyr::select(
    groupe,
    statut,
    rang,
    `Top Secteur` = isic_code_2dg,
    `LQ Top`      = LQ_moyenne
  ) %>%
  full_join(
    bottom_specialises %>%
```

```r
      dplyr::select(
        groupe,
        statut,
        rang,
        `Bottom Secteur` = isic_code_2dg,
        `LQ Bottom`      = LQ_moyenne
      ),
    by = c("groupe", "statut", "rang")
  ) %>%
  dplyr::arrange(groupe, statut, rang)

# Résultat
knitr::kable(
  tableau,
  caption = "Top 10 et Bottom 10 LQ moyens par groupe de revenu et statut",
  digits  = c(0, 0, 0, 0, 4, 0, 4)
)
```

Tableau. Top 10 et Bottom 10 des LQ par groupe de revenu et statu (FPV vs nFPV) - High income et Low income

| groupe | statut | rang | Top Secteur | LQ Top | Bottom Secteur | LQ Bottom |
|---|---|---|---|---|---|---|
| High income | Non-vert | 1 | Activités indifférenciées des ménages | 14.9788 | Industrie du tabac | 0.1815 |
| High income | Non-vert | 2 | Extraction de pétrole brut et de gaz naturel | 5.8210 | Production végétale et animale, chasse et activités annexes | 0.2291 |
| High income | Non-vert | 3 | Services de soutien aux industries extractives | 5.3882 | Industrie de l'habillement | 0.4054 |
| High income | Non-vert | 4 | Autres industries extractives et carrières | 5.1328 | Industrie textile | 0.4229 |

| groupe | statut | rang | Top Secteur | LQ Top | Bottom Secteur | LQ Bottom |
|---|---|---|---|---|---|---|
| High income | Non-vert | 5 | Agence de voyages et services de réservation | 4.0582 | Industrie du cuir et industries connexes | 0.7588 |
| High income | Non-vert | 6 | Activités des organisations extraterritoriales | 3.8857 | Autres industries manufacturières | 1.0735 |
| High income | Non-vert | 7 | Extraction de minerais métalliques | 3.7879 | Fabrication de meubles | 1.0840 |
| High income | Non-vert | 8 | Activités des sièges sociaux et conseil de gestion | 3.3408 | Pêche et aquaculture | 1.2049 |
| High income | Non-vert | 9 | Services de sécurité | 3.3234 | Autres activités de services personnels | 1.4003 |
| High income | Non-vert | 10 | Publicité et études de marché | 3.2997 | Activités vétérinaires | 1.4404 |
| High income | Vert | 1 | Hébergement | 4.2018 | Manufacture d'autres produits minéraux non métalliques | 0.5283 |
| High income | Vert | 2 | Transport maritime | 2.5396 | Transport terrestre et transport par conduites | 0.5590 |

| groupe | statut | rang | Top Secteur | LQ Top | Bottom Secteur | LQ Bottom |
|---|---|---|---|---|---|---|
| High income | Vert | 3 | Administration publique et défense; sécurité sociale obligatoire | 2.0233 | Réparation d'ordinateurs et d'articles personnels | 0.6194 |
| High income | Vert | 4 | Transport aérien | 1.7340 | Construction de bâtiments | 0.6558 |
| High income | Vert | 5 | Collecte, traitement et distribution d'eau | 1.6116 | Génie civil | 0.6750 |
| High income | Vert | 6 | Fabrication de produits informatiques, électroniques et optiques | 1.5700 | Sylviculture et exploitation forestière | 0.7132 |
| High income | Vert | 7 | Programmation et diffusion | 1.5268 | Commerce de détail, à l'exception des véhicules | 0.7339 |
| High income | Vert | 8 | Entreposage et activités de soutien aux transports | 1.5137 | Assainissement | 0.7835 |
| High income | Vert | 9 | Waste collection, treatment and disposal; materials recovery | 1.5119 | Activités postales et de courrier | 0.7846 |

| groupe | statut | rang | Top Secteur | LQ Top | Bottom Secteur | LQ Bottom |
|---|---|---|---|---|---|---|
| High income | Vert | 10 | Recherche-développement scientifique | 1.4322 | Industrie chimique | 0.8493 |
| Low income | Non-vert | 1 | Activités indifférenciées des ménages | 11.0069 | Activités auxiliaires des services financiers et d'assurance | 0.0706 |
| Low income | Non-vert | 2 | Activités des organisations extraterritoriales | 4.5375 | Industrie du tabac | 0.1066 |
| Low income | Non-vert | 3 | Extraction de minerais métalliques | 3.5971 | Activités auxiliaires des services financiers et d'assurance | 0.1167 |
| Low income | Non-vert | 4 | Industrie des boissons | 2.4367 | Activités des sièges sociaux et conseil de gestion | 0.1372 |
| Low income | Non-vert | 5 | Extraction de houille et lignite | 2.2497 | Activités juridiques et comptables | 0.1386 |
| Low income | Non-vert | 6 | Autres industries extractives et carrières | 2.1252 | Activités d'édition | 0.1418 |

| groupe | statut | rang | Top Secteur | LQ Top | Bottom Secteur | LQ Bottom |
|---|---|---|---|---|---|---|
| Low income | Non-vert | 7 | Production végétale et animale, chasse et activités annexes | 1.5629 | Agence de voyages et services de réservation | 0.1610 |
| Low income | Non-vert | 8 | Autres activités de services personnels | 1.4656 | Fabrication d'autres matériels de transport | 0.1716 |
| Low income | Non-vert | 9 | Services de sécurité | 1.0970 | Autres industries manufacturières | 0.1745 |
| Low income | Non-vert | 10 | Industrie du bois et articles en bois | 1.0627 | Publicité et études de marché | 0.1773 |
| Low income | Vert | 1 | Sylviculture et exploitation forestière | 3.5244 | Activités postales et de courrier | 0.1231 |
| Low income | Vert | 2 | Entreposage et activités de soutien aux transports | 3.0743 | Activités d'architecture et d'ingénierie | 0.1294 |
| Low income | Vert | 3 | Transport maritime | 2.6356 | Action sociale sans hébergement | 0.2053 |
| Low income | Vert | 4 | Programmation et diffusion | 2.2933 | Fabrication de véhicules automobiles | 0.2536 |

| groupe | statut | rang | Top Secteur | LQ Top | Bottom Secteur | LQ Bottom |
|---|---|---|---|---|---|---|
| Low income | Vert | 5 | Collecte, traitement et distribution d'eau | 2.2895 | Programmation, conseil et activités informatiques connexes | 0.3016 |
| Low income | Vert | 6 | Bibliothèques, archives, musées et autres activités culturelles | 2.2482 | Activités d'hébergement médico-social | 0.3195 |
| Low income | Vert | 7 | Assainissement | 2.2385 | Activités immobilières | 0.3284 |
| Low income | Vert | 8 | Commerce de détail, à l'exception des véhicules | 2.0217 | Fabrication de machines et équipements | 0.3867 |
| Low income | Vert | 9 | Activités de gamb… | …continues pour chaque combinaison. | | |

```
if ("package:plyr" %in% search()) detach("package:plyr", unload = TRUE)
library(dplyr); library(tidyr); library(ggplot2); library(forcats);
↪  library(knitr)

# Codes -> libellés
secteur_labels <- c(
  "A01" = "Production végétale et animale, chasse",
  "A02" = "Sylviculture et exploitation forestière",
  "A03" = "Pêche et aquaculture",
  "B05" = "Extraction de houille et lignite",
  "B06" = "Extraction de pétrole brut et gaz naturel",
  "B07" = "Extraction de minerais métalliques",
  "B08" = "Autres industries extractives et carrières",
```

```
"B09" = "Services de soutien aux industries extractives",
"C10" = "Industrie agroalimentaire",
"C11" = "Fabrication de boissons",
"C12" = "Fabrication de produits du tabac",
"C13" = "Fabrication textile",
"C14" = "Fabrication d'habillement",
"C15" = "Fabrication de cuir et articles connexes",
"C16" = "Industrie du bois et du liège",
"C17" = "Fabrication de papier et produits dérivés",
"C18" = "Impression et reproduction d'enregistrements",
"C19" = "Fabrication de coke et produits pétroliers raffinés",
"C20" = "Industrie chimique",
"C21" = "Production pharmaceutique de base",
"C22" = "Fabrication de produits en caoutchouc et plastiques",
"C23" = "Fabrication de produits minéraux non métalliques",
"C24" = "Fabrication de métaux de base",
"C25" = "Fabrication de produits métalliques",
"C26" = "Fabrication de matériel informatique, électronique et optique",
"C27" = "Fabrication d'équipements électriques",
"C28" = "Fabrication de machines et équipements",
"C29" = "Fabrication de véhicules automobiles",
"C30" = "Fabrication d'autres matériels de transport",
"C31" = "Fabrication de meubles",
"C32" = "Autres industries manufacturières",
"C33" = "Réparation et installation de machines",
"D35" = "Production/distribution d'énergie et climatisation",
"E36" = "Collecte, traitement et distribution d'eau",
"E37" = "Assainissement",
"E38" = "Collecte et élimination des déchets",
"E39" = "Dépollution et gestion des déchets",
"F41" = "Construction de bâtiments",
"F42" = "Génie civil",
"F43" = "Activités spécialisées de construction",
"G45" = "Commerce et réparation de véhicules",
"G46" = "Commerce de gros (hors véhicules)",
"G47" = "Commerce de détail (hors véhicules)",
"H49" = "Transport terrestre et par pipelines",
"H50" = "Transport maritime",
"H51" = "Transport aérien",
"H52" = "Entreposage et activités de soutien aux transports",
"H53" = "Activités postales et de courrier",
"I55" = "Hébergement",
"I56" = "Restauration",
```

```r
  "J58" = "Édition",
  "J59" = "Production audiovisuelle et musicale",
  "J60" = "Programmation et diffusion radio/TV",
  "J61" = "Télécommunications",
  "J62" = "Services informatiques et conseil",
  "J63" = "Services d'information",
  "K64" = "Services financiers hors assurance",
  "K65" = "Assurance et caisses de retraite",
  "K66" = "Services auxiliaires financiers",
  "L68" = "Activités immobilières",
  "M69" = "Activités juridiques et comptables",
  "M70" = "Sièges sociaux et conseil en gestion",
  "M71" = "Architecture, ingénierie et essais techniques",
  "M72" = "Recherche-développement scientifique",
  "M73" = "Publicité et études de marché",
  "M74" = "Autres activités scientifiques et techniques",
  "M75" = "Activités vétérinaires",
  "N77" = "Location et location-bail",
  "N78" = "Activités liées à l'emploi",
  "N79" = "Agences de voyage et services de réservation",
  "N80" = "Sécurité et investigations",
  "N81" = "Services aux bâtiments et aménagement paysager",
  "N82" = "Services administratifs et de soutien",
  "O84" = "Administration publique et défense",
  "P85" = "Enseignement",
  "Q86" = "Activités de santé humaine",
  "Q87" = "Hébergement médico-social",
  "Q88" = "Action sociale sans hébergement",
  "R90" = "Arts, spectacles et loisirs",
  "R91" = "Musées, bibliothèques et activités culturelles",
  "R92" = "Jeux de hasard et paris",
  "R93" = "Sports et loisirs récréatifs",
  "S94" = "Activités associatives",
  "S95" = "Réparation d'ordinateurs et articles personnels",
  "S96" = "Autres services personnels",
  "T97" = "Ménages employeurs",
  "T98" = "Activités des ménages pour usage propre",
  "U99" = "Organisations extraterritoriales"
)

# Préparation des données pour heatmap
heatmap_base <- LQ %>%
  dplyr::mutate(
```

```
    groupe = dplyr::recode_factor(
      groupe,
      "High income"         = "Revenu élevé",
      "Low income"          = "Revenu faible",
      "Lower middle income" = "Revenu moyen inférieur",
      "Upper middle income" = "Revenu moyen supérieur"
    ),
    groupe = factor(
      groupe,
      levels = c(
        "Revenu élevé",
        "Revenu faible",
        "Revenu moyen inférieur",
        "Revenu moyen supérieur"
      )
    )
  ) %>%
  dplyr::filter(!is.na(groupe)) %>%
  dplyr::group_by(groupe, statut, isic_code_2dg) %>%
  dplyr::summarise(med_LQ = median(LQ, na.rm = TRUE), .groups = "drop") %>%
  dplyr::mutate(secteur = secteur_labels[isic_code_2dg])

# Wide --> long
make_hm_df <- function(df, stat) {
  df %>%
    dplyr::filter(statut == stat) %>%
    dplyr::select(groupe, secteur, med_LQ) %>%
    tidyr::pivot_wider(names_from = groupe, values_from = med_LQ) %>%
    tidyr::pivot_longer(-secteur, names_to="groupe", values_to="med_LQ")
}

hm_vert    <- make_hm_df(heatmap_base, "Vert")
hm_nonvert <- make_hm_df(heatmap_base, "Non-vert")

# Grapjhique
plot_hm <- function(hm_df, title, low_col, high_col) {
  ggplot(hm_df, aes(
      x = groupe,
      y = forcats::fct_rev(secteur),
      fill = med_LQ
    )) +
    geom_tile(color = "white") +
    scale_fill_gradient(low = low_col, high = high_col, na.value = "grey90")
     ↪  +
```

```r
    labs(title = title, x = NULL, y = NULL) +
    theme_minimal() +
    theme(
      axis.text.x = element_text(angle = 45, hjust = 1),
      panel.grid = element_blank()
    )
}

p_vert    <- plot_hm(hm_vert,    "Median RCA - EFPV",   "lightgreen",
 ↪  "darkgreen")
p_nonvert <- plot_hm(hm_nonvert, "Median RCA - EnFPV",  "navajowhite",
 ↪  "saddlebrown")

print(p_vert)
print(p_nonvert)

# Téléchargement
out_dir <- "/Users/.../Chap2/graphiques_rd2"
if (!dir.exists(out_dir)) dir.create(out_dir, recursive = TRUE)

ggsave(
  filename = file.path(out_dir, "heatmap_median_LQ_verts.png"),
  plot     = p_vert,
  width    = 8,
  height   = 10,
  dpi      = 300,
  bg       = "white"
)

ggsave(
  filename = file.path(out_dir, "heatmap_median_LQ_non_verts.png"),
  plot     = p_nonvert,
  width    = 8,
  height   = 10,
  dpi      = 300,
  bg       = "white"
)
```

## 3 Profils des travailleurs en EFPV

### 3.1 Sex (with edu == 5)

```
library(dplyr)
library(knitr)
library(purrr)
library(tidyr)
library(broom)
library(rcompanion)   # pour cramerV

# Préparation des données : on ne garde que edu == 5, sex = 2/3 et
↪  EU_ISIC4_2dg = 1/2
df2 <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(
    edu %in% 5,
    sex %in% c(2,3),
    EU_ISIC4_2dg %in% c(1,2)
  ) %>%
  mutate(
    sex      = recode_factor(sex, `2`="Male", `3`="Female"),
    is_green = (EU_ISIC4_2dg == 2)
  )

# Groupe par income_group et faire l'analyse dans chaque sous-jeu
results <- df2 %>%
  dplyr::group_by(income_group) %>%
  tidyr::nest() %>%
  dplyr::mutate(
    tab      = purrr::map(data, ~ xtabs(value ~ sex + is_green, data = .x)),
    chi      = purrr::map(tab, chisq.test),
    glance   = purrr::map(chi, broom::glance),
    cramersV = purrr::map_dbl(tab, ~ cramerV(.x))
  ) %>%
  tidyr::unnest(cols = glance) %>%
  dplyr::select(
    income_group,
    statistic,
    p_value = p.value,
    cramersV
  )
```

```r
print(results)

# Calcule la part d'EFPV par sexe dans chaque groupe
shares <- df2 %>%
  dplyr::group_by(income_group, sex, is_green) %>%
  dplyr::summarise(jobs = sum(value, na.rm=TRUE), .groups="drop") %>%
  dplyr::group_by(income_group, sex) %>%
  dplyr::mutate(pct_green = jobs / sum(jobs)) %>%
  dplyr::filter(is_green) %>%
  dplyr::select(income_group, sex, pct_green) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(
    income_group = factor(income_group,
                          levels = c(1,2,3,4),
                          labels = c("High income",
                                     "Low income",
                                     "Lower middle",
                                     "Upper middle")),
    sex = recode_factor(sex,
                        `Male`   = "Homme",
                        `Female` = "Femme")
  )

# Vérification des valeurs non arrondies pour "High income"
shares %>%
  filter(income_group == "High income") %>%
  print(digits = 4)

# Résultat
knitr::kable(
  shares,
  col.names = c("Groupe de revenu", "Sexe", "Part verte (%)"),
  digits    = c(NA, NA, 4),
  caption   = "Part des emplois verts, par sexe et par groupe de revenu (edu
   ↪  = TOTAL)"
)
```

## 3.2 Éducation

```r
EMP_TEMP_SEX_EDU_EC2_NB_A <-
 ↪  readRDS("/Users/.../Chap2/new_SAGE/EMP_TEMP_SEX_EDU_EC2_NB_A.rds")
```

```r
library(dplyr)
library(rstatix)        # kruskal_test(), kruskal_effsize(), dunn_test()
library(DescTools)      # JonckheereTerpstraTest()
library(knitr)

# Préparation des données
df_country_prop <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(
    sex == 1,
    edu %in% 1:4,
    EU_ISIC4_2dg %in% c(1,2)
  ) %>%
  group_by(iso3c, year, income_group, edu) %>%
  summarise(
    total_jobs = sum(value, na.rm=TRUE),
    green_jobs = sum(value * (EU_ISIC4_2dg == 2), na.rm=TRUE),
    .groups = "drop"
  ) %>%
  mutate(prop_green = green_jobs / total_jobs) %>%
  group_by(iso3c, income_group, edu) %>%
  summarise(
    country_prop = mean(prop_green, na.rm=TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("High", "Low", "Lower mid", "Upper
                           ↪  mid")),
    edu = factor(edu,
                 levels = 1:4,
                 labels = c("LTB", "BAS", "INT", "ADV"))
  )

# Kruskal-Wallis +  ² par groupe de revenu
kw_stats <- df_country_prop %>%
  group_by(income_group) %>%
  kruskal_test(country_prop ~ edu)

kw_eta2 <- df_country_prop %>%
  group_by(income_group) %>%
```

```
  kruskal_effsize(country_prop ~ edu) %>%
  rename(eta2 = effsize)

kw_results <- kw_stats %>%
  left_join(select(kw_eta2, income_group, eta2), by = "income_group") %>%
  rename(
    Groupe    = income_group,
    `H (KW)`  = statistic,
    `p-KW`    = p
  ) %>%
  select(Groupe, `H (KW)`, `p-KW`, eta2)

# Jonckheere-Terpstra
jt_results <- df_country_prop %>%
  group_by(income_group) %>%
  summarise(
    `JT stat.` = JonckheereTerpstraTest(
                   country_prop,
                   as.numeric(edu),
                   nperm = 5000
                 )$statistic,
    `p-JT`     = JonckheereTerpstraTest(
                   country_prop,
                   as.numeric(edu),
                   nperm = 5000
                 )$p.value,
    .groups = "drop"
  ) %>%
  rename(Groupe = income_group)

# Fusionner KW + JT
trend_tab <- kw_results %>%
  left_join(jt_results, by = "Groupe")

# Dunn post-hoc (Holm) par groupe de revenu
dunn_tab <- df_country_prop %>%
  group_by(income_group) %>%
  dunn_test(country_prop ~ edu,
            p.adjust.method = "holm") %>%
  select(
    Groupe    = income_group,
    `Édu. 1`  = group1,
    `Édu. 2`  = group2,
```

```
    `p-value` = p.adj
  )

# Résultats
# a) KW +  ² + JT
kable(
  trend_tab,
  digits    = c(NA, 2, 4, 3, 4, 4),
  col.names = c("Groupe", "H (KW)", "p-KW", " ²", "JT stat.", "p-JT"),
  caption   = "Kruskal-Wallis (H, p), taille d'effet  ² et Jonckheere-Terpstra par groupe de revenu"
)

# b) Comparaisons post-hoc Dunn
kable(
  dunn_tab,
  digits    = 4,
  col.names = c("Groupe", "Édu. 1", "Édu. 2", "p-value Holm"),
  caption   = "Comparaisons post-hoc Dunn (Holm) par groupe de revenu"
)
```

```
library(dplyr)
library(rstatix)       # kruskal_test(), kruskal_effsize(), dunn_test()
library(DescTools)     # JonckheereTerpstraTest()
library(ggplot2)
library(scales)
library(knitr)

# Préparation des données (EFPV) - df_country_prop
df_country_prop <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(
    sex == 1,
    edu %in% 1:4,
    EU_ISIC4_2dg %in% c(1,2)
  ) %>%
  group_by(iso3c, year, income_group, edu) %>%
  summarise(
    total_jobs = sum(value, na.rm = TRUE),
    green_jobs = sum(value * (EU_ISIC4_2dg == 2), na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(prop_green = green_jobs / total_jobs) %>%
```

```
  group_by(iso3c, income_group, edu) %>%
  summarise(
    country_prop = mean(prop_green, na.rm = TRUE),
    .groups      = "drop"
  ) %>%
  mutate(
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("Revenu élevé","Revenu faible","Revenu
                          ↪  moyen inférieur","Revenu moyen supérieur")),
    edu = factor(edu,
                 levels = 1:4,
                 labels = c("LTB","BAS","INT","ADV"))
  )

# Préparation des données (EnFPV) - df_country_prop_ng
df_country_prop_ng <- EMP_TEMP_SEX_EDU_EC2_NB_A %>%
  filter(
    sex == 1,
    edu %in% 1:4,
    EU_ISIC4_2dg %in% c(1,2)
  ) %>%
  group_by(iso3c, year, income_group, edu) %>%
  summarise(
    total_jobs     = sum(value, na.rm = TRUE),
    green_jobs     = sum(value * (EU_ISIC4_2dg == 2), na.rm = TRUE),
    .groups        = "drop"
  ) %>%
  mutate(
    non_green_jobs  = total_jobs - green_jobs,
    prop_non_green  = non_green_jobs / total_jobs
  ) %>%
  group_by(iso3c, income_group, edu) %>%
  summarise(
    country_prop_ng = mean(prop_non_green, na.rm = TRUE),
    .groups         = "drop"
  ) %>%
  mutate(
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("Revenu élevé","Revenu faible","Revenu
                          ↪  moyen inférieur","Revenu moyen supérieur")),
    edu = factor(edu,
```

```
                 levels = 1:4,
                 labels = c("LTB","BAS","INT","ADV"))
  )

# Agrégation pour la heatmap
# a) Verts
edu_green_inc2 <- df_country_prop %>%
  group_by(income_group, edu) %>%
  summarise(prop_green = mean(country_prop, na.rm = TRUE),
            .groups = "drop")

# b) Non-verts
edu_ng_inc <- df_country_prop_ng %>%
  group_by(income_group, edu) %>%
  summarise(prop_non_green = mean(country_prop_ng, na.rm = TRUE),
            .groups = "drop")

# Résultats
knitr::kable(
  edu_green_inc2 %>% mutate(prop_green = round(prop_green * 100, 1)),
  col.names = c("Groupe de revenu","Éducation","% emplois verts"),
  caption   = "Part moyenne d'emplois VERTS par éducation et groupe de
   ↪  revenu"
)

knitr::kable(
  edu_ng_inc %>% mutate(prop_non_green = round(prop_non_green * 100, 1)),
  col.names = c("Groupe de revenu","Éducation","% emplois non-verts"),
  caption   = "Part moyenne d'emplois NON-VERTS par éducation et groupe de
   ↪  revenu"
)

# Heatmap : EFPV
ggplot(edu_green_inc2, aes(x = edu, y = income_group, fill = prop_green)) +
  geom_tile(color = "white", size = .2) +
  geom_text(aes(label = percent(prop_green, accuracy = 1)), size = 3) +
  scale_fill_gradient(
    low    = "#edf8fb",
    high   = "#006d2c",
    name   = "% EFPV",
    labels = percent_format(1)
  ) +
  labs(
```

```r
    x        = "Niveau d'éducation",
    y        = "Groupe de revenu",
    title    = "Heatmap : part moyenne d'EFPV",
    subtitle = "(moyenne par pays)"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x     = element_text(angle = 45, hjust = 1),
    panel.grid      = element_blank(),
    legend.position = "right"
  )

# Heatmap : EnFPV
ggplot(edu_ng_inc, aes(x = edu, y = income_group, fill = prop_non_green)) +
  geom_tile(color = "white", size = .2) +
  geom_text(aes(label = percent(prop_non_green, accuracy = 1)), size = 3) +
  scale_fill_gradient(
    low    = "#fff7ec",
    high   = "#7f0000",
    name   = "% EnFPV",
    labels = percent_format(1)
  ) +
  labs(
    x        = "Niveau d'éducation",
    y        = "Groupe de revenu",
    title    = "Heatmap : part moyenne d'EnFPV",
    subtitle = "(moyenne par pays)"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x     = element_text(angle = 45, hjust = 1),
    panel.grid      = element_blank(),
    legend.position = "right"
  )

# Téléchargement
# On recrée les objets de plot pour pouvoir les sauver
p_green <- ggplot(edu_green_inc2, aes(x = edu, y = income_group, fill =
↪   prop_green)) +
  geom_tile(color = "white", size = .2) +
  geom_text(aes(label = percent(prop_green, accuracy = 1)), size = 3) +
  scale_fill_gradient(
    low    = "#edf8fb",
```

```
    high   = "#006d2c",
    name   = "% EFPV",
    labels = percent_format(1)
  ) +
  labs(
    x        = "Niveau d'éducation",
    y        = "Groupe de revenu",
    title    = "Heatmap : part moyenne d'EFPV",
    subtitle = "(moyenne par pays)"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x     = element_text(angle = 45, hjust = 1),
    panel.grid      = element_blank(),
    legend.position = "right"
  )

p_non_green <- ggplot(edu_ng_inc, aes(x = edu, y = income_group, fill =
 ↪  prop_non_green)) +
  geom_tile(color = "white", size = .2) +
  geom_text(aes(label = percent(prop_non_green, accuracy = 1)), size = 3) +
  scale_fill_gradient(
    low    = "#fff7ec",
    high   = "#7f0000",
    name   = "% EnFPV",
    labels = percent_format(1)
  ) +
  labs(
    x        = "Niveau d'éducation",
    y        = "Groupe de revenu",
    title    = "Heatmap : part moyenne d'EnFPV",
    subtitle = "(moyenne par pays)"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x     = element_text(angle = 45, hjust = 1),
    panel.grid      = element_blank(),
    legend.position = "right"
  )

ggsave("/Users/.../Chap2/graphiques/heatmap_JHGP_edu.png", plot = p_green,
 ↪  width = 8, height = 6, dpi = 300, bg = "white")
ggsave("/Users/.../Chap2/graphiques/heatmap_JnHGP_edu.png", plot =
 ↪  p_non_green, width = 8, height = 6, dpi = 300, bg = "white")
```

## 3.3 Age

```
EMP_TEMP_AGE_EC2_NB_A <-
↪  readRDS("/Users/.../Chap2/EMP_TEMP_AGE_EC2_NB_A.rds")
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
library(rstatix)   # pour kruskal_effsize(), cramer_v()
library(effsize)   # pour cliff.delta()
library(lme4)
library(broom.mixed)
library(knitr)

# Préparation
df_age_brut <- EMP_TEMP_AGE_EC2_NB_A %>%
  filter(age %in% c(2,3), EU_ISIC4_2dg %in% c(1,2)) %>%
  mutate(
    age          = factor(age, levels = c(2,3),
                          labels = c("Y15-24","YGE25")),
    category     = factor(EU_ISIC4_2dg,
                          levels = c(1,2),
                          labels = c("Non-green","Green")),
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("High income","Low income",
                                     "Lower middle","Upper middle"))
  )

# Calcul des proportions par pays-année-groupe age/category
df_country_cat_prop <- df_age_brut %>%
  group_by(iso3c, year, income_group, age, category) %>%
  summarise(jobs = sum(value, na.rm=TRUE), .groups="drop") %>%
  group_by(iso3c, year, income_group) %>%
  mutate(prop = jobs / sum(jobs)) %>%    # proportion dans chaque pays-année
  ungroup() %>%
  group_by(iso3c, income_group, age, category) %>%
  summarise(country_prop = mean(prop, na.rm=TRUE), .groups="drop")
```

```r
# Tableau de parts moyennes par âge × groupe de revenu
table_shares <- df_country_cat_prop %>%
  group_by(income_group, age, category) %>%
  summarise(mean_share = mean(country_prop), .groups = "drop") %>%
  tidyr::pivot_wider(names_from = category, values_from = mean_share) %>%
  dplyr::mutate(
    `Non-green (%)` = percent(`Non-green`, accuracy = 1),
    `Green (%)`     = percent(Green,       accuracy = 1)
  ) %>%
  dplyr::select(
    `Groupe de revenu` = income_group,
    `Tranche d'âge`    = age,
    `Non-green (%)`,
    `Green (%)`
  )

# 3) Résultat
knitr::kable(
  table_shares,
  caption = "Part moyenne d'emplois non-verts vs verts par tranche d'âge\n(et par groupe de revenu, moyenne par pays)",
  align   = c("l","l","r","r")
)

# Tests de ² + V de Cramér sur effectifs bruts
age_chi2 <- df_age_brut %>%
  dplyr::group_by(income_group) %>%
  dplyr::summarise(
    ` ²` = {
      m <- xtabs(value ~ age + category, data = .)
      stats::chisq.test(m)$statistic
    },
    `p-value` = {
      m <- xtabs(value ~ age + category, data = .)
      stats::chisq.test(m)$p.value
    },
    `V de Cramér` = {
      m <- xtabs(value ~ age + category, data = .)
      rstatix::cramer_v(m)
    },
    .groups = "drop"
  )
```

```r
knitr::kable(
  age_chi2,
  digits  = c(NA, 1, 3, 3),
  caption = "Chi² et V de Cramér de l'association âge × catégorie (EFPV/EnFPV) par groupe de revenu"
)

# Préparer le tableau large pour Y15-24 vs YGE25 (seulement EFPV)
df_wide_age <- df_country_cat_prop %>%
  dplyr::filter(category == "Green") %>%
  dplyr::select(iso3c, income_group, age, country_prop) %>%
  tidyr::pivot_wider(
    names_from  = age,
    values_from = country_prop
  ) %>%

  dplyr::rename(
    Y15_24 = `Y15-24`
  )

# Tests Wilcoxon apparié + cliff's delta par groupe de revenu
res_age_tests <- df_wide_age %>%
  dplyr::group_by(income_group) %>%
  dplyr::summarise(
    W_stat  = stats::wilcox.test(YGE25, Y15_24, paired = TRUE)$statistic,
    p_value = stats::wilcox.test(YGE25, Y15_24, paired = TRUE)$p.value,
    delta   = effsize::cliff.delta(YGE25, Y15_24)$estimate,
    .groups = "drop"
  ) %>%
  dplyr::rename(
    `Groupe de revenu` = income_group,
    `W (Wilcoxon)`     = W_stat,
    `p-value`          = p_value,
    `  de Cliff`       = delta
  )

# Résultat
knitr::kable(
  res_age_tests,
  digits  = c(NA, 0, 3, 3),
  caption = "Wilcoxon apparié et  de Cliff : comparaison YGE25 vs Y15-24 par groupe de revenu"
```

```
)

# Graphique comparatif green vs non-green par âge et revenu
ggplot(df_country_cat_prop,
       aes(x = age, y = country_prop, fill = category)) +
  geom_col(position = position_dodge(width = 0.7), width = 0.6) +
  geom_text(aes(label = percent(country_prop, 1)),
            position = position_dodge(width = 0.7),
            vjust = -0.3, size = 3) +
  facet_wrap(~ income_group, ncol = 2) +
  scale_y_continuous(labels = percent_format(1),
                     expand = expansion(mult = c(0,0.1))) +
  scale_fill_manual(values = c("#d9d9d9","#4daf4a")) +
  labs(
    x      = "Tranche d'âge",
    y      = "Part moyenne d'emplois",
    fill   = "Catégorie",
    title  = "Répartition des EFPV vs EnFPV :\ncomparaison par âge et groupe
     ↪  de revenu",
    caption= "Source : EMP_TEMP_AGE_EC2_NB_A (moyenne par pays)"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major.x = element_blank(),
    legend.position    = "bottom"
  )

# Télécharger
ggsave("/Users/.../Chap2/age_revenu.png", plot = last_plot(), width = 8,
 ↪  height = 6, dpi = 300, bg = "white")
```

# 4 Profils des emplois

## 4.1 Statut d'informalité

```
library(dplyr)
library(tidyr)
library(effsize)    # pour cliff.delta()
library(rstatix)    # pour wilcox_test()
```

```r
library(scales)
library(knitr)
library(ggplot2)

# Préparation
df_ifl_cat <- EMP_TEMP_SEX_IFL_EC2_NB_A %>%
  dplyr::filter(
    EU_ISIC4_2dg %in% c(1,2),   # EnFPV (1) vs EFPV (2)
    ifl          %in% c(2,3)    # Informal (2) / Formal (3)
  ) %>%
  dplyr::mutate(
    category     = factor(EU_ISIC4_2dg,
                          levels = c(1,2),
                          labels = c("EnFPV","EFPV")),
    ifl          = factor(ifl,
                          levels = c(2,3),
                          labels = c("Informel","Formel")),
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("Revenu haut",
                                     "Revenu faible",
                                     "Revenu moyen inf.",
                                     "Revenu moyen sup."))
  )

# Parts Informal/Formal par pays-année-groupe × catégorie
df_country_ifl_cat <- df_ifl_cat %>%
  dplyr::group_by(iso3c, year, income_group, category, ifl) %>%
  dplyr::summarise(
    jobs = sum(value, na.rm = TRUE),
    .groups = "drop_last"
  ) %>%
  dplyr::mutate(
    prop_ifl = jobs / sum(jobs)  # proportion Informal vs Formal dans chaque
     ↪  catégorie
  ) %>%
  dplyr::ungroup() %>%
  dplyr::group_by(iso3c, income_group, category, ifl) %>%
  dplyr::summarise(
    country_prop_ifl = mean(prop_ifl, na.rm = TRUE),
    .groups = "drop"
  )
```

```
# Tableau des parts moyennes (%) par catégorie × revenu × IFL
table_ifl_cat <- df_country_ifl_cat %>%
  dplyr::group_by(income_group, category, ifl) %>%
  dplyr::summarise(
    mean_share = mean(country_prop_ifl, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  dplyr::mutate(
    `% jobs` = scales::percent(mean_share, accuracy = 1)
  ) %>%

  dplyr::select(
    `Groupe de revenu` = income_group,
    `Catégorie`       = category,
    `Type d'emploi`   = ifl,
    `% jobs`
  )

knitr::kable(
  table_ifl_cat,
  caption = "Part moyenne d'emplois Informel vs Formel\npour chaque catégorie
   ↪  (EFPV/EnFPV) et groupe de revenu",
  align   = c("l","l","l","r")
)

# Tests Wilcoxon appariés +   de Cliff par catégorie & revenu
res_ifl_cat_tests <- df_country_ifl_cat %>%
  tidyr::pivot_wider(
    names_from  = ifl,
    values_from = country_prop_ifl
  ) %>%
  group_by(income_group, category) %>%
  group_map(~{
    dat <- .x
    wt  <- wilcox.test(dat$Formel, dat$Informel, paired = TRUE)
    n   <- nrow(dat)
    delta <- if (n >= 3) {
               suppressWarnings(
                 effsize::cliff.delta(dat$Formel, dat$Informel)$estimate
               )
             } else {
               NA_real_
             }
```

```r
    tibble::tibble(
      `Groupe de revenu` = .y$income_group,
      `Catégorie`        = .y$category,
      `W (Wilcoxon)`     = wt$statistic,
      `p-value`          = wt$p.value,
      `  de Cliff`       = delta
    )
  }, .keep = TRUE) %>%
  bind_rows()

knitr::kable(
  res_ifl_cat_tests,
  digits  = c(NA, NA, 0, 3, 3),
  caption = "Wilcoxon apparié et   de Cliff : comparaison Formel vs
   ↪  Informel\nau sein de chaque catégorie et groupe de revenu"
)

# Graphique comparatif Informel vs Formel
p_ifl_cat <- ggplot(df_country_ifl_cat,
       aes(x = ifl, y = country_prop_ifl, fill = ifl)) +
  geom_col(position = position_dodge(width = 0.7),
           width = 0.6, show.legend = FALSE) +
  facet_grid(category ~ income_group) +
  scale_y_continuous(labels = scales::percent_format(1),
                     expand = expansion(mult = c(0,0.1))) +
  labs(
    x       = "Type d'emploi",
    y       = "Part d'emploi moyenne",
    title   = "Informel vs formel : part de chaque catégorie (EFPV/EnFPV) par
     ↪  groupe de revenu",
    caption = ""
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major.x = element_blank(),
    strip.text         = element_text(face = "bold")
  )

# Résultat
print(p_ifl_cat)

# Téléchargement
ggsave(
```

```
  filename = "/Users/.../Chap2/graphique_informal_vs_formal.png",
  plot     = p_ifl_cat,
  width    = 10,
  height   = 6,
  dpi      = 300,
  bg       = "white"
)
```

```
library(dplyr)
library(ggplot2)
library(scales)
library(ggrepel)
library(purrr)
library(gridExtra)

# Préparation des données
df_sect_inc <- EMP_TEMP_SEX_IFL_EC2_NB_A %>%
  filter(sex == 1, EU_ISIC4_2dg %in% c(1,2), ifl %in% c(2,3)) %>%
  mutate(
    is_green     = EU_ISIC4_2dg == 2,
    is_informal  = ifl == 2,
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("Revenu haut",
                                     "Revenu faible",
                                     "Revenu moyen inf.",
                                     "Revenu moyen sup."))
  ) %>%
  group_by(income_group, iso3c, year, isic_code_2dg) %>%
  summarise(
    total_emp = sum(value, na.rm = TRUE),
    green_emp = sum(value[is_green],   na.rm = TRUE),
    inf_emp   = sum(value[is_informal],na.rm = TRUE),
    .groups   = "drop"
  ) %>%
  mutate(prop_inf = inf_emp / total_emp) %>%
  group_by(income_group, isic_code_2dg) %>%
  summarise(
    green_emp_mean = mean(green_emp, na.rm = TRUE) / 1000,
    inf_rate_mean  = mean(prop_inf,  na.rm = TRUE),
    total_emp_mean = mean(total_emp, na.rm = TRUE) / 1000,
    .groups        = "drop"
```

```r
  )

# Sélection des secteurs à étiqueter (volontairement > 1000 verts)
df_labels_inc <- df_sect_inc %>%
  filter(green_emp_mean > 1)

# Génération d'une liste de graphiques, un par income_group
plots_by_group <- df_sect_inc %>%
  group_split(income_group) %>%
  set_names(
    df_sect_inc %>% pull(income_group) %>% unique()
  ) %>%
  map(~{
    df <- .x
    ig <- unique(df$income_group)
    ggplot(df, aes(x = green_emp_mean, y = inf_rate_mean)) +
      geom_point(aes(size = total_emp_mean), color = "#1f78b4", alpha = 0.7)
       ↪  +
      geom_smooth(method = "lm", color = "#e31a1c", se = FALSE) +
      geom_text_repel(
        data = df_labels_inc %>% filter(income_group == ig),
        aes(label = isic_code_2dg),
        size = 3, box.padding = 0.2, point.padding = 0.2
      ) +
      scale_size_continuous(
        range  = c(2, 8),
        name   = "Total des emplois (moyenne × 1 000)",
        labels = comma_format(accuracy = 1)
      ) +
      scale_x_continuous(
        breaks = 0:ceiling(max(df_sect_inc$green_emp_mean, na.rm = TRUE)),
        labels = comma_format(accuracy = 1),
        name   = "Moyenne d'EFPV par secteur (× 1 000)"
      ) +
      scale_y_continuous(
        labels = percent_format(1),
        limits = c(0,1),
        name   = "Taux moyen d'informalité"
      ) +
      labs(
        title   = paste0("Informalité vs EFPV. ", ig),
        subtitle= "",
        caption = ""
```

```
      ) +
      theme_minimal(base_size = 12) +
      theme(
        panel.grid.major.x = element_blank(),
        legend.position    = "none",
        plot.subtitle      = element_text(size = 9, face = "italic")
      )
  })

# Résultats : soit un à un, soit tous ensemble

# Pour voir un seul (par ex. High income) :
plots_by_group[["Revenu haut"]]

# Pour afficher les quatre en une grille 2×2 :
grid.arrange(grobs = plots_by_group, ncol = 2)

# Téléchargement p1_multi
groups <- names(plots_by_group)
file_paths <- paste0(
  "/Users/.../Chap2/plot_",
  gsub(" ", "_", tolower(groups)), ".png"
)
purrr::walk2(
  plots_by_group, file_paths,
  ~ ggsave(
      filename = .y,
      plot     = .x,
      width    = 8,
      height   = 6,
      dpi      = 300,
      bg       = "white"
    )
)
```

```
library(dplyr)
library(ggplot2)
library(scales)
library(ggrepel)
library(purrr)
library(gridExtra)
```

```r
# Calcul des agrégats par secteur et groupe de revenu
df_sect_inc <- EMP_TEMP_SEX_IFL_EC2_NB_A %>%
  filter(sex == 1, EU_ISIC4_2dg %in% c(1,2), ifl %in% c(2,3)) %>%
  mutate(
    is_green     = EU_ISIC4_2dg == 2,
    is_informal  = ifl == 2,
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("Revenu haut",
                                     "Revenu faible",
                                     "Revenu moyen inf.",
                                     "Revenu moyen sup."))
  ) %>%
  group_by(income_group, iso3c, year, isic_code_2dg) %>%
  summarise(
    total_emp = sum(value, na.rm = TRUE),
    green_emp = sum(value[is_green],   na.rm = TRUE),
    inf_emp   = sum(value[is_informal],na.rm = TRUE),
    .groups   = "drop"
  ) %>%
  mutate(prop_inf = inf_emp / total_emp) %>%
  group_by(income_group, isic_code_2dg) %>%
  summarise(
    total_emp_mean     = mean(total_emp, na.rm = TRUE) / 1000,
    green_emp_mean     = mean(green_emp, na.rm = TRUE) / 1000,
    non_green_emp_mean = total_emp_mean - green_emp_mean,
    inf_rate_mean      = mean(prop_inf,  na.rm = TRUE),
    .groups            = "drop"
  )

# On décide quels secteurs on étiquette (ici non_green_emp_mean > 1k)
df_labels_non <- df_sect_inc %>%
  filter(non_green_emp_mean > 1)

# Construction de la liste de 4 graphiques (un par groupe de revenu)
plots_non <- df_sect_inc %>%
  group_split(income_group) %>%
  set_names(df_sect_inc %>% pull(income_group) %>% unique()) %>%
  map(~{
    df <- .x
    grp <- unique(df$income_group)
    ggplot(df, aes(x = non_green_emp_mean, y = inf_rate_mean)) +
      geom_point(aes(size = total_emp_mean),
```

```r
                 color = "#1f78b4", alpha = 0.7) +
      geom_smooth(method = "lm", color = "#e31a1c", se = FALSE) +
      geom_text_repel(
        data = df_labels_non %>% filter(income_group == grp),
        aes(label = isic_code_2dg),
        size = 3, box.padding = 0.2, point.padding = 0.2
      ) +
      scale_size_continuous(
        range  = c(2, 8),
        name   = "Total des emplois (moyenne × 1 000)",
        labels = comma_format(accuracy = 1)
      ) +
      scale_x_continuous(
        breaks = 0:ceiling(max(df_sect_inc$non_green_emp_mean, na.rm =
         ↪  TRUE)),
        labels = comma_format(accuracy = 1),
        name   = "Moyenne d'EnFPV par secteur (× 1 000)"
      ) +
      scale_y_continuous(
        labels = percent_format(1),
        limits = c(0,1),
        name   = "Taux moyen d'informalité"
      ) +
      labs(
        title    = paste0("Informalité vs EnFPV. ", grp),
        subtitle = "",
        caption  = ""
      ) +
      theme_minimal(base_size = 12) +
      theme(
        panel.grid.major.x = element_blank(),
        legend.position    = "none",
        plot.subtitle      = element_text(size = 9, face = "italic")
      )
  })

# Visualisation
# Pour un seul (ex : High income)
plots_non[["Revenu haut"]]

# Pour afficher les quatre à la fois en 2×2
grid.arrange(grobs = plots_non, ncol = 2)
```

```r
# Liste des groupes de revenu
groups_non <- names(plots_non)

# Téléchargement
file_paths_non <- paste0(
  "/Users/.../Chap2/plot_non_green_",
  gsub(" ", "_", tolower(groups_non)), ".png"
)

purrr::walk2(
  plots_non, file_paths_non,
  ~ ggsave(
      filename = .y,
      plot     = .x,
      width    = 8,
      height   = 6,
      dpi      = 300,
      bg       = "white"
    )
)
```

On accompagne ce résultat d'un test :

```r
library(dplyr)
library(broom)
library(tidyr)
library(purrr)
library(knitr)

# Tests statistiques par groupe de revenu

# a) Corrélation de Spearman
spearman_tests <- df_sect_inc %>%
  group_by(income_group) %>%
  group_modify(~ {
    ct <- cor.test(.x$non_green_emp_mean, .x$inf_rate_mean, method =
  ↪  "spearman")
    tibble(
      rho       = unname(ct$estimate),
      p_spearman = ct$p.value
    )
  })
```

```r
# b) Régression linéaire simple
lm_tests <- df_sect_inc %>%
  dplyr::group_by(income_group) %>%
  dplyr::group_modify(~ {
    fit <- lm(inf_rate_mean ~ non_green_emp_mean, data = .x)
    broom::tidy(fit) %>%
      dplyr::filter(term == "non_green_emp_mean") %>%
      dplyr::select(
        estimate,
        std.error,
        statistic,   # t-statistic
        p.value
      )
  }) %>%
  dplyr::ungroup()

knitr::kable(
  lm_tests,
  digits  = 3,
  caption = "Pentes et tests t de la régression\ninf_rate_mean ~
  ↪  non_green_emp_mean par groupe de revenu"
)

# c) On fusionne les deux résultats
stats_non <- spearman_tests %>%
  left_join(lm_tests, by = "income_group") %>%
  rename(
    slope      = estimate,
    se_slope   = std.error,
    t_slope    = statistic,
    p_slope    = p.value
  )

# Résultat
knitr::kable(
  stats_non,
  digits  = 3,
  caption = "Tests de corrélation (Spearman) et régression linéaire\npar
  ↪  groupe de revenu"
)
```

Nous produisons une carte du monde de la répartition emplois formels/informels verts/non verts etc… 4 cartes de fait :

```
if (!requireNamespace("patchwork", quietly = TRUE))
↪  install.packages("patchwork")
library(dplyr)
library(sf)
library(rnaturalearth)
library(ggplot2)
library(scales)
library(knitr)

# Proportions pays-année, neutralisation années
df_map_cy <- EMP_TEMP_SEX_IFL_EC2_NB_A %>%
  filter(
    sex == 1,
    EU_ISIC4_2dg %in% c(1,2),
    ifl %in% c(2,3)
  ) %>%
  group_by(iso3c, year) %>%
  summarise(
    green_inf     = sum(value[EU_ISIC4_2dg==2 & ifl==2], na.rm = TRUE),
    green_form    = sum(value[EU_ISIC4_2dg==2 & ifl==3], na.rm = TRUE),
    non_green_inf = sum(value[EU_ISIC4_2dg==1 & ifl==2], na.rm = TRUE),
    non_green_for = sum(value[EU_ISIC4_2dg==1 & ifl==3], na.rm = TRUE),
    total         = green_inf + green_form + non_green_inf + non_green_for,
    prop_green_inf     = green_inf     / total,
    prop_non_green_inf = non_green_inf / total,
    prop_green_for     = green_form    / total,
    prop_non_green_for = non_green_for / total,
    .groups = "drop"
  )

year_min     <- min(df_map_cy$year, na.rm = TRUE)
year_max     <- max(df_map_cy$year, na.rm = TRUE)
period_label <- paste0(year_min, "-", year_max)

# Moyenne multi-années + renommage des variables
df_map_all <- df_map_cy %>%
  group_by(iso3c) %>%
  summarise(
    prop_green_inf     = mean(prop_green_inf,     na.rm = TRUE),
    prop_non_green_inf = mean(prop_non_green_inf, na.rm = TRUE),
```

```r
    prop_green_for     = mean(prop_green_for,     na.rm = TRUE),
    prop_non_green_for = mean(prop_non_green_for, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(iso3c = toupper(trimws(iso3c))) %>%
  rename(
    prop_EFPV_inf  = prop_green_inf,
    prop_EnFPV_inf = prop_non_green_inf,
    prop_EFPV_for  = prop_green_for,
    prop_EnFPV_for = prop_non_green_for
  )

# Meilleure année + renommage
best_year <- df_map_cy %>%
  group_by(year) %>%
  summarise(n_iso = n_distinct(iso3c), .groups = "drop") %>%
  slice_max(n_iso, n = 1) %>%
  pull(year)

df_map_best <- df_map_cy %>%
  filter(year == best_year) %>%
  select(iso3c, starts_with("prop_")) %>%
  distinct() %>%
  mutate(iso3c = toupper(trimws(iso3c))) %>%
  rename(
    prop_EFPV_inf  = prop_green_inf,
    prop_EnFPV_inf = prop_non_green_inf,
    prop_EFPV_for  = prop_green_for,
    prop_EnFPV_for = prop_non_green_for
  )

# Base carte monde
world <- ne_countries(scale = "medium", returnclass = "sf") %>%
  st_transform(crs = 4326)

# Fonction de carte Bordeaux
plot_map_bordeaux <- function(data, var, title, subtitle) {
  ggplot(data) +
    geom_sf(aes_string(fill = var), color = "gray80", size = 0.1) +
    scale_fill_gradient(
      low       = "#fbeaec",
      high      = "#7b1f3a",
      na.value  = "white",
```

```r
      labels    = percent_format(accuracy = 1),
      name      = title
    ) +
    labs(
      title    = title,
      subtitle = subtitle,
      caption  = ""
    ) +
    theme_minimal(base_size = 12) +
    theme(
      panel.grid      = element_blank(),
      axis.text       = element_blank(),
      axis.ticks      = element_blank(),
      legend.position = "bottom"
    )
}

# Cartes Multi-Années
world_map_multi <- world %>%
  left_join(df_map_all, by = c("adm0_a3" = "iso3c"))
subtitle_multi <- paste0("Période : ", period_label, " (moyenne par pays)")

p1_multi <- plot_map_bordeaux(world_map_multi, "prop_EFPV_inf",  "% EFPV
 ↪  informels",  subtitle_multi)
p2_multi <- plot_map_bordeaux(world_map_multi, "prop_EnFPV_inf", "% EnFPV
 ↪  informels", subtitle_multi)
p3_multi <- plot_map_bordeaux(world_map_multi, "prop_EFPV_for",  "% EFPV
 ↪  formels",    subtitle_multi)
p4_multi <- plot_map_bordeaux(world_map_multi, "prop_EnFPV_for", "% EnFPV
 ↪  formels",   subtitle_multi)

# Cartes Année Optimale
world_map_best <- world %>%
  left_join(df_map_best, by = c("adm0_a3" = "iso3c"))
subtitle_best <- paste0("Année optimale : ", best_year)

p1_best <- plot_map_bordeaux(world_map_best, "prop_EFPV_inf",  "% EFPV
 ↪  informels",  subtitle_best)
p2_best <- plot_map_bordeaux(world_map_best, "prop_EnFPV_inf", "% EnFPV
 ↪  informels", subtitle_best)
p3_best <- plot_map_bordeaux(world_map_best, "prop_EFPV_for",  "% EFPV
 ↪  formels",    subtitle_best)
```

```
p4_best <- plot_map_bordeaux(world_map_best, "prop_EnFPV_for", "% EnFPV
 ↪  formels",   subtitle_best)

# Résultat
print(p1_multi); print(p2_multi); print(p3_multi); print(p4_multi)
print(p1_best);  print(p2_best);  print(p3_best);  print(p4_best)

# Télécharger les 8 cartes
ggsave("p1_multi_verts_informels.png", plot = p1_multi, width = 8, height =
 ↪  5, dpi = 300, bg = "white")
ggsave("p2_multi_nonverts_informels.png", plot = p2_multi, width = 8, height
 ↪  = 5, dpi = 300, bg = "white")
ggsave("p3_multi_verts_formels.png", plot = p3_multi, width = 8, height = 5,
 ↪  dpi = 300, bg = "white")
ggsave("p4_multi_nonverts_formels.png", plot = p4_multi, width = 8, height =
 ↪  5, dpi = 300, bg = "white")
ggsave("p1_best_verts_informels.png", plot = p1_best,  width = 8, height = 5,
 ↪  dpi = 300, bg = "white")
ggsave("p2_best_nonverts_informels.png", plot = p2_best, width = 8, height =
 ↪  5, dpi = 300, bg = "white")
ggsave("p3_best_verts_formels.png", plot = p3_best, width = 8, height = 5,
 ↪  dpi = 300, bg = "white")
ggsave("p4_best_nonverts_formels.png", plot = p4_best, width = 8, height = 5,
 ↪  dpi = 300, bg = "white")

#  Liste des pays par groupe de revenu
pays_par_groupe <- EMP_TEMP_SEX_IFL_EC2_NB_A %>%
  filter(sex == 1) %>%
  mutate(
    income_group = factor(income_group, levels = 1:4,
                          labels = c("Revenu haut","Revenu faible",
                                     "Revenu moyen inf.","Revenu moyen
                                      ↪  sup."))
  ) %>%
  distinct(income_group, iso3c) %>%
  group_by(income_group) %>%
  summarise(Pays = paste(sort(unique(iso3c)), collapse = ", "), .groups =
   ↪  "drop")

knitr::kable(
  pays_par_groupe,
  caption = "Pays présents par groupe de revenu (sex = 1)"
```

```
)
```

## 4.2 Firm size

```
EMP_TEMP_SEX_EST_EC2_NB_A <-
↪  readRDS("/Users/.../Chap2/EMP_TEMP_SEX_EST_EC2_NB_A.rds")
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
library(rstatix)   # pour cramer_v(), kruskal_test(), kruskal_effsize()
library(effsize)   # pour cliff.delta()
library(knitr)

# Préparation
df_est <- EMP_TEMP_SEX_EST_EC2_NB_A %>%
  filter(
    EU_ISIC4_2dg %in% c(1,2),       # 1=Non-green, 2=Green
    est          %in% 2:5           # 2=S1-4,3=S5-49,4=>=50,5=X
    # (optionnel) & sex==1 si nécessaire
  ) %>%
  mutate(
    category     = factor(EU_ISIC4_2dg,
                          levels = c(1,2),
                          labels = c("Non-green","Green")),
    `Taille`     = factor(est,
                          levels = 2:5,
                          labels = c("S1-4","S5-49",">=50","X")),
    `Groupe`     = factor(income_group,
                          levels = 1:4,
                          labels = c("High income","Low income","Lower
                           ↪  middle","Upper middle"))
  )

# Calcul des proportions par pays-année-groupe-taille × catégorie
df_country_est <- df_est %>%
  group_by(iso3c, year, Groupe, Taille, category) %>%
  summarise(jobs = sum(value, na.rm=TRUE), .groups="drop") %>%
  group_by(iso3c, year, Groupe, Taille) %>%
```

```
  mutate(prop = jobs / sum(jobs)) %>%      # proportion dans chaque
   ↪  pays-année-taille
  ungroup() %>%
  group_by(iso3c, Groupe, Taille, category) %>%
  summarise(country_prop = mean(prop, na.rm=TRUE),
            .groups="drop")                # moyenne par pays (neutralise le
             ↪  biais années)

# Tableau des parts moyennes (%) par taille × catégorie × groupe
table_shares_est <- df_country_est %>%
  dplyr::group_by(Groupe, Taille, category) %>%
  dplyr::summarise(mean_share = mean(country_prop), .groups="drop") %>%
  tidyr::pivot_wider(
    names_from  = category,
    values_from = mean_share
  ) %>%
  # on formate en %, on renomme et on sélectionne les colonnes finales
  dplyr::mutate(
    `Non-green (%)` = scales::percent(`Non-green`, accuracy = 1),
    `Green (%)`     = scales::percent(Green,       accuracy = 1)
  ) %>%
  dplyr::rename(
    `Groupe de revenu`       = Groupe,
    `Taille d'établissement` = Taille
  ) %>%
  dplyr::select(
    `Groupe de revenu`,
    `Taille d'établissement`,
    `Non-green (%)`,
    `Green (%)`
  )

knitr::kable(
  table_shares_est,
  caption = "Part moyenne d'emplois verts vs non-verts\npar taille
   ↪  d'établissement et groupe de revenu",
  align   = c("l","l","r","r")
)

# Test du  ² + V de Cramér sur les effectifs bruts
est_chi2 <- df_est %>%
  group_by(Groupe) %>%
  summarise(
```

```
    2      = { m <- xtabs(value ~ Taille + category, data = .);
                 chisq.test(m)$statistic },
    `p-value` = { m <- xtabs(value ~ Taille + category, data = .);
                   chisq.test(m)$p.value },
    `V de Cramér` = { m <- xtabs(value ~ Taille + category, data = .);
                       cramer_v(m) },
    .groups="drop"
  )

knitr::kable(
  est_chi2,
  digits  = c(NA, 1, 3, 3),
  caption = "Chi² et V de Cramér de l’association taille×catégorie par groupe
   ↪  de revenu"
)

# Test non-paramétrique (Kruskal) + taille d’effet
res_krusk <- df_country_est %>%
  # a) Se limiter aux emplois verts
  filter(category == "Green") %>%
  # b) S'assurer que Taille est un facteur
  mutate(Taille = factor(Taille)) %>%
  # c) Grouper par groupe de revenu
  group_by(Groupe) %>%
  # d) Pour chaque groupe, exécuter Kruskal et l'effet
  group_modify(~ {
    kt <- kruskal_test(.x, country_prop ~ Taille)
    es <- kruskal_effsize(.x, country_prop ~ Taille)
    tibble(
      statistique     = kt$statistic,
      `p-value`       = kt$p,
      ` ² de Kruskal` = es$effsize
    )
  }) %>%
  ungroup() %>%
  # e) Renommer pour l'affichage
  rename(`Groupe de revenu` = Groupe)

# 6) Résultat
knitr::kable(
  res_krusk,
  digits  = c(NA, 2, 3, 3),
  caption = "Test de Kruskal-Wallis et  ² de Kruskal sur la part d’emplois
   ↪  verts\npar taille d’établissement et groupe de revenu"
```

```
)
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
library(rstatix)   # pour cramer_v(), kruskal_test(), kruskal_effsize()
library(effsize)   # pour cliff.delta()
library(knitr)

# Préparation
df_est <- EMP_TEMP_SEX_EST_EC2_NB_A %>%
  filter(
    EU_ISIC4_2dg %in% c(1,2),       # 1=Non-green, 2=Green
    est          %in% 2:5           # 2=S1-4,3=S5-49,4=>=50,5=X
  ) %>%
  mutate(
    category     = factor(EU_ISIC4_2dg,
                          levels = c(1,2),
                          labels = c("EnFPV","EFPV")),  # <- changement ici
    `Taille`     = factor(est,
                          levels = 2:5,
                          labels = c("S1-4","S5-49",">=50","X")),
    `Groupe`     = factor(income_group,
                          levels = 1:4,
                          labels = c("Revenu haut","Revenu faible","Revenu
                           ↪  moyen inf.","Revenu moyen sup."))
  )

# Calcul des proportions par pays-année-groupe-taille × catégorie
df_country_est <- df_est %>%
  group_by(iso3c, year, Groupe, Taille, category) %>%
  summarise(jobs = sum(value, na.rm=TRUE), .groups="drop") %>%
  group_by(iso3c, year, Groupe, Taille) %>%
  mutate(prop = jobs / sum(jobs)) %>%      # proportion dans chaque
   ↪  pays-année-taille
  ungroup() %>%
  group_by(iso3c, Groupe, Taille, category) %>%
  summarise(country_prop = mean(prop, na.rm=TRUE),
            .groups="drop")               # moyenne par pays

# Graphique
```

```r
ggplot(df_country_est,
       aes(x = Taille, y = country_prop, fill = category)) +
  geom_col(position = position_dodge(width = 0.7), width = 0.6) +
  facet_wrap(~ Groupe, ncol = 2) +
  scale_y_continuous(labels = percent_format(accuracy = 1),
                     expand = expansion(mult = c(0,0.1))) +
  scale_fill_manual(
    values = c("#d9d9d9","#4daf4a"),
    labels = c("EnFPV","EFPV")  # légende propre
  ) +
  labs(
    x      = "Taille de l'établissement",
    y      = "Part moyenne de l'emploi",
    fill   = "Catégorie",
    title  = "Répartition d'EFPV par rapport aux EnFPV selon la taille de
     ↪  l'établissement et la catégorie de revenus",
    caption= ""
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major.x = element_blank(),
    legend.position    = "bottom"
  )

# Téléchargement
ggsave(
  filename = "/Users/.../Chap2/graphique_JHGP_vs_JnHGP_etablissement.png",
  plot     = last_plot(),
  width    = 10,
  height   = 7,
  dpi      = 300,
  bg       = "white"
)
```

```r
library(dplyr)
library(DescTools)

# On part de df_country_est
# On filtre pour les EFPV dans le groupe « Revenu haut »
dat_hi <- df_country_est %>%
  filter(
    category == "EFPV",
```

```
    Groupe   == "Revenu haut"
  ) %>%
  mutate(
    Taille_num = as.integer(factor(
      Taille,
      levels  = c("S1-4", "S5-49", ">=50", "X"),
      ordered = TRUE
    ))
  )

# Vérification rapide
print(table(dat_hi$Taille_num))

# Test de Jonckheere-Terpstra pour tendance monotone croissante
jt_hi <- JonckheereTerpstraTest(
  x           = dat_hi$country_prop,
  g           = dat_hi$Taille_num,
  alternative = "increasing",
  nperm       = 5000
)

print(jt_hi)
```

```
library(dplyr)
library(DescTools)
library(knitr)

res_jt <- df_country_est %>%
  filter(category == "EFPV") %>%
  mutate(
    Taille_num = as.integer(factor(
      Taille,
      levels  = c("S1-4", "S5-49", ">=50", "X"),
      ordered = TRUE
    ))
  ) %>%
  group_by(Groupe) %>%
  group_modify(~ {
    if (n_distinct(.x$Taille_num) < 2) {

      return(tibble(
        `JT statistic` = NA_real_,
```

```
        `p-value`      = NA_real_
      ))
    }
    jt <- suppressWarnings(
      JonckheereTerpstraTest(
        x           = .x$country_prop,
        g           = .x$Taille_num,
        alternative = "increasing",
        nperm       = 5000
      )
    )
    tibble(
      `JT statistic` = as.numeric(jt$statistic),
      `p-value`      = as.numeric(jt$p.value)
    )
  }) %>%
  ungroup() %>%
  rename(`Groupe de revenu` = Groupe)

knitr::kable(
  res_jt,
  digits  = c(NA, 0, 4),
  caption = "Jonckheere-Terpstra (5000 permutations) : tendance croissante de
   ↪  la part d'EFPV selon la taille de l'établissement, par groupe de
   ↪  revenu"
)
```

Nous testons maintenant la taille de l'effet :

```
library(dplyr)
library(broom)
library(knitr)
library(scales)

res_amplitude <- df_country_est %>%

  filter(category == "EFPV") %>%
  mutate(
    Taille_num = as.integer(factor(
      Taille,
      levels  = c("S1-4", "S5-49", ">=50", "X"),
      ordered = TRUE
```

```r
    ))
  ) %>%
  group_by(Groupe) %>%
  group_modify(~{
    dat <- .x
    # a) moyennes par taille
    moyennes <- dat %>%
      group_by(Taille_num) %>%
      summarise(mean_prop = mean(country_prop, na.rm = TRUE), .groups =
       ↪  "drop")
    amp_abs <- moyennes$mean_prop[4] - moyennes$mean_prop[1]
    # b) régression linéaire sur la prop. d'EFPV selon Taille_num
    fit   <- lm(country_prop ~ Taille_num, data = dat)
    slope <- coef(fit)["Taille_num"]
    r2    <- glance(fit)$r.squared
    # c) corrélation de rang (Spearman)
    cor_sp <- cor(dat$country_prop, dat$Taille_num, method = "spearman")
    tibble(
      `Moyenne S1-4`   = moyennes$mean_prop[1],
      `Moyenne X`      = moyennes$mean_prop[4],
      `Amplitude (pp)` = amp_abs,
      `Pente linéaire` = slope,
      `R²`             = r2,
      `  de Spearman`  = cor_sp
    )
  }) %>%
  ungroup() %>%
  rename(`Groupe de revenu` = Groupe)

# Mise en forme du tableau : % pour les moyennes et l'amplitude, arrondi pour
 ↪  les autres
knitr::kable(
  res_amplitude %>%
    mutate(
      across(
        c(`Moyenne S1-4`, `Moyenne X`, `Amplitude (pp)`),
        ~ percent(., accuracy = 0.1)
      ),
      across(
        c(`Pente linéaire`, `R²`, `  de Spearman`),
        ~ round(., 3)
      )
    ),
```

```
  caption = "Amplitude et pente de la part d’EFPV selon la taille de
  ↪  l’établissement\n(par groupe de revenu)",
  digits = 3,
  align  = "lrrrrr"
)
```

## 4.3 Heure moyen des emplois

```
HOW_TEMP_SEX_EC2_NB_A <-
↪  readRDS("/Users/.../Chap2/HOW_TEMP_SEX_EC2_NB_A.rds")
```

```
library(dplyr)
library(purrr)    # pour map_dbl()
library(tidyr)
library(broom)
library(knitr)

# On calcule d = country_h(Green) - country_h(Non-green) pour chaque pays ×
↪  groupe
df_diff <- HOW_TEMP_SEX_EC2_NB_A %>%
  filter(sex == 1, EU_ISIC4_2dg %in% c(1,2)) %>%
  mutate(
    category     = factor(EU_ISIC4_2dg,
                          levels = c(1,2),
                          labels = c("Non-green","Green")),
    income_group = factor(income_group,
                          levels = 1:4,
                          labels = c("High","Low","Lower","Upper"))
  ) %>%
  group_by(iso3c, year, income_group, category) %>%
  summarise(mean_h = mean(value, na.rm = TRUE), .groups="drop") %>%
  pivot_wider(names_from = category, values_from = mean_h) %>%
  mutate(diff = Green - `Non-green`)

# On agrége par pays sur les années pour neutraliser le biais temporel
df_diff_country <- df_diff %>%
  group_by(iso3c, income_group) %>%
  summarise(diff = mean(diff, na.rm = TRUE), .groups="drop")

# Pour chaque groupe de revenu, on choisit et on exécute le test
```

```r
res_tests <- df_diff_country %>%
  group_by(income_group) %>%
  summarise(
    # a) normalité
    p_shapiro = shapiro.test(diff)$p.value,
    # b) choix du test et stockage de l'objet "htest"
    test_raw = list(
      if (p_shapiro > 0.05) {
        t.test(diff, mu = 0)                   # one-sample t-test
      } else {
        wilcox.test(diff, mu = 0, exact = FALSE)  # Wilcoxon one-sample
      }
    ),
    .groups = "drop"
  ) %>%
  mutate(
    Method    = if_else(p_shapiro > 0.05,
                        "Paired t-test",
                        "Wilcoxon signed-rank"),
    Statistic = map_dbl(test_raw, ~ .x$statistic),
    p_value   = map_dbl(test_raw, ~ .x$p.value)
  ) %>%

  dplyr::select(
    `Groupe de revenu` = income_group,
    Method,
    Statistic,
    `p-value` = p_value
  )

knitr::kable(
  res_tests,
  digits  = c(NA, NA, 3, 4),
  caption = "Test de la différence Green - Non-green par groupe de
  ↪  revenu\n(choix automatique t-test vs Wilcoxon selon Shapiro-Wilk)"
)
```

Après un Wilcoxon signed-rank qui nous a confirmé que, dans chaque groupe de revenu, la différence moyenne (Green - Non-green) n'est pas nulle, nous estimons la taille d'effet avec un r de Wilcoxon (appelé rank-biserial correlation):

```r
library(dplyr)
library(knitr)

eff_sizes_manual <- df_diff_country %>%
  group_by(income_group) %>%
  summarise(
    # a) Taille et statistiques brutes des différences
    n         = sum(!is.na(diff)),
    mean_diff = mean(diff, na.rm = TRUE),
    sd_diff   = sd(diff,   na.rm = TRUE),

    # b) On ne conserve que les diffs non-nuls pour le Wilcoxon
    diffs0    = list(diff[!is.na(diff) & diff != 0]),
    n0        = length(diffs0[[1]]),

    # c) Rangs et somme des rangs positifs W+
    ranks0    = list(rank(abs(diffs0[[1]]))),
    Wpos      = sum(ranks0[[1]][diffs0[[1]] > 0]),

    # d) Espérance et variance de W sous H  (Wilcoxon)
    E_W       = n0 * (n0 + 1) / 4,
    V_W       = n0 * (n0 + 1) * (2 * n0 + 1) / 24,

    # e) z-score et r de Wilcoxon
    z_wilcox  = (Wpos - E_W) / sqrt(V_W),
    r_wilcox  = z_wilcox / sqrt(n0),

    # f) Cohen's d pour données appariées
    cohen_d   = mean_diff / sd_diff,
    .groups   = "drop"
  ) %>%
  select(
    `Groupe de revenu` = income_group,
    n, mean_diff, sd_diff, r_wilcox, cohen_d
  )

knitr::kable(
  eff_sizes_manual,
  digits  = c(0, 3, 3, 3, 3, 3),
  caption = "Taille d'effet de la différence (Green-Non-green) par groupe de
  ↪  revenu"
)
```

Tableau.

| Groupe de revenu | n | mean_diff | sd_diff | r_wilcox | cohen_d |
|---|---|---|---|---|---|
| High income | 45 | 0.673 | 1.271 | 0.669 | 0.529 |
| Low income | 16 | 1.216 | 2.027 | 0.530 | 0.600 |
| Lower middle | 41 | 0.890 | 1.817 | 0.422 | 0.490 |
| Upper middle | 32 | 0.615 | 1.281 | 0.443 | 0.480 |

```r
library(dplyr)
library(boot)

# Statistique de bootstrap : moyenne des différences
stat_diff <- function(data, i) {
  diffs <- data$diff[i]
  mean(diffs, na.rm = TRUE)
}

# Pour chaque groupe, on lance le bootstrap
cis <- df_diff_country %>%
  group_by(income_group) %>%
  group_modify(~ {
    dat <- .x
    b <- boot(dat, statistic = stat_diff, R = 5000)
    ci <- boot.ci(b, type = "perc")$percent[4:5]
    tibble(
      mean_diff = mean(dat$diff, na.rm = TRUE),
      CI_lower  = ci[1],
      CI_upper  = ci[2]
    )
  }) %>%
  ungroup() %>%
  rename(`Groupe de revenu` = income_group)

# Résultat
print(cis)
```

Tableau.

| Groupe de revenu | Δ moyen (h) | IC 95 % bootstrap |
|---|---|---|
| High income | 0,673 | [0,343 ; 1,059] |
| Low income | 1,216 | [0,337 ; 2,246] |

| Groupe de revenu | Δ moyen (h) | IC 95 % bootstrap |
|---|---|---|
| Lower middle | 0,890 | [0,388 ; 1,442] |
| Upper middle | 0,615 | [0,206 ; 1,076] |

```
library(dplyr)
library(boot)
library(rstatix)
library(DescTools)
library(tibble)

# On s'assure d'avoir les bons niveaux pour income_group
# On remplace par les libellés exacts dans df_diff_country
real_levels <- c("High", "Low", "Lower", "Upper")

df_diff_country <- df_diff_country %>%
  mutate(income_group = factor(income_group,
                               levels = real_levels,
                               ordered = TRUE))

# Bootstrap 95% CI pour la moyenne des diffs
stat_diff <- function(x, i) mean(x[i], na.rm = TRUE)

cis <- df_diff_country %>%
  group_by(income_group) %>%
  group_modify(~ {
    diffs <- .x$diff
    b     <- boot(diffs, statistic = stat_diff, R = 5000)
    pct   <- tryCatch(
      boot.ci(b, type = "perc")$percent[4:5],
      error = function(e) c(NA_real_, NA_real_)
    )
    tibble(
      mean_diff = mean(diffs, na.rm = TRUE),
      CI_lower  = pct[1],
      CI_upper  = pct[2]
    )
  }) %>%
  ungroup() %>%
  rename(`Groupe de revenu` = income_group)

# Spearman global
spearman_data <- df_diff_country %>%
```

```r
  filter(!is.na(diff)) %>%
  mutate(rev_rank = as.integer(income_group))

trend_spearman <- if (
  n_distinct(spearman_data$rev_rank) > 1 &&
  n_distinct(spearman_data$diff)  > 1
) {
  ct <- cor.test(spearman_data$diff, spearman_data$rev_rank,
                 method = "spearman", exact = FALSE)
  tibble(rho = unname(ct$estimate), p_val = ct$p.value)
} else {
  tibble(rho = NA_real_, p_val = NA_real_)
}

# Jonckheere-Terpstra global
gnum <- as.integer(df_diff_country$income_group)
jt_income <- if (n_distinct(gnum[!is.na(gnum)]) > 1) {
  JonckheereTerpstraTest(
    x            = df_diff_country$diff,
    g            = gnum,
    alternative  = "increasing",
    nperm        = 5000,
    distribution = "permutation"
  )
} else {
  NA
}

# Résultats
print(cis)
print(trend_spearman)
print(jt_income)
```