Estimating the impact of restrictions to trade in service sectors

B173504

2022-08-15

R Data Compilation

This sections includes the R code to replicate this paper.

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, eval = FALSE)
# Set knit directory to working directory
rm(list=ls())
knitr::opts_knit$set(root.dir = normalizePath(".."))

#download packages and install if needed
list.of.packages <- c('data.table', 'tidyverse', 'lubridate', 'RSQLite', 'sqldf',
'estimatr', 'lmtest', 'haven', 'fixest', 'dplyr', 'tidyr', 'OECD', 'modelsummary',
'kableExtra', "gt", 'flextable', 'corrplot',
"wbstats","ggpubr", "viridis", "countrycode",
"ggh4x", "hrbrthemes", "ggfan")
new.packages <- list.of.packages[!(list.of.packages
      %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packaes)
for (p in list.of.packages) {
library(p, character.only = TRUE)
}
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
##
## Attaching package: 'lubridate'
##
##
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
##
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
##
##
## Loading required package: gsubfn
##
## Loading required package: proto
##
## Loading required package: zoo
##
##
## Attaching package: ’zoo’
##
##
## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric
##
##
##
## Attaching package: ’kableExtra’
##
##
## The following object is masked from ’package:dplyr’:
##
##     group_rows
##
##
##
## Attaching package: ’gt’
##
##
## The following object is masked from ’package:modelsummary’:
##
##     escape_latex
##
##
##
## Attaching package: ’flextable’
##
##
## The following objects are masked from ’package:kableExtra’:
##
##     as_image, footnote
##
##
## The following object is masked from ’package:purrr’:
##
##     compose
##
```

```
##
## corrplot 0.92 loaded
##
##
## Attaching package: 'ggpubr'
##
##
## The following objects are masked from 'package:flextable':
##
##     border, font, rotate
##
##
## Loading required package: viridisLite
##
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
##
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
##
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

ITPD-E

Load the ITPD-E Release 2 dataset

```
df <- fread("B173504/ITPD_E_R02_no_names.csv",
            header = TRUE)
```

```
df_ITPDE <- df %>% filter(year %in% 2014:2019) %>%
  filter(industry_id %in% c(156, 158, 159, 160, 162, 163, 169))
# H       Transport 156
# F       Construction 158
# K (40%) Insurance and pension services 159
# K (60%) Financial services 160
# J       Information services 162
# M+N     Other business services 163
# G       Trade related services 169
```

DGD

Load the DGD

```
df_dgd <- readRDS("B173504/df_dgd.Rda")
```

Rename columns in the DGD

```
df_dgd <- df_dgd %>%
  mutate(exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="MYS.Y", "MYS", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="ZAF.X", "ZAF", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="VNM.X", "VNM", exporter_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="MYS.Y", "MYS", importer_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="ZAF.X", "ZAF", importer_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="VNM.X", "VNM", importer_dynamic_code))
```

```
df_dgd <- df_dgd %>%
  mutate(
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="MYS.Y", "MYS", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="ZAF.X", "ZAF", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="VNM.X", "VNM", exporter_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="MYS.Y", "MYS", importer_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="ZAF.X", "ZAF", importer_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="VNM.X", "VNM", importer_dynamic_code))
```

Additional Domestic Trade Data

Add in missing domestic trade data (production minus exports) sourced from OECD STAN, OECD Supply-Use Tables and OECD Input-Output Tables. Domestic consumption of domestic production (DCDP) is called domestic trade data or domestic consumption in the dissertation.

Exports

Calculate exports using the ITPD-E data. Note: it is important to keep all countries at this stage as we need to calculate total exports per year by country and industry.

```
df_exports <- df_ITPDE %>%
  filter(exporter_dynamic_code != importer_dynamic_code) %>%
  group_by(year, exporter_dynamic_code, industry_id) %>%
  summarise(total_exports = sum(trade)) %>%
    mutate(
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="MYS.Y", "MYS", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="ZAF.X", "ZAF", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="VNM.X", "VNM", exporter_dynamic_code)) %>%
  ungroup()
```

Go back to the ITPDE data set now and filter out all observations where 0 trade value has been assigned due to missing/unknown data

Create a vector of countries included in the STRI data and then create a dataframe for the observations where data has been assigned to 0 due to it being unknown or missing.

```
STRI_countries <- c("AUS", "AUT", "BEL", "CAN", "CZE", "DNK", "FIN", "FRA", "DEU", "GRC",
                    "HUN", "ISL", "ITA", "JPN", "KOR", "LUX", "MEX", "NLD", "NZL", "NOR",
                    "POL", "PRT", "SVK", "ESP", "SWE", "CHE", "TUR", "GBR", "USA", "IRL",
                    "BRA", "CHL", "CHN", "EST", "IND", "IDN", "ISR", "RUS", "SVN",
                    "ZAF.X", "COL", "LVA", "CRI", "LTU", "MYS.Y","THA", "KAZ", "PER",
                    "SGP", "VNM.X")

df_ITPDE <-  df_ITPDE %>%
  filter(exporter_dynamic_code %in%
           STRI_countries & importer_dynamic_code %in% STRI_countries)

df_missing_trade_data <- df_ITPDE %>%
  filter(flag_zero == "u") %>%
  filter(exporter_dynamic_code %in%
           STRI_countries & importer_dynamic_code %in% STRI_countries)
```

Check the data

```
df_test <- df_ITPDE %>%
  filter(exporter_dynamic_code == "JPN" & importer_dynamic_code == "JPN")
```

Filter ITPDE data to only include the 50 countries that have an STRI

```
# filter out all observations where 0 trade value has been assigned due to
#missing/unknown data
df_ITPDE <- df_ITPDE %>%
  filter(flag_zero != "u") %>%
  mutate(
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="MYS.Y", "MYS", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="ZAF.X", "ZAF", exporter_dynamic_code),
    exporter_dynamic_code =
      ifelse(exporter_dynamic_code=="VNM.X", "VNM", exporter_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="MYS.Y", "MYS", importer_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="ZAF.X", "ZAF", importer_dynamic_code),
    importer_dynamic_code =
      ifelse(importer_dynamic_code=="VNM.X", "VNM", importer_dynamic_code))
```

Production Data

Load supplementary production data sourced from OECD and exchange rates (ex_rate), as non-USD production data needs to be converted to USD. Exchange rates are annual averages sourced from the IMF.

```
df_prod <- fread("B173504/prod_data.csv",
            header = TRUE)
df_exrate <- fread("B173504/exrate_data.csv",
            header = TRUE)
# ISIC    Sector                        ITPDE
# H       Transport                       156
# F       Construction                    158
# K (40%) Insurance and pension services  159
# K (60%) Financial services              160
# J       Information services            162
# M+N     Other business services         163
# G       Trade related services          169

# Rename column names
df_prod <- df_prod %>%
  rename(exporter_dynamic_code = LOCATION, year = Year,
         industry_id = ACTIVITY, total_prod = Value)

df_exrate <- df_exrate %>%
  rename(exporter_dynamic_code = Country, year = Year)

df_prod_NCU <- df_prod %>%
     filter(`Unit Code` != "USD")

df_prod_NCU  <- df_prod_NCU %>%
  left_join(df_exrate, by = c("year","exporter_dynamic_code"))
df_prod_NCU  <- df_prod_NCU  %>%
  mutate(total_prod = total_prod / ex_rate) %>%
  mutate(
  industry_id = ifelse(industry_id %in% c("ZF","D41T43"), 158, industry_id),
  industry_id = ifelse(industry_id %in% c("ZG","D45T47"), 169, industry_id),
  industry_id = ifelse(industry_id %in% c("ZH","D49T53"), 156, industry_id),
  industry_id = ifelse(industry_id %in% c("ZJ","D58T63"), 162, industry_id)
  )

df_prod_NCU_final <- df_prod_NCU %>%
  select("year","exporter_dynamic_code", "industry_id", "total_prod") %>%
  filter(industry_id %in% c(156, 158, 162, 169))

# Remove df_exrate dataframe
rm(df_exrate)
```

This production data is messy. For countries and years where there are OECD data, the industry data is organised in a way that we can obtain all sectors included in this analysis. For the ADB, due to the level of granularity in the data, I can only obtain data for construction, finance, insurance and distribution services (and only with some aggregations).

Professional services and Administrative services sourced from the OECD need to be added together to create the other business services sector.

The finance sector needs to be split 60% for finance and 40% for insurance for both the OECD and ADB data.

The distributions services data from the ADB needs to be aggregated together.

The OECD data needs to be converted to USD using the IMF international financial statistics. I convert the data using the average exchange rate for the year.

```
## ZM or "D69T75" + ZN or "D77T82" = Other business services 163
df_prod_obs <- df_prod_NCU %>%
  filter(industry_id %in% c("ZM" , "ZN", "D69T75", "D77T82")) %>%
  group_by(year, exporter_dynamic_code) %>%
  summarise(total_prod = sum(total_prod)) %>%
  ungroup()
df_prod_obs$industry_id <- "163"

## ZK or "D64T66"  = Insurance 159 (40%) and Finance 160 (60%)
df_prod_fin <- df_prod_NCU %>%
  filter(industry_id %in% c("ZK", "D64T66")) %>%
  group_by(year, exporter_dynamic_code) %>%
  summarise("160" = 0.6*total_prod, "159" = 0.4*total_prod) %>%
  ungroup()
df_prod_fin <- df_prod_fin %>%
  gather(industry_id, total_prod, "160":"159", -c(year, exporter_dynamic_code))

# Merge dataframes together to get final production data
df_prod_NCU <- rbind(df_prod_NCU_final, df_prod_obs,
                 df_prod_fin)

# Make industry_id an integer
df_prod_NCU$industry_id <- as.integer(df_prod_NCU$industry_id)

# Remove dataframes that are no longer needed
#rm(df_OECD_final, df_ADB, df_prod_obs, df_prod_fin, df_prod_fin1, df_prod_dis, df_OECD)
```

Load OECD that is in USD terms

```
df_prod_USD <- df_prod %>%
  filter(`Unit Code` == "USD")

# ISIC    Sector                        ITPDE
# H       Transport                       156
# F       Construction                    158
# K (40%) Insurance and pension services  159
# K (60%) Financial services              160
# J       Information services            162
# M+N     Other business services         163
# G       Trade related services          169

df_prod_USD <- df_prod_USD %>%
  mutate(
  industry_id = ifelse(industry_id =="D41T43", 158, industry_id),
  industry_id = ifelse(industry_id =="D45T47", 169, industry_id))

df_prod_USD_final <- df_prod_USD %>%
  select("year","exporter_dynamic_code", "industry_id", "total_prod") %>%
  filter(industry_id %in% c(156, 158, 162, 169))

## "D49" + "D50" + "D51" + "D52" + "D53" = Transport 156 H
df_prod_trans <- df_prod_USD %>%
  filter(industry_id %in% c("D49" , "D50", "D51", "D52", "D53")) %>%
  group_by(year, exporter_dynamic_code) %>%
  summarise(total_prod = sum(total_prod)) %>%
  ungroup()
```

```
df_prod_trans$industry_id <- "156"

## "D58T60" + "D61" + "D62T63" = Info Services 162 J
df_prod_info <- df_prod_USD %>%
  filter(industry_id %in% c("D58T60" , "D61", "D62T63")) %>%
  group_by(year, exporter_dynamic_code) %>%
  summarise(total_prod = sum(total_prod)) %>%
  ungroup()
df_prod_info$industry_id <- "162"

## ZM or "D69T75" + ZN or "D77T82" = Other business services 163 M+N
df_prod_obs <- df_prod_USD %>%
  filter(industry_id %in% c("ZM" , "ZN", "D69T75", "D77T82")) %>%
  group_by(year, exporter_dynamic_code) %>%
  summarise(total_prod = sum(total_prod)) %>%
  ungroup()
df_prod_obs$industry_id <- "163"

## ZK or "D64T66"  = Insurance 159 (40%) and Finance 160 (60%)
df_prod_fin <- df_prod_USD %>%
  filter(industry_id %in% c("ZK", "D64T66")) %>%
  group_by(year, exporter_dynamic_code) %>%
  summarise("160" = 0.6*total_prod, "159" = 0.4*total_prod) %>%
  ungroup()
df_prod_fin <- df_prod_fin %>%
  gather(industry_id, total_prod, "160":"159", -c(year, exporter_dynamic_code))

# Merge dataframes together to get final production data
df_prod_USD <- rbind(df_prod_USD_final, df_prod_obs,
                 df_prod_fin, df_prod_info, df_prod_trans)

# Make industry_id an integer
df_prod_USD$industry_id <- as.integer(df_prod_USD$industry_id)
```

Calculate Domestic Trade Data

```
df_prod <- rbind(df_prod_USD, df_prod_NCU) %>%
# Other business services for Japan in 2019
add_row(exporter_dynamic_code = "JPN", year = 2019, industry_id = 163,
        total_prod = 645158.384) %>%
  filter(!(industry_id == 159 & exporter_dynamic_code == "LUX"))

# Merge production and export dataframes together
df_dom_data <- merge(df_prod, df_exports, by =
                c("year","exporter_dynamic_code", "industry_id"))

# Calculate domestic consumption (trade = production - exports)
# and set exporter = importer
df_dom_data <- df_dom_data %>%
  mutate(trade = total_prod - total_exports) %>%
  mutate(importer_dynamic_code = exporter_dynamic_code)
```

Merge with ITPD-E Data

Create new data bases. Note we do not need to filter out countries yet. This is done under section ‘Merge the STRI indexes into the final database’ when we drop observations with NA for the STRI.

Now we have two different datasets: 1. Only the ITPD-E data (df_ITPDE) 2. ITPD-E data with all additional trade data (df_ITPDE_new)

```
# Bind the new domestic data with the ITPDE data
df_ITPDE_new <- rbind(df_ITPDE, df_dom_data, fill = TRUE)

# Now we have two different datasets:
#1. Only the ITPD-E data (df_ITPDE)
#2. ITPD-E data with all additional trade data (df_ITPDE_new)

# Remove df_dom_data as no longer needed
rm(df_dom_data, df_dom_data_OECD, df_exports, df_prod)
```

Intra-EEA STRI

Load the Intra-EEA STRI data

```
df_STRI_EEA <- fread("B173504/STRI_INTRAEEA.csv", header = TRUE)
```

Filter dataset

```
df_STRI_EEA <- df_STRI_EEA %>%
  select(Year, COU, Value, SECT)
df_STRI_EEA <- df_STRI_EEA %>%
  rename(REPORTER = COU, SECTOR = SECT)
#STRI_sector_code  STRI sector
# "LSCAR"   "Logistics cargo-handling"
# "LSCUS"   "Logistics customs brokerage"
# "LSFGT"   "Logistics freight forwarding"
# "LSSTG"   "Logistics storage and warehouse"
# "PSACC"   "Accounting"
# "PSARC"   "Architecture"
# "PSENG"   "Engineering"
# "PSLEG"   "Legal"
# "ASMOT"   "Motion pictures"
# "ASBRD"   "Broadcasting"
# "ASSOU"   "Sound recording"
# "TC"      "Telecom"
# "TRAIR"   "Air transport"
# "TRROF"   "Road freight transport"
# "TRRAI"   "Rail freight transport"
# "CR"      "Courier"
# "DS"      "Distribution"
# "FSBNK"   "Commercial banking"
# "FSINS"   "Insurance"
# "CS"      "Computer"
# "CO"      "Construction"
# "TRMAR"   "Maritime transport"
```

Filter out sector that do not need aggregation

```
df_EEA_final <- df_STRI_EEA %>%
  filter(SECTOR %in% c("CO", "FSINS", "FSBNK", "DS")) %>%
  select(Year, REPORTER, Value, SECTOR)

# "CO"      Construction 158
# "FSINS"   Insurance and pension services 159
# "FSBNK"   Financial services 160
# "DS"      Trade related services 169
```

Create Intra-EEA Transport Composite STRI

```
# Create freight composite
df_freight <- df_STRI_EEA %>%
  filter(SECTOR %in% c("TRRAI" , "TRROF")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_freight$SECTOR <- "freight_comp"

# Create logistics composite
df_logistics <- df_STRI_EEA %>%
  filter(SECTOR %in% c("LSCAR", "LSCUS", "LSFGT", "LSSTG", "CR")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_logistics$SECTOR <- "logistics_comp"

# Create air and maritime dataframe
df_mari_air <- df_STRI_EEA %>%
  filter(SECTOR %in% c("TRMAR" , "TRAIR")) %>%
  select(Year, REPORTER, Value, SECTOR)

# Merge dataframes together
df_transport_STRI <- rbind(df_freight, df_logistics, df_mari_air)

# Create transport composite STRI
df_transport_STRI <- df_transport_STRI %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_transport_STRI$SECTOR <- "156"

# Check
df_test <- df_transport_STRI %>% filter(Year == 2014)
```

Create Intra-EEA Information Services Composite STRI

```r
# Create audio visual composite
df_audio_visual <- df_STRI_EEA %>%
  filter(SECTOR %in% c("ASMOT" , "ASBRD", "ASSOU")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_audio_visual$SECTOR <- "audio_visual_comp"

# Create telecomms and computer services dataframe
df_tc_cs <- df_STRI_EEA %>%
  filter(SECTOR %in% c("TC" , "CS")) %>%
  select(Year, REPORTER, Value, SECTOR)

# Merge dataframes together
df_info_STRI <- rbind(df_audio_visual, df_tc_cs)

# Create Information Services Composite STRI
df_info_STRI <- df_info_STRI %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_info_STRI$SECTOR <- "162"

df_test <- df_info_STRI %>% filter(Year == 2014)
```

Create Intra-EEA Other Business Services Composite STRI

```r
# Create architecture and engineering composite
df_arc_eng <- df_STRI_EEA %>%
  filter(SECTOR %in% c("PSARC" , "PSENG")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_arc_eng$SECTOR <- "arc_eng_comp"

# Create accounting and legal services dataframe
df_acc_leg <- df_STRI_EEA %>%
  filter(SECTOR %in% c("PSACC" , "PSLEG")) %>%
  select(Year, REPORTER, Value, SECTOR)

# Merge dataframes together
df_obs_STRI <- rbind(df_arc_eng, df_acc_leg)

# Create Other Business Services Composite STRI
df_obs_STRI <- df_obs_STRI %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_obs_STRI$SECTOR <- "163"

df_test <- df_obs_STRI %>% filter(Year == 2014)
```

Merge EEA STRI Sectors

Merge the EEA STRI sector dataframes into one dataframe and rename some sector codes

```
# Merge dataframes together
df_EEA_STRI <- rbind(df_obs_STRI, df_info_STRI, df_transport_STRI, df_EEA_final)
df_EEA_STRI <- df_EEA_STRI %>%
  mutate(
  SECTOR = ifelse(SECTOR=="DS", 169, SECTOR),
  SECTOR = ifelse(SECTOR=="FSINS", 159, SECTOR),
  SECTOR = ifelse(SECTOR=="FSBNK", 160, SECTOR),
  SECTOR = ifelse(SECTOR=="CO", 158, SECTOR),
  )
```

MFN STRI

Load the data. Note: This uses the standard STRI which is on a MFN basis.

```
df_STRI <- fread("B173504/STRI.csv",
           header = TRUE)
#STRI sector code STRI sector desc.
# "LSCAR"   "Logistics cargo-handling"
# "LSCUS"   "Logistics customs brokerage"
# "LSFGT"   "Logistics freight forwarding"
# "LSSTG"   "Logistics storage and warehouse"
# "PSACC"   "Accounting"
# "PSARC"   "Architecture"
# "PSENG"   "Engineering"
# "PSLEG"   "Legal"
# "ASMOT"   "Motion pictures"
# "ASBRD"   "Broadcasting"
# "ASSOU"   "Sound recording"
# "TC"      "Telecom"
# "TRAIR"   "Air transport"
# "TRROF"   "Road freight transport"
# "TRRAI"   "Rail freight transport"
# "CR"      "Courier"
# "DS"      "Distribution"
# "FSBNK"   "Commercial banking"
# "FSINS"   "Insurance"
# "CS"      "Computer"
# "CO"      "Construction"
# "TRMAR"   "Maritime transport"
```

Filter to just include headline STRI as this database also has the five policy categories STRIs

```
df_STRI <- df_STRI %>%
  select(Year, COU, CLAS, Value, SECT)  %>%
  filter(CLAS == "STRI") %>%
  rename(REPORTER = COU, SECTOR = SECT)
```

Filter out the STRI sectors that already have an exact match (i.e. that do not need aggregating)

```
# "CO"      Construction 158
# "FSINS"   Insurance and pension services 159
# "FSBNK"   Financial services 160
# "DS"      Trade related services 169

df_MFN_final <- df_STRI %>%
  filter(SECTOR %in% c("CO", "FSINS", "FSBNK", "DS")) %>%
  select(Year, REPORTER, Value, SECTOR)
```

Create MFN Transport Composite STRI

```
# Create freight composite
df_freight <- df_STRI %>%
  filter(SECTOR %in% c("TRRAI" , "TRROF")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_freight$SECTOR <- "freight_comp"

# Create logistics composite
df_logistics <- df_STRI %>%
  filter(SECTOR %in% c("LSCAR", "LSCUS", "LSFGT", "LSSTG", "CR")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_logistics$SECTOR <- "logistics_comp"

# Create air and maritime dataframe
df_mari_air <- df_STRI %>%
  filter(SECTOR %in% c("TRMAR" , "TRAIR")) %>%
  select(Year, REPORTER, Value, SECTOR)

# Merge dataframes together
df_transport_STRI <- rbind(df_freight, df_logistics, df_mari_air)

# Create transport composite STRI
df_transport_STRI <- df_transport_STRI %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_transport_STRI$SECTOR <- "156"

# Check
df_test <- df_transport_STRI %>% filter(Year == 2014)
```

Create MFN Information Services Composite STRI

```
# Create audio visual composite
df_audio_visual <- df_STRI %>%
  filter(SECTOR %in% c("ASMOT" , "ASBRD", "ASSOU")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_audio_visual$SECTOR <- "audio_visual_comp"

# Create telecomms and computer services dataframe
df_tc_cs <- df_STRI %>%
  filter(SECTOR %in% c("TC" , "CS")) %>%
  select(Year, REPORTER, Value, SECTOR)

# Merge dataframes together
df_info_STRI <- rbind(df_audio_visual, df_tc_cs)

# Create Information Services Composite STRI
df_info_STRI <- df_info_STRI %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_info_STRI$SECTOR <- "162"

df_test <- df_info_STRI %>% filter(Year == 2014)
```

Create MFN Other Business Services Composite STRI

```
# Create architecture and engineering composite
df_arc_eng <- df_STRI %>%
  filter(SECTOR %in% c("PSARC" , "PSENG")) %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_arc_eng$SECTOR <- "arc_eng_comp"

# Create accounting and legal services dataframe
df_acc_leg <- df_STRI %>%
  filter(SECTOR %in% c("PSACC" , "PSLEG")) %>%
  select(Year, REPORTER, Value, SECTOR)

# Merge dataframes together
df_obs_STRI <- rbind(df_arc_eng, df_acc_leg)

# Create Other Business Services Composite STRI
df_obs_STRI <- df_obs_STRI %>%
  group_by(Year, REPORTER) %>%
  summarise(Value = mean(Value)) %>%
  ungroup()
df_obs_STRI$SECTOR <- "163"

df_test <- df_obs_STRI %>% filter(Year == 2014)
```

Merge MFN STRI Sectors

Merge the MFN STRI sector dataframes into one dataframe

```r
# Merge dataframes together
df_MFN_STRI <- rbind(df_obs_STRI, df_info_STRI, df_transport_STRI, df_MFN_final)
df_MFN_STRI <- df_MFN_STRI %>%
  mutate(
  SECTOR = ifelse(SECTOR=="DS", 169, SECTOR),
  SECTOR = ifelse(SECTOR=="FSINS", 159, SECTOR),
  SECTOR = ifelse(SECTOR=="FSBNK", 160, SECTOR),
  SECTOR = ifelse(SECTOR=="CO", 158, SECTOR),
  )
```

Merge MFN and Intra-EEA STRI

Merge MFN and Intra-EEA STRI dataframes together

```r
df_STRI_final <- df_MFN_STRI %>%
  full_join(df_EEA_STRI, by = c("Year", "REPORTER", "SECTOR"))
```

Rename the columns and mutate to get ready to merge with DGD and ITPDE data

```r
df_STRI_final <- df_STRI_final %>%
  rename(importer_dynamic_code = REPORTER,
         year = Year, STRI_MFN = Value.x, industry_id = SECTOR, STRI_EEA = Value.y)

df_STRI_2021 <- df_STRI_final %>%
  filter(year == 2021)

df_STRI_final <- df_STRI_final %>% filter(year %in% 2014:2019)

df_STRI_final$industry_id <- as.integer(df_STRI_final$industry_id)

rm(df_acc_leg, df_arc_eng, df_audio_visual, df_freight, df_EEA_final, df_EEA_STRI,
   df_info_STRI, df_obs_STRI, df_transport_STRI, df_tc_cs, df_STRI, df_STRI_EEA,
   df_prod_USD_final, df_prod_USD, df_prod_NCU, df_prod_NCU_final, df_prod_fin,
   df_prod_info)
```

Merge the DGD and filtered ITPD-E data into one dataset

Merge the DGD and filtered ITPD-E data into one dataset and also create pair_id indicator.

Note again I created three different datasets: 1. Only the ITPD-E data (df_ITPDE) 2. ITPD-E data with all additional domestic trade data, that is, including data from ADB (df_ITPDE_new)

```r
df_EEA_membership <- c("AUT", "BEL", "CZE", "DNK", "FIN", "FRA", "DEU",
                       "GRC", "HUN", "ISL", "IRL", "ITA", "LUX", "NLD",
                       "NOR", "POL", "PRT", "SVK", "ESP", "SWE", "GBR",
                       "EST", "LVA", "LTU", "SVN")

df_1 <- df_ITPDE %>%
  left_join(df_dgd,  by = c("year","exporter_dynamic_code", "importer_dynamic_code")) %>%
select(exporter_dynamic_code, importer_dynamic_code, industry_id, year, trade, distance,
       contiguity, common_language, colony_ever, common_legal_origin,
       agree_pta_services)

df_2 <- df_ITPDE_new %>%
  left_join(df_dgd,  by = c("year","exporter_dynamic_code", "importer_dynamic_code")) %>%
  select(exporter_dynamic_code, importer_dynamic_code, industry_id, year, trade, distance,
       contiguity, common_language, colony_ever, common_legal_origin,
       agree_pta_services) %>%
  mutate(EEA_dummy = ifelse(exporter_dynamic_code %in% df_EEA_membership
                            & importer_dynamic_code %in% df_EEA_membership, 1, 0)) %>%
  mutate(agree_pta_services = ifelse(EEA_dummy==1 &
                                       agree_pta_services == 1,0,agree_pta_services))

#Create pair_id indicator

df_1 <- df_1 %>%
  mutate(pair_id = group_indices(.,
                                 pmax(exporter_dynamic_code,importer_dynamic_code),
pmin(exporter_dynamic_code, importer_dynamic_code)))

df_2 <- df_2 %>%
  mutate(pair_id = group_indices(.,
                                 pmax(exporter_dynamic_code,importer_dynamic_code),
pmin(exporter_dynamic_code, importer_dynamic_code)))
```

Merge the STRI Indexes into the Final Database

Merge the STRI indexes into the final database. If both countries are EEA then use the Intra-EEA STRI but if otherwise use the MFN STRI. I also then drop the Intra-EEA STRI and MFN STRI columns as they have been combined into one STRI column which is used in the model. I also delete all NA observations from the STRI column. This filters out any countries that do not have an STRI

```
df_1 <- df_1 %>%
  full_join(df_STRI_final,  by = c("year", "importer_dynamic_code", "industry_id")) %>%
  mutate(
  STRI = ifelse(exporter_dynamic_code %in% df_EEA_membership & importer_dynamic_code %in%
                  df_EEA_membership, STRI_EEA, STRI_MFN)) %>%
  select(-c(STRI_EEA,STRI_MFN)) %>%
  drop_na(STRI)

df_2 <- df_2 %>%
  full_join(df_STRI_final,  by = c("year", "importer_dynamic_code", "industry_id")) %>%
  mutate(
  STRI = ifelse(exporter_dynamic_code %in% df_EEA_membership & importer_dynamic_code %in%
                  df_EEA_membership, STRI_EEA, STRI_MFN)) %>%
  select(-c(STRI_EEA,STRI_MFN)) %>%
  drop_na(STRI)
```

Complete some final manipulations including: filtering data for 2014 to 2019, creating log dist, international border, STRI interaction, exp_year and imp_year variables. I also create pair_id_2 variable to calculate country-pair fixed effects

```
df_1 <- df_1 %>%
filter(year %in% 2014:2019) %>%
mutate(
exp_year = paste0(exporter_dynamic_code, year),
imp_year = paste0(importer_dynamic_code, year),
log_dist = log(distance),
inter = ifelse(importer_dynamic_code != exporter_dynamic_code, 1, 0),
STRI_int = STRI*inter,
pair_id_2 = ifelse(exporter_dynamic_code == importer_dynamic_code, "0-intra", pair_id)
)

df_2 <- df_2 %>% filter(!is.na(trade)) %>%
filter(year %in% 2014:2019) %>%
mutate(
exp_year = paste0(exporter_dynamic_code, year),
imp_year = paste0(importer_dynamic_code, year),
log_dist = log(distance),
log_trade = log(trade),
inter = ifelse(importer_dynamic_code != exporter_dynamic_code, 1, 0),
STRI_int = STRI*inter,
pair_id_2 = ifelse(exporter_dynamic_code == importer_dynamic_code, "0-intra", pair_id)
)
```

Further checks

```
df_test <- df_2 %>% filter(exporter_dynamic_code == "JPN" & industry_id == 163)
```

```
OECD <- c("AUS", "AUT", "BEL", "CAN", "CZE", "DNK", "FIN", "FRA", "DEU", "GRC",
                    "HUN", "ISL", "ITA", "JPN", "KOR", "LUX", "MEX", "NLD", "NZL", "NOR",
                    "POL", "PRT", "SVK", "ESP", "SWE", "CHE", "TUR", "GBR", "USA", "IRL",
                    "CHL", "EST", "ISR", "SVN", "COL", "LVA",
"CRI", "LTU")

df_3 <- df_2 %>%
filter(exporter_dynamic_code %in% OECD & importer_dynamic_code %in% OECD)
```

R Code Estimation Strategy

Model 1

Model 1 is using the ITPD-E data supplemented with OECD domestic consumption data

```
# Transport 156
# Construction 158
# Insurance and pension services 159
# Financial services 160
# Business services 163
# Information services 162
# Trade related services 169
model1 <- list(
"Transport" = df_2 %>% filter(industry_id == 156) %>% fepois(
 trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Distribution" = df_2 %>% filter(industry_id == 169) %>% fepois(
 trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Construction" = df_2 %>% filter(industry_id == 158) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Insurance" = df_2 %>% filter(industry_id == 159) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Finance" = df_2 %>% filter(industry_id == 160) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Business services" = df_2 %>% filter(industry_id == 163) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Info. services" = df_2 %>% filter(industry_id == 162) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE))
)

#change names of variables in table
cm <- c('STRI_int'    = 'Trade elasticity',
        'inter'    = 'Int. Border',
        'log_dist' = 'Log distance',
        'contiguity' = 'Contiguity',
        'colony_ever' = 'Colony ever',
        'common_language' = 'Common language',
        'agree_pta_services' = 'SPTA',
        'EEA_dummy' = 'EEA'
        )
```

```
rows <- tribble(
~term, ~"Transport", ~"Distribution", ~"Construction" , ~"Finance", ~"Insurance",
~"Business services", ~"Info. services",
'Exporter-year F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Importer-year F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes')

attr(rows, 'position') <- c(19, 20)
```

```
modelsummary(model1, add_rows = rows,
             stars = c('*' = 0.1, '**' = 0.05, '***' = 0.01),
             coef_map = cm,
             estimate = "{estimate}{stars}",
             gof_map = c("nobs", "adj.r.squared"),
             notes = list('Robust standard errors are clustered by exporter, importer
                          and year in parentheses below the parameter estimates.
                          Statistical significance is indicated as follows:
                          * (10%), ** (5%), and *** (1%).'),
             output = "latex")
```

#R Code Robustness Checks

Model 2 Country-Pair FE

Model 2 is runs the data with country-pair fixed effects

```
# Transport 156
# Construction 158
# Insurance and pension services 159
# Financial services 160
# Business services 163
# Information services 162
# Trade related services 169
model2 <- list(
"Transport" = df_2 %>% filter(industry_id == 156) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Distribution" = df_2 %>% filter(industry_id == 169) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Construction" = df_2 %>% filter(industry_id == 158) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Insurance" = df_2 %>% filter(industry_id == 159) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Finance" = df_2 %>% filter(industry_id == 160) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Business services" = df_2 %>% filter(industry_id == 163) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Info. services" = df_2 %>% filter(industry_id == 162) %>% fepois(
  trade ~ STRI_int + agree_pta_services |
    exp_year + imp_year + pair_id_2,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE))
)

rows <- tribble(
~term, ~"Transport", ~"Distribution", ~"Construction" , ~"Finance", ~"Insurance",
~"Business services", ~"Info. services",
'Exporter-year F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Importer-year F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Country-pair F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes')

attr(rows, 'position') <- c(8, 9, 10)

modelsummary(model2, coef_map = cm,
             add_rows = rows,
             stars = c('*' = 0.1, '**' = 0.05, '***' = 0.01),
             estimate = "{estimate}{stars}",
             gof_map = c("nobs", "adj.r.squared"),
   notes = list('Robust standard errors are clustered by exporter, importer and year in
   parentheses below the parameter estimates.
                Statistical significance is indicated as follows:
                * (10%), ** (5%), and *** (1%).
   SPTA indicates a bilateral services preferential trade agreement, excluding
   EEA membership, which is taken into account by the asymmetric pair fixed effect.'),
             output = "kableExtra")
```

Model 3 w/ only OECD countries

Model 3 is only with OECD countries

```
# Transport 156
# Construction 158
# Insurance and pension services 159
# Financial services 160
# Business services 163
# Information services 162
# Trade related services 169
```

```
model3 <- list(
"Transport" = df_3 %>% filter(industry_id == 156) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Distribution" = df_3 %>% filter(industry_id == 169) %>% fepois(
 trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Construction" = df_3 %>% filter(industry_id == 158) %>% fepois(
 trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Insurance" = df_3 %>% filter(industry_id == 159) %>% fepois(
   trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
     agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Finance" = df_3 %>% filter(industry_id == 160) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Business services" = df_3 %>% filter(industry_id == 163) %>% fepois(
   trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
     agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Info. services" = df_3 %>% filter(industry_id == 162) %>% fepois(
   trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
     agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE))
)

rows <- tribble(
~term, ~"Transport", ~"Distribution", ~"Construction" , ~"Finance", ~"Insurance",
~"Business services", ~"Info. services",
'Exporter-year F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Importer-year F.E.', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes')

attr(rows, 'position') <- c(19, 20)
```

```
modelsummary(model3, coef_map = cm,
             add_rows = rows,
             stars = c('*' = 0.1, '**' = 0.05, '***' = 0.01),
             estimate = "{estimate}{stars}",
             notes = list('Robust standard errors are clustered by exporter, importer and
                          year in parentheses below the parameter estimates.
                          Statistical significance is indicated as follows:
                          * (10%), ** (5%), and *** (1%).'),
             gof_map = c("nobs", "adj.r.squared"),
             output = "latex")
```

Model 4 without EEA Dummy

Model 4 is using the ITPD-E data supplemented with OECD domestic consumption data but with an additional EEA dummy

```
# Transport 156
# Construction 158
# Insurance and pension services 159
# Financial services 160
# Business services 163
# Information services 162
# Trade related services 169
model4 <- list(
"Transport" = df_2 %>% filter(industry_id == 156) %>% fepois(
 trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Distribution" = df_2 %>% filter(industry_id == 169) %>% fepois(
 trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Construction" = df_2 %>% filter(industry_id == 158) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Insurance" = df_2 %>% filter(industry_id == 159) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Finance" = df_2 %>% filter(industry_id == 160) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services  |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Business services" = df_2 %>% filter(industry_id == 163) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Info. services" = df_2 %>% filter(industry_id == 162) %>% fepois(
  trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE))
)
```

```
attr(rows, 'position') <- c(15, 16)
modelsummary(model4, coef_map = cm,
             add_rows = rows,
             stars = c('*' = 0.1, '**' = 0.05, '***' = 0.01),
             estimate = "{estimate}{stars}",
             notes = list('Robust standard errors are clustered by exporter, importer and
                          year in parentheses below the parameter estimates.
                          Statistical significance is indicated as follows:
                          * (10%), ** (5%), and *** (1%).'),
             gof_map = c("nobs", "adj.r.squared"),
             output = "latex")
```

Model 5 using OLS

Model 5 run using OLS

```
# Transport 156
# Construction 158
# Insurance and pension services 159
# Financial services 160
# Business services 163
# Information services 162
# Trade related services 169
model5 <- list(
"Transport" = df_2 %>% filter(industry_id == 156) %>% filter(trade > 0) %>% feols(
 log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services + EEA_dummy|
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Distribution" = df_2 %>% filter(industry_id == 169) %>% filter(trade > 0) %>% feols(
 log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
   agree_pta_services + EEA_dummy|
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Construction" = df_2 %>% filter(industry_id == 158) %>% filter(trade > 0) %>% feols(
  log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Insurance" = df_2 %>% filter(industry_id == 159) %>% filter(trade > 0) %>% feols(
  log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Finance" = df_2 %>% filter(industry_id == 160) %>% filter(trade > 0) %>% feols(
  log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy|
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Business services" = df_2 %>% filter(industry_id == 163) %>% filter(trade > 0) %>% feols(
  log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE)),
"Info. services" = df_2 %>% filter(industry_id == 162) %>% filter(trade > 0) %>% feols(
  log_trade ~ STRI_int + inter + log_dist + contiguity +  colony_ever + common_language +
    agree_pta_services + EEA_dummy |
    exp_year + imp_year,
  cluster = ~ pair_id,
  ssc = ssc(adj = FALSE))
)
```

```
attr(rows, 'position') <- c(19, 20)

modelsummary(model5, coef_map = cm,
             add_rows = rows,
             stars = c('*' = 0.1, '**' = 0.05, '***' = 0.01),
             notes = list('Robust standard errors are clustered by exporter, importer and
             year in parentheses below the parameter estimates.
                          Statistical significance is indicated as follows:
                          * (10%), ** (5%), and *** (1%).
                          SPTA indicates a bilateral services preferential trade
                          agreement, excluding EEA membership,
             which is taken into account by the asymmetric pair fixed effect.'),
             estimate = "{estimate}{stars}",
             gof_map = c("nobs", "adj.r.squared"),
             output = "latex")
```

Save Final Estimation Dataset

Save final estimation data (except for model 3 which uses df_3)

```
#write to file path
write.csv(df_2,"df_2.csv", row.names = FALSE)
```

Save Trade Elasticity From Model 1

Here I save the trade elasticity estimates from model 1 and also create Figure 3. df_plot is needed to calculate AVEs in the next step

```
df_plot<-modelplot(model1, coef_map = 'STRI_int', facet = TRUE, color = "#0072B2")+
    labs(x = 'Coefficients and 95% confidence intervals') +
    theme(strip.background = element_blank())
```

Calculate AVEs

Calculate AVEs following the formula from Benz (2017).

```
df_STRI_AVE <- df_STRI_2021
df_STRI_AVE$AVE_mfn = 0L
df_STRI_AVE$AVE_eea = 0L

df_est<-df_plot$data %>%
  select(model, estimate) %>%
  rename(industry_id = model, beta = estimate)

df_est$industry_id = c("156", "169", "158", "159", "160", "163", "162")
df_est$sigma <- c(3.83,3.86,4.01,3.06,2.78,3.63,3.59)

#MFN STRI to AVE
for (i in 1:nrow(df_STRI_AVE)){
  if (df_STRI_AVE$industry_id[i] == 156){
    beta = df_est$beta[1]
    sigma = df_est$sigma[1]

    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 169){
    beta = df_est$beta[2]
    sigma = df_est$sigma[2]
    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 158){
    beta = df_est$beta[3]
    sigma = df_est$sigma[3]
    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 159){
    beta = df_est$beta[4]
    sigma = df_est$sigma[4]
    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 160){
    beta = df_est$beta[5]
    sigma = df_est$sigma[5]
    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 163){
    beta = df_est$beta[6]
    sigma = df_est$sigma[6]
    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 162){
    beta = df_est$beta[7]
    sigma = df_est$sigma[7]
    df_STRI_AVE$AVE_mfn[i] = exp(-df_STRI_AVE$STRI_MFN[i]*beta/(sigma-1))-1
  }
}

#EEA STRI to AVE
for (i in 1:nrow(df_STRI_AVE)){
  if (df_STRI_AVE$industry_id[i] == 156){
    beta = df_est$beta[1]
    sigma = df_est$sigma[1]

    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 169){
    beta = df_est$beta[2]
    sigma = df_est$sigma[2]
    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 158){
    beta = df_est$beta[3]
    sigma = df_est$sigma[3]
    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 159){
    beta = df_est$beta[4]
    sigma = df_est$sigma[4]
    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 160){
    beta = df_est$beta[5]
    sigma = df_est$sigma[5]
    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 163){
    beta = df_est$beta[6]
    sigma = df_est$sigma[6]
    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
  else if (df_STRI_AVE$industry_id[i] == 162){
    beta = df_est$beta[7]
    sigma = df_est$sigma[7]
    df_STRI_AVE$AVE_eea[i] = exp(-df_STRI_AVE$STRI_EEA[i]*beta/(sigma-1))-1
  }
}

df_STRI_AVE <- df_STRI_AVE %>%
  mutate(AVE_eea = AVE_eea*100) %>%
  mutate(AVE_mfn = AVE_mfn*100)
```

Tables

AVE summary table

```
row.swapper = function(df1, df2, idx1, idx2){
  df1[idx1,] = df2[idx2,]
  return(df1)
}

df_199 <- df_STRI_AVE %>%
  select(industry_id, AVE_mfn, AVE_eea) %>%
  mutate(industry_id = as.factor(industry_id)) %>%
  group_by(industry_id) %>%
  mutate(
    mfnMed = median(AVE_mfn),
    mfnMea = mean(AVE_mfn),
    mfnMin = min(AVE_mfn),
    mfnMax = max(AVE_mfn),
  ) %>%
  select(-AVE_mfn) %>%
  na.omit() %>%
  mutate(
    eeaMed = median(AVE_eea),
    eeaMea = mean(AVE_eea),
    eeaMin = min(AVE_eea),
    eeaMax = max(AVE_eea)
  ) %>%
  select(-AVE_eea) %>%
  unique()

df_200 = df_199
df_200 = row.swapper(df_200, df_199, 1, 3)
df_200 = row.swapper(df_200, df_199, 2, 5)
df_200 = row.swapper(df_200, df_199, 3, 4)
df_200 = row.swapper(df_200, df_199, 4, 7)
df_200 = row.swapper(df_200, df_199, 5, 6)
df_200 = row.swapper(df_200, df_199, 6, 1)
df_200 = row.swapper(df_200, df_199, 7, 2)

#df_200<- paste0(as.matrix(df_200), '%')
#df_200  <- df_200 %>% mutate_if(is.numeric, round, digits = 1)
#df_200 <- lapply(df_200[2:9], paste0, '%')
```

```
test <- df_200 %>%
    knitr::kable("latex",
    col.names = c("Industry","Median","Mean","Min","Max","Median","Mean","Min","Max"),
    digits = c(1,1,1,1,1,1,1,1,1),
    align = "rcccccccc",
    vline = "",
    booktabs = T,
    linesep = ""
  ) %>%
  kable_styling(latex_options = c('striped')) %>%
  add_header_above(c(" " = 1, "MFN" = 4, "Intra-EEA" = 4)) %>%
   add_footnote("Note: Intra-EEA refers to services trade between two EEA members.
               MFN refers to all other trading relationships.")
```

Appendix A table

```
#of ilist<- as.list(unique(df_STRI_AVE$industry_id))

df_country <- df_STRI_AVE %>%
  select(importer_dynamic_code)

df_156 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 156) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_156 = AVE_mfn,
     ave_eea_156 = AVE_eea
  )

df_169 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 169) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_169 = AVE_mfn,
     ave_eea_169 = AVE_eea
  )

df_158 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 158) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_158 = AVE_mfn,
     ave_eea_158 = AVE_eea
  )

df_159 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 159) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_159 = AVE_mfn,
     ave_eea_159 = AVE_eea
  )

df_160 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 160) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_160 = AVE_mfn,
     ave_eea_160 = AVE_eea
  )

df_163 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 163) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_163 = AVE_mfn,
     ave_eea_163 = AVE_eea
  )

df_162 <- df_STRI_AVE %>%
  arrange(importer_dynamic_code) %>%
  filter(industry_id == 162) %>%
  select(AVE_mfn, AVE_eea) %>%
  rename(
     ave_mfn_162 = AVE_mfn,
     ave_eea_162 = AVE_eea
  )

ave_plot_1 <- cbind(df_country, df_156, df_169, df_158)
ave_plot_1 <- ave_plot_1[1:50,]
ave_plot_2 <- cbind(df_country, df_159, df_160, df_163, df_162)
ave_plot_2 <- ave_plot_2[1:50,]
#rm(df_160, df_169, df_156, df_162, df_163, df_159, df_158)

options(knitr.kable.NA = '')
opts_1 <-
table_appendixa <- ave_plot_1 %>%
  knitr::kable("latex",
    col.names = c("Country", "MFN","EEA","MFN","EEA","MFN","EEA"),
               digits = 1,
               align = "rcccccc",
               format.args = list(scientific = FALSE),
  ) %>%
  kable_styling(latex_options = c('striped')) %>%
  add_header_above(c(" " = 1, "Transport" = 2, "Distribution" = 2, "Construction" = 2))

opts_2 <-
table_appendixa <- ave_plot_2 %>%
  knitr::kable("latex",
    col.names = c("Country", "MFN","EEA","MFN","EEA","MFN","EEA","MFN","EEA"),
               digits = 1,
               align = "rcccccccc",
               format.args = list(scientific = FALSE),
  ) %>%
  kable_styling(latex_options = c('striped')) %>%
  add_header_above(c(" " = 1, "Insurance" = 2, "Finance" = 2, "Business" = 2,
                     "Info. services" = 2))
```

Elasticity of Substitution Table

```
names <- c("Transport","Distribution", "Construction" ,"Insurance", "Financial",
           "Business services", "Info. services")
df_eos <- data.frame(Sectors = names)
df_eos
df_eos <- df_eos %>%
  mutate(rouzet2017 = c(2.77, 5.40, 2.70, 2.20, 1.60, 2.20, 2.23),
         eggera = c(3.80, 3.17, 3.34, 4.18, 4.18, 4.02, 4.27),
         christen2019 = c(3.59, 3.00, 4.00, 2.59, 2.05, 3.77, 3.95),
         blank2018 = c(5.16, "", 6.00, 3.27, 3.27, 4.51, 3.92),
         simpavg = c(3.83, 3.86, 4.01, 3.06, 2.78, 3.63, 3.59)
  )

df_eos_table <- df_eos %>%
  knitr::kable("latex",
    col.names = c("Sectors", "autocite{rouzet2017}", "autocite{eggera}",
                  "autocite{christen2019}", "autocite{blank2018}", "Simple Avg."),
    align = "lccccc",
    escape = FALSE,
    vline = "",
    booktabs = TRUE,
    linesep = ""
  ) %>%
  column_spec(column = 2:6, width = "3cm") %>%
  kable_styling(latex_options = c("striped")) %>%
  add_footnote("The first column is based on firm-level data of profit margins from the
               United Kingdom and Finland. In the second column, elasticities are based on
               sectoral trade and unit labour costs from 41 countries in the WIOD database
               and the elasticity for distribution services is a simple average of
               retail (2.55) and wholesale trade (3.78). In the third column, elasticities
               are based on firm-level data on profit margins from Austria and the
               elasticity of distribution services is from only wholesale trade.
               In the fourth column, elasticities are based on firm level data on
               profit-margins from Germany.")
```

ITPD-E Summary Table

```
df_9 <- df_2 %>% filter(!is.na(trade)) %>%
  group_by(industry_id) %>% mutate(zeros=sum(trade==0), obs = length(trade)) %>%
  summarise(trade_obs = n(),
            trade_median=median(trade),trade_mean=mean(trade),
            trade_min=min(trade),trade_max=max(trade),
            trade_sd = sd(trade),
            zero = zeros/obs*100) %>%
  ungroup() %>% unique()

# Rename industry_id's

df_9<- df_9 %>%
  mutate(
    industry_id =
      ifelse(industry_id==156, "Transport", industry_id),
    industry_id =
      ifelse(industry_id==158, "Construction", industry_id),
    industry_id =
      ifelse(industry_id==169, "Distribution", industry_id),
    industry_id =
      ifelse(industry_id==159, "Insurance", industry_id),
    industry_id =
      ifelse(industry_id==160, "Finance", industry_id),
    industry_id =
      ifelse(industry_id==163, "Business services", industry_id),
    industry_id =
      ifelse(industry_id==162, "Info. services", industry_id))

# Reorder

df_9$industry_id <- factor(df_9$industry_id,      # Reordering group factor levels
                         levels = c("Transport", "Construction", "Distribution",
                                    "Insurance","Finance",
                                    "Business services", "Info. services"))

trade_table <- df_9 %>%
  knitr::kable("latex",
    col.names = c("Sector", "Obs", "Median","Mean","Min","Max","Std. Dev.", "% zero's"),
    align = "rccccccc",
    digits = c(2,2,2,2,2,2,2,2),
    vline = "",
    booktabs = T,
    linesep = ""
  ) %>%
  kable_styling(latex_options = c("striped"))
```

Figures

Correlogram

```
#correlation plot
df_corr <- df_2 %>%
  select(STRI_int, trade, log_dist, contiguity, common_language, colony_ever,
         agree_pta_services, EEA_dummy) %>%
  cor(use = "pairwise.complete.obs")

colnames(df_corr) <- c('Trade elasticity', 'Trade', 'Log distance', 'Contiguity',
                       'Colony ever', 'Common language', 'SPTA', 'EEA')
rownames(df_corr) <- c('Trade elasticity', 'Trade', 'Log distance', 'Contiguity',
                       'Colony ever', 'Common language', 'SPTA', 'EEA')
corrplot(df_corr, method = "number", type = "upper", tl.col = "black", tl.srt = 40,
         tl.cex = 0.8)

#table for STRI scores
#datasummary(Factor(industry_id)*STRI  ~  Median + Mean + SD +  Min + Max, data = df_2)
```

Figure 1

Figure 1 - Global services exports

Create theme function

```
my_style <- function() {
  font <- "Times New Roman"

  ggplot2::theme(
    #Text format:
    #This sets the font, size, type and colour of text for the chart's title
    plot.title = ggplot2::element_text(family=font,
                                       size=20,
                                       face="bold",
                                       color="#0072B2"),
    #This sets the font, size, type and colour of text for the chart's subtitle,
    #as well as setting a margin between the title and the subtitle
    plot.subtitle = ggplot2::element_text(family=font,
                                          size=16,
                                          margin=ggplot2::margin(3,0,3,0)),
    plot.caption = ggplot2::element_text(family=font,
                                         size=10,
                                         hjust = 0.0
    ),
    #This leaves the caption text element empty, because it is set elsewhere in the
    #finalise plot function

    legend.position = "none",
    #Axis format
    #This sets the text font, size and colour for the axis test, as well as setting the
    #margins and removes lines and ticks.
    axis.title = ggplot2::element_blank(),
    axis.text = ggplot2::element_text(family=font,
                                      size=16,
                                      color="#000000"),
    axis.text.x = ggplot2::element_text(margin=ggplot2::margin(5, b = 10)),
    axis.ticks.y = ggplot2::element_blank(),
    axis.line.y = ggplot2::element_blank(),
    axis.line.x.bottom = ggplot2::element_line(color="#000000",size = 0.5, linetype = 1),
    axis.ticks.x.bottom = ggplot2::element_line(color="#000000",size = 0.5),

    #Grid lines
    #This removes all minor gridlines and adds major y gridlines.
    panel.grid.minor = ggplot2::element_blank(),
    panel.grid.major.y = ggplot2::element_line(color="#dcdcdc"),
    panel.grid.major.x = ggplot2::element_blank(),

    #Blank background
    #This sets the panel background as blank, removing the standard grey ggplot background
    #colour from the plot
    panel.background = ggplot2::element_blank(),

    #Strip background
    strip.background = ggplot2::element_rect(fill="white"),
    strip.text = ggplot2::element_text(size  = 16,  hjust = 0)
  )
}

#' @rdname my_axes
#' @export

my_y_continuous <- function(expand_bottom = 0, expand_top = 0.15, ...) {
  scale_y_continuous(expand = ggplot2::expansion(mult = c(expand_bottom,
                                                          expand_top)),
                     ...)
}


#' @rdname my_axes
#' @export

scale_y_continuous_my <- function(expand_bottom = 0,
                                    expand_top = 0.15,
                                    ...) {

  my_y_continuous(expand_bottom = expand_bottom,
                    expand_top = expand_top,
                    ...)
}

#' @rdname my_axes
```

```r
#' @importFrom ggplot2 scale_y_continuous expand_scale
#' @export

my_x_continuous <- function(expand_left = 0,
                              expand_right = 0.07,
                              ...) {

  scale_x_continuous(expand = ggplot2::expansion(mult = c(expand_left,
                                                          expand_right)),
                     ...)

}

#' @rdname my_axes
#' @export

scale_x_continuous_my <- function(expand_left = 0,
                                    expand_right = 0.07,
                                    ...) {

  my_x_continuous(expand_left = expand_left,
                    expand_right = expand_right,
                    ...)
}
```

Save plot function

```r
#save plot function
save_plot <- function (plot_grid, width, height, save_filepath) {
  grid::grid.draw(plot_grid)
  #save it
  ggplot2::ggsave(filename = save_filepath,
                  plot=plot_grid, width=(width/72), height=(height/72),  bg="white")
}

#Left align text
left_align <- function(plot_name, pieces){
  grob <- ggplot2::ggplotGrob(plot_name)
  n <- length(pieces)
  grob$layout$l[grob$layout$name %in% pieces] <- 2
  return(grob)
}

finalise_plot <- function(plot_name,
                          save_filepath=file.path(Sys.getenv("TMPDIR"), "tmp-nc.png"),
                          width_pixels=640,
                          height_pixels=450) {

  #Draw your left-aligned grid
  plot_left_aligned <- left_align(plot_name, c("subtitle", "title", "caption"))
  plot_grid <- ggpubr::ggarrange(plot_left_aligned,
                                 ncol = 1, nrow = 1,
                                  heights = 1)
  ## print(paste("Saving to", save_filepath))
  save_plot(plot_grid, width_pixels, height_pixels, save_filepath)
  ## Return (invisibly) a copy of the graph. Can be assigned to a
  ## variable or silently ignored.
  invisible(plot_grid)
}
```

Plot Figure 1

```
df_wb = wb_data(indicator='BX.GSR.NFSV.CD', country='WLD', start=1975, end=2021)

figure1 <- ggplot(df_wb, aes(x=date, y=BX.GSR.NFSV.CD/1000000000000)) +
  geom_line(colour = "#0072B2")+
  my_style()+
  my_y_continuous()+
  labs(subtitle = "Annual services exports (US$tn)",
       caption = "Note: Current $US from Balance of Payments Statistics
Source: World Bank")
```

Save Figure 1

```
ggsave("figure1.png", width = 6.4,
              height = 4.5)
# Run code below for clearer figure
#finalise_plot(plot_name = figure1,
              #add file path
              #save_filepath = "figure1.png",
              #width_pixels = 640,
              #height_pixels = 450)
```

Figure 2: Distribution of STRI Scores

```
# Change industry_id to names
theme_STRI <- function(...) {
  theme_minimal() +
    theme(
    axis.line.y = element_blank(),
    axis.line.x.bottom = element_line(color="#000000",size = 0.5, linetype = 1),
    axis.ticks.x.bottom = element_line(color="#000000",size = 0.5),
    #Grid lines
    #This removes all minor gridlines and adds major y gridlines.
      panel.grid.minor = element_blank(),
      panel.grid.major.y =element_line(color="#dcdcdc"),
      panel.grid.major.x = element_blank(),
      axis.line = element_blank(),
      axis.text.x = element_text(family="Times New Roman",color="#000000", size = 14,
                                 margin=ggplot2::margin(5, b = 10)),
      axis.text.y = element_text(color="#000000", size = 14, family="Times New Roman"),
      axis.ticks = element_blank(),
      axis.ticks.y = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      plot.title = element_text(size = 16, face = "bold",
                                hjust=0.5,family="Times New Roman"),
      plot.subtitle = element_text(size = 20,hjust=0.5,
                                   family="Times New Roman"),
      plot.caption = element_text(size=12,
                                  family="Times New Roman"),
      strip.text.x = element_text(size=14,hjust=0.1,vjust=0, face = "bold",
                                  family="Times New Roman"),
      strip.text = element_text(size=16),
      plot.background = element_rect(fill = "white", color = NA),
      plot.margin = margin(0.5, 1, 0.5, 0.5, "cm"),
      panel.background = element_rect(fill = "white", color = NA),
      panel.border = element_blank(),
      legend.position = 'top',
      legend.background = element_rect(fill = "white", color = NA),
      legend.title = element_text(family="Times New Roman",color="#000000", size = 16,
                                  hjust=0.5),
      legend.title.align = 0.5,
      legend.text = element_text(family="Times New Roman",color="#000000", size = 16),
      legend.key = element_rect()
    )}

df_STRI_fig  <- df_2 %>%
  mutate(
    industry_id =
      ifelse(industry_id==156, "Transport", industry_id),
    industry_id =
      ifelse(industry_id==158, "Construction", industry_id),
    industry_id =
      ifelse(industry_id==169, "Distribution", industry_id),
    industry_id =
      ifelse(industry_id==159, "Insurance", industry_id),
    industry_id =
      ifelse(industry_id==160, "Finance", industry_id),
    industry_id =
      ifelse(industry_id==163, "Business services", industry_id),
    industry_id =
      ifelse(industry_id==162, "Info. services", industry_id))

# Reorder
df_STRI_fig$industry_id <- factor(df_STRI_fig$industry_id,
                                  # Reordering group factor levels
                         levels = c("Transport", "Construction", "Distribution",
                                    "Insurance","Finance",
                                    "Business services", "Info. services"))

# Create plot
```

```
plot_STRI <- df_STRI_fig %>%
  ggplot(aes(x = year, y = STRI)) + geom_fan() +
  geom_interval() +
  scale_linetype_manual(values=c("solid", "dashed", "dotted"))+
  scale_fill_gradient(low="#0072B2", high="#f0f0f0") +
  labs(#title = 'Service AVEs in the EEA by sector, 2021',
       caption='Note: Changes in colour represent intervals covering an increasing
       proportion of total density.
       Source: OECD and own estimates',
       x = NULL,
       y = NULL)+
  theme_STRI()+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  guides(fill = guide_colourbar(direction = 'horizontal',
                                title='Proportion of STRI density',
                                title.position='top',
                                title.hjust=0.5,
                                ticks.colour='#f5f5f2',
                                ticks.linewidth=2,
                                barwidth = 30,
                                barheight = 0.7))+
    guides(linetype = guide_legend(direction = 'horizontal',
                                title='Interval', ##rename default legend
                                title.position='top',
                                title.hjust=0.5,
                                ticks.colour='#f5f5f2',
                                ticks.linewidth=2,
                                barwidth = 15,
                                barheight = 0.7))

design <- c(
"
AABBCC
DDEEFF
##GG##
"
)
plot_STRI + ggh4x::facet_manual(~industry_id, design = design) +
  theme(panel.spacing.x = unit(2.5, "lines"))

# Save

ggsave("STRI_figure.png", width = 10,
              height = 14.3)
```

Figure 4: AVE MFN Results

Note: Figure 3 in made in df_plot and is used to save the trade elasticity from Model 1.

Obtain Data

```
# Libraries

# create a dataset
df_mfn <- df_STRI_AVE %>% select(importer_dynamic_code, industry_id, AVE_mfn) %>%
    mutate(
    industry_id =
      ifelse(industry_id==156, "Transport", industry_id),
    industry_id =
      ifelse(industry_id==158, "Construction", industry_id),
    industry_id =
      ifelse(industry_id==169, "Distribution", industry_id),
    industry_id =
      ifelse(industry_id==159, "Insurance", industry_id),
    industry_id =
      ifelse(industry_id==160, "Finance", industry_id),
    industry_id =
      ifelse(industry_id==163, "Business services", industry_id),
    industry_id =
      ifelse(industry_id==162, "Info. services", industry_id))

df_mfn$industry_id <- factor(df_mfn$industry_id,      # Reordering group factor levels
                         levels = c("Transport", "Construction", "Distribution",
                                    "Insurance","Finance",
                                    "Business services", "Info. services"))
```

Only Box Plots

```
# Plot
plot1 <- df_mfn %>%
  filter(industry_id !="Finance") %>%
  filter(industry_id != "Distribution")  %>%
  ggplot(aes(x=industry_id, y=AVE_mfn, fill = industry_id)) +
    geom_boxplot() +
    coord_flip() +
    scale_fill_viridis(discrete = TRUE, alpha=0.6) +
    geom_jitter(color="black", size=0.4, alpha=0.9) +
    theme_ipsum() +
    theme(
      legend.position="none",
      plot.title = element_text(size=11)
    ) +
    ggtitle("") +
    ylab("AVE (%)")+
    xlab("")

plot2 <- df_mfn %>%
  filter(industry_id %in% c("Finance", "Distribution")) %>%
  ggplot(aes(x=industry_id, y=AVE_mfn, fill = industry_id)) +
    geom_boxplot() +
    coord_flip() +
    scale_fill_viridis(discrete = TRUE, alpha=0.6) +
    geom_jitter(color="black", size=0.4, alpha=0.9) +
    theme_ipsum() +
    theme(
      legend.position="none",
      plot.title = element_text(size=11)
    ) +
    ggtitle("") +
    ylab("AVE (%)")+
    xlab("")
```

Boxplots inside Violin plot

```
# Plot
plot3 <-  df_mfn %>%
  filter(industry_id !="Finance") %>%
  filter(industry_id != "Distribution")  %>%
  ggplot( aes(x=industry_id, y=AVE_mfn, fill=industry_id)) +
    geom_violin(width=1.4) +
    geom_boxplot(width=0.1, color="black", alpha=0.8) +
    coord_flip() +
    scale_fill_viridis(discrete = TRUE, alpha = 0.6) +
    theme_minimal() +
    theme(text=element_text(family="Times New Roman", color = "#000000"),
      legend.position="none",
      axis.text.x  = element_text(family="Times New Roman",
                                      size=10, colour = "#000000"),
      axis.text.y  = element_text(family="Times New Roman",
                                      size=10, colour = "#000000"),
      axis.title.x = element_text(family="Times New Roman",
                                      size=10, colour = "#000000")
    )  +
    xlab("") +
    ylab("AVE (%)")
plot3

ggsave("serviceAVEmfn1.png", width = 10,
              height = 7)

plot4 <-  df_mfn %>%
  filter(industry_id %in% c("Finance", "Distribution")) %>%
  ggplot( aes(x=industry_id, y=AVE_mfn, fill=industry_id)) +
    geom_violin(width=1.4) +
    geom_boxplot(width=0.1, color="black", alpha=0.8) +
    coord_flip() +
    scale_fill_viridis(discrete = TRUE, alpha = 0.6) +
    theme_minimal() +
    theme(text=element_text(family="Times New Roman", color = "#000000"),
      legend.position="none",
      axis.text.x  = element_text(family="Times New Roman",
                                      size=10, colour = "#000000"),
      axis.text.y  = element_text(family="Times New Roman",
                                  size=10, colour = "#000000"),
      axis.title.x = element_text(family="Times New Roman",
                                  size=10, colour = "#000000")
    )  +
    xlab("") +
    ylab("AVE (%)")
plot4

ggsave("serviceAVEmfn2.png", width = 10,
              height = 2.8)
```

Figure 5: AVE Intra-EEA Map

Download and Organise Data

```
# Obtain intra-eea AVE data
df_map <- df_STRI_AVE %>% select(importer_dynamic_code, industry_id, AVE_eea)
# Get map data
df_world <- map_data('world')
df_world$code <- countrycode(sourcevar = df_world[["region"]],
                             origin = 'country.name', destination = 'iso3c')

# Define EEA membership
df_EEA_membership <- c("AUT", "BEL", "CZE", "DNK", "FIN", "FRA", "DEU",
                       "GRC", "HUN", "ISL", "IRL", "ITA", "LUX", "NLD",
                       "NOR", "POL", "PRT", "SVK", "ESP", "SWE", "GBR",
                       "EST", "LVA", "LTU", "SVN")

# Join data together
df_99 <- left_join(df_map, df_world, by = c('importer_dynamic_code' = 'code'))
```

Coordinate Data

Organise coordinate data and filter for EEA membership

```
# Filter for EEA membership
df_99  <- df_99 %>% filter(importer_dynamic_code %in% df_EEA_membership)

# Organise coordinate data
lab.data <- df_99 %>%
group_by(importer_dynamic_code) %>%
summarise(long = mean(long), lat = mean(lat))

# Test plot
ggplot(df_99) +
geom_polygon(aes(x = long, y = lat, group = group,fill = AVE_eea),color='gray')+
theme_void()
```

Plot Figure 4

```r
# Create theme function

theme_map <- function(...) {
  theme_minimal() +
    theme(
      axis.line = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.ticks = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      plot.title = element_text(size = 20, face = "bold",
                                hjust=0.5,family="Times New Roman",color="gray35"),
      plot.subtitle = element_text(size = 14,hjust=0.5,
                                   family="Times New Roman",color="gray35"),
      plot.caption = element_text(size=10,
                                  family="Times New Roman",color="gray35"),
      strip.text.x = element_text(size=14,hjust=0.1,vjust=0, face = "bold",
                                  family="Times New Roman",color="gray35"),
      plot.background = element_rect(fill = "white", color = NA),
      plot.margin = margin(0.8, 0.5, 0.5, 0.5, "cm"),
      panel.background = element_rect(fill = "white", color = NA),
      panel.border = element_blank(),
      legend.position = 'bottom',
      legend.background = element_rect(fill = "white", color = NA),
      legend.title = element_text(family="Times New Roman",color="gray35"),
      legend.text = element_text(family="Times New Roman",color="gray35"),
      legend.key = element_rect()
    )}

# Change industry id's to actual names

df_99 <- df_99 %>%
  mutate(
    industry_id =
      ifelse(industry_id==156, "Transport", industry_id),
    industry_id =
      ifelse(industry_id==158, "Construction", industry_id),
    industry_id =
      ifelse(industry_id==169, "Distribution", industry_id),
    industry_id =
      ifelse(industry_id==159, "Insurance", industry_id),
    industry_id =
      ifelse(industry_id==160, "Finance", industry_id),
    industry_id =
      ifelse(industry_id==163, "Business services", industry_id),
    industry_id =
      ifelse(industry_id==162, "Info. services", industry_id))
```

```
# Reorder

df_99$industry_id <- factor(df_99$industry_id,      # Reordering group factor levels
                         levels = c("Transport", "Construction", "Distribution",
                                    "Insurance","Finance",
                                    "Business services", "Info. services"))

# Create base for plot

base <- ggplot(df_99) +
  geom_polygon(aes(x = long, y = lat, group = group, fill = AVE_eea),color='gray') +
  geom_text(data = lab.data, aes(label = importer_dynamic_code,x = long, y = lat),
            size = 3, hjust = 0.5)+
  labs(#title = 'Service AVEs in the EEA by sector, 2021',
       caption='Source: Own estimates',
       x = NULL,
       y = NULL)+
  #facet_wrap(~industry_id, ncol=3)+
  scale_fill_viridis(alpha=0.6,na.value='#f5f5f2', direction = -1)+
  theme_map()+
  guides(fill = guide_colourbar(direction = 'horizontal',
                                title='AVE (%)', ##rename default legend
                                title.position='top',
                                title.hjust=0.5,
                                ticks.colour='#f5f5f2',
                                ticks.linewidth=2,
                                barwidth = 30,
                                barheight = 0.5))

# Make last row appear in the middle

design <- c(
"
AABBCC
DDEEFF
##GG##
"
)
base + ggh4x::facet_manual(~industry_id, design = design)

# Save

ggsave("serviceAVEeea.png", width = 10,
              height = 14.3)
```