CBAM's Effect on Great Britain's Electricity Flows

Luca Taschini, Gian Luca Vriz, Joshua Burke

2024-07-08

Libraries

The libraries utilized for the project are delineated below.

```
library(gtools)
```

```
## Warning: il pacchetto 'gtools' è stato creato con R versione 4.2.3
```

```
library(cowplot)
```

```
## Warning: il pacchetto 'cowplot' è stato creato con R versione 4.2.3
```

```
library(plyr)
```

```
## Warning: il pacchetto 'plyr' è stato creato con R versione 4.2.3
```

```
library(viridis)
```

```
## Warning: il pacchetto 'viridis' è stato creato con R versione 4.2.3
```

```
## Caricamento del pacchetto richiesto: viridisLite
```

```
## Warning: il pacchetto 'viridisLite' è stato creato con R versione 4.2.3
```

```
library(FinTS)
```

```
## Warning: il pacchetto 'FinTS' è stato creato con R versione 4.2.3
```

```
## Caricamento del pacchetto richiesto: zoo
```

```
##
## Caricamento pacchetto: 'zoo'
```

```
## I seguenti oggetti sono mascherati da 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(lubridate)
```

```
##
## Caricamento pacchetto: 'lubridate'
```

```
## Il seguente oggetto è mascherato da 'package:cowplot':
##
##     stamp
```

```
## I seguenti oggetti sono mascherati da 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(data.table)
```

```
## Warning: il pacchetto 'data.table' è stato creato con R versione 4.2.3
```

```
##
## Caricamento pacchetto: 'data.table'
```

```
## I seguenti oggetti sono mascherati da 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```r
library(zoo)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Caricamento pacchetto: 'forecast'
```

```
## Il seguente oggetto è mascherato da 'package:FinTS':
##
##     Acf
```

```r
library(lmtest)
library(stringr)
library(zoo)
library(corrplot)
```

```
## Warning: il pacchetto 'corrplot' è stato creato con R versione 4.2.3
```

```
## corrplot 0.92 loaded
```

```r
library(readxl)
```

```
## Warning: il pacchetto 'readxl' è stato creato con R versione 4.2.3
```

```r
library(readr)
```

```
## Warning: il pacchetto 'readr' è stato creato con R versione 4.2.3
```

```r
library(countrycode)
```

```
## Warning: il pacchetto 'countrycode' è stato creato con R versione 4.2.3
```

```r
library(ggplot2)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.2.3
```

```r
library(tidyr)
library(dplyr)
```

```
##
## Caricamento pacchetto: 'dplyr'
```

```
## I seguenti oggetti sono mascherati da 'package:data.table':
##
##     between, first, last
```

```
## I seguenti oggetti sono mascherati da 'package:plyr':
```

```
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## I seguenti oggetti sono mascherati da 'package:stats':
##
##     filter, lag

## I seguenti oggetti sono mascherati da 'package:base':
##
##     intersect, setdiff, setequal, union
```

Data

Initially, electricity prices are uploaded for both the Great Britain (GB) and the European Union (EU).

```
#EPEX spot day ahead UK baseload
Electricity_price_gb <-
  read_csv("Data/Electricity_price_UK_EPEX2.csv", skip = 1)
```

```
## New names:
## Rows: 1826 Columns: 2
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): ...1 dbl (1): Close
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
colnames(Electricity_price_gb) <- c('Date', 'Daily_ave_GB')
Electricity_price_gb$Date <- dmy(Electricity_price_gb$Date)
Electricity_price_gb$Daily_ave_GB <-
  as.numeric(Electricity_price_gb$Daily_ave_GB)
#Period
Electricity_price_gb <-
  Electricity_price_gb[Electricity_price_gb$Date < '2024-01-01' &
                       Electricity_price_gb$Date > '2019-12-31', ]
```

For the EU's electricity price, the countries with the highest electricity trade volumes with the GB are considered. These include:

- France,
- Belgium,
- Denmark,
- Netherlands,
- Ireland.

```
#Electricity price EU
#Raw data daily values
#https://ember-climate.org/data-catalogue/european-wholesale-electricity-price-data/
Electricity_price_EU <-
  read.csv("Data/Electricity_price_EU.csv")
Electricity_price_EU <-
  Electricity_price_EU[Electricity_price_EU$Country == "Belgium" |
                       Electricity_price_EU$Country == "Ireland" |
                       Electricity_price_EU$Country == "Denmark" |
                       Electricity_price_EU$Country == "France" |
                         Electricity_price_EU$Country == "Netherlands", ]
Electricity_price_EU$Date <- as.Date(Electricity_price_EU$Date)
#Average
Electricity_price_EUg <- Electricity_price_EU  %>%
  group_by(Date) %>%
  dplyr::summarise(Average.price = mean(Price..EUR.MWhe.))
colnames(Electricity_price_EUg) <- c('Date', 'Daily_ave_EU')
#Period
Electricity_price_EUg <-
  Electricity_price_EUg[Electricity_price_EUg$Date < '2024-01-01' &
                          Electricity_price_EUg$Date > '2019-12-31', ]
str(Electricity_price_EUg)
```

```
## tibble [1,461 x 2] (S3: tbl_df/tbl/data.frame)
##  $ Date        : Date[1:1461], format: "2020-01-01" "2020-01-02" ...
##  $ Daily_ave_EU: num [1:1461] 34.8 35.1 35.5 32.7 34.6 ...
```

To convert currency between the two regions, the £/£ exchange rate is adopted.

```
#Exchange rate
#Raw data
#https://www.investing.com/currencies/eur-gbp-historical-data
EUR_GBP <-
  read.csv("Data/EUR_GBP_change_data.csv")
EUR_GBP$Date <- dmy(EUR_GBP$Date)
EUR_GBP <- EUR_GBP[, 1:2]
str(EUR_GBP)
```

```
## 'data.frame':    1043 obs. of  2 variables:
##  $ Date : Date, format: "2023-12-29" "2023-12-28" ...
##  $ Price: num  0.867 0.869 0.867 0.867 0.867 ...
```

Finally, the overall dataset is merged on a daily basis. Additionally, the difference between the GB electricity price and the EU electricity price is calculated (using log-transformations, i.e. GB/EU).

```
#Merging data
merged_data <- merge(EUR_GBP, Electricity_price_EUg, by = 'Date')
merged_data <- merge(Electricity_price_gb, merged_data, by = 'Date')
#From EU to £
merged_data$Daily_ave_EU <-
  merged_data$Daily_ave_EU * merged_data$Price
merged_data$Monthly.Date <-
  as.Date(as.yearmon(format(merged_data$Date, "%Y-%m")))
#Only positive values
merged_data <- merged_data[merged_data$Daily_ave_GB > 0 &
                             merged_data$Daily_ave_EU > 0, ]
#log
merged_data$Daily_ave_GB <- log(merged_data$Daily_ave_GB)
merged_data$Daily_ave_EU <- log(merged_data$Daily_ave_EU)
#Diff
merged_data$Price_diff <-
  (merged_data$Daily_ave_GB - merged_data$Daily_ave_EU)
#NA checking
merged_data <- na.omit(merged_data)
```

Electricity prices and Great Britain's net imports

```
#Data
# gco2/KW -> carbon intensity
# MW -> electricity imports/exports/production/consumption
# £/MW -> EU/UK price
# £/tco2 -> EU/UK ETS
```

Data from Electricity Map is utilized to conduct an empirical analysis on the relationship between GB-EU electricity prices and GB's net imports.

```
#Uploading electricity map data
GB_20 <-
  read.csv("Data/Electricity_map_GB/GB_20.csv")[, 2:82]
GB_21 <-
  read.csv("Data/Electricity_map_GB/GB_21_22.csv")[, 2:62]
GB_23 <-
  read.csv("Data/Electricity_map_GB/GB_23.csv")[, 2:62]
dataset_tot_GB <- bind_rows(GB_20, GB_21, GB_23)
dataset_tot_GB[is.na(dataset_tot_GB)] <- 0
#Net imports EU
dataset_tot_GB$Net_imports_EU <-
  dataset_tot_GB$power_net_import_NL_avg +
  dataset_tot_GB$power_net_import_IE_avg + dataset_tot_GB$power_net_import_FR_avg +
  dataset_tot_GB$power_net_import_BE_avg + dataset_tot_GB$power_net_import_DK.DK1_avg
dataset_tot_GB <- dataset_tot_GB[, -c(2:5, 17, 62:81)]
dataset_tot_GB$datetime <- as.Date(date(dataset_tot_GB$datetime))
#Co2 (gCo2) total
dataset_tot_GB$CO2_production_avg <-
  dataset_tot_GB$carbon_intensity_production_avg *
  dataset_tot_GB$total_production_avg * 1000 #Form MW to gco2 by gco2/KW
#Co2 (gCo2) direct
dataset_tot_GB$CO2_production_direct_avg <-
  dataset_tot_GB$carbon_intensity_direct_avg *
  dataset_tot_GB$total_production_avg * 1000
```

All variables on the platform are expressed on an hourly basis. Therefore, a daily aggregation is performed for a selected subset of variables.

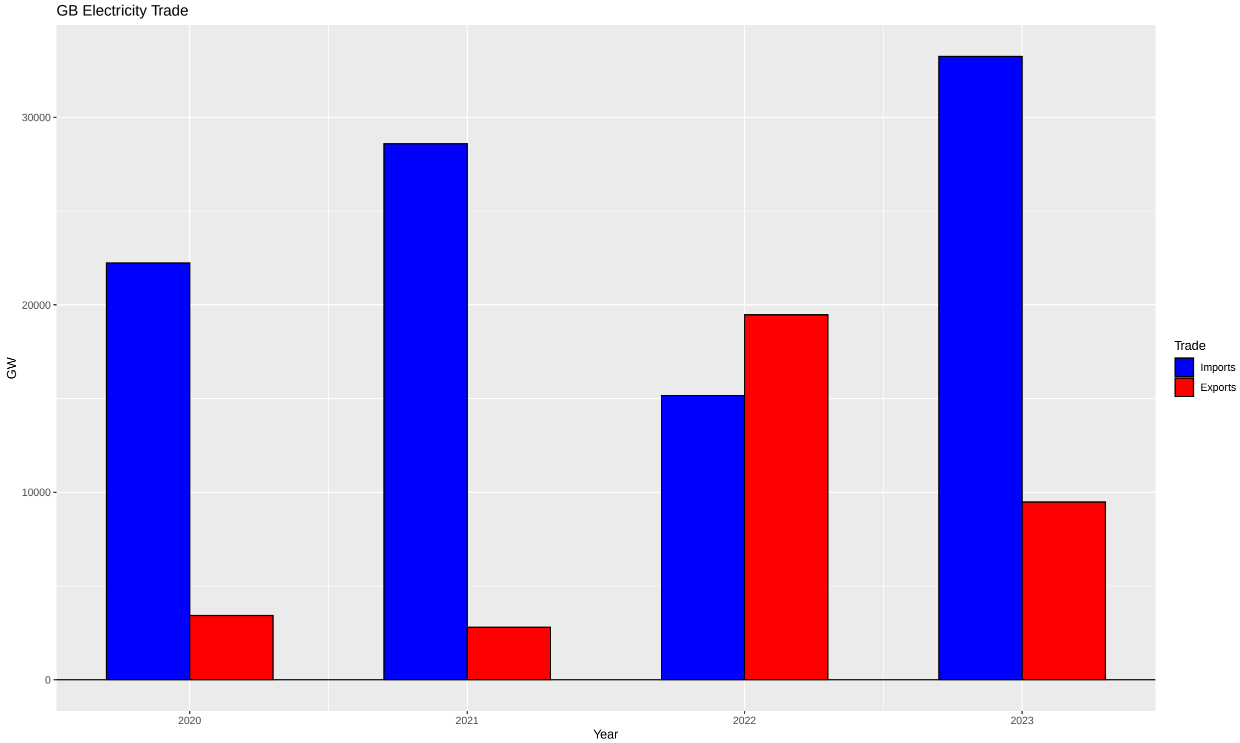
```
#Daily aggregation
dataset_GB_daily <- dataset_tot_GB[, c(
  "datetime",
  "CO2_production_avg",
  "CO2_production_direct_avg",
  "total_production_avg",
  "total_consumption_avg",
  "total_import_avg",
  "total_export_avg",
  "Net_imports_EU"
)]
dataset_GB_daily <- dataset_GB_daily %>% group_by(datetime) %>%
  dplyr::summarize(across(everything(), sum, na.rm = TRUE))
colnames(dataset_GB_daily)[1] <- 'Date'
```

Subsequently, the dynamics of GB's trades with the world are plotted on an annual basis.

```r
dataset_GB_daily$Year <- year(dataset_GB_daily$Date)
plot_data <-
  dataset_GB_daily[, c("Year", "total_import_avg", "total_export_avg")] %>%
  group_by(Year) %>%
  dplyr::summarize(across(everything(), sum, na.rm = TRUE))
plot_data$total_import_avg <-
  plot_data$total_import_avg / 1000 #From MW to GW
plot_data$total_export_avg <-
  plot_data$total_export_avg / 1000 #From MW to GW
colnames(plot_data) <- c('Year', 'Imports', 'Exports')
meltdata <- melt(plot_data, id = c('Year'))
```

```
## Warning in melt(plot_data, id = c("Year")): The melt generic in data.table has
## been passed a tbl_df and will attempt to redirect to the relevant reshape2
## method; please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(plot_data). In the next version, this warning will become an
## error.
```

```r
colnames(meltdata) <- c('Year', 'Trade', 'Value')
meltdata %>%
  ggplot(aes(
    x = Year,
    y = Value,
    group = Trade,
    fill = Trade
  )) +
  geom_col(
    position = position_dodge(width = 0.6),
    colour = "black",
    width = 0.6
  ) +
  geom_hline(aes(yintercept = 0)) + ylab('GW') + ggtitle('GB Electricity Trade') +
  scale_colour_manual(values = c('blue', 'red'),
                      aesthetics = c("colour", "fill"))
```



The same is done considering the dynamics of GB's trades with the EU only (data from GOV-UK).

```
#https://www.gov.uk/government/collections/energy-trends#2024-data
#Raw data quarterly
Electricity_imp_exp <-
  read_excel("Data/Electricity_imp_exp.xlsx",
             sheet = "Quarterly imports and exports",
             skip = 5)
#GB imports/exports/Net imports to EU.
Electricity_imp_exp <-
  Electricity_imp_exp[, c(1, 3, 4, 9, 10, 21, 22)]
colnames(Electricity_imp_exp) <- c(
  'Year',
  'Imports_GB',
  'Exports_GB',
  'Imports_NIR',
  'Exports_NIR',
  'Imports_NR',
  'Exports_NR'
)
#Filtering
Electricity_imp_exp$Imports_NR <-
  as.numeric(Electricity_imp_exp$Imports_NR)
```

```
## Warning: NA introdotti per coercizione
```

```
Electricity_imp_exp$Exports_NR <-
  as.numeric(Electricity_imp_exp$Exports_NR)
```

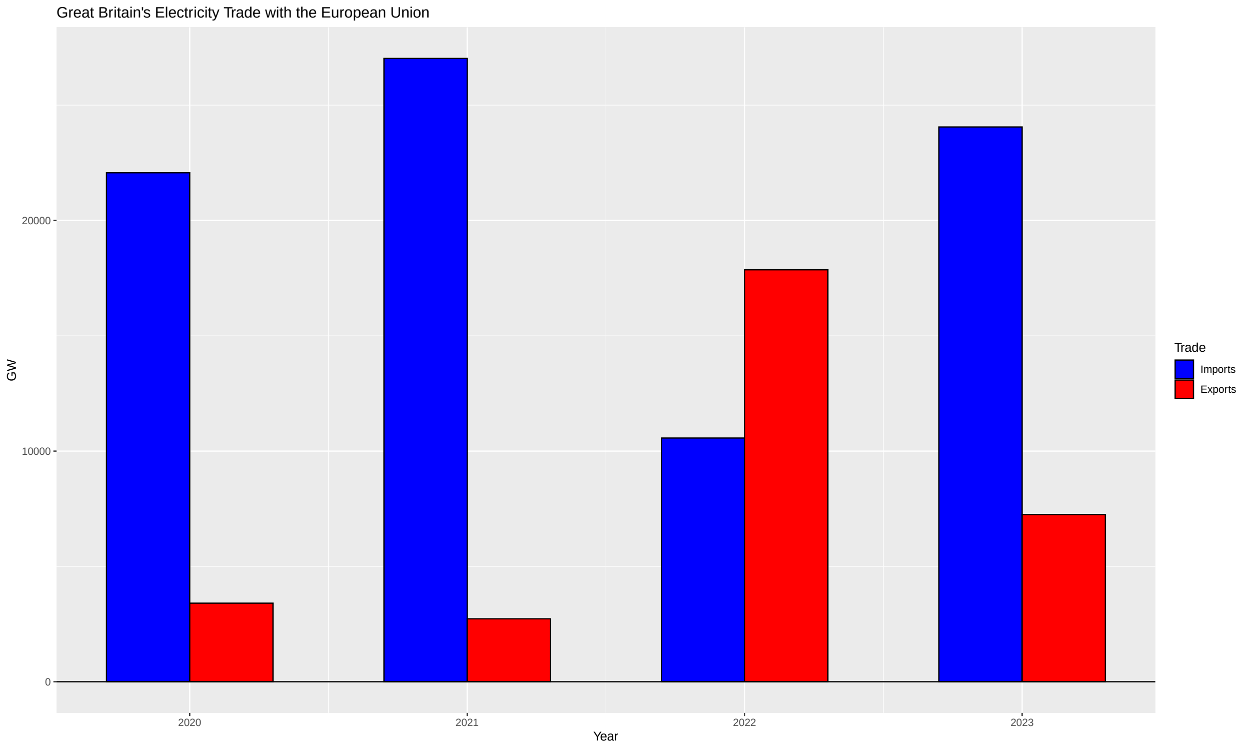
```
## Warning: NA introdotti per coercizione
```

```r
Electricity_imp_exp[is.na(Electricity_imp_exp)] <- 0
Electricity_imp_exp$Imports_GB <-
  Electricity_imp_exp$Imports_GB - Electricity_imp_exp$Imports_NIR -
  Electricity_imp_exp$Imports_NR
Electricity_imp_exp$Exports_GB <-
  Electricity_imp_exp$Exports_GB - Electricity_imp_exp$Exports_NIR -
  Electricity_imp_exp$Exports_NR
#Rearranging
Electricity_imp_exp <- Electricity_imp_exp[, c(1:3)]
colnames(Electricity_imp_exp) <-
  c('Year', 'Imports', 'Exports')
Electricity_imp_exp <-
  Electricity_imp_exp[Electricity_imp_exp$Year > 2019, ]
Electricity_imp_exp$Imports <-
  as.numeric(Electricity_imp_exp$Imports)
Electricity_imp_exp$Exports <-
  as.numeric(Electricity_imp_exp$Exports)
Electricity_imp_exp[is.na(Electricity_imp_exp)] <- 0
```

```r
Electricity_imp_exp <- Electricity_imp_exp %>%
  group_by(Year) %>%
  summarize(across(everything(), sum, na.rm = TRUE))
meltdata <- melt(Electricity_imp_exp, id = c('Year'))
```

```
## Warning in melt(Electricity_imp_exp, id = c("Year")): The melt generic in
## data.table has been passed a tbl_df and will attempt to redirect to the
## relevant reshape2 method; please note that reshape2 is deprecated, and this
## redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(Electricity_imp_exp). In the next version, this
## warning will become an error.
```

```r
colnames(meltdata) <- c('Year', 'Trade', 'Value')
meltdata %>%
  ggplot(aes(
    x = Year,
    y = Value,
    group = Trade,
    fill = Trade
  )) +
  geom_col(
    position = position_dodge(width = 0.6),
    colour = "black",
    width = 0.6
  ) +
  geom_hline(aes(yintercept = 0)) + ylab('GW') + ggtitle('Great Britain\'s Electricity Trade with the Eu
  scale_colour_manual(values = c('blue', 'red'),
                      aesthetics = c("colour", "fill"))
```

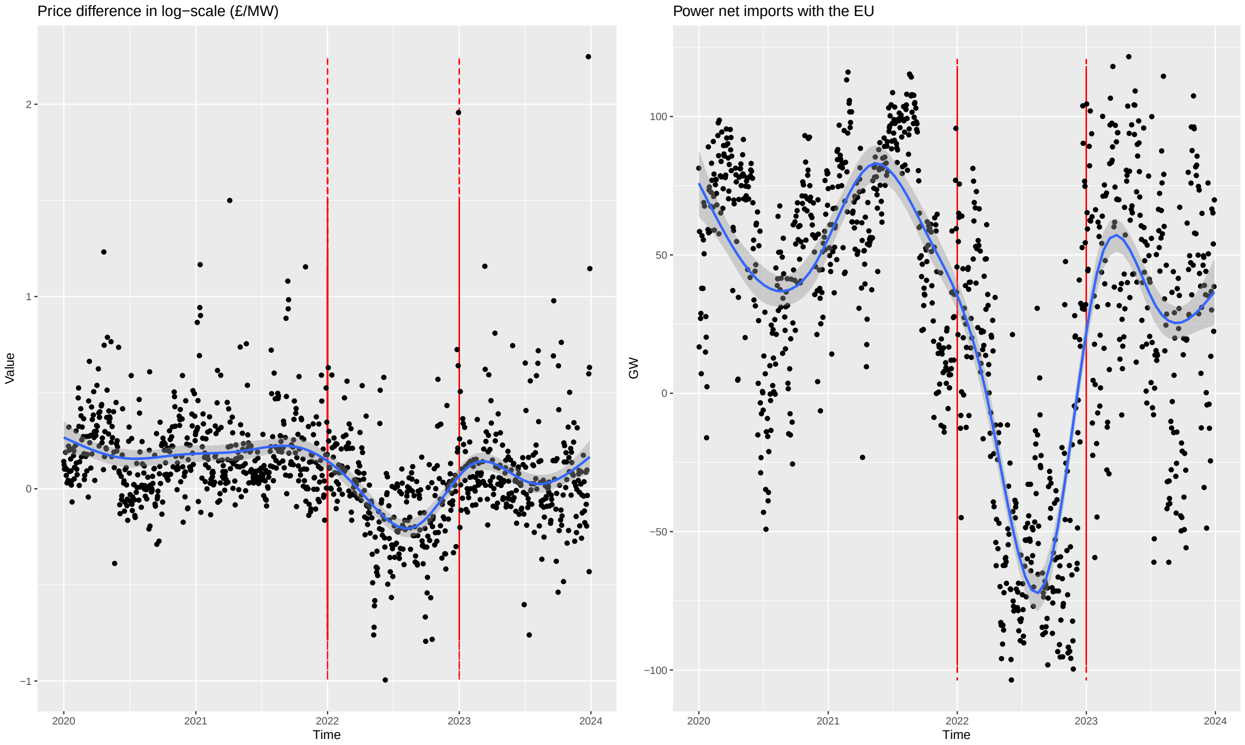


The following chunk provides a descriptive summary of the daily values, focusing on the differences in prices between Great Britain and the European Union, as well as net imports.

```
#Merging -> observations loss is due to finance data not reported for some days
dataset_GB_daily <-
  merge(merged_data[, c(1, 2, 4, 6)], dataset_GB_daily, by = 'Date')
#Total exports
dataset_GB_daily$Net_imports_tot <-
  dataset_GB_daily$total_import_avg -
  dataset_GB_daily$total_export_avg
#Carbon intensity production (all)
dataset_GB_daily$carbon_intensity_production_avg <-
  dataset_GB_daily$CO2_production_avg /
  (dataset_GB_daily$total_production_avg * 1000) #gco2/KW
#Carbon intensity production (direct)
dataset_GB_daily$carbon_intensity_production_direct_avg <-
  dataset_GB_daily$CO2_production_direct_avg /
  (dataset_GB_daily$total_production_avg * 1000) #gco2/KW
#Plots
data_plot <- dataset_GB_daily[dataset_GB_daily$Date < '2023-01-01' &
                                dataset_GB_daily$Date > '2021-12-31', ]
data_plot <- dataset_tot_GB[dataset_tot_GB$datetime < '2023-01-01' &
                              dataset_tot_GB$datetime > '2021-12-31', ]
A1 <- ggplot(dataset_GB_daily, aes(x = Date, y = Price_diff)) +
  geom_line(aes(x = as.Date('2022-1-1')), linetype = "dashed", col = 'red') +
  geom_line(aes(x = as.Date('2023-1-1')), linetype = "dashed", col = 'red') +
  geom_point() +
  geom_smooth(method = "gam") + xlab('Time') + ylab('Value') + ggtitle('Price difference in log-scale (

A2 <-
```

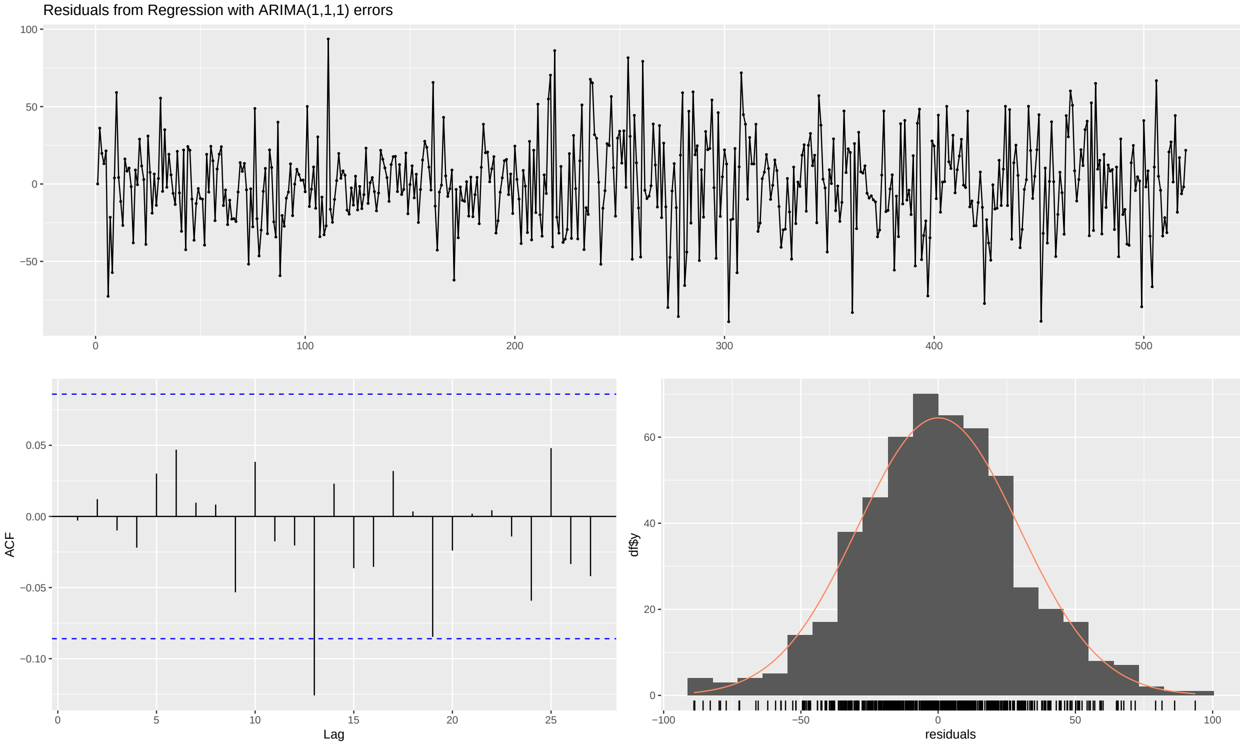
```
  ggplot(dataset_GB_daily, aes(x = Date, y = Net_imports_EU / 1000)) + # from MW to GW
  geom_line(aes(x = as.Date('2022-1-1')), linetype = "dashed", col = 'red') +
  geom_line(aes(x = as.Date('2023-1-1')), linetype = "dashed", col = 'red') +
  geom_point() +
  geom_smooth(method = "gam") + xlab('Time') + ylab('GW') + ggtitle('Power net imports with the EU')

plot_grid(A1, A2, nrow = 1, scale = 1)
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



Finally, a restricted dataset containing all observations from 2022 is used to estimate an ARIMAX model, incorporating GB-EU price differences as an exogenous variable.

```
#Restricted dataset for simulations
dataset_GB_daily <-
  dataset_GB_daily[dataset_GB_daily$Date > '2022-01-01', ]
#GW
dataset_GB_daily$Net_imports_EU <-
  dataset_GB_daily$Net_imports_EU / 1000 #from MW to GW
#Lag-values for overconfidence in predictions
dataset_GB_daily$Price_difflag <- lag(dataset_GB_daily$Price_diff)
#Arimax
Arimax <- auto.arima(
  dataset_GB_daily$Net_imports_EU,
  xreg = matrix(c(dataset_GB_daily$Price_diff), ncol = 1),
  max.p = 10,
  max.q = 10,
  max.d = 10,
  lambda = NULL
)
#Residuals
checkresiduals(Arimax)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,1,1) errors
## Q* = 4.4034, df = 8, p-value = 0.819
##
## Model df: 2.   Total lags used: 10
```

```
bptest(residuals(Arimax) ~ fitted(Arimax))
```

```
##
##  studentized Breusch-Pagan test
##
## data:  residuals(Arimax) ~ fitted(Arimax)
## BP = 2.3936, df = 1, p-value = 0.1218
```

```
#Coefficients
coeftest(Arimax)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.379328   0.070773   5.3598 8.332e-08 ***
## ma1  -0.830715   0.044817 -18.5357 < 2.2e-16 ***
## xreg 17.035590   4.819767   3.5345 0.0004085 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residuals of the model appear to be homoscedastic, normally distributed, and free from autocorrelation.

Scenarios

Finally, a simulation study is conducted to evaluate the possible impact of the EU CBAM. In order to perform the exercise, the CBAM obligation is calculated based on the embedded carbon emissions of GB direct electricity production. Then, the CBAM adjustment is determined by the difference between the price of carbon already paid in the UK and the price of carbon in Europe.

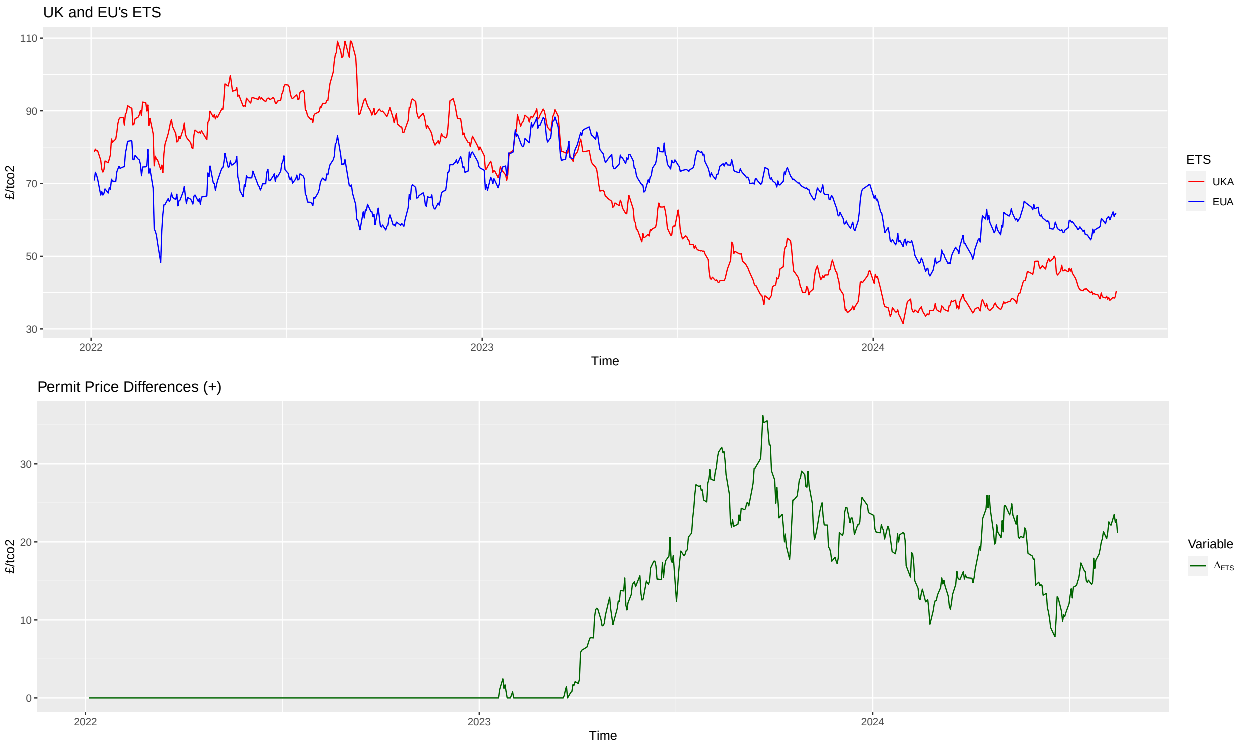
```
#EUA
EUA_dec <- read_excel("Data/EUA_dec_GBP.xlsx")
EUA_dec$Date <- as.Date(date(EUA_dec$Date))
EUA_dec <- na.omit(EUA_dec)
#UKA
UKA_dec <- read_excel("Data/UKA_dec.xlsx")
UKA_dec$Date <- as.Date(date(UKA_dec$Date))
UKA_dec <- na.omit(UKA_dec)
#Merge
merged_ETS <- merge(UKA_dec, EUA_dec, by = 'Date')
colnames(merged_ETS) <- c('Date', 'UKA', 'EUA')
merged_ETS$UKA <- as.numeric(merged_ETS$UKA)
merged_ETS$EUA <- as.numeric(merged_ETS$EUA)
```

The exercise is conducted on the solar year 2023.

```
Plot <- melt(merged_ETS, id.vars = c('Date'))
```

```
## Warning in melt(merged_ETS, id.vars = c("Date")): The melt generic in
## data.table has been passed a data.frame and will attempt to redirect to the
## relevant reshape2 method; please note that reshape2 is deprecated, and this
## redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(merged_ETS). In the next version, this warning
## will become an error.
```

```
colnames(Plot)[2] <- c('ETS')
B1 <- ggplot(data = Plot, aes(
  x = Date,
  y = value,
  group = ETS,
  colour = ETS
)) + geom_line() + labs(y = "£/tco2", x = "Time") + ggtitle("UK and EU's ETS") +
  scale_color_manual(
    name = 'ETS',
    labels = c("UKA", "EUA"),
    values = c("red", "blue")
  )

merged_ETS$Diff <- (merged_ETS$EUA - merged_ETS$UKA)
merged_ETS[merged_ETS$Diff <= 0, 'Diff'] <- 0

B2 <- ggplot(data = merged_ETS, aes(
  x = Date,
  y = Diff,
  colour = 'darkgreen'
)) + geom_line() + labs(y = "£/tco2", x = "Time") + ggtitle("Permit Price Differences (+)") +
  scale_color_manual(
    name = 'Variable',
    labels = c(expression(Delta[ETS])),
    values = c("darkgreen")
  )

plot_grid(B1, B2, nrow = 2, scale = 1)
```



```
merged_ETS <- merged_ETS[merged_ETS$Date > '2022-12-31' &
                           merged_ETS$Date < '2024-01-01',]
```

Due to the temporal characteristics of the ARIMAX model, both the real scenario and the projected CBAM scenario are filtered through the baseline model, which assumes no differences in electricity prices.

```
dataset_scenario <-
  dataset_GB_daily[dataset_GB_daily$Date > '2022-12-31' &
                     dataset_GB_daily$Date < '2024-01-01', ]
#From log to baseline
dataset_scenario$Daily_ave_GB <- exp(dataset_scenario$Daily_ave_GB)
dataset_scenario$Daily_ave_EU <- exp(dataset_scenario$Daily_ave_EU)
dataset_scenario <- merge(dataset_scenario, merged_ETS, by = 'Date')
#CBAM
pred <- forecast(Arimax, xreg = c(rep(0, 251)), h = 251)
xreg_CBAM <- log((dataset_scenario$Daily_ave_GB + (
  (
    dataset_scenario$carbon_intensity_production_direct_avg / 1000
  ) * dataset_scenario$Diff
)) / dataset_scenario$Daily_ave_EU) #/1000 to convert gco2/KW into tco2/MW
```

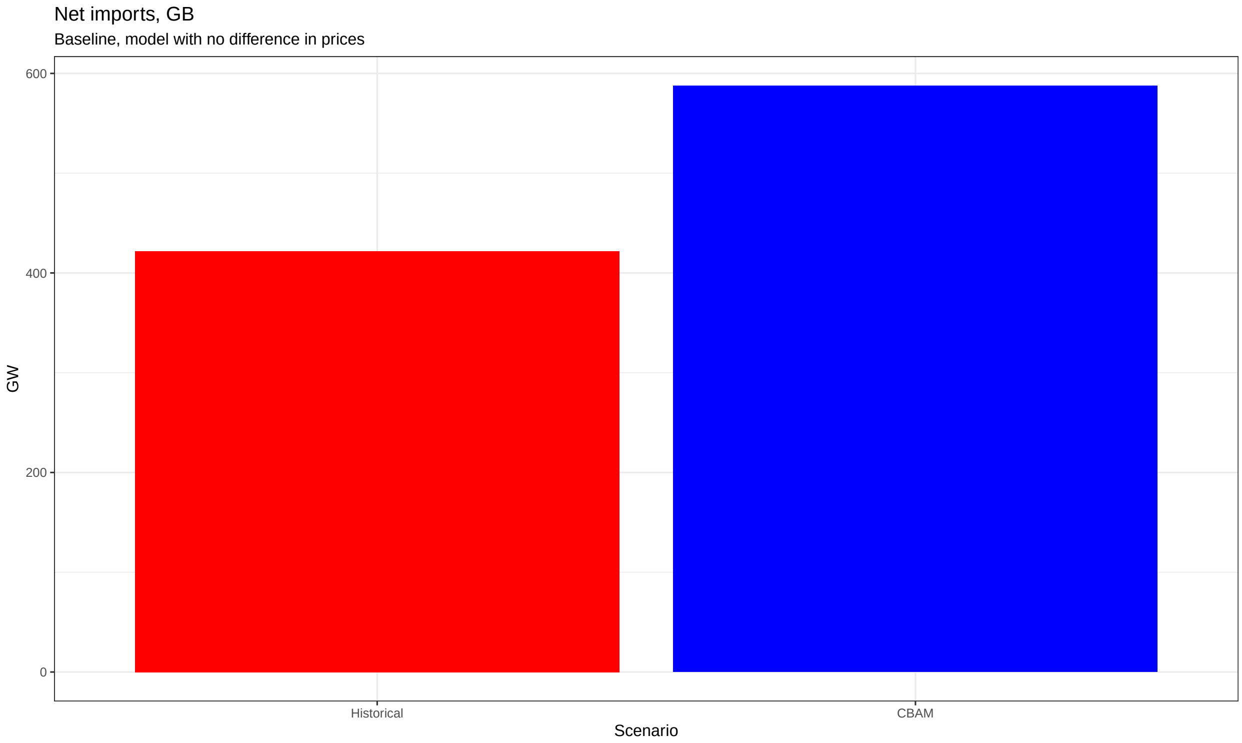
```
pred1 <- forecast(Arimax, xreg = c(xreg_CBAM), h = 251)
#Real
xreg_real <-
  log(dataset_scenario$Daily_ave_GB / dataset_scenario$Daily_ave_EU)
pred2 <- forecast(Arimax, xreg = c(xreg_real), h = 251)
```

The results of the exercise indicate that the introduction of the EU Carbon Border Adjustment Mechanism (CBAM) could lead to a reduction in net electricity exports to the EU by approximately 0.45 GW per day.

```
#Results
meltData <- melt(data.frame(
  CBAM = mean(pred1$mean - pred$mean),
  Real = mean(pred2$mean - pred$mean)
))
```

```
## Warning in melt(data.frame(CBAM = mean(pred1$mean - pred$mean), Real =
## mean(pred2$mean - : The melt generic in data.table has been passed a data.frame
## and will attempt to redirect to the relevant reshape2 method; please note that
## reshape2 is deprecated, and this redirection is now deprecated as well. To
## continue using melt methods from reshape2 while both libraries are attached,
## e.g. melt.list, you can prepend the namespace like
## reshape2::melt(data.frame(CBAM = mean(pred1$mean - pred$mean), Real =
## mean(pred2$mean - pred$mean))). In the next version, this warning will become
## an error.

## No id variables; using all as measure variables
```

```
#Yearly
meltData$value <- meltData$value * 365
ggplot(meltData, aes(
  x = reorder(variable, value),
  y = value,
  fill = variable,
  color
)) +
  geom_bar(position = "dodge",
           stat = "summary",
           fun = "identity") +
  theme_bw(base_size = 14) +
  scale_fill_manual(values = c('blue', 'red')) +
  scale_alpha_manual(values = c(1, 0.1)) +
  theme(legend.position = "none") +
  xlab("Scenario") + ggtitle('Net imports, GB', 'Baseline, model with no difference in prices') +
  ylab('GW') + scale_x_discrete(labels = c('Historical', 'CBAM'))
```

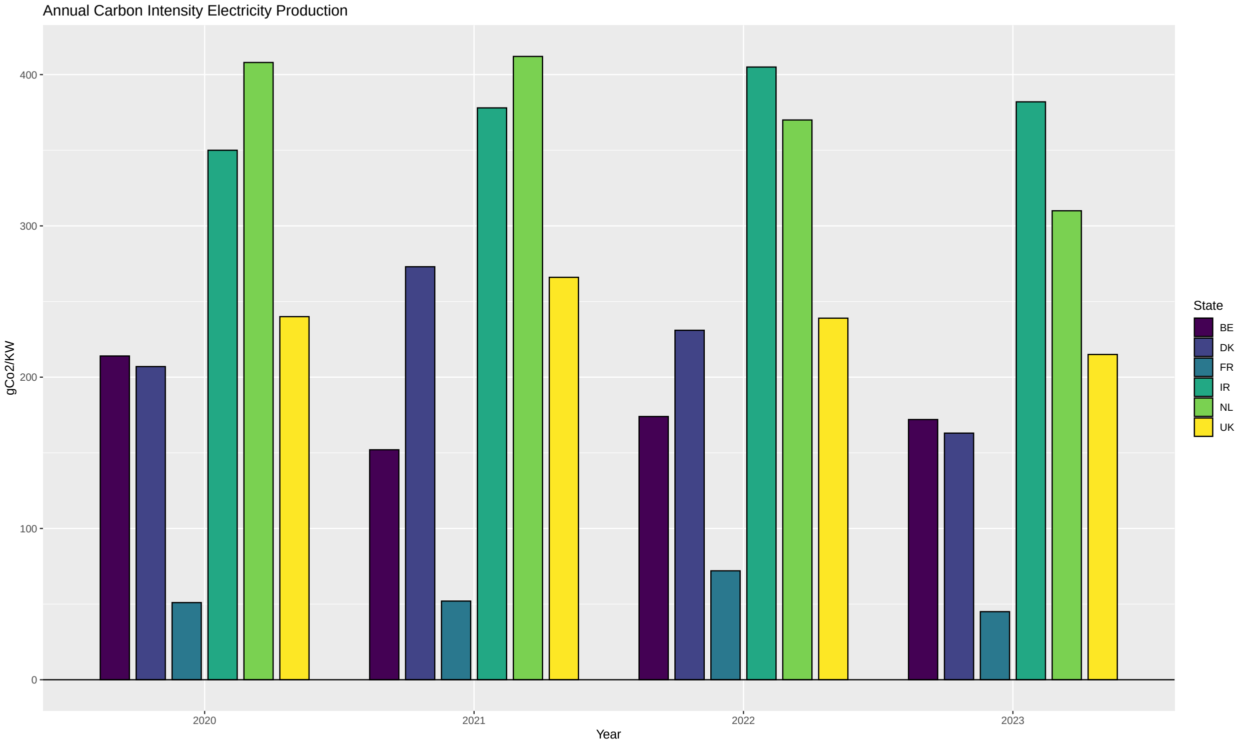


```
#-166 GW yearly
#-0.45 GW daily
#Mean price change
mean(
  dataset_scenario$carbon_intensity_production_direct_avg / 1000 * dataset_scenario$Diff
)
```

```
## [1] 2.255835
```

Finally, the overall carbon intensity production for both GB and EU are reported in the following bar plot.

```
#Carbon intensity UK/EU
data <- data.frame(
  Year = rep(c("2020", '2021', '2022', '2023'), 6),
  State = c(
    rep('DK', 4),
    rep('FR', 4),
    rep('IR', 4),
    rep('UK', 4),
    rep('NL', 4),
    rep('BE', 4)
  ),
  CI = c(
    207,
    273,
    231,
    163,
    51,
    52,
    72,
    45,
    350,
    378,
    405,
    382,
    240,
    266,
    239,
    215,
    408,
    412,
    370,
    310,
    214,
    152,
    174,
    172
  )
)

data %>%
  ggplot(aes(
    x = Year,
    y = CI,
    group = State,
    fill = State
  )) +
  geom_col(
    position = position_dodge(width = 0.8),
    colour = "black",
    width = 0.65
  ) +
  geom_hline(aes(yintercept = 0)) + ylab('gCo2/KW') + ggtitle('Annual Carbon Intensity Electricity Produ
  scale_colour_manual(values =  viridis(6),
                      aesthetics = c("colour", "fill"))
```



```
knitr::knit_exit()
```