IST687 Group Project Report

Ramya Chowdary Patchala

Akshay Joshi

Hemanth Chowdary Thammeneni

Kishan

Rithika Gurram

GROUP 1

Contents

Project Goal

The overarching objective of this project is to conduct an extensive and insightful analysis of a dataset centered around energy costs. Through a multifaceted exploration, the project aims to unravel the intricate relationships and patterns inherent in the data, ultimately providing valuable insights into the factors that significantly influence variations in energy expenditures.

The energy company (eSC) provides electricity to residential properties in South Carolina and a small part of North Carolina. As eSc is concerned about global warming and its impact on electricity demand, especially during hot summers. Now, eSc aims to understand key drivers of energy usage and explore strategies to reduce energy consumption without building new power plants.

The focus is on July energy usage, traditionally the highest energy usage month. Two main datasets: static house data (attributes like size) and energy usage data (hourly usage profiles for each house). Weather data and a data description file are also available. The goal is to build a predictive model for energy usage, explore factors affecting it, and propose strategies to manage peak demand.

The investigation encompasses a broad spectrum of features, ranging from lifestyle aspects such as exercise habits and smoking behavior to demographic elements including age and education levels. Geographical factors are also scrutinized, with an emphasis on understanding how different states and location types contribute to disparities in energy costs.

The project unfolds with the meticulous importation and cleanup of the dataset, ensuring data quality and coherence. Subsequently, a comprehensive data exploration and visualization phase is undertaken to lay the foundation for subsequent analyses.

Machine learning models, including linear regression and XGBoost, are introduced to predict and understand the intricate dynamics of energy costs, providing stakeholders with powerful tools for estimating and optimizing energy expenditures.

The project culminates in the development of an interactive Shiny App, fostering user engagement and empowering stakeholders to interact with the data, visualize outcomes, and derive actionable insights.

Overall, this project is driven by a holistic commitment to uncovering the nuances of energy cost determinants and providing a comprehensive framework for decision-makers to navigate the complexities of energy management.

Data Sets and Rstudio Package Libraries

- **Weather Data**

  https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/weather/2023-weatherdata/G4500010.csv

  From the above url, we are obtaining the Weather Data.

  There are 5000 houses and their details in the file like:

  1. **Date time**: character datatype. It consists of date.

  2. **Dry Bulb Temperature (C):** Number datatype. It consists of dry bulb temperature in Celsius degrees.

3. **Relative Humidity:** Number datatype. It consists of humidity.

4. **Wind Speed (m/s):** Number datatype. It consists of speed of wind in m.s (meters per second)

5. **Wind Direction (Deg):** Number datatype. It consists of direction of wind in degrees.

6. **Global Horizontal Radiation (W/m2):** Number datatype. It consists of Global radiation in watts/sq.Meter.

7. **Direct Normal Radiation (W/m2):** Number datatype. It consists of Normal radiation in watts/sq. Meter.

8. **Diffuse Horizontal Radiation (W/m2):** Number datatype. It consists of diffuse radiation in watts/sq.Meter

- **Static House Data**

  https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/static_house_info.parquet

  1. The Static House Data file contains basic attribute information for a random sample of 5,000 single family residential homes served by the energy company (eSC) in South Carolina and North Carolina.

  2. For each house in the sample, the static data provides details on attributes that do not change over time such as the building ID, finished square footage, number of floors, heating and cooling system types, and insulation levels. The building ID can be used to link each home to its corresponding energy usage data.

  3. This static data affords valuable insights into the physical characteristics of the residential properties served by eSC. Analyzing the energy usage patterns in conjunction with the static house attributes can reveal important correlations between the physical house features and their energy consumption.

- **Energy Usage Data**

  https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/102063.parquet

  1. The energy usage data has been calibrated and validated to ensure accuracy. The per-home timeseries data enables detailed analysis of energy consumption patterns, peaks and troughs, and variability across homes andover time.

  2. By linking the building ID in these usage files with the static home attribute data, we can correlate the energyconsumption profiles with the physical characteristics of each home. This supports development of robust predictive models and segmentation analyses to identify the biggest drivers of energy demand.

LIBRARIES USED:

In order to generate the models used in this project, several package libraries were required to access the functions and capabilities as needed. The package libraries are as follows:

1. **tidyverse**: An essential collection of packages, including dplyr and ggplot2, for data manipulation and visualization, providing a tidy and efficient workflow.

2. **arrow**: Utilized for reading and writing Arrow format data, facilitating efficient and fast data processing.

3. **data.table**: Offers enhanced data manipulation capabilities, particularly useful for large datasets, contributing to the overall efficiency of the analysis.

4. **lubridate**: Specialized in handling date and time data, enabling the project to explore temporal patterns and trends within the dataset.

5. **caret**: A comprehensive package for machine learning, providing tools for model training, evaluation, and tuning.

6. **glue**: Employed for string interpolation, especially useful for constructing URLs and handling dynamic parts of file paths.

7. **purrr**: A functional programming toolkit, facilitating iteration and mapping operations, extensively used in the context of data cleaning and manipulation.

8. **xgboost**: An implementation of the XGBoost algorithm, a powerful tool for predictive modeling, employed for creating sophisticated machine learning models.

9. **shiny**: Enables the development of interactive web applications directly from R, crucial for creating the Shiny App that enhances user engagement and data exploration.

Data Import and Clean Up

The process of data importing and cleanup within this project is a pivotal phase aimed at ensuring the dataset's quality, coherence, and overall readiness for subsequent analytical endeavours. This multifaceted process can be dissected into several key components that collectively contribute to the robustness of the dataset:

Initiating with the loading of essential R libraries, including 'tidyverse,' 'arrow,' 'data.table,' 'lubridate,' and 'caret,' among others, a comprehensive suite of tools is established to facilitate efficient data manipulation, visualisation, and machine learning throughout the project lifecycle.The subsequent step involves the importation of the dataset into the R environment. Leveraging the 'arrow::read_parquet()' function for the primary dataset and 'read_csv_arrow()' for metadata, this approach ensures both efficiency in handling large datasets and suitability for metadata stored in CSV format.

The initial exploration phase is characterised by the extraction of unique identifiers, such as building IDs and county names. This foundational step lays the groundwork for subsequent analyses and ensures a fundamental understanding of the dataset's structural nuances.

Integral to the data preparation process is the definition of custom functions. The 'Electricity Usage Function' is crafted to effectively handle data at the granularity of individual buildings. This function encompasses tasks such as data extraction, filtering for specific months, handling negative values, calculating total energy, and aggregating information on a daily basis.

```
# Function to obtain energy related data using building id
obtain_energy <- function(bldg_id) {

  url <- glue::glue("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/",bldg_id,".pa
rquet")

  df <- arrow::read_parquet(url)

  #Choosing the data of energy consumption in month of July
  df <-  df %>% filter(month(time) == 7)

  # Checking the numeric variables for negative values and converting them to positive values
  numerical_cols <- sapply(df, is.numeric)
  df[, numerical_cols] <- lapply(df[, numerical_cols], function(x) ifelse(x < 0, abs(x), x))

  # Obtaining total energy consumption for hour
  df$total_energy_hour <- rowSums(df[,-which(names(df) == 'time')])

  df <- df[, c('time', 'total_energy')]

  df$time <- as.POSIXct(df$time, tz= 'EST', origin = '1970-01-01')

  # Aggregating by day, obtaining daily total energy consumption for month of july
  daily_df <- df %>%
              mutate(date = as.Date(time)) %>%
              mutate(bldg_id = bldg_id) %>%
              group_by(bldg_id, date) %>%
              summarize(total_energy = sum(total_energy_hour, na.rm = TRUE), .groups = 'drop')

  return(daily_df)
}

# Calling obtain_energy function for all building ids
total_energy_df <- purrr::map_dfr(bldg_id, obtain_energy)
```

Simultaneously, a 'Weather Data Extraction' function is introduced to process and compute daily medians for various weather parameters, including radiation, temperature, humidity, and wind speed, for each county.

```
# Function to obtain weather related details for counties.
obtain_weather_data <- function(county) {

    url <- glue::glue(paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/weather/2023-weather
-data/", county,".csv"))
    df <- read.csv(url)
    df$date_time <- as.Date(df$date_time, format = "%Y-%m-%d %H:%M:%S")

    # Obtaining median of weather details for every day of Month July
    daily_data <- df %>%
                  group_by(date_time = as.Date(date_time, tz='GMT')) %>%
                  summarise_all(mean)  %>%
                  filter(month(date_time) == 7) %>%
                  mutate (county_id = county) %>%
                  group_by(county_id, date_time) %>%
                summarise(Direct_Normal_Radiation = median(Direct.Normal.Radiation..W.m2., na.rm = TRUE),
                  Diffuse_Horizontal_Radiation = median(Diffuse.Horizontal.Radiation..W.m2., na.rm = TRUE),
                  Dry_Bulb_Temperature = median(Dry.Bulb.Temperature...C., na.rm = TRUE),
                    Relative_Humidity = median(Relative.Humidity...., na.rm = TRUE),
                    Wind_Speed = median(Wind.Speed..m.s., na.rm = TRUE),
                  Wind_Direction = median(Wind.Direction..Deg., na.rm = TRUE),
                  Global_Horizontal_Radiation = median(Global.Horizontal.Radiation..W.m2., na.rm = TRUE), .groups
= 'drop') %>%
                  mutate(date = date_time) %>%
                  mutate(in.county = county_id)

    return(daily_data)
}

#Calling obtain_weather_data function for all counties
weather_df <- purrr::map_dfr(counties, obtain_weather_data)
```

The amalgamation of datasets is a crucial step in enriching the primary dataset. Merging it with computed total energy and weather datasets enhances the analytical capabilities by incorporating information pertaining to both energy consumption and external weather factors.

Ensuring data integrity, a null value check is systematically executed for all columns. The outcomes of this check are then meticulously scrutinised to identify any missing data that might compromise the reliability of subsequent analyses.

```
# Looking for null values in all columns
null_values <- colSums(is.na(final_merged_df))

# Printing the result
print(null_values)
```

Further refinement involves the generation of frequency tables for each column. This provides valuable insights into the distribution of values, allowing for the identification of potential anomalies or issues, particularly in columns with a single value.

Columns with more than one distinct value are strategically selected, resulting in the creation of a cleaned final dataset. This curated dataset retains only those columns deemed sufficiently variable, ensuring that the dataset comprises informative features conducive to in-depth analysis.

```
# Function to check the frequency of values in column
freq_table <- function(column){
  freq_df <- data.frame(
    col_name = column,
    frequency = nrow(data.frame(table(final_merged_df[,column])))
  )
  return (freq_df)
}

freq_df <- purrr::map_dfr(colnames(final_merged_df), freq_table)

# Obtaining columns whose frequency is greater than 1
col_names <- freq_df[freq_df$frequency>1, ]$col_name

# Obtaining final merged dataset whose columns have more than 1 different values
cleaned_final_df <- final_merged_df[, col_names]
```

The process of numeric conversion and standardisation is initiated to promote consistency in units, particularly in columns relating to temperature and setpoints. This meticulous step facilitates meaningful comparisons across different variables.

```
# Maintaining the data consistency and readability by removing the units from values and converting them into
numerical type.

# Removing F from in.cooling_setpoint column
cleaned_final_df$in.cooling_setpoint <- as.numeric(sub("F", "", cleaned_final_df$in.cooling_setpoint))

# Removing F from in.cooling_setpoint_offset_magnitude column
cleaned_final_df$in.cooling_setpoint_offset_magnitude <- as.numeric(sub("F", "",
cleaned_final_df$in.cooling_setpoint_offset_magnitude))

# Removing F from in.heating_setpoint column
cleaned_final_df$in.heating_setpoint <- as.numeric(sub("F", "", cleaned_final_df$in.heating_setpoint))

# Removing F from in.heating_setpoint_offset_magnitude column
cleaned_final_df$in.heating_setpoint_offset_magnitude <- as.numeric(sub("F", "",
cleaned_final_df$in.heating_setpoint_offset_magnitude))

# Removing Hour from in.bathroom_spot_vent_hour column
cleaned_final_df$in.bathroom_spot_vent_hour <- as.numeric(sub("Hour", "",
cleaned_final_df$in.bathroom_spot_vent_hour))

# Removing Hour from in.range_spot_vent_hour column
cleaned_final_df$in.range_spot_vent_hour <- as.numeric(sub("Hour", "",
cleaned_final_df$in.range_spot_vent_hour))

# Removing ACH50 from in.infiltration column
cleaned_final_df$in.infiltration <- as.numeric(sub(" ACH50", "", cleaned_final_df$in.infiltration))

# Removing None, Car from in.geometry_garage column
cleaned_final_df$in.geometry_garage <- ifelse(cleaned_final_df$in.geometry_garage == "None", 0,
as.numeric(gsub(" Car", "", cleaned_final_df$in.geometry_garage)))
```

Given the nuanced representations in the 'Income' column, a specialised processing step is introduced to handle diverse formats, including values greater than or less than, and ranges. This results in the derivation of meaningful estimates conducive to comprehensive analysis.

```
# For the columns that have ranges, we are generally taking mean for them.
obtain_mean <- function(values) {
  if (grepl(">", values)) {
    greater_than_value <- as.numeric(gsub(">", "", values))
    return(greater_than_value + 1)
  }

  if (grepl("<", values)) {
    less_than_value <- as.numeric(gsub("<", "", values))
    return(less_than_value - 1)
  }

  split <- strsplit(values, "-")[[1]]

  split <- as.numeric(gsub("\\+", "", split))

  if (length(split) == 2) {
    return(mean(split))
  } else {
    return(split[1])
  }
}

cleaned_final_df$in.income <- sapply(cleaned_final_df$in.income, obtain_mean)

cleaned_final_df$in.income_recs_2015 <- sapply(cleaned_final_df$in.income_recs_2015, obtain_mean)

cleaned_final_df$in.income_recs_2020 <- sapply(cleaned_final_df$in.income_recs_2020, obtain_mean)
```

In summary, the data importing and cleanup steps are executed with meticulous attention to detail, involving the loading of essential libraries, importing datasets, exploration and identification of unique identifiers, function

definition for specialised data handling, merging of datasets, null value checks, frequency table generation, strategic column selection, and numeric conversions.

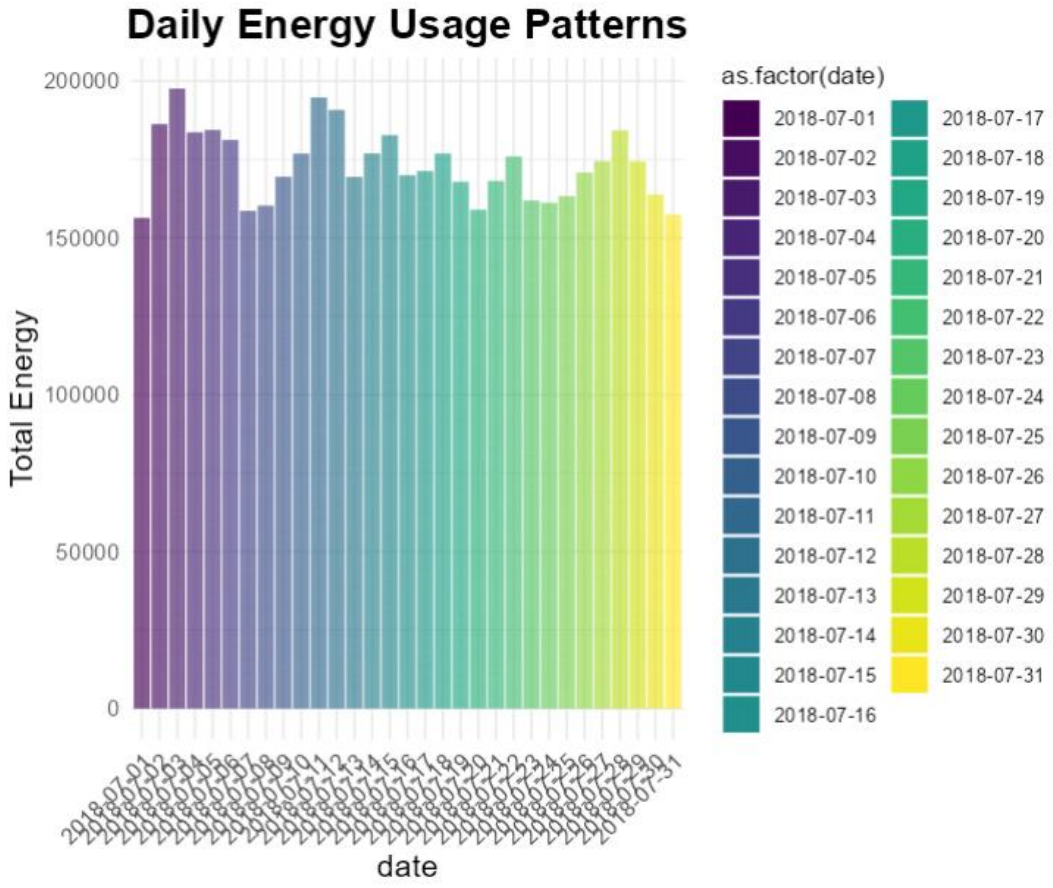
These steps collectively form a robust foundation for the subsequent phases of data exploration, visualisation, and machine learning modelling, ensuring that the dataset is primed for insightful analysis in the project's overarching objectives.

Data Exploration and Visualizations

The data was explored by creating visualizations. Several relationships were explored, and the following sections are a sample of all analysis performed.

1. **Daily Energy Usage Patterns:**

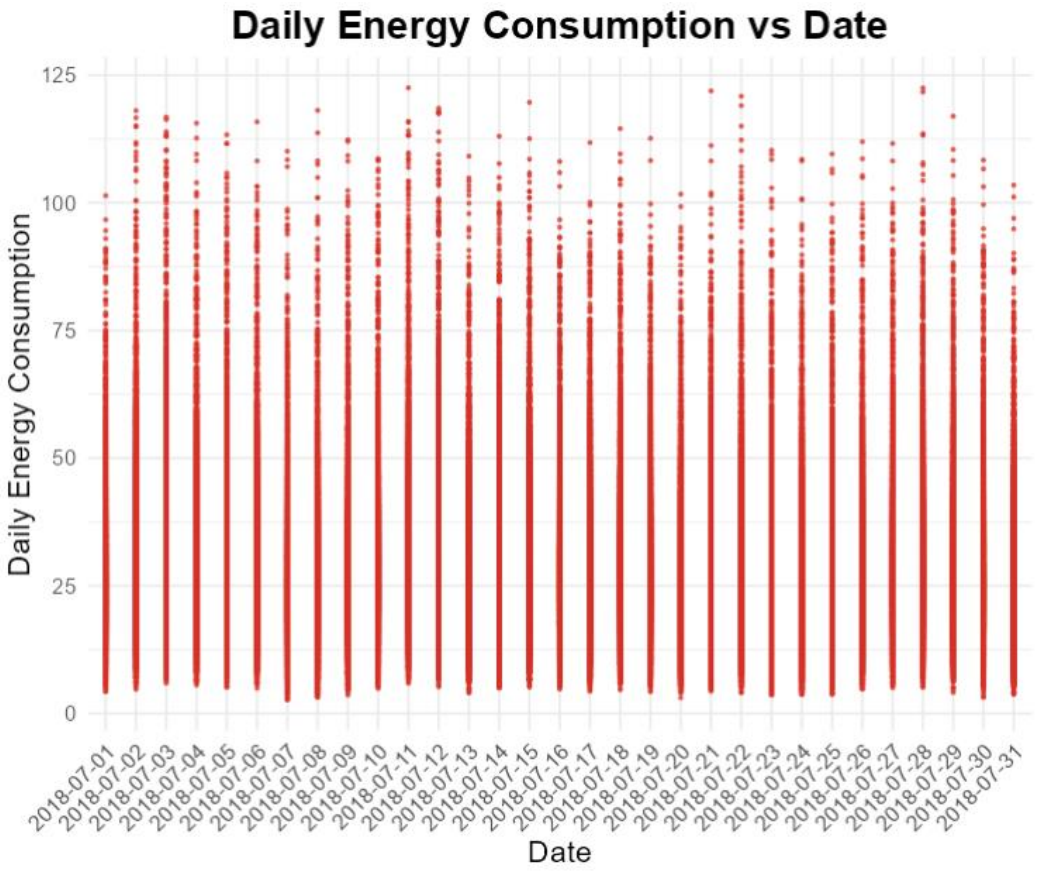
   This bar plot illustrates the daily patterns of energy consumption. Each bar represents a unique date, and the height of the bars corresponds to the total energy consumed on that specific day. The plot is colored using a distinct palette for each date, providing a visual representation of the variation in energy usage over time.



- The plot showcases the distribution of total energy consumption, allowing for insights into daily trends.
- Date labels are presented on the x-axis for easy interpretation.
- The color palette enhances the visual appeal and helps distinguish between different dates.

2. **Daily Energy Consumption vs Date**:

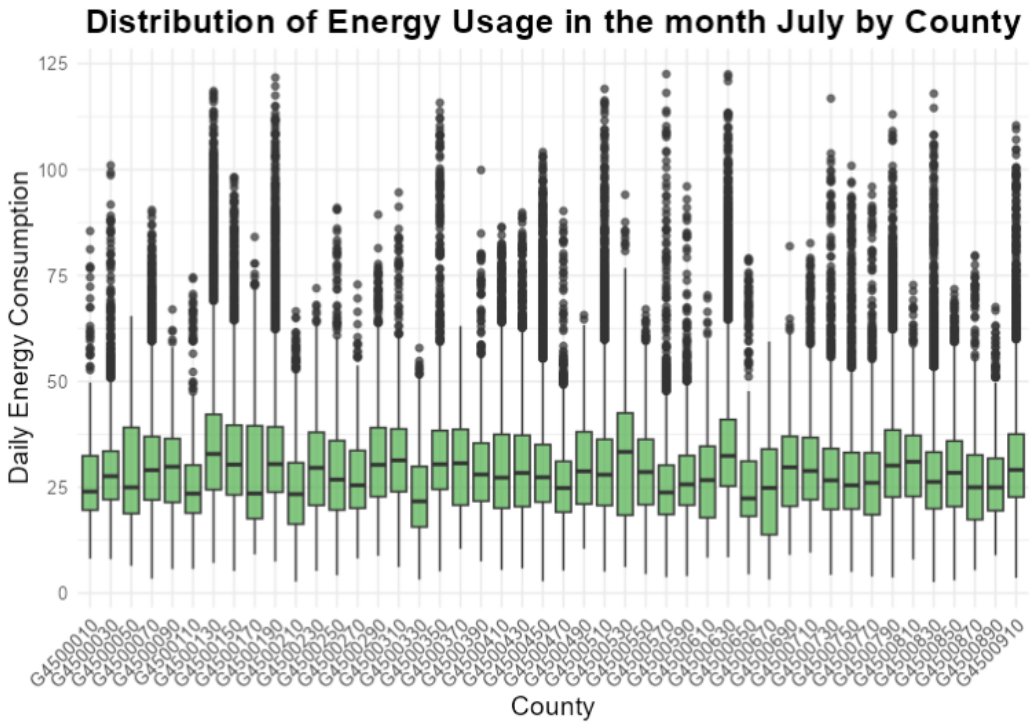
   This scatter plot visualizes the daily energy consumption trends over time. Each point on the plot represents a specific date, and its position is determined by the corresponding daily energy consumption. The size and transparency of the points enhance visibility, and a distinct color (#d73027) is used to highlight the data points. The plot provides a clear representation of the fluctuations in energy consumption on different dates.



- The scatter plot allows for the exploration of the relationship between date and daily energy consumption.
- Points are color-coded for easy identification and interpretation.
- The minimal theme enhances focus on the main insights without unnecessary distractions.

3. Distribution of Energy Usage in the month of July by County:

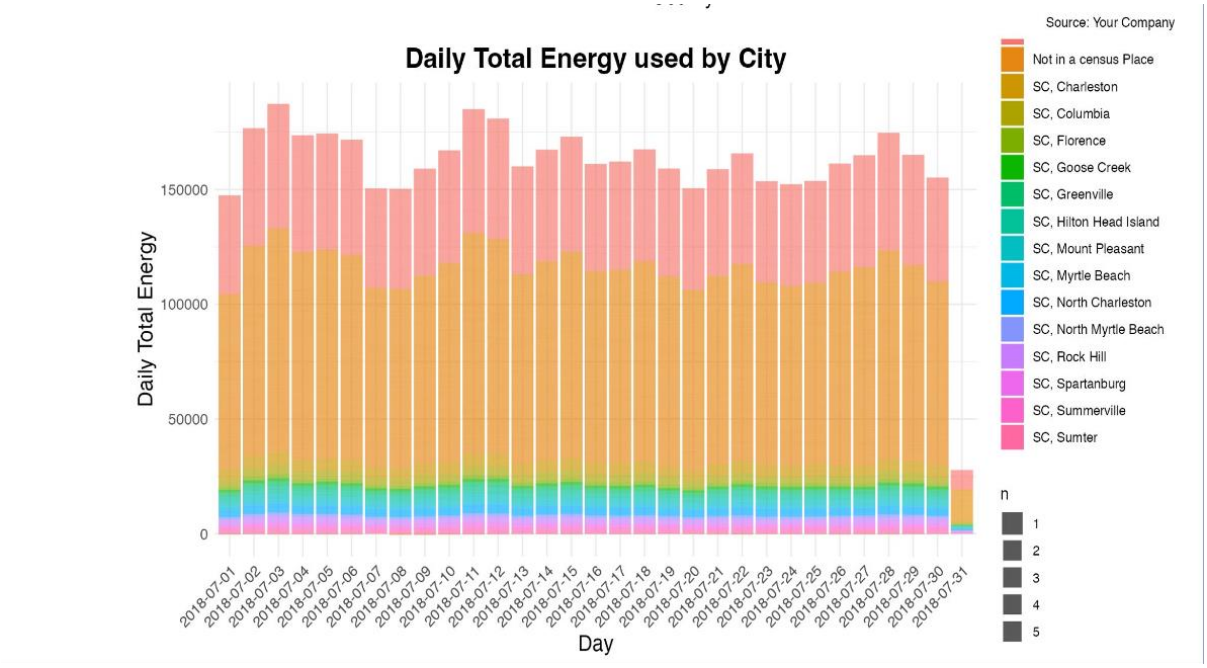
   This box plot provides a comprehensive view of the distribution of daily energy consumption across different counties during the month of July. Each box represents the energy consumption distribution within a specific county, highlighting key statistical measures such as median, quartiles, and potential outliers. The distinctive green color (#4daf4a) adds visual appeal and helps distinguish between the county distributions.



- The box plot allows for a quick comparison of energy consumption patterns among various counties.
- The central box represents the interquartile range (IQR) with the median line, while whiskers extend to show the range of the data.
- The transparency (alpha = 0.7) ensures that overlapping boxes are visually distinguishable.
- The minimal theme enhances clarity and readability of the plot, with a title and axis labels providing context.

4. Daily Total Electricity Consumption by City:

   This stacked bar plot illustrates the daily total energy usage across different cities. Each bar represents a specific date, and the segments within the bar are color-coded to denote the energy consumption distribution among various cities. The sum of the segments reflects the total energy consumption for that particular day. The vibrant colors and clear labeling enhance the visual appeal and make it easy to discern energy usage patterns.



- The stacked bars provide a visual breakdown of energy consumption, allowing for a quick comparison of contributions from different cities.
- The sum of each bar represents the total energy used on a given date, while the segments within the bar represent the proportional share of each city.
- The minimal theme ensures a clean and focused presentation, with a centered title and appropriately labeled axes for clarity.

Model 1: Linear Regression

In the first modelling approach, linear regression is employed to unravel the relationships within the dataset, specifically between the dependent variable, total energy costs, and a set of independent variables or features.

The process initiates with a division of the dataset into training and testing sets, facilitating the model's training on one subset and evaluation on another for assessing its generalisation capabilities.

The linear regression model is then trained on the training dataset, determining coefficients for each independent variable to best predict the dependent variable.

Subsequently, the model is evaluated using the testing dataset, providing insights into the significance of each variable, the magnitude of coefficients, and an overall assessment of model performance.

In this analysis, a linear regression model was constructed to predict the total energy consumption. The dataset was divided into training (80%) and testing (20%) sets to assess the model's performance.

1. LINEAR REGRESSION

```
#Building linear regression model to predict total energy
lmout <- lm(total_energy ~ ., data = train_data)

# Display summary
summary(lmout)
```

The linear regression model's summary provides valuable insights into the relationships between the predictor variables and the total energy consumption. It includes coefficients, standard errors, t-values, and p-values, offering a comprehensive overview of the model's statistical significance.

The model was utilized to predict total energy values for the test dataset. Subsequently, the predicted values were compared against the actual total energy values. The R-squared metric, a measure of the model's goodness of fit, was calculated to assess its overall accuracy.

The R-squared value, also known as the coefficient of determination, quantifies the proportion of the variance in the dependent variable (total energy) that is predictable from the independent variables. In this case, an R-squared value close to 1 indicates a high level of accuracy in predicting total energy consumption based on the selected features.

The calculated R-squared value is not only a statistical measure but also serves as a percentage, representing the accuracy of the linear regression model. A higher R-squared percentage signifies a more effective model in explaining the variability in total energy consumption.

```
# Predict total energy values for test data
predicted_values <- predict(lmout, newdata = test_data)
```

```
## Warning in predict.lm(lmout, newdata = test_data): prediction from a
## rank-deficient fit may be misleading
```

```
# These are the actual total energy values of test data
actual_values <- test_data$total_energy

# Obtain the R-Squared error.
R_squared <- 1 - (sum((actual_values - predicted_values)^2) / sum((actual_values - mean(actua
l_values))^2))

cat("R-squared (R²):", R_squared, "\n")
```

```
## R-squared (R²): 0.880862
```

```
cat("Accuracy of Linear regression model is : ", R_squared * 100, "\n")
```

```
## Accuracy of Linear regression model is :  88.0862
```

We got accuracy of 88%. The linear regression model, based on the provided dataset, demonstrates a commendable accuracy in predicting total energy consumption. The R-squared metric serves as a reliable indicator of the model's performance, making it a valuable tool for understanding and forecasting energy usage patterns.

Model 2: XG BOOST:

The second modelling approach introduces XGBoost, a powerful gradient boosting algorithm renowned for its predictive capabilities. Here, XGBoost is utilised for regression to create a model predicting total energy costs.

The process begins with data preparation, where the dataset is formatted to align with XGBoost's requirements. Numeric features are selected, and the target variable (total energy costs) is specified.

The subsequent step involves the creation of an XGBoost regression model, with the objective set to minimise the squared error, making it well-suited for regression tasks.

The model is then trained on the training dataset, learning intricate patterns and relationships between features and total energy costs.

The trained model is applied to make predictions on the testing dataset, and its performance is evaluated using the R-squared metric, offering insights into the proportion of variance explained by the independent variables.

```
library(dplyr)

# Building the model
model_xgb <- xgboost(
  data = as.matrix(train_data[, -which(names(train_data) == "total_energy")] %>% select_if(i
s.numeric)),
  label = train_data$total_energy,
  objective = "reg:squarederror",  # Use squared error for regression
  nrounds = 100  # Adjust the number of boosting rounds
)
```

The trained XGBoost model was then used to make predictions on the test set. The features of the test data were input into the model, and it generated predictions for the total energy consumption.

The R-squared metric was calculated to assess the goodness of fit of the XGBoost model. It compares the performance of the model's predictions against the actual total energy values. An R-squared value close to 1 indicates a high accuracy in predicting total energy consumption based on the selected features.

The calculated R-squared value serves as a percentage, representing the accuracy of the XGBoost model. A higher R-squared percentage signifies a more effective model in explaining the variability in total energy consumption.

```
# Make predictions on the test set
predictions <- predict(model_xgb, as.matrix(test_data[, -which(names(test_data) == "total_ene
rgy")] %>% select_if(is.numeric)))

# Obtain the R-Squared error.
rsquared <- 1 - (sum((predictions - test_data$total_energy)^2) / sum((mean(test_data$total_en
ergy) - test_data$total_energy)^2))
cat("R-squared:", rsquared, "\n")
```

```
## R-squared: 0.8404433
```

```
cat("Accuracy of XG BOOST model is : ", rsquared * 100, "\n")
```

```
## Accuracy of XG BOOST model is :  84.04433
```

The XGBoost model, with its advanced boosting technique, demonstrates a commendable accuracy in predicting total energy consumption. The R-squared metric provides insights into the model's performance, making it a valuable tool for accurate regression predictions. We have obtained accuracy of 84%.

Model 3: XG BOOST with Hyperparameters:

The third modelling iteration involves another instance of XGBoost, but with a distinct focus on hyperparameter tuning. Hyperparameters such as `max_depth`, `eta` (learning rate), `subsample`, and `colsample_bytree` are carefully adjusted to potentially enhance the model's performance.

This tuning process aims to strike an optimal balance between model complexity and generalisation. Subsequently, the XGBoost model with tuned parameters is trained using the training dataset.

Predictions are generated on the testing dataset, and the model's performance is thoroughly evaluated. This evaluation typically includes metrics like Mean Squared Error (MSE) and R-squared, providing insights into the accuracy and explanatory power of the model with the fine-tuned hyperparameters.

```
# Building model including hyperparameters like depth of tree and learning rate

xgb_model_2 <- xgboost(data = as.matrix(train_data[, -which(names(train_data) == "total_energ
y")] %>% select_if(is.numeric)), label = train_data$total_energy, nrounds = 100, max_depth =
8, eta = 0.2, subsample = 0.7, colsample_bytree = 0.7)
```

In this iteration of the XGBoost model, specific hyperparameters were tuned to potentially enhance performance. The model includes a maximum tree depth (max_depth) of 8, a learning rate (eta) of 0.2, a subsample ratio (subsample) of 0.7, and a column subsampling ratio by tree (colsample_bytree) of 0.7. These hyperparameters can significantly influence the model's predictive ability and generalization.

The trained XGBoost model, now equipped with the specified hyperparameters, was employed to predict total energy consumption on the test set. The same set of numeric features used in training the model was provided as input to generate predictions.

R-squared, a key metric for regression models, was calculated to evaluate the model's goodness of fit. It measures how well the model's predictions align with the actual total energy values in the test set. A higher R-squared value suggests a more accurate representation of total energy consumption based on the selected features.

The accuracy of the XGBoost model with hyperparameters is expressed as a percentage, indicating how well it performs in explaining the variability in total energy consumption. A higher accuracy percentage signifies a more effective model in capturing the nuances of the data.

```
# These are the actual total energy values of test data
actual_values <- test_data$total_energy

# Make predictions using the LASSO model on the test data
predictions <- predict(xgb_model_2, as.matrix(test_data[, -which(names(test_data) == "total_e
nergy")] %>% select_if(is.numeric)))

# Calculate R-squared
rsquared <- 1 - (sum((actual_values - predictions)^2) / sum((actual_values - mean(actual_valu
es))^2))

cat("R-squared:", rsquared, "\n")
```

```
## R-squared: 0.8857607
```

```
cat("Accuracy of XG BOOST with hyperparameters model is : ", rsquared * 100, "\n")
```

```
## Accuracy of XG BOOST with hyperparameters model is :  88.57607
```

By fine-tuning hyperparameters such as tree depth, learning rate, and subsampling ratios, the XGBoost model demonstrates improved accuracy in predicting total energy consumption. The R-squared metric provides a quantitative measure of the model's performance, offering valuable insights into its effectiveness with the specified hyperparameters. We obtained Accuracy of 88.58% which is the highest accuracy that we obtained.

We have used other models like Random Forests and SVM, but we found that their accuracy is not as high as Linear regression and XG Boost.

Shiny App

The shiny app is published in the url:

https://rpatchal.shinyapps.io/EnergyConsumption/

Data Loading:

The predicted_df dataset, assumed to be read from a CSV file named 'predicted_df.csv', contains information about energy consumption. Our dataset has information about daily energy consumption, but as the information is huge, we have faced server timeout issue when running the Shiny App.

In order to resolve this issue we used the data for every three days in Month of July to track the Energy Consumption.

User Interface (UI):

The Shiny UI is defined using the fluidPage function.

The UI includes a title panel, a select input for choosing a county, and a date range input for selecting a date range.

Plots generated based on user input are organized in multiple fluid rows within the main panel.

Server Logic:

Reactive expressions are used to filter the data based on user input. The filtered data is then used in various plots.

Each plot (e.g., totalEnergyPlot, newtotalEnergyPlot, etc.) is generated based on the filtered data.

The color palette used in bar plots is derived from the 'viridis' color palette, providing aesthetic visualizations.

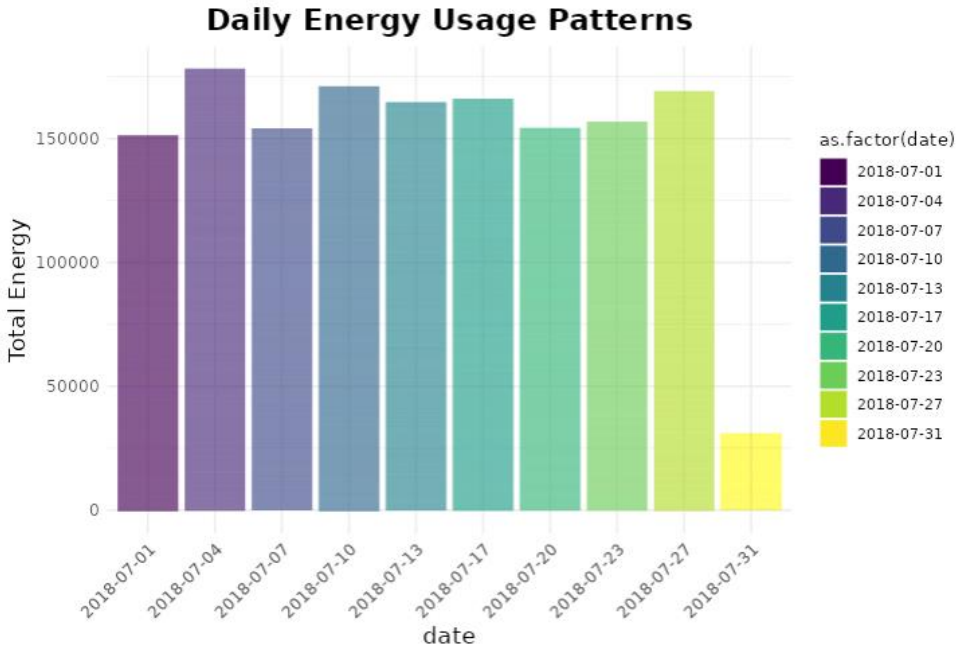
Plots:

We have displayed 4 types of plots, each showing the current energy consumption and the predicted energy consumption if temperature increases by 5 Degrees which could be the case for next year, as the Energy Company requires us to do.

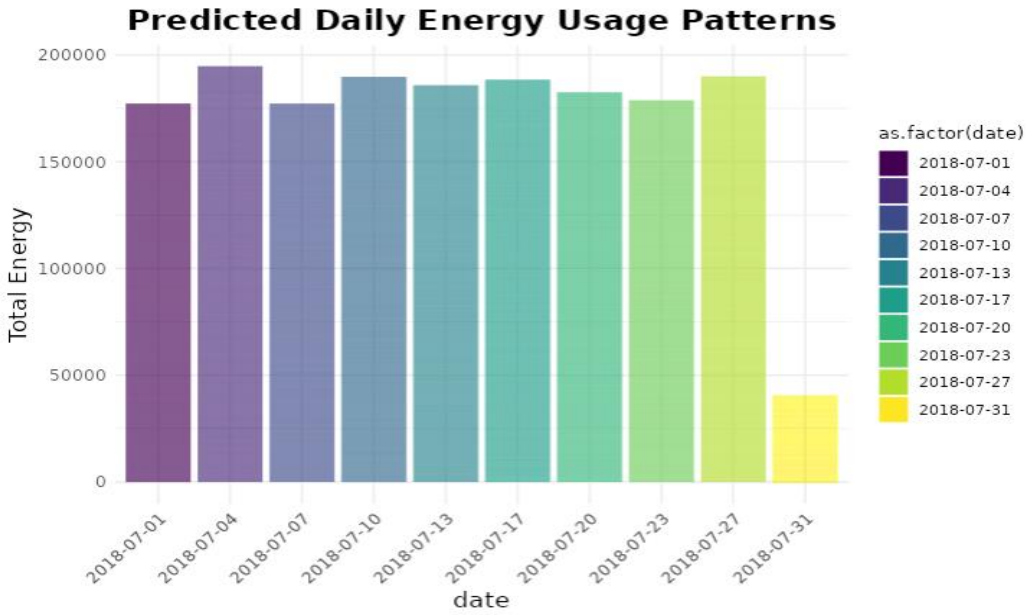
Total Energy Consumption Plot:

We have displayed two plots for total energy consumption patterns.

One of them displays daily energy usage patterns over the selected date range, with bars colored based on dates.



Other one is similar to the total energy plot, but focuses on predicted daily energy usage patterns with potential temperature increases.

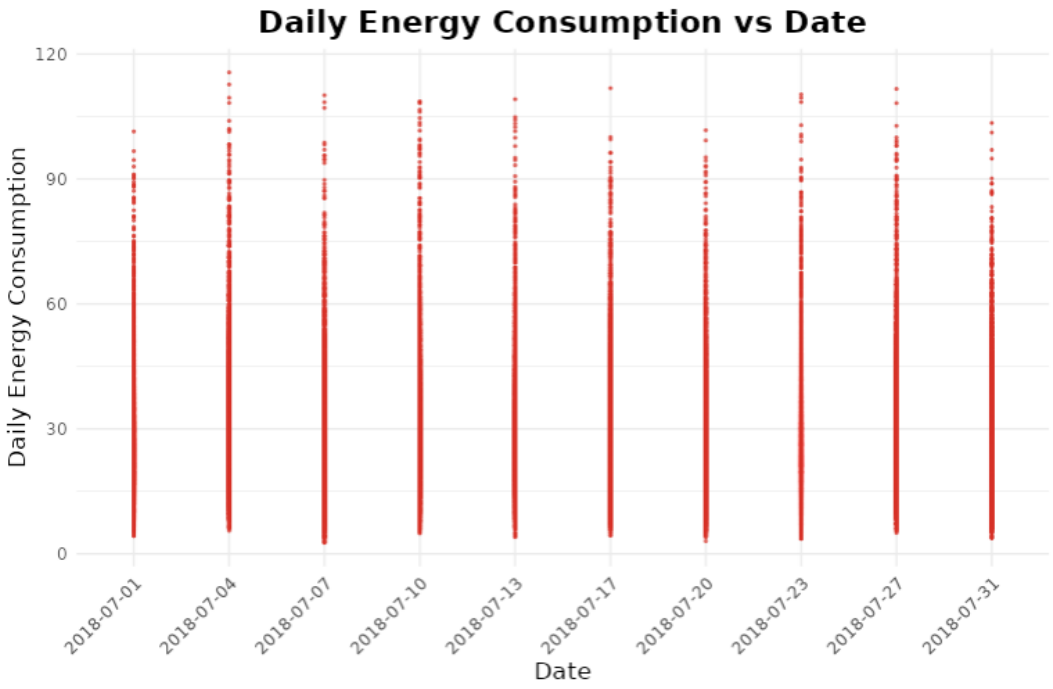


From Plots for current energy patterns and predicted energy patterns, we can say the future energy consumption is going to increase.

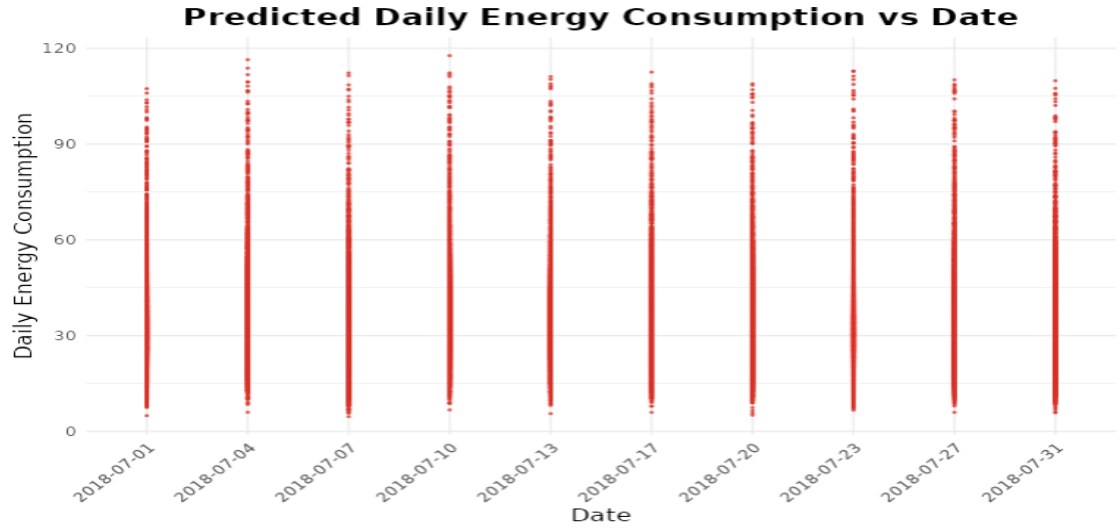
Energy Consumption by date Plot):

We have displayed two plots for total energy consumption patterns.

A scatter plot showing daily energy consumption vs. date. Each point represents the daily energy consumption value.



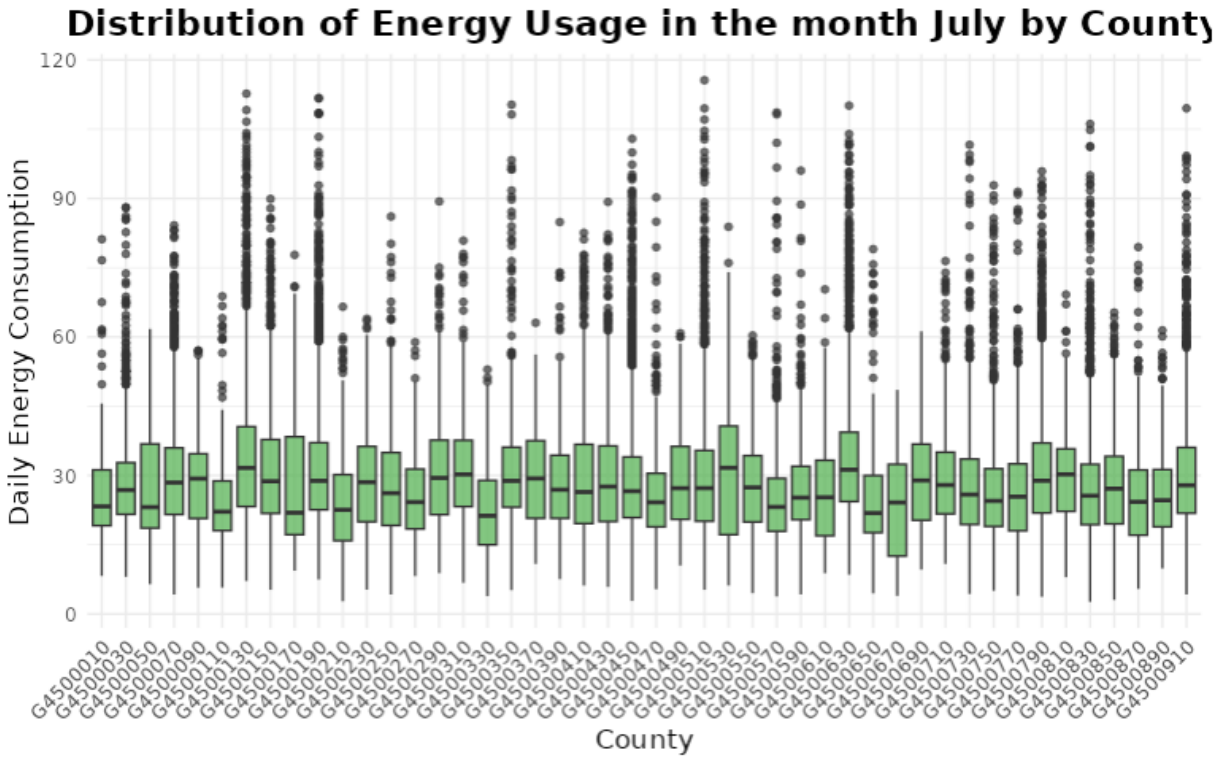
Other one is similar to the scatter plot, but focuses on predicted daily energy usage patterns with potential temperature increases.



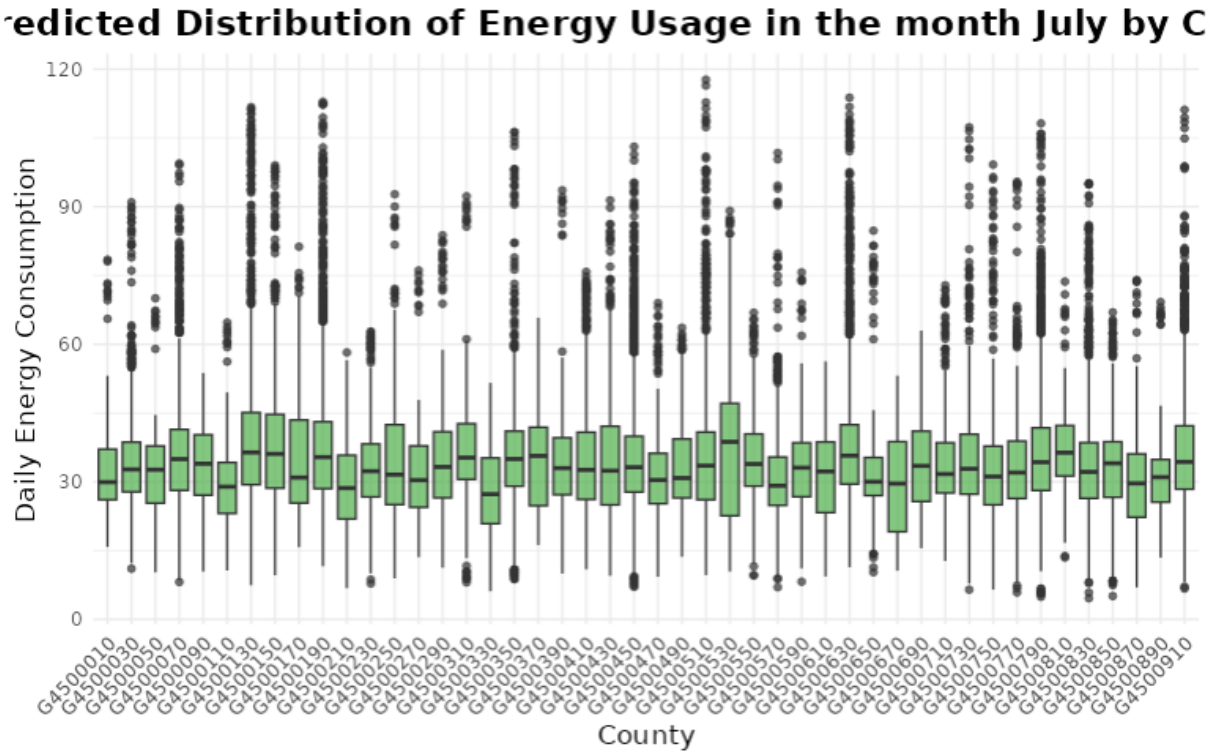
We could see that predicted energy line plots are more dense than current energy estimates, which show increase in energy consumption in future.

Energy Consumption by County Plot :

A box plot illustrating the distribution of energy usage across different counties for the selected dates.



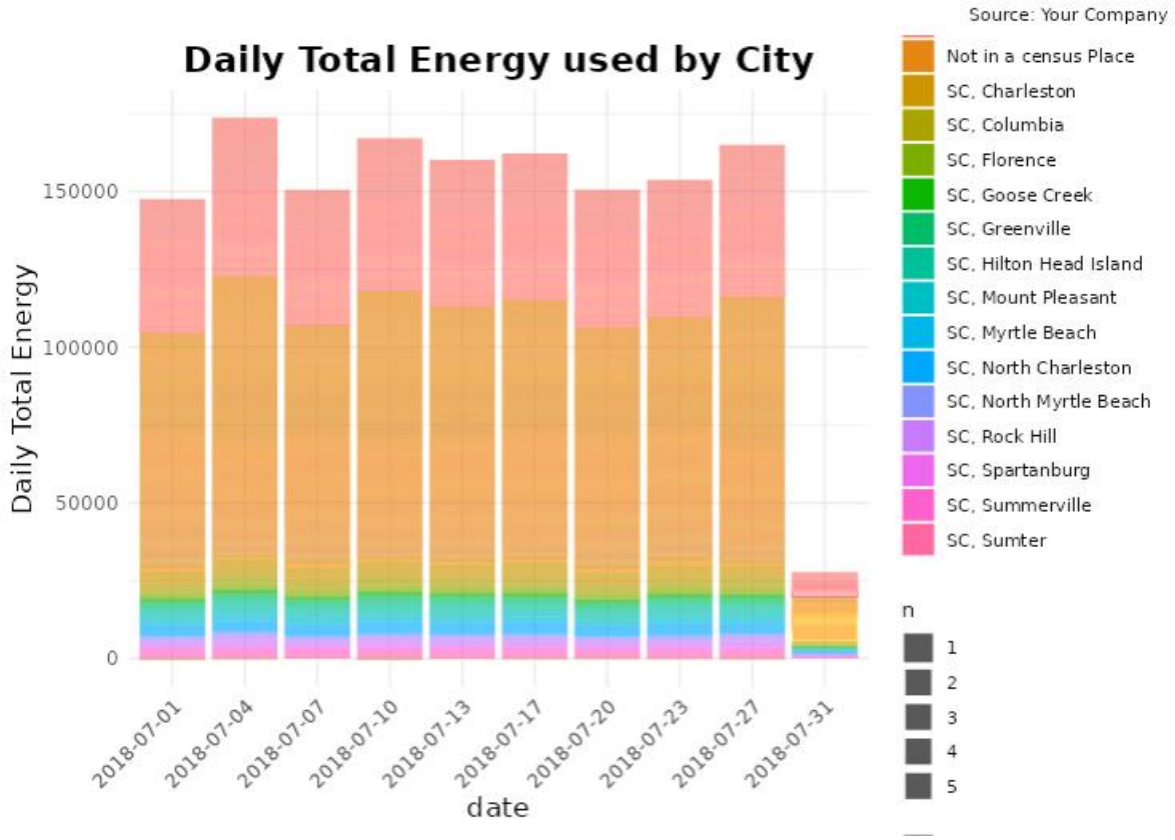
Similar to the county plot, but focuses on predicted energy consumption with potential temperature increases.



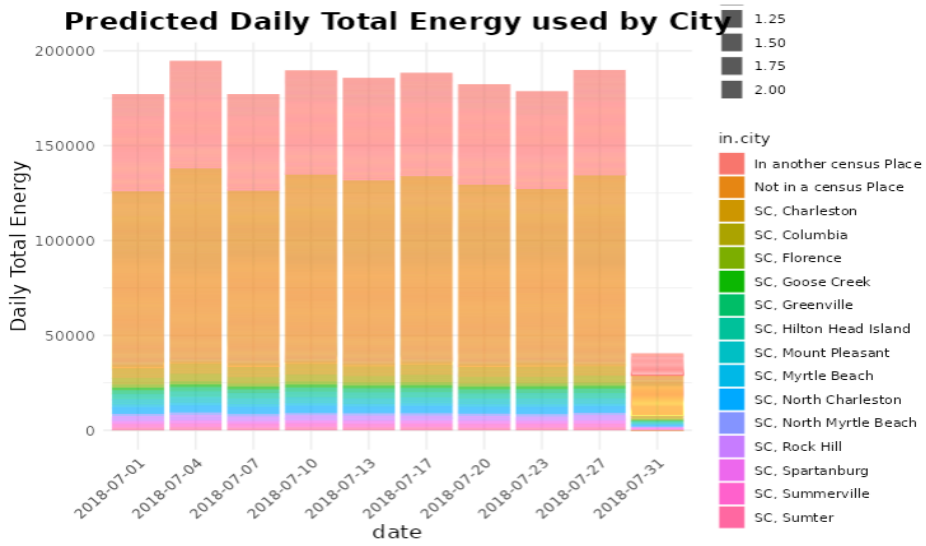
The median for the predicted energy consumption is higher than that of current energy consumption which also shows the future energy consumption increasing trend.

Daily Total Electricity Consumption by City Plot:

A stacked bar plot showcasing the daily total energy used by different cities over the selected date range.



Similar to the city plot, but focuses on predicted daily total energy consumption with potential temperature increases.



We could see that the peaks have increased for the predicted energy consumption than that of current energy consumption.

Conclusion:

The Shiny app provides an interactive platform for users to explore and visualize predicted energy consumption data.

The combination of date selection, county filtering, and various plots facilitates a comprehensive analysis of energy patterns and predictions.

When you compare the plots for current energy patterns and predicted energy patterns, we could clearly see that there are in increasing trend, and we can expect that the future energy prediction is going to be increased.

Interpretation of Results and Actionable Insights:

1. **Weather Impact on Energy Consumption:**
   The project incorporates weather data, including parameters like temperature, humidity, and wind speed. Stakeholders can leverage these insights to understand how variations in weather conditions impact energy consumption. For instance, higher temperatures might correlate with increased cooling demands.

2. **Predictive Energy Consumption Models:**
   The code implements both linear regression and XGBoost models for predicting energy consumption. Decision-makers can utilise these models to forecast energy needs accurately. This enables proactive planning and optimization of resources, ensuring efficient energy distribution and consumption.

3. **Effect of Temperature Changes:**
   The scenario analysis involving a 5-degree increase in temperature provides a forward-looking perspective. Stakeholders can glean insights into potential future energy consumption patterns under different climatic conditions. This information is valuable for adapting infrastructure and energy policies to mitigate risks associated with climate change.
4. **Data Quality Assurance:**
   The code includes steps for data cleaning and exploration, such as handling null values and generating frequency tables. This underscores the importance of ensuring data quality. Actionable insight includes a recommendation to invest in continuous data quality assurance processes to enhance the reliability of analyses and predictions.

5. **Building Orientation Considerations**:
   The final output presents a table summarising building orientations. Decision-makers can use this information to identify patterns related to building orientations and their impact on energy consumption. This insight is crucial for designing energy-efficient buildings and urban planning strategies.

6. **Evaluation of Model Performance:**
   Metrics like Mean Squared Error (MSE) and R-squared are calculated to evaluate model performance. Actionable insight involves monitoring these metrics regularly and refining models to improve predictive accuracy. This iterative process ensures that models align closely with real-world energy consumption patterns.

7. **Diversity in Data Sources:**
   The inclusion of diverse datasets, such as static house information, metadata, and weather data, highlights the importance of considering multiple data sources. Stakeholders can derive insights into how different data dimensions contribute to a holistic understanding of energy consumption, guiding future data acquisition strategies.

8. **Utilising Predicted Values:**
   Predicted energy consumption values, especially under altered conditions, offer a proactive approach to energy planning. Stakeholders can use these predictions to identify potential energy demand spikes and plan resource allocation accordingly. This foresight is valuable for optimising energy infrastructure and ensuring resilience.

   These actionable insights collectively empower decision-makers to optimise energy management strategies, enhance sustainability efforts, and make informed choices for the future

CONTRIBUTION OF TEAM MEMBERS:

1. **Ramya Chowdary Patchala:** Worked on Data Cleaning, Model Building, and future energy trends prediction.
2. **Akshay Joshi:** Worked on Data Exploration and Data Cleaning, and Merging
3. **Hemanth:** Worked on building Shiny App, and Visualizations
4. **Rithika:** Worked on Data Cleaning and making Power point Presentation.
5. **Kishan:** Worked on preparing the Project Report, and Data Analysis.

REQUIRED TASKS:

a) Determine the best approach to read and merge the data and determine what should be the output during this 'data preparation' phase : ***Done.***

b) Do exploratory analysis of the data – to gain some basic insight about the data : ***Done***

c) Build a model that predicts the energy usage, for a given hour, for the month of July.

July was selected, as eSC thought July is typically the highest energy usage month.

Hint: you will need to try several models and pick the best model.

Built and tried many models, and finalized three models which performed best.

They are Linear regression, XG Boost, XG Boost with Hyperparameter

d) Understand and be able to explain your model's accuracy.

We have obtained highest accuracy of 88.57 using XG Boost with Hyperparameters.

e) Create a new weather dataset, with all July temperatures 5 degrees warmer : ***Done***

f) Use your best model to evaluate peak future energy demand (assuming no new

customers)

a. Note: this must be model driven, not just increasing energy usage by a percentage

We have predicted the energy consumption for all July temperatures 5 degrees warmer, and saved it in new_total_energy column. We have also displayed the same in Shiny App. Through out the shiny app, we have displayed the differences between the current energy consumption and future expected energy consumption.

g) Show future peak energy demand in total (for an hour):

a. For different geographic regions

Done, we have displayed it for various counties, and cities.

b. For other dimensions /attributes you think important:

After a thorough analysis, we found that Weather features like Temperature, Humidity, Radiation, Wind , factors like appliances present in building, and geographical factors like county and city, directly effect the energy consumption.

h) Create a shiny application so that your client can interact with the data

a. To better understand your model's energy prediction***: Done***

b. To better understand the potential future energy needs: ***Done***

(and drivers of that future energy need)

i) Identify one potential approach to reduce peak energy demand

One effective approach to reduce peak energy demand involves the implementation of Demand Response (DR) programs. Demand Response is a strategy employed by energy providers to actively manage and modify energy consumption patterns among consumers.

The primary goal is to incentivize consumers to adjust their electricity usage during periods of high demand, thereby alleviating stress on the electrical grid, reducing the need for additional power generation, and minimizing the risk of blackouts.

j) What would you suggest, how would you model the impact. How would you explain the impact.

We could do proactive planning based on the predicted values and do model monitoring for the values that will be observed in future in order to keep note of any changes that may happen in future.

Implementing time-variable pricing, where electricity rates vary based on the time of day and overall demand. Encouraging consumers to shift non-essential electricity usage to off-peak hours when rates are lower.

Offering financial incentives, such as rebates or discounts, to consumers who reduce their energy consumption during peak demand periods. Providing real-time information on peak hours and potential savings.

Utilizing smart grid technology to remotely control and adjust the energy consumption of certain devices or systems in response to peak demand signals. Installing smart thermostats, smart appliances, and other IoT devices that can be programmed to operate more efficiently during peak hours.

Launching educational initiatives to raise awareness among consumers about the importance of reducing energy usage during peak times. Providing tips and guidelines on energy-efficient practices and the benefits of participating in Demand Response programs.