Introduction to Data Science
IST687

Final Project Report

Submitted by

Bhavika Karale
Net ID: bmkarale
Abhishekh Shinde
Net ID: ashinde
Goutham Sri Vishwesh Bikkumalla
Net ID: gbikkuma
Prasanna Lakshmi Chelliboyina
Net ID: pchellib
Minnie McMillian
Net ID: msmcmill

Under the guidance of

Prof. Anderson, Prof. Saltz
Professor

Fall 2023

Contents

1 Project Overview

1.1 Goal

Create a machine learning model to predict the variation in energy usage caused due to global warming and propose ways to reduce the energy consumption.

1.2 Description

eSC energy company provides energy supply to residential properties in South Carolina. With concern of impact of global warming on energy demand, blackouts due to high demand of energy, it is essential to propose a business method which might create awareness about energy usage and energy saving and hence, reduce the total energy usage among the resident. eSC aims to avoid building a new power plant to meet the increasing energy demand and encourage the users to reduce(save) the energy usage.

The focus is to reduce energy usage in the month of July, when the temperatures have risen and the energy demand shows high upward trend.

2 Requirements

2.1 Business Requirements

BR1 **Reduce energy usage**: Propose methods to help eSC create energy usage awareness among the residential owners in South Carolina.

BR2 **Predict future energy usage**: Predict the future energy usage in the region to help eSC prepare for increased energy supply. This will prevent blackouts and help eSC plan in advance.

BR3 **Identify region with peak energy usage**: Identify region having the maximum energy usage. Identify the reasons for high energy usage.

2.2 Technical Requirements

BR1 **Ensure energy supply availability:** The energy supply must be available for 99.9%. Study the energy usage patterns to propose the anticipated energy supply.

BR2 **Predict energy demand** Predict the energy usage for summer of next year to help eSC plan ahead of time

BR3 **Reduce overhead cost** Propose methods to reduce energy usage to help eSC avoiding cost of building a new facility to meet the energy demands.

3 General Overview

3.1 Data Provided

1. **Static House Data:**
   - (a) Description: Contains data on about 5,000 single-family homes that use eSC energy.
   - (b) Attributes: Contains house attributes such as the building ID, house size, and other static details. The file is saved in the 'parquet' format, which is optimal for storage.
2. **Energy Usage Data:**
   - (a) Description: Provides hourly energy consumption statistics for each residence in the Static residence Data.
   - (b) Dataset Structure: Each residence has one dataset file that contains calibrated and validated energy usage with 1-hour load profiles.
   - (c) Data Variety: Describes each house's energy usage per hour from numerous sources (e.g., air conditioning system, dryer).
3. **Meta Data:**
   - (a) Description: A data description file that describes the fields used in the various housing data files.
   - (b) Format: A simple, human-readable CSV file with attribute descriptions.
4. **Weather Data:**
   - (a) Description: Hourly weather data, with one file for each geographic area (county).
   - (b) Data: Time-series weather data is gathered and kept based on a county code.
   - (c) County Code Reference: Each house's county code is identified by the 'in.county' column in the Static House Data.
   - (d) Format: The file is stored in a CSV format which is easy to use.

3.2 Tasks and Deliverables

1. **Data Preparation:** Determine a method for reading and merging the dataset.
2. **Exploratory Analysis:** Gain insights on the merged data. Clean the data and make it effective and usable.
3. **Model Building:** Build a model to predict the energy usage in a particular region and by hour.
4. **Demand Analysis:** Assume high temperatures, daily usage and seasonal variations in temperature to predict the future demand.
5. **Demand reduction strategy:** Propose strategies to enable reduction in demand based on the analysis made. This should help eSC in reducing their overhead costs.
6. **Shiny Applications:** Develop an interative application to present to the CEO of eSC.

4 Detailed Overview

4.1 Data Preparation

1. Data Cleaning for Static House Info:
   (a) The code reads a Parquet file containing static house information
   (b) It removes specific columns defined in columns_to_remove.
   (c) The resulting dataset is stored in static_house_info.
2. Calculating Total Energy for Each Building:
   (a) The code initializes an empty data frame (result_df_daywise) to store building-wise total energy.
   (b) It iterates over each building, reads its Parquet file, and calculates total energy for July.
   (c) The results are appended to result_df_daywise.
3. Processing Weather Data for July:
   (a) The code reads weather data for different counties in July.
   (b) It calculates median values for various weather variables.
   (c) It updates corresponding columns in static_house_info with the calculated median values.
4. Creating the Final Output Dataframe:
   (a) A final output dataframe (merge_static_house_info_df4) is created by selecting specific columns from static_house_info.
5. Merging Dataframes:
   (a) The code attempts to merge
      i. static_house_info (original dataset)
      ii. result_df_datewise (energy dataset daywise)
      iii. weather_final (weather dataset including date and county)
      iv. static_house_info_df1 (merging of static_house_info and result_df_datewise)
      v. merge_static_house_info_df (merging of static_house_info_df1 and weather_final including date and county)
      vi. merge_static_house_info_df4 (dataset after cleaning the original one)
   (b) A left join is performed using the merge function and later using the left_join function from the dplyr package.
   (c) Columns not necessary for further analysis are removed from the merged dataframe.

```r
# Create an empty data frame to store the row sums
result_df_daywise <- data.frame(building_id = character(), day_total_energy = numeric(), date = as.Date(character()))

for (i in 1:nrow(static_house_info)) {
  print(i)
  # Read Parquet file from a URL and create a data frame
  x <- data.frame(read_parquet(sprintf("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/%s.parquet", static_house_info$bldg_id[i])))
  x$time <- as.Date(x$time)

  # Subset data for July
  july_data <- x[format(x$time, "%m") == "07", ]

  # Calculate row sums for each day in July
  daily_sums_july <- tapply(rowSums(july_data[, 1:42], na.rm = TRUE), as.Date(july_data$time), sum, na.rm = TRUE)

  # Create a data frame with building_id, day_total_energy, and date
  daily_result_df_july <- data.frame(
    building_id = static_house_info$bldg_id[i],
    day_total_energy = daily_sums_july,
    date = names(daily_sums_july)
  )

  # Append results to the new data frame
  result_df_daywise <- rbind(result_df_daywise, daily_result_df_july)
}

# Print the resulting data frame
print(result_df_daywise)
```

```r
library(dplyr)

generate_weather_data <- function(temperature_var, humidity_var, wind_speed_var, wind_direction_var,
                                  global_radiation_var, direct_radiation_var, diffuse_radiation_var) {
  unique_counties <- unique(static_house_info$in.county)
  weather_data <- tibble(
    ‘Temperature [ C ]‘ = numeric(),
    ‘Humidity [%]‘ = numeric(),
    `Wind Speed [m/s]` = numeric(),
    `Wind Direction [Deg]` = numeric(),
    `Global Radiation [W/m2]` = numeric(),
    `Direct Radiation [W/m2]` = numeric(),
    `Diffuse Radiation [W/m2]` = numeric(),
    in.county = character()
  )
  for (county in unique_counties) {
    weather <- read_csv(paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/weather/2023-weather-data/", county, ".csv")) %>%
      select(date_time, !!temperature_var, !!humidity_var, !!wind_speed_var, !!wind_direction_var, !!global_radiation_var, !!direct_radiation_var, !!diffuse_radiation_var) %>%
      filter(date_time >= as.Date("2018-07-01"), date_time <= as.Date("2018-07-31")) %>%
      mutate(in.county = county)
    weather_data <- bind_rows(weather_data, weather)
  }
  # Print the resulting weather data frame
  # print(weather)
  weather_final_df <- weather_data
  # Convert ’date_time’ to just a Date object to remove the time part
  weather_final_df$date_time <- as.Date(weather_final_df$date_time, format = "%Y-%m-%d %H:%M:%S")
  # Now you can group by ’date_time’ and ’county_id’ and calculate the mean for the other columns
  weather_final_df <- weather_final_df %>%
    group_by(in.county, date_time) %>%
    summarise(
      median_Direct_Radiation = median(!!direct_radiation_var, na.rm = TRUE),
      median_Diffuse_Radiation = median(!!diffuse_radiation_var, na.rm = TRUE),
      median_Temperature = median(!!temperature_var, na.rm = TRUE),
      median_Humidity = median(!!humidity_var, na.rm = TRUE),
      median_Wind_Speed = median(!!wind_speed_var, na.rm = TRUE),
      median_Wind_Direction = median(!!wind_direction_var, na.rm = TRUE),
      median_Global_Radiation = median(!!global_radiation_var, na.rm = TRUE)
    )
  return(weather_final_df)
}

weather_final_df <- generate_weather_data(
  temperature_var = quote(`Dry Bulb Temperature [ C ]`),
  humidity_var = quote(`Relative Humidity [%]`),
  wind_speed_var = quote(`Wind Speed [m/s]`),
  wind_direction_var = quote(`Wind Direction [Deg]`),
  global_radiation_var = quote(`Global Horizontal Radiation [W/m2]`),
  direct_radiation_var = quote(`Direct Normal Radiation [W/m2]`),
  diffuse_radiation_var = quote(`Diffuse Horizontal Radiation [W/m2]`)
)
```

4.2 Exploratory Analysis

1. Understand the Data:

   (a) Using functions like str(), head(), summary(), and dim(), examine the datasets' structure.

   (b) Determine the variables' types—numerical, categorical, date/time, etc.—and look for any missing data.

2. Descriptive Statistics:

   (a) Use summary() to compute fundamental statistics for numerical variables.

   (b) Use density plots, box plots, or histograms to examine the distribution of numerical variables.

3. Categorical data:

   (a) Use frequency tables, bar charts, or pie charts to examine the distribution of categorical data.

   (b) Look for uncommon or special categories.

4. Correlation Analysis:

   (a) Use scatter plots or correlation matrices to examine correlations between numerical variables.

   (b) Boxplots are a useful tool for visualizing correlation.

5. Data Cleaning:

   (a) Use imputation or removal to address missing values.

   (b) Examine for abnormalities and outliers; using your domain knowledge, determine whether to keep or discard them.

4.3 Model Building

1. Data Splitting:

   (a) The createDataPartition function is used to divide the dataset into training (80%) and testing (20%) sets.

2. Managing Categorical Variables:

   (a) The character columns distinct values are found.

   (b) Only those rows in the test data that have values that match the unique values in the training data are included.

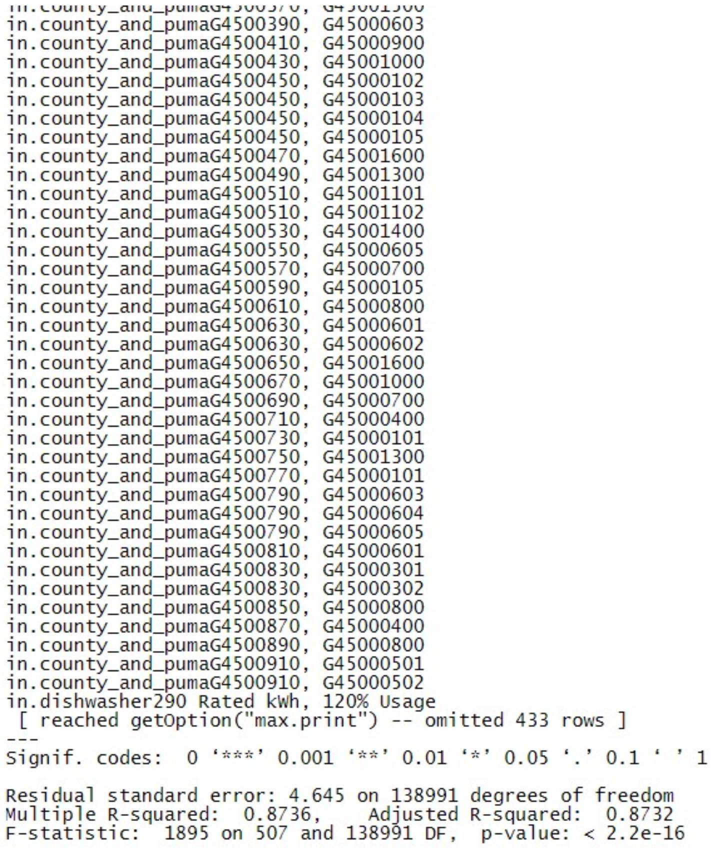
3. Finding Constant Variables:

   (a) In both training and testing data, variables with a single level known as constant variables are located and eliminated.

4. Linear Regression Model:

   (a) The lm function is used to fit a linear regression model to the training set of data.

5. Evaluation of the Model:
   (a) A printed summary of the linear regression model is provided.
   (b) On the basis of the test data, predictions are made, and the target variable's median, minimum, and maximum are computed and printed.
   (c) The model's accuracy is gauged by calculating and printing the Mean Absolute Percentage Error, or MAPE.
6. Multiple R-squared (R²):
   (a) This measure shows how much of the variance in the response variable (total_energy) can be attributed to the predictors.
   (b) The model explains approximately 87.32% of the variance in this case.
   (c) The Adjusted R-squared is a truncated form of R-squared that accounts for the quantity of predictors included in the model.
7. P-value:
   (a) The model may be statistically significant because the p-value for the F-statistic is extremely close to zero (greater than 2.2e-16).
   (b) The null hypothesis, according to which all coefficients are 0, is refuted by this.



```
in.county_and_pumaG4500370, G45001300                             7.402 1.35e-13
in.county_and_pumaG4500390, G45000603                             2.976 0.002923 **
in.county_and_pumaG4500410, G45000900                             4.799 1.59e-06 ***
in.county_and_pumaG4500430, G45001000                            -4.122 3.76e-05 ***
in.county_and_pumaG4500450, G45000102                            -3.195 0.001399 **
in.county_and_pumaG4500450, G45000103                            -0.269 0.788070
in.county_and_pumaG4500450, G45000104                             1.685 0.092049 .
in.county_and_pumaG4500450, G45000105                            -3.266 0.001091 **
in.county_and_pumaG4500470, G45001600                             3.183 0.001458 **
in.county_and_pumaG4500490, G45001300                             1.539 0.123871
in.county_and_pumaG4500510, G45001101                            -4.903 9.43e-07 ***
in.county_and_pumaG4500510, G45001102                            -8.579  < 2e-16 ***
in.county_and_pumaG4500530, G45001400                                NA       NA
in.county_and_pumaG4500550, G45000605                             1.765 0.077571 .
in.county_and_pumaG4500570, G45000700                             0.558 0.577029
in.county_and_pumaG4500590, G45000105                             2.263 0.023652 *
in.county_and_pumaG4500610, G45000800                            -3.193 0.001407 **
in.county_and_pumaG4500630, G45000601                             7.770 7.89e-15 ***
in.county_and_pumaG4500630, G45000602                             7.685 1.54e-14 ***
in.county_and_pumaG4500650, G45001600                             0.155 0.876537
in.county_and_pumaG4500670, G45001000                             4.605 4.12e-06 ***
in.county_and_pumaG4500690, G45000700                             7.251 4.16e-13 ***
in.county_and_pumaG4500710, G45000400                             3.673 0.000240 ***
in.county_and_pumaG4500730, G45000101                             2.029 0.042437 *
in.county_and_pumaG4500750, G45001300                             5.172 2.32e-07 ***
in.county_and_pumaG4500770, G45000101                            -0.499 0.617722
in.county_and_pumaG4500790, G45000603                             5.880 4.11e-09 ***
in.county_and_pumaG4500790, G45000604                             6.264 3.76e-10 ***
in.county_and_pumaG4500790, G45000605                            -0.223 0.823243
in.county_and_pumaG4500810, G45000601                            -3.195 0.001399 **
in.county_and_pumaG4500830, G45000301                            -1.113 0.265500
in.county_and_pumaG4500830, G45000302                            -1.009 0.312999
in.county_and_pumaG4500850, G45000800                             2.052 0.040181 *
in.county_and_pumaG4500870, G45000400                            -2.776 0.005511 **
in.county_and_pumaG4500890, G45000800                             6.226 4.79e-10 ***
in.county_and_pumaG4500910, G45000501                             2.683 0.007291 **
in.county_and_pumaG4500910, G45000502                             3.420 0.000627 ***
in.dishwasher290 Rated kWh, 120% Usage                           -1.547 0.121839
 [ reached getOption("max.print") -- omitted 433 rows ]
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.645 on 138991 degrees of freedom
Multiple R-squared:  0.8736,    Adjusted R-squared:  0.8732
F-statistic:  1895 on 507 and 138991 DF,  p-value: < 2.2e-16
```

Figure 1: Per hour energy usage prediction

```
# Load necessary libraries
library(caret)

# Create a copy of the dataset
merge_static_house_info_df4 <- merge_static_house_info_df3

# Select columns where the number of distinct values is greater than 1
merge_static_house_info_df4 <- merge_static_house_info_df4 %>%
  select(where(˜n_distinct(.) > 1))

# Create a copy for prediction
merge_static_house_info_df_prediction <- merge_static_house_info_df4

# Create a subset with selected columns for building and county information
merge_static_house_info_df_building_and_county <- merge_static_house_info_df4[,c(’bldg_id’,’in.county’,’date’)]

# Remove unnecessary columns for modeling
merge_static_house_info_df4 <- merge_static_house_info_df4 %>% select(-c(’bldg_id’,’in.county’))

# Set seed for reproducibility
set.seed(123)

# Split the dataset into training and testing sets
index <- createDataPartition(merge_static_house_info_df4$day_total_energy, p = 0.8, list = FALSE)
train_df1 <- merge_static_house_info_df4[index, ]
test_df1 <- merge_static_house_info_df4[-index, ]

# Remove rows from the test set where categorical values are not present in the training set
character_columns <- names(train_df1)[sapply(train_df1, is.character)]
for (col in character_columns) {
  unique_values <- unique(train_df1[[col]])
  test_df1 <- test_df1[test_df1[[col]] %in% unique_values, ]
}

# Build a linear regression model
model <- lm(day_total_energy ˜ ., data = train_df1)

# Print the summary of the model
summary(model)

# Make predictions on the test set
predictions <- predict(model, newdata = test_df1)
```

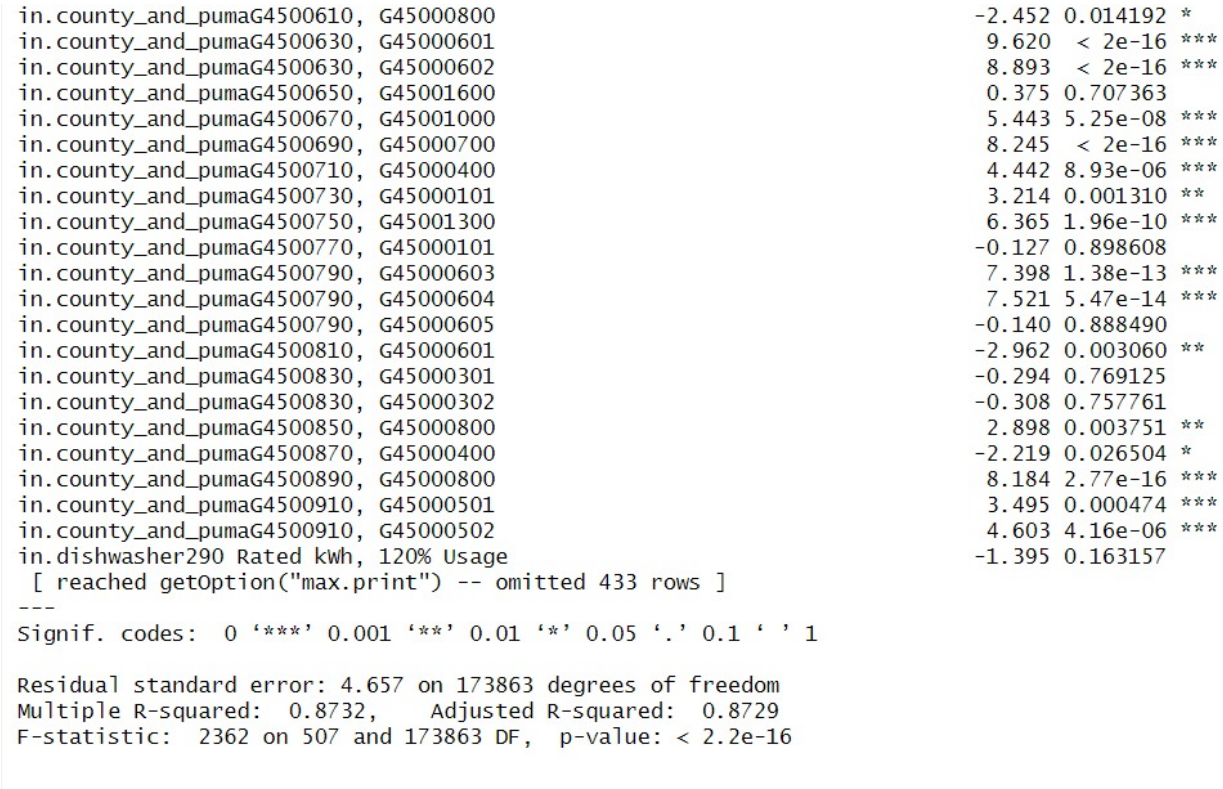
```

# Calculate Root Mean Squared Error (RMSE) on the test data
rmse <- sqrt(mean((test_df1$day_total_energy - predictions)^2))
print(paste("Root Mean Squared Error on test data:", rmse))

# Display summary statistics for the test data
cat("Minimum:", min(test_df1$day_total_energy), "\n")
cat("Maximum:", max(test_df1$day_total_energy), "\n")
cat("Mean:", mean(test_df1$day_total_energy), "\n")

# Calculate Mean Absolute Percentage Error (MAPE)
mape <- mean(abs((test_df1$day_total_energy - predictions) /
    test_df1$day_total_energy )) * 100

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



```
in.county_and_pumaG4500610, G45000800                                    -2.452 0.014192 *
in.county_and_pumaG4500630, G45000601                                     9.620  < 2e-16 ***
in.county_and_pumaG4500630, G45000602                                     8.893  < 2e-16 ***
in.county_and_pumaG4500650, G45001600                                     0.375 0.707363
in.county_and_pumaG4500670, G45001000                                     5.443 5.25e-08 ***
in.county_and_pumaG4500690, G45000700                                     8.245  < 2e-16 ***
in.county_and_pumaG4500710, G45000400                                     4.442 8.93e-06 ***
in.county_and_pumaG4500730, G45000101                                     3.214 0.001310 **
in.county_and_pumaG4500750, G45001300                                     6.365 1.96e-10 ***
in.county_and_pumaG4500770, G45000101                                    -0.127 0.898608
in.county_and_pumaG4500790, G45000603                                     7.398 1.38e-13 ***
in.county_and_pumaG4500790, G45000604                                     7.521 5.47e-14 ***
in.county_and_pumaG4500790, G45000605                                    -0.140 0.888490
in.county_and_pumaG4500810, G45000601                                    -2.962 0.003060 **
in.county_and_pumaG4500830, G45000301                                    -0.294 0.769125
in.county_and_pumaG4500830, G45000302                                    -0.308 0.757761
in.county_and_pumaG4500850, G45000800                                     2.898 0.003751 **
in.county_and_pumaG4500870, G45000400                                    -2.219 0.026504 *
in.county_and_pumaG4500890, G45000800                                     8.184 2.77e-16 ***
in.county_and_pumaG4500910, G45000501                                     3.495 0.000474 ***
in.county_and_pumaG4500910, G45000502                                     4.603 4.16e-06 ***
in.dishwasher290 Rated kWh, 120% Usage                                   -1.395 0.163157
 [ reached getOption("max.print") -- omitted 433 rows ]
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.657 on 173863 degrees of freedom
Multiple R-squared:  0.8732,    Adjusted R-squared:  0.8729
F-statistic:  2362 on 507 and 173863 DF,  p-value: < 2.2e-16
```

Figure 2: Peak future energy demand

4.4 Demand Analysis

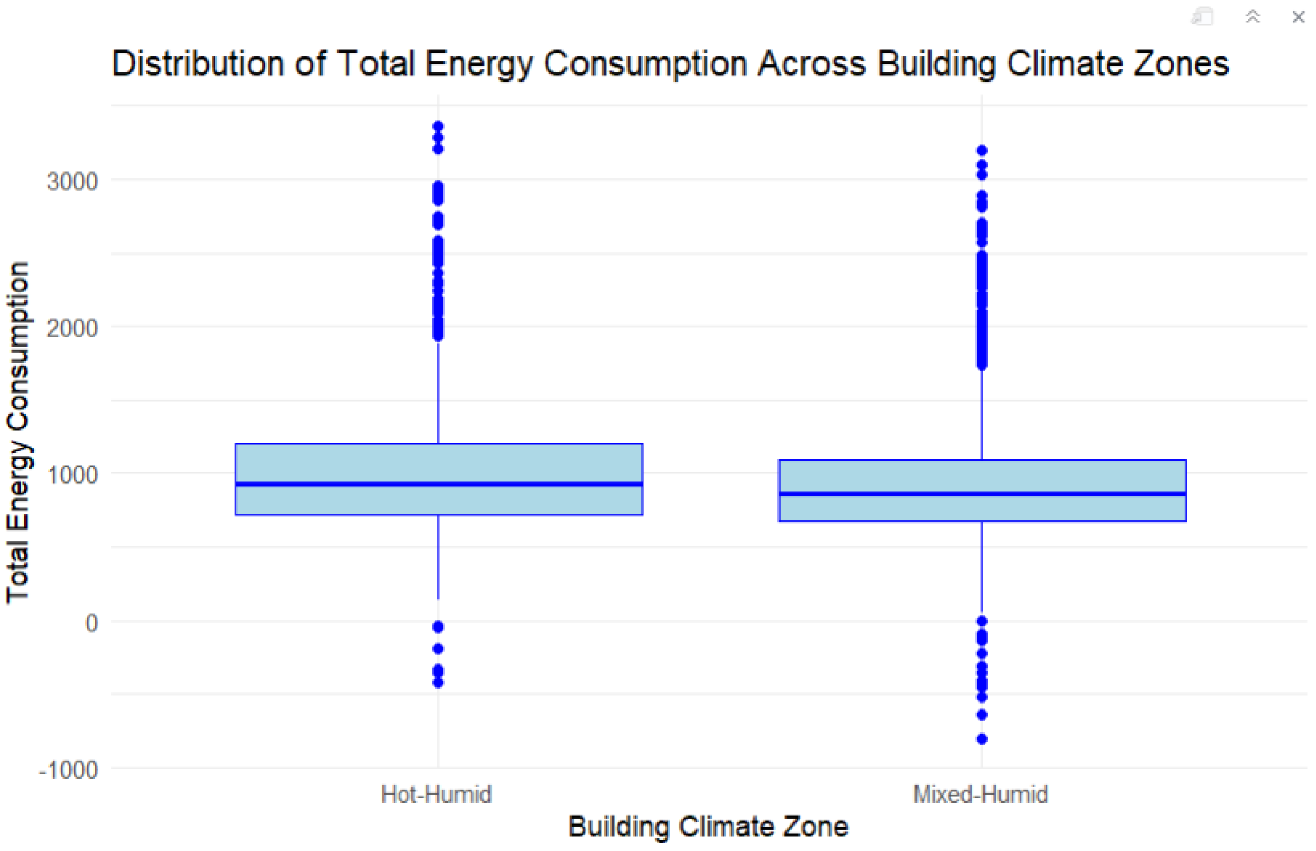


Figure 3: Distribution of Total Energy Consumption

1. The distribution of total energy consumption across building climate zones is right-skewed, meaning that there are more buildings with lower energy consumption than there are buildings with higher energy consumption.

2. The median total energy consumption is around 1,000 units. There is a large range in total energy consumption, from -1,000 units to 3,000 units.

3. The total energy consumption appears to be bimodal, with peaks at around 1,000 units and 2,000 units.

4. There is a lot of variability in total energy consumption within each climate zone. For example, in the Hot-Humid climate zone, there is a range of total energy consumption from -1,000 units to 3,000 units.

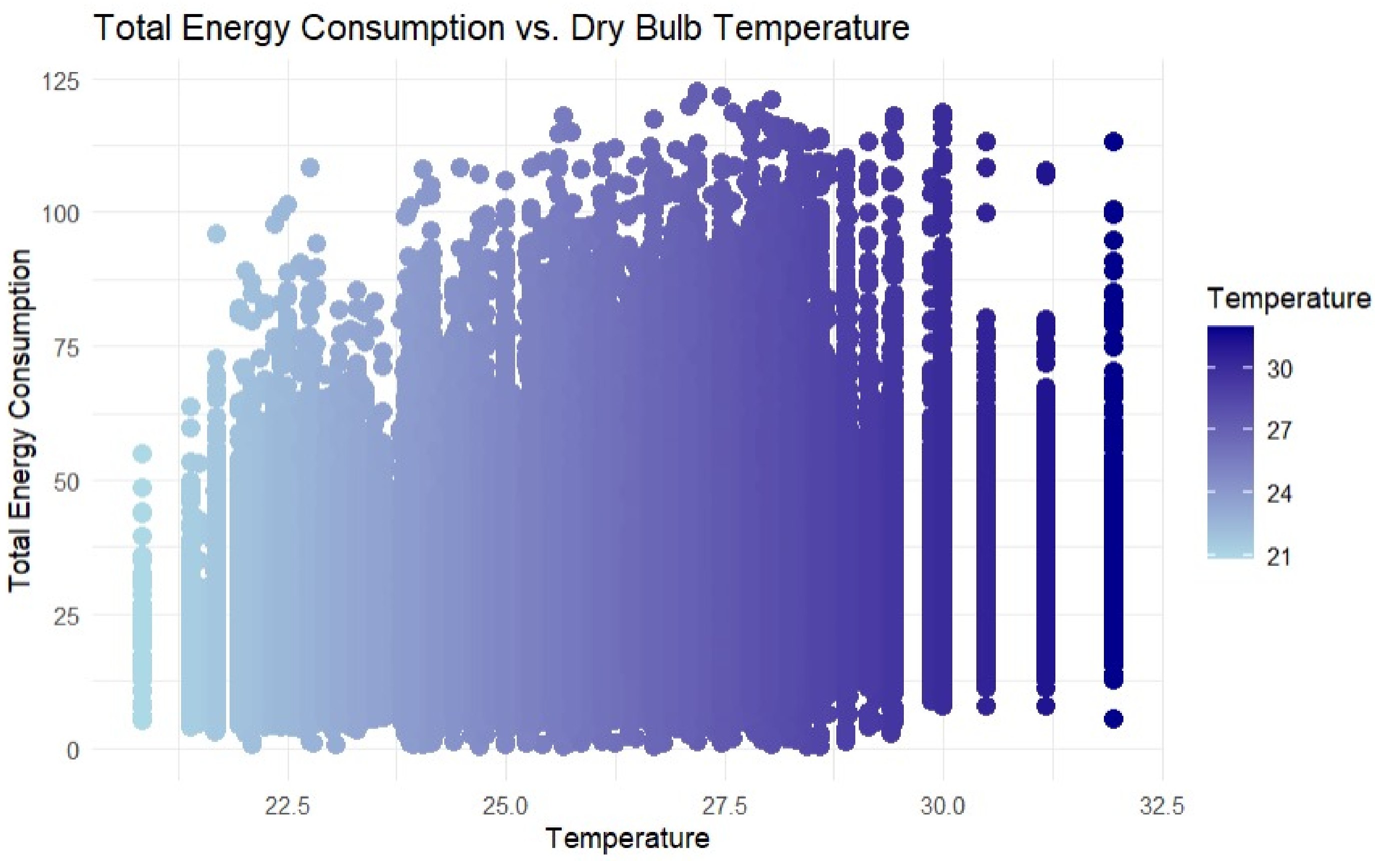


Figure 4: Total Energy Consumption

1. The graph shows that there is a positive correlation between dry bulb temperature and total energy consumption. This means that as the dry bulb temperature increases, the total energy consumption also increases.

2. There are a few possible explanations for this correlation. One possibility is that people tend to use more energy to cool their homes and businesses when it is hot outside. Another possibility is that hot weather can put a strain on the power grid, which can lead to higher energy consumption.

3. It is also important to note that the graph only shows a correlation, not a causation. This means that we cannot say for sure that dry bulb temperature is the cause of the increase in energy consumption. There could be other factors that are also contributing to the increase.

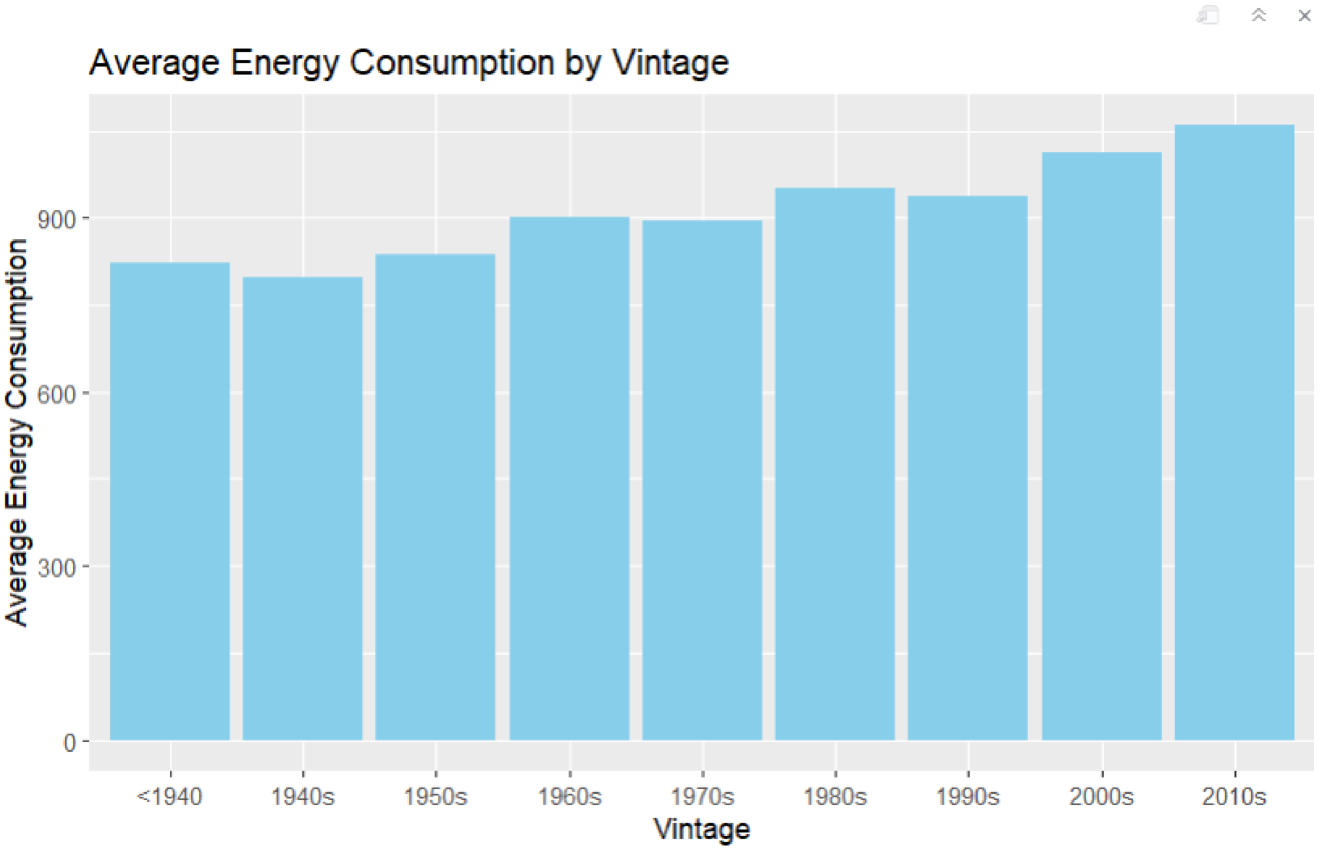


Figure 5: Average Energy Consumption

1. The graph helps in identifying patterns or variations in average energy consumption across different vintage categories.
2. It provides a quick comparison of energy efficiency or consumption trends based on the age of houses
3. Further investigation into the factors contributing to variations in energy consumption by vintage could provide more actionable insights.
4. Understanding whether newer or older houses tend to have higher or lower energy efficiency can inform strategies for demand reduction.

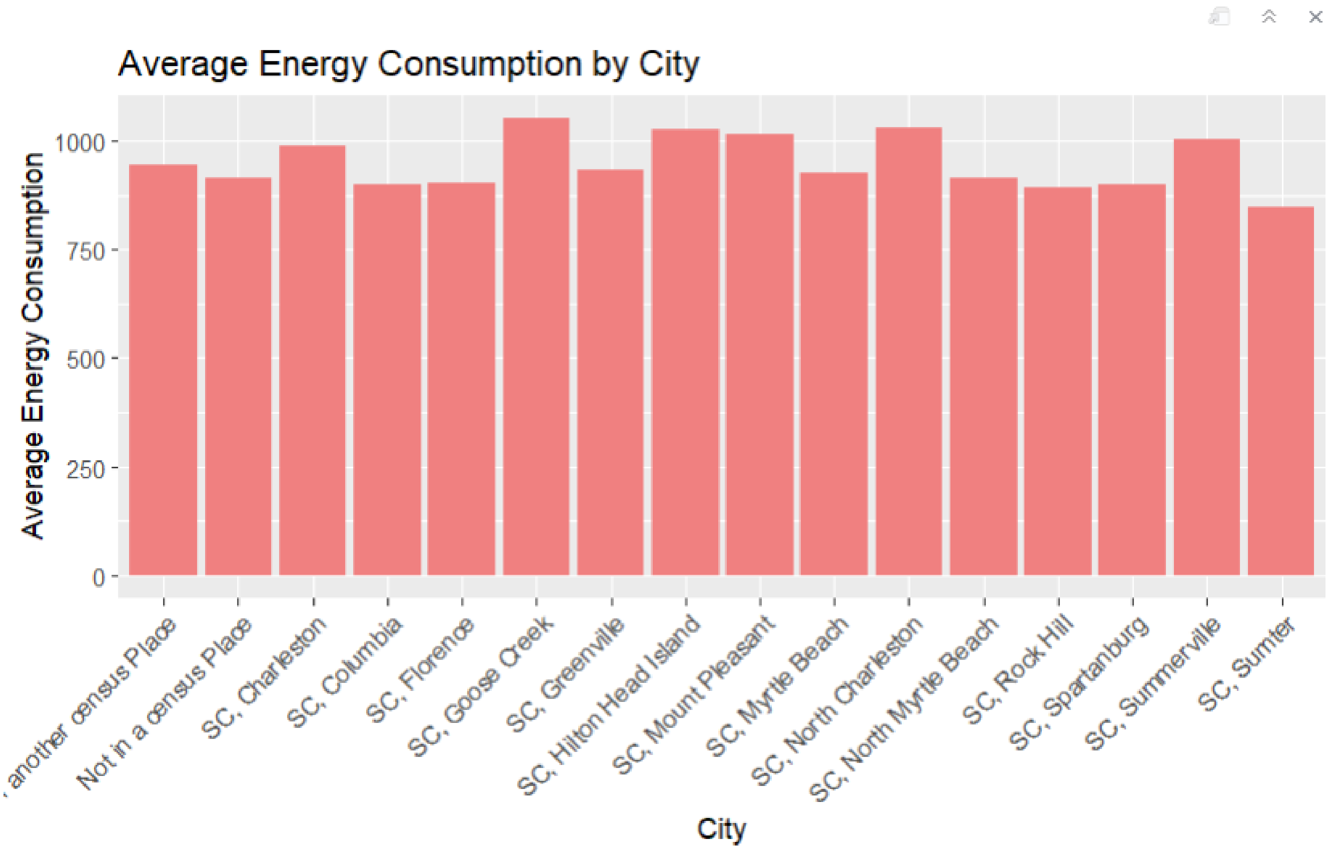


Figure 6: Average Energy Consumption

1. The height of each bar indicates the average energy consumption for houses in a specific city.
2. Higher bars suggest higher average energy consumption for houses in that city.
3. Lower bars indicate lower average energy consumption.
4. The graph facilitates a quick comparison of average energy consumption across different cities.
5. It allows identification of cities with relatively higher or lower average energy consumption.
6. Variations in average energy consumption may be influenced by factors such as climate, population density, or local infrastructure.

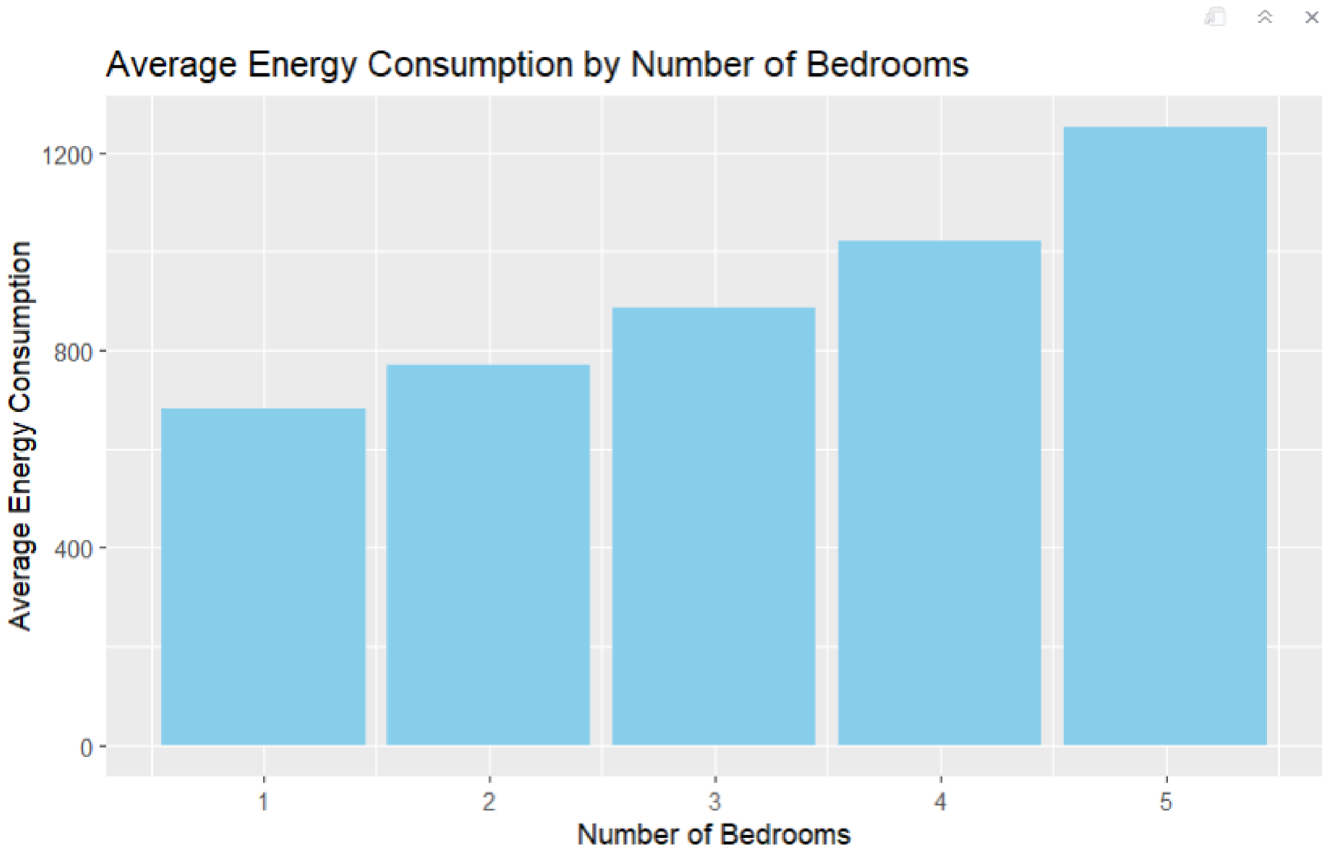


Figure 7: Average Energy Consumption

1. The bar chart simplifies the relationship by presenting average energy consumption for each category of the number of bedrooms.

2. It allows for a clearer comparison of average energy consumption across different bedroom categories.

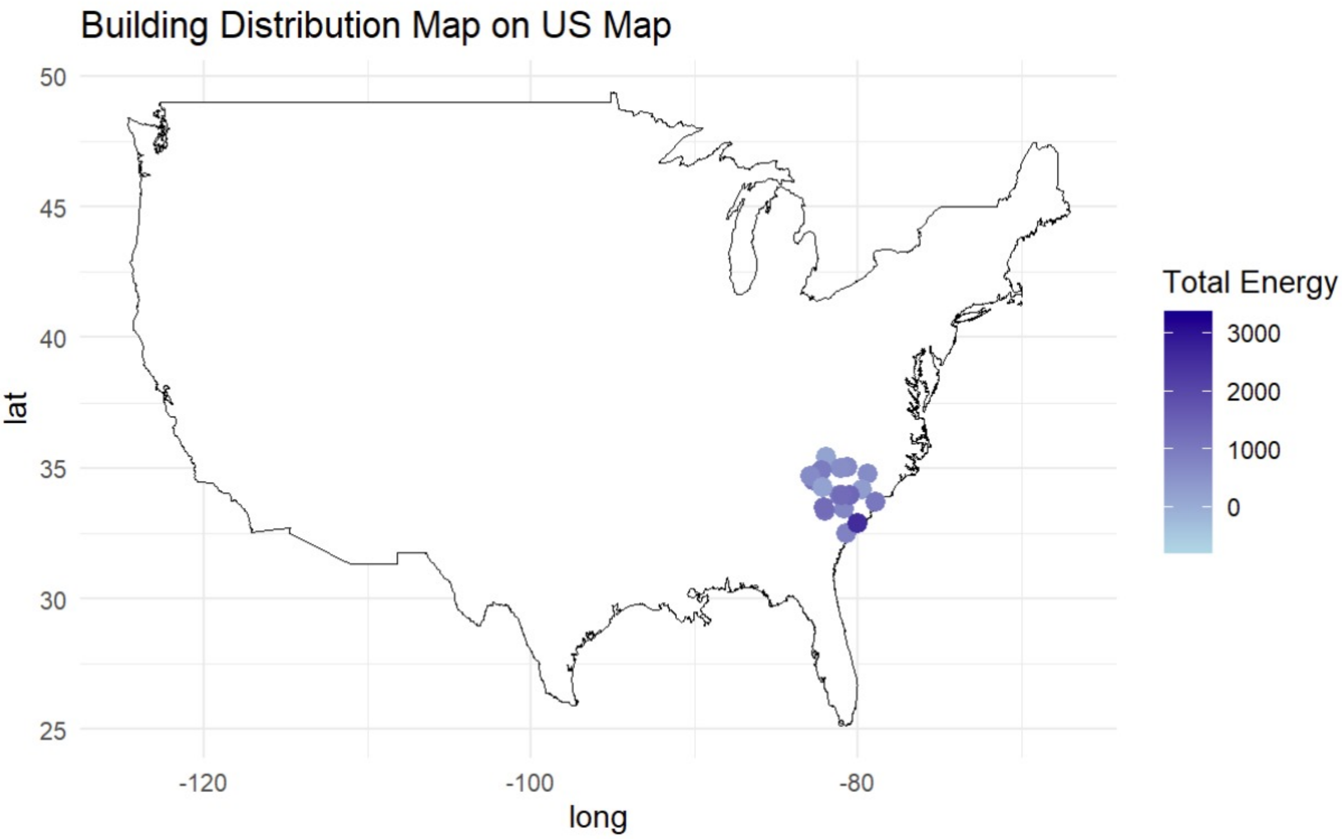


Figure 8: Distribution based on region

1. The graph shows distribution of buildings across regions.

2. The dots represent the energy usage.

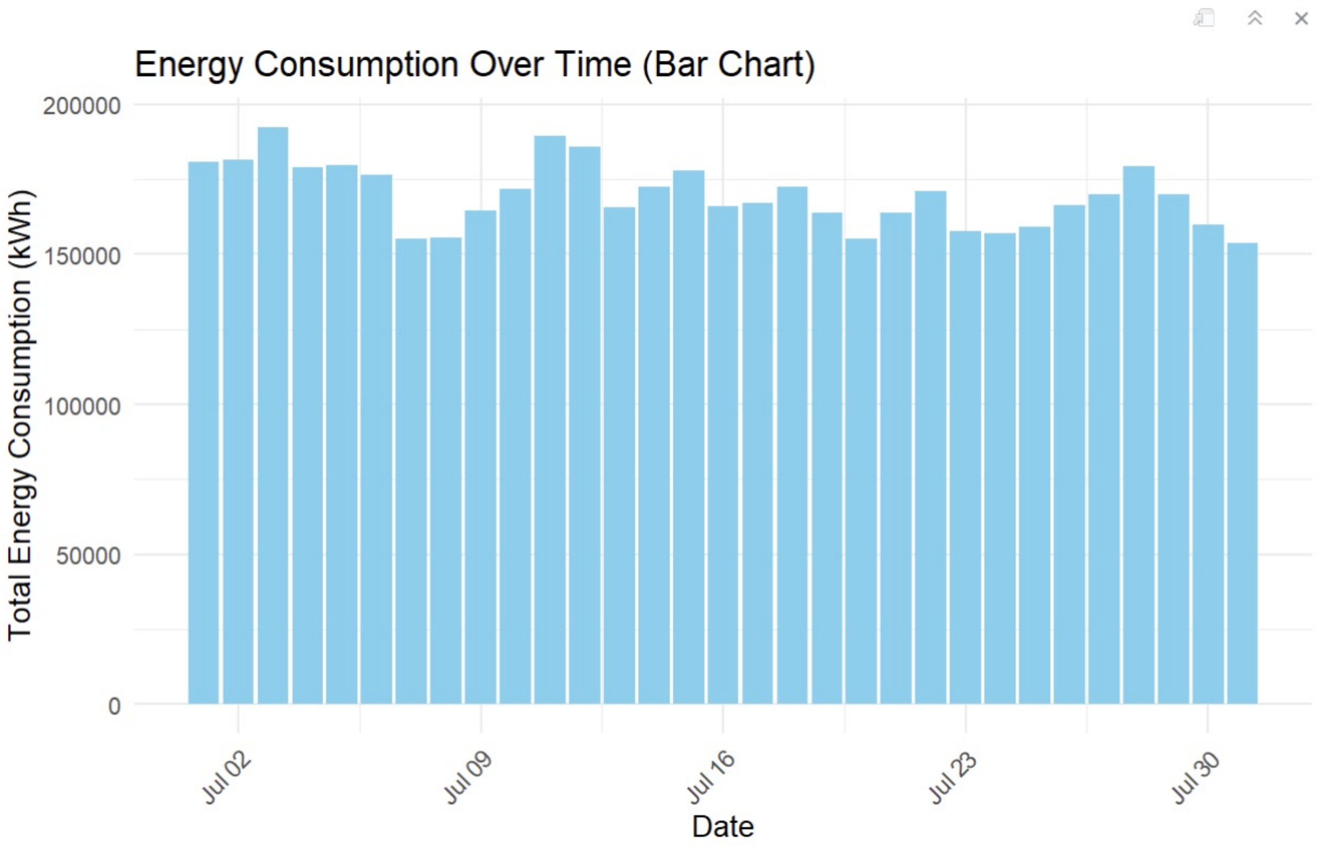
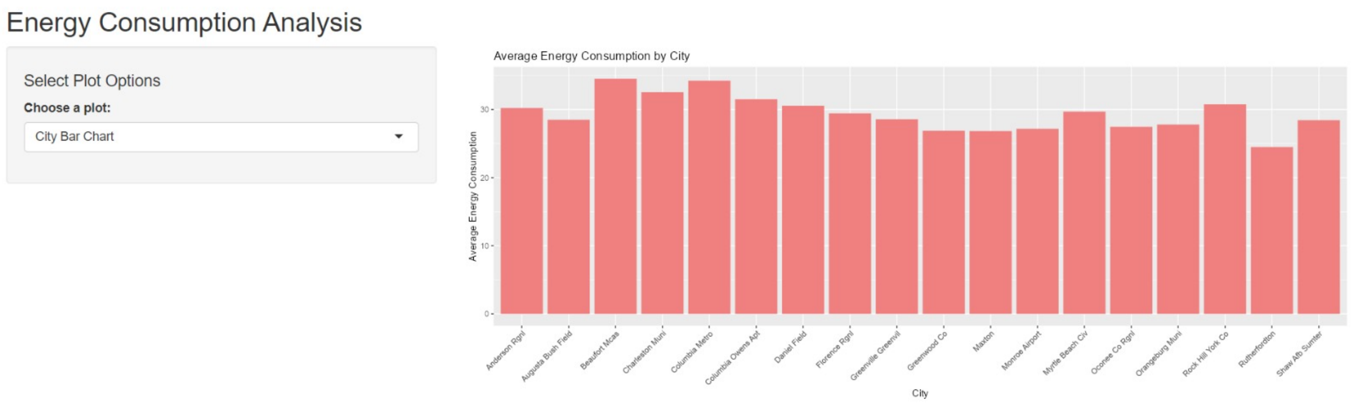


Figure 9: Energy consumption for the month of July

1. The energy consumption for the month of July is variable. This might be because of the temperature variability.

2. In order to reduce the temperature inside the house, the residents might be consuming more energy to cool the indoors.

4.5 Shiny App



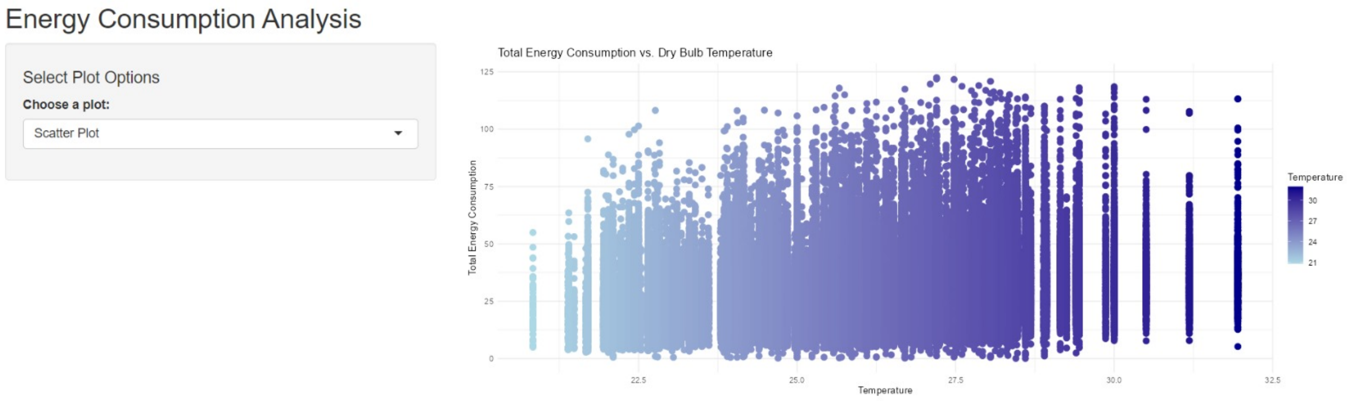
Energy Consumption Analysis

Select Plot Options

Choose a plot:

City Bar Chart

Figure 10: Shiny App: Energy Consumption Analysis by City



Energy Consumption Analysis

Select Plot Options

Choose a plot:

Scatter Plot

Figure 11: Shiny App: Energy Consumption Analysis by Dry Bulb Temperature

4.6 Demand Reduction Strategy

1. Occupancy Monitoring:
   (a) Utilize data on corridor usage and occupancy schedules.
   (b) Implement energy management solutions that are based on real-time occupancy data.
   (c) Adjust lighting, HVAC settings, and other resources based on occupancy patterns, for example.
2. Incorporating Solar Power:
   (a) Examine the viability of integrating solar power systems.
   (b) Determine the building's solar energy generation potential.
   (c) Install solar panels on rooftops to generate clean energy as an example.
3. Implementation of Smart Grids:
   (a) Implement technologies that allow utility providers and buildings to communicate with one another.
   (b) Increase the efficiency of energy distribution by exchanging data in real time.
   (c) Smart meters and demand response systems are two examples.
4. Monitoring Systems for Energy:
   (a) Install systems that continuously monitor energy consumption.
   (b) Determine unusual energy use trends and areas for improvement.
   (c) Smart meters and energy monitoring software are two examples.
5. Academic Programs:
   (a) Organize initiatives to raise awareness about energy-saving practices.
   (b) Encourage residents to adopt energy-saving habits.
   (c) Workshops and educational campaigns are two examples.
6. Maintenance on a regular basis:
   (a) Create a maintenance schedule for all systems and equipment.
   (b) Ensure that the HVAC, lighting, and other systems are operating at maximum efficiency.
   (c) Regular inspections and filter replacements are two examples.
7. Evaluate and test:
   (a) Pilot initiatives on a modest scale should be implemented to test novel energy solutions.
   (b) Before implementing each method widely, assess its effectiveness.
8. Exemplification: Before broad deployment, test energy-efficient solutions in specific locations.

5 Challenges faced

1. Data Availability: Executing data collection (Parquet files, CSV files) through AWS.
2. Column Names and Types: Checking if column names and data types are consistent across datasets for successful merging.
3. Data Matching: Verifying that the building IDs used for merging are similar across datasets.
4. Memory Considerations: Depending on the size of your datasets, merging large datasets consumed a significant amount of time and memory.
5. Model Selection: Selecting and executing a model to receive accurate predictions.

6 Contributions

1. Abhishek: Collect and merge data, Data Cleaning, Data Visualization
2. Bhavika: Data Cleaning,Data Visualization, Presentation, Report
3. Goutham: Data Cleaning, Data Visualization, Data Modeling
4. Prasanna: Shiny App, Presentation
5. Minnie:

7 Conclusion

Firstly, we extend heartfelt gratitude to my Professor, Prof. Anderson and Prof. Saltz, for guiding us through this incredible learning experience in the course. Our engagement in this project was a mix of rewards and challenges while navigating the complexities of programming and analytical thinking.

The evaluation and comparison of predictions and reports depicted insights on the project's efficiency. It also highlighted the areas of improvements and potential bottlenecks in the analysis. The diverse set of tasks in the project along with a need of continuous learning made it intellectually simulating.
To summarize, the experience was a blend of challenges, continuous learning and problem solving.

References

[1] *ChatGPT*.