Agents historic company visit scores

Azeez Akintonde

2024-04-15

Introduction

The goal of this analysis is to explore the historic company visit scores collected by the Bank of England's Agents from 2009-2020. We'll load the dataset, perform data analysis, and generate visualizations to gain insights into sector performance over time.

Task 1

Load Libraries and Dataset

```
# Load libraries
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.1
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

```
# Set directories
data_dir <- "C:/Users/Kintonde/Documents/VSCODE/Agents-historic-company-visit-scores"
setwd(data_dir)

# dataset to be use
library(readxl)
df <- read_excel("C:/Users/Kintonde/Desktop/Byblos/banford/hmd/Agents historic company visit scores.xlsx
    sheet = "DATASET")
head(df)
```

Data Exploration

```
# calculate the total number of columns
num_columns <- ncol(df)

# calculate the total number of Observations
num_observations <- nrow(df)
cat("There are",
    num_columns,
    "columns and",
    num_observations,
    "observations.\n")
```

```
## There are 27 columns and 41701 observations.
```

```
# Check the datset structure
str(df)
```

```
## tibble [41,701 x 27] (S3: tbl_df/tbl/data.frame)
##  $ ActualDateDisplay                    : chr [1:41701] "Q1-2008" "Q1-2008" "Q1-2008" "Q1-2008" ...
##  $ Sector                               : chr [1:41701] "Production" "Production" "Distribution, hot
##  $ DemandScore                          : num [1:41701] 3 2 3 -3 5 4 -1 0 0 -2 ...
##  $ ExportScore                          : num [1:41701] 3 2 3 1 5 1 1 1 3 -2 ...
##  $ InvestmentScore                      : num [1:41701] 1 0 4 1 3 3 2 1 0 0 ...
##  $ CapacityScore                        : num [1:41701] 1 2 0 -1 2 2 -1 0 0 2 ...
##  $ EmploymentScore                      : num [1:41701] 4 0 0 0 2 3 -1 -1 0 -3 ...
##  $ RecruitmentScore                     : num [1:41701] 1 2 2 0 0 2 1 3 0 0 ...
##  $ TotalLabourCostsScore                : num [1:41701] 2 2 3 2 3 3 4 2 2 3 ...
##  $ PayScore                             : num [1:41701] 2 2 1 2 2 3 4 2 2 2 ...
##  $ NonLabourCostsScore                  : num [1:41701] 1 3 4 0 3 2 3 4 3 1 ...
##  $ OutputPricesScore                    : num [1:41701] -3 1 0 2 3 4 3 3 3 2 ...
##  $ PreTaxProfitsScore                   : num [1:41701] 2 0 2 2 3 4 0 1 3 3 ...
##  $ ProfitMarginRelativeNormalScore      : logi [1:41701] NA NA NA NA NA NA ...
##  $ FutureDemandScore                    : num [1:41701] 3 3 4 0 5 1 0 0 2 2 ...
##  $ FutureExportScore                    : num [1:41701] 3 2 0 0 5 -1 1 1 2 1 ...
##  $ FutureInvestmentScore                : num [1:41701] 1 2 -1 2 5 1 5 1 2 -2 ...
##  $ FutureCapacityScore                  : num [1:41701] 1 3 -2 2 3 1 0 0 1 2 ...
##  $ FutureEmploymentScore                : num [1:41701] 4 2 0 0 1 1 3 -1 -1 -3 ...
##  $ FutureRecruitmentScore               : num [1:41701] 1 1 2 2 0 1 1 3 0 0 ...
##  $ FutureTotalLabourCostsScore          : num [1:41701] 2 2 -1 1 1 2 3 4 2 -3 ...
##  $ FuturePayScore                       : num [1:41701] 2 2 2 2 1 2 3 2 2 2 ...
##  $ FutureNonLabourCostsScore            : num [1:41701] 3 2 2 0 1 3 1 3 3 1 ...
##  $ FutureOutputPricesScore              : num [1:41701] -3 3 0 2 0 4 0 3 3 0 ...
##  $ FuturePreTaxProfitsScore             : num [1:41701] 2 1 1 -1 -1 2 3 2 0 -1 ...
##  $ FutureProfitMarginRelativeNormalScore: logi [1:41701] NA NA NA NA NA NA ...
##  $ CreditAvailabilityScore              : logi [1:41701] NA NA NA NA NA NA ...
```

```
# Print the head of the dataset
head(df)
```

Task 2

identify all companies whose Investment Score and Employment Score are both positive – store the company IDs in a vector and display them effectively.

```
# Filter companies with positive Investment and Employment Scores
comp_positive_score <-
  subset(df, InvestmentScore > 0 & EmploymentScore > 0)

# calculate the total number of columns
fil_columns <- ncol(comp_positive_score)

# calculate the total number of Observations
fil_observations <- nrow(comp_positive_score)
cat("There are",
    fil_columns,
    "columns and",
    fil_observations,
```

```
    "observations after Filtering the positive Investment and Employment Scores.\n")
```

```
## There are 27 columns and 9921 observations after Filtering the positive Investment and Employment Sc
```

Task 3

Add a column to your data frame to indicate the Category of each company. You should refer to the SIC column to determine the category of the company. The name of the new column should be "Category" and the data type of the column should be character

```
# rename the Sector column to Category
colnames(comp_positive_score)[colnames(comp_positive_score) == "Sector"] <-
  "Category"
# print the data structure
str(comp_positive_score)
```

```
## tibble [9,921 x 27] (S3: tbl_df/tbl/data.frame)
##  $ ActualDateDisplay                    : chr [1:9921] "Q1-2008" "Q1-2008" "Q1-2008" "Q1-2008" ...
##  $ Category                             : chr [1:9921] "Production" "Production" "Business and finan
##  $ DemandScore                          : num [1:9921] 3 5 4 0 3 3 3 4 4 4 ...
##  $ ExportScore                          : num [1:9921] 3 5 1 0 1 3 4 4 4 0 ...
##  $ InvestmentScore                      : num [1:9921] 1 3 3 1 4 2 2 3 2 1 ...
##  $ CapacityScore                        : num [1:9921] 1 2 2 3 1 4 1 3 3 2 ...
##  $ EmploymentScore                      : num [1:9921] 4 2 3 1 2 2 1 3 3 4 ...
##  $ RecruitmentScore                     : num [1:9921] 1 0 2 1 1 2 1 3 1 2 ...
##  $ TotalLabourCostsScore                : num [1:9921] 2 3 3 0 1 2 3 2 2 4 ...
##  $ PayScore                             : num [1:9921] 2 2 3 2 2 3 2 3 2 4 ...
##  $ NonLabourCostsScore                  : num [1:9921] 1 3 2 1 0 2 2 3 1 2 ...
##  $ OutputPricesScore                    : num [1:9921] -3 3 4 1 -2 3 -1 3 2 3 ...
##  $ PreTaxProfitsScore                   : num [1:9921] 2 3 4 -2 2 2 3 4 4 -1 ...
##  $ ProfitMarginRelativeNormalScore      : logi [1:9921] NA NA NA NA NA NA ...
##  $ FutureDemandScore                    : num [1:9921] 3 5 1 0 4 3 3 0 3 1 ...
##  $ FutureExportScore                    : num [1:9921] 3 5 -1 0 4 3 -2 2 3 2 ...
##  $ FutureInvestmentScore                : num [1:9921] 1 5 1 3 -2 3 2 3 2 0 ...
##  $ FutureCapacityScore                  : num [1:9921] 1 3 1 3 1 4 1 1 2 1 ...
##  $ FutureEmploymentScore                : num [1:9921] 4 1 1 2 2 2 -1 3 3 3 ...
##  $ FutureRecruitmentScore               : num [1:9921] 1 0 1 1 2 2 1 3 0 2 ...
##  $ FutureTotalLabourCostsScore          : num [1:9921] 2 1 2 0 2 2 2 2 3 4 ...
##  $ FuturePayScore                       : num [1:9921] 2 1 2 2 2 2 2 2 2 4 ...
##  $ FutureNonLabourCostsScore            : num [1:9921] 3 1 3 1 0 2 2 3 1 2 ...
##  $ FutureOutputPricesScore              : num [1:9921] -3 0 4 2 -1 3 -1 1 3 3 ...
##  $ FuturePreTaxProfitsScore             : num [1:9921] 2 -1 2 2 3 2 0 1 3 1 ...
##  $ FutureProfitMarginRelativeNormalScore: logi [1:9921] NA NA NA NA NA NA ...
##  $ CreditAvailabilityScore              : logi [1:9921] NA NA NA NA NA NA ...
```

```
# Function to map categories to SIC codes
SIC <- function(category) {
  switch(
    category,
    "Production" = "A, B, C, D, E, F",
    "Business and financial services" = "K, L, M, N",
    "Transport, storage and communications" = "H, J",
    "Distribution, hotels, catering, arts, entertainment, recreation and other services" = "G, I, R, S"
    "Unknown"
```

```
  )}
```

```
# Create 'Potential_SICs' column based on SIC codes using the using the respective category
comp_positive_score$Potential_SICs <-
  sapply(comp_positive_score$Category, SIC)
# print the data structure
str(comp_positive_score)
```

```
## tibble [9,921 x 28] (S3: tbl_df/tbl/data.frame)
##  $ ActualDateDisplay                    : chr [1:9921] "Q1-2008" "Q1-2008" "Q1-2008" "Q1-2008" ...
##  $ Category                             : chr [1:9921] "Production" "Production" "Business and financ
##  $ DemandScore                          : num [1:9921] 3 5 4 0 3 3 3 4 4 4 ...
##  $ ExportScore                          : num [1:9921] 3 5 1 0 1 3 4 4 4 0 ...
##  $ InvestmentScore                      : num [1:9921] 1 3 3 1 4 2 2 3 2 1 ...
##  $ CapacityScore                        : num [1:9921] 1 2 2 3 1 4 1 3 3 2 ...
##  $ EmploymentScore                      : num [1:9921] 4 2 3 1 2 2 1 3 3 4 ...
##  $ RecruitmentScore                     : num [1:9921] 1 0 2 1 1 2 1 3 1 2 ...
##  $ TotalLabourCostsScore                : num [1:9921] 2 3 3 0 1 2 3 2 2 4 ...
##  $ PayScore                             : num [1:9921] 2 2 3 2 2 3 2 3 2 4 ...
##  $ NonLabourCostsScore                  : num [1:9921] 1 3 2 1 0 2 2 3 1 2 ...
##  $ OutputPricesScore                    : num [1:9921] -3 3 4 1 -2 3 -1 3 2 3 ...
##  $ PreTaxProfitsScore                   : num [1:9921] 2 3 4 -2 2 2 3 4 4 -1 ...
##  $ ProfitMarginRelativeNormalScore      : logi [1:9921] NA NA NA NA NA NA ...
##  $ FutureDemandScore                    : num [1:9921] 3 5 1 0 4 3 3 0 3 1 ...
##  $ FutureExportScore                    : num [1:9921] 3 5 -1 0 4 3 -2 2 3 2 ...
##  $ FutureInvestmentScore                : num [1:9921] 1 5 1 3 -2 3 2 3 2 0 ...
##  $ FutureCapacityScore                  : num [1:9921] 1 3 1 3 1 4 1 1 2 1 ...
##  $ FutureEmploymentScore                : num [1:9921] 4 1 1 2 2 2 -1 3 3 3 ...
##  $ FutureRecruitmentScore               : num [1:9921] 1 0 1 1 2 2 1 3 0 2 ...
##  $ FutureTotalLabourCostsScore          : num [1:9921] 2 1 2 0 2 2 2 2 3 4 ...
##  $ FuturePayScore                       : num [1:9921] 2 1 2 2 2 2 2 2 2 4 ...
##  $ FutureNonLabourCostsScore            : num [1:9921] 3 1 3 1 0 2 2 3 1 2 ...
##  $ FutureOutputPricesScore              : num [1:9921] -3 0 4 2 -1 3 -1 1 3 3 ...
##  $ FuturePreTaxProfitsScore             : num [1:9921] 2 -1 2 2 3 2 0 1 3 1 ...
##  $ FutureProfitMarginRelativeNormalScore: logi [1:9921] NA NA NA NA NA NA ...
##  $ CreditAvailabilityScore              : logi [1:9921] NA NA NA NA NA NA ...
##  $ Potential_SICs                       : Named chr [1:9921] "A, B, C, D, E, F" "A, B, C, D, E, F" "
##   ..- attr(*, "names")= chr [1:9921] "Production" "Production" "Business and financial services" "Pr
```

```
# Extract year from 'ActualDateDisplay' and create 'Year' column
comp_positive_score$Year <- substr(
  comp_positive_score$ActualDateDisplay,
  nchar(comp_positive_score$ActualDateDisplay) - 3,
  nchar(comp_positive_score$ActualDateDisplay))

# print the data structure
str(comp_positive_score)
```

```
## tibble [9,921 x 29] (S3: tbl_df/tbl/data.frame)
##  $ ActualDateDisplay                    : chr [1:9921] "Q1-2008" "Q1-2008" "Q1-2008" "Q1-2008" ...
##  $ Category                             : chr [1:9921] "Production" "Production" "Business and financ
##  $ DemandScore                          : num [1:9921] 3 5 4 0 3 3 3 4 4 4 ...
##  $ ExportScore                          : num [1:9921] 3 5 1 0 1 3 4 4 4 0 ...
##  $ InvestmentScore                      : num [1:9921] 1 3 3 1 4 2 2 3 2 1 ...
##  $ CapacityScore                        : num [1:9921] 1 2 2 3 1 4 1 3 3 2 ...
```

```
##  $ EmploymentScore                      : num [1:9921] 4 2 3 1 2 2 1 3 3 4 ...
##  $ RecruitmentScore                     : num [1:9921] 1 0 2 1 1 2 1 3 1 2 ...
##  $ TotalLabourCostsScore                : num [1:9921] 2 3 3 0 1 2 3 2 2 4 ...
##  $ PayScore                             : num [1:9921] 2 2 3 2 2 3 2 3 2 4 ...
##  $ NonLabourCostsScore                  : num [1:9921] 1 3 2 1 0 2 2 3 1 2 ...
##  $ OutputPricesScore                    : num [1:9921] -3 3 4 1 -2 3 -1 3 2 3 ...
##  $ PreTaxProfitsScore                   : num [1:9921] 2 3 4 -2 2 2 3 4 4 -1 ...
##  $ ProfitMarginRelativeNormalScore      : logi [1:9921] NA NA NA NA NA NA ...
##  $ FutureDemandScore                    : num [1:9921] 3 5 1 0 4 3 3 0 3 1 ...
##  $ FutureExportScore                    : num [1:9921] 3 5 -1 0 4 3 -2 2 3 2 ...
##  $ FutureInvestmentScore                : num [1:9921] 1 5 1 3 -2 3 2 3 2 0 ...
##  $ FutureCapacityScore                  : num [1:9921] 1 3 1 3 1 4 1 1 2 1 ...
##  $ FutureEmploymentScore                : num [1:9921] 4 1 1 2 2 2 -1 3 3 3 ...
##  $ FutureRecruitmentScore               : num [1:9921] 1 0 1 1 2 2 1 3 0 2 ...
##  $ FutureTotalLabourCostsScore          : num [1:9921] 2 1 2 0 2 2 2 2 3 4 ...
##  $ FuturePayScore                       : num [1:9921] 2 1 2 2 2 2 2 2 2 4 ...
##  $ FutureNonLabourCostsScore            : num [1:9921] 3 1 3 1 0 2 2 3 1 2 ...
##  $ FutureOutputPricesScore              : num [1:9921] -3 0 4 2 -1 3 -1 1 3 3 ...
##  $ FuturePreTaxProfitsScore             : num [1:9921] 2 -1 2 2 3 2 0 1 3 1 ...
##  $ FutureProfitMarginRelativeNormalScore: logi [1:9921] NA NA NA NA NA NA ...
##  $ CreditAvailabilityScore              : logi [1:9921] NA NA NA NA NA NA ...
##  $ Potential_SICs                       : Named chr [1:9921] "A, B, C, D, E, F" "A, B, C, D, E, F" "
##   ..- attr(*, "names")= chr [1:9921] "Production" "Production" "Business and financial services" "Pr
##  $ Year                                 : chr [1:9921] "2008" "2008" "2008" "2008" ...
```

```r
# Aggregate data to calculate average scores by Category and Year
Avg_score <- aggregate(
  cbind(
    Avg_InvestmentScore = InvestmentScore,
    Avg_EmploymentScore = EmploymentScore
  ) ~ Category
  + Year,
  data = comp_positive_score,
  FUN = mean,
  na.rm = TRUE)
# print the aggregated dataset
head(Avg_score)
```

```r
# calculate the total number of columns in the Aggregated Avg_Score dataset
fil_columns <- ncol(Avg_score)

# calculate the total number of Observations
fil_observations <- nrow(Avg_score)
cat("There are",
    fil_columns,
    "columns and",
    fil_observations,
    "observations in the Avg_score dataset.\n")
```

```
## There are 4 columns and 52 observations in the Avg_score dataset.
```

```r
# print the dataset stucture
str(Avg_score)
```

```
## 'data.frame':    52 obs. of  4 variables:
##  $ Category           : chr  "Business and financial services" "Distribution, hotels, catering, arts
```

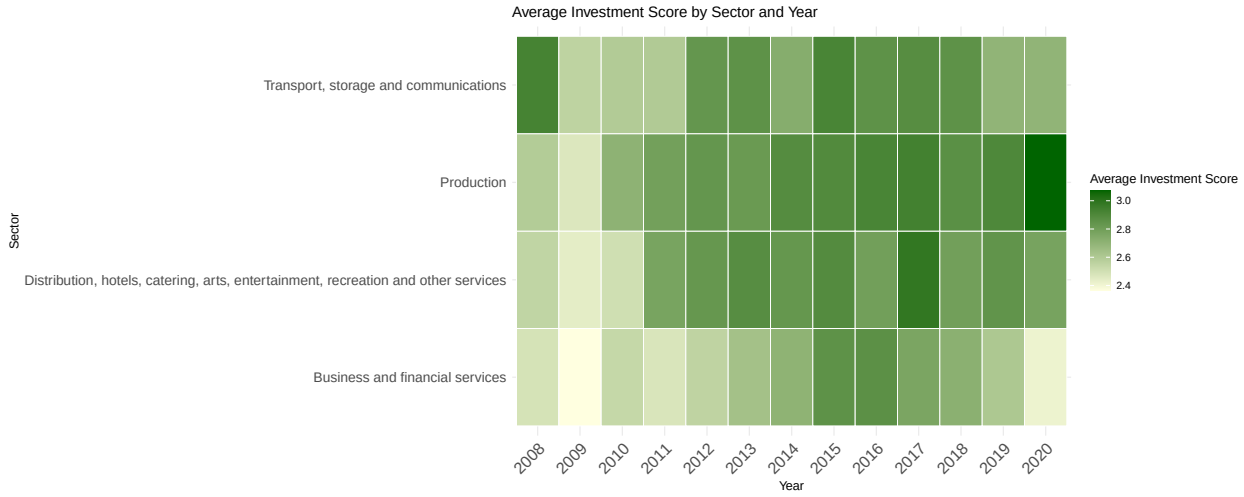
```
##  $ Year                : chr  "2008" "2008" "2008" "2008" ...
##  $ Avg_InvestmentScore: num  2.49 2.56 2.59 2.93 2.37 ...
##  $ Avg_EmploymentScore: num  2.59 2.13 2.35 2.36 2.59 ...
```

Task 4

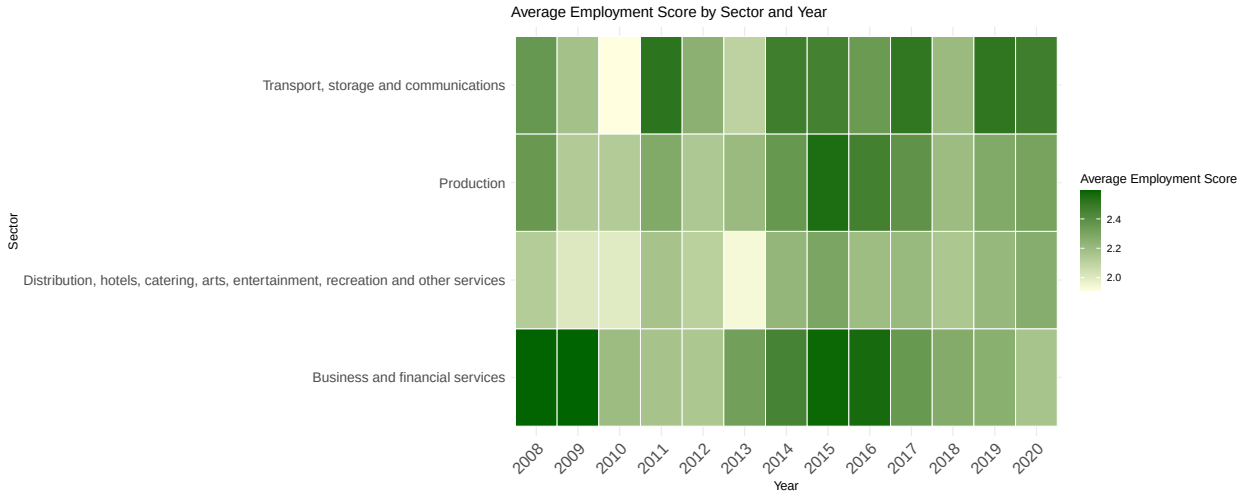
Generate two visualisations/charts based on information from the dataset.

Data Visualization

```
# Create heatmap for Average Investment Score
ggplot(data = Avg_score, aes(x = Year, y = Category,
                             fill = Avg_InvestmentScore)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow",# set lightyellow for the low values
                      high = "darkgreen", # set darkgreen fro the high values
                      name = "Average Investment Score") +
  labs(title = "Average Investment Score by Sector and Year",
       x = "Year", y = "Sector") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 14),  # Adjust size as desired
        axis.text.y = element_text(size = 12),  # Adjust size as desired
        legend.position = "right")
```




```
# Create heatmap for Average Employment Score
ggplot(data = Avg_score, aes(x = Year, y = Category, fill = Avg_EmploymentScore)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow",# set lightyellow for the low values
                      high = "darkgreen",  # set darkgreen fro the high values
                      name = "Average Employment Score") +
  labs(title = "Average Employment Score by Sector and Year",
       x = "Year", y = "Sector") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 14),  # Adjust size as desired
        axis.text.y = element_text(size = 12),  # Adjust size as desired
        legend.position = "right")
```



Year-on-Year Analysis of Investment and Employment Trends Across Sectors

Investment Score: The analysis examines the fluctuation of average investment scores across various sectors from 2008 to 2020, utilizing data collected by the Bank of England's Agents. Notably, the Transportation, storage, and communications sector consistently exhibited the highest average investment score throughout the period, peaking at approximately 3.0 in 2008 before declining to around 2.6 in 2012.

- **Transportation, Storage, and Communications:** This sector demonstrated a sustained high average investment score, indicating relative stability in investment activity.
- **Production:** Despite fluctuations, investment scores remained relatively robust, particularly evident in 2008 and 2012.
- **Distribution, Hospitality, Arts, Entertainment, and Other Services:** Consistently, this sector registered the lowest average investment scores, ranging between 2.2 and 2.4.
- **Business and Financial Services:** Similar to the distribution sector, investment scores fluctuated within a narrow range between 2.2 and 2.4, with a slight upward trend observed from 2018 to 2020.

Employment Score: In parallel, employment trends across sectors depicted varying patterns over the same period.

- **Transportation, Storage, and Communications:** This sector consistently maintained a high average employment score, indicating relative stability in employment levels compared to other sectors.
- **Production:** Employment in the production sector exhibited greater volatility, likely reflecting the influence of economic fluctuations such as the 2008 recession.
- **Distribution, Hospitality, Arts, Entertainment, and Other Services:** While initially low, there was a discernible upward trend in employment scores from 2018 to 2020, suggesting potential growth in these sectors.
- **Business and Financial Services:** Employment trends mirrored those of the distribution sector, with marginal increases observed in later years.

Conclusion:

The analysis provides valuable insights into investment and employment dynamics across sectors over the specified period. While certain sectors consistently outperformed others in terms of investment and employment, the observed fluctuations underscore the influence of economic factors on industry-specific trends.

Limitations:

- **Data Availability:** The analysis relies on data collected by the Bank of England's Agents, which may have limitations in terms of coverage and granularity.
- **Sector Classification:** The broad categorization of sectors may oversimplify the analysis, potentially masking nuanced variations within industries.
- **Economic Context:** External economic factors, such as global economic conditions and government policies, were not explicitly considered in the analysis, which may impact the interpretation of trends.