Analysing UK’s Energy Consumption

submitted by

Candidate number: 241912

A Dissertation Submitted to Business School University of Exeter

Analysing UK's Energy Consumption

Executive summary

UK's energy consumption is mainly contributed by four sectors namely Transportation, Industrial, Domestic and Services. On an average, transportation and domestic sector uses approximately more than 60% of total energy in the nation. The services sector uses the least amount of energy when compared to the other three sectors while the industrial sector seems to follow a consistently decreasing trend in energy consumption. In 2021 UK consumed 128213.5 thousand tonnes of oil equivalent energy in total, excluding the energy consumed which is not under these prime four sectors. Based on the analysis of data on energy usage, we hope to better understand the UK's energy situation while simultaneously looking for ways to make it better. A timeline from 2000 to 2021 reveals that throughout the previous two decades, energy consumption had gradually declined (Fig a). The UK used the least amount of energy overall during 2020.

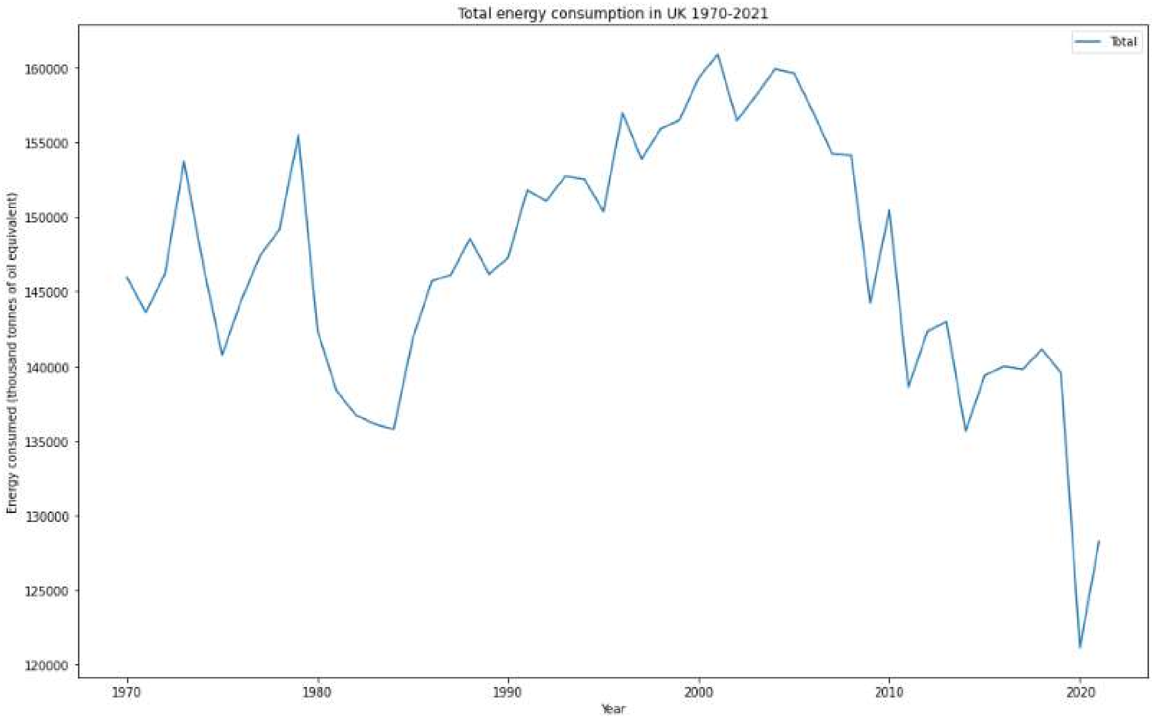


Fig a: Total energy consumption in UK timeline, 1970-2021

The gradual decline in energy consumption was discussed widely as a result of improved energy efficiency as well as the decrease in industries. Over the years improvement in technology as well as social awareness on energy conservation might be the underlying reasons for this improvement in efficiency. While these might be the reasons for the decline in energy usage, the 2020 drop in consumption was mostly influenced by COVID-19 pandemic. During these pandemic years of 2020 and 2021, transportation sector as well as service sectors took noticeable drops in energy consumption. This was mostly due to the fact that lockdowns and restrictions put forward by the government had huge impacts in these sectors. Beyond examining a broad picture of the nation's energy use, we enhanced the study by analysing each sector in detail and identifying the different variables that affect it.

The domestic sector mainly consumes natural gas followed by electricity. However, the use of natural gas was at least twice as much higher than electricity for the past two decades. Keeping these findings in mind it was recommended to shape policies that can decrease reliance on natural gas by promoting renewable sources of energy such as bioenergy. While we explored factors affecting consumption, domestic sector did not show a noticeable relation with annual mean temperatures or the growing population. However, other papers discussed that energy consumption in a residential building is strictly related to the type of building, age of building etc. This means that older buildings may be less prone to acquiring higher energy efficiency. Policies such as EPC (Energy Performance Certificate) helps keep track of energy efficiency measure within different buildings, hence the support for such measures can be continued and improved over time. The services sector also used natural gas and electricity as its prime sources of energy consumption. In 2021, approximately 67.2% energy comes from the commercial sectors followed by 25.6% from public administrations and 7.2% from agriculture services. Even through the services sector made up more than 70% UK's GDP there was no evident relationship between GDP and energy consumption in services sector. However, it was noticed that GDP showed a relation with public administration stating that whenever GDP was declining public administration increased its energy consumption. This was also backed up by historical evidence that public administration ramped up its operations whenever there was an economic decline. The industrial sector showed strong decline in consumption mainly due to the disappearing coal industries and also possibly due to reliance on imported goods and offshore industrialisation. Natural gas, electricity and petroleum products contributes to most of the energy used in industries. The industrial energy consumption was found to have a strong positive correlation with emissions, implying that when emissions went down industrial energy consumption also decreased. Policies that support UK's net-zero targets' forces industries to adapt to more energy efficient and eco-friendly alternatives which can help keep energy consumption at a consistent low. The transportation sector is the highest energy consuming sector over the years by an average of 38% total energy consumed when considering from 2000-2019. Within the transportation sector approximately about 80.7% of total energy consumed comes from road petroleum and 11.8% from air petroleum. 2020 and 2021 did impact the operations on both these modes of transportation and hence showed record low values in consumption. Since there is a clear domination in consumption from road fuel this variable was studied further. The number of new vehicles registered in Great Britain showed a positive correlation with road fuel. This is only because of the fact that most vehicles that are registered are still under fossil fuel-based engine. An increase in electric vehicles or CNG based engines will help diversify energy consumption in transportation sector. However, when shaping policies its recommended to improve public transport systems or city infrastructures to support cycling or walking to reduce the energy usage in transportation sector.

Forecasting future energy requirements play a vital role in estimating energy demand and ensuring energy availability. Based on other papers on energy forecasting in various scenarios we adapted two different models. A statistical model called ARIMA and a machine learning model called LSTM. While statistical models can be well interpreted machine learning models showed more accuracy. We put this to the test by comparing the two models and calculating their efficiency in forecasting energy consumption for various sectors. As a result of white noise in our data the ARIMA model showed a serious disadvantage as it failed to show variations within the predictions it made. Meanwhile custom build ARIMA models that showed accuracy could not be trusted with predicting future data as it might only perform well in the given test set. This problem could be a result of not having enough data or the unavailability of quarterly energy consumption data. In order to have better accuracy in our forecasts we switched to LSTM model. The model showed improved accuracy scores as well as more deviations in the predicted data similar to the actual data (Fig b).

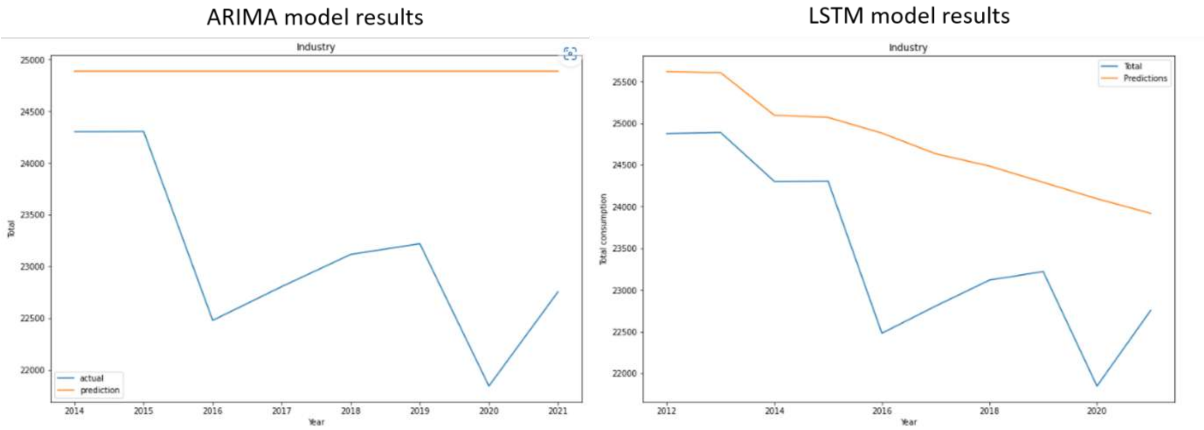


Fig b: Comparison between ARIMA results and LSTM results

The above image is a comparison between the predicated results of these two models in Industrial energy consumption sector. When asked to forecast consumption for the next four years, the LSTM model was observed to perform at its very best. Future studies may be able to identify more directly influencing features which can be put into a multivariant LSTM for further improved accuracy in its predictions.

Table of Contents

1. Introduction

The world has always heavily depended on energy as a resource. Without energy, companies and economies throughout the world cannot function. Energy not only powers our houses and automobiles, but it also influences international relations and the balance of power. The earliest form of energy consumption was through burning wood which generated heat. Surprisingly even now more than two billion people in developing countries still depend on wood as a primary source of energy (Food and Agriculture Organization, 2021). The heat was used in houses to keep the place warm and cook food while services and industries used it in crafting and building materials. Now energy is consumed in more than one forms such as electricity, gas, petroleum fuel etc. Modern energy supplies are diverse and versatile, which creates many challenges and problems in their production, delivery, and escalating demand. The natural resources which are utilized in order to produce consumable energy can be classified as renewable and non-renewable sources of energy. Renewable sources of energy are abundant in nature such as solar, wind and tidal energy, while non-renewable sources such as natural gas, oil and coal are limited and difficult to acquire. Oil is the source of the most energy consumed worldwide, followed by coal, gas, and hydroelectricity (Ritchie & Roser, 2020). The reliance on fossil fuels does have a worrying impact on the environment, since in 2020, coal accounted for 45% of world emissions from fuel combustion, followed by oil at 32% and natural gas at 22%. (IEA, 2021). However, in 2021, there was a 1.4 million rise in global oil output. Countries like Libya, Iran and Canada saw the largest increases while Nigeria, the UK and Angola reported the biggest declines (BP, 2022). Energy as a resource also remains to be a valuable trade commodity and influences relations between countries. Recently, EU nations have voted to establish a $60 limit on the total quantity of Russian oil that may be purchased globally in response to the ongoing conflict between Russia and Ukraine (Foy, 2022). Energy-related global politics have an impact on the economics and national energy security of many countries. Therefore, it is crucial for a country to invest in its energy resources, diversify its energy production, and develop a clear understanding of its consumption and demand trends in order to ensure economic stability and the availability of energy.

UK is a highly developed nation and it relies on energy both domestically produced and imported. According to 2021 figures, UK is the world's 14th largest energy consuming nation (BP,2022). Due to the UK's recent investments in renewable energy, 2020 is the first time in the nation's history that nearly 43% of the electricity generated came exclusively from renewable sources (Grundy, 2021). Despite its efforts in diversifying energy resources and leaning towards energy independence, UK still faces several challenges in managing its energy infrastructure. To keep energy prices from going out of control, the government has already implemented price caps in domestic sector, yet this is no easy solution for a much broader problem (Wait, 2020). There is also a looming threat of possible energy shortages during winter which is also concerning for the nation (N. Thomas, 2022). Imposing increased windfall taxes on energy companies means, these companies will limit their investments and domestic production capabilities (BBC, 2022). These cases still denote that there is room for improvement in UK energy sectors. Policies need to be shaped considering its impact on the nation's energy future while developments in the energy industry should be viable enough to sustain the nation's socioeconomic and environmental integrity. Gaining insights in the energy industry will provide valuable information that can be used in building UK's energy sector. By utilizing data to analyse and predict energy consumption among various sectors will be a stepping stone towards improving the energy sector.

2. Problem statement

Working towards an energy independent future is as important to UK as it for the rest of the world. However, as a result of recent global events, the UK is currently experiencing an energy shortage and a spike in energy prices. By supporting renewable energy development which aligns with the goals of reaching the net zero target by 2050 the nation has committed towards developing diverse energy solutions as well as contribute towards tackling climate change (UK Government, 2020). Yet, there are a multitude of factors which influence the energy sector. Relying primarily on renewable energy sources does not offer a panacea for the current energy crisis because it has its own problems with efficiency and transportation, among many others (TED-Ed, 2017). The creation and modification of rules and regulations that supported energy developments was an often tried and tested strategy for managing the energy industry. One of the many suggestions made to enhance stable pricing has been to deploy energy policies to intervene in the wholesale market pricing (H. Thomas, 2022). We might be able to offer suggestions like these that are supported by empirical data by identifying trends and factors affecting energy demand. The energy industry can benefit from better policies and advancements that are precisely targeted at the issues that need the greatest attention. Rather than to invest the nation's money and time into building possible solutions for the present, analysing and predicting the energy industries different sectors can help build solutions for the future.

As per the national statistics publication "UK ENERGY IN BRIEF 2021" energy consumption can be classified into 4 sectors. These sectors include – Transportation, Domestic, Industries and Services (Department for Business, Energy and Industrial Strategy, 2021). The transportation industry consists of road, rail and air transportations. The domestic industry focuses on housing and non-commercial sectors. The Industrial sector consists of manufacturing and other large or small industrial operations. The services sector includes businesses and other commercial buildings. Each of these sectors consume energy in different manners. The nation-wide energy demand can be represented by studying the consumption trends of all these sectors separately and interpreting the results together. Studying such consumption data may also present us with the opportunity to make recommendations for future policy makers while also being able to setup a framework for future studies in the area.

3. Research questions

Throughout the study we will focus mainly on answering the following questions.

- How can we analyse the existing energy consumption in UK?
  The energy consumption in UK is diverse and complicated. Understanding how energy is distributed across its several key sectors will give us a general insight over the national energy sector and its consumption trends.
- What are the factors influencing each sector?
  There is not a single variable which can accurately define why energy price went up or why consumption has gone low. Understanding the influence on external factors needs to be done sector specifically to validate its impact on the energy sector. There may be several factors which may have an impact on each sector, however finding ones with

numerical historical data will help us back up the claims that these factors are truly impactful in energy consumption.

- How to build a model capable of predicting energy consumption?
  There are several models that can be used for timeline forecasting. We shall test out one statistical model and one machine learning model and compare their results to attain a better one. Rather than focus on the model results we shall focus on the process in order to set a framework of model building that can be followed up for further studies in the energy sector.
- What are the recommendations for the national energy sector?
  Based on the progress of the study we may include sector specific recommendations than could be adapted by policy makers to improve the energy industry or can be adapted for further research in the same area.

4. literature review

Most of the reports regarding the UK's nationwide energy usage is released officially by government administrations, mainly by the Department for Business, Energy and Industrial strategy. These reports are generally released annually and published by the office of National statistics. These reports give detailed explanations of the energy scenario around UK. Energy Consumption in the UK (ECUK) 1970 to 2021 studied energy consumption trend for more than the past 50 years (Department for Business, Energy and Industrial strategy, 2022). This report specifically explains how each sector and their respective sub sectors consumes energy over the years. The report even explores energy consumption by fuel type through its visualisations. However, besides its ability to summarise this information, this report had some obvious flaws as well. The report fails to examine how energy is primarily distributed within these sectors in comparison to the whole energy consumption within the nation. As this is a publicly available report an introductory analysis on overall energy consumption would have made the study much more appealing. Even though the report is titled consumption from 1970-2021, it mostly focuses on recent yearly changes and affects. There is also a lack of forecasting and recommendations which could have improved the value for the report as well as support future studies in the domain. Throughout this study we will focus on rectifying these demerits. More reports such as UK Energy in brief 2022 gives us further robust details in understanding the energy scenario (Department for Business, Energy and Industrial Strategy, 2022). Besides looking at consumption data this report explores more on employment in energy industry, emissions, energy imports and exports among many other topics. This report was used as a main reference source to understand even more about the energy industry and other external factors involved.

During the course of this study, we are heavily reliant on descriptive statistics which can act as a foundation for our exploratory data analysis. Descriptive Statistics and Graphical Displays, is an article which discusses on basic statistical methods that can be adapted into our study to point out useful insights (Larson, 2006). By using a collection of measured BMI data the article explains location statistics such as mean, median, mod as well as dispersion statistics such as standard deviation and variance in a very clear and precise manner. Beside these the article also discusses the application of graphical plots such as histograms and box plots which we may adapt to study our data. An area where the article has not given attention is the description of its statistical results with the real-world scenarios. By explaining outliers, describing patterns of trend or explaining the context of different variables we may utilise statistics to tell us an interesting story.

The demand for energy is influenced by external factors. Many studies tried to establish the connection between such factors while accounting for long term forecasting. the International Institute of Environment and Development in 1979 predicted that while forecasting for the next 50 years, the UK could establish economic growth while reducing the use of energy (Leach, 1979). While analysing the modern energy scenario we find that this prediction from the past has turned out to be true. In fact, understanding such variables such as GDP which represents the economy is crucial towards assessing its relationship with present or future energy demand. However, a prediction that went wrong according to the study was the constant use of coal as an energy resource, which in reality is currently a dying source of energy. Besides considering the long-term forecasting capabilities it is also important to identify sector specific variables that influences energy consumption. The study focusing on household energy consumption in UK successfully aims in discussing several socio-economic and geographical factors that influence energy consumption in domestic sector (Druckman and Jackson, 2008). The paper discusses that dwelling, tenure, household composition and location of the property plays an extremely vital role in how energy is consumed. On a national level, domestic energy consumption was checked to have correlation between household income and emissions to understand their relationship. This can be adapted in our study as finding relationship with external variables is a prime research objective.

Rather than focusing on prediction of a whole nation, a study regarding energy forecasting in a building explains the use of different types of models (Yu et al., 2022). The paper discusses between the choice of 3 model types used for predication. These include White-box models, Black-box model and Grey-box model. The Grey-box models are a combination of Black-box and White-box. The paper explains that White-box models require meteorological, building, and occupant parameters rather than historic data in order to predict energy consumption in buildings. However, in our case on a national level, pinpointing such variables that have a direct impact on energy is doubtful. The study also describes that Black-box models on the other hand operate on historic data, these include machine learning models. The disadvantage of Black-box models mentioned is that they require data without null values, errors or inconsistency. Since we receive data from trusted sources which are clean and accurate, we may overcome this disadvantage. The Black-box models are highly adaptable and hence serves our purpose in nationwide energy forecasting.

Beyond the government reports, analysing other studies and publications gave a good insight into much of the technical requirements. (Debnath and Mourshed, 2018) reviewed several models used in energy demand forecasting. It also highlighted that Artificial neural network models outperformed traditional statistical models. Meanwhile Koreas peak load of electricity was forecasted in a study that focused on SARIMAX and LSTM model (Lee and Cho, 2021). Even in this case the machine learning based LSTM model had outperformed the SARIMAX model in forecasted results. However, since we don’t have seasonality to our annual data, we are bound to use a different statistical model. In this case the ARIMA model was a good candidate as per the study in forecasting electricity consumption in China (Mahia et al., 2019). According to the study an ARIMA model at (1,1,1) forecasted the best results. The proficiency of LSTM models was also backed by other energy forecasting studies which explained its use case in long term consumption forecasting (Wang, Du and Wang, 2020). Based on these researches we choose ARIMA as our statistical model and LSTM as our machine learning model to conduct our forecasting from which we may compare the results.

5. Methodology

5.1 Tools used

The main tool we used to analyse data and deploy our models is the Python programming language. Python is a popular language for data analysis due to its simplicity and ease of use. It has a vast array of libraries such as NumPy, Pandas, Matplotlib and Seaborn which are specifically designed for analysing, visualising and even manipulating data as well as perform statistical analysis. Additionally, Python's strong community support and extensive documentation make it a great choice for more information data analysis. The process of model building and predication was also supported by using libraries such as Statsmodels, Pmdarima, Keras, Sklearn, and Tensorflow. Overall, Python's versatility and accessibility make it a valuable tool for anyone working with data.

5.2 Methods used

Our study we will proceed on a story telling approach. Based on our research on previously published works we adapted several methods that could support the objectives of this study. The primary structure of this study will be as follows. Exploratory data analysis is conducted to establish a general insight into all of the nation's energy consumption. After this we continue to explore relationships between different fuel types as well as external variables influencing energy consumption among various sectors. Finally, we shall try forecasting energy consumption between different sectors via one statistical model and one machine learning model. All 3 of these sections work independently however, they effectively contribute towards our final findings and making recommendations. By utilising visualisations, statistical techniques and models we shall cover these 3 sections of this study.

Descriptive statistics

Mean: Adding up all the values in a dataset and dividing the result by the total number of values yields the mean, which is a measure of central tendency. The average is another name for the mean (Britannica, 2022).

mean = (x1 + x2 + x3 + ... + xn) / n

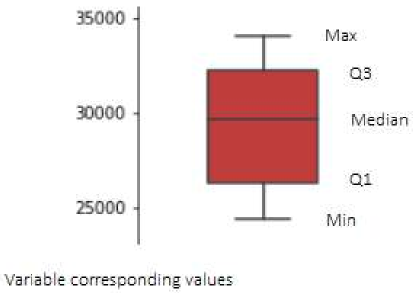
Standard deviation: The standard deviation helps us measure the variability in our data (Bhandari, 2020). It is calculated by finding the difference between each value in the dataset and the mean, squaring those differences, and then taking the square root of the sum of those squares. While a big standard deviation suggests that the values are widely dispersed, a small standard deviation suggests that the values in the dataset are close to the mean.

standard deviation = sqrt(((x1 - mean)^2 + (x2 - mean)^2 + (x3 - mean)^2 + ... + (xn - mean)^2) / n)

Boxplot: An illustration of a dataset that shows the median, quartiles, and range is called a boxplot. It's also considered as a toolkit for exploratory data analysis (Wickham & Stryjewski, 2011). The quartiles are the values that divide the data into four equal halves, and the median is the value in the middle of a dataset. The lower quartile is the value that divides the lowest

25% of the data from the highest 75%, and the upper quartile is the value that divides the lowest 75% of the data from the highest 25%. The range of the data is the difference between the highest and lowest values in the dataset. A boxplot can be created by plotting the median, lower quartile Q1, upper quartile Q3, and range of the data on a graph. Any values outside the box structure are identified as outliers.

An example of box plot structure:



Boxplot

The box plot will help us identify the range of consumption by energy sources through the years as well as identify extreme values that we may be able to investigate further.

Pearson correlation coefficient

The correlation coefficient is a statistical measure that is used to calculate the strength and direction of a linear relationship between two sets of data (Wikipedia Contributors, 2019). It is represented by the letter "r" and ranges in value from -1 to 1. A correlation coefficient of 1 indicates a strong positive correlation between the two variables, meaning that as one variable increases, the other variable also increases. A correlation coefficient of -1 indicates a strong negative correlation between the two variables, meaning that as one variable increases, the other variable decreases. A correlation coefficient of 0 indicates that there is no relationship between the two variables. This helps us identify relations between energy consumption by fuel types in specific sector as well as external numerical variables.

The correlation coefficient is calculated using the following formula:

$r = (n * \sum xy - \sum x * \sum y) / \sqrt{((n * \sum x^2 - (\sum x)^2) * (n * \sum y^2 - (\sum y)^2))}$

where:

n is the number of pairs of data (x, y)

x is the value of the first variable for a given data point

y is the value of the second variable for a given data point

$\sum x$ is the sum of all the x values

$\sum y$ is the sum of all the y values

$\sum xy$ is the sum of the products of the x and y values for each data point

$\sum x^2$ is the sum of the squares of the x values

$\sum y^2$ is the sum of the squares of the y values

In order to interpret correlation values as strong or weak we setup specific range suggested by an article claiming values above + or - 0.5 to 0.75 to have moderately strong correlation while anything above + or - 0.75 is a significant correlation (Zach, 2020). We will increase these measures to consider anything in range of + or - 0.6 to 0.8 as moderately strong correlation while considering values above + or - 0.8 to have significant correlation. This Ensures higher accuracy in establishing corelated variables. Values below these mentioned ranges will not be considered as having a correlation.

Augmented Dickey-Fuller (ADF)

The Augmented Dickey-Fuller (ADF) test is a statistical test used to determine whether a time series data set is stationary (SAID & DICKEY, 1984). The ADF test is a type of unit root test, which is used to determine whether a time series data set is nonstationary. The ADF test works by testing the null hypothesis that the time series data set has a unit root, which means that it is nonstationary.

The ADF test uses the following statistical model:

$$\Delta y[t] = c + \Phi 1\Delta y[t-1] + \Phi 2\Delta y[t-2] + ... + \Phi p y[t-p] + \theta 1\Delta y[t-1] + \theta 2\Delta y[t-2] + ... + \theta q y[t-q] + \varepsilon[t]$$

where:

$\Delta y[t]$ is the difference between the current value and the previous value of the time series data set at time t

c is a constant term

$\Phi 1, \Phi 2, ..., \Phi p$ are the lag coefficients of the difference term

$\theta 1, \theta 2, ..., \theta q$ are the lag coefficients of the additional variables included in the model

$\varepsilon[t]$ is the error term at time t

The ADF test uses the following equation to calculate the test statistic:

$$t = (c - c^*) / s.e.$$

where:

t is the test statistic

c is the estimated value of the constant term in the statistical model

c* is the critical value of the constant term

s.e. is the standard error of the estimate of the constant term

To determine whether the null hypothesis should be rejected, the test statistic is compared to critical values from a table. If the test statistic is greater than the critical value, the null hypothesis is rejected, and the time series data set is considered to be stationary. If the test statistic is less than the critical value, the null hypothesis is not rejected, and the time series data set is considered to be nonstationary. By using the ADF test we can establish stationarity of our model before building our ARIMA model. We can also use the test to see if our non-stationary energy consumption data will become stationary after different orders of differencing.

Differencing

Differencing is a statistical technique that is used to remove trends or patterns from time series data. It involves calculating the difference between consecutive observations, which can help to isolate the underlying fluctuations in the data.

y = x[t] - x[t-1]

Where x[t] is the current observation and x[t-1] is the previous observation. The resulting variable y is the difference between the two observations.

Differencing is often used in time series analysis to make the data stationary, which means that the mean, variance, and autocorrelation structure of the data do not depend on time. This can make it easier to model and forecast the data. As we are using ARIMA model we need to convert our non-stationary data into stationary format and hence, we shall use differencing in this scenario (Hyndman & Athanasopoulos, 2018, p. 8.1).

Standardisation

Standardization is a statistical method that is used to transform data so that it has a mean of 0 and a standard deviation of 1. This is often done so that data from different sources or with different scales can be compared or combined in statistical analyses. To standardize a variable, you subtract the mean of the variable from each value, and then divide the result by the standard deviation of the variable. We standardise our data before pushing it to an LSTM model so that our model isn't affected by the large differences between the value ranges in our input data set.

x' = (x - mean(x)) / stddev(x)

Where x is the original variable and x' is the standardized variable.

5.3 Models used

By researching on different papers discussing on energy modelling and forecasting, we found two models that are capable of taking our data and predicating future results. We have decided to build models using AutoRegressive Integrated Moving Average model, also called as ARIMA and Long Short-Term Memory model known as LSTM. ARIMA is a statistical model while LSTM is a type of Recurrent neural network, which happens to be a machine learning model. statistical models are better suited for problems where interpretability is important, while machine learning models are better suited for problems where prediction accuracy is the primary concern. Since each model has its advantage and disadvantage, we will try both these models to forecast our data and analyse which one gives better results.

ARIMA

ARIMA (short for AutoRegressive Integrated Moving Average) is a class of statistical models for analysing and forecasting time series data (Hyndman & Athanasopoulos, 2018, p. 8.5). It is a generalization of the simpler AutoRegressive Moving Average (ARMA) model, and adds the new notion of integration represented by the 'I'.

An ARIMA model is characterized by three parameters: p, d, and q. Here, p is the order of the autoregressive part of the model, d is the degree of difference, and q is the order of the moving average part of the model. The degree of difference d is used to make the time series stationary, as most time series models work best with stationary data. If a time series is not stationary, we can difference the series until it becomes stationary. To fit an ARIMA model, you need to determine the values of p, d, and q that best fit the data.

The mathematical formula for an ARIMA(p,d,q) model is given by:

$y_t = c + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$

where:

y_t is the value of the time series at time t

c is a constant

$\varphi_1, \ldots, \varphi_p$ are the parameters of the autoregressive part of the model

ε_t is the error term at time t

$\theta_1, \ldots, \theta_q$ are the parameters of the moving average part of the model

The value of d, the degree of difference, determines how many times the time series needs to be differenced in order to make it stationary. If d=0, then the series is already stationary. If d=1, then the series is differenced once to make it stationary, and so on.

LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) (Wikipedia Contributors, 2018). The RNN is a type of neural network that allows for information to be passed from one step of the network to the next. LSTM networks are able to learn long-term dependencies in data. It works by using "memory cells" that can store information for long periods of time, and "gates" that control the flow of information into and out of the memory cells.

The mathematical formula for an LSTM unit is given by:

$i_t = \sigma(W_i*[h_{t-1}, x_t] + b_i)$ $f_t = \sigma(W_f*[h_{t-1}, x_t] + b_f)$ $o_t = \sigma(W_o*[h_{t-1}, x_t] + b_o)$ $g_t = \tanh(W_g*[h_{t-1}, x_t] + b_g)$ $c_t = f_t * c_{t-1} + i_t * g_t$ $h_t = o_t * \tanh(c_t)$

where:

i_t is the input gate at time t

f_t is the forget gate at time t

o_t is the output gate at time t

g_t is the candidate memory cell at time t

c_t is the memory cell at time t

h_t is the hidden state at time t

x_t is the input at time t

W_i, W_f, W_o, W_g are the weights for the input gate, forget gate, output gate, and candidate memory cell, respectively

b_i, b_f, b_o, b_g are the biases for the input gate, forget gate, output gate, and candidate memory cell, respectively

σ is the sigmoid activation function

The input gate controls what information is allowed into the memory cell, the forget gate controls what information is removed from the memory cell, and the output gate controls what information is output from the memory cell.

LSTMs are able to learn long-term dependencies because the memory cell can retain information for long periods of time, and the gates can control what information is stored in the memory cell and what is forgotten. This allows LSTMs to learn patterns in data that span long sequences of time steps.

MAPE

The mean absolute percentage error (MAPE) is a measure of prediction accuracy of a forecast in statistics (Zach, 2020). It is defined as the mean absolute percentage error, which is calculated as:

$MAPE = (1/n) * \sum |(A_i - F_i)/A_i| * 100$

where:

n is the number of observations

A_i is the actual value of the i-th observation

F_i is the forecasted value of the i-th observation

MAPE is a commonly used metric to measure the accuracy of time series forecasts. It is particularly useful when you want to compare the forecast accuracy of different models or techniques, as it is scale-independent and easy to interpret.

RMSE

The error of a model in predicting quantitative data is typically measured using the Root Mean Square Error (RMSE). A (normalised) gap between the vector of anticipated values and the vector of observed values may considered as the RMSE (Moody, 2019). RMSE can help us understand how far may our predicted values will be different from actual values. The advantage of RMSE is that its in the same unit as our data.

The equation for the RMSE is:

RMSE = sqrt(mean((predicted - actual)^2))

Where:

Predicted - is the predicted value for the sample

Actual - is the actual value for the sample

Mean - is the mean of the squared differences between predicted and actual values.

Sqrt - is square root

Generally, the lower the RMSE value the better our model's accuracy. RMSE score can be used to further evaluate our model which we see is fit for our forecasting purposes.

6. Data and ethical implications

6.1 Datasets used for the study

Energy Consumption in the UK (ECUK): Final Energy Consumption Tables (Department for Business, Energy & Industrial Strategy, 2022) is used as our primary dataset to conduct our study. Our dataset includes historic data from 1970-2021 on energy consumption among the various sectors. The data provides energy consumption in UK by four sectors: domestic, services, transportation, and industry. Within each sector total energy was contributed by different sources of energy utilised within that sector, which was also available in our dataset. All energy consumption data are represented in "Thousand tonnes of oil equivalent" as its unit of measurement. By using this data, we may proceed to conduct exploratory data analysis to understand energy consumption and also use them to build our forecasting models.

We strongly relied on several secondary data sources to find influencing factors among each sector, they include.

- CO_2 emissions dataset which shows annual emissions within UK (Ritchie, 2022). The original source of data source was published as Global Carbon Budget 2022 (Friedlingstein et al., 2022).
- Gross domestic product of the United Kingdom from 1948 to 2021, shows GDP in millions (Office for National Statistics, 2022).
- Population of the United Kingdom from 1871 to 2021, gives the total population of UK measured annually (Office for National Statistics, 2022).
- Mean annual temperature in the United Kingdom (UK) from 1990 to 2021, shows average temperature in Celsius (Met Office, 2022).
- Vehicles registered for the first time by body type, shows new vehicles registered in UK (Department for Transport and Driver and Vehicle Licensing Agency, 2022).

6.2 Data cleaning and preparation

All of our data are sourced from trusted publishers. There was no instance when we had to deal with empty or invalid data. In the exploratory analysis part, we mainly focused on data from the past two decades. We did this because there have been several changes in the sector and technology with regard to energy consumption over the years. So, we must ensure that we are doing our general analysis on the past 20 years. Hence data was cropped from 2000-2021 for most of our primary analysis. We also made sure that the same timeline of data was cropped from all secondary datasets to evaluate influencing factors. All data used was in numerical form as a timeseries dataset and hence no categorical data existed. While considering data from 2000-2021 in certain fuel types such as Rail-coke & breeze in transportation, town gas in industries etc. we found zero energy consumption. This is due to the fact that such consumption sources existed before early 2000 but is currently not utilized. We have removed such empty features from our data as it does not contribute towards our analysis.

6.3 Ethics

Our primary data on energy consumption in UK comes under the Open Government License v3 (National Archives, n.d.). This allows the use of data in commercial or non-commercial context with the ability to copy or publish the information as required. Most of our secondary data are sourced from official government releases under the Open Government License. The data regarding UK emissions is publicly available under Creative Commons Attribution 4.0 License (Creative Commons, 2016). This allows us to share and adapt data as required in commercial or non-commercial environments.

During the course of this study, we have followed the UK Data Ethics Framework (Central Digital and Data Office, 2020). Which suggests we adapt some core principles to ensure ethical use of data. The principles used include:

- Transparency
  Ensure that all data, publications and information's conveyed are open, free and easy to access. All underlying data, methods and principles at use are publicly available and transparent.
- Accountability
  Accountability refers to the external oversight and public reviewing of the process. By ensuring all procedures, data and information are well documented and we propose accountability for our actions during the course of this study.
- Fairness
  Ensure that the project or the models created does not have any unintended discriminatory effects on individuals and social groups. Also ensure that the project does not identify any individuals or organizations and does not publish any private data or information.

7. Analysis

7.1 Exploratory data analysis

By focusing on historic data, we can gain a general idea on how energy is consumed across different sectors in UK.

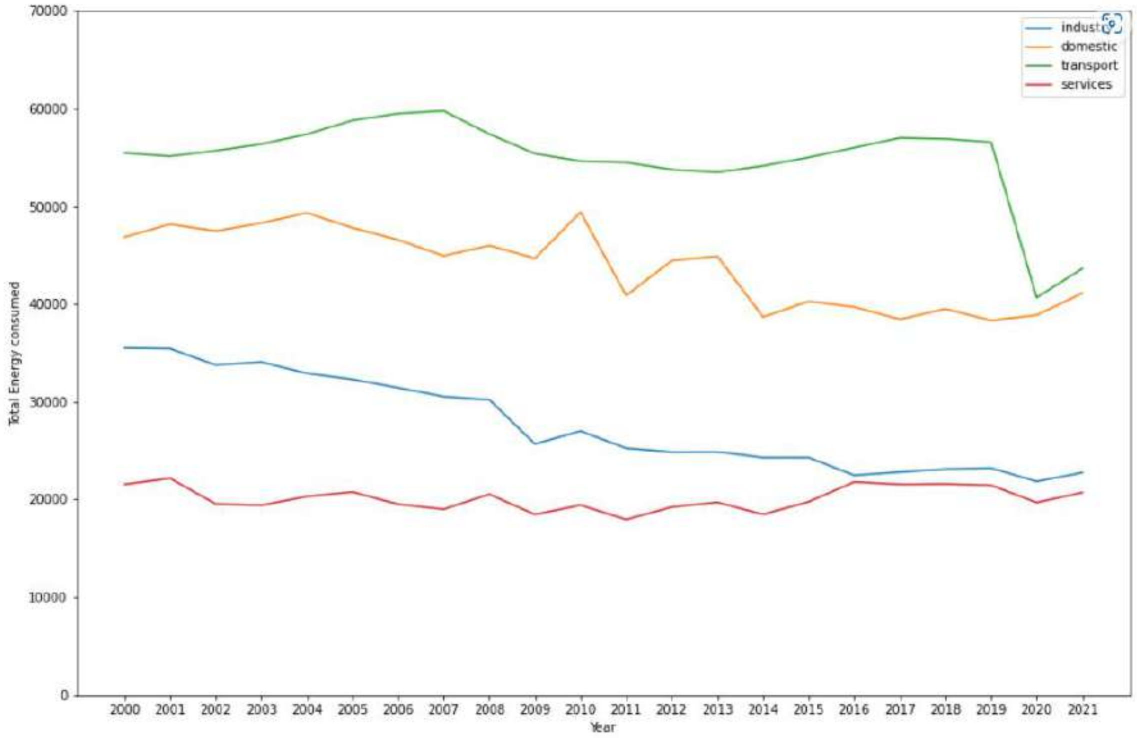


Figure 1: Energy consumption in UK in different sectors 2000-2021

Over the past 2 decades transportation sector had been at the top of most energy consumption in UK before having a historical drop in 2020. The primary reason for this was the nation-wide travel restrictions implemented due to covid which also reduced carbon emissions (Sung & Monschauer, 2020). The second-largest energy consumer was the household sector providing gas and electricity into homes across the UK. The industry sector marked a declining trend from 2000 till 2021. Out of the four sectors, services sector accounted for the minimum energy consumed till 2021.

According to the data recorded in 2021 the following figure (Figure 3) shows how the total energy is distributed across UK.

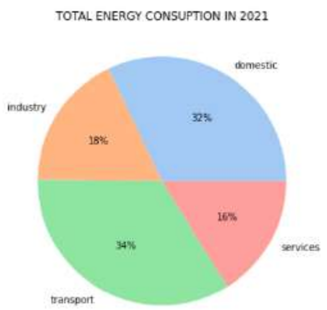


Figure 2: Pie chart of percentage of energy consumed in each sector, 2021

As expected, the majority of consumption is still under domestic and transportation sector. Together contributing to 66% of total energy consumption within the nation. However, in order to say with certainty that these figures are not influenced by the events of pandemic alone we will take the mean of total consumption from years 2000-2019 and compare our results.

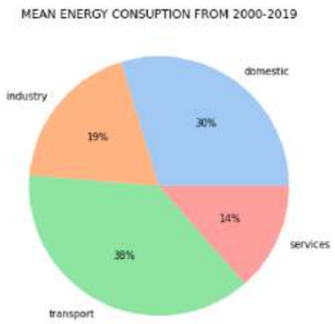


Figure 3: Pie chart of percentage of average energy consumed in each sector, 2000-2019

Based on this visualization (Figure 4) we can confirm that there have been very less significant changes in how energy consumption is distributed between sectors. The highest of change was observed in transportation sector with a 4% drop from its average consumption, considering the total consumption between all sectors.

Besides looking at energy consumption within each industry, it's also important to consider energy consumption in total and how it is changing over time. For this we first considered the total energy consumed within each year by taking the sum of all sectors consumption. After finding the total energy consumption for each year we will find the mean consumption from 2000-2021. Now for the purpose of visualization we will subtract the total energy consumed per year with mean energy consumption over all the years. This way the resulting years with a negative score will indicate total consumption lower than mean while the years with higher consumption will have positive values.

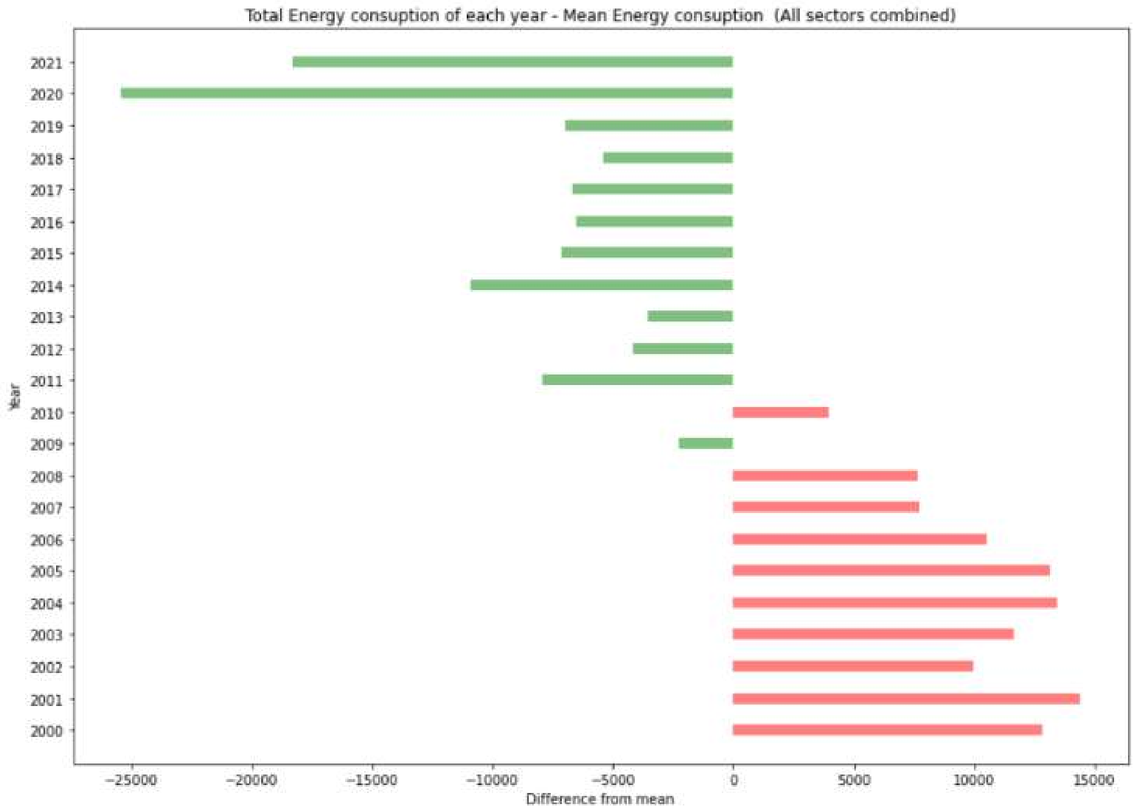


Figure 4: Total consumption in UK each year compared with mean consumption

As we notice the pattern in the above visualisation (Figure 5), we can easily infer that there has been a decrease in total energy consumption within the last decade when compared with the early 2000's. 2020 marked the least energy consumed within UK for the past 2 decades. We can also notice a significant variation in energy consumption between years 2008 till 2011 (Figure 1). In 2008 the UK economy had announced a technical recession as its GDP dropped from the second quarter of 2008 and continued to fall till the second quarter of 2009 (Office for National Statistics, 2018). However, we cannot simply say that this is the only factor that's influencing the energy consumption. Each sector in the energy industry is influenced differently by a variety of external factors. The following figure shows rate of change (%) in energy consumption for each sector between the years 2008-2012 (Figure 6).

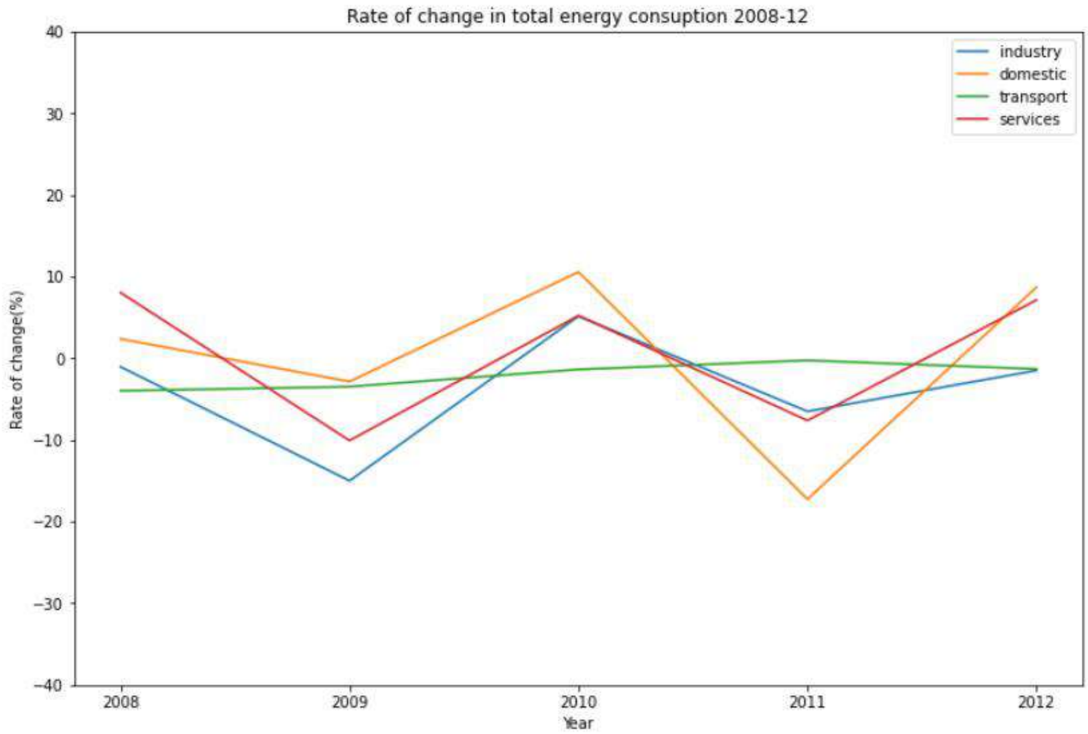


Figure 5: Rate of change in consumption among different sectors, 2008-2012

Notice that the fall in GDP on 2008-2009 had very little to no noticeable effect on the transportation sector but also had the most change in industrial sector. This indicates that the GDP change could be a considerable variable for checking correlation on the services or industrial sector and not the transportation sector. Hence analysing what factors are corelated with energy consumption in each sector will give us a better understanding of the consumption in each sector and even help strengthen our recommendations.

Taking a deeper dive into each sector we find that energy is consumed in different forms such as electricity, natural gas, petroleum etc. In order to understand the spread of each year's consumption within such varied energy forms we will utilize box plots.

Services sector

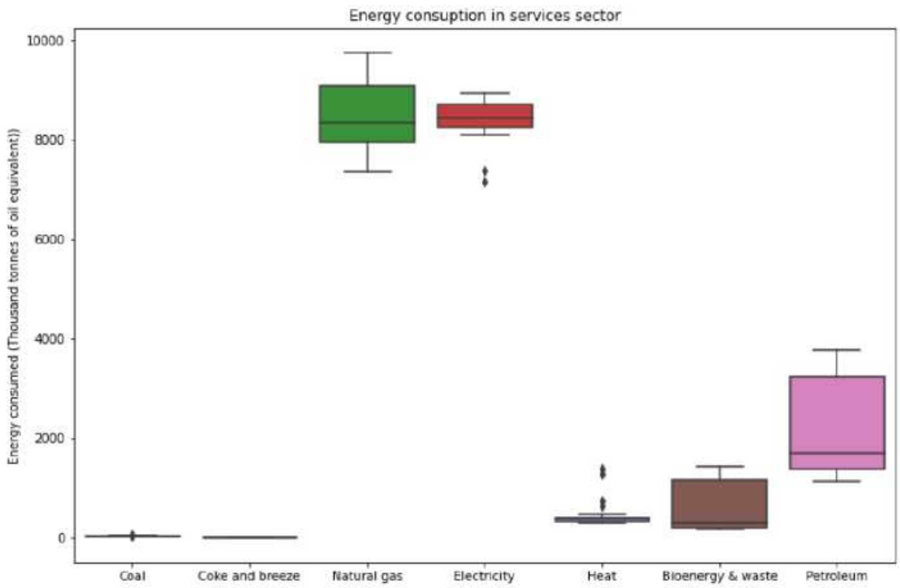


Figure 6: Service sector box plot by sources of consumption

Most of the energy consumed in services sector belongs to natural gas and electricity. Over the years, natural gas consumption had higher variations than electricity. However, its noticeable that electricity does have some outliers which could be observed in detail with historical evidence which can potentially point us towards some variable corelated factors. Petroleum is also a noticeable variable with high variations of consumption over the past two decades.

Domestic sector

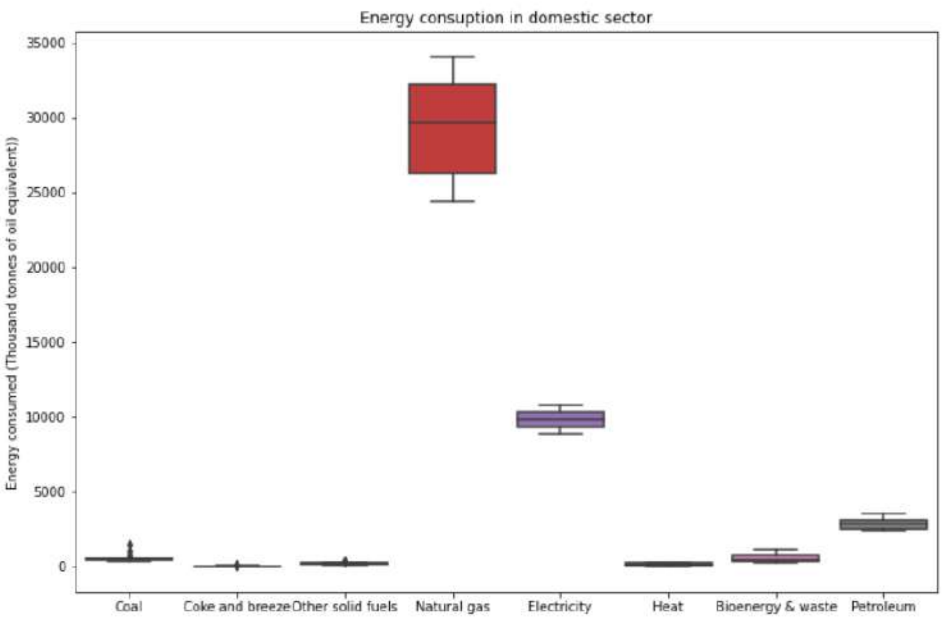


Figure 7: Domestic sector box plot by sources of consumption

In domestic sector, Natural gas dominates consumption more than any other energy type. Natural gas also has a high variation in its distribution. The second largest consumption is in electricity. There are no observable outliers in both these energy types which means we would have to resort back to observing line charts to observe trends in data and then look at historical evidence for finding influencing factors.

Industrial sector

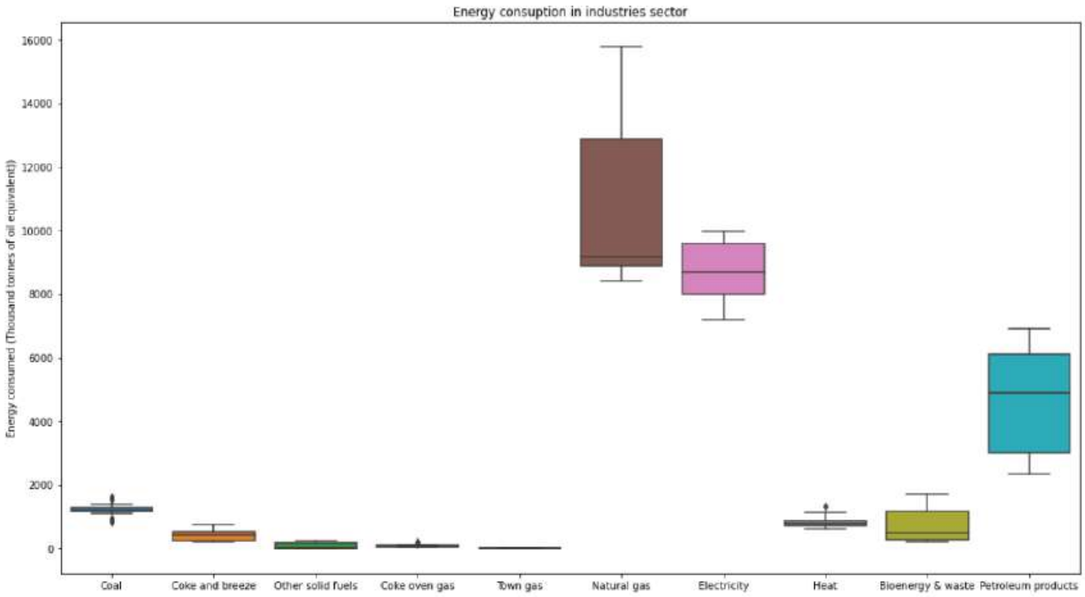


Figure 8: Industrial sector box plot by sources of consumption

In the industrial sector, again Natural gas leads consumption while electricity follows as second. However, observing the natural gas box plot we can identify that there is a very high level of variation in 50% of data above the median. But since we are looking at a timeline of energy consumption, in order to confirm that this variation is occurring continuously or irregularly we have to resort back to a line chart for more detailed explanation. The use of petroleum is higher than that of the services and domestic sector because just from the image we can notice that more than 50% of the data is above 5000 mark.

Transportation sector

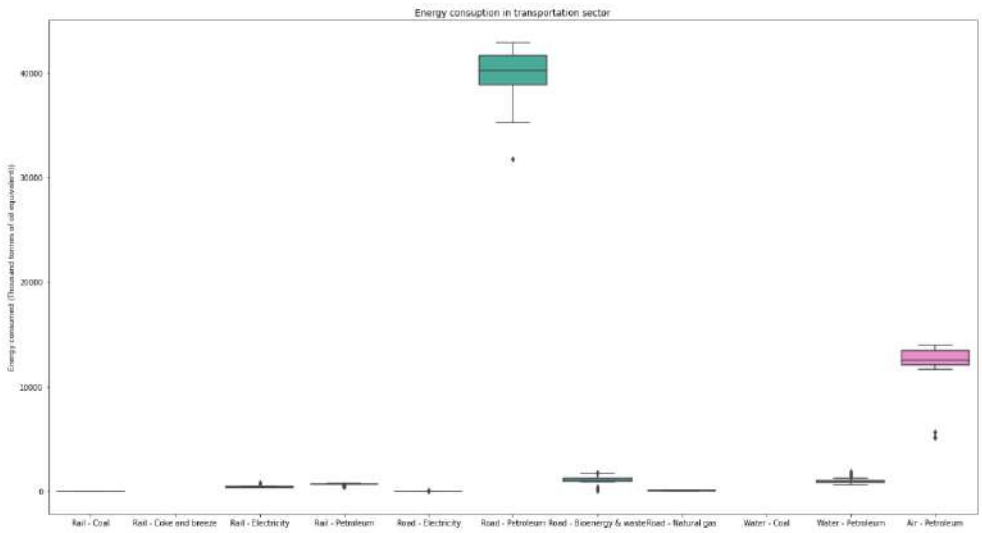


Figure 9: Transportation sector box plot by sources of consumption

Unlike previous sectors depending on Natural gas and Electricity, the transportation sector relies strongly on petroleum fuel as its major source of energy. In fact, it relies on petroleum so much that there are separate classifications within transportation types such as road, rail and air. The highest of consumption is in road transportation followed by air. The rest of the energy types are very small in consumption when compared with these 2 classifications. But we can also notice a few outliers within these sectors which can be observed for identifying influencing factors.

7.2 Finding influencing factors

After conducting a preliminary exploratory data analysis, we now have some general insight about the energy consumption in UK. We also have some visualisations that can aid us towards finding more historical evidence towards factors influencing different types of energy consumed. By finding the reasons for lows and highs in energy consumption we can detect the years of interest which we may investigate. Let's begin our study with the services sector.

Services sector

From our previous box plots we learned that the service sector utilizes both natural gas and electricity as its major source of energy (Figure 7). To further investigate upon the relation between existing variables we need to analyse the correlation between energy types.

	Coal	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum
Coal	1.000000	0.220732	0.087765	0.575436	-0.215158	0.009825
Natural gas	0.220732	1.000000	0.384626	0.540315	-0.618250	-0.352785
Electricity	0.087765	0.384626	1.000000	-0.043546	-0.618017	-0.632118
Heat	0.575436	0.540315	-0.043546	1.000000	-0.468507	-0.117738
Bioenergy & waste	-0.215158	-0.618250	-0.618017	-0.468507	1.000000	0.853211
Petroleum	0.009825	-0.352785	-0.632118	-0.117738	0.853211	1.000000

Figure 10: Service sector correlation coefficient on sources of consumption

There does not exist any significant correlation between natural gas and electricity, which happens to be the most consumed ones. However, we can find a strong positive correlation between petroleum and Bioenergy (Figure 11). Bioenergy is considered as a renewable energy source used in a variety of medians such as electricity generation, biofuel in transportation etc and is expected to decrease UK's reliance on imported fuels (Allen & Hammond, 2019). But since we don't have enough evidence stating the exact relationship between Bio energy and its use within various public/private service sectors, we cannot confirm for sure that this relationship is in fact a valid one.

Observing the box plot earlier we notice that there are 2 outliers in the electricity box plot (Figure 7). These 2 lowest values of electricity recorded are from year 2020 and 2021. As per the report of Bank of England stating "Covid has had a large impact on UK businesses", It's easier to consider this as a covid pandemic affect knowing that there had been high restrictions on businesses to operate, which could have caused the drop in electricity consumption in these years (Bank of England, 2020). This period also had caused a record fall in GDP up to 19.8% in the second quarter of 2020 as services sector did account for a large portion in contributing towards the country's economy (Office for National Statistics, 2020). The services sector contributes to more than 70% of national GDP (World Bank,2022). We could have used GDP as a primary factor however, almost all of the extra consumption we would have seen as a result of services economic growth has been offset by improved efficiency in energy consumption (Department for business, energy and industrial strategy, 2021, pp.12-13).

This encourages us to delve further into the consumption of services, which divides the industry into three categories: commercial and miscellaneous consumption, agricultural consumption, and public administration services. Analysing the latest data from 2021, approximately 67.2% energy comes from the commercial sectors while 25.6% is contributed by public administrations and only 7.2% is represented by the agriculture services. These figures explain why a decline in commercial operations during covid had impacted energy consumption in the services sector as most of its total energy requirement is from commercial services. Throughout the years, majority of the electricity and Natural gas consumption are generated from commercial and public administrations. Electricity is the highest consumed form of energy within the commercial services. When considering the commercial sector alone we found that the rate of change in electricity has a moderately strong correlation with the rate of change in GDP.

	Year	Electricity	gdp
Year	1.000000	-0.393268	-0.118205
Electricity	-0.393268	1.000000	0.734533
gdp	-0.118205	0.734533	1.000000

Figure 11: Correlation between rate of change in (commercial electricity and GDP)

However, the problem of efficiency still remains an unclear factor. The rate at which businesses and other services are adapting newer appliances, technologies and energy saving activates contributes to the improved efficiency in energy usage, which may be the reason for lower consumption trends over the years (Vella, 2017).

The public administration consists of several sub services such as social work, education, health care etc. In 2008 when UK was facing a recession, public sectors provided much support to flatten the curve (Office for National Statistics, 2018). This indicates that public administrations could operate more when under financial crisis. An increase in public housing projects was observed in 2022 as a result of a predicted recession which may last until mid-2024 (Gayne, 2022).

	gdp
Coal	-0.260884
Natural gas	-0.745390
Electricity	-0.704893
Heat	-0.853808

Figure 12: correlation between GDP and sources of consumption in public administration

When checking correlation with GDP we can also see a moderately strong negative correlation with electricity and natural gas. This may indicate that as the economy declines public administration increases its operations which also increases consumption in natural gas and electricity. These 2 forms of energy are also the most used in the public sector.

Industrial sector

The industrial sector primarily consumed natural gas and electricity. While observing several energy forms consumed in the industry, most of them are strongly corelated (Figure 14).

	Coal	Coke and breeze	Other solid fuels	Coke oven gas	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum products
Coal	1.000000	0.429472	0.088278	-0.093532	0.009031	0.259032	0.029122	-0.514681	0.319585
Coke and breeze	0.429472	1.000000	0.845482	0.522612	0.858508	0.780787	0.717962	-0.831401	0.864367
Other solid fuels	0.088278	0.845482	1.000000	0.558536	0.937493	0.904233	0.782213	-0.739478	0.855793
Coke oven gas	-0.093532	0.522612	0.558536	1.000000	0.602511	0.392611	0.411648	-0.204693	0.304014
Natural gas	0.009031	0.858508	0.937493	0.602511	1.000000	0.802882	0.830046	-0.662211	0.794954
Electricity	0.259032	0.780787	0.904233	0.392611	0.802882	1.000000	0.706998	-0.890012	0.915133
Heat	0.029122	0.717962	0.782213	0.411648	0.830046	0.706998	1.000000	-0.611817	0.716223
Bioenergy & waste	-0.514681	-0.831401	-0.739478	-0.204693	-0.662211	-0.890012	-0.611817	1.000000	-0.928099
Petroleum products	0.319585	0.864367	0.855793	0.304014	0.794954	0.915133	0.716223	-0.928099	1.000000

Figure 13: Industrial sector correlation coefficient on sources of consumption

Natural gas and Electricity are highly corelated, implying that the growth in either one is directly affecting the other in the same manner. More electricity will also reflect in more natural gas consumption. Petroleum products which happen to be the third highest consumed form of energy is also very highly corelated with electricity. This could prove that in order for most industrial entities to operate, energy forms such as natural gas, electricity and petroleum are very essential. Bioenergy and waste is the only variable with a strong negative correlation with multiple variables however, this is only due to the recent utilization of this energy form in order to maintain sustainability goals and replace conventional energy sources. Since all 3 major forms of energy has a positive correlation, we shall now consider total energy consumption to explore more external factors.

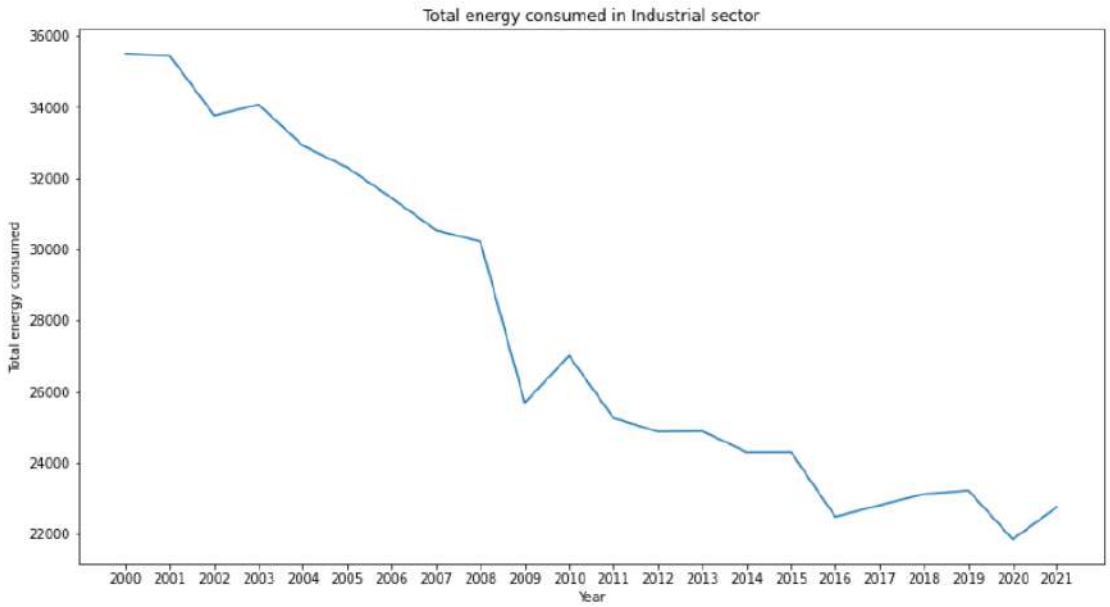


Figure 14: Industrial sector total energy consumption timeline, 2000-2021

Out of all 4 sectors, industrial sector has the most significant downwards/declining trend from the period of 2000-2021. This could be a result of decline in some of the major industries in

UK since the 90's. Emissions had decreased during the past few decades, particularly coal industries (Ritchie, 2019).

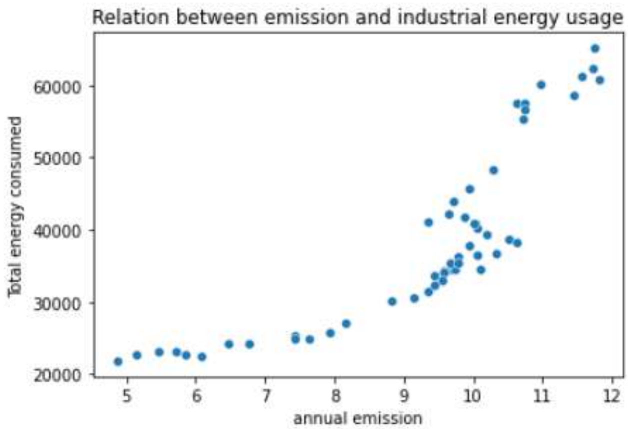
	Annual emissions
Annual emissions	1.000000
Coal	0.338575
Coke and breeze	0.885756
Other solid fuels	0.892093
Coke oven gas	0.365918
Natural gas	0.838097
Electricity	0.959460
Heat	0.747972
Bioenergy & waste	-0.950436
Petroleum products	0.974716
Total	0.943182

While we compare annual emission values with energy consumption, we can find a significantly strong correlation with total energy consumed (Figure 17). As energy usage in industries decreases, emission may also decrease. Across different forms of energy, we can see a similar positive correlation. This gives ample evidence that we can also consider UK's annual emission as an important factor influencing industrial sectors energy consumption. To confirm this with more evidence, we took historical data from 1970-2021 of both UK's annual emission and total energy consumption in industries. When evaluated once again we found a strong correlation between the two factors proving that emission could be a strong evaluating factor for total energy consumption in the industries sector.

Figure 16: correlation between annual emission and consumption sources in industry sector

	Total	annual emission
Total	1.000000	0.846089
annual emission	0.846089	1.000000

Figure 15: correlation between annual emission and total consumption in industry



Transportation sector

In our previous box plot we have found one outlier in road petroleum and two outliers in air petroleum (Figure 10). The year's corresponding these data are from 2020,2021 representing the travel restrictions during covid pandemic which also reduced emissions levels on a record high (Department for Transport, 2022). Over the year's most energy is consumed by road petroleum followed by air petroleum. In 2021, approximately about 80.7% of total energy consumed comes from road petroleum and 11.8% from air petroleum. Due to its less diversity in energy forms and the significance of road petroleum the transportation sector energy consumption scenario is quite easy to understand.

	Road - Electricity	Road - Petroleum	Road - Bioenergy & waste	Road - Natural gas
Road - Electricity	1.000000	-0.699193	0.560560	0.852185
Road - Petroleum	-0.699193	1.000000	-0.760037	-0.842277
Road - Bioenergy & waste	0.560560	-0.760037	1.000000	0.319743
Road - Natural gas	0.852185	-0.842277	0.319743	1.000000

Figure 17: Transportation sector correlation coefficient on sources of consumption

In road petroleum we can see a strong negative correlation with most other energy forms. This may be because of the newer generations of vehicles such as BATTERY ELECTRIC VEHICLE (BEV), COMPRESSED NATURAL GAS VEHICLE (CNG) etc. adapting towards such diverse forms of energy could also be helpful in reducing emissions.

A simple but efficient factor representing road petroleum is the number of registered vehicles in Great Britain.

	Year	total_vehicles	Road - Petroleum
Year	1.000000	-0.409075	-0.722913
total_vehicles	-0.409075	1.000000	0.738532
Road - Petroleum	-0.722913	0.738532	1.000000

Figure 18: correlation between new registered vehicles and road-petroleum

While checking the correlation with new road vehicles registered in Great Britain (excluding Northern Ireland) from 2001-2021 we found a strong positive correlation with road petroleum (Figure 19).

This indicated that as registered vehicles increased, the consumption of road energy also increased. When analysing during the next few years, we might also have to consider how many of these vehicles registered belong in electric or CNG engines.

Although rail energy use is considerably low, it is also a median of public transport in UK.

	Rail - Coal	Rail - Electricity	Rail - Petroleum
Rail - Coal	1.000000	-0.760213	-0.139370
Rail - Electricity	-0.760213	1.000000	0.042336
Rail - Petroleum	-0.139370	0.042336	1.000000

Figure 19: correlation between different sources of consumption by Rail

The transition from coal engine to electrifying rail transports are quite evident from the given correlation. Rail electricity consumption has showed a moderately strong negative correlation with rail coal (Figure 20).

While analysing the air petroleum consumption we found that there is very little changes in its overall trend before hitting a record low in 2020. As a fact, if we are not considering the years 2020 and 2021, air petroleum gives us a very low standard deviation (Figure 21).

	Air - Petroleum
count	19.000000
mean	12770.315789
std	738.164469

Figure 20: descriptive statistics on Air-petroleum

This further implies that air petroleum is very less prone to external factors influencing it.

Domestic sector

The domestic sector primarily consumes natural gas followed by electricity.

	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum
Coal	1.000000	0.784505	0.951028	0.586504	0.267250	-0.523851	-0.652176	0.683145
Coke and breeze	0.784505	1.000000	0.799762	0.596353	0.400675	-0.490157	-0.616761	0.534781
Other solid fuels	0.951028	0.799762	1.000000	0.682307	0.390840	-0.643366	-0.770309	0.771051
Natural gas	0.586504	0.596353	0.682307	1.000000	0.869536	-0.748838	-0.798759	0.929135
Electricity	0.267250	0.400675	0.390840	0.869536	1.000000	-0.770547	-0.785557	0.765162
Heat	-0.523851	-0.490157	-0.643366	-0.748838	-0.770547	1.000000	0.887469	-0.765266
Bioenergy & waste	-0.652176	-0.616761	-0.770309	-0.798759	-0.785557	0.887469	1.000000	-0.833612
Petroleum	0.683145	0.534781	0.771051	0.929135	0.765162	-0.765266	-0.833612	1.000000

Figure 21: Domestic sector correlation coefficient on sources of consumption

Electricity, natural gas and petroleum are all highly positively corelated variables (Figure 22). Bioenergy and waste have also recently started growing in a consistent rate.

In the housing sector energy is used in various ways such as cooking, space heating, water heating, lighting and other appliances (Wikipedia Contributors, 2019). As we discussed before, energy efficiency still remains to be an important factor representing energy consumption in housing sector.

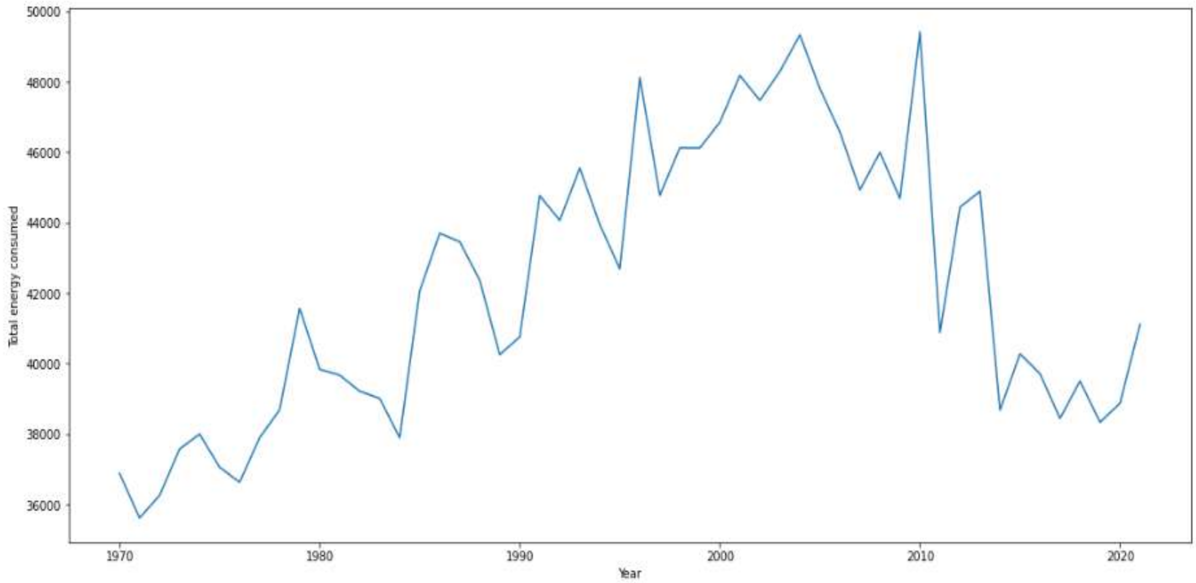


Figure 22: Domestic sector total consumption timeline, 1970-2021

When considered the overall consumption from 1970-2021 we can visualise an almost arc trend indicating an initial increase in consumption trend followed by a decline (Figure 23). Energy performance certificate (EPC) are handed out by the government for each building based on their energy efficiency score (SAP score), which happens to be a mandatory requirement for domestic or non-domestic structures. Studies found that mostly flats and maisonettes showed higher efficiency while detached and terraced housings were the least efficient (Bowers et al., 2022). Besides the efficiency factor we can also spot very large highs and lows in our plot which resembles high variations. The rate of change in consumption from 2010 to 2011 was a record of approximately -17.25 %.

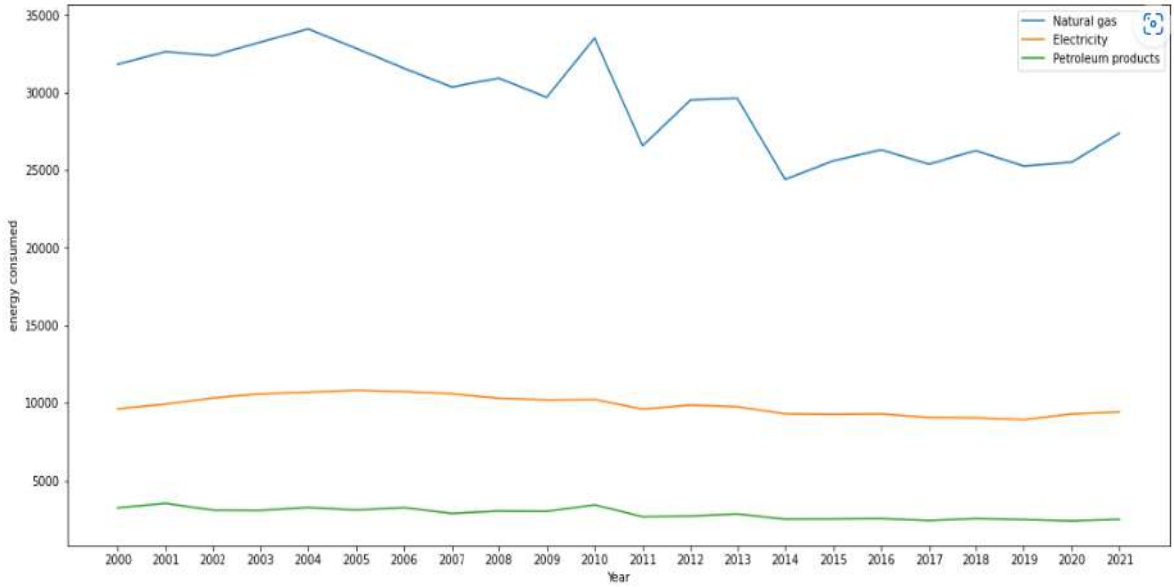


Figure 23: Domestic sector timeline by sources of consumption, 2000-2021

However, when observing different energy forms, we realise that almost all of the variations are influenced by the uneven pattern of natural gas consumption (Figure 24). As a fact, natural gas has a significant correlation score of 0.99 with total energy consumption in domestic sector. Natural gas is primarily used in cooking and heating among other applications. However, when investigated further there is no evident correlation s between any domestic energy forms consumed and annual temperature. Even with the growing population there was no increase in energy consumption. These may be the result of the underlying efficiency factor. According to the reports from the office of national statistics, the age of a property has significant influence on its energy efficiency by indicating that newer buildings have higher efficiency when compared with older ones (ONS, 2022).

7.3 Energy forecasting

There are several models that are suitable for forecasting. However, we will explore 2 models, one that's a statistical model and the other a machine learning model. For the statistical model we shall consider ARIMA model and our machine learning model will be LSTM. We shall test both models and see which one is more compatible for our data. Both these models are widely used in timeseries forecasting. All models use an 80:20 train and test ratio on our dataset provided. This helps us score our models accuracy and plot the results.

ARIMA model

Step 1: Exploring data for stationarity

We can analyse the data from 1970-2021 as a timeline and conclude that the data is non-stationary and has a strong trend. However, we will resort to statistical methods to further strengthen our results. By conducting the Augmented Dickey–Fuller test, we can understand if consumptions in each sector have stationarity in them. The following are the results of the test conducted on all sectors consuming energy.

```
INDUSTRY
        ADF Statistic: -1.566226
        p-value: 0.500534

{'1%': -3.5656240522121956, '5%': -2.920142229157715, '10%': -2.598014675124952}

DOMESTIC
        ADF Statistic: -1.910875
        p-value: 0.326963

{'1%': -3.568485864, '5%': -2.92135992, '10%': -2.5986616}

SERVICES
        ADF Statistic: -2.847493
        p-value: 0.051803

{'1%': -3.5656240522121956, '5%': -2.920142229157715, '10%': -2.598014675124952}

TRANSPORTATION
        ADF Statistic: -1.867004
        p-value: 0.347739

{'1%': -3.5656240522121956, '5%': -2.920142229157715, '10%': -2.598014675124952}
```

Figure 24: ADF results

We can interpret the results in multiple ways. Notice that the p values are all above 0.05, this indicates that the data is non-stationary. We may also compare if the ADF statistics result is smaller than 5% confidence results. The services sector is the only sector that comes close to the probability that the data may be stationary. But based on the statistical evidence at 95% confidence and observing the variations over the years, we will consider the dataset as non-stationary.

Step 2: Differencing to convert data into stationary

By using differencing technique, we can remove trends from our data to make it look stationary. Differencing can range from order 1 to as much as required. We will conduct first order differencing on our data and recheck our data with ADF test to see if they are stationary.

ADF after differencing order: 1

```
p-value INDUSTRY: 0.000000

p-value DOMEATIC: 0.000000

p-value SERVICES: 0.000000

p-value TRANSPORT: 0.000000
```

All p values are << 0.05 indicating stationarity in data. This also helped us identify our value for d as 1, for our ARIMA model.

Step 3: finding p and q

By using the best combination of p and q we may find better prediction results for our ARIMA model. We will determine our value for p using a partial autocorrelation plot and the value for q using autocorrelation plot. For domestic sector data we plotted the following.

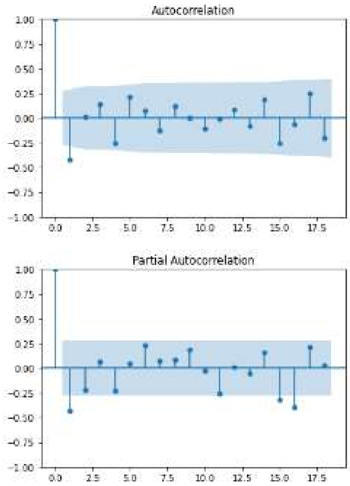


Figure 25: ACF & PACF - domestic sector

Here in the Autocorrelation chart, we only see one significant lag in the beginning while all other lags are not significant. In the Partial Autocorrelation plot we can identify 3 significant lags. Yet, these lags do not show any pattern at all. However, we will consider p = 2 and q=1. This gives us a (2,1,1) ARIMA model for prediction. We do this to explore the difference in building custom ARIMA's to see its prediction capabilities. While an AUTOARIMA function has returned our best fit model to be (0,1,0).

While looking through the rest of 3 sectors via autocorrelation and partial autocorrelation we found that the series has almost no significant lags and can be considered as white noise (Figure 27).

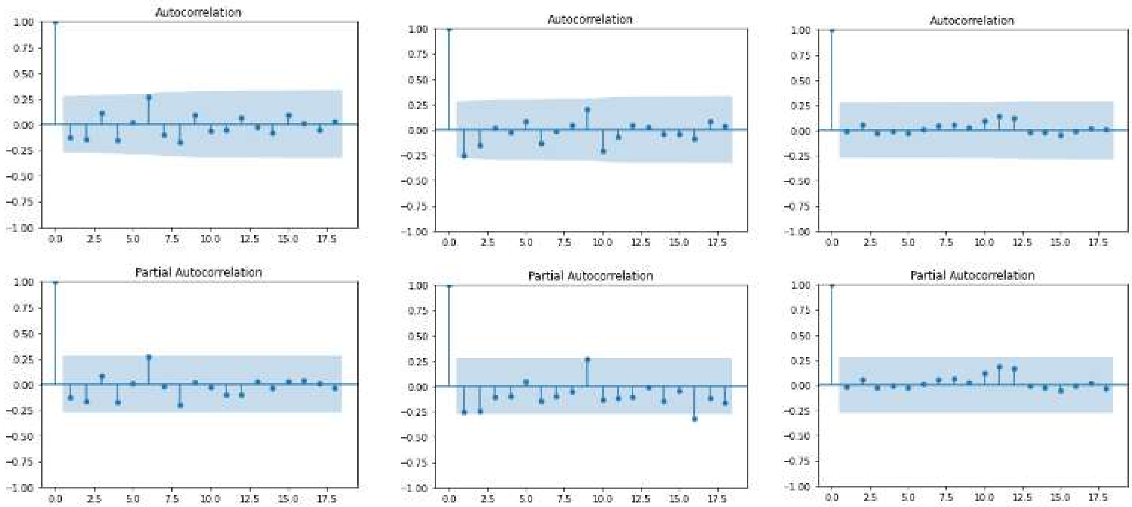


Figure 26: ACF & PACF - industry, Transport, Services

Due to this issue, we shall use an AUTO_ARIMA function in python or alternatively loop though possible combinations of p and q to improve our model. This function returns us the best combination of p, d, q values for a given dataset (G Smith, n.d.). When we build our models via the AUTO_ARIMA function we find that the most optimised models are all by the combination (0,1,0). These types of builds are called Random walk (Nau, n.d.).

Step 4: forecasting

Due to the white noise in data our model does not predict the best of results. Instead, its output is rather an average predictable value which is very week towards high or low variations. We used a train set of data to train our model and a test set to predict our results and also view the accuracy of our model. The graph below shows the difference between predicated results and actual value in the domestic sector using (2,1,1) ARIMA model which has better results than (0,1,0) ARIMA.

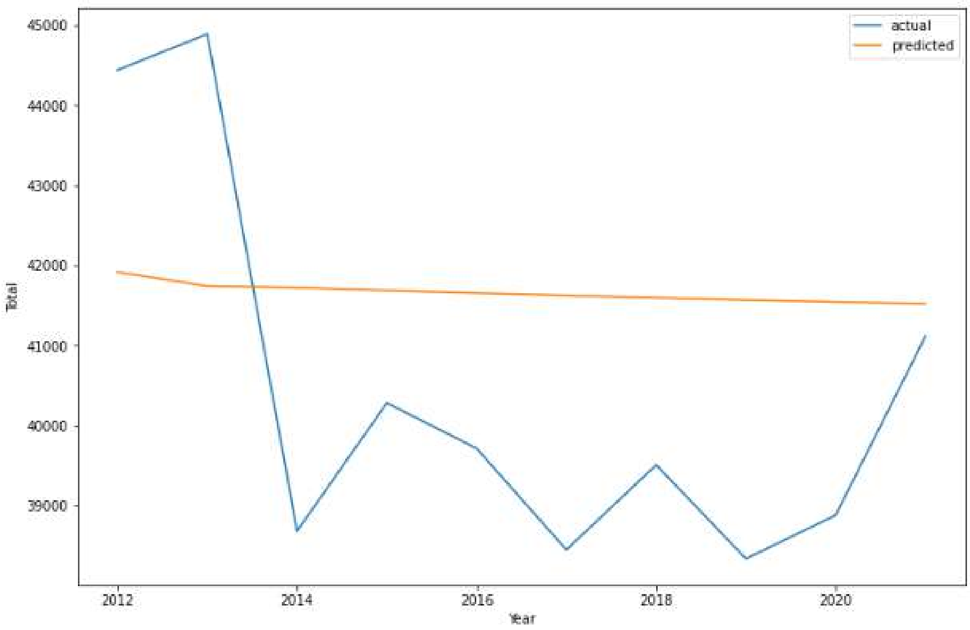


Figure 27: ARIMA results domestic sector

As observed in the graph the predicted values do not show much dips and highs in values (Figure 28). The possibility that the (2,1,1) ARIMA can predict future values is also low because this model may not be accurate when dealing with data past 2021. This model can also be alternatively built at (0,1,0) when using an AUTO ARIMA Function. This also shows inconsistency when predicting for long term. These patterns exist in all sectors of energy consumption when using the ARMIA model at (0,1,0).

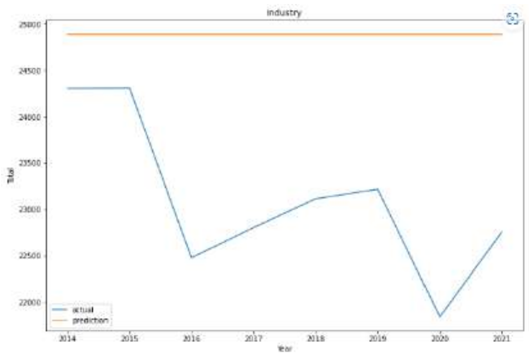


Figure 28: ARIMA results industry

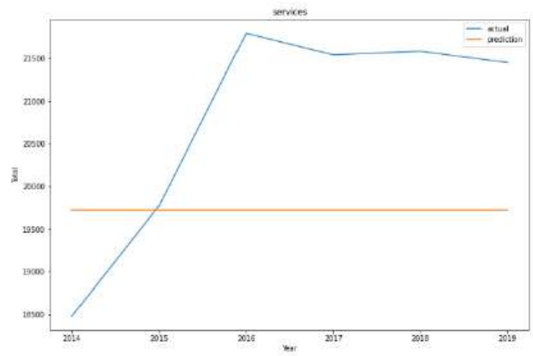


Figure 29: ARIMA results services

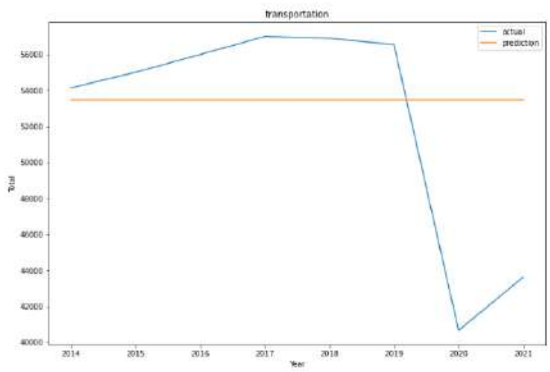


Figure 30: ARIMA results transportation

The following are the accuracy scores obtained for each sector based on MAPE measures. The lower the value the better our accuracy.

Energy sectors	MAPE (%)
Domestic	5.877
Services	5.82
Transportation	10.02
Industry	7.84

The transportation industry showed less accuracy possibly due to the covid pandemic drop in actual values which was not accounted in our model.

LSTM model

Step 1: Standardise values

Standardising values helps remove high variations in data. This also helps our model to be less prone to changes due to outlier values and also improves computation speed. After standardisation the range of our data exists between 0 and 1.

Step 2: Define model

We have defined a single layer LSTM model with an input size of 3. The input size defines the number of data points to consider while predicting the next data. By tweaking the input size, we learned that when the size increases our prediction resorts towards a similar model to that of the ARIMA. Hence, we choose 3 as the input size which gives satisfactory results. We are also using a Single Layer LSTM model. This means there is a single layer of LSTM between the input and output nodes.

Step 3: Forecasting

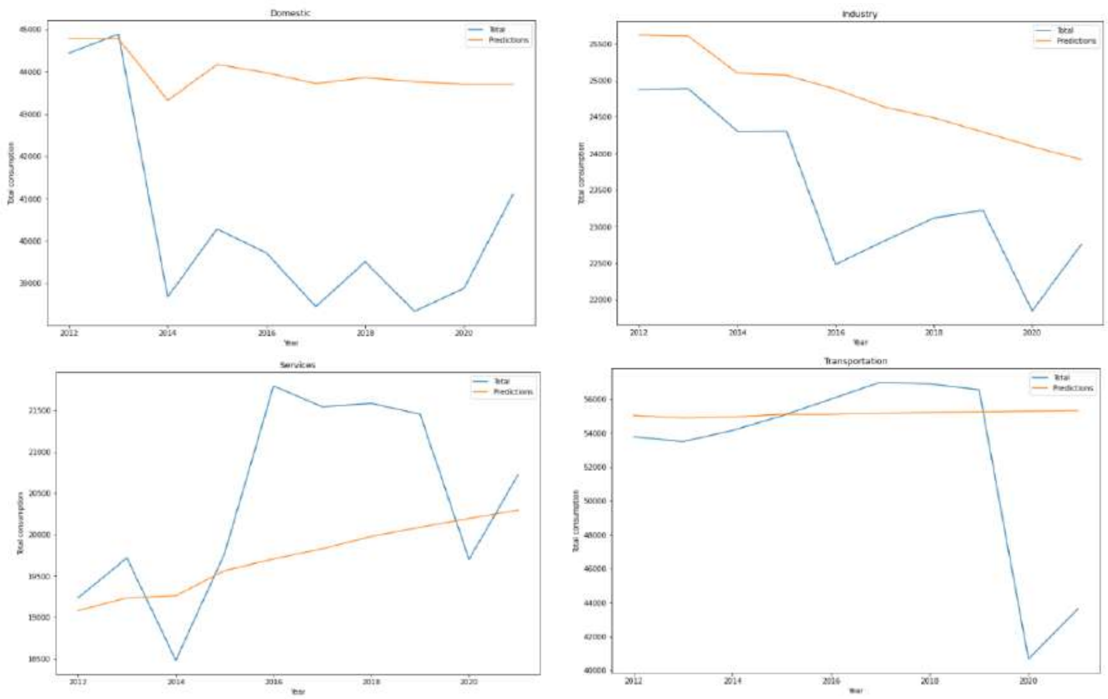


Figure 31: LSTM results on all four sectors

The above are the forecasted results from our LSTM model. The results show much more deviations in data than our previous ARIMA model. Much of our results also has improved accuracy except the domestic model when compared with the custom build (2,1,1) ARIMA model.

The following are the accuracy scores obtained for each sector based on MAPE measures. The lower the value the better our accuracy.

Energy sectors	MAPE (%)
Domestic	9.10
Services	4.44
Transportation	7.95
Industry	5.68

Each epoch consists of one or more batches in which the neural network is trained using a portion of the dataset. By looking at loss in each epoch we can estimate the number of epochs to consider for our model.

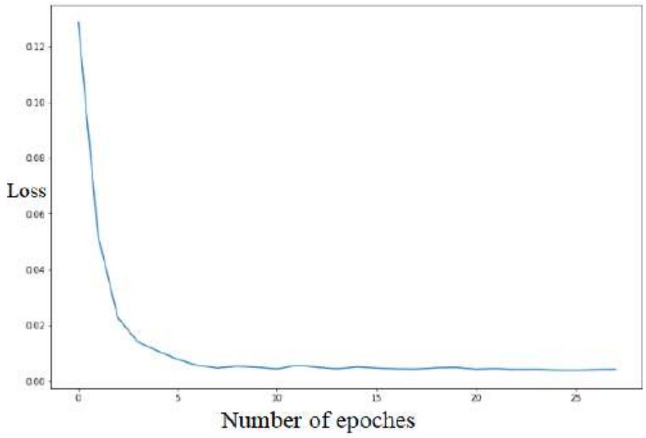


Figure 32: Epoch loss

Notice that after 10 Epochs there is no significant gain made by our model (Figure 33). This means we can set the epoch parameter to 10 indicating without compromising on model accuracy. While working on larger data this can save computation time.

RMSE is another measure of calculating error in order to further asses our model. RMSE score depends on the unit of measurement our data is using. Hence, in this case the error is defined in thousand tonnes of oil equivalent.

Energy sectors	RMSE
Domestic	4382.314
Services	1415.769
Transportation	9880.186
Industry	770.302

A very small RMSE will indicate that our models are closer to overfitting while a very large RMSE can indicate underfitting. Notice that in our sectors, the ones with lower RMSE score belong to sectors with lower energy consumption. Transportation was observed to have a very high RMSE score because of its recent unforeseen dip in 2020 due to the pandemic. Considering the total energy consumption within each sector all RMSE scores are a decent under 10% error values.

8. Findings & Recommendations

Energy forecasting models build using LSTM shows higher accuracy and much more deviations than ARIMA models. However, given the case that there exist more quarterly data for energy consumption we may reassess the ARIMA model and its forecasting capabilities. The LSTM model we build can be used in predicating energy consumption for up to four years with a mean error less than 10%. Factors such as global events, pandemics or political outcomes are not influencing the model and hence real-world scenarios may have different outcome. By finding enough influencing features in each sector through the research framework setup in this study, using a multivariant LSTM model can improve the accuracy of our forecasts.

Domestic sector

Domestic sectors contribute towards approximately more than 30% of total energy consumed in the UK. Its primary source of consumption is by natural gas followed by electricity. However, we cannot establish a direct connection with energy consumption and average annual temperature because the temperatures may vary from different areas and geographic locations. For example, areas with major cities and large number of residential buildings may have a certain temperature while areas with lower residential density may have a different temperature. Again, another factor that didn't show any relation with energy consumption was population. Even with an increasing population there was no observable increase in energy consumption. All these might be the undocumented results of improved energy efficiency over the past decades. However, by exploring other studies we understand that energy consumption in a domestic sector is influenced by each residential building and how they are constructed. Depending on the age of the building, type of structure, among other details will have an effect on how energy is consumed in these domestic structures. While implementing policies ensure importance in improving Energy Performance Certificate SAP scores in homes as well as consider the fact that a structures age is a prime factor in energy usage. By decreasing reliance on natural gas sources and converting to renewable electric or the growing bioenergy sources the domestic sectors energy consumption can be further improved.

Transport sector

Transportation sector is another major contributor to UK's total energy consumption, at a two-decade average more than 30%. Within the transportation sector the most energy is consumed in form of road fuel which is more than twice higher than the second most consumed form, which is air fuel. Over the years the transportation sector showed much more consistency than other sectors in its energy consumption. The only time in the past two decades when transportation energy consumption went down is during the COVID-19 pandemic periods of 2020 and 21. During this period travel restriction and lockdowns affected both air and road fuel consumptions alike, making a historical dip in consumption. A variable indicator that resembles road fuel consumption was the number of new registered vehicles in Great Britain. Currently this variable is able to represent road fuel only because of the fact that there are mostly fossil fuel-based engines on the road. Policies that support electric vehicles or CNG engines can help diversify the consumption style of transportation sector. However, the long-term implications and manufacturing requirement of electric vehicles is a broader topic to give attention to. Meanwhile rather than looking to adapt more energy efficient forms of transport, policies that improve public transportation schemes as well as introducing more cycling lanes and walkable city plans can contribute highly towards decreasing energy consumption.

Service sector

Services sector showed the least trend in energy consumption among the other three sectors. It is the lowest energy consuming sector out of the four sectors in UK. Even though services sectors contribute to more than 70% of national GDP, it has very less direct influence on how energy is consumed. This is mostly due to underlying factor of improved energy efficiency over the years. Energy is mostly consumed in form of Natural gas followed by electricity. Electricity was found to have dropped in the COVID-19 pandemic period most probably due to the lockdowns and restrictions. While taking a deeper dive into subsectors in services, as per 2021 data, we found that more than 67% of total energy consumed comes from commercial services followed by public administration at 25% and agriculture at the very least. Even when rechecking GDP's relations with the commercial energy consumption there was no significant results. But public administration showed an interesting relation which was also supported by historical evidence that whenever GDP was declining public administration ramped up its operations resulting in increased energy consumption. Yet, in order to implement better policies in the service sector much more understanding about influencing factors is needed especially in the commercial service sectors.

Industrial sector

The industrial sector had a strong declining trend starting from early 2000's to 2021. this is most probably due to globalization and increased reliance on imported goods as well as the increase in offshore manufacturing and industrial hubs. Natural gas is the most used energy source followed by electricity and petroleum. Factors such as emission had showed a high relation with industrial energy consumption. As some of the major emission contributing sources such as coal industries started disappearing, emission level decreased and so did industrial energy consumption. While UK pushes forward on its net-zero targets, it's understandable how industries have to play a vital role in being energy efficient as well as emission free. This brings us to the understanding that lifting any emission controlling industrial policies may end up increasing energy consumption in UK's industrial sector.

9. Conclusion

This study has focused on 3 independent sections focusing energy consumption in UK. Exploratory data analysis established a general insight about each sectors energy consumption. Briefly analysing factors helped understand external variables that are correlated with energy consumption. Building energy forecasting models gave us ample idea of best modelling techniques while also comparing a statistical and machine learning model. Based on the output of these three sections of our study we gained ample insight into the nation's energy consumption scenario. Energy consumption in the UK has consistently gone down for the past two decades when compared with consumption trends from 1970's till late 90's. More than 60% of energy usage comes from Domestic and Transportation sectors. Each sectors energy consumption is affected by different external factors such as emission in case of industrial energy consumption and number of new vehicles in road fuel within the transportation sector. Some other factors influencing was not detected via data but by referencing other published papers such as how a buildings age is highly related with energy consumption in the domestic sector. Relating these factors with energy consumption helps recommend better energy saving techniques as well as consider better policies to improve consumption. While modelling for energy forecasting, we bumped into some hurdles with our statistical ARIMA model. As the data we had mainly consisted of white noise we were not able to see much deviations in our model and hence had poor prediction results. This issue was however addressed by our machine learning model LSTM and improved our forecasting abilities which showed better accuracy and proper deviations in predicted data when calculating for the upcoming 4 years. By expanding this study and finding more energy consumption influencing variables we can further improve these models and upgrade our prediction capabilities. Hence this study can also act as a framework for future reference and improvements.

10. References

1. Allen, P. E., & Hammond, G. P. (2019). Bioenergy utilization for a low carbon future in the UK: the evaluation of some alternative scenarios and projections. BMC Energy, 1(1). https://doi.org/10.1186/s42500-019-0002-9
2. Bank of England. (2020, December 17). Impact of Covid-19 on UK businesses – evidence from the Decision Maker Panel in 2020 Q4. Www.bankofengland.co.uk. https://www.bankofengland.co.uk/agents-summary/2020/2020-q4/impact-of-covid-19-on-uk-businesses-evidence-from-the-decision-maker-panel-in-2020-q4#:~:text=For%20the%20first%20time%2C%20the
3. BBC. (2022, February 8). Energy prices: What Is a Windfall Tax and How Would It work? BBC News. https://www.bbc.co.uk/news/business-60295177
4. Bhandari, P. (2020, September 17). Standard Deviation | A Step by Step Guide with Formulas. Scribbr. https://www.scribbr.com/statistics/standard-deviation/
5. Bowers, N., Smith, C., & Wilkins, T. (2022, October 25). Energy efficiency of housing in England and Wales - Office for National Statistics. Www.ons.gov.uk. https://www.ons.gov.uk/peoplepopulationandcommunity/housing/articles/energyefficiencyofhousinginenglandandwales/2022#:~:text=The%20energy%20efficiency%20score%20(SAP
6. BP. (2022). bp Statistical Review of World Energy 2022 | 71. https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2022-full-report.pdf
7. BP. (2022). Statistical Review of World Energy. BP Global. https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html
8. Britannica, T. Editors of Encyclopaedia (2022, December 30). mean. Encyclopedia Britannica. https://www.britannica.com/science/mean
9. Central Digital and Data Office (2020). Data Ethics Framework. [online] GOV.UK. Available at: https://www.gov.uk/government/publications/data-ethics-framework/data-ethics-framework-2020.
10. Creative Commons (2016). Creative Commons — Attribution 4.0 International — CC BY 4.0. [online] Creativecommons.org. Available at: https://creativecommons.org/licenses/by/4.0/.
11. Debnath, K.B. and Mourshed, M. (2018). Forecasting methods in energy planning models. Renewable and Sustainable Energy Reviews, [online] 88, pp.297–325. doi:10.1016/j.rser.2018.02.002.
12. Department for Business, Energy & Industrial Strategy. (2022, sept 29). Energy Consumption in the UK (ECUK): Final Energy Consumption Tables [Dataset]. ECUK 2022: Consumption data tables (Excel). Energy consumption in the UK 2022 - GOV.UK (www.gov.uk)
13. Department for Business, Energy and Industrial strategy (2022). Energy Consumption in the UK (ECUK) 1970 to 2021. [online] National Statistics. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1110483/Energy_Consumption_in_the_UK_2022_10102022.pdf.
14. Department for Business, Energy and Industrial Strategy (2022). UK Energy in Brief 2022. [online] Available at:

https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1094025/UK_Energy_in_Brief_2022.pdf.
15. Department for Business, Energy and Industrial Strategy. (2021). UK ENERGY IN BRIEF 2021. National Statistics. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1032260/UK_Energy_in_Brief_2021.pdf
16. Department for business, energy and industrial strategy. (2021). Energy Consumption in the UK (ECUK) 1970 to 2020 Data tables million tonnes of oil equivalent Industry Transport Services Domestic. Office for National Statistics. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1061644/Energy_Consumption_in_the_UK_2021.pdf
17. Department for Transport and Driver and Vehicle Licensing Agency. (2022, sept 29). Vehicles registered for the first time by body type [Dataset]. VEH0150: Vehicles registered for the first time by body type: Great Britain and United Kingdom (ODS). Vehicle licensing statistics data tables- GOV.UK (www.gov.uk). https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1123253/veh0150.ods
18. Department for Transport. (2022, October 20). Transport and environment statistics 2022. GOV.UK. https://www.gov.uk/government/statistics/transport-and-environment-statistics-2022/transport-and-environment-statistics-2022#:~:text=Over%20half%20(55%25)%20of
19. Druckman, A. and Jackson, T. (2008). Household energy consumption in the UK: A highly geographically and socio-economically disaggregated model. Energy Policy, [online] 36(8), pp.3177–3192. doi:10.1016/j.enpol.2008.03.021.
20. Food and Agriculture Organization. (2021). Wood Energy. www.fao.org; https://www.fao.org/forestry/energy/en/
21. Foy, H. (2022, December 2). EU reaches deal to impose $60 cap on Russian oil exports. Financial Times. https://www.ft.com/content/5b942b07-01cc-4e12-b803-7eb1507f6eec
22. Friedlingstein, P., O'Sullivan, M., Jones, M.W., Andrew, R.M., Gregor, L., Hauck, J., Le Quéré, C., Luijkx, I.T., Olsen, A., Peters, G.P., Peters, W., Pongratz, J., Schwingshackl, C., Sitch, S., Canadell, J.G., Ciais, P., Jackson, R.B., Alin, S.R., Alkama, R. and Arneth, A. (2022). Global Carbon Budget 2022. Earth System Science Data, 14(11), pp.4811–4900. doi:10.5194/essd-14-4811-2022.
23. G Smith, T. (n.d.). API Reference — pmdarima 2.0.2 documentation. Alkaline Ml.com. Retrieved January 6, 2023, from https://alkalineml.com/pmdarima/modules/classes.html#
24. Gayne, D. (2022, November 11). Social housing repairs boost helps construction defy UK economic slump. Housing Today. https://www.housingtoday.co.uk/news/social-housing-repairs-boost-helps-construction-defy-uk-economic-slump/5120391.article
25. Grundy, A. (2021, July 29). Renewables made record-breaking 43.1% share of UK power generation in 2020. Current. https://www.current-news.co.uk/news/renewables-take-home-record-breaking-43-1-of-uk-generation-in-2020
26. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. In otexts.com (2nd edition, p. 8.1). OTexts. https://otexts.com/fpp2/stationarity.html

27. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. In otexts.com (2nd edition, p. 8.5). OTexts. https://otexts.com/fpp2/non-seasonal-arima.html
28. IEA. (2021). Greenhouse Gas Emissions from Energy Data Explorer. IEA.org. https://www.iea.org/data-and-statistics/data-tools/greenhouse-gas-emissions-from-energy-data-explorer
29. Larson, M. G. (2006). Descriptive Statistics and Graphical Displays. *Circulation*, *114*(1), 76–81. https://doi.org/10.1161/circulationaha.105.584474
30. Leach, G. (1979). Future with less energy. New Sci.; (United Kingdom). [online] Available at: https://www.osti.gov/etdeweb/biblio/6108767.
31. Lee, J. and Cho, Y. (2021). National-scale electricity peak load forecasting: Traditional, machine learning, or hybrid model? Energy, p.122366. doi:10.1016/j.energy.2021.122366.
32. Mahia, F., Dey, A.R., Masud, M.A. and Mahmud, M.S. (2019). Forecasting Electricity Consumption using ARIMA Model. 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI). doi:10.1109/sti47673.2019.9068076.
33. Met Office. (February 1, 2022). Mean annual temperature in the United Kingdom (UK) from 1990 to 2021 (in degrees Celsius) [Graph]. In Statista. Retrieved January 03, 2023, from https://www.statista.com/statistics/610124/annual-mean-temperature-in-uk/
34. Moody, J. (2019, September 5). What does RMSE really mean? Medium; Towards Data Science. https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e
35. National Archives (n.d.). Open Government Licence. [online] www.nationalarchives.gov.uk. Available at: https://www.nationalarchives.gov.uk/doc/open-government-licence/version/1/open-government-licence.htm.
36. Nau, R. (n.d.). ARIMA(0,0,0)x(0,1,0): Seasonal random walk model. [online] people.duke.edu. Available at: https://people.duke.edu/~rnau/411searw.html [Accessed 12 Dec. 2022].
37. Office for National Statistics (UK). (December 21, 2022). Population of the United Kingdom from 1871 to 2021 [Graph]. In Statista. Retrieved January 03, 2023, from https://www.statista.com/statistics/281296/uk-population/
38. Office for National Statistics (UK). (May 12, 2022). Gross domestic product of the United Kingdom from 1948 to 2021 (in million GBP) [Graph]. In Statista. Retrieved January 03, 2023, from https://www.statista.com/statistics/281744/gdp-of-the-united-kingdom/
39. Office for National Statistics. (2018). Services sector, UK - Office for National Statistics. Ons.gov.uk. https://www.ons.gov.uk/economy/economicoutputandproductivity/output/articles/servicessectoruk/2008to2018
40. Office for National Statistics. (2018, April 30). The 2008 recession 10 years on - Office for National Statistics. Ons.gov.uk. https://www.ons.gov.uk/economy/grossdomesticproductgdp/articles/the2008recession10yearson/2018-04-30

41. Office for National Statistics. (2020, December 9). The impact of the coronavirus so far: the industries that struggled or recovered - Office for National Statistics. Www.ons.gov.uk. https://www.ons.gov.uk/economy/economicoutputandproductivity/output/articles/theimpactofthecoronavirussofartheindustriesthatstruggledorrecovered/2020-12-09
42. ONS. (2022, January 6). Age of the property is the biggest single factor in energy efficiency of homes - Office for National Statistics. Www.ons.gov.uk. https://www.ons.gov.uk/peoplepopulationandcommunity/housing/articles/ageofthepropertyisthebiggestsinglefactorinenergyefficiencyofhomes/2021-11-01
43. Ritchie, H. (2019, January 28). The death of UK coal in five charts. Our World in Data. https://ourworldindata.org/death-uk-coal
44. Ritchie, H. (2022). CO2 emissions dataset: Our sources and methods. [online] Our World in Data. Available at: https://ourworldindata.org/co2-dataset-sources.
45. Ritchie, H., & Roser, M. (2020). Energy. Our World in Data. https://ourworldindata.org/energy-mix#citation
46. SAID, S. E., & DICKEY, D. A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika, 71(3), 599–607. https://doi.org/10.1093/biomet/71.3.599
47. Sung, J., & Monschauer, Y. (2020, May 27). Changes in transport behaviour during the Covid-19 crisis – Analysis. IEA. https://www.iea.org/articles/changes-in-transport-behaviour-during-the-covid-19-crisis
48. TED-Ed. (2017). Can 100% renewable energy power the world? - Federico Rosei and Renzo Rosei. In YouTube. https://www.youtube.com/watch?v=RnvCbquYeIM
49. Thomas, H. (2022, September 6). Capping UK energy prices is no easy task. Financial Times. https://www.ft.com/content/5ce3807d-eed6-4de4-b31b-47b19ef36b67
50. Thomas, N. (2022, October 7). UK households urged to reduce energy use to avoid winter power cuts. Financial Times. https://www.ft.com/content/f0da82e8-4caa-40a5-88ef-591df08bb190
51. UK Government. (2020, October 6). New plans to make UK world leader in green energy. GOV.UK. https://www.gov.uk/government/news/new-plans-to-make-uk-world-leader-in-green-energy
52. Vella, H. (2017, May 22). What's driving the fall in UK energy demand and can it last? Power Technology | Energy News and Market Analysis. https://www.power-technology.com/features/featurewhats-driving-the-fall-in-uk-energy-demand-and-can-it-last-5819050/
53. Wait, R. (2020, November 17). Energy Price Cap and Govt Price Guarantee – The Latest State of Play. Forbes Advisor UK. https://www.forbes.com/uk/advisor/energy/energy-price-caps/
54. Wickham, H., & Stryjewski, L. (2011). 40 years of boxplots. https://vita.had.co.nz/papers/boxplots.pdf
55. Wikipedia Contributors (2018). Long short-term memory. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Long_short-term_memory.
56. Wikipedia Contributors. (2019, April 30). Pearson correlation coefficient. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

57. Wikipedia Contributors. (2019, September 19). Energy efficiency in British housing. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Energy_efficiency_in_British_housing

58. World Bank. (July 20, 2022). United Kingdom: Distribution of gross domestic product (GDP) across economic sectors from 2011 to 2021 [Graph]. In Statista. Retrieved January 07, 2023, from https://www.statista.com/statistics/270372/distribution-of-gdp-across-economic-sectors-in-the-united-kingdom/

59. Yu, J., Chang, W.-S., & Dong, Y. (2022). Building Energy Prediction Models and Related Uncertainties: A Review. Buildings, 12(8), 1284. https://doi.org/10.3390/buildings12081284

60. Zach (2020). How to Calculate Mean Absolute Percentage Error (MAPE) in Excel. [online] Statology. Available at: https://www.statology.org/mape-excel/.

61. Zach. (2020, January 22). What is Considered to Be a “Strong” Correlation? Statology. https://www.statology.org/what-is-a-strong-correlation/

APPENDIX

Analysing UK's energy scenario - Proposal

Problem Statement

Since its utilization, energy has continued to be a very valuable resource. Global geopolitics and economies are affected. The UK is currently experiencing an energy shortage, which increases its reliance on imported energy resources. Energy price caps are in place, but they are merely a temporary fix for a much broader issue. Non-domestic sectors are not subjected to this price cap affecting economic viability (Bolton & Stewart,2022, p. 6). It is significant that nations are working toward energy independence. One of the many suggestions made to enhance pricing has been to alter energy policies and wholesale market pricing (Financial Times,2022). However, it's crucial to assess the whole situation of the UK's energy generation and supply now and in the future in order to comprehend the ramifications of these policies. Policymakers can better promote future energy independence and planning by confirming this information. When we forecast the future energy requirements for various sectors, we may even provide a recommendation for the best energy production strategy that would be sufficient to render the country energy independent. By deeply analysing and understanding consumption of energy among various sectors we may also be able to shape policies that may result in efficient energy planning.

Literature overview

Existing studies:

Mostly analysis on national energy is conducted by the UK governments Department for Business Energy & Industrial Strategy. This may include energy forecasting, statistical studies and visualizations. These reports are published every year or quarterly and are publicly available on the National statistics website. Some of the latest published reports we reviewed include:

- UK ENERGY IN BRIEF 2022 (Department for Business Energy & Industrial Strategy, 2022).
- Quarterly Energy Prices (Department for Business Energy & Industrial Strategy, 2022).
- Energy Consumption in the UK (ECUK) 1970 to 2021 (Department for Business Energy & Industrial Strategy, 2022).

These reports cover detailed study of several topics within the energy sector such as different sources of energy, investments, employability, fuel poverty etc. However, these reports do not discuss about external factors impacting the energy sector. They also do not include forecasting possible future scenarios for national energy sector to suggest selective changes or decision making in energy planning.

Other studies:

Although not associated with energy prediction throughout the nation, there exists several studies regarding prediction models in energy sector. We may utilize the concepts and technical detailing explained in these papers to support our studies. Some of the publications reviewed includes the following.

- Multivariate time series prediction by RNN architectures for energy consumption forecasting (Amalou et al., 2022)
- Building Energy Prediction Models and Related Uncertainties: A Review (Yu et al., 2022).

Methodology

The study will be based on Secondary Data Analysis / Archival Study /Referring government reports and news articles. Exploratory Data analysis and visualizations will be used to understand changes in consumption as per the data acquired. After this we identify corelating factors that affect consumption. This can be done using Correlation coefficient. After selecting factors that are directly related to consumption, for example: In domestic sector, UK's temperature can be corelated to its consumption. If a year had lower temperatures there might be an increase in consumption. Our goal is to find such variable factors and for understanding the energy sector. For prediction we will be using a timeseries forecasting then test out the model. For example: what models can predict energy consumption in domestic sector. Now we may test our models for accuracy using standard matrixes and then predict future requirements. Based on our predicted results we may now suggest plans for energy industries improvement and police building (Mostly focusing on qualitative study at this point).

Data & Ethics

All data used will be sourced from publicly available datasets. Office for National statistics and data.gov.uk will be our primary sources for most data regarding energy in UK. All information presented here are under 'Open Government Licence'. This allows the use of data for commercial and non-commercial purposes. All sources of data and information will be given as references throughout the research.

License: https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

Analytical outcome

After the research we aim to provide the following results.

- Detailed description of Energy consumption and uses.
- A tested model to predict Energy usage.
- Suggestions and recommendations for Energy planning and policies.

Project plan

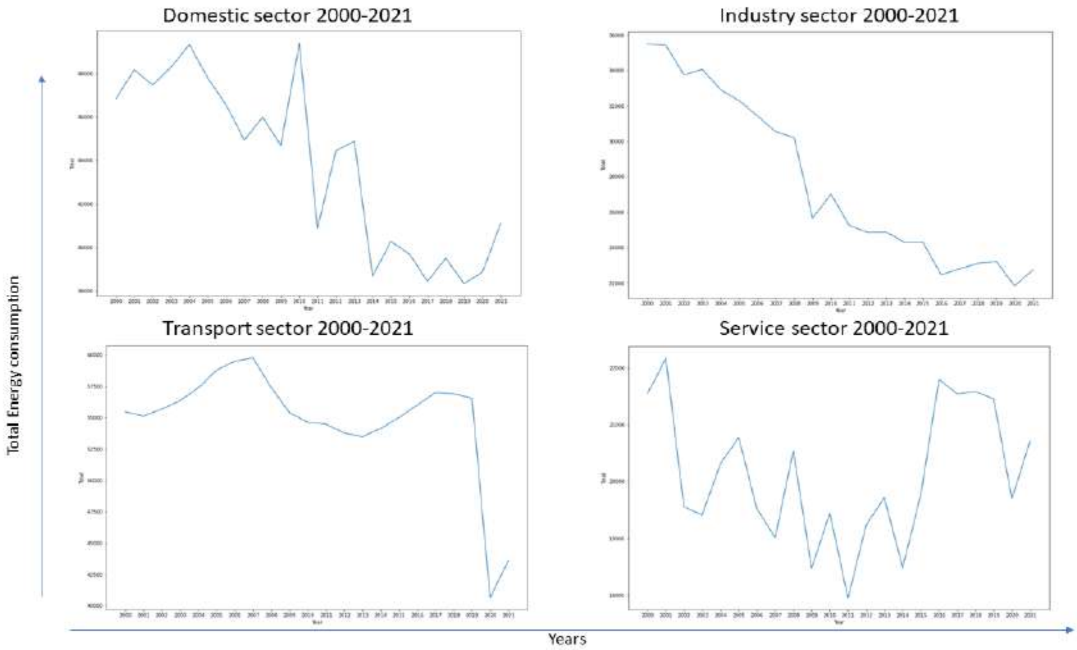
Global primary energy consumption 2021, by country
Primary energy consumption worldwide in 2021, by country (in exajoules)

China*	157.65
United States	92.97
India	35.43
Russia	31.30
Japan	17.74
Canada	13.94
Germany	12.64
South Korea	12.58
Brazil	12.57
Iran	12.19
Saudi Arabia	10.82
France	9.41
Indonesia	8.31
United Kingdom	7.18
Turkey	6.83
Mexico	6.79
Italy	6.36
Australia	5.72
Spain	5.59
Thailand	5.11

(BP,2022)

- UK is the world's 14th largest energy consuming nation-2021.

Understanding the functioning of the UK energy sector and the equilibrium between energy production and consumption is the first step in our process.



visualization created using python, Data source: (National statistics,2021)

- Most needs of energy can be classified into 4 sectors: Industry, domestic, services and transportation. Having look at the timeseries of energy consumption in these sectors we can identify the historic use of energy in UK.
- The variations in energy consumption are highly dependent on a wide variety of factors. We cannot foresee all the events influencing the change in consumption within these sectors however we can further analyse each sector in depth to understand its nature of change to predict possible future variations. For example, in the graph representing Transportation sector we may find a sudden drop in energy consumption during 2020, this is directly influenced by the covid pandemic which drastically decreased the number of vehicles in the UK roads (The Guardian,2020).
- After understanding the factors that influence changes in consumption, we may gain understanding of variables to consider. If sufficient variables are found we can even include then in our prediction model. We will test on forecasting models to compare which one performs with high accuracy to predict future energy demands.
- After this we will compare current energy consumption within the country while also considering output from our analysis. Here we may gain a better understanding of requirements for UK energy structures to perform efficiently and improve policy structures.

REFERENCE

Bolton, P. Stewart, I. (2022) Domestic energy prices. House of Commons Library Briefing Paper no. 9491. London: House of Commons Library. [Online] https://researchbriefings.files.parliament.uk/documents/CBP-9491/CBP-9491.pdf

Financial Times. (2022, September 6). Capping UK energy prices is no easy task. https://www.ft.com/content/5ce3807d-eed6-4de4-b31b-47b19ef36b67

BP. (2022). bp Statistical Review of World Energy 2022 | 71. https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2022-full-report.pdf

National statistics. (2021). Energy consumption in the UK 2021 [Dataset].https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1092940/2021_Consumption_tables_July_2022_update.xlsx

editor, D. C. E. (2020, April 3). UK road travel falls to 1955 levels as Covid-19 lockdown takes hold. The Guardian. https://www.theguardian.com/uk-news/2020/apr/03/uk-road-travel-falls-to-1955-levels-as-covid-19-lockdown-takes-hold-coronavirus-traffic

Department for Business Energy & Industrial Strategy (2022). Energy Consumption in the UK (ECUK) 1970 to 2021. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1061644/Energy_Consumption_in_the_UK_2021.pdf

Department for Business Energy & Industrial Strategy (2022). Quarterly Energy Prices. [April-June 2022] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1107499/quarterly_energy_prices_uk_september_2022.pdf

Department for Business Energy & Industrial Strategy (2022). UK ENERGY IN BRIEF 2022. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1094025/UK_Energy_in_Brief_2022.pdf

Amalou, I., Mouhni, N. and Abdali, A. (2022) "Multivariate time series prediction by RNN architectures for energy consumption forecasting". Available at: https://doi.org/10.1016/j.egyr.2022.07.139.

Yu, J., Chang, W.-S. and Dong, Y. (2022) "Building energy prediction models and related uncertainties: A Review". Available at: https://doi.org/10.3390/buildings12081284.

PYTHON CODE

In [36]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [37]:
```
industry_df = pd.read_csv(r"data\energy_consuption_industry.csv")
domestic_df = pd.read_csv(r"data\energy_consuption_domestic.csv")
services_df = pd.read_csv(r"data\energy_consuption_services.csv")
transport_df = pd.read_csv(r"data\energy_consuption_transport.csv")
```

Unit : **Thousand tonnes of oil equivalent**

In [6]:
```
domestic_df.head(10)
```

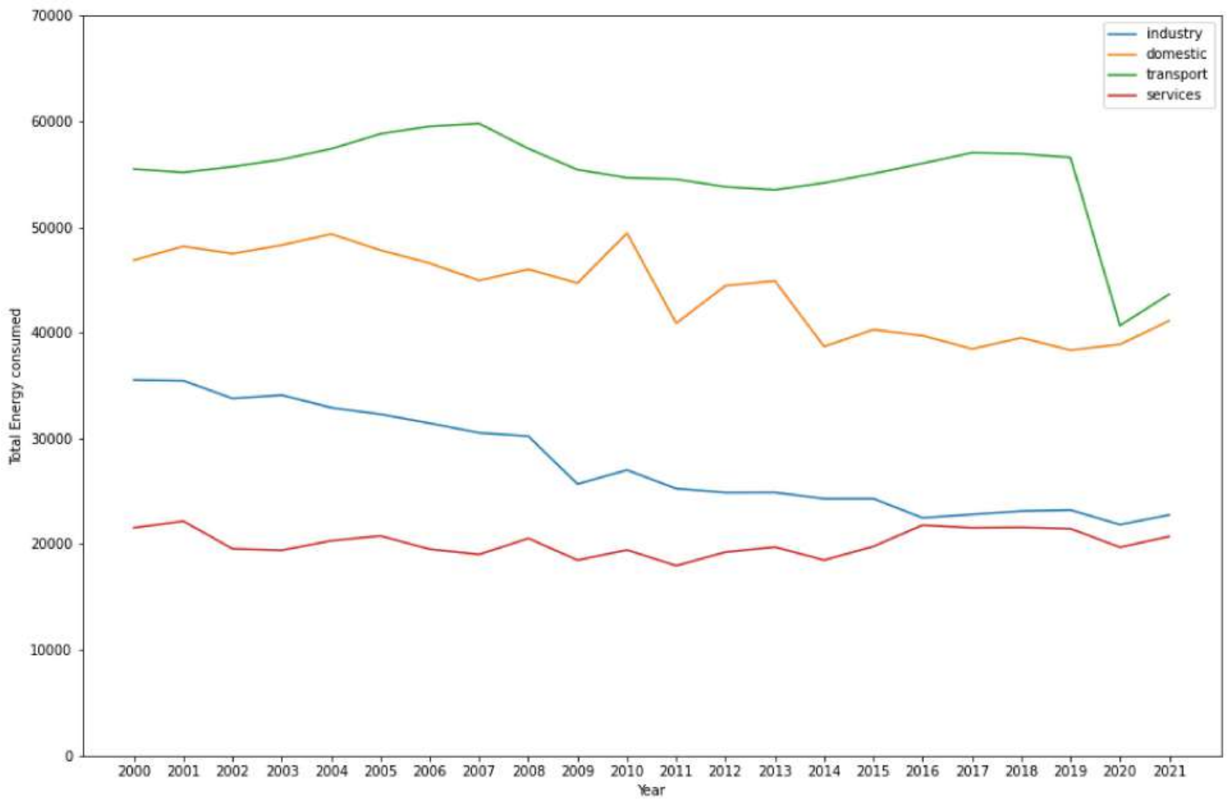
Out[6]:

	Year	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum	Total
0	1970	14242	1761	1975	8922	6622	NaN	NaN	3363	36884
1	1971	12164	1136	2156	9900	6937	NaN	NaN	3328	35621
2	1972	10602	849	2144	11359	7471	NaN	NaN	3836	36261
3	1973	10565	778	2053	12129	7849	NaN	NaN	4202	37576
4	1974	9968	821	1955	13562	7963	NaN	NaN	3733	38002
5	1975	8517	645	1778	14840	7670	NaN	NaN	3612	37062
6	1976	7910	549	1640	15602	7318	NaN	NaN	3615	36634
7	1977	8136	534	1589	16600	7386	NaN	NaN	3653	37898
8	1978	7476	471	1464	18291	7378	NaN	NaN	3610	38689
9	1979	7688	479	1431	20718	7711	NaN	NaN	3539	41566

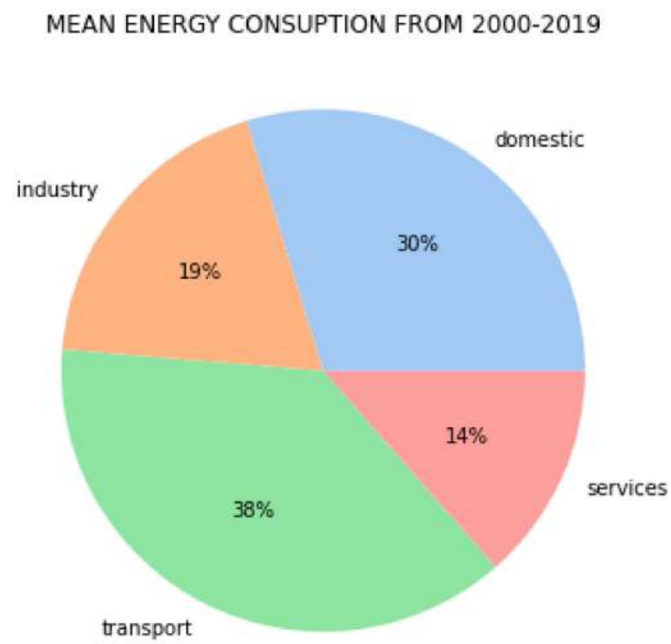
Type *Markdown* and LaTeX: α^2

Compare Energy use within each sector

In [4]:
```
plt.figure(figsize=(15,10))
a=sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y=industry_df[
a.set_xticks(list(industry_df[industry_df['Year']>1999]['Year']));
a.set_ylim([0,70000]);
a.set(
    ylabel='Total Energy consumed')
b=sns.lineplot(data=domestic_df[domestic_df['Year']>1999],x='Year',y=domestic_df[
c=sns.lineplot(data=transport_df[transport_df['Year']>1999],x='Year',y=transport_
d=sns.lineplot(data=services_df[services_df['Year']>1999],x='Year',y=services_df[
plt.legend(['industry','domestic','transport','services'])
plt.show()
```



In [5]:
```
#CALCULATE MEAN CONSUPTION FROM 2000-2019 (Excluding pandemic period)
ed = domestic_df[(domestic_df['Year']<2020) & (domestic_df['Year']>1999)]['Total'
es = services_df[(services_df['Year']<2020) & (services_df['Year']>1999)]['Total'
et = transport_df[(domestic_df['Year']<2020) & (transport_df['Year']>1999)]['Tota
ei = industry_df[(industry_df['Year']<2020) & (industry_df['Year']>1999)]['Total'
data= [ed,ei,et,es]

plt.figure(figsize=(8,6))
plt.title("MEAN ENERGY CONSUPTION FROM 2000-2019")
labels = ['domestic','industry','transport','services']

#define Seaborn color palette to use
colors = sns.color_palette('pastel')[0:5]

#create pie chart
plt.pie(data, labels = labels, colors = colors, autopct='%.0f%%')
plt.show()
```



In [6]:

```
edm = domestic_df[domestic_df['Year']>1999]['Total'].mean()
esm = services_df[services_df['Year']>1999]['Total'].mean()
etm = transport_df[transport_df['Year']>1999]['Total'].mean()
eim = industry_df[industry_df['Year']>1999]['Total'].mean()

total_mean_2k_2k21 = edm+eim+etm+esm
diff=[]
t =[]
for i in range(2000,2022):
    total = int(domestic_df[domestic_df['Year']==i]['Total'])+int(services_df[ser
#     t.append((total,i))
    diff.append((i,total-total_mean_2k_2k21))
diff
# min(t)
```

Out[6]:

```
[(2000, 12848.045454545441),
 (2001, 14408.045454545441),
 (2002, 9959.045454545441),
 (2003, 11630.045454545441),
 (2004, 13419.045454545441),
 (2005, 13158.045454545441),
 (2006, 10524.045454545441),
 (2007, 7742.045454545441),
 (2008, 7639.045454545441),
 (2009, -2275.9545454545587),
 (2010, 3979.0454545454413),
 (2011, -7929.954545454559),
 (2012, -4186.954545454559),
 (2013, -3527.9545454545587),
 (2014, -10907.954545454559),
 (2015, -7144.954545454559),
 (2016, -6526.954545454559),
 (2017, -6718.954545454559),
 (2018, -5405.954545454559),
 (2019, -6953.954545454559),
 (2020, -25423.95454545456),
 (2021, -18302.95454545456)]
```

In [7]:

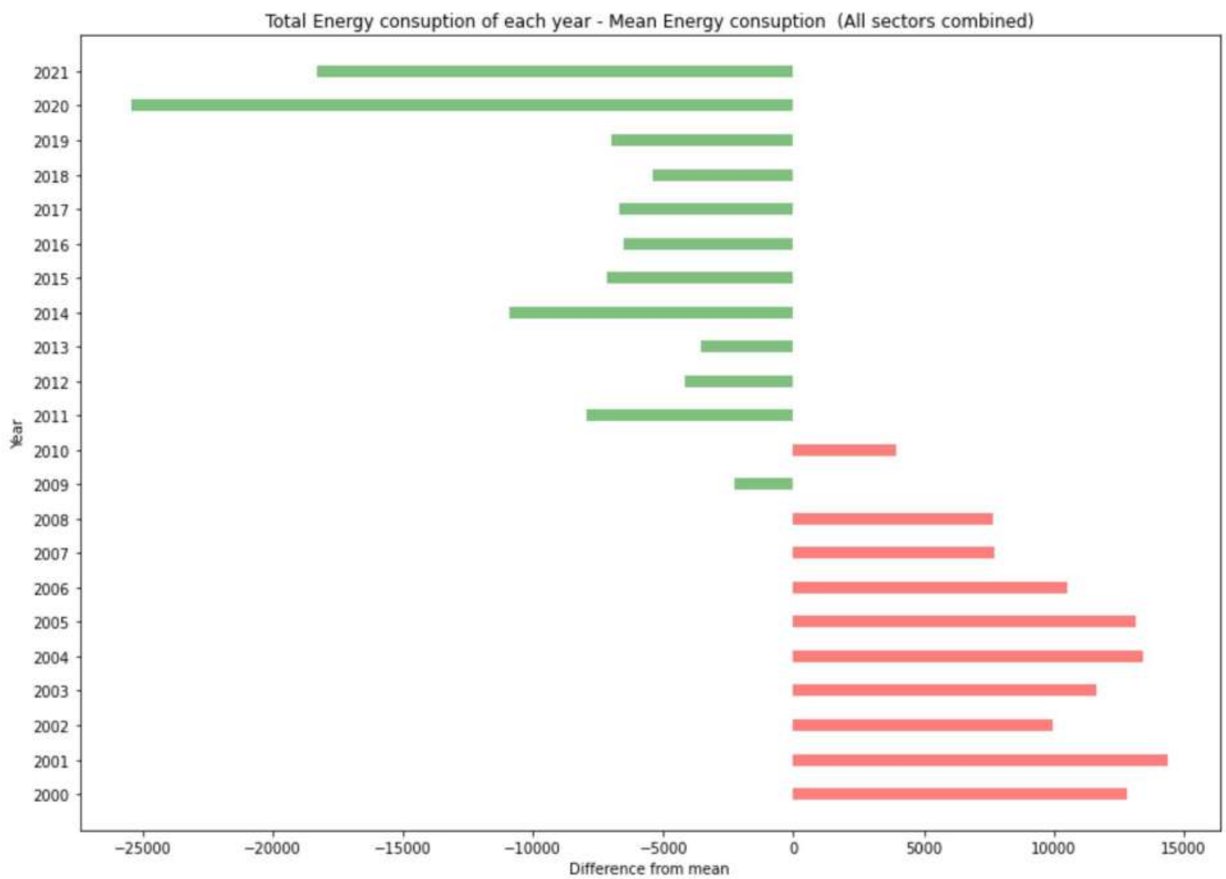
```
mean_diff = pd.DataFrame(diff).rename(columns = {0:'Year',1:'diffrence'})
```

In [8]:
```
mean_diff
```

Out[8]:

	Year	diffrence
0	2000	12848.045455
1	2001	14408.045455
2	2002	9959.045455
3	2003	11630.045455
4	2004	13419.045455
5	2005	13158.045455
6	2006	10524.045455
7	2007	7742.045455
8	2008	7639.045455
9	2009	-2275.954545
10	2010	3979.045455
11	2011	-7929.954545
12	2012	-4186.954545
13	2013	-3527.954545
14	2014	-10907.954545
15	2015	-7144.954545
16	2016	-6526.954545
17	2017	-6718.954545
18	2018	-5405.954545
19	2019	-6953.954545
20	2020	-25423.954545
21	2021	-18302.954545

In [9]:
```
for i in range(22):
    # Colour of bar chart is set to red if the sales
    # is < 60000 and green otherwise
    mean_diff['colors'] = ['green' if float(
        x) < 0 else 'red' for x in mean_diff['diffrence']]

plt.figure(figsize=(14, 10))
# Plotting the horizontal lines
ax = plt.axes()
plt.ylabel('Year')
plt.xlabel('Difference from mean')
plt.title('Total Energy consuption of each year - Mean Energy consuption  (All se
new = plt.hlines(y=mean_diff['Year'],color=mean_diff.colors, xmin=0, xmax=mean_di
ax.set_yticks(list(industry_df[industry_df['Year']>1999]['Year']))
plt.show(new)
```

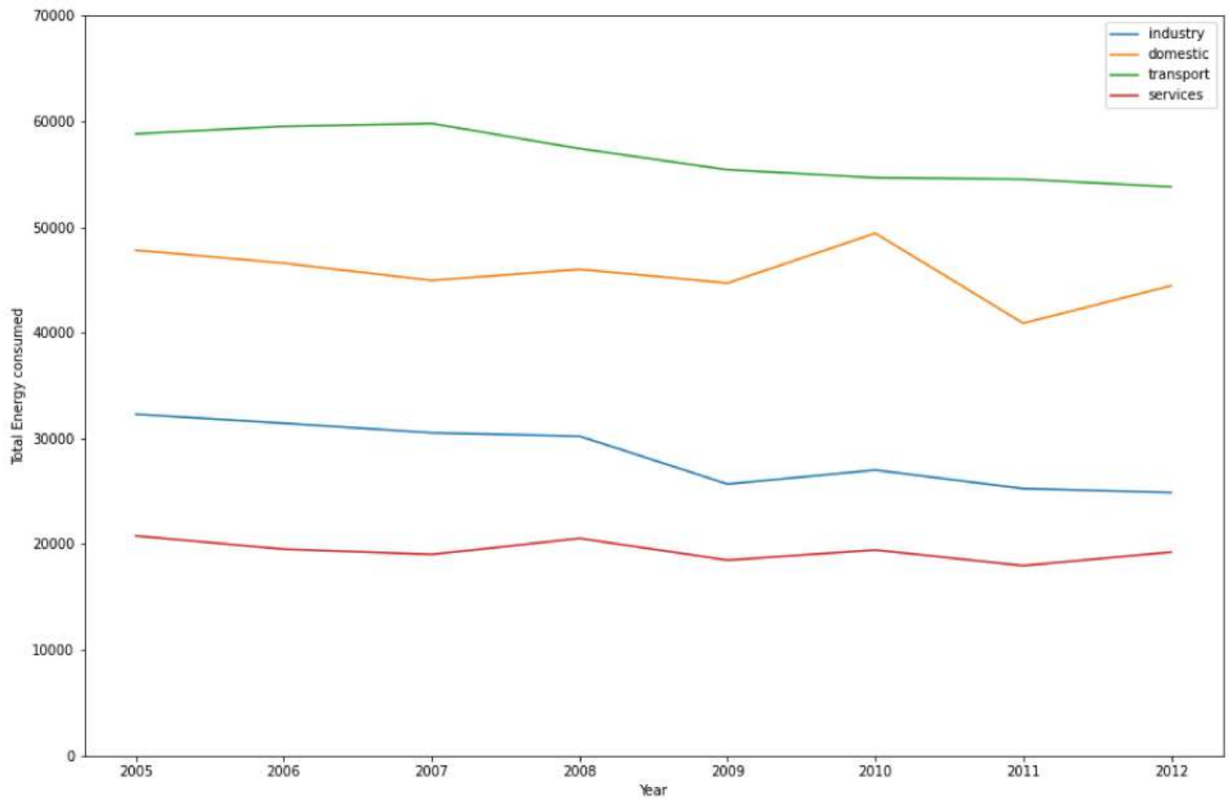


In [10]:
```
#ALL YEARS 1970-2021


# edmx = domestic_df['Total'].mean()
# esmx = services_df['Total'].mean()
# etmx = transport_df['Total'].mean()
# eimx = industry_df['Total'].mean()

# total_mean_2k_2k21 = edmx+eimx+etmx+esmx
# a=[]
# for i in range(1970,2022):
#     total = int(domestic_df[domestic_df['Year']==i]['Total'])+int(services_df[s
#     a.append((i,total-total_mean_2k_2k21))

# mean_diff_all = pd.DataFrame(a).rename(columns = {0:'Year',1:'diffrence'})

# for i in range(22):
#     # Colour of bar chart is set to red if the sales
#     # is < 60000 and green otherwise
#     mean_diff_all['colors'] = ['green' if float(
#         x) < 0 else 'red' for x in mean_diff_all['diffrence']]

# plt.figure(figsize=(14, 10))
# # Plotting the horizontal lines
# ax = plt.axes()
# plt.ylabel('Year')
# plt.xlabel('Difference from mean')
# plt.title('Total Energy consuption of each year - Mean Energy consuption  (All
# new = plt.hlines(y=mean_diff_all['Year'],color=mean_diff_all.colors, xmin=0, xm
# ax.set_yticks(list(industry_df['Year']))
# plt.show(new)
```

In [11]:
```
total_mean_2k_2k19 = ed+ei+et+es
total_2021 = int(domestic_df[domestic_df['Year']==2021]['Total'])+int(services_df
total_2000 = int(domestic_df[domestic_df['Year']==2000]['Total'])+int(services_df
print("Total energy consuption of 2021 : ",total_2021)
print("Total mean energy consuption from 2000-2019 : ",total_mean_2k_2k19)
print("Total energy consuption of 2000 : ",total_2000)
```

```
Total energy consuption of 2021 :  128214
Total mean energy consuption from 2000-2019 :  148703.3
Total energy consuption of 2000 :  159365
```

In [12]:
```
plt.figure(figsize=(15,10))
a=sns.lineplot(data=industry_df[(industry_df['Year']>2004)&(industry_df['Year']<2
a.set_xticks(list(industry_df[(industry_df['Year']>2004)&(industry_df['Year']<201
a.set_ylim([0,70000]);
a.set(
    ylabel='Total Energy consumed')
b=sns.lineplot(data=domestic_df[(domestic_df['Year']>2004)&(domestic_df['Year']<2
c=sns.lineplot(data=transport_df[(transport_df['Year']>2004)&(transport_df['Year'
d=sns.lineplot(data=services_df[(services_df['Year']>2004)&(services_df['Year']<2
plt.legend(['industry','domestic','transport','services'])
plt.show()
```



In [13]:
```
dom_rc = domestic_df[(domestic_df['Year']>2006)&(domestic_df['Year']<2013)][['Yea
ser_rc = services_df[(services_df['Year']>2006)&(services_df['Year']<2013)][['Yea
tra_rc = transport_df[(transport_df['Year']>2006)&(transport_df['Year']<2013)][['
ind_rc = industry_df[(industry_df['Year']>2006)&(industry_df['Year']<2013)][['Yea
```

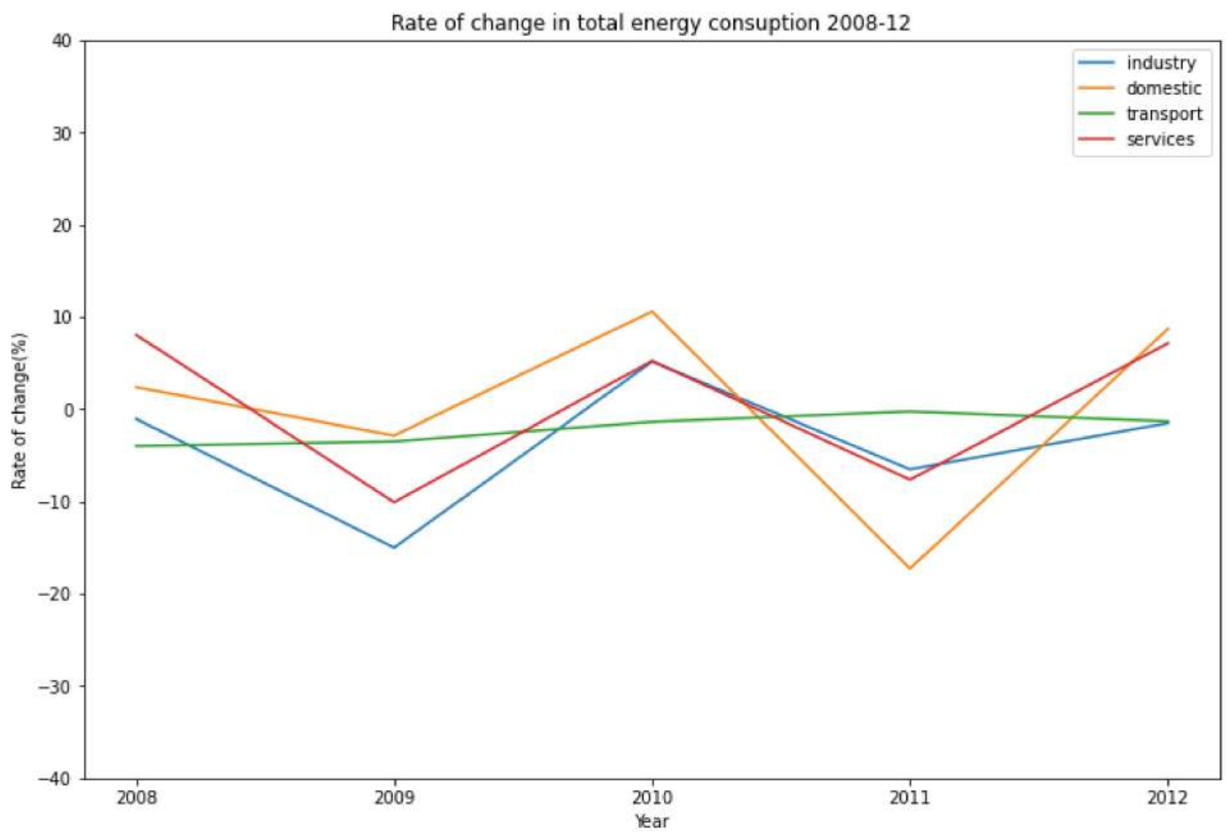
In [14]:
```
dom_rc['rate of change %'] = domestic_df[(domestic_df['Year']>2006)&(domestic_df[
ser_rc['rate of change %'] = services_df[(services_df['Year']>2006)&(services_df[
tra_rc['rate of change %'] = transport_df[(transport_df['Year']>2006)&(transport_
ind_rc['rate of change %'] = industry_df[(industry_df['Year']>2006)&(industry_df[
```

In [15]:
```
dom_rc
```

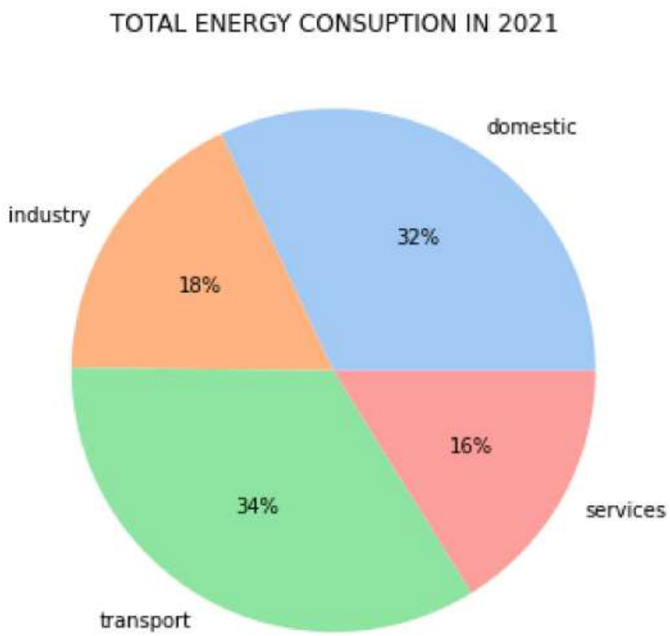
Out[15]:

	Year	Total	rate of change %
37	2007	44932	NaN
38	2008	45998	2.372474
39	2009	44685	-2.854472
40	2010	49410	10.574018
41	2011	40883	-17.257640
42	2012	44441	8.702884

In [16]:
```
plt.figure(figsize=(12,8))
a=sns.lineplot(data=ind_rc,x='Year',y='rate of change %')
a.set_xticks(list(ind_rc[(ind_rc['Year']>2006)&(ind_rc['Year']<2013)]['Year']));
a.set_ylim([-40,40]);
a.set(
    ylabel='Rate of change(%)')
a.set(title='Rate of change in total energy consuption 2008-12')
b=sns.lineplot(data=dom_rc,x='Year',y='rate of change %')
c=sns.lineplot(data=tra_rc,x='Year',y='rate of change %')
d=sns.lineplot(data=ser_rc,x='Year',y='rate of change %')
plt.legend(['industry','domestic','transport','services'])
plt.show()
```



In [17]:
```
# AS PER 2021 RATIO OF TOTAL ENERGY CONSUMED IN THE UK
plt.figure(figsize=(8,6))
data = [int(domestic_df[domestic_df['Year']==2021]['Total']),int(industry_df[indu
       int(transport_df[transport_df['Year']==2021]['Total']),int(services_df[ser
labels = ['domestic','industry','transport','services']
plt.title("TOTAL ENERGY CONSUPTION IN 2021")
#define Seaborn color palette to use
colors = sns.color_palette('pastel')[0:5]

#create pie chart
plt.pie(data, labels = labels, colors = colors, autopct='%.0f%%')
plt.show()
```

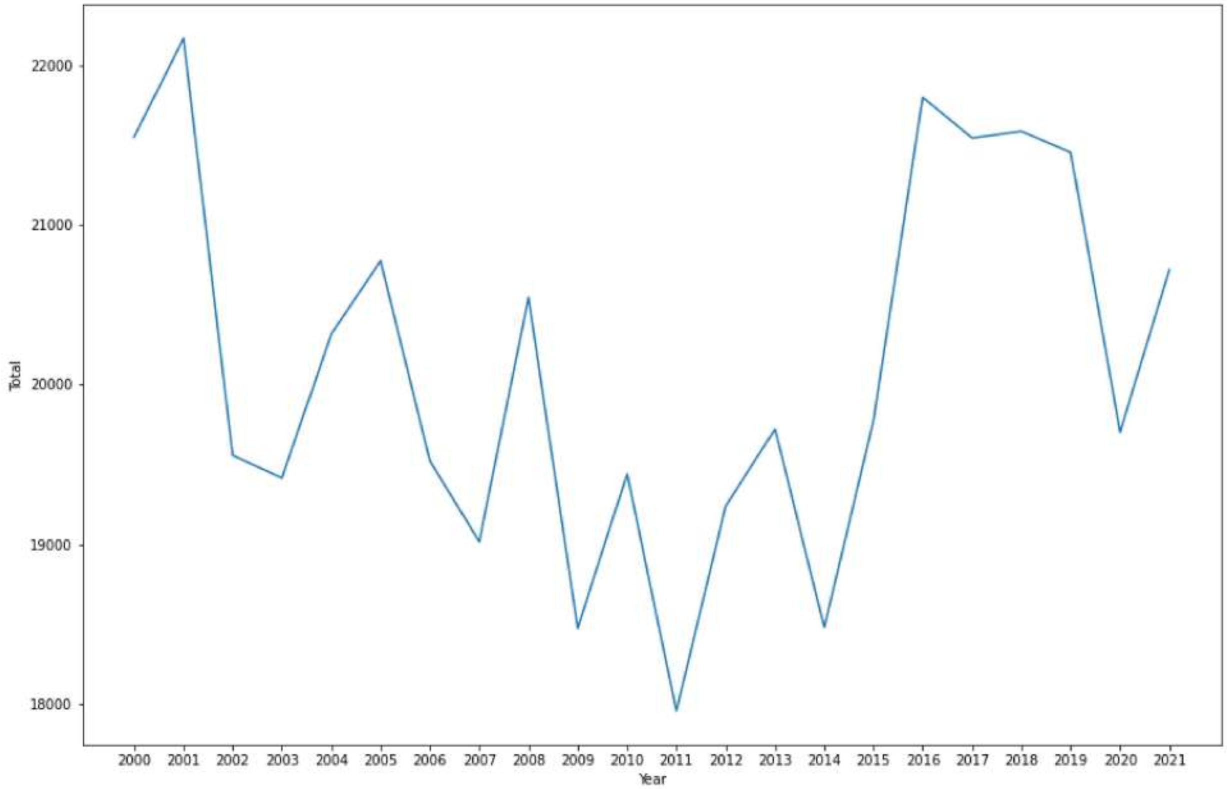


In [ ]:

Analyse rate of change in energy on each year, for each sector (Investigate factors influencing)

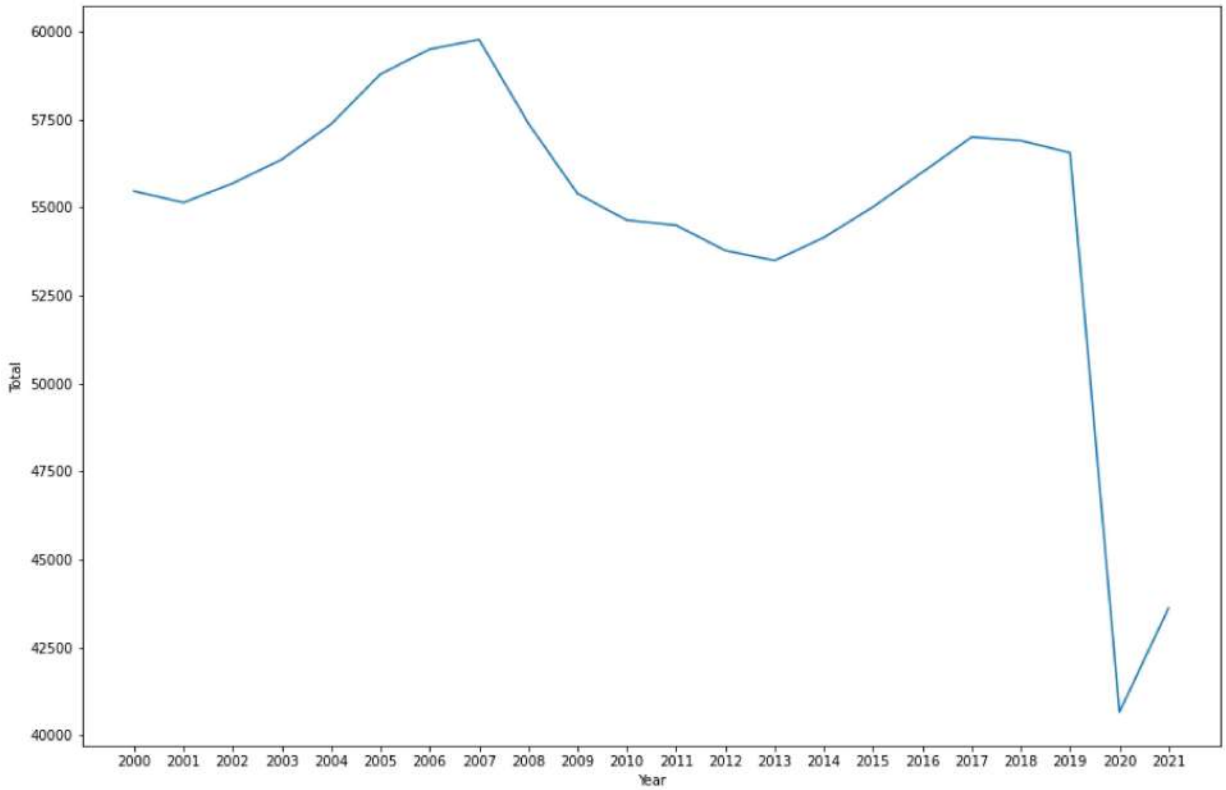
ENERGY consuption in Services sector

In [18]:
```
plt.figure(figsize=(15,10))
g=sns.lineplot(data=services_df[services_df['Year']>1999],x='Year',y=services_df[
g.set_xticks(list(services_df[services_df['Year']>1999]['Year']));
```



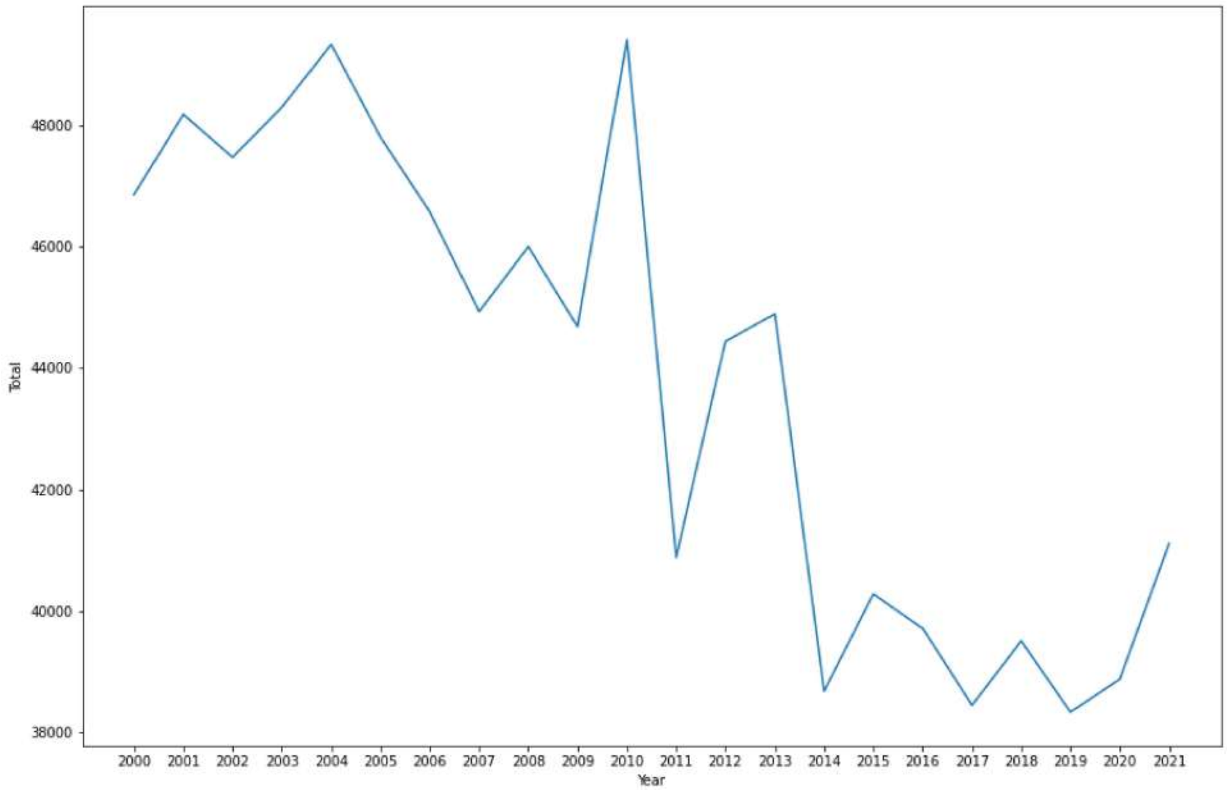
ENERGY consuption in Transport sector

In [19]:
```
plt.figure(figsize=(15,10))
g=sns.lineplot(data=transport_df[transport_df['Year']>1999],x='Year',y=transport_
g.set_xticks(list(transport_df[transport_df['Year']>1999]['Year']));
```



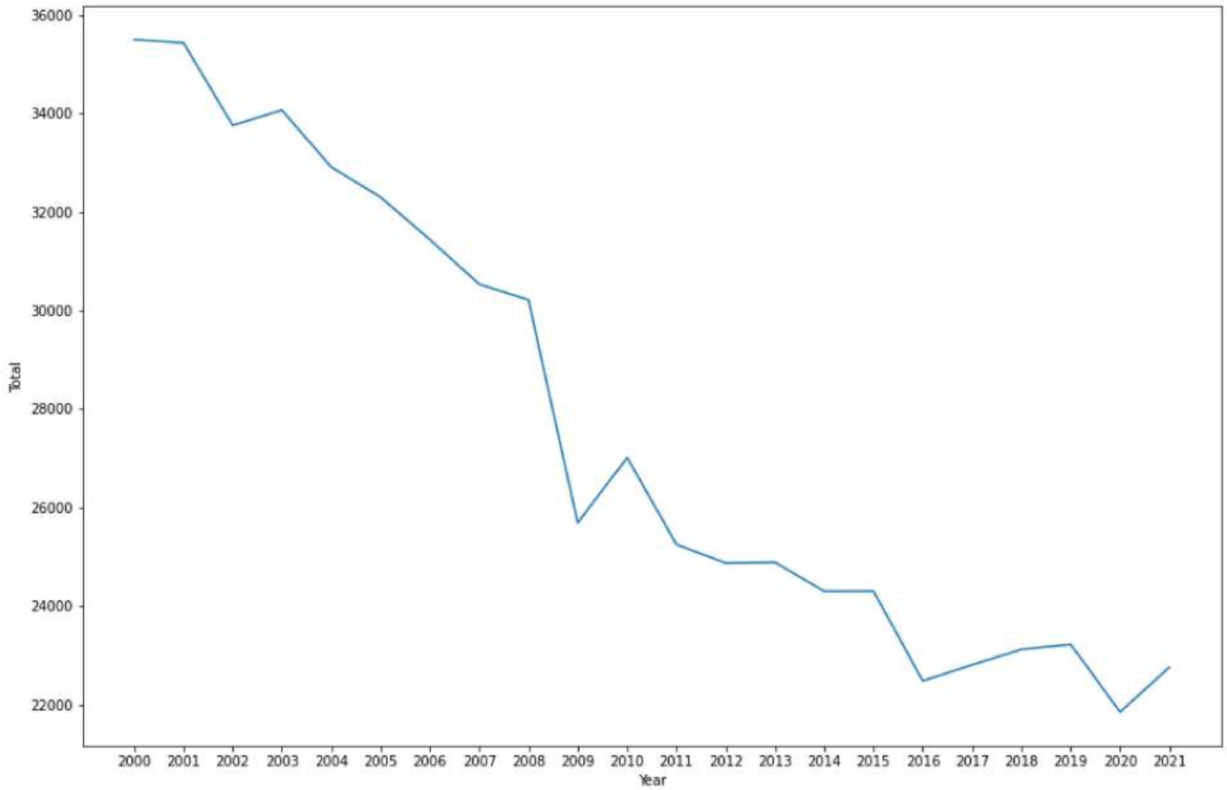
ENERGY consuption in Domestic sector

In [20]:
```
plt.figure(figsize=(15,10))
g=sns.lineplot(data=domestic_df[domestic_df['Year']>1999],x='Year',y=domestic_df[
g.set_xticks(list(domestic_df[domestic_df['Year']>1999]['Year']));
```



ENERGY consuption in industry sector

In [21]:
```
plt.figure(figsize=(15,10))
g=sns.lineplot(data=domestic_df[industry_df['Year']>1999],x='Year',y=industry_df[
g.set_xticks(list(industry_df[industry_df['Year']>1999]['Year']));
```



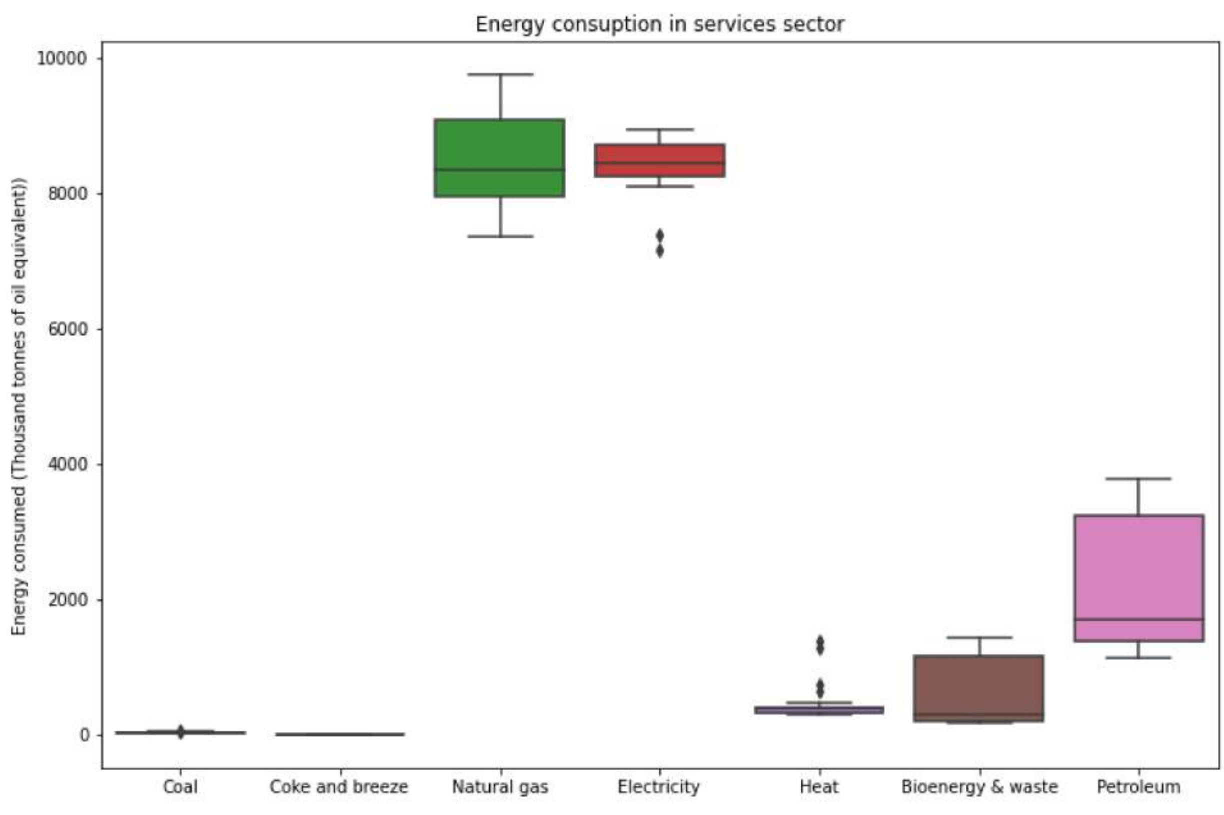
Analyse diffrent modes of energy within each sector

In [22]:
```
services_df[services_df['Year']>1999].describe()
```

Out[22]:

	Year	Coal	Coke and breeze	Natural gas	Electricity	Heat	Bioenergy & waste	Petı
count	22.000000	22.000000	22.0	22.000000	22.000000	22.000000	22.000000	22.(
mean	2010.500000	27.545455	0.0	8487.590909	8384.454545	477.090909	588.727273	2158.ε
std	6.493587	11.396438	0.0	778.844728	430.362940	297.865247	500.337142	998.ε
min	2000.000000	14.000000	0.0	7344.000000	7170.000000	289.000000	172.000000	1145.(
25%	2005.250000	21.000000	0.0	7939.250000	8248.250000	310.500000	198.000000	1372.7
50%	2010.500000	25.500000	0.0	8338.500000	8438.000000	388.000000	286.000000	1694.ᵗ
75%	2015.750000	28.000000	0.0	9077.250000	8703.250000	400.250000	1150.500000	3236.2
max	2021.000000	57.000000	0.0	9757.000000	8935.000000	1371.000000	1436.000000	3786.(

In [23]:
```
#understand spred of data
plt.figure(figsize=(12,8))
bs_plt = sns.boxplot(data=services_df[services_df['Year']>1999][services_df.colun
bs_plt.set(
    ylabel='Energy consumed (Thousand tonnes of oil equivalent))',title='Energy c
plt.show(bs_plt)
```



Most use in Electricity and Natural gas

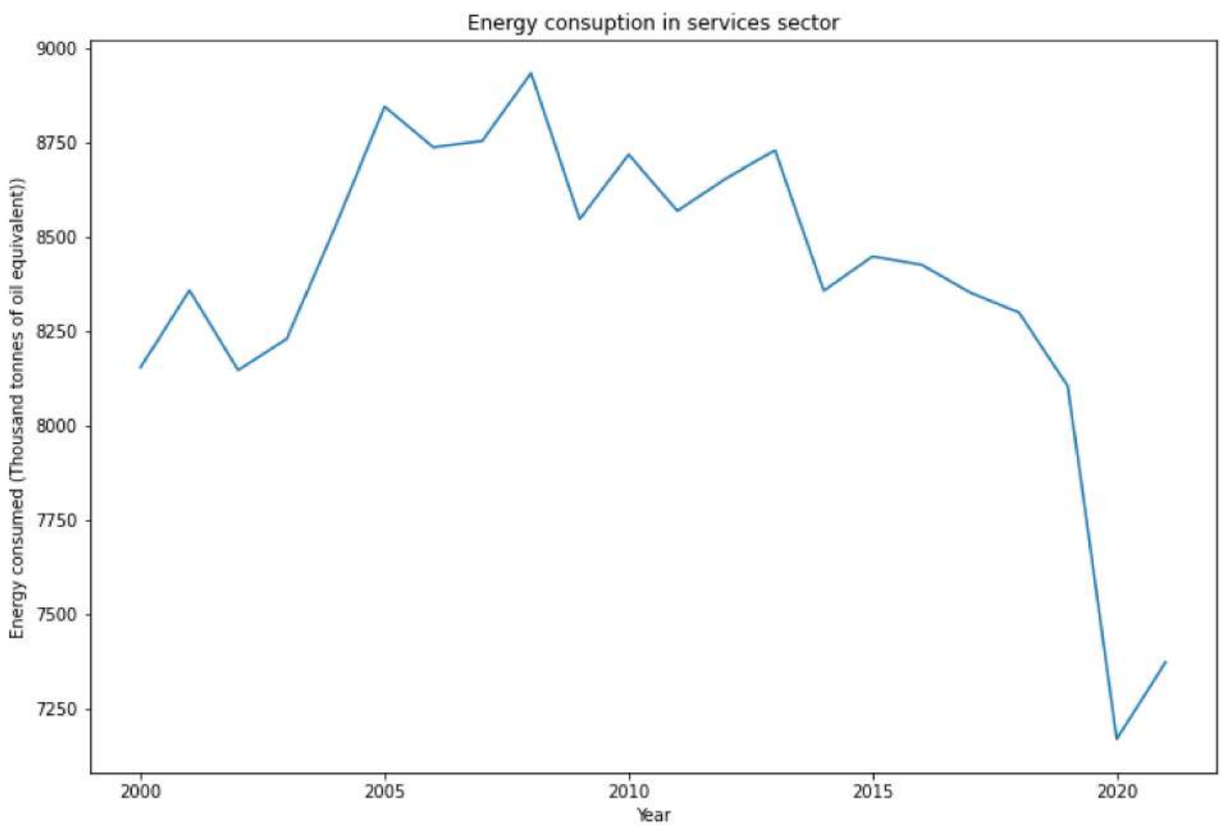
In [24]:
```
test = services_df[(services_df['Year']>1999)&(services_df['Heat']>500)]
test[['Year','Heat']]
```

Out[24]:

	Year	Heat
30	2000	1371
31	2001	1294
32	2002	730
33	2003	648

In [ ]:

In [25]:
```
#understand spred of data
plt.figure(figsize=(12,8))
bs_plt = sns.lineplot(data=services_df[services_df['Year']>1999][services_df.colu
bs_plt.set(
    ylabel='Energy consumed (Thousand tonnes of oil equivalent))',title='Energy c
plt.show(bs_plt)
```

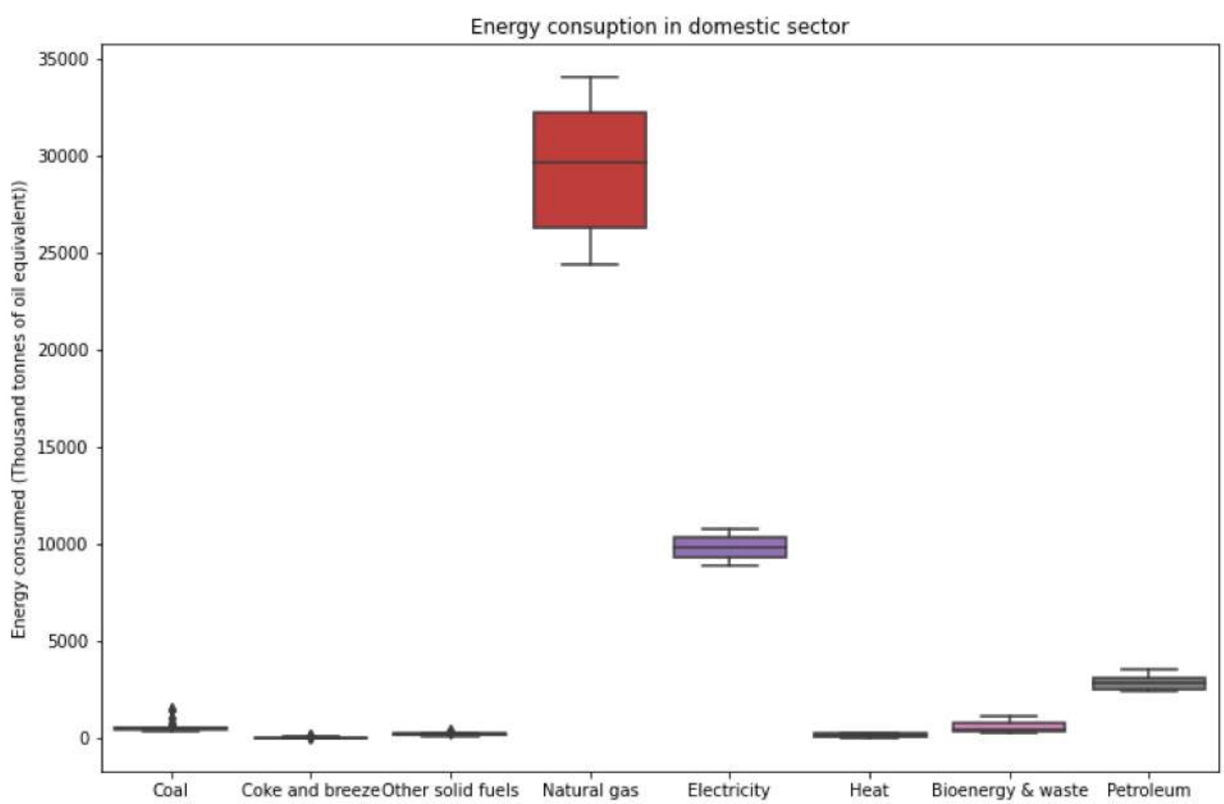


In [26]:
```
domestic_df[domestic_df['Year']>1999].describe()
```

Out[26]:

	Year	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat
count	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000
mean	2010.500000	585.818182	22.409091	206.227273	29302.454545	9849.772727	115.045455
std	6.493587	325.834553	36.121039	59.473829	3221.061060	612.650132	103.738443
min	2000.000000	321.000000	0.000000	122.000000	24393.000000	8918.000000	11.000000
25%	2005.250000	404.500000	0.500000	171.250000	26262.000000	9289.250000	52.000000
50%	2010.500000	486.500000	6.500000	192.000000	29652.000000	9805.500000	52.000000
75%	2015.750000	535.250000	22.000000	227.000000	32223.000000	10314.500000	260.000000
max	2021.000000	1461.000000	127.000000	365.000000	34085.000000	10809.000000	270.000000

In [27]:
```
#understand spred of data
plt.figure(figsize=(12,8))
bs_plt = sns.boxplot(data=domestic_df[domestic_df['Year']>1999][domestic_df.colum
bs_plt.set(
    ylabel='Energy consumed (Thousand tonnes of oil equivalent))',title='Energy c
plt.show(bs_plt)
```



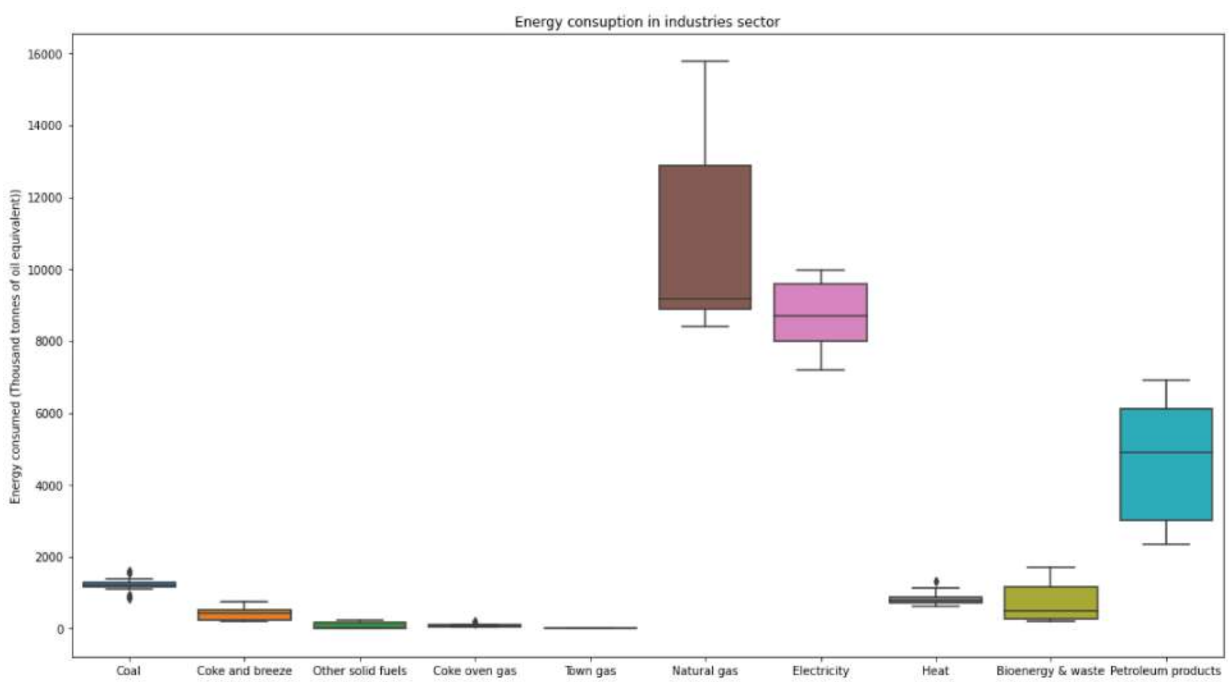
Most use in Natural gas

In [28]:
```
industry_df[industry_df['Year']>1999].describe()
```

Out[28]:

	Year	Coal	Coke and breeze\n	Other solid fuels\n	Coke oven gas	Town gas	Natural gas	Elect
count	22.000000	22.000000	22.000000	22.000000	22.000000	6.0	22.000000	22.0C
mean	2010.500000	1210.363636	420.636364	79.590909	83.954545	0.0	10758.863636	8765.63
std	6.493587	182.111307	170.809988	89.558525	38.439163	0.0	2505.257905	887.06
min	2000.000000	841.000000	193.000000	0.000000	43.000000	0.0	8418.000000	7195.0C
25%	2005.250000	1155.000000	245.250000	0.000000	62.250000	0.0	8884.000000	8003.75
50%	2010.500000	1203.500000	419.000000	17.000000	73.500000	0.0	9163.500000	8691.0C
75%	2015.750000	1289.000000	529.500000	173.250000	95.750000	0.0	12873.500000	9581.25
max	2021.000000	1627.000000	753.000000	225.000000	216.000000	0.0	15773.000000	9976.0C

In [29]:
```
#understand spred of data
plt.figure(figsize=(18,10))
bs_plt = sns.boxplot(data=industry_df[industry_df['Year']>1999][industry_df.colu
bs_plt.set(
    ylabel='Energy consumed (Thousand tonnes of oil equivalent))',title='Energy c
plt.show(bs_plt)
```



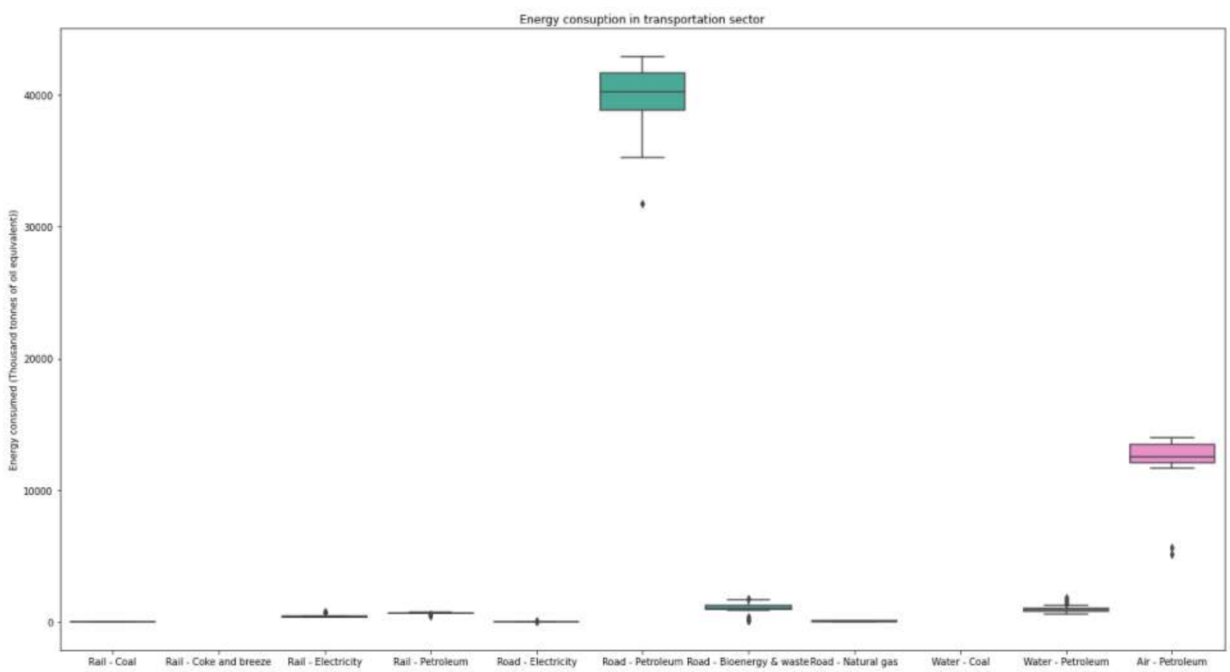
Most use in Natural gas and electricity

In [30]:
```
transport_df[transport_df['Year']>1999].describe()
```

Out[30]:

	Year	Rail - Coal	Rail - Coke and breeze	Rail - Electricity	Rail - Petroleum	Road - Electricity	Road - Petroleum	Roa Bioenergy wa
count	22.000000	22.000000	0.0	22.000000	22.000000	18.000000	22.000000	17.000(
mean	2010.500000	8.454545	NaN	440.363636	644.045455	12.444444	39844.318182	1020.823
std	6.493587	5.261968	NaN	144.559279	48.534207	18.968154	2557.767765	459.134
min	2000.000000	0.000000	NaN	338.000000	472.000000	2.000000	31792.000000	74.000(
25%	2005.250000	4.500000	NaN	347.250000	640.750000	2.000000	38821.250000	958.000(
50%	2010.500000	10.500000	NaN	382.000000	659.000000	2.500000	40194.000000	1038.000(
75%	2015.750000	11.750000	NaN	414.000000	666.750000	14.750000	41641.750000	1243.000(
max	2021.000000	14.000000	NaN	759.000000	700.000000	74.000000	42884.000000	1736.000(

In [31]:
```
#understand spred of data
plt.figure(figsize=(22,12))
bs_plt = sns.boxplot(data=transport_df[transport_df['Year']>1999][transport_df.co
bs_plt.set(
    ylabel='Energy consumed (Thousand tonnes of oil equivalent))',title='Energy c
plt.show(bs_plt)
```



Most use in Petroleum

In [32]:
```
domestic_df.head()
```

Out[32]:

	Year	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum	Total
0	1970	14242	1761	1975	8922	6622	NaN	NaN	3363	36884
1	1971	12164	1136	2156	9900	6937	NaN	NaN	3328	35621
2	1972	10602	849	2144	11359	7471	NaN	NaN	3836	36261
3	1973	10565	778	2053	12129	7849	NaN	NaN	4202	37576
4	1974	9968	821	1955	13562	7963	NaN	NaN	3733	38002

FINDING CORELATIONS

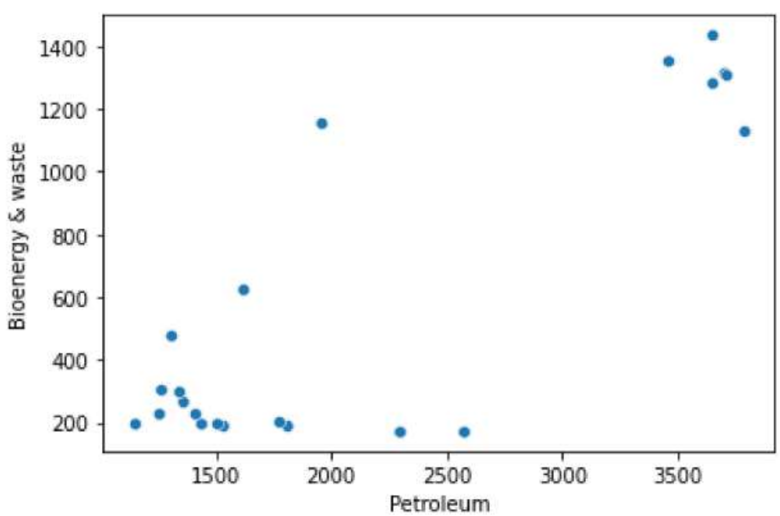
In [33]:
```
services_df[services_df['Year']>1999].tail()
```

Out[33]:

	Year	Coal	Coke and breeze	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum	Total
47	2017	27	0	7937	8353	293	1284	3648	21542
48	2018	26	0	7936	8300	310	1317	3695	21584
49	2019	22	0	8016	8106	291	1309	3709	21452
50	2020	21	0	7409	7170	289	1353	3459	19701
51	2021	22	0	7946	7373	290	1436	3651	20718

In [34]:
```
sns.scatterplot(data=services_df[services_df['Year']>1999],x='Petroleum',y='Bioer
```

Out[34]:
```
<AxesSubplot:xlabel='Petroleum', ylabel='Bioenergy & waste'>
```



In [35]:
```
corr  = services_df[services_df['Year']>1999][services_df.columns[1:8]].corr()
corr.style.background_gradient(cmap='coolwarm')
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[35]:

	Coal	Coke and breeze	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum
Coal	1.000000	nan	0.220732	0.087765	0.575436	-0.215158	0.009825
Coke and breeze	nan	nan	nan	nan	nan	nan	nan
Natural gas	0.220732	nan	1.000000	0.384626	0.540315	-0.618250	-0.352785
Electricity	0.087765	nan	0.384626	1.000000	-0.043546	-0.618017	-0.632118
Heat	0.575436	nan	0.540315	-0.043546	1.000000	-0.468507	-0.117738
Bioenergy & waste	-0.215158	nan	-0.618250	-0.618017	-0.468507	1.000000	0.853211
Petroleum	0.009825	nan	-0.352785	-0.632118	-0.117738	0.853211	1.000000

In [36]:
```
x = services_df[services_df['Year']>1999][services_df.columns[1:8]].mean()
for i in range(len(x)):
    print(str(x[[i]]),"_",(x[i]/sum(x))*100," % of total \n")
```

```
Coal    27.545455
dtype: float64 _ 0.1368767702503083  % of total

Coke and breeze    0.0
dtype: float64 _ 0.0  % of total

Natural gas    8487.590909
dtype: float64 _ 42.17588890846422  % of total

Electricity    8384.454545
dtype: float64 _ 41.663391562428004  % of total

Heat    477.090909
dtype: float64 _ 2.3707237302759667  % of total

Bioenergy & waste    588.727273
dtype: float64 _ 2.9254586275280414  % of total

Petroleum    2158.863636
dtype: float64 _ 10.727660401053454  % of total
```

In [37]:
```
gdp = pd.read_csv(r"data\gdp_uk.csv")
```

In [38]:
```
gdp_ser_comb = pd.merge(gdp,services_df[services_df['Year']>1999], how='outer', c
gdp_ser_comb
```

Out[38]:

	Year	gdp	gdp_roc	unemp_rate	pb	com_turnover	Coal	Coke and breeze	Natural gas	Electricity
0	2000	1632591	3.672369	5.4	3.47	926807.7	57	0	9498	8155
1	2001	1666429	2.072656	5.1	3.50	967114.1	47	0	9726	8359
2	2002	1701811	2.123223	5.2	3.57	993562.0	14	0	8670	8148
3	2003	1753374	3.029890	5.0	3.68	1034772.8	17	0	9177	8231
4	2004	1794677	2.355630	4.8	3.93	1095231.9	19	0	9757	8532
5	2005	1841218	2.593280	4.8	3.93	1194005.5	38	0	9526	8846
6	2006	1888797	2.584105	5.4	4.13	1297630.1	24	0	8655	8738
7	2007	1931663	2.269487	5.3	4.27	1410306.2	19	0	8154	8755
8	2008	1927034	-0.239638	5.7	4.28	1563152.1	21	0	9557	8935
9	2009	1845186	-4.247356	7.6	4.38	1436292.0	53	0	8000	8548
10	2010	1884515	2.131438	7.9	4.48	1588216.8	28	0	8736	8719
11	2011	1911983	1.457563	8.1	4.59	1707264.1	28	0	7344	8570
12	2012	1940087	1.469888	8.0	4.82	1769068.8	17	0	8523	8656
13	2013	1976755	1.890018	7.6	4.91	1918346.8	25	0	8778	8730
14	2014	2035883	2.991165	6.2	5.25	1957614.8	26	0	7381	8358
15	2015	2089276	2.622597	5.4	5.40	1900951.1	27	0	7890	8449
16	2016	2136566	2.263464	4.9	5.50	1966521.2	28	0	8111	8427
17	2017	2182170	2.134453	4.4	5.69	2052687.8	27	0	7937	8353
18	2018	2218196	1.650925	4.1	5.67	2197982.2	26	0	7936	8300
19	2019	2255283	1.671944	3.8	5.87	2274838.5	22	0	8016	8106
20	2020	2046209	-9.270411	4.6	5.98	1955788.8	21	0	7409	7170
21	2021	2198473	7.441273	4.5	5.59	2297596.2	22	0	7946	7373

In [39]:
```
corr  = gdp_ser_comb[gdp_ser_comb['Year']>1999][gdp_ser_comb.columns[1:11]].corr(
corr.style.background_gradient(cmap='coolwarm')
```

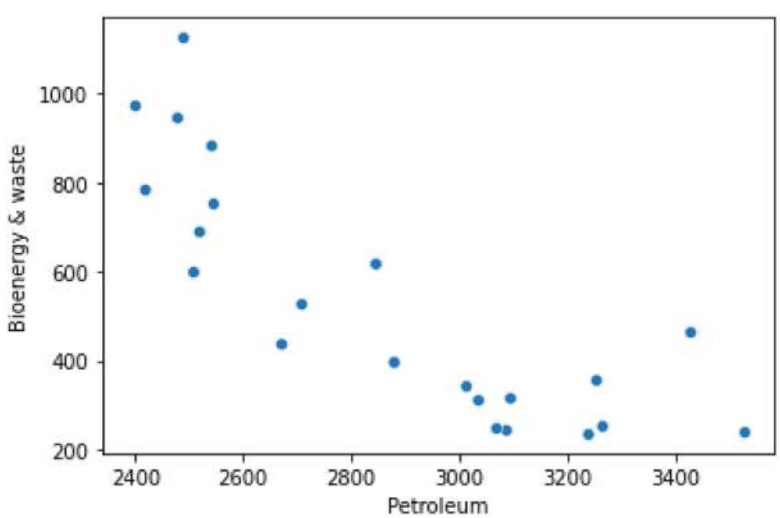
```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[39]:

	gdp	gdp_roc	unemp_rate	pb	com_turnover	Coal	Coke and breeze	Na
gdp	1.000000	0.007077	-0.290962	0.950267	0.959277	-0.363029	nan	-0.67
gdp_roc	0.007077	1.000000	-0.119833	-0.199145	-0.046660	-0.077287	nan	0.25
unemp_rate	-0.290962	-0.119833	1.000000	-0.202868	-0.083832	0.138666	nan	-0.08
pb	0.950267	-0.199145	-0.202868	1.000000	0.963288	-0.298351	nan	-0.76
com_turnover	0.959277	-0.046660	-0.083832	0.963288	1.000000	-0.310130	nan	-0.73
Coal	-0.363029	-0.077287	0.138666	-0.298351	-0.310130	1.000000	nan	0.22
Coke and breeze	nan	nan	nan	nan	nan	nan	nan	
Natural gas	-0.678266	0.250353	-0.084465	-0.761178	-0.731342	0.220732	nan	1.00
Electricity	-0.276501	0.209289	0.452541	-0.425331	-0.314874	0.087765	nan	0.38
Heat	-0.726622	0.176643	-0.022174	-0.665673	-0.670884	0.575436	nan	0.54

In [40]:
```
sns.scatterplot(data=domestic_df[domestic_df['Year']>1999],x='Petroleum',y='Bioer
```

Out[40]: <AxesSubplot:xlabel='Petroleum', ylabel='Bioenergy & waste'>



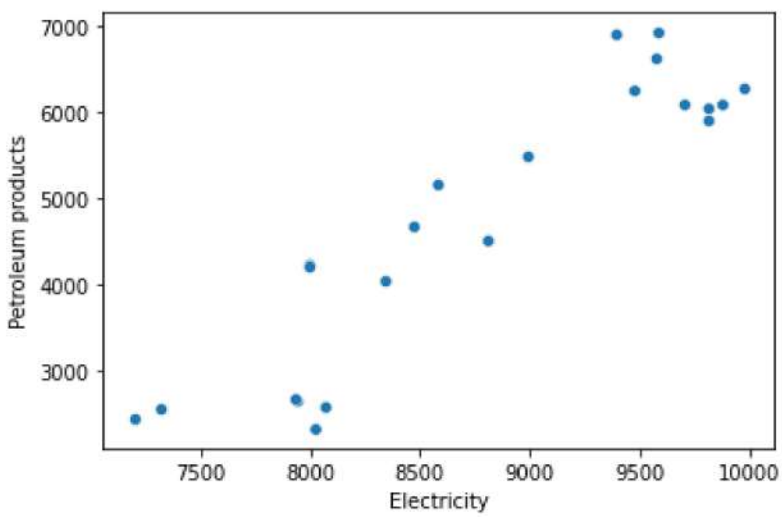
In [41]:
```
corr  = domestic_df[domestic_df['Year']>1999][domestic_df.columns[1:9]].corr()
corr.style.background_gradient(cmap='coolwarm')
```

Out[41]:

	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum
Coal	1.000000	0.784505	0.951028	0.586504	0.267250	-0.523851	-0.652176	0.683145
Coke and breeze	0.784505	1.000000	0.799762	0.596353	0.400675	-0.490157	-0.616761	0.534781
Other solid fuels	0.951028	0.799762	1.000000	0.682307	0.390840	-0.643366	-0.770309	0.771051
Natural gas	0.586504	0.596353	0.682307	1.000000	0.869536	-0.748838	-0.798759	0.929135
Electricity	0.267250	0.400675	0.390840	0.869536	1.000000	-0.770547	-0.785557	0.765162
Heat	-0.523851	-0.490157	-0.643366	-0.748838	-0.770547	1.000000	0.887469	-0.765266
Bioenergy & waste	-0.652176	-0.616761	-0.770309	-0.798759	-0.785557	0.887469	1.000000	-0.833612
Petroleum	0.683145	0.534781	0.771051	0.929135	0.765162	-0.765266	-0.833612	1.000000

In [42]:
```
sns.scatterplot(data=industry_df[industry_df['Year']>1999],x='Electricity',y='Pet
```

Out[42]: <AxesSubplot:xlabel='Electricity', ylabel='Petroleum products'>



In [43]:

```
corr  = industry_df[industry_df['Year']>1999][industry_df.columns[1:11]].corr()
corr.style.background_gradient(cmap='coolwarm')
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[43]:

	Coal	Coke and breeze	Other solid fuels	Coke oven gas	Town gas	Natural gas	Electricity	Heat	Bioer & v
Coal	1.000000	0.429472	0.088278	-0.093532	nan	0.009031	0.259032	0.029122	-0.51
Coke and breeze	0.429472	1.000000	0.845482	0.522612	nan	0.858508	0.780787	0.717962	-0.83
Other solid fuels	0.088278	0.845482	1.000000	0.558536	nan	0.937493	0.904233	0.782213	-0.73
Coke oven gas	-0.093532	0.522612	0.558536	1.000000	nan	0.602511	0.392611	0.411648	-0.20
Town gas	nan	nan	nan	nan	nan	nan	nan	nan	
Natural gas	0.009031	0.858508	0.937493	0.602511	nan	1.000000	0.802882	0.830046	-0.66
Electricity	0.259032	0.780787	0.904233	0.392611	nan	0.802882	1.000000	0.706998	-0.89
Heat	0.029122	0.717962	0.782213	0.411648	nan	0.830046	0.706998	1.000000	-0.61
Bioenergy & waste	-0.514681	-0.831401	-0.739478	-0.204693	nan	-0.662211	-0.890012	-0.611817	1.00
Petroleum products	0.319585	0.864367	0.855793	0.304014	nan	0.794954	0.915133	0.716223	-0.92

In [44]:
```
corr  = transport_df[transport_df['Year']>1999][transport_df.columns[1:12]].corr(
corr.style.background_gradient(cmap='coolwarm')
```

Out[44]:

	Rail - Coal	Rail - Coke and breeze	Rail - Electricity	Rail - Petroleum	Road - Electricity	Road - Petroleum	Road - Bioenergy & waste	Road - Natural gas	W
Rail - Coal	1.000000	nan	-0.760213	-0.139370	-0.043373	-0.272782	0.049269	-0.784103	
Rail - Coke and breeze	nan	nan	nan	nan	nan	nan	nan	nan	
Rail - Electricity	-0.760213	nan	1.000000	0.042336	0.610805	0.191989	0.757009	-0.210177	
Rail - Petroleum	-0.139370	nan	0.042336	1.000000	-0.766916	0.736723	-0.381591	-0.862896	
Road - Electricity	-0.043373	nan	0.610805	-0.766916	1.000000	-0.699193	0.560560	0.852185	
Road - Petroleum	-0.272782	nan	0.191989	0.736723	-0.699193	1.000000	-0.760037	-0.842277	
Road - Bioenergy & waste	0.049269	nan	0.757009	-0.381591	0.560560	-0.760037	1.000000	0.319743	
Road - Natural gas	-0.784103	nan	-0.210177	-0.862896	0.852185	-0.842277	0.319743	1.000000	
Water - Coal	nan	nan	nan	nan	nan	nan	nan	nan	
Water - Petroleum	0.072242	nan	-0.172402	0.244841	-0.478029	0.703650	-0.842168	-0.867337	
Air - Petroleum	0.103173	nan	-0.136244	0.771419	-0.858356	0.801995	-0.520632	-0.913953	

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

SERVICES SECTOR

In [45]:
```
services_df.tail()
```

Out[45]:

	Year	Coal	Coke and breeze	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum	Total
47	2017	27	0	7937	8353	293	1284	3648	21542
48	2018	26	0	7936	8300	310	1317	3695	21584
49	2019	22	0	8016	8106	291	1309	3709	21452
50	2020	21	0	7409	7170	289	1353	3459	19701
51	2021	22	0	7946	7373	290	1436	3651	20718

In [46]:
```
services = pd.read_csv(r"data\ser_depth.csv")
services_pa = services[(services['Year']>1999)&(services['Type']=='Public Adminis
services_ag = services[(services['Year']>1999)&(services['Type']=='Agriculture')]
services_co = services[(services['Year']>1999)&(services['Type']=='Commercial & M
services.head()
```

Out[46]:

	Year	Coal	Manufactured fuel	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum products	Total	Type
0	1970	2063	723	506	1020	0	0	4388	8700	Public Administration
1	1971	1736	441	589	1053	0	0	4798	8617	Public Administration
2	1972	1476	401	912	1088	0	0	4912	8789	Public Administration
3	1973	1431	360	1290	1131	0	0	4751	8963	Public Administration
4	1974	1317	393	1360	1045	0	0	4239	8355	Public Administration

In [47]:
```
(int(services_co[services_co['Year']==2021]['Total']) / int(services_df[services_
```

Out[47]: 67.15899218071243

In [48]:
```
(services_co['Electricity'].mean()/services_co['Total'].mean())*100
```

Out[48]: 50.06156558608653

In [49]:
```
(services_co['Natural gas'].mean()/services_co['Total'].mean())*100
```

Out[49]: 37.9486703617915

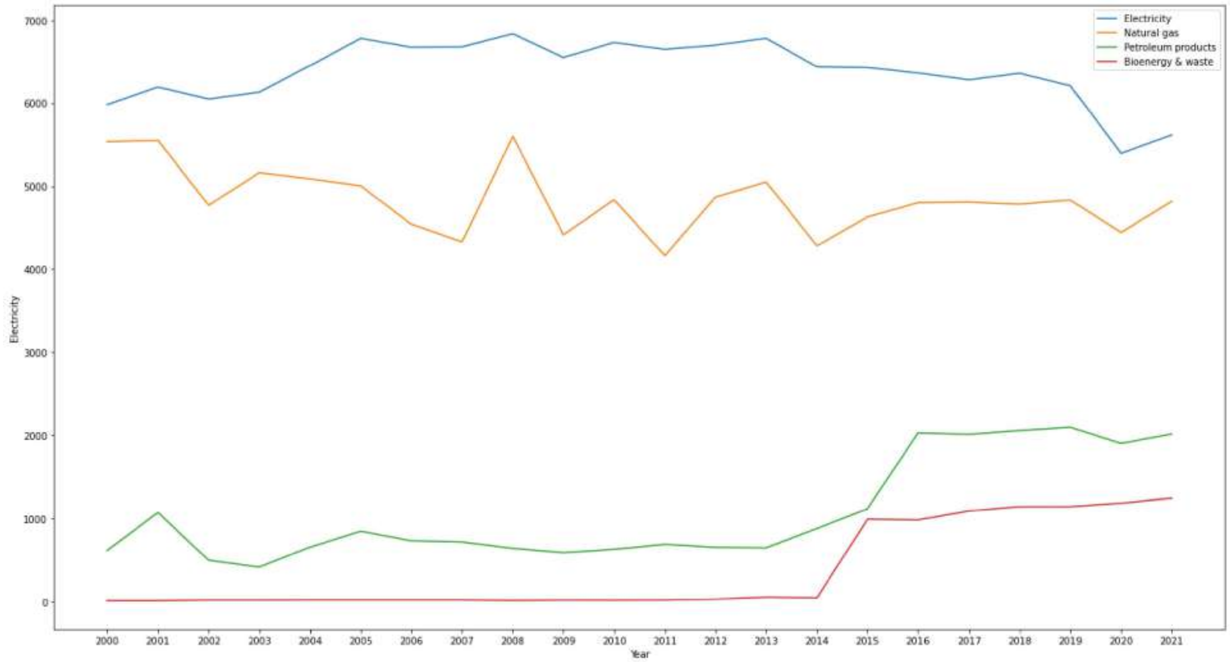
In [50]:
```
(int(services_pa[services_pa['Year']==2021]['Total']) / int(services_df[services_
```

Out[50]: 25.576793126749685

In [51]:
```
(int(services_ag[services_ag['Year']==2021]['Total']) / int(services_df[services_
```

Out[51]: 7.26421469253789

In [52]:
```
plt.figure(figsize=(22,12))
test = sns.lineplot(data=services_co,x='Year',y='Electricity')
sns.lineplot(data=services_co,x='Year',y='Natural gas')
sns.lineplot(data=services_co,x='Year',y='Petroleum products')
sns.lineplot(data=services_co,x='Year',y='Bioenergy & waste')
plt.legend(['Electricity','Natural gas','Petroleum products','Bioenergy & waste']
test.set_xticks(list(services_co['Year']))
plt.show(test)
```



In [53]:
```
ele_com = pd.concat([services_co.iloc[0:]['Year'],services_co['Electricity'].pct_
ele_com
# ele_com.to_csv('data/out.csv',index=False)
```

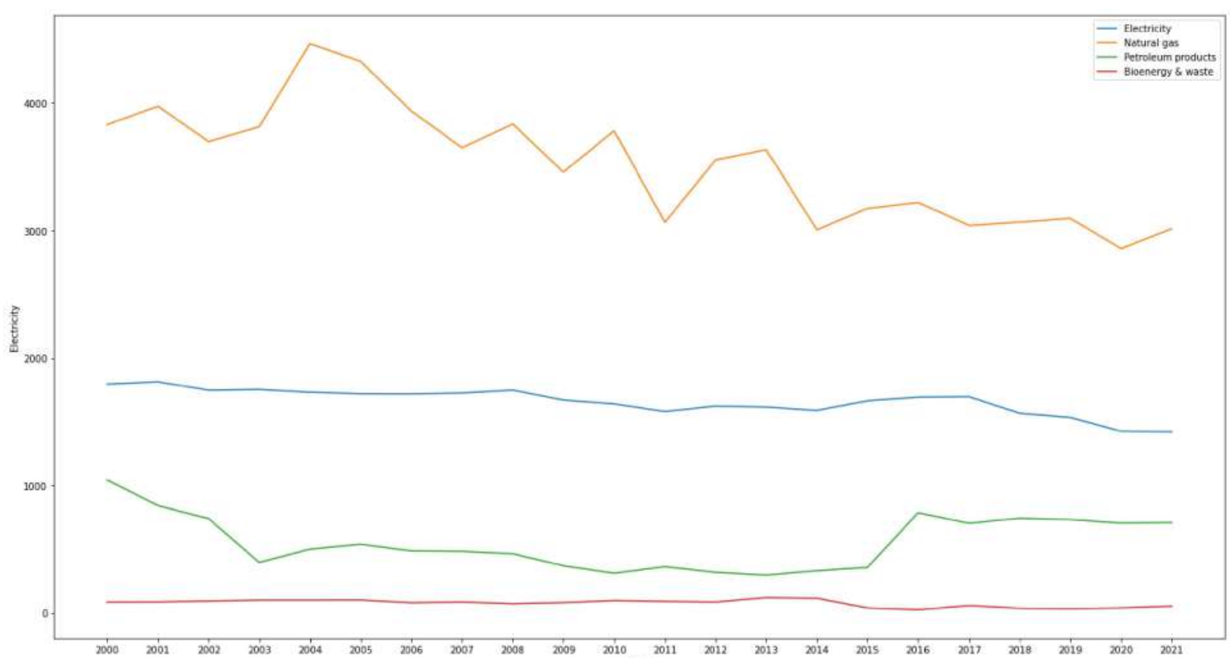
Out[53]:

	Year	Electricity
82	2000	NaN
83	2001	3.510532
84	2002	-2.293282
85	2003	1.338843
86	2004	5.219377
87	2005	5.099984
88	2006	-1.578171
89	2007	0.089915
90	2008	2.335679
91	2009	-4.169715
92	2010	2.748092
93	2011	-1.218425
94	2012	0.752106
95	2013	1.224246
96	2014	-5.073746
97	2015	-0.108763
98	2016	-1.042153
99	2017	-1.273185
100	2018	1.241840
101	2019	-2.358861
102	2020	-13.093896
103	2021	4.077094

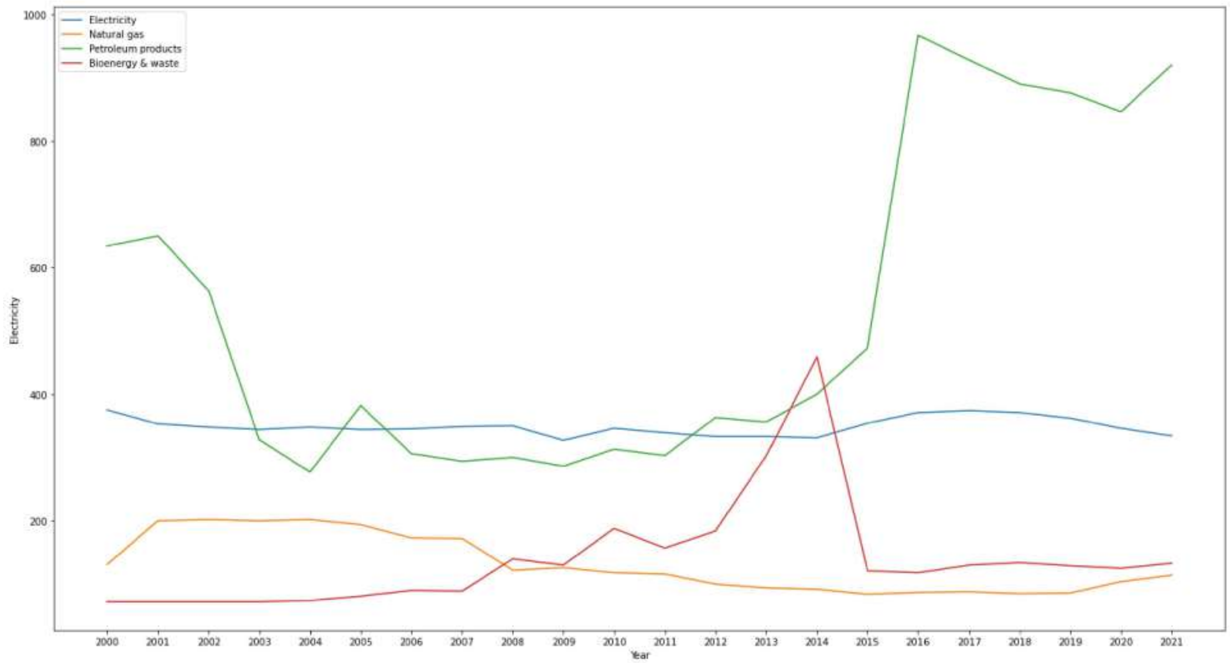
In [54]:
```
services_co['Natural gas'].pct_change()*100
```

Out[54]:
```
82           NaN
83      0.252890
84    -14.036036
85      8.216307
86     -1.433275
87     -1.650619
88     -9.230769
89     -4.644508
90     29.270545
91    -21.160714
92      9.558324
93    -13.934257
94     16.982945
95      3.716632
96    -15.224708
97      8.150397
98      3.692507
99      0.124948
100    -0.499168
101     1.045151
102    -8.067853
103     8.415842
Name: Natural gas, dtype: float64
```

In [55]:
```
plt.figure(figsize=(22,12))
test2 = sns.lineplot(data=services_pa,x='Year',y='Electricity')
sns.lineplot(data=services_pa,x='Year',y='Natural gas')
sns.lineplot(data=services_pa,x='Year',y='Petroleum products')
sns.lineplot(data=services_pa,x='Year',y='Bioenergy & waste')
plt.legend(['Electricity','Natural gas','Petroleum products','Bioenergy & waste']
test2.set_xticks(list(services_pa['Year']))
plt.show(test)
```



In [56]:
```
plt.figure(figsize=(22,12))
test2 = sns.lineplot(data=services_ag,x='Year',y='Electricity')
sns.lineplot(data=services_ag,x='Year',y='Natural gas')
sns.lineplot(data=services_ag,x='Year',y='Petroleum products')
sns.lineplot(data=services_ag,x='Year',y='Bioenergy & waste')
plt.legend(['Electricity','Natural gas','Petroleum products','Bioenergy & waste']
test2.set_xticks(list(services_ag['Year']))
plt.show(test)
```



In [57]:
```
gdp_ser_comm = pd.merge(gdp,services_co[services_co['Year']>1999], how='outer', c
gdp_ser_comm
```

Out[57]:

	Year	gdp	gdp_roc	unemp_rate	pb	com_turnover	Coal	Manufactured fuel	Natural gas	Elect
0	2000	1632591	3.672369	5.4	3.47	926807.7	10	0	5536	
1	2001	1666429	2.072656	5.1	3.50	967114.1	10	0	5550	
2	2002	1701811	2.123223	5.2	3.57	993562.0	5	0	4771	
3	2003	1753374	3.029890	5.0	3.68	1034772.8	5	0	5163	
4	2004	1794677	2.355630	4.8	3.93	1095231.9	5	0	5089	
5	2005	1841218	2.593280	4.8	3.93	1194005.5	6	0	5005	
6	2006	1888797	2.584105	5.4	4.13	1297630.1	7	0	4543	
7	2007	1931663	2.269487	5.3	4.27	1410306.2	6	0	4332	
8	2008	1927034	-0.239638	5.7	4.28	1563152.1	8	0	5600	
9	2009	1845186	-4.247356	7.6	4.38	1436292.0	36	0	4415	
10	2010	1884515	2.131438	7.9	4.48	1588216.8	7	0	4837	
11	2011	1911983	1.457563	8.1	4.59	1707264.1	9	0	4163	
12	2012	1940087	1.469888	8.0	4.82	1769068.8	8	0	4870	
13	2013	1976755	1.890018	7.6	4.91	1918346.8	8	0	5051	
14	2014	2035883	2.991165	6.2	5.25	1957614.8	8	0	4282	
15	2015	2089276	2.622597	5.4	5.40	1900951.1	8	0	4631	
16	2016	2136566	2.263464	4.9	5.50	1966521.2	8	0	4802	
17	2017	2182170	2.134453	4.4	5.69	2052687.8	8	0	4808	
18	2018	2218196	1.650925	4.1	5.67	2197982.2	8	0	4784	
19	2019	2255283	1.671944	3.8	5.87	2274838.5	8	0	4834	
20	2020	2046209	-9.270411	4.6	5.98	1955788.8	8	0	4444	
21	2021	2198473	7.441273	4.5	5.59	2297596.2	8	0	4818	

In [58]:

```
ele_com
gdp_ser_comm2 = pd.merge(gdp,services_pa, how='outer', on = 'Year', suffixes = ('
gdp_ser_comm2.head()
```

Out[58]:

	Year	gdp	gdp_roc	unemp_rate	pb	com_turnover	Coal	Manufactured fuel	Natural gas	Electri
0	2000	1632591	3.672369	5.4	3.47	926807.7	42	0	3831	1
1	2001	1666429	2.072656	5.1	3.50	967114.1	34	0	3975	1
2	2002	1701811	2.123223	5.2	3.57	993562.0	5	0	3697	1
3	2003	1753374	3.029890	5.0	3.68	1034772.8	8	0	3814	1
4	2004	1794677	2.355630	4.8	3.93	1095231.9	9	0	4466	1

In [59]:
```python
corr  = gdp_ser_comm2[gdp_ser_comm2['Year']>1999][gdp_ser_comm2.columns[1:13]].co
corr.style.background_gradient(cmap='coolwarm')
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[59]:

	gdp	gdp_roc	unemp_rate	pb	com_turnover	Coal	Manufactured fuel
gdp	1.000000	0.007077	-0.290962	0.950267	0.959277	-0.260884	nan
gdp_roc	0.007077	1.000000	-0.119833	-0.199145	-0.046660	0.145521	nan
unemp_rate	-0.290962	-0.119833	1.000000	-0.202868	-0.083832	-0.021950	nan
pb	0.950267	-0.199145	-0.202868	1.000000	0.963288	-0.203151	nan
com_turnover	0.959277	-0.046660	-0.083832	0.963288	1.000000	-0.214803	nan
Coal	-0.260884	0.145521	-0.021950	-0.203151	-0.214803	1.000000	nan
Manufactured fuel	nan	nan	nan	nan	nan	nan	nan
Natural gas	-0.745390	0.203585	0.064040	-0.836801	-0.816816	0.119569	nan
Electricity	-0.704893	0.231859	0.009532	-0.795951	-0.803310	0.265155	nan
Heat	-0.853808	0.167429	0.150627	-0.807632	-0.784160	0.532358	nan
Bioenergy & waste	-0.674733	0.194368	0.538091	-0.673028	-0.578433	0.021874	nan
Petroleum products	-0.007050	0.068860	-0.675608	0.004421	-0.081984	0.477251	nan

In [60]:
```
com_g = pd.read_csv(r"data\com_growth.csv")
ser_growth_comm = pd.merge(com_g,services_co[services_co['Year'].between(2008, 20
ser_growth_comm
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_1120\2528881691.py:2: FutureWarning:
Boolean inputs to the `inclusive` argument are deprecated in favour of `both` o
r `neither`.
  ser_growth_comm = pd.merge(com_g,services_co[services_co['Year'].between(200
8, 2018, inclusive = True)], how='outer', on = 'Year', suffixes = ('_left', '_r
ight'))
```

Out[60]:

	Year	growth	Coal	Manufactured fuel	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum products	Total
0	2008	98.550000	8	0	5600	6835	6	14	643	13107
1	2009	95.833333	36	0	4415	6550	9	18	592	11621
2	2010	96.950000	7	0	4837	6730	10	17	631	12233
3	2011	98.541667	9	0	4163	6648	3	19	691	11533
4	2012	101.141667	8	0	4870	6698	7	30	656	12269
5	2013	103.016667	8	0	5051	6780	12	54	650	12555
6	2014	106.308333	8	0	4282	6436	6	47	882	11661
7	2015	109.150000	8	0	4631	6429	220	994	1119	13401
8	2016	111.266667	8	0	4802	6362	241	986	2030	14429
9	2017	113.608333	8	0	4808	6281	215	1093	2013	14418
10	2018	115.508333	8	0	4784	6359	221	1144	2058	14575

In [61]:
```
corr  = ser_growth_comm.corr()
corr.style.background_gradient(cmap='coolwarm')
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[61]:

	Year	growth	Coal	Manufactured fuel	Natural gas	Electricity	Heat	Bioe &
Year	1.000000	0.968819	-0.395874	nan	-0.174463	-0.824998	0.832755	0.8
growth	0.968819	1.000000	-0.408433	nan	-0.011223	-0.821735	0.882508	0.9
Coal	-0.395874	-0.408433	1.000000	nan	-0.303271	-0.014130	-0.234700	-0.2
Manufactured fuel	nan	nan	nan	nan	nan	nan	nan	
Natural gas	-0.174463	-0.011223	-0.303271	nan	1.000000	0.429623	0.022808	0.0
Electricity	-0.824998	-0.821735	-0.014130	nan	0.429623	1.000000	-0.807740	-0.8
Heat	0.832755	0.882508	-0.234700	nan	0.022808	-0.807740	1.000000	0.9
Bioenergy & waste	0.860496	0.909571	-0.244456	nan	0.015807	-0.822250	0.992022	1.0
Petroleum products	0.869074	0.914452	-0.261225	nan	0.010550	-0.834659	0.911775	0.9
Total	0.726436	0.832048	-0.365894	nan	0.417085	-0.577906	0.893100	0.8

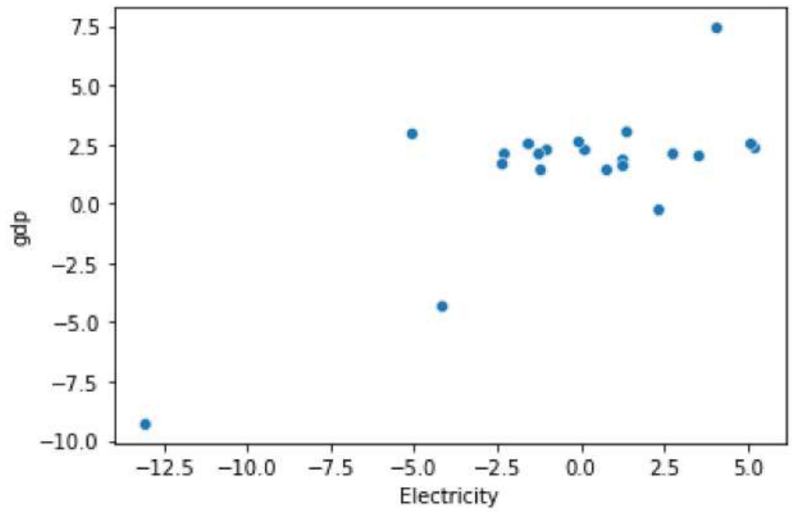
In [62]:
```
roc_df = pd.read_csv(r'data\out.csv')
r_cor = roc_df.corr()
r_cor.style.background_gradient(cmap='coolwarm')
```

Out[62]:

	Year	Electricity	gdp
Year	1.000000	-0.393268	-0.118205
Electricity	-0.393268	1.000000	0.734533
gdp	-0.118205	0.734533	1.000000

In [63]:
```
sns.scatterplot(data=roc_df,x='Electricity',y='gdp')
```

Out[63]: <AxesSubplot:xlabel='Electricity', ylabel='gdp'>



In [64]:
```
new = pd.concat([services_co.iloc[0:]['Year'],services_co['Natural gas'].pct_char
new
ser_growth_comm = pd.merge(new,roc_df, how='outer', on = 'Year', suffixes = ('_le
new = ser_growth_comm.corr()
new.style.background_gradient(cmap='coolwarm')
```

Out[64]:

	Year	Natural gas	Electricity	gdp
Year	1.000000	0.093087	-0.393268	-0.118205
Natural gas	0.093087	1.000000	0.493567	0.260908
Electricity	-0.393268	0.493567	1.000000	0.734533
gdp	-0.118205	0.260908	0.734533	1.000000

In [65]:
```
industry_df.head()
```

Out[65]:

	Year	Coal	Coke and breeze\n	Other solid fuels\n	Coke oven gas	Town gas	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum products
0	1970	12681	9655	209	1164	1778.0	1788	6275	NaN	NaN	28397
1	1971	10232	8298	176	1118	1038.0	5194	6313	NaN	NaN	28130
2	1972	7675	7832	252	1111	1154.0	8136	6292	NaN	NaN	28674
3	1973	7950	8340	226	1290	788.0	10791	6884	NaN	NaN	28691
4	1974	7290	7167	201	975	494.0	12320	6517	NaN	NaN	24968

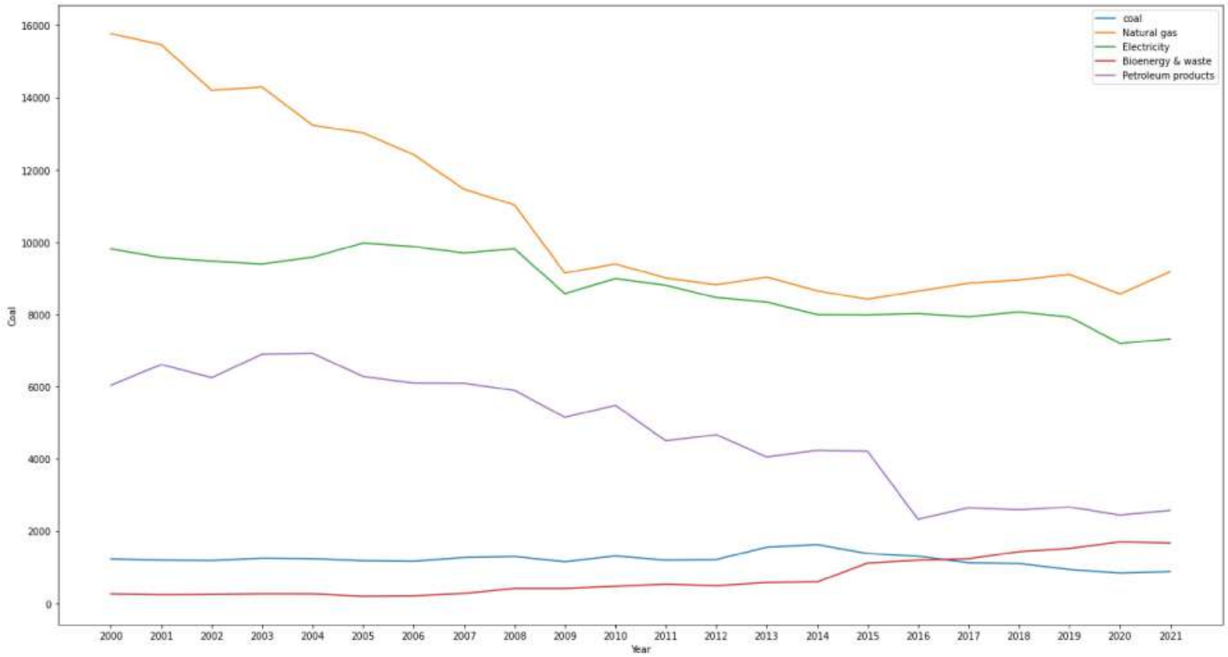
INDUSTRIAL SECTOR

In [66]:
```
industry_df[industry_df['Year']>1999]
```

Out[66]:

	Year	Coal	Coke and breeze\n	Other solid fuels\n	Coke oven gas	Town gas	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum products
30	2000	1228	753	225	216	NaN	15773	9812	1099.0	264.0	6039
31	2001	1195	719	210	154	NaN	15464	9573	1001.0	243.0	6611
32	2002	1186	610	170	78	NaN	14202	9473	1321.0	250.0	6248
33	2003	1248	589	166	53	NaN	14292	9396	1128.0	267.0	6899
34	2004	1235	559	180	67	NaN	13238	9584	832.0	265.0	6918
35	2005	1180	535	171	79	NaN	13022	9976	831.0	201.0	6282
36	2006	1164	488	178	106	NaN	12428	9879	809.0	213.0	6099
37	2007	1268	513	177	101	NaN	11466	9699	896.0	276.0	6095
38	2008	1296	443	174	92	NaN	11030	9815	1021.0	414.0	5895
39	2009	1152	387	20	49	NaN	9146	8576	763.0	415.0	5152
40	2010	1311	339	17	97	NaN	9395	8989	822.0	472.0	5482
41	2011	1194	306	17	59	NaN	9007	8806	769.0	531.0	4500
42	2012	1212	375	17	43	NaN	8821	8466	758.0	488.0	4669
43	2013	1555	504	15	62	NaN	9030	8339	739.0	577.0	4056
44	2014	1627	483	14	55	NaN	8653	7997	627.0	594.0	4238
45	2015	1380	395	0	88	NaN	8418	7991	678.0	1115.0	4212
46	2016	1304	225	0	67	0.0	8647	8024	667.0	1194.0	2331
47	2017	1125	210	0	65	0.0	8862	7937	709.0	1233.0	2645
48	2018	1105	193	0	63	0.0	8950	8071	705.0	1433.0	2589
49	2019	941	198	0	69	0.0	9105	7928	689.0	1522.0	2667
50	2020	841	208	0	86	0.0	8565	7195	700.0	1703.0	2443
51	2021	881	222	0	98	0.0	9181	7318	706.0	1672.0	2572

In [67]:
```
plt.figure(figsize=(22,12))
a = sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y='Coal')
sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y='Natural gas')
sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y='Electricity')
sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y='Bioenergy & w
sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y='Petroleum pro
plt.legend(['coal','Natural gas','Electricity','Bioenergy & waste','Petroleum pro
a.set_xticks(list(industry_df[industry_df['Year']>1999]['Year']))
plt.show(a)
```



In [68]:
```
new = industry_df[industry_df['Year']>1999][industry_df.columns[1:11]].corr()
new.style.background_gradient(cmap='coolwarm')
```

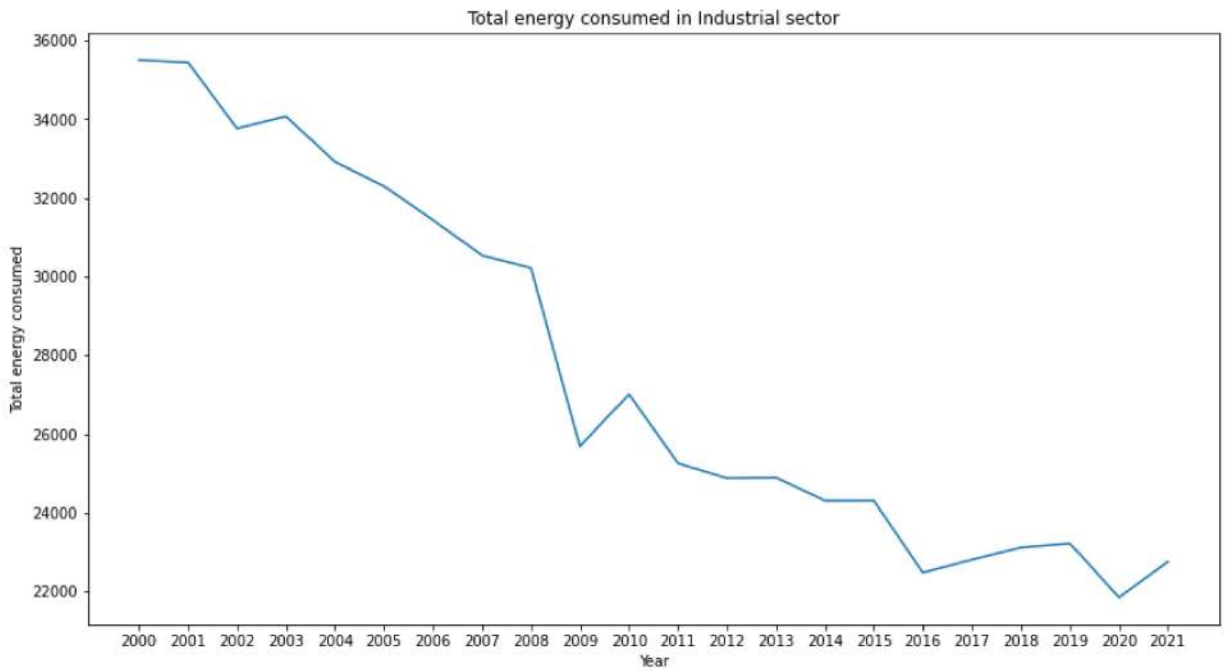
```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[68]:

	Coal	Coke and breeze	Other solid fuels	Coke oven gas	Town gas	Natural gas	Electricity	Heat	Bioer & v
Coal	1.000000	0.429472	0.088278	-0.093532	nan	0.009031	0.259032	0.029122	-0.51
Coke and breeze	0.429472	1.000000	0.845482	0.522612	nan	0.858508	0.780787	0.717962	-0.83
Other solid fuels	0.088278	0.845482	1.000000	0.558536	nan	0.937493	0.904233	0.782213	-0.73
Coke oven gas	-0.093532	0.522612	0.558536	1.000000	nan	0.602511	0.392611	0.411648	-0.20
Town gas	nan	nan	nan	nan	nan	nan	nan	nan	
Natural gas	0.009031	0.858508	0.937493	0.602511	nan	1.000000	0.802882	0.830046	-0.66
Electricity	0.259032	0.780787	0.904233	0.392611	nan	0.802882	1.000000	0.706998	-0.89
Heat	0.029122	0.717962	0.782213	0.411648	nan	0.830046	0.706998	1.000000	-0.61
Bioenergy & waste	-0.514681	-0.831401	-0.739478	-0.204693	nan	-0.662211	-0.890012	-0.611817	1.00
Petroleum products	0.319585	0.864367	0.855793	0.304014	nan	0.794954	0.915133	0.716223	-0.92

In [69]:
```
plt.figure(figsize=(14,7.5))
a = sns.lineplot(data=industry_df[industry_df['Year']>1999],x='Year',y='Total')
a.set_xticks(list(industry_df[industry_df['Year']>1999]['Year']))
a.set_ylabel('Total energy consumed')
a.set_title('Total energy consumed in Industrial sector')
```

Out[69]: Text(0.5, 1.0, 'Total energy consumed in Industrial sector')



In [70]:
```
ems = pd.read_csv(r'data\em_uk.csv')
emission_industry = pd.merge(ems,industry_df[industry_df['Year']>1999], how='oute

new = emission_industry[emission_industry.columns[1:13]].corr()
new.style.background_gradient(cmap='coolwarm')
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3554: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
\io\formats\style.py:3555: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

Out[70]:

	Annual emissions	Coal	Coke and breeze	Other solid fuels	Coke oven gas	Town gas	Natural gas	Electricity	
Annual emissions	1.000000	0.338575	0.885756	0.892093	0.365918	nan	0.838097	0.959460	0.74
Coal	0.338575	1.000000	0.429472	0.088278	-0.093532	nan	0.009031	0.259032	0.02
Coke and breeze	0.885756	0.429472	1.000000	0.845482	0.522612	nan	0.858508	0.780787	0.71
Other solid fuels	0.892093	0.088278	0.845482	1.000000	0.558536	nan	0.937493	0.904233	0.78
Coke oven gas	0.365918	-0.093532	0.522612	0.558536	1.000000	nan	0.602511	0.392611	0.4
Town gas	nan	nan	nan	nan	nan	nan	nan	nan	
Natural gas	0.838097	0.009031	0.858508	0.937493	0.602511	nan	1.000000	0.802882	0.83
Electricity	0.959460	0.259032	0.780787	0.904233	0.392611	nan	0.802882	1.000000	0.70
Heat	0.747972	0.029122	0.717962	0.782213	0.411648	nan	0.830046	0.706998	1.00
Bioenergy & waste	-0.950436	-0.514681	-0.831401	-0.739478	-0.204693	nan	-0.662211	-0.890012	-0.6
Petroleum products	0.974716	0.319585	0.864367	0.855793	0.304014	nan	0.794954	0.915133	0.71
Total	0.943182	0.155916	0.904467	0.963671	0.527733	nan	0.965916	0.909908	0.83

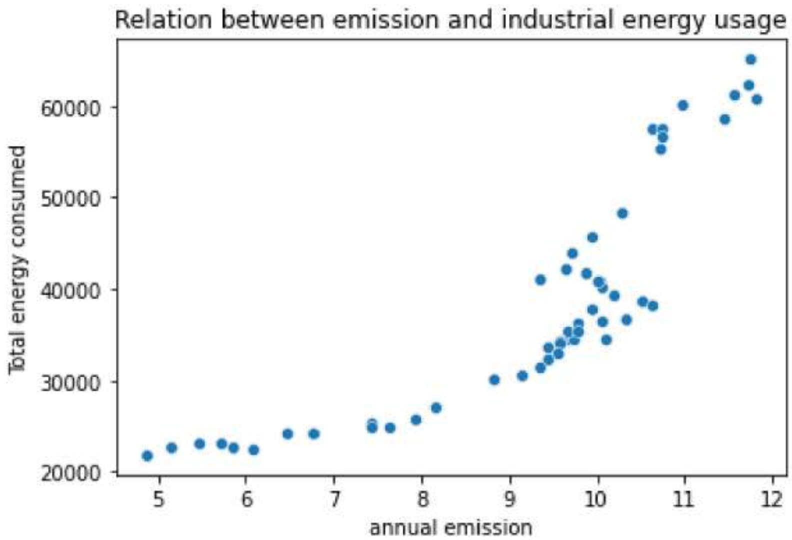
In [71]:
```
ems = pd.read_csv(r'data\1970-2021 emission.csv')
emission_industry = pd.merge(ems,industry_df, how='outer', on = 'Year')
new = emission_industry[emission_industry.columns[1:13]].corr()
new.style.background_gradient(cmap='coolwarm')
```

Out[71]:

	annual emission	Coal	Coke and breeze	Other solid fuels	Coke oven gas	Town gas	Natural gas	Electricity
annual emission	1.000000	0.716282	0.750939	0.669941	0.775259	0.489646	0.377363	-0.195582
Coal	0.716282	1.000000	0.962216	0.333466	0.924109	0.800003	-0.162189	-0.729120
Coke and breeze	0.750939	0.962216	1.000000	0.340531	0.972832	0.667140	-0.006801	-0.721006
Other solid fuels	0.669941	0.333466	0.340531	1.000000	0.407461	0.604084	0.401231	0.052550
Coke oven gas	0.775259	0.924109	0.972832	0.407461	1.000000	0.560037	0.115047	-0.691237
Town gas	0.489646	0.800003	0.667140	0.604084	0.560037	1.000000	-0.697688	-0.518031
Natural gas	0.377363	-0.162189	-0.006801	0.401231	0.115047	-0.697688	1.000000	0.204343
Electricity	-0.195582	-0.729120	-0.721006	0.052550	-0.691237	-0.518031	0.204343	1.000000
Heat	0.763339	0.075686	0.744918	0.801681	0.489856	nan	0.845886	0.718291
Bioenergy & waste	-0.911951	-0.467204	-0.461662	-0.463900	-0.422095	nan	-0.641754	-0.701049
Petroleum products	0.739641	0.921724	0.921191	0.494875	0.897457	0.683137	-0.035906	-0.667108
Total	0.846089	0.907408	0.939625	0.568719	0.939861	0.571901	0.169034	-0.598532

In [72]:
```
a = sns.scatterplot(data = emission_industry,x='annual emission',y='Total')
a.set_title('Relation between emission and industrial energy usage')
a.set_ylabel('Total energy consumed')
```

Out[72]:
```
Text(0, 0.5, 'Total energy consumed')
```



In [73]:
```
emission_industry[['Total','annual emission']].corr()
```

Out[73]:

	Total	annual emission
Total	1.000000	0.846089
annual emission	0.846089	1.000000

In [ ]:

Transportation

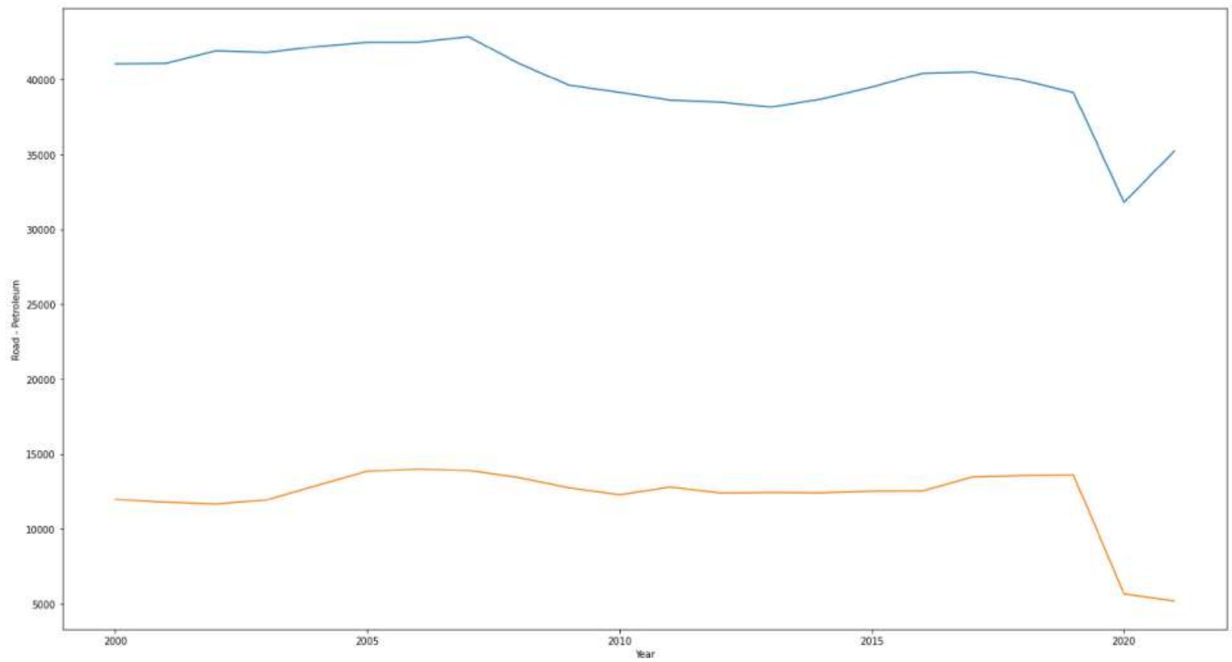
In [38]:
```
transport_df[transport_df['Year']>1999]
```

Out[38]:

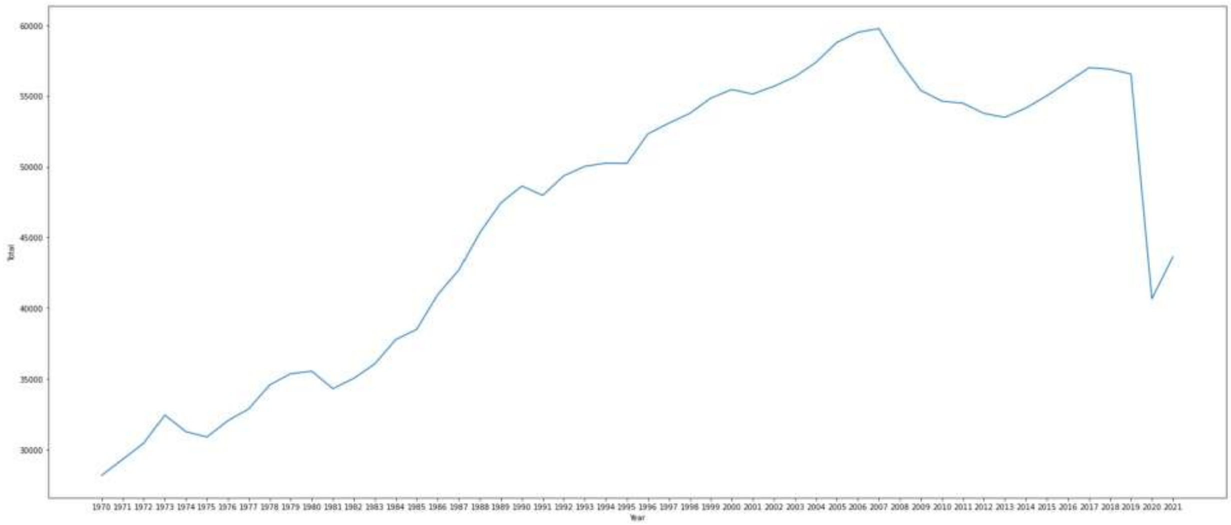
	Year	Rail - Coal	Rail - Coke and breeze	Rail - Electricity	Rail - Petroleum	Road - Electricity	Road - Petroleum	Road - Bioenergy & waste	Road - Natural gas	Water - Coal	P
30	2000	0.0	NaN	741	639	NaN	41071	NaN	NaN	NaN	
31	2001	0.0	NaN	759	664	NaN	41097	NaN	NaN	NaN	
32	2002	0.0	NaN	727	662	NaN	41936	NaN	NaN	NaN	
33	2003	0.0	NaN	706	667	NaN	41823	NaN	NaN	NaN	
34	2004	0.0	NaN	347	700	2.0	42221	NaN	NaN	NaN	
35	2005	3.0	NaN	347	634	2.0	42507	74.0	NaN	NaN	
36	2006	14.0	NaN	342	632	2.0	42513	188.0	NaN	NaN	
37	2007	14.0	NaN	339	646	2.0	42884	362.0	NaN	NaN	
38	2008	14.0	NaN	338	658	2.0	41098	845.0	NaN	NaN	
39	2009	13.0	NaN	347	656	2.0	39635	1038.0	NaN	NaN	
40	2010	14.0	NaN	348	660	2.0	39159	1218.0	NaN	NaN	
41	2011	11.0	NaN	360	651	2.0	38646	1128.0	NaN	NaN	
42	2012	12.0	NaN	383	673	2.0	38508	958.0	NaN	NaN	
43	2013	10.0	NaN	371	667	3.0	38177	1092.0	NaN	NaN	
44	2014	9.0	NaN	381	676	6.0	38713	1243.0	NaN	NaN	
45	2015	9.0	NaN	380	674	8.0	39510	998.0	NaN	NaN	
46	2016	11.0	NaN	392	666	11.0	40429	1010.0	NaN	NaN	
47	2017	11.0	NaN	400	661	16.0	40522	997.0	NaN	NaN	
48	2018	11.0	NaN	408	658	21.0	39959	1365.0	10.0	NaN	
49	2019	11.0	NaN	455	601	23.0	39146	1736.0	43.0	NaN	
50	2020	9.0	NaN	416	472	44.0	31792	1639.0	76.0	NaN	
51	2021	10.0	NaN	401	552	74.0	35229	1463.0	84.0	NaN	

In [39]:
```
plt.figure(figsize=(22,12))
a = sns.lineplot(data=transport_df[transport_df['Year']>1999],x='Year',y='Road -
sns.lineplot(data=transport_df[transport_df['Year']>1999],x='Year',y='Air - Petro
```

Out[39]: <AxesSubplot:xlabel='Year', ylabel='Road - Petroleum'>



In [40]:
```
plt.figure(figsize=(28,12))
a = sns.lineplot(data=transport_df,x='Year',y='Total')
a.set_xticks(list(transport_df['Year']))
plt.show(a)
```



In [41]:
```
transport_df[(transport_df['Year']>1999) & (transport_df['Year']<2019)].describe(
```

Out[41]:

	Year	Rail - Coal	Rail - Coke and breeze	Rail - Electricity	Rail - Petroleum	Road - Electricity	Road - Petroleum	Roa Bioenergy wa
count	19.000000	19.000000	0.0	19.000000	19.000000	15.000000	19.000000	14.000(
mean	2009.000000	8.210526	NaN	442.947368	660.210526	5.533333	40547.789474	894.000(
std	5.627314	5.633027	NaN	155.699309	15.946365	5.998412	1515.741314	397.932
min	2000.000000	0.000000	NaN	338.000000	632.000000	2.000000	38177.000000	74.000(
25%	2004.500000	1.500000	NaN	347.000000	653.500000	2.000000	39334.500000	873.250(
50%	2009.000000	11.000000	NaN	380.000000	661.000000	2.000000	40522.000000	1004.000(
75%	2013.500000	12.500000	NaN	404.000000	667.000000	7.000000	41879.500000	1119.000(
max	2018.000000	14.000000	NaN	759.000000	700.000000	21.000000	42884.000000	1365.000(

In [42]:
```
(int(transport_df[transport_df['Year']==2021]['Road - Petroleum']) / int(transpor
```

Out[42]: 80.75415472779369

In [43]:
```
(int(transport_df[transport_df['Year']==2021]['Air - Petroleum']) / int(transport
```

Out[43]: 11.864756446991404

In [44]:
```
new = transport_df[transport_df.columns[1:5]].corr()
new.style.background_gradient(cmap='coolwarm')
```

Out[44]:

	Rail - Coal	Rail - Coke and breeze	Rail - Electricity	Rail - Petroleum
Rail - Coal	1.000000	0.954457	-0.575131	0.893774
Rail - Coke and breeze	0.954457	1.000000	-0.564218	0.840441
Rail - Electricity	-0.575131	-0.564218	1.000000	-0.614604
Rail - Petroleum	0.893774	0.840441	-0.614604	1.000000

In [45]:
```
new = transport_df[transport_df['Year']>1999][transport_df.columns[5:9]].corr()
new.style.background_gradient(cmap='coolwarm')
```

Out[45]:

	Road - Electricity	Road - Petroleum	Road - Bioenergy & waste	Road - Natural gas
Road - Electricity	1.000000	-0.699193	0.560560	0.852185
Road - Petroleum	-0.699193	1.000000	-0.760037	-0.842277
Road - Bioenergy & waste	0.560560	-0.760037	1.000000	0.319743
Road - Natural gas	0.852185	-0.842277	0.319743	1.000000

In [59]:
```
road_veh = pd.read_csv(r'data\vehicle_reg.csv')
road_veh1 = road_veh[road_veh['geography']=='Great Britain']
road_energy = transport_df[transport_df['Year']>2000][['Year','Road - Petroleum']
road_cor = pd.merge(road_veh1,road_energy, how='outer', on = 'Year', suffixes = (
new = road_cor.corr()
new.style.background_gradient(cmap='coolwarm')
```

Out[59]:

	Year	total_vehicles	Road - Petroleum
Year	1.000000	-0.409075	-0.722913
total_vehicles	-0.409075	1.000000	0.738532
Road - Petroleum	-0.722913	0.738532	1.000000

In [51]:
```
road_cor
```

Out[51]:

	geography	Year	total	Road - Petroleum
0	Great Britain	2010	1095	39159
1	Great Britain	2011	1897	38646
2	Great Britain	2012	3350	38508
3	Great Britain	2013	4137	38177
4	Great Britain	2014	15593	38713
5	Great Britain	2015	30282	39510
6	Great Britain	2016	40144	40429
7	Great Britain	2017	50366	40522
8	Great Britain	2018	63100	39959
9	Great Britain	2019	80170	39146
10	Great Britain	2020	183032	31792
11	Great Britain	2021	322749	35229

In [ ]:

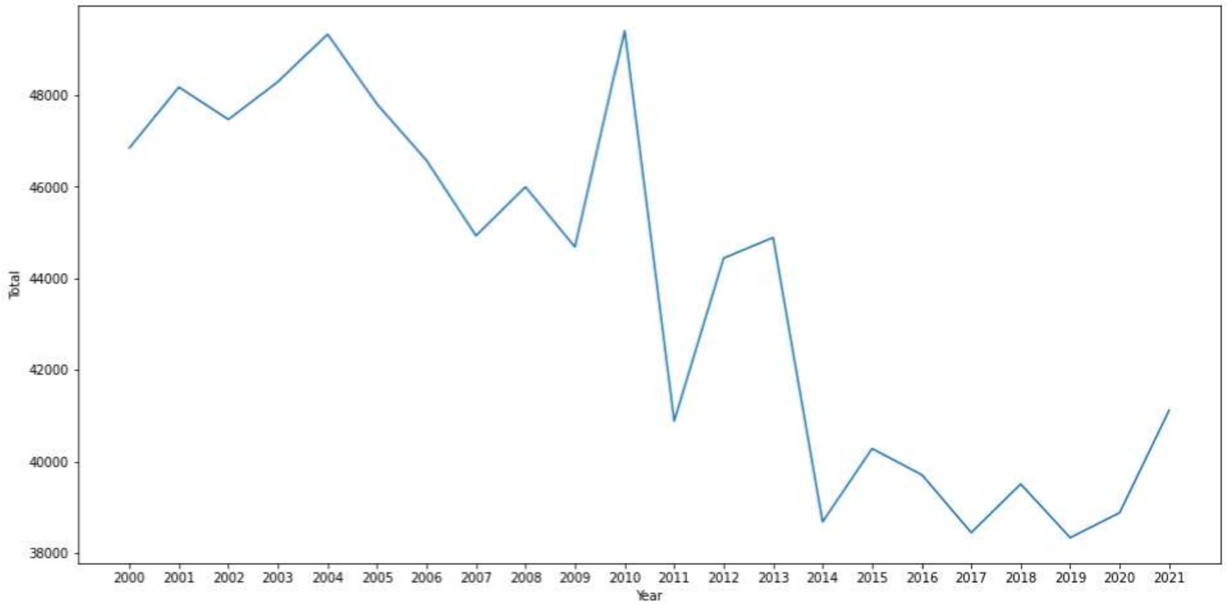
Domestic sector

In [83]:
```
dom = domestic_df[domestic_df['Year']>1999]
dom
```

Out[83]:

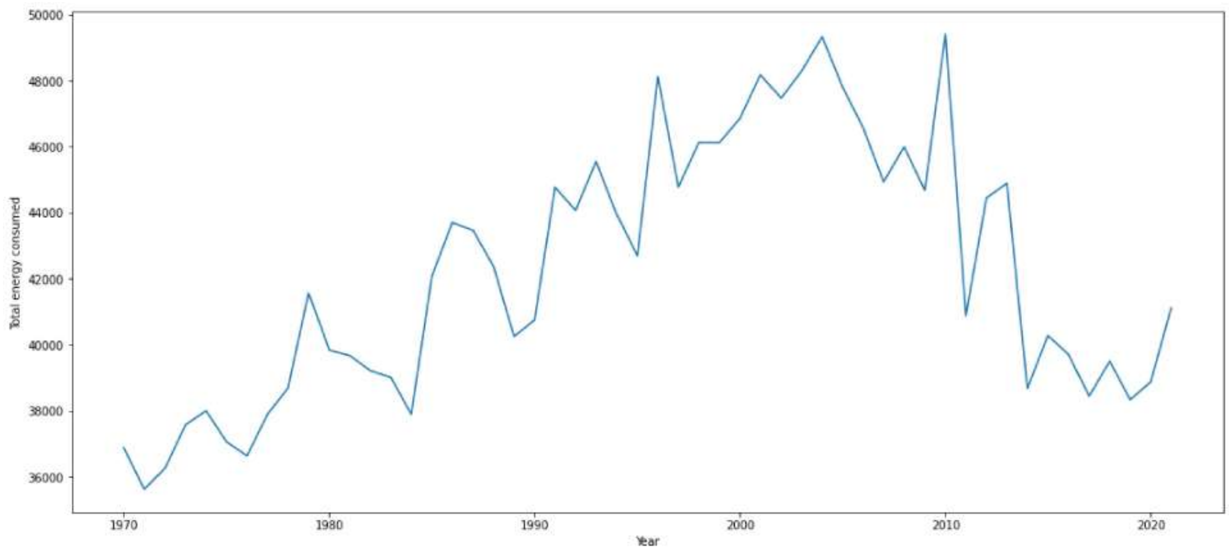
	Year	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum	Total
30	2000	1448	95	365	31806	9617	44.0	236.0	3239	46851
31	2001	1461	48	328	32625	9917	32.0	240.0	3527	48178
32	2002	1009	127	289	32362	10319	33.0	243.0	3087	47471
33	2003	813	92	255	33232	10576	11.0	247.0	3068	48293
34	2004	733	36	230	34085	10679	52.0	252.0	3265	49333
35	2005	474	24	199	32836	10809	52.0	318.0	3094	47805
36	2006	426	16	200	31550	10723	52.0	358.0	3251	46575
37	2007	487	11	182	30341	10583	52.0	400.0	2877	44932
38	2008	515	9	229	30916	10301	52.0	313.0	3033	45998
39	2009	514	7	192	29682	10193	52.0	344.0	3013	44685
40	2010	537	7	221	33499	10218	52.0	465.0	3428	49410
41	2011	530	6	192	26556	9595	52.0	437.0	2669	40883
42	2012	506	5	180	29508	9859	52.0	529.0	2707	44441
43	2013	486	4	216	29622	9752	52.0	621.0	2845	44891
44	2014	415	4	178	24393	9293	52.0	602.0	2508	38680
45	2015	418	2	165	25587	9266	260.0	690.0	2518	40281
46	2016	401	0	168	26301	9288	260.0	752.0	2543	39713
47	2017	371	0	172	25372	9060	269.0	786.0	2416	38446
48	2018	358	0	171	26249	9034	270.0	885.0	2540	39507
49	2019	337	0	142	25255	8918	260.0	946.0	2476	38335
50	2020	321	0	141	25500	9284	260.0	977.0	2397	38879
51	2021	328	0	122	27377	9411	260.0	1127.0	2489	41115

In [84]:
```
plt.figure(figsize=(16,8))
a = sns.lineplot(data=dom,x='Year',y='Total')
a.set_xticks(list(dom['Year']))
plt.show(a)
```



In [85]:
```
plt.figure(figsize=(18,8))
b = sns.lineplot(data=domestic_df,x='Year',y='Total')
b.set_ylabel('Total energy consumed')
```

Out[85]: Text(0, 0.5, 'Total energy consumed')



In [86]:
```
dom.describe()
```

Out[86]:

	Year	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat
count	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000
mean	2010.500000	585.818182	22.409091	206.227273	29302.454545	9849.772727	115.045455
std	6.493587	325.834553	36.121039	59.473829	3221.061060	612.650132	103.738443
min	2000.000000	321.000000	0.000000	122.000000	24393.000000	8918.000000	11.000000
25%	2005.250000	404.500000	0.500000	171.250000	26262.000000	9289.250000	52.000000
50%	2010.500000	486.500000	6.500000	192.000000	29652.000000	9805.500000	52.000000
75%	2015.750000	535.250000	22.000000	227.000000	32223.000000	10314.500000	260.000000
max	2021.000000	1461.000000	127.000000	365.000000	34085.000000	10809.000000	270.000000

In [87]:
```
new = dom[dom.columns[1:9]].corr()
new.style.background_gradient(cmap='coolwarm')
```

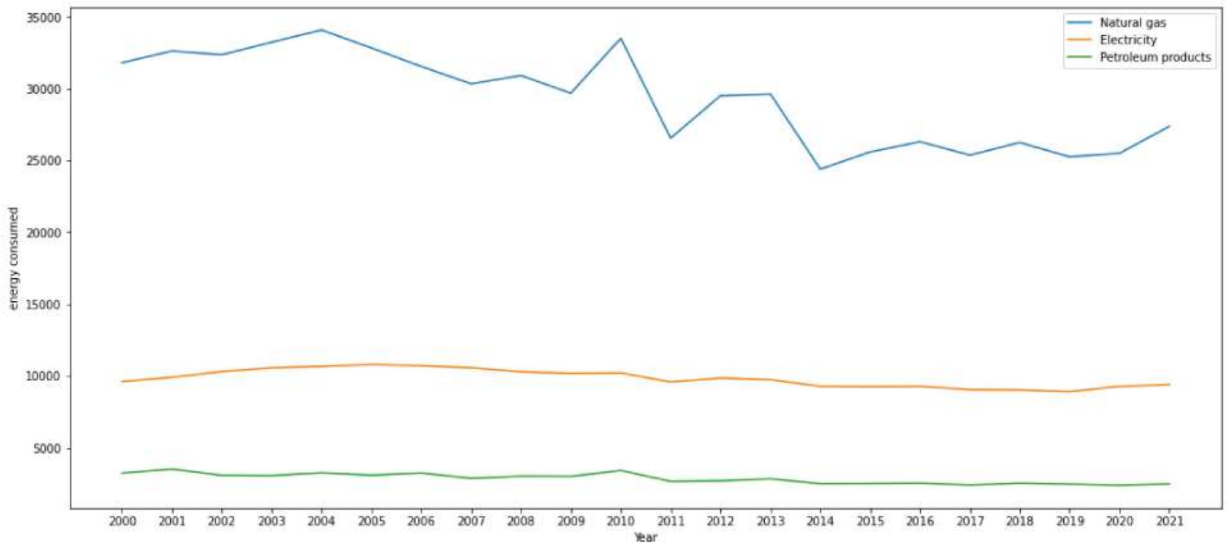
Out[87]:

	Coal	Coke and breeze	Other solid fuels	Natural gas	Electricity	Heat	Bioenergy & waste	Petroleum
Coal	1.000000	0.784505	0.951028	0.586504	0.267250	-0.523851	-0.652176	0.683145
Coke and breeze	0.784505	1.000000	0.799762	0.596353	0.400675	-0.490157	-0.616761	0.534781
Other solid fuels	0.951028	0.799762	1.000000	0.682307	0.390840	-0.643366	-0.770309	0.771051
Natural gas	0.586504	0.596353	0.682307	1.000000	0.869536	-0.748838	-0.798759	0.929135
Electricity	0.267250	0.400675	0.390840	0.869536	1.000000	-0.770547	-0.785557	0.765162
Heat	-0.523851	-0.490157	-0.643366	-0.748838	-0.770547	1.000000	0.887469	-0.765266
Bioenergy & waste	-0.652176	-0.616761	-0.770309	-0.798759	-0.785557	0.887469	1.000000	-0.833612
Petroleum	0.683145	0.534781	0.771051	0.929135	0.765162	-0.765266	-0.833612	1.000000

In [88]:
```
dom['Total'].pct_change ()*100
```

Out[88]:
```
30          NaN
31     2.832384
32    -1.467475
33     1.731583
34     2.153521
35    -3.097318
36    -2.572953
37    -3.527644
38     2.372474
39    -2.854472
40    10.574018
41   -17.257640
42     8.702884
43     1.012578
44   -13.835735
45     4.139090
46    -1.410094
47    -3.190391
48     2.759715
49    -2.966563
50     1.419069
51     5.751177
Name: Total, dtype: float64
```

In [89]:
```
plt.figure(figsize=(18,8))
b = sns.lineplot(data=dom,x='Year',y='Natural gas')
sns.lineplot(data=dom,x='Year',y='Electricity')
sns.lineplot(data=dom,x='Year',y='Petroleum')
b.set_ylabel('energy consumed')
b.set_xticks(list(dom['Year']))
plt.legend(['Natural gas','Electricity','Petroleum products'])
plt.show(b)
```



In [ ]:

In [99]:
```
temp = pd.read_csv('data/temp.csv')
temp
```

Out[99]:

	Year	temp	pop	gas_price
0	2000	9.10	58886100	0.0
1	2001	8.80	59113000	0.0
2	2002	9.44	59365700	0.0
3	2003	9.47	59636700	34.3
4	2004	9.44	59950400	36.7
5	2005	9.42	60413300	41.9
6	2006	9.70	60827100	55.0
7	2007	9.56	61319100	59.3
8	2008	9.02	61823800	70.8
9	2009	9.14	62260500	80.3
10	2010	7.94	62759500	75.8
11	2011	9.61	63285100	84.1
12	2012	8.74	63705000	92.9
13	2013	8.74	64105700	100.0
14	2014	9.88	64596800	104.7
15	2015	9.18	65110000	100.0
16	2016	9.29	65648100	94.1
17	2017	9.53	66040200	93.0
18	2018	9.45	66435600	96.6
19	2019	9.39	66796800	96.1
20	2020	9.62	67081000	86.5
21	2021	9.28	67026300	86.7

In [100]:
```
do = dom[dom['Year']>2002]
te = temp[temp['Year']>2002]
```

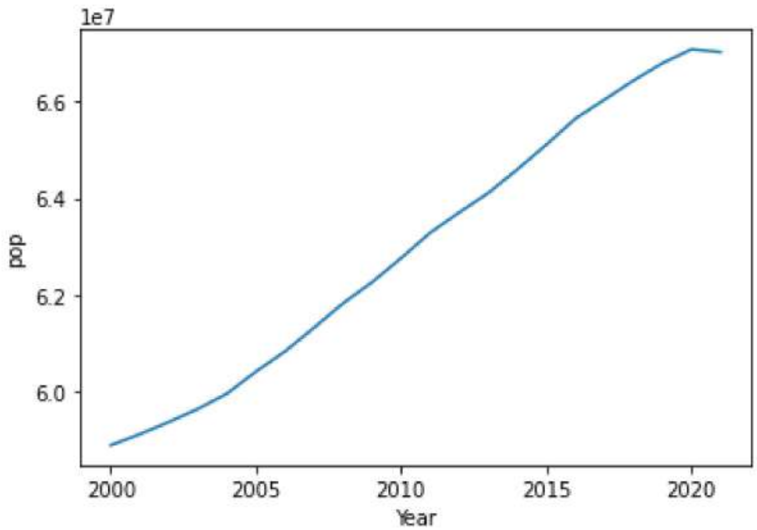
In [103]:
```
temp_nat = pd.merge(gdp,dom, how='outer', on = 'Year', suffixes = ('_left', '_rig
new = temp_nat.corr()
new.style.background_gradient(cmap='coolwarm')
```

Out[103]:

	Year	gdp	gdp_roc	unemp_rate	pb	com_turnover	Coal	C
Year	1.000000	0.947193	-0.163265	-0.144243	0.986536	0.978688	-0.784223	-0
gdp	0.947193	1.000000	0.007077	-0.290962	0.950267	0.959277	-0.811313	-0
gdp_roc	-0.163265	0.007077	1.000000	-0.119833	-0.199145	-0.046660	0.175162	0
unemp_rate	-0.144243	-0.290962	-0.119833	1.000000	-0.202868	-0.083832	-0.002163	-0
pb	0.986536	0.950267	-0.199145	-0.202868	1.000000	0.963288	-0.761016	-0
com_turnover	0.978688	0.959277	-0.046660	-0.083832	0.963288	1.000000	-0.771806	-0
Coal	-0.784223	-0.811313	0.175162	-0.002163	-0.761016	-0.771806	1.000000	0
Coke and breeze	-0.749646	-0.746853	0.193461	-0.146855	-0.735395	-0.768352	0.784505	1
Other solid fuels	-0.862066	-0.863254	0.162201	0.092556	-0.836127	-0.827434	0.951028	0
Natural gas	-0.851856	-0.823420	0.172881	0.155703	-0.886982	-0.853237	0.586504	0
Electricity	-0.763613	-0.707807	0.060896	0.197951	-0.801461	-0.776617	0.267250	0
Heat	0.833891	0.848812	-0.090532	-0.542366	0.862897	0.776997	-0.523851	-0
Bioenergy & waste	0.960523	0.910977	-0.069640	-0.319702	0.946713	0.925959	-0.652176	-0
Petroleum	-0.874441	-0.851192	0.134727	0.220119	-0.892116	-0.857558	0.683145	0
Total	-0.862016	-0.846798	0.168978	0.241922	-0.898550	-0.855367	0.593374	0

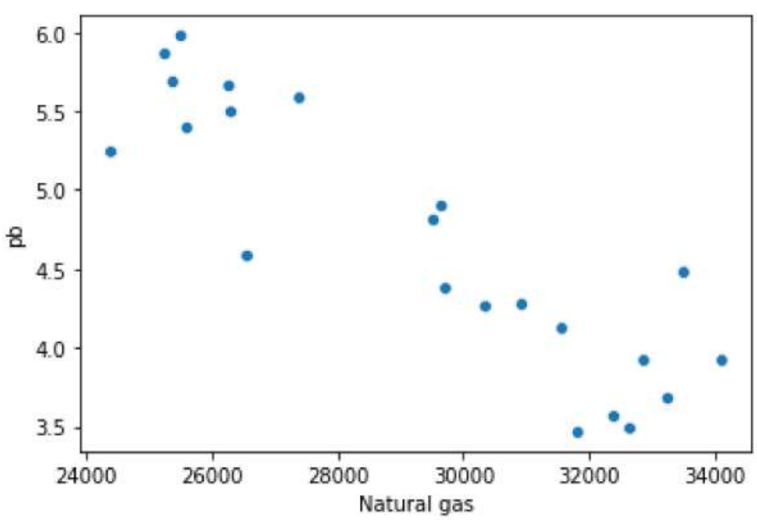
In [98]:
```
sns.lineplot(data=temp,x='Year',y='pop')
```

Out[98]: <AxesSubplot:xlabel='Year', ylabel='pop'>



In [106]:
```
sns.scatterplot(data=temp_nat,x='Natural gas',y='pb')
```

Out[106]: <AxesSubplot:xlabel='Natural gas', ylabel='pb'>



In [ ]:

ARIMA MODEL

In [3]:
```
# industry_df
ind = industry_df[['Year','Total']].set_index('Year')
# domestic_df
dom = domestic_df[['Year','Total']].set_index('Year')
# services_df
ser = services_df[['Year','Total']].set_index('Year')
# transport_df
tra = transport_df[['Year','Total']].set_index('Year')
```

In [25]:
```
#check time series if its stationary or not
from statsmodels.tsa.stattools import adfuller
ind_result = adfuller(ind.Total, maxlag=1)
dom_result = adfuller(dom.Total, maxlag=1)
ser_result = adfuller(ser.Total, maxlag=1)
tra_result = adfuller(tra.Total, maxlag=1)

#indsutry
print('INDUSTRY \n\t','ADF Statistic: %f' % ind_result[0])
print('\t p-value: %f' % ind_result[1],'\n')
print(ind_result[4],'\n')

#domestic
print('DOMESTIC \n\t','ADF Statistic: %f' % dom_result[0])
print('\t p-value: %f' % dom_result[1],'\n')
print(dom_result[4],'\n')

#SERVICE
print('SERVICES \n\t','ADF Statistic: %f' % ser_result[0])
print('\t p-value: %f' % ser_result[1],'\n')
print(ser_result[4],'\n')

#TRANSPORTATION
print('TRANSPORTATION \n\t','ADF Statistic: %f' % tra_result[0])
print('\t p-value: %f' % tra_result[1],'\n')
print(tra_result[4],'\n')
```

```
INDUSTRY
         ADF Statistic: -1.566226
         p-value: 0.500534

{'1%': -3.5656240522121956, '5%': -2.920142229157715, '10%': -2.59801467512495
2}

DOMESTIC
         ADF Statistic: -1.910875
         p-value: 0.326963

{'1%': -3.568485864, '5%': -2.92135992, '10%': -2.5986616}

SERVICES
         ADF Statistic: -2.847493
         p-value: 0.051803

{'1%': -3.5656240522121956, '5%': -2.920142229157715, '10%': -2.59801467512495
2}

TRANSPORTATION
         ADF Statistic: -1.867004
         p-value: 0.347739

{'1%': -3.5656240522121956, '5%': -2.920142229157715, '10%': -2.59801467512495
2}
```

WE ACCEPT ALL HYPOTHESIS at 5% significance, ALL data are currently NON-STATIONARY

In [66]:
```
#testing stationarity after 1st diffrencing
print('\t p-value INDUSTRY: %f' % adfuller(ind.Total.diff().dropna(), maxlag=1)[1
print('\t p-value DOMEATIC: %f' % adfuller(dom.Total.diff().dropna(), maxlag=1)[1
print('\t p-value SERVICES: %f' % adfuller(ser.Total.diff().dropna(), maxlag=1)[1
print('\t p-value TRANSPORT: %f' % adfuller(tra.Total.diff().dropna(), maxlag=1)|
```

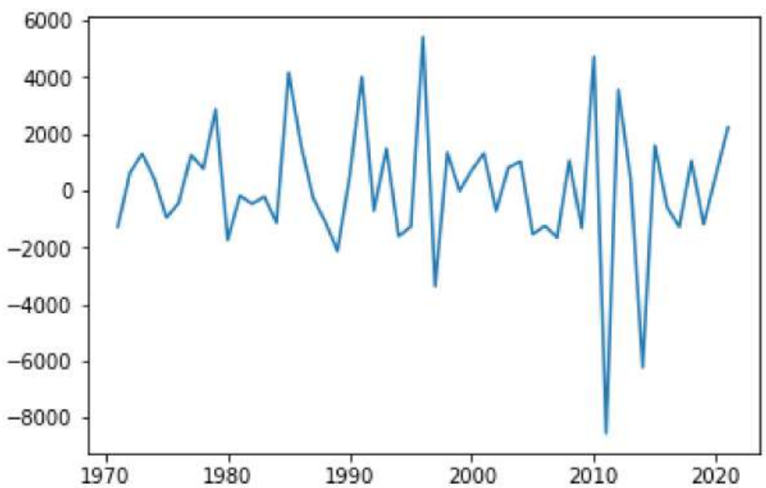
```
	 p-value INDUSTRY: 0.000000

	 p-value DOMEATIC: 0.000000

	 p-value SERVICES: 0.000000

	 p-value TRANSPORT: 0.000000
```

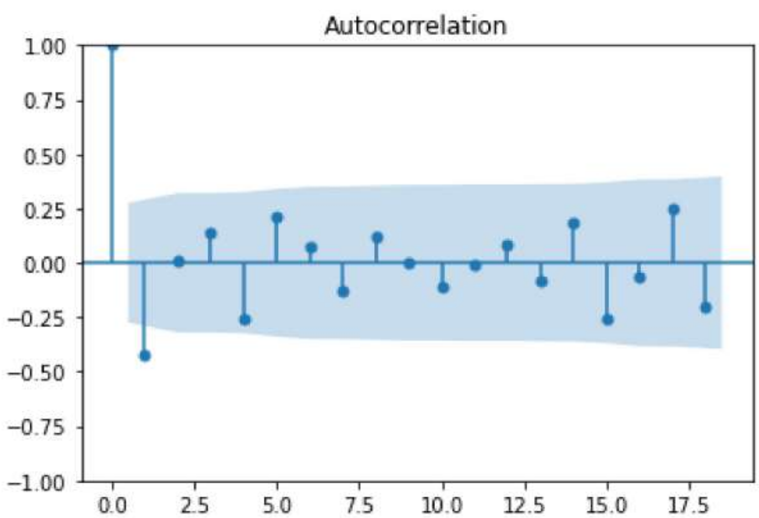
After diffrencing all p values are less than 0.05, implying that they are stationary now

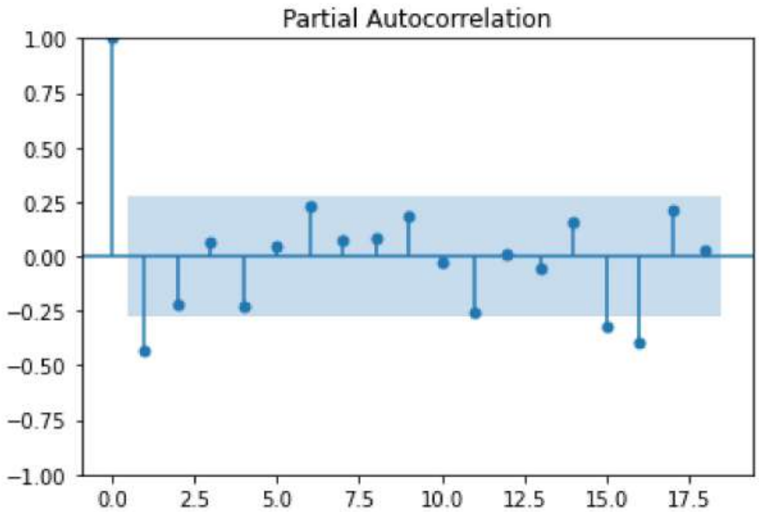
- **ARIMA ON DOMESTIC SECTOR**

In [92]:
```
#plotting to find p and q
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plt.plot(dom.Total.diff().dropna())
plot_acf(dom.Total.diff().dropna()) #to measure moving average

plot_pacf(dom.Total.diff().dropna()) #to measure auto regressive
plt.show()
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\graphics\tsaplots.py:348: FutureWarning: The default method 'yw' can produ
ce PACF values outside of the [-1,1] interval. After 0.13, the default will cha
nge tounadjusted Yule-Walker ('ywm'). You can use this method now by setting me
thod='ywm'.
  warnings.warn(
```





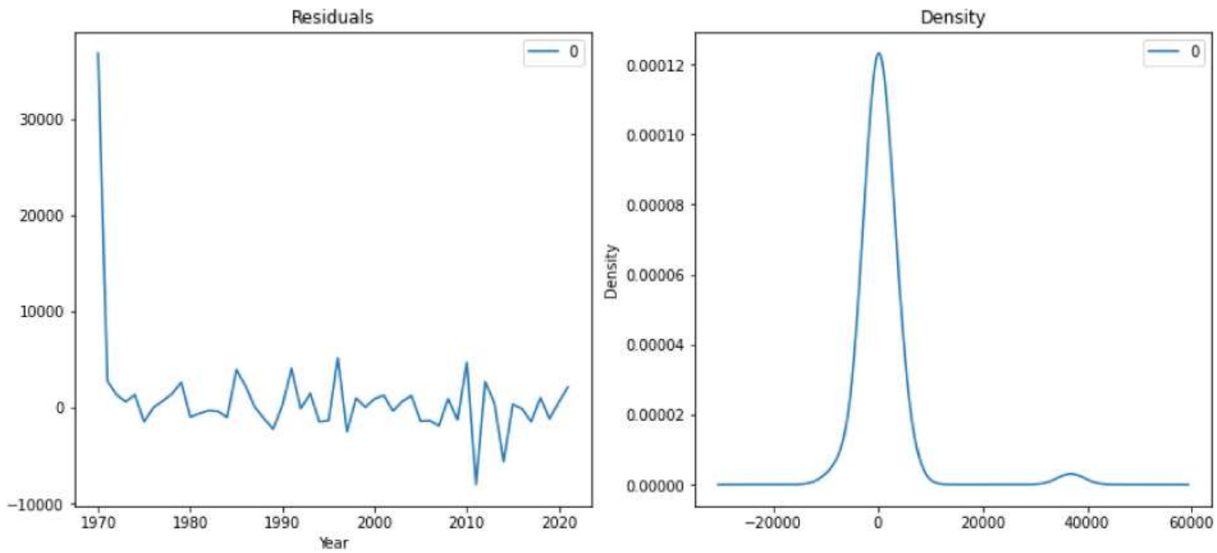

Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

p= 3, d=1, q=1

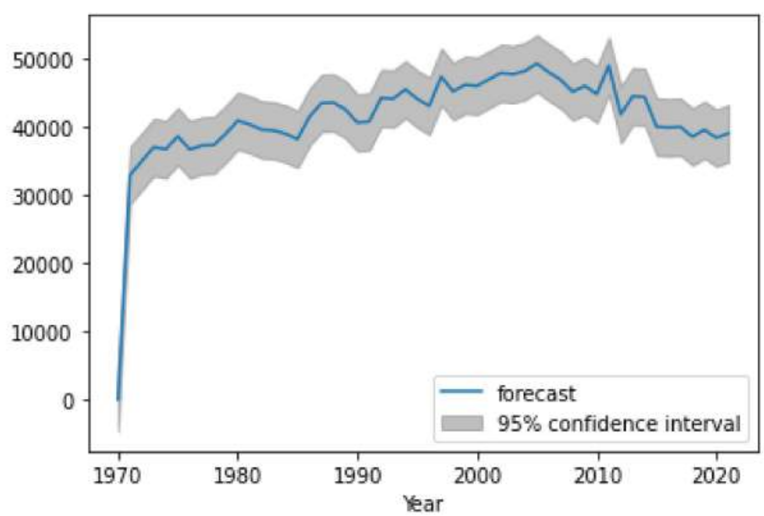
In [6]:
```
from statsmodels.tsa.arima.model import ARIMA
    #ARIMA Model
model_dom = ARIMA(dom.Total, order=(2,1,1)) #changed to (2,1,1) for better p>|z|
model_fit = model_dom.fit()
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Total   No. Observations:                   52
Model:                 ARIMA(2, 1, 1)   Log Likelihood                -466.160
Date:                Mon, 02 Jan 2023   AIC                            940.319
Time:                        17:53:11   BIC                            948.046
Sample:                             0   HQIC                           943.272
                                 - 52
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.0226      0.295     -3.465      0.001      -1.601      -0.444
ar.L2         -0.1465      0.056     -2.618      0.009      -0.256      -0.037
ma.L1          0.8891      0.294      3.022      0.003       0.313       1.466
sigma2      4.579e+06   4.02e-08   1.14e+14      0.000    4.58e+06    4.58e+06
===============================================================================
====
Ljung-Box (L1) (Q):                   5.03   Jarque-Bera (JB):                2
0.98
Prob(Q):                              0.02   Prob(JB):
0.00
Heteroskedasticity (H):               3.14   Skew:                            -
0.75
Prob(H) (two-sided):                  0.02   Kurtosis:
5.76
===============================================================================
====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
[2] Covariance matrix is singular or near-singular, with condition number 8.55e
+29. Standard errors may be unstable.

C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

In [77]:
```
residuals = pd.DataFrame(model_fit.resid)
fig, ax = plt.subplots(1,2, figsize=(14,6))
residuals.plot(title="Residuals", ax=ax[0])
residuals.plot(kind='kde', title='Density', ax=ax[1])
plt.show()
```



In [100]:
```
from statsmodels.graphics.tsaplots import plot_predict
plot_predict(model_fit, dynamic=False)
plt.show()
```




```
################################################################################
################################################################################
################################################################################
```

In [7]:
```
# Create Training and Test
train = dom.Total[:42]
test = dom.Total[42:]
```

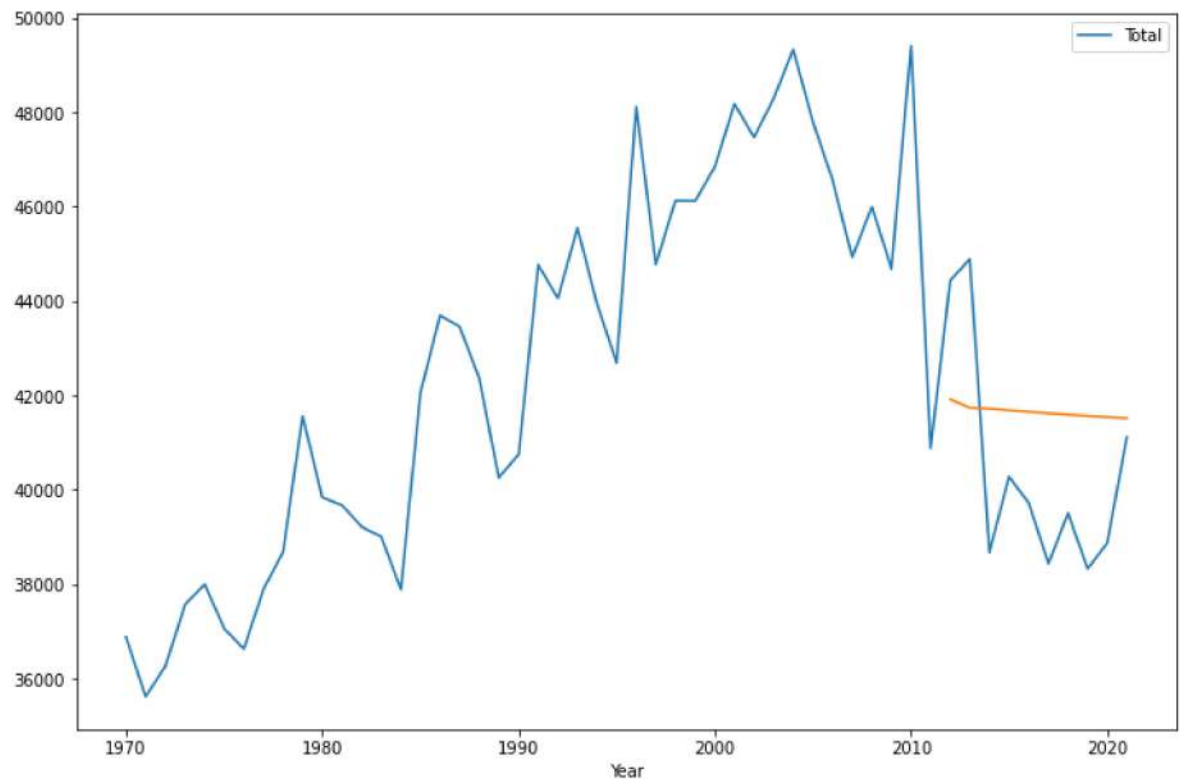
In [8]:
```
# Build Model
# model = ARIMA(train, order=(2,1,1))
model_domestic = ARIMA(train, order=(2, 1,1))
fitted = model_domestic.fit()
print(fitted.summary())

# Forecast
fc = fitted.forecast(len(test),alpha=0.05)

# Make as pandas series
fc_series = pd.Series(list(fc), index=test.index)
fc_series

plt.figure(figsize=(12,8))
sns.lineplot(data=dom)
sns.lineplot(data=fc_series)
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\statespace\sarimax.py:966: UserWarning: Non-stationary starting autore
gressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\statespace\sarimax.py:978: UserWarning: Non-invertible starting MA par
ameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:834: ValueWarning: No supported index is available.
Prediction results will be given with an integer index beginning at `start`.
  return get_prediction_index(
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Total   No. Observations:                   42
Model:                 ARIMA(2, 1, 1)   Log Likelihood                -374.525
Date:                Mon, 02 Jan 2023   AIC                            757.049
Time:                        17:53:15   BIC                            763.904
Sample:                             0   HQIC                           759.545
                                 - 42
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.8146      0.221      3.692      0.000       0.382       1.247
ar.L2          0.1193      0.069      1.721      0.085      -0.017       0.255
```

```
ma.L1          -0.9225      0.232     -3.979      0.000      -1.377      -0.468
sigma2       4.507e+06   5.12e-08    8.8e+13      0.000    4.51e+06    4.51e+06
===============================================================================
====
Ljung-Box (L1) (Q):                   3.97   Jarque-Bera (JB):                1
4.92
Prob(Q):                              0.05   Prob(JB):
0.00
Heteroskedasticity (H):               4.58   Skew:                            -
0.44
Prob(H) (two-sided):                  0.01   Kurtosis:
5.82
===============================================================================
====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
[2] Covariance matrix is singular or near-singular, with condition number 2.28e
+29. Standard errors may be unstable.
```

Out[8]: <AxesSubplot:xlabel='Year'>

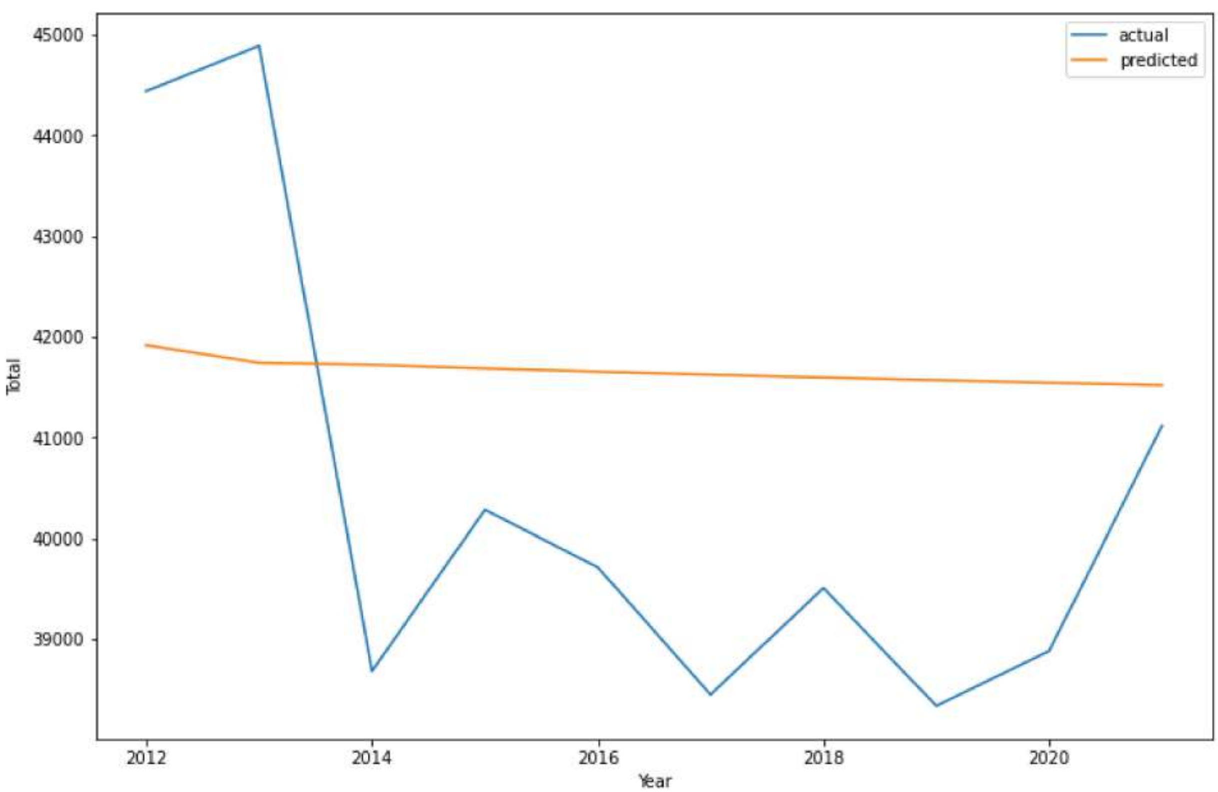


In [29]:
```
mape = np.mean(np.abs(pd.Series(fc) - test.values)/np.abs(test.values))  # MAPE
mape*100 # score implies accuracy
```

Out[29]: 5.877455223147946

In [12]:
```
plt.figure(figsize=(12,8))
a = sns.lineplot(data=test)
sns.lineplot(data=fc_series)
a.legend(['actual','predicted'])
```

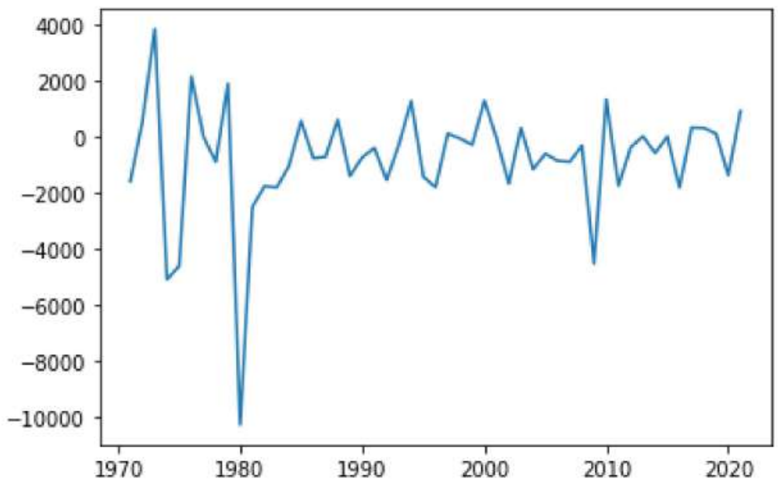
Out[12]: <matplotlib.legend.Legend at 0x249379e2020>

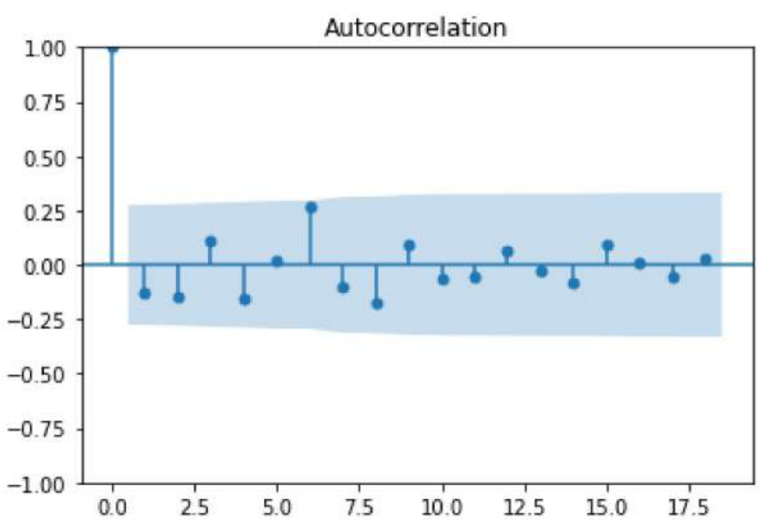


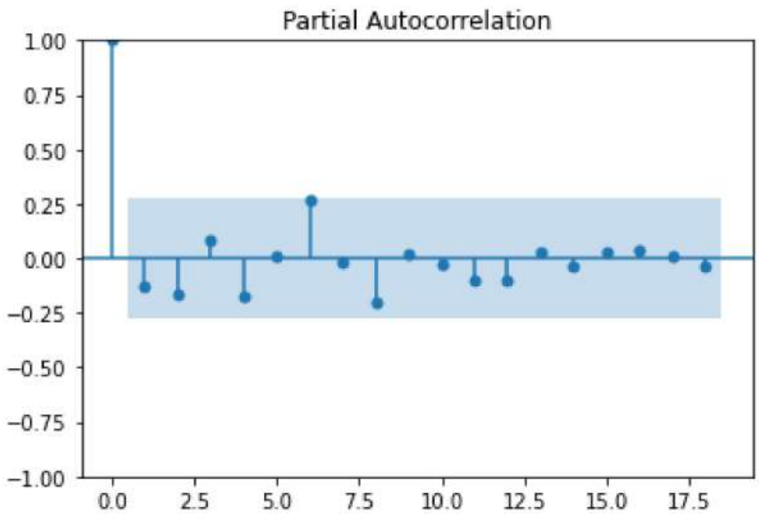
- **ARIMA ON INDUSTRIAL SECTOR**

In [175]:
```
#plotting to find p and q
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plt.plot(ind.Total.diff().dropna())
plot_acf(ind.Total.diff().dropna()) #to measure moving average

plot_pacf(ind.Total.diff().dropna()) #to measure auto regressive
plt.show()
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\graphics\tsaplots.py:348: FutureWarning: The default method 'yw' can produ
ce PACF values outside of the [-1,1] interval. After 0.13, the default will cha
nge tounadjusted Yule-Walker ('ywm'). You can use this method now by setting me
thod='ywm'.
  warnings.warn(
```






Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

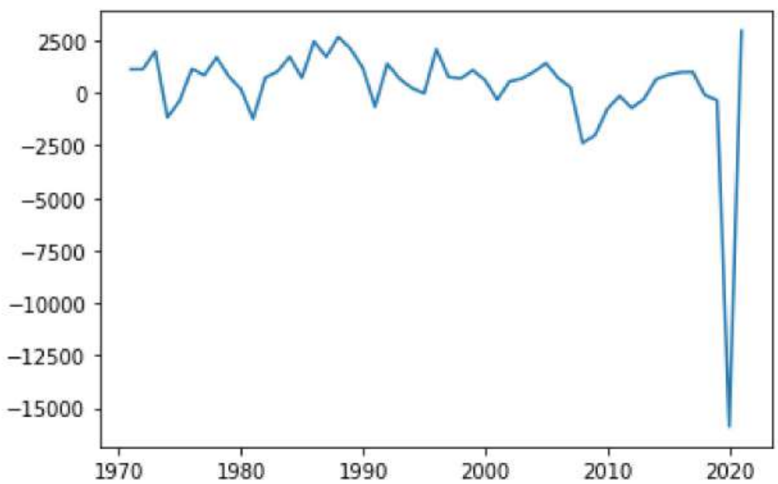
In [180]:
```
from statsmodels.tsa.arima.model import ARIMA
    #ARIMA Model
model_ind = ARIMA(ind.Total, order=(1,1,1)) #changed to (1,1,1) for better p>|z|
model_fit = model_ind.fit()
print(model_fit.summary())
```

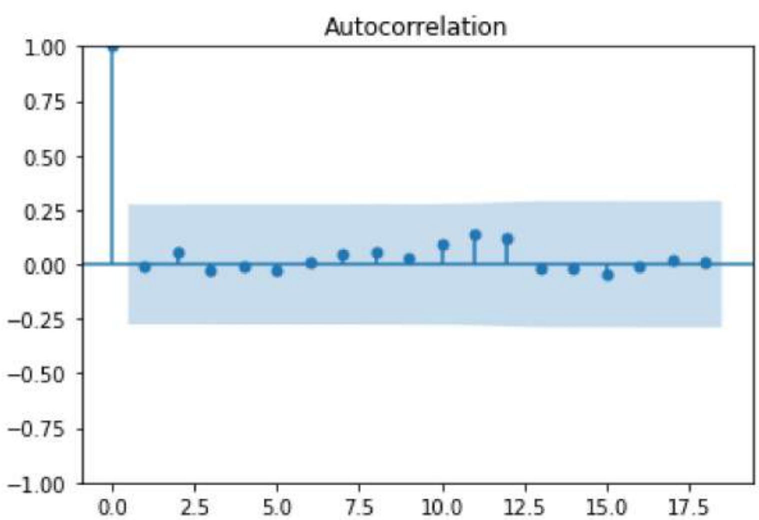
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Total   No. Observations:                   52
Model:                 ARIMA(1, 1, 1)   Log Likelihood                -462.121
Date:                Mon, 26 Dec 2022   AIC                            930.241
Time:                        22:02:10   BIC                            936.037
Sample:                             0   HQIC                           932.456
                                 - 52
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.9539      0.023     41.856      0.000       0.909       0.999
ma.L1         -0.9997      0.115     -8.694      0.000      -1.225      -0.774
sigma2      4.392e+06   2.62e-08   1.68e+14      0.000    4.39e+06    4.39e+06
===============================================================================
====
Ljung-Box (L1) (Q):                   0.95   Jarque-Bera (JB):                 9
9.00
Prob(Q):                              0.33   Prob(JB):
0.00
Heteroskedasticity (H):               0.22   Skew:                             -
1.33
Prob(H) (two-sided):                  0.00   Kurtosis:
9.28
===============================================================================
====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
[2] Covariance matrix is singular or near-singular, with condition number 1.48e
+29. Standard errors may be unstable.

C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

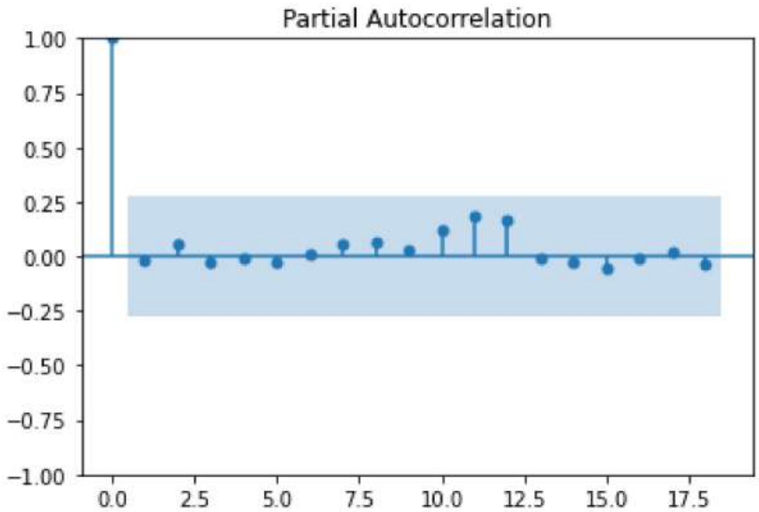
- **ARIMA ON TRANSPORT SECTOR**

In [187]:
```
#plotting to find p and q
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plt.plot(tra.Total.diff().dropna())
plot_acf(tra.Total.diff().dropna()) #to measure moving average

plot_pacf(tra.Total.diff().dropna()) #to measure auto regressive
plt.show()
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\graphics\tsaplots.py:348: FutureWarning: The default method 'yw' can produ
ce PACF values outside of the [-1,1] interval. After 0.13, the default will cha
nge tounadjusted Yule-Walker ('ywm'). You can use this method now by setting me
thod='ywm'.
  warnings.warn(
```






Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

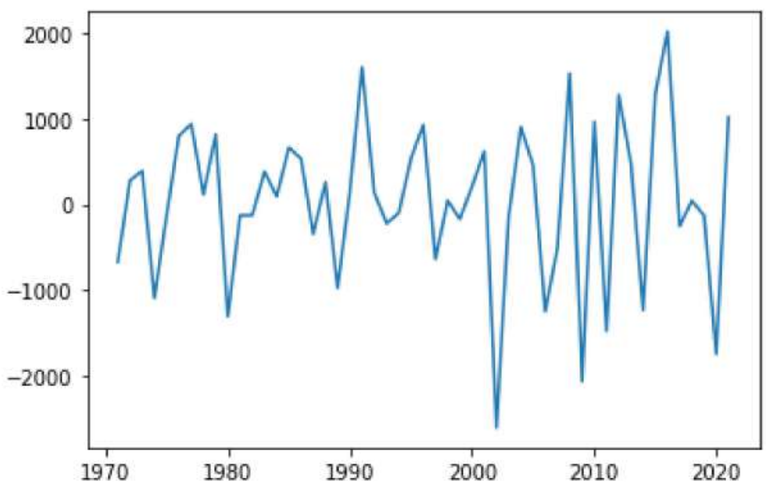
In [188]:
```
from statsmodels.tsa.arima.model import ARIMA
    #ARIMA Model
model_tra = ARIMA(tra.Total, order=(1,1,1)) #changed to (1,1,1) for better p>|z|
model_fit = model_tra.fit()
print(model_fit.summary())
```

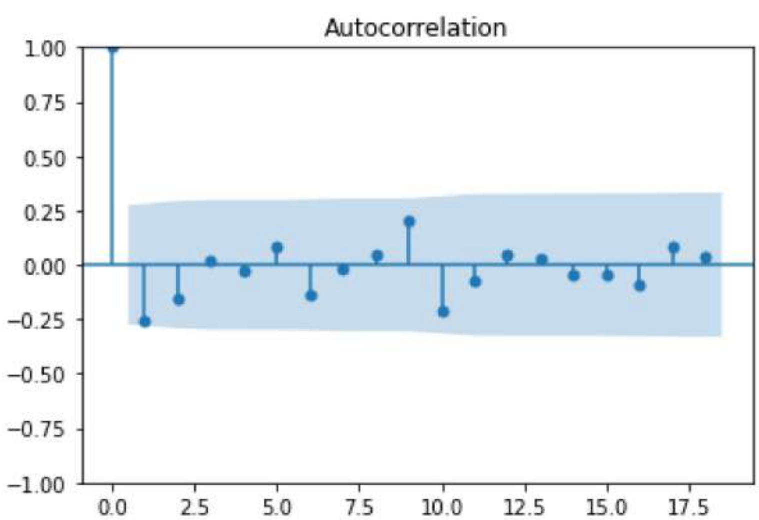
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Total   No. Observations:                   52
Model:                 ARIMA(1, 1, 1)   Log Likelihood                -471.419
Date:                Mon, 26 Dec 2022   AIC                            948.839
Time:                        22:18:00   BIC                            954.634
Sample:                             0   HQIC                           951.053
                                 - 52
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.9512      0.173      5.507      0.000       0.613       1.290
ma.L1         -0.9198      0.221     -4.164      0.000      -1.353      -0.487
sigma2      6.256e+06   5.06e+05     12.374      0.000    5.27e+06    7.25e+06
================================================================================
====
Ljung-Box (L1) (Q):                   0.36   Jarque-Bera (JB):                 201
8.81
Prob(Q):                              0.55   Prob(JB):
0.00
Heteroskedasticity (H):              15.03   Skew:                               -
4.98
Prob(H) (two-sided):                  0.00   Kurtosis:                           3
2.17
================================================================================
====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).

C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\stats
models\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was prov
ided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\stats
models\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was prov
ided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\stats
models\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was prov
ided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

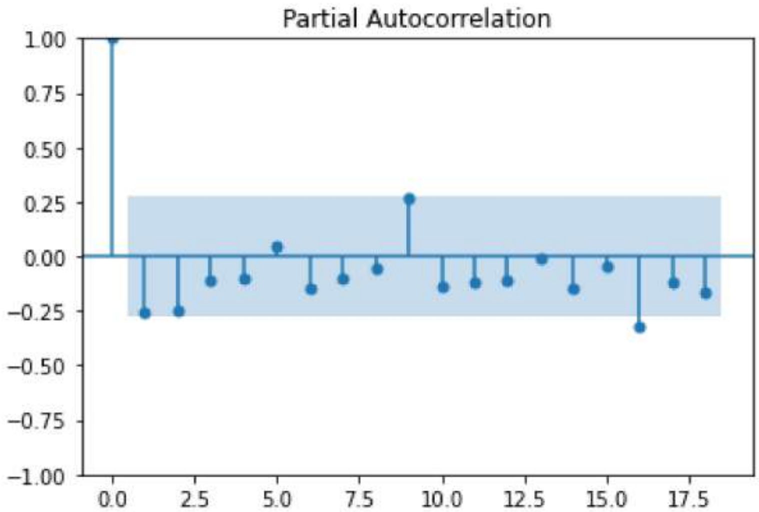
- **ARIMA ON SERVICES SECTOR**

In [195]:
```
#plotting to find p and q
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plt.plot(ser.Total.diff().dropna())
plot_acf(ser.Total.diff().dropna()) #to measure moving average

plot_pacf(ser.Total.diff().dropna()) #to measure auto regressive
plt.show()
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\graphics\tsaplots.py:348: FutureWarning: The default method 'yw' can produ
ce PACF values outside of the [-1,1] interval. After 0.13, the default will cha
nge tounadjusted Yule-Walker ('ywm'). You can use this method now by setting me
thod='ywm'.
  warnings.warn(
```





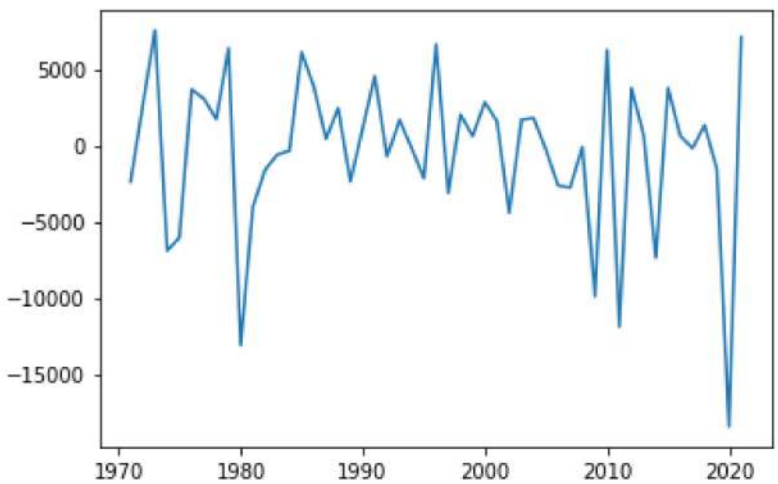

Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

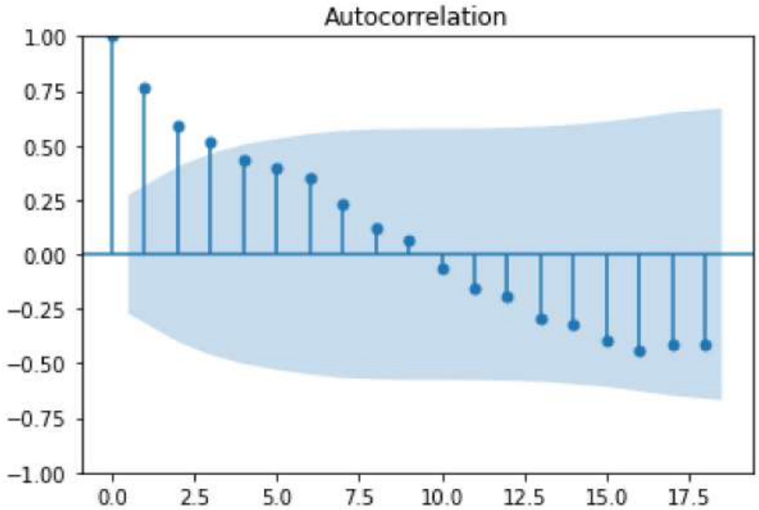
In [204]:
```
from statsmodels.tsa.arima.model import ARIMA
    #ARIMA Model
model_ser = ARIMA(ser.Total, order=(1,1,2)) #changed to (1,1,1) for better p>|z|
model_fit = model_ser.fit()
print(model_fit.summary())
```

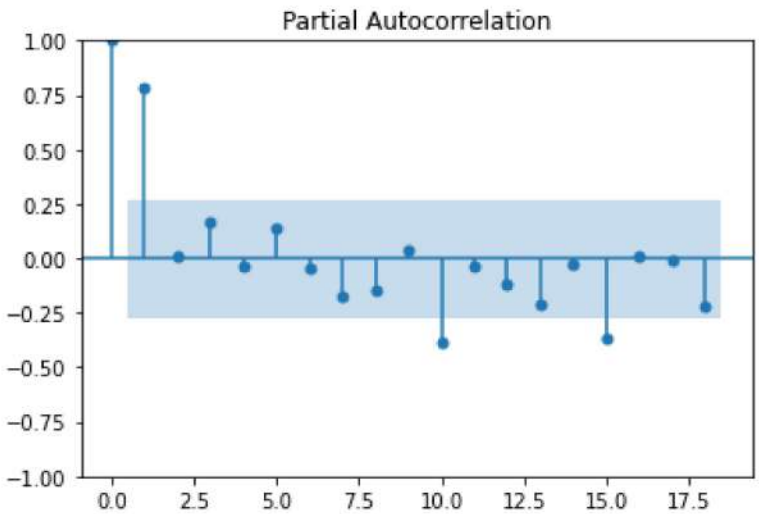
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Total   No. Observations:                   52
Model:                 ARIMA(1, 1, 2)   Log Likelihood                -418.593
Date:                Mon, 26 Dec 2022   AIC                            845.186
Time:                        22:24:28   BIC                            852.913
Sample:                             0   HQIC                           848.139
                                 - 52
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8425      0.162     -5.212      0.000      -1.159      -0.526
ma.L1          0.7117      0.199      3.581      0.000       0.322       1.101
ma.L2         -0.2159      0.114     -1.894      0.058      -0.439       0.008
sigma2       7.47e+05   1.31e+05      5.697      0.000     4.9e+05       1e+06
===============================================================================
====
Ljung-Box (L1) (Q):                   0.73   Jarque-Bera (JB):                 
2.32
Prob(Q):                              0.39   Prob(JB):                         
0.31
Heteroskedasticity (H):               3.29   Skew:                            -
0.43
Prob(H) (two-sided):                  0.02   Kurtosis:                         
3.60
===============================================================================
====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).

C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\stats
models\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was prov
ided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\stats
models\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was prov
ided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\stats
models\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was prov
ided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

In [168]:
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
# TOTAL IN UK

total = dom['Total']+ind['Total']+tra['Total']+ser['Total']
total_energy = pd.DataFrame(total)

#plotting to find p and q

plt.plot(total_energy.Total.diff().dropna())
plot_acf(total_energy.Total.dropna()) #to measure moving average

plot_pacf(total_energy.Total.dropna()) #to measure auto regressive
plt.show()
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\graphics\tsaplots.py:348: FutureWarning: The default method 'yw' can produ
ce PACF values outside of the [-1,1] interval. After 0.13, the default will cha
nge tounadjusted Yule-Walker ('ywm'). You can use this method now by setting me
thod='ywm'.
  warnings.warn(
```




Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

In [73]:
```
from statsmodels.tsa.arima.model import ARIMA
    #ARIMA Model
model_ser = ARIMA(total_energy.Total, order=(2,1,2)) #changed to (1,1,1) for bett
model_fit = model_ser.fit()
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Total   No. Observations:                   52
Model:                 ARIMA(2, 1, 2)   Log Likelihood                -503.785
Date:                Fri, 30 Dec 2022   AIC                           1017.570
Time:                        21:26:14   BIC                           1027.229
Sample:                             0   HQIC                          1021.261
                                 - 52
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.9414      0.044    -21.502      0.000      -1.027      -0.856
ar.L2         -0.9934      0.030    -33.246      0.000      -1.052      -0.935
ma.L1          0.9275      0.056     16.677      0.000       0.818       1.037
ma.L2          0.9908      0.081     12.256      0.000       0.832       1.149
sigma2      2.388e+07   1.14e-09    2.1e+16      0.000    2.39e+07    2.39e+07
===============================================================================
====
Ljung-Box (L1) (Q):                   2.01   Jarque-Bera (JB):                 1
4.28
Prob(Q):                              0.16   Prob(JB):                         
0.00
Heteroskedasticity (H):               1.90   Skew:                             -
1.08
Prob(H) (two-sided):                  0.20   Kurtosis:                         
4.43
===============================================================================
====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
[2] Covariance matrix is singular or near-singular, with condition number 9.06e
+31. Standard errors may be unstable.


C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

ARIMA FUNCTION FOR ALL SECTORS

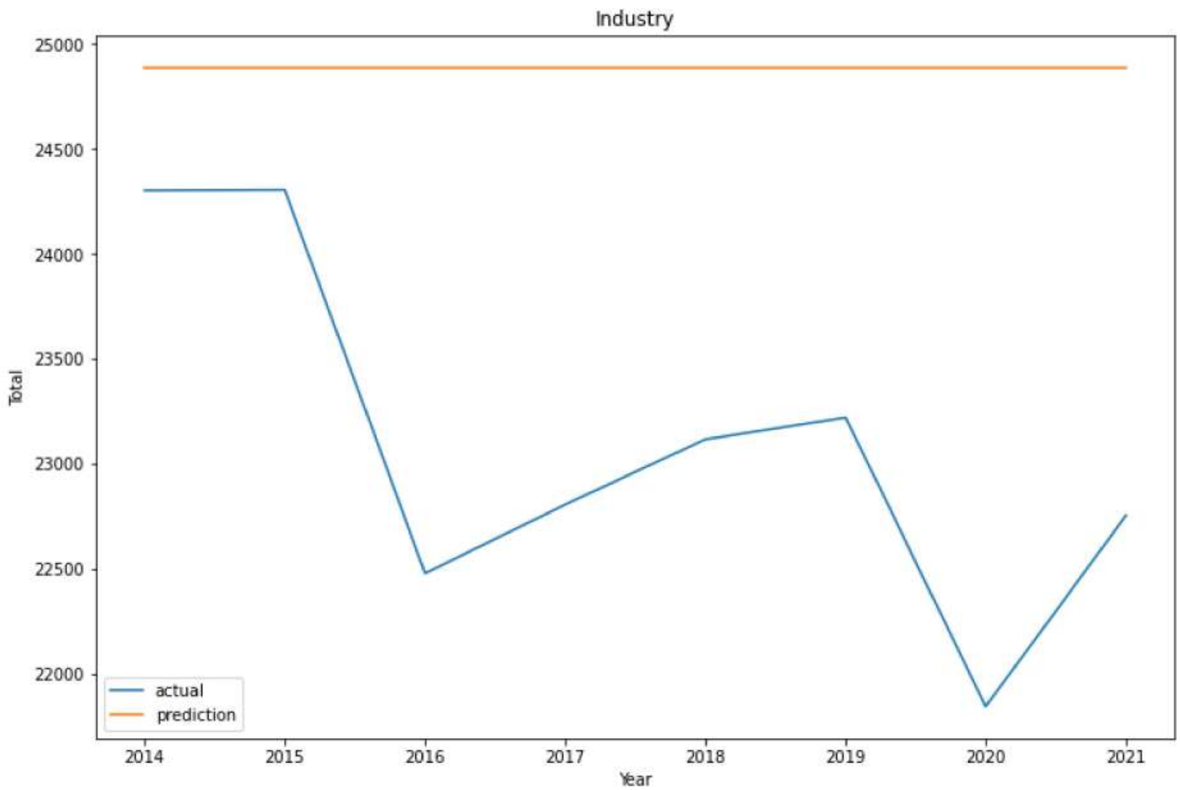
In [35]:
```
from statsmodels.tsa.arima.model import ARIMA
def predict(df,o,titlex):
    # Create Training and Test
    train = df.Total[:44]
    test = df.Total[44:]

    # Build Model
    model = ARIMA(train, order=o)
    fitted = model.fit()
    print(fitted.summary())

    # Forecast
    fc = fitted.forecast(len(test),alpha=0.05)

    # Make as pandas series
    fc_series = pd.Series(list(fc), index=test.index)
    fc_series

    plt.figure(figsize=(12,8))
#     graph = sns.lineplot(data=df)
#     sns.lineplot(data=fc_series)
#     graph.plot()

    mape = np.mean(np.abs(pd.Series(fc) - test.values)/np.abs(test.values))  # MA
    print("\n MAPE : ",mape*100,'%') # score implies accuracy
    me = np.mean(fc - test.values)             # ME
    print("\n me : ",me*100,'%')
    mae = np.mean(np.abs(fc - test.values))    # MAE
    print("\n mae : ",mae*100,'%')
    mpe = np.mean((fc - test.values)/test.values)   # MPE
    print("\n mpe : ",mpe*100,'%')
    rmse = np.mean((fc - test.values)**2)**.5  # RMSE
    print("\n rmse : ",rmse*100,'%')
    corr = np.corrcoef(fc, test.values)[0,1]   # corr
    print("\n mae : ",corr*100,'%')

    a = sns.lineplot(data=test)
    sns.lineplot(data=fc_series)
    a.legend(['actual','prediction'])
    a.set_title(titlex)

predict(ind,(0,1,0),'Industry')
```

```
                               SARIMAX Results
==============================================================================
=
Dep. Variable:                  Total   No. Observations:                    4
4
Model:                 ARIMA(0, 1, 0)   Log Likelihood                 -395.07
6
Date:                Mon, 02 Jan 2023   AIC                             792.15
2
Time:                        18:25:44   BIC                             793.91
4
Sample:                             0   HQIC                            792.80
2
                                 - 44
```

```
Covariance Type:                  opg
==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
sigma2      5.468e+06   5.53e+05      9.896      0.000    4.39e+06    6.55e+0
6
==============================================================================
======
Ljung-Box (L1) (Q):                   0.76   Jarque-Bera (JB):
91.39
Prob(Q):                              0.38   Prob(JB):
0.00
Heteroskedasticity (H):               0.17   Skew:
-1.79
Prob(H) (two-sided):                  0.00   Kurtosis:
9.18
==============================================================================
======

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (comple
x-step).

 MAPE :  7.849948938840565 %

 me :  178437.5 %

 mae :  178437.5 %

 mpe :  7.849948938840565 %

 rmse :  195387.21490926677 %

 mae :  nan %
```

```
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:471: ValueWarning: An unsupported index was provided
and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmo
dels\tsa\base\tsa_model.py:834: ValueWarning: No supported index is available.
Prediction results will be given with an integer index beginning at `start`.
  return get_prediction_index(
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\l
ib\function_base.py:2829: RuntimeWarning: invalid value encountered in true_div
ide
```

```
  c /= stddev[:, None]
C:\Users\ACER\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\l
ib\function_base.py:2830: RuntimeWarning: invalid value encountered in true_div
ide
  c /= stddev[None, :]
```



AUTO ARIMA

In [20]:
```
from statsmodels.tsa.arima_model import ARIMA

import pmdarima as pm

def fun(data):
    train = data.Total[:42]
    test = data.Total[42:]

    model = pm.auto_arima(train, start_p=0, start_q=0,
                          test='adf',       # use adftest to find optimal 'd'
                          max_p=3, max_q=3, # maximum p and q
                          m=1,              # frequency of series
                          d=None,           # let model determine 'd'
                          seasonal=False,   # No Seasonality
                          start_P=0,
                          D=1,
                          trace=True,
                          error_action='warn',
                          suppress_warnings=True,
                          stepwise=True,n_fits=20)

    print(model.summary())
fun(total_energy)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [20], in <module>
      9     model = pm.auto_arima(train, start_p=0, start_q=0,
     10                           test='adf',       # use adftest to find optim
al 'd'
     11                           max_p=3, max_q=3, # maximum p and q
   (...)
     19                           suppress_warnings=True,
     20                           stepwise=True,n_fits=20)
     22     print(model.summary())
---> 23 fun(total_energy)

NameError: name 'total_energy' is not defined
```

Type *Markdown* and LaTeX: α^2

In [236]:
```

```

LSTM (RNN) PREDICTION

In [259]:
```
train = tra.iloc[:42]
test = tra.iloc[42:-2]
```

In [260]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

In [261]:
```
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
```

In [262]:
```
from keras.preprocessing.sequence import TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
```

In [263]:
```
n_input = 3
n_features = 1
generator = TimeseriesGenerator(scaled_train, scaled_train, length=n_input, batch
```

In [264]:
```
# define model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

In [265]:
```
model.summary()
```

```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_11 (LSTM)              (None, 100)               40800

 dense_11 (Dense)            (None, 1)                 101

=================================================================
Total params: 40,901
Trainable params: 40,901
Non-trainable params: 0
_________________________________________________________________
```

In [266]:
```
# fit model
model.fit(generator,epochs=30)
```

```
Epoch 1/30
39/39 [==============================] - 1s 1ms/step - loss: 0.2867
Epoch 2/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0425
Epoch 3/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0139
Epoch 4/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0085
Epoch 5/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0063
Epoch 6/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0052
Epoch 7/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0049
Epoch 8/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0046
Epoch 9/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0038
Epoch 10/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0047
Epoch 11/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0038
Epoch 12/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0036
Epoch 13/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0036
Epoch 14/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0042
Epoch 15/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0028
Epoch 16/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0038
Epoch 17/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0033
Epoch 18/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0035
Epoch 19/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0027
Epoch 20/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0030
Epoch 21/30
39/39 [==============================] - 0s 2ms/step - loss: 0.0027
Epoch 22/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0030
Epoch 23/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0033
Epoch 24/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0029
Epoch 25/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0035
Epoch 26/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0025
Epoch 27/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0028
```

```
Epoch 28/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0023
Epoch 29/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0033
Epoch 30/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0031
```

Out[266]: <keras.callbacks.History at 0x1d5926b37f0>

In [267]:

```
test_predictions = []

first_eval_batch = scaled_train[-n_input:]
current_batch = first_eval_batch.reshape((1, n_input, n_features))

for i in range(len(test)):

    # get the prediction value for the first batch
    current_pred = model.predict(current_batch)[0]

    # append the prediction into the array
    test_predictions.append(current_pred)

    # use the prediction to update the batch and remove the first value
    current_batch = np.append(current_batch[:,1:,:],[[current_pred]],axis=1)
```

```
1/1 [==============================] - 0s 127ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
```

In [268]:

```
true_predictions = scaler.inverse_transform(test_predictions)
```

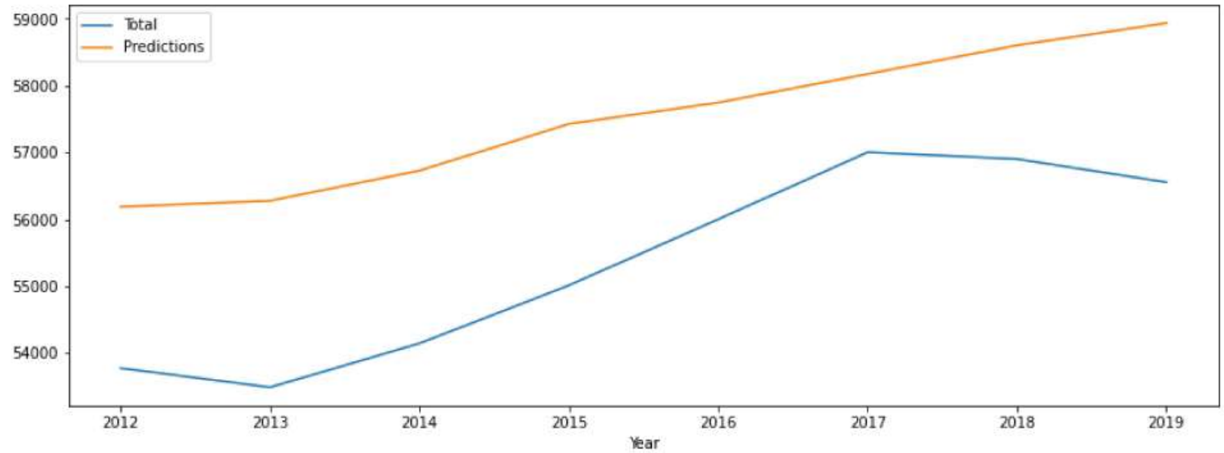
In [269]:

```
test['Predictions'] = true_predictions
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_12452\4269337381.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  test['Predictions'] = true_predictions
```

In [270]:
```
test.plot(figsize=(14,5))
```

Out[270]: <AxesSubplot:xlabel='Year'>



In [271]:
```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse=sqrt(mean_squared_error(test['Total'][:7],test['Predictions'][:7]))
print(rmse)
```

```
2183.424442897467
```

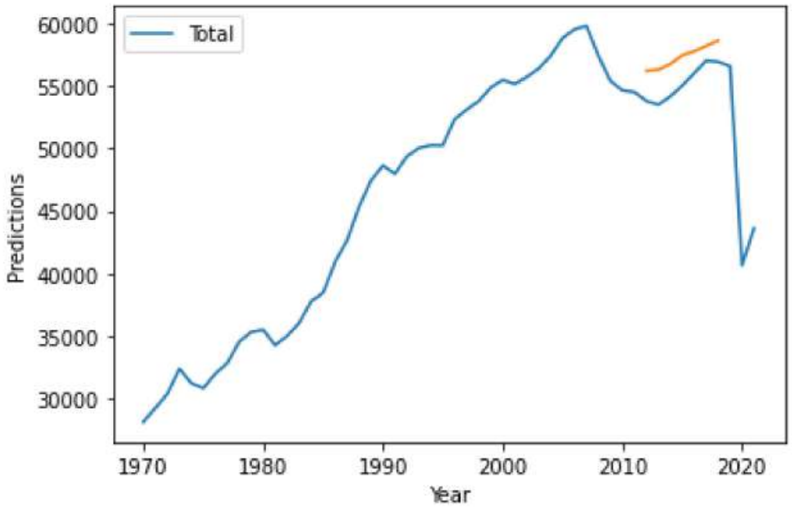
In [272]:
```
test
```

Out[272]:

Year	Total	Predictions
2012	53776	56187.260423
2013	53490	56277.437932
2014	54146	56728.242613
2015	55013	57425.981723
2016	56001	57745.154962
2017	57003	58172.169444
2018	56902	58601.387420
2019	56555	58934.344737

In [273]:
```
sns.lineplot(data=tra)
sns.lineplot(data=test['Predictions'][:7])
```

Out[273]: <AxesSubplot:xlabel='Year', ylabel='Predictions'>



In [ ]:

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing.sequence import TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
```

In [3]:
```
industry_df = pd.read_csv(r"data\energy_consuption_industry.csv")
domestic_df = pd.read_csv(r"data\energy_consuption_domestic.csv")
services_df = pd.read_csv(r"data\energy_consuption_services.csv")
transport_df = pd.read_csv(r"data\energy_consuption_transport.csv")
```

In [4]:
```
# industry_df
ind = industry_df[['Year','Total']].set_index('Year')
# domestic_df
dom = domestic_df[['Year','Total']].set_index('Year')
# services_df
ser = services_df[['Year','Total']].set_index('Year')
# transport_df
tra = transport_df[['Year','Total']].set_index('Year')
```

In [5]:
```
total = dom['Total']+ind['Total']+tra['Total']+ser['Total']
total_energy = pd.DataFrame(total)
```

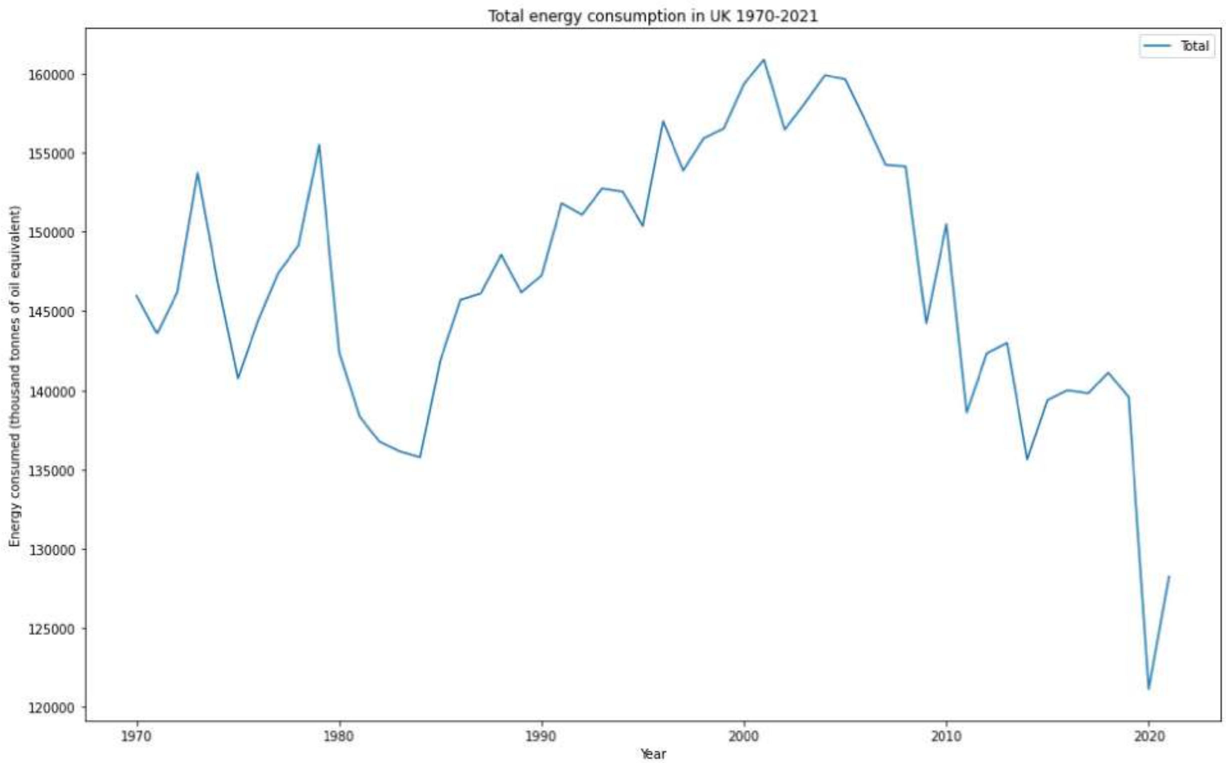
In [6]:
```
total_energy.tail(1)
```

Out[6]:

	Total
Year	
2021	128214

In [12]:
```
plt.figure(figsize=(16,10))
ax = sns.lineplot(data=total_energy)
ax.set_title('Total energy consumption in UK 1970-2021')
ax.set_ylabel('Energy consumed (thousand tonnes of oil equivalent)')
```

Out[12]:
```
Text(0, 0.5, 'Energy consumed (thousand tonnes of oil equivalent)')
```



In [5]:
```
#COMBINE INDUSTRY AND EMISSION DATA
ems = pd.read_csv(r'data\1970-2021 emission.csv')
ems = ems.set_index('Year')
industry_em = pd.merge(ems,ind,left_index=True,right_index=True)
```

LSTM

In [104]:

```
train = ser.iloc[:42]
test = ser.iloc[42:] #dont include covid data for transport
scaler = MinMaxScaler()
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
n_input = 3
n_features = 1
generator = TimeseriesGenerator(scaled_train, scaled_train, length=n_input, batch
# define model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
model.summary()
```

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_13 (LSTM)              (None, 100)               40800

 dense_13 (Dense)            (None, 1)                 101

=================================================================
Total params: 40,901
Trainable params: 40,901
Non-trainable params: 0
_________________________________________________________________
```

In [105]:
```
# fit model
model.fit(generator,epochs=30)
```

```
Epoch 1/30
39/39 [==============================] - 1s 1ms/step - loss: 0.1516
Epoch 2/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0406
Epoch 3/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0350
Epoch 4/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0358
Epoch 5/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0352
Epoch 6/30
39/39 [==============================] - 0s 2ms/step - loss: 0.0396
Epoch 7/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0362
Epoch 8/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0342
Epoch 9/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0351
Epoch 10/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0361
Epoch 11/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0339
Epoch 12/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0376
Epoch 13/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0358
Epoch 14/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0357
Epoch 15/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0344
Epoch 16/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0328
Epoch 17/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0342
Epoch 18/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0334
Epoch 19/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0337
Epoch 20/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0341
Epoch 21/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0341
Epoch 22/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0327
Epoch 23/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0333
Epoch 24/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0345
Epoch 25/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0340
Epoch 26/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0329
Epoch 27/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0334
```

```
Epoch 28/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0331
Epoch 29/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0329
Epoch 30/30
39/39 [==============================] - 0s 1ms/step - loss: 0.0366
```

Out[105]: `<keras.callbacks.History at 0x2b819672c50>`

In [106]:

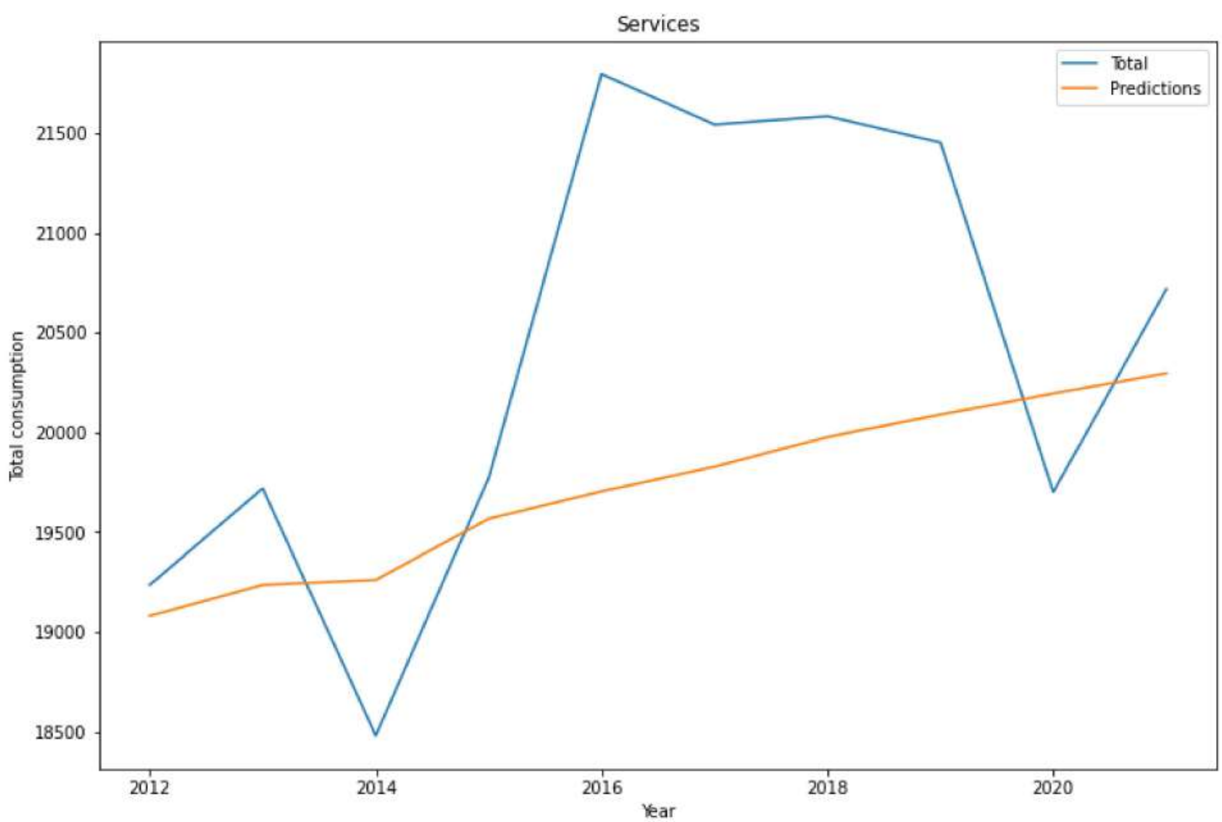
```
test_predictions = []

first_eval_batch = scaled_train[-n_input:]
current_batch = first_eval_batch.reshape((1, n_input, n_features))

for i in range(len(test)):

    # get the prediction value for the first batch
    current_pred = model.predict(current_batch)[0]

    # append the prediction into the array
    test_predictions.append(current_pred)

    # use the prediction to update the batch and remove the first value
    current_batch = np.append(current_batch[:,1:,:],[[current_pred]],axis=1)
```

```
1/1 [==============================] - 0s 127ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
```

In [107]:
```
true_predictions = scaler.inverse_transform(test_predictions)
test['Predictions'] = true_predictions
ax = test.plot(figsize=(12,8))
# ax.title("")
ax.set_ylabel("Total consumption")
ax.set_title("Services")
```

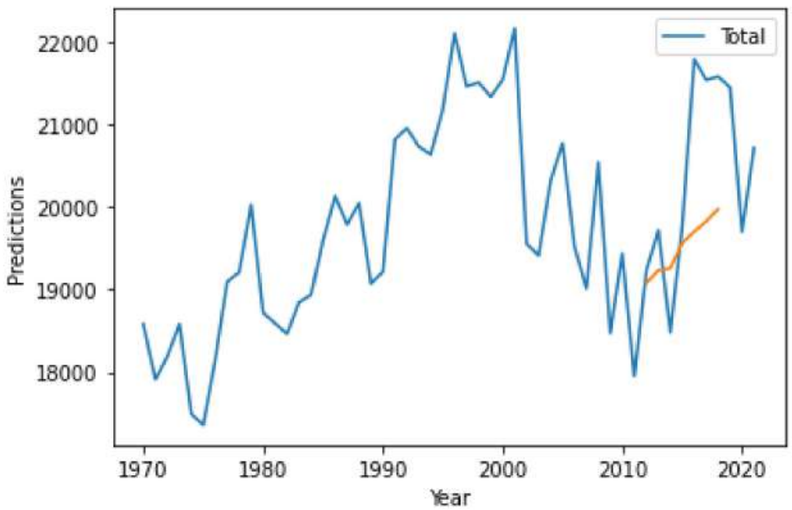
```
C:\Users\ACER\AppData\Local\Temp\ipykernel_10524\2331756986.py:2: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  test['Predictions'] = true_predictions
```

Out[107]: Text(0.5, 1.0, 'Services')



In [111]:
```
sns.lineplot(data=ser)
sns.lineplot(data=test['Predictions'][:7])
```

Out[111]:
```
<AxesSubplot:xlabel='Year', ylabel='Predictions'>
```



In [109]:
```
test
```

Out[109]:

Year	Total	Predictions
2012	19237	19081.737815
2013	19719	19236.323553
2014	18481	19261.071391
2015	19773	19567.161061
2016	21795	19704.152924
2017	21542	19828.085799
2018	21584	19976.944155
2019	21452	20089.600481
2020	19701	20194.945402
2021	20718	20296.145532

In [110]:
```
actual = list(test['Total'])
forecast = list(test['Predictions'])
APE = []

# Iterate over the list values
for day in range(len(list(test['Total']))):
    # Calculate percentage error
    per_err = (actual[day] - forecast[day]) / actual[day]
    # Take absolute value of
    # the percentage error (APE)
    per_err = abs(per_err)
    # Append it to the APE list
    APE.append(per_err)

MAPE = sum(APE)/len(APE)
print('MAPE SCORE : ',MAPE*100,'  %')
```

```
MAPE SCORE :  4.440610908371139   %
```

In [75]:
```
industry_em.describe()
```

Out[75]:

	annual emission	Total
count	52.000000	52.000000
mean	9.255763	38278.557692
std	1.794822	12416.497363
min	4.865282	21847.000000
25%	8.658587	29417.750000
50%	9.720448	35891.000000
75%	10.291534	42645.000000
max	11.818837	65149.000000

In [ ]: