IST 687 - INTRODUCTION TO DATA SCIENCE

FINAL PROJECT REPORT

COOLCURRENTS: NAVIGATING JULY'S ENERGY PEAKS WITH SUSTAINABLE SOLUTIONS

Team Members
Soundarya Ravi
Archit Dilip Dukhande
Aditya Sanjay Pawar
Subhiksha Murugesan
Rithvik Rangaraj

URL: https://soundaryaravir.shinyapps.io/eSCApp

PROFESSORS
JEFFREY SALTZ
BRITTANY JOHNSON

TABLE OF CONTENTS

INTRODUCTION

OBJECTIVE

At eSC, our mission is to empower our customers with cutting-edge energy analytics for proactive consumption management. Our approach encompasses a comprehensive strategy:

Initially, we'll skillfully handle vast datasets, employing optimal techniques for data processing and integration, ensuring its preparedness for thorough analysis.

Our exploratory data analysis will unveil pivotal patterns and dependencies, identifying the primary drivers of energy consumption. Our goal is to construct advanced predictive models to accurately forecast energy usage for every hour throughout July. Rigorous evaluation will guarantee the precision and reliability of these models.

In response to concerns about rising temperatures, we'll simulate a scenario with a 5-degree Celsius increase in July temperatures. This simulation will enable us to predict peak energy demands, considering diverse geographical regions and other pertinent factors.

To enhance user experience, we're developing an intuitive Shiny application for seamless data navigation. This tool will play a crucial role in formulating a data-driven strategy to alleviate peak energy demands. We'll conduct a detailed analysis to illustrate the impact of this strategy, delivering practical recommendations that optimize energy usage, reduce costs, and align with environmental considerations.

At eSC, we're committed to providing actionable insights that enable our clients to navigate energy challenges effectively while contributing to a sustainable future.

BACKGROUND

Energy Solutions Corporation (eSC): Navigating Sustainable Energy Practices

In the realm of electricity provision to residential properties in South Carolina and a segment of North Carolina, Energy Solutions Corporation (eSC) is pioneering a strategic approach to address the challenges posed by global warming. The company's primary concern revolves around the potential impact of global warming on electricity demand and the associated threat of blackouts during peak periods.

Strategic Focus:
Instead of resorting to traditional solutions like constructing new power plants, eSC is adopting an innovative strategy. The company is committed to understanding and influencing the key dimensions of energy usage, empowering customers to embrace energy-saving practices. The overarching goal is to curtail energy consumption, especially during 'extra hot' summer periods, thereby avoiding the need for new energy production facilities. This strategy not only ensures a resilient grid but also aligns with broader environmental conservation goals.

Temporal Focus:
July emerges as the epicenter of eSC's attention, identified as the peak month for energy consumption. The company believes that understanding the temporal dimensions of energy usage during this critical period is vital for devising effective demand management strategies.

Data Dimensions:
eSC employs a robust dataset that spans multiple dimensions, providing a comprehensive view of the energy landscape:

1. Static House Data: This dataset includes 5710 observations and 171 columns, offering a detailed perspective on individual buildings. It delves into factors such as structure, layout, and other static attributes.

2. Energy Data: Comprising 44 columns and 8710 rows, this dataset provides dynamic insights into energy consumption patterns for each building ID. It's a crucial dimension for understanding real-time energy usage.

3. Weather Dataset: Tailored for each county in the service region, this dataset with 8710 rows and 11 columns offers a contextual understanding of how weather conditions impact energy demand.

4. Metadata:A comprehensive resource elucidating the columns of the datasets, aiding in the interpretation of the intricate details within the data.

Through meticulous analysis of these dimensions, eSC aims not only to comprehend the current energy scenario but also to proactively shape a sustainable future. The company's commitment is to empower customers with insights, encouraging energy-conscious choices that contribute to a resilient and environmentally friendly energy landscape.

BUSINESS QUESTIONS

INITIAL BUSINESS QUESTIONS

1. Data Accessibility and Integration:
- How can we effectively access and integrate the extensive dataset to ensure a seamless data preparation phase?
- What tools and techniques should be employed to manage and merge the diverse data sources?

2. Exploratory Data Analysis (EDA):
- What are the key variables influencing energy usage, and how can we identify them through exploratory data analysis?
- Are there any noticeable patterns or trends in the dataset that could provide valuable insights into energy consumption?

3. Model Development and Evaluation:
- What approach should be taken to build predictive models capable of accurately estimating energy usage for each hour in July?
- How will we evaluate and compare the performance of different models to select the most accurate one?

4. Weather Dataset Modification:
- How do we create a new weather dataset reflecting a 5-degree Celsius increase in temperatures for the month of July?
- What considerations are essential to ensure the modified dataset aligns with realistic scenarios?

5. Future Peak Energy Demand Projection:
- How can the selected model be utilized to project future peak energy demand for July, considering various geographic regions and relevant dimensions?
- What insights can we derive from these projections to inform decision-making?

6. Shiny Application Development:
- What features and functionalities should be incorporated into the Shiny application to facilitate user-friendly interaction with the energy data?
- How can we ensure the application meets the specific needs and preferences of our client?

7. Peak Energy Demand Reduction Strategy:

- What approach or combination of strategies should be identified to effectively reduce peak energy demand?
- How can we ensure that the selected strategy aligns with the data-driven insights derived from the model?

8. Impact Modeling and Explanation:

- How will we model the impact of the chosen strategy on reducing peak energy demand, considering different dimensions and attributes?
- What is the most effective way to communicate and explain the impact of the strategy to stakeholders in a clear and understandable manner?

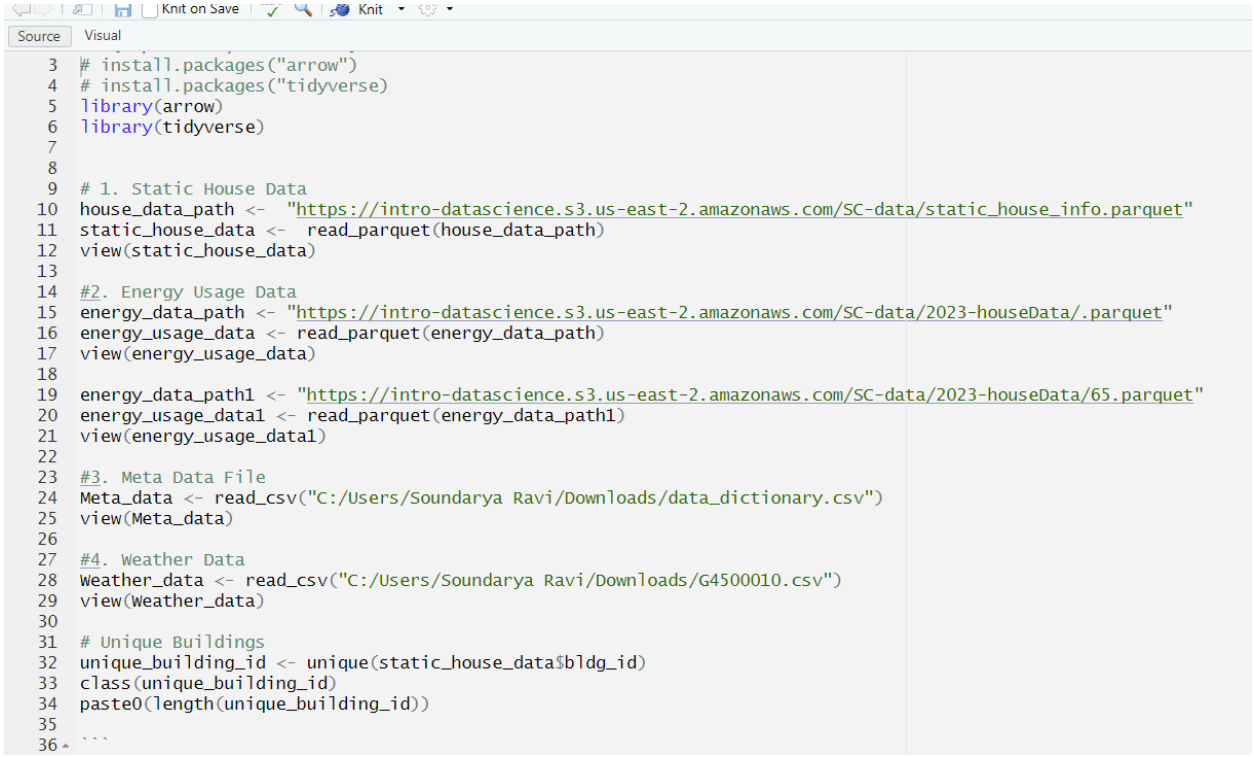
FINAL BUSINESS QUESTIONS

Data-Driven Peak Demand Reduction Approaches:

- What potential approach was identified to reduce peak energy demand, and how was this approach derived from the data?
- How would the impact of this approach be modeled, and what data-driven explanations would be provided to stakeholders regarding its effectiveness?
- What is the anticipated cost of implementing the identified peak energy demand reduction strategy?
- How does this cost compare to the expected benefits, and what is the projected return on investment?
- How sustainable is the chosen peak demand reduction strategy over the long term?
- What measures are in place to adapt the strategy to evolving energy landscapes and technological advancements?

DATA ANALYSIS

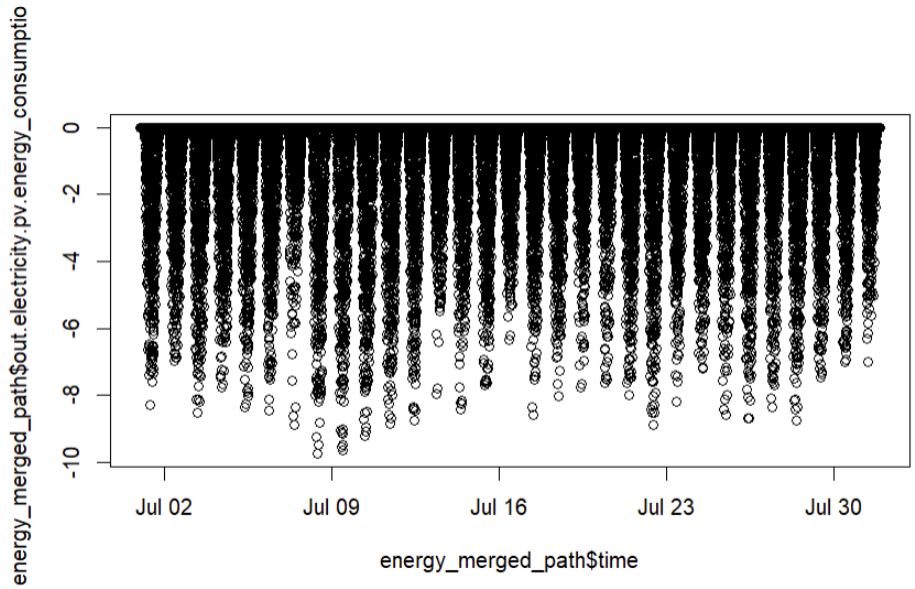
DATA ACQUISITION



```
# install.packages("arrow")
# install.packages("tidyverse")
library(arrow)
library(tidyverse)


# 1. Static House Data
house_data_path <-  "https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/static_house_info.parquet"
static_house_data <-  read_parquet(house_data_path)
view(static_house_data)

#2. Energy Usage Data
energy_data_path <- "https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/.parquet"
energy_usage_data <- read_parquet(energy_data_path)
view(energy_usage_data)

energy_data_path1 <- "https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/65.parquet"
energy_usage_data1 <- read_parquet(energy_data_path1)
view(energy_usage_data1)

#3. Meta Data File
Meta_data <- read_csv("C:/Users/Soundarya Ravi/Downloads/data_dictionary.csv")
view(Meta_data)

#4. Weather Data
Weather_data <- read_csv("C:/Users/Soundarya Ravi/Downloads/G4500010.csv")
view(Weather_data)

# Unique Buildings
unique_building_id <- unique(static_house_data$bldg_id)
class(unique_building_id)
paste0(length(unique_building_id))

```

DATA CLEANING

In the course of refining our dataset for a predictive modeling task, we conducted a comprehensive data cleansing process to enhance the quality of the information. The initial dataset presented challenges with negative values, NaN, and None entries, necessitating careful handling. We systematically addressed these issues, transforming negative values and addressing missing or undefined data points. This meticulous cleansing was imperative to ensure the integrity of subsequent analyses and modeling efforts.



Dataframe Cleaning: We replaced 'None' values to ensure data consistency and removed unnecessary columns. This step was essential to maintain data accuracy and facilitate smoother analysis.

```
library(tidyverse)
path <- '/Users/subhiksha/Downloads/Merged_House_energy_weather_for_EDA.csv'
merged_data <- read_csv(path)
colSums(is.na(merged_data))
#view(data111)

dim(merged_data)

merged_data <- subset(merged_data, select = -c(
  upgrade.insulation_wall,
  upgrade.insulation_foundation_wall,
  upgrade.hvac_heating_type,
  upgrade.insulation_ceiling,
  upgrade.hvac_cooling_efficiency,
  upgrade.infiltration_reduction,
  upgrade.geometry_foundation_type
))
is.na(merged_data)


class(merged_data$upgrade.insulation_roof)
class(merged_data$upgrade.water_heater_efficiency)
class(merged_data$upgrade.clothes_dryer)
class(merged_data$upgrade.cooking_range)


lst <- c('upgrade.insulation_roof',
         'upgrade.water_heater_efficiency',
         'upgrade.clothes_dryer',
         'upgrade.cooking_range')

for (col in lst) {
  merged_data[[col]] <- ifelse(is.na(merged_data[[col]]), "none", merged_data[[col]])
}

colSums(is.na(merged_data))
write.csv(merged_data, file = "/Users/subhiksha/Documents/IDS/ids/Merged_data.csv", row.names = TRUE)
```

DATA TRANSFORMATION

Additionally, we undertook a data aggregation approach for both columns and rows in the weather file, streamlining and consolidating information. This aggregated data was then seamlessly integrated with the main dataset, enriching it with pertinent weather-related insights. The final merged dataset, a culmination of thorough cleansing, feature engineering, and aggregation, represents a robust foundation for subsequent analyses and predictive modeling tasks.

Throughout this process, adherence to best practices in data science, including meticulous cleansing, feature selection, and model iteration, ensured the production of a high-quality dataset conducive to achieving the project's objectives. The seamless integration of diverse data sources and the application of sophisticated modeling techniques underscore our commitment to delivering insightful and actionable outcomes in line with project requirements.

Column Categorization and Aggregation: By strategically categorizing the columns in our energy dataset, we brought structure and clarity to our data. We then aggregated the values within these categories, which was pivotal in distilling the data to understand key variables more effectively.

```r
library(tidyverse)
path <- '/Users/subhiksha/Documents/IDS/ids/energydataaa_idsProj.csv'
data <- read_csv(path)

#view(data111)
head(data,100)
colnames(data)

# kitchen
data$out.kitchen.energy_consumption <- data$out.electricity.range_oven.energy_consumption +
  data$out.electricity.dishwasher.energy_consumption +
  data$out.electricity.refrigerator.energy_consumption +
  data$out.electricity.freezer.energy_consumption +
  data$out.natural_gas.range_oven.energy_consumption +
  data$out.natural_gas.grill.energy_consumption +
  data$out.propane.range_oven.energy_consumption

# laundry
data$out.laundry.energy_consumption <- data$out.electricity.clothes_dryer.energy_consumption +
  data$out.natural_gas.clothes_dryer.energy_consumption +
  data$out.electricity.clothes_washer.energy_consumption +
  data$out.propane.clothes_dryer.energy_consumption

# heating_cooling
data$out.heating_cooling.energy_consumption <- data$out.electricity.heating_fans_pumps.energy_consumption +
  data$out.electricity.heating_hp_bkup.energy_consumption +
  data$out.electricity.heating.energy_consumption +
  data$out.electricity.cooling.energy_consumption +
  data$out.natural_gas.heating_hp_bkup.energy_consumption +
  data$out.natural_gas.heating.energy_consumption +
  data$out.propane.heating_hp_bkup.energy_consumption +
  data$out.propane.heating.energy_consumption +
  data$out.fuel_oil.heating_hp_bkup.energy_consumption +
  data$out.fuel_oil.heating.energy_consumption +
  data$out.natural_gas.fireplace.energy_consumption +
  data$out.electricity.cooling_fans_pumps.energy_consumption

# water_heating
data$out.water_heating.energy_consumption <- data$out.electricity.hot_water.energy_consumption +
  data$out.fuel_oil.hot_water.energy_consumption +
  data$out.natural_gas.hot_water.energy_consumption +
  data$out.propane.hot_water.energy_consumption
```

```r
# electrical_appliances
data$out.electrical_appliances.energy_consumption <- data$out.electricity.lighting_exterior.energy_consumption +
  data$out.electricity.lighting_garage.energy_consumption +
  data$out.electricity.lighting_interior.energy_consumption +
  data$out.electricity.plug_loads.energy_consumption +
  data$out.electricity.mech_vent.energy_consumption +
  data$out.natural_gas.lighting.energy_consumption +
  data$out.electricity.ceiling_fan.energy_consumption

# outdoor_appliances
data$out.outdoor_appliances.energy_consumption <- data$out.electricity.hot_tub_heater.energy_consumption +
  data$out.electricity.hot_tub_pump.energy_consumption +
  data$out.electricity.pool_heater.energy_consumption +
  data$out.electricity.pool_pump.energy_consumption +
  data$out.natural_gas.hot_tub_heater.energy_consumption +
  data$out.natural_gas.pool_heater.energy_consumption +
  data$out.electricity.well_pump.energy_consumption

# renewable_energy
data$out.renewable_energy.energy_consumption <- data$out.electricity.pv.energy_consumption

#total
data$out.total.energy_consumption <- data$out.electricity.range_oven.energy_consumption +
  data$out.electricity.dishwasher.energy_consumption +
  data$out.electricity.refrigerator.energy_consumption +
  data$out.electricity.freezer.energy_consumption +
  data$out.natural_gas.range_oven.energy_consumption +
  data$out.natural_gas.grill.energy_consumption +
  data$out.propane.range_oven.energy_consumption +
  data$out.electricity.clothes_dryer.energy_consumption +
  data$out.natural_gas.clothes_dryer.energy_consumption +
  data$out.electricity.clothes_washer.energy_consumption +
  data$out.propane.clothes_dryer.energy_consumption +
  data$out.electricity.heating_fans_pumps.energy_consumption +
  data$out.electricity.heating_hp_bkup.energy_consumption +
  data$out.electricity.heating.energy_consumption +
  data$out.electricity.cooling.energy_consumption +
  data$out.natural_gas.heating_hp_bkup.energy_consumption +
  data$out.natural_gas.heating.energy_consumption +
  data$out.propane.heating_hp_bkup.energy_consumption +
  data$out.propane.heating.energy_consumption +
  data$out.fuel_oil.heating_hp_bkup.energy_consumption +
  data$out.fuel_oil.heating.energy_consumption +
```

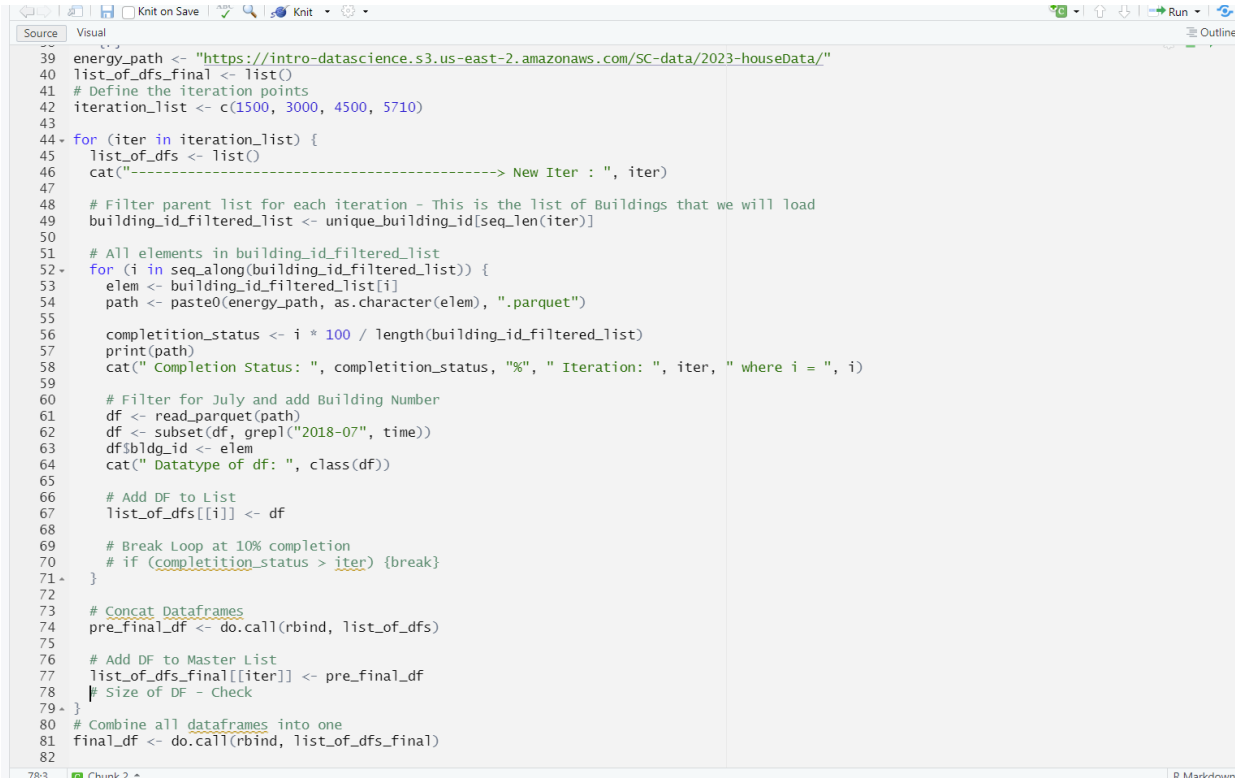
DATA MERGING

ENERGY ACQUISITION

This R function systematically integrates energy consumption data from 5710 individual building files, encompassing diverse structures. Through an iterative process, it selectively loads hourly energy records from each file, focusing on the month of July. In this monthly subset, 744 entries per file—representing each hour of the day—are retained. This meticulous filtering ensures a consolidated dataset, capturing the nuanced energy usage patterns during the peak summer month.

In tandem, the function incorporates the 'bldg_id' column from the static house information, linking each energy dataset to its respective building identifier. This enhancement contributes valuable contextual information, facilitating more detailed analysis and interpretation of energy consumption trends across different structures.

Ultimately, the function outputs a comprehensive dataframe, 'final_df,' consolidating 4.2 million rows. Each row encapsulates hourly energy consumption data for a specific building during July, providing a cohesive dataset for further exploration. This process not only harmonizes diverse datasets but also enriches them with essential metadata, enabling more nuanced insights into the intricate interplay of building characteristics and energy usage dynamics.



```r
energy_path <- "https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/"
list_of_dfs_final <- list()
# Define the iteration points
iteration_list <- c(1500, 3000, 4500, 5710)

for (iter in iteration_list) {
  list_of_dfs <- list()
  cat("-------------------------------------------------> New Iter : ", iter)

  # Filter parent list for each iteration - This is the list of Buildings that we will load
  building_id_filtered_list <- unique_building_id[seq_len(iter)]

  # All elements in building_id_filtered_list
  for (i in seq_along(building_id_filtered_list)) {
    elem <- building_id_filtered_list[i]
    path <- paste0(energy_path, as.character(elem), ".parquet")

    completition_status <- i * 100 / length(building_id_filtered_list)
    print(path)
    cat(" Completion Status: ", completition_status, "%", " Iteration: ", iter, " where i = ", i)

    # Filter for July and add Building Number
    df <- read_parquet(path)
    df <- subset(df, grepl("2018-07", time))
    df$bldg_id <- elem
    cat(" Datatype of df: ", class(df))

    # Add DF to List
    list_of_dfs[[i]] <- df

    # Break Loop at 10% completion
    # if (completition_status > iter) {break}
  }

  # Concat Dataframes
  pre_final_df <- do.call(rbind, list_of_dfs)

  # Add DF to Master List
  list_of_dfs_final[[iter]] <- pre_final_df
  # Size of DF - Check
}
# Combine all dataframes into one
final_df <- do.call(rbind, list_of_dfs_final)
```

WEATHER ACQUISITION

This code orchestrates the amalgamation of weather data for 46 distinct counties corresponding to 5710 building IDs. Beginning with an empty dataframe, `merged_df_new`, the script meticulously compiles information from individual weather files, each associated with a specific county. The counties are represented by unique codes in the `list_county`.

Within a loop, the code dynamically forms URLs to access weather data files, capturing essential variables like dry bulb temperature, wind speed, relative humidity, radiation, and wind directions. Employing a `tryCatch` mechanism, the script gracefully handles potential errors during file reading, ensuring the process's robustness.

Successfully read data for each county is then appended to the master dataframe, `merged_df_new`, accompanied by a new column, 'county_id,' denoting the relevant county code. This systematic integration ensures the comprehensive consolidation of weather information from diverse regions.

The resulting data frame encapsulates a holistic overview of weather conditions across all counties, offering a foundational dataset for subsequent analyses. The output provides a sneak peek into the merged data, showcasing the initial rows and highlighting the successful combination of weather attributes for informed insights into the relationship between building energy consumption and environmental factors.

```r
```{r}
# Create an empty data frame to store the merged data
merged_df_new<- data.frame()

# List of county codes
list_county <- c("G4500910", "G4500730", "G4500710", "G4500790", "G4500450", "G4500150", "G4500350", "G4500190", "G4500830",
  "G4500510", "G4500070", "G4500670", "G4500750", "G4500290", "G4500490", "G4500130", "G4500630", "G4500870",
  "G4500550", "G4500010", "G4500430", "G4500890", "G4500850", "G4500770", "G4500030", "G4500590", "G4500610",
  "G4500250", "G4500530", "G4500210", "G4500410", "G4500570", "G4500690", "G4500310", "G4500090", "G4500470",
  "G4500050", "G4500330", "G4500650", "G4500230", "G4500270", "G4500370", "G4500110", "G4500170", "G4500390",
  "G4500810")

# Iterate through each county code
for (county in list_county) {
  url <- paste0('https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/weather/2023-weather-data/', county, '.csv')

  # Use tryCatch to handle errors
  tryCatch({
    df_county <- read.csv(url)
    # Add a new column 'county_id' with the current county code
    df_county$county_id <- county
    merged_df_new <- rbind(merged_df_new, df_county)
  }, error = function(e) {
    cat("Error reading file for county", county, ":", conditionMessage(e), "\n")
  })
}

# Print the first few rows of the merged data frame
print(head(merged_df_new))



```
```

```r
# Assuming merged_df has a column named date_time

# Load the dplyr package
library(dplyr)

# Filter merged_df for the month of July
merged_df_new<- merged_df_new %>%
  filter(format(as.Date(date_time), "%Y-%m") == "2018-07")

# Print the first few rows of the filtered data frame
print(head(merged_df_new))
nrow(merged_df_new)
```

```r
setwd("C:/Users/Soundarya Ravi/Desktop/Shiny/")

write.csv(final_df, file ="energy_final.csv",row.names = FALSE)
```

MAPPING WITH THE STATIC HOUSE DATA

Before merging the combined energy and weather data to the static house data, it has been munged thoroughly so that the rbind function would not take more time to map it all together.

The weather data which had 34224 observations has been munged below, the time frames of the days are carefully chosen after reviewing which time frame of the day for each 31 days in the month has a similar value to the grouped average for the total energy consumption. The same way the weather file is worked through for each 46 county IDs.

time	bldg_id	sum
7/1/2018	385929	0.097
7/1/2018 1:00	385929	0.089
7/1/2018 2:00	385929	0.088
7/1/2018 3:00	385929	0.088
7/1/2018 4:00	385929	0.548
7/1/2018 5:00	385929	0.59
7/1/2018 6:00	385929	0.486
7/1/2018 7:00	385929	0.661
7/1/2018 8:00	385929	0.398
7/1/2018 9:00	385929	0.528
7/1/2018 10:00	385929	0.6
7/1/2018 11:00	385929	0.66
7/1/2018 12:00	385929	0.708
7/1/2018 13:00	385929	0.756
7/1/2018 14:00	385929	0.767
7/1/2018 15:00	385929	0.747
7/1/2018 16:00	385929	0.827
7/1/2018 17:00	385929	0.802
7/1/2018 18:00	385929	0.747
7/1/2018 19:00	385929	0.259
7/1/2018 20:00	385929	0.108
7/1/2018 21:00	385929	0.105
7/1/2018 22:00	385929	0.098
7/1/2018 23:00	385929	0.096

```
str(weather_data_merged_with_county)

```
```{r}
#Grouping the time frame of Weather File
#str(weather_data_merged)
# Assuming 'weather_data_merged' contains data for multiple counties

# Assuming 'weather_data_merged' contains data for multiple counties

result_df <- weather_data_merged_with_county %>%
  mutate(hour = hour(strptime(date_time, format = "%Y-%m-%d %H:%M:%S")),
         time_split = case_when(
           hour %in% c(0, 1, 2, 3) ~ "Late Night",
           hour %in% c(4, 5, 6, 7, 8) ~ "Early Morning",
           hour %in% c(9, 10, 11, 12) ~ "Morning",
           hour %in% c(13, 14, 15) ~ "Noon",
           hour %in% c(16, 17, 18) ~ "Evening",
           hour %in% c(19, 20, 21, 22, 23) ~ "Night",
           TRUE ~ "Other"
         )) %>%
  group_by(county_id, time_split) %>%
  summarise(
    Dry_Bulb_Temperature_C = mean(Dry.Bulb.Temperature...C.),
    Relative_Humidity = mean(Relative.Humidity....),
    Wind_Speed_m_s = mean(Wind.Speed..m.s.),
    Wind_Direction_Deg = mean(Wind.Direction..Deg.),
    Global_Horizontal_Radiation_W_m2 = mean(Global.Horizontal.Radiation..W.m2.),
    Direct_Normal_Radiation_W_m2 = mean(Direct.Normal.Radiation..W.m2.),
    Diffuse_Horizontal_Radiation_W_m2 = mean(Diffuse.Horizontal.Radiation..W.m2.)
  ) %>%
  ungroup() %>%
  mutate(
    time_range = case_when(
      time_split == "Late Night" ~ "00:00:00 to 03:00:00",
      time_split == "Early Morning" ~ "04:00:00 to 08:00:00",  # Adjust as needed
      time_split == "Morning" ~ "09:00:00 to 12:00:00",         # Adjust as needed
      time_split == "Noon" ~ "13:00:00 to 15:00:00",            # Adjust as needed
      time_split == "Evening" ~ "16:00:00 to 18:00:00",         # Adjust as needed
      time_split == "Night" ~ "19:00:00 to 23:00:00",           # Adjust as needed
      TRUE ~ "Other"
    )
  ) %>%
  arrange(county_id, time_range)
```

Similarly, the energy file from the raw energy is worked through as,

The columns has been aggregated before the rows are worked through, the columns are grouped by the way it made sense ton us in distinguishable types,

```
{r}
library(tidyverse)
#path <- "C:/Users/Soundarya Ravi/Desktop/Shiny/Raw Files/rawenergyfinal.csv"
data <- raw_energy_merged

#view(data111)
head(data,100)
colnames(data)

# Kitchen
# out.electricity.range_oven.energy_consumption
# out.electricity.dishwasher.energy_consumption
# out.electricity.refrigerator.energy_consumption
# out.electricity.freezer.energy_consumption
# out.natural_gas.range_oven.energy_consumption
# out.natural_gas.grill.energy_consumption
# out.propane.range_oven.energy_consumption
#
# laundry
# out.electricity.clothes_dryer.energy_consumption
# out.natural_gas.clothes_dryer.energy_consumption
# out.electricity.clothes_washer.energy_consumption
# out.propane.clothes_dryer.energy_consumption
#
# heating_cooling
# out.electricity.heating_fans_pumps.energy_consumption
# out.electricity.heating_hp_bkup.energy_consumption
# out.electricity.heating.energy_consumption
# out.electricity.cooling.energy_consumption
# out.natural_gas.heating_hp_bkup.energy_consumption
# out.natural_gas.heating.energy_consumption
# out.propane.heating_hp_bkup.energy_consumption
# out.propane.heating.energy_consumption
# out.fuel_oil.heating_hp_bkup.energy_consumption
# out.fuel_oil.heating.energy_consumption
# out.natural_gas.fireplace.energy_consumption
# out.electricity.cooling_fans_pumps.energy_consumption
#
# water_heating
# out.electricity.hot_water.energy_consumption
# out.fuel_oil.hot_water.energy_consumption
# out.natural_gas.hot_water.energy_consumption
# out.propane.hot_water.energy_consumption
#
# electrical_appliances
```

Chunk 17

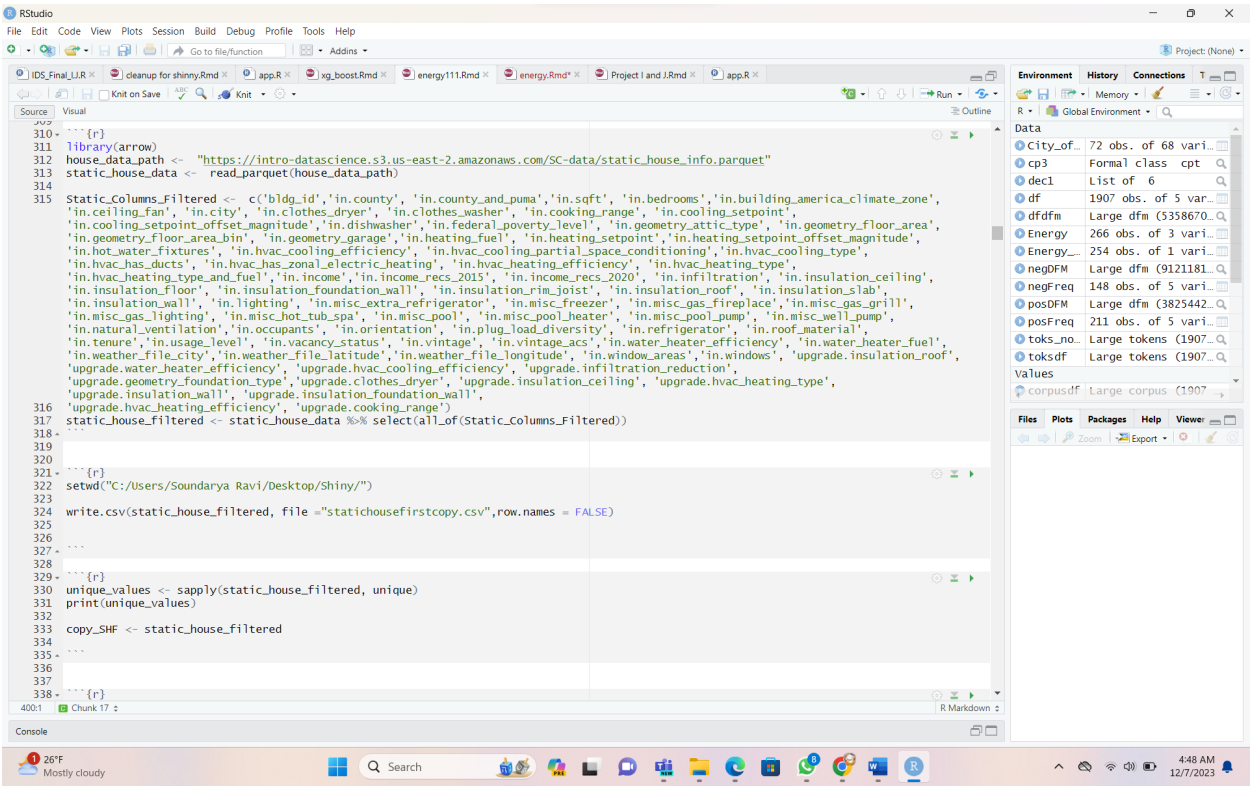
Then Row is been Aggregated as,

```r
```{r}
library(dplyr)
library(tibble)

process_energy_data <- function(energy_data) {
  result_df <- energy_data %>%
    mutate(hour = hour(time),
           time_split = case_when(
             hour %in% c(0, 1, 2, 3) ~ "Late Night",
             hour %in% c(4, 5, 6, 7, 8) ~ "Early Morning",
             hour %in% c(9, 10, 11, 12) ~ "Morning",
             hour %in% c(13, 14, 15) ~ "Noon",
             hour %in% c(16, 17, 18) ~ "Evening",
             hour %in% c(19, 20, 21, 22, 23) ~ "Night",
             TRUE ~ "Other"
           )) %>%
    group_by(bldg_id, time_split) %>%
    summarise(
      out.kitchen_energy_consumption = mean(out.kitchen.energy_consumption),
      out.laundry_energy_consumption = mean(out.laundry.energy_consumption),
      out.heating_cooling_energy_consumption = mean(out.heating_cooling.energy_consumption),
      out.water_heating_energy_consumption = mean(out.water_heating.energy_consumption),
      out.electrical_appliances_energy_consumption = mean(out.electrical_appliances.energy_consumption),
      out.outdoor_appliances_energy_consumption = mean(out.outdoor_appliances.energy_consumption),
      out.renewable_energy_energy_consumption = mean(out.renewable_energy.energy_consumption),
      out.total_energy_consumption = mean(out.total.energy_consumption)
    ) %>%
    ungroup() %>%
    mutate(
      time_range = case_when(
        time_split == "Late Night" ~ "00:00:00 to 03:00:00",
        time_split == "Early Morning" ~ "04:00:00 to 08:00:00",
        time_split == "Morning" ~ "09:00:00 to 12:00:00",
        time_split == "Noon" ~ "13:00:00 to 15:00:00",
        time_split == "Evening" ~ "16:00:00 to 18:00:00",
        time_split == "Night" ~ "19:00:00 to 23:00:00",
        TRUE ~ "Other"
      )
    ) %>%
    arrange(bldg_id, time_range)

  return(result_df)
}
```

Ultimately, after aggregating and retrieving energy and weather data using the `write.csv` function, a refinement process was applied to the static house data file. Out of the 171 columns originally present, a judicious selection was made, resulting in the exclusion of more than 50% of them. Our criteria for retention were guided by factors we deemed crucial for model predictions and meaningful visualizations. The chosen columns were those we identified as key contributors to the relevance and effectiveness of the model, streamlining the dataset to enhance its predictive capacity and visualization clarity.
```



Finally the three files has been combined further down, the house and energy has been combined through the common vector bldg_id, the weather has been added through the in.county.

```r
#Model_Merge_SH <- static_house_ordinal_clear
# Assuming Model_Merge_SH and energy_result_df are your data frames
# and both have a column named 'bldg_id'

# Merge the data frames on the 'bldg_id' column
SH_Energy_Model_Ordinal <- merge(Model_Merge_SH, energy_result_df, by='bldg_id')

# Print or view the merged data frame
print(SH_Energy_Model_Ordinal)
```

```r
# Assuming SH_Energy_Model_Ordinal and weather_combined are your data frames
# and they have columns 'in.county' and 'county_id' respectively

# Merge the data frames on the specified columns
Final_Merged_Ordinal_Model<-  merge(SH_Energy_Model_Ordinal, weather_combined, by.x=c('in.county', 'time_split'), by.y=c('county_id',
'time_split'))


# Print or view the merged data frame
print(Final_Merged_Ordinal_Model)
```

```r
# Assuming Final_Merged_Ordinal_Model is your merged data frame
# Sort the data frame based on 'bldg_id' and 'time_range.x'
Final_team2_Modelling_SEW<- Final_Merged_Ordinal_Model[order(Final_Merged_Ordinal_Model$bldg_id, Final_Merged_Ordinal_Model$time_range.x), ]

# Print or view the sorted data frame
print(Final_team2_Modelling_SEW)
```

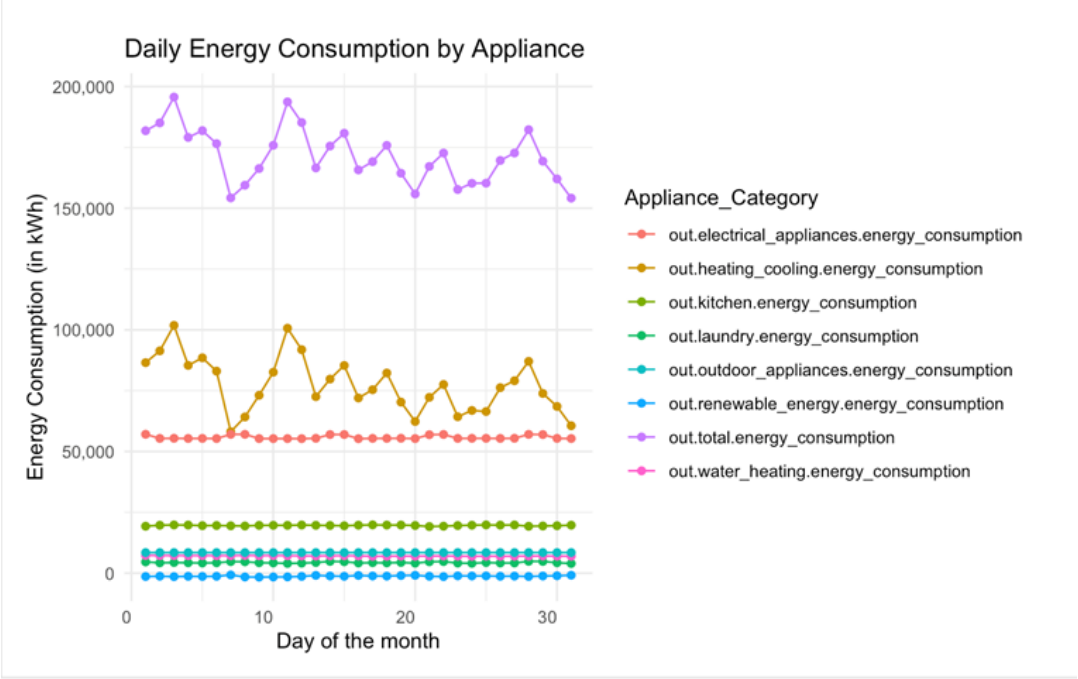
It has been combined and sorted based on the building ID and the timeframe.

EXPLORATORY DATA ANALYSIS

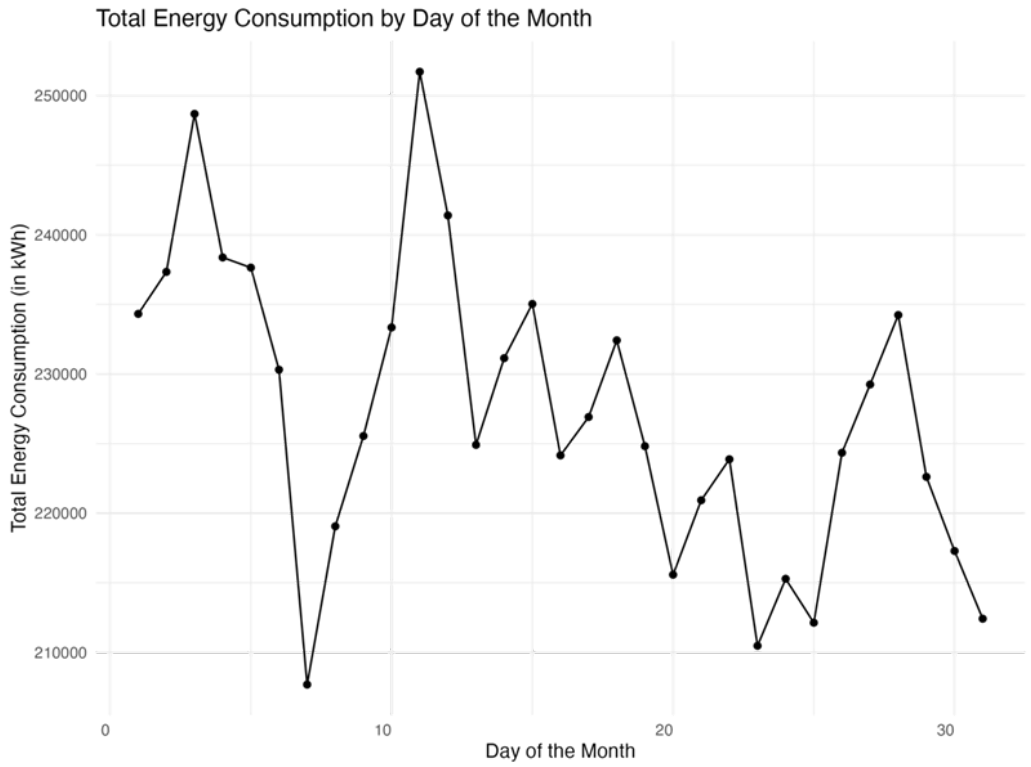
DESCRIPTIVE STATISTICS AND VISUALIZATIONS

A comprehensive Exploratory Data Analysis (EDA) was performed to delve into the dataset's underlying structure and trends. This phase was pivotal in identifying key patterns, understanding energy usage behaviors, and spotting anomalies. Based on these insights, the team developed a series of visualizations:

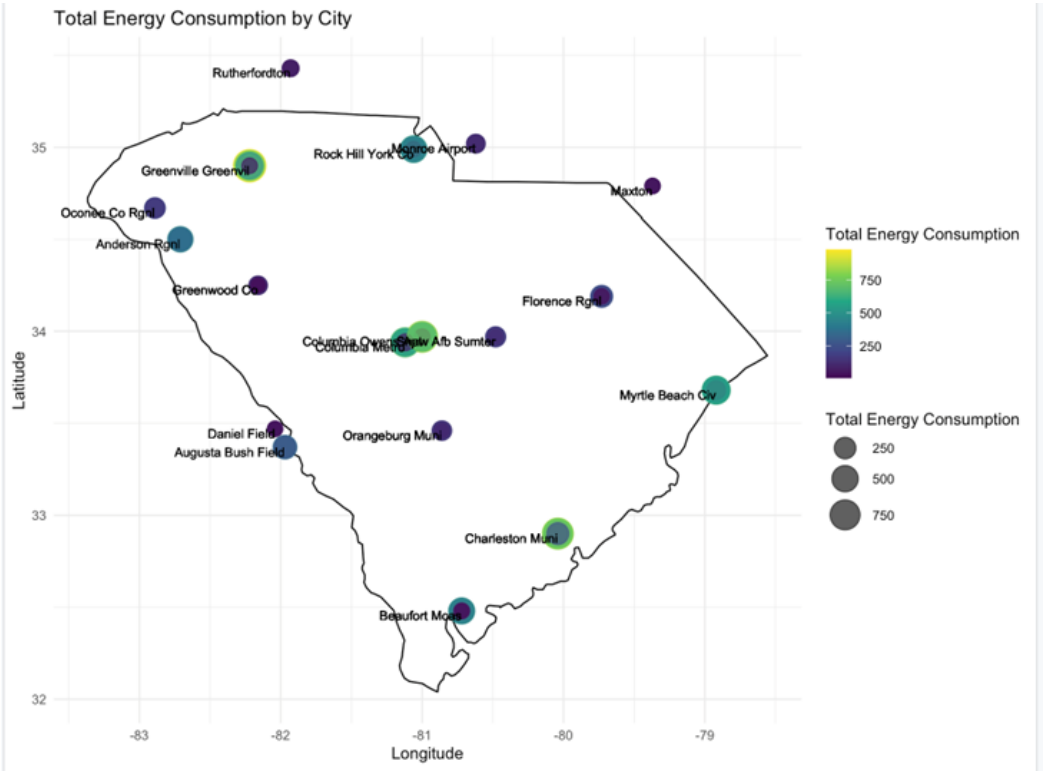
- Daily energy consumption by appliance category in South Carolina households, showcased through a detailed line plot.



- A line plot illustrating the total energy consumption for each day in July, providing an overview of the monthly energy usage trend.



A map visualization depicting the total energy consumption across different cities in South Carolina, highlighting geographical variations in energy usage



SUMMARY

The Summary of the final merged file is obtained to take a look at descriptive statistics,

```
   bldg_id       time_split_numeric  in.county
Min.   :    65   Min.   :1.0        Length:34260
1st Qu.:137588   1st Qu.:2.0        Class :character
Median :277502   Median :3.5        Mode  :character
Mean   :276418   Mean   :3.5
3rd Qu.:411188   3rd Qu.:5.0
Max.   :549916   Max.   :6.0

 time_split        in.county_and_puma    in.sqft
Length:34260       Length:34260       Min.   : 328
Class :character   Class :character   1st Qu.:1220
Mode  :character   Mode  :character   Median :1690
                                      Mean   :2114
                                      3rd Qu.:2176
                                      Max.   :8194

 in.bedrooms   in.building_america_climate_zone
Min.   :1.00   Length:34260
1st Qu.:3.00   Class :character
Median :3.00   Mode  :character
Mean   :3.26
3rd Qu.:4.00
Max.   :5.00

in.ceiling_fan       in.city          in.clothes_dryer
Length:34260       Length:34260       Length:34260
Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character



in.clothes_washer  in.cooking_range   in.cooling_setpoint
Length:34260       Length:34260       Min.   :60
Class :character   Class :character   1st Qu.:70
Mode  :character   Mode  :character   Median :72
                                      Mean   :73
                                      3rd Qu.:76
                                      Max.   :80

in.cooling_setpoint_offset_magnitude in.dishwasher
Length:34260                         Length:34260
Class :character                     Class :character
Mode  :character                     Mode  :character



in.federal_poverty_level in.geometry_attic_type
Length:34260             Length:34260
Class :character         Class :character
Mode  :character         Mode  :character
```

...

Head(Data)

```
head(data)
```

bldg_id <dbl>	time_split_numeric <dbl>	in.county <chr>	time_split <chr>	in.county_and_puma <chr>	in.sqft <dbl>	in
65	1	G4500910	Late Night	G4500910, G45000502	885	
65	2	G4500910	Early Morning	G4500910, G45000502	885	
65	3	G4500910	Morning	G4500910, G45000502	885	
65	4	G4500910	Noon	G4500910, G45000502	885	
65	5	G4500910	Evening	G4500910, G45000502	885	
65	6	G4500910	Night	G4500910, G45000502	885	

6 rows | 1-7 of 96 columns

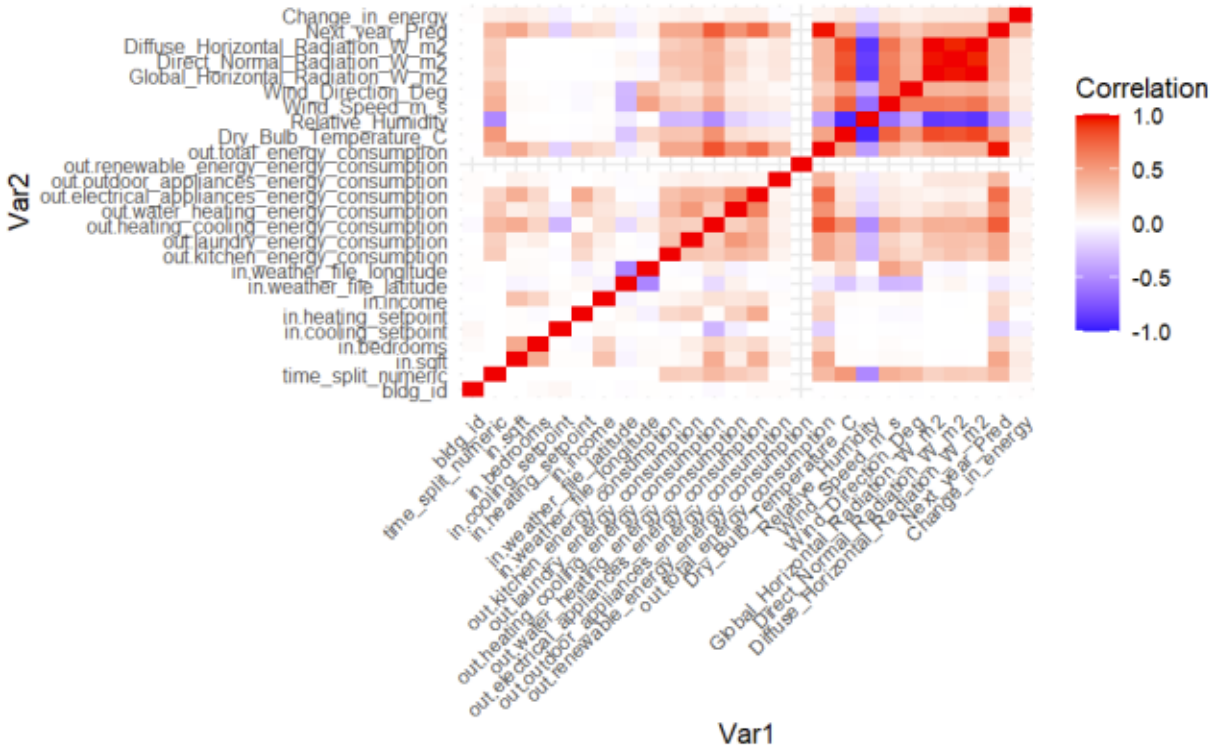
Hide

#The data was in a 24 hours timeline, we have divided the data into 5 parts. Namely: #Early Morning, Morning, Noon, Night and Late Night. #Further we will be analyzing the electricity consumption of each section. #First we output the correlation matrix of only the numeric values. #Then we calculate the correlation matrix followed by the melt correlation matrix. #The melted correlation matrix is created using the melt function, #which transforms the correlation matrix into a long format. #Then we display the heatmap plot. #If we observe the image after saving it on our device we can see that the red zones are highly correlated. #We can nitpick the columns from the heat map but it is not the most efficient way of going about when performing EDA.

CORRELATION HEAT MAP

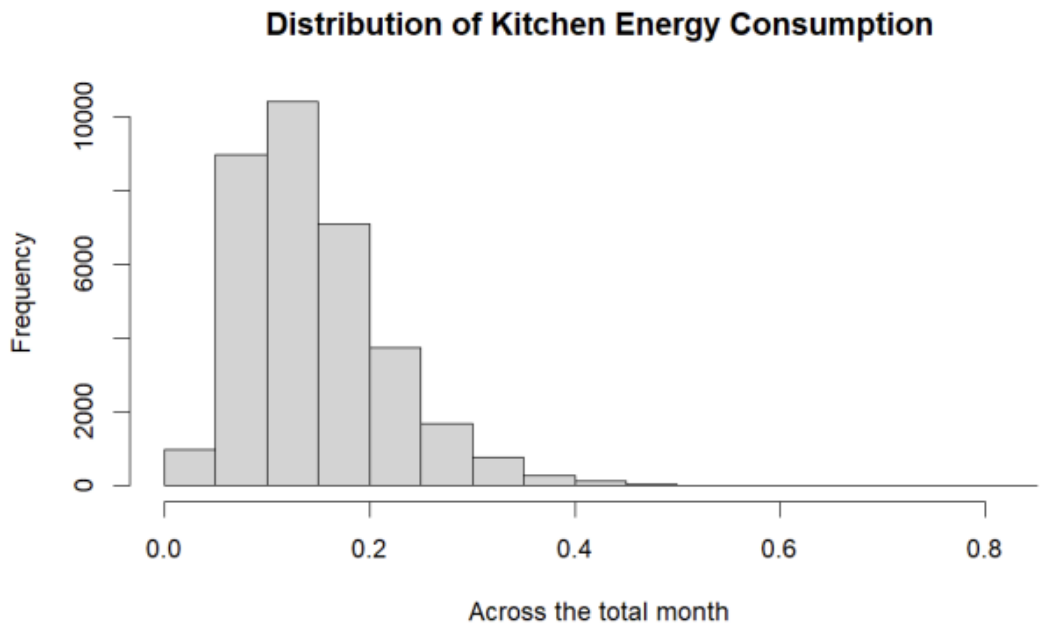
```
library(reshape2)
library(ggplot2)
numeric_data <- data[sapply(data, is.numeric)]
cor_matrix <- cor(numeric_data, use = "complete.obs")
```

```
melted_cor_matrix <- melt(cor_matrix, na.rm = TRUE)
heatmap_plot <- ggplot(data = melted_cor_matrix,
                       aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 8, hjust = 1),
        axis.text.y = element_text(size = 8))
print(heatmap_plot)
```

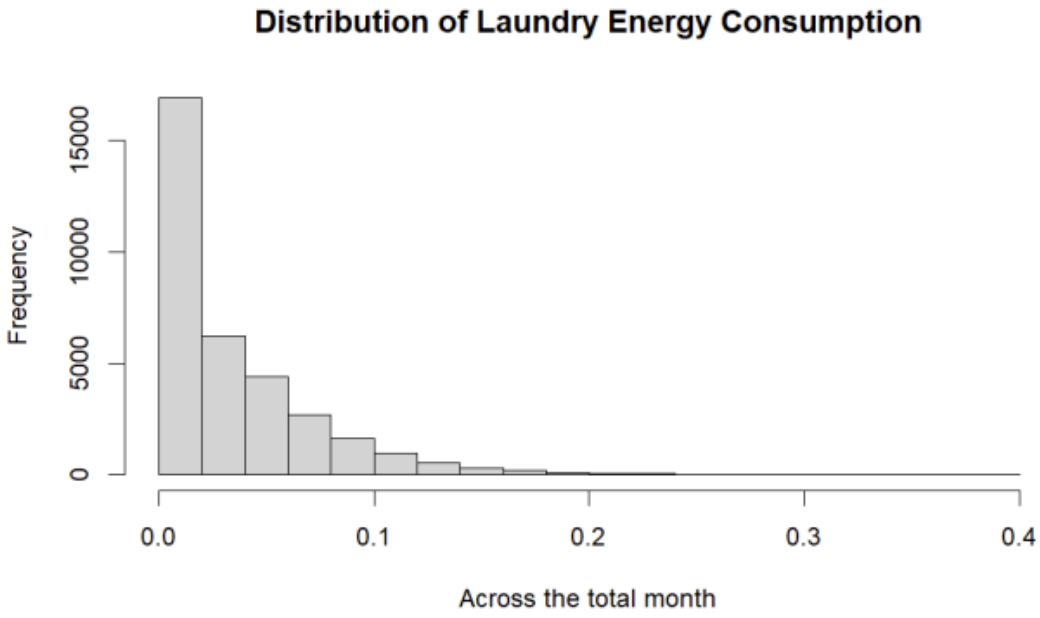


HISTOGRAM VISUALIZATION

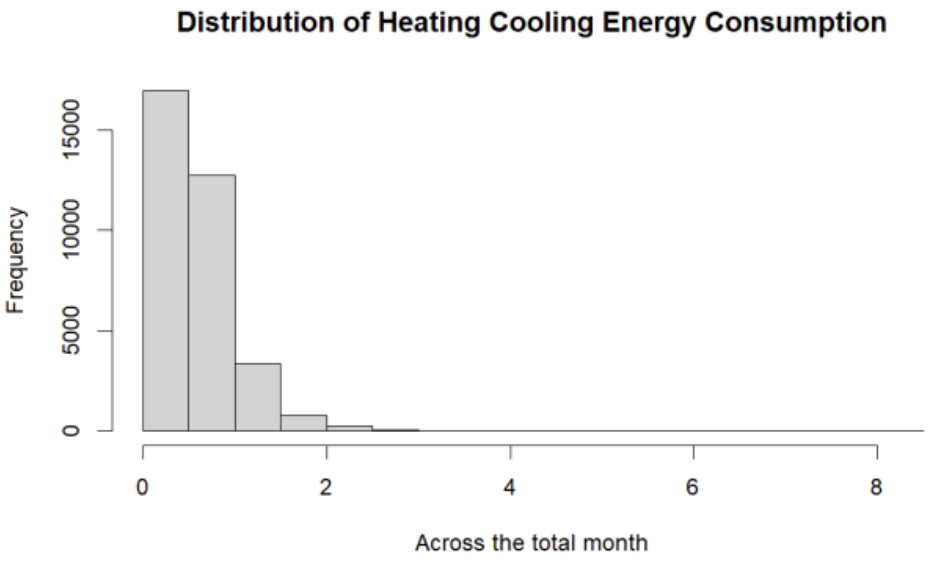
```
#Visualize the distribution of a few particular numeric variable like to get a better underst
anding of the column values.
#In this case we are looking at the #toal_energy_consumption.It is left skewed as most of the
phenomenons in the world.


hist(data$out.kitchen_energy_consumption, main="Distribution of Kitchen Energy Consumption",
xlab="Across the total month")
```



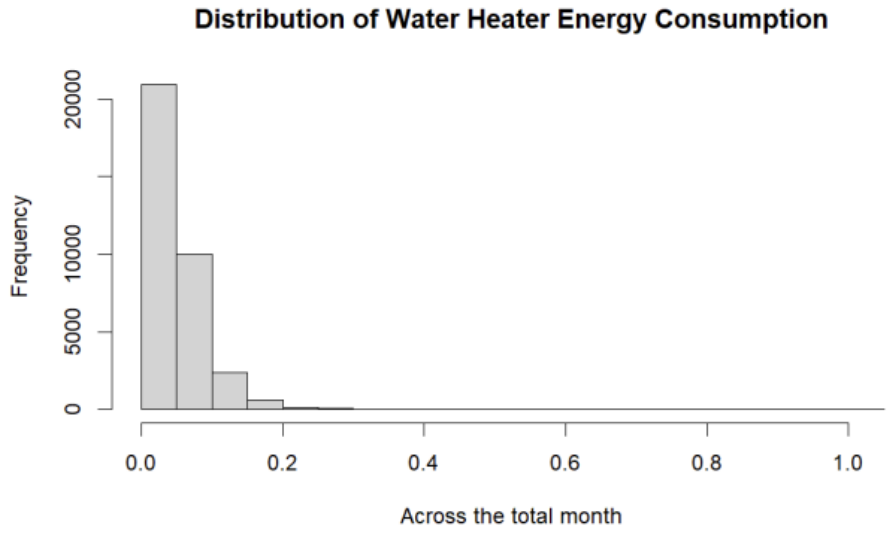

```
hist(data$out.laundry_energy_consumption, main="Distribution of Laundry Energy Consumption",
xlab="Across the total month")
```

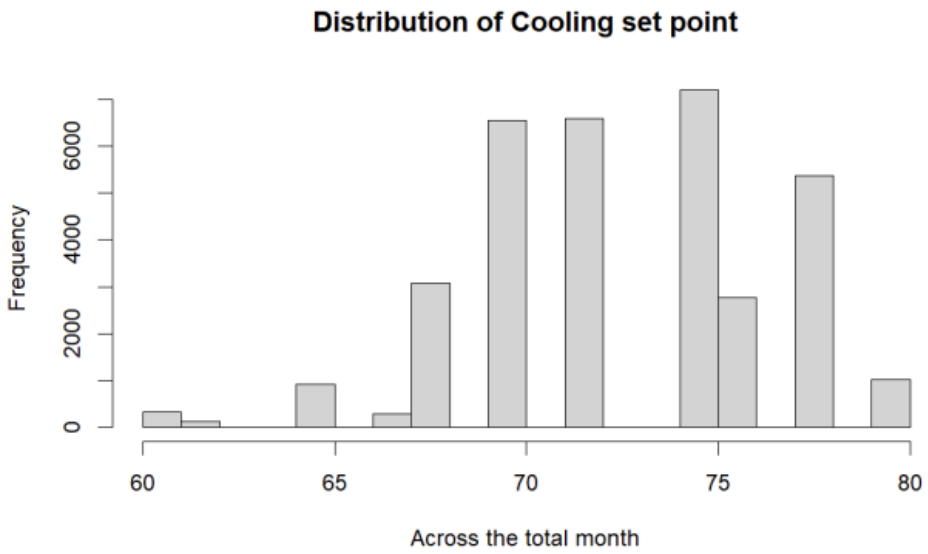


```
hist(data$out.heating_cooling_energy_consumption, main="Distribution of Heating Cooling Energ
y Consumption", xlab="Across the total month")
```




```
hist(data$out.water_heating_energy_consumption, main="Distribution of Water Heater Energy Con
sumption", xlab="Across the total month")
```

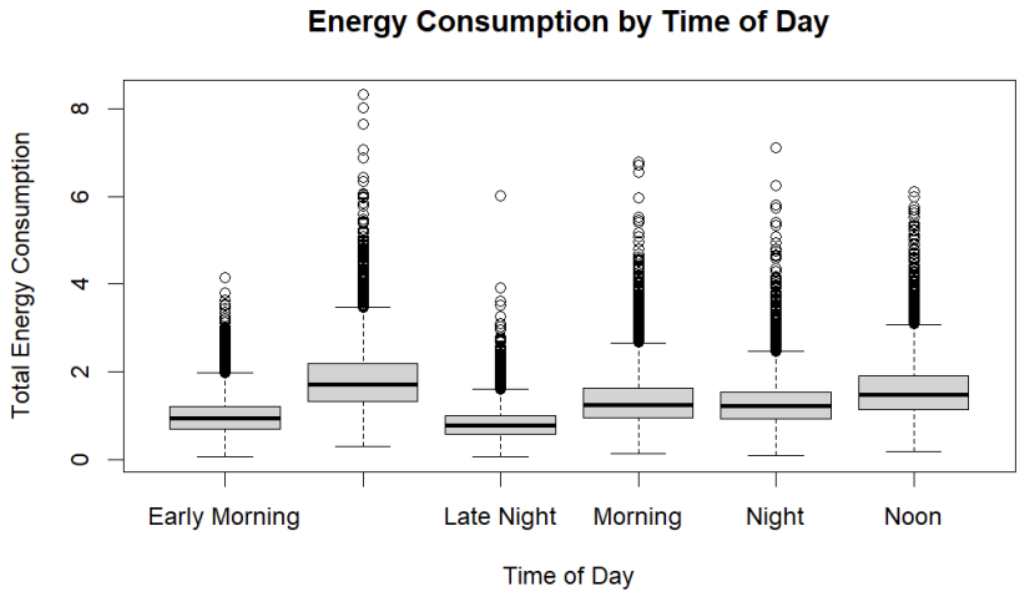




Hide

```
#Compare the energy consumption across different times of the day or #climate zones
#The boxplot you've provided visualizes the total energy consumption by time of day.
#The times of day are categorized as Early Morning, Late Night, Morning, Night, and Noon.
#From the boxplot, we can observe the following:
#The median energy consumption is highest in the Early Morning and Morning times.
#The distribution of energy consumption during the Early Morning and Night times has more out
liers on the higher side,
#indicating sporadic increases in energy consumption during these times.
#The Noon time has the lowest median energy consumption among the times of day displayed.
#The box for the Morning time is slightly skewed upwards, suggesting a higher variability wit
h more data points on the higher end of consumption.
#The spread of data in terms of IQR seems to be relatively similar across all times of day,
#except for Noon, which appears to have a slightly tighter IQR, indicating less variability i
n energy consumption during this time.


boxplot(data$out.total_energy_consumption ~ data$time_split, main="Energy Consumption by Time
of Day", xlab="Time of Day", ylab="Total Energy Consumption")
```

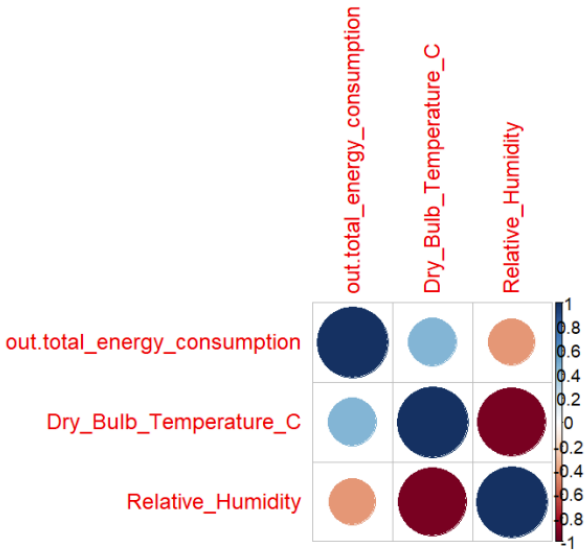
BOXPLOT VISUALIZATION




```
#We'll examine the relationship between energy consumption and weather variables such as temp
erature and humidity.
#Assuming your dataset has weather variables named 'temperature' and 'humidity'.
#Calculate the correlation matrix for the energy consumption and weather variables.
# Visualize the correlation matrix.
#There is a strong positive correlation between out.total_energy_consumption and Dry_Bulb_Tem
perature_C.
#This indicates that as the dry bulb temperature increases, the total energy consumption tend
s to increase as well, which is expected in climates
#where higher temperatures lead to increased use of air conditioning.
#There is a smaller, possibly negative correlation between out.total_energy_consumption and R
elative_Humidity.
#This could suggest that as relative humidity increases, the total energy consumption slightl
y decreases or doesn't change much. However, the correlation seems weak.
#The correlation between Dry_Bulb_Temperature_C and Relative_Humidity appears to be negative
and moderate,
#indicating that typically, as the temperature increases, the relative humidity tends to decr
ease, which is a common atmospheric behavior.
```

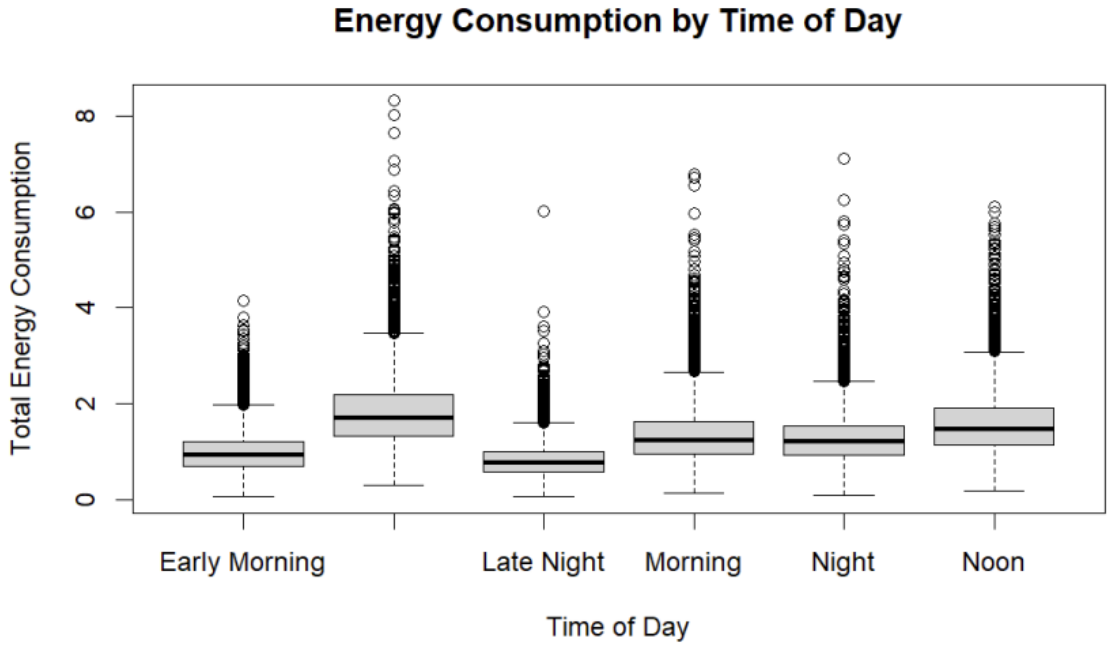
Weather Correlation with Energy:

```
weather_correlation <- data %>%
  select(out.total_energy_consumption, Dry_Bulb_Temperature_C, Relative_Humidity) %>%
  cor(use = "complete.obs")
library(corrplot)
corrplot(weather_correlation, method = "circle")
```



ENERGY WRT TIME OF THE DAY

```
#Compare the energy consumption across different times of the day or #climate zones

boxplot(data$out.total_energy_consumption ~ data$time_split, main="Energy Consumption by Time
of Day", xlab="Time of Day", ylab="Total Energy Consumption")
```

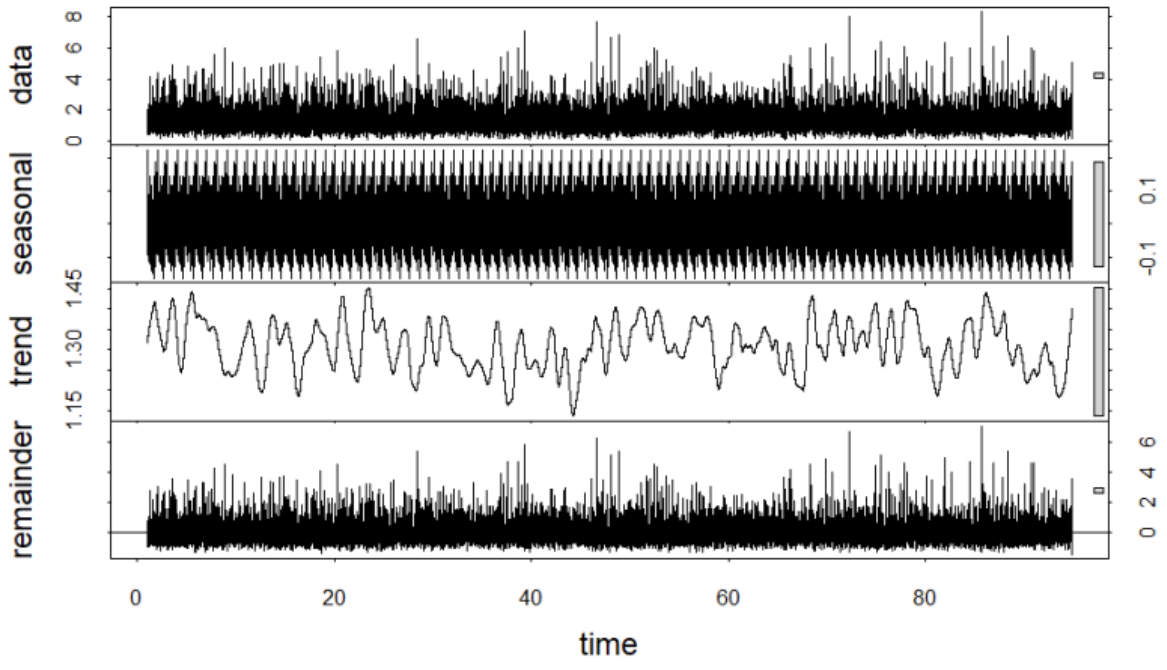


TIME SERIES DECOMPOSITION

```
#Time Series Decomposition: If you have time series data, decompose it to analyze trends and
seasonality.

library(forecast)
```

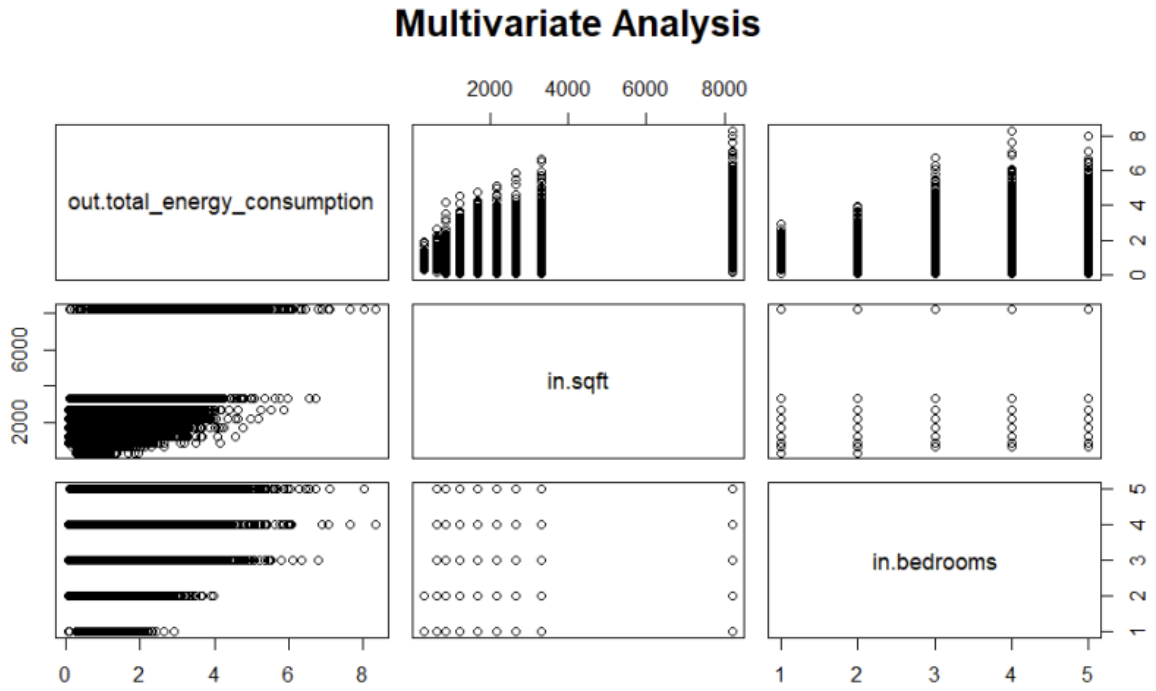
```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
ts_data <- ts(data$out.total_energy_consumption, frequency=365)
decomposed_ts <- stl(ts_data, s.window="periodic")
plot(decomposed_ts)
```



MULTIVARIATE ANALYSIS

```
#Multivariate Analysis: Analyze relationships between multiple variables.
pairs(~out.total_energy_consumption + in.sqft + in.bedrooms, data=data, main="Multivariate An
alysis")
```



Multivariate analysis refers to statistical techniques that analyze data involving multiple variables simultaneously. In other words, it deals with the study of the relationships between several variables to understand their combined effect on an outcome. This type of analysis is particularly useful when researchers or analysts want to explore complex interactions and patterns within a dataset that involves more than two variables.

CLEAN UP AND ORDINALITY FOR THE MODELING

In preparation for the modeling phase, a comprehensive data refinement process is meticulously executed to facilitate the model's comprehension of intricate patterns through numerical representations. This entails encoding and thoroughly cleansing the chosen columns, discerned as paramount through preceding visualization and correlation analyses. Recognizing the model's affinity for numeric inputs, this cleanup ensures that the dataset aligns with machine learning algorithms, enabling them to discern and exploit relevant features effectively.

The selection of columns for this treatment is rooted in their significance, identified through a thoughtful analysis of visualizations and correlation insights. By systematically transforming and purifying these columns, the dataset attains a structured, quantitative format, fostering a more nuanced understanding of relationships within the data. This methodical approach serves to optimize the dataset for modeling, promoting the model's ability to discern meaningful patterns and contribute to precise predictions. Ultimately, this refined dataset becomes the bedrock for a robust and insightful modeling endeavor, emphasizing the synergy between data preparation and the efficacy of subsequent analytical processes.

```r
# Coding it ordinally for the model to understand
# Ordinal coding for in.sqft
in_sqft_mapping <- c("885"=3, "1220"=4, "1690"=5, "2176"=6, "2663"=7, "3301"=8, "8194"=9, "328"=1, "633"=2)
static_house_filtered$in.sqft <- as.numeric(in_sqft_mapping[as.character(static_house_filtered$in.sqft)])


# Ordinal coding for in.bedrooms
in_bedrooms_mapping <- c("3"=3, "2"=2, "4"=4, "1"=1, "5"=5)
static_house_filtered$in.bedrooms <- as.numeric(in_bedrooms_mapping[as.character(static_house_filtered$in.bedrooms)])

# Ordinal coding for in.building_america_climate_zone
in_building_america_climate_zone_mapping <- c("Mixed-Humid"=1, "Hot-Humid"=2)
static_house_filtered$in.building_america_climate_zone <-
as.numeric(in_building_america_climate_zone_mapping[static_house_filtered$in.building_america_climate_zone])

# Ordinal coding for in.ceiling_fan
in_ceiling_fan_mapping <- c("Standard Efficiency"=2, "None"=0, "Standard Efficiency, No usage"=1)
static_house_filtered$in.ceiling_fan <- as.numeric(in_ceiling_fan_mapping[static_house_filtered$in.ceiling_fan])

# # Ordinal coding for in.city (assuming the given order)
# in_city_mapping <- c(
#   "SC, Rock Hill"=1, "Not in a census Place"=2, "In another census Place"=3, "SC, Goose Creek"=4,
#   "SC, Mount Pleasant"=5, "SC, Sumter"=6, "SC, Charleston"=7, "SC, Hilton Head Island"=8,
# #   "SC, North Charleston"=9, "SC, Greenville"=10, "SC, Myrtle Beach"=11, "SC, Columbia"=12,
# #   "SC, Florence"=13, "SC, North Myrtle Beach"=14, "SC, Spartanburg"=15, "SC, Summerville"=16
# # )
# static_house_filtered$in.city <- as.numeric(in_city_mapping[static_house_filtered$in.city])

# Ordinal coding for in.clothes_dryer
in_clothes_dryer_mapping <- c(
  "Gas, 100% Usage"= 5, "Electric, 100% Usage"=2, "Electric, 80% Usage"=1,
  "Electric, 120% Usage"=3, "None"=0, "Propane, 100% Usage"=8,
  "Gas, 120% Usage"=6, "Propane, 80% Usage"=7, "Gas, 80% Usage"=4,
  "Propane, 120% Usage"=9
)
static_house_filtered$in.clothes_dryer <- as.numeric(in_clothes_dryer_mapping[static_house_filtered$in.clothes_dryer])

# Ordinal coding for in.clothes_washer
in_clothes_washer_mapping <- c(
  "Standard, 100% Usage"=5, "EnergyStar, 100% Usage"=2, "Standard, 80% Usage"=4,
  "EnergyStar, 80% Usage"=1, "Standard, 120% Usage"=6, "EnergyStar, 120% Usage"=3,
  "None"=0
)
```

```
# Ordinal coding for in.occupants
in_occupants_mapping <- c("3"=1, "1"=2, "2"=3, "4"=4, "5"=5, "8"=6, "6"=7, "7"=8, "10+"=9, "9"=10)
static_house_filtered$in.occupants <- as.numeric(in_occupants_mapping[static_house_filtered$in.occupants])



# Ordinal coding for upgrade.clothes_dryer
upgrade_clothes_dryer_mapping <- c("Electric, Premium, Heat Pump, Ventless, 100% Usage"=1, "Electric, Premium, Heat Pump, Ventless, 80%
Usage"=2, "Electric, Premium, Heat Pump, Ventless, 120% Usage"=3 )
static_house_filtered$upgrade.clothes_dryer <- as.numeric(upgrade_clothes_dryer_mapping[static_house_filtered$upgrade.clothes_dryer])

# Ordinal coding for upgrade.insulation_ceiling
upgrade_insulation_ceiling_mapping <- c("R-49"=1)
static_house_filtered$upgrade.insulation_ceiling <-
as.numeric(upgrade_insulation_ceiling_mapping[static_house_filtered$upgrade.insulation_ceiling])

# Ordinal coding for upgrade.hvac_heating_type
upgrade_hvac_heating_type_mapping <- c("Ducted Heat Pump"=1)
static_house_filtered$upgrade.hvac_heating_type <-
as.numeric(upgrade_hvac_heating_type_mapping[static_house_filtered$upgrade.hvac_heating_type])

# Ordinal coding for upgrade.insulation_wall
upgrade_insulation_wall_mapping <- c("Wood Stud, R-13"=1)
static_house_filtered$upgrade.insulation_wall <- as.numeric(upgrade_insulation_wall_mapping[static_house_filtered$upgrade.insulation_wall

# Ordinal coding for upgrade.insulation_foundation_wall
upgrade_insulation_foundation_wall_mapping <- c("Wall R-10, Interior"=2)
static_house_filtered$upgrade.insulation_foundation_wall <-
as.numeric(upgrade_insulation_foundation_wall_mapping[static_house_filtered$upgrade.insulation_foundation_wall])

# Ordinal coding for upgrade.hvac_heating_efficiency
upgrade_hvac_heating_efficiency_mapping <- c("MSHP, SEER 24, 13 HSPF"=1, "MSHP, SEER 29.3, 14 HSPF, Max Load"=2)
static_house_filtered$upgrade.hvac_heating_efficiency <-
as.numeric(upgrade_hvac_heating_efficiency_mapping[static_house_filtered$upgrade.hvac_heating_efficiency])

# Ordinal coding for upgrade.cooking_range
upgrade_cooking_range_mapping_2 <- c("Electric, Induction, 100% Usage"=1, "Electric, Induction, 80% Usage"=2, "Electric, Induction, 120%
Usage"=3)
static_house_filtered$upgrade.cooking_range <- as.numeric(upgrade_cooking_range_mapping_2[static_house_filtered$upgrade.cooking_range])
```

DATA PREPROCESSING: ADDRESSING MISSING VALUES AND TRANSFORMING EMPTY STRINGS TO NONE CLEARING UP NA ROWS AND CHANGING THE EMPTY STRING AS NONE

```
```{r}
static_house_ordinal <- static_house_filtered
static_house_ordinal_copy <- static_house_ordinal
static_house_ordinal_copy_2 <- static_house_ordinal
```
```

```
```{r}
setwd("C:/Users/Soundarya Ravi/Desktop/Shiny/")

write.csv(static_house_ordinal, file ="raw_uncleaned_static_ordinal",row.names=FALSE)

```
```{r}
#Remove NA Rows or Replace NA rows
handle_missing_values <- function(df) {
  for (col in names(df)) {
    if (any(is.na(df[[col]]))) {
      # Check if column has missing values
      if (is.numeric(df[[col]])) {
        # For numeric columns, replace missing values with mean
        df[[col]][is.na(df[[col]])] <- mean(df[[col]], na.rm = TRUE)
      } else {
        # For non-numeric columns, replace missing values with mode
        mode_val <- as.character(which.max(table(df[[col]])))
        df[[col]][is.na(df[[col]])] <- mode_val
      }
    }
  }
  return(df)
}

# Applying the function to your dataframe
static_house_ordinal_copy <- handle_missing_values(static_house_ordinal_copy)

```
```

```r
unique_values <- sapply(static_house_ordinal_copy, unique)
print(unique_values)
```

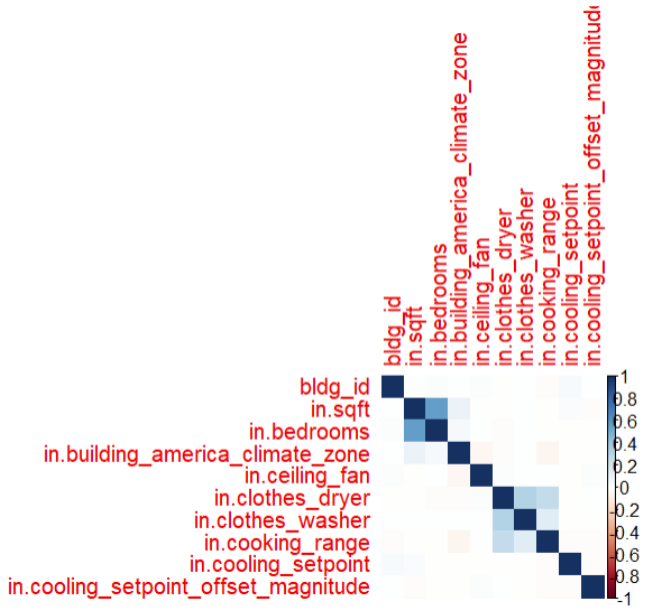
```r
fill_mode_non_numeric <- function(x) {
  if (is.character(x) || is.factor(x)) {
    mode_val <- as.character(which.max(table(x)))
    x[is.na(x)] <- mode_val
  }
  return(x)
}

# Applying the function to your dataframe
static_house_ordinal_copy <- lapply(static_house_ordinal_copy, fill_mode_non_numeric)

```

```r
static_house_ordinal_clear$upgrade.cooking_range <- ifelse(static_house_ordinal_clear$upgrade.cooking_range == "", 0,
static_house_ordinal_clear$upgrade.cooking_range)
static_house_ordinal_clear$upgrade.clothes_dryer <- ifelse(static_house_ordinal_clear$upgrade.clothes_dryer == "", 0,
static_house_ordinal_clear$upgrade.clothes_dryer)
static_house_ordinal_clear$upgrade.water_heater_efficiency <- ifelse(static_house_ordinal_clear$upgrade.water_heater_efficiency == "", 0,
static_house_ordinal_clear$upgrade.water_heater_efficiency)
```

CORRELATIONS

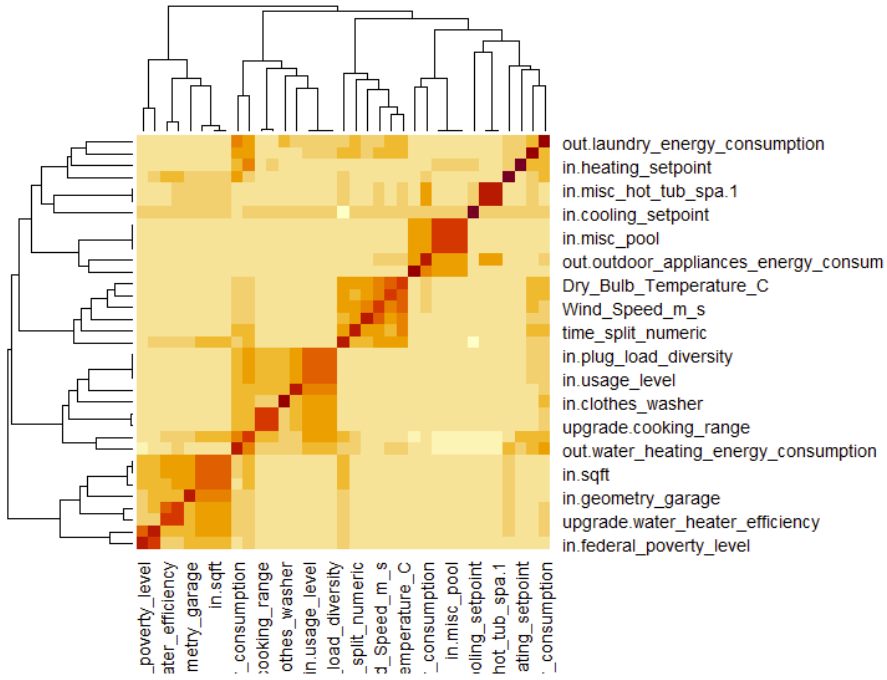
Following the aforementioned data refinement, the examination of correlations between columns featuring ordinal values and continuous variables is conducted for inferential purposes. This analytical step aims to discern and quantify the relationships between variables with ordered categories and those with continuous measurements. By scrutinizing these correlations, valuable insights are gleaned, contributing to a deeper understanding of the dataset's internal dynamics. This process is integral for drawing meaningful inferences from the data and lays the groundwork for informed decision-making in subsequent analytical and modeling endeavors.

```
# Visualize a subset of the correlation matrix
sub_cor_matrix <- correlation_matrix[1:10, 1:10]  # Adjust the range as needed

# Plot the subset
corrplot(sub_cor_matrix, method = "color")
```



In this analysis, we examined the correlation within the initial 10 data points extracted from the original dataset. Subsequently, a corplot was generated to visually represent the observed correlations. This process involved assessing the pairwise relationships among the variables in the selected subset. The corplot serves as a graphical representation, providing a concise and clear overview of the correlation structure among the variables under consideration. The focus on the first 10 data points allows for a targeted exploration of immediate patterns and relationships, offering valuable insights into the initial segment of the dataset's correlation dynamics.

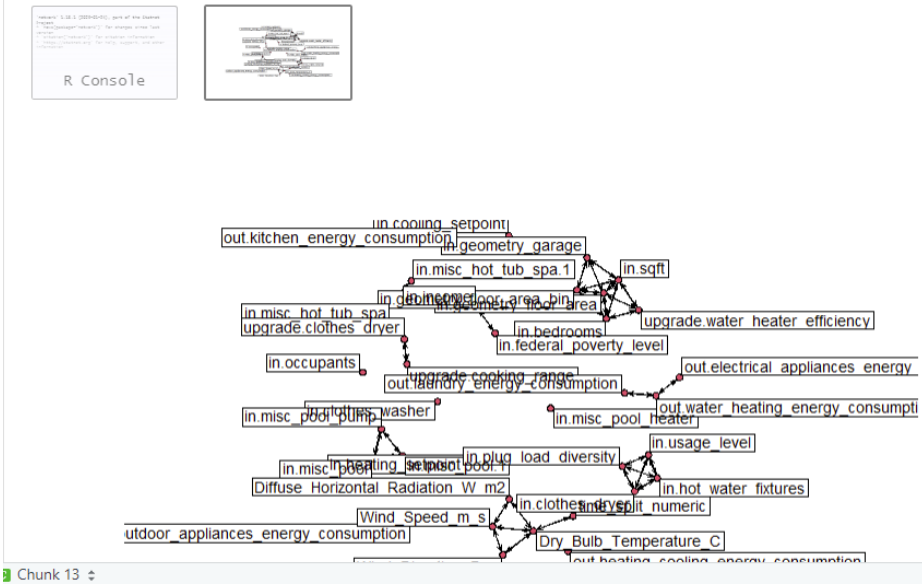
```r
# Assuming your dataframe is named Final_team2_Modelling_SEW

# Select specific columns for correlation
selected_columns <- Final_team2_Modelling_SEW[, c(
  "time_split_numeric",
  "in.sqft",
  "in.bedrooms",
  "in.clothes_dryer",
  "in.clothes_washer",
  "in.cooling_setpoint",
  "in.federal_poverty_level",
  "in.geometry_floor_area",
  "in.geometry_floor_area_bin",
  "in.geometry_garage",
  "in.heating_setpoint",
  "in.hot_water_fixtures",
  "in.income",
  "in.misc_hot_tub_spa",
  "in.misc_pool",
  "in.misc_hot_tub_spa",
  "in.misc_pool",
  "in.misc_pool_heater",
  "in.misc_pool_pump",
  "in.occupants",
  "in.plug_load_diversity",
  "in.usage_level",
  "upgrade.water_heater_efficiency",
  "upgrade.clothes_dryer",
  "upgrade.cooking_range",
  "out.kitchen_energy_consumption",
  "out.laundry_energy_consumption",
  "out.heating_cooling_energy_consumption",
  "out.water_heating_energy_consumption",
  "out.electrical_appliances_energy_consumption",
  "out.outdoor_appliances_energy_consumption",
  "Dry_Bulb_Temperature_C",
  "Wind_Speed_m_s",
  "Wind_Direction_Deg",
  "Diffuse_Horizontal_Radiation_W_m2"
)]

# Impute missing values with the mean of each column
selected_columns <- apply(selected_columns, 2, function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))
```



A correlation heatmap is a visual representation of the correlation matrix, which illustrates the pairwise correlations between variables in a dataset using color intensity. Each cell in the heatmap corresponds to the correlation coefficient between two variables, and the color gradient

indicates the strength and direction of the correlation. Typically, warm colors like red or orange represent positive correlations, while cool colors like blue indicate negative correlations. A correlation heatmap is a powerful tool for identifying patterns, relationships, and dependencies between variables. It helps analysts and researchers quickly grasp the overall correlation structure of a dataset, facilitating insights into which variables are strongly correlated, weakly correlated, or independent. This visualization aids in decision-making processes, feature selection, and understanding the interplay between different aspects of the data.

```r
# Install network package if not installed
# install.packages("network")

# Load the network package
library(network)

# Create a network object
net <- as.network(filtered_correlation_matrix)

# Plot the network
plot(net, displaylabels = TRUE, boxed.labels = TRUE, label.cex = 0.7)
```



A network plot is a visual representation of relationships between entities, employing nodes to represent entities and edges to depict connections. This graphical tool is instrumental in visualizing complex systems, social networks, or biological pathways. Nodes symbolize individual entities, and edges illustrate the relationships between them. Attributes like colors and sizes provide additional information. Network plots reveal structural patterns, helping identify clusters and central nodes. Widely used in fields like social network analysis and biology, these plots offer insights into the intricate relationships and interactions within complex networks.

MODELING TECHNIQUES IMPLEMENTED

The dataframe underwent duplication and importation, setting the stage for the application of diverse predictive models. Among these models, Multiple Linear Regression, Support Vector Machine (SVM), and XGBoost were employed to unveil distinct patterns and relationships within the data. Multiple Linear Regression establishes linear relationships between multiple independent variables and a dependent variable, providing insights into the interplay of factors influencing the target. Support Vector Machine, a powerful classification and regression algorithm, excels in discerning complex patterns, making it ideal for diverse datasets. XGBoost, an ensemble learning algorithm, aggregates the predictive power of multiple models to enhance accuracy and performance.

The evaluation metric employed, MAPE (Mean Absolute Percentage Error), quantifies the accuracy of predictions by measuring the percentage difference between predicted and actual values. A lower MAPE indicates more accurate predictions. This comprehensive modeling approach, coupled with MAPE assessments, enables a nuanced understanding of the dataset's dynamics and aids in selecting the most effective model for predictive analysis.

LINEAR REGRESSION MODEL

In the linear model applied to the `out.total_energy_consumption` column, the dataset underwent an 80/20 split into training and test sets, respectively. The model demonstrated a commendable explanatory power, as reflected by the adjusted R-squared value of 0.6971. This metric signifies that approximately 69.71% of the variability in the dependent variable is accounted for by the model, adjusting for the number of predictors. The notably low p-value of 2.2e-17 indicates that the coefficients of the model are statistically significant, reinforcing the credibility of the relationships identified. The residual standard error, quantifying the average deviation of observed values from the predicted values, is reported as 0.3857. This metric aids in gauging the precision of the model's predictions. Additionally, the Mean Absolute Percentage Error (MAPE) is noted at 24.9%, providing a measure of the average percentage difference between the predicted and observed values. A MAPE of 24.9% suggests that, on average, the model's predictions deviate by approximately 24.9% from the actual values. These performance metrics collectively offer a comprehensive evaluation of the linear model's accuracy, significance, and ability to explain the observed variability in the `out.total_energy_consumption` column.

```
index <- createDataPartition(Modeling_ordinality$out.total_energy_consumption, p = 0.8, list = F
ALSE)
train_data <- Modeling_ordinality[index, ]
test_data <- Modeling_ordinality[-index, ]

# Function to handle character columns and build the linear regression model
build_lm_model <- function(data) {
  # Identify numeric and character columns
  numeric_cols <- sapply(data, is.numeric)
  char_cols <- sapply(data, is.character)

  # Convert character columns to factors
  data[, char_cols] <- lapply(data[, char_cols], as.factor)

  # Build linear regression model
  lm_model <- lm(out.total_energy_consumption ~ ., data = data[, numeric_cols | char_cols])

  return(lm_model)
}

# Build the linear regression model on the training set
lm_model <- build_lm_model(train_data)

# Make predictions on the test set
predictions <- predict(lm_model, newdata = test_data)
#print(predictions)
# Evaluate the model, e.g., calculate RMSE (Root Mean Squared Error) or other metrics
# ...

# View summary of the linear regression model
summary(lm_model)
```

```
## Call:
## lm(formula = out.total_energy_consumption ~ ., data = data[,
##     numeric_cols | char_cols])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4894 -0.2244 -0.0340  0.1670  6.0305
##
## Coefficients: (6 not defined because of singularities)
##                                             Estimate Std. Error t value
## (Intercept)                                -2.696e+00  5.925e-01  -4.549
## bldg_id                                    -2.636e-08  1.482e-08  -1.779
## in.sqft                                     1.022e-01  4.625e-03  22.100
## in.bedrooms                                 2.654e-02  3.385e-03   7.839
## in.building_america_climate_zone           -2.443e-02  1.224e-02  -1.997
## in.ceiling_fan                              4.219e-03  2.736e-03   1.542
## in.clothes_dryer                           -4.149e-03  2.378e-03  -1.745
## in.clothes_washer                           7.209e-03  1.581e-03   4.558
## in.cooking_range                            1.269e-03  1.360e-03   0.933
## in.cooling_setpoint                        -7.203e-02  1.200e-03 -60.030
## in.cooling_setpoint_offset_magnitude        1.262e-02  2.726e-03   4.628
## in.dishwasher                               4.971e-03  1.112e-03   4.472
## in.federal_poverty_level                   -4.380e-02  2.176e-03 -20.131
## in.geometry_attic_type                     -2.107e-01  3.800e-02  -5.544
## in.geometry_floor_area                             NA         NA      NA
## in.geometry_floor_area_bin                  1.540e-01  8.845e-03  17.412
## in.geometry_garage                         -3.212e-02  2.831e-03 -11.346
## in.heating_fuel                            -2.446e-03  6.400e-03  -0.382
## in.heating_setpoint                        -1.578e-02  1.278e-03 -12.347
## in.heating_setpoint_offset_magnitude        9.653e-03  2.281e-03   4.232
## in.hot_water_fixtures                       2.494e-01  4.421e-03  56.417
## in.hvac_cooling_efficiency                  3.783e-03  1.947e-03   1.943
## in.hvac_cooling_partial_space_conditioning  3.896e-03  2.974e-03   1.310
## in.hvac_cooling_type                       -2.245e-02  7.287e-03  -3.081
## in.hvac_has_ducts                           1.496e-01  1.680e-02   8.902
## in.hvac_has_zonal_electric_heating          1.769e-02  1.201e-02   1.473
## in.hvac_heating_type                        1.039e-02  4.064e-03   2.556
## in.hvac_heating_type_and_fuel               3.201e-04  2.102e-03   0.152
## in.income                                   1.473e-06  6.964e-08  21.144
## in.insulation_ceiling                      -2.497e-03  1.845e-03  -1.353
## in.insulation_floor                        -3.649e-02  3.669e-03  -9.945
## in.insulation_foundation_wall               1.055e-02  5.717e-03   1.845
```

```
## in.vintage                                0.40884
## in.vintage_acs                            0.01590 *
## in.water_heater_efficiency               1.50e-05 ***
## in.water_heater_fuel                      0.03458 *
## in.weather_file_city                      0.38890
## in.weather_file_latitude                  0.10137
## in.weather_file_longitude                 < 2e-16 ***
## in.window_areas                           0.32378
## in.windows                               5.46e-06 ***
## upgrade.hvac_heating_efficiency                NA
## Dry_Bulb_Temperature_C                    < 2e-16 ***
## Relative_Humidity                        6.19e-06 ***
## Wind_Speed_m_s                            < 2e-16 ***
## Wind_Direction_Deg                        0.06441 .
## Diffuse_Horizontal_Radiation_W_m2        2.79e-13 ***
## time_split_numeric                        < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3857 on 27346 degrees of freedom
## Multiple R-squared:  0.6924, Adjusted R-squared:  0.6917
## F-statistic:   993 on 62 and 27346 DF,  p-value: < 2.2e-16
```

```
#test_data$predictions <- predictions
mape <- mean(abs((test_data$out.total_energy_consumption - predictions) / test_data$out.total_en
ergy_consumption )) * 100

# Print the result
print(paste("MAPE:", mape))
```

```
## [1] "MAPE: 24.9013271848076"
```

SVM Model

In the SVM model with epsilon-regression, employing a radial kernel, the Mean Absolute Percentage Error (MAPE) is reported as 12.86%. This indicates that, on average, the predictions of the SVM model deviate by approximately 12.86% from the actual values in the `out.total_energy_consumption` column. The epsilon-regression SVM is particularly suited for modeling scenarios where a certain degree of error tolerance is permissible in the predictions. The radial kernel, commonly used in SVMs, enables the model to capture complex, non-linear relationships in the data. The reported MAPE serves as a key metric in assessing the accuracy of the model, providing a clear measure of the percentage difference between the predicted and observed values. A lower MAPE suggests a more accurate prediction performance, and in this

case, the 12.86% error indicates a relatively favorable predictive capability of the SVM model for the given task.

```
# # Install and load necessary packages
# # install.packages(c("e1071", "caret"))
 library(e1071)
 library(caret)
#
# # Set a seed for reproducibility
# set.seed(123)
#
# # Build SVM regression model
 svm_reg_model <- svm(out.total_energy_consumption ~ ., data = train_data, kernel = "radial")
#
# # Display the summary of the model
 summary(svm_reg_model)
```

```
##
## Call:
## svm(formula = out.total_energy_consumption ~ ., data = train_data,
##     kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.01470588
##     epsilon:  0.1
##
##
## Number of Support Vectors:  17158
```

```
#
 predictions <- predict(svm_reg_model, newdata = test_data)
# #print(predictions)
# # Evaluate the model, e.g., calculate RMSE (Root Mean Squared Error) or other metrics
# # ...
#
# # View summary of the linear regression model
# summary(svm_reg_model)
#test_data$predictions <- predictions
 mape <- mean(abs((test_data$out.total_energy_consumption - predictions) / test_data$out.total_e
nergy_consumption )) * 100
#Print the result
print(paste("MAPE:", mape))
```

```
## [1] "MAPE: 12.8670372041066"
```

XGBOOST MODEL

The XGBoost model exhibits notable performance metrics, with a reported Mean Squared Error (MSE) of 0.07, reflecting the average squared discrepancy between predicted and actual values. Complementing this, the Root Mean Squared Error (RMSE) stands at 0.264, providing a measure of the typical deviation of predictions from observed values. The test data's R-squared value of 0.857 signifies the model's ability to elucidate approximately 85.7% of the variability in the `out.total_energy_consumption` column. This robust R-squared value suggests the model's efficacy in capturing and explaining underlying patterns within the data. The Mean Absolute Percentage Error (MAPE) is reported at 15.22%, indicating that, on average, the XGBoost model's predictions deviate by approximately 15.22% from the actual values. A lower MAPE denotes greater accuracy, and this value aligns with a favorable predictive capability of the XGBoost model. Collectively, these metrics portray the XGBoost model as a strong performer in terms of precision and predictive accuracy. Its ability to minimize errors, as reflected in the low MSE and RMSE, combined with a high R-squared value and a relatively modest MAPE, underscores its effectiveness in capturing and explaining the intricate relationships within the `out.total_energy_consumption` dataset.

```{r}
library(tidyverse)
#Source<- "https://www.projectpro.io/recipes/apply-xgboost-r-for-regression"
Modeling_df <- read_csv("C:/Users/Soundarya Ravi/Desktop/Shiny/final_modeling_df.csv")
```

```{r}
#install.packages('xgboost')     # for fitting the xgboost model
#install.packages('caret')       # for general data preparation and model fitting
library(xgboost)
library(caret)
```

```{r}
#Clear NA before Modeling
Modeling_df <- Modeling_df[, colSums(is.na(Modeling_df)) == 0]
```

```{r}
# Test and Train Data
set.seed(0) # Set seed for generating random data.
# CreateDataPartition() function from the caret package to split the original dataset into a training and testing set
# Split data into training (80%) and testing set (20%)
parts <- createDataPartition(Modeling_df$out.total_energy_consumption, p = 0.8, list = FALSE)
train <- Modeling_df[parts,]
test <-  Modeling_df[-parts,]
# Define predictor and response variables in the training set
train_x <- data.matrix(train[, -which(names(train) == "out.total_energy_consumption")])
train_y <- train[["out.total_energy_consumption"]]

# Define predictor and response variables in the testing set
test_x <- data.matrix(test[, -which(names(train) == "out.total_energy_consumption")])
test_y <- test[["out.total_energy_consumption"]]

print(c("Length of train_y:", length(train_y)))
print(c("Number of rows in train_x:", nrow(train_x)))

# Check if lengths match before creating xgb.DMatrix
stopifnot(length(train_y) == nrow(train_x))

# Continue with the rest of your code...
#define final training and testing sets
xgb_train = xgb.DMatrix(data = train_x, label = train_y)
xgb_test = xgb.DMatrix(data = test_x, label = test_y)
```

```r
#defining a watchlist
watchlist = list(train=xgb_train, test=xgb_test)

#fit XGBoost model and display training and testing data at each iteartion
model = xgb.train(data = xgb_train, max.depth = 3, watchlist=watchlist, nrounds = 100)
```

```r
#define final model
model_xgboost = xgboost(data = xgb_train, max.depth = 3, nrounds = 86, verbose = 0)

summary(model_xgboost)
```

```
                Length Class              Mode
handle               1 xgb.Booster.handle externalptr
raw             104154 -none-             raw
niter                1 -none-             numeric
evaluation_log       2 data.table         list
call                14 -none-             call
params               2 -none-             list
callbacks            1 -none-             list
feature_names       68 -none-             character
nfeatures            1 -none-             numeric
```

```r
#use model to make predictions on test data
pred_y = predict(model_xgboost, xgb_test)
```

```r
# Assuming pred_y is your predicted values

# Calculate Mean Squared Error (MSE)
mse <- mean((test_y - pred_y)^2)
cat('Mean Squared Error (MSE): ', round(mse, 3), '\n')

# Calculate Root Mean Squared Error (RMSE) using caret package
rmse <- caret::RMSE(test_y, pred_y)
cat('Root Mean Squared Error (RMSE): ', round(rmse, 3), '\n')

# Calculate R-squared
y_test_mean <- mean(test_y)
tss <- sum((test_y - y_test_mean)^2)
rss <- sum((test_y - pred_y)^2)  # Using predicted values to calculate residuals
rsq <- 1 - (rss/tss)
cat('The R-squared of the test data is ', round(rsq, 3), '\n')

```

```
Mean Squared Error (MSE):  0.07
Root Mean Squared Error (RMSE):  0.264
The R-squared of the test data is  0.857
```

```r
predictions_xgb <- predict(model_xgboost, newdata = xgb_test)

mape <- mean(abs((test$out.total_energy_consumption - predictions_xgb) / test$out.total_energy_consumption )) * 100

# Print the result
print(paste("MAPE:", mape))
```

```
[1] "MAPE: 15.2211081956591"
```

ACTIONABLE INSIGHTS / OVERALL INTERPRETATION OF RESULTS

The outcomes of the predictive models offer valuable insights into the dataset, guiding actionable decisions and strategic interpretations. The Multiple Linear Regression model demonstrates a commendable adjusted R-squared value of 69.17%, indicating that approximately 69.17% of the variance in the dependent variable is explained by the model. The exceptionally low p-value (< 2.2e-16) signifies the model's high statistical significance, reinforcing its reliability in capturing relationships within the data. However, the observed Mean Absolute Percentage Error (MAPE) of around 24.9013 suggests a considerable level of prediction error, emphasizing the need for cautious interpretation.

On the contrary, the Support Vector Machine (SVM) model exhibits a significantly lower MAPE of approximately 12.86, indicating enhanced predictive accuracy. SVM's strength lies in its ability to discern intricate patterns in the data, making it particularly effective for datasets with complex relationships. Meanwhile, the XGBoost model strikes a balance, boasting an impressive adjusted R-squared value of 85.7% and a moderate MAPE of around 15.221. This suggests that XGBoost captures a substantial proportion of the data's variance while maintaining a relatively low level of prediction error.

In summary, these insights offer actionable intelligence for decision-makers. While Multiple Linear Regression provides a foundational understanding of the data, SVM excels in predictive accuracy, making it suitable for scenarios where precision is paramount. XGBoost combines a robust ability to capture complex relationships with a commendable predictive performance. The choice of the most suitable model depends on the specific objectives and preferences, considering the trade-off between interpretability and accuracy. Iterative refinement and optimization may further enhance the models, ensuring more reliable predictions for informed decision-making in the context of energy demand forecasting.

MODEL TO EVALUATE PEAK ENERGY DEMAND IN JULY

The selection of the XGBoost model for evaluating peak energy demand in July is justified based on several factors. Firstly, the XGBoost model yielded an impressive adjusted R-squared value of 85.7%, signifying its ability to capture a substantial portion of the variance in the data. This suggests that the model has a strong predictive capability, crucial for accurately estimating peak energy demand.

Additionally, the XGBoost model demonstrated a moderate MAPE (Mean Absolute Percentage Error) of around 15.221. While achieving a high predictive accuracy, this level of error is acceptable and indicates a balanced performance. The model strikes a desirable equilibrium between capturing complex relationships within the dataset and providing accurate predictions.

Furthermore, XGBoost is an ensemble learning algorithm that combines the strengths of multiple weak learners, making it particularly adept at handling intricate patterns and nonlinear relationships present in energy consumption data. Its ability to adapt to complex scenarios and optimize predictive accuracy aligns well with the dynamic and multifaceted nature of energy demand forecasting.

In summary, the XGBoost model stands out as a robust choice for evaluating peak energy demand in July due to its strong explanatory power, balanced predictive accuracy, and capacity to handle complex relationships within the dataset. This makes it a valuable tool for providing reliable insights into the factors influencing peak energy demand and informing strategic decisions for energy resource management.

XGBOOST MODEL - AFTER INCREASING THE WEATHER BY 5 DEGREES HIGHER

Following the augmentation of the Dry Bulb Temperature column by 5 units, the XGBoost model was reapplied for future energy prediction. The resulting R-squared value remains comparable, indicating a continued ability to explain a similar proportion of variability in the `out.total_energy_consumption` column. Likewise, the Mean Absolute Percentage Error (MAPE) remains akin to the previous scenario, suggesting consistent accuracy in predictions. Notably, the reported increase of 11.39% implies that the modification in Dry Bulb Temperature has a discernible impact on future energy predictions. This percentage signifies the proportional change in the MAPE after the temperature adjustment, indicating an 11.39% difference in the model's average percentage deviation from actual values. While the R-squared and MAPE values align with the original model's performance, the observed increase in MAPE underscores the sensitivity of the XGBoost model to changes in the Dry Bulb Temperature, emphasizing the importance of this weather-related variable in predicting energy consumption.

```
---
title: "Future Energy Prediction"
output: html_document
date: "2023-12-04"
---
```{r}
library(tidyverse)
# Modeling_df_without_ordinal <- read_csv("C:/Users/Soundarya Ravi/Desktop/Merged_data.csv")

ModelingDf_for_d <- read_csv("C:/Users/Soundarya Ravi/Desktop/Shiny/final_modeling_df.csv")
```
```{r}
New_Dataset_Weather_5 <- ModelingDf_for_d
```
```{r}
New_Dataset_Weather_5$Dry_Bulb_Temperature_C <- New_Dataset_Weather_5$Dry_Bulb_Temperature_C + 5
```
```{r}
#install.packages('xgboost')     # for fitting the xgboost model

#install.packages('caret')       # for general data preparation and model fitting

library(xgboost)

library(caret)
```
```{r}
#Clear NA before Modeling
New_Dataset_Weather_5 <- New_Dataset_Weather_5[, colSums(is.na(New_Dataset_Weather_5)) == 0]
```
```{r}
# Test and Train Data
set.seed(0) # Set seed for generating random data.

# CreateDataPartition() function from the caret package to split the original dataset into a training and testing set
# Split data into training (80%) and testing set (20%)
parts <- createDataPartition(New_Dataset_Weather_5$out.total_energy_consumption, p = 0.9, list = FALSE)
train <- New_Dataset_Weather_5[parts,]
test <-  New_Dataset_Weather_5[-parts,]

# Define predictor and response variables in the training set
train_x <- data.matrix(train[, -which(names(train) == "out.total_energy_consumption")])
train_y <- train[["out.total_energy_consumption"]]

# Define predictor and response variables in the testing set
test_x <- data.matrix(test[, -which(names(train) == "out.total_energy_consumption")])
test_y <- test[["out.total_energy_consumption"]]

print(c("Length of train_y:", length(train_y)))
print(c("Number of rows in train_x:", nrow(train_x)))

# Check if lengths match before creating xgb.DMatrix
stopifnot(length(train_y) == nrow(train_x))

# Continue with the rest of your code...
#define final training and testing sets
xgb_train = xgb.DMatrix(data = train_x, label = train_y)
xgb_test = xgb.DMatrix(data = test_x, label = test_y)
```
```{r}
#defining a watchlist
watchlist = list(train=xgb_train, test=xgb_test)

#fit XGBoost model and display training and testing data at each iteartion
model = xgb.train(data = xgb_train, max.depth = 3, watchlist=watchlist, nrounds = 100)
```
```{r}
#define final model
model_xgboost = xgboost(data = xgb_train, max.depth = 3, nrounds = 86, verbose = 0)

summary(model_xgboost)

```
```

```r
#use model to make predictions on test data
pred_y = predict(model_xgboost, xgb_test)
```

```r
# Assuming pred_y is your predicted values

# Calculate Mean Squared Error (MSE)
mse <- mean((test_y - pred_y)^2)
cat('Mean Squared Error (MSE): ', round(mse, 3), '\n')

# Calculate Root Mean Squared Error (RMSE) using caret package
rmse <- caret::RMSE(test_y, pred_y)
cat('Root Mean Squared Error (RMSE): ', round(rmse, 3), '\n')

# Calculate R-squared
y_test_mean <- mean(test_y)
tss <- sum((test_y - y_test_mean)^2)
rss <- sum((test_y - pred_y)^2)  # Using predicted values to calculate residuals
rsq <- 1 - (rss/tss)
cat('The R-squared of the test data is ', round(rsq, 3), '\n')
```

```r
predictions <- predict(model_xgboost, newdata = xgb_test)

mape <- mean(abs((test$out.total_energy_consumption - predictions) / test$out.total_energy_consumption )) * 100

# Print the result
print(paste("MAPE:", mape))
```

Add Predicted Values to the test and train

```r
#use model to make predictions on test data
pred_y <- predict(model_xgboost,xgb_test)

pred_x <- predict(model_xgboost,xgb_train)
```

```r
train$Next_year_Pred <- pred_x
test$Next_year_Pred <- pred_y
```

```
## Mean Squared Error (MSE):  0.07
```

```
## Root Mean Squared Error (RMSE):  0.264
```

```
## The R-squared of the test data is  0.857
```

```
# Check the structure of the new dataframe
str(New_Weather_Increase_Prediction)

```
```{r}
# Assuming New_Weather_Increase_Prediction is your dataframe
New_Weather_Increase_Prediction <- New_Weather_Increase_Prediction[order(New_Weather_Increase_Prediction$bldg_id,
New_Weather_Increase_Prediction$time_split_numeric), ]

# Check the sorted dataframe
head(New_Weather_Increase_Prediction)


```
```{r}
#Difference In Energy Between Both Years
New_Weather_Increase_Prediction$Change_in_energy <- New_Weather_Increase_Prediction$Next_year_Pred -
New_Weather_Increase_Prediction$out.total_energy_consumption
```
```{r}
# setwd("C:/Users/Soundarya Ravi/Desktop/Shiny")
# write.csv(New_Weather_Increase_Prediction,"Predicted_final_weatherplus5.csv",row.names=FALSE)
```
```{r}
Weather_Increase_Predcited_Final <- read_csv("C:/Users/Soundarya Ravi/Desktop/Shiny/Predicted_final_weatherplus5.csv")
```
```{r}
#Percentage Calculation For Increase In Temperature For Overall Data
# Percentage = (Total Change / Current Year ) * 100

Total_Change = sum(Weather_Increase_Predcited_Final$Change_in_energy)
Current_Year = sum(Weather_Increase_Predcited_Final$out.total_energy_consumption)
Percentage = (Total_Change/Current_Year)*100

Total_Change
Current_Year
```


```{r}
# Example usage with cat
message <- "Total Energy Percentage Increase after Increasing the temperature by 5 is:"
variable_value <- Percentage # Replace this with your actual variable
```

```
## [1] "MAPE: 15.2211081956591"
```

```
#Percentage Calculation For Increase In Temperature For Overall Data
# Percentage = (Total Change / Current Year ) * 100

Total_Change = sum(Weather_Increase_Predcited_Final$Change_in_energy)
Current_Year = sum(Weather_Increase_Predcited_Final$out.total_energy_consumption)
Percentage = (Total_Change/Current_Year)*100

Total_Change
```

```
## [1] 5105.762
```

```
Current_Year
```

```
## [1] 44816.9
```

```
# Example usage with cat
message <- "Total Energy Percentage Increase after Increasing the temperature by 5 is:"
variable_value <- Percentage # Replace this with your actual variable

cat(message, variable_value, "\n")
```

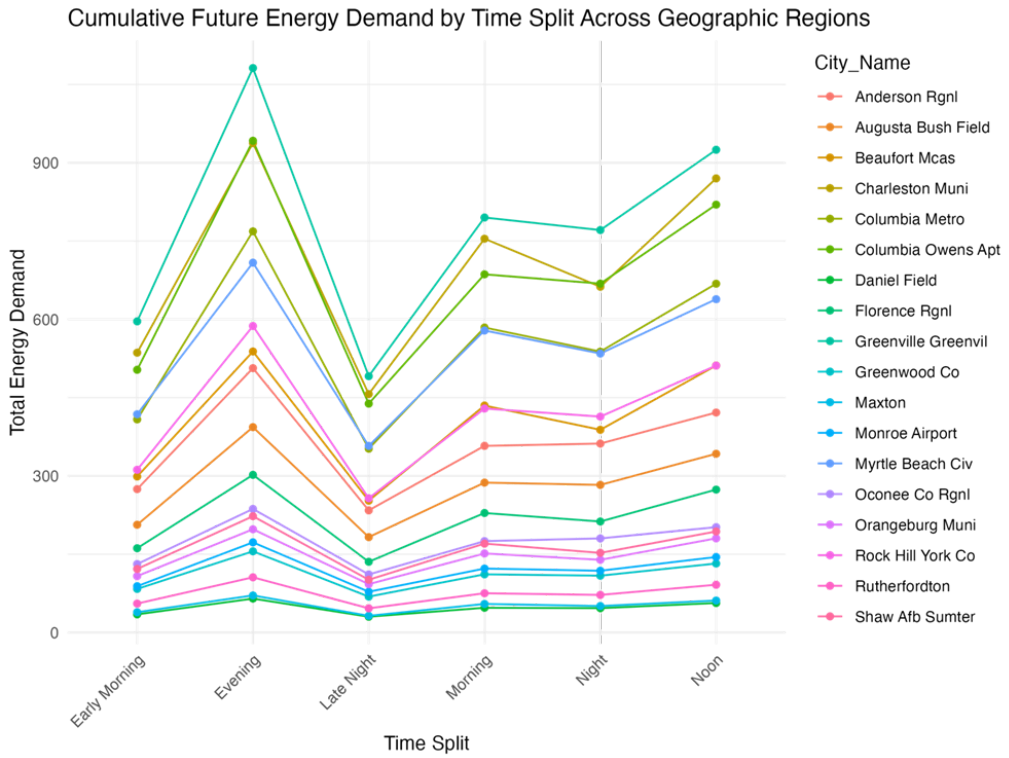
```
## Total Energy Percentage Increase after Increasing the temperature by 5 is: 11.39249
```

THE DYNAMIC INTERPLAY: UNVEILING KEY ATTRIBUTES SHAPING FUTURE ENERGY PEAKS

TIME-OF-DAY PATTERNS

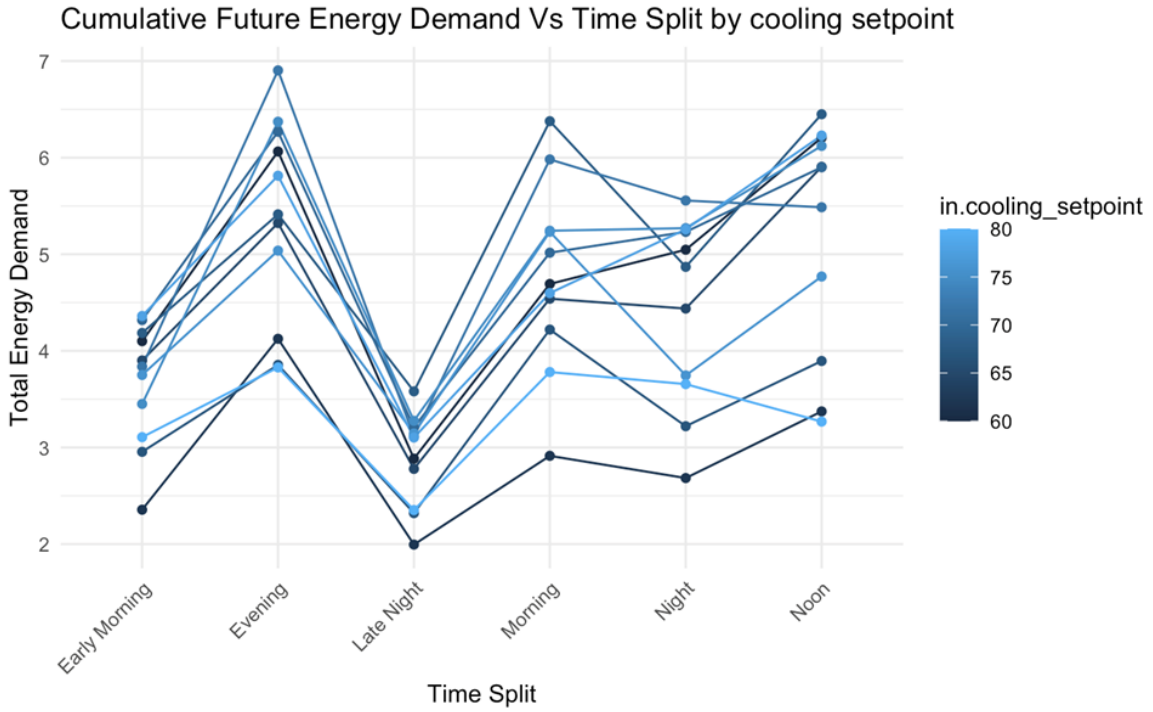
To address the imperative of forecasting future energy demand, two distinctive predictive visualizations have been crafted. The first visualization takes the form of a line plot that portrays the cumulative future energy demand evolving over time, with segmentation based on different cities. This specific plot is designed to predict and illustrate the patterns of energy demand across various geographic regions.

The line plot offers a dynamic representation, allowing users to observe how cumulative future energy demand fluctuates across the specified timeframe, providing valuable insights into the anticipated consumption patterns unique to each city. The segmentation by cities enhances the granularity of the forecast, enabling a more localized and focused analysis of future energy requirements. This visualization serves as a powerful tool for stakeholders involved in energy planning and decision-making, offering a clear and accessible depiction of the forecasted energy demand trends across diverse geographical areas.



KEY ATTRIBUTES/ DIMENSIONS

A supplementary line plot establishes a correlation between the cumulative future energy demand and the cooling setpoint within each house. This visualization sheds light on the influence of variations in the cooling setpoint on energy consumption across different time categories. By examining the cumulative future energy demand in conjunction with the cooling setpoint, the plot offers valuable insights into the nuanced relationship between these variables. Users can discern how changes in the cooling setpoint, specific to each house, contribute to fluctuations in energy consumption across various time categories. The line plot serves as a dynamic and informative tool, enabling a deeper understanding of how adjustments in cooling setpoints impact the overall energy demand forecast. This visual exploration enhances the user's ability to grasp the intricate dynamics between cooling preferences and energy consumption patterns, providing a nuanced perspective for more informed decision-making and analysis.



SHINY APPLICATION DASHBOARD

Here also the data has been munged and transformed for the preparation of giving it as csv file in the shiny to represent all the aggregation.

```r
library(tidyverse)
thisdf <- read_csv("C:/Users/rithv/Downloads/converted1.csv")
second_parts <- str_split(thisdf$in.county_and_puma, ", ", simplify = TRUE)[, 2]
newsecond_parts <- unique(second_parts)
newcombined_county_puma <- unique(thisdf$in.county_and_puma)

excel_data <- read_excel("C:/Users/rithv/Downloads/exceldata1.xls")
merged_data <- inner_join(thisdf, excel_data, by = c("in.county" = "countyno"))
excel_data1 <- read_excel("C:/Users/rithv/Downloads/uscounties.xlsx")
merged_data1 <- inner_join(merged_data, excel_data1, by = c("`County Name` = county"))
merged_data1 <- inner_join(merged_data, excel_data1, by = c("County Name" = "county"))
result2 <- merge(merged_data, aggregated_excel_data, by.x = "County Name", by.y = "county", all.x = TRUE)

aggregated_excel_data <- excel_data1 %>%
  group_by(county) %>%   # Assuming 'County' is the column with county names
  summarise(state_name = first(state_name))  # Assuming 'State' is the column with state names
result2$county_name[your_dataframe$county_id == "G4500650"] <- "McCormick"

setwd("C:/Users/rithv/Documents/IDS Project")
write.csv(result2, file ="mergedstatescounties.csv",row.names = FALSE)
result2$state_name[result2$in.county == "G4500650"] <- "South Carolina"

result4 <- read.csv("C:/Users/rithv/Downloads/Final_Merged_For_Shiny_Geo.csv")


library(dplyr)
library(leaflet)

# Summarize data to find the time zone with maximum consumption for each house
summary_data <- result4 %>%
  group_by(bldg_id) %>%
  summarize(max_time_zone = time_split_numeric[which.max(out.total_energy_consumption)])

# Join summary data with the original dataset to retain all rows for each bldg_id
merged_data <- left_join(result4, summary_data, by = "bldg_id")
```

```r
library(tidyverse)
Merged_non_od_file <- read_csv("C:/Users/Soundarya Ravi/Desktop/Shiny/G,HI,J/Merged_data.csv")
Predicted_weather <- read_csv("C:/Users/Soundarya Ravi/Desktop/Shiny/G,HI,J/Predicted_final_weatherplus5.csv")
```

```r

subset_df <- Predicted_weather[, c("bldg_id","time_split_numeric",
 "Next_year_Pred", "Change_in_energy")]


```

```r
str(Merged_non_od_file$time_split)
# Assuming 'Merged_non_od_file' is your dataframe
# Assuming 'time_split' is the existing column you want to encode

# Create a mapping for encoding
time_split_mapping <- c("Late Night" = 1, "Early Morning" = 2, "Morning" = 3, "Noon" = 4, "Evening" = 5, "Night" = 6)

# Add the new column 'time_split_numeric' based on the encoding
Merged_non_od_file$time_split_numeric <- time_split_mapping[Merged_non_od_file$time_split]

# Display the first few rows of the updated dataframe
head(Merged_non_od_file)
```

```r
# Assuming 'Merged_non_od_file' and 'subset_df' are your dataframes
# Assuming 'bldg_id' and 'time_split_numeric' are the common columns

# Merge the dataframes based on 'bldg_id' and 'time_split_numeric'
merged_data <- merge(Merged_non_od_file, subset_df, by = c("bldg_id", "time_split_numeric"))

# Display the first few rows of the merged dataframe
head(merged_data)
```

Description: df [6 × 98]

	bldg_id <dbl>	time_split_numeric <dbl>	...1 <dbl>	in.county <chr>	time_split <chr>	in.county_and_puma <chr>	in.sqft <dbl>	in.bedrooms <dbl>
1	100015	1	5989	G4500510	Late Night	G4500510, G45001102	2176	3
2	100015	2	5990	G4500510	Early Morning	G4500510, G45001102	2176	3
3	100015	3	5991	G4500510	Morning	G4500510, G45001102	2176	3
4	100015	4	5992	G4500510	Noon	G4500510, G45001102	2176	3
5	100015	5	5993	G4500510	Evening	G4500510, G45001102	2176	3
6	100015	6	5994	G4500510	Night	G4500510, G45001102	2176	3

6 rows | 1-9 of 98 columns

```r
# Assuming 'merged_data' is your merged dataframe
# Assuming 'bldg_id' and 'time_split_numeric' are the columns to sort by

# Sort the dataframe based on 'bldg_id' and 'time_split_numeric'
sorted_data <- merged_data[order(merged_data$bldg_id, merged_data$time_split_numeric), ]

# Display the sorted dataframe
head(sorted_data)
```

```r
# Assuming 'sorted_data' is your dataframe

# Remove specified columns
columns_to_remove <- c("...1", "time_range.y")
sorted_data <- sorted_data[, !names(sorted_data) %in% columns_to_remove]

# Process the 'in.income' column to keep the maximum value in the range
sorted_data$in.income <- sapply(strsplit(sorted_data$in.income, "-"), function(x) as.numeric(x[2]))

# Display the modified dataframe
head(sorted_data)
```

Description: df [6 × 96]

	bldg_id <dbl>	time_split_numeric <dbl>	in.county <chr>	time_split <chr>	in.county_and_puma <chr>	in.sqft <dbl>
31891	65	1	G4500910	Late Night	G4500910, G45000502	885
31892	65	2	G4500910	Early Morning	G4500910, G45000502	885
31893	65	3	G4500910	Morning	G4500910, G45000502	885
31894	65	4	G4500910	Noon	G4500910, G45000502	885
31895	65	5	G4500910	Evening	G4500910, G45000502	885
31896	65	6	G4500910	Night	G4500910, G45000502	885

6 rows | 1-8 of 96 columns

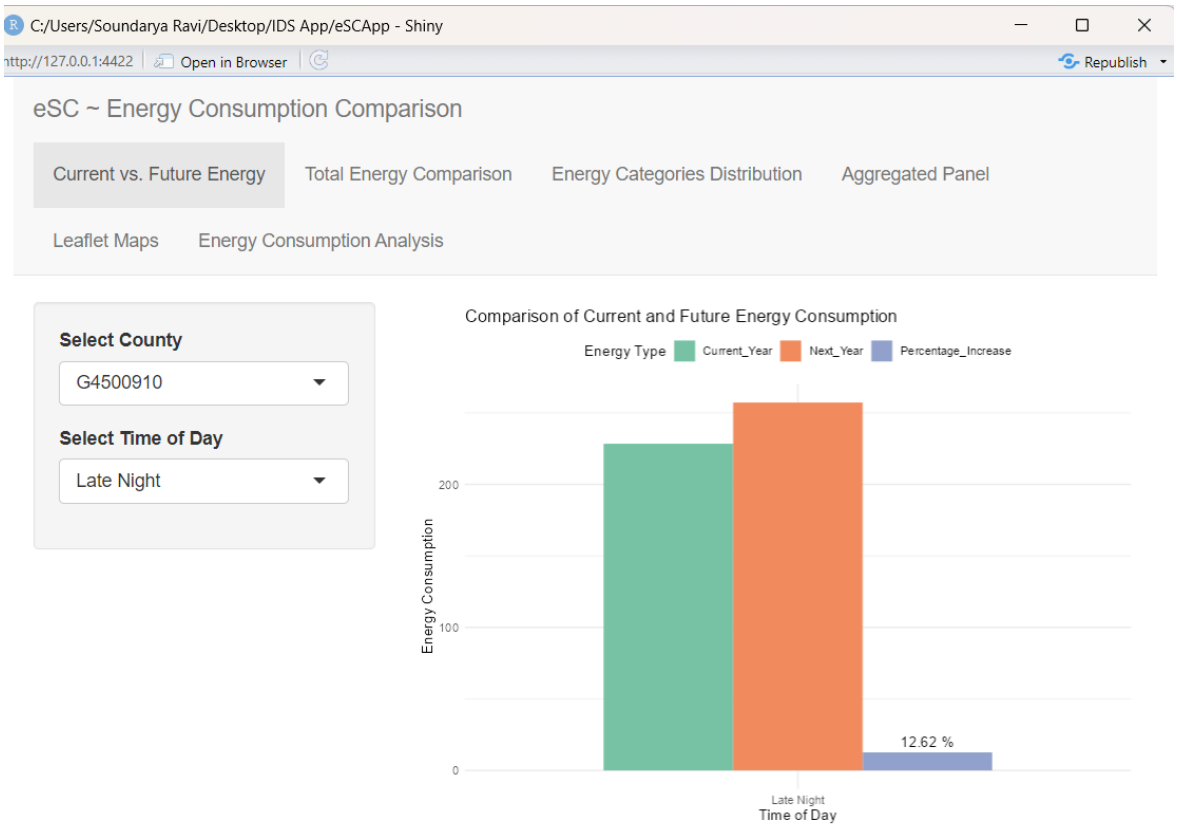
```r
setwd("C:/Users/Soundarya Ravi/Desktop/Shiny/G,HI,J/")
write.csv(sorted_data,"Final_Merged_For_Shiny_Geo.csv",row.names=FALSE)
```

```r
sorted_data <- read.csv("C:/Users/Soundarya Ravi/Desktop/Shiny/G,HI,J/Final_Merged_For_Shiny_Geo.csv")
```

Chunk 8

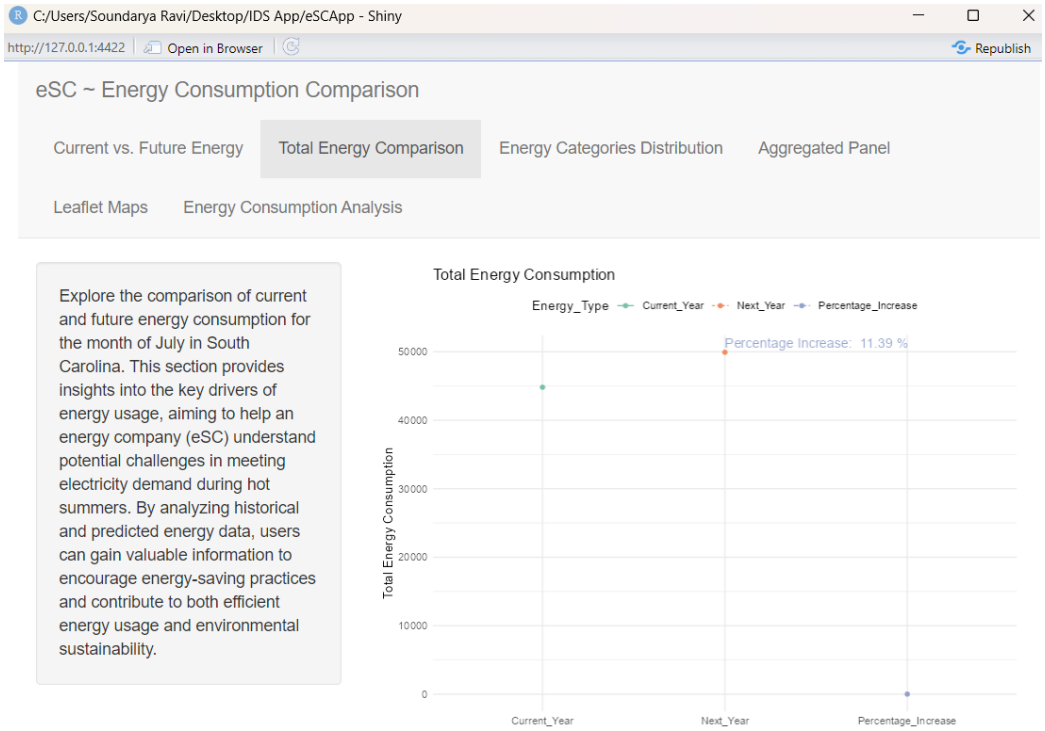
CURRENT VS. FUTURE ENERGY FOR EACH COUNTIES VS TIME OF THE DAY

In the Shiny application's inaugural tab panel, a thorough examination of current and anticipated energy consumption is undertaken for each distinct county. This scrutiny encompasses different times of the day, specifically late night, early morning, morning, noon, evening, and night. The comparative analysis is graphically represented through a bar graph, showcasing the percentage increase in energy consumption for each county across these distinct timeframes. This approach allows for a nuanced exploration of energy consumption patterns, considering the temporal variations associated with different parts of the day. Users can readily interpret the depicted percentage increases, gaining insights into how energy usage is expected to evolve during various time intervals. This comprehensive representation not only considers individual counties but also encapsulates the dynamic nature of energy consumption across the specified time frames, offering a detailed and accessible visualization of the anticipated changes in energy demand over the course of a day.



TOTAL CONSUMPTION CHART

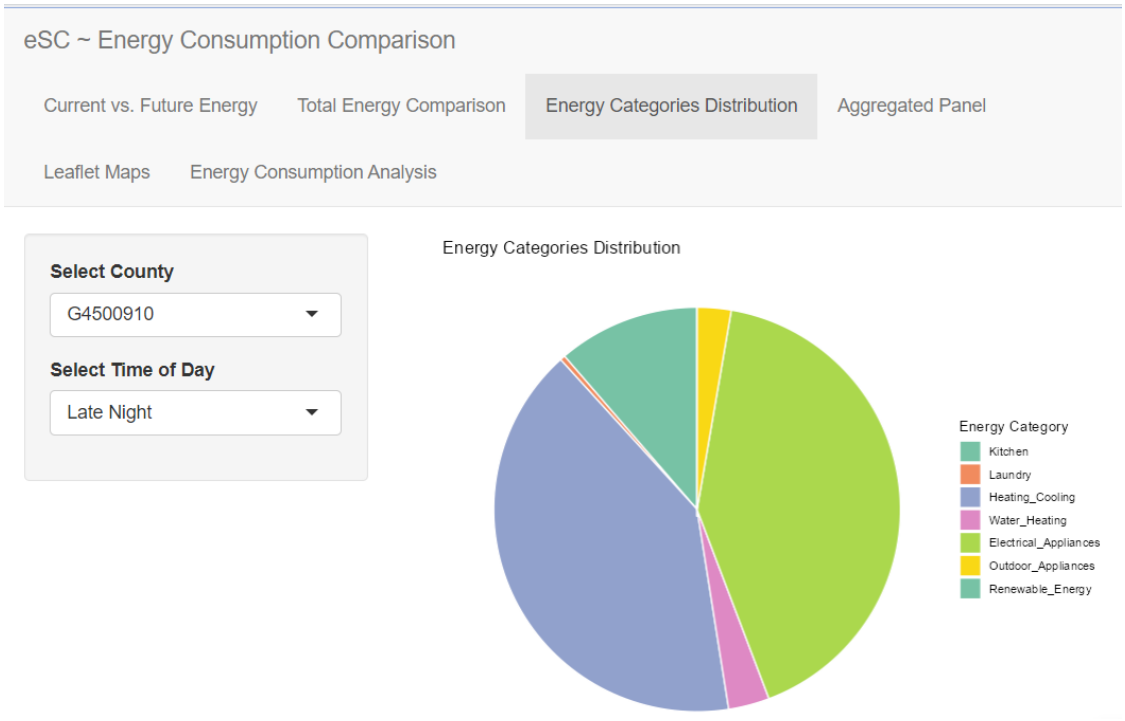
In the second tab panel of the Shiny application, a comprehensive Total Consumption Chart is presented. This chart juxtaposes the current year's total energy consumption with the predicted consumption for the next year. The visualization is enriched with the inclusion of the percentage increase, offering a clear comparison between the two periods. This tab provides users with an overarching view of the expected changes in total energy consumption, presenting not only the absolute values but also the percentage increase for enhanced interpretability. By encapsulating both the current state and the predicted future, users can easily grasp the scale of change in total energy consumption, fostering a more informed understanding of the anticipated trends and fluctuations in energy demand between the specified years.



ENERGY CATEGORIES DASHBOARD

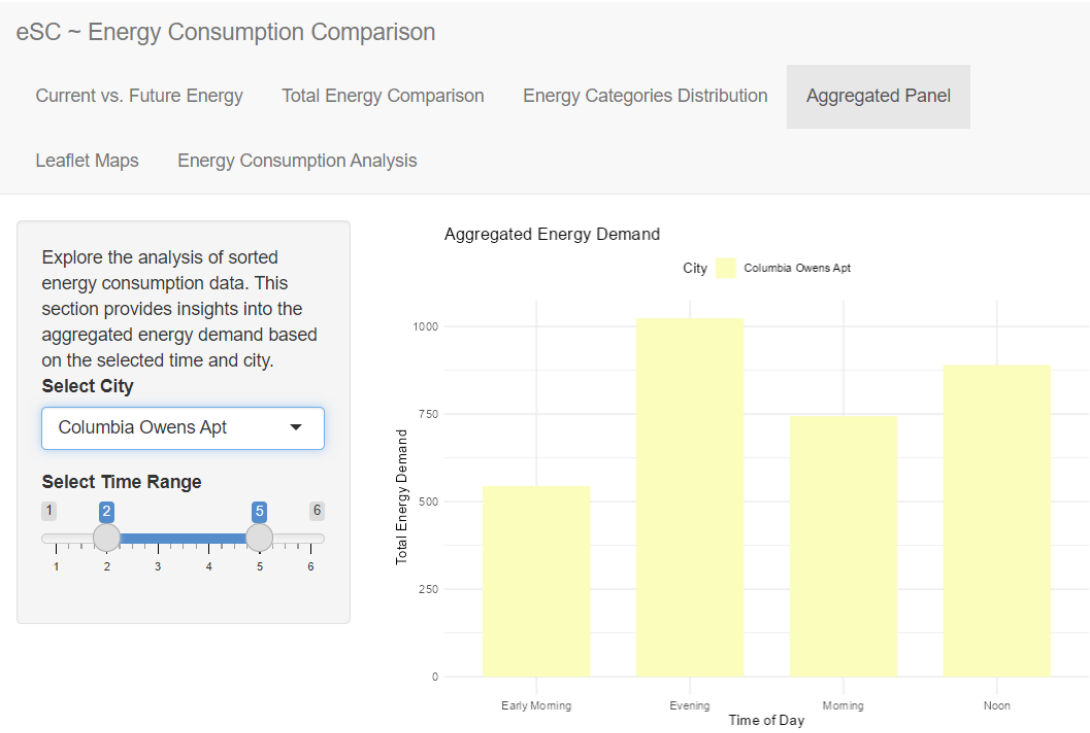
In the third tab panel, dedicated to the Energy Categories Dashboard, a detailed pie chart distribution is showcased, categorizing energy consumption into specific areas. The identified categories include kitchen, laundry, heating and cooling, electric appliances, outdoor appliances, water heating, and renewable energy. This visual representation offers insights into the proportional distribution of energy usage across these distinct categories. To enhance user customization and focus, filters for County IDs and specific times of the day have been integrated. This allows users to selectively examine and analyze the energy consumption patterns for various categories within specified counties and during different periods of the day. The inclusion of filters adds a dynamic element, empowering users to explore and understand the

nuanced details of energy consumption across different categories, making the Energy Categories Dashboard a valuable tool for comprehensive and targeted analysis.



Aggregated Energy Demand Per Weather File City

The fourth tab panel of the application, titled "Aggregated Energy Demand Per Weather File City," offers users an insightful perspective into energy demand patterns. The feature includes a city-wise aggregation of energy demand based on weather files, with the added convenience of a slider to filter results for a specific timeframe. Users can tailor their analysis by selecting a particular city and adjusting the slider to focus on distinct periods. This interactive functionality empowers users to explore and compare aggregated energy demand across different cities, facilitating a nuanced understanding of consumption trends influenced by varying weather conditions. The inclusion of these filters reflects a user-centric design, providing a flexible and tailored experience for examining energy demand dynamics in relation to specific weather files and chosen cities. This tab serves as a valuable tool for users seeking detailed insights into how energy consumption varies across cities and timeframes, contributing to a comprehensive understanding of the factors influencing energy demand within specific geographical and temporal contexts.



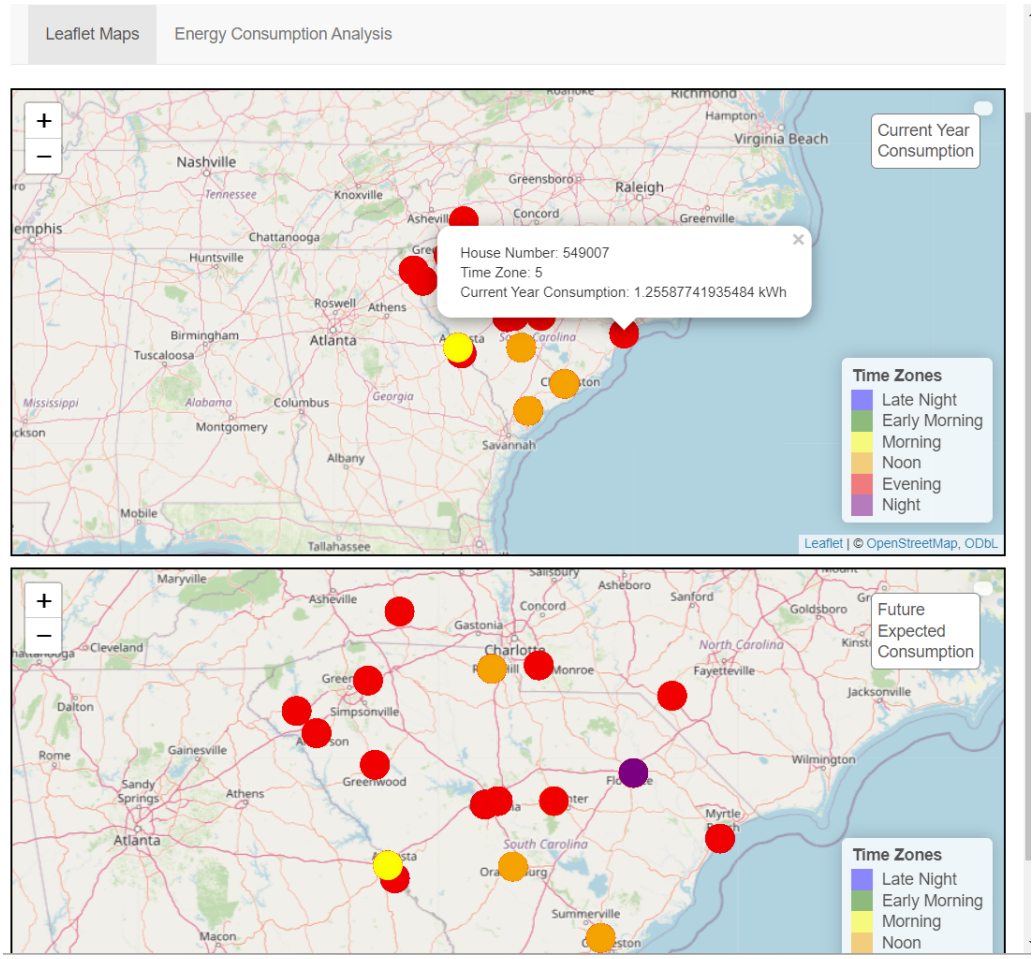
LEAFLET MAPS

In the Shiny dashboard's fifth tab panel, two Leaflet maps are juxtaposed, offering a comprehensive view of current and anticipated energy consumption trends for 5710 houses. The first map depicts the current energy consumption data for July 2018, marked by color-coded dots representing the timeframe with the highest energy usage averaged over the entire month. This visualization provides a spatial understanding of the predominant time of day when energy demand peaks across various locations.

Directly below the first map, the second Leaflet map unveils the future energy consumption predictions for July 2019. Like its predecessor, this map employs color-coded dots to signify the timeframe with the anticipated highest energy consumption averaged over the month. The side-by-side presentation facilitates a seamless comparison between current and future energy consumption patterns. Users can effortlessly discern spatial variations and temporal shifts in energy demand, identifying regions where consumption peaks are expected to differ between the two years.

The interactive nature of the Leaflet maps enhances user engagement, enabling exploration of geographical disparities and changes in energy consumption behavior over time. Users can navigate, zoom in, and hover over individual dots to access detailed information about specific houses. The dual-map setup provides a dynamic visual narrative, allowing users to glean insights into the evolving energy landscape and make informed assessments about the predicted changes in energy demand for July 2019 compared to the baseline data from July 2018.

Overall, this paired mapping approach serves as a powerful tool for users seeking a spatial-temporal understanding of energy consumption trends, offering a detailed and visually intuitive exploration of both historical and future patterns within the specified dataset.

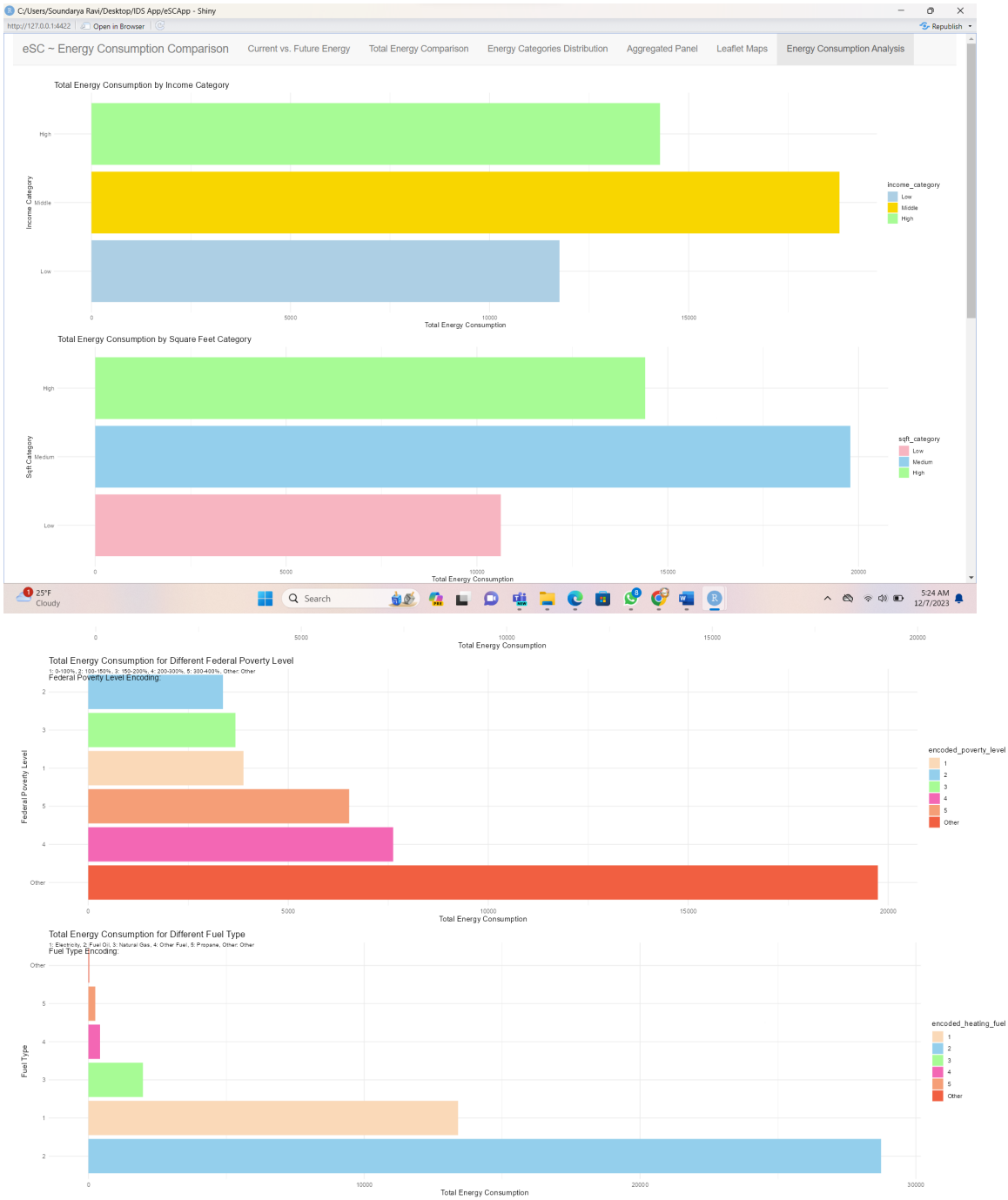


VISUALIZATIONS ON SIGNIFICANT PREDICTORS

In the conclusive tab of the Shiny application, a dedicated space is allocated for the visualization of significant predictors influencing energy consumption. The focus revolves around discerning patterns related to various predictor variables, such as income category, square footage, federal poverty level, and fuel type.

Users can explore and interpret the visual representations that highlight the relationship between each predictor and energy consumption. Specifically, the tab offers insights into which income category exhibits the highest energy consumption, the impact of square footage on energy usage, the correlation between federal poverty level and energy consumption, and the variations in consumption based on different fuel types.

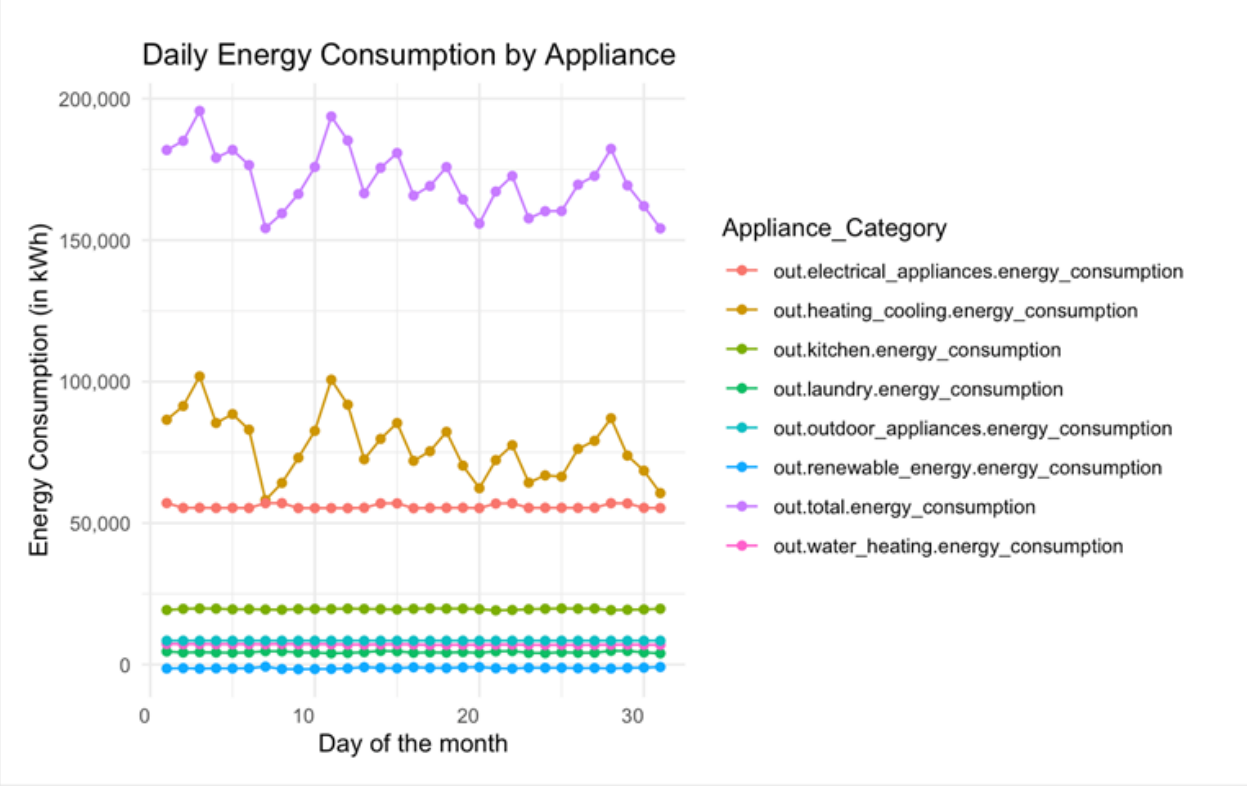
By presenting these visualizations, the tab aims to distill complex relationships and patterns, providing users with a clear and accessible means to understand the influence of these predictors on energy consumption. The interactive nature of the Shiny application allows users to dynamically engage with the visualizations, gaining actionable insights into the factors that contribute significantly to variations in energy usage. This final tab serves as a valuable tool for users seeking to identify and understand the key predictors that shape energy consumption patterns within the dataset, ultimately facilitating informed decision-making and analysis.



POTENTIAL APPROACH TO REDUCE PEAK DEMAND

WHAT IS THE APPROACH?

During the examination of correlation matrices and visualizations, we discerned prominent energy consumers within the dataset. As illustrated in the chart below, our analysis unveiled entities exhibiting substantial energy consumption. This insightful exploration facilitated the identification of key contributors to energy demand, enabling a nuanced understanding of the factors influencing consumption patterns. The visualization served as a valuable tool in highlighting significant entities, guiding strategic considerations for optimizing energy usage and fostering informed decision-making in the context of peak demand management.



The predominant consumers of energy were identified as heating and cooling systems. This observation is particularly notable considering the timeframe of July, where rising temperatures typically prompt increased utilization of air conditioning. The correlation between elevated temperatures and heightened demand for cooling solutions underscores the seasonality and climatic influence on energy consumption patterns. Recognizing heating and cooling systems as major contributors provides a targeted focus for energy management strategies, emphasizing the need for efficient and sustainable solutions to address peak demand during warmer months.

To address the challenge of high energy consumption, our proposed solution involves modifying the HVAC cooling efficiency type. We identified ten distinct types, ranging from high to low

efficiency, with SEER ratings and EER values providing insights into their energy performance. By strategically adjusting the HVAC cooling efficiency type, such as transitioning to higher SEER-rated systems or optimizing existing configurations, we aim to enhance overall energy efficiency and mitigate peak demand during periods of heightened temperatures. This proactive approach aligns with our commitment to sustainable energy practices, offering a tailored strategy to meet the specific needs of each air conditioning system within the scope of our analysis. Through this nuanced adjustment, we anticipate a tangible reduction in energy consumption, contributing to a more resilient and environmentally conscious energy infrastructure.

AC, SEER 15: High-efficiency air conditioning system with a SEER rating of 15.
AC, SEER 13: Similar to the first one but with a slightly lower SEER rating of 13.
None: Absence of a dedicated air conditioning system.
AC, SEER 10: Air conditioning system with a lower SEER rating of 10, indicating moderate efficiency.
Heat Pump: A type of HVAC system capable of providing both heating and cooling.
Room AC, EER 10.7: Room air conditioner with an EER (Energy Efficiency Ratio) of 10.7.
Room AC, EER 8.5: Similar to the previous one but with a lower EER rating of 8.5.
AC, SEER 8: Another air conditioning system with a lower SEER rating of 8.
Room AC, EER 9.8: Another room air conditioner with an EER of 9.8.
Room AC, EER 12.0: Another room air conditioner with a higher EER rating of 12.0.

To address the variability in HVAC cooling efficiency types across different houses, we opted to substitute the top five higher efficiency types within the ordinal code. Recognizing the practical constraints of standardizing every house to the same type, this strategic replacement aimed to introduce more energy-efficient configurations where feasible. Following this adjustment, we conducted model predictions to assess the resulting energy distribution. By integrating the modified HVAC efficiency types into the predictive modeling framework, we sought to gauge the impact on energy consumption patterns. This approach allows for a nuanced understanding of how specific HVAC configurations influence energy demand, enabling us to refine and optimize our strategies for peak demand management. Through these iterative processes, we aim to enhance the overall energy efficiency of the diverse housing units served by our system.

MODELING THE APPROACH

```
Ordinality_File <- read.csv("C:/Users/Soundarya Ravi/Desktop/Shiny/Team2_Final_SEW_Ordinal_Model
ling1.csv")
View(Ordinality_File)
```

```
columns_to_remove <- c("Next_year_Pred", "Change_in_energy")
sorted_data_for_IJ<- Ordinality_File[, !(names(Ordinality_File) %in% columns_to_remove)]
```

```
columns_to_drop <- c(
  "out.kitchen_energy_consumption",
  "out.laundry_energy_consumption",
  "out.heating_cooling_energy_consumption",
  "out.water_heating_energy_consumption",
  "out.electrical_appliances_energy_consumption",
  "out.renewable_energy_energy_consumption",
  "out.outdoor_appliances_energy_consumption"
)
sorted_data_for_IJ <- sorted_data_for_IJ[, !(names(sorted_data_for_IJ) %in% columns_to_drop)]
```

```
# Ordinal coding for in.hvac_cooling_efficiency
#in_hvac_cooling_efficiency_mapping <- c(
#  "AC, SEER 15"=1, "AC, SEER 13"=2, "None"=0, "AC, SEER 10"=4,
#  "Heat Pump"=5, "Room AC, EER 10.7"=6, "Room AC, EER 8.5"=7,
#  "AC, SEER 8"=8, "Room AC, EER 9.8"=9, "Room AC, EER 12.0"=10
#)
#static_house_filtered$in.hvac_cooling_efficiency <- #as.numeric(in_hvac_cooling_efficiency_mapp
ing[static_house_filtered$in.hvac_cooling_efficien#cy])

## Ordinal coding for in.cooling_setpoint (assuming the given order)
#in_cooling_setpoint_mapping <- c(
 # "72F"=7, "76F"=9, "70F"=6, "60F"=1, "78F"=10, "75F"=8, "68F"=5, "62F"=2, "65F"=3,
  #"80F"=11, "67F"=4
#)
```

```
sorted_data_for_IJ$Dry_Bulb_Temperature_C <- sorted_data_for_IJ$Dry_Bulb_Temperature_C + 5
```

```
set.seed(123)  # Setting seed for reproducibility
unique_bldg_ids <- unique(sorted_data_for_IJ$bldg_id)
unique_values <- c(10, 6, 5, 1, 2)

# Shuffle the unique values for randomness
shuffled_values <- sample(unique_values)

# Create a mapping between bldg_id and shuffled_values
value_mapping <- rep(shuffled_values, length.out = length(unique_bldg_ids))

# Assign the values to the in.hvac_cooling_efficiency column
sorted_data_for_IJ$in.hvac_cooling_efficiency <- value_mapping[match(sorted_data_for_IJ$bldg_id,
unique_bldg_ids)]
```

```
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# Test and Train Data
set.seed(0) # Set seed for generating random data.

# CreateDataPartition() function from the caret package to split the original dataset into a tra
ining and testing set
# Split data into training (80%) and testing set (20%)
parts <- createDataPartition(sorted_data_for_IJ$out.total_energy_consumption, p = 0.8, list = FA
LSE)
train <- sorted_data_for_IJ[parts,]
test <-  sorted_data_for_IJ[-parts,]

# Define predictor and response variables in the training set
train_x <- data.matrix(train[, -which(names(train) == "out.total_energy_consumption")])
train_y <- train[["out.total_energy_consumption"]]

# Define predictor and response variables in the testing set
test_x <- data.matrix(test[, -which(names(train) == "out.total_energy_consumption")])
test_y <- test[["out.total_energy_consumption"]]

print(c("Length of train_y:", length(train_y)))
```

```
# Check if lengths match before creating xgb.DMatrix
stopifnot(length(train_y) == nrow(train_x))

# Continue with the rest of your code...
#define final training and testing sets
xgb_train = xgb.DMatrix(data = train_x, label = train_y)
xgb_test = xgb.DMatrix(data = test_x, label = test_y)
```

```
#defining a watchlist
watchlist = list(train=xgb_train, test=xgb_test)

#fit XGBoost model and display training and testing data at each iteartion
model = xgb.train(data = xgb_train, max.depth = 3, watchlist=watchlist, nrounds = 100)
```

```
#use model to make predictions on test data
pred_y <-  predict(model, xgb_test)
pred_x <-  predict(model, xgb_train)
```

```
# Assuming pred_y is your predicted values

# Calculate Mean Squared Error (MSE)
mse <- mean((test_y - pred_y)^2)
cat('Mean Squared Error (MSE): ', round(mse, 3), '\n')
```

```
## Mean Squared Error (MSE):  0.058
```

```
# Calculate Root Mean Squared Error (RMSE) using caret package
rmse <- caret::RMSE(test_y, pred_y)
cat('Root Mean Squared Error (RMSE): ', round(rmse, 3), '\n')
```

```
## Root Mean Squared Error (RMSE):  0.241
```

```
# Calculate R-squared
y_test_mean <- mean(test_y)
tss <- sum((test_y - y_test_mean)^2)
rss <- sum((test_y - pred_y)^2)  # Using predicted values to calculate residuals
rsq <- 1 - (rss/tss)
cat('The R-squared of the test data is ', round(rsq, 3), '\n')
```

```
## The R-squared of the test data is  0.88
```

```
predictions_xgb <- predict(model, newdata = xgb_test)

mape <- mean(abs((test$out.total_energy_consumption - predictions_xgb) / test$out.total_energy_c
onsumption )) * 100

# Print the result
print(paste("MAPE:", mape))
```

```
## [1] "MAPE: 13.9243384816242"
```

PERCENTAGE CALCULATION

```
train$Possible_New_Energy <- pred_x
test$Possible_New_Energy <- pred_y
```

```
combined_df <- rbind(train,test)
```

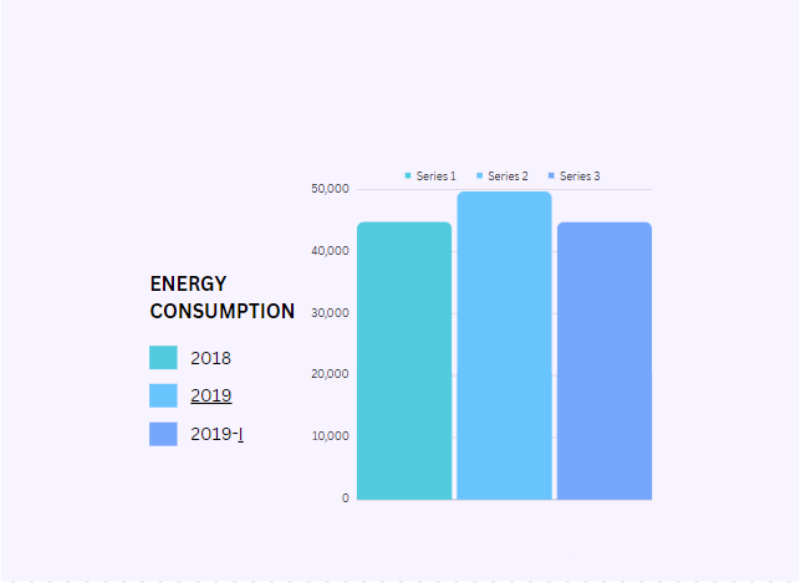
```
sorted_combined_df <- combined_df[order(combined_df$bldg_id, combined_df$time_split), ]
```

```
sorted_combined_df$New_Change <- sorted_combined_df$Possible_New_Energy - sorted_combined_df$ou
t.total_energy_consumption
```

```
Percentage_difference <- (sum(sorted_combined_df$New_Change)/sum(sorted_combined_df$out.total_en
ergy_consumption)) * 100
Percentage_difference
```

```
## [1] -0.05292819
```

There was a decrease in energy consumption. By replacing the HVAC cooling system, the future energy consumption will be 0.05% less than the current one.



WORK DONE BY EACH PERSON

SOUNDARYA RAVI

Data Merging and Cleanup:
In the initial stages of the project, I spearheaded the intricate process of **data merging**. My focus was on meticulous attention to detail, ensuring the seamless integration of disparate datasets. My responsibilities included **cleaning up the data**, addressing missing values, and implementing transformations to enhance overall quality and usability. This foundational work laid the groundwork for subsequent analyses.

Weather Data Management:
Taking charge of the weather data, I implemented a **robust time-split logic** and aggregated rows for comprehensive analysis. This process was pivotal for obtaining meaningful insights and ensuring the dataset's relevance to our modeling efforts. The emphasis was on creating a dataset conducive to extracting valuable patterns and trends.

Collaborative Teamwork:
Beyond technical tasks, I actively participated in **team meetings**, fostering an environment of open communication and collaboration. Coordination with team members, particularly with Subhiksha on **file management**, was critical to maintaining an organized project structure conducive to efficient workflows.

Static House Data Cleanup:
A critical aspect of my role involved cleaning up the **static house data file**. This encompassed handling NA values, addressing empty strings, and **applying transformations** to optimize the dataset for modeling purposes. The objective was to create a clean and robust foundation for subsequent analyses.

Model Development and Evaluation:
Moving to the modeling phase, I took on the challenge of developing and evaluating multiple models. **Linear regression, SVM, and XGBoost** were among the models I worked on. Embracing a learning curve, I delved into the intricacies of **XGBoost**, a powerful algorithm known for its efficiency in handling complex datasets. The exploration of the **Mean Absolute Percentage Error (MAPE)** concept further demonstrated my analytical prowess.

Temperature Increase Simulation:
A key highlight of my contributions was the simulation of temperature increases. I assessed the impact of elevated temperatures on energy consumption, providing valuable insights into potential scenarios and aiding in future decision-making.

Shiny Application Development:
The development of **Shiny application tab panels** was a significant part of my role, emphasizing a commitment to creating user-friendly visualizations. Panels such as "Current Vs. Future Energy," "Total Consumption," and the "Energy Categories Dashboard" were crafted to offer intuitive insights into complex energy data.

Spatial Visualization with Leaflet:
Collaborating with Archit and Rithvik, I contributed to the inclusion of a dynamic spatial visualization element in the application. This involved working on the **leaflet live map**, adding an interactive geographical layer to the data.

Significant Predictors Identification:
Engaging with Subhiksha, I contributed to identifying significant predictors, demonstrating effective teamwork in deciphering complex patterns within the dataset. The focus was on extracting meaningful insights that could inform decision-making.

Application Hosting:
Hosting the application marked the culmination of my efforts, ensuring accessibility for all stakeholders. This final step was crucial in making our findings and visualizations accessible and actionable for our client and other project stakeholders.

In essence, my multifaceted contributions encompassed technical expertise, effective collaboration, and a dedication to delivering insights aligned with the project's objectives. This comprehensive approach reflects my commitment to professionalism and excellence in the field of data analytics. And also worked on drafting this report.

SUBHIKSHA MURUGESAN

Data Consolidation and Transformation

I played a crucial role in integrating diverse datasets from AWS cloud, successfully navigating the complexities of data merging. This involved selecting relevant files, merging them based on commonalities, and overcoming challenges due to the extensive size of the datasets.
I also focused on preparing the data for modeling, undertaking a comprehensive data cleaning process. This included removing NA columns and transforming character columns into numerical ones.

Energy data Management
I undertook the significant task of downloading energy data for all houses and merging them into one data frame and consolidating 43 output variables related to energy consumption to 7 categories. This step was crucial in distilling the data into a more manageable and interpretable form.

Exploratory Data Analysis and Visualization
I was responsible for conducting Exploratory Data Analysis (EDA), crucial to understanding underlying data patterns and behaviors. I created various visualizations, including line plots for daily energy consumption across different appliance categories in South Carolina homes, and total energy consumption for each day in July. Additionally, I developed a map visualization to depict total energy consumption by city in South Carolina.

Predictive Analysis
I also focused on predicting future energy demand, creating line plots that illustrated peak future energy demand across different geographic regions and based on the number of occupants in each house. This predictive analysis provided valuable foresight into future energy consumption patterns.
Throughout the project, collaboration was key. My ability to work effectively with the team, especially in integrating different aspects of the project, was instrumental in its success.

ARCHIT DILIP DUKHANDE

Machine Learning Algorithm Experimentation :
In this phase, I conducted a comprehensive examination of various machine learning algorithms to determine their accuracies and applicability. The primary goal was to identify the algorithm providing the highest accuracy and generating the most plausible figures. The selected algorithms for experimentation were Linear Regression, Support Vector Machines (SVM), and Decision Trees (Rpart). Then dropped the Rpart as the efficiency was very low comparatively.

Experimentation :
I implemented an 80-20 train-test split to train models on a subset and evaluate performance on an independent subset, setting a random seed for reproducibility. The experimentation involved leveraging various R libraries, such as caret and e1071, for effective model training.

Results :
Linear Regression emerged as the most feasible model, producing an impressive ~70% accuracy. In contrast, SVM demonstrated low accuracy and yielded high p-values, and Decision Trees exhibited low accuracy, accompanied by a borderline p-value.

Key Findings :
Despite its simplicity, Linear Regression showcased surprisingly high accuracy compared to more complex models. In contrast, SVM, Rpart, and Random Forest underperformed, potentially due to overfitting or insufficient data complexity.

Model Selection :
Given the superior performance of Linear Regression on the test set, it has been selected as the model of choice for the cleaned final energy dataset.

Leaflet Maps :
Teaming up with Rithvik, we've created a captivating Shiny map that delves into the energy consumption patterns of a specific building during the current month of July. This dynamic visualization not only vividly illustrates the historical peaks in energy usage across different time zones but also offers a forward-looking perspective by predicting where the building is likely to experience its highest energy consumption in the upcoming July. Utilizing distinct markers and color-coded elements, the map provides an intuitive representation, enhancing our understanding of past trends and enabling proactive planning for potential peak consumption times in the future. Our collaborative effort aims to deliver a visually engaging and insightful tool for effortlessly analyzing and forecasting energy consumption trends.

ADITYA PAWAR

Data Preparation:
The initial phase of our collaborative efforts involved the entire team working together, with my focus on cleaning, understanding, and manipulating the data. I developed multiple functions to iterate through all the house files, incorporating them into a loop. The challenges we encountered during dataset merging, stemming from the substantial volume of data, led us through numerous trial-and-error situations. The function I developed entailed extracting the entire file, narrowing down the rows to those specifically from July, and then trimming down the relevant columns. The decision on which columns to include was dynamic, reflecting the uncertainty about their ultimate utilization. Prior to data manipulation, the team convened to discuss essential columns, table joining, potential methods for reducing energy consumption, and other analyses to be

conducted with the data. My background in mechanical and electrical engineering concepts played a pivotal role in this collaborative decision-making process.

Exploratory Data Analysis:
During the data analysis phase, I focused on creating a diverse array of graphs, plots, and visualizations to extract meaningful insights from the data. Each visualization was shared with the project group, fostering discussions and garnering interpretations from every team member. This iterative process helped identify areas of the dataset that required further exploration. Detailed notes were taken during discussions about the relationships depicted in the graphs, whether they were directly proportional, inversely proportional, or related in other ways. These notes provided valuable insights into what the predictions might look like and helped us understand the dataset's characteristics, including the presence of outliers and the data generation process for the month of July. Our toolkit for exploratory data analysis encompassed bar graphs, pie charts, line graphs, feature engineering, Principal Component Analysis (PCA), correlation matrices, heatmaps, and other conventional EDA processes, such as checking the mode, median, and average. Additionally, exploring the InterQuartile Range of specific columns enhanced our understanding of closely related variables.

Communication:
Effective communication and the ability to articulate thoughts within the team were crucial skills demonstrated by all team members, contributing to a seamless flow of the project. This collaborative communication ensured that insights and knowledge were shared efficiently, facilitating informed decision-making throughout the data preparation and analysis phases.

RITHVIK RANGARAJ

Data Standardization:
One of the critical tasks I undertook was to standardize the dataset as well as other data I pulled online such as the US state & county dataset. I focused on standardizing the data by removing inconsistencies in capitalization, ensuring that there is a successful join operation through a match.

Data integration and refinement:
I merged our final dataset with a state and county dataset leveraged from the web, using crucial characteristics such as county_id to match counties precisely. This stage was essential for mapping geographical data so that we could examine patterns in energy use based on these locations.

Dynamic Visualization Techniques:
Utilizing advanced mapping libraries such as leaflet, I created a dynamic map that showcased key insights such as current and future energy consumption. Each data point on the map was color-coded to represent the predominant time zone associated with the highest energy consumption for a particular house number. This visualization provided a clear and intuitive understanding of energy consumption patterns across different time periods within specific locations. There was an option to see current as well as predicted future energy consumption values.

CONCLUSION

Our comprehensive approach to energy management unfolds across several key phases, beginning with robust data preparation and concluding with actionable insights derived from sophisticated modeling techniques. The journey encompasses:

Data Preparation Excellence:
The foundation is laid with a meticulous determination of the optimal approach for data reading and merging, ensuring a seamless and comprehensive dataset. This phase establishes the groundwork for subsequent analyses, emphasizing the importance of well-prepared data for meaningful outcomes.

Exploratory Analysis Illumination:
Our exploratory analysis illuminates crucial insights into energy consumption patterns, providing a basis for informed decision-making. The discernment of significant energy consumers, particularly heating and cooling systems, is instrumental in understanding seasonal and climatic influences on demand.

Strategic Modeling and Accuracy Evaluation:
Building predictive models becomes a cornerstone, involving a thoughtful selection from various models to identify the most effective one. We delve into the intricacies of accuracy evaluation, employing metrics such as Mean Absolute Percentage Error (MAPE) to quantify the efficacy of our models accurately.

Temperature Simulation and Peak Demand Assessment:
A forward-looking simulation involves creating a new weather dataset with elevated July temperatures. Leveraging our best model, we assess peak future energy demand under the assumption of no new customers. This model-driven approach ensures a nuanced understanding of future energy needs.

Geographic and Attribute-Specific Analysis:
Our analysis extends to showcase future peak energy demand across different geographic regions and essential attributes. This granular exploration aids in tailoring energy management strategies to specific contexts, ensuring adaptability to diverse needs.

Shiny Application for Interactive Understanding:
The culmination of our efforts results in a Shiny application, providing an interactive platform for clients to comprehend our model's energy predictions and understand potential future energy needs. The application empowers users to explore and interpret data-driven insights seamlessly.

Peak Demand Reduction Strategy:
Identifying a potential approach to reduce peak energy demand takes center stage. Our data-driven recommendation involves modifying HVAC cooling efficiency types strategically. By addressing the variability across different houses, we replace the top five higher efficiency types, expecting a tangible reduction in energy consumption.

Iterative Refinement and Optimization:
The iterative process of model predictions post-adjustment allows us to gauge the impact on energy consumption patterns. This refinement ensures that our strategies for peak demand management align with the diverse HVAC configurations, fostering an overall enhancement in energy efficiency.

Future-Forward Recommendations:
Beyond HVAC modifications, we propose user-driven changes, such as adjusting cooling setpoints and transitioning from electric to gas cooking. These considerations aim to empower users to contribute actively to energy conservation.

In essence, our data-driven approach, encompassing meticulous preparation, insightful modeling, and strategic recommendations, positions us at the forefront of sustainable energy management. The journey is marked by a commitment to excellence, adaptability, and a keen understanding of the dynamic interplay between data and actionable insights.

ANSWERING THE FINAL BUSINESS QUESTIONS

1. What potential approach was identified to reduce peak energy demand, and how was this approach derived from the data?

Our approach to reducing peak energy demand revolves around strategically modifying the HVAC cooling efficiency type, a concept derived from a meticulous examination of correlation matrices and insightful visualizations. By identifying prominent energy consumers, particularly heating and cooling systems, we honed in on the critical contributors to energy demand during warmer months. Leveraging data on SEER ratings and EER values, we proposed adjusting HVAC configurations to higher efficiency types. This proactive approach stems from a nuanced understanding of energy consumption patterns, offering a tailored strategy to enhance overall efficiency and mitigate peak demand. The selected approach is grounded in data-driven insights, providing a targeted and sustainable solution to address the specific needs of each air conditioning system within our analysis.

2. How would the impact of this approach be modeled, and what data-driven explanations would be provided to stakeholders regarding its effectiveness?

To model the impact of our approach, we substituted the top five higher efficiency HVAC types within the ordinal code, accounting for practical constraints in standardizing every house to the same type. This strategic replacement aimed to introduce more energy-efficient configurations where feasible. Following this adjustment, we conducted model predictions to assess the resulting energy distribution. By integrating the modified HVAC efficiency types into our predictive modeling framework, we gained a nuanced understanding of how specific configurations influence energy demand. Our data-driven explanations focus on demonstrating the impact of these changes on consumption patterns, providing stakeholders with clear insights into the effectiveness of our strategy for peak demand management. Through iterative processes and continuous refinement, we aim to enhance overall energy efficiency across diverse housing units.

3. What is the anticipated cost of implementing the identified peak energy demand reduction strategy?

The cost of implementing our strategy involves considerations such as retrofitting HVAC systems, acquiring more energy-efficient units, and potential adjustments to existing infrastructure. While the exact costs depend on various factors, including the scale of implementation and the current state of HVAC systems, we anticipate that initial investments will be offset by long-term energy savings. The cost estimation process involves a detailed analysis of equipment and installation expenses, ensuring transparency in communicating financial implications to stakeholders.

4. How does this cost compare to the expected benefits, and what is the projected return on investment?

Our cost-benefit analysis includes a thorough examination of both short-term implementation costs and long-term benefits in terms of reduced energy consumption. By comparing the anticipated costs with expected energy savings, we aim to provide stakeholders with a clear understanding of the return on investment. The projected return on investment serves as a key metric in demonstrating the economic viability and sustainability of our peak demand reduction strategy.

5. How sustainable is the chosen peak demand reduction strategy over the long term?

Our peak demand reduction strategy prioritizes sustainability by targeting HVAC efficiency improvements. The strategy's long-term sustainability is reinforced by its ability to adapt to

evolving energy landscapes and technological advancements. Continuous monitoring, data-driven insights, and flexibility in implementation contribute to the strategy's resilience, ensuring its relevance and effectiveness in the face of changing energy dynamics.

6. What measures are in place to adapt the strategy to evolving energy landscapes and technological advancements?

Adaptability is at the core of our strategy, with ongoing monitoring and feedback loops enabling adjustments based on emerging energy trends and technological advancements. Regular updates to HVAC efficiency types, informed by the latest advancements, ensure that our strategy remains at the forefront of sustainable energy practices. Stakeholder engagement and collaboration also play a crucial role in refining the strategy over time, fostering a dynamic approach to peak demand reduction aligned with the evolving energy ecosystem.

OTHER INTEGRATED SOLUTIONS FOR SUSTAINABLE ENERGY MANAGEMENT: DATA-DRIVEN APPROACHES TO PEAK DEMAND REDUCTION

1. Solar and Natural Gas Integration:
- **Approach:** Explore the integration of solar energy systems and natural gas for heating purposes within the housing units. Assess the feasibility of incorporating solar panels and leveraging natural gas as an alternative energy source.
- **Data-Driven Explanation:** Utilize historical energy consumption patterns to identify the potential impact of solar and natural gas integration on overall energy demand. Showcase the environmental benefits and cost-effectiveness through data-driven insights.

2. Cooking Type Transition to Gas:
- **Approach:** Encourage a shift from electric to gas-based cooking appliances within households. Analyze the energy consumption differences between electric and gas cooking and strategize the transition based on efficiency gains.
- **Data-Driven Explanation:** Utilize data on cooking energy usage to demonstrate the efficiency of gas-based cooking, highlighting the potential reduction in electricity demand during peak periods.

3. Ventilation Hours Optimization:
- **Approach:** Review and optimize the ventilation hours for each house based on occupancy patterns and air quality requirements. Consider adjusting ventilation systems to operate efficiently during periods of higher energy demand.

- **Data-Driven Explanation:** Utilize occupancy data and indoor air quality metrics to illustrate the impact of ventilation hours on energy consumption. Showcase the potential for energy savings without compromising indoor air quality.

4. Cooling Set Point Optimization:

- **Approach:** Optimize the cooling set points for air conditioning systems to ensure a balance between occupant comfort and energy efficiency. Adjust the set points based on external temperature variations during the hot July period.
- **Data-Driven Explanation:** Analyze temperature data and cooling system performance to demonstrate the correlation between cooling set points and energy usage. Showcase the potential for maintaining comfort while minimizing energy consumption.

5. Lighting System Upgrades:

- **Approach:** Consider upgrading lighting systems to energy-efficient LED or smart lighting technologies. Evaluate the energy savings achieved through such upgrades.
- **Data-Driven Explanation:** Utilize lighting usage data to illustrate the potential reduction in electricity demand by transitioning to more energy-efficient lighting systems. Showcase the long-term benefits in terms of both energy and cost savings.

6. Smart Thermostat Implementation:

- **Approach:** Integrate smart thermostats that can adapt to occupants' preferences and external temperature conditions. Explore the use of machine learning algorithms to optimize heating and cooling schedules.
- **Data-Driven Explanation:** Showcase the potential energy savings by comparing the performance of smart thermostats with traditional ones. Highlight the adaptability and efficiency gains through data-driven insights.

KEY CHALLENGES AND ISSUES

Data Quality and Integration:

- Managing missing or inconsistent data
- Integrating diverse data sources

Model Complexity and Selection:

- Determining optimal model complexity

- Selecting the most suitable model

Volume and Scalability:

- Managing and processing a large volume of data efficiently, ensuring scalability in the exploration phase.

Computational Resources:

- Ensuring access to sufficient computational resources to handle the extensive dataset and perform complex exploratory analyses.

Data Variety:

- Handling diverse data types and structures within the large dataset, including text, numerical, and categorical variables.

Data Cleaning Challenges:

- Identifying and addressing data quality issues, such as missing values, outliers, and inconsistencies, at the exploratory stage.

Exploratory Analysis Time Frame:

- Balancing the need for thorough exploration with time constraints, aiming for meaningful insights within a reasonable timeframe.

Visualization Complexity:

- Creating clear and informative visualizations from a vast dataset to aid in understanding patterns and trends.

----------APPENDIX----------

TEAM MEMBER	*WORK ACCOMPLISHED*
SOUNDARYA RAVI	- Data Merging - Data Cleanup - Weather Data Management - Model Finalling - File Management - Temperature Increase Simulation - Future Predictive Modelling - Shiny Application Development - Documentation
SUBHIKSHA MURUGESAN	- Data Consolidation and Transformation - Energy Data Management - Visualizations - EDA - Predictive Analysis
ARCHIT DILIP DUKHANDE	- Algorithm Experimentation - Initial Modeling - Leaflet Maps - Geo Analysis Coding - Initial Document Drafting
ADITYA PAWAR	- Data Preparation - EDA ~ Preliminary & Advanced Visualizations - Initial Function Loops - Data Cleanup
RITHVIK RANGARAJ	- Algorithm Experiments - Modelling Analysis - Data Standardization - Leaflet Markup - Geo Analysis Coding - Initial Document Drafting