ISSSV1337 2023 SSB

Anita, Fria, Magne, Julie, Petar

Introduction

The task assigned to our group was formulated by Statistics Norway (SSB), the central Norwegian office for official government statistics. The main goal of the task at hand was to describe the development of the business sector in Norway compared to other relevant countries. Drawing such comparisons could be instrumental in gaining a deeper understanding of the global business environment which is becoming increasingly interdependent, particularly in the context of economic shocks and the ongoing green shift. These substantial changes affect the allocation of labour and capital between different types of enterprises and industries, consequently affecting the research and development strategies on both the sectoral and national levels.

In order to provide a comprehensive analysis required by the task formulation, we needed firstly to select the criteria for the classification of the business sector and industrial structure, as well as to select and justify indicators for development. Moreover, it was necessary to make a proper selection of comparable countries and the most appropriate timeframe that could produce relevant results. The next step was to find the right data sources that could enable us to conduct the analysis and report the results. The final stage was supposed to include a justification of the chosen approach and an explanation of its relevance.

Interactive Dashboard: https://magne.shinyapps.io/ISSSV1337_2023_SSB/

Brainstorming

The fact that all members of our group have a different educational background allowed us to discuss potential solutions from different perspectives. That turned out to be very helpful when faced with a complex issue that requires a certain level of understanding economic, political, and statistical aspects of the issue. Deciding on the list of comparable countries to Norway was the first major problem that needed to be addressed. If we were to include Norway's major trading partners in the list, it would include some EU countries but also such countries as the US, UK, China, and Canada. However, we soon realized that comparing Norway to large countries such as the US and China might be problematic with regard to both sociopolitical and economic dimensions. So, we decided to focus rather on those countries that do not diverge too much from Norway's demographic, political, and economic parameters. It turned out that the list of relevant countries in this case includes the member states of the European Union plus the United Kingdom.

In order to narrow down the number of countries that could give us the most meaningful comparison, we decided to form a criterion based on variables in four different categories, including economy and demographics, political governance, social equality and rights, and corruption. The most suitable data source for this task were datasets provided by the Varieties of Democracy (V-Dem). For instance, some of the variables we used to rank the relevant countries in terms of economy and demographics include "land area", "population", and "GDP per capita", while our evaluation of political governance relies upon variables such as "liberal democracy index", "free and fair elections", or "domestic autonomy". As for the social equality and rights dimension, we decided to use "freedom of religion" and "educational quality" among others, whereas the level of corruption is measured based on the variables like "regime corruption", "media corruption", or "control of corruption".

When it comes to presenting the economic landscape of comparable countries as the central issue of our task, we were initially considering drawing the relevant data from different sources, but we soon realized that having one data source with different economic indicators would be a better choice. Once we decided that our list of relevant countries should include only EU/EEA member states plus the UK, it became apparent that our analysis had to be based on the Eurostat database. The decision to include only European countries into our comparison resolved a potential problem with differing statistical classification of economic activities on different continents, as we could rely solely on the European Nomenclature of Economic Activities (NACE). In that way we were able to gather comparable data through a Eurostat API query to import relevant economic indicators such as "value added" or "persons employed" into R. The next challenge we faced was deciding on the most adequate timeframe for our analysis. It turned out that the most preferred timeframe that spans from early 2000s up to the present day was not feasible due to missing data prior to 2005 and after 2020. So, the only way to go was to conduct our analysis on the available data for the years between 2005 and 2020. However, we were still facing the problem of missing values for some years and some variables. After some consultations, we decided to solve the problem of missing values by applying imputation.

Policy Relevance:

The analysis of the business sector's development is of significant policy relevance. Understanding how different sectors contribute to economic growth can inform policymaking regarding investments, education, and workforce development. For example, identifying growing sectors may suggest the need for increased educational programs to meet the demand for skilled labor. Additionally, studying the distribution of employment across sectors helps anticipate the impact of economic shocks on various industries and allows for better economic planning and policy responses.

Criteria for Classification of Business Sector and Industrial Structure:

To classify the business sector and its industrial structure, we will use the NACE codes (Statistical Classification of Economic Activities in the European Community). NACE provides a standardized way to categorize economic activities, making it suitable for cross-country comparisons. Simply put, we will group businesses based on their economic activities, enabling us to analyze the performance of specific sectors over time.

Indicators for Development:

To describe the development of the business sector, we will use the following indicators: **1. Value Added at Factor Cost.** Which is defined as the gross income from operating activities after adjusting for operating subsidies and indirect taxes. The value added at factor cost is, in other words, the value businesses create after deducting input costs. This indicator can tell us something about the sector's overall economic performance and its capacity to generate income and wealth for the economy. By comparing the Value Added across sectors, we can identify which sectors drive economic growth and contribute the most to the GDP. Additionally, understanding the value-added contributions of different sectors helps policymakers assess the diversification of the economy and its resilience to economic shocks.

2. Number of Employees: The number of persons employed is defined, within the context of structural business statistics, as the total number of persons who work in the observation unit (inclusive of working proprietors, partners working regularly in the unit and unpaid family workers), as well as persons who work outside the unit who belong to it and are paid by it (e.g. sales representatives, delivery personnel, repair and maintenance teams). It excludes manpower supplied to the unit by other enterprises, persons carrying out repair and maintenance work in the enquiry unit on behalf of other enterprises, as well as those on compulsory military service.

The number of employees in each sector provides valuable information about the distribution of employment within the economy. It helps us to understand the labor market structure and the role of different sectors in providing employment opportunities to the workforce. Changes in the number of employees over time can indicate shifts in labor demand and the relative importance of various sectors in job creation.

Policy Relevance: Policymakers can use this indicator to address labor market challenges and develop targeted policies to address potential unemployment issues in specific sectors. For example, if a sector is experiencing a decline in employment, policymakers may consider providing training and support for transitioning workers to find opportunities in other growing sectors. Additionally, understanding the distribution of employment across sectors can inform workforce development initiatives and education policies, aligning them with the needs of the job market.

In addition, we calculated the **"Value Added at Factor Cost per Employee."** This indicator represents the value added to the economy per employed in each sector. It is obtained by dividing the total value added at factor cost by the number of employees within a specific sector.

Loading All Libraries

For this project, we ended up using a number of packages from RStudio that significantly enhanced our data analysis and visualization capabilities. Each package serves a specific purpose and collectively forms a powerful toolkit for R programming. Let's briefly explore the functionalities of some of the key packages we utilized:

- **tidyverse**: Collection of packages for data manipulation and visualization.
- **eurostat**: Facilitates retrieval of statistical data from Eurostat database.
- **httr**: Handles HTTP requests and responses, useful for web APIs.
- **rjstat**: Works with data in the JSON-stat format.
- **rlang**: Provides functions for working with language objects.
- **colorspace**: Manipulates and converts color spaces in R.
- **imputeTS**: Offers imputation methods for time series data.
- **zoo**: Handles irregular time series data with the zoo class.
- **shiny**: Allows building interactive web applications in R.
- **shinydashboard**: Specializes in creating attractive dashboards.
- **plotly**: Creates interactive visualizations in R.
- **RColorBrewer**: Provides color palettes for appealing plots.

```
pacman::p_load(tidyverse, eurostat, httr, rjstat, rlang, colorspace, imputeTS, zoo, shiny,
```

Comparative countries

What are comparative countries? They are probably countries with a lot of similarities.

In order to compare how similar all countries are, it would be quite overwhelming to look at all the countries in the world. Then we need to figure out a starting point.

Trading partners

A suggestion given in the task is to look at Norway's biggest trading partners. This includes EU countries, US, UK, Canada and China.

Is looking at trading partners really a good metric? Probably not.

The reason why we trade with other countries is because, either we produce goods they do not produce enough of, or the opposite. That creates a need to buy them from someone who do produce more than enough. Then we already know that there is a difference in the industries between these countries. Looking at trading partners will then only tell us who we buy the stuff we need from, or sell to.

Norway will also naturally trade more with countries that share a trading agreement or are in close proximity, that will mostly be EU/EEA countries.

So, although we should not base our comparisons on trading partners, it can be a good staring point.

Here is a vector of EU and EEA countries. Lichtenstein did unfortunately not work for this analysis, and is therefore excluded.

```
# Creating a vector of relevant countries
countries <- c("Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czechia", "Denmark"
```

V-Dem

How can we then figure out how similar these countries are?

The V-Dem data frame consists of many variables describing all countries. We can use some of these variables to compare the relevant countries.

We have picked out variables in four different categories:

- Economic and Demographic
- Political Governance
- Social Equality and Rights
- Corruption

Variable description

Here is a description of all the variables used in our analysis, and why they can be relevant.

Economic and Demographic:

e_area: **Land area** - Measures the total area of land within a specific geographical region or country, providing insights into the physical expanse available for various economic, agricultural, and developmental activities.

e_gdppc: **GDP per capita** - A measure of a country's economic output per person, providing insights into the standard of living and economic development.

e_miinflat: **Inflation** - The rate at which the general price level of goods and services rises, affecting purchasing power and the cost of living.

e_total_resources_income_pc: **Petroleum, coal, natural gas, and metals production per capita** - Indicates the availability and utilization of natural resources for economic development.

e_pop: **Population** - The total number of people living in a country, influencing economic growth and resource allocation.

e_pelifeex: **Life expectancy** - The average number of years a person is expected to live, reflecting the overall health and living conditions in a country.

v2clstown: **State ownership of the economy** - Measures the extent of government or state control over economic activities and industries within a country.

Political Governance:

v2x_libdem: **Liberal democracy index** - A measure of the extent to which democratic principles are upheld in a country, including individual rights and freedoms.

v2x_egaldem: **Egalitarian democracy index** - A measure of the extent to which democratic institutions promote equal representation and participation among citizens.

v2elfrfair: **Free and fair elections** - Indicates the integrity and fairness of electoral processes in determining the government.

v2xnp_pres: **Presidentialism Index** - Indicates the concentration of power in the presidency within a democratic system.

v2svdomaut: **Domestic autonomy** - Reflects the independence and autonomy of a country's domestic affairs.

v2svstterr: **State authority over territory** - Indicates the government's control over its territorial boundaries.

Social Equality and Rights:

v2clrelig: **Freedom of religion** - Indicates the level of religious freedom and tolerance in a country.

v2clfmove: **Freedom of foreign movement** - Measures the freedom of citizens to travel and move across borders.

v2pepwrses: **Power distributed by socioeconomic position** - Reflects the equitable distribution of power among different socioeconomic groups.

v2peedueq: **Educational equality** - Measures the level of equality in access to education across different groups.

v2peasbecon: **Access to state business opportunities by socioeconomic position** - Indicates the accessibility of economic opportunities to different socioeconomic groups.

v2cagenmob: **Mass mobilization** - Measures the ability of citizens to mobilize and participate in collective actions, such as demonstrations, strikes and sit-ins.

v2catrauni: **Engagement in independent trade unions** - Reflects the degree of freedom for citizens to engage in trade unions and labor movements.

Corruption:

v2exbribe: **Executive bribery and corruption exchanges** - Measures the prevalence of corruption within the executive branch of government.

v2xnp_regcorr: **Regime corruption** - Measures corruption within the political regime.

v2excrptps: **Public sector corrupt exchanges** - Measures the extent of corruption within the public sector.

v2mecorrpt: **Media corruption** - Measures corruption within the media sector.

e_wbgi_cce: **Control of corruption** - Measures the effectiveness of controlling corruption in a country.

The analysis begins

Here we gather the data in order to start analyzing it.

```
## The original data frame is to large to run in git, therefore the first wrangling is don

## Loading in data
# load("V-Dem-CY-Full+Others-v13.rds")
```

Selecting the relevant variables and observations

The full V-Dem data frame consists of 27555 observations of 4602 variables. We will therefore start by filtering out countries and time periods that are not relevant.

We will filter out years before 2005 because we do not have the right data available for the later analysis before 2005, and those years are therefore not relevant.

We will then select the variables we want to analyze.

```
#This was also done in the local file

## Taking out countries I need for my analysis
# data <- data %>%
#   filter(country_name %in% countries)


## Filtering out years before 2005
# data <- data %>%
#   filter(year >= 2005)


## Choosing variables to include in analysis
# data <- data %>%
#   select(country_name, year, v2x_libdem, v2x_egaldem, v2elfrfair, v2exbribe, v2excrptps,

## Create the files with the relevant data, which is then moved to the github folder
# save(data, file = "relevant_data.rds")
```

```
# Loading in the file that only includes the relevant variables
load("relevant_data.rds")

# For display purposes
data %>%
  select(country_name, year, v2x_libdem) %>%
  head()
```

```
  country_name year v2x_libdem
1       Sweden 2005      0.885
2       Sweden 2006      0.888
3       Sweden 2007      0.892
4       Sweden 2008      0.892
5       Sweden 2009      0.892
```

```
6       Sweden 2010      0.892
```

Creating the mean

In order to compare all the countries we need to let each country only have one value per variable, not one for each year.

We will then need to calculate the mean for each country and variable over all years.

```
# Calculate the mean for each variable for each country
df_means <- data %>%

  # Put all the years in a group for each country
  group_by(country_name) %>%

  # Calculate the mean of each of the variable of all the years
  summarise(across(.cols = c("v2x_libdem", "v2x_egaldem", "v2elfrfair", "v2exbribe", "v2ex


# For display purposes
df_means %>%
  select(country_name, v2x_libdem) %>%
  head()
```

```
# A tibble: 6 x 2
  country_name v2x_libdem
  <chr>             <dbl>
1 Austria           0.778
2 Belgium           0.827
3 Bulgaria          0.574
4 Croatia           0.669
5 Cyprus            0.714
6 Czechia           0.788
```

Now we have a data frame that only has one row per country, with all the variables in the columns.

Data wrangling

In order for the next steps to work we need to omit the country names.

The first data frame is the same as over, just excluding the country names.

The others only have the variables in each category as described in the variable description above, also excluding country names.

```
# All dataframes are created to exclude country names for the next step

# All variables
df_noname <- df_means %>%
  select(v2x_libdem, v2x_egaldem, v2elfrfair, v2exbribe, v2excrptps, v2clrelig, v2clfmove,

# All variables in corruption category
df_corr <- df_means %>%
  select(v2exbribe, v2xnp_regcorr, v2excrptps, v2mecorrpt, e_wbgi_cce)

# All variables in economic and demographic category
df_econ_dem <- df_means %>%
  select(e_area, e_gdppc, e_miinflat, e_total_resources_income_pc, e_pop, e_pelifeex, v2cl

# All variables in political governance category
df_poli_gov <- df_means %>%
  select(v2x_libdem, v2x_egaldem, v2elfrfair, v2xnp_pres, v2svdomaut, v2svstterr)

# All variables in social equality and rights category
df_soceq_right <- df_means %>%
  select(v2clrelig, v2clfmove, v2pepwrses, v2peedueq, v2peasbecon, v2cagenmob, v2catrauni)
```

Creating rankings

In order to create the ranking system, we need to compare each value to Norway's value.

We will calculate the difference in the values between the countries and Norway. Then we will rank all the countries for each variable by how far away their value is from Norway's value. Then we have a ranking by each variable.

For example the country that is most similar will receive the score 2 (since Norway will be no. 1).

In order to create a total ranking we will add up all the rankings, and give each country a total score. In the total ranking, we rank by 25 different variables, and then the lowest possible score is 25 (this will be given to Norway).

In the end, the countries with the lowest scores will be the most similar to Norway.

After receiving a score we will combine the data frames with the vector containing country names and be able to see which countries have what score.

Category ranking

Each category will also be ranked in the exact same way (using the same function), and receive a ranking based only on all the variables in each category.

```r
# The function takes a dataframe 's' as input and calculates the ranking of each element i

final_ranking <- function(s) {

  # Calculate the ranking for each column
  df_rank <- apply(s, MARGIN = 2, FUN = function(x){

    # Store the 22th element in the column (null point)
    null_point = x[22]

    # Calculate the absolute difference between each element and the null point
    y <- abs(x - null_point)

    # Calculate the rank of each element based on the absolute differences
    z <- rank(y)
    return(z)
})

  # Convert the ranking matrix into a data frame
  df_rank <- as.data.frame(df_rank)

  # Calculate the sum of ranks for each row (country) in the data frame
  df_rank$sum_rank <- apply(df_rank, MARGIN = 1, FUN = function(x){
  y <- sum(x, na.rm = TRUE)
})

  # Combine the calculated ranks with the 'countries' vector
  df_rank_f <- df_rank %>%
  cbind(countries)

  # Return the final data frame with the calculated ranks and countries
  return(df_rank_f)
}
```

```
# Create rankings

# Create ranking based on all factors
total_rank <- final_ranking(df_noname)

# Create ranking based on corruption category
corr_rank <- final_ranking(df_corr)

# Create ranking based on economic and demographic category
econ_dem_rank <- final_ranking(df_econ_dem)

# Create ranking based on political governance category
poli_gov_rank <- final_ranking(df_poli_gov)

# Create ranking based on social equality and rights category
soceq_right_rank <- final_ranking(df_soceq_right)
```

Display the rankings

Here each of the rankings will be displayed.

```
display <- function(df) {

  df %>%
    select(countries, sum_rank) %>%
    filter(countries != "Norway") %>%
    arrange(sum_rank) %>%
    head(10)
}

# Display the ranking of the corruption factors
display(corr_rank)
```

```
     countries sum_rank
1      Germany       24
2   Luxembourg       24
3  Netherlands       29
4      Finland       31
5       Sweden       33
6      Iceland       39
```

```
7          Belgium       42
8          Ireland       46
9          Estonia       49
10 United Kingdom       51
```

```
# Display the ranking of the political governance factors
display(poli_gov_rank)
```

```
        countries sum_rank
1         Denmark     46.0
2         Iceland     57.0
3          Sweden     58.0
4     Netherlands     61.0
5      Luxembourg     63.5
6         Belgium     64.0
7         Finland     65.0
8         Estonia     71.0
9  United Kingdom     77.0
10        Czechia     80.5
```

```
# Display the ranking of the social equality and rights factors
display(soceq_right_rank)
```

```
      countries sum_rank
1       Finland       50
2        Sweden       51
3    Luxembourg       55
4       Denmark       65
5       Iceland       79
6       Ireland       89
7   Netherlands       90
8      Slovenia       91
9       Belgium       98
10      Czechia       99
```

```
# Display the ranking of the economic and demographic factors
display(econ_dem_rank)
```

```
          countries sum_rank
1           Finland       56
2           Denmark       66
3            Sweden       67
4           Austria       75
5           Ireland       77
6       Netherlands       95
7    United Kingdom       99
8             Italy      101
9             Spain      108
10           Poland      109
```

```
# Display the total ranking
display(total_rank)
```

```
      countries sum_rank
1       Finland    202.0
2        Sweden    209.0
3       Denmark    231.0
4    Luxembourg    262.5
5   Netherlands    275.0
6       Iceland    289.0
7       Ireland    299.5
8       Belgium    340.0
9       Germany    345.0
10      Estonia    346.0
```

We can then see that Finland is the most similar in several of the categorical rankings as well as in the total ranking. We can also see that the most similar countries in descending order based on all factors are: Finland, Sweden, Denmark, Luxembourg, Netherlands, Iceland, Ireland, Belgium, Germany and Estonia.

Save the tables

The ranking tables are saved as excel files, that are available to look at more closely. It is there possible to look at all the countries and how they score on all variables as well as how they score in each of the categories.

```
library(openxlsx)
```

```r
# Saving the data frames to their own excel files

# All the imported data
write.xlsx(data, file = "data.xlsx")

# Data frame with one value per country per variable
write.xlsx(df_means, file = "means.xlsx")

# Data frames with the rankings for all and categorical variables
write.xlsx(total_rank, file = "total_rank.xlsx")
write.xlsx(corr_rank, file = "corruption_rank.xlsx")
write.xlsx(econ_dem_rank, file = "economic_and_demographic_rank.xlsx")
write.xlsx(poli_gov_rank, file = "political_governance_rank.xlsx")
write.xlsx(soceq_right_rank, file = "social_equality_rank.xlsx")
```

MAIN SOURCES OF ECONOMIC INFORMATION

NAMA (*National Accounts Main Aggregates*) and SBS (*Structural Business Statistics*) are two data frames that provide information on economic activities, specifically value added at different stages of production.

NAMA - National accounts aim to capture economic activity within the domestic territory. They combine data from a host of base statistics, and thus they have no common sampling reference frame. Indicators are calculated as **Value added at Gross (VAG)** i.e. it includes the total value of goods and services produced minus the cost of intermediate inputs.

SBS - It describes the structure, conduct and performance of economic activities, down to the most detailed activity level (*several hundred economic sectors*). Indicators are calculated as **Value added at Factor Cost** i.e. it takes into account the gross income from operating activities after adjusting for operating subsidies and indirect taxes. Income and expenditure classified as financial is excluded from value added.

Why is it hard to measure A?

This means that some categories, such as (A) - Agriculture, Forestry, and Fishing are hard to quantify accurately due to challenges in determining the value added in certain agricultural activities, especially in cases of vertical integration. The breakdown of value added in such cases can be difficult, leading to classification issues.

THE PROBLEM - INCOMPARABLE INDICATORS !!

NAMA has categories that SBS does not have. But SBS uses value added at factor cost, while NAMA uses value added at gross. We tried to compare the two data frames in order to establish relationships for crucial understanding of economic trends and making informed decisions by EU nations. Since both these data frames follow NACE rev. 2 definition of industrial structure, the analysis of the difference between the two data frames was to provide valuable insights into their compatibility & suitability to use them as golden sources for key statistical information on EU nations.

Eurostat does not normally track Agriculture and a number of other categories together with the main structural business statistics categories.

Hence, on the outset, there were only 4 categories that are included in both data frames:

1. Manufacturing
2. Construction
3. Information and Communication
4. Real Estate Activities

PROPOSED APPROACH

More categories were found in NAMA dataframe with slightly different definition of the indicators, so we couldn't compare them right away. Proposed approach was to conduct data wrangling and compare various combinations of averages in relation to the common economic indicators in both data frames, understand variations, perform calculations for all the economic indicators and understand how each country's economic performance is compared to Norway's performance across different indicators.

ANALYSIS Initiated - trying to fit it in from different data sets (how or why it wont work). Analysis begins....

Loading relevant libraries

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

Selecting the relevant variables and observations

We look at the data between 2013 till 2020 to compare common categories with slightly different indicators to see if they are comparable.

```
#Defining data directory & loading raw data from Excel into dataframe "data"
data <- read_excel("SBS NAMA data.xlsx", sheet=2)
data$year2013 <- as.numeric(data$year2013, na.exclude=T)
class(data$year2013)
```

```
[1] "numeric"
```

```
data$year2014 <- as.numeric(data$year2014, na.exclude=T)
data$year2015 <- as.numeric(data$year2015, na.exclude=T)
data$year2016 <- as.numeric(data$year2016, na.exclude=T)
data$year2017 <- as.numeric(data$year2017, na.exclude=T)
data$year2018 <- as.numeric(data$year2018, na.exclude=T)
data$year2019 <- as.numeric(data$year2019, na.exclude=T)
data$year2020 <- as.numeric(data$year2020, na.exclude=T)
```

Data frame Filtration

Filtering NAMA data frame - into data frame “nama” and pivot it to a long format to organize the years and value columns.

```
#Filter NAMA data into dataframe "nama"
nama <- data %>%
  filter(Dataset == "NAMA") %>%
  pivot_longer(cols = starts_with("year"), names_to = "year", values_to = "value_nama")
```

Filtering SBS data frame - into data frame “sbs” and pivoting it to a long format to organize the years and value columns

```
#Filter SBS data into dataframe "sbs"
sbs <- data %>%
  filter(Dataset== "SBS") %>%
  pivot_longer(cols = starts_with("year"), names_to = "year", values_to = "value_sbs")
```

Merging data frames - based on Countries, year, and Category (common denominators) & arrive at:

- Difference between NAMA and SBS values
- Average values
- % differences

```
#Merge NAMA and SBS datasets & save a copy
df <- nama %>%
  left_join(sbs, by = c("Countries", "year", "Category"))
df$tot_diff <- df$value_sbs-df$value_nama
df$average_value <- (df$value_nama+df$value_sbs)/2
df$percentage_diff <- ((df$value_sbs - df$value_nama)/df$value_sbs)*100
df$year <- gsub(pattern = "year",
                replacement = "",
                df$year )
```

```r
save(df, file = "nama_sbs_comparison.rda")
```

Analysis of the Difference

Now we will analyze the differences between the NAMA and SBS data frames after merging them. By comparing these values, we aim to understand the variations in measurement and the compatibility of the data frames.

Step 1: Calculate mean to draw single value representing the average for each indicator for that country. This summary helps us to identify the central tendencies and general patterns in the data for each country and indicator.

```r
# Calculate the mean for each Indicator for each Country
grouped_df <- df %>%
  group_by(Countries)
mean_df <- grouped_df %>%
  summarize(mean_value_nama = mean(value_nama, na.rm = TRUE),
            mean_value_sbs = mean(value_sbs, na.rm = TRUE),
            mean_tot_diff = mean(tot_diff, na.rm = TRUE),
            mean_average_value = mean(average_value, na.rm = TRUE),
            mean_percentage_diff = mean(percentage_diff, na.rm = TRUE))
```

Step 2: We aggregate the data to obtain summary statistics for each country over all the years. This allows us to analyze and compare the performance of each country across different economic indicators over time.

```r
# Group years for each Country
grouped_df_2 <- df %>%
  group_by(Countries, year)
mean_df_2 <- grouped_df_2 %>%
  summarize(mean_value_nama = mean(value_nama, na.rm = TRUE),
            mean_value_sbs = mean(value_sbs, na.rm = TRUE),
            mean_tot_diff = mean(tot_diff, na.rm = TRUE),
            mean_average_value = mean(average_value, na.rm = TRUE),
            mean_percentage_diff = mean(percentage_diff, na.rm = TRUE))
```

Step 3: We obtain a single average value that represents the overall performance of that indicator across all the years for all countries.

```r
# Calculate mean for each Indicator for all years
grouped_data <- data %>%
  group_by(Countries,Dataset)
```

```
mean_df_3 <- grouped_data %>%
  summarize(mean_year2013 = mean(year2013, na.rm = TRUE),
            mean_year2014 = mean(year2014, na.rm = TRUE),
            mean_year2015 = mean(year2015, na.rm = TRUE),
            mean_year2016 = mean(year2016, na.rm = TRUE),
            mean_year2017 = mean(year2017, na.rm = TRUE),
            mean_year2018 = mean(year2018, na.rm = TRUE),
            mean_year2019 = mean(year2019, na.rm = TRUE),
            mean_year2020 = mean(year2020, na.rm = TRUE))
```

CONCLUSION

We are not going to merge the data from both the data frames as both are not comparable.

Looking at the results, 2/3 are further away than 10% from the mean, as such the two data frames are not comparable.

Creating The Dashboard

Downloading and importing data is a critical step in any data analysis project. In this section, we'll explain how we retrieved the necessary data from Eurostat's database and transformed it into a dataframe for further analysis.

Before diving into the code, we identified the specific data requirements for our project. We needed economic indicators, such as "Value added at factor cost - million euro" (V12150) and "Persons employed - number" (V16110), for a set of countries and NACE industry classifications. To access this data, we utilized Eurostat's query builder that to give us the desired indicators, countries, and time period. We additionally tested out the Eurostat API, but in the end we ended up using the Eurostat query builder.

```
### Eurostat Query Builder
# Eurostat data code:SBS_SC_SCA_R2
# Eurostat query builder link: https://ec.europa.eu/eurostat/web/query-builder/tool

# Time option: since 2005
# INDIC_SB:
  # V12150 Value added at factor cost - million euro
  # V16110 - Persons employed - number
# SIZE_EMP : TOTAL
# NACE_R2 : B-N_S95_X_K and all main sections divisions and groups

## Download data through URL
```

```
url <- "https://ec.europa.eu/eurostat/api/dissemination/statistics/1.0/data" #Removed full
getresponse <- GET(url)
json <- content(getresponse, as = "text")
writeLines(json, "Eurostat SBS Data.json")
```

```
## Importing json file and saving it as df
df <- fromJSONstat("Eurostat SBS Data.json")

# Removing scientific number format
options(scipen=999)

###Eurostat API
## Note: Initialy importing the file through API is alot slower than using the Eurostat qu
# comparable_countries_full <- c("Norway", "Finland", "Sweden", "Denmark", "Luxembourg", "
# comparable_countries_short <- c("NO", "FI", "SE", "DK", "LU", "NL", "IS", "IE", "BE", "D
# api <- get_eurostat("sbs_sc_sca_r2",
#   filters = list(size_emp = "TOTAL",
#                   geo = comparable_countries_short),
#                  indic_sb = c("V12150", "V16110"), # This line does not work, no clue why
#   cache = FALSE,
#   time_format = "num")
```

Data Cleaning and Transformation

In the “Wrangling Data” section, we focused on cleaning and reorganizing the data in the dataframe df1 to create a new and more structured dataframe named df2. This process involved several steps to ensure that the data was in a more suitable format for analysis and visualization.

In this step, we performed data cleaning and transformation tasks to prepare the data for analysis. Here’s a summary of the actions taken:

- **Removing Unnecessary Columns:** We removed the columns “Time frequency” and “Size classes in number of persons employed” as they contained redundant information using the distinct() function.
- **Renaming Columns:** For clarity and consistency, we renamed certain columns using the rename() function.
- **Handling Fake Duplicates:** To address fake duplicates in the data, we kept only one entry for each unique combination of “category,” “country,” and “year” using the distinct() function with the .keep_all = TRUE argument.

- **Transforming Data with pivot_wider():** We used the pivot_wider() function to create new columns for different indicators, such as “value_added” and “employed,” reducing the number of rows in the dataframe.
- **Fixing Year Format:** We converted the “year” column from characters to a numeric format using the as.numeric() function.
- **Setting Negative Value Added to Zero:** Negative values in the “value_added” column were replaced with zeros using the mutate() function and the ifelse() statement.

These steps resulted in a cleaned and structured dataframe, **df2** ready for further analysis and visualization.

```r
### Removing and renaming colomns
# Removing duplicates
df1 <- df %>%
  distinct()
df1$`Time frequency` <- NULL
df1$`Size classes in number of persons employed` <- NULL
df1 <- df1 %>%
  rename(category = "Statistical classification of economic activities in the European Com
  indicator = "Economical indicator for structural business statistics",
  country = "Geopolitical entity (reporting)",
  year = "Time",
  value_added1 = "value")

### Find and remove fake duplicates with one NA and one non NA value
df1 <- df1 %>% distinct(category, country, year, indicator, .keep_all = TRUE)


### Make the indicators into new Columns to cut down number of observations
df2 <- df1 %>%
  pivot_wider(names_from = indicator, values_from = value_added1)

df2 <- df2 %>%
  rename(value_added = "Value added at factor cost - million euro",
  employed = "Persons employed - number")

# Fix year to be numeric
df2 <- df2 %>%
  mutate(year = as.numeric(year))

# Set negative value added to equal 0
```

```
df2 <- df2 %>%
  mutate(value_added = ifelse(value_added < 0, 0, value_added))
```

Handling Missing Values - Interpolation

In our dataset, we encountered missing values in the “value_added” and “employed” columns for certain years and categories. The years 2005 to 2008 had a significantly higher number of missing observations compared to other years. We needed to address these missing values before proceeding with our analysis to ensure the accuracy and completeness of the data.

To handle missing values, we explored various methods but encountered limitations with each:

- **Mice and missForest:** Require a substantial amount of complete data, not suitable for our dataset with numerous missing observations.
- **Linear Regression::** Assumes linearity and may not be robust to outliers or non-linear trends.
- **Mean and Median::** Simple imputation may lead to biased estimates and overlook variations between categories and countries.

As a better approach, we chose two interpolation methods:

- **na.locf (Last Observation Carried Forward):** : Imputes missing values with the last known non-missing value within the same category and country, suitable for sequences of missing values.
- **na.approx:** : Performs linear interpolation between non-missing values, estimating missing observations based on a roughly linear trend.

We used these interpolation techniques to create the cleaned dataframe **df3**, which now contains more complete data while preserving overall trends. This dataset is now ready for further analysis and visualization in our project. It also keeps track if a value has been edited through the Yes/No variable estimated.

```
### Interpolation see the trend
#Lising what nations to be kept
comparable_countries_full <- c("Norway", "Finland", "Sweden", "Denmark", "Luxembourg", "Ne

#comparable_countries_full <- c("Norway")

# Create new dataset keeping only comparable countries
df3 <- df2 %>%
  filter(country %in% comparable_countries_full)

# Count number of NAs in 'value_added' for each year
df3_na_counts <- df3 %>%
  group_by(year) %>%
  summarize(na_count = sum(is.na(value_added)))
# Here we see that there is a significant higher amount of NA observations in 2005-2007. A

# Filtering data for years after 2007, removing 'value_per_employee' column,
# converting 'category' and 'country' into factors, and creating NA markers
df3 <- df3 %>%
  filter(year > 2007) %>%
  mutate(
    category = as.factor(category),
    country = as.factor(country),
    value_added_na = ifelse(is.na(value_added), 1, 0),
    employed_na = ifelse(is.na(employed), 1, 0),
    estimated = ifelse(value_added_na == 1 | employed_na == 1, "Yes", "No")
  )

# Grouping by 'country' and 'category', sorting by 'year',
# filling NA with linear approximations or the last non-NA (from the right)
df3 <- df3 %>%
  group_by(country, category) %>%
  arrange(year) %>%
  mutate(
    value_added = ifelse(is.na(value_added), na.approx(value_added, na.rm = FALSE), value_
    employed = ifelse(is.na(employed), na.approx(employed, na.rm = FALSE), employed),
    value_added = ifelse(is.na(value_added), na.locf(value_added, fromLast = TRUE), value_
    employed = ifelse(is.na(employed), na.locf(employed, fromLast = TRUE), employed)
  ) %>%
  ungroup()
```

Adding Letter Codes to Dataset

In this section, we enriched our dataset df3 by adding NACE (Nomenclature of Economic Activities) letter codes, which provide a hierarchical classification of economic activities through the following steps:

- **Importing and Preprocessing NACE Codes:** We imported the NACE codes from a CSV file, converted column names to lowercase, and added an “A” to corresponding sections. We also handled duplicate names and created unified NACE codes that uniquely identify each economic activity.

- **Adding NACE Levels and Total Business Economy Category:** We assigned NACE levels (1 to 3) based on the length of the NACE code. Additionally, we manually added a “total business economy” category at level 0, representing the overall aggregated data.
- **Merging NACE Codes with Main Dataframe:** We merged the NACE codes (nace_codes2) with the main dataframe (df3) based on matching activity names. This added the NACE level and code information to each row in the main dataframe.
- **Reordering and Renaming Columns:** Finally, we rearranged the columns and renamed the “category” column to “name” for clarity.

With these enhancements, our dataset **df3** now contains the NACE letter codes from the sorted **nace_codes2**, allowing for more efficient sorting and operations compared to using the full names as we did previously.

```r
# Import nace files and make names lower case, add A to all coresponding sections
nace_codes <- read.csv("nace codes.csv")
nace_codes1 <- nace_codes %>%
  rename_all(tolower) %>%
  mutate(section = ifelse((section == ""), NA, section)) %>%
  fill(section, .direction = "down")

# Add group value of the subsequent value and divisions as long as class is not NA
for (i in 2:nrow(nace_codes1)) {
  if (!is.na(nace_codes1$class[i])) {
    nace_codes1$group[i] <- nace_codes1$group[i - 1]}
  if (!is.na(nace_codes1$group[i])) {
    nace_codes1$division[i] <- nace_codes1$division[i - 1]}}

# Combine variables to create unified nace_code
nace_codes1$nace_code <- paste(nace_codes1$section, ifelse(is.na(nace_codes1$group), ifels

# Lowercase as thedata set uses different capitalization p1...
nace_codes2 <- nace_codes1 %>% mutate(activity = tolower(activity))

# If a group has same name(activity) as division, then remove group observation
nace_codes2 <- nace_codes2 %>%
  group_by(activity) %>%
  filter((is.na(group) | row_number() == 1) & (is.na(division) | row_number() == 1)) %>% u
```

```
# Removing duplicates and remove unused variables
nace_codes2 <- nace_codes2 %>%
  filter(is.na(class)) %>%
  select(-class, -isicrev..4)

# Add NACE level 1-3 based on length of nace_code
nace_codes2$level <- ifelse(is.na(nace_codes2$division) & is.na(nace_codes2$group), 1,
                            ifelse(!is.na(nace_codes2$division) & is.na(nace_codes2$group)
                                   ifelse(!is.na(nace_codes2$group), 3, NA)))

# Add all NACE code
total_name <- "total business economy; repair of computers, personal and household goods;
total_nace_code <- "B-N_S95_X_K" #set as level 0, manually set this afterwards
nace_codes2 <- rbind(nace_codes2, data.frame(level = 0, nace_code = total_nace_code, secti
  select(level, nace_code, everything())

nace_codes2_example <- nace_codes2 %>%
  mutate(section = ifelse(section == "B-N_S95_X_K", ".B-N_S95_X_K", section)) %>%
  filter(section %in% c(".B-N_S95_X_K", "A", "B"))

### Check for nace_code name(activity), then merge nace_codes 2 variables with main datafr
# Lowercase as the data set uses different capitalization p2...
df3$category_lower <- tolower(df3$category)
# Merge
df3 <- merge(x = df3, y = nace_codes2, by.x = "category_lower", by.y = "activity", all = F

df3 <- df3 %>%
  select(-category_lower)

# Reorder and renaming
df3 <- df3 %>%
  select(year, country, level, nace_code, section, division, group, everything())
df3 <- df3 %>%
  rename(name = category)
```

Adding value added and employe percentages for each level

In this segment, we progress in enhancing our dataset by creating bespoke indicators centered around value added and population metrics. We include value added and employed percentages for each hierarchical level, thereby providing a deeper comprehension of the economic activities within a specific level of a country. This enrichment is achieved through several steps:

- **Creating Value Added and Employee Percentage for each Level:** We set up a new dataframe, df4, and iterate through df3 on each unique level and year. For every country within the current level and year, we compute total value added and total population of employed individuals. From these, we derive value added and population percentages for each level.
- **Computation Value Added per Employee:** Calculate the value added per employee.
- **Creating Growth Rate Indicator:** We determine growth rates for the value added and employed population for each economic activity (nace_code) within each country.

These steps lead to a better **df4** dataframe that includes key metrics for economic activities, aiding in a more detailed analysis.

```
df4 <- data.frame()

# Loop over each level
for(l in unique(df3$level)) {
  for (y in unique(df3$year)) {
  # Filter the data for the current level
  temp_df <- df3 %>%
    filter(level == l & year == y) %>%
    group_by(country) %>%
    mutate(total_value_added_country = sum(value_added, na.rm = TRUE),
           level_value_percentage = ifelse(!is.na(value_added),
                                   (value_added / total_value_added_country)*100,
                                   NA)) %>%
    mutate(total_population_country = sum(employed, na.rm = TRUE),
           level_population_percentage = ifelse(!is.na(employed),
                                   (employed / total_population_country)*100,
                                   NA))
  # Remove unecesary paramters
  temp_df <- temp_df %>% select(-total_value_added_country, -total_population_country)
  # Combine the data
  df4 <- rbind(df4, temp_df)
  }
}

#Calculate value per employe
df4$value_per_employe <- ifelse(df4$employed == 0 | is.na(df4$employed), NA, df4$value_add

# Growth rates
df4 <- df4 %>%
  group_by(country, nace_code) %>%
  arrange(year) %>%
  mutate(value_added_growth_percentage = (value_added/lag(value_added) - 1)*100,
         population_growth_percentage = (employed/lag(employed) - 1)*100) %>%
  ungroup()
```

Preparing Data for Dashboard

In this section, we prepare our dataset for a dashboard, which involves creating a new dataframe (**df5**) and customizing its content for a more user-friendly experience. Here are the key actions performed:

- **Dataset Renaming:** We make df5 and rename its variables into full-length names for enhanced user readability on the dashboard.
- **Combining Codes and Names:** We merge the **`nace_code`** and **`name`** to provide more context for the user of the dashboard later.
- **Creating Dropdown Lists:** We generate dropdown lists for 'Year' and 'Indicators'. These interactive lists will provide users with the ability to select specific years and economic indicators for analysis.
- **Level-wise Data Subset:** We create subsets of df5 for each level (0 to 3), sorting unique names within each. These subsets will aid in displaying data pertaining to individual levels on the dashboard.

Through these actions, we ensure our dataset is tailored to fit into an interactive, user-friendly dashboard.

```
# Renaming dataset variables to full, easily understandable, names. Also combining nace_co
df5 <- df4

df5 <- df5 %>%
  rename("Value Added (in Million Euro)" = value_added,
         "Number of Employed" = employed,
         "Value Added per Employee (in Million Euro)" = value_per_employe,
         "Percentage of Value Added (GDP)" = level_value_percentage,
         "Percentage of Employed" = level_population_percentage,
         "Country" = country, "Year" = year) %>%
  mutate(name = paste0("(", nace_code, ") ", name))

# Saving as RDS
saveRDS(df5, file = "df5_data.rds")
```

```r
## Defining dropdown list options
year_choice <- unique(df5$Year)
indicators <- c("Value Added (in Million Euro)",
               "Number of Employed",
               "Value Added per Employee (in Million Euro)",
               "Percentage of Value Added (GDP)",
               "Percentage of Employed")

# Creating subset of df5 for each level
nace_names_level0 <- sort(unique(subset(df5, level == 0)$name))
nace_names_level1 <- sort(unique(subset(df5, level == 1)$name))
nace_names_level2 <- sort(unique(subset(df5, level == 2)$name))
nace_names_level3 <- sort(unique(subset(df5, level == 3)$name))
```

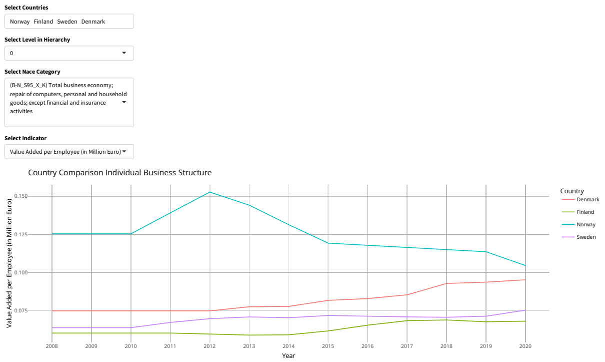
Dashboard

In this section, we focus on the construction of an interactive Shiny dashboard. The dashboard is designed to allow users to explore the changes in the industrial structure in Norway and other comparable countries over time.

- **User Interface (UI):** The UI is developed using **dashboardPage** to construct a page that includes a header, sidebar, and body. The header is titled "Business Structure". The sidebar houses the menu items, "Business Structure over Time" and "Business Structure Comparison".
- **Server:** The server component is where the interactive aspects of the Shiny app are handled. Here, we define what should happen when a user changes an input in the UI. We have a **renderUI** function that dynamically changes the options in the NACE category selector based on the selected hierarchy level.
- **Running the Dashboard:** Finally, we use **`shinyApp`** to combine the UI and server components, creating and launching the Shiny application.

This interactive dashboard presents the data in a comprehensive and dynamic manner, allowing users to customize their view of the business structure over time and conduct country comparisons effectively.

```r
### Changes in The Industrial Structure in Norway and Comparative Countries
## Making UI
ui <- dashboardPage(
  dashboardHeader(title = "Business Structure"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Business Structure over Time", tabName = "tab2"),
      menuItem("Business Structure Comparison", tabName = "tab3")
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(tabName = "tab2",
        selectInput("countries2", "Select Countries", choices = comparable_countries_full,
        selectInput("level2", "Select Level in Hierarchy", choices = c(0,1,2,3), selected
        uiOutput("ui_select_name2"),
        selectInput("indicator2", "Select Indicator", choices = indicators, selected = "Va
        plotlyOutput("plot2", height = "auto"),
      ),

      tabItem(tabName = "tab3",
        selectInput("country_choice3_1", "Select Country 1", choices = comparable_countrie
        selectInput("country_choice3_2", "Select Country 2", choices = comparable_countrie
        selectInput("year3", "Select Focus Year", choices = year_choice, selected = 2020,
        selectInput("indicator3", "Select Indicator", choices = indicators, selected = "Pe
        plotlyOutput("plot3")
      )
    )
  )
)

## Making Server
server <- function(input, output) {
    output$ui_select_name2 <- renderUI({
    switch(input$level2,
      "0" = selectInput("name2", "Select Nace Category", choices = nace_names_level0),
      "1" = selectInput("name2", "Select Nace Category", choices = nace_names_level1),
      "2" = selectInput("name2", "Select Nace Category", choices = nace_names_level2),
      "3" = selectInput("name2", "Select Nace Category", choices = nace_names_level3)
    )
  })

output$plot2 <- renderPlotly({
  # Ensures that input$name2 is available before proceeding
  req(input$name2)
  df5_compare <- df5 %>%
    filter(Country %in% input$countries2, level == input$level2, name == input$name2)

  p2 <- ggplot(df5_compare, aes(x = Year, y = !!rlang::sym(input$indicator2), colour = Cou
    geom_line() +
    scale_x_continuous(breaks = seq(2008,2020,1)) +
    labs(title = "Country Comparison Individual Business Structure",
         x = "Year",
         y = input$indicator2) +
    theme_minimal()

  ggplotly(p2, tooltip = c("x", "y", "colour", "text"))
})

  output$plot3 <- renderPlotly({
  # Defining the color palette with 2 distinct colors
    palette3 <- c("#1a9d49", "#274247")
    df5_sub3 <- subset(df5, level == 1 & Year == input$year3 & Country %in% c(input$countr

    p3 <- ggplot(df5_sub3, aes(x = factor(nace_code, levels = rev(sort(unique(nace_code)))
      geom_col(position = "dodge", width = 0.6) +
      coord_flip() +
      theme_minimal() +
      scale_fill_manual(values = palette3) +
      labs(
        title = paste("Country Business Structural Comparison in", input$country_choice3_1
        y = input$indicator3,
        x = "NACE Categories",
        fill = "Country"
      ) +
      theme_minimal()
    ggplotly(p3, tooltip = c("y", "name"))
})
}

## Running Dashboard
shinyApp(ui, server)
```


Select Countries

Norway Finland Sweden Denmark

Select Level in Hierarchy

0

Select Nace Category

(B-N_S95_X_K) Total business economy; repair of computers, personal and household goods; except financial and insurance activities

Select Indicator

Value Added per Employee (in Million Euro)

Interactive Dashboard Link

https://magne.shinyapps.io/ISSSV1337_2023_SSB/

Conclusion

We have not created a detailed analysis on the development in comparative countries. We encourage you to look at the dashboard to see for yourself how the different industries have developed over time.

Observations

In the dashboard it is possible to see some trends and differences between countries. Here are some of our observations from the visualizations.

Value added per employee in 2012-2014

If you look at the trends of value added per employee in the years 2012 to 2014, it is easily observable that the value added per employee went up quite a lot in those years. That is most likely as a result of the oil price going up in that period.

We can therefore see that the oil price has a significant effect on the Norwegian economy. This same increase was not observed in other countries and we can therefore conclude that the oil price is much more important for the Norwegian economy than other similar countries.

It as also visible from this graph, that Norway consistently has a much higher value added per employee. This means that each Norwegian worker brings in a lot more money on average than workers from other countries. Also this is probably as a result of the oil industry. It is incredibly lucrative, and therefore brings in so much money that the average is very high.

Scientific research and development (M72) in Norway

When looking at the percentage of employed in the sector scientific research and development, we can see that Norway has around 0.6% employed in this sector. The other countries generally have between 0.2% and 1%. We can therefore conclude that Norway is employing a comparable amount of people to the other countries.

Percentage employed in mining and quarrying (B)

In Norway we have around 4% of the working people employed in mining and quarrying. The other countries have between 0.1% and 1.5% employed in the same sector. In Norway a lot of people are employed in the oil industry, and probably accounts for a large percentage of this.

The world is focusing more on green energy and will probably move away from oil in the future. In addition to that the oil reserves will eventually run dry. With such a large percentage employed in this industry, we can see that Norway is more vulnerable than other countries for people loosing their jobs when we move to greener energy.