PSY6122 - Travel from the UK data visualisation

2023-05-15

Background

Travel is an important activity that people in the United Kingdom (UK) participate in for leasure, business and visiting family. These individual travel choices can have implications for the economy and environment making understanding any trends a useful exercise. Understanding these the trends of the past 10 years in particular would be important to understand the implications of a number of recent events that may influence destination choices (austerity measures, the UK exiting the European Union and the Coronavirus pandemic).

Research question

This visualisation aims to show whether there has been a **change in the main country travelled to from the UK for a visit over time**.

Data Origins

The data was provided by the UK's Office for National Statistics and accessed in May 2023 at their Travel Trends Dataset Pages. The only available data was between 2009 and 2021. The specific files used were:

- 2017 to 2021 edition of the data set and
- 2009 to 2019 edition of the data set.

I have used the "Number of visits to specified countries: by main country visited and nationality" as this allowed a more nuanced view of travel which a region level may not pick up.

Data collection

This data provides estimated visits based on the International Passenger Survey (IPS). The IPS uses interviews at all major airports, sea routes and Eurostar/ Eurotunnel terminals with approximately 250,000 people (although up to 800,000 interviews take place). However in 2021 the numbers taking part in the IPS were much lower. There is no further information on the sampling methods however at the scale and coverage these are trustworthy estimates. Additionally no other data sources exist at this level of detail.

A new method for calculating the 2021 estimates were introduced which supersedes the previous years. The estimates that overlap are the same in the 2009 to 2019 datasets so I have continued with the merge.

Raw data

Below is the top of the raw dataset for the years 2017 to 2021.

```
data1721 = (here("raw_data","section3ukresidentsvisitsabroad2017to2021.xlsx"))
#data for 2017 to 2021
raw<- read_excel(data1721,
                    sheet="3.06",
                    skip=10,
                    n_max=68)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...8`
```

```
head(raw)
```

```
## # A tibble: 6 x 9
##   ...1          ...2  `2017` `2018` `2019` `2020`            `2021` ...8  `(%)`
##   <chr>         <lgl>  <dbl>  <dbl>  <dbl> <chr>              <dbl> <lgl> <dbl>
## 1 <NA>          NA        NA     NA     NA  <NA>                NA   NA     NA
## 2 <NA>          NA        NA     NA     NA  <NA>                NA   NA     NA
## 3 Canada        NA       684.   779.   757. -                  120.  NA    -35.2
## 4 USA           NA      4439.  4943.  4805. -                  502.  NA    -42.0
## 5 North America NA      5123.  5722.  5562. 1056.024048354489  623.  NA    -41.0
## 6 Austria       NA       700.   695.   730. -                   88.0 NA    -40.5
```

Data definitions

A full codebook describing the variables from the original dataset and any calculated values are available in the github pages for this project.

The variables taken from the raw data that were not removed are:

variable	description	other_notes
country (1 in the raw data)	main destination travelled to from the UK	Source ONS raw data apart from the dfColour
2009...2021 (each years individual variable)	number of visits in thousands (000s)	This includes business, leasure and family travel. In dfv2 years correspond to ranks

Considerations for visualisation The data for 2021 is more limited

than previous years of the IPS. The smaller number of travellers and issues collecting the data at key terminals led to an overall smaller sample of passengers in the IPS. This means that there is more error in the estimation than other years (excluding 2020 where no country level data is available). Countries are consistent every year but if this changes then some of the data importing and colours will need changed.

Data preperation

This section covers the different processes that this data has been through prior to visualisation.

Libraries and RENV

The following packages were used in this project:

```
renv::restore()
#Libraries
library(here)#Here for loading in the data
library(tidyverse)#Tidyverse for data management
library(ggplot2) #chart making
library(readxl)#Reading in Excel sheets as that is how the data is stored
library(plotly)#Making scroll over charts
library(htmlwidgets)#To save interactive chart
```

To support future replication the this R Project used RENV (version:0.17.3) with package versions available in the following file location

Importing the data

Creating cleaning variables

As an initial step I created a set of values that supported the importation and data cleaning processes. In theory these will allow for a more recent dataset to also be used by changing the input file and changing the years that we are keeping.

```
# 2. Load the data in

#Variables required for data cleaning of the ONS files

ONSList <- 68 #The length of the ONS data table, these are consistent

yearsdf1721 <- c("2017","2018","2019",
                 "2020","2021") # These are the years that we will want to take from this data frame

yearsdf0919 <- c("2009","2010","2011",
                 "2012","2013","2014",
                 "2015","2016") #Years wanted, always choose most recent

summaries <- list('Total World',
                  'Other Countries',"Europe",
                  '- of which EU',
                  '- of which EU Oth',
                  '- of which EU15',
                  "North America") #summary values used in the dataset

# Raw data locations
data1721 = (here("raw_data","section3ukresidentsvisitsabroad2017to2021.xlsx")) #data for 2017 to 2021
data0919 = (here ("raw_data","section3ukresidentsvisitsabroad2009to2019.xlsx")) #data for 2009 to 2017
```

Importing the data

The following code imports the data into R. It then renames the columns to match those in the original ONS data file and removes the rows that are blank.

```
# Load in latest dataset 2017-2021
df1721<- read_excel(data1721,
                    sheet="3.06",
                    skip=10,
                    n_max=ONSList)

##rename columns 2017-2021
df1721_n <- tibble(x = 1:9, y = c("country", "coltoremove", "2017","2018",
                                  "2019","2020","2021","blankroremove",
                                  "avgrowth1519"))#list of column names

names(df1721) <- df1721_n %>% select(y) %>% pull()#function to change the names

##Remove NAs in rows and select years only 2017 to 2021
df1721 <- df1721 %>%
          select("country",all_of(yearsdf1721)) %>% #Select correct years
          drop_na("country")# remove blank rows

## Read the excel file for 2009 to 2019
df0919 <- read_excel(data0919,
                     sheet="3.10",
                     skip=10,
                     n_max=ONSList)

###rename columns 2009 - 2019
df0919_n <- tibble(x=1:19, y= c("country",
                                "coltoremove",
                                "2009","2010","2011",
                                "2012","2013","2014",
                                "2015","2016","2017",
                                "2018","2019",
                                "blank1",
                                "change1819",
                                "blank2",
                                "growth1819",
                                "blank3",
                                "Avgrowth1519"))#list of column names

names(df0919) <- df0919_n %>% select(y) %>% pull() #function to change the names

###Remove NAs in rows and select years only to 2017
df0919 <- df0919 %>%
          select("country",all_of(yearsdf0919)) %>% #Select correct years
          drop_na("country")# remove blank rows
```

Merge the files

Once both data sets were in the same shape so that I could then merge into a data set that has all available years. I then removed the summary variables.

```
#Merge the datasets
df0921 <- left_join(df0919, df1721,by="country")
```

```
##Remove summaries in the dataset
df0921 <-df0921 %>%
         filter((!country %in% summaries))
```

Use of ranks

As 58 countries is tricky to visualise I decided to choose a suitable cut off based on ranks for the visualisation. These were added here using the following code.I decided on 2021 to be the key year for this cut off because it is the most recent and presents a "now" vs the "past" view that is more intuitive.

```
## Add ranking variable for 2021 (this will be used to decide on the cases kept)
df0921 <- df0921 %>%
           mutate(rank_cut = rank(-`2021`, ties.method = "average"))

#save combined data set
comData_n = paste(here("created_data"), "/travel0921.csv", sep = "")
write.csv(df0921, comData_n)   #Write data to CSV
```

```
#check data
head(df0921)
```

```
## # A tibble: 6 x 15
##   country  `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
##   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Canada     517.   498.   541.   471.   463.   552.   528.   658.   684.   779.
## 2 USA       3538.  3922.  4071.  3744.  3831.  4197.  4398.  4382.  4439.  4943.
## 3 Austria    760.   704.   624.   620.   714.   630.   666.   909.   700.   695.
## 4 Belgium   1541.  1660.  1798.  2074.  2013.  2189.  2047.  1897.  2063.  2087.
## 5 Bulgaria   268.   254.   325.   337.   325.   378.   343.   477.   596.   691.
## 6 Czech R~   469.   423.   453.   414.   422.   378.   512.   600.   687.   625.
## # i 4 more variables: `2019` <dbl>, `2020` <chr>, `2021` <dbl>, rank_cut <dbl>
```

Visualising the data

After reviewing the data it was clear that there were some countries with very high number of visits and most that were much lower. This made a line plot difficult to interpret and didn't show if there were any overall trends of interest - the message was lost in noise. To account for this I decided to use just 2009 and 2021 in my visualisations as they are the earliest and latest data available. Additionally any visualisation would be a subset of the countries as 58 made it hard to interpret or pick out countries of interest.

Visualisation 1 - Dumbell plot

With visualisation 1 I wanted to use the raw number of visits to show the change in travel. I decided on a dumbell plot because I felt that it showed the difference between both 2009 and 2021 and the countries well. It is also less cluttered than a traditional clustered bar chart.

Data changes

Initially I created variables to select the years and the number of countries I wanted included. This was to allow for any quick changes based on how the final graph was looking. After this the data was then moved into a long format with the year column name becoming the year variable. The number of visits for each year is included as visits.

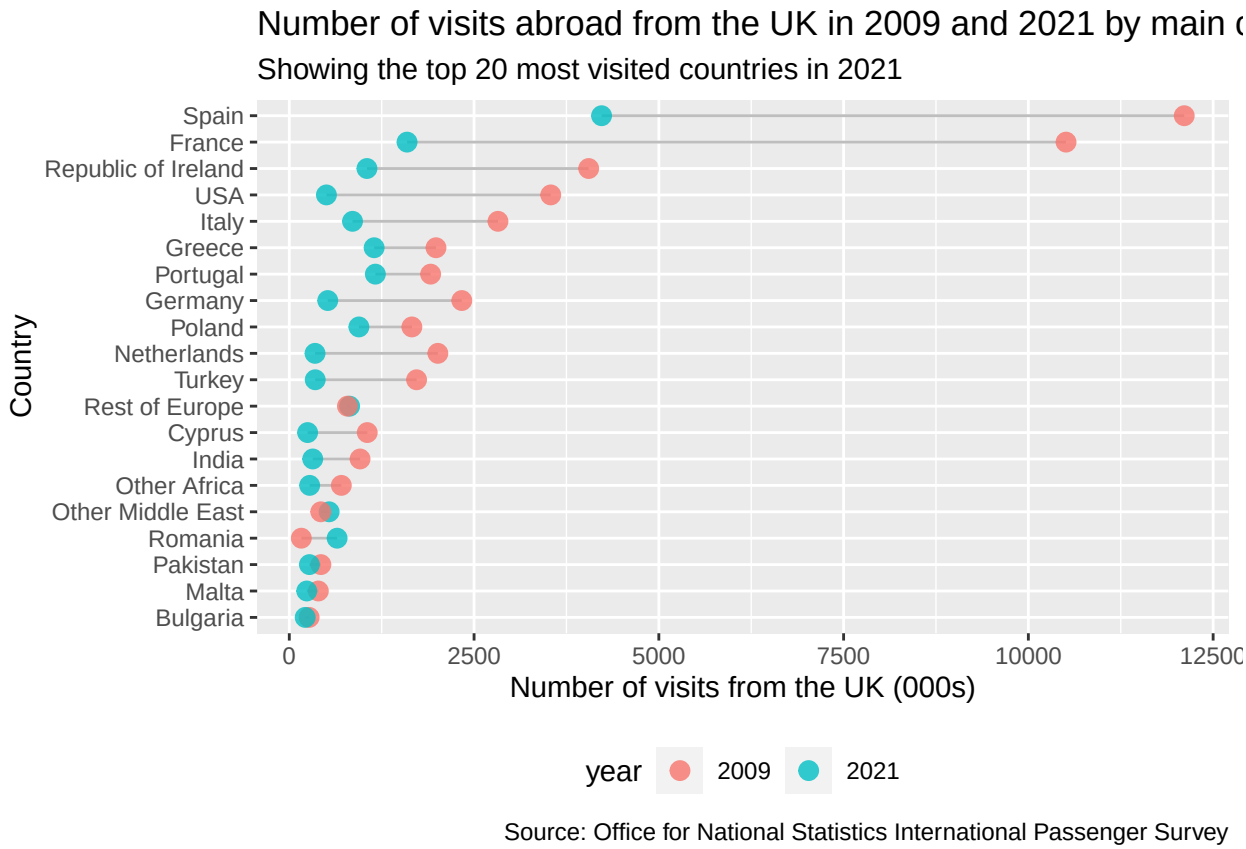
```
#3.1. Create variables to allow us to choose the number of 2021 ranks

cutrank1 <- 20 #number of countries that we want included in the chart
yrs_inc1 <- c(2009, 2021) # years that we want to include (can be any year from 2009-2021 apart from 20

#3.2 Create summary DF based on the number of countries

dfv1 <- df0921 %>%
        filter (rank_cut<=cutrank1)

#3.3. Convert to long format

longdfv1 <- dfv1 %>% select("country",
                            "2009", "2010",
                            "2011", "2012",
                            "2013", "2014",
                            "2015", "2016",
                            "2017", "2018",
                            "2019", "2021") %>% #2020 dropped due to no data
                     pivot_longer(!country,
                                  names_to = "year",
                                  names_transform = list(year = as.factor), #Changed to factor to allo
                                  values_to = "visits") %>%
                     arrange(desc(visits)) %>% #order by the most visited to the least
                     filter(year %in% yrs_inc1) #removes years that we are not interested in as define

head(longdfv1)
```

```
## # A tibble: 6 x 3
##   country             year  visits
##   <chr>               <fct>  <dbl>
## 1 Spain               2009  12109.
## 2 France              2009  10510.
## 3 Spain               2021   4225.
## 4 Republic of Ireland 2009   4051.
## 5 USA                 2009   3538.
## 6 Italy               2009   2824.
```

Visualisation

To create the dumbell plot I added the visits data to geom_point and then added a line to connect the 2 dates together. As some data points overlap I ensured that these points were opaque. Additionally I created a dynamic subtitle in case the years chosen changes in the future (e.g. if we decide that the 2019 pre-covid year is a more appropriate comparison). I kept the colours as default as I felt that they actually worked quite well. The countries were reordered by the largest visits but as you can see this had mixed results.

```r
#3.4. Create a dumbbell plot

#3.4.1. Create a subtitle that changes with the data selected.
subtitle1 <- paste("Showing the top", toString(cutrank1), "most visited countries in 2021")

#3.4.2. Match the aesthetics (basic plot)

p1 <- ggplot(longdfv1,
             aes(x = visits, y = reorder(country, visits)))

#3.4.3. Add layers
v1 <- p1 +   #lines to show the years are connected
             geom_line(color="grey") +
             #shows the year
             geom_point(aes(color = year), size = 3, alpha=0.8) +
             #Legend at the bottom for ease
             theme(legend.position = "bottom") +
             #Labels for the chart
             labs(x = "Number of visits from the UK (000s)",
                  y = "Country",
                  title = "Number of visits abroad from the UK in 2009 and 2021 by main country visited
                  subtitle = subtitle1,
                  caption= "Source: Office for National Statistics International Passenger Survey")

#Display visualisation 1
v1
```

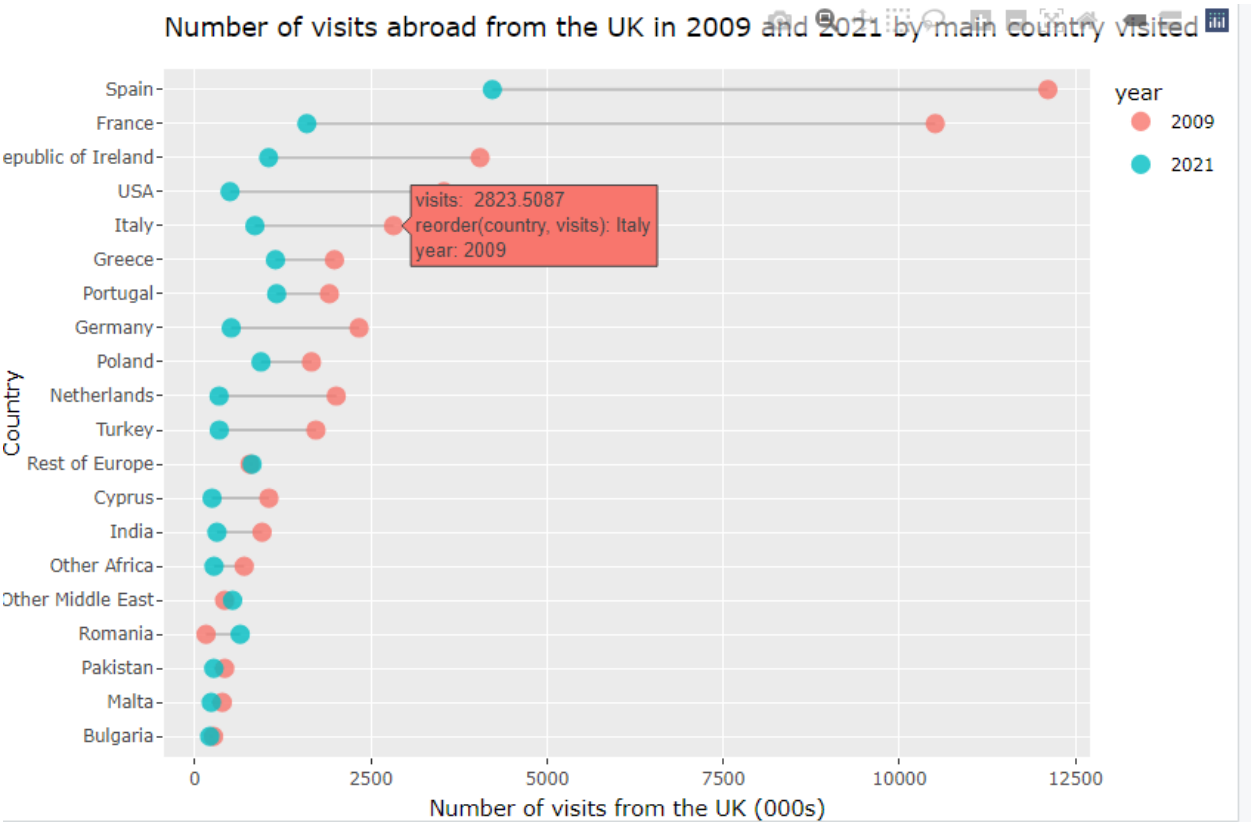



```
#3.5. Save the dumbbell plot
ggsave(here("plots", "dumbell.png"), v1)
```

Interactive plot

As I felt that this plot made reading the actual number difficult I entered it into ggplotly to allow for scrolling over points. This was useful but did make some of the formatting less clear. In this document I have commented the code to show this process and added a screenshot of the plot in action (please see repo).

```
#3.6. Add the dumbbell to plotly to make it interactive

#ip1 <- ggplotly(v1) #Use plotly for scrollover
#Display interactive visualisation 1
#ip1

#3.7. Save the interactive plot
#saveWidget(ip1, file='interactivedumbell.html')

#Image for markdown
screen <- here("documents", "printscreen_ip1.png") #location of file
knitr::include_graphics(screen)
```



Interpretation - has there been a change in travel patterns?

From this visualisation we can see that **most countries have seen a reduction in the number of visits from 2009 to 2021**. This is especially true for Spain and France but they had a much higher number of visitors in 2009. Although this is useful I don't think it shows the changes in travel patterns in an intuitive way. Additionally due to 2021 still being an atypical year for travel following COVID-19 the number of visits may be unreliable to say that is the trend.

Visualisation 2 (final) - Slope chart

Although visualisation 1 was useful I felt that it didn't show the overall trend in an intuitive way. I decided that ranks provided a way to show the same information in a simpler way. The trade off was losing the scale of difference between Spain and the rest of the countries.

I decided on a slope chart as this provides the most information without it being too cluttered.

Data changes

Prior to visualising I needed to convert the data from actual visits to the rank position for that year. Following on from that I needed to select the number of countries that I wanted included and then convert to the long format. Finally I selected only cases in 2009 and 2021.

```
#4.1. Create variable to allow for filtering the data

cutrank2 <- 15 # Create variables to allow us to choose the number of 2021 ranks
```

```r
yrs_inc2 <- c(2009, 2021) #Create variables for the years to include in visualisation


#4.2. Create new DF of just ranks

dfv2 <- df0921 %>%
        mutate(r09 = rank(-`2009`, ties.method = "average"),
               r10 = rank(-`2010`, ties.method = "average"),
               r11 = rank(-`2011`, ties.method = "average"),
               r12 = rank(-`2012`, ties.method = "average"),
               r13 = rank(-`2013`, ties.method = "average"),
               r14 = rank(-`2014`, ties.method = "average"),
               r15 = rank(-`2015`, ties.method = "average"),
               r16 = rank(-`2016`, ties.method = "average"),
               r17 = rank(-`2017`, ties.method = "average"),
               r18 = rank(-`2018`, ties.method = "average"),
               r19 = rank(-`2019`, ties.method = "average"),
               r20 = r19, #no data so saying that it would have been steady from 19
               r21 = rank(-`2021`, ties.method = "average")) %>% #r09-#r21 are the rank scores for each
        select("country",
               "r09", "r10", "r11",
               "r12", "r13", "r14",
               "r15", "r16", "r17",
               "r18", "r19", "r21",
               'rank_cut') %>% #selecting the ranked data but 2020 dropped due to no data
        rename('2009'='r09',
               '2010'='r10',
               '2011'='r11',
               '2012'='r12',
               '2013'='r13',
               '2014'='r14',
               '2015'='r15',
               '2016'='r16',
               '2017'='r17',
               '2018'='r18',
               '2019'='r19',
               '2021'='r21',
               'rank_cut2'='rank_cut') #changing rank names to years to allow for changing to long form

#4.3. Create summary DF based on the number of countries

dfv2 <- dfv2 %>%
        filter (rank_cut2 <=cutrank2)

#4.4. Change to Long format

longdfv2 <- dfv2 %>%
            select("country",
                   "2009", "2010",
                   "2011", "2012",
                   "2013", "2014",
                   "2015", "2016",
                   "2017", "2018",
```

```
                  "2019", "2021") %>%
           pivot_longer(!country,
                        names_to = "year",
                        names_transform = list(year = as.factor), #Changed to factor to allow for it
                        values_to = "rank") %>%
           filter(year %in% yrs_inc2) #remove years that are irrelevant

head(longdfv2)
```

```
## # A tibble: 6 x 3
##   country year   rank
##   <chr>   <fct> <dbl>
## 1 USA     2009      4
## 2 USA     2021     12
## 3 France  2009      2
## 4 France  2021      2
## 5 Germany 2009      6
## 6 Germany 2021     11
```

Colour scheme

An additional factor in the choice of country is the geographic region that it is in. To account for this I created a colour scheme where each country grouping (based on the ONS National statistics country classification). To develop this scheme each grouping got their own main colour with each individual country showing a different shade. The table below shows the data format.

```
#4.5.1. Retrieve data file containing the colours for each country/ continent
dataColour <- here("created_data", "colours.csv") #location of file
dfColour <- read.csv(file = dataColour) # read in data

head(dfColour)
```

```
##   continent            country colour_desc hex_value
## 1    Africa              Egypt   Soft pink   #e51f66
## 2    Africa            Morocco   Soft pink   #e83676
## 3    Africa            Tunisia   Soft pink   #ea4d85
## 4    Africa Other North Africa   Soft pink   #ed6495
## 5    Africa       South Africa   Soft pink   #f07ba5
## 6    Africa            Nigeria   Soft pink   #f292b4
```

This was then combined with the long rank data.

```
#4.5.2. Match the colour file to the data by country
longdfv2 <- left_join(longdfv2, dfColour,by="country")

head(longdfv2)
```

```
## # A tibble: 6 x 6
##   country year   rank continent      colour_desc     hex_value
##   <chr>   <fct> <dbl> <chr>          <chr>           <chr>
## 1 USA     2009      4 North America  Chestnut        #cd5c5c
```

```
## 2 USA     2021     12 North America  Chestnut        #cd5c5c
## 3 France  2009      2 Western Europe Cornflower blue #4d85ea
## 4 France  2021      2 Western Europe Cornflower blue #4d85ea
## 5 Germany 2009      6 Western Europe Cornflower blue #6495ed
## 6 Germany 2021     11 Western Europe Cornflower blue #6495ed
```

Visualisation

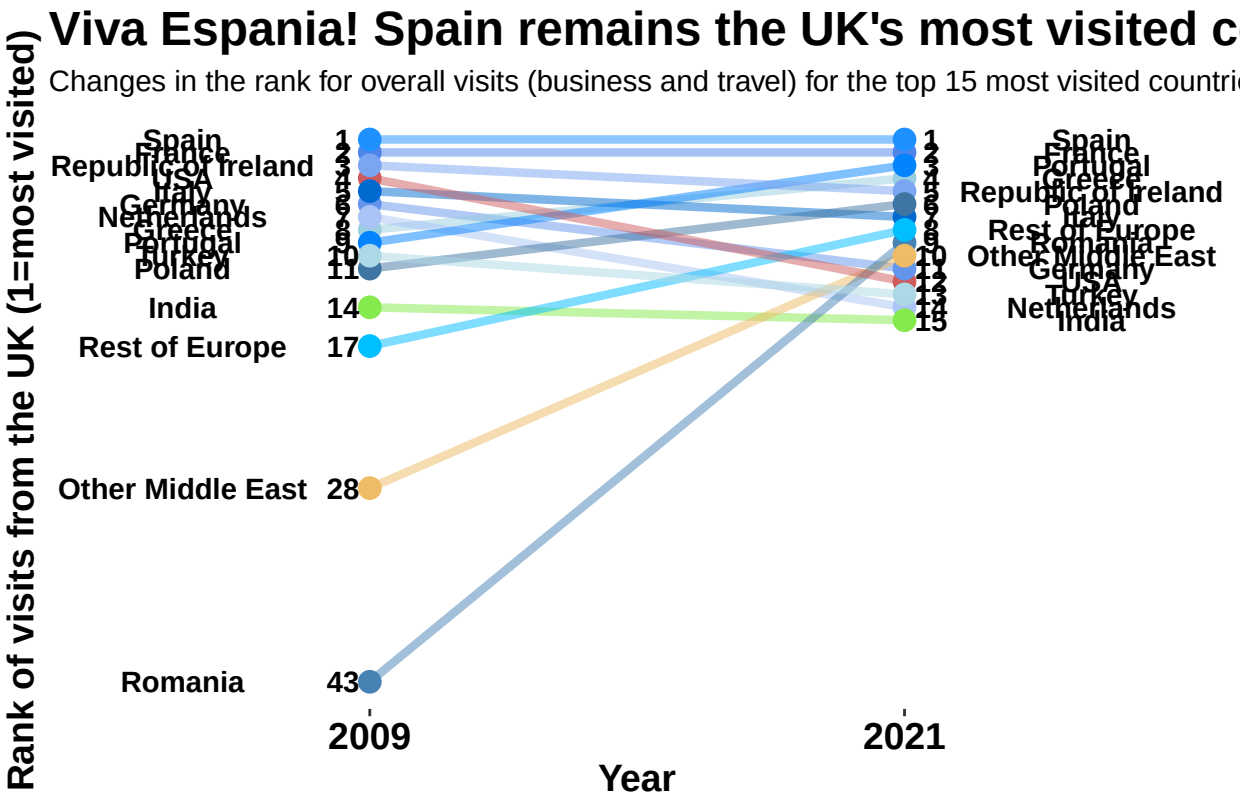
To create the final visualisation I used a line chart added to points at the end of each line to highlight the colour and position changes. To support understanding which countries are in which order I added the country name and rank next to the chart.

Other details:

- Subtitle and source that ammends automatically;
- Slight transparency to see where the lines are changing;
- Ordered 1 to lowest.

```
#4.6. Create the final visualisation (slope plot)

#4.6.1. Create titles for the plot

subtitle2 <- paste("Changes in the rank for overall visits (business and travel) for the top",
                   toString(cutrank2),
                   " most visited countries from the UK between"
                   ,toString(first(yrs_inc2)),
                   "to" ,
                   toString(last(yrs_inc2))) # Label explaining the data

Source2 <- bquote(paste(~bold('Source:'),"ONS International Passenger Survey",
                        ~bold('Note:'),"Colours indicate country grouping; 2021 had disrupted data coll
           #data source and interpretation notes

title2 <- paste("Viva Espania! Spain remains the UK's most visited country")

#4.6.2. Match the aesthetics (basic plot)

p2 <- longdfv2 %>%
      ggplot(aes(x=year, y=rank,group=country, colour= country))


#4.6.3. Add the formatting

final <- p2+
         #slope lines
         geom_line(alpha = 0.5, linewidth=1.5) +

         #End points for the data
         geom_point(size = 3.5) +

         #Labels to show the country and rank next to each of the countries
         geom_text(data= longdfv2 %>% filter(year==(toString(first(yrs_inc2)))), #selects the first yea
                   aes(x=year, y=rank,label = country), #add the country name
```

```
                  size = 4, nudge_x = -0.35, fontface = "bold", color = "#000000") + # format so next

        geom_text(data= longdfv2 %>% filter(year==(toString(last(yrs_inc2)))), #selects the last year
                  aes(x=year, y=rank,label = country), # add the country name
                  size = 4, nudge_x = 0.35, fontface = "bold", color = "#000000") + #format so next to

        geom_text(data= longdfv2 %>% filter(year ==(toString(first(yrs_inc2)))), #select the first yea
                  aes(x=year, y=rank,label = rank), #add the rank position
                  size = 4, nudge_x = -0.05, fontface = "bold", color = "#000000") + #place next to co

        geom_text(data= longdfv2 %>% filter(year ==(toString(last(yrs_inc2)))), #select the last year
                  aes(x=year, y=rank,label = rank), # add the rank to the chart
                  size = 4, nudge_x = 0.05,fontface = "bold", color = "#000000") + #formatting

        #Reverse the scale so that the lowest ranked (most visited) is at the top
        scale_y_reverse( breaks=NULL) +

        #Use custom colours
        scale_color_manual(values = setNames(longdfv2$hex_value, longdfv2$country)) +

        #Add labels
        labs(x = "Year", #X-axis
             y = "Rank of visits from the UK (1=most visited)", #y-axis
             title = title2, #main title
             subtitle = str_wrap(subtitle2, width=150), #subtitle
             caption= Source2) + #caption with each on a new line

        #Specify the theme (fonts, background etc.)
        theme(plot.title = element_text(size=18, face = "bold"), #Main title font
        axis.text.x = element_text(size=14, face="bold", colour="black"), #X-axis element font
        axis.text.y = element_text(size=12, face="bold"), #y-axis years font
        axis.title = element_text(size=14,face = "bold"), #Axis title fonts
        panel.background = element_blank(), #Make the background blank
        legend.position = "none",#Remove the legend
        plot.caption = element_text(size = 10, hjust = 1)) #Caption font


final
```



ernational Passenger Survey **Note:**Colours indicate country grouping; 2021 had disrupted data collection

```
#4.6.4. Save the dumbbell plot
ggsave(here("plots", "finalplot.png"), final, width = 13.44, height = 8.595)
```

Interpretation - has there been a change in travel patterns?

The slope chart of ranked data is much clearer to interpret because the steepness of the slopes shows the trend. Overall there has been a **slight change in travel patterns** but this is not drastic or shifting the areas of the World that people are visiting.

However more specific insights include:

- Spain and France have remained first and second in the ranks.
- Since 2009 only 3 countries have entered/ left the top 15 ranked countries.
- Romania has had the most dramatic increase in rank from 43 to 9 followed by other Middle East.
- The USA saw the greatest drop in rank position and remained in the top 15.
- Most countries are in Europe as indicated by the blue colours with only 3 countries in other global regions.

Reflections

The main reflection that I have on the visualisations is the difficulties in trying to provide all the information available in a simple format. Although it can seem inappropriate to lose data on actual visits and the scale of the Spanish visits this was necessary to see the big picture. I think that visualisation 2 is sufficient to get an overall idea of the trends with UK travel.

The final visualisation does have some limitations: * The colour meanings are unclear as I did not find a neat way to indicate the colours. * Issues with the data collection in 2021 make them slightly less reliable especially on France/ Republic of Ireland. * Scaling issues where the text can get squashed.

If there was more time and I had access to the full data set for the IPS since 1961 I would have liked to create an interactive dashboard of the ranked travel data. This would allow people to type in their countries of interest and see longer term trends. This could also allow people who are interested to click on the position to see the actual number of visits. An example of this is provided by the ONS on 100 years of baby names.

Additionally I would have spent more time learning how to format an rMarkdown document.

References

- The **repo** for this project (including data, code, cookbook) is available online via gihub.
- Access to the **raw data** is available from the ONS (link above) and their publication (accessed 15th May 23) on travel data shows the range of data from the IPS.