Exercices Module 2

La DiD Classique 2×2

Master GPE — UCA FERDI IHEDD

2026-03-24

Objectifs de ce module

À la fin de ces exercices, vous saurez :

1. Simuler des données 2×2 pour la DiD
2. Calculer l'estimateur DiD manuellement et par régression
3. Interpréter les coefficients d'un modèle DiD
4. Visualiser une DiD avec `ggplot2`

Travailler dans RStudio ?

Téléchargez le script R complet du module avec tous les exercices à compléter :

Ouvrez-le dans RStudio et complétez les ___ au fur et à mesure. Exécutez bloc par bloc avec **Ctrl+Enter**.

Pourquoi ce concept est-il crucial pour un fonctionnaire ?

La régression DiD est l'outil standard utilisé par les économistes et les évaluateurs des grandes institutions (Banque mondiale, FMI, BAD) pour évaluer les politiques publiques. Savoir lire — et produire — un tableau de régression DiD vous permettra de communiquer vos évaluations dans le langage universel des décideurs et des bailleurs de fonds. C'est une compétence directement valorisable dans la rédaction de rapports d'impact, de notes de politique ou de projets soumis aux financeurs internationaux.

Création et exploration des données

Exercice 2.1 Simuler des données 2×2

Contexte : Le Ministère de l'Emploi et de la Formation Professionnelle du Burkina Faso souhaite évaluer son programme *Formation Compétences+* lancé en 2021. Ce programme, financé par l'AFD à hauteur de 8 milliards de FCFA, cible 50 régions parmi les 100 qui composent le pays. Il offre des formations certifiantes de 3 mois dans les secteurs du bâtiment, de l'agro-industrie et du numérique. La variable de résultat est le taux d'emploi formel (%) mesuré lors d'une enquête emploi. Vous allez simuler des données similaires pour comprendre la mécanique DiD avant de l'appliquer à des données réelles.

```
{webr-r}
library(tidyverse)

set.seed(2024)
n_regions <- 100  # 50 régions traitées, 50 régions contrôles

# TODO : Créez le data frame de panel
# Chaque région apparaît deux fois : avant (2019) et après (2021)
data_did <- tibble(
  region_id = rep(1:n_regions, 2),
  annee     = rep(c(2019, 2021), each = n_regions),  # Avant et après
  traite    = rep(c(rep(1, n_regions / 2),            # Moitié traitée
                    rep(0, n_regions / 2)), 2)
) |>
  mutate(
    apres      = if_else(annee == ___, 1, 0),   # TODO: quelle année = "après" ?
    traitement = ___ * ___,                      # TODO: interaction traite x
apres

    # Effet vrai : +8 points de % d'emploi pour les traités après
    taux_emploi = 55 + 5 * traite + 3 * apres + ___ * traitement +
                  rnorm(2 * n_regions, mean = 0, sd = 4)
                  # TODO: mettez le vrai effet = 8
  )

# Vérification
head(data_did, 8)
```

i Indice 1 Si vous êtes bloqué

Rappelez-vous la structure DiD : vous avez besoin de (1) une variable indiquant si l'observation est dans la période « après », (2) une variable d'interaction qui vaut 1 seulement pour les traités **et** après (c'est l'interaction). L'effet du programme est codé dans le coefficient de cette interaction.

i Indice 2 Indice plus détaillé

```
# "apres" vaut 1 quand annee == 2021 (la période post-traitement)
apres = if_else(annee == 2021, 1, 0)

# "traitement" = interaction traite × apres
# Ce produit ne vaut 1 que si traite=1 ET apres=1, soit : traités APRÈS le
programme
traitement = traite * apres

# Le vrai effet est 8 points de % → c'est le coefficient de traitement dans la
formule
taux_emploi = 55 + 5 * traite + 3 * apres + 8 * traitement + bruit
```

Voir la solution

```
data_did <- tibble(
  region_id = rep(1:n_regions, 2),
  annee     = rep(c(2019, 2021), each = n_regions),
  traite    = rep(c(rep(1, n_regions / 2),
                    rep(0, n_regions / 2)), 2)
) |>
  mutate(
    apres      = if_else(annee == 2021, 1, 0),
    traitement = traite * apres,
    taux_emploi = 55 + 5 * traite + 3 * apres + 8 * traitement +
                  rnorm(2 * n_regions, mean = 0, sd = 4)
  )
```

Interprétation : Vous avez simulé des données où les régions traitées partent d'un taux d'emploi légèrement supérieur (+5 points) à la ligne de base, toutes les régions connaissent une hausse tendancielle de +3 points, et le programme *Formation Compétences+* ajoute +8 points supplémentaires pour les régions bénéficiaires. Le bruit aléatoire (sd=4) simule les variations non-observées réelles.

Point clé : Simuler des données avec un « vrai » effet connu est une façon puissante de vérifier que votre code R retrouve bien cet effet — si votre estimateur donne environ 8, tout fonctionne correctement.

Exercice 2.2 Calculer les moyennes par cellule

```
{webr-r}
# TODO : Calculez la moyenne de taux_emploi pour chacune des 4 cellules
# (traite=0/1 x apres=0/1)

# Si vous n'avez pas encore le data_did, recréez-le ici :
set.seed(2024)
n_regions <- 100
data_did <- tibble(
  region_id = rep(1:n_regions, 2),
  annee = rep(c(2019, 2021), each = n_regions),
  traite = rep(c(rep(1, 50), rep(0, 50)), 2)
) |>
  mutate(apres = if_else(annee == 2021, 1, 0),
         traitement = traite * apres,
         taux_emploi = 55 + 5*traite + 3*apres + 8*traitement + rnorm(200, 0, 4))

moyennes <- data_did |>
  group_by(___, ___) |>                          # TODO: grouper par traite et
apres
  summarise(moy = mean(___), .groups = "drop")   # TODO: calculer la moyenne

print(moyennes)

# TODO : Calculez l'estimateur DiD manuellement à partir de ces moyennes
# DiD = (Y_11 - Y_10) - (Y_01 - Y_00)
Y_11 <- moyennes |> filter(traite == ___, apres == ___) |> pull(moy)  # traité,
```

```
après
Y_10 <- moyennes |> filter(traite == ___, apres == ___) |> pull(moy)  # traité,
avant
Y_01 <- moyennes |> filter(traite == ___, apres == ___) |> pull(moy)  # contrôle,
après
Y_00 <- moyennes |> filter(traite == ___, apres == ___) |> pull(moy)  # contrôle,
avant

did_manuel <- (Y_11 - Y_10) - (Y_01 - Y_00)
cat("\nEstimateur DiD (manuel) :", round(did_manuel, 2), "points de %\n")
cat("Vrai effet             :", 8, "points de %\n")
```

Voir la solution

```
moyennes <- data_did |>
  group_by(traite, apres) |>
  summarise(moy = mean(taux_emploi), .groups = "drop")

Y_11 <- moyennes |> filter(traite == 1, apres == 1) |> pull(moy)
Y_10 <- moyennes |> filter(traite == 1, apres == 0) |> pull(moy)
Y_01 <- moyennes |> filter(traite == 0, apres == 1) |> pull(moy)
Y_00 <- moyennes |> filter(traite == 0, apres == 0) |> pull(moy)

did_manuel <- (Y_11 - Y_10) - (Y_01 - Y_00)
# Résultat attendu : environ 8 (avec variation aléatoire)
```

Interprétation : Le calcul à 4 cellules est la traduction arithmétique exacte de la logique DiD : on mesure le changement dans le groupe traité, on soustrait le changement dans le groupe contrôle. Le résultat devrait être proche de 8 points de %, avec une légère variation due au bruit aléatoire dans la simulation.

Point clé : Cette formule à 4 moyennes est **mathématiquement équivalente** à la régression DiD du prochain exercice. La régression ajoute simplement des erreurs standard et la possibilité d'inclure des covariables.

Estimation par régression

Exercice 2.3 Modèle DiD en régression

```
{webr-r}
# Recréation des données
set.seed(2024); n_regions <- 100
data_did <- tibble(
  region_id = rep(1:n_regions, 2),
  annee = rep(c(2019, 2021), each = n_regions),
  traite = rep(c(rep(1, 50), rep(0, 50)), 2)
) |> mutate(apres = if_else(annee == 2021, 1, 0), traitement = traite * apres,
            taux_emploi = 55 + 5*traite + 3*apres + 8*traitement + rnorm(200, 0,
4))

# TODO : Estimez le modèle DiD par régression OLS
```

```
# Y = α + β*traite + γ*apres + δ*(traite × apres) + ε
# δ est l'estimateur DiD

modele_did <- lm(___ ~ ___ + ___ + ___:___,  # TODO: complétez la formule
                 data = data_did)

summary(modele_did)
```

Indice 1 Si vous êtes bloqué

La formule de régression DiD suit toujours le même schéma : `Y ~ groupe + temps + groupe:temps`. Ici `Y = taux_emploi`, `groupe = traite`, `temps = apres`. L'interaction `traite:apres` en R s'écrit avec le signe `:`.

Voir la solution

```
modele_did <- lm(taux_emploi ~ traite + apres + traite:apres,
                 data = data_did)
summary(modele_did)

# Lecture des coefficients :
# (Intercept)  ≈ 55 → taux moyen du groupe contrôle AVANT
# traite       ≈  5 → les régions traitées avaient +5% d'emploi au départ
# apres        ≈  3 → tendance temporelle commune (+3% pour tout le monde)
# traite:apres ≈  8 → EFFET DU PROGRAMME = notre estimateur DiD
```

Interprétation : Le coefficient `traite:apres` est l'estimateur DiD. Sa valeur attendue est environ 8 points de %, proche du « vrai effet » programmé dans la simulation. La p-value très faible confirme que cet effet est statistiquement significatif. Dans un rapport pour le Ministère de l'Emploi du Burkina Faso, vous concluriez : « Le programme *Formation Compétences+* a augmenté le taux d'emploi formel de 8 points de pourcentage dans les régions bénéficiaires. »

Point clé : L'estimateur DiD est le coefficient de l'**interaction** `traite × apres`, pas le coefficient de `traite` seul (qui mesure une différence initiale) ni de `apres` seul (qui mesure la tendance temporelle).

Exercice 2.4 Interpréter les coefficients

```
{webr-r}
# Si vous n'avez pas encore le modèle, recréez-le ici
set.seed(2024); n_regions <- 100
data_did <- tibble(
  region_id = rep(1:n_regions, 2),
  annee = rep(c(2019, 2021), each = n_regions),
  traite = rep(c(rep(1, 50), rep(0, 50)), 2)
) |> mutate(apres = if_else(annee == 2021, 1, 0), traitement = traite * apres,
            taux_emploi = 55 + 5*traite + 3*apres + 8*traitement + rnorm(200, 0,
4))
modele_did <- lm(taux_emploi ~ traite + apres + traite:apres, data = data_did)
```

```
library(broom)
resultats <- tidy(modele_did)

# TODO : Extrayez et interprétez chaque coefficient
# Complétez les phrases ci-dessous en remplaçant les ___

intercept  <- resultats |> filter(term == "(Intercept)") |> pull(estimate) |>
round(1)
coef_traite <- resultats |> filter(term == "traite")      |> pull(estimate) |>
round(1)
coef_apres  <- resultats |> filter(term == "apres")       |> pull(estimate) |>
round(1)
coef_did    <- resultats |> filter(term == "traite:apres") |> pull(estimate) |>
round(1)

cat("Intercept (α) =", intercept,   "→ Taux d'emploi moyen du groupe ___  en
___\n")
cat("β =", coef_traite, "→ Les régions traitées avaient ___ % de plus AVANT le
programme\n")
cat("γ =", coef_apres,  "→ Tendance temporelle commune : hausse de ___ % pour
tous\n")
cat("δ =", coef_did,    "→ EFFET DU PROGRAMME : ___ % d'emploi supplémentaire\n")
```

Voir la solution

```
cat("Intercept (α) =", intercept,   "→ Taux d'emploi moyen du groupe CONTRÔLE
en 2019\n")
cat("β =", coef_traite, "→ Les régions traitées avaient", coef_traite,"% de
plus AVANT le programme\n")
cat("γ =", coef_apres,  "→ Tendance temporelle commune : hausse de",
coef_apres, "% pour tous\n")
cat("δ =", coef_did,    "→ EFFET DU PROGRAMME :", coef_did, "% d'emploi
supplémentaire\n")
```

Interprétation : Les quatre coefficients racontent une histoire cohérente : (α) les régions contrôles partaient d'un taux d'emploi de 55 % en 2019 ; (β) les régions ciblées par le programme avaient déjà 5 % de plus au départ, ce qui confirme qu'elles n'étaient pas identiques aux contrôles ; (γ) le marché de l'emploi s'est amélioré de 3 % pour tout le monde entre 2019 et 2021 ; (δ) le programme a ajouté 8 % supplémentaires au-delà de cette tendance commune.

Point clé : Le coefficient β (« différence initiale entre groupes ») n'est PAS un problème pour la DiD. La DiD contrôle précisément ces différences de niveaux. Ce qui compte, c'est que les tendances soient parallèles — pas les niveaux de départ.

Visualisation

Exercice 2.5 Graphique DiD

```
{webr-r}
# Recréation des données
set.seed(2024); n_regions <- 100
data_did <- tibble(
  region_id = rep(1:n_regions, 2),
  annee = rep(c(2019, 2021), each = n_regions),
  traite = rep(c(rep(1, 50), rep(0, 50)), 2)
) |> mutate(apres = if_else(annee == 2021, 1, 0), traitement = traite * apres,
            taux_emploi = 55 + 5*traite + 3*apres + 8*traitement + rnorm(200, 0,
4))

# TODO : Créez un graphique montrant les tendances avant/après pour les deux
groupes
# 1. Calculez les moyennes par groupe et par année
# 2. Ajoutez une ligne "contrefactuelle" pour le groupe traité
# 3. Indiquez la taille de l'effet DiD

moyennes_graph <- data_did |>
  group_by(___, ___) |>            # TODO: grouper par traite et annee
  summarise(moy = mean(taux_emploi), .groups = "drop") |>
  mutate(groupe = if_else(traite == 1, "Traité", "Contrôle"))

ggplot(moyennes_graph, aes(x = annee, y = moy, color = groupe, group = groupe)) +
  geom_line(linewidth = ___) +          # TODO: taille de la ligne (essayez 1.3)
  geom_point(size = ___) +         # TODO: taille des points (essayez 4)
  # Ajoutez une ligne contrefactuelle (ligne pointillée de 2019 traité → 2021
contrôle + tendance)
  labs(
    title = "TODO: Titre du graphique",
    x     = "TODO: label axe x",
    y     = "TODO: label axe y",
    color = "Groupe"
  ) +
  theme_minimal(base_size = 14)
```

Voir la solution

```
moyennes_graph <- data_did |>
  group_by(traite, annee) |>
  summarise(moy = mean(taux_emploi), .groups = "drop") |>
  mutate(groupe = if_else(traite == 1, "Traité", "Contrôle"))

# Valeurs pour la ligne contrefactuelle
moy_traite_avant   <- moyennes_graph |> filter(traite==1, annee==2019) |>
pull(moy)
moy_controle_avant <- moyennes_graph |> filter(traite==0, annee==2019) |>
pull(moy)
moy_controle_apres <- moyennes_graph |> filter(traite==0, annee==2021) |>
pull(moy)
tendance_commune <- moy_controle_apres - moy_controle_avant
contrefactuel_2021 <- moy_traite_avant + tendance_commune

ggplot(moyennes_graph, aes(x = annee, y = moy, color = groupe, group = groupe))
+
  geom_line(linewidth = 1.3) +
  geom_point(size = 4) +
  # Ligne contrefactuelle (pointillée)
  annotate("segment",
           x = 2019, xend = 2021,
           y = moy_traite_avant, yend = contrefactuel_2021,
           linetype = "dashed", color = "#C2185B", linewidth = 1) +
  # Flèche pour l'effet
  annotate("segment",
           x = 2021.05, xend = 2021.05,
           y = contrefactuel_2021, yend =
max(moyennes_graph$moy[moyennes_graph$annee==2021]),
           arrow = arrow(ends = "both", length = unit(0.25, "cm")),
           color = "#C2185B", linewidth = 1) +
  scale_color_manual(values = c("#283593", "#C2185B")) +
  labs(title = "Estimateur DiD — Programme de formation professionnelle",
       subtitle = "La ligne pointillée représente le contrefactuel (sans
programme)",
       x = "Année", y = "Taux d'emploi moyen (%)", color = "Groupe") +
  theme_minimal(base_size = 14)
```

Exercice de synthèse

Exercice 2.6 Évaluation complète

Contexte : La Direction Générale des Collectivités Territoriales du Cameroun souhaite évaluer l'impact de sa réforme de décentralisation fiscale *Commune Autonome 2020*. Cette réforme, mise en oeuvre en 2020 dans 40 communes pilotes sélectionnées parmi 80 communes éligibles, a transféré de nouveaux pouvoirs de taxation locale (taxe foncière, taxe sur les marchés). L'objectif est d'augmenter l'autonomie financière des communes, mesurée par la part des recettes propres dans leur budget total. Le gouvernement veut savoir si cette réforme pilote mérite d'être généralisée aux 360 communes restantes du pays.

```
{webr-r}
# Simulez les données vous-même en vous basant sur les informations suivantes :
# - n = 80 communes (40 traitées, 40 contrôles)
# - Périodes : 2018 (avant) et 2022 (après)
# - Niveau de base (contrôle, avant) : 12 %
# - Différence initiale traités vs contrôles : +2 %
# - Tendance temporelle commune : +1.5 %
# - Effet de la réforme : +6 %
# - Bruit aléatoire : sd = 3

set.seed(123)
n <- ___           # nombre total de communes (2 fois 40 = 80)

data_reforme <- tibble(
  commune_id = rep(1:(n/2 * 2), ___),       # répété pour les 2 périodes
  annee      = rep(c(2018, 2022), each = ___),
  traite     = rep(c(rep(1, n/2), rep(0, n/2)), ___)
) |>
  mutate(
    apres         = if_else(annee == ___, 1, 0),
    traitement    = traite * apres,
    recettes_prop = ___ + ___ * traite + ___ * apres + ___ * traitement +
                    rnorm(n * 2, 0, 3)
  )

# Estimez le modèle DiD
modele_reforme <- lm(___ ~ ___ + ___ + ___:___, data = data_reforme)
summary(modele_reforme)
```

Voir la solution complète

```r
set.seed(123)
n <- 80

data_reforme <- tibble(
  commune_id = rep(1:n, 2),
  annee      = rep(c(2018, 2022), each = n),
  traite     = rep(c(rep(1, n/2), rep(0, n/2)), 2)
) |>
  mutate(
    apres         = if_else(annee == 2022, 1, 0),
    traitement    = traite * apres,
    recettes_prop = 12 + 2 * traite + 1.5 * apres + 6 * traitement +
                    rnorm(n * 2, 0, 3)
  )

modele_reforme <- lm(recettes_prop ~ traite + apres + traite:apres, data =
data_reforme)
summary(modele_reforme)

# Lecture attendue :
# Intercept ≈ 12 → niveau de base communes contrôle en 2018
# traite    ≈  2 → communes traitées avaient +2% de recettes propres au départ
# apres     ≈  1.5 → tendance commune de +1.5%
# interaction ≈ 6 → effet de la réforme = +6 points de %
```

Interprétation : La réforme *Commune Autonome 2020* a permis aux communes pilotes d'augmenter leur part de recettes propres de 6 points de pourcentage (de 14 % à 20 % en moyenne), soit une hausse relative de 43 %. Cet effet est statistiquement significatif ($p < 0.01$). La tendance temporelle commune de +1.5 % reflète l'amélioration générale de la capacité fiscale indépendante de la réforme.

Point clé : La significativité statistique (p-value) est nécessaire mais pas suffisante pour justifier l'extension d'une politique. Il faut aussi évaluer la significativité *pratique* : un gain de 6 points est-il suffisant pour justifier les coûts de réplication aux 360 communes ?

Référence : Pour l'évaluation de la décentralisation fiscale en Afrique, voir Bahl & Bird (2008) « Subnational Taxes in Developing Countries. »

Discussion Application à votre contexte

Pensez à une politique publique de votre pays ou secteur que vous souhaiteriez évaluer.

1. Quel serait le groupe traité et le groupe de contrôle ?
2. Quelle serait la variable de résultat ?
3. Pourquoi la DiD serait-elle (ou ne serait-elle pas) appropriée ici ?
4. Quelle menace principale à l'hypothèse de tendances parallèles identifieriez-vous ?

Discutez en binôme pendant 5 minutes, puis partagez avec le groupe.

ℹ Fin du Module 2

Compétences validées :

- ✅ Simuler des données de panel 2×2
- ✅ Calculer l'estimateur DiD par les 4 moyennes
- ✅ Estimer la DiD par régression OLS avec `lm()`
- ✅ Interpréter les 4 coefficients du modèle
- ✅ Visualiser une DiD avec `ggplot2`

Prochain exercice : Module 3 — TWFE avec données de panel réelles (`mpdta`)