IST 687 M002
Group No- 5

Technical Report Paper

Energy Usage & Savings

Introduction to Data Science

Content:-

1. Describing The Data
2. Exploratory Analysis
3. Visualizing The Data
4. Modeling The Data
5. Shiny App
6. Modeling The Data
7. Potential Approach to Reduce Peak Energy Demand
8. Conclusion
9. Work Log

- **Describing the Data-**

The dataset for this project contained static house data, weather data, and energy consumption data. The static house data includes information on each specific house such as how many floors the house has, the square footage of the house, and its geographical location. The weather data includes daily information such as wind speed, temperature, and humidity, which could possibly affect the energy consumption for each household. Since the house data was created as a .parquet file we had to use the read_parquet() function. We created a function called “process_energy_data” in order to intake multiple building ideas and then read them into R.

```
# Function to process energy data for each house
process_energy_data <- function(bldg_id) {
  energy_test <- paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023
-houseData/", bldg_id, ".parquet")
  energy <- read_parquet(energy_test)
```

Above is what the static building data looks like without any cleaning or merging of the weather or energy data. There are over 5,170 houses within the data and about 171 columns, which signifies the giant amount of data that we are working with in order to predict energy consumption.

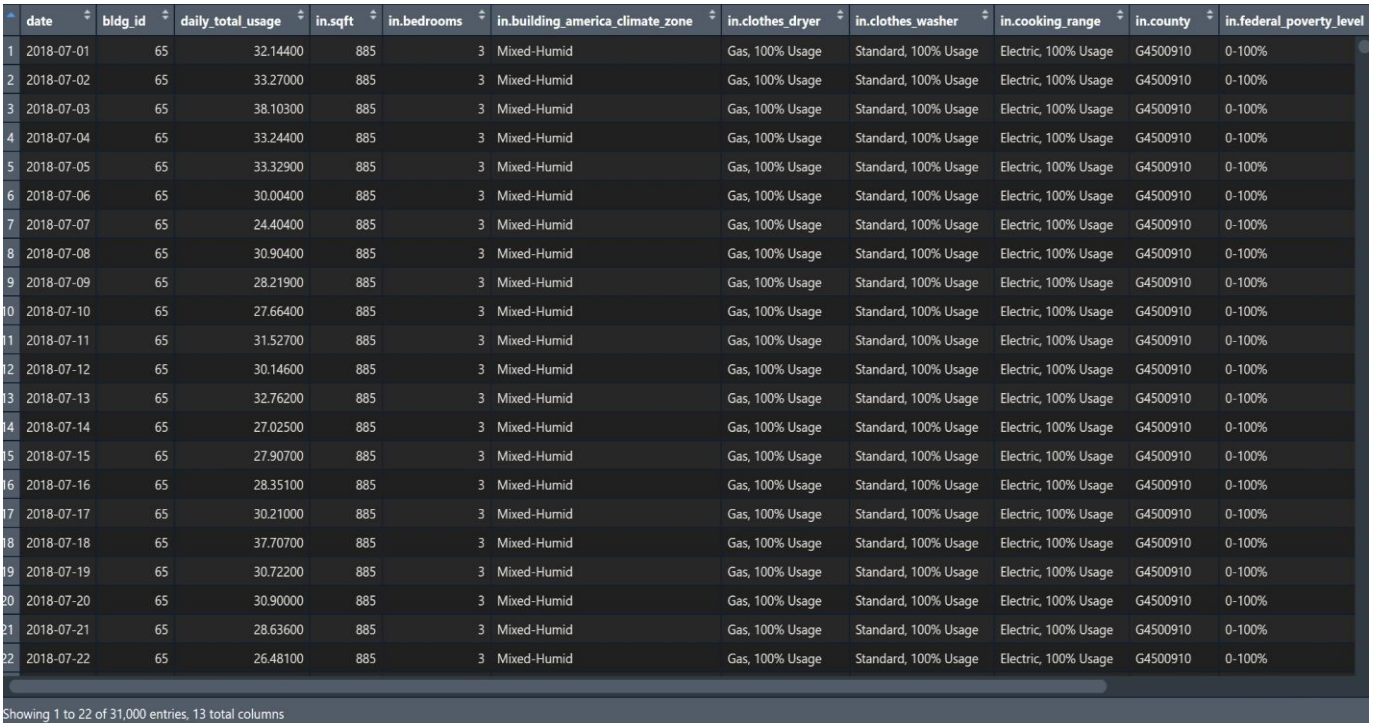
Filter | Cols: « < 1 - 50 > »

	bldg_id	upgrade	weight	applicability	in.sqft	in.ahs_region	in.ashrae_iecc_climate_zone_2004	in.ashrae_iecc_climate_zone_2004_2_a_split	in.bathroom_spot_vent_hour
1	65	10	242.131	TRUE	885	Non-CBSA South Atlantic	3A	3A	Hour23
2	121	10	242.131	TRUE	1220	Non-CBSA South Atlantic	3A	3A	Hour20
3	500	10	242.131	TRUE	1220	Non-CBSA South Atlantic	3A	3A	Hour11
4	504	10	242.131	TRUE	1690	Non-CBSA South Atlantic	3A	3A	Hour13
5	581	10	242.131	TRUE	1690	Non-CBSA South Atlantic	3A	3A	Hour22
6	590	10	242.131	TRUE	2176	Non-CBSA South Atlantic	3A	3A	Hour5
7	670	10	242.131	TRUE	885	Non-CBSA South Atlantic	3A	3A	Hour6
8	736	10	242.131	TRUE	2663	Non-CBSA South Atlantic	3A	3A	Hour7
9	862	10	242.131	TRUE	885	Non-CBSA South Atlantic	3A	3A	Hour19
10	952	10	242.131	TRUE	2663	Non-CBSA South Atlantic	3A	3A	Hour4
11	1202	10	242.131	TRUE	2176	Non-CBSA South Atlantic	3A	3A	Hour5
12	1220	10	242.131	TRUE	1220	Non-CBSA South Atlantic	3A	3A	Hour5
13	1410	10	242.131	TRUE	1220	Non-CBSA South Atlantic	3A	3A	Hour22
14	1514	10	242.131	TRUE	1220	Non-CBSA South Atlantic	3A	3A	Hour22
15	1801	10	242.131	TRUE	2176	Non-CBSA South Atlantic	3A	3A	Hour4
16	1961	10	242.131	TRUE	2176	Non-CBSA South Atlantic	3A	3A	Hour19
17	2077	10	242.131	TRUE	1220	Non-CBSA South Atlantic	3A	3A	Hour19

	date	daily_temperature	daily_humidity	in_county
1	2018-07-01	26.73250	82.86500	G4500910
2	2018-07-02	26.92083	82.94208	G4500910
3	2018-07-03	29.37792	70.74667	G4500910
4	2018-07-04	28.32917	71.63583	G4500910
5	2018-07-05	27.93542	71.94208	G4500910
6	2018-07-06	25.69625	84.08500	G4500910
7	2018-07-07	22.99375	88.96667	G4500910
8	2018-07-08	23.19583	63.49625	G4500910
9	2018-07-09	23.56250	62.08750	G4500910
10	2018-07-10	25.69583	67.34125	G4500910
11	2018-07-11	27.53750	69.83792	G4500910
12	2018-07-12	27.47833	76.97375	G4500910
13	2018-07-13	26.19792	77.12792	G4500910
14	2018-07-14	25.97208	75.78042	G4500910

Here we can see the weather data set before merging and before cleaning the dataset. There are four columns and 1,395 rows. Each row signifies a different day within July from which our data is taken from. Next, our data was merged and cleaned in order to identify which columns we believed to be significant in analysing daily energy consumption for certain households. Those columns were the building ID for each house, their total energy usage (our dependant variable), the square footage of the house, the climate zone of the house, the type of dryer and washing machine, its cooking range, the county in which it recedes, the poverty level of the household, the daily temperature and the daily humidity.

- **Exploratory Analysis-**



	date	bldg_id	daily_total_usage	in.sqft	in.bedrooms	in.building_america_climate_zone	in.clothes_dryer	in.clothes_washer	in.cooking_range	in.county	in.federal_poverty_level
1	2018-07-01	65	32.14400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
2	2018-07-02	65	33.27000	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
3	2018-07-03	65	38.10300	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
4	2018-07-04	65	33.24400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
5	2018-07-05	65	33.32900	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
6	2018-07-06	65	30.00400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
7	2018-07-07	65	24.40400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
8	2018-07-08	65	30.90400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
9	2018-07-09	65	28.21900	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
10	2018-07-10	65	27.66400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
11	2018-07-11	65	31.52700	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
12	2018-07-12	65	30.14600	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
13	2018-07-13	65	32.76200	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
14	2018-07-14	65	27.02500	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
15	2018-07-15	65	27.90700	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
16	2018-07-16	65	28.35100	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
17	2018-07-17	65	30.21000	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
18	2018-07-18	65	37.70700	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
19	2018-07-19	65	30.72200	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
20	2018-07-20	65	30.90000	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
21	2018-07-21	65	28.63600	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%
22	2018-07-22	65	26.48100	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%

Showing 1 to 22 of 31,000 entries, 13 total columns

b. Do exploratory analysis of the data – to gain some basic insight about the data

```{r}
# first 5 rows of the data
head(final)
# last 5 rows of the data
tail(final)
# structure
str(final)
# summary
summary(final)
```

For the data overview part, we have used the first few rows of the dataset to provide an initial glimpse of the structure by using head() function. By using the tail() function we have showcased the last few rows offering insights into the end of the dataset. Used str() to reveal the structure of the dataset, including data types and variable names. summary() is used to provide statistical summary such as mean, minimum, maximum for the variables.

```
> str(final)
tibble [31,000 x 13] (S3: tbl_df/tbl/data.frame)
 $ date                          : Date[1:31000], format: "2018-07-01" "2018-07-02" "2018-07-03" "2018-07-04" ...
 $ bldg_id                       : int [1:31000] 65 65 65 65 65 65 65 65 65 65 ...
 $ daily_total_usage             : num [1:31000] 32.1 33.3 38.1 33.2 33.3 ...
 $ in.sqft                       : int [1:31000] 885 885 885 885 885 885 885 885 885 885 ...
 $ in.bedrooms                   : int [1:31000] 3 3 3 3 3 3 3 3 3 3 ...
 $ in.building_america_climate_zone: chr [1:31000] "Mixed-Humid" "Mixed-Humid" "Mixed-Humid" "Mixed-Humid" ...
 $ in.clothes_dryer              : chr [1:31000] "Gas, 100% Usage" "Gas, 100% Usage" "Gas, 100% Usage" "Gas, 100% Usage" ...
 $ in.clothes_washer             : chr [1:31000] "Standard, 100% Usage" "Standard, 100% Usage" "Standard, 100% Usage" "Standard, 100% Us
age" ...
 $ in.cooking_range              : chr [1:31000] "Electric, 100% Usage" "Electric, 100% Usage" "Electric, 100% Usage" "Electric, 100% Us
age" ...
 $ in.county                     : chr [1:31000] "G4500910" "G4500910" "G4500910" "G4500910" ...
 $ in.federal_poverty_level      : chr [1:31000] "0-100%" "0-100%" "0-100%" "0-100%" ...
 $ daily_temperature             : num [1:31000] 26.7 26.9 29.4 28.3 27.9 ...
 $ daily_humidity                : num [1:31000] 82.9 82.9 70.7 71.6 71.9 ...
> # summary
> summary(final)
      date              bldg_id      daily_total_usage    in.sqft      in.bedrooms    in.building_america_climate_zone
 Min.   :2018-07-01   Min.   :    65   Min.   :-42.86    Min.   : 328   Min.   :1.000   Length:31000
 1st Qu.:2018-07-08   1st Qu.: 27138   1st Qu.: 21.20    1st Qu.:1220   1st Qu.:3.000   Class :character
 Median :2018-07-16   Median : 51216   Median : 28.47    Median :1690   Median :3.000   Mode  :character
 Mean   :2018-07-16   Mean   : 50930   Mean   : 30.06    Mean   :2079   Mean   :3.232
 3rd Qu.:2018-07-24   3rd Qu.: 75923   3rd Qu.: 37.00    3rd Qu.:2176   3rd Qu.:4.000
 Max.   :2018-07-31   Max.   :100020   Max.   :117.60    Max.   :8194   Max.   :5.000
 in.clothes_dryer   in.clothes_washer  in.cooking_range    in.county         in.federal_poverty_level daily_temperature
 Length:31000       Length:31000       Length:31000       Length:31000       Length:31000             Min.   :21.54
 Class :character   Class :character   Class :character   Class :character   Class :character         1st Qu.:25.53
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character         Median :26.62
                                                                                                      Mean   :26.49
                                                                                                      3rd Qu.:27.58
                                                                                                      Max.   :31.00

 daily_humidity
 Min.   :44.03
 1st Qu.:71.76
 Median :78.35
 Mean   :77.16
 3rd Qu.:83.22
 Max.   :93.85
```

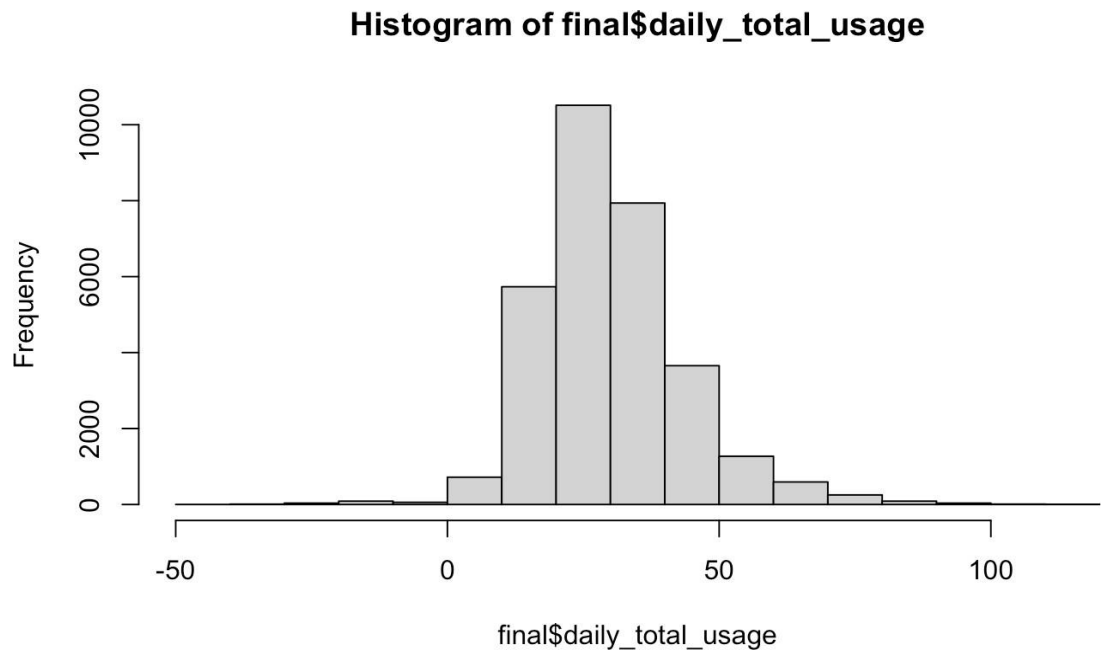
- **Visualizing the Data-**

```
b. Do exploratory analysis of the data – to gain some basic insight about the data

```{r}
# first 5 rows of the data

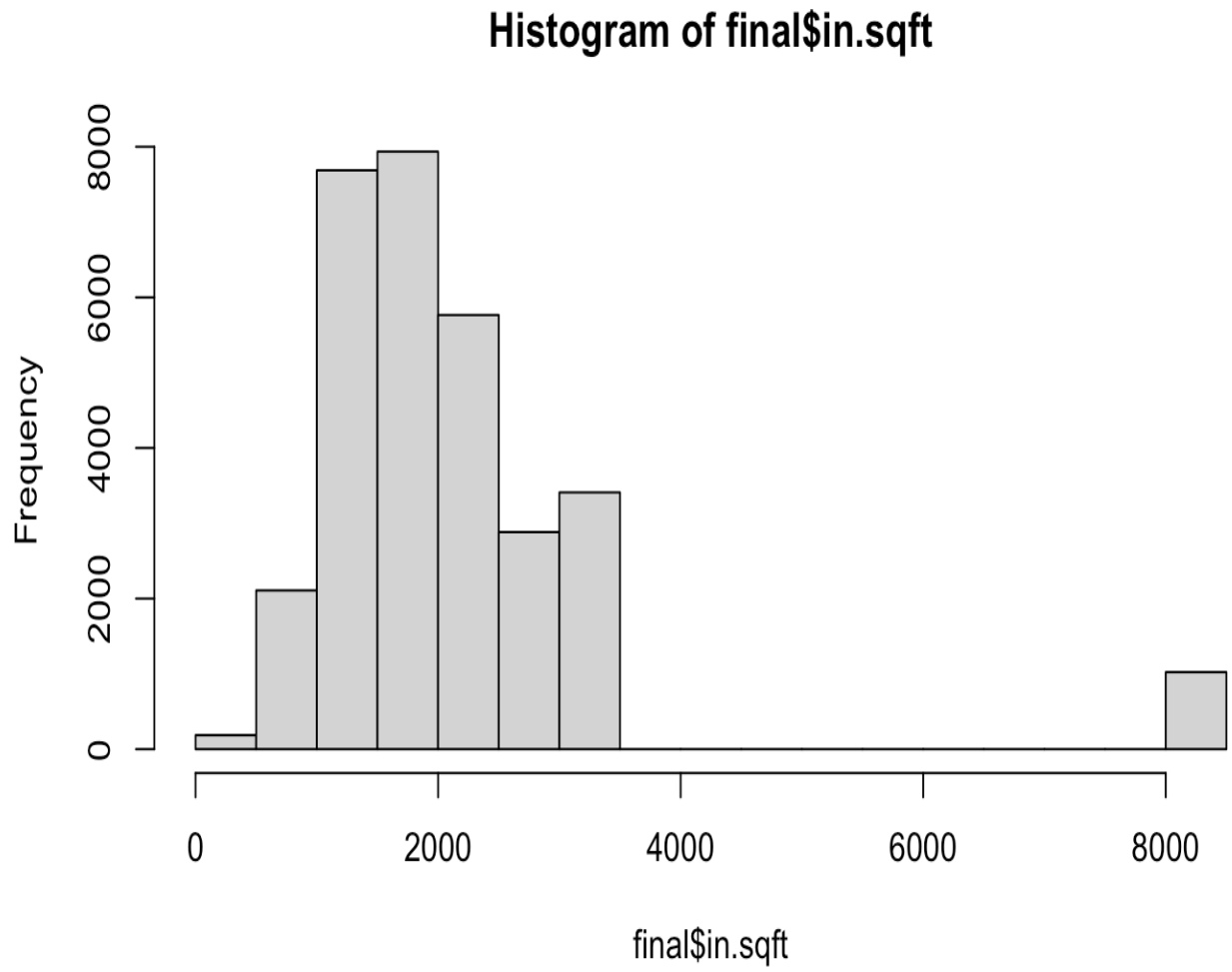
hist(final$in.sqft)
hist(final$daily_total_usage)
hist(final$in.bedrooms)
head(final)
# last 5 rows of the data
tail(final)
# structure
str(final)
# summary
summary(final)
# missing values
final_cleaned <- na.omit(final)
# duplicates
num_duplicates <- sum(duplicated(final))
num_duplicates
```
```

We utilized histograms to visualize the distribution of key variables. The below histogram displays values of daily_total_usage between -25 to 100. But the highest frequency, which is 10000, lies between 20 to 30 units. Also the frequency value 8000 and 6000 lies between 30 to 40 and 10 to 20 respectively. The values of daily_total_usage at -25 to 0 and 90 to 100 are quite negligible.

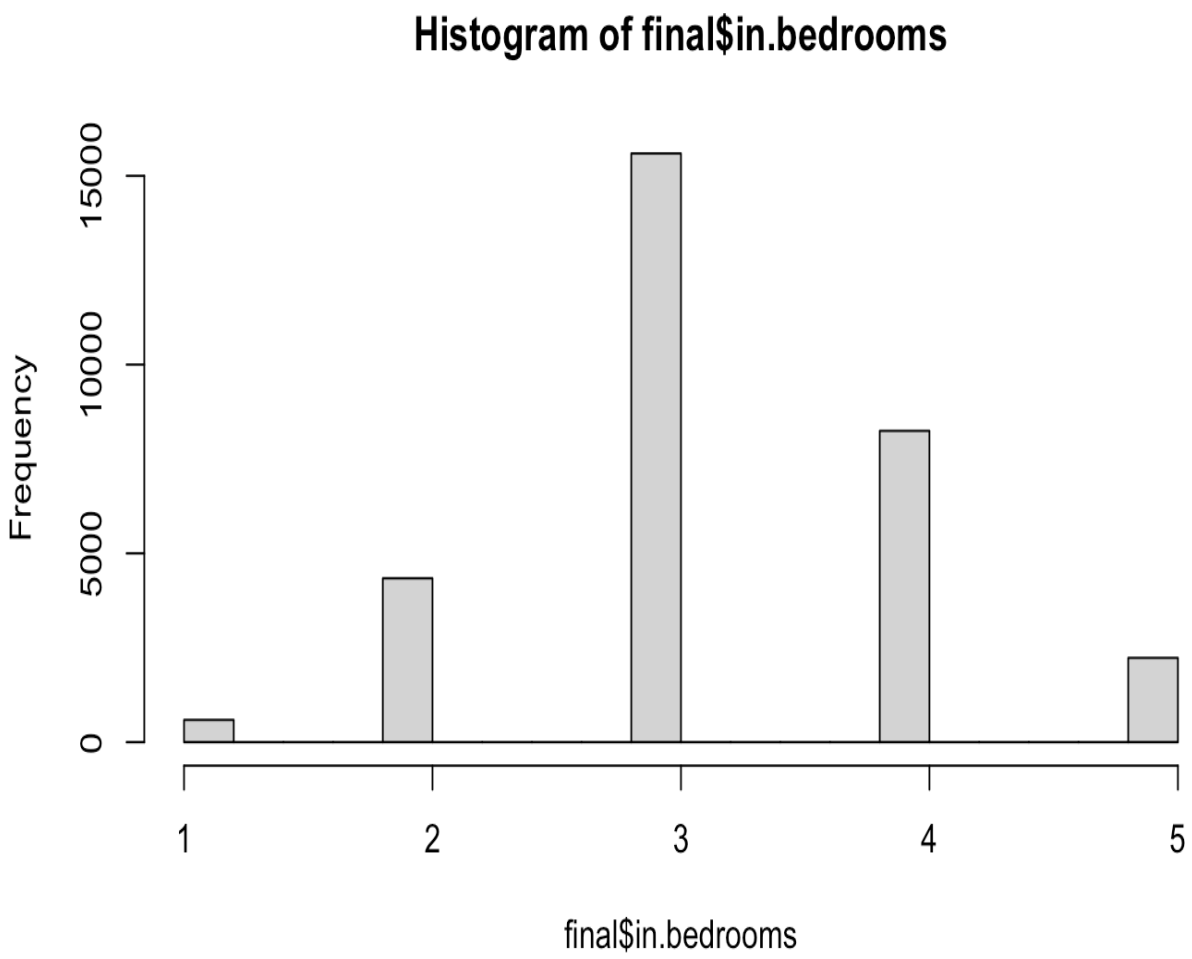


The graph of in.sqft vs frequency is a right skewed graph. As we can see in the below graph,

the value of in.sqft falls in between 0 to 3500 and 8000 to 8500. The highest frequency is 8000 which falls under 1500 to 2000 in.sqft. Also, the second highest frequency i.e. 7500 comes under 1000 to 1500 in.sqft. Similarly, frequencies like 6000, 3500, 2500 and 2000 come under 500 to 3500 in.sqft. The frequency between 3500 to 8000 is zero.



The graph represented further explains the value of the total number of bedrooms in a house to the total number of houses. 15000 houses contain 3 bedrooms in it. Moreover, the number of houses containing 4 bedrooms is up to 9000. The number of houses having 1 bedroom is around 1000.


Histogram of final$in.bedrooms
Frequency
0
5000
10000
15000
1
2
3
4
5
final$in.bedrooms

- **Modeling the Data -**

After the initial data visualization, we began creating a linear regression model to predict the energy usage for a given hour in July as July tends to be the highest energy usage month. First we set a seed for reproducibility so that it could ensure that random processes (e.g., data splitting) produce the same results each time the code is run. Then we split data into training and test sets with the training set comprising 80% of the data and the testing set comprising 20%.

```
library(caret)
set.seed(123)  # For reproducibility
splitIndex <- createDataPartition(final_cleaned$daily_total_usage, p = .8, list = FALSE)
train <- final_cleaned[splitIndex,]
test <- final_cleaned[-splitIndex,]
```

After that, we created a multiple linear regression model using all variables in the dataframe to predict energy usage. But from the output of the summary of the model, not all the variables are strongly related with energy usage. So we picked up some variables and created the second model:

```
model_lm <- lm(daily_total_usage ~ ., data = final_cleaned)
summary(model_lm)
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.543 on 30920 degrees of freedom
Multiple R-squared:  0.5069,    Adjusted R-squared:  0.5057
F-statistic: 402.4 on 79 and 30920 DF,  p-value: < 2.2e-16
```

Just as the output shown above, the p-value is less than 0.05, we accepted this model. 50.57% of the data is accounting for the dependent variable. The output shows that the sqft of people’s houses, the climate of the zones people stay in, the electronic applications they use and people’s poverty level will influence the energy usage most. The results are shown below.

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	2.666e+02	1.192e+02	2.236	0.025346	*
date	-1.679e-02	6.710e-03	-2.502	0.012360	*
bldg_id	-2.002e-05	1.967e-06	-10.175	< 2e-16	***
in.sqft	5.196e-03	4.783e-05	108.635	< 2e-16	***
in.bedrooms	1.014e+00	7.366e-02	13.759	< 2e-16	***
in.building_america_climate_zoneMixed-Humid	5.064e+00	1.021e+00	4.962	7.03e-07	***
in.clothes_dryerElectric, 120% Usage	-2.554e+00	1.360e+00	-1.879	0.060289	.
in.clothes_dryerElectric, 80% Usage	1.422e+00	9.338e-01	1.522	0.127912	
in.clothes_dryerGas, 100% Usage	-1.781e+00	3.552e-01	-5.014	5.37e-07	***
in.clothes_dryerGas, 120% Usage	-7.714e+00	1.423e+00	-5.422	5.93e-08	***
in.clothes_dryerGas, 80% Usage	2.112e+00	1.048e+00	2.014	0.043985	*
in.clothes_dryerNone	2.025e+00	4.954e-01	4.088	4.37e-05	***
in.clothes_dryerPropane, 100% Usage	-3.521e+00	6.093e-01	-5.778	7.63e-09	***
in.clothes_dryerPropane, 120% Usage	-3.705e+00	1.688e+00	-2.195	0.028165	*
in.clothes_dryerPropane, 80% Usage	-8.226e+00	1.321e+00	-6.229	4.76e-10	***
in.clothes_washerEnergyStar, 120% Usage	5.557e+00	1.564e+00	3.554	0.000380	***
in.clothes_washerEnergyStar, 80% Usage	1.926e-01	1.173e+00	0.164	0.869600	
in.clothes_washerNone	-1.148e+00	6.932e-01	-1.656	0.097781	.
in.clothes_washerStandard, 100% Usage	2.203e+00	1.635e-01	13.473	< 2e-16	***
in.clothes_washerStandard, 120% Usage	5.746e+00	1.580e+00	3.638	0.000276	***
in.clothes_washerStandard, 80% Usage	2.548e-02	1.195e+00	0.021	0.982992	
in.cooking_rangeElectric, 120% Usage	6.692e+00	7.954e-01	8.413	< 2e-16	***
in.cooking_rangeElectric, 80% Usage	-3.986e+00	7.338e-01	-5.432	5.62e-08	***
in.cooking_rangeGas, 100% Usage	9.956e-01	2.045e-01	4.869	1.13e-06	***
in.cooking_rangeGas, 120% Usage	9.231e+00	8.101e-01	11.395	< 2e-16	***
in.cooking_rangeGas, 80% Usage	-1.881e+00	7.833e-01	-2.402	0.016323	*
in.cooking_rangeNone	-1.637e+00	5.571e-01	-2.938	0.003305	**
in.cooking_rangePropane, 100% Usage	2.148e+00	2.999e-01	7.161	8.20e-13	***
in.cooking_rangePropane, 120% Usage	1.085e+01	9.210e-01	11.786	< 2e-16	***
in.cooking_rangePropane, 80% Usage	-6.433e-01	8.605e-01	-0.748	0.454756	
in.countyG4500030	-9.511e-01	7.735e-01	-1.230	0.218828	
in.countyG4500050	-1.501e+00	1.241e+00	-1.209	0.226521	
in.countyG4500070	2.751e+00	7.626e-01	3.608	0.000309	***
in.countyG4500090	3.910e+00	1.234e+00	3.169	0.001533	**
in.countyG4500110	1.108e+01	1.139e+00	9.728	< 2e-16	***
in.countyG4500130	4.624e+00	7.603e-01	6.082	1.20e-09	***
in.countyG4500150	2.971e+00	7.725e-01	3.847	0.000120	***
in.countyG4500170	-8.511e-01	1.869e+00	-0.455	0.648857	
in.countyG4500190	5.286e+00	7.477e-01	7.070	1.58e-12	***
in.countyG4500210	9.794e-01	8.606e-01	1.138	0.255126	
in.countyG4500230	-2.242e+00	1.007e+00	-2.227	0.025929	*
in.countyG4500250	-1.493e+00	8.744e-01	-1.707	0.087775	.
in.countyG4500270	7.786e-01	1.123e+00	0.694	0.487936	
in.countyG4500290	3.529e+00	9.170e-01	3.848	0.000119	***
in.countyG4500310	-2.349e-01	8.763e-01	-0.268	0.788684	
in.countyG4500330	-4.422e+00	9.354e-01	-4.728	2.28e-06	***
in.countyG4500350	2.571e+00	8.157e-01	3.152	0.001621	**
in.countyG4500370	9.719e-01	1.010e+00	0.962	0.335969	
in.countyG4500390	-4.645e+00	1.018e+00	-4.562	5.07e-06	***
in.countyG4500410	3.646e-01	7.966e-01	0.458	0.647200	
in.countyG4500430	1.259e+00	8.586e-01	1.466	0.142610	
in.countyG4500450	-2.481e-01	7.306e-01	-0.340	0.734170	
in.countyG4500470	-6.953e+00	8.592e-01	-8.092	6.07e-16	***
in.countyG4500490	4.527e+00	1.014e+00	4.464	8.07e-06	***

```
in.countyG4500510                          1.762e+00  7.538e-01   2.338 0.019412 *
in.countyG4500530                                 NA         NA      NA       NA
in.countyG4500550                         -6.227e-02  8.339e-01  -0.075 0.940476
in.countyG4500570                         -4.751e+00  8.116e-01  -5.854 4.86e-09 ***
in.countyG4500590                          1.450e+00  8.672e-01   1.672 0.094532 .
in.countyG4500610                         -8.764e+00  1.238e+00  -7.078 1.50e-12 ***
in.countyG4500630                          1.480e+00  7.561e-01   1.958 0.050255 .
in.countyG4500650                         -2.545e+00  1.225e+00  -2.078 0.037703 *
in.countyG4500670                         -1.207e+01  1.050e+00 -11.489  < 2e-16 ***
in.countyG4500690                          1.793e+00  1.003e+00   1.788 0.073850 .
in.countyG4500710                         -5.331e+00  9.533e-01  -5.592 2.27e-08 ***
in.countyG4500730                         -2.945e-01  8.167e-01  -0.361 0.718377
in.countyG4500750                         -6.311e+00  7.952e-01  -7.936 2.15e-15 ***
in.countyG4500770                          1.963e+00  7.935e-01   2.474 0.013385 *
in.countyG4500790                         -1.668e+00  7.472e-01  -2.232 0.025597 *
in.countyG4500830                         -9.989e-01  7.413e-01  -1.348 0.177823
in.countyG4500850                         -3.863e+00  8.724e-01  -4.428 9.55e-06 ***
in.countyG4500870                         -5.196e-01  9.700e-01  -0.536 0.592176
in.countyG4500890                          7.871e-01  9.423e-01   0.835 0.403565
in.countyG4500910                         -1.617e+00  7.556e-01  -2.140 0.032342 *
in.federal_poverty_level100-150%           2.029e+00  2.796e-01   7.256 4.08e-13 ***
in.federal_poverty_level150-200%          -1.501e+00  2.512e-01  -5.976 2.31e-09 ***
in.federal_poverty_level200-300%           2.630e+00  2.243e-01  11.727  < 2e-16 ***
in.federal_poverty_level300-400%           1.378e+00  2.243e-01   6.145 8.08e-10 ***
in.federal_poverty_level400%+              1.382e+00  1.987e-01   6.957 3.54e-12 ***
daily_temperature                          1.564e+00  4.946e-02  31.623  < 2e-16 ***
daily_humidity                            -1.634e-02  7.528e-03  -2.171 0.029972 *
```

The code `final_cleaned$daily_temperature <- final_cleaned$daily_temperature + 5` in R increases the values in the 'daily_temperature' column of the 'final_cleaned' dataset by 5. This operation is performed element-wise, adding 5 to each temperature entry. The `View(final_cleaned)` function is then used to display the modified dataset in a tabular view. Essentially, this code adjusts daily temperatures upward by 5 units, which could represent a temperature conversion or an arbitrary adjustment. The `View()` function facilitates visual inspection of the dataset changes after the modification.

	date	bldg_id	daily_total_usage	in.sqft	in.bedrooms	in.building_america_climate_zone	in.clothes_dryer	in.clothes_washer	in.cooking_range	in.county	in.federal_poverty_level	daily_temperature	daily_humidity
1	2018-07-01	65	32.14400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	51.73250	82.86500
2	2018-07-02	65	33.27000	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	51.92083	82.94208
3	2018-07-03	65	38.10300	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	54.37792	70.74667
4	2018-07-04	65	33.24400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	53.32917	71.63583
5	2018-07-05	65	33.32900	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	52.93542	71.94208
6	2018-07-06	65	30.00400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.69625	84.08500
7	2018-07-07	65	24.40400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	47.99375	88.96667
8	2018-07-08	65	30.90400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	48.19583	63.49625
9	2018-07-09	65	28.21900	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	48.56250	62.08750
10	2018-07-10	65	27.66400	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.69583	67.34125
11	2018-07-11	65	31.52700	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	52.53750	69.83792
12	2018-07-12	65	30.14600	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	52.47833	76.97375
13	2018-07-13	65	32.76200	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	51.19792	77.12792
14	2018-07-14	65	27.02500	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.97208	75.78042
15	2018-07-15	65	27.90700	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	51.88167	77.72792
16	2018-07-16	65	28.35100	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.42375	86.78667
17	2018-07-17	65	30.21000	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.48042	87.54875
18	2018-07-18	65	37.70700	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	52.21667	70.03000
19	2018-07-19	65	30.72200	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.16250	72.66125
20	2018-07-20	65	30.90000	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	50.24667	74.11125
21	2018-07-21	65	28.63600	885	3	Mixed-Humid	Gas, 100% Usage	Standard, 100% Usage	Electric, 100% Usage	G4500910	0-100%	49.98917	77.13958

The code utilises a linear regression model implemented in R to forecast energy demand. predicted_energy_demand - The predict() function is utilised in the predict(model_lm, newdata = final_cleaned) operation to generate predicted energy demand values from a linear regression model (model_lm) and a dataset (final_cleaned). Where predicted_energy_demand is the variable containing the results.

The peak energy demand is calculated in the following lines using peak_energy_demand - max(predicted_energy_demand) to identify the utmost value within the predicted energy demand vector. By printing the utmost value with peak_energy_demand, the result is 118.4717.

The code essentially utilises a trained linear regression model (model_lm) to predict energy demand by utilising the final_cleaned dataset's input data. The utmost value in the predictions signifies the peak energy demand, which is determined by storing and analysing the predicted values. The value of 118.4717 represents the peak energy demand that the model forecasts for the provided dataset. This data may prove to be of the utmost importance when it comes to capacity planning and resource allocation in situations involving infrastructure planning or energy management.

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

```{r}
#The code predicts energy usage using a linear regression model, while finding the maximum predicted energy demand using the max function from the model.
predicted_energy_demand <- predict(model_lm, newdata = final_cleaned)
peak_energy_demand <- max(predicted_energy_demand)
peak_energy_demand
```

```
[1] 118.4717
```

The provided code outlines a method for forecasting energy consumption in distinct climate zones, specifically targeting "Hot-Humid" and "Mixed-Humid" regions. The initial step involves filtering the dataset to exclusively encompass observations within the "Hot-Humid" climate zone. Subsequently, a linear regression model is applied to predict energy consumption within this climatic context. The calculated predictions are then analysed to identify the peak

energy demand, elucidating the model's estimate of maximum energy usage in the specified climatic conditions.

In a parallel vein, the dataset is also filtered to isolate observations occurring in the "Mixed-Humid" climate zone. Employing the same linear regression model, energy consumption predictions are generated for this particular region. The subsequent determination of the peak energy demand unveils the highest anticipated energy consumption within the "Mixed-Humid" climate zone. Notably, the values obtained from these predictions serve as key indicators of the model's proficiency in estimating peak energy usage for distinct climatic settings.

Upon execution of the code, the output furnishes valuable insights into the model's predictive capabilities. For the "Hot-Humid" climate zone, the highest predicted energy demand is quantified at 95.00906, while the "Mixed-Humid" region exhibits a slightly lower peak demand at 94.65602. These values, representing the pinnacle of projected energy consumption, offer a nuanced understanding of the model's performance across diverse climatic scenarios. It is imperative to underscore that these predictions, while informative, should ideally be validated against real-world data to ascertain the model's accuracy and generalizability beyond the training dataset. The nuanced interplay of climatic factors and their impact on energy usage is encapsulated in these peak demand predictions, enriching our comprehension of the model's efficacy in forecasting energy needs in specific climate zones.

```
g. Show future peak energy demand in total (for an hour):
    a. For different geographic regions
    b. For other dimensions /attributes you think important
```

```{r}
hot_humid_region <- final_cleaned %>%
  filter(in.building_america_climate_zone == "Hot-Humid")
mixed_humid_region <- final_cleaned %>%
  filter(in.building_america_climate_zone == "Mixed-Humid")
predicted_energy_HH <- predict(model_lm, newdata = hot_humid_region)
peak_energy_demandHH <- max(predicted_energy_HH)
peak_energy_demandHH
predicted_energy_MH <- predict(model_lm, newdata = mixed_humid_region)
peak_energy_demandMH <- max(predicted_energy_MH)
peak_energy_demandMH
```

```
[1] 95.00906
[1] 94.65602
```

- **Shiny App-**

Now that we have described the data in detail and have used our best model, a condensed linear regression model, it is now important to give the CEO the ability to see for themself how each predictor affects energy production. Using the shiny package in R we created a shiny app where the user can adjust each predictor, then press the action button "predict" which then outputs the predicted energy usage amount. This is quite helpful for a CEO as they can see what types of residences use up the most energy. Using the code below, I was able to put input sliders and selectors for each predictor in the model we created.

```
ui <- fluidPage(
  titlePanel("Energy Prediction App"),
  sidebarLayout(
    sidebarPanel(
      # Add input controls for each predictor
      sliderInput("in_sqft", "Square Footage", min = min(train$in.sqft), max =
max(train$in.sqft), value = mean(train$in.sqft)),
      sliderInput("in_bedrooms", "Number of bedrooms", min =
min(train$in.bedrooms), max = max(train$in.bedrooms), value =
mean(train$in.bedrooms)),
      selectInput("in_building_america_climate_zone", "Climate Zone", choices =
unique(train$in.building_america_climate_zone), selected =
(unique(train$in.building_america_climate_zone)[1])),
      sliderInput("daily_temperature", "Temperature", min =
min(train$daily_temperature), max = max(train$daily_temperature), value =
mean(train$daily_temperature)),
    selectInput("in_clothes_dryer", "Clothes Dryer Type", choices =
(unique(train$in.clothes_dryer)), selected = (unique(train$in.clothes_dryer)[1])),
    selectInput("in_clothes_washer", "Clothes Washer Type", choices =
(unique(train$in.clothes_washer)), selected =
(unique(train$in.clothes_washer)[1])),
    selectInput("in_cooking_range", "Cooking Range", choices =
(unique(train$in.cooking_range)), selected = (unique(train$in.cooking_range)[1])),
    selectInput("in_county", "County", choices = (unique(train$in.county)),
selected = (unique(train$in.county)[1])),
    selectInput("in_poverty_level", "Poverty Level", choices =
(unique(train$in.federal_poverty_level)), selected =
(unique(train$in.federal_poverty_level)[1])),
     sliderInput("daily_humidity", "Humidity", min = min(train$daily_humidity), max
= max(train$daily_humidity), value = mean(train$daily_humidity))),
```

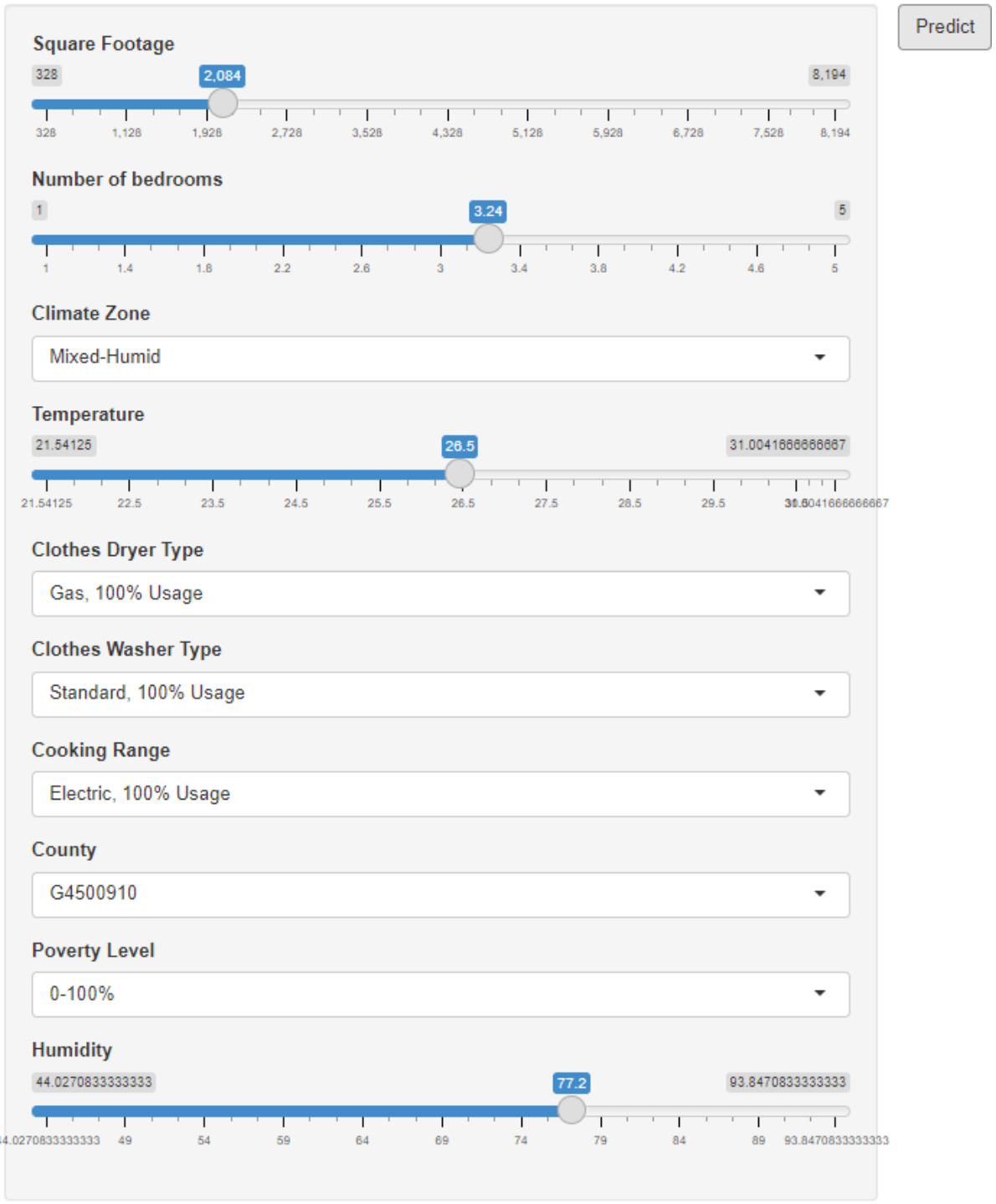
The first part of the code outlines each predictor that the user can adjust in order to predict energy usage. Using the functions "selectInput" and "sliderInput" respectively they create two separate actions for the user. The "selectInput" gives the user a dropdown of different categories they can select. We used this function for our categorical variables such as what type of washer or dryer the household had or what type of climate the house is in. In addition, the

sliderInput function works with quantitative variables, where the user can use the slider to adjust variables such as temperature or humidity closer or farther from its maximum value within the data set.

```
server <- function(input, output) {
  # Function to run the model and make predictions
  predict_energy <- eventReactive(input$predictButton, {
    # Create a new data frame with user inputs
    new_data <- data.frame(
      in.sqft = input$in_sqft,
      in.bedrooms = input$in_bedrooms,
      in.building_america_climate_zone = input$in_building_america_climate_zone,
      daily_temperature = input$daily_temperature,
      in.clothes_dryer = input$in_clothes_dryer,
      in.clothes_washer = input$in_clothes_washer,
      in.cooking_range = input$in_cooking_range,
      in.county = input$in_county,
      in.federal_poverty_level = input$in_poverty_level,
      daily_humidity = input$daily_humidity
      # Add more variables as needed
    )

    # Make predictions
    predictions <- predict(model3, new_data)
    return(predictions)
  })

  # Output the predicted total usage
  output$prediction <- renderText({
    paste("Predicted Total Energy Usage: ", round(predict_energy(), 2))
  })


}
```

The second part of the code for our shiny app is matching the actions that the user chose with the output that they will get. First, we must create a new data frame with all of the inputs that were chosen by the user. Then, using that data we plug it into the “prediction” function in order to get the desired outcome: the predicted energy usage. The user can look to see how warmer weather areas, more humid areas, residences with bigger houses and higher incomes all affect daily energy. Here is a small sample of the app that we created:

Energy Prediction App



Predicted Total Energy Usage:

Predicted Total Energy Usage: 18.25

Shiny App Link - https://sjoshi12.shinyapps.io/Final_app_IDS/

- **Modeling the Impact -**

```r
# copy the data
modeling <- final_cleaned
head(modeling)
# reduce the sqft of each entry by 10% of its own value
modeling$in.sqft <- modeling$in.sqft * 0.9
# unique building IDs
unique_ids <- unique(modeling$bldg_id)

# function to update a subset of rows for each building ID
update_cooking_range <- function(df, bldg_id) {
  rows <- which(df$bldg_id == bldg_id)
  # Calculate the number of rows to update (25% of the rows for this bldg_id)
  n_update <- ceiling(length(rows) * 0.25)
  # Randomly select rows to update
  rows_to_update <- sample(rows, n_update)
  # Update the cooking range for these rows
  df$in.cooking_range[rows_to_update] <- "Electric, 80% Usage"
  return(df)
}

# apply the function to each building ID
for (id in unique_ids) {
  modeling <- update_cooking_range(modeling, id)
}

predicted_energy_demand2 <- predict(model_lm, newdata = modeling)
peak_energy_demand2 <- max(predicted_energy_demand2)
peak_energy_demand2
```

In the pursuit of understanding the effects of building attribute modifications on predicted outcomes, a rigorous data-driven methodology was employed. Through the manipulation of our dataset, encompassing reductions in square footage and alterations in cooking range data, we simulated potential changes in building features. Leveraging predictive modeling techniques, specifically utilizing 'model_lm,' we assessed the impact of these modifications on the predicted outcome, which in our case focused on estimating energy demand. The observed shift in predicted peak energy demand, derived from empirical data and modeling outputs, illustrates the potential influence of these changes on energy consumption patterns. This data-centric analysis aims to provide actionable insights grounded in empirical evidence, contributing to informed decision-making processes.

● Potential Approach to Reduce Peak Energy Demand-

- Enhanced energy efficiency was promoted by advocating for the use of energy-saving washers and dryers, while encouraging off-peak utilization to reduce energy consumption during high-demand periods.
- Encouraged the adoption of energy-efficient cooking appliances and scheduled cooking activities during off-peak hours to minimize energy usage associated with stoves.
- Improved insulation and HVAC efficiency, especially in regions with mixed-humid climates, alleviating the impact of temperature fluctuations on energy demand.
- Advocated for smaller, energy-efficient home designs and efficient utilization of space in larger homes to counteract the higher energy consumption linked to larger square footage and more bedrooms.
- Implemented targeted incentives, such as subsidies, for affluent households to invest in renewable energy solutions, thus offsetting their greater energy consumption.
- Implementing solar panels presents a promising approach to curbing peak energy usage. By leveraging solar-generated electricity during daylight hours, this strategy aims to 'peak shave' by decreasing reliance on conventional grid sources at times of high demand. Through load offsetting, excess generation, and potential participation in demand response programs facilitated by solar energy storage systems, the integration of solar panels offers a viable solution to alleviate strain on the grid during peak periods.

- **Conclusion :-**

In conclusion, our technical paper thoroughly explores the dynamics of energy consumption and savings by employing a comprehensive dataset to construct robust linear regression models. The developed Shiny app serves as an interactive tool, aiding users in understanding the impact of various predictors on energy usage, particularly during peak demand in July. Our research underscores the significance of factors such as dwelling size, climate conditions, and socioeconomic elements in shaping energy consumption trends. Furthermore, our suggestions for mitigating peak energy demand encompass behavioral adjustments, adoption of energy-efficient technologies, and the incorporation of solar panels. Through offering practical insights and a user-friendly resource, our paper contributes to well-informed decision-making aimed at promoting sustainable and efficient energy practices.

- **Work Log :-**

Kunal Ahirrao – Coding, Shiny app, and document

Sukhad Dnyanesh Joshi - Coding, Modeling and document

Marina – Shiny App, Presentation, document

Jackson – Coding, Visualizations, document

Het – Document

Charlotte- Document, Shiny App