Cleaning-EDA-Feature-Engineering

2023-12-02

```
library (arrow)
```

```
## Warning: package 'arrow' was built under R version 4.3.2
```

```
##
## Attaching package: 'arrow'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ lubridate::duration() masks arrow::duration()
## ✖ dplyr::filter()       masks stats::filter()
## ✖ dplyr::lag()          masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(writexl)
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.3.2
```

```
static_housing <- read_parquet("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/
static_house_info.parquet")
#str(static_housing)
write_xlsx(static_housing, "static_housing.xlsx") #writng to excel for easier access (time co
nsuming to pull repititively)
meta_data <- read_csv("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/data_dict
ionary.csv")
```

```
## New names:
## Rows: 269 Columns: 7
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (7): field_location, field_name, data_type, units, field_description, al...
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## • `` -> `...7`
```

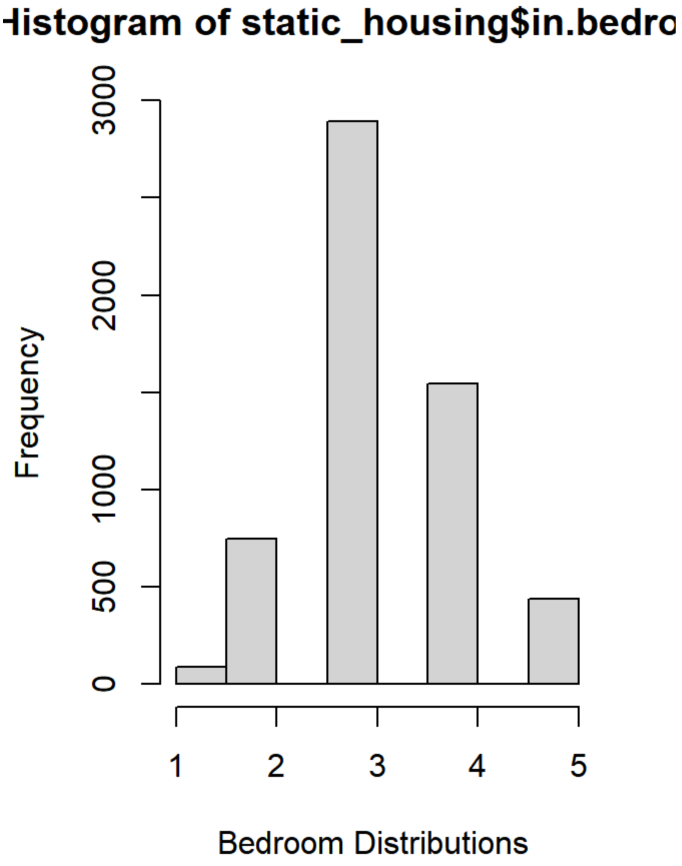
1. Cleaning static_housing dataset

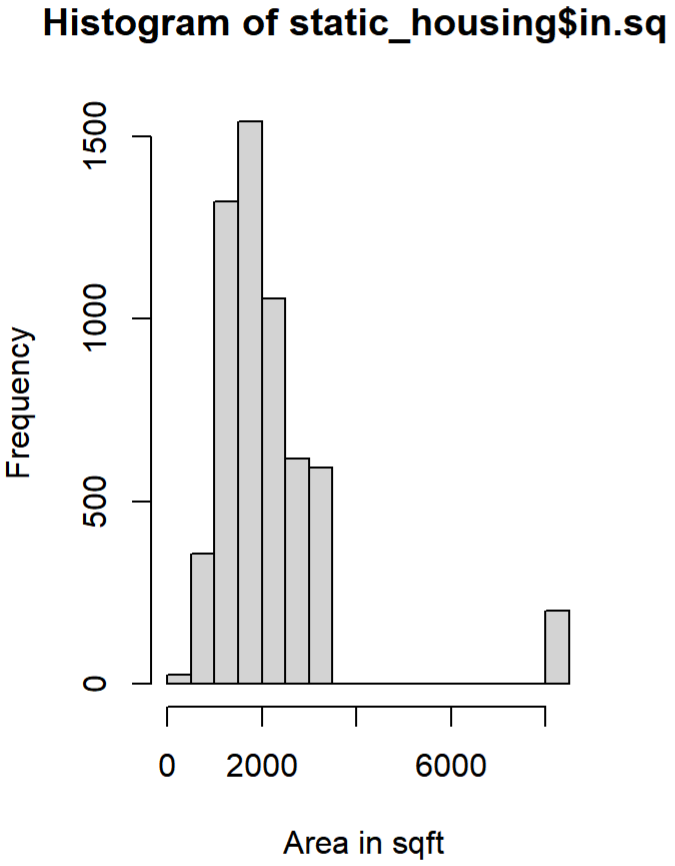
```
# Checking for missing values (NAs) in static_housing
#nas <- sapply(static_housing, function(x) sum(is.na(x)))
#print(nas)

cols_with_na <- names(static_housing)[colSums(is.na(static_housing)) > 1]
# Display columns with more than one NA and since these are none we don't have to take any ac
tions
print(cols_with_na)
```

```
## character(0)
```

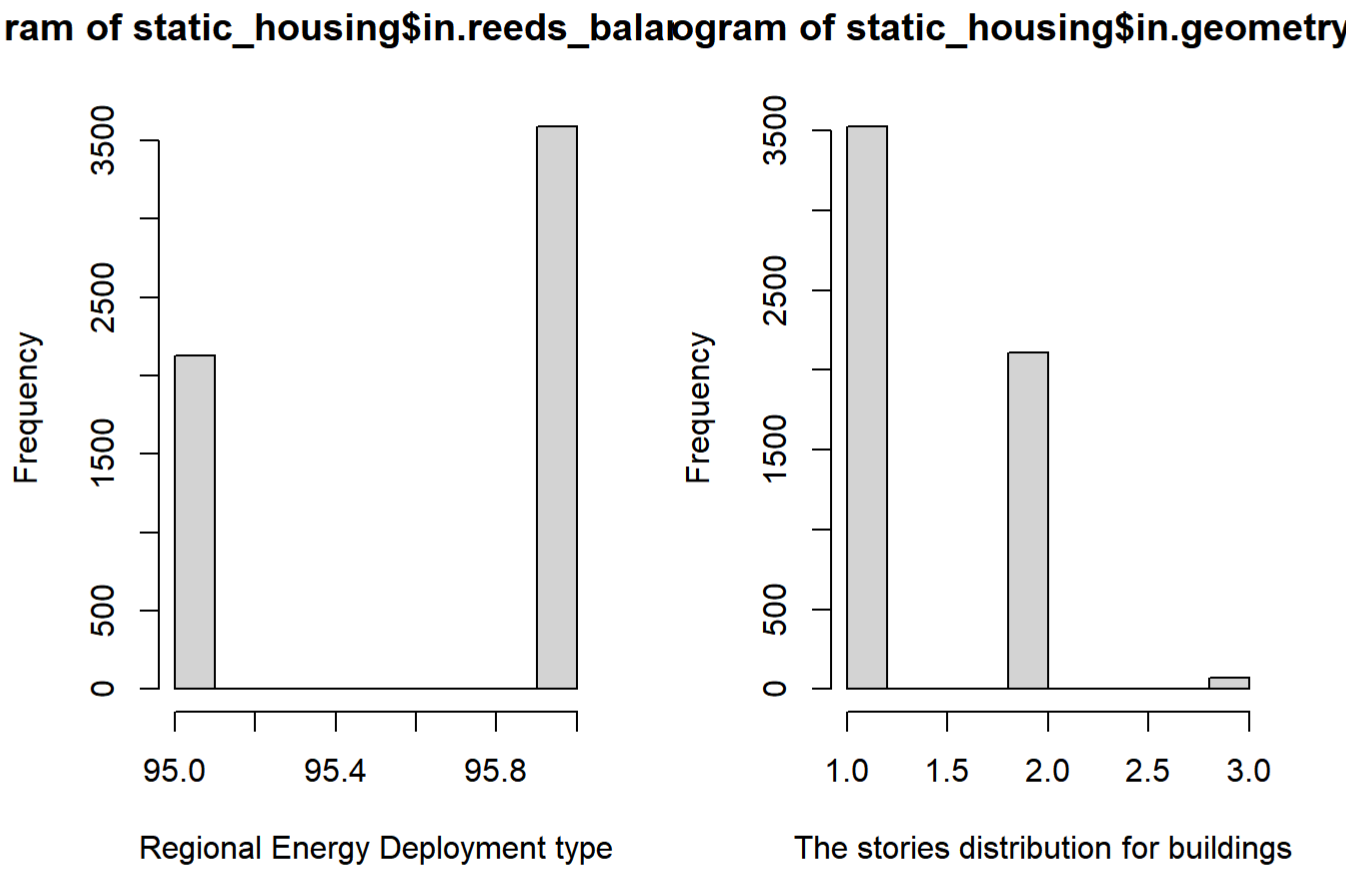
```
# house dataset
#commenting for a shorter document
#summary(static_housing)
```

```
# histograms of numeric values of interest
par(mfrow = c(1, 2))

hist(static_housing$in.bedrooms, xlab="Bedroom Distributions ") #shows a roughly normal distr
ibution
#this graph is inline with what we think
hist(static_housing$in.sqft, xlab="Area in sqft ")
```





```
# although this is an important variable through research it might show insignificant in the model we will keep the varialbe for further testing
hist(static_housing$in.reeds_balancing_area, xlab="Regional Energy Deployment type")



#plot(static_housing$in.sqft~static_housing$in.reeds_balancing_area)
hist(static_housing$in.geometry_stories, xlab="The stories distribution for buildings ")
```



```
# Initialize an empty array to store columns with only one unique value after removing blanks
output_cols <- c()


# Loop through columns in the static_housing dataset
for (col in names(static_housing)) {
  non_blank_values <- na.omit(static_housing[[col]])
  non_blank_values <- non_blank_values[non_blank_values != ""] # Remove blank values
  if (length(unique(non_blank_values)) == 1) {
    output_cols <- c(output_cols, col)
  }
}

# Display columns with only one unique value after removing blanks
length(output_cols)
```

```
## [1] 78
```

```
# Remove columns with only one unique value from the static_housing dataset
static_housing_filtered <- static_housing[, !names(static_housing) %in% output_cols]

#look for blanks row wise

# Calculate percentage of blanks and non-blanks in each column
percent_blanks <- sapply(static_housing_filtered, function(x) mean(x == "") * 100)
percent_non_blanks <- 100 - percent_blanks

# Create a matrix with column names and their respective percentages of blanks and non-blanks
blanks_vs_values_matrix <- matrix(c(percent_blanks, percent_non_blanks), nrow = 2, byrow = TR
UE,
                                  dimnames = list(c("Percentage of Blanks", "Percentage of Va
lues"), names(static_housing_filtered)))

# Print the matrix
blanks_vs_values_matrix[, blanks_vs_values_matrix[1,] > 0]
```

```
##                      upgrade.water_heater_efficiency upgrade.clothes_dryer
## Percentage of Blanks                        0.910683              3.677758
## Percentage of Values                       99.089317             96.322242
##                      upgrade.cooking_range
## Percentage of Blanks               1.17338
## Percentage of Values              98.82662
```

```
#since they have low blanks we can try to do some sort of interpolation

# Display the updated dataset
write_xlsx(static_housing_filtered, "static_housing_filtered.xlsx")
str(static_housing_filtered)
```

```
## 'data.frame':    5710 obs. of  93 variables:
##  $ bldg_id                                   : int  65 121 500 504 581 590 670 736 862 952
...
##  $ in.sqft                                   : int  885 1220 1220 1690 1690 2176 885 2663
885 2663 ...
##  $ in.bathroom_spot_vent_hour                : chr  "Hour23" "Hour20" "Hour11" "Hour13"
...
##  $ in.bedrooms                               : int  3 2 3 3 3 2 2 4 2 3 ...
##  $ in.building_america_climate_zone          : chr  "Mixed-Humid" "Mixed-Humid" "Mixed-Hum
id" "Mixed-Humid" ...
##  $ in.ceiling_fan                            : chr  "Standard Efficiency" "None" "Standard
Efficiency" "Standard Efficiency" ...
##  $ in.city                                   : chr  "SC, Rock Hill" "Not in a census Plac
e" "Not in a census Place" "In another census Place" ...
##  $ in.clothes_dryer                          : chr  "Gas, 100% Usage" "Electric, 100% Usag
e" "Electric, 80% Usage" "Electric, 80% Usage" ...
##  $ in.clothes_washer                         : chr  "Standard, 100% Usage" "EnergyStar, 10
0% Usage" "Standard, 80% Usage" "EnergyStar, 80% Usage" ...
##  $ in.clothes_washer_presence                : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ in.cooking_range                          : chr  "Electric, 100% Usage" "Electric, 100%
Usage" "Gas, 80% Usage" "Electric, 80% Usage" ...
##  $ in.cooling_setpoint                       : chr  "72F" "76F" "70F" "70F" ...
##  $ in.cooling_setpoint_has_offset            : chr  "No" "No" "No" "Yes" ...
##  $ in.cooling_setpoint_offset_magnitude      : chr  "0F" "0F" "0F" "2F" ...
##  $ in.cooling_setpoint_offset_period         : chr  "None" "None" "None" "Night Setup +3h"
...
##  $ in.county                                 : chr  "G4500910" "G4500730" "G4500710" "G450
0790" ...
##  $ in.county_and_puma                        : chr  "G4500910, G45000502" "G4500730, G4500
0101" "G4500710, G45000400" "G4500790, G45000604" ...
##  $ in.dishwasher                             : chr  "None" "290 Rated kWh, 100% Usage" "No
ne" "318 Rated kWh, 80% Usage" ...
##  $ in.ducts                                  : chr  "10% Leakage, R-4" "30% Leakage, R-4"
"20% Leakage, R-8" "None" ...
##  $ in.federal_poverty_level                  : chr  "0-100%" "150-200%" "100-150%" "400%+"
...
##  $ in.geometry_attic_type                    : chr  "Vented Attic" "Vented Attic" "Vented
Attic" "Vented Attic" ...
##  $ in.geometry_floor_area                    : chr  "750-999" "1000-1499" "1000-1499" "150
0-1999" ...
##  $ in.geometry_floor_area_bin                : chr  "0-1499" "0-1499" "0-1499" "1500-2499"
...
##  $ in.geometry_foundation_type               : chr  "Slab" "Ambient" "Slab" "Slab" ...
##  $ in.geometry_garage                        : chr  "1 Car" "None" "1 Car" "None" ...
##  $ in.geometry_stories                       : int  1 1 1 2 1 2 1 2 1 2 ...
##  $ in.geometry_stories_low_rise              : int  1 1 1 2 1 2 1 2 1 2 ...
##  $ in.geometry_wall_exterior_finish          : chr  "Wood, Medium/Dark" "Aluminum, Light"
"Vinyl, Light" "Vinyl, Light" ...
##  $ in.geometry_wall_type                     : chr  "Wood Frame" "Wood Frame" "Wood Frame"
"Wood Frame" ...
##  $ in.has_pv                                 : chr  "No" "Yes" "No" "No" ...
##  $ in.heating_fuel                           : chr  "Natural Gas" "Natural Gas" "Natural G
as" "Natural Gas" ...
##  $ in.heating_setpoint                       : chr  "70F" "65F" "70F" "68F" ...
##  $ in.heating_setpoint_has_offset            : chr  "No" "Yes" "No" "Yes" ...
##  $ in.heating_setpoint_offset_magnitude      : chr  "0F" "3F" "0F" "3F" ...
##  $ in.heating_setpoint_offset_period         : chr  "None" "Night -4h" "None" "Night -3h"
...
##  $ in.hot_water_fixtures                     : chr  "100% Usage" "100% Usage" "50% Usage"
"50% Usage" ...
##  $ in.hvac_cooling_efficiency                : chr  "AC, SEER 15" "AC, SEER 13" "AC, SEER
13" "None" ...
##  $ in.hvac_cooling_partial_space_conditioning: chr  "100% Conditioned" "100% Conditioned"
"100% Conditioned" "None" ...
##  $ in.hvac_cooling_type                      : chr  "Central AC" "Central AC" "Central AC"
"None" ...
##  $ in.hvac_has_ducts                         : chr  "Yes" "Yes" "Yes" "No" ...
##  $ in.hvac_has_zonal_electric_heating        : chr  "No" "No" "No" "No" ...
##  $ in.hvac_heating_efficiency                : chr  "Fuel Furnace, 92.5% AFUE" "Fuel Furna
ce, 60% AFUE" "Fuel Furnace, 76% AFUE" "Fuel Boiler, 80% AFUE" ...
##  $ in.hvac_heating_type                      : chr  "Ducted Heating" "Ducted Heating" "Duc
ted Heating" "Non-Ducted Heating" ...
##  $ in.hvac_heating_type_and_fuel             : chr  "Natural Gas Fuel Furnace" "Natural Ga
s Fuel Furnace" "Natural Gas Fuel Furnace" "Natural Gas Fuel Boiler" ...
##  $ in.income                                 : chr  "10000-14999" "15000-19999" "20000-249
99" "80000-99999" ...
##  $ in.income_recs_2015                       : chr  "<20000" "<20000" "20000-39999" "80000
-99999" ...
##  $ in.income_recs_2020                       : chr  "<20000" "<20000" "20000-39999" "60000
-99999" ...
##  $ in.infiltration                           : chr  "20 ACH50" "15 ACH50" "7 ACH50" "15 AC
H50" ...
##  $ in.insulation_ceiling                     : chr  "R-30" "R-13" "R-30" "R-13" ...
##  $ in.insulation_floor                       : chr  "None" "Uninsulated" "None" "None" ...
##  $ in.insulation_foundation_wall             : chr  "None" "None" "None" "None" ...
##  $ in.insulation_rim_joist                   : chr  "None" "None" "None" "None" ...
##  $ in.insulation_roof                        : chr  "Unfinished, Uninsulated" "Unfinished,
Uninsulated" "Unfinished, Uninsulated" "Unfinished, Uninsulated" ...
##  $ in.insulation_slab                        : chr  "Uninsulated" "None" "2ft R10 Under, H
orizontal" "Uninsulated" ...
##  $ in.insulation_wall                        : chr  "Wood Stud, Uninsulated" "Wood Stud, U
ninsulated" "Wood Stud, R-11" "Wood Stud, Uninsulated" ...
##  $ in.lighting                               : chr  "100% Incandescent" "100% LED" "100% L
ED" "100% LED" ...
##  $ in.misc_extra_refrigerator                : chr  "EF 17.6" "EF 17.6" "None" "None" ...
##  $ in.misc_freezer                           : chr  "EF 12, National Average" "None" "Non
e" "EF 12, National Average" ...
##  $ in.misc_gas_fireplace                     : chr  "None" "None" "None" "None" ...
##  $ in.misc_gas_grill                         : chr  "None" "None" "None" "None" ...
##  $ in.misc_gas_lighting                      : chr  "None" "None" "None" "None" ...
##  $ in.misc_hot_tub_spa                       : chr  "None" "None" "Gas" "None" ...
##  $ in.misc_pool                              : chr  "None" "None" "None" "None" ...
##  $ in.misc_pool_heater                       : chr  "None" "None" "None" "None" ...
##  $ in.misc_pool_pump                         : chr  "None" "None" "None" "None" ...
##  $ in.misc_well_pump                         : chr  "None" "None" "None" "None" ...
##  $ in.occupants                              : chr  "3" "1" "2" "2" ...
##  $ in.orientation                            : chr  "North" "West" "West" "North" ...
##  $ in.plug_load_diversity                    : chr  "100%" "100%" "50%" "50%" ...
##  $ in.puma                                   : chr  "G45000502" "G45000101" "G45000400" "G
45000604" ...
##  $ in.puma_metro_status                      : chr  "In metro area, not/partially in princ
```

```
ipal city" "Not/partially in metro area" "Not/partially in metro area" "In metro area, not/pa
rtially in principal city" ...
##  $ in.pv_orientation                         : chr  "None" "South" "None" "None" ...
##  $ in.pv_system_size                         : chr  "None" "7.0 kWDC" "None" "None" ...
##  $ in.range_spot_vent_hour                   : chr  "Hour14" "Hour17" "Hour16" "Hour6" ...
##  $ in.reeds_balancing_area                   : int  95 95 96 96 95 96 96 96 95 96 ...
##  $ in.refrigerator                           : chr  "EF 6.7, 100% Usage" "EF 17.6, 100% Us
age" "EF 19.9, 100% Usage" "EF 17.6, 100% Usage" ...
##  $ in.roof_material                          : chr  "Composition Shingles" "Composition Sh
ingles" "Composition Shingles" "Composition Shingles" ...
##  $ in.tenure                                 : chr  "Renter" "Owner" "Owner" "Owner" ...
##  $ in.usage_level                            : chr  "Medium" "Medium" "Low" "Low" ...
##  $ in.vacancy_status                         : chr  "Occupied" "Occupied" "Occupied" "Occu
pied" ...
##  $ in.vintage                                : chr  "1950s" "1950s" "2000s" "<1940" ...
##  $ in.vintage_acs                            : chr  "1940-59" "1940-59" "2000-09" "<1940"
...
##  $ in.water_heater_efficiency                : chr  "Natural Gas Standard" "Natural Gas St
andard" "Natural Gas Standard" "Natural Gas Standard" ...
##  $ in.water_heater_fuel                      : chr  "Natural Gas" "Natural Gas" "Natural G
as" "Natural Gas" ...
##  $ in.weather_file_city                      : chr  "Rock Hill York Co" "Oconee Co Rgnl"
"Columbia Metro" "Columbia Owens Apt" ...
##  $ in.weather_file_latitude                  : num  35 34.7 33.9 34 34.9 ...
##  $ in.weather_file_longitude                 : num  -81.1 -82.9 -81.1 -81 -82.2 ...
##  $ in.window_areas                           : chr  "F12 B12 L12 R12" "F18 B18 L18 R18" "F
18 B18 L18 R18" "F9 B9 L9 R9" ...
##  $ in.windows                                : chr  "Double, Low-E, Non-metal, Air, M-Gai
n" "Single, Clear, Non-metal" "Double, Low-E, Non-metal, Air, M-Gain" "Double, Low-E, Non-met
al, Air, M-Gain" ...
##  $ upgrade.water_heater_efficiency           : chr  "Electric Heat Pump, 50 gal, 3.45 UEF"
"Electric Heat Pump, 50 gal, 3.45 UEF" "Electric Heat Pump, 50 gal, 3.45 UEF" "Electric Heat
Pump, 50 gal, 3.45 UEF" ...
##  $ upgrade.clothes_dryer                     : chr  "Electric, Premium, Heat Pump, Ventles
s, 100% Usage" "Electric, Premium, Heat Pump, Ventless, 100% Usage" "Electric, Premium, Heat
Pump, Ventless, 80% Usage" "Electric, Premium, Heat Pump, Ventless, 80% Usage" ...
##  $ upgrade.hvac_heating_efficiency           : chr  "MSHP, SEER 24, 13 HSPF" "MSHP, SEER 2
4, 13 HSPF" "MSHP, SEER 24, 13 HSPF" "MSHP, SEER 29.3, 14 HSPF, Max Load" ...
##  $ upgrade.cooking_range                     : chr  "Electric, Induction, 100% Usage" "Ele
ctric, Induction, 100% Usage" "Electric, Induction, 80% Usage" "Electric, Induction, 80% Usag
e" ...
```

```
library(corrplot)
```

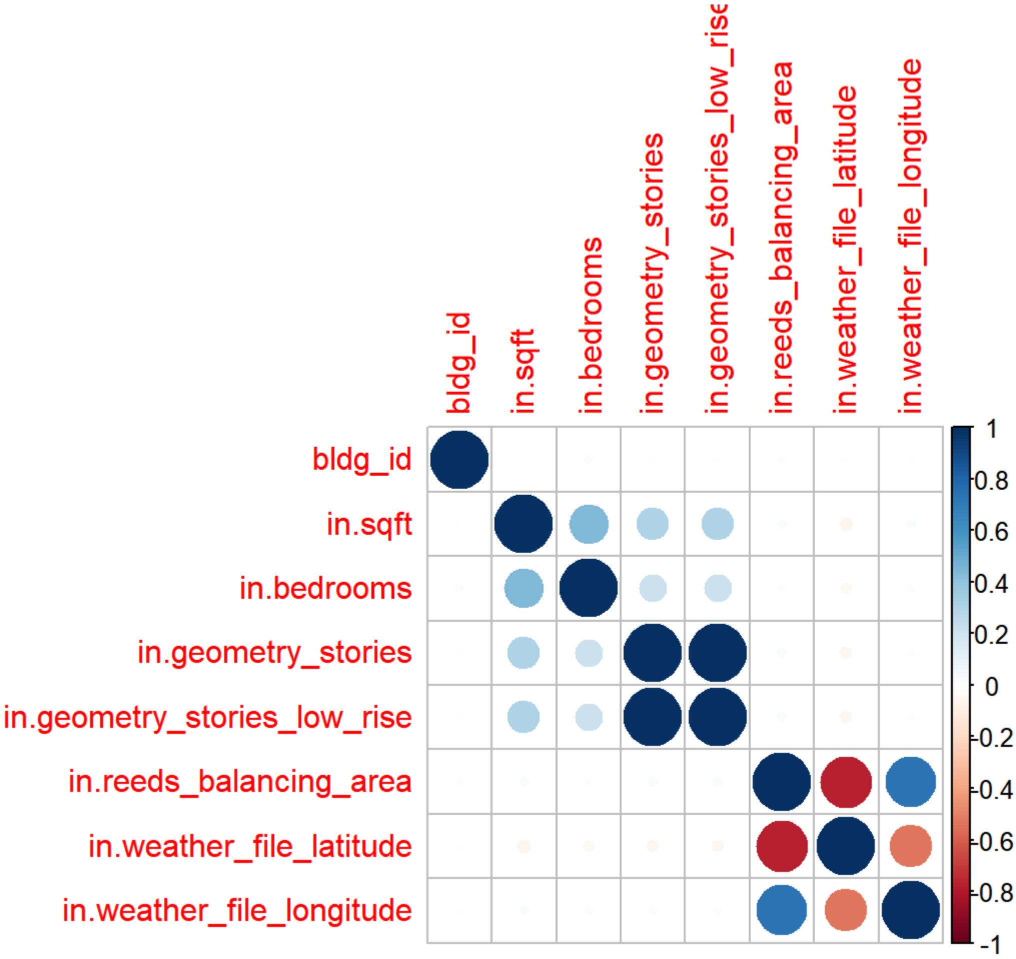
```
## Warning: package 'corrplot' was built under R version 4.3.2
```

```
## corrplot 0.92 loaded
```

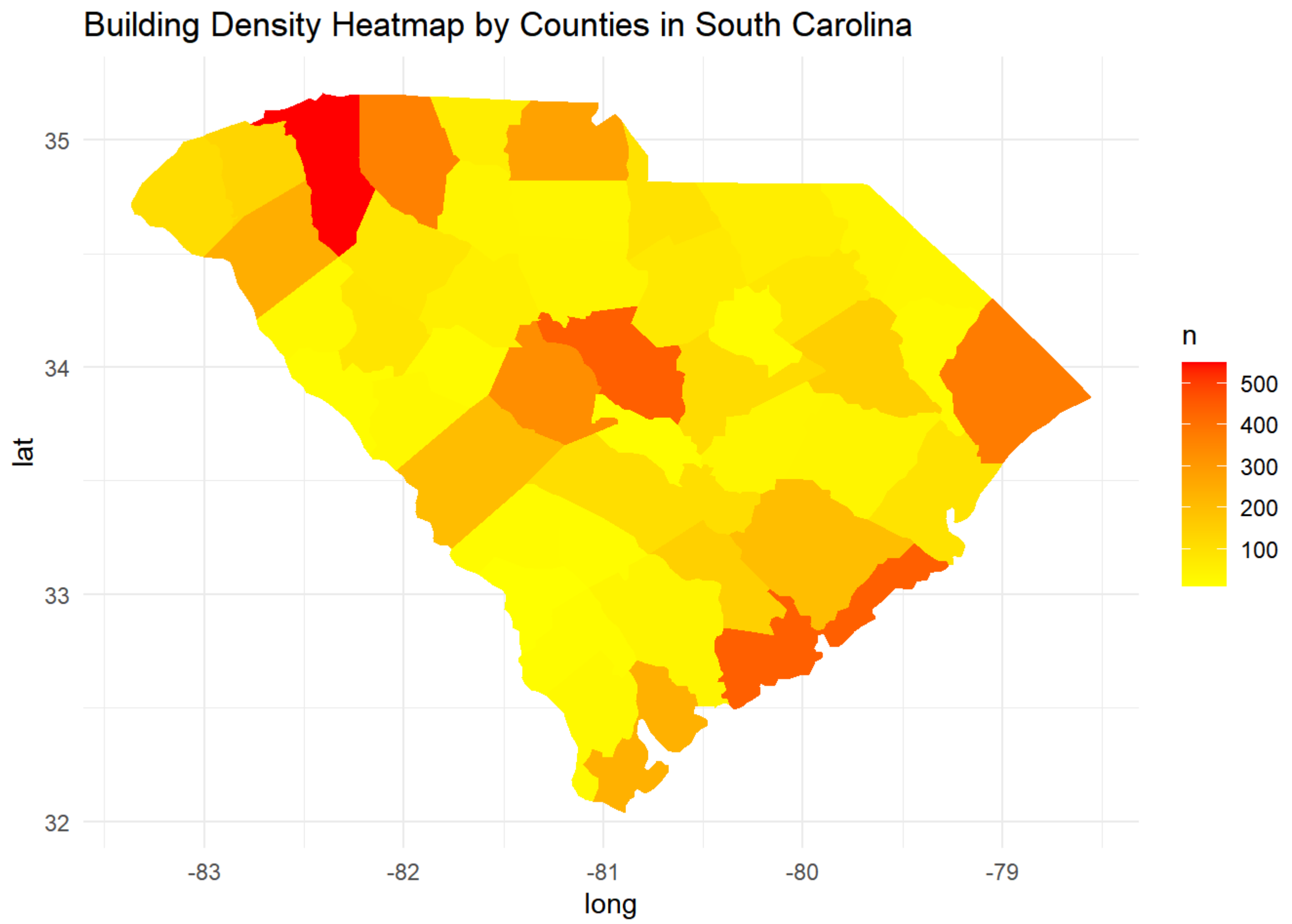
```
library(dplyr)



# Select numeric columns using select_if() and is.numeric()
numeric_cols <- static_housing_filtered %>%
  select_if(is.numeric)

# Select the 'county' column
county_col <- static_housing_filtered %>%
  select(in.county)

# Combining the 'county' column with numeric columns
result <- cbind(county_col, numeric_cols)
str(result)
```

```
## 'data.frame':    5710 obs. of  9 variables:
##  $ in.county                   : chr  "G4500910" "G4500730" "G4500710" "G4500790" ...
##  $ bldg_id                     : int  65 121 500 504 581 590 670 736 862 952 ...
##  $ in.sqft                     : int  885 1220 1220 1690 1690 2176 885 2663 885 2663 ...
##  $ in.bedrooms                 : int  3 2 3 3 3 2 2 4 2 3 ...
##  $ in.geometry_stories         : int  1 1 1 2 1 2 1 2 1 2 ...
##  $ in.geometry_stories_low_rise: int  1 1 1 2 1 2 1 2 1 2 ...
##  $ in.reeds_balancing_area     : int  95 95 96 96 95 96 96 96 95 96 ...
##  $ in.weather_file_latitude    : num  35 34.7 33.9 34 34.9 ...
##  $ in.weather_file_longitude   : num  -81.1 -82.9 -81.1 -81 -82.2 ...
```

```
#interesting to see here that for reeds we see a correlation for the area it is in hence we should keep this variable for further analysis and see if htis is something to do with region

correlation_matrix <- cor(result[, sapply(result, is.numeric)])
corrplot(correlation_matrix)
```


bldg_id
in.sqft
in.bedrooms
in.geometry_stories
in.geometry_stories_low_rise
in.reeds_balancing_area
in.weather_file_latitude
in.weather_file_longitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

```
# make a list with name of county vs the code given in the dataset
 ICPSRNAM = c("ABBEVILLE", "AIKEN", "ALLENDALE", "ANDERSON", "BAMBERG", "BARNWELL", "BEAUFOR
T", "BERKELEY", "CALHOUN", "CHARLESTON",
               "CHEROKEE", "CHESTER", "CHESTERFIELD", "CLARENDON", "COLLETON", "DARLINGTON",
"DILLON", "DORCHESTER", "EDGEFIELD",
               "FAIRFIELD", "FLORENCE", "GEORGETOWN", "GREENVILLE", "GREENWOOD", "HAMPTON",
"HORRY", "JASPER", "KERSHAW", "LANCASTER",
               "LAURENS", "LEE", "LEXINGTON", "MARION", "MARLBORO", "MCCORMICK", "NEWBERRY",
"OCONEE", "ORANGEBURG", "PICKENS",
               "RICHLAND", "SALUDA", "SPARTANBURG", "SUMTER", "UNION", "WILLIAMSBURG", "YOR
K")

GISJOIN = c("G4500010", "G4500030", "G4500050", "G4500070", "G4500090", "G4500110", "G450013
0", "G4500150", "G4500170", "G4500190",
              "G4500210", "G4500230", "G4500250", "G4500270", "G4500290", "G4500310", "G45003
30", "G4500350", "G4500370", "G4500390",
              "G4500410", "G4500430", "G4500450", "G4500470", "G4500490", "G4500510", "G45005
30", "G4500550", "G4500570", "G4500590",
              "G4500610", "G4500630", "G4500670", "G4500690", "G4500650", "G4500710", "G45007
30", "G4500750", "G4500770", "G4500790",
              "G4500810", "G4500830", "G4500850", "G4500870", "G4500890", "G4500910")

List_Name<-data.frame(tolower(ICPSRNAM),(GISJOIN))

# Group by 'in.county' and calculate the average of numeric columns
# Group by 'in.county' and calculate the average of numeric columns while counting bldg_id oc
currences
county_counts <- result %>%
  count(in.county,in.weather_file_latitude,in.weather_file_longitude)

county_counts$County_name<-List_Name$tolower.ICPSRNAM.[match(county_counts$in.county,List_Nam
e$X.GISJOIN.)]

# get a county map from the library ( of south caroline)
county_map <- map_data("county", region = "south carolina")
county_map$subregion<-tolower(county_map$subregion)
county_counts$in.county<-tolower(county_counts$County_name)



# Merge energy data with the county map
merged_data <- merge(county_map, county_counts, by.x = "subregion", by.y = "County_name", al
l.x = TRUE)
#merged_data
# Create the heatmap

ggplot(merged_data, aes(x = long, y = lat, group = group, fill = n)) +
  geom_polygon() +
  scale_fill_gradientn(colors = c("yellow", "red"), values = scales::rescale(c(0, 50, 100)))
+
  labs(title = "Building Density Heatmap by Counties in South Carolina") +
  theme_minimal()
```



Commenting out the code scraping the energy data for over 5.7 homes (takes over 15 minutes)

```
#commneting out the process to optimized computiong power, instead impoerting from an already saved file
# Lets Scrape the energy data

#
# bldg_ids <- unique(static_housing_filtered$bldg_id)
# #appending links
# links <- paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-houseData/", bldg_ids, ".parquet")
# #generating links
# data_df <- data.frame(bldg_id = bldg_ids, link = links)
# # Assuming data_df dataframe is created with bldg_id and link columns
# library(httr)
# # Create an empty list to store data frames
# parquet_data <- list()
#
#
#
# # Loop through each link and read Parquet files
# for (i in 1:nrow(data_df)) {
#     link <- as.character(data_df[i, "link"])
#     bldg_id <- as.character(data_df[i, "bldg_id"])
#
#
#     response <- GET(link)
#
# # Save the content to a temporary file
# temp_parquet <- tempfile(fileext = ".parquet")
# writeBin(content(response), temp_parquet)
#
# # Read the Parquet file into a dataframe
# df <- read_parquet(temp_parquet)
#
#
#     # Assign bldg_id to the first column
#     df$bldg_id <- bldg_id
#   df<-df%>%filter(month(df$time)==7)
#    # df<-df%>%filter(month(df$time) %in% c(5,6,7))
#     #df$month<-month(df$time)
#     # Add the dataframe to the list
#     parquet_data[[i]] <- df
#    cat("Progress: ", i, "/", nrow(data_df), "\n")
#
# }
#
# # Combine all data frames into a single data frame
#
# combined_data <- do.call(rbind, parquet_data)
# head(combined_data)
# combined_data_1<-combined_data
# #combined_data<-combined_dataf%>%filter(month(df$time)==7)
#
# combined_data$hour<-hour(combined_data$time)
# #head(combined_data$hour)
# #taking sum of all the out. energy for 30 days accross each hour
# aggregate_hourly<-combined_data%>%group_by(bldg_id,hour)%>%summarize(across(where(is.numeri
c), sum))
# head(aggregate_hourly)
#
# #write_xlsx(aggregate_hourly,"aggregate_hourly_Energy_Data.xlsx")
```

This is the energy data for all of july but on an hourly basis for all days of july by building id(

a summation of energy simply), we have written it to a file for easier access and save time of repitied preprocessing

merging happens here :

merged_house_Static_energy <- merge(static_housing_filtered,aggregate_hourly , by = “bldg_id”, all = TRUE)

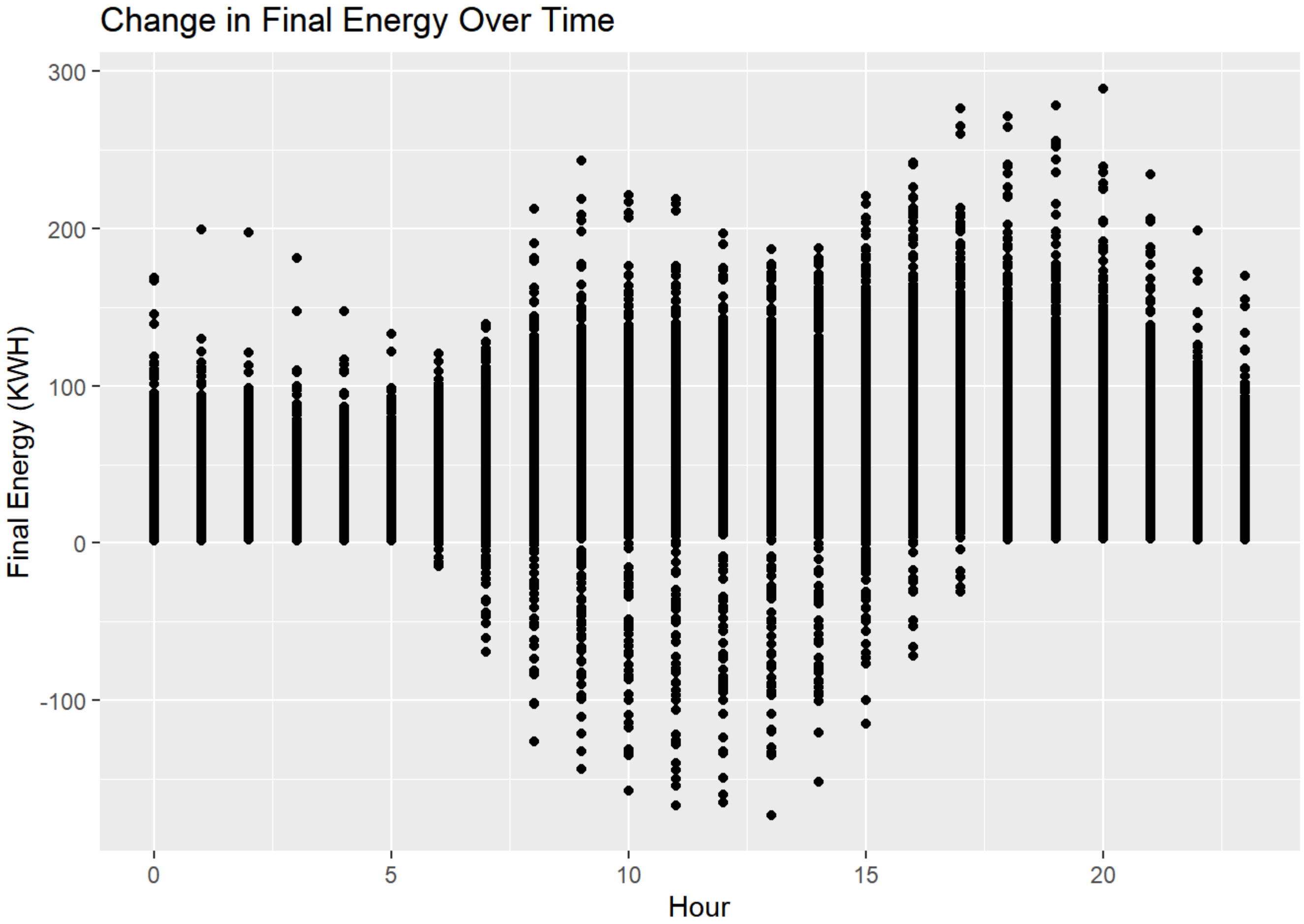
```
# library(tidyverse)
# library(writexl)
# library(readxl)
# aggregate_hourly<-read_xlsx("aggregate_hourly_Energy_Data.xlsx")
# #merging the information  by building id  to get all the categorical variables value sin 1
dataset
#
# head(merged_house_Static_energy)
# write_xlsx(merged_house_Static_energy,"merged_house_Static_energy.xlsx")
```

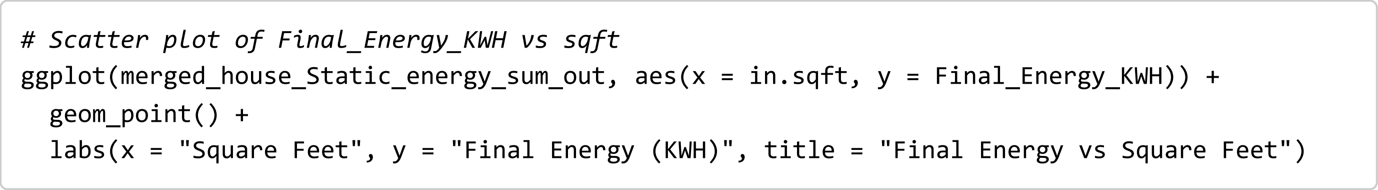
EDA on the merged Energy Data for all the buildings in july on an hours basis (i.e a row signifies 1pm for a building for all 30 days summation

```
merged_house_Static_energy<-read_xlsx("merged_house_Static_energy.xlsx")
#glimpse(merged_house_Static_energy)
#commenting for a better view
#glimpse(merged_house_Static_energy)
#grep("out.", names(merged_house_Static_energy))
out_cols <- c(grep("out.", names(merged_house_Static_energy)))
#out_cols`
```

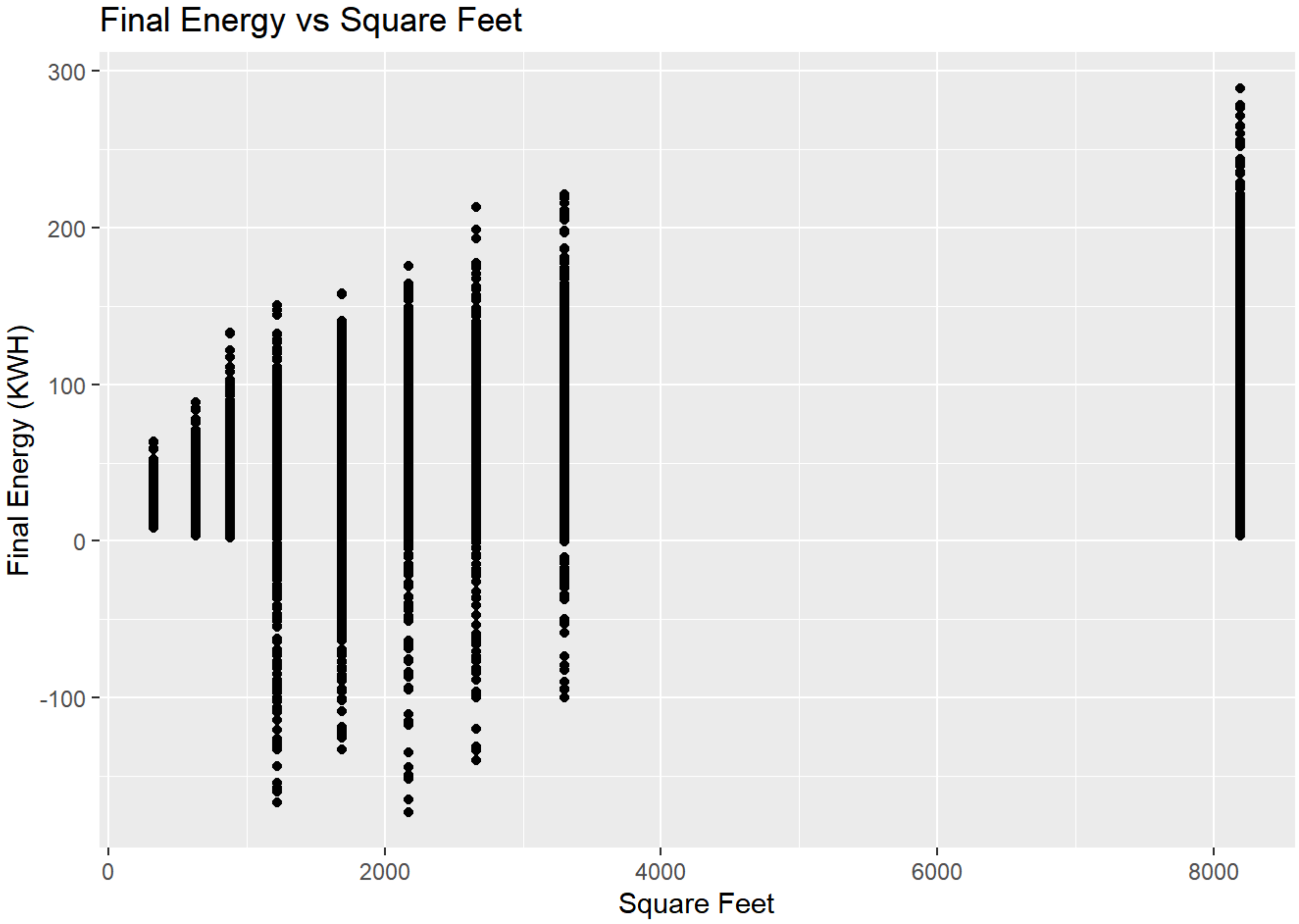
```
# assign to a new dataframe
merged_house_Static_energy_sum_out<-merged_house_Static_energy
#aggregating all the energy coloumns and summing to Final_enery_KWH
merged_house_Static_energy_sum_out$Final_Energy_KWH<- merged_house_Static_energy_sum_out %>%s
elect(starts_with("out")) %>% rowSums(na.rm = TRUE)#

# removing out coloumns
merged_house_Static_energy_sum_out<- merged_house_Static_energy_sum_out[, -out_cols]
#glimpse(merged_house_Static_energy_sum_out)
```

```
# Example: Create a line plot of Final_Energy_KWH over time
ggplot(merged_house_Static_energy_sum_out, aes(x = hour, y = Final_Energy_KWH)) +
  geom_point() +
  labs(x = "Hour", y = "Final Energy (KWH)", title = "Change in Final Energy Over Time")
```

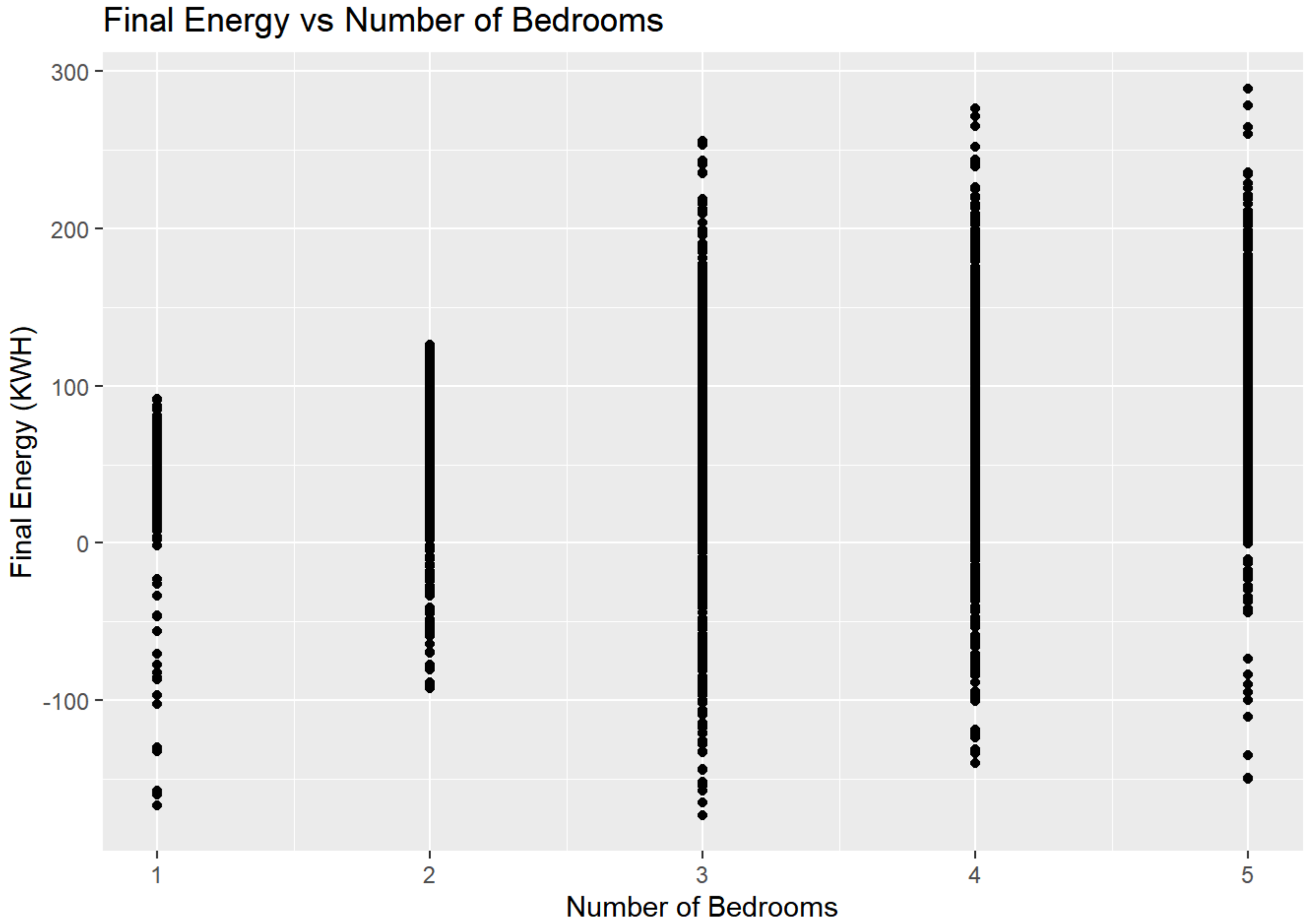




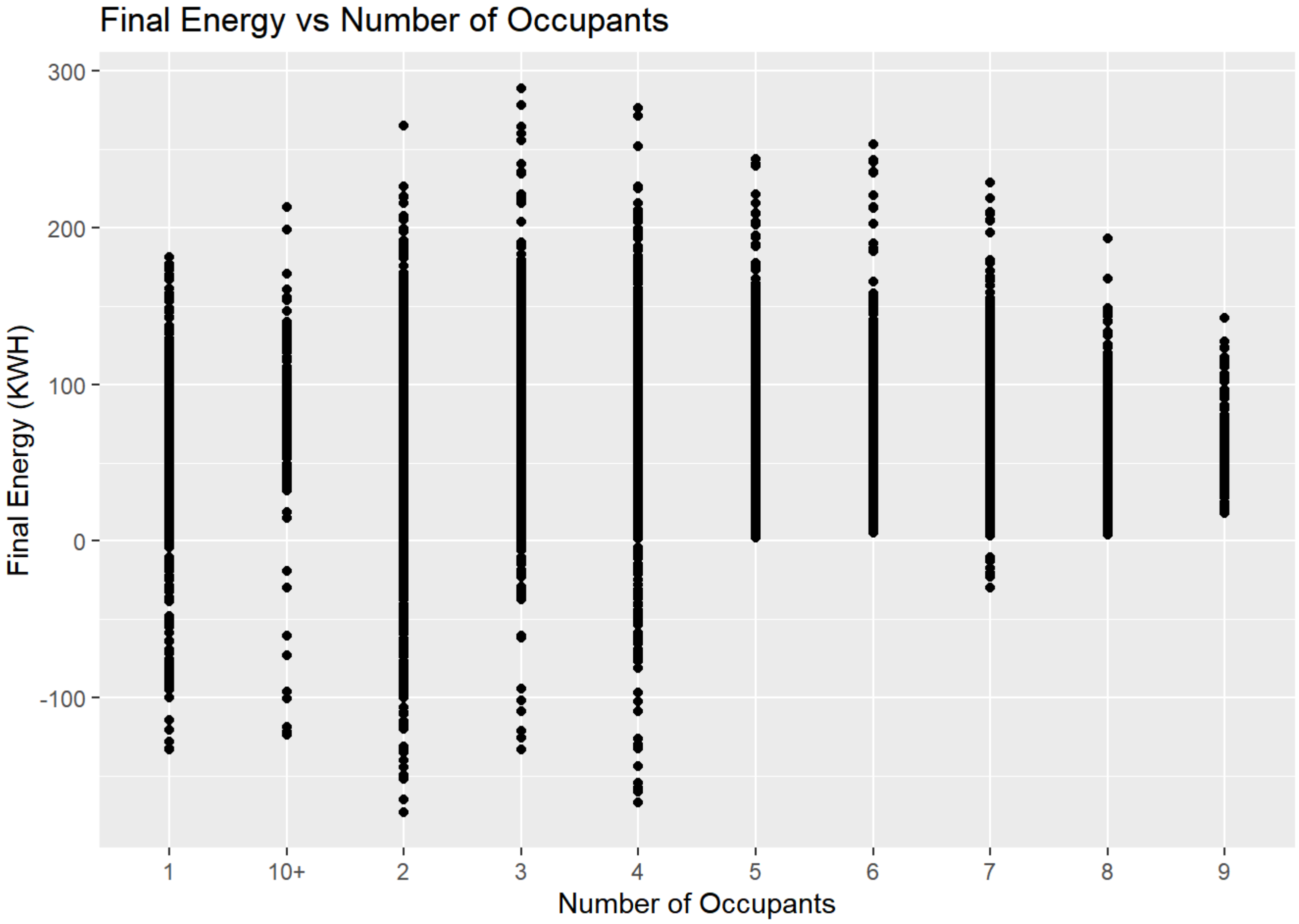
```
# Scatter plot of Final_Energy_KWH vs sqft
ggplot(merged_house_Static_energy_sum_out, aes(x = in.sqft, y = Final_Energy_KWH)) +
  geom_point() +
  labs(x = "Square Feet", y = "Final Energy (KWH)", title = "Final Energy vs Square Feet")
```



```
# Scatter plot of Final_Energy_KWH vs bedrooms
ggplot(merged_house_Static_energy_sum_out, aes(x = in.bedrooms, y = Final_Energy_KWH)) +
  geom_point() +
  labs(x = "Number of Bedrooms", y = "Final Energy (KWH)", title = "Final Energy vs Number of
Bedrooms")
```



```
# Scatter plot of Final_Energy_KWH vs occupants
ggplot(merged_house_Static_energy_sum_out, aes(x = in.occupants, y = Final_Energy_KWH)) +
  geom_point() +
  labs(x = "Number of Occupants", y = "Final Energy (KWH)", title = "Final Energy vs Number o
f Occupants")
```



```
numeric_subset <- merged_house_Static_energy_sum_out %>%
  select(bldg_id,in.occupants,in.county,hour,Final_Energy_KWH,in.sqft,in.bedrooms )  %>%group
_by(hour, in.county) %>%
  summarise(across(where(is.numeric) & !matches("Final_Energy_KWH"), mean, na.rm = TRUE),
            Final_Energy_KWH = sum(Final_Energy_KWH, na.rm = TRUE))
```

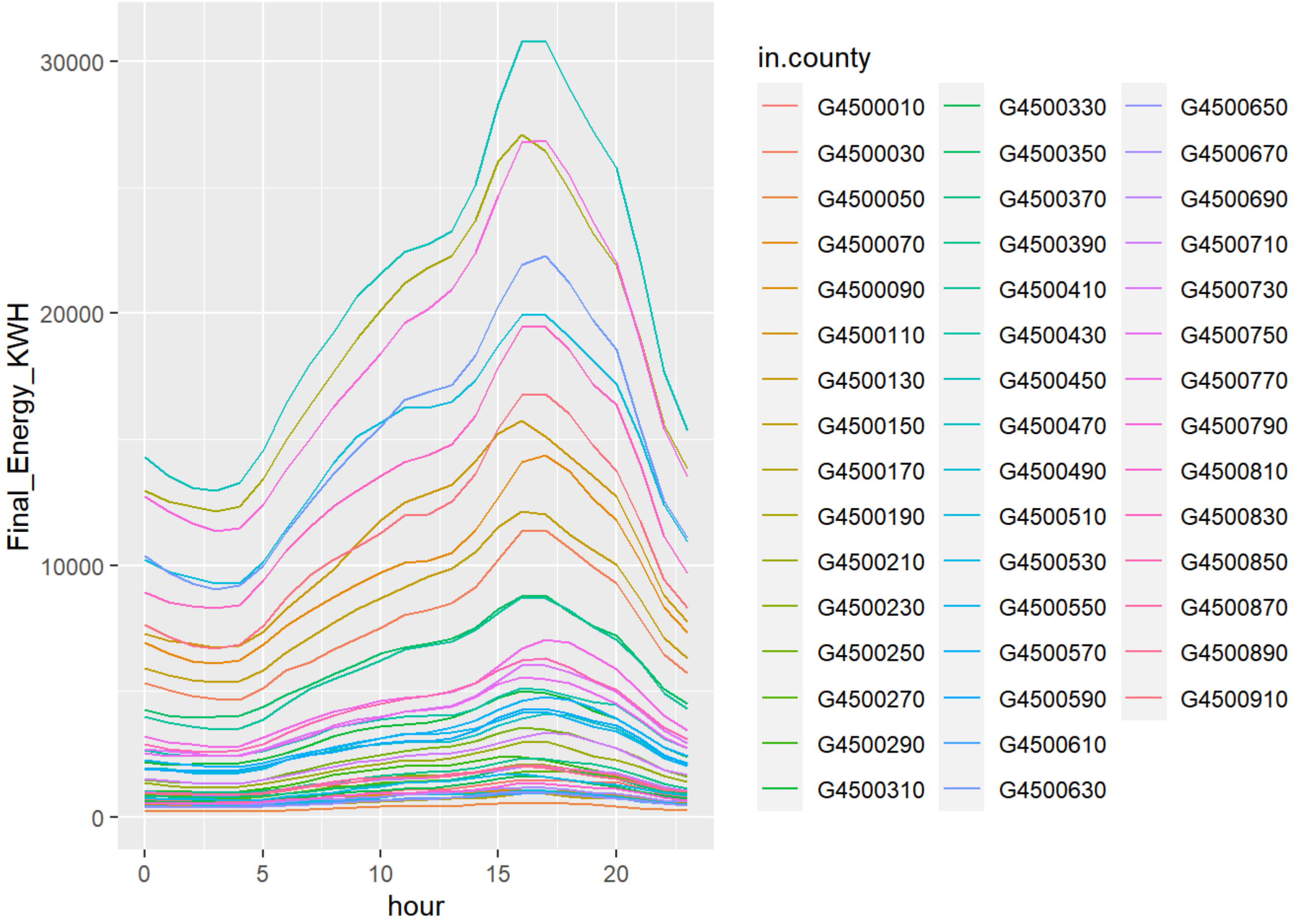
```
## Warning: There was 1 warning in `summarise()`.
## ℹ In argument: `across(...)`.
## ℹ In group 1: `hour = 0`, `in.county = "G4500010"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'hour'. You can override using the
## `.groups` argument.
```

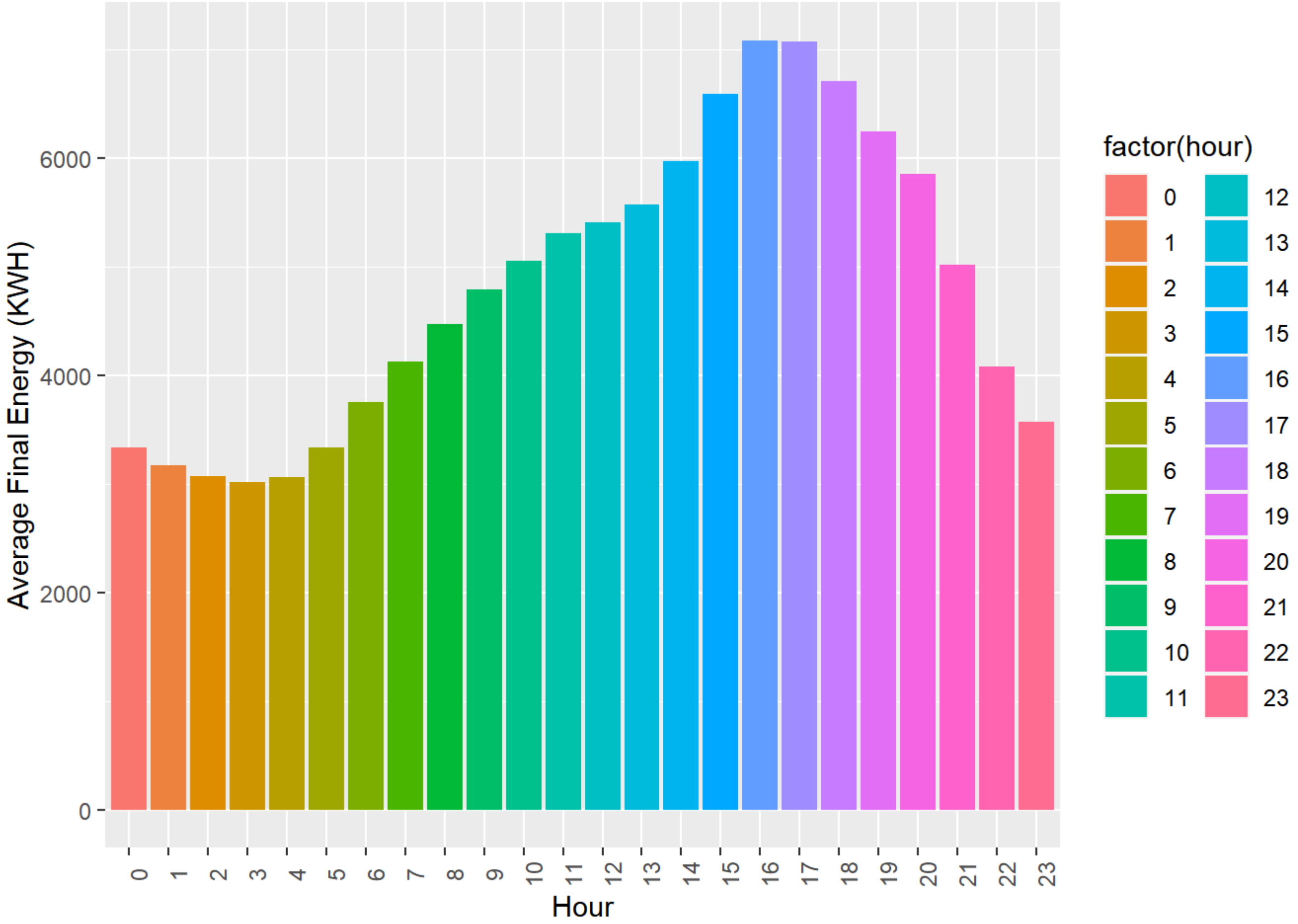
```
glimpse(numeric_subset)
```

```
## Rows: 1,104
## Columns: 6
## Groups: hour [24]
## $ hour             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
## $ in.county        <chr> "G4500010", "G4500030", "G4500050", "G4500070", "G450…
## $ bldg_id          <dbl> 272540.0, 277408.8, 213464.4, 272616.4, 289496.2, 260…
## $ in.sqft          <dbl> 2121.276, 1882.976, 2015.400, 2134.215, 1871.950, 194…
## $ in.bedrooms      <dbl> 3.344828, 3.087805, 3.400000, 3.369919, 3.050000, 3.0…
## $ Final_Energy_KWH <dbl> 712.5697, 5329.1406, 246.3126, 6911.0912, 480.7266, 5…
```

```
#######County Wise Analysis
library(ggplot2)

# Line Plot: Hour vs. Final_Energy_KWH for a single county
ggplot(data = numeric_subset, aes(x = hour, y = Final_Energy_KWH, group = in.county, color =
in.county)) +
  geom_line()
```

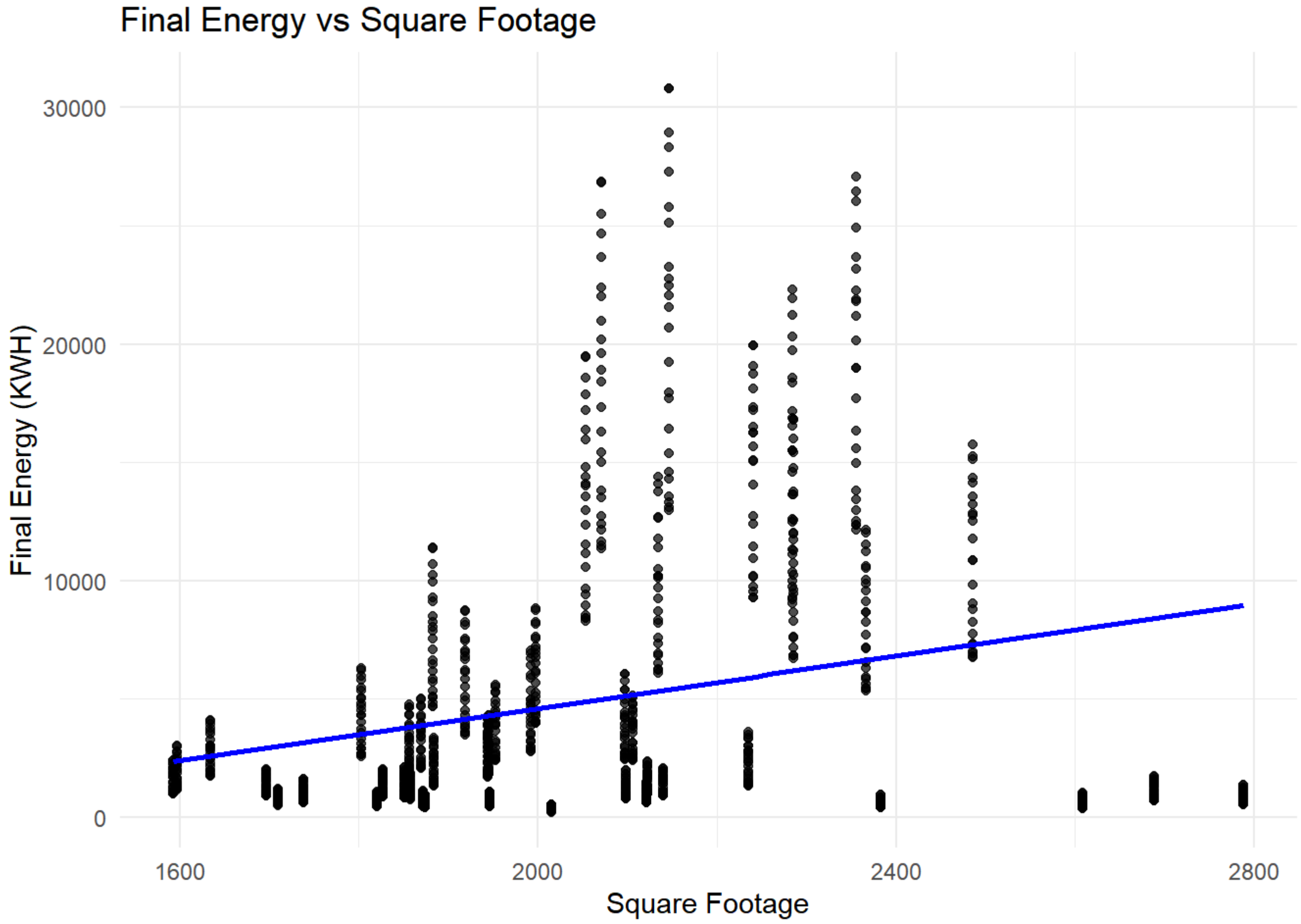


```
# Bar Chart: Average Final_Energy_KWH per hour across all counties
ggplot(data = numeric_subset, aes(x = factor(hour), y = Final_Energy_KWH, fill = factor(hou
r))) +
  stat_summary(fun = mean, geom = "bar") +
  labs(x = "Hour", y = "Average Final Energy (KWH)") +
  theme(axis.text.x = element_text(angle = 90))
```



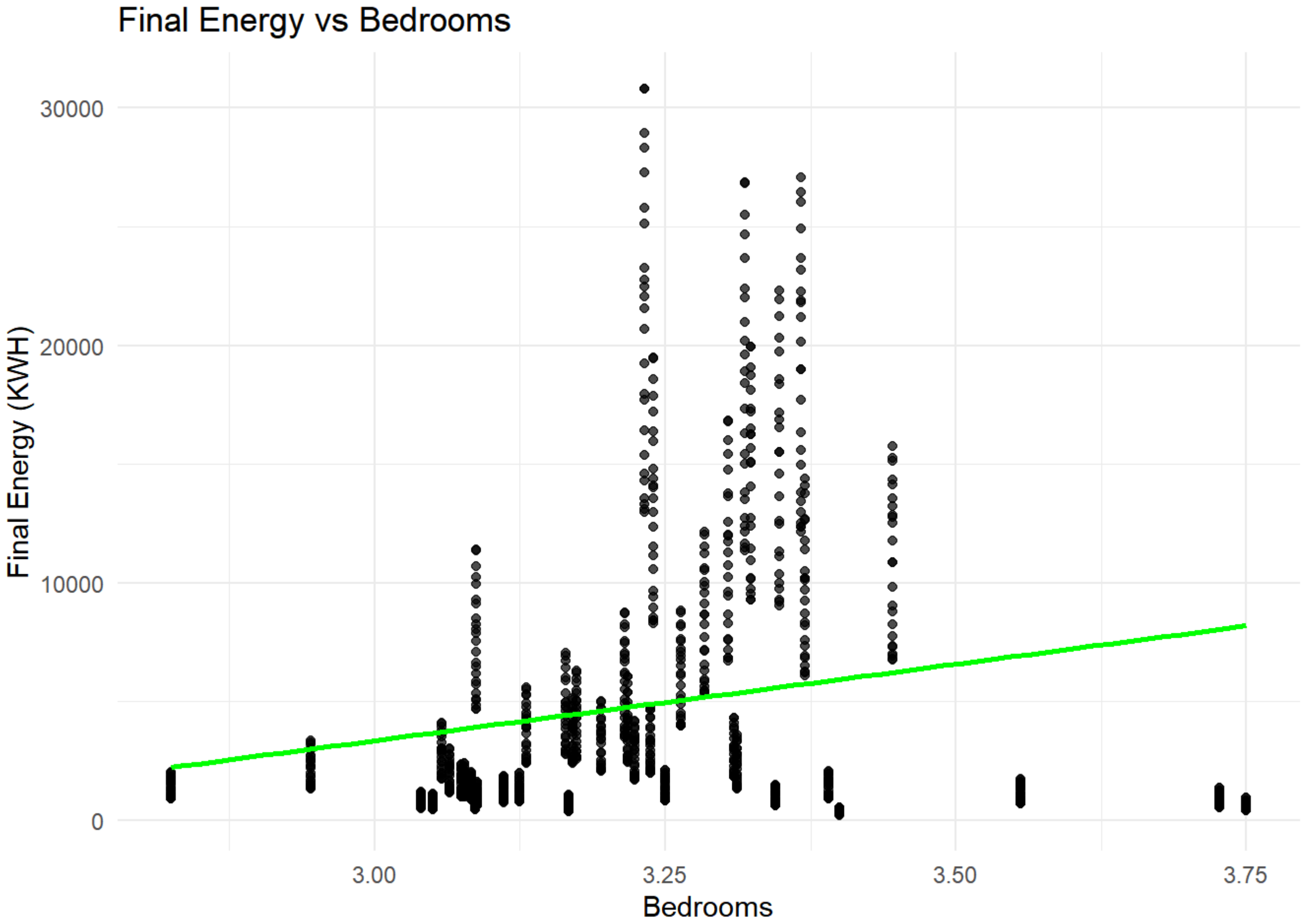
```
# Scatter plot with smooth trend line for Final_Energy_KWH vs in.sqft
ggplot(data = numeric_subset, aes(x = in.sqft, y = Final_Energy_KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Adding linear trend line
  labs(x = "Square Footage", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Final Energy vs Square Footage") +  # Title of the plot
  theme_minimal()  # Using minimal theme
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Scatter plot with smooth trend line for Final_Energy_KWH vs in.bedrooms
ggplot(data = numeric_subset, aes(x = in.bedrooms, y = Final_Energy_KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Bedrooms", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Final Energy vs Bedrooms") +  # Title of the plot
  theme_minimal()  # Using minimal theme
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
Merged_Final<-merged_house_Static_energy_sum_out
range(Merged_Final$Final_Energy_KWH)
```

```
## [1] -173.055  289.258
```

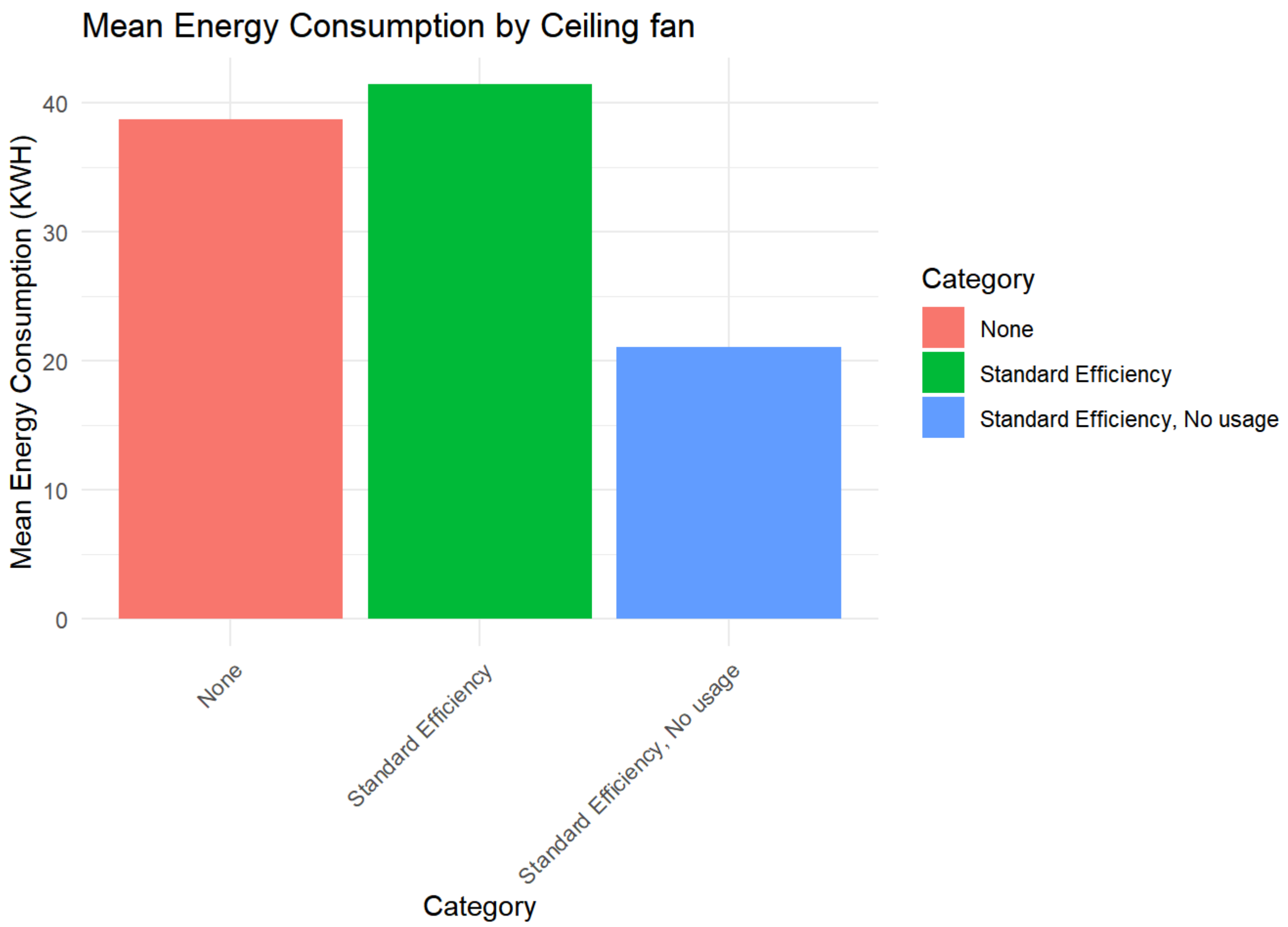
```
nrow(Merged_Final[Merged_Final$Final_Energy_KWH<0,] )# these buildings actually produce elect
ricity
```

```
## [1] 324
```

```
library(dplyr)

# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.building_america_climate_zone) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.building_america_climate_zone))
colnames(averages_table) <- c("Category of Weather", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##   Category of Weather Frequency Mean_Value
## 1           Hot-Humid     39336   41.10031
## 2         Mixed-Humid     97704   37.87466
```

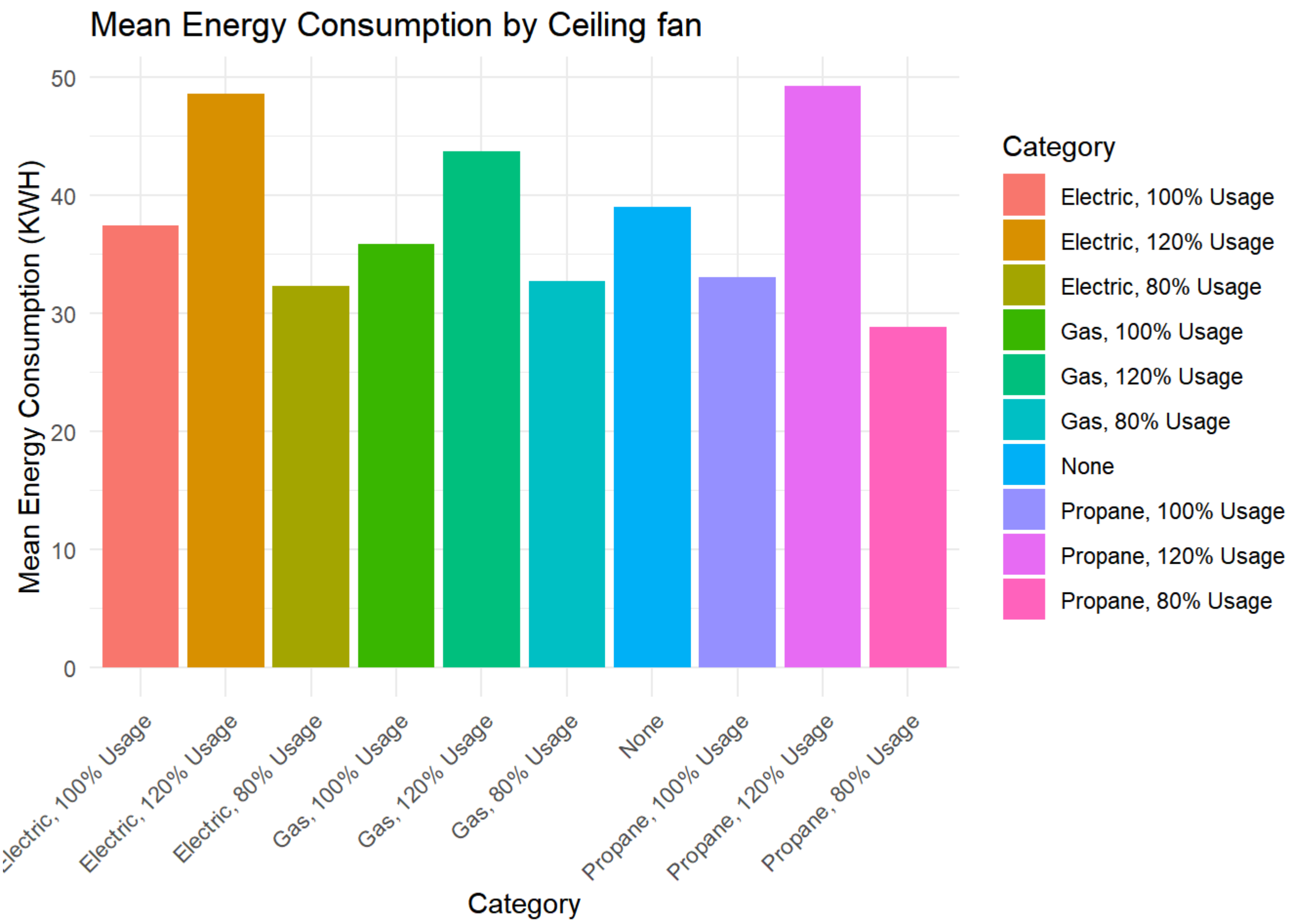
```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.ceiling_fan) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.ceiling_fan))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

#print(averages_table)

ggplot(averages_table, aes(x = Category, y = Mean_Value, fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Mean Energy Consumption by Ceiling fan",

       y = "Mean Energy Consumption (KWH)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.clothes_dryer) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.clothes_dryer))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##                Category Frequency Mean_Value
## 1  Electric, 100% Usage     62280   37.43350
## 2  Electric, 120% Usage     30576   48.62199
## 3   Electric, 80% Usage     30432   32.31367
## 4       Gas, 100% Usage      3480   35.86120
## 5       Gas, 120% Usage      1848   43.73780
## 6        Gas, 80% Usage      1920   32.75009
## 7                  None      5040   38.98246
## 8   Propane, 100% Usage       768   33.04335
## 9   Propane, 120% Usage       264   49.28093
## 10   Propane, 80% Usage       432   28.86421
```

```
ggplot(averages_table, aes(x = Category, y = Mean_Value, fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Mean Energy Consumption by Ceiling fan",

       y = "Mean Energy Consumption (KWH)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
#ommit garages based of consideration of the lighting factor in the variable set instead of g
arage size , can do corr





# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.heating_fuel) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.heating_fuel))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

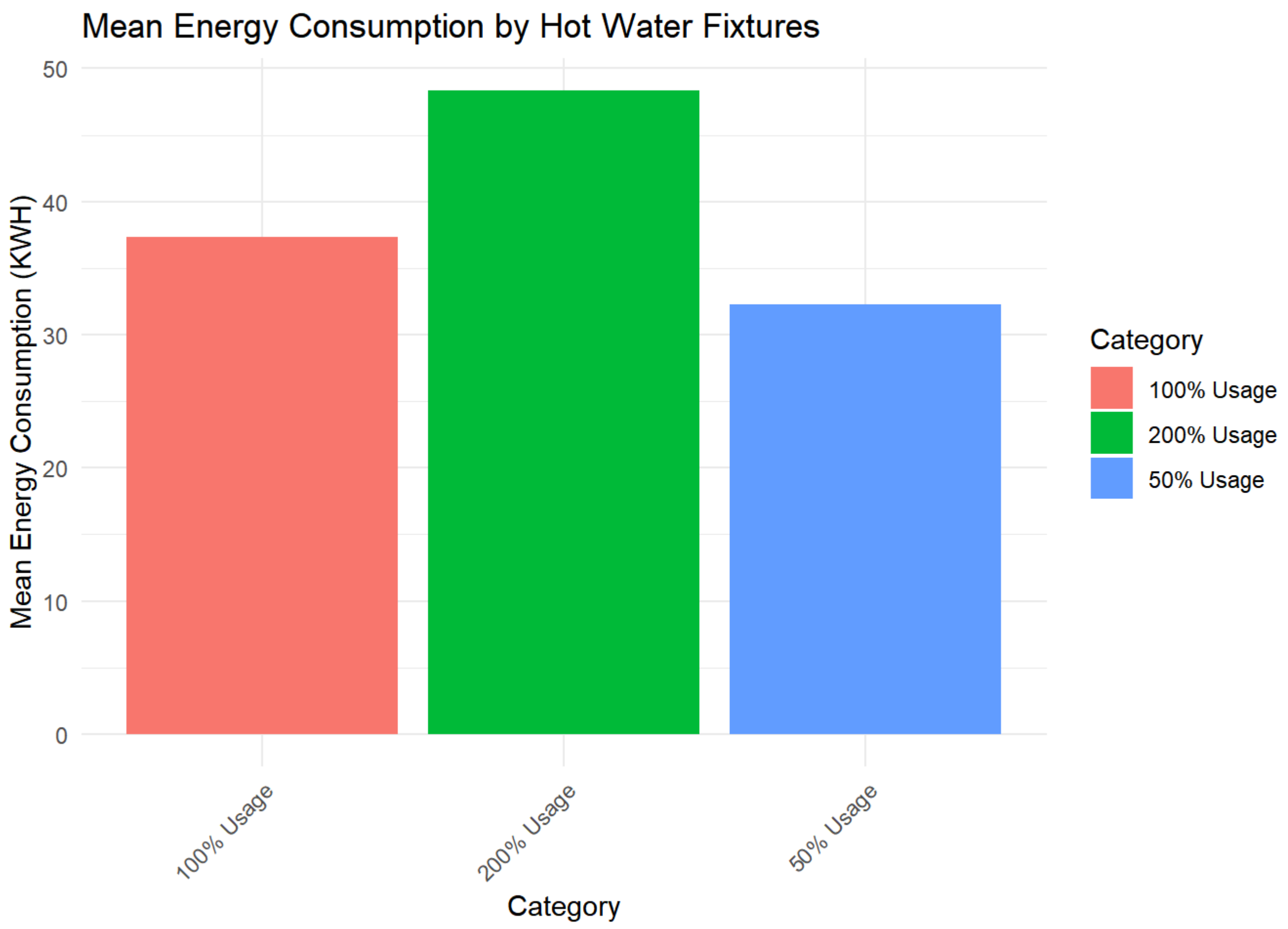
```
##      Category Frequency Mean_Value
## 1 Electricity     87336   39.06592
## 2    Fuel Oil       864   34.62429
## 3 Natural Gas     41112   38.65093
## 4        None        72   48.72160
## 5  Other Fuel      1344   36.58445
## 6     Propane      6312   37.03370
```

```
#
```

```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.hot_water_fixtures) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.hot_water_fixtures))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##     Category Frequency Mean_Value
## 1 100% Usage     69024   37.31091
## 2 200% Usage     33912   48.37150
## 3  50% Usage     34104   32.29840
```

```
ggplot(averages_table, aes(x = Category, y = Mean_Value, fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Mean Energy Consumption by Hot Water Fixtures",

      y = "Mean Energy Consumption (KWH)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



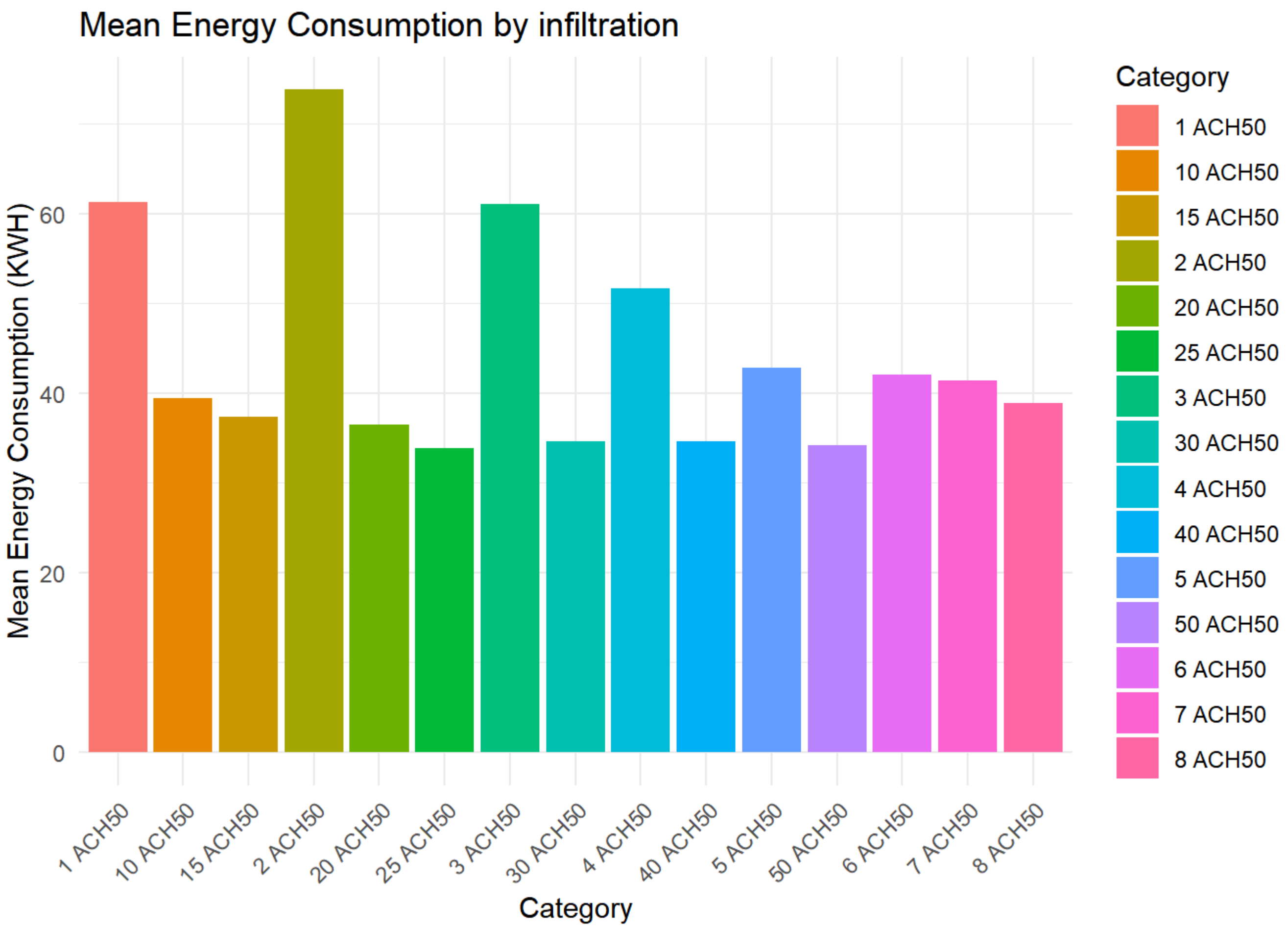
```
Merged_Final <- Merged_Final %>% mutate(in.income = case_when(in.income=='10000-14999'~1,
in.income=='15000-19999'~2,
in.income=='20000-24999'~3,
in.income=='80000-99999'~4,
in.income=='100000-119999'~5,
in.income=='200000+'~6,
in.income=='30000-34999'~7,
in.income=='60000-69999'~8,
in.income=='50000-59999'~9,
in.income=='70000-79999'~10,
in.income=='25000-29999'~11,
in.income=='40000-44999'~12,
in.income=='140000-159999'~13,
in.income=='<10000'~14,
in.income=='45000-49999'~15,
in.income=='35000-39999'~16,
in.income=='120000-139999'~17,
in.income=='160000-179999'~18,
in.income=='180000-199999'~19))

Merged_Final <- Merged_Final %>% mutate(in.income = case_when(in.income <= 6 ~ 1, (in.income
> 6 & in.income <= 12) ~ 2, (in.income > 12 & in.income <= 19) ~ 3))

cor(Merged_Final$Final_Energy_KWH,Merged_Final$in.income)
```

```
## [1] 0.008981471
```

```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.infiltration) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.infiltration))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##    Category Frequency Mean_Value
## 1   1 ACH50       120   61.24603
## 2  10 ACH50     16656   39.38260
## 3  15 ACH50     32880   37.38530
## 4   2 ACH50      1056   73.78386
## 5  20 ACH50     20952   36.43773
## 6  25 ACH50     12240   33.87443
## 7   3 ACH50      2256   61.01379
## 8  30 ACH50      7464   34.64684
## 9   4 ACH50      4440   51.61582
## 10 40 ACH50      6312   34.62654
## 11  5 ACH50      6072   42.84745
## 12 50 ACH50      2808   34.12893
## 13  6 ACH50      7320   42.02966
## 14  7 ACH50      8088   41.42799
## 15  8 ACH50      8376   38.91980
```

```
ggplot(averages_table, aes(x = Category, y = Mean_Value, fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Mean Energy Consumption by infiltration",

      y = "Mean Energy Consumption (KWH)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.occupants) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.occupants))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##    Category Frequency Mean_Value
## 1         1     30672   32.39219
## 2       10+       192   72.60214
## 3         2     52536   37.10760
## 4         3     22440   40.61320
## 5         4     18264   44.18862
## 6         5      8064   48.04031
## 7         6      2760   50.72124
## 8         7      1392   54.67280
## 9         8       552   51.67456
## 10        9       168   58.52462
```

```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.vintage) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.vintage
))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##   Category Frequency Mean_Value
## 1    <1940      7608   33.21512
## 2    1940s      5448   34.78764
## 3    1950s     13128   37.48521
## 4    1960s     15696   37.26652
## 5    1970s     20040   39.57342
## 6    1980s     16680   39.01469
## 7    1990s     20160   42.17735
## 8    2000s     26712   44.15983
## 9    2010s     11568   34.21231
```

```
#in.misc_gas_fireplace  in.misc_gas_grill   in.misc_gas_lighting    in.misc_hot_tub_spa in.mi
sc_pool    in.misc_pool_heater
#not significant due to small sample size
```

```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.water_heater_efficiency) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.water_heater_efficiency
))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##                      Category Frequency Mean_Value
## 1  Electric Heat Pump, 80 gal       408   35.14512
## 2            Electric Premium      8856   38.95527
## 3           Electric Standard     78792   38.78470
## 4           Electric Tankless      1248   47.09146
## 5           Fuel Oil Standard        72   41.59829
## 6         Natural Gas Premium      3888   40.70619
## 7        Natural Gas Standard     38592   38.42177
## 8        Natural Gas Tankless       576   38.94907
## 9                  Other Fuel       552   43.63248
## 10            Propane Premium       312   38.37540
## 11           Propane Standard      3360   37.14984
## 12           Propane Tankless       384   41.29168
```

```
# Calculate average based on category
averages <- Merged_Final %>%
  group_by(in.window_areas) %>%
  summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))

# Display table with averages
averages_table <- as.data.frame(table(Merged_Final$in.window_areas
))
colnames(averages_table) <- c("Category", "Frequency")
averages_table$Mean_Value <- averages$mean_value

print(averages_table)
```

```
##          Category Frequency Mean_Value
## 1 F12 B12 L12 R12     35280   38.10097
## 2 F15 B15 L15 R15     22464   39.46577
## 3 F18 B18 L18 R18     21480   40.08437
## 4 F30 B30 L30 R30      5424   43.43864
## 5     F6 B6 L6 R6     12936   37.85704
## 6     F9 B9 L9 R9     39456   38.02018
```

```
#--------------------------Blanks
# # Calculate average based on category
# averages <- Merged_Final %>%
#   group_by(upgrade.water_heater_efficiency) %>%
#   summarise(mean_value = mean(Final_Energy_KWH, na.rm = TRUE))
#
# # Display table with averages
# averages_table <- as.data.frame(table(Merged_Final$upgrade.water_heater_efficiency
# ))
# colnames(averages_table) <- c("Category", "Frequency")
# averages_table$Mean_Value <- averages$mean_value
#
# print(averages_table)
#
```

We scarped all the weather data. . All the weather data was numeric and we averaged it out on an hourly basis in july . This data was available on a county basis. We saved it in “aggregate_hourly_cdw.xlsx” # Final_Dataset<- merge(aggregate_hourly_cdw,merged_house_Static_energy , by = c(“in.county”,“hour”), all = TRUE)

```
# countys<- unique(merged_house_Static_energy$in.county)
# links_countys <- paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/weath
er/2023-weather-data/", countys, ".csv")
# links_countys
# data_df_countys<- data.frame(countys = countys, links_countys  = links_countys)
#
#
# # Assuming data_df dataframe is created with bldg_id and link columns
# library(httr)
# # Create an empty list to store data frames
# parquet_data_countys <- list()
# x<-(nrow(data_df_countys))
#
# # Loop through each link and read Parquet files
# for (i in 1:x) {
#     link <- as.character(data_df_countys[i, "links_countys"])
#
#     county <- as.character(data_df_countys[i, "countys"])
#
#
# # Read the Parquet file into a dataframe
#   df <- read_csv(link)
#
#
#     # Assign bldg_id to the first column
#    df$county<- county
#    #df<-df%>%filter(month(energy_data$date_time)==7)
#     # Add the dataframe to the list
#    parquet_data_countys[[i]] <- df
#    cat("Progress: ", i, "/",x, "\n")
#
# }
#
# combined_data_weather <- do.call(rbind,  parquet_data_countys)
# combined_data_weather<-combined_data_weather%>% filter(month(combined_data_weather$date_tim
e)==7)
# head(combined_data_weather)
# combined_data_weather$hour<-hour(combined_data_weather$date_time)
#
# aggregate_hourly_cdw<-combined_data_weather%>%group_by(county,hour)%>%summarize(across(wher
e(is.numeric), mean))
# write_xlsx(aggregate_hourly_cdw,"aggregate_hourly_cdw.xlsx")
```

We merged the tow datasets based of county and hour as the weather data was at that geanularity on aggregating by hour for the month of july This file has been saved as “output_file.parquet”

```
# library(readxl)
# library(writexl)
# library(arrow)
aggregate_hourly_cdw<-read_xlsx("aggregate_hourly_cdw.xlsx")
str(aggregate_hourly_cdw)
```

```
## tibble [1,104 × 9] (S3: tbl_df/tbl/data.frame)
##  $ in.county                         : chr [1:1104] "G4500010" "G4500010" "G4500010" "G45
00010" ...
##  $ hour                              : num [1:1104] 0 1 2 3 4 5 6 7 8 9 ...
##  $ Dry Bulb Temperature [°C]         : num [1:1104] 22.4 22.1 21.8 21.6 21.5 ...
##  $ Relative Humidity [%]             : num [1:1104] 95.2 95.7 96.6 96.9 96.9 ...
##  $ Wind Speed [m/s]                  : num [1:1104] 1.089 0.932 0.978 0.729 0.956 ...
##  $ Wind Direction [Deg]              : num [1:1104] 125.6 104.2 127.4 86 83.5 ...
##  $ Global Horizontal Radiation [W/m2] : num [1:1104] 0 0 0 0 0 ...
##  $ Direct Normal Radiation [W/m2]     : num [1:1104] 0 0 0 0 0 ...
##  $ Diffuse Horizontal Radiation [W/m2]: num [1:1104] 0 0 0 0 0 ...
```

```
# merged_house_Static_energy<-read_xlsx("merged_house_Static_energy.xlsx")
#
# Final_Dataset<- merge(aggregate_hourly_cdw,merged_house_Static_energy , by = c("in.count
y","hour"), all = TRUE)
#
# head(Final_Dataset)
#
# write_parquet(Final_Dataset, "output_file.parquet")
```

We did the same out put coloumn summation we did for our cleaning here and saved it finally into one last file called Aggregate_Final_Dataset.parquet for save time. (eachof this scraping and cleaning iteration was taking 1hour vs 3 minutes, on saving each stage into a parquet)

```r
# library(arrow)
# library(tidyverse)
# Final_Dataset<-read_parquet("output_file.parquet")
#
# # Select columns starting with "out"
# grep("out.", names(Final_Dataset))
# out_cols <- c(grep("out.", names(Final_Dataset)))
# out_cols
#
# # View the selected columns
# Aggregate_Final_Dataset<-Final_Dataset
# Aggregate_Final_Dataset$Final_Energy_KWH<- Final_Dataset %>%select(starts_with("out")) %>%
rowSums(na.rm = TRUE)# Displaying the first few rows of the selected columns
# head(Aggregate_Final_Dataset)
# Aggregate_Final_Dataset<- Aggregate_Final_Dataset[, -out_cols]
# glimpse(Aggregate_Final_Dataset)
#
# write_parquet(Aggregate_Final_Dataset, "Aggregate_Final_Dataset.parquet")
```

```r
library(tidyverse)
library(arrow)
Aggregate_Final_Dataset<-read_parquet("Aggregate_Final_Dataset.parquet")
glimpse(Aggregate_Final_Dataset)
```

```
## Rows: 137,040
## Columns: 102
## $ in.county                                  <chr> "G4500010", "G4500010", "G4…
## $ hour                                       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ `Dry Bulb Temperature [°C]`                <dbl> 22.35581, 22.35581, 22.3558…
## $ `Relative Humidity [%]`                    <dbl> 95.18613, 95.18613, 95.1861…
## $ `Wind Speed [m/s]`                         <dbl> 1.089355, 1.089355, 1.08935…
## $ `Wind Direction [Deg]`                     <dbl> 125.5919, 125.5919, 125.591…
## $ `Global Horizontal Radiation [W/m2]`       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ `Direct Normal Radiation [W/m2]`           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ `Diffuse Horizontal Radiation [W/m2]`      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ bldg_id                                    <dbl> 410602, 465218, 473719, 299…
## $ in.sqft                                    <dbl> 1220, 2176, 3301, 2663, 169…
## $ in.bathroom_spot_vent_hour                 <chr> "Hour20", "Hour11", "Hour4"…
## $ in.bedrooms                                <dbl> 4, 4, 5, 3, 3, 4, 3, 4, 3, …
## $ in.building_america_climate_zone           <chr> "Mixed-Humid", "Mixed-Humid…
## $ in.ceiling_fan                             <chr> "Standard Efficiency", "Sta…
## $ in.city                                    <chr> "In another census Place", …
## $ in.clothes_dryer                           <chr> "Electric, 120% Usage", "Ga…
## $ in.clothes_washer                          <chr> "EnergyStar, 120% Usage", "…
## $ in.clothes_washer_presence                 <chr> "Yes", "Yes", "Yes", "Yes",…
## $ in.cooking_range                           <chr> "Electric, 120% Usage", "El…
## $ in.cooling_setpoint                        <chr> "75F", "70F", "75F", "75F",…
## $ in.cooling_setpoint_has_offset             <chr> "No", "No", "No", "No", "Ye…
## $ in.cooling_setpoint_offset_magnitude       <chr> "0F", "0F", "0F", "0F", "9F…
## $ in.cooling_setpoint_offset_period          <chr> "None", "None", "None", "No…
## $ in.county_and_puma                         <chr> "G4500010, G45001600", "G45…
## $ in.dishwasher                              <chr> "290 Rated kWh, 120% Usage"…
## $ in.ducts                                   <chr> "20% Leakage, R-4", "20% Le…
## $ in.federal_poverty_level                   <chr> "300-400%", "150-200%", "40…
## $ in.geometry_attic_type                     <chr> "Vented Attic", "Vented Att…
## $ in.geometry_floor_area                     <chr> "1000-1499", "2000-2499", "…
## $ in.geometry_floor_area_bin                 <chr> "0-1499", "1500-2499", "250…
## $ in.geometry_foundation_type                <chr> "Slab", "Slab", "Slab", "Sl…
## $ in.geometry_garage                         <chr> "None", "2 Car", "2 Car", "…
## $ in.geometry_stories                        <dbl> 1, 1, 2, 1, 2, 2, 1, 2, 1, …
## $ in.geometry_stories_low_rise               <dbl> 1, 1, 2, 1, 2, 2, 1, 2, 1, …
## $ in.geometry_wall_exterior_finish           <chr> "Wood, Medium/Dark", "Brick…
## $ in.geometry_wall_type                      <chr> "Wood Frame", "Wood Frame",…
## $ in.has_pv                                  <chr> "No", "No", "No", "No", "No…
## $ in.heating_fuel                            <chr> "Electricity", "Electricity…
## $ in.heating_setpoint                        <chr> "70F", "72F", "65F", "55F",…
## $ in.heating_setpoint_has_offset             <chr> "Yes", "Yes", "No", "No", "…
## $ in.heating_setpoint_offset_magnitude       <chr> "3F", "3F", "0F", "0F", "3F…
## $ in.heating_setpoint_offset_period          <chr> "Night", "Day and Night -4h…
## $ in.hot_water_fixtures                      <chr> "200% Usage", "100% Usage",…
## $ in.hvac_cooling_efficiency                 <chr> "AC, SEER 15", "Heat Pump",…
## $ in.hvac_cooling_partial_space_conditioning <chr> "100% Conditioned", "100% C…
## $ in.hvac_cooling_type                       <chr> "Central AC", "Heat Pump", …
## $ in.hvac_has_ducts                          <chr> "Yes", "Yes", "Yes", "Yes",…
## $ in.hvac_has_zonal_electric_heating         <chr> "No", "No", "No", "No", "No…
## $ in.hvac_heating_efficiency                 <chr> "Electric Furnace, 100% AFU…
## $ in.hvac_heating_type                       <chr> "Ducted Heating", "Ducted H…
## $ in.hvac_heating_type_and_fuel              <chr> "Electricity Electric Furna…
## $ in.income                                  <chr> "45000-49999", "50000-59999…
## $ in.income_recs_2015                        <chr> "40000-59999", "40000-59999…
## $ in.income_recs_2020                        <chr> "40000-59999", "40000-59999…
## $ in.infiltration                            <chr> "15 ACH50", "25 ACH50", "4 …
## $ in.insulation_ceiling                      <chr> "R-30", "R-30", "R-7", "R-3…
## $ in.insulation_floor                        <chr> "None", "None", "None", "No…
## $ in.insulation_foundation_wall              <chr> "None", "None", "None", "No…
## $ in.insulation_rim_joist                    <chr> "None", "None", "None", "No…
## $ in.insulation_roof                         <chr> "Unfinished, Uninsulated", …
## $ in.insulation_slab                         <chr> "Uninsulated", "2ft R10 Und…
## $ in.insulation_wall                         <chr> "Wood Stud, Uninsulated", "…
## $ in.lighting                                <chr> "100% Incandescent", "100% …
## $ in.misc_extra_refrigerator                 <chr> "EF 15.9", "None", "None", …
## $ in.misc_freezer                            <chr> "None", "EF 12, National Av…
## $ in.misc_gas_fireplace                      <chr> "None", "None", "None", "No…
## $ in.misc_gas_grill                          <chr> "Gas Grill", "None", "None"…
## $ in.misc_gas_lighting                       <chr> "None", "None", "None", "No…
## $ in.misc_hot_tub_spa                        <chr> "None", "None", "None", "El…
## $ in.misc_pool                               <chr> "None", "None", "None", "No…
## $ in.misc_pool_heater                        <chr> "None", "None", "None", "No…
## $ in.misc_pool_pump                          <chr> "None", "None", "None", "No…
## $ in.misc_well_pump                          <chr> "None", "None", "None", "No…
## $ in.occupants                               <chr> "1", "5", "4", "2", "2", "2…
## $ in.orientation                             <chr> "West", "South", "East", "N…
## $ in.plug_load_diversity                     <chr> "200%", "100%", "50%", "100…
## $ in.puma                                    <chr> "G45001600", "G45001600", "…
## $ in.puma_metro_status                       <chr> "Not/partially in metro are…
## $ in.pv_orientation                          <chr> "None", "None", "None", "No…
## $ in.pv_system_size                          <chr> "None", "None", "None", "No…
## $ in.range_spot_vent_hour                    <chr> "Hour9", "Hour19", "Hour2",…
## $ in.reeds_balancing_area                    <dbl> 95, 95, 95, 95, 95, 95, 95,…
## $ in.refrigerator                            <chr> "EF 17.6, 100% Usage", "EF …
## $ in.roof_material                           <chr> "Composition Shingles", "Wo…
## $ in.tenure                                  <chr> "Owner", "Renter", "Owner",…
## $ in.usage_level                             <chr> "High", "Medium", "Low", "M…
## $ in.vacancy_status                          <chr> "Occupied", "Occupied", "Oc…
## $ in.vintage                                 <chr> "1960s", "2000s", "1970s", …
## $ in.vintage_acs                             <chr> "1960-79", "2000-09", "1960…
## $ in.water_heater_efficiency                 <chr> "Electric Standard", "Elect…
## $ in.water_heater_fuel                       <chr> "Electricity", "Electricity…
## $ in.weather_file_city                       <chr> "Greenwood Co", "Greenwood …
## $ in.weather_file_latitude                   <dbl> 34.25, 34.25, 34.25, 34.25,…
## $ in.weather_file_longitude                  <dbl> -82.16, -82.16, -82.16, -82…
## $ in.window_areas                            <chr> "F18 B18 L18 R18", "F12 B12…
## $ in.windows                                 <chr> "Single, Clear, Metal", "Do…
## $ upgrade.water_heater_efficiency            <chr> "Electric Heat Pump, 66 gal…
## $ upgrade.clothes_dryer                      <chr> "Electric, Premium, Heat Pu…
## $ upgrade.hvac_heating_efficiency            <chr> "MSHP, SEER 24, 13 HSPF", "…
## $ upgrade.cooking_range                      <chr> "Electric, Induction, 120% …
## $ Final_Energy_KWH                           <dbl> 24.89468, 35.97000, 18.9830…
```

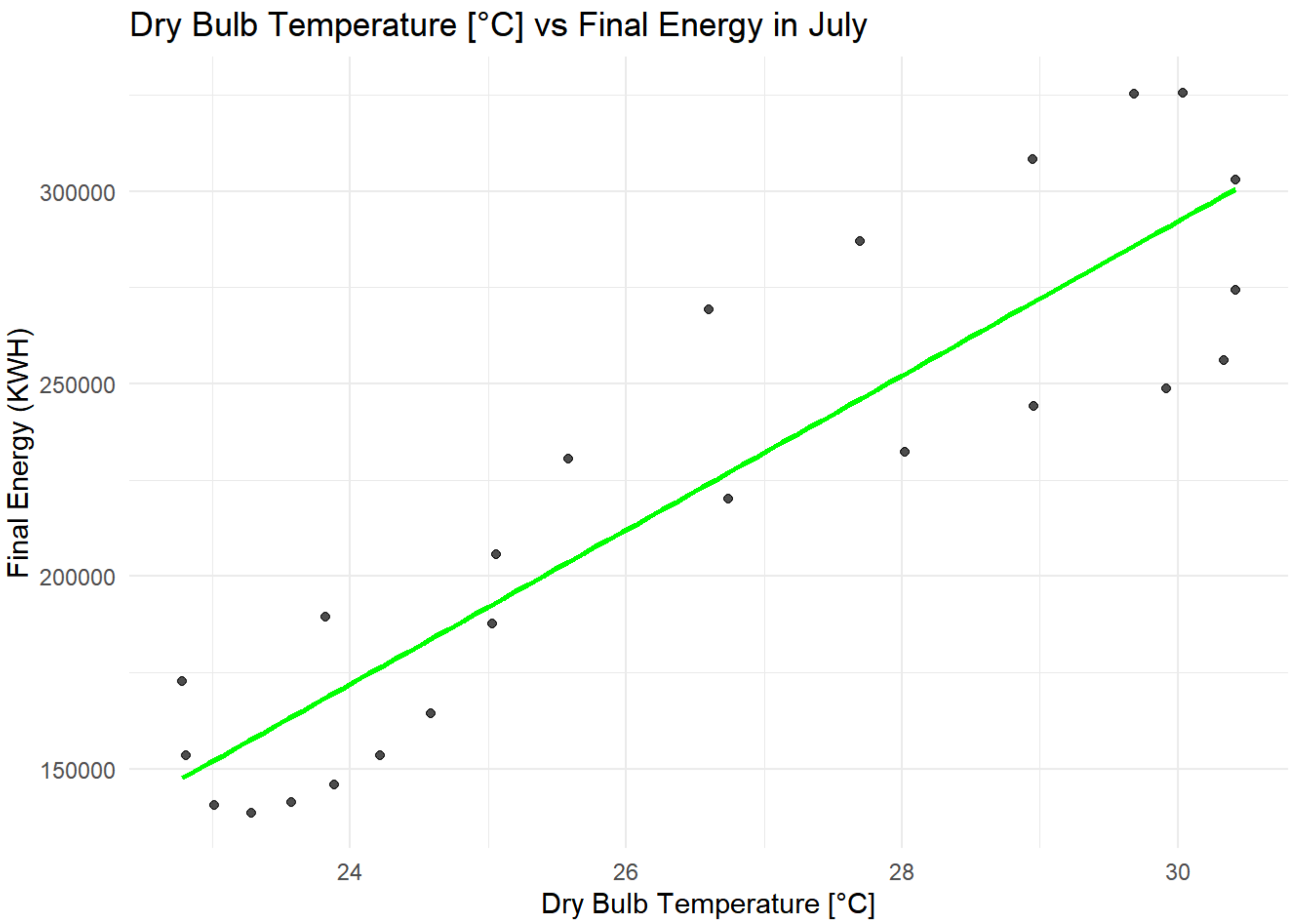
```
Weather_Energy<- Aggregate_Final_Dataset%>% group_by(hour)%>%select(hour,Final_Energy_KWH,`Dr
y Bulb Temperature [°C]`,`Relative Humidity [%]` ,`Wind Direction [Deg]` ,`Global Horizontal
Radiation [W/m2]` ,`Direct Normal Radiation [W/m2]`   ,`Diffuse Horizontal Radiation [W/m2]`)
%>%summarise(across(where(is.numeric) & !matches("Final_Energy_KWH"), mean, na.rm = TRUE),
            Final_Energy_KWH = sum(Final_Energy_KWH, na.rm = TRUE))
```

```
head(meta_data)
```

```
## # A tibble: 6 × 7
##   field_location field_name                      data_type units field_description
##   <chr>          <chr>                           <chr>     <chr> <chr>
## 1 metadata       in.ahs_region                   string    n/a   American Housing…
## 2 metadata       in.ashrae_iecc_climate_zone_… string    n/a   IECC climate zone
## 3 metadata       in.ashrae_iecc_climate_zone_… string    n/a   IECC climate zon…
## 4 metadata       in.bathroom_spot_vent_hour    string    n/a   Bathroom spot ve…
## 5 metadata       in.bedrooms                     integer   n/a   Number of bedroo…
## 6 metadata       in.building_america_climate_… string    n/a   Building America…
## # ℹ 2 more variables: allowable_enumerations_baseline <chr>, ...7 <chr>
```

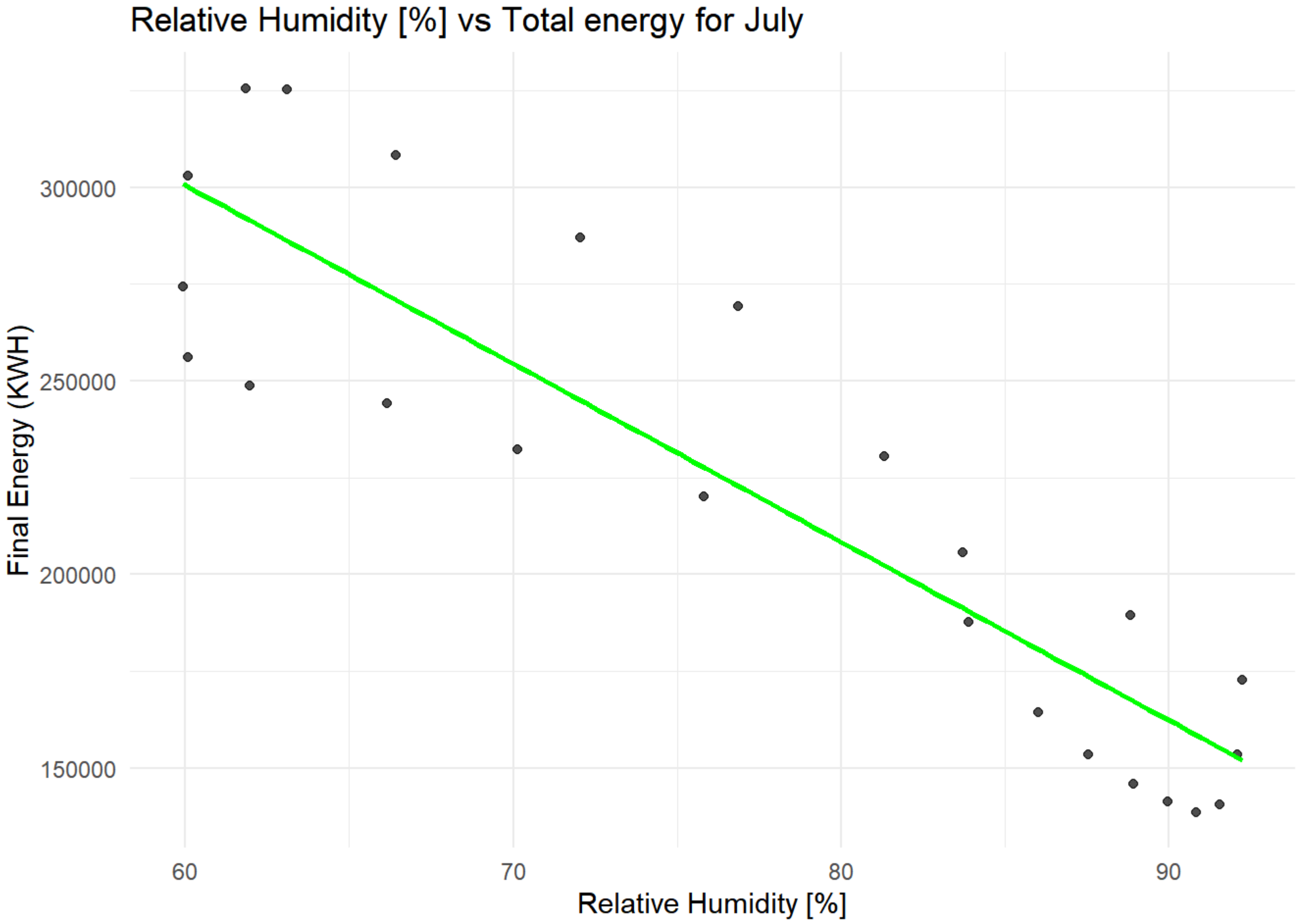
```
ggplot(data = Weather_Energy, aes(x = `Dry Bulb Temperature [°C]`, y = Final_Energy_KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Dry Bulb Temperature [°C]", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Dry Bulb Temperature [°C] vs Final Energy in July") +  # Title of the plot
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



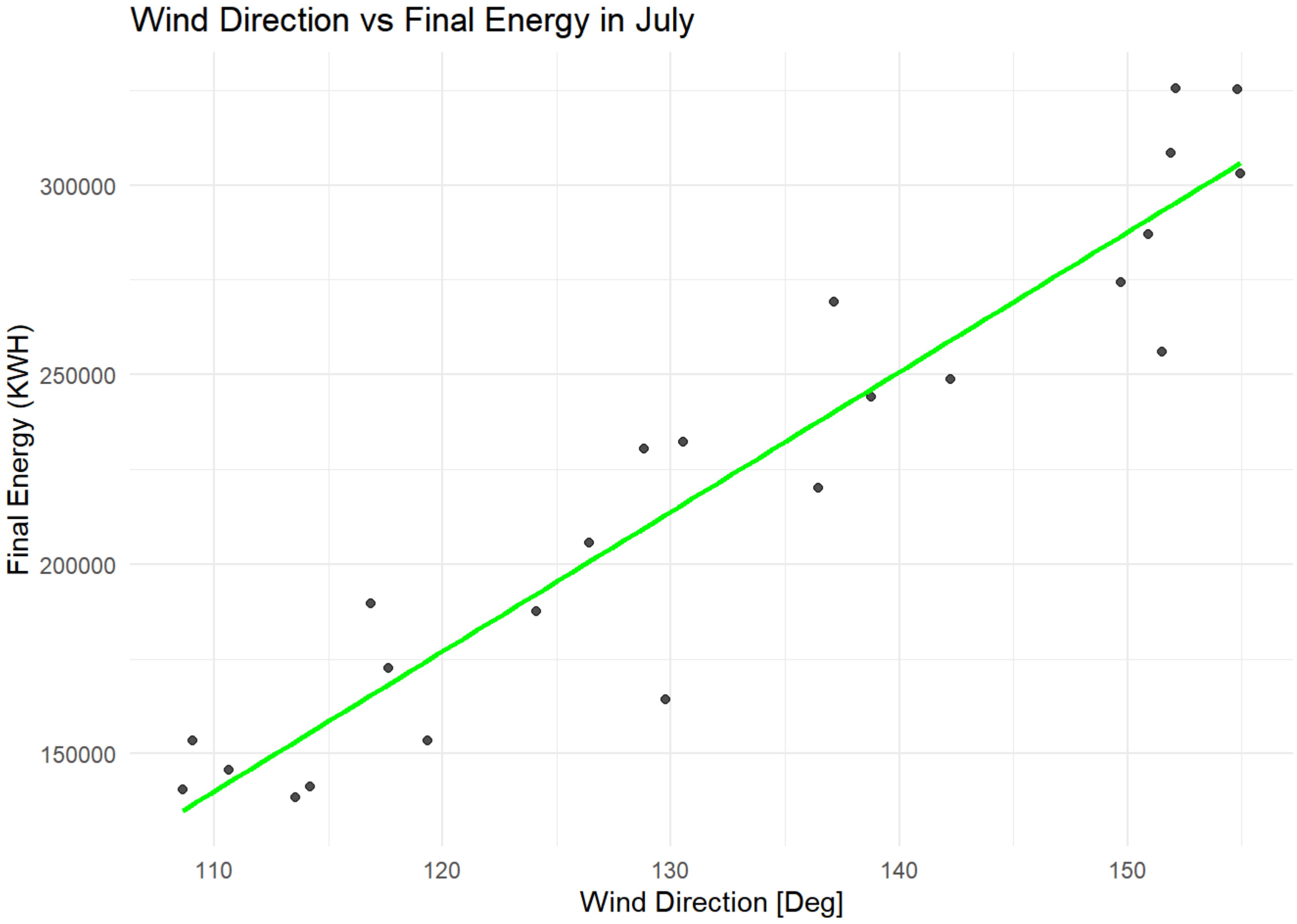
```
ggplot(data = Weather_Energy, aes(x = `Relative Humidity [%]`, y = Final_Energy_KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Relative Humidity [%]", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Relative Humidity [%] vs Total energy for July") +  # Title of the plot
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



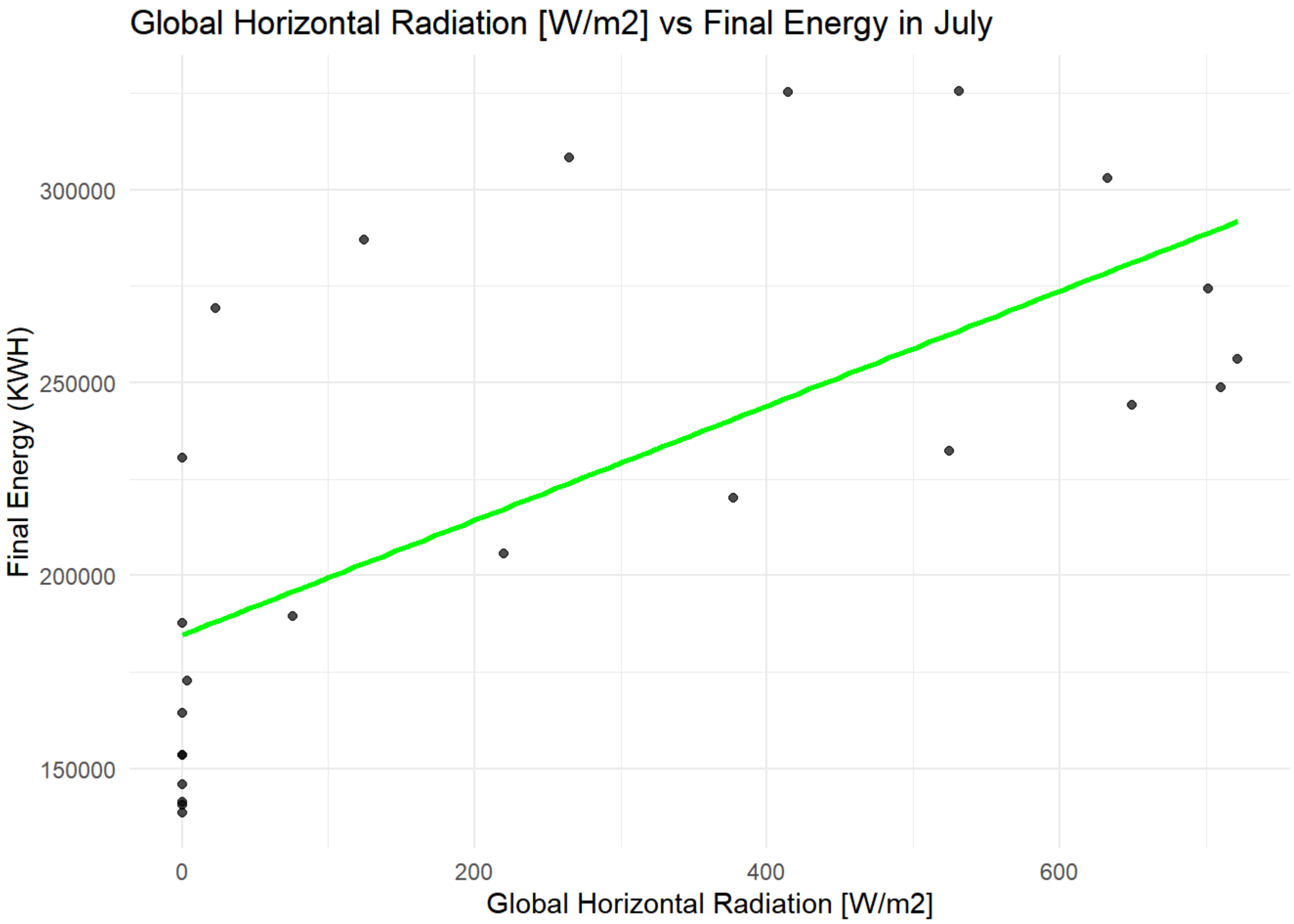
```
ggplot(data = Weather_Energy, aes(x = `Wind Direction [Deg]`, y = Final_Energy_KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Wind Direction [Deg]", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Wind Direction vs Final Energy in July") +  # Title of the plot
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



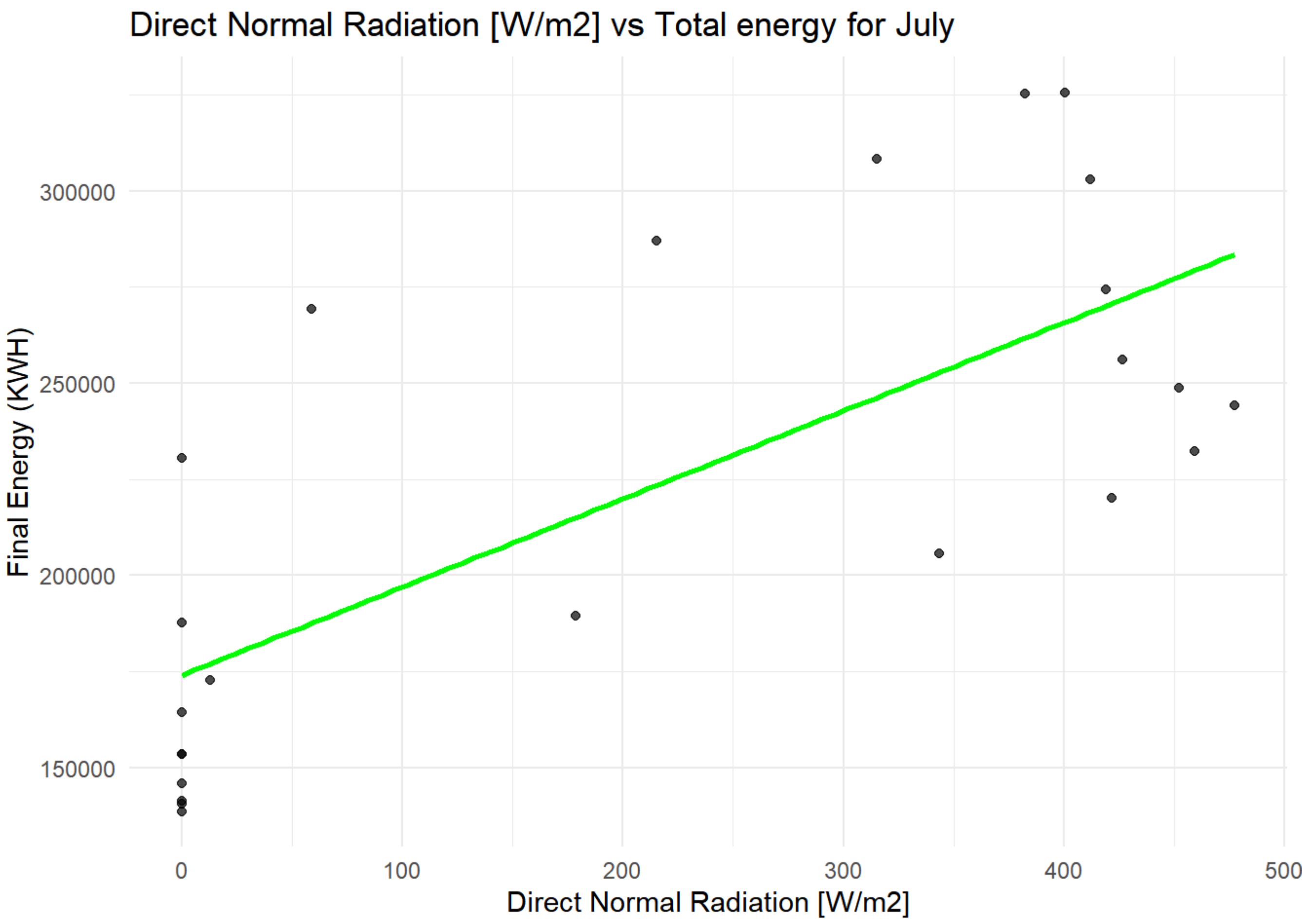
```
ggplot(data = Weather_Energy, aes(x = `Global Horizontal Radiation [W/m2]`, y = Final_Energy_
KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Global Horizontal Radiation [W/m2]", y = "Final Energy (KWH)") +  # Labels for ax
es
  ggtitle("Global Horizontal Radiation [W/m2] vs Final Energy in July") +  # Title of the plo
t
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



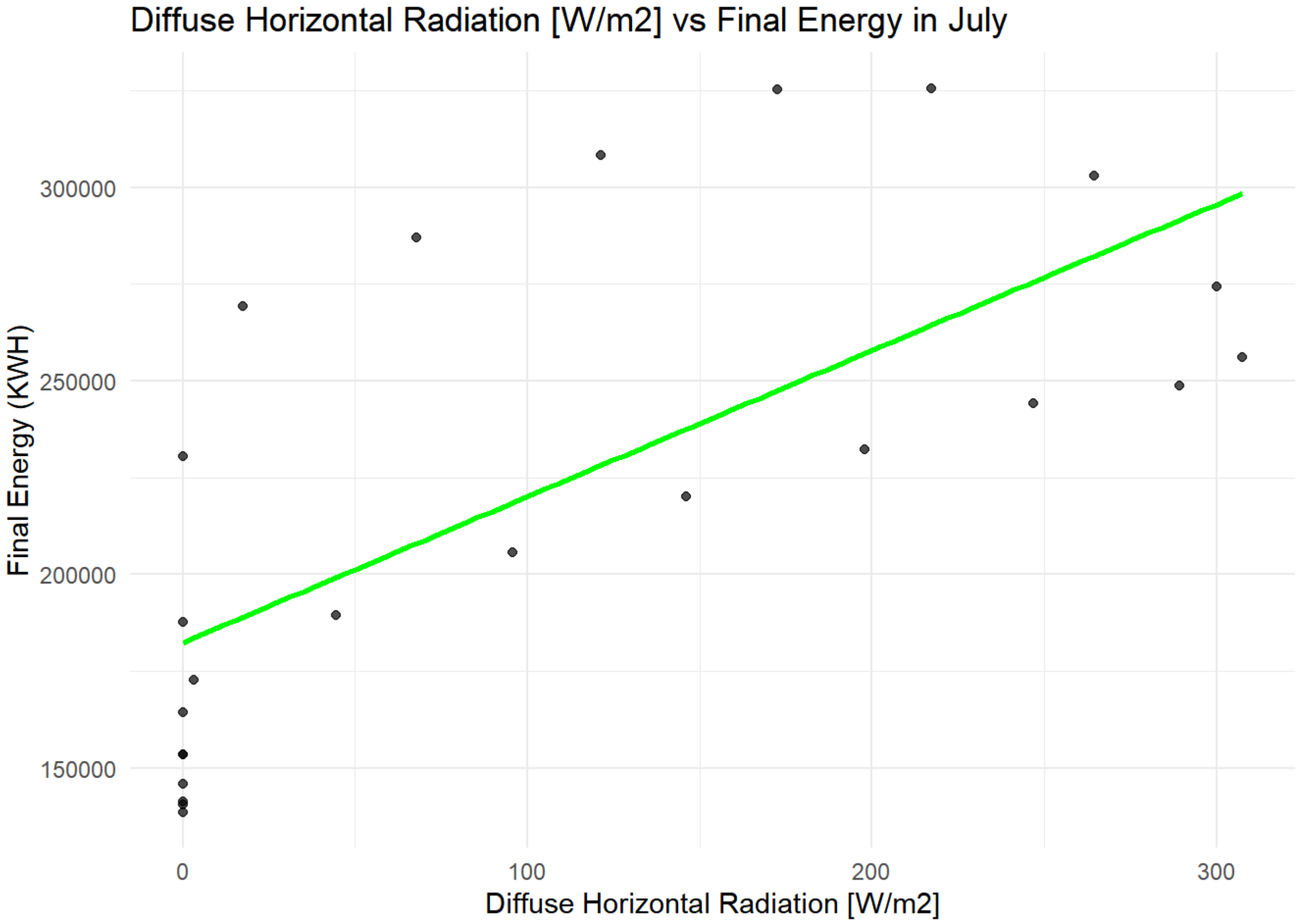
```
ggplot(data = Weather_Energy, aes(x = `Direct Normal Radiation [W/m2]`, y = Final_Energy_KW
H)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Direct Normal Radiation [W/m2]", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Direct Normal Radiation [W/m2] vs Total energy for July") +  # Title of the plot
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

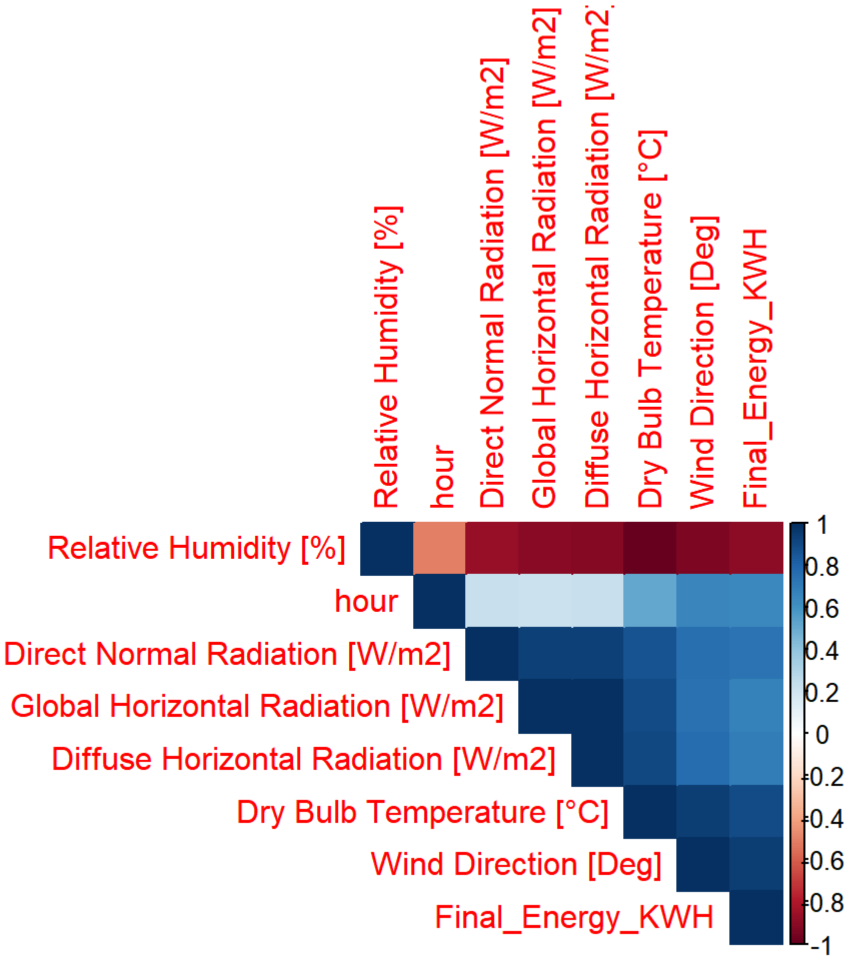


```
ggplot(data = Weather_Energy, aes(x = `Diffuse Horizontal Radiation [W/m2]`, y = Final_Energy_KWH)) +
  geom_point(alpha = 0.7) +  # Adding transparency to points
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Adding linear trend line
  labs(x = "Diffuse Horizontal Radiation [W/m2]", y = "Final Energy (KWH)") +  # Labels for axes
  ggtitle("Diffuse Horizontal Radiation [W/m2] vs Final Energy in July") +  # Title of the plot
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
library(corrplot)
correlation_matrix <- cor(Weather_Energy)



# Plotting the filtered correlation matrix using corrplot
corrplot(correlation_matrix, method = "color", type = "upper",
         order = "hclust", addrect = 2)  # Adjust parameters as needed # Adjust parameters as
needed
```



```
# $ `Dry Bulb Temperature [°C]`            <dbl> 22.35581, 22.35581, 22.35581, 22.35581,
22.35581, 22.35581, 22.35581, 22.35581, 22.35…
# $ `Relative Humidity [%]`                <dbl> 95.18613, 95.18613, 95.18613, 95.18613,
95.18613, 95.18613, 95.18613, 95.18613, 95.18…
# $ `Wind Speed [m/s]`                     <dbl> 1.089355, 1.089355, 1.089355, 1.089355,
1.089355, 1.089355, 1.089355, 1.089355, 1.089…
# $ `Wind Direction [Deg]`                 <dbl> 125.5919, 125.5919, 125.5919, 125.5919,
125.5919, 125.5919, 125.5919, 125.5919, 125.5…
# $ `Global Horizontal Radiation [W/m2]`   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
# $ `Direct Normal Radiation [W/m2]`    ,
# $ `Diffuse Horizontal Radiation [W/m2]`
```