

BUS350 Semester project

Christine Westerboe and Kamilla Aaboen

1.0 Introduction

The United Kingdom's withdrawal from the European Union represents one of the most significant institutional and economic changes in recent decades. The Brexit referendum was held in June 2016, and the UK officially left the European Union (EU) on 31 January 2020, marking the end of nearly five decades of membership. EU has long been one of the UK's largest trading partners, but Brexit introduced new barriers to trade, altered market access conditions, and reshaped the country's external economic relationships.

This project aims to explore how the UK's trade with EU and non-EU partners has evolved since the Brexit referendum, with a particular focus on identifying structural shifts in trade intensity. The analysis is grounded in the CEPII Gravity database of international trade, which provides data for explaining bilateral trade as a function of economic size, distance, and institutional linkages. By examining trade developments between 2010 and 2020, we seek to explore whether Brexit has led to changes in trading patterns.

1.1 Research question

How has the United Kingdom's trading patterns with EU and non-EU partners changed since the Brexit referendum in 2016?

2.0 Data and methods

The data used for this project mainly draw on the CEPII Gravity Database, and contains a targeted selection of bilateral variables between the United Kingdom and its trading partners. When downloading the data, we also downloaded the Gravity documentation file which included information about variables in the data set. Among all available variables, we made a selection of variables that present the main economic, geographic and institutional factors that could reasonably be expected to have influenced trade patterns before and after Brexit.

2.1 AI disclaimer

When making use of AI in this project, we have made efforts to follow the course guidelines for responsible and limited use of AI. Our AI model of choice has been the version of ChatGPT that at any point has been freely available to consumers throughout the fall semester of 2025. To apply AI as effective tools, it has been necessary for us to have a solid understanding of basic R concepts. Since our group had different levels of experience with R, we spent time establishing shared knowledge about the essential components of the R workflow. This included how to run commands in the console, how R projects and working directories function, and how scripts, data sets, and rendering processes are connected. Building this foundation ensured that we could correctly interpret technical explanations from AI and verify whether suggested solutions were appropriate for our project setup.

The use of AI for this project had been primarily as a technical support tool. It helped identify and resolve issues that commonly occur in collaborative R work. Examples include file-path conflicts, missing working directories, and render failures when switching between two different computers. In several cases, R attempted to access folders located only on one team member's system, or tried to load packages that were not installed on the other member's machine. By providing AI with screenshots or detailed descriptions of error messages, we were able to isolate the cause of these problems efficiently. All fixes proposed by AI were reviewed and implemented manually to maintain full control over the workflow. We also used AI to improve the quality of our written text by checking grammar, suggesting clearer wording, and helping us express our explanations in a more precise and understandable way.

AI was also used to clarify basic functionality of the R language, such as how certain commands operate and how to structure an R project correctly. This was particularly useful during the early phase of the project, when we needed to establish routines for handling data, loading packages, and ensuring consistent file organisation across devices and operating systems. As our competence increased, the use of AI shifted toward more targeted queries. These included questions about improving the efficiency of existing code, adjusting minor details in plots, or identifying why a specific line of code did not run as intended. In each case, AI served as a technical aid rather than a source of analytical input. Throughout this report, concrete uses of AI will be disclaimed where appropriate.

2.2 Packages

```
# For working directory guides  
library(here)  
  
# For data manipulation  
library(dplyr)  
library(tidyverse)
```

```

library(countrycode)      # converting country codes to country names
library(modelr)

# For making visualisations
library(ggplot2)
library(patchwork)        # helper package for side-by-side plots
library(scales)           # Scale functions for visualisations

# For report generation
library(knitr)

# Global settings
options(scipen = 999) # makes R write large numbers without scientific notations

```

2.3 Creating the datasets

For this analysis we have mainly depended on data from the CEPII Gravity database, which again bases itself on data collected from a number of different open sources. Further information about data and variables can be found in the Gravity Documentation file. We also gathered data about GDP per capita from the World Development Indicators (WDI) by the World Bank. Although the CEPII Gravity data bank already hold data about GDP and population for all countries, these data were imported to demonstrate our ability to read and combine data from several sources. For instance, the gravity data was used to make a filtering join in the WDI data to remove already aggregated observations or other non-state entities. We named our main data set `gravity`, and kept the name `WDI` for the data gathered from the World Bank.

Several AI requests were made in the process of wrangling the WDI data. The data was delivered in a wide format with each year as an individual column, with the 70th and last column holding only empty values. The first AI request questioned the origin of the empty 70th column. The AI suggested that the empty column was the result of the raw data file including a comma at the end of each row. This explanation was possible to validate by inspecting the raw data files, and the empty column could be safely removed.

The second and third AI requests concerning the wrangling of the WDI data regarded the use of `pivot_longer()` and `pivot_wider()`, both of which were used to achieve the desired data format. As the data was imported in a wide format, an AI request was made to suggest the most appropriate way to select all year columns for the `pivot_longer()` function. The syntax `matches("^(19|20)")` was suggested, and by running the code and inspecting the changes made it was possible to validate the effect was as wanted. The third inquiry regarded an more detailed explanation of the `pivot_wider()` documentation.

```

# Read the gravity data
gravity <- readRDS(here("data", "Gravity_V202211.rds")) %>% # Gravity source file location
  select(
    year, iso3_o, iso3_d, eu_o, eu_d,
    country_exists_o, country_exists_d,
    distcap, pop_o, pop_d, gdp_o, gdp_d,
    tradeflow_comtrade_o, tradeflow_comtrade_d
  ) %>%
  filter(
    iso3_o == "GBR" | iso3_d == "GBR",           # only include observations with the UK
    country_exists_o == 1 & country_exists_d == 1, # remove observations with non-existing o
    year %in% 2010:2020) %>%                   # focus on 2010-2020
  mutate(partner_eu = if_else(iso3_o == "GBR", eu_d, eu_o)) %>% # EU-status for partnerland
  mutate(country_d = countrycode(iso3_d, "iso3c", "country.name", nomatch = NULL),
         country_o = countrycode(iso3_o, "iso3c", "country.name", nomatch = NULL)) %>% #ANT
  mutate(tradeflow_comtrade_d = tradeflow_comtrade_d * 1000, # tradeflow is reported in thous
         tradeflow_comtrade_o = tradeflow_comtrade_o * 1000)

# Read the WDI data
WDI <- read_csv("WDI_NY.GDP.PCAP.KD.csv", col_names = TRUE, skip = 3) %>%
  select(!starts_with(...70),
         -"Indicator Code") %>%
  pivot_longer(
    cols = matches("(19|20)"),
    names_to = "year",
    values_to = "value"
  ) %>%
  pivot_wider(
    names_from = "Indicator Name",
    values_from = "value"
  ) %>%
  rename("GDPpc" = "GDP per capita (constant 2015 US$)") %>%
  filter(year %in% 2010:2024) %>%
  mutate(year = as.numeric(year))

```

New names:

Rows: 266 Columns: 70
-- Column specification
----- Delimiter: "," chr
(4): Country Name, Country Code, Indicator Name, Indicator Code dbl (65): 1960,
1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ... lgl (1): ...70
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.  
* `` -> `...70`
```

```
# Using a filtering join to remove non-state observations from the WDI data (e.g. "Arab World"  
WDI <- WDI %>%  
  semi_join(gravity, join_by(`Country Code` == iso3_d))  
  
# Inspecting the removed observations  
  
# WDI_removed <- WDI %>%  
#   anti_join(gravity, join_by(`Country Code` == iso3_d))  
#  
# check <- WDI_removed %>% count(`Country Name`)
```

Based on our main data set, we also created smaller subsets of data to make subsequent analyses more intuitive. The subsets of data were divided into exports and imports, as well as some filtered by year.

```
# All UK exports 2010-2020  
uk_exports <- gravity %>%  
  select(!starts_with("country_exists")) %>%  
  filter(iso3_o == "GBR")  
  
# All UK imports 2010-2020  
uk_imports <- gravity %>%  
  select(!starts_with("country_exists")) %>%  
  filter(iso3_d == "GBR")  
  
# Exports subsets by year  
uk_exports_2010 <- uk_exports %>% filter(year == 2010)  
uk_exports_2020 <- uk_exports %>% filter(year == 2020)  
  
# Import subsets by year  
uk_imports_2010 <- uk_imports %>% filter(year == 2010)  
uk_imports_2020 <- uk_imports %>% filter(year == 2020)
```

2.4 Variables

Variables in the gravity data set include yearly observation for pairs of countries with a trade relationship, with variables describing both the exporting and the importing state. For any

variable name, the suffixes `_o` and `_d` mean that the variable describe either the origin (exporter) or destination (importer) state.

The trade flow variables `tradeflow_comtrade_o` and `tradeflow_comtrade_d` originally captured the value of exports and imports in thousands current US\$. During the data wrangling phase, the values for these variables were multiplied by 1000 to accurately depict monetary value. These variables form the foundation for analysing how trade volumes have evolved over time. To control for differences in market size and economic capacity, the data set includes GDP (`gdp_o`, `gdp_d`) and population (`pop_o`, `pop_d`) for both the UK and its partners, providing context for the scale and potential intensity of bilateral trade. Geographic distance `distcap` serves as a proxy for trade costs, reflecting how physical separation can constrain trade through higher transport and transaction costs. The EU membership indicators `eu_o` and `eu_d` are central to the analysis, as they allow a distinction between EU-members and non-EU trading partners and thereby facilitate the identification of changes linked to the UK's withdrawal from the EU. Finally, `country_exists_o` and `country_exists_d` ensure that only valid and contemporaneous country pairs are included each year. In total, these core variables were selected from the broader CEPPII database to provide a balanced representation of economic scale, spatial frictions, and institutional context, enabling a systematic analysis of how the UK's trade flows may have been affected by Brexit.

3.0 Data analysis

To answer the research question this section will firstly explore the data descriptively. A descriptive analysis of the data is useful because it provides information on the actual content and composition of the data. This is done by producing and inspecting descriptive statistics and creating visualizations in the form of figures and tables. This section will look at descriptive statistics for the UK, followed by a look into selected UK trading partners for comparison, and lastly a broad comparison of UK trade relations with EU-members and non-EU states.

After the descriptive analyses, the next section will present several attempts at modelling the data to provide a deeper understanding. While descriptive visualizations provide information of the actual observations, data modelling is useful in the way that it creates representations and predictions of the data, sacrificing some accuracy to better understand patterns and trends.

3.1 Descriptive data analysis

3.1.1 The United Kingdom

```
# Get UK GDP per capita (constant 2015 USD)
uk_gdp <- WDI %>%
  filter(`Country Code` == "GBR")
```

```

# Plot
ggplot(uk_gdp, aes(x = year, y = GDPpc )) +
  geom_line(color = "#003399", linewidth = 1.1) +
  geom_vline(xintercept = 2016, linetype = "dashed", color = "red") +
  geom_vline(xintercept = 2021, linetype = "dashed", color = "darkred") +
  scale_x_continuous(n.breaks = 8) +
  scale_y_continuous(limits = c(35000, 50000), labels = scales::label_dollar(big.mark = " "))
  labs(
    title = "UK GDP per capita (constant 2015 US$)",
    x = "Year",
    y = NULL
  ) +
  theme_minimal(base_size = 13)

```

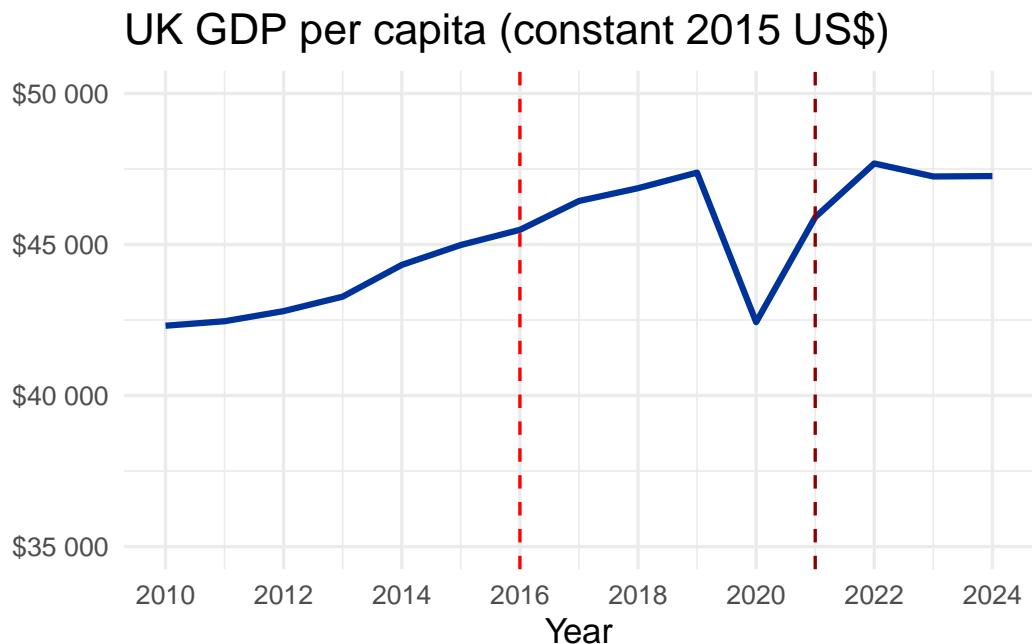


Figure 1: UK GDP per capita (constant 2015 US\$), with 2016 referendum and 2021 Trade and Cooperation Agreement marked by dashed lines.

Starting of the descriptive data analysis, this figure shows the evolution of UK real GDP per capita in constant 2015 US\$ from 2010 to 2024. The dashed vertical lines mark two key Brexit-related events: the 2016 referendum and the 2021 Trade and Cooperation Agreement (TCA). When the United Kingdom formally left the European Union in 2020, it did so without a

replacement agreement for cooperation with the Union. The TCA was signed on 30 December 2020, entering into force in the early months of 2021, and established a new framework for trade between the two economic powers (European Commission, 2021).

GDP per capita is a broad and useful economic measurement that can provide general information on how an economy is developing. The plot reveals that the UK did not experience any immediate fall in GDP per capita in the first years after the Brexit referendum. It does however show a significant drop in 2020, which coincidentally was the year the UK formally left the union. This could suggest an economic shock from leaving the EU before a new trade agreement was reached, as the TCA was only implemented a year after the UK had left the union. However, the downturn in GDP per capita in 2020 may very well be explained by other developments, such as the covid-19 pandemic affecting the global economy. On its own, the UK series gives a sense of the country's long-run income trajectory, but can not provide information on causality. It does not provide information on whether Brexit had an effect on the British economy, because one cannot see how the UK would have evolved in the absence of these shocks. For that, a comparison group would be needed.

3.1.2 Selected partners for comparison

Another way to evaluate trends in the UK economy is by comparing its developments with that of other states during the same period. For this analysis, countries were chosen because they are among the UK's trade partners, and they are all included in standard gravity data sets such as CEPII and UN Comtrade. They represent both nearby European economies and major global markets. Most have long-standing trade or historical ties with the UK, making them suitable for comparative analysis. Their inclusion ensures that it is possible to analyse bilateral trade composition and flows within a consistent gravity-model framework.

The next figure therefore plots the cumulative change in real GDP per capita since 2010 for a set of advanced economies: Belgium, Canada, France, Germany, Italy, the Netherlands, Spain, Switzerland and the UK. Using 2010 as a common baseline highlights how the UK's growth path compares with peers over the same period. The dashed lines indicate the 2016 referendum and the 2021 TCA.

```
# Get GDP per capita (constant 2015 USD) for selected countries
dat <- WDI %>%
  filter(`Country Code` %in% c("GBR", "DEU", "FRA", "ITA", "ESP", "NLD", "BEL", "CHE", "CAN"))
  rename(country = `Country Code`)

# Calculate cumulative percent change since 2010
dat_growth <- dat %>%
  group_by(country) %>%
  mutate(
    base_2010 = GDPpc[year == 2010],
```

```

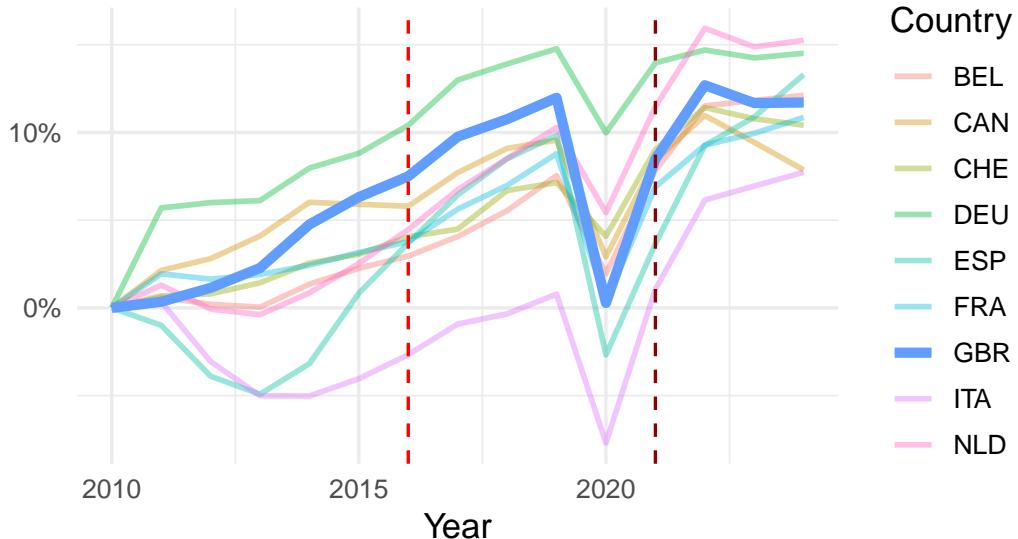
  pct_change = ((GDPpc / base_2010) - 1) * 100
) %>%
ungroup()

# Plot with bold UK line
ggplot(dat_growth, aes(x = year, y = pct_change, group = country)) +
  geom_line(aes(color = country), alpha = 0.4, size = 1) +
  geom_line(
    data = filter(dat_growth, country == "GBR"),
    aes(x = year, y = pct_change, color = country),
    linewidth = 1.8
) +
  geom_vline(xintercept = 2016, linetype = "dashed", color = "red") +
  geom_vline(xintercept = 2021, linetype = "dashed", color = "darkred") +
  labs(
    title = "Cumulative change in real GDP per capita",
    x = "Year",
    y = NULL,
    color = "Country",
    caption = "Source: World Bank, World Development Indicators"
) +
  scale_y_continuous(labels = percent_format(scale = 1)) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold"),
    plot.caption = element_text(hjust = 0.5, size = 10)
)

```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
 i Please use `linewidth` instead.

Cumulative change in real GDP per capita



Source: World Bank, World Development Indicators

Figure 2: Cumulative change in real GDP per capita since 2010. Dashed lines mark the 2016 Brexit referendum and 2021 Trade and Cooperation Agreement.

To make the UK line appear bolder in the figure, the code first plots all countries using `geom_line()` with lower opacity, `alpha (0.4)`, and a moderate size. Then, a second `geom_line()` layer was added that specifically filters the data for the United Kingdom (`data = filter(dat_growth, country == "GBR")`), and plots it with a thicker line for a contrasting effect.

From this comparative view, it becomes easier to evaluate relative performance. We can see whether the UK's cumulative growth accelerates or decelerates around the Brexit milestones compared with countries that were not directly exposed to Brexit-related trade shocks. At the time of the 2016 referendum, the UK was second only to Germany in GDP per capita developments since 2010. By the time the UK formally left the union in 2020, it was among the bottom countries among the selection. Their relative position was however increased in the year of the TCA. At the very end of the observed period, the UK held a middle position among the selected countries. As all the countries in the selection experienced a sharp dip around 2020, not just EU-members or European states, it is reasonable to assume that the same decline for the UK was mainly grounded in the covid-19 pandemic, and not primarily a result of Brexit. However, the relative position of the UK also decreased around the time of Brexit. These findings could correspond with explanations of Brexit slowing down the British economy compared to other states.

3.1.3 EU and non-EU trading partners

This section will explore how UK trading patterns differs from EU-members states to non-EU partners. First up are some simple analyses. The next tables present the three highest observations of UK exports and imports respectively during the time span 2010-2020.

```
# Highest three export value observation
largest_export <- uk_exports %>%
  filter(!is.na(tradeflow_comtrade_o)) %>%
  slice_max(tradeflow_comtrade_o, n = 3) %>%
  select("Year" = year,
         "Country" = iso3_d,
         "Export value (US$)" = tradeflow_comtrade_o)

# Highest three import value observation
largest_import <- uk_imports %>%
  filter(!is.na(tradeflow_comtrade_d)) %>%
  slice_max(tradeflow_comtrade_d, n = 3) %>%
  select("Year" = year,
         "Country" = country_o,
         "Import value (US$)" = tradeflow_comtrade_d)

kable(largest_export, caption = "Largest UK export observations (2010–2020)")
```

Table 1: Largest UK export observations (2010–2020)

Year	Country	Export value (US\$)
2019	USA	71233628586
2015	USA	69299514923
2018	USA	65935483713

```
kable(largest_import, caption = "Largest UK import observations (2010–2020)")
```

Table 2: Largest UK import observations (2010–2020)

Year	Country	Import value (US\$)
2014	Germany	99822005227
2015	Germany	94074991076
2018	Germany	91018614254

The first table shows the top three observations of UK exports in any year between 2010 and 2020. All three observations describe UK export to the US. The second table shows the country from which the UK imported the most in any year over the same period. This table shows that the top three observations are all from Germany.

These peak observations only describe a small bit of UK trade relations, but already suggest patterns of strong and stable bilateral trade relationships. With regards to the research question posed in this project, these initial pieces of information depict both EU-members like Germany, and non-EU members like the US as important trading partners for the UK during the 2010s.

The following tables takes this approach one step further. The next tables show the UK's main trading partners for imports and exports for the years 2010 and 2020. Each table provide the top ten values of tradeflow in the given year, as reported by the UK, providing a slightly broader impression of main trading partners.

```
# Top 10 UK import observations in 2010
uk_imports_2010 %>%
  arrange(desc(tradeflow_comtrade_d)) %>% # Reported by the UK (inlcudes cost, freight, insurance)
  slice(1:10) %>%
  select("Partner country" = country_o,
         "EU member" = eu_o,
         "UK imports (US$)" = tradeflow_comtrade_d) %>%
  kable(caption = "Top 10 UK import observations in 2010")
```

Table 3: Top 10 UK import observations in 2010

Partner country	EU member	UK imports (US\$)
Germany	1	75668009770
China	0	61731379952
United States	0	56071154562
Netherlands	1	42284590441
France	1	37226836824
Norway	0	29759166142
Belgium	1	26542332275
Italy	1	22183948755
Ireland	1	20322916576
Canada	0	18977807949

```
# Top 10 UK import observations in 2020
uk_imports_2020 %>%
  arrange(desc(tradeflow_comtrade_d)) %>% # Reported by the UK (inlcudes cost, freight, insurance)
```

```

slice(1:10) %>%
select("Partner country" = country_o,
      "EU member" = eu_o,
      "UK imports (US$)" = tradeflow_comtrade_d) %>%
kable(caption = "Top 10 UK import observations in 2020")

```

Table 4: Top 10 UK import observations in 2020

Partner country	EU member	UK imports (US\$)
China	0	75387585801
Germany	1	73106644098
Netherlands	1	46242605112
United States	0	45015971784
France	1	30097924575
Belgium	1	28795052781
Italy	1	23461005557
Spain	1	17782562269
Ireland	1	17347578650
Poland	1	13926793946

The above two tables show the top ten origin countries for imports to the UK in 2010 and 2020. Both years are spaced from the referendum, with 2010 being well before and 2020 being the year the UK formally left the union. With this in mind, what is notable about the two tables is that there are notably more EU members in the top ten import observations for 2020 than for 2010. Only China and the United states remain of the list, while both Norway and Canada are replaced by EU-member states. While China replaces Germany as the top origin country for imports to the UK, the US falls on the ranking by both value and position, with the Netherlands climbing upwards. Otherwise, all EU members on the 2010 ranking are still present in 2020.

```

# Top 10 UK export observations in 2010
uk_exports_2010 %>%
  arrange(desc(tradeflow_comtrade_o)) %>%
  slice(1:10) %>%
  select("Partner country" = country_d,
        "EU member" = eu_d,
        "UK exports (US$)" = tradeflow_comtrade_o) %>%
kable(caption = "Top 10 UK export observations in 2010")

```

Table 5: Top 10 UK export observations in 2010

Partner country	EU member	UK exports (US\$)
United States	0	58549442646
Germany	1	44636263455
Netherlands	1	32130229568
France	1	31618613633
Ireland	1	25674226794
Belgium	1	20185973554
Spain	1	15039389368
Switzerland	0	15003904259
Italy	1	13650847335
China	0	11169271226

```
# Top 10 UK export observations in 2020
uk_exports_2020 %>%
  arrange(desc(tradeflow_comtrade_o)) %>%
  slice(1:10) %>%
  select("Partner country" = country_d,
         "EU member" = eu_d,
         "UK exports (US$)" = tradeflow_comtrade_o) %>%
  kable(caption = "Top 10 UK export observations in 2020")
```

Table 6: Top 10 UK export observations in 2020

Partner country	EU member	UK exports (US\$)
United States	0	54559252906
Germany	1	39408894992
Ireland	1	27811794760
Netherlands	1	25302032234
France	1	23844116834
China	0	18029536096
Belgium	1	13645336705
Italy	1	10924072236
Spain	1	10592967561
Hong Kong SAR China	0	9379486980

Similar patterns can be traced in the corresponding tables for UK exports. From these tables, one can observe that the main destinations for UK exports remained fairly stable from 2010 to 2020. Among the top ten export destinations for the UK, nine out of ten countries are

the same for 2010 and 2020. Only Switzerland fell out of the top ten observations, and was replaced by Hong Kong. It is notable from these data that the majority of top trading partners for the UK are European states (and EU-members), although the US and China help counter this image. However, these are only the very top observations for each year. The total number of trading partners is considerably higher. To get a better understanding of the bigger picture provided by the data, further analyses are in place. The next two figures will look at the total values of UK imports and exports to EU and non-EU partners.

```
# Preparing data: UK trade with EU and non-EU partners
uk_trade_eu <- gravity %>%
  group_by(year, partner_eu) %>%
  summarise(exports = sum(tradeflow_comtrade_o, na.rm = TRUE),
            imports = sum(tradeflow_comtrade_d, na.rm = TRUE),
            .groups = "drop")

# Create plot: UK exports to EU vs non-EU
ggplot(uk_trade_eu, aes(x = year, y = exports, color = factor(partner_eu))) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = 2016, linetype = "dashed") +
  geom_vline(xintercept = 2020, linetype = "dashed") +
  labs(title = "UK Exports to EU and Non-EU Partners",
       x = "Year",
       y = NULL,
       color = "EU partner") +
  scale_x_continuous(n.breaks = 7) +
  scale_y_continuous(labels = scales::label_dollar(big.mark = " ")) +
  theme_minimal()
```

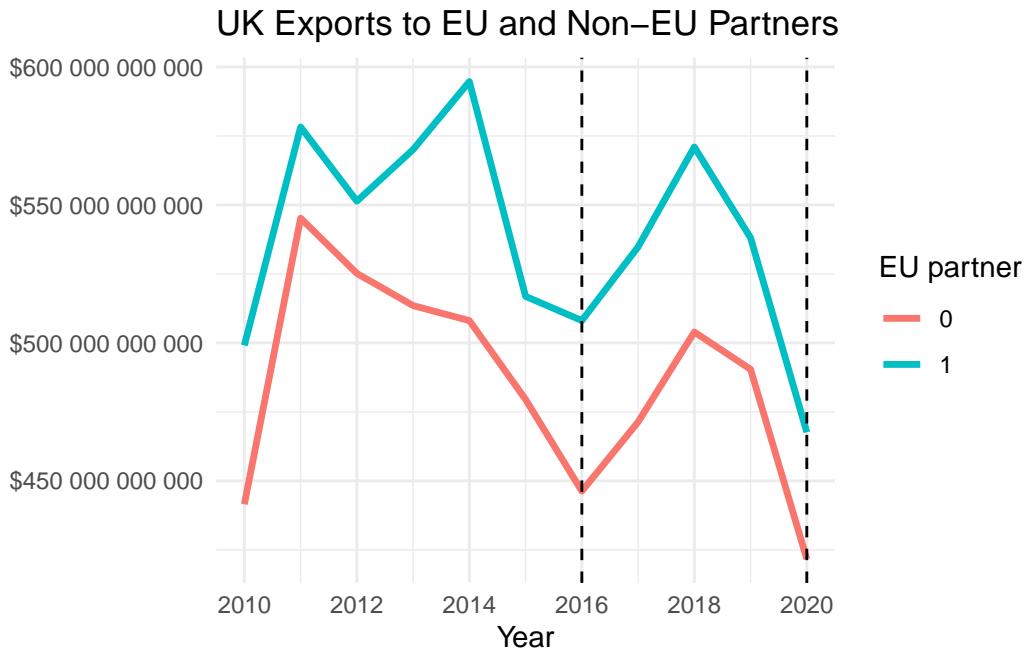


Figure 3: UK exports to EU and non-EU partners (2010-2020)

Figure 3 reveals how the UK has consistently exported for higher values to EU-member states than non-EU members. Otherwise, developments for both groups follow each other fairly close.

```
# Create plot: UK imports from EU vs non-EU
ggplot(uk_trade_eu, aes(x = year, y = imports, color = factor(partner_eu, labels = c("Non-EU",
)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = 2016, linetype = "dashed") +
  geom_vline(xintercept = 2020, linetype = "dashed") +
  labs(
    title = "UK Imports from EU and Non-EU Partners",
    x = "Year",
    y = NULL,
    color = "Partner type"
  ) +
  scale_x_continuous(n.breaks = 7) +
  scale_y_continuous(labels = scales::label_dollar(big.mark = " ")) +
  theme_minimal()
```

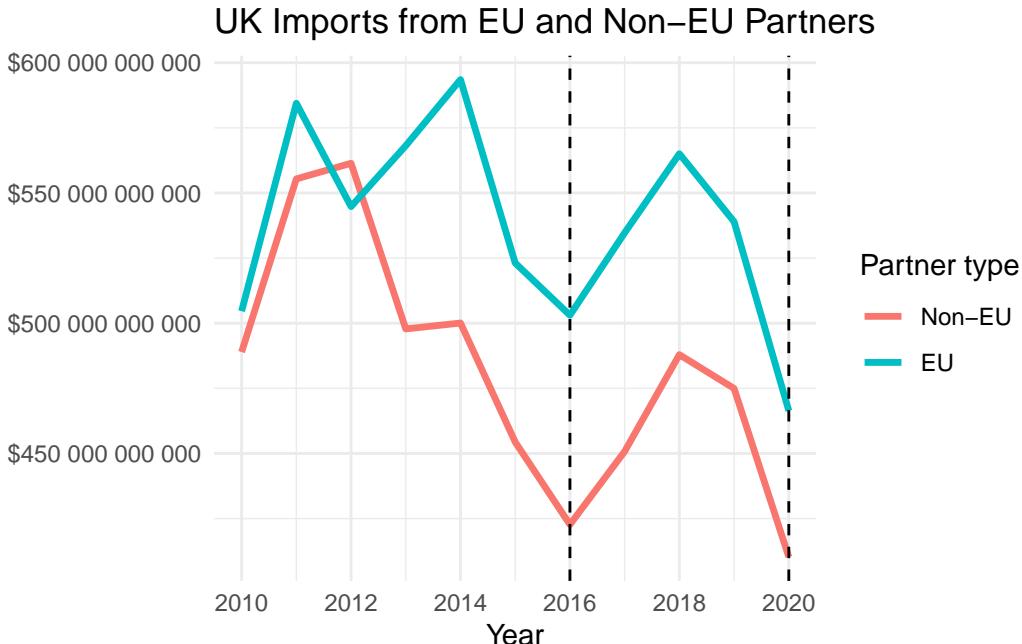


Figure 4: UK imports from EU and non-EU partners (2010-2020)

Figure 4 compares the UK's imports from EU and non-EU partners between 2010 and 2020. Likewise as the export trends showed in Figure 3, imports from EU partners are consistently higher than imports from non-EU countries throughout. This may be reflective the UK's strong integration with the European mainland and the EU single market before Brexit.

Before the next plot is presented, an AI disclaimer is in place. The data wrangling undergone for the next plot included another AI inquiry. The original code for the `eu_share` object included a duplicate observation for every year with missing values. A request was made to an AI model to tidy the existing code, and the following code chunk is a restructuring of the faulty original code. As the `eu_share` object only holds 11 observations it was easy to compare to the faulty version, which again made it possible to verify that the restructured code indeed had only removed the duplicate observations.

```
# Prepare data: EU share of UK trade
# Had AI rewrite this code as the previous version gave duplicate NAs that
# in turn made one of the graphs not functional
eu_share <- uk_trade_eu %>%
  group_by(year) %>%
  summarise(
    total_exports = sum(exports, na.rm = TRUE),
    total_imports = sum(imports, na.rm = TRUE),
```

```

eu_exports = sum(exports[partner_eu == 1], na.rm = TRUE),
eu_imports = sum(imports[partner_eu == 1], na.rm = TRUE),
.groups = "drop"
) %>%
mutate(
  eu_share_exports = eu_exports / total_exports,
  eu_share_imports = eu_imports / total_imports
) %>%
select(year, eu_share_exports, eu_share_imports)

```

```

# Create plot: EU share of UK trade
ggplot(eu_share, aes(x = year)) +
  geom_line(aes(y = eu_share_exports, color = "Exports"), size = 1.2) +
  geom_line(aes(y = eu_share_imports, color = "Imports"), size = 1.2) +
  geom_vline(xintercept = 2016, linetype = "dashed") +
  geom_vline(xintercept = 2020, linetype = "dashed") +
  scale_x_continuous(n.breaks = 11) +
  scale_y_continuous(labels = percent_format()) +
  labs(title = "EU Share of UK Exports and Imports",
       x = "Year",
       y = NULL,
       color = "Flow"
  ) +
  theme_minimal()

```

EU Share of UK Exports and Imports



Figure 5: EU share of UK trade

The graph in Figure 5 displays the EU's share of UK exports and imports from 2010 to 2020. Both export and import shares remain slightly above 50% across the period, indicating that the EU consistently accounts for a little over half of the UK's trade. This corresponds nicely with the findings from the last two plots. The lines fluctuate from year to year, and a small dip appears around 2012. This may be related to broader economic conditions in Europe at the time, such as weaker demand in several EU economies, although the graph alone cannot confirm causality. After the 2016 referendum, there is a modest downward movement, especially for imports, but the shares stabilise again in the following years. Toward 2020, both series remain close to earlier levels. The graph shows moderate fluctuations but no clear or dramatic shift in the UK's relative trade dependence on the EU during this period. The next plot will summarise this information in a more complex constellation.

```
# Total imports

df <- uk_imports %>%
  group_by(year, eu_o) %>%
  summarise(total_imports = sum(tradeflow_comtrade_o, na.rm = TRUE), .groups = "drop")

a <- ggplot(df, aes(year, total_imports, fill = factor(eu_o))) +
  geom_col() +
  scale_x_continuous(n.breaks = 7) +
  theme_minimal()
```

```

scale_y_continuous(labels = scales::label_dollar(big.mark = " ")) +
  labs(#title = "Total UK imports in US$ (2010-2020)",
       x = NULL,
       y = NULL) +
  scale_fill_manual(
    name = "EU member \nstatus",
    labels = c("0" = "Non-members", "1" = "Members"),
    values = c("0" = "#FFCC00", "1" = "#003399")) +
  theme_minimal()

# Total exports
df <- uk_exports %>%
  group_by(year, eu_d) %>%
  summarise(total_exports = sum(tradeflow_comtrade_d, na.rm = TRUE), .groups = "drop")

b <- ggplot(df, aes(year, total_exports, fill = factor(eu_d))) +
  geom_col() +
  scale_x_continuous(n.breaks = 7) +
  scale_y_continuous(labels = scales::label_dollar(big.mark = " "),
                     limits = c(0, 600000000000)) +
  labs(#title = "Total UK exports in US$ (2010-2020)",
       x = NULL,
       y = NULL) +
  scale_fill_manual(
    name = "EU member \nstatus",
    labels = c("0" = "Non-members", "1" = "Members"),
    values = c("0" = "#FFCC00", "1" = "#003399")) +
  theme_minimal()

(a + labs(subtitle = "Imports")) / (b + labs(subtitle = "Exports")) +
  plot_annotation(title = "Total UK imports and exports in US$ (2010-2020)") +
  plot_layout(guides = "collect")

```

Total UK imports and exports in US\$ (2010–2020)

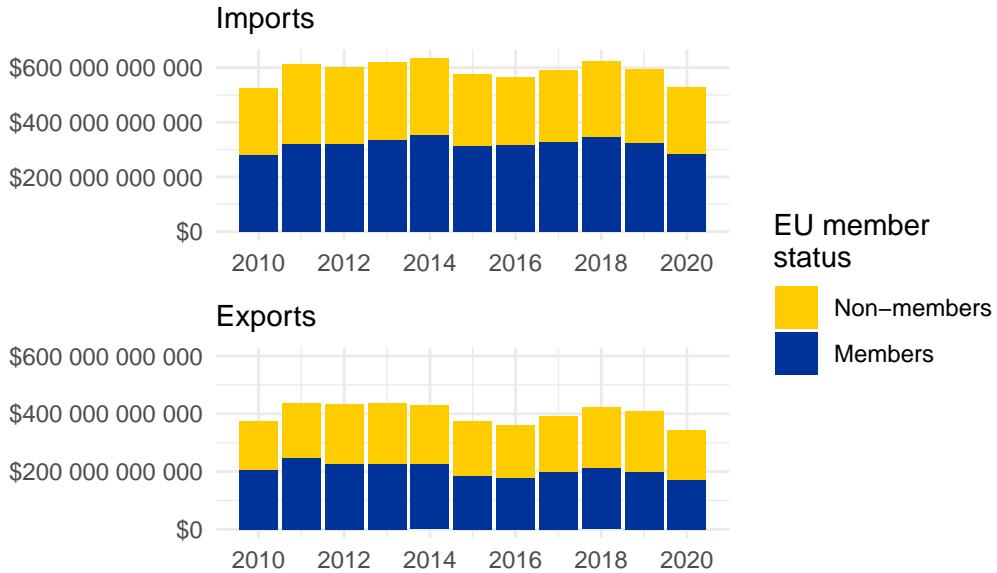


Figure 6: Total of UK imports and exports (2010-2020)

Figure 6 provides a concise visualization of the already presented information of the UK's trading relationship towards EU member states and non-EU members in the period 2010-2020. Consistent with previous plots, both imports and exports seem to dip around the referendum. Moreover, the reported tradeflow with EU members actually increased for the first couple of years after the referendum, before sinking in the years after 2018. This also seems to be the case for tradeflows with states that are not members of the EU, although this is harder to deduce from this figure. To see whether the increase or decrease in tradeflow at any point was greater or lesser for EU-members than others requires new analyses.

3.2 Data modelling

From the last figures it was possible to observe a rise in tradeflow between 2016 and 2018, and a decrease between 2018 and 2020. This poses the question of whether these trends were stronger or weaker with trading partners within the EU than those outside the union. This section will explore this question, first using model visualizations before turning to a regression table.

```
# Creating a reproducible jitter for the function
jitter <- (position_jitter(seed = 10, width = 0.2, height = 0))
```

```

# Preparing data for upcoming models
sim_exports1 <- uk_exports %>%
  filter(year %in% 2016:2018,
    !is.na(tradeflow_comtrade_o))

sim_exports2 <- uk_exports %>%
  filter(year %in% 2018:2020,
    !is.na(tradeflow_comtrade_o))

# Define a function that will create the plots
create_model_plot <- function(model_data) {

  # Model for EU-member states
  model1 <- lm(tradeflow_comtrade_o ~ year, data = model_data %>% filter(eu_d == 1))

  # Model for non-EU partners
  model2 <- lm(tradeflow_comtrade_o ~ year, data = model_data %>% filter(eu_d == 0))

  # Create prediction grids
  grid <- model_data %>%
    data_grid(year)

  grid1 <- grid %>%
    add_predictions(model1)

  grid2 <- grid %>%
    add_predictions(model2)

  # Visualise the models
  plot <- ggplot(model_data, aes(x = year, y = tradeflow_comtrade_o, colour = factor(eu_d))) +
    geom_point(position = jitter,
      aes(alpha = 0.9)) + # slightly transparent points
    geom_line(aes(y = pred), data = grid1, colour = "#003399", size = 1.5) +
    geom_line(aes(y = pred), data = grid2, colour = "orange", size = 1.5) +
    scale_y_log10(labels = scales::label_dollar(big.mark = " "),
      n.breaks = 6) +
    coord_cartesian(ylim = c(1e6, 1e11)) + # scientific notations, will not cut observations
    scale_x_continuous(n.breaks = 3) +
    labs(x = NULL,
      y = NULL,

```

```

    alpha = NULL) +
scale_colour_manual(
  name = "EU member \nstatus",
  labels = c("0" = "Non-members", "1" = "Members"),
  values = c("0" = "#FFCC00", "1" = "#003399")) +
theme_minimal()
}

a <- create_model_plot(sim_exports1)
b <- create_model_plot(sim_exports2)

((a + labs(subtitle = "UK exports (2016–2018 and 2018–2020)")) +
(b + theme(axis.text.y = element_blank()))
)) +
plot_layout(guides = "collect") &
guides(alpha = "none")

```

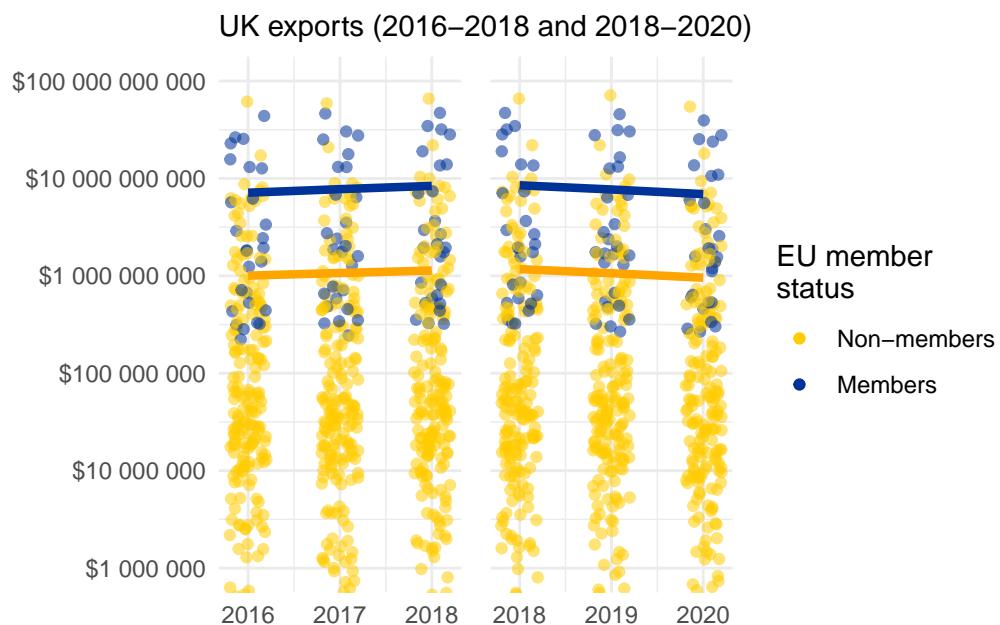


Figure 7: Predicted UK exports by EU-member status

As previous analyses revealed, the years following the referendum in 2016 seemed to include a period of increased trade between 2016-2018, followed by a decrease in 2018-2020. This seemed to be the case for trade with both EU-members and non-EU members, but do these

developments differ significantly? Figure 7 present predicted models for these developments. The blue lines are the predicted models for exports to EU-member states, while the orange lines are the predicted models for exports to non-EU member states. Judging the plot visually, it is difficult to see significant differences in the incline of the models between EU-member states and non-EU members. The models were presumed linear, which may have been a limitation if another model family would have made a better fit. As the y-scale is exponential rather than continuous, this may also have affected the visibility of any significant difference.

```
# Repeat last code chunk but for imports (uses the same jitter)

# Preparing data for upcoming models
sim_imports1 <- uk_imports %>%
  filter(year %in% 2016:2018,
    !is.na(tradeflow_comtrade_d))

sim_imports2 <- uk_imports %>%
  filter(year %in% 2018:2020,
    !is.na(tradeflow_comtrade_d))

# Define a function that will create the plots
create_model_plot <- function(model_data) {

  # Model for EU-member states
  model1 <- lm(tradeflow_comtrade_d ~ year, data = model_data %>% filter(eu_o == 1))

  # Model for non-EU partners
  model2 <- lm(tradeflow_comtrade_d ~ year, data = model_data %>% filter(eu_o == 0))

  # Create prediction grids
  grid <- model_data %>%
    data_grid(year)

  grid1 <- grid %>%
    add_predictions(model1)

  grid2 <- grid %>%
    add_predictions(model2)

  # Visualise the models
  plot <- ggplot(model_data, aes(x = year, y = tradeflow_comtrade_d, colour = factor(eu_o))) +
    geom_point(position = jitter,
      aes(alpha = 0.9)) + # slightly transparent points
    geom_line(aes(y = pred), data = grid1, colour = "#003399", size = 1.5) +
    geom_line(aes(y = pred), data = grid2, colour = "#FF8C00", size = 1.5)
}
```

```

geom_line(aes(y = pred), data = grid2, colour = "orange", size = 1.5) +
  scale_y_log10(labels = scales::label_dollar(big.mark = " "),
    n.breaks = 6) +
  coord_cartesian(ylim = c(1e6, 1e11)) + # scientific notation. Will not cut observations
  scale_x_continuous(n.breaks = 3) +
  labs(x = NULL,
    y = NULL,
    alpha = NULL) +
  scale_colour_manual(
    name = "EU member \nstatus",
    labels = c("0" = "Non-members", "1" = "Members"),
    values = c("0" = "#FFCC00", "1" = "#003399")) +
  theme_minimal()
}

a <- create_model_plot(sim_imports1)
b <- create_model_plot(sim_imports2)

((a + labs(subtitle = "UK imports (2016-2018 and 2018-2020)")) +
  (b + theme(axis.text.y = element_blank()))
  )) +
  plot_layout(guides = "collect") &
  guides(alpha = "none")

```

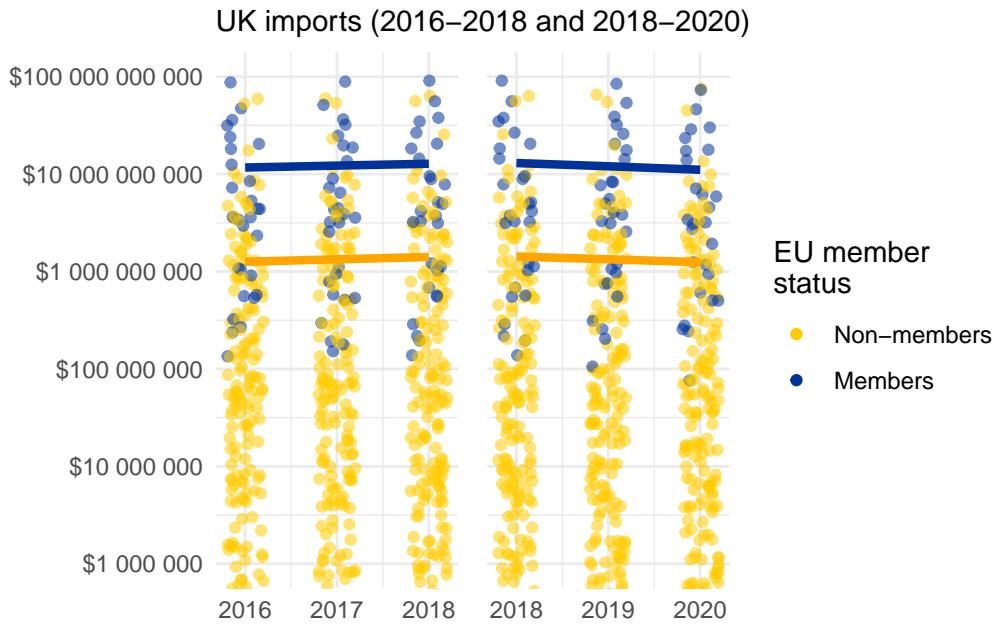


Figure 8: Predicted UK imports by EU-member status

The description provided by Figure 7 can also largely be said to apply for Figure 8. This plot visualizes the model predictions of UK imports to EU-members and non-EU members. The incline for 2016–2018 and the decline for 2018–2020 is visible in all models, but judging by the visuals, the predictions do not vary significantly between EU-members and non-EU members. This may however be the fault of plot design, as the models are all presumed linear and the y-axis is exponential. The last analysis of this report will therefore be the inspection of a regression table.

```
# Prepare data
regdata <- gravity %>%
  mutate(
    ln_exports = log(tradeflow_comtrade_o + 1),
    post2016 = if_else(year >= 2016, 1, 0)
  )

# Run model
model <- lm(ln_exports ~ partner_eu * post2016, data = regdata)

# Show regression results in the knitted report
summary(model)
```

```

Call:
lm(formula = ln_exports ~ partner_eu * post2016, data = regdata)

Residuals:
    Min      1Q  Median      3Q     Max 
-14.3234 -1.6609  0.0447  2.1930  7.3175 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 17.72818   0.06925 256.008 <0.0000000000000002 ***  
partner_eu    4.09955   0.18693  21.931 <0.0000000000000002 ***  
post2016     -0.03746   0.10381  -0.361     0.718    
partner_eu:post2016 -0.02923   0.27626  -0.106     0.916    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.091 on 4174 degrees of freedom
(981 observations deleted due to missingness)
Multiple R-squared:  0.1743,    Adjusted R-squared:  0.1737 
F-statistic: 293.8 on 3 and 4174 DF,  p-value: < 0.0000000000000022

```

The regression seeks to examine whether UK exports to EU partners experienced a different shift after the 2016 referendum compared with exports to non-EU markets. The crucial indicator is the interaction term `partner_eu * post2016`. In the output, this coefficient is -0.0296 with a standard error of 0.2744 and a p-value of 0.914 . The estimate is close to zero, and the p-value is above conventional significance levels, which shows that the model can not statistically distinguish this effect. The regression finds no measurable divergence between EU and non-EU export trends in the post-2016 period. The other coefficients reinforce this interpretation. `Partner_eu` is strongly positive (4.09 , $p < 0.001$), and can indicate that EU partners receive much higher export volumes than non-EU partners before 2016, while `post2016` is small and statistically insignificant (-0.037 , $p = 0.719$), suggesting no systematic change in non-EU trade after 2016.

4.0 Conclusion

The research question asked how the United Kingdom's trading patterns with EU and non-EU partners have changed since the 2016 referendum. The descriptive evidence shows that the EU remained the UK's main trading partner throughout 2010–2020, consistently accounting for slightly more than half of exports and imports. At the same time, the figures indicate some gradual shifts. Both export and import shares to the EU decline modestly after 2016, especially on the import side, suggesting that trade with non-EU partners grew somewhat

faster in relative terms. These adjustments are small and do not reflect any sharp break, but they point to a slow rebalancing rather than a complete change in trade structure.

At the same time, the overall movement of the EU and non-EU series also suggests that broader global conditions influenced UK trade during this period. The dip around 2012 aligns with economic weakness in parts of Europe, while the fluctuations in 2020 correspond to the worldwide COVID-19 shock also visible in the GDP per capita comparison. Because these external factors overlap with the post-referendum years, the descriptive patterns cannot isolate the specific effect of Brexit. Within these constraints, the results indicate relatively stable UK–EU trade levels, with indications of a modest shift toward non-EU partners after 2016.

5.0 Limitations

The analysis is subject to several limitations linked to the scope of the data and the period covered. The gravity dataset used in the project did not include observations after 2021, which limits the ability to assess UK–EU trade developments in the most recent years. This also means that the period following the full implementation of the Trade and Cooperation Agreement cannot be analysed in detail. In addition, the years around 2020–2021 were strongly affected by the global COVID-19 pandemic, which caused large and widespread disruptions in international trade. These shocks make it difficult to separate changes related to Brexit from changes driven by global supply-chain disruptions and demand fluctuations during the pandemic. These disruptions make it difficult to separate Brexit-related changes from broader global shocks in this part of the dataset. Finally, the descriptive patterns presented in the graphs cannot on their own establish causal effects. While the figures help illustrate broad developments in trade shares and trade volumes, they do not isolate the impact of any single event or policy change. Taken together, these limitations imply that the results should be interpreted as descriptive patterns within the available time frame rather than definitive evidence of Brexit-induced changes.

6.0 Bibliography

Conte, M., P. Cotterlaz and T. Mayer (2022), “The CEPII Gravity database”. CEPII Working Paper N°2022-05, July 2022.

European Commission. (2021). The EU-UK Trade and Cooperation Agreement. https://commission.europa.eu/strategy-and-policy/relations-united-kingdom/eu-uk-trade-and-cooperation-agreement_en

World Bank. (2024). World Development Indicators. Retrieved from <https://data.worldbank.org/source/world-development-indicators>