```
#
================================================================
============
#
================================================================
============
# MEMOIRE LSM - How does taxation, economic situation and population
#              impact the real estate prices in Belgium?
#
# Authors: Guillaume Wilbers (39872000) & Louis Lafalize (51252000)
# Supervisor: Leonardo Iania
#
# Script: Model 1 - Fiscal pressure on residential real estate prices
# Data: STATBEL quarterly municipal data (2010Q1 - 2025Q3)
#
# Property scope: ALL houses (2/3/4+ facades, excluding apartments)
# Includes: main regressions, robustness checks, regional interactions,
#           and region-by-region separate regressions.
#
================================================================
============
#
================================================================
============


#
================================================================
============
# 1. PACKAGES
#
================================================================
============
library(readxl)
library(dplyr)
library(fixest)
library(modelsummary)
library(ggplot2)
library(skimr)


#
================================================================
```

```
============
# 2. DATA IMPORT
#
================================================================
============
raw <- read_excel("FR_immo_statbel_trimestre_par_commune.xlsx",
                  sheet = "Par commune",
                  skip  = 3)

cat("=== DATA IMPORT ===\n")
cat("Lines imported:", nrow(raw), "\n")
cat("Columns imported:", ncol(raw), "\n\n")



#
================================================================
============
# 3. DATA PREPARATION
#
================================================================
============

# 3.1 Rename columns
#     Houses: columns 6 (n) and 7 (price) = ALL houses (2/3/4+ facades)
#     Apartments: columns 21 (n) and 22 (price)
df <- raw %>%
rename(
  refnis        = 1,
  commune       = 2,
  year          = 3,
  quarter       = 4,
  n_house       = 6,
  price_house   = 7,
  n_apt         = 21,
  price_apt     = 22,
  surch_mun     = 25,
  surch_prov    = 26,
  surch_pit     = 27,
  rd_owner      = 28,
  rd_investor   = 29,
  rate_regional = 30,
  MS            = 31
) %>%
```

```
select(refnis, commune, year, quarter,
       n_house, price_house, n_apt, price_apt,
       surch_mun, surch_prov, surch_pit,
       rd_owner, rd_investor, rate_regional, MS)

# 3.2 Convert types
df <- df %>%
mutate(across(c(year, n_house, price_house, n_apt, price_apt,
                surch_mun, surch_prov, surch_pit,
                rd_owner, rd_investor, rate_regional, MS),
              ~ as.numeric(.)))

# 3.3 Create time variable
df <- df %>%
mutate(
  quarter_num = as.numeric(gsub("Q", "", quarter)),
  period      = paste0(year, "Q", quarter_num),
  time        = year + (quarter_num - 1) / 4
)

# 3.4 Region classification based on REFNIS code
df <- df %>%
mutate(
  refnis_str = sprintf("%05d", as.integer(refnis)),
  p1 = substr(refnis_str, 1, 1),
  p2 = substr(refnis_str, 1, 2),
  region = case_when(
    p2 == "21"              ~ "Bruxelles",
    p1 == "1"               ~ "Flandre",
    p2 %in% c("23", "24")    ~ "Flandre",
    p2 == "25"              ~ "Wallonie",
    p1 == "3"               ~ "Flandre",
    p1 == "4"               ~ "Flandre",
    p1 == "5"               ~ "Wallonie",
    p1 == "6"               ~ "Wallonie",
    p1 == "7"               ~ "Flandre",
    p1 == "8"               ~ "Wallonie",
    p1 == "9"               ~ "Wallonie",
    TRUE                    ~ NA_character_
  )
) %>%
select(-refnis_str, -p1, -p2)
```

```
# Sanity check
cat("=== REGION CLASSIFICATION CHECK ===\n")
test <- df %>%
select(refnis, commune, region) %>%
distinct() %>%
filter(commune %in% c("ANVERS", "BRUXELLES", "ANDERLECHT", "LIEGE",
"NAMUR",
                      "LOUVAIN", "BRUGES", "GAND", "CHARLEROI", "HASSELT",
                      "ARLON", "BRAINE-L'ALLEUD", "GRAMMONT", "WAVRE", "ALOST"))
print(test)


#
================================================================
============
# 4. BUILD ANALYTICAL DATASETS
#
================================================================
============
build_dataset <- function(price_var, n_var, min_transactions = 10) {
df %>%
  filter(
    !is.na(.data[[price_var]]),
    !is.na(.data[[n_var]]),
    .data[[n_var]] >= min_transactions,
    !is.na(MS),
    !is.na(surch_pit),
    !is.na(rd_owner)
  ) %>%
  mutate(log_price = log(.data[[price_var]]))
}

df_houses     <- build_dataset("price_house", "n_house", min_transactions = 10)
df_apartments <- build_dataset("price_apt",   "n_apt",   min_transactions = 10)

cat("\n=== SAMPLES ===\n")
cat("Houses (ALL facades):", nrow(df_houses), "obs.,",
  length(unique(df_houses$refnis)), "communes\n")
cat("Apartments:          ", nrow(df_apartments), "obs.,",
  length(unique(df_apartments$refnis)), "communes\n\n")


#
```

```
# ============================================================================
# 5. DESCRIPTIVE STATISTICS
# ============================================================================
cat("=== DESCRIPTIVE STATISTICS - HOUSES (ALL FACADES) ===\n")
print(summary(df_houses %>% select(price_house, MS, surch_pit,
                           rd_owner, rd_investor, n_house)))

cat("\nCommune coverage (houses):\n")
print(df_houses %>%
    group_by(region) %>%
    summarise(n_obs       = n(),
              n_communes  = n_distinct(refnis),
              avg_quarters = round(n() / n_distinct(refnis), 1)))

cat("\n=== DESCRIPTIVE STATISTICS - APARTMENTS ===\n")
print(summary(df_apartments %>% select(price_apt, MS, surch_pit,
                               rd_owner, rd_investor, n_apt)))

cat("\nCommune coverage (apartments):\n")
print(df_apartments %>%
    group_by(region) %>%
    summarise(n_obs       = n(),
              n_communes  = n_distinct(refnis),
              avg_quarters = round(n() / n_distinct(refnis), 1)))


# ============================================================================
# 6. SCATTER PLOTS - HOUSES
#    Note: no print() calls (to avoid RStudio device errors).
#    Plots are saved directly to PNG files in the working directory.
#    Each ggsave is wrapped in tryCatch so a graphical glitch cannot stop
#    the rest of the script from running.
# ============================================================================

g_h_ms <- ggplot(df_houses, aes(x = log_price, y = MS, colour = region)) +
geom_point(alpha = 0.25, size = 0.8) +
```

```
geom_smooth(method = "lm", se = TRUE) +
labs(title = "Houses: Global Multiplier (MS) vs log(price)",
    subtitle = "Each point = one commune-quarter observation",
    x = "log(median price)", y = "Global Multiplier (MS)",
    colour = "Region") +
theme_minimal() + theme(plot.title = element_text(face = "bold"))
tryCatch(
ggsave("houses_scatter_MS.png", g_h_ms, width = 9, height = 5, dpi = 300),
error = function(e) message("ggsave houses_scatter_MS skipped: ", e$message)
)

g_h_pit <- ggplot(df_houses, aes(x = log_price, y = surch_pit, colour = region)) +
geom_point(alpha = 0.25, size = 0.8) +
geom_smooth(method = "lm", se = TRUE) +
labs(title = "Houses: Municipal surcharge on PIT vs log(price)",
    subtitle = "Each point = one commune-quarter observation",
    x = "log(median price)", y = "Municipal surcharge on PIT (%)",
    colour = "Region") +
theme_minimal() + theme(plot.title = element_text(face = "bold"))
tryCatch(
ggsave("houses_scatter_surch_pit.png", g_h_pit, width = 9, height = 5, dpi = 300),
error = function(e) message("ggsave houses_scatter_surch_pit skipped: ",
e$message)
)

g_h_rdo <- ggplot(df_houses, aes(x = log_price, y = rd_owner, colour = region)) +
geom_point(alpha = 0.25, size = 0.8) +
geom_smooth(method = "lm", se = TRUE) +
labs(title = "Houses: Registration duty (owner-occupier) vs log(price)",
    subtitle = "Each point = one commune-quarter observation",
    x = "log(median price)", y = "Registration duty for owners (decimal)",
    colour = "Region") +
theme_minimal() + theme(plot.title = element_text(face = "bold"))
tryCatch(
ggsave("houses_scatter_rd_owner.png", g_h_rdo, width = 9, height = 5, dpi = 300),
error = function(e) message("ggsave houses_scatter_rd_owner skipped: ",
e$message)
)


#
============================================================
============
```

```
# 7. SCATTER PLOTS - APARTMENTS
#
================================================================
============

g_a_ms <- ggplot(df_apartments, aes(x = log_price, y = MS, colour = region)) +
geom_point(alpha = 0.25, size = 0.8) +
geom_smooth(method = "lm", se = TRUE) +
labs(title = "Apartments: Global Multiplier (MS) vs log(price)",
    subtitle = "Each point = one commune-quarter observation",
    x = "log(median price)", y = "Global Multiplier (MS)",
    colour = "Region") +
theme_minimal() + theme(plot.title = element_text(face = "bold"))
tryCatch(
ggsave("apartments_scatter_MS.png", g_a_ms, width = 9, height = 5, dpi = 300),
error = function(e) message("ggsave apartments_scatter_MS skipped: ",
e$message)
)

g_a_pit <- ggplot(df_apartments, aes(x = log_price, y = surch_pit, colour = region)) +
geom_point(alpha = 0.25, size = 0.8) +
geom_smooth(method = "lm", se = TRUE) +
labs(title = "Apartments: Municipal surcharge on PIT vs log(price)",
    subtitle = "Each point = one commune-quarter observation",
    x = "log(median price)", y = "Municipal surcharge on PIT (%)",
    colour = "Region") +
theme_minimal() + theme(plot.title = element_text(face = "bold"))
tryCatch(
ggsave("apartments_scatter_surch_pit.png", g_a_pit, width = 9, height = 5, dpi =
300),
error = function(e) message("ggsave apartments_scatter_surch_pit skipped: ",
e$message)
)

g_a_rdo <- ggplot(df_apartments, aes(x = log_price, y = rd_owner, colour = region))
+
geom_point(alpha = 0.25, size = 0.8) +
geom_smooth(method = "lm", se = TRUE) +
labs(title = "Apartments: Registration duty (owner-occupier) vs log(price)",
    subtitle = "Each point = one commune-quarter observation",
    x = "log(median price)", y = "Registration duty for owners (decimal)",
    colour = "Region") +
theme_minimal() + theme(plot.title = element_text(face = "bold"))
```

```
tryCatch(
ggsave("apartments_scatter_rd_owner.png", g_a_rdo, width = 9, height = 5, dpi =
300),
error = function(e) message("ggsave apartments_scatter_rd_owner skipped: ",
e$message)
)


#
================================================================
============
# 8. REGRESSION MODELS - HOUSES (ALL FACADES)
#
================================================================
============

cat("\n\n============================================================\
n")
cat("    REGRESSION MODELS - HOUSES (ALL FACADES)\n")
cat("============================================================\n")

cat("\n=== HOUSES - MAIN MODEL ===\n")
mod_h_main <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                    data    = df_houses,
                    weights = ~ n_house)
summary(mod_h_main)

cat("\n=== HOUSES - AUGMENTED WITH rd_investor ===\n")
mod_h_with_investor <- feols(log_price ~ MS + surch_pit + rd_owner + rd_investor |
                               refnis + period,
                             data    = df_houses,
                             weights = ~ n_house)
summary(mod_h_with_investor)

cat("\n=== HOUSES - ROBUSTNESS POST-2018 ===\n")
mod_h_post <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                    data    = filter(df_houses, year >= 2018),
                    weights = ~ n_house)
summary(mod_h_post)

cat("\n=== HOUSES - ROBUSTNESS WITHOUT BRUSSELS ===\n")
mod_h_nobrx <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                     data    = filter(df_houses, region != "Bruxelles"),
```

```
                 weights = ~ n_house)
summary(mod_h_nobrx)

cat("\n=== HOUSES - HETEROGENEITY BY REGION (interaction terms) ===\n")
mod_h_region <- feols(log_price ~ MS:region + surch_pit:region + rd_owner:region |
                        refnis + period,
                      data    = df_houses,
                      weights = ~ n_house)
summary(mod_h_region)


#
================================================================
============
# 9. REGRESSION MODELS - APARTMENTS
#
================================================================
============

cat("\n\n============================================================\
n")
cat("    REGRESSION MODELS - APARTMENTS\n")
cat("============================================================\n")

cat("\n=== APARTMENTS - MAIN MODEL ===\n")
mod_a_main <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                    data    = df_apartments,
                    weights = ~ n_apt)
summary(mod_a_main)

cat("\n=== APARTMENTS - AUGMENTED WITH rd_investor ===\n")
mod_a_with_investor <- feols(log_price ~ MS + surch_pit + rd_owner + rd_investor |
                               refnis + period,
                             data    = df_apartments,
                             weights = ~ n_apt)
summary(mod_a_with_investor)

cat("\n=== APARTMENTS - ROBUSTNESS POST-2018 ===\n")
mod_a_post <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                    data    = filter(df_apartments, year >= 2018),
                    weights = ~ n_apt)
summary(mod_a_post)
```

```
cat("\n=== APARTMENTS - ROBUSTNESS WITHOUT BRUSSELS ===\n")
mod_a_nobrx <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                     data    = filter(df_apartments, region != "Bruxelles"),
                     weights = ~ n_apt)
summary(mod_a_nobrx)

cat("\n=== APARTMENTS - HETEROGENEITY BY REGION (interaction terms)
===\n")
mod_a_region <- feols(log_price ~ MS:region + surch_pit:region + rd_owner:region |
                        refnis + period,
                      data    = df_apartments,
                      weights = ~ n_apt)
summary(mod_a_region)


#
================================================================
============
# 10. POOLED FINAL COMPARISON TABLES (HTML)
#
================================================================
============
modelsummary(
list("Main"             = mod_h_main,
    "With rd_investor" = mod_h_with_investor,
    "Post-2018"        = mod_h_post,
    "Without Brussels" = mod_h_nobrx,
    "By region"        = mod_h_region),
stars    = TRUE,
gof_omit = "AIC|BIC|Log.Lik|RMSE",
output   = "Model1_Houses_FINAL.html",
title    = "Model 1 - Fiscal pressure on median house prices (all facades)"
)

modelsummary(
list("Main"             = mod_a_main,
    "With rd_investor" = mod_a_with_investor,
    "Post-2018"        = mod_a_post,
    "Without Brussels" = mod_a_nobrx,
    "By region"        = mod_a_region),
stars    = TRUE,
gof_omit = "AIC|BIC|Log.Lik|RMSE",
output   = "Model1_Apartments_FINAL.html",
```

```
title    = "Model 1 - Fiscal pressure on median apartment prices"
)


#
================================================================
============
# 11. REGION-BY-REGION REGRESSIONS - HOUSES
# ----------------------------------------------------------------------
# Note: registration duties (rd_owner) are set at the regional level and apply
# uniformly to all communes of a given region at each point in time. They are
# therefore mechanically collinear with the period fixed effects in any single-
# region regression. fixest will automatically remove rd_owner from the model
# in each of the three region-specific specifications, which is the correct
# behaviour. The estimate of rd_owner relies on the cross-region differential
# variation exploited in the heterogeneity specification (Table 11).
#
================================================================
============

cat("\n\n============================================================\
n")
cat("    REGION-BY-REGION REGRESSIONS - HOUSES\n")
cat("============================================================\n")

cat("\n=== HOUSES - BRUSSELS ONLY ===\n")
mod_h_brx <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                   data    = filter(df_houses, region == "Bruxelles"),
                   weights = ~ n_house)
summary(mod_h_brx)

cat("\n=== HOUSES - FLANDERS ONLY ===\n")
mod_h_fla <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                   data    = filter(df_houses, region == "Flandre"),
                   weights = ~ n_house)
summary(mod_h_fla)

cat("\n=== HOUSES - WALLONIA ONLY ===\n")
mod_h_wal <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                   data    = filter(df_houses, region == "Wallonie"),
                   weights = ~ n_house)
summary(mod_h_wal)
```

```
#
================================================================
============
# 12. REGION-BY-REGION REGRESSIONS - APARTMENTS
#
================================================================
============

cat("\n\n============================================================\
n")
cat("    REGION-BY-REGION REGRESSIONS - APARTMENTS\n")
cat("============================================================\n")

cat("\n=== APARTMENTS - BRUSSELS ONLY ===\n")
mod_a_brx <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                   data    = filter(df_apartments, region == "Bruxelles"),
                   weights = ~ n_apt)
summary(mod_a_brx)

cat("\n=== APARTMENTS - FLANDERS ONLY ===\n")
mod_a_fla <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                   data    = filter(df_apartments, region == "Flandre"),
                   weights = ~ n_apt)
summary(mod_a_fla)

cat("\n=== APARTMENTS - WALLONIA ONLY ===\n")
mod_a_wal <- feols(log_price ~ MS + surch_pit + rd_owner | refnis + period,
                   data    = filter(df_apartments, region == "Wallonie"),
                   weights = ~ n_apt)
summary(mod_a_wal)


#
================================================================
============
# 13. REGION-BY-REGION COMPARISON TABLES (HTML)
#
================================================================
============

modelsummary(
list("Brussels" = mod_h_brx,
```

```
    "Flanders" = mod_h_fla,
    "Wallonia" = mod_h_wal),
stars    = TRUE,
gof_omit = "AIC|BIC|Log.Lik|RMSE",
output   = "Model1_Houses_ByRegion.html",
title    = "Model 1 - Houses by region (separate regressions, all houses)"
)

modelsummary(
list("Brussels" = mod_a_brx,
    "Flanders" = mod_a_fla,
    "Wallonia" = mod_a_wal),
stars    = TRUE,
gof_omit = "AIC|BIC|Log.Lik|RMSE",
output   = "Model1_Apartments_ByRegion.html",
title    = "Model 1 - Apartments by region (separate regressions)"
)


#
============================================================
============
# END OF SCRIPT
#
============================================================
============
cat("\n\n=======================================\n")
cat("Script completed successfully.\n")
cat("All tables saved as HTML files in the working directory.\n")
cat("=======================================\n")
```

```
#================================================================
==============
# MODEL 2 v3 — ANALYSE PRINCIPALE
# Méthodologie : variables en log-différence (Δlog X_t = log X_t − log X_{t-1})
#                appliquée aux niveaux bruts ; FE région ; SE clusterisés (HC1)
# Données : fichier v3 avec CstrPI désaisonnalisé
# Réf. : Granger & Newbold (1974) ; Hilbers et al. (2008) ; Andrews (2010)
#
================================================================
==============

suppressMessages({
  library(readxl); library(dplyr); library(tidyr)
  library(lmtest); library(sandwich); library(stargazer); library(zoo)
  library(writexl); library(corrplot); library(ggplot2)
})

WORKDIR <- "/Users/guillaumewilbers/memoirecladuecodev2"
# Source des données : fichier STATBEL original enrichi par l'auteur
# (formules de variation, ajout MedInc depuis statistiques fiscales STATBEL)
# Note : copie locale du fichier original (le chemin Desktop/Sources Mémoire
# pose un problème d'unzip avec les accents sur macOS). L'original reste en
# annexe-source dans Desktop/Sources Mémoire/.
DATA_PATH <- file.path(WORKDIR, "memoire_model2_data_statbel.xlsx")
OUT_TAB <- file.path(WORKDIR, "outputs/v3/tables")
OUT_FIG <- file.path(WORKDIR, "outputs/v3/figures")
dir.create(OUT_TAB, showWarnings = FALSE, recursive = TRUE)
dir.create(OUT_FIG, showWarnings = FALSE, recursive = TRUE)

cat("\n", paste(rep("=", 70), collapse=""), "\n", sep="")
```

```
cat("MODEL 2 v3 — Analyse principale\n")
cat(paste(rep("=", 70), collapse=""), "\n\n", sep="")

# ---- 1) LECTURE DU FICHIER ----
cat("[1] Lecture :", basename(DATA_PATH), "\n")
raw <- read_excel(DATA_PATH, sheet = "Par région", skip = 2)
price_cols <- grep("^prix médian\\(€\\)", names(raw), value = TRUE)
col_price_all <- price_cols[1]
col_price_ord <- price_cols[2]
col_price_apt <- price_cols[4]

data <- raw %>%
  select(
    region   = `localité`,
    year     = `année`,
    quarter  = `période`,
    price            = all_of(col_price_all),
    price_ordinary   = all_of(col_price_ord),
    price_apartment  = all_of(col_price_apt),
    CR       = `Credit rate`,
    CPI      = `CPI`,
    PPI      = `PPI`,
    CstrPI   = `CstrPI`,
    Pop      = `Pop`,
    MedInc   = `MedInc`,
    HS       = `HS`,
    ER       = `ER`
  ) %>%
  filter(!is.na(region)) %>%
```

```
  mutate(
    year = as.integer(year),
    q    = as.integer(gsub("Q", "", quarter)),
    yq   = as.yearqtr(paste0(year, " Q", q))
  ) %>%
  arrange(region, yq)

cat("[1] N observations totales :", nrow(data), "\n")
cat("[1] Régions :", paste(unique(data$region), collapse = " | "), "\n")
cat("[1] Période :", as.character(min(data$yq)), "→", as.character(max(data$yq)),
"\n\n")

# ---- 2) FENÊTRE 2010-2023 ----
data <- data %>% filter(year >= 2010, year <= 2023)
cat("[2] Après fenêtre 2010-2023 : N =", nrow(data), "\n")

# ---- 3) CALCUL DES LOG-DIFFÉRENCES PAR RÉGION ----
# Δlog(X) = log(X_t) - log(X_{t-1}) — transformation stationnaire standard
# (Granger & Newbold, 1974 ; Hilbers et al., 2008 ; Andrews, 2010)
# Note: pour CPI/PPI/CstrPI, on RECALCULE depuis les niveaux bruts pour avoir
# une vraie log-différence et pas une approximation %growth.
data <- data %>%
  group_by(region) %>%
  arrange(yq) %>%
  mutate(
    dlog_P      = log(price) - log(lag(price)),
    dlog_P_ord  = log(price_ordinary) - log(lag(price_ordinary)),
    dlog_P_apt  = log(price_apartment) - log(lag(price_apartment)),
    dlog_CPI    = log(CPI) - log(lag(CPI)),
```

```
    dlog_PPI    = log(PPI) - log(lag(PPI)),
    dlog_CstrPI = log(CstrPI) - log(lag(CstrPI)),
    dlog_Pop    = log(Pop) - log(lag(Pop)),
    dlog_MedInc = log(MedInc) - log(lag(MedInc)),
    dlog_HS     = log(HS) - log(lag(HS)),
    d_CR        = CR - lag(CR),
    d_ER        = ER - lag(ER)
  ) %>%
  ungroup()

# ---- 4) DATASET POUR RÉGRESSION (drop NA) ----
reg_data <- data %>%
  select(region, year, q, yq,
         dlog_P, dlog_P_ord, dlog_P_apt,
         d_CR, dlog_CPI, dlog_PPI, dlog_CstrPI,
         dlog_Pop, dlog_MedInc, dlog_HS, d_ER) %>%
  na.omit()

cat("[4] N final régression :", nrow(reg_data), "\n")
cat("[4] Obs par région :\n")
print(table(reg_data$region))

# Sauvegarde du dataset
write_xlsx(reg_data, file.path(OUT_TAB, "dataset_final.xlsx"))
write.csv(reg_data, file.path(OUT_TAB, "dataset_final.csv"), row.names = FALSE)

# ---- 5) RÉGRESSION PRINCIPALE (toutes maisons) ----
reg_data$region <- factor(reg_data$region)
```

```
formule <- dlog_P ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
                dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region

model_main <- lm(formule, data = reg_data)
clust_se_main <- sqrt(diag(vcovCL(model_main, cluster = ~ region, type = "HC1")))

cat("\n========== MODÈLE PRINCIPAL (toutes maisons) ==========\n")
print(coeftest(model_main, vcov. = vcovCL(model_main, cluster = ~ region, type =
"HC1")))
cat(sprintf("\nR² = %.4f  Adj.R² = %.4f  N = %d\n",
        summary(model_main)$r.squared,
        summary(model_main)$adj.r.squared,
        nrow(reg_data)))

# ---- 6) RÉGRESSIONS BY HOUSING TYPE ----
formule_ord <- dlog_P_ord ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
                    dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region
formule_apt <- dlog_P_apt ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
                    dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region

reg_ord <- reg_data %>% filter(!is.na(dlog_P_ord))
reg_apt <- reg_data %>% filter(!is.na(dlog_P_apt))

model_ord <- lm(formule_ord, data = reg_ord)
clust_se_ord <- sqrt(diag(vcovCL(model_ord, cluster = ~ region, type = "HC1")))
model_apt <- lm(formule_apt, data = reg_apt)
clust_se_apt <- sqrt(diag(vcovCL(model_apt, cluster = ~ region, type = "HC1")))

# ---- 7) STARGAZER : tableau principal ----
```

```
covariate_labels <- c("Δ Mortgage rate",
                      "Δlog CPI", "Δlog PPI",
                      "Δlog CstrPI (seasonally adj.)",
                      "Δlog Population", "Δlog Median income/declaration",
                      "Δlog Household size", "Δ Employment rate")

stargazer(model_main, model_ord, model_apt,
          se = list(clust_se_main, clust_se_ord, clust_se_apt),
          type = "html",
          out = file.path(OUT_TAB, "model2_main_results.html"),
          title = "Model 2 v3 — Main regression (log-difference specification)",
          column.labels = c("All houses", "Ordinary houses", "Apartments"),
          dep.var.labels = "Δlog(Median price), quarterly",
          covariate.labels = covariate_labels,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
          notes = c("Cluster-robust SE (HC1) by region.",
                    "CstrPI is the STATBEL seasonally-adjusted construction price index."),
          digits = 4)

stargazer(model_main, model_ord, model_apt,
          se = list(clust_se_main, clust_se_ord, clust_se_apt),
          type = "text",
          out = file.path(OUT_TAB, "model2_main_results.txt"),
          title = "Model 2 v3 — Main regression",
          column.labels = c("All", "Ordinary", "Apartments"),
          dep.var.labels = "Δlog(Median price)",
          covariate.labels = covariate_labels,
          omit = "region",
```

```
        add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
        digits = 4)

# ---- 8) ROBUSTESSE : sans Pop, avec lag CR ----
reg_data <- reg_data %>%
  arrange(region, year, q) %>%
  group_by(region) %>%
  mutate(d_CR_lag1 = lag(d_CR)) %>%
  ungroup()

formule_nopop <- dlog_P ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
                    dlog_MedInc + dlog_HS + d_ER + region
formule_lag <- dlog_P ~ d_CR + d_CR_lag1 + dlog_CPI + dlog_PPI + dlog_CstrPI +
                  dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region

reg_lag <- reg_data %>% filter(!is.na(d_CR_lag1))

model_nopop <- lm(formule_nopop, data = reg_data)
clust_se_nopop <- sqrt(diag(vcovCL(model_nopop, cluster = ~ region, type =
"HC1")))
model_lag <- lm(formule_lag, data = reg_lag)
clust_se_lag <- sqrt(diag(vcovCL(model_lag, cluster = ~ region, type = "HC1")))

covariate_labels_lag <- c("Δ Mortgage rate",
                          "Δ Mortgage rate (lag 1)",
                          "Δlog CPI", "Δlog PPI",
                          "Δlog CstrPI (seasonally adj.)",
                          "Δlog Population", "Δlog Median income/declaration",
                          "Δlog Household size", "Δ Employment rate")
```

```
stargazer(model_main, model_nopop, model_lag,
          se = list(clust_se_main, clust_se_nopop, clust_se_lag),
          type = "html",
          out = file.path(OUT_TAB, "model2_robustness.html"),
          title = "Model 2 v3 — Robustness checks",
          column.labels = c("Baseline", "No Pop", "With ΔCR lag"),
          dep.var.labels = "Δlog(Median price), quarterly",
          covariate.labels = covariate_labels_lag,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
          digits = 4)

stargazer(model_main, model_nopop, model_lag,
          se = list(clust_se_main, clust_se_nopop, clust_se_lag),
          type = "text",
          out = file.path(OUT_TAB, "model2_robustness.txt"),
          title = "Model 2 v3 — Robustness",
          covariate.labels = covariate_labels_lag,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
          digits = 4)

# ---- 9) COEFFICIENTS STANDARDISÉS ----
sd_X <- sapply(reg_data[, c("d_CR","dlog_CPI","dlog_PPI","dlog_CstrPI",
                            "dlog_Pop","dlog_MedInc","dlog_HS","d_ER")],
               function(x) sd(x, na.rm = TRUE))
sd_Y <- sd(reg_data$dlog_P, na.rm = TRUE)
beta_raw <- coef(model_main)[c("d_CR","dlog_CPI","dlog_PPI","dlog_CstrPI",
```

```
                          "dlog_Pop","dlog_MedInc","dlog_HS","d_ER")]
beta_std <- beta_raw * (sd_X / sd_Y)

std_df <- data.frame(
  Variable = names(beta_std),
  Coef_brut = round(beta_raw, 4),
  SD_X = round(sd_X, 5),
  SD_Y = round(rep(sd_Y, length(beta_raw)), 5),
  Beta_std = round(beta_std, 4),
  Rang = rank(-abs(beta_std))
)
cat("\n========== COEFFICIENTS STANDARDISÉS ==========\n")
print(std_df)
write_xlsx(std_df, file.path(OUT_TAB, "standardized_coefficients.xlsx"))

# Figure : coefficients standardisés
std_df$Variable <- factor(std_df$Variable,
                          levels = std_df$Variable[order(abs(std_df$Beta_std))])
p_std <- ggplot(std_df, aes(x = Variable, y = Beta_std)) +
  geom_bar(stat = "identity", fill = "#4682B4", alpha = 0.85) +
  geom_hline(yintercept = 0, color = "darkred", linetype = "dashed") +
  coord_flip() +
  labs(title = "Standardized coefficients (β*)",
       subtitle = "Each variable's |β*| measures its relative economic importance",
       x = "", y = "β* (standardized)") +
  theme_minimal()
ggsave(file.path(OUT_FIG, "fig_standardized_coefs.png"),
       p_std, width = 9, height = 5, dpi = 300)
```

```
# ---- 10) SAUVEGARDE OBJETS R ----
saveRDS(list(
  main = model_main, main_se = clust_se_main,
  ord = model_ord, ord_se = clust_se_ord,
  apt = model_apt, apt_se = clust_se_apt,
  nopop = model_nopop, nopop_se = clust_se_nopop,
  lag = model_lag, lag_se = clust_se_lag,
  reg_data = reg_data,
  std_df = std_df
), file.path(OUT_TAB, "models_v3.rds"))

cat("\n", paste(rep("=", 70), collapse=""), "\n", sep="")
cat("[v3 / 01] ANALYSE PRINCIPALE TERMINÉE\n")
cat(paste(rep("=", 70), collapse=""), "\n", sep="")
cat("\nOutputs :\n")
cat("  outputs/v3/tables/model2_main_results.{html,txt}\n")
cat("  outputs/v3/tables/model2_robustness.{html,txt}\n")
cat("  outputs/v3/tables/dataset_final.xlsx\n")
cat("  outputs/v3/tables/standardized_coefficients.xlsx\n")
cat("  outputs/v3/tables/models_v3.rds\n")
cat("  outputs/v3/figures/fig_standardized_coefs.png\n")

#
==========================================================================
==============
# MODEL 2 v3 — COMPARAISON RÉGIONALE
# Régression OLS par région avec Newey-West HAC SE
#
==========================================================================
==============
```

```
#
================================================================
==============
# MODEL 2 v3 — ANALYSE PRINCIPALE
# Méthodologie : variables en log-différence (Δlog X_t = log X_t − log X_{t-1})
#                appliquée aux niveaux bruts ; FE région ; SE clusterisés (HC1)
# Données : fichier v3 avec CstrPI désaisonnalisé
# Réf. : Granger & Newbold (1974) ; Hilbers et al. (2008) ; Andrews (2010)
#
================================================================
==============

suppressMessages({
  library(readxl); library(dplyr); library(tidyr)
  library(lmtest); library(sandwich); library(stargazer); library(zoo)
  library(writexl); library(corrplot); library(ggplot2)
})

WORKDIR <- "/Users/guillaumewilbers/memoirecladuecodev2"
# Source des données : fichier STATBEL original enrichi par l'auteur
# (formules de variation, ajout MedInc depuis statistiques fiscales STATBEL)
# Note : copie locale du fichier original (le chemin Desktop/Sources Mémoire
# pose un problème d'unzip avec les accents sur macOS). L'original reste en
# annexe-source dans Desktop/Sources Mémoire/.
DATA_PATH <- file.path(WORKDIR, "memoire_model2_data_statbel.xlsx")
OUT_TAB <- file.path(WORKDIR, "outputs/v3/tables")
OUT_FIG <- file.path(WORKDIR, "outputs/v3/figures")
dir.create(OUT_TAB, showWarnings = FALSE, recursive = TRUE)
dir.create(OUT_FIG, showWarnings = FALSE, recursive = TRUE)
```

```
cat("\n", paste(rep("=", 70), collapse=""), "\n", sep="")
cat("MODEL 2 v3 — Analyse principale\n")
cat(paste(rep("=", 70), collapse=""), "\n\n", sep="")

# ---- 1) LECTURE DU FICHIER ----
cat("[1] Lecture :", basename(DATA_PATH), "\n")
raw <- read_excel(DATA_PATH, sheet = "Par région", skip = 2)
price_cols <- grep("^prix médian\\(€\\)", names(raw), value = TRUE)
col_price_all <- price_cols[1]
col_price_ord <- price_cols[2]
col_price_apt <- price_cols[4]

data <- raw %>%
  select(
    region   = `localité`,
    year     = `année`,
    quarter  = `période`,
    price            = all_of(col_price_all),
    price_ordinary   = all_of(col_price_ord),
    price_apartment  = all_of(col_price_apt),
    CR       = `Credit rate`,
    CPI      = `CPI`,
    PPI      = `PPI`,
    CstrPI   = `CstrPI`,
    Pop      = `Pop`,
    MedInc   = `MedInc`,
    HS       = `HS`,
    ER       = `ER`
  ) %>%
```

```
  filter(!is.na(region)) %>%
  mutate(
    year = as.integer(year),
    q    = as.integer(gsub("Q", "", quarter)),
    yq   = as.yearqtr(paste0(year, " Q", q))
  ) %>%
  arrange(region, yq)

cat("[1] N observations totales :", nrow(data), "\n")
cat("[1] Régions :", paste(unique(data$region), collapse = " | "), "\n")
cat("[1] Période :", as.character(min(data$yq)), "→", as.character(max(data$yq)),
"\n\n")

# ---- 2) FENÊTRE 2010-2023 ----
data <- data %>% filter(year >= 2010, year <= 2023)
cat("[2] Après fenêtre 2010-2023 : N =", nrow(data), "\n")

# ---- 3) CALCUL DES LOG-DIFFÉRENCES PAR RÉGION ----
# Δlog(X) = log(X_t) - log(X_{t-1}) — transformation stationnaire standard
# (Granger & Newbold, 1974 ; Hilbers et al., 2008 ; Andrews, 2010)
# Note: pour CPI/PPI/CstrPI, on RECALCULE depuis les niveaux bruts pour avoir
# une vraie log-différence et pas une approximation %growth.
data <- data %>%
  group_by(region) %>%
  arrange(yq) %>%
  mutate(
    dlog_P      = log(price) - log(lag(price)),
    dlog_P_ord  = log(price_ordinary) - log(lag(price_ordinary)),
    dlog_P_apt  = log(price_apartment) - log(lag(price_apartment)),
```

```
    dlog_CPI    = log(CPI) - log(lag(CPI)),
    dlog_PPI    = log(PPI) - log(lag(PPI)),
    dlog_CstrPI = log(CstrPI) - log(lag(CstrPI)),
    dlog_Pop    = log(Pop) - log(lag(Pop)),
    dlog_MedInc = log(MedInc) - log(lag(MedInc)),
    dlog_HS     = log(HS) - log(lag(HS)),
    d_CR        = CR - lag(CR),
    d_ER        = ER - lag(ER)
  ) %>%
  ungroup()

# ---- 4) DATASET POUR RÉGRESSION (drop NA) ----
reg_data <- data %>%
  select(region, year, q, yq,
         dlog_P, dlog_P_ord, dlog_P_apt,
         d_CR, dlog_CPI, dlog_PPI, dlog_CstrPI,
         dlog_Pop, dlog_MedInc, dlog_HS, d_ER) %>%
  na.omit()

cat("[4] N final régression :", nrow(reg_data), "\n")
cat("[4] Obs par région :\n")
print(table(reg_data$region))

# Sauvegarde du dataset
write_xlsx(reg_data, file.path(OUT_TAB, "dataset_final.xlsx"))
write.csv(reg_data, file.path(OUT_TAB, "dataset_final.csv"), row.names = FALSE)

# ---- 5) RÉGRESSION PRINCIPALE (toutes maisons) ----
reg_data$region <- factor(reg_data$region)
```

```
formule <- dlog_P ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
  dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region

model_main <- lm(formule, data = reg_data)
clust_se_main <- sqrt(diag(vcovCL(model_main, cluster = ~ region, type = "HC1")))

cat("\n========== MODÈLE PRINCIPAL (toutes maisons) ==========\n")
print(coeftest(model_main, vcov. = vcovCL(model_main, cluster = ~ region, type =
"HC1")))
cat(sprintf("\nR² = %.4f  Adj.R² = %.4f  N = %d\n",
            summary(model_main)$r.squared,
            summary(model_main)$adj.r.squared,
            nrow(reg_data)))

# ---- 6) RÉGRESSIONS BY HOUSING TYPE ----
formule_ord <- dlog_P_ord ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
  dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region
formule_apt <- dlog_P_apt ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
  dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region

reg_ord <- reg_data %>% filter(!is.na(dlog_P_ord))
reg_apt <- reg_data %>% filter(!is.na(dlog_P_apt))

model_ord <- lm(formule_ord, data = reg_ord)
clust_se_ord <- sqrt(diag(vcovCL(model_ord, cluster = ~ region, type = "HC1")))
model_apt <- lm(formule_apt, data = reg_apt)
clust_se_apt <- sqrt(diag(vcovCL(model_apt, cluster = ~ region, type = "HC1")))
```

```
# ---- 7) STARGAZER : tableau principal ----
covariate_labels <- c("Δ Mortgage rate",
                      "Δlog CPI", "Δlog PPI",
                      "Δlog CstrPI (seasonally adj.)",
                      "Δlog Population", "Δlog Median income/declaration",
                      "Δlog Household size", "Δ Employment rate")

stargazer(model_main, model_ord, model_apt,
          se = list(clust_se_main, clust_se_ord, clust_se_apt),
          type = "html",
          out = file.path(OUT_TAB, "model2_main_results.html"),
          title = "Model 2 v3 — Main regression (log-difference specification)",
          column.labels = c("All houses", "Ordinary houses", "Apartments"),
          dep.var.labels = "Δlog(Median price), quarterly",
          covariate.labels = covariate_labels,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
          notes = c("Cluster-robust SE (HC1) by region.",
                    "CstrPI is the STATBEL seasonally-adjusted construction price index."),
          digits = 4)

stargazer(model_main, model_ord, model_apt,
          se = list(clust_se_main, clust_se_ord, clust_se_apt),
          type = "text",
          out = file.path(OUT_TAB, "model2_main_results.txt"),
          title = "Model 2 v3 — Main regression",
          column.labels = c("All", "Ordinary", "Apartments"),
          dep.var.labels = "Δlog(Median price)",
          covariate.labels = covariate_labels,
```

```
        omit = "region",
        add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
        digits = 4)

# ---- 8) ROBUSTESSE : sans Pop, avec lag CR ----
reg_data <- reg_data %>%
  arrange(region, year, q) %>%
  group_by(region) %>%
  mutate(d_CR_lag1 = lag(d_CR)) %>%
  ungroup()

formule_nopop <- dlog_P ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
  dlog_MedInc + dlog_HS + d_ER + region
formule_lag <- dlog_P ~ d_CR + d_CR_lag1 + dlog_CPI + dlog_PPI + dlog_CstrPI +
  dlog_Pop + dlog_MedInc + dlog_HS + d_ER + region

reg_lag <- reg_data %>% filter(!is.na(d_CR_lag1))

model_nopop <- lm(formule_nopop, data = reg_data)
clust_se_nopop <- sqrt(diag(vcovCL(model_nopop, cluster = ~ region, type =
"HC1")))
model_lag <- lm(formule_lag, data = reg_lag)
clust_se_lag <- sqrt(diag(vcovCL(model_lag, cluster = ~ region, type = "HC1")))

covariate_labels_lag <- c("Δ Mortgage rate",
                          "Δ Mortgage rate (lag 1)",
                          "Δlog CPI", "Δlog PPI",
                          "Δlog CstrPI (seasonally adj.)",
                          "Δlog Population", "Δlog Median income/declaration",
```

```
                    "Δlog Household size", "Δ Employment rate")

stargazer(model_main, model_nopop, model_lag,
          se = list(clust_se_main, clust_se_nopop, clust_se_lag),
          type = "html",
          out = file.path(OUT_TAB, "model2_robustness.html"),
          title = "Model 2 v3 — Robustness checks",
          column.labels = c("Baseline", "No Pop", "With ΔCR lag"),
          dep.var.labels = "Δlog(Median price), quarterly",
          covariate.labels = covariate_labels_lag,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
          digits = 4)

stargazer(model_main, model_nopop, model_lag,
          se = list(clust_se_main, clust_se_nopop, clust_se_lag),
          type = "text",
          out = file.path(OUT_TAB, "model2_robustness.txt"),
          title = "Model 2 v3 — Robustness",
          covariate.labels = covariate_labels_lag,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "Yes", "Yes")),
          digits = 4)

# ---- 9) COEFFICIENTS STANDARDISÉS ----
sd_X <- sapply(reg_data[, c("d_CR","dlog_CPI","dlog_PPI","dlog_CstrPI",
                            "dlog_Pop","dlog_MedInc","dlog_HS","d_ER")],
               function(x) sd(x, na.rm = TRUE))
sd_Y <- sd(reg_data$dlog_P, na.rm = TRUE)
```

```
beta_raw <- coef(model_main)[c("d_CR","dlog_CPI","dlog_PPI","dlog_CstrPI",
                               "dlog_Pop","dlog_MedInc","dlog_HS","d_ER")]
beta_std <- beta_raw * (sd_X / sd_Y)

std_df <- data.frame(
  Variable = names(beta_std),
  Coef_brut = round(beta_raw, 4),
  SD_X = round(sd_X, 5),
  SD_Y = round(rep(sd_Y, length(beta_raw)), 5),
  Beta_std = round(beta_std, 4),
  Rang = rank(-abs(beta_std))
)
cat("\n========== COEFFICIENTS STANDARDISÉS ==========\n")
print(std_df)
write_xlsx(std_df, file.path(OUT_TAB, "standardized_coefficients.xlsx"))

# Figure : coefficients standardisés
std_df$Variable <- factor(std_df$Variable,
                          levels = std_df$Variable[order(abs(std_df$Beta_std))])
p_std <- ggplot(std_df, aes(x = Variable, y = Beta_std)) +
  geom_bar(stat = "identity", fill = "#4682B4", alpha = 0.85) +
  geom_hline(yintercept = 0, color = "darkred", linetype = "dashed") +
  coord_flip() +
  labs(title = "Standardized coefficients (β*)",
       subtitle = "Each variable's |β*| measures its relative economic importance",
       x = "", y = "β* (standardized)") +
  theme_minimal()
ggsave(file.path(OUT_FIG, "fig_standardized_coefs.png"),
       p_std, width = 9, height = 5, dpi = 300)
```

```
# ---- 10) SAUVEGARDE OBJETS R ----
saveRDS(list(
  main = model_main, main_se = clust_se_main,
  ord = model_ord, ord_se = clust_se_ord,
  apt = model_apt, apt_se = clust_se_apt,
  nopop = model_nopop, nopop_se = clust_se_nopop,
  lag = model_lag, lag_se = clust_se_lag,
  reg_data = reg_data,
  std_df = std_df
), file.path(OUT_TAB, "models_v3.rds"))

cat("\n", paste(rep("=", 70), collapse=""), "\n", sep="")
cat("[v3 / 01] ANALYSE PRINCIPALE TERMINÉE\n")
cat(paste(rep("=", 70), collapse=""), "\n", sep="")
cat("\nOutputs :\n")
cat("  outputs/v3/tables/model2_main_results.{html,txt}\n")
cat("  outputs/v3/tables/model2_robustness.{html,txt}\n")
cat("  outputs/v3/tables/dataset_final.xlsx\n")
cat("  outputs/v3/tables/standardized_coefficients.xlsx\n")
cat("  outputs/v3/tables/models_v3.rds\n")
cat("  outputs/v3/figures/fig_standardized_coefs.png\n")

suppressMessages({
  library(readxl); library(dplyr); library(tidyr)
  library(lmtest); library(sandwich); library(stargazer); library(zoo)
  library(writexl); library(ggplot2); library(RColorBrewer)
})
```

```
WORKDIR <- "/Users/guillaumewilbers/memoirecladuecodev2"
OUT_TAB <- file.path(WORKDIR, "outputs/v3/tables")
OUT_FIG <- file.path(WORKDIR, "outputs/v3/figures")

# Reprend les données depuis l'analyse principale
models_obj <- readRDS(file.path(OUT_TAB, "models_v3.rds"))
reg_data <- as.data.frame(models_obj$reg_data)

regions     <- c("REGION DE BRUXELLES-CAPITALE",
                 "REGION FLAMANDE",
                 "REGION WALLONNE")
short_names <- c("Brussels", "Flemish", "Walloon")

# Newey-West SE helper
nw_se <- function(model, lag = 1) {
  sqrt(diag(NeweyWest(model, lag = lag, prewhite = FALSE, adjust = TRUE)))
}

# ---- 1) HOUSES — par région ----
formule_h <- dlog_P ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
                      dlog_Pop + dlog_MedInc + dlog_HS + d_ER

models_houses <- lapply(regions, function(r) {
  lm(formule_h, data = subset(reg_data, region == r))
})
names(models_houses) <- short_names
se_houses <- lapply(models_houses, function(m) nw_se(m, lag = 1))

for (i in seq_along(models_houses)) {
```

```
  cat("\n========== HOUSES —", short_names[i], "==========\n")
  m <- models_houses[[i]]
  cat("N obs :", nobs(m), "\n")
  cat("R²    :", round(summary(m)$r.squared, 4), "\n")
  cat("Adj R²:", round(summary(m)$adj.r.squared, 4), "\n\n")
  print(coeftest(m, vcov. = NeweyWest(m, lag = 1, prewhite = FALSE, adjust =
TRUE)))
}

# Stargazer comparatif
covariate_labels <- c("Δ Mortgage rate", "Δlog CPI", "Δlog PPI",
                      "Δlog CstrPI (S.A.)", "Δlog Population",
                      "Δlog Median income", "Δlog Household size",
                      "Δ Employment rate")

# Pooled model pour colonne 1
pooled <- models_obj$main
clust_pooled <- models_obj$main_se

stargazer(pooled, models_houses[[1]], models_houses[[2]], models_houses[[3]],
          se = c(list(clust_pooled), se_houses),
          type = "html",
          out = file.path(OUT_TAB, "model2_houses_by_region.html"),
          title = "Model 2 v3 — Houses by region",
          column.labels = c("Pooled (FE)", "Brussels", "Flemish", "Walloon"),
          dep.var.labels = "Δlog(Median price), quarterly",
          covariate.labels = covariate_labels,
          omit = "region",
          add.lines = list(c("Region FE", "Yes", "No", "No", "No")),
```

```
        notes = c("Column (1): pooled with regional FE, clustered SE (HC1).",
                  "Columns (2)-(4): region-specific OLS with Newey-West HAC SE (lag
1)."),
        digits = 4)

stargazer(pooled, models_houses[[1]], models_houses[[2]], models_houses[[3]],
        se = c(list(clust_pooled), se_houses),
        type = "text",
        out = file.path(OUT_TAB, "model2_houses_by_region.txt"),
        column.labels = c("Pooled", "Brussels", "Flemish", "Walloon"),
        covariate.labels = covariate_labels,
        omit = "region", digits = 4)

# ---- 2) APARTMENTS — par région ----
formule_a <- dlog_P_apt ~ d_CR + dlog_CPI + dlog_PPI + dlog_CstrPI +
                    dlog_Pop + dlog_MedInc + dlog_HS + d_ER

reg_apt <- reg_data %>% filter(!is.na(dlog_P_apt))
models_apt <- lapply(regions, function(r) {
  lm(formule_a, data = subset(reg_apt, region == r))
})
names(models_apt) <- short_names
se_apt <- lapply(models_apt, function(m) nw_se(m, lag = 1))

pooled_apt <- models_obj$apt
clust_pooled_apt <- models_obj$apt_se

stargazer(pooled_apt, models_apt[[1]], models_apt[[2]], models_apt[[3]],
        se = c(list(clust_pooled_apt), se_apt),
```

```
        type = "html",
        out = file.path(OUT_TAB, "model2_apartments_by_region.html"),
        title = "Model 2 v3 — Apartments by region",
        column.labels = c("Pooled (FE)", "Brussels", "Flemish", "Walloon"),
        dep.var.labels = "Δlog(Median price), quarterly",
        covariate.labels = covariate_labels,
        omit = "region",
        add.lines = list(c("Region FE", "Yes", "No", "No", "No")),
        digits = 4)

# ---- 3) FIGURE COMPARATIVE T-STATS ----
extract_tstats <- function(model, region) {
  coefs <- summary(model)$coefficients
  se <- nw_se(model, lag = 1)
  data.frame(
    region = region,
    variable = rownames(coefs),
    estimate = coefs[, "Estimate"],
    se = se,
    t_stat = coefs[, "Estimate"] / se,
    stringsAsFactors = FALSE
  )
}

tstat_houses <- do.call(rbind, lapply(seq_along(models_houses), function(i) {
  extract_tstats(models_houses[[i]], short_names[i])
})) %>% filter(variable != "(Intercept)") %>% mutate(property = "Houses")

tstat_apt <- do.call(rbind, lapply(seq_along(models_apt), function(i) {
```

```
  extract_tstats(models_apt[[i]], short_names[i])
})) %>% filter(variable != "(Intercept)") %>% mutate(property = "Apartments")

var_relabel <- c(
  "d_CR" = "Δ Mortgage rate",
  "dlog_CPI" = "Δlog CPI", "dlog_PPI" = "Δlog PPI",
  "dlog_CstrPI" = "Δlog CstrPI (S.A.)",
  "dlog_Pop" = "Δlog Population",
  "dlog_MedInc" = "Δlog Median income",
  "dlog_HS" = "Δlog Household size",
  "d_ER" = "Δ Employment rate"
)

tstat_combined <- rbind(tstat_houses, tstat_apt) %>%
  mutate(variable_lbl = var_relabel[variable],
         region = factor(region, levels = c("Brussels", "Flemish", "Walloon")),
         property = factor(property, levels = c("Houses", "Apartments")))

p_combined <- ggplot(tstat_combined, aes(x = variable_lbl, y = t_stat,
                                         color = region, shape = region)) +
  geom_point(position = position_dodge(width = 0.6), size = 3) +
  geom_hline(yintercept = c(-1.96, 1.96), linetype = "dashed", color = "darkred") +
  geom_hline(yintercept = 0, color = "gray60") +
  coord_flip() +
  facet_wrap(~ property, nrow = 1) +
  scale_color_brewer(palette = "Set1") +
  labs(title = "Model 2 v3 — t-statistics by region and property type",
       subtitle = "Newey-West HAC SE (lag 1). Dashed red lines: ±1.96 (5%
significance)",
```

```
      x = "", y = "t-statistic",
      color = "Region", shape = "Region") +
  theme_minimal() +
  theme(legend.position = "bottom")

ggsave(file.path(OUT_FIG, "fig_regional_tstats.png"),
       p_combined, width = 13, height = 6, dpi = 300)

# ---- 4) EXPORT COEFFICIENTS ----
coefs_export <- tstat_combined %>%
  mutate(estimate = round(estimate, 4),
         se = round(se, 4),
         t_stat = round(t_stat, 3),
         significant = ifelse(abs(t_stat) > 1.96, "yes", "no")) %>%
  select(property, region, variable_lbl, estimate, se, t_stat, significant) %>%
  rename(Property = property, Region = region, Variable = variable_lbl,
         Estimate = estimate, SE = se, `Significant 5%` = significant)
write_xlsx(coefs_export, file.path(OUT_TAB, "regional_coefficients.xlsx"))

cat("\n[v3 / 02] COMPARAISON RÉGIONALE TERMINÉE\n")
cat("  outputs/v3/tables/model2_houses_by_region.{html,txt}\n")
cat("  outputs/v3/tables/model2_apartments_by_region.html\n")
cat("  outputs/v3/tables/regional_coefficients.xlsx\n")
cat("  outputs/v3/figures/fig_regional_tstats.png\n")
```

```
#
================================================================
==============
# MODEL 2 v3 — TABLEAUX & FIGURES descriptifs
#
================================================================
==============

suppressMessages({
  library(readxl); library(dplyr); library(tidyr)
  library(writexl); library(ggplot2); library(corrplot); library(patchwork)
})

WORKDIR <- "/Users/guillaumewilbers/memoirecladuecodev2"
# Source des données : copie locale du fichier STATBEL original
DATA_PATH <- file.path(WORKDIR, "memoire_model2_data_statbel.xlsx")
OUT_TAB <- file.path(WORKDIR, "outputs/v3/tables")
OUT_FIG <- file.path(WORKDIR, "outputs/v3/figures")
dir.create(OUT_TAB, showWarnings = FALSE, recursive = TRUE)
dir.create(OUT_FIG, showWarnings = FALSE, recursive = TRUE)

out_t <- function(f) file.path(OUT_TAB, f)
out_f <- function(f) file.path(OUT_FIG, f)

# ---- Charger objets de l'analyse principale ----
models_obj <- readRDS(out_t("models_v3.rds"))
reg_data <- as.data.frame(models_obj$reg_data)

# Lire les niveaux pour stats descriptives
raw <- read_excel(DATA_PATH, sheet = "Par région", skip = 2)
```

```
price_cols <- grep("^prix médian\\(€\\)", names(raw), value = TRUE)
data_lvl <- raw %>%
  select(region = `localité`, year = `année`, quarter = `période`,
         price = all_of(price_cols[1]),
         CR = `Credit rate`, CPI = `CPI`, PPI = `PPI`,
         CstrPI = `CstrPI`, Pop = `Pop`, MedInc = `MedInc`,
         HS = `HS`, ER = `ER`) %>%
  filter(!is.na(region)) %>%
  mutate(year = as.integer(year)) %>%
  filter(year >= 2010, year <= 2023) %>%
  # Renommer les régions en anglais pour tous les outputs descriptifs
  mutate(region = recode(region,
                         "REGION DE BRUXELLES-CAPITALE" = "Brussels-Capital",
                         "REGION FLAMANDE"              = "Flemish Region",
                         "REGION WALLONNE"              = "Walloon Region"))

# ---- TABLE 1 : Variable definitions ----
var_def <- data.frame(
  Variable = c("P_houses", "P_apartments", "CR",
               "CPI", "PPI", "CstrPI", "Pop", "MedInc", "HS", "ER"),
  Description = c(
    "Median transaction price (all houses, by region and quarter)",
    "Median transaction price (apartments, by region and quarter)",
    "Mortgage interest rate, fixation > 10 years (Belgium, monthly avg.)",
    "Consumer price index, all items (base 2015=100)",
    "Producer price index, industrial total domestic (base 2015=100)",
    "Construction price index, Section F (base 2015=100, seasonally adjusted)",
    "Resident population (mid-year, by region)",
    "Median net taxable income per tax declaration (by region)",
```

```
  "Average household size (persons per household, by region)",
  "Employment rate (15-64, by region)"
),
Unit = c("EUR", "EUR", "% per annum",
         "Index", "Index", "Index",
         "Persons", "EUR / year", "Persons / hh", "Ratio (0-1)"),
Source = c(
  "STATBEL — Real estate stats",
  "STATBEL — Real estate stats",
  "National Bank of Belgium (MIR)",
  "STATBEL — CPI",
  "STATBEL — PPI",
  "STATBEL — Construction PI (S.A.)",
  "STATBEL — Demography",
  "STATBEL — Fiscal statistics",
  "STATBEL — Housing census",
  "STATBEL — LFS"
),
Transformation = c(
  "Δlog (log-difference of levels)",
  "Δlog",
  "Δ (first difference, already in %)",
  "Δlog (quarterly inflation, continuously compounded)",
  "Δlog",
  "Δlog (computed on seasonally-adjusted index)",
  "Δlog",
  "Δlog",
  "Δlog",
  "Δ (first difference, already in %)"
```

```
  ),
  Frequency = c("Quarterly", "Quarterly", "Monthly → quarterly avg",
                "Monthly → quarterly avg", "Monthly → quarterly avg",
                "Monthly → quarterly avg (S.A.)",
                "Annual (linearly interpolated)",
                "Annual (linearly interpolated)",
                "Annual (linearly interpolated)",
                "Quarterly")
)
write_xlsx(var_def, out_t("table1_variable_definitions.xlsx"))
cat("[Table 1] Variable definitions exported.\n")

# ---- TABLE 2 : Descriptive stats (levels, pooled) ----
desc_pooled <- data_lvl %>%
  summarise(across(c(price, CR, CPI, PPI, CstrPI, Pop, MedInc, HS, ER),
                   list(mean = ~mean(.x, na.rm = TRUE),
                        sd   = ~sd(.x,   na.rm = TRUE),
                        min  = ~min(.x,  na.rm = TRUE),
                        p25  = ~quantile(.x, 0.25, na.rm = TRUE),
                        med  = ~median(.x, na.rm = TRUE),
                        p75  = ~quantile(.x, 0.75, na.rm = TRUE),
                        max  = ~max(.x,   na.rm = TRUE))))

desc_long <- desc_pooled %>%
  tidyr::pivot_longer(everything(), names_to = "varstat", values_to = "value") %>%
  tidyr::separate(varstat, into = c("Variable", "Statistic"), sep = "_(?=[^_]+$)")
desc_wide <- desc_long %>%
  tidyr::pivot_wider(names_from = "Statistic", values_from = "value")
```

```
write_xlsx(desc_wide, out_t("table2_descriptive_stats_pooled.xlsx"))
cat("[Table 2] Descriptive stats (pooled) exported.\n")

# Stats par région
desc_region <- data_lvl %>%
  group_by(region) %>%
  summarise(across(c(price, CR, CPI, PPI, CstrPI, Pop, MedInc, HS, ER),
                   list(mean = ~mean(.x, na.rm = TRUE),
                        sd   = ~sd(.x,   na.rm = TRUE))),
            .groups = "drop")
write_xlsx(desc_region, out_t("table2b_descriptive_stats_by_region.xlsx"))

# ---- TABLE 3 : Correlation matrix (variables in log-differences) ----
cor_vars <- reg_data %>%
  select(dlog_P, d_CR, dlog_CPI, dlog_PPI, dlog_CstrPI,
         dlog_Pop, dlog_MedInc, dlog_HS, d_ER)
# Labels académiques propres pour la heatmap
colnames(cor_vars) <- c("Δlog P", "ΔCR", "Δlog CPI", "Δlog PPI", "Δlog CstrPI",
                        "Δlog Pop", "Δlog MedInc", "Δlog HS", "ΔER")
cor_mat <- cor(cor_vars, use = "pairwise.complete.obs")
cor_mat_rounded <- round(cor_mat, 3)

cor_df <- data.frame(Variable = rownames(cor_mat_rounded), cor_mat_rounded,
                     check.names = FALSE)
write_xlsx(cor_df, out_t("table3_correlation_matrix.xlsx"))
cat("[Table 3] Correlation matrix exported.\n")

# Heatmap
```

```
png(out_f("fig_correlation_heatmap.png"), width = 9, height = 7, units = "in", res =
300)
corrplot(cor_mat, method = "color", type = "upper",
         tl.col = "black", tl.cex = 0.85,
         addCoef.col = "black", number.cex = 0.7, diag = FALSE,
         col = colorRampPalette(c("#3B4CC0", "white", "#B40426"))(200),
         title = "Correlation matrix — variables in log-differences",
         mar = c(0, 0, 2, 0))
dev.off()
cat("[Figure] Correlation heatmap exported.\n")

# ---- FIGURE : Prix vs CR (dual axis) ----
data_belgium <- data_lvl %>%
  group_by(year, quarter) %>%
  summarise(price = mean(price, na.rm = TRUE),
            CR = mean(CR, na.rm = TRUE), .groups = "drop") %>%
  mutate(q = as.integer(gsub("Q", "", quarter)),
         yq = as.numeric(year) + (q - 1)/4) %>%
  arrange(yq)

scale_factor <- max(data_belgium$price, na.rm = TRUE) / 5
p_dual <- ggplot(data_belgium, aes(x = yq)) +
  geom_line(aes(y = price, color = "House price (EUR)"), size = 1) +
  geom_line(aes(y = CR * scale_factor, color = "Mortgage rate (%)"), size = 1) +
  scale_y_continuous(name = "House price (EUR)",
                     sec.axis = sec_axis(~ . / scale_factor, name = "Mortgage rate (%)")) +
  scale_color_manual(values = c("House price (EUR)" = "#2E5984",
                                "Mortgage rate (%)" = "#C0392B")) +
  labs(title = "House prices and mortgage rates — Belgium 2010-2023",
```

```
      x = "", color = "") +
  theme_minimal() +
  theme(legend.position = "top")
ggsave(out_f("fig_prices_vs_rates.png"), p_dual, width = 10, height = 5.5, dpi = 300)

# ---- FIGURE : distribution Y ----
p_dep <- ggplot(reg_data, aes(x = dlog_P)) +
  geom_histogram(bins = 30, fill = "#4682B4", color = "white", alpha = 0.85) +
  geom_vline(xintercept = 0, color = "darkred", linetype = "dashed") +
  labs(title = "Distribution of Δlog(Median price), quarterly",
      x = "Δlog(Median price)", y = "Frequency") +
  theme_minimal()
ggsave(out_f("fig_dependent_distribution.png"), p_dep, width = 9, height = 5.5, dpi =
300)

# ---- FIGURE : Residuals vs fitted ----
model_main <- models_obj$main
diag_df <- data.frame(fitted = fitted(model_main), residuals = resid(model_main))
p_res <- ggplot(diag_df, aes(x = fitted, y = residuals)) +
  geom_point(alpha = 0.55, color = "#34495E") +
  geom_hline(yintercept = 0, color = "darkred", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "darkorange") +
  labs(title = "Residuals vs fitted",
      x = "Fitted values", y = "Residuals") +
  theme_minimal()
ggsave(out_f("fig_residuals_vs_fitted.png"), p_res, width = 8, height = 5.5, dpi = 300)

# ---- FIGURE : Q-Q plot ----
p_qq <- ggplot(diag_df, aes(sample = residuals)) +
```

```
  stat_qq(alpha = 0.6) + stat_qq_line(color = "darkred") +
  labs(title = "Q-Q plot of residuals",
       x = "Theoretical quantiles", y = "Sample quantiles") +
  theme_minimal()
ggsave(out_f("fig_qq_residuals.png"), p_qq, width = 7, height = 6, dpi = 300)

# ---- FIGURE : Boxplot prix par région ----
p_box <- ggplot(data_lvl, aes(x = region, y = price/1000, fill = region)) +
  geom_boxplot(alpha = 0.75) +
  labs(title = "Median house prices by region — Belgium 2010-2023",
       x = "", y = "Price (k€)") +
  theme_minimal() +
  theme(legend.position = "none") +
  scale_fill_brewer(palette = "Set2")
ggsave(out_f("fig_boxplot_prices.png"), p_box, width = 9, height = 5.5, dpi = 300)

cat("\n[v3 / 03] TABLEAUX & FIGURES TERMINÉS\n")
cat("  outputs/v3/tables/table1_variable_definitions.xlsx\n")
cat("  outputs/v3/tables/table2_descriptive_stats_pooled.xlsx\n")
cat("  outputs/v3/tables/table3_correlation_matrix.xlsx\n")
cat("  outputs/v3/figures/ (6 PNG files)\n")
```